
CircuitPython

DisplayIO *FlipInputLibraryDocumentation*

Release 1.0

Kevin Matocha

Jul 06, 2021

CONTENTS

1 Dependencies 3

2 Installing from PyPI 5

3 Usage Example 7

4 Contributing 9

5 Documentation 11

6 Table of Contents 13

6.1 Simple test 13

6.2 displayio_flipinput 15

6.2.1 Implementation Notes 15

7 Indices and tables 19

Python Module Index 21

Index 23

A flip style input selector. The value changes based on touch inputs on the

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](#) or individual libraries can be installed using [circup](#).

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 circuitpython-displayio-flipinput
```

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

```
sudo pip3 install circuitpython-displayio-flipinput
```

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 circuitpython-displayio-flipinput
```


USAGE EXAMPLE

See scripts in the examples directory of this repository.

CONTRIBUTING

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

DOCUMENTATION

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

TABLE OF CONTENTS

6.1 Simple test

Ensure your device works with this simple test.

Listing 1: examples/displayio_flipinput_simpletest.py

```
1  # SPDX-FileCopyrightText: 2021 Kevin Matocha
2  #
3  # SPDX-License-Identifier: MIT
4  #####
5  """
6  This is a basic demonstration of a FlipInput widget.
7  """
8
9  # pylint: disable=invalid-name
10
11 import time
12 import board
13 import displayio
14 import adafruit_touchscreen
15 from adafruit_bitmap_font import bitmap_font
16 from displayio_flipinput import FlipInput
17
18 display = board.DISPLAY # create the display on the PyPortal,
19 # otherwise change this to setup the display
20 # for display chip driver and pinout you have (e.g. ILI9341)
21
22 # setup the touchscreen
23 ts = adafruit_touchscreen.Touchscreen(
24     board.TOUCH_XL,
25     board.TOUCH_XR,
26     board.TOUCH_YD,
27     board.TOUCH_YU,
28     calibration=((5200, 59000), (5800, 57000)),
29     size=(display.width, display.height),
30 )
31
32 # Select the font file for use
33 font_file = "fonts/DSEG14Classic-Regular-64-ModS.pcf"
34 my_font = bitmap_font.load_font(font_file)
```

(continues on next page)

(continued from previous page)

```

35
36 my_flip1 = FlipInput(
37     display,
38     anchor_point=[0.0, 0.0],
39     anchored_position=[50, 40],
40     color=0xFF2222, # reddish orange color
41     value_list=[ # list of month strings, using three characters
42         "JAN",
43         "FEB",
44         "MAR",
45         "APR",
46         "MAY",
47         "JUN",
48         "JUL",
49         "AUG",
50         "SEP",
51         "OCT",
52         "NOV",
53         "DEC",
54     ],
55     font_scale=5,
56     horizontal=False, # use vertical arrows
57     animation_time=0.4,
58 )
59
60 my_flip2 = FlipInput(
61     display,
62     anchor_point=[0.0, 0.0],
63     anchored_position=[220, 40],
64     color=0xFF2222, # reddish orange color
65     value_list=["{:0:02d}".format(x) for x in range(1, 31 + 1)],
66     # use a list of strings from 01 through 31
67     # use the {:0:02d} format string to always use two digits (e.g. '03')
68     font_scale=5,
69     horizontal=False, # use vertical arrows
70     animation_time=0.4,
71 )
72
73 my_flip3 = FlipInput(
74     display,
75     anchor_point=[0.5, 1.0],
76     anchored_position=[320 // 2, 240 - 10],
77     color=0xFF2222, # reddish orange color
78     value_list=["{}".format(x) for x in range(1985, 2022, 1)],
79     # use a list with values of strings from 1985 to 2022
80     font=my_font,
81     horizontal=True, # use horizontal arrows
82     animation_time=0.8, # add more time since the animation covers a longer distance
83 )
84
85 # Pick an interesting date to start
86 #

```

(continues on next page)

