LG290P&LGx80P Series

GNSS Protocol Specification

GNSS Module Series

Version: 1.1

Date: 2025-07-25

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

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

Use and Disclosure Restrictions

License Agreements

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

Copyright

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

Trademarks

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

Third-Party Rights

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

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

Privacy Policy

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

Disclaimer

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

About the Document

Document Information	
Title	**LG290P (03)&LGx80P (03) GNSS Protocol Specification**
Subtitle	GNSS Module Series
Document Type	GNSS Protocol Specification
Document Status	Released

Revision History

Version	Date	Description
-	2023-12-18	Creation of the document
1.0	2024-09-02	First official release
1.1	2025-07-25	1. Added applicable modules LG580P (03) and LG680P (03). 2. Added the support of M for <ModeInd> parameter in RMC, VTG, GLL, and the support of 7 for <Quality> in GGA (*Chapters 2.2.1*, *2.2.2 2.2.5* and *2.2.6*). 3. Added new standard NMEA0183 messages (*Chapters 2.2.7* to *2.2.12*). 4. Updated $PQTMCFGMSGRATE message (*Chapter 2.3.15*). 5. Changed <Heading> to <COG> in (*Chapter 2.3.16*). 6. Added the <Distance> on $PQTMCFGSVIN and the range and default value on <3D_AccLimit> and (*Chapter 2.3.22*). 7. Added the <Timeout> on $PQTMCFGRTK (*Chapter 2.3.29*). 8. Added a note about if $PQTMRESETODO is sent when the position is lost (*Chapter 2.3.34*). 9. Added the support for QZSS L6 band in <QZS_Sig> parameter (*Chapter 2.3.36*). 10. Added the support for QZSS L6 band in <SignalID> and <MaskLow> parameters (*Chapter 2.3.37*). 11. Added PQTM messages $PQTMCFGSBAS, $PQTMCFGNMEATID,

Version	Date	Description
		$PQTMTAR, $PQTMCFGBLD, $PQTMCFGRTKSRCTYPE, $PQTMSN, $PQTMCFGANTINF, $PQTMCFGANTDELTA, $PQTMCFGSIGGRP, $PQTMCFGSIGNAL2, $PQTMCFGGEOSEP, $PQTMCFGCNRTHD, $PQTMCFGELETHD, $PQTMNAV, $PQTMEOE, $PQTMCFGWN, $PQTMANTENNASTATUS and antenna 2 proprietary messages (*Chapters 2.3.40* to *2.3.57*). 12. Added QGC protocol (*Chapter 3*). 13. Added a new standard RTCM3 message (*Chapter 4*).

Contents

Table Index

Figure Index

1 Introduction

Quectel LG290P (03), LG580P (03)and LG680P (03) GNSS modules support GPS, GLONASS, Galileo, BDS, QZSS and NavIC (IRNSS) constellations, providing fast and accurate acquisition and making this module an ideal solution for positioning and navigation in various vertical markets.

Table 1: Applicable Modules and Supported Frequency Bands

Module	Frequency Band
LG290P (03) LG580P (03) LG680P (03)	GPS: L1 C/A, L1C [1], L2C, L5-Q GLONASS: G1 C/A, G2 C/A Galileo: E1, E5a, E5b, E6 BDS: B1I, B1C, B2a, B2b, B2I, B3I QZSS: L1 C/A, L1C [1], L2C, L5-Q, L6 NavIC: L5

This document describes the software commands that are needed to control and modify the module configuration. The software commands are NMEA proprietary commands defined by Quectel (PQTM commands). To report GNSS information, the module supports message outputting in NMEA 0183 protocol and RTCM protocol format.

The LG290P (03), LG580P (03) and LG680P (03) modules support the following protocols:

Table 2: Supported Protocol

Protocol	Type
NMEA 0183 V4.11	Output, ASCII, standard
	Input/output, ASCII, proprietary
RTCM 10403.3	Input/output, binary

NOTE

1. [1)] GPS L1C and QZSS L1C frequency bands are under development. Contact Quectel Technical Support (support@quectel.com) for details.
2. Quectel assumes no responsibility if commands other than the ones listed herein are used.

2 NMEA Protocol

2.1. Structure of NMEA Protocol Messages

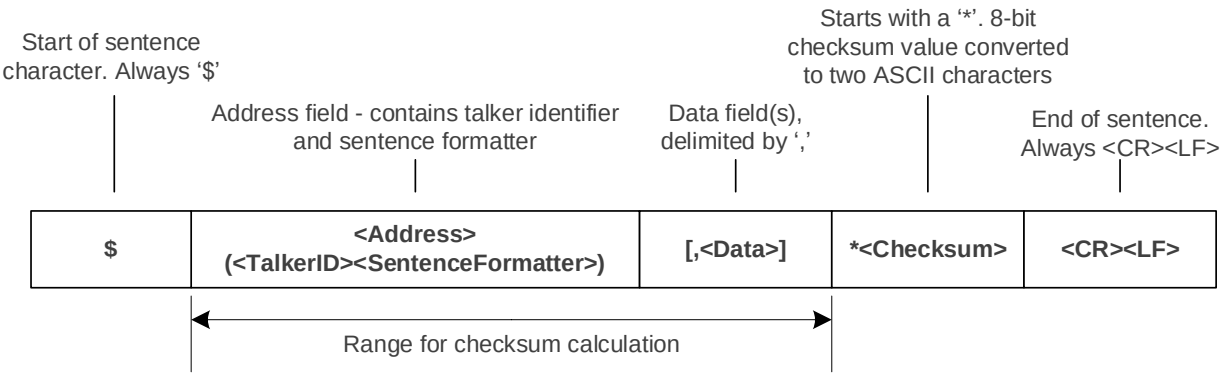

Figure 1: Structure of NMEA Protocol Messages

Table 3: Structure of NMEA Protocol Messages

Field	Description
$	Start of the sentence (Hex 0x24).
<Address>	**In Standard Messages:** In standard messages, this field consists of a two-character talker identifier (TalkerID) and a three-character sentence formatter (SentenceFormatter). The talker identifier identifies the type of talker. For more information on the TalkerID, see *Table 4: NMEA Talker ID*. The sentence formatter identifies the data type and the string format of the successive fields. **In Proprietary Messages:** In proprietary messages, this field consists of the proprietary character **P** followed by a three-character Manufacturer's Mnemonic Code, used to identify the TALKER issuing a proprietary sentence, and any additional characters as required.

Field	Description
<Data>	Data fields, delimited by the data field delimiter **‘,’**. Variable length (depending on the NMEA message type).
<Checksum>	Checksum field follows the checksum delimiter character *****. Checksum is the 8-bit exclusive OR of all characters in the sentence, including **‘,’** the field delimiter, between but not including the **$** and the ***** delimiters.
<CR><LF>	End of sentence (Hex 0x0D 0x0A).

Table 4: NMEA Talker ID

GNSS Constellation Configuration	TalkerID (NMEA 0183 V4.11)
GPS	GP
GLONASS	GL
Galileo	GA
BDS	GB
QZSS	GQ
NavIC (IRNSS)	GI
Combination of Multiple Satellite Systems	GN

NMEA Checksum Sample Code:

```
// pData is the data array whose checksum needs to be calculated:

unsigned char Ql_Check_XOR(const unsigned char *pData, unsigned int Length)
{
    unsigned char result = 0;
    unsigned int i = 0;

    if((NULL == pData) || (Length < 1))
    {
        return 0;
    }
    for(i = 0; i < Length; i++)
    {
        result ^= *(pData + i);
    }
```

```
    return result;
}
```

2.2. Standard Messages

This chapter explains the standard NMEA 0183 V4.11 messages supported by the module.

2.2.1. RMC

Recommended Minimum Specific GNSS Data. Time, date, position, course, and speed data provided by a GNSS receiver.

Type:

Output

Synopsis:

```
$<TalkerID>RMC,<UTC>,<Status>,<Lat>,<N/S>,<Lon>,<E/W>,<SOG>,<COG>,<Date>,<MagVar>,<MagVarDir>,<ModeInd>,<NavStatus>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Example	Description
$	Character	-	$	Each NMEA message starts with **$**.
<TalkerID>	String	-	GN	Talker identifier. See *Table 4: NMEA Talker ID.*
RMC	String	-	RMC	Recommended Minimum Specific GNSS Data.
<UTC>	hhmmss.sss	-	025159.000	Position fix UTC. hh: Hours (00–23) mm: Minutes (00–59) ss: Seconds (00–59) sss: Decimal fraction of seconds
<Status>	Character	-	A	Positioning system status. A = Data valid. V = Navigation receiver warning.
<Lat>	ddmm.mmmmmmmm	-	3149.29993210	Latitude. dd: Degrees (00–90) mm: Minutes (00–59) mmmmmmmm: Decimal fraction of

Field	Format	Unit	Example	Description
				minutes Null if invalid.
<N/S>	Character	-	N	North-south direction. N = North S = South Null if invalid.
<Lon>	dddmm.mmmmmmmm	-	11706.91264104	Longitude. ddd: Degrees (000–180) mm: Minutes (00–59) mmmmmmmm: Decimal fraction of minutes Null if invalid.
<E/W>	Character	-	E	East-west direction. E = East W = West Null if invalid.
<SOG>	Numeric	Knot	0.001	Speed over ground. Variable length. Null if invalid.
<COG>	Numeric	Degree	043.43	Course over ground. Variable length. Maximum value: 359.9. Null if invalid.
<Date>	ddmmyy	-	291123	Date. dd: Day of month mm: Month yy: Year
<MagVar>	-	-	-	Magnetic variation. Not supported.
<MagVarDir>	-	-	-	Direction of magnetic variation. Not supported.
<ModeInd>	Character	-	A	Mode indicator. A = Autonomous mode. Satellite system used in non-differential mode in position fix. D = Differential mode. Satellite system used in differential mode in position fix. Corrections from ground stations or Satellite Based Augmentation System (SBAS). E = Estimated (dead reckoning) mode. F = Float RTK. Satellite system used in RTK mode with floating integers. M = Manual input mode or Survey-in

Field	Format	Unit	Example	Description
				mode. N = No fix. Satellite system not used in position fix, or fix not valid. R = Real Time Kinematic (RTK). Satellite system used in RTK mode with fixed integers.
<NavStatus>	Character	-	V	Navigational status. Not supported. Always “V” (Navigational status information is not provided).
<Checksum>	Hexadecimal	-	*33	Checksum.
<CR><LF>	Character	-	-	Carriage return and line feed.

Example:

```
$GNRMC,025159.000,A,3149.29993210,N,11706.91264104,E,0.001,043.43,291123,,,A,V*33
```

2.2.2. GGA

Global Positioning System Fix Data. Time, position, and fix-related data for a GNSS receiver.

Type:

Output

Synopsis:

```
$<TalkerID>GGA,<UTC>,<Lat>,<N/S>,<Lon>,<E/W>,<Quality>,<NumSatUsed>,<HDOP>,<Alt>,M,<Sep>,M,<DiffAge>,<DiffStation>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Example	Description
$	Character	-	$	Each NMEA message starts with **$**.
<TalkerID>	String	-	GN	Talker identifier. See *Table 4: NMEA Talker ID.*
GGA	String	-	GGA	Global Positioning System Fix Data.
<UTC>	hhmmss.sss	-	025159.000	Position fix UTC. hh: Hours (00–23) mm: Minutes (00–59) ss: Seconds (00–59)

Field	Format	Unit	Example	Description
				sss: Decimal fraction of seconds
<Lat>	ddmm.mmmmmmmm	-	3149.29993210	Latitude. dd: Degrees (00–90) mm: Minutes (00–59) mmmmmmmm: Decimal fraction of minutes Null if invalid.
<N/S>	Character	-	N	North-south direction. N = North S = South Null if invalid.
<Lon>	dddmm.mmmmmmmm	-	11706.91264104	Longitude. ddd: Degrees (000–180) mm: Minutes (00–59) mmmmmmmm: Decimal fraction of minutes Null if invalid.
<E/W>	Character	-	E	East-west direction. E = East W = West Null if invalid.
<Quality>	Numeric, 1 digit	-	1	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. 7 = Manual input mode or Survey-in mode.
<NumSatUsed>[1)]	Numeric, 2 digits	-	16	Number of satellites in use.
<HDOP>	Numeric	-	1.26	Horizontal dilution of precision.
<Alt>	Numeric	Meter	97.250	Altitude above mean-sea-level (geoid).

Field	Format	Unit	Example	Description
M	Character	-	M	Unit of **<Alt>**. "M" = Meter.
<Sep>	Numeric	Meter	-4.945	Geoid separation (the difference between the earth ellipsoid surface and the mean-sea-level (geoid) surface defined by the reference datum used in the position solution).
M	Character	-	M	Unit of **<Sep>**. "M" = Meter.
<DiffAge>	Numeric	Second	-	Differential GPS data age. Null if invalid.
<DiffStation>	Numeric	-	-	Differential reference station ID. Range: 0000–4095. Null if invalid.
<Checksum>	Hexadecimal	-	*5A	Checksum.
<CR><LF>	Character	-	-	Carriage return and line feed.

Example:

```
$GNGGA,025159.000,3149.29993210,N,11706.91264104,E,1,16,1.26,97.250,M,-4.945,M,,*5A
```

NOTE

1. The NMEA 0183 specification indicates that the **GGA** messages are GPS specific. However, when the receiver is configured for multi-constellations, the content of a **GGA** message will be generated from the multi-constellation solution.
2. [1)] According to the NMEA 0183 specification, the number of satellites in use is between 00 and 12. However, in the multi-constellation solution, the number of satellites in use may exceed 12.

2.2.3. GSV

GNSS Satellites in View. The GSV sentence provides the number of satellites in view (SV), satellite ID numbers, elevation, azimuth, and SNR value, and contains maximum four satellites per transmission. Therefore, it may take several sentences to get complete information. The total number of sentences being transmitted and the sentence number are indicated in the first two data fields.

Type:

Output

Synopsis:

$<TalkerID>GSV,<TotalNumSen>,<SenNum>,<TotalNumSat>{,<SatID>,<SatElev>,<SatAz>,<SatCN0>},<SignalID>*<Checksum><CR><LF>

Parameter:

Field	Format	Unit	Example	Description
$	Character	-	$	Each NMEA message starts with **$**.
<TalkerID>	String	-	GP	Talker identifier. See *Table 4: NMEA Talker ID.*
GSV	String	-	GSV	GNSS Satellites in View.
<TotalNumSen>	Numeric	-	2	Total number of sentences. Range: 1–9.
<SenNum>	Numeric	-	1	Sentence number. Range: 1–**<TotalNumSen>**.
<TotalNumSat>	Numeric	-	05	Total number of satellites in view.
Start of repeat block. Repeat times: 1–4.				
<SatID>	Numeric	-	10	Satellite ID. See *Table 18: GNSS Satellites (NMEA) Numbering*.
<SatElev>	Numeric	Degree	77	Satellite elevation. Range: 00–90. Null if invalid.
<SatAz>	Numeric	Degree	300	Satellite azimuth, with true north as the reference plane. Range: 000–360. Null if invalid.
<SatCN0>	Numeric	dB-Hz	36	Satellite C/N_0. Range: 00–99. Null when not tracking.
End of repeat block.				
<SignalID>	Numeric	-	1	GNSS signal ID. See *Table 18: GNSS Satellites (NMEA) Numbering*.
<Checksum>	Hexadecimal	-	*67	Checksum.
<CR><LF>	Character	-	-	Carriage return and line feed.

Example:

$GPGSV,2,1,05,10,77,300,36,12,40,082,31,23,58,153,35,25,46,137,33,1*67
$GPGSV,2,2,05,32,45,316,34,1*52

```
$GPGSV,2,1,05,10,77,300,31,12,40,082,25,23,58,153,29,25,46,137,28,6*65
$GPGSV,2,2,05,32,45,316,25,6*55
$GPGSV,1,1,04,10,77,300,32,23,58,153,30,25,46,137,30,32,45,316,26,8*61
$GLGSV,1,1,03,67,57,036,37,68,30,328,34,78,53,184,27,1*4B
$GLGSV,1,1,03,67,57,036,31,68,30,328,27,78,53,184,31,3*4A
```

NOTE

GN cannot be used for **GSV** sentences. If satellites of multiple constellations are in view, **GSV** sentences are output with the corresponding talker ID for each constellation, respectively.

2.2.4. GSA

GNSS DOP and Active Satellites. GNSS receiver operating mode, satellites used in the navigation solution reported by the **GGA** sentence, and DOP values.

