Adafruit NeoPixel Library Documentation

Release 1.0

Scott Shawcroft
Damien P. George

Apr 13, 2020

Contents

1 Dependencies **3**

2 Installing from PyPI **5**

3 Usage Example **7**

4 Contributing **9**

5 Documentation **11**

6 Table of Contents **13**

6.1 Simple test 13

6.2 `neopixel` - NeoPixel strip driver 16

7 Indices and tables **19**

Python Module Index **21**

Index **23**

Higher level NeoPixel driver that presents the strip as a sequence. This is a supercharged version of the original MicroPython driver. Its now more like a normal Python sequence and features slice support, `repr` and `len` support.

Colors are stored as tuples by default. However, you can also use int hex syntax to set values similar to colors on the web. For example, `0x100000` (`#100000` on the web) is equivalent to `(0x10, 0, 0)`.

Note: The int hex API represents the brightness of the white pixel when present by setting the RGB channels to identical values. For example, full white is 0xffffff but is actually (0, 0, 0, 0xff) in the tuple syntax. Setting a pixel value with an int will use the white pixel if the RGB channels are identical. For full, independent, control of each color component use the tuple syntax.

CHAPTER 1

Dependencies

This driver depends on:

- Adafruit CircuitPython

Please ensure all dependencies are available on the CircuitPython filesystem. This is easily achieved by downloading the Adafruit library and driver bundle.

CHAPTER 2

Installing from PyPI

On supported GNU/Linux systems like the Raspberry Pi, you can install the driver locally from PyPI. To install for current user:

```
pip3 install adafruit-circuitpython-neopixel
```

To install system-wide (this may be required in some cases):

```
sudo pip3 install adafruit-circuitpython-neopixel
```

To install in a virtual environment in your current project:

```
mkdir project-name && cd project-name
python3 -m venv .env
source .env/bin/activate
pip3 install adafruit-circuitpython-neopixel
```

CHAPTER 3

Usage Example

This example demonstrates the library with the single built-in NeoPixel on the Feather M0 Express and Metro M0 Express.

```
import board
import neopixel

pixels = neopixel.NeoPixel(board.NEOPIXEL, 1)
pixels[0] = (10, 0, 0)
```

This example demonstrates the library with the ten built-in NeoPixels on the Circuit Playground Express. It turns off `auto_write` so that all pixels are updated at once when the `show` method is called.

```
import board
import neopixel

pixels = neopixel.NeoPixel(board.NEOPIXEL, 10, auto_write=False)
pixels[0] = (10, 0, 0)
pixels[9] = (0, 10, 0)
pixels.show()
```

This example demonstrates using a single NeoPixel tied to a GPIO pin and with a `pixel_order` to specify the color channel order. Note that `bpp` does not need to be specified as it is computed from the supplied `pixel_order`.

```
import board
import neopixel

pixel = neopixel.NeoPixel(board.D0, 1, pixel_order=neopixel.RGBW)
pixel[0] = (30, 0, 20, 10)
```

CHAPTER 4

Contributing

Contributions are welcome! Please read our Code of Conduct before contributing to help this project stay welcoming.

CHAPTER 5

Documentation

For information on building library documentation, please check out this guide.

CHAPTER 6

Table of Contents

6.1 Simple test

Ensure your device works with this simple test.

Listing 1: examples/neopixel_simpletest.py

```
import time
import board
import neopixel


# On CircuitPlayground Express, and boards with built in status NeoPixel -> board.
↪NEOPIXEL
# Otherwise choose an open pin connected to the Data In of the NeoPixel strip, i.e.␣
↪board.D1
pixel_pin = board.NEOPIXEL

# On a Raspberry pi, use this instead, not all pins are supported
# pixel_pin = board.D18

# The number of NeoPixels
num_pixels = 10

# The order of the pixel colors - RGB or GRB. Some NeoPixels have red and green␣
↪reversed!
# For RGBW NeoPixels, simply change the ORDER to RGBW or GRBW.
ORDER = neopixel.GRB

pixels = neopixel.NeoPixel(
    pixel_pin, num_pixels, brightness=0.2, auto_write=False, pixel_order=ORDER
)


