

LG29xP&LGx80P Series

GNSS Protocol Specification

GNSS Products

Version: 1.2

Date: 2026-03-16

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.

No. 8 Waipojing Road, Sijing Town, Songjiang District, Shanghai 201601, China

Tel: +86 21 5108 6236

Email: info@quectel.com

Or our local offices. For more information, please visit:

<https://www.quectel.com/contact/>.

For technical support, or to report documentation errors, please visit:

<https://www.quectel.com/tech-support/>.

Or email us at: support@quectel.com.

Legal Notices

We provide this document to support your product design. You are required to design your products based on the specifications and parameters set forth herein. 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. You acknowledge and agree that we may add to, amend, or restate this document at any time at our sole discretion without any prior notice to you, and such additions, amendments, or restatements shall be binding upon you.

Use and Disclosure Restrictions

License Agreements

The recipient of any hardware, software, materials, or documentation provided by us shall keep such content confidential, unless expressly authorized by us. The recipient shall not disclose, access, or use any part of the received content for any purpose other than the execution and implementation of the intended project.

Copyright

Our and third-party products hereunder may contain copyrighted materials, including but not limited to protected content, hardware, software, and documentation owned by us or applicable third parties. Unless prior written consent is obtained, you shall not access, use, or disclose any documents or information provided by us, nor shall you copy, reproduce, republish, display, translate, distribute, merge, modify, or create derivative works from any such copyrighted materials. We and the applicable third party retain exclusive rights to all copyrighted materials. No license to any patents, copyrights, trademarks, or service marks shall be granted or transferred. For the avoidance of doubt, no form of purchase shall be construed as granting any license beyond a normal, non-exclusive, royalty-free license to use the product. We reserve the right to pursue legal action against any violation of confidentiality obligations,

unauthorized use, or any other unlawful or malicious use of the aforementioned documents and information.

Trademarks

Unless otherwise expressly provided, nothing in this document shall be construed as conferring any rights to use any trademark, trade name, name, abbreviation, or counterfeit thereof owned by us or any third party in advertising, publicity, or any other contexts.

Third-Party Rights

You understand that 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 is subject to all applicable restrictions and obligations set forth herein.

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, course of performance, or usage of trade.

Privacy Policy

To enable product functionality, certain device data may be uploaded to our or third-party servers, including those operated by carriers, chipset suppliers, or servers designated by you. We strictly comply with applicable laws and regulations and will retain, use, disclose, or otherwise process relevant data solely for the purpose of enabling product functionality, or as permitted by applicable laws. Before interacting with any third party regarding data exchange, please be informed of and understand their privacy and data security policies.

Disclaimer

- a) We shall not be liable for any damages resulting from failure to comply with applicable operational or design specifications.
- b) We shall bear no liability for any inaccuracies or omissions in this document, nor for any damages arising from the use of the information contained herein.
- c) While we make every effort to ensure the integrity, accuracy, and timeliness of the features and functions under development, errors or omissions may nevertheless occur. Unless otherwise provided in a valid written agreement, we make no warranties of any kind, express, implied, or statutory, and disclaim all liability for any loss or damage arising from the use of any features or functions under development, to the maximum extent permitted by law, regardless of whether such loss or damage is foreseeable.
- d) We assume no legal responsibility for the accessibility, safety, accuracy, availability, legality, or completeness of any information, content, advertising, commercial offers, products, services, or materials on third-party websites or third-party resources.

Copyright © Quectel Wireless Solutions Co., Ltd. 2026. All rights reserved.

About the Document

Document Information	
Title	LG29xP&LGx80P Series GNSS Protocol Specification
Subtitle	GNSS Products
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	<ol style="list-style-type: none"> 1. Added applicable modules LG580P (03) and LG680P (03). 2. Added the support of M for <ModelInd> 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
		<p>\$PQTMSTAR, \$PQTMCFGBLD, \$PQTMCFGRTKSRCTYPE, \$PQTMMSN, \$PQTMCFGANTINF, \$PQTMCFGANTDELTA, \$PQTMCFGSIGGRP, \$PQTMCFG SIGNAL2, \$PQTMCFGGEOSEP, \$PQTMCFG CNRTHD, \$PQTMCFGELETHD, \$PQTMNAV, \$PQTMEOE, \$PQTMCFGWN, \$PQTMANTENNASTATUS and antenna 2 proprietary messages (Chapters 2.3.40 to 2.3.56 and 2.3.71).</p> <p>12. Added QGC protocol (Chapter 3).</p> <p>13. Added a new standard RTCM3 message (Chapter 4).</p>
1.2	2026-03-16	<p>1. Added applicable LG580P (06) and LG293P series modules.</p> <p>2. Added a table to list the applicable modules (Table 1).</p> <p>3. Updated the table of supported protocols (Table 3).</p> <p>4. Added the QGC protocol to <InputProt> and <OutputProt> and updated their default values in \$PQTMCFGPROT (Chapter 2.3.12).</p> <p>5. Updated the default value corresponding to Galileo constellation in <MaskLow> of \$PQTMCFG SAT (Chapter 2.3.37).</p> <p>6. Added \$PQTMCFGRTKRL, \$PQTMENV, \$PQTMCFGPPS2, \$PQTMCFGPPP, \$PQTMCLRMSG, \$PQTM LSTMSG, \$PQTMRTCMIS, \$PQTMCFGNAVMODE, \$PQTMCFGSTANDALONE, \$PQTMPPPNV, \$PQTMCFG PINALT \$PQTMCFGANTENNA, \$PQTMJAMMINGSTATUS, and \$PQTMCFGEVENT messages (Chapters 2.3.57 to 2.3.70).</p> <p>7. Added NAV-POS, NAV-VEL, NAV-TIME, NAV-NAV, NAV-EVENTTIME and NAV-EVENTPOS messages (Chapter 3.3).</p>

Contents

About the Document.....	3
Contents.....	5
Table Index.....	8
Figure Index.....	9
1 Introduction	10
2 NMEA Protocol.....	12
2.1. Structure of NMEA Protocol Messages.....	12
2.2. Standard Messages	14
2.2.1. RMC	14
2.2.2. GGA.....	16
2.2.3. GSV.....	18
2.2.4. GSA.....	20
2.2.5. VTG.....	21
2.2.6. GLL.....	23
2.2.7. GBS.....	24
2.2.8. GNS.....	26
2.2.9. GST.....	28
2.2.10. ZDA.....	29
2.2.11. HDT.....	30
2.2.12. THS.....	31
2.3. PQTM Messages.....	32
2.3.1. PQTMVER.....	33
2.3.2. PQTMCOLD.....	34
2.3.3. PQTMWARM.....	34
2.3.4. PQTMHOT.....	34
2.3.5. PQTMSRR.....	35
2.3.6. PQTMUNIQID.....	35
2.3.7. PQTMSAVEPAR.....	36
2.3.8. PQTMRESTOREPAR.....	37
2.3.9. PQTMVERNO.....	37
2.3.10. PQTMCFGUART.....	38
2.3.11. PQTMCFGPPS.....	40
2.3.12. PQTMCFGPROT.....	42
2.3.13. PQTMCFGNMEADP.....	43
2.3.14. PQTMEPE.....	45
2.3.15. PQTMCFGMSGRATE.....	45
2.3.16. PQTMVEL.....	51
2.3.17. PQTMCFGGEOFENCE.....	52
2.3.18. PQTMGEOFENCESTATUS.....	55
2.3.19. PQTMGNSSSTART.....	55

2.3.20.	PQTMGNSSSTOP	56
2.3.21.	PQTMTXT	57
2.3.22.	PQTMCFGSVIN	57
2.3.23.	PQTMSTATUS	60
2.3.24.	PQTMPVT	61
2.3.25.	PQTMCFGRCVRMODE	63
2.3.26.	PQTMDEBUGON	64
2.3.27.	PQTMDEBUGOFF	65
2.3.28.	PQTMCFGFIXRATE	65
2.3.29.	PQTMCFGRTK	67
2.3.30.	PQTMCFGCNST	68
2.3.31.	PQTM DOP	70
2.3.32.	PQTMPL	71
2.3.33.	PQTMCFGODO	72
2.3.34.	PQTMRESETO	73
2.3.35.	PQTMODO	74
2.3.36.	PQTMCFG SIGNAL	75
2.3.37.	PQTMCFG SAT	77
2.3.38.	PQTMCFG RSID	81
2.3.39.	PQTMCFG RTCM	82
2.3.40.	PQTMCFG SBAS	84
2.3.41.	PQTMCFG NMEATID	85
2.3.42.	PQMTAR	86
2.3.43.	PQTMCFG BLD	88
2.3.44.	PQTMCFG RTK SRCTYPE	89
2.3.45.	PQTM SN	90
2.3.46.	PQTMCFG ANT INF	91
2.3.47.	PQTMCFG ANT DELTA	92
2.3.48.	PQTMCFG SIGGRP	93
2.3.49.	PQTMCFG SIGNAL2	95
2.3.50.	PQTMCFG GEOSEP	98
2.3.51.	PQTMCFG CNR THD	99
2.3.52.	PQTMCFG ELE THD	100
2.3.53.	PQTM NAV	101
2.3.54.	PQTMEOE	104
2.3.55.	PQTMCFG WN	104
2.3.56.	PQTMANTENNA STATUS	105
2.3.57.	PQTMCFG RTK RL	106
2.3.58.	PQTMENV	107
2.3.59.	PQTMCFG PPS2	109
2.3.60.	PQTMCFG PPP	111
2.3.61.	PQTM CLRMSG	112
2.3.62.	PQTM LSTMSG	113
2.3.63.	PQTM RTCMIS	115

