LG290P GNSS
Protocol Specification

GNSS Module Series

Version: 1.0.0

Date: 2024-05-17

Status: Preliminary

At Quectel, our aim is to provide timely and comprehensive services to our customers. If you require any assistance, please contact our headquarters:

Quectel Wireless Solutions Co., Ltd.
Building 5, Shanghai Business Park Phase III (Area B), No.1016 Tianlin Road, Minhang District, Shanghai 200233, China
Tel: +86 21 5108 6236
Email: info@quectel.com

Or our local offices. For more information, please visit:
http://www.quectel.com/support/sales.htm.

For technical support, or to report documentation errors, please visit:
http://www.quectel.com/support/technical.htm.
Or email us at: support@quectel.com.

Legal Notices

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

Use and Disclosure Restrictions

License Agreements

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

Copyright

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

Trademarks

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

Third-Party Rights

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

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

Privacy Policy

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

Disclaimer

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

About the Document

Document Information	
Title	**LG290P GNSS Protocol Specification**
Subtitle	GNSS Module Series
Document Type	GNSS Protocol Specification
Document Status	Preliminary

Revision History

Version	Date	Description
-	2023-12-18	Creation of the document
1.0.0	2024-05-17	Preliminary

Contents

Table Index

1 Introduction

Quectel LG290P GNSS module supports GPS, GLONASS, Galileo, BDS, QZSS and NavIC (IRNSS) constellations. GPS + GLONASS + Galileo + BDS + QZSS + NavIC are supported by default. Concurrent tracking of GPS L1 C/A、GPS L2C、GPS L5-Q、GLONASS G1 C/A、GLONASS G2 C/A、Galileo E1、Galileo E5a、Galileo E5b、Galileo E6、BDS B1I、BDS B1C、BDS B2a、BDS B2b、BDS B2I、BDS B3I、QZSS L1 C/A、QZSS L2C、QZSS L5-Q、NavIC L5 frequency bands provides fast and accurate acquisition and makes this module an ideal solution for positioning and navigation in various vertical markets.

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

The LG290P module supports the following protocols:

Table 1: Supported Protocol

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

NOTE

Quectel assumes no responsibility if commands other than the ones listed herein are used.

2 NMEA Protocol

2.1. Structure of NMEA Protocol Messages

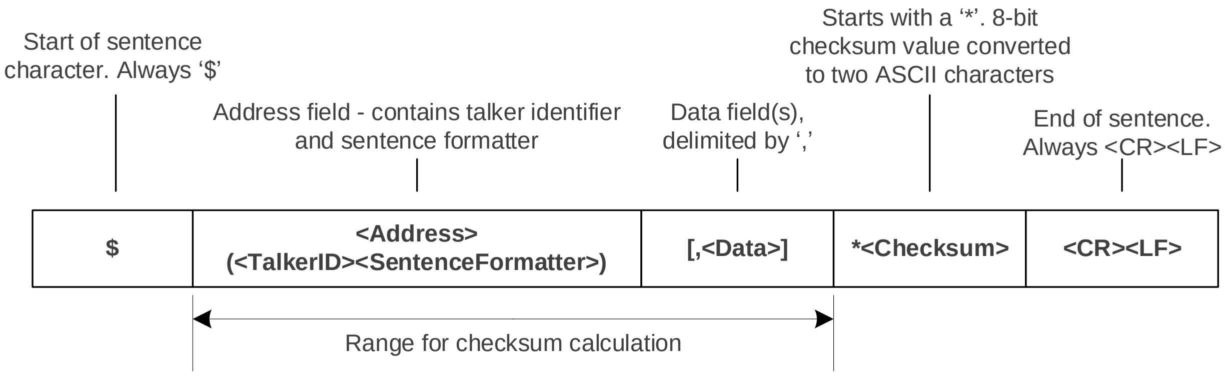
Start of sentence character. Always '$'
Address field - contains talker identifier and sentence formatter
Data field(s), delimited by ','
Starts with a '*'. 8-bit checksum value converted to two ASCII characters
End of sentence. Always <CR><LF>
$
<Address> (<TalkerID><SentenceFormatter>)
[,<Data>]
*<Checksum>
<CR><LF>
Range for checksum calculation

Figure 1: Structure of NMEA Protocol Messages

Table 2: Structure of NMEA Protocol Messages

Field	Description
$	Start of the sentence (Hex 0x24).
<Address>	**In Standard Messages:** In standard messages, this field consists of a two-character talker identifier (TalkerID) and a three-character sentence formatter (SentenceFormatter). The talker identifier identifies the type of talker. For more information on the TalkerID, see *Table 3: NMEA Talker ID*. The sentence formatter identifies the data type and the string format of the successive fields. **In Proprietary Messages:** In proprietary messages, this field consists of the proprietary character **P** followed by a three-character Manufacturer's Mnemonic Code, used to identify the TALKER issuing a proprietary sentence, and any additional characters as required.

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

Table 3: 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 of which the 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, 2 characters	-	GN	Talker identifier. See *Table 3: NMEA Talker ID.*
RMC	String, 3 characters	-	RMC	Recommended Minimum Specific GNSS Data.
<UTC>	hhmmss.sss	-	025159.000	Position fix UTC. hh: Hours (00–23) mm: Minutes (00–59) ss: Seconds (00–59) sss: Decimal fraction of seconds

<Status>	Character	-	A	Positioning system status.
				A = Data valid.
				V = Navigation receiver warning.

