

---

# **sparkfun***<sub>q</sub>wiic<sub>p</sub>proximity*

***Release 0.9.0***

**Jul 15, 2019**



---

## Contents:

---

<b>1</b>	<b>Contents</b>	<b>3</b>
<b>2</b>	<b>Supported Platforms</b>	<b>5</b>
<b>3</b>	<b>Dependencies</b>	<b>7</b>
<b>4</b>	<b>Documentation</b>	<b>9</b>
4.1	PyPi Installation . . . . .	9
4.2	Local Installation . . . . .	9
<b>5</b>	<b>Example Use</b>	<b>11</b>
<b>6</b>	<b>Table of Contents</b>	<b>13</b>
6.1	API Reference . . . . .	13
6.1.1	qwiic_proximity . . . . .	13
6.2	Example 1 . . . . .	18
6.3	Example 2 . . . . .	19
6.4	Example 3 . . . . .	21
6.5	Example 4 . . . . .	23
6.6	Example 5 . . . . .	24
6.7	Example 8 . . . . .	26
<b>7</b>	<b>Indices and tables</b>	<b>29</b>
	<b>Python Module Index</b>	<b>31</b>
	<b>Index</b>	<b>33</b>



Python module to interface with the [Qwiic Proximity](#) board.

This package is a port of the [SparkFun VCNL4040 Proximity Sensor Arduino Library](#)

This package can 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 Proximity Python package current supports the following platforms:

- [Raspberry Pi](#)
- [NVidia Jetson Nano](#)
- [Google Coral Development Board](#)



## CHAPTER 3

---

### Dependencies

---

This driver package depends on the qwiic I2C driver: [Qwiic\\_I2C\\_Py](#)



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

### 4.1 PyPi Installation

This repository is hosted on PyPi as the [sparkfun-qwiic-proximity](#) 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-proximity
```

For the current user:

```
pip install sparkfun-qwiic-proximity
```

### 4.2 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_proximity-<version>.tar.gz
```



## CHAPTER 5

---

### Example Use

---

See the examples directory for more detailed use examples.

```
import qwiic_proximity
import time
import sys

def runExample():

    print("\nSparkFun Proximity Sensor VCN4040 Example 1\n")
    oProx = qwiic_proximity.QwiicProximity()

    if oProx.isConnected() == False:
        print("The Qwiic Proximity device isn't connected to the system. Please check ↪
your connection", \
              file=sys.stderr)
        return

    oProx.begin()

    while True:
        proxValue = oProx.getProximity()
        print("Proximity Value: %d" % proxValue)
        time.sleep(.4)

runExample()
```





## 6.1 API Reference

### 6.1.1 qwiic\_proximity

Python module for the [SparkFun Qwiic Proximity Sensor Breakout](<https://www.sparkfun.com/products/15177>)

This python package is a port of the existing [SparkFun VCNL4040 Proximity Sensor Arduino Library]([https://github.com/sparkfun/SparkFun\\_VCNL4040\\_Arduino\\_Library](https://github.com/sparkfun/SparkFun_VCNL4040_Arduino_Library))

This package can be used in conjunction with the overall [SparkFun qwiic Python Package]([https://github.com/sparkfun/Qwiic\\_Py](https://github.com/sparkfun/Qwiic_Py))

New to qwiic? Take a look at the entire [SparkFun qwiic ecosystem](<https://www.sparkfun.com/qwiic>).

**class** qwiic\_proximity.QwiicProximity (address=None, i2c\_driver=None)

#### 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** The Proximity device object.

**Return type** Object

#### ambient

Read the Ambient light value

**Returns** The current ambient value value

**Return type** integer

#### begin ()

Initialize the operation of the Proximity module

**Returns** Returns true if the initialization was successful, otherwise False.

**Return type** bool

**connected**

Determine if a Proximity device is connected to the system..

**Returns** True if the device is connected, otherwise False.

**Return type** bool

**disable\_active\_force\_mode ()**

Disable active force mode

**Returns** No return value

**disable\_ambient\_interrupts ()**

Disable Ambient Interrupts

**Returns** No return value

**disable\_prox\_logic\_mode ()**

Disable the proximity detection logic output mode

**Returns** No return value

**disable\_smart\_persistence ()**

Disable smart persistence

**Returns** No return value

**disable\_white\_channel ()**

Disable the white measurement channel

**Returns** No return value

**enable\_active\_force\_mode ()**

Enable active force mode An extreme power saving way to use PS is to apply PS active force mode. Anytime host would like to request one proximity measurement, enable the active force mode. This triggers a single PS measurement, which can be read from the PS result registers. VCNL4040 stays in standby mode constantly.