2.3.64.	PQTMCFGNAVMODE	117
2.3.65.	PQTMCFGSTANDALONE	118
2.3.66.	PQTMPPPNV	120
2.3.67.	PQTMCFGPINALT	123
2.3.68.	PQTMCFGANTENNA	124
2.3.69.	PQTMJAMMINGSTATUS	125
2.3.70.	PQTMCFGEVENT	126
2.3.71.	Antenna 2 Proprietary Message	127
3	QGC Protocol	130
3.1.	QGC Protocol Message Structure.....	130
3.2.	Raw Messages	134
3.2.1.	RAW-PPPB2B (0x0A 0xB2)	134
3.2.2.	RAW-QZSSL6 (0x0A 0xB6)	135
3.2.3.	RAW-HASE6 (0x0A 0xE6)	136
3.3.	Navigation Messages	138
3.3.1.	NAV-POS (0x08 0x01)	138
3.3.2.	NAV-VEL (0x08 0x11).....	139
3.3.3.	NAV-TIME (0x08 0x21)	141
3.3.4.	NAV-NAV (0x08 0x41).....	142
3.3.5.	NAV-EVENTTIME (0x08 0x51)	144
3.3.6.	NAV-EVENTPOS (0x08 0x52)	146
4	RTCM Protocol	149
5	Appendix A References	151
6	Appendix B GNSS (NMEA) Numbering	154
7	Appendix C Special Characters	155

Table Index

Table 1: Applicable Modules.....	10
Table 2: Supported Frequency Bands.....	10
Table 3: Supported Protocol.....	11
Table 4: Structure of NMEA Protocol Messages.....	12
Table 5: NMEA Talker ID	13
Table 6: Error Codes	32
Table 7: Supported Messages.....	49
Table 8: <SigGrpNum> Supports Mode and Corresponding Signal Bands	94
Table 9: Module Supports <SigGrpNum> Values and Default Values.....	95
Table 10: Structure of QGC Protocol Message.....	130
Table 11: Data Type of QGC Protocol.....	131
Table 12: Message Group and Message Number Overview	132
Table 13: Solution Status	132
Table 14: Position/Velocity Type.....	133
Table 15: RAW-PPPB2B Message Payload	134
Table 16: RAW-QZSSL6 Message Payload	135
Table 17: RAW-HASE6 Message Payload	137
Table 18: NAV-POS Message Payload	138
Table 19: NAV-VEL Message Payload	140
Table 20: NAV-TIME Message Payload.....	141
Table 21: NAV-NAV Message Payload	142
Table 22: NAV-EVENTTIME Message Payload.....	145
Table 23: NAV-EVENTPOS Message Payload.....	146
Table 24: Supported RTCM3 Messages	149
Table 25: Related Documents.....	151
Table 26: Terms and Abbreviations	151
Table 27: GNSS Satellites (NMEA) Numbering.....	154
Table 28: Special Characters	155

Figure Index

Figure 1: Structure of NMEA Protocol Messages	12
Figure 2: RTCM MSM Time Offset.....	47
Figure 3: Structure of QGC Protocol Messages	130

1 Introduction

Quectel LG29xP family and LGx80P family GNSS module supports 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

Module Family	Module Series	Variant
LG29xP	-	LG290P (03)
	LG293P	LG293P (00)
		LG293P (03)
LGx80P	LG580P	LG580P (03)
		LG580P (06)
	-	LG680P (03)

Table 2: Supported Frequency Bands

Module Series/Variant	Frequency Band
LG290P (03) LG580P Series 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 SBAS L1
LG293P Series	GPS: L1 C/A, L1C ¹⁾ , L2C, L5-Q GLONASS: G1 C/A Galileo: E1, E5a, E5b BDS: B1I, B1C, B2a, B2b, B2I QZSS: L1 C/A, L1C ¹⁾ , L2C, L5-Q NavIC: L5 SBAS L1

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 LG29xP family and LGx80P family modules support the following protocols:

Table 3: Supported Protocol

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

NOTE

- 1) LG29xP family and LGx80P family modules support GPS L1C and QZSS L1C frequency band which is still 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

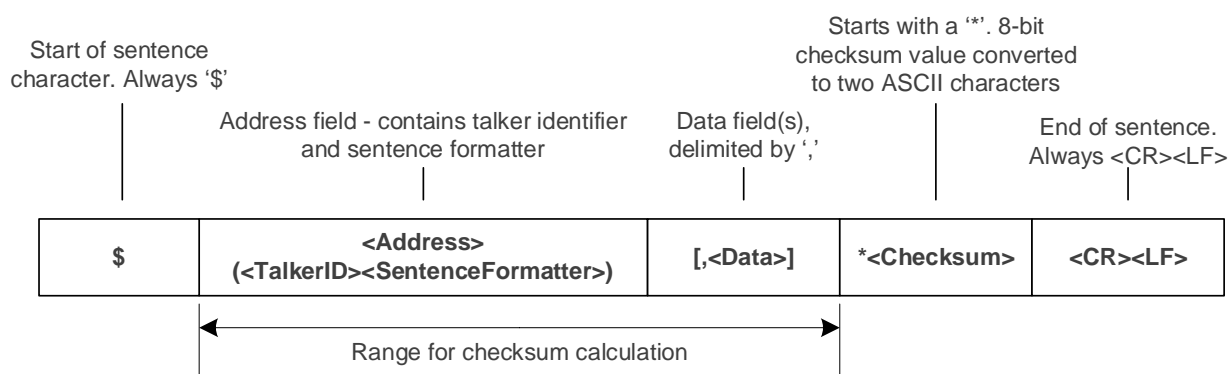


Figure 1: Structure of NMEA Protocol Messages

Table 4: Structure of NMEA Protocol Messages

Field	Description
\$	Start of the sentence (Hex 0x24).
<Address>	<p>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 5: NMEA Talker ID.</p> <p>The sentence formatter identifies the data type and the string format of the successive fields.</p> <p>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</p>

Field	Description
	proprietary sentence, and any additional characters as required.
<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 5: 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 5: 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.mmmmm mmm	-	3149.299932 10	Latitude. dd: Degrees (00–90)

Field	Format	Unit	Example	Description
				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.mmmmm mmm	-	11706.91264 104	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

Field	Format	Unit	Example	Description
				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.
<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 5: NMEA Talker ID .
GGA	String	-	GGA	Global Positioning System Fix Data.
<UTC>	hhmmss.sss	-	025159.00 0	Position fix UTC. hh: Hours (00–23)

Field	Format	Unit	Example	Description
				mm: Minutes (00–59) ss: Seconds (00–59) sss: Decimal fraction of seconds
<Lat>	ddmm.mmmmmmm m	-	3149.2999 3210	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.mmmmmmm mm	-	11706.912 64104	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 5: 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 27: 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: 000–359. 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 27: 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 5: 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 27: 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 27: 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 5: 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 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 5: NMEA Talker ID .
GLL	String	-	GLL	Geographic Position – Latitude/Longitude.
<Lat>	ddmm.mmmmm mmm	-	3149.299932 10	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.mmmmm mmmm	-	11706.91264 104	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 5: 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 invalid.
<LonExpErr>	Numeric	meter	0.5	Expected error in longitude. Null if invalid.
<AltExpErr>	Numeric	meter	1.4	Expected error in altitude. Null if invalid.
<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 27: GNSS Satellites (NMEA) Numbering .
<SignalID>	Hexadecimal	-	1	GNSS signal ID. See Table 27: 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>,<Modelnd>,<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 5: 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.mmmm mmmm	-	2516.14836 731	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.mmm mmmm	-	11020.0467 9558	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 5: 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 \$.
<TalkerID>	String	-	GN	Talker identifier. See Table 5: 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 \$.
<TalkerID>	String	-	GN	Talker identifier. See Table 5: 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 series 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 \$.

Field	Format	Unit	Example	Description
<TalkerID>	String	-	GN	Talker identifier. See Table 5: NMEA Talker ID.
THS	String	-	THS	True Heading and Status.
<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

This message only applies to LG580P series.

2.3. PQTM Messages

This chapter explains the PQTM messages (proprietary NMEA messages defined by Quectel) supported by LG29xP family and LGx80P family modules.

Table 6: 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 6: 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 6: 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 6: 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 6: 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 <u>460800</u> 921600
<DataBit>	Numeric	bit	UART data bit. <u>8</u> = 8 bits
<Parity>	Numeric	-	Parity. <u>0</u> = No parity 1 = Odd parity 2 = Even parity 3 = Mark 4 = Space
<StopBit>	Numeric	-	Stop bit(s). <u>1</u> = 1 stop bit 2 = 2 stop bits
<FlowCtrl>	Numeric	-	Flow control. <u>0</u> = 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 6: 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 6: 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

Field	Format	Unit	Description
<InputProt>	Hexadecimal	-	Input protocol.(32 bit) Bit 0 = NMEA Bit 1 = QGC Bit 2 = RTCM3 Default value: 00000007
<OutputProt>	Hexadecimal	-	Output protocol. (32 bit) Bit 0 = NMEA Bit 1 = QGC Bit 2 = RTCM3 Default value: 00000007

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 6: Error Codes](#).

