RION TECH
en.rion-tech.net


V1.0

HIGH-PRECISION DYNAMIC TILT SENSOR

# RION HDA436T

## Technical Manual

## PRODUCTION EXECUTION STANDARD REFERENCE

○ Enterprise quality system standard: ISO9001: 2015 standard (certification number: 128101) ;
○ The intellectual property management system complies with the standard: GB / T 29490-2013 (Certificate No.: 18117IP1529R0S) ;
○ High-tech enterprise (Certificate No .: GR201844204379) ;
○ Angle sensor production standard: SJ20873-2003 General specification for inclinometer and spirit level
○ Gyro acceleration test standard: QJ 2318-92 gyro accelerometer test method ;
○ Software development reference standard: GJB 2786A-2009 ;
○ Product environmental test detection standard: GJB150 ;
○ Electromagnetic immunity test standard: GB / T 17626 ;
○ Revision date: 2020-3-30

Note: Product functions, parameters, appearance, etc. will be adjusted with technical upgrades, please contact the Division's pre-sales to confirm when purchasing.

# *HDA436T* HIGH-PRECISION DYNAMIC TILT SENSOR


## ▶ PRODUCT DESCRIPTION

HDA436T， A new generation of digital MEMS dynamic tilt sensor launched by RION Technology. It can measure the attitude parameters (roll, pitch, and azimuth) of a moving carrier, and suitable for tilt angle measurement under motion or vibration.

HDA436T has a built-in high-precision acceleration and gyro sensor, and integrates the Carman filter algorithm, which can measure the real-time motion data of the carrier under motion or vibration. The signal output modes such as RS485 / 232 / TTL are optional, and the scalability is strong.

This product adopts non-contact measuring principle, which can output the current attitude and inclination in real time. It is simple to use and does not need to retrieve the relative changes of the two surfaces.

Internally integrated high-precision AD and high-precision gyro units to compensate for non-linear, orthogonal coupling, temperature drift and centrifugal acceleration in real time; greatly eliminate centrifugal errors caused by motion acceleration interference, improve product dynamic measurement accuracy; adapt to long-term complex movements Work in places and harsh environments.

The product is a dynamic and static dual-mode measurement sensor with strong resistance to external electromagnetic interference. It is an ideal choice for industrial automation control and military-civilian dual-use measurement posture.

## ▶ KEY FEATURES

★ Range (Roll angle ± 180 °, pitch angle ± 90 °, azimuth angle ± 180 °)
★ Dynamic accuracy: ± 0.3 °
★ DC 9~6V wide voltage input
★ Wide temperature operation -40~+ 85 ℃
★ Resolution 0.01 °
★ IP67 protection level
★ High anti-vibration performance> 2000g
★ Size: L72× W47× H22.5mm

## ▶ PRODUCT APPLICATION

★ Railway locomotive monitoring
★ Oil drilling equipment
★ Shield pipe jacking application
★ Geological equipment tilt monitoring
★ Underground drilling rig attitude navigation
★ Ship navigation attitude measurement
★ Shield pipe jacking application
★ Satellite communication vehicle attitude detection
★ Orientation measurement based on inclination angle
★ Various engineering machinery inclination measurement

## ▶ TECHNICAL PARAMETERS

| HDA436T | CONDITION | INDEX |
|---|---|---|
| Measuring range | \ | Roll ± 180 °, pitch ± 90 °, azimuth ± 180 ° (initial value at power-down is 0 °) |
| Measuring axis | \ | X axis / Y axis / Z axis |
| Resolution | \ | 0.01° |
| Static accuracy | @25℃ | ± 0.05° |
| Dynamic accuracy | @25℃ | ± 0.3° |
| Bias Instability (Allan) | \ | 8°/h |
| Zero point temp. coefficient | -40 ~ 85° | ±0.01°/℃ |
| Sensitivity temp. coefficient | -40 ~ 85° | ≤100ppm/℃ |
| Power-on startup time | 1S | |
| Response time | 0.01S | |
| Output signal | RS485/RS232/TTL | |
| Electromagnetic compatibility | According to EN61000 and GBT17626 | |
| MTBF | ≥ 50000 hours / time | |
| Insulation resistance | ≥100 Megohm | |
| Impact resistance | 100g @ 11ms, Triaxial and identical (half sine wave) | |
| Anti-vibration | 10grms、10 ~ 1000Hz | |
| Waterproof level | IP67 | |
| Cable | Standard without wiring, optional 2m M12 aviation plug with PVC unshielded cable | |
| Weight | 160g (excluding cable) | |

## ▶ ELECTRICAL PARAMETERS

| PARAMETER | CONDITION | MIN | TYP | MAX | UNIT |
|---|---|---|---|---|---|
| Supply voltage | standard | 9 | 12、24 | 36 | V |
| Working current | without load | | 60mA(12V) | | mA |
| Operating temp | | -40 | | +85 | ℃ |
| storage temp | | -40 | | +85 | ℃ |

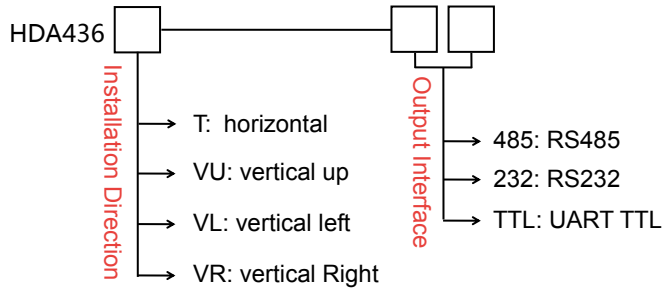
**Glossary**

Resolution : It refers to the smallest change in the measured value that can be detected and resolved by the sensor within the measurement range.

