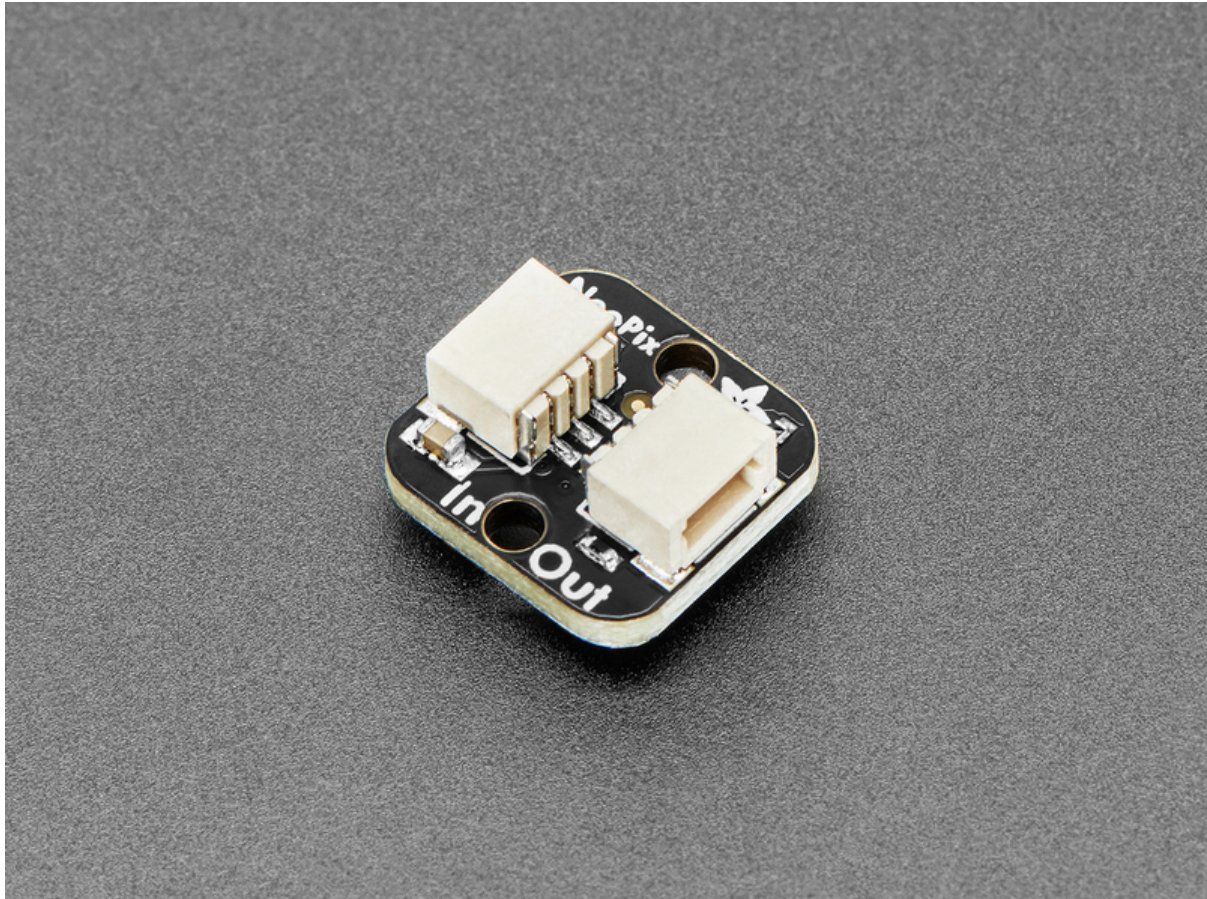




# Adafruit NeoPixel Breakout

Created by Liz Clark



<https://learn.adafruit.com/adafruit-neopixel-breakout>

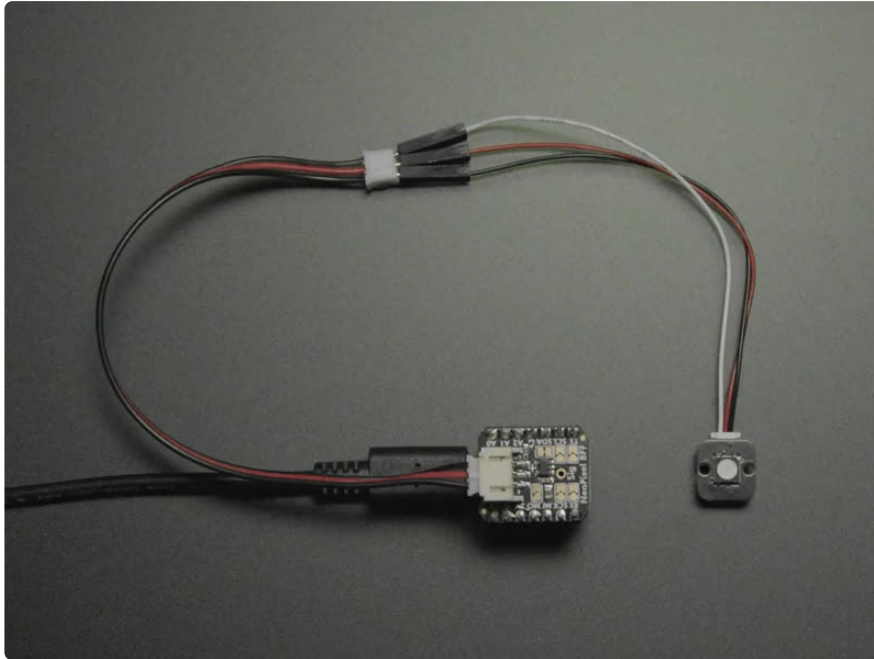
Last updated on 2024-06-17 01:14:45 PM EDT