Example:

```
//Set:
$PQTMCFGPROT,W,1,1,00000007,00000007*38
$PQTMCFGPROT,OK*6B
//Get:
$PQTMCFGPROT,R,1,1*3D
$PQTMCFGPROT,OK,1,1,00000007,00000007*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 6: 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>
//Get the message rate configuration on current interface:
$PQTMCFGMSGRATE,R,<MsgName>[,<MsgVer/Offset>]*<Checksum><CR><LF>
//Get 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 7: 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 7: Supported Messages .
<MsgVer/Offset>	Numeric	-	<ul style="list-style-type: none"> The parameter is the message version for PQTM messages. 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: <ul style="list-style-type: none"> Range of PQTM message version depends on the

Field	Format	Unit	Description
			specific message.
			<ul style="list-style-type: none"> Range of RTCM3 MSM time offset: 0 to <Rate> - 1.

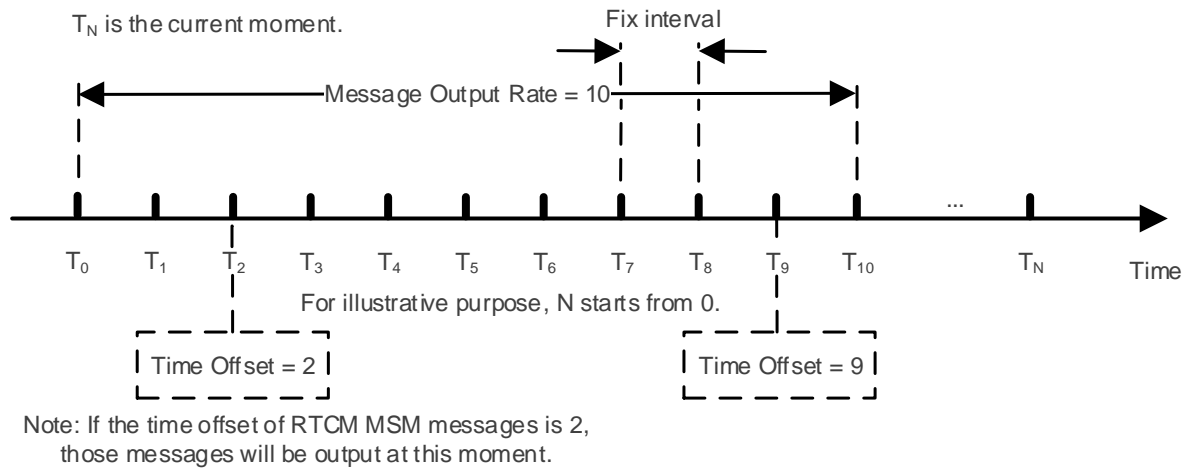


Figure 2: RTCM MSM Time Offset

Result:

- If successful, the module returns:

```
//Response to Set command:
$PQTMCFGMSGRATE,OK*<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 6: 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*4F
$PQTMCFGMSGRATE,OK,1,1,0AB2,1,1*04
    
```

Table 7: Supported Messages

Message	Description	Range/Value of N
RMC	-	1–255
GGA	-	1–255
GSV	-	1–255
GSA	-	1–255
VTG	-	1–255
GLL	-	1–255
GBS	-	1–255
GNS	-	1–255
GST	-	1–255
ZDA	-	1–255
HDT	-	1–255
THS	-	1–255
PQTMPEPE	-	1–255
PQTMVEL	-	1–255
PQTMGEOFENCESTATUS	-	1–255
PQMTTXT	-	1–255
PQTMVINSTSTATUS	-	1–255
PQTMPVT	-	1–255
PQTM DOP	-	1–255
PQTMPL	-	1–255
PQTMODO	-	1–255
PQMTAR	-	1–255
PQTMNAV	-	1–255
PQTMEOE	-	1–255
PQTMANTENNASTATUS	-	1–255

Message	Description	Range/Value of N
PQTMENV	-	1–255
PQTMRTCMIS	-	1
PQTMPPPNV	-	1–255
PQTMJAMMINGSTATUS		1–255
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
RTCM3-1046	Galileo I/NAV-EPH	1
RTCM3-1230	GLONASS Bias Information	1
RAW-PPPB2B (0AB2)	-	1
RAW-QZSSL6 (0AB6)	-	1
RAW-HASE6 (0AE6)	-	1
NAV-POS	-	1–255
NAV-VEL	-	1–255
NAV-TIME	-	1–255

Message	Description	Range/Value of N
NAV-NAV	-	1–255
NAV-EVENTTIME	-	1
NAV-EVENTPOS	-	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.
8. The **PQTMRTCMIS** message requires the input RTCM data. Thus, its output frequency directly reflects the input frequency, and **PQTMRTCMIS** message will not output if there is no input data.

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 6: 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 6: 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 6: 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,1,01,01,01,01,0x105f0cf810417c00*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 6: 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.
<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.

Field	Format	Unit	Description
<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:

```
$PQTMSTATUS,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 PQTMCFGRCVRLMODE](#).

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
<Time>	hhmmss.sss	-	UTC time. hh: Hours (0–23) mm: Minutes (0–59) ss: Seconds (0–59) sss: Decimal fraction of seconds

Field	Format	Unit	Description
<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,,3,09,18,31.12738291,117.26372910,34.212,5.267,3.21
2,2.928,0.238,4.346,34.12,2.16,4.38*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 6: 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 6: 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 6: 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 6: 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 PQTMCFGRCVRODE](#).

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 6: 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
```

NOTE

The LG293P (00) does not support this command.

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 <u>1</u> = Enable
<GLONASS>	Numeric	-	Enable/disable GLONASS. 0 = Disable <u>1</u> = Enable
<Galileo>	Numeric	-	Enable/disable Galileo. 0 = Disable <u>1</u> = Enable
<BDS>	Numeric	-	Enable/disable BDS. 0 = Disable <u>1</u> = Enable
<QZSS>	Numeric	-	Enable/disable QZSS. 0 = Disable <u>1</u> = Enable
<NavIC>	Numeric	-	Enable/disable NavIC. 0 = Disable. <u>1</u> = 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 6: 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>*<C
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:
$PQTMDO,1,570643000,1.01,0.88,0.49,0.73,0.50,0.36,0.35*7C
//No fix:
$PQTMDO,1,,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.
<PL_VelD>	Numeric	mm/s	Protection level of down velocity. Null if invalid.

Field	Format	Unit	Description
<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 6: 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 6: 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:

$$\text{<Dist>} = \text{Accumulated Distance} + \text{<InitDist>}$$
2. Accumulated distance cannot be saved to NVM.

2.3.36. PQTMCFG SIGNAL

Sets/gets GNSS signal mask.

Type:

Set/Get

Synopsis:

```
//Set:
$PQTMCFG SIGNAL,W,<GPS_Sig>,<GLO_Sig>,<GAL_Sig>,<BDS_Sig>,<QZS_Sig>,<NAC_Sig>*<Checksum><CR><LF>
//Get:
$PQTMCFG SIGNAL,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 Bit 1 = G2 C/A For LG290P (03) and LG680P (03), default value: 0x03

Field	Format	Unit	Description
			For LG580P series, default value: 0x00 For LG293P series, default value: 0x01
<GAL_Sig>	Hexadecimal	-	Galileo signal mask. 0 = Disable 1 = Enable Bit 0 = E1 Bit 1 = E5a Bit 2 = E5b Bit 3 = E6 For LG290P (03) and LG680P (03), default value: 0x0F For LG580P series and LG293P series, Default value: 0x07
<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 For LG290P (03) and LG680P (03), default value: 0x3F For LG580P series and LG293P series, default value: 0x3B
<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 For LG290P (03) and LG680P (03), default value: 0x0F For LG580P series and LG293P series, default value: 0x07
<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:
$PQTMCFG SIGNAL,OK*<Checksum><CR><LF>
//Response to Get command:
$PQTMCFG SIGNAL,OK,<GPS_Sig>,<GLO_Sig>,<GAL_Sig>,<BDS_Sig>,<QZS_Sig>,<NAC_Sig>*<Checksum><CR><LF>
```

- If failed, the module returns:

```
$PQTMCFG SIGNAL,ERROR,<ErrCode>*<Checksum><CR><LF>
```

For details about <ErrCode>, see [Table 6: Error Codes](#).

Example:

```
//Set GNSS signal mask:
$PQTMCFG SIGNAL,W,7,3,F,3F,7,1*0E
$PQTMCFG SIGNAL,OK*6C
//Get GNSS signal mask:
$PQTMCFG SIGNAL,R*3A
$PQTMCFG SIGNAL,OK,07,03,0F,3F,07,01*6D
```

NOTE

1. The L1 frequency bands of LG290P (03), LG580P series and LG680P (03) module cannot be disabled.
2. LG29xP family and LGx80P family modules also supports GPS L1C and QZSS L1C frequency band which is still under development. Contact Quectel Technical Support (support@quectel.com) for details.
3. The priority of GNSS configuration commands: **PQTMCFG CNST** > **PQTMCFG SIGNAL** > **PQTMCFG SAT**. For instance, if the GPS constellation is disabled by **PQTMCFG CNST** command, the enabling of GPS constellation and the frequency bands in **PQTMCFG SIGNAL** and **PQTMCFG SAT** commands will be ineffective.
4. All frequency bands cannot be disabled at the same time.

2.3.37. PQTMCFG SAT

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: 0–0x3FFF 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: 0x77D4DFFE Galileo E5a: 0x77D4DFFE Galileo E5b: 0x77D4DFFE Galileo E6: 0x77D4DFFE 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

Field	Format	Unit	Description
			Range: 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 6: 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 B1l:
$PQTMCFGSAT,W,4,1,BFFCBFFF,1C003FFF*60
$PQTMCFGSAT,OK*34
//Get BDS B1l:
$PQTMCFGSAT,R,4,1*67
$PQTMCFGSAT,R,4,01,BFFCBFFF,1C003FFF*55
```

NOTE

LG29xP family and LGx80P family modules support GPS L1C and QZSS L1C frequency band which is still 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.

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 6: 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 RTCM MSM messages. Range: [-90.0,90.0]. Default value: -90.0 (means no

Field	Format	Unit	Description
			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 6: 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

Sets/gets SBAS.

