GL720 IMU Dynamic Attitude Gauge

Product manual

V2.7-20250623

Technical Highlights

❖ Real-time output of horizontal attitude angle;

❖ Three-axis angular rate signal output;

❖ Axial acceleration measurement in the forward direction;

❖ Supports configurable dual RS232/RS485 interfaces;

❖ Compact structure and lightweight;

❖ High long-term working stability and long service life.

Product Introduction

The GL720 is a small IMU dynamic attitude sensor developed by RUICOM Technology using MEMS inertial measurement technology. By dynamically calculating the gyroscope angular rate signal, it can output the object's horizontal azimuth, angular rate, and body acceleration along the forward axis in real time. This product incorporates a self-developed inertial navigation algorithm, constructing a Kalman filter model to provide real-time feedback and correction of system errors, effectively suppressing divergence and significantly reducing the impact of short-term gyroscope drift on measurement accuracy.

Designed specifically for applications such as robot vehicle, AGV (Automated Guided Vehicle) orientation guidance, and Model Aircraft Attitude Control, this product overcomes the limitations of traditional magnetic guidance methods that require laying magnetic strips. It eliminates the need for deploying a large number of magnetic strips on the site, making it an ideal navigation component for next-generation robots to achieve automatic line-following navigation.

Application Scope

❖ AGV Transport Vehicles

❖ 3D Virtual Reality

❖ Industrial Automation Control

❖ Vehicle Navigation Systems

❖ Platform Stability Control

❖ Robotics Technology

Performance Parameters

GL720		Parameters
Azimuth measurement axial direction		Z-axis azimuth angle (±180)°
Acquisition bandwidth		>100Hz
Nonlinearity		0.1% of FS
Accelerometer	Acceleration range	±4g
	Acceleration resolution	0.001g
	Acceleration accuracy	5mg
	Bias instability (Allan)	0.05mg
	Velocity random walk coefficient (Allan)	0.015m/s/sqrt(h)
	Bias stability (10s mean)	0.15mg
Gyroscope	Gyro angular velocity range	±250°/s
	Angle random walk coefficient (Allan)	0.30°/sqrt(h)
	Zero bias stability (10s smooth)	10°/h
Start-up time		AGV Application: 5~6s
		AGM Application: 1.5s
Power supply voltage		9~36VDC / 5VDC (optional)
Operating current		30mA(12V)
Operating temperature		-40 ~ +80℃
Storage temperature		-40 ~ +85℃
Vibration		5g~10g
Shock		200g pk，2ms，½sine
Service life		10 years
Output rate		5Hz~200Hz configurable
Output signal		RS232 or RS485
Insulation resistance		≥100 megohms
Shock resistance		100g@11ms, triaxial (half-sine wave)
Vibration resistance		10grms / 10~1000Hz
Waterproof rating		IP67
Weight		≤117g
Note: The AGV application startup time is 5 seconds of inactivity.		

Ordering and Selecting

Model	Power Supply		Signal Output	Communication Protocol			Application Type
GL720	L: 5VDC		232: RS232	STD: Standard Protocol	1: Standard Format 1 Output		AGV : AGV Application
	W: 9~36VDC	-	485: RS485		2: Standard Format 2 Output	-	AGM : Agricultural machinery application
			TTL: TTL Level		3: Standard Format 3 Output		
					4: 9-Axis Output		
					5: 9-Axis Output + Timestamp		
				MB: MODBUS-RTU Protocol			

E.g: GL720L-232STD1-AGV:5VDC power supply / RS232 output / Standard protocol standard format 1 output / AGV application.

