L26-DR (ADR)

DR Application Note

GNSS Module Series

Version: 1.2

Date: 2025-02-25

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

Use and Disclosure Restrictions

License Agreements

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 (ADR) DR Application Note**
Subtitle	GNSS Module Series
Document Type	Application Note
Document Status	Released

Revision History

Version	Date	Description
-	2019-07-26	Creation of the document
1.0	2019-07-26	First official release
1.1	2023-07-28	1. Numerous changes were made to this document. It should be read in its entirety. 2. Updated the applicable module from L26-DR to L26-DR (ADR).
1.2	2025-02-25	1. Deleted the information on CAN Bus mode. 2. Added DR application architecture (*Figure 1*). 3. Added a note about the recommended DR mode (*Chapter 1.1.1*). 4. Added DR application scenarios (*Chapter 1.1.2*). 5. Updated the pitch angle figure (*Figure 6*). 6. Updated vehicle speed injection modes (*Chapter 2.2*): ● Renamed the chapter from "DR Operating Modes (Speed & Reverse Data Acquisition" to "Vehicle Speed Injection"; ● Added notes about requirements for injecting speed information, enabling the PSTMDRSENMSG message output and odometer count; ● Added the requirement of injecting direction data in UART Speed Mode. 7. Updated the field indicating successful ADR calibration in

Version	Date	Description
		PSTMDRCAL (Chapter 2.4.3). 8. Updated the description of $PSTMDRCAL message and its parameters (Chapter 3.10). 9. Updated the <Reverse> field to <Direction> (Chapters 3.17.2 and 3.17.3). 10. Added acceleration calculation formula (Chapter 3.17.6). 11. Added angular rate calculation formula (Chapter 3.17.7). 12. Added DR operation guide (Chapter 4). 13. Added FAQs (Chapter 5).

Contents

Table Index

Figure Index

1 Introduction

This document describes the Dead Reckoning (DR) feature, including frame conventions, vehicle speed injection, mounting, calibration and messages related to the DR functionality of L26-DR (ADR) module, DR operation guide, and FAQs.

1.1. DR Overview

1.1.1. DR Introduction

The L26-DR (ADR) module supports Dead Reckoning (DR) function, which is a technology that combines inertial sensors with GNSS to seamlessly fuse satellite measurements, angular rate and acceleration data. It is designed to improve the accuracy, availability and integrity of positioning data for automotive users in environments with limited or no GNSS reception.

L26-DR (ADR) supports two DR modes: ADR (Automotive Dead Reckoning) and UDR (Untethered Dead Reckoning). In ADR mode, the module relies on speed data from the vehicle and the onboard 6-axis sensor for enhanced accuracy in environments with poor or non-existent GNSS coverage. On the other hand, UDR mode does not require speed data and is automatically activated by the firmware if no speed data is injected.

The module's DR function is supported by a MEMS Inertial Measurement Unit (IMU) sensor, which is a 6-axis inertial sensor (consisting of a 3-axis gyroscope and 3-axis accelerometer). In addition, the DR function may require information about vehicle speed, which is provided by WHEELTICK or UART. See *Chapter 2.2 Vehicle Speed Injection* for details.

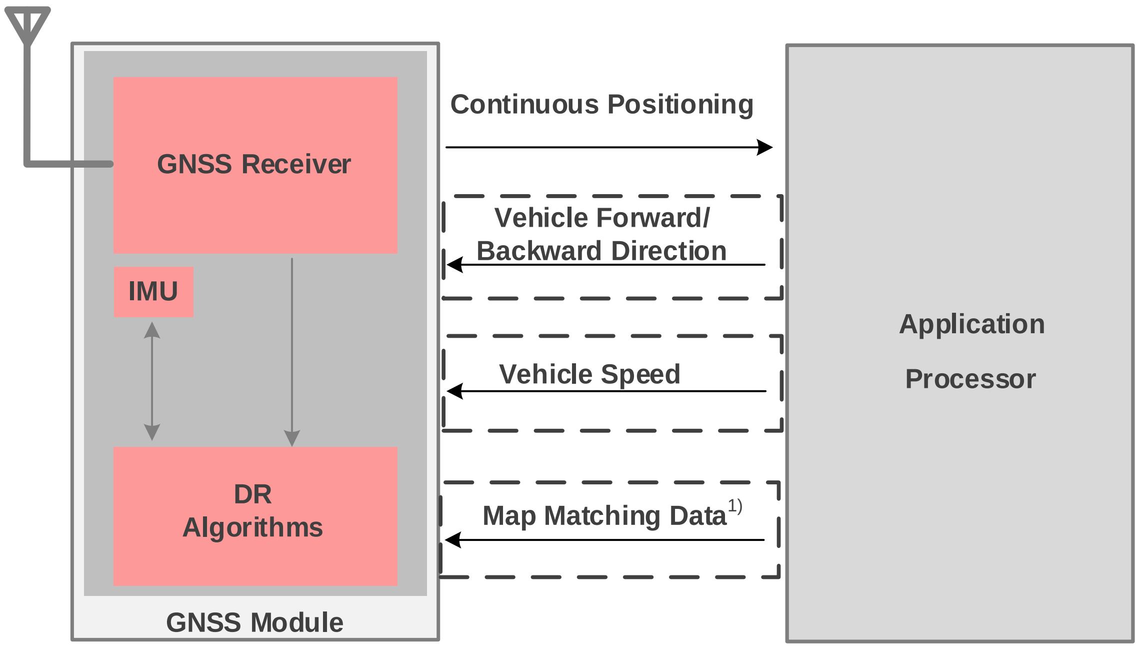


Figure 1: DR Application Architecture

NOTE

1. In case of temporary unavailability of speed signal, the system exploits the presence of accelerometer to seamlessly switch to UDR mode. This capability is called fallback mode and is triggered after the speed signal has been absent for 3 seconds. Once the speed signal becomes available again, the system automatically switches back to ADR mode.
2. DR performance cannot be guaranteed in UDR mode. For optimal performance with the module, please use ADR mode with access to vehicle speed.
3. [1)] The injection of map matching data is only for enabling the map matching feedback function of the module. For details, contact Quectel Technical Support (support@quectel.com).

1.1.2. DR Application Scenarios

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

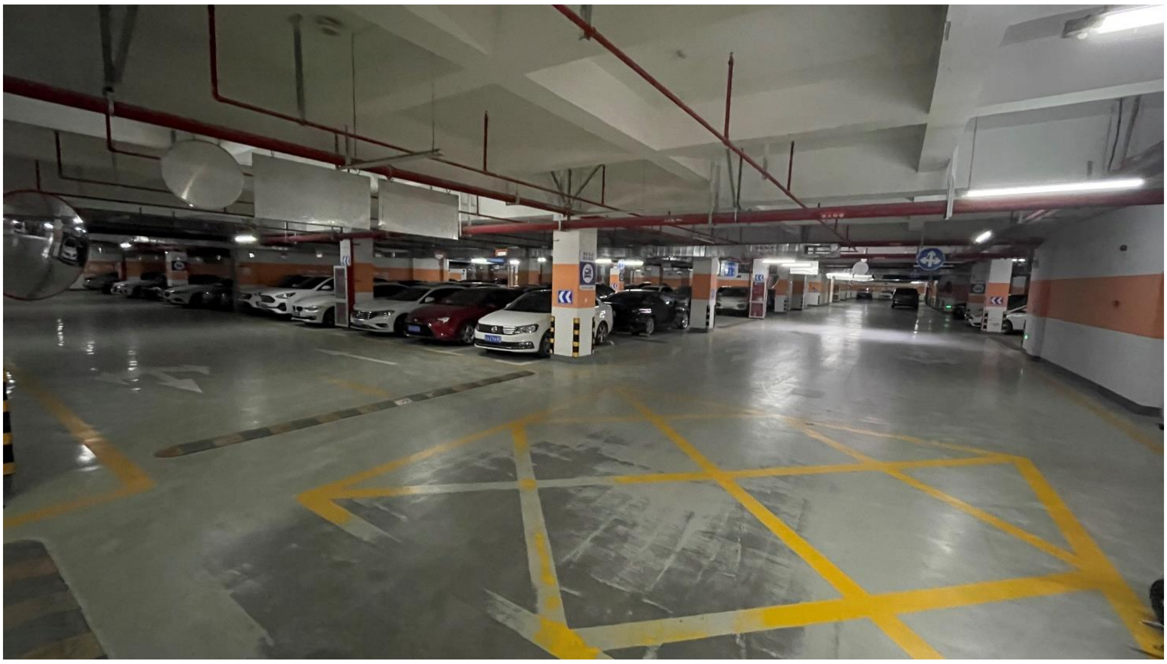


Figure 2: Underground Parking

2 DR Configuration

2.1. Frame Conventions

2.1.1. Vehicle Reference Frame

ADR and UDR internal processing assume that the IMU axes are aligned in a specific manner to the so-called vehicle reference frame. The vehicle reference frame is defined as follows:

- X axis: aligned with vehicle travel direction, pointing forward.
- Y axis: perpendicular to vehicle travel direction, pointing to the left.
- Z axis: perpendicular to vehicle travel direction, pointing upwards (vertical).

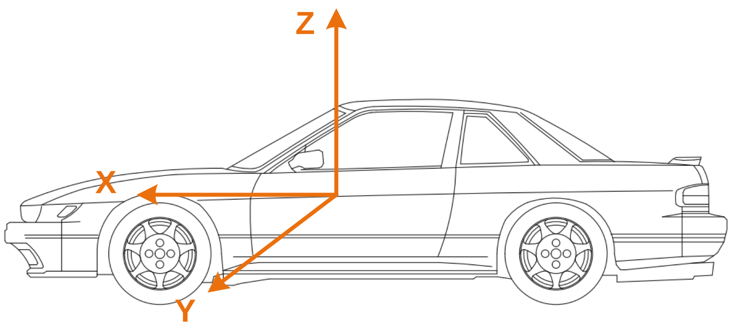


Figure 3: Vehicle Reference Frame

2.1.2. Navigation Reference Frame

The navigation reference frame is aligned with the geodetic reference frame and is defined as follows:

- X_n axis: aligned with vehicle travel direction excluding pitch, pointing forward.
- Y_n axis: perpendicular to vehicle travel direction excluding roll, pointing to the left.
- Z_n axis: perpendicular to the horizontal plane, pointing upwards.

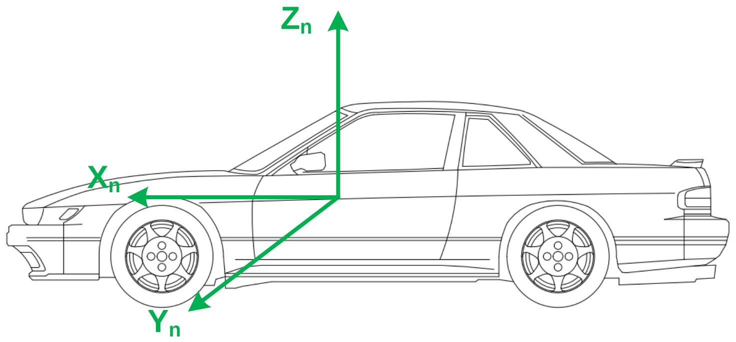


Figure 4: Navigation Reference Frame

The module reports the vehicle's position, velocity and attitude. The attitude is defined by 3 angles.

