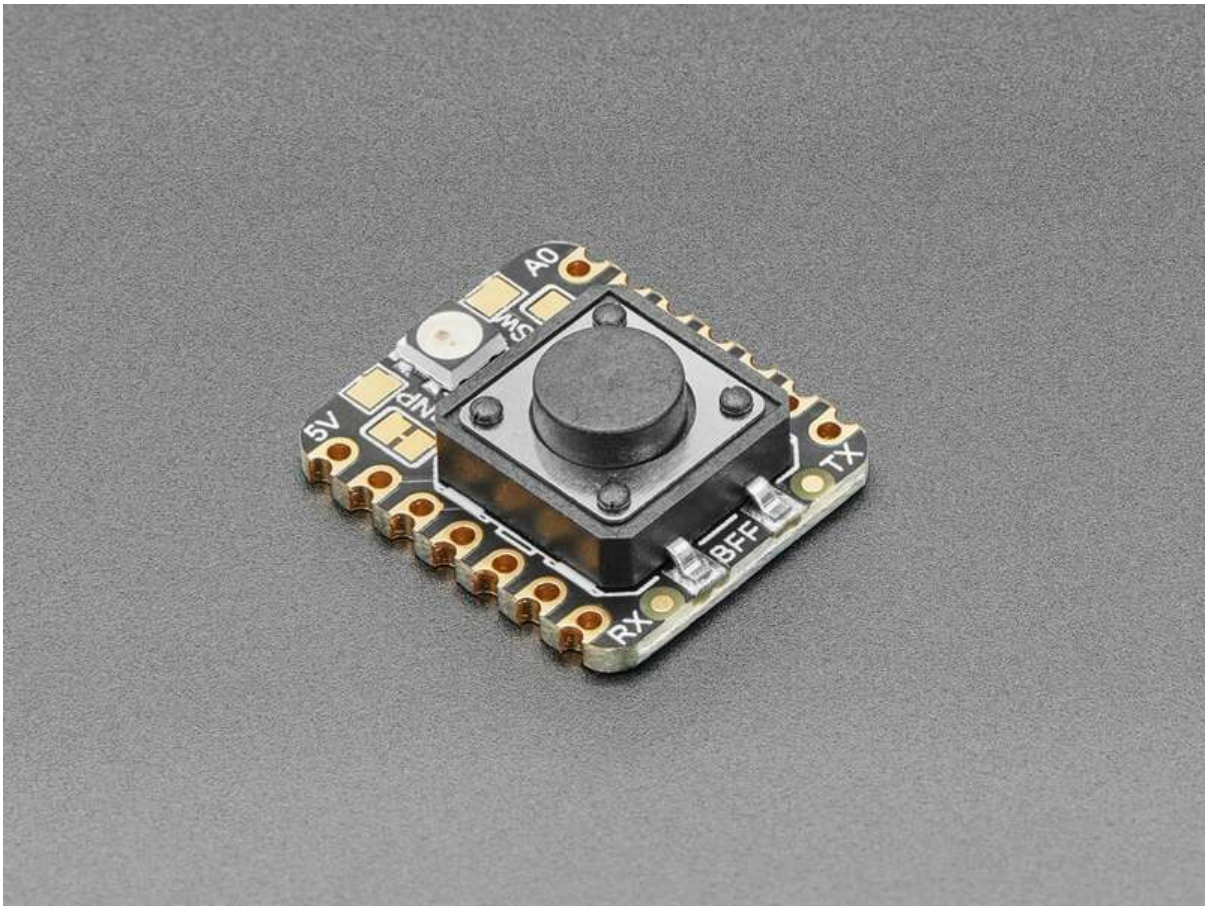
Adafruit IoT Button with NeoPixel BFF

Created by Liz Clark

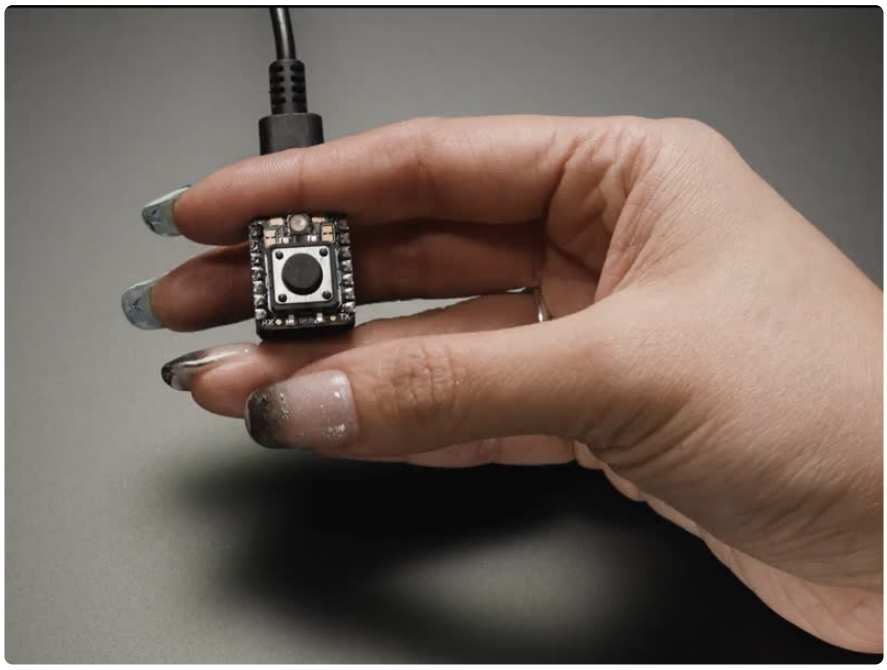


https://learn.adafruit.com/adafruit-iot-button-with-neopixel-bff

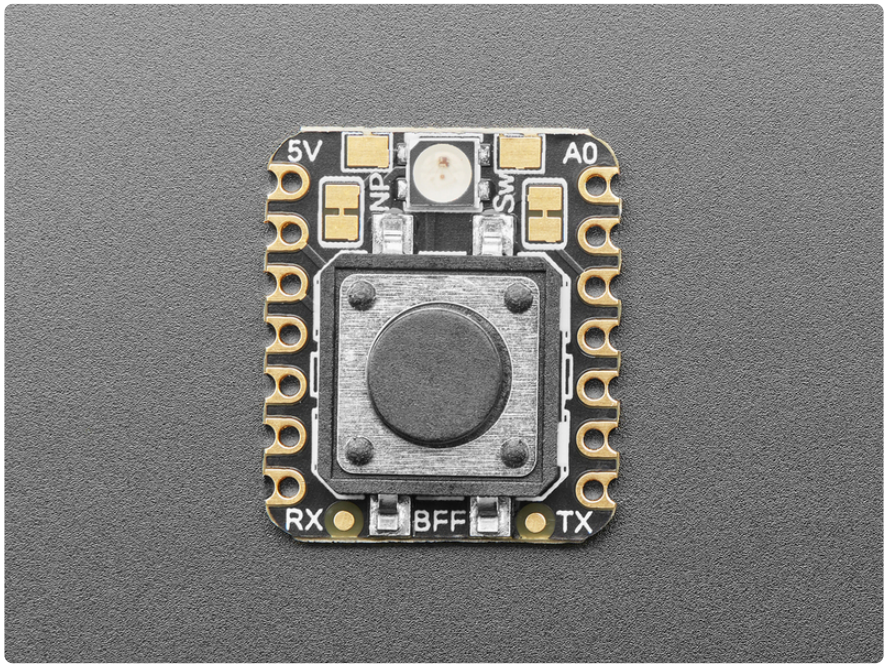
Last updated on 2024-06-03 03:45:22 PM EDT

Table of Contents

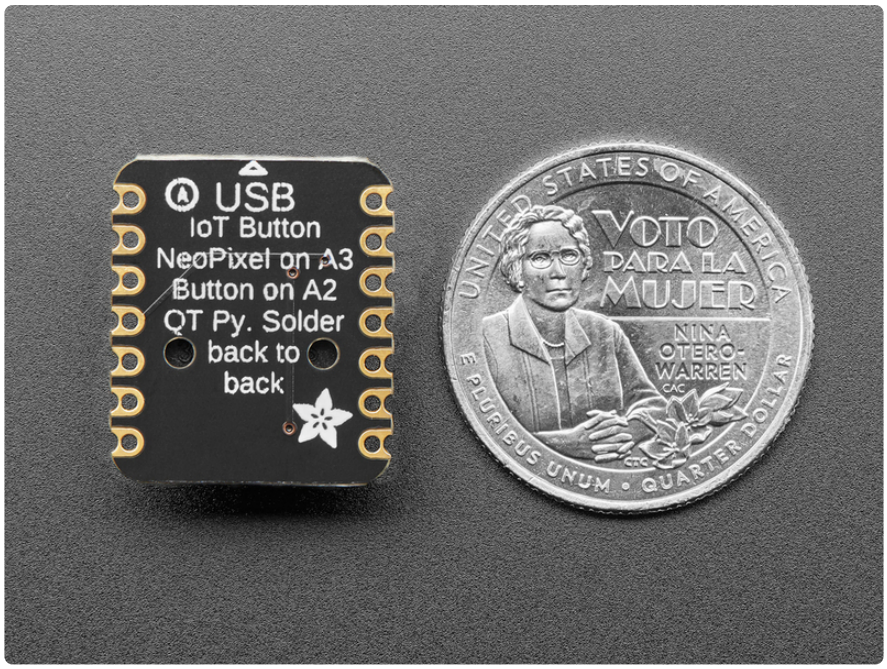
