
qwiic_max3010x
Release 0.0.1

Feb 15, 2021

Contents:

1	Contents	3
2	Supported Platforms	5
3	Dependencies	7
3.1	Documentation	7
3.2	Installation	7
3.2.1	PyPi Installation	7
3.2.2	Local Installation	7
3.3	Example Use	8
4	Table of Contents	9
4.1	API Reference	9
4.1.1	qwiic_max3010x	9
4.2	Basic Readings Example	15
4.3	Presence Sensing Example	17
4.4	Temperature Sense Example	19
4.5	Heartbeat Plotter Example	21
4.6	Heartrate Example	24
4.7	FIFO Readings Example	27
5	Indices and tables	31
	Python Module Index	33
	Index	35

Python package for the [SparkFun Photodetector Breakout - MAX30101](#) (Qwiic)

This package is a port of the existing [SparkFun MAX3010x Sensor 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

- *Supported Platforms*
- *Dependencies*
- *Installation*
- *Documentation*
- *Examples*

CHAPTER 2

Supported Platforms

The qwiic Python package current supports the following platforms:

- [Raspberry Pi](#)
- [NVidia Jetson Nano](#)

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

3.1 Documentation

The SparkFun qwiic Max3010x module documentation is hosted at [ReadTheDocs](#)

3.2 Installation

3.2.1 PyPi Installation

This repository is hosted on PyPi as the [sparkfun-qwiic-max3010x](#) 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-max3010x
```

For the current user:

```
pip install sparkfun_qwiic_max3010x
```

3.2.2 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_max3010x-<version>.tar.gz
```