- **Yaw**, representing the mismatch between vehicle frame X axis and navigation frame X_n axis.
- **Pitch**, representing the mismatch between vehicle frame Z axis and navigation frame Z_n axis.
- **Roll**, representing the mismatch between vehicle frame Y axis and navigation frame Y_n axis.

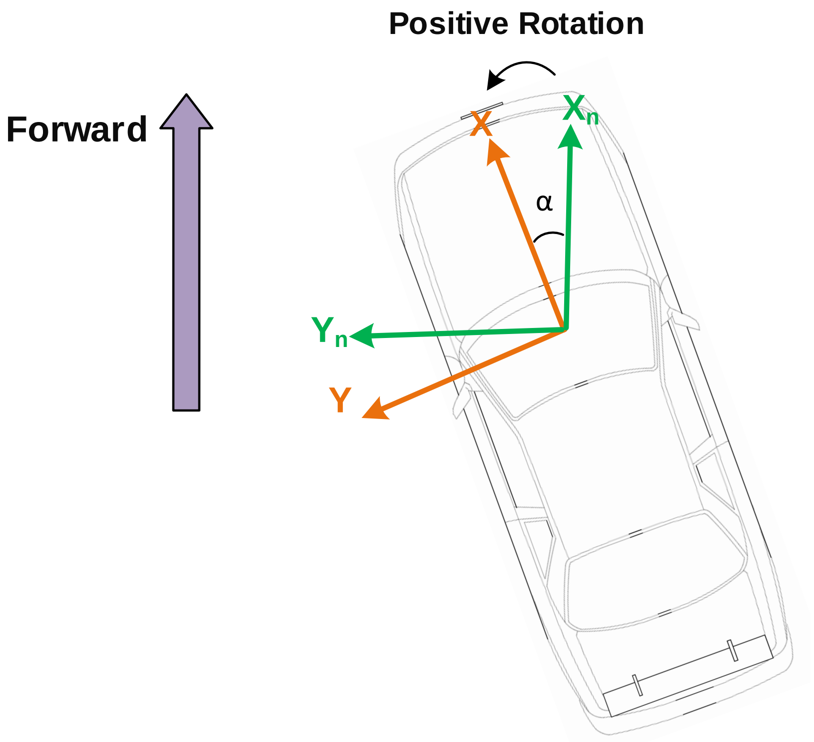


Figure 5: Yaw Angle

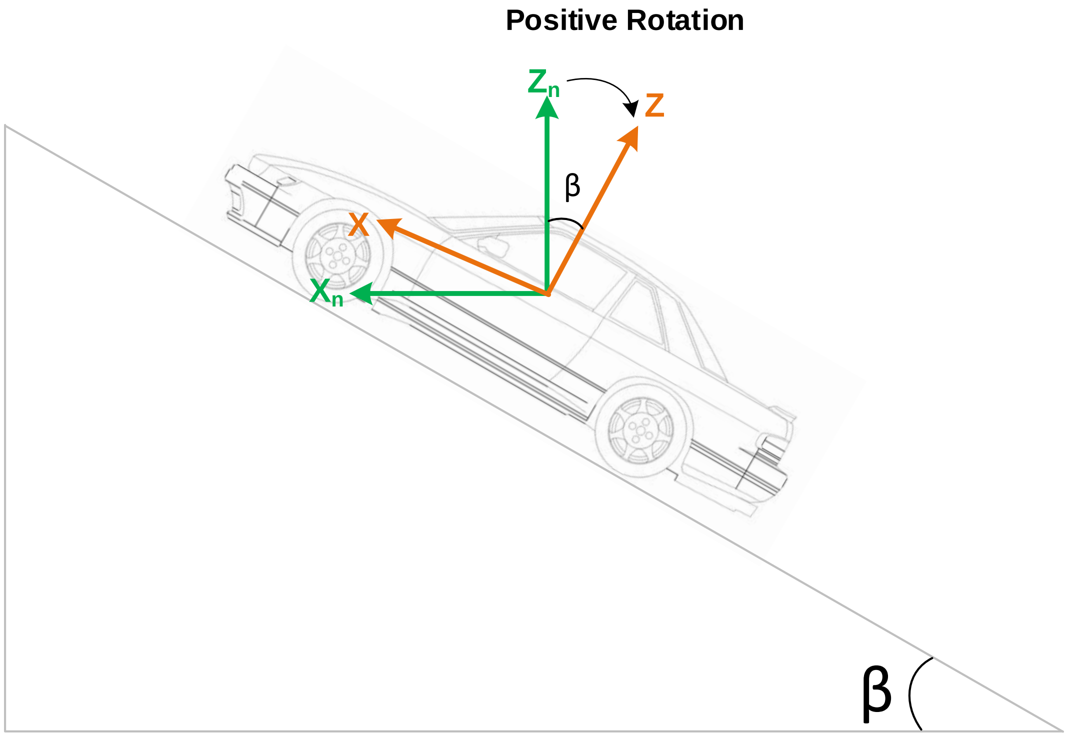


Figure 6: Pitch Angle

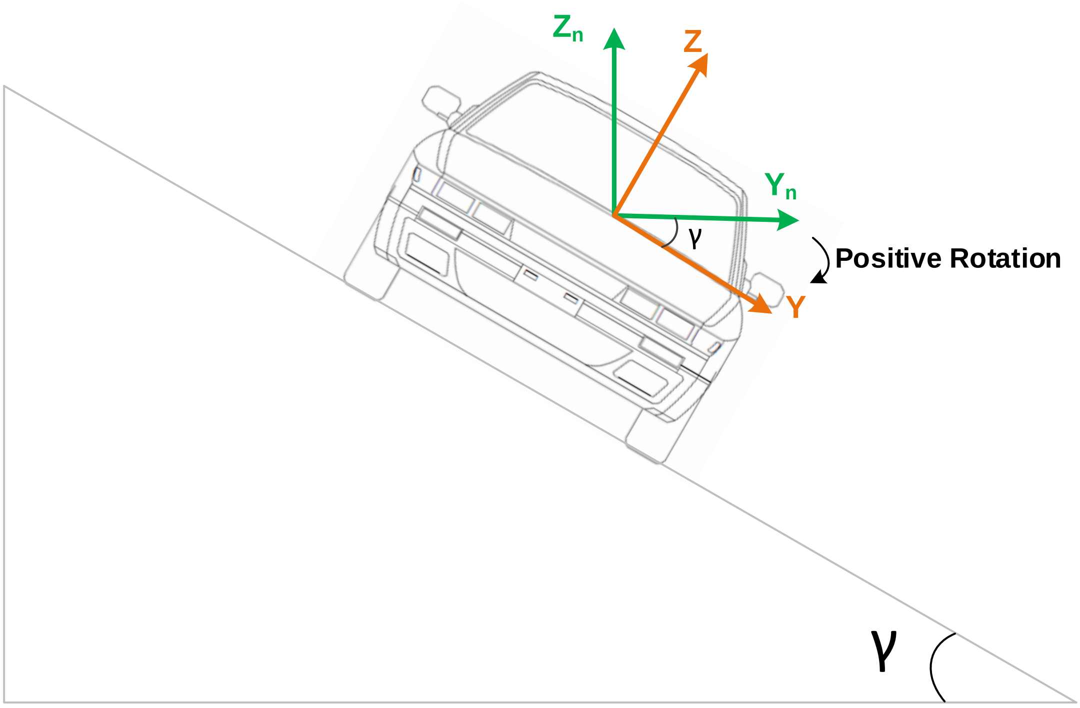


Figure 7: Roll Angle

2.1.3. Sensor Reference Frame

MEMS IMU outputs its raw data (angular rates and accelerations) in the sensor reference frame. L26-DR (ADR) is equipped with a 6-axis sensor, which contains a 3-axis accelerometer and a 3-axis gyroscope, and the axes for both sensors have the same direction in the sensor reference frame. The IMU sensor frame is illustrated below:

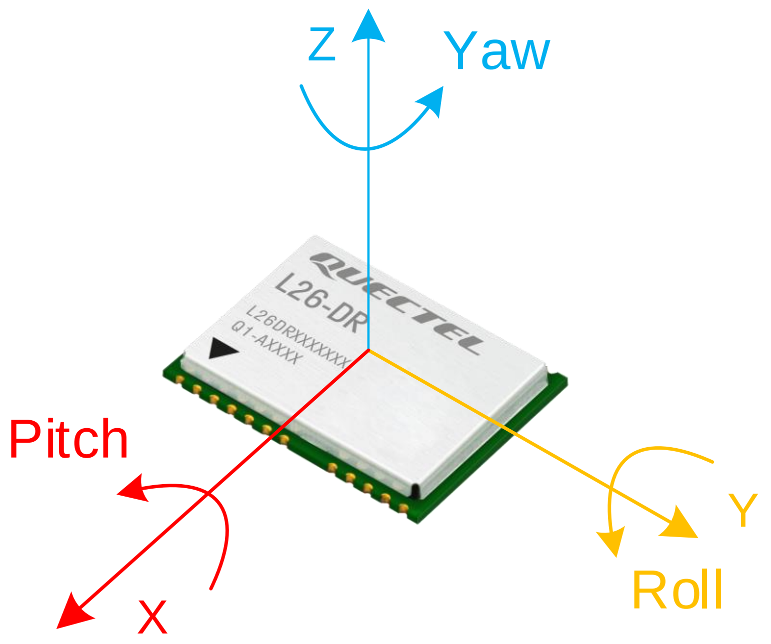


Figure 8: IMU Sensor Frame of L26-DR (ADR)

NOTE

The IMU sensor frame of L26-DR (ADR) 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 (ADR) 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 module can acquire speed and direction data in the following modes:

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

- UART Mode (including UART Odometer Mode and UART Speed Mode).

NOTE

$PSTMSETPAR can be used to enable Wheel Tick Mode or UART Mode and the output of **$PSTMDRSENMSG** message. For details about **$PSTMSETPAR**, contact Quectel Technical Support (support@quectel.com).

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*.

PSTM commands for switching to Wheel Tick Mode:

```
$PSTMSETPAR,1600,80010106*12
$PSTMSAVEPAR*58
$PSTMSRR*49
```

PSTM commands for enabling the output of $PSTMDRSENMSG:

```
$PSTMSETPAR,1228,10000000,1*0E
$PSTMSAVEPAR*58
$PSTMSRR*49
```

Result:

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

```
$PSTMDRSENMSG,3,<Timestamp>,<Odometer>,<Direction>*<Checksum><CR><LF>
```

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

For details about **$PSTMDRSENMSG**, refer to *Chapter 3.17.3 Odometer and Direction Message (<MsgID> = 3)*. For details about the **$PSTMSAVEPAR** and **$PSTMSRR** messages, refer to *document [2] protocol specification*.

