
sparkfun*piservo*hat
Release 0.0.1

Mar 20, 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 (to be edited)	13
7	Table of Contents	15
7.1	API Reference	15
7.1.1	pi_servo_hat	15
7.2	Example 1: Full Sweep for 90 Degree Servo	16
7.3	Example 2: Full Sweep for 180 Degree Servo	18
7.4	Example 3: Get Servo Position for 180 Degree Servo	20
7.5	Example 4: Change PWM Frequency for 180 Degree Servo	22
	Python Module Index	27
	Index	29

Python module for the [SparkFun Servo pHAT for Raspberry Pi](#) and [SparkFun Pi Servo HAT](#)

This package should 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 PiServoHat Python package current supports the following platforms:

- Raspberry Pi <!-- Platforms to be tested
- NVidia Jetson Nano
- Google Coral Development Board ->

CHAPTER 3

Dependencies

This package depends on the qwiic I2C driver: `Qwiic_I2C_Py`

This package depends on the PCA9685 PWM controller: `Qwiic_PCA9685_Py`

CHAPTER 4

Documentation

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

5.1 PyPi Installation

This repository is hosted on PyPi as the `sparkfun-pi-servo-hat` 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-pi-servo-hat
```

For the current user:

```
pip install sparkfun-pi-servo-hat
```

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_pi-servo-hat-<version>.tar.gz
```

Example Use (to be edited)

See the examples directory for more detailed use examples.

```
import pi_servo_hat
import time
import sys

def runExample():

    print("\nSparkFun Pi Servo Hat Demo\n")
    mySensor = pi_servo_hat.PiServoHat()

    if mySensor.isConnected() == False:
        print("The Qwiic PCA9685 device isn't connected to the system. Please check_
↪your connection", \
            file=sys.stderr)
        return

    mySensor.restart()

    # Test Run
    #####
    # Moves servo position to 0 degrees (1ms), Channel 0
    mySensor.move_servo_position(0, 0)

    # Pause 1 sec
    time.sleep(1)

    # Moves servo position to 90 degrees (2ms), Channel 0
    mySensor.move_servo_position(0, 90)
```


7.1 API Reference

7.1.1 pi_servo_hat

Python module for the [SparkFun Pi Servo HAT](<https://www.sparkfun.com/products/14328>) and [SparkFun Servo pHAT for Raspberry Pi](<https://www.sparkfun.com/products/15316>). This package should be used in conjunction with the `sparkfun_pca9685` package.

class `pi_servo_hat.PiServoHat` (*address=None, debug=None*)

SparkFun PiServoHat Initialise the `qwiic_pca9685` python module at `address` with `i2c_driver`.

param address The I2C address to use for the device. If not provided, the default address is used.

param i2c_driver An existing `i2c` driver object. If not provided a driver object is created.

return Constructor Initialization True- Successful False- Issue loading I2C driver

rtype Bool

get_pwm_frequency ()

Reads the PWM frequency used on outputs. 50 Hz is recommended for most servos.

Returns PWM Frequency Range: 24 Hz to 1526 Hz

Return type Integer

get_servo_position (*channel, swing=None*)

Reads the specified location for the servo in degrees.

Parameters

- **channel** – Channel of Servo to Control Range: 0 to 15
- **swing** – Range of Servo Movement 90- 90 Degree Servo 180- 180 Degree Servo

Returns Estimated Position (Degrees)

Return type Float**move_servo_position** (*channel, position, swing=None*)

Moves servo to specified location in degrees.

Parameters

- **channel** – Channel of Servo to Control Range: 0 to 15
- **position** – Position (Degrees) Range: Open, but should be between 0 and specified servo ‘swing’. The range is not regulated because most servos have extra room for play (i.e. a 90 degree servo may have a +120 degree usable swing). If ‘None’ is specified, the default setting is 90 degrees.
- **swing** – Range of Servo Movement 90- 90 Degree Servo 180- 180 Degree Servo

restart ()

Soft resets the chip and then clears the MODE1 register to restart the PWM functionality. The PWM frequency is returned to the default 50 Hz setting.

set_duty_cycle (*channel, duty_cycle*)

Moves servo to specified location based on duty-cycle.

