

---

# **AdafruitSlideshow Library Documentation**

*Release 1.0*

**Kattni Rembor**

**Dec 19, 2020**



---

## Contents

---

<b>1</b>	<b>Dependencies</b>	<b>3</b>
<b>2</b>	<b>Installing from PyPI</b>	<b>5</b>
<b>3</b>	<b>Usage Example</b>	<b>7</b>
<b>4</b>	<b>Contributing</b>	<b>9</b>
<b>5</b>	<b>Documentation</b>	<b>11</b>
<b>6</b>	<b>Table of Contents</b>	<b>13</b>
6.1	Simple test . . . . .	13
6.2	adafruit_slideshow . . . . .	14
6.2.1	Implementation Notes . . . . .	14
<b>7</b>	<b>Indices and tables</b>	<b>17</b>
	<b>Python Module Index</b>	<b>19</b>
	<b>Index</b>	<b>21</b>



CircuitPython helper library for displaying a slideshow of images on a display.



# 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](#).





---

### 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-slideshow
```

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

```
sudo pip3 install adafruit-circuitpython-slideshow
```

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-slideshow
```



## CHAPTER 3

---

### Usage Example

---

```
from adafruit_slideshow import PlayBackOrder, SlideShow
import board
import pulseio

# Create the slideshow object that plays through once alphabetically.
slideshow = SlideShow(board.DISPLAY, pulseio.PWMOut(board.TFT_BACKLIGHT), folder="/",
                      loop=False, order=PlayBackOrder.ALPHABETICAL)

while slideshow.update():
    pass
```



## CHAPTER 4

---

### Contributing

---

Contributions are welcome! Please read our [Code of Conduct](#) before contributing to help this project stay welcoming.



## CHAPTER 5

---

### Documentation

---

For information on building library documentation, please check out [this guide](#).





## 6.1 Simple test

Ensure your device works with this simple test.

Listing 1: examples/slideshow\_simpletest.py

```
1  """Basic demonstration script will create a slideshow
2  object that plays through once alphabetically."""
3  import board
4  from adafruit_slideshow import PlaybackOrder, SlideShow
5
6  # use built in display (PyPortal, PyGamer, PyBadge, CLUE, etc.)
7  # see guide for setting up external displays (TFT / OLED breakouts, RGB matrices, etc.
8  # → https://learn.adafruit.com/circuitpython-display-support-using-displayio/display-
9  # → and-display-bus
10 display = board.DISPLAY
11
12 # pylint: disable=no-member
13
14 slideshow = SlideShow(
15     board.DISPLAY,
16     None,
17     folder="/images/",
18     loop=False,
19     order=PlaybackOrder.ALPHABETICAL,
20     dwell=10,
21 )
22 while slideshow.update():
23     pass
```

## 6.2 adafruit\_slideshow

CircuitPython helper library for displaying a slideshow of images on a display.

- Author(s): Kattni Rembor, Carter Nelson, Roy Hooper, Melissa LeBlanc-Williams

### 6.2.1 Implementation Notes

#### Hardware:

- Adafruit Hallowing M0 Express

#### Software and Dependencies:

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

**class** `adafruit_slideshow.HorizontalAlignment`

Defines possible horizontal alignment orders.

**class** `adafruit_slideshow.PlayBackDirection`

Defines possible slideshow playback directions.

**BACKWARD = -1**

The next image is before the current image. When alphabetically sorted, this is towards A.

**FORWARD = 1**

The next image is after the current image. When alphabetically sorted, this is towards Z.

**class** `adafruit_slideshow.PlayBackOrder`

Defines possible slideshow playback orders.

**ALPHABETICAL = 0**

Orders by alphabetical sort of filenames

**RANDOM = 1**

Randomly shuffles the images

**class** `adafruit_slideshow.SlideShow`(*display*, *backlight\_pwm=None*, \*, *folder='/'*, *order=0*, *loop=True*, *dwell=3*, *fade\_effect=True*, *auto\_advance=True*, *direction=1*, *h\_align=1*, *v\_align=1*)

Class for displaying a slideshow of .bmp images on displays.

#### Parameters

- **folder** (*str*) – Specify the folder containing the image files, in quotes. Default is the root directory, `"/"`.
- **order** (`PlayBackOrder`) – The order in which the images display. You can choose random (`RANDOM`) or alphabetical (`ALPHABETICAL`). Default is `ALPHABETICAL`.
- **loop** (*bool*) – Specify whether to loop the images or play through the list once. `True` if slideshow will continue to loop, `False` if it will play only once. Default is `True`.
- **dwell** (*int*) – The number of seconds each image displays, in seconds. Default is 3.
- **fade\_effect** (*bool*) – Specify whether to include the fade effect between images. `True` tells the code to fade the backlight up and down between image display transitions. `False` maintains max brightness on the backlight between image transitions. Default is `True`.