<Lat>	ddmm.mmmmmmmm	-	3149.29993210	Latitude. dd: Degrees (00–90) mm: Minutes (00–59) mmmmmmmm: Decimal fraction of

Field	Format	Unit	Example	Description
				minutes Note that this field is empty in case of an invalid value.

<N/S>	Character	-	N	North-south direction.
				N = North
				S = South
				Note that this field is empty in case of an invalid value.

<Lon>	dddmm.mmmmmmmm	-	11706.91264104	Longitude. ddd: Degrees (000–180) mm: Minutes (00–59) mmmmmmmm: Decimal fraction of minutes Note that this field is empty in case of an invalid value.

<E/W>	Character	-	E	East-west direction.
				E = East
				W = West
				Note that this field is empty in case of an invalid value.

<SOG>	Numeric	Knot	0.001	Speed over ground. Variable length. Note that this field is empty in case of an invalid value.
<COG>	Numeric	Degree	043.43	Course over ground. Variable length. Maximum value: 359.9. Note that this field is empty in case of an invalid value.
<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

Field	Format	Unit	Example	Description
				Satellite Based Augmentation System (SBAS). F = Float RTK. Satellite system used in RTK mode with floating integers. 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 not valid).
<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, 2 characters	-	GN	Talker identifier. See *Table 3: NMEA Talker ID.*
GGA	String, 3 characters	-	GGA	Global Positioning System Fix Data.
<UTC>	hhmmss.sss	-	025159.00	Position fix UTC.

Field	Format	Unit	Example	Description
			0	hh: Hours (00–23) mm: Minutes (00–59) ss: Seconds (00–59) sss: Decimal fraction of seconds
<Lat>	ddmm.mmmmmmmm	-	3149.29993210	Latitude. dd: Degrees (00–90) mm: Minutes (00–59) mmmmmmmm: Decimal fraction of minutes Note that this field is empty in case of an invalid value.

<N/S>	Character	-	N	North-south direction.
				N = North
				S = South
				Note that this field is empty in case of an invalid value.

<Lon>	dddmm.mmmmmmmm	-	11706.91264104	Longitude. ddd: Degrees (000–180) mm: Minutes (00–59) mmmmmmmm: Decimal fraction of minutes Note that this field is empty in case of an invalid value.

<E/W>	Character	-	E	East-west direction.
				E = East
				W = West
				Note that this field is empty in case of an invalid value.

<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.
<NumSatUsed>1)	Numeric, 2 digits	-	16	Number of satellites in use.

Field	Format	Unit	Example	Description
<HDOP>	Numeric	-	1.26	Horizontal dilution of precision.
<Alt>	Numeric	Meter	97.250	Altitude above mean-sea-level (geoid).
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>	-	Second	-	Differential GPS data age. Note that this field is empty in case of an invalid value.
<DiffStation>	-	-	-	Differential reference station ID. Note that this field is empty in case of an invalid value.
<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, 2 characters	-	GP	Talker identifier. See *Table 3: NMEA Talker ID.*
GSV	String, 3 characters	-	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 8: GNSS Satellites (NMEA) Numbering*.
<SatElev>	Numeric	Degree	77	Satellite elevation. Range: 00–90.
<SatAz>	Numeric	Degree	300	Satellite azimuth, with true north as the reference plane. Range: 000–360.
<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 8: 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** sentence 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, 2 characters	-	GN	Talker identifier. See *Table 3: NMEA Talker ID.*
GSA	String, 3 characters	-	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

Field	Format	Unit	Example	Description
				2 = 2D 3 = 3D
Start of repeat block. Repeat times: 12.				
<SatID>	Numeric	-	10	ID numbers of satellites used in solution. See *Table 8: 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 8: 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 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, 2 characters	-	GN	Talker identifier. See *Table 3: NMEA Talker ID.*
VTG	String, 3 characters	-	VTG	Course Over Ground & Ground Speed.
<COGT>	Numeric	Degrees	043.43	Course over ground, in true north direction.
T	Character	-	T	Fixed field: true.
<COGM>	Numeric	Degrees	-	Course over ground (magnetic). Not supported.
M	Character	-	M	Fixed field: magnetic.
<SOGN>	Numeric	Knots	0.001	Speed over ground in knots.
N	Character	-	N	Fixed field: knot.
<SOGK>	Numeric	km/h	0.001	Speed over ground in kilometers per hour.
K	Character	-	K	Fixed field: kilometers per hour

<ModeInd>	Character	-	A	Mode indicator.
				A = Autonomous mode. Satellite system used in non-differential mode in position fix.
				D = Differential mode. Satellite system used in differential mode in position fix. Corrections from ground stations or Satellite Based Augmentation System (SBAS).
				E = Estimated (dead reckoning) mode.
				M = Manual input mode.
				N = Data not valid.

<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, 2 characters	-	GN	Talker identifier. See *Table 3: NMEA Talker ID.*
GLL	String, 3 characters	-	GLL	Geographic Position – Latitude/Longitude.
<Lat>	ddmm.mmmmmmmm	-	3149.29993210	Latitude. dd: Degrees (00–90) mm: Minutes (00–59) mmmmmmmm: Decimal fraction of minutes Note that this field is empty in case of an invalid value.

<N/S>	Character	-	N	North-south direction.
				N = North
				S = South
				Note that this field is empty in case of an invalid value.

<Lon>	dddmm.mmmmmmmm	-	11706.91264104	Longitude. ddd: Degrees (000–180) mm: Minutes (00–59) mmmmmmmm: Decimal fraction of minutes Note that this field is empty in case of an invalid value.

