LC29H&LC79H Series
GNSS Protocol Specification

GNSS Products

Version: 1.5

Date: 2025-09-12

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	**LC29H&LC79H Series GNSS Protocol Specification**
Subtitle	GNSS Products
Document Type	GNSS Protocol Specification
Document Status	Released

Revision History

Version	Date	Description
-	2021-03-18	Creation of the document
1.0	2021-08-19	First official release
1.1	2022-08-26	Numerous changes were made to this document. It should be read in its entirety.
1.2	2023-06-20	1. Updated applicable variant LC79H series to LC79H (AL). 2. Added the table of supported protocols (*Table 1*). 3. Added Sample Code for NMEA Checksum (*Chapter 2.1*). 4. Updated the note on <NumSatUsed> in GGA message (*Chapter 2.2.2*). 5. Updated the note for GSA message (*Chapter 2.2.4*). 6. Added a note about calculation of range residual in GRS message (*Chapter 2.2.8*). 7. Added PQTM messages (*Chapter 2.2.10*). 8. Updated the example and notes for Packet Type 066 message (*Chapter 2.4.15*). 9. Added a table about altitude and speed ranges of navigation modes for Packet Type 080 message (*Chapter 2.4.23*). 10. Updated the descriptions of <NavMode> for Packet Type 080 and Packet Type 081 messages (*Chapters 2.4.23* and *2.4.24*). 11. Updated the notes of LC29H (BA, CA, DA) supporting the Packet Type

Version	Date	Description
		410 and 411 messages (*Chapters 2.4.35* and *2.4.36*).
1.3	2024-03-12	1. Deleted the note that LC29H (EA) does not support the $PQTMEPE message (*Chapter 2.3.4*). 2. Updated the message output rate range of $PQTMPVT and added the note on the message output rate supported on LC29H (BA, CA, DA, EA) (*Chapter 2.3.5*). 3. Deleted the note that the $PQTMCFGGEOFENCE command is supported on LC29H (AA, EA) and LC79H (AL) (*Chapter 2.3.6*). 4. Deleted the notes on LC29H (CA) supporting $PQTMGNSSSTART and $PQTMGNSSSTOP (*Chapters 2.3.10* and *2.3.11*). 5. Deleted the note that $PQTMPVT is supported on LC29H (BA, CA) (*Chapter 2.3.12*). 6. Added the following PQTM messages (*Chapters 2.3.15*–*2.3.30*): ● $PQTMPL ● $PQTMCOLD ● $PQTMHOT ● $PQTMWARM ● $PQTMSRR ● $PQTMCFGSBAS ● $PQTMCFGCNST ● $PQTMDOP ● $PQTMCFGFIXRATE ● $PQTMCFGPPS ● $PQTMDEBUGON ● $PQTMDEBUGOFF ● $PQTMVEL ● $PQTMCFGODO ● $PQTMRESETODO ● $PQTMODO 7. Added BDS-only GNSS search modes for $PAIR066 message (*Chapter 2.4.15*). 8. Deleted $PAIR490, $PAIR491, $PAIR690, $PAIR691, $PAIR830 and $PAIR831 messages.
1.4	2025-01-13	1. Added NavIC constellation. 2. Updated the note of <NumSatUsed> and the value of <HDOP> when the field is invalid (*Chapter 2.2.2*). 3. Updated the value of <PDOP>, <HDOP> and <VDOP> when the fields are invalid (*Chapter 2.2.4*). 4. Added GNS message (*Chapter 2.2.10*). 5. Updated the supported values of <ErrCode> (*Table 5*). 6. Added the note for $PQTMRESTOREPAR execution result (*Chapter 2.3.3*).

Version	Date	Description
		7. Updated the example, supported messages, and note about message output rate for $PQTMCFGMSGRATE (*Chapter 2.3.5*). 8. Updated the default values of <ALT_DP> and <SPD_DP>parameters for $PQTMCFGNMEADP (*Chapter 2.3.13*). 9. Updated <Reserved> to <NavIC> for $PQTMCFGCNST and notes about constellation configurations supported by the module, and added a note about the requirement for $PQTMCFGCNST to take effect (*Chapter 2.3.21*). 10. Updated the supported values of <MsgVer> for $PQTMODO and added the related note (*Chapter 2.3.30*). 11. Added the following PQTM messages (*Chapters 2.3.31*–*2.3.40*): ● $PQTMJAMMINGSTATUS ● $PQTMLS ● $PQTMCFGRSID ● $PQTMCFGNMEATID ● $PQTMUNIQID ● $PQTMVER ● $PQTMCFGUART ● $PQTMCFGPROT ● $PQTMBKP ● $PQTMGETUTC 12. Added the note about the interval between power on/off the GNSS system via $PAIR002/$PAIR003 (*Chapters 2.4.2* and *2.4.3*). 13. Updated the supported values of <Type> for $PAIR062 and $PAIR063 (*Chapters 2.4.13* and *2.4.14*). 14. Updated <Reserved> to <NavIC> for $PAIR066 and $PAIR067 (*Chapters 2.4.15* and *2.4.16*). 15. Updated the note about constellation configurations supported by the module (*Chapter 2.4.15*). 16. Added the default value of the <Degree> parameter for $PAIR073 (*Chapter 2.4.20*). 17. Updated the example for $PAIR104 (*Chapter 2.4.29*). 18. Updated the supported values of the <Mode> parameter for $PAIR400 and $PAIR401 (*Chapters 2.4.33* and *2.4.34*). 19. Added the $PAIR420 and $PAIR421 messages (*Chapters 2.4.37* and *2.4.38*). 20. Added message types 1006, 1075, 1085, 1095, 1115, 1125, 1134, 1135 and 1137, and updated the modes for message types 1005, 1074, 1077, 1084, 1087, 1094, 1097, 1114, 1117, 1124 and 1127 (*Chapter 3*).
1.5	2025-09-12	1. Added applicable module LC79H (EL). 2. Added the support of M for <ModeInd> parameter in RMC, VTG, GLL, GNS, and the support of 7 for <Quality> in GGA (*Chapters 2.2.1*, *2.2.2*,

Version	Date	Description
		2.2.5, 2.2.6 and 2.2.10) 3. Updated ZDA and GST messages to be supported on LC29H (BA, CA, DA, EA) (Chapters 2.2.7 and 2.2.9). 4. Added the optional parameter <Distance> in $PQTMCFGSVIN and a note about the parameter value requirement in survey-in mode (Chapter 2.3.8). 5. Updated $PQTMCOLD, $PQTMHOT and $PQTMWARM to be supported on LC29H (BA, CA, DA, EA) (Chapters 2.3.16, 2.3.17 and 2.3.18). 6. Updated descriptions of $PQTMCOLD and $PAIR006 (Chapters 2.3.16 and 2.4.6). 7. Updated the supported constellation configurations on LC29H (BA, CA, DA, EA) (Chapter 2.3.21). 8. Added a note about the supported value of <Interval> for LC29H (DA) in $PQTMCFGFIXRATE (Chapter 2.3.23). 9. Updated <Interval> to <Reserved> in $PQTMCFGPPS (Chapter 2.3.24). 10. Added the value range of <InitDist> in $PQTMCFGODO (Chapter 2.3.28). 11. Added following PQTM messages: ● $PQTMCFGNAVMODE (Chapter 2.3.41); ● $PQTMQVER (Chapter 2.3.42); ● $PQTMSN (Chapter 2.3.43); ● $PQTMSTD (Chapter 2.3.44); ● $PQTMCFGAIC (Chapter 2.3.45). 12. Updated the default value of <Time> in $PAIR050 and added notes about setting the parameter (Chapter 2.4.9). 13. Updated supported values of <NavMode> in $PAIR080 and $PAIR081 and added the note about message applicability (Chapters 2.4.23 and 2.4.24). 14. Added the description about the default value of <PPSType> in $PAIR752 and the corresponding note (Chapter 2.4.48). 15. Updated notes about baud rates in $PAIR864 (Chapter 2.4.49). 16. Updated the supported values of <Mode> in $PAIR400 and $PAIR401 (Chapters 2.4.33 and 2.4.34). 17. Updated LC29H (EA) to support SBAS feature (Chapters 2.4.35 and 2.4.36).

Contents

Table Index

1 Introduction

Quectel LC29H series and LC79H series GNSS modules support GPS, GLONASS, Galileo, BDS, QZSS and NavIC constellations. Tracking of GPS L1 C/A, GLONASS L1, Galileo E1, BDS B1I, QZSS L1 C/A, GPS L5, Galileo E5a, BDS B2a, QZSS L5 and NavIC L5 frequency bands provides fast and accurate acquisition and makes these modules ideal solutions for positioning and navigation in various vertical markets.

This document describes the software commands that are needed to control and modify the module configuration. The software commands are NMEA proprietary commands defined by the chipset supplier (PAIR/PQTM messages). To report GNSS information, the modules support output messages in NMEA 0183 standard protocol format and RTCM protocol format.

The LC29H series and LC79H series GNSS modules support the following protocols:

Table 1: Supported Protocols

Protocol	Type
NMEA 0183 V3.01/V4.10	Output, ASCII, standard
	Input/output, ASCII, proprietary
RTCM 10403.3	Input/output, binary, standard

NOTE

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

1.1. Applicable Modules

Table 2: Applicable Modules

Module Series	Model
LC29H	LC29H (AA)
	LC29H (BA)
	LC29H (CA)
	LC29H (DA)
	LC29H (EA)
LC79H	LC79H (AL)
	LC79H (EL)

2 NMEA Protocol

2.1. Structure of NMEA Protocol Messages

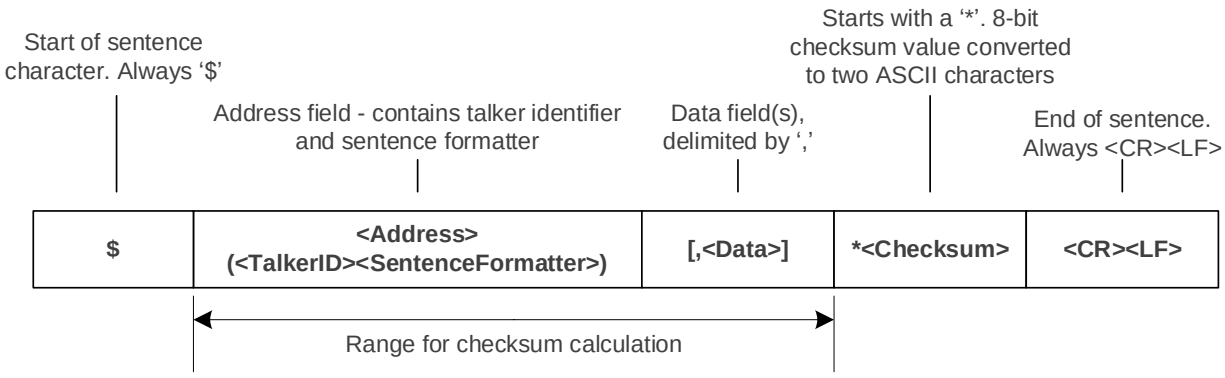

Figure 1: Structure of NMEA Protocol Messages

Table 3: Structure of NMEA Protocol Messages

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

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

Table 4: NMEA Talker ID

GNSS Constellation Configuration	Talker ID (NMEA 0183 V3.01)	Talker ID (NMEA 0183 V4.10)
GPS	GP	GP
GLONASS	GL	GL
Galileo	GA	GA
BDS	GB	GB
QZSS	GP	GP
NavIC	-	GI
Combination of Multiple Satellite Systems	GN	GN

NOTE

The Talker ID of QZSS in NMEA 0183 V4.10 is GQ for LC29H (BA, CA, DA, EA) and LC79H (EL).

