
sparkfun*ubloxgps*
Release 1.1.0

Aug 15, 2023

Contents:

1	Contents	3
2	Supported Platforms	5
3	Documentation	7
4	Installation	9
4.1	PyPi Installation	9
4.2	Local Installation	9
5	Example Use	11
5.1	Examples Directory	11
5.2	Attribution	12
5.3	To Do	12
5.3.1	Table of Contents	12
5.3.1.1	API Reference	12
5.3.1.2	Get latitude and longitude	12
5.3.1.3	Get GPS time	13
5.3.1.4	Dead Reckoning	15
5.3.1.5	Stream NMEA data	16
5.3.1.6	Modifying Configuration Settings	17
5.3.1.7	SPI and Ublox Modules	18
5.3.2	Indices and tables	20

This is a Python module for the SparkFun GPS products based on u-blox GPS modules.

This package is included in the overall [SparkFun qwiic Python Package](#). While the module itself does not use IC, it may none the less join the ranks when the Raspberry Pi has better support for clock stretching. None the less, a Qwiic connector has been included onboard so the GPS module can be used with our along side SparkFun's Qwiic products.

CHAPTER 1

Contents

- *Supported Platforms*
- *Dependencies*
- *Installation*
- *Documentation*
- *Example Use*

CHAPTER 2

Supported Platforms

The u-blox gps Python package currently supports the following platforms:

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

CHAPTER 3

Documentation

The SparkFun u-blox gps module documentation is hosted at [ReadTheDocs](#)

4.1 PyPi Installation

This repository is hosted on PyPi as the `sparkfun-ublox_gps` 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-ublox-gps
```

For the current user:

```
sudo pip install sparkfun-ublox-gps
```

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



```
from ublox_gps import UbloxGps
import serial
# Can also use SPI here - import spidev
# I2C is not supported

port = serial.Serial('/dev/serial0', baudrate=38400, timeout=1)
gps = UbloxGps(port)

def run():

    try:
        print("Listening for UBX Messages.")
        while True:
            try:
                coords = gps.geo_coords()
                print(coords.lon, coords.lat)
            except (ValueError, IOError) as err:
                print(err)

    finally:
        port.close()

if __name__ == '__main__':
    run()
```

5.1 Examples Directory

- geo_coords_ex1.py
 - Simple example showing how to get and print out latitude, longitude, and heading.

- `gps_time_ex2.py`
 - Simple example showing how to UTC time and how to check its' validity.
- `dead_reckoning_ex3.py`
 - **Simple example showing how to use dead reckoning on dead reckoning modules.** Make sure to take a look at our [hookup guide](#) for a detailed explanation on where to attach the module and how to calibrate it.
- `stream_nmea_gps_ex4.py`
 - Simple example showing how to stream NMEA data from Ublox Modules.
- `modifying_configuration_settings_ex5.py`
 - **Simple example showing how change the configuration settings for the Ublox Module.**
- `using_spi_ex6.py`
 - **Simple example showing how to use SPI Module.**

5.2 Attribution

This code is dependent on the work by [daylomople](#) and the awesome parsing capabilities of `ubxtranslator`.

5.3 To Do

- [] Some bugs associated with SPI writes.
- [] **Add more Classes and Messages to `sparkfun_predefines.py` so that a greater variety of messages can be parsed.**
- [] Fix bug when getting configuration settings.

5.3.1 Table of Contents

5.3.1.1 API Reference

5.3.1.2 Get latitude and longitude

Listing 1: `examples/geo_coords_ex1.py`

```
1 #!/usr/bin/env python3
2 #-----
3 # geo_coords_ex1.py
4 #
5 # Simple Example for SparkFun ublox GPS products
6 #-----
7 #
8 # Written by SparkFun Electronics, July 2020
```

(continues on next page)

(continued from previous page)

