

MPU6050 Sensor module

Generated by Doxygen 1.14.0

1 Namespace Index	1
1.1 Package List	1
2 Class Index	3
2.1 Class List	3
3 File Index	5
3.1 File List	5
4 Namespace Documentation	7
4.1 mpu6050 Namespace Reference	7
4.1.1 Variable Documentation	8
4.1.1.1 _MPU6050_ACCEL_CONFIG	8
4.1.1.2 _MPU6050_ACCEL_OUT	8
4.1.1.3 _MPU6050_CONFIG	8
4.1.1.4 _MPU6050_DEFAULT_ADDR	8
4.1.1.5 _MPU6050_DEVICE_ID	8
4.1.1.6 _MPU6050_GYRO_CONFIG	8
4.1.1.7 _MPU6050_GYRO_OUT	8
4.1.1.8 _MPU6050_PWR_MGMT_1	9
4.1.1.9 _MPU6050_SMPLRT_DIV	9
4.1.1.10 _MPU6050_TEMP_OUT	9
4.1.1.11 _MPU6050_WHO_AM_I	9
4.1.1.12 SENSORS_GRAVITY_STANDARD	9
5 Class Documentation	11
5.1 MPU6050.AccelRange Class Reference	11
5.1.1 Detailed Description	11
5.1.2 Member Data Documentation	11
5.1.2.1 RANGE_16_G	11
5.1.2.2 RANGE_2_G	11
5.1.2.3 RANGE_4_G	12
5.1.2.4 RANGE_8_G	12
5.2 MPU6050.FilterBandwidth Class Reference	12
5.2.1 Detailed Description	12
5.2.2 Member Data Documentation	12
5.2.2.1 BAND_10_HZ	12
5.2.2.2 BAND_184_HZ	13
5.2.2.3 BAND_21_HZ	13
5.2.2.4 BAND_260_HZ	13
5.2.2.5 BAND_44_HZ	13
5.2.2.6 BAND_5_HZ	13
5.2.2.7 BAND_94_HZ	13
5.3 MPU6050.GyroRange Class Reference	13

5.3.1 Detailed Description	14
5.3.2 Member Data Documentation	14
5.3.2.1 RANGE_1000_DEG	14
5.3.2.2 RANGE_2000_DEG	14
5.3.2.3 RANGE_250_DEG	14
5.3.2.4 RANGE_500_DEG	14
5.4 MPU6050 Class Reference	15
5.4.1 Detailed Description	16
5.4.2 Constructor & Destructor Documentation	16
5.4.2.1 __init__()	16
5.4.3 Member Function Documentation	16
5.4.3.1 _read_bytes()	16
5.4.3.2 _read_u8()	16
5.4.3.3 _write_u8()	16
5.4.3.4 get_accel_data()	16
5.4.3.5 get_all_data()	17
5.4.3.6 get_gyro_data()	17
5.4.3.7 get_temp_data()	17
5.4.3.8 reset()	17
5.4.3.9 set_accel_range()	17
5.4.3.10 set_filter_bandwidth()	18
5.4.3.11 set_gyro_range()	18
5.4.4 Member Data Documentation	18
5.4.4.1 _accel_range	18
5.4.4.2 _ACCEL_SCALES	18
5.4.4.3 _gyro_range	19
5.4.4.4 _GYRO_SCALES	19
5.4.4.5 address	19
5.4.4.6 i2c	19
6 File Documentation	21
6.1 mpu6050.py File Reference	21
6.1.1 Detailed Description	22
Index	23

Chapter 1

Namespace Index

1.1 Package List

Here are the packages with brief descriptions (if available):

mpu6050	7
-----------------------------------	---

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

MPU6050.AccelRange	
Accelerometer measurement range	11
MPU6050.FilterBandwidth	
Digital Low Pass Filter bandwidth	12
MPU6050.GyroRange	
Gyroscope measurement range	13
MPU6050	
Driver class for the MPU6050 sensor	15

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

mpu6050.py	
Micropython module for the MPU6050 6-DoF sensor	21

Chapter 4

Namespace Documentation

4.1 mpu6050 Namespace Reference

Classes

- class [MPU6050](#)
Driver class for the [MPU6050](#) sensor.

Variables

- float [SENSORS_GRAVITY_STANDARD](#) = 9.80665
Standard gravity for conversion to m/s^2 .

