sparkfun$_{t}op_{p}hat_{b}utton_{p}y$

Release 0.0.2

Apr 13, 2020

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 top_phat_button 15
- 7.2 Example 1 17
- 7.3 Example 2 19

8 Indices and tables 23

Python Module Index 25

Index 27

Python module for the buttons aboard the SparkFun Top pHAT

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 Top pHAT Button 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 Top pHAT Button module documentation is hosted at ReadTheDocs

CHAPTER 5

Installation

5.1 PyPi Installation

This repository is hosted on PyPi as the sparkfun-top-phat-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-top-phat-button
```

For the current user:

```
pip install sparkfun-top-phat-button
```

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

CHAPTER 6

Example Use

See the examples directory for more detailed use examples.

```python
from __future__ import print_function
import top_phat_button
import time
import sys

myButtons = top_phat_button.ToppHATButton()

def runExample():

    print("\nSparkFun Top pHAT Button  Example 1\n")

    if myButtons.is_connected() == False:
        print("The Top pHAT Button device isn't connected to the system. Please check your connection", \
            file=sys.stderr)
        return

    myButtons.pressed_interrupt_enable = False
    myButtons.clicked_interrupt_enable = False

    while True:
        myButtons.button_pressed #These functions must be called to update button variables to their latest setting
        myButtons.button_clicked #These functions must be called to update button variables to their latest setting
        if myButtons.a_pressed == True:
            print("A Pressed")
        if myButtons.a_clicked == True:
            print("A Released")
        if myButtons.b_pressed == True:
            print("B Pressed")
        if myButtons.b_clicked == True:
```

(continues on next page)

(continued from previous page)

```
            print("B Released")
        if myButtons.up_pressed == True:
            print("Up Pressed")
        if myButtons.up_clicked == True:
            print("Up Released")
        if myButtons.down_pressed == True:
            print("Down Pressed")
        if myButtons.down_clicked == True:
            print("Down Released")
        if myButtons.left_pressed == True:
            print("Left Pressed")
        if myButtons.left_clicked == True:
            print("Left Released")
        if myButtons.right_pressed == True:
            print("Right Pressed")
        if myButtons.right_clicked == True:
            print("Right Released")
        if myButtons.center_pressed == True:
            print("Center Pressed")
        if myButtons.center_clicked == True:
            print("Center Released")

        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 top_phat_button

Python module for the[SparkFun Qwiic Joystick](https://www.sparkfun.com/products/15168)

This python package is a port of the existing [SparkFun Qwiic Joystick Arduino Library](https://github.com/sparkfun/SparkFun_Qwiic_Joystick_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 `top_phat_button.`**`ToppHATButton`**(*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 ToppHATButton device object.

Return type Object

`begin`()

Initialize the operation of the button module

Returns Returns true of the initializtion was successful, otherwise False.

Return type bool

`button_clicked`

Returns 1 when a button has received a full click cycle (press and release). The interrupt must be cleared by the user. 7(MSB) 6 5 4 3 2 1 0(LSB)

INT CTR RGT LFT DWN UP B A

Returns Clicked status of all buttons in a byte

Return type integer

button_pressed

Updates and returns buffer for all buttons and whether or not they are pressed as well as the pressed interrupt flag Reading this register also clears it. 7(MSB) 6 5 4 3 2 1 0(LSB)

INT CTR RGT LFT DWN UP B A

Returns button status

Return type integer

clicked_interrupt_enable

Returns the status of the clicked interrupt enable

Returns The clicked interrupt enable bit

Return type bool

connected

Determine if the Top pHAT Buttons are connected to the system..

Returns True if the device is connected, otherwise False.

Return type bool

get_button_clicked()

Returns 1 when a button has received a full click cycle (press and release). The interrupt must be cleared by the user. 7(MSB) 6 5 4 3 2 1 0(LSB)

INT CTR RGT LFT DWN UP B A

Returns Clicked status of all buttons in a byte

Return type integer

get_button_pressed()

Updates and returns buffer for all buttons and whether or not they are pressed as well as the pressed interrupt flag Reading this register also clears it. 7(MSB) 6 5 4 3 2 1 0(LSB)

INT CTR RGT LFT DWN UP B A

Returns button status

Return type integer

get_clicked_interrupt()

Returns the status of the clicked interrupt enable

Returns The clicked interrupt enable bit

Return type bool

get_pressed_interrupt()

Returns the status of the pressed interrupt enable

Returns The pressed interrupt enable bit

Return type bool

get_version()

Returns a string of the firmware version number

Returns The firmware version

Return type string

is_connected()

Determine if the Top pHAT Buttons are connected to the system..

Returns True if the device is connected, otherwise False.

Return type bool

pressed_interrupt_enable

Returns the status of the pressed interrupt enable

Returns The pressed interrupt enable bit

Return type bool

set_clicked_interrupt(*bit_setting*)

Sets the status of the clicked interrupt enable bit

Param The clicked interrupt enable bit

Returns The status of the I2C transaction

Return type bool

set_pressed_interrupt(*bit_setting*)

Sets the status of the pressed interrupt enable bit

Param The pressed interrupt enable bit

Returns The status of the I2C transaction

Return type bool

version

Returns a string of the firmware version number

Returns The firmware version

Return type string

7.2 Example 1

Listing 1: examples/top_phat_button_ex1.py

```
#!/usr/bin/env python
#-----------------------------------------------------------------------------
# top_phat_button_ex1.py
#
# Polling example for the Top pHAT Buttons
#------------------------------------------------------------------------
#
# Written by  SparkFun Electronics, April 2020
#
# This python library supports the SparkFun Electroncis qwiic
# qwiic sensor/board ecosystem on a Raspberry Pi (and compatable) single
# board computers.
```