Type:

Output

Synopsis:

```
$<TalkerID>GSA,<Mode>,<FixMode>{,<SatID>},<PDOP>,<HDOP>,<VDOP>,<SystemID>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Example	Description
$	Character	-	$	Each NMEA message starts with **$**.
<TalkerID>	String	-	GN	Talker identifier. See *Table 4: NMEA Talker ID.*
GSA	String	-	GSA	GNSS DOP and Active Satellites.
<Mode>	Character	-	A	Selection of 2D or 3D fix. M = Manual, forced to operate in 2D or 3D mode. A = Automatic, allowed to automatically switch to 2D or 3D mode.
<FixMode>	Numeric	-	3	Fix mode. 1 = Fix not available 2 = 2D 3 = 3D
Start of repeat block. Repeat times: 12.				

Field	Format	Unit	Example	Description
<SatID>	Numeric	-	10	ID numbers of satellites used in solution. See *Table 18: GNSS Satellites (NMEA) Numbering*.
End of repeat block.				
<PDOP>	Numeric	-	2.38	Position dilution of precision. Maximum value: 99.99.
<HDOP>	Numeric	-	1.26	Horizontal dilution of precision. Maximum value: 99.99.
<VDOP>	Numeric	-	2.01	Vertical dilution of precision. Maximum value: 99.99.
<SystemID>	Numeric	-	1	GNSS system ID. See *Table 18: GNSS Satellites (NMEA) Numbering*.
<Checksum>	Hexadecimal	-	*0B	Checksum.
<CR><LF>	Character	-	-	Carriage return and line feed.

Example:

```
$GNGSA,A,3,10,12,23,25,32,,,,,,,,2.38,1.26,2.01,1*0B
$GNGSA,A,3,67,68,78,,,,,,,,,,2.38,1.26,2.01,2*0D
$GNGSA,A,3,21,,,,,,,,,,,,2.38,1.26,2.01,3*0F
$GNGSA,A,3,06,13,16,32,37,41,,,,,,,2.38,1.26,2.01,4*08
$GNGSA,A,3,,,,,,,,,,,,,2.38,1.26,2.01,5*0A
$GNGSA,A,3,03,,,,,,,,,,,,2.38,1.26,2.01,6*0A
```

NOTE

If less than 12 satellites are used for navigation, the remaining **<SatID>** fields are left empty. If more than 12 satellites are used for navigation, only the IDs of the first 12 satellites are output.

2.2.5. VTG

Course Over Ground & Ground Speed. The actual course and speed relative to the ground.

Type:

Output

Synopsis:

```
$<TalkerID>VTG,<COGT>,T,<COGM>,M,<SOGN>,N,<SOGK>,K,<ModeInd>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Example	Description
$	Character	-	$	Each NMEA message starts with **$**.
<TalkerID>	String	-	GN	Talker identifier. See *Table 4: NMEA Talker ID.*
VTG	String	-	VTG	Course Over Ground & Ground Speed.
<COGT>	Numeric	Degrees	043.43	Course over ground, in true north direction.
T	Character	-	T	Fixed field: true.
<COGM>	Numeric	Degrees	-	Course over ground (magnetic). Not supported.
M	Character	-	M	Fixed field: magnetic.
<SOGN>	Numeric	Knots	0.001	Speed over ground in knots.
N	Character	-	N	Fixed field: knot.
<SOGK>	Numeric	km/h	0.001	Speed over ground in kilometers per hour.
K	Character	-	K	Fixed field: kilometers per hour
<ModeInd>	Character	-	A	Mode indicator. A = Autonomous mode. Satellite system used in non-differential mode in position fix D = Differential mode. Satellite system used in differential mode in position fix. Corrections from ground stations or Satellite Based Augmentation System (SBAS) E = Estimated (dead reckoning) mode M = Manual input mode or Survey-in mode. N = No fix. Satellite system is not used for positioning, or positioning is invalid
<Checksum>	Hexadecimal	-	*23	Checksum.
<CR><LF>	Character	-	-	Carriage return and line feed.

Example:

```
$GNVTG,043.43,T,,M,0.001,N,0.001,K,A*23
```

2.2.6. GLL

Geographic Position – Latitude/Longitude. Latitude and longitude of the GNSS receiver position, the time of position fix and status.

Type:

Output

Synopsis:

```
$<TalkerID>GLL,<Lat>,<N/S>,<Lon>,<E/W>,<UTC>,<Status>,<ModeInd>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Example	Description
$	Character	-	$	Each NMEA message starts with **$**.
<TalkerID>	String	-	GN	Talker identifier. See *Table 4: NMEA Talker ID.*
GLL	String	-	GLL	Geographic Position – Latitude/Longitude.
<Lat>	ddmm.mmmmmmmm	-	3149.29993210	Latitude. dd: Degrees (00–90) mm: Minutes (00–59) mmmmmmmm: Decimal fraction of minutes Null if invalid.
<N/S>	Character	-	N	North-south direction. N = North S = South Null if invalid.
<Lon>	dddmm.mmmmmmmm	-	11706.91264104	Longitude. ddd: Degrees (000–180) mm: Minutes (00–59) mmmmmmmm: Decimal fraction of minutes Null if invalid.
<E/W>	Character	-	E	East-west direction. E = East W = West Null if invalid.
<UTC>	hhmmss.sss	-	025159.000	Position fix UTC. hh: Hours (00–23) mm: Minutes (00–59)

Field	Format	Unit	Example	Description
				ss: Seconds (00–59) sss: Decimal fraction of seconds
<Status>	Character	-	A	Positioning system status. A = Data valid. V = Data not valid.
<ModeInd>	Character	-	A	Mode indicator. A = Autonomous mode. Satellite system used in non-differential mode in position fix. D = Differential mode. Satellite system used in differential mode in position fix. Corrections from ground stations or Satellite Based Augmentation System (SBAS). E = Estimated (dead reckoning) mode. M = Manual input mode or Survey-in mode N = Data not valid.
<Checksum>	Hexadecimal	-	*45	Checksum.
<CR><LF>	Character	-	-	Carriage return and line feed.

Example:

```
$GNGLL,3149.29993210,N,11706.91264104,E,025159.000,A,A*45
```

2.2.7. GBS

GNSS Satellite Fault Detection. This sentence is used to support Receiver Autonomous Integrity Monitoring (RAIM). Given that a GNSS receiver is tracking enough satellites to perform integrity checks of the positioning quality of the position solution a sentence is needed to report the output of this process to other systems to advise the system user.

Type:

Output

Synopsis:

```
$<TalkerID>GBS,<UTC>,<LatExpErr>,<LonExpErr>,<AltExpErr>,<FailSatID>,<FailPr>,<EstBias>,<StdBias>,<SystemID>,<SignalID>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Example	Description
$	Character	-	$	Each NMEA message starts with **$**.
<TalkerID>	String	-	GN	Talker identifier. See *Table 4: NMEA Talker ID.*
GBS	String	-	GBS	GNSS Satellite Fault Detection.
<UTC>	hhmmss.sss	-	054915.000	Position fix UTC. hh: Hours (00–23) mm: Minutes (00–59) ss: Seconds (00–59) sss: Decimal fraction of seconds
<LatExpErr>	Numeric	meter	0.6	Expected error in latitude. Null if invaild.
<LonExpErr>	Numeric	meter	0.5	Expected error in longitude. Null if invaild.
<AltExpErr>	Numeric	meter	1.4	Expected error in altitude. Null if invaild.
<FailSatID>	Numeric	-	27	ID number of most likely failed satellite. Null if invalid.
<FailPr>	Numeric	-		Probability of missed detection for most likely failed satellite. Null if invalid.
<EstBias>	Numeric	meter	33.2	Estimate of bias in meters on most likely failed satellite. Null if invalid.
<StdBias>	Numeric	meter	20.2	Standard deviation of bias estimate. Null if invalid.
<SystemID>	Numeric	-	1	GNSS system ID. See *Table 18: GNSS Satellites (NMEA) Numbering*.
<SignalID>	Hexdecimal	-	1	GNSS signal ID. See *Table 18: GNSS Satellites (NMEA) Numbering*.
<Checksum>	Hexadecimal	-	*62	Checksum.
<CR><LF>	Character	-	-	Carriage return and line feed.

Example:

```
$GNGBS,054915.000,0.6,0.5,1.4,27,,33.2,20.2,1,1*62
```

2.2.8. GNS

GNSS Fix Data. Fix data for single or combined satellite navigation systems (GNSS). This sentence provides fix data for GPS, GLONASS, BDS, QZSS, NavIC (IRNSS) and possible future satellite systems, and systems combining these.

Type:

Output

Synopsis:

```
$<TalkerID>GNS,<UTC>,<Lat>,<N/S>,<Lon>,<E/W>,<ModeInd>,<NumSatUsed>,<HDOP>,<Alt>,<Sep>
,<DiffAge>,<DiffStation>,<NavStatus>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Example	Description
$	Character	-	$	Each NMEA message starts with **$**.
<TalkerID>	String	-	GN	Talker identifier. See *Table 4: NMEA Talker ID.*
GNS	String	-	GNS	GNSS Fix Data.
<UTC>	hhmmss.sss	-	020602.900	Position fix UTC. hh: Hours (00–23) mm: Minutes (00–59) ss: Seconds (00–59) sss: Decimal fraction of seconds
<Lat>	ddmm.mmmmmmmm	-	2516.14836731	Latitude. dd: Degrees (00–90) mm: Minutes (00–59) mmmmmmmm: Decimal fraction of minutes Null if invalid.
<N/S>	Character	-	N	North-south direction. N = North S = South Null if invalid.
<Lon>	dddmm.mmmmmmmm	-	11020.046795558	Longitude. ddd: Degrees (000–180) mm: Minutes (00–59) mmmmmmmm: Decimal fraction of minutes Null if invalid.

Field	Format	Unit	Example	Description
<E/W>	Character	-	E	East-west direction. E = East W = West Null if invalid.
<ModeInd>	Character	-	DDDDDD	Mode indicator. A = Autonomous mode. Satellite system used in non-differential mode in position fix. D = Differential mode. Satellite system used in differential mode in position fix. Corrections from ground stations or Satellite Based Augmentation System (SBAS). E = Estimated (dead reckoning) mode. F = Float RTK. Satellite system used in RTK mode with floating integers. M = Manual input mode or Survey-in mode N = No fix. Satellite system not used in position fix, or fix not valid. R = Real Time Kinematic (RTK). Satellite system used in RTK mode with fixed integers.
<NumSatUsed>	Numeric	-	54	Number of satellites in use. Range: 00-99.
<HDOP>	Numeric	-	0.29	Horizontal dilution of precision. The maximum value: 99.99. 99.99 if invalid.
<Alt>	Numeric	Meter	173	Antenna altitude above mean sea level (geoid). Null if invalid.
<Sep>	Numeric	Meter	-20.052	Geoid separation (the difference between the earth ellipsoid surface and the mean-sea-level (geoid) surface defined by the reference datum used in the position solution).
<DiffAge>	Numeric	-	-	Differential GPS data age. Note that this field is empty in case of an invalid value or not support.
<DiffStation>	Numeric	-	-	Differential reference station ID. Note that this field is empty in case of an invalid value or not support.
<NavStatus>	Character	-	V	Navigational status. Not supported. Always “V” (Navigational status information

Field	Format	Unit	Example	Description
				is not provided).
<Checksum>	Hexadecimal	-	*09	Checksum.
<CR><LF>	Character	-	-	Carriage return and line feed.

Example:

```
$GNGNS,020602.900,2516.14836731,N,11020.04679558,E,DDDDDD,54,0.29,173.293,-20.052,,,V*09
```

2.2.9. GST

GNSS Pseudorange Error Statistics. This sentence supports Receiver Autonomous Integrity Monitoring (RAIM). Pseudorange measurement error statistics can be translated in the position domain in order to give statistical measures of the quality of the position solution.

Type:

Output

Synopsis:

```
$<TalkerID>GST,<UTC>,<RMS_D>,<MajorD>,<MinorD>,<Orient>,<LatD>,<LonD>,<AltD>*<Checksum>
<CR><LF>
```

Parameter:

Field	Format	Unit	Example	Description
$	Character	-	$	Each NMEA message starts with **$**.
<TalkerID>	String	-	GN	Talker identifier. See *Table 4: NMEA Talker ID.*
GST	String	-	GST	GNSS Pseudorange Error Statistics.
<UTC>	hhmmss.sss	-	013902.500	Position fix UTC. hh: Hour (00–23) mm: Minute (00–59) ss: Second (00–59) sss: Decimal fraction of second
<RMS_D>	Numeric	Meter	5.387	RMS value of the standard deviation of the range inputs to the navigation process. Null if invalid.

Field	Format	Unit	Example	Description
<MajorD>	Numeric	Meter	3.734	Standard deviation of semi-major axis of error ellipse. Null if invalid.
<MinorD>	Numeric	Meter	2.172	Standard deviation of semi-minor axis of error ellipse. Null if invalid.
<Orient>	Numeric	Degree	143.088	Orientation of semi-major axis of error ellipse. Null if invalid.
<LatD>	Numeric	Meter	2.836	Standard deviation of latitude error. Null if invalid.
<LonD>	Numeric	Meter	3.258	Standard deviation of longitude error. Null if invalid.
<AltD>	Numeric	Meter	9.274	Standard deviation of altitude error. Null if invalid.
<Checksum>	Hexadecimal	-	*74	Checksum.
<CR><LF>	Character	-	-	Carriage return and line feed.

Example:

```
//Unfixed.
$GNGST,000001.000,,,,,,,*48
//fixed.
$GNGST,013902.500,5.387,3.734,2.172,143.088,2.836,3.258,9.274*74
```

2.2.10. ZDA

Time & Date. UTC, day, month, year and local time zone.

Type:

Output

Synopsis:

```
$<TalkerID>ZDA,<UTC>,<Day>,<Month>,<Year>,<LocalHour>,<LocalMin>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Example	Description
$	Character	-	$	Each NMEA message starts with **$**.

Field	Format	Unit	Example	Description
<TalkerID>	String	-	GN	Talker identifier. See *Table 4: NMEA Talker ID.*
ZDA	String	-	ZDA	Time & Date. UTC, day, month, year and local time zone.
<UTC>	hhmmss.sss	-	102210.014	Position fix UTC. hh: Hours (00–23) mm: Minutes (00–59) ss: Seconds (00–59) sss: Decimal fraction of seconds
<Day>	Numeric	Day	23	Day of month. Range: 01–31.
<Month>	Numeric	Month	12	Month. Range: 01–12.
<Year>	Numeric	Year	2021	Year.
<LocalHour>	Numeric	-	00	Local zone hours, 00 to ±13 hours. Null if invalid.
<LocalMin>	Numeric	-	00	Local zone minutes, 00 to +59 minutes. Null if invalid.
<Checksum>	Hexadecimal	-	*4E	Checksum
<CR><LF>	Character	-	-	Carriage return and line feed.

Example:

```
$GNZDA,102210.014,23,12,2021,00,00*4E
```

2.2.11. HDT

Actual vessel heading in degrees true produced by any device or system producing true heading.

Type:

Output

Synopsis:

```
$<TalkerID>HDT,<Heading>,T*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Example	Description
$	Character	-	$	Each NMEA message starts with **$**.

Field	Format	Unit	Example	Description
<TalkerID>	String	-	GN	Talker identifier. See *Table 4: NMEA Talker ID.*
HDT	String	-	HDT	Actual vessel heading in degrees true produced by any device or system producing true heading.
<Heading>	Numeric	Degree	15.621	Actual vessel heading. Range: [0, 360) Null if invalid.
T	Character	-	T	Fixed field. Always T.
<Checksum>	Hexadecimal	-	*1A	Checksum
<CR><LF>	Character	-	-	Carriage return and line feed.

Example:

```
$GNHDT,15.621,T*1A
```

NOTE

Only LG580P (03) supports the message.

2.2.12. THS

True Heading and Status.

Type:

Output

Synopsis:

```
$<TalkerID>THS,<Heading>,<Mode>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Example	Description
$	Character	-	$	Each NMEA message starts with **$**.
<TalkerID>	String	-	GN	Talker identifier. See *Table 4: NMEA Talker ID.*
THS	String	-	THS	True Heading and Status.

Field	Format	Unit	Example	Description
<Heading>	Numeric	Degree	15.621	Actual vessel heading. Range: [0, 360) Null if invalid.
<Mode>	Character	-	A	Mode indication. A = Autonomous E = Estimated (dead reckoning) V = Data not valid (including standby)
<Checksum>	Hexadecimal	-	*18	Checksum
<CR><LF>	Character	-	-	Carriage return and line feed.

Example:

```
$GNTHS,15.621,A*18
```

NOTE

Only LG580P (03) supports the message.

2.3. PQTM Messages

This chapter explains the PQTM messages (proprietary NMEA messages defined by Quectel) supported by LG290P (03), LG580P (03) and LG680P (03) modules.

Table 5: Error Codes

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

NOTE

To avoid uncertainties, you need to send **$PQTMSAVEPAR*5A** to save the configuration after setting parameters through the Set type command, and then restart the module to ensure that all configurations take effect. Otherwise, the module will restore default values after powering on.

2.3.1. PQTMVER

Outputs the firmware version.

