



Welcome to Adafruit IO

Created by Brent Rubell



Last updated on 2020-06-15 05:18:24 PM EDT

Overview

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?

Adafruit IO (<https://adafru.it/fsU>) is a platform designed (by us! (<https://adafru.it/Bo5>)) to *display*, *respond*, and *interact* with your project's data. We also keep your data **private** (data feeds are private by default) and *secure* ([we will never sell or give this data away to another company](#) (<https://adafru.it/BII>)) for you. It's the internet of things - for everyone!

Why We Built Adafruit IO

Here at Adafruit, [we sell all of these amazing components](#) (<https://adafru.it/dAR>), but we couldn't find a good way to interact with them over the Internet. There are certainly a lot of great services out there for datalogging, or communicating with your microcontroller over the web, but these services are either too complicated to get started, or they aren't particularly fun to use. So, we decided to experiment with our own system, and that's Adafruit IO.

What is Adafruit IO?

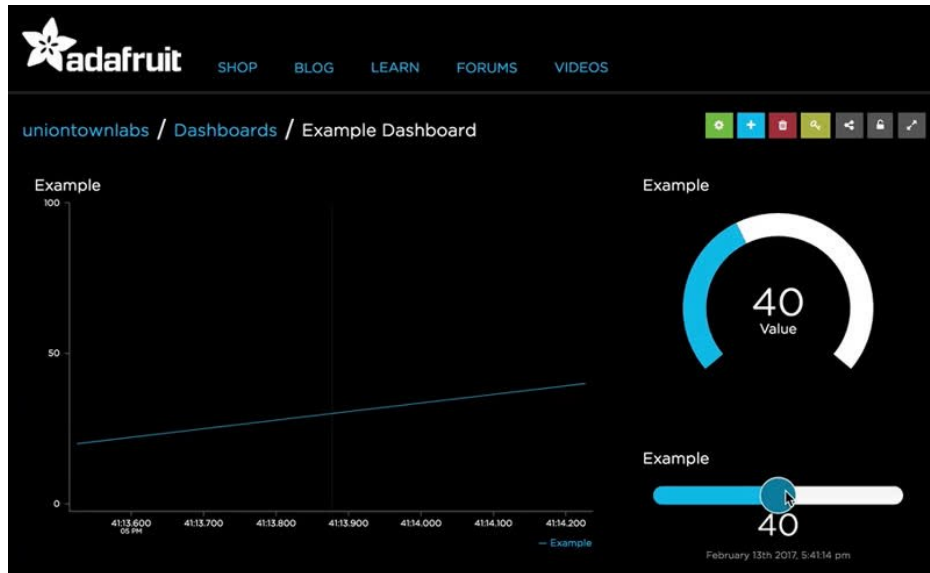


Adafruit.io is a *cloud service* - that just means we run it for you and you don't have to manage it. You can connect to it over the Internet. It's meant primarily for storing and then retrieving data but it can do a lot more than just that!

What can Adafruit IO do for me?

- Display your data in real-time, online
- Make your project internet-connected: Control motors, read sensor data, and more!
- Connect projects to web services like Twitter, RSS feeds, weather services, etc.
- Connect your project to other internet-enabled devices
- The best part? All of the above is do-able for **free** with Adafruit IO

Dashboards



Adafruit.io can handle and visualize multiple feeds of data. Want to display data from a [temperature-humidity sensor \(https://adafru.it/sb4\)](https://adafru.it/sb4) alongside data from an [air quality sensor \(https://adafru.it/BIK\)](https://adafru.it/BIK) and add a button to turn on the air-conditioner in your room?

No problem! [Dashboards \(https://adafru.it/f5m\)](https://adafru.it/f5m) are a feature integrated into Adafruit IO which allow you to chart, graph, gauge, log, and display your data. You can view your dashboards from anywhere in the world.

Triggers

Use triggers in Adafruit IO to control and react to your data. Configure triggers to email you when your system goes offline, react to a temperature sensor getting too hot, and publish a message to a new feed.

Integration with IFTTT and Zapier

Want to make your project react to an email, display trending tweets, or turn on the front lights when your pizza is on the way? We baked in integrations with [IFTTT \(https://adafru.it/Bvr\)](https://adafru.it/Bvr) and [Zapier \(https://adafru.it/BIL\)](https://adafru.it/BIL) to connect your project's sensors to hundreds of web services.

Getting Started with Adafruit IO

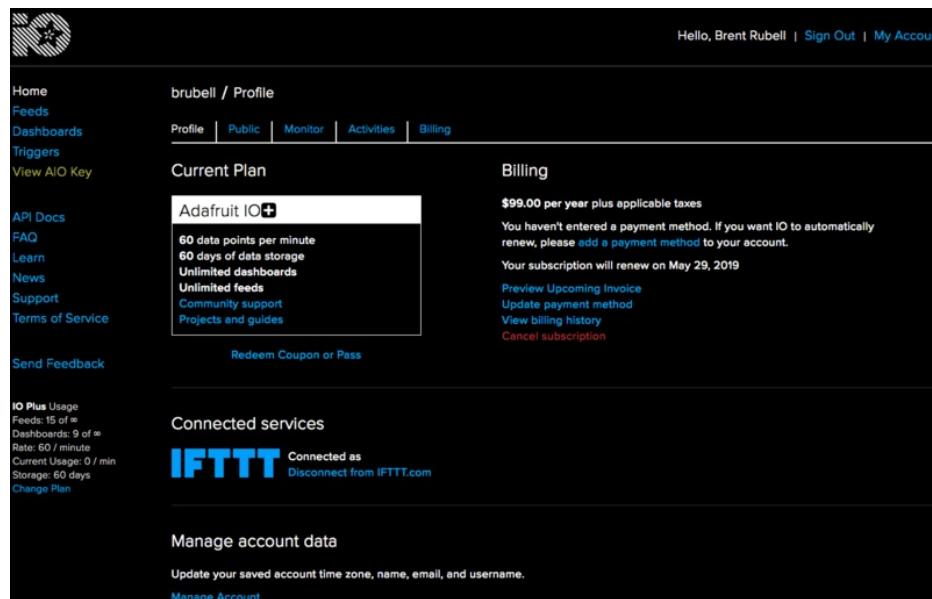
GOOD NEWS - we integrated Adafruit IO with your adafruit.com account (<https://adafru.it/dyy>) so you don't need to create yet-another-online-account!



You need an account because we want to make sure the data you upload is available to you and only you until you are ready to make it public

Head over to io.adafruit.com (<https://adafru.it/fsU>) and click **Sign In** to log into IO using your Adafruit account. It's free and fast to join.

You will be directed to your profile page showing your plan, connected services, and account data:



From here, you'll want to learn about two important features of Adafruit IO before proceeding further - Feeds and Dashboards

Feeds

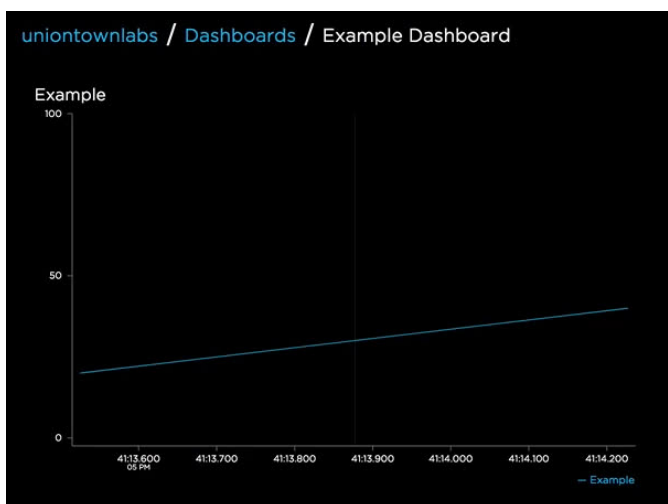


Feeds are the core of Adafruit IO. They hold both the data you uploaded **and** *meta-data* about the data your sensors push to Adafruit IO. For example, the date and time when it was uploaded. Or, the GPS coordinates where the data came from.

Want to share your data feed with the world, a group of people, or just a friend? Adafruit IO has integrated feed sharing with fine-tuned privacy controls to allow (and restrict) access to your data.

