
SparkfunQwiicTwist Library Documentation

Release 1.0

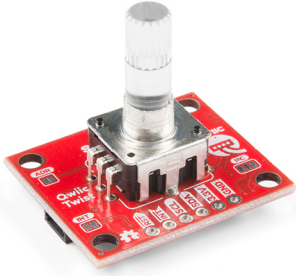
Gaston Williams

Jun 21, 2019

Contents

1	Dependencies	3
1.1	Raspberry Pi Setup	3
1.2	Installing from PyPI	4
2	Usage Example	5
3	Contributing	7
4	Building locally	9
4.1	Zip release files	9
4.2	Sphinx documentation	9
4.3	License Information	10
5	Table of Contents	11
5.1	Simple test	11
5.2	Examples	12
5.3	sparkfun_qwiictwist	24
5.3.1	Implementation Notes	24
6	Indices and tables	27
	Python Module Index	29
	Index	31

CircuitPython library for Sparkfun Qwiic Twist RGB Rotary Encoder. This library is ported from [SparkFun Qwiic Twist Arduino Library](#)



SparkFun Qwiic Twist RGB Rotary Encoder (DEV-15083)

This driver depends on:

- [Adafruit CircuitPython](#)
- [Adafruit Bus Device](#)
- [Qwiic Twist RGB Rotary Encoder Hardware](#)

Please ensure all dependencies are available on the CircuitPython filesystem. This is easily achieved by downloading the Adafruit library and driver bundle.

1.1 Raspberry Pi Setup

Adafruit has an excellent tutorial on [Installing CircuitPython Libraries on Raspberry Pi](#).

Quick Start Summary:

- Start with the latest version of Raspbian with Wifi configured.
- Enable SSH, I2C and SPI.

```
sudo raspi-config
```

- Update your system to the latest version.

```
sudo apt-get update
sudo apt-get upgrade
```

- Update the python tools

```
sudo pip3 install --upgrade setuptools
```

(If pip3 is not installed, install it and rerun the command)

```
sudo apt-get install python3-pip
```

- Install the CircuitPython libraries

```
pip3 install RPI.GPIO  
pip3 install adafruit-blinka
```

1.2 Installing from PyPI

On supported GNU/Linux systems like the Raspberry Pi, you can install the driver locally [from PyPI](#).

Installing this library will also install the dependency `adafruit-circuitpython-busdevice`.

Installing from PyPI

```
pip3 install sparkfun-circuitpython-qwiictwist
```

To install system-wide (this may be required in some cases):

```
sudo pip3 install sparkfun-circuitpython-qwiictwist
```

To install in a virtual environment in your current project:

```
mkdir project-name && cd project-name  
python3 -m venv .env  
source .env/bin/activate  
pip3 install sparkfun-circuitpython-qwiictwist
```


CHAPTER 2

Usage Example

- [Qwiic Twist Hookup Guide](#) - The Arduino examples in the Hookup Guide are available for Python with this library
- [CircuitPython on a Raspberry Pi](#) - Basic information on how to install CircuitPython on a Raspberry Pi.
- Code Example:

```
# import the CircuitPython board and busio libraries
import board
import busio

# Create bus object using the board's I2C port
i2c = busio.I2C(board.SCL, board.SDA)

twist = QwiicTwist(i2c) # default address is 0x3F

# For a different address use QwiicTwist(i2c, address)
# twist = QwiicTwist(i2c, 0x3E)
```


CHAPTER 3

Contributing

Contributions are welcome! Please read our [Code of Conduct](#) before contributing to help this project stay welcoming.

4.1 Zip release files

To build this library locally you'll need to install the `circuitpython-build-tools` package.

```
python3 -m venv .env
source .env/bin/activate
pip install circuitpython-build-tools
```

Once installed, make sure you are in the virtual environment:

```
source .env/bin/activate
```

Then run the build:

```
circuitpython-build-bundles --filename_prefix sparkfun-circuitpython-qwiictwist --
↳library_location .
```

4.2 Sphinx documentation

Sphinx is used to build the documentation based on rST files and comments in the code. First, install dependencies (feel free to reuse the virtual environment from above):

```
python3 -m venv .env
source .env/bin/activate
pip install Sphinx sphinx-rtd-theme
```

Now, once you have the virtual environment activated:

```
cd docs
sphinx-build -E -W -b html . _build/html
```

This will output the documentation to `docs/_build/html`. Open the `index.html` in your browser to view them. It will also (due to `-W`) error out on any warning like Travis will. This is a good way to locally verify it will pass.

