
sparkfun*gpio.py*
Release 0.0.2

Mar 05, 2021

Contents:

1	Contents	3
2	Supported Platforms	5
3	Dependencies	7
4	Documentation	9
5	Installation	11
5.1	PyPi Installation	11
5.2	Local Installation	11
6	Example Use	13
7	Table of Contents	15
7.1	API Reference	15
7.1.1	qwiiic_gpio	15
7.2	Example One	16
7.3	Example 2	18
7.4	Example 3	20
7.5	Example 4	22
8	Indices and tables	25
	Python Module Index	27
	Index	29

Python module for the Qwiic GPIO

- [SparkFun Qwiic GPIO](#)

This package can be used in conjunction with the overall [SparkFun qwiic Python Package](#)

New to qwiic? Take a look at the entire [SparkFun qwiic ecosystem](#).

CHAPTER 1

Contents

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

CHAPTER 2

Supported Platforms

The Qwiic GPIO Python package current supports the following platforms:

- [Raspberry Pi](#)

CHAPTER 3

Dependencies

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

CHAPTER 4

Documentation

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

5.1 PyPi Installation

This repository is hosted on PyPi as the `sparkfun-qwiic-gpio` 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-gpio
```

For the current user:

```
pip install sparkfun-qwiic-gpio
```

5.2 Local Installation

To install, make sure the `setuptools` package is installed on the system.

Direct installation at the command line:

```
python setup.py install
```

To build a package for use with `pip`:

```
python setup.py sdist
```

A package file is built and placed in a subdirectory called `dist`. This package file can be installed using `pip`.

```
cd dist  
pip install sparkfun_qwiic_gpio-<version>.tar.gz
```


CHAPTER 6

Example Use

See the examples directory for more detailed use examples.

```
from __future__ import print_function
import qwiic_gpio
import time
import sys

def runExample():

    print("\nSparkFun Qwiic GPIO Example 1\n")
    myGPIO = qwiic_gpio.QwiicGPIO()

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

    myGPIO.begin()
    myGPIO.mode_0 = myGPIO.GPIO_OUT
    myGPIO.mode_1 = myGPIO.GPIO_OUT
    myGPIO.mode_2 = myGPIO.GPIO_OUT
    myGPIO.mode_3 = myGPIO.GPIO_OUT
    myGPIO.mode_4 = myGPIO.GPIO_OUT
    myGPIO.mode_5 = myGPIO.GPIO_OUT
    myGPIO.mode_6 = myGPIO.GPIO_OUT
    myGPIO.mode_7 = myGPIO.GPIO_OUT
    myGPIO.setMode()

    while True:
        myGPIO.out_status_0 = myGPIO.GPIO_HI
        myGPIO.out_status_1 = myGPIO.GPIO_HI
        myGPIO.out_status_2 = myGPIO.GPIO_HI
        myGPIO.out_status_3 = myGPIO.GPIO_HI
```

(continues on next page)

(continued from previous page)

```
myGPIO.out_status_4 = myGPIO.GPIO_HI
myGPIO.out_status_5 = myGPIO.GPIO_HI
myGPIO.out_status_6 = myGPIO.GPIO_HI
myGPIO.out_status_7 = myGPIO.GPIO_HI
myGPIO.setGPIO()
print("set hi")
time.sleep(1)
myGPIO.out_status_0 = myGPIO.GPIO_LO
myGPIO.out_status_1 = myGPIO.GPIO_LO
myGPIO.out_status_2 = myGPIO.GPIO_LO
myGPIO.out_status_3 = myGPIO.GPIO_LO
myGPIO.out_status_4 = myGPIO.GPIO_LO
myGPIO.out_status_5 = myGPIO.GPIO_LO
myGPIO.out_status_6 = myGPIO.GPIO_LO
myGPIO.out_status_7 = myGPIO.GPIO_LO
myGPIO.setGPIO()
print("set lo")
time.sleep(1)

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

7.1 API Reference

7.1.1 qwiic_gpio

Python module for the Qwiic GPIO.

