NAV680-AG

Dual-Antenna High-Precision

Integrated Navigation System

Product Sheet

Forsense (Shanghai) Technology Co., Ltd

Please read this manual carefully before using this product.

CONTENT

1 Description

1.1 Product List

When opening the package, please confirm the following products:

Name	Quantity	Diagram
① NAV680-AG Integrated Navigation System	1	
② Satellite Antenna	2	
③ Antenna Cable Primary antenna: TNC male head to fakra female head (Type C)(blue) Secondary antenna:TNC male head to fakra female head (Type D)(Purple)	2	
④ Antenna Suckers	2	
⑤ Antenna column	2	
⑥ Harness (include 3×RS232, 2×CAN, 1×PPS, and 1×Power Connector(9-24v)).	1	

Fig.1 Product Material Drawing

If there's anything missing, please contact salesperson in time.

1.2. Product Description

Aiming for precision positioning and continuous positioning of low-speed automatic driving in farmland, orchards, scenic spots, outdoors and other lightly shaded scenarios, this satellite inertial combination product NAV680-AG has been developed. It is based on Forsense self-developed IMU module and combination algorithms, and integrates full-system, full-frequency point, high-precision positioning and directional module UM982. The product can support BDS B1I, B2I, B3I; GPS L1, L2, L5; GLONASS G1, G2; Galileo E1, E5a, E5b; QZSS L1, L2, L5 With the built-in Forsense multi-model intelligent position fusion algorithms for on-board scenarios, it can provide high-precision on-board attitude, heading, continuous localization, and speed measurement for autonomous driving customers such as agricultural machines, inspection vehicles, and scenic tour buses, etc. without accessing to the wheel odometer. no matter in open sky or lightly shaded areas, which can help a reliable operation of the whole scenario of low-speed autonomous driving, and enhance the end-user application experience of the autonomous driving equipment.

In order to meet the needs of agricultural machinery, mining trucks and other scenarios which are in need of the front wheel angle, NAV680-AG can be connected to our front wheel angle sensor G200, which can directly output the front wheel steering angle.

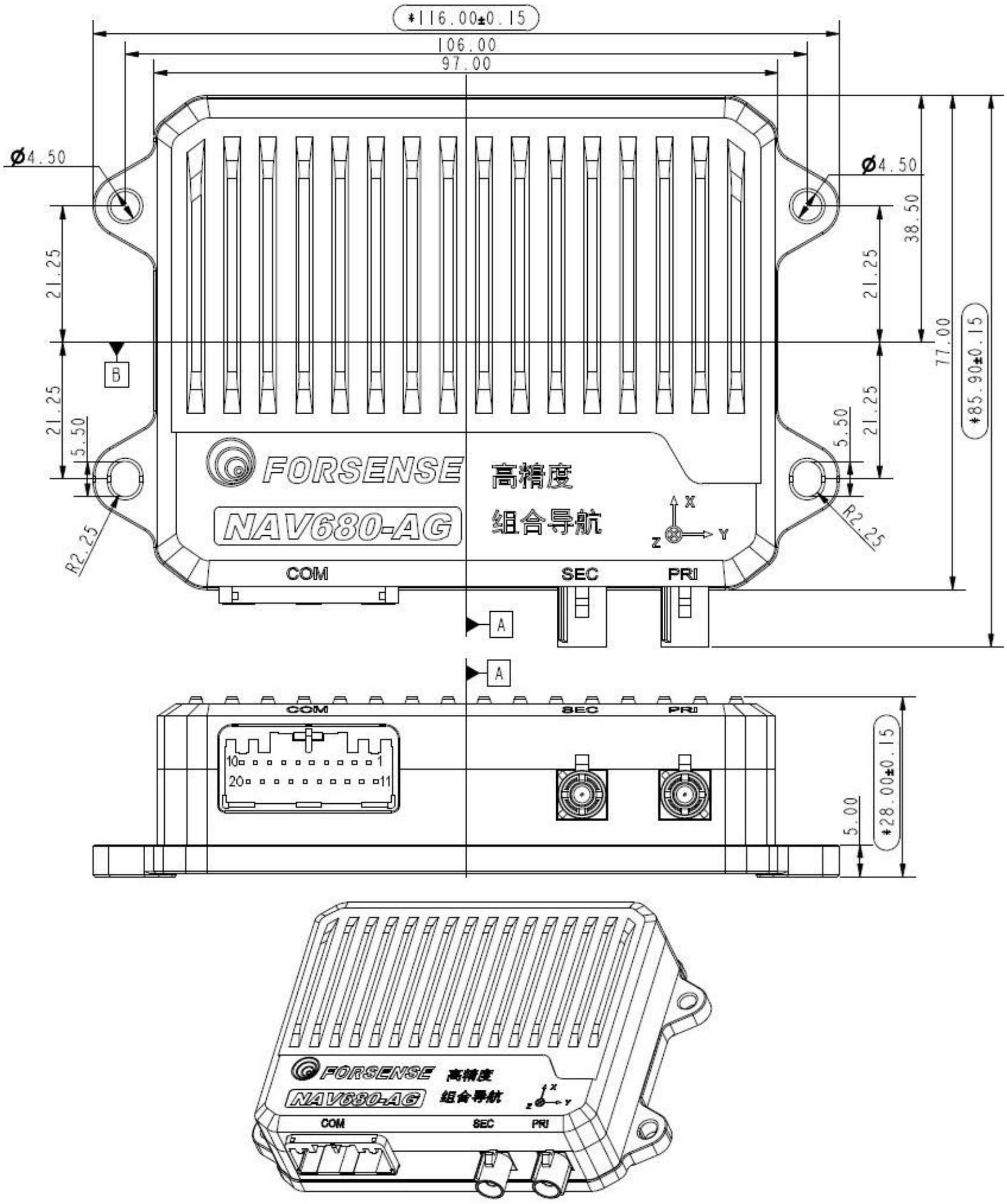
1.3 Technical Indicators

Table 1

Attitude Accuracy	Roll/Pitch: <0.2° rms Heading:<0.3° rms (Vehicle speed > 1km/h, on-board Ackermann steering mechanism)	
Update Rate	100hz	
Gyroscope Measurement Range	±500°/s	
Gyroscope Bias Instability	4deg/h @1σ	
Accelerometer Measurement Range	±6g	
Accelerometer Bias Instability	0.04mg @1σ	
Position Reckoning Accuracy	<2% @1σ (On-board scenario, Satellite signal loss@(30 s), without wheel odometer)	
RTK Indicators:	Positioning Accuracy (RMS)	Single Point: Horizontal 1.5m/ Elevation 2.5m RTK: Horizontal 1cm+1ppm Elevation 2cm+1ppm
	Dual Antenna Positioning Accuracy (RMS)	0.1°/1m Baseline
	Velocity Accuracy (RMS)	0.03m/s
	PPS Accuracy (RMS)	20ns
	Update Rate	20hz
	RTK Initialization Time	<5S (Typical value)
Interface	3×RS232 2×CAN 2×GNSS Antenna Interface 1×Power Connector 1×PPS Interface	

