
AdafruitRGBLED Library Documentation

Release 1.0

Brent Rubell

Jan 14, 2020

Contents

1	Dependencies	3
1.1	Installing from PyPI	3
2	Usage Example	5
3	Contributing	7
4	Documentation	9
5	Table of Contents	11
5.1	Simple test	11
5.2	adafruit_rgbled	12
5.2.1	Implementation Notes	12
6	Indices and tables	15
	Python Module Index	17
	Index	19

CircuitPython driver for RGB LEDs. Works with native microcontroller pins, [Adafruit Blinka](#), or the [PCA9685 PWM driver](#).

This driver depends on:

- [Adafruit CircuitPython](#)
- [SimpleIO Library](#)

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

1.1 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-rgbled
```

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

```
sudo pip3 install adafruit-circuitpython-rgbled
```

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-rgbled
```


CHAPTER 2

Usage Example

Initialize a common-cathode RGB LED with three PWM-capable pins.

```
import board
import adafruit_rgbled

# Pin the Red LED is connected to
RED_LED = board.D5

# Pin the Green LED is connected to
GREEN_LED = board.D6

# Pin the Blue LED is connected to
BLUE_LED = board.D7

# Create a RGB LED object
led = adafruit_rgbled.RGBLED(RED_LED, BLUE_LED, GREEN_LED)
```

Initialize a common-anode RGB LED with three PWM-capable pins

```
import board
import adafruit_rgbled

# Pin the Red LED is connected to
RED_LED = board.D5

# Pin the Green LED is connected to
GREEN_LED = board.D6

# Pin the Blue LED is connected to
BLUE_LED = board.D7

# Create a RGB LED object
led = adafruit_rgbled.RGBLED(RED_LED, BLUE_LED, GREEN_LED, invert_pwm = True)
```

Set the RGB LED's color to a RGB Tuple (Red, Green, Blue).

```
led.color = (255, 0, 0)
```

Set the RGB LED's color to a 24-bit integer (in hex syntax), 0x100000.

```
led.color = 0x100000
```

Setting a common-anode RGB LED using a ContextManager:

```
import board
import adafruit_rgbled
with adafruit_rgbled.RGBLED(board.D5, board.D6, board.D7, invert_pwm = True) as rgb_
    ↪led:
        rgb_led.color = (0, 255, 0)
```

CHAPTER 3

Contributing

Contributions are welcome! Please read our [Code of Conduct](#) before contributing to help this project stay welcoming.

CHAPTER 4

Documentation

For information on building library documentation, please check out [this guide](#).

5.1 Simple test

Ensure your device works with this simple test.

Listing 1: examples/rgbled_simpletest.py

```
1 import time
2 import board
3 import adafruit_rgbled
4
5 # Pin the Red LED is connected to
6 RED_LED = board.D5
7
8 # Pin the Green LED is connected to
9 GREEN_LED = board.D6
10
11 # Pin the Blue LED is connected to
12 BLUE_LED = board.D7
13
14 # Create the RGB LED object
15 led = adafruit_rgbled.RGBLED(RED_LED, GREEN_LED, BLUE_LED)
16
17 # Optionally, you can also create the RGB LED object with inverted PWM
18 # led = adafruit_rgbled.RGBLED(RED_LED, GREEN_LED, BLUE_LED, invert_pwm=True)
19
20 def wheel(pos):
21     # Input a value 0 to 255 to get a color value.
22     # The colours are a transition r - g - b - back to r.
23     if pos < 0 or pos > 255:
24         return 0, 0, 0
25     if pos < 85:
26         return int(255 - pos * 3), int(pos * 3), 0
27     if pos < 170:
```

(continues on next page)

```

28     pos -= 85
29     return 0, int(255 - pos * 3), int(pos * 3)
30 pos -= 170
31 return int(pos * 3), 0, int(255 - (pos * 3))
32
33 def rainbow_cycle(wait):
34     for i in range(255):
35         i = (i + 1) % 256
36         led.color = wheel(i)
37         time.sleep(wait)
38
39 while True:
40     # setting RGB LED color to RGB Tuples (R, G, B)
41     led.color = (255, 0, 0)
42     time.sleep(1)
43
44     led.color = (0, 255, 0)
45     time.sleep(1)
46
47     led.color = (0, 0, 255)
48     time.sleep(1)
49
50     # setting RGB LED color to 24-bit integer values
51     led.color = 0xFF0000
52     time.sleep(1)
53
54     led.color = 0x00FF00
55     time.sleep(1)
56
57     led.color = 0x0000FF
58     time.sleep(1)
59
60     # rainbow cycle the RGB LED
61     rainbow_cycle(0.1)

```

5.2 adafruit_rgbled

CircuitPython driver for RGB LEDs

- Author(s): Brent Rubell

5.2.1 Implementation Notes

Software and Dependencies:

- Adafruit CircuitPython firmware for the supported boards: <https://github.com/adafruit/circuitpython/releases>
- Adafruit's SimpleIO library: https://github.com/adafruit/Adafruit_CircuitPython_SimpleIO

class `adafruit_rgbled.RGBLED` (*red_pin*, *green_pin*, *blue_pin*, *invert_pwm=False*)
Creates a RGBLED object given three physical pins or PWMOut objects.

Parameters

- **red_pin** – The physical pin connected to a red LED anode.

- **green_pin** – The physical pin connected to a green LED anode.
- **blue_pin** – The physical pin connected to a blue LED anode.
- **invert_pwm** (*bool*) – False if the RGB LED is common cathode, true if the RGB LED is common anode.

Example for setting a RGB LED using a RGB Tuple (Red, Green, Blue):

```
import board
import adafruit_rgbled

RED_LED = board.D5
GREEN_LED = board.D6
BLUE_LED = board.D7

# Create a RGB LED object
led = adafruit_rgbled.RGBLED(RED_LED, BLUE_LED, GREEN_LED)
led.color = (255, 0, 0)
```

Example for setting a RGB LED using a 24-bit integer (hex syntax):

```
import board
import adafruit_rgbled

RED_LED = board.D5
GREEN_LED = board.D6
BLUE_LED = board.D7

# Create a RGB LED object
led = adafruit_rgbled.RGBLED(RED_LED, BLUE_LED, GREEN_LED)
led.color = 0x100000
```

Example for setting a RGB LED using a ContextManager:

```
import board
import adafruit_rgbled
with adafruit_rgbled.RGBLED(board.D5, board.D6, board.D7) as rgb_led:
    rgb_led.color = (255, 0, 0)
```

Example for setting a common-anode RGB LED using a ContextManager:

```
import board
import adafruit_rgbled
with adafruit_rgbled.RGBLED(board.D5, board.D6, board.D7, invert_pwm=True) as rgb_
↪led:
    rgb_led.color = (0, 255, 0)
```

color

Returns the RGB LED's current color.

deinit ()

Turn the LEDs off, deinit pwmout and release hardware resources.

CHAPTER 6

Indices and tables

- `genindex`
- `modindex`
- `search`

a

`adafruit_rgbled`, 12

A

adafruit_rgbled (*module*), 12

C

color (*adafruit_rgbled.RGBLED attribute*), 13

D

deinit () (*adafruit_rgbled.RGBLED method*), 13

R

RGBLED (*class in adafruit_rgbled*), 12