
Adafruit VL53L0X Library Documentation

Release 1.0

Tony DiCola

Jun 07, 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	Multiple VL53L0X on Same I2C Bus	14
6.3	adafruit_vl53l0x	15
6.3.1	Implementation Notes	15
7	Indices and tables	17
	Python Module Index	19
	Index	21

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](#).

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-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  # SPDX-FileCopyrightText: 2021 ladyada for Adafruit Industries
2  # SPDX-License-Identifier: MIT
3
4  # Simple demo of the VL53L0X distance sensor.
5  # Will print the sensed range/distance every second.
6  import time
7
8  import board
9  import busio
10
11 import adafruit_vl53l0x
12
13 # Initialize I2C bus and sensor.
14 i2c = busio.I2C(board.SCL, board.SDA)
15 vl53 = adafruit_vl53l0x.VL53L0X(i2c)
16
17 # Optionally adjust the measurement timing budget to change speed and accuracy.
18 # See the example here for more details:
19 # https://github.com/pololu/vl53l0x-arduino/blob/master/examples/Single/Single.ino
20 # For example a higher speed but less accurate timing budget of 20ms:
21 # vl53.measurement_timing_budget = 20000
22 # Or a slower but more accurate timing budget of 200ms:
23 # vl53.measurement_timing_budget = 200000
24 # The default timing budget is 33ms, a good compromise of speed and accuracy.
25
26 # Main loop will read the range and print it every second.
27 while True:
```

(continues on next page)

(continued from previous page)

```

28 print("Range: {0}mm".format(vl53.range))
29 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  # SPDX-FileCopyrightText: 2021 ladyada for Adafruit Industries
2  # SPDX-License-Identifier: MIT
3
4  """
5  Example of how to use the adafruit_vl53l0x library to change the assigned address of
6  multiple VL53L0X sensors on the same I2C bus. This example only focuses on 2 VL53L0X
7  sensors, but can be modified for more. BE AWARE: a multitude of sensors may require
8  more current than the on-board 3V regulator can output (typical current consumption,
9  ↳during
10 active range readings is about 19 mA per sensor).
11 """
12 import time
13 import board
14 from digitalio import DigitalInOut
15 from adafruit_vl53l0x import VL53L0X
16
17 # declare the singleton variable for the default I2C bus
18 i2c = board.I2C()
19
20 # declare the digital output pins connected to the "SHDN" pin on each VL53L0X sensor
21 xshut = [
22     DigitalInOut(board.D7),
23     DigitalInOut(board.D9),
24     # add more VL53L0X sensors by defining their SHDN pins here
25 ]
26
27 for power_pin in xshut:
28     # make sure these pins are a digital output, not a digital input
29     power_pin.switch_to_output(value=False)
30     # These pins are active when Low, meaning:
31     #   if the output signal is LOW, then the VL53L0X sensor is off.
32     #   if the output signal is HIGH, then the VL53L0X sensor is on.
33 # all VL53L0X sensors are now off
34
35 # initialize a list to be used for the array of VL53L0X sensors
36 vl53 = []
37
38 # now change the addresses of the VL53L0X sensors
39 for i, power_pin in enumerate(xshut):
40     # turn on the VL53L0X to allow hardware check
41     power_pin.value = True
42     # instantiate the VL53L0X sensor on the I2C bus & insert it into the "vl53" list
43     vl53.insert(i, VL53L0X(i2c)) # also performs VL53L0X hardware check
44     # no need to change the address of the last VL53L0X sensor

```

(continues on next page)

(continued from previous page)

```

44     if i < len(xshut) - 1:
45         # default address is 0x29. Change that to something else
46         vl53[i].set_address(i + 0x30) # address assigned should NOT be already in use
47 # there is a helpful list of pre-designated I2C addresses for various I2C devices at
48 # https://learn.adafruit.com/i2c-addresses/the-list
49 # According to this list 0x30-0x34 are available, although the list may be incomplete.
50 # In the python REPR, you can scan for all I2C devices that are attached and determine
51 # their addresses using:
52 # >>> import board
53 # >>> i2c = board.I2C()
54 # >>> if i2c.try_lock():
55 # >>>     [hex(x) for x in i2c.scan()]
56 # >>>     i2c.unlock()
57
58
59 def detect_range(count=5):
60     """ take count=5 samples """
61     while count:
62         for index, sensor in enumerate(vl53):
63             print("Sensor {} Range: {}mm".format(index + 1, sensor.range))
64             time.sleep(1.0)
65             count -= 1
66
67
68 print(
69     "Multiple VL53L0X sensors' addresses are assigned properly\n"
70     "execute detect_range() to read each sensors range readings"
71 )

```

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

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*),
16

signal_rate_limit (*adafruit_vl53l0x.VL53L0X at-
tribute*), 16

V

VL53L0X (*class in adafruit_vl53l0x*), 15