Type:

Output

Synopsis:

```
$PQTMVER,<MsgVer>,<VerName>,<VerStr>,<BuildDate>,<BuildTime>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<MsgVer>	Numeric	-	Message version. Always 1 for this version.
<VerName>	String	-	Version name. Fixed at “MODULE”.
<VerStr>	String	-	Main version string.
<BuildDate>	yyyy/mm/dd	-	Firmware build date. yyyy: Year mm: Month dd: Day of month
<BuildTime>	hh:mm:ss	-	Firmware build time. hh: Hours mm: Minutes ss: Seconds

Example:

```
$PQTMVER,1,MODULE,LG290P03AANR01A03S,2024/04/30,10:53:07*32
```

NOTE

Upon each successful startup, the module will output this message first.

2.3.2. PQTMCOLD

Performs a cold start.

Type:

Command

Synopsis:

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

Parameter:

None

Example:

```
$PQTMCOLD*1C
```

2.3.3. PQTMWARM

Performs a warm start.

Type:

Command

Synopsis:

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

Parameter:

None

Example:

```
$PQTMWARM*11
```

2.3.4. PQTMHOT

Performs a hot start.

Type:

Command

Synopsis:

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

Parameter:

None

Example:

```
$PQTMHOT*4B
```

2.3.5. PQTMSRR

Performs a system reset and reboots the receiver.

Type:

Command

Synopsis:

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

Parameter:

None

Example:

```
$PQTMSRR*4B
```

2.3.6. PQTMUNIQID

Queries the module unique ID.

Type:

Command

Synopsis:

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

Parameter:

None

Result:

- If successful, the module returns:

```
$PQTMUNIQID,OK,<Length>,<ID>*<Checksum><CR><LF>
```

Parameter included in the result:

Field	Format	Unit	Description
<Length>	Numeric	Byte	Length of module unique ID.
<ID>	Hexadecimal	-	Module unique ID.

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 5: Error Codes*.

Example:

```
$PQTMUNIQID*16
$PQTMUNIQID,OK,16,81D62010EE0AF375BDF5952CDC3757A1*3E
```

2.3.7. PQTMSAVEPAR

Saves the configurations into NVM.

Type:

Command

Synopsis:

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

Parameter:

None

Result:

- If successful, the module returns:

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

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 5: Error Codes*.

Example:

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

2.3.8. PQTMRESTOREPAR

Restores the parameters configured by all commands to their default values. This command takes effect after restarting.

Type:

Command

Synopsis:

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

Parameter:

None

Result:

- If successful, the module returns:

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

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 5: Error Codes*.

Example:

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

2.3.9. PQTMVERNO

Queries the firmware version.

Type:

Command

Synopsis:

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

Parameter:

None

Result:

- If successful, the module returns:

```
$PQTMVERNO,<VerStr>,<BuildDate>,<BuildTime>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<VerStr>	String	-	Firmware version.
<BuildDate>	yyyy/mm/dd	-	Firmware build date. yyyy: Year mm: Month dd: Day of month
<BuildTime>	hh:mm:ss		Firmware build time. hh: Hours mm: Minutes ss: Seconds

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 5: Error Codes*.

Example:

```
$PQTMVERNO*58
$PQTMVERNO,LG290P03AANR01A03S,2024/04/30,10:53:07*18
```

2.3.10. PQTMCFGUART

Sets/gets the UART interface.

Type:

Set/Get

Synopsis:

```
//Set the current UART interface:
$PQTMCFGUART,W,<BaudRate>[,<DataBit>,<Parity>,<StopBit>,<FlowCtrl>]*<Checksum><CR><LF>
//Set the specified UART interface:
$PQTMCFGUART,W,<Index>,<BaudRate>[,<DataBit>,<Parity>,<StopBit>,<FlowCtrl>]*<Checksum><CR><LF>
//Get the configuration on the current UART interface or a specified UART interface:
$PQTMCFGUART,R[,<Index>]*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Index>	Numeric	-	UART interface index. 1 = UART1 2 = UART2 3 = UART3
<BaudRate>	Numeric	bps	UART baud rate. 9600 115200 230400 460800 921600
<DataBit>	Numeric	bit	UART data bit. 8 = 8 bits
<Parity>	Numeric	-	Parity. 0 = No parity 1 = Odd parity 2 = Even parity 3 = Mark 4 = Space
<StopBit>	Numeric	-	Stop bit(s). 1 = 1 stop bit 2 = 2 stop bits
<FlowCtrl>	Numeric	-	Flow control. 0 = None

Result:

- If successful, the module returns:

```
//Response to Set command:
$PQTMCFGUART,OK*<Checksum><CR><LF>
//Response to Get command:
$PQTMCFGUART,OK,<Index>,<BaudRate>,<DataBit>,<Parity>,<StopBit>,<FlowCtrl>*<Checksum><CR><LF>
```

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 5: Error Codes*.

Example:

```
//Configure the baud rate on the current UART interface:
$PQTMCFGUART,W,115200*18
$PQTMCFGUART,OK*60

//Configure the baud rate on UART1:
$PQTMCFGUART,W,1,115200*05
$PQTMCFGUART,OK*60

//Configure all parameters on the current UART interface:
$PQTMCFGUART,W,115200,8,0,1,0*11
$PQTMCFGUART,OK*60

//Configure all parameters on UART1:
$PQTMCFGUART,W,1,115200,8,0,1,0*0C
$PQTMCFGUART,OK*60

//Get the configuration on the current UART interface:
$PQTMCFGUART,R*36
$PQTMCFGUART,OK,1,115200,8,0,1,0*5F

//Get the configuration on UART1.
$PQTMCFGUART,R,1*2B
$PQTMCFGUART,OK,1,115200,8,0,1,0*5F
```

NOTE

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 Set command. If the default setting had been changed by Set command, contact Quectel Technical Support (support@quectel.com) to get the default setting if necessary.

2.3.11. PQTMCFGPPS

Sets/gets the PPS feature.

Type:

Set/Get

Synopsis:

```
//Set:
$PQTMCFGPPS,W,<Index>,<Enable>,<Duration>,<Mode>,<Polarity>,<Reserved>*<Checksum><CR><LF>
//Get:
$PQTMCFGPPS,R,<Index>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Index>	Numeric	-	PPS index. 1 = PPS1
<Enable>	Numeric	-	Enable/disable PPS output. 0 = Disable 1 = Enable Note that if **<Enable>** is set to 0, the fields after **<Enable>** should be omitted.
<Duration>	Numeric	ms	Pulse duration. Range: 0–900 (Default value: 100)
<Mode>	Numeric	-	PPS output mode. 1 = PPS always output 2 = PPS output only in 2D/3D fix mode
<Polarity>	Numeric	-	Pulse polarity. 0 = Low 1 = High
<Reserved>	Numeric	-	Reserved. Always 0.

Result:

- If successful, the module returns:

```
//Response to Set command:
$PQTMCFGPPS,OK*<Checksum><CR><LF>
//Response to Get command:
$PQTMCFGPPS,OK,<Index>,<Enable>,<Duration>,<Mode>,<Polarity>,<Reserved>*<Checksum><CR><LF>
```

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 5: Error Codes*.

Example:

```
//Set PPS1 feature:
$PQTMCFGPPS,W,1,1,100,1,1,0*73
$PQTMCFGPPS,OK*21

//Get PPS1 feature:
$PQTMCFGPPS,R,1*6A
$PQTMCFGPPS,OK,1,1,100,1,1,0*20

//Disable PPS1 feature:
$PQTMCFGPPS,W,1,0*73
$PQTMCFGPPS,OK*21
```

2.3.12. PQTMCFGPROT

Sets/gets the input and output protocol for a specified port.

Type:

Set/Get

Synopsis:

```
//Set:
$PQTMCFGPROT,W,<PortType>,<PortID>,<InputProt>,<OutputProt>*<Checksum><CR><LF>
//Get:
$PQTMCFGPROT,R,<PortType>,<PortID>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<PortType>	Numeric	-	Port type. 1 = UART
<PortID>	Numeric	-	Port ID. If **<PortType>** is set to 1, the **<PortID>** range: 1–3 1 = UART1 2 = UART2 3 = UART3
<InputProt>	Hexadecimal	-	Input protocol.(32 bit) Bit 0 = NMEA Bit 2 = RTCM3 When the port is UART1 to UART3, default input protocols are NMEA and RTCM3 (corresponding value: 00000005).
<OutputProt>	Hexadecimal	-	Output protocol. (32 bit) Bit 0 = NMEA Bit 2 = RTCM3 When the port is UART1 to UART 3, default output protocols are NMEA and RTCM3 (corresponding value: 00000005).

Result:

- If successful, the module returns:

```
//Response to Set command:
$PQTMCFGPROT,OK*<Checksum><CR><LF>
//Response to Get command:
$PQTMCFGPROT,OK,<PortType>,<PortID>,<InputProt>,<OutputProt>*<Checksum><CR><LF>
```

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 5: Error Codes*.

Example:

```
//Set:
$PQTMCFGPROT,W,1,1,00000005,00000005*38
$PQTMCFGPROT,OK*6B
//Get:
$PQTMCFGPROT,R,1,1*3D
$PQTMCFGPROT,OK,1,1,00000005,00000005*6B
```

2.3.13. PQTMCFGNMEADP

Sets/gets the decimal places of standard NMEA messages.

Type:

Set/Get

Synopsis:

```
//Set:
$PQTMCFGNMEADP,W,<UTC_DP>,<POS_DP>,<ALT_DP>,<DOP_DP>,<SPD_DP>,<COG_DP>*<Checksum><CR><LF>
//Get:
$PQTMCFGNMEADP,R*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<UTC_DP>	Numeric	-	Number of decimal places for UTC seconds in Standard NMEA messages. Range: 0–3. Default value: 3. 0 = No fractional part
<POS_DP>	Numeric	-	Number of decimal places for latitude and longitude in Standard NMEA messages. Range: 0–8. Default value: 8. 0 = No fractional part
<ALT_DP>	Numeric	-	Number of decimal places for altitude and geoidal separation in Standard NMEA messages. Range: 0–3. Default value: 3. 0 = No fractional part
<DOP_DP>	Numeric	-	Number of decimal places for DOP in Standard NMEA messages. Range: 0–3. Default value: 2. 0 = No fractional part
<SPD_DP>	Numeric	-	Number of decimal places for speed in Standard NMEA messages. Range: 0–3. Default value: 3. 0 = No fractional part
<COG_DP>	Numeric	-	Number of decimal places for COG in Standard NMEA messages. Range: 0–3. Default value: 2. 0 = No fractional part

Result:

- If successful, the module returns:

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

```
$PQTMCFGNMEADP,OK,<UTC_DP>,<POS_DP>,<ALT_DP>,<DOP_DP>,<SPD_DP>,<COG_DP>*<Checksum><CR><LF>
```

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 5: Error Codes*.

Example:

```
//Set:
$PQTMCFGNMEADP,W,3,8,3,2,3,2*39
$PQTMCFGNMEADP,OK*61

//Get:
$PQTMCFGNMEADP,R*37
$PQTMCFGNMEADP,OK,3,8,3,2,3,2*6A
```

2.3.14. PQTMEPE

Outputs the estimated position error.

Type:

Output

Synopsis:

```
$PQTMEPE,<MsgVer>,<EPE_North>,<EPE_East>,<EPE_Down>,<EPE_2D>,<EPE_3D>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<MsgVer>	Numeric	-	Message version. Always 2 for this version.
<EPE_North>	Numeric	Meter	Estimated north error.
<EPE_East>	Numeric	Meter	Estimated east error.
<EPE_Down>	Numeric	Meter	Estimated down error.
<EPE_2D>	Numeric	Meter	Estimated 2D positioning error.
<EPE_3D>	Numeric	Meter	Estimated 3D positioning error.

Example:

```
$PQTMEPE,2,1.000,1.000,1.000,1.414,1.732*52
```

2.3.15. PQTMCFGMSGRATE

Sets/gets the message output rate on the current interface or the specific interface.

Type:

Set/Get

Synopsis:

```
//Configure the message rate on current interface:
$PQTMCFGMSGRATE,W,<MsgName>,<Rate>[,<MsgVer/Offset>]*<Checksum><CR><LF>
//Configure the message rate for a specific interface:
$PQTMCFGMSGRATE,W,<PortType>,<PortID>,<MsgName/MsgID>,<Rate>[,<MsgVer/Offset>]*<Checksum><CR><LF>
//Read the message rate configuration on current interface:
$PQTMCFGMSGRATE,R,<MsgName>[,<MsgVer/Offset>]*<Checksum><CR><LF>
//Read the message rate configuration for a specific interface:
$PQTMCFGMSGRATE,R,<PortType>,<PortID>,<MsgName/MsgID>[,<MsgVer/Offset>]*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<PortType>	Numeric	-	Port type. 1 = UART
<PortID>	Numeric	-	Port ID. 1 = UART1. 2 = UART2. 3 = UART3.
<MsgName/MsgID>	String/Hex	-	Message name. See *Table 6: Supported Messages*.
<Rate>	Numeric	-	Message output rate. 0 = Not output. N = Output once every N position fix(es). For details on the range of N, See *Table 6: Supported Messages*.
<MsgVer/Offset>	Numeric	-	● The parameter is the message version for PQTM messages.

Field	Format	Unit	Description
			● The parameter is the time offset for RTCM MSM messages. For illustration of time offset, see *Figure 2: RTCM MSM Time Offset*. ● The parameter is omitted for others messages, such as standard NMEA messages, **RTCM3-1005**, **RTCM3-1006**, **RTCM3-1019**. Range: ● Range of PQTM message version depends on the specific message. ● Range of RTCM3 MSM time offset: 0 to **<Rate>** - 1.

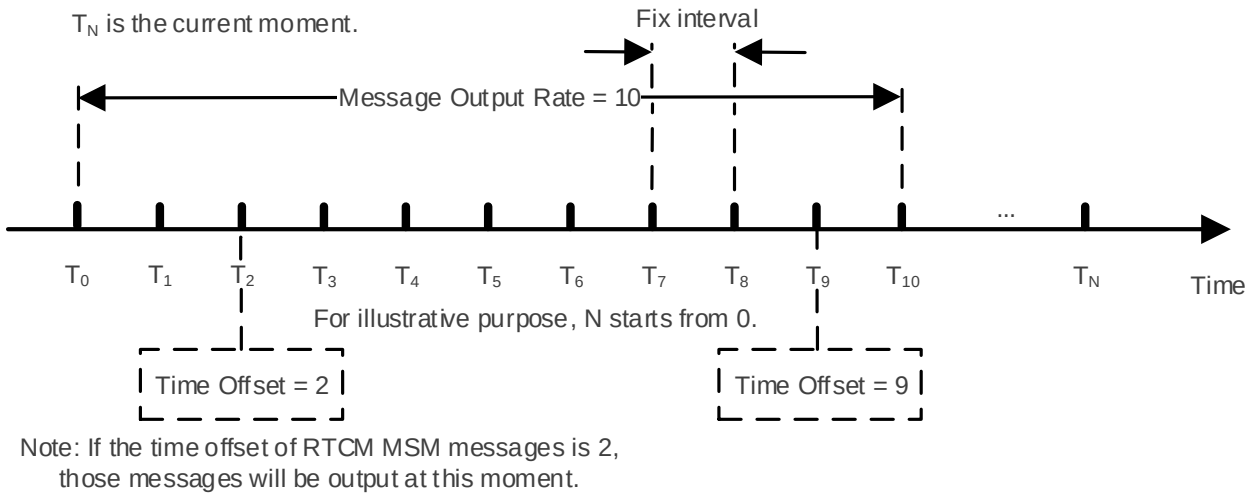

Figure 2: RTCM MSM Time Offset

Result:

● If successful, the module returns:

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

● If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 5: Error Codes*.

