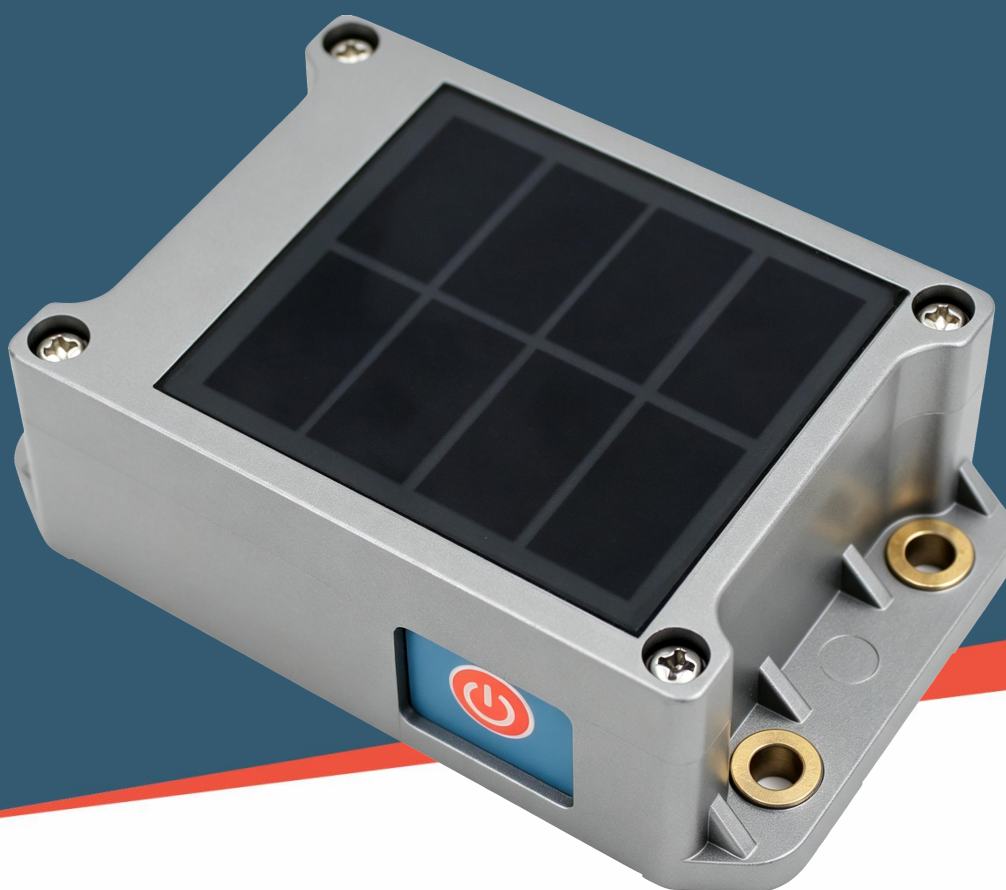




V1.5

Bluetooth Dynamic Attitude Sensor

BDA450

Technical Manual

BDA450 Bluetooth Dynamic Attitude Sensor

○ Revision date: 2026-7-1

Note: Product functions, parameters, appearance, etc. will be adjusted as the technology upgrades. Please contact our pre-sales business to confirm when purchasing.

Disclaimer

This product is developed exclusively for commercial applications and is prohibited from being used for illegal purposes, such as military activities, research related to nuclear, chemical, or biological weapons, or any other actions that violate laws and regulations. It is strictly forbidden to engage in activities related to goods embargoed by the United Nations, the European Union, or the OSCE, as well as any other activities prohibited by the Export Administration Regulations.

Without the written consent of our company, this product must not be transferred to any third party. Purchasing this product signifies your acceptance of this disclaimer and your agreement to sign the related liability agreement. If these regulations are violated, all responsibilities will be borne by the purchaser, and our company will not be held liable.

BDA450 Bluetooth Dynamic Attitude Sensor



● **PRODUCT DESCRIPTION**

BDA450 is a high cost-effective Bluetooth output dynamic attitude sensor that can be widely used in equipment that requires high dynamic balance, such as model aircraft, robots, agricultural machinery, construction vehicles, forklifts, etc. This device collects sensor data, integrates Kalman filtering, and outputs real-time attitude data. Due to the use of three-axis accelerometer, three-axis gyroscopes, and temperature compensation algorithm technology, the product achieves excellent stability and real-time performance in all attitudes, and is widely used in various motion attitude monitoring fields. This product not only outputs attitude data, but also three-axis acceleration, three-axis gyroscope and other data. It can also modify and expand system operating parameters, making it suitable for various application platforms. This product adopts Bluetooth wireless output, Bluetooth 5.1 wireless transmission module, and BLE protocol. Integrated 2.4GHz adaptive frequency hopping technology enables 40 meter line of sight transmission (unobstructed environment) and BLE DFU Bluetooth wireless firmware upgrade, significantly improving the flexibility of industrial field deployment.