NMEA Checksum Sample Code:

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

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

    if((NULL == pData) || (Length < 1))
    {
```

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

    return result;
}
```

2.2. Standard Messages

This chapter explains the standard NMEA 0183 V3.01 and NMEA 0183 V4.10 messages supported by the modules.

2.2.1. RMC

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

Type:

Output

Synopsis:

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

Field	Format	Unit	Example	Description
RMC	String	-	RMC	Recommended Minimum Specific GNSS Data.
<UTC>	hhmmss.sss	-	093316.000	Position fix UTC. hh: Hours (00–23) mm: Minutes (00–59) ss: Seconds (00–59) sss: Decimal fraction of seconds
<Status>	Character	-	A	Positioning system status. A = Data valid V = Navigation receiver warning
<Lat>	ddmm.mmmmmm	-	3149.332558	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	-	11706.912570	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	0.00	Speed over ground. Variable length. Note that this field is empty in case of an invalid value.
<COG>	Numeric	Degree	237.67	Course over ground. Variable length. Maximum value: 359.99. Note that this field is empty in case of an invalid value.

Field	Format	Unit	Example	Description
<Date>	ddmmyy	-	140122	Date. dd: Day of month mm: Month yy: Year
<MagVar>	-	-	-	Magnetic variation. Not supported. Always null.
<MagVarDir>	-	-	-	Direction of magnetic variation. Not supported. Always null.
<ModeInd>	Character	-	A	Mode indicator. A = Autonomous mode. Satellite system used for position fixing in non-differential mode. D = Differential mode. Satellite system used for position fixing in differential mode. Corrections from ground stations or Satellite Based Augmentation System (SBAS). E = Estimated (dead reckoning) mode. F = Float RTK. Satellite system used in RTK mode with floating integers. M = Manual input mode or derived from Survey-in result. N = No fix. Satellite system not used for position fixing, or fix not valid. R = Real Time Kinematic (RTK). Satellite system used in RTK mode with fixed integers.
<NavStatus>	Character	-	V	Navigational status. Not supported. Always “V” (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	-	*0B	Checksum
<CR><LF>	Character	-	-	Carriage return and line feed.

Example:

NMEA 0183 V3.01 example:

```
$GNRMC,093423.000,A,3149.332006,N,11706.913200,E,0.01,0.00,140122,,,A*79
```

NMEA 0183 V4.10 example:

```
$GNRMC,093316.000,A,3149.332558,N,11706.912570,E,0.00,237.67,140122,,,A,V*0B
```

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	-	GN	Talker identifier. See *Table 4: NMEA Talker ID*.
GGA	String	-	GGA	Global Positioning System Fix Data.
<UTC>	hhmmss.sss	-	093316.000	Position fix UTC. hh: Hours (00–23) mm: Minutes (00–59) ss: Seconds (00–59) sss: Decimal fraction of seconds
<Lat>	ddmm.mmmmmm	-	3149.332558	Latitude. dd: Degrees (00–90) mm: Minutes (00–59)

Field	Format	Unit	Example	Description
				mmmmmm: Decimal fraction of minutes. Variable length. Note that this field is empty in case of an invalid value.
<N/S>	Character	-	N	North-south direction. N = North S = South Note that this field is empty in case of an invalid value.
<Lon>	dddmm.mmmmmm	-	11706.912570	Longitude. ddd: Degrees (000–180) mm: Minutes (00–59) mmmmmm: Decimal fraction of minutes. Variable length. Note that this field is empty in case of an invalid value.
<E/W>	Character	-	E	East-west direction. E = East W = West Note that this field is empty in case of an invalid value.
<Quality>	Numeric, 1 digit	-	1	GPS quality indicator. 0 = Fix not available or invalid. 1 = GPS SPS Mode, fix valid. 2 = Differential GPS, SPS Mode, or Satellite Based Augmentation System (SBAS), fix valid. 3 = GPS PPS Mode, fix valid. 4 = Real Time Kinematic (RTK) System used in RTK mode with fixed integers. 5 = Float RTK. Satellite system used in RTK mode, floating integers. 6 = Estimated (dead reckoning) mode. 7 = Manual input mode or derived from Survey-in result.
<NumSatUsed> [1)]	Numeric, 2 digits	-	28	Number of satellites in use. Range: 00–99.

Field	Format	Unit	Example	Description
<HDOP>	Numeric	-	0.67	Horizontal dilution of precision. Variable length. Maximum value: 99.99. Note that this field is 99.99 in case of an invalid value.
<Alt>	Numeric	Meter	54.481	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	-0.337	Geoid separation (the difference between the earth ellipsoid surface and the mean-sea-level (geoid) surface defined by the reference datum used in the position solution). Variable length. Note that this field is empty in case of an invalid value.
M	Character	-	M	Unit of <Sep>. "M" = Meter.
<DiffAge> [2)]	-	-	-	Differential GPS data age.
<DiffStation> [2)]	-	-	-	Differential reference station ID.
<Checksum>	Hexadecimal	-	*5D	Checksum
<CR><LF>	Character	-	-	Carriage return and line feed.

Example:

NMEA 0183 V3.01 example:

```
$GNGGA,093423.000,3149.332006,N,11706.913200,E,1,22,0.75,45.346,M,-0.337,M,,*56
```

NMEA 0183 V4.10 example:

```
$GNGGA,093316.000,3149.332558,N,11706.912570,E,1,28,0.67,54.481,M,-0.337,M,,*5D
```

NOTE

1. The NMEA 0183 specification indicates that **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] For **<NumSatUsed>**:
   According to the NMEA 0183 specification, the number of satellites in use is between 00 and 12. However, in the multi-constellation solution, the number of satellites in use may exceed 12.
3. [2] For **<DiffAge>** and **<DiffStation>**:
   LC29H (BA, DA, EA) and LC79H (EL) support these parameters, whereas LC29H (AA, CA) and LC79H (AL) do not support them.

2.2.3. GSV

GNSS Satellites in View. The GSV sentence provides the number of satellites in view (SV), satellite ID numbers, elevation, azimuth, and SNR value, and it 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 **$**.
<TalkerID>	String	-	GP	Talker identifier. See *Table 4: NMEA Talker ID*.
GSV	String	-	GSV	GNSS Satellites in View.
<TotalNumSen>	Numeric	-	3	Total number of sentences. Range: 1–9.
<SenNum>	Numeric	-	1	Sentence number. Range: 1–**<TotalNumSen>**.
<TotalNumSat>	Numeric	-	11	Total number of satellites in view.

Field	Format	Unit	Example	Description
Start of repeat block. Repeat times: 1–4.				
<SatID>	Numeric	-	08	Satellite ID. See *Table 11: GNSS Satellites (NMEA) Numbering*.
<SatElev>	Numeric	Degree	76	Satellite elevation. Range: 00–90. Note that this field is empty in case of an invalid value.
<SatAz>	Numeric	Degree	353	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	46	Satellite C/N_0. Range 00–99. Null when not tracking.
End of repeat block.				
<SignalID>	Numeric	-	1	GNSS signal ID. See *Table 11: 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	-	*5C	Checksum
<CR><LF>	Character	-	-	Carriage return and line feed.

Example:

NMEA 0183 V3.01 example:

```
$GPGSV,3,1,12,08,75,355,46,195,68,113,44,194,64,081,42,199,51,162,36*47
$GPGSV,3,2,12,27,45,037,44,21,44,156,43,16,30,072,39,30,18,317,38*72
$GPGSV,3,3,12,03,,,19,04,,,37,09,,,40,07,,,43*7C
$GPGSV,2,1,08,08,75,355,38,195,68,113,38,194,64,081,38,199,51,162,38*4C
$GPGSV,2,2,08,27,45,037,38,30,18,317,31,04,,,29,09,,,32*70
$GLGSV,1,1,04,86,63,182,32,85,59,041,42,76,36,333,41,66,07,278,*6A
$GAGSV,1,1,00*68
$GAGSV,1,1,00*68
$GBGSV,3,1,12,29,82,083,41,07,64,319,41,09,54,220,40,36,47,113,45*66
$GBGSV,3,2,12,06,45,201,39,39,34,192,40,01,,,38,30,,,40*69
$GBGSV,3,3,12,16,,,41,05,,,31,02,,,36,13,,,35*6E
$GBGSV,1,1,04,29,82,083,36,36,47,113,36,39,34,192,32,30,,,28*5F
```

NMEA 0183 V4.10 example:

```
$GPGSV,3,1,11,08,76,353,46,195,68,113,44,194,64,080,42,199,51,162,37,1*5C
$GPGSV,3,2,11,27,45,037,43,07,44,315,43,21,43,156,44,16,30,071,39,1*6B
$GPGSV,3,3,11,09,28,244,40,04,20,206,37,30,,,38,1*6F
$GPGSV,2,1,08,08,76,353,35,195,68,113,34,194,64,080,33,199,51,162,34,8*5A
$GPGSV,2,2,08,27,45,037,33,09,28,244,28,04,20,206,26,30,,,27,8*5E
$GLGSV,1,1,03,85,59,042,41,76,35,333,40,66,07,279,32,1*43
$GAGSV,1,1,00,7*73
$GAGSV,1,1,00,1*75
$GBGSV,4,1,13,07,64,319,41,40,61,345,43,36,47,113,43,06,45,201,40,1*7E
$GBGSV,4,2,13,16,41,195,41,35,40,213,45,26,17,313,39,56,,,19,1*49
$GBGSV,4,3,13,57,,,37,01,,,39,02,,,36,60,,,40,1*7B
$GBGSV,4,4,13,03,,,40,1*73
$GBGSV,1,1,04,40,61,345,32,36,47,113,32,35,40,213,29,26,17,313,23,5*79
```

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 **$**.

Field	Format	Unit	Example	Description
<TalkerID>	String	-	GN	Talker identifier. See *Table 4: NMEA Talker ID*.
GSA	String	-	GSA	GNSS DOP and Active Satellites.
<Mode>	Character	-	A	Selection of 2D or 3D fix. M = Manual, forced to operate in 2D or 3D mode. A = Automatic, allowed to automatically switch to 2D/3D.
<FixMode>	Numeric	-	3	Fix mode. 1 = Fix not available 2 = 2D 3 = 3D
Start of repeat block. Repeat times: 12.				
<SatID>	Numeric	-	08	ID numbers of satellites used in solution. See *Table 11: GNSS Satellites (NMEA) Numbering*. Note that this field is empty in case of an invalid value.
End of repeat block.				
<PDOP>	Numeric	-	1.03	Position dilution of precision. Variable length. Maximum value: 99.99. Note that this field is 99.99 in case of an invalid value.
<HDOP>	Numeric	-	0.67	Horizontal dilution of precision. Variable length. Maximum value: 99.99. Note that this field is 99.99 in case of an invalid value.
<VDOP>	Numeric	-	0.78	Vertical dilution of precision. Variable length. Maximum value: 99.99. Note that this field is 99.99 in case of an invalid value.
<SystemID>	Numeric	-	1	GNSS system ID. See *Table 11: GNSS Satellites (NMEA) Numbering*. Please note that this parameter is only available in messages in line with NMEA 0183 V4.10 or higher.

