L26-DR (UDR&ADRC)
DR Application Note

GNSS Module Series

Version: 1.2

Date: 2024-12-19

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

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	**L26-DR (UDR&ADRC) DR Application Note**
Subtitle	GNSS Module Series
Document Type	Application Note
Document Status	Released

Revision History

Version	Date	Description
-	2022-04-15	Creation of the document
1.0	2022-06-28	First official release
1.1	2023-07-20	1. Added the description about the DR hot-start feature (*Chapter 1*). 2. Added notes on $PSTMSETPAR and the output messages configured with bit 27. 3. Added examples for configuring DR hot-start feature. 4. Added $PQTM messages (*Chapter 3.2*).
1.2	2024-12-19	Numerous changes were made to this document. It should be read in its entirety.

Contents

Table Index

Figure Index

1 Introduction

This document provides a comprehensive overview of the dead reckoning (DR) feature of the Quectel L26-DR (ADRC, UDR) modules, including frame conventions, speed data injection, mounting, calibration, messages related to the DR functionality, operation guide and frequently asked questions.

This document applies to the following variants of the L26-DR module:

- L26-DR (ADRC)
- L26-DR (UDR)

Module	Vehicle Type	Vehicle Speed Injection
L26-DR (ADRC)	Two-wheeler	Yes. See *Chapter 2.2 Vehicle Speed Injection* for details.
L26-DR (UDR)	Four-wheeler	No

1.1. DR Overview

1.1.1. DR Introduction

Quectel offers two DR products: L26-DR (ADRC) and L26-DR (UDR) modules. L26-DR (ADRC) incorporates a 6-axis inertial sensor (3-axis gyroscope and 3-axis accelerometer) and an attached sensor such as a wheel speed sensor, is required to implement the automotive dead reckoning (ADR) function. It is primarily designed for two-wheel applications like motorcycles and scooters. On the other hand, L26-DR (UDR) is suitable for 4-wheel applications and does not require a wheel speed sensor.

L26-DR (UDR) module features the DR hot start function, which allows it to save calibration parameters in areas without GNSS signal. The calibrated module operating in a no-signal zone quickly acquires a position fix upon the next restart. This functionality enables efficient and accurate positioning even in challenging environments with limited or no GNSS reception.

L26-DR (ADRC) module is typically used in OEM use cases, while L26-DR (UDR) is typically used in aftermarket products.

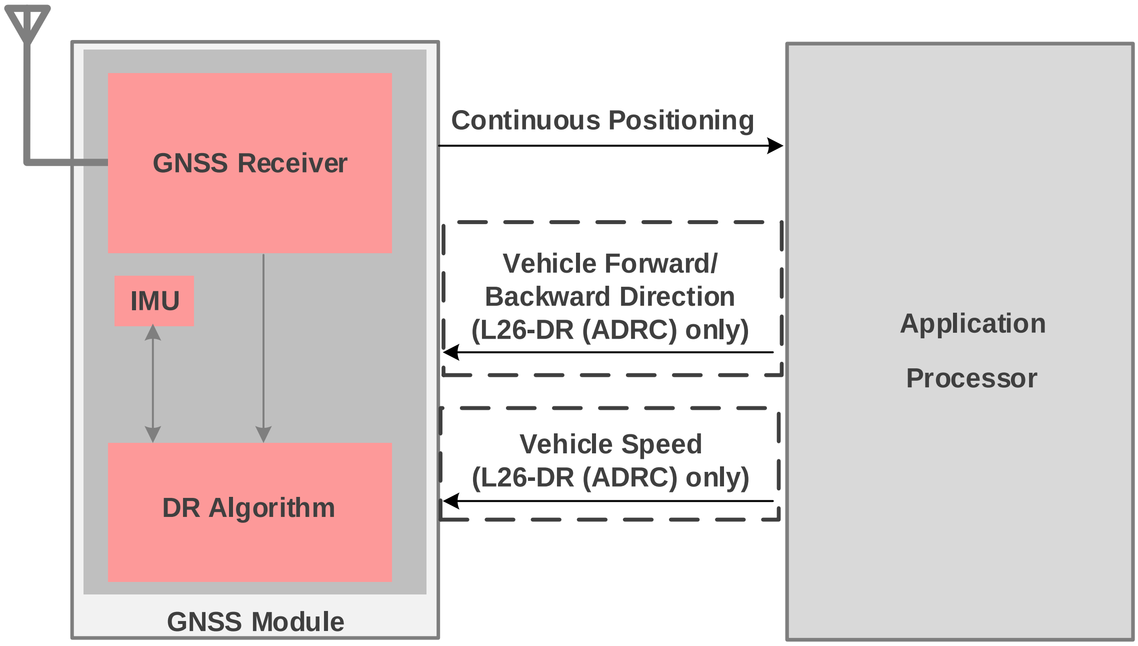

Figure 1: DR Application Architecture

1.1.2. DR Application Scenarios

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

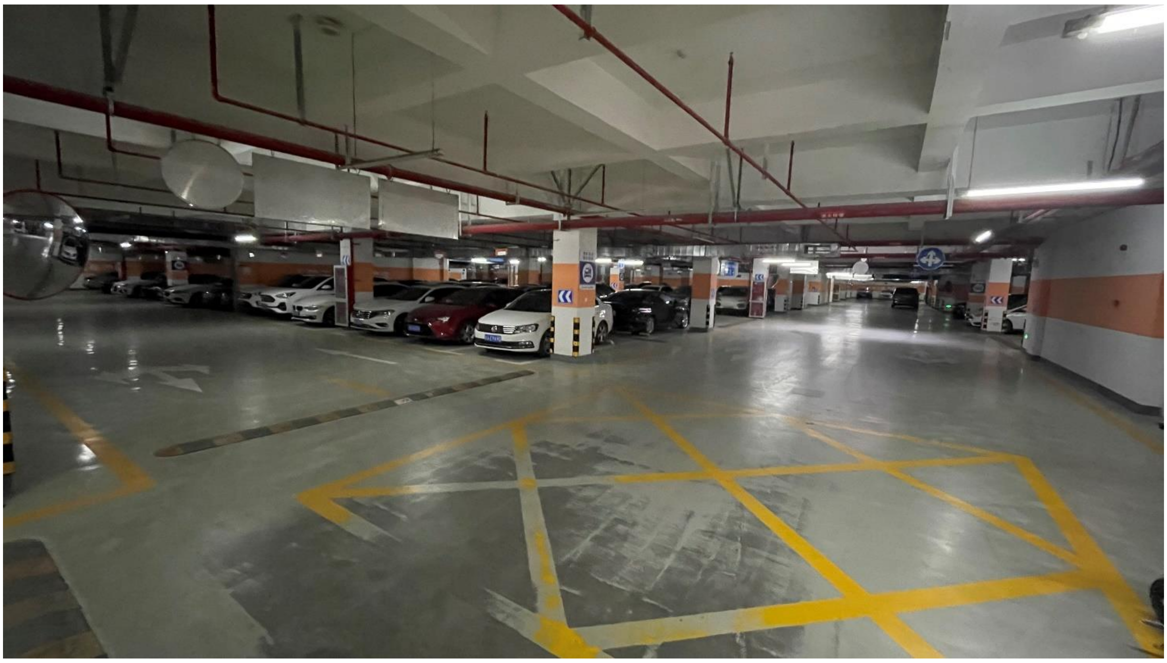

Figure 2: Underground Parking

2 DR Configuration

2.1. Sensor Reference Frame

MEMS IMU outputs its raw data (angular rates and acceleration) in the sensor reference frame. L26-DR (ADRC, UDR) is equipped with a 6-axis sensor which contains a 3-axis accelerometer and a 3-axis gyroscope. Both the accelerometer and the gyroscope measure along the same set of axes within the sensor reference frame and their outputs correspond to the same spatial directions. The IMU sensor frame is illustrated below:

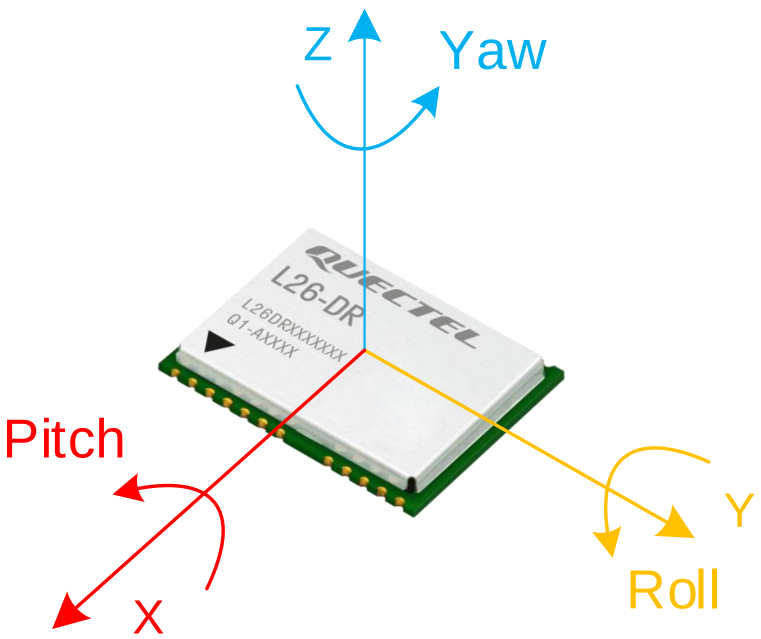

Figure 3: IMU Sensor Frame of L26-DR (ADRC, UDR)

NOTE

The IMU sensor frame of L26-DR (ADRC, UDR) shown in the figure above is applicable to both the gyroscope and the accelerometer within the IMU.

2.2. Vehicle Speed Injection

Vehicle speed injection refers to providing the L26-DR (ADRC) module with real-time vehicle speed information from external sources at a fixed frequency. This data is used in DR calculation to improve DR performance.

The L26-DR (ADRC) module can acquire speed and direction data in the following modes:

- Wheel Tick Mode (through the WHEELTICK and FWD pins).
- UART Mode (including UART Wheel Tick Mode and UART Speed Mode).

NOTE

PQTMCFGDRMODE can be used to enable Wheel Tick Mode or UART Mode, **PSTMSETPAR,1228** can be used to enable **PQTMVEHMSG** message output. For details about **PQTMCFGDRMODE**, **PSTMSETPAR,1228** and **PQTMVEHMSG**, see *Chapters 3.2.1 PQTMCFGDRMODE*, *3.3.3 PSTMSETPAR,1228* and *3.2.3 PQTMVEHMSG*.