**Returns** No return value

**enable\_ambient\_interrupts ()**

Enable Ambient Interrupts

**Returns** No return value

**enable\_prox\_logic\_mode ()**

Enable the proximity detection logic output mode When this mode is selected, the INT pin is pulled low when an object is close to the sensor (value is above high threshold) and is reset to high when the object moves away (value is below low threshold). Register: PS\_THDH / PS\_THDL define where these threshold levels are set.

**Returns** No return value

**enable\_smart\_persistence ()**

Enable smart persistence To accelerate the PS response time, smart persistence prevents the misjudgment of proximity sensing but also keeps a fast response time.

**Returns** No return value

**enable\_white\_channel ()**

Enable the white measurement channel

**Returns** No return value

**get\_ambient()**

Read the Ambient light value

**Returns** The current ambient value value**Return type** integer**get\_id()**

Read the sensor ID

**Returns** The sensor ID**Return type** integer**get\_proximity()**

Get the current proximity value

**Returns** The current proximity value**Return type** integer**get\_white()**

Read the White light value

**Returns** The current white value value**Return type** integer**is\_away**

Returns true if the prox value drops below the lower threshold

**Returns** True if away**Return type** boolean**is\_close**

Returns true if the prox value rises above the upper threshold

**Returns** True if close**Return type** boolean**is\_connected()**

Determine if a Proximity device is connected to the system..

**Returns** True if the device is connected, otherwise False.**Return type** bool**is\_dark**

Returns true if the prox value drops below the lower threshold

**Returns** True if dark**Return type** boolean**is\_light**

Returns true if the prox value rises above the upper threshold

**Returns** True if value light**Return type** boolean**power\_off\_ambient()**

Power off the ambient light sensing portion of the sensor

**Returns** No return value

**power\_off\_proximity()**

Power off the prox sensing portion of the device

**Returns** No return value

**power\_on\_ambient()**

Power on the ambient light sensing portion of the sensor

**Returns** No return value

**power\_on\_proximity()**

Power on the prox sensing portion of the device

**Returns** No return value

**proximity**

Get the current proximity value

**Returns** The current proximity value

**Return type** integer

**sensor\_id**

Read the sensor ID

**Returns** The sensor ID

**Return type** integer

**set\_als\_high\_threshold(threshold)**

Value that ALS must go above to trigger an interrupt

**Parameters threshold** – the new trigger threshold value for ALS

**Returns** No return value

**set\_als\_low\_threshold(threshold)**

Value that ALS must go below to trigger an interrupt

**Parameters threshold** – the new trigger threshold value for ALS

**Returns** No return value

**set\_ambient\_integration\_time(timeValue)**

Sets the integration time for the ambient light sensor

**Parameters timeValue** – The integration time

**Returns** No return value

**set\_ambient\_interrupt\_persistence(persValue)**

Set the Ambient interrupt persistence value The ALS persistence function (ALS\_PERS, 1, 2, 4, 8) helps to avoid false trigger of the ALS INT. It defines the amount of consecutive hits needed in order for a ALS interrupt event to be triggered.

**Parameters persValue** – The ambient interrupt persistence value

**Returns** No return value

**set\_ir\_dutycycle(dutyValue)**

Set the duty cycle of the IR LED. The higher the duty ratio, the faster the response time achieved with higher power consumption. For example, PS\_Duty = 1/320, peak IRED current = 100 mA, averaged current consumption is 100 mA/320 = 0.3125 mA.

**Parameters dutyValue** – The duty cycle value for the IR LED on the sensor

**Returns** No return value

**set\_led\_current** (*currentValue*)

Set the IR LED sink current to one of 8 settings

**Parameters** **currentValue** – The new current value. Valid values are VCNL4040\_LED\_50MA thru VCNL4040\_LED\_200MA at 25MA increments

**Returns** No return value

**set\_prox\_cancellation** (*cancelValue*)

Set the proximity sensing cancelation value - helps reduce cross talk with ambient light

**Parameters** **cancelValue** – the new cancelation value

**Returns** No return value

**set\_prox\_high\_threshold** (*threshold*)

Value that Proximity Sensing must go above to trigger an interrupt

**Parameters** **threshold** – The new Proximity High Value

**Returns** No return value

**set\_prox\_integration\_time** (*timeValue*)

Sets the integration time for the proximity sensor

**Parameters** **timeValue** – The integration time

**Returns** No return value

**set\_prox\_interrupt\_persistence** (*persValue*)

Set the Prox interrupt persistence value The PS persistence function (PS\_PERS, 1, 2, 3, 4) helps to avoid false trigger of the PS INT. It defines the amount of consecutive hits needed in order for a PS interrupt event to be triggered.