Field	Format	Unit	Example	Description

<E/W>	Character	-	E	East-west direction.
				E = East
				W = West
				Note that this field is empty in case of an invalid value.

<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 = 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.
				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
```

2.3. PQTM Messages

This chapter explains the PQTM messages (proprietary NMEA messages defined by Quectel) supported by LG290P module.

Table 4: Error Codes

Feild	Format	Unit	Description
<ErrCode>	Numeric	-	Error code. 1 = Invalid parameters 2 = Execution failed. 3 = The command is not supported.

NOTE

To avoid uncertainties, you need to send **$PQTMSAVEPAR*5A** to save the configuration after setting parameters through the “Set“ command, and then restart the module to ensure that all configurations take effect. Otherwise, the module will restore default value 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 as “MODULE”.
<VerStr>	String	-	Main version string.
<BuildDate>	YYYY/MM/DD	-	Firmware build date.
<BuildTime>	hh:mm:ss	-	Firmware build time.

Example:

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

NOTE

This message will output firstly after the module is powered on.

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. The receiver will be rebooted.

Type:

Command

Synopsis:

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

Parameter:

None

Example:

```
$PQTMSRR*4B
```

2.3.6. PQTMUNIQID

Queries the unique ID of the module.

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	Chip unique ID length.
<ID>	Hexadecimal	-	Chip unique ID.

- If failed, the module returns:

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

For detail about **<ErrCode>**, see *Table 4: 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 detail about **<ErrCode>**, see *Table 4: Error Codes*.

Example:

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

2.3.8. PQTMRESTOREPAR

Restores all configurations to 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 detail about **<ErrCode>**, see *Table 4: Error Codes*.

Example:

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

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.

2.3.9. PQTMVERNO

Queries the firmware version information.

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
<BuildTime>	hh:mm:ss		Firmware build time

- If failed, the module returns:

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

For detail about **<ErrCode>**, see *Table 4: Error Codes*.

Example:

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

2.3.10. PQTMCFGUART

Configures the UART port.

Type:

Set/Get

Synopsis:

```
//Configure UART on current port:
$PQTMCFGUART,W,<BaudRate>[,<DataBit>,<Parity>,<StopBit>,<FlowCtrl>]*<Checksum><CR><LF>
//Configure the specific UART port:
$PQTMCFGUART,W,<Index>,<BaudRate>[,<DataBit>,<Parity>,<StopBit>,<FlowCtrl>]*<Checksum><CR><LF>
//Read the configuration on current port or specific port:
$PQTMCFGUART,R[,<Index>]*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Index>	Numeric	-	The UART port index. 1 = UART1 2 = UART2 3 = UART3 4 = UART4
<BaudRate>	Numeric	bps	The baud rate of UART. 9600~4000000.
<DataBit>	Numeric	bit	The data bit of UART. 8 = 8 bits
<Parity>	Numeric	-	Parity. 0 = No parity 1 = Odd parity 2 = Even parity 3 = Mark 4 = Space
<StopBit>	Numeric	-	Stop bits. 1 = 1 stop bit

Field	Format	Unit	Description
			2 = 2 stop bit
<FlowCtrl>	Numeric	-	Flow control. 0 = None

Result:

- If successful, the module returns:

```
//Write:
$PQTMCFGUART,OK*<Checksum><CR><LF>
//Read:
$PQTMCFGUART,OK,<Index>,<BaudRate>,<DataBit>,<Parity>,<StopBit>,<FlowCtrl>*<Checksum><C
R><LF>
```

- If failed, the module returns:

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

For detail about **<ErrCode>**, see *Table 4: Error Codes*.

Example:

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

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

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

//Configure all parameters on UART port 1:
$PQTMCFGUART,W,1,115200,8,0,1,0*0C
$PQTMCFGUART,OK*60
//Get the configuration of current UART port:
$PQTMCFGUART,R*36
$PQTMCFGUART,OK,1,115200,8,0,1,0*5F
```

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

2.3.11. PQTMCFGPPS

Configures the PPS feature.

Type:

Set/Get

Synopsis:

```
//Write:
$PQTMCFGPPS,W,<Index>,<Enable>,<Duration>,<Mode>,<Polarity>,<Interval>*<Checksum><CR><LF>
//Read:
$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 (default) Note: If <Enable> is set to 0, the fields after <Enable> should be omitted.
<Duration>	Numeric	Millisecond	Pulse duration. Range: 0–900 (Default: 100)
<Mode>	Numeric	-	PPS output with fix mode. 1 = Always 2 = 2D/3D
<Polarity>	Numeric	-	Pulse polarity. 0 = Low 1 = High
<Reserved>	Numeric	-	Reserved. Always 0.

Result:

- If successful, the module returns:

```
//Write:
$PQTMCFGPPS,OK*<Checksum><CR><LF>
//Read:
$PQTMCFGPPS,OK,<Index>,<Enable>,<Duration>,<Mode>,<Polarity>,<Interval>*<Checksum><CR><
LF>
```

- If failed, the module returns:

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

For detail about **<ErrCode>**, see *Table 4: 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

Configures the protocol for the input and output of the 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>	Numberic	-	Port type. 1 = UART
<PortID>	Numberic	-	Port ID. When <PortType> = 1, the <PortID> supports setting to 1–4, corresponding to UART1–4.
<InputProt>	Hexadecimal	-	Input protocol.(32 bit) Bit 0 = NMEA Bit 2 = RTCM3 When the port is UART1–4, default input protocols are NMEA and RTCM3: 00000005
<OutputProt>	Hexadecimal,	-	Output protocol. (32 bit) Bit 0 = NMEA Bit 2 = RTCM3 When the port is UART1–4, default output protocols are NMEA and RTCM3: 00000005

Result:

- If successful, the module returns:

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

- If failed, the module returns:

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

For detail about **<ErrCode>**, see *Table 4: Error Codes*.