Absolute accuracy : Absolute accuracy refers to the comprehensive error of the sensor's absolute linearity, repeatability, hysteresis, zero point deviation, and horizontal axis error under normal temperature conditions.

Response time : The response time is the time required for the sensor output to reach the measured value when the sensor changes its angle once.

## ▶ ORDERING




E.g: HDA436T-485: single axis / horizontal installation / RS485 output interface.

## ▶ MECHANICAL PARAMETERS

○ Connector: M12 aviation plug 5P connector

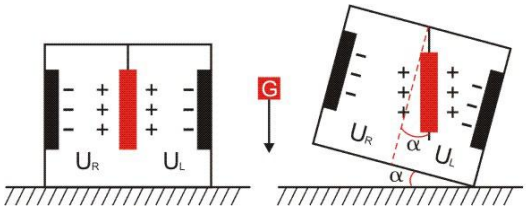
○ Protection level: IP67

○ Shell material: frosted aluminum alloy

○ Installation: Four M4 screws

## ▶ WORKING PRINCIPLE

Adopt imported core control unit and apply the principle of capacitive miniature pendulum. Using the earth's gravity principle, when the tilt unit tilts, the earth's gravity will produce a gravitational component on the corresponding pendulum, and the corresponding capacitance will change. By amplifying and filtering the capacitance, the inclination will be obtained after conversion.

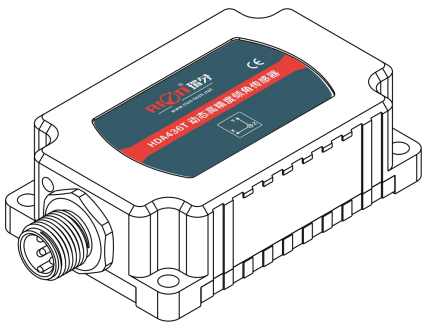


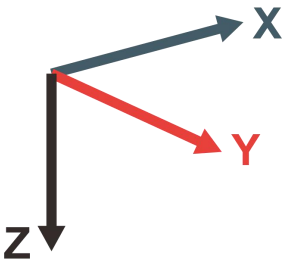

$U_R$, $U_L$ are the voltage between the left and right pole plates of the pendulum and their corresponding electrodes,when the tilt sensor is tilted, $U_R$, $U_L$ will change according to a certain rule, so $\int (U_R\ U_L)$ is a function on α of tilt angle : $\alpha = \int (U_R, U_L)$ .

## ▶ INSTALLATION AXIAL

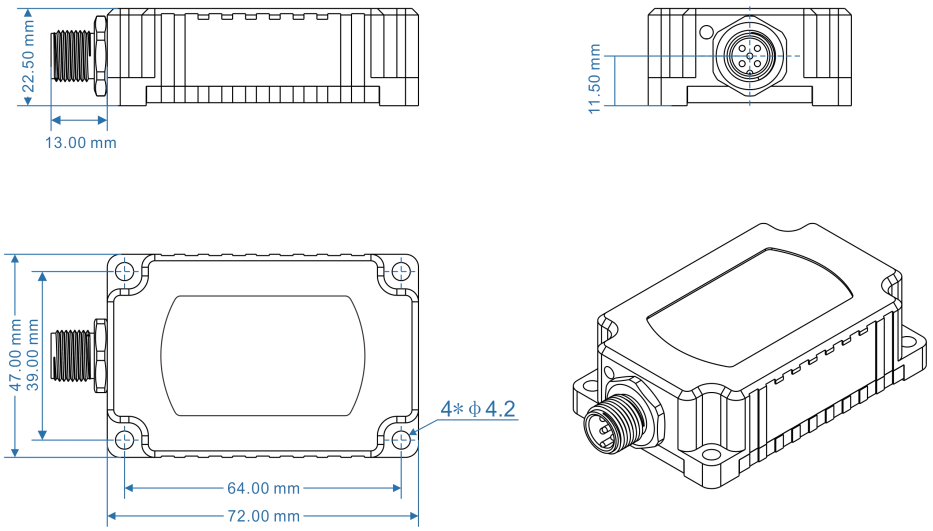
HDA436T follows the NED coordinate system, right-handed.

According to the rotation sequence of ZYX, when the positive direction of the X axis is directed to the front of the carrier, the rotation angle around the Z axis is the heading angle, the rotation angle around the Y axis is the pitch angle, and the rotation angle around the X axis is the roll angle.





