LG77L (C)&Lx0&Lx6&LC86L Series

GNSS Protocol Specification

GNSS Module Series

Version: 2.2

Date: 2024-03-29

Status: Released

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

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

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

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

Legal Notices

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

Use and Disclosure Restrictions

License Agreements

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

Copyright

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

Trademarks

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

Third-Party Rights

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

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

Privacy Policy

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

Disclaimer

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

About the Document

Document Information	
Title	**LG77L (C)&Lx0&Lx6&LC86L Series GNSS Protocol Specification**
Subtitle	GNSS Module Series
Document Type	GNSS Protocol Specification
Document Status	Released

Revision History

Version	Date	Description
1.0	2019-05-15	Initial
1.1	2020-03-07	1. Added applicable modules L26-LB and LC86L 2. Deleted the following packet types: ● 300 PMTK_API_SET_FIX_CTL ● 458 PMTK_API_GET_POS_XYZ ● 461 PMTK_API_GET_VEL_XYZ 3. Updated NMEA standard messages according to NMEA V4.10 4. Updated the following packet types: ● 001 PMTK_ACK ● 225 PMTK_SET_PERIODIC_MODE ● 183 PMTK_LOCUS_QUERY_STATUS
2.0	2021-11-30	1. Updated the overall structure of the document. 2. Added LG77L, L26, L76, L76-L, L86, L96, L70 and L80 as the applicable modules of this document. 3. Added GBS and DTM messages (Chapter 2.2.8 and Chapter 2.2.9). 4. Updated NMEA Talker ID (Table 2). 5. Added a new note for the parameter <Mode> in the Query Result Message returned with PMTK_LOCUS_QUERY_STATUS (Chapter 2.3.9).

Version	Date	Description
		6. Detailed the description for the parameter <Interval> in PMTK_SET_POS_FIX (Chapter 2.3.12). 7. Added range and default value for the parameter <Ext_Gap> in PMTK_SET_AL_DEE_CFG (Chapter 2.3.13). 8. Added the default value for the parameter <PPSPulseWidth> in PMTK_SET_PPS_CONFIG (Chapter 2.3.18). 9. Added new notes for PMTK_API_SET_DGPS_MODE (Chapter 2.3.20). 10. Added PQ messages (Chapter 2.4). 11. Deleted the following packet types: ● 400 PMTK_API_Q_FIX_CTL ● 500 PMTK_DT_FIX_CTL
2.1	2022-04-01	1. Updated the note for NMEA Talker ID (Chapter 2.1). 2. Added PQVERNO message (Chapter 2.4.17).
2.2	2024-03-09	1. Deleted the EOL products: LG77L (A, B). 2. Added the table about Supported protocol (*Chapter 1*). 3. Added the NMEA Checksum Sample Code (*Chapter 2.1*). 4. Updated some parameter descriptions and examples of standard NMEA messages (*Chapter 2.2*). 5. Updated the range of repeat times of <SatID>, <SatElev>, <SatAz> and <SatCN0> (*Chapter 2.2.3*). 6. Added ZDA, GRS and GST messages and updated some reserved parameters to <GRS>,<GST>, <ZDA> and <DTM> (*Chapters 2.2.10* to *2.2.12*, *Chapters 2.3.24* and *2.3.35*). 7. Updated the message type of $PMTK 183 message (*Chapter 2.3.9*). 8. Added a note that when <Type> is 0, the fields after <Type> can be omitted (*Chapter 2.3.14*). 9. Updated the notes for $PMTK311 and $PQGEO commands (*Chapters 2.3.22* and *2.4.14*). 10. Added the notes for $PMTK353, $PMTK355, $PMTK622, $PMTKSPF and $PQRLM commands (*Chapters 2.3.27*, *2.3.28*, *2.3.37*, *2.3.43* and *2.4.13*). 11. Added $PMTK355 message (*Chapter 2.3.28*). 12. Updated the message types and added the results of $PMTK401, $PMTK413, $PMTK414 and $PMTK605 commands (*Chapters 2.3.30*, *2.3.31*, *2.3.32*, and *2.3.36*). 13. Updated the message types and results of $PMTK869 and $PMTK875 commands (*Chapters 2.3.40* and *0*). 14. Updated the GNSS Numbering and added notes (*Chapter 4*).

Contents

Table Index

1 Introduction

Quectel L70, L80, L26, L76, L76-L, L86, L96, L26-LB, L76-LB, LC86L (A, B, C) and LG77L (C) are multi-constellation modules. See *Table 1: Applicable Modules and Constellations Supported* on the actual constellations supported on each applicable module. Also, they support autonomous GNSS C/A code, SBAS function (including WAAS, EGNOS, MSAS and GAGAN) and AGNSS (EASY function). The tracking of GPS L1 C/A, GLONASS L1, BDS B1, QZSS L1, Galileo E1 frequency bands provides fast and accurate acquisition and makes those modules 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 the chipset supplier (PMTK commands) and Quectel (PQ commands). To report GNSS information, the modules support output messages in NMEA 0183 standard protocol.

Table 1: Applicable Modules and Constellations Supported

Module Family	Module Series	Model	Constellation
Lx0	-	L70	GPS + QZSS
	-	L80	GPS + QZSS
Lx6	-	L26	GPS + GLONASS + Galileo + BDS + QZSS
	-	L76	GPS + GLONASS + Galileo + BDS + QZSS
	-	L76-L	GPS + GLONASS + Galileo + BDS + QZSS
	-	L86	GPS + GLONASS + Galileo + BDS + QZSS
	-	L96	GPS + GLONASS + Galileo + BDS + QZSS
	-	L26-LB	GPS + GLONASS + BDS + QZSS
	-	L76-LB	GPS + GLONASS + BDS + QZSS
-	LC86L	LC86L (A)	GPS + GLONASS + BDS + QZSS
		LC86L (B)	GPS + GLONASS + BDS + QZSS

Module Family	Module Series	Model	Constellation
		LC86L (C)	GPS + GLONASS + Galileo + BDS + QZSS
-	-	LG77L (C)	GPS + GLONASS + Galileo + BDS + QZSS

The modules support the following protocols:

Table 2: Supported Protocol

Protocol	Type
NMEA 0183 V4.10	Output, ASCII, standard
	Input/output, ASCII, proprietary

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

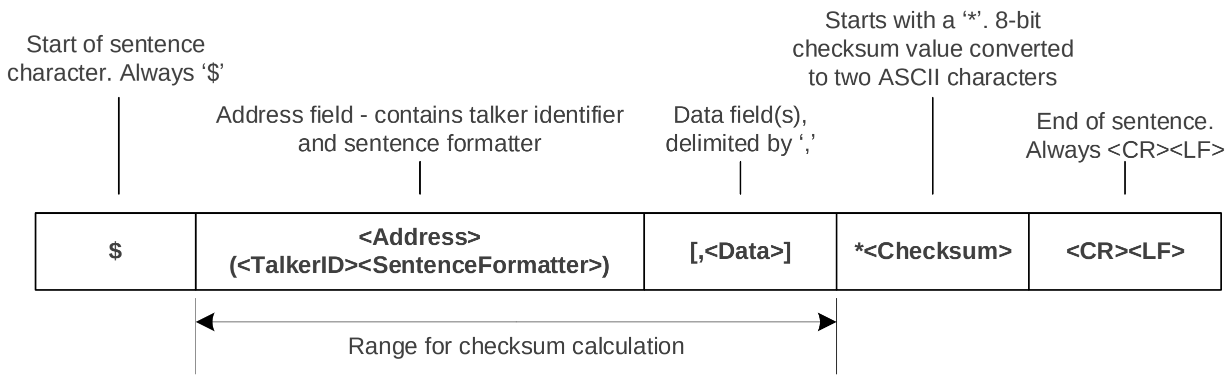
Start of sentence character. Always ‘$’
Address field - contains talker identifier and sentence formatter
Data field(s), delimited by ‘,’
Starts with a ‘*’. 8-bit checksum value converted to two ASCII characters
End of sentence. Always <CR><LF>
$
<Address> (<TalkerID><SentenceFormatter>)
[,<Data>]
*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

Figure 1: Structure of NMEA Protocol Messages

Table 3: Structure of NMEA Protocol Messages

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

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

Table 4: NMEA Talker ID

GNSS Constellation Configuration	TalkerID (NMEA 0183 V4.10)
GPS	GP
GLONASS	GL
Galileo	GA
BDS	BD
QZSS	QZ
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;
}
```

NOTE

On L26, L76, L76-L, L86 and L96, the Talker ID in the following listed messages has been modified to be compatible with the previous protocol versions and the Talker ID in other standard NMEA messages. Unless otherwise specified, the Talker ID complies with the rules listed in *Table 4: NMEA Talker ID*.

1. The Talker ID listed the messages when the position is fixed.

Constellation	RMC	GGA	GSA	GLL	VTG
GPS + GLONASS	GN	GP	GN	GN	GP
GPS + BDS	GP	GP	GP	GP	GP
GPS + Galileo	GN	GP	GN	GN	GP
GPS + GLONASS + Galileo	GN	GP	GN	GN	GP

2. The Talker ID is always **GP** when the position is unfixed.

2.2. Standard Messages

This chapter explains the standard NMEA 0183 V4.10 messages supported by the modules.

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 4: NMEA Talker ID*.

Field	Format	Unit	Example	Description
RMC	String, 3 characters	-	RMC	Recommended Minimum Specific GNSS Data.
<UTC>	hhmmss.sss	-	080608.000	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.mmmmmm	-	3029.461489	Latitude. dd: Degrees (00–90) mm: Minutes (00–59) mmmmmm: Decimal fraction of minutes. (variable length 4 to 6 digits) 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.mmmmmm	-	11430.072002	Longitude. ddd: Degrees (000–180) mm: Minutes (00–59) mmmmmm: Decimal fraction of minutes. (variable length 4 to 6 digits) 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	0.00	Speed over ground. Variable length.
<COG>	Numeric	Degree	148.41	Course over ground. Variable length. Maximum value: 359.99
<Date>	ddmmyy	-	210423	Date. dd: Day of month mm: Month yy: Year

Field	Format	Unit	Example	Description
<MagVar>	-	-	-	Magnetic variation. Not supported.
<MagVarDir>	-	-	-	Direction of magnetic variation. Not supported.
<ModeInd>	Character	-	D	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 N = No fix. Satellite system not used in position fix, or fix not valid
<NavStatus>	Character	-	V	Navigational status. Not supported. Always "V" (Navigational status not valid).
<Checksum>	Hexadecimal	-	*09	Checksum.
<CR><LF>	Character	-	-	Carriage return and line feed.

Example:

Take LC86L (B) for example:

```
//GPS + GLONASS + QZSS mode:
$GNRMC,080608.000,A,3029.461489,N,11430.072002,E,0.00,148.41,210423,,,D,V*09
//GPS only mode:
$GPRMC,072316.000,A,3029.464977,N,11430.067548,E,0.00,5.09,230423,,,D,V*10
```

NOTE

The length of minutes decimal fraction is 4 digits in **<Lat>** and **<Lon>** when **RMC** is supported by L26, L76, L76-L, L86, L96, L70 and L80 modules.

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 4: NMEA Talker ID*.
GGA	String, 3 characters	-	GGA	Global Positioning System Fix Data.
<UTC>	hhmmss.sss	-	072741.000	Position fix UTC. hh: Hours (00–23) mm: Minutes (00–59) ss: Seconds (00–59) sss: Decimal fraction of seconds
<Lat>	ddmm.mmmmmm	-	3029.463944	Latitude. dd: Degrees (00–90) mm: Minutes (00–59) mmmmmm: Decimal fraction of minutes (variable length, 4 to 6 digits) 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.mmmmmm	-	11430.070240	Longitude. ddd: Degrees (000–180) mm: Minutes (00–59) mmmmmm: Decimal fraction of

Field	Format	Unit	Example	Description
				minutes (variable length, 4 to 6 digits) 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	-	2	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 6 = Estimated (dead reckoning) mode
<NumSatUsed>1)	Numeric, 2 digits	-	17	Number of satellites in use.
<HDOP>	Numeric	-	0.64	Horizontal dilution of precision. Note that this field is empty in case of an invalid value.
<Alt>	Numeric	Meter	141.895	Altitude above mean-sea-level (geoid). Note that this field is empty in case of an invalid value.
M	Character	-	M	Unit of <Alt>. “M” = Meter.
<Sep>	Numeric	Meter	-13.122	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). Note that this field is empty in case of an invalid value.
M	Character	-	M	Unit of <Sep>. “M” = Meter.
<DiffAge>	-	-	-	Differential GPS data age. Not supported.
<DiffStation>	-	-	-	Differential reference station ID. Not supported.

Field	Format	Unit	Example	Description
<Checksum>	Hexadecimal	-	*5A	Checksum.
<CR><LF>	Character	-	-	Carriage return and line feed.

Example:

Take LC86L (B) for example:

```
//GPS + GLONASS + QZSS mode:
$GNGGA,072741.000,3029.463944,N,11430.070240,E,2,17,0.64,141.895,M,-13.122,M,,*5A
//GPS only mode:
$GPGGA,072357.000,3029.464977,N,11430.067548,E,1,11,0.90,142.212,M,-13.122,M,,*41
```

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 **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.
3. The length of minutes decimal fraction is 4 digits in **<Lat>** and **<Lon>** when the **GGA** message is supported by L26, L76, L76-L, L86, L96, L70 and L80 modules.
4. The length of minutes decimal fraction is 1 digit in **<Alt>** and **<Sep>** when the **GGA** message is supported by L26, L76, L76-L, L86, L96, L70 and L80 modules.

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 **$**.
<TalkerID>	String, 2 characters	-	GP	Talker identifier. See *Table 4: NMEA Talker ID*.
GSV	String, 3 characters	-	GSV	GNSS Satellites in view.
<TotalNumSen>	Numeric	-	3	Total number of sentences. Range: 1–9.
<SenNum>	Numeric	-	1	Sentence number. Range: 1–**<TotalNumSen>**.
<TotalNumSat>	Numeric	-	11	Total number of satellites in view.
Start of repeat block. Repeat times: 0–4.				
<SatID>	Numeric	-	19	Satellite ID. See *Table 6: GNSS Numbering*.
<SatElev>	Numeric	Degree	67	Satellite elevation. Range: 00–90. Note that this field is empty in case of an invalid value.
<SatAz>	Numeric	Degree	075	Satellite azimuth, with true north as the reference plane. Range: 000–359. Note that this field is empty in case of an invalid value.
<SatCN0>	Numeric	dB-Hz	44	Satellite C/N$_0$. Range 00–99. Null when not tracking.
End of repeat block.				
<SignalID>	Numeric	-	1	GNSS signal ID. See *Table 6: GNSS Numbering*.
<Checksum>	Hexadecimal	-	*60	Checksum.
<CR><LF>	Character	-	-	Carriage return and line feed.

Example:

Take LC86L (B) for example:

```
//GPS + GLONASS + QZSS mode:
$GPGSV,3,1,12,19,68,070,41,06,57,003,18,17,50,104,40,11,41,290,39,1*69
$GPGSV,3,2,12,50,41,130,33,20,37,226,38,09,28,096,38,12,27,308,37,1*61
$GPGSV,3,3,12,04,17,064,,05,11,219,31,14,10,168,,03,02,039,,1*6A
$QZGSV,1,1,02,03,64,068,38,02,30,166,29,1*75
```

```
$GLGSV,1,1,04,67,50,135,28,66,39,054,44,80,29,268,32,79,16,213,29,1*72
//GPS only mode:
$GPGSV,3,1,11,19,67,075,44,06,56,006,45,17,49,107,44,42,44,136,,1*60
$GPGSV,3,2,11,11,42,292,42,20,38,227,42,09,29,094,42,12,28,307,40,1*64
$GPGSV,3,3,11,04,16,063,24,05,12,220,28,14,08,168,,1*5A
```

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 4: 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 2 = 2D 3 = 3D

Field	Format	Unit	Example	Description
Start of repeat block. Repeat times: 12.				
<SatID>	Numeric	-	30	ID numbers of satellites used in solution. See *Table 6: GNSS Numbering*.
End of repeat block.				
<PDOP>	Numeric	-	1.30	Position dilution of precision. Maximum value：99.99.
<HDOP>	Numeric	-	0.67	Horizontal dilution of precision. Maximum value: 99.99.
<VDOP>	Numeric	-	1.12	Vertical dilution of precision. Maximum value: 99.99.
<SystemID>	Numeric	-	1	GNSS system ID. See *Table 6: GNSS Numbering*.
<Checksum>	Hexadecimal	-	*00	Checksum.
<CR><LF>	Character	-	-	Carriage return and line feed.

Example:

Take LC86L (B) for example:

```
//GPS + GLONASS + QZSS mode:
$GNGSA,A,3,30,01,04,07,08,195,16,194,21,27,09,,1.30,0.67,1.12,1*00
$GNGSA,A,3,87,76,78,88,77,68,67,,,,,,1.30,0.67,1.12,2*0D
//GPS only mode:
$GPGSA,A,3,30,14,19,17,03,06,,,,,,,1.32,0.99,0.87,1*1D
```

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 4: NMEA Talker ID*.
VTG	String, 3 characters	-	VTG	Course Over Ground & Ground Speed.
<COGT>	Numeric	Degree	191.88	Course over ground, in true north course direction.
T	Character	-	T	Fixed field: true.
<COGM>	Numeric	Degree	-	Course over ground (magnetic). Not supported.
M	Character	-	M	Fixed field: magnetic.
<SOGN>	Numeric	Knot	0.14	Speed over ground in knots.
N	Character	-	N	Fixed field: knot.
<SOGK>	Numeric	km/h	0.26	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	-	*2B	Checksum.
<CR><LF>	Character	-	-	Carriage return and line feed.

Example:

Take LC86L (B) for example:

```
//GPS + GLONASS +QZSS mode:
$GNVTG,191.88,T,,M,0.14,N,0.26,K,A*2B
//GPS only mode:
$GPVTG,321.97,T,,M,0.00,N,0.00,K,A*33
```

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 4: NMEA Talker ID*.
GLL	String, 3 characters	-	GLL	Geographic Position – Latitude/Longitude.
<Lat>	ddmm.mmmmmm	-	3029.465454	Latitude. dd: Degrees (00–90) mm: Minutes (00–59) mmmmmm: Decimal fraction of minutes (variable length, 4 to 6 digits) 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.mmmmmm	-	11430.074441	Longitude. ddd: Degrees (000–180)

Field	Format	Unit	Example	Description
				mm: Minutes (00–59) mmmmmm: Decimal fraction of minutes (variable length, 4 to 6 digits) 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.

<UTC>	hhmmss.sss	-	020549.000	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	-	*42	Checksum.
<CR><LF>	Character	-	-	Carriage return and line feed.

Example:

Take LC86L (B) for example:

```
//GPS + GLONASS +QZSS mode:
$GNGLL,3029.465454,N,11430.074441,E,020549.000,A,A*42
//GPS only mode:
$GPGLL,3029.464259,N,11430.069356,E,020628.000,A,A*5F
```

NOTE

The length of minutes decimal fraction is 4 digits in **<Lat>** and **<Lon>** when the **GLL** is supported by L26, L76, L76-L, L86, L96, L70 and L80 modules.

2.2.7. TXT

Text Transmission.

Type:

Output

Synopsis:

```
$<TalkerID>TXT,<TotalNumSen>,<SenNum>,<TextID>,<TextMsg>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Example	Description
$	Character	-	$	Each NMEA message starts with **$**.
<TalkerID>	String, 2 characters	-	GP	Talker identifier. Always "**GP**".
TXT	String, 3 characters	-	TXT	Text Transmission.
<TotalNumSen>	Numeric	-	01	Total number of sentences. Always "01".
<SenNum>	Numeric	-	01	Sentence number. Always "01".
<TextID>	Numeric	-	02	Text identifier, used to identify different text messages. Always "02".
<TextMsg>	String	-	ANTSTATUS=OK	ASCII characters and code delimiters, if needed; up to the maximum permitted sentence length (i.e., up to 61 characters including any code delimiters).
<Checksum>	Hexadecimal	-	*3B	Checksum.
<CR><LF>	Character	-	-	Carriage return and line feed.

Example:

Take LC86L (B) for example:

```
//GPS only or GPS + GLONASS + QZSS mode:
$GPTXT,01,01,02,ANTSTATUS=OK*3B
```

NOTE

This message is supported by L26, L86, L26-LB, L80, LC86L and LG77L (C) modules and the **<TalkerID>** is always "**GP**".

2.2.8. GBS

GNSS Satellite Fault Detection. This sentence is used to support Receiver Autonomous Integrity Monitoring (RAIM).

Type:

Output

Synopsis:

```
$<TalkerID>GBS,<UTC>,<LatExpErr>,<LonExpErr>,<AltExpErr>,<SatID>,<Probability>,<SatEstBias>,<StdDev>,<SystemID>,<SignalID>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Example	Description
$	Character	-	$	Each NMEA message starts with **$**.
<TalkerID>	String, 2 characters	-	GN	Talker identifier. See *Table 4: NMEA Talker ID*.
GBS	String, 3 characters	-	GBS	GNSS Satellite Fault Detection.
<UTC>	hhmmss.sss	-	023333.000	Position fix UTC. hh: Hours (00–23) mm: Minutes (00–59) ss: Seconds (00–59) sss: Decimal fraction of seconds
<LatExpErr>	Numeric	Meter	2.95	Expected Error in latitude.
<LonExpErr>	Numeric	Meter	3.12	Expected Error in longitude.

Field	Format	Unit	Example	Description
<AltExpErr>	Numeric	Meter	12.06	Expected Error in altitude.
<SatID>	Numeric	-	-	ID number of most likely failed satellite. Note that this field is empty in case of an invalid value.
<Probability>	Numeric	-	-	Probability of missed detection for most likely failed satellite. Note that this field is empty in case of an invalid value.
<SatEstBias>	Numeric	Meter	-	Estimate of bias in meters on most likely failed satellite. Note that this field is empty in case of an invalid value.
<StdDev>	Numeric	-	-	Standard deviation of bias estimate. Note that this field is empty in case of an invalid value.
<SystemID>	Numeric	-	-	GNSS system ID. See *Table 6: GNSS Numbering*. Note that this field is empty in case of an invalid value.
<SignalID>	Numeric	-	-	GNSS signal ID. See *Table 6: GNSS Numbering*. Note that this field is empty in case of an invalid value.
<Checksum>	Hexadecimal	-	*66	Checksum.
<CR><LF>	Character	-	-	Carriage return and line feed.

Example:

Take LC86L (B) for example:

```
//GPS + GLONASS + QZSS mode:
$GNGBS,023333.000,2.95,3.12,12.06,,,,,,*66
//GPS only mode:
$GPGBS,023620.000,4.08,4.01,14.58,,,,,,*75
```

NOTE

This message is not supported on L70 and L80 modules.

2.2.9. DTM

Local geodetic datum and datum offsets from a reference datum. This sentence is used to define the datum to which a position location, and geographic locations in subsequent sentences, is referenced. Latitude, longitude and altitude offsets from the reference datum, and the selection of reference datum, are also provided.

Type:

Output

Synopsis:

```
$<TalkerID>DTM,<Datum>,<DatumSub>,<LatOffset>,<N/S>,<LonOffset>,<E/W>,<AltOffset>,<RefDatum>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Example	Description
$	Character	-	$	Each NMEA message starts with **$**.
<TalkerID>	String, 2 characters	-	GN	Talker identifier. See *Table 4: NMEA Talker ID*.
DTM	String, 3 characters	-	DTM	Datum Reference.

<Datum>	String	-	W84	Local datum code.
				WGS84 = W84
				WGS72 = W72
				SGS85 = S85
				PE90 = P90
				User defined = 999
				IHO datum code.

<DatumSub>	Character	-	-	Local datum subdivision code. One character subdivision datum code when available or user defined reference character for user defined datums, null field otherwise.
<LatOffset>	Numeric	Minute	0.000000	Offset in Latitude.

<N/S>	Character	-	S	North-south direction.
				N = North
				S = South

<LonOffset>	Numeric	Minute	0.000000	Offset in Longitude.

Field	Format	Unit	Example	Description

<E/W>	character	-	E	East-west direction.
				E = East
				W = West

<AltOffset>	Numeric	Meter	0.00	Offset in altitude.

<RefDatum>	string	-	W84	Reference datum code.
				WGS84 = W84
				WGS72 = W72
				SGS85 = S85
				PE90 = P90

<Checksum>	Hexadecimal	-	*41	Checksum.
<CR><LF>	Character	-	-	Carriage return and line feed.

Example:

Take LC86L (B) for example:

```
//GPS + GLONASS + QZSS mode:
$GNDTM,W84,,0.000000,N,0.000000,E,0.00,W84*41
//GPS only mode:
$GPDTM,W84,,0.000000,N,0.000000,E,0.00,W84*5F
```

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, 2 characters	-	GN	Talker identifier. See *Table 4: NMEA Talker ID*.