3.3 Example Use

See the *examples directory* for more detailed use examples.

```
from __future__ import print_function
import qwiic_max3010x
import time
import sys

def runExample():

    print("\nSparkFun MAX3010x Photodetector Sensor - Example 1\n")
    sensor = qwiic_max3010x.QwiicMax3010x()

    if sensor.begin() == False:
        print("The Qwiic MAX3010x device isn't connected to the system. Please check_
↳ your connection", \
            file=sys.stderr)
        return
    else:
        print("The Qwiic MAX3010x is connected.")

    if sensor.setup() == False:
        print("Device setup failure. Please check your connection", \
            file=sys.stderr)
        return
    else:
        print("Setup complete.")

    while True:
        print(\
            'R[' , sensor.getRed() , ']' \t'\
            'IR[' , sensor.getIR() , ']' \t'\
            'G[' , sensor.getGreen() , ']' \
        )
        time.sleep(0.1)

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

4.1 API Reference

4.1.1 qwiic_max3010x

Python module for the [SparkFun Photodetector Breakout - MAX30101 (Qwiic)](<https://www.sparkfun.com/products/16474>)

This python package is a port of the existing [SparkFun MAX3010x Sensor Arduino Library](https://github.com/sparkfun/SparkFun_MAX3010x_Sensor_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_max3010x.**QwiicMax3010x** (*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 QwiicMax3010x device object.

Return type Object

available ()

Tell caller how many samples are available

Returns number of samples available

Return type integer

begin ()

Initialize the operation of the Qwiic MAX3010x module

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

Return type bool

bit_mask (*reg, mask, thing*)

Given a register, read it, mask it, and then set the thing

Parameters

- **reg** – the register you’d like to effect
- **mask** – the mask needed to zero-out the portion of the register we’re interested in
- **thing** – the thing we are affecting aka the control bits of the register

Returns Returns true of the register write was successful, otherwise False.

Return type bool

check ()

Polls the sensor for new data Call regularly If new data is available, it updates the head and tail in the main lists of data

Returns number of new samples obtained

Return type integer

checkForBeat (*sample*)

Wrapper function to allow access to function within supporting heart_rate.py file

Parameters **sample** – IR sample

Returns True if a beat is detected, otherwise False

Return type boolean

clearFIFO ()

Resets all points to start in a known state

Returns no return value

connected

Determine if a device is connected to the system..

Returns True if the device is connected, otherwise False.

Return type bool

disableAFULL ()

Disable AFULL Interrupt

Returns no return value

disableALCOVF ()

Disable ALCOVF Interrupt

Returns no return value

disableDATARDY ()

Disable DATARDY Interrupt

Returns no return value

disableDIETEMPRDY ()

Disable DIETEMPRDY Interrupt

Returns no return value

disableFIFORollover()

Disable roll over if FIFO over flows

Returns no return value**disablePROXINT()**

Disable PROXINT Interrupt

Returns no return value**disableSlots()**

Clears all slot assignments

Returns no return value**enableAFULL()**

Enable AFULL Interrupt

Returns no return value**enableALCOVF()**

Enable ALCOVF Interrupt

Returns no return value**enableDATARDY()**

Enable DATARDY Interrupt

Returns no return value**enableDIETEMPRDY()**

Enable DIETEMPRDY Interrupt

Returns no return value**enableFIFORollover()**

Enable roll over if FIFO over flows

Returns no return value**enablePROXINT()**

Enable PROXINT Interrupt

Returns no return value**enableSlot(slotNumber, device)**

Given a slot number assign a thing to it Devices are SLOT_RED_LED or SLOT_RED_PILOT (proximity)

Assigning a SLOT_RED_LED will pulse LED Assigning a SLOT_RED_PILOT will ??

Parameters

- **slotNumber** – slot number as int 1,2,3,4
- **device** – which device (aka led) you'd like to assign to the given slot

Returns Whether or not the configuration write was successful**Return type** bool**getFIFOGreen()**

Report the next Green value in the FIFO

Returns the next Green value in the FIFO**Return type** integer

getFIFOIR()

Report the next IR value in the FIFO

Returns the next IR value in the FIFO

Return type integer

getFIFOred()

Report the next Red value in the FIFO

Returns the next Red value in the FIFO

Return type integer

getGreen()

Report the most recent GREEN value

Returns value of GREEN light sensor from most recent sample

Return type integer

getINT1()

Returns the value of the INTSTAT1 Register

Returns value of the INTSTAT1 Register

Return type integer

getINT2()

Returns the value of the INTSTAT2 Register

Returns value of the INTSTAT2 Register

Return type integer

getIR()

Report the most recent IR value

Returns value of IR light sensor from most recent sample

Return type integer

getReadPointer()

Read the FIFO Read Pointer

Returns FIFO read pointer value

Return type integer

getRed()

Report the most recent red value

Returns value of RED light sensor from most recent sample

Return type integer

getRevisionID()

Report Revision ID from current variable in this class

Returns Revision ID

Return type integer

getWritePointer()

Read the FIFO Write Pointer

Returns FIFO write pointer value

Return type integer

is_connected()

Determine if a device is connected to the system..

Returns True if the device is connected, otherwise False.

Return type bool

millis()

Returns the current time in milliseconds

Returns Returns current system time in milliseconds

Return type int32_t

nextSample()

Advance the tail

Returns no return value

readPartID()

Report Part ID from the sensor

Returns Part ID

Return type integer

readRevisionID()

Report Revision ID from the sensor

Returns Revision ID

Return type integer

readTemperature()

Report Die Temperature in C

Returns die temp in C

Return type float

readTemperatureF()

Returns die temp in F

Returns die temp in F

Return type float

safeCheck(maxTimeToCheck)

Check for new data but give up after a certain amount of time Returns true if new data was found Returns false if new data was not found

Parameters **maxTimeToCheck** – milliseconds to timeout

Returns True if new data was found, otherwise False

Return type boolean

setADCRange(adcRange)

Set adcRange: one of MAX30105_ADCRANGE_2048, _4096, _8192, _16384

Parameters **adcRange** – MAX30105_ADCRANGE_2048, _4096, _8192, _16384

Returns no return value

setFIFOAlmostFull(numberOfSamples)

Set number of samples to trigger the almost full interrupt

Parameters **numberOfSamples** – default is 32 samples. Note it's reverse (0x00 is 32 samples, 0x0F is 17 samples)

Returns no return value

setFIFOAverage (*numberOfSamples*)

Set sample average

Parameters **numberOfSamples** – MAX30105_SAMPLEAVG_1, _2, _4, _8, _16, _32

Returns no return value

setLEDMode (*mode*)

Set which LEDs are used for sampling - Red only, RED+IR only, or custom

Parameters **mode** – Red only, RED+IR only, or custom

Returns no return value

setProximityThreshold (*threshMSB*)

Set the IR ADC count that will trigger the beginning of particle-sensing mode. The threshMSB signifies only the 8 most significant-bits of the ADC count. Note, this is specific to the MAX30105, and not included in the MAX30101

Parameters **threshMSB** – threshold of ADC count to cause trigger

Returns no return value

setPulseAmplitudeGreen (*amplitude*)

Set pulse amplitude (mA) of green LED

Parameters **amplitude** – 0x00 = 0mA, 0x7F = 25.4mA, 0xFF = 50mA (typical)

Returns no return value

setPulseAmplitudeIR (*amplitude*)

Set pulse amplitude (mA) of IR LED

Parameters **amplitude** – 0x00 = 0mA, 0x7F = 25.4mA, 0xFF = 50mA (typical)

Returns no return value

setPulseAmplitudeProximity (*amplitude*)

Set pulse amplitude while in proximity mode (only MAX30105) Note, this is specific to the MAX30105, and not included in the MAX30101

Parameters **amplitude** – amplitude

Returns no return value

setPulseAmplitudeRed (*amplitude*)

Set pulse amplitude (mA) of red LED

Parameters **amplitude** – 0x00 = 0mA, 0x7F = 25.4mA, 0xFF = 50mA (typical)

Returns no return value

setPulseWidth (*pulseWidth*)

Set pulseWidth: one of MAX30105_PULSEWIDTH_69, _188, _215, _411

Parameters **pulseWidth** – MAX30105_PULSEWIDTH_69, _188, _215, _411

Returns no return value

setSampleRate (*sampleRate*)

Set sampleRate: one of MAX30105_SAMPLERATE_50, _100, _200, _400, _800, _1000, _1600, _3200

Parameters `sampleRate` – MAX30105_SAMPLERATE_50, _100, _200, _400, _800, _1000, _1600, _3200

Returns no return value

setup (`powerLevel=31`, `sampleAverage=4`, `ledMode=3`, `sampleRate=400`, `pulseWidth=411`, `adcRange=4096`)

Setup the MAX3010x with default or custom settings

Parameters

- **powerLevel** – 0x00 = 0mA, 0x7F = 25.4mA, 0xFF = 50mA
- **sampleAverage** – int, 1,2,4,8,16,32, default is 4
- **ledMode** – 1 = RED, 2=RED+IR , 3=RED+IR+GREEN
- **sampleRate** – 0-3200
- **pulseWidth** – 0-411 (microseconds)
- **adcRange** – 2048,4096,8192,16384

Returns no return value

shutDown ()

Put IC into low power mode

Returns no return value

softReset ()

Command a soft reset

Returns Returns true if the soft reset was successful, otherwise False.

Return type bool

wakeUp ()

Pull IC out of low power mode

Returns no return value

4.2 Basic Readings Example

Listing 1: examples/ex1_Basic_Readings.py