```

9 #
10 # Do you like this library? Help support SparkFun. Buy a board!
11 # https://sparkfun.com
12 #=====
13 # GNU GPL License 3.0
14 # Copyright (c) 2020 SparkFun Electronics
15 #
16 # This program is free software: you can redistribute it and/or modify
17 # it under the terms of the GNU General Public License as published by
18 # the Free Software Foundation, either version 3 of the License, or
19 # (at your option) any later version.
20 #
21 # This program is distributed in the hope that it will be useful,
22 # but WITHOUT ANY WARRANTY; without even the implied warranty of
23 # MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
24 # GNU General Public License for more details.
25 #
26 # You should have received a copy of the GNU General Public License
27 # along with this program. If not, see <https://www.gnu.org/licenses/>.
28 #=====
29 # Example 1
30 # This example sets up the serial port and then passes it to the UbloxGps
31 # library. From here we call geo_coords() and to get longitude and latitude. I've
32 # also included heading of motion here as well.
33
34 import serial
35
36 from ublox_gps import UbloxGps
37
38 port = serial.Serial('/dev/serial0', baudrate=38400, timeout=1)
39 gps = UbloxGps(port)
40
41 def run():
42
43     try:
44         print("Listening for UBX Messages")
45         while True:
46             try:
47                 geo = gps.geo_coords()
48                 print("Longitude: ", geo.lon)
49                 print("Latitude: ", geo.lat)
50                 print("Heading of Motion: ", geo.headMot)
51             except (ValueError, IOError) as err:
52                 print(err)
53
54     finally:
55         port.close()
56
57
58 if __name__ == '__main__':
59     run()

```

5.3.1.3 Get GPS time

Listing 2: examples/gps_time_ex2.py

```

1  #!/usr/bin/env python3
2  #-----
3  # gps_time_ex2.py
4  #
5  # Simple Example for SparkFun ublox GPS products
6  #-----
7  #
8  # Written by SparkFun Electronics, July 2020
9  #
10 # Do you like this library? Help support SparkFun. Buy a board!
11 # https://sparkfun.com
12 #=====
13 # GNU GPL License 3.0
14 # Copyright (c) 2020 SparkFun Electronics
15 #
16 # This program is free software: you can redistribute it and/or modify
17 # it under the terms of the GNU General Public License as published by
18 # the Free Software Foundation, either version 3 of the License, or
19 # (at your option) any later version.
20 #
21 # This program is distributed in the hope that it will be useful,
22 # but WITHOUT ANY WARRANTY; without even the implied warranty of
23 # MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
24 # GNU General Public License for more details.
25 #
26 # You should have received a copy of the GNU General Public License
27 # along with this program. If not, see <https://www.gnu.org/licenses/>.
28 #=====
29 # Example 2
30 # This example sets up the serial port and then passes it to the UbloxGPS
31 # library. From here we call date_time() and to get gps time and check whether
32 # the data received is "valid" which indicates that the probability of the time
33 # to be correct is very high.
34
35 import serial
36
37 from ublox_gps import UbloxGps
38
39 port = serial.Serial('/dev/serial0', baudrate=38400, timeout=1)
40 gps = UbloxGps(port)
41
42 def run():
43
44     try:
45         print("Listening for UBX Messages")
46         while True:
47             try:
48                 gps_time = gps.date_time()
49                 print("{}{}/{}{}/{}".format(gps_time.day, gps_time.month,
50                                             gps_time.year))
51                 print("UTC Time {}: {}: {}".format(gps_time.hour, gps_time.min,
52                                                     gps_time.sec))
53                 print("Valid date: {} \n Valid Time: {}".format(gps_time.valid.validDate,
54                                                                 gps_time.valid.
55 ↪ validTime))

```

(continues on next page)

(continued from previous page)

```

55         except (ValueError, IOError) as err:
56             print(err)
57
58     finally:
59         port.close()
60
61
62 if __name__ == '__main__':
63     run()

```

5.3.1.4 Dead Reckoning

Listing 3: examples/dead_reckoning_ex3.py

