AdafruitPyPortal Library Documentation

Release 1.0

Limor Fried

Sep 24, 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 Internet test . . . 12
- 5.3 QR Test . . . 13
- 5.4 `adafruit_pyportal` . . . 14
  - 5.4.1 Implementation Notes . . . 14
- 5.5 `adafruit_pyportal.graphics` . . . 16
  - 5.5.1 Implementation Notes . . . 16
- 5.6 `adafruit_pyportal.network` . . . 17
  - 5.6.1 Implementation Notes . . . 17
- 5.7 `adafruit_pyportal.peripherals` . . . 18
  - 5.7.1 Implementation Notes . . . 18

6 Indices and tables **19**

Python Module Index **21**

Index **23**

CircuitPython driver for Adafruit PyPortal.

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

See examples/pyportal_simpletest.py

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/pyportal_simpletest.py

```
# SPDX-FileCopyrightText: 2017 Scott Shawcroft, written for Adafruit Industries
#
# SPDX-License-Identifier: MIT

# NOTE: Make sure you've created your secrets.py file before running this example
# https://learn.adafruit.com/adafruit-pyportal/internet-connect#whats-a-secrets-file-17-2
import board
from adafruit_pyportal import PyPortal

# Set a data source URL
TEXT_URL = "http://wifitest.adafruit.com/testwifi/index.html"

# Create the PyPortal object
pyportal = PyPortal(url=TEXT_URL, status_neopixel=board.NEOPIXEL)

# Set display to show REPL
board.DISPLAY.show(None)

# Go get that data
print("Fetching text from", TEXT_URL)
data = pyportal.fetch()

# Print out what we got
print("-" * 40)
print(data)
print("-" * 40)
```

5.2 Internet test

Example to illustrate the device capability to get json data

Listing 2: examples/pyportal_internet_json_fetching.py

```
# SPDX-FileCopyrightText: 2019 ladyada for Adafruit Industries
# SPDX-License-Identifier: MIT

"""
Example to illustrate the device capability to get json data
"""

# NOTE: Make sure you've created your secrets.py file before running this example
# https://learn.adafruit.com/adafruit-pyportal/internet-connect#whats-a-secrets-file-17-2
import board
from digitalio import DigitalInOut
import adafruit_requests as requests
import adafruit_esp32spi.adafruit_esp32spi_socket as socket
from adafruit_esp32spi import adafruit_esp32spi


# Get wifi details and more from a secrets.py file
try:
    from secrets import secrets
except ImportError:
    print("WiFi secrets are kept in secrets.py, please add them there!")
    raise

print("ESP32 SPI webclient test")

TEXT_URL = "http://wifitest.adafruit.com/testwifi/index.html"
JSON_URL = "http://api.coindesk.com/v1/bpi/currentprice/USD.json"


# ESP32 Pins:
esp32_cs = DigitalInOut(board.ESP_CS)
esp32_ready = DigitalInOut(board.ESP_BUSY)
esp32_reset = DigitalInOut(board.ESP_RESET)

# SPI Configuration
spi = board.SPI()
esp = adafruit_esp32spi.ESP_SPIcontrol(spi, esp32_cs, esp32_ready, esp32_reset)
requests.set_socket(socket, esp)

if esp.status == adafruit_esp32spi.WL_IDLE_STATUS:
    print("ESP32 found and in idle mode")
print("Firmware vers.", esp.firmware_version)
print("MAC addr:", [hex(i) for i in esp.MAC_address])

for ap in esp.scan_networks():
    print("\t%s\t\tRSSI: %d" % (str(ap["ssid"], "utf-8"), ap["rssi"]))

print("Connecting to AP...")
while not esp.is_connected:
    try:
```

(continues on next page)

(continued from previous page)

```
        esp.connect_AP(secrets["ssid"], secrets["password"])
    except RuntimeError as e:
        print("could not connect to AP, retrying: ", e)
        continue
print("Connected to", str(esp.ssid, "utf-8"), "\tRSSI:", esp.rssi)
print("My IP address is", esp.pretty_ip(esp.ip_address))
print(
    "IP lookup adafruit.com: %s" % esp.pretty_ip(esp.get_host_by_name("adafruit.com"))
)
print("Ping google.com: %d ms" % esp.ping("google.com"))

# esp._debug = True
print("Fetching text from", TEXT_URL)
r = requests.get(TEXT_URL)
print("-" * 40)
print(r.text)
print("-" * 40)
r.close()

print()
print("Fetching json from", JSON_URL)
r = requests.get(JSON_URL)
print("-" * 40)
print(r.json())
print("-" * 40)
r.close()
print("Done!")
```

