
sparkfun_qwiic_pir

Release 0.0.04

SparkFun Electronics

Mar 26, 2021

CONTENTS:

1	Contents	3
2	Supported Platforms	5
3	Dependencies	7
4	Documentation	9
5	Installation	11
5.1	PyPi Installation	11
6	Example Use	13
7	Table of Contents	15
7.1	API Reference	15
7.1.1	qwiic_pir	15
7.2	Example 1 - Raw Reading	18
7.3	Example 2 - Object Detected/Removed	19
7.4	Example 3 - Queue Usage	21
7.5	Example 4 - Pop Queue	23
7.6	Example 5 - Change I2C Address	24
8	Indices and tables	27
Python Module Index		29
Index		31

Python module for the SparkFun Qwiic PIR - 1 uA (EKMB1107112) and SparkFun Qwiic PIR - 170 uA (EKMC4607112K).

This python package is a port of the existing SparkFun Qwiic PIR 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
ONE**

CONTENTS

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

**CHAPTER
TWO**

SUPPORTED PLATFORMS

The Qwiic Button Python package current supports the following platforms:

- Raspberry Pi

**CHAPTER
THREE**

DEPENDENCIES

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

**CHAPTER
FOUR**

DOCUMENTATION

The SparkFun Qwiic PIR module documentation is hosted at [ReadTheDocs](#)

INSTALLATION

5.1 PyPi Installation

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

For the current user:

```
pip install sparkfun-qwiic-pir
```

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-pir-<version>.tar.gz
```

CHAPTER
SIX

EXAMPLE USE

See the examples directory for more detailed use examples.

```
from __future__ import print_function
import qwiic_pir
import time
import sys

debounce_time = .20

def run_example():

    print("\nSparkFun Qwiic PIR Example 1\n")
    my_PIR = qwiic_pir.QwiicPIR()

    if my_PIR.begin() == False:
        print("The Qwiic PIR isn't connected to the system. Please check your connection", \
              file=sys.stderr)
        return

    print ("Waiting 30 seconds for PIR to stabilize")
    for i in range(0, 30):
        print(i)
        time.sleep(1)

    print("Device Stable")

    while True:
        if my_PIR.raw_reading() is True:
            print("Object Detected")
        else:
            print("Object Removed")
        time.sleep(debounce_time)

if __name__ == '__main__':
    try:
        run_example()
    except (KeyboardInterrupt, SystemExit) as exErr:
        print("\nEnding Example 1")
        sys.exit(0)
```


TABLE OF CONTENTS

7.1 API Reference

7.1.1 qwiic_pir

Python module for the Qwiic PIR. This python package is a port of the existing [SparkFun Qwiic Arduino Library](https://github.com/sparkfun/SparkFun_Qwiic_PIR_Arduino_Library) This package can be used in conjunction with the overall [SparkFun Qwiic Python Package](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_pir.QwiicPIR(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 GPIO device object.

Return type Object

`available()`

Return the event_available bit of the EVENT_STATUS register

Returns event_available bit

Rtype bool

`begin()`

Initialize the operation of the Qwiic PIR Run is_connected() and check the ID in the ID register

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

Return type bool

`clear_event_bits()`

Clear the object_remove, object_detect, and event_available bits of the EVENT_STATUS register

Returns Nothing

Return type Void

`disable_interrupt()`

Clear the interrupt_enable bit of the INTERRUPT_CONFIG register

Returns Nothing

Return type Void

enable_interrupt()
Set interrupt_enable bit of the INTERRUPT_CONFIG register to a 1

Returns Nothing

Return type Void

get_I2C_address()
Returns the current I2C address of the Qwiic PIR

Returns current I2C address

Return type int

get_debounce_time()
Returns the value in the EVENT_DEBOUNCE_TIME register

Returns debounce time in milliseconds

Return type int

get_firmware_version()
Read the register and get the major and minor firmware version number.

Returns 16 bytes version number

Return type int

is_connected()
Determine if a Qwiic PIR device is connected to the system.

Returns True if the device is connected, otherwise False.

Return type bool

is_detected_queue_empty()
Returns the is_empty bit of the DETECTED_QUEUE_STATUS register

Returns detected_is_empty

Return type bool

is_detected_queue_full()
Returns the is_full bit of the DETECTED_QUEUE_STATUS register

Returns detected_is_full

Return type bool

is_removed_queue_empty()
Returns the is_empty bit of the REMOVED_QUEUE_STATUS register

Returns removed_is_empty

Return type bool

is_removed_queue_full()
Returns the is_full bit of the REMOVED_QUEUE_STATUS register

Returns removed_is_full

Return type bool

object_detected()
Returns the value of the object_detect status bit of the EVENT_STATUS register

Returns object_detect bit

Return type bool

object_removed()
Returns the value of the object_remove status bit of the EVENT_STATUS register

Returns object_remove bit

Return type bool

pop_detected_queue()
Returns contents of DETECTED_QUEUE_BACK register and writes a 1 to pop_request bit of DETECTED_QUEUE_STATUS register.

Returns DETECTED_QUEUE_BACK

Return type int

pop_removed_queue()
Returns contents of REMOVED_QUEUE_BACK register and writes a 1 to pop_request bit of REMOVED_QUEUE_STATUS register.

Returns REMOVED_QUEUE_BACK

Return type int

raw_reading()
Returns the value of the raw_reading status bit of the EVENT_STATUS register

Returns raw_object_detected bit

Return type bool

reset_interrupt_config()
Enable detect interrupt and clear the event_available bit of EVENT_STATUS register

Returns Nothing

Return type Void

set_I2C_address(new_address)
Change the I2C address of the Qwiic PIR

Parameters **new_address** – the new I2C address to set the Qwiic PIR to

Returns True if the change was successful, false otherwise.

Return type bool

set_debounce_time(time)
Write two bytes into the EVENT_DEBOUNCE_TIME register

Parameters **time** – the time in milliseconds to set debounce time to The max deounce time is 0xFFFF milliseconds, but the function checks if the entered parameter is valid

Returns Nothing

Return type void

time_since_first_detect()
Returns the four bytes of DETECTED_QUEUE_BACK. Time in milliseconds

Returns DETECTED_QUEUE_BACK

Return type int

time_since_first_remove()
Returns the four bytes of REMOVED_QUEUE_BACK. Time in milliseconds

Returns REMOVED_QUEUE_BACK

Return type int

time_since_last_detect()

Returns the four bytes of DETECTED_QUEUE_FRONT. Time in milliseconds.

Returns DETECTED_QUEUE_FRONT

Return type int

time_since_last_remove()

Returns the four bytes of REMOVED_QUEUE_FRONT. Time in milliseconds.

Returns REMOVED_QUEUE_FRONT

Return type int

7.2 Example 1 - Raw Reading

Listing 1: examples/qwiic_pir_ex1.py