```

1  #!/usr/bin/env python3
2  #-----
3  # dead_reckoning_ex3.py
4  #
5  # Simple Example for SparkFun ublox GPS products
6  #-----
7  #
8  # Written by SparkFun Electronics, July 2020
9  #
10 # Do you like this library? Help support SparkFun. Buy a board!
11 # https://sparkfun.com
12 #=====
13 # GNU GPL License 3.0
14 # Copyright (c) 2020 SparkFun Electronics
15 #
16 # This program is free software: you can redistribute it and/or modify
17 # it under the terms of the GNU General Public License as published by
18 # the Free Software Foundation, either version 3 of the License, or
19 # (at your option) any later version.
20 #
21 # This program is distributed in the hope that it will be useful,
22 # but WITHOUT ANY WARRANTY; without even the implied warranty of
23 # MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
24 # GNU General Public License for more details.
25 #
26 # You should have received a copy of the GNU General Public License
27 # along with this program. If not, see <https://www.gnu.org/licenses/>.
28 #
29 #=====
30 # Example 3
31 # This example sets up the serial port and then passes it to the UbloxGPS
32 # library. From here we call veh_attitude() to get the
33 # the data received is "valid" which indicates that the probability of the time
34 # to be correct is very high.
35
36 import serial
37
38 from ublox_gps import UbloxGps
39
40 port = serial.Serial('/dev/serial0', baudrate=38400, timeout=1)
41 gps = UbloxGps(port)

```

(continues on next page)

(continued from previous page)

```

42
43 def run():
44
45     try:
46         print("Listening for UBX Messages")
47         while True:
48             try:
49                 veh = gps.veh_attitude()
50                 print("Roll: ", veh.roll)
51                 print("Pitch: ", veh.pitch)
52                 print("Heading: ", veh.heading)
53                 print("Roll Acceleration: ", veh.accRoll)
54                 print("Pitch Acceleration: ", veh.accPitch)
55                 print("Heading Acceleration: ", veh.accHeading)
56             except (ValueError, IOError) as err:
57                 print(err)
58
59         finally:
60             port.close()
61
62
63 if __name__ == '__main__':
64     run()

```

5.3.1.5 Stream NMEA data

Listing 4: examples/stream_nmea_gps_ex4.py

```

1  #!/usr/bin/env python3
2  #-----
3  # stream_nmea_gps_ex4.py
4  #
5  # Simple Example for SparkFun ublox GPS products
6  #-----
7  #
8  # Written by SparkFun Electronics, July 2020
9  #
10 # Do you like this library? Help support SparkFun. Buy a board!
11 # https://sparkfun.com
12 #=====
13 # GNU GPL License 3.0
14 # Copyright (c) 2020 SparkFun Electronics
15 #
16 # This program is free software: you can redistribute it and/or modify
17 # it under the terms of the GNU General Public License as published by
18 # the Free Software Foundation, either version 3 of the License, or
19 # (at your option) any later version.
20 #
21 # This program is distributed in the hope that it will be useful,
22 # but WITHOUT ANY WARRANTY; without even the implied warranty of
23 # MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
24 # GNU General Public License for more details.
25 #
26 # You should have received a copy of the GNU General Public License
27 # along with this program. If not, see <https://www.gnu.org/licenses/>.

```

(continues on next page)

(continued from previous page)

```

28 #=====
29 # Example 4
30 # This example sets up the serial port and then passes it to the UbloxGPS
31 # library. Now it's as simple as calling stream_nmea(). The function
32 # stream_nmea() returns the full NMEA sentence which could be passed to an NMEA
33 # parser.
34
35 import serial
36
37 from ublox_gps import UbloxGps
38
39 port = serial.Serial('/dev/serial0', baudrate=38400, timeout=1)
40 gps = UbloxGps(port)
41
42 def run():
43
44     try:
45         print("Listening for UBX Messages")
46         while True:
47             try:
48                 print(gps.stream_nmea())
49             except (ValueError, IOError) as err:
50                 print(err)
51
52     finally:
53         port.close()
54
55
56 if __name__ == '__main__':
57     run()

```

5.3.1.6 Modifying Configuration Settings

Listing 5: examples/modifying_configuration_settings_ex5.py

