

---

# **sparkfun\_qwiic\_button**

***Release 2.0.1***

**SparkFun Electronics**

**Apr 05, 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\_button . . . . . 15
  - 7.2 Example One . . . . . 19
  - 7.3 Example 2 . . . . . 20
  - 7.4 Example 3 . . . . . 22
  - 7.5 Example 4 . . . . . 24
  - 7.6 Example 5 . . . . . 25
  - 7.7 Example 6 . . . . . 27
  - 7.8 Example 7 . . . . . 29
- 8 Indices and tables 31**
- Python Module Index 33**
- Index 35**



Python module for the [SparkFun Qwiic Button](#)

This module is also compatible with the following products:

- [SparkFun Qwiic Button - Green LED](#)
- [SparkFun Qwiic Button Breakout](#)
- [Qwiic Arcade - Red](#)
- [Qwiic Arcade - Blue](#)
- [Qwiic Switch](#)

Please remember that SparkX products are experimental, therefore full functionality is not gauranteed.

This python package is a port of the existing [SparkFun Qwiic Button 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](#).



## CONTENTS

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





## SUPPORTED PLATFORMS

The Qwiic Button Python package current supports the following platforms:

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



## DEPENDENCIES

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



## DOCUMENTATION

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



## INSTALLATION

### 5.1 PyPi Installation

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

For the current user:

```
pip install sparkfun-qwiic-button
```

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





## EXAMPLE USE

See the examples directory for more detailed use examples.

```
from __future__ import print_function
import qwiic_button
import time
import sys

def run_example():

    print("\nSparkFun Qwiic Button Example 1")
    my_button = qwiic_button.QwiicButton()

    if my_button.begin() == False:
        print("\nThe Qwiic Button isn't connected to the system. Please check your_
↪connection", \
            file=sys.stderr)
        return
    print("\nButton ready!")

    while True:

        if my_button.is_button_pressed() == True:
            print("\nThe button is pressed!")

        else:
            print("\nThe button is not pressed!")

        time.sleep(0.02)

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\_button

Python module for the Qwiic Button.

This python package is a port of the existing [SparkFun Qwiic Button Arduino Library]([https://github.com/sparkfun/SparkFun\\_Qwiic\\_Button\\_Arduino\\_Library](https://github.com/sparkfun/SparkFun_Qwiic_Button_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_button.QwiicButton` (*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

**LED\_config** (*brightness, cycle\_time, off\_time, granularity=1*)

Write brightness, cycle\_time, off\_time, and granularity parameters to their respective registers: LED\_BRIGHTNESS, LED\_PULSE\_CYCLE\_TIME, LED\_PULSE\_OFF\_TIME, LED\_PULSE\_GRANULARITY

##### Parameters

- **brightness** – between 0 (led off) and 255 (max brightness)
- **cycle\_time** – total pulse cycle in in milliseconds Range 0 to 0xFFFF
- **off\_time** – off time between pulses in milliseconds Range 0 to 0xFFFF
- **granularity** – the amount of steps it takes to get to led brightness If not provided, granularity defaults to 1

**Returns** Nothing

**Return type** Void

**LED\_off ()**

Write zero's to all the LED registers: LED\_BRIGHTNESS, LED\_PULSE\_CYCLE\_TIME, LED\_PULSE\_OFF\_TIME, and LED\_PULSE\_GRANULARITY defaults to zero.

**Returns** Nothing

**Return type** void

**LED\_on (brightness)**

Set LED on without pulse

**Parameters** **brightness** – between 0 (led off) and 255 (max brightness)

**Returns** Nothing

**Return type** Void

**available ()**

Return the event\_available bit of the BUTTON\_STATUS register

