
AdafruitL0373 Library Documentation

Release 1.0

Scott Shawcroft

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	Device Specific Examples	16
7.3	adafruit_il10373	23
7.3.1	Implementation Notes	23
8	Indices and tables	25
	Python Module Index	27
	Index	29

CircuitPython `displayio` driver for IL0373-based ePaper 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](#).

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-10373
```

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

```
sudo pip3 install adafruit-circuitpython-10373
```

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-10373
```


CHAPTER 3

Usage Example

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](#).

7.1 Simple test

Ensure your device works with this simple test.

Listing 1: examples/il0373_simpletest.py

```
1 # SPDX-FileCopyrightText: 2021 ladyada for Adafruit Industries
2 # SPDX-License-Identifier: MIT
3
4 """Simple test script for 2.13" 212x104 tri-color featherwing.
5
6 Supported products:
7 * Adafruit 2.13" Tri-Color FeatherWing
8 * https://www.adafruit.com/product/4128
9 """
10
11 import time
12 import board
13 import displayio
14 import adafruit_il0373
15
16 displayio.release_displays()
17
18 epd_cs = board.D9
19 epd_dc = board.D10
20
21 display_bus = displayio.FourWire(
22     board.SPI(), command=epd_dc, chip_select=epd_cs, baudrate=1000000
23 )
24 time.sleep(1)
25
26 display = adafruit_il0373.IL0373(
27     display_bus, width=212, height=104, rotation=90, highlight_color=0xFF0000
```

(continues on next page)

(continued from previous page)

```

28 )
29
30 g = displayio.Group()
31
32 with open("/display-ruler.bmp", "rb") as f:
33     pic = displayio.OnDiskBitmap(f)
34     # CircuitPython 6 & 7 compatible
35     t = displayio.TileGrid(
36         pic, pixel_shader=getattr(pic, "pixel_shader", displayio.ColorConverter())
37     )
38     # CircuitPython 7 compatible only
39     # t = displayio.TileGrid(pic, pixel_shader=pic.pixel_shader)
40     g.append(t)
41
42     display.show(g)
43
44     display.refresh()
45
46     print("refreshed")
47
48     time.sleep(120)

```

7.2 Device Specific Examples

Listing 2: examples/il0373_1.54_color.py

```

1  # SPDX-FileCopyrightText: 2021 ladyada for Adafruit Industries
2  # SPDX-License-Identifier: MIT
3
4  """Simple test script for 1.54" 152x152 tri-color display.
5
6  Supported products:
7  * Adafruit 1.54" Tri-Color Display Breakout
8  * https://www.adafruit.com/product/3625
9  """
10
11 import time
12 import board
13 import displayio
14 import adafruit_il0373
15
16 displayio.release_displays()
17
18 # This pinout works on a Feather M4 and may need to be altered for other boards.
19 spi = board.SPI() # Uses SCK and MOSI
20 epd_cs = board.D9
21 epd_dc = board.D10
22 epd_reset = board.D5
23 epd_busy = board.D6
24
25 display_bus = displayio.FourWire(
26     spi, command=epd_dc, chip_select=epd_cs, reset=epd_reset, baudrate=1000000
27 )
28 time.sleep(1)

```

(continues on next page)

(continued from previous page)

```

29
30 display = adafruit_il0373.IL0373(
31     display_bus,
32     width=152,
33     height=152,
34     busy_pin=epd_busy,
35     highlight_color=0xFF0000,
36     rotation=180,
37 )
38
39 g = displayio.Group()
40
41 with open("/display-ruler.bmp", "rb") as f:
42     pic = displayio.OnDiskBitmap(f)
43     # CircuitPython 6 & 7 compatible
44     t = displayio.TileGrid(
45         pic, pixel_shader=getattr(pic, "pixel_shader", displayio.ColorConverter())
46     )
47     # CircuitPython 7 compatible only
48     # t = displayio.TileGrid(pic, pixel_shader=pic.pixel_shader)
49     g.append(t)
50
51     display.show(g)
52
53     display.refresh()
54
55     print("refreshed")
56
57     time.sleep(120)