```

1  #!/usr/bin/env python
2  #-----
3  # ex1_Basic_Readings.py
4  #
5  # Simple example for the qwiic MAX3010x device
6  # Outputs all Red/IR/Green values.
7  #
8  #-----
9  #
10 # Written by Pete Lewis
11 # SparkFun Electronics, May 2020
12 #
13 # Based on code from the SparkFun MAX3010x Sensor Arduino Library
14 # https://github.com/sparkfun/SparkFun_MAX3010x_Sensor_Library
15 # By: Nathan Seidle @ SparkFun Electronics, October 2nd, 2016

```

(continues on next page)

(continued from previous page)

```

16 #
17 # This python library supports the SparkFun Electronics qwiic
18 # qwiic sensor/board ecosystem on a Raspberry Pi (and compatible) single
19 # board computers.
20 #
21 # More information on qwiic is at https://www.sparkfun.com/qwiic
22 #
23 # Do you like this library? Help support SparkFun. Buy a board!
24 #
25 #=====
26 # Copyright (c) 2019 SparkFun Electronics
27 #
28 # Permission is hereby granted, free of charge, to any person obtaining a copy
29 # of this software and associated documentation files (the "Software"), to deal
30 # in the Software without restriction, including without limitation the rights
31 # to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
32 # copies of the Software, and to permit persons to whom the Software is
33 # furnished to do so, subject to the following conditions:
34 #
35 # The above copyright notice and this permission notice shall be included in all
36 # copies or substantial portions of the Software.
37 #
38 # THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
39 # IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
40 # FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
41 # AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
42 # LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
43 # OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
44 # SOFTWARE.
45 #=====
46 # Example 1
47 #
48
49 from __future__ import print_function
50 import qwiic_max3010x
51 import time
52 import sys
53
54 def runExample():
55
56     print("\nSparkFun MAX3010x Photodetector - Example 1\n")
57     sensor = qwiic_max3010x.QwiicMax3010x()
58
59     if sensor.begin() == False:
60         print("The Qwiic MAX3010x device isn't connected to the system.␣
↪Please check your connection", \
61             file=sys.stderr)
62         return
63     else:
64         print("The Qwiic MAX3010x is connected.")
65
66     if sensor.setup() == False:
67         print("Device setup failure. Please check your connection", \
68             file=sys.stderr)
69         return
70     else:
71         print("Setup complete.")

```

(continues on next page)

(continued from previous page)

```

72
73     while True:
74         print(\
75             'R[' , sensor.getRed() , ']' \t' \
76             'IR[' , sensor.getIR() ,
77             'G[' , sensor.getGreen() ,
78             )
79         time.sleep(0.1)
80
81 if __name__ == '__main__':
82     try:
83         runExample()
84     except (KeyboardInterrupt, SystemExit) as exErr:
85         print("\nEnding Example 1")
86         sys.exit(0)
87
88

```

4.3 Presence Sensing Example

Listing 2: examples/ex2_Presence_Sensing.py

```

1  #!/usr/bin/env python
2  # -----
3  # ex2_Presence_Sensing.py
4  #
5  # Simple example for the qwiic MAX3010x device
6  # This takes an average reading at power up and if the reading changes more than 100,
7  # then print 'Something is there!'.
8  #
9  # -----
10 #
11 # Written by Pete Lewis
12 # SparkFun Electronics, May 2020
13 #
14 # Based on code from the SparkFun MAX3010x Sensor Arduino Library
15 # https://github.com/sparkfun/SparkFun_MAX3010x_Sensor_Library
16 # By: Nathan Seidle @ SparkFun Electronics, October 2nd, 2016
17 #
18 # This python library supports the SparkFun Electronics qwiic
19 # qwiic sensor/board ecosystem on a Raspberry Pi (and compatible) single
20 # board computers.
21 #
22 # More information on qwiic is at https://www.sparkfun.com/qwiic
23 #
24 # Do you like this library? Help support SparkFun. Buy a board!
25 #
26 # =====
27 # Copyright (c) 2019 SparkFun Electronics
28 #
29 # Permission is hereby granted, free of charge, to any person obtaining a copy
30 # of this software and associated documentation files (the "Software"), to deal

```

(continues on next page)

(continued from previous page)

