LC29H (BA,CA,DA,EA)&QLM29H Series DR&RTK Application Note

GNSS Module Series

Version: 1.3

Date: 2024-09-30

Status: Released

At Quectel, our aim is to provide timely and comprehensive services to our customers. If you require any assistance, please contact our headquarters:

Quectel Wireless Solutions Co., Ltd.
Building 5, Shanghai Business Park Phase III (Area B), No.1016 Tianlin Road, Minhang District, Shanghai 200233, China
Tel: +86 21 5108 6236
Email: info@quectel.com

Or our local offices. For more information, please visit:
http://www.quectel.com/support/sales.htm.

For technical support, or to report documentation errors, please visit:
http://www.quectel.com/support/technical.htm.
Or email us at: support@quectel.com.

Legal Notices

We offer information as a service to you. The provided information is based on your requirements and we make every effort to ensure its quality. You agree that you are responsible for using independent analysis and evaluation in designing intended products, and we provide reference designs for illustrative purposes only. Before using any hardware, software or service guided by this document, please read this notice carefully. Even though we employ commercially reasonable efforts to provide the best possible experience, you hereby acknowledge and agree that this document and related services hereunder are provided to you on an “as available” basis. We may revise or restate this document from time to time at our sole discretion without any prior notice to you.

Use and Disclosure Restrictions

License Agreements

Documents and information provided by us shall be kept confidential, unless specific permission is granted. They shall not be accessed or used for any purpose except as expressly provided herein.

Copyright

Our and third-party products hereunder may contain copyrighted material. Such copyrighted material shall not be copied, reproduced, distributed, merged, published, translated, or modified without prior written consent. We and the third party have exclusive rights over copyrighted material. No license shall be granted or conveyed under any patents, copyrights, trademarks, or service mark rights. To avoid ambiguities, purchasing in any form cannot be deemed as granting a license other than the normal non-exclusive, royalty-free license to use the material. We reserve the right to take legal action for noncompliance with abovementioned requirements, unauthorized use, or other illegal or malicious use of the material.

Trademarks

Except as otherwise set forth herein, nothing in this document shall be construed as conferring any rights to use any trademark, trade name or name, abbreviation, or counterfeit product thereof owned by Quectel or any third party in advertising, publicity, or other aspects.

Third-Party Rights

This document may refer to hardware, software and/or documentation owned by one or more third parties ("third-party materials"). Use of such third-party materials shall be governed by all restrictions and obligations applicable thereto.

We make no warranty or representation, either express or implied, regarding the third-party materials, including but not limited to any implied or statutory, warranties of merchantability or fitness for a particular purpose, quiet enjoyment, system integration, information accuracy, and non-infringement of any third-party intellectual property rights with regard to the licensed technology or use thereof. Nothing herein constitutes a representation or warranty by us to either develop, enhance, modify, distribute, market, sell, offer for sale, or otherwise maintain production of any our products or any other hardware, software, device, tool, information, or product. We moreover disclaim any and all warranties arising from the course of dealing or usage of trade.

Privacy Policy

To implement module functionality, certain device data are uploaded to Quectel's or third-party's servers, including carriers, chipset suppliers or customer-designated servers. Quectel, strictly abiding by the relevant laws and regulations, shall retain, use, disclose or otherwise process relevant data for the purpose of performing the service only or as permitted by applicable laws. Before data interaction with third parties, please be informed of their privacy and data security policy.

Disclaimer

a) We acknowledge no liability for any injury or damage arising from the reliance upon the information.
b) We shall bear no liability resulting from any inaccuracies or omissions, or from the use of the information contained herein.
c) While we have made every effort to ensure that the functions and features under development are free from errors, it is possible that they could contain errors, inaccuracies, and omissions. Unless otherwise provided by valid agreement, we make no warranties of any kind, either implied or express, and exclude all liability for any loss or damage suffered in connection with the use of features and functions under development, to the maximum extent permitted by law, regardless of whether such loss or damage may have been foreseeable.
d) We are not responsible for the accessibility, safety, accuracy, availability, legality, or completeness of information, advertising, commercial offers, products, services, and materials on third-party websites and third-party resources.

About the Document

Document Information	
Title	**LC29H (BA,CA,DA,EA)&QLM29H Series DR&RTK Application Note**
Subtitle	GNSS Module Series
Document Type	Application Note
Document Status	Released

Revision History

Version	Date	Description
-	2022-02-25	Creation of the document
1.0	2022-09-19	First official release
1.1	2023-04-27	1. Added applicable module LC29H (EA). 2. Added an overview on DR (*Chapter 1.1*). 3. Changed the driving speed of more than 3 m/s to 2 m/s in step 4 of DR calibration, updated the testing scenarios in open sky area, urban main road, and added a note about scenarios where the module exits DR mode (*Chapter 2.7*). 4. Added PQTMCFGDRRTD, PQTMCFGIMUTC, PQTMDRPVA, PQTMCFGDRHOT and PQTMCFGDR commands (*Chapters 4.1.12*, *4.1.13*, *4.1.14*, *4.1.15* and *4.1.16*).
1.2	2024-07-24	1. Added applicable device QLM29H series. 2. Updated the overview on DR, and added overviews on RTK and DR + RTK (*Chapter 1*). 3. Added the following DR related information: ● Lever arm (*Chapter 2.2*); ● DR calibration data saving (*Chapter 2.3*); ● DR hot start (*Chapter 2.4*); ● Temperature compensation (*Chapter 2.5*);

Version	Date	Description
		● Module/device mounting (*Chapter 2.6*). 4. Added supported RTCM input messages 1075, 1085, 1095, 1115 and 1125, and added a note about distance between the rover station and the base station (*Chapter 3.1*). 5. Updated the supported values of <ErrCode> (*Table 2: Error Codes*). 6. Updated the supported values of <CalState> in PQTMDRCAL and added notes on message configuration (*Chapter 4.1.1*). 7. Updated the supported values of <Type> in PQTMIMUTYPE (*Chapter 4.1.2*). 8. Updated the note about supported values of <MsgVer> in PQTMVEHMSG (*Chapter 4.1.3*). 9. Updated the ranges of <Roll>, <Pitch>, and <Yaw> in PQTMINS (*Chapter 4.1.6*). 10. Updated the description of <FixMode> and the range of <Heading> in PQTMGPS, and added note on the message output result (*Chapter 4.1.8*). 11. Updated the description of <Rate> in PQTMCFGEINSMSG and added a note on enabling the message (*Chapter 4.1.9*). 12. Added notes on enabling PQTMSENMSG (*Chapter 4.1.11*). 13. Added a note on modules and devices supporting PQTMCFGDRRTD message (*Chapter 4.1.12*). 14. Added a note on modules and devices supporting PQTMCFGIMUTC and PQTMCFGDR messages (*Chapters 4.1.13* and *4.1.16*). 15. Added notes on modules and devices supporting PQTMDRPVA and on enabling the message (*Chapter 4.1.14*). 16. Added the following PQTM messages: ● PQTMDRSAVE (*Chapter 4.1.17*); ● PQTMDRCLR (*Chapter4.1.18*); ● PQTMCFGRTK (*Chapter 4.1.19*); ● PQTMCFGLA (*Chapter 4.1.20*); ● PQTMCFGLAM (*Chapter 4.1.21*); ● PQTMVEHATT (*Chapter 4.1.22*); ● PQTMCFGMSGRATE (*Chapter 4.1.23*); ● PQTMCFGSTATICHOLD (*Chapter 4.1.24*). 17. Added the note on the recommended message to configure PQTM messages (*Chapters 4.2.1* and *4.2.2*). 18. Added information on operation guide (*Chapters 5*). 19. Added common issues and troubleshooting (*Chapter 6*).
1.3	2024-09-30	1. Added the note about the output of PQTMIMUTYPE by LC29H (BA, CA) with firmware versions dedicated for two-wheel vehicles (*Chapter 4.1.2*). 2. Deleted the note about the limitations of PQTMVEHMSG (If <MsgVer> = 2) message for LC29H (BA, CA) with firmware versions dedicated for two-wheel vehicles (*Chapter 4.1.3.2*).

Version	Date	Description
		3. Added the following PQTM messages: ● PQTMVEHMOT (<MsgVer> = 2) (*Chapter 4.1.10.2*); ● PQTMSENMSG (<MsgVer> = 2) (*Chapter 4.1.11.1*). 4. Updated the description of <RelMode> (*Chapter 4.1.19*). 5. Updated the supported messages of PQTMCFGMSGRATE by adding PQTMIMUTYPE (*Table 3*). 6. Added PQTMCFGVEHMOT message (*Chapter 4.1.25*). 7. Updated the <Output_State> to <Output_Rate> in PAIR6010 and PAIR6011 and updated related note (*Chapters 4.2.1* and *4.2.2*).

Contents

Table Index

Figure Index

1 Introduction

This document describes the dead reckoning (DR) and real-time kinematic (RTK) features, including DR and RTK configurations and DR and RTK related messages for Quectel QLM29H series device and LC29H (BA), LC29H (CA), LC29H (DA) and LC29H (EA) modules. The features supported by each module/device are as follows:

- QLM29HBAA-GM, QLM29HBAP-GM and LC29H (BA) supports DR and RTK.
- QLM29HCAA-GM, QLM29HCAP-GM and LC29H (CA) only supports DR.
- LC29H (DA) only supports RTK (update rate: 1 Hz).
- LC29H (EA) only supports RTK (max. update rate: 10 Hz, 10 Hz by default).

1.1. DR Overview

1.1.1. DR Introduction

LC29H (BA), LC29H (CA), and QLM29H series provide two DR modes: ADR (Automotive Dead Reckoning) and UDR (Untethered Dead Reckoning). ADR requires the speed and reverse signal from the vehicle. You can connect the module to vehicle sensors or inject the speed and reverse signal via the module's communication interfaces. On the contrary, the UDR does not require the speed and reverse signal, and the module integrates high-end DR technology to provide enhanced positioning performance without vehicle sensors. Therefore, UDR can be more easily installed. It's an ideal solution for aftermarket applications where access to vehicle speed data is not possible. LC29H (BA), LC29H (CA) and QLM29H series can automatically identify the availability of speed data, hence enabling automatic switch between the UDR and ADR modes after each power on.

In addition, Quectel provides QLM29H series with firmware versions designed for four-wheel vehicles, while LC29H (BA) and LC29H (CA) modules are provided with two different firmware versions:

- Firmware versions dedicated for four-wheel vehicles, such as cars;
- Firmware versions dedicated for two-wheel vehicles, such as bicycles, scooters, and motorcycles.

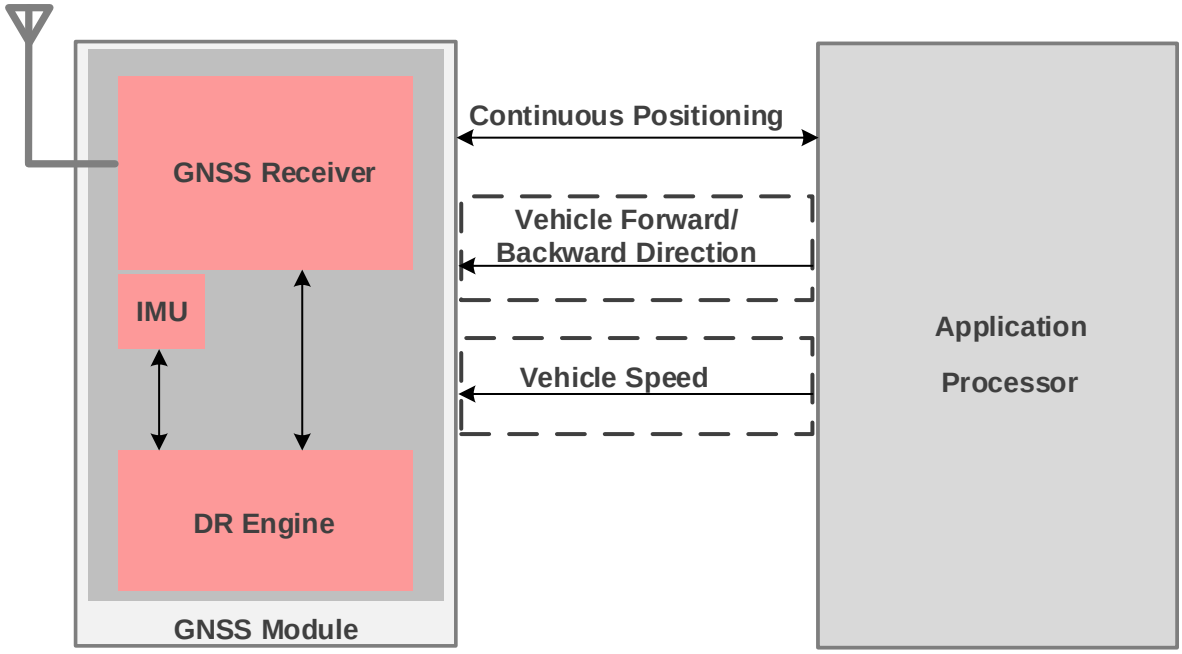


Figure 1: DR Application Architecture

1.1.2. DR Application Scenarios

DR application scenarios include dense forests, urban canyons with tall buildings, areas under viaducts, and even tunnels and underground parking.

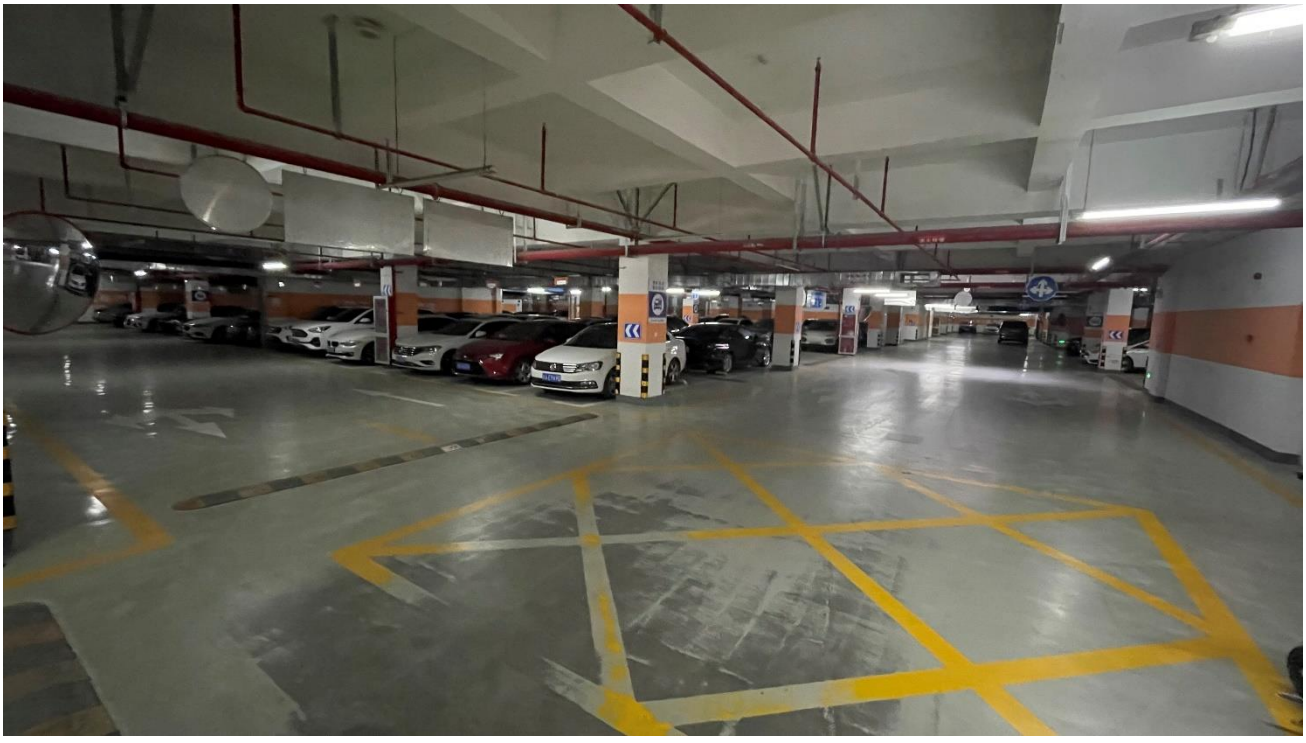


Figure 2: Underground Parking

1.2. RTK Overview

1.2.1. RTK Introduction

RTK (Real Time Kinematic) is a real-time differential GNSS technology based on carrier phase measurements, which achieves high-precision positioning with centimeter-level accuracy. For more details about RTK, see *document [1] GNSS RTK application note*.

1.2.2. RTK Application Scenarios

The main application scenarios of RTK include lawn mowers, drones, and lane-level navigation.

Figure 3: UAV

1.3. DR + RTK Overview

1.3.1. DR + RTK Introduction

QLM29HBAA-GM, QLM29HBAP-GM, and LC29H (BA) support both DR and RTK. In many scenarios, DR and RTK are used together for a better positioning performance.

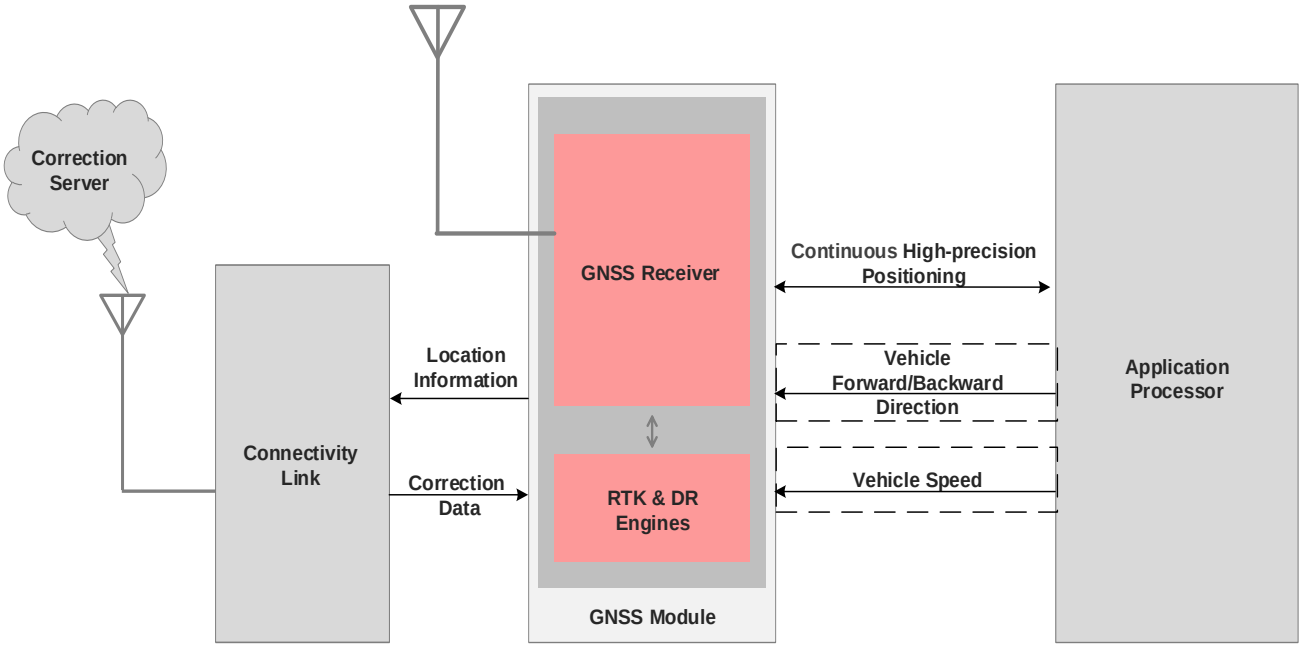


Figure 4: DR + RTK Application Architecture

1.3.2. DR + RTK Application Scenarios

Main application scenarios of DR + RTK include high-level autonomous driving, shuttle bus, and automatic inspection robots.

