LC26G&LC26G-T&
LC76G&LC86G Series
GNSS Protocol Specification

GNSS Module Series

Version: 1.4

Date: 2024-03-05

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	**LC26G&LC26G-T&LC76G&LC86G Series GNSS Protocol Specification**
Subtitle	GNSS Module Series
Document Type	GNSS Protocol Specification
Document Status	Released

Revision History

Version	Date	Description
-	2022-05-20	Creation of the document
1.0	2022-07-19	First official release
1.1	2022-12-20	1. Added applicable variants LC76G (PB) and LC86G (AB). 2. Added the table listing the protocols supported by the modules (*Table 2*). 3. Added the Sample Code for NMEA Checksum (*Chapter 2.1*). 4. Added ZDA, GNS, GST, GRS, RLM messages (*Chapters 2.2.7*, *2.2.8*, *2.2.9*, *2.2.10* and *2.2.11*). 5. Added a note about the execution of PAIR messages in the general description paragraph (*Chapter 2.3)*. 6. Updated <Type> field in $PAIR062 and $PAIR063 messages (*Chapters 2.4.13* and *2.4.15*). 7. Added the note in $PAIR050, $PAIR062, $PAIR066, $PAIR080 and $PAIR081 (*Chapters 2.4.9*, *2.4.13*, *2.4.16* and *2.4.24*). 8. Added $PAIR154, $PAIR155, $PAIR158, $PAIR680, $PAIR681, $PAIR690, $PAIR691, $PAIR730, $PAIR731, $PAIR732, $PAIR733, $PAIR900, $PAIR901, $PAIR902, $PAIR903, $PAIR904, $PAIR905, $PAIR906, $PAIR907, $PAIR908, $PAIR909 commands (*Chapters 2.4.30*, *2.4.31*, *2.4.32*, *2.4.52*, *2.4.53*, *2.4.54*, *2.4.55*, *2.4.56*, *2.4.57*, *2.4.58*, *2.4.59*, *2.4.71*, *2.4.72*, *2.4.73*, *2.4.74*, *2.4.75*, *2.4.76*, *2.4.77*, *2.4.78*,

Version	Date	Description
		2.4.79, 2.4.80 and 2.4.81). 9. Added the format and field description of binary data. 10. Added PQTM messages (Chapter 2.3).
1.2	2023-05-18	1. Added notes on GRS message (Chapter 2.2.10). 2. Added a note on $PQTMCFGMSGRATE and a note on parameter default value in Set command (Chapter 2.3.1). 3. Added a table of messages supported by the $PQTMCFGMSGRATE command (Chapter 2.3.1). 4. Added $PQTMCFGANTENNA and $PQTMANTENNASTATUS messages (Chapters 2.3.6 and 2.3.7). 5. Removed the description that a PAIR command may need to be sent once or multiple times until $PAIR001 is returned for LC86G series (Chapter 2.4). 6. Updated the notes of $PAIR003*39 for LC86G series (Chapters 2.4.2 and 2.4.3). 7. Updated the software versions of LC86G (LA) supporting $PAIR158 (Chapter 2.4.32). 8. Added $PAIR159 command (Chapter 2.2.10). 9. Updated the note about $PAIR382 (Chapter 2.4.34). 10. Updated the examples of $PAIR491 and $PAIR511 (Chapters 2.4.48 and 2.4.49).
1.3	2023-12-02	1. Added applicable variants LC26G-T (AA) and LC86G (PA). 2. Added $PQTMPVT, $PQTMCFGGEOFENCE, $PQTMGEOFENCESTATUS, $PQTMCFGPPS and $PQTMPL messages (Chapters 2.3.8, 2.3.9, 2.3.10, 2.3.11, and 2.3.12). 3. Added $PAIR011, $PAIR098, $PAIR099, $PAIR753, $PAIR754, $PAIR761, $PAIR762, $PAIR763, $PAIR764 and $PAIR765 messages (Chapters 2.4.9, 2.4.28, 2.4.29, 2.4.62, 2.4.63, 2.4.64, 2.4.65, 2.4.66, 2.4.67 and 2.4.68). 4. Added the table of modules and software versions supporting B1C band (Chapter 2.4.32). 5. Updated the description of <NavMode> and related notes for $PAIR080 and $PAIR081 messages (Chapters 2.4.24 and 2.4.25). 6. Added a note about $PAIR513 message (Chapter 2.4.49). 7. Updated the description of <Enabled> and notes about $PAIR732, $PAIR733 messages (Chapters 2.4.58 and 2.4.59). 8. Removed $PAIR890 and PAIR891 (Chapter 2.4).
1.4	2024-03-05	1. Updated the field <Res2> to <Interval> and its related description for $PQTMCFGPPS (Chapter 2.3.11). 2. Added the following messages: ● $PQTMLS (Chapter 2.3.13); ● $PQTMJAMMINGSTATUS (Chapter 2.3.14);

