
sparkfun*qwiiCVL53L1X*

Release 0.0.1

Jan 17, 2020

Contents:

1	Contents	3
2	Supported Platforms	5
3	Dependencies	7
4	Documentation	9
5	Installation	11
5.1	PyPi Installation	11
5.2	Local Installation	11
6	Example Use	13
7	Table of Contents	15
7.1	API Reference	15
7.2	Example 1: Basic Distance Measurement	21
7.3	Example 2: Set Sensor Distance Mode to Short	22
7.4	Example 3: Get Sensor Status and Sampling Rate, with Running Average	24
7.5	Example 4: Set Intermeasurement Period	26
	Python Module Index	29
	Index	31

Python module for the SparkFun Distance Sensor Breakout - 4 Meter, VL53L1X (Qwiic).

This package should be used in conjunction with the overall SparkFun qwiic Python Package. New to qwiic? Take a look at the entire SparkFun qwiic ecosystem.

CHAPTER 1

Contents

- *Supported Platforms*
- *Dependencies*
- *Installation*
- *Documentation*
- *Example Use*

CHAPTER 2

Supported Platforms

The qwiic VL53L1X Python package current supports the following platforms:

- Raspberry Pi <!-- Platforms to be tested
- NVidia Jetson Nano
- Google Coral Development Board ->

CHAPTER 3

Dependencies

This package depends on the qwiic I2C driver: [Qwiic_I2C_Py](#)

CHAPTER 4

Documentation

The SparkFun qwiic VL53L1X module documentation is hosted at [ReadTheDocs](#)

5.1 PyPi Installation

This repository is hosted on PyPi as the `sparkfun-qwiic-v15311x` package. On systems that support PyPi installation via `pip`, this library is installed using the following commands

For all users (note: the user must have `sudo` privileges):

```
sudo pip install sparkfun-qwiic-v15311x
```

For the current user:

```
pip install sparkfun-qwiic-v15311x
```

5.2 Local Installation

To install, make sure the `setuptools` package is installed on the system.

Direct installation at the command line:

```
python setup.py install
```

To build a package for use with `pip`:

```
python setup.py sdist
```

A package file is built and placed in a subdirectory called `dist`. This package file can be installed using `pip`.

```
cd dist  
pip install sparkfun_qwiic_v15311x-<version>.tar.gz
```


CHAPTER 6

Example Use

See the examples directory for more detailed use examples.

```
import qwiic_vl53l1x
import time
import sys

def runExample():

    print("\nSparkFun VL53L1X Example 1\n")
    mySensor = qwiic_vl53l1x.QwiicVL53L1X()

    if mySensor.isConnected() == False:
        print("The Qwiic VL53L1X device isn't connected to the system. Please check_
↳your connection", \
            file=sys.stderr)
        return

    mySensor.sensor_init()

    while True:
        try:
            mySensor.start_ranging()                # Write configuration_
↳bytes to initiate measurement
            time.sleep(.005)
            distance = mySensor.get_distance()     # Get the result of the measurement_
↳from the sensor
            time.sleep(.005)
            mySensor.stop_ranging()

            print("Distance(mm): %s" % distance)

        except Exception as e:
            print(e)
```


7.1 API Reference

class `qwiiic_vl53l1x.QwiiicVL53L1X` (*address=None, debug=None, i2c_driver=None*)
SparkFunVL53L1X Initialise the VL53L1X chip at address with `i2c_driver`.

Parameters

- **address** – The I2C address to use for the device.
 - If not provided, the default address is used.
- **i2c_driver** – An existing i2c driver object.
 - If not provided a driver object is created.

Returns

Constructor Initialization -

- True- Successful
- False- Issue loading I2C driver

Return type Bool

boot_state ()

This function returns the boot state of the device (1:booted, 0:not booted)

Returns

- 1- booted
- 0- not booted

Return type Integer

calibrate_offset (*TargetDistInMm*)

This function performs the offset calibration. The function returns the offset value found and programs the offset compensation into the device.

Parameters TargetDistInMm –

- Target distance in mm, ST recommended 100 mm
- Target reflectance = grey17%

Return status

- 0- success
- !=0- failed

Return offset offset found in mm #?

calibrate_xtalk (*TargetDistInMm*)

This function performs the xtalk calibration. The function returns the xtalk value found and programs the xtalk compensation to the device

Parameters TargetDistInMm – Target distance in mm

- **The target distance** [the distance where the sensor start to “under range” due to the influence of the photons reflected back from the cover glass becoming strong (also called the inflection point).]
 - Target reflectance = grey 17%

Return status

- 0- success
- !=0- failed

Return xtalk xtalk value found in cps (number of photons in count per second) #?

check_for_data_ready ()

This function checks if the new ranging data is available by polling the dedicated register.