Figure 5: Autonomous Vehicle

2 DR

2.1. Orientation

The LC29H (BA) and LC29H (CA) modules are designed to work on two-wheel or four-wheel vehicles, while the QLM29H series is designed for use on four-wheel vehicles, all of which integrate an IMU as well as the GNSS receiver. Therefore, you must ensure that the device incorporating LC29H (BA, CA)/QLM29H series is firmly fixed to vehicle body. No relative movement is allowed between vehicle, GNSS antenna, and device, and maximum isolation from shock or vibration must be applied. Manually holding the device is not acceptable. The best way to guarantee good installation is to firmly screw the device down to the vehicle frame. Mounting location should permit easy access to power supply and GNSS antenna, and should not be exposed to excessive heat.

Definitions of reference frame axes:

- X-axis points towards the right of the vehicle.
- Y-axis points towards the front of the vehicle.
- Z-axis points towards the roof of the vehicle.

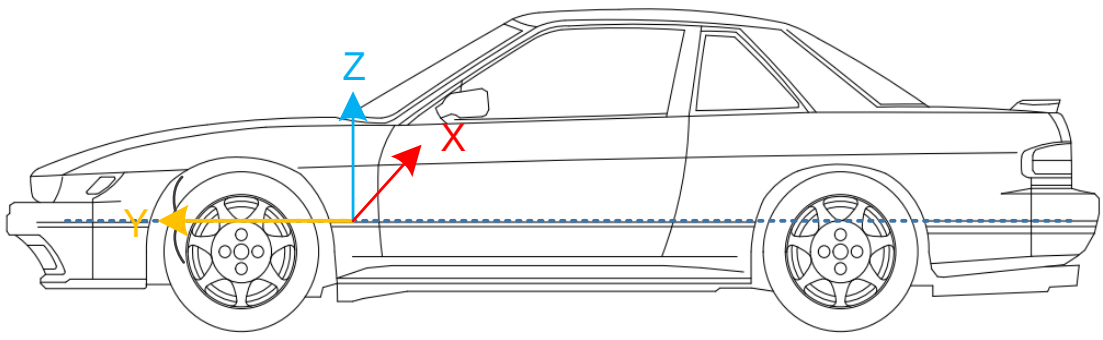


Figure 6: Reference Frame

Module orientation of LC29H (BA) and LC29H (CA) is shown below:

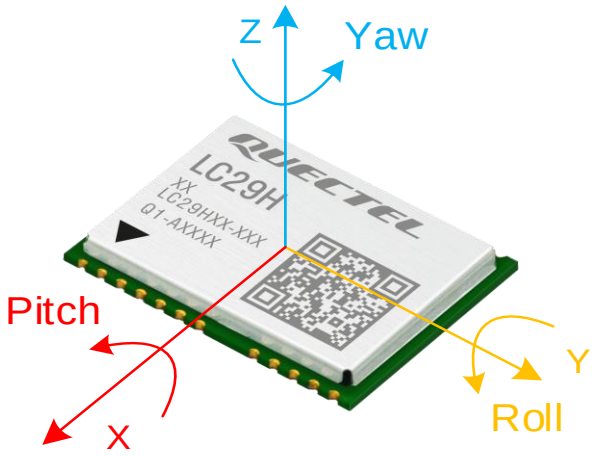


Figure 7: Module Orientation

Device orientation of QLM29H series is shown below (using QLM29HBAA-GM as an example):

- Z-axis is vertically downward.

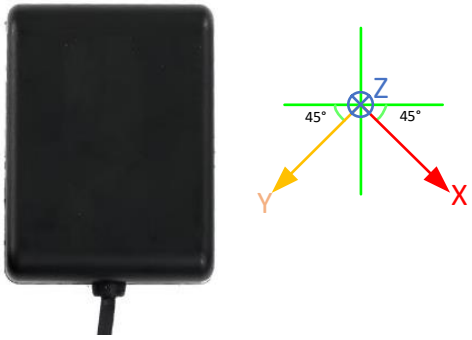


Figure 8: QLM29H Series Device Orientation

2.2. Lever Arm

QLM29H series, LC29H (BA) and LC29H (CA) support free mounting that enables automatic estimation of the lever arm. In addition, they offer the flexibility of manual lever arm configuration. The lever arm configuration refers to the vehicle frame. For details, see *Chapter 4.1.20 PQTMCFGLA*.

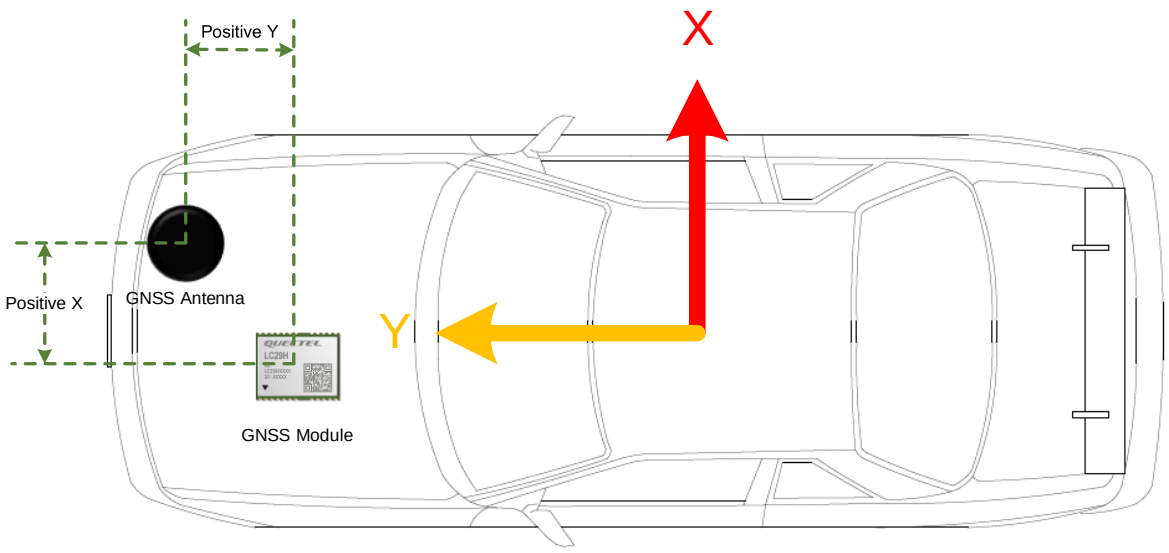


Figure 9: Positive X and Positive Y Lever Arm

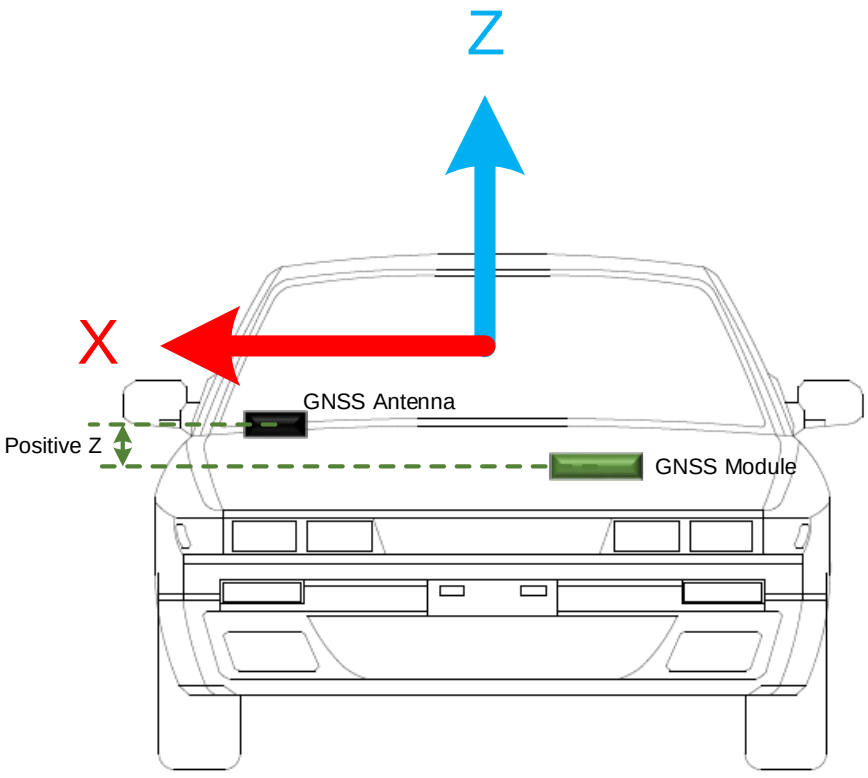


Figure 10: Positive Z Lever Arm

2.3. DR Calibration Data Saving

LC29H (BA, CA)/QLM29H series support both automatic and manual saving of DR calibration data. When the **<Mode>** is 1 or 2 in **PQTMCFGDRHOT** (see *Chapter 4.1.15 PQTMCFGDRHOT* for details), the automatic saving function is enabled. Additionally, LC29H (BA, CA)/QLM29H series allow DR calibration data to be saved manually by issuing the **PQTMDRSAVE** command (see *Chapter 4.1.17 PQTMDRSAVE* for details) once the vehicle has come to a complete stop. Do not change the installation position of the module/device after it has been fully calibrated, otherwise trajectory anomalies may occur when it operates after the next power-up. If the installation position changes after calibration is completed, you must clear the DR related data using **PQTMDRCLR**. See *Chapter 4.1.18 PQTMDRCLR* for details.

2.4. DR Hot Start

LC29H (BA, CA)/QLM29H series support the DR hot start feature. When this feature works, the DR engine within the module/device can immediately output the current position coordinates and perform position estimation after the vehicle starts and before the GNSS receiver fixes a position, without the need for recalibration. This feature is enabled by default and can automatically identify application scenarios such as underground garages. Note that the following conditions must be met for automatic scenario recognition:

- The module/device has been calibrated.
- After the vehicle comes to a complete stop, wait for 10 seconds before cutting off the power.
- Once the power is cut off, neither the installation position of the module/device nor the vehicle's position should have changed.

This feature cannot fully recognize all GNSS signal-deprived scenarios, and you can actively save DR calibration data by using **PQTMDRSAVE** when needed in the current scenario to achieve the DR hot start feature upon restart. For details on how to save DR calibration data, refer to *Chapter 2.3 DR Calibration Data Saving*.

For details on the DR hot start configuration, see *Chapter 4.1.15 PQTMCFGDRHOT*.

NOTE

Since there is no external RTC connected, the timestamp of the module/device may experience inaccuracies or anomalies, potentially reverting to an earlier time, when it encounters an environment devoid of GNSS signal coverage. These issues are resolved once the module/device exits the affected area.

2.5. Temperature Compensation

Since the IMU is a temperature-sensitive sensor, the temperature compensation feature should be enabled when the operating environment deviates by more than 25 degrees from the normal temperature. Requirements to be met before using this feature:

- Run the LC29H (BA, CA)/QLM29H series in open air for approximately 30 minutes, either stationary or moving.
- Ensure LC29H (BA, CA)/QLM29H series experience a complete temperature cycle exceeding a 25-degree change, such as transitioning from a low to a high temperature, or vice versa.

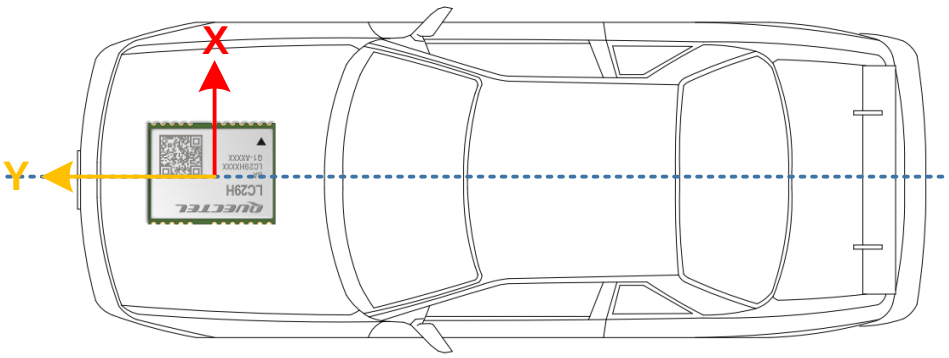
For optimal performance, maintain the installation position of LC29H (BA, CA)/QLM29H series throughout tests after the initial setup to accumulate data on various temperature effects. Refer to *Chapter 4.1.13 PQTMCFGIMUTC* to configure the feature.

2.6. Module/Device Mounting

The LC29H (BA), LC29H (CA) and QLM29H series support free mounting with no installation angle requirements. Ensure that the module/device is firmly mounted to the vehicle body to prevent any relative movement. Avoid exposing the vehicle to conditions such as frequent vibration and high temperature.

2.6.1. Mounting on Four-wheel Vehicles

No mounting direction and angle limitations are required for mounting the Quectel LC29H (BA), LC29H (CA), or QLM29H series on a four-wheel vehicle. The reference model is as follows:



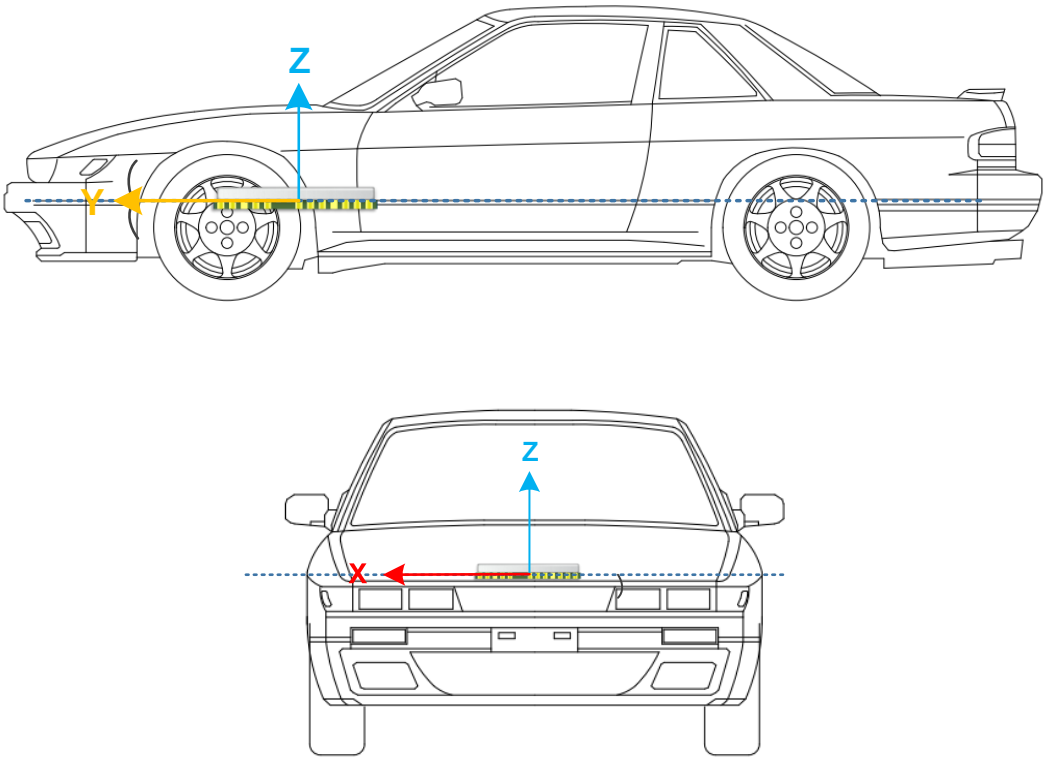
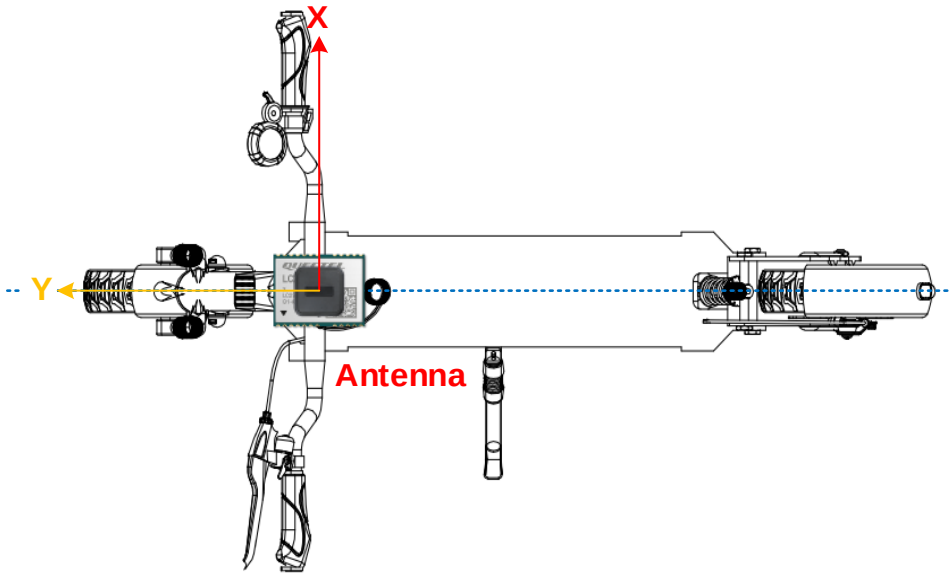