2.2.1. Wheel Tick Mode

In this mode, the vehicle's wheel tick pulse signal is input through the module's WHEELTICK pin. The vehicle's forward/backward direction signal is input through the FWD pin. For details about WHEELTICK pin and FWD pin, see *document [1] hardware design*.

PQTM command for setting Wheel Tick Mode:

```
$PQTMCFGDRMODE,1,3*4D
```

PSTM commands for enabling the output of PQTMVEHMSG:

```
$PSTMSETPAR,1228,8000000,1*37
$PSTMSAVEPAR*58
$PSTMSRR*49
```

Result:

- If the Wheel Tick Mode is set successfully and **PQTMVEHMSG** output is enabled successfully, the module returns:

```
$PQTMVEHMSG, <Count>,<MsgID>, <WheelTickCount>,<FWD_Ind>*<Checksum><CR><LF>
```

- If failed, the module does not return the above message.

For details about the above **PQTMVEHMSG** message, refer to *Chapter 3.2.3.2 Wheel Tick Mode (Debug Message)*. For details about the **PSTMSAVEPAR** and **PSTMSRR** messages, refer to *document [2]* protocol specification.

NOTE

1. To ensure DR performance, the pulse width must not exceed 0.05 meters/pulse.
2. Enabling the **PQTMVEHMSG** message output is not necessary for vehicle speed injection and DR function use; once enabled, the **PQTMVEHMSG** message can be used to verify if the vehicle wheel tick pulse and direction signals are successfully injected:
   - You can verify if pulse signal injection is successful by observing the **<WheelTickCount>** field value while the vehicle is in motion. Successful injection is indicated by a value that is increasing.
   - You can verify if direction signal injection is successful by observing the **<FWD_Ind>** field value when the vehicle is moving forward/backward. Successful injection is indicated by a value that is changing.

2.2.2. UART Mode

2.2.2.1. UART Wheel Tick Mode

In this mode, the module obtains the vehicle wheel tick pulse count and direction data via the UART interface.

PQTM command for setting UART Mode:

```
$PQTMCFGDRMODE,1,2*4C
```

PSTM commands for enabling the output of PQTMVEHMSG:

```
$PSTMSETPAR,1228,8000000,1*37
$PSTMSAVEPAR*58
$PSTMSRR*49
```

Wheel tick pulse count and direction data sent to the module via UART:

```
$PQTMVEHMSG,<MsgID>,<TimeStamp>,<WheelTickCount>,<FWD_Ind>*<Checksum><CR><LF>
```

Result:

- If the module gets vehicle wheel tick pulse count and direction data via UART successfully, it returns:

```
$PQTMVEHMSG,<MsgID>,<TimeStamp>,<WheelTickCount>,<FWD_Ind>*<Checksum><CR><LF>
$PQTMVEHMSG,<Count>,<MsgID>, <WheelTickCount>,<FWD_Ind>*<Checksum><CR><LF>
```

- If failed, the module does not return the above messages.

For details about the above **PQTMVEHMSG** messages, refer to Chapter 3.2.3.1.3 Wheel Tick and

Chapter 3.2.3.1.4 Wheel Tick (Debug Message). For details about the **PSTMSAVEPAR** and **PSTMSRR** messages, refer to document [2] protocol specification.

NOTE

1. To ensure DR performance, the pulse width must not exceed 0.05 meters/pulse.
2. Enabling the **PQTMVEHMSG** message output is not necessary for vehicle speed injection and DR function use; once enabled, the **PQTMVEHMSG** message can be used to verify if the vehicle wheel tick pulse count and direction signals are successfully injected.
3. In UART Wheel Tick Mode, the wheel tick pulse count and direction data need to be injected at a frequency of at least 10 Hz.
4. Ensure that the baud rates of the host and module match.

2.2.2.2. UART Speed Mode

In this mode, the module obtains vehicle speed and direction data via the UART interface.

PQTM command for setting UART Mode:

```
$PQTMCFGDRMODE,1,2*4C
```

PSTM commands for enabling the output of PQTMVEHMSG:

```
$PSTMSETPAR,1228,8000000,1*37
$PSTMSAVEPAR*58
$PSTMSRR*49
```

Speed and direction data sent to the module via UART:

```
$PQTMVEHMSG,<MsgID>,<TimeStamp>,<VehSpeed>*<Checksum><CR><LF>
```

Result:

- If the module gets vehicle speed and direction data via UART successfully, it returns:

```
$PQTMVEHMSG,<MsgID>,<TimeStamp>,<VehSpeed>*<Checksum><CR><LF>
$PQTMVEHMSG,<Count>,<MsgID>,<VehSpeed>,<FWD_Ind>*<Checksum><CR><LF>
```

- If failed, the module does not return the above messages.

For details about the above **PQTMVEHMSG** messages, refer to Chapter 3.2.3.1.1 Speed and Chapter 3.2.3.1.2 Speed (Debug Message). For details about the **PSTMSAVEPAR** and **PSTMSRR** messages, refer to document [2] protocol specification.

NOTE

1. To ensure the DR performance, the injected speed value must be rounded to two decimal places.
2. Enabling the **PQTMVEHMSG** message output is not necessary for vehicle speed injection and DR function use, once enabled, the **PQTMVEHMSG** message can be used to verify if the vehicle speed and direction data are successfully injected.
3. In UART Speed Mode, vehicle speed and direction data need to be injected at a frequency of at least 10 Hz.
4. Ensure that the baud rates of the host and module match.

2.3. Mounting

The L26-DR (ADRC, UDR) module integrates both an IMU and a GNSS receiver. Therefore, you must ensure that the device incorporating the module is firmly fixed to vehicle body. No relative movement is allowed between the vehicle and the 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 to the vehicle frame. Mounting location should allow for easy access to the power supply and GNSS antenna, and should not be exposed to excessive heat.

2.3.1. L26-DR (ADRC) Mounting

When installing the L26-DR (ADRC) module, the projection of at least one axis on the plane of the vehicle must be parallel to the direction of vehicle's front. Depending on the installation orientation, use the **PQTMCFGIMUORI** command to configure the installation direction of the IMU relative to the vehicle. For details about this commend, refer to *Chapter 3.2.2 PQTMCFGIMUORI*.

For example, if the projection of the X-axis on the plane of the vehicle is parallel with the direction of the front, the installation reference model is as follows:

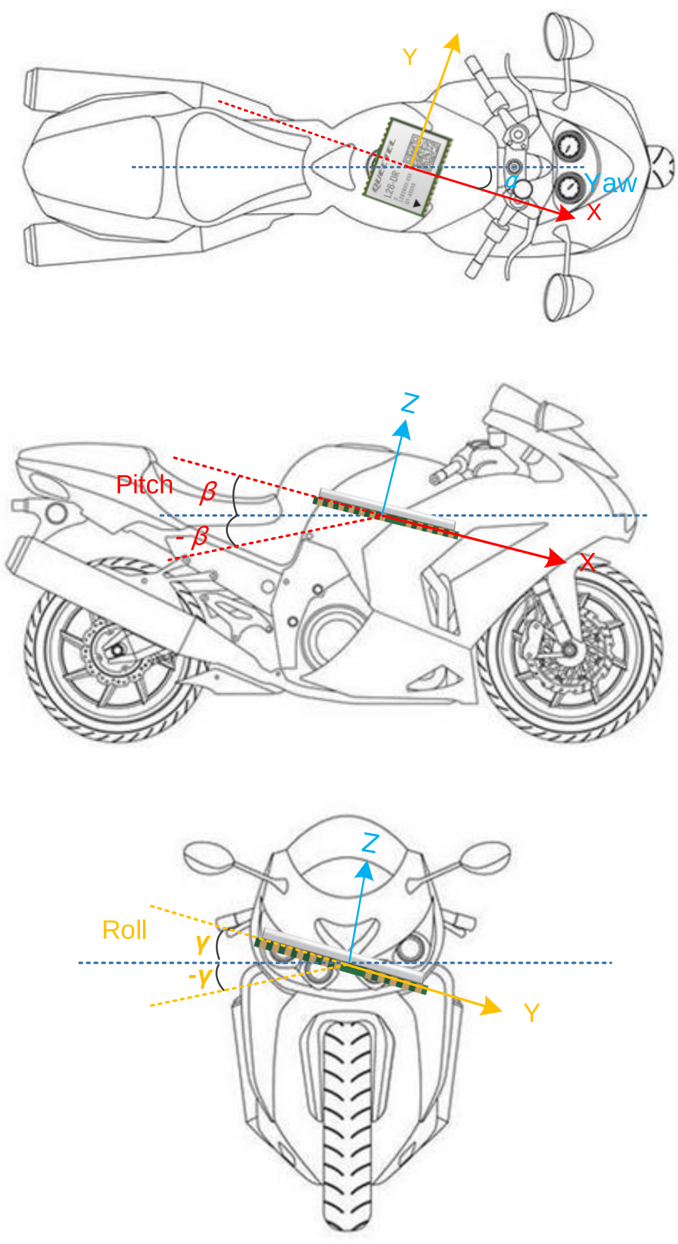

Figure 4: L26-DR (ADRC) Module Mounting Example

Referring to the above installation model, during module installation, the α angle cannot exceed ±5°; the β and γ angles are unrestricted and can rotate 360°.

Because the **PQTMCFGIMUORI** command defines 24 installation directions, the following two installation models are provided as examples to explain how to select the **<Mask>** parameter value:

Example 1:

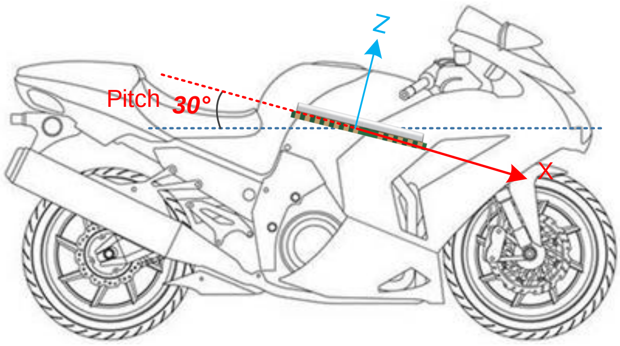

Figure 5: 30° Angle Mounting Example

In the above installation example, the pitch angle is 30°. This means that the angle between the X axis and the direction of the vehicle's front is 30°, and the angle between the Z axis and the direction of the vehicle's front is 60°. Select the axis with the smallest angle relative to the direction of the vehicle's front, i.e. the X axis. According to the rule of minimum rotation angle, the X-axis is virtually rotated counterclockwise to be parallel to the direction of the vehicle's front as shown in the figure above. In this orientation, the Z-axis is oriented upwards, and the X-axis is oriented forward. Therefore the **<Mask>** parameter in the IMU direction configuration command **PQTMCFGIMUORI** should be set to 2.