Type:

Set/Get

Synopsis:

```
//Set:
$PQTMCFGSBAS,W,<Value>* <Checksum><CR><LF>
//Get:
$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:

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

- If failed, the module returns:

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

For details about <ErrCode>, see [Table 6: Error Codes](#).

Example:

```
//Set:
$PQTMCFGSBAS,W,003F*7B
$PQTMCFGSBAS,OK*71
//Get:
$PQTMCFGSBAS,R*27
$PQTMCFGSBAS,OK,003F*28
```

2.3.41. PQTMCFGNMEATID

Sets/gets 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. <u>00</u> = 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. <u>0</u> = Determined by the GNSS constellation configuration <u>1</u> = Same value as the <Main_TalkerID>

Result:

- If successful, the module returns:

```
//Response to Set command:
$PQTMCFGNMEATID,OK*<Checksum><CR><LF>
//Response to Get command:
$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 6: 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)

Field	Format	Unit	Description
			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:

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

NOTE

1. Only LG580P series supports the message.
2. More information for the direction of **<Heading>** in the **PQMTAR** message, see [document \[2\] application note](#).

2.3.43. PQTMCFGBLD

Ses/gets the baseline distance between the two antennas.

Type:

Set/Get

Synopsis:

```
//Set:
$PQTMCFGBLD,W,<Distance>*<Checksum><CR><LF>
//Get:
$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:

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

- If failed, the module returns:

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

For details about <ErrCode>, see [Table 6: Error Codes](#).

Example:

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

\$PQTMCFGBLD,OK,1.000*3B

NOTE

This command only applies to LG580P series.

2.3.44. PQTMCFGRTKSRCTYPE

Sets/gets RTK differential source type.

Type:

Set/Get

Synopsis:

```
//Set:
$PQTMCFGRTKSRCTYPE,W,<SrcType>*<Checksum><CR><LF>
//Get:
$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 6: Error Codes](#).

Example:

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

NOTE

The LG293P (00) does not support this command.

2.3.45. PQTMSN

Queries 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 at 1.
<Length>	Numeric	-	The length of the SN.
<SN>	String	-	The SN string.

- If failed, the module returns:

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

For details about <ErrCode>, see [Table 6: Error Codes](#).

Example:

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

2.3.46. PQTMCFGANTINF

Sets/gets 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	-	Antenna description. This field will be reported in RTCM3 DF030. Max 31 characters. This field is empty as default.
<AntSetupID>	Numeric	-	Antenna setup ID. This field will be reported in RTCM3 DF031. Range: [0, 255] Default: 0
<AntSN>	String	-	Antenna serial number. This field will be reported in RTCM3 DF033. Max 31 characters. This field 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 6: Error Codes](#).

Example:

```
//Set:
$PQTMCFGANTINF,W,,0,*27
$PQTMCFGANTINF,OK*68
//Get:
$PQTMCFGANTINF,R*3E
$PQTMCFGANTINF,OK,,0,*74
```

2.3.47. PQTMCFGANTDELTA

Sets/gets 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	East deviation from the reference point to the antenna. Range: [-200.0000, 200.0000] Default: 0.0000
<North>	Numeric	Meter	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 6: Error Codes](#).

Example:

```
//Set:
$PQTMCFGANTDELTA,W,0.0000,0.0000,0.0000*10
$PQTMCFGANTDELTA,OK*71
//Get:
$PQTMCFGANTDELTA,R*27
$PQTMCFGANTDELTA,OK,0.0000,0.0000,0.0000*43
```

2.3.48. PQTMCFGSIGGRP

Sets/gets 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 8: <SigGrpNum> Supports Mode and Corresponding Signal Bands

SigGrpNum	Signal Band (Antenna 1)	Signal Band (Antenna 2)
0	<p>LG290P (03)&LGx80P family: 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</p> <p>LG293P series: GPS: L1 C/A, L2C, L5-Q GLONASS: G1 C/A Galileo: E1, E5a, E5b BDS: B1I, B1C, B2a, B2b, B2I QZSS: L1 C/A, L2C, L5-Q NavIC: L5</p>	<p>LG29xP family and LG680P (03): Not support. LG580P series: Antenna 2 is disabled.</p>
1	<p>LG580P series: 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</p>	<p>LG580P series: 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</p>
2	<p>LG580P series: 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</p>	<p>LG580P series: GPS: L1 C/A GLONASS: / Galileo: E1 BDS: B1I, B1C QZS: L1 IRN: / SBAS</p>

Table 9: Module Supports <SigGrpNum> Values and Default Values.

Module	Default	Support
LG290P (03)	0	0
LG293P series	0	0
LG580P series	1	0 (Disable antenna 2), 1, 2
LG680P (03)	0	0

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 6: Error Codes](#).

Example:

```
//Set:
$PQTMCFGSIGGRP,W,1*24
$PQTMCFGSIGGRP,OK*6A
//Get:
$PQTMCFGSIGGRP,R*3C
$PQTMCFGSIGGRP,OK,1*77
```

NOTE

LG29xP family and LGx80P family modules support GPS L1C and QZSS L1C frequency band which is still under development. Contact Quectel Technical Support (support@quectel.com) for details.

2.3.49. PQTMCFG SIGNAL2

Sets/gets GNSS module antenna 2 signal.

Type:

Set/Get

Synopsis:

```
//Set:
$PQTMCFG SIGNAL2,W,<GPS_Sig>,<GLO_Sig>,<GAL_Sig>,<BDS_Sig>,<QZS_Sig>,<NAC_Sig>*<Checksum><CR><LF>
//Get:
$PQTMCFG SIGNAL2,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 Bit 3 = B1C Bit 4 = B2a Bit 5 = B2b
<QZS_Sig>	Hexadecimal	-	QZSS signal mask. 0 = Disable

Field	Format	Unit	Description
			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:

```
//Response to Set command:
$PQTMCFG SIGNAL2,OK* <Checksum><CR><LF>
//Response to Get command:
$PQTMCFG SIGNAL2,OK,<GPS_Sig>,<GLO_Sig>,<GAL_Sig>,<BDS_Sig>,<QZS_Sig>,<NAC_Sig>* <Checksum><CR><LF>
```

- If failed, the module returns:

```
$PQTMCFG SIGNAL2,ERROR,<ErrCode>* <Checksum><CR><LF>
```

For details about <ErrCode>, see [Table 6: Error Codes](#).

Example:

```
// Set
$PQTMCFG SIGNAL2,W,7,3,F,3F,7,1*3C
$PQTMCFG SIGNAL2,OK*5E
//Get
$PQTMCFG SIGNAL2,R*08
$PQTMCFG SIGNAL2,OK,07,03,0F,3F,07,01*5F
```

NOTE

1. The priority of GNSS configuration: **PQTMCFG CNST > PQTMCFG SIGNAL > PQTMCFG SAT**
2. Cannot disable all signals, otherwise an error will be returned.
3. Only LG580P series supports this command.

2.3.50. PQTMCFGGEOSEP

Sets/gets 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 6: Error Codes](#).

Example:

```
//Set:
$PQTMCFGGEOSEP,W,0,0.0000*04
$PQTMCFGGEOSEP,OK*79
//Get:
```

```
$PQTMCFGGESEP,R*2F
$PQTMCFGGESEP,OK,0,0.0000*57
```

2.3.51. PQTMCFGCNRTHD

Sets/gets 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 6: Error Codes](#).

Example:

```
//Set:
$PQTMCFGCNRTHD,W,10.0*15
$PQTMCFGCNRTHD,OK*75
//Get:
$PQTMCFGCNRTHD,R*23
$PQTMCFGCNRTHD,OK,10.0*46
```

2.3.52. PQTMCFGELETHD

Sets/gets 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 6: Error Codes](#).

Example:

```
//Set:
$PQTMCFGELETHD,W,5.0*32
$PQTMCFGELETHD,OK*66
//Get:
$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 at 1.
<TimeStatus>	Numeric	-	The time status. 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.

Field	Format	Unit	Description
			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.
<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.

Field	Format	Unit	Description
<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.
<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 at 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

Sets/gets the reference start week number.

Type:

Set/Get

Synopsis:

```
//Set:
$PQTMCFGWN,W,<WN>*<Checksum><CR><LF>
//Get:
$PQTMCFGWN,R*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<WN>	Numeric	week	<p>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.</p> <p>Range: [1, 65535]</p> <p>Default: 2200.</p>

Result:

- If successful, the module returns:

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

- If failed, the module returns:

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

For details about <ErrCode>, see [Table 6: 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. PQTMCFGRTKRL

Sets/gets the RTK reliability level.

Type:

Set/Get

Synopsis:

```
//Set:
$PQTMCFGRTKRL,W,<Reliability>,<Reserved1>,<Reserved2>,<Reserved3>*<Checksum><CR><LF>
//Get:
$PQTMCFGRTKRL,R*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Reliability>	Numeric	-	RTK reliability. 1 = Very relax 2 = Relax 3 = Medium 4 = Strict 5 = Very strict
<Reserved1–3>	Numeric	-	Reserved. Always 0.

Result:

- If successful, the module returns:

```
//Response to Set command:
$PQTMCFGRTKRL,OK*<Checksum><CR><LF>
//Response to Get command:
$PQTMCFGRTKRL,OK,<Reliability>,<Reserved1>,<Reserved2>,<Reserved3>*<Checksum><CR><LF>
```

- If failed, the module returns:

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

For details about <ErrCode>, see [Table 6: Error Codes](#).