2 Hardware Components

2.1 Mechanical Dimensions



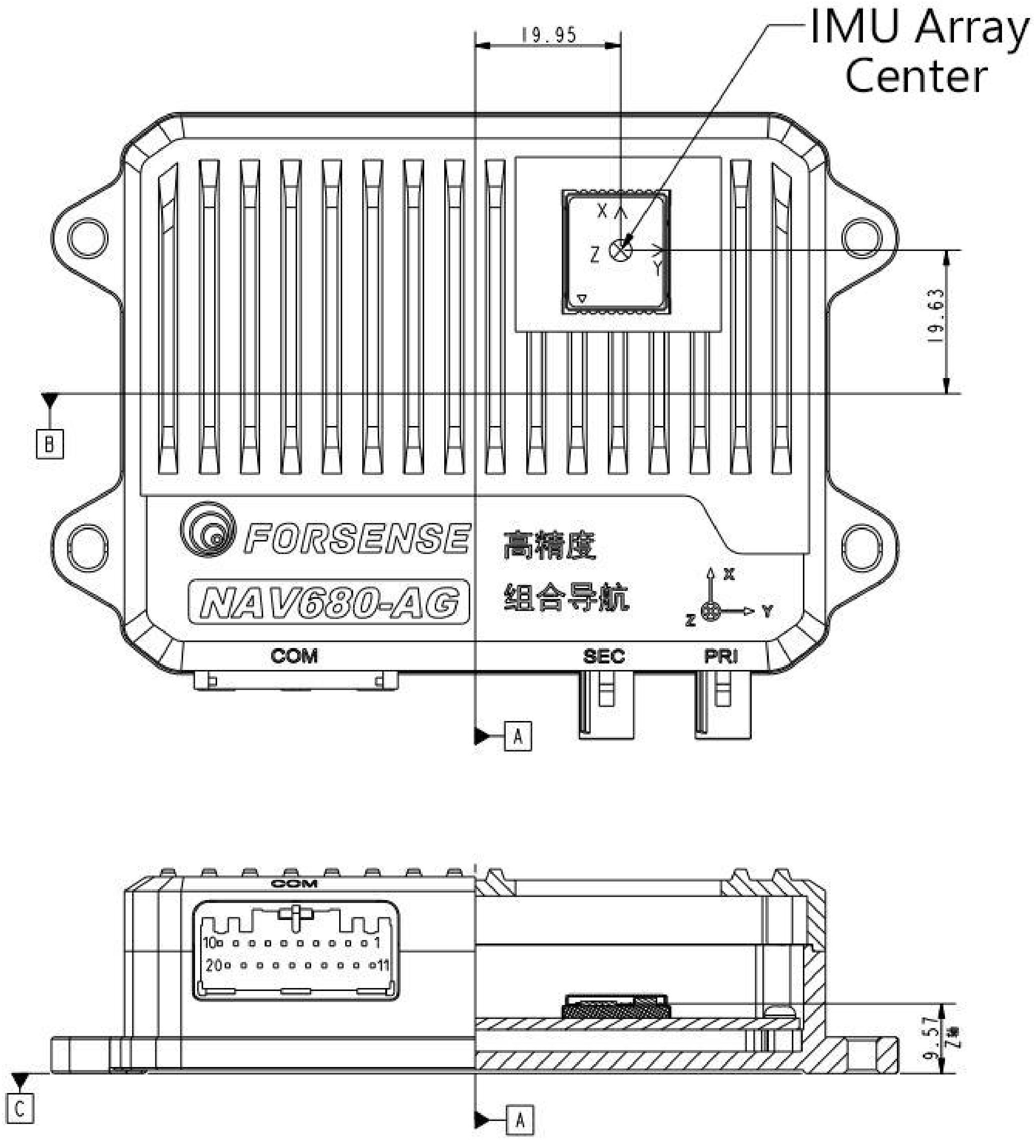
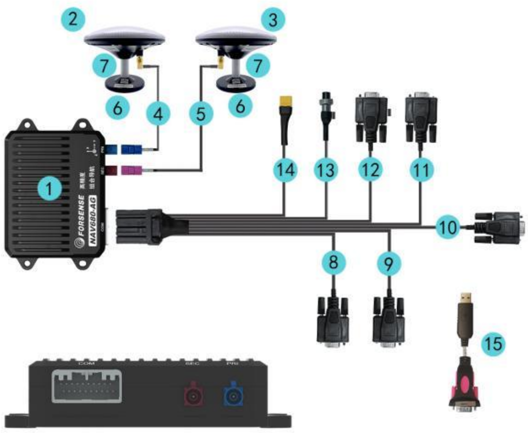


Fig. 2 Mechanical Dimensions (unit: mm)

2.2 Interface Instructions

NAV680-AG includes COM bus interface, which brings out the power cable and signal cable, primary antenna(PRI) and secondary antenna(SEC).



	Name	Quantity
1	FSS-NAV680-AG Integrated Navigation System	1
2	Primary antenna (Positioning Antenna)	1
3	Secondary antenna (Directional Antenna)	1
4	Primary antenna Cable	1
5	Primary antenna Cable	1
6	Antenna suction Cup	2
7	Antenna Pole	2
8	Wire Harness - Power Supply Interface	1
9	Wire Harness - RS232-A Interface	1
10	Wire Harness - RS232-B Interface	1
11	Wire Harness - CAN-A Interface	1
12	Wire Harness - CAN-B Interface	1
13	Wire Harness - RS232-RTK Interface	1
14	Wire Harness - IO Interface	1
	Accessory Name	**Quantity**
15	RS232 to USB Serial Cable	1

2.3 Definitions for Digital Interface

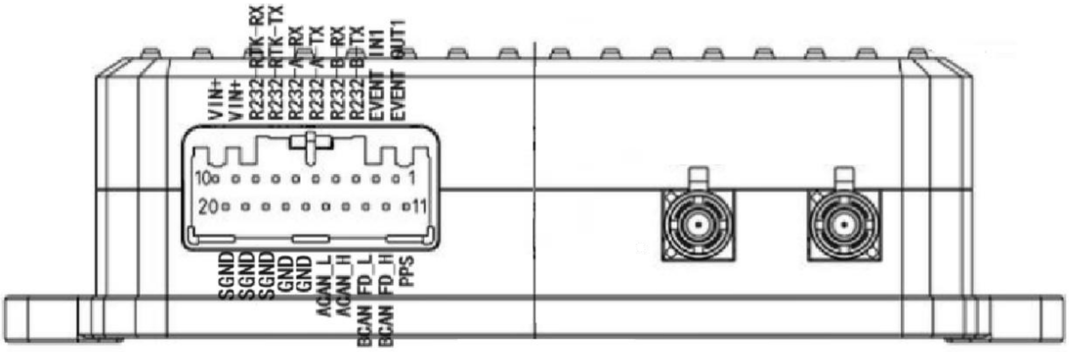
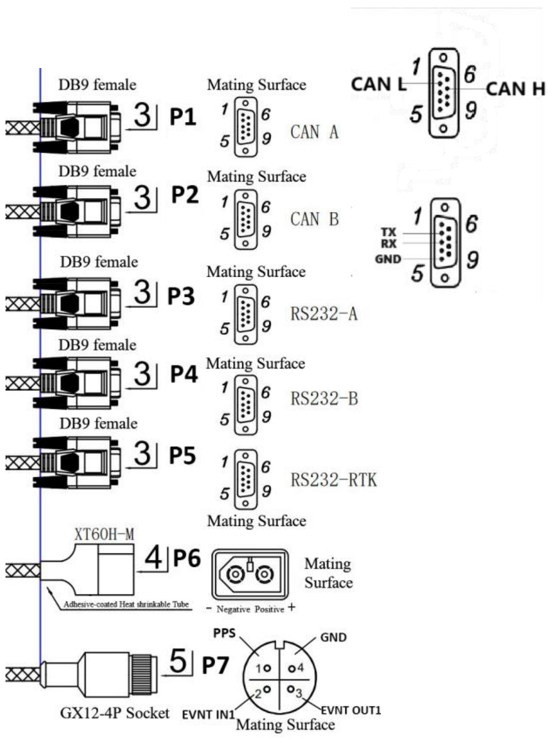


Fig. 3 Definitions for Digital Interface

The corresponding Pin are defined as follows

PIN No.	Definitions	Ports
1	EVENT =OUT1	IO Port (Reserved)
2	EVENT =IN1	
3	MCU TX3 232	RS232-B
4	MCU RX3 232	
5	MCU TX2 232	RS232-A
6	MCU RX2 232	
7	RTK TX1 232	RS232-RTK
8	RTK RX1 232	
9	VIN+	Positive 9-24V
10	VIN+	Positive 9-24V
11	PPS	RTK PPS
12	BCAN FD_H	CAN_FD B (Reserved)
13	BCAN FD_L	
14	ACAN_H	CAN_ A (The default baud rate is 500k)
15	ACAN_L	
16	GND	Ground
17	GND	
18	SGND	Signal ground



19	SGND		
20	PGND	Shielding	

Instruction: RS232-A Serial Port A, RS232-B Serial Port B, RS232-RTK Serial Port Differential Import, CAN A, CAN B(undefined), XT60H-M Power Supply, IO(undefined).