NOTE

1. To ensure ADR performance, the pulse width must not exceed 0.4 meters/pulse.
2. Enabling the **$PSTMDRSENMSG** message output is not necessary for vehicle speed injection and ADR function use; once enabled, the **$PSTMDRSENMSG** message can be used to verify if the vehicle wheel tick pulse count and direction signals are successfully injected:
   - You can verify if pulse signal injection is successful by observing the **<Odometer>** 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 **<Direction>** 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 Odometer Mode

In this mode, the module gets vehicle odometer count and direction data via the UART interface.

PSTM commands for switching to UART Odometer Mode:

```
$PSTMSETPAR,1600,80010306*10
$PSTMSAVEPAR*58
$PSTMSRR*49
```

PSTM commands for enabling the output of $PSTMDRSENMSG:

```
$PSTMSETPAR,1228,10000000,1*0E
$PSTMSAVEPAR*58
$PSTMSRR*49
```

Odometer count and direction data sent to the module via UART:

```
$PSTMDRSENMSG,1,0,<Odometer>*<Checksum><CR><LF>
$PSTMDRSENMSG,2,0,<Direction>*<Checksum><CR><LF>
```

Result:

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

```
$PSTMDRSENMSG,1,<TimeStamp>,<Odometer>*<Checksum><CR><LF>
$PSTMDRSENMSG,2,<TimeStamp>,<Direction>*<Checksum><CR><LF>
```

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

For details about **$PSTMDRSENMSG**, refer to *Chapters 3.17.1 Odometer Message (<MsgID> = 1)* and *3.17.2 Direction Message (<MsgID> = 2)*. For details about the **$PSTMSAVEPAR** and **$PSTMSRR** messages, refer to *document [2] protocol specification*.

NOTE

1. To ensure ADR performance, the odometer scale must not exceed 0.4 meters/pulse.
2. In UART Odometer Mode, it is recommended to inject vehicle odometer count and direction data at

a frequency of at least 10 Hz.
3. Enabling the **$PSTMDRSENMSG** message output is not necessary for vehicle speed injection and ADR function use; once enabled, the **$PSTMDRSENMSG** message can be used to verify if the vehicle odometer count and direction signals are successfully injected.
4. Ensure that the baud rate of the host and module is consistent.
5. Odometer count refers to the cumulative count of pulses.

2.2.2.2. UART Speed Mode

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

PSTM commands for switching to UART Speed Mode:

```
$PSTMSETPAR,1600,80010506*16
$PSTMSAVEPAR*58
$PSTMSRR*49
```

PSTM commands for enabling the output of $PSTMDRSENMSG:

```
$PSTMSETPAR,1228,10000000,1*0E
$PSTMSAVEPAR*58
$PSTMSRR*49
```

Speed and direction data sent to the module via UART:

```
$PSTMDRSENMSG,14,0,<VehicleSpeed>*<Checksum><CR><LF>
$PSTMDRSENMSG,2,0,<Direction>*<Checksum><CR><LF>
```

Result:

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

```
$PSTMDRSENMSG,14,<TimeStamp>,<VehicleSpeed>*<Checksum><CR><LF>
$PSTMDRSENMSG,2,<TimeStamp>,<Direction>*<Checksum><CR><LF>
```

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

For details about **$PSTMDRSENMSG**, refer to *Chapter 3.17.4 Vehicle Speed Message (<MsgID> = 14)* and *3.17.2 Direction Message (<MsgID> = 2)*. For details about the **$PSTMSAVEPAR** and **$PSTMSRR** messages, refer to *document [2] protocol specification*.

NOTE

1. To ensure ADR performance, the injected speed value must be rounded to one decimal place.
2. In UART Speed Mode, it is recommended to inject vehicle speed and direction data at a frequency of at least 10 Hz.

3. Enabling the **$PSTMDRSENMSG** message output is not necessary for vehicle speed injection and ADR function use; once enabled, the **$PSTMDRSENMSG** message can be used to verify if the vehicle speed and direction signals are successfully injected.
4. Ensure that the baud rate of the host and module is consistent.

2.3. Mounting

L26-DR (ADR) module integrates an IMU and a GNSS receiver and is designed for four-wheel vehicles. 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 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.

DR firmware offers Full Free Mount (FFM) capability, i.e., it automatically estimates the device's installation and seamlessly compensates for any misalignments. For this reason, there is no specific requirement for the device's orientation in relation to the vehicle travel direction. However, a sequence of manoeuvres is required to allow the module to calibrate successfully. This sequence is described in *Chapter 2.4 DR Calibration*.

2.4. DR Calibration

This section describes the conditions (driving maneuvers and GNSS reception) needed at the beginning of a test session to calibrate ADR or UDR for the purpose of evaluation. The procedure is required to compensate misalignment (installation pitch, roll and yaw) and MEMS errors (IMU biases and sensitivity). The target is to lead the device in the so-called "system-ready" status, where the PVT output is given by the fusion of GNSS and sensor measurements. Until this condition is reached the system operates in GNSS-only mode. When the system-ready status is reached, calibration parameters are stored in Non-Volatile Memory. This means that at the next startup the device will start in system-ready mode (the calibration procedure will not be requested anymore).

NOTE

This procedure is required each time the system starts in not-ready condition, i.e. in the following cases:

- First test after device installation in vehicle.
- Firmware upgrade.
- Device removed from vehicle since the previous test.
- Any modifications have been made to the installation/orientation of the device.

2.4.1. Conditions

The calibration phase length mostly depends on GNSS signal quality and the dynamics encountered by the vehicle, i.e. dynamic driving actions including acceleration, deceleration, braking and fast turns. Therefore, to reduce the calibration phase length, it is recommended to perform it in a location where there are no restrictions on vehicle maneuvering. All calibration maneuvers must happen in benign GNSS signal conditions. The vehicle should be driven to an open and flat area such as an empty parking lot or road with an unobstructed view of the sky, i.e., avoiding areas with urban structures, forests, tunnels, or any other objects that may obstruct or reflect the GNSS signal. It is important to note that calibration success does not depend on its duration, but rather on how well the driving style and sky conditions match the recommended guidelines.

2.4.2. Guidelines

The calibration drive should involve different phases, including stationary periods where the car is stopped with engine turned on, normal left and right turns while driving, and periods of driving above 30 km/h under good GNSS reception conditions. Accelerations/braking during driving are beneficial to the process.

The following sequence is an example of good calibration procedure:

Step 1 Start with a short stationary section (at least 30 seconds).
Step 2 Drive for 10 minutes at sufficient speed (> 30 km/h), including acceleration and braking maneuvers.
Step 3 Perform several “fast” turns (circles, U-turns and corners) at sufficient speed (> 30 km/h).

NOTE

1. The calibration process typically takes about 15 minutes if the driving conditions reported above are met. The calibration process may take longer if the guidelines are not followed, e.g., if the vehicle moves slowly due to traffic.
2. Fast turns such as corners, roundabouts, circles and U-turns, where the road direction changes by 90 degrees or more in a few seconds, are recommended for calibration. Large turns like those on highways or lane changes are not useful. The same turn can be repeated several times to achieve the desired effect.

2.4.3. Calibration Result Monitoring

NMEA output contains data that enables understanding if the calibration phase is over and the system is ready for operation.

$PSTMDRCAL and **$PSTMDR2** proprietary NMEA messages provide useful information about calibration status. In ADR mode, the **<OdoIsCalib>** field in the **$PSTMDRCAL** message and the **<Sr>**

field in the **$PSTMDR2** message output by the module are both 1 to indicate that the system is fully calibrated and operational. In UDR mode, **<Sr>** field of **$PSTMDR2** should read 1 to indicate that the system is fully operational. For more details, refer to *Chapter 3.10 PSTMDRCAL* and *Chapter 3.12 PSTMDR2*.

2.4.4. Calibration Status Storage

It is important to save the position, heading alignment and calibration status to NVM at the end of each positioning session to ensure that the module can immediately enter the calibrated state in the following session. To save the information to NVM, the vehicle must remain stationary for at least 5 seconds before turning the unit off.

At the next startup, if the relative position of the module has not changed, device will start from the saved position and heading, guaranteeing accurate PVT output even before GNSS acquisition. If the relative position of the module has changed, it is recommended to clear the calibration parameters and re-perform the DR calibration process. See *Chapter 5 FAQs* for details.

3 PSTM Messages

This chapter explains PSTM messages (proprietary NMEA messages about DR function defined by the chipset supplier) supported by Quectel L26-DR (ADR) module. Output messages (except **$PSTMDRSENMSG** and **$PSTMDRSENCONFIG**) are provided at 1 Hz.

NOTE

1. Use **$PSTMSETPAR** to enable/disable the output of messages mentioned in this chapter. For details about **$PSTMSETPAR**, contact Quectel Technical Support (support@quectel.com).
2. After issuing the enable/disable command (**$PSTMSETPAR**), you must save the configuration (using **$PSTMSAVEPAR**) and reset the system (using **$PSTMSRR**) for the command to take effect.

3.1. PSTMDRCONFID

Reports the error estimations (standard deviation) for navigation and calibration estimates. The message is only supported in ADR mode.