```
1 #!/usr/bin/env python
2 #-----
3 # qwiic_pir_ex1.py
4 #
5 # Simple Example for the Qwiic PIR Device
6 #-----
7 #
8 # Written by Andy England @ SparkFun Electronics, January 2021
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) 2021 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
```

(continues on next page)

(continued from previous page)

```

37 # SOFTWARE.
38 =====
39 # Example 1
40 #
41
42 from __future__ import print_function
43 import qwiic_pir
44 import time
45 import sys
46
47 debounce_time = 0.2
48
49 def run_example():
50
51     print("\nSparkFun Qwiic PIR Example 1\n")
52     my_PIR = qwiic_pir.QwiicPIR()
53
54     if my_PIR.begin() == False:
55         print("The Qwiic PIR isn't connected to the system. Please check your connection", \
56             file=sys.stderr)
57     return
58
59     print ("Waiting 30 seconds for PIR to stabilize")
60     for i in range(0, 30):
61         print(i)
62         time.sleep(1)
63
64     print("Device Stable")
65
66     while True:
67         if my_PIR.raw_reading() is True:
68             print("Object Detected")
69         else:
70             print("Object Removed")
71         time.sleep(debounce_time)
72
73 if __name__ == '__main__':
74     try:
75         run_example()
76     except (KeyboardInterrupt, SystemExit) as exErr:
77         print("\nEnding Example 1")
78         sys.exit(0)

```

7.3 Example 2 - Object Detected/Removed

Listing 2: examples/qwiic_pir_ex2.py

```

1#!/usr/bin/env python
2=====
3# qwiic_pir_ex2.py
4#
5# Simple Example for the Qwiic PIR Device

```

(continues on next page)

(continued from previous page)

```
6 #-----
7 #
8 # Written by Andy England @ SparkFun Electronics, January 2021
9 #
10 # This python library supports the SparkFun Electronicis 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) 2021 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 2
40 #
41
42 from __future__ import print_function
43 import qwiic_pir
44 import time
45 import sys
46
47 def run_example():
48
49     print("\nSparkFun Qwiic PIR Example 2\n")
50     my_PIR = qwiic_pir.QwiicPIR()
51
52     if my_PIR.begin() == False:
53         print("The Qwiic PIR isn't connected to the system. Please check your_"
54             "connection", \
55                 file=sys.stderr)
56         return
57
58     print ("Waiting 30 seconds for PIR to stabilize")
59     for i in range(0, 30):
60         print(i)
61         time.sleep(1)
```

(continues on next page)

(continued from previous page)