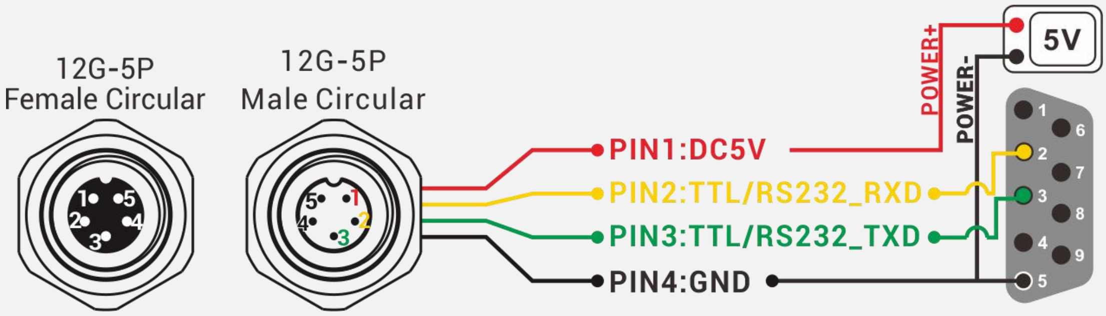
Note: AGM does not currently have the MODBUS protocol.

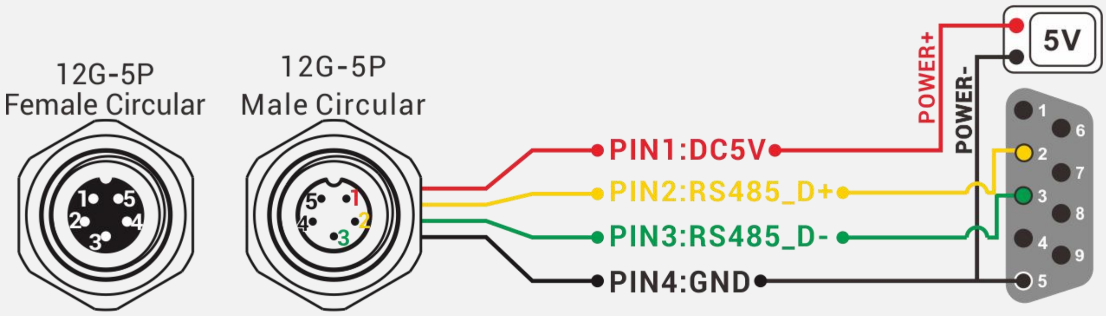
Note:

1: Standard Format 1 Output: (Z-axis angular rate + Y-axis forward acceleration + Z-axis yaw angle)
2: Standard Format 2 Output: (X-axis left/right acceleration + Y-axis forward acceleration + Z-axis yaw angle)
3: Standard Format 3 Output: (Z-axis angular rate + X-axis left/right acceleration + Y-axis forward acceleration + Z-axis yaw angle)
4: 9-axis Output: (Attitude angle + 3-axis acceleration + 3-axis gyroscope speed)
5: 9-axis Output with timestamp: (Attitude angle + 3-axis acceleration + 3-axis gyroscope speed + timestamp)

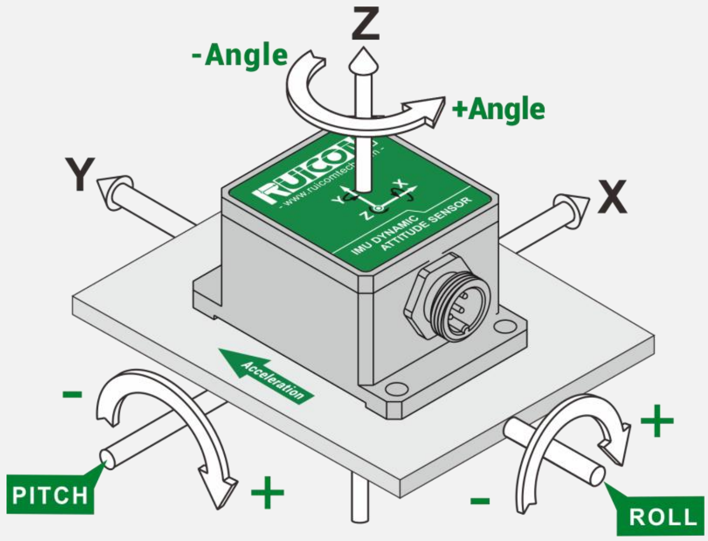
PIN Definition

PIN/Thread Color	Definition	Description
PIN1/Brown	5V/9~36V	VCC
PIN2/White	RS232_RXD/TTL_RXD/485_D+	Signal Output
PIN3/Blue	RS232_TXD/TTL_TXD/485_D-	Signal Output
PIN4/Black	GND	GND
PIN5/Gray	NC	NC