# Table of Contents

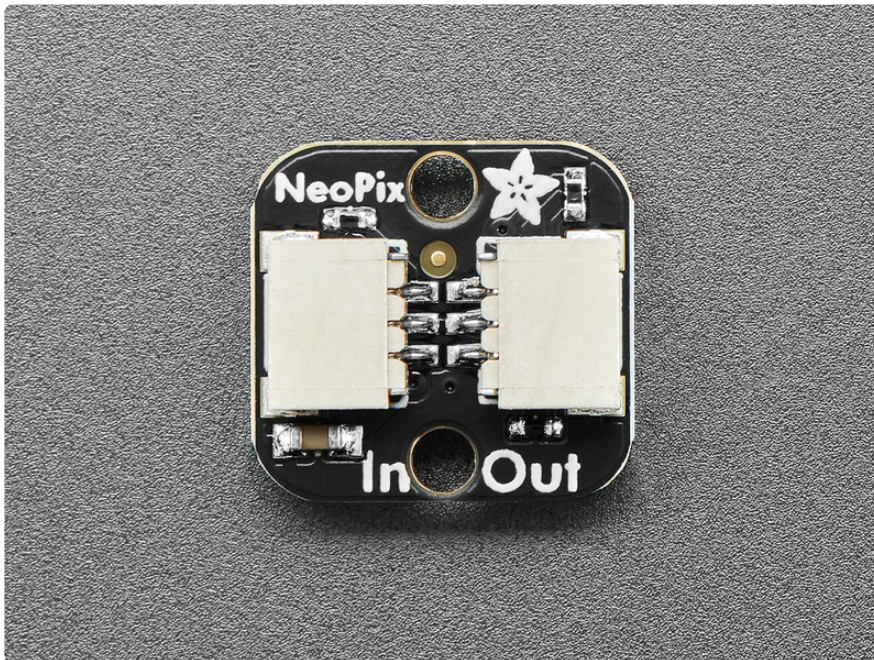
<a href="#">Overview</a>	3
<a href="#">Pinouts</a>	4
<ul style="list-style-type: none"><li>• <a href="#">NeoPixel</a></li><li>• <a href="#">JST SH Input</a></li><li>• <a href="#">JST SH Output</a></li></ul>	
<a href="#">CircuitPython and Python</a>	5
<ul style="list-style-type: none"><li>• <a href="#">CircuitPython Microcontroller Wiring</a></li><li>• <a href="#">Python Computer Wiring</a></li><li>• <a href="#">Python Installation of NeoPixel Library</a></li><li>• <a href="#">CircuitPython Usage</a></li><li>• <a href="#">Python Usage</a></li><li>• <a href="#">Example Code</a></li></ul>	
<a href="#">Python Docs</a>	10
<a href="#">Arduino</a>	10
<ul style="list-style-type: none"><li>• <a href="#">Wiring</a></li><li>• <a href="#">Library Installation</a></li><li>• <a href="#">Example Code</a></li></ul>	
<a href="#">Arduino Docs</a>	13
<a href="#">Downloads</a>	13
<ul style="list-style-type: none"><li>• <a href="#">Files</a></li><li>• <a href="#">Schematic and Fab Print</a></li></ul>	