Overview



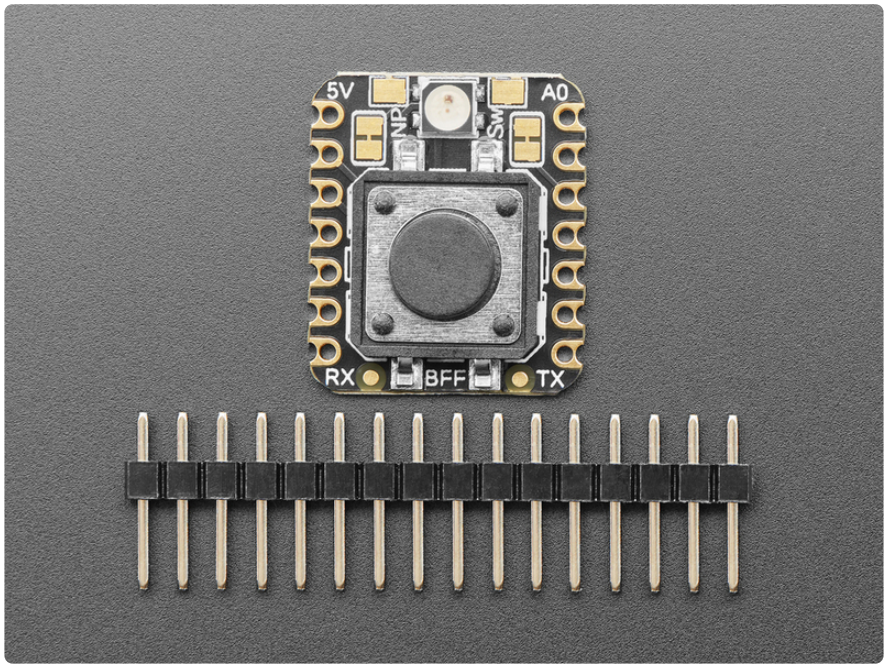
Our QT Py boards are a great way to make very small microcontroller projects that pack a ton of power - and now we have a way for you to quickly add a chunky 12mm tactile button with a NeoPixel. It's an excellent way to create simple 'IoT Button' type projects with basic interactivity.