Register Map

- int [_MPU6050_DEFAULT_ADDR](#) = 0x68
Default I2C address.
- int [_MPU6050_DEVICE_ID](#) = 0x68
Device ID stored in WHO_AM_I register.
- int [_MPU6050_SMPLRT_DIV](#) = 0x19
Sample Rate Divider register.
- int [_MPU6050_CONFIG](#) = 0x1A
Configuration register.
- int [_MPU6050_GYRO_CONFIG](#) = 0x1B
Gyroscope Configuration register.
- int [_MPU6050_ACCEL_CONFIG](#) = 0x1C
Accelerometer Configuration register.
- int [_MPU6050_ACCEL_OUT](#) = 0x3B
Base address for accelerometer data registers.
- int [_MPU6050_TEMP_OUT](#) = 0x41
Base address for temperature data registers.
- int [_MPU6050_GYRO_OUT](#) = 0x43
Base address for gyroscope data registers.
- int [_MPU6050_PWR_MGMT_1](#) = 0x6B
Power Management 1 register.
- int [_MPU6050_WHO_AM_I](#) = 0x75
WHO_AM_I register, contains device ID.

4.1.1 Variable Documentation

4.1.1.1 `_MPU6050_ACCEL_CONFIG`

```
int _MPU6050_ACCEL_CONFIG = 0x1C [protected]
```

Accelerometer Configuration register.

4.1.1.2 `_MPU6050_ACCEL_OUT`

```
int _MPU6050_ACCEL_OUT = 0x3B [protected]
```

Base address for accelerometer data registers.

4.1.1.3 `_MPU6050_CONFIG`

```
int _MPU6050_CONFIG = 0x1A [protected]
```

Configuration register.

4.1.1.4 `_MPU6050_DEFAULT_ADDR`

```
int _MPU6050_DEFAULT_ADDR = 0x68 [protected]
```

Default I2C address.

4.1.1.5 `_MPU6050_DEVICE_ID`

```
int _MPU6050_DEVICE_ID = 0x68 [protected]
```

Device ID stored in WHO_AM_I register.

4.1.1.6 `_MPU6050_GYRO_CONFIG`

```
int _MPU6050_GYRO_CONFIG = 0x1B [protected]
```

Gyroscope Configuration register.

4.1.1.7 `_MPU6050_GYRO_OUT`

```
int _MPU6050_GYRO_OUT = 0x43 [protected]
```

Base address for gyroscope data registers.

4.1.1.8 _MPU6050_PWR_MGMT_1

```
int _MPU6050_PWR_MGMT_1 = 0x6B [protected]
```

Power Management 1 register.

4.1.1.9 _MPU6050_SMPLRT_DIV

```
int _MPU6050_SMPLRT_DIV = 0x19 [protected]
```

Sample Rate Divider register.

4.1.1.10 _MPU6050_TEMP_OUT

```
int _MPU6050_TEMP_OUT = 0x41 [protected]
```

Base address for temperature data registers.

4.1.1.11 _MPU6050_WHO_AM_I

```
int _MPU6050_WHO_AM_I = 0x75
```

WHO_AM_I register, contains device ID.

4.1.1.12 SENSORS_GRAVITY_STANDARD

```
float SENSORS_GRAVITY_STANDARD = 9.80665
```

Standard gravity for conversion to m/s^2 .

Chapter 5

Class Documentation

5.1 MPU6050.AccelRange Class Reference

Accelerometer measurement range.

Static Public Attributes

- int [RANGE_16_G](#) = 0x03
+/- 16g
- int [RANGE_2_G](#) = 0x00
+/- 2g
- int [RANGE_4_G](#) = 0x01
+/- 4g
- int [RANGE_8_G](#) = 0x02
+/- 8g

5.1.1 Detailed Description

Accelerometer measurement range.

5.1.2 Member Data Documentation

5.1.2.1 RANGE_16_G

```
int RANGE_16_G = 0x03 [static]
```

+/- 16g

5.1.2.2 RANGE_2_G

```
int RANGE_2_G = 0x00 [static]
```

+/- 2g

5.1.2.3 RANGE_4_G

```
int RANGE_4_G = 0x01 [static]
```

+/- 4g

5.1.2.4 RANGE_8_G

```
int RANGE_8_G = 0x02 [static]
```

+/- 8g

The documentation for this class was generated from the following file:

- [mpu6050.py](#)

5.2 MPU6050.FilterBandwidth Class Reference

Digital Low Pass Filter bandwidth.

