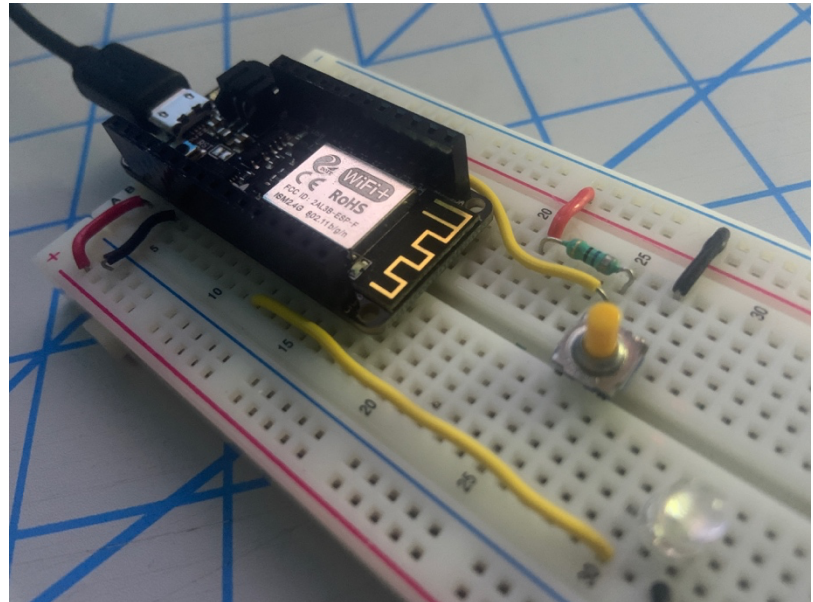


Adafruit 8266 Feather Huzzah Guide 2: networking basics

In the last guide we finished with a working circuit consisting of a simple button and led. But the real purpose was to set up the necessary dependencies and libraries to program the Feather using the Arduino IDE. In this guide we will cover all of the steps necessary to

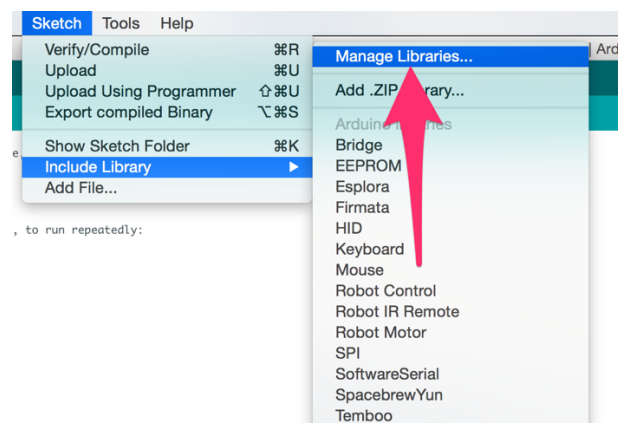


connect your Feather to Adafruit IO and then connect that live data to a basic P5.js sketch uploaded to your Firebird server.

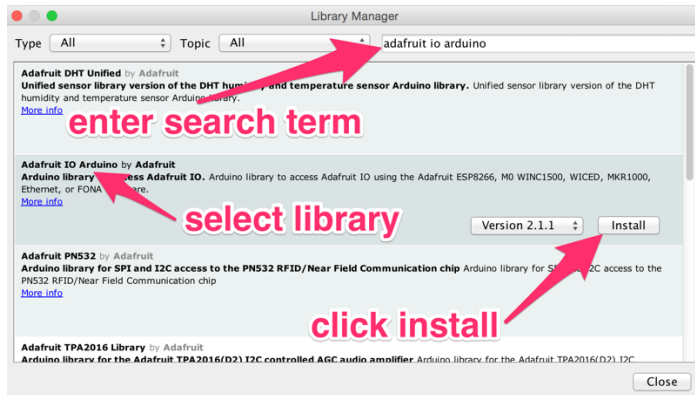
Step 1:

We will need to install the libraries that will allow the Feather to communicate with the Adafruit network.