Example:

```
//Set:
$PQTMCFGRTKRL,W,3,0,0,0*71
$PQTMCFGRTKRL,OK*21
//Get:
$PQTMCFGRTKRL,R*77
$PQTMCFGRTKRL,OK,3,0,0,0*22
```

NOTE

The LG293P (00) does not support this command.

2.3.58. PQTMENV

Outputs the environment information.

Type:

Output

Synopsis:

```
$PQTMENV,<MsgVer>,<TOW>,<WN>,<Date>,<Time>,<BaseScore>,<ConfidenceLevel>,<SatVis>,<SatSlo>,<MovedFlag>,<MovingFlag>,<PosDiff>,<SloType>,<BaseSatNum>,<PubSatNum>,<Reserved1>,<Reserved2>,<Reserved3>,<Reserved4>,<Reserved5>*<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.
<WN>	Numeric	week	Week number, include week rollover. Null if invalid.
<Date>	yyyymmdd	-	The UTC date.
<Time>	hhmmss.sss	-	The UTC time.
<BaseScore>	Numeric	-	Score the environment in which the base station is located. Applicable to base station mode. Null if invalid. [100,90): Excellent [90,85): Good [85,80): General [80,0): Not suitable for building a base station.
<ConfidenceLevel>	Numeric	-	The solution confidence level. Applicable to rover mode. Null if invalid.
<SatVis>	Numeric	-	Satellite visibility. Applicable to rover mode.
<SatSlo>	Numeric	-	Satellite utilization. Applicable to rover mode.
<MovedFlag>	Numeric	-	Moved flag. 0 = No movement 1 = Compared with before power on or during power-on, there has been movement 2 = This function is not available (due to poor environment or large position deviation, which makes it impossible to judge)
<MovingFlag>	Numeric	-	Moving flag. 0 = Static

Field	Format	Unit	Description
			1 = Moving
<PosDiff>	Numeric	-	The distance between the current position and the fixed coordinates; if it is non-base station mode, the value of this field is 0.
<SloType>	Numeric	-	Position type.
<BaseSatNum>	Numeric	-	Number of satellites received by the base station.
<PubSatNum>	Numeric	-	Number of satellites in common view of base station and rover.
<Reserved1~5>	Numeric	-	Reserved, always null.

Example:

```
$PQTMENV,1,212681000,2236,20241224,190423.000,90,34.52,64,34,0,0,0.000,50,51,27,,,,,*55
```

NOTE

The LG293P (00) does not support this message.

2.3.59. PQTMCFGPPS2

Sets/gets the PPS extend feature.

Type:

Set/Get

Synopsis:

```
//Set:
$PQTMCFGPPS2,W,<Index>,<Enable>[,<Duration>,<Mode>,<Polarity>,<Reserved1>,<Period>,<Userdelay>,<Reserved2>,<Reserved3>,<Reserved4>,<Reserved5>]*<Checksum><CR><LF>
//Get:
$PQTMCFGPPS2,R,<Index>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Index>	Numeric	-	PPS index. 1 = PPS1

Field	Format	Unit	Description
<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, <Period> value]. 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
<Reserved1>	Numeric	-	Reserved. Always 0.
<Period>	Numeric	ms	Pulse output period. Range: [10, 60000]. Default value:1000.
<Userdelay>	Numeric	ns	User delay. Range: -2147483648 to 2147483647. Default value: 0
<Reserved2>	Numeric	-	Reserved, Always 1
<Reserved3–5>	Numeric	-	Reserved, Always 0

Result:

- If successful, the module returns:

```
//Response to Set command:
$PQTMCFGPPS2,OK*<Checksum><CR><LF>
//Response to Get command:
$PQTMCFGPPS2,OK,<Index>,<Enable>[,<Duration>,<Mode>,<Polarity>,<Reserved1>,<Period>,<User
delay>,<Reserved2>,<Reserved3>,<Reserved4>,<Reserved5>]*<Checksum><CR><LF>
```

- If failed, the module returns:

```
$PQTMCFGPPS2,ERROR,<ErrCode>*<Checksum><CR><LF>
```

For details about <ErrCode>, see [Table 6: Error Codes](#).

Example:

```
//Set
$PQTMCFGPPS2,W,1,1,100,1,1,0,1000,0,1,0,0,0*71
$PQTMCFGPPS2,OK*13
//Get
```

```
$PQTMCFGPPS2,R,1*58
$PQTMCFGPPS2,OK,1,1,100,1,1,0,1000,0,1,0,0,0*22
```

2.3.60. PQTMCFGPPP

Sets/gets the PPP feature.

Type:

Set/Get

Synopsis:

```
//Set:
$PQTMCFGPPP,W,<Mode>[[[,<Datum>,<Timeout>,<HorStd>,<VerStd>]*<Checksum><CR><LF>
//Get:
$PQTMCFGPPP,R*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Mode>	Hexadecimal	-	PPP mode. 0x00 = Disable 0x01 = B2b PPP 0x02 = E6 HAS
<Datum>	Numeric	-	Reference coordinate system. 1 = WGS84 2 = PPP original 3 = CGCS2000
<Timeout>	Numeric	s	PPP timeout. Range: [90,180]. Default value: 120.
<HorStd>	Numeric	m	Horizontal convergence accuracy threshold. Range: [0,5]. Default value: 0.10.
<VerStd>	Numeric	m	Vertical convergence accuracy threshold. Range: [0,5]. Default value: 0.15.

Result:

- If successful, the module returns:

```
//Response to Set command:
$PQTMCFGPPP,OK*<Checksum><CR><LF>
//Response to Get command:
$PQTMCFGPPP,OK,<Mode>,<Datum>,<Timeout>,<HorStd>,<VerStd>*<Checksum><CR><LF>
```

- If failed, the module returns:

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

For details about **<ErrCode>**, see [Table 6: Error Codes](#).

Example:

```
//Set
$PQTMCFGPPP,W,1,2,120,0.10,0.15*68
$PQTMCFGPPP,OK*22
//Get
$PQTMCFGPPP,R*74
$PQTMCFGPPP,OK,01,2,120,0.10,0.15*0B
```

NOTE

The LG293P (00) does not support this command.

2.3.61. PQTMCLRMSG

Disables serial message output.

Type:

Command

Synopsis:

```
$PQTMCLRMSG[,<PortType>,<PortID>]*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<PortType>	Numeric	-	Port type. 1 = UART
<PortID>	Numeric	-	Port ID. 1 = UART1 2 = UART2 3 = UART3

Result:

- If successful, the module returns:

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

- If failed, the module returns:

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

For details about **<ErrCode>**, see [Table 6: Error Codes](#).

Example:

```
//Disable all messages output of the current UART interface:
```

```
$PQTMCLRMSG*1C
```

```
$PQTMCLRMSG,OK*34
```

```
//Disable all messages output of the UART1:
```

```
$PQTMCLRMSG,1,1*1C
```

```
$PQTMCLRMSG,OK*34
```

2.3.62. PQTMLSTMSG

Queries message output of serial port.

Type:

Command

Synopsis:

```
$PQTMLSTMSG[,<PortType>,<PortID>]*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<PortType>	Numeric	-	Port type. 1 = UART
<PortID>	Numeric	-	Port ID. 1 = UART1 2 = UART2 3 = UART3

Result:

- If successful, the module returns:

```
$PQTMLSTMSG,OK,<PortType>,<PortID>,<MsgName>,<Rate>[,<MsgVer/Offset>]*<Checksum>
.....
$PQTMLSTMSG,OK,<PortType>,<PortID>,<MsgName>,<Rate>[,<MsgVer/Offset>]*<Checksum>
$PQTMLSTMSG,OK,End*41
```

Parameters included in the result:

Field	Format	Unit	Description
<PortType>	Numeric	-	Port type. 1 = UART
<PortID>	Numeric	-	Port ID. 1 = UART1 2 = UART2 3 = UART3
<MsgName>	-	-	Message name.
<Rate>	Numeric	-	Message output rate of the message.
<MsgVer/Offset>	Numeric	-	<ul style="list-style-type: none"> ● The parameter is the message version for PQTM messages. ● 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.

- If failed, the module returns:

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

For details about <ErrCode>, see [Table 6: Error Codes](#).

Example:

```
//Query the message output of current UART port:
$PQTMLSTMSG*0A
$PQTMLSTMSG,OK,1,1,GGA,1*52
$PQTMLSTMSG,OK,1,1,RMC,1*4F
$PQTMLSTMSG,OK,End*41
//Query the message output of current UART port (no message output):
```

```
$PQTMSTMSG*0A
$PQTMSTMSG,OK,End*41
//Query the message output of UART1:
$PQTMSTMSG,1,1*0A
$PQTMSTMSG,OK,1,1,RTCM3-107X,1,0*77
$PQTMSTMSG,OK,End*41
```

2.3.63. PQTMRTCMIS

Outputs the RTCM input status.

Type:

Output

Synopsis:

```
$PQTMRTCMIS,<MsgVer>,<RecvUTC>,<PortType>,<PortID>,<MsgType>,<SubType>,<RefStaID>,<Flag>,<MsgLen>,<MsgNum>,<SatNum>,<SigNum>{,<SigID>,<SigSatNum>}*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<MsgVer>	Numeric	-	Message version. Always 1 for this version.
<RecvUTC>	hhmmss.sss	-	UTC time when received the RTCM message.
<PortType>	Numeric	-	Port type. 1 = UART
<PortID>	Numeric	-	Port ID. 1 = UART1 2 = UART2 3 = UART3
<MsgType>	Numeric	-	Received RTCM message type.
<SubType>	Numeric	-	Received RTCM message sub-type. Only available for proprietary RTCM messages. Null if invalid.
<RefStaID>	Numeric	-	Reference station ID. Only available if the received RTCM message contained DF003 (Reference Station ID). Null if invalid or not contain DF003.
<Flag>	Hexadecimal	-	The flag of this received RTCM message.

