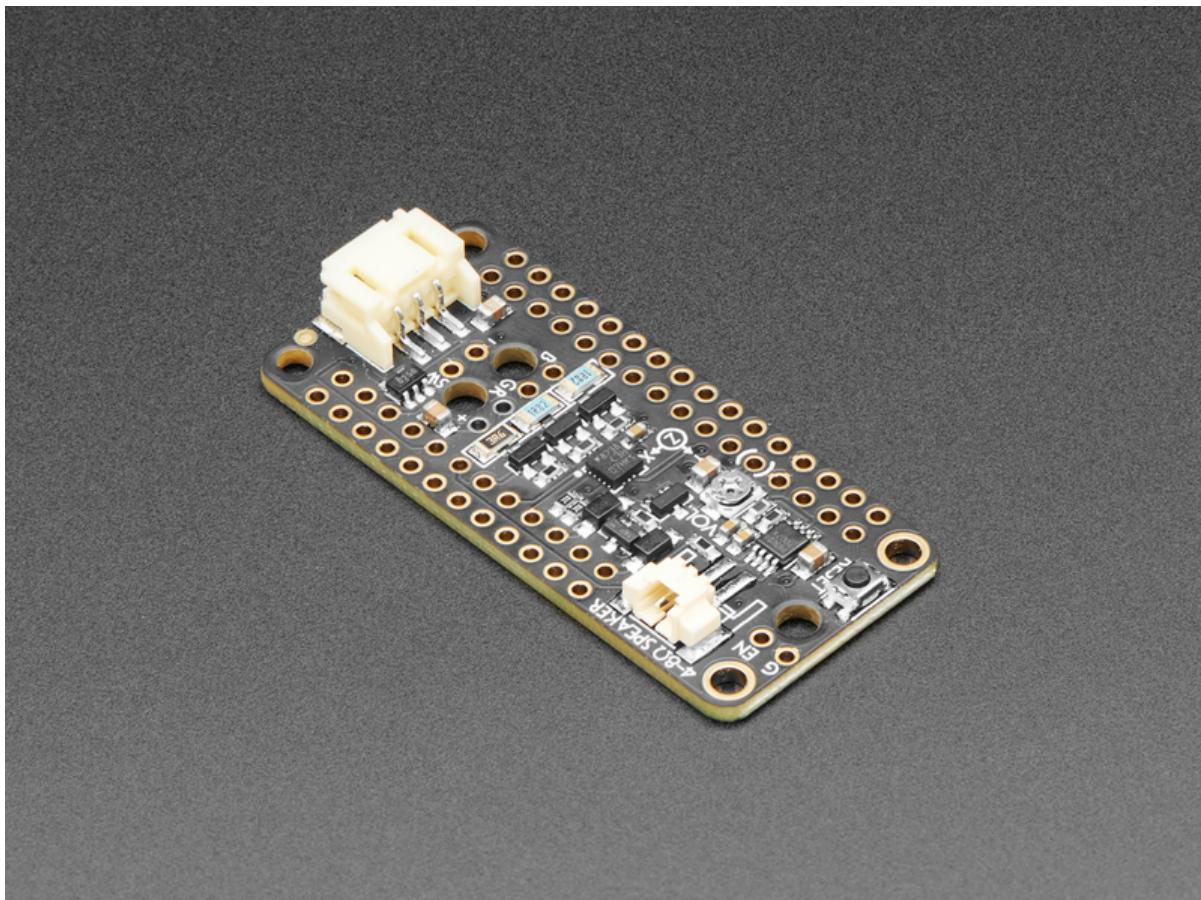




Adafruit Prop-Maker FeatherWing

Created by Kattni Rembor



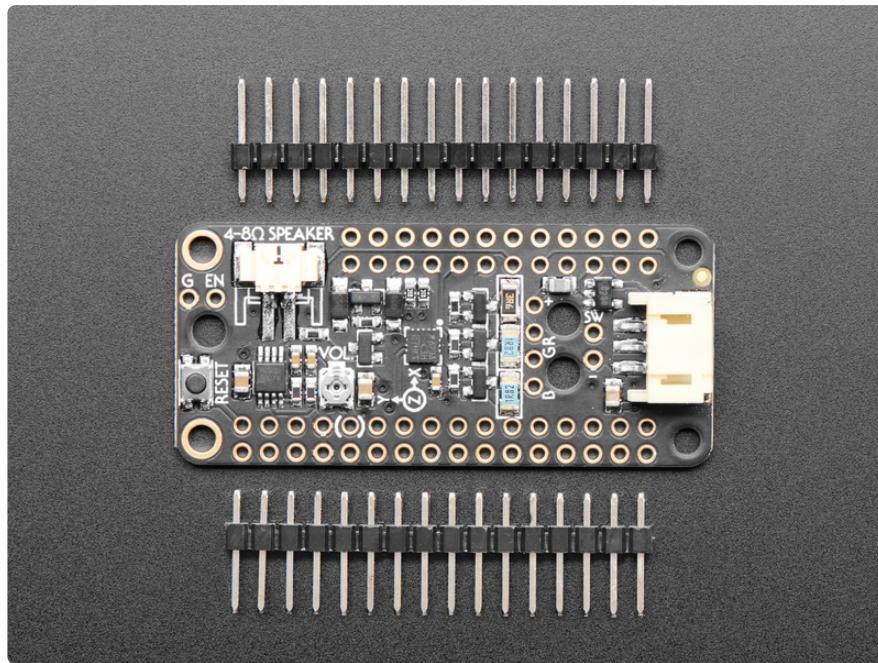
<https://learn.adafruit.com/adafruit-prop-maker-featherwing>

Last updated on 2024-06-03 02:37:02 PM EDT

Table of Contents

Overview	3
Pinouts	5
• Logic Pins	
• Power Pins	
• NeoPixel Connector	
• Audio	
• 3 Watt LED	
• Switch Pins	
• The PWR Pin	
Arduino Code	11
• NeoPixels	
• Audio	
• Accelerometer	
• 3 Watt LED	
• Switch	
CircuitPython Code	18
• NeoPixels	
• Audio	
• Accelerometer	
• 3 Watt LED	
• Switch	
Downloads	22
• Files	
• Schematics and Fab Print	