Example:

```
//Set the output rate of GGA message to once per position fix:
$PQTMCFGMSGRATE,W,GGA,1*0A
$PQTMCFGMSGRATE,OK*29

//Get the output rate of GGA message:
$PQTMCFGMSGRATE,R,GGA*12
$PQTMCFGMSGRATE,OK,GGA,1*59

//Set the output rate of PQTMEPE (version 2) message to once per position fix:
$PQTMCFGMSGRATE,W,PQTMEPE,1,2*1D
$PQTMCFGMSGRATE,OK*29

//Get the output rate of PQTMEPE (version 2) message:
$PQTMCFGMSGRATE,R,PQTMEPE,2*05
$PQTMCFGMSGRATE,OK,PQTMEPE,1,2*4E

//Set the output rate of RTCM3-1005 message to once per position fix:
$PQTMCFGMSGRATE,W,RTCM3-1005,1*59
$PQTMCFGMSGRATE,OK*29

//Get the output rate of RTCM3-1005 message:
$PQTMCFGMSGRATE,R,RTCM3-1005*41
$PQTMCFGMSGRATE,OK,RTCM3-1005,1*0A

//Set the output rate of RTCM3 GPS MSM message to once per position fix and time offset to 0:
$PQTMCFGMSGRATE,W,RTCM3-107X,1,0*2F
$PQTMCFGMSGRATE,OK*29

//Get the output rate of RTCM3 GPS MSM message:
$PQTMCFGMSGRATE,R,RTCM3-107X*2B
$PQTMCFGMSGRATE,OK,RTCM3-107X,1,0*7C

//Set the output rate of RTCM GPS EPH message to once per position fix:
$PQTMCFGMSGRATE,W,RTCM3-1019,1*54
$PQTMCFGMSGRATE,OK*29

//Get the output rate of RTCM GPS EPH message:
```

$PQTMCFGMSGRATE,R,RTCM3-1019*4C
$PQTMCFGMSGRATE,OK,RTCM3-1019,1*07

//Set the UART1 output rate of **GGA** message to once per position fix:
$PQTMCFGMSGRATE,W,1,1,GGA,1*0A
$PQTMCFGMSGRATE,OK*29

//Get the UART1 output rate of **GGA** message:
$PQTMCFGMSGRATE,R,1,1,GGA*12
$PQTMCFGMSGRATE,OK,1,1,GGA,1*59

//Set the UART1 output rate of **RAW-PPPB2B** message to once per position fix:
$PQTMCFGMSGRATE,W,1,1,0AB2,1,1*57
$PQTMCFGMSGRATE,OK*29

//Get the UART1 output rate of **RAW-PPPB2B** message
$PQTMCFGMSGRATE,R,1,1,0AB2,1,1*4F
$PQTMCFGMSGRATE,OK,1,1,0AB2,1,1*04

Table 6: Supported Messages

Message	Description	Range (N)
RMC	-	0–255
GGA	-	0–255
GSV	-	0–255
GSA	-	0–255
VTG	-	0–255
GLL	-	0–255
GBS	-	0–255
GNS	-	0–255
GST	-	0–255
ZDA	-	0–255
HDT	-	0–255

Message	Description	Range (N)
THS	-	0–255
PQTMEPE	-	0–255
PQTMVEL	-	0–255
PQTMGEOFENCESTATUS	-	0–255
PQTMTXT	-	0–255
PQTMSVINSTATUS	-	0–255
PQTMPVT	-	0–255
PQTMDOP	-	0–255
PQTMPL	-	0–255
PQTMODO	-	0–255
PQTMTAR	-	0–255
PQTMNAV	-	0–255
RTCM3-1005	-	1–1200
RTCM3-1006	-	1–1200
RTCM3-1033	-	1-1200
RTCM3-107X	GPS-MSM	1–1200
RTCM3-108X	GLONASS-MSM	1–1200
RTCM3-109X	Galileo-MSM	1–1200
RTCM3-111X	QZSS-MSM	1–1200
RTCM3-112X	BDS-MSM	1–1200
RTCM3-113X	NavIC/IRNSS-MSM	1–1200
RTCM3-1019	GPS-EPH	1
RTCM3-1020	GLONASS-EPH	1
RTCM3-1041	NavIC/IRNSS-EPH	1
RTCM3-1042	BDS-EPH	1
RTCM3-1044	QZSS-EPH	1

Message	Description	Range (N)
RTCM3-1046	Galileo I/NAV-EPH	1
RAW-PPPB2B (0AB2)	-	1
RAW-QZSSL6 (0AB6)	-	1
RAW-HASE6 (0AE6)	-	1

NOTE

1. If the configuration message is a PQTM message, use **<MsgVer>** field to specify the message version, otherwise an error will be returned.
2. If the configuration message is a standard NMEA message/RTCM message (excluding RTCM MSM message), or it is unnecessary to set the message version, the **<MsgVer>** field can be omitted.
3. All RTCM MSM messages have the same **<Rate>** value and only the last setting is valid. For details on RTCM MSM messages, see *Chapter 4 RTCM Protocol*. The epoch time is aligned to **<FixInterval>** × **<Rate>**. For details on **<FixInterval>**, see *Chapter 2.3.28 PQTMCFGFIXRATE*.
4. The output time for RTCM MSM messages is influenced by the **<Offset>** value via the formula: Output Time = **<FixInterval>** × **<Rate>** + **<FixInterval>** × **<Offset>**. If the **<Offset>** in other previously configured messages exceeds the range from 0 to **<Rate>** - 1 due to reconfiguring **<Rate>** in new messages, it is necessary to change the **<Offset>** in the conflicting messages to 0. This process is automatically implemented by software.
5. The RTCM EPH message output rate is independent of the **<FixInterval>**.
6. The message output rate of **GSA** and **GSV** messages is fixed at 1 Hz and independent of **<FixInterval>**.
7. When the PPP raw message (**RAW-PPPB2B, RAW-QZSSL6, RAW-HASE6**) updated, once the **<Rate>** of **$PQTMCFGMSGRATE** command is one, they will be outputted by the module.

2.3.16. PQTMVEL

Outputs the velocity information.

Type:

Output

Synopsis:

```
$PQTMVEL,<MsgVer>,<Time>,<VelN>,<VelE>,<VelD>,<GrdSpd>,<Spd>,<COG>,<GrdSpdAcc>,<SpdAcc>,<HeadingAcc>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<MsgVer>	Numeric	-	Message version. Always 1 for this version.
<Time>	hhmmss.sss	-	UTC time. hh: Hours (0–23) mm: Minutes (0–59) ss: Seconds (0–59) sss: Decimal fraction of seconds
<VelN>	Numeric	m/s	North velocity.
<VelE>	Numeric	m/s	East velocity.
<VelD>	Numeric	m/s	Down velocity.
<GrdSpd>	Numeric	m/s	2D speed.
<Spd>	Numeric	m/s	3D speed.
<COG>	Numeric	Degree	Course over ground. The maximum value is 359.999. Null if invalid.
<GrdSpdAcc>	Numeric	m/s	Estimated 2D speed accuracy.
<SpdAcc>	Numeric	m/s	Estimated 3D speed accuracy.
<COG_Acc>	Numeric	Degree	Estimated COG accuracy.

Example:

```
$PQTMVEL,1,154512.100,1.251,2.452,1.245,2.752,3.021,180.512,0.124,0.254,0.250*67
```

2.3.17. PQTMCFGGEOFENCE

Sets/gets geofence feature.

Type:

Set/Get

Synopsis:

```
//Set:
$PQTMCFGGEOFENCE,W,<Index>,<Mode>,<Reserved>,<Shape>,<Lat0>,<Lon0>,<Lat1/Radius>[,<Lon1>,<Lat2>,<Lon2>,<Lat3>,<Lon3>]*<Checksum><CR><LF>
```

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

Parameter:

Field	Format	Unit	Description
<Index>	Numeric	-	Geofence index. Range: 0–3
<Mode>	Numeric	-	Geofence mode. 0 = Disable 1 = Enable
<Reserved>	Numeric	-	Reserved. Always 0.
<Shape>	Numeric	-	Geofence shape. 0 = Circle defined by the center and the radius 1 = Circle defined by the center and a point on the circle 2 = Triangle 3 = Quadrangle (such as square, rectangle, trapezium)
<Lat0>	Numeric	Degree	Latitude of the first point.
<Lon0>	Numeric	Degree	Longitude of the first point.
<Lat1/Radius>	Numeric	Degree/Meter	If the geofence shape is a circle with a certain radius, this value will be the radius of the circle; otherwise, this value will be the latitude of the second point.
<Lon1>	Numeric	Degree	Longitude of the second point.
<Lat2>	Numeric	Degree	Latitude of the third point.
<Lon2>	Numeric	Degree	Longitude of the third point.
<Lat3>	Numeric	Degree	Latitude of the fourth point.
<Lon3>	Numeric	Degree	Longitude of the fourth point.

Result:

- If successful, the module returns:

```
//Response to Set command:
$PQTMCFGGEOFENCE,OK*<Checksum><CR><LF>
//Response to Get command:
$PQTMCFGGEOFENCE,OK,<Index>,<Mode>,<Reserved>,<Shape>,<Lat0>,<Lon0>,<Lat1/Radius>[,<Lon1>,<Lat2>,<Lon2>,<Lat3>,<Lon3>]*<Checksum><CR><LF>
```

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 5: Error Codes*.

Example:

```
//Set and enable a geofence.
$PQTMCFGGEOFENCE,W,0,1,0,0,31.451248,117.451245,100.5*18
$PQTMCFGGEOFENCE,OK*74

//Disable a geofence.
$PQTMCFGGEOFENCE,W,0,0*27
$PQTMCFGGEOFENCE,OK*74

//Get the configuration of a geofence.
$PQTMCFGGEOFENCE,R,0*3E
//Geofence is enabled whose shape is a circle defined by the center and the radius.
$PQTMCFGGEOFENCE,OK,0,1,0,0,31.451248,117.451245,100.5*4B

//Get the configuration of a geofence.
$PQTMCFGGEOFENCE,R,0*3E
//Geofence is disabled.
$PQTMCFGGEOFENCE,OK,0,0*74
```

NOTE

1. When the geofence is disabled, the fields after **<Mode>** in the Set command should be omitted. If geofence has been disabled, the fields after **<Mode>** in the module response will be omitted when retrieving the configuration of the geofence. See the example above for details.
2. If the number of input points exceeds the number of points that the shape should have, an error will be returned.
3. The latitude range is [-90,+90], where negatives indicate south latitude. The longitude range is [-180,+180], where negative values indicate west longitude.

2.3.18. PQTMGEOFENCESTATUS

Outputs the geofence status.

Type:

Output

Synopsis:

```
$PQTMGEOFENCESTATUS, <MsgVer>,<Time>{,<StateN>}*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<MsgVer>	Numeric	-	Message version. Always 1 for this version.
<Time>	hhmmss.sss	-	UTC time. hh: Hours (0–23) mm: Minutes (0–59) ss: Seconds (0–59) sss: Decimal fraction of seconds
Start of repeat block. Repeat times: 4.			
<StateN>	Numeric	-	Geofence state: (N is the number of **<State>**. Range of N: 0–3.) 0 = Unknow 1 = Inside the geofence 2 = Outside the geofence Note: If the module did not get a fixed, the **<StateN>** should be 0.
End of repeat block.			

Example:

```
$PQTMGEOFENCESTATUS,1,124521.000,1,2,2,2*27
```

2.3.19. PQTMGNSSSTART

Starts GNSS engine.

Type:

Command

Synopsis:

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

Parameter:

None

Result:

- If successful, the module returns:

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

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 5: Error Codes*.

Example:

```
$PQTMGNSSSTART*51
$PQTMGNSSSTART,OK*79
```

2.3.20. PQTMGNSSSTOP

Stops GNSS engine.

Type:

Command

Synopsis:

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

Parameter:

None

Result:

- If successful, the module returns:

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

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 5: Error Codes*.

Example:

```
$PQTMGNSSSTOP*09
$PQTMGNSSSTOP,OK*21
```

2.3.21. PQTMTXT

Outputs short text messages. Long text messages can be transmitted by multiple messages.

Type:

Output

Synopsis:

```
$PQTMTXT,<MsgVer>,<TotalNumSen>,<SenNum>,<TextID>,<Text>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<MsgVer>	Numeric	-	Message version. Always 1 for this version.
<TotalNumSen>	Numeric	-	Total number of sentences. Range: 01–99.
<SenNum>	Numeric	-	Sentence number. Range: 01–**<TotalNumSen>**.
<TextID>	Numeric	-	Text identifier. 01 = Notice 02 = Warning 03 = Error
<Text>	String	-	Text message. Up to 57 characters including any code delimiters.

Example:

```
//Outputs debug data.
$PQTMTXT,1,01,01,01,0x105f0cf810417c00*1B
```

2.3.22. PQTMCFGSVIN

Sets/gets the Survey-in feature. This feature can determine the antenna location either by Survey-in mode or Fixed mode.

In order to work as a base station, the module external antenna should be mounted on a static point (try to mount it with a clear sky visibility). The antenna accurate coordinate location can be acquired through a self-survey process. The Survey-in mode (**<Mode>** = 1) determines the receiver's position by building a weighted mean of all valid 3D positioning solutions. You can set values of **<CFG_CNT>** and **<3D_AccLimit>** to define the minimum positioning times and 3D position standard deviation used for the position estimation. The Fixed mode (**<Mode>** = 2) requires user to manually enter the receiver position coordinates. Any error in the base station position will translate directly into rover position error.

Type:

Set/Get

Synopsis:

```
//Set:
$PQTMCFGSVIN,W,<Mode>,<CFG_CNT>,<3D_AccLimit>,<ECEF_X>,<ECEF_Y>,<ECEF_Z>[,<Distance>]*<Checksum><CR><LF>
//Get:
$PQTMCFGSVIN,R*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Mode>	Numeric	-	Receiver mode. 0 = Disable 1 = Survey-in mode 2 = Fixed mode (APR position is given in ECEF Coordinate)
<CFG_CNT>	Numeric	-	Minimum positioning times in Survey-in mode. Range: 0–86400.
<3D_AccLimit>	Numeric	Meter	Limit 3D positioning accuracy in Survey-in mode. Range: [0.0, 1000.0]. Default: 0.0 If this field is 0, it means there is no limit on 3D positioning accuracy.
<ECEF_X>	Numeric	Meter	WGS84 ECEF X coordinate.
<ECEF_Y>	Numeric	Meter	WGS84 ECEF Y coordinate.
<ECEF_Z>	Numeric	Meter	WGS84 ECEF Z coordinate.
<Distance>	Numeric	Meter	The distance between current survey-in result and last result. When the calculated value is greater than the set value, the current Survey-in result is output; when the calculated value is less than the set value, the previous Survey-in result is output. Only work when <Mode> = 1.

Field	Format	Unit	Description
			Range: [0.0, 10.0] Default: 0.0 (always use the latest survey-in result)

Result:

- If successful, the module returns:

```
//Response to Set command :
$PQTMCFGSVIN,OK*<Checksum><CR><LF>
//Response to Get command :
$PQTMCFGSVIN,OK,<Mode>,<CFG_CNT>,<3D_AccLimit>,<ECEF_X>,<ECEF_Y>,<ECEF_Z>[,<Distance>]*<Checksum><CR><LF>
```

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 5: Error Codes*.

Example:

- Survey-in mode

```
//Set without distance:
$PQTMCFGSVIN,W,1,3600,1.2,0,0,0*0A
$PQTMCFGSVIN,OK*70

//Get:
$PQTMCFGSVIN,R*26
$PQTMCFGSVIN,OK,1,3600,1.2,0.0000,0.0000,0.0000,0.0*75

//Set with distance:
$PQTMCFGSVIN,W,1,3600,1.2,0,0,0,2.5*0F
$PQTMCFGSVIN,OK*70

//Get:
$PQTMCFGSVIN,R*26
$PQTMCFGSVIN,OK,1,3600,1.2,0.0000,0.0000,0.0000,2.5*72
```

- Fixed mode

```
//Set
$PQTMCFGSVIN,W,2,0,0,-2519265.0514,4849534.9045,3277834.6432*2A
```

$PQTMCFGSVIN,OK*70

//Get:

$PQTMCFGSVIN,R*26

$PQTMCFGSVIN,OK,2,0,0.0,-2519265.0514,4849534.9045,3277834.6432,0.0*65

NOTE

For more examples about Survey-in feature, see *document [1] application note*.

2.3.23. PQTMSVINSTATUS

Outputs the Survey-in status.

Type:

Output

Synopsis:

$PQTMSVINSTATUS,<MsgVer>,<TOW>,<Valid>,<Res0>,<Res1>,<Obs>,<CfgDur>,<MeanX>,<MeanY>,<MeanZ>,<MeanAcc>*<Checksum><CR><LF>

Parameter:

Field	Format	Unit	Description
<MsgVer>	Numeric	-	Message version. Always 1 for this version.
<TOW>	Numeric	ms	GPS time of week.
<Valid>	Numeric	-	Survey-in position validity flag. 0 = Invalid or <Mode>=0 on **PQTMCFGSVIN**. 1 = In-progress 2 = Valid
<Res0>	Numeric	-	Reserved. Always null.
<Res1>	Numeric	-	Reserved.
<Obs>	Numeric	-	Number of position observations used during Survey-in.
<CfgDur>	Numeric	-	Same as **<CFG_CNT>** field (minimum positioning times in Survey-in mode) configured via **PQTMCFGSVIN** command.

Field	Format	Unit	Description
<MeanX>	Numeric	Meter	Current Survey-in mean position along X axis of ECEF coordinate system.
<MeanY>	Numeric	Meter	Current Survey-in mean position along Y axis of ECEF coordinate system.
<MeanZ>	Numeric	Meter	Current Survey-in mean position along Z axis of ECEF coordinate system.
<MeanAcc>	Numeric	Meter	Current Survey-in mean position accuracy.