Field	Format	Unit	Example	Description
ZDA	String, 3 characters	-	ZDA	Time & Date. UTC, day, month, year and local time zone.
<UTC>	hhmmss.sss	-	024112.000	Position fix UTC. hh: Hours (00–23) mm: Minutes (00–59) ss: Seconds (00–59) sss: Decimal fraction of seconds
<Day>	Numeric	-	24	Day of month. Range: 01–31.
<Month>	Numeric	-	04	Month. Range: 01–12.
<Year>	Numeric	-	2023	Year.
<LocalHour>	Numeric	-	-	Local zone hours, 00 to ±13 hours. Not supported.
<LocalMin>	Numeric	-	-	Local zone minutes, 00 to +59 minutes. Not supported.
<Checksum>	Hexadecimal	-	*4D	Checksum.
<CR><LF>	Character	-	-	Carriage return and line feed.

Example:

Take LC86L (B) for example:

```
//GPS + GLONASS + QZSS mode:
$GNZDA,024112.000,24,04,2023,,*4D
//GPS only mode:
$GPZDA,024222.000,24,04,2023,,*53
```

2.2.11. GRS

GNSS range residuals. This sentence supports Receiver Autonomous Integrity Monitoring (RAIM). Range residuals can be computed in two ways for this process. The basic measurement integration cycle of most navigation filters generates a set of residuals and uses these to update the position state of the receiver.

Type:

Output

Synopsis:

```
$<TalkerID>GRS,<UTC>,<Mode>{,<Resi>},<SystemID>,<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 4: NMEA Talker ID*.
GRS	String, 3 characters	-	GRS	GNSS range residuals.
<UTC>	hhmmss.sss	-	024321.000	Position fix UTC. hh: Hours (00–23) mm: Minutes (00–59) ss: Seconds (00–59) sss: Decimal fraction of seconds
<Mode>	Numeric	-	1	Calculation method. 0 = Residuals were used to calculate the position given in the matching GGA sentence. 1 = Residuals were recomputed after the GGA position was computed.
Start of repeat block. Repeat time: 12.				
<Resi>	Numeric	m	4.21	Range residuals for satellites used in navigation. Range: -999 to 999. Note that this field is empty in case of an invalid value.
End of repeat block.				
<SystemID>	Numeric	-	1	GNSS system ID. See *Table 6: GNSS Numbering*. Note that this parameter is only available in NMEA 0183 V4.10 or later.
<SignalID>	Numeric	-	1	GNSS signal ID. See *Table 6: GNSS Numbering*. Note that this parameter is only available in NMEA 0183 V4.10 or later.
<Checksum>	Hexadecimal	-	*71	Checksum.
<CR><LF>	Character	-		Carriage return and line feed.

Example:

Take LC86L (B) for example:

```
//GPS + GLONASS + QZSS mode:
$GPGRS,025820.000,1,-0.07,1.02,1.23,-2.96,-0.60,-5.48,2.52,-5.68,-5.72,-10.6,1.73,2.78,1,1*59
//GPS only mode:
$GPGRS,024321.000,1,4.21,0.07,-0.25,-4.18,-1.50,-0.57,3.48,1.63,-1.86,5.82,10.6,-8.98,1,1*71
```

NOTE

1. The satellite order in a GRS sentence should match the order of satellite ID numbers in a **GSA** sentence. If the range residual exceeds ±99.9 meters, then the decimal part is dropped, resulting in an integer.
2. The calculation method is: Range Residual = Calculated Range - Measured Range
3. If less than 12 satellites are used for navigation, the remaining **<Resi>** fields are left empty. If more than 12 satellites are used for navigation, only the IDs of the first 12 are output.

2.2.12. 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, 2 characters	-	GP	Talker identifier. See *Table 4: NMEA Talker ID*.
GST	String, 3 characters	-	GST	GNSS Pseudorange Error Statistics.
<UTC>	hhmmss.sss	-	024703.000	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
<RMS_D>	Numeric	Meter	3.3	RMS value of the standard deviation of the range inputs to the navigation process.
<MajorD>	Numeric	Meter	3.4	Standard deviation of semi-major axis of error ellipse.
<MinorD>	Numeric	Meter	2.8	Standard deviation of semi-minor axis of error ellipse.
<Orient>	Numeric	Degree	35.8	Orientation of semi-major axis of error ellipse.
<LatD>	Numeric	Meter	3.2	Standard deviation of latitude error.
<LonD>	Numeric	Meter	3.0	Standard deviation of longitude error.
<AltD>	Numeric	Meter	12	Standard deviation of altitude error.
<Checksum>	Hexadecimal	-	*79	Checksum.
<CR><LF>	Character	-	-	Carriage return and line feed.

Example

Take LC86L (B) for example:

```
//GPS + GLONASS + QZSS mode:
$GPGST,024703.000,3.3,3.4,2.8,35.8,3.2,3.0,12*79
//GPS only mode:
$GPGST,024422.000,3.6,3.9,3.7,140.6,3.8,3.8,15*47
```

2.3. PMTK Messages

This chapter explains the **PMTK** messages (proprietary NMEA messages defined by the chipset supplier) supported by the modules.

PMTK Message Format:

```
$PMTK<PacketType>[,<Data>]*<Checksum><CR><LF>
```

Packet Type: Three-byte character string, from 000 to 999. An identifier for each PMTK message.
Data: This field can be omitted, or multiple fields can be delimited by a data field delimiter **‘,’**. Different commands correspond to different data. See the specific values below.

2.3.1. PMTK001 PMTK_ACK

Acknowledges a PMTK command to inform the sender whether the receiver has received the packet.

The following commands will make the GNSS modules restart or change the baud rate, and thus there will be no **PMTK_ACK** for those commands.

- **PMTK_CMD_HOT_START**
- **PMTK_CMD_WARM_START**
- **PMTK_CMD_COLD_START**
- **PMTK_CMD_FULL_COLD_START**
- **PMTK_SET_NMEA_BAUDRATE**

Type:

Output

Synopsis:

```
$PMTK001,<Cmd>,<Flag>[,<para 1>,…,<para N>]*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Cmd>	Numeric	-	The packet type that the acknowledgement responds.
<Flag>	Numeric	-	ACK flag. 0 = Invalid packet 1 = Unsupported packet type 2 = Valid packet, but action failed 3 = Valid packet, and action succeeded

Field	Format	Unit	Description
[,<para 1>,…,<para N>]	Numeric	-	Extended parameters (Optional).

Example:

```
$PMTK001,869,3*37
```

2.3.2. PMTK010 PMTK_SYS_MSG

Automatically outputs system message when the modules are powered up.

Type:

Output

Synopsis:

```
$PMTK010,<Message>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Message>	Numeric	-	System message. 000 = Unknown 001 = Startup 002 = Notification for the host aiding EPO 003 = Notification that the transition to normal mode is successfully done

Example:

```
$PMTK010,002*2D
```

2.3.3. PMTK011 PMTK_TXT_MSG

Automatically outputs text message when the modules are powered up.

Type:

Output

Synopsis:

```
$PMTK011,<Message>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Message>	String	-	The system's text message.

Example:

```
$PMTK011,MTKGPS*08
```

2.3.4. PMTK101 PMTK_CMD_HOT_START

Performs a hot start on the modules (using all available data in the NVM). Normally a hot start means the GNSS modules are powered down less than 2 hours with RTC in operation and the ephemeris is still valid. As there is no need for downloading ephemeris, it is the fastest startup method.

Type:

Command

Synopsis:

```
$PMTK101*<Checksum><CR><LF>
```

Parameter:

None

Example:

```
$PMTK101*32
```

2.3.5. PMTK102 PMTK_CMD_WARM_START

Performs a warm start on the modules. A warm start means the GNSS modules contains approximate information of time, position and coarse data on satellite positions, but they need to download ephemeris until they can get a fix. Using this message will force a warm restart on the modules without using the ephemeris data in NVM.

Type:

Command

Synopsis:

```
$PMTK102*<Checksum><CR><LF>
```

Parameter:

None

Example:

```
$PMTK102*31
```

2.3.6. PMTK103 PMTK_CMD_COLD_START

Performs a cold start on the modules. This command will force a cold restart on the modules without using any prior location information, including position and ephemeris data.

Type:

Command

Synopsis:

```
$PMTK103*<Checksum><CR><LF>
```

Parameter:

None

Example:

```
$PMTK103*30
```

2.3.7. PMTK104 PMTK_CMD_FULL_COLD_START

Perfoms a full cold restart on the modules. It additionally clears system and user configurations at restart, that is, resetting the modules to the factory settings. A full cold start means there are no last position information in the modules. The modules need to search for the full time and frequency space, and also all possible satellites before they can get a fix.

Type:

Command

Synopsis:

```
$PMTK104*<Checksum><CR><LF>
```

Parameter:

None

Example:

```
$PMTK104*37
```

2.3.8. PMTK161 PMTK_CMD_STANDBY_MODE

Makes the modules enter Standby mode for power saving. Sending any data will make the modules exit Standby mode.

Type:

Command

Synopsis:

```
$PMTK161,<Type>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Type>	Numeric	-	Always "0".

Result:

Returns a **PMTK_ACK** message.

Example:

```
$PMTK161,0*28
$PMTK001,161,3*36
```

2.3.9. PMTK183 PMTK_LOCUS_QUERY_STATUS

Queries the LOCUS logging status.

Type:

Command

Synopsis:

```
$PMTK183*<Checksum><CR><LF>
```

Parameter:

None

Result:

Returns a **PMTK_ACK** message and the query result.

Query Result Message Format:

```
$PMTKLOG,<Serial#>,<Type>,<Mode>,<Content>,<Interval>,<Distance>,<Speed>,<Status>,<Number>,
<Percent>*<Checksum><CR><LF>
```

The parameters included in the result above are listed as below:

Field	Format	Unit	Description
<Serial#>	Numeric	-	Logging serial number. Range: 0–65535.
<Type>	Numeric	-	Logging type. 0 = Overlap 1 = Stop logging when full
<Mode>[1)]	Character	-	Logging mode: 0x02 = Fix only mode (logging when 3D-fix only) 0x04 = Normal mode (logging per positioning, e.g. 1 s) Customization mode: 0x08 = Interval mode (logging per pre-set interval, e.g. 15 s) 0x10 = Distance mode (logging when the distance exceeds the preset value, e.g. 10 m) 0x20 = Speed mode (logging when the speed exceeds the preset value, e.g. 10 m/s) Default value is 0x0b: Fix only mode + Interval mode.
<Content>	Numeric	-	Logging contents of configuration.
<Interval>	Numeric	-	Logging interval setting (valid when interval mode is selected).
<Distance>	Numeric	-	Logging distance setting (valid when distance mode is selected).
<Speed>	Numeric	-	Logging speed setting (valid when speed mode is selected).
<Status>	Numeric	-	Logging status. 0 = Logging 1 = Stop logging
<Number>	Numeric	-	Logging number of data record.
<Percent>	Numeric	-	Used percentage of logging capacity (0 %–100 %).

Example:

```
$PMTK183*38
$PMTK001,183,3*3A
$PMTKLOG,0,1,b,31,15,0,0,1,0,0*12
```

NOTE

[1] The “Fix only mode” is compatible with all other options. The “Interval mode”, “Distance mode” and “Speed mode” are called “Customization mode” and all of them are && condition with “Fix only mode” and “Normal mode“.

2.3.10. PMTK184 PMTK_LOCUS_ERASE_FLASH

Erases the LOCUS logger flash data.

Type:

Command

Synopsis:

```
$PMTK184,<Type>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Type>	Numeric	-	Erasing type. 1 = Erase all logger internal flash data

Result:

Returns a **PMTK_ACK** message.

Example:

```
$PMTK184,1*22
$PMTK001,184,3*3D
```

2.3.11. PMTK185 PMTK_LOCUS_STOP_LOGGER

Stops or starts LOCUS data logging.

Type:

Command

Synopsis:

```
$PMTK185,<Status>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Status>	Numeric	-	Start or stop LOCUS data logging. 0 = Start 1 = Stop

Result:

Returns a **PMTK_ACK** message.

Example:

```
$PMTK185,1*23
$PMTK001,185,3*3C
```

2.3.12. PMTK220 PMTK_SET_POS_FIX

Sets position fix interval.

Type:

Set

Synopsis:

```
$PMTK220,<Interval>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Interval>	Numeric	Millisecond	Position fix interval. Range: 100–10000. For value ≤ 1000, position fix interval is the value (time) set. For value > 1000 and value ≤ 10000, position fix interval is an integer multiple of 1000 (1000, 2000, …, 10000). Default value: 1000.

Result:

Returns a **PMTK_ACK** message.