We call this the **Adafruit IoT Button with NeoPixel BFF** - a "Best Friend Forever". When you were a kid you may have learned about the "buddy" system, well this product is kinda like that! A board that will watch your QT Py's back and give it more capabilities.



This PCB is designed to fit onto the back of any QT Py or Xiao board, it can be soldered into place or use pin and socket headers to make it removable. Onboard is a 12mm tactile button with a nice large actuator, above is a 3.5mm RGB NeoPixel.



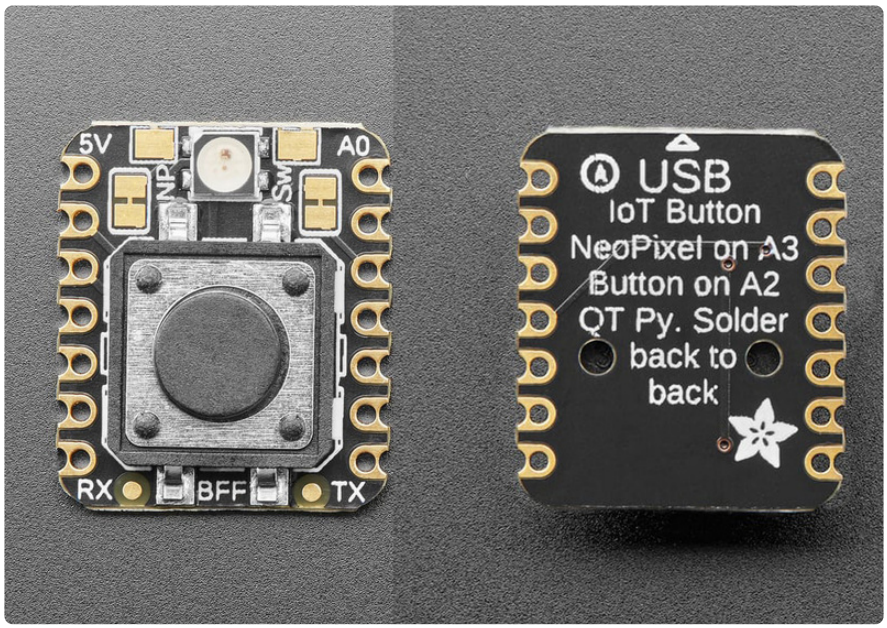
We include some header that you can solder to your QT Py. You can also pick up an Itsy Bitsy short female header kit to make it removable but compact (http://adafru.it/4174), you'll just need to trim down the headers to 7 pins long.

- Comes as an assembled and tested PCB
- For any QT Py or Xiao boards

- Tactile switch momentarily connects **A2** to ground when pressed - can cut/re-wire an onboard jumper to change pins if desired. Don't forget to enable an internal pull-up resistor on A2 or whatever pin you use.,
- Single NeoPixel on GPIO **A3**- can cut/re-wire an onboard jumper to change pins if desired.

QT Py is not included.

Pinouts



- The default button pin is **A2**
- The default NeoPixel pin is **A3**

NeoPixel Jumper

- **NP** - This jumper is located on the front of the board to the left of the NeoPixel LED and is labeled **NP**. If you cut this jumper, it disconnects the NeoPixel data input pin from pin **A3**.

NeoPixel Pad

- **NP** - This solder pad is located at the top of the board between the 5V silk text and the NeoPixel LED. It is labeled **NP**. This pad is attached to the NeoPixel data input pin and can be used to wire the NeoPixel to a different pin if the NeoPixel jumper is cut to disconnect it from pin **A3**.

Tactile Switch Jumper

- **Sw** - This jumper is located on the front of the board to the right of the NeoPixel LED and is labeled **Sw**. If you cut this jumper, it disconnects the button output from pin **A2**.

Tactile Switch Pad

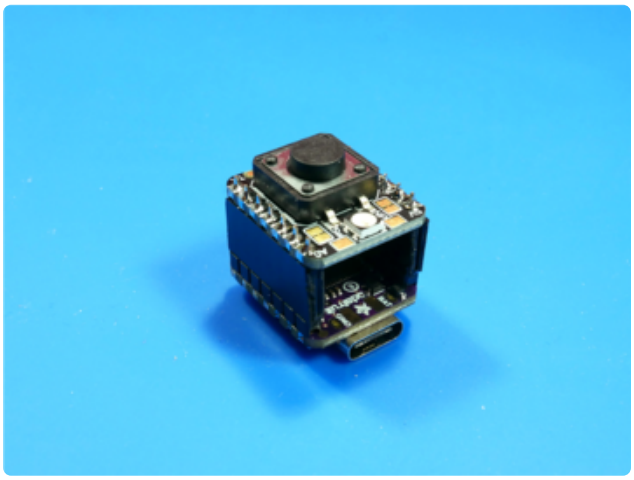
- **Sw** - This solder pad is located at the top of the board between the **A0** silk text and the NeoPixel LED. It is labeled **Sw**. This pad is attached to the button output pin and can be used to wire the button to a different pin if the tactile switch jumper is cut to disconnect it from pin **A2**.