Static Public Attributes

- int [BAND_10_HZ](#) = 0x05
10 Hz
- int [BAND_184_HZ](#) = 0x01
184 Hz
- int [BAND_21_HZ](#) = 0x04
21 Hz
- int [BAND_260_HZ](#) = 0x00
260 Hz
- int [BAND_44_HZ](#) = 0x03
44 Hz
- int [BAND_5_HZ](#) = 0x06
5 Hz
- int [BAND_94_HZ](#) = 0x02
94 Hz

5.2.1 Detailed Description

Digital Low Pass Filter bandwidth.

5.2.2 Member Data Documentation

5.2.2.1 BAND_10_HZ

```
int BAND_10_HZ = 0x05 [static]
```

10 Hz

5.2.2.2 BAND_184_HZ

```
int BAND_184_HZ = 0x01 [static]
```

184 Hz

5.2.2.3 BAND_21_HZ

```
int BAND_21_HZ = 0x04 [static]
```

21 Hz

5.2.2.4 BAND_260_HZ

```
int BAND_260_HZ = 0x00 [static]
```

260 Hz

5.2.2.5 BAND_44_HZ

```
int BAND_44_HZ = 0x03 [static]
```

44 Hz

5.2.2.6 BAND_5_HZ

```
int BAND_5_HZ = 0x06 [static]
```

5 Hz

5.2.2.7 BAND_94_HZ

```
int BAND_94_HZ = 0x02 [static]
```

94 Hz

The documentation for this class was generated from the following file:

- [mpu6050.py](#)

5.3 MPU6050.GyroRange Class Reference

Gyroscope measurement range.

Static Public Attributes

- int [RANGE_1000_DEG](#) = 0x02
+/- 1000 deg/s
- int [RANGE_2000_DEG](#) = 0x03
+/- 2000 deg/s
- int [RANGE_250_DEG](#) = 0x00
+/- 250 deg/s
- int [RANGE_500_DEG](#) = 0x01
+/- 500 deg/s

5.3.1 Detailed Description

Gyroscope measurement range.

5.3.2 Member Data Documentation

5.3.2.1 RANGE_1000_DEG

```
int RANGE_1000_DEG = 0x02 [static]
```

+/- 1000 deg/s

5.3.2.2 RANGE_2000_DEG

```
int RANGE_2000_DEG = 0x03 [static]
```

+/- 2000 deg/s

5.3.2.3 RANGE_250_DEG

```
int RANGE_250_DEG = 0x00 [static]
```

+/- 250 deg/s

5.3.2.4 RANGE_500_DEG

```
int RANGE_500_DEG = 0x01 [static]
```

+/- 500 deg/s

The documentation for this class was generated from the following file:

- [mpu6050.py](#)

5.4 MPU6050 Class Reference

Driver class for the [MPU6050](#) sensor.

Classes

- class [AccelRange](#)
Accelerometer measurement range.
- class [FilterBandwidth](#)
Digital Low Pass Filter bandwidth.
- class [GyroRange](#)
Gyroscope measurement range.

Public Member Functions

- [__init__](#) (self, I2C [i2c](#), int [address](#)=[_MPU6050_DEFAULT_ADDR](#))
Initializes the [MPU6050](#) sensor.
- tuple[float, float, float] [get_accel_data](#) (self, bool [as_g](#)=False)
Reads and converts the accelerometer data.
- dict [get_all_data](#) (self)
Reads all sensor data in a single, efficient transaction.
- tuple[float, float, float] [get_gyro_data](#) (self)
Reads and converts the gyroscope data in degrees per second.
- float [get_temp_data](#) (self)
Reads and converts the temperature data.
- [reset](#) (self)
Resets sensor registers to their default values.
- [set_accel_range](#) (self, int [accel_range](#))
Sets the accelerometer measurement range.
- [set_filter_bandwidth](#) (self, int [bandwidth](#))
Sets the Digital Low-Pass Filter bandwidth.
- [set_gyro_range](#) (self, int [gyro_range](#))
Sets the gyroscope measurement range.

