AdafruitWS2801 Library Documentation

Release 1.0

Damien P. George, Limor Fried
Scott Shawcroft, Kevin J Walters

Dec 21, 2018

Contents

Higher level WS2801 driver that presents the LED string 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, `0x800000` (`#800000` on the web) is equivalent to `(0x80, 0, 0)`.

CHAPTER 1

Dependencies

This driver depends on:

- Adafruit CircuitPython
- Bus Device

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

Usage Example

This example demonstrates the library driving a strand of 25 RGB leds by a Gemma M0 using the hardware SPI capable outputs.

```python
import board
import adafruit_ws2801

leds = adafruit_ws2801.WS2801(board.D2, board.D0, 25)
leds.fill((0x80, 0, 0))
```

Contributing

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

Building locally

4.1 Zip release files

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-ws2801 --library_
→location .
```

4.2 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/ws2801_simpletest.py

```python
### Based on example from
### https://github.com/adafruit/Adafruit_CircuitPython_DotStar/tree/master/examples

import time
import random
import board
import adafruit_ws2801

### Example for a GEMMA M0 driving 50 12mm leds
oclock = board.D2
odata = board.D0
numleds = 50
bright = 1.0
leds = adafruit_ws2801.WS2801(oclock, odata, numleds, brightness=bright, auto_
↪write=False)

######################### HELPERS #############################

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

######################### MAIN LOOP #############################
n_leds = len(leds)
while True:
    #fill each led with a random color
    for idx in range(n_leds):
```

(continues on next page)

```
27          leds[idx] = (random_color(), random_color(), random_color())
28
29      # show all leds in led string
30      leds.show()
31
32      time.sleep(.25)
```

5.2 `adafruit_ws2801` - WS2801 LED pixel string driver

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

class adafruit_ws2801.**WS2801**(*clock*, *data*, *n*, ***, *brightness=1.0*, *auto_write=True*)
    A sequence of WS2801 controlled LEDs.

> **Parameters**
>
> - **clock** (`Pin`) – The pin to output dotstar clock on.
>
> - **data** (`Pin`) – The pin to output dotstar data on.
>
> - **n** (`int`) – The number of LEDs in the chain.
>
> - **brightness** (`float`) – The brightness between 0.0 and (default) 1.0.
>
> - **auto_write** (`bool`) – True if the dotstars should immediately change when set. If False, *show* must be called explicitly.

Example for Gemma M0:

```python
import adafruit_ws2801
import time
import board

darkred = 0x100000

with adafruit_ws2801.WS2801(board.D2, board.D0, 25, brightness=1.0) as pixels:
    pixels[0] = darkred
    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

Index

A

adafruit_ws2801 (module), 12

B

brightness (adafruit_ws2801.WS2801 attribute), 12

D

deinit() (adafruit_ws2801.WS2801 method), 12

F

fill() (adafruit_ws2801.WS2801 method), 12

S

show() (adafruit_ws2801.WS2801 method), 12

W

WS2801 (class in adafruit_ws2801), 12