[Click here to learn more about Feeds in Adafruit IO Basics: Feeds](https://adafru.it/ioA) (<https://adafru.it/ioA>)

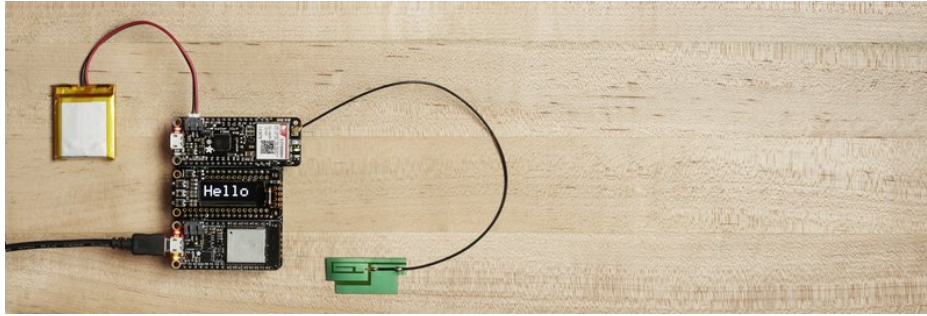
Dashboards



Dashboards are a feature integrated into Adafruit IO which allow you to chart, graph, gauge, log, and visualize your data. You can view your dashboards from anywhere in the world.

[Click here to learn more about Dashboards in Adafruit IO Basics: Dashboards](https://adafru.it/f5m) (<https://adafru.it/f5m>)

Arduino and Adafruit IO



Have an Arduino or Arduino-Compatible board which you want to use with Adafruit IO? Want to adapt your sketch to Adafruit IO? It's easy! We provide two libraries which work with lots of compatible hardware.

Need inspiration or project examples? We built sixteen examples sketches to get you up and running, along with six guides to follow-along to.

Compatible Hardware

The Adafruit IO Arduino Libraries can connect with Adafruit IO using Arduino. You can connect your projects using a connected-interface of your choosing: WiFi (ESP8266, ESP82, WINC1500, and WICED), Ethernet, or Cellular.

There's a ton of hardware using these interfaces available, but this guide is going to highlight boards Adafruit produces and sells:

WiFi

You know WiFi - you use it all-day every-day. It's the standard wireless protocol for connecting to the Internet. It's available not only at home and work but also in stores, cafes, trains, planes and in most major cities. The worldwide universality of WiFi makes it a common first choice for IoT. - [All the Internet of Things \(https://adafru.it/BIX\)](https://adafru.it/BIX)

The Adafruit IO Arduino and MQTT libraries are compatible with the following products:

WiFi - AirLift

First up is hardware with **AirLift** - boards which let you use the powerful **ESP32 as a WiFi co-processor**. **What's AirLift?** You probably have your favorite microcontroller (like the ATmega328 or ATSAM51), awesome peripherals and lots of libraries. But it doesn't have WiFi built in! So lets give that chip a best friend, the ESP32. This chip can handle all the heavy lifting of connecting to a WiFi network and transferring data from a site, even if its using the latest TLS/SSL encryption (it has root certificates pre-burned in). Having WiFi managed by a separate chip means your code is simpler, you don't have to cache socket data, or compile in & debug an SSL library.

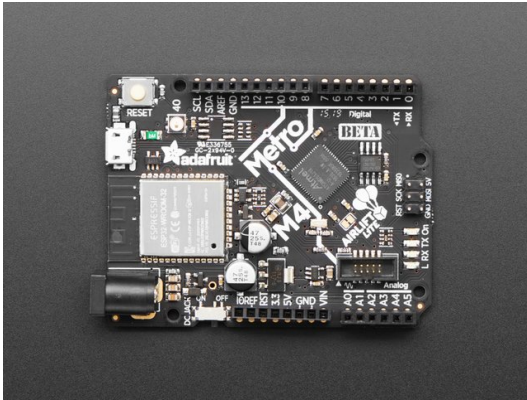
We sell standalone AirLift FeatherWings/Shields/Breakouts along with all-in-one AirLift boards such as the Metro M4 Express AirLift and the PyPortal:



Adafruit PyPortal - CircuitPython Powered Internet Display

OUT OF STOCK

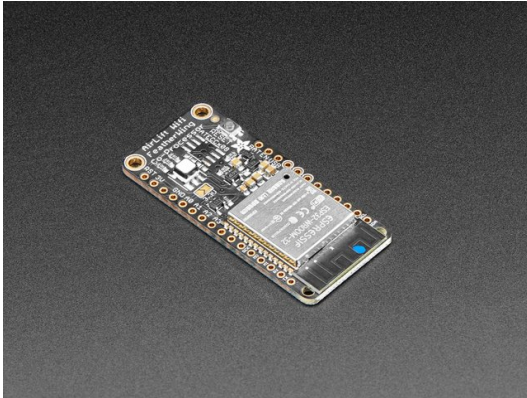
Out Of Stock



Adafruit Metro M4 Express AirLift (WiFi) - Lite

\$34.95
IN STOCK

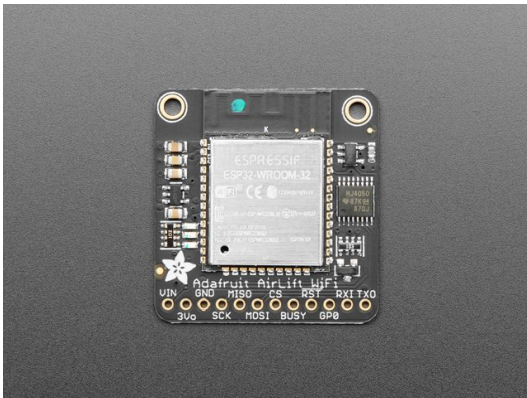
Add To Cart



Adafruit AirLift FeatherWing – ESP32 WiFi Co-Processor

OUT OF STOCK

Out Of Stock

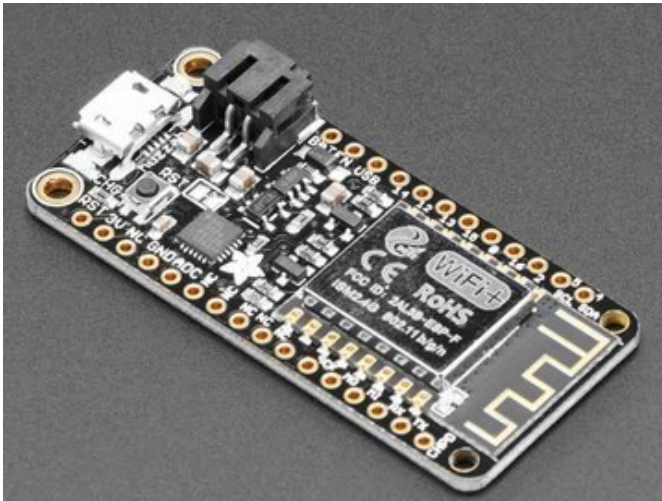


Adafruit AirLift – ESP32 WiFi Co-Processor Breakout Board

OUT OF STOCK

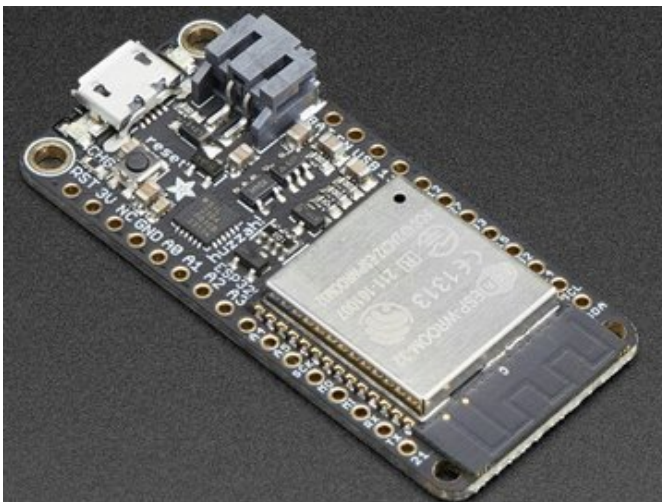
Out Of Stock

WiFi - ESP-based



We offer two **ESP8266**-based platforms:

- [Adafruit Feather HUZZAH with ESP8266](https://adafru.it/n6A) (<https://adafru.it/n6A>) is our take on an 'all-in-one' ESP8266 WiFi development board with built in USB and battery charging. Its an ESP8266 WiFi module with all the extras you need, ready to rock!
- [Adafruit HUZZAH ESP8266 Breakout](https://adafru.it/f9X) (<https://adafru.it/f9X>) is what we designed to make working with this chip super easy and a lot of fun. We took a certified module with an onboard antenna, and plenty of pins, and soldered it onto our designed breakout PCBs. This version is tougher to use than the Feather because it needs a separate cable, so its not recommended for beginners