Example:

```
//Set:
$PQTMCFGPROT,W,1,0,00000005,00000005*39
$PQTMCFGPROT,OK*6B

//Get:
$PQTMCFGPROT,R,1,0*3C
$PQTMCFGPROT,OK,1,0,00000005,00000005*6A
```

2.3.13. PQTMCFGNMEADP

Configures the decimal places of NMEA standard messages.

Type:

Set/Get

Synopsis:

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

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

Parameter:

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

Result:

- If successful, the module returns:

```
//Write:
```

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

//Read:
$PQTMCFGNMEADP,OK,<UTC_DP>,<POS_DP>,<ALT_DP>,<DOP_DP>,<SPD_DP>,<COG_DP>*<C
hecksum><CR><LF>
```

- If failed, the module returns:

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

For detail about **<ErrCode>**, see *Table 4: 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

Reports 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.

Field	Format	Unit	Description
<EPE_2D>	Numeric	Meter	Estimated 2D position error.
<EPE_3D>	Numeric	Meter	Estimated 3D position error.

Example:

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

2.3.15. PQTMCFGMSGRATE

Configures the message output rate on the current port.

Type:

Set/Get

Synopsis:

```
//Write:
$PQTMCFGMSGRATE,W,<MsgName>,<Rate>[,<MsgVer/Offset>]*<Checksum><CR><LF>
//Read:
$PQTMCFGMSGRATE,R,<MsgName>[,<MsgVer/Offset>]*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<MsgName>	String/Hex	-	The configuration message name. See *Table 5: Supported Messages*.
<Rate>	Numeric	-	The output rate of the message. 0 = Not output. N = Output once every N position fixed. For details on the range of N, See *Table 5: Supported Messages*.
<MsgVer/Offset>	Numeric	-	Message version or offset. ● Message version for **PQTM** message. ● Message offset for **RTCM MSM** message, and the offset. ● For others (such as standard NMEA messages, RTCM3-1005, RTCM3-1006, RTCM3-1019 and so on), should be omitted. **Range:** ● **Range of message version for PQTM:** depend on

Field	Format	Unit	Description
			the specific message. ● **Range of offset for RTCM3 MSM:** 0 to (**<Rate>**-1).

Result:

- If successful, the module returns:

```
//Write:
$PQTMCFGMSGRATE,OK*<Checksum><CR><LF>
//Read:
$PQTMCFGMSGRATE,OK,<MsgName>,<Rate>[,<MsgVer/Offset>]*<Checksum><CR><LF>
```

- If failed, the module returns:

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

For detail about **<ErrCode>**, see *Table 4: Error Codes*.

Example:

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

//Read the output rate of GGA message:
$PQTMCFGMSGRATE,R,GGA*12
$PQTMCFGMSGRATE,OK,GGA,1,*75

//Write the output rate of $PQTMEPE (version 1) message to once every position fix:
$PQTMCFGMSGRATE,W,PQTMEPE,1,1*1E
$PQTMCFGMSGRATE,OK*29


//Read the output rate of $PQTMEPE (version 1) message:
$PQTMCFGMSGRATE,R,PQTMEPE,1*06
$PQTMCFGMSGRATE,OK,PQTMEPE,1,1*4D


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

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

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

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

//Write the output rate of RTCM3 GPS MSM message to each epoch with delay 0:
$PQTMCFGMSGRATE,W,RTCM3-107X,1,0*2F
$PQTMCFGMSGRATE,OK*29

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

Table 5: Supported Messages

Message	Range (N)
RMC	1–20
GGA	1–20
GSV	1–20
GSA	1–20
VTG	1–20
GLL	1–20
PQTMEPE	1–20
PQTMVEL	1–20
PQTMGEOFENCESTATUS	1–20
PQTMTXT	1

Message	Range (N)
PQTMSVINSTATUS	1–20
PQTMPVT	1–20
PQTMDOP	1–20
PQTMPL	1–20
PQTMODO	1–20
RTCM3-1005	1-1200
RTCM3-1006	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

NOTE

1. If the configuration message includes message version, need use **<MsgVer>** field to specify the message version, otherwise an error will be returned.
2. If the configuration message is standard NMEA0183 message or RTCM message or without message version, the **<MsgVer>** field is omitted.
3. No synchronization required if messages have different rate profiles for standard NMEA0183 messages and PQTM messages, but the counter should start from 0 after each power on.
4. All MSM messages have the same **<Rate>** and only the last setting is valid, and the epoch time is aligned to **<FixInterval>*<Rate>**. For the details of **<FixInterval>**, please refer to *Chapter 0*

PQTMCFGFIXRATE.

5. The <Offset> used to configure the MSM messages output time = **<FixInterval>*<Rate>** + **<FixInterval>*<Offset>**, If the <Offset> in other messages which already configured does not meet the condition of (<Rate>-1) due to reconfiguring <Rate> in new messages, it is necessary to change the <Offset> in the configured conflicting messages to 0. This process is automatically implemented by software.
6. The RTCM EPH data output rate is independent of the <FixInterval>.