Overview

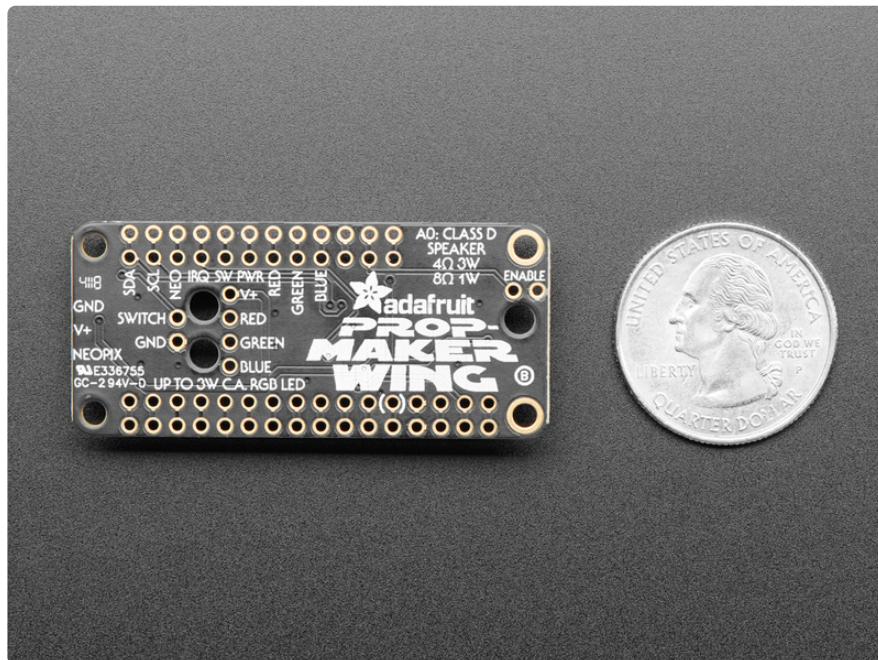


The Adafruit Feather series gives you lots of options for a small, portable, rechargeable microcontroller board. Perfect for fitting into your next prop build! This FeatherWing will unlock the prop-maker inside all of us, with tons of stuff packed in to make sabers & swords, props, toys, cosplay pieces, and more.

We looked at hundreds of prop builds, and thought about what would make for a great low-cost (but well-designed) add-on for our Feather boards. Here's what we came up with:

- **Snap-in NeoPixel port** - With a 3-pin JST connector, you can [plug in one of our JST-wired NeoPixel strips directly](#) (<http://adafru.it/3919>), or [use a 3-pin JST connector to wire up your favorite shape of addressable NeoPixel LEDs](#) (<https://adafru.it/DBh>). This port provides high current drive from either the Feather Lipoly or USB port, whichever is higher. A level shifter gives you a clean voltage signal to reduce glitchiness no matter what chip you're using
- **3W RGB LED drivers** - 3 high current MOSFETs will let you [connect a 3W RGB LED for powerful eye-blasting glory](#) (<http://adafru.it/2530>). For most Feathers, the 3 pins are PWM capable so you can generate any color you like. Available as pin breakouts plus strain-relief holes
- **Triple-Axis Accelerometer with Tap Detection** - The LIS3DH is our favorite accelerometer, you can use this for detection motion, tilt or taps. [Here's an example of a light saber that makes sounds when swung or hit.](#) (<https://adafru.it/DBi>) We have code for this chip in both Arduino and CircuitPython.

- **Class D Audio Amplifier** - Drive a 8Ω 1Watt speaker or 4Ω 3W speaker for sound effects. [Plug and play with our cute and slim oval speaker](http://adafru.it/3923) (<http://adafru.it/3923>), or [connect a picoblade cable](http://adafru.it/3922) (<http://adafru.it/3922>) for your favorite speaker. For use only with Feathers that have analog audio out such as the Feather M0 Express and M4 series.
- **Low power mode!** The power system for the RGB LED, NeoPixels and speaker amplifier can be controlled by a pin to cut power to them, so you have lower power usage when the prop is in sleep or off mode (but can wake up fast by listening to the button press or accelerometer data). When the power pin is set low, the current draw for just the wing is under 1mA and no there's current draw from any attached NeoPixels - normally they're about 1mA even when not lit.
- Breakouts plus strain-relief hole for the enable pin and ground (for a mechanical switch that will power down the whole board)
- Breakouts plus strain-relief holes for an external switch pin and ground (for a mechanical mode button)



Please note: A few of the onboard hardware elements use PWM and analog output so we recommend the **Feather M0 Express** or **Feather M4** series, they'll work best with this wing and let you make the most of it. For example:

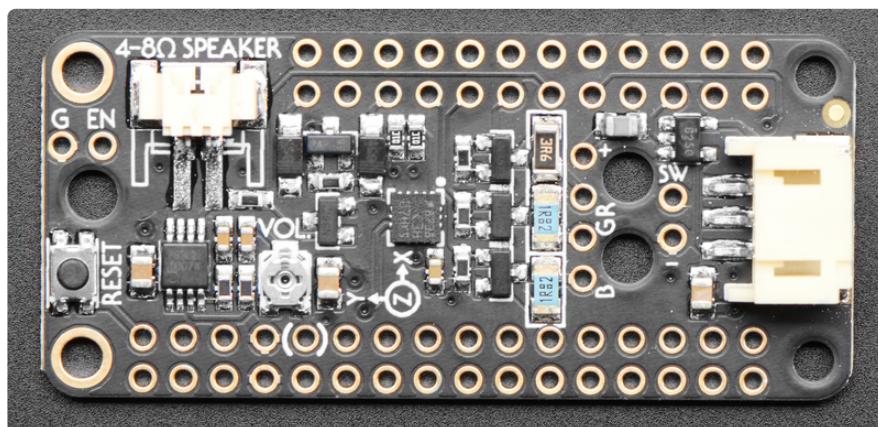
- Feather 32u4 and 328p do not have PWMs for all the RGB LED pins, and no analog audio output support
- Feather nRF52, ESP32 ESP8266 do have PWMs on all the RGB LED pins but no analog audio output support
- Feather M0 basic boards only have audio output support in Arduino, not CircuitPython.

That doesn't mean you can't use this 'wing with the Feather ESP8266 or nRF52832, just that you won't get any sound effects. You can still use the accelerometer, NeoPixels, RGB LED, etc.