Version	Date	Description
		● $PQTMDOP (*Chapter 2.3.15*); ● $PQTMCFGODO (*Chapter 2.3.16*); ● $PQTMRESTODO (*Chapter 2.3.17*); ● $PQTMODO (*Chapter 2.3.18*); ● $PAIR514 (*Chapter 2.4.50*); ● $PAIR751 (*Chapter 2.4.60*). 3. Added note on the position fix rate for using EASY (*Chapter 2.4.46*). 4. Updated the description of <PPSType> and added the related note for $PAIR752 and $PAIR763 (*Chapters 2.4.61* and *2.4.66*). 5. Added a note about disabling LOCUS feature before powering off or restarting the module (*Chapter 2.4.71*).

Contents

Table Index

1 Introduction

Quectel LC26G (AB), LC26G-T (AA), LC76G series and LC86G series GNSS modules support multi-constellations and multi-frequency bands, providing fast and accurate acquisition and making these modules ideal solutions for positioning and navigation in various vertical markets.

Table 1: Applicable Variants and Supported Frequency Bands

Module	Variant	Frequency Band
LC26G	LC26G (AB)	GPS L1 C/A + GLONASS L1 + Galileo E1 + BDS B1I, B1C + QZSS L1 C/A
LC26G-T	LC26G-T (AA)	GPS L1 C/A + GLONASS L1 + Galileo E1 + BDS B1I, B1C + QZSS L1 C/A
LC76G Series	LC76G (AB)	GPS L1 C/A + GLONASS L1 + Galileo E1 + BDS B1I, B1C + QZSS L1 C/A
	LC76G (PA)	GPS L1 C/A + GLONASS L1 + Galileo E1 + BDS B1I, B1C + QZSS L1 C/A
	LC76G (PB)	GPS L1 C/A + GLONASS L1 + Galileo E1 + BDS B1I, B1C + QZSS L1 C/A
LC86G Series	LC86G (AA)	GPS L1 C/A + Galileo E1 + BDS B1I, B1C + QZSS L1 C/A
	LC86G (AB)	GPS L1 C/A + GLONASS L1 + Galileo E1 + QZSS L1 C/A
	LC86G (LA)	GPS L1 C/A + GLONASS L1 + Galileo E1 + BDS B1I, B1C + QZSS L1 C/A
	LC86G (PA)	GPS L1 C/A + GLONASS L1 + Galileo E1 + BDS B1I, B1C + QZSS L1 C/A

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

LC26G (AB), LC26G-T (AA), LC76G series and LC86G series support the following protocols:

Table 2: Supported Protocols

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

NOTE

1. Quectel assumes no responsibility if commands other than the ones listed herein are used.
2. For conciseness purposes, LC26G (AB), LC26G-T (AA), LC76G series and LC86G series modules will hereinafter be referred to collectively as “the module/modules” in parts hereof applicable to all modules, and individually as “LC26G (AB)”, “LC26G-T (AA)”, “LC76G series” and “LC86G series” in parts hereof referring to the differences between them.

