# sparkfun_qwiic_pir

***Release 0.0.04***

**SparkFun Electronics**

**Mar 26, 2021**

# CONTENTS:

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

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

CHAPTER

FIVE

# 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)
```

CHAPTER

SEVEN

# 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 exisiting [SparkFun Qwiic PIR 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

**Rtye** 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 intialization 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

```
#!/usr/bin/env python
#-----------------------------------------------------------------------------
# qwiic_pir_ex1.py
#
# Simple Example for the Qwiic PIR Device
#------------------------------------------------------------------------
#
# Written by Andy England @ SparkFun Electronics, January 2021
#
# This python library supports the SparkFun Electroncis qwiic
# qwiic sensor/board ecosystem on a Raspberry Pi (and compatable) single
# board computers.
#
# More information on qwiic is at https://www.sparkfun.com/qwiic
#
# Do you like this library? Help support SparkFun. Buy a board!
#
#==================================================================================
# Copyright (c) 2021 SparkFun Electronics
#
# Permission is hereby granted, free of charge, to any person obtaining a copy
# of this software and associated documentation files (the "Software"), to deal
# in the Software without restriction, including without limitation the rights
# to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
# copies of the Software, and to permit persons to whom the Software is
# furnished to do so, subject to the following conditions:
#
# The above copyright notice and this permission notice shall be included in all
# copies or substantial portions of the Software.
#
# THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
# IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
# FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
# AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
# LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
# OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
```

(continues on next page)

(continued from previous page)

```
# SOFTWARE.
#==================================================================================
# Example 1
#

from __future__ import print_function
import qwiic_pir
import time
import sys

debounce_time = 0.2

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)
```

## 7.3 Example 2 - Object Detected/Removed

Listing 2: examples/qwiic_pir_ex2.py

```
#!/usr/bin/env python
#-----------------------------------------------------------------------------
# qwiic_pir_ex2.py
#
# Simple Example for the Qwiic PIR Device
```

(continues on next page)

(continued from previous page)

```python
#-------------------------------------------------------------------------
#
# Written by Andy England @ SparkFun Electronics, January 2021
#
# This python library supports the SparkFun Electroncis qwiic
# qwiic sensor/board ecosystem on a Raspberry Pi (and compatable) single
# board computers.
#
# More information on qwiic is at https://www.sparkfun.com/qwiic
#
# Do you like this library? Help support SparkFun. Buy a board!
#
#===================================================================================
# Copyright (c) 2021 SparkFun Electronics
#
# Permission is hereby granted, free of charge, to any person obtaining a copy
# of this software and associated documentation files (the "Software"), to deal
# in the Software without restriction, including without limitation the rights
# to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
# copies of the Software, and to permit persons to whom the Software is
# furnished to do so, subject to the following conditions:
#
# The above copyright notice and this permission notice shall be included in all
# copies or substantial portions of the Software.
#
# THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
# IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
# FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
# AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
# LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
# OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
# SOFTWARE.
#===================================================================================
# Example 2
#

from __future__ import print_function
import qwiic_pir
import time
import sys

def run_example():

        print("\nSparkFun Qwiic PIR  Example 2\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)

```

(continues on next page)

(continued from previous page)

```
        print("Device Stable")

        while True:
                if my_PIR.available() is True:
                        if my_PIR.object_detected():
                                print("Object Detected")
                        if my_PIR.object_removed():
                                print("Object Removed")
                        my_PIR.clear_event_bits()
                time.sleep(0.2)

if __name__ == '__main__':
        try:
                run_example()
        except (KeyboardInterrupt, SystemExit) as exErr:
                print("\nEnding Example 2")
                sys.exit(0)
```

## 7.4 Example 3 - Queue Usage

Listing 3: examples/qwiic_pir_ex3.py

```
#!/usr/bin/env python
#-----------------------------------------------------------------------------
# qwiic_pir_ex3.py
#
# Queue Example for the Qwiic PIR Device
#------------------------------------------------------------------------
#
# Written by Andy England @ SparkFun Electronics, January 2021
#
# This python library supports the SparkFun Electroncis qwiic
# qwiic sensor/board ecosystem on a Raspberry Pi (and compatable) single
# board computers.
#
# More information on qwiic is at https://www.sparkfun.com/qwiic
#
# Do you like this library? Help support SparkFun. Buy a board!
#
#==================================================================================
# Copyright (c) 2021 SparkFun Electronics
#
# Permission is hereby granted, free of charge, to any person obtaining a copy
# of this software and associated documentation files (the "Software"), to deal
# in the Software without restriction, including without limitation the rights
# to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
# copies of the Software, and to permit persons to whom the Software is
# furnished to do so, subject to the following conditions:
#
# The above copyright notice and this permission notice shall be included in all
# copies or substantial portions of the Software.
#
# THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
# IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
```

(continues on next page)

(continued from previous page)