4.3 License Information

This product is **open source!**

Please review the `LICENSE.md` file for license information.

Please use, reuse, and modify these files as you see fit.

Please maintain the attributions to SparkFun Electronics and Adafruit and release any derivative under the same license.

Distributed as-is; no warranty is given.

5.1 Simple test

Ensure your device works with this simple test.

Listing 1: examples/qwiictwist_simpletest.py

```
1  # This is example is for the SparkFun Qwiic Twist RGB Rotary Encoder.
2  # SparkFun sells these at its website: www.sparkfun.com
3  # Do you like this library? Help support SparkFun. Buy a board!
4  # https://www.sparkfun.com/products/15083
5
6  """
7  Qwiic Twist Simple Test - qwiictwist_simpletest.py
8  Written by Gaston Williams, June 19th, 2019
9  The Qwiic Twist is a I2C controlled RGB Rotary Encoder
10
11  Simple Test:
12  This program uses the Qwiic Twist CircuitPython Library to change
13  that status of the Qwiic Twist Rotary Encoder.
14  """
15
16  from time import sleep
17  import board
18  import busio
19  import sparkfun_qwiictwist
20
21  # Create bus object using our board's I2C port
22  i2c = busio.I2C(board.SCL, board.SDA)
23
24  # Create joystick object
25  twist = sparkfun_qwiictwist.Sparkfun_QwiicTwist(i2c)
26
27  # Check if connected
```

(continues on next page)

(continued from previous page)

```

28 if twist.connected:
29     print('Twist connected.')
30 else:
31     print('Twist does not appear to be connected. Please check wiring.')
32     exit()
33
34 # Print firmware version and current status
35 print('Firmware version ' + twist.version)
36
37 # Turn the relay on and off
38 print('Press Ctrl-C to exit program')
39
40 try:
41     while True:
42         print('Count: ' + str(twist.count))
43         if twist.pressed:
44             print('Pressed!')
45             sleep(0.5)
46
47 except KeyboardInterrupt:
48     pass

```

5.2 Examples

1. Basic Readings - Print number of steps knob has been twisted.

Listing 2: examples/example1_basic_readings.py

```

1 # This is example is for the SparkFun Qwiic Single Twist.
2 # SparkFun sells these at its website: www.sparkfun.com
3 # Do you like this library? Help support SparkFun. Buy a board!
4 # https://www.sparkfun.com/products/15083
5
6 """
7 Qwiic Twist Example 1 - example1_basic_readings.py
8 Written by Gaston Williams, June 19th, 2019
9 Based on Arduino code written by
10 Nathan Seidle @ Sparkfun, December 3rd, 2018
11 The Qwiic Twist is an I2C controlled RGB Rotary Encoder produced by sparkfun
12
13 Example 1 - Basic Readings:
14 This program uses the Qwiic Twist CircuitPython Library to
15 control the Qwiic Twist RGB Rotrary Encoder over I2C and print
16 the number of steps the encoder has been twisted.
17 """
18
19 from time import sleep
20 import board
21 import busio
22 import sparkfun_qwiictwist
23
24 # Create bus object using our board's I2C port
25 i2c = busio.I2C(board.SCL, board.SDA)
26

```

(continues on next page)

(continued from previous page)

```

27 # Create twist object
28 twist = sparkfun_qwiictwist.Sparkfun_QwiicTwist(i2c)
29
30 print('Qwiic Twist Example 1 Basic Readings')
31
32 # Check if connected
33 if twist.connected:
34     print('Twist connected.')
35 else:
36     print('Twist does not appear to be connected. Please check wiring.')
37     exit()
38
39 print('Type Ctrl-C to exit program.')
40
41 try:
42     while True:
43         print('Count: ' + str(twist.count))
44         if twist.pressed:
45             print('Pressed!')
46             sleep(0.5)
47
48 except KeyboardInterrupt:
49     pass

