L89 (HA) GNSS Protocol Specification

GNSS Module Series

Version: 1.1

Date: 2025-04-03

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 (HA) GNSS Protocol Specification**
Subtitle	GNSS Module Series
Document Type	GNSS Protocol Specification
Document Status	Released

Revision History

Version	Date	Description
-	2021-03-25	Creation of the document
1.0	2022-10-25	First official release
1.1	2025-04-03	1. Deleted L89 (HB) according to Quectel's product marketing strategies. 2. Updated the note about output satellite IDs for GSA message (*Chapter 2.2.4*). 3. Added a note about range residual calculation method for GRS message (*Chapter 2.2.8*). 4. Added the table of error codes (*Table 3*). 5. Added a note about antenna status indication for PQTMANTENNASTATUS (*Chapter 2.3.1*). 6. Updated the PQTMVER message (*Chapter 2.3.4*). 7. Added the following PQTM messages (*Chapters 2.3.5*–*2.3.37*): ● PQTMSAVEPAR; ● PQTMRESTOREPAR; ● PQTMCFGMSGRATE; ● PQTMEPE; ● PQTMPL; ● PQTMCFGGEOFENCE; ● PQTMGEOFENCESTATUS;

Version	Date	Description
		● PQTMPVT; ● PQTMCFGNMEADP; ● PQTMCOLD; ● PQTMHOT; ● PQTMWARM; ● PQTMSRR; ● PQTMCFGSBAS; ● PQTMCFGCNST; ● PQTMDOP; ● PQTMCFGFIXRATE; ● PQTMCFGPPS; ● PQTMDEBUGON; ● PQTMDEBUGOFF; ● PQTMVEL; ● PQTMCFGODO; ● PQTMRESETODO; ● PQTMODO; ● PQTMJAMMINGSTATUS; ● PQTMLS; ● PQTMCFGNMEATID; ● PQTMUNIQID; ● PQTMCFGUART; ● PQTMCFGPROT; ● PQTMBKP; ● PQTMGNSSTART; ● PQTMGNSSTOP. 8. Added the following PAIR messages: ● PAIR011 (*Chapter 2.4.9*); ● PAIR080 (*Chapter 2.4.24*); ● PAIR081 (*Chapter 2.4.25*); ● PAIR382 (*Chapter 2.4.32*); ● PAIR512 (*Chapter 2.4.39*). 9. Deleted the note about the limitation on NMEA sentence output for PAIR050 (*Chapter 2.4.10*). 10. Updated the note about supported constellation configurations for PAIR066 (*Chapter 2.4.16*). 11. Updated the supported values of <Mode> for PAIR400 and PAIR401 (*Chapters 2.4.34* and *2.4.35*). 12. Added a note about limitations on SBAS usage in Fitness and Swimming modes (*Chapter 2.4.36*). 13. Updated the example of using PAIR511 and the related note (*Chapter 2.4.38*). 14. Updated the example of using PAIR513 and the related note (*Chapter*

Version	Date	Description
		2.4.40). 15. Updated the supported values of <BaudRate> for PAIR864 and PAIR865 (Chapters 2.4.43 and 2.4.44). 16. Added a note about baud rate configuration and the corresponding result for PAIR864 (Chapter 2.4.43). 17. Deleted PAIR490 and PAIR491 messages (Chapter 2.4).

Contents

Table Index

1 Introduction

Quectel L89 (HA) GNSS module supports GPS, Galileo, GLONASS, 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 (IRNSS) L5 frequency bands provides fast and accurate acquisition and makes this module 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) and the chipset supplier (PAIR 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. GLONASS L1 and BDS B1I are supported by L89 (HA) with L89HANR01A06S or higher versions.

2 NMEA Protocol

2.1. Structure of NMEA Protocol Messages

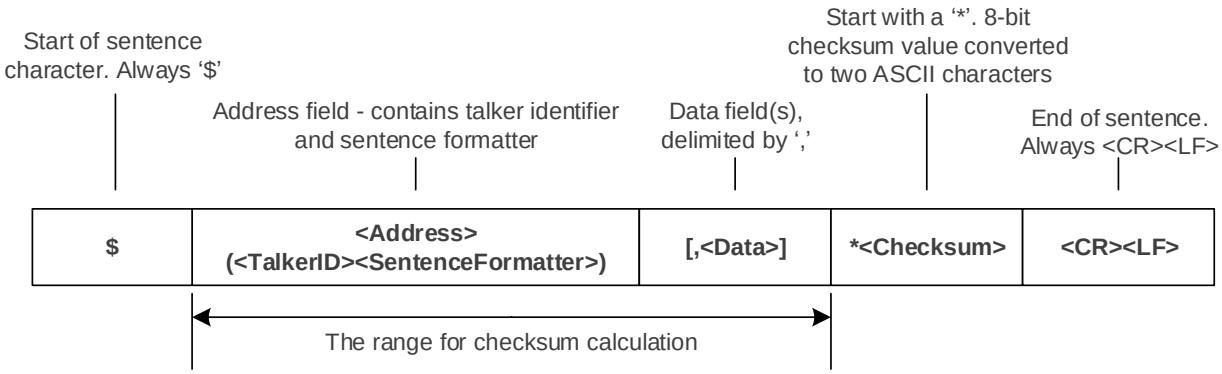

Figure 1: Structure of NMEA Protocol Messages

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 (SentenceFormatter). 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	TalkerID (NMEA 0183 V3.01)	TalkerID (NMEA 0183 V4.10)
GPS	GP	GP
GLONASS	GL	GL
Galileo	GA	GA
BDS	GB	GB
NavIC (IRNSS)	GI	GI
QZSS	GP	GP
Combination of Multiple Satellite Systems	GN	GN

Sample Code for NMEA Checksum:

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

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

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

```
    return result;
}
```

2.2. Standard Messages

This chapter explains the standard NMEA 0183 V3.01 and NMEA 0183 V4.10 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:

NMEA 0183 V3.01 format:

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

NMEA 0183 V4.10 format (default):

```
$<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	-	073925.000	Position fix UTC. hh: Hours (00–23) mm: Minutes (00–59) ss: Seconds (00–59)

Field	Format	Unit	Example	Description
				sss: Decimal fraction of seconds
<Status>	Character	-	A	Positioning system status. A = Data valid V = Navigation receiver warning
<Lat>	ddmm.mmmmmm	-	3149.333680	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	-	11706.947520	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	0.08	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	-	230222	Date. dd: Day of month mm: Month yy: Year
<MagVar>	-	-	-	Magnetic variation. Not supported. Always null.

Field	Format	Unit	Example	Description
<MagVarDir>	-	-	-	Direction of magnetic variation. Not supported. Always null.
<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" (Invalid. The device cannot provide navigational status indication). Please note that this parameter is only available in messages in line with NMEA 0183 V4.10 and above.
<Checksum>	Hexadecimal	-	*00	Checksum
<CR><LF>	Character	-	-	Carriage return and line feed.

Example:

NMEA 0183 V3.01 example:

```
$GNRMC,073551.000,A,3149.333056,N,11706.945606,E,0.00,0.00,230222,,,A*70
```

NMEA 0183 V4.10 example:

```
$GNRMC,073925.000,A,3149.333680,N,11706.947520,E,0.08,000.00,230222,,,D,V*00
```

2.2.2. GGA

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

Type:

Output

Synopsis:

NMEA 0183 V3.01 format:

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

NMEA 0183 V4.10 format (default):

```
$<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	-	073925.000	Position fix UTC. hh: Hours (00–23) mm: Minutes (00–59) ss: Seconds (00–59) sss: Decimal fraction of seconds
<Lat>	ddmm.mmmmmm	-	3149.333680	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	-	11706.947520	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.

Field	Format	Unit	Example	Description
<E/W>	Character	-	E	East-west direction. E = East W = West Note that this field is empty in case of an invalid value.
<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. 3 = GPS PPS Mode, fix valid. 6 = Estimated (dead reckoning) mode.
<NumSatUsed> 1)	Numeric, 2 digits	-	39	Number of satellites in use.
<HDOP>	Numeric	-	0.46	Horizontal dilution of precision. Note that this field is empty in case of an invalid value.
<Alt>	Numeric	Meter	62.014	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	-0.334	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	-	*55	Checksum
<CR><LF>	Character	-	-	Carriage return and line feed.