Type:

Output

Synopsis:

```
$PSTMDRCONFID,<LatStdDev>,<LonStdDev>,<HeadingStdDev>,<Reserved>,<GyroBiasStdDev>,<OdoScaleStdDev>,<Reserved>,<AccOffsetStdDev>,<HeightStdDev>,<MajorSemiAxis>,<MinorSemiAxis>,<EllipseAngle>,<SpeedStdDev>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<LatStdDev>	Numeric	Meter	Latitude standard deviation
<LonStdDev>	Numeric	Meter	Longitude standard deviation
<HeadingStdDev>	Numeric	Degree	Heading standard deviation

Field	Format	Unit	Description
<Reserved>	-	-	Reserved for future use
<GyroBiasStdDev>	Numeric	Millivolt	Gyro bias standard deviation
<OdoScaleStdDev>	Numeric	Millimeter/pulse	Odometer scale standard deviation
<Reserved>	-	-	Reserved for future use
<AccOffsetStdDev>	Numeric	g	Accelerometer offset standard deviation
<HeightStdDev>	Numeric	Meter	Height standard deviation
<MajorSemiAxis>	Numeric	Meter	Major semi axis of 1 sigma error ellipse
<MinorSemiAxis>	Numeric	Meter	Minor semi axis of 1 sigma error ellipse
<EllipseAngle>	Numeric	Degree	Angle vs North of 1 sigma error ellipse
<SpeedStdDev>	Numeric	m/s	Speed standard deviation

Example:

$PSTMDRCONFID,100.00,100.00,180.0419,0.1004,0.00,0.0857,0.000993,0.050002,100.3594,100.0,100.0,90.0,0.0*0A

NOTE

Commands to enable or disable the output of **$PSTMDRCONFID** are shown below.
//Enable **$PSTMDRCONFID**:
$PSTMSETPAR,1228,40000000,1*0B
$PSTMSAVEPAR*58
$PSTMSRR*49

//Disable **$PSTMDRCONFID**:
$PSTMSETPAR,1228,40000000,2*08
$PSTMSAVEPAR*58
$PSTMSRR*49

3.2. PSTMDRBSD

Reports the standard deviation of estimates for IMU biases. The message is only supported in UDR mode.

Type:

Output

Synopsis:

```
$PSTMDRBSD,<TimeStamp>,<CPUTime>,<Gyro_X_Bias_SD>,<Gyro_Y_Bias_SD>,<Gyro_Z_Bias_SD>,<Acc_X_Bias_SD>,<Acc_Y_Bias_SD>,<Acc_Z_Bias_SD>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<TimeStamp>	hhmmss.sss	-	Position fix timestamp (UTC). hh: Hours. Range: 00–23. mm: Minutes. Range: 00–59. ss: Seconds. Range: 00–59. sss: Decimal fraction of seconds.
<CPUTime>	Numeric	Microsecond	Time of DR estimation (CPU ticks).
<Gyro_X_Bias_SD>	Numeric	dps	Gyro X bias standard deviation.
<Gyro_Y_Bias_SD>	Numeric	dps	Gyro Y bias standard deviation.
<Gyro_Z_Bias_SD>	Numeric	dps	Gyro Z bias standard deviation.
<Acc_X_Bias_SD>	Numeric	m/s^2	Accelerometer X bias standard deviation.
<Acc_Y_Bias_SD>	Numeric	m/s^2	Accelerometer Y bias standard deviation.
<Acc_Z_Bias_SD>	Numeric	m/s^2	Accelerometer Z bias standard deviation.

Example:

```
$PSTMDRBSD,073922.000,1837870321,0.058,0.058,0.019,0.078,0.077,0.058*42
```

NOTE

Commands to enable or disable the output of **$PSTMDRBSD** are shown below.

//Enable **$PSTMDRBSD**:

$PSTMSETPAR,1228,40000000,1*0B
$PSTMSAVEPAR*58
$PSTMSRR*49

//Disable **$PSTMDRBSD**:

$PSTMSETPAR,1228,40000000,2*08
$PSTMSAVEPAR*58

$PSTMSRR*49

3.3. PSTMDRGPS

Reports relevant GNSS information. It includes position and velocity data, as well as quality metrics related to GNSS measurements and constellation used.

Type:

Output

Synopsis:

$PSTMDRGPS,<Lat>,<Lon>,<Vn>,<Ve>,<PDOP>,<HDOP>,<VDOP>,<RMS_PosResidual>,<RMS_VelResidual>,<V_Up>,<Height>*<Checksum><CR><LF>

Parameter:

Field	Format	Unit	Description
<Lat>	Numeric	Degree	GNSS latitude.
<Lon>	Numeric	Degree	GNSS longitude.
<Vn>	Numeric	m/s	GNSS velocity north component.
<Ve>	Numeric	m/s	GNSS velocity east component.
<PDOP>	Numeric	-	3D position dilution of precision of used GNSS constellation. Maximum value: 99.000.
<HDOP>	Numeric	-	Horizontal position dilution of precision of used GNSS constellation. Maximum value: 99.000.
<VDOP>	Numeric	-	Vertical position dilution of precision of used GNSS constellation. Maximum value: 99.000.
<RMS_PosResidual>	Numeric	Meter	RMS error on GNSS pseudo range measurements.
<RMS_VelResidual>	Numeric	m/s	RMS error on GNSS frequency measurements.
<V_Up>	Numeric	m/s	GNSS velocity up component.
<Height>	Numeric	Meter	GNSS altitude.

Example:

$PSTMDRGPS,31.853778470,117.253363350,0.69824,-3.58118,1.254,0.793,0.969,30.811,0.520,-0.07642,28.7*76

NOTE

Commands to enable or disable the output of **$PSTMDRGPS** are shown below.
//Enable **$PSTMDRGPS**:
$PSTMSETPAR,1228,800000,1*07
$PSTMSAVEPAR*58
$PSTMSRR*49

//Disable **$PSTMDRGPS**:
$PSTMSETPAR,1228,800000,2*04
$PSTMSAVEPAR*58
$PSTMSRR*49

3.4. PSTMDRPVA

Reports the position, velocity and attitude estimated by DR.

Type:

Output

Synopsis:

$PSTMDRPVA,<TimeStamp>,<CPUTime>,<Latitude>,<Longitude>,<Height>,<V_North>,<V_East>,<V_Up>,<Pitch>,<Roll>,<Heading>*<Checksum><CR><LF>

Parameter:

Field	Format	Unit	Description
<TimeStamp>	hhmmss.sss	-	Position fix timestamp (UTC). hh: Hours. Range: 00–23 mm: Minutes. Range: 00–59 ss: Seconds. Range: 00–59 sss: Decimal fraction of seconds
<CPUTime>	Numeric	Microsecond	Time of DR estimation (CPU ticks).
<Latitude>	Numeric	Degree	DR latitude.

Field	Format	Unit	Description
<Longitude>	Numeric	Degree	DR Longitude.
<Height>	Numeric	Meter	DR height.
<V_North>	Numeric	m/s	DR velocity north component.
<V_East>	Numeric	m/s	DR velocity east component.
<V_Up>	Numeric	m/s	DR velocity up component.
<Pitch>	Numeric	Degree	DR pitch angle. Range: -90.00 to 90.00.
<Roll>	Numeric	Degree	DR roll angle. Range: -180.00 to 180.00
<Heading>	Numeric	Degree	DR heading angle. Range: 0.00 to 360.00

Example:

$PSTMDRPVA,064420.000,24295094,31.8222713,117.1157794,33.1,0.00,0.00,0.21,1.64,-2.10,26.61*48

NOTE

Commands to enable or disable the output of **$PSTMDRPVA** are shown below.
//Enable **$PSTMDRPVA**:
$PSTMSETPAR,1228,4000000,1*3B
$PSTMSAVEPAR*58
$PSTMSRR*49

//Disable **$PSTMDRPVA**:
$PSTMSETPAR,1228,4000000,2*38
$PSTMSAVEPAR*58
$PSTMSRR*49

3.5. PSTMDRPVASD

Reports the standard deviation of DR estimates for position, velocity and attitude.

Type:

Output

Synopsis:

```
$PSTMDRPVASD,<TimeStamp>,<CPUTime>,<Latitude_SD>,<Longitude_SD>,<Height_SD>,<V_North_SD>,<V_East_SD>,<V_Up_SD>,<Pitch_SD>,<Roll_SD>,<Heading_SD>,<P_NE_SD>,<P_NU_SD>,<P_EU_SD>,<V_NE_SD>,<V_NU_SD>,<V_EU_SD>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<TimeStamp>	hhmmss.sss	-	Position fix timestamp (UTC). hh: Hours. Range: 00–23. mm: Minutes. Range: 00–59. ss: Seconds. Range: 00–59. sss: Decimal fraction of seconds.
<CPUTime>	Numeric	Microsecond	Time of DR estimation (CPU ticks).
<Latitude_SD>	Numeric	Degree	DR latitude standard deviation.
<Longitude_SD>	Numeric	Degree	DR longitude standard deviation.
<Height_SD>	Numeric	Meter	DR height standard deviation.
<V_North_SD>	Numeric	m/s	DR velocity north component standard deviation.
<V_East_SD>	Numeric	m/s	DR velocity east component standard deviation.
<V_Up_SD>	Numeric	m/s	DR velocity up component standard deviation.
<Pitch_SD>	Numeric	Degree	DR pitch angle standard deviation.
<Roll_SD>	Numeric	Degree	DR roll angle standard deviation.
<Heading_SD>	Numeric	Degree	DR heading angle standard deviation.
<P_NE_SD>	Numeric	Meter	DR position north east standard deviation.
<P_NU_SD>	Numeric	Meter	DR position north up standard deviation.
<P_EU_SD>	Numeric	Meter	DR position east up standard deviation.
<V_NE_SD>	Numeric	m/s	DR velocity north east standard deviation.
<V_NU_SD>	Numeric	m/s	DR velocity north up standard deviation.
<V_EU_SD>	Numeric	m/s	DR velocity east up standard deviation.

Example:

```
$PSTMDRPVASD,064420.000,24295094,17.9,22.0,13.5,2.83,1.77,1.80,0.00,0.00,180.01,11.5,7.2,8.9,1.32,0.61,0.79*78
```

NOTE

Commands to enable or disable the output of **$PSTMDRPVASD** are shown below.
//Enable **$PSTMDRPVASD**:
$PSTMSETPAR,1228,2000000,1*3D
$PSTMSAVEPAR*58
$PSTMSRR*49

//Disable **$PSTMDRPVASD**:
$PSTMSETPAR,1228,2000000,2*3E
$PSTMSAVEPAR*58
$PSTMSRR*49

3.6. PSTMDRSINT

Reports the result for sensors' integration samples between two consecutive GNSS epochs, i.e. over 1 second.

Type:

Output

Synopsis:

```
$PSTMDRSINT,<GyroSampleCount>,<AccSampleCount>,<PresSampleCount>,<OdometerCount>,<GyroIntTime>,<AccIntTime>,<PressureIntTime>,<Gyro_X_Avg>,<Gyro_Y_Avg>,<Gyro_Z_Avg>,<Acc_X_Avg>,<Acc_Y_Avg>,<Acc_Z_Avg>,<PresAvg>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<GyroSampleCount>	Numeric	-	Number of received gyro samples.
<AccSampleCount>	Numeric	-	Number of received accelerometer samples.
<PresSampleCount>	Numeric	-	Number of received pressure samples.
<OdometerCount>	Numeric	-	Number of received odometer pulses.

Field	Format	Unit	Description
<GyroIntTime>	Numeric	Second	Elapsed time between the receipt of 1st and last gyro sample.
<AccIntTime>	Numeric	Second	Elapsed time between the receipt of 1st and last accelerometer sample.
<PressureIntTime>	Numeric	Second	Elapsed time between the receipt of 1st and last pressure sample.
<Gyro_X_Avg>	Numeric	dps	Average of X gyro samples received in last second, expressed in vehicle frame.
<Gyro_Y_Avg>	Numeric	dps	Average of Y gyro samples received in last second, expressed in vehicle frame.
<Gyro_Z_Avg>	Numeric	dps	Average of Z gyro samples received in last second, expressed in vehicle frame.
<Acc_X_Avg>	Numeric	m/s^2	Average of X accelerometer samples received in last second, expressed in vehicle frame.
<Acc_Y_Avg>	Numeric	m/s^2	Average of Y accelerometer samples received in last second, expressed in vehicle frame.
<Acc_Z_Avg>	Numeric	m/s^2	Average of Z accelerometer samples received in last second, expressed in vehicle frame.
<PresAvg>	Numeric	hPa	Average of received pressure samples.