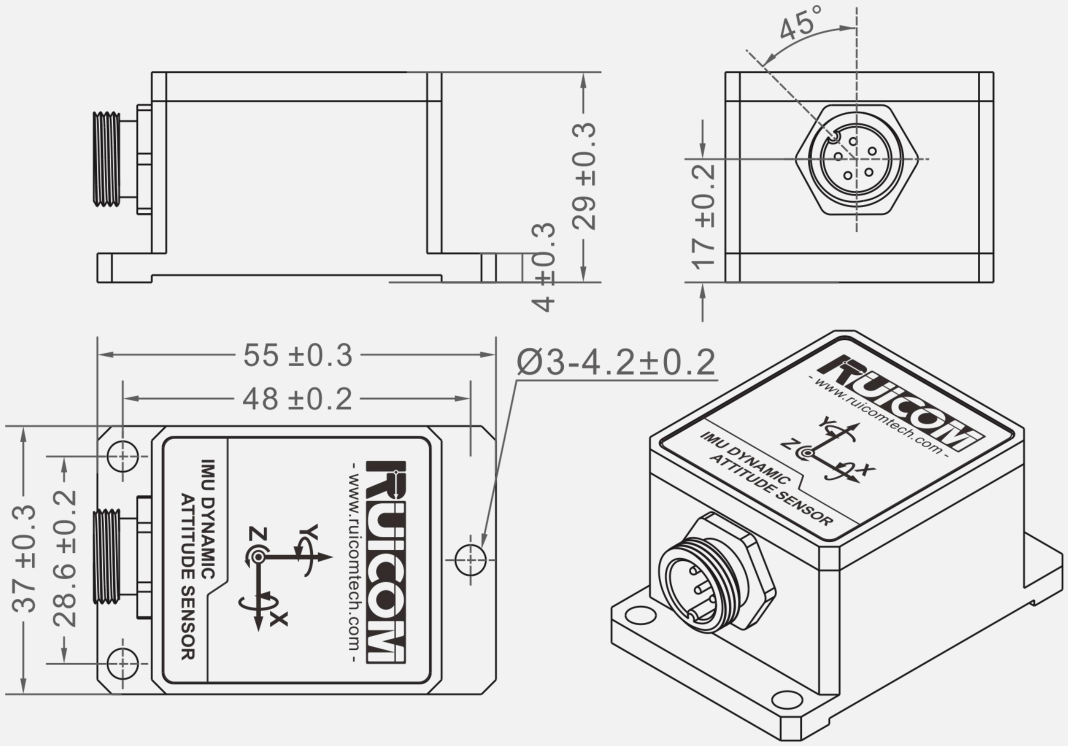




Installation measurement diagram



Product Dimensions



Housing Dimensions: L55.5×W37×H29mm
Recommended screws: 3 M4 screws

Installation Precautions

1. The angle gauge should be installed at the center of the object being measured to minimize the impact of linear acceleration on measurement accuracy, as shown in the diagram below.

The product is installed in the central position inside the AGV vehicle.

2. During installation, the mounting surface of the angle gauge should be parallel to the surface of the target object, minimizing the influence of dynamic forces and acceleration on the angle gauge. Incorrect installation will lead to measurement errors, paying particular attention to the following:
3. The mounting surface of the angle gauge must be tightly, flat, and stable against the surface being measured. Uneven mounting surfaces can easily cause angular errors in the angle measurement.
4. The axis of the angle gauge must be parallel to the axis of the object being measured, and the two axes should ideally not form an angle.
5. Do not shake the product violently during use. Avoid violent vibrations during use and keep it away from vibration sources (if unavoidable, install a vibration damping device) to prevent affecting the product's measurement accuracy.
6. During use, avoid rapid acceleration, sudden stops, sharp turns, and other movements with angular velocities exceeding 300°/s to prevent affecting the product's measurement accuracy.

