L89 (HD)&LC29H (AI)
GNSS Protocol Specification

GNSS Module Series

Version: 1.1

Date: 2024-11-25

Status: Released

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

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

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

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

Legal Notices

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

Use and Disclosure Restrictions

License Agreements

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

Copyright

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

Trademarks

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

Third-Party Rights

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

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

Privacy Policy

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

Disclaimer

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

About the Document

Document Information	
Title	**L89 (HD)&LC29H (AI) GNSS Protocol Specification**
Subtitle	GNSS Module Series
Document Type	GNSS Protocol Specification
Document Status	Released

Revision History

Version	Date	Description
-	2023-10-11	Creation of the document
1.0	2024-05-13	First official release
1.1	2024-11-25	1. Updated the supported values of <ErrCode> (*Table 3: Error Codes*). 2. Added a note about the requirement for the PQTMCFGCNST command to take effect (*Chapter 2.3.19*). 3. Updated the default values of <ALT_DP> and <SPD_DP> in the PQTMCFGNMEADP command (*Chapter 2.3.22*). 4. Added the PQTMBKP command (*Chapter 2.3.36*).

Contents

Table Index

1 Introduction

Quectel L89 (HD) and LC29H (AI) GNSS modules support GPS, GLONASS, Galileo, BDS, QZSS and NavIC (IRNSS) constellations. Concurrent tracking of GPS L1 C/A, GLONASS L1, Galileo E1, BDS B1I, QZSS L1 C/A and NavIC L5 frequency bands provides fast and accurate acquisition and makes the modules the ideal positioning and navigation solution in various vertical markets.

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

NOTE

1. Quectel assumes no responsibility if commands other than the ones listed herein are used.
2. QZSS is enabled by default and it is always switchable.
3. For conciseness purposes, L89 (HD) and LC29H (AI) modules will hereinafter be referred to collectively as “the module/modules” in parts hereof applicable to all modules, and individually as “L89 (HD)” and “LC29H (AI)” in parts hereof referring to the differences between them.

2 NMEA Protocol

2.1. Structure of NMEA Protocol Messages

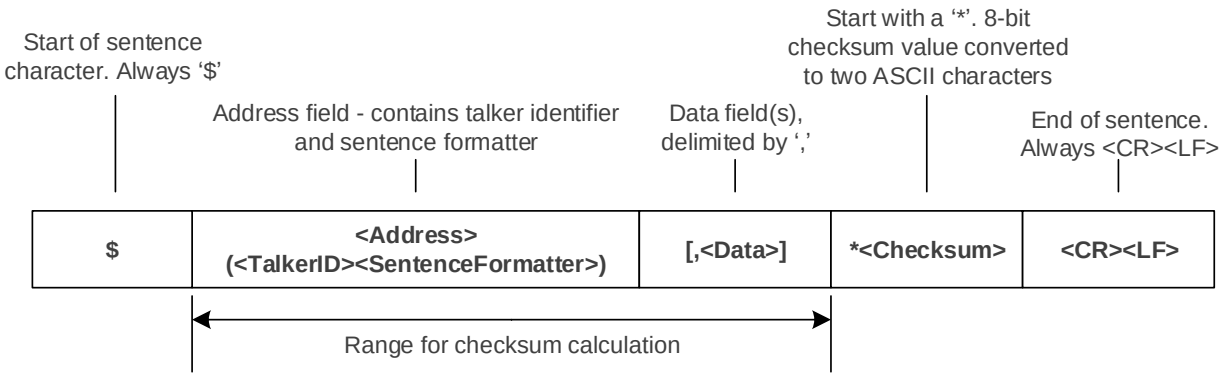

Figure 1: Structure of NMEA Protocol Messages

Table 1: 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 (Sentence Formatter). The talker identifier identifies the type of talker. For more information on the TalkerID, see *Table 2: 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 data field delimiter ‘,’. Variable length (depends 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 the sentence (Hex 0x0D 0x0A).

Table 2: NMEA Talker ID

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

Sample Code for NMEA Checksum:

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

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

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

```
    return result;
}
```

2.2. Standard Messages

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

2.2.1. RMC

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

Type:

Output

Synopsis:

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

Parameter:

Field	Format	Unit	Example	Description
$	Character	-	$	Each NMEA message starts with **$**.
<TalkerID>	String, 2 characters	-	GN	Talker identifier. See *Table 2: NMEA Talker ID*.
RMC	String, 3 characters	-	RMC	Recommended Minimum Specific GNSS Data.
<UTC>	hhmmss.sss	-	075721.000	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	-	3149.301729	Latitude. dd: Degrees (00–90) mm: Minutes (00–59)

Field	Format	Unit	Example	Description
				mmmmmm: Decimal fraction of minutes. Variable length. 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	-	11706.925505	Longitude. ddd: Degrees (000–180) mm: Minutes (00–59) mmmmmm: Decimal fraction of minutes. Variable length. 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	0.18	Speed over ground. Variable length. Note that this field is empty in case of an invalid value.
<COG>	Numeric	Degree	000.00	Course over ground. Variable length. Maximum value: 359.99. Note that this field is empty in case of an invalid value.
<Date>	ddmmyy	-	211023	Date. dd: Day of month mm: Month yy: Year
<MagVar>	-	-	-	Magnetic variation. Not supported. Always null.
<MagVarDir>	-	-	-	Direction of magnetic variation. Not supported. Always null.
<ModeInd>	Character	-	A	Mode indicator. A = Autonomous mode. Satellite system used in non-differential mode in position fix. D = Differential mode. Satellite

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

Example:

```
$GNRMC,075721.000,A,3149.301729,N,11706.925505,E,0.18,000.00,211023,,,A,V*08
```

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 2: NMEA Talker ID*.
GGA	String, 3 characters	-	GGA	Global Positioning System Fix Data.
<UTC>	hhmmss.sss	-	075721.000	Position fix UTC.