## ▶ PRODUCT SIZE CHART

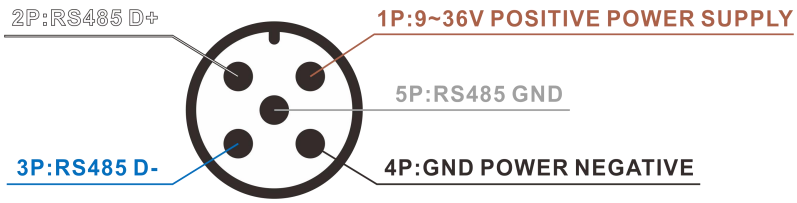



Dimension:72*47*22.5mm

## ▶ ELECTRICAL CONNECTIONS

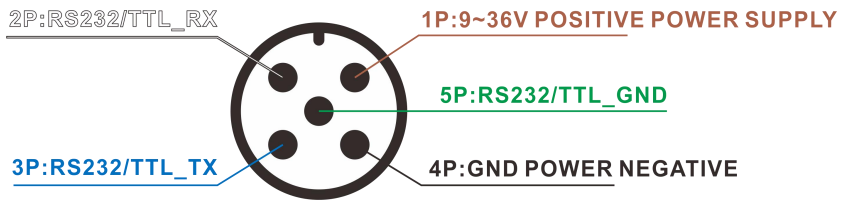
RS485 WIRING DEFINITION

| PIN | 1 PIN(BROWN) | 2 PIN(WHITE) | 3 PIN(BLUE) | 4 PIN(BLACK) | 5 PIN(GRAY) |
|---|---|---|---|---|---|
| FUNCTION | 9~36V POSITIVE POWER SUPPLY | 485 D+ | 485 D- | GND POWER NEGATIVE | 485 GND |

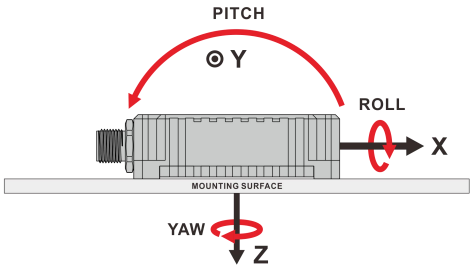



RS232/TTLWIRING DEFINITION

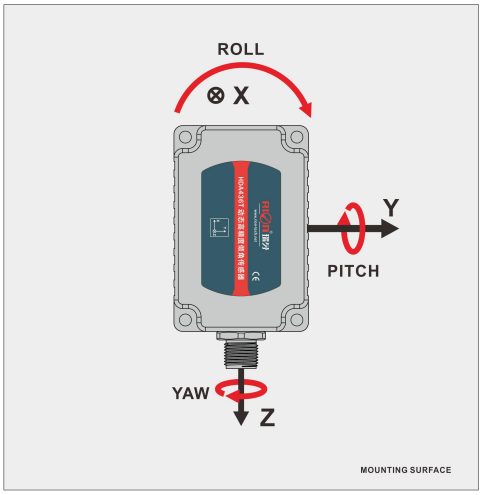
| PIN | 1 PIN(BROWN) | 2 PIN(WHITE) | 3 PIN(BLUE) | 4 PIN(BLACK) | 5 PIN(GREEN) |
|---|---|---|---|---|---|
| FUNCTION | 9~36V POSITIVE POWER SUPPLY | RS232-RX/ TTL-RX | RS232-TX/ TTL-TX | GND POWER NEGATIVE | RS232/ TTL _GND |



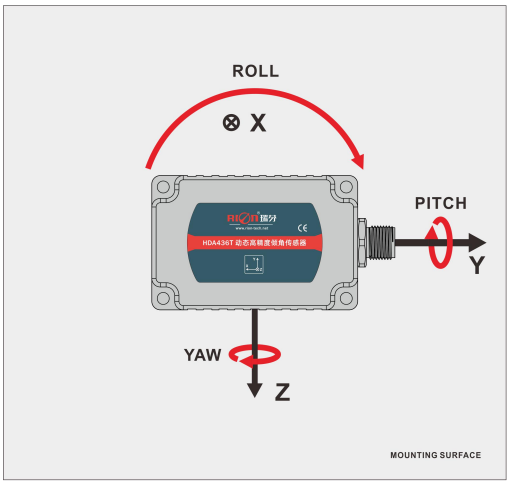

## ▶ INSTALLATION METHOD



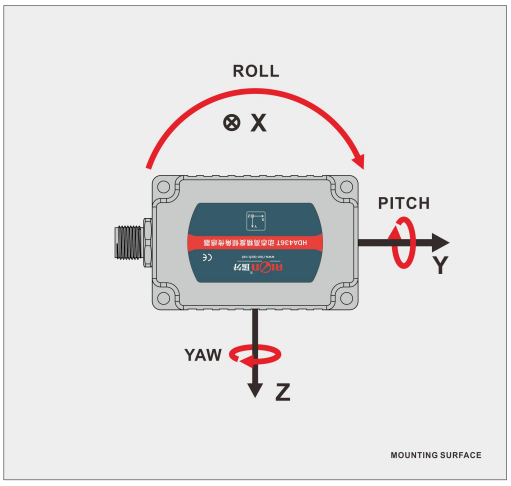

Horizontal Installation




Vertical Installation




Vertical-Left Installation




Vertical-Right Installation

## ▶ COMMUNICATION PROTOCOL

**1.0 Communication Frame Format**

| Field | Frame sync byte | Frame start byte | ADDR | CMD Command field | Length | DATAs Data field | CRC | Frame end byte |
|---|---|---|---|---|---|---|---|---|
| Number of bytes (byte) | 1 | 1 | 1 | 1 | 2 | 0 to 504 | 2 | 1 |
| Description | Sync. byte | Start of byte | Address | Command | Length of the Datas | Datas | CRC | End of Tx byte |
| Value | 0xFF | 0x02 | 0-FF | - | - | - | - | 0x03 |

Notes:
A. The data length field includes the number of data fields (Datas). MSB comes first, LSB comes after. A length of 0 indicates no data field. The longest data field is 504 bytes, and the longest frame byte is 512. Bytes.
B. The calculation of CRC starts from ADDR, including CMD, length field and data field, MSB first, LSB last, and calculate CRC according to the following function:

```
uint16 calcCRC(const uint8 *pBuffer, uint16 bufferSize)
{
      uint16 poly = 0x8408;
      uint16 crc = 0;
      uint8 carry;
      uint8 i_bits;
      uint16 j;

      for (j=0; j<bufferSize; j++)
      {
            crc = crc ^ pBuffer[j];
            for (i_bits=0; i_bits<8; i_bits++)
            {
                  carry = crc & 1;
                  crc = crc / 2;
                  if (carry)
                  {
                        crc = crc^poly;
                  }
            }
      }
      return crc;
}
```

**1.1 Data Format And Endian**

The data output of the device is in little-endian mode (such as integer and floating-point, with the low byte first and the high byte last).

