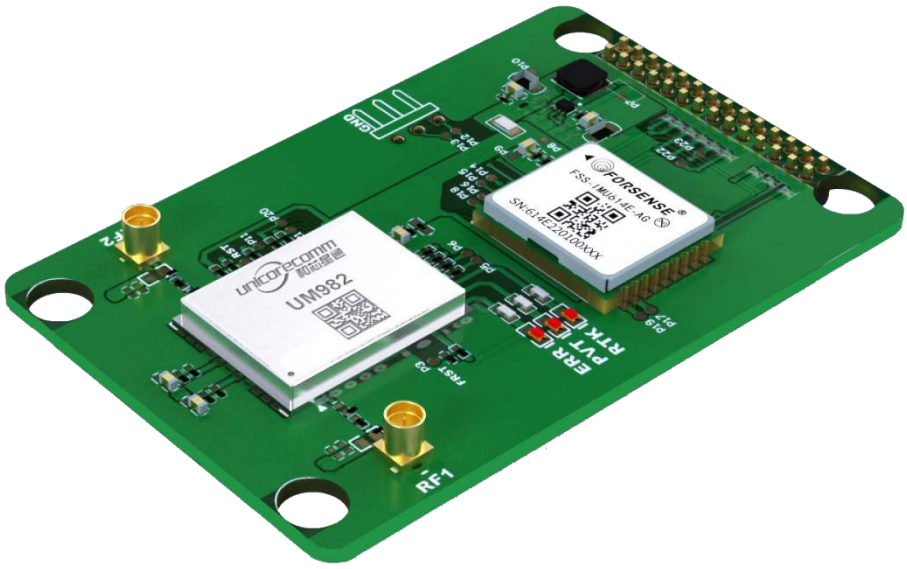
FS982-AG
High Precision Integrated Navigation Board
User Manuel



Forsense (Shanghai) Technology Co., Ltd

Read this user manual carefully before using the product.

Content

1. Product Introduction

1.1 Technical Specifications

Attitude Accuracy	Roll/Pitch: <0.2° rms Heading: <0.3° rms	
Update Rate	100Hz	
Gyroscope Measurement Range	±500°/s	
Gyroscope Bias Instability	4deg/h @1σ	
Accelerometer Measurement Range	±6g	
Accelerometer Bias Instability	0.04mg @1σ	
Estimated Position Accuracy	<2% @1σ （in-vehicle scenario, loss of GPS signal for 30s, no combination of wheel speedometer）	
RTK Indicators	Positioning accuracy (RMS)	Single point positioning: 1.5m RTK: 2cm+1ppm
	Directional accuracy (RMS)	0.2 °/1m baseline
	Velocity accuracy (RMS)	0.05m/s
	PPS accuracy (RMS)	20 ns
	Update rate	20Hz
	RTK initialization time	<5s

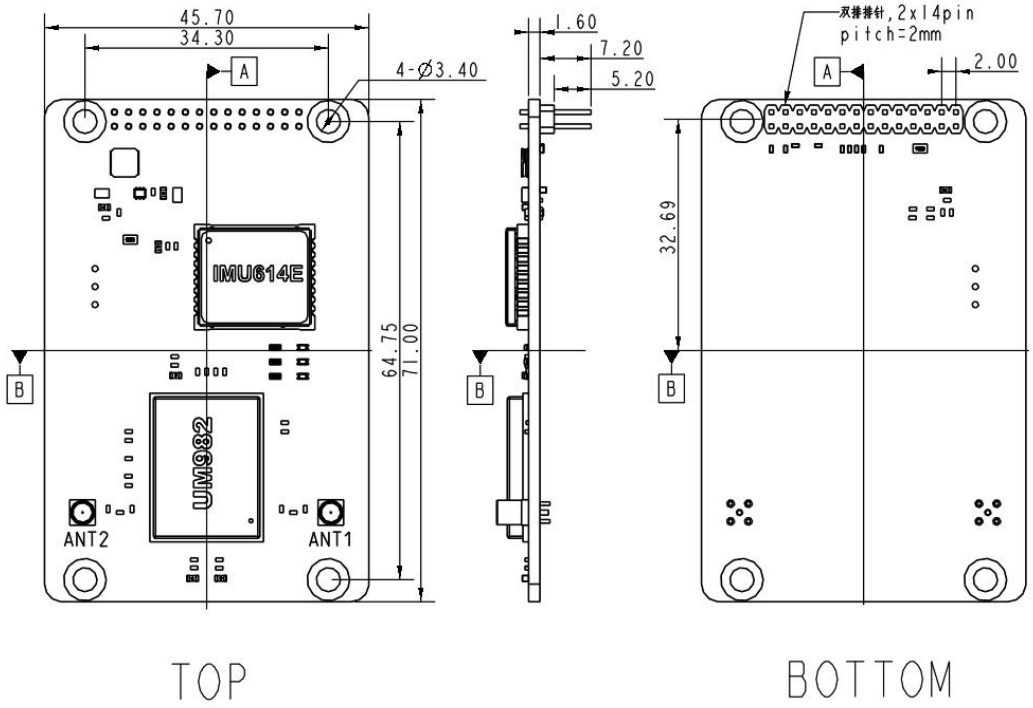
1.2 **Board Overview**

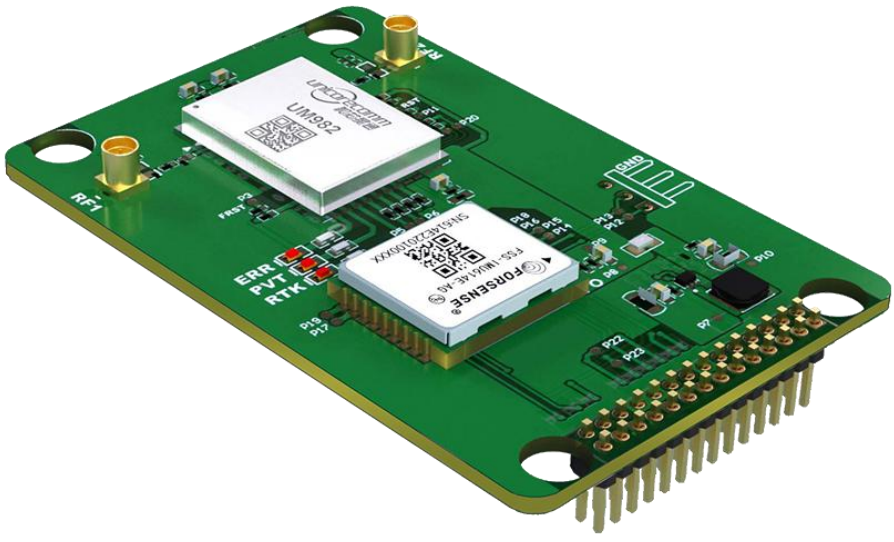
In meet the demand for precise and continuous positioning in lightly shaded scenarios, such as orchards, Forsense has designed the FS982-AG, an integrated navigation board specifically tailored for agricultural applications. This product combines the agricultural single-antenna module IMU614E-AG with Unicore's full-system full-frequency high-precision positioning and directional module UM982. Equipped with our multi-model intelligent position fusion algorithms optimized for in-vehicle scenarios, FS982-AG provides customers with high-precision attitude and heading information in complex scenarios and enables them to achieve continuous positioning and velocity measurement even in scenarios with signal interference or shielding. The product contributes to the reliable full-scene operation and enhanced end-user experience of agricultural autopilot equipment.

2. Hardware Structure

2.1 Mechanical Dimensions

Fig. 1 Mechanical Dimensions (unit: mm)

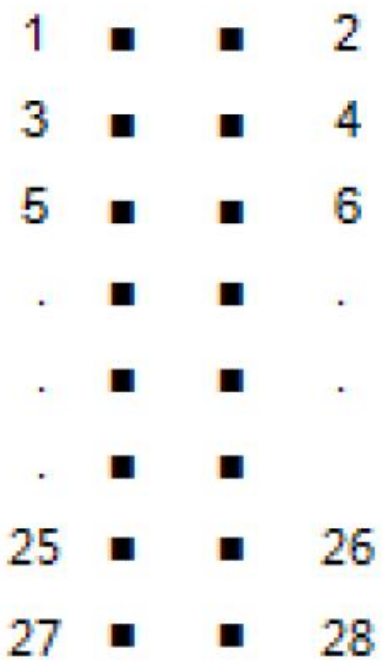




2.2 Connector & Pin Definition

In addition to the MMCX interfaces of ANT1 and ANT2, FS982-AG provides the following double row pin 28pin Cvilux. PIN pitch: 2.0mm; PIN length: 3.9mm; Base thickness: 2.0mm.