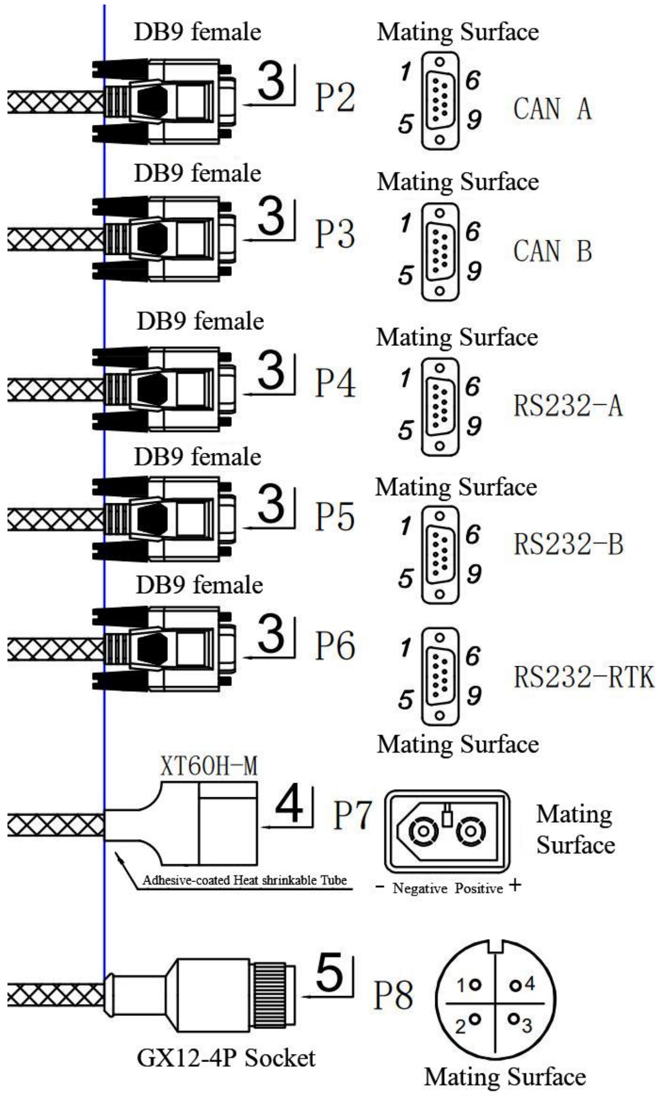
2.4. Electrical Properties

Table 2

Parameter	Symbol	Minimum	Typical	Maximum	Unit
Power Input	VIN	9	12	24	V
Ground	GND				
Power Consumption	P		1.8		W
Operating Temperature	T	-40	\	85	℃
Storage Temperature	T	-40	\	95	℃
Humidity	95% non-condensing				
Size	116*85.9*28mm				
Weight	203g				

3 Hardware Connection Method

3.1 Hardware Connection Method



3.2 Antenna

NAV680-AG leads out primary and secondary antenna interfaces without connected to the antenna. When using a multimeter to test, i.e., no-load voltage is DC4.8-5.4V; when connecting the antenna to the RF port of the module, the module is able to provide an antenna feed of DC4.6V±0.2V at room temperature with an operating current of 30-100mA. When adapting active antenna to NAV680-AG, please noted the 50Ω impedance matching between antennas.

4 Examples of Use

4.1 Device Installation

1. Install requirements:

1) Single Antenna It needs to be installed on a carrier and connected to the integrated navigation system, with the satellite antenna in the air and the carrier unobstructed to the antenna. The antenna is connected to the PRI connector shown in Fig. 21 for positioning, and the installation schematic is shown in Fig. 19;

2) Dual Antenna If needed to use the dual-antenna directional function, two antennas (primary and secondary antennas) need to be installed on the carrier and connected to the integrated navigation system. There shall be no obstruction to the antenna and the carrier, the primary antenna shall be connected to the PRI for positioning, and the secondary antenna shall be connected to the SEC interface as shown in Fig. 21 for directional purposes. Baseline shall be greater than 0.5m, and the height of the two antennas shall be kept the same as Fig. 20 shows;

3) The installation should be away from sources of interference, such as electrical devices or systems that generate electromagnetic interference signals.

4) The GNSS antenna is screwed onto a strong magnetic plate and positioned at the center of test carrier. Place the antenna on the highest point of the vessel possible to ensure good GNSS signal reception.

5) To ensure optimum performance, the distance between GNSS primary antenna and the master should be shortened, especially the horizontal distance.

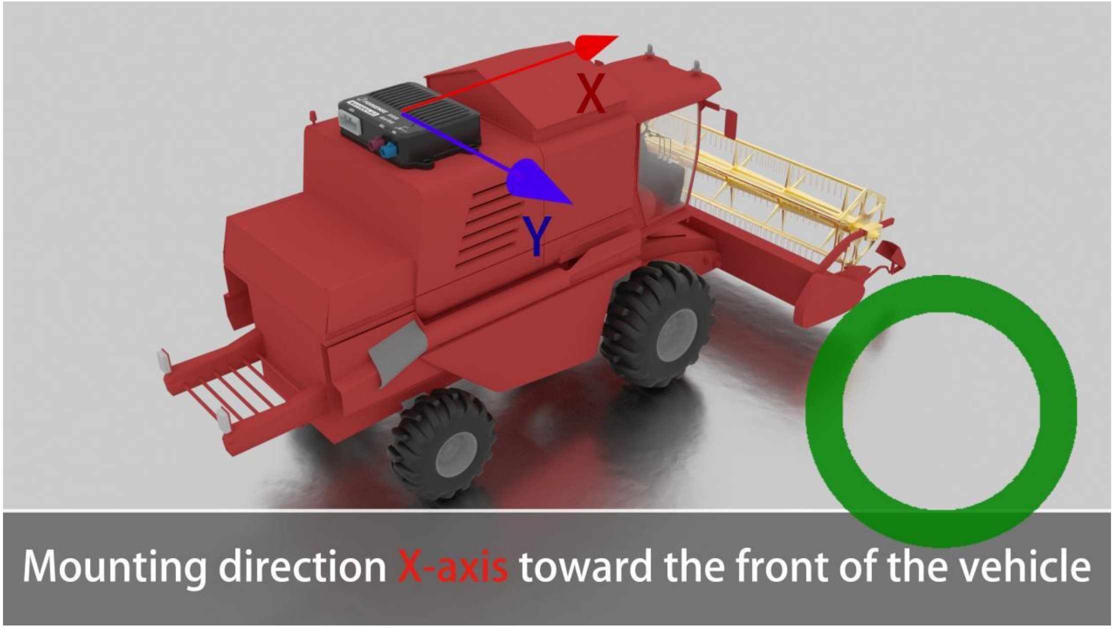
6) The module should be firmly fixed on a rigid plane rather than in a position of high vibration;

7) The x-axis should be installed toward the bow of the boat as shown below;

	
Single Antenna Installation Diagram	Dual Antenna Installation Diagram

The correct installation method is as follows

X-axis toward the front of the vehicle



The following mounting methods are incorrect:

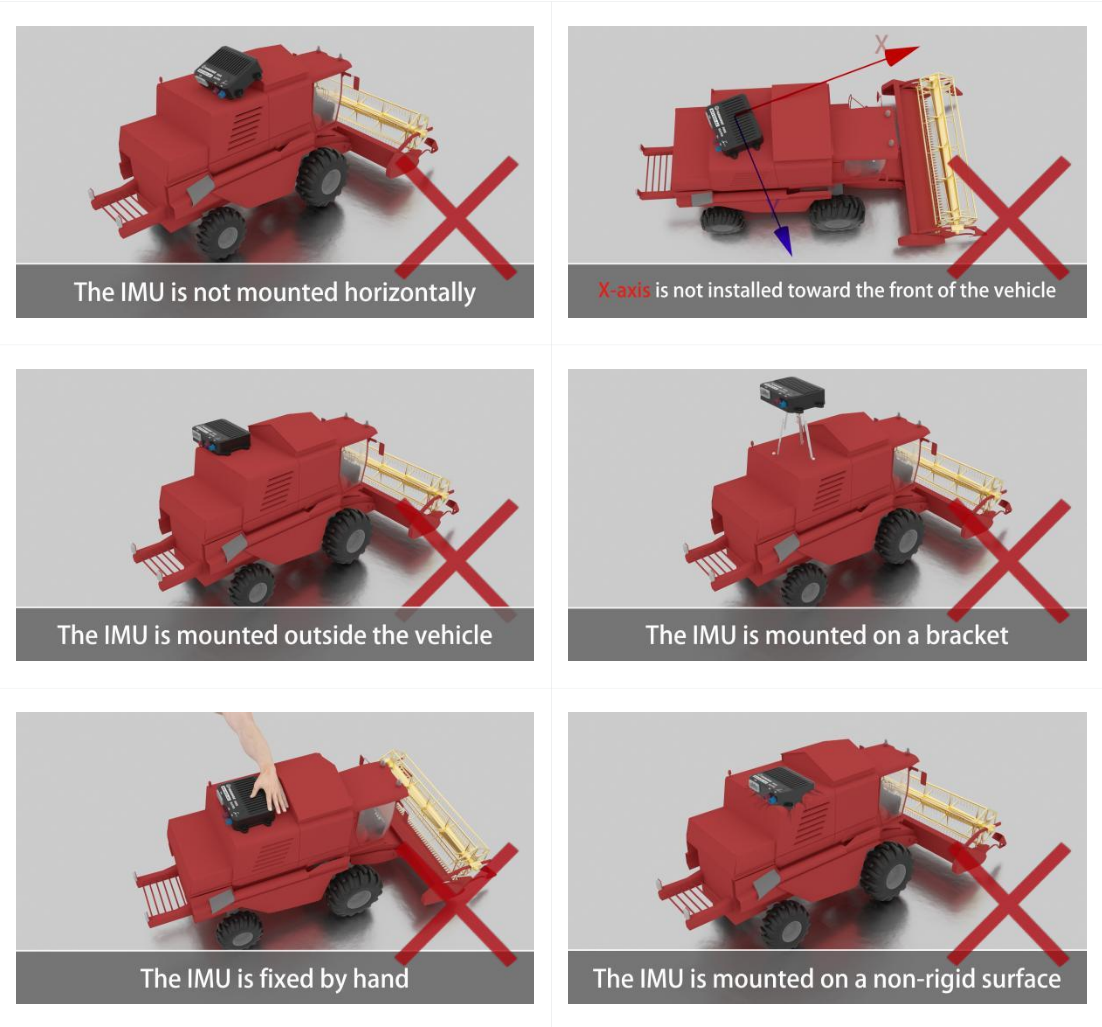


Fig. 4 Correct Installation Diagram

4.2 RTK Differential Data Import

Users have to get the GPGGA message of RTK through RS232-RTK interface, log in the network CORS account of Qianxun SI or Sixents Technology, and import the differential data, so as to make RTK enter into the fixed solution state.

4.3 Parameters Setting

4.3.1 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 which:

Measure the distance from the center of the antenna to the center of the device along the X/Y/Z axis of the boat's coordinate system, and get the three values of X1 Offset/Y1 Offset/Z1 Offset, please pay attention to the positive and negative values.

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. (Usually the antenna is above the IMU.)

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

4.3.2 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+OD_VECTOR=0.5,-0.6,1.0\r\n

Response: OD_POS_X=0.5,OD_POS_Y=-0.6,OD_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 which:

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)

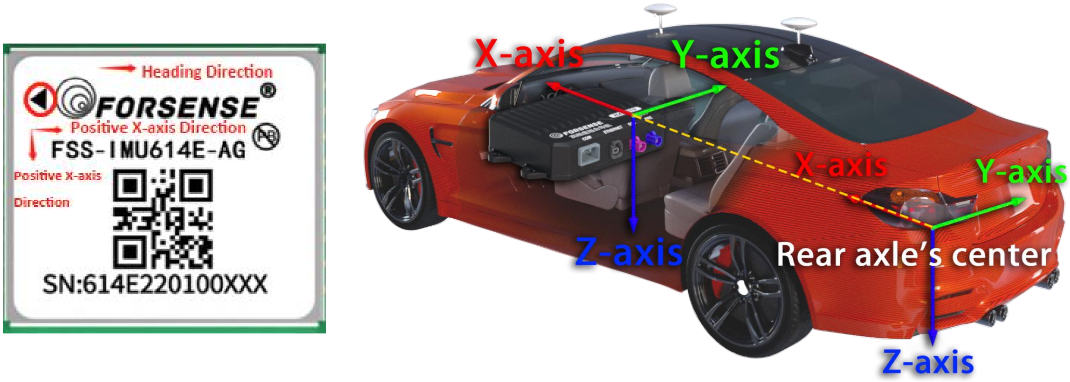


Fig. 6 Coordinate System Diagram

4.3.3 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.

4.3.4 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: Installation 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. The distance between the two antennas should be more than 50cm.

4.3.5 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.

4.3.6 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.

4.3.7 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.

4.4 Save Parameters

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

5 Definition of Coordinate System

Fig. 7 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°。

6 Output Protocols

6.1 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 3 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 Bit0 1 means the attitude is valid, 0 is invalid Bit1: 1 means front wheel angle is valid, 0 is invalid	Uint8	51
Status bits: Bit 0: 1 indicates RTK data is valid; 0 indicates invalid. Bit 1: 1 indicates the PPS signal is valid; 0 indicates invalid. Bit 2: 1 indicates the integrated navigation system is initialized; 0 indicates uninitialized. (Single-Antenna Initialization Conditions: RTK Fix status 4	Uint32	52

(Fixed) + Speed above 0.5 m/s + Linear motion with acceleration/deceleration) Dual antenna initialization conditions: RTK positioning state 4 + RTK orientation state 50) Bit3:1 indicates that the front wheel angle is valid, 0 indicates invalid. Bit4:1 indicates that the integrated navigation has converged, 0 is not converged Bit5:1 indicates that the front wheel gyro data is valid, 0 is invalid (no external gyro can be ignored) Bit6:1 indicates that the steering wheel motor data is valid, 0 is invalid Bit7 and Bit8: 01 means that the vehicle moves forward (Bit7=1, Bit8=0) 10 means that the vehicle moves backward (Bit7=0, Bit8=1) 00 means invalid (Bit7=0, Bit8=0)		
CRC verification	Uint32	56

6.2 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.
- See section 7.1.9 for configuration method, command: AT+SETNAV\r\n
- Binary data stream can not output at the same time with the NMEA data stream. Before switching the data stream, please stop the current data stream output following the command from section 7.1.11: AT+SETNO\r\n.

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 used as a reference) Dual antenna: RTK dual-antenna heading (°) Front wheel gyro G200: Z-axis angular velocity (°/s)	Float	48
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 positioning 4: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: Bit 0: 1 indicates RTK data is valid; 0 indicates invalid. Bit 1: 1 indicates the PPS signal is valid; 0 indicates invalid. Bit 2: 1 indicates the integrated navigation system is initialized; 0 indicates uninitialized. (Single-Antenna Initialization Conditions: RTK Fix status 4 (Fixed) + Speed above 0.5 m/s + Linear motion with acceleration/deceleration) Dual antenna initialization conditions: RTK positioning state 4 + RTK orientation state 50) Bit3:1 indicates that the front wheel angle is	Uint16	90

valid, 0 indicates invalid. Bit4:1 indicates that the integrated navigation has converged, 0 is not converged Bit5:1 indicates that the front wheel gyro data is valid, 0 is invalid (no external gyro can be ignored) Bit6:1 indicates that the steering wheel motor data is valid, 0 is invalid Bit7 and Bit8: 01 means that the vehicle moves forward (Bit7=1, Bit8=0) 10 means that the vehicle moves backward (Bit7=0, Bit8=1) 00 means invalid (Bit7=0, Bit8=0)		
Reserved 1	Uint32	92
Reserved 2	Uint32	96
CRC verification	Uint32	100

6.3 NMEA Protocol

- The product Supports output of combined data in NEMA format.
- Binary data stream can not output at the same time with the NMEA data stream. Before switching the data stream, please stop the current data stream output following the 7.2.2 instruction.
- Currently following statement are supported: See section 7.3 for the configuration method.