Field	Format	Unit	Example	Description
<Checksum>	Hexadecimal	-	*3E	Checksum
<CR><LF>	Character	-	-	Carriage return and line feed.

Example:

NMEA 0183 V3.01 example:

```
$GPGSA,A,3,08,195,194,199,27,21,16,30,,,,,1.09,0.75,0.79*3C
$GLGSA,A,3,86,85,76,,,,,,,,,,1.09,0.75,0.79*18
$GAGSA,A,3,,,,,,,,,,,,,1.09,0.75,0.79*17
$GBGSA,A,3,07,09,36,06,39,,,,,,,,1.09,0.75,0.79*13
```

NMEA 0183 V4.10 example:

```
$GNGSA,A,3,08,195,194,199,27,07,21,16,09,04,,,1.03,0.67,0.78,1*3E
$GNGSA,A,3,85,76,66,,,,,,,,,,1.03,0.67,0.78,2*02
$GNGSA,A,3,,,,,,,,,,,,,1.03,0.67,0.78,3*0F
$GNGSA,A,3,07,40,36,06,16,35,26,,,,,,1.03,0.67,0.78,4*0D
```

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	-	GN	Talker identifier. See *Table 4: NMEA Talker ID*.
VTG	String	-	VTG	Course Over Ground & Ground Speed.
<COGT>	Numeric	Degrees	237.67	Course over ground, in true north direction. Range: 000.00–359.00. 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	0.00	Speed over ground in knots. Variable length. Note that this field is empty in case of an invalid value.
N	Character	-	N	Fixed field: knot.
<SOGK>	Numeric	km/h	0.00	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	-	A	Mode indicator. A = Autonomous mode. Satellite system used for position fixing in non-differential mode. D = Differential mode. Satellite system used for position fixing in differential mode. Corrections from ground stations or Satellite Based Augmentation System (SBAS). E = Estimated (dead reckoning) mode. M = Manual input mode or derived from Survey-in result. N = Data not valid.

Field	Format	Unit	Example	Description
<Checksum>	Hexadecimal	-	*24	Checksum
<CR><LF>	Character	-	-	Carriage return and line feed.

Example:

NMEA 0183 V3.01 example:

```
$GNVTG,0.00,T,,M,0.01,N,0.02,K,A*20
```

NMEA 0183 V4.10 example:

```
$GNVTG,237.67,T,,M,0.00,N,0.00,K,A*24
```

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
$	-	-	$	Each NMEA message starts with **$**.
<TalkerID>	String	-	GN	Talker identifier. See *Table 4: NMEA Talker ID*.
GLL	String	-	GLL	Geographic Position – Latitude/Longitude.
<Lat>	ddmm.mmmmmm	-	3149.332558	Latitude. dd: Degrees (00–90)

Field	Format	Unit	Example	Description
				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	-	11706.912570	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	-	093316.000	Position fix UTC. hh: Hours (00–23) mm: Minutes (00–59) ss: Seconds (00–59) sss: Decimal fraction of seconds
<Status>	Character	-	A	Positioning system status. A = Data valid V = Data not valid
<ModeInd>	Character	-	A	Mode indicator. A = Autonomous mode. Satellite system used for position fixing in non-differential mode . D = Differential mode. Satellite system used for position fixing in differential mode. Corrections from ground stations or Satellite Based Augmentation System (SBAS). E = Estimated (dead reckoning) mode. M = Manual input mode or derived from Survey-in result.

Field	Format	Unit	Example	Description
				N = Data not valid.
<Checksum>	Hexadecimal	-	*45	Checksum
<CR><LF>	Character	-	-	Carriage return and line feed.

Example:

NMEA 0183 V3.01 example:

```
$GNGLL,3149.332006,N,11706.913200,E,093423.000,A,A*4B
```

NMEA 0183 V4.10 example:

```
$GNGLL,3149.332558,N,11706.912570,E,093316.000,A,A*45
```

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	-	GN	Talker identifier. See *Table 4: NMEA Talker ID*.
ZDA	String	-	ZDA	Time & Date. UTC, day, month, year and local time zone.
<UTC>	hhmmss.sss	-	093316.000	Position fix UTC. hh: Hours (00–23)

Field	Format	Unit	Example	Description
				mm: Minutes (00–59) ss: Seconds (00–59) sss: Decimal fraction of seconds
<Day>	Numeric	-	14	Day of month. Range: 01–31.
<Month>	Numeric	-	01	Month. Range: 01–12.
<Year>	Numeric	-	2022	Year.
<LocalHour>	Numeric	-	-	Local zone hours, 00 to ±13 hours. Not supported.
<LocalMin>	Numeric	-	-	Local zone minutes, 00 to 59 minutes. Not supported.
<Checksum>	Hexadecimal	-	*40	Checksum.
<CR><LF>	Character	-	-	Carriage return and line feed.

Example:

NMEA 0183 V3.01 example:

```
$GNZDA,093423.000,14,01,2022,,*41
```

NMEA 0183 V4.10 example:

```
$GNZDA,093316.000,14,01,2022,,*40
```

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	-	GN	Talker identifier. See *Table 4: NMEA Talker ID*.
GRS	String	-	GRS	GNSS range residuals.
<UTC>	hhmmss.sss	-	061549.000	Position fix UTC. hh: Hours (00–23) mm: Minutes (00–59) ss: Seconds (00–59) sss: Decimal fraction of seconds
<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	6.2	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 11: 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 11: 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	-	*6F	Checksum.
<CR><LF>	Character	-	-	Carriage return and line feed.

Example:

NMEA 0183 V3.01 example:

```
$GNGRS,072520.000,1,-4.5,-133,-6.9,2.3,-4.3,-144,-4.9,-4.4,-2.2,-131,-76.7,-3.3*61
$GNGRS,072520.000,1,-4.5,,,,-4.3,,,-4.4,,,,-3.3*64
$GNGRS,072520.000,1,-2.3,0.3,-2.0,,,,,,,,*4C
$GNGRS,072520.000,1,-3.9,-4.6,3.0,-15.1,0.7,,,,,,,*58
$GNGRS,072520.000,1,-3.9,-4.6,3.0,-15.1,0.7,,,,,,,*58
$GNGRS,072520.000,1,-4.1,0.3,-0.2,6.1,-4.3,3.8,-0.2,-10.3,-6.6,1.8,12.2,-15.5*79
$GNGRS,072520.000,1,6.5,-6.3,1.9,-4.1,-141,0.5,,,,,,*5B
$GNGRS,072520.000,1,-4.1,0.3,-0.2,,,,-0.2,,-6.6,,12.2,*55
$GNGRS,072520.000,1,,,1.9,,,0.5,,,,,,*6F
```

NMEA 0183 V4.10 example:

```
$GNGRS,061549.000,1,6.2,-7.5,0.4,-0.7,0.8,-8.6,4.0,4.1,7.7,7.2,2.0,-6.3,1,1*6F
$GNGRS,061549.000,1,-4.8,5.2,-82.4,1.7,,,,,,,,,1,1*5C
$GNGRS,061549.000,1,6.2,-7.5,,,0.8,-8.6,,4.1,,,,,1,8*4D
$GNGRS,061549.000,1,-4.8,5.2,,,,,,,,,,,1,8*40
$GNGRS,061549.000,1,-2.5,11.2,-16.4,-1.5,,,,,,,,,2,1*43
$GNGRS,061549.000,1,-9.8,8.0,8.0,74.7,-16.2,-6.2,-101,,,,,,3,7*5F
$GNGRS,061549.000,1,-9.8,8.0,8.0,74.7,-16.2,-6.2,,,,,,,3,1*44
$GNGRS,061549.000,1,-0.6,-2.0,-6.4,1.0,12.7,7.8,-18.0,3.3,-5.3,16.5,-7.6,-91.2,4,1*4D
$GNGRS,061549.000,1,-3.8,5.9,-0.4,1.1,-107,-5.2,,,,,,,4,1*76
$GNGRS,061549.000,1,-0.6,-2.0,-6.4,,,,,3.3,,,,,4,5*45
$GNGRS,061549.000,1,-3.8,5.9,-0.4,1.1,,,,,,,,,4,5*6D
```

NOTE

1. **GRS** is not supported on LC29H (BA, CA, DA, EA) and LC79H (EL).
2. 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.
3. The calculation method is: Range Residual = Calculated Range - Measured Range.
4. 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>,<STD_RMS>,<STD_Major>,<STD_Minor>,<Orient>,<STD_Lat>,<STD_Lon>,<STD_Alt>*<Checksum><CR><LF>
```

NMEA 0183 V4.10 format (default):

```
$<TalkerID>GST,<UTC>,<STD_RMS>,<STD_Major>,<STD_Minor>,<Orient>,<STD_Lat>,<STD_Lon>,<STD_Alt>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Example	Description
$	Character	-	$	Each NMEA message starts with **$**.
<TalkerID>	String	-	GN	Talker identifier. See *Table 4: NMEA Talker ID*.
GST	String	-	GST	GNSS Pseudorange Error Statistics.
<UTC>	hhmmss.sss	-	061549.000	Position fix UTC. hh: Hours (00–23) mm: Minutes (00–59) ss: Seconds (00–59) sss: Decimal fraction of seconds
<STD_RMS>	Numeric	Meter	8.2	RMS value of the standard deviation of the range inputs to the navigation process.
<STD_Major>	Numeric	Meter	2.6	Standard deviation of semi-major axis of error ellipse.
<STD_Minor>	Numeric	Meter	2.4	Standard deviation of semi-minor axis of error ellipse.
<Orient>	Numeric	Degree	74.7	Orientation of semi-major axis of error ellipse.
<STD_Lat>	Numeric	Meter	2.4	Standard deviation of latitude error.
<STD_Lon>	Numeric	Meter	2.6	Standard deviation of longitude error.
<STD_Alt>	Numeric	Meter	8.5	Standard deviation of altitude error.
<Checksum>	Hexadecimal	-	*45	Checksum.

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

Example:

NMEA 0183 V3.01 example:

```
$GNGST,072520.000,6.5,2.9,2.3,98.2,2.3,2.9,11.2*79
```

NMEA 0183 V4.10 example:

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

NOTE

For LC29H (BA, CA, DA, EA) and LC79H (EL), the number of decimal places for **<STD_RMS>**, **<STD_Major>**, **<STD_Minor>**, **<Orient>**, **<STD_Lat>**, **<STD_Lon>** and **<STD_Alt>** is 3.

2.2.10. GNS

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

Type:

Output

Synopsis:

NMEA 0183 V3.01 format:

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

NMEA 0183 V4.10 format (default):

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

Parameter:

Field	Format	Unit	Example	Description
$	Character	-	$	Each NMEA message starts with **$**.
<TalkerID>	String	-	GN	Talker identifier. See *Table 4: NMEA Talker ID*.