Fig. 2 Connector Pin Diagram



2.3 Pin Function

Table 1 Pin Function

Pin No.	Symbol	Input/Output	Description	Remark
1	RSV	-	Reserved	-
2	RSV	-	Reserved	-
3	RSV	-	Reserved	-
4	RSV	-	Reserved	-
5	RSV	-	Reserved	-
6	VCC	Power	Power Input	5VDC
7	RSV	-	Reserved	-
8	RXD2_IMU	I	IMU Serial Port 2 - Receiver (for firmware upgrading)	LVTTL voltage level
9	RSV	-	Reserved	-
10	RSV	-	Reserved	-
11	RSV	-	Reserved	-

12	RSV	-	Reserved	-
13	TXD2_IMU	O	IMU Serial Port 2 - Transmitter (for firmware upgrading)	LVTTL voltage level
14	GND	Power	Digital & Power Ground	
15	TXD1_IMU	O	IMU Serial Port 1 - Transmitter (for clients' use)	LVTTL voltage level
16	RXD1_IMU	I	IMU Serial Port 1 - Receiver (for clients' use)	LVTTL voltage level
17	GND	Power	Digital & Power Ground	
18	TX2_982	O	982 Serial Port 2 - Transmitter	LVTTL voltage level
19	RX2_982	I	982 Serial Port 2 - Receiver	LVTTL voltage level
20	GND	Power	Digital & Power Ground	
21	RSV	-	Reserved	-
22	GND	Power	Digital & Power Ground	
23	PPS	O	PPS	LVTTL voltage level
24	RSV	-	Reserved	-
25	RSV	-	Reserved	-
26	CAN_TX	O		LVTTL voltage level
27	RSV	-	Reserved	-
28	CAN_RX	I		LVTTL voltage level

2.4 Electrical Characteristics

Table 2 Absolute Maximum Ratings

Parameter	Symbol	Min.	Max.	Unit
Power Supply Voltage (VCC)	Vcc	-0.3	5.5	V
Input Pin Voltage	Vin	-0.3	3.3	V
VCC Max Ripple	Vrpp	0	40	mV
Input Pin Voltage (All pins except the aforementioned ones)	Vin	-0.3	3.6	V
Primary Antenna RF Input Power	ANT1_IN input power		±15	dBm
Secondary Antenna RF Input Power	ANT2_IN input power		±15	dBm
Maximum Tolerable ESD Stress Level	VESD(HBM)		±2000V	V

2.5 Operating Conditions

Table 3 Operating Conditions

Parameter	Symbol	Min.	Typ.	Max.	Unit	Condition
Power Supply Voltage (VCC)		4.75	5	5.25	V	
Power-on Impulse Current	Iccp			10	A	Vcc=5V
Input Pin Low Voltage	Vin_low_1	-0.3		0.9	V	
Input Pin High Voltage	Vin_high_1	2.4		3.6	V	
Output Pin Low Voltage	Vout_low	0		0.45	V	Iout=4mA
Output Pin High Voltage	Vout_high	2.85		3.3	V	Iout=4mA
Optimal Input Gain	Gant	20		36	dB	
Power Consumption	P		1.15		W	

2.6 Physical Characteristics

Table 4 Physical Characteristics

Operating Temperature	-40℃~+85℃
Storage Temperature	-55℃~+95℃
Humidity	95% non-condensing

3. Hardware Integration Guide

3.1 Design Considerations

1. To enable the FS982-AG to function properly, the following signals need to be connected correctly:
2. The module VCC exhibits a smooth ramp-up when powered on, with an initial voltage level below 0.4V, ensuring that undershoot and ringing are kept within 5% of VCC.
3. Use VCC pin to provide a reliable power supply, and keep all GND pins of the board grounded.
4. The MMCX interfaces of ANT1 and ANT2 supply power to the antennas. When the module interfaces are tested with a multimeter in an unloaded state (i.e., without an antenna connected), the voltage ranges from DC 4.8V to 5.4V. When the module's RF interface is connected to an antenna and operates within the working current of 30mA to 100mA under normal temperature, the voltage is DC 4.6V ± 0.2V. It is crucial to achieve impedance matching of 50 ohms for both ANT1 and ANT2.
5. To ensure the output of IMU serial port 1, which will be applied by users to receive IMU data.
6. To ensure that the IMU serial port 2 is connected to a pad or connector, which will be applied by users to upgrade firmware.
7. To ensure the output of the 982 serial port 2, which will be applied by users to receive positioning data.
8. The board reset pin FRESET_N is used to restore the factory default settings, and RESETIN is a quick reset. Please ensure correct connection for a reliable reset of the board.
9. To achieve good performance, more attention should also be paid to the following items in the design:
10. Power supply: Good performance requires a stable and low ripple power supply. The ripple voltage peak value should ideally not exceed 50mVpp. It is recommended that a power supply chip with current output greater than 2A can be used to power the board. In addition to using the LDO to ensure pure power supply, the following factors need to be considered:
11. − Widen the power traces or adopt split copper planes for current transmission.
12. − Place the LDOs as close as possible to the board.
13. − Avoid routing power traces near high-power and high-inductance components, such as magnetic coils.
14. The UART interface is used to ensure that the signal and the baud rate of the master device will match those of the FS982-AG board.

15. The antenna line should be as short and smooth as possible, avoiding routing traces with sharp angles and ensuring impedance matching.
16. Avoid routing directly below FS982-AG.
17. Keep the board far away from the hot airflow as much as possible.

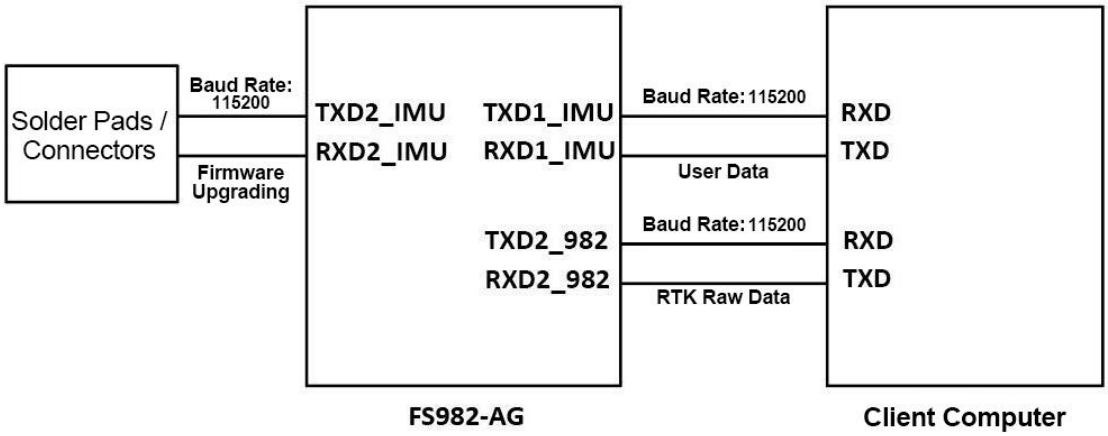
3.2 Pin Considerations

Table 5 Pin Considerations

Function	Pin	I/O	Description	Remark
Power Supply	VCC	Power	Power Supply	Stable, pure, and low-ripple power supply, ideally with a ripple voltage peak-to-peak value kept below 50mVpp.
	ANT1/ANT2	Power	Antenna Power Supply	The active antenna provides power with an appropriate voltage. When the module interfaces are tested with a multimeter in an unloaded state (i.e., without an antenna connected), the voltage ranges from DC 4.8V to 5.4V. When the module's RF interface is connected to an antenna and operates within the working current of 30mA to 100mA under normal temperature, the voltage is DC 4.6V ± 0.2V.
	GND	Power	Ground	All GND signals of the board are grounded, and a large copper area is ideally used for grounding.
UART	TXD1_IMU	O	IMU Serial Port 1 - Transmitter	The output of IMU serial port 1 needs to be connected to the user's computer.
	RXD1_IMU	I	IMU Serial Port 1 - Receiver	
	TXD2_IMU	O	IMU Serial Port 2 - Transmitter (for firmware upgrading)	The output of IMU serial port 2 needs to be connected to solder pads or connectors for firmware upgrading.
	RXD2_IMU	I	IMU Serial Port 2 - Receiver (for firmware upgrading)	
	TXD2_982	O	982 Serial Port 2 - Transmitter	The output of 982 Serial Port 2 needs to be connected to the user's computer.
	RXD2_982	I	982 Serial Port 2 - Receiver	