Example:

```
$PMTK220,1000*1F
$PMTK001,220,3,1000*1D
```

NOTE

1. A higher fixed rate requires a higher baud rate, so that the data can be fully output. For example, if the fixed rate is 10 Hz (interval = 100 ms), the baud rate should be 57600 bps or higher.
2. 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.

2.3.13. PMTK223 PMTK_SET_AL_DEE_CFG

Configures DEE. This message sets dynamic ephemeris extension parameters in Periodic mode and should follow **PMTK_SET_PERIODIC_MODE**.

Type:

Set

Synopsis:

```
$PMTK223,<SV>,<SNR>,<Ext_Thr>,<Ext_Gap>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<SV>	Numeric	-	Required number of SV which satisfies the SNR condition to trigger dynamic ephemeris extension. Range: 1–4. Default value: 1.
<SNR>	Numeric	-	SV signal SNR criteria used to trigger dynamic ephemeris extension. Range: 25–30. Default value: 30.
<Ext_Thr>	Numeric	Millisecond	Time duration of dynamic ephemeris extension. Range: 40000–180000. Default value: 180000.
<Ext_Gap>	Numeric	Millisecond	The limitation of the interval between neighboring DEE intervals. Range: 0–3600000. Default value: 60000.

Result:

Returns a **PMTK_ACK** message.

Example:

```
$PMTK223,1,30,180000,60000*3C
$PMTK001,223,3*33
```

2.3.14. PMTK225 PMTK_SET_PERIODIC_MODE

Sets the modules to Periodic mode for power saving.

Type:

Set

Synopsis:

```
$PMTK225,<Type>[,<Run_Time>,<Sleep_Time>,<Sec_Run_Time>,<Sec_Sleep_Time>]*<Checksum><
CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Type>	Numeric	-	Power saving mode. 0 = Back to normal mode 1 = Periodic backup mode 2 = Periodic standby mode 4 = Perpetual backup mode Note that when **<Type>** is 0, the fields after **<Type>** can be omitted.
<Run_Time>	Numeric	Millisecond	Run time. 0 = Disable 1000–518400000 = Run time
<Sleep_Time>	Numeric	Millisecond	Sleep time. Range: 1000–518400000.
<Sec_Run_Time>	Numeric	Millisecond	Second run time. 0 = Disable 1000–518400000 = Second run time
<Sec_Sleep_Time>	Numeric	Millisecond	Second sleep time. Range: 1000–518400000.

Result:

Returns a **PMTK_ACK** message.

Example:

```
//Periodic standby mode:
$PMTK225,0*2B
$PMTK001,225,3*35
$PMTK225,2,3000,12000,18000,72000*15
$PMTK001,225,3*35
```

NOTE

The second run time should be longer than the first run time when the first run time is a non-zero value.

2.3.15. PMTK251 PMTK_SET_NMEA_BAUDRATE

Sets the baud rate of NMEA port. The baud rate will be restored to the default setting when the modules perform full cold restart or enter Standby mode.

Type:

Set

Synopsis:

```
$PMTK251,<Baudrate>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Baudrate>	Numeric	bps	The baud rate. 4800 9600 14400 19200 38400 57600 115200

Example:

```
$PMTK251,38400*27
```

2.3.16. PMTK255 PMTK_SET_SYNC_PPS_NMEA

Enables or disables the function of fixing NMEA output time in PPS function.

Type:

Set

Synopsis:

```
$PMTK255,<Enable>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Enable>	Numeric	-	Enable or disable the function of fixing NMEA output time in PPS function. 0 = Disable 1 = Enable

Result:

Returns a **PMTK_ACK** message.

Example:

```
$PMTK255,0*2C
$PMTK001,255,3*32
```

2.3.17. PMTK256 PMTK_SET_TIMING_PRODUCT

Enables or disables timing product mode. The timing product mode is used to enhance the PPS output timing accuracy.

Type:

Set

Synopsis:

```
$PMTK256,<Enable>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Enable>	Numeric	-	Enable or disable timing product mode. 0 = Disable 1 = Enable

Result:

Returns a **PMTK_ACK** message.

Example:

```
$PMTK256,0*2F
$PMTK001,256,3*31
```

2.3.18. PMTK285 PMTK_SET_PPS_CONFIG

Sets PPS type.

Type:

Set

Synopsis:

```
$PMTK285,<Type>,<PPSPulseWidth>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Type>	Numeric	-	PPS type. 0 = Disable 1 = After the first fix 2 = 3D fix only 3 = 2D/3D fix only 4 = Always
<PPSPulseWidth>	Numeric	Millisecond	PPS pulse width. Range: 2–998. Default value: 100.

Result:

Returns a **PMTK_ACK** message.

Example:

```
$PMTK285,2,100*3E
$PMTK001,285,3*3F
```

NOTE

Before you send **PMTK_SET_PPS_CONFIG**, make sure the baud rate is 19200 bps at least.

2.3.19. PMTK286 PMTK_SET_AIC_ENABLED

Enables or disables AIC function. It is recommended to send **PMTK_CMD_COLD_START** to force a cold restart on the modules before sending this command.

Type:

Set

Synopsis:

```
$PMTK286,<Enable>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Enable>	Numeric	-	Enable or disable AIC function. 0 = Disable 1 = Enable

Result:

Returns a **PMTK_ACK** message.

Example:

```
$PMTK286,1*23
$PMTK001,286,3*3C
```

2.3.20. PMTK301 PMTK_API_SET_DGPS_MODE

Configures the source mode of DGPS correction data.

Type:

Set

Synopsis:

```
$PMTK301,<Mode>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Mode>	Numeric	-	DGPS data source mode. 0 = No DGPS source 1 = RTCM 2 = SBAS (includes WAAS/EGNOS/GAGAN/MSAS)

Result:

Returns a **PMTK_ACK** message.

Example:

```
$PMTK301,1*2D
$PMTK001,301,3*32
```

NOTE

1. Before you send **PMTK_API_SET_DGPS_MODE** to select RTCM mode, make sure the baud rate is 115200 bps at least.
2. Send **PMTK_API_SET_SBAS_ENABLED** to enable SBAS searching first before setting SBAS as DGPS mode.

2.3.21. PMTK306 PMTK_API_SET_MIN_SNR

Sets the minimum SNR of satellites in use. If the minimum SNR threshold value is set, the modules would not use the satellite whose SNR is smaller than the shreshold value.

Type:

Set

Synopsis:

```
$PMTK306,<MIN_SNR>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<MIN_SNR>	Numeric	-	Minimum SNR threshold of satellites in use. Range: 9–37. Default value: 9.

Result:

Returns a **PMTK_ACK** message.

Example:

```
$PMTK306,15*1F
$PMTK001,306,3,15*1D
```

NOTE

This command is not supported on L70 and L80.

2.3.22. PMTK311 PMTK_API_SET_ELEV_MASK

Sets satellite elevation mask.

Type:

Set

Synopsis:

```
$PMTK311,<Degree>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Degree>	Numeric	Degree	Satellite elevation mask. Range: 0–90.

Result:

Returns a **PMTK_ACK** message.

Example:

```
$PMTK311,5*28
$PMTK001,311,3*33
```

NOTE

1. The satellite elevation mask is recommended to be no more than 10 degrees. With the increase of satellite elevation mask, the number of satellites involved in positioning will decrease.
2. This command is not supported on L70 modules.

2.3.23. PMTK313 PMTK_API_SET_SBAS_ENABLED

Enables or disables SBAS satellite searching. SBAS supports wide-area or regional augmentation through geostationary satellite broadcast messages. The geostationary satellite broadcasts GNSS integrity and correction data with the assistance of multiple ground stations which are located at accurately-surveyed points.

Type:

Set

Synopsis:

```
$PMTK313,<Enable>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Enable>	Numeric	-	Enable or disable SBAS satellite searching. 0 = Disable 1 = Enable

Result:

Returns a **PMTK_ACK** message.

Example:

```
$PMTK313,1*2E
$PMTK001,313,3*31
```

2.3.24. PMTK314 PMTK_API_SET_NMEA_OUTPUT

Sets NMEA sentence output frequencies. There are totally 22 data fields that present output frequency for the 22 supported NMEA sentences individually.

Type:

Set

Synopsis:

```
$PMTK314,<GLL>,<RMC>,<VTG>,<GGA>,<GSA>,<GSV>,<GRS>,<GST>,<Res1>,<Res2>,<Res3>,<R
es4>,<Res5>,<Res6>,<Res7>,<Res8>,<Res9>,<ZDA>,<Res10>,<DTM>,<GBS>,<Res11>*<Checksum
><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<GLL>	Numeric	-	**GLL** sentence output frequency. 0 = Disabled or not supported sentence n = Output once every "n" position fix. "n" ranges from 1 to 5

Field	Format	Unit	Description
<RMC>	Numeric	-	**RMC** sentence output frequency. 0 = Disabled or not supported sentence n = Output once every "n" position fix. "n" ranges from 1 to 5
<VTG>	Numeric	-	**VTG** sentence output frequency. 0 = Disabled or not supported sentence n = Output once every "n" position fix. "n" ranges from 1 to 5
<GGA>	Numeric	-	**GGA** sentence output frequency. 0 = Disabled or not supported sentence n = Output once every "n" position fix. "n" ranges from 1 to 5
<GSA>	Numeric	-	**GSA** sentence output frequency. 0 = Disabled or not supported sentence n = Output once every "n" position fix. "n" ranges from 1 to 5
<GSV>	Numeric	-	**GSV** sentence output frequency. 0 = Disabled or not supported sentence n = Output once every "n" position fix. "n" ranges from 1 to 5
<GRS>	Numeric	-	**GRS** sentence output frequency. 0 = Disabled or not supported sentence n = Output once every "n" position fix. "n" ranges from 1 to 5
<GST>	Numeric	-	**GST** sentence output frequency. 0 = Disabled or not supported sentence n = Output once every "n" position fix. "n" ranges from 1 to 5
<Res1>	Numeric	-	Always "0".
<Res2>	Numeric	-	Always "0".
<Res3>	Numeric	-	Always "0".
<Res4>	Numeric	-	Always "0".
<Res5>	Numeric	-	Always "0".
<Res6>	Numeric	-	Always "0".
<Res7>	Numeric	-	Always "0".
<Res8>	Numeric	-	Always "0".
<Res9>	Numeric	-	Always "0".
<ZDA>	Numeric	-	**ZDA** sentence output frequency. 0 = Disabled or not supported sentence n = Output once every "n" position fix. "n" ranges from 1 to 5
<Res10>	Numeric	-	Always "0".

Field	Format	Unit	Description
<DTM>	Numeric	-	**DTM** sentence output frequency. 0 = Disabled or not supported sentence n = Output once every "n" position fix. "n" ranges from 1 to 5
<GBS>	Numeric	-	**GBS** sentence output frequency. 0 = Disabled or not supported sentence n = Output once every “n” position fix. "n" ranges from 1 to 5
<Res11>	Numeric	-	Always "0".

Result:

Returns a **PAIR_ACK** message.

Example:

```
$PMTK314,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0*35
$PMTK001,314,3*36
```

NOTE

$PMTK314,-1*<Checksum><CR><LF> can be used to restore the NMEA sentence output frequency to the system default setting.

Result:

Returns a **PAIR_ACK** message. L26, L76, L76-L, L86, L96, L26-LB, L76-LB, LC86L and LG77L (C) will output **RMC**, **GGA**, **GSV**, **GSA**, **VTG**, **GLL** and **GBS** messages. L70 and L80 will output **RMC**, **GGA**, **GSV**, **GSA**, **VTG** and **GLL** messages.

Example:

```
$PMTK314,-1*04
$PMTK001,314,3*36
```

2.3.25. PMTK351 PMTK_API_SET_SUPPORT_QZSS_NMEA

Enables or disables QZSS NMEA format.

Type:

Set

Synopsis:

```
$PMTK351,<Enable>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Enable>	Numeric	-	Enable or disable QZSS NMEA format. 0 = Disable 1 = Enable

Result:

Returns a **PMTK_ACK** message.

Example:

```
$PMTK351,1*28
$PMTK001,351,3*37
```

2.3.26. PMTK352 PMTK_API_SET_STOP_QZSS

Enables or disables QZSS function.

Type:

Set

Synopsis:

```
$PMTK352,<Enable>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Enable>	Numeric	-	Enable or disable QZSS function. 0 = Enable 1 = Disable

Result:

Returns a **PMTK_ACK** message.

Example:

```
$PMTK352,0*2A
$PMTK001,352,3*34
```

2.3.27. PMTK353 PMTK_API_SET_GNSS_SEARCH_MODE

Configures the modules to start searching satellite system.