- **auto\_advance** (*bool*) – Specify whether to automatically advance after dwell seconds. True if slideshow should auto play, False if you want to control advancement manually. Default is True.
- **direction** (*PlayBackDirection*) – The playback direction.
- **h\_align** (*HorizontalAlignment*) – The Horizontal alignment of smaller/larger images
- **v\_align** (*VerticalAlignment*) – The Vertical alignment of smaller/larger images

Example code for Hallowing Express. With this example, the slideshow will play through once in alphabetical order:

```
from adafruit_slideshow import PlayBackOrder, SlideShow
import board
import pulseio

slideshow = SlideShow(board.DISPLAY, pulseio.PWMOut(board.TFT_BACKLIGHT), folder=
↳"/",
                        loop=False, order=PlayBackOrder.ALPHABETICAL)

while slideshow.update():
    pass
```

Example code for Hallowing Express. Sets dwell to 0 seconds, turns auto\_advance off, and uses capacitive touch to advance backwards and forwards through the images and to control the brightness level of the backlight:

```
from adafruit_slideshow import PlayBackOrder, SlideShow, PlayBackDirection
import touchio
import board
import pulseio

forward_button = touchio.TouchIn(board.TOUCH4)
back_button = touchio.TouchIn(board.TOUCH1)

brightness_up = touchio.TouchIn(board.TOUCH3)
brightness_down = touchio.TouchIn(board.TOUCH2)

slideshow = SlideShow(board.DISPLAY, pulseio.PWMOut(board.TFT_BACKLIGHT), folder=
↳"/",
                        auto_advance=False, dwell=0)

while True:
    if forward_button.value:
        slideshow.direction = PlayBackDirection.FORWARD
        slideshow.advance()
    if back_button.value:
        slideshow.direction = PlayBackDirection.BACKWARD
        slideshow.advance()

    if brightness_up.value:
        slideshow.brightness += 0.001
    elif brightness_down.value:
        slideshow.brightness -= 0.001
```

**advance** ()

Displays the next image. Returns True when a new image was displayed, False otherwise.

**auto\_advance = None**

Enable auto-advance based on dwell time. Set to `False` to manually control.

**brightness**

Brightness of the backlight when an image is displaying. Clamps to 0 to 1.0

**current\_slide\_name**

Returns the current image name.

**direction = None**

Specify the playback direction. Default is `PlayBackDirection.FORWARD`. Can also be `PlayBackDirection.BACKWARD`.

**dwell = None**

The number of seconds each slide displays, in seconds.

**fade\_effect = None**

Whether to include the fade effect between slides. `True` tells the code to fade the backlight up and down between slide display transitions. `False` maintains max brightness on the backlight between slide transitions.

**h\_align**

Get or Set the Horizontal Alignment

**loop = None**

Specifies whether to loop through the slides continuously or play through the list once. `True` will continue to loop, `False` will play only once.

**order**

The order in which the images display. You can choose random (`RANDOM`) or alphabetical (`ALPHA`).

**update ()**

Updates the slideshow to the next image.

**v\_align**

Get or Set the Vertical Alignment

**class** `adafruit_slideshow.VerticalAlignment`

Defines possible vertical alignment orders.

# CHAPTER 7

---

## Indices and tables

---

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



**a**

`adafruit_slideshow`, 13





## A

adafruit\_slideshow (*module*), 13  
advance () (*adafruit\_slideshow.SlideShow method*), 15  
ALPHABETICAL (*adafruit\_slideshow.PlayBackOrder attribute*), 14  
auto\_advance (*adafruit\_slideshow.SlideShow attribute*), 15

## B

BACKWARD (*adafruit\_slideshow.PlayBackDirection attribute*), 14  
brightness (*adafruit\_slideshow.SlideShow attribute*), 16

## C

current\_slide\_name  
(*adafruit\_slideshow.SlideShow attribute*), 16

## D

direction (*adafruit\_slideshow.SlideShow attribute*), 16  
dwell (*adafruit\_slideshow.SlideShow attribute*), 16

## F

fade\_effect (*adafruit\_slideshow.SlideShow attribute*), 16  
FORWARD (*adafruit\_slideshow.PlayBackDirection attribute*), 14

## H

h\_align (*adafruit\_slideshow.SlideShow attribute*), 16  
HorizontalAlignment (*class in adafruit\_slideshow*), 14

## L

loop (*adafruit\_slideshow.SlideShow attribute*), 16

## O

order (*adafruit\_slideshow.SlideShow attribute*), 16

## P

PlayBackDirection (*class in adafruit\_slideshow*), 14  
PlayBackOrder (*class in adafruit\_slideshow*), 14

## R

RANDOM (*adafruit\_slideshow.PlayBackOrder attribute*), 14

## S

SlideShow (*class in adafruit\_slideshow*), 14

## U

update () (*adafruit\_slideshow.SlideShow method*), 16

## V

v\_align (*adafruit\_slideshow.SlideShow attribute*), 16  
VerticalAlignment (*class in adafruit\_slideshow*), 16