```
# FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
# AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
# LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
# OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
# SOFTWARE.
#==================================================================================
# Example 3
#

from __future__ import print_function
import qwiic_pir
import time
import sys

def run_example():

        print("\nSparkFun Qwiic PIR  Example 3\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.is_detected_queue_empty() is False:
                        last_detect = my_PIR.time_since_last_detect() / 1000.0
                        first_detect =  my_PIR.time_since_first_detect() / 1000.0
                        print("\n" + str(last_detect) + "s since last PIR detect   ")
                        print(str(first_detect) + "s since first PIR detect")
                else:
                        print("Detected queue is empty")

                if my_PIR.is_removed_queue_empty() is False:
                        last_remove = my_PIR.time_since_last_remove() / 1000.0
                        first_remove =  my_PIR.time_since_first_remove() / 1000.0
                        print("\n" + str(last_remove) + "s since last PIR remove   ")
                        print(str(first_remove) + "s since first PIR remove")
                else:
                        print("Removed queue is empty")

                time.sleep(0.2)

if __name__ == '__main__':
        try:
                run_example()
        except (KeyboardInterrupt, SystemExit) as exErr:
                print("\nEnding Example 3")
                sys.exit(0)
```

## 7.5 Example 4 - Pop Queue

Listing 4: examples/qwiic_pir_ex4.py

```
#!/usr/bin/env python
#-----------------------------------------------------------------------------
# qwiic_pir_ex4.py
#
# Queue Popping Example for the Qwiic PIR Device
#------------------------------------------------------------------------
#
# Written by Priyanka Makin @ SparkFun Electronics, March 2021
#
# This python library supports the SparkFun Electroncis qwiic
# qwiic sensor/board ecosystem on a Raspberry Pi (and compatable) single
# board computers.
#
# More information on qwiic is at https://www.sparkfun.com/qwiic
#
# Do you like this library? Help support SparkFun. Buy a board!
#
#==================================================================================
# Copyright (c) 2021 SparkFun Electronics
#
# Permission is hereby granted, free of charge, to any person obtaining a copy
# of this software and associated documentation files (the "Software"), to deal
# in the Software without restriction, including without limitation the rights
# to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
# copies of the Software, and to permit persons to whom the Software is
# furnished to do so, subject to the following conditions:
#
# The above copyright notice and this permission notice shall be included in all
# copies or substantial portions of the Software.
#
# THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
# IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
# FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
# AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
# LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
# OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
# SOFTWARE.
#==================================================================================
# Example 4
#

from __future__ import print_function
import qwiic_pir
import time
import sys

def run_example():

        print("\nSparkFun Qwiic PIR  Example 4\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", \
```

(continues on next page)

(continued from previous page)

```
                        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:
                print("\nType 'd' to pop from the detected queue.")
                val = raw_input("Type 'r' to pop from the removed queue: ")
                # If the character is 'd' or 'D', then pop a value off the detected␣
↪queue
                if val == 'd' or val == 'D':
                        print("\nPopped detected queue! The first timestamp in␣
↪detected queue was: ")
                        print(str(my_PIR.pop_detected_queue() / 1000.0))

                # If the character is 'r' or 'R', then pop a value off the removed␣
↪queue
                if val == 'r' or val == 'R':
                        print("\nPopped removed queue! The first timestamp in removed␣
↪queue was: ")
                        print(str(my_PIR.pop_removed_queue() / 1000.0))

                time.sleep(0.2)

if __name__ == '__main__':
        try:
                run_example()
        except (KeyboardInterrupt, SystemExit) as exErr:
                print("\nEnding Example 4")
                sys.exit(0)
```

## 7.6 Example 5 - Change I2C Address

Listing 5: examples/qwiic_pir_ex5.py

```
#!/usr/bin/env python
#-----------------------------------------------------------------------------
# qwiic_pir_ex5.py
#
# Simple Example to change the I2C address of the Qwiic PIR Device
#------------------------------------------------------------------------
#
# Written by Priyanka Makin @ SparkFun Electronics, March 2021
#
# This python library supports the SparkFun Electroncis qwiic
# qwiic sensor/board ecosystem on a Raspberry Pi (and compatable) single
# board computers.
#
# More information on qwiic is at https://www.sparkfun.com/qwiic
```

(continues on next page)

(continued from previous page)

```
#
# Do you like this library? Help support SparkFun. Buy a board!
#
#==================================================================================
# Copyright (c) 2021 SparkFun Electronics
#
# Permission is hereby granted, free of charge, to any person obtaining a copy
# of this software and associated documentation files (the "Software"), to deal
# in the Software without restriction, including without limitation the rights
# to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
# copies of the Software, and to permit persons to whom the Software is
# furnished to do so, subject to the following conditions:
#
# The above copyright notice and this permission notice shall be included in all
# copies or substantial portions of the Software.
#
# THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
# IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
# FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
# AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
# LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
# OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
# SOFTWARE.
#==================================================================================
# Example 5
#

from __future__ import print_function
import qwiic_pir
import time
import sys

def run_example():

        print("\nSparkFun Qwiic PIR  Example 5\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("\nEnter a new I2C address for the Qwiic PIR to use.")
        print("\nDon't use the 0x prefix. For instance, if you wanted to")
        print("\nchange the address to 0x5B, you would type 5B and hit enter.")

        new_address = input("\nNew address: ")
        # Change to hex
        new_address = int(new_address, 16)

        # Check if the user entered a valid address
        if new_address > 0x08 and new_address < 0x77:
                print("\nCharacters received and new address valid!")
                print("\nAttempting to set Qwiic PIR to new address...")

                my_PIR.set_I2C_address(new_address)
```

(continues on next page)

(continued from previous page)

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

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

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

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

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

# CHAPTER EIGHT

# INDICES AND TABLES

- genindex
- modindex
- search

# PYTHON MODULE INDEX

# INDEX