CircuitPython equalizer Library Documentation

Release 1.0

Jose David M.

Jan 31, 2023

CONTENTS

1 Dependencies **3**

2 Usage Example **5**

3 Contributing **7**

4 Documentation **9**

5 Table of Contents **11**
- 5.1 Simple test 11
- 5.2 Equalizer Library 12
  - 5.2.1 `equalizer` 12

6 Indices and tables **15**

Python Module Index **17**

Index **19**

CircuitPython graphic equalizer

CHAPTER

ONE

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 or individual libraries can be installed using circup.

CHAPTER

TWO

USAGE EXAMPLE

Please see the `equalizer_simlpetest.py` example for initial reference

CHAPTER

THREE

CONTRIBUTING

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

CHAPTER

FOUR

DOCUMENTATION

For information on building library documentation, please check out this guide.

CHAPTER

FIVE

TABLE OF CONTENTS

5.1 Simple test

Ensure your device works with this simple test.

Listing 1: examples/equalizer_simpletest.py

```
# SPDX-FileCopyrightText: 2021 Jose David M.
#
# SPDX-License-Identifier: MIT
##############################
"""
This is a basic demonstration of a Equalizer widget.
"""

import time
import board
import displayio
from equalizer.equalizer import Equalizer

display = board.DISPLAY  # create the display on the PyPortal or Clue (for example)
# otherwise change this to setup the display
# for display chip driver and pinout you have (e.g. ILI9341)


# Create a Equalizer widget
my_equa = Equalizer(
    x=100,
    y=100,
    width=100,
    height=100,
    number_bars=5,
    bar_width=10,
    number_segments=6,
    segments_height=25,
    bar_best_fit=True,
    pad_x=2,
)

my_group = displayio.Group()
my_group.append(my_equa)
```

(continues on next page)

(continued from previous page)

```
display.show(my_group)  # add high level Group to the display

while True:
    # We updates the values for 5 bars. We update the values for
    # each bar
    my_equa.show_bars((10, 0, 10, 35, 85))
    time.sleep(0.5)

    my_equa.show_bars((70, 10, 0, 10, 35))
    time.sleep(0.5)

    my_equa.show_bars((0, 10, 35, 0, 90))
    time.sleep(0.5)

    my_equa.show_bars((35, 85, 10, 0, 10))
    time.sleep(0.5)

    my_equa.show_bars((10, 35, 85, 10, 0))
    time.sleep(0.5)

    my_equa.show_bars((0, 10, 35, 56, 90))
    time.sleep(0.5)
```

5.2 Equalizer Library

5.2.1 equalizer

equalizer.**float_rgb**(*mag*, *cmin*, *cmax*)

Return a tuple of floats between 0 and 1 for the red, green and blue amplitudes.

equalizer.**rectangle_helper**(*x0: int*, *y0: int*, *height: int*, *width: int*, *bitmap*, *color_index: int*, *palette*, *bitmaptool: bool = True*) → None

rectangle_helper function Draws a rectangle to the bitmap given using `bitmapstools.bitmap` or `vectorio.rectangle` functions

Parameters

- **x0** (*int*) – rectangle lower corner x position
- **y0** (*int*) – rectangle lower corner y position
- **width** (*int*) – rectangle upper corner x position
- **height** (*int*) – rectangle upper corner y position
- **color_index** (*int*) – palette color index to be used
- **palette** – palette object to be used to draw the rectangle
- **bitmap** – bitmap for the rectangle to be drawn
- **bitmaptool** (*bool*) – uses `draw_line()` to draw the rectangle. when `False` uses `Rectangle()`

Returns
None

Return type
None

equalizer.**rgb**(*mag*, *cmin*, *cmax*)

Return a tuple of integers to be used in AWT/Java plots.

CHAPTER

SIX

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

e
equalizer, 12

INDEX

E

equalizer
    module, 12

F

float_rgb() (*in module equalizer*), 12

M

module
    equalizer, 12

R

rectangle_helper() (*in module equalizer*), 12
rgb() (*in module equalizer*), 13