

---

# **Adafruit DotStar Library Documentation**

***Release 1.0***

**Scott Shawcroft, Limor Fried  
Damien P. George**

**Jul 19, 2021**



---

## Contents

---

<b>1</b>	<b>Dependencies</b>	<b>3</b>
<b>2</b>	<b>Installing from PyPI</b>	<b>5</b>
<b>3</b>	<b>Usage Example</b>	<b>7</b>
<b>4</b>	<b>Contributing</b>	<b>9</b>
<b>5</b>	<b>Documentation</b>	<b>11</b>
<b>6</b>	<b>Table of Contents</b>	<b>13</b>
6.1	Simple test . . . . .	13
6.2	adafruit_dotstar - DotStar strip driver (for CircuitPython 5.0+ with _pixelbuf) . . . . .	14
<b>7</b>	<b>Indices and tables</b>	<b>17</b>
	<b>Python Module Index</b>	<b>19</b>
	<b>Index</b>	<b>21</b>



Higher level DotStar driver that presents the strip as a sequence. It is the same api as the [NeoPixel library](#).

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)`.

If you send a tuple with 4 values, you can control the brightness value, which appears in DotStar but not NeoPixels. It should be a float. For example, `(0xFF,0,0, 1.0)` is the brightest red possible, `(1,0,0,0.01)` is the dimmest red possible.

---

**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 `(0xff, 0xff, 0xff)` in 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-dotstar
```

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

```
sudo pip3 install adafruit-circuitpython-dotstar
```

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-dotstar
```



## CHAPTER 3

---

### Usage Example

---

This example demonstrates the library with the single built-in DotStar on the [Trinket M0](#) and [Gemma M0](#).

```
import board
import adafruit_dotstar

pixels = adafruit_dotstar.DotStar(board.APA102_SCK, board.APA102_MOSI, 1)
pixels[0] = (10, 0, 0)
```



## 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](#).





### 6.1 Simple test

Ensure your device works with this simple test.

Listing 1: examples/dotstar\_simpletest.py

```
1  # SPDX-FileCopyrightText: 2021 ladyada for Adafruit Industries
2  # SPDX-License-Identifier: MIT
3
4  import time
5  import random
6  import board
7  import adafruit_dotstar as dotstar
8
9  # On-board DotStar for boards including Gemma, Trinket, and ItsyBitsy
10 dots = dotstar.DotStar(board.APA102_SCK, board.APA102_MOSI, 1, brightness=0.2)
11
12 # Using a DotStar Digital LED Strip with 30 LEDs connected to hardware SPI
13 # dots = dotstar.DotStar(board.SCK, board.MOSI, 30, brightness=0.2)
14
15 # Using a DotStar Digital LED Strip with 30 LEDs connected to digital pins
16 # dots = dotstar.DotStar(board.D6, board.D5, 30, brightness=0.2)
17
18
19 # HELPERS
20 # a random color 0 -> 192
21 def random_color():
22     return random.randrange(0, 7) * 32
23
24
25 # MAIN LOOP
26 n_dots = len(dots)
27 while True:
```

(continues on next page)

(continued from previous page)

```

28 # Fill each dot with a random color
29 for dot in range(n_dots):
30     dots[dot] = (random_color(), random_color(), random_color())
31
32 time.sleep(0.25)

```

## 6.2 adafruit\_dotstar - DotStar strip driver (for CircuitPython 5.0+ with \_pixelbuf)

- Author(s): Damien P. George, Limor Fried, Scott Shawcroft & Rose Hooper

adafruit\_dotstar.BGR = 'PBGR'  
Blue Green Red

adafruit\_dotstar.BRG = 'PBRG'  
Blue Red Green

**class** adafruit\_dotstar.DotStar(*clock, data, n, \*, brightness=1.0, auto\_write=True, pixel\_order='PBGR', baudrate=4000000*)

A sequence of dotstars.

### Parameters

- **clock** (*Pin*) – The pin to output dotstar clock on.
- **data** (*Pin*) – The pin to output dotstar data on.
- **n** (*int*) – The number of dotstars in the chain
- **brightness** (*float*) – Brightness of the pixels between 0.0 and 1.0
- **auto\_write** (*bool*) – True if the dotstars should immediately change when set. If False, *show* must be called explicitly.
- **pixel\_order** (*str*) – Set the pixel order on the strip - different strips implement this differently. If you send red, and it looks blue or green on the strip, modify this! It should be one of the values above.
- **baudrate** (*int*) – Desired clock rate if using hardware SPI (ignored if using 'soft' SPI). This is only a recommendation; the actual clock rate may be slightly different depending on what the system hardware can provide.

Example for Gemma M0:

```

import adafruit_dotstar
import time
from board import *

RED = 0x100000

with adafruit_dotstar.DotStar(APA102_SCK, APA102_MOSI, 1) as pixels:
    pixels[0] = RED
    time.sleep(2)

```

### show()

Shows the new colors on the dotstars 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.

**fill** (*color*)

Colors all dotstars the given *\*color\**.

**brightness**

Overall brightness of all dotstars (0 to 1.0)

**deinit** ()

Blank out the DotStars and release the resources.

**n**

The number of dotstars in the chain (read-only)

`adafruit_dotstar.GBR = 'PGBR'`

Green Blue Red

`adafruit_dotstar.GRB = 'PGRB'`

Green Red Blue

`adafruit_dotstar.RBG = 'PRBG'`

Red Blue Green

`adafruit_dotstar.RGB = 'PRGB'`

Red Green Blue



## CHAPTER 7

---

### Indices and tables

---

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



### a

adafruit\_dotstar, [14](#)





### A

`adafruit_dotstar` (*module*), 14

### B

BGR (*in module adafruit\_dotstar*), 14

BRG (*in module adafruit\_dotstar*), 14

`brightness` (*adafruit\_dotstar.DotStar attribute*), 15

### D

`deinit()` (*adafruit\_dotstar.DotStar method*), 15

`DotStar` (*class in adafruit\_dotstar*), 14

### F

`fill()` (*adafruit\_dotstar.DotStar method*), 14

### G

GBR (*in module adafruit\_dotstar*), 15

GRB (*in module adafruit\_dotstar*), 15

### N

`n` (*adafruit\_dotstar.DotStar attribute*), 15

### R

RBG (*in module adafruit\_dotstar*), 15

RGB (*in module adafruit\_dotstar*), 15

### S

`show()` (*adafruit\_dotstar.DotStar method*), 14