● **MAIN FEATURES**

- ★ Low noise
- ★ Solar power supply
- ★ Real time 3D dynamic measurement
- ★ Maximum update rate of 100Hz
- ★ Integrated nine axis sensor and temperature sensor
- ★ Bluetooth 5.1 wireless transmission
- ★ Kalman filtering algorithm
- ★ Long lifespan, strong stability
- ★ High performance drift stability

● **APPLICATION FIELD**

- ★ Leveling and flattening
- ★ Mobile machinery
- ★ Boring and drilling applications
- ★ Mobile elevators and fire trucks
- ★ Automatic Guided Vehicles (AGV)
- ★ Robot applications
- ★ Agricultural machinery
- ★ Elevated platforms
- ★ Cranes and construction machinery
- ★ Automatic assembly machinery



● SPECIFICATIONS

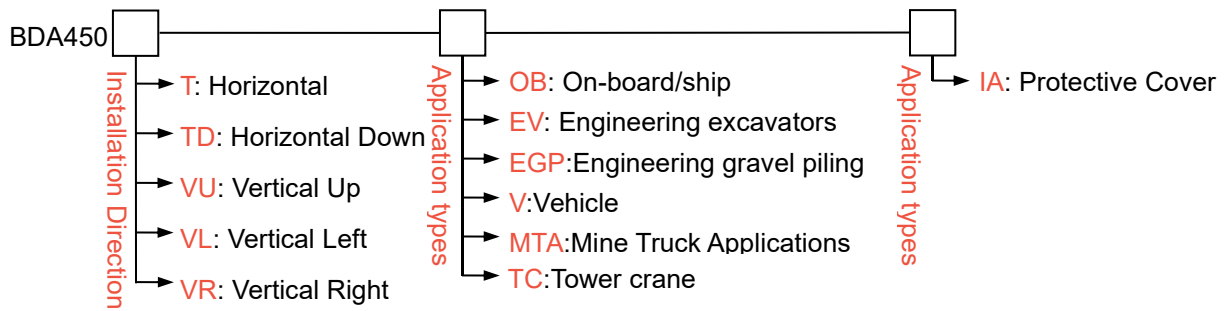
BDA450		Condition	Parameter
Measure range		-	Roll $\pm 180^\circ$, pitch $\pm 90^\circ$, azimuth $\pm 180^\circ$ (initial value at power-down is 0°)
Measure axis		-	X axis / Y axis / Z axis
Roll&Pitch Static accuracy ¹⁾		@25 $^\circ\text{C}$	$\pm 0.05^\circ$
Roll&Pitch Dynamic accuracy(rms)		@25 $^\circ\text{C}$	$\pm 0.1^\circ$
Gyro	range	-	$\pm 300^\circ/\text{s}$
	Zero bias stability (10s Smooth)	-	8.5 $^\circ/\text{h}$
	Angle random walk coefficient(Allan)	-	0.25 $^\circ/\text{sqrt}(\text{h})$
Acceleration	range	-	$\pm 8\text{g}$
	Bias stability(10s mean)	-	0.1mg
	Bias instability(Allan)	-	0.005mg
	Speed random walk coefficient(Allan)	-	0.005m/s/sqrt(h)
Zero point temp. Coefficient ²⁾		-20~70 $^\circ\text{C}$	$\pm 0.01^\circ/\text{C}$
Sensitivity temp. Coefficient ³⁾		-20~70 $^\circ\text{C}$	$\leq 100\text{ppm}/^\circ\text{C}$
Power-on startup time			$\leq 3.5\text{S}$
Response time			0.01S
Electromagnetic compatibility			According to EN61000 and GBT17626
reception sensitivity			-96dBm
transmit power			8dBm
Bluetooth transmission distance			50m (in an open and unobstructed state)
Bluetooth frequency band			2405-2480MHz (16 channels)
Usage time (battery fully charged)			About 20 days (at 25 $^\circ\text{C}$, working for 8 hours a day, sleeping for 16 hours)
Insulation resistance			≥ 100 megaohms
Impact resistance			100g@11ms Triaxial (half sine wave)
Anti-vibration			10grms / 10~1000Hz
waterproof rating			IP67
Shell material			PPS plastic shell
weight			$\leq 230\text{g}$ (excluding cable)

1) Accuracy: Refers to the root mean square difference between the measured value and the actual angle error obtained by measuring the angle multiple times (>16 times) at room temperature.

2) Zero point temp. Coefficient: refers to the rate of change in the displayed value of a sensor relative to normal temperature within its rated operating temperature range when it is in a zero state.

3) Sensitivity temp. Coefficient: refers to the percentage of full-scale indication of a sensor relative to full-scale indication at room temperature within its rated operating temperature range, and the rate of change with temperature.

● ORDER INFORMATION



E.g: BDA450T-OB: Horizontal installation/On-board/ship application.

BDA450T-OB-IA: Accessories are installed based on customer requirements. If a protective cover is needed, please add IA.

● BATTERY PARAMETERS

PROJECT	PARAMETER
Nominal capacity	1400mAh
Rated voltage	3.65V
Charging cut-off voltage	4.20±0.05V

● ELECTRICAL PARAMETERS

Parameter	Condition	MIN	TYP	MAX	unit
Solar panel parameters	Dimensions: 50*55*2.4mm / 4V / 120mA (1000lm)				-
Maximum charging current			0.02	0.25	A
Solar charging temp	-20 ~ +70 (Note: Charging temperatures that are too high or too low can damage the battery)				°C
Solar discharge temp		-20 ~ +70			°C
Operating Temp		-20 ~ +70			°C
operating current			6.5		mA
Shutdown and sleep current			70		uA
Operating Temp		-20		+70	°C
Storage temp		-20		+70	°C

Attention: For solar panels, we recommend facing upwards, south, east, and west.

