AdafruitDisplayIO$_{S}H1107Library Documentation

Release 1.0

Mark Roberts

Oct 25, 2021

Contents

1 Dependencies 3

2 Installing from PyPI 5

3 Usage Example 7

4 Documentation 9

5 Contributing 11

6 Documentation 13

7 Table of Contents 15
- 7.1 Simple test . . . 15
- 7.2 adafruit_displayio_sh1107 . . . 17
  - 7.2.1 Implementation Notes . . . 17

8 Indices and tables 19

Python Module Index 21

Index 23

DisplayIO driver for SH1107 monochrome displays. DisplayIO drivers enable terminal output.

CHAPTER 1

Dependencies

This driver depends on:

- Adafruit CircuitPython Version 6+ A new quirk in 6.0 for SH1107
- An SH1107 OLED display, eg. Adafruit FeatherWing 128 x 64 OLED

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-displayio-sh1107
```

To install system-wide (this may be required in some cases):

```
sudo pip3 install adafruit-circuitpython-displayio-sh1107
```

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-displayio-sh1107
```

CHAPTER 3

Usage Example

```
import board
import displayio
import terminalio
import bitmap_label as label # from adafruit_display_text
import adafruit_displayio_sh1107

displayio.release_displays()
#oled_reset = board.D9

# Use for I2C
i2c = board.I2C()
display_bus = displayio.I2CDisplay(i2c, device_address=0x3C)

# SH1107 is vertically oriented 64x128
WIDTH = 128
HEIGHT = 64
BORDER = 2

display = adafruit_displayio_sh1107.SH1107(display_bus, width=WIDTH, height=HEIGHT)

# Make the display context
splash = displayio.Group()
display.show(splash)

color_bitmap = displayio.Bitmap(WIDTH, HEIGHT, 1)
color_palette = displayio.Palette(1)
color_palette[0] = 0xFFFFFF  # White

bg_sprite = displayio.TileGrid(color_bitmap, pixel_shader=color_palette, x=0, y=0)
splash.append(bg_sprite)

# Draw a smaller inner rectangle in black
inner_bitmap = displayio.Bitmap(WIDTH - BORDER * 2, HEIGHT - BORDER * 2, 1)
inner_palette = displayio.Palette(1)
```

(continues on next page)

(continued from previous page)

```
inner_palette[0] = 0x000000  # Black
inner_sprite = displayio.TileGrid(inner_bitmap, pixel_shader=inner_palette, x=BORDER,␣
↪y=BORDER)
splash.append(inner_sprite)
```

CHAPTER 4

Documentation

API documentation for this library can be found on Read the Docs.

CHAPTER 5

Contributing

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

CHAPTER 6

Documentation

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

CHAPTER 7

Table of Contents

7.1 Simple test

Ensure your device works with this simple test.

Listing 1: examples/displayio_sh1107_simpletest.py

```
# SPDX-FileCopyrightText: 2021 ladyada for Adafruit Industries
#
# SPDX-License-Identifier: Unlicense
"""
Author: Mark Roberts (mdroberts1243) from Adafruit code
This test will initialize the display using displayio and draw a solid white
background, a smaller black rectangle, miscellaneous stuff and some white text.

"""


import board
import displayio
import terminalio

# can try import bitmap_label below for alternative
from adafruit_display_text import label
import adafruit_displayio_sh1107

displayio.release_displays()
# oled_reset = board.D9

# Use for I2C
i2c = board.I2C()
display_bus = displayio.I2CDisplay(i2c, device_address=0x3C)

# SH1107 is vertically oriented 64x128
```

(continues on next page)

