GMU200-BL Bluetooth attitude sensor

Product manual

V1.2-20260526

Technical Highlights

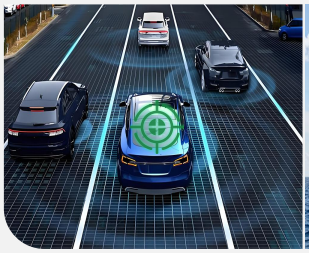
- ❖ Integrated nine-axis sensor and temperature sensor;
- ❖ Bluetooth 5.1 wireless transmission;
- ❖ Supports VRU and AHRS;
- ❖ Real-time 3D dynamic measurement;
- ❖ Compact structure, easy installation.

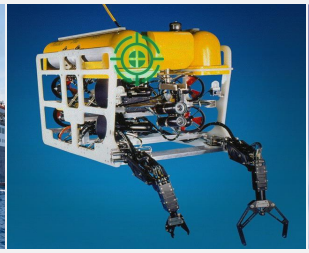
Product Introduction

The GMU200-BL is a cost-effective miniature attitude reference system (AHRS) suitable for equipment requiring high dynamic balance capabilities, such as model aircraft, robots, agricultural machinery, engineering vehicles, and forklifts. This device collects sensor data, integrates a Kalman filter algorithm, and outputs attitude information in real time. It employs a three-axis accelerometer and a three-axis magnetometer to assist the three-axis gyroscope, and integrates temperature compensation technology, giving the product excellent stability and real-time performance across the entire attitude range, making it applicable to various motion attitude monitoring scenarios. In addition to outputting attitude data, this product provides nine-axis data, including three-axis acceleration, three-axis gyroscope, and three-axis magnetic field data. It also supports modification and expansion of system operating parameters to adapt to different application platforms. This product uses Bluetooth wireless output, equipped with a Bluetooth 5.1 wireless transmission module and BLE protocol, and integrates 2.4GHz adaptive frequency hopping technology, enabling line-of-sight transmission up to 40 meters in unobstructed environments. It also supports Bluetooth wireless firmware upgrades via BLE DFU, effectively improving the flexibility of industrial deployment.

Application Scope

- ❖ Medical rehabilitation equipment monitoring
- ❖ Limb rehabilitation monitoring
- ❖ Human body capture
- ❖ Posture angle monitoring






Performance parameters

GMU200-BL	Parameter
Measurement Range	Roll ±180°, Pitch ±90°, Azimuth ±180° (In VRU mode, the initial value after Bluetooth disconnection and reconnection is 0°; in AHRS mode, the output is the magnetic heading angle)
Measurement Axis	X Axis / Y Axis / Z Axis
Roll and Pitch Static Accuracy [1)]	±0.1° (@25℃)
Roll and Pitch Dynamic Accuracy (rms)	±0.25° (@25℃)
Magnetic Heading Accuracy (AHRS)	±1.5° (@25℃)
Gyroscope	
Gyroscope Range (Adjustable)	±125°/s, ±250°/s, ±500°/s, ±1000°/s, ±2000°/s
Zero-Bias Stability (10s Smoothing)	10°/h
Zero-Bias Instabilit(Allan)	5°/h
Angle Random Walk Coefficient(Allan)	0.25°/sqrt(h)
Accelerometer	
Acceleration Range (Settable)	±2g, ±4g, ±8g, ±16g
Bias Stability (10s Mean)	0.1mg
Bias Instability(Allan)	0.05mg
Velocity Random Walk Coefficient (Allan)	0.005m/s/sqrt(h)
Magnetometer	
Magnetometer range	±8Gauss
Resolution	<2.5mgauss
Bandwidth	<100Hz
Other	
Zero-point temperature coefficient [2)]	±0.01°/℃ (@-20~65℃)
Sensitivity temperature coefficient [3)]	≤100ppm/℃ (@-20~65℃)
Power-on time	≤3.5S
Response time	0.01S
Electromagnetic compatibility	According to EN61000 and GBT17626
Receiver sensitivity	-96dBm
Transmit power	8dBm
Bluetooth transmission distance	40m (in an open, unobstructed area)
Usage time (fully charged battery)	Continuous operation for 6 hours, standby time for 83 days
Insulation resistance	≥100 megohms
Shock resistance	100g@11ms, triaxial (half-sine wave)
Vibration resistance	10grms , 10~1000Hz
Casing material	ABS plastic housing
Weight	≤10g

1) Accuracy: Refers to the root mean square difference between the measured value and the actual angle error after multiple angle measurements (>16 times) under normal temperature conditions.
2) Zero-point temperature coefficient: Refers to the rate of change of the sensor's reading relative to normal temperature within its rated operating temperature range when the sensor is at zero.
3) Sensitivity temperature coefficient: Refers to the rate of change of the sensor's full-scale reading within its rated operating temperature range relative to its full-scale reading at normal temperature as a percentage of temperature.

Battery parameters

Item	Parameter
Nominal Capacity	100mAh
Nominal Voltage	3.7V
Charging Cut-off Voltage	4.20

Electrical Parameters

Parameters	Condition	MIN	TYP	MAX	Unit
Charging Voltage (DC)	Standard	4.5V	5V	5.5	V
Charging Current			0.1		A
Charging Temperature	0~+55℃ (Note: Charging temperatures that are too high or too low can damage the battery)				
Charging Interface	Type-C interface				
Charging Time	2.5 hours(25℃)				
Operating Current			18		mA
Power Off and Sleep Current			50		uA
Operating Temp		-20		+70	℃
Storage Temp		-20		+70	℃

