LG69T (AM,AP) DR&RTK Application Note

GNSS Module Series

Version: 1.0

Date: 2023-08-10

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	**LG69T (AM,AP) DR&RTK Application Note**
Subtitle	GNSS Module Series
Document Type	Application Note
Document Status	Released

Revision History

Version	Date	Description
-	2023-06-12	Creation of the document
1.0.0	2023-08-10	First official release

Contents

Table Index

Figure Index

1 Introduction

This document describes the dead reckoning (DR) and real-time kinematic (RTK) features, including orientation, mounting, calibration, and messages related to DR and RTK of Quectel LG69T (AM) and LG69T (AP). The supported features on each module are as follows:

- LG69T (AM) supports RTK only.
- LG69T (AP) supports DR and RTK.

The modules are recommended for use in four-wheel vehicle.

Figure 1: Four-Wheel Vehicle

1.1. Overview on DR

Quectel LG69T (AP) supports the ADR mode, where the module utilizes speed data from the vehicle and the onboard 6-axis sensor to enhance accuracy in environments without GNSS coverage.

The module obtains vehicle speed data through CAN interface, supporting receiving data that conforms to ISO11898-1 standard. The vehicle speed requirement includes a minimum injection frequency of 10 Hz, delay below 10 ms, and speed RMS below 0.5 m/s.

1.2. Overview on RTK

Quectel offers high-precision RTK positioning products that support multiple GNSS constellations and frequencies, enabling centimeter-level accuracy. RTK (Real Time Kinematic) is a real-time differential GPS (RTDGPS) technology based on carrier phase observations. It consists of three components: a base station (which may be an actual satellite receiver or a virtual reference station (VRS), differential correction data link, and rover. The base station continuously observes GNSS satellites, and transmits its observation data and station information to the rover in real-time through a data transmission network. While receiving GNSS satellite signals, the rover uses the data transmitted by the base station through wireless receiving equipment to calculate its three-dimensional coordinates in real-time relative to the base station, using the principle of relative positioning.

