
AdafruitMatrixPortal Library Documentation

Release 1.0

Melissa LeBlanc-Williams

Jun 07, 2021

Contents

1	Dependencies	3
2	Usage Example	5
3	Contributing	7
4	Documentation	9
5	Table of Contents	11
5.1	Simple test	11
5.2	Other tests	12
5.3	adafruit_matrixportal.matrix	13
5.3.1	Implementation Notes	13
5.4	adafruit_matrixportal.graphics	14
5.4.1	Implementation Notes	14
5.5	adafruit_matrixportal.matrixportal	15
5.5.1	Implementation Notes	15
5.6	adafruit_matrixportal.network	17
5.6.1	Implementation Notes	17
6	Indices and tables	19
	Python Module Index	21
	Index	23

CircuitPython helper for Adafruit MatrixPortal M4, Adafruit RGB Matrix Shield + Metro M4 Airlift Lite, and Adafruit RGB Matrix FeatherWings

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

Usage Example

```
import time
import board
import terminalio
from adafruit_matrixportal.matrixportal import MatrixPortal

# --- Display setup ---
matrixportal = MatrixPortal(status_neopixel=board.NEOPIXEL, debug=True)

# Create a new label with the color and text selected
matrixportal.add_text(
    text_font=terminalio.FONT,
    text_position=(0, (matrixportal.graphics.display.height // 2) - 1),
    scrolling=True,
)

SCROLL_DELAY = 0.03

contents = [
    { 'text': 'THIS IS RED', 'color': '#cf2727'},
    { 'text': 'THIS IS BLUE', 'color': '#0846e4'},
]

while True:
    for content in contents:
        matrixportal.set_text(content['text'])

        # Set the text color
        matrixportal.set_text_color(content['color'])

        # Scroll it
        matrixportal.scroll_text(SCROLL_DELAY)
```


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](#).

5.1 Simple test

Ensure your device works with this simple test.

Listing 1: examples/matrixportal_simpletest.py

```
1 # SPDX-FileCopyrightText: 2020 Melissa LeBlanc-Williams, written for Adafruit_
   ↳ Industries
2 #
3 # SPDX-License-Identifier: Unlicense
4 """
5 This example checks the current Bitcoin price and displays it in the middle of the_
   ↳ screen
6 """
7 import time
8 import board
9 import terminalio
10 from adafruit_matrixportal.matrixportal import MatrixPortal
11
12 # You can display in 'GBP', 'EUR' or 'USD'
13 CURRENCY = "USD"
14 # Set up where we'll be fetching data from
15 DATA_SOURCE = "https://api.coindesk.com/v1/bpi/currentprice.json"
16 DATA_LOCATION = ["bpi", CURRENCY, "rate_float"]
17
18
19 def text_transform(val):
20     if CURRENCY == "USD":
21         return "$%d" % val
22     if CURRENCY == "EUR":
23         return "€%d" % val
24     if CURRENCY == "GBP":
25         return "£%d" % val
```

(continues on next page)

(continued from previous page)

```

26     return "%d" % val
27
28
29 # the current working directory (where this file is)
30 cwd = ("/" + __file__).rsplit("/", 1)[0]
31
32 matrixportal = MatrixPortal(
33     url=DATA_SOURCE,
34     json_path=DATA_LOCATION,
35     status_neopixel=board.NEOPIXEL,
36 )
37
38 matrixportal.add_text(
39     text_font=terminalio.FONT,
40     text_position=(16, 16),
41     text_color=0xFFFFFF,
42     text_transform=text_transform,
43 )
44 matrixportal.preload_font(b"$012345789") # preload numbers
45 matrixportal.preload_font((0x00A3, 0x20AC)) # preload gbp/euro symbol
46
47 while True:
48     try:
49         value = matrixportal.fetch()
50         print("Response is", value)
51     except (ValueError, RuntimeError) as e:
52         print("Some error occured, retrying! -", e)
53
54     time.sleep(3 * 60) # wait 3 minutes