Serial communication protocol

1-1.Data Frame Format

(8 data bits, 1 stop bit, no parity, default speed 115200)

Identifier (1byte)	Data Length (1byte)	Address Code (1byte)	Command Word (1byte)	Data Field	Checksum (1byte)
68h					

Identifier: Fixed at 68H;
Data Length: Length from the data length to the checksum (including the checksum);
Address Code: Address of the acquisition module, default is 00;
Data Field varies according to the content and length of the command word;
Checksum: The sum of the data length, address code, command word, and data field, without considering carry.

Usage Notes: Because this product needs to build an internal attitude calculation model during startup, the AGV needs to remain stationary for 5-6 seconds during startup. If the product moves during these 5-6 seconds, the timer will restart for another 5-6 seconds. After startup is complete, data packets will be automatically output; no data packets will be output during the initial 5-6 seconds of startup. (AGM applications do not require 5-6 seconds of stillness during startup.)

1-2.Command word parsing

Description	Meaning/Example	Description
0X04	**command to read angle data** E.g: **68 04 00 04 08**	Data Field (0 bytes) No data field command
0X84	Automatic Angle Output by	Data Field (9 bytes) in BCD format

	Sensor E.g: **68 0D 00 84 10 50 23 00 23 04 01 80 00 BC**	**10 50 23:** 3 characters represent Z-axis angular velocity = **-50.23(°/s)** **00 23 04:** 3 characters represent forward acceleration = **+2.304(g)** **01 80 00 :** 3 characters represent Z-axis azimuth = **+180.00 (°)** Data Explanation: In each 3-byte sequence, the high-order bit of the first byte is "1" for negative and "0" for positive. Acceleration is resolved to 3 decimal places, while azimuth and angular velocity are resolved to 2 decimal places. BC: Checksum, the hexadecimal sum of all data, excluding the prefix 68. If there is a carry-over, only the least significant bit is used.
0X0C	**Set Sensor Output Mode** Automatic Output: The sensor automatically outputs the angle after power-on. **AGV default output frequency is 25Hz, and AGM default output frequency is 100Hz.** (This function can be remembered after power failure) E.g: **68 05 00 0C 03 14** **Set to 25Hz output**	Data Field 00: Query 01: 5Hz Auto Output Mode 02: 15Hz Auto Output Mode 03: 25Hz Auto Output Mode 04: 35Hz Auto Output Mode 05: 50Hz Auto Output Mode 06: 100Hz Auto Output Mode 07: 200Hz Auto Output Mode
0X8C	Sensor Response Command E.g: **68 05 00 8C 00 91**	Data Field (1 byte) The number in the data field represents the sensor's response result: 00: Success FF: Fail
0X0B	**Set Baud Rate** **E.g: *68 05 00 0B 03 13*** This command setting requires a power failure and restart to take effect, and the power failure save function is also enabled.	Data Field (1 byte) Baud Rate: 02 represents 9600 03 represents 19200 04 represents 38400 **05 represents 115200 (factory default)** 06 represents 230400 07 represents 256000
0X8B	Sensor Response Command E.g: **68 05 00 8B 90**	Data Field (1 byte) The number in the data field represents the sensor's response result: 00: Success FF: Fail
0X28	**Azimuth Zeroing Command** When the azimuth angle has an error after long-term operation, this command can be sent to zero the azimuth angle output. E.g: **68 04 00 28 2c**	Data Field None

0X28	Sensor response command E.g: **68 05 00 28 00 2D**	Data Field (1 byte) The number in the data field represents the sensor's response result 00: Success FF: Fail