---

## Overview

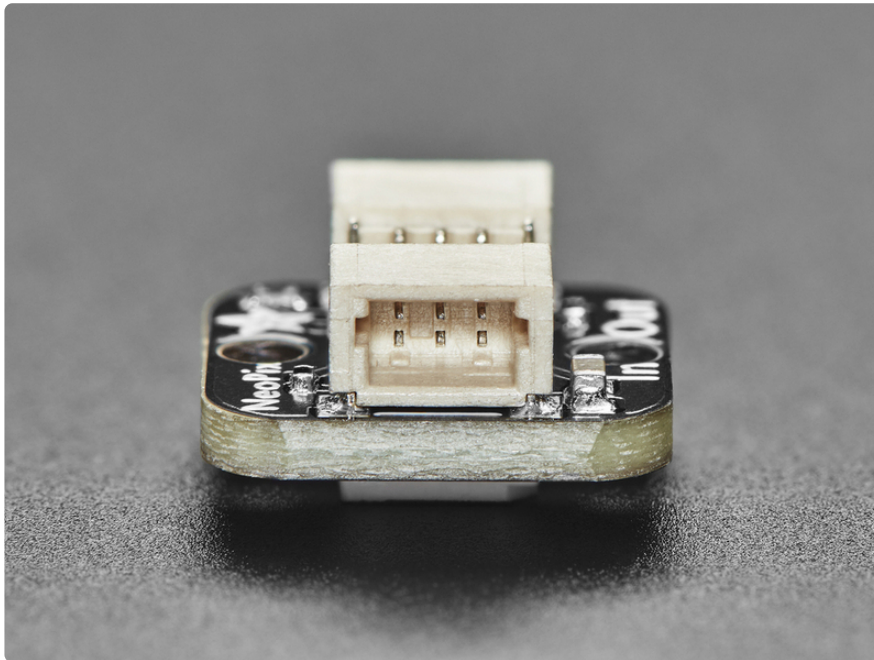


This little breakout makes it easy to add a single NeoPixel to a project, with mounting holes and a pluggable connector. The PCB is less than 0.5"x0.5" and comes with two [3-pin JST SH 1mm pitch connectors](http://adafru.it/5755) (<http://adafru.it/5755>) for input and output. On the opposite side, a 5050 (5mm square) classic RGB NeoPixel that can be powered and controlled with 3.3V or 5V power.



The two mounting holes are M2 size and have clearance for the screw head, or could be used for sewing onto a fabric backing or attaching to a small model. Of course this design will never be as affordable as a pre-made strand with dozens of LEDs, but

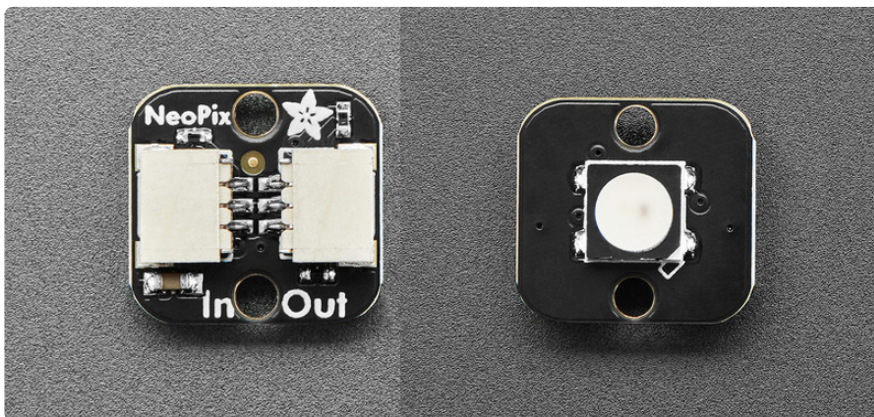
there's probably some time it would be useful - especially since it can be easily swapped out.



Comes as a single NeoPixel, no cables included so check out our [JST SH compatible cables in plug](http://adafru.it/5755) (<http://adafru.it/5755>) or [socket](http://adafru.it/5765) (<http://adafru.it/5765>).

---

## Pinouts



## NeoPixel

On the back of the breakout is the star of the show: the NeoPixel. It is a 5050 (5mm square) classic RGB NeoPixel that can be powered and controlled with 3.3V or 5V power and logic.

## JST SH Input

- **JST SH** (<http://adafru.it/5755>) - On the left side of the breakout, above the **In** label on the board silk, is a 1mm pitch JST port for use with [3-pin JST SH cables](http://adafru.it/5765) (<http://adafru.it/5765>). It has connections for:
  - **GND** - common ground for power and data. It is the black wire on the JST SH cable.
  - **VIN** - power input to the NeoPixel. Provide 3.3V or 5V power. Give it the same power as the logic level of your microcontroller - e.g. for a 5V microcontroller like Arduino, use 5V. It is the red wire on the JST SH cable.
  - **In** - data input for the NeoPixel. It is the white wire on the JST SH cable.

## JST SH Output

- **JST SH** (<http://adafru.it/5755>) - On the right side of the breakout, above the **Out** label on the board silk, is a 1mm pitch JST port for use with [3-pin JST SH cables](http://adafru.it/5765) (<http://adafru.it/5765>). It has connections for:
  - **GND** - common ground for power and data. It is the black wire on the JST SH cable.
  - **VIN** - power input to the NeoPixel. Provide 3.3V or 5V power. power input to the NeoPixel. Provide 3.3V or 5V power. Give it the same power as the logic level of your microcontroller - e.g. for a 5V microcontroller like Arduino, use 5V. It is the red wire on the JST SH cable.
  - **Out** - data output for the NeoPixel. It can be used to daisy chain multiple NeoPixels together by carrying the data input signal. It is the white wire on the JST SH cable.

---

## CircuitPython and Python

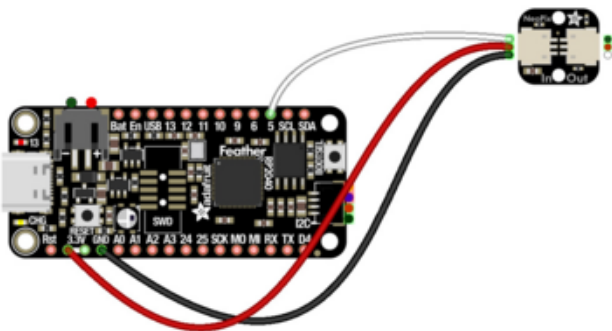
It's easy to use the **NeoPixel Breakout** with CircuitPython and the [Adafruit\\_CircuitPython\\_NeoPixel](https://adafru.it/yew) (<https://adafru.it/yew>) module. This module allows you to easily write Python code for NeoPixels.