CircuitPython

It's easy to use the **IoT Button with NeoPixel BFF** with CircuitPython and the Adafruit_CircuitPython_NeoPixel (https://adafru.it/yew) module. This module allows you to easily write Python code that lets you control NeoPixels. A second example will use the Adafruit_CircuitPython_AdafruitIO (https://adafru.it/Ean) module to let you write Python code to communicate with Adafruit IO, Adafruit's IoT platform.

CircuitPython Microcontroller Wiring

Plug an IoT Button with NeoPixel BFF into your QT Py or Xiao form factor board exactly as shown below. Here's an example of connecting a QT Py ESP32-S2 to the BFF.



Connect the QT Py ESP32-S2 with pin headers into the IoT Button with NeoPixel BFF with socket headers. They should be plugged in with the backs of the boards facing each other.

For more information on soldering socket headers, check out this Learn Guide (https://adafru.it/18gf).

How to Solder Headers Learn Guide

https://adafru.it/18gf

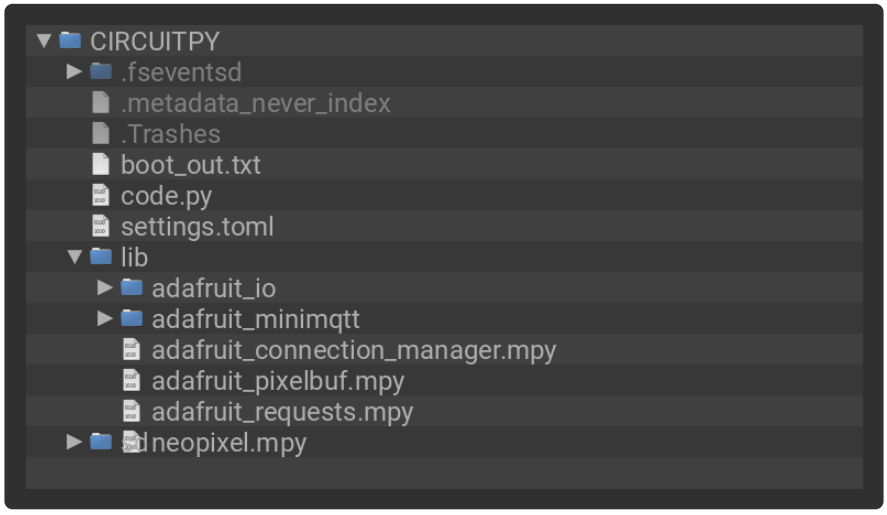
CircuitPython Usage

To use with CircuitPython, you need to first install the NeoPixel library, and its dependencies, into the **lib** folder on your **CIRCUITPY** drive. Then you need to update **code.py** with the example script.

Thankfully, we can do this in one go. In the example below, click the **Download Project Bundle** button below to download the necessary libraries and the **code.py** file in a zip file. Extract the contents of the zip file, and copy the **entire lib folder** and the **code.py** file to your **CIRCUITPY** drive.

Your **CIRCUITPY/lib** folder should contain the following folders and files:

- **/adafruit_io**
- **/adafruit_minimqtt**
- **adafruit_pixelbuf.mpy**
- **adafruit_requests.mpy**
- **neopixel.mpy**