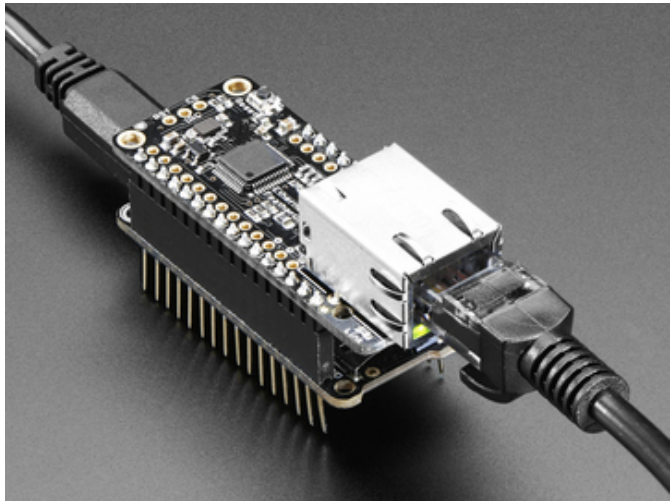


The [Adafruit HUZZAH32](https://adafru.it/wcN) is our **ESP32-based Feather** (<https://adafru.it/wcN>), made with the official WROOM32 module. We packed everything you love about Feathers: built in USB-to-Serial converter, automatic bootloader reset, Lithium Ion/Polymer charger, and just about all of the GPIOs brought out so you can use it with any of our Feather Wings.

Ethernet

Good ol' Ethernet. This 'ancient' protocol has withstood the test of time. You almost certainly use Ethernet at home and at work. Ethernet has a standardized connector, the venerable RJ-45. When you need something to "just work," often a wire will do that. (Just because it's "IoT" doesn't mean it's wireless, just that many Things happen to be wireless.) - [All the Internet of Things](https://adafru.it/BIY) (<https://adafru.it/BIY>)

Our Adafruit IO libraries can also talk to Adafruit IO using ethernet!



Ethernet FeatherWing (<https://adafru.it/wiB>) will let you add quick and easy wired Internet. Just plug in a standard [ethernet cable](https://adafru.it/Bmw) (<https://adafru.it/Bmw>), and run the Ethernet2 library for cross-platform networking. **Works with all/any of our Feather boards!**

Cellular Connectivity

Cell phones can go just about anywhere - there are towers across every country, in almost any place there are people. And cellular towers can be very powerful. When coupled with a high power transceiver on your device, range can easily hit miles away. - [All the Internet of Things](https://adafru.it/BIZ) (<https://adafru.it/BIZ>)



The **Adafruit FONA** (<https://adafru.it/BIW>) is a all-in-one cellular phone module which lets you connect your project using Adafruit IO to a cellular network. This is great for projects which either move around (like a car-interface) or are in remote places (like a data-logger!)

Libraries

Before you get started using Adafruit IO with your Arduino, you'll need to select a library. We provide and support both the Adafruit IO and Adafruit MQTT libraries listed below, but try starting with the Adafruit IO Arduino Library below:

Adafruit IO Arduino Library



The library supports the following network platforms and hardware:

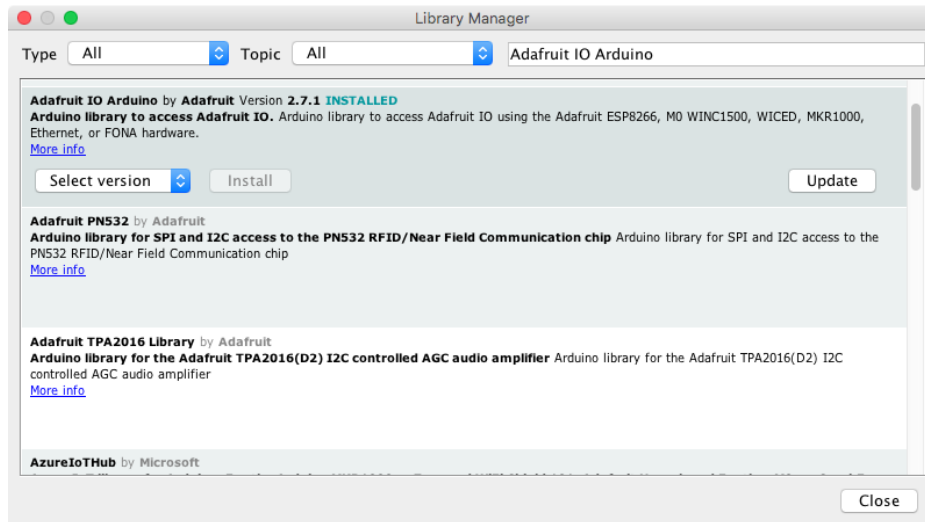
- [Adafruit AirLift](https://adafru.it/F5X) (<https://adafru.it/F5X>)
- [ESP8266](https://adafru.it/fpd) (<https://adafru.it/fpd>)
- [ESP32](https://adafru.it/fpd) (<https://adafru.it/fpd>)
- [ATWINC1500](https://adafru.it/fpd) (<https://adafru.it/fpd>)
- [WICED](https://adafru.it/fpd) (<https://adafru.it/fpd>)
- [Ethernet](https://adafru.it/fpd) (<https://adafru.it/fpd>)
- [FONA \(Cellular\)](https://adafru.it/fpd) (<https://adafru.it/fpd>)

Installation



If you're using an Adafruit AirLift board - there is some extra configuration required to set up Adafruit IO Arduino: <https://learn.adafruit.com/adafruit-io-basics-airlift>

You can install the library through the Arduino Library Manager (click: *Sketch -> Include Library -> Manage Libraries...*)



Alternatively, you can [download the Adafruit IO Arduino Library from GitHub \(https://adafru.it/fpd\)](https://adafru.it/fpd) and [install it manually \(https://adafru.it/fCN\)](https://adafru.it/fCN).

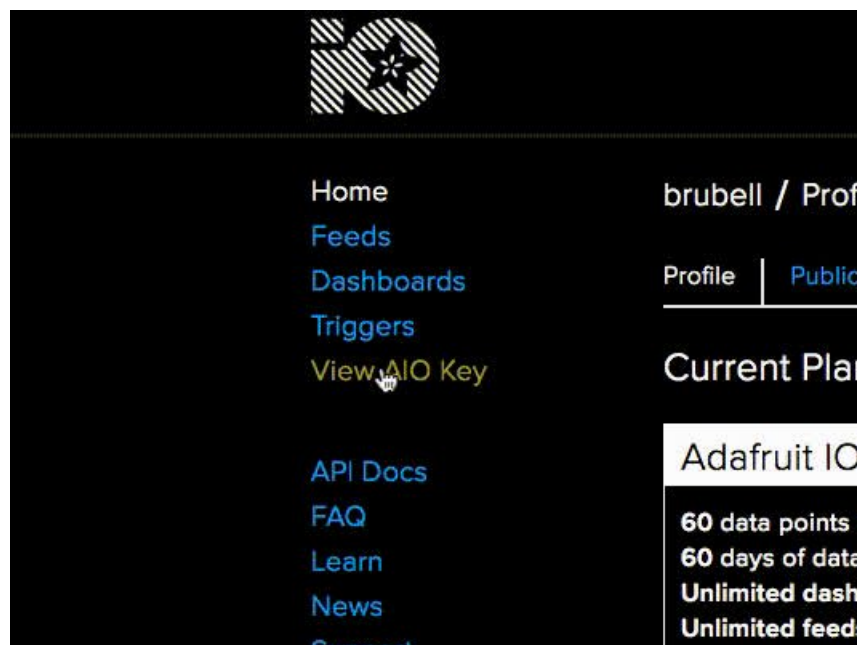
Usage

The included examples sketches will walk you through all of the features of the library. They can be used on all platforms, but they default to WiFi. Most of the sketches have [companion projects on the Adafruit Learning system. \(https://adafru.it/BmC\)](https://adafru.it/BmC)

To change between platforms, you will need to change two lines of code in the `config.h` tab of the example. It is recommended that you start with one of the Adafruit WiFi feathers before moving on to cellular or ethernet.