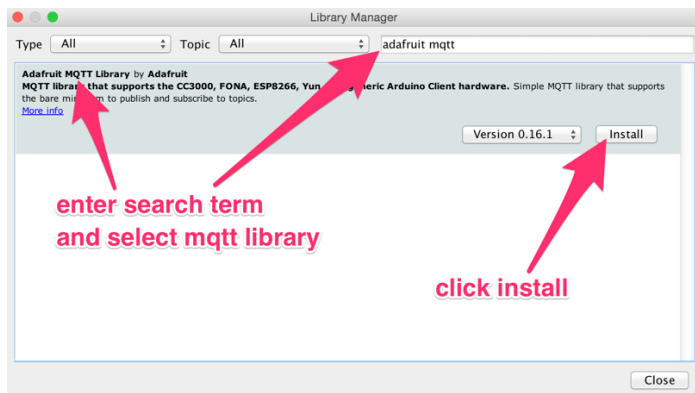
In the Tools menu click on Manage Libraries



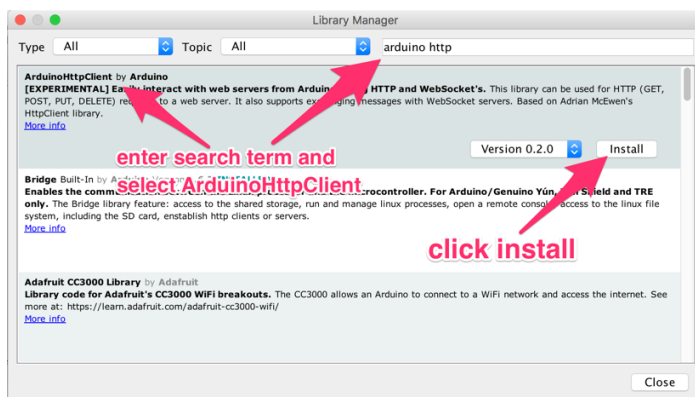
This will open the Library Manager, from here you can add specific libraries, this is likely not your first time using this feature. Start typing 'Adafruit IO Arduino' and the correct library should display. Install the most recent Version available..



Next search for 'Adafruit MQTT', get the most recent version and click install.



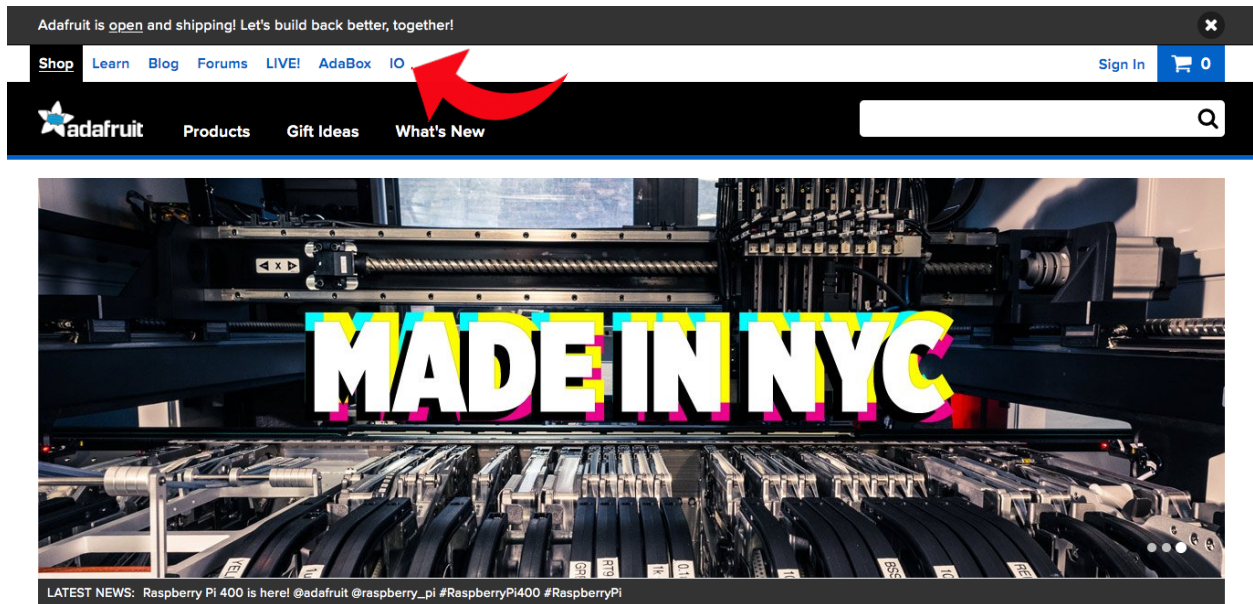
Next we want to install the 'Arduino HTTP Client'.