This python package is a port of the existing [SparkFun GPIO Arduino Library](https://github.com/sparkfun/SparkFun_gpio_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_gpio.QwiicGPIO (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

```
begin ()
```

Initialize the operation of the Qwiic GPIO

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

Return type bool

```
getGPIO ()
```

Updates mode_X variables with values from Qwiic GPIO

Returns The value of the mode register.

Return type 8 bit unsigned integer

getInversion ()

Updates inversion_X variables with values from Qwiic GPIO

Returns The value of the inversion register.

Return type 8 bit unsigned integer

getMode ()

Updates mode_X variables with values from Qwiic GPIO

Returns The value of the mode register.

Return type 8 bit unsigned integer

isConnected ()

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

Returns True if the device is connected, otherwise False.

Return type bool

setGPIO ()

Send all current output settings to the GPIO. This should be called after calling myGPIO.out_status_0 = myGPIO.GPIO_HI to set the GPIO.

Returns No return value

setInversion ()

Send the inversion modes of all pins. This function must be called after editing modes using myGPIO.inversion_0 = myGPIO.INVERT

Returns No return value

setMode ()

Sends all 8 pin modes (input or output) to the GPIO to set all 8 pins. Setting the value to input or output is done using myGPIO.mode_0 = myGPIO.GPIO_OUT

Returns No return value

7.2 Example One

Listing 1: examples/qwiic_gpio_ex1.py

```

1  #!/usr/bin/env python
2  #-----
3  # qwiic_gpio_ex1.py
4  #
5  # Simple Example for the Qwiic GPIO Device, toggles GPIO 0 on and off.
6  #-----
7  #
8  # Written by SparkFun Electronics, May 2019
9  #
10 # This python library supports the SparkFun Electronics qwiic
11 # qwiic sensor/board ecosystem on a Raspberry Pi (and compatible) single
12 # board computers.
13 #
14 # More information on qwiic is at https://www.sparkfun.com/qwiic
15 #

```

(continues on next page)

(continued from previous page)

```

16 # Do you like this library? Help support SparkFun. Buy a board!
17 #
18 #=====
19 # Copyright (c) 2019 SparkFun Electronics
20 #
21 # Permission is hereby granted, free of charge, to any person obtaining a copy
22 # of this software and associated documentation files (the "Software"), to deal
23 # in the Software without restriction, including without limitation the rights
24 # to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
25 # copies of the Software, and to permit persons to whom the Software is
26 # furnished to do so, subject to the following conditions:
27 #
28 # The above copyright notice and this permission notice shall be included in all
29 # copies or substantial portions of the Software.
30 #
31 # THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
32 # IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
33 # FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
34 # AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
35 # LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
36 # OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
37 # SOFTWARE.
38 #=====
39 # Example 1
40 #
41
42 from __future__ import print_function
43 import qwiic_gpio
44 import time
45 import sys
46
47 def runExample():
48
49     print("\nSparkFun Qwiic GPIO Example 1\n")
50     myGPIO = qwiic_gpio.QwiicGPIO()
51
52     if myGPIO.isConnected() == False:
53         print("The Qwiic GPIO isn't connected to the system. Please check your
54 ↪connection", \
55             file=sys.stderr)
56         return
57
58     myGPIO.begin()
59     myGPIO.mode_0 = myGPIO.GPIO_OUT
60     myGPIO.mode_1 = myGPIO.GPIO_OUT
61     myGPIO.mode_2 = myGPIO.GPIO_OUT
62     myGPIO.mode_3 = myGPIO.GPIO_OUT
63     myGPIO.mode_4 = myGPIO.GPIO_OUT
64     myGPIO.mode_5 = myGPIO.GPIO_OUT
65     myGPIO.mode_6 = myGPIO.GPIO_OUT
66     myGPIO.mode_7 = myGPIO.GPIO_OUT
67     myGPIO.setMode()
68
69     while True:
70         myGPIO.out_status_0 = myGPIO.GPIO_HI
71         myGPIO.out_status_1 = myGPIO.GPIO_HI
72         myGPIO.out_status_2 = myGPIO.GPIO_HI