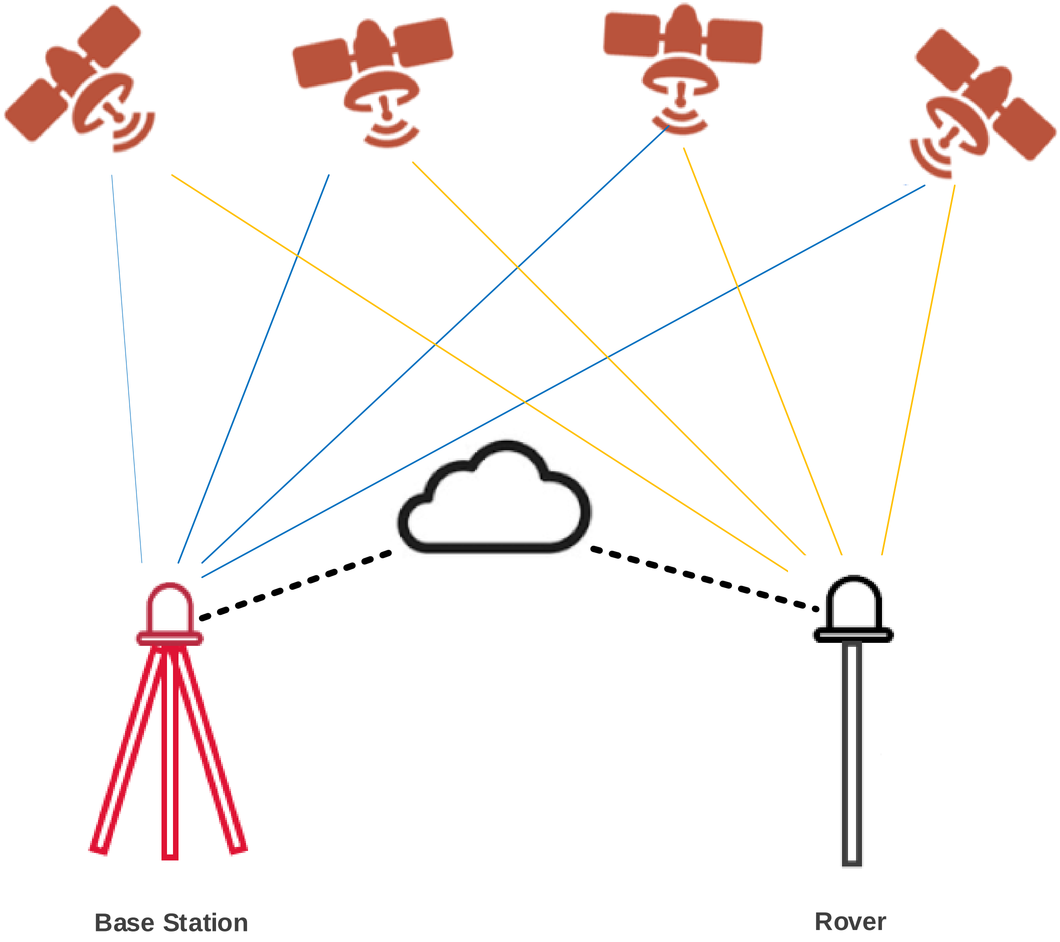

Figure 2: RTK Schematic Diagram

2 Configuration

2.1. DR Configuration

2.1.1. Orientation

The LG69T (AP) modules are specifically designed for use in four-wheel vehicles. Module with GNSS and motion sensor ICs must be firmly and securely fixed to vehicle body. No relative movement is allowed between vehicle and device, and maximum isolation from both shock and vibration must be provided. Manually holding the board is not accepted. The recommended installation method is to firmly screw the device down to the vehicle frame. Mounting location should allow easy access to power supply and GNSS antenna, while avoiding exposure to excessive heat.

The reference frame axes definitions are as follows:

- The X-axis points towards the right side of the vehicle.
- The Y-axis points towards the front of the vehicle.
- The Z-axis points upwards from the vehicle.

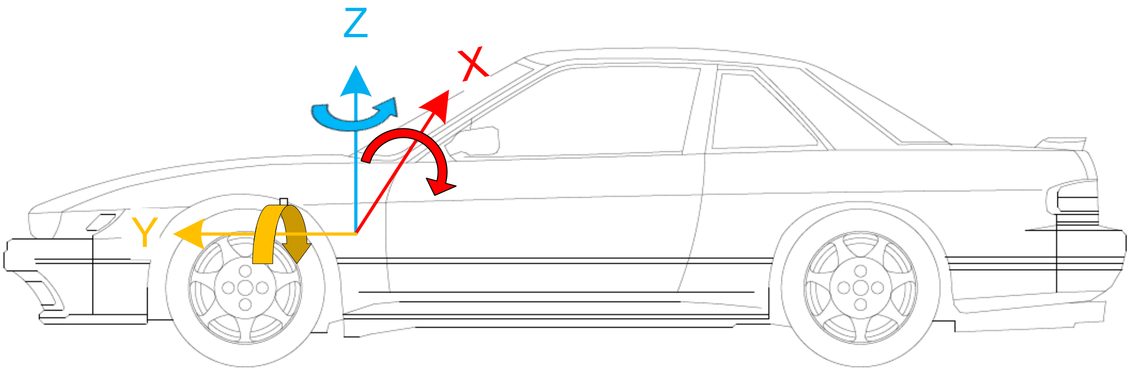

Figure 3: Reference Frame

The IMU reference frame of the module is shown below:

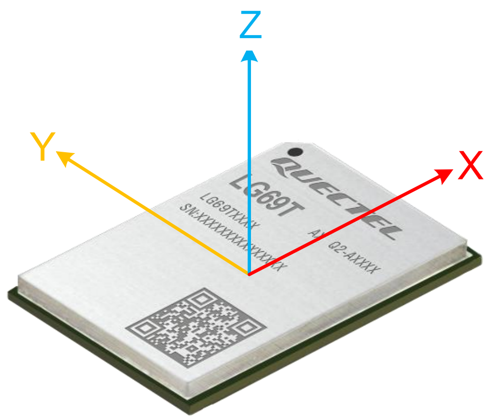

Figure 4: IMU Reference Frame

2.1.2. Mounting

When mounting the Quectel LG69T (AP) module on the vehicle:

1) Firmly attach LG69T (AP) to the vehicle body, preferably as close to the center of the rear wheel as possible.
2) The default installation of LG69T (AP) should be horizontal, aligning the X/Y axis of the module with the X/Y axis of the vehicle body as closely as possible, and ensuring the misalignment angles (α, β, γ) are less than 10°.
3) Avoid misalignment angles (α, β, γ) greater than 30°.