3.3 **Hardwiring**

Fig. 3 Hardwiring

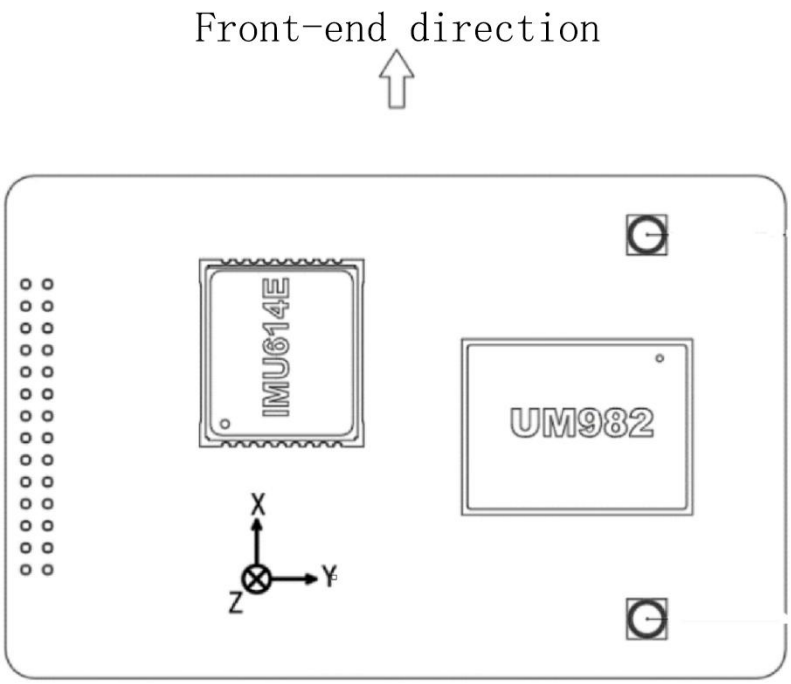


3.4 **Antenna**

The MMCX interfaces of ANT1 and ANT2 supply power to the antennas of the FS982-AG board. When the module interfaces are tested with a multimeter in an unloaded state (i.e., without an antenna connected), the voltage ranges from DC 4.8V to 5.4V. When the module's RF interface is connected to an antenna and operates within the working current of 30mA to 100mA under normal temperature, the voltage is DC 4.6V ± 0.2V. It is crucial to achieve impedance matching of 50 ohms when active antennas are utilized in the FS982-AG board.

4. Coordinate System Definition

Fig. 4 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 direction: Yaw angle range (0° ~ 360°);

Rotate around X-axis direction: Roll angle range (-180° ~ 180°);

Rotate around Y-axis: Pitch angle range (-90° ~ 90°).

The specific diagram is shown below:

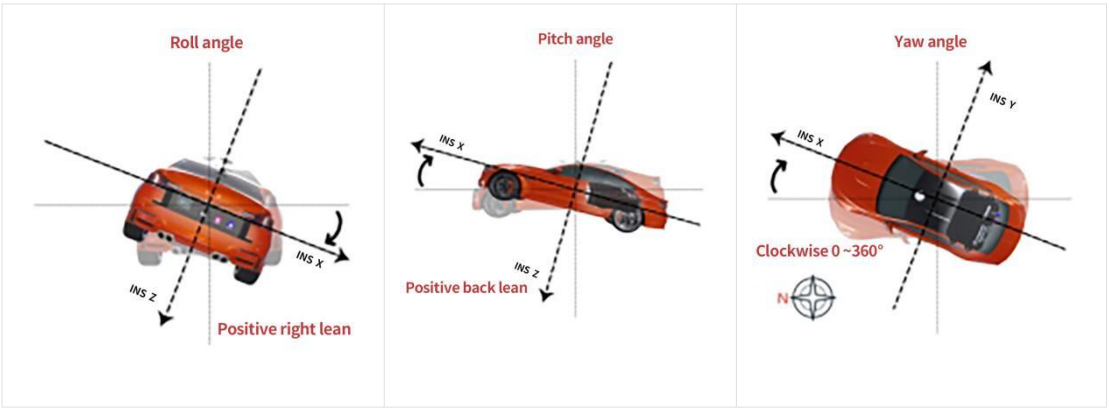


Fig. 5 Diagram of Roll, Pitch and Heading Angles

5. Connection & Setting

5.1 ESD Protection

Many components on the FS982-AG board are susceptible to electrostatic damage, thereby impairing the IC circuit and other components. Please ensure that the following electrostatic protection measures are taken before opening the anti-static blister box.

Electrostatic discharge (ESD) may damage components. Please handle the board on an anti-static workbench while wearing an anti-static wrist strap and using a conductive foam padded board. If no anti-static workbench is available, please wear an anti-static wrist strap and connect it to the metal part of the chassis for static protection.

When inserting or removing the board, do not directly touch the components on it. After removing the board, please carefully check the components for any obvious sign of loosening or damage.

5.2 Power On

The FS982-AG operates at a supply voltage of 5V. Once powered on, the receiver initiates its startup process and is capable of quickly establishing communication.

5.3 Test PC software

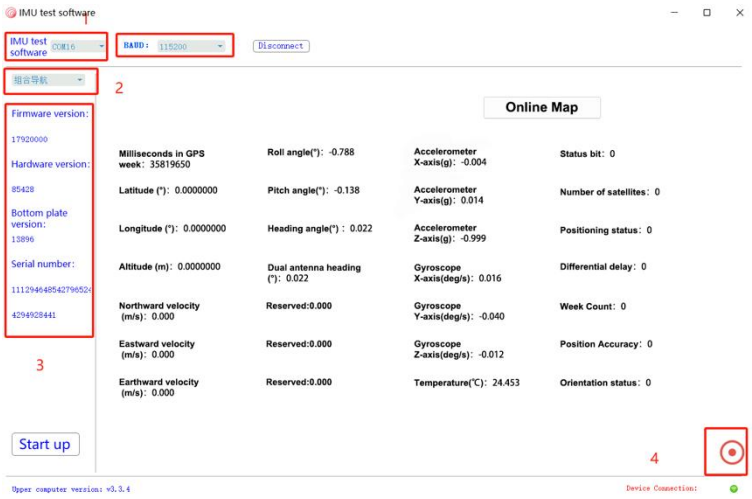
Download address: https://data.forsense-imu.com/ProductFile/link/Upper%20computer%20software/FS982-AG%20Map%20version/download.php

Select serial port number and baud rate at 1. Connect FS982 with USB to RS232 serial cable and ensure FS982 properly powered to display the serial port number;

Select INS data stream at 2;

Displayed at 3 are the name of connected hardware, the corresponding firmware version number, and the product serial number;

Click 4 to record and save current data.



6. Board LED Indicator

The FS982-AG board is equipped with LED indicators to display its basic operating status:

Table 6 Board LED Indicator

NO.	Indicator	Status	Description	Remark
1	Red	Always On	The board system fails the self check.	
		Always Off	The board system passes the self check.	
2	Green	Always On	The positioning information is accessible	
		Always Off	The positioning information is inaccessible	
3	Blue	Always On	RTK fixed solution	
		Always Off	Other RTK positioning statuses or a non-positioning state	

7. Integrated Navigation Output Protocol

7.1 Binary Protocol

Notes:

- CRC check starts from the frame header covering all bytes of the frame except the CRC check 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 where the low bytes are sent first.
- See section 7.2 for configuration methods.
- The binary protocol cannot be output at the same time as the nmea data stream, and before switching the data stream, you need to stop the output of current data stream by following the commands in 7.2.2.

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 angle (°) (cannot be used as a reference) Dual antenna: RTK dual-antenna heading angle (°) Front wheel gyro G200: Z-axis angular velocity (°/s)	Float	48
Reserved (displayed as the front wheel angle when 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:Unpositioned 1:Single point positioning 2:Pseudo-range differential positioning 4:Fixed solution 5:Floating solution	Uint8	84
Number of satellites	Uint8	85
Differential delay	Uint8	86
Dual antenna positioning status 50 indicates positioned	Uint8	87

Others indicate unpositioned		
Position accuracy factor (cm) Valid after the INS initialization	Uint16	88
Status bits: Bit0: 1 means that RTK data is valid, and 0 means invalid Bit1: 1 means that the PPS signal is valid, and 0 means invalid Bit2: 1 means that the INS is initialized, and 0 means uninitialized (The INS configured with single antenna mode can be initialized if vehicles travel at the speed of over 0.5m/s after starting up, while the INS configured with dual antenna mode can be initialized without the need to start up or enter positioning state.) Bit3: 1 means that the front wheel angle is valid, and 0 means invalid Bit4: 1 means that the INS has converged, and 0 means not converged Bit5: 1 means that the front wheel gyro data is valid, and 0 means invalid (ignore if no external gyro) Bit6: 1 means that the steering wheel motor data is valid, and 0 means invalid Bit7 and Bit8: 01 means that the vehicle moves forward 10 means that the vehicle moves backward 00 means invalid	Uint16	90
Reserved 1	Uint32	92
Reserved 2	Uint32	96
CRC Check	Uint32	100

7.2 NMEA Protocol

- Support the output of combined data in nmea format.
- The NMEA protocol and the binary data stream cannot be output at the same time, and before switching the data stream, stop the output of current data stream by following the commands in 7.2.2.
- The following statements are supported. See section 7.3 for configuration methods.