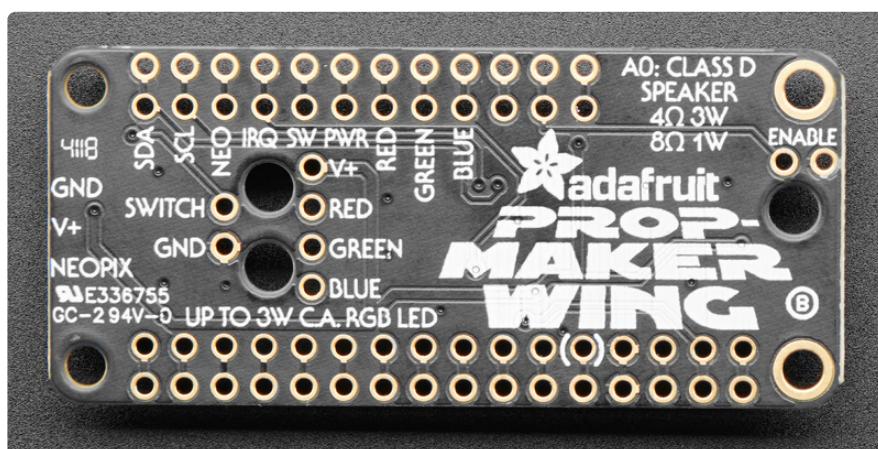
As you can tell, the **M0 Express** and **M4** series is what we recommend cause you'll get everything and with CircuitPython, its really easy to play audio directly off the built in flash chips!

Comes as an assembled wing with some headers. Some light soldering is required to attach to your Feather.

Pinouts

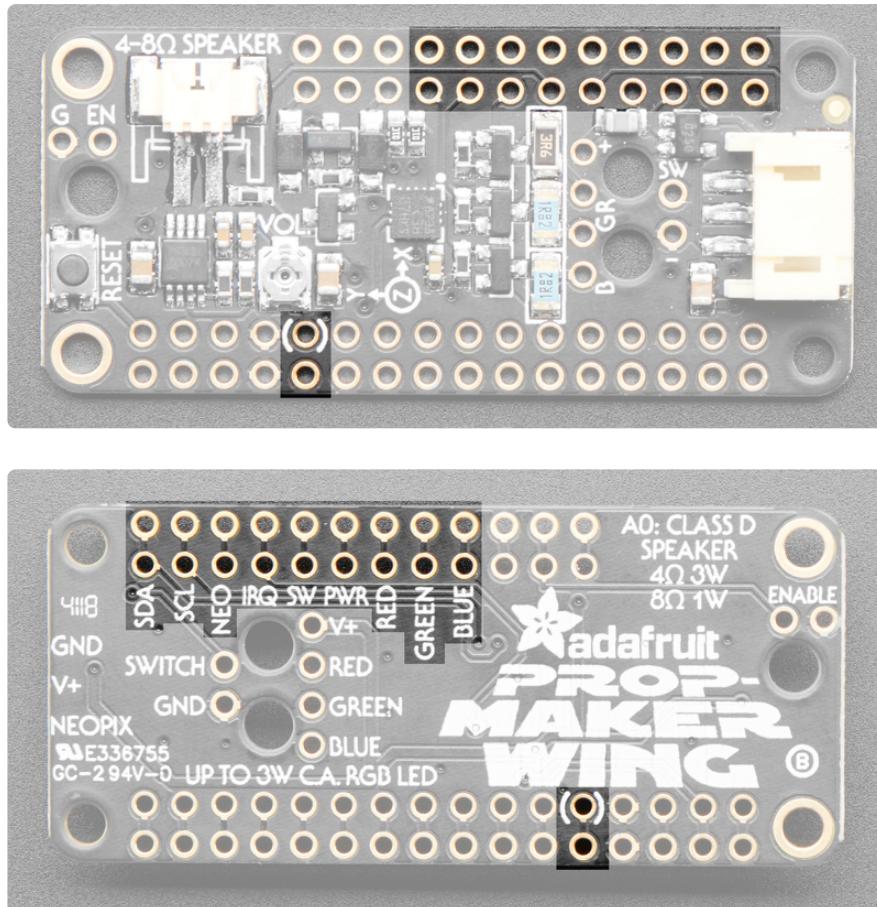


The Adafruit Prop-Maker FeatherWing is designed to unlock the prop-maker inside all of us, with tons of stuff packed in to make sabers & swords, props, toys, cosplay pieces, and more.



Let's take a tour of the board!

Logic Pins



These are the logic pins used by the FeatherWing.

We note the Feather M0 and M4 pin numbers these are connected but other Feather variants will have different pin numbers! Check your Feather silkscreen or documentation to check!

Bottom Row

- **A0** - This pin is true analog input to the audio amplifier on the FeatherWing. Matches A0 on the Feather M0's and M4's which is the DAC output. Other Feathers (like the ESP8266) don't have a DAC so you can leave this pin unused

Top Row

- **Blue** - Blue pin for 3W RGB LED - connected through a MOSFET. Connected to D13 on Feather M0 or M4.
- **Green** - Green pin for 3W RGB LED - connected through a MOSFET. Connected to D12 on Feather M0 or M4.

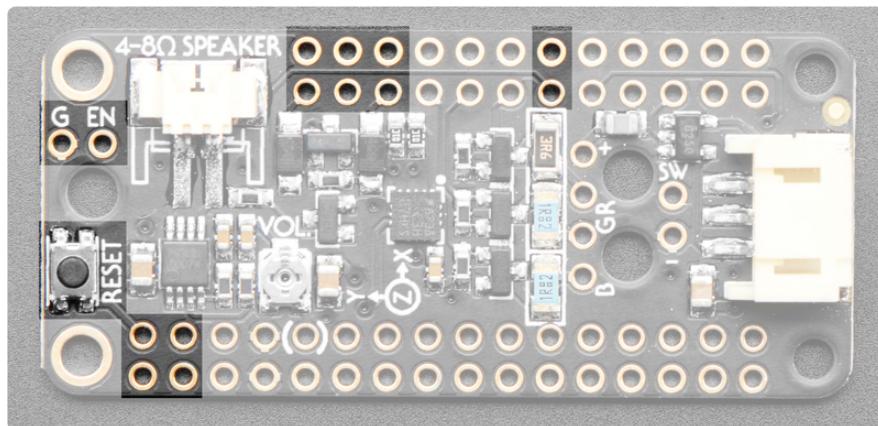
- **Red** - Red pin for 3W RGB LED - connected through a MOSFET. Connected to D11 on Feather M0 or M4.
- **Power (PWR)** - This pin is used to enable power to the NeoPixels, audio amplifier and RGB LED. By default it is pulled low so you must set high to turn on the NeoPixels/Audio/RGB. Connected to D10 on Feather M0 or M4.
- **Switch (SW)** - This is connected to the Switch pin, for button or switch sensing. There's no pullup resistor so you'll have to connect to D9 on Feather M0 or M4.
- **IRQ** - Interrupt pin for LIS3DH accelerometer, used for tap or shake detection. Connected to D6 on Feather M0 or M4.
- **NeoPixel (NEO)** - Data pin for NeoPixel connector. This pin is level shifted to 5V output. Connected to D5 on Feather M0 or M4.
- **SCL** - I2C clock pin. Connected to SCL on the Feather. This pin is used to control the onboard I2C accelerometer
- **SDA** - I2C data pin. Connected to SDA on the Feather. This pin is used to control the onboard I2C accelerometer

The top row pins are labeled on the back of the board!