1-3 Detailed Output Format Table

1: Standard Format 1 Output

SOF	0x68(1 byte)
Length	0x0D(1 byte)
Address	0x00(1 byte)
Payload Contents	See below

Byte Offset	Number Format	name	content	bytes	
0	INT8U	command	0x84	1	Data representation
1	INT8U	Gyro_Z	Z-axis Angular velocity	3	10 05 23: 3 characters represent -5.23°/S 00 05 23: 3 characters represent +5.23°/S
4	INT8U	ACC _Y	Forward acceleration	3	00 10 00: 3 characters represent +1.000g 10 10 00: 3 characters represent -1.000g
7	INT8U	YAW	Azimuth	3	11 60 00: 3 characters represent -160.00° 01 60 00: 3 characters represent +160.00°
10	INT8U	Checksum	Checksum	1	

2: Standard Format 2 Output

SOF	0x68 (1 byte)
Length	0x0D (1 byte)
Address	0x00 (1 byte)
Payload Contents	See below:

Byte Offset	Number Format	name	content	bytes	
0	INT8U	command	0x84	1	Data representation
1	INT8U	ACC_X	Left and right body acceleration	3	00 00 50: 3 characters represent +0.050g(right) 10 00 50: 3 characters represent -0.050g(left)
4	INT8U	ACC_Y	Forward body acceleration	3	00 10 00: 3 characters represent +1.000g(front) 10 10 00: 3 characters represent -1.000g (back)
7	INT8U	YAW	Azimuth angle	3	11 60 00: 3 characters represent -160.00°(clockwise) 01 60 00: 3 characters represent +160.00°(counter-clockwise)
10	INT8U	Checksum	Checksum	1	

3: Standard Format 3 Output

SOF	0x68 (1 byte)				
Length	0x10 (1 byte)				
Address	0x00 (1 byte)				
Payload Contents	See below				
Byte Offset	Number Format	name	content	bytes	
0	INT8U	command	0x84	1	Data representation
1	INT8U	Gyro_Z	Z-axis angular velocity	3	10 05 23: 3 characters represent -5.23°/S 00 05 23: 3 characters represent +5.23°/S
4	INT8U	ACC_ X	left and right bodies acceleratio n	3	00 00 50: 3 characters represent +0.050g(right) 10 00 50: 3 characters represent -0.050g(left)
7	INT8U	ACC_ Y	forward body acceleratio n	3	00 10 00: 3 characters represent +1.000g(front) 10 10 00: 3 characters represent -1.000g (back)
10	INT8U	YAW	azimuth	3	11 60 00: 3 characters represent -160.00°(clockwise) 01 60 00: 3 characters represent +160.00°(counter-clockwise)
13	INT8U	Checksum	checksum	1	

4: 9-axis output: attitude angle + 3-axis acceleration + 3-axis gyroscope speed;

SOF	0x68 (1 byte)				
Length	0x1F (1 byte)				
Address	0x00 (1 byte)				
Payload Contents	See below				
Byte Offset	Number Format	name	content	bytes	
0	INT8U	command	0x84	1	Data representation
1	INT8U	ROLL	Roll angle	3	10 50 23: 3 characters represent -50.23°
4	INT8U	PITCH	Pitch angle	3	01 60 00: 3 characters represent +160.00°
7	INT8U	YAW	Yaw angle	3	11 60 00: 3 characters represent -160.00°
10	INT8U	ACC X	X-axis acceleration	3	00 23 04: 3 characters represent acceleration +2.304g
13	INT8U	ACC Y	Y-axis acceleration	3	10 23 04: 3 characters represent acceleration -2.304g
16	INT8U	ACC Z	Z-axis acceleration	3	10 23 04: 3 characters represent acceleration -2.304g
19	INT8U	Gyro_ X	X-axis gyroscope	3	10 50 23: 3 characters represent -50.23°/S