```

2. Set Color - Set the knob color.

Listing 3: examples/example2_set_color.py

```

1 # This is example is for the SparkFun Qwiic Single Twist.
2 # SparkFun sells these at its website: www.sparkfun.com
3 # Do you like this library? Help support SparkFun. Buy a board!
4 # https://www.sparkfun.com/products/15083
5
6 """
7 Qwiic Twist Example 2 - example2_set_color.py
8 Written by Gaston Williams, June 20th, 2019
9 Based on Arduino code written by
10 Nathan Seidle @ Sparkfun, December 3rd, 2018
11 The Qwiic Twist is an I2C controlled RGB Rotary Encoder produced by sparkfun
12
13 Example 2 - Set Color:
14 This program uses the Qwiic Twist CircuitPython Library to
15 control the Qwiic Twist RGB Rotary Encoder over I2C to set
16 the knob color to pink. This value is stored in the Qwiic Twist
17 and will be loaded after each power-on.
18 """
19
20 from time import sleep
21 import board
22 import busio
23 import sparkfun_qwiictwist
24
25 # Create bus object using our board's I2C port
26 i2c = busio.I2C(board.SCL, board.SDA)
27
28 # Create twist object
29 twist = sparkfun_qwiictwist.Sparkfun_QwiicTwist(i2c)

```

(continues on next page)

(continued from previous page)

```

30
31 print('Qwiic Twist Example 2 Set Color')
32
33 # Check if connected
34 if twist.connected:
35     print('Twist connected.')
36 else:
37     print('Twist does not appear to be connected. Please check wiring.')
38     exit()
39
40 print('Type Ctrl-C to exit program.')
41
42 # Turn off any color connections
43 twist.connect_color(0, 0, 0)
44
45 # Set the knob color to pink (r =100, g=10, b=50)
46 twist.set_color(100, 10, 50)
47
48 # Set a flag to toggle color
49 is_pink = True
50
51 try:
52     while True:
53         print('Count: ' + str(twist.count))
54         if twist.pressed:
55             print('Pressed!')
56             if is_pink:
57                 # Set the knob color to blue (r =10, g=10, b=100)
58                 print('Change color to blue.')
59                 twist.set_color(10, 10, 100)
60                 is_pink = False
61             else:
62                 # Set the knob color to pink (r =100, g=10, b=50)
63                 print('Change color to pink.')
64                 twist.set_color(100, 10, 50)
65                 is_pink = True
66
67         sleep(0.2)
68
69 except KeyboardInterrupt:
70     pass

```

3. Crazy Color - Set the knob to random colors.

Listing 4: examples/example3_crazy_color.py

```

1 # This is example is for the SparkFun Qwiic Single Twist.
2 # SparkFun sells these at its website: www.sparkfun.com
3 # Do you like this library? Help support SparkFun. Buy a board!
4 # https://www.sparkfun.com/products/15083
5
6 """
7 Qwiic Twist Example 3 - example3_crazy_color.py
8 Written by Gaston Williams, June 20th, 2019
9 Based on Arduino code written by
10 Nathan Seidle @ Sparkfun, December 3rd, 2018
11 The Qwiic Twist is an I2C controlled RGB Rotary Encoder produced by sparkfun

```

(continues on next page)

(continued from previous page)

```

12
13 Example 3 - Crazy Color:
14 This program uses the Qwiic Twist CircuitPython Library to
15 control the Qwiic Twist RGB Rotary Encoder over I2C to set
16 the knob color to an endless number of random colors.
17 """
18
19 from time import sleep
20 import random
21
22 import board
23 import busio
24 import sparkfun_qwiictwist
25
26 # Create bus object using our board's I2C port
27 i2c = busio.I2C(board.SCL, board.SDA)
28
29 # Create twist object
30 twist = sparkfun_qwiictwist.Sparkfun_QwiicTwist(i2c)
31
32 print('Qwiic Twist Example 3 Crazy Color')
33
34 # Check if connected
35 if twist.connected:
36     print('Twist connected.')
37 else:
38     print('Twist does not appear to be connected. Please check wiring.')
39     exit()
40
41 print('Type Ctrl-C to exit program.')
42
43 # Turn off any color connections
44 twist.connect_color(0, 0, 0)
45
46 try:
47     while True:
48         print('Count: ' + str(twist.count))
49         if twist.pressed:
50             print('Pressed!')
51
52         # Generate a random rgb value
53         red = random.randint(0, 256)
54         green = random.randint(0, 256)
55         blue = random.randint(0, 256)
56         twist.set_color(red, green, blue)
57
58         sleep(0.1)
59
60 except KeyboardInterrupt:
61     pass