You can use this driver with any CircuitPython microcontroller board or with a computer that has GPIO and Python [thanks to Adafruit\\_Blinka, our CircuitPython-for-Python compatibility library](https://adafru.it/BSN) (<https://adafru.it/BSN>). You should note though that many single board computers (SBCs) don't have NeoPixel support due to the precision timing required to send data.

Many single board computers don't have NeoPixel support due to the precision timing required to send data

## CircuitPython Microcontroller Wiring

Here is how you'll wire the breakout to a Feather RP2040 with a JST SH cable:

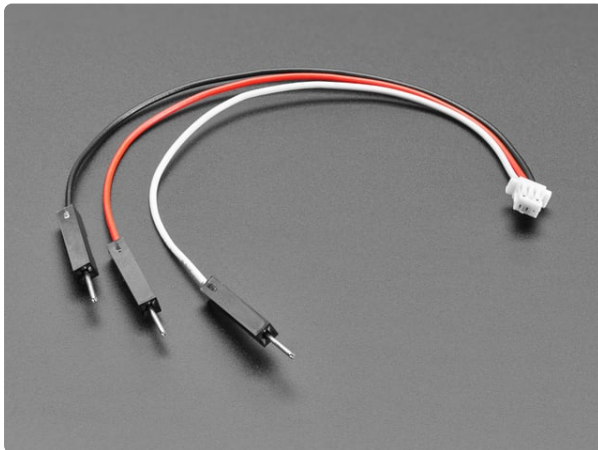


fritzing

Feather 3V to NeoPixel JST SH VIN (red wire)

Feather GND to NeoPixel JST SH GND (black wire)

Feather pin 5 to NeoPixel JST SH In (white wire)



[JST SH Compatible 1mm Pitch 3 Pin to Premium Male Headers Cable](#)

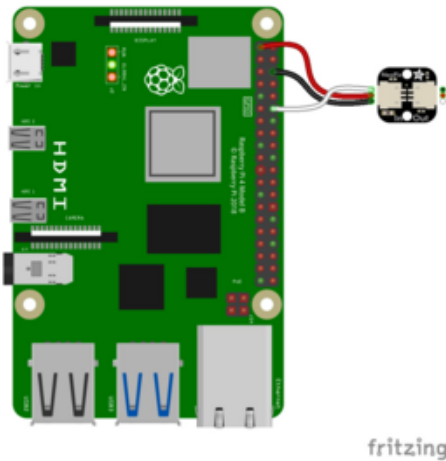
If you're fancying to connect to the debug port on a Pico WH, or the two ports on a <https://www.adafruit.com/product/5755>

## Python Computer Wiring

Since there are dozens of Linux computers/boards you can use, we will show wiring for Raspberry Pi. For other platforms, [please visit the guide for CircuitPython on Linux to see whether your platform is supported \(https://adafru.it/BSN\)](#).

Here's the Raspberry Pi wired to the breakout with a JST SH cable:

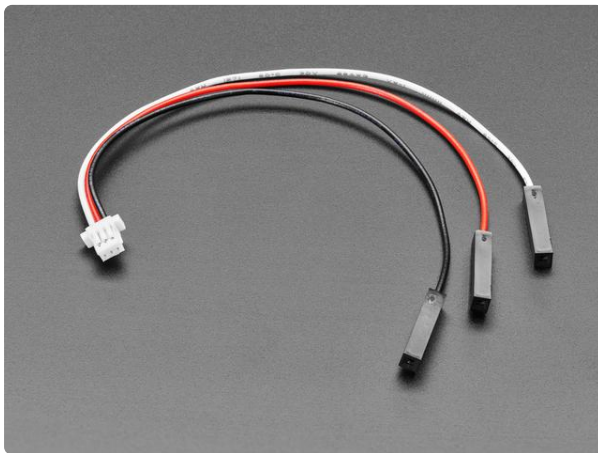
On the Raspberry Pi, NeoPixels must be connected to GPIO10, GPIO12, GPIO18 or GPIO21 to work!



Pi 3V to NeoPixel JST SH VIN (red wire)  
Pi GND to NeoPixel JST SH GND (black wire)

Pi GPIO18 to NeoPixel JST SH In (white wire)

On the Raspberry Pi, NeoPixels must be connected to GPIO10, GPIO12, GPIO18 or GPIO21 to work!



JST SH 1mm Pitch 3 Pin to Socket Headers Cable - 100mm long

If you're fancying to connect to the debug port on a Pico WH, or the two ports on a <https://www.adafruit.com/product/5765>

## Python Installation of NeoPixel Library

You'll need to install the **Adafruit\_Blinka** library that provides the CircuitPython support in Python. This may also require enabling I2C on your platform and verifying you are running Python 3. [Since each platform is a little different, and Linux changes often, please visit the CircuitPython on Linux guide to get your computer ready \(https://adafru.it/BSN\)](#)!

Once that's done, from your command line run the following command:

- `pip3 install adafruit-circuitpython-neopixel`

If your default Python is version 3 you may need to run 'pip' instead. Just make sure you aren't trying to use CircuitPython on Python 2.x, it isn't supported!