```

31 # in the Software without restriction, including without limitation the rights
32 # to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
33 # copies of the Software, and to permit persons to whom the Software is
34 # furnished to do so, subject to the following conditions:
35 #
36 # The above copyright notice and this permission notice shall be included in all
37 # copies or substantial portions of the Software.
38 #
39 # THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
40 # IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
41 # FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
42 # AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
43 # LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
44 # OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
45 # SOFTWARE.
46 #=====
47 # Example 2
48 #
49
50 from __future__ import print_function
51 import qwiic_max3010x
52 import time
53 import sys
54
55 def millis():
56     return int(round(time.time() * 1000))
57
58 def runExample():
59
60     print("\nSparkFun MAX3010x Photodetector - Example 1\n")
61     sensor = qwiic_max3010x.QwiicMax3010x()
62
63     if sensor.begin() == False:
64         print("The Qwiic MAX3010x device isn't connected to the system.␣
→Please check your connection", \
65             file=sys.stderr)
66         return
67     else:
68         print("The Qwiic MAX3010x is connected.")
69
70     # Setup to sense up to 18 inches, max LED brightness
71     ledBrightness = 0xFF # Options: 0=Off to 255=50mA
72     sampleAverage = 4 # Options: 1, 2, 4, 8, 16, 32
73     ledMode = 2 # Options: 1 = Red only, 2 = Red + IR, 3 = Red + IR + Green
74     sampleRate = 400 # Options: 50, 100, 200, 400, 800, 1000, 1600, 3200
75     pulseWidth = 411 # Options: 69, 118, 215, 411
76     adcRange = 2048 # Options: 2048, 4096, 8192, 16384
77
78     if sensor.setup(ledBrightness, sampleAverage, ledMode, sampleRate, pulseWidth,
→ adcRange) == False:
79         print("Device setup failure. Please check your connection", \
80             file=sys.stderr)
81         return
82     else:
83         print("Setup complete.")
84
85     sensor.setPulseAmplitudeRed(0) # Turn off Red LED

```

(continues on next page)

(continued from previous page)

```

86     sensor.setPulseAmplitudeGreen(0) # Turn off Green LED
87
88     samplesTaken = 0           # Counter for calculating the Hz or read rate
89     unblockedValue = 0         # Average IR at power up
90     startTime = 0             # Used to calculate measurement rate
91
92     # Take an average of IR readings at power up
93     unblockedValue = 0
94     for i in range(0,32):
95         unblockedValue += sensor.getIR() # Read the IR value
96     unblockedValue /= 32
97
98     startTime = millis()
99
100
101     while True:
102         samplesTaken += 1
103
104         IRSample = sensor.getIR()
105         hertz = samplesTaken / ((millis() - startTime) / 1000)
106         currentDelta = (IRSample - unblockedValue)
107
108         hertz = round(hertz, 2)
109         currentDelta = round(currentDelta, 2)
110
111         message = ' ' # blank message
112         if currentDelta > 100:
113             message = 'Something is there!'
114
115         print(\
116             'IR[' , IRSample , ']' \t',\
117             'Hz[' , hertz , ']' \t',\
118             'delta[' , currentDelta , ']' ,\
119             message
120         )
121
122
123 if __name__ == '__main__':
124     try:
125         runExample()
126     except (KeyboardInterrupt, SystemExit) as exErr:
127         print("\nEnding Example 2")
128         sys.exit(0)
129
130

```

4.4 Temperature Sense Example

Listing 3: examples/ex3_Temperature_Sense.py

```

1  #!/usr/bin/env python
2  #-----
3  # ex3_Temperature_Sesnse.py
4  #

```

(continues on next page)

(continued from previous page)

```

5  # Simple example for the qwiic MAX3010x device
6  # This demo outputs the onboard temperature sensor.
7  # The temp sensor is accurate to +/-1 C but
8  # has an astonishing precision of 0.0625 C.
9  #
10 #-----
11 #
12 # Written by Pete Lewis
13 # SparkFun Electronics, May 2020
14 #
15 # Based on code from the SparkFun MAX3010x Sensor Arduino Library
16 # https://github.com/sparkfun/SparkFun_MAX3010x_Sensor_Library
17 # By: Nathan Seidle @ SparkFun Electronics, October 2016
18 #
19 # This python library supports the SparkFun Electronics qwiic
20 # qwiic sensor/board ecosystem on a Raspberry Pi (and compatable) single
21 # board computers.
22 #
23 # More information on qwiic is at https://www.sparkfun.com/qwiic
24 #
25 # Do you like this library? Help support SparkFun. Buy a board!
26 #
27 #=====
28 # Copyright (c) 2019 SparkFun Electronics
29 #
30 # Permission is hereby granted, free of charge, to any person obtaining a copy
31 # of this software and associated documentation files (the "Software"), to deal
32 # in the Software without restriction, including without limitation the rights
33 # to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
34 # copies of the Software, and to permit persons to whom the Software is
35 # furnished to do so, subject to the following conditions:
36 #
37 # The above copyright notice and this permission notice shall be included in all
38 # copies or substantial portions of the Software.
39 #
40 # THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
41 # IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
42 # FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
43 # AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
44 # LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
45 # OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
46 # SOFTWARE.
47 #=====
48 # Example 3
49 #
50
51 from __future__ import print_function
52 import qwiic_max3010x
53 import time
54 import sys
55
56 def runExample():
57
58     print("\nSparkFun MAX3010x Photodetector - Example 3\n")
59     sensor = qwiic_max3010x.QwiicMax3010x()
60
61     if sensor.begin() == False:

```

(continues on next page)

(continued from previous page)