Parameters

- **channel** – Channel of Servo to Control Range: 0 to 15
- **duty_cycle** – Duty-Cycle (Percentage) Float Range: 0 to 100 (%) Resolution: 1/4096

set_pwm_frequency (*frequency=None*)

Configures the PWM frequency used on outputs. 50 Hz is the default and recommended for most servos.

Parameters **frequency** – PWM Frequency Range: 24 Hz to 1526 Hz**Returns** Function Operation True- Successful False- Issue in Execution**Return type** Bool

NOTE: Changing PWM frequency affects timing for servo positioning. Additionally, the servo position needs to be reset for the output control (on all channels).

The output on all channels is initially turned off after the frequency change, but is re-enabled after any of the channels is reconfigured. However, the new PWM frequency will be in affect, so the timing of the outputs on the other channels will be off. (i.e. if a PWM frequency is doubled; the timing of that signal may be halved.)

7.2 Example 1: Full Sweep for 90 Degree Servo

Listing 1: examples/ex1_full_sweep_with_90_deg_servo.py

```

1 #-----
2 # Pi Servo Hat - Example 1
3 #-----
4 #
5 # Written by SparkFun Electronics, June 2019
6 # Author: Wes Furuya
7 #
8 # Compatibility:
9 #   * Original: https://www.sparkfun.com/products/14328
10 #   * v2: https://www.sparkfun.com/products/15316

```

(continues on next page)

(continued from previous page)

```

11 #
12 # Do you like this library? Help support SparkFun. Buy a board!
13 # For more information on Pi Servo Hat, check out the product page
14 # linked above.
15 #
16 # This program is distributed in the hope that it will be useful, but
17 # WITHOUT ANY WARRANTY without even the implied warranty of
18 # MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
19 # General Public License for more details.
20 #
21 # You should have received a copy of the GNU General Public License
22 # along with this program. If not, see <http://www.gnu.org/licenses/>.
23 #
24 #=====
25 # Copyright (c) 2019 SparkFun Electronics
26 #
27 # Permission is hereby granted, free of charge, to any person obtaining
28 # a copy of this software and associated documentation files (the
29 # "Software"), to deal in the Software without restriction, including
30 # without limitation the rights to use, copy, modify, merge, publish,
31 # distribute, sublicense, and/or sell copies of the Software, and to
32 # permit persons to whom the Software is furnished to do so, subject to
33 # the following conditions:
34 #
35 # The above copyright notice and this permission notice shall be
36 # included in all copies or substantial portions of the Software.
37 #
38 # THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
39 # EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
40 # MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
41 # IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
42 # CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
43 # TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
44 # SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
45 #=====
46
47 """
48 This example should be used with a 90 degree (range of rotation) servo
49 on channel 0 of the Pi Servo Hat.
50
51 The extended code (commented out), at the end of the example could be
52 used to test the full range of the servo motion. However, users should
53 be wary as they can damage their servo by giving it a position outside
54 the standard range of motion.
55 """
56
57 import pi_servo_hat
58 import time
59
60 # Initialize Constructor
61 test = pi_servo_hat.PiServoHat()
62
63 # Restart Servo Hat (in case Hat is frozen/locked)
64 test.restart()
65
66 # Test Run
67 #####

```

(continues on next page)

(continued from previous page)

```

68 # Moves servo position to 0 degrees (1ms), Channel 0
69 test.move_servo_position(0, 0)
70
71 # Pause 1 sec
72 time.sleep(1)
73
74 # Moves servo position to 90 degrees (2ms), Channel 0
75 test.move_servo_position(0, 90)
76
77 # Pause 1 sec
78 time.sleep(1)
79
80 # Sweep
81 #####
82 while True:
83     for i in range(0, 90):
84         print(i)
85         test.move_servo_position(0, i)
86         time.sleep(.001)
87     for i in range(90, 0, -1):
88         print(i)
89         test.move_servo_position(0, i)
90         time.sleep(.001)
91
92 #####
93 # Code below may damage servo, use with caution
94 # Test sweep for full range of servo (outside 0 to 90 degrees).
95 # while True:
96 #     for i in range(-23, 100):
97 #         print(i)
98 #         test.move_servo_position(0, i)
99 #         time.sleep(.001)
100 #     for i in range(100, -23, -1):
101 #         print(i)
102 #         test.move_servo_position(0, i)
103 #         time.sleep(.001)