For all examples, you will need to set `IO_USERNAME` and `IO_KEY` in the `config.h` tab:

To find your `IO_USERNAME`, [navigate to your profile on Adafruit IO \(https://adafru.it/BmD\)](https://adafru.it/BmD) and click **View AIO Key**. Copy the **Username** field (ctrl+c or command+c)



Then, in the `config.h` tab, replace the `"your_username"` text with your the username from your profile:



```
adafruitio_06_digital_in config.h
/***** Adafruit IO Config *****/

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

To find your `IO_Key`, navigate to your profile, click **View AIO Key**, and copy the **ACTIVE KEY** field to your clipboard (ctrl+c or command+c).

In `config.h`, replace the `IO_KEY` with the IO Key copied to your clipboard.



```
adafruitio_06_digital_in config.h
/***** Adafruit IO Config *****/

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

Using the Adafruit IO Arduino Library with WiFi

If you are using the included examples, you do not need to change anything for the Adafruit WiFi Feathers. All WiFi based Feathers (ESP8266, M0 WiFi, WICED) will work with the examples out of the box once you add your WiFi SSID and Password to the `config.h` file.

In the `config.h` tab, replace `"your_ssid"` with your WiFi's SSID and `"your_pass"` with your WiFi's password:

```
adafruitio_06_digital_in - config.h | Arduino 1.8.5
Verify

adafruitio_06_digital_in config.h

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

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

/***** WIFI *****/

// the AdafruitIO_WiFi client will work with the following boards:
// - HUZZAH ESP8266 Breakout -> https://www.adafruit.com/products/2471
// - Feather HUZZAH ESP8266 -> https://www.adafruit.com/products/2821
// - Feather M0 WiFi -> https://www.adafruit.com/products/3010
// - Feather WICED -> https://www.adafruit.com/products/3056

#define WIFI_SSID "your_ssid"
#define WIFI_PASS "your_pass"

// 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);
```

Using the Adafruit IO Arduino Library with Ethernet

For Ethernet, you will only need to change from the default WiFi constructor to the Ethernet specific constructor in the `config.h` tab. The rest of the sketch remains the same.

You will need to comment out these WiFi lines in `config.h`:

```
#include "AdafruitIO_WiFi.h"
AdafruitIO_WiFi io(IO_USERNAME, IO_KEY, WIFI_SSID, WIFI_PASS);
```

and uncomment the Ethernet lines in `config.h`:

```
#include "AdafruitIO_Ethernet.h"
AdafruitIO_Ethernet io(IO_USERNAME, IO_KEY);
```

Using the Adafruit IO Arduino Library with Cellular (32u4 FONA)

For FONA, you will only need to change from the default WiFi constructor to the FONA specific constructor in the `config.h` tab. The rest of the sketch remains the same.

You will need to comment out these WiFi lines in `config.h`:

```
#include "AdafruitIO_WiFi.h"
AdafruitIO_WiFi io(IO_USERNAME, IO_KEY, WIFI_SSID, WIFI_PASS);
```

and uncomment the FONA lines in `config.h`:

```
#include "AdafruitIO_FONA.h"
AdafruitIO_FONA io(IO_USERNAME, IO_KEY);
```

If your carrier requires APN info, you can set it by adding a call to `io.setAPN()` after `io.connect()` in the `setup()` function of the sketch.

```
void setup() {
  // start the serial connection
  Serial.begin(115200);
  // connect to io.adafruit.com
  io.connect();
  io.setAPN(F("your_apn"), F("your_apn_user"), F("your_apn_pass"));
}
```

Adafruit MQTT Arduino Library

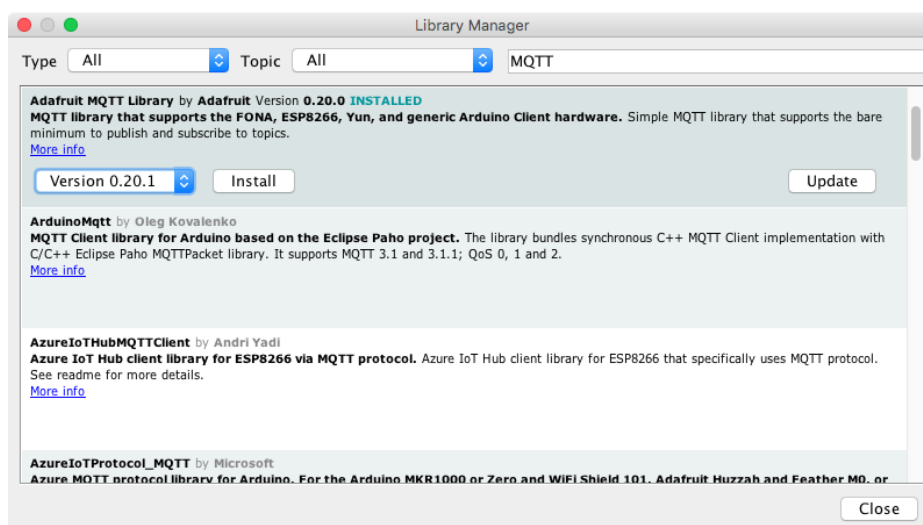
We also have a library to provide support for accessing Adafruit IO using MQTT. This is a general-purpose MQTT library for Arduino that's built to use as few resources as possible so that it can work with platforms like the Arduino Uno. Unfortunately platforms like the Trinket 3.3V or 5V (based on the ATtiny85) have too little program memory to use the library--stick with a Metro 328p or better!

The MQTT library supports the following network platforms and hardware:

- [Adafruit CC3000](https://adafru.it/BmG) (<https://adafru.it/BmG>)
- [Adafruit FONA](https://adafru.it/BIW) (<https://adafru.it/BIW>)
- [ESP8266](https://adafru.it/BmH) (<https://adafru.it/BmH>)
- Generic Arduino Client Interface (incl. ethernet shield and similar networking hardware)

Installation

You can install the library through the Arduino Library Manager (click: *Sketch -> Include Library -> Manage Libraries...*)



Alternatively, you can download the MQTT library from [GitHub](https://adafru.it/fp7) (<https://adafru.it/fp7>) and manually install it (<https://adafru.it/fCN>).

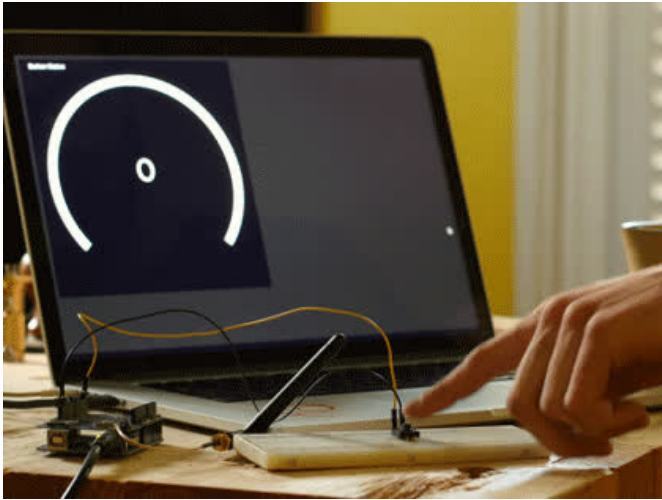
If you're using an Adafruit FONA, you'll need to install an additional two libraries for the MQTT Library to work properly:

- [Adafruit SleepyDog \(https://adafru.it/fp8\)](https://adafru.it/fp8), watchdog library used by FONA code for reliability.
- [Adafruit FONA \(https://adafru.it/dDC\)](https://adafru.it/dDC), required for the FONA hardware.