Figure 11: Module Mounting Example (4-wheel Vehicle)

2.6.2. Mounting on Two-wheel Vehicles

The reference model for installing Quectel LC29H (BA) or LC29H (CA) on a two-wheel vehicle is as follows:



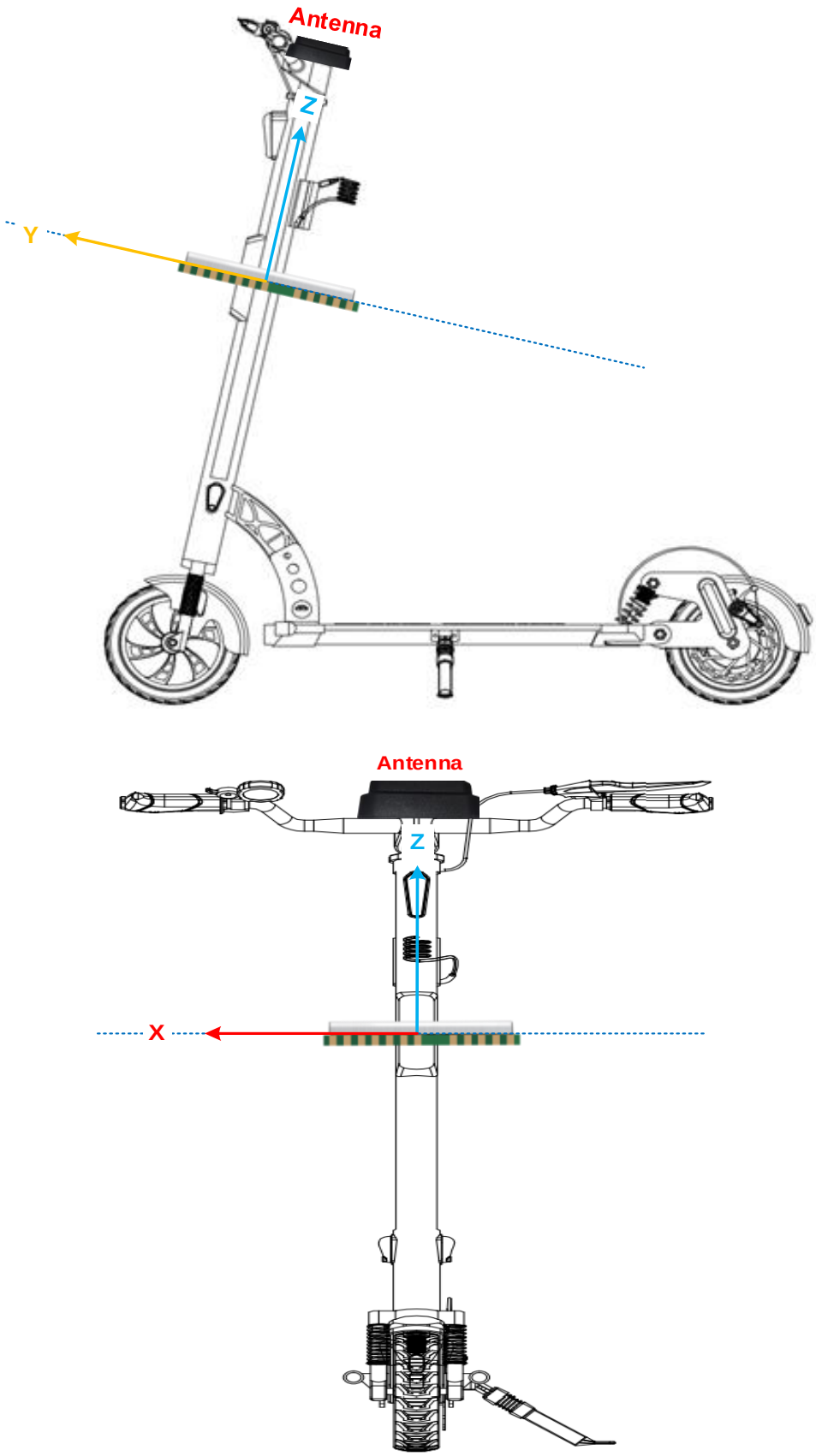


Figure 12: Module Mounting Example (2-wheel Vehicle)

NOTE

1. Firmly affix the device incorporating LC29H (BA, CA)/QLM29H series to vehicle body. Select a structurally sound location that is not prone to flexing (without relative motion). GNSS antenna, LC29H (BA, CA)/QLM29H series, and any supported sensors should be placed on the same body frame.
2. Installations on both two-wheel and four-wheel vehicles require a relatively fixed lever arm displacement between the antenna and LC29H (BA, CA)/QLM29H series. Since the module/device supports free-mounting feature, it is capable of estimating the lever arm automatically. In addition, the module/device also allows for manual configuration of lever arm; for details, see *Chapter 4.1.20 PQTMCFGLA*.
3. If your two-wheeler (e.g., electric scooter) can provide speed data, such as wheel ticks or wheel speeds, and ADR is to be used, ideally the speed data should come from the rear wheel when LC29H (BA, CA) are mounted on the deck, and from the front wheel when they are mounted on the handlebar.
4. When LC29H (BA, CA) are installed on the handlebar of a two-wheeler, the navigation performance will be degraded. The reasons are as follows:
   - Handlebars are typically more prone to movement and vibration than other parts of the vehicle. The lack of stability can cause relative displacement between LC29H (BA, CA) and the wheel providing speed data, which can introduce errors into the navigation calculation.
   - Handlebars are subject to more vibration and can generate noise from various sources. This noise can lead to degraded measurement results.
5. Place the GNSS antenna in an upright position relative to the sky for optimal signal reception and positioning performance.

2.7. DR Calibration

For the DR functional module/device, calibration is a key step in achieving DR extrapolation. The DR trajectory will be accurate only if LC29H (BA, CA)/QLM29H series are accurately calibrated. Consequently, it is necessary to complete the calibration process before using the DR feature.

The DR calibration steps are as listed below:

Step 1: Fix the device incorporating LC29H (BA, CA)/QLM29H series on the vehicle frame firmly. Any displacement, rotation or tilt of the device (incorporating LC29H (BA, CA)/QLM29H series) relative to vehicle plane, however small, may cause performance issues or calibration failures.

Step 2: Calibration should be performed under good GNSS signal and clear sky conditions.

Step 3: Power up LC29H (BA, CA)/QLM29H series, then start the vehicle on a plain surface.

Step 4: Drive at a speed of more than 2 m/s, and perform 3 or 4 turning movements.

LC29H (BA, CA)/QLM29H series will start self-calibration, which will be completed in approximately 3 minutes.

Step 5: The calibration process ends when **<CalState>** of **PQTMDRCAL** message value is 2 (DR is fully calibrated). See *Chapter 4.1.1 PQTMDRCAL* for details about the message.

NOTE

1. If the wheel speed sensor of the vehicle is connected to LC29H (BA, CA)/QLM29H series, make sure that its precision is at least 0.05 m/tick.
2. LC29H (BA, CA)/QLM29H series will exit GNSS + DR or DR only mode in the following scenarios:
   - No speed data injection for 5 seconds in ADR mode;
   - Handling scenario;
   - High-speed elevator scenario;
   - IMU data interruption;
   - Inadequate inertial navigation calibration before entering a tunnel or underground garage;
   - Exceeding the set running time or distance of the receiver in environments with nonexistent GNSS coverage (See *Chapter 4.1.12 PQTMCFGDRRTD*).

3 RTK

3.1. RTCM Input

Quectel QLM29HBAA-GM, QLM29HBAP-GM, LC29H (BA), LC29H (DA) and LC29H (EA) support the RTCM 10403.3 input messages listed in the table below.

Table 1: Supported RTCM Input Messages

Message Type	Message Name
1005	Stationary RTK Reference Station ARP
1006	Stationary RTK Reference Station ARP with Antenna Height
1074	GPS MSM4
1075	GPS MSM5
1077	GPS MSM7
1084	GLONASS MSM4
1085	GLONASS MSM5
1087	GLONASS MSM7
1094	Galileo MSM4
1095	Galileo MSM5
1097	Galileo MSM7
1114	QZSS MSM4
1115	QZSS MSM5
1117	QZSS MSM7

Message Type	Message Name
1124	BDS MSM4
1125	BDS MSM5
1127	BDS MSM7

NOTE

The distance between the rover station and the base station should not exceed 15 km, otherwise the accuracy of RTK cannot be guaranteed.

4 Related Messages

4.1. PQTM Messages

This chapter outlines the Quectel DR and RTK related PQTM (proprietary NMEA) messages supported by the Quectel LC29H (BA), LC29H (CA) and QLM29H series.

Table 2: Error Codes

Field	Format	Unit	Description
<ErrCode>	Numeric	-	Error code. 1 = Invalid parameters 2 = Execution failed 3 = Unsupported command

4.1.1. PQTMDRCAL

Indicates the DR calibration state.

Type:

Output

Synopsis:

```
$PQTMDRCAL,<MsgVer>,<CalState>,<NavType>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<MsgVer>	Numeric	-	Message version. 1 = Version 1 (Always 1 for this version.)
<CalState>	Numeric	-	DR calibration state. 0 = Not calibrated 1 = DR is lightly calibrated 2 = DR is fully calibrated

Field	Format	Unit	Description
			3 = DR is fully calibrated with high-precision heading
<NavType>	Numeric	-	Navigation type. 0 = No position 1 = GNSS only 2 = DR only 3 = Combination (GNSS + DR)

Example:

```
$PQTMDRCAL,1,0,1*5C
```

NOTE

1 For LC29H (BA) and LC29H (CA) with firmware versions dedicated for two-wheel vehicles, this message is output by default. Disabling the output of this message is not supported.

2 For LC29H (BA) and LC29H (CA) with firmware versions dedicated for four-wheel vehicles and QLM29H series, the output of this message should be enabled via *Chapter 4.1.23 PQTMCFGMSGRATE* or *Chapter 4.2.1 PAIR6010: PAIR_CUSTOM_SET_MSG_OUTPUT*.

4.1.2. PQTMIMUTYPE

Outputs the IMU status once after each boot-up.

Type:

Output

Synopsis:

```
$PQTMIMUTYPE,<MsgVer>,<Status>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<MsgVer>	Numeric	-	Message version. 1 = Version 1 (Always 1 for this version.)
<Status>	Numeric	-	IMU initialization status. 0 = IMU initialization failed Others = IMU initialization successful

Example:

```
$PQTMIMUTYPE,1,2*52
```

NOTE

This message is output once after each boot-up by default. For LC29H (BA) and LC29H (CA) with firmware versions dedicated for two-wheel vehicles, disabling the output of this message is not supported.

4.1.3. PQTMVEHMSG

Inputs/outputs vehicle information.

Type:

Input/Output

Synopsis:

```
$PQTMVEHMSG,<MsgVer>,<Timestamp>,<Par1>[,<Par2>,…,<ParN>]*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<MsgVer>	Numeric	-	Message version. 1 = Vehicle speed (in m/s) 2 = Cumulative wheel tick 3 = Speeds of four wheels (in m/s) 4 = Cumulative wheel ticks of four wheels
<Timestamp>	Numeric	Millisecond	Timestamp since power-on. 32-bit unsigned integer. Always 0 when this message is input.
<Par1> to <ParN>	Numeric	-	Vehicle information. This field varies with the message type. See *Chapter 4.1.3.1 If <MsgVer> = 1* to *4.1.3.4 If <MsgVer> = 4* for details.

NOTE

For LC29H (BA) and LC29H (CA) with firmware versions dedicated for two-wheel vehicles, **<MsgVer>** can only be 1 or 2. Contact Quectel Technical Support (support@quectel.com) for details about the firmware versions.

4.1.3.1. If <MsgVer> = 1

Synopsis:

```
$PQTMVEHMSG,1,<Timestamp>,<VehSpeed>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Timestamp>	Numeric	Millisecond	Timestamp since power-on. 32-bit unsigned integer. Always 0 when this message is input.
<VehSpeed>	Numeric	m/s	Speed. Range: -100 to 100.

Result:

Returns the input vehicle speed with timestamp:

```
$PQTMVEHMSG,1,<Timestamp>,<VehSpeed>*<Checksum><CR><LF>
```

Example:

```
$PQTMVEHMSG,1,0,3.6*1C
$PQTMVEHMSG,1,3748292,3.6*1D
```

4.1.3.2. If <MsgVer> = 2

Synopsis:

```
$PQTMVEHMSG,2,<Timestamp>,<WheelTickCNT>,<FWD_Ind>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Timestamp>	Numeric	Millisecond	Timestamp since power-on. 32-bit unsigned integer. Always 0 when this message is input.
<WheelTickCNT>	Numeric	Tick	Cumulative wheel ticks.
<FWD_Ind>	Numeric	-	Forward/backward indicator. 0 = Invalid state 1 = Forward 2 = Backward

Result:

Returns the input cumulative wheel ticks with timestamp:

```
$PQTMVEHMSG,2,<Timestamp>,<WheelTickCNT>,<FWD_Ind>*<Checksum><CR><LF>
```

Example:

```
$PQTMVEHMSG,2,0,100,1*18
$PQTMVEHMSG,2,153954,100,1*27
```

NOTE

When inputting cumulative wheel ticks through UART interface, ensure the input rate is at least 10 Hz.

4.1.3.3. If <MsgVer> = 3

Synopsis:

```
$PQTMVEHMSG,3,<Timestamp>,<LF_Spd>,<RF_Spd>,<LR_Spd>,<RR_Spd>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<TimeStamp>	Numeric	Millisecond	Timestamp since power-on. 32-bit unsigned integer. Always 0 when this message is input.
<LF_Spd>	Numeric	m/s	Left front wheel speed. Range: -100 to 100.
<RF_Spd>	Numeric	m/s	Right front wheel speed. Range: -100 to 100.
<LR_Spd>	Numeric	m/s	Left rear wheel speed. Range: -100 to 100.
<RR_Spd>	Numeric	m/s	Right rear wheel speed. Range: -100 to 100.

Result:

Returns the input speeds of four wheels with timestamp:

```
$PQTMVEHMSG,3,<Timestamp>,<LF_Spd>,<RF_Spd>,<LR_Spd>,<RR_Spd>*<Checksum><CR><LF>
```

Example:

```
$PQTMVEHMSG,3,0,3.6,3.6,3.6,3.6*19
$PQTMVEHMSG,3,3748292,3.6,3.6,3.6,3.6*18
```

4.1.3.4. If <MsgVer> = 4

Synopsis:

```
$PQTMVEHMSG,4,<Timestamp>,<LF_TickCNT>,<RF_TickCNT>,<LR_TickCNT>,<RR_TickCNT><FWD_Ind>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Timestamp>	Numeric	Millisecond	Timestamp since power-on. 32-bit unsigned integer. Always 0 when this message is input.
<LF_TickCNT>	Numeric	Tick	Left front wheel tick count.
<RF_TickCNT>	Numeric	Tick	Right front wheel tick count.
<LR_TickCNT>	Numeric	Tick	Left rear wheel tick count.
<RR_TickCNT>	Numeric	Tick	Right rear wheel tick count.
<FWD_Ind>	Numeric	-	Forward/backward indicator. 0 = Invalid state 1 = Forward 2 = Backward

Result:

Returns the input cumulative wheel ticks of four wheels with timestamp:

```
$PQTMVEHMSG,4,<Timestamp>,<LF_TickCNT>,<RF_TickCNT>,<LR_TickCNT>,<RR_TickCNT><FWD_Ind>*<Checksum><CR><LF>
```

Example:

```
$PQTMVEHMSG,4,0,100,100,100,100,1*03
$PQTMVEHMSG,4,153954,100,100,100,100,1*3C
```

4.1.4. PQTMSAVEPAR

Saves the configurations set via **PQTM** commands or **PAIR6010** into NVM.

Type:

Command

Synopsis:

```
$PQTMSAVEPAR*<Checksum><CR><LF>
```

Parameter:

None

Result:

- If successful, the module/device returns:

```
$PQTMSAVEPAR,OK*<Checksum><CR><LF>
```

- If failed, the module/device returns:

```
$PQTMSAVEPAR,ERROR,<ErrCode>*<Checksum><CR><LF>
```

For details about <**ErrCode**>, see *Table 2: Error Codes*.

Example:

```
$PQTMSAVEPAR*5A
$PQTMSAVEPAR,OK*72
```

4.1.5. PQTMRESTOREPAR

Restore all configurations to their default values. This command takes effect after a reboot.

Type:

Command

Synopsis:

```
$PQTMRESTOREPAR*<Checksum><CR><LF>
```

Parameter:

None

Result:

- If successful, the module/device returns:

```
$PQTMRESTOREPAR,OK*<Checksum><CR><LF>
```

- If failed, the module/device returns:

```
$PQTMRESTOREPAR,ERROR,<ErrCode>*<Checksum><CR><LF>
```

For details about <**ErrCode**>, see *Table 2: Error Codes*.

Example:

```
$PQTMRESTOREPAR*13
$PQTMRESTOREPAR,OK*3B
```

4.1.6. PQTMINS

Outputs navigation results.

Type:

Output

Synopsis:

```
$PQTMINS,<Timestamp>,<SolType>,<Lat>,<Lon>,<Height>,<VEL_N>,<VEL_E>,<VEL_D>,<Roll>,<Pitch>,<Yaw>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Timestamp>	Numeric	Millisecond	Timestamp since power-on. 32-bit unsigned integer.
<SolType>	Numeric	-	Solution type. 0 = DR not ready. Roll and pitch ready. 1 = DR not ready. GNSS, roll, pitch, and relative heading ready. 2 = GNSS + DR mode. DR calibrated. 3 = DR only mode.
<Lat>	Numeric	Degree	Latitude.
<Lon>	Numeric	Degree	Longitude.
<Height>	Numeric	Meter	Height.
<VEL_N>	Numeric	m/s	Northward velocity.
<VEL_E>	Numeric	m/s	Eastward velocity.
<VEL_D>	Numeric	m/s	Downward velocity.
<Roll>	Numeric	Degree	Roll angle. Range: -180.00 to 180.00
<Pitch>	Numeric	Degree	Pitch angle. Range: -90.00 to 90.00

Field	Format	Unit	Description
<Yaw>	Numeric	Degree	Heading angle. Range: 0.00–360.00

Example:

```
$PQTMINS,240951,1,31.82222216,117.11578436,62.555605,-0.004233,0.005535,-0.004011,0.00,0.00,127.41*40
```

NOTE

This message is only supported by LC29H (BA) and LC29H (CA) with firmware versions dedicated for two-wheel vehicles. Contact Quectel Technical Support (support@quectel.com) for details about the firmware versions.