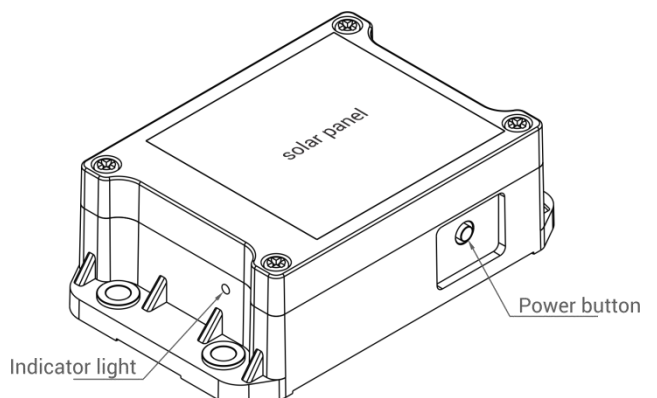
Warning: The minimum power generation is towards the north, which affects the use of the product. Ensure that the solar panels are unobstructed and clean.

● FUNCTIONAL DESCRIPTION

1. Work indicator light: green;

① Press and hold the power button for 1 second, and the green light will flash for 2 seconds, indicating power on/off.

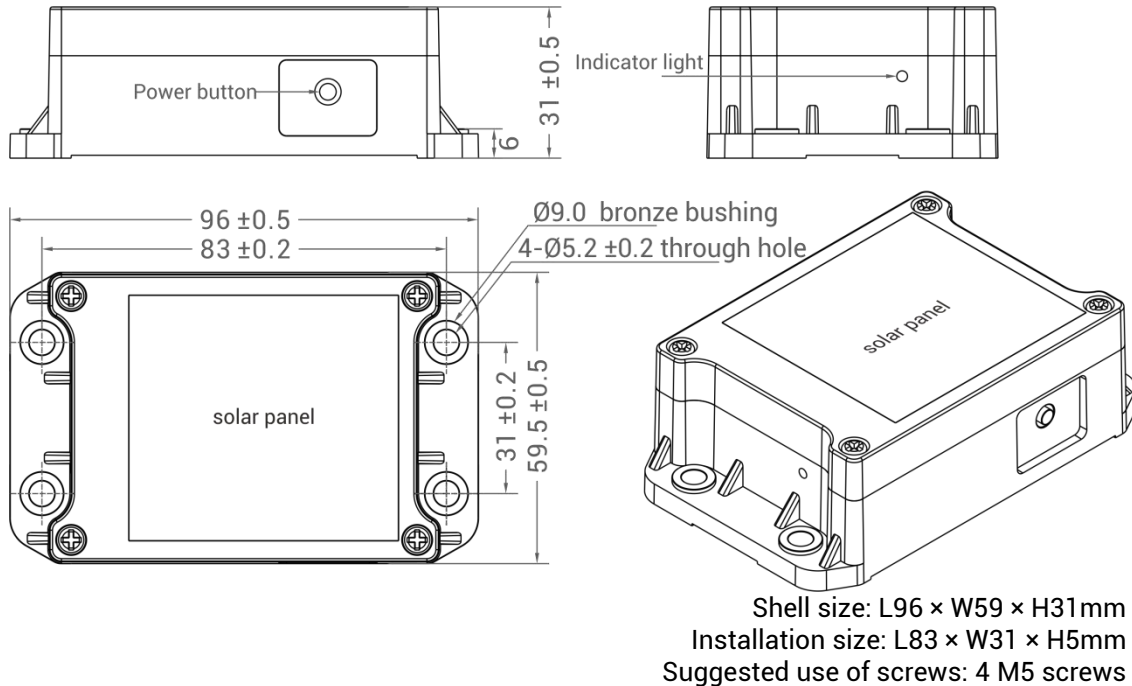
② During broadcasting, the green light flashes once every 2 seconds.



BDA450 Bluetooth Dynamic Attitude Sensor

- ③ After connecting to the host, the green light flashes once every 5 seconds.
2. After booting up, the tilt angle remains in low-power Bluetooth broadcast mode, waiting for the host to connect. Turning on the host Bluetooth can scan and discover the slave and connect it. If the host is successfully connected, the slave will transmit tilt data to the host. At this time, the green light will flash once every 5 seconds. If a connection timeout event occurs, the Bluetooth will disconnect and the tilt will be in low-power Bluetooth broadcast mode again, waiting for the host to connect.

