# INDEX

# FSS-IMU614E-AG Product Sheet

## 1. Description

The IMU614E-AG is equipped with the data fusion algorithms of IMU and GNSS, enabling low-cost, high-precision, anti-magnetic interference and attitude measurement. It is specially applied in the single-antenna controllers in the field of agricultural autopilot.

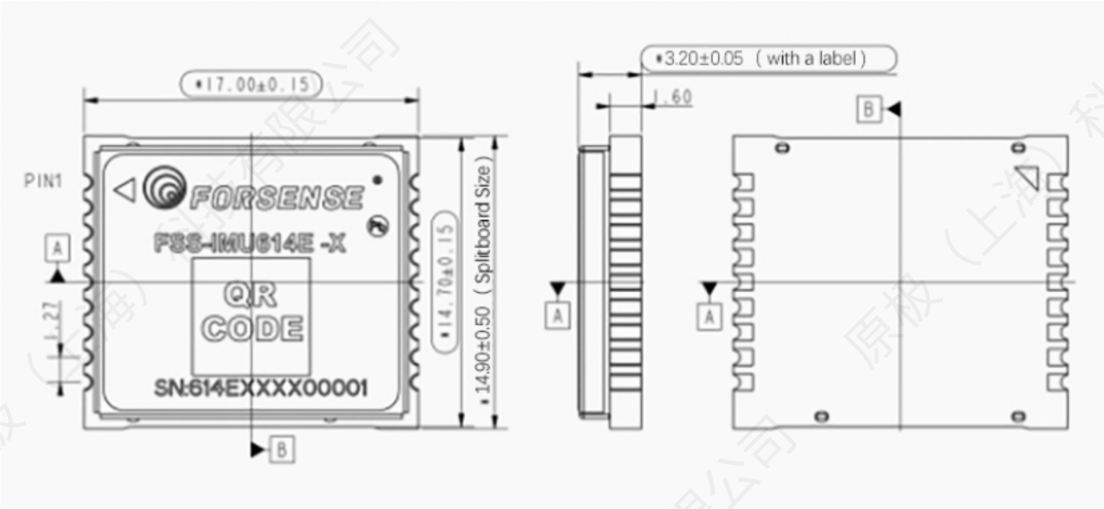
Table 1 Performance Indicators

| | | |
|---|---|---|
| Performance Indicators | Attitude Accuracy | Roll/Pitch: <0.2° rms<br>Heading:<0.3° rms (Vehicle speed > 1km/h, on-board Ackermann steering mechanism) |
| | Positional Accuracy | <2% @1σ<br>(On-board scenario, DR accuracy@(30 s), without wheel odometer) |
| | Update Rate | 100hz |
| | Gyroscope Measurement Range | ±500°/s |
| | Gyroscope Bias Instability | XY:4°/h Z:3°/h @25℃,ALLAN Variance, 1σ |
| | Accelerometer Measurement Range | ±6g |
| | Accelerometer Bias Instability | XY:20μg Z:40 μg @25℃,ALLAN Variance, 1σ |
| Operating Conditions | Voltage at the Common Collector | 3.3V |
| | Power Consumption | 0.085w |
| | Operating Temperature | -40~85℃ |
| | Storage Temperature | -40~85℃ |
| | Connection Protocol Support | COM:TTL<br>CAN (peripheral circuitry required) |

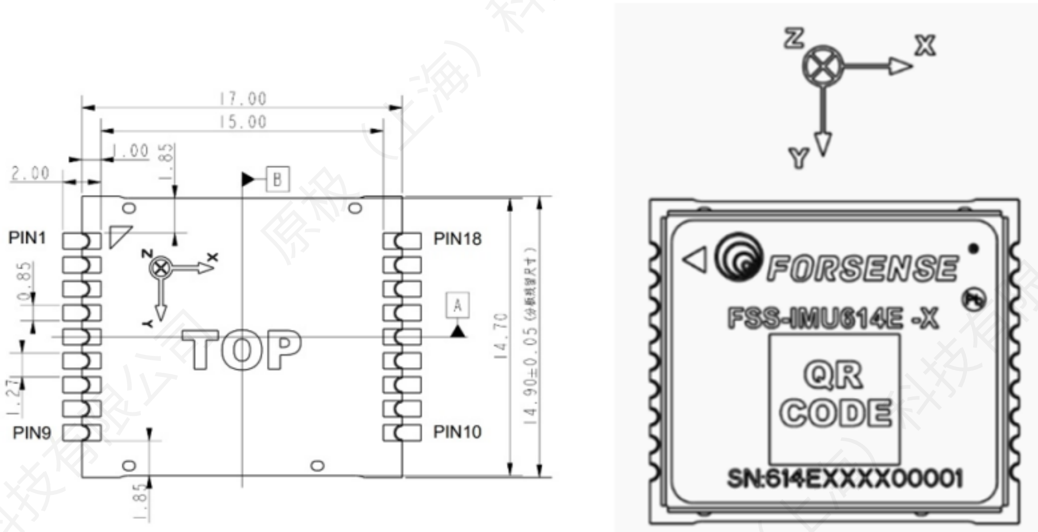
# 2. External Structure

X-axis should be installed toward the front of the vehicle and Y-axis facing right.

Fig. 1 Outline Structure and Dimensions (unit: mm)




External Structure




Recommended size for bonding pad

# 3. Electrical Properties

## 3.1 Absolute Maximum Ratings

Table 2 Absolute Maximum Ratings

| Parameter | Symbol | Minimum | Maximum | Unit |
|---|---|---|---|---|
| Voltage at the Common Collector | VCC | -0.3 | 4.0 | V |
| Ground | GND | - | - | - |
| Input Voltage | Vin | -0.3 | VCC+0.2 | V |
| Operating Temperature | Tot | -40 | 85 | ℃ |
| Storage Temperature | Tstg | -40 | 85 | ℃ |

## 3.2 Operating Conditions

Table 3 Operating Conditions

| Parameter | Symbol | Minimum | Typical | Maximum | Unit |
|---|---|---|---|---|---|
| Voltage at the Common Collector | VCC | 3.2 | 3.3 | 3.4 | V |
| VCC Maximum Ripple | Vrpp | | ±40 | | mV |
| Power Consumption | P | | 0.085 | | W |
| Operating Temperature | T | -40 | | 85 | ℃ |
| Storage Temperature | T | -40 | | 85 | ℃ |

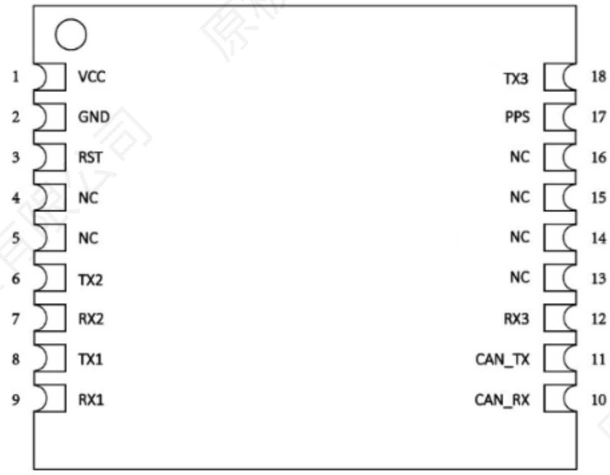
## 3.3 IO Threshold Characteristics

Table 4 IO Threshold Characteristics

| Parameter | Symbol | Minimum | Typical | Maximum | Unit |
|---|---|---|---|---|---|
| Input Pin Low Voltage | Vin_low | 0 | | VCC*0.2 | V |
| Input Pin High Voltage | Vin_high | VCC*0.7 | | VCC+0.2 | V |
| Output Pin Low Voltage | Vout_low | 0 | | 0.45 | V |
| Output Pin High Voltage | Vout_high | VCC-0.45 | | VCC | V |

# 4. Pin Definitions

Fig. 4 Pin Diagram




IMU614E- AG Pin Layout (Top View)

Table 5 Pin Definitions

| Pin No. | Pin Name | Description |
|---|---|---|
| 1 | VCC | Power Input: +3.3V, 40mA, with ripple not exceeding ±40mV |
| 2 | GND | Ground |
| 3 | NC | No connection |
| 4 | NC | No connection |
| 5 | NC | No connection |
| 6 | TX2 | Receive asynchronous data output |
| 7 | RX2 | Receive asynchronous data input |
| 8 | TX1 | Receive asynchronous data output<br>(Data communication interface (LVTTL)) |
| 9 | RX1 | Receive asynchronous data input<br>(Data communication interface (LVTTL)) |
| 10 | CAN_RX | CAN RX pin: reading data from bus and sending them to CAN controller |
| 11 | CAN_TX | CAN TX pin: reading data from CAN controller and sending them to bus driver |
| 12 | RX3 | Receive asynchronous data input |
| 13 | Reserved | Reserved |
| 14 | Reserved | Reserved |
| 15 | Reserved | Reserved |
| 16 | Reserved | Reserved |
| 17 | PPS | Signal to trigger external synchronized sampling;<br>(access to the pulse per second pin of RTK) |
| 18 | TX3 | Receive asynchronous data output |

For more information about IMU hardware design, please refer to the document. FSS-IMU614E-XX Hardware Design Manual

# 5. Recommended Welding Furnace Temperature Profile

Fig. 3 Welding Furnace Temperature Profile

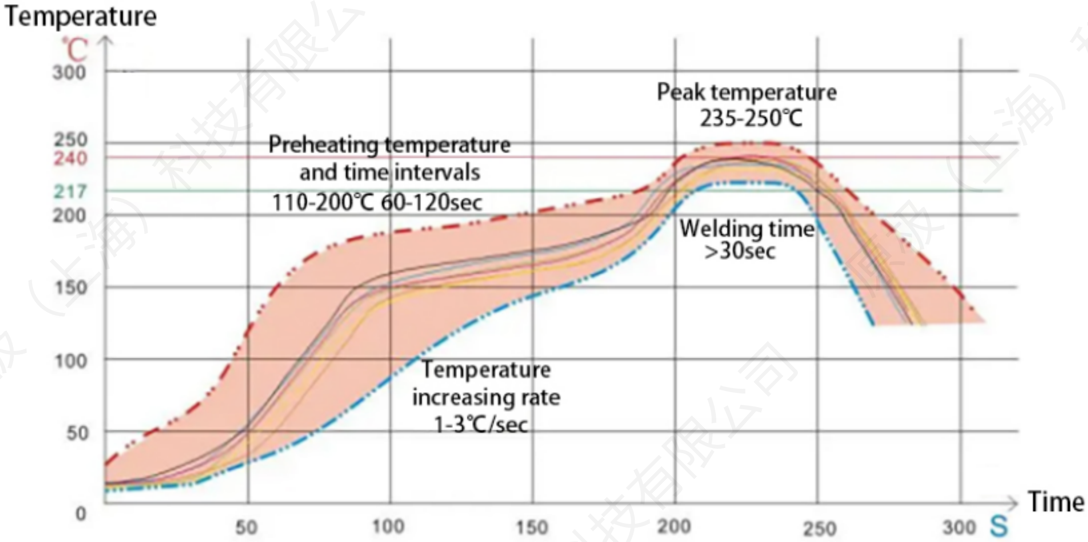



Table 6 Temperature Setting Modes

| Parameter | Minimum | Maximum | Unit |
|---|---|---|---|
| Maximum ramp-up rate (target = 0.8) (calculated every 60 seconds) | 1 | 3 | °/s |
| Maximum ramp-down rate (calculated every 60 seconds) | -3 | -1 | °/s |
| Preheating temperature and time intervals | 60 | 120 | s |
| Reflow time (period over 217°C) | 40 | 70 | s |
| Maximum temperature | 235 | 250 | °C |
| Maximum number of reflow | | 1 | Times |

For more information about SMT modules, please refer to the document .Forsense - LCC Module SMT Application Guide

**Notes:**

1.For modules, it is recommended to use reflow welding equipment with eight or higher temperature zones;

2.The module is a high-precision sensor sensitive to any deformation;

- If the thickness of the PCB board is less than 1.0 mm, it is recommended to use reflow fixtures to prevent the board from getting deformed under high temperature, thereby ensuring the coplanarity of pins.

- We recommend customers to use high TG value boards as PCB main boards to avoid deformation during high temperature reflow, thus reducing the possibility of warping, extrusion, empty soldering, and solder bridging.

3.Due to the sensitive devices inside the module, the maximum temperature of the reflow soldering machine must not exceed 260°C (referring to the the top surface temperature of the package);

4. It is recommended to use lead-free solder paste, including the recommended product Alpha OM-338 SAC305 Sn96.5Ag3.0Cu0.5;

5. Given sensitive devices in the module, the second reflow should be avoided to ensure proper performance of the module;

6. Cooling;

- Controlled cooling ramp rate can help reduce negative soldering effects (e.g. more brittle solder joints) and mechanical stresses within the product. Controlled cooling contributes to bright soldered surfaces with fine crystalline particles and low contact angles, avoiding the warping of the shielding cap caused by rapid cooling changes.

7. Exterior inspection:

- After the module is soldered, the X-ray and optical magnifying glass are used to test the welding quality. For details, please refer to the IPC-A-610F standards.