22	INT8U	Gyro_ Y	Y-axis gyroscope	3	01 80 00: 3 characters represent +180.00°/S
25	INT8U	Gyro_ Z	Z-axis gyroscope	3	00 50 23: 3 characters represent +50.23°/S
28	INT8U	Checksum	Checksum	1	

5: 9-axis + timestamp output: attitude angle + 3-axis acceleration + 3-axis gyroscope speed + timestamp;

SOF	0x68 (1 byte)
Length	0x23 (1 byte)
Address	0x00 (1 byte)
Payload Contents	See below

Byte Offset	Number Format	name	content	bytes	
0	INT8U	command	0x84	1	Data representation
1	INT8U	ROLL	Roll angle	3	10 50 23: 3 characters represent -50.23°
4	INT8U	PITCH	Pitch angle	3	01 60 00: 3 characters represent +160.00°
7	INT8U	YAW	Yaw angle	3	11 60 00: 3 characters represent -160.00°
10	INT8U	ACC X	X-axis acceleration	3	00 23 04: 3 characters represent acceleration +2.304g
13	INT8U	ACC Y	Y-axis acceleration	3	10 23 04: 3 characters represent acceleration -2.304g
16	INT8U	ACC Z	Z-axis acceleration	3	10 23 04: 3 characters represent acceleration -2.304g
19	INT8U	Gyro_ X	X-axis gyroscope	3	10 50 23: 3 characters represent -50.23°/S
22	INT8U	Gyro_ Y	Y-axis gyroscope	3	01 80 00: 3 characters represent +180.00°/S
25	INT8U	Gyro_ Z	Z-axis gyroscope	3	00 50 23: 3 characters represent +50.23°/S
28	INT8U	tStam[3]	32-25 bit	1	Uint32_t TimeStamp(mS) ; TimeStamp = (tStam[3]<<24) | (tStam[2]<<16) | (tStam[1] << 8) | tStam[0];
29	INT8U	tStam[2]	24-17 bit	1	
30	INT8U	tStam[1]	16-9 bit	1	
31	INT8U	tStam[0]	8-1 bit	1	
32	INT8U	Checksum	Checksum	1	

Note: The timestamp is the running time of angle calculation after the IMU is powered on, uint32 (4 bytes), in milliseconds (ms).

◆ MODBUS-RTU Data Frame Format

1. RTU Mode Communication Parameters: Baud rate 115200 bps, Data frame: 1 start bit, 8 data bits, even parity, 1 stop bit.

Please carefully read the following items before use:

1) The MODBUS protocol stipulates that there should be at least 3.5 bytes of time between two data frames (e.g., at a baud rate of 9600, this time is 3.5 × (1/9600) × 11 = 0.004s). However, to leave sufficient margin, this sensor increases this time to 10ms. Therefore, please leave at least a 10ms time interval between each data frame.

Master sends command -- 10ms idle -- Slave replies -- 10ms idle -- Master sends command...

2) The MODBUS protocol specifies the content related to broadcast address ---- 0 This sensor can also receive broadcast address content, but will not reply. Therefore, broadcast address 0 can be used for the following purposes, for reference only:

1. Set the address of all tilt sensors of this model connected to the bus to a certain address.
2. Set the azimuth angle of all tilt sensors of this model connected to the bus to zero.

3) To improve system reliability, commands such as setting the address, setting the baud rate, and changing the parity bit must be sent twice consecutively to be effective. "Sent twice consecutively" means that both transmissions are successful (the slave device responds each time), and the two responses must be consecutive; that is, the master device cannot insert other data frames in between the two responses. Otherwise, the commands must be retransmitted twice. The setting process is as follows:

Send the address setting command -- wait for the slave device to send the success command -- (no other commands can appear) Send the address setting command again -- wait for the slave device to send the success command -- Modification successful.

4) Changing the parity bit requires a power-on restart to be effective.

