AdafruitEPD Library Documentation

Release 1.0

Dean Miller

Dec 21, 2018

Contents

This library is for using CircuitPython with e-ink displays with built in SRAM.

CHAPTER 1

Dependencies

This driver depends on:

- Adafruit CircuitPython
- Bus Device

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

```
import digitalio
import busio
import board
from adafruit_epd.epd import Adafruit_EPD
from adafruit_epd.il0373 import Adafruit_IL0373

# create the spi device and pins we will need
spi = busio.SPI(board.SCK, MOSI=board.MOSI, MISO=board.MISO)
ecs = digitalio.DigitalInOut(board.D10)
dc = digitalio.DigitalInOut(board.D9)
srcs = digitalio.DigitalInOut(board.D8)
rst = digitalio.DigitalInOut(board.D7)
busy = digitalio.DigitalInOut(board.D6)

# give them all to our driver
display = Adafruit_IL0373(152, 152, rst, dc, busy, srcs, ecs, spi)

# clear the buffer
display.clear_buffer()

r_width = 5
r_pos = display.height

#draw some rectangles!
color = Adafruit_EPD.BLACK
while r_pos > display.height/2:
    if r_pos < display.height - 50:
        color = Adafruit_EPD.RED
    display.rect(display.width - r_pos, display.height - r_pos,
                 display.width - 2*(display.width - r_pos),
                 display.height - 2*(display.height - r_pos), color)
    r_pos = r_pos - r_width

display.display()
```

CHAPTER 3

Contributing

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

CHAPTER 4

Building locally

4.1 Zip release files

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-epd --library_
↪location .
```

4.2 Sphinx documentation

Sphinx is used to build the documentation based on rST files and comments in the code. First, install dependencies (feel free to reuse the virtual environment from above):

```
python3 -m venv .env
source .env/bin/activate
pip install Sphinx sphinx-rtd-theme
```

Now, once you have the virtual environment activated:

```
cd docs
sphinx-build -E -W -b html . _build/html
```

This will output the documentation to `docs/_build/html`. Open the index.html in your browser to view them. It will also (due to -W) error out on any warning like Travis will. This is a good way to locally verify it will pass.

CHAPTER 5

Table of Contents

5.1 Simple test

Ensure your device works with this simple test.

Listing 1: examples/epd_simpletest.py

```python
import digitalio
import busio
import board
from adafruit_epd.epd import Adafruit_EPD
from adafruit_epd.il0373 import Adafruit_IL0373

# create the spi device and pins we will need
spi = busio.SPI(board.SCK, MOSI=board.MOSI, MISO=board.MISO)
ecs = digitalio.DigitalInOut(board.D10)
dc = digitalio.DigitalInOut(board.D9)
srcs = digitalio.DigitalInOut(board.D8)
rst = digitalio.DigitalInOut(board.D7)
busy = digitalio.DigitalInOut(board.D6)

# give them all to our driver
display = Adafruit_IL0373(152, 152, rst, dc, busy, srcs, ecs, spi)

# clear the buffer
display.clear_buffer()

r_width = 5
r_pos = display.height

color = Adafruit_EPD.BLACK
while r_pos > display.height/2:
    if r_pos < display.height - 50:
        color = Adafruit_EPD.RED
```

(continues on next page)

(continued from previous page)

```
    display.rect(display.width - r_pos, display.height - r_pos,
                 display.width - 2*(display.width - r_pos),
                 display.height - 2*(display.height - r_pos), color)
    r_pos = r_pos - r_width

display.display()
```

5.2 `adafruit_epd.epd` - Adafruit EPD - ePaper display driver

CircuitPython driver for Adafruit ePaper display breakouts * Author(s): Dean Miller

class adafruit_epd.epd.**Adafruit_EPD**(*width*, *height*, *rst_pin*, *dc_pin*, *busy_pin*, *srcs_pin*, *cs_pin*, *spi*)

Base class for EPD displays

begin(*reset=True*)
Begin display and reset if desired.

command(*cmd*, *data=None*, *end=True*)
Send command byte to display.

data(*dat*)
Send data to display.

draw_pixel(*x*, *y*, *color*)
This should be overridden in the subclass

fill(*color*)
fill the screen with the passed color

fill_rect(*x*, *y*, *width*, *height*, *color*)
fill a rectangle with the passed color

hline(*x*, *y*, *width*, *color*)
draw a horizontal line

pixel(*x*, *y*, *color=None*)
draw a pixel

rect(*x*, *y*, *width*, *height*, *color*)
draw a rectangle

vline(*x*, *y*, *height*, *color*)
draw a vertical line

CHAPTER 6

Indices and tables

- genindex
- modindex
- search

Python Module Index

Index