```

(continues on next page)

(continued from previous page)

```
def wheel(pos):
    # Input a value 0 to 255 to get a color value.
    # The colours are a transition r - g - b - back to r.
    if pos < 0 or pos > 255:
        r = g = b = 0
    elif pos < 85:
        r = int(pos * 3)
        g = int(255 - pos * 3)
        b = 0
    elif pos < 170:
        pos -= 85
        r = int(255 - pos * 3)
        g = 0
        b = int(pos * 3)
    else:
        pos -= 170
        r = 0
        g = int(pos * 3)
        b = int(255 - pos * 3)
    return (r, g, b) if ORDER in (neopixel.RGB, neopixel.GRB) else (r, g, b, 0)


def rainbow_cycle(wait):
    for j in range(255):
        for i in range(num_pixels):
            pixel_index = (i * 256 // num_pixels) + j
            pixels[i] = wheel(pixel_index & 255)
        pixels.show()
        time.sleep(wait)


while True:
    # Comment this line out if you have RGBW/GRBW NeoPixels
    pixels.fill((255, 0, 0))
    # Uncomment this line if you have RGBW/GRBW NeoPixels
    # pixels.fill((255, 0, 0, 0))
    pixels.show()
    time.sleep(1)

    # Comment this line out if you have RGBW/GRBW NeoPixels
    pixels.fill((0, 255, 0))
    # Uncomment this line if you have RGBW/GRBW NeoPixels
    # pixels.fill((0, 255, 0, 0))
    pixels.show()
    time.sleep(1)

    # Comment this line out if you have RGBW/GRBW NeoPixels
    pixels.fill((0, 0, 255))
    # Uncomment this line if you have RGBW/GRBW NeoPixels
    # pixels.fill((0, 0, 255, 0))
    pixels.show()
    time.sleep(1)

    rainbow_cycle(0.001)  # rainbow cycle with 1ms delay per step
```

Listing 2: examples/neopixel_rpi_simpletest.py

```
# Simple test for NeoPixels on Raspberry Pi
import time
import board
import neopixel


# Choose an open pin connected to the Data In of the NeoPixel strip, i.e. board.D18
# NeoPixels must be connected to D10, D12, D18 or D21 to work.
pixel_pin = board.D18

# The number of NeoPixels
num_pixels = 30

# The order of the pixel colors - RGB or GRB. Some NeoPixels have red and green␣
↪reversed!
# For RGBW NeoPixels, simply change the ORDER to RGBW or GRBW.
ORDER = neopixel.GRB

pixels = neopixel.NeoPixel(
    pixel_pin, num_pixels, brightness=0.2, auto_write=False, pixel_order=ORDER
)


def wheel(pos):
    # Input a value 0 to 255 to get a color value.
    # The colours are a transition r - g - b - back to r.
    if pos < 0 or pos > 255:
        r = g = b = 0
    elif pos < 85:
        r = int(pos * 3)
        g = int(255 - pos * 3)
        b = 0
    elif pos < 170:
        pos -= 85
        r = int(255 - pos * 3)
        g = 0
        b = int(pos * 3)
    else:
        pos -= 170
        r = 0
        g = int(pos * 3)
        b = int(255 - pos * 3)
    return (r, g, b) if ORDER in (neopixel.RGB, neopixel.GRB) else (r, g, b, 0)


def rainbow_cycle(wait):
    for j in range(255):
        for i in range(num_pixels):
            pixel_index = (i * 256 // num_pixels) + j
            pixels[i] = wheel(pixel_index & 255)
        pixels.show()
        time.sleep(wait)


while True:
```

(continues on next page)

