sparkfun$_{gpiopy}$

Release 0.0.2

Mar 05, 2021

Contents:

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

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

CHAPTER 5

Installation

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)
```

CHAPTER 7

Table of Contents

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 of the initializtion 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

```
#!/usr/bin/env python
#-----------------------------------------------------------------------------
# qwiic_gpio_ex1.py
#
# Simple Example for the Qwiic GPIO Device, toggles GPIO 0 on and off.
#-----------------------------------------------------------------------------
#
# Written by  SparkFun Electronics, May 2019
#
# 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) 2019 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 1
#

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
```

(continues on next page)

(continued from previous page)

```
        myGPIO.out_status_3 = myGPIO.GPIO_HI
        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)
```

7.3 Example 2

Listing 2: examples/qwiic_gpio_ex2.py

```
#!/usr/bin/env python
# ------------------------------------------------------------------------------
# qwiic_gpio_ex2.py
#

# Simple Example for the Qwiic GPIO Device, reads every GPIO
# ------------------------------------------------------------------------
#
# Written by  SparkFun Electronics, May 2019
#
# 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) 2019 SparkFun Electronics
#
# Permission is hereby granted, free of charge, to any person obtaining a copy
```

(continues on next page)

(continued from previous page)

```
# 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 1
#

from __future__ import print_function
import qwiic_gpio
import time
import sys


def runExample():

    print("\nSparkFun Qwiic GPIO Example 2\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_IN
    myGPIO.mode_1 = myGPIO.GPIO_IN
    myGPIO.mode_2 = myGPIO.GPIO_IN
    myGPIO.mode_3 = myGPIO.GPIO_IN
    myGPIO.mode_4 = myGPIO.GPIO_IN
    myGPIO.mode_5 = myGPIO.GPIO_IN
    myGPIO.mode_6 = myGPIO.GPIO_IN
    myGPIO.mode_7 = myGPIO.GPIO_IN
    myGPIO.setMode()

    while True:
        myGPIO.getGPIO() #This function updates each in_status_x variable
        print("GPIO 0:", end=" ")
        print(myGPIO.in_status_0, end=" ")
        print("GPIO 1:", end=" ")
        print(myGPIO.in_status_1, end=" ")
        print("GPIO 2:", end=" ")
        print(myGPIO.in_status_2, end=" ")
        print("GPIO 3:", end=" ")
```

(continues on next page)

(continued from previous page)

```
        print(myGPIO.in_status_3, end=" ")
        print("GPIO 4:", end=" ")
        print(myGPIO.in_status_4, end=" ")
        print("GPIO 5:", end=" ")
        print(myGPIO.in_status_5, end=" ")
        print("GPIO 6:", end=" ")
        print(myGPIO.in_status_6, end=" ")
        print("GPIO 7:", end=" ")
        print(myGPIO.in_status_7)
        time.sleep(.25)

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

7.4 Example 3

Listing 3: examples/qwiic_gpio_ex3.py

```
#!/usr/bin/env python
# ------------------------------------------------------------------------------
# qwiic_gpio_ex3.py
#

# Simple Example for the Qwiic GPIO Device, reads every GPIO
# ------------------------------------------------------------------------
#
# Written by  SparkFun Electronics, May 2019
#
# 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) 2019 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
```

(continues on next page)

(continued from previous page)

```
# 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 1
#

from __future__ import print_function
import qwiic_gpio
import time
import sys


def runExample():

    print("\nSparkFun Qwiic GPIO Example 3\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_IN
    myGPIO.mode_1 = myGPIO.GPIO_IN
    myGPIO.mode_2 = myGPIO.GPIO_IN
    myGPIO.mode_3 = myGPIO.GPIO_IN
    myGPIO.mode_4 = myGPIO.GPIO_IN
    myGPIO.mode_5 = myGPIO.GPIO_IN
    myGPIO.mode_6 = myGPIO.GPIO_IN
    myGPIO.mode_7 = myGPIO.GPIO_IN
    myGPIO.setMode()

    myGPIO.inversion_0 = myGPIO.INVERT
    myGPIO.inversion_1 = myGPIO.NO_INVERT
    myGPIO.inversion_2 = myGPIO.INVERT
    myGPIO.inversion_3 = myGPIO.NO_INVERT
    myGPIO.inversion_4 = myGPIO.INVERT
    myGPIO.inversion_5 = myGPIO.NO_INVERT
    myGPIO.inversion_6 = myGPIO.INVERT
    myGPIO.inversion_7 = myGPIO.NO_INVERT
    myGPIO.setInversion()

    while True:
        myGPIO.getGPIO()  # This function updates each in_status_x variable
        print("GPIO 0:", end=" ")
        print(myGPIO.in_status_0, end=" ")
        print("GPIO 1:", end=" ")
        print(myGPIO.in_status_1, end=" ")
        print("GPIO 2:", end=" ")
        print(myGPIO.in_status_2, end=" ")
        print("GPIO 3:", end=" ")
```