Available Logic Pins

The logic pins in the longer section along the bottom on the PropMaker FeatherWing are not used, **except for A0**. These pins differ by Feather, therefore a list is not included here. The pins are available to use for other purposes, such as multiple buttons, etc.

Power Pins



Top Row

- **VBAT** - Connected to BAT power on the Feather.
- **Enable** - Enable pin for switch that will power down entire board.
- **VBUS** - Connected to USB power on Feather.
- **Power (PWR)** - This pin is used to enable power to the NeoPixels, audio amplifier and RGB LED. By default it is pulled low so **you must set high to turn on the NeoPixels/Audio/RGB**. Connected to D10 on Feather M0 or M4.

Bottom Row

- **Reset pin** - Tie to ground to reset the Feather
- **3.3V pin** - 3.3V output from the Feather's regulator

Left Side

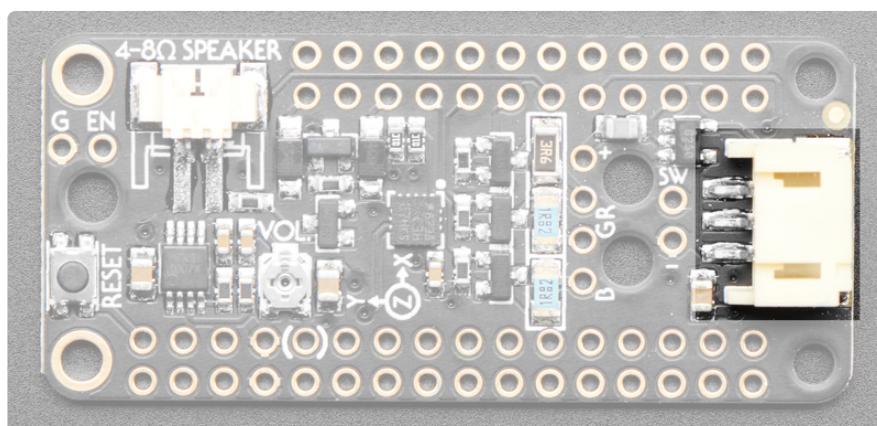
- **RESET button** - Resets the Feather and the FeatherWing.

The pins on the left above the RESET button are for a mechanical switch that will power down the whole board.

- **G** - Ground pin for mechanical switch
- **EN** - Enable pin, connected to the Enable pin on the Feather. Connect this pin to ground to completely turn off the Feather and FeatherWing (e.g. a power switch)

NeoPixel Connector

You must set the PWR pin high to enable power for the NeoPixel output.

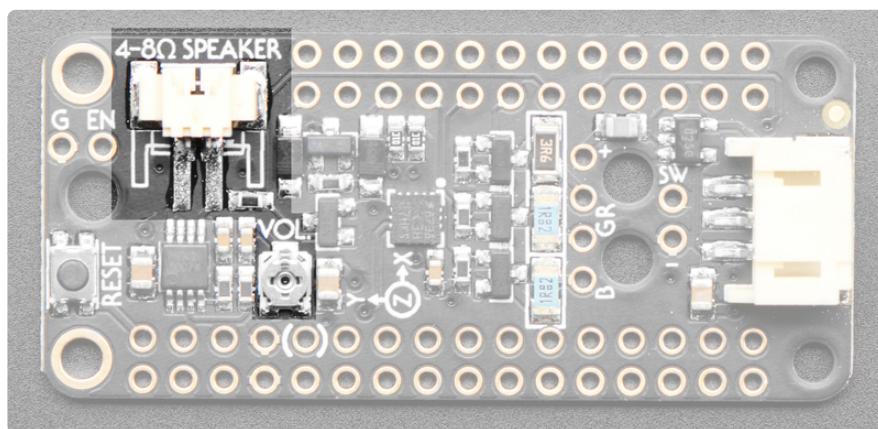


This is the **Snap-In NeoPixel port!** With a 3-pin JST connector, you can [plug in one of our JST-wired NeoPixel strips directly](#) (<http://adafru.it/3919>), or [use a 3-pin JST connector to wire up your favorite shape of addressable NeoPixel LEDs](#) (<https://adafru.it/DBh>). This port provides high current drive from either the Feather Lipoly or USB port, whichever is higher. A level shifter gives you a clean voltage signal to reduce glitchiness no matter what chip you're using!

The pinout for the connector is labeled on the back of the board.

Audio

You must set the PWR pin high to enable power for the Audio output.



The Prop-Maker FeatherWing includes a **audio connector** for a 8Ω 1Watt speaker or 4Ω 3Watt speaker for sound effects. [Plug and play with our cute and slim oval speaker](#) (<http://adafru.it/3923>), or [connect a picoblade cable](#) (<http://adafru.it/3922>) for your favorite speaker. The connector uses pin A0.

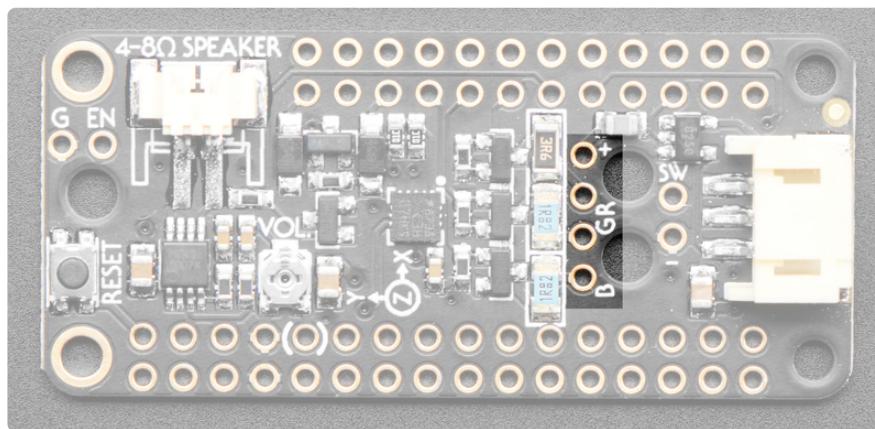
For use only with Feathers that have analog audio out such as the Feather M0 Express and M4 series.