Field	Format	Unit	Example	Description
GNS	String	-	GNS	GNSS Fix Data.
<UTC>	hhmmss.sss	-	023035.000	Position fix UTC. hh: Hours (00–23) mm: Minutes (00–59) ss: Seconds (00–59) sss: Decimal fraction of seconds
<Lat>	ddmm.mmmmmm	-	3149.300445	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	-	11706.920807	Longitude. ddd: Degrees (000–180) mm: Minutes (00–59) mmmmmm: Decimal fraction of minutes. Variable length. Note that this field is empty in case of an invalid value.
<E/W>	Character	-	E	East-west direction. E = East W = West Note that this field is empty in case of an invalid value.
<ModeInd> [1)]	Character	-	AAAANN	Mode indicator. A = Autonomous mode. Satellite system used for position fixing in non-differential mode. D = Differential mode. Satellite system used for position fixing in differential mode. Corrections from ground stations or Satellite Based Augmentation System (SBAS).

Field	Format	Unit	Example	Description
				E = Estimated (dead reckoning) mode. M = Manual input mode or derived from Survey-in result. N = No fix. Satellite system not used for position fixing, or fix not valid.
<NumSatUsed>	Numeric	-	27	Number of satellites in use. Range: 00–99.
<HDOP>	Numeric	-	0.59	Horizontal dilution of precision. Variable length. Maximum value: 99.99. Note that this field is 99.99 in case of an invalid value.
<Alt>	Numeric	Meter	92.193	Antenna altitude above the mean-sea-level (geoid). Note that this field is empty in case of an invalid value.
<Sep>	Numeric	Meter	-0.336	Geoid separation (the difference between the earth ellipsoid surface and the mean-sea-level (geoid) surface defined by the reference datum used in the position solution). Variable length. Note that this field is empty in case of an invalid value.
<DiffAge>	-	-	-	Differential GPS data age. Not supported. Always null.
<DiffStation>	-	-	-	Differential reference station ID. Not supported. Always null.
<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	-	*0B	Checksum.
<CR><LF>	Character	-	-	Carriage return and line feed.

Example:

NMEA 0183 V3.01 example:

```
$GNGNS,063841.000,3149.302984,N,11706.921882,E,AAAAAN,34,0.48,91.149,-0.337,,*78
```

NMEA 0183 V4.10 example:

```
$GNGNS,023035.000,3149.300445,N,11706.920807,E,AAAANN,27,0.59,92.193,-0.336,,,V*0B
```

NOTE

1. [1)] **<ModeInd>** is a variable length field. The first character indicates the use of GPS satellites, the second character indicates the use of GLONASS satellites, and the third character indicates the use of Galileo satellites, the fourth character indicates the use of BDS satellites, the fifth character indicates the use of QZSS satellites, and the sixth character indicates the use of NavIC (IRNSS) satellites.
2. **GNS** message is not supported on LC29H (BA, CA, DA, EA) and LC79H (EL).

2.3. PQTM Messages

This chapter explains the **PQTM** messages (proprietary NMEA messages defined by Quectel) supported by LC29H series and LC79H series GNSS modules.

Table 5: Error Codes

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

2.3.1. PQTMVERNO

Queries the firmware version information.

Type:

Command

Synopsis:

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

Parameter:

None

Result:

- If successful, the module returns:

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

Parameters included in the result:

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

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 5: Error Codes*.

Example:

```
$PQTMVERNO*58
$PQTMVERNO,LC29HAANR01A04S,2022/11/04,16:39:48*34
```

2.3.2. PQTMSAVEPAR

Saves the configurations into NVM.

Type:

Command

Synopsis:

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

Parameter:

None

Result:

- If successful, the module returns:

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

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 5: Error Codes*.

Example:

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

2.3.3. 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><CR><LF>
```

For details about **<ErrCode>**, see *Table 5: Error Codes*.

Example:

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

NOTE

For LC29H (BA, CA, DA, EA) and LC79H (EL) modules, **$PQTMRESTOREPAR** can only restore the parameters configured by **PQTM** commands, **$PAIR050** and **$PAIR062** to their default values. For LC29H (AA) and LC79H (AL), **$PQTMRESTOREPAR** can restore the parameters configured by all commands to their default values.

2.3.4. 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.)
<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
```

2.3.5. 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 6: 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 6: 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 5: Error Codes*.

Example:

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

$PQTMCFGMSGRATE,R,GGA*12
$PQTMCFGMSGRATE,OK,GGA,1,*75

//Set the output rate of **$PQTMGEOFENCESTATUS** to once every position fix:
$PQTMCFGMSGRATE,W,PQTMGEOFENCESTATUS,1,1*5C
$PQTMCFGMSGRATE,OK*29
//Get the output rate of **$PQTMGEOFENCESTATUS**:
$PQTMCFGMSGRATE,R,PQTMGEOFENCESTATUS,1*44
$PQTMCFGMSGRATE,OK,PQTMGEOFENCESTATUS,1,1*0F

Table 6: 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
GNS	1–20
$PQTMEPE	1–20
$PQTMGEOFENCESTATUS	1–20
$PQTMSVINSTATUS	1–20
$PQTMPVT	1–20
$PQTMPL	1–20
$PQTMDOP	1–20

Message Name	Message Output Rate Range (N)
$PQTMVEL	1–20
$PQTMODO	1–20
$PQTMJAMMINGSTATUS	1–20
$PQTMLS	1–20
$PQTMSTD	1–20

NOTE

1. If the configuration message is a **$PQTM** message, use **<MsgVer>** field to specify the message version, otherwise an error will be returned.
2. The maximum output rate of **GSA**, **GSV**, **$PQTMPVT** and **$PQTMDOP** messages on LC29H (BA, CA, DA) is 1 Hz. The configured output rate N is based on 1 Hz. For example, if N is set to 5, the messages will be output every 5 seconds.
3. If the default value is not given for any parameter in a Set command, you can query it with the corresponding Get command provided that the default setting has not been changed by the Set command. If the default setting had been changed by the Set command, contact Quectel Technical Support (support@quectel.com) to get the default setting, if necessary.

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

Result:

- If successful, the module returns:

```
//Response to Set command:
$PQTMCFGGEOFENCE,OK*<Checksum><CR><LF>
//Response to Get command:
$PQTMCFGGEOFENCE,OK,<Index>,<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 5: Error Codes*.

Example:

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

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

2.3.7. 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.)
<Time>	hhmmss.sss	-	UTC time.
Start of repeat block. Repeat times: 4.			
<StateN>	Numeric	-	Geofence N state. Range of N: 0–3. 0 = Unknow 1 = Inside 2 = Outside
End of repeat block.			

Example:

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

2.3.8. PQTMCFGSVIN

Sets/gets the survey-in feature.

In order to operate as a base station, the module external antenna should be mounted on a fix point. The antenna accurate coordinate location can be acquired through a self-survey process. The Survey-in mode (**<Mode>** = 1) determines the receiver's position by building a weighted mean of all valid 3D positioning solutions. You can set values of **<MinDur>** and **<3D_AccLimit>** to define the minimum observation time and 3D position standard deviation used for the position estimation. The Fixed mode (**<Mode>** = 2) requires user to manually enter the receiver position coordinates. Any error in the base station position will translate directly into rover position error.

Type:

Set/Get

Synopsis:

```
//Set:
$PQTMCFGSVIN,W,<Mode>,<MinDur>,<3D_AccLimit>,<ECEF_X>,<ECEF_Y>,<ECEF_Z>[,<Distance>]*<Checksum><CR><LF>
//Get:
$PQTMCFGSVIN,R*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Mode>	Numeric	-	Configure the receiver mode. 0 = Disable 1 = Survey-in mode 2 = Fixed mode (ARP position is given in ECEF.)
<MinDur>	Numeric	-	Minimum position fix times in survey-in mode. Range: 0–86400.
<3D_AccLimit>	Numeric	Meter	Limit the 3D position accuracy in survey-in mode. When this field is 0, it means no limit on 3D position accuracy.
<ECEF_X>	Numeric	Meter	WGS84 ECEF X coordinate.
<ECEF_Y>	Numeric	Meter	WGS84 ECEF Y coordinate.
<ECEF_Z>	Numeric	Meter	WGS84 ECEF Z coordinate.
<Distance>	Numeric	Meter	When the calculated value is greater than or equal to the set value, the current Survey-in result is

Field	Format	Unit	Description
			output; when the calculated value is less than the set value, the previous Survey-in result is output. The set value only works when **<Mode>** = 1. Range: 0.0–10.0. Default value: 0.0 (always use the latest survey-in result).

Result:

- If successful, the module returns:

```
//Response to Set command:
$PQTMCFGSVIN,OK*<Checksum><CR><LF>
//Response to Get command:
$PQTMCFGSVIN,OK,<Mode>,<MinDur>,<3D_AccLimit>,<ECEF_X>,<ECEF_Y>,<ECEF_Z>,<Distance>*<Checksum><CR><LF>
```

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 5: Error Codes*.

Example:

```
//Set (without <Distance>):
$PQTMCFGSVIN,W,1,3600,1.2,-2519265.0514,4849534.9045,3277834.6432*01
$PQTMCFGSVIN,OK*70

//Set (with <Distance>):
$PQTMCFGSVIN,W,1,3600,1.2,-2519265.0514,4849534.9045,3277834.6432,2.5*04
$PQTMCFGSVIN,OK*70

//Get:
$PQTMCFGSVIN,R*26
$PQTMCFGSVIN,OK,1,3600,1.2,-2519265.0514,4849534.9045,3277834.6432,2.5*57
```

NOTE

1. This command is supported on LC29H (BA, CA, DA, EA) and LC79H (EL).
2. The configuration set by this command takes effect after executing **$PQTMSAVEPAR** and then restarting the module.

3. When **<Mode>** = 1 (survey-in mode), **<ECEF_X>**, **<ECEF_Y>** and **<ECEF_Z>** should be set to 0.

2.3.9. PQTMSVINSTATUS

Outputs the Survey-in status.