2. Read angle data

MODBUS Function Code 03H

Host query command:		Slave Response:		
Sensor Address	01H	Sensor Address	01H	
Function Code	03H	Function Code	03H	
Access Register Start Address	00H	Data Length (12 bytes)	0CH	
	02H	Data Word 1 (High 8 bits)	F3H	Z-axis angular velocity data (azimuth rate)
Data Length 6 Words	00H	Data Word 1 (Low 8 bits)	49H	
	06H	Data Word 2 (High 8 bits)	02H	
CRCLH	6408H	Data Word 2 (Low 8 bits)	00H	
		Data Word 3 (High 8 bits)	1DH	Y-axis acceleration data (forward)
		Data Word 3 (Low 8 bits)	4EH	
		Data Word 4 (High 8 bits)	00H	
		Data Word 4 (Low 8 bits)	00H	
		Data Word 5 (High 8 bits)	02H	Z-axis azimuth data
		Data Word 5 (Low 8 bits)	4FH	
		Data Word 6 (High 8 bits)	00H	
		Data Word 6 (Low 8 bits)	00H	
		CRCLH	501CH	

Example of using the command to read measurement data:

Host sends		01H	03H	00H	02H	00H	06H	64H	08H
Slave Response		01H	03H	0CH	F3H	49H	02H	00H	1DH
	4EH	00H	00H	02H	4FH	00H	00H	50H	1CH

Note: The data fields of the slave response frame are F3H/49H/02H/00H/1DH/4EH/00H/00H/02H/4FH/00H/00H;
The Z-axis angular velocity data (azimuth angular velocity) is represented by bytes 1-4 of the data field, the Y-axis acceleration data (forward) by bytes 5-8, and the Z-axis azimuth data by bytes 9-12, with the least significant byte first.
The Z-axis angular velocity data (azimuth angular velocity) is represented using a point-count method, where one point corresponds to 0.01°/s. 0.01 × (number of points - offset) equals the angular velocity. The offset for the angular velocity is taken as 150,000, resulting in a total of 300,000 points. Therefore, 150,000 corresponds to 0°/s, 151,000 corresponds to +10°/s, and 149,000 corresponds to -10°/s.
The Y-axis acceleration data (forward) is also represented using a point-count method, where one point corresponds to 0.001g. 0.001 × (number of points - offset) equals the acceleration. The acceleration offset is set to 20000, resulting in a total of 40000 points. Therefore, 20000 corresponds to 0g, 20100 corresponds to +0.100g, and 19900 corresponds to -0.100g.
The Z-axis azimuth data uses a point-based representation, with one point corresponding to 0.01°. 0.01 × (point count - offset) is the azimuth angle. The azimuth offset is set to 18000, resulting in a total of 36000 points. Therefore, 18000 corresponds to 0°/s, 19000 corresponds to +10°, and 17000 corresponds to -10°/s.
Taking the above data frame as an example: the data conversion process is as follows:

1) Obtain the current angle point count. Note that the least significant byte comes first. The Z-axis angular velocity data is 249F3H, the Y-axis acceleration data (forward) is 4E1DH, and the Z-axis azimuth data is 4F02H.

2) Convert to decimal: Z-axis angular velocity: 249F3H → 150003, Y-axis acceleration: 4E1DH → 19997, Z-axis azimuth: 4F02H → 20226.

3) Subtract the offset: Z-axis angular velocity: (150003 - 150000) × 0.01 = 0.03°/s; Y-axis acceleration: (19997 - 20000) × 0.001 = -0.003g; Z-axis azimuth: (20226 - 18000) × 0.01 = 22.26°.

4) Final result: Z-axis angular velocity: 0.03°/s; Y-axis acceleration: -0.003g; Z-axis azimuth: 22.26°.