```

62         print("The Qwiic MAX3010x device isn't connected to the system.\n
↳Please check your connection", \
63             file=sys.stderr)
64         return
65     else:
66         print("The Qwiic MAX3010x is connected.")
67
68     # Setup Sensor
69     # The LEDs are very low power and won't affect the temp reading much but
70     # we will call setup() with LEDs off, to avoid any local heating (ledMode = 0)
71
72     if sensor.setup(ledMode = 0) == False:
73         print("Device setup failure. Please check your connection", \
74             file=sys.stderr)
75         return
76     else:
77         print("Setup complete.")
78
79     sensor.enabledIETEMPRDY() # Enable the temp ready interrupt. This is required.
80
81     while True:
82         temperature = sensor.readTemperature()
83         temperatureF = sensor.readTemperatureF()
84
85         temperature = round(temperature, 4)
86         temperatureF = round(temperatureF, 4)
87
88         print(\
89             'temperatureC[', temperature , ']' \t',\
90             'temperatureF[', temperatureF , ']',\
91             )
92
93
94 if __name__ == '__main__':
95     try:
96         runExample()
97     except (KeyboardInterrupt, SystemExit) as exErr:
98         print("\nEnding Example 3")
99         sys.exit(0)
100
101

```

4.5 Heartbeat Plotter Example

Listing 4: examples/ex4_HeartBeat_Plotter.py

```

1  #!/usr/bin/env python
2  #-----
3  # ex4_HeartBeat_Plotter.py
4  #
5  # Simple example for the qwiic MAX3010x device
6  # Shows the user's heart beat on a graphical plotter
7  # Using Matplotlib
8  # To learn more about plotting data in python check out this tutorial:

```

(continues on next page)

(continued from previous page)

```

9  # https://learn.sparkfun.com/tutorials/graph-sensor-data-with-python-and-matplotlib/
10 # The example code below was built using the code from the previously mentioned
    ↳ tutorial.
11 # Thanks Shawn Hymel!
12 #
13 # Instructions:
14 #   1) Install Matplotlib (see below)
15 #   2) Connect sensor to system via qwiic cable
16 #   3) Attach sensor to your finger with a rubber band (see below)
17 #   4) Run this python example
18 #   5) Checkout the blips!
19 #   6) Feel the pulse on your neck and watch it mimic the blips
20 #
21 # It is best to attach the sensor to your finger using a rubber band or other
    ↳ tightening
22 # device. Humans are generally bad at applying constant pressure to a thing. When
    ↳ you
23 # press your finger against the sensor it varies enough to cause the blood in your
24 # finger to flow differently which causes the sensor readings to go wonky.
25 #
26 # Matplotlib install
27 # Install Dependencies
28 # Like any good Linux project, we need to install a number of dependencies and
    ↳ libraries
29 # in order to get matplotlib to run properly. Make sure you have an Internet
    ↳ connection
30 # and in a terminal, enter the following commands. You may need to wait several
    ↳ minutes
31 # while the various packages are downloaded and installed.
32 #
33 # sudo apt-get update
34 # sudo apt-get install libatlas3-base libffi-dev at-spi2-core python3-gi-cairo
35 # sudo pip3 install cairocffi
36 # sudo pip3 install matplotlib
37 #
38 #-----
39 #
40 # Written by Pete Lewis
41 # SparkFun Electronics, May 2020
42 #
43 # Based on code from the SparkFun MAX3010x Sensor Arduino Library
44 # https://github.com/sparkfun/SparkFun_MAX3010x_Sensor_Library
45 # By: Nathan Seidle @ SparkFun Electronics, October 2nd, 2016
46 #
47 # This python library supports the SparkFun Electronics qwiic
48 # qwiic sensor/board ecosystem on a Raspberry Pi (and compatible) single
49 # board computers.
50 #
51 # More information on qwiic is at https://www.sparkfun.com/qwiic
52 #
53 # Do you like this library? Help support SparkFun. Buy a board!
54 #
55 #=====
56 # Copyright (c) 2019 SparkFun Electronics
57 #
58 # Permission is hereby granted, free of charge, to any person obtaining a copy
59 # of this software and associated documentation files (the "Software"), to deal

```

(continues on next page)

(continued from previous page)

```

60 # in the Software without restriction, including without limitation the rights
61 # to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
62 # copies of the Software, and to permit persons to whom the Software is
63 # furnished to do so, subject to the following conditions:
64 #
65 # The above copyright notice and this permission notice shall be included in all
66 # copies or substantial portions of the Software.
67 #
68 # THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
69 # IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
70 # FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
71 # AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
72 # LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
73 # OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
74 # SOFTWARE.
75 #=====
76 # Example 4
77 #
78
79 from __future__ import print_function
80 import qwiic_max3010x
81 import time
82 import sys
83
84 sensor = qwiic_max3010x.QwiicMax3010x()
85
86 #Plotter Stuff
87 import matplotlib.pyplot as plt
88 import matplotlib.animation as animation
89
90 # Create figure for plotting
91 fig = plt.figure()
92 ax = fig.add_subplot(1, 1, 1)
93 xlen= 100 #sample number, increments and is used for labeling x axis in plot
94 xs = list(range(0,xlen))
95 ys = [0]*xlen
96 line, = ax.plot(xs, ys)
97 plt.title('Heartbeat over time')
98 plt.ylabel('IR Value')
99
100 # This function is called periodically from FuncAnimation
101 def animate(i, ys):
102     # Read IR from MAX3010x
103     ir = sensor.getIR()
104
105     ys.append(ir)
106     ys = ys[-xlen:]
107     line.set_ydata(ys)
108     ax.set_ylim([min(ys),max(ys)])
109
110     return line,
111
112
113 def runExample():
114
115     print("\nSparkFun MAX3010x Photodetector - Example 4\n")
116

```

(continues on next page)

(continued from previous page)