**Returns** event\_available bit

**Rtype** bool

**begin ()**

Initialize the operation of the Qwiic Button 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 is\_pressed, has\_been\_clicked, and event\_available bits of the BUTTON\_STATUS register

**Returns** Nothing

**Return type** Void

**disable\_clicked\_interrupt ()**

Clear the clicked\_enable bit of the INTERRUPT\_CONFIG register

**Returns** Nothing

**Return type** Void

**disable\_pressed\_interrupt ()**

Clear the pressed\_enable bit of the INTERRUPT\_CONFIG register

**Returns** Nothing

**Return type** Void

**enable\_clicked\_interrupt ()**

Set the clicked\_enable bit of the INTERRUPT\_CONFIG register

**Returns** Nothing

**Return type** Void

**enable\_pressed\_interrupt ()**

Set pressed\_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 Button

**Returns** current I2C address

**Return type** int

**get\_debounce\_time()**

Returns the value in the BUTTON\_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

**has\_button\_been\_clicked()**

Returns the value of the has\_been\_clicked status bit of the BUTTON\_STATUS register

**Returns** has\_been\_clicked bit

**Return type** bool

**is\_button\_pressed()**

Returns the value of the is\_pressed status bit of the BUTTON\_STATUS register

**Returns** is\_pressed bit

**Return type** bool

**is\_clicked\_queue\_empty()**

Reads the is\_empty bit of the CLICKED\_QUEUE\_STATUS register

**Returns** clicked\_is\_empty

**Return type** bool

**is\_clicked\_queue\_full()**

Reads the is\_full bit of the CLICKED\_QUEUE\_STATUS register

**Returns** clicked\_is\_full

**Return type** bool

**is\_connected()**

Determine if a Qwiic Button device is connected to the system.

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

**Return type** bool

**is\_pressed\_queue\_empty()**

Returns the is\_empty bit of the PRESSED\_QUEUE\_STATUS register

**Returns** pressed\_is\_empty

**Return type** bool

**is\_pressed\_queue\_full()**

Returns the is\_full bit of the PRESSED\_QUEUE\_STATUS register

**Returns** pressed\_is\_full

**Return type** bool

**pop\_clicked\_queue()**

Returns contents of CLICKED\_QUEUE\_BACK register and writes a 1 to popRequest bit of CLICKED\_QUEUE\_STATUS register.

**Returns** CLICKED\_QUEUE\_BACK

**Return type** int

**pop\_pressed\_queue()**

Returns contents of PRESSED\_QUEUE\_BACK register and writes a 1 to popRequest bit of PRESSED\_QUEUE\_STATUS register.

**Returns** PRESSED\_QUEUE\_BACK

**Return type** int

**reset\_interrupt\_config()**

Enable pressed and clicked interrupts and clear the event\_available bit of BUTTON\_STATUS register

**Returns** Nothing

**Return type** Void

**set\_I2C\_address(new\_address)**

Change the I2C address of the Qwiic Button

**Parameters** **new\_address** – the new I2C address to set the Qwiic Button to The function itself checks if the entered parameter is a valid I2C address

**Returns** True if the change was successful, false otherwise.

**Return type** bool

**set\_debounce\_time(time)**

Write two bytes into the BUTTON\_DEBOUNCE\_TIME register

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

**Returns** Nothing

**Return type** void

**time\_since\_first\_click()**

Returns the four bytes of CLICKED\_QUEUE\_BACK register. Time in milliseconds

**Returns** CLICKED\_QUEUE\_BACK

**Return type** int

**time\_since\_first\_press()**

Returns the four bytes of PRESSED\_QUEUE\_BACK. Time in milliseconds

**Returns** PRESSED\_QUEUE\_BACK

**Return type** int

**time\_since\_last\_click()**

Returns the four bytes of CLICKED\_QUEUE\_FRONT register. Time in milliseconds

**Returns** CLICKED\_QUEUE\_FRONT

**Return type** int

**time\_since\_last\_press()**

Returns the four bytes of PRESSED\_QUEUE\_FRONT. Time in milliseconds.

**Returns** PRESSED\_QUEUE\_FRONT

**Return type** int

**class** qwiic\_button.QwiicButton (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

## 7.2 Example One

Listing 1: examples/qwiic\_button\_ex1\_buttonPress.py