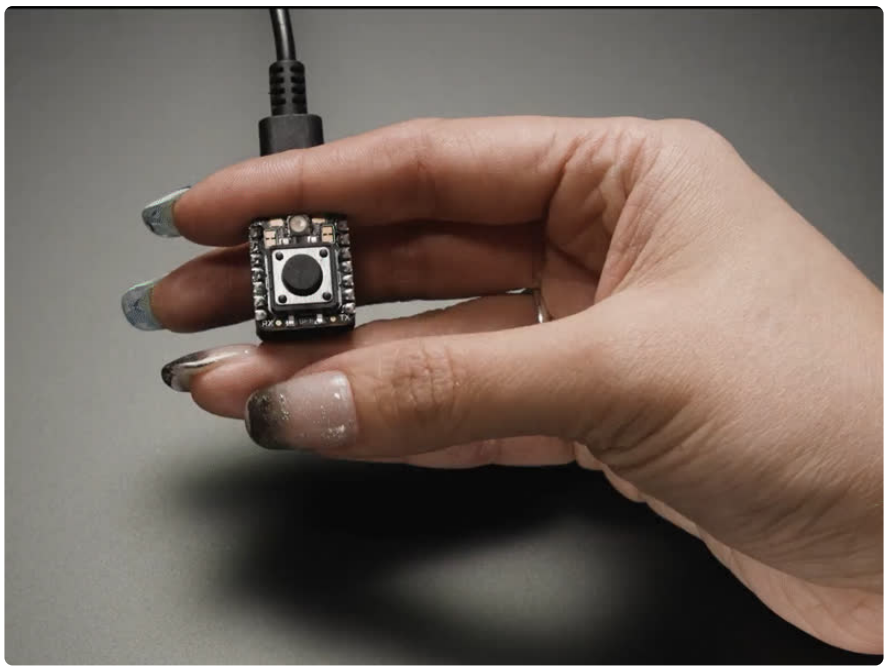


Simple Example

```
# SPDX-FileCopyrightText: 2023 Liz Clark for Adafruit Industries
#
# SPDX-License-Identifier: MIT

"""Basic IoT Button with NeoPixel BFF Example"""
import time
import board
from digitalio import DigitalInOut, Direction, Pull
from rainbowio import colorwheel
```

```
import neopixel

# setup onboard NeoPixel
pixel_pin = board.A3
num_pixels = 1

pixels = neopixel.NeoPixel(pixel_pin, num_pixels, brightness=0.3, auto_write=False)

# setup onboard button
switch = DigitalInOut(board.A2)
switch.direction = Direction.INPUT
switch.pull = Pull.UP

# rainbow cycle function
def rainbow_cycle(wait):
    for j in range(255):
        for i in range(num_pixels):
            rc_index = (i * 256 // num_pixels) + j
            pixels[i] = colorwheel(rc_index & 255)
        pixels.show()
        time.sleep(wait)

while True:
    # run rainbow cycle animation
    rainbow_cycle(0)

    # if the button is not pressed..
    if switch.value:
        # neopixel brightness is zero and appears to be "off"
        pixels.brightness = 0
    # if the button is pressed..
    else:
        # neopixel brightness is 0.3 and rainbow animation is visible
        pixels.brightness = 0.3
```

Once everything is saved to the **CIRCUITPY** drive, you can press and hold the button on the BFF to make the rainbow animation show on the NeoPixel. If you release the button, the NeoPixel will stop showing the animation.



Adafruit IO Example

To run this example for Adafruit IO, be sure to first review the Welcome to Adafruit IO Learn Guide (https://adafru.it/BRB) and read how to setup your **settings.toml** file (https://adafru.it/18f9) to store your WiFi and Adafruit IO credentials on your **CIRCUITPY** drive.

Welcome to Adafruit IO Learn Guide

https://adafru.it/BRB

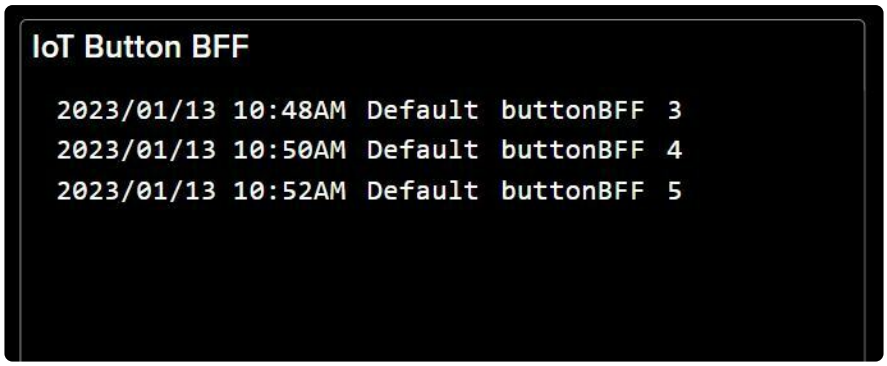
Create Your settings.toml File

https://adafru.it/18f9

```
# SPDX-FileCopyrightText: 2023 Liz Clark for Adafruit Industries
# SPDX-License-Identifier: MIT

"""Simple Adafruit IO Example for IoT Button with NeoPixel BFF"""
import os
import time
import ssl
import wifi
import socketpool
import microcontroller
import board
from digitalio import DigitalInOut, Direction, Pull
import neopixel
import adafruit_requests
from adafruit_io.adafruit_io import IO_HTTP, AdafruitIO_RequestError

# setup onboard button
switch = DigitalInOut(board.A2)
switch.direction = Direction.INPUT
switch.pull = Pull.UP

# setup onboard NeoPixel
pixel_pin = board.A3
num_pixels = 1

pixels = neopixel.NeoPixel(pixel_pin, num_pixels, brightness=0.3, auto_write=False)

# neopixel status colors
RED = (255, 0, 0)
GREEN = (0, 255, 0)
BLUE = (0, 0, 255)

# red until connecting
pixels.fill(RED)
pixels.show()

wifi.radio.connect(os.getenv('CIRCUITPY_WIFI_SSID'),
os.getenv('CIRCUITPY_WIFI_PASSWORD'))

aio_username = os.getenv('aio_username')
aio_key = os.getenv('aio_key')

pool = socketpool.SocketPool(wifi.radio)
requests = adafruit_requests.Session(pool, ssl.create_default_context())
# Initialize an Adafruit IO HTTP API object
```

```
io = IO_HTTP(aio_username, aio_key, requests)
print("connected to io")
# blue when talking to IO
pixels.fill(BLUE)
pixels.show()

try:
    # get feed
    button_feed = io.get_feed("buttonbff")
except AdafruitIO_RequestError:
    # if no feed exists, create one
    button_feed = io.create_new_feed("buttonbff")

# green once connected
pixels.fill(GREEN)
pixels.show()

# button press count sent to IO
count = 0

while True:
    try:
        # if the button is pressed..
        if not switch.value:
            # blue when talking to IO
            pixels.fill(BLUE)
            pixels.show()
            # increase by 1 with press
            count += 1
            # send count to feed
            io.send_data(button_feed["key"], count)
            print("sent %d" % count)
            print()
            # delay
            time.sleep(5)
        else:
            # green if connected
            pixels.fill(GREEN)
            pixels.show()

    # pylint: disable=broad-except
    # any errors, reset board
    except Exception as e:
        # neopixels red with an error
        pixels.fill(RED)
        pixels.show()
        print("Error:\n", str(e))
        print("Resetting microcontroller in 10 seconds")
        time.sleep(10)
        microcontroller.reset()
```

In the code, the value of the variable `count` is sent to Adafruit IO every time you press the button. The value increases by `1` with each button press. You can see this data in your Adafruit IO feed.