Example:

NMEA 0183 V3.01 example:

```
$GNGGA,073551.000,3149.333056,N,11706.945606,E,1,36,0.52,62.883,M,-0.335,M,,*5C
```

NMEA 0183 V4.10 example:

```
$GNGGA,073925.000,3149.333680,N,11706.947520,E,2,39,0.46,62.014,M,-0.334,M,,*55
```

NOTE

1. The NMEA 0183 specification indicates that the **GGA** messages are GPS specific. However, when the receiver is configured for multi-constellation operation, the content of **GGA** messages is 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 (SV) in view, 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:

NMEA 0183 V3.01 format:

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

NMEA 0183 V4.10 format (default):

```
$<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 **$**.

Field	Format	Unit	Example	Description
<TalkerID>	String, 2 characters	-	GP	Talker identifier. See *Table 2: NMEA Talker ID*.
GSV	String, 3 characters	-	GSV	GNSS Satellites in View
<TotalNumSen>	Numeric	-	5	Total number of sentences. Range: 1–9.
<SenNum>	Numeric	-	1	Sentence number. Range: 1–**<TotalNumSen>**.
<TotalNumSat>	Numeric	-	17	Total number of satellites in view.
Start of repeat block. Repeat times: 1–4.				
<SatID>	Numeric	-	195	Satellite ID. See *Table 8: GNSS Satellites (NMEA) Numbering*.
<SatElev>	Numeric	Degree	70	Satellite elevation. Range: 00–90. Note that this field is empty in case of an invalid value.
<SatAz>	Numeric	Degree	093	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	41	Satellite C/N_0. Range: 00–99. Null when not tracking.
End of repeat block.				
<SignalID>	Numeric	-	1	GNSS Signal ID. See *Table 8: GNSS Satellites (NMEA) Numbering*. Please note that this parameter is only available in messages in line with NMEA 0183 V4.10 or higher.
<Checksum>	Hexadecimal	-	*61	Checksum
<CR><LF>	Character	-	-	Carriage return and line feed.

Example:

NMEA 0183 V3.01 example:

```
$GPGSV,5,1,17,195,70,095,44,194,64,093,38,21,63,128,40,07,61,295,49*78
$GPGSV,5,2,17,08,58,024,47,199,51,162,39,01,40,174,44,30,35,312,45*41
```

```
$GPGSV,5,3,17,27,26,046,42,16,18,091,22,196,16,157,38,09,15,226,37*44
$GPGSV,5,4,17,14,10,296,41,49,05,262,,04,05,194,34,17,03,239,28*71
$GPGSV,5,5,17,193,03,160,36*75
$GAGSV,2,1,08,01,63,055,46,33,57,170,45,26,55,065,45,31,40,318,44*6A
$GAGSV,2,2,08,04,12,230,23,09,11,281,37,12,11,201,24,13,08,035,33*6B
$GIGSV,2,1,05,04,83,208,39,03,36,231,34,02,24,279,32,07,,,37*52
$GIGSV,2,2,05,05,,,29*6B
```

NMEA 0183 V4.10 example:

```
$GPGSV,5,1,17,195,70,093,41,21,64,125,40,194,64,095,37,07,62,292,49,1*61
$GPGSV,5,2,17,08,57,025,48,199,51,162,38,45,45,219,39,01,41,174,45,1*58
$GPGSV,5,3,17,30,38,310,44,27,25,046,41,16,17,092,23,196,16,157,,1*5A
$GPGSV,5,4,17,09,14,225,29,14,10,297,,04,04,193,30,193,03,159,36,1*51
$GPGSV,5,5,17,17,03,240,33,1*51
$GAGSV,3,1,09,01,63,060,46,33,58,170,46,26,54,062,44,31,41,319,45,7*7D
$GAGSV,3,2,09,21,22,103,24,12,13,201,22,04,11,229,28,09,11,279,37,7*78
$GAGSV,3,3,09,13,06,035,36,7*4D
$GIGSV,1,1,04,04,83,206,28,03,36,231,19,02,24,280,,07,,,27,1*4A
```

NOTE

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

2.2.4. GSA

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

Type:

Output

Synopsis:

NMEA 0183 V3.01 format:

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

NMEA 0183 V4.10 format (default):

```
$<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 mode. A = Automatic, allowed to automatically switch 2D/3D.
<FixMode>	Numeric	-	3	Fix mode. 1 = Fix not available 2 = 2D fix 3 = 3D fix
Start of repeat block. Repeat times: 12.				
<SatID>	Numeric	-	195	ID numbers of satellites used in solution. See *Table 8: GNSS Satellites (NMEA) Numbering*. Note that this field is empty in case of an invalid value.
End of repeat block.				
<PDOP>	Numeric	-	0.70	Position dilution of precision. Maximum value: 99.99. Note that this field is empty in case of an invalid value.
<HDOP>	Numeric	-	0.46	Horizontal dilution of precision. Maximum value: 99.99. Note that this field is empty in case of an invalid value.
<VDOP>	Numeric	-	0.53	Vertical dilution of precision. Maximum value: 99.99. Note that this field is empty in case of an invalid value.
<SystemID>	Numeric	-	1	GNSS system ID. See *Table 8: GNSS Satellites (NMEA) Numbering*. Please note that this parameter is only available in messages in line with NMEA

Field	Format	Unit	Example	Description
				0183 V4.10 or higher.
<Checksum>	Hexadecimal	-	*37	Checksum
<CR><LF>	Character	-	-	Carriage return and line feed.

Example:

NMEA 0183 V3.01 example:

```
$GPGSA,A,3,195,194,21,07,08,199,01,30,27,09,14,04,0.84,0.52,0.66*3A
$GAGSA,A,3,01,33,26,31,04,09,12,13,,,,,0.84,0.52,0.66*13
$GIGSA,A,3,04,03,02,,,,,,,,,,0.84,0.52,0.66*15
```

NMEA 0183 V4.10 example:

```
$GNGSA,A,3,195,21,194,07,08,199,01,30,27,16,09,,0.70,0.46,0.53,1*37
$GNGSA,A,3,01,33,26,31,12,04,09,13,,,,,0.70,0.46,0.53,3*0B
$GNGSA,A,3,04,03,,,,,,,,,,,0.70,0.46,0.53,6*02
```

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:

NMEA 0183 V3.01 format:

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

NMEA 0183 V4.10 format (default):

```
$<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	0.00	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	0.08	Speed over ground in knots. Note that this field is empty in case of an invalid value.
N	Character	-	N	Fixed field: knot.
<SOGK>	Numeric	km/h	0.14	Speed over ground in kilometers per hour. Variable length. Note that this field is empty in case of an invalid value.
K	Character	-	K	Fixed field: kilometers per hour.
<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 = Data not valid
<Checksum>	Hexadecimal	-	*2B	Checksum
<CR><LF>	Character	-	-	Carriage return and line feed.

Example:

NMEA 0183 V3.01 example:

```
$GNVTG,0.00,T,,M,0.00,N,0.01,K,A*22
```

NMEA 0183 V4.10 example:

```
$GNVTG,0.00,T,,M,0.08,N,0.14,K,D*2B
```

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:

NMEA 0183 V3.01 format:

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

NMEA 0183 V4.10 format (default):

```
$<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*.
GLL	String, 3 characters	-	GLL	Geographic Position – Latitude/Longitude
<Lat>	ddmm.mmmmmm	-	3149.333680	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

Field	Format	Unit	Example	Description
				S = South Note that this field is empty in case of an invalid value.
<Lon>	dddmm.mmmmmm	-	11706.947520	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	-	073925.000	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	-	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 = Data not valid
<Checksum>	Hexadecimal	-	*46	Checksum
<CR><LF>	Character	-	-	Carriage return and line feed

Example:

NMEA 0183 V3.01 example:

```
$GNGLL,3149.333056,N,11706.945606,E,073551.000,A,A*44
```

NMEA 0183 V4.10 example:

```
$GNGLL,3149.333680,N,11706.947520,E,073925.000,A,D*46
```

2.2.7. ZDA

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

Type:

Output

Synopsis:

NMEA 0183 V3.01 format:

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

NMEA 0183 V4.10 format (default):

```
$<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	-	081531.000	Position fix UTC. hh: Hours (00–23) mm: Minutes (00–59) ss: Seconds (00–59) sss: Decimal fraction of seconds
<Day>	Numeric	-	25	Day of month. Range: 01–31.
<Month>	Numeric	-	08	Month. Range: 01–12.
<Year>	Numeric	-	2022	Year.

Field	Format	Unit	Example	Description
<LocalHour>	Numeric	-	-	Local zone hours, 00 to ±13 hours. Not supported. Always null.
<LocalMin>	Numeric	-	-	Local zone minutes, 00 to 59 minutes. Not supported. Always null.
<Checksum>	Hexadecimal	-	*4B	Checksum
<CR><LF>	Character	-	-	Carriage return and line feed.

Example:

NMEA 0183 V3.01 example:

```
$GNZDA,082250.000,25,08,2022,,*48
```

NMEA 0183 V4.10 example:

```
$GNZDA,081531.000,25,08,2022,,*4B
```

2.2.8. 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:

NMEA 0183 V3.01 format:

```
$<TalkerID>GRS,<UTC>,<Mode>{,<Resi>}*<Checksum><CR><LF>
```

NMEA 0183 V4.10 format (default):

```
$<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	-	GN	Talker identifier. See *Table 2: NMEA Talker ID*.

Field	Format	Unit	Example	Description
GRS	String, 3 characters	-	GRS	GNSS range residuals.
<UTC>	hhmmss.sss	-	081531.000	Position fix UTC. hh: Hours (00–23) mm: Minutes (00–59) ss: Seconds (00–59) sss: Decimal fraction of seconds
<Mode>	Numeric	-	1	Computation method used. 0 = Residuals were used to calculate the position given in the matching **GGA** or **GNS** sentence. 1 = Residuals were recomputed after the **GGA** or **GNS** position was computed.
Start of repeat block. Repeat time: 12.				
<Resi>	Numeric	m	-0.8	Range residuals for SVs 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 8: GNSS Satellites (NMEA) Numbering*. Please note that this parameter is only available in messages in line with NMEA 0183 V4.10 or higher.
<SignalID>	Numeric	-	1	GNSS signal ID. See *Table 8: GNSS Satellites (NMEA) Numbering*. Please note that this parameter is only available in messages in line with NMEA 0183 V4.10 or higher.
<Checksum>	Hexadecimal	-	*4C	Checksum
<CR><LF>	Character	-	-	Carriage return and line feed.