Public Attributes

- [address](#) = address
- [i2c](#) = [i2c](#)

Protected Member Functions

- bytes [_read_bytes](#) (self, int register, int length)
- int [_read_u8](#) (self, int register)
- [_write_u8](#) (self, int register, int value)

Protected Attributes

- [_accel_range](#) = self.AccelRange.RANGE_2_G
- [_gyro_range](#) = self.GyroRange.RANGE_250_DEG

Static Protected Attributes

- dict [_ACCEL_SCALES](#)
- dict [_GYRO_SCALES](#)

5.4.1 Detailed Description

Driver class for the [MPU6050](#) sensor.

Allows reading of acceleration, gyroscope, and temperature data.

5.4.2 Constructor & Destructor Documentation

5.4.2.1 `__init__()`

```
__init__ (
    self,
    I2C i2c,
    int address = \_MPU6050\_DEFAULT\_ADDR)
```

Initializes the [MPU6050](#) sensor.

Parameters

<i>i2c</i>	The Micropython I2C object.
<i>address</i>	The I2C address of the sensor (default is 0x68).

5.4.3 Member Function Documentation

5.4.3.1 `_read_bytes()`

```
bytes _read_bytes (
    self,
    int register,
    int length) [protected]
```

5.4.3.2 `_read_u8()`

```
int _read_u8 (
    self,
    int register) [protected]
```

5.4.3.3 `_write_u8()`

```
_write_u8 (
    self,
    int register,
    int value) [protected]
```

5.4.3.4 `get_accel_data()`

```
tuple[float, float, float] get_accel_data (
    self,
    bool as_g = False)
```

Reads and converts the accelerometer data.

Parameters

<code>as↔ _g</code>	If True, returns values in 'g'. Otherwise, returns in m/s ² (default).
-------------------------	---

Returns

A tuple of 3 floats (x, y, z).

5.4.3.5 get_all_data()

```
dict get_all_data (
    self)
```

Reads all sensor data in a single, efficient transaction.

Returns

A dictionary containing 'accel' (m/s²), 'gyro' (°/s), and 'temp' (°C).

5.4.3.6 get_gyro_data()

```
tuple[float, float, float] get_gyro_data (
    self)
```

Reads and converts the gyroscope data in degrees per second.

Returns

A tuple of 3 floats (x, y, z) in °/s.

5.4.3.7 get_temp_data()

```
float get_temp_data (
    self)
```

Reads and converts the temperature data.

Returns

The temperature in degrees Celsius.

5.4.3.8 reset()

```
reset (
    self)
```

Resets sensor registers to their default values.

5.4.3.9 set_accel_range()

```
set_accel_range (
    self,
    int accel_range)
```

Sets the accelerometer measurement range.

Parameters

<i>accel_range</i>	Use constants from MPU6050.AccelRange .
--------------------	---

5.4.3.10 set_filter_bandwidth()

```
set_filter_bandwidth (
    self,
    int bandwidth)
```

Sets the Digital Low-Pass Filter bandwidth.

Parameters

<i>bandwidth</i>	Use constants from MPU6050.FilterBandwidth .
------------------	--

5.4.3.11 set_gyro_range()

```
set_gyro_range (
    self,
    int gyro_range)
```

Sets the gyroscope measurement range.

Parameters

<i>gyro_range</i>	Use constants from MPU6050.GyroRange .
-------------------	--

5.4.4 Member Data Documentation**5.4.4.1 _accel_range**

```
_accel_range = self.AccelRange.RANGE_2_G [protected]
```

5.4.4.2 _ACCEL_SCALES

```
dict _ACCEL_SCALES [static], [protected]
```

Initial value:

```
= {
    AccelRange.RANGE_2_G: 16384.0,
    AccelRange.RANGE_4_G: 8192.0,
    AccelRange.RANGE_8_G: 4096.0,
    AccelRange.RANGE_16_G: 2048.0,
}
```

5.4.4.3 `_gyro_range`

```
_gyro_range = self.GyroRange.RANGE_250_DEG [protected]
```

5.4.4.4 `_GYRO_SCALES`

```
dict _GYRO_SCALES [static], [protected]
```

Initial value:

```
= {
    GyroRange.RANGE_250_DEG: 131.0,
    GyroRange.RANGE_500_DEG: 65.5,
    GyroRange.RANGE_1000_DEG: 32.8,
    GyroRange.RANGE_2000_DEG: 16.4,
}
```

5.4.4.5 `address`

```
address = address
```

5.4.4.6 `i2c`

```
i2c = i2c
```

The documentation for this class was generated from the following file:

- [mpu6050.py](#)

Chapter 6

File Documentation

6.1 mpu6050.py File Reference

Micropython module for the MPU6050 6-DoF sensor.