The NeoPixel acts as an Adafruit IO status light. If your board is connected, the NeoPixel is green. If your board is actively uploading data to Adafruit IO, the NeoPixel is blue. If there is an error or connection problem, the NeoPixel is red.

NeoPixel Python Docs

NeoPixel Python Docs (https://adafru.it/18gA)

Adafruit IO Python Docs

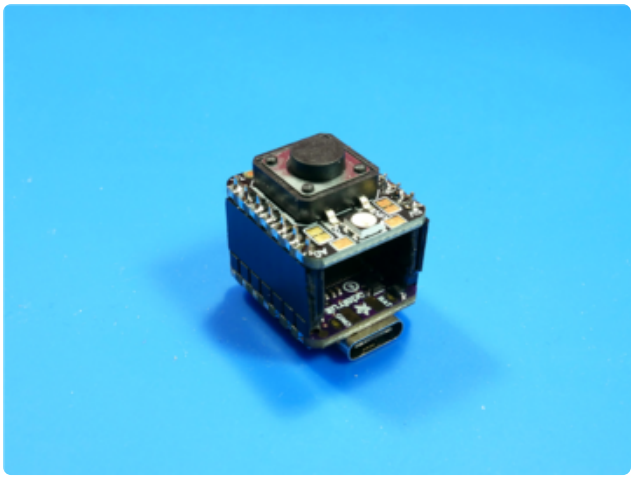
Adafruit IO Python Docs (https://adafru.it/18oc)

Arduino

Using the IoT Button with NeoPixel BFF with Arduino involves plugging the breakout into your Arduino-compatible QT Py or Xiao form factor board, installing the Adafruit_NeoPixel (https://adafru.it/aZU) and Adafruit_IO_Arduino (https://adafru.it/fpd) libraries, and running the provided example code.

Wiring

Plug an IoT Button with NeoPixel BFF into your QT Py or Xiao form factor board exactly as shown below. Here's an example of connecting a QT Py ESP32-S2 to the BFF.



Connect the QT Py ESP32-S2 with pin headers into the IoT Button with NeoPixel BFF with socket headers. They should be plugged in with the backs of the boards facing each other.

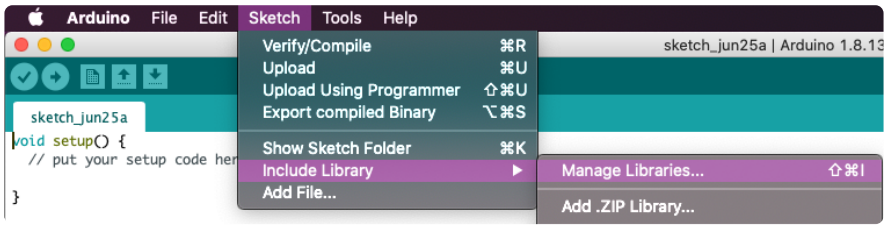
For more information on soldering socket headers, check out this Learn Guide (https://adafru.it/18gf).

How to Solder Headers Learn Guide

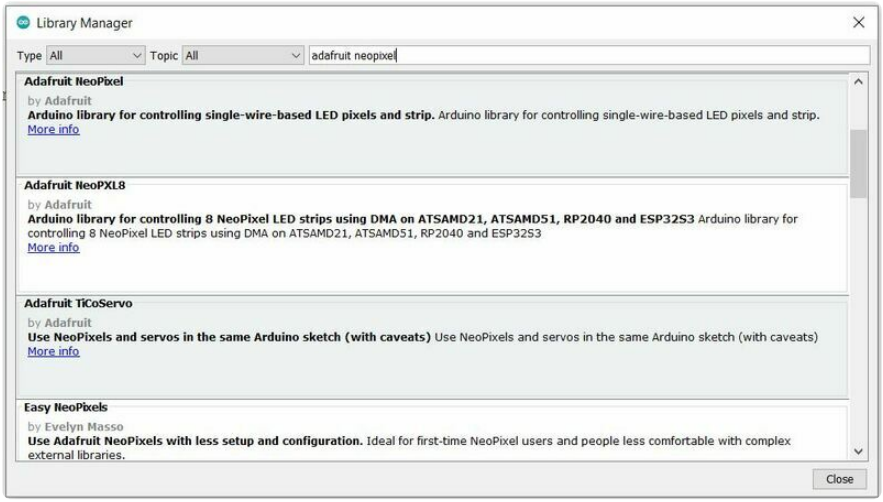
https://adafru.it/18gf

Library Installation

You can install the **Adafruit NeoPixel** and **Adafruit IO Arduino** libraries for Arduino using the Library Manager in the Arduino IDE.