Example:

NMEA 0183 V3.01 example:

```
$GNGRS,082250.000,1,-9.8,-9.3,-6.7,,,,,,,,*64
$GNGRS,082250.000,1,-10.9,-15.0,-8.1,-8.2,-11.3,,,,,,,*52
```

$GNGRS,082250.000,1,-1.0,,,,,,,,,,*6F

NMEA 0183 V4.10 example:

$GNGRS,081531.000,1,-0.8,2.5,0.4,1.7,2.0,1.3,4.1,2.5,,,,,1,1*4C
$GNGRS,081531.000,1,1.2,2.7,-0.2,1.1,5.7,,,,,,,,3,7*6F
$GNGRS,081531.000,1,-1.9,20.4,,,,,,,,,,6,1*7A

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, multiple **GRS** sentences containing all **<Resi>** fields will be output.

2.2.9. GST

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

Type:

Output

Synopsis:

NMEA 0183 V3.01 format:

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

NMEA 0183 V4.10 format (default):

$<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*.

Field	Format	Unit	Example	Description
GST	String, 3 characters	-	GST	GNSS Pseudorange Error Statistics.
<UTC>	hhmmss.sss	-	081531.000	Position fix UTC. hh: Hours (00–23) mm: Minutes (00–59) ss: Seconds (00–59) sss: Decimal fraction of seconds
<RMS_D>	Numeric	Meter	3.5	RMS value of the standard deviation of the range inputs to the navigation process.
<MajorD>	Numeric	Meter	4.3	Standard deviation of semi-major axis of error ellipse.
<MinorD>	Numeric	Meter	2.5	Standard deviation of semi-minor axis of error ellipse.
<Orient>	Numeric	Degree	148.7	Orientation of semi-major axis of error ellipse.
<LatD>	Numeric	Meter	3.9	Standard deviation of latitude error.
<LonD>	Numeric	Meter	3.1	Standard deviation of longitude error.
<AltD>	Numeric	Meter	11.4	Standard deviation of altitude error.
<Checksum>	Hexadecimal	-	*47	Checksum
<CR><LF>	Character	-	-	Carriage return and line feed.

Example:

NMEA 0183 V3.01 example:

```
$GNGST,082250.000,5.7,6.9,2.0,143.2,5.7,4.5,18.0*45
```

NMEA 0183 V4.10 example:

```
$GNGST,081531.000,3.5,4.3,2.5,148.7,3.9,3.1,11.4*47
```

2.3. PQTM Messages

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

Table 3: Error Codes

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

2.3.1. PQTMANTENNASTATUS

Outputs antenna status.

Type:

Output

Synopsis:

```
$PQTMANTENNASTATUS,<Status>,<Mode>,<Power>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Status>	Numeric	-	Antenna status. 0 = Normal 1 = Open circuit 2 = Short-circuited
<Mode>	Numeric	-	Antenna operation mode. 0 = Automatic 1 = Internal antenna (patch antenna) 2 = External antenna
<Power>	Numeric	-	External antenna power status. 0 = Power off 1 = Power on

Example:

```
$PQTMANTENNASTATUS,0,0,0*4F
```

NOTE

The antenna status output by this message reflects the status of the internal antenna when the antenna operation mode is manually configured to internal antenna. In other cases, the antenna status output

indicates the status of the external antenna.

2.3.2. PQTMCFGANTENNA

Sets/gets antenna operation mode.

Type:

Set/Get

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<R/W>	Numeric	-	Read/write configuration. 0 = Read 1 = Write
<Mode>	Numeric	-	Antenna operation mode. 0 = Automatic 1 = Internal antenna (patch antenna) 2 = External antenna

Result:

- If successful, the module returns:

```
//Response to Set command:
$PQTMCFGANTENNAOK*<Checksum><CR><LF>
//Response to Get command:
$PQTMCFGANTENNA,<R/W>,<Mode>*47
```

- If failed, the module returns:

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

Example:

```
//Set antenna to automatic mode:
$PQTMCFGANTENNA,1,0*04
```

```
//Response to set command:
$PQTMCFGANTENNAOK*01

//Get antenna operation mode:
$PQTMCFGANTENNA,0*19
//Response to get command:
$PQTMCFGANTENNA,0,0*47
```

NOTE

If the default value is not given for any parameter in a Set command, you can query it with the corresponding Get command if the default setting has not been changed. If the default setting had been changed with the Set command, contact Quectel Technical Support (support@quectel.com) for the default setting.

2.3.3. 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. yyyy: Year mm: Month

Field	Format	Unit	Description
			dd: Day of month
<BuildTime>	hh:mm:ss	-	Firmware build time. hh: Hours mm: Minutes ss: Seconds

- If failed, the module returns:

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

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

Example:

```
$PQTMVERNO*58
$PQTMVERNO,L89HANR01A06S,2022/07/28,18:27:04*3E
```

2.3.4. PQTMVER

Outputs the firmware version.

Type:

Output

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<MsgVer>	Numeric	-	Message version. 1 = Version 1 (Always 1 for this message version).
<VerName>	String	-	Version name. Fixed as “MODULE”.
<VerStr>	String	-	Version string.
<BuildDate>	yyyy/mm/dd	-	Firmware build date. yyyy: Year mm: Month dd: Day of month

Field	Format	Unit	Description
<BuildTime>	hh:mm:ss	-	Firmware build time. hh: Hours mm: Minutes ss: Seconds

Example:

```
$PQTMVER,1,MODULE,L89HANR01A08S,2024/12/06,15:53:42*18
```

NOTE

The message is output immediately upon module power-up and is output only once.

2.3.5. 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>
```

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

Example:

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

2.3.6. 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
```

2.3.7. 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 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,*75

//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

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
GRS	1–20
GST	1–20
PQTMANTENNASTATUS	1–20
PQTMEPE	1–20
PQTMGEOFENCESTATUS	1–20
PQTMPL	1-20
PQTMPVT	1–20
PQTMDOP	1–20
PQTMVEL	1–20
PQTMODO	1–20
PQTMJAMMINGSTATUS	1–20
PQTMLS	1–20

NOTE

If the configuration message is a **PQTM** message, use **<MsgVer>** field to specify the message version, otherwise an error will be returned.

2.3.8. 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 message 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
```

2.3.9. 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	Millisecond	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	The protection level of time. Null if invalid.

Example:

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

2.3.10. 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 function 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 and trapezium.)
<Lat0>	Numeric	Degree	Latitude of the first point.
<Lon0>	Numeric	Degree	Longitude of the first point.
<Lat1/Radius>	Numeric	Degree or Meter	If the geofence shape is a circle with a certain radius, this value will be the radius of the circle, otherwise, this value will be the latitude of the second point.
<Lon1>	Numeric	Degree	Longitude of the second point.
<Lat2>	Numeric	Degree	Latitude of the third point.
<Lon2>	Numeric	Degree	Longitude of the third point.
<Lat3>	Numeric	Degree	Latitude of the fourth point.
<Lon3>	Numeric	Degree	Longitude of the fourth point.

Result:

- If successful, the module returns:

```
//Response to Set command:
```

```
$PQTMCFGGEOFENCE,OK*<Checksum><CR><LF>
//Response to Get command:
$PQTMCFGGEOFENCE,OK,<Index>,<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,0,1,0,0,31.451248,117.451245,100.5*4B
```

2.3.11. PQTMGEOFENCESTATUS

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

Field	Format	Unit	Description
Start of repeat block. Repeat times: 4.			
<StateN>	Numeric	-	Geofence state. N is the number of **<State>**. Range of N: 0–3. 0 = Unknow 1 = Inside the geofence 2 = Outside the geofence Note that if the module has not achieved a fix, the **<StateN>** is 0.
End of repeat block.			

Example:

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

2.3.12. 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 message version.)
<TOW>	Numeric	Millisecond	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

Field	Format	Unit	Description
<Res>	Numeric	-	Reserved. A|ways 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. 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. Note that this field is empty in case of an invalid value. Range: 0.00–360.00.

Field	Format	Unit	Description
<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:

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

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

2.3.13. 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 NMEA standard messages. Range: 0–3. Default value: 3. 0 = No fractional part.
<POS_DP>	Numeric	-	Number of decimal places for latitude and longitude in NMEA standard messages.

Field	Format	Unit	Description
			Range: 0–8. Default value: 6. 0 = No fractional part.
<ALT_DP>	Numeric	-	Number of decimal places for altitude and geoidal separation in NMEA standard messages. Range: 0–3. Default value: 3. 0 = No fractional part.
<DOP_DP>	Numeric	-	Number of decimal places for DOP in NMEA standard messages. Range: 0–3. Default value: 2. 0 = No fractional part.
<SPD_DP>	Numeric	-	Number of decimal places for speed in NMEA standard messages. Range: 0–3. Default value: 2. 0 = No fractional part.
<COG_DP>	Numeric	-	Number of decimal places for COG in NMEA standard messages. Range: 0–3. Default value: 2. 0 = No fractional part.