```

1  #!/usr/bin/env python
2  #-----
3  # qwiic_button_ex1.py
4  #
5  # Simple Example for the Qwiic Button. Checks whether the button is pressed and
6  # then prints its status.
7  #-----
8  #
9  # Written by Priyanka Makin @ SparkFun Electronics, January 2021
10 #
11 # This python library supports the SparkFun Electronics qwiic
12 # qwiic sensor/board ecosystem on a Raspberry Pi (and compatible) single
13 # board computers.
14 #
15 # More information on qwiic is at https://www.sparkfun.com/qwiic
16 #
17 # Do you like this library? Help support SparkFun. Buy a board!
18 #
19 #=====
20 # Copyright (c) 2019 SparkFun Electronics
21 #
22 # Permission is hereby granted, free of charge, to any person obtaining a copy
23 # of this software and associated documentation files (the "Software"), to deal
24 # in the Software without restriction, including without limitation the rights
25 # to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
26 # copies of the Software, and to permit persons to whom the Software is
27 # furnished to do so, subject to the following conditions:
28 #
29 # The above copyright notice and this permission notice shall be included in all
30 # copies or substantial portions of the Software.
31 #
32 # THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
33 # IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
34 # FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
35 # AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
36 # LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
37 # OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE

```

(continues on next page)

(continued from previous page)

```

38 # SOFTWARE.
39 #=====
40 # Example 1
41
42 from __future__ import print_function
43 import qwiic_button
44 import time
45 import sys
46
47 def run_example():
48
49     print("\nSparkFun Qwiic Button Example 1")
50     my_button = qwiic_button.QwiicButton()
51
52     if my_button.begin() == False:
53         print("\nThe Qwiic Button isn't connected to the system. Please check your_
↪connection", \
54             file=sys.stderr)
55         return
56     print("\nButton ready!")
57
58     while True:
59
60         if my_button.is_button_pressed() == True:
61             print("\nThe button is pressed!")
62
63         else:
64             print("\nThe button is not pressed!")
65
66         time.sleep(0.02)
67
68 if __name__ == '__main__':
69     try:
70         run_example()
71     except (KeyboardInterrupt, SystemExit) as exErr:
72         print("\nEnding Example 1")
73         sys.exit(0)

```

## 7.3 Example 2

Listing 2: examples/qwiic\_button\_ex2\_LEDOn.py

```

1  #!/usr/bin/env python
2  #-----
3  # qwiic_button_ex2.py
4  #
5  # Simple Example for the Qwiic Button. Turns on the button's built in LED
6  # when pressed and prints status.
7  #-----
8  #
9  # Written by Priyanka Makin @ SparkFun Electronics, January 2021
10 #
11 # This python library supports the SparkFun Electronics qwiic
12 # qwiic sensor/board ecosystem on a Raspberry Pi (and compatable) single

```

(continues on next page)



(continued from previous page)

```

13 # board computers.
14 #
15 # More information on qwiic is at https://www.sparkfun.com/qwiic
16 #
17 # Do you like this library? Help support SparkFun. Buy a board!
18 #
19 #=====
20 # Copyright (c) 2019 SparkFun Electronics
21 #
22 # Permission is hereby granted, free of charge, to any person obtaining a copy
23 # of this software and associated documentation files (the "Software"), to deal
24 # in the Software without restriction, including without limitation the rights
25 # to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
26 # copies of the Software, and to permit persons to whom the Software is
27 # furnished to do so, subject to the following conditions:
28 #
29 # The above copyright notice and this permission notice shall be included in all
30 # copies or substantial portions of the Software.
31 #
32 # THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
33 # IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
34 # FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
35 # AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
36 # LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
37 # OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
38 # SOFTWARE.
39 #=====
40 # Example 2
41
42 from __future__ import print_function
43 import qwiic_button
44 import time
45 import sys
46
47 brightness = 100
48
49 def run_example():
50
51     print("\nSparkFun Qwiic Button Example 2")
52     my_button = qwiic_button.QwiicButton()
53
54     if my_button.begin() == False:
55         print("\nThe Qwiic Button isn't connected to the system. Please check your_
↪connection", \
56             file=sys.stderr)
57         return
58
59     print("\nButton ready!")
60
61     while True:
62
63         if my_button.is_button_pressed() == True:
64             print("\nThe button is pressed!")
65             my_button.LED_on(brightness)
66
67         else:
68             print("\nThe button is not pressed.")

```

(continues on next page)

(continued from previous page)