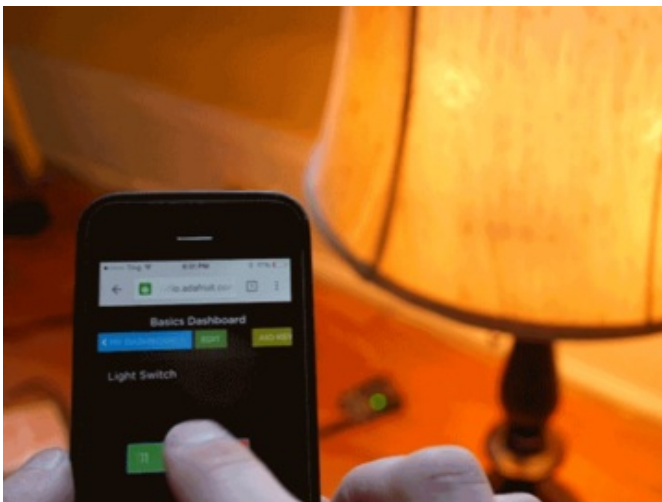
Projects

Adafruit IO Basics

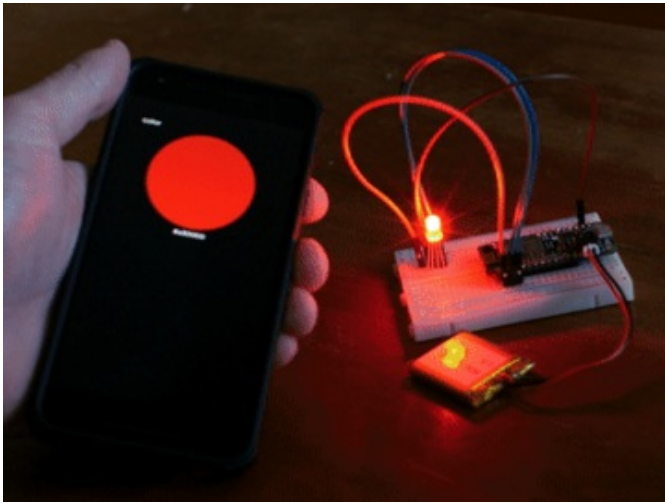
Todd Treece (<https://adafru.it/ubi>) has put together seven guides using Adafruit IO for input, temperature, humidity, servo control, and so much more:



Digital Input (<https://adafru.it/wDS>): Wire a circuit to send momentary button press data to your Adafruit IO Dashboard



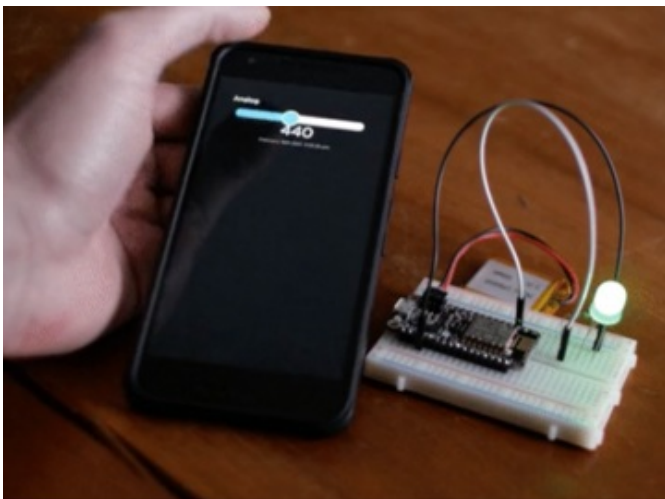
Digital Output: (<https://adafru.it/iRe>) Turn on a LED or a lamp from anywhere in the world!



Color: (<https://adafru.it/uFC>) Send data from Adafruit IO to control a RGB LED.



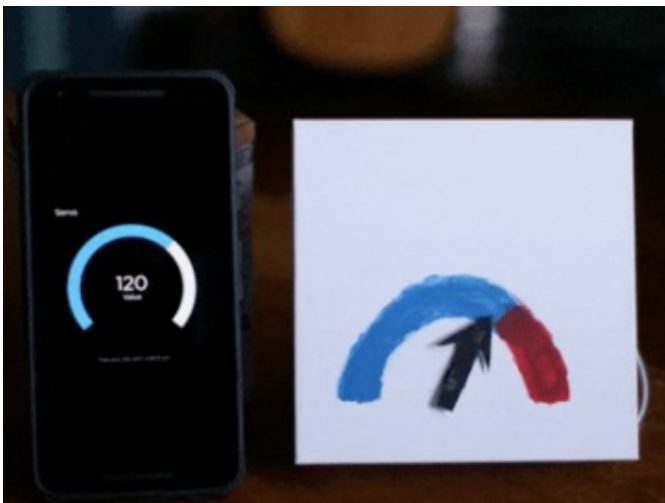
Temperature and Humidity: (<https://adafru.it/uFD>) Send humidity and temperature values from a DHT22 sensor to Adafruit IO



Analog Output: (<https://adafru.it/uFE>) LED and other analog hardware control from Adafruit IO.



Analog Input: (<https://adafru.it/uFF>) Send light sensor data in real-time and view it using Adafruit IO.



Servo Control: (<https://adafru.it/uFG>) Move physical objects...without wires!

Adafruit IO on the Adafruit Learning System

The Adafruit Learning System is also a great place to find learning guides which answer the age-old maker question of *"what do I build next?"*.

Here are some of our favorite guides using Adafruit IO:



Gmailbox (<https://adafru.it/CrU>)

Build a physical mailbox for your emails using Adafruit IO and IFTTT



Adafruit IO Environmental Monitor for Feather or Raspberry Pi (<https://adafru.it/CrV>)

Build an internet-enabled environmental monitor to see what's in the air you breathe using Arduino



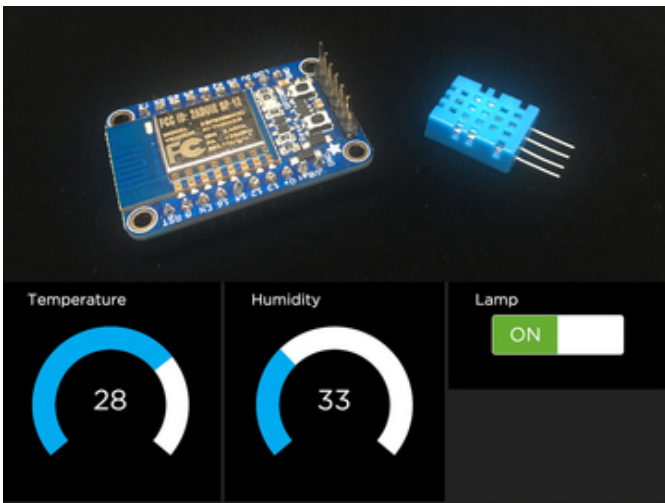
Adafruit IO Home: Lights and Temperature (<https://adafru.it/CrW>)

Learn all about home automation by building a cardboard smart-house, controlling its lights, and monitoring temperature!



Using IFTTT with Adafruit IO to Make an IoT Door Detector (<https://adafru.it/Bo3>)

Know who is comin' and goin'



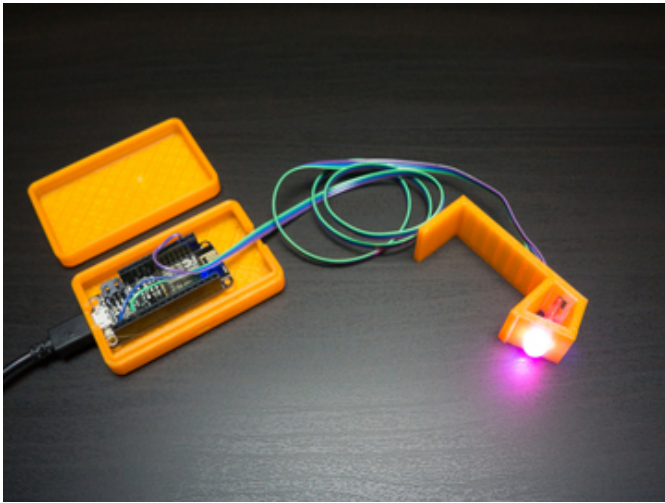
Home Automation in the Cloud with the ESP8266 & Adafruit IO (<https://adafru.it/Bo2>)

Build a simple home automation system & control it from anywhere!



Cloud Cam: Internet-Connected Security Camera (<https://adafru.it/BmA>)

Build a camera that detects motion and sends images to cloud services like Dropbox & Adafruit IO!



Smart Toilet Light (<https://adafru.it/Bo4>)

Make your toilet bowl glow with data from Adafruit IO!

Check out more guides using Adafruit IO here! (<https://adafru.it/BI->)

Adafruit IO Projects on the Blog!

The Adafruit Blog is also a good resource for finding your next project. [Check out the Adafruit IO blog posts here!](#) (<https://adafru.it/Bm0>)

CircuitPython and Adafruit IO

Ready to bring your CircuitPython project *online*? Quickly connect it to Adafruit IO using the [CircuitPython Adafruit IO](https://adafru.it/EFZ) (<https://adafru.it/EFZ>) library!