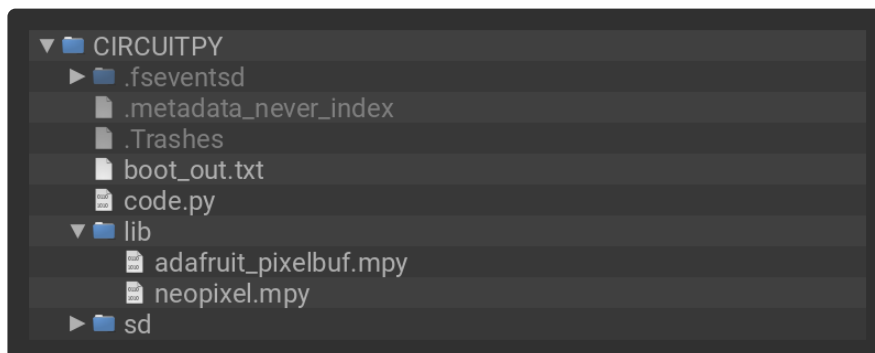
## CircuitPython Usage

To use with CircuitPython, you need to first install the **Adafruit\_CircuitPython\_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 files:

- **neopixel.mpy**
- **adafruit\_pixelbuf.mpy**



## Python Usage

Once you have the library **pip3** installed on your computer, copy or download the following example to your computer, and run the following, replacing **code.py** with whatever you named the file:

```
python3 code.py
```

## Example Code

**If running CircuitPython:** Once everything is saved to the **CIRCUITPY** drive, [connect to the serial console \(https://adafru.it/Bec\)](https://adafru.it/Bec) to see the data printed out!

**If running Python:** The console output will appear wherever you are running Python.



```

# SPDX-FileCopyrightText: 2024 Liz Clark for Adafruit Industries
#
# SPDX-License-Identifier: MIT

import time
import board
from rainbowio import colorwheel
import neopixel

num_pixels = 1
# pylint: disable=simplifiable-condition
# check to see if its a raspberry pi
if "CE0" and "CE1" in dir(board): # pi only zone
    pixel_pin = board.D18
# otherwise assume a microcontroller
else:
    pixel_pin = board.D5

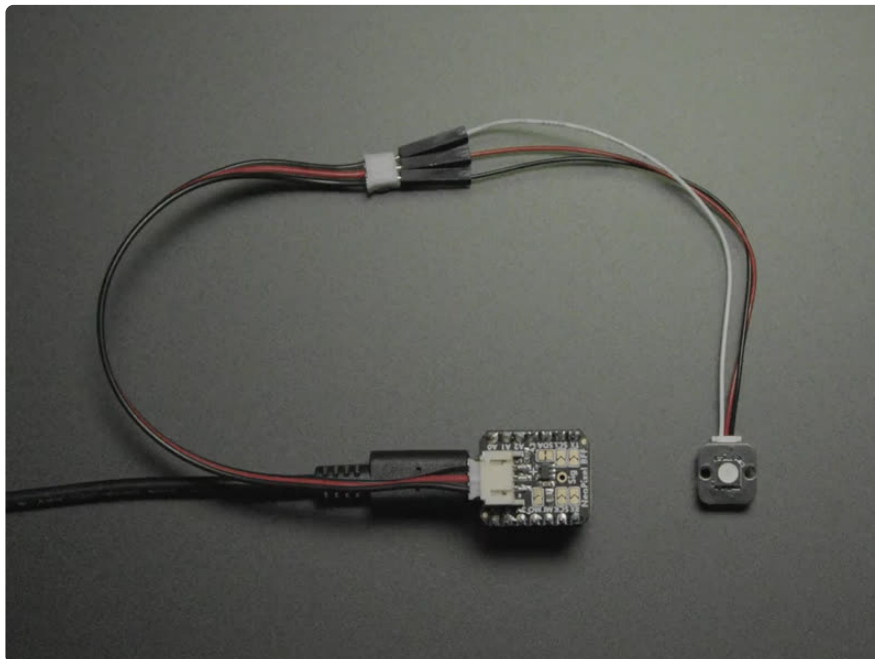
pixels = neopixel.NeoPixel(pixel_pin, num_pixels)

color_offset = 0

while True:
    for i in range(num_pixels):
        rc_index = (i * 256 // num_pixels) + color_offset
        pixels[i] = colorwheel(rc_index & 255)
    pixels.show()
    color_offset += 1
    time.sleep(0.01)

```

The code has a check to determine if you are running the code on a Raspberry Pi or not. If you are, the NeoPixel pin is set as **GPIO18**. Otherwise, the NeoPixel pin is set as **D5**. Once the loop starts, you'll see your NeoPixel cycle through the colors of the rainbow.