8. **Using electric soldering iron requires the temperature to be controlled at 260 °C to 290 °C, the single welding time shall not exceed 3s, and the anti-static treatment shall be done.**

# 6. ESD Protection


Static electricity may cause intermittent or permanent circuit damage, which is very harmful to electronic products. Most of them are identified as ESD damage.

Therefore, the electrostatic protection of modules is particularly important. The production and transportation process needs to be strictly subject to the following conditions:

- It is prohibited to touch the module, especially the pin position, with bare hands;
- SMT patch machines, workstations, soldering irons and other equipment need to be grounded;
- Operators wear human anti-static bracelets with grounding wires (cordless static bracelets are not allowed and anti-static gloves are recommended);
- Packaging and PCBs must be made of qualified anti-static materials.

# 7. Examples of Use

## 7.1 Device Installation

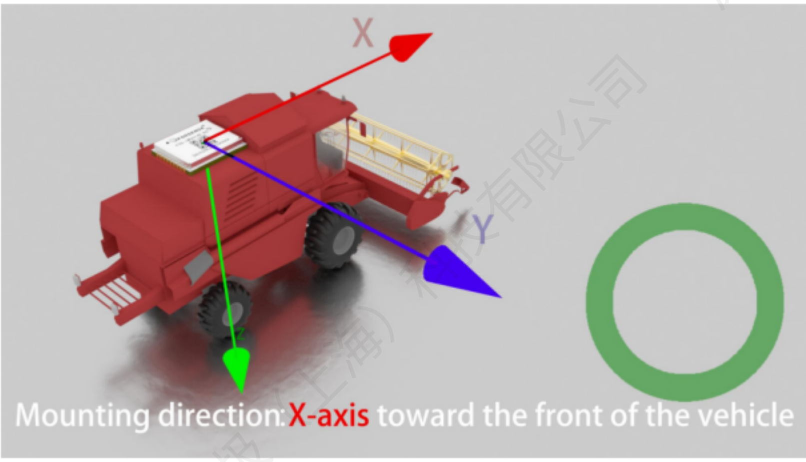
1. The module should be firmly fixed on a rigid plane rather than in a position of high vibration;

2. The module should be installed toward the front of the vehicle as shown in the figure below;

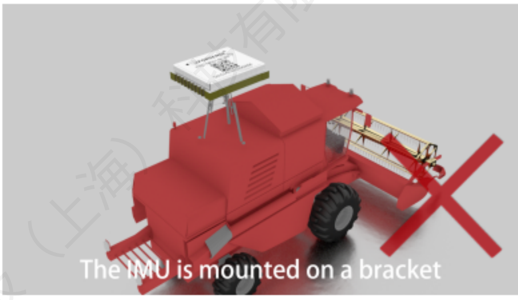
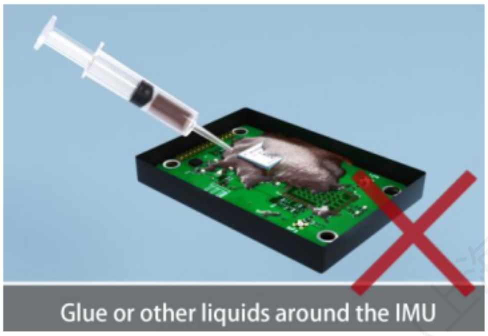
The correct installation method is as follows

X-axis toward the front of the vehicle

Fig. 14 Correct Installation Diagram



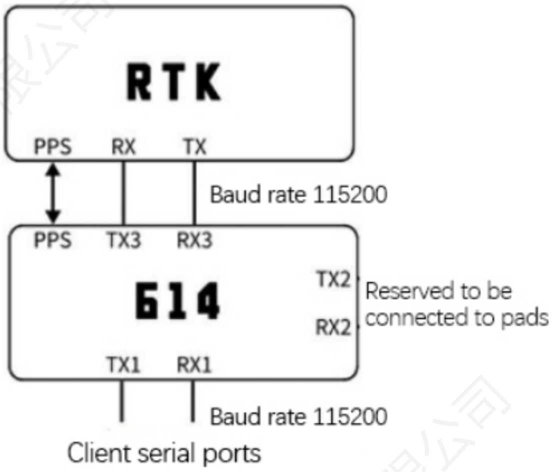

The following mounting methods are incorrect:


| | |
|---|---|
|  The IMU is mounted outside the vehicle |  The IMU is fixed by hand |
|  The IMU is mounted on a bracket  Glue or other liquids around the IMU |  X<br>Y<br>X-axis is not installed toward the front of the vehicle |

## 7.2 Connecting RTK Board

### 7.2.1 Recommended Connection Methods of RTK Board and AG Module

Fig. 4 Connection Diagram




### 7.2.2 RTK Configuration Requirements

Two protocols are currently supported for configuration, NovAtel and National Marine Electronics Association (NMEA)

NovAtel:

Command: AT+GNSS_CARD=UNICORE/r/n

Requirement on Message Input:

BESTPOSB 10hz

PSRVELB 10hz

GPGGA =10hz

Disable the INS-related function of RTK board and close other statements Baud rate 115200

Pulse per second (PPS): once per second, rising edge triggered, pulse width 5ms, and aligned to UTC time. High level must not be higher than 5v.

NMEA:

Command: AT+GNSS_CARD=OEM/r/n

Requirement on Message Input:

GPGGA =10hz

GPRMC =10hz

Disable the INS-related function of RTK board and close other statements Baud rate 115200

Pulse per second (PPS): once per second, rising edge triggered, pulse width 5ms, and aligned to UTC time. High level must not be higher than 5v.

### 7.2.3 RTK Differential Data Import

Users have to log in CORS account of Qianxun SI or Sixents Technology through the user's host, then the RTK board will output the GPGGA message to the differential server, which returns the differential data and forwards it to the RTK board, and then the RTK enters into the fixed solution state.

## 7.3 Parameters Setting

### 7.3.1 Configuration of the Primary Antenna Lever Arm

For example, configure the lever arm vector as X=0.5m, Y=-0.6m, Z=-1.0m.

Command: AT+CLUB_VECTOR=0.5,-0.6,-1.0\r\n

Response: GPS_POS_X=0.5,GPS_POS_Y=-0.6,GPS_POS_Z=-1.0/r/n

Note: The lever arm vector is the 3D vector (X,Y,Z) of the RTK main antenna phase center relative to the IMU phase center. (unit: m) In the Front-Right-Down coordinate system,

if the RTK main antenna is in front of the IMU, the value is positive, otherwise is negative;

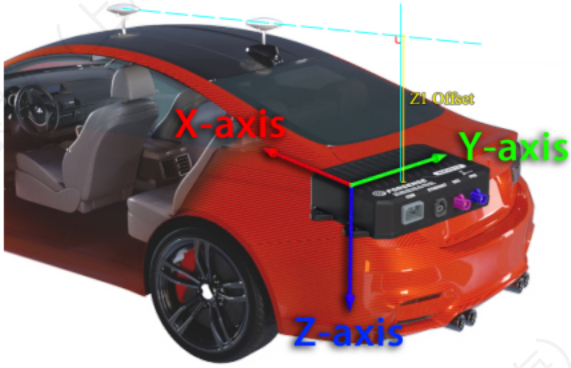
if the RTK main antenna is on the right side of the IMU, the value is positive , otherwise is negative;

if the RTK main antenna is above the IMU, the value is **negative**, otherwise is positive.

The coordinate system diagram is shown in the following figure: (the sticker needs to face upwards, and the IMU needs to be reconfigured in terms of mounting orientation if it is not installed in accordance with the following)

Fig. 5 Coordinate System Diagram


Fig. 6 Antenna Lever Arm Diagram


### 7.3.2 Configuration of the Rear Axle Center Lever Arm

For example, configure the lever arm vector as X=0.5m, Y=-0.6m, Z=1.0m.

Command: AT+PROJ_VECTOR=0.5,-0.6,1.0\r\n

Response: GPS_POS_X=0.5,GPS_POS_Y=-0.6,GPS_POS_Z=1.0/r/n

Note: The lever arm vector is the 3D vector (X,Y,Z) of the RTK main antenna phase center relative to the IMU phase center. (unit: m) In the Front-Right-Down coordinate system,

if the rear axle center is in front of the IMU, the value is positive, otherwise is negative;

if the rear axle center is on the right side of the IMU, the value is positive, otherwise is negative;

If the rear axle center is below the IMU, the value is **positive**, otherwise is negative. (Usually it places below the equipment)

The coordinate system diagram is shown in the following figure: (the sticker needs to face upwards, and the IMU needs to be reconfigured in terms of mounting orientation by referring to the section 8.2.2 if it is not installed in accordance with the following)

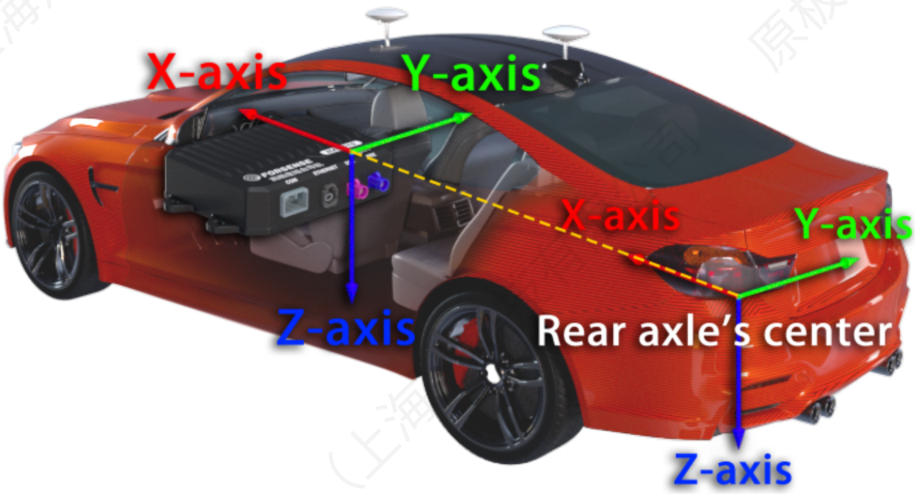




Fig. 7 Coordinate System Diagram

### 7.3.3 Configure the Vehicle Wheelbase

If the vehicle wheelbase is configured to be 2m, the configuration command is:
AT+WHEEL_BASE=2\r\n
Note: The command needs to be saved after configuration.

### 7.3.4 Configure RTK Dual Antenna Mounting Angle

If the RTK dual antenna mounting angle is configured to be 0 °, the configuration command is:
Command: AT+RTK_ANGLE=0\r\n
Response: ANGLE=0\r\n
Explanation: Mounting angle is the angle between the ray of the primary antenna pointing to the secondary antenna and the direction of the front end of the vehicle. Clockwise motion is positive, while counterclockwise motion is considered negative. The angle input range is -180° to 180°.
Note: After saving the configuration commands, power off and restart are necessary steps. The distance between the two antennas should be more than 50cm.

### 7.3.5 Open the Dual Antenna Fusion

Note: The default status of dual antenna fusion stays closed and will only be used for for quick initialization when stationary, if needed to use full dual-antenna heading, please press the following command to turn on the dual-antenna fusion: 1 is to turn on the dual-antenna fusion, and the other parameters are to turn off the dual-antenna fusion, please noted that synchronously calibrate the dual-antenna mounting error is needed, or there will be a fixed error with the car body heading.

Command: AT+RTK_ANGLE=1\r\n

Response: OK

Note: After saving the configuration commands, power off and restart are necessary steps. The distance between the two antennas should be more than 50cm.

### 7.3.6 Calibrate Dual Antenna

The dual antenna calibration process is as follows:

1. Connect serial ports and enter AT+SETNO\r\n to stop data output;

2. Send the command AT+RTK_BIAS_EST=1\r\n to start the calibration process;

3.Vehicles travel straight at the speed of no less than 3km/h, and the calibration is successfully done when serial ports output RTK_BIASE_CORRECT_DONE;

4.Send AT+CONFIG\r\n command to see data output, RTK_BIAS_FLAG_AND_VALUE=99,XX (calibration angle), and check whether it is reasonable. The installation angle is normally less than 1°;

5. Enter AT+SAVE\r\n to save the result and re-power on.

## 7.4 Save Parameters

After all the configuration commands are finished, send the command “AT+SAVE\r\n” to save the parameters.

# 8. Output Protocols

## 8.1 Binary Protocols - AG Data Stream

Notes:

- CRC checksum starts from the frame header covering all bytes of the frame except the CRC checksum bit itself, seeing the appendix for calculation methods and routines.
- The frame length is the total number of data bytes excluding frame header, frame ID, frame length and parity bits;
- In little-endian mode, and the low byte is sent first.