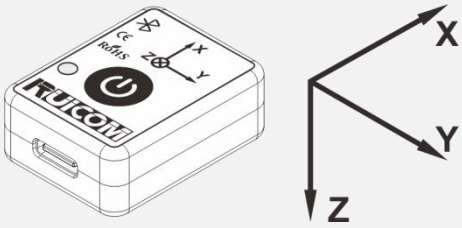
Ordering and Selecting

Model		Installation Direction		Application Scenarios
GMU200-BL	-	H: Horizontal installation	-	OB: Marine Applications
		HD: Horizontal downwards		EV: Construction Excavators
		V: Vertical installation		EGP: Construction Pile Drivers
		VL: Vertical leftwards		V: Vehicle-Mounted Applications
		VR: Vertical rightwards		MTA: Mining Truck Applications
				TC: Tower Cranes

E.g: GMU200-BL-H-OB: Horizontal installation / Marine Applications

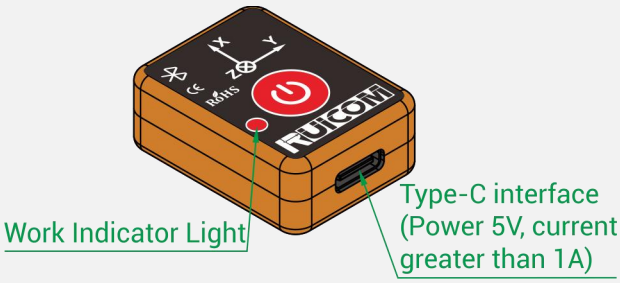
Measurement Directions

Proper sensor installation can avoid measurement errors. The key points are: the mounting surface and the surface being measured must be flat, close, and firmly fixed. An uneven base surface will cause angular deviations; the sensor axis must be parallel to the axis of the measured object, with no included angle.



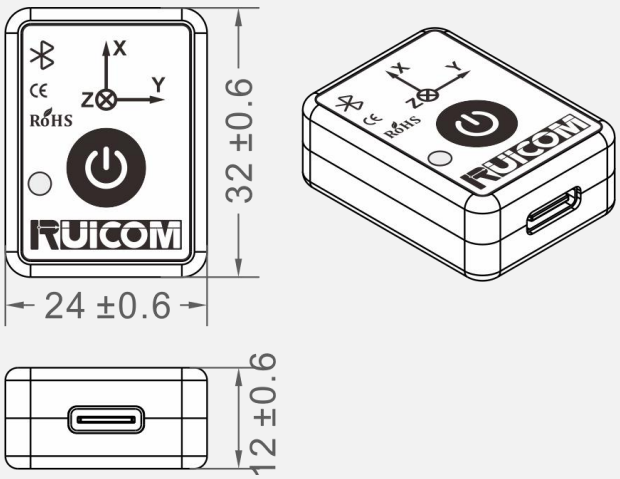
Instructions for use of indicator lights

① Power On: Press and hold for 2 seconds; the green light will flash for 2 seconds.
② Power Off: Press and hold for 2 seconds; the green light will remain on for 2 seconds.
③ After connecting to the host, the green light flashes once every 5 seconds.
④ While charging: The red light remains on.



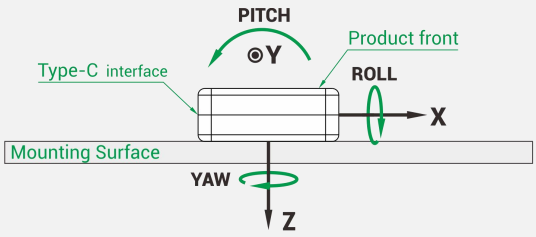
After powering on, the sensor remains in low-power Bluetooth broadcast mode, waiting for the host to connect. The host's Bluetooth can be turned on to scan and discover the slave sensor and connect. If the host connection is successful, the slave sensor will transmit sensor data with the host. During this time, the green light will flash once every 5 seconds. If a connection timeout event occurs, the Bluetooth connection will be lost, and the sensor will return to low-power Bluetooth broadcast mode, waiting for the host to connect.

Shell Size

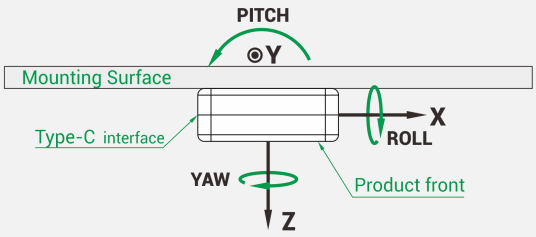


Shell Size: L32×W24×H12mm

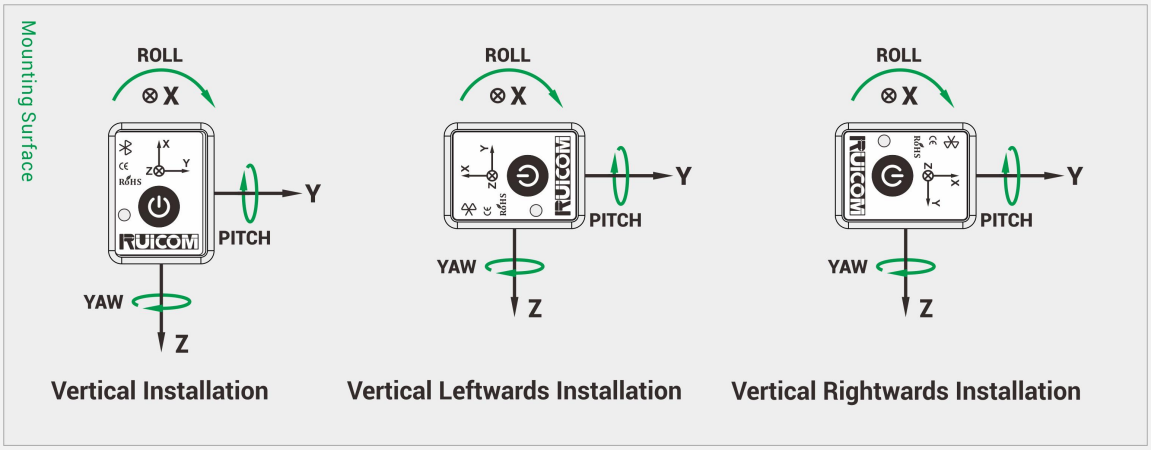
Installation Direction



Horizontal Installation



Horizontal Downward Installation



Vertical Installation | Vertical Leftwards Installation | Vertical Rightwards Installation