GPGGA

GPRMC

GPHDT (Heading information)

GPVTG (Ground velocity)

GPZDA (UTC time and date)

GPATT (Customized message)

The GPATT formats are shown in the following table

Name	*Unit*	*Format*	*Example*	*Description*
Sentence Identifier		String	$GNATT	
Time		hhmmss.sss	170834.000	17:08:34 UTC
Status		Character	1	0: invalid 1: valid
Roll Angle	degree	3 decimal places	-4.891	range ±90, right side down defined as positive
Indicator for roll		character	R	Roll indicator
Pitch Angle	degree	3 decimal places	3.122	range ±90, head up defined as positive
Indicator for Pitch		character	P	Pitch indicator
Heading Angle	degree	3 decimal places	124.005	range 0~360, to true North, counter clockwise defined as positive
Roll Angle uncertainty	degree	3 decimal places	0.432	range 0~360
Pitch Angle uncertainty	degree	3 decimal places	0.811	range 0~360
Heading Angle uncertainty	degree	3 decimal places	1.202	range 0~360
Checksum		Hex	*68	Used by program to check for transmission errors

7.3 CAN Protocol

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		Heading angle		Track angle	

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

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)

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

Extended Frame ID	1	2	3	4	5	6	7	8
0X19FFCE9B	Seconds in GPS week				Estimated antenna mounting deviation angle		Speed direction	Reserved

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

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 uninitialized (only for the gyro of single antenna vehicle).

7.4 RTK Positioning Status Table

Binary	ASCII	Description
0	NONE	No solution
1	FIXEDPOS	The position is specified by the FIX POSITION command
2	FIXEDHEIGHT	Unsupported
8	DOPPLER_VELOCITY	Velocity is derived instantly from Doppler information
16	SINGLE	Single point positioning
17	PSRDIFF	Pseudo-range differential positioning
18	WAAS	SBAS positioning
32	L1_FLOAT	L1 floating solution
33	IONOFREE_FLOAT	Floating solution for mitigating ionospheric effects
34	NARROW_FLOAT	Narrow lane floating solution
48	L1_INT	L1 fixed solution
49	WIDE_INT	Wide lane fixed solution
50	NARROW_INT	Narrow lane fixed solution

8. Parameter Configuration

8.1 Use Serial Ports to Configure Parameters

8.1.1 Configure Lever Arms

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 three dimensional vector (X, Y, Z) representing the position of the RTK main antenna phase center relative to the IMU phase center. (unit: m)

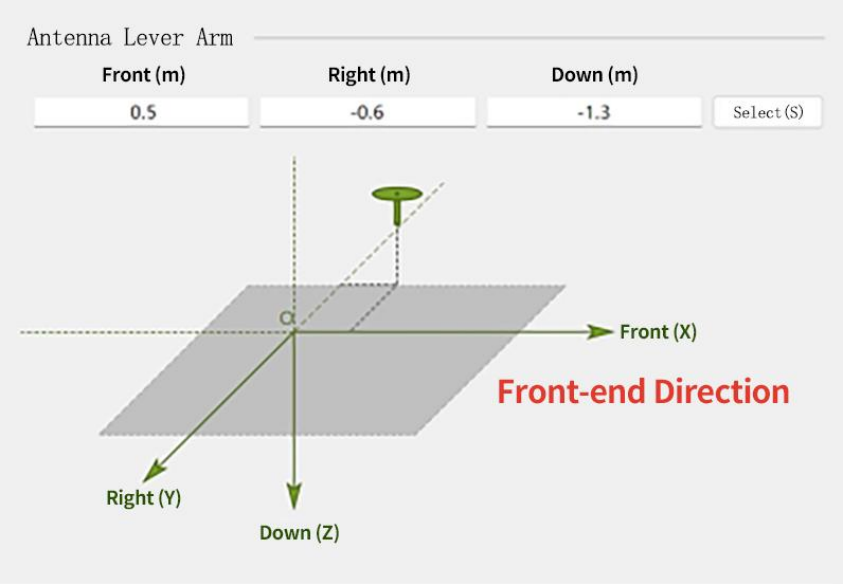
In the Front-Right-Down coordinate system,

the value is positive if the RTK main antenna is in front of the IMU, and it is negative otherwise;
the value is positive if the RTK main antenna is on the right side of the IMU, and it is negative otherwise;
the value is **negative** if the RTK main antenna is above the IMU, and it is positive otherwise.

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



8.1.2 Configure the Output of Binary 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

8.1.3 Configure the Output of Data Stream in NMEA Format

If the NMEA statement is configured to output, the 7.2 integrated navigation data stream will not be output.

If you want to switch to 7.2 integrated navigation data stream output, you need to stop the current data stream output by entering 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)

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

8.1.4 Configure the Current Data Stream to Stop Output

Command: AT+SETNO\r\n

Response: OK\r\n

8.1.5 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 INS default output is projection points of the antenna phase center. To output results from other positions, you need to configure the corresponding lever arm vectors. The configuration method is the same as 7.1 lever arm configuration.

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

Note: After saving the configuration commands, it is crucial to power off and restart. The distance between the two antennas should be more than 50cm.

8.1.7 Calibrate Dual Antenna

The dual antenna calibration process is as follows:

1. Connect serial ports and enter AT+SETNO\r\n to stop all 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 the serial port sends the statement RTK_BIASE_CORRECT_DONE;

4. Send the command AT+CONFIG\r\n to view the current configuration, and send the statement RTK_BIAS_FLAG_AND_VALUE=99,XX (°) to confirm the calibration angle. The installation angle is normally less than 1°; otherwise, reinstallation is a must;

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

8.1.8 Configure 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

8.1.9 Configure Baud Rate

Only support the configured baud rate 115200 and 230400, and the default is 115200.

If the IMU serial baud rate is configured to 230400, the configuration command is:

Command: AT+BAUD=230400\r\n

Response: BAUD=230400\r\n

Note: It is essential to power off and restart after configuring and saving commands.

8.1.10 Configure 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.

8.1.11 Query All Configuration Information

If you query all configured information, the configuration command is:

AT+CONFIG\r\n

8.1.12 Query Version Number

AT+VERSION\r\n

8.1.13 Configure 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

8.1.14 Turn on the Slope Mode

For scenarios 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 the slope mode will work after reboot.

8.1.15 Save Parameters

The above parameters involve lever arm, data stream, output frequency, etc. After the configuration, you need to enter the SAVE command and restart the software.

Command: AT+SAVE\r\n

Response: OK\r\n

8.2 Use CAN Interface to Configure Parameters

8.2.1 Query Version Number

Example:

ID = 0x19FFF326

Data = 0x9A 0x07 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF

Response (version number V1.21.2.2)

ID = 0x19FFF29A

Data = 0x26 0xC7 0x01 0x15 0x02 0x02 0xFF 0xFF

8.2.2 Configure Wheelbase and Lever Arm

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)

8.2.3 Configure Dual Antenna Mounting Angle

ID = 0X0DFFC326

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

8.2.4 Query Wheelbase, Lever Arm 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

8.2.5 Configure Filter Cutoff Frequency

ID = 0X0DFFC626

Data = cutoff frequency (uint8, see definition below) + 0xFF*7

Response

ID = 0x19FFF69A

Data = cutoff frequency (uint8, see definition below) + 0xFF*7

Common cutoff frequency definitions:

0x44：10Hz

0x66：20Hz

0xBB：47Hz

8.2.6 Query Cutoff Frequency

ID = 0X0DFFC726

Data =0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF

Response:

ID =0x19FFF79A

Data = cutoff frequency (uint8, see definition below) + 0xFF*7

8.2.7 Configure CAN Interface to Output Update Rate and Baud Rate

Baud rates: 1M, 500K, 250K (0x01, 0x02, 0x03 respectively)

Update rates: 20Hz, 50Hz, 100Hz (0x14, 0x32, 0x64 respectively).