GPGGA

Chinese Description of the Message Format

Field	Structure	Descriptions	Pattern	Example
1	$GPGGA	Message Header		$GPGGA
2	utc	UTC Time (Hour/Minute/Second/Decimal Second)	hhmmss.ss	202134
3	lat	Latitude (DDmm.mm)	llll.ll	5106.9847
4	lat dir	Latitude Direction (N = North, S = South)	a	N
5	lon	Longitude (DDDmm.mm)	yyyyy.yy	11402.2986
6	lon dir	Longitude Direction (E = East, W = West)	a	W
7	quality	Solution Status	x	1
8	# sats	Number of satellites used in the solution, which may differ from the number of visible satellites.	xx	2
9	hdop	Horizontal Dilution of Precision (HDOP)	x.x	1
10	alt	Antenna Height, above or below sea	x.x	1062.22

		level		
11	a-units	Antenna Height Unit (M = metres)	M	M
12	undulation	Height Anomaly	x.x	-16.271
13	u-units	Height Anomality Unit (M = metres)	M	M
14	age	Differential Data Age, s	xx	1
15	stn ID	Differential Base Station ID	xxxx	8
16	*xx	Verification	*hh	*48
17	[CR][LF]	Sentence Abort Character		[CR][LF]

Official Message Documentation

Field	Structure	Description	Symbol	Example
1	$GPGGA	Log header		$GPGGA
2	utc	UTC time status of position (hours/minutes/seconds/ decimal seconds)	hhmmss.ss	202134.00
3	lat	Latitude (DDmm.mm)	llll.ll	5106.9847
4	lat dir	Latitude direction (N = North, S = South)	a	N
5	lon	Longitude (DDDmm.mm)	yyyyy.yy	11402.2986
6	lon dir	Longitude direction (E = East, W = West)	a	W
7	quality	refer to Table: GPS Quality Indicators	x	1
8	# sats	Number of satellites in use. May be different to the number in view	xx	10
9	hdop	Horizontal dilution of precision	x.x	1.0
10	alt	Antenna altitude above/below mean sea level	x.x	1062.22
11	a-units	Units of antenna altitude (M = metres)	M	M
12	undulation	Undulation - the relationship between the geoid and the WGS84 ellipsoid	x.x	-16.271
13	u-units	Units of undulation (M = metres)	M	M
14	age	Age of correction data (in seconds) The maximum age reported here is limited to 99 seconds.	xx	(empty when no differential data is present)
15	stn ID	Differential base station ID	xxxx	(empty when no differential data is present)
16	*xx	Check sum	*hh	*48
17	[CR][LF]	Sentence terminator		[CR][LF]

GPS Quality Indicators

Indicator	Description
0	Fix not available or invalid
1	Single point
	Converging PPP (TerraStar-L)
2	Pseudorange differential
	Converged PPP (TerraStar-L)
	Converging PPP (TerraStar-C PRO, TerraStar-X)
4	RTK fixed ambiguity solution
5	RTK floating ambiguity solution
	Converged PPP (TerraStar-C PRO, TerraStar-X)
6	Dead reckoning mode
7	Manual input mode (fixed position)
8	Simulator mode
9	WAAS (SBAS)[1]

GPRMC

Chinese Description of the Message Format

Field	Structure	Descriptions	Pattern	Example
1	$GPRMC	Message Header		$GPRMC
2	utc	Scheduled UTC Time	hhmmss.ss	144326
3	pos status	Positioning Status (A = Valid Positioning, V = Invalid Positioning)	A	A
4	lat	Latitude (DDmm.mm)	llll.ll	5107.001774
5	lat dir	Latitude Direction (N = North, S = South)	a	N
6	lon	Longitude (DDDmm.mm)	yyyyy.yy	11402.32916
7	lon dir	Longitude Direction (E = East, W = West)	a	W
8	speed Kn	Ground Speed	x.x	0.08
9	track true	Ground Track, referenced to true north	x.x	323.3
10	date	UTC Date: (Year/Month/Day)	xxxxxx	210307
11	mag var	Magnetic Declination	x.x	0
12	var dir	Magnetic Declination Direction	a	E
13	mode ind	Positioning Mode Indicator	a	A
14	*xx	Verification	*hh	*20
15	[CR][LF]	Sentence Terminator		[CR][LF]

Official Message Documentation

Field	Structure	Description	Symbol	Example
1	$GPRMC	Log header		$GPRMC
2	utc	UTC of position	hhmmss.ss	144326.00
3	pos status	Position status (A = data valid, V = data invalid)	A	A
4	lat	Latitude (DDmm.mm)	llll.ll	5107.0017737
5	lat dir	Latitude direction: (N = North, S = South)	a	N
6	lon	Longitude (DDDmm.mm)	yyyyy.yy	11402.3291611
7	lon dir	Longitude direction: (E = East, W = West)	a	W
8	speed Kn	Speed over ground, knots	x.x	0.080
9	track true	Track made good, degrees True	x.x	323.3
10	date	Date: dd/mm/yy	xxxxxx	210307
11	mag var	Magnetic variation, degrees Note that this field is the actual magnetic variation and will always be positive. The direction of the magnetic variation is always positive.	x.x	0.0
12	var dir	Magnetic variation direction E/W Easterly variation (E) subtracts from True course. Westerly variation (W) adds to True course.	a	E
13	mode ind	Positioning system mode indicator, see Table: NMEA Positioning System Mode Indicator	a	A
14	*xx	Check sum	*hh	*20
15	[CR][LF]	Sentence terminator		[CR][LF]

NMEA Positioning System Mode Indicator

Mode	Indicator
A	Autonomous
D	Differential
E	Estimated (dead reckoning) mode
M	Manual input
N	Data not valid

GPHDT (Heading information)

Chinese Description of the Message Format

Field	Structure	Descriptions	Pattern	Example
1	$GPHDT	Message Header		$GPHDT
2	heading	Yaw angle	x.x	75.5554
3	TRUE	True North Angle	T	T
4	*xx	Verification	*hh	*36
5	[CR][LF]	Sentence Terminator		[CR][LF]

Official Message Documentation

Field	Structure	Description	Symbol	Example
1	$GPHDT	Log header		$GPHDT
2	heading	Heading in degrees	x.x	75.5554
3	True	Degrees True	T	T
4	*xx	Check sum	*hh	*36
5	[CR][LF]	Sentence terminator		[CR][LF]

GPVTG (Ground velocity information)

Chinese Description of the Message Format

Field	Structure	Descriptions	Pattern	Example
1	$GPVTG	Message Header		$GPVTG
2	track true	Bearing, referenced to true north	x.x	24.168
3	T	True North Indicator	T	T
4	track mag	Bearing, referenced to magnetic north	x.x	24.168
5	M	Magnetic North Indicator	M	M
6	speed Kn	Horizontal Velocity	x.x	0.4220347
7	N	Speed Indicator, knots	N	N
8	speed Km	Horizontal Velocity, km/s	x.x	0.781608
9	K	Speed Indicator (K = km/hr)	K	K
10	mode ind	Positioning Mode Indicator	a	A
11	*xx	Verification	*hh	*7A
12	[CR][LF]	Sentence Terminator		[CR][LF]

Official Message Documentation

Field	Structure	Description	Symbol	Example
1	$GPVTG	Log header		$GPVTG
2	track true	Track made good, degrees True	x.x	24.168
3	T	True track indicator	T	T
4	track mag	Track made good, degrees Magnetic; Track mag = Track true + (MAGVAR correction) See the **MAGVAR** command	x.x	24.168
5	M	Magnetic track indicator	M	M
6	speed Kn	Speed over ground, knots	x.x	0.4220347
7	N	Nautical speed indicator (N = Knots)	N	N
8	speed Km	Speed, kilometres/hour	x.x	0.781608
9	K	Speed indicator (K = km/hr)	K	K
10	mode ind	Positioning system mode indicator, see Table: NMEA Positioning System Mode Indicator	a	A
11	*xx	Check sum	*hh	*7A
12	[CR][LF]	Sentence terminator		[CR][LF]

NMEA Positioning System Mode Indicator

Mode	Indicator
A	Autonomous
D	Differential
E	Estimated (dead reckoning) mode
M	Manual input
N	Data not valid

GPZDA (UTC time and date)

Chinese Description of the Message Format

Field	Structure	Descriptions	Pattern	Example
1	$GPZDA	Message Header		$GPZDA
2	utc	UTC Time	hhmmss.ss	220238
3	day	±01/-31	xx	15
4	month	±01/-12	xx	7
5	year	year	xxxx	1992
6	null	Area Code, Not Available	xx	(empty when no data is present)
7	null	Local Area Minute Description, Not Available	xx	(empty when no data is present)
8	*xx	Verification	*hh	*6F
9	[CR][LF]	Sentence Terminator		[CR][LF]