Calibration Method

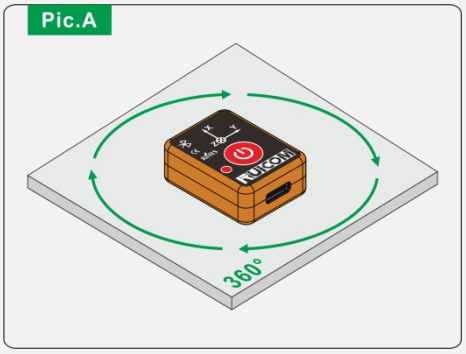
Calibration prerequisites:
1) The sensor accuracy does not meet the requirements;
2) The sensor installation environment has magnetic field interference. This interference is constant. Ensure that the distance between the sensor and the interfering magnetic field does not change after installation (e.g., if the sensor is installed on a carrier (iron) with magnetic interference, the sensor and carrier cannot be separated. Fix it and rotate it for calibration. If separation occurs due to lack of fixation, recalibration is required. If the distance between the sensor and the mounting carrier cannot be fixed, choose an installation location as far away from the interference source as possible, with a safe distance of at least 30cm).

Horizontal installation calibration mode:

1)Rotate the plane more than one full circle (360°): As shown in Pic A

Horizontal Calibration Start Command
TX: 54 4D 0F 16 00 00 E2 D0
RX: 54 4D 0F 16 00 00 E2 D0
Data Field (0 bytes)
The value will only be valid after rotating horizontally at a constant speed for more than 1 revolution, ensuring the time is more than 20 seconds. Values saved in less than 20 seconds

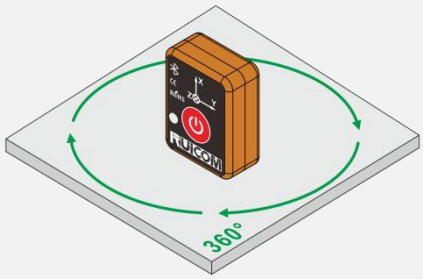
Pic.A



will be invalid.
Horizontal Calibration Stop Command
TX: 54 4D 0F 17 00 00 B3 10
RX: 54 4D 0F 17 00 00 B3 10
Data Field (0 bytes)
Command without data field

2)Rotate vertically upwards more than one full circle (360°): as shown in Pic B
Start calibration command vertically upward
TX: 54 4D 0F 0B 00 00 72 D6
RX: 54 4D 0F 0B 00 00 72 D6
Data Field (0 bytes)
Rotate vertically at a constant speed for at least one revolution, ensuring the time is at least 20 seconds. Only then will the saved value be valid; values saved in less than 20 seconds will be invalid.
Vertical Upward Calibration Stop Command
TX: 54 4D 0F 0C 00 00 C3 17
RX: 54 4D 0F 0C 00 00 C3 17
Data Field (0 bytes)
No Data Field Command

Pic.B



BLE Protocol

This protocol applies to communication between a host and a slave device based on Bluetooth Low Energy (BLE) technology. It is mainly used for data acquisition, data transmission, and remote control between the host and the Bluetooth slave device (GMU200-BL, hereinafter referred to as the slave device).
BLE Protocol Basics: Based on the GATT (General Attribute Protocol) model, the slave device acts as the GATT server, and the master device (mobile APP or PC) acts as the GATT client.
GATT (General Attribute Protocol): The core protocol for BLE data transmission, based on a client-server model.
Service: Defines device functions (e.g., RX, TX services).
Characteristic: Specific data points within a service (e.g., tilt angle, acceleration, gyro angular velocity, battery level, etc.).
Descriptor: Additional information configuring the characteristic (e.g., notification enablement).
Communication model
Role Definitions:
Master (Central): Initiates connection requests, reads or writes data to the slave.
Slave (Peripheral): Broadcasts its own services, initiates connection and data transfer requests to the master, and responds to the master's read and write requests.
Communication mode: Supports master-slave bidirectional communication mode.
Scenario 1: Bluetooth slave devices periodically broadcast data to the host (the timing

period can be adjusted remotely by the host);
Scenario 2: The host can initiate communication requests to Bluetooth peripherals, read and write data to the peripherals, such as modifying peripheral parameters.

Terms and Definitions

GATT: Generic Attribute Protocol, used for BLE data transfer.
UUID: Universally Unique Identifier, used to identify services and characteristics.

Data format

Service and eigenvalue definition

The slave device provides the following GATT services and characteristics:
Client UUID type: 0x02(BLE_UUID_TYPE_VENDOR_BEGIN)
Base UUID: 0x00000000-32F7-11F1-8000-CD55B35033B3
Main service UUID: 0x6801 (tilt sensor)
Characteristic 1: Data reception (UUID: 0x6802, writable, write without response)
Characteristic 2: Data transmission (UUID: 0x6803, readable, notification)

OTA DFU Firmware Upgrade Service (refer to Nordic's official website for details):
Customer UUID type: 0x01 (BLE_UUID_TYPE_BLE)
Basic UUID: 0x8EC90000-F315-4F60-9FB8-838830DAEA50
Main Service UUID: 0xFE59
Feature Value: BLE_DFU_BUTTONLESS_CHAR_UUID (0x0003)
Device Name: GMU200-BL-XXXXXXXX
GMU200-BL: Represents the product model;
XXXXXXXX: Represents the product identification number (e.g., GMU200-BL-25090001);

Bluetooth BLE data protocol

Data Frame Format:	
Sync1	'T'
Sync2	'M'
Class	1Byte Data Frame Class
ID	1Byte Data Frame ID
Length	Length of the Data(2 Bytes) Not including sync,class,ID,length and Checksum Number of bytes in the field
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 type of data frame.
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. The Class, ID, and Checksum fields are: two-byte unsigned integers, little-endian, with the least significant byte first (LCM).
Data Field: The data portion, variable length, determined by the Class and ID fields.
Checksum Field: Two-byte unsigned integers, little-endian, with the least significant byte

first (LCM). Calculation method:

