
volux

Release 0.9.1

Mar 11, 2023

1	Installation	1
1.1	Installing with Pip	1
2	Running Demos	3
2.1	Bar Demo	3
3	Examples	5
3.1	Classes	5
4	Python	7
4.1	Classes	7
5	Installing from source	9
6	Writing your own modules	11
6.1	Example Module	11
7	volux	13
7.1	volux package	13
7.2	voluxbar package	14
7.3	voluxclprint package	14
7.4	voluxdisplay package	14
7.5	voluxlight package	14
7.6	voluxvolume package	14
8	Indices and tables	15

1.1 Installing with Pip

```
$ pip install volux
```

Congratulations, you’ve just installed Volux!

Note: If you would rather build the module yourself, see “Installing from source”

Next, let’s try running the ‘bar’ demo.

2.1 Bar Demo

Once volux is finished installing, you can run the bar demo like so:

```
$ volux demo bar
```

You should now see a transparent bar at the bottom of your screen.

2.1.1 Bar Demo Explained

In different modes, performing identical actions on the bar produce different results.

Below, you'll see a table of representing the result of performing certain actions on the bar in a given mode.

Bar Color	Action	Result
Any	Right-click Double right-click	Change bar mode Exit demo
Green	Scroll up Scroll down Middle-click	Increase volume Decrease volume Mute
Red	Scroll up Scroll down Middle-click	Increase volume Decrease volume Unmute
Blue	Scroll up Scroll down Middle-click	Increase bulb brightness Decrease bulb brightness Toggle bulb power

3.1 Classes

3.1.1 VoluxLight

Warning: This section is a work-in-progress

Adding New Lights

```
from voluxlight import VoluxLight
vlight = VoluxLight(<label>)

<operator instance>.add_module(vlight)
```

<label> must be a string. This is the name you've set via your mobile for a LIFX bulb on your network.

In the bar demo, we have a bulb with the label 'Demo Bulb', so we set <label> to match it accordingly.

4.1 Classes

4.1.1 VoluxOperator

Instantiating a New Operator

```
import volux
vlx = volux.VoluxOperator()
```

Note: `vlx` will serve as the name of the operator instance in the following examples

Adding Modules to an Operator

```
from voluxdemomodule import VoluxDemoModule
demo_module = VoluxDemoModule()
vlx.add_module()
```

Installing from source

Warning: If any of the code below looks unfamiliar or scary to you, building from source is not a good idea. It is highly recommended to install volux from pip or at least via the latest wheel release on pypi.

```
$ git clone https://github.com/DrTexx/Volux.git
$ cd Volux
$ python3 -m venv venv
$ source venv/bin/activate
$ pip install -r volux/demos/demo_volbar_requirements.txt
$ pip install wheel setuptools --upgrade
$ python3 setup.py bdist_wheel
$ cd dist
$ pip install volux-*.whl
```

Writing your own modules

Warning: This section is a work-in-progress

6.1 Example Module

```
from volux import VoluxModule

class DecoyVoluxModule(VoluxModule):
    def __init__(self, *args, **kwargs):
        super().__init__(
            module_name="Decoy Module",
            module_attr="decoy",
            module_get=self.get,
            module_set=self.set,
        )
        self.val = 0

    def get(self):

        return self.val

    def set(self, new_val):

        self.val = new_val
```

`module_name` is the human-readable name for your volux module.

`module_attr` is the attribute which will be added to the *VoluxOperator* object.

`module_get` is the class method for getting the modules generic data

`module_set` is the class method for setting the modules generic data

7.1 volux package

7.1.1 Subpackages

volux.demos package

Submodules

volux.demos.vol_lux_bar module

Module contents

7.1.2 Submodules

7.1.3 volux.core module

7.1.4 volux.demo module

7.1.5 volux.module module

7.1.6 volux.operator module

7.1.7 Module contents

7.2 voluxbar package

7.2.1 Module contents

7.3 voluxcliprint package

7.3.1 Module contents

7.4 voluxdisplay package

7.4.1 Module contents

7.5 voluxlight package

7.5.1 Module contents

7.6 voluxvolume package

7.6.1 Module contents

CHAPTER 8

Indices and tables

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