Field	Format	Unit	Example	Description
				hh: Hours (00–23) mm: Minutes (00–59) ss: Seconds (00–59) sss: Decimal fraction of seconds
<Lat>	ddmm.mmmmmm	-	3149.301729	Latitude. dd: Degrees (00–90) mm: Minutes (00–59) mmmmmm: Decimal fraction of minutes. Variable length. 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	-	11706.925505	Longitude. ddd: Degrees (000–180) mm: Minutes (00–59) mmmmmm: Decimal fraction of minutes. Variable length. Note that this field is empty in case of an invalid value.
<E/W>	Character	-	E	East-west direction. E = East W = West Note that this field is empty in case of an invalid value.
<Quality>	Numeric, 1 digit	-	1	GPS quality indicator. 0 = Fix not available or invalid. 1 = GPS SPS Mode, fix valid. 2 = Differential GPS, SPS Mode, or Satellite Based Augmentation System (SBAS), fix valid. 6 = Estimated (dead reckoning) mode.
<NumSatUsed> [1)]	Numeric, 2 digits	-	13	Number of satellites in use.
<HDOP>	Numeric	-	0.93	Horizontal dilution of precision. Variable length. Note that this field is 99.99 in case of an invalid value.

Field	Format	Unit	Example	Description
<Alt>	Numeric	Meter	116.671	Altitude above mean-sea-level (geoid). Variable length. Note that this field is empty in case of an invalid value.
M	Character	-	M	Unit of <Alt>. “M” = meter.
<Sep>	Numeric	Meter	-0.336	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. Always null.
<DiffStation>	-	-	-	Differential reference station ID. Not supported. Always null.
<Checksum>	Hexadecimal	-	*6F	Checksum.
<CR><LF>	Character	-	-	Carriage return and line feed.

Example:

```
$GNGGA,075721.000,3149.301729,N,11706.925505,E,1,13,0.93,116.671,M,-0.336,M,,*6F
```

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** messages will be generated from the multi-constellation solution.
2. [1)] According to the NMEA 0183 specification, the number of satellites in use is between 00 and 12. However, in the multi-constellation solution, the number of satellites in use may exceed 12.

2.2.3. GSV

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

Type:

Output

Synopsis:

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

Parameter:

Field	Format	Unit	Example	Description
$	Character	-	$	Each NMEA message starts with **$**.
<TalkerID>	String, 2 characters	-	GP	Talker identifier. See *Table 2: NMEA Talker ID*.
GSV	String, 3 characters	-	GSV	GNSS Satellites in View
<TotalNumSen>	Numeric	-	2	Total number of sentences. Range: 1–9.
<SenNum>	Numeric	-	1	Sentence number. Range: 1–**<TotalNumSen>**.
<TotalNumSat>	Numeric	-	07	Total number of satellites in view.
Start of repeat block. Repeat times: 0–4.				
<SatID>	Numeric	-	32	Satellite ID. See *Table 6: GNSS Satellites (NMEA) Numbering*.
<SatElev>	Numeric	Degree	68	Satellite elevation. Range: 00–90. Note that this field is empty in case of an invalid value.
<SatAz>	Numeric	Degree	095	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	37	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 Satellites (NMEA) Numbering*.
<Checksum>	Hexadecimal	-	*6F	Checksum.

Field	Format	Unit	Example	Description
<CR><LF>	Character	-	-	Carriage return and line feed.

Example:

```
$GPGSV,2,1,07,32,68,095,37,28,61,349,33,31,51,300,34,26,46,216,28,1*6F
$GPGSV,2,2,07,25,33,044,31,29,23,099,25,10,21,180,26,1*54
$GLGSV,1,1,00,1*78
$GAGSV,1,1,04,07,78,023,30,30,67,054,31,27,54,209,28,08,36,118,24,7*7A
$GBGSV,1,1,00,1*76
$GQGSV,1,1,00,1*65
$GIGSV,1,1,00,1*7D
```

NOTE

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

2.2.4. GSA

GNSS DOP and Active Satellites. GNSS receiver operating mode, satellites used in the navigation solution reported by the **GGA** or **GNS** 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 2: 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

Field	Format	Unit	Example	Description
				mode. A = Automatic, allowed to automatically switch to 2D or 3D mode.
<FixMode>	Numeric	-	3	Fix mode. 1 = Fix not available 2 = 2D fix 3 = 3D fix
Start of repeat block. Repeat times: 12.				
<SatID>	Numeric	-	32	ID numbers of satellites used in solution. See *Table 6: GNSS Satellites (NMEA) Numbering*. Note that this field is empty in case of an invalid value.
End of repeat block.				
<PDOP>	Numeric	-	1.25	Position dilution of precision. Note that this field is 99.99 in case of an invalid value.
<HDOP>	Numeric	-	0.93	Horizontal dilution of precision. Note that this field is 99.99 in case of an invalid value.
<VDOP>	Numeric	-	0.82	Vertical dilution of precision. Note that this field is 99.99 in case of an invalid value.
<SystemID>	Numeric	-	1	GNSS system ID. See *Table 6: GNSS Satellites (NMEA) Numbering*.
<Checksum>	Hexadecimal	-	*06	Checksum.
<CR><LF>	Character	-	-	Carriage return and line feed.

Example:

```
$GNGSA,A,3,32,28,31,26,25,29,10,02,03,,,,1.25,0.93,0.82,1*06
$GNGSA,A,3,,,,,,,,,,,,,1.25,0.93,0.82,2*04
$GNGSA,A,3,07,30,27,08,,,,,,,,,1.25,0.93,0.82,3*0C
$GNGSA,A,3,,,,,,,,,,,,,1.25,0.93,0.82,4*02
$GNGSA,A,3,,,,,,,,,,,,,1.25,0.93,0.82,5*03
$GNGSA,A,3,,,,,,,,,,,,,1.25,0.93,0.82,6*00
```

NOTE

If less than 12 satellites are used for navigation, the remaining **<SatID>** fields are left empty. If more than 12 satellites are used, only the IDs of the first 12 satellites will be 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 2: NMEA Talker ID*.
VTG	String, 3 characters	-	VTG	Course Over Ground & Ground Speed.
<COGT>	Numeric	Degrees	294.71	Course over ground, in true north direction. Note that this field is empty in case of an invalid value.
T	Character	-	T	Fixed field: true.
<COGM>	Numeric	Degrees	-	Course over ground (magnetic). Not supported. Always null.
M	Character	-	M	Fixed field: magnetic.
<SOGN>	Numeric	Knots	0.30	Speed over ground in knots. Variable length. Note that this field is empty in case of an invalid value.
N	Character	-	N	Fixed field: knot.
<SOGK>	Numeric	km/h	0.55	Speed over ground in kilometers per hour. Variable length. Note that this field is empty in case of an

Field	Format	Unit	Example	Description
				invalid value.
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 N = Data not valid
<Checksum>	Hexadecimal	-	*29	Checksum.
<CR><LF>	Character	-	-	Carriage return and line feed.