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;
}
```

List of Class and ID function codes:

Class	ID	Data Length	Data Type	Function Description	
0x0F	0x03	0x0001	Uint8	Set Installation Measurement Direction	**00: Horizontal (default);** 01: Vertical upward; 02: Lateral to the left; 03: Lateral to the right; 04: Horizontal reverse; E.g: Setting the installation direction to horizontal. TX: 54 4D 0F 03 01 00 00 05 85 RX: 54 4D 0F 03 00 00 F3 14
0x0F	0x83	0x0000	NULL	Query Current Installation Measurement Direction	E.g: Read current installation measurement direction. TX: 54 4D 0F 83 00 00 F2 FC RX: 54 4D 0F 83 01 00 00 2C 45 The current installation direction is horizontal
0x0F	0x04	0x0001	Uint8	Set Output Data Item MAP	OUTMAP_EULER: 0x01 represents Euler angles; OUTMAP_ACCEL: 0x02 represents acceleration; OUTMAP_GYRO: 0x04 represents gyro angular velocity; OUTMAP_MAG: 0x08 represents magnetometer; OUTMAP_TC: 0x10 represents temperature; OUTMAP_BAT: 0x20 represents battery level; OUTMAP_STS: 0x40 represents IMU status; OUTMAP_TPS: 0x80 represents time; 0x07: represents Euler angles, accelerometer readings, and gyro angular velocity; **(Default is 0xFF)** E.g: Setting output data item MAP=0xFF: TX: 54 4D 0F 04 01 00 FF 44 B1 RX: 54 4D 0F 04 00 00 42 D5
0x0F	0x84	0x0000	NULL	Query Output Data Item MAP	E.g: Read the current data item and output the MAP. TX: 54 4D 0F 84 00 00 43 3D RX: 54 4D 0F 84 01 00 FF 6D 71 Output data item MAP=0xFF;
0x0F	0x05	0x0001	Uint8	Set Current	**0x00: Linear Acceleration (Default)**

				Acceleration Output Type	0x01: Volume Acceleration E.g: Sets the acceleration output type to volume acceleration TX: 54 4D 0F 05 01 00 01 C4 CD RX: 54 4D 0F 05 00 00 13 15
0x0F	0x85	0x0000	NULL	Query Current Acceleration Output Type	E.g: Read the current acceleration output type. TX: 54 4D 0F 85 00 00 12 FD RX: 54 4D 0F 85 01 00 01 ED 0D The acceleration output type is volume acceleration;
0x0F	0x06	0x0001	Uint8	Manual Gyro Zero Bias Clear	0x00: Manual gyroscope zero-bias clearing stopped; 0x01: Manual gyroscope zero-bias clearing started; Routine: Manual gyroscope zero-bias clearing started. TX: 54 4D 0F 06 01 00 01 C4 89 RX: 54 4D 0F 06 00 00 E3 15
0x0F	0x86	0x0000	NULL	Query Manual Gyro Zero Bias Clear Status	Response: 0x00: Manual gyroscope zero-bias clearing has stopped; 0x01: Manual gyroscope zero-bias clearing is in progress; Routine: Query the status of manual gyroscope zero-bias clearing TX: 54 4D 0F 86 00 00 E2 FD RX: 54 4D 0F 86 01 00 00 2C 89 Manual gyroscope zero-bias clearing has stopped
0x0F	0x07	0x0001	0x0A	Save Gyro Zero Bias	E.g: Save manual gyroscope zero bias TX: 54 4D 0F 07 01 00 0A 84 B2 RX: 54 4D 0F 07 00 00 B2 D5
0x0F	0x15	0x0001	Uint8	Set Data Output Frequency	0x00: Stop output; 0x01: 1Hz; 0x02: 5Hz; 0x03: 10Hz; 0x04: 20Hz; **0x05: 25Hz(Default);** 0x06: 50Hz; 0x07: 100Hz; E.g: Set the output frequency to 5Hz TX: 54 4D 0F 15 01 00 02 80 0C RX: 54 4D 0F 15 00 00 12 D0
0x0F	0x95	0x0000	NULL	Query Data	E.g: Query output frequency

				Output Frequency	TX: 54 4D 0F 95 00 00 13 38 RX: 54 4D 0F 15 01 00 02 A9 CC Output frequency is 5Hz
0x0F	0x16	0x0000	NULL	Start Horizontal Rotation Calibration	Routine: Horizontal rotation calibration begins TX: 54 4D 0F 16 00 00 E2 D0 RX: 54 4D 0F 16 00 00 E2 D0
0x0F	0x17	0x0000	NULL	Stop Horizontal Rotation Calibration	Routine: Horizontal rotation calibration stop TX: 54 4D 0F 17 00 00 B3 10 RX: 54 4D 0F 17 00 00 B3 10
0x0F	0x0B	0x0000	NULL	Start Vertical Rotation Calibration	Routine: Vertical rotation calibration begins TX: 54 4D 0F 0B 00 00 72 D6 RX: 54 4D 0F 0B 00 00 72 D6
0x0F	0x0C	0x0000	NULL	Stop Vertical Rotation Calibration	Routine: Vertical rotation calibration stop TX: 54 4D 0F 0C 00 00 C3 17 RX: 54 4D 0F 0C 00 00 C3 17
0x0F	0x0F	0x0004	Ft32	Set Magnetic Declination	E.g: Set the magnetic declination to 3.5 degrees: TX: 54 4D 0F 0F 04 00 00 00 60 40 97 34 RX: 54 4D 0F 0F 00 00 32 FF
0x0F	0x8F	0x0000	NULL	Query Magnetic Declination	E.g: Query magnetic declination: TX: 54 4D 0F 8F 00 00 32 FF RX: 54 4D 0F 8F 04 00 00 00 60 40 16 FC Magnetic declination is 3.5 degrees
0x0F	0x24	0x0001	Uint8	Set Operating Mode	0x00: IMU; **0x01:VRU(default - heading angle is gyro heading, 0 upon power-on);** 0x02:AHRS(Heading angle is magnetic heading); E.g: Set operating mode to AHRS; TX: 54 4D 0F 24 01 00 02 8E F0 RX: 54 4D 0F 24 00 00 43 1F
0x0F	0xA4	0x0000	NULL	Query Current Operating Mode	E.g: Query execution mode: TX: 54 4D 0F A4 00 00 42 F7 RX: 54 4D 0F A4 01 00 02 A7 30 Execution mode is AHRS;
0x0F	0x1A	0x0000	NULL	Software Reset	For use with new configurations, such as range, operating mode, etc. Routine: Perform a software reset; TX: 54 4D 0F 1A 00 00 22 D3 RX: 54 4D 0F 1A 00 00 22 D3 Reconnection is required after a software reset