Example:

```
$PQTMSVINSTATUS,1,1000,1,,01,20,100,-2484434.3645,4875976.9741,3266161.3412,1.2415*3C
```

NOTE

The module must be Base station mode to execute this command. For details on base station mode, see *Chapter 2.3.25 PQTMCFGRCVRMODE*.

2.3.24. PQTMPVT

Outputs the PVT (GNSS only) result.

Type:

Output

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<MsgVer>	Numeric	-	Message version. Always 1 for this version.
<TOW>	Numeric	ms	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 (0–23) mm: Minutes (0–59) ss: Seconds (0–59) sss: Decimal fraction of seconds
<Res>	Numeric	-	Reserved.
<FixType>	Numeric	-	Fix mode. 0 = No fix. 1 = Reserved. 2 = 2D fix. 3 = 3D fix.
<NumSV>	Numeric	-	Number of satellites in use.
<LeapS>	Numeric	Second	Leap seconds. Null if invalid.
<Lat>	Numeric	Degree	Latitude. Null if invalid.
<Lon>	Numeric	Degree	Longitude. Null if invalid.
<Alt>	Numeric	Meter	Altitude above mean sea level. Null if invalid.
<Sep>	Numeric	Meter	Geoidal separation (the difference between the WGS84 earth ellipsoid surface and the mean-sea-level surface). Null if invalid.
<VelN>	Numeric	m/s	North velocity. Null if invalid.
<VelE>	Numeric	m/s	East velocity. Null if invalid.
<VelD>	Numeric	m/s	Down velocity. Null if invalid.
<Spd>	Numeric	m/s	Ground speed. Null if invalid.
<Heading>	Numeric	Degree	Heading. Null if invalid.
<COG>	Numeric	Degree	Course over ground. Null if invalid.
<HDOP>	Numeric	-	Horizontal dilution of precision. 99.99 if this field is invalid.
<PDOP>	Numeric	-	Position (3D) dilution of precision. 99.99 if this field is invalid.

Example:

```
//No fix:
$PQTMPVT,1,1000,20221225,163355.000,,0,00,,,,,,,,,,99.99,99.99*79
//3D fix:
```

$PQTMPVT,1,31075000,20221225,083737.000,,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*51

2.3.25. PQTMCFGRCVRMODE

Sets/gets the receiver working mode.

Type:

Set/Get

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<Mode>	Numeric	-	Receiver working mode. 0 = Unknow 1 = Rover. When set to this mode, the receiver will restore to default NMEA message output state. 2 = Base station. When set to this mode, the receiver will automatically disable NMEA message output and enable RTCM MSM4 and **RTCM3-1005** message output.

Result:

- If successful, the module returns:

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

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 5: Error Codes*.

Example:

```
//Set:
$PQTMCFGRCVRMODE,W,2*29
$PQTMCFGRCVRMODE,OK*64

//Get:
$PQTMCFGRCVRMODE,R*32
$PQTMCFGRCVRMODE,OK,2*7A
```

NOTE

After switching the operating mode, save the configuration and then reset the module. Otherwise, it will continue to operate in the original mode.

2.3.26. PQTMDEBUGON

Enables debug log output. The debug state can be saved by **PQTMSAVEPAR** command, see *Chapter 2.3.7 PQTMSAVEPAR*.

Type:

Command

Synopsis:

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

Parameter:

None

Result:

- If successful, the module returns:

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

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 5: Error Codes*.

Example:

```
$PQTMDEBUGON*48
$PQTMDEBUGON,OK*60
```

2.3.27. PQTMDEBUGOFF

Disables debug log output. The debug state can be saved by **PQTMSAVEPAR** command, see *Chapter 2.3.7 PQTMSAVEPAR*.

Type:

Command

Synopsis:

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

Parameter:

None

Result:

- If successful, the module returns:

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

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 5: Error Codes*.

Example:

```
$PQTMDEBUGOFF*06
$PQTMDEBUGOFF,OK*2E
```

2.3.28. PQTMCFGFIXRATE

Sets/gets the fix interval.

Type:

Set/Get

Synopsis:

```
//Set:
$PQTMCFGFIXRATE,W,<FixInterval>*<Checksum><CR><LF>
//Get:
$PQTMCFGFIXRATE,R*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<FixInterval>	Numeric	ms	Fix interval

Result:

- If successful, the module returns:

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

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 5: Error Codes*.

Example:

```
//Set:
$PQTMCFGFIXRATE,W,1000*59
$PQTMCFGFIXRATE,OK*27

//Get:
$PQTMCFGFIXRATE,R*71
$PQTMCFGFIXRATE,OK,1000*0A
```

NOTE

The fix rate of the module is 1 Hz and cannot be changed in Base station mode. In Rover mode, the fix rate is 10 Hz before changing the default value. For details on base station and Rover, see *Chapter 2.3.25 PQTMCFGRCVRMODE*.

2.3.29. PQTMCFGRTK

Sets/gets the RTK mode.

Type:

Set/Get

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<DiffMode>	Numeric	-	Differential mode. 0 = Disable RTK/RTD feature. Differential data is not used. 1 = Auto mode. 2 = RTD only mode. Only pseudoranges is used.
<RelMode>	Numeric	-	Absolute/relative mode. 1 = Absolute mode, ensure absolute position accuracy. 2 = Relative mode, ensure relative position accuracy. Note: This field only takes effect when **<DiffMode>** = 1 and the module enters the RTK only mode.
<Timeout>	Numeric	Second	The max differential age of RTK fix. Range: [1, 600] Default: 120.

Result:

- If successful, the module returns:

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

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 5: Error Codes*.

Example:

```
//Set without timeout:
$PQTMCFGRTK,W,1,1*6C
$PQTMCFGRTK,OK*3F

//Get:
$PQTMCFGRTK,R*69
$PQTMCFGRTK,OK,1,1,120*20

//Set with timeout:
$PQTMCFGRTK,W,1,1,60*46
$PQTMCFGRTK,OK*3F

//Get:
$PQTMCFGRTK,R*69
$PQTMCFGRTK,OK,1,1,60*15
```

2.3.30. PQTMCFGCNST

Sets/gets the constellation configuration.

Type:

Set/Get

Synopsis:

```
//Set:
$PQTMCFGCNST,W,<GPS>,<GLONASS>,<Galileo>,<BDS>,<QZSS>,<NavIC>*<Checksum><CR><LF>
//Get:
$PQTMCFGCNST,R*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<GPS>	Numeric	-	Enable/disable GPS. 0 = Disable 1 = Enable
<GLONASS>	Numeric	-	Enable/disable GLONASS. 0 = Disable 1 = Enable
<Galileo>	Numeric	-	Enable/disable Galileo. 0 = Disable 1 = Enable
<BDS>	Numeric	-	Enable/disable BDS. 0 = Disable 1 = Enable
<QZSS>	Numeric	-	Enable/disable QZSS. 0 = Disable 1 = Enable
<NavIC>	Numeric	-	Enable/disable NavIC. 0 = Disable. 1 = Enable.

Result:

- If successful, the module returns:

```
//Response to Set command:
$PQTMCFGCNST,OK*<Checksum><CR><LF>
//Response to Get command:
$PQTMCFGCNST,OK,<GPS>,<GLONASS>,<Galileo>,<BDS>,<QZSS>,<NavIC>*<Checksum><CR><LF>
```

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 5: Error Codes*.

Example:

```
//Set the constellation configuration.
$PQTMCFGCNST,W,1,1,1,1,0,0*2B
$PQTMCFGCNST,OK*78
```

```
//Get the constellation configuration.
$PQTMCFGCNST,R*2E
$PQTMCFGCNST,OK,1,1,1,1,0,0*78
```

NOTE

It is not supported to disable all constellations at the same time.

2.3.31. PQTMDOP

Outputs dilution of precision.

Type:

Output

Synopsis:

```
$PQTMDOP,<MsgVer>,<TOW>,<GDOP>,<PDOP>,<TDOP>,<VDOP>,<HDOP>,<NDOP>,<EDOP>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<MsgVer>	Numeric	-	Message version. Always 1 for this version.
<TOW>	Numeric	ms	Time of week. Null if invalid.
<GDOP>	Numeric	-	Geometric dilution of precision. 99.99 if this field is invalid.
<PDOP>	Numeric	-	Position (3D) dilution of precision. 99.99 if this field is invalid.
<TDOP>	Numeric	-	Time dilution of precision. 99.99 if this field is invalid.
<VDOP>	Numeric	-	Vertical dilution of precision. 99.99 if this field is invalid.
<HDOP>	Numeric	-	Horizontal dilution of precision. 99.99 if this field is invalid.
<NDOP>	Numeric	-	Northing dilution of precision. 99.99 if this field is invalid.
<EDOP>	Numeric	-	Easting dilution of precision. 99.99 if this field is invalid.

Example:

```
//Fixed:
$PQTMDOP,1,570643000,1.01,0.88,0.49,0.73,0.50,0.36,0.35*7C

//No fix:
$PQTMDOP,1,,99.99,99.99,99.99,99.99,99.99,99.99,99.99*70
```

2.3.32. PQTMPL

Outputs protection level information.

Type:

Output

Synopsis:

```
$PQTMPL,<MsgVer>,<TOW>,<PUL>,<Res1>,<Res2>,<PL_PosN>,<PL_PosE>,<PL_PosD>,<PL_VelN>,<PL_VelE>,<PL_VelD>,<Res3>,<Res4>,<PL_Time>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<MsgVer>	Numeric	-	Message version. Always 1 for this version.
<TOW>	Numeric	ms	Time of week. Null if invalid.
<PUL>	Numeric	%	Probability of uncertainty level per epoch.
<Res1>	Numeric	-	Reserved. Always 1.
<Res2>	Numeric	-	Reserved. Always 1.
<PL_PosN>	Numeric	mm	Protection level of north position. Null if invalid.
<PL_PosE>	Numeric	mm	Protection level of east position. Null if invalid.
<PL_PosD>	Numeric	mm	Protection level of down position. Null if invalid.
<PL_VelN>	Numeric	mm/s	Protection level of north velocity. Null if invalid.
<PL_VelE>	Numeric	mm/s	Protection level of east velocity. Null if invalid.

Field	Format	Unit	Description
<PL_VelD>	Numeric	mm/s	Protection level of down velocity. Null if invalid.
<Res3>	Numeric		Reserved. Always null.
<Res4>	Numeric		Reserved. Always null.
<PL_Time>	Numeric	ns	Protection level of time. Null if invalid.

Example:

```
$PQTMPL,1,55045200,5.00,1,1,2879,2718,4766,5344,4323,10902,,,*1C
```

2.3.33. PQTMCFGODO

Sets/gets the odometer feature.

Type:

Set/Get

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<State>	Numeric	-	Odometer feature state. 0 = Disabled 1 = Enabled
<InitDist>	Numeric	Meter	Initial distance. Default value: 0.

Result:

- If successful, the module returns:

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

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

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 5: Error Codes*.

Example:

```
//Set odometer feature:
$PQTMCFGODO,W,1,10.5*4E
$PQTMCFGODO,OK*36

//Get odometer feature:
$PQTMCFGODO,R*60
$PQTMCFGODO,OK,1,10.5*1D
```

2.3.34. PQTMRESETODO

Resets the accumulated distance recorded by the odometer.

Type:

Command

Synopsis:

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

Parameter:

None

Result:

- If successful, the module returns:

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

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 5: Error Codes*.

Example:

```
$PQTMRESETODO*09
$PQTMRESETODO,OK*21
```

NOTE

1. Reset the accumulated distance recorded by the odometer with **PQTMRESETODO** command or power off the module. Disabling the odometer feature with **PQTMCFGODO** command when the module is still working will stop distance calculation, but it cannot reset the distance to zero.
2. If this command is sent when the position is lost, the odometer will no longer accumulate the current lost position period distance until two new positioning points are regained.

2.3.35. PQTMODO

Outputs the odometer information.

Type:

Output

Synopsis:

```
$PQTMODO,<MsgVer>,<Time>,<State>,<Dist>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<MsgVer>	Character	-	Message version. Always 1 for this version.
<Time>	hhmmss.sss	-	UTC time. hh: Hours (00–23) mm: Minutes (00–59) ss: Seconds (00–59) sss: Decimal fraction of seconds
<State>	Numeric	-	Odometer status. 0 = Disabled 1 = Enabled
<Dist>	Numeric	Meter	Distance since last reset. The distance equals to the accumulated distance and the initial distance configured via **<InitDist>** in **PQTMCFGODO** command.

Example:

```
$PQTMODO,1,120635.000,1,112.3*6E
```

NOTE

1. **<Dist>** in **PQTMODO** represents the sum of **<InitDist>** value set in **PQTMCFGODO** and accumulated distance. The accumulated distance starts from 0 m and resets to 0 m after a power outage or when cleared with **PQTMRESETODO**. If **<InitDist>** value in the **PQTMCFGODO** is modified, the actual **<Dist>** output in **PQTMODO** will reflect the sum of the accumulated distance and the new **<InitDist>** value, as shown below:
   <Dist> = Accumulated Distance + **<InitDist>**.
2. Accumulated distance cannot be saved to NVM.

2.3.36. PQTMCFGSIGNAL

Sets/gets GNSS signal mask.

Type:

Set/Get

Synopsis:

```
//Set:
$PQTMCFGSIGNAL,W,<GPS_Sig>,<GLO_Sig>,<GAL_Sig>,<BDS_Sig>,<QZS_Sig>,<NAC_Sig>*<Checksum><CR><LF>
//Get:
$PQTMCFGSIGNAL,R*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<GPS_Sig>	Hexadecimal	-	GPS signal mask. 0 = Disable 1 = Enable Bit 0 = L1 C/A Bit 1 = L2C Bit 2 = L5-Q Default value: 0x07
<GLO_Sig>	Hexadecimal	-	GLONASS signal mask. 0 = Disable 1 = Enable Bit 0 = G1 C/A

Field	Format	Unit	Description
			Bit 1 = G2 C/A Default value: 0x03 (LG290P (03) and LG680P (03)) Default value: 0x00 (LG580P (03))
<GAL_Sig>	Hexadecimal	-	Galileo signal mask. 0 = Disable 1 = Enable Bit 0 = E1 Bit 1 = E5a Bit 2 = E5b Bit 3 = E6 Default value: 0x0F (LG290P (03) and LG680P (03)) Default value: 0x07 (LG580P (03))
<BDS_Sig>	Hexadecimal	-	BDS signal mask. 0 = Disable 1 = Enable Bit 0 = B1I Bit 1 = B2I Bit 2 = B3I Bit 3 = B1C Bit 4 = B2a Bit 5 = B2b Default value: 0x3F (LG290P (03) and LG680P (03)) Default value: 0x3B (LG580P (03))
<QZS_Sig>	Hexadecimal	-	QZSS signal mask. 0 = Disable 1 = Enable Bit 0 = L1 C/A Bit 1 = L2C Bit 2 = L5-Q Bit 3 = L6 Default value: 0x0F
<NAC_Sig>	Hexadecimal	-	NavIC signal mask. 0 = Disable 1 = Enable Bit 0 = L5 Default value: 0x01

Result:

- If successful, the module returns:

```
//Response to Set command:
$PQTMCFGSIGNAL,OK*<Checksum><CR><LF>
//Response to Get command:
$PQTMCFGSIGNAL,OK,<GPS_Sig>,<GLO_Sig>,<GAL_Sig>,<BDS_Sig>,<QZS_Sig>,<NAC_Sig>*<Checksum><CR><LF>
```

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 5: Error Codes*.

Example:

```
//Set GNSS signal mask:
$PQTMCFGSIGNAL,W,7,3,F,3F,7,1*0E
$PQTMCFGSIGNAL,OK*6C

//Get GNSS signal mask:
$PQTMCFGSIGNAL,R*3A
$PQTMCFGSIGNAL,OK,07,03,0F,3F,07,01*6D
```

NOTE

1. The L1 frequency bands of LG290P (03), LG580P (03) and LG680P(03) module cannot be disabled.
2. GPS L1C and QZSS L1C frequency bands are under development. Contact Quectel Technical Support (support@quectel.com) for details.
3. The priority of GNSS configuration commands: **PQTMCFGCNST** > **PQTMCFGSIGNAL** > **PQTMCFGSAT**. For instance, if the GPS constellation is disabled by **PQTMCFGCNST** command, the enabling of GPS constellation and the frequency bands in **PQTMCFGSIGNAL** and **PQTMCFGSAT** commands will be ineffective.
4. All frequency bands cannot be disabled at the same time.

2.3.37. PQTMCFGSAT

Sets/gets GNSS satellite mask.

