

---

# **Adafruit VL53L0X Library Documentation**

*Release 1.0*

**Tony DiCola**

**Feb 03, 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	Multiple VL53L0X on Same I2C Bus . . . . .	14
6.3	adafruit_vl53l0x . . . . .	15
6.3.1	Implementation Notes . . . . .	15
<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 driver for the VL53L0X distance 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](#).





---

### 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-v15310x
```

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

```
sudo pip3 install adafruit-circuitpython-v15310x
```

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-v15310x
```



## CHAPTER 3

---

### Usage Example

---

See usage in the `examples/v15310x_simpletest.py` file.



## 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/vl53l0x\_simpletest.py

```
1 # Simple demo of the VL53L0X distance sensor.
2 # Will print the sensed range/distance every second.
3 import time
4
5 import board
6 import busio
7
8 import adafruit_vl53l0x
9
10 # Initialize I2C bus and sensor.
11 i2c = busio.I2C(board.SCL, board.SDA)
12 vl53 = adafruit_vl53l0x.VL53L0X(i2c)
13
14 # Optionally adjust the measurement timing budget to change speed and accuracy.
15 # See the example here for more details:
16 # https://github.com/pololu/vl53l0x-arduino/blob/master/examples/Single/Single.ino
17 # For example a higher speed but less accurate timing budget of 20ms:
18 #vl53.measurement_timing_budget = 20000
19 # Or a slower but more accurate timing budget of 200ms:
20 #vl53.measurement_timing_budget = 200000
21 # The default timing budget is 33ms, a good compromise of speed and accuracy.
22
23 # Main loop will read the range and print it every second.
24 while True:
25     print('Range: {0}mm'.format(vl53.range))
26     time.sleep(1.0)
```

## 6.2 Multiple VL53L0X on Same I2C Bus

Copy “./examples/vl53l0x\_multiple\_sensors.py” to your “CIRCUITPY” drive, then run the script with from vl53l0x\_multiple\_sensors import \*

Listing 2: examples/vl53l0x\_multiple\_sensors.py

```

1  """
2  Example of how to use the adafruit_vl53l0x library to change the assigned address of
3  multiple VL53L0X sensors on the same I2C bus. This example only focuses on 2 VL53L0X
4  sensors, but can be modified for more. BE AWARE: a multitude of sensors may require
5  more current than the on-board 3V regulator can output (typical current consumption,
6  ↳during
7  active range readings is about 19 mA per sensor).
8  """
9  import time
10 import board
11 from digitalio import DigitalInOut
12 from adafruit_vl53l0x import VL53L0X
13
14 # declare the singleton variable for the default I2C bus
15 i2c = board.I2C()
16
17 # declare the digital output pins connected to the "SHDN" pin on each VL53L0X sensor
18 xshut = [
19     DigitalInOut(board.D7),
20     DigitalInOut(board.D9),
21     # add more VL53L0X sensors by defining their SHDN pins here
22 ]
23
24 for power_pin in xshut:
25     # make sure these pins are a digital output, not a digital input
26     power_pin.switch_to_output(value=False)
27     # These pins are active when Low, meaning:
28     #   if the output signal is LOW, then the VL53L0X sensor is off.
29     #   if the output signal is HIGH, then the VL53L0X sensor is on.
30 # all VL53L0X sensors are now off
31
32 # initialize a list to be used for the array of VL53L0X sensors
33 vl53 = []
34
35 # now change the addresses of the VL53L0X sensors
36 for i, power_pin in enumerate(xshut):
37     # turn on the VL53L0X to allow hardware check
38     power_pin.value = True
39     # instantiate the VL53L0X sensor on the I2C bus & insert it into the "vl53" list
40     vl53.insert(i, VL53L0X(i2c)) # also performs VL53L0X hardware check
41     # no need to change the address of the last VL53L0X sensor
42     if i < len(xshut) - 1:
43         # default address is 0x29. Change that to something else
44         vl53[i].set_address(i + 0x30) # address assigned should NOT be already in use
45 # there is a helpful list of pre-designated I2C addresses for various I2C devices at
46 # https://learn.adafruit.com/i2c-addresses/the-list
47 # According to this list 0x30-0x34 are available, although the list may be incomplete.
48 # In the python REPR, you can scan for all I2C devices that are attached and determine
49 # their addresses using:
50 # >>> import board

```

(continues on next page)

(continued from previous page)

```

50 # >>> i2c = board.I2C()
51 # >>> if i2c.try_lock():
52 # >>>     [hex(x) for x in i2c.scan()]
53 # >>>     i2c.unlock()
54
55 def detect_range(count=5):
56     """ take count=5 samples """
57     while count:
58         for index, sensor in enumerate(vl53):
59             print('Sensor {} Range: {}mm'.format(index + 1, sensor.range))
60             time.sleep(1.0)
61             count -= 1
62
63 print("Multiple VL53L0X sensors' addresses are assigned properly\n"
64       "execute detect_range() to read each sensors range readings")

```

## 6.3 adafruit\_vl53l0x

CircuitPython driver for the VL53L0X distance sensor. This code is adapted from the pololu driver here: <https://github.com/pololu/vl53l0x-arduino>

See usage in the examples/vl53l0x\_simpletest.py file.

- Author(s): Tony DiCola

### 6.3.1 Implementation Notes

#### Hardware:

- Adafruit VL53L0X Time of Flight Distance Sensor - ~30 to 1000mm (Product ID: 3317)

#### 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_vl53l0x.VL53L0X` (*i2c*, *address=41*, *io\_timeout\_s=0*)

Driver for the VL53L0X distance sensor.

#### **measurement\_timing\_budget**

The measurement timing budget in microseconds.

#### **range**

Perform a single reading of the range for an object in front of the sensor and return the distance in millimeters.

#### **set\_address** (*new\_address*)

Set a new I2C address to the instantiated object. This is only called when using multiple VL53L0X sensors on the same I2C bus (SDA & SCL pins). See also the [example](#) for proper usage.

**Parameters** *new\_address* (*int*) – The 7-bit *int* that is to be assigned to the VL53L0X sensor. The address that is assigned should NOT be already in use by another device on the I2C bus.

---

**Important:** To properly set the address to an individual VL53L0X sensor, you must first ensure that all other VL53L0X sensors (using the default address of  $0x29$ ) on the same I2C bus are in their off state by pulling the “SHDN” pins LOW. When the “SHDN” pin is pulled HIGH again the default I2C address is  $0x29$ .

---

**signal\_rate\_limit**

The signal rate limit in mega counts per second.

# CHAPTER 7

---

## Indices and tables

---

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



**a**

`adafruit_vl53l0x`, 15





## A

adafruit\_vl53l0x (*module*), 15

## M

measurement\_timing\_budget  
(*adafruit\_vl53l0x.VL53L0X attribute*), 15

## R

range (*adafruit\_vl53l0x.VL53L0X attribute*), 15

## S

set\_address() (*adafruit\_vl53l0x.VL53L0X method*),  
15

signal\_rate\_limit (*adafruit\_vl53l0x.VL53L0X at-  
tribute*), 16

## V

VL53L0X (*class in adafruit\_vl53l0x*), 15