5.3 QR Test

This example shows a web address QR in the display

Listing 3: examples/pyportal_qrcode_generation.py

```
# SPDX-FileCopyrightText: 2021 Jose David M.
# SPDX-License-Identifier: MIT

"""
This example shows a web address QR in the display
"""

import board
from adafruit_pyportal.graphics import Graphics

# Set display to show
display = board.DISPLAY

# Background Information
base = Graphics(default_bg=0x990099, debug=True)

# WebPage to show in the QR
webpage = "http://www.adafruit.com"

# QR size Information
```

(continues on next page)

(continued from previous page)

```
qr_size = 9  # Pixels
scale = 3

# Create a barcode
base.qrcode(
    webpage,
    qr_size=scale,
    x=display.width // 2 - qr_size * scale,
    y=display.height // 2 - qr_size * scale,
)

while True:
    pass
```

5.4 adafruit_pyportal

CircuitPython driver for Adafruit PyPortal.

- Author(s): Limor Fried, Kevin J. Walters, Melissa LeBlanc-Williams

5.4.1 Implementation Notes

Hardware:

- Adafruit PyPortal

Software and Dependencies:

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

class adafruit_pyportal.**PyPortal**(*, *url=None*, *headers=None*, *json_path=None*, *regexp_path=None*, *convert_image=True*, *default_bg=0*, *status_neopixel=None*, *text_font=<sphinx.ext.autodoc.importer._MockObject object>*, *text_position=None*, *text_color=8421504*, *text_wrap=False*, *text_maxlen=0*, *text_transform=None*, *text_scale=1*, *json_transform=None*, *image_json_path=None*, *image_resize=None*, *image_position=None*, *image_dim_json_path=None*, *caption_text=None*, *caption_font=None*, *caption_position=None*, *caption_color=8421504*, *image_url_path=None*, *success_callback=None*, *esp=None*, *external_spi=None*, *debug=False*, *secrets_data=None*)

Class representing the Adafruit PyPortal.

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.

- **convert_image** – Determine whether or not to use the AdafruitIO image converter service. Set as False if your image is already resized. Defaults to True.
- **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
- **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** – Whether or not to wrap text (for long text data chunks). Defaults to `False`, no wrapping.
- **text_maxlen** – The max length of the text for text wrapping. 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
- **json_transform** – A function or a list of functions to call with the parsed JSON. Changes and additions are permitted for the `dict` object.
- **image_json_path** – The JSON traversal path for a background image to display. Defaults to `None`.
- **image_resize** – What size to resize the image we got from the json_path, make this a tuple of the width and height you want. Defaults to `None`.
- **image_position** – The position of the image on the display as an (x, y) tuple. Defaults to `None`.
- **image_dim_json_path** – The JSON traversal path for the original dimensions of image tuple. Used with fetch(). Defaults to `None`.
- **success_callback** – A function we'll call if you like, when we fetch data successfully. Defaults to `None`.
- **caption_text** (*str*) – The text of your caption, a fixed text not changed by the data we get. Defaults to `None`.
- **caption_font** (*str*) – The path to the font file for your caption. Defaults to `None`.
- **caption_position** – The position of your caption on the display as an (x, y) tuple. Defaults to `None`.
- **caption_color** – The color of your caption. Must be a hex value, e.g. `0x808000`.
- **image_url_path** – The HTTP traversal path for a background image to display. Defaults to `None`.
- **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`.
- **debug** – Turn on debug print outs. Defaults to False.

fetch(*refresh_url=None*, *timeout=10*)
Fetch data from the url we initialized with, perfom any parsing, and display text or graphics. This function does pretty much everything Optionally update the URL

set_caption(*caption_text*, *caption_position*, *caption_color*)
A caption. Requires setting `caption_font` in init!

Parameters

- **caption_text** – The text of the caption.
- **caption_position** – The position of the caption text.
- **caption_color** – The color of your caption text. Must be a hex value, e.g. `0x808000`.

5.5 adafruit_pyportal.graphics

CircuitPython driver for Adafruit PyPortal.

- Author(s): Limor Fried, Kevin J. Walters, Melissa LeBlanc-Williams

5.5.1 Implementation Notes

Hardware:

- Adafruit PyPortal

Software and Dependencies:

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

class adafruit_pyportal.graphics.**Graphics**(*, *default_bg=None*, *debug=False*)
Graphics Helper Class for the PyPortal Library

Parameters

- **default_bg** – The path to your default background image file or a hex color. Defaults to 0x000000.
- **debug** – Turn on debug print outs. Defaults to False.

hide_QR()
Clear any QR codes that are currently on the screen

qrcode(*qr_data*, *, *qr_size=1*, *x=0*, *y=0*, *hide_background=False*)
Display a QR code

Parameters

- **qr_data** – The data for the QR code.
- **qr_size** (*int*) – The scale of the QR code.
- **x** – The x position of upper left corner of the QR code on the display.
- **y** – The y position of upper left corner of the QR code on the display.