0x0F	0x1D	0x0004	Uint32	Set Product ID Number	Bluetooth device name: GMU200-BL-XXXXXXXX in XXXXXXX E.g: Set PRODUCT ID to

					0x78C2EAC1(2026040001) TX: 54 4D 0F 1D 04 00 C1 EA C2 78 C1 8F RX: 54 4D 0F 1D 00 00 93 12
0x0F	0x9D	0x0000	NULL	Query Product ID Number	E.g: Query PRODUCT ID: TX: 54 4D 0F 9D 00 00 92 FA RX: 54 4D 0F 9D 04 00 C1 EA C2 78 40 47 Product ID is 0x78C2EAC1(2026040001)
0x0F	0x11	0x0001	Uint8	Set Zero Point	**0x00: Sets the current point to absolute zero (default);** 0x01: Sets the current point to relative zero; Note: Used for small-angle installation error correction E.g: Setting relative zero: TX: 54 4D 0F 11 01 00 01 C1 3D RX: 54 4D 0F 11 00 00 53 15
0x0F	0x91	0x0000	NULL	Query Current Zero Point Type	E.g: Query the current zero point type: TX: 54 4D 0F 91 00 00 52 F9 RX: 54 4D 0F 01 01 00 01 E8 FD Currently a relative zero point
0x0F	0x28	0x0000	NULL	Reset Current Heading Angle	This only applies to IMU and VRU modes; E.g: Zero the gyro heading angle: TX: 54 4D 0F 28 00 00 83 1C RX: 54 4D 0F 28 00 00 83 1C After execution, the current heading angle is 0.
0x0D	0x12	0x0001	Uint8	Set Startup Mode	**0x00: 0-second startup mode (default);** 0x01: 5-second static start mode; E.g: Setting a 5-second static start mode: TX: 54 4D 0D 12 01 00 01 B8 B9 RX: 54 4D 0D 12 00 00 A2 A9 Note: The 5-second static start mode is mainly for applications that require high heading accuracy and very small gyroscope bias (such as AGVs). After the IMU is connected and started, it must remain still for 5 seconds to perform one gyroscope bias removal operation at a time.
0x0D	0x92	0x0000	NULL	Query Startup Mode	E.g: Query startup mode: TX: 54 4D 0D 92 00 00 A3 41 RX: 54 4D 0D 92 01 00 01 91 79 Current mode: 5-second static start mode
0x0D	0x13	0x0001	Uint8	Set Acceleration Range	0x00: 2g; **0x01: 4g(Default);** 0x02: 8g; 0x03: 16g;

					E.g: Set the acceleration range to 8g: TX: 54 4D 0D 13 01 00 02 F9 44 RX: 54 4D 0D 13 00 00 F3 69
0x0D	0x93	0x0000	NULL	Query Acceleration Range	E.g: Query acceleration range: TX: 54 4D 0D 93 00 00 F2 81 RX: 54 4D 0D 93 01 00 02 D0 84 Acceleration range: 8g
0x0D	0x14	0x0001	Uint8	Set Angular Velocity Range	0x00: 125deg/sec; 0x01: 250deg/sec; **0x02: 500deg/sec(Default);** 0x03: 1000deg/sec; 0x04: 2000deg/sec; E.g: Set the angular velocity range to 2000 dec/sec: TX: 54 4D 0D 14 01 00 04 78 32 RX: 54 4D 0D 14 00 00 42 AB
0x0D	0x94	0x0000	NULL	Query Angular Velocity Range	E.g: Query angular velocity range: TX: 54 4D 0D 94 00 00 43 40 RX: 54 4D 0D 94 01 00 04 51 F2 Angular velocity range: 2000deg/sec
0x0D	0x15	0x0001	Uint8	Set 3dB Bandwidth	0x00: 25Hz; 0x01: 50Hz; **0x02: 99Hz(Default);** 0x03: 249Hz; E.g: Setting the 3dB bandwidth to 249Hz: TX: 54 4D 0D 15 01 00 03 38 0C RX: 54 4D 0D 15 00 00 13 68
0x0D	0x95	0x0000	NULL	Query 3dB Bandwidth	E.g: Query 3dB bandwidth: TX: 54 4D 0D 95 00 00 12 80 RX: 54 4D 0D 95 01 00 03 11 CC 3dB bandwidth is 249Hz
0x0D	0x10	0x0001	Uint8	Set Operating Platform	0x00: Construction excavator (arm); 0x01: Construction tower crane; 0x02: Construction crushing; 0x03: AGV/vehicle-mounted application; 0x04: Mining truck application; 0x05: Shipborne application; 0x06: Generator fan; 0x07: Agricultural machinery or robot application; E.g: Setting up the application platform for agricultural machinery/robot application: TX: 54 4D 0D 10 01 00 07 39 03 RX: 54 4D 0D 10 00 00 03 69
0x0D	0x90	0x0000	NULL	Query Operating	E.g: Query the running platform: TX: 54 4D 0D 90 00 00 02 81

				Platform	RX: 54 4D 0D 90 01 00 07 10 C3 The current application platform is agricultural machinery/robot application

IMU Data Output(Class-0x0F, ID=0x01)

1. Data output items Out Map

OutMap specifies the data items to be output and the order in which the data is output. The data can be parsed based on the value of OutMap.

Data Output Map Items	Description	Sequential	Map Value
IMU_OUTMAP_EULER	Attitude angle	1	0x01
IMU_OUTMAP_ACCEL	Three-axis acceleration	2	0x02
IMU_OUTMAP_GYRO	Three-axis gyro angular velocity	3	0x04
IMU_OUTMAP_MAG	Three-axis magnetometer	4	0x08
IMU_OUTMAP_TC	Temperature	5	0x10
IMU_OUTMAP_BAT	Battery charge %	6	0x20
IMU_OUTMAP_STS	IMU status	7	0x40
IMU_OUTMAP_TPS	IMU local time (ms)	8	0x80

2. Data Item Analysis

2.1 Attitude angle IMU_OUTMAP_EULER

Attitude angles include Roll, Pitch, and Yaw, represented by float, totaling 12 bytes, stored in little-endian mode, and the unit is degrees.

Storage format:

Roll	Pitch	Yaw

2.2 Triaxial acceleration IMU_OUTMAP_ACCEL

The calibrated triaxial accelerations Ax, Ay, and Az are represented by float, totaling 12 bytes, stored in little-endian mode, and the unit is g.