```

1  #!/usr/bin/env python3
2  #-----
3  # modifying_configuration_settings_ex5.py
4  #
5  # Simple Example for SparkFun ublox GPS products
6  #-----
7  #
8  # Written by SparkFun Electronics, July 2020
9  #
10 # Do you like this library? Help support SparkFun. Buy a board!
11 # https://sparkfun.com
12 #=====
13 # GNU GPL License 3.0
14 # Copyright (c) 2020 SparkFun Electronics
15 #
16 # This program is free software: you can redistribute it and/or modify
17 # it under the terms of the GNU General Public License as published by
18 # the Free Software Foundation, either version 3 of the License, or
19 # (at your option) any later version.
20 #

```

(continues on next page)

(continued from previous page)

```

21 # This program is distributed in the hope that it will be useful,
22 # but WITHOUT ANY WARRANTY; without even the implied warranty of
23 # MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
24 # GNU General Public License for more details.
25 #
26 # You should have received a copy of the GNU General Public License
27 # along with this program. If not, see <https://www.gnu.org/licenses/>.
28 #=====
29 # Example 5
30 # Ublox has changed their method for changing your module's settings. The new
31 # method requires passing the module the "key id" of the configuration you want
32 # to modify followed by the value you want to set. If you're requesting
33 # information then you'd pass the key id without a value. In both cases the
34 # response will be the configuration data that was set upon success and a NAK
35 # on a failure. Check the interface description datasheet for more information.
36
37 import serial
38
39 from ublox_gps import UbloxGps
40
41 port = serial.Serial('/dev/serial0', baudrate=38400, timeout=1)
42 gps = UbloxGps(port)
43
44 def run():
45
46     try:
47         print("Listening for UBX Messages")
48         while True:
49             try:
50                 # Get NMEA Protocol Version
51                 get_set = gps.ubx_get_set_del(0x20930001)
52                 print(get_set)
53             except (ValueError, IOError) as err:
54                 print(err)
55
56         finally:
57             port.close()
58
59
60 if __name__ == '__main__':
61     run()

```

5.3.1.7 SPI and Ublox Modules

Listing 6: examples/using_spi_ex6.py

```

1 #!/usr/bin/env python3
2 #-----
3 # using_spi_ex6.py
4 #
5 # Simple Example for SparkFun ublox GPS products
6 #-----
7 #
8 # Written by SparkFun Electronics, July 2020
9 #

```

(continues on next page)

(continued from previous page)

```

10 # Do you like this library? Help support SparkFun. Buy a board!
11 # https://sparkfun.com
12 #=====
13 # GNU GPL License 3.0
14 # Copyright (c) 2020 SparkFun Electronics
15 #
16 # This program is free software: you can redistribute it and/or modify
17 # it under the terms of the GNU General Public License as published by
18 # the Free Software Foundation, either version 3 of the License, or
19 # (at your option) any later version.
20 #
21 # This program is distributed in the hope that it will be useful,
22 # but WITHOUT ANY WARRANTY; without even the implied warranty of
23 # MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
24 # GNU General Public License for more details.
25 #
26 # You should have received a copy of the GNU General Public License
27 # along with this program. If not, see <https://www.gnu.org/licenses/>.
28 #=====
29 # Example 6
30 # To set up SPI is as simple as calling spidev and passing the port to the
31 # UbloxGps library. Other than this the library functions the same. There's no
32 # need to adjust the SPI settings because they are set behind the scenes
33 # according to the settings specified in the datasheet.
34 #
35 # Note as of 7/18 the SPI implementation still needs a little work.
36
37 import spidev
38
39 from ublox_gps import UbloxGps
40
41 port = spidev.SpiDev()
42 gps = UbloxGps(port)
43
44 def run():
45
46     try:
47         print("Listening for UBX Messages")
48         while True:
49             try:
50                 geo = gps.geo_coords()
51                 print("Longitude: ", geo.lon)
52                 print("Latitude: ", geo.lat)
53                 print("Heading of Motion: ", geo.headMot)
54             except (ValueError, IOError) as err:
55                 print(err)
56
57     finally:
58         port.close()
59
60
61 if __name__ == '__main__':
62     run()

```

5.3.2 Indices and tables

- genindex
- modindex
- search