**Parameters** **persValue** – The persistence value

**Returns** No return value

**set\_prox\_interrupt\_type** (*interruptValue*)

Sets the proximity interrupt type

**Parameters** **interruptValue** – The interrupt type

**Returns** No return value

**set\_prox\_low\_threshold** (*threshold*)

Value that Proximity Sensing must go below to trigger an interrupt

**Parameters** **threshold** – The new Proximity Low Value

**Returns** No return value

**set\_prox\_resolution** (*resolutionValue*)

Sets the proximity resolution

**Parameters** **resolutionValue** – The resolution time

**Returns** No return value

**take\_single\_prox\_measurement** ()

Set trigger bit so sensor takes a force mode measurement and returns to standby

**Returns** No return value

**white\_light**

Read the White light value

**Returns** The current white value value

**Return type** integer

## 6.2 Example 1

Listing 1: examples/qwiic\_proximity\_ex1.py

```
1  #!/usr/bin/env python
2  #-----
3  # qwiic_proximity_ex1.py
4  #
5  # Simple Example for the Qwiic Proximity Device
6  #-----
7  #
8  # Written by SparkFun Electronics, May 2019
9  #
10 # This python library supports the SparkFun Electroncis qwiic
11 # qwiic sensor/board ecosystem on a Raspberry Pi (and compatable) single
12 # board computers.
13 #
14 # More information on qwiic is at https://www.sparkfun.com/qwiic
15 #
16 # Do you like this library? Help support SparkFun. Buy a board!
17 #
18 #=====
19 # Copyright (c) 2019 SparkFun Electronics
20 #
21 # Permission is hereby granted, free of charge, to any person obtaining a copy
22 # of this software and associated documentation files (the "Software"), to deal
23 # in the Software without restriction, including without limitation the rights
24 # to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
25 # copies of the Software, and to permit persons to whom the Software is
26 # furnished to do so, subject to the following conditions:
27 #
28 # The above copyright notice and this permission notice shall be included in all
29 # copies or substantial portions of the Software.
30 #
31 # THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
32 # IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
33 # FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
34 # AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
35 # LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
36 # OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
37 # SOFTWARE.
38 #=====
39 # Example 1
40 #
41 # - Setup the device
42 # - Output the proximity value
43
44 from __future__ import print_function
45 import qwiic_proximity
46 import time
47 import sys
48
```

(continues on next page)

(continued from previous page)

```

49 def runExample():
50
51     print("\nSparkFun Proximity Sensor VCN4040 Example 1\n")
52     oProx = qwiic_proximity.QwiicProximity()
53
54     if oProx.connected == False:
55         print("The Qwiic Proximity device isn't connected to the system.
56 ↪Please check your connection", \
57             file=sys.stderr)
58         return
59
60     oProx.begin()
61
62     while True:
63         proxValue = oProx.get_proximity()
64         print("Proximity Value: %d" % proxValue)
65         time.sleep(.4)
66
67 if __name__ == '__main__':
68     try:
69         runExample()
70     except (KeyboardInterrupt, SystemExit) as exErr:
71         print("\nEnding Example 1")
72         sys.exit(0)
73
74

```

## 6.3 Example 2

Listing 2: examples/qwiic\_proximity\_ex2.py

```

1  #!/usr/bin/env python
2  #-----
3  # qwiic_proximity_ex2.py
4  #
5  # Simple Example for the Qwiic Proximity Device
6  #-----
7  #
8  # Written by SparkFun Electronics, May 2019
9  #
10 # This python library supports the SparkFun Electronics qwiic
11 # qwiic sensor/board ecosystem on a Raspberry Pi (and compatible) single
12 # board computers.
13 #
14 # More information on qwiic is at https://www.sparkfun.com/qwiic
15 #
16 # Do you like this library? Help support SparkFun. Buy a board!
17 #
18 #=====
19 # Copyright (c) 2019 SparkFun Electronics
20 #
21 # Permission is hereby granted, free of charge, to any person obtaining a copy
22 # of this software and associated documentation files (the "Software"), to deal