Example 2:

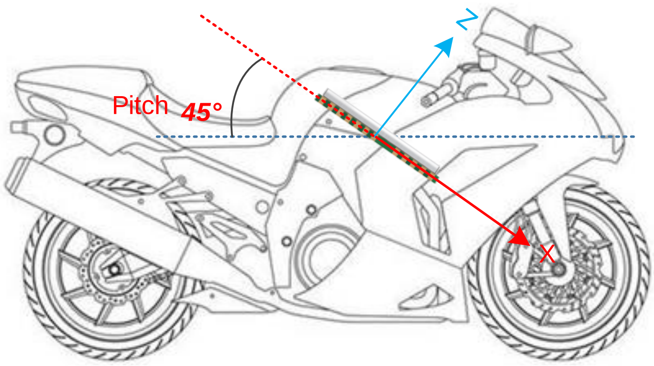

Figure 6: 45° Angle Mounting Example

In the above installation example, pitch angle is 45°. This means that the angle between the X axis and the direction of the vehicle's front is 45°, and the angle between the Z axis and the direction of the vehicle's front is 45°.Select the axis with the smallest angle relative to the direction of the vehicle's front, i.e. the X axis (or Z axis). According to the rule of minimum rotation angle, virtually rotate the selected axis counterclockwise (or clockwise) to be parallel to the direction of the vehicle's front as shown in the figure. In this rotation, Z-axis is positioned upwards, and X-axis points forward (or the X-axis points downward, and the Z-axis points forward). Therefore the **<Mask>** parameter in the IMU direction configuration command **PQTMCFGIMUORI** should be set to 2 (or 22).

NOTE

The “-” in the diagram is only used to indicate the relative rotation direction and does not correspond to the positive or negative value of the installation angle.

2.3.2. L26-DR (UDR) Mounting

When installing the L26-DR (UDR) module, one axis (Z axis is recommended) must be perpendicular to the plane of the vehicle. For example, when the Z axis is perpendicular to the plane of the vehicle, the installation reference model is as follows:

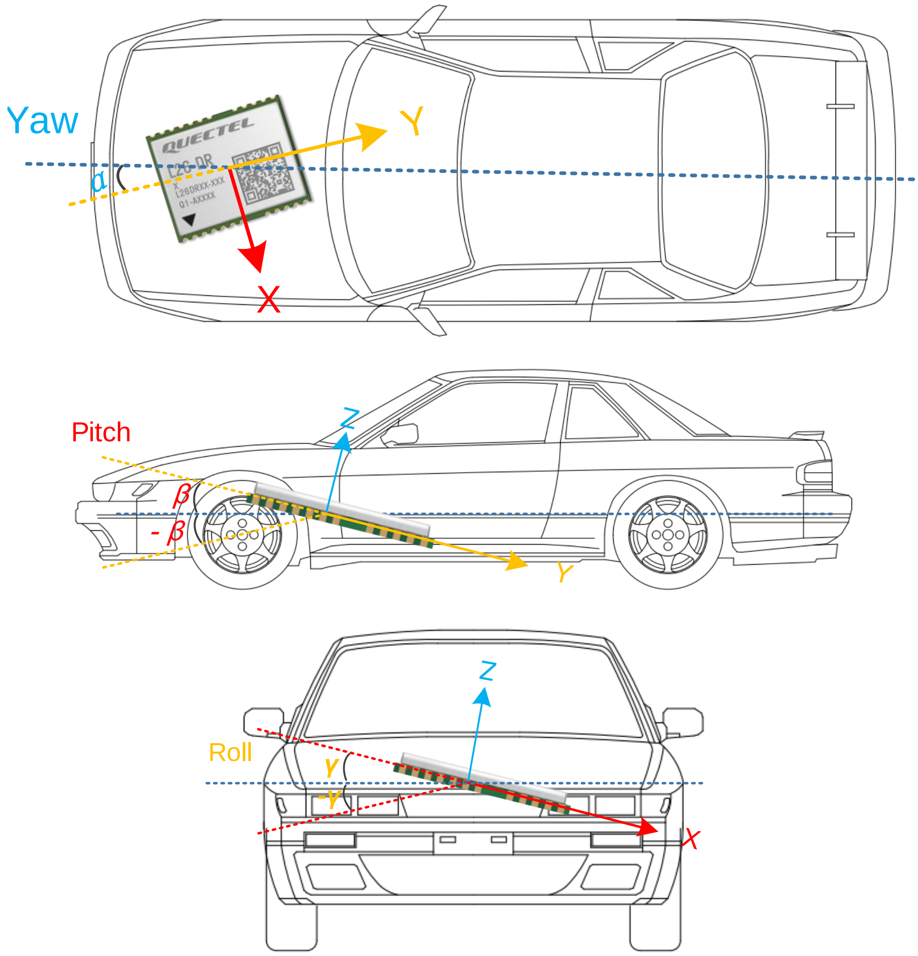

Figure 7: L26-DR (UDR) Module Mounting Example

Referring to the above installation model, during the installation process of the module there is no restriction on the α angle (0° ≤ *α* ≤ 360°). However, for optimal performance, it is recommended to maintain the β angle and the γ angle within the range of -15 degrees to 15 degrees (-15° ≤ β ≤ 15°, -15° ≤ γ ≤ 15°). In other words, the angle between the plane of the module and the horizontal plane should not exceed 15 degrees.

NOTE

1. Yaw angle (*α*) is unrestricted and can range from 0° to 360°.
2. The "-" in the diagram is only used to indicate the relative rotation direction and does not correspond to the positive or negative value of the installation angle.
3. When the β and γ angles exceed ±15°, positioning performance of the module degrades.

2.4. DR Calibration

2.4.1. L26-DR (ADRC) Calibration

To ensure accurate functioning of the dead reckoning functionality, the L26-DR (ADRC) module requires calibration. Follow the steps below for DR calibration:

Step 1 Fix the device incorporating the module on the vehicle frame firmly. Any displacement, rotation or tilt of the device relative to the vehicle plane, however small, may cause performance issues or void calibration.

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

Step 3 Power on the module. Do not interfere with or move the device after the module is powered on.

Step 4 Start the vehicle on a flat surface and drive in a straight line at a speed greater than 10 km/h for more than 3 minutes.

Step 5 The calibration process is considered complete when the **<NavMode>** field in **PQSOL** message indicates a combined solution (GNSS + DR). See *Chapter 3.1.1 PQSOL* for details about the message.

2.4.2. L26-DR (UDR) Calibration

To ensure accurate functioning of the dead reckoning functionality, the L26-DR (UDR) module requires calibration. Follow the steps below for DR calibration:

Step 1 Fix the device incorporating the module on the vehicle frame firmly. Any displacement, rotation or tilt of the device relative to the vehicle plane, however small, may cause performance issues or void calibration.

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

Step 3 Power on the module. Do not interfere with or move the device after the module is powered on.

Step 4 Start the vehicle on a flat surface and keep it stationary for 30 seconds at least. During driving, perform one or two dynamic maneuvers (such as waiting at traffic lights, acceleration and deceleration) to facilitate normal driving and satellite signal reception.

Step 5 The calibration process is considered complete when the **<NavMode>** field in **PQSOL** message indicates a combined solution (GNSS + DR). See *Chapter 3.1.1 PQSOL* for details about the message.

3 Messages

3.1. PQ Messages

This chapter outlines the PQ messages (proprietary NMEA messages defined by Quectel) supported by Quectel L26-DR (ADRC, UDR) modules.

3.1.1. PQSOL

Reports the navigation results.

Type:

Output

Synopsis:

```
$PQSOL,<SubID>,<Year>,<Month>,<Day>,<Hour>,<Minute>,<Millisecond>,<TimeStamp>,<WN_Ext>,<TOW>,<NavMode>,<Lat>,<Lon>,<AltElli>,<AltMSL>,<N_Vel>,<E_Vel>,<D_Vel>,<GroundSpeed>,<TravelDist>,<Roll>,<Pitch>,<Heading>,<Res1>,<Res2>,<N_Uncertain>,<E_Uncertain>,<D_Uncertain>,<VN_Uncertain>,<VE_Uncertain>,<VD_Uncertain>,<RollUncertain>,<PitchUncertain>,<YawUncertain>,<RollMisAng>,<PitchMisAng>,<YawMisAng>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<SubID>	Numeric	-	Message subtype ID. Fixed to 1.
<Year>	Numeric	Year	UTC year.
<Month>	Numeric	Month	UTC month. Range: 1–12.
<Day>	Numeric	Day	UTC day. Range: 1–31.
<Hour>	Numeric	Hour	UTC hour. Range: 0–23.
<Minute>	Numeric	Minute	UTC minute. Range: 0–59.

Field	Format	Unit	Description
<Millisecond>	Numeric	Millisecond	UTC millisecond. Range: 0–59950 (20 Hz output rate), only supported by L26-DR (UDR); 0–59900 (10 Hz output rate); 0–59800 (5 Hz output rate); 0–59000 (1 Hz output rate).
<TimeStamp>	Numeric	Millisecond	Milliseconds since power-on. 32-bit unsigned integer.
<WN_Ext>	Numeric	Week	GPS week number, including rollover.
<TOW>	Numeric	Millisecond	Time of week.
<NavMode>	Numeric	-	Navigation mode. 0 = No navigation. Bit 0: Reserved. Bit 1: GNSS-only. Bit 2: DR-only. Bit 3: Combined solution (GNSS + DR). Bit 4–8: Reserved. Bit 9: Odometer (speed/wheel tick pulse) in use. Bit 10, 11: Reserved. Bit 12: DGNSS. Bit 13: GNSS-only for when the DR algorithm cannot be enabled due to IMU data abnormality. Only supported by L26-DR (ADRC). Bit 14–31: Reserved.
<Lat>	Numeric	Degree	Latitude, scaled by 10^7.
<Lon>	Numeric	Degree	Longitude, scaled by 10^7.
<AltElli>	Numeric	cm	Ellipsoidal height.
<AltMSL>	Numeric	cm	Height above mean-sea-level.
<N_Vel>	Numeric	cm/s	Northward velocity.
<E_Vel>	Numeric	cm/s	Eastward velocity.
<D_Vel>	Numeric	cm/s	Downward velocity.
<GroundSpeed>	Numeric	cm/s	Speed over ground.
<TravelDist>	Numeric	cm	Traveled distance since power-on.
<Roll>	Numeric	Degree	Roll angle, scaled by 100. Range: -180° to 180°.

Field	Format	Unit	Description
<Pitch>	Numeric	Degree	Pitch angle, scaled by 100. Range: -90° to 90°.
<Heading>	Numeric	Degree	Heading angle, scaled by 100. Range: 0° to 360°.
<Res1>	Numeric	-	Reserved. Fixed to 0.
<Res2>	Numeric	-	Reserved. Variable value.
<N_Uncertain>	Numeric	cm	Uncertainty of position in northward direction.
<E_Uncertain>	Numeric	cm	Uncertainty of position in eastward direction.
<D_Uncertain>	Numeric	cm	Uncertainty of position in downward direction.
<VN_Uncertain>	Numeric	cm/s	Northward velocity uncertainty.
<VE_Uncertain>	Numeric	cm/s	Eastward velocity uncertainty.
<VD_Uncertain>	Numeric	cm/s	Downward velocity uncertainty.
<RollUncertain>	Numeric	Degree	Roll angle uncertainty, scaled by 100.
<PitchUncertain>	Numeric	Degree	Pitch angle uncertainty, scaled by 100.
<YawUncertain>	Numeric	Degree	Yaw angle uncertainty, scaled by 100.
<RollMisAng>	Numeric	Degree	Installation roll angle, scaled by 100.
<PitchMisAng>	Numeric	Degree	Installation pitch angle, scaled by 100.
<YawMisAng>	Numeric	Degree	Installation yaw angle, scaled by 100.

Example:

$PQSOL,1,2021,12,21,8,25,4000,61300,2189,203121999,2,318221924,1171155060,6587,6618,-3,4,0,5,156,1182,331,0,0,0,3267,6444,3273,211,206,137,20,10,10,0,0,0*68

NOTE

Attitude-related values (including roll, yaw and pitch) are invalid in GNSS-only mode.

3.1.2. PQGNSS

Reports GNSS-only fix information.

Type:

Output

Synopsis:

```
$PQGNSS,<SubID>,<Year>,<Month>,<Day>,<Hour>,<Minutes>,<Second>,<WN_Ext>,<TOW>,<NumSV>,<Lat>,<Lon>,<AltElli>,<GroundSpeed>,<Heading>,<Accuracy>,<N_Vel>,<E_Vel>,<D_Vel>,<Res1>,<Res2>,<N_Uncertain>,<E_Uncertain>,<D_Uncertain>,<VN_Uncertain>,<VE_Uncertain>,<VD_Uncertain>,<LeapSec>,<Res3>,<AltMSL>,<Res4>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<SubID>	Numeric	-	Message subtype ID. Fixed to 6.
<Year>	Numeric	Year	UTC year.
<Month>	Numeric	Month	UTC month.
<Day>	Numeric	Day	UTC day.
<Hour>	Numeric	Hour	UTC hour.
<Minutes>	Numeric	Minute	UTC minute.
<Second>	Numeric	Second	UTC second.
<WN_Ext>	Numeric	Week	GPS week number, including rollover.
<TOW>	Numeric	Millisecond	Time of week.
<NumSV>	Numeric	-	Number of satellites in use.
<Lat>	Numeric	Degree	Latitude, scaled by 10^7.
<Lon>	Numeric	Degree	Longitude, scaled by 10^7.
<AltElli>	Numeric	cm	Ellipsoidal height.
<GroundSpeed>	Numeric	cm/s	Speed over ground.
<Heading>	Numeric	Degree	Heading angle, scaled by 100.
<Accuracy>	Numeric	cm	Horizontal accuracy.

Field	Format	Unit	Description
<N_Vel>	Numeric	cm/s	Northward velocity.
<E_Vel>	Numeric	cm/s	Eastward velocity.
<D_Vel>	Numeric	cm/s	Downward velocity.
<Res1>	Numeric	-	Reserved. Fixed as 0.
<Res2>	Numeric	-	Reserved. Fixed as 0.
<N_Uncertain>	Numeric	cm	Uncertainty of position in northward direction.
<E_Uncertain>	Numeric	cm	Uncertainty of position in eastward direction.
<D_Uncertain>	Numeric	cm	Uncertainty of position in downward direction.
<VN_Uncertain>	Numeric	cm/s	Northward velocity uncertainty.
<VE_Uncertain>	Numeric	cm/s	Eastward velocity uncertainty.
<VD_Uncertain>	Numeric	cm/s	Downward velocity uncertainty.
<LeapSec>	Numeric	Second	Leap seconds.
<Res3>	Numeric	-	Reserved. Variable value.
<AltMSL>	Numeric	cm	Height above mean-sea-level.
<Res4>	Numeric	-	Reserved. Variable value.

Example:

$PQGNSS,6,2022,4,15,11,20,34,2205,472852000,11,318221499,1171158627,3673,5,0,2421,1,5,-1,0,0,1580,1834,1632,272,232,285,18,0,3704,12*34

3.1.3. PQSNR,21

Reports sensor data.

Type:

Output

Synopsis:

$PQSNR,<Type>,<NumSets>{,<TimeTag>,<X_GYRO>,<Y_GYRO>,<Z_GYRO>,<X_ACC>,<Y_ACC>,<Z_ACC>,<VehSpeed>,<Reverse>}*<Checksum><CR><LF>

Parameter:

Field	Format	Unit	Description
<Type>	Numeric	-	Data type. 21 = Mixed DR (GNSS + DR) sensor data.
<NumSets>	Numeric	-	Number of data set(s).
Start of repeat block. Repeat times: **<NumSets>**.			
<TimeTag>	Numeric	Millisecond	Milliseconds since power-on. 32-bit unsigned integer.
<X_GYRO>	Numeric	Deg/s	X-axis angular velocity of gyroscope, scaled by 1000.
<Y_GYRO>	Numeric	Deg/s	Y-axis angular velocity of gyroscope, scaled by 1000.
<Z_GYRO>	Numeric	Deg/s	Z-axis angular velocity of gyroscope, scaled by 1000.
<X_ACC>	Numeric	m/s^2	X-axis acceleration of accelerometer, scaled by 1000.
<Y_ACC>	Numeric	m/s^2	Y-axis acceleration of accelerometer, scaled by 1000.
<Z_ACC>	Numeric	m/s^2	Z-axis acceleration of accelerometer, scaled by 1000.
<VehSpeed>	Numeric	m/s	Vehicle speed, scaled by 100.
<Reverse>	Numeric	-	Vehicle direction. 0 = Forward 1 = Backward 8 = Un known
End of repeat block.			

Example:

```
$PQSNR,21,10,9800,449,-447,-178,566,1937,9589,0,8,9900,465,-412,-110,562,1937,9593,0,8,10000,5
21,-420,-128,567,1946,9596,0,8,10100,448,-441,-94,565,1937,9591,0,8,10200,454,-437,-118,565,1942,
9599,0,8,10300,487,-403,-129,559,1946,9603,0,8,10400,472,-461,-110,563,1940,9590,0,8,10500,443,-4
31,-121,561,1940,9587,0,8,10600,473,-416,-122,563,1943,9600,0,8,10700,474,-441,-86,564,1947,9597
,0,8*4B
```

3.1.4. PQSNR,23

Reports raw sensor data. The message is output only when the output rate is configured to multi-Hz. The entire data block is comprised of multiple messages.

Type:

Output

Synopsis:

```
$PQSNR,<Type>,<TimeTag>,<X_GYRO>,<Y_GYRO>,<Z_GYRO>,<X_ACC>,<Y_ACC>,<Z_ACC>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Type>	Numeric	-	Data type. 23 = Raw sensor data
<TimeTag>	Numeric	Millisecond	Milliseconds since power-on. 32-bit unsigned integer.
<X_GYRO>	Numeric	Deg/s	X-axis angular velocity of gyroscope, scaled by 1000.
<Y_GYRO>	Numeric	Deg/s	Y-axis angular velocity of gyroscope, scaled by 1000.
<Z_GYRO>	Numeric	Deg/s	Z-axis angular velocity of gyroscope, scaled by 1000.
<X_ACC>	Numeric	m/s^2	X-axis acceleration of accelerometer, scaled by 1000.
<Y_ACC>	Numeric	m/s^2	Y-axis acceleration of accelerometer, scaled by 1000.
<Z_ACC>	Numeric	m/s^2	Z-axis acceleration of accelerometer, scaled by 1000.

Example:

```
$PQSNR,23,3100,399,-519,-454,466,-511,9906*5C
```

3.1.5. PQEND

Indicates the end of an epoch.

Type:

Output

Synopsis:

```
$PQEND,<EndFlag1>,<EndFlag2>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<EndFlag1>	Numeric	-	End flag 1. Fixed to 255.
<EndFlag2>	Numeric	-	End flag 2. Fixed to 255.

Example:

```
$PQEND,255,255*4E
```

3.2. PQTM Messages

This chapter outlines the PQTM messages (proprietary NMEA messages defined by Quectel) supported by Quectel L26-DR (ADRC) module. Please note that these messages are not supported on the L26-DR (UDR) module.

3.2.1. PQTMCFGDRMODE

Sets/gets the speed injection mode. It takes effect immediately. Once set, the configurations remain unchanged even after a power cycling.

Type:

Set/Get

Synopsis:

```
//Set:
$PQTMCFGDRMODE,1,<SpeedMode>*<Checksum><CR><LF>
//Get:
$PQTMCFGDRMODE,0*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<SpeedMode>	Numeric	-	Speed mode. 0 = No speed information input 2 = UART Mode 3 = Wheel Tick Mode (WHEELTICK pin)

Result:

- If successful, the module returns:

```
//Set:
$PQTMCFGDRMODEOK*<Checksum><CR><LF>
//Get:
$PQTMDRMODE,0,<SpeedMode>*<Checksum><CR><LF>
```

- If failed, no message is sent as a reply.

Example:

```
//Set UART Mode to get vehicle speed:
$PQTMCFGDRMODE,1,2*4C
$PQTMCFGDRMODEOK*4B

//Set Wheel Tick Mode to get vehicle speed:
$PQTMCFGDRMODE,1,3*4D
$PQTMCFGDRMODEOK*4B

//Get speed injection mode:
$PQTMCFGDRMODE,0*53
$PQTMDRMODE,0,0*0D
```

NOTE

1. When configuring UART Mode to get vehicle speed, for corresponding input and output messages, refer to *Chapter 3.2.3.1 UART Mode*. When configuring Wheel Tick Mode to get vehicle speed, for corresponding output message, refer to *Chapter 3.2.3.2 Wheel Tick Mode (Debug Message)*.
2. 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.

3.2.2. PQTMCFGIMUORI

Sets/gets the IMU orientation relative to the vehicle. It takes effect immediately. Once set, the configurations remain unchanged even after a power cycling.

Type:

Set/Get

Synopsis:

```
//Set:
$PQTMCFGIMUORI,W,<Mask>*<Checksum><CR><LF>
//Get:
$PQTMCFGIMUORI,R*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Mask>	Numeric	-	IMU orientation mask. 1 = Z-axis up, X-axis right

Field	Format	Unit	Description
			2 = Z-axis up, X-axis forward 3 = Z-axis up, X-axis left 4 = Z-axis up, X-axis backward 5 = Z-axis down, X-axis right 6 = Z-axis down, X-axis forward 7 = Z-axis down, X-axis left 8 = Z-axis down, X-axis backward 9 = Y-axis up, X-axis right 10 = Y-axis up, X-axis forward 11 = Y-axis up, X-axis left 12 = Y-axis up, X-axis backward 13 = Y-axis down, X-axis right 14 = Y-axis down, X-axis forward 15 = Y-axis down, X-axis left 16 = Y-axis down, X-axis backward 17 = X-axis up, Z-axis right 18 = X-axis up, Z-axis forward 19 = X-axis up, Z-axis left 20 = X-axis up, Z-axis backward 21 = X-axis down, Z-axis right 22 = X-axis down, Z-axis forward 23 = X-axis down, Z-axis left 24 = X-axis down, Z-axis backward

Result:

- If successful, the module returns:

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

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 1: Error Code*

Table 1: Error Code

Field	Format	Unit	Description
<ErrCode>	Numeric	-	Error code. 1 = Invalid parameters. 2 = Failed execution.