(continued from previous page)

```

87 # You can set the value by direct integer indexes of the list or by a string
88 # Here are three ways to set the values:
89 my_flip1.value = 9 # use direct integer indexing to set the value to the 10th month
90 my_flip2.value = my_flip2.value_list.index("21") # find the index yourself by
91 # searching the value_list
92 my_flip3.value = "2015" # or set the value based on a string that is in the value_list
93
94 # Create the group to display and append the FlipInput widgets
95 my_group = displayio.Group()
96 my_group.append(my_flip1)
97 my_group.append(my_flip2)
98 my_group.append(my_flip3)
99
100 display.show(my_group) # add high level Group to the display
101 display.auto_refresh = True
102
103 while True:
104
105     p = ts.touch_point
106     # print("touch_point p: {}".format(p)) # print the touch point
107
108     if p: # if touched, check if any of the widgets was triggered
109         if my_flip1.contains(p):
110             my_flip1.selected(p)
111             time.sleep(0.15) # add a short delay to reduce accidental press
112         elif my_flip2.contains(p):
113             my_flip2.selected(p)
114             time.sleep(0.15) # add a short delay to reduce accidental press
115         elif my_flip3.contains(p):
116             my_flip3.selected(p)
117             time.sleep(0.15) # add a short delay to reduce accidental press
118
119     time.sleep(0.01) # small delay seems to improve touch response

```

6.2 displayio_flipinput

A flip style input selector.

- Author(s): Kevin Matocha

6.2.1 Implementation Notes

Hardware:

Software and Dependencies:

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

```
class displayio_flipinput.FlipInput(display, *, value_list=None, font=<fontio.BuiltinFont object>,
                                     font_scale=1, color=16777215, value=0, arrow_touch_padding=0,
                                     arrow_color=3355443, arrow_outline=5592405, arrow_height=30,
                                     arrow_width=30, arrow_gap=5, alt_touch_padding=0,
                                     horizontal=True, animation_time=None, cool_down=0.0, **kwargs)
```

A flip style input selector. The value changes based on touch inputs on the two halves of the indicator with optional arrows added.

Parameters

- **x** (*int*) – pixel position
- **y** (*int*) – pixel position
- **display** (*displayio.Display*) – the display where the widget will be displayed
- **value_list** (*List[str]*) – the list of strings that will be displayed
- **font** (*Font*) – the font used for the text (defaults to `terminalio.FONT`)
- **font_scale** (*int*) – the scaling of the font in integer values (default is 1)
- **color** (*int*) – the color used for the font (default is 0xFFFFFF)
- **value** (*int*) – the index into the value_list that is initially displayed (default is 0)
- **arrow_color** (*int*) – the color used for the arrow fill (default is 0x333333)
- **arrow_outline** (*int*) – the color used for the arrow outline (default is 0x555555)
- **arrow_height** (*int*) – the height of the arrows, in pixels (default is 30 pixels)
- **arrow_width** (*int*) – the width of the arrows, in pixels (default is 30 pixels)
- **arrow_gap** (*int*) – distance from text to the arrow, in pixels (default is 5), can also be a negative value
- **arrow_touch_padding** (*int*) – additional padding on the arrow sides of the widget where touch response is accepted, in pixels (default = 0)
- **alt_touch_padding** (*int*) – additional padding on the non-arrow sides of the widget where touch response is accepted, in pixels (default = 0)
- **horizontal** (*Boolean*) – set to `True` to display arrows are in the horizontal direction, set `False` for arrows in the vertical direction (default = `True`)
- **animation_time** (*float*) – duration for the animation during flipping between values, in seconds (default is 0.4 seconds), set to 0.0 or `None` for no animation.
- **cool_down** (*float*) – minimum duration between activations of the widget with a continuous pressing, this can be used to reduce the chance of accidental multiple activations, in seconds (default is 0.0 seconds, no delay). Set to -1.0 to require the button be released and pressed again for activation (Note: This requires calling the `released` function prior to the next call to `selected`.)

Create a Group of a given size and scale. Scale is in one dimension. For example, `scale=2` leads to a layer's pixel being 2x2 pixels when in the group.

contains(*touch_point*)

Returns True if the touch_point is within the widget's touch_boundary.

released()

Response function when the Control is released. Resets the state variables for handling situation when cool_down is < 0 that requires `released()` before reacting another another `selected()`.

selected(*touch_point*)

Response function when the Control is selected. Increases value when upper half is pressed and decreases value when lower half is pressed.

property value

The value index displayed on the widget. For the setter, the input can either be an `int` index into the `value_list` or can be a `str` that matches one of the items in the `value_list`. If `int`, the value will be set based on the modulus of the input `new_value`.

Returns `int`

INDICES AND TABLES

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

PYTHON MODULE INDEX

d

`displayio_flipinput`, [15](#)

INDEX

C

`contains()` (*displayio_flipinput.FlipInput method*), 16

D

`displayio_flipinput`
module, 15

F

`FlipInput` (*class in displayio_flipinput*), 15

M

module
 `displayio_flipinput`, 15

R

`released()` (*displayio_flipinput.FlipInput method*), 16

S

`selected()` (*displayio_flipinput.FlipInput method*), 16

V

`value` (*displayio_flipinput.FlipInput property*), 17