There also **volume control** on the board between A0 and the speaker connector. Rotate it to increase or decrease the audio volume from the connected speaker!

To rotate the volume control, gently use a small flathead screwdriver!

3 Watt LED

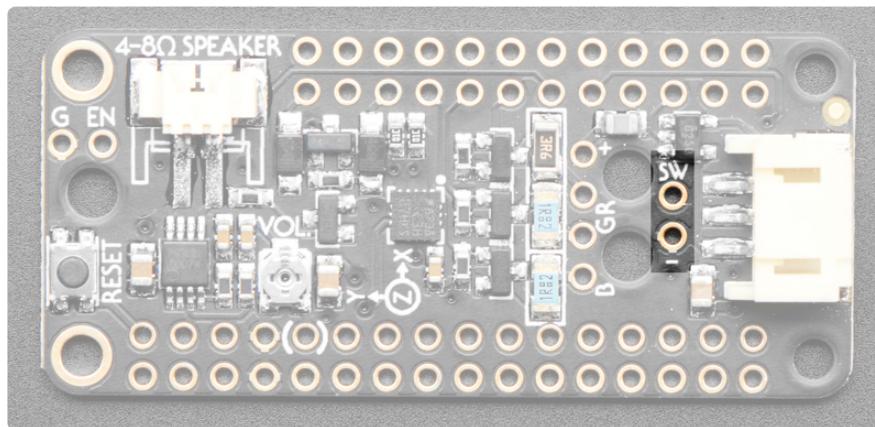
You must set the PWR pin high to enable power for the LED V+ output.



3 high current MOSFETs will let you [connect a 3W RGB LED for powerful eye-blasting glory](http://adafru.it/2530) (<http://adafru.it/2530>). For most Feathers, the 3 pins are PWM capable so you can generate any color you like. Available as pin breakouts plus strain-relief holes. The pin labels are also on the back of the board.

- **B (Blue)** - Connect to B pin on LED. Uses D13 on the Feather M0 or M4.
- **G (Green)** - Connect to the G pin on LED. Uses D12 on the Feather M0 or M4.
- **R (Red)** - Connect to the R pin on the LED. Uses D11 on the Feather M0 or M4.
- **+ (V+)** - Connect to the power (+) pin on the LED.

Switch Pins



External switch pins for a mechanical switch.

- **SW (Switch)** - Connected to D9 on the Feather M0 or M4

• - (Gnd) -

The PWR Pin

As mentioned above, the PWR pin is low (off) by default. To turn on power to the NeoPixels, Audio output, and RGB LED output, you must set this pin high (on).

You can check the pinout for your Feather to figure out which pin to use. But here's a quick reference for some of the boards:

- Feather M0/M4 = **D10**
- Feather 32u4 = **D10**
- Feather 328 = **D10**
- Feather NRF52 = **D11**
- Feather ESP8266 = **D15**
- Feather ESP32 = **D33**

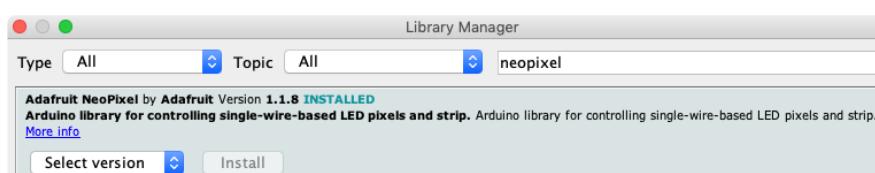
And be sure to checkout the examples to see how this is done in code.

Arduino Code

NeoPixels

You can use Arduino to control the NeoPixels connected via the snap-in NeoPixel connector on the Prop-Maker FeatherWing.

To use NeoPixels with the Prop-Maker FeatherWing, install the [Adafruit NeoPixel library](https://adafru.it/aZU) (<https://adafru.it/aZU>) from the Arduino Library Manager.



This example lights up a 30-pixel NeoPixel strip with a rainbow pattern.

Change `NUM_PIXELS` to match the number of NeoPixels you've connected to your FeatherWing.

```
// SPDX-FileCopyrightText: 2019 Brent Rubell for Adafruit Industries
// SPDX-License-Identifier: MIT
```

```

/*
 * Adafruit Prop-Maker Featherwing
 * NeoPixel Example
 *
 * Rainbow example for 30-Pixel NeoPixel Strip
 */
#include <Adafruit_NeoPixel.h>

#ifndef USE_TINYUSB // For Serial when selecting TinyUSB
#include <Adafruit_TinyUSB.h>
#endif

// NeoPixel strip length, in pixels
#define NUM_PIXELS 30

#if defined(__SAMD21G18A__) || defined(__AVR_ATmega32U4__) || defined(NRF52840_XXAA)
#define NEOPIXEL_PIN 5
#define POWER_PIN 10
#elif defined(__AVR_ATmega328P__)
#define NEOPIXEL_PIN 5
#define POWER_PIN 10
#elif defined(NRF52)
#define NEOPIXEL_PIN 27
#define POWER_PIN 11
#elif defined(ESP8266)
#define NEOPIXEL_PIN 2
#define POWER_PIN 15
#elif defined(TEEENSYDUINO)
#define NEOPIXEL_PIN 8
#define POWER_PIN 10
#elif defined(ESP32)
#define NEOPIXEL_PIN 14
#define POWER_PIN 33
#endif

// create a neopixel strip
Adafruit_NeoPixel strip = Adafruit_NeoPixel(NUM_PIXELS, NEOPIXEL_PIN, NEO_GRB +
NEO_KHZ800);

void setup() {
    Serial.begin(115200);
    Serial.println("\nProp-Maker Wing: NeoPixel");

    // Set power pin to output
    pinMode(POWER_PIN, OUTPUT);
    // Disable the pin, we're not currently writing to the neopixels.
    digitalWrite(POWER_PIN, LOW);

    // This initializes the NeoPixel library.
    strip.begin();
}

uint32_t Wheel(byte WheelPos) {
    // Input a value 0 to 255 to get a color value.
    // The colours are a transition r - g - b - back to r.
    WheelPos = 255 - WheelPos;
    if(WheelPos < 85) {
        return strip.Color(255 - WheelPos * 3, 0, WheelPos * 3);
    }
    if(WheelPos < 170) {
        WheelPos -= 85;
        return strip.Color(0, WheelPos * 3, 255 - WheelPos * 3);
    }
    WheelPos -= 170;
    return strip.Color(WheelPos * 3, 255 - WheelPos * 3, 0);
}

```

```
void rainbow(uint8_t wait) {
    uint16_t i, j;

    for(j=0; j<256; j++) {
        for(i=0; i<strip.numPixels(); i++) {
            strip.setPixelColor(i, Wheel((i+j) & 255));
        }
        // turn on the Prop-Maker FeatherWing's power pin
        digitalWrite(POWER_PIN, HIGH);
        // write the pixel values
        strip.show();
        delay(wait);
    }
}

void loop()
{
    // cycle a the rainbow with a 20ms wait
    rainbow(20);
}
```

For more information on using NeoPixels, check out the [Adafruit NeoPixel Überguide](#) (<https://adafru.it/dhw>)!

