AdafruitSHARPMemoryDisplay Library Documentation

Release 1.0

ladyada

Jun 07, 2021

Contents

A display control library for Sharp ‘memory’ displays

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.

1.1 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-sharpmemorydisplay
```

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

```
sudo pip3 install adafruit-circuitpython-sharpmemorydisplay
```

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-sharpmemorydisplay
```

CHAPTER 2

Usage Example

See the examples folder for a demonstration of drawing pixels, lines and text

CHAPTER 3

Contributing

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

CHAPTER 4

Documentation

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

CHAPTER 5

Table of Contents

5.1 Simple test

Ensure your device works with this simple test.

Listing 1: examples/sharpmemorydisplay_simpletest.py

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

import time
import board
import busio
import digitalio

import adafruit_sharpmemorydisplay

# Initialize SPI bus and control pins
spi = busio.SPI(board.SCK, MOSI=board.MOSI)
scs = digitalio.DigitalInOut(board.D6)  # inverted chip select

# pass in the display size, width and height, as well
# display = adafruit_sharpmemorydisplay.SharpMemoryDisplay(spi, scs, 96, 96)
display = adafruit_sharpmemorydisplay.SharpMemoryDisplay(spi, scs, 144, 168)

print("Pixel test")

# Clear the display.  Always call show after changing pixels to make the display
# update visible!
display.fill(1)
display.show()

# Set a pixel in the origin 0,0 position.
display.pixel(0, 0, 0)
```

(continues on next page)

(continued from previous page)

```
# Set a pixel in the middle position.
display.pixel(display.width // 2, display.width // 2, 0)
# Set a pixel in the opposite corner position.
display.pixel(display.width - 1, display.height - 1, 0)
display.show()
time.sleep(2)

print("Lines test")
# we'll draw from corner to corner, lets define all the pair coordinates here
corners = (
    (0, 0),
    (0, display.height - 1),
    (display.width - 1, 0),
    (display.width - 1, display.height - 1),
)

display.fill(1)
for corner_from in corners:
    for corner_to in corners:
        display.line(corner_from[0], corner_from[1], corner_to[0], corner_to[1], 0)
display.show()
time.sleep(2)

print("Rectangle test")
display.fill(1)
w_delta = display.width / 10
h_delta = display.height / 10
for i in range(11):
    display.rect(0, 0, int(w_delta * i), int(h_delta * i), 0)
display.show()
time.sleep(2)

print("Text test")
display.fill(1)
display.text(" hello world!", 0, 0, 0)
display.text(" This is the", 0, 8, 0)
display.text(" CircuitPython", 0, 16, 0)
display.text("adafruit library", 0, 24, 0)
display.text(" for the SHARP", 0, 32, 0)
display.text(" Memory Display :) ", 0, 40, 0)
display.show()
```

5.2 adafruit_sharpmemorydisplay

A display control library for Sharp 'memory' displays

- Author(s): ladyada

5.2.1 Implementation Notes

Hardware:

- Adafruit SHARP Memory Display Breakout - 1.3 inch 144x168 Monochrome
- Adafruit SHARP Memory Display Breakout - 1.3 inch 96x96 Monochrome

Software and Dependencies:

- Adafruit CircuitPython firmware for the supported boards: https://github.com/adafruit/circuitpython/releases

class adafruit_sharpmemorydisplay.**SharpMemoryDisplay**(*spi*, *scs_pin*, *width*, *height*, *, *baudrate=2000000*)

A driver for sharp memory displays, you can use any size but the full display must be buffered in memory!

show()

write out the frame buffer via SPI, we use MSB SPI only so some bit-swapping is rquired. The display also uses inverted CS for some reason so we con't use bus_device

adafruit_sharpmemorydisplay.**reverse_bit**(*num*)

Turn an LSB byte to an MSB byte, and vice versa. Used for SPI as it is LSB for the SHARP, but 99% of SPI implementations are MSB only!

CHAPTER 6

Indices and tables

- genindex
- modindex
- search

Python Module Index

Index