Example:

```
//Set:
$PQTMCFGIMUORI,W,1*39
$PQTMCFGIMUORI,OK*77

//Get:
$PQTMCFGIMUORI,R*21
$PQTMCFGIMUORI,OK,1*6A
```

NOTE

The selection of **<Mask>** parameter values involves choosing the axis with the smallest angle relative to the vehicle's heading direction. According to the rule of minimum rotation angle, this axis is virtually rotated to be parallel with the vehicle's heading direction. Then, the directions of the three IMU axes are matched with the 24 parameter values. If the pitch or roll angle is 45°, there are two options for the parameter value, and you can choose either one. For details, see *Chapter 2.3.1 L26-DR (ADRC) Mounting*.

3.2.3. PQTMVEHMSG

Inputs/outputs the vehicle speed or wheel tick pulse count and direction information.

Type:

Input/Output

Synopsis:

```
$PQTMVEHMSG[,<Count>],<MsgID>[,<TimeStamp>][,<VehSpeed>][,<WheelTickCount>][,<FWD_Ind>]*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Count>	Numeric	-	**<Count>** = Number of message(s) × 100 This field is only available in **PQTMVEHMSG** debug

Field	Format	Unit	Description
			messages. See Chapters 3.2.3.1.2 Speed (Debug Message), 3.2.3.1.4 Wheel Tick (Debug Message) and 3.2.3.2 Wheel Tick Mode (Debug Message) for details.
<MsgID>	Numeric	-	Type of message input/output. 1 = Vehicle speed 2 = Wheel tick pulse
<TimeStamp>	Numeric	Millisecond	Timestamp since power-on. 32-bit unsigned integer. Always 0 when input. This field is not available in **PQTMVEHMSG** debug messages. See Chapters 3.2.3.1.1 Speed and 3.2.3.1.3 Wheel Tick for details.
<VehSpeed>	Numeric	m/s or cm/s	Vehicle speed information. This field is only available in UART Speed Mode. See Chapters 3.2.3.1.1 Speed and 3.2.3.1.2 Speed (Debug Message) for details.
<WheelTickCount>	Numeric	-	Wheel tick pulse count. This field is only available in UART Wheel Tick Mode and Wheel Tick Mode. See Chapters 3.2.3.1.3 Wheel Tick, 3.2.3.1.4 Wheel Tick (Debug Message) and 3.2.3.2 Wheel Tick Mode (Debug Message) for details.
<FWD_Ind>	Numeric	-	Forward/backward movement indicator. 0 = Forward 1 = Backward See Chapters 3.2.3.1.2 Speed (Debug Message), 3.2.3.1.3 Wheel Tick, 3.2.3.1.4 Wheel Tick (Debug Message) and 3.2.3.2 Wheel Tick Mode (Debug Message) for details.

3.2.3.1. UART Mode

This chapter outlines the relevant messages for injecting or obtaining vehicle speed or wheel tick pulse count through UART.

3.2.3.1.1. Speed

Inputs/outputs vehicle speed and direction information. Frequency of at least 10 Hz is required to inject vehicle speed and direction data.

Type:

Input/Output

Synopsis:

$PQTMVEHMSG,<MsgID>,<TimeStamp>,<VehSpeed>*<Checksum><CR><LF>

Parameter:

Field	Format	Unit	Description
<MsgID>	Numeric	-	Message type. Fixed to 1.
<TimeStamp>	Numeric	Millisecond	Timestamp since power-on. 32-bit unsigned integer. Always 0 when this message is input.
<VehSpeed>	Numeric	m/s	Vehicle speed. Negative values indicate backward speed.

Example:

$PQTMVEHMSG,1,0,3.60*2C
$PQTMVEHMSG,1,3748292,3.600000*2D

3.2.3.1.2. Speed (Debug Message)

Outputs vehicle speed and direction information injected via UART. It is for debugging purpose only.

Type:

Output

Synopsis:

$PQTMVEHMSG,<Count>,<MsgID>,<VehSpeed>,<FWD_Ind>*<Checksum><CR><LF>

Parameter:

Field	Format	Unit	Description
<Count>	Numeric	-	**<Count>** = Number of message(s) × 100.
<MsgID>	Numeric	-	Message type. Fixed to 1.
<VehSpeed>	Numeric	cm/s	Vehicle speed. Nonnegative number.
<FWD_Ind>	Numeric	-	Forward/backward indicator. 0 = Forward 1 = Backward

Example:

$PQTMVEHMSG,100,1,360,0*1F

3.2.3.1.3. Wheel Tick

Inputs/outputs vehicle wheel tick pulse count and direction information. Frequency of at least 10 Hz is required to inject vehicle wheel tick pulse count and direction data.

Type:

Input/Output

Synopsis:

$PQTMVEHMSG,<MsgID>,<TimeStamp>,<WheelTickCount>,<FWD_Ind>*<Checksum><CR><LF>

Parameter:

Field	Format	Unit	Description
<MsgID>	Numeric	-	Message type. Fixed to 2.
<TimeStamp>	Numeric	Millisecond	Timestamp since power-on. 32-bit unsigned integer. Always 0 when this message is input.
<WheelTickCount>	Numeric	-	Wheel tick pulse count.
<FWD_Ind>	Numeric	-	Forward/backward indicator. 0 = Forward 1 = Backward

Example:

$PQTMVEHMSG,2,0,0,0*18
$PQTMVEHMSG,2,3422954,0,0*17

3.2.3.1.4. Wheel Tick (Debug Message)

Outputs wheel tick pulse count and direction information injected via UART. It is for debugging purpose only.

Type:

Output

Synopsis:

$PQTMVEHMSG,<Count>,<MsgID>, <WheelTickCount>,<FWD_Ind>*<Checksum><CR><LF>

Parameter:

Field	Format	Unit	Description
<Count>	Numeric	-	**<Count>** = Number of message(s) × 100
<MsgID>	Numeric	-	Message type. Fixed to 2.
<WheelTickCount>	Numeric	-	Wheel tick pulse count.
<FWD_Ind>	Numeric	-	Forward/backward indicator. 0 = Forward 1 = Backward

Example:

$PQTMVEHMSG,100,2,0,0*19

3.2.3.2. Wheel Tick Mode (Debug Message)

Outputs wheel tick pulse count and direction information injected through WHEELTICK and FWD pins. It is for debugging purpose only.

Type:

Output

Synopsis:

$PQTMVEHMSG, <Count>,<MsgID>,<WheelTickCount>,<FWD_Ind>*<Checksum><CR><LF>

Parameter:

Field	Format	Unit	Description
<Count>	Numeric	-	**<Count>** = Number of message(s) × 100
<MsgID>	Numeric	-	Message type. Fixed to 2.
<WheelTickCount>	Numeric	-	Wheel tick pulse count.
<FWD_Ind>	Numeric	-	Forward/backward indicator. 0 = Forward 1 = Backward

Example:

$PQTMVEHMSG,100,2,0,0*19

NOTE

In Wheel Tick Mode, the output frequency is 10 Hz.

3.3. PSTM Messages

This chapter outlines the PSTM messages (proprietary NMEA messages defined by the chipset supplier) supported by Quectel L26-DR (ADRC, UDR) modules.

3.3.1. PSTMDRSENMSG

Outputs the IMU raw data.

Type:

Output

Synopsis:

$PSTMDRSENMSG,<MsgID>[,<…>]*<Checksum><CR><LF>

Parameter:

Field	Format	Unit	Description
<MsgID>	Numeric	-	Message ID. 24 = IMU temperature. 30 = IMU accelerometer. 31 = IMU gyroscope.
<…>	-	-	This field varies with the message type. See *Chapters 3.3.1.1 IMU Temperature Message (<MsgID> = 24)*, *3.3.1.2 IMU Accelerometer Message (<MsgID> = 30)* and *3.3.1.3 IMU Gyroscope Message (<MsgID> = 31)* for details.

3.3.1.1. IMU Temperature Message (<MsgID> = 24)

Outputs the IMU temperature.

Type:

Output

Synopsis:

```
$PSTMDRSENMSG,24,<TimeStamp>,<Temperature>,<Validity>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<TimeStamp>	Numeric	Microsecond	CPU tick count since power-on.
<Temperature>	Numeric	-	IMU temperature. Real temperature in degrees Celsius: Degrees Celsius = **<Temperature>** / 256 + 25.0
<Validity>	Numeric	-	Validity. 0 = Temperature is not valid. 1 = Temperature is valid.