```

Listing 3: examples/il0373_2.9_color.py

```

1  # SPDX-FileCopyrightText: 2021 ladyada for Adafruit Industries
2  # SPDX-License-Identifier: MIT
3
4  """Simple test script for Adafruit 2.9" 296x128 tri-color display
5  Supported products:
6  * Adafruit 2.9" Tri-Color Display Breakout
7  * https://www.adafruit.com/product/1028
8  """
9
10 import time
11 import board
12 import displayio
13 import adafruit_il0373
14
15 # Used to ensure the display is free in CircuitPython
16 displayio.release_displays()
17
18 # Define the pins needed for display use
19 # This pinout is for a Feather M4 and may be different for other boards
20 spi = board.SPI() # Uses SCK and MOSI
21 epd_cs = board.D9
22 epd_dc = board.D10
23 epd_reset = board.D5
24 epd_busy = board.D6

```

(continues on next page)

(continued from previous page)

```

25
26 # Create the displayio connection to the display pins
27 display_bus = displayio.FourWire(
28     spi, command=epd_dc, chip_select=epd_cs, reset=epd_reset, baudrate=1000000
29 )
30 time.sleep(1) # Wait a bit
31
32 # Create the display object - the third color is red (0xff0000)
33 display = adafruit_il0373.IL0373(
34     display_bus,
35     width=296,
36     height=128,
37     rotation=270,
38     busy_pin=epd_busy,
39     highlight_color=0xFF0000,
40 )
41
42 # Create a display group for our screen objects
43 g = displayio.Group()
44
45 # Display a ruler graphic from the root directory of the CIRCUITPY drive
46 with open("/display-ruler.bmp", "rb") as f:
47     pic = displayio.OnDiskBitmap(f)
48     # Create a Tilegrid with the bitmap and put in the displayio group
49     # CircuitPython 6 & 7 compatible
50     t = displayio.TileGrid(
51         pic, pixel_shader=getattr(pic, "pixel_shader", displayio.ColorConverter())
52     )
53     # CircuitPython 7 compatible only
54     # t = displayio.TileGrid(pic, pixel_shader=pic.pixel_shader)
55     g.append(t)
56
57     # Place the display group on the screen
58     display.show(g)
59
60     # Refresh the display to have it actually show the image
61     # NOTE: Do not refresh eInk displays sooner than 180 seconds
62     display.refresh()
63     print("refreshed")
64
65     time.sleep(180)

```

Listing 4: examples/il0373_2.9_grayscale.py

```

1 # SPDX-FileCopyrightText: 2021 ladyada for Adafruit Industries
2 # SPDX-License-Identifier: MIT
3
4 """Simple test script for 2.9" 296x128 grayscale display.
5
6 Supported products:
7 * Adafruit 2.9" Grayscale
8 * https://www.adafruit.com/product/4777
9 """
10
11 import time
12 import busio

```

(continues on next page)

(continued from previous page)

```

13 import board
14 import displayio
15 import adafruit_il0373
16
17 displayio.release_displays()
18
19 # This pinout works on a Feather M4 and may need to be altered for other boards.
20 spi = busio.SPI(board.SCK, board.MOSI) # Uses SCK and MOSI
21 epd_cs = board.D9
22 epd_dc = board.D10
23
24 display_bus = displayio.FourWire(
25     spi, command=epd_dc, chip_select=epd_cs, baudrate=1000000
26 )
27 time.sleep(1)
28
29 display = adafruit_il0373.IL0373(
30     display_bus,
31     width=296,
32     height=128,
33     rotation=270,
34     black_bits_inverted=False,
35     color_bits_inverted=False,
36     grayscale=True,
37     refresh_time=1,
38 )
39
40 g = displayio.Group()
41
42 with open("/display-ruler.bmp", "rb") as f:
43     pic = displayio.OnDiskBitmap(f)
44     # CircuitPython 6 & 7 compatible
45     t = displayio.TileGrid(
46         pic, pixel_shader=getattr(pic, "pixel_shader", displayio.ColorConverter())
47     )
48     # CircuitPython 7 compatible only
49     # t = displayio.TileGrid(pic, pixel_shader=pic.pixel_shader)
50     g.append(t)
51
52     display.show(g)
53
54     display.refresh()
55
56     print("refreshed")
57
58     time.sleep(120)