Type:

Output

Synopsis:

```
$PQTMSVINSTATUS,<MsgVer>,<TOW>,<Valid>,<Res0>,<Res1>,<Obs>,<CfgDur>,<MeanX>,<MeanY>,<MeanZ>,<MeanAcc>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<MsgVer>	Numeric	-	Message version. 1 = Version 1 (Always 1 for this message.)
<TOW>	Numeric	Millisecond	GPS time of week.
<Valid>	Numeric	-	Survey-in position validity flag. 0 = Invalid 1 = In-progress 2 = Valid
<Res0>	Numeric	-	Reserved.
<Res1>	Numeric	-	Reserved.
<Obs>	Numeric	-	Number of position observations used during survey-in.
<CfgDur>	Numeric	-	Duration configured via the **<MinDur>** field of **$PQTMCFGSVIN** command.
<MeanX>	Numeric	Meter	Current survey-in mean position along X axis of ECEF coordinate system.
<MeanY>	Numeric	Meter	Current survey-in mean position along Y axis of ECEF coordinate system.
<MeanZ>	Numeric	Meter	Current survey-in mean position along Z axis of ECEF coordinate system.
<MeanAcc>	Numeric	Meter	Current survey-in mean position accuracy.

Example:

```
$PQTMSVINSTATUS,1,2241,1,,01,538,43200,-2472436.0802,4828383.0026,3343698.4839,9.5*38
```

NOTE

This command is supported on LC29H (BA, CA, DA, EA) and LC79H (EL).

2.3.10. 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 5: Error Codes*.

Example:

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

2.3.11. 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 5: Error Codes*.

Example:

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

NOTE

This command is only used to stop GNSS engine, and the DR engine keep working if it is enabled.

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.)
<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
<Res>	Numeric	-	Reserved.
<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

Field	Format	Unit	Description
			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.
<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	-	Configure the number of decimal places for UTC seconds in NMEA standard messages. Range: 0–3. Default value: 3. 0 = No fractional part.
<POS_DP>	Numeric	-	Configure the number of decimal places for latitude and longitude in NMEA standard messages. Range: 0–8. Default value: 6. 0 = No fractional part.
<ALT_DP>	Numeric	-	Configure the 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	-	Configure the number of decimal places for DOP in NMEA standard messages. Range: 0–3. Default value: 2. 0 = No fractional part.
<SPD_DP>	Numeric	-	Configure the number of decimal places for speed in NMEA standard messages. Range: 0–3. Default value for LC29H (AA) and LC79H (AL): 2; Default value for LC29H (BA, CA, DA, EA) and LC79H (EL): 3. 0 = No fractional part.
<COG_DP>	Numeric	-	Configure the 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 5: Error Codes*.

Example:

```
//Set:
$PQTMCFGNMEADP,W,3,6,1,2,3,2*35
$PQTMCFGNMEADP,OK*61
//Get:
$PQTMCFGNMEADP,R*37
$PQTMCFGNMEADP,OK,3,6,1,2,3,2*66
```

2.3.14. PQTMCFGRCVRMODE

Sets/gets the receiver working mode.

Type:

Set/Get

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<Mode>	Numeric	-	Receiver working mode. 0 = Unknown. 1 = Rover. When set the module to this mode, the receiver will restore to default NMEA messages output state. 2 = Base station. When set the module to this mode, the receiver will automatically disable NMEA messages output and enable RTCM MSM4, 1005 messages output.

Result:

- If successful, the module returns:

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

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 5: Error Codes*.

Example:

```
//Set:
$PQTMCFGRCVRMODE,W,2*29
$PQTMCFGRCVRMODE,OK*64

//Get:
$PQTMCFGRCVRMODE,R*32
$PQTMCFGRCVRMODE,OK,2*7A
```

NOTE

1. This command is supported on LC29H (BA, CA, DA, EA) and LC79H (EL).
2. The configuration set by this command takes effect after executing **$PQTMSAVEPAR** and then restarting the module.

2.3.15. 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).
<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	Protection level of time. Null if invalid.

Example:

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

2.3.16. PQTMCOLD

Performs a cold start, which restarts the GNSS engine without using some location information, including position 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 5: Error Codes*.

Example:

```
$PQTMCOLD*1C
```

2.3.17. 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 5: Error Codes*.

Example:

```
$PQTMHOT*4B
```

2.3.18. 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 5: Error Codes*.

Example:

```
$PQTMWARM*11
```

2.3.19. 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 5: Error Codes*.

Example:

```
$PQTMSRR*4B
```

NOTE

This command is only supported on LC29H (AA) and LC79H (AL).

2.3.20. 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 5: Error Codes*.

Example:

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

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

NOTE

This command is only supported on LC29H (AA) and LC79H (AL).

2.3.21. 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 5: 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 on LC29H (AA) and LC79H (AL):
   - 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 on LC29H (AA) and LC79H (AL).
2. Supported constellation configurations on LC29H (BA, CA, DA, EA) and LC79H (EL) (currently, NavIC is unsupported on LC29H (DA, EA) and LC79H (EL)):
   - GPS + GLONASS + Galileo + BDS + NavIC + QZSS
   - GPS + GLONASS + Galileo + BDS + NavIC
   - GPS + GLONASS + Galileo + BDS + QZSS
   - GPS + Galileo + BDS + QZSS
   - GPS + Galileo + NavIC + QZSS
   - GPS + Galileo + QZSS
   - GPS + Galileo + NavIC
   - GPS + BDS + QZSS
   - GPS + BDS
   - GPS + QZSS
   - BDS
   - GPS
   - NavIC
3. The configuration set by this command takes effect after executing **$PQTMSAVEPAR** and then restarting the module.

2.3.22. 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).
<TOW>	Numeric	Millisecond	Time of week. Null if invalid.
<GDOP>	Numeric	-	Geometric dilution of precision. Note that the value is 99.99 in case of an invalid value.
<PDOP>	Numeric	-	Position (3D) dilution of precision. Note that the value is 99.99 in case of an invalid value.
<TDOP>	Numeric	-	Time dilution of precision. Note that the value is 99.99 in case of an invalid value.
<VDOP>	Numeric	-	Vertical dilution of precision. Note that the value is 99.99 in case of an invalid value.
<HDOP>	Numeric	-	Horizontal dilution of precision. Note that the value is 99.99 in case of an invalid value.
<NDOP>	Numeric	-	Northing dilution of precision. Note that the value is 99.99 in case of an invalid value.
<EDOP>	Numeric	-	Easting dilution of precision. Note that the value is 99.99 in case of an invalid value.

Example:

```
//Fixed:
$PQTMDOP,1,570643000,1.01,0.88,0.49,0.73,0.50,0.36,0.35*7C

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

2.3.23. 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> 1)	Numeric	ms	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 5: Error Codes*.

Example:

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

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

NOTE

1. [1)] For LC29H (DA), **<Interval>** can only be set to 1000.
2. For LC29H (AA) and LC79H (AL), this command takes effect immediately. For LC29H (BA, CA, EA) and LC79H (EL), the configuration set by this command takes effect after executing **$PQTMSAVEPAR** and then restarting the module.

2.3.24. PQTMCFGPPS

Sets/gets the 1PPS feature configuration.

Type:

Set/Get

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<Index>	Numeric	-	PPS index. 1 = PPS 1
<Enable>	Numeric	-	Enable/disable PPS output. 0 = Disable 1 = Enable
<Duration>	Numeric	Millisecond	Pulse duration. Range: 1–999. Default value: 100.

Field	Format	Unit	Description
<Mode>	Numeric	-	PPS output with fix mode. 1 = Always 2 = 2D fix 3 = 3D fix
<Polarity>	Numeric	-	Pulse polarity. 0 = Low 1 = High
<Reserved>	Numeric	-	Reserved. Fixed to 0.

Result:

- If successful, the module returns:

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

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 5: Error Codes*.

Example:

```
//Set:
$PQTMCFGPPS,W,1,1,100,1,1,0*73
$PQTMCFGPPS,OK*21

//Get:
$PQTMCFGPPS,R,1*6A
$PQTMCFGPPS,OK,1,1,100,1,1,0*20
```

NOTE

This command is only supported on LC29H (AA) and LC79H (AL).

2.3.25. PQTMDEBUGON

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

Type:

Command

Synopsis:

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

Parameter:

None

Result:

- If successful, the module returns:

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

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 5: Error Codes*.

Example:

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

2.3.26. PQTMDEBUGOFF

Disables debug logging.

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 5: Error Codes*.

Example:

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

2.3.27. 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.
<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.

Field	Format	Unit	Description
<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
```

2.3.28. PQTMCFGODO

Sets/gets the odometer feature configuration.

Type:

Set/Get

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<State>	Numeric	-	Odometer feature state. 0 = Disabled 1 = Enabled
<InitDist>	Numeric	Meter	Initial distance. Default value: 0.0. Range: -1000000000.0 to 1000000000.0.

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 5: Error Codes*.

Example:

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

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

2.3.29. 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 5: 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.

2.3.30. 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 = Accumulate mileage via GNSS position. 2 = Accumulate mileage via injected wheel ticks.
<Time>	hhmmss.sss	-	UTC time. hh: Hour (00–23) mm: Minute (00–59) ss: Second (00–59) sss: Decimal fraction of seconds
<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 mileage starts from 0 m and resets to 0 m after a power outage or when cleared with **$PQTMRESETODO**. If **<InitDist>** value in **$PQTMCFGODO** is modified, the actual **<Dist>** output in **$PQTMODO** reflects the sum of the accumulated mileage and the new **<InitDist>** value, as shown below:
   <Dist> = Accumulated Mileage + **<InitDist>**.
2. There are two ways to accumulate mileage for **$PQTMODO**, and the accumulation principles are different.
   When **<MsgVer>** = 1, mileage is accumulated via GNSS position.
   When **<MsgVer>** = 2, mileage is accumulated via injected wheel ticks, which is related to the DR feature.
   - **$PQTMODO (<MsgVer> = 2)** accumulates mileage when DR completes wheel tick pulse scale estimation calibration. Before calibration, the value of **<Dist>** is the initial distance. Once the calibration is complete, the mileage accumulated before calibration will be added.
   - If the carrier or installation position of the module is changed, you must execute **$PQTMDRCLR** to clear the wheel tick pulse estimation information before the next test.
   - For details about **$PQTMDRCLR**, see *document [1] DR&RTK application note*. For additional support, contact Quectel Technical Support (support@quectel.com).
3. **$PQTMODO (<MsgVer> = 2)** is only supported on LC29H (BA) and LC29H (CA) with firmware versions for four-wheel vehicles.

2.3.31. 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.)
<Status>	Numeric	-	Jamming detection status. 0 = Unknown 1 = No jamming, healthy status 2 = Warning status

Field	Format	Unit	Description
			3 = Critical status

Example:

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

NOTE

This message is only supported on LC29H (AA) and LC79H (AL).

2.3.32. 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.)
<TOW>	Numeric	Second	Time of week.
<LS_Ref>	Hexadecimal	-	Referenced constellation by the current leap second information: 0 = No source 1 = GPS 2 = GLONASS (not supported) 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

Field	Format	Unit	Description
			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 1 = GPS 2 = GLONASS (not supported) 3 = Galileo 4 = BDS Field value is invalid if **<Flag>** = 0.
<Reserved>	-	-	Reserved. Always null.
<WNLSF>	Numeric	-	Week number of the new leap second. 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. Field value is invalid if **<Flag>** = 0.
<LSF>	Numeric	Second	Leap second count after future leap second changes. Field value is invalid if **<Flag>** = 0.

Example:

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

NOTE

1. This message is only supported on LC79H (AL).
2. 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.
3. 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.
4. <**LSF**> minus <**LS**> can have one of the following three values, respectively:
   - +1 = Positive leap second;

- -1 = Negative leap second;
- 0 = No future leap second event scheduled or no information available.

5. Reference priority for leap second information: GPS > Galileo > BDS.

2.3.33. PQTMCFGRSID

Sets/gets the reference station ID.

Type:

Set/Get

Synopsis:

```
//Set:
$PQTMCFGRSID,W,<ID>*<Checksum><CR><LF>
//Get:
$PQTMCFGRSID,R*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<ID>	Numeric	-	Reference station ID. Range: 0–4095.

Result:

- If successful, the module returns:

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

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 5: Error Codes*.

Example:

```
//Set:
$PQTMCFGRSID,W,1024*06
$PQTMCFGRSID,OK*7E
```

```
//Get:
$PQTMCFGRSID,R*28
$PQTMCFGRSID,OK,1024*55
```

NOTE

This message is only supported on LC29H (AA) and LC79H (AL).

2.3.34. PQTMCFGNMEATID

Sets/gets the NMEA Talker ID.