Return isDataReady

- 0 -> not ready
- 1 -> ready

clear_interrupt ()

This function clears the interrupt, to be called after a ranging data reading to arm the interrupt for the next data ready event.

get_ambient_per_spad ()

This function returns the ambient per SPAD in kcps/SPAD

Returns Ambient per SPAD

get_ambient_rate ()

This function returns the ambient rate in kcps

Returns Ambient rate in kcps

get_distance ()

This function returns the distance measured by the sensor in mm

Returns Distance measured by the sensor in mm

Return type Integer

get_distance_mode ()

This function returns the current distance mode (1=short, 2=long).

Returns

- 1- Short mode max distance is limited to 1.3 m but better ambient immunity.
- 2- Long mode can range up to 4 m in the dark with 200 ms timing budget (**default**).

get_distance_threshold_high()

This function returns the high threshold in mm

Returns High threshold in mm

Return type Integer

get_distance_threshold_low()

This function returns the low threshold in mm

Returns Low threshold in mm

Return type Integer

get_distance_threshold_window()

This function returns the window detection mode (0=below 1=above 2=out 3=in)

Returns

Window detection mode:

- 0- below
- 1- above
- 2- out
- 3- in

Return type Integer

get_inter_measurement_in_ms()

This function returns the Intermeasurement period in ms.

Returns Intermeasurement period in ms

Return type Integer

get_interrupt_polarity()

This function returns the current interrupt polarity

Returns

- 1 = active high (**default**)
- 0 = active low

Rtype Integer

get_offset()

This function returns the programmed offset correction value in mm

Returns Offset correction value in mm

Return type Integer

get_range_status()

This function returns the ranging status error

Returns

Ranging status error

- 0- no error

- 1- sigma failed
- 2- signal failed
- 7- wrap-around

get_roi_xy()

This function returns width X and height Y

Returns Region of Interest Width (X) and Height (Y)

Return type List

get_sensor_id()

This function returns the sensor id, sensor Id must be 0xEEAC

Returns Sensor ID

Return type Integer

get_sigma_threshold()

This function returns the current sigma threshold in mm

Returns Sigma threshold in mm

Return type Integer

get_signal_per_spad()

This function returns the returned signal per SPAD in kcps/SPAD (kcps stands for Kilo Count Per Second).

Returns Signal per SPAD (Kilo Count Per Second/SPAD).

get_signal_rate()

This function returns the returned signal in kcps.

Returns signal in kcps

get_signal_threshold()

This function returns the current signal threshold in kcps

Returns Signal threshold in kcps

get_spad_nb()

This function returns the current number of enabled SPADs

Returns Number of enabled SPADs

get_sw_version()

This function returns the SW driver version

Returns [major, minor, build, revision] numbers

Return type List

get_timing_budget_in_ms()

This function returns the current timing budget in ms.

get_xtalk()

This function returns the current programmed xtalk correction value in cps

Returns xtalk correction value in cps

init_sensor(address)

Initialize the sensor with default values

Parameters address – Device address

Returns 0 on Success

sensor_init ()

This function loads the 135 bytes default values to initialize the sensor.

Returns

- 0:success
- != 0:failed

set_distance_mode (DM)

This function programs the distance mode (1=short, 2=long(default)).

Parameters DM –

- 1- Short mode max distance is limited to 1.3 m but better ambient immunity.
- 2- Long mode can range up to 4 m in the dark with 200 ms timing budget (**default**).

set_distance_threshold (ThreshLow, ThreshHigh, Window, IntOnNoTarget)

This function programs the threshold detection mode Example:

- self.set_distance_threshold(100,300,0,1): Below 100
- self.set_distance_threshold(100,300,1,1): Above 300
- self.set_distance_threshold(100,300,2,1): Out of window
- self.set_distance_threshold(100,300,3,1): In window

Parameters

- **ThreshLow** (*mm*) – The threshold under which one the device raises an interrupt if Window = 0
- **ThreshHigh** (*mm*) – The threshold above which one the device raises an interrupt if Window = 1
- **Window** – Window detection mode:
 - 0- below
 - 1- above
 - 2- out
 - 3- in
- **IntOnNoTarget** – = 1 (*No longer used - just set to 1*)

set_i2c_address (new_address)

This function sets the sensor I2C address used in case multiple devices application, default address **0x29** (0x52 >> 1)

Parameters new_address – I2C address to change device to

set_inter_measurement_in_ms (InterMeasMs)

This function programs the Intermeasurement period in ms.