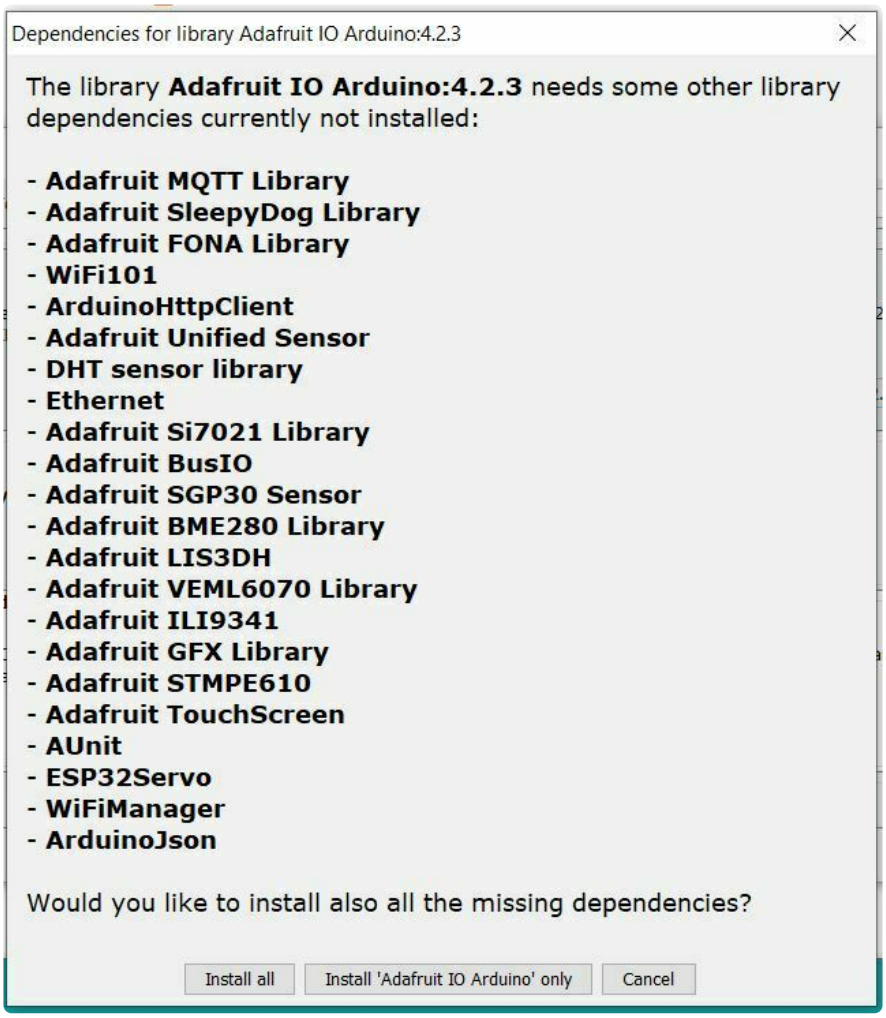


Click the **Manage Libraries ...** menu item, search for **Adafruit NeoPixel** and select the **Adafruit NeoPixel** library:



Then, search for the **Adafruit IO Arduino** library:

If asked about dependencies for any of the libraries, click "Install all".



If the "Dependencies" window does not come up, then you already have the dependencies installed.

If the dependencies are already installed, you must make sure you update them through the Arduino Library Manager before loading the example!

Basic Example

```
// SPDX-FileCopyrightText: 2023 Liz Clark for Adafruit Industries
//
// SPDX-License-Identifier: MIT

// Basic IoT Button with NeoPixel BFF Demo

#include <Adafruit_NeoPixel.h>

#define LED_PIN    A3
#define BUTTON_PIN A2
#define LED_COUNT 1

int buttonState = 0;

Adafruit_NeoPixel pixel(LED_COUNT, LED_PIN, NEO_GRB + NEO_KHZ800);

void setup() {
  pinMode(BUTTON_PIN, INPUT);
  pixel.begin();
  pixel.show();
  pixel.setBrightness(50);
}

void loop() {
  //pixel.clear();
  buttonState = digitalRead(BUTTON_PIN);

  if(buttonState == HIGH) {
    pixel.setPixelColor(0, pixel.Color(150, 0, 0));
    pixel.show();
  }

  if(buttonState == LOW) {
    pixel.setPixelColor(0, pixel.Color(0, 0, 0));
    pixel.show();
  }

}
```

Upload the sketch to your board. Press the button and you will see the NeoPixel light up red.

Adafruit IO Example

Before running this code, be sure to review the Welcome to Adafruit IO Learn Guide (https://adafru.it/BRB).