Audio

While technically the Feather M0 and M4 have an analog output pin that is connected to the amplifier on this Wing, Arduino does not have great audio playback support. So we don't recommend using Arduino if you need to play audio.

If you'd like to use the Adafruit Prop-Maker FeatherWing's Audio capabilities, [check out the CircuitPython code](#) (<https://adafru.it/DLx>).

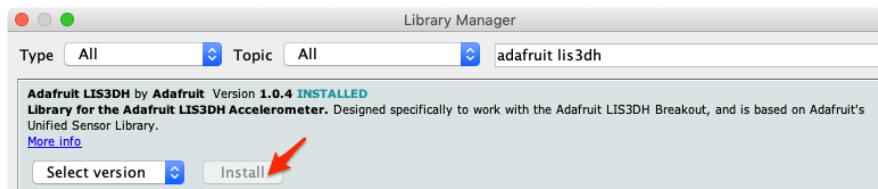
Other than the M0 and M4 series, most Feathers don't have analog out on `A0` at all, so you wouldn't be able to use this feature anyways.

Accelerometer

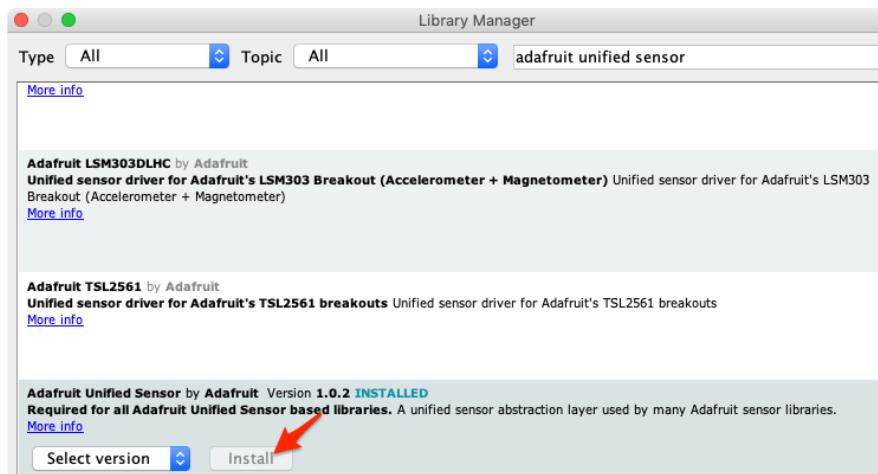
You can easily access the acceleration data from the LIS3DH accelerometer in the center of the Prop-Maker FeatherWing using Arduino.

To use the Accelerometer on the Prop-Maker FeatherWing, you'll need to install two Arduino Libraries.

First, [install the Adafruit LIS3DH library](#) (<https://adafru.it/jzb>) from the Arduino Library Manager.



Then, install [the Adafruit Unified Sensor Library](https://adafru.it/aZm) (<https://adafru.it/aZm>) from the Arduino Library Manager:



This example prints the acceleration data to the [serial monitor](https://adafru.it/DLy) (<https://adafru.it/DLy>). Move the board around to see the values change.

The LIS3DH can also detect taps - quickly tap the Prop-Maker FeatherWing to display a tap on the Serial Monitor.

```
// SPDX-FileCopyrightText: 2019 Brent Rubell for Adafruit Industries
// SPDX-License-Identifier: MIT

/*
 * Adafruit Prop-Maker Featherwing
 * Accelerometer Example
 *
 * Prints acceleration data to the Serial Monitor
 */
// include the accelerometer library
#include <Adafruit_LIS3DH.h>

// Adjust this number for the sensitivity of the 'click' force
// this strongly depend on the range! for 16G, try 5-10
// for 8G, try 10-20. for 4G try 20-40. for 2G try 40-80
#define CLICKTHRESHOLD 40

// inst. i2c accelerometer
Adafruit_LIS3DH lis = Adafruit_LIS3DH();

void setup() {
    Serial.begin(115200);
    Serial.println("\nProp-Maker Wing: Accelerometer");

    Serial.println("Starting LIS3DH...");
    if (! lis.begin(0x18)) { // change this to 0x19 for alternative i2c address
        Serial.println("Couldn't start LIS3DH");
```

```

        while (1);
    }
    Serial.println("LIS3DH Started!");

    // Set Accelerometer Range (2, 4, 8, or 16 G!)
    lis.setRange(LIS3DH_RANGE_4_G);

    /* Set Accelerometer Click Detection
     * 0 = turn off click detection & interrupt
     * 1 = single click only interrupt output
     * 2 = double click only interrupt output
     * NOTE: Higher numbers are less sensitive
     */
    lis.setClick(1, CLICKTHRESHOLD);

}

void loop() {

    // Detect a click
    uint8_t click = lis.getClick();

    // Get a new accel. sensor event
    sensors_event_t event;
    lis.getEvent(&event);

    /* Display the results (acceleration is measured in m/s^2) */
    Serial.print("\t\tX: "); Serial.print(event.acceleration.x);
    Serial.print("\t\tY: "); Serial.print(event.acceleration.y);
    Serial.print("\t\tZ: "); Serial.print(event.acceleration.z);
    Serial.println(" m/s^2 ");
    Serial.println();

    delay(1000);

    // detect click
    if (click & 0x10) Serial.print(" single click");
}

```

3 Watt LED

You can control the colors and brightness of a 3 Watt RGB LED using Arduino.

This example uses PWM to light up the LED with a rainbow swirl.

