
# GMU100-BL Bluetooth Attitude Sensor

## Product manual

V1.1-20260623

## Technical Highlights


- ❖ Integrated nine-axis sensor and temperature sensor;
- ❖ Bluetooth 5.1 wireless transmission;
- ❖ Excellent vibration performance;
- ❖ 100Hz refresh rate;
- ❖ DC 3.0-4.2V power supply.

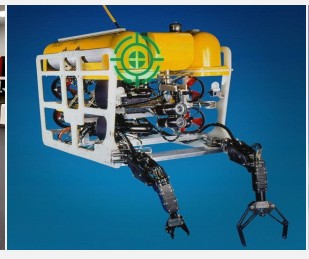
## Product Introduction

The GMU100-BL is a Bluetooth attitude sensor (AHRS miniature attitude reference system) designed for dynamic balancing applications in model aircraft, drones, robots, agricultural machinery, engineering vehicles, and forklifts, offering excellent cost-effectiveness. This product integrates data from a three-axis accelerometer, a three-axis magnetometer, and a three-axis gyroscope, supplemented by a temperature compensation algorithm and Kalman filtering for data fusion, enabling real-time attitude information output. Thanks to this technical architecture, the product maintains stable response performance and timeliness across the entire attitude range, making it suitable for various motion attitude monitoring applications. In addition to attitude data, the sensor supports outputting raw data across nine axes: three-axis acceleration, three-axis angular velocity, and three-axis magnetic field. Users can modify and expand system operating parameters to adapt to different application platforms. The product incorporates on-chip Flash IAP (in-circuit programming) technology, allowing users to save configuration parameters as needed, retaining the memory even after power loss. Data output methods include a USB 2.0 wired interface and Bluetooth wireless transmission (supporting Bluetooth 5.1 and BLE protocols). It integrates 2.4GHz adaptive frequency hopping technology, which can achieve stable transmission within a 40-meter line-of-sight range in unobstructed environments. It also supports firmware upgrades and multi-device networking synchronization, which helps to improve the convenience and flexibility of deployment in industrial sites and facilitates secondary development and application expansion for users.

## Application Scope

- ❖ Medical rehabilitation equipment monitoring
- ❖ Limb rehabilitation monitoring
- ❖ Human body capture
- ❖ Precision movement training for limbs