Example:

```
$GNVTG,294.71,T,,M,0.30,N,0.55,K,A*29
```

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 2: NMEA Talker ID*.

Field	Format	Unit	Example	Description
GLL	String, 3 characters	-	GLL	Geographic Position – Latitude/Longitude
<Lat>	ddmm.mmmmmm	-	3149.302881	Latitude. dd: Degrees (00–90) mm: Minutes (00–59) mmmmmm: Decimal fraction of minutes. Variable length. 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	-	11706.921911	Longitude. ddd: Degrees (000–180) mm: Minutes (00–59) mmmmmm: Decimal fraction of minutes. Variable length. 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	-	083706.000	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

Field	Format	Unit	Example	Description
				(SBAS). E = Estimated (dead reckoning) mode. N = Data not valid.
<Checksum>	Hexadecimal	-	*40	Checksum.
<CR><LF>	Character	-	-	Carriage return and line feed.

Example:

```
$GNGLL,3149.302881,N,11706.921911,E,083706.000,A,A*40
```

2.2.7. 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 2: NMEA Talker ID*.
ZDA	String, 3 characters	-	ZDA	Time & Date. UTC, day, month, year and local time zone.
<UTC>	hhmmss.sss	-	013813.000	Position fix UTC. hh: Hours (00–23) mm: Minutes (00–59) ss: Seconds (00–59) sss: Decimal fraction of seconds
<Day>	Numeric	-	31	Day of month. Range: 01–31.
<Month>	Numeric	-	01	Month. Range: 01–12.
<Year>	Numeric	-	2024	Year.

Field	Format	Unit	Example	Description
<LocalHour>	Numeric	-	00	Local zone hours, 00 to ±13 hours. Not supported. Always “00”.
<LocalMin>	Numeric	-	00	Local zone minutes, 00 to 59 minutes. Not supported. Always “00”.
<Checksum>	Hexadecimal	-	*47	Checksum.
<CR><LF>	Character	-	-	Carriage return and line feed.

Example:

```
$GNZDA,013813.000,31,01,2024,00,00*47
```

2.2.8. 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	-	GN	Talker identifier. See *Table 2: NMEA Talker ID*.
GST	String, 3 characters	-	GST	GNSS Pseudorange Error Statistics.
<UTC>	hhmmss.sss	-	061549.000	Position fix UTC. hh: Hours (00–23) mm: Minutes (00–59) ss: Seconds (00–59) sss: Decimal fraction of seconds
<RMS_D>	Numeric	Meter	8.2	RMS value of the standard deviation of the range inputs to the navigation

Field	Format	Unit	Example	Description
				process.
<MajorD>	Numeric	Meter	2.6	Standard deviation of semi-major axis of error ellipse.
<MinorD>	Numeric	Meter	2.4	Standard deviation of semi-minor axis of error ellipse.
<Orient>	Numeric	Degree	74.7	Orientation of semi-major axis of error ellipse.
<LatD>	Numeric	Meter	2.4	Standard deviation of latitude error.
<LonD>	Numeric	Meter	2.6	Standard deviation of longitude error.
<AltD>	Numeric	Meter	8.5	Standard deviation of altitude error.
<Checksum>	Hexadecimal	-	*45	Checksum.
<CR><LF>	Character	-	-	Carriage return and line feed.

Example:

```
$GNGST,061549.000,8.2,2.6,2.4,74.7,2.4,2.6,8.5*45
```

NOTE

This message is only supported by LC29H (AI).

2.2.9. GNS

GNSS Fix Data. Fix data for single or combined satellite navigation systems (GNSS).

Type:

Output

Synopsis:

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

Parameter:

Field	Format	Unit	Example	Description
$	Character	-	$	Each NMEA message starts with **$**.
<TalkerID>	String, 2 characters	-	GN	Talker identifier. See *Table 2: NMEA Talker ID*.
GNS	String, 3 characters	-	GNS	GNSS Fix Data.
<UTC>	hhmmss.sss	-	033933.000	Position fix UTC. hh: Hours (00–23) mm: Minutes (00–59) ss: Seconds (00–59) sss: Decimal fraction of seconds
<Lat>	ddmm.mmmmmm	-	3149.302701	Latitude. dd: Degrees (00–90) mm: Minutes (00–59) mmmmmm: Decimal fraction of minutes. Variable length. 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	-	11706.919283	Longitude. ddd: Degrees (000–180) mm: Minutes (00–59) mmmmmm: Decimal fraction of minutes. Variable length. 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.
<ModeInd> [1)]	Character	-	DDDDD	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.

Field	Format	Unit	Example	Description
				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.
<NumSatUsed>	Numeric	-	36	Total number of satellites in use. Range: 00–99.
<HDOP>	Numeric	-	0.46	Horizontal dilution of precision. Maximum value: 99.00.
<Alt>	Numeric	Meter	95.808	Antenna altitude above the mean-sea-level (geoid).
<Sep>	Numeric	Meter	-0.337	Geoid separation (the difference between the earth ellipsoid surface and the mean-sea-level (geoid) surface defined by the reference datum used in the position solution).
<DiffAge>	-	-	-	Differential GPS data age. Not supported. Always null.
<DiffStation>	-	-	-	Differential reference station ID. Not supported. Always null.
<NavStatus>	Character	-	V	Navigational status indicator. Not supported. Always “V” (navigational status not valid).
<Checksum>	Hexadecimal	-	*0B	Checksum.
<CR><LF>	Character	-	-	Carriage return and line feed.

Example:

```
$GNGNS,033933.000,3149.302701,N,11706.919283,E,DDDDDD,36,0.46,95.808,-0.337,,,V*0B
```

NOTE

1. **<ModeInd>** is a variable length field. The first character indicates the use of GPS satellites, the second character indicates the use of GLONASS satellites, and the third character indicates the use of Galileo satellites, the fourth character indicates the use of BDS satellites, the fifth character indicates the use of QZSS satellites, and the sixth character indicates the use of NavIC (IRNSS) satellites.
2. This message is only supported by LC29H (AI).

2.3. PQTM Messages

This chapter explains the **PQTM** messages (proprietary NMEA messages defined by Quectel) supported by L89 (HD) and LC29H (AI) GNSS modules.

Table 3: Error Codes

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

2.3.1. PQTMCOLD

Performs a cold start, which restart the GNSS engine without location information, including time, position, almanacs, and ephemeris data.

Type:

Command

Synopsis:

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

Parameter:

None

Result:

- If successful, the module is restarted and no message is sent as a reply.
- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 3: Error Codes*.