2-3. Set sensor azimuth angle to zero

Modbus function code 06H

Set sensor azimuth angle zeroing command:		Slave Response:	
Sensor Address	01H	Sensor Address	01H
Function Code	06H	Function Code	06H
Access Register Start Address	00H	Register Address	00H
	10H		10H
If this word is non-zero, the azimuth angle is cleared	00H	If this word is non-zero, the azimuth angle is cleared	00H
	FFH		FFH
CRC	C84FH	CRC	C84FH

Example of setting a sensor azimuth angle zeroing command:								
Host sends	01H	06H	00H	10H	00H	FFH	C8H	4FH
Slave Response	01H	06H	00H	10H	00H	FFH	C8H	4FH

Note: 0010 is the register address. Writing 00FFH to this register (as in the example above, where 00FFH was written) will clear the current azimuth angle to zero. The last two bytes are the CRC checksum.

2-4. Set sensor address

Command to set sensor address code:		Slave Response:	
Sensor Address	01H	Sensor Address	01H
Function Code	06H	Function Code	06H
Address	00H	Register Address	00H
	11H		11H
New address of the sensor	00H	New address of the sensor	00H
	04H		04H
CRC	D80C	CRC	D80C

Commands must be sent twice consecutively to be effective.

Example of using the command to set the sensor address:								
Host sends	01H	06H	00H	11H	00H	04H	D8H	0CH
Slave Response	01H	06H	00H	11H	00H	04H	D8H	0CH

Note: 0011H is the register address, which controls the sensor address. In the example above, the sensor address has been changed to 0004H, and the last two bytes are the CRC checksum.

2-5. Set the sensor baud rate: (The factory default is 115200bps)

Command to set sensor baud rate:		Slave Response:	
Sensor Address	01H	Sensor Address	01H
Function Code	06H	Function Code	06H
Address	00H	Register Address	00H
	12H		12H
sensor baud rate	00H	sensor baud rate	00H
	A2		A2
CRC	A876	CRC	A876

A1H: 9600 A2H: 19200 A3H: 38400 A4H: 115200 A5H: 230400 A6H:256000

Commands must be sent twice consecutively to be effective.

Example of using the command to set the sensor baud rate:								
Host sends	01H	06H	00H	12H	00H	A2H	A8H	76H
Slave Response	01H	06H	00H	12H	00H	A2H	A8H	76H

Note: 0012H is the register address, which controls the sensor baud rate. In the example above, the sensor baud rate is set to 19200, and the last two bytes are the CRC checksum.

2-6. Set sensor to automatic output: (Factory default is 0Hz, query mode)

Configure the sensor to automatically output command:		Slave Response:	
Sensor Address	01H	Sensor Address	01H
Function Code	06H	Function Code	06H
Address	00H	Register Address	00H
	13H		13H
Sensor output frequency	00H	Sensor output frequency	00H
	00H		00H
CRC	780FH	CRC	780FH

XX= 00: Inquiry mode; 01: 5HZ; 02: 10HZ; 03: 20HZ; 04: 25HZ;
05: 50HZ; 06: 100HZ 07: 200Hz

Example of setting up automatic command output for sensors:								
Host sends	01H	06H	00H	13H	00H	00H	78H	0FH
Slave Response	01H	06H	00H	13H	00H	00H	78H	0FH

Configure the sensor to use a query method.

2-7. Configure the parity bit for sensor serial communication: (The factory default is even parity)

Command to set sensor parity bit:		Slave Response:	
Sensor Address	01H	Sensor Address	01H
Function Code	06H	Function Code	06H
Address	00H	Register Address	00H
	18H		18H
Parity bit of the sensor	00H	Parity bit of the sensor	00H
	02H		02H
CRC	880C	CRC	880C

Set parity bit:
0x0000: PARITY_NONE (No parity bit)
0x0001: PARITY_ODD (Odd parity bit)
0x0002: PARITY_EVEN (Even parity bit)
Commands must be sent twice consecutively, and a power cycle is required for them to take effect.

Example of using the command to set the sensor parity bit:								
Host sends	01H	06H	00H	18H	00H	02H	88H	0CH
Slave Response	01H	06H	00H	18H	00H	02H	88H	0CH

Note: 0018H is the register address, which controls the sensor's parity bit. In the example above, the sensor's parity bit is set to even parity, and the last two bytes are the CRC checksum.