4.1.7. PQTMIMU

Outputs the IMU raw data: acceleration, angular rate, and hardware wheel ticks. These values should match module frame, and see *Figure 7: Module Orientation* for details.

Type:

Output

Synopsis:

```
$PQTMIMU,<Timestamp>,<ACC_X>,<ACC_Y>,<ACC_Z>,<AngRate_X>,<AngRate_Y>,<AngRate_Z>,<WheelTickCNT>,<LastTick_Timestamp>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Timestamp>	Numeric	Millisecond	Timestamp since power-on. 32-bit unsigned integer.
<ACC_X>	Numeric	m/s^2	Acceleration in X-axis direction.
<ACC_Y>	Numeric	m/s^2	Acceleration in Y-axis direction.
<ACC_Z>	Numeric	m/s^2	Acceleration in Z-axis direction.
<AngRate_X>	Numeric	deg/s	Angular rate in X-axis direction.
<AngRate_Y>	Numeric	deg/s	Angular rate in Y-axis direction.

Field	Format	Unit	Description
<AngRate_Z>	Numeric	deg/s	Angular rate in Z-axis direction.
<WheelTickCNT>	Numeric	Tick	Cumulative wheel ticks.
<LastTick_Timestamp>	Numeric	Millisecond	Last tick timestamp.

Example:

```
$PQTMIMU,45454,-1.356730,-0.210568,9.757930,0.564879,0.549612,-0.412209,0,0*77
```

NOTE

This message is only supported by LC29H (BA) and LC29H (CA) with firmware versions dedicated for two-wheel vehicles. Contact Quectel Technical Support (support@quectel.com) for details about the firmware versions.

4.1.8. PQTMGPS

Outputs the position status in GNSS only mode.

Type:

Output

Synopsis:

```
$PQTMGPS,<Timestamp>,<TOW>,<Lat>,<Lon>,<Altitude>,<Speed>,<Heading>,<Accuracy>,<HDOP>,
<PDOP>,<NumSatUsed>,<FixMode>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Timestamp>	Numeric	Millisecond	Timestamp since power-on. 32-bit unsigned integer.
<TOW>	Numeric	Second	Time of week.
<Lat>	Numeric	Degree	Latitude.
<Lon>	Numeric	Degree	Longitude.
<Altitude>	Numeric	Meter	Altitude.
<Speed>	Numeric	m/s	Ground speed (two-dimensional).

Field	Format	Unit	Description
<Heading>	Numeric	Degree	Heading of vehicle (two-dimensional). Range: 0.00–360.00
<Accuracy>	Numeric	Meter	Horizontal accuracy estimate.
<HDOP>	Numeric	-	Horizontal dilution of precision.
<PDOP>	Numeric	-	Position dilution of precision.
<NumSatUsed>	Numeric	-	Number of satellites used in navigation.
<FixMode>	Numeric	-	Fix mode. 0 = No fix 2 = 2D fix 3 = 3D fix (including RTK float or RTK fixed, depending on correction data input)

Example:

```
$PQTMGPS,86139,94183,31.82218794,117.11579022,65.755080,0.027,94.68,2.533952,0.555471,0.886183,29,3*6B
```

NOTE

1. This message is only supported by LC29H (BA) and LC29H (CA) with firmware versions dedicated for two-wheel vehicles. Contact Quectel Technical Support (support@quectel.com) for details about the firmware versions.
2. If no differential data is available, the **PQTMGPS** output will be the PVT result only, otherwise the **PQTMGPS** output will be the RTK result only.

4.1.9. PQTMCFGEINSMSG

Sets/gets **PQTMINS**, **PQTMIMU** and **PQTMGPS** message configurations.

Type:

Set/get

Synopsis:

```
//Set message configurations:
$PQTMCFGEINSMSG,<Type>,<INS_Enabled>,<IMU_Enabled>,<GPS_Enabled>,<Rate>*<Checksum><CR><LF>
//Get message configurations:
$PQTMCFGEINSMSG,<Type>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Type>	Numeric	-	Set/get message configurations. 0 = Get 1 = Set
<INS_Enabled>	Numeric	-	Enable/disable the output of **PQTMINS** message. 0 = Disable 1 = Enable
<IMU_Enabled>	Numeric	-	Enable/disable the output of **PQTMIMU** message. 0 = Disable 1 = Enable
<GPS_Enabled>	Numeric	-	Enable/disable the output of **PQTMGPS** message. 0 = Disable 1 = Enable
<Rate>	Numeric	Hz	Set the output rate of **PQTMINS**, **PQTMIMU** or **PQTMGPS** message. It can be set to 0, 1, 2, 4, 5, 10, 20, 50, or 100. For **PQTMGPS**, the output rate is fixed at 1 Hz when **<Rate>** is set to be greater than 1. For **PQTMINS**, when **<Rate>** is set to be greater than 10, the message will be output at 10 Hz. For **PQTMIMU**, the output rate can be set to 0, 1, 10, 20, 50, or 100 Hz.

Result:

- If successful, the module returns:

```
//Set:
$PQTMCFGEINSMSGOK*16
//Get:
$PQTMCFGEINSMSG,<Type>,<INS_Enabled>,<IMU_Enabled>,<GPS_Enabled>,<Rate>*<Checksum>
<CR><LF>
```

- If failed, the module returns:

```
$PQTMCFGEINSMSGERROR*4A
```

Example:

```
//Set message configurations:
$PQTMCFGEINSMSG,1,1,1,1,10*3F
$PQTMCFGEINSMSGOK*16

//Get message configurations:
```

$PQTMCFGEINSMSG,0*0E
$PQTMEINSMSG,0,1,1,1,10*3E

NOTE

1. Send **$PQTMSAVEPAR*5A** and reset the module for **PQTMCFGEINSMSG** to take effect.
2. This command is only supported by LC29H (BA) and LC29H (CA) with firmware versions dedicated for two-wheel vehicles. Contact Quectel Technical Support (support@quectel.com) for details about the firmware versions.
3. Note the following:
   - If you enable the output of **PQTMINS**, **PQTMIMU**, and **PQTMGPS** messages using the **PQTMCFGEINSMSG** command, and then try to set the position fix rate using either **PQTMCFGFIXRATE** or **PAIR050**, or enable the output of **PQTMDRPVA**, **PQTMPVT**, and **PQTMSENMSG** (**<MsgVer>** = 4), an error will be returned.
   - Similarly, setting the position fix rate using **PQTMCFGFIXRATE** or **PAIR050**, or enabling the output of **PQTMDRPVA**, **PQTMPVT**, and **PQTMSENMSG** (**<MsgVer>** = 4) first, and then using the **PQTMCFGEINSMSG** command to enable the output of **PQTMINS**, **PQTMIMU**, and **PQTMGPS** messages, also leads to an error. When disabling **PQTMINS**, **PQTMIMU** and **PQTMGPS** messages via the **PQTMCFGEINSMSG** command, the parameter **<Rate>** also needs to be set to 0, otherwise the attempting to set position fix rate or enable the output of **PQTMDRPVA**, **PQTMPVT**, and **PQTMSENMSG** (**<MsgVer>** = 4) will still return an error.
   - For deatials about **PQTMCFGFIXRATE**, **PAIR050**, and **PQTMPVT**, see *document [2] protocal specification*. Contact Quectel Technical Support (support@quectel.com) for more information.
4. If the default value is not given for any parameter in a Set command, you can query it with the corresponding Get command provided that the default setting has not been changed by the Set command. If the default setting had been changed by the Set command, contact Quectel Technical Support (support@quectel.com) to get the default setting, if necessary.

4.1.10. PQTMVEHMOT

Outputs vehicle motion information after DR calibration.

Type:

Output

Synopsis:

$PQTMVEHMOT,<MsgVer>,<Par1>[,<Par2>,…,<ParN>]*<Checksum><CR><LF>

Parameter:

Field	Format	Unit	Description
<MsgVer>	Numeric	-	Message version. 1 = Version 1 2 = Version 2
<Par1> to <ParN>	Numeric	-	Vehicle motion information. This field varies with the message type. See *Chapters 4.1.10.1 If <MsgVer> = 1* and *4.1.10.2 If <MsgVer> = 2* for details.

NOTE

This message is only supported by LC29H (BA) and LC29H (CA) with firmware versions dedicated for four-wheel vehicles and QLM29H series. Contact Quectel Technical Support (support@quectel.com) for details about the firmware versions.

4.1.10.1. If <MsgVer> = 1

Synopsis:

```
$PQTMVEHMOT,1,<PeakAcceleration>,<PeakAngularRate>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<PeakAcceleration>	Numeric	m/s^2	Peak acceleration of vehicle.
<PeakAngularRate>	Numeric	deg/s	Peak angular rate of vehicle.

Example:

```
$PQTMVEHMOT,1,0.288124,0.159930*0A
```

4.1.10.2. If <MsgVer> = 2

Synopsis:

```
$PQTMVEHMOT,2,<UTC>,<VehType>,<MotState>,<AccStatus>,<TurningStatus>,<Res0>,<Res1>,<Res2>,<Res3>,<Res4>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<UTC>	hhmmss.sss	-	UTC time.
<VehType>	Numeric	-	Vehicle type. 1 = General-purpose vehicle 2 = Small vehicle 3 = Medium vehicle 4 = Large vehicle
<MotState>	Numeric	-	Motion state of vehicle. 1 = Stopped 2 = Moving Null if invalid (when motion detection algorithm is not available or this feature is disabled).
<AccStatus>	Numeric	-	Vehicle acceleration status. 1 = None (No accelerated motion detected) 2 = Normal acceleration 3 = Rapid acceleration 4 = Normal deceleration 5 = Rapid deceleration Null if invalid (when motion detection algorithm is not available or this feature is disabled).
<TurningStatus>	Numeric	-	Vehicle turning status. 1 = None (No turning motion detected) 2 = Normal turn left 3 = Sharp turn left 4 = Normal turn right 5 = Sharp turn right Null if invalid (when motion detection algorithm is not available or this feature is disabled).
<Res0>	Numeric	-	Reserved. Always null.
<Res1>	Numeric	-	Reserved. Always null.
<Res2>	Numeric	-	Reserved. Always null.
<Res3>	Numeric	-	Reserved. Always null.
<Res4>	Numeric	-	Reserved. Always null.

Example:

```
$PQTMVEHMOT,2,204159.000,1,2,1,1,,,,,*1D
```

4.1.11. PQTMSENMSG

Outputs sensor information.

Type:

Output

Synopsis:

```
$PQTMSENMSG,<MsgVer>,<TimeStamp>,<Par1>[,<Par2>,…,<ParN>]*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<MsgVer>	Numeric	-	Message version. 2 = IMU sensor data matching module reference frame without any filtering. For details on the frame, refer to *Figure 7: Module Orientation* or *Figure 8: QLM29H Series Device Orientation*. 4 = IMU sensor data matching vehicle reference frame. See *Figure 6: Reference Frame* for details.
<TimeStamp>	Numeric	Millisecond	Timestamp since power-on. 32-bit unsigned integer.
<Par1> to <ParN>	Numeric	-	Sensor information. See *Chapters 4.1.11.1 If <MsgVer> = 2* and *4.1.11.2 If <MsgVer> = 4* for details.

NOTE

1. This message is only supported by LC29H (BA), LC29H (CA) and QLM29H series. Contact Quectel Technical Support (support@quectel.com) for details about the firmware versions.
2. For LC29H (BA) and LC29H (CA) with firmware versions dedicated for two-wheel vehicles:
   - If you enable the output of **PQTMINS**, **PQTMIMU**, and **PQTMGPS** messages using the **PQTMCFGEINSMSG** command, and then try to enable the output of **PQTMSENMSG** (**<MsgVer>** = 4), an error will be returned.
   - Similary, enabling the output of **PQTMSENMSG** (**<MsgVer>** = 4) first, and then using the **PQTMCFGEINSMSG** command to enable the output of **PQTMINS**, **PQTMIMU**, and **PQTMGPS** messages, also leads to an error. When disabling **PQTMINS**, **PQTMIMU** and **PQTMGPS** messages via the **PQTMCFGEINSMSG** command, the parameter **<Rate>** also needs to be set to 0, otherwise the attempting to enable the output of **PQTMSENMSG** (**<MsgVer>** = 4) will still return an error.
   - The **PQTMSENMSG** (**<MsgVer>** = 2) message is not supported.
3. The output of **PQTMSENMSG** (**<MsgVer>** = 4) can only be enabled when position fix rate is 10 Hz and the message is output after the DR has completed calibration.

4.1.11.1. If <MsgVer> = 2

Synopsis:

```
$PQTMSENMSG,2,<TimeStamp>,<IMU_Temp>,<IMU_GYRO_X>,<IMU_GYRO_Y>,<IMU_GYRO_Z>,<IMU_ACC_X>,<IMU_ACC_Y>,<IMU_ACC_Z>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<TimeStamp>	Numeric	Millisecond	Timestamp since power-on. 32-bit unsigned integer.
<IMU_Temp>	Numeric	Celsius	IMU temperature.
<IMU_GYRO_X>	Numeric	dps	IMU X-axis gyroscope value.
<IMU_GYRO_Y>	Numeric	dps	IMU Y-axis gyroscope value.
<IMU_GYRO_Z>	Numeric	dps	IMU Z-axis gyroscope value.
<IMU_ACC_X>	Numeric	g	IMU X-axis accelerometer value.
<IMU_ACC_Y>	Numeric	g	IMU Y-axis accelerometer value.
<IMU_ACC_Z>	Numeric	g	IMU Z-axis accelerometer value.

Example:

```
$PQTMSENMSG,2,1000,22.21,0.124521,1.241541,0.912451,0.145785,1.241541,8.954214*2D
```

4.1.11.2. If <MsgVer> = 4

Synopsis:

```
$PQTMSENMSG,4,<TimeStamp>,<IMU_Temp>,<IMU_GYRO_X>,<IMU_GYRO_Y>,<IMU_GYRO_Z>,<IMU_ACC_X>,<IMU_ACC_Y>,<IMU_ACC_Z>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<TimeStamp>	Numeric	Millisecond	Timestamp since power-on. 32-bit unsigned integer.
<IMU_Temp>	Numeric	Celsius	IMU temperature.
<IMU_GYRO_X>	Numeric	dps	IMU X-axis gyroscope value.

Field	Format	Unit	Description
<IMU_GYRO_Y>	Numeric	dps	IMU Y-axis gyroscope value.
<IMU_GYRO_Z>	Numeric	dps	IMU Z-axis gyroscope value.
<IMU_ACC_X>	Numeric	g	IMU X-axis accelerometer value.
<IMU_ACC_Y>	Numeric	g	IMU Y-axis accelerometer value.
<IMU_ACC_Z>	Numeric	g	IMU Z-axis accelerometer value.

Example:

```
$PQTMSENMSG,4,1000,22.21,0.124521,1.241541,0.912451,0.145785,1.241541,8.954214*2B
```

4.1.12. PQTMCFGDRRTD

Sets/gets the DR running time and distance in DR only mode. If the running time or distance of the receiver exceeds the set values, the receiver will exit DR only mode.

Type:

Set/Get

Synopsis:

```
//Set:
$PQTMCFGDRRTD,W,<Time>,<Dist>*<Checksum><CR><LF>
//Get:
$PQTMCFGDRRTD,R*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Time>	Numeric	Second	Configuration of DR running time. 0 = No limitation
<Dist>	Numeric	Meter	Configuration of DR running distance. 0 = No limitation

Result:

- If successful, the module/device returns:

```
//Set:
$PQTMCFGDRRTD,OK*<Checksum><CR><LF>
//Get:
```

```
$PQTMCFGDRRTD,OK,<Time>,<Dist>*<Checksum><CR><LF>
```

- If failed, the module/device returns:

```
$PQTMCFGDRRTD,ERROR,<ErrCode>*<Checksum><CR><LF>
```

For details about <**ErrCode**>, see *Table 2: Error Codes*.

Example:

```
//Set:
$PQTMCFGDRRTD,W,600,10000*72
$PQTMCFGDRRTD,OK*26

//Get:
$PQTMCFGDRRTD,R*70
$PQTMCFGDRRTD,OK,600,10000*21
```

NOTE

This message is only supported by LC29H (BA), LC29H (CA) and QLM29H series. Contact Quectel Technical Support (support@quectel.com) for details about the firmware versions.

4.1.13. PQTMCFGIMUTC

Sets/gets the IMU temperature compensation feature.

Type:

Set/Get

Synopsis:

```
//Set:
$PQTMCFGIMUTC,W,<State>*<Checksum><CR><LF>
//Get:
$PQTMCFGIMUTC,R*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<State>	Numeric	-	State of IMU temperature compensation feature. 0 = Disabled 1 = Enabled

Result:

- If successful, the module/device returns:

```
//Set:
$PQTMCFGIMUTC,OK*<Checksum><CR><LF>
//Get:
$PQTMCFGIMUTC,OK,<State>*<Checksum><CR><LF>
```

- If failed, the module/device returns:

```
$PQTMCFGIMUTC,ERROR,<ErrCode>*<Checksum><CR><LF>
```

For details about <**ErrCode**>, see *Table 2: Error Codes*.

Example:

```
//Enable temperature compensation feature.
$PQTMCFGIMUTC,W,1*7A
$PQTMCFGIMUTC,OK*34

//Query temperature compensation state.
$PQTMCFGIMUTC,R*62
$PQTMCFGIMUTC,OK,1*29
```

NOTE

This message is only supported by LC29H (BA) and LC29H (CA) with firmware versions dedicated for four-wheel vehicles and QLM29H series. Contact Quectel Technical Support (support@quectel.com) for details about the firmware versions.

4.1.14. PQTMDRPVA

Outputs the DR position, velocity and attitude.