**1.2 Serial Port Default**

The default baud rate of the serial port is 115200 bps, 1 start bit, 1 stop bit, no parity.

**1.3 The default output is automatic output and the output frequency is 10Hz.**

**1.4 The default address is 0. The range is 0x00-0xFE, settable. 0xFF is a universal address.**

**2.0 Communication Mode**

The sensor has two communication modes: query mode (question and answer mode) and

continuous output mode (automatic data output at a certain frequency). Question and answer query operation can also be performed in continuous output mode.

**2.1 Query Mode**

In a question and answer mode, the user sends an inquiry or setting instruction to the sensor, and the device responds to the inquiry or setting instruction one-to-one. The device responds to the query command in two formats: reply (CMD = ASK) and reply (CMD = RET). For details, refer to the command list in Chapter 4.

**2.1.1 Response**

The response frame format is as follows:

| Field | Frame sync byte | Frame start byte | ADDR | CMD | LEN (MSB) | LEN (LSB) | DATA | CRC (MSB) | CRC (LSB) | Frame end byte |
|---|---|---|---|---|---|---|---|---|---|---|
| No. of bytes (byte) | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Value | 0xFF | 0x02 | 0x00 | IMU_ ACK (0x01) | 0x00 | 0x01 | ERRO R CODE | 0xXX | 0xXX | 0x03 |

The response is a reply to the command. The execution status of the instruction is reflected by the error code. The following error code (ERROR CODE) determines the device's response to the query instruction. The following is a list of error codes;

| IMU_Error Code | Value | Description |
|---|---|---|
| IMU_NO_ERROR | 0x00 | Command has been executed correctly |
| IMU_ERROR | 0x01 | Command did not execute correctly |
| IMU_INVALID_FRAME | 0x04 | Invalid command |
| IMU_INVALID_PARAMETER | 0x09 | Invalid parameter |
| IMU_NOT_READY | 0x0A | Sensor is not ready |

Note: The error code constitutes the data field (DATA) of the ACK frame.

**2.1.2 Reply**

The reply is a one-to-one response to the user command. The CMD and data fields of different query commands are different. Please refer to the command list in Chapter 4.

**2.2 Continuous Output Mode**

The continuous output mode is an automatic output mode. The selected data items are output at a certain frequency. The output frequency can be set.

**3.0 Output Data Types**

**3.1 Output Data Types**

By using output masks (uint32) to set the output data items and format, the user can choose the following data item output:

| Valid mask | Description | sequence | support | Mask value |
|---|---|---|---|---|
| IMU_OUTPUT_EULER | Attitude angle | 1 | Support | 0x00000001 |
| IMU_OUTPUT_GYROSCOPES | Triaxial angular velocity after calibration | 2 | Support | 0x00000002 |
| IMU_OUTPUT_ACCELEROMETERS | Triaxial acceleration after calibration | 3 | Support | 0x00000004 |

**The order indicates the order of data output.**

**3.2 Data Output Format**

3.2.1 Attitude angles Roll, Pitch and Yaw, expressed in real32 (float), 12 bytes, in radians.

Storage format:

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

**3.2.2 Three-Axis Gyro Angular Rate Gx, Gy And Gz, Expressed In Real32 (Float), 12 Bytes, The Unit Is Rad.S-1 (Radian / Second).**

Storage format:

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

**3.2.3Three-Axis Acceleration Ax, Ay And Az, Expressed In Real32 (Float), 12 Bytes, The Unit Is $m.S^{-2}$。**

Storage format:

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

**3.3 Output Data Order**

According to different values of output masks (uint32), the corresponding output order will be generated.

The default output masks (uint32) = 0x00000007 is as follows:

IMU_OUTPUT_EULER | IMU_OUTPUT_GYROSCOPES | IMU_OUTPUT_ACCELEROMETERS (0x00000001 | 0x00000002 | 0x00000004) The corresponding output data sequence and storage format are as follows:

| Name | Roll | Pitch | Yaw | Gx | Gy | Gz | Ax | Ay | Az |
|---|---|---|---|---|---|---|---|---|---|
| Size (bytes) | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| | 12 | | | 12 | | | 12 | | |

The analysis of the data field is performed in the above storage format.

For example, output masks (uint32) = 0x00000001 is as follows:

IMU_OUTPUT_EULERS (0x00000001) corresponds to the output data sequence and storage format as follows:

| Name | Roll | Pitch | Yaw |
|---|---|---|---|
| Size (bytes) | 4 | 4 | 4 |
| | 12 | | |

For example, output masks (uint32) = 0x00000005 is as follows:

IMU_OUTPUT_EULER | IMU_OUTPUT_ACCELEROMETERS (0x00000001 | 0x00000004), the corresponding output data sequence and storage format are as follows:

| Name | Roll | Pitch | Yaw | Ax | Ay | Az |
|---|---|---|---|---|---|---|
| Size (bytes) | 4 | 4 | 4 | 4 | 4 | 4 |
| | 12 | | | 12 | | |

**4.0 Command List**

**4.1 Set Automatic Output Data Items**

* **IMU_SET_DEFAULT_OUTPUT_MASK------(0x50)**

Function: Used to set the output MASK (output mask (uint32)), that is, the output data type (Buffer).