Field	Format	Unit	Description
			<ul style="list-style-type: none"> ● Bit 0: CRC flag. 0 = CRC pass 1 = CRC fail ● Bits 2–1: Message used flag. 0 = Unknow 1 = Not used 2 = Used
<MsgLen>	Numeric	-	Length of received RTCM message from RTCM frame preamble to CRC. Unit: Byte.
<MsgNum>	Numeric	-	Total number of RTCM message types received since power on. 32-bit unsigned value, when the maximum value is reached, recounts from 0.
<SatNum>	Numeric	-	Total satellite number of received RTCM message (Bit numbers which are set to 1 in DF394 (satellite mask)). Only available for RTCM MSM messages, otherwise it should be 0.
<SigNum>	Numeric	-	Total number of signals of received RTCM message (Bit numbers which are set to 1 in DF395 (signal mask)). Only available for RTCM MSM messages, otherwise it should be 0.
Start of repeat block. Repeat times: <SigNum>.			
<SigID>	Numeric	-	GNSS signal ID, corresponds to DF395. Range: [1,32].
<SigSatNum>	Numeric	-	GNSS satellite corresponding to <SigID>. Range: [0,64].
End of repeat block.			

Example:

\$PQTMRTCMIS,1,190423.412,1,1,1074,,290,04,82,1451,9,3,2,9,15,5,24,3*1C

NOTE

The LG293P (00) does not support this message.

2.3.64. PQTMCFGNAVMODE

Sets/gets the navigation mode.

Type:

Set/Get

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<Mode>	Numeric	-	Navigation mode. 0 = Normal mode. (Basic mode applied to most scenarios, for example, driving scenario) 1~4 = Reserved 5 = Dynamic flight mode (applied to Dynamic flight mode with equivalent dynamics range and vertical acceleration on different flight phase) 6~10 = Reserved 11 = Mower mode (applied to mower application) 12~13 = Reserved 14 = Agriculture mode (applied to agriculture application)
			Default value: LG290 (03), LG293P (03) and LG680P (03): 11 (Mower mode) LG580P series: 5 (Dynamic flight mode)

Result:

- If successful, the module returns:

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

- If failed, the module returns:

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

For details about <ErrCode>, see [Table 6: Error Codes](#).

Example:

```
//Set
$PQTMCFGNAVMODE,W,0*67
$PQTMCFGNAVMODE,OK*28
//Get
$PQTMCFGNAVMODE,R*7E
$PQTMCFGNAVMODE,OK,0*34
```

NOTE

The LG293P (00) does not support this command.

2.3.65. PQTMCFGSTANDALONE

Sets/gets the Standalone mode.

Type:

Set/Get

Synopsis:

```
//Set:
$PQTMCFGSTANDALONE,W,<Mode>[[[,<Time>,<Timeout>,<Latitude>,<Longitude>,<Altitude>]*<Checksum><CR><LF>
//Get:
$PQTMCFGSTANDALONE,R*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Mode>	Numeric	-	Enable/disable Standalone mode. 0 = Disable 1 = Enable
<Time>	Numeric	Second	Wait time for automatically entering Standalone mode. Range: [3, 100]. Default value: 100.
<Timeout>	Numeric	Second	Duration of Standalone mode. Range: [1800, 86400]. Default value: 86400.

Field	Format	Unit	Description
<Latitude>	Numeric	Degree	Initial latitude for entering Standalone mode. Range: [-90, 90]. Omitted if not set or invalid.
<Longitude>	Numeric	Degree	Initial longitude for entering Standalone mode. Range: [-180, 180]. Omitted if not set or invalid.
<Altitude>	Numeric	Meter	Initial altitude for entering Standalone mode. Omitted if not set or invalid.

Result:

- If successful, the module returns:

```
//Response to Set command:
$PQTMCFGSTANDALONE,OK*<Checksum><CR><LF>
//Response to Get command:
$PQTMCFGSTANDALONE,OK,<Mode>[,<Time>,<Timeout>,<Latitude>,<Longitude>,<Altitude>]*<Checksum><CR><LF>
```

- If failed, the module returns:

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

For details about <ErrCode>, see [Table 6: Error Codes](#).

Example:

```
//Set
$PQTMCFGSTANDALONE,W,1,100,86400,31.82200600,117.11544530,44.51*0C
$PQTMCFGSTANDALONE,OK*77
//Get
$PQTMCFGSTANDALONE,R*21
$PQTMCFGSTANDALONE,OK,1,100,86400,31.82200600,117.11544530,44.51*5F
```

NOTE

The LG293P (00) does not support this command.

2.3.66. PQTMPPPNAV

Outputs the PPP navigation information.

Type:

Output

Synopsis:

```
$PQTMPPPNAV,<MsgVer>,<TimeStatus>,<TimeRef>,<UTC>,<Date>,<TOW>,<WN>,<LeapSec>,<Datumid>,<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 at 1.
<TimeStatus>	Numeric	-	Time validity. 0 = Time is invalid 1 = Time is valid
<TimeRef>	Numeric	-	Reference time system. 1 = GPS time 4 = Reserved.
<UTC>	hhmmss.sss	-	UTC time.
<Date>	yyyymmdd	-	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.
<Datumid>	-	-	Reference coordinate system. 1 = WGS84 2 = PPP original 3 = CGCS2000
<Reserved>	-	-	Reserved. Always null.
<SolType>	Numeric	-	Solution type. 0 = Not fixed 1 = Single

Field	Format	Unit	Description
			2 = SBAS 3 = Manual or from Survey-in 4 = Reserved 5 = Pseudorange differential 6 = PPP converging 7 = PPP converged 8 = RTK float 12 = RTK fixed
<Reserved>	-	-	Reserved. Always null.
<Lat>	Numeric	Degree	Latitude. Positive value indicates north latitude. Null if invalid.
<Lon>	Numeric	Degree	Longitude. Positive value 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.
<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: [0,9999] 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.

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

Example:

```
$PQTMPPNAV,1,1,1,190423.000,20241224,212681000,2346,18,1,,7,,31.45874521,117.41532415,45.1254,-6.1245,,,1.2451,2.1254,5.1242,,,9001,1.0,,78,56,,,,,,,,,1.2101,1.2148,0.4578,1.1547,,,45.124,,*25
```

NOTE

The LG293P (00) does not support this message.

2.3.67. PQTMCFGPINALT

Sets/gets the function of module’s pin 14. For pin details, see the module’s hardware design document.

Type:

Set/Get

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<PinNum>	Numeric	-	Pin number. 14 = Pin 14
<Mode>	Numeric	-	Function mode. 1 = RTK_STAT (RTK status indication) 2 = ANT_ON (Power control for external LNA or active antenna)

Result:

- If successful, the module returns:

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

- If failed, the module returns:

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

For details about <ErrCode>, see [Table 6: Error Codes](#).

Example:

```
//Set
$PQTMCFGPINALT,W,14,1*1B
$PQTMCFGPINALT,OK*7C
//Get
$PQTMCFGPINALT,R,14*03
$PQTMCFGPINALT,OK,14,1*48
```

NOTE

Only LG290P (03) and LG293P (03) supports this command.

2.3.68. PQTMCFGANTENNA

Sets/gets GNSS antenna power supply.

Type:

Set/Get

Synopsis:

```
//Set:
$PQTMCFGANTENNA,W,<Power>,<Reserved>*<Checksum><CR><LF>
//Get:
$PQTMCFGANTENNA,R*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Power>	Numeric	-	Antenna power supply. 0 = Power off 1 = Power on For LG290P (03) and LG293P (03): Only take effect when the pin is selected in ANT_ON mode. Power off (ANT_ON is set to low level) Power on (ANT_ON is set to high level)
<Reserved>	Numeric	-	Reserved. Always 0.

Result:

- If successful, the module returns:

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

- If failed, the module returns:

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

For details about **<ErrCode>**, see [Table 6: Error Codes](#).

Example:

```
//Set
$PQTMCFGANTENNA,W,0,0*7E
$PQTMCFGANTENNA,OK*2D
//Get
$PQTMCFGANTENNA,R*7B
$PQTMCFGANTENNA,OK,0,0*2D
```

NOTE

1. After the antenna power is shut off due to short-circuit protection, the **<Power>** value returned by the Get command remains the original preset value.
2. Only LG290P (03) and LG293P (03) supports the message.

2.3.69. PQTMJAMMINGSTATUS

Reports the jamming detection status.

Type:

Output

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<MsgVer>	Numeric	-	Message version. Always 1 for this version.
<Status>	Numeric	-	Jamming detection status. 0 = Unknown 1 = No jamming, healthy status 2 = Warning status 3 = Critical status

Example:

```
$PQTMJAMMINGSTATUS,1,0*46
```

2.3.70. PQTMCFGEVENT

Sets/gets external interrupt event.

Type:

Set/Get

Synopsis:

```
//Set:
$PQTMCFGEVENT,W,<Index>,<Mode>[,<Edge>,<Guard>]*<Checksum><CR><LF>
//Get:
$PQTMCFGEVENT,R,<Index>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Index>	Numeric	-	Event index. 1 = Event 1.
<Mode>	Numeric	-	Event mode. 0 = Disable 1 = Enable Note: When set to 0, the parameters after <Mode> should be omitted.
<Edge>	Numeric	-	Event interrupt trigger edge. 1 = Rising edge 2 = Falling edge
<Guard>	Numeric	ms	Guard time between two interrupts. Range: [2, 3599999]. Default value: 4

Result:

- If successful, the module returns:

```
//Response to Set command:
$PQTMCFGEVENT,OK*<Checksum><CR><LF>
//Response to Get command:
$PQTMCFGEVENT,OK,<Index>,<Mode>[,<Edge>,<Guard>]*<Checksum><CR><LF>
```

- If failed, the module returns:

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

For details about <ErrCode>, see [Table 6: Error Codes](#).