```

(continues on next page)

(continued from previous page)

```

23 # in the Software without restriction, including without limitation the rights
24 # to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
25 # copies of the Software, and to permit persons to whom the Software is
26 # furnished to do so, subject to the following conditions:
27 #
28 # The above copyright notice and this permission notice shall be included in all
29 # copies or substantial portions of the Software.
30 #
31 # THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
32 # IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
33 # FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
34 # AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
35 # LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
36 # OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
37 # SOFTWARE.
38 #=====
39 # Example 2 - Is something there
40 #
41
42 from __future__ import print_function
43 import qwiic_proximity
44 import time
45 import sys
46
47 def runExample():
48
49     print("\nSparkFun Proximity Sensor VCN4040 Example 2\n")
50     oProx = qwiic_proximity.QwiicProximity()
51
52     if oProx.connected == False:
53         print("The Qwiic Proximity device isn't connected to the system.␣
↪Please check your connection", \
54             file=sys.stderr)
55         return
56
57     # begin Setup
58     oProx.begin()
59
60     oProx.set_led_current(200)
61     oProx.set_prox_integration_time(8) # 1 to 8 is valid
62
63     # Take 8 readings and average them
64     startingProxValue=0
65     for x in range(8):
66         startingProxValue += oProx.get_proximity()
67
68     startingProxValue /= 8
69
70     deltaNeeded = startingProxValue * 0.05 # Look for %5 change
71     if deltaNeeded < 5:
72         deltaNeeded = 5 # set a min value
73
74     # Begin operation loop
75     nothingThere = True
76
77     while True:
78         proxValue = oProx.get_proximity()

```

(continues on next page)



(continued from previous page)

```

79         print("Proximity Value: %d" % proxValue)
80
81         if proxValue > startingProxValue + deltaNeeded:
82             nothingThere = False
83             print("\tSomething is there!")
84
85         elif not nothingThere:
86             print("\tI don't see anything")
87
88             nothingThere=True
89
90         time.sleep(.4)
91
92
93 if __name__ == '__main__':
94     try:
95         runExample()
96     except (KeyboardInterrupt, SystemExit) as exErr:
97         print("\nEnding Example 2")
98         sys.exit(0)
99
100

```

## 6.4 Example 3

Listing 3: examples/qwiic\_proximity\_ex3.py

```

1  #!/usr/bin/env python
2  #-----
3  # qwiic_proximity_ex3.py
4  #
5  # Simple Example for the Qwiic Proximity Device
6  #-----
7  #
8  # Written by SparkFun Electronics, May 2019
9  #
10 # This python library supports the SparkFun Electronics qwiic
11 # qwiic sensor/board ecosystem on a Raspberry Pi (and compatible) single
12 # board computers.
13 #
14 # More information on qwiic is at https://www.sparkfun.com/qwiic
15 #
16 # Do you like this library? Help support SparkFun. Buy a board!
17 #
18 #=====
19 # Copyright (c) 2019 SparkFun Electronics
20 #
21 # Permission is hereby granted, free of charge, to any person obtaining a copy
22 # of this software and associated documentation files (the "Software"), to deal
23 # in the Software without restriction, including without limitation the rights
24 # to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
25 # copies of the Software, and to permit persons to whom the Software is
26 # furnished to do so, subject to the following conditions:
27 #

```

(continues on next page)

(continued from previous page)

```

28 # The above copyright notice and this permission notice shall be included in all
29 # copies or substantial portions of the Software.
30 #
31 # THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
32 # IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
33 # FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
34 # AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
35 # LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
36 # OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
37 # SOFTWARE.
38 #=====
39 # Example 3 - Ambient Light
40 #
41
42 from __future__ import print_function
43 import qwiic_proximity
44 import time
45 import sys
46
47 def runExample():
48
49     print("\nSparkFun Proximity Sensor VCN4040 Example 3\n")
50     oProx = qwiic_proximity.QwiicProximity()
51
52     if oProx.connected == False:
53         print("The Qwiic Proximity device isn't connected to the system.
54 ↪Please check your connection", \
55             file=sys.stderr)
56         return
57
58     # begin Setup
59     oProx.begin()
60
61     oProx.power_off_proximity()          # Power down the proximity portion of
62 ↪the sensor
63     oProx.power_on_ambient()            # Power Up the ambient sensor
64
65     while True:
66         ambientValue = oProx.get_ambient()
67         print("Ambient Value: %d" % ambientValue)
68
69         time.sleep(.4)
70
71 if __name__ == '__main__':
72     try:
73         runExample()
74     except (KeyboardInterrupt, SystemExit) as exErr:
75         print("\nEnding Example 3")
76         sys.exit(0)
77
78