```

7.3 Example 2: Full Sweep for 180 Degree Servo

Listing 2: examples/ex2_full_sweep_with_180_deg_servo.py

```

1 #-----
2 # Pi Servo Hat - Example 2
3 #-----
4 #
5 # Written by SparkFun Electronics, June 2019
6 # Author: Wes Furuya
7 #
8 # Compatibility:
9 #     * Original: https://www.sparkfun.com/products/14328
10 #     * v2: https://www.sparkfun.com/products/15316
11 #
12 # Do you like this library? Help support SparkFun. Buy a board!
13 # For more information on Pi Servo Hat, check out the product page

```

(continues on next page)

(continued from previous page)

```

14 # linked above.
15 #
16 # This program is distributed in the hope that it will be useful, but
17 # WITHOUT ANY WARRANTY without even the implied warranty of
18 # MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
19 # General Public License for more details.
20 #
21 # You should have received a copy of the GNU General Public License
22 # along with this program. If not, see <http://www.gnu.org/licenses/>.
23 #
24 #=====
25 # Copyright (c) 2019 SparkFun Electronics
26 #
27 # Permission is hereby granted, free of charge, to any person obtaining
28 # a copy of this software and associated documentation files (the
29 # "Software"), to deal in the Software without restriction, including
30 # without limitation the rights to use, copy, modify, merge, publish,
31 # distribute, sublicense, and/or sell copies of the Software, and to
32 # permit persons to whom the Software is furnished to do so, subject to
33 # the following conditions:
34 #
35 # The above copyright notice and this permission notice shall be
36 # included in all copies or substantial portions of the Software.
37 #
38 # THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
39 # EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
40 # MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
41 # IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
42 # CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
43 # TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
44 # SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
45 #=====
46
47 """
48 This example should be used with a 180 degree (range of rotation) servo
49 on channel 0 of the Pi Servo Hat.
50
51 The extended code (commented out), at the end of the example could be
52 used to test the full range of the servo motion. However, users should
53 be wary as they can damage their servo by giving it a position outside
54 the standard range of motion.
55 """
56
57 import pi_servo_hat
58 import time
59
60 # Initialize Constructor
61 test = pi_servo_hat.PiServoHat()
62
63 # Restart Servo Hat (in case Hat is frozen/locked)
64 test.restart()
65
66 # Test Run
67 #####
68 # Moves servo position to 0 degrees (1ms), Channel 0
69 test.move_servo_position(0, 0, 180)
70

```

(continues on next page)

(continued from previous page)

```

71 # Pause 1 sec
72 time.sleep(1)
73
74 # Moves servo position to 180 degrees (2ms), Channel 0
75 test.move_servo_position(0, 180, 180)
76
77 # Pause 1 sec
78 time.sleep(1)
79
80 # Sweep
81 #####
82 while True:
83     for i in range(0, 180):
84         print(i)
85         test.move_servo_position(0, i, 180)
86         time.sleep(.001)
87     for i in range(180, 0, -1):
88         print(i)
89         test.move_servo_position(0, i, 180)
90         time.sleep(.001)
91
92 #####
93 # Code below may damage servo, use with caution
94 # Test sweep for full range of servo (outside 0 to 180 degrees).
95 # while True:
96 #     for i in range(-45, 200):
97 #         print(i)
98 #         test.move_servo_position(0, i, 180)
99 #         time.sleep(.001)
100 #     for i in range(200, -45, -1):
101 #         print(i)
102 #         test.move_servo_position(0, i, 180)
103 #         time.sleep(.001)

```

7.4 Example 3: Get Servo Position for 180 Degree Servo

Listing 3: examples/ex3_get_position_180_deg_servo.py

```

1 #-----
2 # Pi Servo Hat - Example 3
3 #-----
4 #
5 # Written by SparkFun Electronics, June 2019
6 # Author: Wes Furuya
7 #
8 # Compatibility:
9 #     * Original: https://www.sparkfun.com/products/14328
10 #     * v2: https://www.sparkfun.com/products/15316
11 #
12 # Do you like this library? Help support SparkFun. Buy a board!
13 # For more information on Pi Servo Hat, check out the product page
14 # linked above.
15 #
16 # This program is distributed in the hope that it will be useful, but