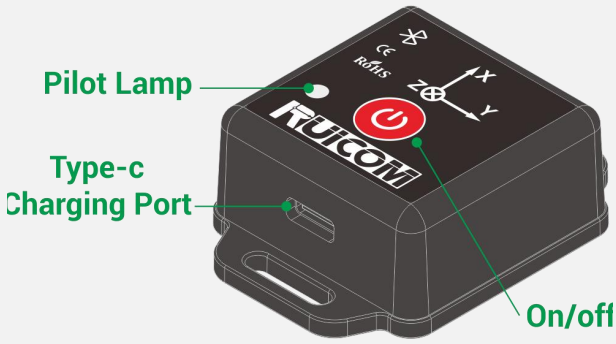





## Performance Parameters

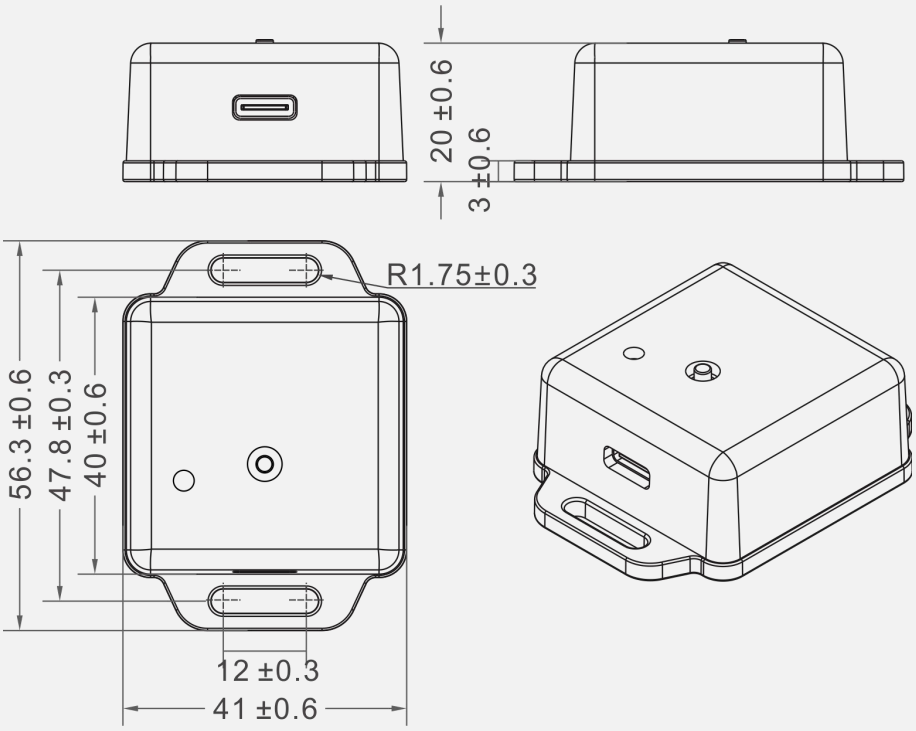
| GMU100-BL | | Parameter |
|---|---|---|
| Azimuth angle | Measurement Range | ±180°(Gyroscope and magnetic orientation are selectable.) |
| | Measurement Accuracy (RMS) @ 25℃ | <3° (No magnetic interference strip) |
| | Resolution | 0.1° |
| Tilt angle | Measurement Range | Roll ±180° Pitch ±90° |
| | Measurement Accuracy (RMS) @ 25℃ | <±0.5°(Dynamic) <±0.2°(Static) |
| | Resolution | <0.1° |
| Gyroscope | Measurement Range | ±500°/sec |
| | Zero Bias Stability (10s Smoothing) | 20°/hr |
| | Resolution | <0.1°/sec |
| | Bandwidth | >100Hz |
| Vibration Sensor | Measurement Range | ±8g |
| | Resolution | <10mg |
| | Bandwidth | >100Hz |
| Magnetic field meter | Measurement Range | ±8gauss |
| | Resolution | <2.5mgauss |
| | Bandwidth | 14-17Hz |
| Electrical connection | Power Supply Voltage | DC 3.0-4.2V |
| | Power Consumption | 50mA(Typical Value) |
| | Output Interface | USB2.0 |
| | Operating Temp | -5℃~55℃ |
| | Storage Temp | -10℃~+55℃ |
| Shock resistance | | 100g@11ms, triaxial (half-sine wave) |
| Vibration resistance | | 10grms、10～1000Hz |
| Waterproof rating | | IP65 |
| Connector | | Type-C Connector |
| Weight | | ≤35g |

## Function Description




1. ON/OFF Button: Press the button to power on, press the button to power off;
2. Indicator Light Status:
Flashing Red LED: The red LED flashes rapidly when powered on, indicating a connection to a Bluetooth host.
Solid Red LED: The red LED remains solid after connecting to the host.
Slow Flashing Red LED: The red LED flashes slowly when the battery is low, and the device automatically powers off after 30 seconds.
Solid Green LED: The green light remains solid while charging.
3.Type-C Charging Port: A 5V charging plug is recommended. Charging time is approximately 4 hours, and the theoretical working time is 8 hours.

## Shell Size




Shell Size: L56.3×W41×H20mm

Mounting Size: L47.8×W12×H3mm

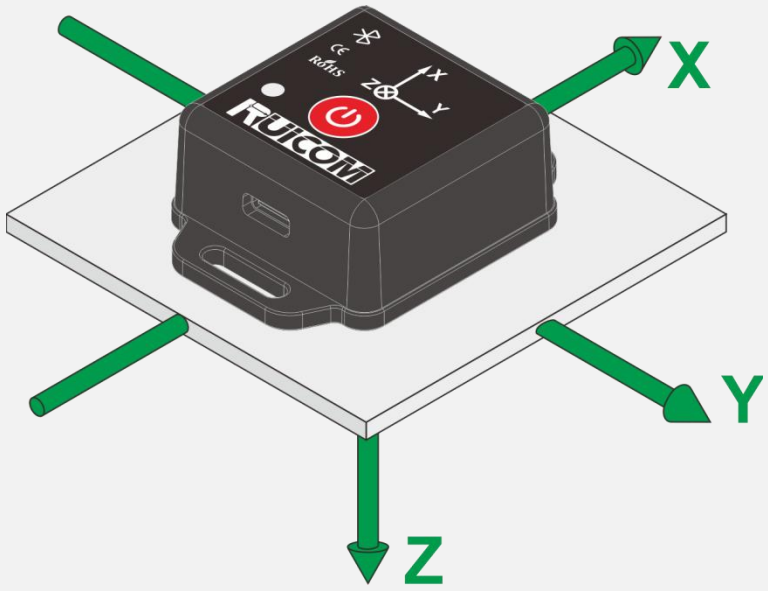
Mounting method: 2 M3 screws or Velcro strap

