
Adafruit DotStar Library Documentation

Release 1.0

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

Mar 20, 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	adafruit_dotstar - DotStar strip driver	14
7	Indices and tables	15
	Python Module Index	17
	Index	19

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](#).

CHAPTER 6

Table of Contents

6.1 Simple test

Ensure your device works with this simple test.

Listing 1: examples/dotstar_simpletest.py

```
1 import time
2 import random
3 import board
4 import adafruit_dotstar as dotstar
5
6 # On-board DotStar for boards including Gemma, Trinket, and ItsyBitsy
7 dots = dotstar.DotStar(board.APA102_SCK, board.APA102_MOSI, 1, brightness=0.2)
8
9 # Using a DotStar Digital LED Strip with 30 LEDs connected to hardware SPI
10 # dots = dotstar.DotStar(board.SCK, board.MOSI, 30, brightness=0.2)
11
12 # Using a DotStar Digital LED Strip with 30 LEDs connected to digital pins
13 # dots = dotstar.DotStar(board.D6, board.D5, 30, brightness=0.2)
14
15
16 # HELPERS
17 # a random color 0 -> 224
18 def random_color():
19     return random.randrange(0, 7) * 32
20
21
22 # MAIN LOOP
23 n_dots = len(dots)
24 while True:
25     # Fill each dot with a random color
26     for dot in range(n_dots):
27         dots[dot] = (random_color(), random_color(), random_color())
```

(continues on next page)

(continued from previous page)

```

28
29     time.sleep(0.25)

```

6.2 adafruit_dotstar - DotStar strip driver

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

class adafruit_dotstar.DotStar(*clock*, *data*, *n*, *, *brightness*=1.0, *auto_write*=True, *pixel_order*=(2, 1, 0), *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** (*tuple*) – 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)

```

brightness

Overall brightness of the pixel

deinit()

Blank out the DotStars and release the resources.

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.

CHAPTER 7

Indices and tables

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

a

adafruit_dotstar, [14](#)

A

`adafruit_dotstar` (*module*), [14](#)

B

`brightness` (*adafruit_dotstar.DotStar attribute*), [14](#)

D

`deinit()` (*adafruit_dotstar.DotStar method*), [14](#)

`DotStar` (*class in adafruit_dotstar*), [14](#)

F

`fill()` (*adafruit_dotstar.DotStar method*), [14](#)

S

`show()` (*adafruit_dotstar.DotStar method*), [14](#)