Type:

Set/Get

Synopsis:

```
//Set:
$PQTMCFGSAT,W,<SystemID>,<SignalID>,<MaskLow>[,MaskHigh]*<Checksum><CR><LF>
//Get:
$PQTMCFGSAT,R,<SystemID>,<SignalID>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<SystemID>	Numeric	-	GNSS system ID. 1 = GPS 2 = GLONASS 3 = Galileo 4 = BDS 5 = QZSS 6 = NavIC
<SignalID>	Hexadecimal	-	GNSS signal ID. For GPS: 1 = L1 C/A 2 = L2C 3 = L5-Q For GLONASS: 1 = G1 C/A 2 = G2 C/A For Galileo: 1 = E1 2 = E5a 3 = E5b 4 = E6 For BDS: 1 = B1I 2 = B2I 3 = B3I 4 = B1C 5 = B2a 6 = B2b For QZSS: 1 = L1 C/A 2 = L2C

Field	Format	Unit	Description
			3 = L5-Q 4 = L6 For NavIC: 1 = L5
<MaskLow>	Hexadecimal	-	GNSS satellite low 32-bit mask, Bit 0 for the satellite PRN 1. 0 = Disable 1 = Enable Range: GPS: 0–0xFFFFFFFF GLONASS: 0–0x3FFF Galileo: 0–0xFFFFFFFF BDS: 0–0xFFFFFFFF QZSS: 0–0x3FF NavIC: 0–0x7FFF Default value: GPS L1 C/A: 0xFFFFFFFF GPS L2C: 0xFFC36FFD GPS L5-Q: 0xAFC227AD GLONASS G1 C/A: 0x00003FFF GLONASS G2 C/A: 0x00003FFF Galileo E1: 0x67967FDF Galileo E5a: 0x67967FDF Galileo E5b: 0x67967FDF Galileo E6: 0x67967FDF BDS B1I: 0xBFFCBFFF BDS B2I: 0x0000BFFF BDS B3I: 0xBFFCBFFF BDS B1C: 0xBFFC0000 BDS B2a: 0xBFFC0000 BDS B2b: 0xBFFC0000 QZSS L1 C/A: 0x0000004E QZSS L2C: 0x0000004E QZSS L5-Q: 0x0000004E QZSS L6: 0x0000004E NavIC L5: 0x000001FF
<MaskHigh>	Hexadecimal	-	GNSS satellite high 32-bit mask, Bit 0 for the satellite PRN 33. (Only available for BDS and Galileo. It should be omitted for other GNSS systems). 0 = Disable 1 = Enable Range:

Field	Format	Unit	Description
			GPS: None GLONASS: None Galileo: 0–0x0F BDS: 0–0xFFFFFFFF QZSS: None NavIC: None Default value: Galileo E1: 0x0000000B Galileo E5a: 0x0000000B Galileo E5b: 0x0000000B Galileo E6: 0x0000000B BDS B1I: 0x1C003FFF BDS B2I: 0x00000000 BDS B3I: 0x1C003FFF BDS B1C: 0x00003FFF BDS B2a: 0x00003FFF BDS B2b: 0x1C003FFF

Result:

- If successful, the module returns:

```
//Response to Set command:
$PQTMCFGSAT,OK*<Checksum><CR><LF>
//Response to Get command:
$PQTMCFGSAT,OK,<SystemID>,<SignalID>,<MaskLow>[,<MaskHigh>]*<Checksum><CR><LF>
```

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 5: Error Codes*.

Example:

```
//Set GPS L1 C/A:
$PQTMCFGSAT,W,1,1,FFFFFFFF*4B
$PQTMCFGSAT,OK*34

//Get GPS L1 C/A:
$PQTMCFGSAT,R,1,1*62
$PQTMCFGSAT,OK,1,01,FFFFFFFF*28
```

```
//Set BDS B1I:
$PQTMCFGSAT,W,4,1,BFFCBFFF,1C003FFF*60
$PQTMCFGSAT,OK*34

//Get BDS B1I:
$PQTMCFGSAT,R,4,1*67
$PQTMCFGSAT,R,4,01,BFFCBFFF,1C003FFF*55
```

NOTE

GPS L1C and QZSS L1C frequency bands are under development. Contact Quectel Technical Support (support@quectel.com) for details.

2.3.38. PQTMCFGRSID

Sets/gets the reference station ID.

Type:

Set/Get

Synopsis:

```
//Set:
$PQTMCFGRSID,W,<ID>*<Checksum><CR><LF>
//Get:
$PQTMCFGRSID,R*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<ID>	Numeric	-	Reference station ID. Range: 0–4095. Default value: 290.

Result:

- If successful, the module returns:

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

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 5: Error Codes*.

Example:

```
//Set:
$PQTMCFGRSID,W,1024*06
$PQTMCFGRSID,OK*7E

//Get:
$PQTMCFGRSID,R*28
$PQTMCFGRSID,OK,1024*55
```

2.3.39. PQTMCFGRTCM

Sets/gets RTCM.

Type:

Set/Get

Synopsis:

```
//Set:
$PQTMCFGRTCM,W,<MSM_Type>,<MSM_Mode>,<MSM_ElevThd>,<Reserved>,<Reserved>,<EPH_Mode>,<EPH_Interval>*<Checksum><CR><LF>
//Get:
$PQTMCFGRTCM,R*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<MSM_Type>	Numeric	-	RTCM MSM type. Range: 3–7 (corresponding to RTCM MSM3–MSM7). Default value: 4.
<MSM_Mode>	Numeric	-	RTCM MSM output mode when no satellite is searched. Always 0. 0 = Not output RTCM MSM message when no satellite is searched.
<MSM_ElevThd>	Numeric	Degree	Satellite elevation threshold to report measurements by

Field	Format	Unit	Description
			RTCM MSM messages. Range: [-90.0,90.0]. Default value: -90.0 (means no limitation).
<Reserved>	Numeric	-	Reserved. Default value: 07.
<Reserved>	Numeric	-	Reserved. Default value: 06.
<EPH_Mode>	Numeric	-	Ephemeris output mode. 0 = Disable 1 = Output when updating 2 = Output when updating and at regular intervals defined by **<EPH_Interval>** 3 = Output on each epoch
<EPH_Interval>	Numeric	Second	Ephemeris output interval. Only available when **<EPH_Mode>** = 2. Range: 0–7200. Default value: 0.

Result:

- If successful, the module returns:

```
//Response to Set command:
$PQTMCFGRTCM,OK*<Checksum><CR><LF>
//Response to Get command:
$PQTMCFGRTCM,OK,<MSM_Type>,<MSM_Mode>,<MSM_ElevThd>,<Reserved>,<Reserved>,<EPH_Mode>,<EPH_Interval>*<Checksum><CR><LF>
```

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 5: Error Codes*.

Example:

```
//Set RTCM feature
$PQTMCFGRTCM,W,4,0,-90.0,07,06,1,0*3B
$PQTMCFGRTCM,OK*7A

//Get RTCM feature
$PQTMCFGRTCM,R*2C
$PQTMCFGRTCM,OK,4,0,-90.0,07,06,1,0*68
```

2.3.40. PQTMCFGSBAS

Configures SBAS.

Type:

Set/Get

Synopsis:

```
//Write:
$PQTMCFGSBAS,W,<Value>*<Checksum><CR><LF>
//Read:
$PQTMCFGSBAS,R*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Value>	Hexadecimal	-	Value of SBAS configuration. (The corresponding bit = 0 means disabled; 1 means enabled) Bit 0 = WAAS Bit 1 = SDCM Bit 2 = EGNOS Bit 3 = BDSBAS Bit 4 = MSAS Bit 5 = GAGAN Bit 6 = KASS Bit 7 = ASECNA Bit 8 = SouthPAN Default: 0x003F

Result:

- If successful, the module returns:

```
//Write:
$PQTMCFGSBAS,OK*<Checksum><CR><LF>
//Read:
$PQTMCFGSBAS,OK,<Value>*<Checksum><CR><LF>
```

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 5: Error Codes*.

Example:

```
//Set:
$PQTMCFGSBAS,W,003F*7B
$PQTMCFGSBAS,OK*71

//Get:
$PQTMCFGSBAS,R*27
$PQTMCFGSBAS,OK,003F*28
```

2.3.41. PQTMCFGNMEATID

Configures the NMEA Talker ID.

Type:

Set/Get

Synopsis:

```
//Set:
$PQTMCFGNMEATID,W,<Main_TalkerID>,<GSV_TalkerID>*<Checksum><CR><LF>

//Get:
$PQTMCFGNMEATID,R*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Main_TalkerID>	Character	-	The main Talker ID, which is used for all NMEA standard messages other than GSV. 00 = Auto mode. The main talker ID is determined by the GNSS constellation configuration. If it is not "00", use a specific two characters as talker ID.
<GSV_TalkerID>	Numeric	-	GSV Talker ID. 0 = Determined by the GNSS constellation configuration 1 = Same value as the **<Main_TalkerID>**

Result:

- If successful, the module returns:

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

```
//Get:
$PQTMCFGNMEATID,OK,<Main_TalkerID>,<GSV_TalkerID>*<Checksum><CR><LF>
```

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 5: Error Codes*.

Example:

```
//Set:
$PQTMCFGNMEATID,W,GP,0*58
$PQTMCFGNMEATID,OK*2C

//Get:
$PQTMCFGNMEATID,R*7A
$PQTMCFGNMEATID,OK,GP,0*0B

//Set:
$PQTMCFGNMEATID,W,00,0*4F
$PQTMCFGNMEATID,OK*2C
//Get:
$PQTMCFGNMEATID,R*7A
$PQTMCFGNMEATID,OK,00,0*1C
```

2.3.42. PQTMTAR

Outputs the time and attitude. The attitude computation in this message is computed from the two-antenna system.

Type:

Output

Synopsis:

```
$PQTMTAR,<MsgVer>,<Time>,<Quality>,<Res>,<Length>,<Pitch>,<Roll>,<Heading>,<Acc_Pitch>,<Acc_Roll>,<Acc_Heading>,<UsedSV>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<MsgVer>	Numeric	-	Message version. 1 = Version 1 (Always 1 for this message version.)
<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 heading 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 = Fix Heading. System used in RTK mode with fixed integers. 5 = Float Heading. Satellite system used in RTK mode, floating integers.
<Res>	-	-	Reserved. Always null.
<Length>	Numeric	Meter	Base line length.
<Pitch>	Numeric	Degree	Pitch angle. Null if invalid. Range: -90.000000 to 90.000000
<Roll>	Numeric	Degree	Roll angle. Null if invalid. Range: -180.000000 to 180.000000
<Heading>	Numeric	Degree	Heading. Null if invalid. Range: 0.000000–360.000000
<Acc_Pitch>	Numeric	Degree	Vehicle pitch accuracy. Null if invalid.
<Acc_Roll>	Numeric	Degree	Vehicle roll accuracy. Null if invalid.
<Acc_Heading>	Numeric	Degree	Vehicle heading accuracy. Null if invalid.
<UsedSV>	Numeric	-	Satellite number which used in heading solution.

Example:

```
$PQTMTAR,1,165034.000,4,,0.860,1.124780,1.254125,50.968541,0.254125,0.125485,0.012547,21*52
```

NOTE

1. Only LG580P (03) supports the sentence.
2. More information for the direction of **<Heading>** in the **$PQTMTAR** message, see *document [1] application note*.

2.3.43. PQTMCFGBLD

Configures the baseline distance between the two antennas.

Type:

Set/Get

Synopsis:

```
//Write:
$PQTMCFGBLD,W,<Distance>*<Checksum><CR><LF>
//Read:
$PQTMCFGBLD,R*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Distance>	Numeric	Meter	Baseline distance. Default 0.000. Range: [0, 5.00] (When the baseline distance is 0, the baseline distance will calculate by software.)

Result:

- If successful, the module returns:

```
//Write:
$PQTMCFGBLD,OK*38
//Read:
$PQTMCFGBLD,OK,<Distance>*<Checksum><CR><LF>
```

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 5: Error Codes*.

Example:

```
//Set:
$PQTMCFGBLD,W,1.000*68
$PQTMCFGBLD,OK*38
//Get:
$PQTMCFGBLD,R*6E
$PQTMCFGBLD,OK,1.000*3B
```

NOTE

Only LG580P (03) supports this command.

2.3.44. PQTMCFGRTKSRCTYPE

Configures RTK differential source type.

Type:

Set/Get

Synopsis:

```
//Write:
$PQTMCFGRTKSRCTYPE,W,<SrcType>*<Checksum><CR><LF>
//Read:
$PQTMCFGRTKSRCTYPE,R*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<SrcType>	Numeric	-	RTK differential source type. 0 = Auto 1 = Normal 2 = Wide Lane

Result:

- If successful, the module returns:

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

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 5: Error Codes*.

Example:

```
//Set:
$PQTMCFGRTKSRCTYPE,W,1*2B
$PQTMCFGRTKSRCTYPE,OK*65
//Get:
$PQTMCFGRTKSRCTYPE,R*33
$PQTMCFGRTKSRCTYPE,OK,1*78
```

2.3.45. PQTMSN

Reads the SN of module.

Type:

Command

Synopsis:

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

Result:

- If successful, the module returns:

```
//Response to Set command:
$PQTMSN,OK,<Reversed>,<Length>,<SN>*<Checksum><CR><LF>
```

Field	Format	Unit	Description
<Reversed>	Numeric	-	Fixed as 1.

Field	Format	Unit	Description
<Length>	Numeric	-	The length of the SN.
<SN>	String	-	The SN srting.

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 5: Error Codes*.

Example:

```
$PQTMSN*05
$PQTMSN,OK,1,16,ESG5241364AUIDE5*3C
```

2.3.46. PQTMCFGANTINF

Configures the antenna information, which will be reported in RTCM3 1033.

Type:

Set/Get

Synopsis:

```
//Set
$PQTMCFGANTINF,W,<AntDsc>,<AntSetupID>,<AntSN>*<Checksum><CR><LF>
//Get
$PQTMCFGANTINF,R*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<AntDsc>	String	-	The antenna description. This field will be reported in RTCM3 DF030. Max 31 characters. This filed is empty as default.
<AntSetupID>	Numeric	-	The antenna setup ID. This field will be reported in RTCM3 DF031. Range: [0, 255] Default: 0
<AntSN>	String	-	The antenna serial number. This field will be reported in RTCM3 DF033.

Field	Format	Unit	Description
			Max 31 characters. This filed is empty as default.

Result:

- If successful, the module returns:

```
//Response to Set command:
$PQTMCFGANTINF,OK*<Checksum><CR><LF>
//Response to Get command:
$PQTMCFGANTINF,OK,<AntDsc>,<AntSetupID>,<AntSN>*<Checksum><CR><LF>
```

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 5: Error Codes*.

Example:

```
//Write the configuration
$PQTMCFGANTINF,W,,0,*27
$PQTMCFGANTINF,OK*68

//Read the configuration
$PQTMCFGANTINF,R*3E
$PQTMCFGANTINF,OK,,0,*74
```

2.3.47. PQTMCFGANTDELTA

Configures the delta between reference point and antenna.

Type:

Set/Get

Synopsis:

```
//Set
$PQTMCFGANTDELTA,W,<East>,<North>,<Height>*<Checksum><CR><LF>
//Get
$PQTMCFGANTDELTA,R*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<East>	Numeric	Meter	The East deviation from the reference point to the antenna. Range: [-200.0000, 200.0000] Default: 0.0000
<North>	Numeric	Meter	The North deviation from the reference point to the antenna. Range: [-200.0000, 200.0000] Default: 0.0000
<Height>	Numeric	Meter	The Height deviation from the reference point to the antenna, this value will be reported in RTCM3 1006 DF028. Range: [0.0000, 6.5535] Default: 0.0000

Result:

- If successful, the module returns:

```
//Response to Set command:
$PQTMCFGANTDELTA,OK*<Checksum><CR><LF>
//Response to Get command:
$PQTMCFGANTDELTA,OK,<East>,<North>,<Height>*<Checksum><CR><LF>
```

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 5: Error Codes*.

Example:

```
//Write the configuration
$PQTMCFGANTDELTA,W,0.0000,0.0000,0.0000*10
$PQTMCFGANTDELTA,OK*71

//Read the configuration
$PQTMCFGANTDELTA,R*27
$PQTMCFGANTDELTA,OK,0.0000,0.0000,0.0000*43
```

2.3.48. PQTMCFGSIGGRP

Configures the GNSS signal group.