```

69         my_button.LED_off()
70
71         time.sleep(0.02)
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

Listing 3: examples/qwiic\_button\_ex3\_LEDconfig.py

```

1  #!/usr/bin/env python
2  #-----
3  # qwiic_button_ex3.py
4  #
5  # Simple Example for the Qwiic Button. Checks whether the button is pressed and
6  # and the LED pulses if it is.
7  #-----
8  #
9  # Written by Priyanka Makin @ SparkFun Electronics, January 2021
10 #
11 # This python library supports the SparkFun Electronics qwiic
12 # qwiic sensor/board ecosystem on a Raspberry Pi (and compatible) single
13 # board computers.
14 #
15 # More information on qwiic is at https://www.sparkfun.com/qwiic
16 #
17 # Do you like this library? Help support SparkFun. Buy a board!
18 #
19 #=====
20 # Copyright (c) 2019 SparkFun Electronics
21 #
22 # Permission is hereby granted, free of charge, to any person obtaining a copy
23 # of this software and associated documentation files (the "Software"), to deal
24 # in the Software without restriction, including without limitation the rights
25 # to use, copy, modify, merge, publish, sublicense, and/or sell
26 # copies of the Software, and to permit persons to whom the Software is
27 # furnished to do so, subject to the following conditions:
28 #
29 # The above copyright notice and this permission notice shall be included in all
30 # copies or substantial portions of the Software.
31 #
32 # THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
33 # IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
34 # FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
35 # AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
36 # LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
37 # OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
38 # SOFTWARE.
39 #=====

```

(continues on next page)

(continued from previous page)

```

40 # Example 3
41
42 from __future__ import print_function
43 import qwiic_button
44 import time
45 import sys
46
47 brightness = 250      # The maximum brightness of the pulsing LED. Can be between 0 and
↳255
48 cycle_time = 1000     # The total time for the pulse to take. Set to a bigger number
↳for a slower pulse or a smaller number for a faster pulse
49 off_time = 200        # The total time to stay off between pulses. Set to 0 to be
↳pulsing continuously.
50
51 def run_example():
52
53     print("\nSparkFun Qwiic Button Example 3")
54     my_button = qwiic_button.QwiicButton()
55
56     if my_button.begin() == False:
57         print("\nThe Qwiic Button isn't connected to the system. Please check your
↳connection", \
58             file=sys.stderr)
59         return
60
61     print("\nButton ready!")
62
63     my_button.LED_off()
64
65     while True:
66
67         if my_button.is_button_pressed() == True:
68
69             print("\nThe button is pressed!")
70             my_button.LED_config(brightness, cycle_time, off_time)
71
72         else:
73             print("\nThe button is not pressed.")
74             my_button.LED_off()
75
76         time.sleep(0.02)      # Let's not hammer too hard on the I2C bus
77
78 if __name__ == '__main__':
79     try:
80         run_example()
81     except (KeyboardInterrupt, SystemExit) as exErr:
82         print("\nEnding Example 3")
83         sys.exit(0)

```

## 7.5 Example 4

Listing 4: examples/qwiic\_button\_ex4\_queueUsage.py

```

1  #!/usr/bin/env python
2  #-----
3  # qwiic_button_ex4.py
4  #
5  # Simple Example for the Qwiic Button. Shows how to use the FIFO Queue on the Qwiic_
   ↳ Button.
6  #-----
7  #
8  # Written by Priyanka Makin @ 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) 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
40
41 from __future__ import print_function
42 import qwiic_button
43 import time
44 import sys
45
46 def run_example():
47
48     print("\nSparkFun Qwiic Button Example 4")
49     my_button = qwiic_button.QwiicButton()
50
51     if my_button.begin() == False:
52         print("\nThe Qwiic Button isn't connected to the system. Please check your_
   ↳ connection", \

```

(continues on next page)