Using CircuitPython with Adafruit IO lets you **rapidly update your code without having to re-compile**. You can also store your WiFi and Adafruit IO API keys *directly* on the device. This means that there's no editing code and re-uploading whenever you move your project to another network - just update a file and you're set.

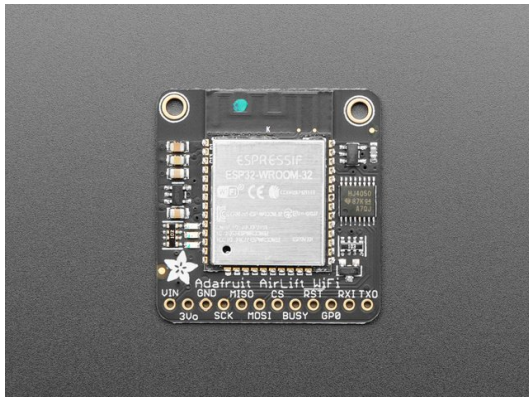
The CircuitPython Adafruit IO module can communicate with Adafruit IO using either the HTTP API or the MQTT API.

Compatible Hardware



Adafruit IO for CircuitPython requires an M4 or better microcontroller! The M0 will not work.

You can connect a CircuitPython board with a M4 microcontroller to Adafruit IO with an AirLift breakout.

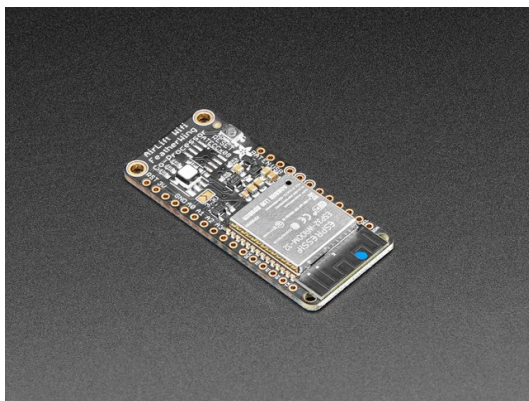


[Adafruit AirLift – ESP32 WiFi Co-Processor Breakout Board](#)

OUT OF STOCK

Out Of Stock

If you're using a [Feather-form-factor CircuitPython device](https://adafru.it/Eg4) (<https://adafru.it/Eg4>) (with a M4 microcontroller), the AirLift FeatherWing can plug into a doubler. But, you can also pick up a set of stacking headers to stack above/below your Feather.



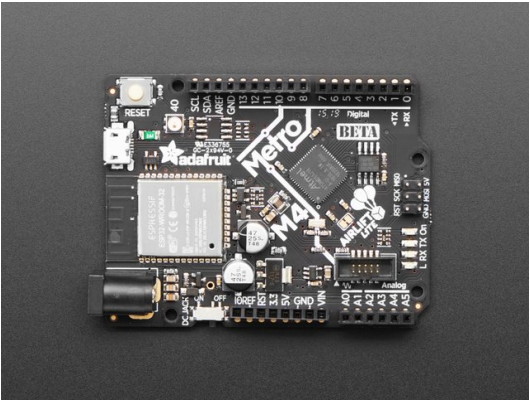
[Adafruit AirLift FeatherWing – ESP32 WiFi Co-Processor](#)

OUT OF STOCK

Out Of Stock

If you do not want to attach extra hardware and want a powerful all-in-one CircuitPython device, pick up a board such

as the Metro M4 AirLift or PyPortal.



Adafruit Metro M4 Express AirLift (WiFi) - Lite

\$34.95
IN STOCK

Add To Cart




Adafruit PyPortal - CircuitPython Powered Internet Display

OUT OF STOCK

Out Of Stock

For a full list of Adafruit IO compatible CircuitPython hardware, visit [the CircuitPython.org downloads page \(https://adafru.it/Em8\)](https://adafru.it/Em8), click the sort icon, and *select WiFi* from the options listed.



Downloads

Wi-Fi

Manufacturers

☐ Adafruit

☐ Nordic

☐ Arduino

Features

☐ Display

☐ Speaker

☒ Wi-Fi

Get Started with CircuitPython and Adafruit IO

Have a CircuitPython project you'd like to connect to Adafruit IO? We've written an [Adafruit IO CircuitPython module](https://adafru.it/EFZ) (<https://adafru.it/EFZ>) to provide simple interaction with the Adafruit IO MQTT or HTTP API.

<https://adafru.it/EYj>

<https://adafru.it/EYj>

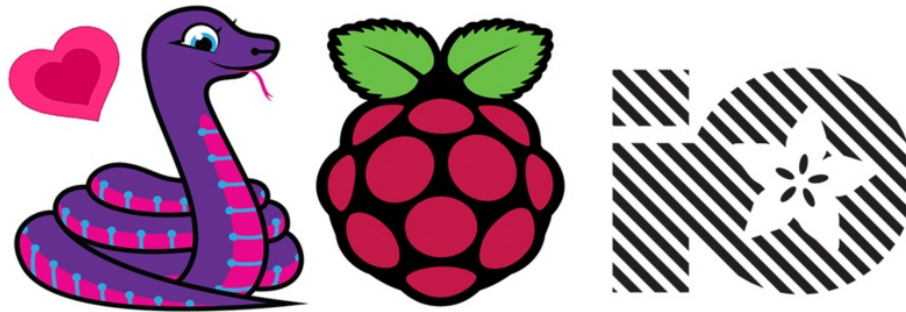
Python and Adafruit IO



Want to use Adafruit IO with your computer, a Raspberry Pi, Beaglebone, or your favorite single-board linux computer?

The [Adafruit IO Python library](https://adafru.it/DOJ) (<https://adafru.it/DOJ>) provides two clients for accessing Adafruit IO (MQTT and HTTP) and *lots* of examples. It's compatible with any system running CPython3 and also compatible with Single-Board computers like the Raspberry Pi or BeagleBone

Compatible Hardware



Write your *code in Python* and connect it to the internet with Adafruit IO!

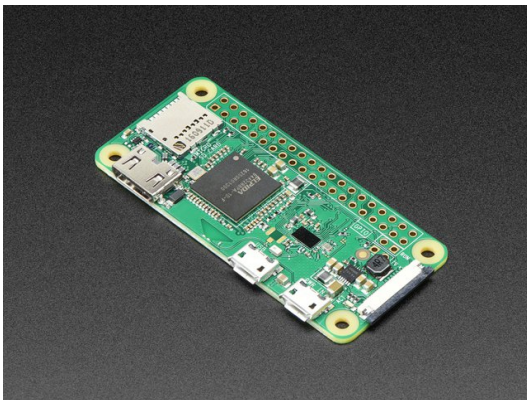
What does this mean for Adafruit IO?

You can control your Raspberry Pi by writing Python code, but what about connecting that to the internet?

We've updated the Adafruit IO library with usage examples for [Python and Adafruit IO \(https://adafru.it/CrX\)](https://adafru.it/CrX).

Suggested Hardware

The Raspberry Pi Zero W offers all the benefits of the full-sized Pi and built-in WiFi. This board is *perfect* for Adafruit IO projects which require a Feather-like form factor.

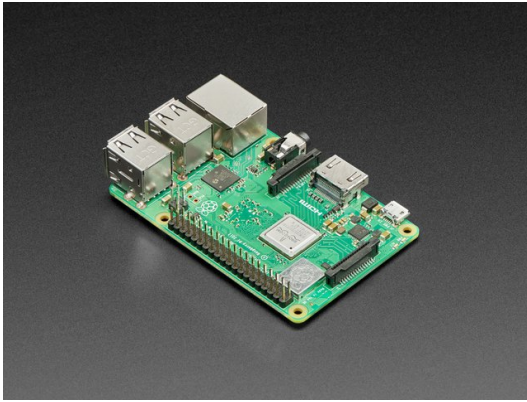


Raspberry Pi Zero W

\$10.00
IN STOCK

Add To Cart

The Raspberry Pi 3 Model B+ boasts a 64-bit quad core processor, 5GHz wireless LAN and faster (300mbps) Ethernet. If you'd like to have a beefier hardware platform to use with Adafruit IO and possibly some more intensive libraries (such as OpenCV) - this is the ideal hardware platform for your project..