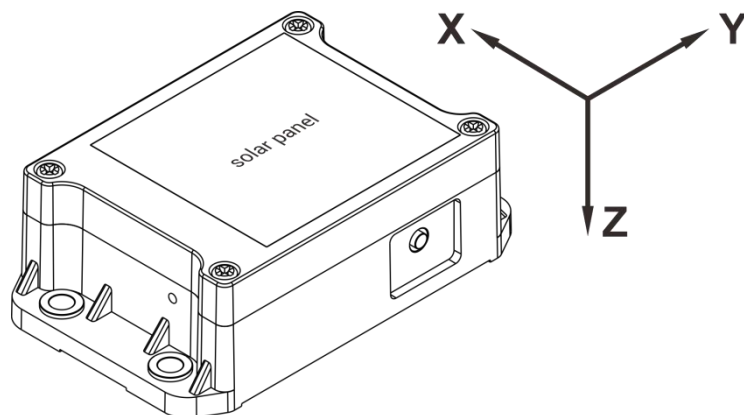
● SIZE CHART



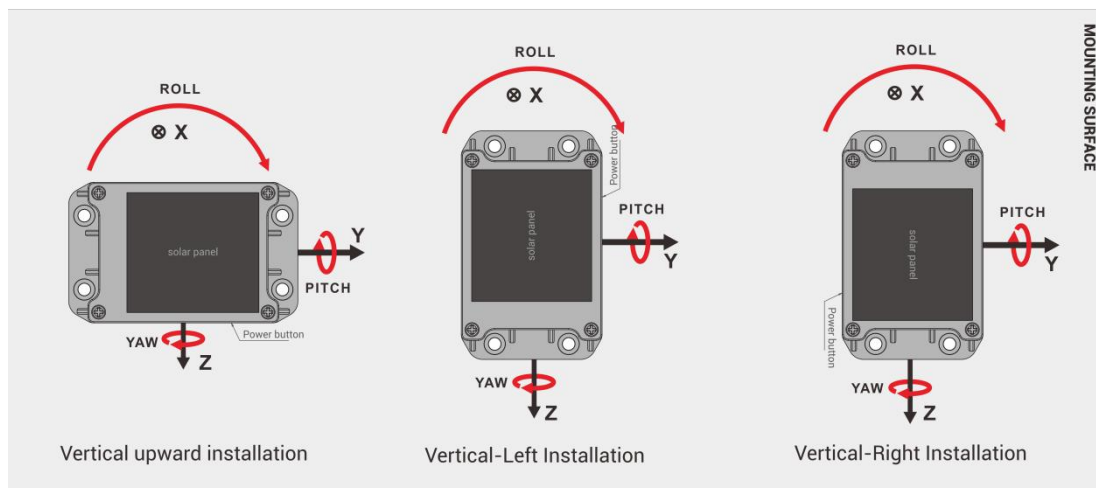
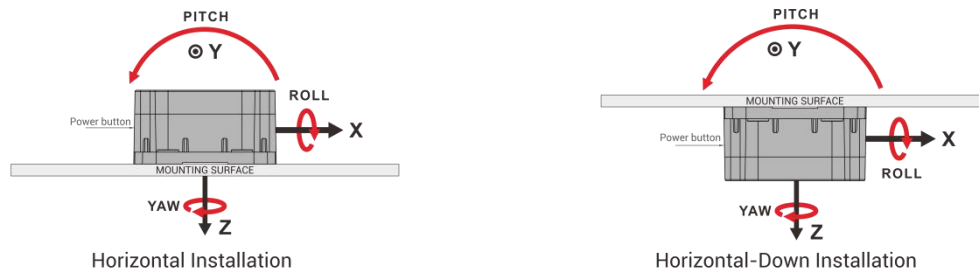
● MEASUREMENT DIRECTION

Please install the tilt sensor according to the correct method. Incorrect installation can cause measurement errors, especially paying attention to the first "surface" and second "line":

- 1) The installation surface of the sensor must be tightly, smoothly, and stably fixed to the measured surface. If the installation surface is uneven, it can easily cause measurement angle errors in the sensor.
- 2) The axis of the sensor must be parallel to the axis of the object being measured, and the two axes should not form an angle as much as possible.



● INSTALLATION DIRECTION



● BLE PROTOCOL

This protocol is applicable to the communication between the master and the slave based on Bluetooth low energy (BLE) technology, and is mainly applied to the data acquisition, data transmission, remote control, etc. Of the master and the Bluetooth slave (BDA450, hereinafter referred to as the slave).

BLE protocol basis: Based on the GATT (General Attribute Protocol) model, the slave is the GATT server, and the host (mobile APP, or PC) is the GATT client.

GATT (Generic Attribute Protocol): The core protocol for BLE data transfer, based on the client-server model.

Service: Defines device capabilities (e.g. RX, TX services).

Characteristic: Specific data points in the service (such as inclination value, acceleration value, gyro angular velocity value, battery level, etc.).

Descriptor: Additional information for the configuration feature (such as notification enable).

Communication model

Role definition:

Central: Initiates a connection request to read or write data to the slave.

Peripheral: Broadcast its own services, initiate connection and data transmission requests to the host, and respond to the read and write requests of the host.

Communication mode: master-slave two-way communication mode is supported.

Scenario 1: The Bluetooth slave device broadcasts the transmission data to the host regularly (the timing period can be adjusted by the remote command issued by the host);

Scenario 2: The host can initiate a communication request to the Bluetooth peripheral, read and write data to the peripheral, such as modifying the relevant parameters of the peripheral.

Terms and definitions

GATT: General Attribute Protocol for BLE data transfer.

UUID: a universally unique identifier used to identify services and characteristic values.

Data format

Service and characteristic value definition

The slave provides the following GATT services and characteristic values:

Customer UUID type: 0x02 (BLE _ UUID _ TYPE _ VENDOR _ BEGIN)

Base UUID: 0x00000000-AFDD-43A1-9651-C6407A7C2515

Main Service UUID: 0x6801 (Tilt Sensor)

Feature value 1: data reception (UUID: 0x6802, writable, no response writing)

Feature value 2: data transmission (UUID: 0x6803, readable, notification)

OTA DFU firmware upgrade service (refer to Nordic official website for details):

Customer UUID type: 0x01 (BLE _ UUID _ TYPE _ BLE)

Base UUID: 0x8EC90000-F315-4F60-9FB8-838830DAEA50

Primary Service UUID: 0xFE59

Characteristic value: BLE _ DFU _ BUTTONLESS _ CHAR _ UUID (0x0003)

Device Name: BDA450-XXXXXXXX

BDA450: product model;

XXXXXXXX: indicates the product identification number ID (e.g. BDA450-25090001);

● BLE PROTOCOL

This protocol is applicable to communication between hosts and slaves based on Bluetooth Low Energy (BLE) technology, mainly used for data acquisition, data transmission, remote control, etc. between hosts and Bluetooth slaves (BDA450T, hereinafter referred to as slaves).