Note: It does not have the power-down save function. You must execute the save setting command to have the power-down save function.

The frame format is as follows:

| Frame sync byte | Frame start byte | ADDR | CMD | LEN (MSB) | LEN (LSB) | DATA | CRC (MSB) | CRC (LSB) | Frame end byte |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 5 | 1 | 1 | 1 |
| 0xFF | 0x02 | 0x00 | 0x50 | 0x00 | 0x05 | DATA | 0xXX | 0xXX | 0x03 |

Of which: DATA part

| DATA | |
|---|---|
| Reserved.Leave to 0(bytes) | Default output mask(uint32) |
| 1 | 4 |

Reply：

**IMU_ACK**

| Frame sync byte | Frame start byte | ADDR | CMD | LEN (MSB) | LEN (LSB) | DATA | CRC (MSB) | CRC (LSB) | Frame end byte |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0xFF | 0x02 | 0x00 | 0x01 | 0x00 | 0x01 | ERRO R_CO DE | 0xXX | 0xXX | 0x03 |

**4. 2 Get Data Items For Automatic Output**

* **IMU_GET_DEFAULT_OUTPUT_MASK------(0x51)**

Function: It is used to read the output MASK (output mask (uint32)) data item, which is used for data analysis.

The frame format is as follows:

| Frame sync byte | Frame start byte | ADDR | CMD | LEN (MSB) | LEN (LSB) | DATA | CRC (MSB) | CRC (LSB) | Frame end byte |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| 0xFF | 0x02 | 0x00 | 0x51 | 0x00 | 0x00 | NULL | 0xXX | 0xXX | 0x03 |

Reply: * IMU_RET_DEFAULT_OUTPUT_MASK ------ (0x52)

Role: Returns the output mask;

The frame format is as follows:

| Frame sync byte | Frame start byte | ADDR | CMD | LEN (MSB) | LEN (LSB) | DATA | CRC (MSB) | CRC (LSB) | Frame end byte |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 4 | 1 | 1 | 1 |
| 0xFF | 0x02 | 0x00 | 0x52 | 0x00 | 0x04 | DATA | 0xXX | 0xXX | 0x03 |

Of which: DATA part

| DATA |
|---|
| Default output mask(uint32) |
| 4 |

**4.3 Continuous Data Output**

* IMU_CONTINUOUS_DEFAULT_OUTPUT ------ (0x90)

Function: Data items are continuously output.

| Frame sync byte | Frame start byte | ADDR | CMD | LEN (MSB) | LEN (LSB) | DATA | CRC (MSB) | CRC (LSB) | Frame end byte |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | See below | 1 | 1 | 1 |
| 0xFF | 0x02 | 0x00 | 0x90 | 0x00 | 0x24 | See below | 0xXX | 0xXX | 0x03 |

Data part is as follows (total length: 0x0024):

| Name | Roll | Pitch | Yaw | Gx | Gy | Gz | Ax | Ay | Az |
|---|---|---|---|---|---|---|---|---|---|
| Size (bytes) | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| | 12 | | | 12 | | | 12 | | |

| Number Field | Real32 | Float | Unit |
|---|---|---|---|
| Roll | BBFF0F9E | -0.007784 | radian |
| Pitch | 3C35E11F | 0.011101 | radian |
| Yaw | BD2E03F5 | -0.042484 | radian |
| Gx | B99D4848 | -0.000300 | Radians / second |
| Gy | 3925ACF1 | 0.000158 | Radians / second |
| Gz | 389202DA | 0.000070 | Radians / second |
| Ax | 3DE1E839 | 0.110306 | $m.S^{-2}$ |

| | | | |
|---|---|---|---|
| Ay | 3D9D7DFA | 0.076900 | $m.S^{-2}$ |
| Az | C11C1675 | -9.755483 | $m.S^{-2}$ |

**4.4 Output Mode Setting**

Output work mode

There are two modes of Mode, as follows:

IMU_QEURY_MODE (Query mold or Q & A mode): 0x00

IMU_CONTINUOUS_MODE (Continuous output mode): 0x01

The CONTINUOUS_MODE (continuous output mode) has different Divider values, corresponding to the following different output frequencies:

| Divider | OUT(Hz) |
|---|---|
| 0 | INVALID |
| 1 | 200 |
| 2 | 100 |
| 4 | 50 |
| 8 | 25 |
| 10 | 20 |
| 20 | 10 |
| 40 | 5 |
| 200 | 1 |

Note: The default is IMU_CONTLNUOUS_MODE (continuous output mode), and the frequency is 10Hz.

**4.5 Set Output Mode Command**

*** IMU_SET_CONTINUOUS_MODE------(0x53)**

Function: Set the output mode;

Note: It does not have the power-down save function. You must execute the save setting command to save.

