AdafruitDisplayIO$_{S}SD1306Library Documentatic

Release 1.0

Scott Shawcroft

Sep 24, 2021

Contents

DisplayIO driver for SSD1306 monochrome displays. DisplayIO drivers enable terminal output

For the framebuf based driver see Adafruit CircuitPython SSD1306.

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

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-ssd1306
```

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

```
sudo pip3 install adafruit-circuitpython-displayio-ssd1306
```

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-ssd1306
```

CHAPTER 3

Usage Example

```
import board
import displayio
import adafruit_displayio_ssd1306
import busio

displayio.release_displays()

# This pinout works on a Metro and may need to be altered for other boards.
spi = busio.SPI(board.SCL, board.SDA)
tft_cs = board.D9
tft_dc = board.D8
tft_reset = board.D7

display_bus = displayio.FourWire(spi, command=tft_dc, chip_select=tft_cs, reset=tft_
↪reset, baudrate=1000000)
display = adafruit_displayio_ssd1306.SSD1306(display_bus, width=128, height=64)
```

CHAPTER 4

Contributing

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

CHAPTER 5

Documentation

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

CHAPTER 6

Table of Contents

6.1 Simple test

Ensure your device works with this simple test.

Listing 1: examples/displayio_ssd1306_simpletest.py

```
# SPDX-FileCopyrightText: 2021 ladyada for Adafruit Industries
# SPDX-License-Identifier: MIT

"""
This test will initialize the display using displayio and draw a solid white
background, a smaller black rectangle, and some white text.
"""

import board
import displayio
import terminalio
from adafruit_display_text import label
import adafruit_displayio_ssd1306

displayio.release_displays()

oled_reset = board.D9

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

# Use for SPI
# spi = board.SPI()
# oled_cs = board.D5
# oled_dc = board.D6
# display_bus = displayio.FourWire(spi, command=oled_dc, chip_select=oled_cs,
```

(continues on next page)

(continued from previous page)

```
#                                   reset=oled_reset, baudrate=1000000)

WIDTH = 128
HEIGHT = 32  # Change to 64 if needed
BORDER = 5

display = adafruit_displayio_ssd1306.SSD1306(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
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 a label
text = "Hello World!"
text_area = label.Label(
    terminalio.FONT, text=text, color=0xFFFFFF, x=28, y=HEIGHT // 2 - 1
)
splash.append(text_area)

while True:
    pass
```

6.2 adafruit_displayio_ssd1306

DisplayIO driver for SSD1306 monochrome displays

- Author(s): Scott Shawcroft

6.2.1 Implementation Notes

Hardware:

- Monochrome 1.3” 128x64 OLED graphic display
- Monochrome 128x32 I2C OLED graphic display
- Monochrome 0.96” 128x64 OLED graphic display
- Monochrome 128x32 SPI OLED graphic display

- Adafruit FeatherWing OLED - 128x32 OLED

Software and Dependencies:

- Adafruit CircuitPython (version 5+) firmware for the supported boards: https://github.com/adafruit/circuitpython/releases

class adafruit_displayio_ssd1306.**SSD1306**(*bus*, ***kwargs*)

SSD1306 driver

Parameters

- **width** (*int*) – The width of the display
- **height** (*int*) – The height of the display
- **rotation** (*int*) – The rotation of the display in degrees. Default is 0. Must be one of (0, 90, 180, 270)

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.

Display uses < 10uA in sleep mode. Display remembers display data and operation mode active prior to sleeping. MP can access (update) the built-in display RAM.

wake()

Wake display from sleep mode

CHAPTER 7

Indices and tables

- genindex
- modindex
- search

Python Module Index

a

Index