```

5.2 Other tests

Scrolling Bitcoin Test

Listing 2: examples/matrixportal_scrolling_bitcoin.py

```

1  # SPDX-FileCopyrightText: 2020 Melissa LeBlanc-Williams, written for Adafruit_
   ↪ Industries
2  #
3  # SPDX-License-Identifier: Unlicense
4  """
5  This example checks the current Bitcoin price and scrolls it across the screen
6  """
7  import time
8  import board
9  import terminalio
10 from adafruit_matrixportal.matrixportal import MatrixPortal
11
12 # You can display in 'GBP', 'EUR' or 'USD'
13 CURRENCY = "USD"
14 # Set up where we'll be fetching data from
15 DATA_SOURCE = "https://api.coindesk.com/v1/bpi/currentprice.json"
16 DATA_LOCATION = ["bpi", CURRENCY, "rate_float"]
17

```

(continues on next page)

(continued from previous page)

```

18
19 def text_transform(val) :
20     if CURRENCY == "USD":
21         return "$%d" % val
22     if CURRENCY == "EUR":
23         return "€%d" % val
24     if CURRENCY == "GBP":
25         return "£%d" % val
26     return "%d" % val
27
28
29 # the current working directory (where this file is)
30 cwd = ("/" + __file__).rsplit("/", 1)[0]
31
32 matrixportal = MatrixPortal(
33     url=DATA_SOURCE,
34     json_path=DATA_LOCATION,
35     status_neopixel=board.NEOPIXEL,
36 )
37
38 matrixportal.add_text(
39     text_font=terminalio.FONT,
40     text_position=(16, 16),
41     text_color=0xFFFFFF,
42     text_transform=text_transform,
43     scrolling=True,
44 )
45 matrixportal.preload_font(b"$012345789") # preload numbers
46 matrixportal.preload_font((0x00A3, 0x20AC)) # preload gbp/euro symbol
47
48 last_check = None
49
50 while True:
51     if last_check is None or time.monotonic() > last_check + 180:
52         try:
53             value = matrixportal.fetch()
54             print("Response is", value)
55             last_check = time.monotonic()
56         except (ValueError, RuntimeError) as e:
57             print("Some error occured, retrying! -", e)
58     matrixportal.scroll()
59     time.sleep(0.03)

```

5.3 adafruit_matrixportal.matrix

Helper library for the MatrixPortal M4 or Adafruit RGB Matrix Shield + Metro M4 Airlift Lite.

- Author(s): Melissa LeBlanc-Williams

5.3.1 Implementation Notes

Hardware:

- Adafruit Matrix Portal

- [Adafruit Metro M4 Express AirLift](#)
- [Adafruit RGB Matrix Shield](#)
- [64x32 RGB LED Matrix](#)

Software and Dependencies:

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

```
class adafruit_matrixportal.matrix.Matrix(*, width=64, height=32, bit_depth=2,
                                          alt_addr_pins=None, color_order='RGB',
                                          serpentine=True, tile_rows=1, rotation=0)
```

Class representing the Adafruit RGB Matrix. This is used to automatically initialize the display.

Parameters

- **width** (*int*) – The width of the display in Pixels. Defaults to 64.
- **height** (*int*) – The height of the display in Pixels. Defaults to 32.
- **bit_depth** (*int*) – The number of bits per color channel. Defaults to 2.
- **alt_addr_pins** (*list*) – An alternate set of address pins to use. Defaults to None
- **color_order** (*string*) – A string containing the letter “R”, “G”, and “B” in the order you want. Defaults to “RGB”
- **width** – The total width of the display(s) in Pixels. Defaults to 64.
- **height** – The total height of the display(s) in Pixels. Defaults to 32.
- **Serpentine** (*bool*) – Used when panels are arranged in a serpentine pattern rather than a Z-pattern. Defaults to True.
- **tiles_rows** (*int*) – Used to indicate the number of rows the panels are arranged in. Defaults to 1.

5.4 `adafruit_matrixportal.graphics`

Helper library for the MatrixPortal M4 or Adafruit RGB Matrix Shield + Metro M4 Airlift Lite.