Classes

- class [MPU6050.AccelRange](#)
Accelerometer measurement range.
- class [MPU6050.FilterBandwidth](#)
Digital Low Pass Filter bandwidth.
- class [MPU6050.GyroRange](#)
Gyroscope measurement range.
- class [MPU6050](#)
Driver class for the [MPU6050](#) sensor.

Namespaces

- namespace [mpu6050](#)

Variables

- float [SENSORS_GRAVITY_STANDARD](#) = 9.80665
Standard gravity for conversion to m/s^2 .

Register Map

- int [_MPU6050_ACCEL_CONFIG](#) = 0x1C
Accelerometer Configuration register.
- int [_MPU6050_ACCEL_OUT](#) = 0x3B
Base address for accelerometer data registers.
- int [_MPU6050_CONFIG](#) = 0x1A
Configuration register.
- int [_MPU6050_DEFAULT_ADDR](#) = 0x68
Default I2C address.

- `int _MPU6050_DEVICE_ID = 0x68`
Device ID stored in WHO_AM_I register.
- `int _MPU6050_GYRO_CONFIG = 0x1B`
Gyroscope Configuration register.
- `int _MPU6050_GYRO_OUT = 0x43`
Base address for gyroscope data registers.
- `int _MPU6050_PWR_MGMT_1 = 0x6B`
Power Management 1 register.
- `int _MPU6050_SMPLRT_DIV = 0x19`
Sample Rate Divider register.
- `int _MPU6050_TEMP_OUT = 0x41`
Base address for temperature data registers.
- `int _MPU6050_WHO_AM_I = 0x75`
WHO_AM_I register, contains device ID.

6.1.1 Detailed Description

Micropython module for the MPU6050 6-DoF sensor.

This module provides a class to interface with the MPU6050 accelerometer and gyroscope sensor over I2C.

Author

Adafruit (Original C++ library)
Gemini (Micropython conversion)