(continues on next page)

(continued from previous page)

```
#
# 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
# 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 top_phat_button
import time
import sys

myButtons = top_phat_button.ToppHATButton()

def runExample():

    print("\nSparkFun Top pHAT Button  Example 1\n")

    if myButtons.is_connected() == False:
        print("The Top pHAT Button device isn't connected to the system. Please check␣
↪your connection", \
            file=sys.stderr)
        return

    myButtons.pressed_interrupt_enable = False
    myButtons.clicked_interrupt_enable = False

    while True:
        myButtons.button_pressed #These functions must be called to update button␣
↪variables to their latest setting
        myButtons.button_clicked #These functions must be called to update button␣
↪variables to their latest setting
        if myButtons.a_pressed == True:
            print("A Pressed")
        if myButtons.a_clicked == True:
```

(continues on next page)

(continued from previous page)

```
            print("A Released")
        if myButtons.b_pressed == True:
            print("B Pressed")
        if myButtons.b_clicked == True:
            print("B Released")
        if myButtons.up_pressed == True:
            print("Up Pressed")
        if myButtons.up_clicked == True:
            print("Up Released")
        if myButtons.down_pressed == True:
            print("Down Pressed")
        if myButtons.down_clicked == True:
            print("Down Released")
        if myButtons.left_pressed == True:
            print("Left Pressed")
        if myButtons.left_clicked == True:
            print("Left Released")
        if myButtons.right_pressed == True:
            print("Right Pressed")
        if myButtons.right_clicked == True:
            print("Right Released")
        if myButtons.center_pressed == True:
            print("Center Pressed")
        if myButtons.center_clicked == True:
            print("Center Released")

        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/top_phat_button_ex2.py

```
#!/usr/bin/env python
#-----------------------------------------------------------------------------
# top_phat_button_ex2.py
#
# Interrupt example for the Top pHAT Buttons
#------------------------------------------------------------------------
#
# Written by  SparkFun Electronics, April 2020
#
# 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)

```python
#
# 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 2
#

from __future__ import print_function
import top_phat_button
import time
import sys
import RPi.GPIO as GPIO

INTERRUPT_PIN = 25
GPIO.setmode(GPIO.BCM)
GPIO.setup(INTERRUPT_PIN, GPIO.IN)

myButtons = top_phat_button.ToppHATButton()

def interruptCallback(channel):
    myButtons.button_pressed
    myButtons.button_clicked #Both interrupts are configured, so we need to read both registers to clear the interrupt and update our button data.
    if myButtons.a_pressed == True:
        print("A Pressed")
    if myButtons.a_clicked == True:
        print("A Released")
    if myButtons.b_pressed == True:
        print("B Pressed")
    if myButtons.b_clicked == True:
        print("B Released")
    if myButtons.up_pressed == True:
        print("Up Pressed")
    if myButtons.up_clicked == True:
        print("Up Released")
    if myButtons.down_pressed == True:
        print("Down Pressed")
```

(continues on next page)

(continued from previous page)

```
    if myButtons.down_clicked == True:
        print("Down Released")
    if myButtons.left_pressed == True:
        print("Left Pressed")
    if myButtons.left_clicked == True:
        print("Left Released")
    if myButtons.right_pressed == True:
        print("Right Pressed")
    if myButtons.right_clicked == True:
        print("Right Released")
    if myButtons.center_pressed == True:
        print("Center Pressed")
    if myButtons.center_clicked == True:
        print("Center Released")

GPIO.add_event_detect(INTERRUPT_PIN, GPIO.FALLING, callback=interruptCallback, bouncetime=5)

def runExample():

    print("\nSparkFun Top pHAT Button  Example 1\n")

    if myButtons.is_connected() == False:
        print("The Top pHAT Button device isn't connected to the system. Please check your connection", \
            file=sys.stderr)
        return

    myButtons.pressed_interrupt_enable = True
    myButtons.clicked_interrupt_enable = True #Enable both hardware interrupts

    while True:

        time.sleep(.1)


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

t

top_phat_button, 15

Index

B

begin() (*top_phat_button.ToppHATButton method*), 15

button_clicked (*top_phat_button.ToppHATButton attribute*), 15

button_pressed (*top_phat_button.ToppHATButton attribute*), 16

C

clicked_interrupt_enable (*top_phat_button.ToppHATButton attribute*), 16

connected (*top_phat_button.ToppHATButton attribute*), 16

G

get_button_clicked() (*top_phat_button.ToppHATButton method*), 16

get_button_pressed() (*top_phat_button.ToppHATButton method*), 16

get_clicked_interrupt() (*top_phat_button.ToppHATButton method*), 16

get_pressed_interrupt() (*top_phat_button.ToppHATButton method*), 16

get_version() (*top_phat_button.ToppHATButton method*), 16

I

is_connected() (*top_phat_button.ToppHATButton method*), 17

P

pressed_interrupt_enable (*top_phat_button.ToppHATButton attribute*), 17

S

set_clicked_interrupt() (*top_phat_button.ToppHATButton method*), 17

set_pressed_interrupt() (*top_phat_button.ToppHATButton method*), 17

T

top_phat_button (*module*), 15

ToppHATButton (*class in top_phat_button*), 15

V

version (*top_phat_button.ToppHATButton attribute*), 17