- Author(s): Melissa LeBlanc-Williams

5.4.1 Implementation Notes

Hardware:

- [Adafruit MatrixPortal M4](#)
- [Adafruit Metro M4 Express AirLift](#)
- [Adafruit RGB Matrix Shield](#)
- [64x32 RGB LED Matrix](#)

Software and Dependencies:

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

```
class adafruit_matrixportal.graphics.Graphics (**kwargs)
```

Graphics Helper Class for the MatrixPortal Library

Parameters

- **default_bg** – The path to your default background image file or a hex color. Defaults to 0x000000.
- **width** (*int*) – The total width of the display(s) in Pixels. Defaults to 64.
- **height** (*int*) – The total height of the display(s) in Pixels. Defaults to 32.
- **bit_depth** (*int*) – The number of bits per color channel. Defaults to 2.
- **alt_addr_pins** (*list*) – An alternate set of address pins to use. Defaults to None
- **color_order** (*string*) – A string containing the letter “R”, “G”, and “B” in the order you want. Defaults to “RGB”
- **Serpentine** (*bool*) – Used when panels are arranged in a serpentine pattern rather than a Z-pattern. Defaults to True.
- **tiles_rows** (*int*) – Used to indicate the number of rows the panels are arranged in. Defaults to 1.
- **debug** – Turn on debug print outs. Defaults to False.

5.5 adafruit_matrixportal.matrixportal

Helper library for the MatrixPortal M4 or Adafruit RGB Matrix Shield + Metro M4 Airlift Lite.

- Author(s): Melissa LeBlanc-Williams

5.5.1 Implementation Notes

Hardware:

- Adafruit MatrixPortal M4
- Adafruit Metro M4 Express AirLift
- Adafruit RGB Matrix Shield
- 64x32 RGB LED Matrix

Software and Dependencies:

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

```
class adafruit_matrixportal.matrixportal.MatrixPortal(*, url=None, headers=None, json_path=None,
    regexp_path=None, default_bg=0, status_neopixel=None, json_transform=None,
    esp=None, external_spi=None, bit_depth=2, alt_addr_pins=None, color_order='RGB',
    debug=False, width=64, height=32, serpentine=True, tile_rows=1, rotation=0)
```

Class representing the Adafruit RGB Matrix Portal.

Parameters

- **url** – The URL of your data source. Defaults to `None`.
- **headers** – The headers for authentication, typically used by Azure API's.
- **json_path** – The list of json traversal to get data out of. Can be list of lists for multiple data points. Defaults to `None` to not use json.
- **regexp_path** – The list of regexp strings to get data out (use a single regexp group). Can be list of regexps for multiple data points. Defaults to `None` to not use regexp.
- **default_bg** – The path to your default background image file or a hex color. Defaults to `0x000000`.
- **status_neopixel** – The pin for the status NeoPixel. Use `board.NEOPIXEL` for the on-board NeoPixel. Defaults to `None`, not the status LED
- **json_transform** – A function or a list of functions to call with the parsed JSON. Changes and additions are permitted for the `dict` object.
- **esp** – A passed ESP32 object, Can be used in cases where the ESP32 chip needs to be used before calling the `pyportal` class. Defaults to `None`.
- **external_spi** (*busio.SPI*) – A previously declared spi object. Defaults to `None`.
- **bit_depth** (*int*) – The number of bits per color channel. Defaults to 2.
- **alt_addr_pins** (*list*) – An alternate set of address pins to use. Defaults to `None`
- **color_order** (*string*) – A string containing the letter “R”, “G”, and “B” in the order you want. Defaults to “RGB”
- **debug** – Turn on debug print outs. Defaults to `False`.
- **width** (*int*) – The total width of the display(s) in Pixels. Defaults to 64.
- **height** (*int*) – The total height of the display(s) in Pixels. Defaults to 32.
- **Serpentine** (*bool*) – Used when panels are arranged in a serpentine pattern rather than a Z-pattern. Defaults to `True`.
- **tiles_rows** (*int*) – Used to indicate the number of rows the panels are arranged in. Defaults to 1.

