sparkfun$_{q}wiic_{k}eypad$

Release 0.0.9

Mar 27, 2020

Contents:

1 Contents **3**

2 Dependencies **5**

3 Documentation **7**

4 Installation **9**
- 4.1 PyPi Installation . . . 9
- 4.2 Local Installation . . . 9

5 Example Use **11**

6 Table of Contents **13**
- 6.1 API Reference . . . 13
  - 6.1.1 qwiic_keypad . . . 13
- 6.2 Read a Key . . . 14
- 6.3 Read a Key - Report Delay Time . . . 16
- 6.4 Validate a Key Code . . . 18

7 Indices and tables **23**

Python Module Index **25**

Index **27**

Python module for the qwiic keypad, which is part of the SparkFun Qwiic Keypad - 12 Button

This python package is a port of the existing SparkFun Qwiic Keypad Arduino Library

This package can be used in conjunction with the overall SparkFun qwiic Python Package

New to qwiic? Take a look at the entire SparkFun qwiic ecosystem.

CHAPTER 1

Contents

- *Dependencies*
- *Installation*
- *Documentation*
- *Example Use*

CHAPTER 2

Dependencies

This driver package depends on the qwiic I2C driver: Qwiic_I2C_Py

CHAPTER 3

Documentation

The SparkFun qwiic Keypad module documentation is hosted at ReadTheDocs

CHAPTER 4

Installation

4.1 PyPi Installation

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

For the current user:

```
pip install sparkfun-qwiic-keypad
```

4.2 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_keypad-<version>.tar.gz
```

CHAPTER 5

Example Use

See the examples directory for more detailed use examples.

```
import qwiic_keypad
import time
import sys

def runExample():

    print("\nSparkFun qwiic Keypad   Example 1\n")
    myKeypad = qwiic_keypad.QwiicKeypad()

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

    myKeypad.begin()

    button = 0
    while True:

        # necessary for keypad to pull button from stack to readable register
        myKeypad.update_fifo()
        button = myKeypad.get_button()

        if button == -1:
            print("No keypad detected")
            time.sleep(1)

        elif button != 0:

            # Get the character version of this char
            charButton = chr(button)
```

(continues on next page)

(continued from previous page)

```
            if charButton == '#':
                print()
            elif charButton == '*':
                print(" ", end="")
            else:
                print(charButton, end="")

            # Flush the stdout buffer to give immediate user feedback
            sys.stdout.flush()

        time.sleep(.25)
        # Development in progress
```

CHAPTER 6

Table of Contents

6.1 API Reference

6.1.1 qwiic_keypad

Python module for the[SparkFun Qwiic Keypad - 12 Button Breakout](https://www.sparkfun.com/products/15290)

This python package is a port of the existing [SparkFun Qwiic Keypad Arduino Library](https://github.com/sparkfun/SparkFun_Qwiic_Keypad_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_keypad.**QwiicKeypad**(*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 QwiicKeypad device object.

Return type Object

begin()

Initialize the operation of the Keypad module

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

Return type bool

connected

Determine if a Keypad device is conntected to the system..

Returns True if the device is connected, otherwise False.

Return type bool

get_button()
Returns the button at the top of the stack (aka the oldest button).

The return value is the 'ascii' value of th key pressed. To convert to a character, use the python char() function.

Returns The next button value

Return type byte as integer

get_version()
Returns a string of the firmware version number

Returns The firmware version

Return type string

is_connected()
Determine if a Keypad device is conntected to the system..

Returns True if the device is connected, otherwise False.

Return type bool

time_since_pressed()
Returns the number of milliseconds since the current button in FIFO was pressed.

Returns The elapsed time since button was pressed

Return type integer

update_fifo()
"commands" keypad to plug in the next button into the registerMap note, this actually sets the bit0 on the updateFIFO register

Returns No return value

version
Returns a string of the firmware version number

Returns The firmware version

Return type string

6.2 Read a Key

Listing 1: examples/qwiic_keypad_ex1.py

```
#!/usr/bin/env python
#-----------------------------------------------------------------------------
# qwiic_env_keypad_ex1.py
#
# Simple Example for the Qwiic Keypad Device
#-----------------------------------------------------------------------------
#
# 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
```

(continues on next page)

(continued from previous page)

```
# 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
# 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_keypad
import time
import sys

def runExample():

        print("\nSparkFun qwiic Keypad   Example 1\n")
        myKeypad = qwiic_keypad.QwiicKeypad()

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

        myKeypad.begin()

        print("Initialized. Firmware Version: %s" % myKeypad.version)
        print("Press a button: * to do a space. # to go to next line.")

        button = 0
        while True:

                # necessary for keypad to pull button from stack to readable register
                myKeypad.update_fifo()
                button = myKeypad.get_button()
```

(continues on next page)

(continued from previous page)

```
                if button == -1:
                        print("No keypad detected")
                        time.sleep(1)

                elif button != 0:

                        # Get the character version of this char
                        charButton = chr(button)
                        if charButton == '#':
                                print()
                        elif charButton == '*':
                                print(" ", end="")
                        else:
                                print(charButton, end="")

                        # Flush the stdout buffer to give immediate user feedback
                        sys.stdout.flush()

                time.sleep(.25)

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


```

6.3 Read a Key - Report Delay Time

Listing 2: examples/qwiic_keypad_ex2.py

```
#!/usr/bin/env python
#-----------------------------------------------------------------------------
# qwiic_env_keypad_ex2.py
#
# Simple Example for the Qwiic Keypad Device including Time
#-----------------------------------------------------------------------------
#
# 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
#
```

(continues on next page)

(continued from previous page)