BLE protocol basis: Based on the GATT (General Attribute Protocol) model, the slave is the GATT server, and the host (mobile APP, or PC) is the GATT client.

GATT (Generic Attribute Protocol): The core protocol for BLE data transfer, based on the client-server model.

Service: Defines device capabilities (e.g. RX, TX services).

Characteristic: Specific data points in the service (such as inclination value, acceleration value, gyro angular velocity value, battery level, etc.).

Descriptor: Additional information for the configuration feature (such as notification enable).

Communication model

Role definition:

Central: Initiates a connection request to read or write data to the slave.

Peripheral: Broadcast its own services, initiate connection and data transmission requests to the host, and respond to the read and write requests of the host.

Communication mode: master-slave two-way communication mode is supported.

Scenario 1: The Bluetooth slave device broadcasts the transmission data to the host regularly (the timing period can be adjusted by the remote command issued by the host);

Scenario 2: The host can initiate a communication request to the Bluetooth peripheral, read and write data to the peripheral, such as modifying the relevant parameters of the peripheral.

Terms and definitions

GATT: General Attribute Protocol for BLE data transfer.

UUID: a universally unique identifier used to identify services and characteristic values.

Data format

Service and characteristic value definition

The slave provides the following GATT services and characteristic values:

Customer UUID type: 0x02 (BLE _ UUID _ TYPE _ VENDOR _ BEGIN)

Base UUID: 0x00000000-AFDD-43A1-9651-C6407A7C2515

Main Service UUID: 0x6801 (Tilt Sensor)

Feature value 1: data reception (UUID: 0x6802, writable, no response writing)

Feature value 2: data transmission (UUID: 0x6803, readable, notification)

OTA DFU firmware upgrade service (refer to Nordic official website for details):

Customer UUID type: 0x01 (BLE _ UUID _ TYPE _ BLE)

Base UUID: 0x8EC90000-F315-4F60-9FB8-838830DAEA50

Primary Service UUID: 0xFE59

Characteristic value: BLE _ DFU _ BUTTONLESS _ CHAR _ UUID (0x0003)

Device Name: BDA450-XXXXXXXX

BDA450: product model;

XXXXXXXX: indicates the product identification number ID (e.g. BDA450-25090001);

● BLUETOOTH BLE DATA PROTOCOL

1. Data frame format: (8 data bits, 1 stop bit, no check)

Identifier (1byte)	Data length (1byte)	Address Code (1byte)	Command word (1byte)	Data domain	Checksum (1byte)
68h					

Identifier: fixed as 68H;

Data length: length from data length to checksum (including checksum);

Address code: the address of acquisition module, which is always 00;

And that data field is correspondingly Chan according to different content and lengths of the command words;

Checksum: The sum of the data length, address code, command word, and data field without regard to carry.

II. Command Word Analysis

CMD	Meaning/example	Explain
0X04	Query default data item output command 68 04 00 04 08	Data field (0 byte) No data domain command When working in the query mode (data output cycle is 0), use these commands to query the data of the current OUTMAP data item.
0X84	Sensor Response E.g:68 1D 00 84 7F 1D 13 00 DD 8B FF 17 FC FB FF FC FF 01 00 FF FF 00 00 F6 0A 60 10 98 4E 14	Data domain See the subsequent detailed explanation for the specific data output analysis.
0X03	Query the specified OUTMAP data item output 68 05 00 03 07 0F	Data field (1byte) The data field is the or of the following; The following data items are currently supported: OUTMAP _ EULER: 0x01 represents Euler angle; OUTMAP _ ACCEL: 0x02 indicates acceleration; OUTMAP _ GYRO: 0x04 indicates the angular velocity of gyro; OUTMAP _ TC: 0x08 indicates temperature; OUTMAP _ BAT: 0x10 indicates the battery level; OUTMAP _ STS: 0x20 indicates IMU status; OUTMAP _ TPS: 0x40 represents time; 0x07: Euler angle, angular velocity of accelerometer and gyro;
0X83	Sensor Response E.g:68 17 00 83 07 4C E1 4D F8 21 F8 A1 FE 97 03 51 FF 5C FF F1 00 D6 FF D7	Data Domain See the subsequent detailed explanation for the specific data output analysis.
0XF2	Set Installation Measurement Direction Command 68 06 00 F2 00 01 F9	Data Domain (2-byte) 00 01 The first byte 00: indicates that the sub command is set; Second byte 01: indicates the installation measurement method; 0x00: represents the horizontal installation measurement method; 0x01: indicates the vertical upward installation