Storage format:

Ax	Ay	Az

2.3 Triaxial angular velocity IMU_OUTMAP_GYRO

The calibrated triaxial angular velocities Gx, Gy, and Gz are represented by float, totaling 12 bytes, stored in little-endian mode, and the unit is degrees per second.

Storage format:

Gx	Gy	Gz

2.4 Triaxial magnetometer IMU_OUTMAP_MAG

The calibrated triaxial angular velocities Mx, My, and Mz are represented by float, totaling 12 bytes, stored in little-endian mode, and the unit is uT.

Storage format:

Mx	My	Mz

2.5 Temperature IMU_OUTMAP_TC

float represents a value of 4 bytes, stored in little-endian mode, with the unit being °C.

Storage format:

Tc

2.6 Battery capacity IMU_OUTMAP_BAT

Unsigned uint8 representation, 1 byte, 0-100%.

Storage format:

Bat%

2.7 Tilt Angle Status IMU_OUTMAP_STS

An unsigned uint8 value represents a memory location of 1 byte.

Storage format:

status

status($B_7B_6B_5B_4B_3B_2B_1B_0$): An unsigned single byte indicates the installation measurement method and IMU self-test status:

B_1B_0: 00: Indicates the IMU operating mode; 01 represents VRU, and 10 represents AHRS;

B_3B_2: Indicates the power-on self-test result:

00: Self-test successful;

01: Gyroscope accelerometer self-test failed;

10: Magnetometer self-test failed;

11: Both gyroscope accelerometer and magnetometer self-tests failed;

B7B6B5B4: The high 4 bits of the status byte indicate the installation test method (refer to the measurement installation method diagram for details):

0000: Horizontal installation measurement method.

0001: Vertical upward installation measurement method.

0010: Vertical left (lateral left) installation measurement method.

0011: Vertical right (lateral right) installation measurement method.

0100: Horizontal downward installation measurement method.

2.8 IMU local time IMU_OUTMAP_TPS

An unsigned uint32 is represented, consisting of 4 bytes, stored in little-endian mode, and measured in milliseconds (mS).

3. Data Parsing

Ensure that the data output frequency is not zero

Description	Meaning/E.g	Description
Class=0x0F ***ID = 0x01***	**Sensor response** 54 4D 0F 01 3B 00 FF 79 42 26 43 26 69 BE C0 67 4A 27 C2 A5 84 82 BD 2A 10 48 3E 91 54 3B C0 26 3A 53 BD 99 5E 8A C0 65 41 F3 BE 27 5C CB 40 04 E8 2D C2 7A FA 04 42 92 96 03 42 5D 02 3E 43 24 00 82 57	Data Fields OUTMAP_EULER: 0x01 represents Euler angles; OUTMAP_ACCEL: 0x02 represents acceleration; OUTMAP_GYRO: 0x04 represents gyro angular velocity; OUTMAP_MAG: 0x08 represents magnetometer readings; OUTMAP_TC: 0x10 represents temperature; OUTMAP_BAT: 0x20 represents battery level; OUTMAP_STS: 0x40 represents IMU status; OUTMAP_TPS: 0x80 represents time; OUTMAP=0xFF: Indicates that all the above data items are output, and the corresponding data

		items are output byte by byte from high to low. 54 4D: Sync1, Sync2 respectively; 0F 01: Class, ID respectively; 3B 00: Length; FF: OUTMAP; 79 42 26 43: represents Roll; 26 69 BE C0: represents Pitch; 67 4A 27 C2: represents Heading; A5 84 82 BD : represents Accel_X 2A 10 48 3E:represents Accel_Y 91 54 3B C0:represents Accel_Z 26 3A 53 BD: represents Gyro_X 99 5E 8A C0: represents Gyro_Y 65 41 F3 BE: represents Gyro_Z 27 5C CB 40: represents Mag_X 04 E8 2D C2: represents Mag_Y 7A FA 04 42: represents Mag_Z 92 96 03 42: represents temperature 5D: Indicates battery percentage% 02: Indicates IMU status 3E 43 24 00: Indicates local time

Byte Offset	No. Format	name	Content (HEX)	bytes	
0	INT8U	Sync1	54	1	Frame header 1
1	INT8U	Sync2	4D	1	Frame header 2
2	INT8U	CLASS	0F	1	CLASS
3	INT8U	ID	01	1	ID=01represents data
4	INT16U	length	3B 00	2	Data length
6	INT8U	OUTMAP	FF	1	MAP=0xFF, Output all data items
7	FT32	Roll	79 42 26 43	4	0x43264279=166.259659 deg
11	FT32	Yaw	26 69 BE C0	4	0xC0BE6926 = -5.950336 deg
11	FT32	Heading	67 4A 27 C2	4	0xC2274A67=-41.822659 deg
13	FT32	Acc_X	A5 84 82 BD	**4**	0xBD8284A5=-0.063730g
15	FT32	Acc_Y	2A 10 48 3E	4	0x3E48102A=0.195374g
17	FT32	Acc_Z	91 54 3B C0	4	0xC03B5491=-2.927037g
19	FT32	Gyro_X	26 3A 53 BD	4	0xBD533A26=-0.051569deg/sec
21	FT32	Gyro_Y	99 5E 8A C0	4	0xC08A5E99=-4.324048deg/sec
23	FT32	Gyro_Z	65 41 F3 BE	4	0xBEF34165=-0.475108deg/sec
25	FT32	Mag_x	27 5C CB 40	4	0x40CB5C27=6.354999uT
27	FT32	Mag_Y	04 E8 2D C2	4	0xC22DE804=-43.476578uT
29	FT32	Mag_Z	7A FA 04 42	4	0x4204FA7A=33.244606uT

31	FT32	TC	92 96 03 42	4	0x42039692=32.897041C
32	INT8U	BATT	5D	1	0x5D=93%
	INT8U	STS	02	1	$B_7B_6B_5B_4$=0000: Horizontal installation measurement method; B_3B_2=00: Force gauge self-test and gyroscope accelerator self-test successful; B_1B_0=10: AHRS Work mode.
33	INT32U	TPS	3E 43 24 00	4	0x0024433E=2376510mS
35	INT16U	CRC16	82 57	2	-

Appendix A: Data Transmission Routine