(continued from previous page)

```

53         file=sys.stderr)
54     return
55
56     print("\nButton ready!")
57
58     while True:
59
60         # If the queue of pressed events is not empty
61         if my_button.is_pressed_queue_empty() == False:
62             # Then print the time since the last and first button press
63             print("\n" + str(my_button.time_since_last_press() / 1000.0) + "s since_
↪he button was last pressed ")
64             print(str(my_button.time_since_first_press() / 1000.0) + "s since the_
↪button was first pressed ")
65             # If the queue is empty
66         else:
67             print("\nButton Pressed Queue is empty! ")
68
69         # If the queue of clicked events is not empty
70         if my_button.is_clicked_queue_empty() == False:
71             # Then print the time since the last and first button click
72             print("\n" + str(my_button.time_since_last_click() / 1000.0) + "s since_
↪the button was last clicked ")
73             print(str(my_button.time_since_first_click() / 1000.0) + "s since the_
↪button was first clicked")
74             # If the queue is empty
75         else:
76             print("\nButton Clicked Queue is empty!")
77
78         time.sleep(0.02)    # Let's not hammer too hard on the I2C bus
79
80 if __name__ == '__main__':
81     try:
82         run_example()
83     except (KeyboardInterrupt, SystemExit) as exErr:
84         print("\nEnding Example 4")
85         sys.exit(0)

```

## 7.6 Example 5

Listing 5: examples/qwiic\_button\_ex5\_popQueue.py

```

1  #!/usr/bin/env python
2  #-----
3  # qwiic_button_ex5.py
4  #
5  # Simple Example for the Qwiic Button. Shows how to use the FIFO Queue on the Qwiic_
↪Button.
6  #-----
7  #
8  # Written by Priyanka Makin @ 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

```

(continues on next page)

(continued from previous page)

```

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 5
40
41 from __future__ import print_function
42 import qwiic_button
43 import time
44 import sys
45
46 def run_example():
47
48     print("\nSparkFun Qwiic Button Example 5")
49     my_button = qwiic_button.QwiicButton()
50
51     if my_button.begin() == False:
52         print("\nThe Qwiic Button isn't connected to the system. Please check your_
↳connection", \
53             file=sys.stderr)
54         return
55
56     print("\nButton ready!")
57
58     while True:
59
60         print("\nType 'p' to pop a value from the pressed queue.")
61         val = raw_input("Type 'c' to pop a value from the clicked queue: ")
62         # If the character is c or C, then pop a value off of the clicked queue
63         if val == "c" or val == "C":
64             print("\nPopped Clicked Queue! The first value of clicked queue was: ")
65             print(str(my_button.pop_clicked_queue() / 1000.0))
66             # If the character is p or P, then pop a value off of the pressed queue
67             if val == "p" or val == "P":

```

(continues on next page)

(continued from previous page)

```

68         print("\nPopped Pressed Queue!")
69         print(str(my_button.pop_clicked_queue() / 1000.0))
70
71         time.sleep(0.02)      # Let's not hammer too hard on the I2C bus
72
73     if __name__ == '__main__':
74         try:
75             run_example()
76         except (KeyboardInterrupt, SystemExit) as exErr:
77             print("\nEnding Example 5")
78             sys.exit(0)

```

## 7.7 Example 6

Listing 6: examples/qwiic\_button\_ex6\_changeI2CAddress.py

```

1  #!/usr/bin/env python
2  #-----
3  # qwiic_button_ex5.py
4  #
5  # Simple Example for the Qwiic Button. Shows how to change the I2C address of
6  # the Qwiic Button
7  #-----
8  #
9  # Written by Priyanka Makin @ SparkFun Electronics, January 2021
10 #
11 # This python library supports the SparkFun Electronics qwiic
12 # qwiic sensor/board ecosystem on a Raspberry Pi (and compatible) single
13 # board computers.
14 #
15 # More information on qwiic is at https://www.sparkfun.com/qwiic
16 #
17 # Do you like this library? Help support SparkFun. Buy a board!
18 #
19 #=====
20 # Copyright (c) 2019 SparkFun Electronics
21 #
22 # Permission is hereby granted, free of charge, to any person obtaining a copy
23 # of this software and associated documentation files (the "Software"), to deal
24 # in the Software without restriction, including without limitation the rights
25 # to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
26 # copies of the Software, and to permit persons to whom the Software is
27 # furnished to do so, subject to the following conditions:
28 #
29 # The above copyright notice and this permission notice shall be included in all
30 # copies or substantial portions of the Software.
31 #
32 # THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
33 # IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
34 # FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
35 # AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
36 # LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
37 # OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
38 # SOFTWARE.