Type:

Set

Synopsis:

```
$PMTK353,<GPS_Enable>,<GLONASS_Enable>,<Galileo_Enabled>,<Res>,<BDS_Enable>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<GPS_Enable>	Numeric	-	Enable or disable GPS satellite searching. 0 = Disable 1 or other non-zero values = Enable
<GLONASS_Enable>	Numeric	-	Enable or disable GLONASS satellite searching. 0 = Disable 1 or other non-zero values = Enable
<Galileo_Enabled>	Numeric	-	Enable or disable Galileo satellite searching. 0 = Disable 1 or other non-zero values = Enable
<Res>	Numeric	-	Always "0".
<BDS_Enable>	Numeric	-	Enable or disable BDS satellite searching. 0 = Disable 1 or other non-zero values = Enable

Result:

Returns a **PMTK_ACK** message.

Example:

```
//Search GPS + GLONASS:
$PMTK353,1,1,0,0,0*2B
$PMTK001,353,3,1,1,0,0,0,3*36
```

```
//Search GPS + BDS:
$PMTK353,1,0,0,0,1*2B
$PMTK001,353,3,1,0,0,0,1,49*08
```

NOTE

1. This command is not supported on the L70 and L80.
2. GLONASS and BDS cannot be enabled at the same time. Galileo and BDS cannot be enabled at the same time.
3. GLONASS only, BDS only and Galileo only mode is only for testing purpose.
4. If the satellite system to be set is inconsistent with the current satellite system, the module will restart and satellite system set successfully. If the satellite system to be set is consistent with the current satellite system, the module will not restart and return an ACK response.

2.3.28. PMTK355 PMTK_API_GET_GNSS_SEARCH_MODE

Gets whether the GPS/GLONASS/BDS/Galileo satellite seraching is enabled or not.

Type:

Get

Synopsis:

```
$PMTK355*<Checksum><CR><LF>
```

Parameter:

None

Result:

```
$PMTK001,355,<Flag>,<GPS_Enable>,<GLONASS_Enable>,<BDS_Enable>,<Galileo_Enabled>*<Checksum><CR><LF>
```

Parameter included in the query result:

Field	Format	Unit	Description
<Flag>	Numeric	-	ACK flag. 0 = Invalid packet 1 = Unsupported packet 2 = Valid packet, but action failed

Field	Format	Unit	Description
			3 = Valid packet, and action succeeded
<GPS_Enable>	Numeric	-	Enable or disable GPS satellite searching. 0 = Disable 1 or other non-zero values = Enable
<GLONASS_Enable>	Numeric	-	Enable or disable GLONASS satellite searching. 0 = Disable 1 or other non-zero values = Enable
<BDS_Enable>	Numeric	-	Enable or disable BDS satellite searching. 0 = Disable 1 or other non-zero values = Enable
<Galileo_Enabled>	Numeric	-	Enable or disable Galileo satellite searching. 0 = Disable 1 or other non-zero values = Enable

Example:

```
$PMTK355*31
$PMTK001,355,3,1,1,0,0*33
```

NOTE

This command is not supported by L70 and L80.

2.3.29. PMTK386 PMTK_API_SET_STATIC_NAV_THD

Sets the speed threshold for static navigation. If the actual speed is below the threshold, the output position will remain the same and the output speed will be zero. If the threshold value is set to 0, this function is disabled.

Type:

Set

Synopsis:

```
$PMTK386,<Speed_Threshold>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Speed_Threshold>	Numeric	m/s	Speed threshold. Range: 0–2. Default value: 0.4.

Result:

Returns a **PMTK_ACK** message.

Example:

```
$PMTK386,0.4*39
$PMTK001,386,3*3D
```

2.3.30. PMTK401 PMTK_API_Q_DGPS_MODE

Queries the setting of DGPS mode. See **PMTK_API_SET_DGPS_MODE** for the setting of DGPS mode and **PMTK_DT_DGPS_MODE** for the query result.

Type:

Command

Synopsis:

```
$PMTK401*<Checksum><CR><LF>
```

Parameter:

None

Result:

Returns a **PMTK_DT_DGPS_MODE** message.

Example:

```
$PMTK401*37
$PMTK501,1*2B
```

2.3.31. PMTK413 PMTK_API_Q_SBAS_ENABLED

Queries the setting of SBAS. See **PMTK_API_SET_SBAS_ENABLED** for SBAS setting and **PMTK_DT_SBAS_ENABLED** for the query result.

Type:

Command

Synopsis:

```
$PMTK413*<Checksum><CR><LF>
```

Parameter:

None

Result:

Returns a **PMTK_DT_SBAS_ENABLED** message.

Example:

```
$PMTK413*34
$PMTK513,1*28
```

2.3.32. PMTK414 PMTK_API_Q_NMEA_OUTPUT

Queries the current NMEA sentence output frequency. See **PMTK_API_SET_NMEA_OUTPUT** for the frequency setting and **PMTK_DT_NMEA_OUTPUT** for the query result.

Type:

Command

Synopsis:

```
$PMTK414*<Checksum><CR><LF>
```

Parameter:

None

Result:

Returns a **PMTK_DT_NMEA_OUTPUT** message.

Example:

```
$PMTK414*33
$PMTK514,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0*33
```

2.3.33. PMTK501 PMTK_DT_DGPS_MODE

This message is the response to **PMTK_API_Q_DGPS_MODE**.

Type:

Output

Synopsis:

```
$PMTK501,<Mode>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Mode>	Numeric	-	DGPS data source mode. 0 = No DGPS source 1 = RTCM 2 = SBAS

Example:

```
$PMTK501,1*2B
```

2.3.34. PMTK513 PMTK_DT_SBAS_ENABLED

This message is the response to **PMTK_API_Q_SBAS_ENABLED**.

Type:

Output

Synopsis:

```
$PMTK513,<Enable>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Enable>	Numeric	-	Setting result. 0 = Disable 1 = Enable

Example:

```
$PMTK513,1*28
```

2.3.35. PMTK514 PMTK_DT_NMEA_OUTPUT

This message is the response to **PMTK_API_Q_NMEA_OUTPUT**.

Type:

Output

Synopsis:

```
$PMTK514,<GLL>,<RMC>,<VTG>,<GGA>,<GSA>,<GSV>,<GRS>,<GST>,<Res1>,<Res2>,<Res3>,<Res4>,<Res5>,<Res6>,<Res7>,<Res8>,<Res9>,<ZDA>,<Res10>,<DTM>,<GBS>,<Res11>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<GLL>	Numeric	-	**GLL** sentence output frequency. 0 = Disabled or not supported sentence n = Output once every "n" position fix. "n" ranges from 1 to 5
<RMC>	Numeric	-	**RMC** sentence output frequency. 0 = Disabled or not supported sentence n = Output once every "n" position fix. "n" ranges from 1 to 5.
<VTG>	Numeric	-	**VTG** sentence output frequency. 0 = Disabled or not supported sentence n = Output once every "n" position fix. "n" ranges from 1 to 5.
<GGA>	Numeric	-	**GGA** sentence output frequency. 0 = Disabled or not supported sentence n = Output once every "n" position fix. "n" ranges from 1 to 5.
<GSA>	Numeric	-	**GSA** sentence output frequency. 0 = Disabled or not supported sentence n = Output once every "n" position fix. "n" ranges from 1 to 5.
<GSV>	Numeric	-	**GSV** sentence output frequency. 0 = Disabled or not supported sentence n = Output once every "n" position fix. "n" ranges from 1 to 5.
<GRS>	Numeric	-	**GRS** sentence output frequency. 0 = Disabled or not supported sentence n = Output once every "n" position fix. "n" ranges from 1 to 5.
<GST>	Numeric	-	**GST** sentence output frequency. 0 = Disabled or not supported sentence

Field	Format	Unit	Description
			n = Output once every "n" position fix. "n" ranges from 1 to 5.
<Res1>	Numeric	-	Always "0".
<Res2>	Numeric	-	Always "0".
<Res3>	Numeric	-	Always "0".
<Res4>	Numeric	-	Always "0".
<Res5>	Numeric	-	Always "0".
<Res6>	Numeric	-	Always "0".
<Res7>	Numeric	-	Always "0".
<Res8>	Numeric	-	Always "0".
<Res9>	Numeric	-	Always "0".
<ZDA>	Numeric	-	**ZDA** sentence output frequency. 0 = Disabled or not supported sentence n = Output once every "n" position fix. "n" ranges from 1 to 5.
<Res10>	Numeric	-	Always "0".
<DTM>	Numeric	-	**DTM** sentence output frequency. 0 = Disabled or not supported sentence n = Output once every "n" position fix. "n" ranges from 1 to 5.
<GBS>	Numeric	-	**GBS** sentence output frequency. 0 = Disabled or not supported sentence n = Output once every "n" position fix. "n" ranges from 1 to 5
<Res11>	Numeric	-	Always "0".

Example:

```
$PMTK514,1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0*32
```

2.3.36. PMTK605 PMTK_Q_RELEASE

Queries the firmware release information. See **PMTK_DT_RELEASE** for the query result.

Type:

Command

Synopsis:

```
$PMTK605*<Checksum><CR><LF>
```

Parameter:

None

Result:

Returns a **PMTK_DT_RELEASE** message.

Example:

```
$PMTK605*31
$PMTK705,AG3331_AXN5.1.9_MODULE_STD_F8_P2,000B,Quectel-LC86L,1.0*26
```

2.3.37. PMTK622 PMTK_Q_LOCUS_DATA

Dumps LOCUS flash data.

Type:

Input

Synopsis:

```
$PMTK622,<Type>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Type>	Numeric	-	Type of dumping. 0 = Dump all the LOCUS data in the flash 1 = Dump the LOCUS data in the current flash sector

Result:

Returns a **PMTK_ACK** message.

Example:

```
$PMTK622,1*29
$PMTK001,622,3*36
```

NOTE

When the UTC time is not a non-integer second, the UTC time stored in LOCUS may differ from the actual UTC time by ±1 s.

2.3.38. PMTK705 PMTK_DT_RELEASE

This message is the response to **PMTK_Q_RELEASE**.

Type:

Output

Synopsis:

```
$PMTK705,<Rel_String>,<Build_ID>,<Pro_Model>,<SDK_Version>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Rel_String>	Numeric	-	Firmware release version and name.
<Build_ID>	Numeric	-	Build ID for firmware version control.
<Pro_Model>	Numeric	-	Product model for product identification.
<SDK_Version>	Numeric	-	Showing SDK version if the firmware is used for SDK.

Example:

```
$PMTK705,AG3331_AXN5.1.9_MODULE_STD_F8_P2,000B,Quectel-LC86L,1.0*26
```

2.3.39. PMTK838 PMTK_TEST_ANTI_SPOOFING

Enables or disables jamming detection function.

Type:

Set

Synopsis:

```
$PMTK838,<CmdType>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<CmdType>	Numeric	-	Enable or disable jamming detection function. 0 = Disable 1 = Enable

Result:

Returns a **PMTK_ACK** message and the query result.

Query Result Message Format:

```
$PMTKSPF,<Status>*<Checksum><CR><LF>
```

Parameter included in the result above is listed as below:

Field	Format	Unit	Description
<Status>	Numeric	-	Status of jamming detection function. 1 = No jamming, healthy status 2 = Warning status 3 = Critical status

Example:

```
$PMTK838,1*2C
$PMTK001,838,3,1*2E
//Healthy status:
$PMTKSPF,1*5A
//Warning status:
$PMTKSPF,2*59
//Critical status:
$PMTKSPF,3*58
```

NOTE

After jamming detection is enabled, the modules start to detect whether there is any jamming.

1. If there is no jamming, **$PMTKSPF,1*5A** will be reported to indicate healthy status (status 1).
2. If there is continuous jamming, the module status will change from 1 to 2 and finally 3.
    1) In the case of not being positioned: after jamming detection is enabled, the module status will be 1 at the very beginning, and then change to 2 when jamming is detected. During the process, the modules will attempt to fix position. If position fix still fails after 200 s, the module status will change to 3 finally.
    2) In the case of being positioned: after jamming detection is enabled, the module status will be 1 at the very beginning. When jamming is detected, the module status will change to 2 and then 3 consecutively.
3. This command is only supported on L70 and L80.

2.3.40. PMTK869 PMTK_EASY_ENABLE

Enables or disables EASY function, and it can also get whether EASY is enabled or disabled.