## Installation Orientation

Please install the tilt sensor correctly. Incorrect installation will lead to measurement errors. Pay special attention to two key aspects:

1) The mounting surface of the sensor must be tightly, flat, and stable against the surface being measured. Uneven mounting surfaces can easily cause angular errors in the sensor measurement.

2) The sensor axis must be parallel to the axis of the object being measured, and the two axes should ideally not form an angle.




## BLE Protocol

* The GMU100-BL uses Bluetooth and USB 2.0 communication protocols; the baud rate modification protocol is only applicable to USB 2.0.

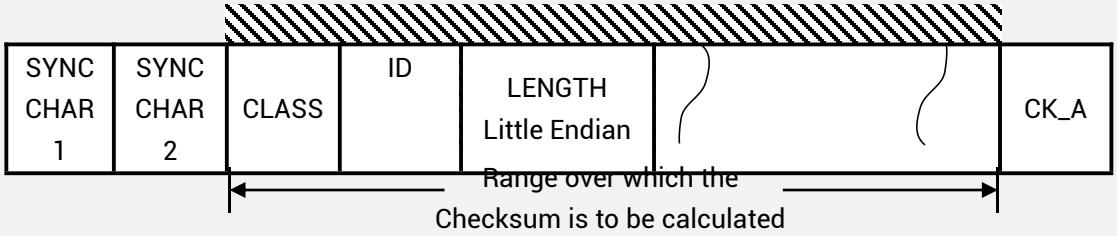
**Default parameters:** Baud Rate **115200bps**; Output Frequency **100Hz**; Output Data **Euler angles + sensor**; Operating mode **6 Axis**.

| **Data Frame Format:** | |
|---|---|
| Sync1 | 'T' |
| Sync2 | 'M' |
| Class | 1Byte Data Frame Class |
| ID | 1Byte Data Frame ID |
| Length | Length of the Data(2 Bytes)<br>The number of bytes excluding the sync, class, ID, length, and checksum fields. |
| Data | The length of the data field varies depending on the Class and ID. |
| Checksum | 2Bytes CRC16 Checksum |

Frame Header Sync1, Sync2: Frame start synchronization words. All data frames begin with two synchronization characters 'T' and 'M'.

Class Field: 1 Byte. Defines the data frame type.
ID Field: 1 Byte. Defines the frame function ID.
Length Field: 2 Bytes. Defines the byte length of the Data field, excluding the frame synchronization words, Class field, ID field, and Checksum field: Two-byte unsigned integer, little-endian, low byte first, high byte last.

Data Field: Data portion, variable length, content determined by the Class and ID fields.
Checksum Field: Two-byte unsigned integer, little-endian, low byte first, high byte last. Calculated as follows:
Checksum Calculation:




The checksum is calculated using a 16-bit CRC checksum, which includes only the CLASS, ID, LENGTH, and DATA portions.

```
INT16U const CRC16Table[256] = {
    0x0000, 0xC0C1, 0xC181, 0x0140, 0xC301, 0x03C0, 0x0280, 0xC241,
    0xC601, 0x06C0, 0x0780, 0xC741, 0x0500, 0xC5C1, 0xC481, 0x0440,
    0xCC01, 0x0CC0, 0x0D80, 0xCD41, 0x0F00, 0xCFC1, 0xCE81, 0x0E40,
    0x0A00, 0xCAC1, 0xCB81, 0x0B40, 0xC901, 0x09C0, 0x0880, 0xC841,
    0xD801, 0x18C0, 0x1980, 0xD941, 0x1B00, 0xDBC1, 0xDA81, 0x1A40,
    0x1E00, 0xDEC1, 0xDF81, 0x1F40, 0xDD01, 0x1DC0, 0x1C80, 0xDC41,
    0x1400, 0xD4C1, 0xD581, 0x1540, 0xD701, 0x17C0, 0x1680, 0xD641,
    0xD201, 0x12C0, 0x1380, 0xD341, 0x1100, 0xD1C1, 0xD081, 0x1040,
    0xF001, 0x30C0, 0x3180, 0xF141, 0x3300, 0xF3C1, 0xF281, 0x3240,
    0x3600, 0xF6C1, 0xF781, 0x3740, 0xF501, 0x35C0, 0x3480, 0xF441,
    0x3C00, 0xFCC1, 0xFD81, 0x3D40, 0xFF01, 0x3FC0, 0x3E80, 0xFE41,
    0xFA01, 0x3AC0, 0x3B80, 0xFB41, 0x3900, 0xF9C1, 0xF881, 0x3840,
    0x2800, 0xE8C1, 0xE981, 0x2940, 0xEB01, 0x2BC0, 0x2A80, 0xEA41,
    0xEE01, 0x2EC0, 0x2F80, 0xEF41, 0x2D00, 0xEDC1, 0xEC81, 0x2C40,
    0xE401, 0x24C0, 0x2580, 0xE541, 0x2700, 0xE7C1, 0xE681, 0x2640,
    0x2200, 0xE2C1, 0xE381, 0x2340, 0xE101, 0x21C0, 0x2080, 0xE041,
    0xA001, 0x60C0, 0x6180, 0xA141, 0x6300, 0xA3C1, 0xA281, 0x6240,
    0x6600, 0xA6C1, 0xA781, 0x6740, 0xA501, 0x65C0, 0x6480, 0xA441,
    0x6C00, 0xACC1, 0xAD81, 0x6D40, 0xAF01, 0x6FC0, 0x6E80, 0xAE41,
    0xAA01, 0x6AC0, 0x6B80, 0xAB41, 0x6900, 0xA9C1, 0xA881, 0x6840,
    0x7800, 0xB8C1, 0xB981, 0x7940, 0xBB01, 0x7BC0, 0x7A80, 0xBA41,
    0xBE01, 0x7EC0, 0x7F80, 0xBF41, 0x7D00, 0xBDC1, 0xBC81, 0x7C40,
```

```
        0xB401, 0x74C0, 0x7580, 0xB541, 0x7700, 0xB7C1, 0xB681, 0x7640,
        0x7200, 0xB2C1, 0xB381, 0x7340, 0xB101, 0x71C0, 0x7080, 0xB041,
        0x5000, 0x90C1, 0x9181, 0x5140, 0x9301, 0x53C0, 0x5280, 0x9241,
        0x9601, 0x56C0, 0x5780, 0x9741, 0x5500, 0x95C1, 0x9481, 0x5440,
        0x9C01, 0x5CC0, 0x5D80, 0x9D41, 0x5F00, 0x9FC1, 0x9E81, 0x5E40,
        0x5A00, 0x9AC1, 0x9B81, 0x5B40, 0x9901, 0x59C0, 0x5880, 0x9841,
        0x8801, 0x48C0, 0x4980, 0x8941, 0x4B00, 0x8BC1, 0x8A81, 0x4A40,
        0x4E00, 0x8EC1, 0x8F81, 0x4F40, 0x8D01, 0x4DC0, 0x4C80, 0x8C41,
        0x4400, 0x84C1, 0x8581, 0x4540, 0x8701, 0x47C0, 0x4680, 0x8641,
        0x8201, 0x42C0, 0x4380, 0x8341, 0x4100, 0x81C1, 0x8081, 0x4040
    };

INT16U CRC16(INT8U *p, INT16U length)
{
      INT16U checksum = 0;
      for( ; length > 0; length-- )
      {
      checksum = ( checksum >> 8 ) ^ CRC16Table[ (checksum&0xFF) ^ *p ];
      p++;
      }
      return checksum;
}
```

Set the communication baud rate(**0x0c 0x02**):

Host sends:

| Message | Set the communication baud rate | | | | |
|---|---|---|---|---|---|
| Description | Set the communication baud rate | | | | |
| Transmission direction | Command | | | | |
| Message Structure | Header 'T''M' | Class,ID0x0C, 0x02 | Length(1) | Data (Length)Bytes | Checksum |
| Data Contents: | | | | | |
| Byte Offset | No. Format | | Name | Unit | Comment |
| 0 | INT8U | - | Baud Rate | - | 3: 9600<br>4: 14400<br>5: 19200<br>6: 28800<br>7: 38400<br>8: 56000<br>9: 57600<br>**10:115200** |