`add_text` (*text_position=None*, *text_font=<fontio.BuiltinFont object>*, *text_color=8421504*, *text_wrap=False*, *text_maxlen=0*, *text_transform=None*, *text_scale=1*, *scrolling=False*, *line_spacing=1.25*, *text_anchor_point=(0, 0.5)*, *is_data=True*)

Add text labels with settings

Parameters

- **text_font** (*str*) – The path to your font file for your data text display.
- **text_position** – The position of your extracted text on the display in an (x, y) tuple. Can be a list of tuples for when there's a list of `json_paths`, for example.
- **text_color** – The color of the text, in `0xRRGGBB` format. Can be a list of colors for when there's multiple texts. Defaults to `None`.
- **text_wrap** – When non-zero, the maximum number of characters on each line before text is wrapped. (for long text data chunks). Defaults to 0, no wrapping.
- **text_maxlen** – The max length of the text. If non-zero, it will be truncated to this length. Defaults to 0.
- **text_transform** – A function that will be called on the text before display
- **text_scale** (*int*) – The factor to scale the default size of the text by

- **scrolling** (*bool*) – If true, text is placed offscreen and the scroll() function is used to scroll text on a pixel-by-pixel basis. Multiple text labels with the scrolling set to True will be cycled through.
- **float) text_anchor_point** (*(float,)*) – Values between 0 and 1 to indicate where the text position is relative to the label
- **is_data** (*bool*) – If True, fetch will attempt to update the label

scroll()

Scroll any text that needs scrolling by a single frame. We also we want to queue up multiple lines one after another. To get simultaneous lines, we can simply use a line break.

scroll_text (*frame_delay=0.02*)

Scroll the entire text all the way across. We also we want to queue up multiple lines one after another. To get simultaneous lines, we can simply use a line break.

set_background (*file_or_color, position=None*)

The background image to a bitmap file.

Parameters file_or_color – The filename of the chosen background image, or a hex color.

5.6 adafruit_matrixportal.network

Helper library for the MatrixPortal M4 or Adafruit RGB Matrix Shield + Metro M4 Airlift Lite.

- Author(s): Melissa LeBlanc-Williams

5.6.1 Implementation Notes

Hardware:

- Adafruit MatrixPortal M4
- Adafruit Metro M4 Express AirLift
- Adafruit RGB Matrix Shield
- 64x32 RGB LED Matrix

Software and Dependencies:

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

class adafruit_matrixportal.network.**Network** (***kwargs*)

Class representing the Adafruit RGB Matrix Portal.

Parameters

- **status_neopixel** – The pin for the status NeoPixel. Use board.NEOPIXEL for the on-board NeoPixel. Defaults to None, not the status LED
- **esp** – A passed ESP32 object, Can be used in cases where the ESP32 chip needs to be used before calling the pyportal class. Defaults to None.
- **external_spi** (*busio.SPI*) – A previously declared spi object. Defaults to None.
- **extract_values** (*bool*) – If true, single-length fetched values are automatically extracted from lists and tuples. Defaults to True.
- **debug** – Turn on debug print outs. Defaults to False.

ip_address

Return the IP Address nicely formatted

CHAPTER 6

Indices and tables

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

a

`adafruit_matrixportal.graphics`, 14
`adafruit_matrixportal.matrix`, 13
`adafruit_matrixportal.matrixportal`, 15
`adafruit_matrixportal.network`, 17

A

`adafruit_matrixportal.graphics` (*module*),
14
`adafruit_matrixportal.matrix` (*module*), 13
`adafruit_matrixportal.matrixportal` (*mod-
ule*), 15
`adafruit_matrixportal.network` (*module*), 17
`add_text()` (*adafruit_matrixportal.matrixportal.MatrixPortal
method*), 16

G

`Graphics` (*class in adafruit_matrixportal.graphics*), 14

I

`ip_address` (*adafruit_matrixportal.network.Network
attribute*), 17

M

`Matrix` (*class in adafruit_matrixportal.matrix*), 14
`MatrixPortal` (*class in
adafruit_matrixportal.matrixportal*), 15

N

`Network` (*class in adafruit_matrixportal.network*), 17

S

`scroll()` (*adafruit_matrixportal.matrixportal.MatrixPortal
method*), 17
`scroll_text()` (*adafruit_matrixportal.matrixportal.MatrixPortal
method*), 17
`set_background()` (*adafruit_matrixportal.matrixportal.MatrixPortal
method*), 17