Not all Feather boards have PWMs available on the RGB pins! If so you won't see a smooth color swirl.

```

// SPDX-FileCopyrightText: 2019 Brent Rubell for Adafruit Industries
//
// SPDX-License-Identifier: MIT

/*
 * Adafruit Prop-Maker Featherwing
 * LED Example
 *
 * Rainbow swirl example for 3W LED.
 */

#ifndef USE_TINYUSB // For Serial when selecting TinyUSB

```

```

#include <Adafruit_TinyUSB.h>
#endif

#if defined(__SAMD21G18A__) || defined(__AVR_ATmega32U4__) || defined(NRF52840_XXAA)
    // No green PWM on 32u4
    #define POWER_PIN      10
    #define RED_LED        11
    #define GREEN_LED      12 // no PWM on atmega32u4
    #define BLUE_LED       13
#elif defined(__AVR_ATmega328P__)
    // No Red or Blue PWM
    #define POWER_PIN      10
    #define RED_LED        2
    #define GREEN_LED      3 // the only PWM pin!
    #define BLUE_LED       4
#elif defined(NRF52)
    #define POWER_PIN      11
    #define RED_LED        7
    #define GREEN_LED      15
    #define BLUE_LED       16
#elif defined(ESP8266)
    #define POWER_PIN      15
    #define RED_LED        13
    #define GREEN_LED      12
    #define BLUE_LED       14
#elif defined(TEENSYDUINO)
    #define POWER_PIN      10
    #define RED_LED        9
    #define GREEN_LED      6
    #define BLUE_LED       5
#elif defined(ARDUINO_ADAFRUIT_FEATHER_ESP32S2)
    #define POWER_PIN      10
    #define RED_LED        0
    #define GREEN_LED      1
    #define BLUE_LED       2
    #define RED_PIN         11
    #define GREEN_PIN       12
    #define BLUE_PIN        13
    #define analogWrite ledcWrite
#elif defined(ESP32)
    #define POWER_PIN      33
    #define RED_LED        0
    #define GREEN_LED      1
    #define BLUE_LED       2
    #define RED_PIN         27
    #define GREEN_PIN       12
    #define BLUE_PIN        13
    #define analogWrite ledcWrite
#endif

uint8_t i=0;

uint8_t red_out = RED_LED;
uint8_t green_out = GREEN_LED;
uint8_t blue_out = BLUE_LED;

void setup() {
    Serial.begin(115200);
    Serial.println("\nProp-Maker Wing: LED Example");

    // set up the power pin
    pinMode(POWER_PIN, OUTPUT);
    // disable the power pin, we're not writing to the LEDs
    digitalWrite(POWER_PIN, LOW);

    // Set up the LED Pins
    #if defined(ESP32) // and ESP32-S2!
        #if ESP_IDF_VERSION >= ESP_IDF_VERSION_VAL(5, 1, 1)
            // newer LEDC API, use pins instead of channel

```

```

red_out = RED_PIN;
green_out = GREEN_PIN;
blue_out = BLUE_PIN;
ledcAttach(RED_PIN, 5000, 8);
ledcAttach(GREEN_PIN, 5000, 8);
ledcAttach(BLUE_PIN, 5000, 8);
#else
// older LEDC API, use channel, attach pin to channel
ledcSetup(RED_LED, 5000, 8);
ledcAttachPin(RED_PIN, RED_LED);
ledcSetup(GREEN_LED, 5000, 8);
ledcAttachPin(GREEN_PIN, GREEN_LED);
ledcSetup(BLUE_LED, 5000, 8);
ledcAttachPin(BLUE_PIN, BLUE_LED);
#endif
#else
pinMode(red_out, OUTPUT);
pinMode(green_out, OUTPUT);
pinMode(blue_out, OUTPUT);
#endif

analogWrite(red_out, 0);
analogWrite(green_out, 0);
analogWrite(blue_out, 0);
}

uint32_t Color(uint8_t r, uint8_t g, uint8_t b) {
    return ((uint32_t)r << 16) | ((uint32_t)g << 8) | b;
}

uint32_t Wheel(byte WheelPos) {
    WheelPos = 255 - WheelPos;
    if(WheelPos < 85) {
        return Color(255 - WheelPos * 3, 0, WheelPos * 3);
    }
    if(WheelPos < 170) {
        WheelPos -= 85;
        return Color(0, WheelPos * 3, 255 - WheelPos * 3);
    }
    WheelPos -= 170;
    return Color(WheelPos * 3, 255 - WheelPos * 3, 0);
}

void loop()
{
    // set RGB Colors
    uint32_t color = Wheel(i++);
    uint8_t red = color >> 16;
    uint8_t green = color >> 8;
    uint8_t blue = color;

    // turn on the power pin
    digitalWrite(POWER_PIN, HIGH);

    // write colors to the 3W LED
    analogWrite(red_out, red);
    analogWrite(green_out, green);
    analogWrite(blue_out, blue);
    delay(2);
}

```

Switch

You can easily use a mechanical switch on the Switch pin with Arduino.

The following example prints to the [serial monitor](https://adafru.it/DLy) (<https://adafru.it/DLy>) when the switch is closed.

```
// SPDX-FileCopyrightText: 2019 Brent Rubell for Adafruit Industries
//
// SPDX-License-Identifier: MIT

/*
 * Adafruit Prop-Maker Featherwing
 * Switch Example
 *
 * Print to the Serial Monitor when a switch is pressed.
 */

#ifndef USE_TINYUSB // For Serial when selecting TinyUSB
#include <Adafruit_TinyUSB.h>
#endif

#if defined(__SAMD21G18A__) || defined(__AVR_ATmega32U4__) || defined(NRF52840_XXAA)
#define SWITCH_PIN 9
#elif defined(__AVR_ATmega328P__)
#define SWITCH_PIN 9
#elif defined(NRF52)
#define SWITCH_PIN 31
#elif defined(ESP8266)
#define SWITCH_PIN 0
#elif defined(TEENSYDUINO)
#define SWITCH_PIN 4
#elif defined(ESP32)
#define SWITCH_PIN 15
#endif

void setup() {
    Serial.begin(115200);
    Serial.println("\nProp-Maker Wing: Switch Example");
    // set up the switch
    pinMode(SWITCH_PIN, INPUT_PULLUP);
}

void loop()
{
    if (!digitalRead(SWITCH_PIN)) {
        Serial.println("Switch pressed!");
    }
}
```

CircuitPython Code

The Prop-Maker FeatherWing is loaded with features to help with your next build. Here, we'll go over how to use each of the features in CircuitPython. We'll provide a simple example that you can expand on to include it in your project. Save the code to `code.py` on your board to run the example!