Table 7 Binary Protocols - AG Data Stream

| Parameter | Type | Relative position |
|---|---|---|
| Frame header 1: 0xAA | Uint8 | 0 |
| Frame header 2: 0x55 | Uint8 | 1 |
| Frame ID: 0x0156 | Uint16 | 2 |
| Frame length: 0x0032 | Uint16 | 4 |
| Seconds in GPS week (ms) | Uint32 | 6 |
| Roll angle (°) | Float | 10 |
| Pitch angle (°) | Float | 14 |
| Yaw angle (°) | Float | 18 |
| Reserved | Float | 22 |
| 3-axis angular velocity (deg/s, front-right-down) | Float*3 | 26 |
| 3-axis acceleration (g, front-right-down) | Float*3 | 38 |
| RTK positioning status | Uint8 | 50 |
| Attitude valid bits<br>Bit0: 1 means the attitude is valid, 0 is invalid<br>Bit1: 1 means front wheel angle is valid, 0 is invalid | Uint8 | 51 |
| Status bits:<br>Bit0: 1 means the RTK data is valid, 0 is invalid;<br>Bit1: 1 means PPS signal is valid, 0 is invalid;<br>Bit2: 1means the integrated navigation has been initialized, 0 means otherwise.<br>(Single antenna initialization conditions: RTK positioning status 4 + speed to 0.5 m/s or more + with acceleration and deceleration of straight line driving. Dual antenna initialization conditions: RTK positioning state 4 + RTK orientation state 50)<br>Bit3: 1 indicates that the front wheel angle is valid, 0 is invalid<br>Bit4: 1 indicates that the integrated navigation has converged, 0 is not converged<br>Bit5: 1 indicates that the front wheel gyro data is valid, 0 is invalid (no external gyro can be ignored)<br>Bit6: 1 indicates that the steering wheel motor data is valid, 0 is invalid | Uint32 | 52 |

| | | |
|---|---|---|
| Bit7 and Bit8:<br>01 means that the vehicle moves forward (Bit7=1、Bit8=0)<br>10 means that the vehicle moves backward (Bit7=0、Bit8=1)<br>00 means invalid (Bit7=0、Bit8=0) | | |
| CRC verification | Uint32 | 56 |

## 8.2 Binary Protocols - INS Data Stream

Notes:

- CRC checksum starts from the frame header covering all bytes of the frame except the CRC checksum bit itself, seeing the appendix for calculation methods and routines.
- The frame length is the total number of data bytes excluding frame header, frame ID, frame length and parity bits;
- In little-endian mode, and the low byte is sent first.

Configuration command is:

Command: AT+SETNAV\r\n

Response: OK\r\n

If there's no data output, the configuration command is:

Command: AT+SETNO\r\n

Response: OK\r\n

Table 8 Binary Protocols - INS data stream

| Parameter | Type | Relative position |
|---|---|---|
| Frame header 1: 0xAA | Uint8 | 0 |
| Frame header 2: 0x55 | Uint8 | 1 |
| Frame ID: 0x0166 | Uint16 | 2 |
| Frame length: 0x005E | Uint16 | 4 |
| Seconds in GPS week (ms) | Uint32 | 6 |
| GPS week number count | Uint16 | 10 |
| Latitude (deg x 10,000,000) | Int32 | 12 |
| Longitude (deg x 10,000,000) | Int32 | 16 |
| Height (mm) | Int32 | 20 |
| Northward velocity (m/s) | Float | 24 |
| Eastward velocity (m/s) | Float | 28 |
| Ground velocity (m/s) | Float | 32 |
| Roll angle (°) | Float | 36 |
| Pitch angle (°) | Float | 40 |
| Yaw angle (°) | Float | 44 |
| Single antenna: AHRS heading (°) (cannot be | Float | 48 |

| | | |
|---|---|---|
| used as a reference)<br>Dual antenna: RTK dual-antenna heading (°)<br>Front wheel gyro G200: Z-axis angular velocity (°/s) | | |
| Reserved (displayed as the front wheel angle when the model is connected to G200) | Float | 52 |
| Accelerometer X-axis (g) | Float | 56 |
| Accelerometer Y-axis (g) | Float | 60 |
| Accelerometer Z-axis (g) | Float | 64 |
| Gyroscope X-axis (deg/s) | Float | 68 |
| Gyroscope Y-axis (deg/s) | Float | 72 |
| Gyroscope Z-axis (deg/s) | Float | 76 |
| IMU temperature (°C) | Float | 80 |
| RTK positioning status (same as that in GGA)0:No positioning 1:Single point positioning 2:Pseudorange differential positioning4:Fixed solution 5:Floating point solution | Uint8 | 84 |
| Number of satellites | Uint8 | 85 |
| Differential delay | Uint8 | 86 |
| Dual Antenna Orientation Status, 50 means oriented, otherwise means not oriented | Uint8 | 87 |
| Position accuracy factor (cm) Effective INS after the initialization | Uint16 | 88 |
| Status bits<br>Bit0: 1 means the RTK data is valid, 0 means invalid; Bit1: 1 means PPS signal is valid, 0 means invalid;<br>Bit2: 1means the integrated navigation has been initialized, 0 means otherwise. (Single antenna initialization conditions: RTK positioning status 4 + speed to 0.5m / S or more + with acceleration and deceleration of straight line driving.<br>Dual antenna initialization conditions: RTK positioning state 4 + RTK orientation state 50)<br>Bit3: 1 indicates that the front wheel angle is valid, 0 is invalid<br>Bit4: 1 indicates that the integrated navigation has converged, 0 is not converged<br>Bit5: 1 indicates that the front wheel gyro data is valid, 0 is invalid (no external gyro can be ignored)<br>Bit6: 1 indicates that the steering wheel motor data is valid, 0 is invalid<br>Bit7 and Bit8:<br>01 means that the vehicle moves forward (Bit7=1、Bit8=0)<br>10 means that the vehicle moves backward | Uint16 | 90 |

| | | |
|---|---|---|
| (Bit7=0、Bit8=1)<br>00 means invalid (Bit7=0、Bit8=0) | | |
| Reserved 1 | Uint32 | 92 |
| Reserved 2 | Uint32 | 96 |
| CRC verification | Uint32 | 100 |

## 8.3 CAN Communication Protocols

Table 9 CAN Extended Frame Format 0x19FF CC9A (unit: deg*100, int16)

| Extended Frame ID | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| 0x19FF CC9A | Roll angle | | Pitch angle | | Yaw angle | | Track angle | |

Table 10 CAN Extended Frame Format 0x19FF CD9A (unit: deg/s*52.0127, int16)

Note: Reserved, and if connected to G200, output the front wheel angle.

| Extended Frame ID | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| 0x19FF CD9A | gyro_x | | gyro_y | | gyro_z | | Reserved | |

Table 11 CAN Extended Frame Format 0x19FF CE9A (unit: g*3276.8, int16)

Status bits:

Bit0: RTK board data valid flag bit. 1 means valid and 0 means invalid

Bit1: PPS valid flag bit. 1 means valid and 0 means invalid.

Bit2: Front wheel angle valid flag bit. 1 means valid and 0 means invalid

Bit3: Front wheel gyro valid flag bit. 1 means valid and 0 means invalid.

Bit4: RTK positioning fixed solution flag bit. 1 means fixed solution and 0 means unfixed solution.

Bit5: RTK directional fixed solution flag bit. 1 means fixed solution and 0 means unfixed solution.

Bit6: Heading initialization flag bit. 1 means initialized and 0 means not initialized (only for the gyro of single antenna vehicle)

| Extended Frame ID | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| 0x19FF CE9A | accel_x | | accel_y | | accel_z | | Status bits | |

Table 12 CAN Extended Frame Format 0x19FF CE9B

Note: Speed direction (1: forward, -1: backward)

| Extended Frame ID | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| 0X19FF CE9B | Seconds in GPS week | | | | Estimated antenna mounting deviation angle | | Speed directio n | Reser ved |

Table 13 CAN Extended Frame Format 0X19CCFF9A (Note: The unit is °, and 1e7 and Int32 needed to be excluded)

The equipment has adopted the WGS84 Coordinated Syetem, with the default output of latitude and longitude is the phase center position of the primary antenna, and the output is the position of the projection point if the projection point is configured;

Latitudes: Taking equator as 0°, when latitude is greater than 0° , it means northern hemisphere, otherwise is southern hemisphere;

Longitudes: Taking the prime meridian as 0°, Longitude greater than 0° is the Eastern Hemisphere, while the opposite is the Western Hemisphere.

| Extended Frame ID | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| 0X19CCFF9 A | Longitudes | | | | Latitudes | | | |

Table 14: CAN Extended Frame Format 0X19CCFF9B ( Ellipsoid height, 1e7 and Int32 excluded)

| Extended Frame ID | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| 19CCFF9B | Elevation | | | | Seconds in GPS week | | | |

# 9. Input Protocols

## 9.1 Binary Configuration Protocols

Notes:

- CRC checksum starts from the frame header covering all bytes of the frame except the CRC checksum bit itself, seeing the appendix for calculation methods and routines;
- The frame length is the total number of data bytes excluding frame header, frame ID, frame length and parity bits;
- In little-endian mode, and the low byte is sent first.
- This frame needs to be sent continuously and is not saved in case of power failure.

Table 13 Binary Configuration Protocols

| Parameter | Type | Relative position |
|---|---|---|
| Frame header 1: 0x55 | Uint8 | 0 |
| Frame header 2: 0xAA | Uint8 | 1 |
| Frame ID: 0x0101 | Uint16 | 2 |
| Frame length: 0x0018 | Uint16 | 4 |
| (Reserved) | Uint8 | 6 |
| (Reserved) | Uint16 | 7 |
| (Reserved) | Uint8 | 9 |
| Antenna lever arm X (cm) | Int16 | 10 |
| Antenna lever arm Y (cm) | Int16 | 12 |
| Antenna lever arm Z (cm) | Int16 | 14 |
| (Reserved) | Int16 | 16 |
| (Reserved) | Int16 | 18 |
| (Reserved) | Int16 | 20 |
| (Reserved) | Int32 | 22 |
| (Reserved) | Uint16 | 26 |
| (Reserved) | Uint8 | 28 |
| Output Protocols<br>1: output the frames described in section 6.1<br>2: output reserved protocols (normally not used) | Uint8 | 29 |
| CRC check bit | Uint32 | 30 |

## 9.2 String Configuration Protocols

### 9.2.1 Configure the primary antenna arm

For example, configure the lever arm vector as X=0.5m, Y=-0.6m, Z=-1.0m.

Command: AT+CLUB_VECTOR=0.5,-0.6,-1.0\r\n

Response: GPS_POS_X=0.5,GPS_POS_Y=-0.6,GPS_POS_Z=-1.0/r/n

Note: The lever arm vector is the 3D vector (X,Y,Z) of the RTK main antenna phase center relative to the IMU phase center. (unit: m) In the Front-Right-Down coordinate system,

the value is positive if the RTK main antenna is in front of the IMU, otherwise is negative;

the value is positive if the RTK main antenna is on the right side of the IMU, otherwise is negative;

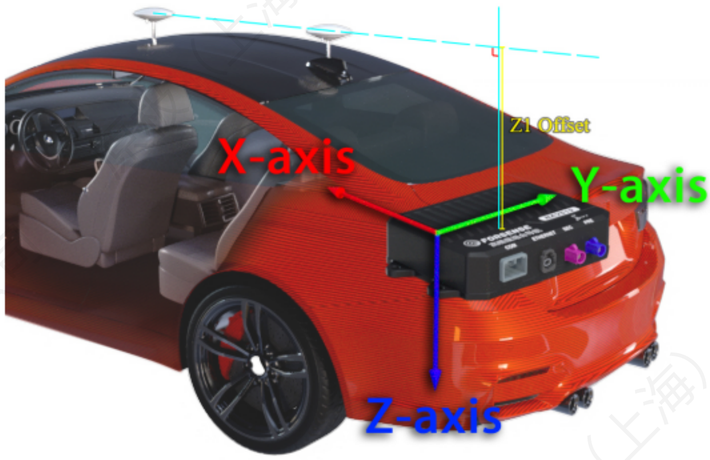
the value is **negative** if the RTK main antenna is above the IMU, otherwise is positive.

The coordinate system diagram is shown in the following figure: (the sticker needs to face upwards, and the IMU needs to be reconfigured in terms of mounting orientation by referring to the section 8.2.2 if it is not installed in accordance with the following)

Fig. 5 Coordinate System Diagram


Fig. 6 Antenna Lever Arm Diagram


SLAVE
MASTER


X-axis
Y-axis
Z-axis
Z1 Offset

## 9.2.2 Configure the Rear Axle Center Arm

For example, configure the lever arm vector as X=0.5m, Y=-0.6m, Z=1.0m.

Command: AT+PROJ_VECTOR=0.5,-0.6,1.0\r\n

Response: GPS_POS_X=0.5,GPS_POS_Y=-0.6,GPS_POS_Z=1.0/r/n

Note: The lever arm vector is the 3D vector (X,Y,Z) of the RTK main antenna phase center relative to the IMU phase center. (unit: m) In the Front-Right-Down coordinate system,

the value is positive if the RTK main antenna is in front of the IMU, otherwise is negative;

the value is positive if the RTK main antenna is on the right side of the IMU, otherwise is negative;

If the rear axle center is below the IMU, the value is **positive**, otherwise is negative. (Usually it places below the equipment)

The coordinate system diagram is shown in the following figure: (the sticker needs to face upwards, and the IMU needs to be reconfigured in terms of mounting orientation by referring to the section 8.2.2 if it is not installed in accordance with the following)

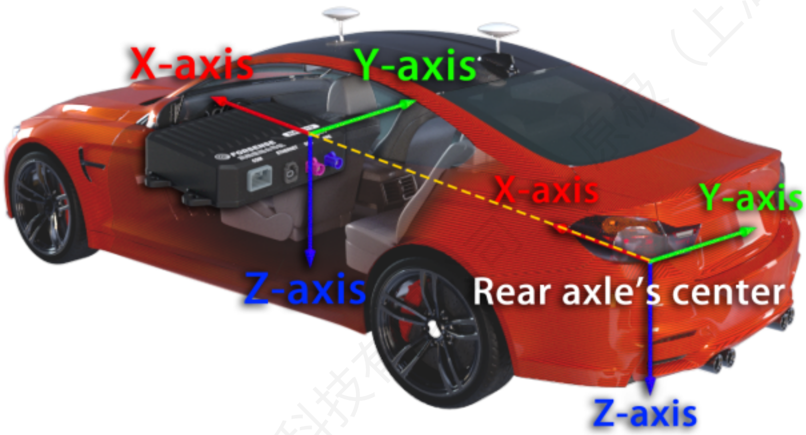



Fig. 7 Coordinate System Diagram

## 9.2.3 Configure the Vehicle Wheelbase

If the vehicle wheelbase is configured to be 2m, the configuration command is:

AT+WHEEL_BASE=2\r\n

Note: The command needs to be saved after configuration.