After the 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

8.2.8 Query Update Rate and Baud Rate

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

8.2.9 Save Parameter Instructions

ID = 0X0DFFCF26

Data =0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF

Response

ID = 0X19FFCF9A

8.3 Dual Antenna Calibration

The calibration process is as follows:

1. Connect serial ports, enter AT+SETNO to stop all data output, and send the command AT+RTK_BIAS_EST=1 to start the calibration process;

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

3. Send the command AT+CONFIG to view the current configuration, and send the statement RTK_BIAS_FLAG_AND_VALUE=99,XX (°) to confirm the calibration angle. The installation angle is normally less than 1°; otherwise, reinstallation is a must;

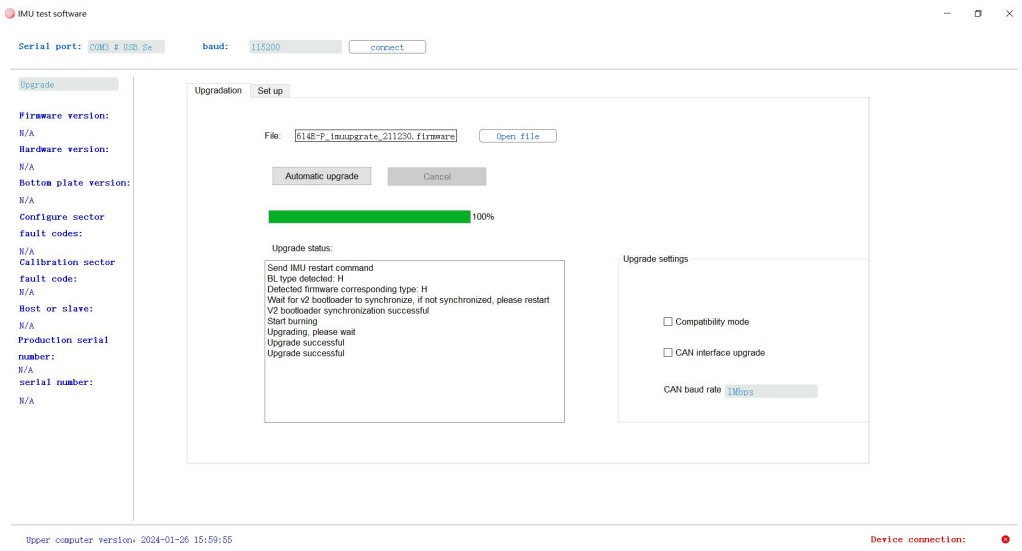
4. Enter AT+SAVE to save the result and re-power on.

9. Firmware Upgrading

9.1 Firmware Upgrading via the PC Software

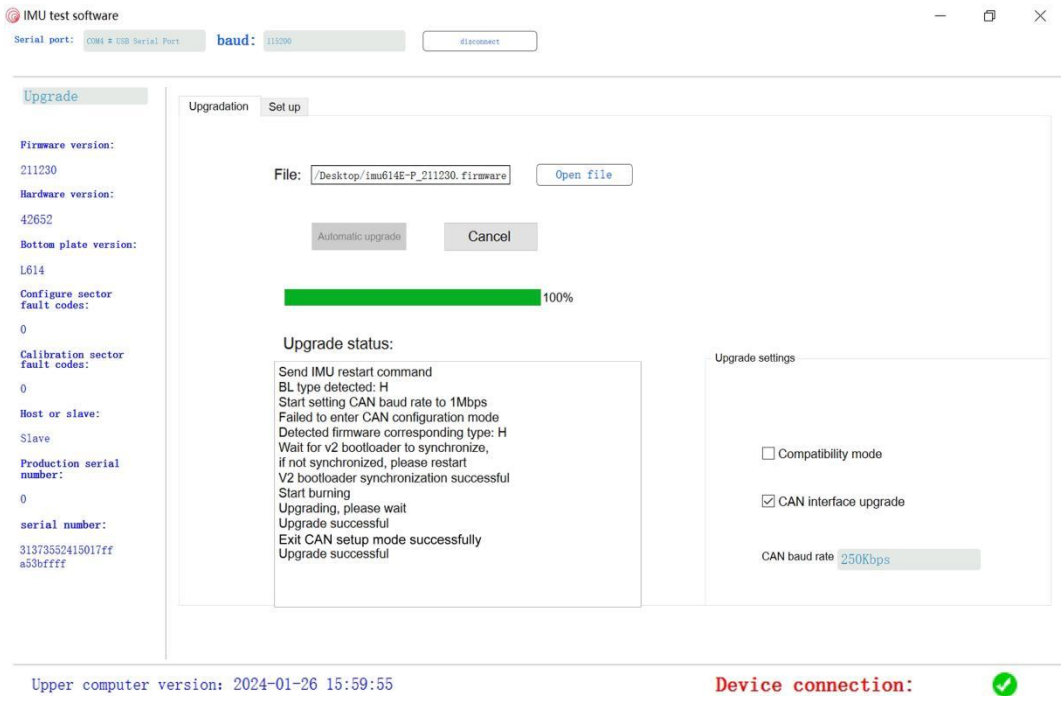
9.1.1 RS232

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



9.1.2 Firmware Upgrading via the CAN Interface

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



9.2 **Serial OTA**

The upgrade process is divided into the following steps:

Step 1: Send 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. 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 string **HC32MCU_FORSENSE**

After entering the bootloader, the IMU module will actively send the string HC32_UPLOADER. During this period, the PC software needs to send the string HC32MCU_FORSENSE. If the module receives this string, it will not jump to APP, but stay in a pending upgrade state in the bootloader and 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 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 this command, IMU replies with Send CMD ACK (0x64,0x10,0) immediately. Once the PC software determines that it has received this response, it indicates that the synchronization process 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 all the content in the APP area and send the final execution result 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 the 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 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 if it fails to write data to flash
send_CMD_ACK(0x64,0x11,0);

      //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 The imu module will send the failure command if it fails to write data to flash
send_CMD_ACK(0x64,0x11,0);



     //Send the 3rd packet:
      3.1 After the PC software receives the response data from imu, iit 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), it will acquire the CRC check digits calculated by the IMU module.

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);
     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);
```

Once the firmware upgrading is completed, you can determine whether the upgrading process is successful by reading the version number after power cycling.

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 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,
```

10. Usage Examples

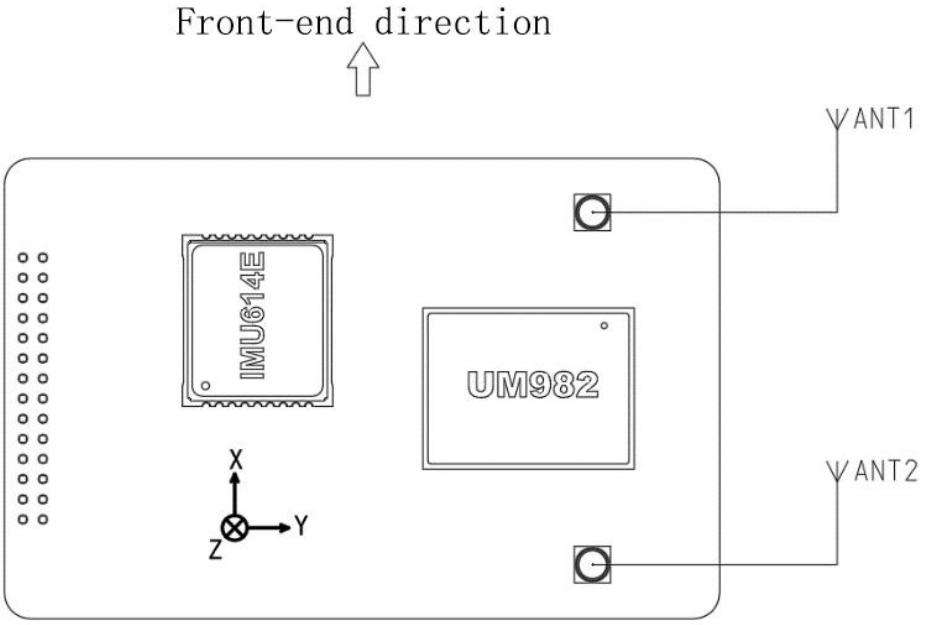
10.1 Device Installation

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.

3. ANT1 is the primary antenna (positioning antenna) RF connector, and ANT2 is the secondary antenna (directional antenna) RF connector. The dual antenna orientation result of RTK board is the angle between the ray extending from the primary antenna to the secondary antenna and the geographic (true) north direction.