Parameters InterMeasMs – Intermeasurement period must be >= timing budget. This condition is not checked by the API, the customer has the duty to check the condition.
Default = 100 ms

set_interrupt_polarity (NewPolarity)

This function programs the interrupt polarity

Parameters NewPolarity –

- 1 = active high (**default**)
- 0 = active low

set_offset (*OffsetValue*)

This function programs the offset correction in mm

Parameters OffsetValue – The offset correction value to program in mm

set_roi (*X, Y, OpticalCenter=199*)

This function programs the ROI (Region of Interest). The height and width of the ROI (X, Y) are set in SPADs; the smallest acceptable ROI size = 4 (4 x 4). The optical center is set based on table below. To set the center, use the pad that is to the right and above (i.e. upper right of) the exact center of the region you'd like to measure as your optical center.

Table of Optical Centers:

128,136,144,152,160,168,176,184,	192,200,208,216,224,232,240,248
129,137,145,153,161,169,177,185,	193,201,209,217,225,233,241,249
130,138,146,154,162,170,178,186,	194,202,210,218,226,234,242,250
131,139,147,155,163,171,179,187,	195,203,211,219,227,235,243,251
132,140,148,156,164,172,180,188,	196,204,212,220,228,236,244,252
133,141,149,157,165,173,181,189,	197,205,213,221,229,237,245,253
134,142,150,158,166,174,182,190,	198,206,214,222,230,238,246,254
135,143,151,159,167,175,183,191, 199,207,215,223,231,239,247,255	
127,119,111,103,095,087,079,071,	063,055,047,039,031,023,015,007
126,118,110,102,094,086,078,070,	062,054,046,038,030,022,014,006
125,117,109,101,093,085,077,069,	061,053,045,037,029,021,013,005
124,116,108,100,092,084,076,068,	060,052,044,036,028,020,012,004
123,115,107,099,091,083,075,067,	059,051,043,035,027,019,011,003
122,114,106,098,090,082,074,066,	058,050,042,034,026,018,010,002
121,113,105,097,089,081,073,065,	057,049,041,033,025,017,009,001
120,112,104,096,088,080,072,064, 056,048,040,032,024,016,008,0 Pin 1	

(Each SPAD has a number which is not obvious.)

Parameters

- **x** – ROI Width
- **y** – ROI Height
- **OpticalCenter** – The pad that is to the upper right of the exact center of the ROI (see table above). **Default = 199**

set_sigma_threshold (*Sigma*)

This function programs a new sigma threshold in mm (default=15 mm)

Parameters Sigma – Sigma threshold in mm (**default=15 mm**)

set_signal_threshold (*Signal*)

This function programs a new signal threshold in kcps (default=1024 kcps)

Parameters Signal – Signal threshold in kcps (**default=1024 kcps**)

set_timing_budget_in_ms (*TimingBudgetInMs*)

This function programs the timing budget in ms.

Parameters TimingBudgetInMs – Predefined values = 15, 20, 33, 50, 100 (**default**), 200, 500.

set_xtalk (*XtalkValue*)

This function programs the xtalk correction value in cps (Count Per Second). This is the number of photons reflected back from the cover glass in cps.

Parameters **XTalkValue** – xtalk correction value in count per second to avoid float type

start_ranging ()

This function starts the ranging distance operation The ranging operation is continuous. The clear interrupt has to be done after each get data to allow the interrupt to raise when the next data is ready 1=active high (**default**), 0=active low, use `set_interrupt_polarity()` to change the interrupt polarity if required.

start_temperature_update ()

This function performs the temperature calibration. It is recommended to call this function any time the temperature might have changed by more than 8 deg C without sensor ranging activity for an extended period.

stop_ranging ()

This function stops the ranging.