## 9.2.4 Configure RTK Dual Antenna Mounting Angle

If you configure the RTK dual antenna mounting angle to be 0 °, the configuration command is:

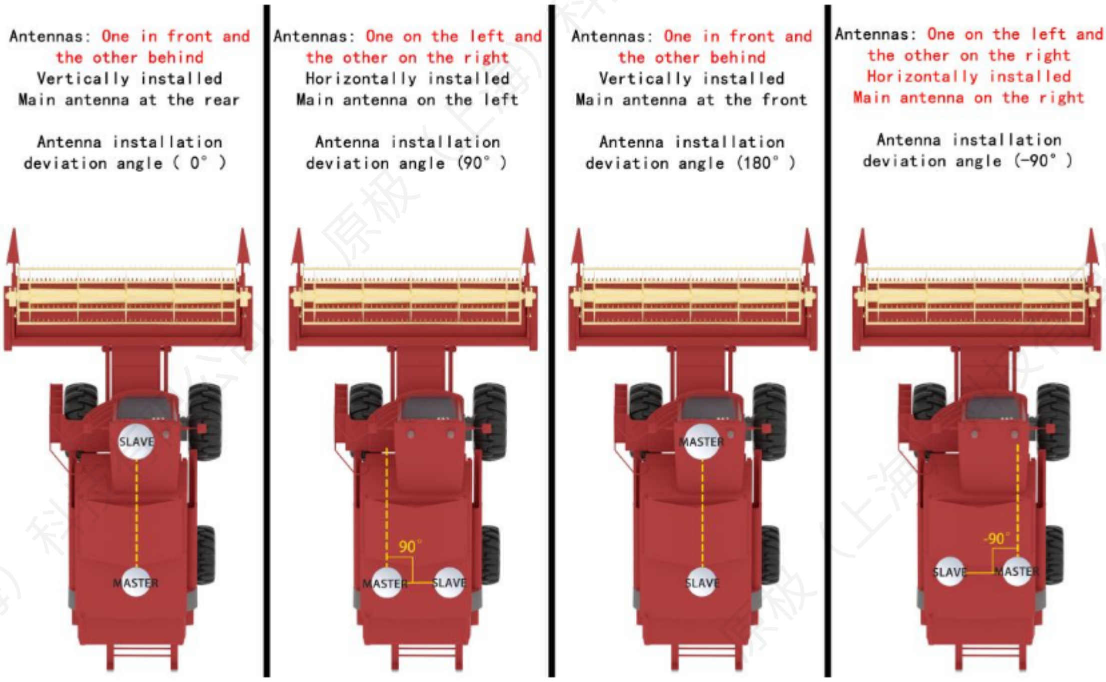
Command: AT+RTK_ANGLE=0\r\n

Response: ANGLE=0\r\n

Explanation: The mounting angle is between the main antenna's ray pointing to the secondary antenna and the direction of the front of the vehicle. Clockwise motion is positive, while counterclockwise motion is considered negative. The angle input range is -180° to 180°.

Note: After saving the configuration commands, power off and restart are necessary steps. The distance between the two antennas should be more than 50cm.

Fig. 7 RTK Dual Antenna Installation Angle Diagram



### 9.2.5 Open the Dual Antenna Fusion

Note: The default status of dual antenna fusion stays closed and will only be used for for quick initialization when stationary, if needed to use full dual-antenna heading, please press the following command to turn on the dual-antenna fusion: 1 is to turn on the dual-antenna fusion, and the other parameters are to turn off the dual-antenna fusion, please noted that synchronously calibrate the dual-antenna mounting error is needed, or there will be a fixed error with the car body heading.

Command: AT+RTK_ANGLE=1\r\n

Response: OK

Note: After saving the configuration commands, power off and restart are necessary steps. The distance between the two antennas should be more than 50cm.

### 9.2.6 Calibrate Dual Antenna

The dual antenna calibration process is as follows:

1. Connect serial ports and enter AT+SETNO\r\n to stop data output;
2. Send the command AT+RTK_BIAS_EST=1\r\n to start the calibration process;
3. Vehicles travel straight at the speed of no less than 3km/h, and the calibration is successfully done when serial ports output RTK_BIASE_CORRECT_DONE;
4. Send AT+CONFIG\r\n command to see data output stop, RTK_BIAS_FLAG_AND_VALUE=99,XX (calibration angle), and check whether it is reasonable. The installation angle is normally less than 1°;
5. Enter AT+SAVE\r\n to save the result and re-power on.

### 9.2.7 Configure the INS to Output Projection Points of Position and Velocity

If the INS is configured to output projection points, the configuration command is:

Command: AT+PROJ_VECTOR=1.0,2.0,3.0\r\n

Response: PROJ_VECTOR_X=1.0, PROJ_VECTOR_Y=2.0, PROJ_VECTOR_Z=3.0/r/n

Explanation: The default output of the integrated navigation is the antenna phase center coordinates, if other position coordinates is needed, configure the IMU phase center relative to the projected point position of the rod arm vector. The configuration method is the same as 8.1.1 Rod Arm Configuration.

### 9.2.8 Configure the Binary Output - AG Data Stream

Command: AT+SETAG\r\n

Response: OK\r\n

If the data output stops

Command: AT+SETNO\r\n

Response: OK\r\n

### 9.2.9 Binary Protocols - INS Data Stream

Command: AT+SETNAV\r\n
Response: OK\r\n
If the data output stops
Command: AT+SETNO\r\n
Response: OK\r\n

### 9.2.10 Configure the Output of Data Stream in NMEA Format

If the NEMA statement is configured for output, the 7.2 INS data stream will not be output. If you want to switch to 7.2 INS data stream output, you need to stop the current data stream output by pressing the 7.2.2 command first.
The configuration commands are as follows
GPGGA
Example: Output GPGGA at 5Hz Statement: AT+GPGGA=5\r\n
Response: OK\r\n
GPRMC
Example: Output GPRMC at 1Hz Statement: AT+GPRMC=1\r\n
Response: OK\r\n
GPHDT (Heading information)
Example: Output GPHDT at 1Hz Statement: AT+GPHDT= 1\r\n
Response: OK\r\n
GPVTG (Ground velocity information)
Example: Output GPVTG at 1Hz Statement: AT+GPVTG= 1\r\n
Response: OK\r\n
GPZDA (UTC time and date)
Example: Output GPZDA at 1Hz Statement: AT+GPZDA= 1\r\n
Response: OK\r\n
GPATT (Customized message)
Example: Output GPATT at 1Hz Statement: AT+GPATT= 1\r\n
Response: OK\r\n
If the data output stops
Command: AT+SETNO\r\n
Response: OK\r\n

### 9.2.11 Stop the Output of the Current Data Stream

Command: AT+SETNO\r\n
Response: OK\r\n

### 9.2.12 Configure the Data Output Frequency

If the configuration data output frequency is 10Hz, the configuration command is:
Command: AT+OUTRATE=10\r\n

Response: OK\r\n

### 9.2.13 Configure the Baud Rate

Only support the configured baud rate 115200 and 230400, and the default is 115200.
If you configure the IMU serial baud rate to 230400, the configuration command is:
Command: AT+BAUD=230400\r\n
Response: BAUD=230400\r\n
Note: It is necessary to power off and restart after configuring and saving the commands.

### 9.2.14 Configure CAN Serial Baud Rate

Only support the configured baud rate 250K, 500K, 1M, and the default is 500K.
Example: If you configure the CAN serial baud rate to 500K, the configuration command is:
Command: AT+CAN_BAUD=500\r\n
Response: OK
Note: It is necessary to power off and restart after configuring and saving the commands.

### 9.2.15 Coinfigurate CAN Serial Output Frequancy

If the configuration CAN data output frequency is 10Hz, the configuration command is:
Command: AT+CAN_ODR=10
Response: OK\r\n
Note: Maximum support for 100HZ, the modification will take effect immediately and will continue to take effect after saving, power off and restart.

### 9.2.16 Output all Configuration Information

If you want to query all the configured information, command:
AT+CONFIG\r\n

### 9.2.17 Querying Version Numbers

Command: AT+VERSION\r\n

### 9.2.18 Configure Output Carrier XYZ Speed

After Configuration and Initialization, the North, East and Ground Speed in the integrated data Stream will switch to Carrier's XYZ Velocity in three axes.
1 means open the carrier's speed output, while all other parameters means close.
Command: AT+NAV_OUTPUT_XYZ=1\r\n
Response: OK

### 9.2.19 Acceleration Minus Gravity

1. To minus gravity, command: AT+DEDUCTIONG=1\r\n
Return OK after sending means successfully sent and will be effective immediately.
Command: AT+SAVE\r\n to save the configuration, otherwise it will fail after rebooting.
2. To restore gravity, command: AT+DEDUCTIONG=0\r\n

Return OK after sending means successfully sent and will be effective immediately.
Command: AT+SAVE\r\n to save the configuration, otherwise it will fail after rebooting.

### 9.2.20 Configure the Mounting Rotation Angle

Currently only the following rotation angles are supported
x-axis rotation 180°
z-axis rotation 90° 180° 270°
The configuration commands are as follows
If the mounting rotation angle is 180° on the X-axis, the configuration command is:
Command: AT+INSTALL_ANGLE=180,0,0\r\n
Response: INST_ANGLE_X=180.000,INST_ANGLE_Y=0.000,INST_ANGLE_Z=0.000
If the mounting rotation angle is 180° on the Z-axis, the configuration command is:
Command: AT+INSTALL_ANGLE=0,0,180\r\n
Response: INST_ANGLE_X=0.000,INST_ANGLE_Y=0.000,INST_ANGLE_Z=180.000

### 9.2.21 Turn on the Slope Mode

For scenes where there is always a slope of more than 5°, you can turn on the slope mode for compensation.
The configuration command is:
AT+SLOPE=0.05\r\n Agricultural scenarios with slopes can be configured with this statement, which defaults to 0.
Then enter AT+SAVE\r\n to save the configuration and it will work after reboot.

### 9.2.22 Save Parameters

The above parameters include lever arms, data stream, output frequency, etc. After configuration, you need to enter the AT+SAVE command and restart the software.
Command: AT+SAVE\r\n
Response: OK\r\n

## 9.3 Use CAN Interface to Configure Parameters

Note: All CAN messages are in little-endian mode with low byte in the front.

### 9.3.1 Querying Version Numbers

Example:
ID = 0x19FFF326
Data = 0x9A 0x07 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF
Response
Example: (Version 1805)
ID = 0x19FFF29A
Data = 0x26 0xC7 0x00 0x00 0x02 0x02 0xFF 0xFF
Note: 0x26 0xC7 are fixed frame header

### 9.3.2 Configure the Wheelbases and Lever Arms

Example
ID = 0X0DFFC126
Data = wheelbase 2 bytes + lever arm 2 bytes*3 (unit: cm)
Response
ID = 0X19FFC09A
Data = wheelbase 2 bytes + lever arm 2 bytes*3 (unit: cm)

### 9.3.3 Configure the Dual Antenna Mounting Angle

ID = 0X0DFFC326
Data = angle (int16, unit: deg*100) + 0xFF*6

### 9.3.4 Query the Wheelbases, Lever Arms and Dual Antenna Mounting Angle

ID = 0X0DFFC226
Data =0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF
Response
ID = 0X19FFC29A
Data = wheelbase 2 bytes + lever arm 2 bytes*3 (unit: cm)
Mounting angle query command:
ID = 0X0DFFC526
Data =0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF
Response
ID = 0X19FFC29A
Data = angle*100 2 bytes

### 9.3.5 Configure CAN Interface to Output Update Rates and Baud Rates

Baud rates: 1M, 500K, 250K (0x01, 0x02, 0x03 respectively)
Update rates: 20, 50, 100Hz (0x14, 0x32, 0x64 respectively).
After configuration, restart the software.
ID = 0x19FFF326
Data = 0x9A 0x4A Baud rate Update Rate 0xFE 0xFB 0xF9 0xFF
Response
ID =0x19FFF59A
Data =0xFF 0x4A Baud rate Update Rate 0xFF 0xFF 0xFF 0xFF

### 9.3.6 Query Update Rates and Baud Rates

ID = 0x19FFF426
Data =0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF
Response
ID =0x19FFF49A
Data =0xFF 0x4A Baud Rate Update Rate 0XFF 0XFF 0XFF 0xFF

### 9.3.7 Save Parameter Command

ID = 0X0DFFCF26
Data =0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF
Response
ID = 0X19FFCF9A

# 10. Directional Accuracy Test

Test conditions:

- Use a tractor for turning and straight line tests at 1 to 18km/h; The heading angle from the Unicore UM982 board is used as the true value reference;
- Baseline is 1.5 meters long;
- The heading angle of single-antenna INS modules is synchronized with that of RTK by using PPS signal.