Fig. 6 Antenna Connection Diagram



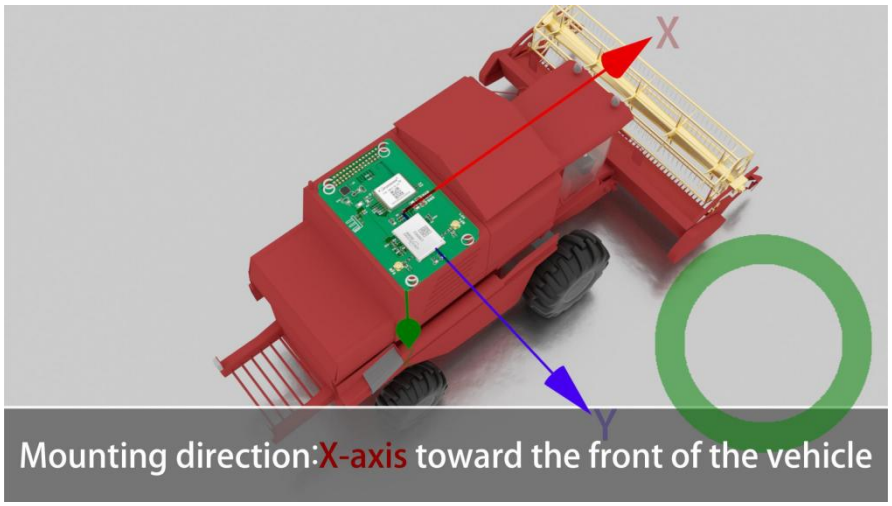
Since there is no directional marker on the IMU module, you can determine the installation orientation by referencing the triangular marker on the module, as shown in the following figure.

Fig. 7 Coordinate System Diagram

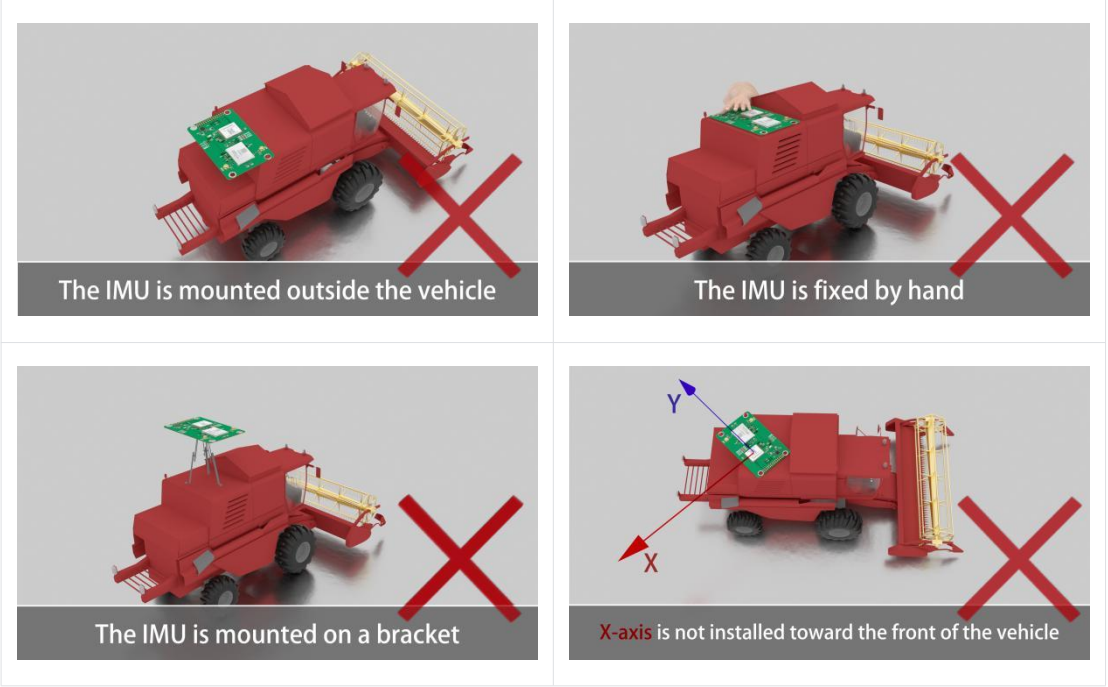
The correct installation is shown as follows

X-axis toward the front of the vehicle

Fig. 8 Correct Installation Diagram



The following mounting methods are incorrect:



Module Installation Precaution:

No glue or other liquids around the IMU module to prevent them from flowing into the interior of the module through gaps, thus impairing the IMU's performance.

Fig. 22 Wrong Installation Diagram

RTK Differential Data Import

Users should receive GPGGA messages from the RTK board via 18 and 19 pins (corresponding to the serial port 2 of the UM982 board), log in the network CORS of Qianxun SI or Sixents Technology and import the differential data, thereby making the RTK

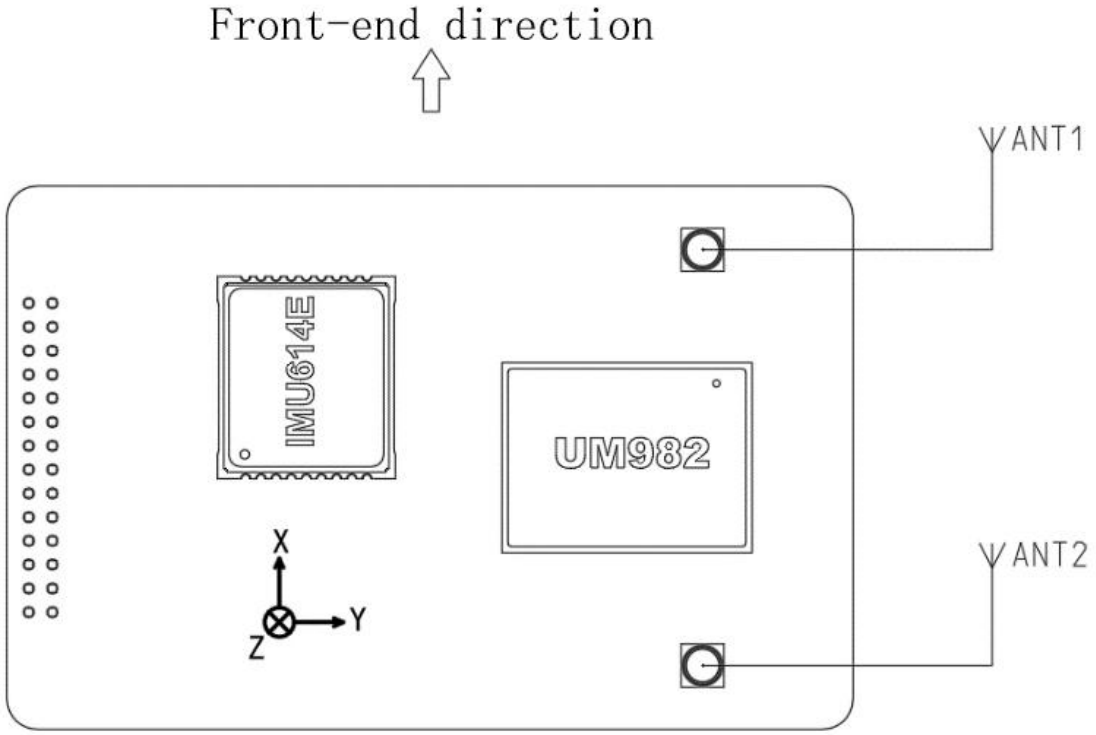
board enter a fixed solution state.

10.2 Configure Lever Arm Parameters

The lever arm vector is the three-dimensional vector (X,Y,Z) of the RTK main antenna phase center relative to the IMU installation position. (unit: m)

- The value is positive if the RTK main antenna is aligned with the positive X-axis of the IMU, and it is negative otherwise;
- The value is positive if the RTK main antenna is aligned with the positive Y-axis of the IMU, and it is negative otherwise;
- The value is positive if the RTK main antenna is below the IMU, and it is negative otherwise;

Fig. 15 FS982-AG Antenna Three-dimensional Vector (X,Y,Z)