Example:

```
$PQTMCOLD*1C
```

2.3.2. PQTMWARM

Performs a warm start, which restarts the GNSS engine with the valid position, time, and almanac data. However, the ephemeris data is invalid, therefore, the receiver must download the updated ephemeris data before achieving a fix.

Type:

Command

Synopsis:

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

Parameter:

None

Result:

- If successful, the module is restarted and no message is sent as a reply.
- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 3: Error Codes*.

Example:

```
$PQTMWARM*11
```

2.3.3. PQTMHOT

Performs a hot start, which restarts the GNSS engine with the valid position, time, ephemeris, and almanac data, enabling the fastest location acquisition.

Type:

Command

Synopsis:

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

Parameter:

None

Result:

- If successful, the module is restarted and no message is sent as a reply.
- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 3: Error Codes*.

Example:

```
$PQTMHOT*4B
```

2.3.4. PQTMSRR

Performs a system reset and reboots the receiver.

Type:

Command

Synopsis:

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

Parameter:

None

Result:

- If successful, the module is rebooted and no message is sent as a reply.
- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 3: Error Codes*.

Example:

```
$PQTMSRR*4B
```

2.3.5. PQTMVERNO

Queries the firmware version information.

Type:

Command

Synopsis:

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

Parameter:

None

Result:

- If successful, the module returns:

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

Parameters included in the result:

Field	Format	Unit	Description
<VerStr>	String	-	Version string.
<BuildDate>	yyyy/mm/dd	-	Firmware build date.
<BuildTime>	hh:mm:ss	-	Firmware build time.

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 3: Error Codes*.

Example:

```
$PQTMVERNO*58
$PQTMVERNO,LC29HAINR01A01S,2023/09/15,09:55:24*3F
```

2.3.6. PQTMSAVEPAR

Saves the configurations into NVM.

Type:

Command

Synopsis:

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

Parameter:

None

Result:

- If successful, the module returns:

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

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 3: Error Codes*.

Example:

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

2.3.7. PQTMRESTOREPAR

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

Type:

Command

Synopsis:

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

Parameter:

None

Result:

- If successful, the module returns:

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

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 3: Error Codes*.

Example:

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

2.3.8. PQTMCFGMSGRATE

Sets/gets the message output rate on the current port.

Type:

Set/Get

Synopsis:

```
//Set:
$PQTMCFGMSGRATE,W,<MsgName>,<Rate>[,<MsgVer>]*<Checksum><CR><LF>
//Get:
$PQTMCFGMSGRATE,R,<MsgName>[,<MsgVer>]*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<MsgName>	String	-	Configuration message name. See *Table 4: Supported Messages* for details.
<Rate>	Numeric	-	Message output rate. 0 = Output disabled. N = Output once every N position fix(es). Range of N: see *Table 4: Supported Messages* for details.
<MsgVer>	Numeric	-	Message version. Optional. This field can be omitted when the configuration message is standard NMEA 0183 message.

Result:

- If successful, the module returns:

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

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 3: Error Codes*.

Example:

```
//Set the output rate of $PQTMGEOFENCESTATUS message to once every position fix:
$PQTMCFGMSGRATE,W,PQTMGEOFENCESTATUS,1,1*5C
$PQTMCFGMSGRATE,OK*29

//Get the output rate of $PQTMGEOFENCESTATUS message:
$PQTMCFGMSGRATE,R,PQTMGEOFENCESTATUS,1*44
$PQTMCFGMSGRATE,OK,PQTMGEOFENCESTATUS,1,1*0F

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

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

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

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

Table 4: Supported Messages

Message Name	Message Output Rate Range (N)
RMC	1–20
GGA	1–20
GSV	1–20
GSA	1–20
VTG	1–20
GLL	1–20
ZDA	1–20
GST	1–20
GNS	1–20
PQTMEPE	1–20
PQTMPL	1–20
PQTMGEOFENCESTATUS	1–20
PQTMJAMMINGSTATUS	1–20
PQTMPVT	1–20
PQTMDOP	1–20
PQTMANTENNASTATUS	1–20
PQTMODO	1–20
PQTMLS	1–20
PQTMVEL	1–20

NOTE

1. If the configuration message is a **PQTM** message, use **<MsgVer>** field to specify the message version, otherwise an error will be returned.
2. If the 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 the Set command. If the default setting had been changed by the Set command, contact Quectel Technical

Support (support@quectel.com) to get the default setting, if necessary.

2.3.9. PQTMEPE

Outputs the estimated positioning error.

Type:

Output

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<MsgVer>	Numeric	-	Message version. 2 = Version 2 (Always “2” for this version.)
<EPE_North>	Numeric	Meter	Estimated north error.
<EPE_East>	Numeric	Meter	Estimated east error.
<EPE_Down>	Numeric	Meter	Estimated down error.
<EPE_2D>	Numeric	Meter	Estimated 2D position error.
<EPE_3D>	Numeric	Meter	Estimated 3D position error.

Example:

```
$PQTMEPE,2,3.393,3.476,12.713,4.857,13.609*5D
```

NOTE

This message is only supported by LC29H (AI).

2.3.10. PQTMPL

Outputs protection level information.

Type:

Output

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<MsgVer>	Numeric	-	Message version. 1 = Version 1 (Always “1” for this message version.)
<TOW>	Numeric	ms	Time of week. Null if invalid.
<PUL>	Numeric	%	Probability of uncertainty level per epoch.
<Res1>	Numeric	-	Reserved. Always “1”.
<Res2>	Numeric	-	Reserved. Always “1”.
<PL_PosN>	Numeric	mm	Protection level of north position. Null if invalid.
<PL_PosE>	Numeric	mm	Protection level of east position. Null if invalid.
<PL_PosD>	Numeric	mm	Protection level of down position. Null if invalid.
<PL_VelN>	Numeric	mm/s	Protection level of north velocity. Null if invalid.
<PL_VelE>	Numeric	mm/s	Protection level of east velocity. Null if invalid.
<PL_VelD>	Numeric	mm/s	Protection level of down velocity. Null if invalid.
<Res3>	Numeric	-	Reserved. Always null.
<Res4>	Numeric	-	Reserved. Always null.
<PL_Time>	Numeric	ns	Protection level of time. Null if invalid.

Example:

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

NOTE

This message is only supported by LC29H (AI).

2.3.11. PQTMCFGGEOFENCE

Sets/gets geofence feature.