		measurement method; 0x02: Indicates the lateral to left installation measurement method; 0x03: Indicates the lateral to right installation measurement method; 0x04: Indicates the horizontal reverse installation measurement method.
0XF2	Sensor Response E.g: 68 06 00 F2 00 00 8B	Data Domain (2 bytes) The numbers in the data domain represent the results of the sensor response Byte 1 00: Indicates that the sub command is set 2nd byte: 00 successfully set FF setting failed
0XF2	Query installation measurement direction command 68 05 00 F2 01 F8	Data Domain (1 byte) 01 The first byte 01: indicates that the sub command is a query;
0XF2	Sensor Response E.g: 68 06 00 F2 01 01 FA	Data Domain (2-byte) 01 01 The first byte 01: indicates that the sub command is a query; Second byte 01: indicates the installation measurement method; 0x00: represents the horizontal installation measurement method; 0x01: indicates the vertical upward installation measurement method; 0x02: Indicates the lateral to left installation measurement method; 0x03: Indicates the lateral to right installation measurement method; 0x04: Indicates the horizontal reverse installation measurement method
0X88	Software reset command 68 04 00 88 8C	Data field (0 byte) No data domain command
0X88	Sensor Response E.g: 68 04 00 88 8C	Data field (0 byte) The sensor is reset immediately after the reply, and the broadcast is restarted
0X0D	Set data output items MAP 68 05 00 0D 7F 91	Data Domain (1 byte) The data domain is either;; Currently supports the following data items: OUTMAP_SULER: 0x01 represents Euler angle; OUTMAP-ACCEL: 0x02 represents acceleration; OUTMAP_CYRO: 0x04 represents gyroscope angular velocity; OUTMAP_TC: 0x08 represents temperature; OUTMAP_SAT: 0x10 represents battery level; OUTMAP-STs: 0x20 represents IMU status; OUTMAP_TPS: 0x40 represents time; 0x7F: indicates that all the above data items are output;
0X8D	Sensor Response E.g: 68 05 00 8D 00 92	Data field (1byte) The number in the data field indicates the result of the sensor respond 0 0 Setting succeeded FF Setting failed
0X1D	Query data output item MAP 68 04 00 1D 21	Data field (0 byte) No data domain command
0X9D	Sensor Response	Data field (1byte)

BDA450 Bluetooth Dynamic Attitude Sensor

	E.g: 68 05 00 9D 7F 21	The number in the data field indicates the result of the sensor response The following data items are currently supported: OUTMAP _ EULER: 0x01 represents Euler angle; OUTMAP _ ACCEL: 0x02 indicates acceleration; OUTMAP _ GYRO: 0x04 indicates the angular velocity of gyro; OUTMAP _ TC: 0x08 indicates temperature; OUTMAP _ BAT: 0x10 indicates the battery level; OUTMAP _ STS: 0x20 indicates IMU status; OUTMAP _ TPS: 0x40 represents time; 0 x7F: indicates that all the above data items are output;
0X0F	Set Product Identification Number ID Command 68 08 00 0F D1 D7 7E 01 3E	Data domain (4 bytes, uint32 small end mode) The ID ranges from 0x0000000 to 0xFFFFFFFF. For example, ID=0x017ED7D1=25090001, The device name is "BDA450T-25090001", which is used to display the slave device name during slave broadcasting.
0X8F	Sensor Response E.g: 68 05 00 8F 00 94	Data Domain (1 byte) The numbers in the data domain represent the results of the sensor response 00 setting successful FF setting failed
0X1F	Query Product ID Command 68 04 00 1F 23	Data domain (4 bytes, uint32 small end mode) The ID ranges from 0x0000000 to 0xFFFFFFFF. For example, ID=0x017ED7D1=25090001, The device name is "BDA450T-25090001", which is used to display the slave device name during slave broadcasting.
0X9F	Sensor response reply command E.g: 68 08 00 9F D1 D7 7E 01 CE	Data Domain (4 bytes) The numbers in the data domain represent the results of the sensor response ID=0x017ED7D1=25090001
0X0C	Set Bluetooth output cycle 68 08 00 0C E8 03 00 00 FF	Data domain (4 bytes, uint32 small end mode) Currently supported: 0, 1000mS, 200mS, 100mS, 50mS, 40mS, 20mS, 10mS. 00 00 00 00: Question and answer format (data query mode) E8 03 00 00: output cycle 1000mS; C8 00 00 00: output cycle 200mS; 64 00 00 00: output cycle 100mS; 32 00 00 00: output cycle 50mS; 28 00 00 00: output cycle 40mS; 14 00 00 00: output cycle 20mS; 0A 00 00 00: output cycle 10mS; Note: To reduce the output cycle (increase the output frequency), software reset is required, and re broadcasting the connection is effective.
0X8C	Sensor response reply command E.g: 68 05 00 8C 00 91	Data Domain (1 byte) The numbers in the data domain represent the results of the sensor response 00 setting successful FF setting failed

0X1C	Query Bluetooth output cycle 68 04 00 1C 20	Data domain (none)
0X9C	Sensor response reply command E.g: 68 08 00 9C E8 03 00 00 8F	Data Domain (4 bytes) The numbers in the data domain represent the results of the sensor response Currently supported: 0, 1000mS, 200mS, 100mS, 50mS, 40mS, 20mS, 10mS. 00 00 00 00: Q&A style (data query mode) E8 03 00 00: output cycle 1000mS; C8 00 00 00: output cycle 200mS; 64 00 00 00: output cycle 100mS; 32 00 00 00: output cycle 50mS; 28 00 00 00: output cycle 40mS; 14 00 00 00: output cycle 20mS; 0A 00 00 00: output cycle 10mS;

Data format description:

SOF	Item 1	Item 2	Item 3	Item 4	Item 5	Item 6
1byte	1byte	1byte	1byte	1byte	N bytes	1byte
0x68	Length	Address	0x84/0x83	Out Map	Load	Checksum

Description of Agreement:

SOF: indicates the frame start character, which is the fixed item 0x68;

Length of the first item: the total number of bytes of other items except SOF;

The second item, Address, is the address (reserved, always 0);

Item 3: data type, 0x84 represents the output data of the default data output item MAP or query the default output data; 0x83 represents the output data of the specified data output item MAP of the query;

The fourth item, Out Map, is used to output the data item MAP, map the corresponding data item, and provide the data analysis basis;

The fifth item, Load, consists of a command or the content of a command;

Checksum of the sixth item: It is the cumulative sum of the first item to the third item, single byte, excluding the high-order byte.