Slave Response:

| | | | | | |
|---|---|---|---|---|---|
| Message | Set the communication baud rate | | | | |
| Description | Set the communication baud rate | | | | |
| Transmission direction | Response | | | | |
| Message Structure | Header ‘T’‘M’ | Class,ID0x0c, 0x02 | Length(0) | Data (Length)Bytes | Checksum |

Example: Baud Rate set to 115200:

TX: 54 4D 0C 02 01 00 0A C0 7E

RX: 54 4D 0C 02 00 00 XX XX

Set data output frequency**(0x0f 0x15)**:

Host sends:

| | | | | | |
|---|---|---|---|---|---|
| Message | Set data output frequency | | | | |
| Description | Set data output frequency | | | | |
| Transmission direction | Command | | | | |
| Message Structure | Header ‘T’‘M’ | Class,ID0x0f, 0x15 | Length(1) | Data (Length)Bytes | Checksum |
| Data Contents: | | | | | |
| Byte Offset | No. Format | Scaling | Name | Unit | Comment |
| 0 | INT8U | - | Data Output Frequency | - | 0: 1HZ<br>1: 5HZ<br>2: 10HZ<br>3: 25HZ<br>4: 50HZ<br>**5: 100HZ** |

Slave Response:

| | | | | | |
|---|---|---|---|---|---|
| Message | Data output frequency | | | | |
| Description | Data output frequency | | | | |
| Transmission direction | B Response | | | | |
| Message Structure | Header ‘T’ ‘M’ | Class,ID0x0f, 0x15 | Length(0) | Data (Length)Bytes | Checksum |

Set data output type**(0x0f 0x25)**:

Hose sends:

| | | | | | |
|---|---|---|---|---|---|
| Message | Output data type | | | | |
| Description | Output data type | | | | |
| Transmission direction | Command | | | | |
| Message | Header | Class,ID0x0f, | Length | Data | Checksum |

| Structure | 'T' 'M' | 0x25 | (1) | (Length)Bytes | |
|---|---|---|---|---|---|
| Data Contents: | | | | | |
| Byte Offset | Number Format | Scaling | Name | Unit | Comment |
| 0 | INT8U | - | Data output mode | - | **0: Euler angles + 9-axis data**<br>1: Euler angles + 9-axis data (volume acceleration)<br>2: Euler angles |

Slave Response:

| Message | Output data type | | | | |
|---|---|---|---|---|---|
| Description | Output data type | | | | |
| Transmission direction | B Response | | | | |
| Message Structure | Header 'T' 'M' | Class,ID0x0f,0x25 | Length(0) | Data (Length )Bytes | Checksum |

0: Euler Angles + 9-Axis Data: Normal operating mode; data output is Euler angles and sensor values.

1: Euler Angles + 9-Axis Data (Volume Acceleration): Data output is Euler angles and linear acceleration after removing gravity.

Other settings are the same as **0: Euler Angles + 9-Axis Data** mode.

2: Euler Angles: Compared to the **1: Euler Angles + 9-axis Data (Volume Acceleration)** mode, the payload only contains Euler angles, and the sensor values are not included in the data segment. It is suitable for situations where the baud rate is low but a relatively high frequency data packet output is required.

Set running mode**(0x0f 0x24)**:

Host sends:

| Message | Operating mode | | | | |
|---|---|---|---|---|---|
| Description | Operating mode | | | | |
| Transmission direction | Command | | | | |
| Message Structure | Header 'T' 'M' | Class,ID0x0f, 0x24 | Length(1) | Data (Length)Bytes | Checksum |
| Data Contents: | | | | | |
| Byte Offset | No. Format | Scaling | Name | Unit | Comment |
| 0 | INT8U | - | Running mode | - | **1:6 Axis**<br>2:9 Axis |

Slave Response:

| Message | Operating mode | | | | |
|---|---|---|---|---|---|
| Description | Operating mode | | | | |
| Transmission direction | B Response | | | | |
| Message Structure | Header ‘T’ ‘M’ | Class,ID0x0f, 0x24 | Length(0) | Data (Length)Bytes | Checksum |

Magnetic orientation level calibration begins**(0x0f, 0x16)**:

Host sends:

| Message | Horizontal calibration begins | | | | |
|---|---|---|---|---|---|
| Description | Horizontal calibration begins | | | | |
| Transmission direction | B Response | | | | |
| Message Structure | Header ‘T’ ‘M’ | Class,ID0x0f, 0x16 | Length(0) | Data (Length)Bytes | Checksum |

Slave Response:

| Message | Horizontal calibration begins | | | | |
|---|---|---|---|---|---|
| Description | Horizontal calibration begins | | | | |
| Transmission direction | B Response | | | | |
| Message Structure | Header ‘T’ ‘M’ | Class,ID 0x0f, 0x16 | Length(0) | Data (Length)Bytes | Checksum |

Magnetic orientation level calibration stopped**(0x0f, 0x17)**:

Host sends:

| Message | Horizontal calibration stopped | | | | |
|---|---|---|---|---|---|
| Description | Horizontal calibration stopped | | | | |
| Transmission direction | B Response | | | | |
| Message Structure | Header ‘T’ ‘M’ | Class,ID0x0f, 0x17 | Length(0) | Data (Length)Bytes | Checksum |

Slave Response:

| Message | Horizontal calibration stopped | | | | |
|---|---|---|---|---|---|
| Description | Horizontal calibration stopped | | | | |
| Transmission direction | B Response | | | | |
| Message Structure | Header ‘T’ ‘M’ | Class,ID0x0f, 0x17 | Length(0) | Data (Length)Bytes | Checksum |

Magnetic orientation vertical upward calibration begins**(0x0f, 0x0B)**:

Host sends:

| | | | | | |
|---|---|---|---|---|---|
| Message | Vertical upward calibration starts | | | | |
| Description | Vertical upward calibration starts | | | | |
| Transmission direction | B Response | | | | |
| Message Structure | Header ‘T’ ‘M’ | Class,ID0x0f, 0x0B | Length(0) | Data (Length)Bytes | Checksum |

Slave Response:

| | | | | | |
|---|---|---|---|---|---|
| Message | Vertical upward calibration starts | | | | |
| Description | Vertical upward calibration starts | | | | |
| Transmission direction | B Response | | | | |
| Message Structure | Header ‘T’ ‘M’ | Class,ID0x0f, 0x0B | Length(0) | Data (Length)Bytes | Checksum |

Magnetic orientation vertical upward calibration stop**(0x0f, 0x0C)**:

Host sends:

| | | | | | |
|---|---|---|---|---|---|
| Message | Vertical upward calibration stop | | | | |
| Description | Vertical upward calibration stop | | | | |
| Transmission direction | B Response | | | | |
| Message Structure | Header ‘T’ ‘M’ | Class,ID0x0f, 0x0C | Length(0) | Data (Length)Bytes | Checksum |

Slave Response:

| | | | | | |
|---|---|---|---|---|---|
| Message | Vertical upward calibration stop | | | | |
| Description | Vertical upward calibration stop | | | | |
| Transmission direction | B Response | | | | |
| Message Structure | Header ‘T’ ‘M’ | Class,ID0x0f, 0x0C | Length(0) | Data (Length)Bytes | Checksum |

Read the magnitude of magnetic declination**(0x0f, 0x0F)**:

Host sends:

| | | | | | |
|---|---|---|---|---|---|
| Message | Reading magnetic declination | | | | |
| Description | Reading magnetic declination | | | | |
| Transmission direction | B Response | | | | |
| Message Structure | Header ‘T’ ‘M’ | Class,ID0x0f, 0x0F | Length(0) | Data (Length)Bytes | Checksum |

Slave Response:

| Message | Reading magnetic declination | | | | |
|---|---|---|---|---|---|
| Description | Reading magnetic declination | | | | |
| Transmission direction | Command | | | | |
| Message Structure | Header 'T' 'M' | Class,ID0x0f, 0x0F | Length (0x0035) | Data (Length )Bytes | Checksum |
| Data Contents: | | | | | |
| Byte Offset | Number Format | Scaling | Name | Unit | Comment |
| 0 | INT8U | - | 0 | - | 0 |
| 1 | INT8U | - | 0 | - | 0 |
| … | | - | 0 | - | 0 |
| 48 | INT8U | - | 0 | - | 0 |
| 49 | FP32 | - | magnetic declination | - | Magseth, Little-endian storage |