Type:

Set/Get/Output

Synopsis:

```
$PMTK869,<Type>[,<Enable_CDM>][,<Extension_Day>]*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Type>	Numeric	-	Message type. 0 = Query 1 = Set 2 = Query result
<Enable_CMD>	Numeric	-	Enable or disable EASY function. 0 = Disable 1 = Enable
<Extension_Day>	Numeric	-	Finished extension days. Range:0–3.

Result:

1) Set:

- Returns a **PMTK_ACK** message.

2) Get:

- Outputs a **PMTK_EASY_ENABLE** message.

Example:

```
//Set:
$PMTK869,1,1*35
$PMTK001,869,3*37

//Get:
$PMTK869,0*29
$PMTK869,2,0,0*2B
```

NOTE

1. If EASY is disabled, the modules return: **$PMTK869,2,0,0*2B**.
2. If EASY is enabled and is not finished yet, the modules may return: **$PMTK869,2,1,0*2A**.
3. If EASY is enabled and is finished after 1 day, the modules may return: **$PMTK869,2,1,1*2B**.
4. If EASY is enabled and is finished after 2 days, the modules may return: **$PMTK869,2,1,2*28**.
5. If EASY is enabled and is finished after 3 days, the modules may return: **$PMTK869,2,1,3*29**.

2.3.41. PMTK875 PMTK_PMTKLSC_STN_OUTPUT

PMTKLSC message is the leap second indication statement. This command enables or disables **PMTKLSC** message output, and it can also query whether **PMTKLSC** message output is enabled or disabled.

Type:

Set/Get/Output

Synopsis:

```
$PMTK875,<Type>[,<Enabled>]*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Type>	Numeric	-	Message type. 0 = Query 1 = Set 2 = Query result

Field	Format	Unit	Description
<Enabled>	Numeric	-	Enable or disable **PMTKLSC** message output. 0 = Disable 1 = Enable

Result:

1) Set command:

- Returns a **PMTK_ACK** message and query result messages.

2) Get command:

- Outputs a **PMTK_PMTKLSC_STN_OUTPUT** message and query result messages.

Query Result Message Format:

```
$PMTKLSC,<Current_Leap_Sec>,<Leap_Indicator>,<Next_Leap_Sec>*<Checksum><CR><LF>
$PMTKLSCB,<Current_Leap_Sec>,<Leap_Indicator>,<Next_Leap_Sec>*<Checksum><CR><LF>
```

The parameters included in the result above is listed as below:

Packet Data	Format	Unit	Description
<Current_Leap_Sec>	Numeric	Second	Current leap second.
<Leap_Indicator>	Numeric	-	Leap indicator. 1 = Updated from broadcast data
<Next_Leap_Sec>	Numeric	Second	Next leap second.

Example:

```
//Set:
$PMTK875,1,1*38
$PMTK001,875,3*3A
$PMTKLSC,18,1,18*43
$PMTKLSCB,0,0,0*00

//Get:
$PMTK875,0*24
$PMTK875,2,1*3B
$PMTKLSC,18,1,18*43
$PMTKLSCB,0,0,0*00
```

2.3.42. PMTK886 PMTK_FR_MODE

Sets the navigation mode.

Type:

Set

Synopsis:

```
$PMTK886,<NavMode>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<NavMode>	Numeric	-	The navigation mode. 0 = Normal Mode. For general purposes 1 = Fitness Mode. For running and walking purposes that the low-speed (< 5 m/s) movement will have more effect on the position calculation 2 = Aviation Mode. For high-dynamic purposes that the large-acceleration movement will have more effect on the position calculation 3 = Balloon Mode. For high-altitude balloon purposes that the vertical movement will have more effect on the position calculation 4 = Stationary Mode. For stationary applications that zero dynamics is assumed

Result:

Returns **PMTK_ACK** message.

Example:

```
$PMTK886,3*2B
$PMTK001,886,3*36
```

NOTE

Each mode has its altitude limitation. Please choose an appropriate mode based on the altitude limitations listed below, otherwise the position calculation will be incorrect.

Mode	Unit	Altitude Limitation
Normal Mode	Meter	10000
Fitness Mode	Meter	10000
Aviation Mode	Meter	10000
Stationary Mode	Meter	10000
Balloon Mode	Meter	80000

2.3.43. PMTKSPF

Outputs Jamming status when jamming detection function is enabled.

Type:

Output

Synopsis:

```
$PMTKSPF,<Status>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Status>	Numeric	-	Jamming status. 0 = No jamming, healthy status 1 = Warning status 2 = Critical status

Example:

```
$PMTKSPF,1*5A
```

NOTE

When the Jamming detection function is enabled on L96 module by **$PQJAM** command, the pin 20 (JAM_IND) is at a high level when the Jamming status is 0 or 1; the pin 20 (JAM_IND) is at a low level when the Jamming status is 2. For details on **PQJAM** command, see *Chapter 2.4.12 PQJAM* for details.

2.4. PQ Messages

This chapter explains the PQ messages which are defined and developed by Quectel. The configuration parameters saved by PQ messages will still exist in flash even after upgrading the modules' firmware.

2.4.1. PQBAUD

Sets NMEA port default baud rate.

Type:

Set

Synopsis:

```
$PQBAUD,W,<Baudrate>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Baudrate>	Numeric	bps	NMEA port baud rate: 4800 9600 14400 19200 38400 57600 115200

Result:

- In case of no errors:

```
$PQBAUD,W,OK*<Checksum><CR><LF>
```

- In case of any error:

```
$PQBAUD,W,ERROR*<Checksum><CR><LF>
```

Example:

```
//Set NMEA port default baud rate to 115200 bps:
$PQBAUD,W,115200*43
//Set successfully:
$PQBAUD,W,OK*40
```

NOTE

1. The command takes effect immediately after setting.
2. The parameter is automatically saved.
3. There is no response returned if the baud rate is changed to a different value.

2.4.2. PQEPE

Enables/disables **PQEPE** message output. If enabled, **PQEPE** message will be automatically output.

Type:

Set/Output

Synopsis:

```
//Set Command:
$PQEPE,W,<Mode>,<Save>*<Checksum><CR><LF>
//Output message:
$PQEPE,<EPE_Hori>,<EPE_Vert>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Mode>	Numeric	-	Enable or disable **PQEPE** message output. 0 = Disable 1 = Enable
<Save>	Numeric	-	Saving operation. 0 = Parameter is not saved, and is invalid after restart 1 = Parameter is saved in flash, and is valid after restart
<EPE_Hori>	Numeric	Meter	Estimated horizontal position error.

Field	Format	Unit	Description
<EPE_Vert>	Numeric	Meter	Estimated vertical position error.

Result:

- In case of no errors:

```
$PQEPE,W,OK*<Checksum><CR><LF>
```

- In case of any error:

```
$PQEPE,W,ERROR*<Checksum><CR><LF>
```

Example:

```
//Enable PQEPE message output and save parameters into flash:
$PQEPE,W,1,1*2A
//Set successfully:
$PQEPE,W,OK*02

//Output message:
$PQEPE,5.3050,3.2000*53
```

NOTE

The command takes effect immediately after setting.

2.4.3. PQ1PPS

Sets the type and pulse width of 1PPS output.

Type:

Set

Synopsis:

```
$PQ1PPS,W,<Type>,<Width>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Type>	Numeric	-	The type of 1PPS output. 0 = Disable 1PPS output 1 = Send 1PPS after the first fix 2 = Send 1PPS after 3D fix 3 = Send 1PPS after 2D fix 4 = Send 1PPS always
<Width>	Numeric	Millisecond	PPS pulse width. Range: 2–998.

Result:

- In case of no errors:

```
$PQ1PPS,W,OK*<Checksum><CR><LF>
```

- In case of any error:

```
$PQ1PPS,W,ERROR*<Checksum><CR><LF>
```

Example:

```
//Set the type of 1PPS output and PPS pulse width:
$PQ1PPS,W,4,100*1D
//Set successfully:
$PQ1PPS,W,OK*30
//Disable 1PPS output:
$PQ1PPS,W,0,0*18
//Set successfully:
$PQ1PPS,W,OK*30
```

NOTE

1. The command takes effect immediately after setting.
2. The parameters are automatically saved.
3. If **<Type>** is set as 0, **<Width>** could be set as 0–998.

2.4.4. PQFLP

Sets the modules into FLP mode and gets module operation mode.

Type:

Set/Get

Synopsis:

```
$PQFLP,<Type>,<Mode>,<Save>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description

<Type>	String	-	Command type.
			W = Set
			R = Get
			When <Type> is **R,** other parameters can be omitted.

<Mode>	Numeric	-	Module operation mode. 0 = Normal mode 1 = FLP mode
<Save>	Numeric	-	Saving operation. 0 = Parameter is not saved, and is invalid after restart 1 = Parameter is saved in flash, and is valid after restart

Result:

1) Set:

- In case of no errors:

```
$PQFLP,W,OK*<Checksum><CR><LF>
```

- In case of any error:

```
$PQFLP,W,ERROR*<Checksum><CR><LF>
```

2) Get:

```
$PQFLP,R,<Mode>*<Checksum><CR><LF>
```

Example:

```
//Change to FLP mode:
$PQFLP,W,1,1*20
//Set successfully:
```

```
$PQFLP,W,OK*08

//Get the mode:
$PQFLP,R*25
//Get successfully, FLP mode is enabled:
$PQFLP,R,1*38
```

NOTE

1. The command takes effect immediately after setting.
2. This command is supported on L70 and L80 modules only.

2.4.5. PQTXT

Enables or disables **GPTXT** message output.

Type:

Set

Synopsis:

```
$PQTXT,W,<Mode>,<Save>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Mode>	Numeric	-	Enable or disable **GPTXT** message output. 0 = Disable 1 = Enable
<Save>	Numeric	-	Saving operation. 0 = Parameter is not saved, and is invalid after restart 1 = Parameter is saved in flash, and is valid after restart

Result:

- In case of no errors:

```
$PQTXT,W,OK*<Checksum><CR><LF>
```

- In case of any error:

```
$PQTXT,W,ERROR*<Checksum><CR><LF>
```

Example:

```
//Enable GPTXT message output and save the parameter into flash:
$PQTXT,W,1,1*22
//Set successfully:
$PQTXT,W,OK*0A
```

NOTE

1. The command takes effect immediately after setting.
2. This command is only supported on the modules which support antenna detection feature.

2.4.6. PQECEF

Enables or disables **ECEFPOSVEL** message output. If enabled, **ECEFPOSVEL** message is automatically output. This command can also get module operation mode.

Type:

Set/Get

Synopsis:

```
$PQECEF,<Type>,<Mode>,<Save>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description

<Type>	String	-	Command type.
			W = Set
			R = Get
			When **<Type>** is **R,** other parameters can be omitted.

<Mode>	Numeric	-	Enable or Disable **ECEFPOSVEL** message output. 0 = Disable 1 = Enable
<Save>	Numeric	-	Saving operation. 0 = Parameter is not saved, and is invalid after restart 1 = Parameter is saved in flash, and is valid after restart

Result:

1) Set:

- In case of no errors:

```
$PQECEF,W,OK*<Checksum><CR><LF>
```

- In case of any error:

```
$PQECEF,W,ERROR*<Checksum><CR><LF>
```

2) Get:

```
$PQECEF,R,<Mode>*<Checksum><CR><LF>
```

Example:

```
//Enable ECEFPOSVEL message output, and save the parameter into flash:
$PQECEF,W,1,1*7F
//Set successfully:
$PQECEF,W,OK*57

//Get the mode:
$PQECEF,R*7A
//Get successfully, ECEFPOSVEL message output is enabled:
$PQECEF,R,1*67
```

NOTE

The command takes effect immediately after setting.

2.4.7. ECEFPOSVEL

This message is automatically output when **PQECEF** is enabled.

Type:

Output

Synopsis:

```
$ECEFPOSVEL,<Time>,<X>,<Y>,<Z>,<V_X>,<V_Y>,<V_Z>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Time>	Numeric	-	UTC from the internal real-time clock.
<X>	Numeric	-	The value of X axis in **ECEFPOSVEL** message.
<Y>	Numeric	-	The value of Y axis in **ECEFPOSVEL** message.
<Z>	Numeric	-	The value of Z axis in **ECEFPOSVEL** message.
<V_X>	Numeric	-	Velocity component of X axis in **ECEFPOSVEL** message.
<V_Y>	Numeric	-	Velocity component of Y axis in **ECEFPOSVEL** message.
<V_Z>	Numeric	-	Velocity component of Z axis in **ECEFPOSVEL** message.

Example:

```
$ECEFPOSVEL,052743.000,-1526672.867459,6191083.982801,143008.780911,0,0,0*14
```

2.4.8. PQODO

Starts or stops odometer reading. This command can also get whether the odometer reading is started and query the distance value.

Type:

Set/Get/Query

Synopsis:

```
$PQODO,<Type>,<Mode>,<Initial_Distance>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description

<Type>	String	-	Command type.
			W = Set
			R = Get
			Q = Query
			When <Type> is **R** or **Q,** other parameters can be omitted.

