

---

# **AdafruitNeoTrellis Library Documentation**

***Release 1.0***

**Dean Miller**

**May 06, 2019**



---

## Contents

---

<b>1</b>	<b>Dependencies</b>	<b>3</b>
<b>2</b>	<b>Usage Example</b>	<b>5</b>
<b>3</b>	<b>Contributing</b>	<b>7</b>
<b>4</b>	<b>Building locally</b>	<b>9</b>
4.1	Zip release files . . . . .	9
4.2	Sphinx documentation . . . . .	9
<b>5</b>	<b>Table of Contents</b>	<b>11</b>
5.1	Simple test . . . . .	11
5.2	adafruit_neotrellis . . . . .	12
5.2.1	Implementation Notes . . . . .	12
<b>6</b>	<b>Indices and tables</b>	<b>13</b>
	<b>Python Module Index</b>	<b>15</b>



This is a library for using the Adafruit\_NeoTrellis boards with circuitpython.



# CHAPTER 1

---

## Dependencies

---

This driver depends on:

- [Adafruit CircuitPython](#)
- [Bus Device](#)
- [Register](#)
- [Adafruit Seesaw](#)

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/neotrellis_simpletest.py` for usage example



## CHAPTER 3

---

### Contributing

---

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



### 4.1 Zip release files

To build this library locally you'll need to install the `circuitpython-build-tools` package.

```
python3 -m venv .env
source .env/bin/activate
pip install circuitpython-build-tools
```

Once installed, make sure you are in the virtual environment:

```
source .env/bin/activate
```

Then run the build:

```
circuitpython-build-bundles --filename_prefix adafruit-circuitpython-neotrellis --
↳library_location .
```

### 4.2 Sphinx documentation

Sphinx is used to build the documentation based on rST files and comments in the code. First, install dependencies (feel free to reuse the virtual environment from above):

```
python3 -m venv .env
source .env/bin/activate
pip install Sphinx sphinx-rtd-theme
```

Now, once you have the virtual environment activated:

```
cd docs
sphinx-build -E -W -b html . _build/html
```

This will output the documentation to `docs/_build/html`. Open the `index.html` in your browser to view them. It will also (due to `-W`) error out on any warning like Travis will. This is a good way to locally verify it will pass.

## 5.1 Simple test

Ensure your device works with this simple test.

Listing 1: examples/neotrellis\_simpletest.py

```
1  import time
2
3  from board import SCL, SDA
4  import busio
5  from adafruit_neotrellis.neotrellis import NeoTrellis
6
7  #create the i2c object for the trellis
8  i2c_bus = busio.I2C(SCL, SDA)
9
10 #create the trellis
11 trellis = NeoTrellis(i2c_bus)
12
13 #some color definitions
14 OFF = (0, 0, 0)
15 RED = (255, 0, 0)
16 YELLOW = (255, 150, 0)
17 GREEN = (0, 255, 0)
18 CYAN = (0, 255, 255)
19 BLUE = (0, 0, 255)
20 PURPLE = (180, 0, 255)
21
22 #this will be called when button events are received
23 def blink(event):
24     #turn the LED on when a rising edge is detected
25     if event.edge == NeoTrellis.EDGE_RISING:
26         trellis.pixels[event.number] = CYAN
27     #turn the LED off when a rising edge is detected
```

(continues on next page)

(continued from previous page)

```

28     elif event.edge == NeoTrellis.EDGE_FALLING:
29         trellis.pixels[event.number] = OFF
30
31 for i in range(16):
32     #activate rising edge events on all keys
33     trellis.activate_key(i, NeoTrellis.EDGE_RISING)
34     #activate falling edge events on all keys
35     trellis.activate_key(i, NeoTrellis.EDGE_FALLING)
36     #set all keys to trigger the blink callback
37     trellis.callbacks[i] = blink
38
39     #cycle the LEDs on startup
40     trellis.pixels[i] = PURPLE
41     time.sleep(.05)
42
43 for i in range(16):
44     trellis.pixels[i] = OFF
45     time.sleep(.05)
46
47 while True:
48     #call the sync function call any triggered callbacks
49     trellis.sync()
50     #the trellis can only be read every 17 milliseconds or so
51     time.sleep(.02)

```

## 5.2 adafruit\_neotrellis

4x4 elastomer buttons and RGB LEDs

- Author(s): Dean Miller

### 5.2.1 Implementation Notes

#### Hardware:

#### Software and Dependencies:

- Adafruit CircuitPython firmware for the supported boards: <https://github.com/adafruit/circuitpython/releases>
- Adafruit Seesaw CircuitPython library [https://github.com/adafruit/Adafruit\\_CircuitPython\\_seesaw/releases](https://github.com/adafruit/Adafruit_CircuitPython_seesaw/releases)

**class** adafruit\_neotrellis.neotrellis.**NeoTrellis** (*i2c\_bus*, *interrupt=False*, *addr=46*,  
*drdy=None*)

Driver for the Adafruit NeoTrellis.

**activate\_key** (*key*, *edge*, *enable=True*)

Activate or deactivate a key on the trellis. Key is the key number from 0 to 16. Edge specifies what edge to register an event on and can be NeoTrellis.EDGE\_FALLING or NeoTrellis.EDGE\_RISING. enable should be set to True if the event is to be enabled, or False if the event is to be disabled.

**sync** ()

read any events from the Trellis hardware and call associated callbacks



## CHAPTER 6

---

### Indices and tables

---

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



### **a**

`adafruit_neotrellis.neotrellis`, [12](#)



## A

`activate_key()` (*adafruit\_neotrellis.neotrellis.NeoTrellis*  
*method*), [12](#)

`adafruit_neotrellis.neotrellis` (*module*),  
[12](#)

## N

`NeoTrellis` (*class in adafruit\_neotrellis.neotrellis*), [12](#)

## S

`sync()` (*adafruit\_neotrellis.neotrellis.NeoTrellis*  
*method*), [12](#)