Example:

```
$PSTMDRSENMSG,24,3248649605,-486,1*28
```

3.3.1.2. IMU Accelerometer Message (<MsgID> = 30)

Outputs the IMU accelerometer data.
FS = ±2g
Sensitivity = 0.061mg/LSB

Type:

Output

Synopsis:

```
$PSTMDRSENMSG,30,<TimeStamp>,<X_ACC>,<Y_ACC>,<Z_ACC>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<TimeStamp>	Numeric	Microsecond	CPU tick count since power-on.
<X_ACC>	Numeric	-	Raw signed 16-bit integer representing X-axis acceleration data in the sensor reference frame.
<Y_ACC>	Numeric	-	Raw signed 16-bit integer representing Y-axis acceleration data in the sensor reference frame.
<Z_ACC>	Numeric	-	Raw signed 16-bit integer representing Z-axis acceleration data in the sensor reference frame.

Example:

$PSTMDRSENMSG,30,3248512115,299,-35,16514*02

3.3.1.3. IMU Gyroscope Message (<MsgID> = 31)

Outputs the IMU gyroscope data.
FS = ±125 dps
Sensitivity = 4.375 mdps/LSB

Type:

Output

Synopsis:

$PSTMDRSENMSG,31,<TimeStamp>,<X_GYRO>,<Y_GYRO>,<Z_GYRO>*<Checksum><CR><LF>

Parameter:

Field	Format	Unit	Description
<TimeStamp>	Numeric	Microsecond	CPU tick count since power-on.
<X_GYRO>	Numeric	-	Raw signed 16-bit integer representing X-axis angular rate data in the sensor reference frame.
<Y_GYRO>	Numeric	-	Raw signed 16-bit integer representing Y-axis angular rate data in the sensor reference frame.
<Z_GYRO>	Numeric	-	Raw signed 16-bit integer representing Z-axis angular rate data in the sensor reference frame.

Example:

$PSTMDRSENMSG,31,5855224,62,-119,-49*2A

3.3.2. PSTMIMUSELFTESTCMD

Executes the self-test procedure in IMU.

Type:

Command

Synopsis:

$PSTMIMUSELFTESTCMD,<IMU_Cat>*<Checksum><CR><LF>

Parameter:

Field	Format	Unit	Description
<IMU_Cat>	Numeric	-	IMU type. 0 = Accelerometer 1 = Gyroscope

Result:

- If successful, the module returns:

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

- If failed, the module returns:

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

- If the self-test command is not supported by the mounted IMU or by the FW (sensor layer is not present in FW), the module returns:

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

Example:

```
$PSTMIMUSELFTESTCMD,1*16
$PSTMIMUSELFTESTCMDOK*0F
```

3.3.3. PSTMSETPAR,1228

Configures the message list and the output rate. The settings can only take effect if you save the configuration with the **PSTMSAVEPAR** message and then reset the system with the **PSTMSRR** message. Once saved, the configurations remain unchanged even after a power cycling.

Type:

Set

Synopsis:

```
$PSTMSETPAR,1228,<Bitmask>,<ON_OFF>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Bitmask>	Hexadecimal	-	See *Table 2: List of Configuration Bits* for details.
<ON_OFF>	Numeric	-	Enable/disable. 1 = Enable

Field	Format	Unit	Description
			2 = Disable (Clear)

Table 2: List of Configuration Bits

Bit	Bitmask (32 bits)	Function/Message
22, 23	0x000000	Output rate control: 1 Hz (default).
	0x400000	Output rate control: 5 Hz.
	0x800000	Output rate control: 10 Hz (UART baud rate ≥ 230400 bps).
	0xC00000	Output rate control: 20 Hz (UART baud rate ≥ 230400 bps). Only supported by L26-DR (UDR).
24	0x1000000	Enable/Disable **PQSOL** message. 0 = Disable 1 = Enable
25	0x2000000	Enable/Disable **PQGNSS** message. 0 = Disable 1 = Enable
26	0x4000000	Enable/Disable **PQSNR,21** and **PQEND** messages. 0 = Disable 1 = Enable
27 [1]	0x8000000	Enable/Disable **PQSNR,23** and **PQTMVEHMSG** messages. 0 = Disable 1 = Enable
28	0x10000000	Enable/Disable **PSTMDRSENMSG** message. 0 = Disable 1 = Enable For L26-DR (UDR), default = 1. For L26-DR (ADRC), default = 0.
29	0x20000000	Enable/Disable DR feature. 0 = Enable 1 = Disable Only supported by L26-DR (UDR).
30	0x40000000	Enable/Disable DR hot start feature. 0 = Disable 1 = Enable Only supported by L26-DR (UDR).

Result:

- If successful, the module returns:

```
$PSTMSETPAROK,1228*<Checksum><CR><LF>
```

- If failed, the module returns:

```
$PSTMSETPARERROR,1228*<Checksum><CR><LF>
```

Example:

```
1.   Set the output rate to 5 Hz:
//Clear the rate bitmask:
$PSTMSETPAR,1228,0xC00000,2*32
//Response:
$PSTMSETPAROK,1228*3A
//Set the bitmask:
$PSTMSETPAR,1228,0x400000,1*43
//Response:
$PSTMSETPAROK,1228*3A
//Save the configuration:
$PSTMSAVEPAR*58
//Response:
$PSTMSAVEPAROK*5C
//Reset the system:
$PSTMSRR*49

2.   Disable PQSOL message:
//Clear the bitmask:
$PSTMSETPAR,1228,0x1000000,2*75
//Response:
$PSTMSETPAROK,1228*3A
//Save the configuration:
$PSTMSAVEPAR*58
//Response:
$PSTMSAVEPAROK*5C
//Reset the system:
$PSTMSRR*49

3.   Enable PQSOL message
//Set the bitmask:
$PSTMSETPAR,1228,0x1000000,1*76
//Response:
$PSTMSETPAROK,1228*3A
//Save the configuration:
$PSTMSAVEPAR*58
```

```
//Response:
$PSTMSAVEPAROK*5C
//Reset the system:
$PSTMSRR*49

4.  Disable DR hot start feature:
//Clear the bitmask:
$PSTMSETPAR,1228,0x40000000,2*40
//Response:
$PSTMSETPAROK,1228*3A
//Save the configuration:
$PSTMSAVEPAR*58
//Response:
$PSTMSAVEPAROK*5C
//Reset the system:
$PSTMSRR*49

5.  Enable DR hot start feature:
//Set the bitmask:
$PSTMSETPAR,1228,0x40000000,1*43
//Response:
$PSTMSETPAROK,1228*3A
//Save the configuration:
$PSTMSAVEPAR*58
//Response:
$PSTMSAVEPAROK*5C
//Reset the system:
$PSTMSRR*49
```

NOTE

1. [1)] When the output rate is 1 Hz, the module only outputs **PQTMVEHMSG**, and when the output rate is greater than 1 Hz, the module outputs **PQTMVEHMSG** and **PQSNR,23**.
2. For details about **PSTMSAVEPAR** and **PSTMSRR**, see *document [2] protocol specification*.
3. 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.

3.3.4. PSTMGETPAR,1228

Gets the message output list and output rate configuration.

Type:

Get

Synopsis:

$PSTMGETPAR,1228*<Checksum><CR><LF>

Result:

- If successful, the module returns:

$PSTMSETPAR,1228,<Bitmask>*<Checksum><CR><LF>

Parameter included in the result:

Field	Format	Unit	Description
<Bitmask>	Hexadecimal	-	See *Table 2: List of Configuration Bits* for details.

- If failed, the module returns:

$PSTMGETPARERROR*<Checksum><CR><LF>

Example:

$PSTMGETPAR,1228*2A
$PSTMSETPAR,1228,0x41020030*5E

4 DR Operation Guide

This chapter explains how to inject vehicle speed in Wheel Tick Mode using the GNSS MODULE EVB equipped with the L26-DR (ADRC) module. It also explains how L26-DR (ADRC) implements the DR function using the Quectel QGNSS tool. For more information about GNSS MODULE EVB, you can refer to *document [3] EVB user guide*.

4.1. EVB and Antenna Mounting

The EVB and antenna must be securely mounted in a low-vibration area of the vehicle and rigidly attached to the vehicle body to avoid any relative movement. Using the Quectel test vehicle and test tooling as an example follow the steps below to mount the EVB and antenna:

Step 1 Mount the rigid tooling on the motorcycle with screws to ensure that it is firmly fixed to the vehicle.

Step 2 Mount the EVB with the module on the tooling surface using adhesive.

Step 3 Make sure that the GNSS antenna is rigidly fixed to the vehicle (The antenna should be mounted on the vehicle in a position without GNSS signal obstructions).

EVB installation is shown in the figure below:

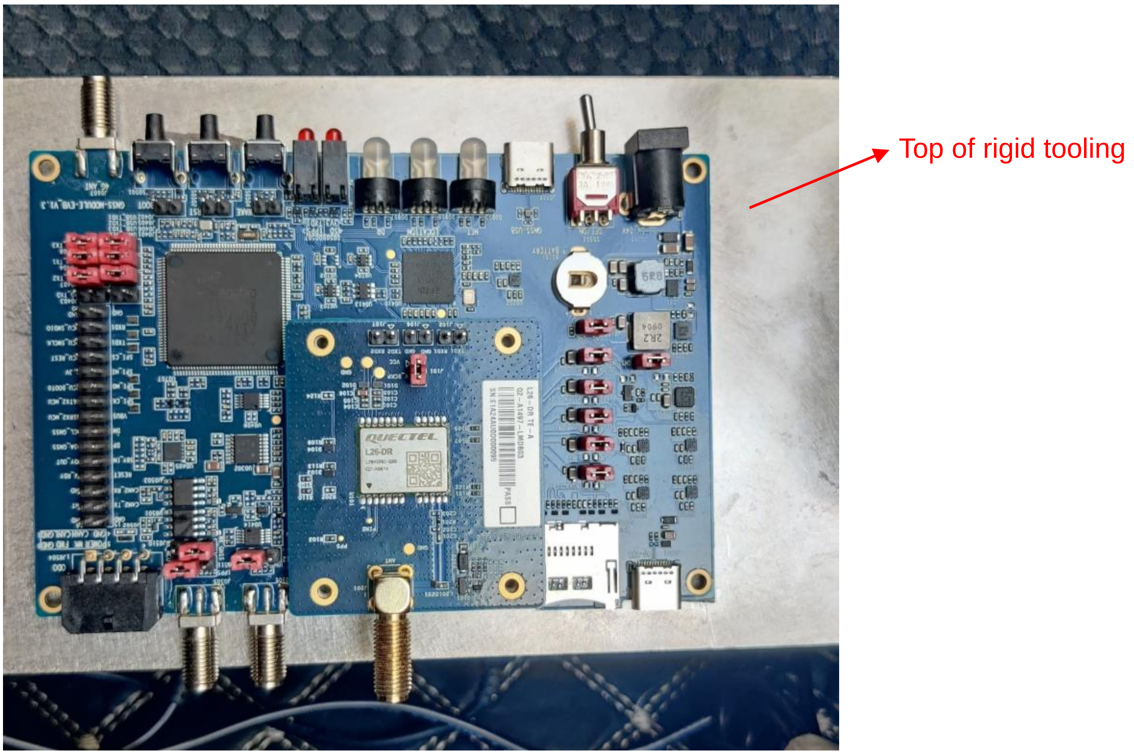

Figure 8: EVB Mounting on Rigid Tooling

4.2. Vehicle Speed Injection and EVB Connection

Step 1 To inject the vehicle speed in the Wheel Tick Mode, you need to use jumper caps to short circuit the pins of J0510, J0511 and J0512 interfaces. The jumper cap connection is shown in the figure below:

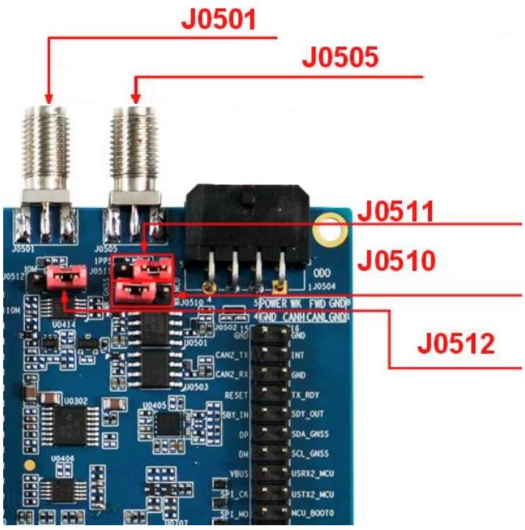

Figure 9: Wheel Tick Mode Jumper Cap Connections

Step 2 The vehicle's forward/backward direction and wheel tick pulse signals are injected through the SMA connectors J0501 and J0505.