7.2 Example 1: Basic Distance Measurement

Listing 1: examples/Example1_ReadDistance.py

```

1  #-----
2  # VL53L1X - Example 1
3  #-----
4  #
5  # Ported by SparkFun Electronics, October 2019
6  # Author: Nathan Seidle
7  # Ported: Wes Furuya
8  # SparkFun Electronics
9  #
10 # License: This code is public domain but you buy me a beer if you use
11 # this and we meet someday (Beerware license).
12 #
13 # Compatibility: https://www.sparkfun.com/products/14722
14 #
15 # Do you like this library? Help support SparkFun. Buy a board!
16 # For more information on VL53L1x (ToF), check out the product page
17 # linked above.
18 #
19 # This program is distributed in the hope that it will be useful, but
20 # WITHOUT ANY WARRANTY without even the implied warranty of
21 # MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
22 # General Public License for more details.
23 #
24 # You should have received a copy of the GNU General Public License
25 # along with this program. If not, see <http://www.gnu.org/licenses/>.
26 #
27 #=====
28 # Copyright (c) 2019 SparkFun Electronics
29 #
30 # Permission is hereby granted, free of charge, to any person obtaining
31 # a copy of this software and associated documentation files (the
32 # "Software"), to deal in the Software without restriction, including
33 # without limitation the rights to use, copy, modify, merge, publish,

```

(continues on next page)

(continued from previous page)

```

34 # distribute, sublicense, and/or sell copies of the Software, and to
35 # permit persons to whom the Software is furnished to do so, subject to
36 # the following conditions:
37 #
38 # The above copyright notice and this permission notice shall be
39 # included in all copies or substantial portions of the Software.
40 #
41 # THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
42 # EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
43 # MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
44 # IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
45 # CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
46 # TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
47 # SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
48 #=====
49
50 """
51     Reading distance from the laser based VL53L1X
52
53     This example prints the distance to an object. If you are getting weird
54     readings, be sure the vacuum tape has been removed from the sensor.
55 """
56
57 import qwiic
58 import time
59
60 print("VL53L1X Qwiic Test\n")
61 ToF = qwiic.QwiicVL53L1X()
62 if (ToF.sensor_init() == None):                                     # Begin
63     ↪returns 0 on a good init
64     print("Sensor online!\n")
65
66 while True:
67     try:
68         ToF.start_ranging()                                         #
69         ↪Write configuration bytes to initiate measurement
70         time.sleep(.005)
71         distance = ToF.get_distance()                               # Get the result of the
72         ↪measurement from the sensor
73         time.sleep(.005)
74         ToF.stop_ranging()
75
76         distanceInches = distance / 25.4
77         distanceFeet = distanceInches / 12.0
78
79         print("Distance(mm): %s Distance(ft): %s" % (distance, distanceFeet))
80
81     except Exception as e:
82         print(e)

```

7.3 Example 2: Set Sensor Distance Mode to Short

Listing 2: examples/Example2_SetDistanceMode.py

```

1  #-----
2  # VL53L1X - Example 2
3  #-----
4  #
5  # Ported by SparkFun Electronics, October 2019
6  # Author: Nathan Seidle
7  # Ported: Wes Furuya
8  # SparkFun Electronics
9  #
10 # License: This code is public domain but you buy me a beer if you use
11 # this and we meet someday (Beerware license).
12 #
13 # Compatibility: https://www.sparkfun.com/products/14722
14 #
15 # Do you like this library? Help support SparkFun. Buy a board!
16 # For more information on VL53L1x (ToF), check out the product page
17 # linked above.
18 #
19 # This program is distributed in the hope that it will be useful, but
20 # WITHOUT ANY WARRANTY without even the implied warranty of
21 # MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
22 # General Public License for more details.
23 #
24 # You should have received a copy of the GNU General Public License
25 # along with this program. If not, see <http://www.gnu.org/licenses/>.
26 #
27 #=====
28 # Copyright (c) 2019 SparkFun Electronics
29 #
30 # Permission is hereby granted, free of charge, to any person obtaining
31 # a copy of this software and associated documentation files (the
32 # "Software"), to deal in the Software without restriction, including
33 # without limitation the rights to use, copy, modify, merge, publish,
34 # distribute, sublicense, and/or sell copies of the Software, and to
35 # permit persons to whom the Software is furnished to do so, subject to
36 # the following conditions:
37 #
38 # The above copyright notice and this permission notice shall be
39 # included in all copies or substantial portions of the Software.
40 #
41 # THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
42 # EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
43 # MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
44 # IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
45 # CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
46 # TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
47 # SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
48 #=====
49
50 """
51     Reading distance from the laser based VL53L1X
52
53     This example configures the sensor to short distance mode and then
54     prints the distance to an object. If you are getting weird readings,
55     be sure the vacuum tape has been removed from the sensor.