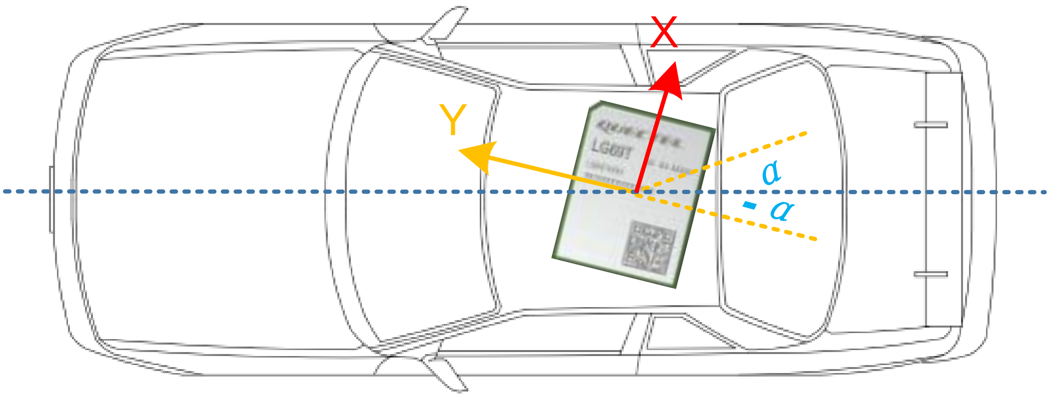

Figure 5: Installation Yaw Angle

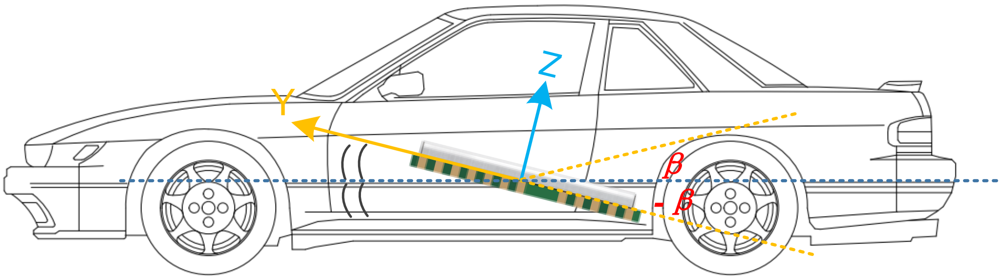

Figure 6: Installation Pitch Angle

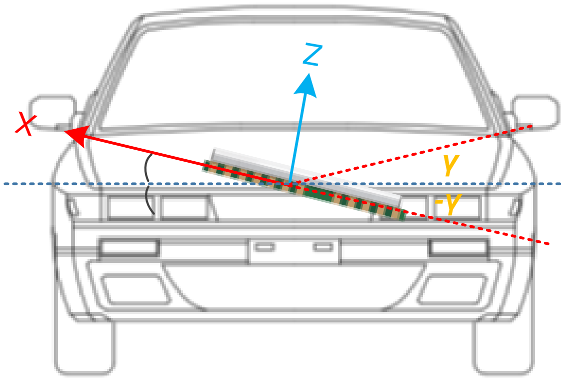

Figure 7: Installation Roll Angle

NOTE

Firmly affix the module to the vehicle body. Select a structurally sound location that is not prone to flexing.

2.1.3. DR Calibration

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

Step 1: Fix 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 and/or void calibration.

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

Step 3: Power up the module, then start the vehicle on a plain surface.

Step 4: Drive at a speed of more than 2.5 m/s, and preferably perform at least 1 turning movement. Then, the module will start self-calibration, which will be completed in 2 minutes.

Step 5: Once the calibration process is completed, the module immediately outputs the **$PQTMDRPVA** message. See *Chapter 4.2 PQTMDRPVA* for details about the message.