Official Message Documentation

Field	Structure	Description	Symbol	Example
1	$GPZDA	Log header		$GPZDA
2	utc	UTC time status	hhmmss.ss	220238.00
3	day	Day, 01 to 31	xx	15
4	month	Month, 01 to 12	xx	07
5	year	Year	xxxx	1992
6	null	Local zone description—not available Local time zones are not supported by OEM7 family receivers. Fields 6 and 7 are always null.	xx	(empty when no data is present)
7	null	Local zone minutes description—not available	xx	(empty when no data is present)
8	*xx	Check sum	*hh	*6F
9	[CR][LF]	Sentence terminator		[CR][LF]

GPATT (Forsense Customized message)

Name	Unit	Pattern	Example	Descriptions
Message Header		String	$GNATT	
UTC Time		hhmmss.sss	170834.000	17：08：34 UTC
Status bits		Character	1	0 Unavailable 1: Available
Roll angle	degree	3 decimal place	-4.891	Range: ±180°, defined as downward to the right. positive
Roll Indicator		Character	R	Roll Indicator
Pitch angle	degree	3 decimal place	3.122	Range: ±90°, defined as upward. positive
Pitch Indicator		Character	P	Pitch Indicator
Yaw angle	degree	3 decimal place	124.005	Range: 0° to 360°, defined as True North with clockwise being positive.
Roll Angle Uncertainty	degree	3 decimal place	0.432	0~360
Pitch Angle Uncertainty	degree	3 decimal place	0.811	0~360
Yaw Uncertainty	degree	3 decimal place	1.202	0~360
Checksum		HEX	*68	Used to check for transmission errors.

The format of GPATT are as follows

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

GPCHC (Forsense Customized message: GPCHC Data Protocol)

The format of GPCHC are as follows

$GPCHC,GPSWeek,GPSTime,Heading,Pitch,Roll,gyrox,gyroy,gyroz,accx,accy,accz,Lattitude,Longitude,Altitude,Ve,Vn,Vu,V,NSV1,NSV2,Status,Age,WarmingCs<CR><LF>

Note: If the system status is not initialized, the output is GNSS results, including position, velocity, and attitude (Status low nibble: 0 and 1 indicate GNSS data; 2 and 3 indicate integrated data).

Field	Name	Description	Data format	Exampls
1	Header	GPCHC Protocol header	$GPCHC	$GPCHC
2	GPSWeek	Week number since 1980-1-6 (GPS Time)	www	1980
3	GPSTime	Seconds since this Sunday at 0:00:00 (GPS Time)	ssssss.ss	16897.68
4	Heading	Yaw angle(0 to 359.99)	hhh.hh	289.19
5	Pitch	Pitch angle (-90 to 90)	+/-pp.pp	-0.42
6	Roll	Roll angle(-180 to 180)	+/-rrr.rr	0.21
7	gyro x	Gyroscope X-axis, Unit (°/s)	+/-ggg.gg	-0.23
8	gyro y	Gyroscope Y-axis, Unit (°/s)	+/-ggg.gg	0.07
9	gyro z	Gyroscope Z-axis, Unit (°/s)	+/-ggg.gg	-0.06
10	acc x	Accelerometer X-axis, Unit (G)	+/-a.aaaa	0.0009

11	acc y	Accelerometer Y-axis, Unit (G)	+/-a.aaaa	0.0048
12	acc z	Accelerometer Z-axis, Unit (G)	+/-a.aaaa	-1.0037
13	Latitude	Latitudes (-90° to 90°)	+/-ll.lllllll	38.8594969
14	Longitud e	Longitudes (-180° to 180°)	+/-ll.lllllll	121.515007 3
15	Altitude	Altitudes, unit: m	+/-aaaaa.aa	121.51
16	Ve	Eastward Speed, unit: m/s	+/-eee.eee	-0.023
17	Vn	Northward Speed, unit:m/s	+/-nnn.nnn	0.011
18	Vu	Skyward Speed, unit: m/s	+/-uuu.uuu	0.000
19	V	Car Speed, unit: m/s	+/-uuu.uuu	1.500
20	NSV1	Primary antenna 1 Satellite number	nn	14
21	NSV2	Secondary antenna 2 Satellite number	nn	6
22	Status	System Status (low half byte): 0: Initialization ; 1: Satellite Navigation Mode ; 2: Integrated Navigation mode ; 3: Native inertial sensor mode ;Satellite status(high half byte): 0: no position and orientation; 1: Single point position and orientation; 2: Pseudo-range differential positioning and orientation;3: integrated calculation; 4:RTK stationary solution position and orientation;5: RTK floating position and orientation;6: Single point position and without orientation;7: Pseudo-range differential positioning and without orientation;8: RTK stationary solution positioning and without orientation;9: RTK floating solution positioning and without orientation; RTK Float Solution Positioning, No Orientation;	ss	4
23	Age	Differential delay	aa	0
24	Warning	bit0: 1 No GPS message, 0: Normal bit1: 1 No vehicle message, 0: Normal bit3: 1 Gyroscope error, 0: Normal bit4: 1 Accelerometer error, 0: Normal	ww	2
25	Cs	Verification	*hh	*47
26	<CR><L F>	Fixed datagram tail		<CR><LF>

6.4 CAN 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 0X19FFCE9B (Note: Velocity direction: 1: Forward, -1: Reverse)

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

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
0X19CCFF9A	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			

Table 15 CAN Extended Frame Format 0X19FFD29A (Note: Unit is cm/s, int16 type)

Extended Frame ID	1	2	3	4	5	6	7	8
19FFD29A	Northward velocity		Eastward velocity		Ground velocity		Ground Speed (Horizontal Speed)	

6.5 RTK Positioning Stats Diagram

Binary	ASCII	Descriptions
0	NONE	unsolvable
1	FIXEDPOS	The position is fixed by FIX POSITION command.
2	FIXEDHEIGHT	Not supported
8	DOPPLER_VELOCITY	The velocity is exported by real-time Doppler information.
16	SINGLE	Single Point:
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 land fixed solution

7 Parameters Setting

7.1 Use CAN Interface to Configure Parameters

7.1.1 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 which:

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. (Usually the antenna is above the IMU.)

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)

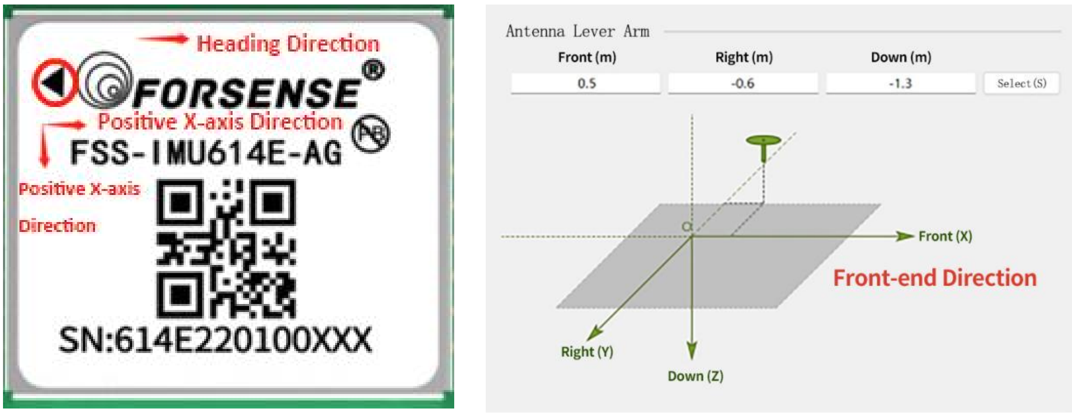


Fig. 8 Antenna Lever Arm Diagram

7.1.2 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+OD_VECTOR=0.5,-0.6,1.0\r\n

Response: OD_POS_X=0.5,OD_POS_Y=-0.6,OD_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 which:

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)

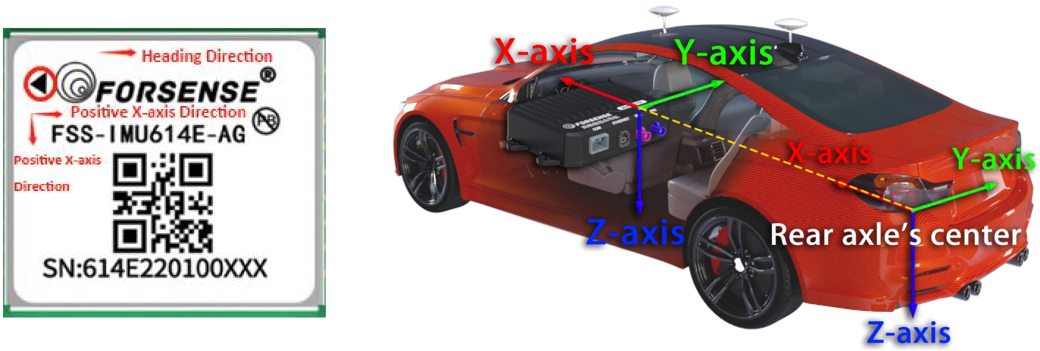


Fig. 9 Coordinate System Diagram

7.1.3 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.