Step 3:

Let's setup Adafruit IO so your Feather can send its data somewhere.

In your browser navigate to: [https:// www.Adafruit.com](https://www.Adafruit.com), type Adafruit.com in the address bar, input Adafruit.com in a search. Once there click on IO



On the Adafruit IO page you will see a link to 'Get Started For Free', click the link



Create a free account.

SIGN UP

The best way to shop with Adafruit is to create an account which allows you to shop faster, track the status of your current orders, review your previous orders and take advantage of our other member benefits.

FIRST

LAST NAME

EMAIL

USERNAME

Username is viewable to the public on the forums, Adafruit IO, and elsewhere.

PASSWORD

[CREATE ACCOUNT](#)

HAVE AN ADAFRUIT ACCOUNT?

[SIGN IN](#)

Sign in to your new account using the username and password you just created and you will find yourself at your Dashboard, this is where all of your Feather feeds will be displayed.

Click on My Key.

The image shows two parts of the Adafruit IO website. On the left is the 'SIGN IN' page with fields for 'EMAIL OR USERNAME' and 'PASSWORD', and buttons for 'SIGN IN' and 'NEED AN ADAFRUIT ACCOUNT? SIGN UP'. On the right is the 'My Key' page, which displays a table of dashboards and a section for 'YOUR ADAFRUIT IO KEY'. The 'My Key' section includes a QR code, a 'REGENERATE KEY' button, and code snippets for Arduino, Linux Shell, and Scripting. Red arrows indicate the flow from the 'My Key' link in the top navigation bar to the 'YOUR ADAFRUIT IO KEY' section.

Name	Key	Created At
button	button	January 16, 2020
button	button	May 16, 2020

```
#define IO_USERNAME "adafruit"
#define IO_KEY "adafruit"

export IO_USERNAME="adafruit"
export IO_KEY="adafruit"

ADAFRUIT_IO_USERNAME = "adafruit"
ADAFRUIT_IO_KEY = "adafruit"
```

This will open a page with your confidential key information, keep this information to yourself, we will need some specific information here for the code we will upload to the Feather so keep it handy.

Click Feeds then click on New Feed, then view all, name it button, and click Create and name the feed 'button'.

The image shows the 'Feeds' page on the Adafruit IO website. The top navigation bar includes 'Shop', 'Learn', 'Blog', 'Forums', 'LIVE!', 'AdaBox', and 'IO'. The 'Feeds' tab is selected. Below the navigation bar, there is a 'Dashboards' section with a table of dashboards. A red arrow points to the 'Feeds' tab. Another red arrow points to the 'view all' link under the 'Feeds' section. A third red arrow points to the '+ New Feed' button. The bottom of the page shows the Adafruit logo and a footer with links for 'Get Help', 'Quick Guides', 'API Documentation', 'FAQ', 'Terms of Service', 'Privacy Policy', and 'Send Feedback'.

Name	Key	Created At
button	button	January 16, 2020
button	button	May 16, 2020

Feeds | [view all](#)

+ New Feed + New Group

Now under Feeds you should have a Feed called button, we will come back here once we are sending data to this feed.

Default

Feed Name	Key
<input type="checkbox"/> button	button

Step 5:

Let's load the provided sketch in this guide onto the Feather to start connecting our button data to an Adafruit IO Feed.

Open the Arduino IDE and/or load the sketch 'ixd_feather2web.ino',



```
ixd_feather2web | Arduino 1.8.13
// ixd_feather2web $ config.h
// Adafruit IO Digital Input Example
// Tutorial Link: https://learn.adafruit.com/adafruit-io-basics-digital-input
//
// 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 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,
// or ethernet clients.
#include "config.h"
//***** Example Starts Here *****
// digital pin 16
#define BUTTON_PIN 16
#define LED_PIN 13
// button state
bool current = false;
bool last = false;
// set up the 'digital' feed
// change "myFeed" to the feed name you have created in Adafruit IO
AdafruitIO_Feed *reName = io.feed("reName"); //New!
void setup() {
  // set button pin as an input, set led pin as output
  pinMode(LED_BUILTIN, OUTPUT);
  pinMode(BUTTON_PIN, INPUT);
  pinMode(LED_PIN, OUTPUT); //New!
```

