

---

# **AdafruitVL6180X Library Documentation**

***Release 1.0***

**Tony DiCola**

**Feb 05, 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_vl6180x . . . . .	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 module for the VL6180X distance sensor. See `examples/vl6180x_simpletest.py` for a demo of the usage.



# 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-vl6180x
```

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

```
sudo pip3 install adafruit-circuitpython-vl6180x
```

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-vl6180x
```



## CHAPTER 3

---

### Usage Example

---

See `examples/vl6180x_simpletest.py` for a demo of the usage.



## 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/vl6180x\_simpletest.py

```
1  # Demo of reading the range and lux from the VL6180x distance sensor and
2  # printing it every second.
3  # Author: Tony DiCola
4  import time
5
6  import board
7  import busio
8
9  import adafruit_vl6180x
10
11
12 # Create I2C bus.
13 i2c = busio.I2C(board.SCL, board.SDA)
14
15 # Create sensor instance.
16 sensor = adafruit_vl6180x.VL6180X(i2c)
17
18 # Main loop prints the range and lux every second:
19 while True:
20     # Read the range in millimeters and print it.
21     range_mm = sensor.range
22     print('Range: {0}mm'.format(range_mm))
23     # Read the light, note this requires specifying a gain value:
24     # - adafruit_vl6180x.ALS_GAIN_1 = 1x
25     # - adafruit_vl6180x.ALS_GAIN_1_25 = 1.25x
26     # - adafruit_vl6180x.ALS_GAIN_1_67 = 1.67x
27     # - adafruit_vl6180x.ALS_GAIN_2_5 = 2.5x
```

(continues on next page)

(continued from previous page)

```
28 # - adafruit_vl6180x.ALS_GAIN_5 = 5x
29 # - adafruit_vl6180x.ALS_GAIN_10 = 10x
30 # - adafruit_vl6180x.ALS_GAIN_20 = 20x
31 # - adafruit_vl6180x.ALS_GAIN_40 = 40x
32 light_lux = sensor.read_lux(adafruit_vl6180x.ALS_GAIN_1)
33 print('Light (1x gain): {0}lux'.format(light_lux))
34 # Delay for a second.
35 time.sleep(1.0)
```

## 6.2 adafruit\_vl6180x

CircuitPython module for the VL6180X distance sensor. See examples/simpletest.py for a demo of the usage.

- Author(s): Tony DiCola

### 6.2.1 Implementation Notes

#### Hardware:

- Adafruit VL6180X Time of Flight Distance Ranging Sensor (VL6180) (Product ID: 3316)

#### Software and Dependencies:

- Adafruit CircuitPython firmware for the ESP8622 and M0-based 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\_vl6180x.VL6180X(i2c, address=41)

Create an instance of the VL6180X distance sensor. You must pass in the following parameters:

**Parameters** **i2c** – An instance of the I2C bus connected to the sensor.

Optionally you can specify:

**Parameters** **address** – The I2C address of the sensor. If not specified the sensor's default value will be assumed.

#### **range**

Read the range of an object in front of sensor and return it in mm.

#### **range\_status**

Retrieve the status/error from a previous range read. This will return a constant value such as:

- ERROR\_NONE - No error
- ERROR\_SYSERR\_1 - System error 1 (see datasheet)
- ERROR\_SYSERR\_5 - System error 5 (see datasheet)
- ERROR\_ECEFAIL - ECE failure
- ERROR\_NOCONVERGE - No convergence
- ERROR\_RANGEIGNORE - Outside range ignored
- ERROR\_SNR - Too much noise
- ERROR\_RAWUFLOW - Raw value underflow

- ERROR\_RAWOFLOW - Raw value overflow
- ERROR\_RANGEUFLOW - Range underflow
- ERROR\_RANGEOFLOW - Range overflow

**read\_lux** (*gain*)

Read the lux (light value) from the sensor and return it. Must specify the gain value to use for the lux reading: - ALS\_GAIN\_1 = 1x - ALS\_GAIN\_1\_25 = 1.25x - ALS\_GAIN\_1\_67 = 1.67x - ALS\_GAIN\_2\_5 = 2.5x - ALS\_GAIN\_5 = 5x - ALS\_GAIN\_10 = 10x - ALS\_GAIN\_20 = 20x - ALS\_GAIN\_40 = 40x



## CHAPTER 7

---

### Indices and tables

---

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



### **a**

adafruit\_vl6180x, [14](#)





### A

`adafruit_vl6180x` (*module*), [14](#)

### R

`range` (*adafruit\_vl6180x.VL6180X attribute*), [14](#)

`range_status` (*adafruit\_vl6180x.VL6180X attribute*),  
[14](#)

`read_lux()` (*adafruit\_vl6180x.VL6180X method*), [15](#)

### V

`VL6180X` (*class in adafruit\_vl6180x*), [14](#)