1. Data Output Item Out Map

Data Output Out Map Item	Description	Order	Map value
IMU_OUTMAP_EULER	Attitude angle	1	0x01
IMU_OUTMAP_ACCEL	Triaxial acceleration	2	0x02
IMU_OUTMAP_GYRO	Three-axis gyro angular velocity	3	0x04
IMU_OUTMAP_TC	Temperature	4	0x08
IMU_OUTMAP_BAT	Battery Power%	5	0x10
IMU_OUTMAP_STS	IMU status	6	0x20
IMU_OUTMAP_TPS	IM U local time mS	7	0x40

2. Data item parsing

2.1 Attitude angle IMU _ OUTMAP _ EULER.

The attitude angle includes Roll, Pitch and Yaw, represented by the symbol int16, 6 bytes in total, stored in little-endian mode, divided by 100 to obtain the floating-point angle value, and the unit is degree.

Storage format:

Roll	Pitch	Yaw
------	-------	-----

2.2 Three-axis acceleration IMU _ OUTMAP _ ACCEL.

Calibrated three-axis acceleration Ax, Ay and Az, represented by symbol int16, 6 bytes in total, stored in little-endian mode, divided by 1000 to obtain floating-point acceleration value, unit: G.

Storage format:

Ax	Ay	Az
----	----	----

2.3 Three-axis angular velocity IMU _ OUTMAP _ GYRO.

Calibrated three-axis angular velocities Gx, Gy and Gz are represented by the symbol int16, 6 bytes in total, stored in little-endian mode, divided by 100 to obtain the floating-point angular velocity value, and the unit is deg/sec.

Storage format:

Gx	Gy	Gz
----	----	----

2.4 Temperature IMU _ OUTMAP _ TC.

Symbolic int16, 2 bytes in total, stored in little-endian mode, divided by 100 to obtain the floating-point temperature value, in °C.

Storage format:

Tc

2.5 Battery capacity IMU _ OUTMAP _ BAT.

Unsigned uint8, 1 byte, 0-100%.

Storage format:

Bat%

2.6 Tilt State IMU _ OUTMAP _ STS.

Unsigned uint8, 1 byte total.

Storage format:

status

Status (B₇B₆B₅B₄B₃B₂B₁B₀): unsigned single byte, indicating installation measurement mode and IMU BIT status:

B₁B₀: 00: reserved;

B₃B₂: indicates the result of power-on self-test:

00: Self-test is successful;

01: The redundant acceleration self-test fails;

10: Self-test of main accelerometer fails;

11: BIT of main acceleration and redundant acceleration fails;

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

0000: Horizontal installation measurement mode.

0001: Vertical upward installation measurement mode.

0010: indicates the measurement mode of vertical left (lateral left) installation.

0011: indicates the measurement mode of vertical rightward (lateral rightward) installation.

0100: indicates the measurement mode of horizontal downward installation.

2.7 IMU local time IMU _ OUTMAP _ TPS.

Unsigned uint16, 2 bytes in total, little-endian mode storage, in mS.

3. Data analysis

When working in query or automatic data output mode

CMD	Meaning/example	Explain
0X04	Query data command 68 04 00 04 08	Data field (0 byte) No data domain command When working in the query mode (data output cycle is 0), use these commands to query the data of the current OUTMAP data item.
0X84	Sensor Response 68 1D 00 84 7F 1D 13 00 DD 8B FF 17 FC FB FF FC FF 01 00 FF FF 00 00 F6 0A 60 10 98 4E 14	Data domain OUTMAP _ EULER: 0x01 represents Euler angle; OUTMAP _ ACCEL: 0x02 indicates acceleration; OUTMAP _ GYRO: 0x04 indicates the angular velocity of gyro; OUTMAP _ TC: 0x08 indicates temperature; OUTMAP _ BAT: 0x10 indicates the battery level; OUTMAP _ STS: 0x20 indicates IMU status; OUTMAP _ TPS: 0x40 represents time; OUTMAP = 0x7F: indicates that all the above data items are output, and the corresponding data

BDA450 Bluetooth Dynamic Attitude Sensor

			items are output in byte order from high to low.		
Byte Offset	No. Format	name	Content (HEX)	bytes	
0	INT8U	command	68	1	A frame head indicates that start of a frame;
1	INT8U	Length	1D	1	Indicates that the data field length is 29 bytes
2	INT8U	Address	00	1	The address is always 0
3	INT8U	Data type	84	1	Indicate the output data of the default data output item MAP or query the default output data
4	INT8U	Out Map	7F	1	Data output item
5	INT16	Roll	131D	2	0x131D=4893/100=48.93°
7	INT16	Pitch	DD00	2	0xDD00=-8960/100=-89.60°
9	INT16	Yaw	FF8B	2	0xFF8B = -75/100=-0.75°
11	INT16	Acc_X	FC17	2	0xFC17=-1001/1000=-1.001g
13	INT16	Acc_Y	FFFB	2	0xFFFFB=-5/1000=-0.005g
15	INT16	Acc_Z	FFFC	2	0xFFFFC=-4/1000=-0.004g
17	INT16	Gyro_X	0001	2	0x0001=1/100=0.01deg/sec
19	INT16	Gyro_Y	FFFF	2	0xFFFF=-1/100=-0.01deg/sec
21	INT16	Gyro_Z	0000	2	0x0000=0/100=0.00deg/sec
23	INT16	Tc	0AF6	2	0x0AF6=2806/100=28.06°C
25	INT8U	Bat	60	1	0x60=90%
26	INT8U	Status	10	1	B ₇ B ₆ B ₅ B ₄ = 0001: vertical upward installation measurement mode; B ₃ B ₂ = 00: BIT succeeded;
27	INT16U	TPS	4E98	2	0x4E98=20120mS
29	INT8U	Checksum	14	1	