---

# Python Docs

[Python Docs \(https://adafru.it/18gA\)](https://adafru.it/18gA)

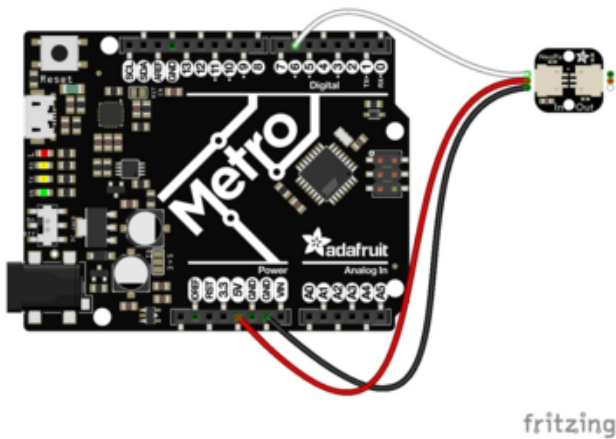
---

## Arduino

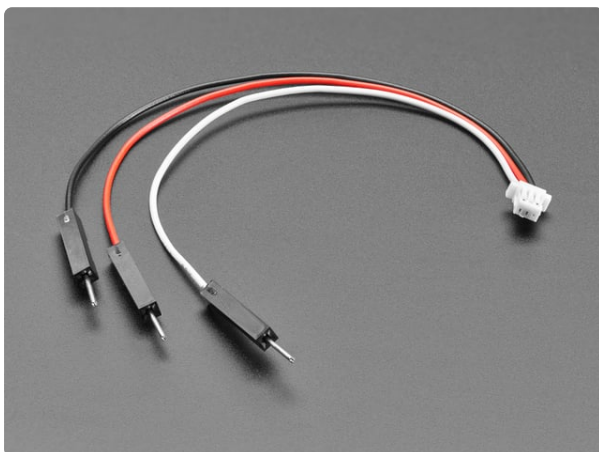
Using the NeoPixel Breakout with Arduino involves wiring up the breakout to your Arduino-compatible microcontroller, installing the [Adafruit\\_NeoPixel \(https://adafru.it/aZU\)](https://adafru.it/aZU) library and running the provided example code.

### Wiring

Here is how you'll wire the breakout to an Adafruit Metro with a JST SH cable:



- Metro 5V to NeoPixel JST SH VIN (red wire)
- Metro GND to NeoPixel JST SH GND (black wire)
- Metro pin 6 to NeoPixel JST SH In (white wire)

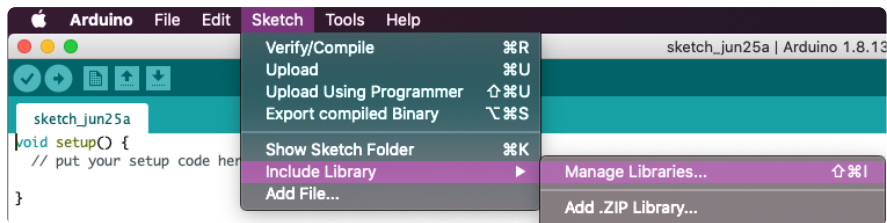


[JST SH Compatible 1mm Pitch 3 Pin to Premium Male Headers Cable](https://www.adafruit.com/product/5755)

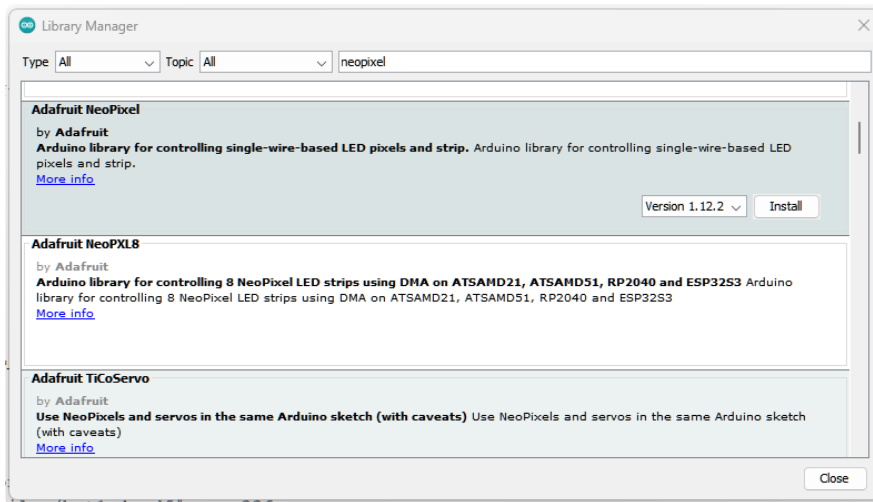
If you're fancying to connect to the debug port on a Pico WH, or the two ports on a <https://www.adafruit.com/product/5755>

### Library Installation

You can install the **Adafruit NeoPixel** library 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:



There are no additional dependencies for the Adafruit NeoPixel library.