Type:

Output

Synopsis:

```
$PQTMDRPVA,<MsgVer>,<Timestamp>,<Time>,<SolType>,<Lat>,<Lon>,<Alt>,<Sep>,<VelN>,<VelE>,<VelD>,<Spd>,<Roll>,<Pitch><Heading>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<MsgVer>	Numeric	-	Message version. 1 = Version 1 (Always 1 for this version.)
<Timestamp>	Numeric	Millisecond	Milliseconds since module turn-on. 32-bit unsigned integer.
<Time>	hhmmss.sss	-	UTC time. hh: Hours (00–23) mm: Minutes (00–59) ss: Seconds (00–59) sss: Decimal fraction of seconds
<SolType>	Numeric	-	Solution type. 0 = No fix 1 = GNSS only 2 = Combination (GNSS + DR) 3 = DR only
<Lat>	Numeric	Degree	Latitude. Note that this field is empty in case of an invalid value.
<Lon>	Numeric	Degree	Longitude. Note that this field is empty in case of an invalid value.
<Alt>	Numeric	Meter	Altitude above mean-sea-level. Note that this field is empty in case of an invalid value.
<Sep>	Numeric	Meter	Geoidal separation (the difference between the WGS84 earth ellipsoid surface and the mean-sea-level surface). Note that this field is empty in case of an invalid value.
<VelN>	Numeric	m/s	North velocity. Note that this field is empty in case of an invalid value.
<VelE>	Numeric	m/s	East velocity. Note that this field is empty in case of an invalid value.
<VelD>	Numeric	m/s	Down velocity. Note that this field is empty in case of an invalid value.
<Spd>	Numeric	m/s	Ground speed. Note that this field is empty in case of an invalid value.
<Roll>	Numeric	Degree	Roll angle. Note that this field is empty in case of an invalid value. Range: -180.000000 to 180.000000
<Pitch>	Numeric	Degree	Pitch angle. Note that this field is empty in case of an invalid value. Range: -90.000000 to 90.000000
<Heading>	Numeric	Degree	Heading.

Field	Format	Unit	Description
			Note that this field is empty in case of an invalid value. Range: 0.000000–360.000000

Example:

```
//No fix.
$PQTMDRPVA,1,1000,163355.000,0,,,,,,,,,,*7C

//GNSS + DR fix.
$PQTMDRPVA,1,75000,083737.000,2,31.12738291,117.26372910,34.212,5.267,3.212,2.928,0.238,4.346,0.392663,1.300793,0.030088*5E
```

NOTE

1. This message is only supported by LC29H (BA), LC29H (CA) and QLM29H series. Contact Quectel Technical Support (support@quectel.com) for details about the firmware versions.
2. For LC29H (BA) and LC29H (CA) with firmware versions dedicated for two-wheel vehicles:
   - If you enable the output of **PQTMINS**, **PQTMIMU**, and **PQTMGPS** messages using the **PQTMCFGEINSMSG** command, and then try to enable the output of **PQTMDRPVA**, an error will be returned.
   - Similarly, enabling the output of **PQTMDRPVA** first, and then using the **PQTMCFGEINSMSG** command to enable the output of **PQTMINS**, **PQTMIMU**, and **PQTMGPS** messages, also leads to an error. When disabling **PQTMINS**, **PQTMIMU** and **PQTMGPS** messages via the **PQTMCFGEINSMSG** command, the parameter **<Rate>** also needs to be set to 0, otherwise the attempting to enable the output of **PQTMDRPVA** will still return an error.

4.1.15. PQTMCFGDRHOT

Sets/gets the DR hot start feature.

Type:

Set/Get

Synopsis:

```
//Set:
$PQTMCFGDRHOT,W,<Mode>*<Checksum><CR><LF>
//Get:
$PQTMCFGDRHOT,R*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Mode>	Numeric	-	Enable/disable the DR hot start mode. 0 = Disable DR hot start. DR needs to be recalibrated after each power-up. 1 = Enable DR hot start. DR does not need to be recalibrated after each power-up, and the module outputs position information immediately after power-up in the case of no signal. 2 = Enable DR hot start. DR does not need to be recalibrated after each power-up, but the module does not output position information after power-up in the case of no signal.

Result:

- If successful, the module/device returns:

```
//Set:
$PQTMCFGDRHOT,OK*<Checksum><CR><LF>
//Get:
$PQTMCFGDRHOT,OK,<Mode>*<Checksum><CR><LF>
```

- If failed, the module/device returns:

```
$PQTMCFGDRHOT,ERROR,<ErrCode>*<Checksum><CR><LF>
```

For details about <**ErrCode**>, see *Table 2: Error Codes*.

Example:

```
//Set:
$PQTMCFGDRHOT,W,1*79
$PQTMCFGDRHOT,OK*37

//Get:
$PQTMCFGDRHOT,R*61
$PQTMCFGDRHOT,OK,1*2A
```

NOTE

This message is only supported by LC29H (BA) and LC29H (CA) with firmware versions dedicated for four-wheel vehicles and QLM29H series. Contact Quectel Technical Support (support@quectel.com) for details about the firmware versions.

4.1.16. PQTMCFGDR

Sets/gets the DR state.

Type:

Set/Get

Synopsis:

```
//Set:
$PQTMCFGDR,W,<State>*<Checksum><CR><LF>
//Get:
$PQTMCFGDR,R*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<State>	Numeric	-	Enable/disable the DR feature. 0 = Disable DR feature 1 = Enable DR feature

Result:

- If successful, the module/device returns:

```
//Set:
$PQTMCFGDR,OK*<Checksum><CR><LF>
//Get:
$PQTMCFGDR,OK,<State>*<Checksum><CR><LF>
```

- If failed, the module/device returns:

```
$PQTMCFGDR,ERROR,<ErrCode>*<Checksum><CR><LF>
```

For details about <**ErrCode**>, see *Table 2: Error Codes*.

Example:

```
//Set:
$PQTMCFGDR,W,1*2A
$PQTMCFGDR,OK*64

//Get:
$PQTMCFGDR,R*32
```

```
$PQTMCFGDR,OK,1*79
```

NOTE

This message is only supported by LC29H (BA) and LC29H (CA) with firmware versions dedicated for four-wheel vehicles and QLM29H series. Contact Quectel Technical Support (support@quectel.com) for details about the firmware versions.

4.1.17. PQTMDRSAVE

Saves DR calibration data, including current DR position, velocity, and attitude.

Type:

Command

Synopsis:

```
$PQTMDRSAVE*<Checksum><CR><LF>
```

Parameter:

None

Result:

- If successful, the module/device returns:

```
$PQTMDRSAVE,OK*<Checksum><CR><LF>
```

- If failed, the module/device returns:

```
$PQTMDRSAVE,ERROR,<ErrCode>*<Checksum><CR><LF>
```

For details about **<ErrCode>**, see *Table 2: Error Codes*.

Example:

```
$PQTMDRSAVE*0F
$PQTMDRSAVE,OK*27
```

NOTE

This message is only supported by LC29H (BA) and LC29H (CA) with firmware versions dedicated for four-wheel vehicles and QLM29H series. Contact Quectel Technical Support (support@quectel.com)

for details about the firmware versions.

4.1.18. PQTMDRCLR

Clears DR calibration data, including current DR position, velocity, and attitude.

Type:

Command

Synopsis:

```
$PQTMDRCLR*<Checksum><CR><LF>
```

Parameter:

None

Result:

- If successful, the module/device returns:

```
$PQTMDRCLR,OK*<Checksum><CR><LF>
```

- If failed, the module/device returns:

```
$PQTMDRCLR,ERROR,<ErrCode>*<Checksum><CR><LF>
```

For details about **<ErrCode>**, see *Table 2: Error Codes*.

Example:

```
$PQTMDRCLR*53
$PQTMDRCLR,OK*7B
```

4.1.19. PQTMCFGRTK

Sets/gets the RTK mode configuration.

Type:

Set/Get

Synopsis:

```
//Set:
```

```
$PQTMCFGRTK,W,<DiffMode>,<RelMode>*<Checksum><CR><LF>
//Get:
$PQTMCFGRTK,R*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<DiffMode>	Numeric	-	Differential mode. 0 = Disable RTK/RTD feature, without using differential data. 1 = Auto mode. The GNSS system automatically switches between RTK only mode and RTD only mode. 2 = RTD only mode, using pseudoranges only.
<RelMode>	Numeric	-	Absolute/relative mode. 1 = Absolute mode. In this mode, the module will check the accuracy of the base station location. If the base station location deviation exceeds 10 m, the module will not be able to stably maintain the RTK fixed mode or even fail to get RTK fixed. 2 = Relative mode. In this mode, the module will ignore the accuracy of the base station location and can use the RTK function normally. Note that this field takes effect only when the differential mode is RTK only mode (in auto mode configured by **<DiffMode>**).

Result:

- If successful, the module/device returns:

```
//Response to Set command:
$PQTMCFGRTK,OK*<Checksum><CR><LF>
//Response to Get command:
$PQTMCFGRTK,OK,<DiffMode>,<RelMode>*<Checksum><CR><LF>
```

- If failed, the module/device returns:

```
$PQTMCFGRTK,ERROR,<ErrCode>*<Checksum><CR><LF>
```

For details about **<ErrCode>**, see *Table 2: Error Codes*.

Example:

```
//Set:
$PQTMCFGRTK,W,1,1*6C
$PQTMCFGRTK,OK*3F
```

```
//Get:
$PQTMCFGRTK,R*69
$PQTMCFGRTK,OK,1,1*3F
```

NOTE

This message is only supported by QLM29HBAA-GM, QLM29HBAP-GM and LC29H (BA, DA, EA). Contact Quectel Technical Support (support@quectel.com) for details about the firmware versions.

4.1.20. PQTMCFGLA

Sets/gets the lever arm.

Type:

Set/Get

Synopsis:

```
//Set:
$PQTMCFGLA,W,<Type>,<LA_X>,<LA_Y>,<LA_Z>*<Checksum><CR><LF>
//Get:
$PQTMCFGLA,R,<Type>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Type>	Numeric	-	Lever arm type. 1 = IMU to antenna lever arm
<LA_X>	Numeric	m	X axis lever arm.
<LA_Y>	Numeric	m	Y axis lever arm.
<LA_Z>	Numeric	m	Z axis lever arm.

Result:

- If successful, the module/device returns:

```
//Response to Set command:
$PQTMCFGLA,OK*<Checksum><CR><LF>
//Response to Get command:
$PQTMCFGLA,OK,<Type>,<LA_X>,<LA_Y>,<LA_Z>*<Checksum><CR><LF>
```

- If failed, the module/device returns:

```
$PQTMCFGLA,ERROR,<ErrCode>*<Checksum><CR><LF>
```

For details about **<ErrCode>**, see *Table 2: Error Codes*.

Example:

```
//Set:
$PQTMCFGLA,W,1,0.212,0.514,0.113*31
$PQTMCFGLA,OK*7F

//Get:
$PQTMCFGLA,R,1*34
$PQTMCFGLA,OK,1,0.212,0.514,0.113*62
```

NOTE

This message is only supported by QLM29H series, LC29H (BA) and LC29H (CA). Contact Quectel Technical Support (support@quectel.com) for details about the firmware versions.

4.1.21. PQTMCFGLAM

Sets/gets lever arm mode.

Type:

Set/Get

Synopsis:

```
//Set:
$PQTMCFGLAM,W,<Mode>*<Checksum><CR><LF>
//Get:
$PQTMCFGLAM,R*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Mode>	Numeric	-	Lever arm mode. 1 = Automatic estimation mode. The algorithm automatically estimates the lever arm information and never uses the lever arm configured by **PQTMCFGLA** command.

Field	Format	Unit	Description
			2 = Strict mode. The algorithm uses the lever arm configured by **PQTMCFGLA** command. 3 = Smart mode. The algorithm uses the lever arm configured by the **PQTMCFGLA** command and intelligently adjusts the lever arm configuration if discrepancies are detected.

Result:

- If successful, the module/device returns:

```
//Response to Set command:
$PQTMCFGLAM,OK*<Checksum><CR><LF>
//Response to Get command:
$PQTMCFGLAM,OK,<Mode>*<Checksum><CR><LF>
```

- If failed, the module/device returns:

```
$PQTMCFGLAM,ERROR,<ErrCode>*<Checksum><CR><LF>
```

For details about **<ErrCode>**, see *Table 2: Error Codes*.

Example:

```
//Set:
$PQTMCFGLAM,W,2*7F
$PQTMCFGLAM,OK*32

//Get:
$PQTMCFGLAM,R*64
$PQTMCFGLAM,OK,2*2C
```

NOTE

This message is only supported by QLM29H series, LC29H (BA) and LC29H (CA). Contact Quectel Technical Support (support@quectel.com) for details about the firmware versions.

4.1.22. PQTMVEHATT

Outputs vehicle attitude, include roll, pitch and heading angles. Refer to *Figure 6: Reference Frame* for details about vehicle frame.

Type:

Output

Synopsis:

```
$PQTMVEHATT,<MsgVer>,<Timestamp>,<Roll>,<Pitch>,<Heading>,<Acc_Roll>,<Acc_Pitch>,<Acc_Heading>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<MsgVer>	Numeric	-	Message version. 1 = Version 1 (Always 1 for this version.)
<Timestamp>	Numeric	Millisecond	Timestamp since power-on. 32-bit unsigned integer. The field restarts from 0 if overflow occurs.
<Roll>	Numeric	Degree	Vehicle roll. Range: -180.000000 to 180.000000. Note that this field is empty in case of an invalid value.
<Pitch>	Numeric	Degree	Vehicle pitch. Range: -90.000000 to 90.000000. Note that this field is empty in case of an invalid value.
<Heading>	Numeric	Degree	Vehicle heading. Range: 0.000000–360.000000. Note that this field is empty in case of an invalid value.
<Acc_Roll>	Numeric	Degree	Vehicle roll accuracy. Range: 0.000000–360.000000. Note that this field is empty in case of an invalid value.
<Acc_Pitch>	Numeric	Degree	Vehicle pitch accuracy. Range: 0.000000–180.000000. Note that this field is empty in case of an invalid value.
<Acc_Heading>	Numeric	Degree	Vehicle heading accuracy. Range: 0.000000–360.000000. Note that this field is empty in case of an invalid value.

Example:

```
$PQTMVEHATT,1,1000,10.002154,20.235412,160.145185,1.254123,5.451214,5.102154*3D
```

NOTE

This message is only supported by LC29H (BA) and LC29H (CA) with firmware versions dedicated for four-wheel vehicles and QLM29H series. Contact Quectel Technical Support (support@quectel.com) for details about the firmware versions.

4.1.23. PQTMCFGMSGRATE

Sets/gets the message output rate on the current port.

Type:

Set/Get

Synopsis:

```
//Set:
$PQTMCFGMSGRATE,W,<MsgName>,<Rate>[,<MsgVer>]*<Checksum><CR><LF>
//Get:
$PQTMCFGMSGRATE,R,<MsgName>[,<MsgVer>]*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<MsgName>	String	-	Configuration message name. See *Table 3: Supported Messages* for details.
<Rate>	Numeric	-	Message output rate. 0 = Output disabled. N = Output once every N position fix(es). Range of N: see *Table 3: Supported Messages* for details.
<MsgVer>	Numeric	-	Message version. Optional. This field can be omitted when the configuration message is standard NMEA 0183 message.

Result:

- If successful, the module/device returns:

```
//Response to Set command:
$PQTMCFGMSGRATE,OK*<Checksum><CR><LF>
//Response to Get command:
$PQTMCFGMSGRATE,OK,<MsgName>,<Rate>[,<MsgVer>]*<Checksum><CR><LF>
```

- If failed, the module/device returns:

```
$PQTMCFGMSGRATE,ERROR,<ErrCode>*<Checksum><CR><LF>
```

For details about **<ErrCode>**, see *Table 2: Error Codes*.

Example:

```
//Set the output rate of PQTMDRPVA message to once every position fix:
$PQTMCFGMSGRATE,W,PQTMDRPVA,1,1*1F
$PQTMCFGMSGRATE,OK*29

//Get the output rate of PQTMDRPVA message:
$PQTMCFGMSGRATE,R,PQTMDRPVA,1*07
$PQTMCFGMSGRATE,OK,PQTMDRPVA,1,1*4C
```

Table 3: Supported Messages

Message Name	Message Output Rate Range (N)
PQTMVEHMSG	1–20
PQTMVEHMOT	1–20
PQTMSENMSG	10, 20, 50, 100 (in Hz)
PQTMVEHATT	1–20
PQTMDRPVA	1–20
PQTMDRCAL	1–20
PQTMIMUTYPE	1

NOTE

1. If the configuration message is a **PQTM** message, use **<MsgVer>** field to specify the message version, otherwise an error will be returned.
2. The output rate N of **PQTMSENMSG** message means N **PQTMSENMSG** messages are to be output per second.
3. The **PQTMIMUTYPE** message is output once after each boot-up by default. Only LC29H (BA) and LC29H (CA) with firmware versions dedicated for four-wheel vehicles and QLM29H series support configuring the output of this message. For LC29H (BA) and LC29H (CA) with firmware versions dedicated for two-wheel vehicles, disabling the output of this message is not supported.

4.1.24. PQTMCFGSTATICHOLD

Sets/gets the parking position hold feature state.

Type:

Set/Get

Synopsis:

```
//Set:
$PQTMCFGSTATICHOLD,W,<State>*<Checksum><CR><LF>
//Get:
$PQTMCFGSTATICHOLD,R*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<State>	Numeric	-	Enable/disable the parking position hold feature. 0 = Disable 1 = Enable

Result:

- If successful, the module returns:

```
//Response to Set command:
$PQTMCFGSTATICHOLD,OK*<Checksum><CR><LF>
//Response to Get command:
$PQTMCFGSTATICHOLD,OK,<State>*<Checksum><CR><LF>
```

- If failed, the module returns:

```
$PQTMCFGSTATICHOLD,ERROR,<ErrCode>*<Checksum><CR><LF>
```

For details about **<ErrCode>**, see *Table 2: Error Codes*.

Example:

```
//Set:
$PQTMCFGSTATICHOLD,W,1*2B
$PQTMCFGSTATICHOLD,OK*65

//Get:
$PQTMCFGSTATICHOLD,R*33
$PQTMCFGSTATICHOLD,OK,1*78
```

NOTE

This message is only supported by LC29H (BA) and LC29H (CA) with firmware versions dedicated for two-wheel vehicles. Contact Quectel Technical Support (support@quectel.com) for details about the firmware versions.

4.1.25. PQTMCFGVEHMOT

Sets/gets the vehicle motion detection feature.

Type:

Set/Get

Synopsis:

```
//Set:
$PQTMCFGVEHMOT,W,<Mode>[,<VehType>]*<Checksum><CR><LF>
//Get:
$PQTMCFGVEHMOT,R*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Mode>	Numeric	-	Disable/enable vehicle motion detection. 0 = Disable 1 = Enable
<VehType>	Numeric	-	Vehicle type. 1 = General-purpose vehicle 2 = Small vehicle 3 = Medium vehicle 4 = Large vehicle

Result:

- If successful, the module/device returns:

```
//Response to set command:
$PQTMCFGVEHMOT,OK*<Checksum><CR><LF>
//Response to get command:
$PQTMCFGVEHMOT,OK,<Mode>[,<VehType>]*<Checksum><CR><LF>
```

- If failed, the module/device returns:

```
$PQTMCFGVEHMOT,ERROR,<ErrCode>*<Checksum><CR><LF>
```

For details about <ErrCode>, see *Table 2: Error Codes*.