After the calibration, there is no limit to driving trajectory and driving dynamics, and you can perform verification tests in the following scenarios:

1) Open sky area, urban main road (good satellite signal).
2) Urban tunnels (absence of satellite signal).
3) Urban viaduct (weak satellite signal).
4) Underground vehicle park (absence of satellite signal).
5) Areas surrounded with dense buildings (multi-path in urban canyon).
6) City boulevards (weak satellite signal).
7) High-rise dense area (severe urban canyon multipath).

NOTE

1. **$PQTMDRPVA** is only output when calibration process is completed.
2. The standard NMEA message has a fixed output frequency of 1 Hz, however the **GGA**, **RMC**, **GST** and **VTG** output frequency will shift to 10 Hz after the DR calibration procedure is completed. The combination of **$PQTMCFGFIXRATE** and **$PQTMCFGMSGRATE** can be used to configure the output frequency of standard NMEA message. For more details about these commands, see *document [1] protocol specification*.

3 RTK Input

Quectel LG69T (AM) and LG69T (AP) modules support the RTCM 10403.3 input messages listed in table below.

Table 1: Supported RTCM Input Messages

Message Type	Description
1005	Stationary RTK Reference Station ARP
1006	Stationary RTK Reference Station ARP with Antenna Height
1074	GPS MSM4
1077	GPS MSM7
1094	Galileo MSM4
1097	Galileo MSM7
1114	QZSS MSM4
1117	QZSS MSM7
1124	BDS MSM4
1127	BDS MSM7

4 PQTM Messages

This chapter outlines the PQTM messages (proprietary NMEA messages defined by Quectel) for DR function, which are supported by LG69T (AP).

Table 2: Error Code

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

4.1. PQTMPVT

Outputs the PVT (GNSS-only) result.

Type:

Output

Synopsis:

```
$PQTMPVT,<MsgVer>,<TOW>,<Date>,<Time>,<Quality>,<FixMode>,<NumSatUsed>,<LeapS>,<Lat>,<Lon>,<Alt>,<Sep>,<VelN>,<VelE>,<VelD>,<Spd>,<Heading>,<HDOP>,<PDOP>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<MsgVer>	Numeric	-	Message version. 1 = Version 1 (Always 1 for this version.)
<TOW>	Numeric	Millisecond	Time of week.
<Date>	yyyymmdd	-	UTC date. yyyy: Year mm: Month dd: Day of month

Field	Format	Unit	Description
<Time>	hhmmss.sss	-	UTC time. hh: Hours (00–23) mm: Minutes (00–59) ss: Seconds (00–59) sss: Decimal fraction of seconds
<Quality>	Numeric	-	GPS quality indicator. 0 = Fix not available or invalid. 1 = GPS SPS mode, fix valid. 2 = Differential GPS, SPS mode, or Satellite Based Augmentation. System (SBAS), fix valid. 3 = GPS PPS mode, fix valid. 4 = Real Time Kinematic (RTK) System used in RTK mode with fixed integers. 5 = Float RTK. Satellite system used in RTK mode, floating integers. 6 = Estimated (DR) mode. Note that this field is the same as **<Quality>** in **GGA**.
<FixMode>	Numeric	-	Fix mode. 0 = No fix. 1 = Reserved. 2 = 2D fix. 3 = 3D fix.
<NumSatUsed>	Numeric	-	Number of satellites in use.
<LeapS>	Numeric	Second	Leap seconds. Note that this field is empty in case of an invalid value.
<Lat>	Numeric	Degree	Latitude. Note that this field is empty in case of an invalid value.
<Lon>	Numeric	Degree	Longitude. Note that this field is empty in case of an invalid value.
<Alt>	Numeric	Meter	Altitude above mean-sea-level. Note that this field is empty in case of an invalid value.
<Sep>	Numeric	Meter	Geoidal separation (the difference between the WGS84 earth ellipsoid surface and the mean-sea-level surface). Note that this field is empty in case of an invalid value.

Field	Format	Unit	Description
<VelN>	Numeric	m/s	North velocity. Note that this field is empty in case of an invalid value.
<VelE>	Numeric	m/s	East velocity. Note that this field is empty in case of an invalid value.
<VelD>	Numeric	m/s	Down velocity. Note that this field is empty in case of an invalid value.
<Spd>	Numeric	m/s	Ground speed. Note that this field is empty in case of an invalid value.
<Heading>	Numeric	Degree	Heading. Note that this field is empty in case of an invalid value. Range: 0.00–360.00
<HDOP>	Numeric	-	Horizontal dilution of precision. Note that the value is 99.99 in case of an invalid value.
<PDOP>	Numeric	-	Position (3D) dilution of precision. Note that the value is 99.99 in case of an invalid value.

Example:

```
//No fix.
$PQTMPVT,1,1000,20221225,163355.000,0,0,00,,,,,,,,,,99.99,99.99*49

//3D fix.
$PQTMPVT,1,31075000,20221225,083737.000,4,3,09,18,31.12738291,117.26372910,34.212,5.267,3.212,2.928,0.238,4.346,34.12,2.16,4.38*65
```

4.2. PQTMDRPVA

Outputs the DR position, velocity and attitude.

Type:

Output

Synopsis:

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

Parameter:

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

Field	Format	Unit	Description
<Pitch>	Numeric	Degree	Pitch angle. Note that this field is empty in case of an invalid value. Range: -90.000000 to 90.000000
<Heading>	Numeric	Degree	Heading. Note that this field is empty in case of an invalid value. Range: 0.000000–360.000000

Example:

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

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

4.3. PQTMSENMSG

Outputs sensor information.

Type:

Output

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<MsgVer>	Numeric	-	Message version. 2 = IMU sensor data. See *Figure 4: IMU Reference Frame*.
<TimeStamp>	Numeric	-	Timestamp since power-on. 32-bit unsigned integer.
<Par1> to <ParN>	Numeric	-	Sensor information. See *Chapter 4.3.1 When <MsgVer> = 2*.

4.3.1. When <MsgVer> = 2

This message is used to output the IMU raw data, these values are read directly from the IMU without any filtering. Refer to IMU reference frame.

Synopsis:

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

Parameter:

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

Example:

$PQTMSENMSG,2,2248864,36.70,0.6118,-0.4064,-0.5419,0.0132,-0.0151,1.0045*34

NOTE

1. **$PQTMCFGMSGRATE** can be used to enable or disable **$PQTMSENMSG** message output. For more details about this command, see *document [1] protocol specification*.
   //Enable **$PQTMSENMSG** command:
   $PQTMCFGMSGRATE,W,PQTMSENMSG,100,2*4C
   $PQTMCFGMSGRATE,OK*29

   //Disable **$PQTMSENMSG** command:
   $PQTMCFGMSGRATE,W,PQTMSENMSG,0,2*4D
   $PQTMCFGMSGRATE,OK*29

4.4. PQTMVEHMSG

Outputs vehicle information.

Type:

Output

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<MsgType>	Numeric	-	Message type: 1 = Outputs vehicle speed (in m/s) information via UART interface
<Timestamp>	Numeric	-	Timestamp since power-on. 32-bit unsigned integer.
<Par1> to <ParN>	Numeric	-	This field varies with the message type. See *Chapter 4.4.1 When <MsgType> = 1*.

4.4.1. When <MsgType> = 1

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<Timestamp>	Numeric	Millisecond	Timestamp since power-on. 32-bit unsigned integer.
<VehSpeed>	Numeric	m/s	Vehicle speed. Range: -100 to 100.

Example:

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

NOTE

1. **$PQTMCFGMSGRATE** can be used to enable or disable **$PQTMVEHMSG** message output. For more details about this command, see *document [1] protocol specification*.
   //Enable **$PQTMVEHMSG** command:
   $PQTMCFGMSGRATE,W,PQTMVEHMSG,1,1*4C
   $PQTMCFGMSGRATE,OK*29

   //Disable **$PQTMVEHMSG** command:
   $PQTMCFGMSGRATE,W,PQTMVEHMSG,0,1*4D
   $PQTMCFGMSGRATE,OK*29

4.5. PQTMCFGCAN

Configures CAN interface and baud rate.

Type:

Set/Get

Synopsis:

```
//Set:
$PQTMCFGCAN,W,<PortID>,<Enable>,<FrameFormat>,<Baudrate>,<DataBaudrate>*<Checksum><CR><LF>
//Get:
$PQTMCFGCAN,R,<PortID>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<PortID>	Numeric	-	CAN interface index. 0 = CAN
<Enable>	Numeric	-	Enable/disable CAN interface. 0 = Disable 1 = Enable
<FrameFormat>	Numeric	-	Frame format of CAN interface. 0 = Classic CAN 1 = CAN FD without bitrate switching 2 = CAN FD with bitrate switching
<Baudrate>	Numeric	bps	Nominal baud rate of CAN interface.

