
AdafruitVL6180X Library Documentation

Release 1.0

Tony DiCola

Mar 03, 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	adafruit_vl6180x	14
6.2.1	Implementation Notes	14
7	Indices and tables	17
	Python Module Index	19
	Index	21

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  # SPDX-FileCopyrightText: 2018 Tony DiCola for Adafruit Industries
2  # SPDX-License-Identifier: MIT
3
4  # Demo of reading the range and lux from the VL6180x distance sensor and
5  # printing it every second.
6
7  import time
8
9  import board
10 import busio
11
12 import adafruit_vl6180x
13
14
15 # Create I2C bus.
16 i2c = busio.I2C(board.SCL, board.SDA)
17
18 # Create sensor instance.
19 sensor = adafruit_vl6180x.VL6180X(i2c)
20
21 # Main loop prints the range and lux every second:
22 while True:
23     # Read the range in millimeters and print it.
24     range_mm = sensor.range
25     print("Range: {0}mm".format(range_mm))
26     # Read the light, note this requires specifying a gain value:
27     # - adafruit_vl6180x.ALS_GAIN_1 = 1x
```

(continues on next page)

(continued from previous page)

```
28 # - adafruit_vl6180x.ALS_GAIN_1_25 = 1.25x
29 # - adafruit_vl6180x.ALS_GAIN_1_67 = 1.67x
30 # - adafruit_vl6180x.ALS_GAIN_2_5 = 2.5x
31 # - adafruit_vl6180x.ALS_GAIN_5 = 5x
32 # - adafruit_vl6180x.ALS_GAIN_10 = 10x
33 # - adafruit_vl6180x.ALS_GAIN_20 = 20x
34 # - adafruit_vl6180x.ALS_GAIN_40 = 40x
35 light_lux = sensor.read_lux(adafruit_vl6180x.ALS_GAIN_1)
36 print("Light (1x gain): {0}lux".format(light_lux))
37 # Delay for a second.
38 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

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](#)