Example:

```
//Set:
$PQTMCFGVEHMOT,W,1,1*2C
$PQTMCFGVEHMOT,OK*7F

//Get:
$PQTMCFGVEHMOT,R*29
```

```
$PQTMCFGVEHMOT,OK,1,1*7F
```

NOTE

1. If the vehicle motion detection feature is to be disabled, or has been disabled, the field **<VehType>** should be omitted in the Set or Get command.
2. This message is only supported by LC29H (BA) and LC29H (CA) with firmware versions dedicated for four-wheel vehicles and QLM29H series. Contact Quectel Technical Support (support@quectel.com) for details about the firmware versions.

4.2. PAIR Messages

This chapter explains DR related PAIR messages (proprietary NMEA messages defined by the chipset supplier). "**P**" means proprietary message, "**AIR**" means the command defined by the chipset supplier.

4.2.1. PAIR6010: PAIR_CUSTOM_SET_MSG_OUTPUT

Enables/disables the output of **PQTMVEHMSG**, **PQTMSENMSG**, **PQTMDRCAL**, **PQTMIMUTYPE** and **PQTMVEHMOT** messages.

Type:

Set

Synopsis:

```
$PAIR6010,<Type>,<Output_Rate>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Type>	Numeric	-	Message type. -1 = Reset output state of all following sentence types to the default value 0 = **PQTMVEHMSG** (Default: disabled) 1 = **PQTMSENMSG** (Default: disabled) 2 = **PQTMDRCAL** (Default: disabled) 3 = **PQTMIMUTYPE** (Default: enabled) 4 = **PQTMVEHMOT** (Default: disabled)
<Output_Rate>	Numeric	-	Message output rate. 0 = Output disabled.

Field	Format	Unit	Description
			N = Output once every N position fix(es). Range of N: see *Table 4: Supported Messages of PAIR6010 and PAIR6011* for details.

Result:

Returns **PAIR001** message. See *document [2] protocol specification* for details.

Example:

```
$PAIR6010,0,1*0C
$PAIR001,6010,0*0C
```

Table 4: Supported Messages of PAIR6010 and PAIR6011

Message Name	Message Output Rate Range (N)
PQTMVEHMSG (**<MsgVer>** = 2)	1–20
PQTMSENMSG (**<MsgVer>** = 4)	10, 20, 50, 100 (in Hz)
PQTMDRCAL	1–20
PQTMIMUTYPE	1
PQTMVEHMOT (**<MsgVer>** = 1)	1–20

NOTE

1. This command is only supported by LC29H (BA) and LC29H (CA) with firmware versions dedicated for four-wheel vehicles and QLM29H series. Contact Quectel Technical Support (support@quectel.com) for details about the firmware versions.
2. Send **$PQTMSAVEPAR*5A** and reset the module/device for **PAIR6010** to take effect.
3. The output rate N of **PQTMVEHMSG**, **PQTMDRCAL** and **PQTMVEHMOT** depends on position fix rate. **PQTMIMUTYPE** is only output once after each boot-up. The output rate N of **PQTMSENMSG** message means N **PQTMSENMSG** messages are to be output per second.
4. It's recommended to use **PQTMCFGMSGRATE** to configure the above-mentioned messages instead of using this command. For details, refer to *Chapter 4.1.23 PQTMCFGMSGRATE*.

4.2.2. PAIR6011: PAIR_CUSTOM_GET_MSG_OUTPUT

Gets whether the output of **PQTMVEHMSG**, **PQTMSENMSG**, **PQTMDRCAL**, **PQTMIMUTYPE** and **PQTMVEHMOT** messages is enabled.

Type:

Get

Synopsis:

```
$PAIR6011,<Type>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Type>	Numeric	-	Message type. 0 = **PQTMVEHMSG** 1 = **PQTMSENMSG** 2 = **PQTMDRCAL** 3 = **PQTMIMUTYPE** 4 = **PQTMVEHMOT**

Result:

Returns **PAIR001** message and query result. See *document [2] protocol specification* for details.

Query result message format:

```
$PAIR6011,<Type>,<Output_Rate >*<Checksum><CR><LF>
```

Parameters included in the result:

Field	Format	Unit	Description
<Type>	Numeric	-	Message type. 0 = **PQTMVEHMSG** 1 = **PQTMSENMSG** 2 = **PQTMDRCAL** 3 = **PQTMIMUTYPE** 4 = **PQTMVEHMOT**
<Output_Rate>	Numeric	-	Message output rate. 0 = Output disabled. N = Output once every N position fix(es). Range of N: see *Table 4: Supported Messages of PAIR6010 and PAIR6011* for details.

Example:

```
$PAIR6011,1*11
$PAIR001,6011,0*0D
$PAIR6011,1,0*0D
```

NOTE

1. This command is only supported by LC29H (BA) and LC29H (CA) with firmware versions dedicated for four-wheel vehicles and QLM29H series. Contact Quectel Technical Support (support@quectel.com) for details about the firmware versions.
2. It's recommended to use **PQTMCFGMSGRATE** to get the enable state of the above-mentioned messages instead of using this command. For details, refer to *Chapter 4.1.23 PQTMCFGMSGRATE*.

5 Operation Guide

This chapter explains how to use DR and RTK features using Quectel GNSS MODULE EVB and LC29H (BA) module as an example.

5.1. DR Operation Guide

Preparation:

- A PC with Windows system (Windows 10 or higher versions).
- A Quectel GNSS MODULE EVB with USB cable.
- A dual-band GNSS antenna under open sky.
- QGNSS tool (V1.9 or higher versions).
- QCOM tool.

NOTE

The DR operation guide in Chapter 5.1 does not apply to QLM29H series.

5.1.1. EVB Top View

EVB top view is shown in the figure below. For more information, please refer to *document [3] EVB user guide*.

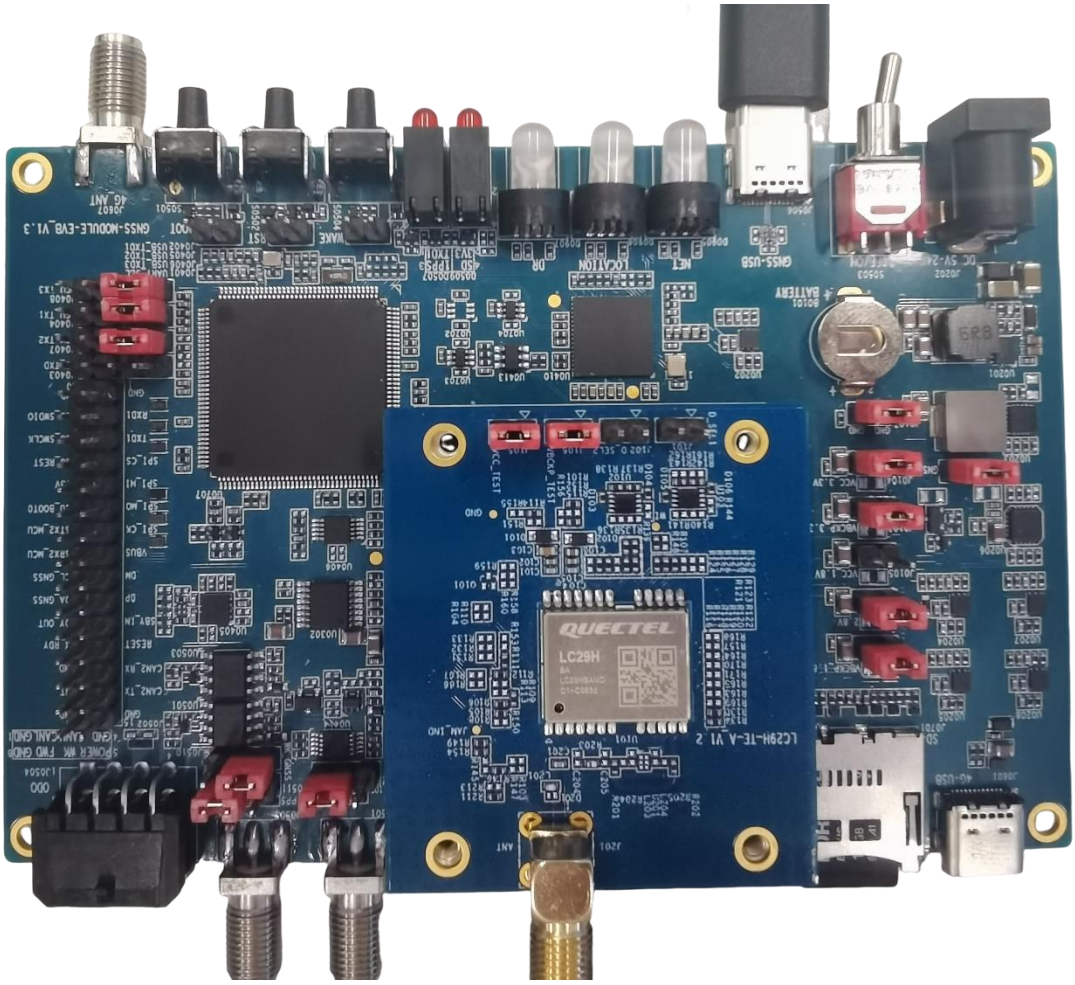


Figure 13: EVB Top View

5.1.2. Injecting Vehicle Speed via CAN/WHEELTICK/UART Interface

If the ADR mode is used, connect the module to your vehicle's speed information source via the corresponding interfaces.

5.1.2.1. Injecting Vehicle Speed via CAN Interface

Vehicle speed is sent to the MCU through the CAN interface on the EVB, and then the MCU injects the converted vehicle speed into the module through UART or WHEELTICK, as shown in *Figure 14: Injecting Vehicle Speed via CAN Interface*. If UART is selected to inject vehicle speed, the WHEELTICK pin of the MCU must be disconnected from the WHEELTICK pin of the GNSS module. Concurrent injection of vehicle speed through the two interfaces is not allowed.

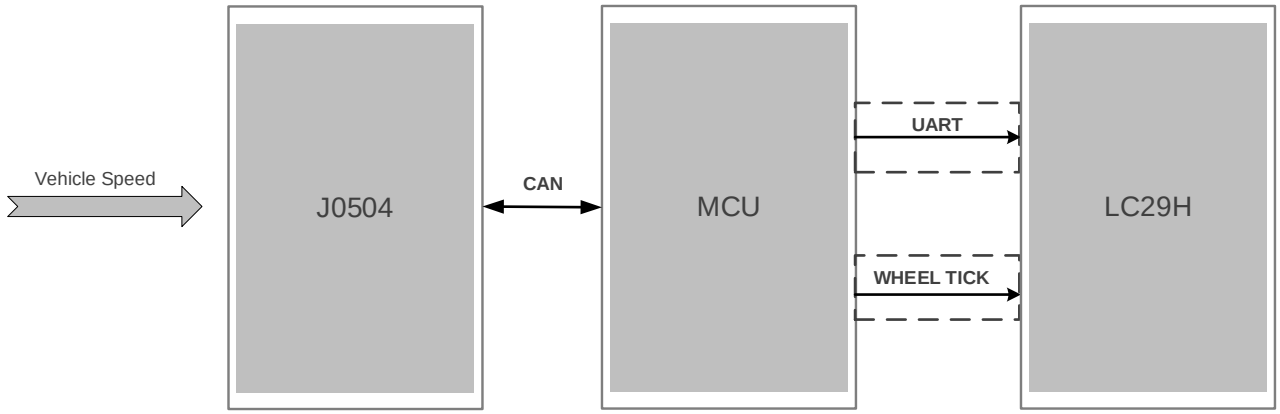


Figure 14: Injecting Vehicle Speed via CAN Interface

To inject vehicle speed via the CAN interface, follow the steps below.

Step 1: Connect the EVB to the PC. After a successful connection, four random but consecutive serial port numbers will appear as shown in *Figure 15: Device Manager*.

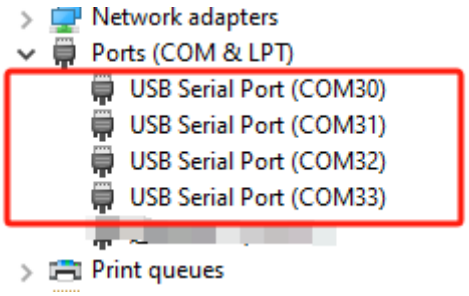


Figure 15: Device Manager

Step 2: Open your serial port tool to select the serial port. Using the QCOM tool as an example, from "**COM Port Setting**" window as shown in *Figure 16: Connection Settings*, select the port with the largest port number among the 4 COM ports and configure the following settings:

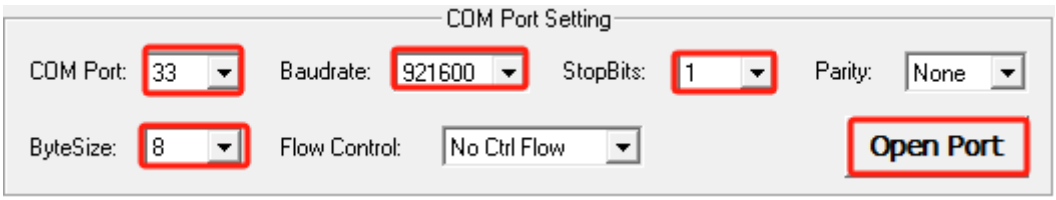


Figure 16: Connection Settings

Step 3: Enter **can --link mcu** and **vehicle --mode 1** commands in the command window and send them to EVB, as shown in *Figure 17: Command Sending*. If a configuration is successful, an "**ok**" will be returned.

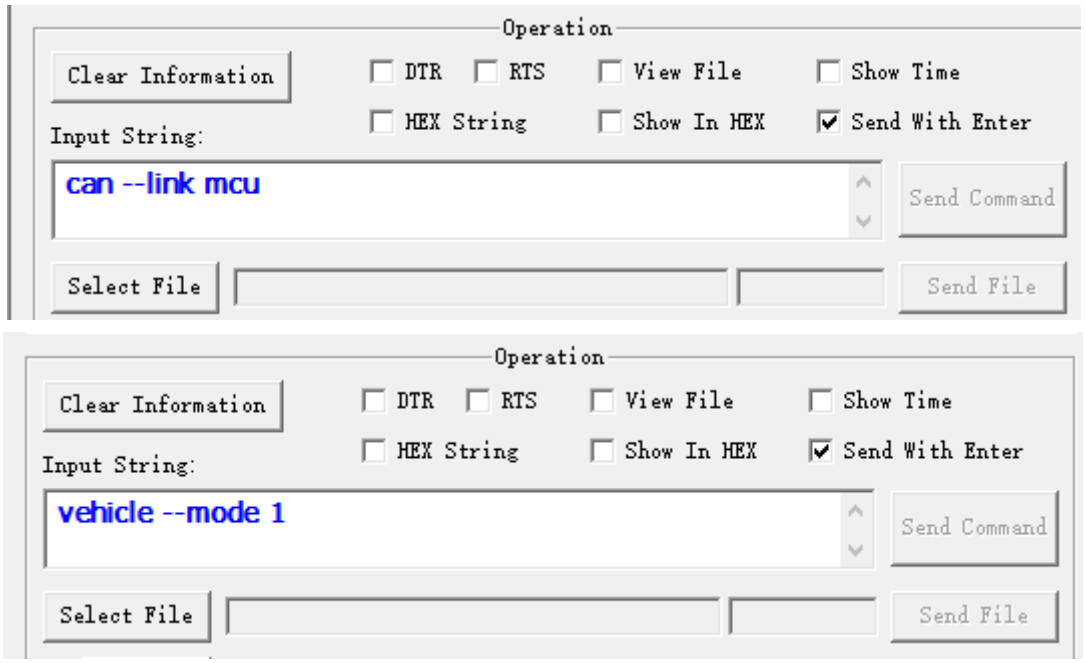


Figure 17: Command Sending

Example:

```
can --link mcu
link mcu ok
vehicle --mode 1
mode 1 ok
```

Step 4: Connect the J0504 of the EVB with the corresponding interface of the vehicle, as shown in *Figure 18: Connecting to EVB CAN Interface*. Regarding J0504 pin assignment, see *Figure 19: J0504 Pin Assignment* and *Table 5: J0504 Pin Description* for details.

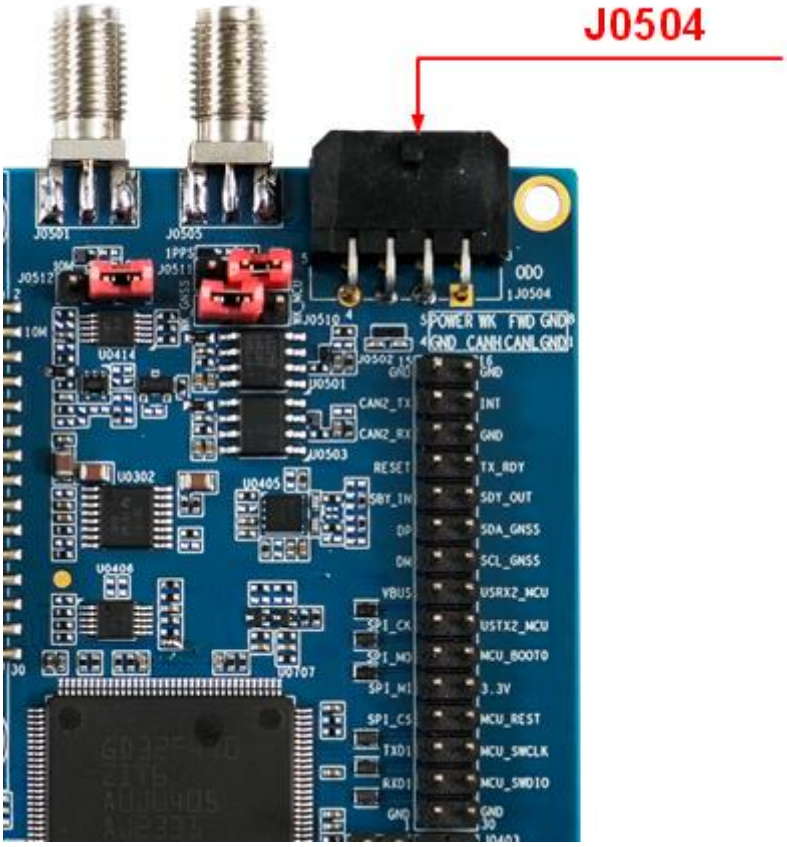


Figure 18: Connecting to EVB CAN Interface

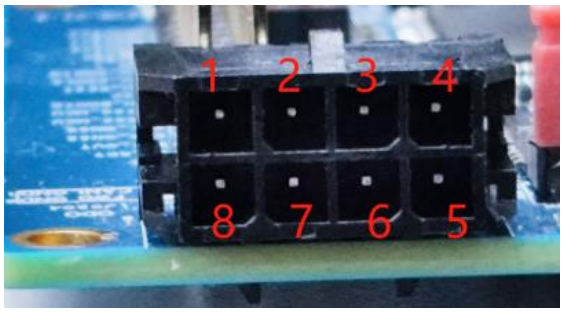


Figure 19: J0504 Pin Assignment

Table 5: J0504 Pin Description

No.	Pin Name	I/O	Description
1	GND	-	Ground
2	CANL	DIO	CAN transceiver low

No.	Pin Name	I/O	Description
3	CANH	DIO	CAN transceiver high
4	GND	-	Ground
5	POWER	PI	Supply power for GNSS module
6	WK	DI	WHEELTICK: Odometer/Wheel-tick pulse input
7	FWD	DI	Forward/Backward status signal input
8	GND	-	Ground

Step 5: Choose either of the following methods to inject the vehicle speed into the module:

- To inject the converted vehicle speed into the module via UART, refer to *Figure 20: Connection Between MCU UART1 and Module UART1* to short-circuit the two pins of J0401.
- To inject the converted vehicle speed into the module via WHEELTICK (i.e., using wheel-tick pulses), refer to *Figure 21: Connection Between MCU WHEELTICK and Module WHEELTICK* to short-circuit the two pins of J0510.