NeoPixels

CircuitPython makes it super simple to use the NeoPixels connected via the snap-in NeoPixel connector on the Prop-Maker FeatherWing.

This example lights up a 30-pixel NeoPixel strip in a rainbow. Change `NUM_PIXELS` to match the number of NeoPixels you've connected to your FeatherWing.

```
# SPDX-FileCopyrightText: 2019 Kattni Rembor for Adafruit Industries
#
# SPDX-License-Identifier: MIT

"""Simple rainbow example for 30-pixel NeoPixel strip"""
import digitalio
import board
from rainbowio import colorwheel
import neopixel

NUM_PIXELS = 30 # NeoPixel strip length (in pixels)

enable = digitalio.DigitalInOut(board.D10)
enable.direction = digitalio.Direction.OUTPUT
enable.value = True

strip = neopixel.NeoPixel(board.D5, NUM_PIXELS, brightness=1)

while True:
    for i in range(255):
        strip.fill((colorwheel(i)))
```

For more information, check out the [Adafruit NeoPixel Überguide](https://adafru.it/dhw) (<https://adafru.it/dhw>)!

Audio

The speaker connector on the Prop-Maker FeatherWing makes it easy to attach a speaker to playback audio for your project.

This example only works on Feathers that have analog audio out, such as the Feather M0 Express or Feather M4.

This example plays once through a wave file called "StreetChicken.wav". Click the button below to download the file and copy it to your board. You can also add any [compatible wave file](https://adafru.it/DFb) (<https://adafru.it/DFb>) and change the code to match the name of your file.

[StreetChicken.wav](#)

```
# SPDX-FileCopyrightText: 2019, 2023 Kattni Rembor for Adafruit Industries
#
# SPDX-License-Identifier: MIT

"""Simple example to play a wave file"""
# This example only works on Feathers that have analog audio out!
import digitalio
import board
import audiocore

try:
    from audioio import AudioOut
except ImportError:
    try:
        from audiopwmio import PWMAudioOut as AudioOut
    except ImportError:
        pass # not always supported by every board!

WAV_FILE_NAME = "StreetChicken.wav" # Change to the name of your wav file!

enable = digitalio.DigitalInOut(board.D10)
enable.direction = digitalio.Direction.OUTPUT
enable.value = True

with AudioOut(board.A0) as audio: # Speaker connector
    wave_file = open(WAV_FILE_NAME, "rb")
    wave = audiocore.WaveFile(wave_file)

    audio.play(wave)
    while audio.playing:
        pass
```

For more information, check out [Audio Out in the CircuitPython Essentials Guide \(https://adafru.it/BRj\)](https://adafru.it/BRj)!

Accelerometer

You can easily access the acceleration data from the LIS3DH accelerometer in the center of the Prop-Maker FeatherWing using CircuitPython.

This example prints the acceleration data to the [serial console \(https://adafru.it/Bec\)](https://adafru.it/Bec). Move the board around to see the values change.

```
# SPDX-FileCopyrightText: 2019 Kattni Rembor for Adafruit Industries
#
# SPDX-License-Identifier: MIT

"""Simple example to print acceleration data to console"""
import time
import digitalio
import board
import busio
import adafruit_lis3dh

# Set up accelerometer on I2C bus, 4G range:
i2c = busio.I2C(board.SCL, board.SDA)
```

```

int1 = digitalio.DigitalInOut(board.D6)
accel = adafruit_lis3dh.LIS3DH_I2C(i2c, int1=int1)
accel.range = adafruit_lis3dh.RANGE_4_G
accel.set_tap(1, 100)

while True:
    x, y, z = accel.acceleration
    print(x, y, z)
    time.sleep(0.1)
    if accel.tapped:
        print("Tapped!")

```

For more information, check out the [LIS3DH guide \(https://adafru.it/AJo\)](https://adafru.it/AJo)!

3 Watt LED

You can control the colors and brightness of a 3 Watt RGB LED using CircuitPython.

This example uses PWM to light up the LED with a rainbow swirl.

```

# SPDX-FileCopyrightText: 2019, 2023 Kattni Rembor for Adafruit Industries
#
# SPDX-License-Identifier: MIT

"""Simple rainbow swirl example for 3W LED"""
import time
import pwmio
import board
import digitalio

enable = digitalio.DigitalInOut(board.D10)
enable.direction = digitalio.Direction.OUTPUT
enable.value = True

def colorwheel(pos):
    if pos < 0 or pos > 255:
        return 0, 0, 0
    if pos < 85:
        return int(255 - pos * 3), int(pos * 3), 0
    if pos < 170:
        pos -= 85
        return 0, int(255 - pos * 3), int(pos * 3)
    pos -= 170
    return int(pos * 3), 0, int(255 - pos * 3)

red = pwmio.PWMOut(board.D11, duty_cycle=0, frequency=20000)
green = pwmio.PWMOut(board.D12, duty_cycle=0, frequency=20000)
blue = pwmio.PWMOut(board.D13, duty_cycle=0, frequency=20000)

while True:
    for i in range(255):
        r, g, b = colorwheel(i)
        red.duty_cycle = int(r * 65536 / 256)
        green.duty_cycle = int(g * 65536 / 256)
        blue.duty_cycle = int(b * 65536 / 256)
        time.sleep(0.05)

```

For more information, check out [PWM in the CircuitPython Essentials Guide](https://adafru.it/CTk) (<https://adafru.it/CTk>)!

Switch

You can easily use the mechanical switch on the Switch pin with CircuitPython.

The following example prints to the [serial console](https://adafru.it/Bec) (<https://adafru.it/Bec>) when the switch is closed.

```
# SPDX-FileCopyrightText: 2019 Kattni Rembor for Adafruit Industries
#
# SPDX-License-Identifier: MIT

"""Simple example to print when switch is pressed"""
import time
import digitalio
import board

switch = digitalio.DigitalInOut(board.D9)
switch.switch_to_input(pull=digitalio.Pull.UP)

while True:
    if not switch.value:
        print("Switch pressed!")
    time.sleep(0.1)
```

For more information check out [Digital In & Out in the CircuitPython Essentials Guide](https://adafru.it/Beo) (<https://adafru.it/Beo>)!

Downloads

Files

- [Fritzing object in the Adafruit Fritzing Library](https://adafru.it/DCN) (<https://adafru.it/DCN>)
- [EagleCAD files on GitHub](https://adafru.it/DCO) (<https://adafru.it/DCO>)
- [3D CAD files on GitHub](https://adafru.it/DHw) (<https://adafru.it/DHw>)

Schematics and Fab Print