```
#define     IMU_OUTPUTMASK_EULER        0x01
#define     IMU_OUTPUTMASK_ACCEL        0x02
#define     IMU_OUTPUTMASK_GYRO         0x04
#define     IMU_OUTPUTMASK_MAG          0x08
#define     IMU_OUTPUTMASK_TEMPTC       0x10
#define     IMU_OUTPUTMASK_BATT         0x20
#define     IMU_OUTPUTMASK_STS          0x40
#define     IMU_OUTPUTMASK_TIME         0x80

unsigned short int 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
};

unsigned short int CRC16_Check(unsigned char    *p, unsigned short int length)
{
     unsigned short int checksum = 0;
     for( ; length > 0; length-- )
   {
       checksum = ( checksum >> 8 ) ^ CRC16Table[ (checksum&0xFF) ^ *p ];
       p++;
   }
   return checksum;
}

typedef union
{
       float    ft32_data;
       uint8_t u8_data[4];
}ft32_Type;

void IMU_Ft32_To_Int8_UnPack(uint8_t *p8, float f32) // float to u8
{
         ft32_Type ft32_value;
         uint8_t i;
         ft32_value.ft32_data = f32;
         for(i = 0; i<4; i++)
       {
                p8[i]=ft32_value.u8_data[i];
       }
}

void AH100B_BLE_AutoOut(uint8_t mask)
{
```

```
unsigned short int checksum, index=0;

BLE_Nus_Txd_Buf[0] = 'T';
BLE_Nus_Txd_Buf[1] = 'M';
BLE_Nus_Txd_Buf[2] = 0x0F;
BLE_Nus_Txd_Buf[3] = 0x01;

BLE_Nus_Txd_Buf[6+index] = mask;
 index++;
if(mask&IMU_OUTPUTMASK_EULER)
{
        IMU_Ft32_To_Int8_UnPack(BLE_Nus_Txd_Buf+6+index, cr);
         index += 4;
        IMU_Ft32_To_Int8_UnPack(BLE_Nus_Txd_Buf+6+index, cp);
         index += 4;
        IMU_Ft32_To_Int8_UnPack(BLE_Nus_Txd_Buf+6+index, ch);
         index += 4;
}
if(mask&IMU_OUTPUTMASK_ACCEL)
{
    if(userConfig.accelOutType == 0){
        IMU_Ft32_To_Int8_UnPack(BLE_Nus_Txd_Buf+6+index,imuData.absnrzi);
        index += 4;
        IMU_Ft32_To_Int8_UnPack(BLE_Nus_Txd_Buf+6+index,imuData.absnpzi);
        index += 4;
        IMU_Ft32_To_Int8_UnPack(BLE_Nus_Txd_Buf+6+index,imuData.absnzzi);
         index += 4;
        }else{
        IMU_Ft32_To_Int8_UnPack( BLE_Nus_Txd_Buf+6+index, imuData.obj_AC_x);
        index += 4;
        IMU_Ft32_To_Int8_UnPack( BLE_Nus_Txd_Buf+6+index, imuData.obj_AC_z);
        index += 4;
        IMU_Ft32_To_Int8_UnPack( BLE_Nus_Txd_Buf+6+index,-imuData.obj_AC_y);
         index += 4;
        }
    }
    if(mask&IMU_OUTPUTMASK_GYRO)
    {
        IMU_Ft32_To_Int8_UnPack(BLE_Nus_Txd_Buf+6+index,imuData.tempgroyxx);
        index += 4;
        IMU_Ft32_To_Int8_UnPack(BLE_Nus_Txd_Buf+6+index,imuData.tempgroyyy);
        index += 4;
        IMU_Ft32_To_Int8_UnPack(BLE_Nus_Txd_Buf+6+index,imuData.tempgroyzz);
        index += 4;
```