```

(continues on next page)

(continued from previous page)

```

56 """
57
58 import qwiic
59 import time
60
61 print("VL53L1X Qwiic Test\n")
62 ToF = qwiic.QwiicVL53L1X()
63
64 if (ToF.sensor_init() == None):                                # Begin
65     ↪returns 0 on a good init
66     print("Sensor online!\n")
67
68 ToF.set_distance_mode(1)          # Sets Distance Mode Short (Long- Change value to 2)
69
70 while True:
71     try:
72         ToF.start_ranging()                                     #
73         ↪Write configuration bytes to initiate measurement
74         time.sleep(.005)
75         distance = ToF.get_distance()                          # Get the result of the
76         ↪measurement from the sensor
77         time.sleep(.005)
78         ToF.stop_ranging()
79
80         distanceInches = distance / 25.4
81         distanceFeet = distanceInches / 12.0
82
83         print("Distance(mm): %s Distance(ft): %s" % (distance, distanceFeet))
84
85     except Exception as e:
86         print(e)

```

7.4 Example 3: Get Sensor Status and Sampling Rate, with Running Average

Listing 3: examples/Example3_StatusandRate.py

```

1 #-----
2 # VL53L1X - Example 3
3 #-----
4 #
5 # Ported by SparkFun Electronics, October 2019
6 # Author: Nathan Seidle
7 # Ported: Wes Furuya
8 # SparkFun Electronics
9 #
10 # License: This code is public domain but you buy me a beer if you use
11 # this and we meet someday (Beerware license).
12 #
13 # Compatibility: https://www.sparkfun.com/products/14722
14 #
15 # Do you like this library? Help support SparkFun. Buy a board!
16 # For more information on VL53L1x (ToF), check out the product page

```

(continues on next page)

(continued from previous page)

```

17 # linked above.
18 #
19 # This program is distributed in the hope that it will be useful, but
20 # WITHOUT ANY WARRANTY without even the implied warranty of
21 # MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
22 # General Public License for more details.
23 #
24 # You should have received a copy of the GNU General Public License
25 # along with this program. If not, see <http://www.gnu.org/licenses/>.
26 #
27 #=====
28 # Copyright (c) 2019 SparkFun Electronics
29 #
30 # Permission is hereby granted, free of charge, to any person obtaining
31 # a copy of this software and associated documentation files (the
32 # "Software"), to deal in the Software without restriction, including
33 # without limitation the rights to use, copy, modify, merge, publish,
34 # distribute, sublicense, and/or sell copies of the Software, and to
35 # permit persons to whom the Software is furnished to do so, subject to
36 # the following conditions:
37 #
38 # The above copyright notice and this permission notice shall be
39 # included in all copies or substantial portions of the Software.
40 #
41 # THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
42 # EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
43 # MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
44 # IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
45 # CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
46 # TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
47 # SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
48 #=====
49
50 """
51     Reading distance from the laser based VL53L1X
52
53     This example configures the sensor to short distance mode and then
54     prints the distance to an object. The output also includes a
55     running average of the last 10 readings along with some statistics,
56     like signal rate and frequency (of the measurements).
57
58     If you are getting weird readings, be sure the vacuum tape has been
59     removed from the sensor.
60 """
61
62 import qwiic
63 import time, statistics
64
65 print("VL53L1X Qwiic Test\n")
66 ToF = qwiic.QwiicVL53L1X()
67
68 if (ToF.sensor_init() == None):                                     # Begin_
69     ↪returns 0 on a good init
70     print("Sensor online!\n")
71
72 ToF.set_distance_mode(1)          # Sets Distance Mode Short (Long- Change value to 2)

```

(continues on next page)