Example:

$PSTMDRSINT,100,100,0,0,1.000,1.000,0.000,-0.42,0.79,-0.07,0.26,0.35,10.05,0.00*05

NOTE

Commands to enable or disable the output of **$PSTMDRSINT** are shown below.
//Enable **$PSTMDRSINT**:
$PSTMSETPAR,1228,800000,1*07
$PSTMSAVEPAR*58
$PSTMSRR*49

//Disable **$PSTMDRSINT**:
$PSTMSETPAR,1228,800000,2*04
$PSTMSAVEPAR*58
$PSTMSRR*49

3.7. PSTMDRSVF

Reports vehicle dynamics (accelerations and angular rates) measured by the IMU. The measurements are illustrated in *Chapter 2.1.1 Vehicle Reference Frame*.

Type:

Output

Synopsis:

$PSTMDRSVF,<TimeStamp>,<CPUTime>,<X_Acceleration>,<Y_Acceleration>,<Z_Acceleration>,<X_AngularRate>,<Y_AngularRate>,<Z_AngularRate>*<Checksum><CR><LF>

Parameter:

Field	Format	Unit	Description
<TimeStamp>	hhmmss.sss	-	Position fix timestamp (UTC). hh: Hours. Range: 00–23. mm: Minutes. Range: 00–59. ss: Seconds. Range: 00–59. sss: Decimal fraction of seconds.
<CPUTime>	Numeric	Microsecond	Time of estimation (CPU ticks).
<X_Acceleration>	Numeric	m/s^2	Vehicle acceleration in X direction.
<Y_Acceleration>	Numeric	m/s^2	Vehicle acceleration in Y direction.
<Z_Acceleration>	Numeric	m/s^2	Vehicle acceleration in Z direction.
<X_AngularRate>	Numeric	dps	Vehicle angular rate in X direction.
<Y_AngularRate>	Numeric	dps	Vehicle angular rate in Y direction.
<Z_AngularRate>	Numeric	dps	Vehicle angular rate in Z direction.

Example:

$PSTMDRSVF,064420.000,24295094,0.00,0.00,0.00,0.00,0.00,0.00*50

NOTE

Commands to enable or disable the output of **$PSTMDRSVF** are shown below.
//Enable **$PSTMDRSVF**:
$PSTMSETPAR,1228,200,1*3D

$PSTMSAVEPAR*58
$PSTMSRR*49

//Disable **$PSTMDRSVF**:
$PSTMSETPAR,1228,200,2*3E
$PSTMSAVEPAR*58
$PSTMSRR*49

3.8. PSTMDRUPD

This message outputs information for Quectel's internal debugging purpose only.

3.9. PSTMDRSTYPE

Reports DR sensor configuration (combination) type.

Type:

Output

Synopsis:

$PSTMDRSTYPE,<SensorType>*<Checksum><CR><LF>

Parameter:

Field	Format	Unit	Description
<SensorType>	Numeric	-	Type of sensor for generating yaw value. 3 = GYRO3, meaning yaw is calculated from a 3D MEMS gyro.

Example:

$PSTMDRSTYPE,3*58

NOTE

Commands to enable or disable the output of **$PSTMDRSTYPE** are shown below.
//Enable **$PSTMDRSTYPE**:
$PSTMSETPAR,1228,2000000,1*3D

```
$PSTMSAVEPAR*58
$PSTMSRR*49

//Disable $PSTMDRSTYPE:
$PSTMSETPAR,1228,2000000,2*3E
$PSTMSAVEPAR*58
$PSTMSRR*49
```

3.10. PSTMDRCAL

Reports odometer calibration status (other calibration fields are not indicative). The message is only supported in ADR mode.

Type:

Output

Synopsis:

$PSTMDRCAL,<DR_IsCalib>,<OdoIsCalib>,<GyroSensitivityIsCalib>,<GyroBiasIsCalib>,<IMU_Flag>,<GyroIntegrityFlag>,<AccIntegrity>,<CalState>*<Checksum><CR><LF>

Parameter:

Field	Format	Unit	Description
<DR_IsCalib>	Numeric	-	DR calibration status. Reserved. 0 = DR is not calibrated. 1 = DR is calibrated.
<OdoIsCalib>	Numeric	-	Odometer scale calibration status. 0 = Odometer scale is not calibrated. 1 = Odometer scale is calibrated.
<GyroSensitivityIsCalib>	Numeric	-	Gyro sensitivity calibration status. Reserved. 0 = Gyro sensitivity is not calibrated. 1 = Gyro sensitivity is calibrated.
<GyroBiasIsCalib>	Numeric	-	Gyro offset calibration status. Reserved. 0 = Gyro offset is not calibrated. 1 = Gyro offset is calibrated.
<IMU_Flag>	Hexadecimal	-	Bit fields. Reserved. Bits 0–5: IMU calibration status flags. They can only be all 0 or all 1. 0 = Not calibrated 1 = Calibrated

Field	Format	Unit	Description
			Bits 6 and 7: IMU installation status flags. Bit 6 refers to roll installation angle and bit 7 pitch installation angle. 0 = Not self-detected 1 = Self-detected
<GyroIntegrityFlag>	Numeric	-	Gyro signal status flags. Reserved. 0 = Gyro signal is faulty. 1 = Gyro signal is healthy.
<AccIntegrity>	Numeric	-	Acceleration signal status flags. Reserved. 0 = Acceleration signal is faulty. 1 = Acceleration signal is healthy.
<CalState>	Character	-	DR calibration state. Reserved. N =Not calibrated. L = DR is lightly calibrated. F = DR is fully calibrated.

Example:

$PSTMDRCAL,1,1,1,1,ff,1,1,F*04

NOTE

Commands to enable or disable the output of **$PSTMDRCAL** are shown below.
//Enable **$PSTMDRCAL**:
$PSTMSETPAR,1228,800000,1*07
$PSTMSAVEPAR*58
$PSTMSRR*49

//Disable **$PSTMDRCAL**:
$PSTMSETPAR,1228,800000,2*04
$PSTMSAVEPAR*58
$PSTMSRR*49

3.11. PSTMDR1

Reports the details about alignment angles, bias and sensitivity values for gyroscope and accelerometer.

Type:

Output

Synopsis:

```
$PSTMDR1,<M_Pitch>,<M_Roll>,<M_Yaw>,<Gsz>,<Gbx>,<Gby>,<Gbz>,<Abx>,<Aby>,<Abz>,<OdometerScale>,<Res1Baro>,<Res2Baro>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<M_Pitch>	Numeric	Degree	Misalignment sensor vs. vehicle frame – Pitch angle
<M_Roll>	Numeric	Degree	Misalignment sensor vs. vehicle frame – Roll angle
<M_Yaw>	Numeric	Degree	Misalignment sensor vs. vehicle frame – Yaw angle
<Gsz>	Numeric	-	Gyroscope Z axis sensitivity
<Gbx>	Numeric	dps	Gyroscope X axis bias
<Gby>	Numeric	dps	Gyroscope Y axis bias
<Gbz>	Numeric	dps	Gyroscope Z axis bias
<Abx>	Numeric	m/s^2	Accelerometer X axis bias
<Aby>	Numeric	m/s^2	Accelerometer Y axis bias
<Abz>	Numeric	m/s^2	Accelerometer Z axis bias
<OdometerScale>	Numeric	Meter/Pulse	Odometer scale
<Res1Baro>	Numeric	-	Reserved for barometer
<Res2Baro>	Numeric	-	Reserved for barometer

Example:

```
$PSTMDR1,-1.63,-1.67,-91.75,1.009,-0.754,-0.197,0.118,-0.029,0.254,0.176,0.081,0.000,0.000*39
```

NOTES

Commands to enable or disable the output of **$PSTMDR1** are shown below.
//Enable **$PSTMDR1**:
$PSTMSETPAR,1228,20000000,1*0D
$PSTMSAVEPAR*58
$PSTMSRR*49

//Disable **$PSTMDR1**:
$PSTMSETPAR,1228,20000000,2*0E

```
$PSTMSAVEPAR*58
$PSTMSRR*49
```

3.12. PSTMDR2

Reports calibration status and availability.

Type:

Output

Synopsis:

```
$PSTMDR2,<IMU_Cal>,<AS_Cal>,<MotionStatus>,<ErrCode>,<Sr>,<CrossTrackError>,<AlongTrackError>,<Sa>,<VroWs>,<VroSt>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<IMU_Cal>	String	-	6-axis IMU calibration status. The first character refers to accelerometer status and the second gyroscope status. A = Not Available (sensor not supported) N = Not calibrated C = Calibrated
<AS_Cal>	String	-	Additional sensor calibration status. Fix as AA. A = Not Available (sensor not supported) N = Not calibrated C = Calibrated
<MotionStatus>	Numeric	-	Motion status indicator. 0 = Unclassified 1 = Stopped 2 = Moving straight 3 = Not moving straight or turning 4 = Turning 5 = Accelerating 6 = Braking 7 = Moving in backward
<ErrCode>	Numeric	-	FW error code. 0 = No error 1 = Gyroscope failure 2 = Accelerometer failure

Field	Format	Unit	Description
			3 = Odometer failure 4 = GNSS failure 5 = Pressure sensor failure 6 = Magnetometer failure 7 = Misalignment failure 8 = Accelerometer KF failure 9 = Gyro KF failure 10 = Generic software failure
<Sr>	Numeric	-	System ready flag. 0 = System not ready (GNSS-only PVT output) 1 = System ready (PVT output from GNSS sensor fusion)
<CrossTrackError>	Numeric	Meter	Cross-track error: compared from GNSS-only mode
<AlongTrackError>	Numeric	Meter	Along-track error: compared from GNSS-only mode
<Sa>	Numeric	-	System aligned flag. 0 = System not aligned (Full Free Mount procedure is not finished) 1 = System aligned (Full Free Mount procedure has been completed successfully)
<VroWs>	Numeric	Meter	Estimated vehicle tire circumference
<VroSt>	Numeric	-	VRO flag. 0 = Not started 1 = Search ongoing 2 = Speed available 3 = Speed not available for weak signal

Example:

$PSTMDR2,CC,AA,2,0,1,0.48,2.23,1,0.000,0*2D

NOTE

Commands to enable or disable the output of **$PSTMDR2** are shown below.
//Enable **$PSTMDR2**:
$PSTMSETPAR,1228,20000000,1*0D
$PSTMSAVEPAR*58
$PSTMSRR*49

//Disable **$PSTMDR2**:
$PSTMSETPAR,1228,20000000,2*0E

$PSTMSAVEPAR*58
$PSTMSRR*49

3.13. PSTMDREPE

Reports the error of DR estimated horizontal position.