Raspberry Pi 3 - Model B+ - 1.4GHz Cortex-A53 with 1GB RAM

\$35.00
IN STOCK

Add To Cart

Client Library

Installing Adafruit IO Python

If you have [PIP \(https://adafru.it/ncM\)](https://adafru.it/ncM) installed (typically with `apt-get install python-pip` on a Debian/Ubuntu-based system), run:

```
sudo pip3 install adafruit-io
```

This will automatically install the Adafruit IO Python client code for your Python scripts to use. There are usage examples [within the examples folder of the GitHub repository \(https://adafru.it/fpg\)](https://adafru.it/fpg) for use with MQTT, API, and Basic IO functionality.

Note: This library requires Python3. It is incompatible with Python2.

Library Usage

We provide usage examples [within the examples folder of the GitHub repository \(https://adafru.it/fpg\)](https://adafru.it/fpg) for the REST API and MQTT.

If you want to download all the examples to your computer, clone the repository in a easy-to-access location.

To do this, type this in your terminal to **navigate to your desktop**:

```
cd Desktop
```

Download the latest version of the repository by entering the following command into your terminal:

```
git clone https://github.com/adafruit/Adafruit_IO_Python.git (https://adafru.it/EAa)
```

Then, **navigate to the examples folder** by entering the following into your terminal:

```
cd Adafruit_IO_Python/examples/
```

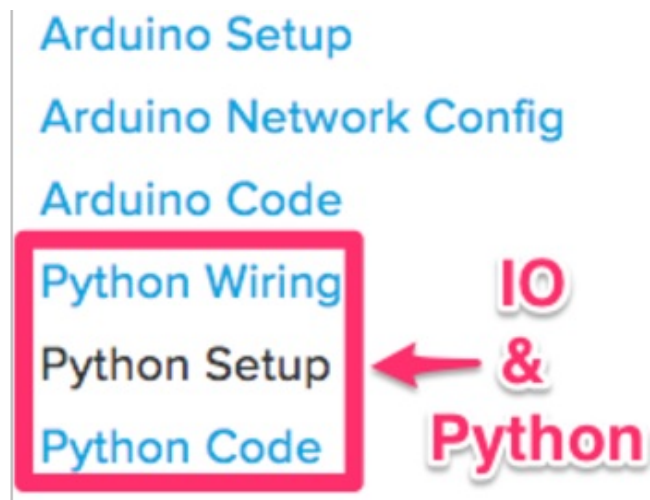
Note: For all examples in this folder, you'll need to set the `ADAFRUIT_IO_KEY` and `ADAFRUIT_IO_USERNAME` before running the program.

Library Documentation

Documentation for all methods and classes in the Adafruit IO Python library [can be found on the ReadTheDocs page for this project \(https://adafru.it/BQt\)](https://adafru.it/BQt).

Projects

Adafruit IO Basics for CircuitPython



We've updated our [Adafruit IO Basics Series \(https://adafru.it/iDX\)](https://adafru.it/iDX) to add compatibility for using Adafruit IO and Python. These are the best examples if you're looking to get started building projects.

Projects and Inspiration

You can find projects which use Adafruit IO Python in the [Adafruit Learning System's IoT Category \(https://adafru.it/BI-\)](https://adafru.it/BI-).

Other Client Libraries

We also support a few other languages with our client libraries:

Node.js

Writing a Node.js web application? We provide [a Node.js client, local server, and TLS tunnel for io.adafruit.com on our GitHub \(https://adafru.it/Bm2\)](#). This library can use *both* the REST API and the MQTT API to access feeds and data on Adafruit IO.

The [library readme on GitHub \(https://adafru.it/elj\)](#) describes how to install and use the library.

Ruby

If you have a Ruby program you'd like to interface with Adafruit IO, [we provide a Ruby Client on our Github \(https://adafru.it/enB\)](#). This library wraps the REST API to access feeds and data on Adafruit IO. The [library readme shown on GitHub \(https://adafru.it/enB\)](#) describes how to install and use the library.

Be sure to also see the [examples \(https://adafru.it/fpf\)](#) included for use with this library.

Go

You can also talk to your io.adafruit.com account using our [go client library \(https://adafru.it/Bvt\)](#). The library requires go version 1.2 and running tests requires the testify library, which can be installed with:

```
$ go get github.com/stretchr/testify
```

Be sure to also see our [usage notes \(https://adafru.it/Bvt\)](#) along with the [godoc page \(https://adafru.it/Bvu\)](#) for full package documentation.

IO Plus



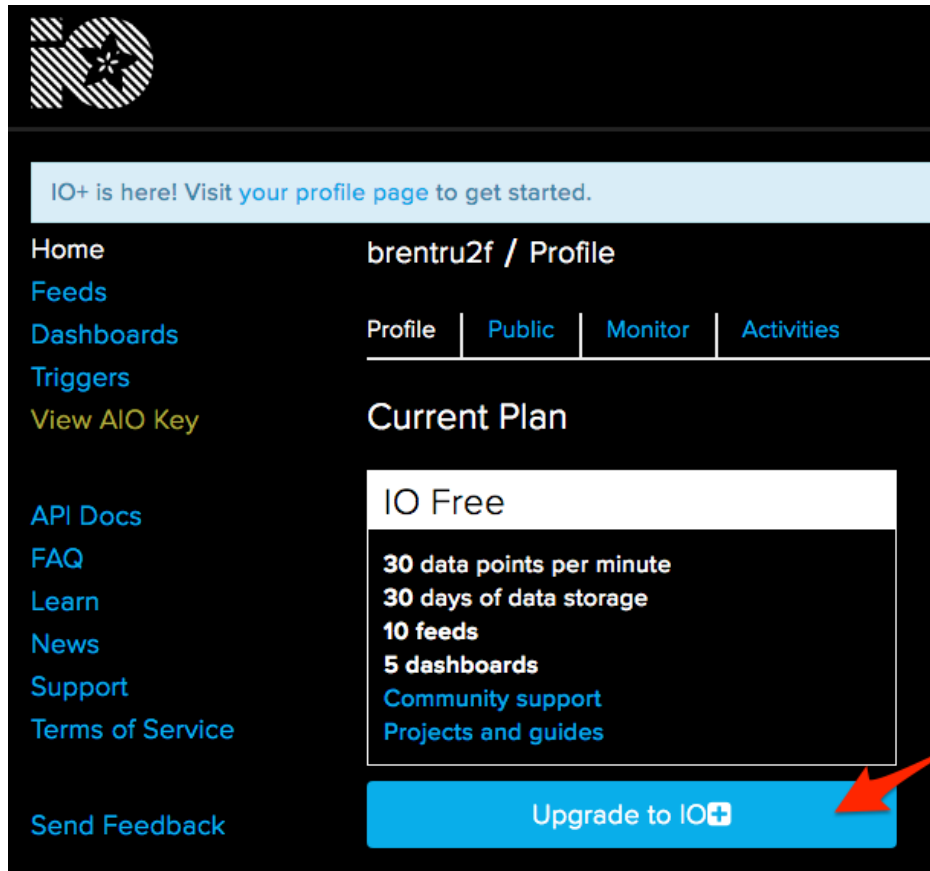
Your Adafruit IO experience is *even better* with IO+. The 'plus' stands for MORE STUFF! More feeds, dashboards, storage, speed. IO+ unlocks more data, more storage, and more feeds than our free service.

Upgrading your account to IO+ gives you:

- 60 data points per minute
- 60 days of data storage
- **Unlimited** dashboards
- **Unlimited** feeds

Subscribing to Adafruit.IO

Ready to subscribe to Adafruit IO+? [Visit your profile page \(https://adafru.it/BmD\)](https://adafru.it/BmD) and click the **Upgrade to IO+** button:



Then, fill out your payment information, and your Adafruit IO account will be upgraded to IO Plus!

IO Plus FAQ

Why should I subscribe to IO Plus? How is it billed?

For one, uncomplicated price you get **unlimited feeds and dashboards**, a **higher data input rate**, and **longer data storage**.

IO+ subscriptions are billed immediately when you subscribe and then monthly or yearly, repeating each month or year on the same day you signed up until you cancel.

□ Payments and Credit Card Processing

We use [Stripe](#) for payment processing and subscription management and are currently able to accept every major credit card including: Visa, Master Card, American Express, Discover, JCB, and Diners Club. All charges are denominated in USD (United States dollars) but international customers are welcome!

□ What if I want to cancel my IO+ Subscription?