(continued from previous page)

```

73 distance = [] # Initialize list
74
75 while True:
76     start = time.time()
77
78     try:
79         ToF.start_
↳ranging() #
↳Write configuration bytes to initiate measurement
80         time.sleep(.005)
81         distance.append(ToF.get_distance()) # Get the result of the_
↳measurement from the sensor
82         time.sleep(.005)
83         ToF.stop_ranging()
84
85     except Exception as e:
86         print(e)
87
88     end = time.time()
89
90     distanceInches = distance[len(distance)-1] / 25.4
91     distanceFeet = distanceInches / 12.0
92
93     if len(distance) < 10:
94         avgdistance = statistics.mean(distance)
95     else:
96         distance.remove(distance[0])
97         avgdistance = statistics.mean(distance[len(distance)-
↳10:len(distance)+1]) # Running average of last 10 measurements
98
99     signalrate = ToF.get_signal_rate()
100     rangestatus = ToF.get_range_status()
101
102     #print("Distance(mm): %s avgDistance(mm): %s Distance(ft): %.3f Signal Rate:
↳%s Range Status: %s Hz: %.5f" % (distance[len(distance)-1], avgdistance,
↳distanceFeet, signalrate, rangestatus, (end-start)))
103     print("Distance(mm): %s avgDistance(mm): %.2f Signal Rate: %s Range Status:
↳%s Hz: %.5f" % (distance[len(distance)-1], avgdistance, signalrate, rangestatus,
↳(end-start)))
104
105
106     #if rangestatus == 0:
107         #print("Good ")
108     #elif rangestatus == 1:
109         #print("Signal Fail ")
110     #elif rangestatus == 2:
111         #print("Sigma Fail ")
112     #elif rangestatus == 7:
113         #print("Wrapped Target Fail ")
114     #else:
115         #print("Unknown (code: %s) ", rangestatus)

```

7.5 Example 4: Set Intermeasurement Period

Listing 4: examples/Example4_SetIntermeasurementPeriod.py

```

1  #-----
2  # VL53L1X - Example 4
3  #-----
4  #
5  # Ported by SparkFun Electronics, October 2019
6  # Author: Nathan Seidle
7  # Ported: Wes Furuya
8  # SparkFun Electronics
9  #
10 # License: This code is public domain but you buy me a beer if you use
11 # this and we meet someday (Beerware license).
12 #
13 # Compatibility: https://www.sparkfun.com/products/14722
14 #
15 # Do you like this library? Help support SparkFun. Buy a board!
16 # For more information on VL53L1x (ToF), check out the product page
17 # linked above.
18 #
19 # This program is distributed in the hope that it will be useful, but
20 # WITHOUT ANY WARRANTY without even the implied warranty of
21 # MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
22 # General Public License for more details.
23 #
24 # You should have received a copy of the GNU General Public License
25 # along with this program. If not, see <http://www.gnu.org/licenses/>.
26 #
27 #=====
28 # Copyright (c) 2019 SparkFun Electronics
29 #
30 # Permission is hereby granted, free of charge, to any person obtaining
31 # a copy of this software and associated documentation files (the
32 # "Software"), to deal in the Software without restriction, including
33 # without limitation the rights to use, copy, modify, merge, publish,
34 # distribute, sublicense, and/or sell copies of the Software, and to
35 # permit persons to whom the Software is furnished to do so, subject to
36 # the following conditions:
37 #
38 # The above copyright notice and this permission notice shall be
39 # included in all copies or substantial portions of the Software.
40 #
41 # THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
42 # EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
43 # MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
44 # IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
45 # CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
46 # TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
47 # SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
48 #=====
49
50 """
51     Reading distance from the laser based VL53L1X
52
53     This example configures the inter-measurement period of the sensor
54     and then prints the distance to an object. If you are getting weird
55     readings, be sure the vacuum tape has been removed from the sensor.

```

(continues on next page)

(continued from previous page)

```
56 """
57
58 import qwiic
59 import time
60
61 print("VL53L1X Qwiic Test\n")
62 ToF = qwiic.QwiicVL53L1X()
63
64 if (ToF.sensor_init() == None):                                # Begin
65     ↪returns 0 on a good init
66     print("Sensor online!\n")
67
68 ToF.set_inter_measurement_in_ms(40)
69
70 print("Inter Measurement Period (ms): %s \n", ToF.get_inter_measurement_in_ms())
71
72 while True:
73     try:
74         ToF.start_ranging()                                    #
75         ↪Write configuration bytes to initiate measurement
76         time.sleep(.005)
77         distance = ToF.get_distance()                          # Get the result of the
78         ↪measurement from the sensor
79         time.sleep(.005)
80         ToF.stop_ranging()
81
82         distanceInches = distance / 25.4
83         distanceFeet = distanceInches / 12.0
84
85         print("Distance(mm): %s Distance(ft): %s" % (distance, distanceFeet))
86
87     except Exception as e:
88         print(e)
```