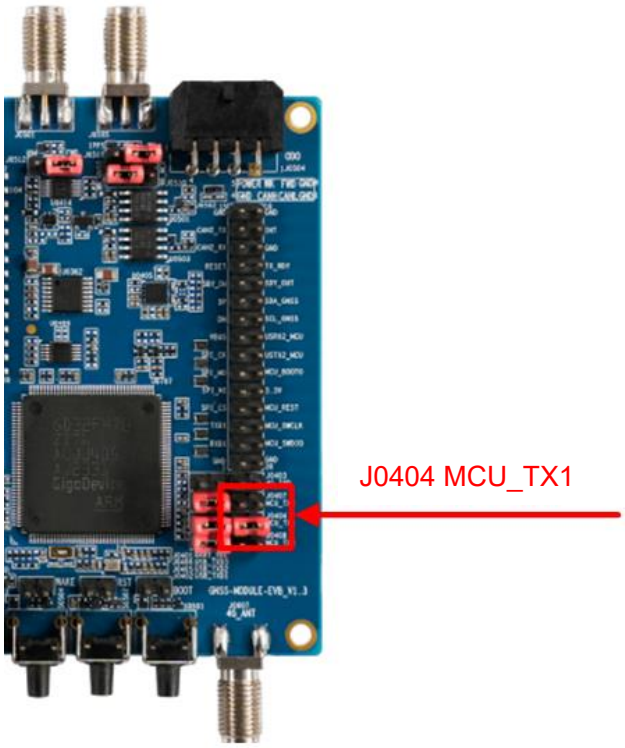


Figure 20: Connection Between MCU UART1 and Module UART1

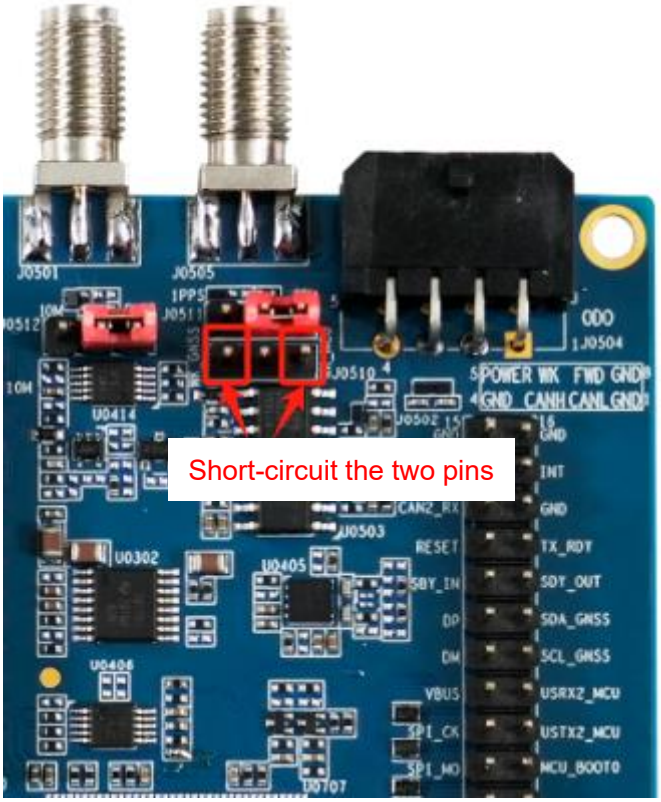


Figure 21: Connection Between MCU WHEELTICK and Module WHEELTICK

NOTE

Currently, the GNSS MODULE EVB supports CAN-injected speed data from vehicle models such as Maxus G20 (by default) and Buick GL8.

5.1.2.2. Injecting Vehicle Speed via WHEELTICK Interface

To inject vehicle speed via the WHEELTICK interface, follow the steps below.

Step 1: Connect the WK pin, the FWD pin, and a GND pin of J0504 to the corresponding interfaces on your device, and short-circuit pins of J0510, J0511, and J0512 as indicated in *Figure 22: Injecting Vehicle Speed via WHEELTICK*.

Step 2: Inject the vehicle speed through J0504.

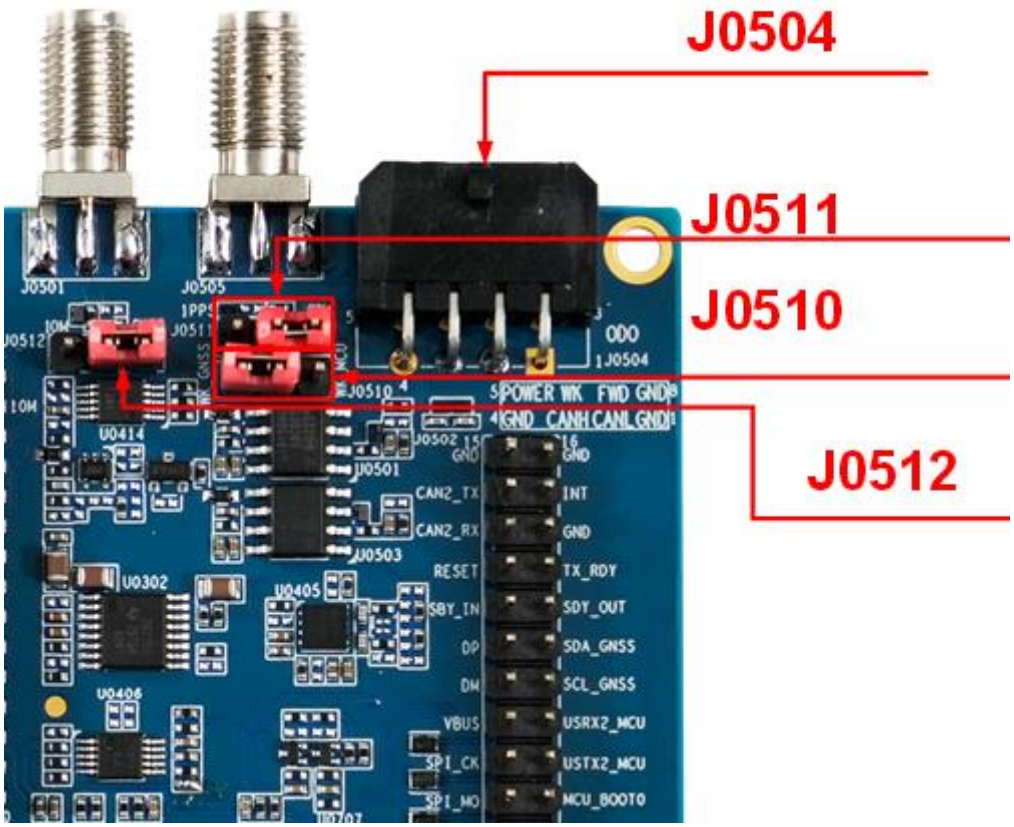


Figure 22: Injecting Vehicle Speed via WHEELTICK

NOTE

On the GNSS MODULE EVB, the polarity of the input signal at the FWD pin of the J0504 interface is inverted upon reaching the module end.

5.1.2.3. Injecting Vehicle Speed via UART Interface

To inject vehicle speed via the UART interface, follow the steps below.

Step 1: Remove the jumper cap of MCU_TX1 (J0404) to disconnect MCU_TX1 of MCU and RXD1 of the GNSS module.

Step 2: Connect TXD1, RXD1, and GND of J0502 to your device, as shown in *Figure 23: Injecting Vehicle Speed via UART*.

Figure 23: Injecting Vehicle Speed via UART

NOTE

1. For optimal performance, ensure that the speed sensor has a resolution of 0.05 m or less per tick when injecting speed data via the WHEELTICK pin.
2. The vehicle direction defaults to forward if the FWD pin is unconnected.
3. The WHEELTICK pin supports a maximum input frequency of 500 Hz. For frequencies exceeding this threshold, it is recommended to inject vehicle speed data via the UART interface instead.

5.1.3. Installation and Calibration

See *Chapter 2.6 Module/Device Mounting* for module/device installation and *Chapter 2.7 DR Calibration* for calibration process. After the module/device is installed and calibrated, you can test DR performance in a weak signal scenario.

5.2. RTK Operation Guide

To implement RTK feature with the LC29H (BA) module, the host must be capable of connecting to an RTK server to receive correction data, which is then injected into the module. This section mainly outlines the process of implementing RTK with QGNSS. It is assumed that you have an RTK server account; both NTRIP and QuecRTK accounts are compatible with the LC29H (BA) module.

Preparation:

- A PC with Windows system (Windows 10 or higher versions).
- A Quectel GNSS MODULE EVB with USB cable.
- A dual-band GNSS antenna under open sky.
- QGNSS tool (V1.9 or higher versions).

Click "**NTRIP**" in the "**Tools**" tab drop-down menu to open "**NTRIP Client**" window. Then, follow the steps provided in *Chapter 5.2.1.1 NTRIP Client* or *Chapter 5.2.1.2 QuecRTK Client*.

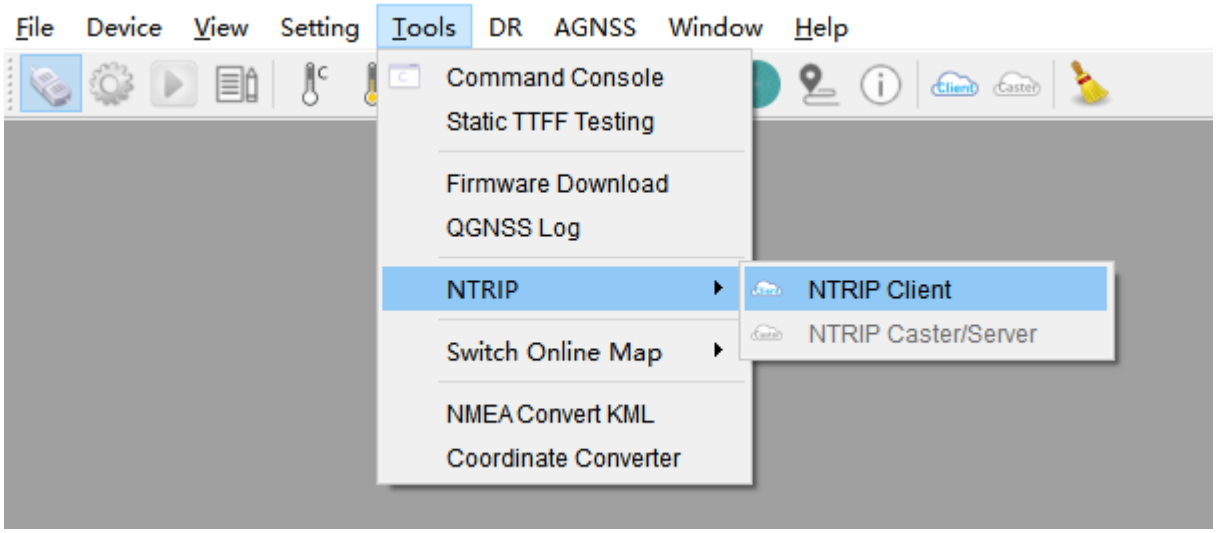


Figure 24: Opening NTRIP Client

5.2.1.1. NTRIP Client

Use an NTRIP (V1.0) Client to establish a connection with a standard NTRIP Caster, and follow the steps below.

Step 1: Enter the Address, Port, Username, and Password. Contact Quectel Technical Support to get the Username and Password if necessary.

Step 2: Click "**Update NTRIP source table**" and wait for the server to return mount point information.

Step 3: Select "**NTRIP mount point**".

Step 4: Enter "**Request Interval**". Default value: 1.

Step 5: To utilize a fixed position, you can select the checkbox labeled "**Use manual position**", and a "**Set Manual Position**" window will appear as shown in *Figure 26: Setting Manual Position*. Once you input the required data, the uploaded position will be the fixed value you have manually set.

If the checkbox remains unchecked, the real-time position in the **GGA** message output by the module will be used.

Step 6: Switch ON the "**Connect to Host**". If the configuration is correct, you can see the messages sent to the RTK server and differential data received in the box on the right as shown in *Figure 27: Connected to Server Successfully*

Step 7: Click "**Monitor**" button to open the "**NTRIP Client: Data Monitor**" window where you can view the differential correction data sent by the server, as shown in *Figure 28: RTK Data Monitor*.