```

62     print("Device Stable")
63
64     while True:
65         if my_PIR.available() is True:
66             if my_PIR.object_detected():
67                 print("Object Detected")
68             if my_PIR.object_removed():
69                 print("Object Removed")
70             my_PIR.clear_event_bits()
71             time.sleep(0.2)
72
73 if __name__ == '__main__':
74     try:
75         run_example()
76     except (KeyboardInterrupt, SystemExit) as exErr:
77         print("\nEnding Example 2")
78         sys.exit(0)

```

7.4 Example 3 - Queue Usage

Listing 3: examples/qwiic_pir_ex3.py

```

1  #!/usr/bin/env python
2  #-----
3  # qwiic_pir_ex3.py
4  #
5  # Queue Example for the Qwiic PIR Device
6  #-----
7  #
8  # Written by Andy England @ SparkFun Electronics, January 2021
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) 2021 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,

```

(continues on next page)

(continued from previous page)

```
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
40 #
41
42 from __future__ import print_function
43 import qwiic_pir
44 import time
45 import sys
46
47 def run_example():
48
49     print("\nSparkFun Qwiic PIR Example 3\n")
50     my_PIR = qwiic_pir.QwiicPIR()
51
52     if my_PIR.begin() == False:
53         print("The Qwiic PIR isn't connected to the system. Please check your_"
54             "connection", \
55             file=sys.stderr)
56     return
57
58     print ("Waiting 30 seconds for PIR to stabilize")
59     for i in range(0, 30):
60         print(i)
61         time.sleep(1)
62
63     print("Device Stable")
64
65     while True:
66         if my_PIR.is_detected_queue_empty() is False:
67             last_detect = my_PIR.time_since_last_detect() / 1000.0
68             first_detect = my_PIR.time_since_first_detect() / 1000.0
69             print("\n" + str(last_detect) + "s since last PIR detect   ")
70             print(str(first_detect) + "s since first PIR detect")
71         else:
72             print("Detected queue is empty")
73
74         if my_PIR.is_removed_queue_empty() is False:
75             last_remove = my_PIR.time_since_last_remove() / 1000.0
76             first_remove = my_PIR.time_since_first_remove() / 1000.0
77             print("\n" + str(last_remove) + "s since last PIR remove   ")
78             print(str(first_remove) + "s since first PIR remove")
79         else:
80             print("Removed queue is empty")
81
82         time.sleep(0.2)
83
84 if __name__ == '__main__':
85     try:
86         run_example()
87     except (KeyboardInterrupt, SystemExit) as exErr:
88         print("\nEnding Example 3")
89         sys.exit(0)
```

7.5 Example 4 - Pop Queue

Listing 4: examples/qwiic_pir_ex4.py

```

1 #!/usr/bin/env python
2 -----
3 # qwiic_pir_ex4.py
4 #
5 # Queue Popping Example for the Qwiic PIR Device
6 #
7 #
8 # Written by Priyanka Makin @ SparkFun Electronics, March 2021
9 #
10 # This python library supports the SparkFun Electronics 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) 2021 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
40 #
41
42 from __future__ import print_function
43 import qwiic_pir
44 import time
45 import sys
46
47 def run_example():
48
49     print("\nSparkFun Qwiic PIR Example 4\n")
50     my_PIR = qwiic_pir.QwiicPIR()
51
52     if my_PIR.begin() == False:
53         print("The Qwiic PIR 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     print ("Waiting 30 seconds for PIR to stabilize")
58     for i in range(0, 30):
59         print(i)
60         time.sleep(1)
61
62     print("Device Stable")
63
64     while True:
65         print("\nType 'd' to pop from the detected queue.")
66         val = raw_input("Type 'r' to pop from the removed queue: ")
67         # If the character is 'd' or 'D', then pop a value off the detected
68         # queue
69         if val == 'd' or val == 'D':
70             print("\nPopped detected queue! The first timestamp in"
71         # detected queue was: ")
72             print(str(my_PIR.pop_detected_queue() / 1000.0))
73
74         # If the character is 'r' or 'R', then pop a value off the removed
75         # queue
76         if val == 'r' or val == 'R':
77             print("\nPopped removed queue! The first timestamp in removed"
78         # queue was: ")
79             print(str(my_PIR.pop_removed_queue() / 1000.0))
80
81         time.sleep(0.2)
82
83 if __name__ == '__main__':
84     try:
85         run_example()
86     except (KeyboardInterrupt, SystemExit) as exErr:
87         print("\nEnding Example 4")
88         sys.exit(0)
```

7.6 Example 5 - Change I2C Address

Listing 5: examples/qwiic_pir_ex5.py