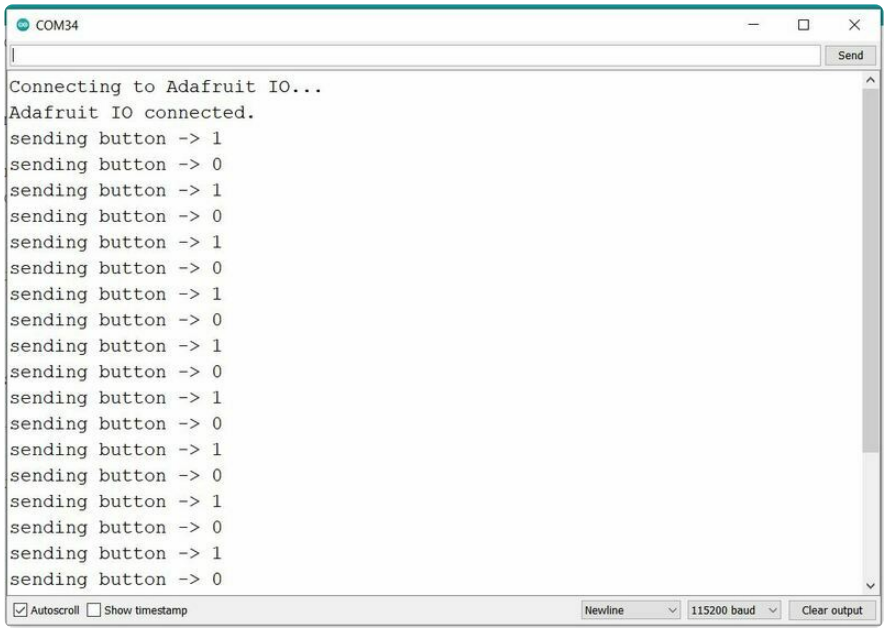
Welcome to Adafruit IO Learn Guide

https://adafru.it/BRB

```
// SPDX-FileCopyrightText: 2016 Todd Treece, Adapted 2023 Liz Clark for Adafruit
Industries
//
// SPDX-License-Identifier: MIT
```

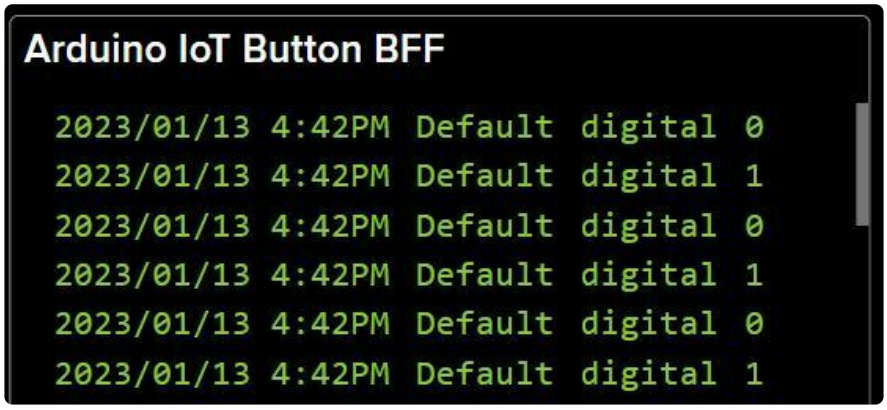
```
// Adafruit IO IoT Button with NeoPixel BFF Demo
//
// Adafruit invests time and resources providing this open source code.
// Please support Adafruit and open source hardware by purchasing
// products from Adafruit!
//
// Written by Todd Treece for Adafruit Industries
// Copyright (c) 2016 Adafruit 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, cellular,
// or ethernet clients.
#include "config.h"
#include <Adafruit_NeoPixel.h>

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

#define BUTTON_PIN A2
#define LED_PIN    A3
#define LED_COUNT 1
Adafruit_NeoPixel pixel(LED_COUNT, LED_PIN, NEO_GRB + NEO_KHZ800);

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

// set up the 'digital' feed
AdafruitIO_Feed *digital = io.feed("digital");

void setup() {
  pixel.begin();
  pixel.show();
  pixel.setBrightness(50);
  pixel.setPixelColor(0, pixel.Color(150, 0, 0));
  pixel.show();

  // set button pin as an input
  pinMode(BUTTON_PIN, INPUT);

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

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

  // 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());
  pixel.setPixelColor(0, pixel.Color(0, 150, 0));
  pixel.show();

}

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();

  // grab the current state of the button.
  // we have to flip the logic because we are
  // using a pullup resistor.
  if(digitalRead(BUTTON_PIN) == LOW){
    current = true;
    pixel.setPixelColor(0, pixel.Color(0, 0, 150));
    pixel.show();
  }
  else {
    current = false;
    pixel.setPixelColor(0, pixel.Color(0, 150, 0));
    pixel.show();
  }
  // return if the value hasn't changed
  if(current == last)
    return;

  // save the current state to the 'digital' feed on adafruit io
  Serial.print("sending button -> ");
  Serial.println(current);
  digital->save(current);

  // store last button state
  last = current;

}
```



Edit the included **config.h** file with your Adafruit IO username, Adafruit IO key, SSID name and SSID password. Save the file and then upload the sketch to your board and open up the Serial Monitor (**Tools -> Serial Monitor**) at 115200 baud. While your board is connecting to Adafruit IO, you'll see "`Connecting to Adafruit IO...`" print in the Serial Monitor. After establishing a connection, "`Adafruit IO connected.`" will print.

When you press the button, a value of `0` or `1` will be sent to Adafruit IO and will print to the Serial Monitor. You'll be able to see this data in your feed on Adafruit IO.



The NeoPixel acts as an Adafruit IO status light. Before connecting to Adafruit IO, the NeoPixel is red. If your board is connected, the NeoPixel is green. If your board is actively uploading data to Adafruit IO, the NeoPixel is blue.

NeoPixel Arduino Library Docs

NeoPixel Arduino Library Docs (https://adafru.it/18gB)

Adafruit IO Arduino Library Docs

Adafruit IO Arduino Library Docs (https://adafru.it/18of)

Downloads

Files

- NeoPixel Datasheet (http://adafru.it/56665666)
- Tactile Switch Datasheet (http://adafru.it/56665666)
- EagleCAD PCB files on GitHub (https://adafru.it/18oA)
- Fritzing object in the Adafruit Fritzing Library (https://adafru.it/18oB)

Schematic and Fab Print