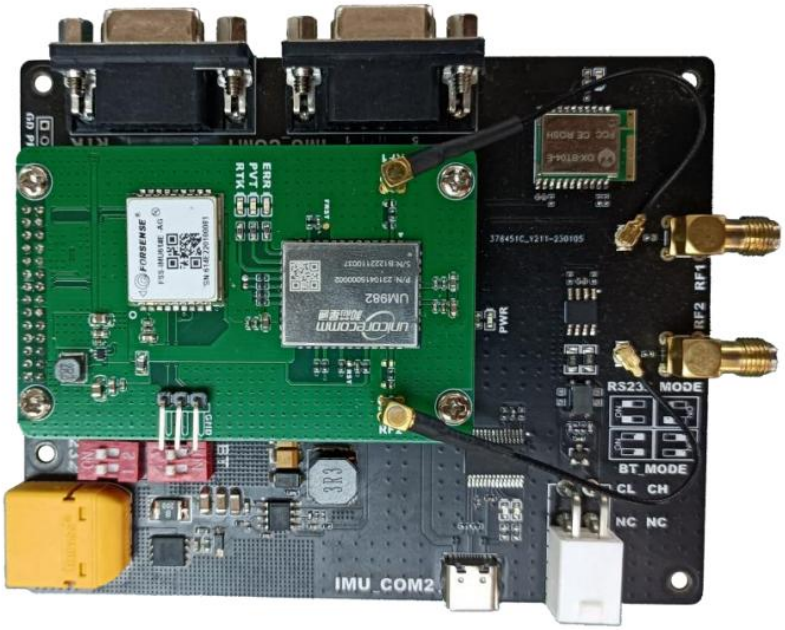


10.3 Save Parameters

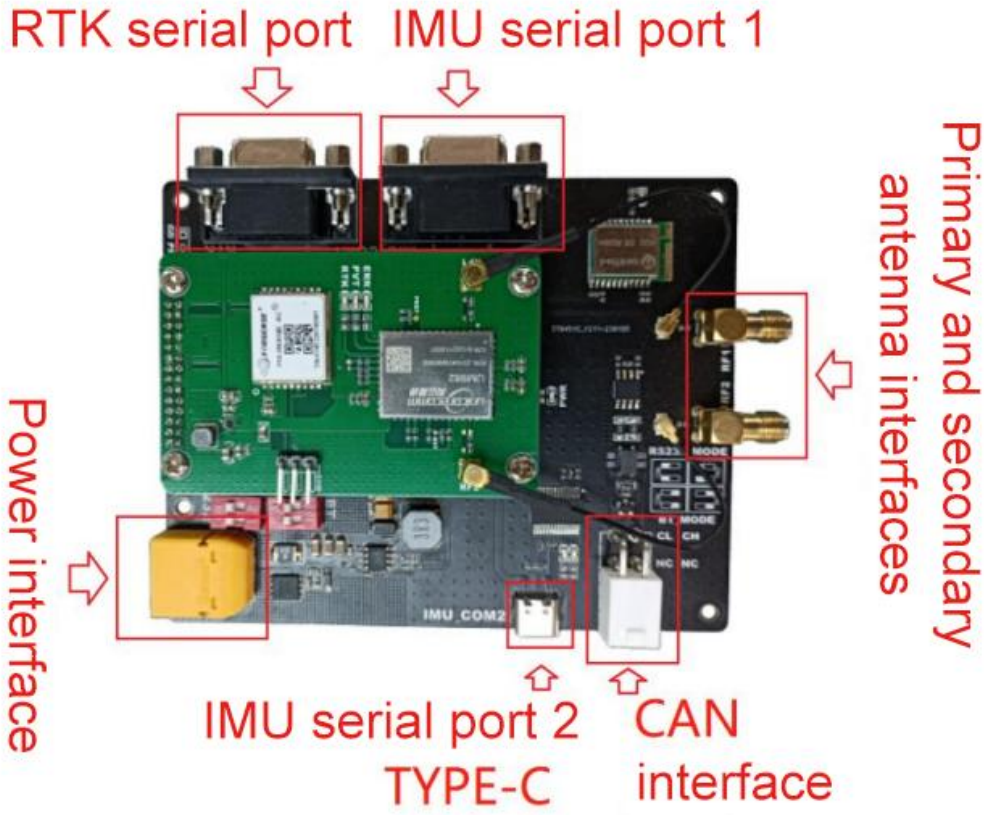
After all commands are configured, send the command AT+SAVE\r\n to save parameters.