2.3.16. PQTMVEL

Outputs the velocity.

Type:

Output

Synopsis:

```
$PQTMVEL,<MsgVer>,<Time>,<VelN>,<VelE>,<VelD>,<GrdSpd>,<Spd>,<Heading>,<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.
<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.
<Heading>	Numeric	Degree	Heading.
<GrdSpdAcc>	Numeric	m/s	Estimate 2D speed accuracy.
<SpdAcc>	Numeric	m/s	Estimate 3D speed accuracy.
<HeadingAcc>	Numeric	Degree	Estimate heading 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

Configures geofence feature.

Type:

Set/Get

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<Index>	Numeric	-	The geofence index. Range: 0–3
<Status>	Numeric	-	The status of geofence. 0 = Disable. 1 = Enable.
<Reserved>	Numeric	-	Always 0.
<Shape>	Numeric	-	The shape of geofence. 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, etc.)
<Lat0>	Numeric	Degree	The latitude of the first point.
<Lon0>	Numeric	Degree	The 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	The longitude of the second point.

Field	Format	Unit	Description
<Lat2>	Numeric	Degree	The latitude of the third point.
<Lon2>	Numeric	Degree	The longitude of the third point.
<Lat3>	Numeric	Degree	The latitude of the fourth point.
<Lon3>	Numeric	Degree	The 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>,<State>,<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 detail about **<ErrCode>**, see *Table 4: 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 if it is enabled.
$PQTMCFGGEOFENCE,R,0*3E
$PQTMCFGGEOFENCE,OK,0,1,0,0,31.451248,117.451245,100.5*4B

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

NOTE

1. If geofencing is disabled, the fields after **<Status>** of set and get should be omitted, the response should only contain **<Index>** and **<Status>** when get the configuration of the geofence, please see the example above.
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] and negative indicates south latitude, the longitude range is [-180,+180] and negative indicates west longitude.

2.3.18. PQTMGEOFENCESTATUS

Outputs the geofences 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.
Start of repeat block. Repeat times: 4.			
<StateN>	Numeric	-	The state of geofence N. The range of N: 0-3. 0 = Unknow. 1 = Inside. 2 = Outside. Note: If the module is not 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 detail about **<ErrCode>**, see *Table 4: 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 detail about **<ErrCode>**, see *Table 4: Error Codes*.

Example:

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

2.3.21. PQTMTXT

Outputs short text messages, and longer text messages may be transmitted by using multiple sentences.

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 – 99.
<TextID>	Numeric	-	Text identifier. 01 = Notice. 02 = Warning. 03 = Error.
<Text>	String	-	Text message. Up to 57 characters including any code delimiters.

Example:

```
//The debug data outputs:
$PQTMTXT,1,01,01,01,0x105f0cf810417c00*1B
```

2.3.22. PQTMCFGSVIN

Configures the survey-in feature.

Type:

Set/Get

Synopsis:

```
//Write:
$PQTMCFGSVIN,W,<Mode>,<CFG_CNT>,<3D_AccLimit>,<ECEF_X>,<ECEF_Y>,<ECEF_Z>*<Checksum><CR><LF>
//Read:
$PQTMCFGSVIN,R*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Mode>	Numeric	-	Configure the survey-in mode. 0 = Disable. 1 = Survey-in mode. 2 = Fixed with ECEF mode.
<CFG_CNT>	Numeric	-	Survey-in minimum fixed times. Range: 0–86400.
<3D_AccLimit>	Numeric	Meter	Limit the 3D accuracy position in survey-in mode. when this field is 0 that means no limit.
<ECEF_X>	Numeric	Meter	WGS84 ECEF X coordinate.
<ECEF_Y>	Numeric	Meter	WGS84 ECEF Y coordinate.
<ECEF_Z>	Numeric	Meter	WGS84 ECEF Z coordinate.

Result:

- If successful, the module returns:

```
//Write:
$PQTMCFGSVIN,OK*<Checksum><CR><LF>
```

```
//Read:
$PQTMCFGSVIN,OK,<Mode>,<CFG_CNT>,<3D_AccLimit>,<ECEF_X>,<ECEF_Y>,<ECEF_Z>*<Checksum><CR><LF>
```

- If failed, the module returns:

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

For detail about **<ErrCode>**, see *Table 4: Error Codes*.

Example:

```
//Write:
$PQTMCFGSVIN,W,1,3600,1.2,-2519265.0514,4849534.9045,3277834.6432*01
$PQTMCFGSVIN,OK*70

//Read:
$PQTMCFGSVIN,R*26
$PQTMCFGSVIN,OK,1,3600,1.2,-2519265.0514,4849534.9045,3277834.6432*52
```

2.3.23. PQTMSVINSTATUS

Outputs the survey-in status.