Type:

Set/Get

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<Main_TalkerID>	Character	-	Main Talker ID, which is used for all standard NMEA messages other than **GSV**. 00 = Automatic mode. The main talker ID is determined by the GNSS constellation configuration. If it is not "00", 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 **<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 5: 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
```

NOTE

This message is only supported on LC29H (AA) and LC79H (AL).

2.3.35. 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 5: Error Codes*.

Example:

```
$PQTMUNIQID*16
$PQTMUNIQID,OK,16,81D62010EE0AF375BDF5952CDC3757A1*3E
```

NOTE

This message is only supported on LC29H (AA) and LC79H (AL).

2.3.36. PQTMVER

Outputs the firmware version. The message is immediately output after the module is powered on, and it is only output once after the power-on.

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).
<VerName>	String	-	Version name. Fixed to “MODULE”.
<VerStr>	String	-	Version string.
<BuildDate>	yyyy/mm/dd	-	Firmware build date.
<BuildTime>	hh:mm:ss	-	Firmware build time.

Example:

```
$PQTMVER,1,MODULE,LC79HALNR11A04S,2024/07/05,15:59:32*1F
```

NOTE

This message is only supported on LC29H (AA) and LC79H (AL).

2.3.37. PQTMCFGUART

Sets/gets UART configuration.

Type:

Set/Get

Synopsis:

```
//Set the current UART port:
$PQTMCFGUART,W,<BaudRate>[,<DataBit>,<Parity>,<StopBit>,<FlowCtrl>]*<Checksum><CR><LF>
//Set a specified UART port:
$PQTMCFGUART,W,<Index>,<BaudRate>[,<DataBit>,<Parity>,<StopBit>,<FlowCtrl>]*<Checksum><CR><LF>
//Get the configuration 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. Supported values: 4800 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 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 5: 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 of 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. This message is only supported on LC29H (AA) and LC79H (AL).
2. The configuration set by this command takes effect after executing **$PQTMSAVEPAR** and then restarting the module.

2.3.38. PQTMCFGPROT

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

Type:

Set/Get

Synopsis:

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

Parameter:

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

Result:

- If successful, the module returns:

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

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 5: Error Codes*.

Example:

```
//Set:
$PQTMCFGPROT,W,1,1,5,1*3C
$PQTMCFGPROT,OK*6B

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

NOTE

1. This message is only supported on LC29H (AA) and LC79H (AL).
2. Only LC29H (AA) supports the SPI.

2.3.39. PQTMBKP

Shuts down 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, module remains in

Backup mode indefinitely.

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 5: 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
```

NOTE

This message is only supported on LC29H (AA) and LC79H (AL).

2.3.40. PQTMGETUTC

Obtains the current UTC time information.

Type:

Command

Synopsis:

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

Parameter:

None

Result:

- If successful, the module returns:

```
$PQTMGETUTC,OK,<Year>,<Month>,<Day>,<Hour>,<Minute>,<Second>,<Millisecond>,<Reserved>,<Leapsecond>*<Checksum><CR><LF>
```

Parameters included in the result:

Field	Format	Unit	Description
<Year>	Numeric	-	Year.

Field	Format	Unit	Description
<Month>	Numeric	-	Month.
<Day>	Numeric	-	Day of month.
<Hour>	Numeric	-	Hours.
<Minute>	Numeric	-	Minutes.
<Second>	Numeric	-	Seconds.
<Millisecond>	Numeric	-	Milliseconds.
<Reserved>	-	-	Reserved. Always null.
<Leapsecond>	Numeric	-	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.

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 5: Error Codes*.

Example:

```
$PQTMGETUTC*0C
$PQTMGETUTC,OK,2024,10,22,2,52,30,295,,18*0C
```

NOTE

This message is only supported on LC29H (BA, CA).

2.3.41. PQTMCFGNAVMODE

Sets/gets the navigation mode.

Type:

Set/Get

Synopsis:

```
//Set:
```

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

Parameter:

Field	Format	Unit	Description
<Mode>	Numeric	-	Navigation mode. 0 = Normal mode. It is a basic mode that applies to most scenarios (for example, driving scenarios). 1–9 = Reserved. 10 = Train mode. It is suitable for train scenarios.

Result:

- If successful, the module returns:

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

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 5: Error Codes*.

Example:

```
//Set the navigation mode:
$PQTMCFGNAVMODE,W,10*56
$PQTMCFGNAVMODE,OK*28

//Get the navigation mode:
$PQTMCFGNAVMODE,R*7E
$PQTMCFGNAVMODE,OK,10*05
```

NOTE

1. This command is supported on LC29H (CA).
2. The configuration set by this command takes effect after executing **$PQTMSAVEPAR** and then restarting the module.

2.3.42. PQTMQVER

Queries the version information.

Type:

Command

Synopsis:

```
$PQTMQVER[,<VerID>]*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<VerID>	Numeric	-	Version ID. 1 = Firmware version.

Result:

If successful, the module returns:

```
$PQTMQVER,OK,<VerID>,<DesStr>,<VerStr>,<BuildDate>,<BuildTime>*<Checksum><CR><LF>
```

Parameters included in the result:

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

If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 5: Error Codes*.

Example:

```
$PQTMQVER,1*15
$PQTMQVER,OK,1,MODULE,LC29HCANR11A07S_DSA4,2025/06/03,15:28:01*52
```

$PQTMQVER*08

$PQTMQVER,OK,1,MODULE,LC29HCANR11A07S_DSA4,2025/06/03,15:28:01*52

NOTE

This command is supported on LC29H (BA, CA).

2.3.43. PQTMSN

Queries module SN code.

Type:

Command

Synopsis:

$PQTMSN*<Checksum><CR><LF>

Parameter:

None

Result:

If successful, the module returns:

$PQTMSN,OK,<ID>,<Length>,<SN>*<Checksum><CR><LF>

Parameters included in the result:

Field	Format	Unit	Description
<ID>	Numeric	-	SN ID. Fixed to 1.
<Length>	Numeric	-	Length of the SN.
<SN>	String	-	Module SN code.

If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 5: Error Codes*.

Example:

```
$PQTMSN*05
$PQTMSN,OK,1,16,1234567890ABCDEF*31
```

NOTE

This command is supported on LC29H (AA) and LC79H(AL).

2.3.44. PQTMSTD

Outputs time, position deviations, velocity deviations and attitude deviations.

Type:

Output

Synopsis:

```
$PQTMSTD,<Reserved>,<UTC>,<WN>,<TOW>,<STD_Lat>,<STD_Lon>,<STD_Alt>,<STD_Sep>,<STD_VelN>,<STD_VelE>,<STD_VelD>,<STD_Spd>,<STD_Roll>,<STD_Pitch>,<STD_Heading>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Reserved>	-	-	Reserved. Fixed to 1.
<UTC>	hhmmss.sss	-	UTC time. hh: Hours (00–23) mm: Minutes (00–59) ss: Seconds (00–59) sss: Decimal fraction of seconds
<WN>	Numeric	-	Week number, counted from 00:00:00 on January 6, 1980. Note that this field is 0 in case of an invalid value.
<TOW>	Numeric	Millisecond	Milliseconds of week. Range: 0–604799999.
<STD_Lat>	Numeric	Meter	Standard deviation of latitude error. Note that this field is empty in case of an invalid value.
<STD_Lon>	Numeric	Meter	Standard deviation of longitude error. Note that this field is empty in case of an invalid value.

Field	Format	Unit	Description
<STD_Alt>	Numeric	Meter	Standard deviation of altitude above mean-sea-level error. Note that this field is empty in case of an invalid value.
<STD_Sep>	Numeric	Meter	Standard deviation of geoidal separation error. Note that this field is empty in case of an invalid value.
<STD_VelN>	Numeric	m/s	Standard deviation of north velocity error. Note that this field is empty in case of an invalid value.
<STD_VelE>	Numeric	m/s	Standard deviation of east velocity error. Note that this field is empty in case of an invalid value.
<STD_VelD>	Numeric	m/s	Standard deviation of down velocity error. Note that this field is empty in case of an invalid value.
<STD_Spd>	Numeric	m/s	Standard deviation of ground speed error. Note that this field is empty in case of an invalid valuer.
<STD_Roll>	Numeric	Degree	Standard deviation of roll angle. Note that this field is empty in case of an invalid value.
<STD_Pitch>	Numeric	Degree	Standard deviation of pitch angle. Note that this field is empty in case of an invalid value.
<STD_Heading>	Numeric	Degree	Standard deviation of heading angle. Note that this field is empty in case of an invalid value.

Example:

```
$PQTMSTD,1,131137.000,2368,393115,5.36,4.62,12.12,,0.13,0.13,0.14,0.23,,,*77
$PQTMSTD,1,131523.314,0,393341,,,,,,,,,,*66
```

NOTE

This message is only supported on LC79H (AL).

2.3.45. PQTMCFGAIC

Sets/gets the AIC (Active Interference Cancellation) function.

Type:

Set/Get

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<State>	Numeric	-	AIC status. 0 = Disabled 1 = Enabled

Result:

- If successful, the module returns:

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

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 5: Error Codes*.

Example:

```
//Set AIC:
$PQTMCFGAIC,W,1*77
$PQTMCFGAIC,OK*39

//Get AIC status:
$PQTMCFGAIC,R*6F
```

```
$PQTMCFGAIC,OK,1*24
```

NOTE

This message is only supported on LC79H (AL).

2.4. PAIR Messages

This chapter explains **PAIR** messages (proprietary NMEA messages defined by the chipset supplier) supported by LC29H series and LC79H series GNSS modules.

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	-	Result. 0 = Command has been successfully sent. 1 = Command is being processed. Please wait for the result. 2 = Command sending failed.

Field	Format	Unit	Description
			3 = <CommandID> is not supported 4 = Command parameter error. Out of range/Some parameters were lost/Checksum error. 5 = MNL service is busy. You can try again soon.

Example:

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

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
```

NOTE

Wait more than two seconds before powering on/off the GNSS system to avoid system anomalies.

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
```

NOTE

Wait more than two seconds before powering on/off the GNSS system to avoid system anomalies.

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) and its ephemeris is still valid. Therefore, there is no need to download an ephemeris 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 an ephemeris 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 will not use some valid location information, including position 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
<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
```

2.4.9. 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 for LC29H (EA) and LC79H (EL): 100; default value for other modules: 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. For LC29H (BA, CA, EA) and LC79H (EL), if the output frequency is set to be greater than 1 Hz, only **RMC** and **GGA** massages will be output at the set frequency, whereas **GSA** and **GSV** messages maintain a 1 Hz output frequency. Other NMEA messages will not be output.
3. For LC29H (DA), **<Time>** can only be set to 1000.
4. To avoid system abnormalities, send the save command and restart the module after configuring parameters for the configuration to take effect.
   - For LC29H (AA) and LC79H (AL) (when **<Time>** < 1000):
     1) First, execute **$PAIR382,1*2E** and **$PAIR003*39** in sequence to power off the GNSS system.
     2) Then, execute **$PAIR050** to configure the position fix interval.
     3) Send **$PAIR513*3D** or **$PQTMSAVEPAR*5A** to save the configuration.
     4) Finally, send **$PAIR002*38** to re-power on the GNSS system.

     The configuration takes effect **immediately** upon completion of the above operations.
   - For LC29H (EA) and LC79H (EL):
     1) First, execute **$PAIR382,1*2E** and **$PAIR003*39** in sequence to power off the GNSS system.
     2) Then, execute **$PAIR050** to configure the position fix interval.
     3) Send **$PAIR513*3D** or **$PQTMSAVEPAR*5A** to save the configuration.
     4) Finally, send **$PAIR002*38** to re-power on the GNSS system.

     The configuration takes effect **after an additional restart** upon completion of the above operations.
   - For other modules or other scenarios, every time the position fix interval is set, the command **$PAIR513*3D** or **$PQTMSAVEPAR*5A** must be sent to save the configuration. This configuration takes effect after restart.

2.4.10. 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
```

2.4.11. 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
```

NOTE

This command is only supported on LC29H (AA) and LC79H (AL).

2.4.12. 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
```

NOTE

This command is only supported on LC29H (AA) and LC79H (AL).