Type:

Set/Get

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<Index>	Numeric	-	Geofence index. Range: 0–3.
<Status>	Numeric	-	Geofence feature status. 0 = Disabled 1 = Enabled
<Reserved>	Numeric	-	Reserved. Always “0”.
<Shape>	Numeric	-	Geofence shape. 0 = Circle defined by the center and the radius 1 = Circle defined by the center and a point on the circle 2 = Triangle 3 = Quadrangle (such as square, rectangle, trapezium.)
<Lat0>	Numeric	Degree	Latitude of the first point.
<Lon0>	Numeric	Degree	Longitude of the first point.
<Lat1/Radius>	Numeric	Degree Meter	If the geofence shape is a circle with a certain radius, this value will be the radius of the circle, otherwise, this value will be the latitude of the second point.
<Lon1>	Numeric	Degree	Longitude of the second point.

Field	Format	Unit	Description
<Lat2>	Numeric	Degree	Latitude of the third point.
<Lon2>	Numeric	Degree	Longitude of the third point.
<Lat3>	Numeric	Degree	Latitude of the fourth point.
<Lon3>	Numeric	Degree	Longitude of the fourth point.

Result:

- If successful, the module returns:

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

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 3: Error Codes*.

Example:

```
//Set:
$PQTMCFGGEOFENCE,W,0,1,0,0,31.451248,117.451245,100.5*18
$PQTMCFGGEOFENCE,OK*74

//Get:
$PQTMCFGGEOFENCE,R,0*3E
$PQTMCFGGEOFENCE,OK,0,1,0,0,31.451248,117.451245,100.500000*7B
```

2.3.12. PQTMGEOFENCESTATUS

Outputs the geofence status.

Type:

Output

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<MsgVer>	Numeric	-	Message version. 1 = Version 1 (Always “1” for this version.)
<Time>	hhmmss.sss	-	UTC time.
Start of repeat block. Repeat times: 4.			
<StateN>	Numeric	-	Geofence N state. Range of N: 0–3. 0 = Unknow 1 = Inside 2 = Outside
End of repeat block.			

Example:

```
$PQTMGEOFENCESTATUS,1,093444.000,2,0,0,0*29
```

2.3.13. PQTMJAMMINGSTATUS

Outputs the jamming detection status.

Type:

Output

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<MsgVer>	Numeric	-	Message version. 1 = Version 1 (Always “1” for this message version.)
<Status>	Numeric	-	Jamming detection status. 0 = Unknown 1 = No jamming, healthy status 2 = Warning status

Field	Format	Unit	Description
			3 = Critical status

Example:

```
$PQTMJAMMINGSTATUS,1,1*47
```

2.3.14. PQTMCFGPPS

Sets/gets the 1PPS feature configuration.

Type:

Set/Get

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<Index>	Numeric	-	PPS index. 1 = PPS 1
<Enable>	Numeric	-	Enable/disable PPS output. 0 = Disable 1 = Enable
<Duration>	Numeric	ms	Pulse duration. Range: 0–900. Default value: 100.
<Mode>	Numeric	-	PPS output with fix mode. 1 = Always 2 = 2D 3 = 3D 4 = After first fix
<Polarity>	Numeric	-	Pulse polarity. 0 = Low 1 = High
<Reserved>	Numeric	-	Reserved. Always “0”.

Result:

- If successful, the module returns:

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

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 3: Error Codes*.

Example:

```
//Set:
$PQTMCFGPPS,W,1,1,100,4,1,0*76
$PQTMCFGPPS,OK*21

//Get:
$PQTMCFGPPS,R,1*6A
$PQTMCFGPPS,OK,1,1,100,4,1,0*25
```

2.3.15. PQTMCFGSBAS

Sets/gets the SBAS configuration.

Type:

Set/Get

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<Value>	Hexadecimal	-	SBAS configuration bitmask. Bit 0: WAAS Bit 1: Reserved. Always “0”. Bit 2: EGNOS Bit 3: Reserved. Always “0”. Bit 4: MSAS Bit 5: GAGAN

Result:

- If successful, the module returns:

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

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 3: Error Codes*.

Example:

```
//Set:
$PQTMCFGSBAS,W,35*08
$PQTMCFGSBAS,OK*71

//Get:
$PQTMCFGSBAS,R*27
$PQTMCFGSBAS,OK,35*5B
```

2.3.16. PQTMGNSSSTART

Starts GNSS engine.

Type:

Command

Synopsis:

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

Parameter:

None

Result:

- If successful, the module returns:

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

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 3: Error Codes*.

Example:

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

2.3.17. PQTMGNSSSTOP

Stops GNSS engine.

Type:

Command

Synopsis:

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

Parameter:

None

Result:

- If successful, the module returns:

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

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 3: Error Codes*.

Example:

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

2.3.18. PQTMCFGFIXRATE

Sets/gets the position fix interval.

Type:

Set/Get

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<Interval>	Numeric	ms	Position fix interval. Supported values: 100, 200, 500, and 1000.

Result:

- If successful, the module returns:

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

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 3: Error Codes*.

Example:

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

$PQTMCFGFIXRATE,OK*27

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

2.3.19. PQTMCFGCNST

Sets/gets the constellation configuration.

Type:

Set/Get

Synopsis:

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

Parameter:

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

Result:

- If successful, the module returns:

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

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 3: Error Codes*.

Example:

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

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

NOTE

1. Supported constellation configurations (QZSS is always switchable):
   - GPS + GLONASS + Gailieo + BDS + NavIC
   - GPS + GLONASS + Galileo + BDS
   - GPS + GLONASS + Galileo
   - GPS + GLONASS + BDS
   - GPS + Galileo + BDS
   - GPS + Gailieo + NavIC
   - GPS + GLONASS
   - GPS + Galileo
   - GPS + BDS
   - BDS
   - GPS
   - NavIC
2. The configuration made by this command takes effect after executing **PQTMSAVEPAR** and then restarting the module.

2.3.20. PQTMPVT

Outputs the PVT (GNSS only) result.