```

## 6.5 Example 4

Listing 4: examples/qwiic\_proximity\_ex4.py

```

1  #!/usr/bin/env python
2  #-----
3  # qwiic_proximity_ex4.py
4  #
5  # Simple Example for the Qwiic Proximity Device
6  #-----
7  #
8  # Written by SparkFun Electronics, May 2019
9  #
10 # This python library supports the SparkFun Electronics qwiic
11 # qwiic sensor/board ecosystem on a Raspberry Pi (and compatible) single
12 # board computers.
13 #
14 # More information on qwiic is at https://www.sparkfun.com/qwiic
15 #
16 # Do you like this library? Help support SparkFun. Buy a board!
17 #
18 #=====
19 # Copyright (c) 2019 SparkFun Electronics
20 #
21 # Permission is hereby granted, free of charge, to any person obtaining a copy
22 # of this software and associated documentation files (the "Software"), to deal
23 # in the Software without restriction, including without limitation the rights
24 # to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
25 # copies of the Software, and to permit persons to whom the Software is
26 # furnished to do so, subject to the following conditions:
27 #
28 # The above copyright notice and this permission notice shall be included in all
29 # copies or substantial portions of the Software.
30 #
31 # THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
32 # IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
33 # FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
34 # AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
35 # LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
36 # OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
37 # SOFTWARE.
38 #=====
39 # Example 4 - All Readings
40 #
41
42 from __future__ import print_function
43 import qwiic_proximity
44 import time
45 import sys
46
47 def runExample():
48
49     print("\nSparkFun Proximity Sensor VCN4040 Example 4\n")
50     oProx = qwiic_proximity.QwiicProximity()
51
52     if oProx.connected == False:
53         print("The Qwiic Proximity device isn't connected to the system.
→Please check your connection", \

```

(continues on next page)

(continued from previous page)

```

54         file=sys.stderr)
55     return
56
57     # begin Setup
58     oProx.begin()
59
60     oProx.power_on_proximity()          # Power up the proximity portion of the
↳sensor
61     oProx.power_on_ambient()           # Power Up the ambient sensor
62     oProx.enable_white_channel()
63
64
65     while True:
66
67         proxValue = oProx.get_proximity()
68         print("Proximity Value: \t[%5d]" % proxValue)
69
70         ambientValue = oProx.get_ambient()
71         print("Ambient Value: \t\t[%5d]" % ambientValue)
72
73         whiteValue = oProx.get_white()
74         print("White Value: \t\t[%5d]\n" % whiteValue)
75
76         time.sleep(.5)
77
78
79
80 if __name__ == '__main__':
81     try:
82         runExample()
83     except (KeyboardInterrupt, SystemExit) as exErr:
84         print("\nEnding Example 4")
85         sys.exit(0)
86
87

```

## 6.6 Example 5

Listing 5: examples/qwiic\_proximity\_ex5.py

```

1  #!/usr/bin/env python
2  #-----
3  # qwiic_proximity_ex5.py
4  #
5  # Simple Example for the Qwiic Proximity Device
6  #-----
7  #
8  # Written by SparkFun Electronics, May 2019
9  #
10 # This python library supports the SparkFun Electronics qwiic
11 # qwiic sensor/board ecosystem on a Raspberry Pi (and compatible) single
12 # board computers.
13 #
14 # More information on qwiic is at https://www.sparkfun.com/qwiic

```

(continues on next page)