```

(continues on next page)

(continued from previous page)

```

72     myGPIO.out_status_3 = myGPIO.GPIO_HI
73     myGPIO.out_status_4 = myGPIO.GPIO_HI
74     myGPIO.out_status_5 = myGPIO.GPIO_HI
75     myGPIO.out_status_6 = myGPIO.GPIO_HI
76     myGPIO.out_status_7 = myGPIO.GPIO_HI
77     myGPIO.setGPIO()
78     print("set hi")
79     time.sleep(1)
80     myGPIO.out_status_0 = myGPIO.GPIO_LO
81     myGPIO.out_status_1 = myGPIO.GPIO_LO
82     myGPIO.out_status_2 = myGPIO.GPIO_LO
83     myGPIO.out_status_3 = myGPIO.GPIO_LO
84     myGPIO.out_status_4 = myGPIO.GPIO_LO
85     myGPIO.out_status_5 = myGPIO.GPIO_LO
86     myGPIO.out_status_6 = myGPIO.GPIO_LO
87     myGPIO.out_status_7 = myGPIO.GPIO_LO
88     myGPIO.setGPIO()
89     print("set lo")
90     time.sleep(1)
91
92
93 if __name__ == '__main__':
94     try:
95         runExample()
96     except (KeyboardInterrupt, SystemExit) as exErr:
97         print("\nEnding Example 1")
98         sys.exit(0)

```

7.3 Example 2

Listing 2: examples/qwiic_gpio_ex2.py

```

1  #!/usr/bin/env python
2  # -----
3  # qwiic_gpio_ex2.py
4  #
5
6  # Simple Example for the Qwiic GPIO Device, reads every GPIO
7  # -----
8  #
9  # Written by SparkFun Electronics, May 2019
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

```

(continues on next page)

(continued from previous page)

```

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 1
41 #
42
43 from __future__ import print_function
44 import qwiic_gpio
45 import time
46 import sys
47
48
49 def runExample():
50
51     print("\nSparkFun Qwiic GPIO Example 2\n")
52     myGPIO = qwiic_gpio.QwiicGPIO()
53
54     if myGPIO.isConnected() == False:
55         print("The Qwiic GPIO isn't connected to the system. Please check your
↳connection",
56               file=sys.stderr)
57         return
58
59     myGPIO.begin()
60     myGPIO.mode_0 = myGPIO.GPIO_IN
61     myGPIO.mode_1 = myGPIO.GPIO_IN
62     myGPIO.mode_2 = myGPIO.GPIO_IN
63     myGPIO.mode_3 = myGPIO.GPIO_IN
64     myGPIO.mode_4 = myGPIO.GPIO_IN
65     myGPIO.mode_5 = myGPIO.GPIO_IN
66     myGPIO.mode_6 = myGPIO.GPIO_IN
67     myGPIO.mode_7 = myGPIO.GPIO_IN
68     myGPIO.setMode()
69
70     while True:
71         myGPIO.getGPIO() #This function updates each in_status_x variable
72         print("GPIO 0:", end=" ")
73         print(myGPIO.in_status_0, end=" ")
74         print("GPIO 1:", end=" ")
75         print(myGPIO.in_status_1, end=" ")
76         print("GPIO 2:", end=" ")
77         print(myGPIO.in_status_2, end=" ")
78         print("GPIO 3:", end=" ")

```

(continues on next page)

(continued from previous page)

```

79     print(myGPIO.in_status_3, end=" ")
80     print("GPIO 4:", end=" ")
81     print(myGPIO.in_status_4, end=" ")
82     print("GPIO 5:", end=" ")
83     print(myGPIO.in_status_5, end=" ")
84     print("GPIO 6:", end=" ")
85     print(myGPIO.in_status_6, end=" ")
86     print("GPIO 7:", end=" ")
87     print(myGPIO.in_status_7)
88     time.sleep(.25)
89
90 if __name__ == '__main__':
91     try:
92         runExample()
93     except (KeyboardInterrupt, SystemExit) as exErr:
94         print("\nEnding Example 1")
95         sys.exit(0)