Type:

Output

Synopsis:

$PSTMDREPE,<EHPE>,<Reserved>*<Checksum><CR><LF>

Parameter:

Field	Format	Unit	Description
<EHPE>	Numeric	Meter	Error of DR estimated horizontal position.
<Reserved>	Numeric	-	Fixed as -1.00. Reserved for future use.

Example:

$PSTMDREPE,141.42,-1.00*72

NOTE

Commands to enable or disable the output of **$PSTMDREPE** are shown below.
//Enable **$PSTMDREPE**:
$PSTMSETPAR,1228,2000000,1*3D
$PSTMSAVEPAR*58
$PSTMSRR*49

//Disable **$PSTMDREPE**:
$PSTMSETPAR,1228,2000000,2*3E
$PSTMSAVEPAR*58
$PSTMSRR*49

3.14. PSTMDRDEBUG

Reports DR debug information consisting of the following 6 error values that contain the differences between the GNSS Kalman and DR Kalman calculated values for key parameters.

Type:

Output

Synopsis:

$PSTMDRDEBUG,<LatError>,<LonError>,<HeadingError>,<SpeedError>,<HeightError>,<VvError>*<Checksum><CR><LF>

Parameter:

Field	Format	Unit	Description
<LatError>	Numeric	Meter	Latitude error
<LonError>	Numeric	Meter	Longitude error
<HeadingError>	Numeric	Degree	Heading error
<SpeedError>	Numeric	m/s	Speed error
<HeightError>	Numeric	Meter	Height error
<VvError>	Numeric	m/s	Rise speed error

Example:

$PSTMDRDEBUG,0.6,-2.0,-8.7,0.0,10.5,0.0*62

NOTE

Commands to enable or disable the output of **$PSTMDRDEBUG** are shown below.
//Enable **$PSTMDRDEBUG**:
$PSTMSETPAR,1228,2000000,1*3D
$PSTMSAVEPAR*58
$PSTMSRR*49

//Disable **$PSTMDRDEBUG**:
$PSTMSETPAR,1228,2000000,2*3E
$PSTMSAVEPAR*58
$PSTMSRR*49

3.15. PSTMDRSENCONFIG

Reports data about sensor internal configuration. This message is output at system startup.

Type:

Output

Synopsis:

$PSTMDRSENCONFIG,<IMU_Type>,<SensorName>,<FullScale>,<Sensitivity>,<OutputDataRate>,<LowPassFilterBandwidth>,<TemperatureCompensation>*<Checksum><CR><LF>

Parameter:

Field	Format	Unit	Description
<IMU_Type>	String	-	Type of reported sensor. IMU ACC: Accelerometer IMU GYRO: Gyroscope
<SensorName>	String	-	Sensor name. Fixed as LSM6DSR.
<FullScale>	Numeric	g or dps	Maximum sensible value. Default value for accelerometer (unit: g): 2. Default value for gyroscope (unit: dps): 125.
<Sensitivity>	Numeric	g/LSB or dps/LSB	Nominal sensitivity value. Default value for accelerometer (unit: g/LSB): 61. Default value for gyroscope (unit: dps/LSB): 4375.
<OutputDataRate>	Numeric	Hz	Digital data output rate. Default value: 104.
<LowPassFilterBandwidth>	Numeric	Hz	Bandwidth of sensor's analog low-pass filter. Default value for accelerometer: 52.0. Default value for gyroscope: 34.0.
<TemperatureCompensation>	Numeric	-	Status of the sensors' embedded temperature compensation block. 0 = Disabled 1 = Enabled This field is only present when **<IMU_Type>** is IMU Gyro.

Example:

$PSTMDRSENCONFIG,IMU GYRO,LSM6DSR,125,4375,104,34.0,1*0F
$PSTMDRSENCONFIG,IMU ACC,LSM6DSR,2,61,104,52.0*56

3.16. 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

Results:

- 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 not supported by the FW (sensor layer not present in FW), the module returns:

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

Example:

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

3.17. PSTMDRSENMSG

Outputs the IMU raw data and inputs/outputs vehicle speed and direction data

Type:

Output/Input

Synopsis:

```
$PSTMDRSENMSG,<MsgID>[<Par1>,…,<ParN>]*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<MsgID>	Numeric	-	Message ID. 1 = Odometer 2 = Direction 3 = Odometer and direction 14 = Vehicle speed 24 = IMU temperature 30 = IMU accelerometer 31 = IMU gyroscope
<Par1> to <ParN>	Numeric	-	This field varies with the message type. See *Chapters 3.17.1 Odometer Message (<MsgID> = 1)*, *3.17.2 Direction Message (<MsgID> = 2)*, *3.17.3 Odometer and Direction Message (<MsgID> = 3)*, *3.17.4 Vehicle Speed Message (<MsgID> = 14)*, *3.17.5 IMU Temperature Message (<MsgID> = 24)*, *3.17.6 IMU Accelerometer Message (<MsgID> = 30)* and *3.17.7 IMU Gyroscope Message (<MsgID> = 31)* for details.

NOTE

Commands to enable or disable the output of **$PSTMDRSENMSG** are shown below.
//Enable **$PSTMDRSENMSG**:
$PSTMSETPAR,1228,10000000,1*0E
$PSTMSAVEPAR*58
$PSTMSRR*49

//Disable **$PSTMDRSENMSG**:
$PSTMSETPAR,1228,10000000,2*0D
$PSTMSAVEPAR*58
$PSTMSRR*49

3.17.1. Odometer Message (<MsgID> = 1)

Inputs/outputs the count of accumulated pulses acquired by an odometer-type wheel rotation sensor. It is recommended to inject odometer count at a frequency of at least 10 Hz.

Type:

Input/output

Synopsis:

```
$PSTMDRSENMSG,1,<TimeStamp>,<Odometer>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<TimeStamp>	Numeric	Microsecond	CPU tick count since power-on. For input messages, this field is fixed as 0.
<Odometer>	Numeric	-	Unsigned odometer count (accumulated pulse count).

Example:

```
$PSTMDRSENMSG,1,0,0*10
$PSTMDRSENMSG,1,1924310999,0*25
```

3.17.2. Direction Message (<MsgID> = 2)

Inputs/outputs the status of the indicator of vehicle direction motion. It is recommended to inject status at a frequency of at least 10 Hz.

Type:

Input/output

Synopsis:

```
$PSTMDRSENMSG,2,<TimeStamp>,<Direction>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<TimeStamp>	Numeric	Microsecond	CPU tick count since power-on. For input messages, this field is fixed as 0.
<Direction>	Numeric	-	Vehicle direction. 0 = Forward 1 = Backward

Example:

```
$PSTMDRSENMSG,2,0,0*13
```

$PSTMDRSENMSG,2,1924310999,0*26

3.17.3. Odometer and Direction Message (<MsgID> = 3)

Outputs the accumulated count of pulses from an odometer-type wheel rotation sensor, along with the indication of vehicle direction.

Type:

Output

Synopsis:

$PSTMDRSENMSG,3,<TimeStamp>,<Odometer>,<Direction>*<Checksum><CR><LF>

Parameter:

Field	Format	Unit	Description
<TimeStamp>	Numeric	Microsecond	CPU tick count since power-on.
<Odometer>	Numeric	-	Unsigned odometer count (accumulated pulse count).
<Direction>	Numeric	-	Vehicle direction. 0 = Forward 1 = Backward

Example:

$PSTMDRSENMSG,3,1924310999,0,0*3B

3.17.4. Vehicle Speed Message (<MsgID> = 14)

Inputs/outputs vehicle speed expressed in kilometer per hour. It is recommended to inject vehicle speed at a frequency of at least 10 Hz.

Type:

Input/output

Synopsis:

$PSTMDRSENMSG,14,<TimeStamp>,<VehicleSpeed>*<Checksum><CR><LF>

Parameter:

Field	Format	Unit	Description
<Timestamp>	Numeric	Microsecond	CPU tick count since power-on. For input messages, this field is fixed as 0.
<VehicleSpeed>	Numeric	km/h	Vehicle speed.

Example:

```
$PSTMDRSENMSG,14,0,7.0*3D
$PSTMDRSENMSG,14,1375473563,7.0*0D
```

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

Outputs the IMU temperature. This message is provided at 1 Hz.

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 Celsius degrees: Celsius degrees = **<Temperature>** / 256 + 25.0
<Validity>	Numeric	-	Temperature validity. 0 = Not valid 1 = Valid

Example:

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

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

Outputs the raw data measured by the IMU accelerometer. This message is provided at 100 Hz.
Full Scale = ±2 g

Sensitivity = 0.061 mg/LSB
Acceleration (unit: mg) = Sensitivity × Raw acceleration

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 X-axis acceleration data in sensor frame.
<Y_ACC>	Numeric	-	Raw signed 16-bit integer Y-axis acceleration data in sensor frame.
<Z_ACC>	Numeric	-	Raw signed 16-bit integer Z-axis acceleration data in sensor frame.

Example:

```
$PSTMDRSENMSG,30,3248512115,299,-35,16514*02
$PSTMDRSENMSG,30,3248570594,302,-37,16524*09
```

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

Outputs the raw data measured by the IMU gyroscope. This message is provided at 100 Hz.
Full Scale = ±125 dps
Sensitivity = 4.375 mdps/LSB
Angular rate (unit: mdps) = Sensitivity × Raw angular rate

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 X-axis angular rate data in sensor frame.
<Y_GYRO>	Numeric	-	Raw signed 16-bit integer Y-axis angular rate data in sensor frame.
<Z_GYRO>	Numeric	-	Raw signed 16-bit integer Z-axis angular rate data in sensor frame.

Example:

```
$PSTMDRSENMSG,31,5855224,62,-119,-49*2A
$PSTMDRSENMSG,31,5913732,68,-117,-43*25
```

4 DR Operation Guide

This chapter explains how to inject vehicle speed in Wheel Tick Mode using the Quectel GNSS MODULE EVB V1.3 equipped with the L26-DR (ADR) module. It also explains how L26-DR (ADR) implements the ADR 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 test tooling on the vehicle floor with screws to ensure that it is firmly fixed to the vehicle.

Step 2 Mount the EVB with the module on the acrylic board using adhesive.

Step 3 Make sure that the GNSS antenna is rigidly fixed to the vehicle (it is recommended that the antenna should be mounted on the roof or other positions without GNSS signal obstructions on the vehicle).

EVB installation is shown in the figure below:

Figure 9: EVB Mounting on Acrylic Board

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:

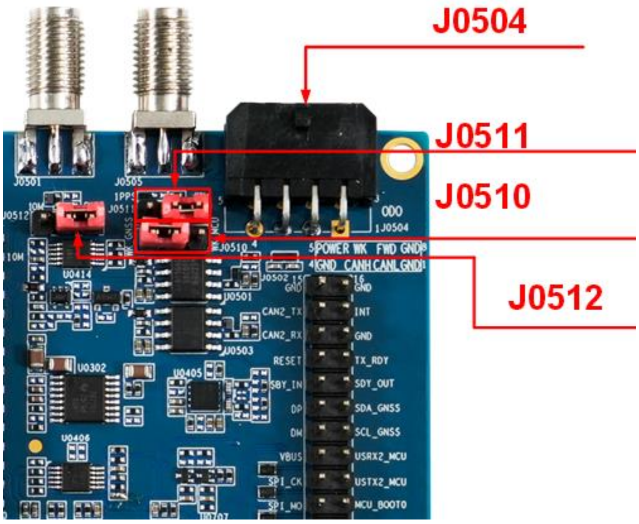


Figure 10: Wheel Tick Mode Jumper Cap Connections

Step 2 The vehicle’s forward/backward direction and wheel tick pulse signals are injected through the J0504 interface on the EVB.

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

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

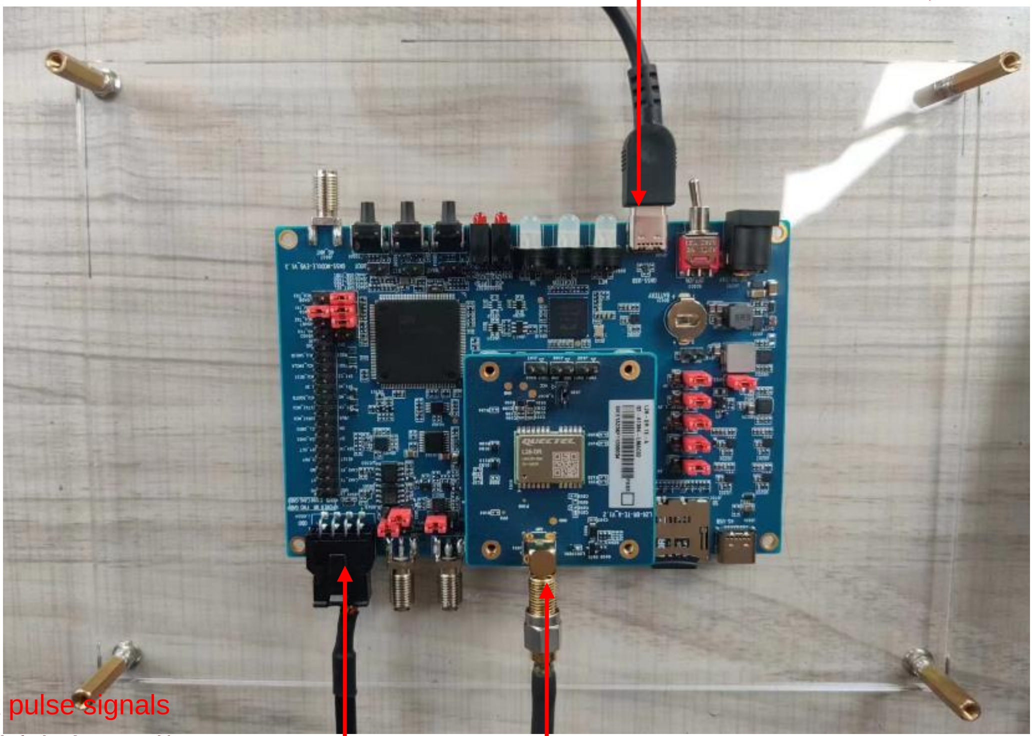


Figure 11: EVB Connections in Wheel Tick Mode

NOTE

On the GNSS MODULE EVB V1.3, the polarity of the input signal of the FWD pin on the J0504 interface 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 "**L26ADR**" 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 (ADR) firmware version is 115200 bps).
5) Click "**OK**".

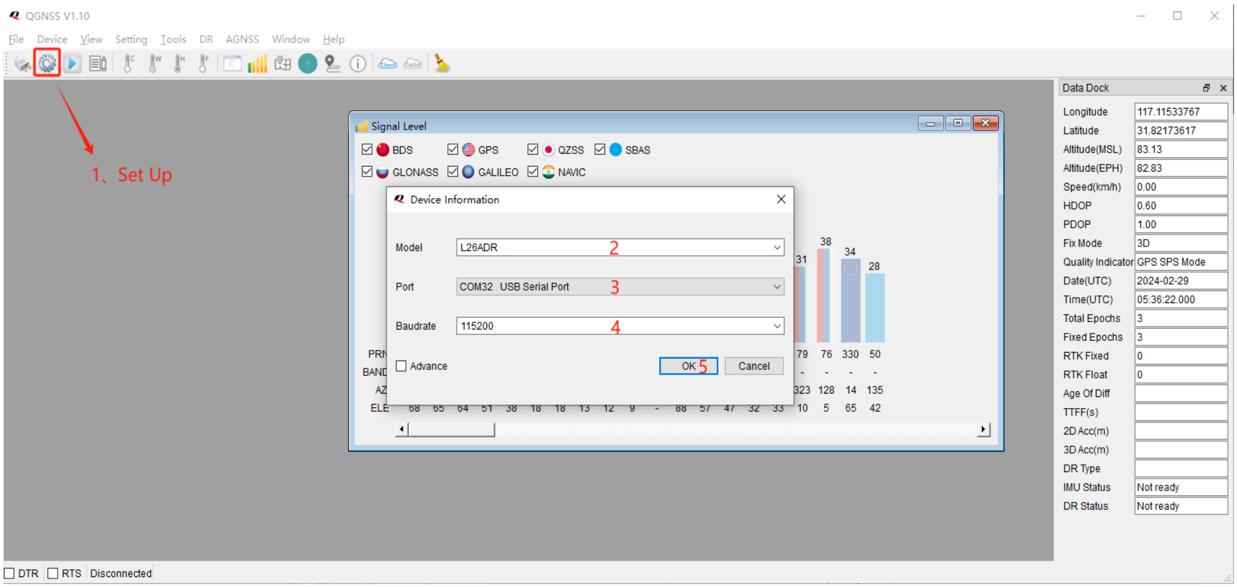


Figure 12: Serial Port Configuration

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.

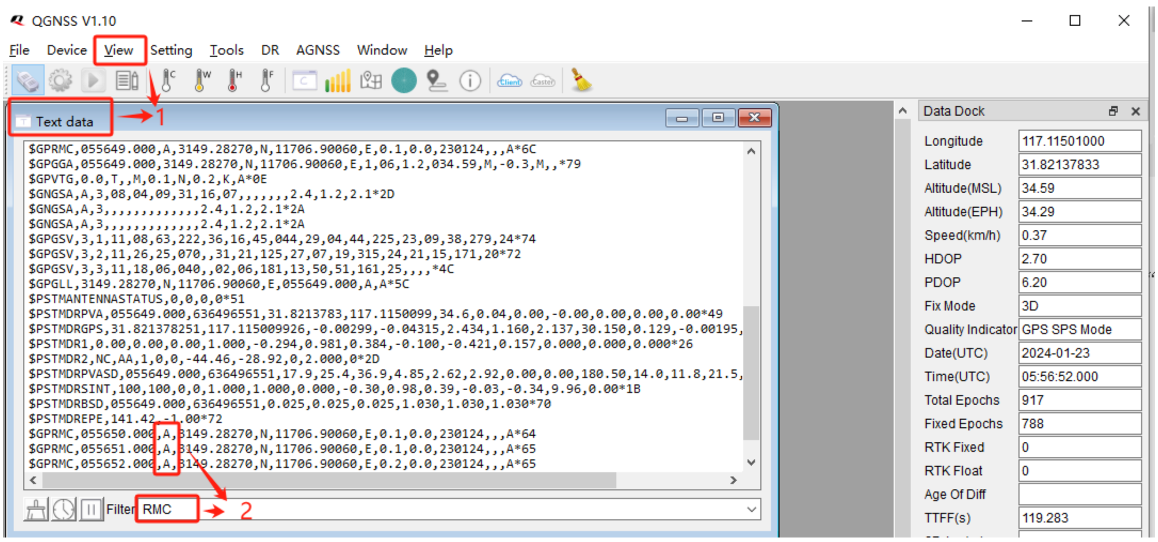


Figure 13: Status Check

Step 4 Click to open the "**Command console**" window. Refer to *Chapter 2.2.1 Wheel Tick Mode* and enter the configuration commands on the command line to set the vehicle speed injection mode. Click "**Send**" and observe whether the "**Text data**" window returns the result indicating successful command execution.

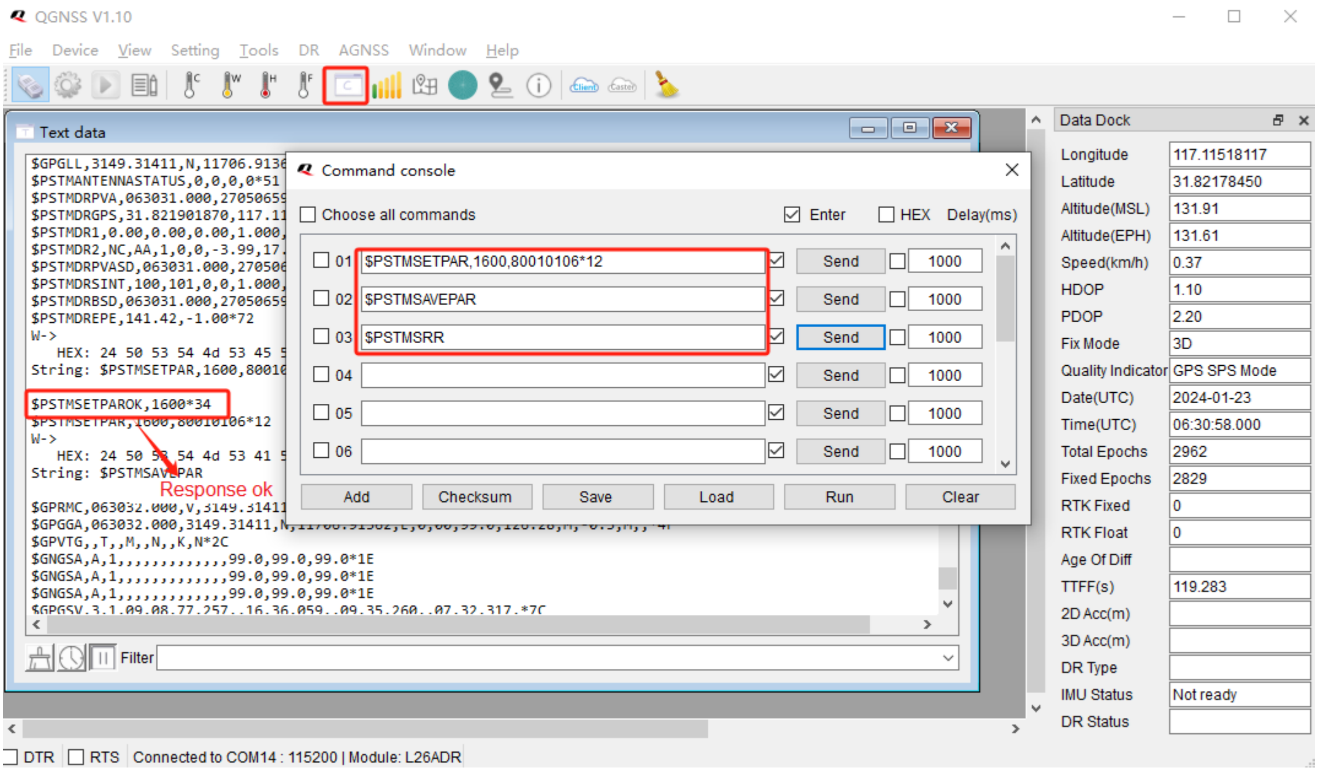


Figure 14: Vehicle Speed Injection Mode

4.4. DR Calibration

ADR calibration requires dynamic driving of a vehicle with a mounted module. You can refer to *Chapter 2.4.2 Guidelines* for instructions on how to perform module DR calibration.

Step 1 To achieve quick calibration, if driving conditions allow, drive along the “figure-**8**” route illustrated below. This type of driving route generally meets the conditions recommended in the guidelines, such as driving at speeds greater than 30 km/h, performing left and right turns, accelerating, braking, and parking.

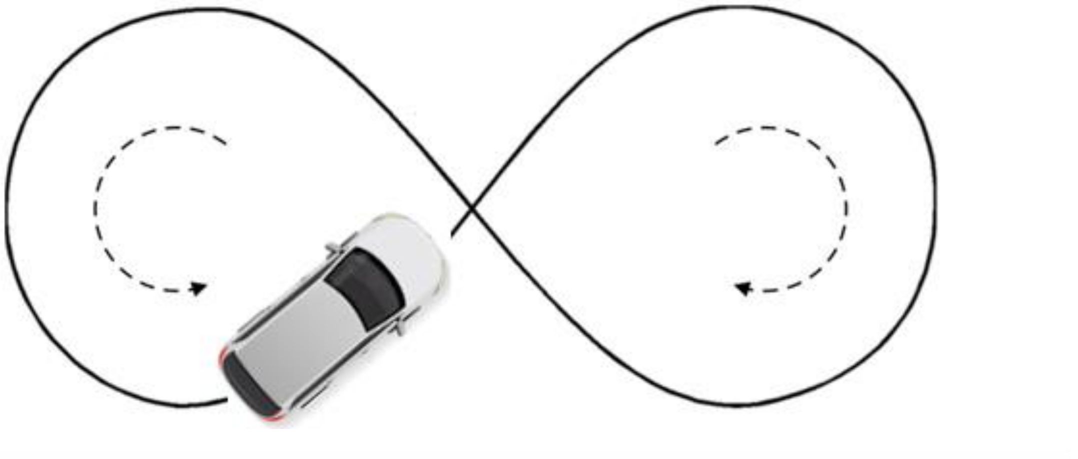


Figure 15: Calibration Route

Step 2 During the calibration process, check the **<OdoIsCalib>** field in the **$PSTMDRCAL** message and the **<Sr>** field in the **$PSTMDR2** message in the "**Text data**" window. The output values are both 1, indicating that the module ADR has been calibrated successfully. For more details, refer to *Chapter 3.10 PSTMDRCAL* and *Chapter 3.12 PSTMDR2*.

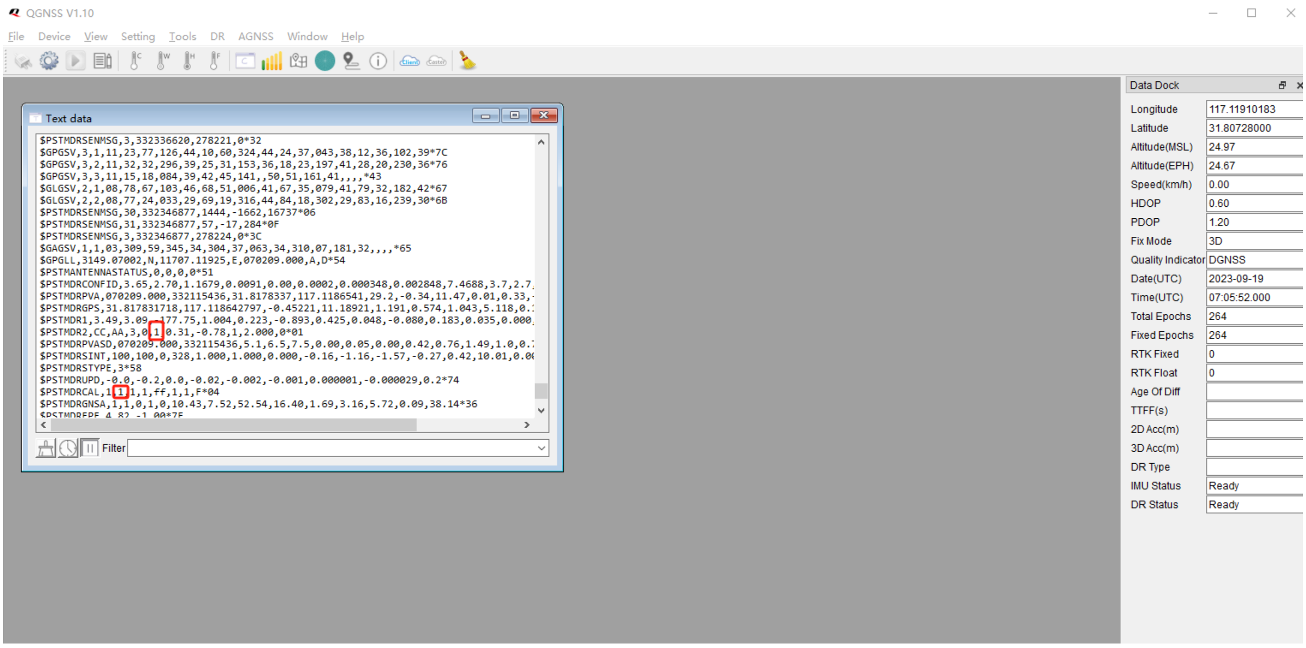
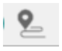


Figure 16: Messages Indicating Successful DR Calibration

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.

Step 1 Click  to open the "**Deviation Map**" window;

Step 2 Observe the real-time positioning trajectory of the vehicle. As shown in the figure below, a continuous positioning trajectory in the garage indicates successful implementation of the ADR function.

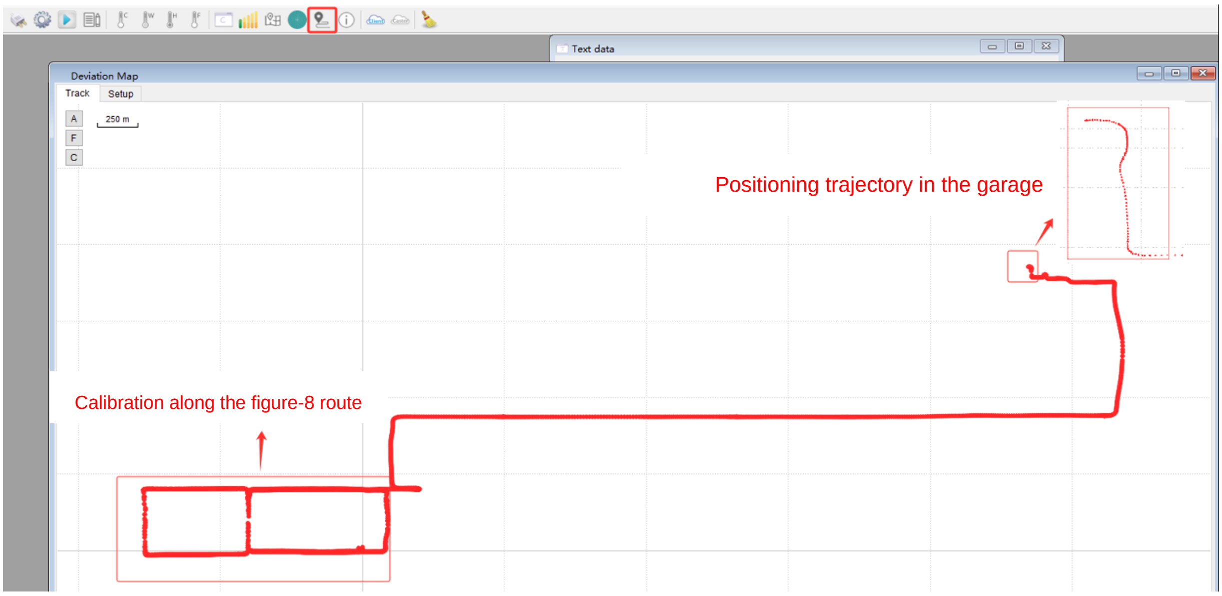


Figure 17: ADR Function Implementation

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 [3] QGNSS user guide*.

5 FAQs

1. **Is it necessary to perform DR calibration process every time the DR function is used?**
   - The module is configured by default to save calibration status. For details on the conditions for saving calibration status, refer to *Chapters 2.4.4 Calibration Status Storage*. This allows the module to bypass calibration actions and avoid entering calibration mode during the subsequent startup. However, it is important to note that if the module changes its relative position after being turned off, the saved calibration parameters will become invalid. It is recommended to clear the calibration parameters using the following commands and then re-perform the DR calibration process.