(continued from previous page)

```
WIDTH = 128
HEIGHT = 64
BORDER = 2

display = adafruit_displayio_sh1107.SH1107(
    display_bus, width=WIDTH, height=HEIGHT, rotation=0
)

# Make the display context
splash = displayio.Group()
display.show(splash)

color_bitmap = displayio.Bitmap(WIDTH, HEIGHT, 1)
color_palette = displayio.Palette(1)
color_palette[0] = 0xFFFFFF  # White

bg_sprite = displayio.TileGrid(color_bitmap, pixel_shader=color_palette, x=0, y=0)
splash.append(bg_sprite)

# Draw a smaller inner rectangle in black
inner_bitmap = displayio.Bitmap(WIDTH - BORDER * 2, HEIGHT - BORDER * 2, 1)
inner_palette = displayio.Palette(1)
inner_palette[0] = 0x000000  # Black
inner_sprite = displayio.TileGrid(
    inner_bitmap, pixel_shader=inner_palette, x=BORDER, y=BORDER
)
splash.append(inner_sprite)

# Draw some white squares
sm_bitmap = displayio.Bitmap(8, 8, 1)
sm_square = displayio.TileGrid(sm_bitmap, pixel_shader=color_palette, x=58, y=17)
splash.append(sm_square)

med_bitmap = displayio.Bitmap(16, 16, 1)
med_square = displayio.TileGrid(med_bitmap, pixel_shader=color_palette, x=71, y=15)
splash.append(med_square)

lrg_bitmap = displayio.Bitmap(32, 32, 1)
lrg_square = displayio.TileGrid(lrg_bitmap, pixel_shader=color_palette, x=91, y=28)
splash.append(lrg_square)

# Draw some label text
text1 = "0123456789ABCDEF123456789AB"  # overly long to see where it clips
text_area = label.Label(terminalio.FONT, text=text1, color=0xFFFFFF, x=8, y=8)
splash.append(text_area)
text2 = "SH1107"
text_area2 = label.Label(
    terminalio.FONT, text=text2, scale=2, color=0xFFFFFF, x=9, y=44
)
splash.append(text_area2)

while True:
    pass
```

7.2 adafruit_displayio_sh1107

DisplayIO driver for SH1107 monochrome displays

- Author(s): Scott Shawcroft, Mark Roberts (mdroberts1243), James Carr

7.2.1 Implementation Notes

Hardware:

- Adafruit FeatherWing 128 x 64 OLED - SH1107 128x64 OLED

Software and Dependencies:

- Adafruit CircuitPython (version 6+) firmware for the supported boards: https://github.com/adafruit/circuitpython/releases

adafruit_displayio_sh1107.**DISPLAY_OFFSET_ADAFRUIT_FEATHERWING_OLED_4650**
The hardware display offset to use when configuring the SH1107 for the Adafruit Featherwing 128x64 OLED. This is the default if not passed in.

adafruit_displayio_sh1107.**DISPLAY_OFFSET_PIMORONI_MONO_OLED_PIM374**
The hardware display offset to use when configuring the SH1107 for the Pimoroni Mono 128x128 OLED

class adafruit_displayio_sh1107.**SH1107**(*bus*, *display_offset=<sphinx.ext.autodoc.importer._MockObject object>*, *rotation=0*, ***kwargs*)
SH1107 driver for use with DisplayIO

Parameters

- **bus** – The bus that the display is connected to.
- **width** (*int*) – The width of the display. Maximum of 128
- **height** (*int*) – The height of the display. Maximum of 128
- **rotation** (*int*) – The rotation of the display. 0, 90, 180 or 270.
- **display_offset** (*int*) – The display offset that the first column is wired to. This will be dependent on the OLED display and two displays with the same dimensions could have different offsets. This defaults to *DISPLAY_OFFSET_ADAFRUIT_FEATHERWING_OLED_4650*

is_awake
The power state of the display. (read-only)

`True` if the display is active, `False` if in sleep mode.

Type bool

sleep()
Put display into sleep mode. The display uses < 5uA in sleep mode

Sleep mode does the following:

1) Stops the oscillator and DC-DC circuits
2) Stops the OLED drive
3) Remembers display data and operation mode active prior to sleeping
4) The MP can access (update) the built-in display RAM

wake()
Wake display from sleep mode

CHAPTER 8

Indices and tables

- genindex
- modindex
- search

Python Module Index

a

adafruit_displayio_sh1107, 16

Index

A

adafruit_displayio_sh1107 (*module*), 16

D

DISPLAY_OFFSET_ADAFRUIT_FEATHERWING_OLED_4650
(*in module adafruit_displayio_sh1107*), 17
DISPLAY_OFFSET_PIMORONI_MONO_OLED_PIM374
(*in module adafruit_displayio_sh1107*), 17

I

is_awake (*adafruit_displayio_sh1107.SH1107 attribute*), 17

S

SH1107 (*class in adafruit_displayio_sh1107*), 17
sleep() (*adafruit_displayio_sh1107.SH1107 method*), 17

W

wake() (*adafruit_displayio_sh1107.SH1107 method*), 17