```

7.4 Example 3

Listing 3: examples/qwiic_gpio_ex3.py

```

1  #!/usr/bin/env python
2  # -----
3  # qwiic_gpio_ex3.py
4  #
5
6  # Simple Example for the Qwiic GPIO Device, reads every GPIO
7  # -----
8  #
9  # Written by SparkFun Electronics, May 2019
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

```

(continues on next page)

(continued from previous page)

```

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 1
41 #
42
43 from __future__ import print_function
44 import qwiic_gpio
45 import time
46 import sys
47
48
49 def runExample():
50
51     print("\nSparkFun Qwiic GPIO Example 3\n")
52     myGPIO = qwiic_gpio.QwiicGPIO()
53
54     if myGPIO.isConnected() == False:
55         print("The Qwiic GPIO isn't connected to the system. Please check your_
↳connection",
56             file=sys.stderr)
57         return
58
59     myGPIO.begin()
60     myGPIO.mode_0 = myGPIO.GPIO_IN
61     myGPIO.mode_1 = myGPIO.GPIO_IN
62     myGPIO.mode_2 = myGPIO.GPIO_IN
63     myGPIO.mode_3 = myGPIO.GPIO_IN
64     myGPIO.mode_4 = myGPIO.GPIO_IN
65     myGPIO.mode_5 = myGPIO.GPIO_IN
66     myGPIO.mode_6 = myGPIO.GPIO_IN
67     myGPIO.mode_7 = myGPIO.GPIO_IN
68     myGPIO.setMode()
69
70     myGPIO.inversion_0 = myGPIO.INVERT
71     myGPIO.inversion_1 = myGPIO.NO_INVERT
72     myGPIO.inversion_2 = myGPIO.INVERT
73     myGPIO.inversion_3 = myGPIO.NO_INVERT
74     myGPIO.inversion_4 = myGPIO.INVERT
75     myGPIO.inversion_5 = myGPIO.NO_INVERT
76     myGPIO.inversion_6 = myGPIO.INVERT
77     myGPIO.inversion_7 = myGPIO.NO_INVERT
78     myGPIO.setInversion()
79
80     while True:
81         myGPIO.getGPIO() # This function updates each in_status_x variable
82         print("GPIO 0:", end=" ")
83         print(myGPIO.in_status_0, end=" ")
84         print("GPIO 1:", end=" ")
85         print(myGPIO.in_status_1, end=" ")
86         print("GPIO 2:", end=" ")
87         print(myGPIO.in_status_2, end=" ")
88         print("GPIO 3:", end=" ")

```

(continues on next page)

(continued from previous page)

```

89     print(myGPIO.in_status_3, end=" ")
90     print("GPIO 4:", end=" ")
91     print(myGPIO.in_status_4, end=" ")
92     print("GPIO 5:", end=" ")
93     print(myGPIO.in_status_5, end=" ")
94     print("GPIO 6:", end=" ")
95     print(myGPIO.in_status_6, end=" ")
96     print("GPIO 7:", end=" ")
97     print(myGPIO.in_status_7)
98     time.sleep(.25)
99
100 if __name__ == '__main__':
101     try:
102         runExample()
103     except (KeyboardInterrupt, SystemExit) as exErr:
104         print("\nEnding Example 1")
105         sys.exit(0)

```

7.5 Example 4

Listing 4: examples/qwiic_gpio_ex4.py

```

1  #!/usr/bin/env python
2  # -----
3  # qwiic_gpio_ex3.py
4  #
5
6  # Simple Example for the Qwiic GPIO Device, reads every GPIO
7  # -----
8  #
9  # Written by SparkFun Electronics, May 2019
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

```

(continues on next page)