q

`qwiic_vl53l1x`, 15

B

`boot_state()` (*qwiic_vl53l1x.QwiicVL53L1X* method), 15

C

`calibrate_offset()` (*qwiic_vl53l1x.QwiicVL53L1X* method), 15

`calibrate_xtalk()` (*qwiic_vl53l1x.QwiicVL53L1X* method), 16

`check_for_data_ready()` (*qwiic_vl53l1x.QwiicVL53L1X* method), 16

`clear_interrupt()` (*qwiic_vl53l1x.QwiicVL53L1X* method), 16

G

`get_ambient_per_spad()` (*qwiic_vl53l1x.QwiicVL53L1X* method), 16

`get_ambient_rate()` (*qwiic_vl53l1x.QwiicVL53L1X* method), 16

`get_distance()` (*qwiic_vl53l1x.QwiicVL53L1X* method), 16

`get_distance_mode()` (*qwiic_vl53l1x.QwiicVL53L1X* method), 16

`get_distance_threshold_high()` (*qwiic_vl53l1x.QwiicVL53L1X* method), 17

`get_distance_threshold_low()` (*qwiic_vl53l1x.QwiicVL53L1X* method), 17

`get_distance_threshold_window()` (*qwiic_vl53l1x.QwiicVL53L1X* method), 17

`get_inter_measurement_in_ms()` (*qwiic_vl53l1x.QwiicVL53L1X* method),

17

`get_interrupt_polarity()` (*qwiic_vl53l1x.QwiicVL53L1X* method), 17

`get_offset()` (*qwiic_vl53l1x.QwiicVL53L1X* method), 17

`get_range_status()` (*qwiic_vl53l1x.QwiicVL53L1X* method), 17

`get_roi_xy()` (*qwiic_vl53l1x.QwiicVL53L1X* method), 18

`get_sensor_id()` (*qwiic_vl53l1x.QwiicVL53L1X* method), 18

`get_sigma_threshold()` (*qwiic_vl53l1x.QwiicVL53L1X* method), 18

`get_signal_per_spad()` (*qwiic_vl53l1x.QwiicVL53L1X* method), 18

`get_signal_rate()` (*qwiic_vl53l1x.QwiicVL53L1X* method), 18

`get_signal_threshold()` (*qwiic_vl53l1x.QwiicVL53L1X* method), 18

`get_spad_nb()` (*qwiic_vl53l1x.QwiicVL53L1X* method), 18

`get_sw_version()` (*qwiic_vl53l1x.QwiicVL53L1X* method), 18

`get_timing_budget_in_ms()` (*qwiic_vl53l1x.QwiicVL53L1X* method), 18

`get_xtalk()` (*qwiic_vl53l1x.QwiicVL53L1X* method), 18

I

`init_sensor()` (*qwiic_vl53l1x.QwiicVL53L1X* method), 18

Q

`qwiic_vl53l1x` (module), 15

QwiicVL53L1X (class in qwiic_vl53l1x), 15

S

sensor_init() (qwiic_vl53l1x.QwiicVL53L1X method), 18

set_distance_mode() (qwiic_vl53l1x.QwiicVL53L1X method), 19

set_distance_threshold() (qwiic_vl53l1x.QwiicVL53L1X method), 19

set_i2c_address() (qwiic_vl53l1x.QwiicVL53L1X method), 19

set_inter_measurement_in_ms() (qwiic_vl53l1x.QwiicVL53L1X method), 19

set_interrupt_polarity() (qwiic_vl53l1x.QwiicVL53L1X method), 19

set_offset() (qwiic_vl53l1x.QwiicVL53L1X method), 20

set_roi() (qwiic_vl53l1x.QwiicVL53L1X method), 20

set_sigma_threshold() (qwiic_vl53l1x.QwiicVL53L1X method), 20

set_signal_threshold() (qwiic_vl53l1x.QwiicVL53L1X method), 20

set_timing_budget_in_ms() (qwiic_vl53l1x.QwiicVL53L1X method), 20

set_xtalk() (qwiic_vl53l1x.QwiicVL53L1X method), 20

start_ranging() (qwiic_vl53l1x.QwiicVL53L1X method), 21

start_temperature_update() (qwiic_vl53l1x.QwiicVL53L1X method), 21

stop_ranging() (qwiic_vl53l1x.QwiicVL53L1X method), 21