Adafruit DotStar Library Documentation

Release 1.0

Scott Shawcroft, Limor Fried
Damien P. George

Oct 16, 2018

Contents

1 Dependencies **3**

2 Usage Example **5**

3 Contributing **7**

4 Building locally **9**
4.1 Sphinx documentation 9

5 Table of Contents **11**
5.1 Simple test 11
5.2 `adafruit_dotstar` - DotStar strip driver 12

6 Indices and tables **13**

Python Module Index **15**

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

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 3

Contributing

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

CHAPTER 4

Building locally

To build this library locally you'll need to install the circuitpython-build-tools package.

```
python3 -m venv .env
source .env/bin/activate
pip install circuitpython-build-tools
```

Once installed, make sure you are in the virtual environment:

```
source .env/bin/activate
```

Then run the build:

```
circuitpython-build-bundles --filename_prefix adafruit-circuitpython-dotstar --
↪library_location .
```

4.1 Sphinx documentation

Sphinx is used to build the documentation based on rST files and comments in the code. First, install dependencies (feel free to reuse the virtual environment from above):

```
python3 -m venv .env
source .env/bin/activate
pip install Sphinx sphinx-rtd-theme
```

Now, once you have the virtual environment activated:

```
cd docs
sphinx-build -E -W -b html . _build/html
```

This will output the documentation to `docs/_build/html`. Open the index.html in your browser to view them. It will also (due to -W) error out on any warning like Travis will. This is a good way to locally verify it will pass.

CHAPTER 5

Table of Contents

5.1 Simple test

Ensure your device works with this simple test.

Listing 1: examples/dotstar_simpletest.py

```
import time
import random
import board
import adafruit_dotstar as dotstar

# On-board DotStar for boards including Gemma, Trinket, and ItsyBitsy
dots = dotstar.DotStar(board.APA102_SCK, board.APA102_MOSI, 1, brightness=0.2)

# Using a DotStar Digital LED Strip with 30 LEDs connected to hardware SPI
# dots = dotstar.DotStar(board.SCK, board.MOSI, 30, brightness=0.2)

# Using a DotStar Digital LED Strip with 30 LEDs connected to digital pins
# dots = dotstar.DotStar(board.D6, board.D5, 30, brightness=0.2)


# HELPERS
# a random color 0 -> 224
def random_color():
    return random.randrange(0, 7) * 32


# MAIN LOOP
n_dots = len(dots)
while True:
    # Fill each dot with a random color
    for dot in range(n_dots):
        dots[dot] = (random_color(), random_color(), random_color())
```

(continues on next page)

(continued from previous page)

```
28
29      time.sleep(.25)
```

5.2 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)*)

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

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 6

Indices and tables

- genindex
- modindex
- search

Python Module Index

a

adafruit_dotstar, 12

Index

A

adafruit_dotstar (module), 12

B

brightness (adafruit_dotstar.DotStar attribute), 12

D

deinit() (adafruit_dotstar.DotStar method), 12
DotStar (class in adafruit_dotstar), 12

F

fill() (adafruit_dotstar.DotStar method), 12

S

show() (adafruit_dotstar.DotStar method), 12