The frame format is as follows:

| Frame sync byte | Frame start byte | ADDR | CMD | LEN (MSB) | LEN (LSB) | DATA | CRC (MSB) | CRC (LSB) | Frame end byte |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 3 | 1 | 1 | 1 |
| 0xFF | 0x02 | 0x00 | 0x53 | 0x00 | 0x03 | As Follow | 0xXX | 0xXX | 0x03 |

Of which: DATA part

| DATA | | |
|---|---|---|
| Reserved.Leave to 0(bytes) | Mode | Divider |
| 1 | 1 | 1 |
| 0 | 1 | 1 |

Note: Above Mode = 1, continuous output, Divider = 1, output frequency: 200Hz.

Reply:

IMU_ACK

| Frame sync byte | Frame start byte | ADDR | CMD | LEN (MSB) | LEN (LSB) | DATA | CRC (MSB) | CRC (LSB) | Frame end byte |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 0xFF | 0x02 | 0x00 | 0x01 | 0x00 | 0x01 | ERRO R_CO DE | 0xXX | 0xXX | 0x03 |

**4.6 Get Output Mode**

*** IMU_GET_CONTINUOUS_MODE------(0x54)**

Function: used to read the current mode;

The frame format is as follows:

| Frame sync byte | Frame start byte | ADDR | CMD | LEN (MSB) | LEN (LSB) | DATA | CRC (MSB) | CRC (LSB) | Frame end byte |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| 0xFF | 0x02 | 0x00 | 0x54 | 0x00 | 0x00 | NULL | 0xXX | 0xXX | 0x03 |

Reply:

* IMU_RET_CONTINUOUS_MODE ------ (0x55)

Function: Used to return Mode and Divider;

The frame format is as follows:

| Frame sync byte | Frame start byte | ADDR | CMD | LEN (MSB) | LEN (LSB) | DATA | CRC (MSB) | CRC (LSB) | Frame end byte |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 2 | 1 | 1 | 1 |
| 0xFF | 0x02 | 0x00 | 0x55 | 0x00 | 0x02 | DATA | 0xXX | 0xXX | 0x03 |

Of which: DATA part

| DATA | |
|---|---|
| Mode(uint8) | Divider(uint8) |
| 1 | 1 |

**4.7 Setting The Baud Rate**

The baudrate is a 32-bit integer. The valid values are: 9600, 19200, 38400, 57600, 115200, 256000.

*** IMU_SET_PROTOCOL_MODE-----(0x12)**

Function: Used to set the communication baud rate; (default is 115200)

Note: It does not have the power-down save function. You must execute the save setting command to save.

The frame format is as follows:

| Frame sync byte | Frame start byte | ADDR | CMD | LEN (MSB) | LEN (LSB) | DATA | CRC (MSB) | CRC (LSB) | Frame end byte |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 5 | 1 | 1 | 1 |
| 0xFF | 0x02 | 0x00 | 0x12 | 0x00 | 0x05 | As follow | 0xXX | 0xXX | 0x03 |

Of which: DATA part

| DATA | |
|---|---|
| Reserved.Leave to 0(bytes) | Baudrate ( uint32) |
| 1 | 4 |
| 0 | 115200 |

Note: Baudrate (uint32) is little-endian mode, with the low byte first and the high byte last.

Reply :

IMU_ACK

| Frame sync | Frame start | ADDR | CMD | LEN (MSB) | LEN (LSB) | DATA | CRC (MSB) | CRC (LSB) | Frame end |
|---|---|---|---|---|---|---|---|---|---|

| byte | byte | | | | | | | | byte |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0xFF | 0x02 | 0x00 | 0x01 | 0x00 | 0x01 | ERRO R_CO DE | 0xXX | 0xXX | 0x03 |

**4.8 Get Baud Rate**

*** IMU_GET_PROTOCOL_MODE------(0x13)**

Function: Used to get the communication baud rate;

The frame format is as follows:

| Frame sync byte | Frame start byte | ADDR | CMD | LEN (MSB) | LEN (LSB) | DATA | CRC (MSB) | CRC (LSB) | Frame end byte |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| 0xFF | 0x02 | 0x00 | 0x13 | 0x00 | 0x00 | NULL | 0xXX | 0xXX | 0x03 |

Reply：

***IMU_RET_PROTOCOL_MODE------(0x14)**

Function: Return the current baud rate;

The frame format is as follows:

| Frame sync byte | Frame start byte | ADDR | CMD | LEN (MSB) | LEN (LSB) | DATA | CRC (MSB) | CRC (LSB) | Frame end byte |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 4 | 1 | 1 | 1 |
| 0xFF | 0x02 | 0x00 | 0x14 | 0x00 | 0x04 | DATA | 0xXX | 0xXX | 0x03 |

Of which: DATA part

| DATA |
|---|
| baudrate |
| 4 |

**4.9 Setting the sensor address**

The sensor address is the target address during communication. Only when the address field matches the address of the sensor can the communication succeed.

*** IMU_SET_ADDR-----(0x0F)**

Function: used to set the address of the sensor; (default is 0x00)

Note: It does't have the power-down save function. You must execute the save setting command to save

The frame format is as follows:

| Frame sync byte | Frame start byte | ADDR | CMD | LEN (MSB) | LEN (LSB) | DATA | CRC (MSB) | CRC (LSB) | Frame end byte |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0xFF | 0x02 | 0x00 | 0x0F | 0x00 | 0x01 | addres s | 0xXX | 0xXX | 0x03 |

Note: ADDR is the current sensor address to be changed, and DATA is the new sensor address to be set.