```

4. Connect Color - Change the knob color as it is twisted.

Listing 5: examples/example4_connect_colors.py

```

1 # This is example is for the SparkFun Qwiic Single Twist.
2 # SparkFun sells these at its website: www.sparkfun.com

```

(continues on next page)

(continued from previous page)

```
3 # Do you like this library? Help support SparkFun. Buy a board!
4 # https://www.sparkfun.com/products/15083
5
6 """
7 Qwiic Twist Example 4 - example4_connect_colors.py
8 Written by Gaston Williams, June 19th, 2019
9 Based on Arduino code written by
10 Nathan Seidle @ Sparkfun, December 3rd, 2018
11 The Qwiic Twist is an I2C controlled RGB Rotary Encoder produced by sparkfun
12
13 Example 4 - Connect Colors:
14 This program uses the Qwiic Twist CircuitPython Library to
15 control the Qwiic Twist RGB Rotrary Encoder over I2C to make
16 the knob change color as the user turns the device.
17 We don't have to send a setColor() or setRed() command each time,
18 Qwiic Twist can change its color independently of the master.
19 By connecting a -10 value to red, the red LED will go down 10 in
20 brightness with each encoder tick. Connecting 10 to blue will
21 increase the blue value with each tick. These values are stored
22 in the Qwiic Twist and will be loaded after each power-on.
23 """
24
25 from time import sleep
26 import board
27 import busio
28 import sparkfun_qwiictwist
29
30 # Create bus object using our board's I2C port
31 i2c = busio.I2C(board.SCL, board.SDA)
32
33 # Create twist object
34 twist = sparkfun_qwiictwist.Sparkfun_QwiicTwist(i2c)
35
36 print('Qwiic Twist Example 4 Connect Colors')
37
38 # Check if connected
39 if twist.connected:
40     print('Twist connected.')
41 else:
42     print('Twist does not appear to be connected. Please check wiring.')
43     exit()
44
45 print('Type Ctrl-C to exit program.')
46
47 # Set Red and Blue LED brightnesses to half of max.
48 twist.set_color(128, 0, 128)
49
50 #Set the individual color connections
51
52 # Red LED will go down 10 in brightness with each encoder tick
53 twist.red_connection = -10
54 # Blue LED will go up 10 in brightness with each encoder tick
55 twist.blue_connection = 10
56
57 # Or use the function below to set all color connections at once
58 # twist.connect_color(-10, 0, 10)
59
```

(continues on next page)

(continued from previous page)

```

60 try:
61     while True:
62         print('Count: ' + str(twist.count))
63         if twist.pressed:
64             print('Pressed!')
65             sleep(0.5)
66
67 except KeyboardInterrupt:
68     pass

```

5. Timestamps - Show timestamps for when knob is twisted or pressed.

Listing 6: examples/example5_timestamps.py

```

1  # This is example is for the SparkFun Qwiic Single Twist.
2  # SparkFun sells these at its website: www.sparkfun.com
3  # Do you like this library? Help support SparkFun. Buy a board!
4  # https://www.sparkfun.com/products/15083
5
6  """
7  Qwiic Twist Example 5 - example5_timestamps.py
8  Written by Gaston Williams, June 21st, 2019
9  Based on Arduino code written by
10 Nathan Seidle @ Sparkfun, December 3rd, 2018
11 The Qwiic Twist is an I2C controlled RGB Rotary Encoder produced by sparkfun
12
13 Example 5 - Timestamps:
14 This program uses the Qwiic Twist CircuitPython Library to
15 control the Qwiic Twist RGB Rotrary Encoder over I2C to display
16 the time between when the user did an action (such as twist
17 knob or press button) and when we queried the device. You don't
18 need to constantly poll the Qwiic Twist to see if the user has
19 twisted the knob or pressed the button. Instead, check every so
20 often and when the isMoved or isClick goes true, then read the
21 timestamp and you'll know when the user did their thing.
22 """
23
24 from time import sleep
25 import board
26 import busio
27 import sparkfun_qwiictwist
28
29 # Create bus object using our board's I2C port
30 i2c = busio.I2C(board.SCL, board.SDA)
31
32 # Create twist object
33 twist = sparkfun_qwiictwist.Sparkfun_QwiicTwist(i2c)
34
35 print('Qwiic Twist Example 5 Timestamps')
36
37 # Check if connected
38 if twist.connected:
39     print('Twist connected.')
40 else:
41     print('Twist does not appear to be connected. Please check wiring.')
42     exit()
43