Type:

Output

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<MsgVer>	Numeric	-	Message version. 1 = Version 1 (Always “1” for this version.)
<TOW>	Numeric	ms	Time of week.
<Date>	YYYYMMDD	-	UTC date. YYYY: Year MM: Month DD: Day of month
<Time>	hhmmss.sss	-	UTC time. hh: Hour (00–23) mm: Minute (00–59) ss: Second (00–59) sss: Decimal fraction of second.
<Res>	Numeric	-	Reserved. Always null.
<FixMode>	Numeric	-	Fix mode. 0 = No fix 1 = Reserved 2 = 2D fix 3 = 3D fix
<NumSatUsed>	Numeric	-	Number of satellites in use.
<LeapS>	Numeric	Second	Leap seconds. Note that this field is empty in case of an invalid value.
<Lat>	Numeric	Degree	Latitude. Note that this field is empty in case of an invalid value.
<Lon>	Numeric	Degree	Longitude.

Field	Format	Unit	Description
			Note that this field is empty in case of an invalid value.
<Alt>	Numeric	Meter	Altitude above mean sea level. Note that this field is empty in case of an invalid value.
<Sep>	Numeric	Meter	Geoidal separation (the difference between the WGS84 earth ellipsoid surface and the mean-sea-level surface). Note that this field is empty in case of an invalid value.
<VelN>	Numeric	m/s	North velocity. Note that this field is empty in case of an invalid value.
<VelE>	Numeric	m/s	East velocity. Note that this field is empty in case of an invalid value.
<VelD>	Numeric	m/s	Down velocity. Note that this field is empty in case of an invalid value.
<Spd>	Numeric	m/s	Ground speed. Note that this field is empty in case of an invalid value.
<Heading>	Numeric	Degree	Heading. Range: 0.000–360.000. Note that this field is empty in case of an invalid value.
<HDOP>	Numeric	-	Horizontal dilution of precision. Note that the value is 99.99 in case of an invalid value.
<PDOP>	Numeric	-	Position (3D) dilution of precision. Note that the value is 99.99 in case of an invalid value.

Example:

```
$PQTMPVT,1,207373000,20231121,093555.000,,3,42,18,31.82169570,117.11532720,91.134,-0.336,0.005,-0.002,-0.010,0.005,174.720,0.43,0.68*4C
```

2.3.21. PQTMDOP

Outputs dilution of precision.

Type:

Output

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<MsgVer>	Numeric	-	Message version. 1 = Version 1 (Always “1” for this message version.)
<TOW>	Numeric	ms	Time of week. Note that this field is empty in case of an invalid value.
<GDOP>	Numeric	-	Geometric dilution of precision. Note that the value is 99.99 in case of an invalid value.
<PDOP>	Numeric	-	Position (3D) dilution of precision. Note that the value is 99.99 in case of an invalid value.
<TDOP>	Numeric	-	Time dilution of precision. Note that the value is 99.99 in case of an invalid value.
<VDOP>	Numeric	-	Vertical dilution of precision. Note that the value is 99.99 in case of an invalid value.
<HDOP>	Numeric	-	Horizontal dilution of precision. Note that the value is 99.99 in case of an invalid value.
<NDOP>	Numeric	-	Northing dilution of precision. Note that the value is 99.99 in case of an invalid value.
<EDOP>	Numeric	-	Easting dilution of precision. Note that the value is 99.99 in case of an invalid value.

Example:

```
$PQTMDOP,1,2313,0.85,0.74,0.42,0.62,0.40,99.99,99.99*4B
```

2.3.22. PQTMCFGNMEADP

Sets/gets the decimal places of NMEA messages.

Type:

Set/Get

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<UTC_DP>	Numeric	-	Number of decimal places for UTC seconds in standard NMEA messages. Range: 0–3. Default value: 3. 0 = No fractional part.
<POS_DP>	Numeric	-	Number of decimal places for latitude and longitude in standard NMEA messages. Range: 0–8. Default value: 6. 0 = No fractional part.
<ALT_DP>	Numeric	-	Number of decimal places for altitude and geoidal separation in standard NMEA messages. Range: 0–3. Default value: 1 for L89 (HD) and 3 for LC29H (AI). 0 = No fractional part.
<DOP_DP>	Numeric	-	Number of decimal places for DOP in standard NMEA messages. Range: 0–3. Default value: 2. 0 = No fractional part.
<SPD_DP>	Numeric	-	Number of decimal places for speed in standard NMEA messages. Range: 0–3. Default value: 3 for L89 (HD) and 2 for LC29H (AI). 0 = No fractional part.
<COG_DP>	Numeric	-	Number of decimal places for COG in standard NMEA messages. Range: 0–3. Default value: 2.

Field	Format	Unit	Description
			0 = No fractional part.

Result:

- If successful, the module returns:

```
//Response to Set command:
$PQTMCFGNMEADP,OK*<Checksum><CR><LF>
//Response to Get command:
$PQTMCFGNMEADP,OK,<UTC_DP>,<POS_DP>,<ALT_DP>,<DOP_DP>,<SPD_DP>,<COG_DP>*<Checksum><CR><LF>
```

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 3: Error Codes*.

Example:

```
//Set:
$PQTMCFGNMEADP,W,3,6,1,2,3,2*35
$PQTMCFGNMEADP,OK*61

//Get:
$PQTMCFGNMEADP,R*37
$PQTMCFGNMEADP,OK,3,6,1,2,3,2*66
```

2.3.23. PQTMCFGNMEATID

Sets/gets the NMEA Talker ID.

Type:

Set/Get

Synopsis:

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

Parameter:

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

Result:

- If successful, the module returns:

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

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 3: Error Codes*.

Example:

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

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

2.3.24. PQTMCFGPROT

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

Type:

Set/Get

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<PortType>	Numeric	-	Port type. 1 = UART 2 = I2C 3 = SPI
<PortID>	Numeric	-	Port ID. 1 = Port 1 (For example, when **<PortType>** is 1 and **<PortID>** is 1, the current port is UART1.)
<InputProt>	Hexadecimal	-	Input protocol. Bit 0: NMEA Bit 1: Reserved. Always “0”. Bit 2: RTCM3 Bit 3–Bit 31: Reserved. Always “0”.
<OutputProt>	Hexadecimal	-	Output protocol. Bit 0: NMEA Bit 1: Reserved. Always “0”. Bit 2: RTCM3 Bit 3–Bit 31: Reserved. Always “0”.

Result:

- If successful, the module returns:

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

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 3: Error Codes*.

Example:

```
//Set:
```

$PQTMCFGPROT,W,1,1,5,1*3C
$PQTMCFGPROT,OK*6B

//Get:
$PQTMCFGPROT,R,1,1*3D
$PQTMCFGPROT,OK,1,1,00000005,00000001*6F

NOTE

Only LC29H (AI) supports the SPI.