Type:

Set/Get

Synopsis:

```
//Set
$PQTMCFGSIGGRP,W,<SigGrpNum>*<Checksum><CR><LF>
//Get
$PQTMCFGSIGGRP,R*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<SigGrpNum>	Numeric	-	GNSS signal group number. See table below.

Table 7: <SigGrpNum> Supports Mode and Corresponding Signal Bands

SigGrpNum	Signal Band (Antenna 1)	Signal Band (Antenna 2)
0	**LG290P (03)&LG580P (03)&LG680P (03):** GPS: L1 C/A, L2C, L5-Q GLONASS: G1 C/A, G2 C/A Galileo: E1, E5a, E5b, E6 BDS: B1I, B1C, B2I, B2a, B2b, B3I QZS: L1 C/A, L2C, L5-Q, L6 IRN: L5 SBAS	**LG290P (03)&LG680P (03):** Not support. **LG580P (03):** Antenna 2 is disabled
1	**LG580P (03):** GPS: L1 C/A, L2C, L5-Q GLONASS: / Galileo: E1, E5a, E5b BDS: B1I, B1C, B2I, B2a, B2b QZS: L1 C/A, L2C, L5-Q IRN: L5 SBAS	**LG580P (03):** GPS: L1 C/A, L2C, L5-Q GLONASS: / Galileo: E1, E5a, E5b BDS: B1I, B1C, B2I, B2a, B2b QZS: L1 C/A, L2C, L5-Q IRN: L5 SBAS
2	**LG580P (03):** GPS: L1 C/A, L2C, L5-Q GLONASS: G1 C/A, G2 C/A Galileo: E1, E5a, E5b, E6	**LG580P (03):** GPS: L1 C/A GLONASS: / Galileo: E1

SigGrpNum	Signal Band (Antenna 1)	Signal Band (Antenna 2)
	BDS: B1I, B1C, B2I, B2a, B2b, B3I QZS: L1 C/A, L2C, L5-Q, L6 IRN: L5 SBAS	BDS: B1I, B1C QZS: L1 IRN: / SBAS

Table 8: Module Supports <SigGrpNum> Values and Default Values.

Module	Default	Support
LG290P (03)&LG680P (03)	0	0
LG580P (03)	1	0 (Disable antenna 2), 1, 2

Result:

- If successful, the module returns:

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

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 5: Error Codes*.

Example:

```
//Write the configuration
$PQTMCFGSIGGRP,W,1*24
$PQTMCFGSIGGRP,OK*6A

//Read the configuration
$PQTMCFGSIGGRP,R*3C
$PQTMCFGSIGGRP,OK,1*77
```

2.3.49. PQTMCFGSIGNAL2

Configures GNSS module antenna2 signal.

Type:

Set/Get

Synopsis:

```
//Write:
$PQTMCFGSIGNAL2,W,<GPS_Sig>,<GLO_Sig>,<GAL_Sig>,<BDS_Sig>,<QZS_Sig>,<NAC_Sig>*<Checksum><CR><LF>

//Read:
$PQTMCFGSIGNAL2,R*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<GPS_Sig>	Hexadecimal	-	GPS signal mask. 0 = Disable 1 = Enable Bit 0 = L1 C/A Bit 1 = L2C Bit 2 = L5-Q
<GLO_Sig>	Hexadecimal	-	GLONASS signal mask. 0 = Disable 1 = Enable Bit 0 = G1 C/A Bit 1 = G2 C/A
<GAL_Sig>	Hexadecimal	-	Galileo signal mask. 0 = Disable 1 = Enable Bit 0 = E1 Bit 1 = E5a Bit 2 = E5b Bit 3 = E6
<BDS_Sig>	Hexadecimal	-	BDS signal mask. 0 = Disable 1 = Enable Bit 0 = B1I Bit 1 = B2I Bit 2 = B3I

Field	Format	Unit	Description
			Bit 3 = B1C Bit 4 = B2a Bit 5 = B2b
<QZS_Sig>	Hexadecimal	-	QZSS signal mask. 0 = Disable 1 = Enable Bit 0 = L1C/A Bit 1 = L2C Bit 2 = L5-Q Bit 3 = L6
<NAC_Sig>	Hexadecimal	-	NavIC signal mask. 0 = Disable 1 = Enable Bit 0 = L5

Result:

- If successful, the module returns:

```
//Write:
$PQTMCFGSIGNAL2,OK*<Checksum><CR><LF>
//Read:
$PQTMCFGSIGNAL2,OK,<GPS_Sig>,<GLO_Sig>,<GAL_Sig>,<BDS_Sig>,<QZS_Sig>,<NAC_Sig>*<Checksum><CR><LF>
```

- If failed, the module returns:

```
$PQTMCFGSIGNAL2,ERROR,<ErrCode>*<Checksum><CR><LF>
```

For details about **<ErrCode>**, see *Table 5: Error Codes*.

Example:

```
// Set
$PQTMCFGSIGNAL2,W,7,3,F,3F,7,1*3C
$PQTMCFGSIGNAL2,OK*5E

//Get
$PQTMCFGSIGNAL2,R*08
$PQTMCFGSIGNAL2,OK,07,03,0F,3F,07,01*5F
```

NOTE

1. The priority of GNSS configuration: PQTMCFGCNST > PQTMCFGSIGNAL > PQTMCFGSAT
2. Cannot disable all signals, otherwise an error will be returned.
3. Only LG580P (03) supports this command.

2.3.50. PQTMCFGGEOSEP

Configures geoidal separation.

Type:

Set/Get

Synopsis:

```
//Set
$PQTMCFGGEOSEP,W,<Mode>,<GeoSep>*<Checksum><CR><LF>
//Get
$PQTMCFGGEOSEP,R*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Mode>	Numeric	-	The setting mode of geoidal separation. 0 = Auto, the module will use the built-in geoidal separation table 1 = Manual, use the value which provided in <GeoSep>
<GeoSep>	Numeric	Meter	Geoidal separation value. Range: [-1000.0000,1000.0000] Default: 0.0000

Result:

- If successful, the module returns:

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

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 5: Error Codes*.

Example:

```
//Write the configuration
$PQTMCFGGEOSEP,W,0,0.0000*04
$PQTMCFGGEOSEP,OK*79

//Read the configuration
$PQTMCFGGEOSEP,R*2F
$PQTMCFGGEOSEP,OK,0,0.0000*57
```

2.3.51. PQTMCFGCNRTHD

Configures the CNR threshold for position engine.

Type:

Set/Get

Synopsis:

```
//Set
$PQTMCFGCNRTHD,W,<CNR>*<Checksum><CR><LF>
//Get
$PQTMCFGCNRTHD,R*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<CNR>	Numeric	dBHz	The CNR threshold for position engine. Range: [0.0, 99.0] 0 = No limitation Default: 10.0

Result:

- If successful, the module returns:

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

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 5: Error Codes*.

Example:

```
//Write the configuration
$PQTMCFGCNRTHD,W,10.0*15
$PQTMCFGCNRTHD,OK*75
//Read the configuration
$PQTMCFGCNRTHD,R*23
$PQTMCFGCNRTHD,OK,10.0*46
```

2.3.52. PQTMCFGELETHD

Configures the elevation threshold for position engine.

Type:

Set/Get

Synopsis:

```
//Set
$PQTMCFGELETHD,W,<Ele>*<Checksum><CR><LF>
//Get
$PQTMCFGELETHD,R*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Ele>	Numeric	Degree	The elevation threshold for position engine. Range: [-90.0, 90.0] -90.0 = No limitation Default: 5.0

Result:

- If successful, the module returns:

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

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 5: Error Codes*.

Example:

```
//Write the configuration
$PQTMCFGELETHD,W,5.0*32
$PQTMCFGELETHD,OK*66
//Read the configuration
$PQTMCFGELETHD,R*30
$PQTMCFGELETHD,OK,5.0*61
```

2.3.53. PQTMNAV

Outputs the navigation information.

Type:

Output

Synopsis:

```
$PQTMNAV,<MsgVer>,<TimeStatus>,<TimeRef>,<UTC>,<Date>,<TOW>,<WN>,<LeapSec>,<Reserved>,<Reserved>,<SolType>,<Reserved>,<Lat>,<Lon>,<Alt>,<Sep>,<Reserved>,<Reserved>,<LatStd>,<LonStd>,<AltStd>,<Reserved>,<Reserved>,<DiffID>,<DiffAge>,<Reserved>,<SatView>,<SatUsed>,<Reserved>,<Reserved>,<Reserved>,<Reserved>,<Reserved>,<Reserved>,<HVel>,<VVel>,<HVelStd>,<VVelStd>,<Reserved>,<Reserved>,<COG>,<Reserved>,<Reserved>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<MsgVer>	Numeric	-	Message version, fixed as 1.
<TimeStatus>	Numeric	-	The time status.

Field	Format	Unit	Description
			0 = The time is invalid 1 = The time is valid
<TimeRef>	Numeric	-	The reference time system. 1 = GPS time 4 = Reserved.
<UTC>	hhmmss.sss	-	The UTC time.
<Date>	yyyymmdd	-	The UTC date.
<TOW>	Numeric	ms	Time of week. Null if invalid.
<WN>	Numeric	Week	Week number, include week rollover. Null if invalid.
<LeapSec>	Numeric	Second	Leap second. Null if invalid.
<Reserved>	-	-	Reserved. Always null.
<Reserved>	-	-	Reserved. Always null.
<SolType>	Numeric	-	Solution type. 0 = Not fixed 1 = Single 2 = SBAS 5 = Pseudorange differential 8 = RTK float 12 = RTK fixed
<Reserved>	-	-	Reserved. Always null.
<Lat>	Numeric	Degree	Latitude. Positive indicates north latitude. Null if invalid.
<Lon>	Numeric	Degree	Longitude. Positive indicates east longitude. Null if invalid.
<Alt>	Numeric	Meter	Height above mean sea level. Null if invalid.
<Sep>	Numeric	Meter	Geoid separation, difference between ellipsoid and mean sea level. Negative means mean-sea-level surface below WGS-84 ellipsoid surface. Null if invalid.
<Reserved>	-	-	Reserved. Always null.
<Reserved>	-	-	Reserved. Always null.
<LatStd>	Numeric	Meter	Latitude standard deviation. Null if invalid.

Field	Format	Unit	Description
<LonStd>	Numeric	Meter	Longitude standard deviation. Null if invalid.
<AltStd>	Numeric	Meter	Altitude standard deviation. Null if invalid.
<Reserved>	-	-	Reserved. Always null.
<Reserved>	-	-	Reserved. Always null.
<DiffID>	Numeric	-	Differential reference station ID. Range: [0,4095] Null if invalid.
<DiffAge>	Numeric	Second	Differential age. Null if invalid.
<Reserved>	-	-	Reserved. Always null.
<SatView>	Numeric	-	Number of satellites in view.
<SatUsed>	Numeric	-	Number of satellites in use.
<Reserved>	-	-	Reserved. Always null.
<Reserved>	-	-	Reserved. Always null.
<Reserved>	-	-	Reserved. Always null.
<Reserved>	-	-	Reserved. Always null.
<Reserved>	-	-	Reserved. Always null.
<Reserved>	-	-	Reserved. Always null.
<HVel>	Numeric	m/s	Horizontal velocity. Null if invalid.
<VVel>	Numeric	m/s	Vertical velocity, upward is positive. Null if invalid.
<HVelStd>	Numeric	m/s	Horizontal velocity standard deviation. Null if invalid.
<VVelStd>	Numeric	m/s	Vertical velocity standard deviation. Null if invalid.
<Reserved>	-	-	Reserved. Always null.
<Reserved>	-	-	Reserved. Always null.
<COG>	Numeric	Degree	Course over ground. Range: [0.000, 360.000)
<Reserved>	-	-	Reserved. Always null.

Field	Format	Unit	Description
<Reserved>	-	-	Reserved. Always null.

Example:

```
$PQTMNAV,1,1,1,190423.000,20241224,212681000,2346,18,,,12,,31.45874521,117.41532415,45.1254
,-6.1245,,,1.2451,2.1254,5.1242,,,290,1.0,,78,56,,,,,,,1.2101,1.2148,0.4578,1.1547,,,45.124,,*43
```

2.3.54. PQTMEOE

Outputs the end of epoch information, this message should output at the end of each epoch.

Type:

Output

Synopsis:

```
$PQTMEOE,<MsgVer>,<UTC>,<Date>,<WN>,<TOW>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<MsgVer>	Numeric	-	Message version, fixed as 1.
<UTC>	hhmmss.sss	-	The UTC time.
<Date>	yyyymmdd	-	The UTC date.
<WN>	Numeric	Week	Week number, include week rollover. Null if invalid.
<TOW>	Numeric	ms	Time of week. Null if invalid.

Example:

```
$PQTMEOE,1,190423.000,20241224,2346,212681000*45
```

2.3.55. PQTMCFGWN

Configures the reference start week number.

Type:

Set/Get

Synopsis:

```
//Write:
$PQTMCFGWN,W,<WN>*<Checksum><CR><LF>
//Read:
$PQTMCFGWN,R*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<WN>	Numeric	week	The reference GPS week number include rollover. GPS week numbers will be set correctly from <WN> up to <WN> + 1024 weeks. If the correct number of week number is obtained from the ephemeris, the ephemeris should take precedence. Range: [1, 65535] Default: 2200.

Result:

- If successful, the module returns:

```
//Write:
$PQTMCFGWN,OK*<Checksum><CR><LF>
//Read:
$PQTMCFGWN,OK,<WN>*<Checksum><CR><LF>
```

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 5: Error Codes*.

Example:

```
//Set
$PQTMCFGWN,W,2346*17
$PQTMCFGWN,OK*6B
//Get
$PQTMCFGWN,R*3D
$PQTMCFGWN,OK,2346*44
```

2.3.56. PQTMANTENNASTATUS

Reports the antenna status.

Type:

Output

Synopsis:

```
$PQTMANTENNASTATUS,<MsgVer>,<Status>,<PowerInd>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<MsgVer>	Numeric	-	Message version. Always 1.
<Status>	Numeric	-	Antenna status. 0 = Unknown. 1 = Normal. 2 = Open circuit. 3 = Short circuit.
<PowerInd>	Numeric	-	Antenna power indicator. 0 = Power off. 1 = Power on. 2 = Unknown.

Example:

```
$PQTMANTENNASTATUS,1,1,1*4E
```

NOTE

Only LG680P (03) supports this message.

2.3.57. Antenna 2 Proprietary Message

For outputting the receiver information which from Antenna2 for dual antenna module, the format follow below:

Type:

Output

Synopsis:

```
$PQTM<Name>,<MsgVer>,<NMEA_Msg>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
$	Character	-	Each NMEA message starts with **$**.
<Name>	String, characters	-	3 characters standard NMEA name.
<MsgVer>	Numeric	-	Message version. Always 2 means antenna 2 information.
<NMEA_Msg>	String	-	Standard NMEA0183 message content. The content follows standard NMEA messages. See below examples for details.
<Checksum>	Hexadecimal	-	Checksum.
<CR><LF>	Character	-	Carriage return and line feed.

Example:

```
//RMC
$PQTMRMC,2,GNRMC,024022.000,A,3149.33250501,N,11706.91294841,E,8.061,359.99,130823,,,A,V
*46
//GSV
$PQTMGSV,2,GPGSV,2,1,08,05,11,123,37,10,37,319,42,12,17,137,41,15,32,061,41,1*0C
$PQTMGSV,2,GPGSV,2,2,08,18,49,213,43,23,70,349,43,24,68,057,44,32,19,272,39,1*06
$PQTMGSV,2,GPGSV,1,1,01,18,49,213,41,6*3B
$PQTMGSV,2,GPGSV,1,1,03,10,37,319,32,24,68,057,37,32,19,272,26,8*3D
$PQTMGSV,2,GLGSV,2,1,06,74,49,140,29,73,44,061,37,88,12,325,32,87,30,285,34,1*10
$PQTMGSV,2,GLGSV,2,2,06,71,43,046,35,72,41,334,32,1*16
$PQTMGSV,2,GLGSV,1,1,02,71,43,046,35,72,41,334,32,3*10
$PQTMGSV,2,GAGSV,1,1,02,08,60,345,41,13,45,258,41,7*19
$PQTMGSV,2,GAGSV,1,1,02,08,60,345,34,13,45,258,33,1*18
$PQTMGSV,2,GBGSV,2,1,07,06,65,285,33,08,67,193,33,09,52,263,31,13,72,235,36,1*1F
$PQTMGSV,2,GBGSV,2,2,07,16,67,295,35,23,59,103,37,32,63,359,37,1*23
$PQTMGSV,2,GBGSV,1,1,03,16,67,295,35,23,59,103,37,32,63,359,37,3*25
$PQTMGSV,2,GBGSV,1,1,03,16,67,295,35,23,59,103,37,32,63,359,37,5*23
```

```
$PQTMGSV,2,GBGSV,1,1,03,16,67,295,35,23,59,103,37,32,63,359,37,8*2E
$PQTMGSV,2,GBGSV,1,1,03,16,67,295,35,23,59,103,37,32,63,359,37,B*54
$PQTMGSV,2,GQGSV,1,1,02,01,67,066,41,02,38,131,40,1*04
$PQTMGSV,2,GQGSV,1,1,02,01,67,066,41,02,38,131,40,6*03
$PQTMGSV,2,GQGSV,1,1,02,01,67,066,34,02,38,131,30,8*08
$PQTMGSV,2,GIGSV,1,1,01,01,67,066,40,1*20
//GSA
$PQTMGSA,2,GNGSA,A,3,02,05,06,09,12,17,19,20,25,,,,1.65,0.82,1.43,1*70
$PQTMGSA,2,GNGSA,A,3,85,71,,,,,,,,,,1.65,0.82,1.43,2*78
$PQTMGSA,2,GNGSA,A,3,07,13,21,26,,,,,,,,1.65,0.82,1.43,3*70
$PQTMGSA,2,GNGSA,A,3,07,08,10,12,13,19,22,35,,,,,1.65,0.82,1.43,4*74
$PQTMGSA,2,GNGSA,A,3,03,,,,,,,,,,,,1.65,0.82,1.43,5*77
$PQTMGSA,2,GNGSA,A,3,01,,,,,,,,,,,,1.65,0.82,1.43,6*76
```

NOTE

Only LG580P (03) supports the message.