7.1.4 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: Installation 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. The distance between the two antennas should be more than 50cm.

7.1.5 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.

7.1.6 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.

7.1.7 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.

7.1.8 Configure the Binary Output - AG Data Stream

If AG data stream output is configured, then Section 7.2 Integrated Navigation Data Stream and Section 7.3 NMEA Protocol will not output data.

If wanted to switch to 7.2 INS data stream or 7.3 NMEA protocol output, please stop the current data stream output by pressing the 7.1.11 command first.

Command: AT+SETAG\r\n

Response: OK\r\n

If configured to stop output,

Command: AT+SETNO\r\n

Response: OK\r\n

7.1.9 Binary Protocols - INS Data Stream

If configured integrated navigation data stream output, then AG data stream in section 7.1 and NMEA protocol in section 7.3 stop output.

If wanted to switch to 7.2 INS data stream or 7.3 NMEA protocol output, please stop the current data stream output by pressing the 7.1.11 command first.

The configuration commands are as follows

Command: AT+SETNAV\r\n

Response: OK\r\n

If configured to stop output,

Command: AT+SETNO\r\n

Response: OK\r\n

7.1.10 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 wanted to switch to 7.2 INS data stream output, please stop the current data stream output by pressing the 7.2.11 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

7.1.11 Stop the Output of the Current Data Stream

Command: AT+SETNO\r\n

Response: OK\r\n

7.1.12 Configure the Data Output Frequency

If the configuration data output frequency is 10Hz, the configuration command is:

100HZ Supported

Command: AT+OUTRATE=10\r\n

Response: OK\r\n

7.1.13 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.

7.1.14 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.

7.1.15 Configure CAN Serial Output Frequency

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.

7.1.16 Output all Configuration Information

If you want to query all the configured information, command:

AT+CONFIG\r\n

7.1.17 Query Version Numbers

Command: Command: AT+VERSION\r\n

7.1.18 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

7.1.19 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.

7.1.20 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

7.1.21 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

7.2 Use CAN Interface to Configure Parameters

7.2.1 Querying Version Numbers

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

7.2.2 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)

7.2.3 Configure the Dual Antenna Mounting Angle

ID = 0X0DFFC326

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

7.2.4 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

7.2.5 Configure CAN Interface to Output Update Rates and Baud Rates

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

7.2.6 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

7.2.7 Save Parameter Command

ID = 0X0DFFCF26

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

Response

ID = 0X19FFCF9A

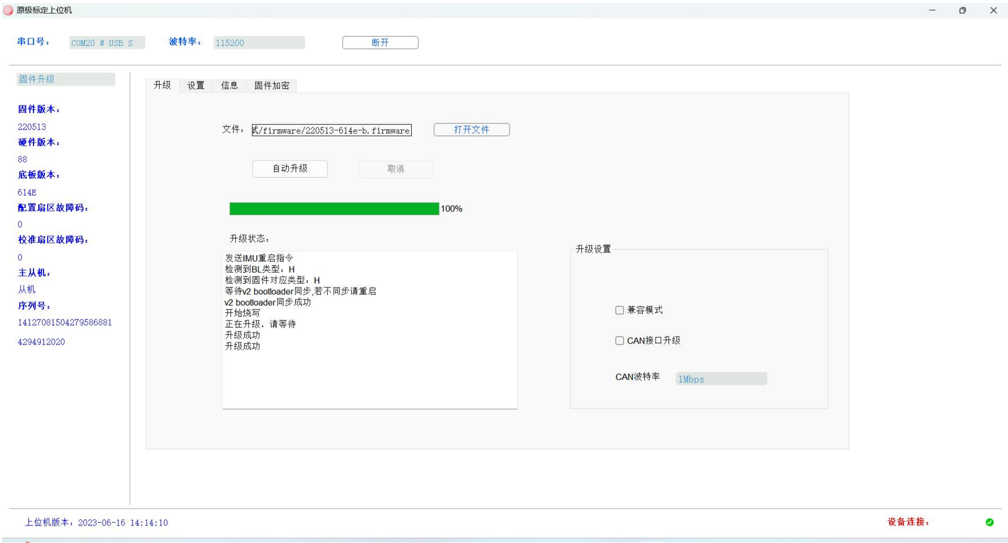
Data =0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF

8 Firmware Upgrading

8.1 Firmware Upgrading via the PC Software

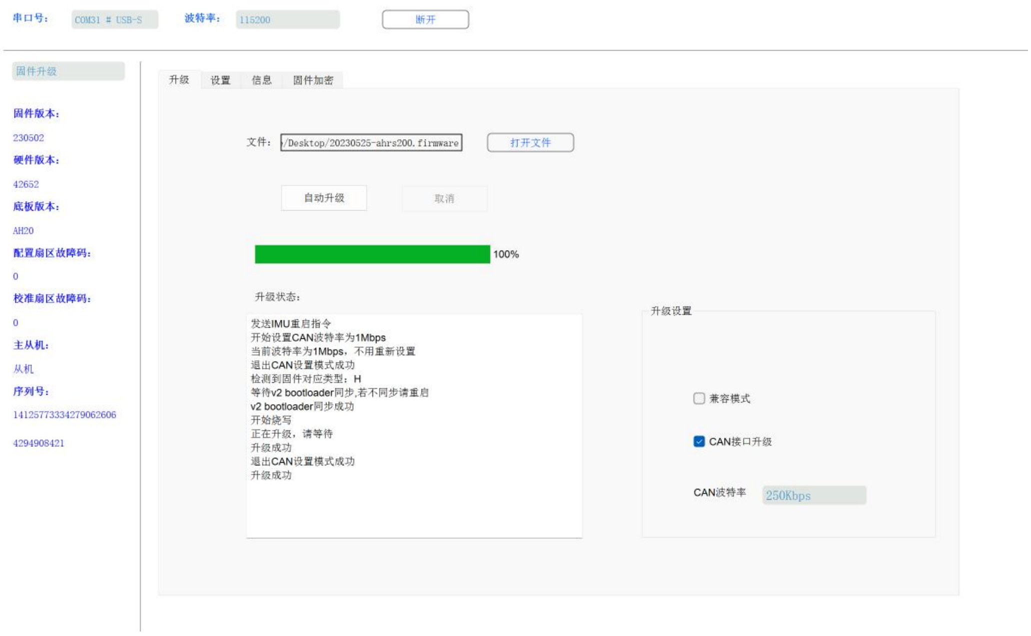
8.1.1 RS232 Serial Port Direct Connection

Use the Forsense testing PC software - Select firmware upgrade - Open the firmware - Click Auto Upgrade.



8.1.2 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.



8.2 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.

In BOOTLOADER.

```
上位机发送升级指令如下:
  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};
  此时 imu 模块将会重启后进入 bootloader 中
```

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.

```
//发送完升级指令后，紧接着发送如下字符串，确保 imu 模块不会跳转到 APP 区域，同时处于待升级状态
```

```
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.