```

Listing 5: examples/il0373_2.13_color.py

```

1 # SPDX-FileCopyrightText: 2021 ladyada for Adafruit Industries
2 # SPDX-License-Identifier: MIT
3
4 """Simple test script for Adafruit 2.13" 212x104 tri-color display
5 Supported products:
6 * Adafruit 2.13" Tri-Color Display Breakout
7 * https://www.adafruit.com/product/4086 (breakout) or

```

(continues on next page)

(continued from previous page)

```
8     * https://www.adafruit.com/product/4128 (FeatherWing)
9     """
10
11 import time
12 import board
13 import displayio
14 import adafruit_il0373
15
16 # Used to ensure the display is free in CircuitPython
17 displayio.release_displays()
18
19 # Define the pins needed for display use
20 # This pinout is for a Feather M4 and may be different for other boards
21 spi = board.SPI() # Uses SCK and MOSI
22 epd_cs = board.D9
23 epd_dc = board.D10
24 epd_reset = board.D5
25 epd_busy = board.D6
26
27 # Create the displayio connection to the display pins
28 display_bus = displayio.FourWire(
29     spi, command=epd_dc, chip_select=epd_cs, reset=epd_reset, baudrate=1000000
30 )
31 time.sleep(1) # Wait a bit
32
33 # Create the display object - the third color is red (0xff0000)
34 display = adafruit_il0373.IL0373(
35     display_bus,
36     width=212,
37     height=104,
38     rotation=90,
39     busy_pin=epd_busy,
40     highlight_color=0xFF0000,
41 )
42
43 # Create a display group for our screen objects
44 g = displayio.Group()
45
46 # Display a ruler graphic from the root directory of the CIRCUITPY drive
47 with open("/display-ruler.bmp", "rb") as f:
48     pic = displayio.OnDiskBitmap(f)
49     # Create a Tilegrid with the bitmap and put in the displayio group
50     # CircuitPython 6 & 7 compatible
51     t = displayio.TileGrid(
52         pic, pixel_shader=getattr(pic, "pixel_shader", displayio.ColorConverter())
53     )
54     # CircuitPython 7 compatible only
55     # t = displayio.TileGrid(pic, pixel_shader=pic.pixel_shader)
56     g.append(t)
57
58 # Place the display group on the screen
59 display.show(g)
60
61 # Refresh the display to have it actually show the image
62 # NOTE: Do not refresh eInk displays sooner than 180 seconds
63 display.refresh()
64 print("refreshed")
```

(continues on next page)

(continued from previous page)

```

65
66     time.sleep(180)