Set magnetic declination**(0x0f, 0x10)**:

Host sends:

| Message | Set magnetic declination | | | | |
|---|---|---|---|---|---|
| Description | Set magnetic declination | | | | |
| Transmission direction | Command | | | | |
| Message Structure | Header 'T' 'M' | Class,ID0x0f, 0x10 | Length (0x0035) | Data (Length )Bytes | Checksum |
| Data Contents: | | | | | |
| Byte Offset | Number Format | Scaling | Name | Unit | Purpose/ Comment |
| 0 | INT8U | - | 0 | - | 0 |
| 1 | INT8U | - | 0 | - | 0 |
| … | | - | 0 | | 0 |
| 48 | INT8U | - | 0 | | 0 |
| 49 | FP32 | | magnetic declination | | Magseth, Little-endian storage |

Slave Response:

| Message | Set magnetic declination | | | | |
|---|---|---|---|---|---|
| Description | Set magnetic declination | | | | |
| Transmission direction | B Response | | | | |
| Message Structure | Header 'T' 'M' | Class,ID0x0f, 0x10 | Length(0) | Data (Length )Bytes | Checksum |

Clear magnetic declination**(0x0f, 0x11)**:

Host sends:

| Message | Clear magnetic declination | | | | |
|---|---|---|---|---|---|
| Description | Clear magnetic declination | | | | |
| Transmission direction | B Response | | | | |
| Message Structure | Header ‘T’ ‘M’ | Class,ID0x0f, 0x11 | Length(0) | Data (Length )Bytes | Checksum |

Slave Response:

| Message | Clear magnetic declination | | | | |
|---|---|---|---|---|---|
| Description | Clear magnetic declination | | | | |
| Transmission direction | B Response | | | | |
| Message Structure | Header ‘T’ ‘M’ | Class,ID0x0f, 0x11 | Length(0) | Data (Length )Bytes | Checksum |

**IMU Data Output**

**Configure via data output mode.**

Standard attitude and sensor output**(0x0f, 0x01):**

| Message | Attitude output | | | | |
|---|---|---|---|---|---|
| Description | Standard attitude and sensor output | | | | |
| Transmission direction | Output only | | | | |
| Message Structure | Header ‘T’ ‘M’ | Class,ID0x0f, 0x01 | Length (50) | Payload (Length )Bytes | Checksum |
| Payload Contents: | | | | | |
| Byte Offset | Number Format | Scaling | Name | Unit | Purpose/Comment |
| 0 | INT8U | - | Flags | - | Reserved |
| 1 | FP32 | - | Roll | degree | +/-180 Roll |
| 5 | FP32 | - | Pitch | degree | +/-90 Pitch |
| 9 | FP32 | - | Yaw | degree | +/-180 Heading |
| 13 | FP32 | - | Gx | deg/s | Gyro x |
| 17 | FP32 | - | Gy | deg/s | Gyro y |
| 21 | FP32 | - | Gz | deg/s | Gyro z |
| 25 | FP32 | - | Ax | g | Acceleration x |
| 29 | FP32 | - | Ay | g | Acceleration y |
| 33 | FP32 | - | Az | g | Acceleration z |
| 37 | FP32 | - | Mx | uT | Mag x |
| 41 | FP32 | - | My | uT | Mag y |
| 45 | FP32 | - | Mz | uT | Mag z |
| 49 | INT8U | | SOC | % | Battery percentage |

**Note: Floating-point numbers are stored in little-endian mode, with the least significant byte first and the most significant byte last.**

Attitude output**(0x0f, 0x01):**

| Message | Attitude output | | | | |
|---|---|---|---|---|---|
| Description | Attitude output | | | | |
| Transmission direction | Output only | | | | |
| Message Structure | Header 'T' 'M' | Class,ID0x0f ,0x01 | Length (14) | Payload (Length )Bytes | Checksum |
| Payload Contents: | | | | | |
| Byte Offset | Number Format | Scaling | Name | Unit | Purpose/Comment |
| 0 | INT8U | - | Flags | - | Reserved |
| 1 | FP32 | - | Roll | degree | +/-180 Roll |
| 5 | FP32 | - | Pitch | degree | +/-90 Pitch |
| 9 | FP32 | - | Yaw | degree | +/-180 Heading |
| 13 | INT8U | | SOC | % | Battery percentage |