## Example Code

```
// SPDX-FileCopyrightText: 2024 Limor Fried for Adafruit Industries
//
// SPDX-License-Identifier: MIT

#include <Adafruit_NeoPixel.h>
#ifdef __AVR__
  #include <avr/power.h> // Required for 16 MHz Adafruit Trinket
#endif

// Which pin on the Arduino is connected to the NeoPixels?
// On a Trinket or Gemma we suggest changing this to 1:
#define LED_PIN 6

// How many NeoPixels are attached to the Arduino?
#define LED_COUNT 1

// Declare our NeoPixel strip object:
Adafruit_NeoPixel strip(LED_COUNT, LED_PIN, NEO_GRB + NEO_KHZ800);
// Argument 1 = Number of pixels in NeoPixel strip
// Argument 2 = Arduino pin number (most are valid)
// Argument 3 = Pixel type flags

void setup() {
```

```

// These lines are specifically to support the Adafruit Trinket 5V 16 MHz.
// Any other board, you can remove this part (but no harm leaving it):
#if defined(__AVR_ATtiny85__) && (F_CPU == 16000000)
  clock_prescale_set(clock_div_1);
#endif
// END of Trinket-specific code.

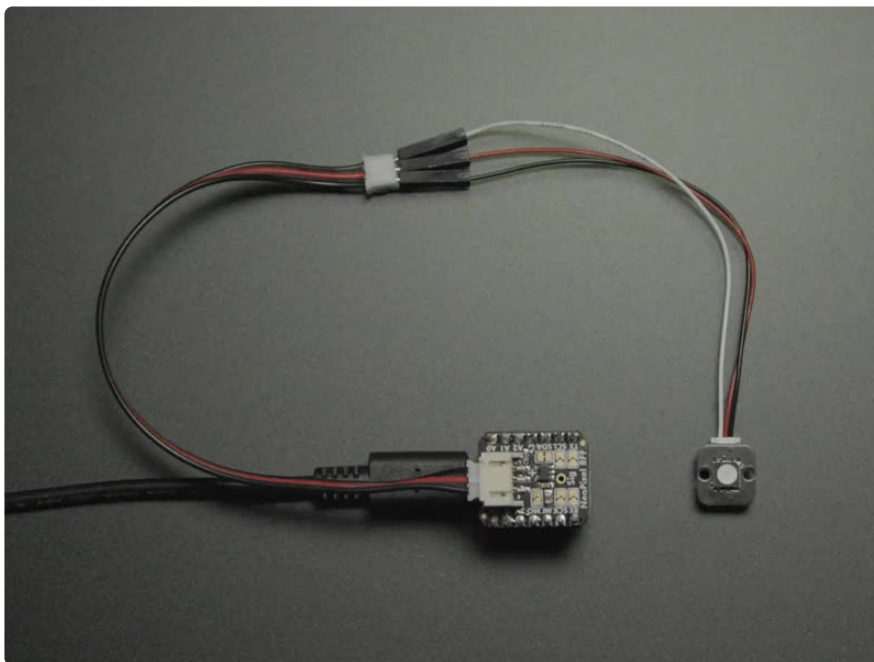
strip.begin();           // INITIALIZE NeoPixel strip object (REQUIRED)
strip.show();            // Turn OFF all pixels ASAP
strip.setBrightness(50); // Set BRIGHTNESS to about 1/5 (max = 255)
}
void loop() {

  rainbow(10);           // Flowing rainbow cycle along the whole strip
}

// Rainbow cycle along whole strip. Pass delay time (in ms) between frames.
void rainbow(int wait) {
  // Hue of first pixel runs 5 complete loops through the color wheel.
  // Color wheel has a range of 65536 but it's OK if we roll over, so
  // just count from 0 to 5*65536. Adding 256 to firstPixelHue each time
  // means we'll make 5*65536/256 = 1280 passes through this loop:
  for(long firstPixelHue = 0; firstPixelHue < 5*65536; firstPixelHue += 256) {
    // strip.rainbow() can take a single argument (first pixel hue) or
    // optionally a few extras: number of rainbow repetitions (default 1),
    // saturation and value (brightness) (both 0-255, similar to the
    // ColorHSV() function, default 255), and a true/false flag for whether
    // to apply gamma correction to provide 'truer' colors (default true).
    strip.rainbow(firstPixelHue);
    // Above line is equivalent to:
    // strip.rainbow(firstPixelHue, 1, 255, 255, true);
    strip.show(); // Update strip with new contents
    delay(wait); // Pause for a moment
  }
}
}

```

Upload the sketch to your board. You'll see your NeoPixel cycle through a rainbow swirl animation.



---

# Arduino Docs

[Arduino Docs \(https://adafru.it/Etk\)](https://adafru.it/Etk)