There are a few places we need to specify the code. On the commented set up the 'digital' feed line we will change the word 'reName' to 'button'

```

//***** Example Starts Here *****

// digital pin 16
#define BUTTON_PIN 16
#define LED_PIN 13

// button state
bool current = false;
bool last = false;

// set up the 'digital' feed
// change "myFeed" to the feed name you have created in Adafruit IO
AdafruitIO_Feed *reName = io.feed("reName"); //New!

// set up the 'digital' feed
// change "myFeed" to the feed name you have created in Adafruit IO
AdafruitIO_Feed *button = io.feed("button"); //New!
```

At the bottom of the code you will find another mention of 'rename' needing to be replaced with 'button'.

```
//place your specific feed name
reName->save(current); //New!
//place your specific feed name
button->save(current); //New!
```

In the future when you are sending data from your Feather these are the three locations you need to change to the name of your feed.

Step 6:

There is a separate file in the folder called 'config.h', you can see it in the Arduino IDE tab. This file is where the information concerning your Wi-Fi password and Adafruit Key will be kept, do not share this information and always make sure you have blurred out or removed it when sharing files or documenting your process. This file will be loaded onto your Arduino but cannot be accessed by anyone unless you share it.

First right at the top of the code there is a location to add your Adafruit IO Key

```
// visit io.adafruit.com if you need to create an account,  
// or if you need your Adafruit IO key.  
#define IO_USERNAME "yourUsername"  
#define IO_KEY "yourKey"
```

Using the specific information from your Key fill in the areas named "yourUsername" and "yourKey".

YOUR ADAFRUIT IO 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 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

Active Key REGENERATE KEY

[Hide Code Samples](#)

Arduino

```
#define IO_USERNAME "yourUsername"  
#define IO_KEY "yourKey"
```

Linux Shell

```
export IO_USERNAME="yourUsername"  
export IO_KEY="yourKey"
```

Scripting

```
ADAFRUIT_IO_USERNAME = "yourUsername"  
ADAFRUIT_IO_KEY = "yourKey"
```

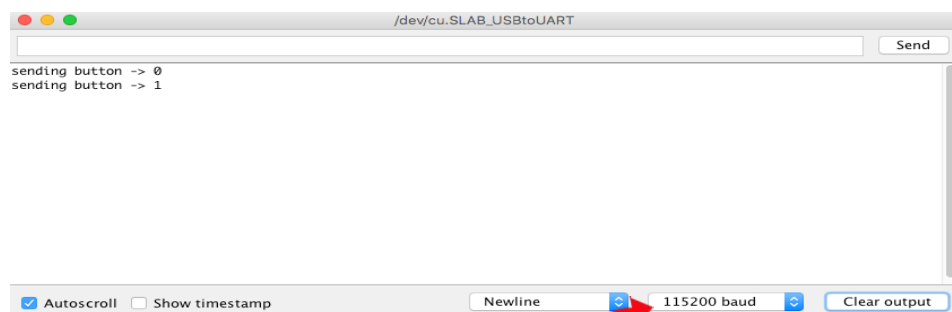
Next right below the board info comments there is a line of code we need to add your local wi-fi network name and password:

```
#define WIFI_SSID "yourSSID"  
#define WIFI_PASS "yourPass"
```

Put the name of your Wi-Fi Network here
Put your Wi-Fi Network password here

Save and upload the code to the Feather. Don't press any buttons. Once uploaded open the serial monitor. Press the button, on the board everything should happen the same, the led should light up.