Test results:

- The heading angle accuracy of single antenna INS modules converges to no more than 1° within 5s after the start;

Fig. 8 Directional Accuracy Variance information Diagram

- The heading angle accuracy of single antenna INS modules is lower than 0.3 ° rms.

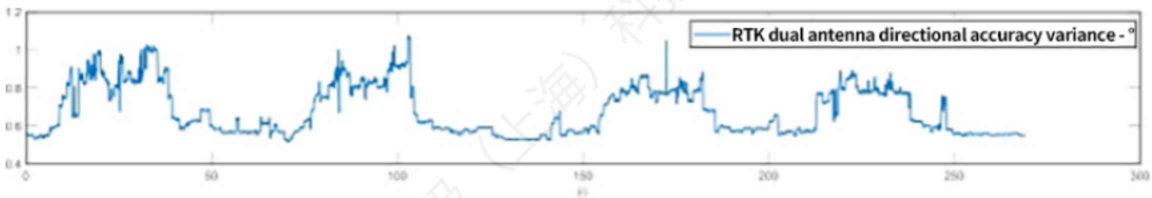
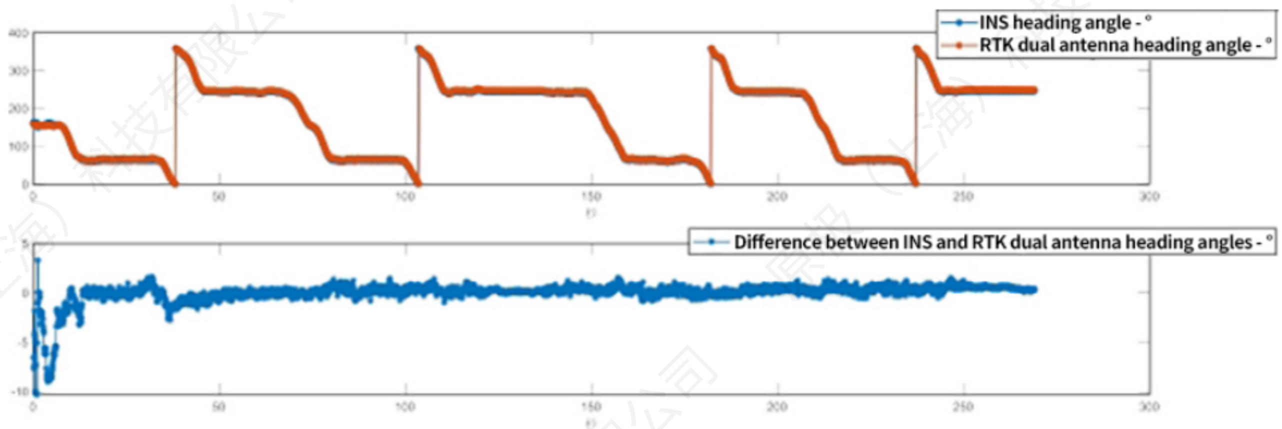
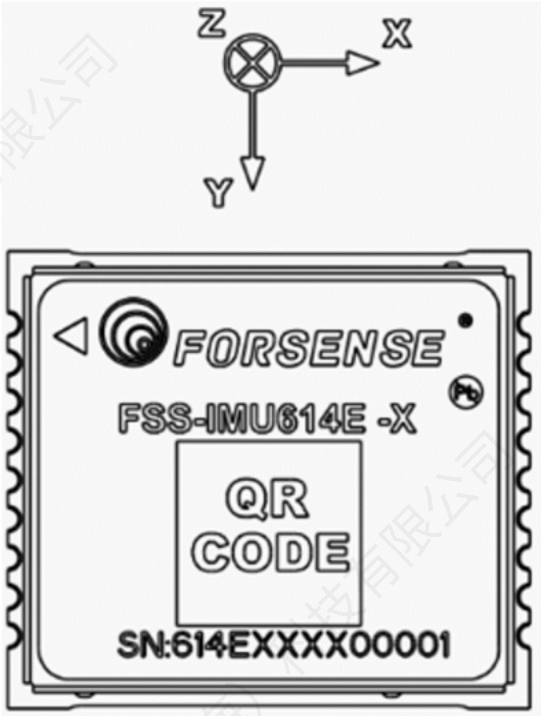



Fig. 9 Diagram of Roll, Pitch, and Heading Angles



# 11. Definition of Coordinate System

Fig. 10 Coordinate System Diagram




This product is equipped with the Front-Right-Down (FRD) coordinate system, and the range of Euler angles is as follows:

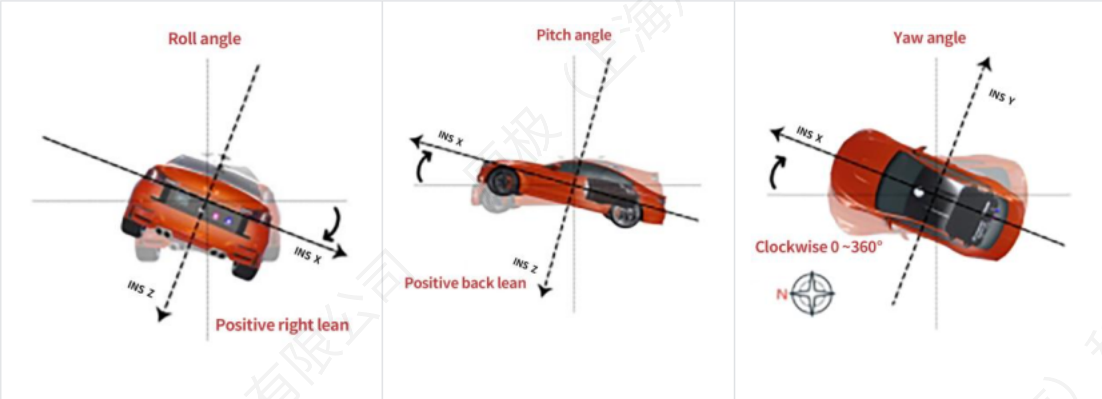
Rotate around Z-axis: Yaw angle: 0° ~360°;

Rotate around X-axis: Roll angle: -180°~180°;

Rotate around Y-axis: Pitch angle: -90°~90°。

The specific diagram is shown below:

Fig. 11 Diagram of Roll, Pitch, and Heading Angles


Roll angle
INS X
INS Z
Positive right lean
Pitch angle
INS X
INS Z
Positive back lean
Yaw angle
INS Y
INS X
Clockwise 0 ~360°
N

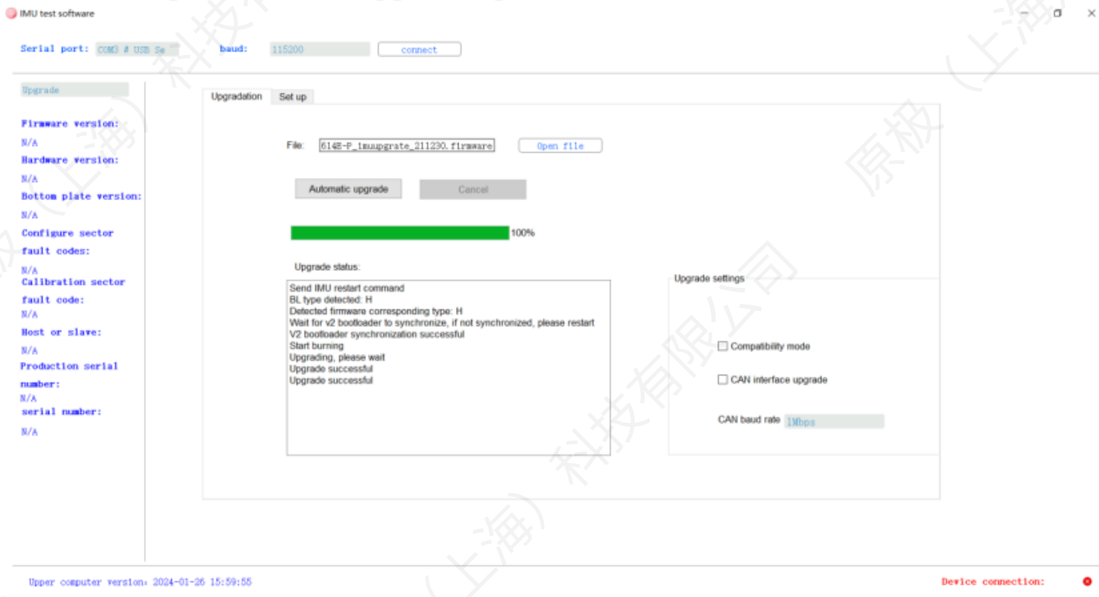
# 12. Firmware Upgrading

## 12.1 Firmware Upgrading via the PC Software

### 12.1.1 R232 Serial Port Direct Connection

Use FSS IMUs to test the PC software - Select firmware upgrade - Open the firmware - Click Auto Upgrade.

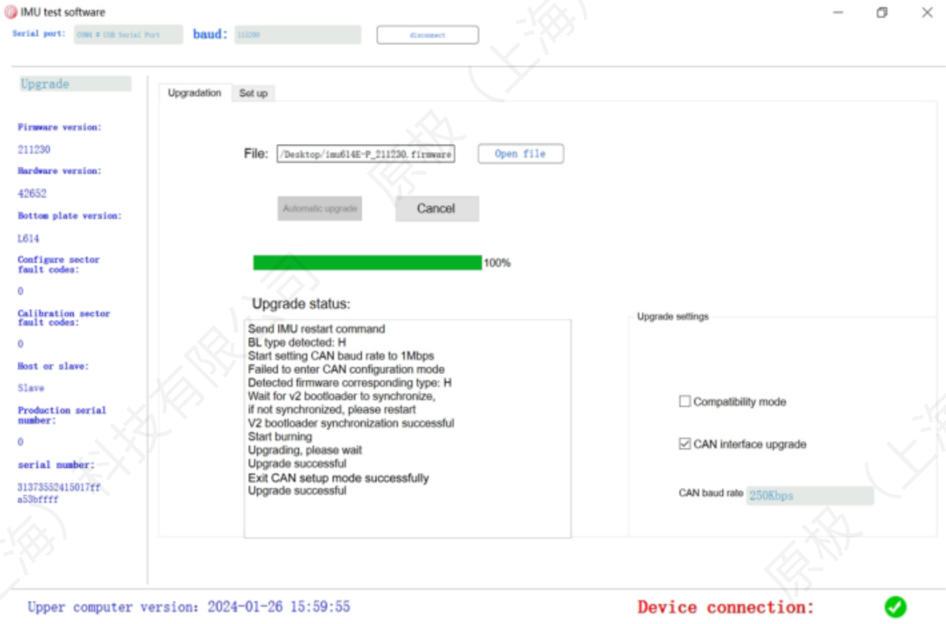
Fig. 12 Firmware Upgrade Page in Forsense PC Software - Serial Port




### 12.1.2 Use the Included USB to CAN Module

Use FSS IMUs to test the PC software - Select firmware upgrade - Open the firmware - Select the CAN interface upgrade - Set the baud rate of the upgraded firmware - Click Auto Upgrade.

Fig. 13 Firmware Upgrade Page in Forsense PC Software - USB to CAN




## 12.2 Serial OTA

The upgrade process is divided into the following steps:

Step 1: Send the upgrade command

Send upgrade command to the IMU module. After that, the device will record the upgrade flag bit in a certain area of flash and then perform a soft reboot. At this time, the module will enter the BOOTLOADER.

```
The PC software sends the upgrade command as follows:
  cmd_bl[34] ={0x55,0xaa,0xbb,0x88,0x18,0x00,0x00,0x00,0xc8,0x42,0x00,0x00,
  0x48,0x43,0x2c,0x01,0x00,0x00,0x90,0x01,0x00,0x00,0xf4,0x01,0x00,0x00,
  0x58,0x02,0x00,0x00,0x40,0x97,0x46,0x6a};
  Then the imu module will reboot and enter the bootloader
```

Step 2: Send the **HC32MCU_FORSENSE** string

After the IMU module enters the bootloader, it will actively send the HC32_UPLOADER string. During this period, the PC software needs to send the HC32MCU_FORSENSE string. If the module receives the string, it will not jump to APP, but stay in a pending upgrade state in the bootloader, and will also stop actively sending any messages.

```
  //After sending the upgrade command, send the following string to ensure that the imu module will not
jump to the APP area but stay in a pending upgrade state
   QString str = "HC32MCU_FORSENSE" ;
   for(int i=0;i<10;i++)
   {
     _port_device->write(str.toLatin1()); sleep_ms(50);
   }
```

Step 3: Send the synchronization command

When the IMU module stays in a pending upgrade state in the bootloader, it will not send messages actively, but respond to the commands of the PC software passively. At this time, the PC software enters Send_CMD_LONG(0X21,0,0,0,0,0,0,0,0) and waits for the response from the IMU module. After receiving the command, IMU immediately replies with Send

CMD ACK (0x64,0x10,0).