Date

August 2, 2025

Version

1.1

Index

- [_ACCEL_SCALES](#)
 - [MPU6050, 18](#)
 - [_GYRO_SCALES](#)
 - [MPU6050, 19](#)
 - [_MPU6050_ACCEL_CONFIG](#)
 - [mpu6050, 8](#)
 - [_MPU6050_ACCEL_OUT](#)
 - [mpu6050, 8](#)
 - [_MPU6050_CONFIG](#)
 - [mpu6050, 8](#)
 - [_MPU6050_DEFAULT_ADDR](#)
 - [mpu6050, 8](#)
 - [_MPU6050_DEVICE_ID](#)
 - [mpu6050, 8](#)
 - [_MPU6050_GYRO_CONFIG](#)
 - [mpu6050, 8](#)
 - [_MPU6050_GYRO_OUT](#)
 - [mpu6050, 8](#)
 - [_MPU6050_PWR_MGMT_1](#)
 - [mpu6050, 8](#)
 - [_MPU6050_SMPLRT_DIV](#)
 - [mpu6050, 9](#)
 - [_MPU6050_TEMP_OUT](#)
 - [mpu6050, 9](#)
 - [_MPU6050_WHO_AM_I](#)
 - [mpu6050, 9](#)
 - [__init__](#)
 - [MPU6050, 16](#)
 - [_accel_range](#)
 - [MPU6050, 18](#)
 - [_gyro_range](#)
 - [MPU6050, 18](#)
 - [_read_bytes](#)
 - [MPU6050, 16](#)
 - [_read_u8](#)
 - [MPU6050, 16](#)
 - [_write_u8](#)
 - [MPU6050, 16](#)
- [address](#)
 - [MPU6050, 19](#)
- [BAND_10_HZ](#)
 - [MPU6050.FilterBandwidth, 12](#)
- [BAND_184_HZ](#)
 - [MPU6050.FilterBandwidth, 12](#)
- [BAND_21_HZ](#)
 - [MPU6050.FilterBandwidth, 13](#)
- [BAND_260_HZ](#)
 - [MPU6050.FilterBandwidth, 13](#)
- [BAND_44_HZ](#)
 - [MPU6050.FilterBandwidth, 13](#)
- [BAND_5_HZ](#)
 - [MPU6050.FilterBandwidth, 13](#)
- [BAND_94_HZ](#)
 - [MPU6050.FilterBandwidth, 13](#)
- [get_accel_data](#)
 - [MPU6050, 16](#)
- [get_all_data](#)
 - [MPU6050, 17](#)
- [get_gyro_data](#)
 - [MPU6050, 17](#)
- [get_temp_data](#)
 - [MPU6050, 17](#)
- [i2c](#)
 - [MPU6050, 19](#)
- [MPU6050, 15](#)
 - [_ACCEL_SCALES, 18](#)
 - [_GYRO_SCALES, 19](#)
 - [__init__, 16](#)
 - [_accel_range, 18](#)
 - [_gyro_range, 18](#)
 - [_read_bytes, 16](#)
 - [_read_u8, 16](#)
 - [_write_u8, 16](#)
 - [address, 19](#)
 - [get_accel_data, 16](#)
 - [get_all_data, 17](#)
 - [get_gyro_data, 17](#)
 - [get_temp_data, 17](#)
 - [i2c, 19](#)
 - [reset, 17](#)
 - [set_accel_range, 17](#)
 - [set_filter_bandwidth, 18](#)
 - [set_gyro_range, 18](#)
- [mpu6050, 7](#)
 - [_MPU6050_ACCEL_CONFIG, 8](#)
 - [_MPU6050_ACCEL_OUT, 8](#)
 - [_MPU6050_CONFIG, 8](#)
 - [_MPU6050_DEFAULT_ADDR, 8](#)
 - [_MPU6050_DEVICE_ID, 8](#)
 - [_MPU6050_GYRO_CONFIG, 8](#)
 - [_MPU6050_GYRO_OUT, 8](#)
 - [_MPU6050_PWR_MGMT_1, 8](#)
 - [_MPU6050_SMPLRT_DIV, 9](#)
 - [_MPU6050_TEMP_OUT, 9](#)
 - [_MPU6050_WHO_AM_I, 9](#)

- SENSORS_GRAVITY_STANDARD, [9](#)
- MPU6050.AccelRange, [11](#)
 - RANGE_16_G, [11](#)
 - RANGE_2_G, [11](#)
 - RANGE_4_G, [11](#)
 - RANGE_8_G, [12](#)
- MPU6050.FilterBandwidth, [12](#)
 - BAND_10_HZ, [12](#)
 - BAND_184_HZ, [12](#)
 - BAND_21_HZ, [13](#)
 - BAND_260_HZ, [13](#)
 - BAND_44_HZ, [13](#)
 - BAND_5_HZ, [13](#)
 - BAND_94_HZ, [13](#)
- MPU6050.GyroRange, [13](#)
 - RANGE_1000_DEG, [14](#)
 - RANGE_2000_DEG, [14](#)
 - RANGE_250_DEG, [14](#)
 - RANGE_500_DEG, [14](#)
- mpu6050.py, [21](#)
- RANGE_1000_DEG
 - MPU6050.GyroRange, [14](#)
- RANGE_16_G
 - MPU6050.AccelRange, [11](#)
- RANGE_2000_DEG
 - MPU6050.GyroRange, [14](#)
- RANGE_250_DEG
 - MPU6050.GyroRange, [14](#)
- RANGE_2_G
 - MPU6050.AccelRange, [11](#)
- RANGE_4_G
 - MPU6050.AccelRange, [11](#)
- RANGE_500_DEG
 - MPU6050.GyroRange, [14](#)
- RANGE_8_G
 - MPU6050.AccelRange, [12](#)
- reset
 - MPU6050, [17](#)
- SENSORS_GRAVITY_STANDARD
 - mpu6050, [9](#)
- set_accel_range
 - MPU6050, [17](#)
- set_filter_bandwidth
 - MPU6050, [18](#)
- set_gyro_range
 - MPU6050, [18](#)