```

(continues on next page)

(continued from previous page)

```

17 # WITHOUT ANY WARRANTY without even the implied warranty of
18 # MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
19 # General Public License for more details.
20 #
21 # You should have received a copy of the GNU General Public License
22 # along with this program. If not, see <http://www.gnu.org/licenses/>.
23 #
24 #=====
25 # Copyright (c) 2019 SparkFun Electronics
26 #
27 # Permission is hereby granted, free of charge, to any person obtaining
28 # a copy of this software and associated documentation files (the
29 # "Software"), to deal in the Software without restriction, including
30 # without limitation the rights to use, copy, modify, merge, publish,
31 # distribute, sublicense, and/or sell copies of the Software, and to
32 # permit persons to whom the Software is furnished to do so, subject to
33 # the following conditions:
34 #
35 # The above copyright notice and this permission notice shall be
36 # included in all copies or substantial portions of the Software.
37 #
38 # THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
39 # EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
40 # MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
41 # IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
42 # CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
43 # TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
44 # SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
45 #=====
46
47 """
48 This example should be used with a 180 degree (range of rotation) servo
49 on channel 0 of the Pi Servo Hat.
50
51 The extended code (commented out), at the end of the example could be
52 used to test the full range of the servo motion. However, users should
53 be wary as they can damage their servo by giving it a position outside
54 the standard range of motion.
55 """
56
57 import pi_servo_hat
58 import time
59
60 # Initialize Constructor
61 test = pi_servo_hat.PiServoHat()
62
63 # Restart Servo Hat (in case Hat is frozen/locked)
64 test.restart()
65
66 # Test Run
67 #####
68 # Moves servo position to 0 degrees (1ms), Channel 0
69 test.move_servo_position(0, 0, 180)
70
71 # Pause 1 sec
72 time.sleep(1)
73

```

(continues on next page)

(continued from previous page)

```

74 # Moves servo position to 180 degrees (2ms), Channel 0
75 test.move_servo_position(0, 180, 180)
76
77 # Pause 1 sec
78 time.sleep(1)
79
80 # Sweep
81 #####
82 while True:
83     for i in range(0, 180):
84         print("Input: ", end = '')
85         print(i, end = '')
86         test.move_servo_position(0, i, 180)
87         print(" Estimated Pos: ", end = '')
88         print(test.get_servo_position(0, 180))
89         time.sleep(.05)
90     for i in range(180, 0, -1):
91         print("Input: ", end = '')
92         print(i, end = '')
93         test.move_servo_position(0, i, 180)
94         print(" Estimated Pos: ", end = '')
95         print(test.get_servo_position(0, 180))
96         time.sleep(.05)

```

7.5 Example 4: Change PWM Frequency for 180 Degree Servo

Listing 4: examples/ex4_change_pwm_frequency_180_deg_servo.py

```

1 #-----
2 # Pi Servo Hat - Example 4
3 #-----
4 #
5 # Written by SparkFun Electronics, June 2019
6 # Author: Wes Furuya
7 #
8 # Compatibility:
9 #     * Original: https://www.sparkfun.com/products/14328
10 #     * v2: https://www.sparkfun.com/products/15316
11 #
12 # Do you like this library? Help support SparkFun. Buy a board!
13 # For more information on Pi Servo Hat, check out the product page
14 # linked above.
15 #
16 # This program is distributed in the hope that it will be useful, but
17 # WITHOUT ANY WARRANTY without even the implied warranty of
18 # MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
19 # General Public License for more details.
20 #
21 # You should have received a copy of the GNU General Public License
22 # along with this program. If not, see <http://www.gnu.org/licenses/>.
23 #
24 #-----
25 # Copyright (c) 2019 SparkFun Electronics
26 #

```

(continues on next page)