```

117     if sensor.begin() == False:
118         print("The Qwiic MAX3010x device isn't connected to the system.␣
↪Please check your connection", \
119             file=sys.stderr)
120         return
121     else:
122         print("The Qwiic MAX3010x is connected.")
123
124     # Setup to sensor
125     ledBrightness = 0x1F # Options: 0=Off to 255=50mA
126     sampleAverage = 8 # Options: 1, 2, 4, 8, 16, 32
127     ledMode = 3 # Options: 1 = Red only, 2 = Red + IR, 3 = Red + IR + Green
128     sampleRate = 100 # Options: 50, 100, 200, 400, 800, 1000, 1600, 3200
129     pulseWidth = 411 # Options: 69, 118, 215, 411
130     adcRange = 4096 # Options: 2048, 4096, 8192, 16384
131
132     if sensor.setup(ledBrightness, sampleAverage, ledMode, sampleRate, pulseWidth,
↪ adcRange) == False:
133         print("Device setup failure. Please check your connection", \
134             file=sys.stderr)
135         return
136     else:
137         print("Setup complete.")
138         # Set up plot to call animate() function periodically
139         ani = animation.FuncAnimation(fig, animate, fargs=(ys,), interval=10,␣
↪ blit=True)
140         plt.show()
141
142
143 if __name__ == '__main__':
144     try:
145         runExample()
146     except (KeyboardInterrupt, SystemExit) as exErr:
147         print("\nEnding Example 4")
148         sys.exit(0)
149
150

```

4.6 Heartrate Example

Listing 5: examples/ex5_HeartRate.py

```

1  #!/usr/bin/env python
2  #-----
3  # ex5_HeartRate.py
4  #
5  # Simple example for the qwiic MAX3010x device
6  # This is a demo to show the reading of heart rate or beats per minute (BPM) using
7  # a Penpheral Beat Amplitude (PBA) algorithm.
8  #
9  # It is best to attach the sensor to your finger using a rubber band or other␣
↪tightening
10 # device. Humans are generally bad at applying constant pressure to a thing. When you
11 # press your finger against the sensor it varies enough to cause the blood in your

```

(continues on next page)

(continued from previous page)

```

12 # finger to flow differently which causes the sensor readings to go wonky.
13 #
14 #-----
15 #
16 # Written by Pete Lewis
17 # SparkFun Electronics, May 2020
18 #
19 # Based on code from the SparkFun MAX3010x Sensor Arduino Library
20 # https://github.com/sparkfun/SparkFun_MAX3010x_Sensor_Library
21 # By: Nathan Seidle @ SparkFun Electronics, October 2nd, 2016
22 #
23 # This python library supports the SparkFun Electroncis qwiic
24 # qwiic sensor/board ecosystem on a Raspberry Pi (and compatable) single
25 # board computers.
26 #
27 # More information on qwiic is at https://www.sparkfun.com/qwiic
28 #
29 # Do you like this library? Help support SparkFun. Buy a board!
30 #
31 #=====
32 # Copyright (c) 2019 SparkFun Electronics
33 #
34 # Permission is hereby granted, free of charge, to any person obtaining a copy
35 # of this software and associated documentation files (the "Software"), to deal
36 # in the Software without restriction, including without limitation the rights
37 # to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
38 # copies of the Software, and to permit persons to whom the Software is
39 # furnished to do so, subject to the following conditions:
40 #
41 # The above copyright notice and this permission notice shall be included in all
42 # copies or substantial portions of the Software.
43 #
44 # THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
45 # IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
46 # FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
47 # AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
48 # LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
49 # OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
50 # SOFTWARE.
51 #=====
52 # Example 5
53 #
54
55 from __future__ import print_function
56 import qwiic_max3010x
57 import time
58 import sys
59
60 def millis():
61     return int(round(time.time() * 1000))
62
63 def runExample():
64
65     print("\nSparkFun MAX3010x Photodetector - Example 5\n")
66     sensor = qwiic_max3010x.QwiicMax3010x()
67
68     if sensor.begin() == False:

```

(continues on next page)

(continued from previous page)

```

69         print("The Qwiic MAX3010x device isn't connected to the system.␣
↪Please check your connection", \
70             file=sys.stderr)
71         return
72     else:
73         print("The Qwiic MAX3010x is connected.")
74
75     print("Place your index finger on the sensor with steady pressure.")
76
77     if sensor.setup() == False:
78         print("Device setup failure. Please check your connection", \
79             file=sys.stderr)
80         return
81     else:
82         print("Setup complete.")
83
84     sensor.setPulseAmplitudeRed(0x0A) # Turn Red LED to low to indicate sensor is␣
↪running
85     sensor.setPulseAmplitudeGreen(0) # Turn off Green LED
86
87     RATE_SIZE = 4 # Increase this for more averaging. 4 is good.
88     rates = list(range(RATE_SIZE)) # list of heart rates
89     rateSpot = 0
90     lastBeat = 0 # Time at which the last beat occurred
91     beatsPerMinute = 0.00
92     beatAvg = 0
93     samplesTaken = 0 # Counter for calculating the Hz or read rate
94     startTime = millis() # Used to calculate measurement rate
95
96     while True:
97
98         irValue = sensor.getIR()
99         samplesTaken += 1
100        if sensor.checkForBeat(irValue) == True:
101            # We sensed a beat!
102            print('BEAT')
103            delta = ( millis() - lastBeat )
104            lastBeat = millis()
105
106            beatsPerMinute = 60 / (delta / 1000.0)
107            beatsPerMinute = round(beatsPerMinute,1)
108
109            if beatsPerMinute < 255 and beatsPerMinute > 20:
110                rateSpot += 1
111                rateSpot %= RATE_SIZE # Wrap variable
112                rates[rateSpot] = beatsPerMinute # Store this reading␣
↪in the array
113
114                # Take average of readings
115                beatAvg = 0
116                for x in range(0, RATE_SIZE):
117                    beatAvg += rates[x]
118                beatAvg /= RATE_SIZE
119                beatAvg = round(beatAvg)
120
121        Hz = round(float(samplesTaken) / ( ( millis() - startTime ) / 1000.0␣
↪) , 2)

```

(continues on next page)

(continued from previous page)