```

(continues on next page)

(continued from previous page)

```

39 #=====
40 # Example 5
41
42 from __future__ import print_function
43 import qwiic_i2c
44 import qwiic_button
45 import time
46 import sys
47
48 def run_example():
49
50     print("\nSparkFun Qwiic Button Example 6")
51     my_button = qwiic_button.QwiicButton()
52
53     if my_button.begin() == False:
54         print("\nThe Qwiic Button isn't connected to the system. Please check your_
55 ↪connection", \
56             file=sys.stderr)
57         return
58
59     print("\nButton ready!")
60
61     print("\nEnter a new I2C address for the Qwiic Button to use.")
62     print("\nDon't use the 0x prefix. For instance, if you wanted to")
63     print("\nchange the address to 0x5B, you would type 5B and hit enter.")
64
65     new_address = raw_input("\nNew Address: ")
66     new_address = int(new_address, 16)
67
68     # Check if the user entered a valid address
69     if new_address > 0x08 and new_address < 0x77:
70         print("\nCharacters received and new address valid!")
71         print("\nAttempting to set Qwiic Button address...")
72
73         my_button.set_I2C_address(new_address)
74         print("\nAddress successfully changed!")
75         # Check that the Qwiic Button acknowledges on the new address
76         time.sleep(0.02)
77         if my_button.begin() == False:
78             print("\nThe Qwiic Button isn't connected to the system. Please check_
79 ↪your connection", \
80                 file=sys.stderr)
81
82         else:
83             print("\nButton acknowledged on new address!")
84
85     else:
86         print("\nAddress entered not a valid I2C address")
87
88 if __name__ == '__main__':
89     try:
90         run_example()
91     except (KeyboardInterrupt, SystemExit) as exErr:
92         print("\nEnding Example 6")
93         sys.exit(0)

```



## 7.8 Example 7

Listing 7: examples/qwiic\_button\_ex7\_2Buttons.py

```

1  #!/usr/bin/env python
2  #-----
3  # qwiic_button_ex7.py
4  #
5  # Simple Example for the Qwiic Button. Daisy chain together two Qwiic Buttons
6  # with different I2C addresses. This example works for a Qwiic Button with the
7  # default address of 0x6F and one with an address changed to 0x5B. To change
8  # the address of a Qwiic Button, please visit example 6.
9  #-----
10 #
11 # Written by Priynka Makin @ SparkFun Electronics, January 2021
12 #
13 # This python library supports the SparkFun Electronics qwiic
14 # qwiic sensor/board ecosystem on a Raspberry Pi (and compatable) single
15 # board computers.
16 #
17 # More information on qwiic is at https://www.sparkfun.com/qwiic
18 #
19 # Do you like this library? Help support SparkFun. Buy a board!
20 #
21 #=====
22 # Copyright (c) 2019 SparkFun Electronics
23 #
24 # Permission is hereby granted, free of charge, to any person obtaining a copy
25 # of this software and associated documentation files (the "Software"), to deal
26 # in the Software without restriction, including without limitation the rights
27 # to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
28 # copies of the Software, and to permit persons to whom the Software is
29 # furnished to do so, subject to the following conditions:
30 #
31 # The above copyright notice and this permission notice shall be included in all
32 # copies or substantial portions of the Software.
33 #
34 # THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
35 # IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
36 # FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
37 # AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
38 # LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
39 # OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
40 # SOFTWARE.
41 #=====
42 # Example 7
43
44 from __future__ import print_function
45 import qwiic_i2c
46 import qwiic_button
47 import time
48 import sys
49
50 def run_example():
51
52     print("\nSparkFun Qwiic Button Example 7")
53     my_button1 = qwiic_button.QwiicButton()