```

(continues on next page)

(continued from previous page)

```

44 print('Type Ctrl-C to exit program.')
45
46 try:
47     while True:
48         print('Count: ' + str(twist.count))
49         if twist.moved:
50             print('Last Twist time: ' + str(twist.time_since_last_movement))
51         if twist.clicked:
52             print('Last Button time: ' + str(twist.time_since_last_press))
53         if twist.pressed:
54             print('Pressed!')
55         sleep(1)
56
57 except KeyboardInterrupt:
58     pass

```

6. Difference - Show the difference between encoder counts when knob is twisted.

Listing 7: examples/example6_difference.py

```

1  # This is example is for the SparkFun Qwiic Single Twist.
2  # SparkFun sells these at its website: www.sparkfun.com
3  # Do you like this library? Help support SparkFun. Buy a board!
4  # https://www.sparkfun.com/products/15083
5
6  """
7  Qwiic Twist Example 6 - example6_difference.py
8  Written by Gaston Williams, June 19th, 2019
9  Based on Arduino code written by
10 Nathan Seidle @ Sparkfun, December 3rd, 2018
11 The Qwiic Twist is an I2C controlled RGB Rotary Encoder produced by sparkfun
12
13 Example 6 - Difference:
14 This program uses the Qwiic Twist CircuitPython Library to
15 control the Qwiic Twist RGB Rotrary Encoder over I2C to display
16 the difference since the last reading. This is helpful if you
17 don't care what the cumulative value is, just difference.
18
19 Things like volume control, brightness, etc. your system may
20 not need to know an absolute value like 417, but instead that
21 the user has moved the encoder 4 ticks since the last reading.
22 """
23
24 from time import sleep
25 import board
26 import busio
27 import sparkfun_qwiictwist
28
29 # Create bus object using our board's I2C port
30 i2c = busio.I2C(board.SCL, board.SDA)
31
32 # Create twist object
33 twist = sparkfun_qwiictwist.Sparkfun_QwiicTwist(i2c)
34
35 print('Qwiic Twist Example 6 Difference')
36
37 # Check if connected

```

(continues on next page)

(continued from previous page)

```

38 if twist.connected:
39     print('Twist connected.')
40 else:
41     print('Twist does not appear to be connected. Please check wiring.')
42     exit()
43
44 print('Type Ctrl-C to exit program.')
45
46 try:
47     while True:
48         print('Count: ' + str(twist.count))
49         print('Difference: ' + str(twist.difference))
50         sleep(0.250)
51
52 except KeyboardInterrupt:
53     pass

```

7. Set Count - Change the encoder count reported when knob is twisted.

Listing 8: examples/example7_set_count.py

```

1  # This is example is for the SparkFun Qwiic Single Twist.
2  # SparkFun sells these at its website: www.sparkfun.com
3  # Do you like this library? Help support SparkFun. Buy a board!
4  # https://www.sparkfun.com/products/15083
5
6  """
7  Qwiic Twist Example 7 - example7_set_count.py
8  Written by Gaston Williams, June 21st, 2019
9  Based on Arduino code written by
10 Nathan Seidle @ Sparkfun, December 3rd, 2018
11 The Qwiic Twist is an I2C controlled RGB Rotary Encoder produced by sparkfun
12
13 Example 7 - Set Count:
14 This program uses the Qwiic Twist CircuitPython Library to
15 control the Qwiic Twist RGB Rotrary Encoder over I2C to set
16 the encoder count. There are times when it's necessary encoder
17 count to a specific value. Useful when the encoder value is mapped directly
18 onto a volume setting, FM freq, etc.
19 """
20
21 from time import sleep
22 import board
23 import busio
24 import sparkfun_qwiictwist
25
26 # Create bus object using our board's I2C port
27 i2c = busio.I2C(board.SCL, board.SDA)
28
29 # Create twist object
30 twist = sparkfun_qwiictwist.Sparkfun_QwiicTwist(i2c)
31
32 print('Qwicc Twist Example 7 Set Count')
33
34 # Check if connected
35 if twist.connected:
36     print('Twist connected.')