Once the PC software determines that it has received the response data, it indicates that the synchronization is completed.

```
  Synchronization process between the PC software and imu module:
  1.The PC software sends the synchronization command with the command code of 0x21, and then waits for the imu module to respond.
   Send_CMD_LONG(0x21,0,0,0,0,0,0);
  2.After the imu module receives the command code of 0x21, it will send the response data with the command as follows:
  Send_CMD_ACK(0x64,0x10,0);
  3.The PC software determines whether it has received the response data from the imu. If received, the synchronization process is completed.
```

Step 4: Send erase command

The PC software sends the erase command Send_CMD_LONG(0x23,0,0,0,0,0,0), and then the IMU module will erase everything in the APP area and send the final execution result back to the PC software. The PC software will decide whether to send erase command again according to the result. Once the erase operation is successful, the APP of IMU module cannot be restored. In addition, the PC software must wait for the response of successful erase before proceeding to the next step, otherwise it may cause the subsequent upgrade failure.

```
1.The PC software sends the erase command with the command code of 0x23, and then waits patiently for imu to reply with the execution result.
  Send_CMD_LONG(0x23,0,0,0,0,0,0);
2.After imu receives the command code of 0x23, it will erase the APP data and reply the final execution result to the PC software.
   send_CMD_ACK(0x64,0x10,0); //Indicating that the erase process is successful
   send_CMD_ACK(0x64,0x11,0); //Indicating that the erase process fails and needs to operate again
   Note: The PC software must receive a successful erase response before proceeding to the next step.
```

Step 5: Send upgrade packets

After a successful erase operation, it will enter the most important part of sending firmware data by using the function Send_Upload_Data. The PC software will divide firmware data into packets, and each packet has a fixed size of 64 bytes. If the last packet contains fewer than 64 bytes, it will be sent with the actual number of bytes. Each data frame contains the valid data length and the offset address of this packet within the entire firmware. Every time the PC software sends a single data frame, it must wait for the response from the IMU module and ensure that IMU has successfully obtained this frame data before sending the next. After the IMU module successfully receives the data packet from the PC software, it will send the response data and write it to the specified flash address according to the offset address.

If the writing process fails, it will send failure command; otherwise it will send nothing.

```
 Example: send upgrade file uint8_t Upgrade_Data[1000] of 1000 bytes to imu module
  //Send the 1st packet:
   1.1 Send 0~63 bytes to imu module by using the function
   Send_Upload_Data (0x27,0,0,0x40,Upgrade_Data);
   //The first parameter 0x27 of the above function is a fixed value, the second parameter 0 is a fixed
value, the third parameter 0 is the offset address, the fourth parameter 0x40 is the valid byte length, and
the fifth parameter is the first address of the sent data.

  1.2 After imu successfully receives the data, it will send the response data and write it to the
specified flash address according to the offset address
    send_CMD_ACK(0x753D,0x00,0);
    //The first parameter of the above function 0x753D is a fixed value, the second parameter 0 is a
fixed value, and the third parameter is the offset address.
    The imu module will send the failure command send_CMD_ACK(0x64,0x11,0) if it fails to write data to
flash;

    //Send the 2nd packet:
    2.1 After the PC software receives the response data from imu, it will send the 2nd data packet
Send_Upload_Data (0x27,0,0x40,0x40,Upgrade_Data+0x40);
    2.2 After the imu successfully receives the 2nd data packet, it will send the response:
send_CMD_ACK(0x753D,0x00,0x40);
    2.3 If the imu module fails to write data to flash, it will send the failure command
send_CMD_ACK(0x64,0x11,0);


    //Send the 3rd packet:
    3.1 After the PC software receives the response data from imu, it will send the 3rd data packet
(0x27,0,0x80,0x40,Upgrade_Data+0x80);
    3.2 After the imu successfully receives the 3rd data packet, it will send the response:
send_CMD_ACK(0x753D,0x00,0x80);
    3.3 The imu module will send the failure command if it fails to write data to flash
send_CMD_ACK(0x64,0x11,0);
   ——————————
   //Send the 15th packet:
    15.1 After the PC software receives the response data from imu, it will send the 15th data packet
Send_Upload_Data (0x27,0,0x380,0x40,Upgrade_Data+0x380);
    15.2 After the imu successfully receives the 15th data packet, it will send the response:
send_CMD_ACK(0x753D,0x00,0x380);
    15.3 The imu module will send the failure command if it fails to write data to flash
send_CMD_ACK(0x64,0x11,0);
    //Send the 16th packet:
    15.1 After the PC software receives the response data from imu, it will send the 16th data packet
Send_Upload_Data (0x27,0,0x3C0,0x28,Upgrade_Data+0x3C0);
```

```
     15.2 After the imu successfully receives the 16th data packet, it will send the response:
send_CMD_ACK(0x753D,0x00,0x3C0);
     15.3 The imu module will send the failure command if it fails to write data to flash
send_CMD_ACK(0x64,0x11,0);
     END
```

Step 6: Acquire **CRC** check digits

Generally speaking, the command rule for upgrading firmware is imu614e-b#CRC1373387121.firmware, and the CRC string is immediately followed by the calculated CRC checksum value. After the upgrade file is sent, the PC software needs to send a check command to determine if the upgrade file received by the IMU module is correct. After the PC software enters Send_CMD_LONG(0x29,0,0,0,0,0,0,0,0) command, it will get the CRC check digits calculated by the IMU module itself.

If the PC software judges that there is an error in the CRC check digits, it will restart the upgrade process by resuming the 4th step of sending erase command.

```
     The PC software sends the command to acquire the crc check digits, and waits for the imu response
      Send_CMD_LONG(0x29,0,0,0,0,0,0,0);
     The imu module responds by sending the crc checksum value data:
     send_CMD_ACK(0x753C,0x10,crc32_data);
      The crc32_data value is calculated by the imu module itself.
```

Step 7: Send reboot command

After the PC software judges that the CRC check digits are correct, it will send reboot command, indicating successful upgrading.

```
   After determining that the crc checksum value is correct, send the reboot command:
      Send_CMD_LONG(0x30,0,0,0,0,0,0,0);
```

When firmware upgrade is complete, powering off and rebooting to determine if the upgrade is successful by reading the version number.

Function definitions:

1. The definition of Send_CMD_LONG is as follows:

```
struct  MULTI_LONG_CMD_STRUCT
{
uint8_t header1;
uint8_t header2;
uint16_t id;
uint16_t length;
float param1;
float param2;
uint32_t param3;
uint32_t param4;
int32_t param5;
int32_t param6;
uint32_t check_crc;
}__attribute__((packed));
```

```
void  :Send_CMD_LONG(uint16_t cmd_id,float cm1,float cm2,uint32_t cm3,uint32_t cm4,int32_t cm5,int32_t cm6)
{
    uint8_t check_sum=0;
    struct MULTI_LONG_CMD_STRUCT data_cmd_long __attribute__((packed));
    data_cmd_long.header1=0x55;
    data_cmd_long.header2=0xAA;
    data_cmd_long.id=cmd_id;
    data_cmd_long.length=sizeof(data_cmd_long)-10;
    data_cmd_long.param1=cm1;
    data_cmd_long.param2=cm2;
    data_cmd_long.param3=cm3;
    data_cmd_long.param4=cm4;
    data_cmd_long.param5=cm5;
    data_cmd_long.param6=cm6;
    int len=sizeof(data_cmd_long)-4;
    uint32_t check_crc=1;
    data_cmd_long.check_crc=crc_crc32(check_crc,(uint8_t *)(&data_cmd_long), len);
    send((uint8_t *)(&data_cmd_long),sizeof(data_cmd_long));
}
```

2. The definition of Send_CMD_ACK is as follows:

```
struct  CMD_ACK_STRUCT
{
uint8_t header1;
uint8_t header2;
uint16_t id;
```

```
uint16_t length;
uint32_t command;/*<  Command ID (of acknowledged command).*/
uint32_t result;/*<  Result of command.*/
uint32_t check_crc;
}__attribute__((packed));

void Send_CMD_ACK(uint16_t cmd_id,uint16_t ack_id,uint32_t result)
{
    uint32_t check_crc=0;
    struct CMD_ACK_STRUCT data_cmd_ack __attribute__((packed));
    data_cmd_ack.header1=0xAA;
    data_cmd_ack.header2=0x55;
    data_cmd_ack.id=cmd_id;
    data_cmd_ack.length=sizeof(data_cmd_ack)-10;
    data_cmd_ack.command=ack_id;
    data_cmd_ack.result=result;
    int len=sizeof(data_cmd_ack)-4;
    check_crc=1;
    data_cmd_ack.check_crc=crc_crc32(check_crc,(uint8_t *)(&data_cmd_ack), len);
    Cout((uint8_t *)(&data_cmd_ack),sizeof(data_cmd_ack));
}
```

3. The definition of Send_Upload_Data is as follows:

```
struct  UPLOAD_DATA
{
uint8_t header1;
uint8_t header2;
uint16_t id;
uint16_t length;
uint8_t param[64];
uint32_t offset;
uint16_t size;
uint8_t cmd;
uint32_t check_crc;
}__attribute__((packed));

struct UPLOAD_DATA upload_data;
void Send_Upload_Data(uint8_t cmd_id,uint8_t cmd,uint32_t offset,uint16_t size,uint8_t* param)
{
    upload_data.header1=0x55;
```

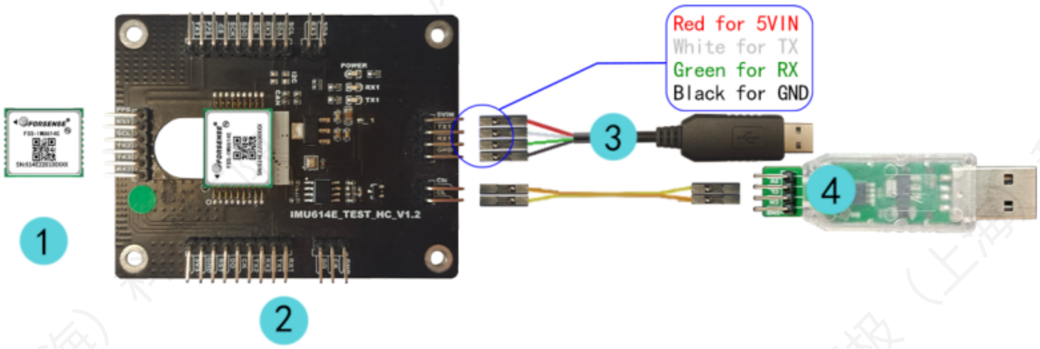
```
    upload_data.header2=0xAA;
    upload_data.id=cmd_id;
    upload_data.length=sizeof(UPLOAD_DATA)-10;
    upload_data.cmd=cmd;
    for(int i=0;i<size;i++)
    upload_data.param[i] = *(param+i);
    upload_data.offset=offset;
    upload_data.size=size;
    int len=sizeof(UPLOAD_DATA)-4;
    uint32_t check_crc=1;
    upload_data.check_crc=crc_crc32(check_crc,(uint8_t *)(&upload_data), len);
    send((uint8_t *)(&upload_data),sizeof(UPLOAD_DATA));
}
```

4. The definition of CRC32 checksum function is as follows:

```
static const uint32_t crc32_tab[] = {
    0x00000000, 0x77073096, 0xee0e612c, 0x990951ba, 0x076dc419, 0x706af48f,
    0xe963a535, 0x9e6495a3, 0x0edb8832, 0x79dcb8a4, 0xe0d5e91e, 0x97d2d988,
    0x09b64c2b, 0x7eb17cbd, 0xe7b82d07, 0x90bf1d91, 0x1db71064, 0x6ab020f2,
    0xf3b97148, 0x84be41de, 0x1adad47d, 0x6ddde4eb, 0xf4d4b551, 0x83d385c7,
    0x136c9856, 0x646ba8c0, 0xfd62f97a, 0x8a65c9ec, 0x14015c4f, 0x63066cd9,
    0xfa0f3d63, 0x8d080df5, 0x3b6e20c8, 0x4c69105e, 0xd56041e4, 0xa2677172,
    0x3c03e4d1, 0x4b04d447, 0xd20d85fd, 0xa50ab56b, 0x35b5a8fa, 0x42b2986c,
    0xdbbbc9d6, 0xacbcf940, 0x32d86ce3, 0x45df5c75, 0xdcd60dcf, 0xabd13d59,
    0x26d930ac, 0x51de003a, 0xc8d75180, 0xbfd06116, 0x21b4f4b5, 0x56b3c423,
    0xcfba9599, 0xb8bda50f, 0x2802b89e, 0x5f058808, 0xc60cd9b2, 0xb10be924,
    0x2f6f7c87, 0x58684c11, 0xc1611dab, 0xb6662d3d, 0x76dc4190, 0x01db7106,
    0x98d220bc, 0xefd5102a, 0x71b18589, 0x06b6b51f, 0x9fbfe4a5, 0xe8b8d433,
    0x7807c9a2, 0x0f00f934, 0x9609a88e, 0xe10e9818, 0x7f6a0dbb, 0x086d3d2d,
    0x91646c97, 0xe6635c01, 0x6b6b51f4, 0x1c6c6162, 0x856530d8, 0xf262004e,
    0x6c0695ed, 0x1b01a57b, 0x8208f4c1, 0xf50fc457, 0x65b0d9c6, 0x12b7e950,
    0x8bbeb8ea, 0xfcb9887c, 0x62dd1ddf, 0x15da2d49, 0x8cd37cf3, 0xfbd44c65,
    0x4db26158, 0x3ab551ce, 0xa3bc0074, 0xd4bb30e2, 0x4adfa541, 0x3dd895d7,
    0xa4d1c46d, 0xd3d6f4fb, 0x4369e96a, 0x346ed9fc, 0xad678846, 0xda60b8d0,
    0x44042d73, 0x33031de5, 0xaa0a4c5f, 0xdd0d7cc9, 0x5005713c, 0x270241aa,
    0xbe0b1010, 0xc90c2086, 0x5768b525, 0x206f85b3, 0xb966d409, 0xce61e49f,
    0x5edef90e, 0x29d9c998, 0xb0d09822, 0xc7d7a8b4, 0x59b33d17, 0x2eb40d81,
    0xb7bd5c3b, 0xc0ba6cad, 0xedb88320, 0x9abfb3b6, 0x03b6e20c, 0x74b1d29a,
    0xead54739, 0x9dd277af, 0x04db2615, 0x73dc1683, 0xe3630b12, 0x94643b84,
    0x0d6d6a3e, 0x7a6a5aa8, 0xe40ecf0b, 0x9309ff9d, 0x0a00ae27, 0x7d079eb1,
    0xf00f9344, 0x8708a3d2, 0x1e01f268, 0x6906c2fe, 0xf762575d, 0x806567cb,
    0x196c3671, 0x6e6b06e7, 0xfed41b76, 0x89d32be0, 0x10da7a5a, 0x67dd4acc,
```