(continues on next page)

(continued from previous page)

```
        print(myGPIO.in_status_3, end=" ")
        print("GPIO 4:", end=" ")
        print(myGPIO.in_status_4, end=" ")
        print("GPIO 5:", end=" ")
        print(myGPIO.in_status_5, end=" ")
        print("GPIO 6:", end=" ")
        print(myGPIO.in_status_6, end=" ")
        print("GPIO 7:", end=" ")
        print(myGPIO.in_status_7)
        time.sleep(.25)

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

7.5 Example 4

Listing 4: examples/qwiic_gpio_ex4.py

```
#!/usr/bin/env python
# ------------------------------------------------------------------------------
# qwiic_gpio_ex3.py
#

# Simple Example for the Qwiic GPIO Device, reads every GPIO
# ------------------------------------------------------------------------
#
# Written by  SparkFun Electronics, May 2019
#
# 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) 2019 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
```

(continues on next page)

(continued from previous page)

```
# 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 1
#

from __future__ import print_function
import qwiic_gpio
import time
import sys


def runExample():

    print("\nSparkFun Qwiic GPIO Example 3\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_IN
    myGPIO.mode_1 = myGPIO.GPIO_IN
    myGPIO.mode_2 = myGPIO.GPIO_IN
    myGPIO.mode_3 = myGPIO.GPIO_IN
    myGPIO.mode_4 = myGPIO.GPIO_IN
    myGPIO.mode_5 = myGPIO.GPIO_IN
    myGPIO.mode_6 = myGPIO.GPIO_IN
    myGPIO.mode_7 = myGPIO.GPIO_IN
    myGPIO.setMode()

    myGPIO.inversion_0 = myGPIO.INVERT
    myGPIO.inversion_1 = myGPIO.NO_INVERT
    myGPIO.inversion_2 = myGPIO.INVERT
    myGPIO.inversion_3 = myGPIO.NO_INVERT
    myGPIO.inversion_4 = myGPIO.INVERT
    myGPIO.inversion_5 = myGPIO.NO_INVERT
    myGPIO.inversion_6 = myGPIO.INVERT
    myGPIO.inversion_7 = myGPIO.NO_INVERT
    myGPIO.setInversion()

    while True:
        myGPIO.getGPIO()  # This function updates each in_status_x variable
        print("GPIO 0:", end=" ")
        print(myGPIO.in_status_0 == myGPIO.GPIO_LO, end=" ")
        print("GPIO 1:", end=" ")
        print(myGPIO.in_status_1 == myGPIO.GPIO_LO, end=" ")
        print("GPIO 2:", end=" ")
        print(myGPIO.in_status_2 == myGPIO.GPIO_LO, end=" ")
        print("GPIO 3:", end=" ")
```

(continues on next page)

(continued from previous page)

```
        print(myGPIO.in_status_3 == myGPIO.GPIO_LO, end=" ")
        print("GPIO 4:", end=" ")
        print(myGPIO.in_status_4 == myGPIO.GPIO_LO, end=" ")
        print("GPIO 5:", end=" ")
        print(myGPIO.in_status_5 == myGPIO.GPIO_LO, end=" ")
        print("GPIO 6:", end=" ")
        print(myGPIO.in_status_6 == myGPIO.GPIO_LO, end=" ")
        print("GPIO 7:", end=" ")
        print(myGPIO.in_status_7 == myGPIO.GPIO_LO)
        time.sleep(.25)

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

CHAPTER 8

Indices and tables

- genindex
- modindex
- search

Python Module Index

q

Index