```
# 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 qwiic_keypad
import time
import sys

def runExample():

        print("\nSparkFun qwiic Keypad   Example 2\n")
        myKeypad = qwiic_keypad.QwiicKeypad()

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

        myKeypad.begin()

        print("Initialized. Firmware Version: %s" % myKeypad.version)


        button = 0
        while True:

                # necessary for keypad to pull button from stack to readable register
                myKeypad.update_fifo()
                button = myKeypad.get_button()
                deltaT = myKeypad.time_since_pressed()

                if button == -1:
                        print("No keypad detected")
                        time.sleep(1)
                        continue

                elif button != 0:
                        print("Button %s was pressed, %d milliseconds ago." %(chr(button), deltaT))
```

(continues on next page)

(continued from previous page)

```
                time.sleep(.25)


if __name__ == '__main__':
        try:
                runExample()
        except (KeyboardInterrupt, SystemExit) as exErr:
                print("\nEnding Example 2")
                sys.exit(0)


```

6.4 Validate a Key Code

Listing 3: examples/qwiic_keypad_ex3.py

```
#!/usr/bin/env python
#-----------------------------------------------------------------------------
# qwiic_env_keypad_ex3.py
#
# Simple Example for the Qwiic Keypad Device
#-----------------------------------------------------------------------------
#
# 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
# 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.
```

(continues on next page)

(continued from previous page)

```
#===============================================================================
# Example 3
#

from __future__ import print_function
import qwiic_keypad
import time
import sys


# the correct keyCode - change to your own unique set of keys if you like.
keyCode = ['1', '2', '3', '4']

def runExample():
        # used to store the presses coming in from user. init with blanks
        userEntry = [' ']*len(keyCode)
        # used to know when a user is active and therefore we want to engage timeout stuff
        userIsActive = False

        TIMEOUT=30  # in 100s of millisecs

        timeOutCounter = 0  # variable this is incremented to keep track of timeouts.
        userEntryIndex = 0 # used to keep track of where we are in the userEnt

        print("\nSparkFun qwiic Keypad   Example 3\n")
        myKeypad = qwiic_keypad.QwiicKeypad()

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

        myKeypad.begin()

        print("Initialized. Firmware Version: %s" % myKeypad.version)
        print("Please type in the correct 4 digit KeyCode:")


        button = 0
        while True:

                # necessary for keypad to pull button from stack to readable register
                myKeypad.update_fifo()
                button = myKeypad.get_button()

                if button == -1:
                        print("No keypad detected")
                        time.sleep(1)

                elif button > 100:
                        # At startup a series 127's come accoss -- way out of range
                        # just noise
                        # Skip
                        pass
                elif button != 0:
```

(continues on next page)

(continued from previous page)

```
                        # store button into next spot in array, note, index is␣
↪incremented later
                        userEntry[userEntryIndex] = chr(button)

                        printEntry(userEntry)
                        userIsActive = True  # used to only timeout when user is␣
↪active

                        if checkEntry(userEntry):
                                print("\n\nKeycode correct. Wahooooooooooo!")
                                userIsActive = False # don't display timeout stuff.
                                time.sleep(1)

                        userEntryIndex += 1
                        if userEntryIndex == len(keyCode):
                                userEntryIndex = 0 # reset

                                printEntry(userEntry)
                                time.sleep(.3)
                                clearEntry(userEntry)
                                printEntry(userEntry)
                        timeOutCounter = 0 #reset with any new presses.

                time.sleep(.2)
                timeOutCounter += 1

        # this means the user is actively inputing
        if timeOutCounter == TIMEOUT  and userIsActive == True:
                print("\n\nTimed out... try again.")
                timeOutCounter = 0
                userEntryIndex = 0
                clearEntry(userEntry)
                userIsActive = False # so we don't continuously timeout while␣
↪inactive.

#  check user entry against keyCode array.
#  if they all match up, then respond with true.
def checkEntry(userEntry):

        for i in range(len(keyCode)):
                if userEntry[i] != keyCode[i]:
                        return False

        return True

#  "clear" entry with all spaces
def clearEntry(userEntry):

        userEntry[:] = [' ']*len(userEntry)

def printEntry(userEntry):

        print("\rUserEntry:%s" % (''.join(userEntry)), end="")
        sys.stdout.flush()


if __name__ == '__main__':
```

(continues on next page)

(continued from previous page)

```
147         try:
148                 runExample()
149         except (KeyboardInterrupt, SystemExit) as exErr:
150                 print("\nEnding Example 1")
151                 sys.exit(0)
152
153
```

CHAPTER 7

Indices and tables

- genindex
- modindex
- search

Python Module Index

q

qwiic_keypad, 13

Index

B

begin() (*qwiic_keypad.QwiicKeypad method*), 13

C

connected (*qwiic_keypad.QwiicKeypad attribute*), 13

G

get_button() (*qwiic_keypad.QwiicKeypad method*), 14
get_version() (*qwiic_keypad.QwiicKeypad method*), 14

I

is_connected() (*qwiic_keypad.QwiicKeypad method*), 14

Q

qwiic_keypad (*module*), 13
QwiicKeypad (*class in qwiic_keypad*), 13

T

time_since_pressed() (*qwiic_keypad.QwiicKeypad method*), 14

U

update_fifo() (*qwiic_keypad.QwiicKeypad method*), 14

V

version (*qwiic_keypad.QwiicKeypad attribute*), 14