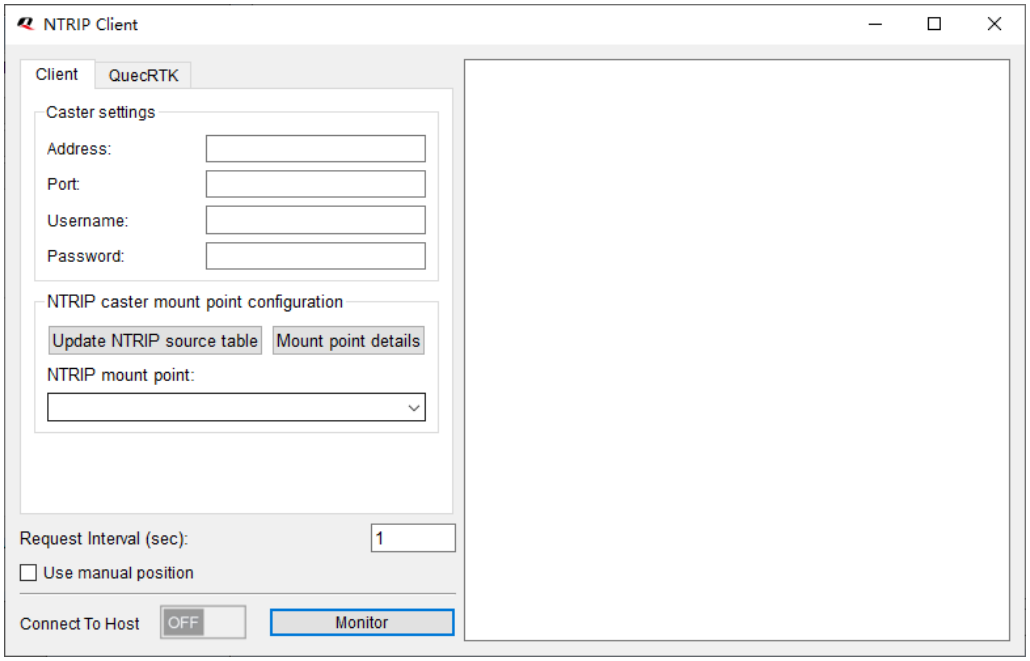


Figure 25: NTRIP Client

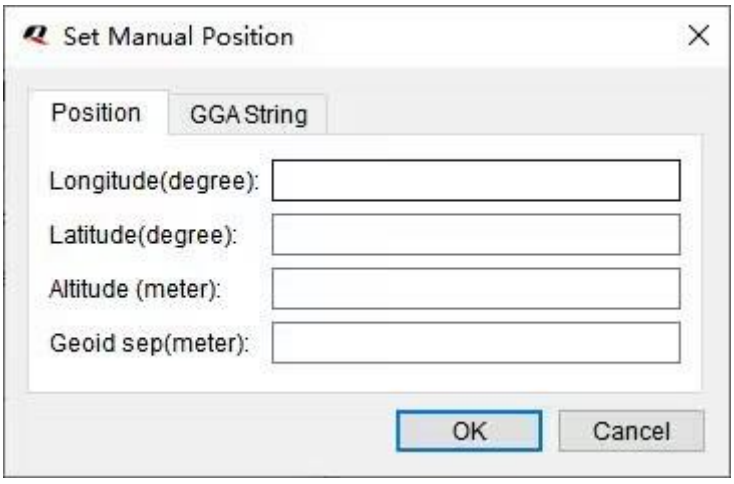


Figure 26: Setting Manual Position

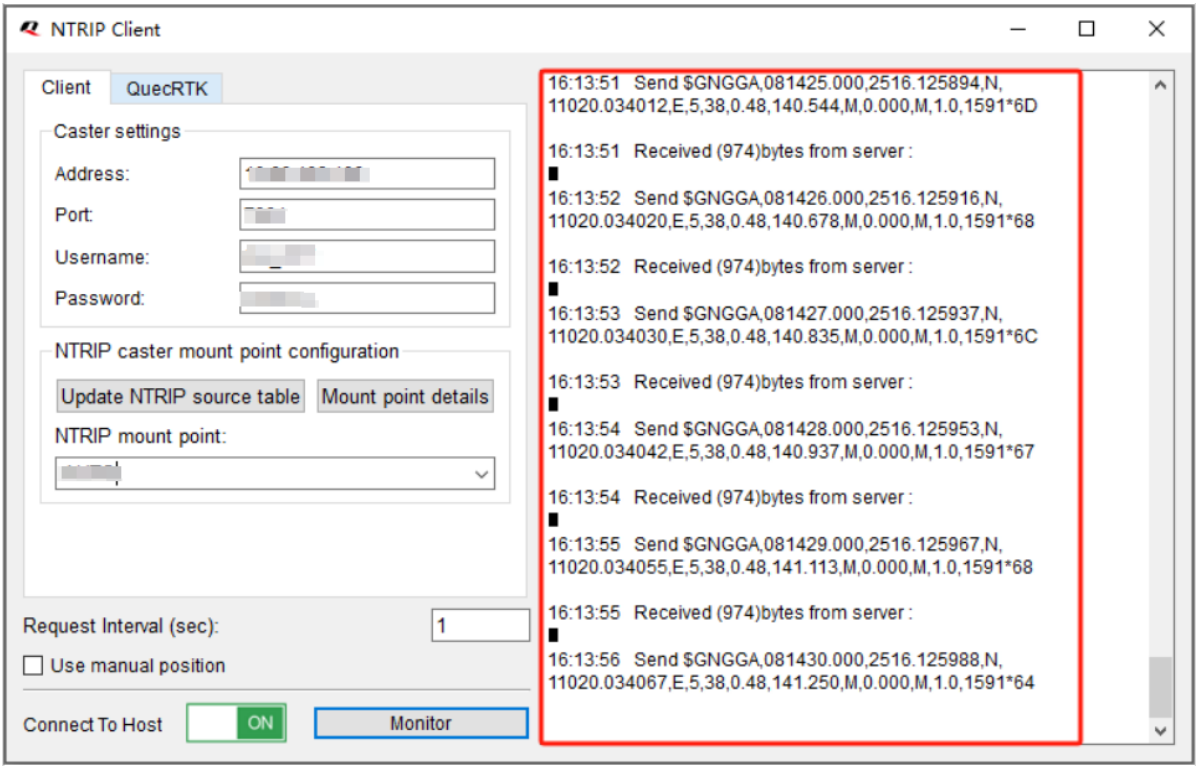


Figure 27: Connected to Server Successfully

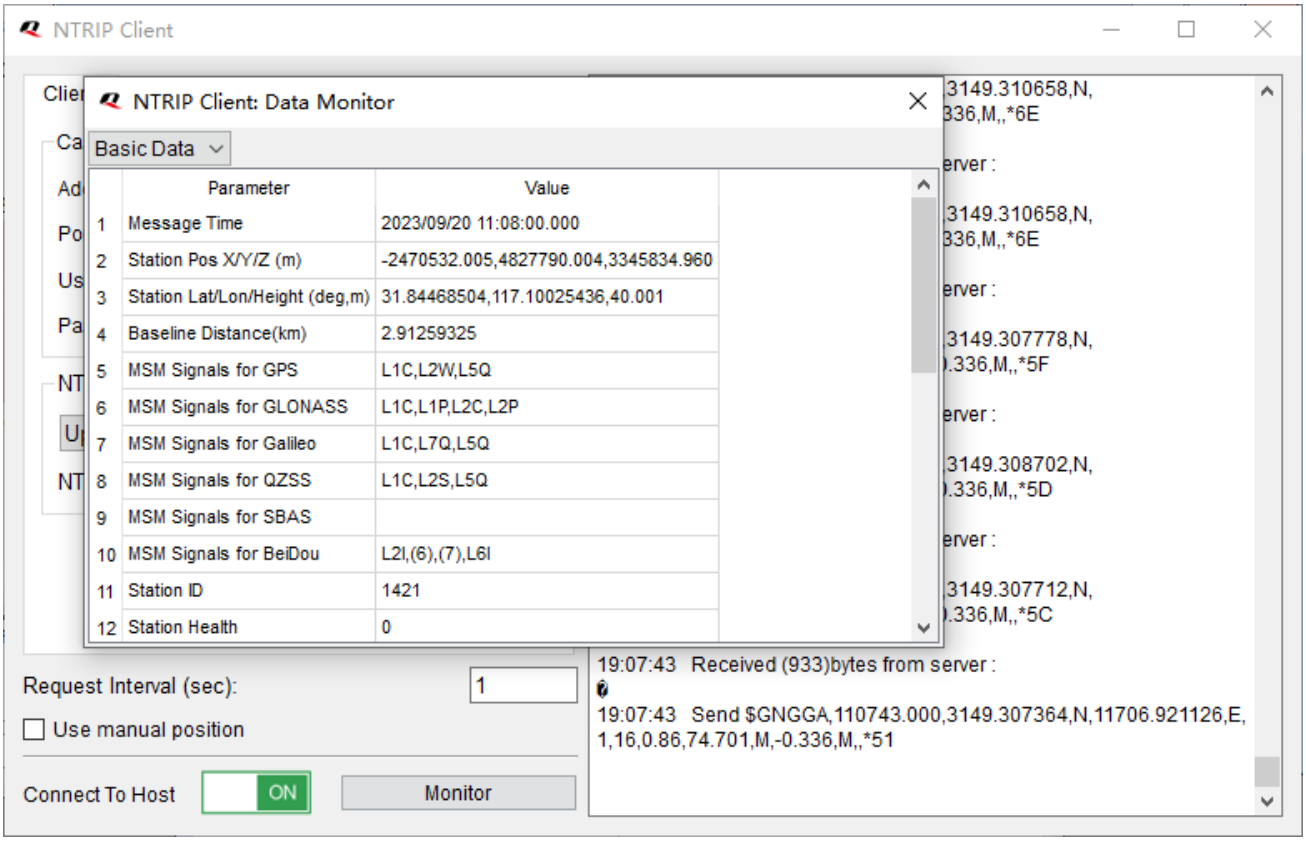


Figure 28: RTK Data Monitor

5.2.1.2. QuecRTK Client

QuecRTK is a high-precision RTK positioning and navigation service provided by Quectel. Based on network RTK technology, QuecRTK supports multiple satellite constellations and frequency bands. The positioning accuracy can reach centimeter-level. By following the steps below, use QuecRTK to improve positioning accuracy of the module:

Step 1: Enter the MCC, Company ID, Device ID, and Licence Key. Contact Quectel Technical Support to get the Device ID and Licence Key if necessary.

Step 2: Enter "**Request Interval**". Default value: 1.

Step 3: To utilize a fixed position, you can select the checkbox labeled "**Use manual position**", and a "**Set Manual Position**" window will appear as shown in *Figure 26: Setting Manual Position*. Once you input the required data, the uploaded position will be the fixed value you have manually set. If the checkbox remains unchecked, the real-time position in the **GGA** message output by the module will be used.

Step 4: Switch ON the "**Connect to Host**". If the configuration is correct, you can see the messages sent to the RTK server and differential data received in the box on the right as shown in *Figure 29: QuecRTK Client*

Step 5: Click "**Monitor**" button to open the "**NTRIP Client: Data Monitor**" window where you can view the differential correction data sent by the server, as shown in *Figure 30: RTK Data Monitor*.

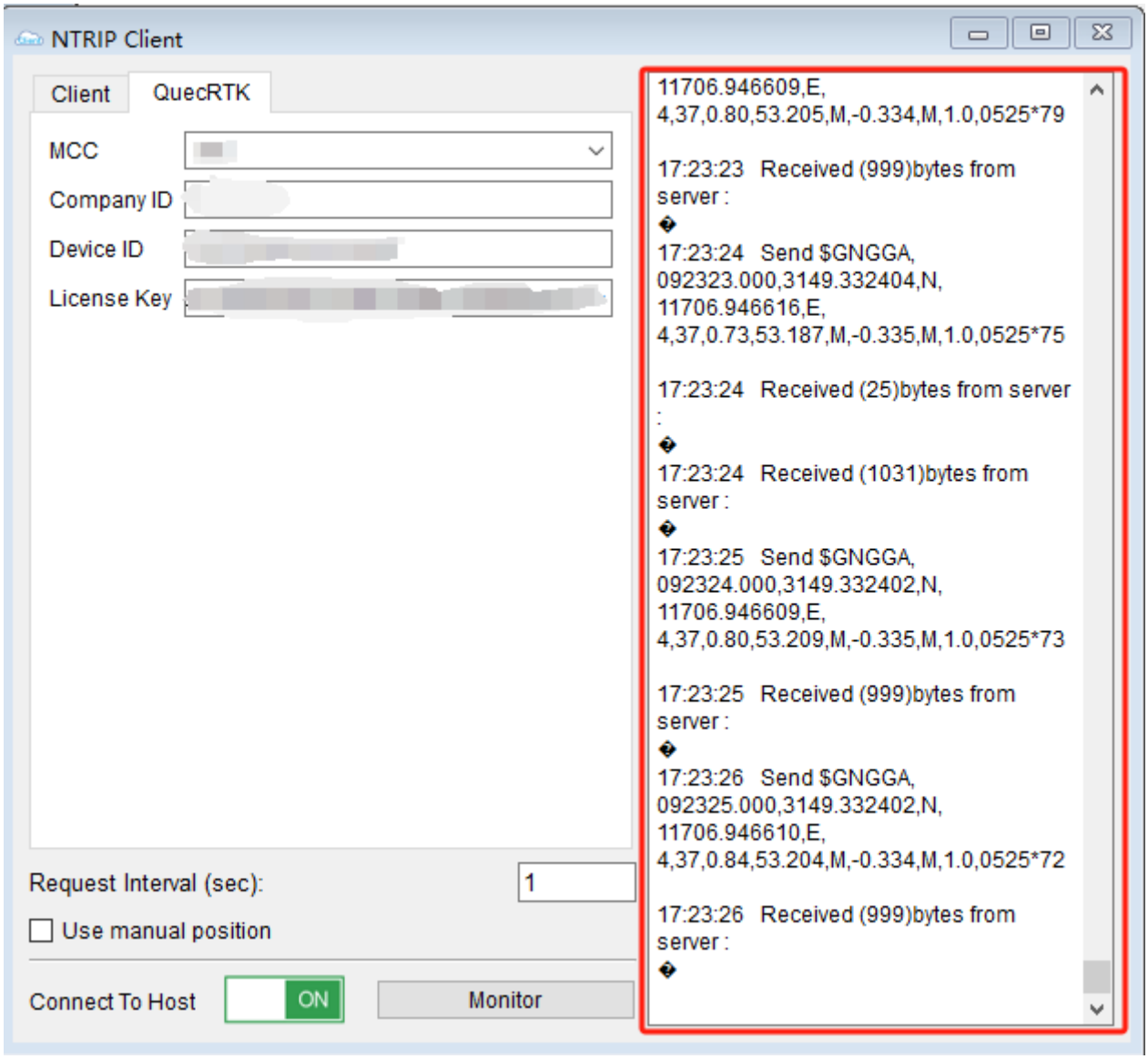


Figure 29: QuecRTK Client

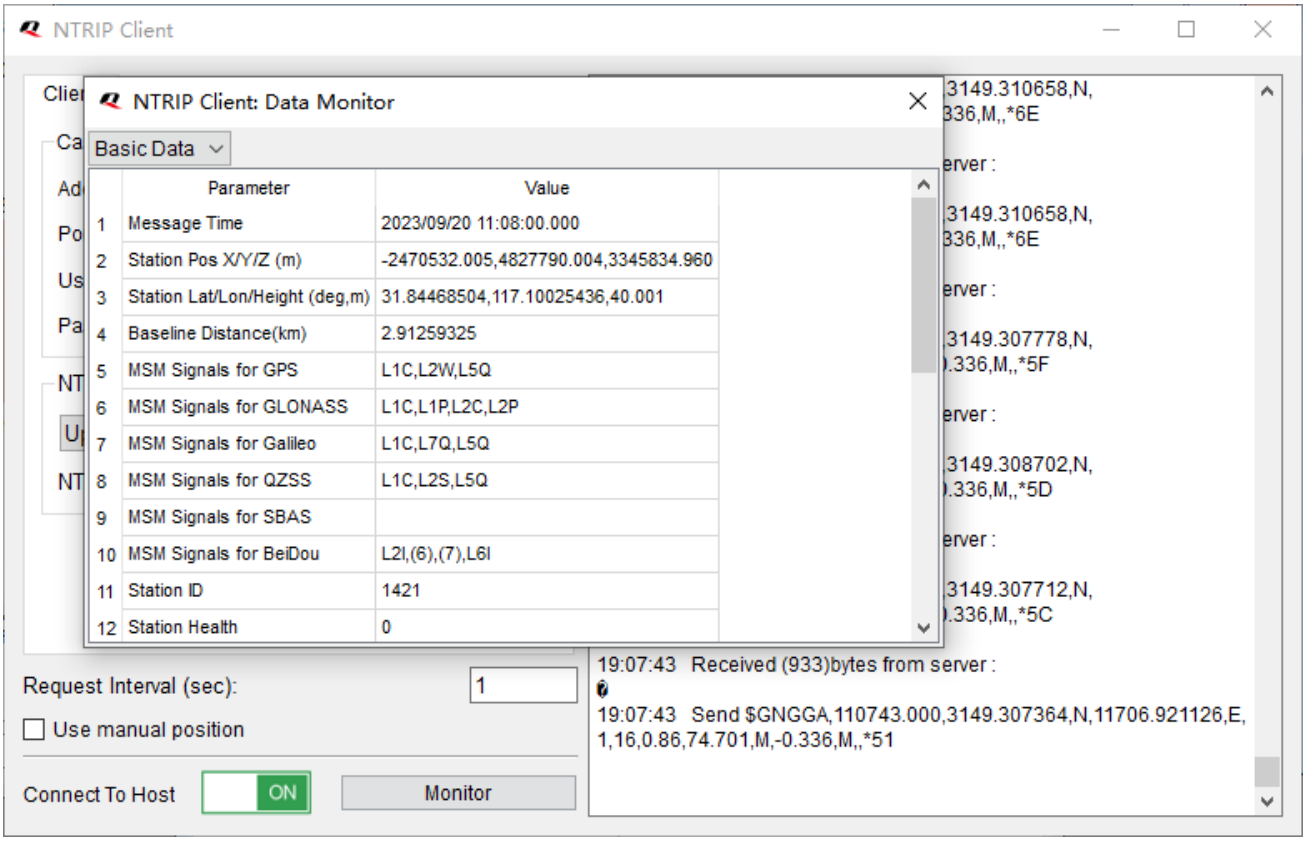


Figure 30: RTK Data Monitor

6 FAQs

6.1. DR

1. How to switch between ADR and UDR mode?

The LC29H (BA, CA)/QLM29H series can intelligently identify UDR and ADR modes. Upon initial power-up, the module/device defaults to UDR mode. Once it detects the input vehicle speed data, it will automatically switch to ADR mode.

2. What are the precautions for the module/device installation?

- Strictly fixed the module/device on the rigid structure of the carrier and ensure no relative motion between the module/device and the carrier.
- Ensure no relative motion between the module/device and the GNSS antenna, maintaining a fixed relative spatial position.

3. Does DR support fast calibration?

Yes. The premise is that the installation position of the module/device remains unchanged between two consecutive runs after it has been fixed onto the carrier. The module/device supports saving the calibration data into NVM. DO NOT change the installation position after the module/device has been calibrated.

4. Does DR support hot start function?

Yes. Currently, the DR hot start feature is enabled by default in the firmware, and **takes effect only when the vehicle is started from a stationary position**. To ensure its effectiveness, the following requirements must be met:

- The module/device must be strictly fixed onto the carrier;
- The installation position must remain unchanged between the vehicle being powering off and restarted. If the installation position of the module/device is changed, you must clear the DR calibration data (using **$PQTMDRCLR*53**) before recalibrating the DR. Failure to do so will result in degraded navigation performance and positioning anomalies before the DR can complete its re-calibration process.

An alternative method to achieve DR hot start is by manually saving the DR calibration data with **$PQTMDRSAVE*0F**. See *Chapter 2.3 DR Calibration Data Saving* for details.

6.2. RTK

1. How to address the issue that LC29H (BA, DA, EA)/QLM29H series cannot to enter differential mode?

1) Assess the environment. Observe the surrounding environment for potential interference sources, such as high-voltage lines, tall buildings, or dense forests. Once you have moved the module/device away from any identified interference sources, recheck the differential status to see if the issue is resolved.
2) Verify data injection. Ensure that the differential data is be correctly injected into the module/device.
3) Confirm RTK service compatibility. Check that the RTK service you are using matches the module/device in terms of supported satellite constellations and signal frequency bands.

2. How to address the issue that the differential mode remains in float state?

1) Assess the environment. Observe the surrounding environment for potential obstructions, such as houses, tall buildings, or dense forests that may cause occlusion. Ensure that the module/device is positioned away from such occlusions to minimize signal blockage.
2) Verify RTK service. Confirm that the module/device is connected to the appropriate RTK services. Verify that the GNSS constellations and satellite signal frequency bands supported by the service align with that of the module/devices to ensure compatibility.
3) Confirm differential data integrity. Ensure that differential data is being stably injected into the module/device. Monitor the injection period and data integrity. Be aware that any byte errors can result in an incorrect checksum, rendering the entire message frame unusable. This issue may also affect the usability of several subsequent differential data frames. User can verify the differential age through **GGA** message.
4) Check satellite usage. Examine the number of satellites being searched for and the number actively used by the module/device, as indicated in the **GSV** and **GSA** messages.
5) Evaluate signal strength. View the **GSV** message to assess the signal level of the satellites.

7 Appendix A References

Table 6: Related Documents

Document Name
[1] Quectel_GNSS_RTK_Application_Note
[2] Quectel_LC29H_Series&LC79H(AL)_GNSS_Protocol_Specification
[3] Quectel_GNSS_Module_EVB_User_Guide

Table 7: Terms and Abbreviations

Abbreviation	Description
ARP	Antenna Reference Point
BDS	BeiDou Navigation Satellite System
CORS	Continuously Operating Reference Stations
DI	Digital Input
DIO	Digital Input and Output
DR	Dead Reckoning
Galileo	Galileo Satellite Navigation System (EU)
GLONASS	Global Navigation Satellite System (Russia)
GGA	Global Positioning System Fix Data
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GSV	GNSS Satellites in View
GSA	GPS DOP and Active Satellites

Abbreviation	Description
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
MCC	Mobile Country Code
MSM	Multiple Signal Message
NMEA	NMEA (National Marine Electronics Association) Interface Standard
NTRIP	Networked Transport of RTCM via Internet Protocol
PAIR	Proprietary Protocol of Airoha
PI	Power Input
PQTM	Proprietary Protocol of Quectel
PVT	Position Velocity and Time
QZSS	Quasi-Zenith Satellite System
RTC	Real-Time Clock
RTD	Real Time Differential
RTCM	Radio Technical Commission for Maritime Services
RTK	Real-Time Kinematic
UAV	Unmanned Aerial Vehicle

8 Appendix B Special Characters

Table 8: Special Characters

Special Character	Definition
<CR>	Carriage return character.
<LF>	Line feed character.
<...>	Parameter name. Angle brackets do not appear in the message.
[...]	Optional field of a message. Square brackets do not appear in the message.
{…}	Repeated field of a message. Curly brackets do not appear in the message.
Underline	Default setting of a parameter.