<Mode>	Numeric	-	Start or stop odometer reading. 0 = Stop odometer reading and remember the distance value 1 = Start odometer reading and initialize the distance

Field	Format	Unit	Description
			according to the **<Initial_Distance>**
<Initial_Distance>	Numeric	Meter	Set the initial distance. Range: 0-1e09. When **<Mode>** is 1, this parameter can be omitted, and its default value is 0. When **<Mode>** is 0, this parameter must be omitted.

Result:

1) Set:

- In case of no errors:

```
$PQODO,W,OK*<Checksum><CR><LF>
```

- In case of any error:

```
$PQODO,W,ERROR*<Checksum><CR><LF>
```

2) Get:

- In case of no errors:

```
$PQODO,R,<Mode>*<Checksum><CR><LF>
```

- In case of any error:

```
$PQODO,R,ERROR*<Checksum><CR><LF>
```

3) Query:

- In case of no errors:

```
$PQODO,Q,<Distance>*<Checksum><CR><LF>
```

- In case of any error:

```
$PQODO,Q,ERROR*<Checksum><CR><LF>
```

Parameter included in the query result:

Field	Format	Unit	Description
<Distance>	Numeric	Meter	Current distance.

Example:

```
//Start odometer reading, and initial distance is 0 m:
$PQODO,W,1*23
//Set successfully:
$PQODO,W,OK*16
//Start odometer reading, and initial distance is 1,000,000 m:
$PQODO,W,1,1000000*3E
//Set successfully:
$PQODO,W,OK*16

//Get the mode:
$PQODO,R*3B
//Get successfully, odometer reading has already been started:
$PQODO,R,1*26

//Query the distance value:
$PQODO,Q*38
//Current distance value is returned:
$PQODO,Q,123.45*0B
```

NOTE

1. The command takes effect immediately after setting.
2. After module is restarted, the **PQODO** Set Command must be executed again to re-start odometer reading.
3. The command is not supported in Backup mode.

2.4.9. PQPZ90

Enables or disables switching from WGS84 to PZ-90.11. This command can also get whether switching from WGS84 to PZ-90.11 is enabled.

Type:

Set/Get

Synopsis:

```
$PQPZ90,<Type>,<Mode>,<Save>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description

<Type>	String	-	Command type.
			W = Set
			R = Get
			When <Type> is **R,** other parameters can be omitted.

<Mode>	Numeric	-	Enable or Disable switching from WGS84 to PZ-90.11. 0 = Disable 1 = Enable
<Save>	Numeric	-	Saving operation. 0 = Parameter is not saved, and is invalid after restart 1 = Parameter is saved in flash, and is valid after restart

Result:

1) Set:

- In case of no errors:

```
$PQPZ90,W,OK*<Checksum><CR><LF>
```

- In case of any error:

```
$PQPZ90,W,ERROR*<Checksum><CR><LF>
```

2) Get:

```
$PQPZ90,R,<Mode>*<Checksum><CR><LF>
```

Example:

```
//Enable switching from WGS84 to PZ-90.11, and save the parameter into flash:
$PQPZ90,W,1,1*79
//Set successfully:
$PQPZ90,W,OK*51

//Get the mode:
$PQPZ90,R*7C
//Get successfully, switching from WGS84 to PZ-90.11 is enabled:
$PQPZ90,R,0*60
```

NOTE

1. The command takes effect immediately after setting **<Save>** to 0. However, when **<Save>** is set to 1, it takes effect only after restart for versions earlier than L76NR03A01S (on L76 module), L76LNR02A01S (on L76-L module), L86NR02A01S (on L86 module) and L26NR02A01S (on L26 module).
2. If switching from WGS84 to PZ-90.11 is enabled and takes effect, the coordinate values in **RMC** and **GGA** sentences will be switched to PZ-90.11 after fixing. Also, a **DTM** sentence will be displayed to identify the datum used.

2.4.10. PQGLP

Sets the modules to GLP mode and gets whether the GLP mode is enabled.

Type:

Set/Get

Synopsis:

```
$PQGLP,<Type>,<Mode>,<Save>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description

<Type>	String	-	Command type.
			W = Set
			R = Get
			When **<Type>** is **R,** other parameters can be omitted.

<Mode>	Numeric	-	Module operation mode. 0 = Normal mode 1 = GLP mode
<Save>	Numeric	-	Saving operation. 0 = Parameter is not saved, and is invalid after restart 1 = Parameter is saved in flash, and is valid after restart

Result:

1) Set:

- In case of no errors:

```
$PQGLP,W,OK*<Checksum><CR><LF>
```

- In case of any error:

```
$PQGLP,W,ERROR*<Checksum><CR><LF>
```

2) Get:

```
$PQGLP,R,<Mode>*<Checksum><CR><LF>
```

Example:

```
//Change to GLP mode:
$PQGLP,W,1,1*21
//Set successfully:
$PQGLP,W,OK*09

//Get the mode:
$PQGLP,R*24
//Get successfully. GLP mode is enabled:
$PQGLP,R,1*39
```

NOTE

1. The command takes effect immediately after setting.
2. This command is not supported on L70 and L80.

2.4.11. PQVEL

Enables or disables **PQVEL** message output. If enabled, **PQVEL** message will be automatically output. This command can also get whether **PQVEL** message output is enabled.

Type:

Set/Get/Output

Synopsis:

```
//Set/Get command:
$PQVEL,<Type>,<Mode>,<Save>*<Checksum><CR><LF>
//Output message:
$PQVEL,<North_Vel>,<East_Vel>,<Down_Vel>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description

<Type>	String	-	Command type.
			W = Set
			R = Get
			When **<Type>** is **R,** other parameters can be omitted.

<Mode>	Numeric	-	Enable or disable **PQVEL** message output. 0 = Disable 1 = Enable
<Save>	Numeric	-	Saving operation. 0 = Parameter is not saved, and is invalid after restart 1 = Parameter is saved in flash, and is valid after restart
<North_Vel>	Numeric	m/s	North velocity.
<East_Vel>	Numeric	m/s	East velocity.
<Down_Vel>	Numeric	m/s	Down velocity.

Result:

1) Set:

- In case of no errors:

```
$PQVEL,W,OK*<Checksum><CR><LF>
```

- In case of any error:

```
$PQVEL,W,ERROR*<Checksum><CR><LF>
```

2) Get:

```
$PQVEL,R,<Mode>*<Checksum><CR><LF>
```

Example:

```
//Set message output:
$PQVEL,W,1,1*25
//Set successfully:
$PQVEL,W,OK*0D

//Get the mode:
$PQVEL,R*20
//The mode is enabled:
```

```
$PQVEL,R,1*3D

//Output message:
$PQVEL,1.000000,2.000000,-0.000000*42
```

NOTE

The command takes effect immediately after setting.

2.4.12. PQJAM

Enables or disables jamming detection function. This command can also get whether jamming detection function is enabled.

Type:

Set/Get

Synopsis:

```
$PQJAM,<Type>,<Mode>,<Save>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description

<Type>	String	-	Command type.
			W = Set
			R = Get
			When **<Type>** is **R,** other parameters can be omitted.

<Mode>	Numeric	-	Enable or disable jamming detection function. 0 = Disable 1 = Enable
<Save>	Numeric	-	Saving operation. 0 = Parameter is not saved, and will is invalid restart 1 = Parameter is saved in flash, and is valid after restart

Result:

1) Set:

- In case of no errors:

```
$PQJAM,W,OK*<Checksum><CR><LF>
```

- In case of any error:

```
$PQJAM,W,ERROR*<Checksum><CR><LF>
```

2) Get:

```
$PQJAM,R,<Mode>*<Checksum><CR><LF>
```

Example:

```
//Enable jamming detection function:
$PQJAM,W,1,1*3C
//Set successfully:
$PQJAM,W,OK*14

//Get operation mode:
$PQJAM,R*39
//Get successfully:
$PQJAM,R,1*24
```

NOTE

1. The command takes effect immediately after setting.
2. The command is not supported in Backup mode.

2.4.13. PQRLM

Enables/disables the return link message output. This command can also receive distress signal.

Type:

Set/Get/Output

Synopsis:

```
//Set/Get Command:
$PQRLM,<Type>,<Mode>,<Save>*<Checksum><CR><LF>
//Output message:
$PQRLM,<BeaconID>,<Gps_Sec>,<Msg_Code>,<Para>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description

<Type>	String	-	Command type.
			W = Set
			R = Get
			When **<Type>** is **R**, other parameters can be omitted.

<Mode>	Numeric	-	Enable or disable return link message output. 0 = Disable 1 = Enable
<Save>	Numeric	-	Saving operation. 0 = Parameter is not saved, and is invalid after restart 1 = Parameter is saved in flash, and is valid after restart
<BeaconID>	Numeric	-	Beacon ID of RLM.
<Gps_Sec>	Numeric	Second	The GPS seconds when receiving RLM. Note: The output format is hhmmss.ss for L76L module. hh: Hours (00–23) mm: Minutes (00–59) ss: Seconds (00–59) ss: Decimal fraction of seconds
<Msg_Code>	Numeric	-	Message code.
<Para>	Numeric	-	The data parameters provided by RLS. Short message contains 4 hex characters and long message contains 24 hex characters.

Result:

1) Set:

- In case of no errors:

```
$PQRLM,W,OK*<Checksum><CR><LF>
```

- In case of any error:

```
$PQRLM,W,ERROR*<Checksum><CR><LF>
```

2) Get:

```
$PQRLM,R,<Mode>*<Checksum><CR><LF>
```

Example:

```
//Enable return link message output:
$PQRLM,W,1,1*29
//Set successfully:
$PQRLM,W,OK*01

//Get the mode:
$PQRLM,R*2C
//Get successfully:
$PQRLM,R,1*31
//Output message:
$PQRLM,a042be29630f190,1370324891,15,6503*66

//Output message for L76L module:
$PQRLM,9c02be29630f0a0,054131.00,15,6503*1C
```

NOTE

1. The command takes effect immediately after setting.
2. This command is not supported on the following modules:
   - L70
   - L80
   - L26-LB
   - L76-LB
   - LC86L (A, B)
3. The **$PQRLM** can output up to 10 messages. If there are more than 10 output messages, the new one will overwrite the old one.

2.4.14. PQGEO

Configures parameters of Geo-fence. This command can also get Geo-fence setting and query the Geo-fence ID status of current position.

Type:

Set/Get/Query/Output

Synopsis:

```
//Set/Get/Query Command:
$PQGEO,<Type>,<GEO_ID>,<Mode>,<Shape>,<Lat0>,<Lon0>,<Lat1/Radius>,<Lon1>,<Lat2>,<Lon2>,
<Lat3>,<Lon3>*<Checksum><CR><LF>
//Output message:
$PQGEO,<GEO_ID>,<Action>,<FixStatus>,<UTC&Time>,<Lat>,<Lon>,<MSL_Alt>,<SOG>,<COG>,<Fix
Mode>,<Res1>,<HDOP>,<PDOP>,<VDOP>,<Res2>,<GPS_SatUsed>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description

<Type>	String	-	Command type.
			W = Set
			R = Get
			Q = Query
			When **<Type>** is **R** or **Q**, other parameters can be omitted except **<GEO_ID>**.