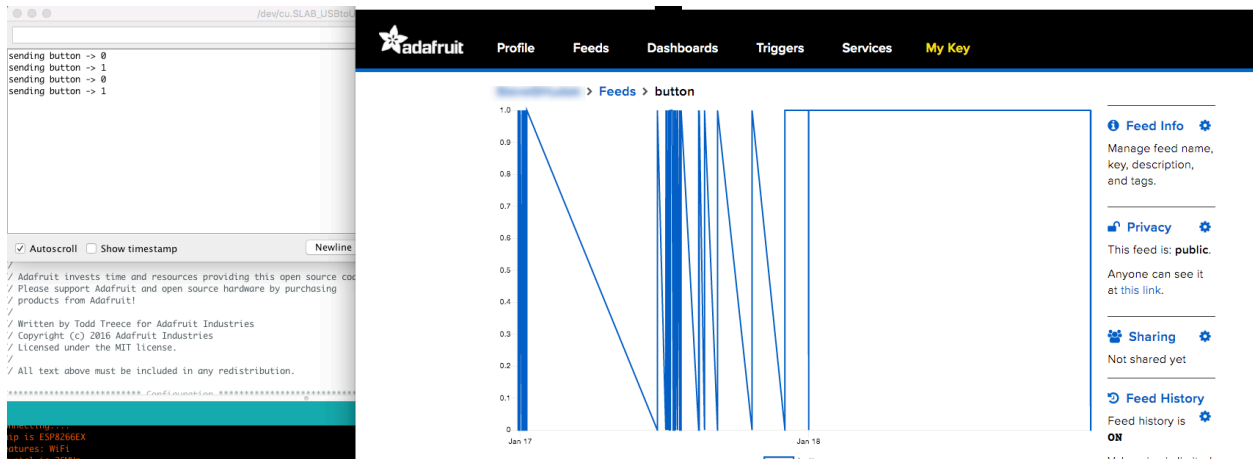
In the serial monitor should you see some text verifying the connection but when you press the button you should see this:



make sure you are set to 115200 baud.

Step 7:

Head back to the Adafruit IO site, click on Feeds and choose the button feed. Now when you press the button you should see a spike response as it registers the data being received.



These steps need to be completed successfully before continuing, once your feed is received we can do something web based with the data. Before we can access it outside of Adafruit IO we need to authorize the data feed to be public. This does mean anyone with the link can access your feed but that would only let people see the data point itself not the context.

First click on the Privacy tab on the right side of the Feeds page, then change it from Private to Public in the dropdown menu

Privacy

This feed is: **public.**

Anyone can see it at this link.

Update Visibility

Public means anyone in the world will be able to read data from your feed.

Private means your feed will only be visible to you.

Visibility

Pub...
Private
Public

Edit Feed

Name

button

Maximum length: 128 characters. Used: 6

Key

button

Changing the key will change API URLs and MQTT subscription topics. The only characters we permit are lower case english letters ("a" to "z"), numbers, and dash ("-").

See our guide to naming things in Adafruit IO for more information about how we handle the formatting of names and keys.

Current Endpoints

Web

API

MQTT

By Key

Description

Show detailed feed JSON record

Cancel

Save

Feed Info

Manage feed name, key, description, and tags.

Second we head to Feed Info and check our link for accessing the data from our now public feed:

We will use this API info directly in our P5.js sketch next.

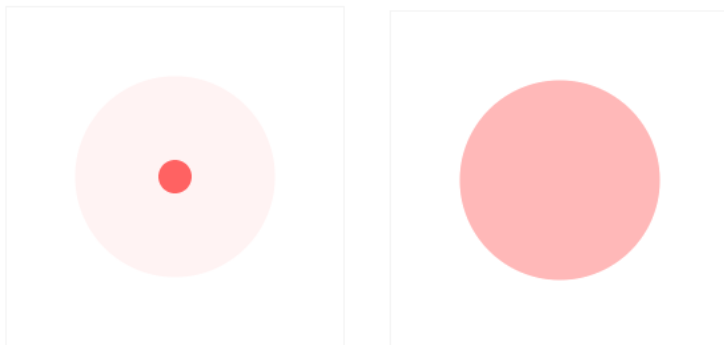
Step 8:

Lets look at the p5.js code we will use to connect to this data feed. Drag the 'AIO-feather2web' folder into Brackets:



Save and upload the HTML and the sketch to your Firebird server.

Once uploaded make your way to the URL and click the button on your circuit. You should see an small blinking ellipse that responds to your button by affecting the diameter of the ellipse.



This concludes the 8266 Feather Huzzah Guide 2, to move on to the opposite interaction of programming your Feather to respond to an interaction from a p5.js sketch please download and follow the 8266 Feather Huzzah Guide 3.