Type:

Output

Synopsis:

```
$PQTMSVINSTATUS,<MsgVer>,<TOW>,<Valid>,<Res0>,<CurrentCNT>,<CFG_CNT>,<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. 1 = In-progress. 2 = Valid.

Field	Format	Unit	Description
<Res0>	Numeric	-	Reserved. Always null.
<Res1>	Numeric	-	Reserved.
<CurrentCNT>	Numeric	-	The number of position observations used during survey-in.
<CFG_CNT>	Numeric	-	The <CFG_CNT> field configured through the **$PQTMCFGSVIN** command. (Survey-in minimum fixed times).
<MeanX>	Numeric	Meter	Current survey-in mean position along X axis of ECEF coordinate system.
<MeanY>	Numeric	Meter	Current survey-in mean position along Y axis of ECEF coordinate system.
<MeanZ>	Numeric	Meter	Current survey-in mean position along Z axis of ECEF coordinate system.
<MeanAcc>	Numeric	Meter	Current survey-in mean position accuracy.

Example:

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

NOTE

This message requires the module to work in Base station mode, for details of base station mode, please see *Chapter 2.3.25 PQTMCFGRCVRMODE*.

2.3.24. PQTMPVT

Outputs the PVT (GNSS only) result. This message is controlled by **$PQTMCFGMSGRATE** command.

Type:

Output

Synopsis:

```
$PQTMPVT,<MsgVer>,<TOW>,<Date>,<Time>,<Rev>,<FixType>,<NumSV>,<LeapS>,<Lat>,<Lon>,<Alt>,<AltMsl>,<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.
<Time>	hhmmss.sss	-	UTC time.
<Rev>	Numeric	-	Reserved. Always null.
<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.
<HDOP>	Numeric	-	Horizontal dilution of precision. 99.99 if invalid.
<PDOP>	Numeric	-	Position (3D) dilution of precision. 99.99 if invalid.

Example:

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

```
//3D fix:
$PQTMPVT,1,31075000,20221225,083737.000,,3,09,18,31.12738291,117.26372910,34.212,5.267,3.212,2.928,0.238,4.346,34.12,2.16,4.38*51
```

2.3.25. PQTMCFGRCVRMODE

Configures the receiver working mode.

Type:

Set/Get

Synopsis:

```
//Write:
$PQTMCFGRCVRMODE,W,<Mode>*<Checksum><CR><LF>
//Read:
$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 output state. 2 = Base station. When set to this mode, the receiver will auto disable NMEA and enable RTCM MSM4, 1005 messages.

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 detail about **<ErrCode>**, see *Table 4: Error Codes*.

Example:

```
//Set:
$PQTMCFGRCVRMODE,W,2*29
$PQTMCFGRCVRMODE,OK*64
//Get:
$PQTMCFGRCVRMODE,R*32
$PQTMCFGRCVRMODE,OK,2*7A
```

2.3.26. PQTMDEBUGON

Enables debug log outputting. The debug on state can be saved by command **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 detail about **<ErrCode>**, see *Table 4: Error Codes*.

Example:

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

2.3.27. PQTMDEBUGOFF

Disables debug log outputting. The debug state can be saved by command **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 detail about **<ErrCode>**, see *Table 4: Error Codes*.

Example:

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

2.3.28. PQTMCFGFIXRATE

Configures the position fix rate.

Type:

Set/Get

Synopsis:

```
//Write:
$PQTMCFGFIXRATE,W,<FixInterval>*<Checksum><CR><LF>

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

Parameter:

Field	Format	Unit	Description
<FixInterval>	Numeric	ms	Configure the fix interval.

Result:

- If successful, the module returns:

```
//Write:
$PQTMCFGFIXRATE,OK*<Checksum><CR><LF>

//Read:
$PQTMCFGFIXRATE,OK,<FixInterval>*<Checksum><CR><LF>
```

- If failed, the module returns:

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

For detail about **<ErrCode>**, see *Table 4: Error Codes*.

Example:

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

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

NOTE

Fix rate of the module is fixed to 1 Hz and does not support to change in Base station mode and fix rate of the module is 10 Hz before changing fault value in Rover station mode, for details of base station mode, please see *Chapter 2.3.25 PQTMCFGRCVRMODE*.

2.3.29. PQTMCFGRTK

Configures the RTK mode.

Type:

Set/Get

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<DiffMode>	Numeric	-	Differential mode. 0 = Disable RTK/RTD feature, not using differential data. 1 = Auto mode. 2 = RTD only mode, only using the pseudoranges.
<RelMode>	Numeric	-	The absolute/relative mode. 1 = Absolute mode, ensure absolute position accuracy. 2 = Relative mode, ensure relative position accuracy. Note: This field only takes effect when the differential mode is RTK only mode (in auto mode configured by **<DiffMode>**).

Result:

- If successful, the module returns:

```
//Write:
$PQTMCFGRTK,OK*<Checksum><CR><LF>
//Read:
$PQTMCFGRTK,OK,<DiffMode>,<RelMode>*<Checksum><CR><LF>
```

- If failed, the module returns:

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

For detail about **<ErrCode>**, see *Table 4: Error Codes*.

Example:

```
//Set:
$PQTMCFGRTK,W,1,1*6C
$PQTMCFGRTK,OK*3F
```

```
//Get:
$PQTMCFGRTK,R*69
$PQTMCFGRTK,OK,1,1*3F
```

2.3.30. PQTMCFGCNST

Configures a constellation.

Type:

Set/Get

Synopsis:

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

Parameter:

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

Result:

- If successful, the module returns:

```
//Write:
$PQTMCFGCNST,OK*<Checksum><CR><LF>
//Read:
$PQTMCFGCNST,OK,<GPS>,<GLONASS>,<Galileo>,<BDS>,<QZSS>,<NavIC>*<Checksum><CR><LF>
```

- If failed, the module returns:

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

For detail about **<ErrCode>**, see *Table 4: Error Codes*.

Example:

```
//Set:
$PQTMCFGCNST,W,1,1,1,1,0,0*2B
$PQTMCFGCNST,OK*78

//Get:
$PQTMCFGCNST,R*2E
$PQTMCFGCNST,OK,1,1,1,1,0,0*78
```

2.3.31. PQTMDOP

Outputs dilution of precision.