<GEO_ID>	Numeric	-	Geo-fence ID. Range: 0–9.
<Mode>	Numeric	-	URC Report mode. 0 = Disable URC to be reported when entering or leave the Geo-fence 1 = Enable URC to be reported when entering the Geo-fence 2 = Enable URC to be reported when leaving the Geo-fence 3 = Enable URC to be reported when entering or leaving the Geo-fence If **<Mode>** is 0, the parameters after **<Mode>** can be omitted.
<Shape>	Numeric	-	Fence shape. 0 = Circularity with center and radius 1 = Circularity with center and one point on the circle 2 = Triangle 3 = Quadrangle
<Lat0>	±dd.dddddd	Degree	The latitude of a point which is defined as the center of the Geo-fence circular region or the first point. Range: -90.000000–90.000000 (variable length, 4 to 6 digits).
<Lon0>	±ddd.dddddd	Degree	The longitude of a point which is defined as the center of the Geo-fence circular region or the first point. Range: -180.000000–180.000000 (variable length, 4

Field	Format	Unit	Description
			to 6 digits).
<Lat1/Radius>	±dd.dddddd	Meter/ Degree	When **<Shape>** is 0, this parameter is radius. Range: 0-6000000.0. When **<Shape>** is other values, this parameter is latitude1. Range: -90.000000–90.000000 (variable length, 4 to 6 digits). If **<Shape>** is 0, the parameters after **<Lat1/radius>** must be omitted.
<Lon1>	±ddd.dddddd	Degree	The longitude of the second point. Range: -180.000000–180.000000 (variable length, 4 to 6 digits). If **<Shape>** is 1, the parameters after **<Lon1>** must be omitted.
<Lat2>	±dd.dddddd	Degree	The latitude of the third point. Range: -90.000000–90.000000 (variable length, 4 to 6 digits).
<Lon2>	±ddd.dddddd,	Degree	The longitude of the third point. Range: -180.000000–180.000000 (variable length, 4 to 6 digits). If **<Shape>** is 2, the parameters after **<Lon2>** must be omitted.
<Lat3>	±dd.dddddd	Degree	The latitude of fourth point. Range: -90.000000–90.000000 (variable length, 4 to 6 digits).
<Lon3>	±ddd.dddddd	Degree	The longitude of fourth point. Range: -180.000000–180.000000 (variable length, 4 to 6 digits).
<Status>	Numeric	-	The status of current position. 0 = Unknown position 1 = Inside the Geo-fence 2 = Outside the Geo-fence
<GEO_ID>	Numeric	-	Geo-fence ID. Range: 0–9.
<Action>	Numeric	-	The current action of the module. 1 = Entering the Geo-fence 2 = Leaving the Geo-fence
<FixStatus>	Numeric	-	Fix status. 0 = No fix 1 = 2D fix 2 = 3D fix

Field	Format	Unit	Description
<UTC&Time>	DyyMMddThh mmss.sss	-	UTC time. D: char 'D', refers to date yy: current year–2000 MM: 1–12 dd: 1–31 T: char 'T', refers to time hh: 0–23 mm: 0–59 ss.sss: 00.000–59.999
<Lat>	±dd.dddddd	Degree	The latitude of current position. Range: -90.000000–90.000000.
<Lon>	±ddd.dddddd	Degree	The longitude of current position. Range: -180.000000–180.000000.
<MSL_Alt>	Numeric	Meter	Mean sea level (MSL) altitude.
<SOG>	Numeric	km/h	Speed over ground.
<COG>	Numeric	Degree	Course over ground. Range: 0–359.99
<FixMode>	Numeric	Degree	Fix mode. 0 = No fix 1 = Estimated mode 2 = Position fixed 3 = Position fixed in DGPS mode
<Res1>	Numeric	-	Reserved. Always null.
<HDOP>	Numeric	-	Horizontal dilution of precision.
<PDOP>	Numeric	-	Position dilution of precision.
<VDOP>	Numeric	-	Vertical dilution of precision.
<Res2>	Numeric	-	Reserved. Always null.
<GPS_SatUsed>	Numeric	-	GPS satellites used.

Result:

1) Set:

- In case of no errors:

```
$PQGEO,W,OK*<Checksum><CR><LF>
```

- In case of any error:

```
$PQGEO,W,ERROR*<Checksum><CR><LF>
```

2) Get:

- In case of no errors:

```
$PQGEO,R,<GEO_ID>,<Mode>,<Shape>,<Lat0>,<Lon0>,<Lat1/Radius>,<Lon1>,<Lat2>,<Lon2>,<Lat3
>,<Lon3>*<Checksum><CR><LF>
```

- In case of any error:

```
$PQGEO,R,ERROR*<Checksum><CR><LF>
```

3) Query:

- In case of no errors:

```
$PQGEO,Q,<GEO_ID>,<Status>*<Checksum><CR><LF>
```

- In case of any error:

```
$PQGEO,Q,ERROR*<Checksum><CR><LF>
```

Example:

```
//Enable the Gen-fence 0 to report when entering the circularity with center and radius:
$PQGEO,W,0,1,0,30.491225,114.501417,100.0*11
//Set successfully:
$PQGEO,W,OK*1F
//Disable the Gen-fence 4 to report when entering or leaving the circularity with center and one point on
the circle:
$PQGEO,W,4,3,1,30.491079,114.501307,30.494856,114.070281*21
//Set successfully:
$PQGEO,W,OK*1F
//Set the Geo-fence 3 not report when entering or leaving the Geo-fence:
$PQGEO,W,3,0*34
//Set successfully:
$PQGEO,W,OK*1F

//Get the Geo-fence 0 settings:
$PQGEO,R,0*2E
//Get successfully:
```

```
$PQGEO,R,0,1,0,30.491225,114.501417,100.0*14
//Get the Geo-fence 4 settings:
$PQGEO,R,4*2A
//Get successfully:
$PQGEO,R,4,3,1,30.491079,114.501307,30.494856,114.070281*24

//Query the Geo-fence 0 status of current position:
$PQGEO,Q,0*2D
//Query successfully:
$PQGEO,Q,0,1*30
//Query the Geo-fence 4 status of current position:
$PQGEO,Q,4*29
//Query successfully:
$PQGEO,Q,4,2*37

//Output message:
$PQGEO,0,1,2,D230608T032927.000,30.491130,114.501387,175.1,2.62,320.16,2,,0.78,1.06,0.71,,13*4E
```

NOTE

1. If **<Mode>** is 0 and no parameters follow **<Mode>**, this command can delete the Geo-fence.
2. Only L96 module can indicate whether it is inside or outside the Geofence via the voltage level change of pin 24 (GEO_FENCE). To do this, the **<GEO_ID>** must be set to 0. The truth table below shows the GEO_FENCE voltage level when the module is inside or outside the Geofence.

<Mode>	Inside the Geofence		Outside the Geofence	
	Voltage Level	Message	Voltage Level	Message
Mode = 0	High	None	High	None
Mode = 1	Low	Output	High	None
Mode = 2	High	None	Low	Output
Mode = 3	High	Output	High	Output

3. **$PQGEO,R,10*1F** can query parameters of all Geo-fences.
4. The command takes effect immediately after setting, and the parameters will be automatically saved into flash.
5. Input the latitude and longitude in sequence in clockwise or counter-clockwise order.

2.4.15. PQPREC

Configures the parameter (latitude/longitude/altitude) in NMEA sentences through setting the number of digits after the decimal point.

Type:

Set/Get

Synopsis:

```
$PQPREC,<Type>,<Lat_Bits>,<Lon_Bits>[,<Alt_Bits>],<Save>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description

<Type>	String	-	Command type.
			W = Set
			R = Get
			When **<Type>** is **R,** other parameters can be omitted.

<Lat_Bits>	Numeric	-	The number of digits after the decimal point of latitude in NMEA sentences. Range: 4–6. Default value: 4.
<Lon_Bits>	Numeric	-	The number of digits after the decimal point of longitude in NMEA sentences. Range: 4–6. Default value: 4.
<Alt_Bits>	Numeric	-	The number of digits after the decimal point of altitude and geoidal separation in NMEA sentences. This parameter can be omitted. Range: 1–3. Default value: 1.
<Save>	Numeric	-	Saving operation. 0 = Configuration is not saved, and is invalid after restart 1 = Configuration is saved in flash, and is valid after restart

Result:

1) Set:

- In case of no errors:

```
$PQPREC,W,OK*<Checksum><CR><LF>
```

- In case of any error:

```
$PQPREC,W,ERROR*<Checksum><CR><LF>
```

2) Get:

```
$PQPREC,R,<Lat_Bits>,<Lon_Bits>,<Alt_Bits>*<Checksum><CR><LF>
```

Example:

```
//Set the number of digits after the decimal point of latitude and longitude to 5, and altitude to 2, then save the parameter into flash:
$PQPREC,W,5,5,2,1*7D
//Set successfully:
$PQPREC,W,OK*56

//The following parameters in bold are variables:
$xxRMC,030037.000,A,3150.77801,N,11711.95112,E,0.00,37.74,070816,,,A*xx
$xxGGA,030037.000,3150.77801,N,11711.95112,E,1,6,1.66,96.54,M,0.01,M,,*xx
$xxGLL,3150.77801,N,11711.95112,E,030037.000,A,A*xx

//Get parameter precision setting in NMEA sentences:
$PQPREC,R*7B
//Get successfully:
$PQPREC,R,5,5,2*65
```

NOTE

1. The command takes effect immediately after setting.
2. This command is not supported on L70 and L80.

2.4.16. PQGBS

The command enables/disables the output of GBS sentence which is used to support receiver autonomous integrity monitoring (RAIM).

Type:

Set/Get

Synopsis:

```
$PQGBS,<Type>,<Mode>,<Save>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description

<Type>	String	-	Command type.
			W = Set
			R = Get

Field	Format	Unit	Description
			When **<Type>** is **R,** other parameters can be omitted.
<Mode>	Numeric	-	Enable or disable **GBS** sentence output. 0 = Disable 1 = Enable
<Save>	Numeric	-	Saving operation. 0 = Parameter is not saved, and is invalid after restart 1 = Parameter is saved in flash, and is valid after restart

Result:

1) Set:

- In case of no errors:

```
$PQGBS,W,OK*<Checksum><CR><LF>
```

- In case of any error:

```
$PQGBS,W,ERROR*<Checksum><CR><LF>
```

2) Get:

```
$PQGBS,R,<Mode>*<Checksum><CR><LF>
```

Example:

```
//Enable GBS sentence output and save into flash:
$PQGBS,W,1,1*2C
//Set successfully:
$PQGBS,W,OK*04

//Get the mode:
$PQGBS,R*29
//Get successfully. GBS sentence output is enabled:
$PQGBS,R,1*34
```

NOTE

1. The command takes effect immediately after setting.
2. This command is not supported on L70 and L80.

2.4.17. PQVERNO

Queries the information about firmware version.

Type:

Query/Output

Synopsis:

```
//Query Command:
$PQVERNO,R*<Checksum><CR><LF>
//Output message:
$PQVERNO,R,<Version>,<Date>,<Time>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Version>	String	-	Firmware version.
<Date>	yyyy/mm/dd	-	The date when the firmware version is generated. yyyy: Year mm: Month dd: Day of month
<Time>	hh:mm	-	The time when the firmware version is generated. hh: Hours (00–23) mm: Minutes (00–59)

Example:

```
//Query Command:
$PQVERNO,R*3F
//Output message:
$PQVERNO,R,LC86LIBNR11A01S,2023/02/09,09:20*63
```

3 Appendix A References

Table 5: Terms and Abbreviations

Abbreviation	Description
1PPS	1 Pulse Per Second
2D	2 Dimension
3D	3 Dimension
ACK	Acknowledgement
AGNSS	Assisted GNSS
AIC	Active Interference Cancellation
DEE	Dynamic Ephemeris Extension
DGPS	Differential Global Positioning System
DOP	Dilution of Precision
EASY	Embedded Assist System
ECEF	Earth-centered, Earth-fixed
EGNOS	European Geostationary Navigation Overlay Service
EPE	Estimated Position Error
EPO	Extended Prediction Orbit
FLP	Fitness Low Power
GAGAN	GPS-aided GEO Augmented Navigation
GBS	GNSS Satellite Fault Detection
GGA	Global Positioning System Fix Data
GLL	Geographic Position – Latitude/Longitude

Abbreviation	Description
GLONASS	Global Navigation Satellite System (The Russian GNSS)
GLP	GNSS Low Power
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
MSAS	Multi-functional Satellite Augmentation System
MSL	Mean Sea Level
NMEA	National Marine Electronics Association
NVM	Non-volatile Memory
ODO	Odometer
PDOP	Position Dilution of Precision
PMTK	Proprietary Protocol of MTK
PPS	Pulse Per Second
PZ90	Parametry Zemli 1990
QZSS	Quasi-Zenith Satellite System
RAIM	Receiver Autonomous Integrity Monitoring
RLM	Return Link Message
RMC	Recommended Minimum Specific GNSS Data
RTC	Real-time Clock
RTCM	Radio Technical Commission for Maritime Services
SBAS	Satellite-Based Augmentation System
SNR	Signal-to-noise Ratio

Abbreviation	Description
SV	Satellites in View
TXT	Text Transmission
URC	Unsolicited Result Code
UTC	Coordinated Universal Time
VDOP	Vertical Dilution of Precision
VTG	Course Over Ground and Ground Speed
WAAS	Wide Area Augmentation System
WGS84	World Geodetic System 1984

4 Appendix B GNSS Numbering

Table 6: GNSS Numbering

GNSS Type	System ID	Satellite ID	Signal ID
GPS	1	1–32 are reserved for GPS 33–64 are reserved for SBAS	1 = L1 C/A
GLONASS	2	65–99	1 = L1 C/A
Galileo	3	1–36	6 = L1
BDS	4	1–36	1 = B1I
QZSS	5	193–202	1 = L1 C/A

NOTE

1. The table above is only applicable to standard NMEA messages.
2. The GPS **Signal ID** of **Lx6** modules are 0.

5 Appendix C Special Characters

Table 7: 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.