```
上位机与 imu 模块同步过程：
1.上位机发送同步命令，命令码为 0x21，发送后等待 imu 模块响应
 Send_CMD_LONG(0x21,0,0,0,0,0,0);
2.imu 模块收到 0x21 的命令码后，发送响应数据，数据命令如下：
Send_CMD_ACK(0x64,0x10,0);
3.上位机判断是否收到了 imu 的响应数据，如收到则表示完成了同步
注：Send_CMD_LONG 函数与 Send_CMD_ACK 函数见文章末尾
```

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.上位机发送擦除命令，命令码为 0x23，发送完此命令后耐心等待 imu 回复执行结果
  Send_CMD_LONG(0x23,0,0,0,0,0,0);
2.imu 收到 0x23 的命令码后，将会把 APP 区域数据擦除干净，并将最终的执行结果回复给上位机
   send_CMD_ACK(0x64,0x10,0); //表示擦除成功
```

```
   send_CMD_ACK(0x64,0x11,0); //表示擦除失败，需要重新擦除
   注意：上位机必须收到擦除成功的响应，才能进行下一步
```

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.

```
例：将大小 1000 字节的升级文件 uint8_t Upgrade_Data[1000]发送给 imu 模块
 //发送第一包：
  1.1 通过函数将 0~63 字节发送到 imu 模块
  Send_Upload_Data（0x27,0,0,0x40,Upgrade_Data);
  //上面函数第一参数 0x27 为固定值，第二参数 0 为固定值，第三参数 0 为偏移地址，
    第四参数 0x40 为有效字节长度，第五参数为发送数据的首地址

 1.2 imu 成功收到数据后，会发送响应数据，并根据偏移地址写入指定的 flash 地址
   send_CMD_ACK(0x753D,0x00,0);
   //上面函数第一参数 0x753D 为固定值，第二参数 0 为固定值，第三参数为偏移地址。
   imu 模块将数据写入 flash 过程中如果写入失败将发送失败命令  send_CMD_ACK(0x64,0x11,0);

   //发送第二包：
   2.1  上位机收到 imu 响应数据后，将发送第二包数据 Send_Upload_Data（0x27,0,0x40,0x40,Upgrade_Data+0x40);
   2.2  imu 成功收取到第二包数据后，发送响应：  send_CMD_ACK(0x753D,0x00,0x40);
   2.3  imu 模块将数据写入 flash 过程中如果写入失败将发送失败命令  send_CMD_ACK(0x64,0x11,0);


   //发送第三包
   3.1  上位机收到 imu 响应数据后，将发送第三包数据 Send_Upload_Data（0x27,0,0x80,0x40,Upgrade_Data+0x80);
   3.2  imu 成功收取到第三包数据后，发送响应：  send_CMD_ACK(0x753D,0x00,0x80);
   3.3  imu 模块将数据写入 flash 过程中如果写入失败将发送失败命令  send_CMD_ACK(0x64,0x11,0);
  ----------
  //发送第十五包：
```

```
   15.1  上位机收到  imu 响应数据后，将发送第十五包数据 Send_Upload_Data（0x27,0,0x380,0x40,Upgrade_Data+0x380);
   15.2  imu 成功收取到第十五包数据后，发送响应：  send_CMD_ACK(0x753D,0x00,0x380);
   15.3  imu 模块将数据写入 flash 过程中如果写入失败将发送失败命令  send_CMD_ACK(0x64,0x11,0);
   //发送第十六包：
   15.1  上位机收到  imu 响应数据后，将发送第十六包数据 Send_Upload_Data（0x27,0,0x3C0,0x28,Upgrade_Data+0x3C0);
   15.2  imu 成功收取到第十六包数据后，发送响应：  send_CMD_ACK(0x753D,0x00,0x3C0);
   15.3  imu 模块将数据写入 flash 过程中如果写入失败将发送失败命令  send_CMD_ACK(0x64,0x11,0);
  发送结束
```

Step 6: Acquire **CRC** check digits

Usually 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(0x29,0,0,0,0,0,0,0,0,0) 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.

```
上位机发送获取 crc 校验码指令，等待 imu 响应
Send_CMD_LONG(0x29,0,0,0,0,0,0);
imu 模块响应发送 crc 校验值数据:
send_CMD_ACK(0x753C,0x10,crc32_data);
其中 crc32_data 值为 imu 模块本身计算的 crc32 数据
```

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.

```
判断 crc 校验值正确后，发送重启命令:
Send_CMD_LONG(0x30,0,0,0,0,0,0);
```

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));
}
```

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,
```

9 ROS Driver

ROS driver download address: https://data.forsense-imu.com/ProductFile/en/link/Upper%20computer%20software/NAV%20ROS2%20dirve/download.php

9.1 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

9.2 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

9.3 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

9.4 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: [-0.00536619778722546, 0.00353265739977359 77, -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
---
```

10 Used with G200

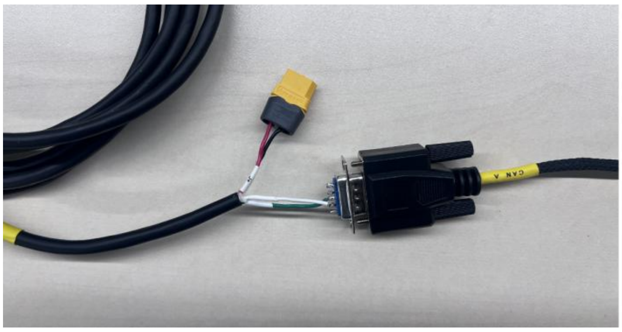
1. G200 mounting requirements:

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.

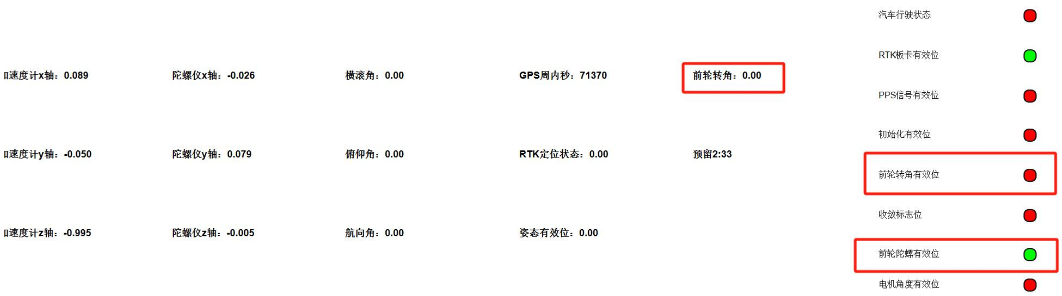
Fig. 10 Wiring Diagram



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 traveling.

Fig. 12 Correct Access to G200



11 Accessories

Standard accessories:

NAV680-AG host

Harness

Primary antenna (for positioning)

Secondary antenna (for orientation)

Primary antenna connector

Secondary antenna connector

Standard accessories:

RS232 Serial Cable XT60

G200

11 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: 1. If directional status is always 0, please check if the secondary antenna is properly connected and if the antenna feeder is damaged. 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 equipment 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: Single antenna: 1. Fixed solution of the positioning status; 2. Traveling speed above 0.5m/s; 3. Driving straight with acceleration and deceleration. Dual antenna: 1. Fixed solution of the positioning status; 2. Orientation status 50
COM packet loss	Please check the following: 1. The serial cable needs to support at lease 115200 Baud Rates. 2. Computer's serial delay needs to be configured to 2ms
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: 1. The prerequisite for correct front wheel angle output is that the AG device has to complete the initialization before it can output the correct front wheel angle. 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: Little-endian mode, low bits places in the front, unit: cm, and 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: 1. If G200 is horizontal installed 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: 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: 1. Wrong configuration in lever arm and wheelbase, which unit has amplified by 10 times. 2. Please make sure if it's rigidly fixed. 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.

12 Revision History

Latest manual version address: NAV680-AG_Product Sheet

Version	Date	Status/Notes
Version 1.0	01/09/2024	First release
Version 1.1	01/12/2024	Added PTK Performance Indicators
Version 1.2	01/15/2024	Added revision history
Version 1.3	01/18/2024	Adjusted primary and secondary antenna attachment pictures
Version 1.4	01/24/2024	Deleted CAN B
Version 1.5	01/26/2024	1. Added related content of CAN protocol, 2. Added RTK positioning status indicators
Version 1.6	02/20/2024	Revised lever arm configuration description
Version 1.7	02/22/2024	Corrected Rotation Angle Configuration Answer Description
Version 1.8	03/26/2024	Added slope mode and using methods of combining IMU614E-AG with G200
Version 1.9	06/12/2024	1. Added some new AT commands 2. Added CAN protocol latitude, longitude, and elevation messages. 3. Added spreadsheet for frequent questions
Version 1.10	01，23，2025，	Adjusted table and figure sequence