```
1 #!/usr/bin/env python
2 -----
3 # qwiic_pir_ex5.py
4 #
5 # Simple Example to change the I2C address of the Qwiic PIR Device
6 -----
7 #
8 # Written by Priyanka Makin @ SparkFun Electronics, March 2021
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) 2021 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
40 #
41
42 from __future__ import print_function
43 import qwiic_pir
44 import time
45 import sys
46
47 def run_example():
48
49     print("\nSparkFun Qwiic PIR Example 5\n")
50     my_PIR = qwiic_pir.QwiicPIR()
51
52     if my_PIR.begin() == False:
53         print("The Qwiic PIR isn't connected to the system. Please check your connection", \
54               file=sys.stderr)
55     return
56
57     print("\nEnter a new I2C address for the Qwiic PIR to use.")
58     print("\nDon't use the 0x prefix. For instance, if you wanted to")
59     print("\nchange the address to 0x5B, you would type 5B and hit enter.")
60
61     new_address = input("\nNew address: ")
62     # Change to hex
63     new_address = int(new_address, 16)
64
65     # Check if the user entered a valid address
66     if new_address > 0x08 and new_address < 0x77:
67         print("\nCharacters received and new address valid!")
68         print("\nAttempting to set Qwiic PIR to new address...")
69
70     my_PIR.set_I2C_address(new_address)

```

(continues on next page)

(continued from previous page)

```
71     print("\nAddress successfully changed!")

72

73     # Check that the Qwiic PIR acknowledges on the new address
74     time.sleep(0.02)
75     if my_PIR.begin() == False:
76         print("\nThe Qwiic PIR is not connected to the system. Please\n"
77             "check you're connection", \
78             file=sys.stderr)

79     else:
80         print("\nPIR acknowledged on new address!")

81

82     else:
83         print("\nAddress entered not valid I2C address.")

84

85 if __name__ == '__main__':
86     try:
87         run_example()
88     except (KeyboardInterrupt, SystemExit) as exErr:
89         print("\nEnding Example 5")
90         sys.exit(0)
```

**CHAPTER
EIGHT**

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

q

qwiic_pir, 15

INDEX

A

available() (*qwiic_pir.QwiicPIR method*), 15

B

begin() (*qwiic_pir.QwiicPIR method*), 15

C

clear_event_bits() (*qwiic_pir.QwiicPIR method*),
15

D

disable_interrupt() (*qwiic_pir.QwiicPIR
method*), 15

E

enable_interrupt() (*qwiic_pir.QwiicPIR method*),
15

G

get_debounce_time() (*qwiic_pir.QwiicPIR
method*), 16
get_firmware_version() (*qwiic_pir.QwiicPIR
method*), 16
get_I2C_address() (*qwiic_pir.QwiicPIR method*),
16

I

is_connected() (*qwiic_pir.QwiicPIR method*), 16
is_detected_queue_empty()
 (*qwiic_pir.QwiicPIR method*), 16
is_detected_queue_full() (*qwiic_pir.QwiicPIR
method*), 16
is_removed_queue_empty() (*qwiic_pir.QwiicPIR
method*), 16
is_removed_queue_full() (*qwiic_pir.QwiicPIR
method*), 16

M

module
 qwiic_pir, 15

O

object_detected() (*qwiic_pir.QwiicPIR method*),
16

object_removed() (*qwiic_pir.QwiicPIR method*), 17

P

pop_detected_queue() (*qwiic_pir.QwiicPIR
method*), 17
pop_removed_queue() (*qwiic_pir.QwiicPIR
method*), 17

Q

qwiic_pir
 module, 15
QwiicPIR (*class in qwiic_pir*), 15

R

raw_reading() (*qwiic_pir.QwiicPIR method*), 17
reset_interrupt_config() (*qwiic_pir.QwiicPIR
method*), 17

S

set_debounce_time() (*qwiic_pir.QwiicPIR
method*), 17
set_I2C_address() (*qwiic_pir.QwiicPIR method*),
17

T

time_since_first_detect()
 (*qwiic_pir.QwiicPIR method*), 17
time_since_first_remove()
 (*qwiic_pir.QwiicPIR method*), 17
time_since_last_detect() (*qwiic_pir.QwiicPIR
method*), 18
time_since_last_remove() (*qwiic_pir.QwiicPIR
method*), 18