# 13. Connection Diagram of Test Base Plate

Fig. 24 Connection Diagram of Module and PC software




| | Name | Quantity |
|---|---|---|
| 1 | IMU 614E Series Module | 1 |
| | | |
| | **Accessories Name** | **Quantity** |
| 2 | CAN Test Substrates | 1 |
| 3 | 4-PIN Connector | 1 |
| 4 | USB to CAN Module | 1 |

# 14. ROS Driver (INS Data Stream)

ROS driver download address: https://data.forsense-imu.com/page/download.html


## 14.1 Install ROS Serial

Install the ROS serial package. This routine relies on the serial package provided by ROS to realize serial communication.

First, execute the following command to download and install the serial package:

sudo apt-get install ros -melodic-serial

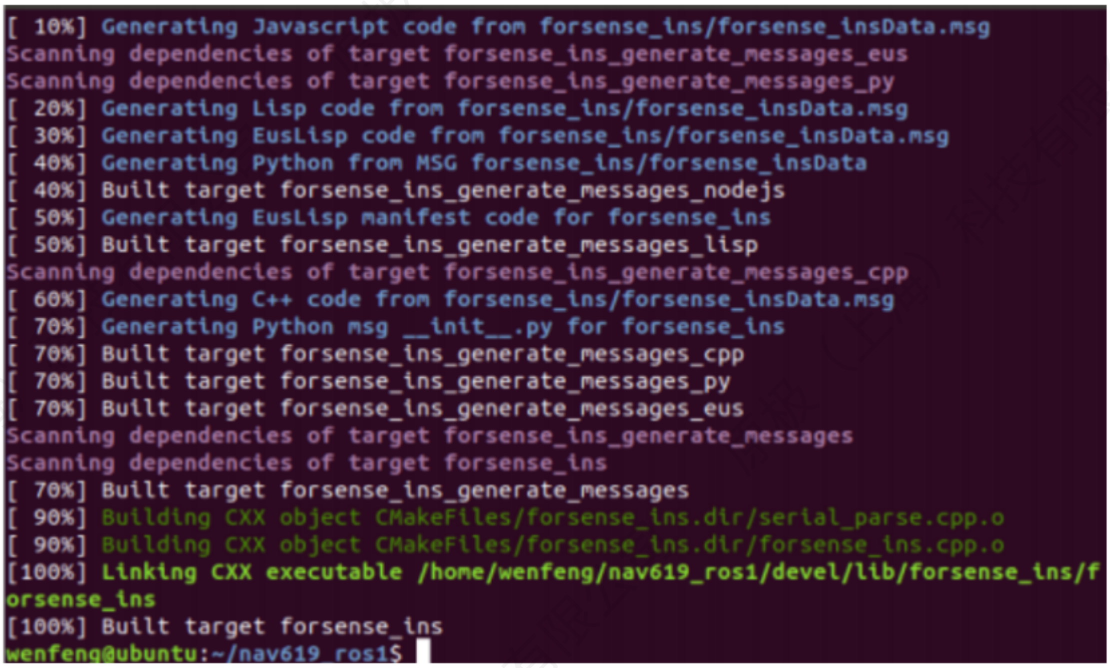
Second, enter the roscd serial command to find the serial download location. If the installation is successful, the following message will appear:

/opt/ros/melodic/share/serial

## 14.2 Compile Code

cd FS982_ros/

catkin_make



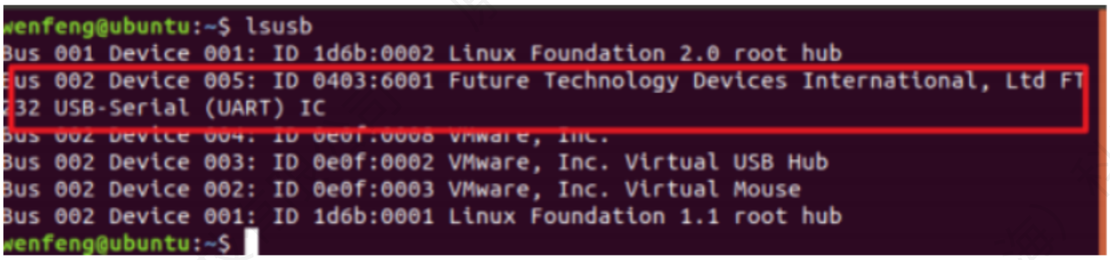
```
[ 10%] Generating Javascript code from forsense_ins/forsense_insData.msg
Scanning dependencies of target forsense_ins_generate_messages_eus
Scanning dependencies of target forsense_ins_generate_messages_py
[ 20%] Generating Lisp code from forsense_ins/forsense_insData.msg
[ 30%] Generating EusLisp code from forsense_ins/forsense_insData.msg
[ 40%] Generating Python from MSG forsense_ins/forsense_insData
[ 40%] Built target forsense_ins_generate_messages_nodejs
[ 50%] Generating EusLisp manifest code for forsense_ins
[ 50%] Built target forsense_ins_generate_messages_lisp
Scanning dependencies of target forsense_ins_generate_messages_cpp
[ 60%] Generating C++ code from forsense_ins/forsense_insData.msg
[ 70%] Generating Python msg __init__.py for forsense_ins
[ 70%] Built target forsense_ins_generate_messages_cpp
[ 70%] Built target forsense_ins_generate_messages_py
[ 70%] Built target forsense_ins_generate_messages_eus
Scanning dependencies of target forsense_ins_generate_messages
Scanning dependencies of target forsense_ins
[ 70%] Built target forsense_ins_generate_messages
[ 90%] Building CXX object CMakeFiles/forsense_ins.dir/serial_parse.cpp.o
[ 90%] Building CXX object CMakeFiles/forsense_ins.dir/forsense_ins.cpp.o
[100%] Linking CXX executable /home/wenfeng/nav619_ros1/devel/lib/forsense_ins/f
orsense_ins
[100%] Built target forsense_ins
wenfeng@ubuntu:~/nav619_ros1$
```

Compilation completed

## 14.3 Connect the IMU to Linux System via USB

Check if it is connected:

lsusb



```
wenfeng@ubuntu:~$ lsusb
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 002 Device 005: ID 0403:6001 Future Technology Devices International, Ltd FT
232 USB-Serial (UART) IC
Bus 002 Device 004: ID 0e0f:0008 VMware, Inc.
Bus 002 Device 003: ID 0e0f:0002 VMware, Inc. Virtual USB Hub
Bus 002 Device 002: ID 0e0f:0003 VMware, Inc. Virtual Mouse
Bus 002 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
wenfeng@ubuntu:~$
```

Check the USB port number:

ls /dev/ttyU*

```
wenfeng@ubuntu:~$ ls /dev/ttyU*
/dev/ttyUSB0
wenfeng@ubuntu:~$
```

Turn on and configure USB to serial port privileges:

sudo chmod 777 /dev/ttyUSB0

## 14.4 Check Data

Execute roscore to start ROS
Go back to the serial_imu_ws folder and run
source devel/setup.bash
Start rosrun
rosrun forsense_ins forsense_ins

```
wenfeng@ubuntu:~/nav619_ros1$ rosrun forsense_ins forsense_ins
[ INFO] [1695457979.128623440]: /dev/ttyUSB0 is opened.
```

Open a new window
source devel/setup.bash
rostopic list

```
wenfeng@ubuntu:~/nav619_ros1$ rostopic list
/nav619Data
/rosout
/rosout_agg
```

Enter a command to view IMU data
rostopic echo /FS982Data

```
  frame_id: "WGS84"
itow: 549636980
week_num: 2280
lat: 312627286
lon: 1216155393
hgt: 38859
vn: 0.00240602344275
ve: 0.000262897461653
vd: 0.00270945159718
roll: -0.169113516808
pitch: -0.286453634501
yaw: 0.0
rtk_yaw: 359.766906738
wheel_angle: 0.0
imu: [-0.005366197787225246, 0.0035326573997735977, -1.004271149635315, -0.04756217822432518, -0.11066819727420807, -0.06515973061323166, 35.8017578125]
fix_type: 16
sv_num: 28
diff_age: 0
heading_type: 0
pos_acc: 0
status: 3
---
```

# 15. Use with Front Wheel Gyro G200

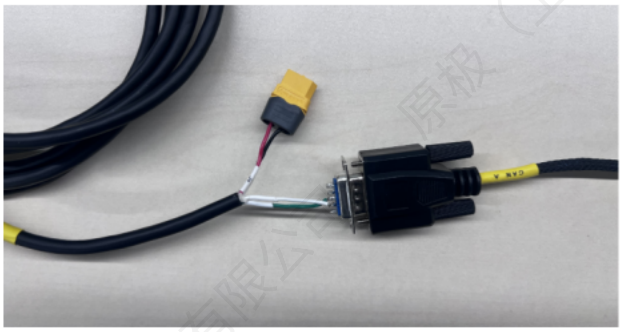
1. G200 mounting requirements:

A. G200 must be mounted on the front axle of the front wheel (a horizontally rigid position that can rotate with the front wheel);

B. G200 port is installed toward the rear of the car.

2. The CAN interface of G200 is connected to CAN A of the hub, and the power supply is 5V.

Fig. 10 Wiring Diagram



3. Measure the wheelbase of the vehicle. Take the 2m wheelbase as an example. The serial port of the host computer connects to the serial port assistant, sends AT+WHEEL_BASE=2, receives the response WHEEL_BASE=2, and then enters AT+SAVE. If it receives the response OK, the device needs to be re-powered.

4. Connect the device to FS982-AG decoding PC software V3.3.4. Enter the home page of the PC software, select the serial port number and baud rate (default 115200), and then click 'Connect'. The device information on the left side indicates a successful connection. Click 'Mode Activation' in the lower-left corner, and reach correct access to G200. The valid bit for the front wheel gyroscope will be displayed as green; otherwise, it will be red. Once the vehicle initialization is successful, the valid bit for the front wheel angle will be displayed as green; otherwise, it will be red. The front wheel angle data will be output in real time when the vehicle is travelling.

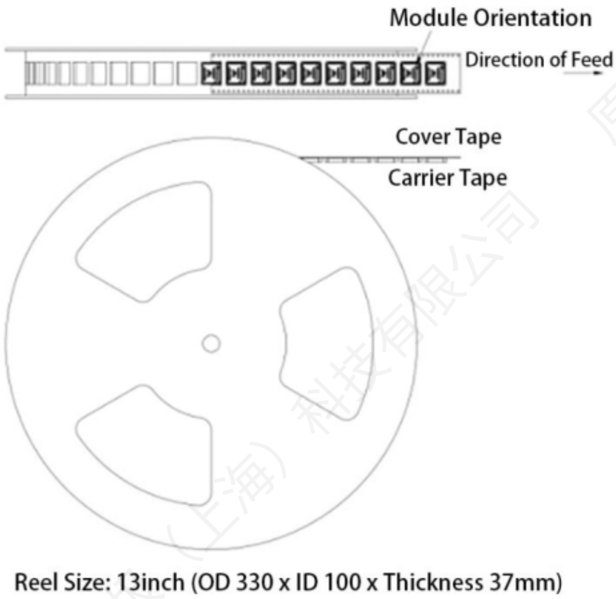
Fig. 11 Correct Access to G200

# 16. Packaging

The IMU614E-AG module is packaged in sealed tape and reel, which contributes to efficient production.