```
        }
        if(mask&IMU_OUTPUTMASK_MAG)
        {
                IMU_Ft32_To_Int8_UnPack(BLE_Nus_Txd_Buf+6+index,magicData.Xraw);
                index += 4;
                IMU_Ft32_To_Int8_UnPack(BLE_Nus_Txd_Buf+6+index,magicData.Yraw);
                index += 4;
                IMU_Ft32_To_Int8_UnPack(BLE_Nus_Txd_Buf+6+index,magicData.Zraw);
                index += 4;
        }
        if(mask&IMU_OUTPUTMASK_TEMPTC)
        {
                IMU_Ft32_To_Int8_UnPack(BLE_Nus_Txd_Buf+6+index, imuData.tc);
                index += 4;
        }
        if(mask&IMU_OUTPUTMASK_BATT)
        {
                BLE_Nus_Txd_Buf[6+index] = imuData.batVol;
                index++;
        }
        if(mask&IMU_OUTPUTMASK_STS)
        {
                BLE_Nus_Txd_Buf[6+index] = facConfig.fixed_Mode<<4;
                if(b_MMCSts==1){BLE_Nus_Txd_Buf[6+index] |= 0x08;}
                if(b_IimSts==1){BLE_Nus_Txd_Buf[6+index] |= 0x04;}
                BLE_Nus_Txd_Buf[6+index]|=facConfig.navMode;
                index++;
        }
        if(mask&IMU_OUTPUTMASK_TIME)
        {
                *((uint32_t *)(&BLE_Nus_Txd_Buf[6+index])) = elapsedTime;
                index+=4;
        }
          BLE_Nus_Txd_Buf[4] = index;
        BLE_Nus_Txd_Buf[5] = index>>8;

        checksum = CRC16_Check(BLE_Nus_Txd_Buf+2,     4+index);
        BLE_Nus_Txd_Buf[6+index] =    checksum&0x00FF;
        BLE_Nus_Txd_Buf[7+index] =    checksum>>8;
        BLE_Nus_Transmit_Enable(BLE_Nus_Txd_Buf,index+8);
}
```

Appendix B: Data Parsing Routine

```
unsigned char GetAH100BMapData(unsigned char *pxdata)
{
    unsigned char map, datalen = 0;
    map = pxdata[0];
    datalen++;
    if (map&IMU_OUTPUTMASK_EULER)
    {
        imuData.phi = *((float *)(pxdata + datalen));
        datalen += 4;
        imuData.theta = *((float *)(pxdata + datalen));
        datalen += 4;
        imuData.psi = *((float *)(pxdata + datalen));
        datalen += 4;
    }
    if (map&IMU_OUTPUTMASK_ACCEL)
    {
        imuData.accx = *((float *)(pxdata + datalen));
        datalen += 4;
        imuData.accy = *((float *)(pxdata + datalen));
        datalen += 4;
        imuData.accz = *((float *)(pxdata + datalen));
        datalen += 4;
    }
    if (map&IMU_OUTPUTMASK_GYRO)
    {
        imuData.groyx = *((float *)(pxdata + datalen));
        datalen += 4;
        imuData.groyy = *((float *)(pxdata + datalen));
        datalen += 4;
        imuData.groyz = *((float *)(pxdata + datalen));
        datalen += 4;
    }
    if (map&IMU_OUTPUTMASK_MAG)
    {
        imuData.magx = *((float *)(pxdata + datalen));
        datalen += 4;
        imuData.magy = *((float *)(pxdata + datalen));
        datalen += 4;
        imuData.magz = *((float *)(pxdata + datalen));
        datalen += 4;
    }
    if (map&IMU_OUTPUTMASK_TEMPTC)
    {
        imuData.imu_tc = *((float *)(pxdata + datalen));
```

```
            datalen += 4;
        }
        if (map&IMU_OUTPUTMASK_BATT)
        {
            imuData.batvol = *(pxdata + datalen);
            datalen += 1;
        }
        if (map&IMU_OUTPUTMASK_STS)
        {
            imuData.imuStatus = *(pxdata + datalen);
            datalen += 1;
        }
        if (map& IMU_OUTPUTMASK_TIME)
        {
            imuData.elapsedtime = *((unsigned int*)(pxdata + datalen));
            datalen += 4;
        }
        return datalen;
}
void AH100B_Command_Check(unsigned char *pxdata)
{
        unsigned char ahrs_class, ahrs_id, map, temp;
        unsigned short cmd_length;
        CEdit *pedit;
        float ppfloat;
        CString str1;

        ahrs_class = *pxdata;
        ahrs_id = *(pxdata + 1);
        temp = *(pxdata + 2);
        cmd_length = *(pxdata + 3);
        cmd_length <<= 8;
        cmd_length |= temp;

        if (ahrs_class == 0x0F)
        {
            switch (ahrs_id)
            {
                case 0x01:
                {
                    map = *(pxdata + 4);
                    if (map != 0)
                    {
                        if (cmd_length != GetAH100BMapData(pxdata + 4))
```

```
                    {
                        return;
                    }
                }
                break;
            }
            Default:
                Break;
}
}
}
```

```
Void main(viod)
{
    If(b_FrameRec == 1)
{
      AH100B_Command_Check(frameAH100BPack.Usart_Rxd_Buf + 2);
}
}
```

Appendix C: Manual Gyroscope Zero-Bias Clearing

Step1: Set the IMU to standstill and send the manual gyroscope zero bias clear start command.

Manual gyroscope zero bias clearing begins

TX: 54 4D 0F 06 01 00 01 C4 89

RX: 54 4D 0F 06 00 00 E3 15

Step2: Wait 5 seconds to check the current manual gyroscope zero-bias clear status.

Check manual gyroscope zero-bias clear status

TX: 54 4D 0F 86 00 00 E2 FDRX: 54 4D 0F 86 01 00 01 ED 49 The return value is 01 ,

indicating manual gyroscope zero-bias is in progress.

Continue querying until the following response returns 00, indicating that manual gyroscope zero-bias has ended.

TX: 54 4D 0F 86 00 00 E2 FD

RX: 54 4D 0F 86 01 00 00 2C 89

Manual gyroscope zero-bias clearing has ended.

Step3: Preserve the zero bias of the gyroscope

TX: 54 4D 0F 07 01 00 0A 84 B2

RX: 54 4D 0F 07 00 00 B2 D5

Appendix D: Magnetic Field Horizontal and Vertical Rotation Calibration in AHRS Mode