```

(continues on next page)

(continued from previous page)

```

37 else:
38     print('Twist does not appear to be connected. Please check wiring.')
39     exit()
40
41 # Set initial value to 1000. Not stored to non-volatile memory.
42 twist.count = 1000
43
44 print('Type Ctrl-C to exit program.')
45
46 try:
47     while True:
48         print('Count: ' + str(twist.count))
49         print('Difference: ' + str(twist.difference))
50         sleep(0.250)
51
52 except KeyboardInterrupt:
53     pass

```

8. Interrupts - Enable button and twist interrupts.

Listing 9: examples/example8_interruptst.py

```

1 # This is example is for the SparkFun Qwiic Single Twist.
2 # SparkFun sells these at its website: www.sparkfun.com
3 # Do you like this library? Help support SparkFun. Buy a board!
4 # https://www.sparkfun.com/products/15083
5
6 """
7 Qwiic Twist Example 8 - example8_interrupts.py
8 Written by Gaston Williams, June 19th, 2019
9 Based on Arduino code written by
10 Nathan Seidle @ Sparkfun, December 3rd, 2018
11 The Qwiic Twist is an I2C controlled RGB Rotary Encoder produced by sparkfun
12
13 Example 8 - Interrupts:
14 This program uses the Qwiic Twist CircuitPython Library to
15 control the Qwiic Twist RGB Rotary Encoder over I2C to enable
16 the button and twist interrupts. Once an interrupt is read,
17 it is cleared by the library. For this example you will need to
18 connect the INT pin on Qwiic to GPIO D6 on the Raspberry Pi.
19
20 The interrupt will not fire until 250ms after the user has stopped
21 turning the encoder. This is so the master is not overwhelmed with
22 interrupts while the user is still turning the dial.
23 """
24
25 # from time import sleep
26 import board
27 import digitalio
28 import busio
29 import sparkfun_qwiictwist
30
31 # Create bus object using our board's I2C port
32 i2c = busio.I2C(board.SCL, board.SDA)
33
34 # Set up Interrupt pin on GPIO D6 with a pull-up resistor
35 twist_interrupt_pin = digitalio.DigitalInOut(board.D6)

```

(continues on next page)

(continued from previous page)

```

36 twist_interrupt_pin.direction = digitalio.Direction.INPUT
37 twist_interrupt_pin.pull = digitalio.Pull.UP
38
39 # Create twist object
40 twist = sparkfun_qwiictwist.Sparkfun_QwiicTwist(i2c)
41
42 print('Qwiic Twist Example 8 Interrupts')
43
44 # Check if connected
45 if twist.connected:
46     print('Twist connected.')
47 else:
48     print('Twist does not appear to be connected. Please check wiring.')
49     exit()
50
51 # Optional: You can modify the time between when the user has stopped turning
52 # and when interrupt is raised
53
54 # Set twist timeout to 500ms before interrupt assertion
55 # twist.int_timeout = 500
56
57 print('Type Ctrl-C to exit program.')
58
59 try:
60     while True:
61         # When the interrupt goes low
62         if not twist_interrupt_pin.value:
63             print('Interrupt:')
64             if twist.moved:
65                 print('Count: ' + str(twist.count))
66             if twist.pressed:
67                 print('Pressed!')
68             if twist.clicked:
69                 print('Clicked!')
70             twist.clear_interrupts()
71
72 except KeyboardInterrupt:
73     pass

```

9. Change I2C Address - Change the device I2C address.

Listing 10: examples/example9_change_i2c_address.py

```

1 # This is example is for the SparkFun Qwiic Twist.
2 # SparkFun sells these at its website: www.sparkfun.com
3 # Do you like this library? Help support SparkFun. Buy a board!
4 # https://www.sparkfun.com/products/15083
5
6 """
7 Qwiic Twist Example 9 - example9_change_i2c_address.py
8 Written by Gaston Williams, June 13th, 2019
9 Based on Arduino code written by
10 Nathan Seidle @ SparkFun Electronics, December 3rd
11 , 2019
12 The Qwiic Twist is an I2C controlled RGB Rotary Encoder
13
14 Example 9 - Change I2C Address and Read Firmware Version:

```

(continues on next page)