(continued from previous page)

```

27 # Permission is hereby granted, free of charge, to any person obtaining
28 # a copy of this software and associated documentation files (the
29 # "Software"), to deal in the Software without restriction, including
30 # without limitation the rights to use, copy, modify, merge, publish,
31 # distribute, sublicense, and/or sell copies of the Software, and to
32 # permit persons to whom the Software is furnished to do so, subject to
33 # the following conditions:
34 #
35 # The above copyright notice and this permission notice shall be
36 # included in all copies or substantial portions of the Software.
37 #
38 # THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
39 # EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
40 # MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
41 # IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
42 # CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
43 # TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
44 # SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
45 #=====
46
47 """
48 This example should be used with a 180 degree (range of rotation) servo
49 on channel 0 of the Pi Servo Hat.
50
51 The extended code (commented out), at the end of the example could be
52 used to test the full range of the servo motion. However, users should
53 be wary as they can damage their servo by giving it a position outside
54 the standard range of motion.
55 """
56
57 import pi_servo_hat
58 import time
59
60 # Initialize Constructor
61 test = pi_servo_hat.PiServoHat()
62
63 # Restart Servo Hat (in case Hat is frozen/locked)
64 test.restart()
65
66 # Test Run
67 #####
68 # Moves servo position to 0 degrees (1ms), Channel 0
69 test.move_servo_position(0, 0, 180)
70
71 # Pause 1 sec
72 time.sleep(1)
73
74 # Moves servo position to 180 degrees (2ms), Channel 0
75 test.move_servo_position(0, 180, 180)
76
77 # Pause 1 sec
78 time.sleep(1)
79
80 # 50 Hz Test
81 #####
82 # Set PWM Frequency to 50 Hz
83 test.set_pwm_frequency(50)

```

(continues on next page)

(continued from previous page)

```
84
85 # Sweep from 0 to 180 degrees and back
86 for i in range(0, 180):
87     print("Input: ", end = '')
88     print(i, end = '')
89     test.move_servo_position(0, i, 180)
90     print(" Estimated Pos: ", end = '')
91     print(test.get_servo_position(0, 180))
92     time.sleep(.01)
93 for i in range(180, 0, -1):
94     print("Input: ", end = '')
95     print(i, end = '')
96     test.move_servo_position(0, i, 180)
97     print(" Estimated Pos: ", end = '')
98     print(test.get_servo_position(0, 180))
99     time.sleep(.01)
100
101
102 # 100 Hz Test
103 #####
104 # Set PWM Frequency to 100 Hz
105 test.set_pwm_frequency(100)
106
107 # Sweep from 0 to 180 degrees and back
108 for i in range(0, 180):
109     print("Input: ", end = '')
110     print(i, end = '')
111     test.move_servo_position(0, i, 180)
112     print(" Estimated Pos: ", end = '')
113     print(test.get_servo_position(0, 180))
114     time.sleep(.05)
115 for i in range(180, 0, -1):
116     print("Input: ", end = '')
117     print(i, end = '')
118     test.move_servo_position(0, i, 180)
119     print(" Estimated Pos: ", end = '')
120     print(test.get_servo_position(0, 180))
121     time.sleep(.05)
122
123
124 # 200 Hz Test
125 #####
126 # Set PWM Frequency to 200 Hz
127 test.set_pwm_frequency(200)
128
129 # Sweep from 0 to 180 degrees and back
130 for i in range(0, 180):
131     print("Input: ", end = '')
132     print(i, end = '')
133     test.move_servo_position(0, i, 180)
134     print(" Estimated Pos: ", end = '')
135     print(test.get_servo_position(0, 180))
136     time.sleep(.05)
137 for i in range(180, 0, -1):
138     print("Input: ", end = '')
139     print(i, end = '')
140     test.move_servo_position(0, i, 180)
```

(continues on next page)

(continued from previous page)

```
141 print(" Estimated Pos: ", end = '')
142 print(test.get_servo_position(0, 180))
143 time.sleep(.05)
```


p

pi_servo_hat, 15

G

`get_pwm_frequency()` (*pi_servo_hat.PiServoHat* method), 15

`get_servo_position()` (*pi_servo_hat.PiServoHat* method), 15

M

`move_servo_position()`
(*pi_servo_hat.PiServoHat* method), 16

P

`pi_servo_hat` (module), 15

`PiServoHat` (class in *pi_servo_hat*), 15

R

`restart()` (*pi_servo_hat.PiServoHat* method), 16

S

`set_duty_cycle()` (*pi_servo_hat.PiServoHat* method), 16

`set_pwm_frequency()` (*pi_servo_hat.PiServoHat* method), 16