Step 3 Connect the EVB to the PC via a Micro-USB cable.

The antenna, EVB, and PC connection methods are shown in the figure below:

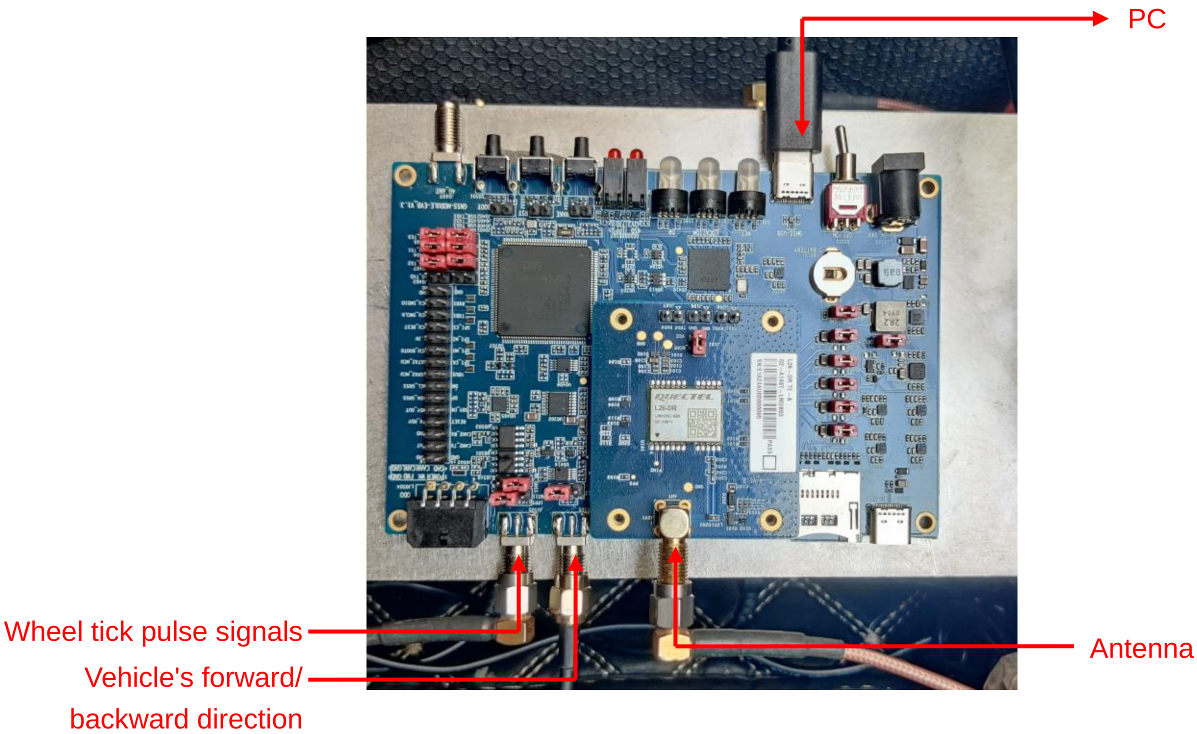

Figure 10: EVB Connections in Wheel Tick Mode

NOTE

On the GNSS MODULE EVB V1.3, the polarity of the input signal of the J0501 interface (FWD pin) is inverted upon reaching the module end.

4.3. Configuration

Step 1 Toggle the power switch to ON position to power on the EVB.

Step 2 Open the QGNSS tool and follow the steps below to configure the serial port:

1) Click  to open the "**Device information**" window;
2) Select the "**L26ADRC**" module;
3) Select the corresponding serial port "**COM xx**" (When the device connects to the PC, 4 random but consecutive COM port numbers appear. Choose the lowest COM port number);

4) Select the baud rate (The default baud rate for the L26-DR (ADRC) firmware version is 115200 bps);
5) Click "**OK**".

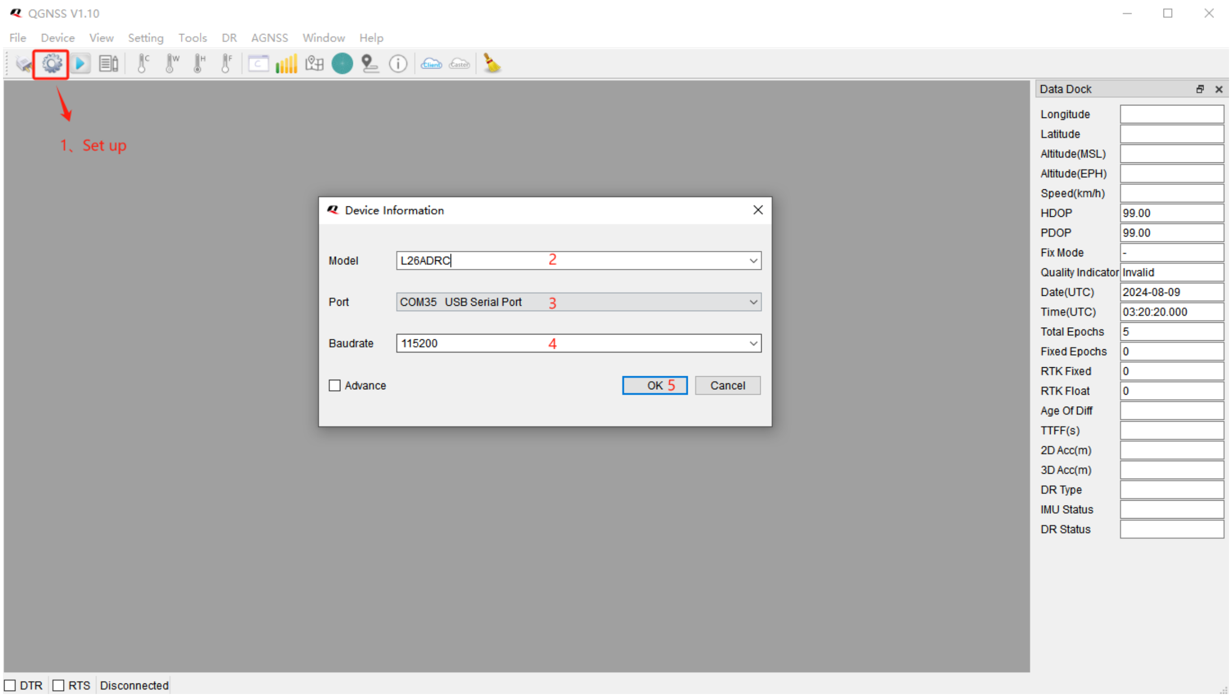

Figure 11: Serial Port Configuration

Step 3 Verify if the message output and positioning status in the "**Text data**" window are valid:

1) Click "**View**" > "**Text data**" to check whether the output is valid;
2) Enter "RMC" in "**Filter**" to get **RMC** sentences and check if the third field is "**A**", which means the position has been fixed.

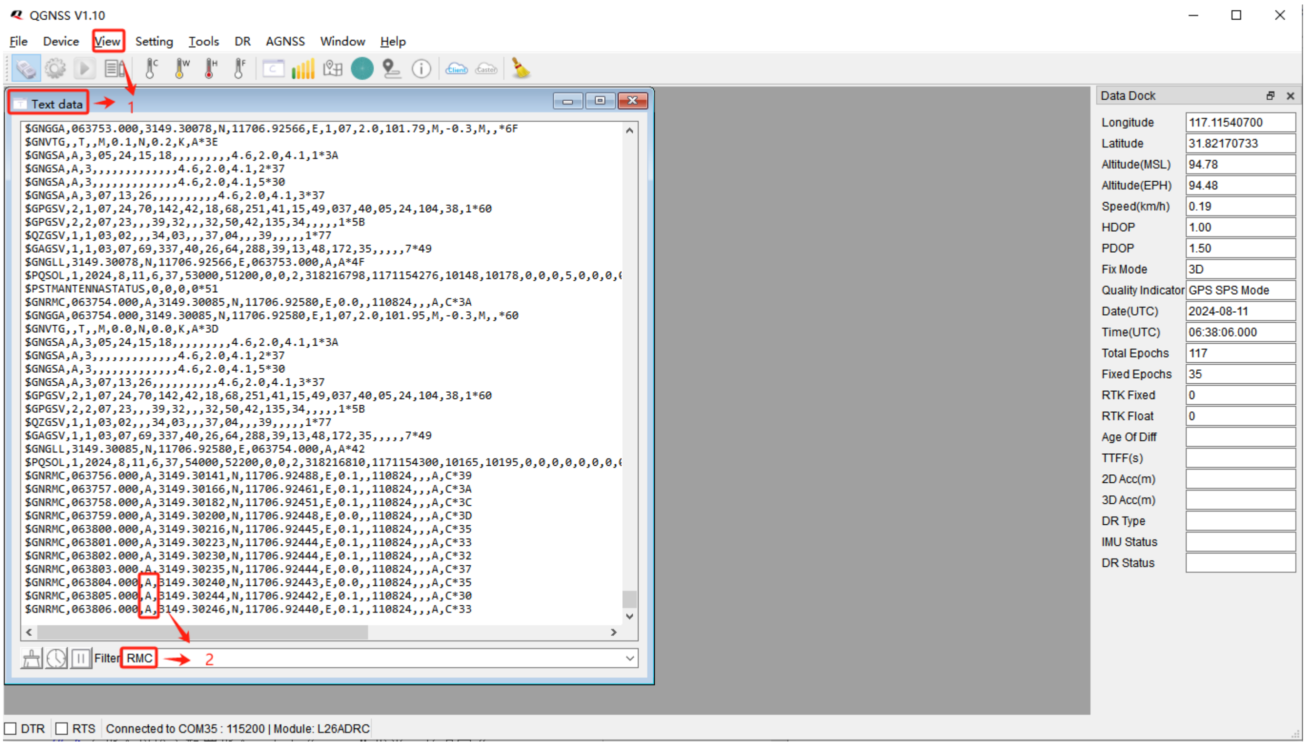

Figure 12: Positioning Status Check

Step 4 Click to open the "**Command console**" window and enter the relevant commands to set the vehicle speed injection mode and installation direction.