It's easy and painless - when you cancel your Adafruit IO+ subscription, your access to IO+ will continue until the end of your current billing cycle. At then end of your subscription period you account will be restored to IO Free account limits. Right now that's 10 feeds, 5 dashboards, 30 days of data storage, and a 30 data points per minute data rate. Data that has already been stored will be preserved, and all extra feeds and dashboards will be archived.

📄 I'm having trouble with billing, or payment processing. Can you help?

Absolutely! If you have a question we haven't answered here or on [the Adafruit IO forums](#), or if you have a question about your subscription specifically, you can use [the Adafruit Contact page](#) to get in touch with our support team directly.

□ Is there a way to purchase IO+ for a yearly fee instead of billing monthly?

Sure, we offer a [one-year subscription pass](#) in the Adafruit store.

□ Can I gift an IO Plus subscription to my friend, coworker, family-member, or a fellow maker?

You can [purchase the subscription pass](#) for yourself or as a gift, and send it to a family or friend. Code is redeemable within <https://io.adafruit.com> once you are signed in to your account. The code is not locked to any specific account until it is redeemed.

Securing Your IO Account

Your IO Account includes a secret API key. This key is private and should not be shared with anyone.

Common pitfalls of the key is it may be accidentally shared by posted code on GitHub or social media.

If anyone gains access to your key, they will have access to your Adafruit IO account.

Key Regeneration

Your key can be regenerated within IO by clicking the yellow *AIO Key* text in the header of the page.



Once you click that link, a dialog box will show which will allow you to click *REGENERATE AIO KEY*.

This process is not reversible.

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.

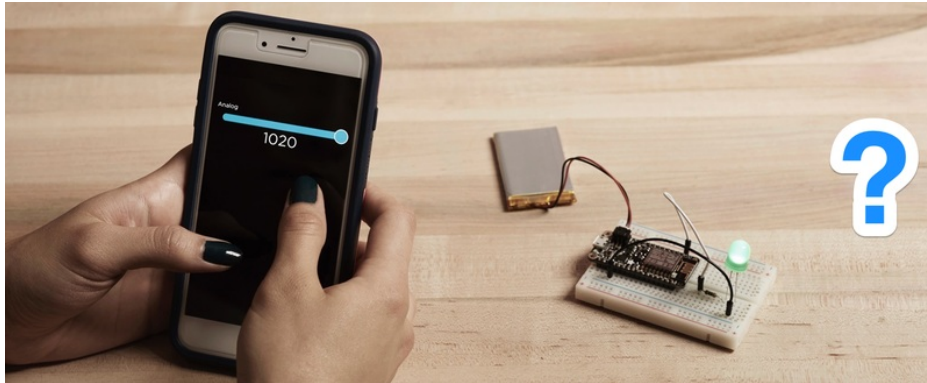


Username	<input type="text" value="io_example"/>	
Active Key	<input type="text" value="aio_tkyo82A9FFbyOYPmpeBp4VQCwguu"/>	
		<input type="button" value="REGENERATE AIO KEY"/>

For further reading on IoT security, we have an [entire guide that explains various types of attacks and how to protect yourself \(https://adafru.it/laB\)](https://adafru.it/laB).

Troubleshooting your Adafruit IO Project

The Problem with IoT Project Troubleshooting



Much like a sandwich, Adafruit IO projects have a lot of layers where things can go wrong. As a result, it becomes much harder to troubleshoot these types of projects.

There's the **hardware** (like a Feather ESP8266) layer which brings issues such as "why isn't my motor turning". Your wiring may be an issue. The code could be incorrect. It could even be a networking issue, which brings us to the next layer...

Then, there's **networking** which connects your internet-controlled project to the internet. Problems could arise here resulting to DNS configuration, port forwarding problems, or your router simply not talking to The Internet™.

There's also The Internet™, and that brings a lot of problems too. Finally there's Adafruit IO, a web service that we built and support.

...and there could be problems with any layer of this delicious sandwich.

How to Help Out ("I have a problem that's not listed here, what now?")

If your project is not working properly, the Adafruit community is here to help you out. Post up in the [Adafruit IO Forums \(https://adafru.it/plc\)](https://adafru.it/plc) or chat with Adafruit staff and community members in real-time on the [adafruit-io channel on the Adafruit Discord server \(https://adafru.it/BmL\)](https://adafru.it/BmL).

While we help out and see repeated issues, we'll add them to this page to help others.

Common Issues

My Serial Monitor prints "." endlessly after the "Connecting to Adafruit IO" message

Your board is not connecting to Adafruit IO, but why? Let's find out:

First, check in `config.h` that you have the correct `IO_USERNAME`, `IO_KEY`, `WIFI_SSID`, and `WIFI_PASS` are set correctly.

Next, we're going to modify the while loop which waits for an IO connection in your sketch. Change the line in the

status check loop from `Serial.println(.);` to `Serial.println(io.statusText());`

```
// wait for a connection
while(io.status() < AIO_CONNECTED) {
  Serial.println(io.statusText());
  delay(500);
}
```

Verify and re-upload the sketch. If you're receiving a **Network disconnected** error message, the board is not able to talk to the internet. Re-check your hardware, connections, and router settings.

If it's still not showing **Adafruit IO connected**, check the [IO status on the Adafruit Status page](#) to make sure the service is online.

❏ My data isn't displaying, is Adafruit IO's {service/MQTT/API} down?

Possibly - check [IO status on the Adafruit Status page](#).

□ Is my data being sent properly? Am I sending too much data?

There's a [monitor page built-into Adafruit IO](#) which provides a live view of incoming data and error messages. Keep this page open while you send data to your Adafruit IO devices to monitor data and errors.

▢ My dashboard is reporting a temperature reading in Fahrenheit/Celsius but I need it in the opposite unit

The dashboard displays information from feeds. Chances are, the sensor you are using is outputting a temperature in Fahrenheit/Celsius but you need to convert it (in your code) to the opposite unit. Here's some handy pseudo-code to help out:

From Fahrenheit to Celsius:

$$\text{degreesFahrenheit} = \text{degreesCelsius} \times 9/5 + 32$$

From Celsius to Fahrenheit:

$$\text{degreesCelsius} = (\text{degreesFahrenheit} - 32) \times 5/9$$

□ I'm using an Feather M0 WiFi (ATWINC1500) and it's not connecting to Adafruit IO.

If you're using a Feather M0 WiFi and receiving a *Disconnected from Adafruit IO* message in your serial monitor, you'll want to check two things:

1. Make sure that you have the correct Adafruit IO Keys in your sketch, and that the router/network configuration is correct.
2. The SSL certificates on your device for Adafruit IO need to be up-to-date. It's hard to check if they're up-to-date from the device, so we suggest running through the process of adding a new SSL certificate through the Arduino IDE. [You can find the guide for that here.](#)

Board-Specific Issues

ESP8266

❏ Error: 'AdafruitIO_WiFi' does not name a type

From Boards Manager, downgrade from ESP8266 v2.5.0-beta2 to **stable 2.4.2**.

REST API Documentation

[REST API Documentation \(https://adafru.it/uff\)](https://adafru.it/uff)

MQTT API Documentation

[MQTT API Documentation \(https://adafru.it/EY7\)](https://adafru.it/EY7)

IO FAQ

□ Can I build my own Client Library for Adafruit IO?

Absolutely - the same API that drives our user interface is available to you. [We provide documentation so you can build a library in your favorite language to talk to IO](#)

□ My data isn't displaying, is Adafruit IO's {service/MQTT/API} down?

Possibly - you can check [IO status on the Adafruit Status page](#).

□ Is my data being sent properly? Am I sending too much data?

There's a [monitor page built-into Adafruit IO](#) which provides a live view of incoming data and error messages. Keep this page open while you send data to your Adafruit IO devices to monitor data and errors.

□ What about the data that my project generates? Where does it go? What happens to it? Are you going to sell it?

The data your store with IO is yours to manage and control. You can download it all anytime and we will never sell or give it away to another company. We feel strongly enough about this that we put it in [our IoT Bill of Rights](#).

□ I have an Alexa/Echo Dot/AI Voice Assistant. How do I interface it with Adafruit IO?

Currently, the easiest way of doing this is by chaining events through a web-service such as [IFTTT](#) which has integrated [Alexa Skills](#) and [Google Assistant Skills](#).

📄 I don't see my question listed.

If you have any questions or issues with Adafruit IO, post up in the [Adafruit IO Forums](#) or chat with Adafruit staff and community members in real-time on the [adafruit-io channel on the Adafruit Discord server](#).

