Adafruitfancyled Library Documentation

Release 1.0

PaintYourDragon

Jan 29, 2018

Contents

1 Dependencies **3**

2 Usage Example **5**

3 API Reference **7**
3.1 adafruit_fancyled 7

4 Contributing **9**

5 Building locally **11**

Python Module Index **13**

TODO

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

TODO

CHAPTER 3

API Reference

3.1 adafruit_fancyled

TODO(description)

- Author(s): PaintYourDragon

class adafruit_fancyled.**CHSV**(*h*, *s=1.0*, *v=1.0*)
HSV (hue, saturation, value) color class.

class adafruit_fancyled.**CRGB**(*red*, *green=0.0*, *blue=0.0*)
RGB (red, green, blue) color class.

adafruit_fancyled.**clamp**(*val*, *lower*, *upper*)
Constrain value within a numeric range (inclusive).

adafruit_fancyled.**denormalize**(*val*, *inplace=False*)
Convert normalized (0.0 to 1.0) value to 8-bit (0 to 255) value ACCEPTS: float, 0.0 to 1.0 range or a list or tuple of floats. In list

> case, 'inplace' can be used to control whether the original list is modified (True) or a new list is generated and returned (False).

RETURNS: integer, 0 to 255 range, or list of integers (or None if inplace).

adafruit_fancyled.**expand_gradient**(*grad*, *length*)
Convert gradient palette into standard equal-interval palette. ACCEPTS: List or tuple of of 2-element lists/tuples containing position

> (0.0 to 1.0) and color (packed int, CRGB or CHSV). It's OK if the list/tuple elements are either lists OR tuples, but don't mix and match lists and tuples – use all one or the other.

RETURNS: CRGB list, can be used with palette_lookup() function.

adafruit_fancyled.**gamma_adjust**(*val*, *gamma_value=None*, *brightness=1.0*, *inplace=False*)

Provides gamma adjustment for single values, CRGB and CHSV types and lists of any of these.

ACCEPTS: One of three ways:

1. A single normalized level (0.0 to 1.0) and optional gamma-adjustment factor (float usu. > 1.0, default if unspecified is GFACTOR) and brightness (float 0.0 to 1.0, default is 1.0).
2. A single CRGB or CHSV type, optional single gamma factor OR a (R,G,B) gamma tuple (3 values usu. > 1.0), optional single brightness factor OR a (R,G,B) brightness tuple. The input tuples are RGB even when a CHSV color is passed.
3. A list or tuple of normalized levels, CRGB or CHSV types (and optional gamma and brightness levels or tuples applied to all).

In cases 2 and 3, if the input is a list (NOT a tuple!), the 'inplace' flag determines whether a new tuple/list is calculated and returned, or the existing value is modified in-place. By default this is 'False'. If you try to inplace-modify a tuple, there will be... trouble.

RETURNS: Corresponding to above cases:

1. Single normalized gamma-corrected brightness level (0.0 to 1.0).
2. A normalized gamma-corrected CRGB type (NOT CHSV!).
3. A list of gamma-corrected values or CRGB types (NOT CHSV!).

In cases 2 and 3, there is NO return value if 'inplace' is True – the original values are modified.

adafruit_fancyled.**mix**(*color1*, *color2*, *weight2=0.5*)

Blend between two colors using given ratio. ACCEPTS: two colors (each may be CRGB, CHSV or packed integer),

and weighting (0.0 to 1.0) of second color.

RETURNS: CRGB color in most cases, CHSV if both inputs are CHSV.

adafruit_fancyled.**normalize**(*val*, *inplace=False*)

Convert 8-bit (0 to 255) value to normalized (0.0 to 1.0) value. ACCEPTS: integer, 0 to 255 range (input is clamped) or a list or tuple

of integers. In list case, 'inplace' can be used to control whether the original list is modified (True) or a new list is generated and returned (False).

RETURNS: float, 0.0 to 1.0 range, or list of floats (or None if inplace).

adafruit_fancyled.**pack**(*val*)

'Pack' a CRGB or CHSV color into a 24-bit RGB integer. ACCEPTS: CRGB or CHSV color. RETURNS: 24-bit integer a la 0x00RRGGBB.

adafruit_fancyled.**palette_lookup**(*pal*, *pos*)

Fetch color from color palette, with interpolation. ACCEPTS: color palette (list of CRGB, CHSV and/or packed integers),

palette position (0.0 to 1.0, wraps around).

RETURNS: CRGB or CHSV instance, no gamma correction applied.

adafruit_fancyled.**unpack**(*val*)

'Unpack' a 24-bit color into a CRGB instance. ACCEPTS: 24-bit integer a la 0x00RRGGBB. RETURNS: CRGB color.

CHAPTER 4

Contributing

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

CHAPTER 5

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-fancyled --
↪library_location .
```

Python Module Index

a

adafruit_fancyled, 7

Index

A

adafruit_fancyled (module), 7

C

CHSV (class in adafruit_fancyled), 7
clamp() (in module adafruit_fancyled), 7
CRGB (class in adafruit_fancyled), 7

D

denormalize() (in module adafruit_fancyled), 7

E

expand_gradient() (in module adafruit_fancyled), 7

G

gamma_adjust() (in module adafruit_fancyled), 7

M

mix() (in module adafruit_fancyled), 8

N

normalize() (in module adafruit_fancyled), 8

P

pack() (in module adafruit_fancyled), 8
palette_lookup() (in module adafruit_fancyled), 8

U

unpack() (in module adafruit_fancyled), 8