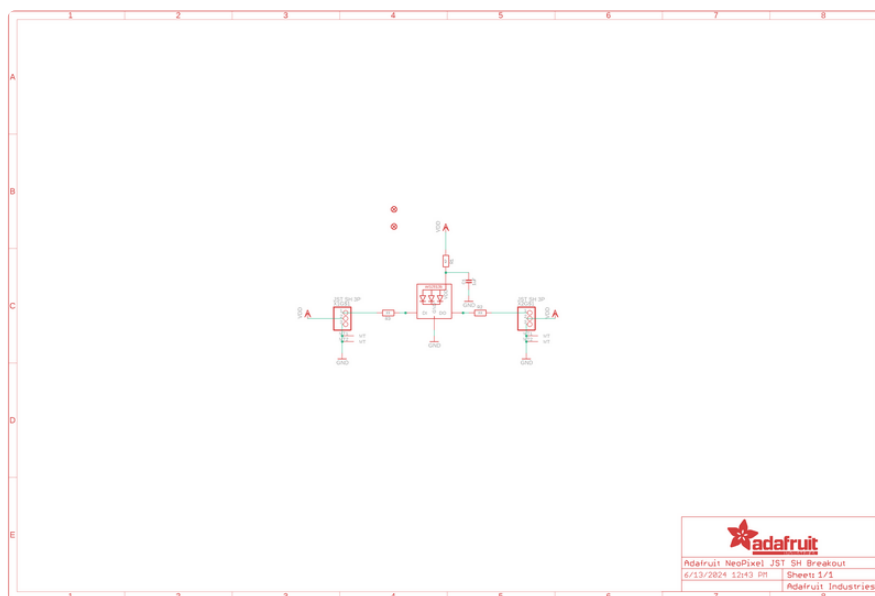
---

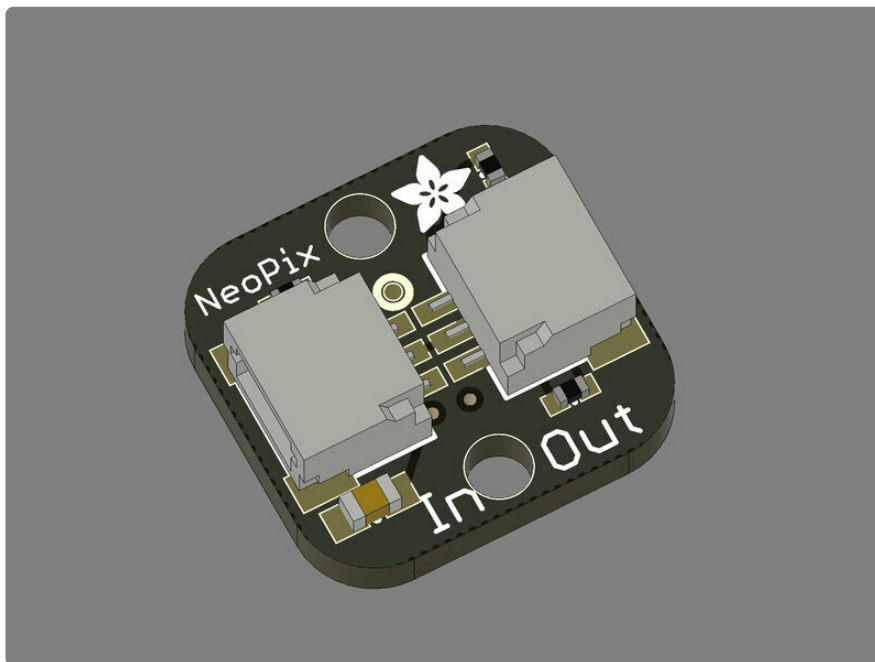
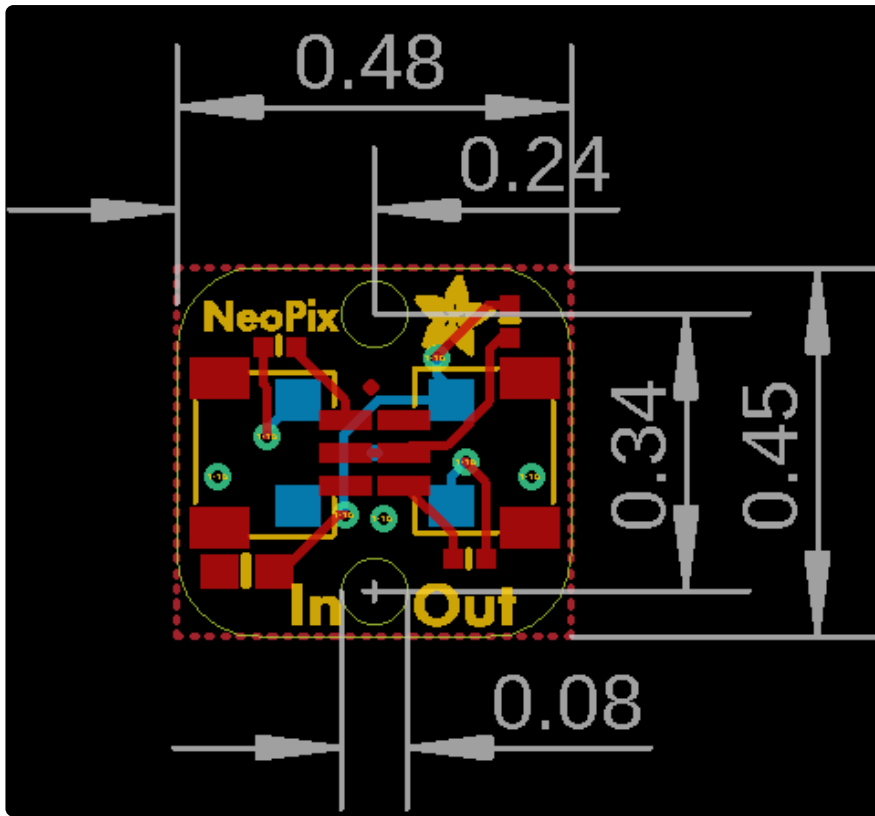
## Downloads

### Files

- [NeoPixel Datasheet \(https://adafru.it/uaR\)](https://adafru.it/uaR)
- [EagleCAD PCB Files on GitHub \(https://adafru.it/1a2N\)](https://adafru.it/1a2N)
- [3D Models on GitHub \(https://adafru.it/1a2V\)](https://adafru.it/1a2V)
- [Fritzing object in the Adafruit Fritzing Library \(https://adafru.it/1a2O\)](https://adafru.it/1a2O)

### Schematic and Fab Print





# Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

[Adafruit:](#)

[5975](#)