3 QGC Protocol

QGC protocol is a proprietary protocol defined by Quectel. This paper introduces the QGC protocol transmitted through the UART interface of LG290P (03). The UART interface allows direct transmission of QGC messages.

3.1. QGC Protocol Message Structure

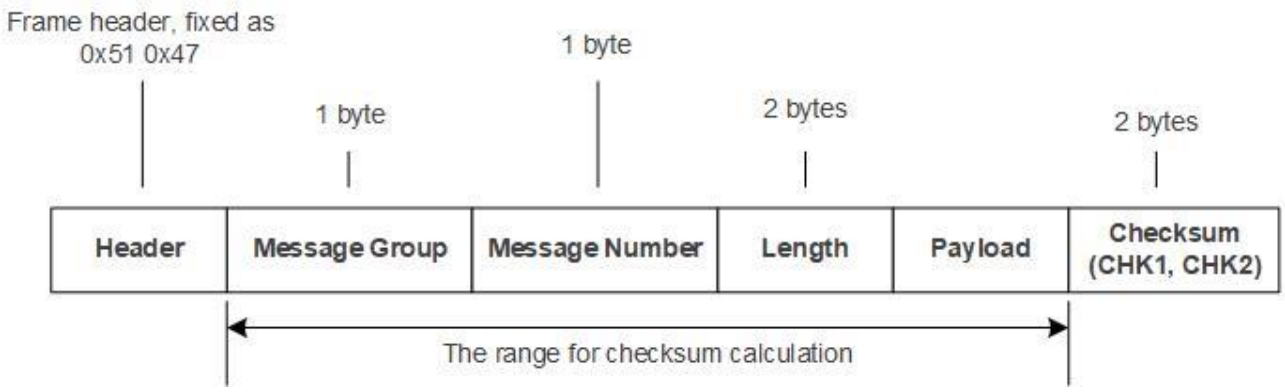

Figure 3: Structure of QGC Protocol Messages

Table 9: Structure of QGC Protocol Message

Field	Description
Header	QGC protocol frame header consisting of 2 bytes: 0x51, 0x47.
Message Group	1-byte message group. See *Table 10: Data Type of QGC Protocol* for details.
Message Number	1-byte message number. See *Table 10: Data Type of QGC Protocol* for details.
Length	Payload field length. It does not include the Header, Message Group, Message Number, Length or Checksum fields. The length format is a little-endian unsigned 16-bit integer.
Payload	Message payload, with a variable number of bytes. Data items are in little-endian format.
Checksum	2-byte checksum (CHK1 and CHK2).

QGC Checksum Sample Code:

```
// Buffer is the data array whose checksum needs to be calculated:
uint16_t Ql_Checksum(const unsigned char *Data, unsigned int Length)
{
    int16_t result = 0;
    uint8_t chk1 = 0;
    uint8_t chk2 = 0;
    unsigned int i = 0;
```

```
    if((NULL == Data) || (Length < 4))
    {
        return 0;
    }

    for(i = 0; i < Length; i++)
    {
        chk1 = chk1 + Data[i];
        chk2 = chk2 + chk1;
    }

    result = (chk1 << 8) + chk2;
}
```

Table 10: Data Type of QGC Protocol

Name	Data Type	Size (Byte)	Range
U1	unsigned char	1	[0,255]
S1	signed char	1	[-128,127]
U2	unsigned short int	2	[0,65535]
S2	signed short int	2	[-32768,32767]
U4	unsigned int	4	[0,4294967295]
S4	signed int	4	[-2147483648,2147483647]
U8	unsigned long long int	8	[0,2^64-1]
S8	signed long long int	8	[-2^63,2^63-1]
X1	8 bits field	1	Bit 7-0
X2	16 bits field	2	Bit 15-0
X4	32 bits field	4	Bit 31-0
X8	64 bits field	8	Bit 63-0
R4	single precision float	4	[-1*2^127,2^127]
R8	double precision float	8	[-1*2^1023,2^1023]

Table 11: Message Group and Message Number Overview

Message Name	Message Group	Message Number	Type	Description
GQC Message				
RAW-PPPB2B	0x0A	0xB2	Output	BDS PPPB2B binary raw messages.
RAW-QZSSL6	0x0A	0xB6	Output	QZSSL6 binary raw messages.
RAW-HASE6	0x0A	0xE6	Output	Galileo HASE6 binary raw messages.

3.2. Raw Messages

3.2.1. RAW-PPPB2B (0x0A 0xB2)

Outputs B2b PPP data.

Type:

Output

Structure:

Header	Message Group	Message Number	Length	Payload	Checksum
0x51 0x47	0x0A	0xB2	0x55 0x00	See *Table 12: RAW-PPPB2B Message Payload*	CHK1 CHK2

Table 12: RAW-PPPB2B Message Payload

Byte Offset	Data Type	Scaling	Name	Unit	Description
0	U1	-	MsgVer	-	Message version. Fixed as 1.
1	U1[4]	-	Reserved	-	Always 0.
5	U1	-	PRN	-	PRN number. Range: 1–64.
6	X1	-	Flag	-	The flag of PPP service. Bit 5: Indicate PPP service status of the current satellite. 0 = Normal 1 = Abnormal

7	U1	-	MsgType	-	Message type.
8	U1[16]	-	Reserved	-	Always 0.
24	U1[61]	-	MsgData	-	Message data, 486 bits since message type ID to message CRC, the first bit corresponds to the highest bit of MsgData[0], and only the high 6 bits of MsgData[60] are valid.

Example:

51 47 0A B2 55 00 01 00 00 00 00 3C 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 10 35 FC 49 04 40 01 3F 77 04 00 11 00 04 40 01 10 00 44 00 11 00 05 80 00 5F 6B 84 00 11 00 07 7D 63 10 00 78 17 0F FD D1 02 57 10 00 44 00 11 00 04 40 01 10 00 58 7F 00 01 81 36 B0 2B C6

NOTE

Contact Quectel Technical Support (support@quectel.com) for details.

3.2.2. RAW-QZSSL6 (0x0A 0xB6)

Outputs QZSS L6 data.

Type:

Output

Structure:

Header	Message Group	Message Number	Length	Payload	Checksum
0x51 0x47	0x0A	0xB6	0x12 0x01	See *Table 13: RAW-QZSSL6 Message Payload*	CHK1 CHK2

Table 13: RAW-QZSSL6 Message Payload

Byte Offset	Data Type	Scaling	Name	Unit	Description
0	U1	-	MsgVer	-	Message version. Fixed as 1.
1	U1[4]	-	Reserved	-	Always 0.
5	U1	-	PRN	-	PRN number. Range: 1–10.

Byte Offset	Data Type	Scaling	Name	Unit	Description
6	X1	-	Flag	-	Message flag. ● Bit 1-0: Reed-Solomon decoding status. 00 = The original data frame did not pass RS verification, and even after error correction, it did not pass RS verification. 01 = The original data frame passes RS verification. 10 = The original data frame did not pass RS verification, but after error correction, it passed RS verification. 11 = Reserved. ● Bit 7: Message type. 0 = L6E message. 1 = L6D message.
7	U1[17]	-	Reserved	-	Always 0.
24	U1[250]	-	MsgData	-	Message data, 2000bits since header to the end of RS code, the first bit corresponds to the highest bit of MsgData[0].

Example:

51 47 0A B6 12 01 01 00 00 00 00 C3 81 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 1A CF FC 1D C3 A9 7F 48 AB 08 92 88 C0 3A A8 40 30 02 02 93 DF DF F5 D5 DE 43 1C 00 00 02 04 80 04 70 00 00 00 00 82 40 7F 49 E8 11 08 04 7F ED 40 0E 9F E2 01 8B 02 B1 02 5C 00 5C 06 2F B1 9F EA BF EA BF F6 00 5D 07 7B F1 03 E8 A7 F5 10 6E DF D2 5A 04 A2 3B FD 0D FC 40 30 0E FC 5C 02 20 0B C8 BF 0B 03 98 0D 60 5F 0A 80 D4 03 98 BF 68 FF DF 03 82 87 D5 4F BA 01 8C D7 F0 88 4B FE D4 31 F4 34 08 A0 3A 19 FC E7 F0 10 01 17 7C 3A FD B8 23 04 BF B5 1F F2 FF E8 93 F7 4C 01 C0 04 13 BE D9 00 2B FE A2 77 DE D0 08 7F D8 4E FA BD FF 70 03 09 DF 89 BF FA 00 CA 5F CD 1F B5 FD CD 2F D1 B0 0D FF 8E 97 BF 17 F6 80 55 FD 29 A0 4A 20 01 FE 80 28 00 05 C0 A7 F3 00 15 9C CD E1 8B C4 EE 2D E8 5C D1 F1 28 F9 A2 81 83 DF EB 8E 1B 5F FE 18 F7 D9 21 54 33 1C 5C 10 02 5B

NOTE

Contact Quectel Technical Support (support@quectel.com) for details.

3.2.3. RAW-HASE6(0x0A 0xE6)

Outputs Galileo HAS E6 data.

Type:

Output

Structure:

Header	Message Group	Message Number	Length	Payload	Checksum
0x51 0x47	0x0A	0xE6	24+53*Page	See *Table 14: RAW-HASE6 Message Payload*.	CHK1 CHK2

Table 14: RAW-HASE6 Message Payload

Byte Offset	Data Type	Scaling	Name	Unit	Description
0	U1	-	MsgVer	-	Message version. Fixed as 1.
1	U1[3]	-	Reserved	-	Always 0.
4	U1	-	PRN	-	PRN number. Range: 1–36.
5	X1	-	Status	-	HAS Status. ● Bit 1-0: HAS working mode. 00 = Testing mode 01 = Operational mode ● Bit 7-2: Reserved.
6	U1	-	MsgType	-	Data packet type. 0x01 = MT1
7	U1	-	Reserved	-	Always 0.
8	U1	-	Page	-	The pages of message data.
9	U1[15]	-	Reserved		Always 0.
24	U1[K]	-	MsgData	-	Message data, K = 53 * Page. Include HAS message type and message body.

Example:

```
51 47 0A E6 82 00 01 00 00 00 22 01 01 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 38 B2 00 E8 50 E9 A0 5E 7F C6 0D 00 31 FF 2E 00 00 5B FE 50 2C C0 E1 00 00 2F 77 E0 0B 20 C6 E5 3F 49 79 F0 10 50 11 F8 CB EB 7F 31 04 67 D0 80 F2 05 C0 0E 81 B2 00 E8 50 E9 A0 5E 7F C6 0D 00 31 FF 2E 00 00 5B FE 50 2C C0 E1 00 00 2F 77 E0 0B 20 C6 E5 3F 49 79 F0 10 50 11 F8 CB EB 7F 31 04 67 D0 80 F2 05 C0 0E 81 C8 C1 AD
```

NOTE

1. Contact Quectel Technical Support (support@quectel.com) for details.
2. Only LG580P (03) and LG680P (03) support the sentence.

4 RTCM Protocol

The LG290P (03), LG580P (03) and LG680P (03) GNSS module supports the RTCM protocol that is in accordance with *RTCM Standard 10403.3 Differential GNSS (Global Navigation Satellite Systems) Services - Version 3*. This protocol is used for transferring GNSS raw measurement data and is available from https://www.rtcm.org/.

Table 15: Supported RTCM3 Messages

Message Type	Mode	Message Name
1005	Input/Output	Stationary RTK Reference Station ARP
1006	Input/Output	Stationary RTK Reference Station ARP with height
1019	Input/Output	GPS Ephemerides
1020	Input/Output	GLONASS Ephemerides
1033	Input/Output	Receiver and Antenna Descriptor
1041	Input/Output	NavIC/IRNSS Ephemerides
1042	Input/Output	BDS Satellite Ephemeris Data
1044	Input/Output	QZSS Ephemerides
1046	Input/Output	Galileo I/NAV Satellite Ephemeris Data
1073	Input/Output	GPS MSM3
1074	Input/Output	GPS MSM4
1075	Input/Output	GPS MSM5
1076	Input/Output	GPS MSM6
1077	Input/Output	GPS MSM7
1083	Input/Output	GLONASS MSM3
1084	Input/Output	GLONASS MSM4

Message Type	Mode	Message Name
1085	Input/Output	GLONASS MSM5
1086	Input/Output	GLONASS MSM6
1087	Input/Output	GLONASS MSM7
1093	Input/Output	Galileo MSM3
1094	Input/Output	Galileo MSM4
1095	Input/Output	Galileo MSM5
1096	Input/Output	Galileo MSM6
1097	Input/Output	Galileo MSM7
1113	Input/Output	QZSS MSM3
1114	Input/Output	QZSS MSM4
1115	Input/Output	QZSS MSM5
1116	Input/Output	QZSS MSM6
1117	Input/Output	QZSS MSM7
1123	Input/Output	BDS MSM3
1124	Input/Output	BDS MSM4
1125	Input/Output	BDS MSM5
1126	Input/Output	BDS MSM6
1127	Input/Output	BDS MSM7
1133	Input/Output	NavIC/IRNSS MSM3
1134	Input/Output	NavIC/IRNSS MSM4
1135	Input/Output	NavIC/IRNSS MSM5
1136	Input/Output	NavIC/IRNSS MSM6
1137	Input/Output	NavIC/IRNSS MSM7

5 Appendix A References

Table 16: Related Documents

Document Name
[1] Quectel_LG290P(03)&LGx80P(03)_Base_Station_Mode_Application_Note
[2] Quectel_LG580P(03)_Dual-antenna_Heading_Application_Note

Table 17: Terms and Abbreviations

Abbreviation	Description
2D	2 Dimension
3D	3 Dimension
ARP	Antenna Reference Point
ASECNA	Agency for Aviation Security and Navigation in Africa and Madagascar
BDS	Beidou Navigation Satellite System
BDSBAS	BDS Satellite-based Augmentation System
C/N_0	Carrier-to-Noise-Density Ratio
COG	Course over Ground
COGM	Course over Ground (in Magnetic North Course Direction)
COGT	Course over Ground (in True North Course Direction)
DOP	Dilution of Precision
EGNOS	European Geostationary Navigation Overlay Service
EPH	Ephemeris
GAGAN	GPS and GEO Augmented Navigation

Abbreviation	Description
GLONASS	Global Navigation Satellite System (Russia)
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
HAS	High Accuracy Service
HDOP	Horizontal Dilution of Precision
IRNSS	Indian Regional Navigation Satellite System
KASS	Korean Augmentation Satellite System
MSAS	Multi-functional Satellite Augmentation System
MSM	Multiple Signal Messages
NavIC	Navigation with Indian Constellation
NMEA	NMEA (National Marine Electronics Association) 0183 Interface Standard
NVM	Non-Volatile Memory
PDOP	Position Dilution of Precision
PPS	Pulse Per Second
PRN	Pseudo-Random Noise
QZSS	Quasi-Zenith Satellite System
RTD	Real-Time Differential
RTK	Real-Time Kinematic
SBAS	Satellite-Based Augmentation System
SDCM	System for Differential Corrections and Monitoring
SOG	Speed over Ground
SouthPAN	Southern Positioning Augmentation System
SPS	Standard Positioning Service
TXT	Text

Abbreviation	Description
UART	Universal Asynchronous Receiver/Transmitter
UTC	Coordinated Universal Time
VDOP	Vertical Dilution of Precision
VTG	Course Over Ground and Ground Speed
WAAS	Wide Area Augmentation System

6 Appendix B GNSS (NMEA) Numbering

Table 18: GNSS Satellites (NMEA) Numbering

GNSS Type	System ID	Satellite ID	Signal ID
GPS	1	1–32	1 = L1 C/A 6 = L2C 8 = L5-Q
GLONASS	2	65–96	1 = G1 C/A 3 = G2 C/A
Galileo	3	1–36	1 = E5a 2 = E5b 5 = E6 7 = E1
BDS	4	1–64	1 = B1I 3 = B1C 5 = B2a 6 = B2b 8 = B3I B = B2I
QZSS	5	1–10	1 = L1 C/A 6 = L2C 8 = L5-Q
NavIC (IRNSS)	6	1–15	1 = L5
SBAS	-	33–64	-

NOTE

The table above is only applicable to standard NMEA messages.

7 Appendix C Special Characters

Table 19: Special Characters

Special Character	Definition
<...>	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.