

---

# **AdafruitVEML6075 Library Documentation**

***Release 1.0***

**ladyada**

**Jan 23, 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_veml6075 . . . . .    | 13        |
| 6.2.1    | Implementation Notes . . . . . | 13        |
| <b>7</b> | <b>Indices and tables</b>      | <b>15</b> |
|          | <b>Python Module Index</b>     | <b>17</b> |
|          | <b>Index</b>                   | <b>19</b> |



CircuitPython library to support VEML6075 UVA & UVB sensor.



# 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

---

### 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-veml6075
```

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

```
sudo pip3 install adafruit-circuitpython-veml6075
```

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-veml6075
```



## CHAPTER 3

---

### Usage Example

---

```
import time
import board
import busio
import adafruit_veml6075

i2c = busio.I2C(board.SCL, board.SDA)

veml = adafruit_veml6075.VEML6075(i2c, integration_time=100)

while True:
    print(veml.uv_index)
    time.sleep(1)
```



## 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/veml6075\_simpletest.py

```
1 import time
2 import board
3 import busio
4 import adafruit_veml6075
5
6 i2c = busio.I2C(board.SCL, board.SDA)
7
8 veml = adafruit_veml6075.VEML6075(i2c, integration_time=100)
9
10 print("Integration time: %d ms" % veml.integration_time)
11
12 while True:
13     print(veml.uv_index)
14     time.sleep(1)
```

### 6.2 adafruit\_veml6075

CircuitPython library to support VEML6075 UVA & UVB sensor.

- Author(s): ladyada

#### 6.2.1 Implementation Notes

**Hardware:**

**Software and Dependencies:**

- Adafruit CircuitPython firmware for the supported boards: <https://github.com/adafruit/circuitpython/releases>
- Adafruit's Bus Device library: [https://github.com/adafruit/Adafruit\\_CircuitPython\\_BusDevice](https://github.com/adafruit/Adafruit_CircuitPython_BusDevice)

```
class adafruit_veml6075.VEML6075 (i2c_bus, *, integration_time=50, high_dynamic=True,
                                   uva_a_coef=2.22, uva_b_coef=1.33, uvb_c_coef=2.95,
                                   uvb_d_coef=1.74, uva_response=0.001461,
                                   uvb_response=0.002591)
```

Driver base for the VEML6075 UV Light Sensor :param i2c\_bus: The `busio.I2C` object to use. This is the only required parameter. :param int integration\_time: The integration time you'd like to set initially. Available options - each in milliseconds: 50, 100, 200, 400, 800. The higher the `'_x_'` value, the more accurate the reading is (at the cost of less samples per reading). Defaults to 100ms if parameter not passed. To change setting after initialization, use `[veml6075].integration_time = new_it_value`. :param bool high\_dynamic: whether to put sensor in 'high dynamic setting' mode :param float uva\_a\_coef: the UVA visible coefficient :param float uva\_b\_coef: the UVA IR coefficient :param float uvb\_c\_coef: the UVB visible coefficient :param float uvb\_d\_coef: the UVB IR coefficient :param float uva\_response: the UVA responsivity :param float uvb\_response: the UVA responsivity

**integration\_time**

The amount of time the VEML is sampling data for, in millis. Valid times are 50, 100, 200, 400 or 800ms

**uv\_index**

The calculated UV Index

**uva**

The calibrated UVA reading, in 'counts' over the sample period

**uvb**

The calibrated UVB reading, in 'counts' over the sample period

## CHAPTER 7

---

### Indices and tables

---

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



### **a**

adafruit\_veml6075, [13](#)



## A

`adafruit_veml6075` (*module*), [13](#)

## I

`integration_time` (*adafruit\_veml6075.VEML6075 attribute*), [14](#)

## U

`uv_index` (*adafruit\_veml6075.VEML6075 attribute*), [14](#)

`uva` (*adafruit\_veml6075.VEML6075 attribute*), [14](#)

`uvb` (*adafruit\_veml6075.VEML6075 attribute*), [14](#)

## V

`VEML6075` (*class in adafruit\_veml6075*), [14](#)