## 16.1 Tape and Reel Packaging

Fig. 15 Tape and Reel Packaging Diagram




## 16.2 Carrier Tape

The IMU614E-AG module is placed on the carrier tape in the position and orientation shown below before leaving the factory:

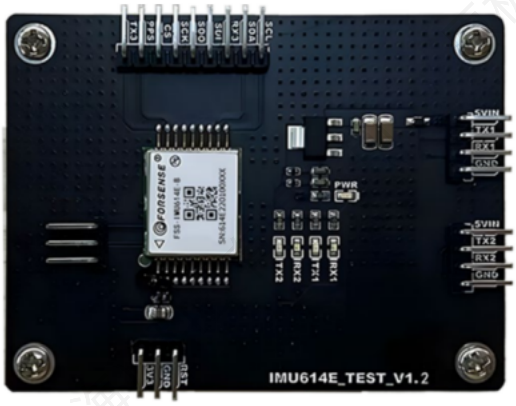
Fig. 16 Module Position and Orientation on Carrier Tape

# 17. CRC Look-up Table Calculations

```C++
static const uint32_t crc32_tab [ ] = {
0x00000000, 0x77073096, 0xee0e612c, 0x990951ba, 0x076dc419, 0x706af48f,
0xe963a535, 0x9e6495a3, 0x0edb8832, 0x79dcb8a4, 0xe0d5e91e, 0x97d2d988,
0x09b64c2b, 0x7eb17cbd, 0xe7b82d07, 0x90bf1d91, 0x1db71064, 0x6ab020f2,
0xf3b97148, 0x84be41de, 0x1adad47d, 0x6ddde4eb, 0xf4d4b551, 0x83d385c7,
0x136c9856, 0x646ba8c0, 0xfd62f97a, 0x8a65c9ec, 0x14015c4f, 0x63066cd9,
0xfa0f3d63, 0x8d080df5, 0x3b6e20c8, 0x4c69105e, 0xd56041e4, 0xa2677172,
0x3c03e4d1, 0x4b04d447, 0xd20d85fd, 0xa50ab56b, 0x35b5a8fa, 0x42b2986c,
0xdbbbc9d6, 0xacbcf940, 0x32d86ce3, 0x45df5c75, 0xdcd60dcf, 0xabd13d59,
0x26d930ac, 0x51de003a, 0xc8d75180, 0xbfd06116, 0x21b4f4b5, 0x56b3c423,
0xcfba9599, 0xb8bda50f, 0x2802b89e, 0x5f058808, 0xc60cd9b2, 0xb10be924,
0x2f6f7c87, 0x58684c11, 0xc1611dab, 0xb6662d3d, 0x76dc4190, 0x01db7106,
0x98d220bc, 0xefd5102a, 0x71b18589, 0x06b6b51f, 0x9fbfe4a5, 0xe8b8d433,
0x7807c9a2, 0x0f00f934, 0x9609a88e, 0xe10e9818, 0x7f6a0dbb, 0x086d3d2d,
0x91646c97, 0xe6635c01, 0x6b6b51f4, 0x1c6c6162, 0x856530d8, 0xf262004e,
0x6c0695ed, 0x1b01a57b, 0x8208f4c1, 0xf50fc457, 0x65b0d9c6, 0x12b7e950,
0x8bbeb8ea, 0xfcb9887c, 0x62dd1ddf, 0x15da2d49, 0x8cd37cf3, 0xfbd44c65,
0x4db26158, 0x3ab551ce, 0xa3bc0074, 0xd4bb30e2, 0x4adfa541, 0x3dd895d7,
0xa4d1c46d, 0xd3d6f4fb, 0x4369e96a, 0x346ed9fc, 0xad678846, 0xda60b8d0,
0x44042d73, 0x33031de5, 0xaa0a4c5f, 0xdd0d7cc9, 0x5005713c, 0x270241aa,
0xbe0b1010, 0xc90c2086, 0x5768b525, 0x206f85b3, 0xb966d409, 0xce61e49f,
0x5edef90e, 0x29d9c998, 0xb0d09822, 0xc7d7a8b4, 0x59b33d17, 0x2eb40d81,
0xb7bd5c3b, 0xc0ba6cad, 0xedb88320, 0x9abfb3b6, 0x03b6e20c, 0x74b1d29a,
0xead54739, 0x9dd277af, 0x04db2615, 0x73dc1683, 0xe3630b12, 0x94643b84,
0x0d6d6a3e, 0x7a6a5aa8, 0xe40ecf0b, 0x9309ff9d, 0x0a00ae27, 0x7d079eb1,
0xf00f9344, 0x8708a3d2, 0x1e01f268, 0x6906c2fe, 0xf762575d, 0x806567cb,
0x196c3671, 0x6e6b06e7, 0xfed41b76, 0x89d32be0, 0x10da7a5a, 0x67dd4acc,
0xf9b9df6f, 0x8ebeeff9, 0x17b7be43, 0x60b08ed5, 0xd6d6a3e8, 0xa1d1937e,
0x38d8c2c4, 0x4fdff252, 0xd1bb67f1, 0xa6bc5767, 0x3fb506dd, 0x48b2364b,
0xd80d2bda, 0xaf0a1b4c, 0x36034af6, 0x41047a60, 0xdf60efc3, 0xa867df55,
0x316e8eef, 0x4669be79, 0xcb61b38c, 0xbc66831a, 0x256fd2a0, 0x5268e236,
0xcc0c7795, 0xbb0b4703, 0x220216b9, 0x5505262f, 0xc5ba3bbe, 0xb2bd0b28,
0x2bb45a92, 0x5cb36a04, 0xc2d7ffa7, 0xb5d0cf31, 0x2cd99e8b, 0x5bdeae1d,
0x9b64c2b0, 0xec63f226, 0x756aa39c, 0x026d930a, 0x9c0906a9, 0xeb0e363f,
0x72076785, 0x05005713, 0x95bf4a82, 0xe2b87a14, 0x7bb12bae, 0x0cb61b38,
0x92d28e9b, 0xe5d5be0d, 0x7cdcefb7, 0x0bdbdf21, 0x86d3d2d4, 0xf1d4e242,
0x68ddb3f8, 0x1fda836e, 0x81be16cd, 0xf6b9265b, 0x6fb077e1, 0x18b74777,
0x88085ae6, 0xff0f6a70, 0x66063bca, 0x11010b5c, 0x8f659eff, 0xf862ae69,
0x616bffd3, 0x166ccf45, 0xa00ae278, 0xd70dd2ee, 0x4e048354, 0x3903b3c2,
```

```
0xa7672661, 0xd06016f7, 0x4969474d, 0x3e6e77db, 0xaed16a4a, 0xd9d65adc,
0x40df0b66, 0x37d83bf0, 0xa9bcae53, 0xdebb9ec5, 0x47b2cf7f, 0x30b5ffe9,
0xbdbdf21c, 0xcabac28a, 0x53b39330, 0x24b4a3a6, 0xbad03605, 0xcdd70693,
0x54de5729, 0x23d967bf, 0xb3667a2e, 0xc4614ab8, 0x5d681b02, 0x2a6f2b94,
0xb40bbe37, 0xc30c8ea1, 0x5a05df1b, 0x2d02ef8d,
}
uint32_t crc_crc32 (uint32_t crc, const uint8_t *buf, uint32_t size ) {
for (uint32_t i=0; i<size ; i++) {
crc = crc32_tab [ (crc ^ buf [i ] ) & 0xff] ^ (crc >> 8 ) ;
}
return crc;
}
```

# 18. Accessories




IMU614E-X Test Base Plate

TTL Serial Cable

USB to CAN Module

Type-C Cable

# 19. Frequent Questions

| Question | Solution |
|---|---|
| Unable to connect PC software | Please check the following: whether the serial port is occupied and the product is properly powered up. If PC software is disconnected during the connection process, possible reason may be loose COM, solution is re-plug the serial cable and turn on PC software again. |
| Positioning status bit is always 1 | Please make sure the differential data is accessed normally, check the following information: Baud Rate, differential account number, mount point, etc. |
| Dual antenna directional status can't reach 50 | Please check the following:<br>1. If directional status is always 0, please check if the secondary antenna is properly connected and if the antenna feeder is damaged.<br>2. If directional status is not 0 but can't reach 50, please check if the testing environment is open and if the distance between two antenna is over 50cm. |
| When connecting active antenna, the signal is normal; while connecting passive antenna, the signal can't be found. | The feeder of passive antenna can't exceed 1.5M |
| When connecting active power splitter (OPS), the antenna has a normal signal; while using the equippment to power the antenna, the signal can't be found. | Please check the following: 1. Whether the antenna's feed line is short-circuited; 2. Whether there's an abnormal power supply circuit caused by static electricity or other issue due to plugging and unplugging the antenna when powering-on. The current circuit design has already equipped with a fuse, and can be restored after re-powering up the antenna. |
| Unable to initialize | The initialization requires the following condition:<br>Single antenna: 1. Fixed solution of the positioning status; 2. Traveling speed above 0.5m/s; 3. Driving straight with acceleration and deceleration.<br>Dual antenna: 1. Fixed solution of the positioning status; 2. Orientation status 50 |
| COM packet loss | Please check the following:<br>1. The serial cable needs to support at lease 115200 Baud Rates.<br>2. Computer's serial delay needs to be configured to 2ms |
| Invalid RTK Data | Please check the following:<br>1. If the RTK board and AG module is connected according to the recommended method in Chapter 7,<br>2. If Baud Rate and import message comply with the RTK configuration requirements in Chapter 8. |
| How to identify the physical connecting of G200 is correct | Please check the data in resolution status bits, see if bit5 is valid. |
| An invalid significant bit of the front-wheel gyroscope when G200 is connected. | Please use CAN analyzer to check if the CAN baud rate of G200 and AG is in consistency, if not, please change to consistent. Please check the parameter configuration in G200 and AG product sheet for specific modification commands. |
| When connecting to G200, the significant bits of the front-wheel gyroscope is valid while the front-wheel angle is abnormal. | Please check the following:<br>1. Whether AG is completed initialization, the correct angle can only output correct data after initialization.<br>2. Whether the wheelbase is configured, if the wheelbase is not configured, the front wheel angle will be invalid. |

| | |
|---|---|
| An inconsistency between the result and configuration of CAN-configured lever arm, dual antenna angle and other messages. | Please check the following:<br>Little-endian mode, low bits places in the front, unit: cm, anf if the data is negative, there's a need for complement. |
| Is it possible to synchronize the output of nmea protocol and binary protocol? | No, only one of the protocols can be output at the same time. |
| When connecting G200, the front-wheel maximum angle between left and right is asymmetrical. | Please check the following:<br>1. If G200 is horizontal installed<br>2. If only powering up in the stationary state, the correct degree needs to be 0, other solution include manually drive straight for a period of time to correct it back. |
| The satellite number still exists after unplug the antenna. | The tracing ability of UM982 is rather strong, if there's signal source surrounds, it will continue to trace it through coupling. However, in actual satellite lockout scenario, the antenna is still in connection, the external signal can't enter, hence doesn't effect the usage. |
| When using a metal pot to cover the antenna to simulate loss of lock, the status is still fixed solution. | This is a wrong examine method because:<br>The metal pot can only cover the signal from direct above, while lateral signal are reflected from the roof to the bottom of the lid and back to the antenna, causing false fixation. |
| ROS driver unable to read the data | The device has not configured with default power-on integrated data stream, the problem can be solved after the default power-on integrated data stream is configured. |
| Large error in yaw angle | Possible reasons may include:<br>1. Wrong configuration in lever arm and wheelbase, which unit has amplified by 10 times.<br>2. Please make sure if it's rigidly fixed.<br>3. Please make sure if it's away from strong vibrant source. |
| Excessive attitude error with reference frame | Please confirm whether the installation error is deducted from the statistics. Usually, the average value of the error between the reference frame and the reference frame is taken as the installation error. |
| How to complement little-endian CAN command? | The complement targets at the negative data.The corresponding hexadecimal code for -20， X-65536=-20 X=65516 65516 is FFEC. The corresponding hexadecimal code for 65535 is FFFF. Because all data is calculated at 0, the maximum data is 65536. The current result minus the maximum value is the complementary code value. |
| Is there any effect of laying copper underneath the device? | No, but please remember to avoid high frequency signals. |

# 20. Revision History

| Version | Date | Status/Notes |
|---|---|---|
| Version 1.0 | 28/07/2023 | First release |
| Version 1.1 | 07/10/2023 | Updated coordinate system definition |
| Version 1.2 | 14/12/2023 | Added accessories |
| Version 1.3 | 04/02/2024 | Updated electrical characteristics |
| Version 1.4 | 27/02/2024 | Updated reflow welding curves and ESD protection notes |
| Version 1.5 | 26/03/2024 | Added slope mode and using methods of combining IMU614E-AG with G200 |
| Version 1.6 | 12/06/2024 | 1. Added some new AT commands<br>2. Added CAN protocol latitude, longitude, and elevation messages.<br>3. Added spreadsheet for frequent questions |
| Version 1.7 | 10/05/2024 | Added sample connection to PC software |
| Version 1.8 | 05/07/2024 | 1. Added some new AT commands<br>2. Added CAN protocol latitude, longitude, and elevation messages.<br>3. Added spreadsheet for frequent questions |
| Version 1.9 | 30/09/2024 | 1. Added support for RTK input NMEA protocol content.<br>2. Added recommended pad size |