```

Listing 6: examples/il0373_flexible_2.9_monochrome.py

```

1  # SPDX-FileCopyrightText: 2021 ladyada for Adafruit Industries
2  # SPDX-License-Identifier: MIT
3
4  """Simple test script for 2.9" 296x128 monochrome display.
5
6  Supported products:
7  * Adafruit Flexible 2.9" Monochrome
8  * https://www.adafruit.com/product/4262
9  """
10
11 import time
12 import board
13 import displayio
14 import adafruit_il0373
15
16 displayio.release_displays()
17
18 # This pinout works on a Feather M4 and may need to be altered for other boards.
19 spi = board.SPI() # Uses SCK and MOSI
20 epd_cs = board.D9
21 epd_dc = board.D10
22 epd_reset = board.D5
23 epd_busy = board.D6
24
25 display_bus = displayio.FourWire(
26     spi, command=epd_dc, chip_select=epd_cs, reset=epd_reset, baudrate=1000000
27 )
28 time.sleep(1)
29
30 display = adafruit_il0373.IL0373(
31     display_bus, width=296, height=128, rotation=90, busy_pin=epd_busy, swap_rams=True
32 )
33
34 g = displayio.Group()
35
36 with open("/display-ruler.bmp", "rb") as f:
37     pic = displayio.OnDiskBitmap(f)
38     # CircuitPython 6 & 7 compatible
39     t = displayio.TileGrid(
40         pic, pixel_shader=getattr(pic, "pixel_shader", displayio.ColorConverter())
41     )
42     # CircuitPython 7 compatible only
43     # t = displayio.TileGrid(pic, pixel_shader=pic.pixel_shader)
44     g.append(t)
45
46     display.show(g)
47
48     display.refresh()
49
50     time.sleep(120)

```

Listing 7: examples/il0373_2.13_monochrome.py

```
1 # SPDX-FileCopyrightText: 2021 ladyada for Adafruit Industries
2 # SPDX-License-Identifier: MIT
3
4 """Simple test script for 2.13" 212x104 monochrome display.
5
6 Supported products:
7 * Adafruit Flexible 2.13" Monochrome
8 * https://www.adafruit.com/product/4243
9 """
10
11 import time
12 import board
13 import displayio
14 import adafruit_il0373
15
16 displayio.release_displays()
17
18 # This pinout works on a Feather M4 and may need to be altered for other boards.
19 spi = board.SPI() # Uses SCK and MOSI
20 epd_cs = board.D9
21 epd_dc = board.D10
22 epd_reset = board.D5
23 epd_busy = board.D6
24
25 display_bus = displayio.FourWire(
26     spi, command=epd_dc, chip_select=epd_cs, reset=epd_reset, baudrate=1000000
27 )
28 time.sleep(1)
29
30 display = adafruit_il0373.IL0373(
31     display_bus, width=212, height=104, rotation=90, busy_pin=epd_busy, swap_rams=True
32 )
33
34 g = displayio.Group()
35
36 with open("/display-ruler.bmp", "rb") as f:
37     pic = displayio.OnDiskBitmap(f)
38     # CircuitPython 6 & 7 compatible
39     t = displayio.TileGrid(
40         pic, pixel_shader=getattr(pic, "pixel_shader", displayio.ColorConverter())
41     )
42     # CircuitPython 7 compatible only
43     # t = displayio.TileGrid(pic, pixel_shader=pic.pixel_shader)
44     g.append(t)
45
46 display.show(g)
47
48 display.refresh()
49
50 time.sleep(120)
```

7.3 adafruit_i10373

CircuitPython `displayio` driver for IL0373-based ePaper displays

- Author(s): Scott Shawcroft

7.3.1 Implementation Notes

Hardware:

- Adafruit 1.54" Tri-Color Display Breakout
- Adafruit 2.13" Tri-Color Display Breakout
- Adafruit Flexible 2.9" Black and White
- Adafruit Flexible 2.13" Black and White
- Adafruit 2.13" Tri-Color FeatherWing

Software and Dependencies:

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

class `adafruit_i10373.IL0373` (*bus*, *swap_rams=False*, ***kwargs*)
IL0373 driver

Parameters

- **bus** – The data bus the display is on
- **swap_rams** (*bool*) – Color and black rams/commands are swapped
- ****kwargs** – See below

Keyword Arguments

- *width* (*int*) – Display width
- *height* (*int*) – Display height
- *rotation* (*int*) – Display rotation
- *color_bits_inverted* (*bool*) – Invert color bit values
- *black_bits_inverted* (*bool*) – Invert black bit values

CHAPTER 8

Indices and tables

- `genindex`
- `modindex`
- `search`

a

[adafruit_il0373](#), [22](#)

A

adafruit_il0373 (*module*), 22

I

IL0373 (*class in adafruit_il0373*), 23