Type:

Output

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<MsgVer>	Numeric	-	Message version. Always 1 for this version.

Field	Format	Unit	Description
<TOW>	Numeric	Millisecond	Time of week. Null if invalid.
<GDOP>	Numeric	-	Geometric dilution of precision. Note that the value is 99.99 in case of an invalid value.
<PDOP>	Numeric	-	Position (3D) dilution of precision. Note that the value is 99.99 in case of an invalid value.
<TDOP>	Numeric	-	Time dilution of precision. Note that the value is 99.99 in case of an invalid value.
<VDOP>	Numeric	-	Vertical dilution of precision. Note that the value is 99.99 in case of an invalid value.
<HDOP>	Numeric	-	Horizontal dilution of precision. Note that the value is 99.99 in case of an invalid value.
<NDOP>	Numeric	-	Northing dilution of precision. Note that the value is 99.99 in case of an invalid value.
<EDOP>	Numeric	-	Easting dilution of precision. Note that the value is 99.99 in case of an invalid value.

Example:

```
//Fixed:
$PQTMDOP,1,570643000,1.01,0.88,0.49,0.73,0.50,0.36,0.35*7C
//No fixed:
$PQTMDOP,1,,99.99,99.99,99.99,99.99,99.99,99.99,99.99*70
```

2.3.32. PQTMPL

Outputs protection level information.

Type:

Output

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<MsgVer>	Numeric	-	Message version. Always 1 for this version.
<TOW>	Numeric	Millisecond	Time of week. Null if invalid.
<PUL>	Numeric	%	The probability of uncertainty level per epoch.
<Res1>	Numeric	-	Always 1.
<Res2>	Numeric	-	Always 1.
<PL_PosN>	Numeric	mm	The protection level of north position. Null if invalid.
<PL_PosE>	Numeric	mm	The protection level of east position. Null if invalid.
<PL_PosD>	Numeric	mm	The protection level of down position. Null if invalid.
<PL_VelN>	Numeric	mm/s	The protection level of north velocity. Null if invalid.
<PL_VelE>	Numeric	mm/s	The protection level of east velocity. Null if invalid.
<PL_VelD>	Numeric	mm/s	The protection level of down velocity. Null if invalid.
<Res3>	Numeric		Always null.
<Res4>	Numeric		Always null.
<PL_Time>	Numeric	ns	The 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

Configures the odometer feature.

Type:

Set/Get

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<State>	Numeric	-	Odometer feature state. 0 = Disabled. 1 = Enabled
<InitDist>	Numeric	Meter	The 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 4: Error Codes*.

Example:

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

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

2.3.34. PQTMRESETODO

Resets the accumulated distance of 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 4: Error Codes*.

Example:

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

NOTE

To reset the accumulated distance of the odometer, you can use **$PQTMRESETODO** command or power off the module. Disabling the odometer feature with **$PQTMCFGODO** command while the module is still working will stop distance calculation, but it cannot reset the distance to zero.

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: Hour (00–23) mm: Minute (00–59) ss: Second (00–59) sss: Decimal fraction of second
<State>	Numeric	-	The odometer status. 0 = Disabled 1 = Enabled
<Dist>	Numeric	Meter	The distance since last reset.

Example:

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

NOTE

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

2.3.36. PQTMCFGSIGNAL

Configures GNSS signal.

Type:

Set/Get

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<GPS_Sig>	Hexadecimal	-	GPS signal mask. 0 = Disable 1 = Enable Bit 0 = L1 C/A Bit 1 = L2C Bit 2 = L5-Q Default: 0x07
<GLO_Sig>	Hexadecimal	-	GLONASS signal mask. 0 = Disable 1 = Enable Bit 0 = G1 C/A Bit 1 = G2 C/A Default: 0x03
<GAL_Sig>	Hexadecimal	-	Galileo signal mask. 0 = Disable 1 = Enable Bit 0 = E1 Bit 1 = E5a Bit 2 = E5b Bit 3 = E6 Default: 0x0F
<BDS_Sig>	Hexadecimal	-	BDS signal mask. 0 = Disable 1 = Enable Bit 0 = B1I Bit 1 = B2I Bit 2 = B3I Bit 3 = B1C Bit 4 = B2a Bit 5 = B2b Default: 0x3F
<QZS_Sig>	Hexadecimal	-	QZSS signal mask.

Field	Format	Unit	Description
			0 = Disable 1 = Enable Bit 0 = L1 C/A Bit 1 = L2C Bit 2 = L5-Q Default: 0x07
<NAC_Sig>	Hexadecimal	-	NavIC signal mask. 0 = Disable 1 = Enable Bit 0 = L5 Default: 0x01

Result:

- If successful, the module returns:

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

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 4: Error Codes*.

Example:

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

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

NOTE

1. The L1 frequency bands of LG290P module cannot be disabled.
2. The priority of GNSS configuration: PQTMCFGCNST > PQTMCFGSIGNAL > PQTMCFGSAT.