5.6 adafruit_pyportal.network

CircuitPython driver for Adafruit PyPortal.

- Author(s): Limor Fried, Kevin J. Walters, Melissa LeBlanc-Williams

5.6.1 Implementation Notes

Hardware:

- Adafruit PyPortal

Software and Dependencies:

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

class adafruit_pyportal.network.**Network**(*, *status_neopixel=None*, *esp=None*, *external_spi=None*, *extract_values=True*, *debug=False*, *convert_image=True*, *image_url_path=None*, *image_json_path=None*, *image_resize=None*, *image_position=None*, *image_dim_json_path=None*, *secrets_data=None*)

Class representing the Adafruit PyPortal.

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.
- **convert_image** – Determine whether or not to use the AdafruitIO image converter service. Set as False if your image is already resized. Defaults to True.
- **image_url_path** – The HTTP traversal path for a background image to display. Defaults to `None`.
- **image_json_path** – The JSON traversal path for a background image to display. Defaults to `None`.
- **image_resize** – What size to resize the image we got from the json_path, make this a tuple of the width and height you want. Defaults to `None`.
- **image_position** – The position of the image on the display as an (x, y) tuple. Defaults to `None`.
- **image_dim_json_path** – The JSON traversal path for the original dimensions of image tuple. Used with fetch(). Defaults to `None`.

image_converter_url(*image_url*, *width*, *height*, *color_depth=16*)

Generate a converted image url from the url passed in, with the given width and height. aio_username and aio_key must be set in secrets.

ip_address
Return the IP Address nicely formatted

process_image(*json_data*, *sd_card=False*)
Process image content

Parameters

- **json_data** – The JSON data that we can pluck values from
- **sd_card** (*bool*) – Whether or not we have an SD card inserted

5.7 adafruit_pyportal.peripherals

CircuitPython driver for Adafruit PyPortal.

- Author(s): Limor Fried, Kevin J. Walters, Melissa LeBlanc-Williams

5.7.1 Implementation Notes

Hardware:

- Adafruit PyPortal

Software and Dependencies:

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

class adafruit_pyportal.peripherals.**Peripherals**(*spi*, *display*, *splash_group*, *debug=False*)
Peripherals Helper Class for the PyPortal Library

play_file(*file_name*, *wait_to_finish=True*)
Play a wav file.

Parameters **file_name** (*str*) – The name of the wav file to play on the speaker.

sd_check()
Returns True if there is an SD card preset and False if there is no SD card. The _sdcard value is set in _init

set_backlight(*val*)
Adjust the TFT backlight.

Parameters **val** – The backlight brightness. Use a value between `0` and `1`, where `0` is off, and `1` is 100% brightness.

speaker_disable
Enable or disable the speaker for power savings

CHAPTER 6

Indices and tables

- genindex
- modindex
- search

Python Module Index

a

adafruit_pyportal, 14
adafruit_pyportal.graphics, 16
adafruit_pyportal.network, 16
adafruit_pyportal.peripherals, 18

Index

A

adafruit_pyportal (*module*), 14
adafruit_pyportal.graphics (*module*), 16
adafruit_pyportal.network (*module*), 16
adafruit_pyportal.peripherals (*module*), 18

F

fetch() (*adafruit_pyportal.PyPortal method*), 15

G

Graphics (*class in adafruit_pyportal.graphics*), 16

H

hide_QR() (*adafruit_pyportal.graphics.Graphics method*), 16

I

image_converter_url() (*adafruit_pyportal.network.Network method*), 17
ip_address (*adafruit_pyportal.network.Network attribute*), 17

N

Network (*class in adafruit_pyportal.network*), 17

P

Peripherals (*class in adafruit_pyportal.peripherals*), 18
play_file() (*adafruit_pyportal.peripherals.Peripherals method*), 18
process_image() (*adafruit_pyportal.network.Network method*), 18
PyPortal (*class in adafruit_pyportal*), 14

Q

qrcode() (*adafruit_pyportal.graphics.Graphics method*), 16

S

sd_check() (*adafruit_pyportal.peripherals.Peripherals method*), 18
set_backlight() (*adafruit_pyportal.peripherals.Peripherals method*), 18
set_caption() (*adafruit_pyportal.PyPortal method*), 16
speaker_disable (*adafruit_pyportal.peripherals.Peripherals attribute*), 18