1) Refer to *Chapter 2.2.1 Wheel Tick Mode* and enter the configuration command on the command line. Click "**Send**", and observe whether the "**Text data**" window returned the result indicating successful command execution.
2) According to the installation of the module, refer to *Chapter 2.3.1 L26-DR (ADRC) Mounting* and *3.2.2 PQTMCFGIMUORI* to configure the installation direction of the IMU relative to the vehicle. For example, Z-axis is up and X-axis is forward, so the **<Mask>** parameter value should be set to 2. Enter the configuration command on the command line, click "**Send**" to execute it, and check if the "**Text data**" window returned the result indicating successful command execution.

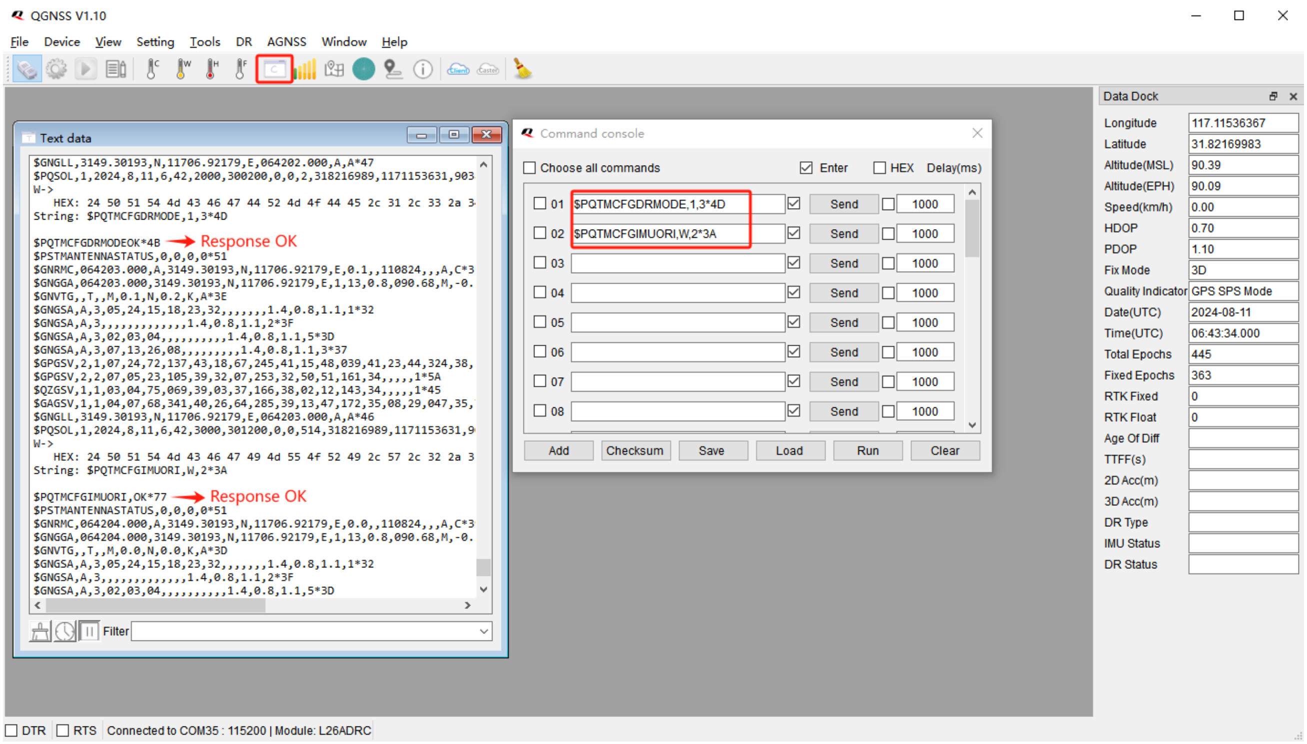

Figure 13: Vehicle Speed Injection Mode and Installation Direction Configuration

4.4. DR Calibration

DR calibration requires dynamic driving of a vehicle with a mounted module. You can refer to *Chapter 2.4.1 L26-DR (ADRC) Calibration* for instructions on how to perform module DR calibration.

During the calibration process, check the **<NavMode>** parameter value of the **PQSOL** message in the "**Text data**" window. The output value is a decimal number 4648 in the figure below. When converted to binary, bit 3 is 1, which indicates GNSS + DR navigation mode, that is, successful DR calibration. For details about the **PQSOL** message, see *Chapter 3.1.1 PQSOL*.

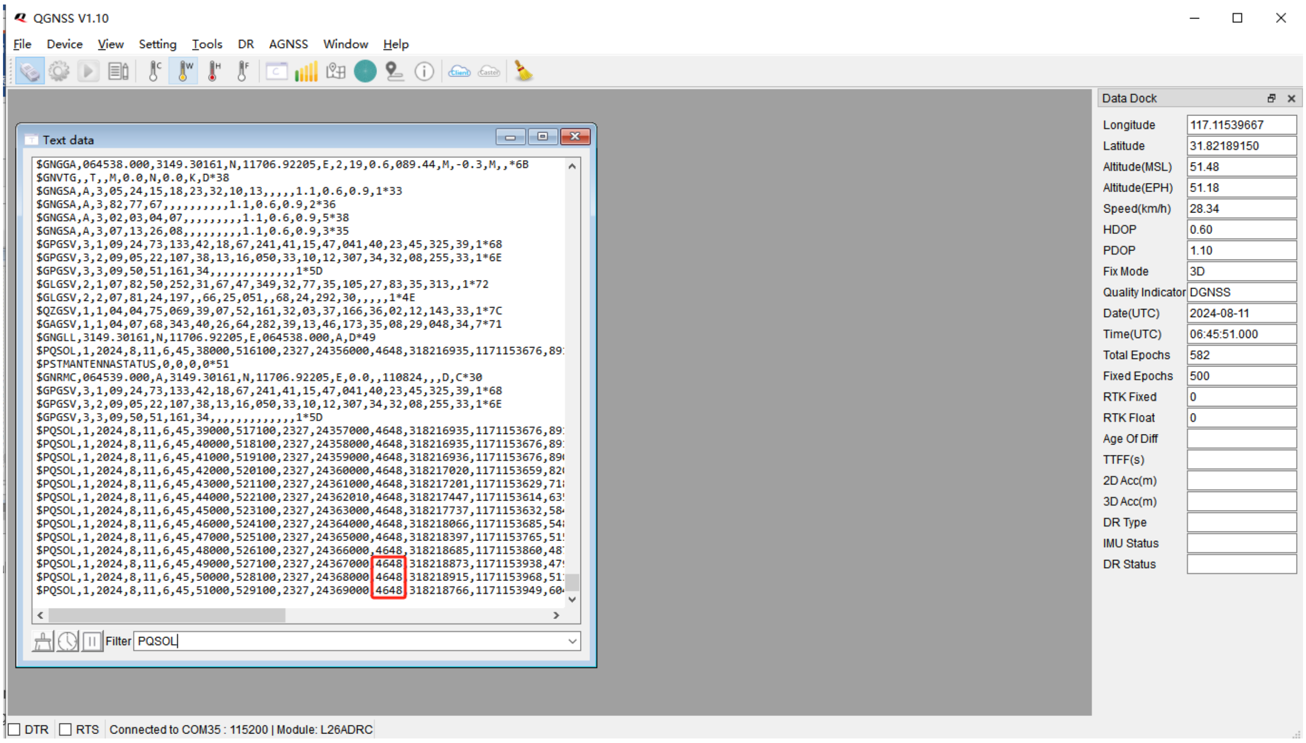

Figure 14: Message Indicating Successful DR Calibration

4.5. Function Implementation

After completing the installation, configuration and calibration of the module, test the DR function by driving through some specific environments, such as garages and tunnels.

NOTE

The above examples use the QGNSS V1.10 tool. There may be differences between QGNSS tool versions. For detailed usage of the tool, see *document [4] QGNSS user guide*.

5 FAQs

1. Under what circumstances ADR performance may not be guaranteed?

- Abnormal vehicle speed input;
- No reversing signal is input while the vehicle is in reverse gear;
- Poor GNSS signal quality during DR calibration, such as severe signal obstruction;
- Scenarios with extremely poor GNSS signal;
- The module is not securely mounted.

2. What precautions should be taken for module/device installation?

- Firmly affix the module to the rigid structure of the vehicle and ensure that there is no relative movement between the module and the vehicle.
- Ensure that there is no relative movement between the module/device and the GNSS antenna.

6 Appendix A References

Table 3: Related Documents

Document Name
[1] Quectel_L26-DR_Series_Hardware_Design
[2] Quectel_L26-DR&L26-P&L26-T&LC98S_Series_GNSS_Protocol_Specification
[3] Quectel_GNSS_Module_EVB_User_Guide
[4] Quectel_QGNSS_User_Guide

Table 4: Terms and Abbreviations

Abbreviation	Description
ADR	Automotive Dead Reckoning
CPU	Central Processing Unit
DGNSS	Difference Global Navigation Satellite System
DR	Dead Reckoning
EVB	Evaluation Board
FW	Firmware
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
ID	Identifier
IMU	Inertial Measurement Unit
LSB	Least Significant Bit
MEMS	Micro-Electro-Mechanical System

Abbreviation	Description
MSL	Mean-sea-level
NMEA	NMEA (National Marine Electronics Association) 0183 Interface Standard
OEM	Original Equipment Manufacturer
PC	Personal Computer
SMA	SubMiniature Version A
SV	Satellite Vehicle
TOW	Time of Week
UART	Universal Asynchronous Receiver/Transmitter
UDR	Untethered Dead Reckoning
USB	Universal Serial Bus
UTC	Coordinated Universal Time
WN	Week Number

7 Appendix B Special Characters

Table 5: Special Characters

Special Character	Description
<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.
<u>Underline</u>	Default setting of a parameter.