Result:

- If successful, the module returns:

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

- If failed, the module returns:

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

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

Example:

```
//Set:
$PQTMCFGNMEADP,W,3,6,3,2,2,2*36
$PQTMCFGNMEADP,OK*61

//Get:
$PQTMCFGNMEADP,R*37
```

$PQTMCFGNMEADP,OK,3,6,3,2,2,2*65

2.3.14. PQTMCOLD

Performs a cold start, which restarts 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

2.3.15. 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
```

2.3.16. 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
```

2.3.17. PQTMSRR

Performs a system reset and reboots the receiver.

Type:

Command

Synopsis:

```
$PQTMSRR*<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:

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

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

Example:

```
$PQTMSRR*4B
```

2.3.18. 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. Bit 0 = WAAS Bit 2 = EGNOS 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
```

2.3.19. 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:
   - GPS + GLONASS + Galileo + BDS + NavIC + QZSS
   - GPS + GLONASS + Galileo + BDS + QZSS
   - GPS + GLONASS + Galileo + QZSS
   - GPS + GLONASS + BDS + QZSS
   - GPS + Galileo + NavIC + QZSS
   - GPS + Galileo + BDS + QZSS
   - GPS + GLONASS + QZSS
   - GPS + Galileo + QZSS
   - GPS + BDS + QZSS
   - GPS + QZSS
   - BDS
   - NavIC

   Note that QZSS constellation can be enabled or disabled in the constellation configurations supporting QZSS.
2. The configuration set by this command takes effect after executing **PQTMSAVEPAR** and then restarting the module.

2.3.20. 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	Millisecond	Time of week. Null if invalid.
<GDOP>	Numeric	-	Geometric dilution of precision. Note that the value is 99.99 in case of an invalid value.
<PDOP>	Numeric	-	Position (3D) dilution of precision. Note that the value is 99.99 in case of an invalid value.
<TDOP>	Numeric	-	Time dilution of precision. Note that the value is 99.99 in case of an invalid value.
<VDOP>	Numeric	-	Vertical dilution of precision. Note that the value is 99.99 in case of an invalid value.
<HDOP>	Numeric	-	Horizontal dilution of precision. Note that the value is 99.99 in case of an invalid value.
<NDOP>	Numeric	-	Northing dilution of precision. Not supported. Always null.
<EDOP>	Numeric	-	Easting dilution of precision. Not supported. Always null.

Example:

```
//Fixed:
$PQTMDOP,1,366967972,0.98,0.86,0.46,0.65,0.56,,*7A

//Not fixed:
$PQTMDOP,1,,99.99,99.99,99.99,99.99,99.99,,*70
```

2.3.21. 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	Millisecond	Position fix interval.

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
```

2.3.22. PQTMCFGPPS

Sets/gets the 1PPS feature configuration.

Type:

Set/Get

Synopsis:

```
//Set:
$PQTMCFGPPS,W,<Index>,<Enable>,<Duration>,<Mode>,<Polarity>,<Interval>*<Checksum><CR><L
```

```
F>
//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	Millisecond	Pulse duration. Range: 1–999. Default value: 100.
<Mode>	Numeric	-	PPS output with fix mode. 1 = Always 2 = 2D fix 3 = 3D fix
<Polarity>	Numeric	-	Pulse polarity. 0 = Low 1 = High
<Interval>	Numeric	-	PPS generated interval. 0 = Every second 1 = Odd seconds 2 = Even seconds

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>,<Interval>*<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,1,1,100,3,0,0*70
$PQTMCFGPPS,OK*21

//Get:
$PQTMCFGPPS,R,1*6A
$PQTMCFGPPS,OK,1,1,100,3,0,0*23

2.3.23. PQTMDEBUGON

Enables debug logging. The debug-on state can be saved by **PQTMSAVEPAR** command.

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

2.3.24. PQTMDEBUGOFF

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

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
```

2.3.25. PQTMVEL

Outputs the velocity information.

Type:

Output

Synopsis:

```
$PQTMVEL,1,<Time>,<VelN>,<VelE>,<VelD>,<GrdSpd>,<Spd>,<Heading>,<GrdSpdAcc>,<SpdAcc>,<HeadingAcc>*<Checksum><CR><LF>
```

Parameter:

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

Field	Format	Unit	Description
<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
```

2.3.26. 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

Field	Format	Unit	Description
<InitDist>	Numeric	Meter	Initial distance. Default value: 0.

Result:

- If successful, the module returns:

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

- If failed, the module returns:

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

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

Example:

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

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

2.3.27. 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.

2.3.28. 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 message version.)
<Time>	hhmmss.sss	-	UTC time. hh: Hour (00–23) mm: Minute (00–59) ss: Second (00–59) sss: Decimal fraction of seconds

Field	Format	Unit	Description
<State>	Numeric	-	Odometer status. 0 = Disabled 1 = Enabled
<Dist>	Numeric	Meter	Distance since last reset.

Example:

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

NOTE

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

2.3.29. 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 3 = Critical status

Example:

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

2.3.30. 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 message version.)
<TOW>	Numeric	s	Time of week.
<LS_Ref>	Hexadecimal	-	Referenced constellation by the current leap second information. 0 = No source 1 = GPS 2 = GLONASS 3 = Galileo 4 = BDS
<WN>	Numeric	-	UTC reference week number.
<LS>	Numeric	Second	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>	Hexadecimal		Referenced constellation by the leap second forecast information. 0 = No source

Field	Format	Unit	Description
			1 = GPS 2 = GLONASS 3 = Galileo 4 = BDS The field value is invalid if **<Flag>** = 0.
<Reserved>	-	-	Reserved. Always null.
<WNLSF>	Numeric	-	Week number of the new leap second. The field value is invalid if **<Flag>** = 0.
<DN>	Numeric	-	Day of the week when the new leap second takes effect. GPS & Galileo: 1 to 7 from Sunday to Saturday; BDS: 0 to 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.

2.3.31. PQTMCFGNMEATID

Sets/gets the NMEA Talker ID.

Type:

Set/Get

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<Main_TalkerID>	Character	-	The main Talker ID, which is used for all 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", you can set a user-defined two-character talker ID.
<GSV_TalkerID>	Numeric	-	**GSV** Talker ID. 0 = Determined by the GNSS constellation configuration 1 = Same value as the **<Main_TalkerID>**

Result:

- If successful, the module returns:

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

- If failed, the module returns:

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

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

Example:

```
//Set:
$PQTMCFGNMEATID,W,GP,0*58
$PQTMCFGNMEATID,OK*2C
//Get:
$PQTMCFGNMEATID,R*7A
$PQTMCFGNMEATID,OK,GP,0*0B
```

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

2.3.32. 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
<Length>	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,16,81D62010EE0AF375BDF5952CDC3757A1*3E

2.3.33. PQTMCFGUART

Sets/gets the UART port.

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 of 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 = UART1
<BaudRate>	Numeric	bps	UART baud rate. 9600 19200 38400 57600 115200 230400 460800 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	Bit	Stop bit(s). 1 = 1 stop bit

Field	Format	Unit	Description
			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:

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

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

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

//Set 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

1. The configuration made by this command takes effect after executing **PQTMSAVEPAR** and then restarting the module.
2. When the baud rate is set to 9600 bps, the command response and NMEA message output may be delayed due to high data volume.

2.3.34. PQTMCFGPROT

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

Type:

Set/Get

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<PortType>	Numberic	-	Port type. 1 = UART 2 = I2C
<PortID>	Numberic	-	Port ID. 1 = Port 1 When **<PortType>** = 1, the specified port is UART. When **<PortType>** = 2, the specified port is I2C.
<InputProt>	Hexadecimal	-	Input protocol. Bit 0 = NMEA Bit 1 = Reserved. Always “0”.

Field	Format	Unit	Description
			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,1,1,5,5*38
$PQTMCFGPROT,OK*6B

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

2.3.35. PQTMBKP

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

The module can exit Backup mode either via the WAKEUP pin or the timer. When the module exits Backup mode, all system resources are re-initialized.

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 the value is 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 (**<Second>** ≥ 10); otherwise, the module 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
```

2.3.36. 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
```

2.3.37. 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
```

2.4. PAIR Messages

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

PAIR Message Format:

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

Packet Type: Three-byte character string, from 000 to 999. An identifier for each PAIR 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.

2.4.1. PAIR001: PAIR_ACK

Acknowledges a **PAIR** command. An acknowledgement packet **$PAIR001** is returned to inform the sender that the receiver has received the packet.

Type:

Output

Synopsis:

```
$PAIR001,<CommandID>,<Result>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<CommandID>	Numeric	-	Type of command/packet to be acknowledged.
<Result>	Numeric	-	0 = Command has been successfully sent. 1 = Command is being processed. Please wait for the result. 2 = Command sending failed. 3 = <CommandID> is not supported. 4 = Command parameter error. Out of range/Some parameters

Field	Format	Unit	Description
			were lost/Checksum error. 5 = MNL service is busy. You can try again soon.