Example:

```
//Set
$PQTMCFGEVENT,W,1,1,1,150*68
$PQTMCFGEVENT,OK*3E
//Get
$PQTMCFGEVENT,R,1*75
$PQTMCFGEVENT,OK,1,1,1,150*3B
```

NOTE

Only LG290P (03) supports the message.

2.3.71. Antenna 2 Proprietary Message

Outputs the data received from antenna 2.

Type

Output

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<Name>	String, characters	-	3 characters standard NMEA messagename.

Field	Format	Unit	Description
<MsgVer>	Numeric	-	Message version. Always 2 means antenna 2 information.
<NMEA_Msg>	String	-	Content corresponding to a specific standard NMEA message, complying with NMEA 0183 protocol specification. See below examples for details.

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 series supports the message.

3 QGC Protocol

LG29xP family and LGx80P family support QGC protocol which is a proprietary protocol defined by Quectel. This paper introduces the QGC protocol transmitted through the UART interface, using LG290P (03) as an example. The UART interface allows direct transmission of QGC messages.

3.1. QGC Protocol Message Structure

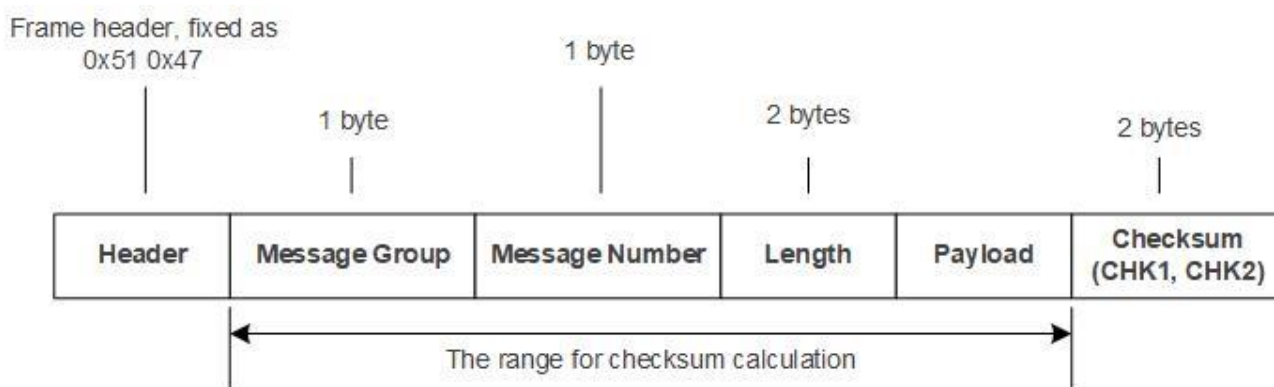


Figure 3: Structure of QGC Protocol Messages

Table 10: 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 12: Message Group and Message Number Overview for details.
Message Number	1-byte message number. See Table 12: Message Group and Message Number Overview 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.

Field	Description
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 11: 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]

Name	Data Type	Size (Byte)	Range
X1	8 bits field	1	Bit 7–Bit 0
X2	16 bits field	2	Bit 15–Bit 0
X4	32 bits field	4	Bit 31–Bit 0
X8	64 bits field	8	Bit 63–Bit 0
R4	single precision float	4	$[-1*2^{127}, 2^{127}]$
R8	double precision float	8	$[-1*2^{1023}, 2^{1023}]$

Table 12: Message Group and Message Number Overview

Message Name	Message Group	Message Number	Type	Description
RAW Message				
RAW-PPPB2B	0x0A	0xB2	Output	Outputs B2b PPP data.
RAW-QZSSL6	0x0A	0xB6	Output	Outputs B2b PPP data.
RAW-HASE6	0x0A	0xE6	Output	Outputs Galileo HAS E6 data.
Navigation Message				
NAV-POS	0x08	0x01	Output	Outputs position information.
NAV-VEL	0x08	0x11	Output	Outputs velocity information.
NAV-TIME	0x08	0x21	Output	Outputs time information.
NAV-NAV	0x08	0x41	Output	Outputs navigation information.
NAV-EVENTTIME	0x08	0x51	Output	Outputs event information.
NAV-EVENTPOS	0x08	0x52	Output	Outputs navigation information when event triggered.

Table 13: Solution Status

Value	Status	Description
0	SOL_COMPUTED	Solution computed
1	INSUFFICIENT_OBS	Insufficient observations
2	NO_CONVERGENCE	No convergence
4	COV_TRACE	Covariance trace exceeds maximum (trace > 1000 m)

Table 14: Position/Velocity Type

Value	Type	Description
0	NONE	No solution.
1	FIXEDPOS	Position has been fixed by manual command.
2	FIXEDHEIGHT	Height has been fixed by manual command.
8	DOPPLER_VELOCITY	Velocity computed using instantaneous Doppler.
16	SINGLE	Single point position.
17	PSRDIFF	Pseudorange differential solution.
18	WAAS	Solution calculated using corrections from an WAAS.
32	L1_FLOAT	Floating L1 ambiguity solution.
33	IONOFREE_FLOAT	Floating ionospheric-free ambiguity solution.
34	NARROW_FLOAT	Floating narrow-lane ambiguity solution.
48	L1_INT	Integer L1 ambiguity solution.
49	WIDE_INT	Integer wide-lane ambiguity solution
50	NARROW_INT	Integer narrow-lane ambiguity solution
52	INS	Inertial navigation system only.
53	INS_PSRSP	INS and single point position.
54	INS_PSRDIFF	INS and pseudorange differential solution.
55	INS_RTKFLOAT	INS and RTK float.
56	INS_RTKFIXED	INS and RTK fixed.

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 15: RAW-PPPB2B Message Payload	CHK1 CHK2

Table 15: RAW-PPPB2B Message Payload

Byte Offset	Data Type	Scaling	Name	Unit	Description
0	U1	-	MsgVer	-	Message version. Fixed at 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 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 B2b PPP data Outputs B2b PPP data Outputs B2b PPP data .

Type:

Output

Structure:

Header	Message Group	Message Number	Length	Payload	Checksum
0x51 0x47	0x0A	0xB6	0x12 0x01	See Table 16: RAW-QZSSL6 Message Payload	CHK1 CHK2

Table 16: RAW-QZSSL6 Message Payload

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

Byte Offset	Data Type	Scaling	Name	Unit	Description
					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 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

1. Contact Quectel Technical Support (support@quectel.com) for details.
2. The LG293P (00) does not support this message.

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 17: RAW-HASE6 Message Payload	CHK1 CHK2

3.3. Navigation Messages

3.3.1. NAV-POS (0x08 0x01)

Outputs position information.

Type:

Output

Structure:

Header	Msg Grp	Msg Num	Length	Payload	Checksum
0x51 0x47	0x08	0x01	0x54 0x00	See Table 18: NAV-POS Message Payload	CHK1 CHK2

Table 18: NAV-POS Message Payload

Byte Offset	Data Type	Scaling	Name	Unit	Description
0	U1	-	MsgVer	-	Message version. Fixed at 1.
1	U1	-	TimeStatus	-	Time status. 0x00 = Time invalid 0x01 = RTC time 0x02 = Accurate time which synchronized with satellite.
2	U1[3]	-	Reserved	-	Always 0.
5	U1	-	Hour	Hour	UTC time – hour.
6	U1	-	Min	Minute	UTC time – minute.
7	U1	-	Second	Second	UTC time – second.
8	U2	-	Millisecond	ms	UTC time – millisecond.
10	U2	-	WN	Week	Week number, include rollover.
12	U4	-	TOW	ms	Time of week.
16	U4	-	SolType	-	Solution type. See Table 13: Solution Status.

Table 19: NAV-VEL Message Payload

Byte Offset	Data Type	Scaling	Name	Unit	Description
0	U1	-	MsgVer	-	Message version. Fixed at 0x01.
1	U1	-	TimeStatus	-	Time status. 0x00 = Time invalid 0x01 = RTC time 0x02 = Accurate time which synchronized with satellite.
2	U1[3]	-	Reserved	-	Reserved.
5	U1	-	Hour	Hour	UTC time – hour.
6	U1	-	Min	Minute	UTC time – minute.
7	U1	-	Second	Second	UTC time – second.
8	U2	-	Millisecond	ms	UTC time – millisecond.
10	U2	-	WN	Week	Week number, include rollover.
12	U4	-	TOW	ms	Time of week.
16	U4	-	SolType	-	Solution type. See Table 13: Solution Status .
20	U4	-	VelType	-	Velocity type. See Table 14: Position/Velocity Type .
24	R4	-	VelN	m/s	North velocity component.
28	R4	-	VelE	m/s	East velocity component.
32	R4	-	VelU	m/s	Up velocity component. Positive values indicate upward velocity.
36	R4	-	SpdHor	m/s	Horizontal(2D) speed.
40	R4	-	Spd	m/s	Ground speed(3D).
44	R4	-	COG	Degree	Course over ground.
48	R4	-	SpdHorAcc	m/s	Horizontal(2D) speed accuracy.
52	R4	-	SpdAcc	m/s	Speed(3D) accuracy.
56	R4	-	COGAcc	Degree	COG accuracy.