(continued from previous page)

```
    # Comment this line out if you have RGBW/GRBW NeoPixels
    pixels.fill((255, 0, 0))
    # Uncomment this line if you have RGBW/GRBW NeoPixels
    # pixels.fill((255, 0, 0, 0))
    pixels.show()
    time.sleep(1)

    # Comment this line out if you have RGBW/GRBW NeoPixels
    pixels.fill((0, 255, 0))
    # Uncomment this line if you have RGBW/GRBW NeoPixels
    # pixels.fill((0, 255, 0, 0))
    pixels.show()
    time.sleep(1)

    # Comment this line out if you have RGBW/GRBW NeoPixels
    pixels.fill((0, 0, 255))
    # Uncomment this line if you have RGBW/GRBW NeoPixels
    # pixels.fill((0, 0, 255, 0))
    pixels.show()
    time.sleep(1)

    rainbow_cycle(0.001)  # rainbow cycle with 1ms delay per step
```

6.2 `neopixel` - NeoPixel strip driver

- Author(s): Damien P. George & Scott Shawcroft

`neopixel.GRB = (1, 0, 2)`
Green Red Blue

`neopixel.GRBW = (1, 0, 2, 3)`
Green Red Blue White

class `neopixel.NeoPixel`(*pin*, *n*, *, *bpp=3*, *brightness=1.0*, *auto_write=True*, *pixel_order=None*)
A sequence of neopixels.

Parameters

- **pin** (*Pin*) – The pin to output neopixel data on.
- **n** (*int*) – The number of neopixels in the chain
- **bpp** (*int*) – Bytes per pixel. 3 for RGB and 4 for RGBW pixels.
- **brightness** (*float*) – Brightness of the pixels between 0.0 and 1.0 where 1.0 is full brightness
- **auto_write** (*bool*) – True if the neopixels should immediately change when set. If False, *show* must be called explicitly.
- **pixel_order** (*tuple*) – Set the pixel color channel order. GRBW is set by default.

Example for Circuit Playground Express:

```
import neopixel
from board import *

RED = 0x100000 # (0x10, 0, 0) also works
```

(continues on next page)

(continued from previous page)

```
pixels = neopixel.NeoPixel(NEOPIXEL, 10)
for i in range(len(pixels)):
    pixels[i] = RED
```

Example for Circuit Playground Express setting every other pixel red using a slice:

```
import neopixel
from board import *
import time

RED = 0x100000 # (0x10, 0, 0) also works

# Using ``with`` ensures pixels are cleared after we're done.
with neopixel.NeoPixel(NEOPIXEL, 10) as pixels:
    pixels[::2] = [RED] * (len(pixels) // 2)
    time.sleep(2)
```

brightness
Overall brightness of the pixel

deinit()
Blank out the NeoPixels and release the pin.

fill(*color*)
Colors all pixels the given ***color***.

show()
Shows the new colors on the pixels themselves if they haven't already been autowritten.

The colors may or may not be showing after this function returns because it may be done asynchronously.

write()
Use `show` instead. It matches Micro:Bit and Arduino APIs.

neopixel.**RGB = (0, 1, 2)**
Red Green Blue

neopixel.**RGBW = (0, 1, 2, 3)**
Red Green Blue White

CHAPTER 7

Indices and tables

- genindex
- modindex
- search

Python Module Index

n

neopixel, 16

Index

B

`brightness` (*neopixel.NeoPixel attribute*), 17

D

`deinit()` (*neopixel.NeoPixel method*), 17

F

`fill()` (*neopixel.NeoPixel method*), 17

G

`GRB` (*in module neopixel*), 16
`GRBW` (*in module neopixel*), 16

N

`NeoPixel` (*class in neopixel*), 16
`neopixel` (*module*), 16

R

`RGB` (*in module neopixel*), 17
`RGBW` (*in module neopixel*), 17

S

`show()` (*neopixel.NeoPixel method*), 17

W

`write()` (*neopixel.NeoPixel method*), 17