Reply：

IMU_ACK

| Frame sync byte | Frame start byte | ADDR | CMD | LEN (MSB) | LEN (LSB) | DATA | CRC (MSB) | CRC (LSB) | Frame end byte |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0xFF | 0x02 | 0x00 | 0x01 | 0x00 | 0x01 | ERRO R_CO DE | 0xXX | 0xXX | 0x03 |

Note: The reply ADDR is the previous sensor address.

**4.10 Obtaining the Sensor Address**

*** IMU_GET_ADDR------(0x0D)**

Function: used to obtain the sensor address;

The frame format is as follows:

| Frame sync byte | Frame start byte | ADDR | CMD | LEN (MSB) | LEN (LSB) | DATA | CRC (MSB) | CRC (LSB) | Frame end byte |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| 0xFF | 0x02 | 0x00 | 0x0D | 0x00 | 0x00 | NULL | 0xXX | 0xXX | 0x03 |

Reply:

***IMU_RET_ADDR------(0x0E)**

Function: return the sensor address;

The frame format is as follows:

| Frame sync byte | Frame start byte | ADDR | CMD | LEN (MSB) | LEN (LSB) | DATA | CRC (MSB) | CRC (LSB) | Frame end byte |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0xFF | 0x02 | 0x00 | 0x0E | 0x00 | 0x01 | ADDR | 0xXX | 0xXX | 0x03 |

Of which: DATA part

| DATA |
|---|
| ADDR |
| 1 |

### 4.11 Setting The Measurement Installation Method

The measuring direction of the sensor is divided into three types: horizontal, vertical upward, and lateral. So you can choose the appropriate installation method.

*** IMU_SET_FIXED_MODE-----(0x20)**

Function: used to set the installation method of the sensor; (default is 0x00, horizontal direction)

Note: It does't have the power-down save function. You must execute the save setting command to save

The frame format is as follows:

| Frame sync byte | Frame start byte | ADDR | CMD | LEN (MSB) | LEN (LSB) | DATA | CRC (MSB) | CRC (LSB) | Frame end byte |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0xFF | 0x02 | 0x00 | 0x20 | 0x00 | 0x01 | Fixed_ Mode | 0xXX | 0xXX | 0x03 |

Note: Fixed_Mode: 0x00: horizontal mode, 0x01: vertical upward, 0x02: lateral mode;

Reply:

IMU_ACK

| Frame sync byte | Frame start byte | ADDR | CMD | LEN (MSB) | LEN (LSB) | DATA | CRC (MSB) | CRC (LSB) | Frame end byte |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0xFF | 0x02 | 0x00 | 0x01 | 0x00 | 0x01 | ERRO R_CO DE | 0xXX | 0xXX | 0x03 |

### 4.12 Get Sensor Address

*** IMU_GET_FIXED_MODE------(0x21)**

Function: used to obtain the installation method of the sensor;

The frame format is as follows:

| Frame sync byte | Frame start byte | ADDR | CMD | LEN (MSB) | LEN (LSB) | DATA | CRC (MSB) | CRC (LSB) | Frame end byte |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| 0xFF | 0x02 | 0x00 | 0x21 | 0x00 | 0x00 | NULL | 0xXX | 0xXX | 0x03 |

Reply:

***IMU_RET_FIXED_MODE------(0x22)**

Function: return the sensor address;

The frame format is as follows:

| Frame sync | Frame start | ADDR | CMD | LEN (MSB) | LEN (LSB) | DATA | CRC (MSB) | CRC (LSB) | Frame end |
|---|---|---|---|---|---|---|---|---|---|

| byte | byte | | | | | | | | byte |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0xFF | 0x02 | 0x00 | 0x22 | 0x00 | 0x01 | Fixed_ Mode | 0xXX | 0xXX | 0x03 |

Of which: DATA part

| DATA |
|---|
| Fixed_Mode |
| 1 |

**4.13 Set Gravity Magnitude**
*** IMU_SET_GRAVITY_MAGNITUDE------(0xB1)**

Function：set current gravity magnitude level；

Note: It does not have the power-down save function. You must execute the save setting command to save

The frame format is as follows:

| Frame sync byte | Frame start byte | ADDR | CMD | LEN (MSB) | LEN (LSB) | DATA | CRC (MSB) | CRC (LSB) | Frame end byte |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 5 | 1 | 1 | 1 |
| 0xFF | 0x02 | 0x00 | 0xB1 | 0x00 | 0x05 | DATA | 0xXX | 0xXX | 0x03 |

Of which: DATA part

| DATA | |
|---|---|
| Reserved.Leave to 0(bytes) | magnitude ( real32) |
| 1 | 4 |

Magnitude: gravity magnitude level. Floating point type, little-endian mode.

Reply:

**IMU_ACK**

| Frame sync byte | Frame start byte | ADDR | CMD | LEN (MSB) | LEN (LSB) | DATA | CRC (MSB) | CRC (LSB) | Frame end byte |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0xFF | 0x02 | 0x00 | 0x01 | 0x00 | 0x01 | ERRO R_CO DE | 0xXX | 0xXX | 0x03 |

**4.14 Get Gravity Magnitude**
*** IMU_GET_GRAVITY_MAGNITUDE------(0xB2)**

Function：get current gravity magnitude level；

The frame format is as follows:

| Frame sync byte | Frame start byte | ADDR | CMD | LEN (MSB) | LEN (LSB) | DATA | CRC (MSB) | CRC (LSB) | Frame end byte |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| 0xFF | 0x02 | 0x00 | 0xB2 | 0x00 | 0x00 | NULL | 0xXX | 0xXX | 0x03 |