```
$PSTMGPSSUSPEND*14
$PSTMNVMITEMINV,80,1*32
$PSTMNVMITEMINV,81,1*33
$PSTMSRR*49
```

2. **Under what circumstances ADR performance may not be guaranteed?**
   - Abnormal vehicle speed input in ADR mode;
   - No reversing signal is input while the vehicle is in reverse gear;
   - During DR calibration, the driving manoeuvres are limited, such as only performing unidirectional driving;
   - Poor GNSS signal quality during DR calibration, such as severe signal obstruction;
   - Scenarios with extremely poor GNSS signal;
   - The module is not securely mounted.

6 Appendix A References

Table 1: 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 2: Terms and Abbreviations

Abbreviation	Description
3D	3-Dimensional
ADR	Automotive Dead Reckoning
DR	Dead Reckoning
FFM	Full Free Mount
FW	Firmware
GNSS	Global Navigation Satellite System
IMU	Inertial Measurement Unit
KF	Kalman Filter
LSB	Least Significant Bit
MEMS	Micro-Electro-Mechanical System
NMEA	NMEA (National Marine Electronics Association) 0183 Interface Standard
NVM	Non-volatile Memory

Abbreviation	Description
OBD	On-Board Diagnostics
PC	Personal Computer
PVT	Position, Velocity and Time
RMS	Root Mean Square
UART	Universal Asynchronous Receiver/Transmitter
UDR	Untethered Dead Reckoning
USB	Universal Serial Bus
UTC	Universal Time Coordinated
VRO	Virtual Odometer

7 Appendix B Special Characters

Table 3: 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.
Underline	Default setting of a parameter.