(continued from previous page)

```

15 #
16 # Do you like this library? Help support SparkFun. Buy a board!
17 #
18 #=====
19 # Copyright (c) 2019 SparkFun Electronics
20 #
21 # Permission is hereby granted, free of charge, to any person obtaining a copy
22 # of this software and associated documentation files (the "Software"), to deal
23 # in the Software without restriction, including without limitation the rights
24 # to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
25 # copies of the Software, and to permit persons to whom the Software is
26 # furnished to do so, subject to the following conditions:
27 #
28 # The above copyright notice and this permission notice shall be included in all
29 # copies or substantial portions of the Software.
30 #
31 # THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
32 # IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
33 # FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
34 # AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
35 # LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
36 # OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
37 # SOFTWARE.
38 #=====
39 # Example 5 - Advanced Settings
40 #
41
42 from __future__ import print_function
43 import qwiic_proximity
44 import time
45 import sys
46
47 def runExample():
48
49     print("\nSparkFun Proximity Sensor VCN4040 Example 5\n")
50     oProx = qwiic_proximity.QwiicProximity()
51
52     if oProx.connected == False:
53         print("The Qwiic Proximity device isn't connected to the system.␣
→Please check your connection", \
54             file=sys.stderr)
55         return
56
57     # begin Setup
58     oProx.begin()
59
60     oProx.power_on_ambient()          # Power Up the ambient sensor
61
62     # Set the integration time for the proximity sensor
63     # 1 to 8 is valid
64     oProx.prox_integration_time = 8
65
66     # Set the integration time for the ambient light sensor in milliseconds
67     # 80 to 640ms is valid
68     oProx.ambient_integration_time = 80
69
70     # If sensor sees more than this, interrupt pin will go low

```

(continues on next page)

(continued from previous page)

```

71     oProx.prox_high_threshold = 2000
72
73     # The int pin will stay low until the value goes below the low threshold value
74     oProx.prox_low_threshold = 150
75
76     # Enable both 'away' and 'close' interrupts
77     oProx.prox_interrupt_type = oProx.VCNL4040_PS_INT_BOTH
78
79     # This causes the int pin to go low every time a reading is outside the
↪ thresholds
80     # Get a multimeter and probe the INT pin to see this feature in action
81     oProx.enable_prox_logic_mode()
82
83     while True:
84
85         proxValue = oProx.proximity
86         print("Proximity Value: \t[%5d]" % proxValue)
87
88         ambientValue = oProx.ambient
89         print("Ambient Value: \t\t[%5d]\n" % ambientValue)
90
91
92         time.sleep(1)
93
94
95 if __name__ == '__main__':
96     try:
97         runExample()
98     except (KeyboardInterrupt, SystemExit) as exErr:
99         print("\nEnding Example 5")
100        sys.exit(0)
101
102

```

## 6.7 Example 8

Listing 6: examples/qwiic\_proximity\_ex8.py

```

1  #!/usr/bin/env python
2  #-----
3  # qwiic_proximity_ex8.py
4  #
5  # Simple Example for the Qwiic Proximity Device
6  #-----
7  #
8  # Written by SparkFun Electronics, May 2019
9  #
10 # This python library supports the SparkFun Electronics qwiic
11 # qwiic sensor/board ecosystem on a Raspberry Pi (and compatible) single
12 # board computers.
13 #
14 # More information on qwiic is at https://www.sparkfun.com/qwiic
15 #
16 # Do you like this library? Help support SparkFun. Buy a board!

```

(continues on next page)

(continued from previous page)

```

17 #
18 #=====
19 # Copyright (c) 2019 SparkFun Electronics
20 #
21 # Permission is hereby granted, free of charge, to any person obtaining a copy
22 # of this software and associated documentation files (the "Software"), to deal
23 # in the Software without restriction, including without limitation the rights
24 # to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
25 # copies of the Software, and to permit persons to whom the Software is
26 # furnished to do so, subject to the following conditions:
27 #
28 # The above copyright notice and this permission notice shall be included in all
29 # copies or substantial portions of the Software.
30 #
31 # THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
32 # IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
33 # FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
34 # AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
35 # LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
36 # OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
37 # SOFTWARE.
38 #=====
39 # Example 8 - Unit test example
40
41 from __future__ import print_function
42 import qwiic_proximity
43 import time
44 import sys
45
46 def runExample():
47
48     print("\nSparkFun Proximity Sensor VCN4040 Example 1\n")
49     oProx = qwiic_proximity.QwiicProximity()
50
51     if oProx.connected == False:
52         print("The Qwiic Proximity device isn't connected to the system.␣
↪Please check your connection", \
53             file=sys.stderr)
54         return
55
56     oProx.begin()
57
58     while True:
59
60         if oProx.connected:
61             oProx.power_on_proximity()
62
63             proxValue = oProx.get_proximity()
64             print("Good Proximity Value: %d" % proxValue)
65         else:
66             print("Not Connected")
67
68         time.sleep(.4)
69
70
71 if __name__ == '__main__':
72     try:

```

(continues on next page)

(continued from previous page)

```
73         runExample()
74     except (KeyboardInterrupt, SystemExit) as exErr:
75         print("\nEnding Example 8")
76         sys.exit(0)
77
78
```



## CHAPTER 7

---

### Indices and tables

---

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



## q

`qwiic_proximity`, [13](#)



## A

ambient (*qwiic\_proximity.QwiicProximity* attribute),  
13

## B

begin() (*qwiic\_proximity.QwiicProximity* method), 13

## C

connected (*qwiic\_proximity.QwiicProximity* attribute), 14

## D

disable\_active\_force\_mode()  
(*qwiic\_proximity.QwiicProximity* method),  
14

disable\_ambient\_interrupts()  
(*qwiic\_proximity.QwiicProximity* method),  
14

disable\_prox\_logic\_mode()  
(*qwiic\_proximity.QwiicProximity* method),  
14

disable\_smart\_persistence()  
(*qwiic\_proximity.QwiicProximity* method),  
14

disable\_white\_channel()  
(*qwiic\_proximity.QwiicProximity* method),  
14

## E

enable\_active\_force\_mode()  
(*qwiic\_proximity.QwiicProximity* method),  
14

enable\_ambient\_interrupts()  
(*qwiic\_proximity.QwiicProximity* method),  
14

enable\_prox\_logic\_mode()  
(*qwiic\_proximity.QwiicProximity* method),  
14

enable\_smart\_persistence()  
(*qwiic\_proximity.QwiicProximity* method),  
14

enable\_white\_channel()  
(*qwiic\_proximity.QwiicProximity* method),  
14

## G

get\_ambient() (*qwiic\_proximity.QwiicProximity* method), 14

get\_id() (*qwiic\_proximity.QwiicProximity* method),  
15

get\_proximity() (*qwiic\_proximity.QwiicProximity* method), 15

get\_white() (*qwiic\_proximity.QwiicProximity* method), 15

## I

is\_away (*qwiic\_proximity.QwiicProximity* attribute),  
15

is\_close (*qwiic\_proximity.QwiicProximity* attribute),  
15

is\_connected() (*qwiic\_proximity.QwiicProximity* method), 15

is\_dark (*qwiic\_proximity.QwiicProximity* attribute),  
15

is\_light (*qwiic\_proximity.QwiicProximity* attribute),  
15

## P

power\_off\_ambient()  
(*qwiic\_proximity.QwiicProximity* method),  
15

power\_off\_proximity()  
(*qwiic\_proximity.QwiicProximity* method),  
15

power\_on\_ambient()  
(*qwiic\_proximity.QwiicProximity* method),  
16

power\_on\_proximity() (qwiic\_proximity.QwiicProximity method), 17  
 proximity (qwiic\_proximity.QwiicProximity attribute), 16

## Q

qwiic\_proximity (module), 13  
 QwiicProximity (class in qwiic\_proximity), 13

## S

sensor\_id (qwiic\_proximity.QwiicProximity attribute), 16  
 set\_als\_high\_threshold() (qwiic\_proximity.QwiicProximity method), 16  
 set\_als\_low\_threshold() (qwiic\_proximity.QwiicProximity method), 16  
 set\_ambient\_integration\_time() (qwiic\_proximity.QwiicProximity method), 16  
 set\_ambient\_interrupt\_persistence() (qwiic\_proximity.QwiicProximity method), 16  
 set\_ir\_dutycycle() (qwiic\_proximity.QwiicProximity method), 16  
 set\_led\_current() (qwiic\_proximity.QwiicProximity method), 16  
 set\_prox\_cancellation() (qwiic\_proximity.QwiicProximity method), 17  
 set\_prox\_high\_threshold() (qwiic\_proximity.QwiicProximity method), 17  
 set\_prox\_integration\_time() (qwiic\_proximity.QwiicProximity method), 17  
 set\_prox\_interrupt\_persistence() (qwiic\_proximity.QwiicProximity method), 17  
 set\_prox\_interrupt\_type() (qwiic\_proximity.QwiicProximity method), 17  
 set\_prox\_low\_threshold() (qwiic\_proximity.QwiicProximity method), 17  
 set\_prox\_resolution() (qwiic\_proximity.QwiicProximity method), 17

## T

take\_single\_prox\_measurement()

## W

white\_light (qwiic\_proximity.QwiicProximity attribute), 17