(continued from previous page)

```

15  This program uses the Qwiic Twist CircuitPython Library to change
16  the I2C address for the device. You enter in the DEC value (8-119) or
17
18  HEX value (0x08-0x77) for the new Twist address. The address is
19  changed and the firmware version is printed out to validate the connection.
20  You can run example10+i2c_scanner.py to validate the address after the change.
21
22  Syntax: python3 change_i2c_address.py [address]
23  where address is an optional address value in decimal or hex
24  The default value for the address is 63 [0x3F]
25  """
26
27  import sys
28  import board
29  import busio
30  import sparkfun_qwiictwist
31
32  # The default QwiicTwist i2c address is 0x3F (63)
33  i2c_address = 0x3F
34
35  # print('Argument count: ', len(sys.argv))
36  # print('List: ' + str(sys.argv))
37
38  # If we were passed an argument, then use it as the address
39  if len(sys.argv) > 1:
40      try:
41          # check to see if hex or decimal argument
42          if '0x' in sys.argv[1]:
43              i2c_address = int(sys.argv[1], 16)
44          else:
45              i2c_address = int(sys.argv[1])
46      except ValueError:
47          print('Ignoring invalid argument: ' + str(sys.argv[1]))
48
49  # Show the initial address
50  print('Current i2c address = ' + str(i2c_address)
51        + ' [' + hex(i2c_address) + ']')
52
53  # Create library object using our Bus I2C port
54  i2c = busio.I2C(board.SCL, board.SDA)
55  twist = sparkfun_qwiictwist.Sparkfun_QwiicTwist(i2c, i2c_address)
56
57  if twist.connected:
58      print('Qwiic Twist Example.')
59  else:
60      # if we can't connect, something is wrong so just quit
61      print('Twist does not appear to be connected. Please check wiring.')
62      exit()
63
64  print('Address: ' + str(i2c_address) + ' [' + hex(i2c_address) + ']'
65        + ' Version: ' + twist.version)
66
67  text = input('Enter a new I2C address (as a decimal from 8 to 119 or hex 0x08 to
68  ↪0x77):')
69
70  # check to see if hex or decimal value
71  if '0x' in text:

```

(continues on next page)

(continued from previous page)

```

71     new_address = int(text, 16)
72 else:
73     new_address = int(text)
74
75 print('Changing address to ' + str(new_address)
76       + ' [' + hex(new_address) + ']')
77
78 result = twist.change_address(new_address)
79
80 if result:
81     print('Address changed to ' + str(new_address)
82           + ' [' + hex(new_address) + ']')
83     # After the change check the new connection and show firmware version
84     if twist.connected:
85         print('Connected to Twist after address change.')
86         print('Firmware Version: ' + twist.version)
87     else:
88         print('Error after address change. Cannot connect to Twist.')
89
90 else:
91     print('Address change failed.')
92
93 # good advice whether the address changed worked or not
94 print('Run example10_i2c_scanner.py to verify the Qwiic Twist address.')

```

10. I2C Scanner - Scan the IC2 bus for devices.

Listing 11: examples/example10_i2c_scanner.py

```

1  # This is example is for the SparkFun Qwiic Twist.
2  # SparkFun sells these at its website: www.sparkfun.com
3  # Do you like this library? Help support SparkFun. Buy a board!
4  # https://www.sparkfun.com/products/15083
5
6  """
7  Qwiic Twist Example 10 - example10_i2c_Scanner.py
8  Written by Gaston Williams, June 13th, 2019
9  The Qwiic Twist is an I2C controlled RGB Rotary Encoder
10
11 Example 10 - I2C Scanner
12 This program uses CircuitPython BusIO library to find the current
13 address of the Qwiic Twist. It uses the I2C Scanner Example from
14 https://learn.adafruit.com/circuitpython-basics-i2c-and-spi/i2c-devices
15
16 The factory default address is 0x3F.
17 """
18
19 import time
20
21 import board
22 import busio
23
24 i2c = busio.I2C(board.SCL, board.SDA)
25
26 while not i2c.try_lock():
27     pass
28

```

(continues on next page)

(continued from previous page)

```

29 print('Press Ctrl-C to exit program')
30
31 try:
32     while True:
33         print('I2C addresses found:',
34               [hex(device_address) for device_address in i2c.scan()])
35         time.sleep(5)
36 except KeyboardInterrupt:
37     pass
38
39 finally:
40     i2c.unlock()

```

5.3 sparkfun_qwiictwist

CircuitPython library for the Sparkfun Qwiic Twist Rotary Encoder

- Author(s): Gaston Williams

5.3.1 Implementation Notes

Hardware:

- This library is for the SparkFun Qwiic Joystick.
- SparkFun sells these at its website: www.sparkfun.com
- Do you like this library? Help support SparkFun. Buy a board! <https://www.sparkfun.com/products/15083>

Software and Dependencies:

- Adafruit CircuitPython firmware for the supported boards: <https://github.com/adafruit/circuitpython/releases>
- Adafruit's Bus Device library: https://github.com/adafruit/Adafruit_CircuitPython_BusDevice

class `sparkfun_qwiictwist.Sparkfun_QwiicTwist` (*i2c*, *address=63*, *debug=False*)
 CircuitPython class for the Sparkfun QwiicTwist RGB Rotary Encoder

blue

Get the value of the blue LED

blue_connection

Get the value of the blue LED connection.

change_address (*new_address*)

Change the i2c address of Twist Rotary Encoder and return True if successful.

clear_interrupts ()

Clears the moved, clicked, and pressed bits

clicked

Return true if a click event has occurred. Event flag is then reset.

connect_color (*red_value*, *green_value*, *blue_value*)

Connect all the rgb color for the encoder LEDs

connected

Check the id of Rotary Encoder. Returns True if successful.

count

Returns the number of indents since the user turned the knob.

difference

Return the difference in number of clicks since previous check. The value is cleared after it is read.

green

Get the value of the green LED.

green_connection

Get the value of the green LED connection.

int_timeout

Get number of milliseconds that elapse between the end of the knob turning and interrupt firing.

moved

Return true if the knob has been twisted.

pressed

“Return true if button is currently pressed.

red

Get the value of the red LED.

red_connection

Get the value of the red LED connection

set_color (*red_value*, *green_value*, *blue_value*)

Set the rgb color of the encoder LEDs

time_since_last_movement

Return the number of milliseconds since the last encoder movement

time_since_last_press

Return the number of milliseconds since the last button press and release

version

Return the version string for the Twist firmware.

CHAPTER 6

Indices and tables

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

S

`sparkfun_qwiictwist`, 24

B

blue (*sparkfun_qwiictwist.Sparkfun_QwiicTwist* attribute), 24

blue_connection (*sparkfun_qwiictwist.Sparkfun_QwiicTwist* attribute), 24

C

change_address() (*sparkfun_qwiictwist.Sparkfun_QwiicTwist* method), 24

clear_interrupts() (*sparkfun_qwiictwist.Sparkfun_QwiicTwist* method), 24

clicked (*sparkfun_qwiictwist.Sparkfun_QwiicTwist* attribute), 24

connect_color() (*sparkfun_qwiictwist.Sparkfun_QwiicTwist* method), 24

connected (*sparkfun_qwiictwist.Sparkfun_QwiicTwist* attribute), 24

count (*sparkfun_qwiictwist.Sparkfun_QwiicTwist* attribute), 24

D

difference (*sparkfun_qwiictwist.Sparkfun_QwiicTwist* attribute), 25

G

green (*sparkfun_qwiictwist.Sparkfun_QwiicTwist* attribute), 25

green_connection (*sparkfun_qwiictwist.Sparkfun_QwiicTwist* attribute), 25

I

int_timeout (*sparkfun_qwiictwist.Sparkfun_QwiicTwist* attribute), 25

M

moved (*sparkfun_qwiictwist.Sparkfun_QwiicTwist* attribute), 25

P

pressed (*sparkfun_qwiictwist.Sparkfun_QwiicTwist* attribute), 25

R

red (*sparkfun_qwiictwist.Sparkfun_QwiicTwist* attribute), 25

red_connection (*sparkfun_qwiictwist.Sparkfun_QwiicTwist* attribute), 25

S

set_color() (*sparkfun_qwiictwist.Sparkfun_QwiicTwist* method), 25

Sparkfun_QwiicTwist (class in *sparkfun_qwiictwist*), 24

sparkfun_qwiictwist (module), 24

T

time_since_last_movement (*sparkfun_qwiictwist.Sparkfun_QwiicTwist* attribute), 25

time_since_last_press (*sparkfun_qwiictwist.Sparkfun_QwiicTwist* attribute), 25

V

version (*sparkfun_qwiictwist.Sparkfun_QwiicTwist* attribute), 25