Field	Format	Unit	Description
<DataBaudrate>	Numeric	bps	Data baud rate of CAN interface. Note: This field is only valid in CAN FD frame format. It can be set to 0 in classic CAN frame format.

Result:

- If successful, the module returns:

```
//Set:
$PQTMCFGCAN,OK*74
//Get:
$PQTMCFGCAN,OK,<PortID>,<Enable>,<FrameFormat>,<Baudrate>,<DataBaudrate>*<Checksum><CR><LF>
```

- If failed, the module returns:

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

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

Example:

```
//Sets the CAN interface
$PQTMCFGCAN,W,0,1,0,500000,0*45
$PQTMCFGCAN,OK*3E

//Gets the configuration of CAN interface
$PQTMCFGCAN,R,0*74
$PQTMCFGCAN,OK,0,1,0,500000,0*16
```

4.6. PQTMCFGCANFILTER

Configures the CAN filter. The module only receives messages that match the filtering rules.

Type:

Set/Get

Synopsis:

```
//Set:
$PQTMCFGCANFILTER,W,<PortID>,<Index>,<Enable>,<FilterType>,<MsgID_Type>,<MsgID1>,<MsgID2>*<Checksum><CR><LF>
```

```
//Get:
$PQTMCFGCANFILTER,R,<PortID>,<Index>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<PortID>	Numeric	-	CAN interface index. 0 = CAN
<Index>	Numeric	-	Filter index. 0 = Filter 0 1 = Filter 1
<Enable>	Numeric	-	Enable/disable filter. 0 = Disable 1 = Enable
<FilterType>	Numeric	-	Filter type. 0 = Range filter 1 = Dual ID filter 2 = Mask filter
<MsgID_Type>	Numeric	-	Identifier type. 0 = Standard ID 1 = Extended ID
<MsgID1>	Hexadecimal	-	Filter identification 1. Range: 0–0x7FF, if **<MsgID_Type>** is Standard ID Range: 0–0x1FFFFFFF, if **<MsgID_Type>** is Extended ID
<MsgID2>	Hexadecimal	-	Filter identification 2. Range: 0–0x7FF, if **<MsgID_Type>** is Standard ID Range: 0–0x1FFFFFFF, if **<MsgID_Type>** is Extended ID

Result:

- If successful, the module returns:

```
//Set:
$PQTMCFGCANFILTER,OK*3E
//Get:
$PQTMCFGCANFILTER,OK,<PortID>,<Index>,<Enable>,<FilterType>,<MsgID_Type>,<MsgID1>,<MsgI
D2>*<Checksum><CR><LF>
```

- If failed, the module returns:

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

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

Example:

```
//Sets the CAN filter 0.
$PQTMCFGCANFILTER,W,0,0,1,0,0,3E9,1F5*7D
$PQTMCFGCANFILTER,OK*3E

//Gets the configuration of CAN filter 0
$PQTMCFGCANFILTER,R,0,0*68
$PQTMCFGCANFILTER,OK,0,0,1,1,0,3E9,1F5*2F
```

NOTE

If the filter type is mask filter (**<FilterType>** = 2):

- **<MsgID1>** is the filter ID, and **<MsgID2>** is the mask.
- If a mask bit is set to 1, the corresponding ID bit will be compared with the value of the filter ID bit; and if it matches then the message will be accepted, otherwise the message will be rejected.

4.7. PQTMCFGVEHDBC

Configure the vehicle Database for CAN (DBC).

Type:

Set/Get

Synopsis:

```
//Set:
$PQTMCFGVEHDBC,W,<Index>,<MsgID>,<StartBit>,<BitSize>,<ByteOrder>,<ValueType>,<Factor>,<Offset>,<Min>,<Max>*<Checksum><CR><LF>
//Get:
$PQTMCFGVEHDBC,R,<Index>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Index>	Numeric	-	DBC index. 0 = Vehicle speed DBC 1 = Vehicle gear DBC
<MsgID>	Numeric	-	CAN message ID.
<StartBit>	Numeric	-	Start bit of data.