2.4.13. PAIR062: PAIR_COMMON_SET_NMEA_OUTPUT_RATE

Sets the output rate of standard NMEA messages 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

Field	Format	Unit	Description
			9 = NMEA_SEN_GNS
<OutputRate>	Numeric	-	Message output rate setting. 0 = Disable sentence output N = Output message once every N position fix(es) Range of N: 0–20.

Result:

Returns **$PAIR001** message.

Example:

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

NOTE

1. LC29H (BA, CA, DA, EA) and LC79H (EL) only support setting **<Type>** to 0–5.
2. LC29H (EA) only supports setting **<OutputRate>** to 0 or 1.
3. **GGA**, **GLL**, **GSA**, **GSV**, **RMC** and **VTG** messages are output by default.

2.4.14. PAIR063: PAIR_COMMON_GET_NMEA_OUTPUT_RATE

Gets the output rate of standard NMEA messages 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

Field	Format	Unit	Description
			4 = NMEA_SEN_RMC 5 = NMEA_SEN_VTG 6 = NMEA_SEN_ZDA 7 = NMEA_SEN_GRS 8 = NMEA_SEN_GST 9 = NMEA_SEN_GNS

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 9 = NMEA_SEN_GNS
<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
```

NOTE

LC29H (BA, CA, DA, EA) and LC79H (EL) only support setting **<Type>** to 0–5.

2.4.15. 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:

Field	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 satellites) 1 = Search for NavIC satellites

Result:

Returns **$PAIR001** message.

Example:

```
//Switch to GPS + Galileo + BDS:
$PAIR066,1,0,1,1,0,0*3B
$PAIR001,066,0*3B
```

NOTE

1. This command is only supported on LC29H (AA) and LC79H (AL).
2. Supported constellation configurations on LC29H (AA) and LC79H (AL):
   - 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.

2.4.16. 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:

Field	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 satellites) 1 = Search for NavIC satellites

Example:

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

NOTE

This command is only supported on LC29H (AA) and LC79H (AL).

2.4.17. 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: 0–20. Default value: 0.

Result:

Returns **$PAIR001** message.

Example:

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

NOTE

This command is only supported on LC29H (AA) and LC79H (AL).

2.4.18. 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	Static navigation speed threshold. Range: 0–2. Default value: 0.

Example:

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

NOTE

This command is only supported on LC29H (AA) and LC79H (AL).

2.4.19. PAIR072: PAIR_COMMON_SET_ELEV_MASK

Sets the 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

1. The satellites below the elevation mask are not used for positioning.
2. This command is only supported on LC29H (AA) and LC79H (AL).

2.4.20. 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. Default value: 5.

Example:

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

NOTE

This command is only supported on LC29H (AA) and LC79H (AL).

2.4.21. PAIR074: PAIR_COMMON_SET_AIC_ENABLE

Enables/disables the active interference cancellation (AIC) function. For details about AIC function, see *documents [2]* and *[3] hardware designs*.

Type:

Set

Synopsis

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

Parameter:

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

Result:

Returns **$PAIR001** message.

Example:

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

2.4.22. 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
```

2.4.23. 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. It is a basic mode that applies to most scenarios. (for example, driving scenarios). 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

Field	Format	Unit	Description
			vertical movement has a greater impact on the position calculation. 4 = Stationary mode. For stationary applications where zero dynamics is assumed. 5 = Dynamic flight 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 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 7: Altitude and Speed Ranges of Navigation Modes

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

Result:

Returns **$PAIR001** message.

Example:

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

NOTE

This command is only supported on LC29H (AA) and LC79H (AL).

2.4.24. 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. It is a basic mode that applies to most scenarios. (for example, driving scenarios). 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 zero dynamics is assumed.

Field	Format	Unit	Description
			5 = Dynamic flight 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
```

NOTE

This command is only supported on LC29H (AA) and LC79H (AL).

2.4.25. 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
```

2.4.26. 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
```

2.4.27. 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
```

NOTE

For LC29H (BA, CA, DA, EA) and LC79H (EL), standard NMEA sentences are output in NMEA 0183 V4.10 format and this command is not supported.

2.4.28. 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
```

NOTE

This command is only supported on LC29H (AA) and LC79H (AL).

2.4.29. PAIR104: PAIR_COMMON_SET_DUAL_BAND

Enables/disables the dual band feature only when the GNSS system is powered off.

Type:

Set

Synopsis:

```
$PAIR104,<DualBandEnabled>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<DualBandEnabled>	Numeric	-	Enable/disable the dual band feature. 0 = Disable 1 = Enable

Result:

Returns **$PAIR001** message.

Example:

```
//Switch from dual-band to single-band:
$PAIR382,1*2E
$PAIR001,382,0*32
$PAIR003*39
$PAIR001,003,0*38
$PAIR104,0*23
$PAIR001,104,0*3E
$PAIR002*38
$PAIR001,002,0*39
```

NOTE

1. This command is only supported on LC29H (AA) and LC79H (AL).
2. Before setting the dual band feature by **$PAIR104**, execute **$PAIR382,1*2E** and **$PAIR003*39** in sequence to power off the GNSS system. Once the dual band feature is set, send **$PAIR002*38** to power on the GNSS system again.

2.4.30. PAIR105: PAIR_COMMON_GET_DUAL_BAND

Queries whether the dual band feature is enabled or disabled.

Type:

Get

Synopsis:

```
$PAIR105*<Checksum><CR><LF>
```

Parameter:

None

Result:

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

Query result message format:

```
$PAIR105,<Enabled>*<Checksum><CR><LF>
```

Parameter included in the result:

Field	Format	Unit	Description
<Enabled>	Numeric	-	Status of the dual band feature. 0 = Disabled 1 = Enabled

Example:

```
$PAIR105*3E
$PAIR001,105,0*3F
$PAIR105,1*23
```

NOTE

This command is only supported on LC29H (AA) and LC79H (AL).

2.4.31. 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

1. 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. The module can still receive commands if it receives **$PAIR382,1*2E** before **$PAIR003*39** and **$PAIR001,382,0*32** is returned correctly. Otherwise, any other commands cannot be received.

2.4.32. PAIR391: PAIR_TEST_JAMMING_DETECT

Enables/disables jamming detection. Jamming status messages will be returned when 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 **$PAIR001** message and enables **$PAIRSPF** message output periodically (at 1 Hz).

Query result message format:

```
$PAIRSPF,<Status>*<Checksum><CR><LF>
$PAIRSPF5,<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
$PAIRSPF5,0*66

//Good status:
$PAIRSPF,1*52
$PAIRSPF5,1*67

//Warning status:
$PAIRSPF,2*51
$PAIRSPF5,2*64

//Critical status:
$PAIRSPF,3*50
$PAIRSPF5,3*65
```

NOTE

1. The **$PAIRSPF,<Status>*<Checksum><CR><LF>** sentence indicates L1 jamming status.
2. The **$PAIRSPF5,<Status>*<Checksum><CR><LF>** sentence indicates L5 jamming status. This message will not be output when only L1 band signals are received and tracked.
3. The module starts jamming detection once the feature is enabled.

- If there is no jamming, **$PAIRSPF,1*52**, or **$PAIRSPF,1*52** and **$PAIRSPF5,1*62**, will be reported to indicate good status (**<Status>** = 1).
- In case of continuous jamming, the jamming status will change from 1 to 2 and finally to 3.
  1) When no position fix has been completed: module status is 1 right after the 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.
  2) After a successful position fix: jamming status is 1 right after jamming detection is enabled, and changes to 2 and 3 consecutively when jamming is detected.

2.4.33. 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 data source 2 = SBAS (including WAAS/EGNOS/GAGAN/MSAS) 3 = SLAS

Result:

Returns **$PAIR001** message.

Example:

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

NOTE

This command is only supported on LC29H (AA) and LC79H (AL).

2.4.34. 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 data source 2 = SBAS (including WAAS/EGNOS/GAGAN/MSAS) 3 = SLAS

Example:

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

NOTE

This command is only supported on LC29H (AA) and LC79H (AL).

2.4.35. PAIR410: PAIR_SBAS_ENABLE

Enables/disables SBAS satellite search. 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 or disable the search of SBAS satellites. 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 command **$PAIR080**), SBAS is not supported.

2.4.36. PAIR411: PAIR_SBAS_GET_STATUS

Queries the status of SBAS satellite search.

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 search. 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.

2.4.37. PAIR420: PAIR_SLAS_ENABLE

Enables/disables SLAS satellite search.

Type:

Set

Synopsis

```
$PAIR420,<Enabled>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Enabled>	Numeric	-	Enable or disable the search of SLAS satellites. 0 = Disable 1 = Enable

Result:

Returns **$PAIR001** message.

Example:

```
$PAIR420,1*21
$PAIR001,420,0*3D
```

NOTE

This command is only supported on LC29H (AA) and LC79H (AL).

2.4.38. PAIR421: PAIR_SLAS_GET_STATUS

Queries the status of SLAS satellite search.

Type:

Get

Synopsis

```
$PAIR421*<Checksum><CR><LF>
```

Parameter:

None

Result:

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

Query result message format:

```
$PAIR421,<Enabled>*<Checksum><CR><LF>
```

Parameter included in the result:

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

Example:

```
$PAIR421*3D
$PAIR001,421,0*3C
$PAIR421,1*20
```

NOTE

This command is only supported on LC29H (AA) and LC79H (AL).

2.4.39. PAIR432: PAIR_RTCM_SET_OUTPUT_MODE

Sets RTCM output mode.

Type:

Set

Synopsis

```
$PAIR432,<Mode>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Mode>	Numeric	-	RTCM output mode setting. -1 = Disable outputting RTCM 0 = Enable output RTCM3 with message type MSM4 1 = Enable output RTCM3 with message type MSM7

Result:

Returns **$PAIR001** message.

Example:

```
$PAIR432,1*22
```

$PAIR001,432,0*3E

2.4.40. PAIR433: PAIR_RTCM_GET_OUTPUT_MODE

Queries RTCM output mode.

Type:

Get

Synopsis

$PAIR433*<Checksum><CR><LF>

Parameter:

None

Result:

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

Query result message format:

$PAIR433,<Mode>*<Checksum><CR><LF>

Parameter included in the result:

Field	Format	Unit	Description
<Mode>	Numeric	-	RTCM output mode setting. -1 = Outputting RTCM disabled 0 = Outputting RTCM3 with message type MSM4 enabled 1 = Outputting RTCM3 with message type MSM7 enabled

Example:

$PAIR433*3E
$PAIR001,433,0*3F
$PAIR433,-1*0E

2.4.41. PAIR434: PAIR_RTCM_SET_OUTPUT_ANT_PNT

Enables/disables outputting stationary antenna reference point in RTCM format.