11. Base Plate

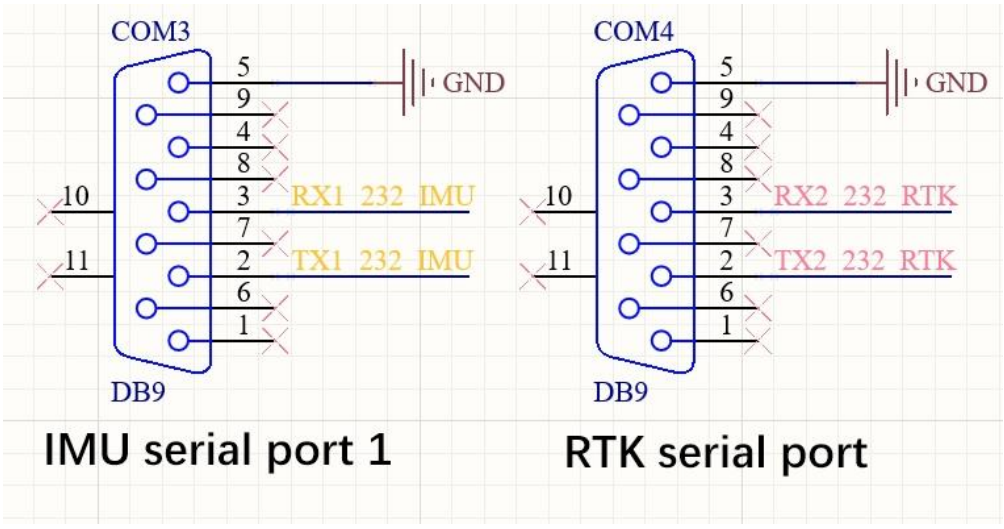
11.1 Base Plate Diagram

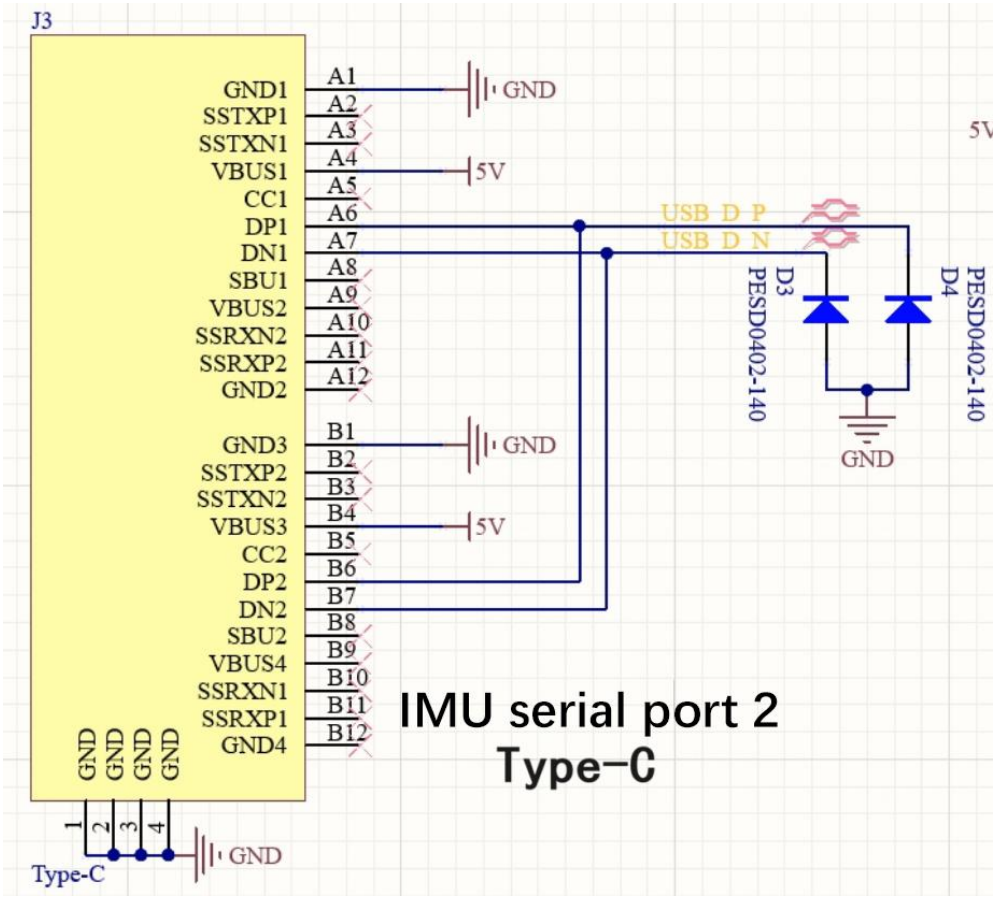


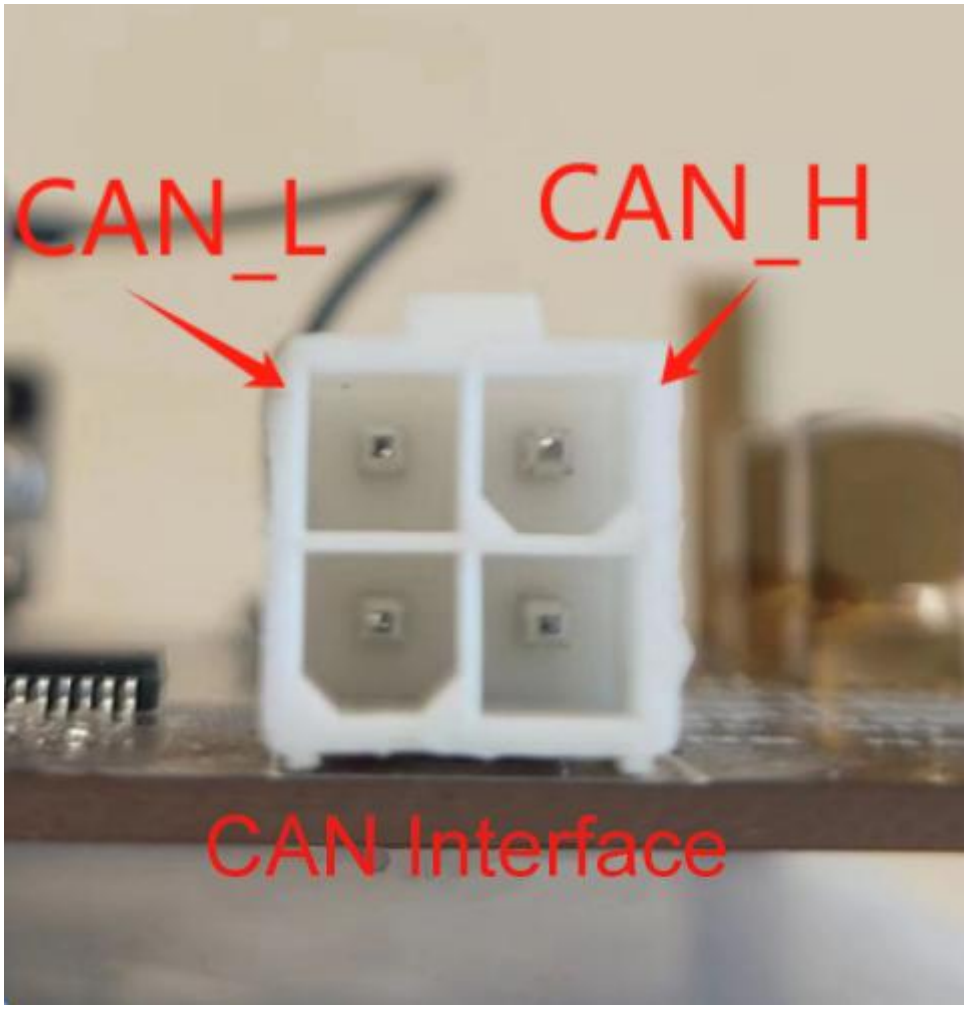
11.2 Interface Diagram



11.3 Interface Definition







11.4 Interface Description

IMU serial ports 1 and 2 cannot be configured independently. After sending the configuration command, both serial ports will work at the same time and output the same data stream.

11.4.1 RTK Serial Port

- Interface type: Serial port (DB9);
- This interface is connected to RTK serial port 2 for the purpose of receiving RTK data, importing differential data and configuring RTK parameters.

11.4.2 IMU Serial Port 1

- Interface type: Serial port (DB9);
- This interface is connected to IMU serial port 1 for the purpose of connecting to the PC software, reading IMU data, and changing IMU parameters.

11.4.3 Primary and Secondary Antenna Interfaces

- Interface type: SMA RF interface
- This interface is connected to the RTK primary antenna (RF1) and secondary antenna (RF2).

11.4.4 CAN Interface

- Interface type: CAN;
- This interface is connected to the CAN port of the IMU for the purpose of receiving data from the front wheel gyro G200.

11.4.5 IMU Serial Port 2

- Interface type: Serial port (type-C);
- This interface is connected to IMU serial port 2 for the purpose of connecting to the PC software, reading IMU data, changing IMU parameters, and supplying power (5V) to base plates.

11.4.6 Power Interface

- Interface type: XT60;
- This interface is used to power the FS982-AG base plate;
- Supply voltage: 12V ± 0.3V.

12. ROS Driver

ROS driver download address: https://www.forsense.cn/download/

12.1 Install ROS Serial

Install the ROS serial package. This example program relies on the serial package provided by ROS to achieve 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

12.2 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.

12.3 Connect 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:~$
```

Configure and enable USB to serial port permissions:

sudo chmod 777 /dev/ttyUSB0

12.4 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

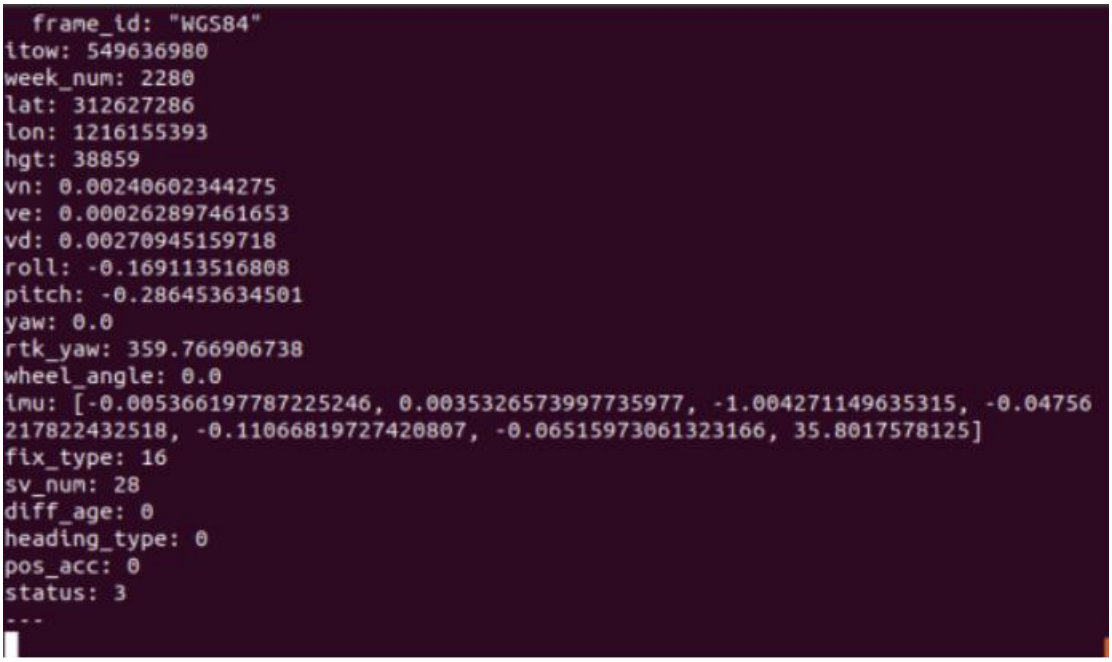
source devel/setup.bash

rostopic list

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

Enter a command to check 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, -0.04756
217822432518, -0.11066819727420807, -0.06515973061323166, 35.8017578125]
fix_type: 16
sv_num: 28
diff_age: 0
heading_type: 0
pos_acc: 0
status: 3
---

13. Used with G200

1. G200 installation 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 vehicles. 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 on.

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

14. Accessories

Standard accessories:

	
FS982-AG Base Plate	RS232 Serial Cable
	
Type-C Cable	USB to CAN Module
	
Primary Antenna (positioning antenna)	Secondary Antenna (directional antenna)
	
Primary Antenna Connector	Secondary Antenna Connector

Optional accessories:

G200

15. Revision History

Version	Date	Status/Notes
Version 1.0	2023.07.08	First release
Version 1.1	2023.10.07	Updated the definition of coordinate system
Version 1.2	2023.12.14	Added accessories
Version 1.3	2024.01.18	1. Added commands for configuring NMEA statement data flow 2. Added commands for configuring the rotation angle while installing the board 3. Added a description of base plates
Version 1.4	2024.01.18	1. Added a description of ROS driver 2. Added instructions on how to use the PC software
Version 1.5	2024.01.25	Added the definition of base plate interfaces
Version 1.6	2024.02.21	Added a description of PPS accuracy
Version 1.7	2024.02.22	Corrected the description of rotation angle configuration and response
Version 1.8	2024.03.22	Updated installation diagrams
Version 1.9	2024.03.26	Added slope mode and using methods of combining FS982-AG and G200
Version 2.0	2024.04.09	Added dual antenna calibration