Field	Format	Unit	Description
<BitSize>	Numeric	-	Bit size of data.
<ByteOrder>	Numeric	-	Byte order. 0 = Motorola mode (Big endian) 1 = Intel mode (Litter endian)
<ValueType>	Numeric	-	Value type. Not support, always 0
<Factor>	Numeric	-	Factor of value.
<Offset>	Numeric	-	Offset of value.
<Min>	Numeric	-	Minimal value.
<Max>	Numeric	-	Maximum value.

Result:

- If successful, the module returns:

```
//Set:
$PQTMCFGVEHDBC,OK*6C
//Get:
$PQTMCFGVEHDBC,OK,<Index>,<MsgID>,<StartBit>,<BitSize>,<ByteOrder>,<ValueType>,<Factor>,<
Offset>,<Min>,<Max>*<Checksum><CR><LF>
```

- If failed, the module returns:

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

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

Example:

```
//Sets the vehicle speed DBC
$PQTMCFGVEHDBC,W,0,3E9,8,16,0,0,0.015625,0,0,300*57
$PQTMCFGVEHDBC,OK*6C

//Gets the configuration of vehicle speed DBC
$PQTMCFGVEHDBC,R,0*26
$PQTMCFGVEHDBC,OK,0,3E9,8,16,0,0,0.015625,0.000000,0.000000,300.000000*2A
```

4.8. PQTMCFGVEHRVAL

Configures the vehicle reverse gear value based on CAN forward/backward information.

Type:

Set/Get

Synopsis:

```
//Set:
$PQTMCFGVEHRVAL,W,<RVal>*<Checksum><CR><LF>
//Get:
$PQTMCFGVEHRVAL,R*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<RVal>	Numeric		Vehicle reverse gear value. Range: 0–255

Result:

- If successful, the module returns:

```
//Set:
$PQTMCFGVEHRVAL,OK*20
//Get:
$PQTMCFGVEHRVAL,OK,<RVal>*<Checksum><CR><LF>
```

- If failed, the module returns:

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

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

Example:

```
//Set:
$PQTMCFGVEHRVAL,W,2*6D
$PQTMCFGVEHRVAL,OK*20

//Get:
$PQTMCFGVEHRVAL,R*76
$PQTMCFGVEHRVAL,OK,2*3E
```

4.9. PQTMCFGLA

Configures the lever arm.

Type:

Set/Get

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<Type>	Numeric	-	Type of lever arm. 1 = IMU to antenna 1 lever arm. 3 = IMU to user lever arm.
<LA_X>	Numeric	Meter	Lever arm of X axis.
<LA_Y>	Numeric	Meter	Lever arm of Y axis.
<LA_Z>	Numeric	Meter	Lever arm of Z axis.

Result:

- If successful, the module returns:

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

- If failed, the module returns:

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

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

Example:

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

NOTE

After issuing this command, the module must be restarted for the command to take effect.

5 Appendix A References

Table 3: Related Documents

Document Name
[1] Quectel_LG69T(AM,AP)_GNSS_Protocol_Specification

Table 4: Terms and Abbreviations

Abbreviation	Description
2D	2 Dimension
3D	3 Dimension
ADR	Automotive Dead Reckoning
ARP	Antenna Reference Point
CAN	Controller Area Network
CAN FD	CAN with Flexible Data rate
BDS	BeiDou Navigation Satellite System
DBC	Database for CAN
DR	Dead Reckoning
GNSS	Global Navigation Satellite System
GGA	Global Positioning System Fix Data
GPS	Global Positioning System
IMU	Inertial Measurement Unit
MSM	Multiple Signal Message
NMEA	NMEA (National Marine Electronics Association) 0183 Interface Standard

Abbreviation	Description
PPS	Pulse Per Second
PQTM	Proprietary Protocol of Quectel
QZSS	Quasi-Zenith Satellite System
RMS	Root mean square
RTCM	Radio Technical Commission for Maritime Services
RTDGPS	Real Time Differential Global Positioning System
RTK	Real-Time Kinematic
SPS	Standard Positioning Service
UART	Universal Asynchronous Receiver/Transmitter
UTC	Coordinated Universal Time
VRS	Virtual Reference Station
WGS84	World Geodetic System 1984

6 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.
Underline	Default setting of a parameter.