Type:

Set

Synopsis

```
$PAIR434,<Enable>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Enable>	Numeric	-	Enable/disable outputting stationary antenna reference point (message type 1005). 0 = Disable 1 = Enable

Result:

Returns **$PAIR001** message.

Example:

```
$PAIR434,1*24
$PAIR001,434,0*38
```

2.4.42. PAIR435: PAIR_RTCM_GET_OUTPUT_ANT_PNT

Queries the setting of outputting stationary antenna reference point in RTCM format.

Type:

Get

Synopsis

```
$PAIR435*<Checksum><CR><LF>
```

Parameter:

None

Result:

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

Query result message format:

```
$PAIR435,<Enable>*<Checksum><CR><LF>
```

Parameter included in the result:

Field	Format	Unit	Description
<Enable>	Numeric	-	Status of outputting stationary antenna reference point (message type 1005). 0 = Disabled 1 = Enabled

Example:

```
$PAIR435*38
$PAIR001,435,0*39
$PAIR435,1*25
```

2.4.43. PAIR436: PAIR_RTCM_SET_OUTPUT_EPHEMERIS

Enables/disables outputting satellite ephemeris in RTCM format.

Type:

Set

Synopsis

```
$PAIR436,<Enable>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Enable>	Numeric	-	Enable/disable outputting satellite ephemeris. 0 = Disable 1 = Enable

Result:

Returns **$PAIR001** message.

Example:

```
$PAIR436,1*26
$PAIR001,436,0*3A
```

2.4.44. PAIR437: PAIR_RTCM_GET_OUTPUT_EPHEMERIS

Queries the status of satellite ephemeris in RTCM format.

Type:

Get

Synopsis

```
$PAIR437*<Checksum><CR><LF>
```

Parameter:

None

Result:

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

Query result message format:

```
$PAIR437,<Enable>*<Checksum><CR><LF>
```

Parameter included in the result:

Field	Format	Unit	Description
<Enable>	Numeric	-	Status of outputting satellite ephemeris. 0 = Disabled 1 = Enabled

Example:

```
$PAIR437*3A
$PAIR001,437,0*3B
$PAIR437,1*27
```

2.4.45. PAIR511: PAIR_NVRAM_SAVE_NAVIGATION_DATA

Saves current navigation data from RTC RAM to NVM.

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 **$PAIR511*3F**, send **$PAIR002*38** to re-power the module. This limitation does not apply to fix rate below 1 Hz.

2.4.46. PAIR513: PAIR_NVRAM_SAVE_SETTING

Saves the current configurations from RTC RAM to NVM.

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 **$PAIR513*3D**, send **$PAIR002*38** to re-power the module. This limitation does not apply to fix rate below 1 Hz.

2.4.47. PAIR650: PAIR_LOW_POWER_ENTRY_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 *documents [2]* and *[3] hardware designs*.

Type:

Set

Synopsis:

```
$PAIR650,<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.

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, only the **$PAIR001** message will be returned.

Example:

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

NOTE

Refer to *documents [2]* and *[3] hardware designs* for details about entering/exiting the Backup mode.

2.4.48. 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 fix 2 = 3D fix only 3 = 2D/3D fix only 4 = Always

Field	Format	Unit	Description
			Default value for LC29H (AA) and LC79H (AL): 2; default value for LC29H (BA, CA, DA, EA) and LC79H (EL): 1.
<PPSPulseWidth>	Numeric	Millisecond	PPS Pulse Width. Range: 1–999. Default value: 100.

Result:

Returns **$PAIR001** message.

Example:

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

NOTE

For LC29H (BA, CA) modules, it is recommended to keep the **<PPSType>** parameter at its default value of 1, as modifying this setting may cause DR functionality abnormalities.

2.4.49. 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	-	Hardware port type. 0 = UART
<PortIndex>	Numeric	-	Hardware port index. 0 = UART1
<Baudrate>	Numeric	bps	Baud rate. 4800 9600 19200

Field	Format	Unit	Description
			38400 57600 115200 230400 460800 921600 3000000 1)

Result:

Returns **$PAIR001** message.

Example:

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

NOTE

1. The module must be rebooted after changing the port baud rate, and the change will take effect after the reboot.
2. Messages may be lost when the baud rate is below 115200 bps at 1 Hz output frequency. And messages may be lost when the baud rate is below 460800 bps at 10 Hz output frequency.
3. 1) LC29H (AA) and LC79H (AL) do not support 3000000 bps.

2.4.50. PAIR865: PAIR_IO_GET_BAUDRATE

Gets the baud rate of the UART interface.

Type:

Get

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<PortType>	Numeric	-	Hardware port type. 0 = UART

Field	Format	Unit	Description
<PortIndex>	Numeric	-	Hardware port index. 0 = UART1

Result:

Returns **$PAIR001** 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. 4800 9600 19200 38400 57600 115200 230400 460800 921600 3000000

Example:

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

2.4.51. PAIR866: PAIR_IO_SET_FLOW_CONTROL

Sets UART flow control.

Type:

Set

Synopsis:

```
$PAIR866,<PortType>,<PortIndex>,<FlowControl>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<PortType>	Numeric	-	Hardware port type. 0 = UART
<PortIndex>	Numeric	-	Hardware port index. 0 = UART1
<FlowControl>	Numeric	-	Flow control setting. 0 = Flow control disabled 1 = Software flow control enabled

Result:

Returns **$PAIR001** message.

Example:

```
$PAIR866,0,0,1*2F
$PAIR001,866,0*33
```

NOTE

Reboot the module after changing the flow control setting, and the change will take effect after the module reboot.

2.4.52. PAIR867: PAIR_IO_GET_FLOW_CONTROL

Gets UART flow control setting.

Type:

Get

Synopsis:

```
$PAIR867,<PortType>,<PortIndex>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<PortType>	Numeric	-	Hardware port type. 0 = UART
<PortIndex>	Numeric	-	Hardware port index. 0 = UART1

Result:

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

Query result message format:

```
$PAIR867,<FlowControl>*<Checksum><CR><LF>
```

Parameter included in the result:

Field	Format	Unit	Description
<FlowControl>	Numeric	-	Flow control setting. 0 = Flow control disabled 1 = Software flow control enabled

Example:

```
$PAIR867,0,0*33
$PAIR001,867,0*32
$PAIR867,0*2F
```

3 RTCM Protocol

The LC29H series and LC79H series modules support the RTCM protocol which is in accordance with *RTCM Standard 10403.3 Differential GNSS (Global Navigation Satellite Systems) Services - Version 3*. This protocol is used to transfer GNSS raw measurement data and is available from https://www.rtcm.org/.

Table 8: Supported RTCM3 Messages

Message Type	Mode [1)]	Message Name
1005	Input/output	Stationary RTK Reference Station ARP.
1006	Input	Stationary RTK Reference Station ARP with Antenna Height
1019	Output	GPS Ephemerides.
1020	Output	GLONASS Ephemerides.
1042	Output	BDS Satellite Ephemeris Data.
1044	Output	QZSS Ephemerides.
1046	Output	Galileo I/NAV Satellite Ephemeris Data.
1074	Input/output	GPS MSM4.
1075	Input	GPS MSM5.
1077	Input/output	GPS MSM7.
1084	Input/output	GLONASS MSM4.
1085	Input	GLONASS MSM5.
1087	Input/output	GLONASS MSM7.
1094	Input/output	Galileo MSM4.
1095	Input	Galileo MSM5.
1097	Input/output	Galileo MSM7.

Message Type	Mode [1)]	Message Name
1114	Input/output	QZSS MSM4.
1115	Input	QZSS MSM5.
1117	Input/output	QZSS MSM7.
1124	Input/output	BDS MSM4.
1125	Input	BDS MSM5.
1127	Input/output	BDS MSM7.
1134	Input	NavIC MSM4
1135	Input	NavIC MSM5
1137	Input	NavIC MSM7

NOTE

1. **$PAIR432** can enable/disable the output of MSM4/MSM7 (1074, 1077, 1084, 1087, 1094, 1097, 1114, 1117, 1124 and 1127) messages if the corresponding constellation is enabled.
2. **$PAIR434** can enable/disable the output of Stationary RTK Reference Station ARP (1005) message.
3. **$PAIR436** can enable/disable the output of ephemeris (1019, 1020, 1042, 1044 and 1046) messages in RTCM format if the corresponding constellation is enabled.
4. [1)] The input mode of RTCM messages is only supported on the LC29H (BA, DA, EA) and LC79H (EL) modules.

4 Appendix A References

Table 9: Related Documents

Document Name
[1] Quectel_LC29H(BA,CA,DA,EA)&&LC79H(EL)_DR&RTK_Application_Note
[2] Quectel_LC79H_Series_Hardware_Design
[3] Quectel_LC29H_Series_Hardware_Design

Table 10: Terms and Abbreviations

Abbreviation	Description
2D	2 Dimensional
3D	3 Dimensional
ACK	Acknowledgement
AIC	Active Interference Cancellation
ARP	Antenna Reference Point
BDS	BeiDou Navigation Satellite System
DGPS	Differential Global Positioning System
DOP	Dilution of Precision
DSP	Digital Signal Processing
ECEF	Earth Centered, Earth Fixed
EGNOS	European Geostationary Navigation Overlay Service
EPO	Extended Prediction Orbit
GAGAN	GPS Aided Geo Augmented Navigation

Abbreviation	Description
Galileo	Galileo Satellite Navigation System (EU)
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
RMS	Root Mean Square
HDOP	Horizontal Dilution of Precision
ID	Identifier
MNL	Middleware Navigation Library
MSAS	Multi-functional Satellite Augmentation System (Japan)
NavIC	Navigation with Indian Constellation
NMEA	NMEA (National Marine Electronics Association) 0183 Interface Standard
NVM	Non-Volatile Memory
NVRAM	Non-Volatile Random Access Memory
PAIR	Proprietary Protocol of Airoha
PDOP	Position Dilution of Precision
PE	Positioning Engine
PPS	Pulse Per Second
QZSS	Quasi-Zenith Satellite System
RF	Radio Frequency
RTC	Real-time Clock
RTK	Real Time Kinematic
SBAS	Satellite-Based Augmentation System
SLAS	Sub-meter Level Augmentation Service
SNR	Signal-to-noise Ratio
SV	Satellites in View

Abbreviation	Description
PVT	Position, Velocity, and Timing
UART	Universal Asynchronous Receiver/Transmitter
UTC	Coordinated Universal Time
VDOP	Vertical Dilution of Precision
WAAS	Wide Area Augmentation System

5 Appendix B GNSS Numbering

Table 11: GNSS Satellites (NMEA) Numbering

GNSS Type	System ID	Satellite ID	Signal ID
GPS	1	1–32 33–51 for SBAS	1 = L1 C/A 8 = L5
GLONASS	2	65–88	1 = L1
Galileo	3	1–36	1 = E5a 7 = E1
BDS	4	1–63	1 = B1I 5 = B2a
QZSS	5	193–199	1 = L1 C/A 8 = L5
NavIC	6	1–14	1 = L5

NOTE

1. QZSS System ID is 1 for LC29H (AA) and LC79H (AL) modules.
2. QZSS Satellite ID numbers range from 1 to 10 for LC29H (BA, CA, DA, EA) and LC79H (EL).

6 Appendix C Special Characters

Table 12: 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.