Byte Offset	Data Type	Scaling	Name	Unit	Description
					0x00 = Time invalid 0x01 = RTC time 0x02 = Accurate time which synchronized with satellite.
2	U1[3]	-	Reserved	-	Reserved.
5	U1	-	Hour	Hour	UTC time – hour.
6	U1	-	Min	Minute	UTC time – minute.
7	U1	-	Second	Second	UTC time – second.
8	U2	-	Millisecond	ms	UTC time – millisecond.
10	U2	-	WN	Week	Week number, include rollover.
12	U4	-	TOW	ms	Time of week.
16	U2	-	LeapSec	Second	Leap second.
18	U1[2]	-	Reserved	-	Reserved.
20	U4	-	SolType	-	Solution type. See Table 13: Solution Status .
24	U4	-	PosType	-	Position type. See Table 14: Position/Velocity Type .
28	U1[4]		Reserved	-	Reserved.
32	R8	-	Lat	Degree	Latitude. Positive indicates north latitude.
40	R8	-	Lon	Degree	Longitude. Positive indicates east longitude.
48	R8	-	Alt	Meter	Height above mean sea level.
56	R4	-	Sep	Meter	Geoid separation, difference between ellipsoid and mean sea level. Negative means mean-sea-level surface below WGS-84 ellipsoid surface.
60	R4	-	AccLat	Meter	The accuracy of latitude.
64	R4	-	AccLon	Meter	The accuracy of longitude.
68	R4	-	AccAlt	Meter	The accuracy of altitude.
72	U4	-	RSID	-	The reference station ID.

Byte Offset	Data Type	Scaling	Name	Unit	Description
76	R4	-	DiffAge	Second	Differential age.
80	R4	-	Reserved	-	Reserved.
84	U2	-	SatView	-	Number of satellites in view.
86	U2	-	SatUsed	-	Number of satellites in used.
88	U1[14]	-	Reserved	-	Reserved.
102	R4	-	SpdHor	m/s	Horizontal(2D) speed.
106	R4	-	SpdVert	m/s	Vertical speed. Positive values indicate upward velocity.
110	R4	-	SpdHorAcc	m/s	Horizontal(2D) speed accuracy.
114	R4	-	SpdVertAcc	m/s	Vertical speed accuracy.
118	R4	-	COG	Degree	Course over ground.
122	R4	-	COG_Acc	Degree	Accuracy of COG.

Example:

```
51 47 08 41 7E 00 01 02 00 00 00 06 38 26 58 02 5A 09 18 15 A4 06 12 00 00 00 00 00 00 10 00 00
00 00 00 00 00 C2 60 B4 DE 93 D2 3F 40 D8 8A 95 19 53 47 5D 40 00 00 10 0A E2 42 65 40 00 66 9E
C0 6F 65 8D 3F 58 74 2F 3F D2 EB 05 40 00 00 00 00 00 00 00 00 00 00 0A 00 0A 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 AB 0151 47 05 3E 4F E6 91 3C 6A F8 27 3F 81 8C B8 3D C7 6F 59 43
FD FF B3 43 D9 24
```

3.3.5. NAV-EVENTTIME (0x08 0x51)

Outputs event information.

Type:

Output

Structure:

Header	Msg Grp	Msg Num	Length	Payload	Checksum
0x51 0x47	0x08	0x51	0x20 0x00	See Table 22: NAV-EVENTTIME Message Payload	CHK1 CHK2

Table 22: NAV-EVENTTIME Message Payload

Byte Offset	Data Type	Scaling	Name	Unit	Description
0	U1	-	MsgVer	-	Message version. Fixed at 0x01.
1	U1	-	TimeStatus	-	Time status. 0x00 = Time invalid 0x01 = RTC time 0x02 = Accurate time which synchronized with satellite.
2	U1[3]	-	Reserved	-	Reserved.
5	U1	-	Hour	Hour	UTC time – hour.
6	U1	-	Min	Minute	UTC time – minute.
7	U1	-	Second	Second	UTC time – second.
8	U2	-	Millisecond	ms	UTC time – millisecond.
10	U2	-	WN	Week	Week number, include rollover.
12	U4	-	TOW	ms	Time of week.
16	U2	-	LeapSec	Second	Leap second.
18	U1[2]	-	Reserved	-	Reserved.
20	U4	-	Nanosec	ns	Nanosecond when event triggered.
24	S4	-	OffsetSec	s	The offset between current epoch and the event triggered in second. The offset (in seconds) between the EVENT time and the absolute time of the nearest GGA output, calculated using the fastest output frequency of all serial GGA ports. If all serial GGA messages are disabled, the value is -1.
28	S4	-	OffsetNanosec	ns	The offset between current epoch and the event triggered in nanosecond. The offset (in nanosecond) between the EVENT time and the absolute time of the nearest GGA output, calculated using the fastest output frequency of all serial GGA ports. If all serial GGA messages are disabled, the value is -1.

Example:

51 47 08 51 20 00 01 02 00 00 00 07 03 1E 22 00 5A 09 42 5C AA 06 12 00 00 00 EB 23 0B 02 00 00 00
00 EB 23 0B 02 BF 1B

NOTE

Only LG290P (03) supports the message.

3.3.6. NAV-EVENTPOS (0x08 0x52)

Outputs navigation information when event triggered.

Type:

Output

Structure:

Header	Msg Grp	Msg Num	Length	Payload	Checksum
0x51 0x47	0x08	0x52	0x70 0x00	See Table 23: NAV-EVENTPOS Message Payload	CHK1 CHK2

Table 23: NAV-EVENTPOS Message Payload

Byte Offset	Data Type	Scaling	Name	Unit	Description
0	U1	-	MsgVer	-	Message version. Fixed at 0x01.
1	U1	-	TimeStatus	-	Time status. 0x00 = Time invalid 0x01 = RTC time 0x02 = Accurate time which synchronized with satellite.
2	U1[3]	-	Reserved	-	Reserved.
5	U1	-	Hour	Hour	UTC time – hour.

Byte Offset	Data Type	Scaling	Name	Unit	Description
6	U1	-	Min	Minute	UTC time – minute.
7	U1	-	Second	Second	UTC time – second.
8	U2	-	Millisecond	ms	UTC time – millisecond.
10	U2	-	WN	Week	Week number, include rollover.
12	U4	-	TOW	ms	Time of week.
16	U2	-	LeapSec	Second	Leap second.
18	U1[2]	-	Reserved	-	Reserved.
20	U4	-	Nanosec	ns	Nanosecond when event triggered.
24	S4	-	OffsetSec	s	<p>The offset between current epoch and the event triggered in second.</p> <p>The offset (in seconds) between the EVENT time and the absolute time of the nearest GGA output, calculated using the fastest output frequency of all serial GGA ports. If all serial GGA messages are disabled, the value is -1.</p>
28	S4	-	OffsetNanosec	ns	<p>The offset between current epoch and the event triggered in nanosecond.</p> <p>The offset (in nanosecond) between the EVENT time and the absolute time of the nearest GGA output, calculated using the fastest output frequency of all serial GGA ports. If all serial GGA messages are disabled, the value is -1.</p>
32	U4	-	SolType	-	Solution type. See Table 13: Solution Status .
36	U4	-	PosType	-	Position type. See Table 14: Position/Velocity Type .
40	R8	-	Lat	Degree	Latitude. Positive indicates north latitude.
48	R8	-	Lon	Degree	Longitude. Positive indicates east longitude.
56	R8	-	Alt	Meter	Height above mean sea level.
64	R4	-	Sep	Meter	Geoid separation, difference between ellipsoid and mean sea level. Negative

Byte Offset	Data Type	Scaling	Name	Unit	Description
					means mean-sea-level surface below WGS-84 ellipsoid surface.
68	R4	-	AccLat	Meter	The accuracy of latitude.
72	R4	-	AccLon	Meter	The accuracy of longitude.
76	R4	-	AccAlt	Meter	The accuracy of altitude.
80	U4	-	RSID	-	The reference station ID.
84	R4	-	DiffAge	Second	Differential age.
88	R4	-	Reserved	-	Reserved.
92	U2	-	SatView	-	Number of satellites in view.
94	U2	-	SatUsed	-	Number of satellites in used.
96	R4	-	VelN	m/s	North velocity component.
100	R4	-	VelE	m/s	East velocity component.
104	R4	-	VelU	m/s	Up velocity component. Positive values indicate upward velocity.
108	R4	-	COG	Degree	Course over ground.

Example:

```
51 47 08 52 70 00 01 02 00 00 00 07 03 1E 22 00 5A 09 42 5C AA 06 12 00 00 00 EB 23 0B 02 00 00 00
00 EB 23 0B 02 00 00 00 00 10 00 00 00 50 53 83 49 6A D2 3F 40 7E 85 79 96 63 47 5D 40 C3 E2 3B 26
CD 73 47 40 44 3E 9E C0 FA F6 15 40 AD E2 75 3F 96 49 52 40 00 00 00 00 00 00 00 00 00 00 00 00 26
00 26 00 DE A4 11 3A 5F 14 DD BA B5 D3 18 3A E9 49 0C 42 C6 75
```

NOTE

Only LG290P (03) supports the message.

4 RTCM Protocol

LG29xP family and LGx80P family 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.rtcn.org/>.

Table 24: 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
1230	Output	GLONASS Bias Information

5 Appendix A References

Table 25: Related Documents

Document Name
[1] Quectel_LG290P(03)&LGx80P_Series_Base_Station_Mode_Application_Note
[2] Quectel_LG580P_Series_Dual-Antenna_Heading_Application_Note

Table 26: 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
CGCS2000	China Geodetic Coordinate System 2000
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

Abbreviation	Description
GAGAN	GPS and GEO Augmented Navigation
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

Abbreviation	Description
TXT	Text
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 27: 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 28: 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.
<u>Underline</u>	Default setting of a parameter.