(continued from previous page)

```

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 1
41 #
42
43 from __future__ import print_function
44 import qwiic_gpio
45 import time
46 import sys
47
48
49 def runExample():
50
51     print("\nSparkFun Qwiic GPIO Example 3\n")
52     myGPIO = qwiic_gpio.QwiicGPIO()
53
54     if myGPIO.isConnected() == False:
55         print("The Qwiic GPIO isn't connected to the system. Please check your_
↳connection",
56             file=sys.stderr)
57         return
58
59     myGPIO.begin()
60     myGPIO.mode_0 = myGPIO.GPIO_IN
61     myGPIO.mode_1 = myGPIO.GPIO_IN
62     myGPIO.mode_2 = myGPIO.GPIO_IN
63     myGPIO.mode_3 = myGPIO.GPIO_IN
64     myGPIO.mode_4 = myGPIO.GPIO_IN
65     myGPIO.mode_5 = myGPIO.GPIO_IN
66     myGPIO.mode_6 = myGPIO.GPIO_IN
67     myGPIO.mode_7 = myGPIO.GPIO_IN
68     myGPIO.setMode()
69
70     myGPIO.inversion_0 = myGPIO.INVERT
71     myGPIO.inversion_1 = myGPIO.NO_INVERT
72     myGPIO.inversion_2 = myGPIO.INVERT
73     myGPIO.inversion_3 = myGPIO.NO_INVERT
74     myGPIO.inversion_4 = myGPIO.INVERT
75     myGPIO.inversion_5 = myGPIO.NO_INVERT
76     myGPIO.inversion_6 = myGPIO.INVERT
77     myGPIO.inversion_7 = myGPIO.NO_INVERT
78     myGPIO.setInversion()
79
80     while True:
81         myGPIO.getGPIO() # This function updates each in_status_x variable
82         print("GPIO 0:", end=" ")
83         print(myGPIO.in_status_0 == myGPIO.GPIO_LO, end=" ")
84         print("GPIO 1:", end=" ")
85         print(myGPIO.in_status_1 == myGPIO.GPIO_LO, end=" ")
86         print("GPIO 2:", end=" ")
87         print(myGPIO.in_status_2 == myGPIO.GPIO_LO, end=" ")
88         print("GPIO 3:", end=" ")

```

(continues on next page)

(continued from previous page)

```
89     print(myGPIO.in_status_3 == myGPIO.GPIO_LO, end=" ")
90     print("GPIO 4:", end=" ")
91     print(myGPIO.in_status_4 == myGPIO.GPIO_LO, end=" ")
92     print("GPIO 5:", end=" ")
93     print(myGPIO.in_status_5 == myGPIO.GPIO_LO, end=" ")
94     print("GPIO 6:", end=" ")
95     print(myGPIO.in_status_6 == myGPIO.GPIO_LO, end=" ")
96     print("GPIO 7:", end=" ")
97     print(myGPIO.in_status_7 == myGPIO.GPIO_LO)
98     time.sleep(.25)
99
100 if __name__ == '__main__':
101     try:
102         runExample()
103     except (KeyboardInterrupt, SystemExit) as exErr:
104         print("\nEnding Example 1")
105         sys.exit(0)
```

CHAPTER 8

Indices and tables

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

q

`qwiic_gpio`, 15

B

`begin()` (*qwiic_gpio.QwiicGPIO method*), 15

G

`getGPIO()` (*qwiic_gpio.QwiicGPIO method*), 15

`getInversion()` (*qwiic_gpio.QwiicGPIO method*),
16

`getMode()` (*qwiic_gpio.QwiicGPIO method*), 16

I

`isConnected()` (*qwiic_gpio.QwiicGPIO method*), 16

Q

`qwiic_gpio` (*module*), 15

`QwiicGPIO` (*class in qwiic_gpio*), 15

S

`setGPIO()` (*qwiic_gpio.QwiicGPIO method*), 16

`setInversion()` (*qwiic_gpio.QwiicGPIO method*),
16

`setMode()` (*qwiic_gpio.QwiicGPIO method*), 16