Adafruit VL53L0X Library Documentation

Release 1.0

Tony DiCola

Apr 10, 2020

Contents

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-vl53l0x
```

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

```
sudo pip3 install adafruit-circuitpython-vl53l0x
```

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-vl53l0x
```

CHAPTER 3

Usage Example

See usage in the examples/vl53l0x_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.

CHAPTER 6

Table of Contents

6.1 Simple test

Ensure your device works with this simple test.

Listing 1: examples/vl53l0x_simpletest.py

```
# Simple demo of the VL53L0X distance sensor.
# Will print the sensed range/distance every second.
import time

import board
import busio

import adafruit_vl53l0x

# Initialize I2C bus and sensor.
i2c = busio.I2C(board.SCL, board.SDA)
vl53 = adafruit_vl53l0x.VL53L0X(i2c)

# Optionally adjust the measurement timing budget to change speed and accuracy.
# See the example here for more details:
#   https://github.com/pololu/vl53l0x-arduino/blob/master/examples/Single/Single.ino
# For example a higher speed but less accurate timing budget of 20ms:
# vl53.measurement_timing_budget = 20000
# Or a slower but more accurate timing budget of 200ms:
# vl53.measurement_timing_budget = 200000
# The default timing budget is 33ms, a good compromise of speed and accuracy.

# Main loop will read the range and print it every second.
while True:
    print("Range: {0}mm".format(vl53.range))
    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

```
"""
Example of how to use the adafruit_vl53l0x library to change the assigned address of
multiple VL53L0X sensors on the same I2C bus. This example only focuses on 2 VL53L0X
sensors, but can be modified for more. BE AWARE: a multitude of sensors may require
more current than the on-board 3V regulator can output (typical current consumption␣
↪during
active range readings is about 19 mA per sensor).
"""
import time
import board
from digitalio import DigitalInOut
from adafruit_vl53l0x import VL53L0X

# declare the singleton variable for the default I2C bus
i2c = board.I2C()

# declare the digital output pins connected to the "SHDN" pin on each VL53L0X sensor
xshut = [
    DigitalInOut(board.D7),
    DigitalInOut(board.D9),
    # add more VL53L0X sensors by defining their SHDN pins here
]

for power_pin in xshut:
    # make sure these pins are a digital output, not a digital input
    power_pin.switch_to_output(value=False)
    # These pins are active when Low, meaning:
    #   if the output signal is LOW, then the VL53L0X sensor is off.
    #   if the output signal is HIGH, then the VL53L0X sensor is on.
# all VL53L0X sensors are now off

# initialize a list to be used for the array of VL53L0X sensors
vl53 = []

# now change the addresses of the VL53L0X sensors
for i, power_pin in enumerate(xshut):
    # turn on the VL53L0X to allow hardware check
    power_pin.value = True
    # instantiate the VL53L0X sensor on the I2C bus & insert it into the "vl53" list
    vl53.insert(i, VL53L0X(i2c))  # also performs VL53L0X hardware check
    # no need to change the address of the last VL53L0X sensor
    if i < len(xshut) - 1:
        # default address is 0x29. Change that to something else
        vl53[i].set_address(i + 0x30)  # address assigned should NOT be already in use
# there is a helpful list of pre-designated I2C addresses for various I2C devices at
# https://learn.adafruit.com/i2c-addresses/the-list
# According to this list 0x30-0x34 are available, although the list may be incomplete.
# In the python REPR, you can scan for all I2C devices that are attached and detirmine
# their addresses using:
#   >>> import board
```

(continues on next page)

(continued from previous page)

```
#   >>> i2c = board.I2C()
#   >>> if i2c.try_lock():
#   >>>     [hex(x) for x in i2c.scan()]
#   >>>     i2c.unlock()


def detect_range(count=5):
    """ take count=5 samples """
    while count:
        for index, sensor in enumerate(vl53):
            print("Sensor {} Range: {}mm".format(index + 1, sensor.range))
        time.sleep(1.0)
        count -= 1


print(
    "Multiple VL53L0X sensors' addresses are assigned properly\n"
    "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

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 instantaited 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

Python Module Index

Index