```

122         if (samplesTaken % 200 ) == 0:
123
124             print(\
125                 'IR=', irValue , ' \t',\
126                 'BPM=', beatsPerMinute , '\t',\
127                 # 'DCE
→ ', getDCE() , '\t',\
128                 'Avg=', beatAvg , '\t',\
129                 'Hz=', Hz, \
130                 )
131
132 if __name__ == '__main__':
133     try:
134         runExample()
135     except (KeyboardInterrupt, SystemExit) as exErr:
136         print("\nEnding Example 5")
137         sys.exit(0)
138
139

```

4.7 FIFO Readings Example

Listing 6: examples/ex6_FIFO_Readings.py

```

1  #!/usr/bin/env python
2  #-----
3  # ex6_FIFO_Readings.py
4  #
5  # Simple example for the qwiic MAX3010x device
6  # Outputs all Red/IR/Green values at about 5Hz by polling the FIFO
7  # Note, the Hz is slowed down by printing to the python terminal,
8  # It can be dramatically speed up if printing is omitted or intermittent
9  #
10 #-----
11 #
12 # Written by Pete Lewis
13 # SparkFun Electronics, May 2020
14 #
15 # Based on code from the SparkFun MAX3010x Sensor Arduino Library
16 # https://github.com/sparkfun/SparkFun_MAX3010x_Sensor_Library
17 # By: Nathan Seidle @ SparkFun Electronics, October 2nd, 2016
18 #
19 # This python library supports the SparkFun Electronics qwiic
20 # qwiic sensor/board ecosystem on a Raspberry Pi (and compatible) single
21 # board computers.
22 #
23 # More information on qwiic is at https://www.sparkfun.com/qwiic
24 #
25 # Do you like this library? Help support SparkFun. Buy a board!
26 #
27 #=====
28 # Copyright (c) 2019 SparkFun Electronics
29 #
30 # Permission is hereby granted, free of charge, to any person obtaining a copy

```

(continues on next page)

(continued from previous page)

```

31 # of this software and associated documentation files (the "Software"), to deal
32 # in the Software without restriction, including without limitation the rights
33 # to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
34 # copies of the Software, and to permit persons to whom the Software is
35 # furnished to do so, subject to the following conditions:
36 #
37 # The above copyright notice and this permission notice shall be included in all
38 # copies or substantial portions of the Software.
39 #
40 # THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
41 # IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
42 # FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
43 # AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
44 # LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
45 # OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
46 # SOFTWARE.
47 =====
48 # Example 6
49 #
50
51 from __future__ import print_function
52 import qwiic_max3010x
53 import time
54 import sys
55
56 def millis():
57     return int(round(time.time() * 1000))
58
59 def runExample():
60
61     print("\nSparkFun MAX3010x Photodetector - Example 6\n")
62     sensor = qwiic_max3010x.QwiicMax3010x()
63
64     if sensor.begin() == False:
65         print("The Qwiic MAX3010x device isn't connected to the system.␣
↪Please check your connection", \
66             file=sys.stderr)
67         return
68     else:
69         print("The Qwiic MAX3010x is connected.")
70
71         # Setup to sense up to 18 inches, max LED brightness
72         ledBrightness = 0xFF # Options: 0=Off to 255=50mA
73         sampleAverage = 4 # Options: 1, 2, 4, 8, 16, 32
74         ledMode = 2 # Options: 1 = Red only, 2 = Red + IR, 3 = Red + IR + Green
75         sampleRate = 400 # Options: 50, 100, 200, 400, 800, 1000, 1600, 3200
76         pulseWidth = 411 # Options: 69, 118, 215, 411
77         adcRange = 2048 # Options: 2048, 4096, 8192, 16384
78
79         if sensor.setup(ledBrightness, sampleAverage, ledMode, sampleRate, pulseWidth,
↪ adcRange) == False:
80             print("Device setup failure. Please check your connection", \
81                 file=sys.stderr)
82             return
83         else:
84             print("Setup complete.")
85

```

(continues on next page)

(continued from previous page)