```

(continues on next page)

(continued from previous page)

```
54     my_button2 = qwiic_button.QwiicButton(0x5B)
55
56     if my_button1.begin() == False:
57         print("\nThe Qwiic Button 1 isn't connected to the system. Please check your_
↪connection", \
58             file=sys.stderr)
59         return
60     if my_button2.begin() == False:
61         print("\nThe Qwiic Button 2 isn't connected to the system. Please check your_
↪connection", \
62             file=sys.stderr)
63         return
64
65     print("\nButton's ready!")
66
67     while 1:
68
69         # Check if button 1 is pressed
70         if my_button1.is_button_pressed() == True:
71             print("\nButton 1 is pressed!")
72
73         # Check if button2 is pressed
74         if my_button2.is_button_pressed() == True:
75             print("\nButton 2 is pressed!")
76
77         time.sleep(0.02)    # Don't hammer too hard on the I2C bus
78
79 if __name__ == '__main__':
80     try:
81         run_example()
82     except (KeyboardInterrupt, SystemExit) as exErr:
83         print("\nEnding Example 7")
84         sys.exit(0)
```

## INDICES AND TABLES

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



## PYTHON MODULE INDEX

### q

`qwiic_button`, [15](#)



## A

`available()` (*qwiic\_button.QwiicButton method*), 16

## B

`begin()` (*qwiic\_button.QwiicButton method*), 16

## C

`clear_event_bits()` (*qwiic\_button.QwiicButton method*), 16

## D

`disable_clicked_interrupt()`  
(*qwiic\_button.QwiicButton method*), 16

`disable_pressed_interrupt()`  
(*qwiic\_button.QwiicButton method*), 16

## E

`enable_clicked_interrupt()`  
(*qwiic\_button.QwiicButton method*), 16

`enable_pressed_interrupt()`  
(*qwiic\_button.QwiicButton method*), 16

## G

`get_debounce_time()` (*qwiic\_button.QwiicButton method*), 17

`get_firmware_version()`  
(*qwiic\_button.QwiicButton method*), 17

`get_I2C_address()` (*qwiic\_button.QwiicButton method*), 16

## H

`has_button_been_clicked()`  
(*qwiic\_button.QwiicButton method*), 17

## I

`is_button_pressed()` (*qwiic\_button.QwiicButton method*), 17

`is_clicked_queue_empty()`  
(*qwiic\_button.QwiicButton method*), 17

`is_clicked_queue_full()`  
(*qwiic\_button.QwiicButton method*), 17

`is_connected()` (*qwiic\_button.QwiicButton method*), 17

`is_pressed_queue_empty()`  
(*qwiic\_button.QwiicButton method*), 17

`is_pressed_queue_full()`  
(*qwiic\_button.QwiicButton method*), 17

## L

`LED_config()` (*qwiic\_button.QwiicButton method*), 15

`LED_off()` (*qwiic\_button.QwiicButton method*), 15

`LED_on()` (*qwiic\_button.QwiicButton method*), 16

## M

module  
qwiic\_button, 15

## P

`pop_clicked_queue()` (*qwiic\_button.QwiicButton method*), 17

`pop_pressed_queue()` (*qwiic\_button.QwiicButton method*), 18

## Q

qwiic\_button  
module, 15

QwiicButton (*class in qwiic\_button*), 15, 19

## R

`reset_interrupt_config()`  
(*qwiic\_button.QwiicButton method*), 18

## S

`set_debounce_time()` (*qwiic\_button.QwiicButton method*), 18

`set_I2C_address()` (*qwiic\_button.QwiicButton method*), 18

## T

`time_since_first_click()`  
(*qwiic\_button.QwiicButton method*), 18

`time_since_first_press()`  
    (*qwiic\_button.QwiicButton method*), [18](#)  
`time_since_last_click()`  
    (*qwiic\_button.QwiicButton method*), [18](#)  
`time_since_last_press()`  
    (*qwiic\_button.QwiicButton method*), [18](#)