The host queries the specified data item MAP data output:

CMD	Meaning/example	Explain
0X03	Query the specified OUTMAP data item output 68 05 00 03 07 0F	Data field (1byte) The data field is the or of the following; The following data items are currently supported: OUTMAP _ EULER: 0x01 represents Euler angle; OUTMAP _ ACCEL: 0x02 indicates acceleration; OUTMAP _ GYRO: 0x04 indicates the angular velocity of gyro; OUTMAP _ TC: 0x08 indicates temperature; OUTMAP _ BAT: 0x10 indicates the battery level; OUTMAP _ STS: 0x20 indicates IMU status; OUTMAP _ TPS: 0x40 represents time; 0x07: Euler angle, angular velocity of accelerometer and gyro;
0X83	Sensor Response E.g: 68 17 00 83 07 4C E1 4D F8	Data Domain See the following for specific data output

			21 F8 A1 FE 97 03 51 FF 5C FF		analysis.
			F1 00 D6 FF D7		
Byte Offset	No. Format	name	Content (HEX)	bytes	
0	INT8U	command	68	1	A frame head indicates that start of a frame;
1	INT8U	Length	17	1	Indicates that the data field length is 23 bytes
2	INT8U	Address	00	1	The address is always 0
3	INT8U	Data type	83	1	Indicates the output data of the specified data output item MAP
4	INT8U	Out Map	07	1	Data output item, representing Euler angle, accelerometer and gyroscope angular velocity
5	INT16	Roll	E14C	2	0xE14C=-7860/100=-78.60 degree
7	INT16	Pitch	F84D	2	0xF84D=-1971/100=-19.71 degree
9	INT16	Yaw	F821	2	0xF821 = -2015/100=-20.15 degree
11	INT16	Acc_X	FEA1	2	0xFE A1=-351/1000=-0.351g
13	INT16	Acc_Y	0397	2	0x0397=919/1000= 0.919g
15	INT16	Acc_Z	FF51	2	0xFF51=-175/1000=-0.175g
17	INT16	Gyro_X	FF5C	2	0xFF5C=164/100=1.64deg/sec
19	INT16	Gyro_Y	00F1	2	0x00F1=241/100=2.41deg/sec
21	INT16	Gyro_Z	FFD6	2	0xFFD6=-42/100=-0.42deg/sec
23	INT8U	Check sum	D7	1	

Appendix: Data parsing routines

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

```
float BDA_Angle_UnPack(unsigned char * pbuf)
{
    short int int16_data;
    float angle;
    int16_data = *((short int*)(pbuf));
    angle = (float)int16_data / 100.0f;
    return angle;
}
```

```
float BDA_Accel_UnPack(unsigned char * pbuf)
{
    short int int16_data;
    float acc;
    int16_data = *((short int*)(pbuf));
    acc = (float)int16_data/1000.0f;
    return acc;
}
```

```
}

float BDA_Gyro_UnPack(unsigned char * pbuf)
{
    short int int16_data;
    float gyro;
    int16_data = *((short int*)(pbuf));
    gyro = (float)int16_data/100.0f;
    return gyro;
}

unsigned char GetMapData(unsigned char *pxdata)
{
    unsigned char map, datalen=0;
    map = pxdata[0];
    datalen++;
    if (map&IMU_OUTPUTMASK_EULER)
    {
        imuData.phi = BDA_Angle_UnPack(pxdata + datalen);
        datalen += 2;
        imuData.theta = BDA_Angle_UnPack(pxdata + datalen);
        datalen += 2;
        imuData.psi = BDA_Angle_UnPack(pxdata + datalen);
        datalen += 2;
    }
    if (map&IMU_OUTPUTMASK_ACCEL)
    {
        imuData.accx= BDA_Accel_UnPack(pxdata + datalen);
        datalen += 2;
        imuData.accy = BDA_Accel_UnPack(pxdata + datalen);
        datalen += 2;
        imuData.accz = BDA_Accel_UnPack(pxdata + datalen);
        datalen += 2;
    }
    if (map&IMU_OUTPUTMASK_GYRO)
    {
        imuData.groyx = BDA_Gyro_UnPack(pxdata + datalen);
        datalen += 2;
        imuData.groyy = BDA_Gyro_UnPack(pxdata + datalen);
        datalen += 2;
        imuData.groyz = BDA_Gyro_UnPack(pxdata + datalen);
        datalen += 2;
    }
    if (map&IMU_OUTPUTMASK_TEMPTC)
    {
        imuData.imu_tc = BDA_Gyro_UnPack(pxdata + datalen);
        datalen += 2;
    }
    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 short int*)(pxdata + datalen));  
        datalen += 2;  
    }  
    return datalen;  
}  
  
void USART_68_Command_Check(unsigned char *pxdata)  
{  
    unsigned char cmd_length;  
    unsigned char cmd;  
    unsigned char map;  
    cmd_length = *(pxdata+1);  
    addr = *(pxdata + 2);  
    cmd = *(pxdata + 3);  
    switch (cmd)  
    {  
        case 0x84:  
        {  
            cntPerOneSec++;  
            map = *(pxdata + 4);  
            if (cmd_length != (GetMapData(pxdata + 4)+4))  
            {  
                return;  
            }  
            break;  
        }  
        Default:  
        Break;  
    }  
}
```