2.3.37. PQTMCFGSAT

Configures 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

Field	Format	Unit	Description
			4 = B1C 5 = B2a 6 = B2b For QZSS: 1 = L1 C/A 2 = L2C 3 = L5-Q For NavIC: 1 = L5
<MaskLow>	Hexadecimal	-	GNSS satellite low 32-bit mask, bit 0 for the satellite PRN 1. 0 = Disable 1 = Enable Range: GPS: 0-0xFFFFFFFF GLONASS: 0-0x3FFF Galileo: 0-0xFFFFFFFF BDS: 0-0xFFFFFFFF QZSS: 0-0x3FF NavIC: 0-0x7FFF Defalut: GPS L1 C/A: 0xFFFFFFFF GPS L2C: 0xFFC36FFD GPS L5-Q: 0xAFC227AD GLONASS G1 C/A: 0x00003FFF GLONASS G2 C/A: 0x00003FFF Galileo E1: 0x67967FDF Galileo E5a: 0x67967FDF Galileo E5b: 0x67967FDF Galileo E6: 0x67967FDF BDS B1I: 0xBFFCBFFF BDS B2I: 0x0000BFFF BDS B3I: 0xBFFCBFFF BDS B1C: 0xBFFC0000 BDS B2a: 0xBFFC0000 BDS B2b: 0xBFFC0000 QZSS L1 C/A: 0x0000004E QZSS L2C: 0x0000004E QZSS L5-Q: 0x0000004E NavIC L5: 0x000001FF

Field	Format	Unit	Description
<MaskHigh>	Hexadecimal	-	GNSS satellite high 32-bit mask, bit 0 for the satellite PRN 33. (only available for BDS and Galileo, and for other GNSS system should be omitted). 0 = Disable 1 = Enable Range: GPS: None GLONASS: None Galileo: 0-0x0F BDS: 0-0xFFFFFFFF QZSS: None NavIC: None Defalut: 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:

```
//Set:
$PQTMCFGSAT,OK*<Checksum><CR><LF>
//Get:
$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 4: Error Codes*.

Example:

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

```
$PQTMCFGSAT,OK*34

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

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

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

2.3.38. PQTMCFGRSID

Configures 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	-	The reference station ID. Range: 0–4095. Default: 290.

Result:

- If successful, the module returns:

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

```
//Get:
$PQTMCFGRSID,OK,<ID>*<Checksum><CR><LF>
```

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 4: Error Codes*.

Example:

```
//Set:
$PQTMCFGRSID,W,1024*06
$PQTMCFGRSID,OK*7E
//Get:
$PQTMCFGRSID,R*28
$PQTMCFGRSID,OK,1024*55
```

2.3.39. PQTMCFGRTCM

Configure RTCM.

Type:

Set/Get

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<MSM_Type>	Numeric	-	The type of MSM. Range: 3-7, corresponding to MSM3-MSM7. Default: 4.
<MSM_Mode>	Numeric	-	The output mode of MSM when no satellite is searched. 0 = Not output MSM when no satellite is searched. Always 0.
<MSM_ElevThd	Numeric	Degree	The elevation threshold of satellite which report

Field	Format	Unit	Description
>			measurements in MSM. Range: [-90, 90], -90 means no limitation. Default: -90
<Reserved>	Numeric	-	Default: 0x07.
<Reserved>	Numeric	-	Default: 0x06.
<EPH_Mode>	Numeric	-	The output mode of ephemeris. 0 = Disable 1 = Output when updating 2 = Output when updating or output at certain intervals 3 = Output on each epoch Default: 1.
<EPH_Interval>	Numeric	Second	The output interval of ephemeris. Only available for **<EPH_Mode>** = 2. Range: 0-7200. Default: 0.

Result:

- If successful, the module returns:

```
//Write:
$PQTMCFGRTCM,OK*<Checksum><CR><LF>
//Read:
$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 4: Error Codes*.

Example:

```
$PQTMCFGRTCM,W,4,0,-90,07,06,1,0*25
$PQTMCFGRTCM,OK*7A
```

3 RTCM Protocol

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

Table 6: 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
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
1085	Input/Output	GLONASS MSM5
1086	Input/Output	GLONASS MSM6

Message Type	Mode	Message Name
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

4 Appendix A References

Table 7: Terms and Abbreviations

Abbreviation	Description
2D	2 Dimension
3D	3 Dimension
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
EPH	Ephemeris
GGA	Global Positioning System Fix Data
GLL	Geographic Position - Latitude and Longitude
GLONASS	Global Navigation Satellite System (Russia)
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GSA	GNSS DOP and Active Satellites
GSV	GNSS Satellites in View
HDOP	Horizontal Dilution of Precision
IRNSS	Indian Regional Navigation Satellite System
NavIC	Navigation with Indian Constellation
NMEA	NMEA (National Marine Electronics Association) 0183 Interface Standard

Abbreviation	Description
PDOP	Position Dilution of Precision
PPS	Pulse Per Second
QZSS	Quasi-Zenith Satellite System
RMC	Recommended Minimum Specific GNSS Data
RTD	Real-Time Differential
RTK	Real-Time Kinematic
SBAS	Satellite-Based Augmentation System
SOG	Speed over Ground
SPS	Standard Positioning Service
TXT	Text
UART	Universal Asynchronous Receiver/Transmitter
UTC	Coordinated Universal Time
VDOP	Vertical Dilution of Precision
VTG	Course Over Ground & Ground Speed

5 Appendix B GNSS (NMEA) Numbering

Table 8: 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 = L1
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.

6 Appendix C Special Characters

Table 9: Special Characters

Special Character	Definition
<...>	Parameter name. Angle brackets do not appear in the message.
[...]	Optional field of a message. Square brackets do not appear in the message.
{…}	Repeated field of a message. Curly brackets do not appear in the message.
Underline	Default setting of a parameter.