Reply:

*** IMU_RET_GRAVITY_MAGNITUDE------(0xB3)**

Function：return current gravity magnitude level；

The frame format is as follows:

| Frame sync byte | Frame start byte | ADDR | CMD | LEN (MSB) | LEN (LSB) | DATA | CRC (MSB) | CRC (LSB) | Frame end byte |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 4 | 1 | 1 | 1 |
| 0xFF | 0x02 | 0x00 | 0xB3 | 0x00 | 0x04 | DATA | 0xXX | 0xXX | 0x03 |

Of which: DATA part

| DATA |
|---|
| magnitude ( real32) |
| 4 |

**4.15 Set Save Command**

**The set parameters do not have a power-down saving function. After changing the setting parameters, you must execute the setting save command to save the updated parameters.**

*** IMU_SAVE_SETTINGS ------ (0x24)**

Function: It is used to save all the setting parameters to the EEPROM, and it has the function of power-down saving. No data fields. The frame format is as follows:

| Frame sync byte | Frame start byte | ADDR | CMD | LEN (MSB) | LEN (LSB) | DATA | CRC (MSB) | CRC (LSB) | Frame end byte |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| 0xFF | 0x02 | 0x00 | 0x24 | 0x00 | 0x00 | NULL | 0xXX | 0xXX | 0x03 |

Reply:

**IMU_ACK**

| Frame sync byte | Frame start byte | ADDR | CMD | LEN (MSB) | LEN (LSB) | DATA | CRC (MSB) | CRC (LSB) | Frame end byte |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0xFF | 0x02 | 0x00 | 0x01 | 0x00 | 0x01 | ERROR_CODE | 0xXX | 0xXX | 0x03 |

Note: EEROR CODE refers to 6;

**4.16 Default Data Query Command (When Output Mode Mode = 0, One Question And One Answer Mode)**

*** IMU_GET_DEFAULT_OUTPUT------(0x56)**

Function: Get the current default output data item.。

The frame format is as follows:

| Frame sync byte | Frame start byte | ADDR | CMD | LEN (MSB) | LEN (LSB) | DATA | CRC (MSB) | CRC (LSB) | Frame end byte |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| 0xFF | 0x02 | 0x00 | 0x56 | 0x00 | 0x00 | NULL | 0xXX | 0xXX | 0x03 |

Reply:

***IMU_RET_DEFAULT_OUTPUT------(0x57)**

Function: Returns the default data item output.

When outputmask = 0x00000007j for continuous automatic output, the Data part is as follows:

The frame format is as follows:

| Frame sync byte | Frame start byte | ADDR | CMD | LEN (MSB) | LEN (LSB) | DATA | CRC (MSB) | CRC (LSB) | Frame end byte |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 24 | 1 | 1 | 1 |
| 0xFF | 0x02 | 0x00 | 0x57 | 0x00 | 0x24 | See below | 0xXX | 0xXX | 0x03 |

Data part as follow ( total length: 0x0024) :

| Name | Roll | Pitch | Yaw | Gx | Gy | Gz | Ax | Ay | Az |
|---|---|---|---|---|---|---|---|---|---|
| Size (bytes) | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| | 12 | | | 12 | | | 12 | | |

**4.17 Get the specified data item output**

*** IMU_GET_SPECIFIC_OUTPUT------(0x58)**

Function: Get the specially designated MASK data items.

The frame format is as follows:

| Frame | Frame | ADDR | CMD | LEN | LEN | DATA | CRC | CRC | Frame |
|---|---|---|---|---|---|---|---|---|---|

| sync byte | start byte | | | (MSB) | (LSB) | | (MSB) | (LSB) | end byte |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 4 | 1 | 1 | 1 |
| 0xFF | 0x02 | 0x00 | 0x58 | 0x00 | 0x04 | DATA | 0xXX | 0xXX | 0x03 |

Of which: DATA part

| DATA |
|---|
| Output masks(uint32) |
| 4 |

The specified specified data items Output masks are any of the following specified data items: IMU_OUTPUT_EULER | IMU_OUTPUT_GYROSCOPES | IMU_OUTPUT_ACCELEROMETERS Any one or more combinations.

Reply:

* **IMU_RET_SPECIFIC_OUTPUT------(0x59)**

Function: Returns the output of the specified data item.

The frame format is as follows:

| Frame sync byte | Frame start byte | ADDR | CMD | LEN (MSB) | LEN (LSB) | DATA | CRC (MSB) | CRC (LSB) | Frame end byte |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 0x24 | 1 | 1 | 1 |
| 0xFF | 0x02 | 0x00 | 0x59 | 0x00 | 0x24 | DATA | 0xXX | 0xXX | 0x03 |

DATA part as follow（total length: 0x0024）:

| Name | Roll | Pitch | Yaw | Gx | Gy | Gz | Ax | Ay | Az |
|---|---|---|---|---|---|---|---|---|---|
| Size (bytes) | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| | 12 | | | 12 | | | 12 | | |

Output masks(uint32) = 0x00000007

IMU_OUTPUT_EULER|IMU_OUTPUT_GYROSCOPES|IMU_OUTPUT_ACCELEROMETERS ;

RION TECH

ADD：Block 1, COFCO(FUAN) Robotics Industrial Park , Da Yang Road 90, Fuyong Distict, Shenzhen City,China

TEL：(86) 755-29657137 (86) 755-29761269

FAX：(86) 755-29123494

E-mail：sales@rion-tech.net

WEB：www.rion-tech.net