```

86     startTime = millis() # Used to calculate measurement rate
87     samplesTaken = 0 # Counter for calculating the Hz or read rate
88
89     while True:
90
91         sensor.check() # Check the sensor, read up to 3 samples
92
93         while (sensor.available() > 0): # do we have new data?
94
95             samplesTaken += 1
96
97             hertz = samplesTaken / ((millis() - startTime) / 1000)
98             hertz = round(hertz, 2)
99
100            print(\
101                'R[' , sensor.getFIFORed() , ']' \t'\
102                'IR[' , sensor.getFIFOIR() , ']' \t'\
103                'G[' , sensor.getFIFOGreen() , ']' \t'\
104                'Hz[' , hertz , ']' \
105            )
106
107            sensor.nextSample() # We're finished with this sample so move_
↳to next sample
108
109 if __name__ == '__main__':
110     try:
111         runExample()
112     except (KeyboardInterrupt, SystemExit) as exErr:
113         print("\nEnding Example 6")
114         sys.exit(0)
115
116

```


CHAPTER 5

Indices and tables

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

q

`qwiic_max3010x`, 9

A

`available()` (*qwiic_max3010x.QwiicMax3010x method*), 9

B

`begin()` (*qwiic_max3010x.QwiicMax3010x method*), 9
`bit_mask()` (*qwiic_max3010x.QwiicMax3010x method*), 10

C

`check()` (*qwiic_max3010x.QwiicMax3010x method*), 10
`checkForBeat()` (*qwiic_max3010x.QwiicMax3010x method*), 10
`clearFIFO()` (*qwiic_max3010x.QwiicMax3010x method*), 10
`connected` (*qwiic_max3010x.QwiicMax3010x attribute*), 10

D

`disableAFULL()` (*qwiic_max3010x.QwiicMax3010x method*), 10
`disableALCOVF()` (*qwiic_max3010x.QwiicMax3010x method*), 10
`disableDATARDY()` (*qwiic_max3010x.QwiicMax3010x method*), 10
`disableDIETEMPRDY()` (*qwiic_max3010x.QwiicMax3010x method*), 10
`disableFIFORollover()` (*qwiic_max3010x.QwiicMax3010x method*), 10
`disablePROXINT()` (*qwiic_max3010x.QwiicMax3010x method*), 11
`disableSlots()` (*qwiic_max3010x.QwiicMax3010x method*), 11

E

`enableAFULL()` (*qwiic_max3010x.QwiicMax3010x method*), 11

`enableALCOVF()` (*qwiic_max3010x.QwiicMax3010x method*), 11
`enableDATARDY()` (*qwiic_max3010x.QwiicMax3010x method*), 11
`enableDIETEMPRDY()` (*qwiic_max3010x.QwiicMax3010x method*), 11
`enableFIFORollover()` (*qwiic_max3010x.QwiicMax3010x method*), 11
`enablePROXINT()` (*qwiic_max3010x.QwiicMax3010x method*), 11
`enableSlot()` (*qwiic_max3010x.QwiicMax3010x method*), 11

G

`getFIFOGreen()` (*qwiic_max3010x.QwiicMax3010x method*), 11
`getFIFOIR()` (*qwiic_max3010x.QwiicMax3010x method*), 11
`getFIFORed()` (*qwiic_max3010x.QwiicMax3010x method*), 12
`getGreen()` (*qwiic_max3010x.QwiicMax3010x method*), 12
`getINT1()` (*qwiic_max3010x.QwiicMax3010x method*), 12
`getINT2()` (*qwiic_max3010x.QwiicMax3010x method*), 12
`getIR()` (*qwiic_max3010x.QwiicMax3010x method*), 12
`getReadPointer()` (*qwiic_max3010x.QwiicMax3010x method*), 12
`getRed()` (*qwiic_max3010x.QwiicMax3010x method*), 12
`getRevisionID()` (*qwiic_max3010x.QwiicMax3010x method*), 12
`getWritePointer()` (*qwiic_max3010x.QwiicMax3010x method*), 12

I

`is_connected()` (*qwiic_max3010x.QwiicMax3010x method*), 13

M

`millis()` (*qwiic_max3010x.QwiicMax3010x method*), 13

N

`nextSample()` (*qwiic_max3010x.QwiicMax3010x method*), 13

Q

`qwiic_max3010x` (*module*), 9

`QwiicMax3010x` (*class in qwiic_max3010x*), 9

R

`readPartID()` (*qwiic_max3010x.QwiicMax3010x method*), 13

`readRevisionID()` (*qwiic_max3010x.QwiicMax3010x method*), 13

`readTemperature()` (*qwiic_max3010x.QwiicMax3010x method*), 13

`readTemperatureF()` (*qwiic_max3010x.QwiicMax3010x method*), 13

S

`safeCheck()` (*qwiic_max3010x.QwiicMax3010x method*), 13

`setADCRange()` (*qwiic_max3010x.QwiicMax3010x method*), 13

`setFIFOAlmostFull()` (*qwiic_max3010x.QwiicMax3010x method*), 13

`setFIFOAverage()` (*qwiic_max3010x.QwiicMax3010x method*), 14

`setLEDMode()` (*qwiic_max3010x.QwiicMax3010x method*), 14

`setProximityThreshold()` (*qwiic_max3010x.QwiicMax3010x method*), 14

`setPulseAmplitudeGreen()` (*qwiic_max3010x.QwiicMax3010x method*), 14

`setPulseAmplitudeIR()` (*qwiic_max3010x.QwiicMax3010x method*), 14

`setPulseAmplitudeProximity()` (*qwiic_max3010x.QwiicMax3010x method*), 14

`setPulseAmplitudeRed()` (*qwiic_max3010x.QwiicMax3010x method*), 14

`setPulseWidth()` (*qwiic_max3010x.QwiicMax3010x method*), 14

`setSampleRate()` (*qwiic_max3010x.QwiicMax3010x method*), 14

`setup()` (*qwiic_max3010x.QwiicMax3010x method*), 15

`shutDown()` (*qwiic_max3010x.QwiicMax3010x method*), 15

`softReset()` (*qwiic_max3010x.QwiicMax3010x method*), 15

W

`wakeUp()` (*qwiic_max3010x.QwiicMax3010x method*), 15