2.3.25. PQTMDEBUGON

Enables debug log outputting. The debug-on state can be saved by using the command **PQTMSAVEPAR**.

Type:

Command

Synopsis:

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

Parameter:

None

Result:

- If successful, the module returns:

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

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 3: Error Codes*.

Example:

$PQTMDEBUGON*48
$PQTMDEBUGON,OK*60

2.3.26. PQTMDEBUGOFF

Disables debug logging. The debug-off state can be saved by using the command **PQTMSAVEPAR**.

Type:

Command

Synopsis:

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

Parameter:

None

Result:

- If successful, the module returns:

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

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 3: Error Codes*.

Example:

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

2.3.27. PQTMUNIQID

Queries the chip unique ID of the module.

Type:

Command

Synopsis:

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

Parameter:

None

Result:

- If successful, the module returns:

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

Parameters included in the result:

Field	Format	Unit	Description
<Lenth>	Numeric	Byte	Length of chip unique ID.
<ID>	Hexadecimal	-	Chip unique ID.

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 3: Error Codes*.

Example:

```
$PQTMUNIQID*16
$PQTMUNIQID,OK,16,151688874DEB6B8CE6DD34C8DD1789E4*34
```

2.3.28. PQTMCFGUART

Sets/gets UART configuration.

Type:

Set/Get

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<Index>	Numeric	-	UART port index. 1 = UART (for L89 (HD)) or UART1 (for LC29H (AI))
<BaudRate>	Numeric	bps	UART baud rate. Supported values: 9600, 19200, 38400, 57600, 115200, 230400, 460800, and 921600.
<DataBit>	Numeric	bit	UART data bit. 7 = 7 bits 8 = 8 bits
<Parity>	Numeric	-	Parity. 0 = No parity 1 = Odd parity 2 = Even parity
<StopBit>	Numeric	-	Stop bit(s). 1 = 1 stop bit 2 = 2 stop bits
<FlowCtrl>	Numeric	-	Flow control. 0 = None 4 = Software flow control

Result:

- If successful, the module returns:

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

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 3: Error Codes*.

Example:

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

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

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

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

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

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

NOTE

The configuration made by this command takes effect after executing **PQTMSAVEPAR** and then restarting the module.

2.3.29. PQTMCFGANTENNA

Sets/gets the antenna working mode.

Type:

Set/Get

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<Reserved>	-	-	Reserved. Always "0".
<Mode>	Numeric	-	Antenna working mode. 0 = Automatic mode 1 = Manual mode, using integrated antenna 2 = Manual mode, using external antenna

Result:

- If successful, the module returns:

```
//Set:
$PQTMCFGANTENNA,OK*<Checksum><CR><LF>
//Get:
$PQTMCFGANTENNA,OK,<Reserved>,<Mode>*<Checksum><CR><LF>
```

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 3: Error Codes*.

Example:

```
//Set:
$PQTMCFGANTENNA,W,0,0*7E
$PQTMCFGANTENNA,OK*2D

//Get:
$PQTMCFGANTENNA,R*7B
$PQTMCFGANTENNA,OK,0,0*2D
```

NOTE

This command is only supported by L89 (HD).

2.3.30. PQTMANTENNASTATUS

Outputs the antenna status for the module which supports internal antenna and external antenna.

Type:

Output

Synopsis:

```
$PQTMANTENNASTATUS,<MsgVer>,<AntStatus>,<AntPowerInd>,<ModeInd>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<MsgVer>	Numeric	-	Message version. 3 = Version 3 (Always “3” for this version.)
<AntStatus>	Numeric	-	Antenna status. 0 = Unknow 1 = Normal 2 = Open-circuit 3 = Short-circuit
<AntPowerInd>	Numeric	-	Antenna power indicator. 0 = Power-off 1 = Power-on 2 = Unknow
<ModeInd>	Numeric	-	Antenna mode indicator. 0 = Unknow 1 = Automatic mode, using integrated antenna 2 = Automatic mode, using external antenna 3 = Manual mode, using integrated antenna 4 = Manual mode, using external antenna

Example:

```
$PQTMANTENNASTATUS,3,2,1,1*52
```

NOTE

This message is only supported by L89 (HD).

2.3.31. PQTMCFGODO

Sets/gets the odometer feature configuration.

Type:

Set/Get

Synopsis:

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

Parameter:

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

Result:

- If successful, the module returns:

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

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 3: Error Codes*.

Example:

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

//Get:
$PQTMCFGODO,R*60
```

$PQTMCFGODO,OK,1,10.5*1D

2.3.32. PQTMRESETODO

Resets the accumulated distance recorded by the odometer.

Type:

Command

Synopsis:

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

Parameter:

None

Result:

- If successful, the module returns:

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

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 3: Error Codes*.

Example:

$PQTMRESETODO*09
$PQTMRESETODO,OK*21

NOTE

To reset the accumulated distance recorded by the odometer, you have two options. You can either use **PQTMRESETODO** command or power off the module. Disabling the odometer feature with **PQTMCFGODO** command while the module is still working will stop distance calculation, but it cannot reset the distance to zero.

2.3.33. PQTMODO

Outputs the odometer information.

Type:

Output

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<MsgVer>	Numeric	-	Message version. 1 = Version 1 (Always “1” for this version.)
<Time>	hhmmss.sss	-	UTC time. hh: Hour (00–23) mm: Minute (00–59) ss: Second (00–59) sss: Decimal fraction of second
<State>	Numeric	-	Odometer status. 0 = Disabled 1 = Enabled
<Dist>	Numeric	Meter	Distance since last reset.

Example:

```
$PQTMODO,1,085601.000,1,12.0*55
```

NOTE

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

2.3.34. PQTMLS

Outputs leap second forecast information.