Example:

```
$PAIR001,006,0*3D
```

2.4.2. PAIR002: PAIR_GNSS_SUBSYS_POWER_ON

Powers on the GNSS system, including DSP, RF, PE and Clock.

Type:

Command

Synopsis:

```
$PAIR002*<Checksum><CR><LF>
```

Parameter:

None

Result:

Returns **$PAIR001** message.

Example:

```
$PAIR002*38
$PAIR001,002,1*38
$PAIR001,002,0*39
```

2.4.3. PAIR003: PAIR_GNSS_SUBSYS_POWER_OFF

Powers off the GNSS system, including DSP, RF, PE and clock.

Type:

Command

Synopsis:

```
$PAIR003*<Checksum><CR><LF>
```

Parameter:

None

Result:

Returns **$PAIR001** message.

Example:

```
$PAIR003*39
$PAIR001,003,1*39
$PAIR001,003,0*38
```

2.4.4. PAIR004: PAIR_GNSS_SUBSYS_HOT_START

Performs a hot start (uses all available data in the NVRAM). Normally a hot start means that the GNSS module has been powered down for less than 2 hours (RTC must be alive) with its ephemeris still valid. Therefore, there is no need to download the ephemeris data again upon a hot start, thus making this startup method the fastest.

Type:

Command

Synopsis:

```
$PAIR004*<Checksum><CR><LF>
```

Parameter:

None

Result:

Returns **$PAIR001** message.

Example:

```
$PAIR004*3E
$PAIR001,004,0*3F
```

2.4.5. PAIR005: PAIR_GNSS_SUBSYS_WARM_START

Performs a warm start. A warm start means that the GNSS module remembers only rough time, position, and almanac data, and thus needs to download the ephemeris data before it can fix a position.

Type:

Command

Synopsis:

```
$PAIR005*<Checksum><CR><LF>
```

Parameter:

None

Result:

Returns **$PAIR001** message.

Example:

```
$PAIR005*3F
$PAIR001,005,0*3E
```

2.4.6. PAIR006: PAIR_GNSS_SUBSYS_COLD_START

Performs a cold start, which means that no location information is stored in the receiver, including time, position, and almanacs and ephemeris data.

Type:

Command

Synopsis:

```
$PAIR006*<Checksum><CR><LF>
```

Parameter:

None

Result:

Returns **$PAIR001** message.

Example:

```
$PAIR006*3C
$PAIR001,006,0*3D
```

2.4.7. PAIR007: PAIR_GNSS_SUBSYS_FULL_COLD_START

Performs a cold start and clears system and user configurations at the start, i.e., resets the module to its factory settings. Upon a full cold start, the module loses all data on the previous position. Therefore, it needs to search over the full frequency spectrum for all visible satellites before it can fix a position.

Type:

Command

Synopsis:

```
$PAIR007*<Checksum><CR><LF>
```

Parameter:

None

Result:

Returns **$PAIR001** message.

Example:

```
$PAIR007*3D
$PAIR001,007,0*3C
```

2.4.8. PAIR010: PAIR_REQUEST_AIDING

Notifies the expiration of GNSS aiding data stored in the module. This message is automatically output when the module powers on.

Type:

Output

Synopsis:

```
$PAIR010,<Type>,<GNSS_System>,<WN>,<TOW>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Type>	Numeric	-	Type of data to be updated. 0 = EPO data 1 = Time 2 = Location

Field	Format	Unit	Description
<GNSS_System>	Numeric	-	Type of required GNSS data. 0 = GPS data 1 = GLONASS data 2 = Galileo data 3 = BDS data 4 = QZSS data
<WN>	Numeric	Week	Week Number (accommodating roll-over).
<TOW>	Numeric	Second	Time of Week.

Example:

```
$PAIR010,0,0,2044,369413*33
```

NOTE

The GNSS system outputs this message automatically. Do not send **$PAIR010** manually.

2.4.9. PAIR011: PAIR_INDICATION_SYSTEM_MESSAGE

GNSS system message notification.

Type:

Output

Synopsis:

```
$PAIR011,<Type>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Type>	Numeric	-	001 = Notification for GNSS system startup

Example:

```
$PAIR011,001*27
```

NOTE

The GNSS system outputs this message automatically. Do not send **PAIR011** manually.

2.4.10. PAIR050: PAIR_COMMON_SET_FIX_RATE

Sets position fix interval.

Type:

Set

Synopsis:

```
$PAIR050,<Time>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Time>	Numeric	Millisecond	Position fix interval. Range: 100–1000. Default value: 1000.

Result:

Returns **$PAIR001** message.

Example:

```
$PAIR050,1000*12
$PAIR001,050,0*3E
```

NOTE

1. 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) for the default setting.
2. This command is supported on L89 (HA) with L89HANR01A05S or higher versions.

2.4.11. PAIR051: PAIR_COMMON_GET_FIX_RATE

Gets the position fix interval.

Type:

Get

Synopsis:

```
$PAIR051*<Checksum><CR><LF>
```

Parameter:

None

Result:

Returns **$PAIR001** message and the query result.

Query result message format:

```
$PAIR051,<Time>*<Checksum><CR><LF>
```

Parameter included in the result:

Field	Format	Unit	Description
<Time>	Numeric	Millisecond	Position fix interval. Range: 100–1000. Default value: 1000.

Example:

```
$PAIR051*3E
$PAIR001,051,0*3F
$PAIR051,1000*13
```

NOTE

This command is supported on L89 (HA) with L89HANR01A05S or higher versions.

2.4.12. PAIR058: PAIR_COMMON_SET_MIN_SNR

Sets the minimum SNR of satellites in use. If the minimum SNR threshold is set, the module will not use the satellites with SNR below the threshold.

Type:

Set

Synopsis:

```
$PAIR058,<MIN_SNR>*<Checksum><CR><LF>
```

Parameter:

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

Result:

Returns **$PAIR001** message.

Example:

```
$PAIR058,15*1F
$PAIR001,058,0*36
```

2.4.13. PAIR059: PAIR_COMMON_GET_MIN_SNR

Gets the minimum SNR of satellites in use.

Type:

Get

Synopsis:

```
$PAIR059*<Checksum><CR><LF>
```

Parameter:

None

Result:

Returns **$PAIR001** message and the query result.

Query result message format:

```
$PAIR059,<MIN_SNR>*<Checksum><CR><LF>
```

Parameter included in the result:

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

Example:

```
$PAIR059*36
$PAIR001,059,0*37
$PAIR059,15*1E
```

2.4.14. PAIR062: PAIR_COMMON_SET_NMEA_OUTPUT_RATE

Sets the output rate of standard NMEA sentences of each type.

Type:

Set

Synopsis:

```
$PAIR062,<Type>,<OutputRate>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Type>	Numeric	-	Type of standard NMEA sentence. -1 = Reset the output rates of all types of sentences to default values 0 = NMEA_SEN_GGA 1 = NMEA_SEN_GLL 2 = NMEA_SEN_GSA 3 = NMEA_SEN_GSV 4 = NMEA_SEN_RMC 5 = NMEA_SEN_VTG 6 = NMEA_SEN_ZDA 7 = NMEA_SEN_GRS 8 = NMEA_SEN_GST
<OutputRate>	Numeric	-	Message output rate setting. 0 = Disabled or not supported N = Output message once every N position fix(es) Range of N: 1–20.

Result:

Returns **$PAIR001** message.

Example:

```
$PAIR062,0,3*3D
$PAIR001,062,0*3F
```

2.4.15. PAIR063: PAIR_COMMON_GET_NMEA_OUTPUT_RATE

Gets the output rate of standard NMEA sentences of each type.

Type:

Get

Synopsis:

```
$PAIR063,<Type>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Type>	Numeric	-	Type of standard NMEA sentence. -1 = Return the output rates of all types of standard NMEA sentences 0 = NMEA_SEN_GGA 1 = NMEA_SEN_GLL 2 = NMEA_SEN_GSA 3 = NMEA_SEN_GSV 4 = NMEA_SEN_RMC 5 = NMEA_SEN_VTG 6 = NMEA_SEN_ZDA 7 = NMEA_SEN_GRS 8 = NMEA_SEN_GST

Result:

Returns **$PAIR001** message and the query result.

Query result message format:

```
$PAIR063,<Type>,<OutputRate>*<Checksum><CR><LF>
```

Parameters included in the result:

Field	Format	Unit	Description
<Type>	Numeric	-	Type of standard NMEA sentence. 0 = NMEA_SEN_GGA 1 = NMEA_SEN_GLL 2 = NMEA_SEN_GSA 3 = NMEA_SEN_GSV 4 = NMEA_SEN_RMC 5 = NMEA_SEN_VTG 6 = NMEA_SEN_ZDA 7 = NMEA_SEN_GRS 8 = NMEA_SEN_GST
<OutputRate>	Numeric	-	Message output rate setting. 0 = Disabled or not supported N = Output message once every N position fix(es) Range of N: 1–20.

Example:

```
$PAIR063,0*23
$PAIR001,063,0*3E
$PAIR063,0,3*3C
```

2.4.16. PAIR066: PAIR_COMMON_SET_GNSS_SEARCH_MODE

Sets the GNSS search mode. The setting is valid if the NVRAM data are valid.

Type:

Set

Synopsis:

```
$PAIR066,<GPS_Enabled>,<GLONASS_Enabled>,<Galileo_Enabled>,<BDS_Enabled>,<QZSS_Enabled>,<NavIC_Enabled>*<Checksum><CR><LF>
```

Parameter:

Packet Data	Format	Unit	Description
<GPS_Enabled>	Numeric	-	0 = Disable (DO NOT search for GPS satellites) 1 = Search for GPS satellites
<GLONASS_Enabled>	Numeric	-	0 = Disable (DO NOT search for GLONASS satellites) 1 = Search for GLONASS satellites

Packet Data	Format	Unit	Description
<Galileo_Enabled>	Numeric	-	0 = Disable (DO NOT search for Galileo satellites) 1 = Search for Galileo satellites
<BDS_Enabled>	Numeric	-	0 = Disable (DO NOT search for BDS satellites) 1 = Search for BDS satellites
<QZSS_Enabled>	Numeric	-	0 = Disable (DO NOT search for QZSS satellites) 1 = Search for QZSS satellites
<NavIC_Enabled>	Numeric	-	0 = Disable (DO NOT search for NavIC (IRNSS) satellites) 1 = Enable (Search for NavIC (IRNSS) satellites)

Result:

Returns **$PAIR001** message.

Example:

```
//Search for GPS satellites only:
$PAIR066,1,0,0,0,0,0*3B
$PAIR001,066,0*3B
```

NOTE

1. QZSS is always enabled by default.
2. Supported GNSS search modes:
   - GPS + GLONASS + Galileo + BDS + NavIC + QZSS
   - GPS + GLONASS + Galileo + BDS + QZSS
   - GPS + GLONASS + Galileo + QZSS
   - GPS + GLONASS + BDS + QZSS
   - GPS + Galileo + NavIC + QZSS
   - GPS + Galileo + BDS + QZSS
   - GPS + GLONASS + QZSS
   - GPS + Galileo + QZSS
   - GPS + BDS + QZSS
   - GPS + QZSS
   - BDS
   - NavIC

   Note that QZSS constellation can be enabled or disabled in the constellation configurations supporting QZSS.
3. GLONASS and BDS constellations are supported on L89 (HA) with L89HANR01A06S or higher versions.

2.4.17. PAIR067: PAIR_COMMON_GET_GNSS_SEARCH_MODE

Gets the GNSS search mode.

Type:

Get

Synopsis:

```
$PAIR067*<Checksum><CR><LF>
```

Parameter:

None

Result:

Returns **$PAIR001** message and the query result.

Query result message format:

```
$PAIR067,<GPS_Enabled>,<GLONASS_Enabled>,<Galileo_Enabled>,<BDS_Enabled>,<QZSS_Enabled>,<NavIC_Enabled>*<Checksum><CR><LF>
```

Parameters included in the result:

Packet Data	Format	Unit	Description
<GPS_Enabled>	Numeric	-	0 = Disable (DO NOT search for GPS satellites) 1 = Search for GPS satellites
<GLONASS_Enabled>	Numeric	-	0 = Disable (DO NOT search for GLONASS satellites) 1 = Search for GLONASS satellites
<Galileo_Enabled>	Numeric	-	0 = Disable (DO NOT search for Galileo satellites) 1 = Search for Galileo satellites
<BDS_Enabled>	Numeric	-	0 = Disable (DO NOT search for BDS satellites) 1 = Search for BDS satellites
<QZSS_Enabled>	Numeric	-	0 = Disable (DO NOT search for QZSS satellites) 1 = Search for QZSS satellites
<NavIC_Enabled>	Numeric	-	0 = Disable (DO NOT search for NavIC (IRNSS) satellites) 1 = Search for NavIC (IRNSS) satellites

Example:

```
$PAIR067*3B
$PAIR001,067,0*3A
$PAIR067,1,0,0,0,0,0*3A
```

2.4.18. PAIR070: PAIR_COMMON_SET_STATIC_THRESHOLD

Sets the static navigation speed threshold. If the actual speed is below the threshold, the output position remains unchanged and the output speed is 0. If the threshold value is set to 0, this function is disabled.

Type:

Set

Synopsis:

```
$PAIR070,<SpeedThreshold>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<SpeedThreshold>	Numeric	dm/s	Speed threshold. Range: 0 to 20. Default value: 0.

Result:

Returns **$PAIR001** message.

Example:

```
$PAIR070,4*25
$PAIR001,070,0*3C
```

2.4.19. PAIR071: PAIR_COMMON_GET_STATIC_THRESHOLD

Gets the static navigation speed threshold.

Type:

Get

Synopsis:

```
$PAIR071*<Checksum><CR><LF>
```

Parameter:

None

Result:

Returns **$PAIR001** message and the query result.

Query result message format:

```
$PAIR071,<SpeedThreshold>*<Checksum><CR><LF>
```

Parameter included in the result:

Field	Format	Unit	Description
<SpeedThreshold>	Numeric	m/s	Speed threshold. Range: 0 to 2.0. Default value: 0.

Example:

```
$PAIR071*3C
$PAIR001,071,0*3D
$PAIR071,0.4*3A
```

2.4.20. PAIR072: PAIR_COMMON_SET_ELEV_MASK

Sets satellite elevation mask.

Type:

Set

Synopsis:

```
$PAIR072,<Degree>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Degree>	Numeric	Degree	Satellite elevation mask. Range: -90 to 90. Default value: 5.

Result:

Returns **$PAIR001** message.

Example:

```
$PAIR072,5*26
$PAIR001,072,0*3E
```

NOTE

The satellites below the elevation mask are not used for positioning.

2.4.21. PAIR073: PAIR_COMMON_GET_ELEV_MASK

Gets satellite elevation mask.

Type:

Get

Synopsis:

```
$PAIR073*<Checksum><CR><LF>
```

Parameter:

None

Result:

Returns **$PAIR001** message and the query result.

Query result message format:

```
$PAIR073,<Degree>*<Checksum><CR><LF>
```

Parameter included in the result:

Field	Format	Unit	Description
<Degree>	Numeric	Degree	Satellite elevation mask. Range: -90 to 90.

Example:

```
$PAIR073*3E
$PAIR001,073,0*3F
$PAIR073,5*27
```

2.4.22. PAIR074: PAIR_COMMON_SET_AIC_ENABLE

Enables/disables the active interference cancellation (AIC) function. For details about AIC function, see *document [1] hardware design*.

Type:

Set

Synopsis:

```
$PAIR074,<Enabled>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Enabled>	Numeric	-	Enable/Disable AIC feature. 0 = Disable 1 = Enable

Result:

Returns **$PAIR001** message.

Example:

```
$PAIR074,1*24
$PAIR001,074,0*38
```

2.4.23. PAIR075: PAIR_COMMON_GET_AIC_STATUS

Queries the status of active interference cancellation (AIC) function.

Type:

Get

Synopsis

```
$PAIR075*<Checksum><CR><LF>
```

Parameter:

None

Result:

Returns **$PAIR001** message and the query result.

Query result message format:

```
$PAIR075,<Status>*<Checksum><CR><LF>
```

Parameter included in the result:

Field	Format	Unit	Description
<Status>	Numeric	-	Status of AIC function. 0 = Disabled 1 = Enabled

Example:

```
$PAIR075*38
$PAIR001,075,0*39
$PAIR075,1*25
```

2.4.24. PAIR080: PAIR_COMMON_SET_NAVIGATION_MODE

Sets navigation mode.

Type:

Set

Synopsis:

```
$PAIR080,<NavMode>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<NavMode>	Numeric	-	Navigation mode. 0 = Normal mode. This mode is a basic mode. It is applied to most of scenarios (for example, driving scenario). 1 = Fitness mode. For running and walking activities, the low-speed (< 5 m/s) movement will have a greater effect on the position calculation. It reduces measurement noise caused by arm movement and optimizes the navigation performance in low-speed scenarios. 2 = Reserved. 3 = Balloon mode. Used for high-altitude balloon scenario where the vertical movement has a greater impact on the position calculation. 4 = Stationary mode. For stationary applications where a zero-dynamic is assumed. 5 = Drone mode. Used for drone applications with equivalent dynamics range and vertical acceleration at different flight phases (for example, hovering and cruising).

Field	Format	Unit	Description
			6 = Reserved. 7 = Swimming mode. This mode is designed for swimming activities. It reduces measurement noise caused by specific arm movement and improves the positioning capability after the module is out of water. It also smooths the trajectory and improves the accuracy in distance calculation. 8 = Reserved. 9 = Bike mode. For sharing bike applications.

Table 5: Altitude and Speed Ranges of Navigation Modes

Mode	Max Altitude (m)	Max Speed (m/s)
Normal	10000	100
Fitness	10000	30
Stationary	10000	10
Balloon	80000	10
Drone	10000	30
Swimming	10000	10
Bike	10000	30

Result:

Returns **$PAIR001** message.

Example:

```
$PAIR080,1*2F
$PAIR001,080,0*33
```

2.4.25. PAIR081: PAIR_COMMON_GET_NAVIGATION_MODE

Queries navigation mode.

Type:

Get

Synopsis:

```
$PAIR081*<Checksum><CR><LF>
```

Parameter:

None

Result:

Returns **$PAIR001** message and the query result.

Query result message format:

```
$PAIR081,<NavMode>*<Checksum><CR><LF>
```

Parameter included in the result:

Field	Format	Unit	Description
<NavMode>	Numeric	-	Navigation mode. 0 = Normal mode. This mode is a basic mode. It is applied to most of scenarios (for example, driving scenario). 1 = Fitness mode. For running and walking activities, the low-speed (< 5 m/s) movement will have a greater effect on the position calculation. It reduces measurement noise caused by arm movement and optimizes the navigation performance in low-speed scenarios. 2 = Reserved. 3 = Balloon mode. Used for high-altitude balloon scenario where the vertical movement has a greater impact on the position calculation. 4 = Stationary mode. For stationary applications where a zero-dynamic is assumed. 5 = Drone mode. Used for drone applications with equivalent dynamics range and vertical acceleration at different flight phases (For example, hovering and cruising). 6 = Reserved. 7 = Swimming mode. This mode is designed for swimming activity. It reduces measurement noise caused by specific arm movement and improves the positioning capability after the module is out of water. It also smooths the trajectory and improves the accuracy of distance calculation. 8 = Reserved. 9 = Bike mode. For sharing bike applications.

Example:

```
$PAIR081*33
```

```
$PAIR001,081,0*32
$PAIR081,0*2F
```

2.4.26. PAIR086: PAIR_COMMON_SET_DEBUGLOG_OUTPUT

Enables/disables debug log output in binary format.

Type:

Set

Synopsis

```
$PAIR086,<Status>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Status>	Numeric	-	Debug log output setting. 0 = Disable 1 = Enable with full debug log output 2 = Enable with lite debug log output

Result:

Returns **$PAIR001** message.

Example:

```
$PAIR086,1*29
$PAIR001,086,0*35
```

2.4.27. PAIR087: PAIR_COMMON_GET_DEBUGLOG_OUTPUT

Queries the debug log output setting.

Type:

Get

Synopsis

```
$PAIR087*<Checksum><CR><LF>
```

Parameter:

None

Result:

Returns **$PAIR001** message and the query result.

Query result message format:

```
$PAIR087,<Status>*<Checksum><CR><LF>
```

Parameter included in the result:

Field	Format	Unit	Description
<Status>	Numeric	-	Debug log output setting. 0 = Disabled 1 = Enabled with full debug log output 2 = Enabled with lite debug log output

Example:

```
$PAIR087*35
$PAIR001,087,0*34
$PAIR087,0*29
```

2.4.28. PAIR098: PAIR_COMMON_SET_NMEA_POS_DECIMAL_PRECISION

Sets the coordinate precision, i.e., the decimal places in the output coordinates.

Type:

Set

Synopsis:

```
$PAIR098,<Mode>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Mode>	Numeric	-	Coordinate precision mode. 0 = Latitude, Longitude: 4; Altitude: 1 1 = Latitude, Longitude: 5; Altitude: 2 2 = Latitude, Longitude: 6; Altitude: 3 3 = Latitude, Longitude: 7; Altitude: 3

Result:

Returns **$PAIR001** message.

Example:

```
$PAIR098,2*25
$PAIR001,098,0*3A
```

2.4.29. PAIR099: PAIR_COMMON_GET_NMEA_POS_DECIMAL_PRECISION

Gets the coordinate precision.

Type:

Get

Synopsis:

```
$PAIR099*<Checksum><CR><LF>
```

Parameter:

None

Result:

Returns **$PAIR001** message and the query result.

Query result message format:

```
$PAIR099,<Mode>*<Checksum><CR><LF>
```

Parameter included in the result:

Field	Format	Unit	Description
<Mode>	Numeric	-	Coordinate precision mode. 0 = Latitude, Longitude: 4; Altitude: 1 1 = Latitude, Longitude: 5; Altitude: 2 2 = Latitude, Longitude: 6; Altitude: 3 3 = Latitude, Longitude: 7; Altitude: 3

Example:

```
$PAIR099*3A
$PAIR001,099,0*3B
$PAIR099,2*24
```

2.4.30. PAIR100: PAIR_COMMON_SET_NMEA_OUTPUT_MODE

Sets output mode of standard NMEA sentences.

Type:

Set

Synopsis:

```
$PAIR100,<NMEA_Mode>,<Res>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<NMEA_Mode>	Numeric	-	Output mode of standard NMEA sentences. 0 = Disabled 1 = ASCII NMEA 0183 V4.10 output enabled 2 = ASCII NMEA 0183 V3.01 output enabled
<Res>	Numeric	-	Reserved. Default value: 0

Result:

Returns **$PAIR001** message.

Example:

```
$PAIR100,1,0*3A
$PAIR001,100,0*3A
```

2.4.31. PAIR101: PAIR_COMMON_GET_NMEA_OUTPUT_MODE

Queries output mode of standard NMEA sentences.

Type:

Get

Synopsis:

```
$PAIR101*<Checksum><CR><LF>
```

Parameter:

None

Result:

Returns **$PAIR001** message and the query result.

Query result message format:

```
$PAIR101,<NMEA_Mode>,<Res>*<Checksum><CR><LF>
```

Parameters included in the result:

Field	Format	Unit	Description
<NMEA_Mode>	Numeric	-	Output mode of standard NMEA sentences. 0 = Disabled 1 = ASCII NMEA 0183 V4.10 output enabled 2 = ASCII NMEA 0183 V3.01 output enabled
<Res>	Numeric	-	Reserved. Default value: 0.

Example:

```
$PAIR101*3A
$PAIR001,101,0*3B
$PAIR101,1,0*3B
```

2.4.32. PAIR382: PAIR_TEST_LOCK_SYSTEM_SLEEP

Enables/disables the locking of sleep mode. The CPU core will lock into the sleep mode after the command is sent.

Type:

Set

Synopsis:

```
$PAIR382,<Enabled>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Enabled>	Numeric	-	Sleep mode locking. 0 = Disabled 1 = Enabled

Result:

Returns **$PAIR001** message.

Example:

```
$PAIR382,1*2E
$PAIR001,382,0*32
```

NOTE

This configuration will not be saved in the flash or RTC RAM. Please send this command every time after the GNSS subsystem or main power reboots.

2.4.33. PAIR391: PAIR_TEST_JAMMING_DETECT

Enables/disables jamming detection. Jamming status messages will be returned if jamming detection is enabled.

Type:

Set/Output

Synopsis:

```
$PAIR391,<CmdType>*<Checksum><CR><LF>
```

Parameter:

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

Result:

Returns **$PAIR001** message and enables periodic output of **$PAIRSPF** message (at 1 Hz).

Query result message format:

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

Parameter included in the result:

Field	Format	Unit	Description
<Status>	Numeric	-	Jamming status. 0 = Unknown Status 1 = No jamming, good status 2 = Warning status 3 = Critical status

Example:

```
$PAIR391,1*2C
$PAIR001,391,0*30
//Unknown status:
$PAIRSPF,0*53
//Good status:
$PAIRSPF,1*52
//Warning status:
$PAIRSPF,2*51
//Critical status:
$PAIRSPF,3*50
```

NOTE

The module starts jamming detection once the function is enabled.

1) If there is no jamming, **$PAIRSPF,1*52** will be reported to indicate good status (**<Status>** = 1).
2) If there is continuous jamming, the jamming status will change from 1 to 2 and finally to 3.
   - If there is no position fix: module status is 1, once jamming detection is enabled, and then changes to 2 when jamming is detected. During this process, the module keeps attempting to get a fix; if the anti-jamming repair fails, the jamming status changes to 3 at last.
   - After a successful position fix: jamming status is 1, once jamming detection is enabled, and changes to 2 and 3 consecutively when jamming is detected.

2.4.34. PAIR400: PAIR_DGPS_SET_MODE

Sets the DGPS correction data source.

Type:

Set

Synopsis:

```
$PAIR400,<Mode>*<Checksum><CR><LF>
```

Parameter:

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

Result:

Returns **$PAIR001** message.

Example:

```
$PAIR400,2*20
$PAIR001,400,0*3F
```

2.4.35. PAIR401: PAIR_DGPS_GET_MODE

Queries the DGPS correction data source.

Type:

Get

Synopsis:

```
$PAIR401*<Checksum><CR><LF>
```

Parameter:

None

Result:

Returns **$PAIR001** message and the query result.

Query result message format:

```
$PAIR401,<Mode>*<Checksum><CR><LF>
```

Parameter included in the result:

Field	Format	Unit	Description
<Mode>	Numeric	-	DGPS data source. 0 = No DGPS source

	1 = Reserved 2 = SBAS (including WAAS/EGNOS/GAGAN/MSAS)

Example:

```
$PAIR401*3F
$PAIR001,401,0*3E
$PAIR401,2*21
```

2.4.36. PAIR410: PAIR_SBAS_ENABLE

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

Type:

Set

Synopsis:

```
$PAIR410,<Enabled>*<Checksum><CR><LF>
```

Parameter:

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

Result:

Returns **$PAIR001** message.

Example:

```
$PAIR410,1*22
$PAIR001,410,0*3E
```

NOTE

When the navigation mode is Fitness or Swimming mode (see *Chapter 2.4.24 PAIR080: PAIR_COMMON_SET_NAVIGATION_MODE* for details), SBAS is not supported.

2.4.37. PAIR411: PAIR_SBAS_GET_STATUS

Queries the status of SBAS satellite searching.

Type:

Get

Synopsis:

```
$PAIR411*<Checksum><CR><LF>
```

Parameter:

None

Result:

Returns **$PAIR001** message and the query result.

Query result message format:

```
$PAIR411,<Enabled>*<Checksum><CR><LF>
```

Parameter included in the result:

Field	Format	Unit	Description
<Enabled>	Numeric	-	Status of SBAS satellite searching. 0 = Disabled 1 = Enabled

Example:

```
$PAIR411*3E
$PAIR001,411,0*3F
$PAIR411,1*23
```

NOTE

When the navigation mode is Fitness or Swimming mode (see command **$PAIR080**), SBAS is not supported.

2.4.38. PAIR511: PAIR_NVRAM_SAVE_NAVIGATION_DATA

Saves current navigation data from RTC RAM to flash.

Type:

Command

Synopsis

```
$PAIR511*<Checksum><CR><LF>
```

Parameter:

None

Result:

Returns **$PAIR001** message.

Example:

```
//In case the position fix rate is 1 Hz.
$PAIR511*3F
$PAIR001,511,0*3E

//In case the position fix rate is greater than 1 Hz.
$PAIR382,1*2E
$PAIR001,382,0*32
$PAIR003*39
$PAIR001,003,0*38
$PAIR511*3F
$PAIR001,511,0*3E
$PAIR002*38
$PAIR001,002,0*39
```

NOTE

1. If the backup domain cannot be powered after the power supply of the module is cut off, this command needs to be sent every time the parameters are modified.
2. In case the position fix rate is greater than 1 Hz, power off the GNSS system with **$PAIR382,1*2E** and **$PAIR003*39** in sequence before sending this command. After sending **$PAIR511*3F**, re-power the module with **$PAIR002*38**. This limitation does not apply to fix rates below 1 Hz.

2.4.39. PAIR512: PAIR_NVRAM_CLEAR_NAVIGATION_DATA

Clears navigation data in both RTC RAM and NVM.

Type:

Command

Synopsis

```
$PAIR512*<Checksum><CR><LF>
```

Parameter:

None

Result:

Returns **$PAIR001** message.

Example:

```
$PAIR512*3C
$PAIR001,512,0*3D
```

2.4.40. PAIR513: PAIR_NVM_SAVE_SETTING

Saves the current configuration from RTC RAM to flash.

Type:

Command

Synopsis:

```
$PAIR513*<Checksum><CR><LF>
```

Parameter:

None

Result:

Returns **$PAIR001** message.

Example:

```
//In case the position fix rate is 1 Hz.
$PAIR513*3D
```

```
$PAIR001,513,0*3C

//In case the position fix rate is greater than 1 Hz.
$PAIR382,1*2E
$PAIR001,382,0*32
$PAIR003*39
$PAIR001,003,0*38
$PAIR513*3D
$PAIR001,513,0*3C
$PAIR002*38
$PAIR001,002,0*39
```

NOTE

1. If the backup domain cannot be powered after the power supply of the module is cut off, this command needs to be sent every time the parameters are modified.
2. In case the position fix rate is greater than 1 Hz, power off the GNSS system with **$PAIR382,1*2E** and **$PAIR003*39** in sequence before sending this command. After sending **$PAIR513*3D**, re-power the module with **$PAIR002*38**. This limitation does not apply to fix rates below 1 Hz.

2.4.41. PAIR650: PAIR_LOW_POWER_ENTER_RTC_MODE

Shuts down the GNSS system, except the clock. The CPU core will be set to the Backup mode after the command is sent, in which it cannot receive any commands. For details about Backup mode, see *document [1] hardware design*.

Type:

Set

Synopsis:

```
$PAIR650,<Second>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Second>	Numeric	Second	Time to stay in Backup mode before exiting. Range: 0 and 10–62208000; 0 means entering the Backup mode without any timer.

Result:

- If there is no error, the **$PAIR001** and **$PAIR650** messages will be returned. The module will be set to Backup mode and cannot receive any commands.
- In case of any command parameter error, the **$PAIR001** message will be returned.

Example:

```
$PAIR650,0*25
$PAIR001,650,0*38
$PAIR650,0*25
```

NOTE

Refer to *document [1] hardware design* for details about entering/exiting the Backup mode.

2.4.42. PAIR752: PAIR_PPS_SET_CONFIG_CMD

Sets PPS configurations.

Type:

Set

Synopsis:

```
$PAIR752,<PPSType>,<PPSPulseWidth>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<PPSType>	Numeric	-	PPS pulse type. 0 = Disable 1 = After the first position fix 2 = 3D position fix only 3 = 2D/3D position fix only 4 = Always
<PPSPulseWidth>	Numeric	Millisecond	PPS pulse width. Range: 1 to 999. Default value: 100.

Result:

Returns **$PAIR001** message.

Example:

```
$PAIR752,2,100*39
$PAIR001,752,0*3B
```

2.4.43. PAIR864: PAIR_IO_SET_BAUDRATE

Sets the baud rate of UART interface.

Type:

Set

Synopsis:

```
$PAIR864,<PortType>,<PortIndex>,<BaudRate>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<PortType>	Numeric	-	HW Port Type. 0 = UART
<PortIndex>	Numeric	-	HW Port Index. 0 = UART1
<BaudRate>	Numeric	bps	Baud rate value. 9600 19200 38400 57600 115200 230400 460800 921600

Result:

Returns **$PAIR001** message.

Example:

```
$PAIR864,0,0,115200*1B
$PAIR001,864,0*31
```

NOTE

1. For the configuration to take effect reboot the module after changing the port baud rate.
2. When the baud rate is set to 9600 bps, the command response and NMEA message output may be delayed due to high data volume.

2.4.44. PAIR865: PAIR_IO_GET_BAUDRATE

Gets the baud rate of UART interface.

Type:

Get

Synopsis:

```
$PAIR865,<PortType>,<PortIndex>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<PortType>	Numeric	-	HW Port Type. 0 = UART
<PortIndex>	Numeric	-	HW Port Index. 0 = UART1

Result:

Returns **$PAIR001** message and the query result.

Query result message format:

```
$PAIR865,<Baudrate>*<Checksum><CR><LF>
```

Parameter included in the result:

Field	Format	Unit	Description
<Baudrate>	Numeric	bps	Baud rate value. 9600 19200 38400 57600 115200 230400

Field	Format	Unit	Description
			460800 921600

Example:

```
$PAIR865,0,0*31
$PAIR001,865,0*30
$PAIR865,115200*1A
```

3 Appendix A References

Table 6: Related Document

Document Name
[1] Quectel_L89_R2.0_Hardware Design

Table 7: Terms and Abbreviations

Abbreviation	Description
2D	2 Dimension
3D	3 Dimension
ACK	Acknowledgement
AIC	Active Interference Cancellation
ASCII	American Standard Code for Information Interchange
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)
DGPS	Differential Global Positioning System
DOP	Dilution of Precision
DSP	Digital Signal Processing
EASY	Embedded Assist System
EGNOS	European Geostationary Navigation Overlay Service

Abbreviation	Description
EPO	Extended Prediction Orbit
GAGAN	GPS Aided Geo Augmented Navigation
Galileo	Galileo Satellite Navigation System (EU)
GGA	Global Positioning System Fix Data
GLL	Geographic Position-Latitude and Longitude
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GRS	GNSS Range Residuals
GSA	GNSS DOP and Active Satellites
GST	GNSS Pseudorange Error Statistics
GSV	GNSS Satellites in View
HDOP	Horizontal Dilution of Precision
HW	Hardware
IRNSS/NavIC	Indian Regional Navigation Satellite System
MNL	MTK Navigation Lib
MSAS	Multi-functional Satellite Augmentation System (Japan)
NMEA	NMEA (National Marine Electronics Association) 0183 Interface Standard
NVM	Non-Volatile Memory
NVRAM	Non-Volatile Random Access Memory
PAIR	Proprietary Protocol of MTK
PDOP	Position Dilution of Precision
PE	Positioning Engine
PPS	Pulse Per Second
QZSS	Quasi-Zenith Satellite System

Abbreviation	Description
RAM	Random Access Memory
RMC	Recommended Minimum Specific GNSS Data
RTC	Real-Time Clock
RTK	Real-Time Kinematic
SBAS	Satellite-Based Augmentation System
SNR	Signal-to-Noise Ratio
SV	Satellites in View
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 8: GNSS Satellites (NMEA) Numbering

GNSS Type	System ID	Satellite ID	Signal ID
GPS	1	1–32	1 = L1 C/A
GLONASS	2	65–88	1 = L1
Galileo	3	1–36	7 = E1
BDS	4	1–63	1 = B1I
QZSS	1	193–199	1 = L1 C/A
NavIC (IRNSS)	6	1–14	1 = L5
SBAS	-	33–51	-

5 Appendix C Special Characters

Table 9: Special Characters

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