Type:

Output

Synopsis:

```
$PQTMLS,<MsgVer>,<TOW>,<LS_Ref>,<WN>,<LS>,<Flag>,<LSF_Ref>,<Reserved>,<WNLSF>,<DN>
,<LSF>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<MsgVer>	Numeric	-	Message version. 1 = Version 1 (Always “1” for this version.)
<TOW>	Numeric	s	Time of week.
<LS_Ref>	Numeric	-	Referenced constellation by the current leap second information: 0 = No source 1 = GPS 2 = Reserved 3 = Galileo 4 = BDS
<WN>	Numeric	-	UTC reference week number.
<LS>	Numeric	s	Current number of leap seconds since the beginning of GPS time (January 6, 1980). It reflects how far ahead GPS time is compared to UTC time. Galileo has the same number of leap seconds as GPS. BDS has 14 fewer leap seconds than GPS. GLONASS follows UTC time, so no leap seconds.
<Flag>	Numeric	-	Valid marker for future occurrences of leap seconds. 0 = Invalid 1 = Available
<LSF_Ref>	Numeric		Referenced constellation by the leap second forecast information. 0 = No source 1 = GPS 2 = Reserved 3 = Galileo 4 = BDS The field value is invalid if **<Flag>** = 0.
<Reserved>	-	-	Reserved. Always null.

Field	Format	Unit	Description
<WNLSF>	Numeric	-	Week number of the new leap second. The field value is invalid if **<Flag>** = 0.
<DN>	Numeric	-	The day of the week when the new leap second takes effect. GPS & Galileo: 1–7 from Sunday to Saturday; BDS: 0–6 from Sunday to Saturday. The field value is invalid if **<Flag>** = 0.
<LSF>	Numeric	s	Leap second count after future leap second changes. The field value is invalid if **<Flag>** = 0.

Example:

```
$PQTMLS,1,195494,1,2299,18,0,1,,137,7,18*2C
```

NOTE

1. GPS Week is a time system used internally by the GPS system. Time zero is: 01/06/1980 00:00:00. Every 1024 weeks (7168 days) is a cycle. The first GPS weekly cycle commenced on 08/22/1999 00:00:00, signifying the reset of the week count to 0. After this point, weeks are counted again and the week numbering follows the rule where Sunday is designated as 1, and is sequentially recorded as 1–7.
2. The starting time of BDS satellite navigation time system is 01/01/2006 00:00:00 UTC. The system utilizes Week and intra-week seconds counts. The week counting rule is: Sunday is designated as 0 and is sequentially recorded as 0–6.
3. The parameters **<LSF>** minus **<LS>** can have one of the three values:
   - +1 = Positive leap second;
   - -1 = Negative leap second;
   - 0 = No future leap second event scheduled or no information available.
4. Reference priority for leap second information: GPS > Galileo > BDS.

2.3.35. PQTMVEL

Output the velocity information.

Type:

Output

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<MsgVer>	Numeric	-	Message version. 1 = Version 1 (Always “1” for this version.)
<Time>	hhmmss.sss	-	UTC time.
<VelN>	Numeric	m/s	North velocity.
<VelE>	Numeric	m/s	East velocity.
<VelD>	Numeric	m/s	Down velocity.
<GrdSpd>	Numeric	m/s	2D speed.
<Spd>	Numeric	m/s	3D speed.
<Heading>	Numeric	Degree	Heading.
<GrdSpdAcc>	Numeric	m/s	Estimate of 2D speed accuracy.
<SpdAcc>	Numeric	m/s	Estimate of 3D speed accuracy.
<HeadingAcc>	Numeric	Degree	Estimate of heading accuracy.

Example:

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

2.3.36. PQTMBKP

Shuts down GNSS system at once to save power (that is, sets the module to Backup mode). In Backup mode, the module cannot execute PQTM commands and the GNSS engine no longer provides position-related information.

The module can be awakened from Backup mode by using the WAKEUP pin or the timer. When the module exits Backup mode, all system resources are reinitialized.

Type:

Command

Synopsis:

```
$PQTMBKP[,<Second>]*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Second>	Numeric	Second	Timer for exiting Backup mode. Range: 0 and 10–62208000 (about 2 years); 0 means entering the Backup mode without any timer. This field can be omitted when it is set to 0.

Result:

- If successful, the module enters Backup mode and no message is sent as a reply. It exits Backup mode upon the timer's expiration if a timer is set; otherwise, it remains in Backup mode indefinitely.
- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 3: Error Codes*.

Example:

```
//Enter Backup mode on a 66-second timer, and no message is sent as a reply:
$PQTMBKP,66*6D
//Enter Backup mode without any timer, and no message is sent as a reply:
$PQTMBKP,0*5D
```

3 Appendix A References

Table 5: Terms and Abbreviations

Abbreviation	Description
2D	2 Dimension
3D	3 Dimension
ACK	Acknowledgement
BDS	BeiDou Navigation Satellite System
C/N_0	Carrier-to-Noise-Density Ratio
COG	Course over Ground
COGM	Course over Ground (in Magnetic North Course Direction)
COGT	Course over Ground (in True North Course Direction)
DOP	Dilution of Precision
EDOP	Easting Dilution of Precision
EGNOS	European Geostationary Navigation Overlay Service
GAGAN	GPS Aided Geo Augmented Navigation
Galileo	Galileo Satellite Navigation System (EU)
GDOP	Geometric Dilution of Precision
GGA	Global Positioning System Fix Data
GLL	Geographic Position-Latitude and Longitude
GNS	GNSS Fix Data
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GSA	GNSS DOP and Active Satellites

Abbreviation	Description
GST	GNSS Pseudorange Error Statistics
GSV	GNSS Satellites in View
HDOP	Horizontal Dilution of Precision
I2C	Inter-Integrated Circuit
IRNSS/NavIC	Indian Regional Navigation Satellite System
MSAS	Multi-functional Satellite Augmentation System (Japan)
NDOP	Northing Dilution of Precision
NMEA	NMEA (National Marine Electronics Association) 0183 Interface Standard
PDOP	Position Dilution of Precision
PPS	Pulse Per Second
QZSS	Quasi-Zenith Satellite System
RMC	Recommended Minimum Specific GNSS Data
RTC	Real-Time Clock
SBAS	Satellite-Based Augmentation System
SNR	Signal-to-Noise Ratio
SPI	Serial Peripheral Interface
SV	Satellites in View
TDOP	Time Dilution of Precision
UART	Universal Asynchronous Receiver/Transmitter
UTC	Coordinated Universal Time
VDOP	Vertical Dilution of Precision
VTG	Course Over Ground & Ground Speed
WAAS	Wide Area Augmentation System
ZDA	Time & Date

4 Appendix B GNSS Numbering

Table 6: GNSS Satellites (NMEA) Numbering

GNSS Type	System ID	Satellite ID	Signal ID
GPS	1	1–32 33–51 for SBAS	1 = L1 C/A
GLONASS	2	65–88	1 = L1
Galileo	3	1–36	7 = E1
BDS	4	1–63	1 = B1I
QZSS	5	1–7	1 = L1 C/A
NavIC (IRNSS)	6	1–14	1 = L5

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.
Underline	Default setting of a parameter.