2 NMEA Protocol

2.1. Structure of NMEA Protocol Messages

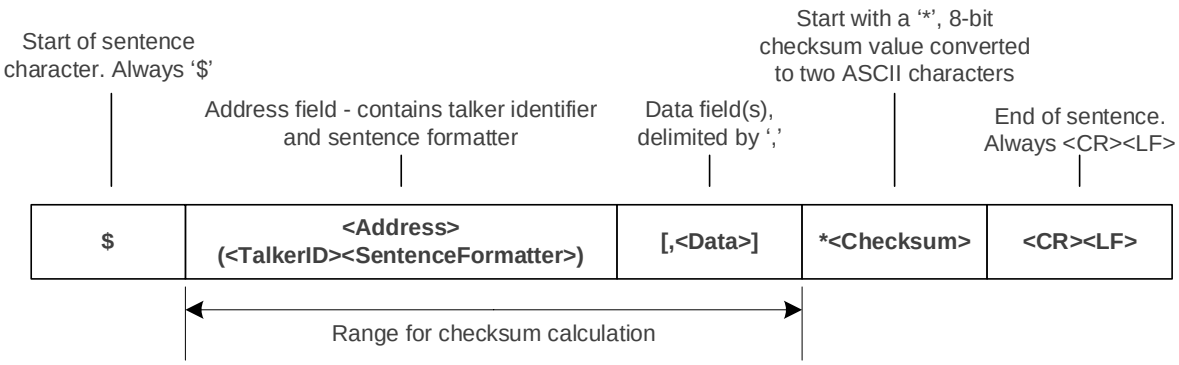

Figure 1: Structure of NMEA Protocol Messages

Table 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 (Sentence Formatter). The talker identifier identifies the data 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.
<Data>	Data fields, delimited by data field delimiter **','**.

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

Table 4: NMEA Talker ID

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

Sample Code for NMEA Checksum:

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

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

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

    return result;
}
```

2.2. Standard Messages

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

2.2.1. RMC

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

Type:

Output

Synopsis:

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

Parameter:

Field	Format	Unit	Example	Description
$	Character	-	$	Each NMEA message starts with **$**.
<TalkerID>	String, 2 characters	-	GN	Talker identifier. See *Table 4: NMEA Talker ID*.
RMC	String, 3 characters	-	RMC	Recommended Minimum Specific GNSS Data.
<UTC>	hhmmss.sss	-	040143.000	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	-	3149.334166	Latitude. dd. Degrees (00–90) mm. Minutes (00–59) mmmmmm: Decimal fraction of minutes Note that this field is empty in case of an invalid value.

Field	Format	Unit	Example	Description
<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	-	11706.941670	Longitude. ddd: Degrees (000–180) mm: Minutes (00–59) mmmmmm: Decimal fraction of minutes. Note that this field is empty in case of an invalid value.
<E/W>	Character	-	E	East-west direction. E = East W = West Note that this field is empty in case of an invalid value.
<SOG>	Numeric	Knot	0.01	Speed over ground. Variable length. Note that this field is empty in case of an invalid value.
<COG>	Numeric	Degree	0.00	Course over ground. Variable length. Maximum value: 359.99. Note that this field is empty in case of an invalid value.
<Date>	ddmmyy	-	010522	Date. dd: Day of month mm: Month yy: Year
<MagVar>	-	-	-	Magnetic variation. Not supported.
<MagVarDir>	-	-	-	Direction of magnetic variation. Not supported.
<ModeInd>	Character	-	D	Mode indicator. A = Autonomous mode. Satellite system used in non-differential mode for position fixing. D = Differential mode. Satellite system used in differential mode for position fixing. Corrections from ground stations or Satellite Based Augmentation System (SBAS). E = Estimated (dead reckoning)

Field	Format	Unit	Example	Description
				mode. N = No fix. Satellite system not used for position fixing, or fix not valid.
<NavStatus>	Character	-	V	Navigational status. Not supported. V = Navigational status not valid, equipment is not providing navigational status indication. Note that this parameter is only available in messages in line with NMEA0183 V4.10 and newer versions.
<Checksum>	Hexadecimal	-	*0E	Checksum.
<CR><LF>	Character	-	-	Carriage return and line feed.

LC76G (AB) Example:

```
$GNRMC,040143.000,A,3149.334166,N,11706.941670,E,0.01,0.00,010522,,,D,V*0E
```

2.2.2. GGA

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

Type:

Output

Synopsis:

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

Parameter:

Field	Format	Unit	Example	Description
$	Character	-	$	Each NMEA message starts with **$**.
<TalkerID>	String, 2 characters	-	GN	Talker identifier. See *Table 4: NMEA Talker ID*.
GGA	String, 3 characters	-	GGA	Global Positioning System Fix Data.
<UTC>	hhmmss.sss	-	040143.000	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
<Lat>	ddmm.mmmmmm	-	3149.334166	Latitude. dd: Degrees (00–90) mm: Minutes (00–59) mmmmmm: Decimal fraction of minutes Note that this field is empty in case of an invalid value.
<N/S>	Character	-	N	North-south direction. N = North S = South Note that this field is empty in case of an invalid value.
<Lon>	dddmm.mmmmmm	-	11706.941670	Longitude. ddd: Degrees (000–180) mm: Minutes (00–59) mmmmmm: Decimal fraction of minutes Note that this field is empty in case of an invalid value.
<E/W>	Character	-	E	East-west direction. E = East W = West Note that this field is empty in case of an invalid value.
<Quality>	Numeric, 1 digit	-	2	GPS quality indicator. 0 = Fix not available or invalid 1 = GPS SPS Mode, fix valid 2 = Differential GPS, SPS Mode, or Satellite Based Augmentation. System (SBAS), fix valid 6 = Estimated (dead reckoning) mode
<NumSatUsed> [1)]	Numeric, 2 digits	-	36	Number of satellites in use.
<HDOP>	Numeric	-	0.48	Horizontal dilution of precision. Note that this field is empty in case of an invalid value.
<Alt>	Numeric	Meter	61.496	Altitude above mean-sea-level (geoid).

Field	Format	Unit	Example	Description
				Note that this field is empty in case of an invalid value.
M	Character	-	M	Unit of <Alt>. "M" = Meter.
<Sep>	Numeric	Meter	-0.335	Geoid separation (the difference between the earth ellipsoid surface and the mean-sea-level (geoid) surface defined by the reference datum used in the position solution). Note that this field is empty in case of an invalid value.
M	Character	-	M	Unit of <Sep>. "M" = Meter.
<DiffAge>	-	-	-	Differential GPS data age. Not supported.
<DiffStation>	-	-	-	Differential reference station ID. Not supported.
<Checksum>	Hexadecimal	-	*58	Checksum.
<CR><LF>	Character	-	-	Carriage return and line feed.

LC76G (AB) Example:

```
$GNGGA,040143.000,3149.334166,N,11706.941670,E,2,36,0.48,61.496,M,-0.335,M,,*58
```

NOTE

1. The NMEA 0183 specification indicates that **GGA** messages are GPS specific. However, when the receiver is configured for multi-constellations, the content of **GGA** messages will be generated from the multi-constellation solution.
2. [1] According to the NMEA 0183 specification, the number of satellites in use is between 00 and 12. However, in the multi-constellation solution, the number of satellites in use may exceed 12.

2.2.3. GSV

GNSS Satellites in View. The GSV sentence provides the number of satellites in view (SV), satellite ID numbers, elevation, azimuth, and SNR value, and 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:

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

Parameter:

Field	Format	Unit	Example	Description
$	Character	-	$	Each NMEA message starts with **$**.
<TalkerID>	String, 2 characters	-	GP	Talker identifier. See *Table 4: NMEA Talker ID*.
GSV	String, 3 characters	-	GSV	GNSS Satellites in view.
<TotalNumSen>	Numeric	-	3	Total number of sentences. Range: 1–9.
<SenNum>	Numeric	-	1	Sentence number. Range: 1–**<TotalNumSen>**.
<TotalNumSat>	Numeric	-	12	Total number of satellites in view.
Start of repeat block. Repeat times: 1–4.				
<SatID>	Numeric	-	195	Satellite ID. See *Table 13: GNSS Satellites (NEMA) Numbering.*
<SatElev>	Numeric	Degree	72	Satellite elevation. Range: 00–90.
<SatAz>	Numeric	Degree	076	Satellite azimuth, with true north as the reference plane. Range: 000–359.
<SatCN0>	Numeric	dB-Hz	42	Satellite C/N_0. Range 00–99. Null when not tracking.
End of repeat block.				
<SignalID>	Numeric	-	1	GNSS signal ID. See *Table 13: GNSS Satellites (NEMA) Numbering*. Note that this parameter is only available in messages in line with NMEA 0183 V4.10 and newer versions.
<Checksum>	Hexadecimal	-	*6D	Checksum.

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

LC76G (AB) Example:

```
$GPGSV,3,1,12,195,72,076,42,01,69,158,45,194,66,111,29,21,61,060,44,1*6D
$GPGSV,3,2,12,07,61,233,42,30,52,284,44,199,51,162,37,08,39,045,42,1*59
$GPGSV,3,3,12,14,29,312,29,196,20,148,36,17,18,258,36,27,07,061,36,1*53
$GLGSV,2,1,05,79,80,068,47,82,62,248,44,81,56,014,38,78,31,137,24,1*7F
$GLGSV,2,2,05,88,07,034,29,1*46
$GAGSV,2,1,06,26,80,095,42,01,69,353,13,21,49,106,26,33,42,207,41,7*72
$GAGSV,2,2,06,13,28,040,34,31,19,313,34,7*72
$GBGSV,4,1,16,46,81,194,38,07,68,349,31,40,61,016,40,30,60,259,43,1*71
$GBGSV,4,2,16,10,59,321,,03,51,192,36,36,41,314,38,02,37,229,32,1*71
$GBGSV,4,3,16,09,31,219,26,08,27,175,31,37,25,146,29,06,23,202,29,1*78
$GBGSV,4,4,16,16,20,199,31,13,17,186,26,39,12,192,29,28,09,048,30,1*7C
```

NOTE

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

2.2.4. GSA

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

Type:

Output

Synopsis:

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

Parameter:

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

Field	Format	Unit	Example	Description
GSA	String, 3 characters	-	GSA	GNSS DOP and Active Satellites.
<Mode>	Character	-	A	Selection of 2D or 3D fix. A = Automatic, allowed to automatically switch to 2D or 3D
<FixMode>	Numeric	-	3	Fix mode. 1 = Fix not available 2 = 2D 3 = 3D
Start of repeat block. Repeat times: 12.				
<SatID>	Numeric	-	195	ID numbers of satellites used in solution. See *Table 13: GNSS Satellites (NEMA) Numbering*. Note that this field is empty in case of an invalid value.
End of repeat block.				
<PDOP>	Numeric	-	0.71	Position dilution of precision. Maximum value: 99.99. Note that this field is empty in case of an invalid value.
<HDOP>	Numeric	-	0.48	Horizontal dilution of precision. Maximum value: 99.99 Note that this field is empty in case of an invalid value.
<VDOP>	Numeric	-	0.52	Vertical dilution of precision. Maximum value: 99.99. Note that this field is empty in case of an invalid value.
<SystemID>	Numeric	-	1	GNSS system ID. See *Table 13: GNSS Satellites (NEMA) Numbering*. Please note that this parameter is only available in messages in line with NMEA 0183 V4.10 or newer versions.
<Checksum>	Hexadecimal	-	*34	Checksum.
<CR><LF>	Character	-	-	Carriage return and line feed.

LC76G (AB) Example:

```
$GNGSA,A,3,195,01,194,21,07,30,199,08,14,17,27,,0.71,0.48,0.52,1*34
$GNGSA,A,3,79,82,81,78,88,,,,,,,0.71,0.48,0.52,2*0D
$GNGSA,A,3,26,21,33,13,31,,,,,,,0.71,0.48,0.52,3*09
$GNGSA,A,3,46,07,40,30,03,36,02,09,08,37,06,16,0.71,0.48,0.52,4*0B
$GNGSA,A,3,13,39,28,,,,,,,,,0.71,0.48,0.52,4*0B
```

NOTE

If less than 12 satellites are used for navigation, the remaining **<SatID>** fields are left empty.

2.2.5. VTG

Course Over Ground & Ground Speed. The actual course and speed relative to the ground.

Type:

Output

Synopsis:

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

Parameter:

Field	Format	Unit	Example	Description
$	Character	-	$	Each NMEA message starts with **$**.
<TalkerID>	String, 2 characters	-	GN	Talker identifier. See *Table 4: NMEA Talker ID*.
VTG	String, 3 characters	-	VTG	Course Over Ground & Ground Speed.
<COGT>	Numeric	Degree	0.00	Course over ground, in true north course direction. Note that this field is empty in case of an invalid value.
T	Character	-	T	Fixed field: true.
<COGM>	Numeric	Degree	-	Course over ground (magnetic). Not supported.
M	Character	-	M	Fixed field: magnetic.
<SOGN>	Numeric	Knot	0.01	Speed over ground in knots. Note that this field is empty in case of an

Field	Format	Unit	Example	Description
				invalid value.
N	Character	-	N	Fixed field: knot.
<SOGK>	Numeric	km/h	0.02	Speed over ground in kilometers per hour. Note that this field is empty in case of an invalid value.
K	Character	-	K	Fixed field: kilometers per hour.
<ModeInd>	Character	-	D	Mode indicator. A = Autonomous mode. Satellite system used in non-differential mode in position fix. D = Differential mode. Satellite system used in differential mode in position fix. Corrections from ground stations or Satellite Based Augmentation System (SBAS). E = Estimated (dead reckoning) mode. N = Data not valid.
<Checksum>	Hexadecimal	-	*25	Checksum.
<CR><LF>	Character	-	-	Carriage return and line feed.

LC76G (AB) Example:

```
$GNVTG,0.00,T,,M,0.01,N,0.02,K,D*25
```

2.2.6. GLL

Geographic Position – Latitude/Longitude. Latitude and longitude of the GNSS receiver position, the time of position fix and status.

Type:

Output

Synopsis:

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

Parameter:

Field	Format	Unit	Example	Description
$	Character	-	$	Each NMEA message starts with **$**.
<TalkerID>	String, 2 characters	-	GN	Talker identifier. See *Table 4: NMEA Talker ID*.
GLL	String, 3 characters	-	GLL	Geographic Position – Latitude/Longitude.
<Lat>	ddmm.mmmmmm	-	3149.334166	Latitude. dd: Degrees (00–90) mm: Minutes (00–59) mmmmmm: Decimal fraction of minutes Note that this field is empty in case of an invalid value.
<N/S>	Character	-	N	North-south direction. N = North S = South Note that this field is empty in case of an invalid value.
<Lon>	dddmm.mmmmmm	-	11706.941670	Longitude. ddd: Degrees (000–180) mm: Minutes (00–59) mmmmmm: Decimal fraction of minutes Note that this field is empty in case of an invalid value.
<E/W>	Character	-	E	East-west direction. E = East W = West Note that this field is empty in case of an invalid value.
<UTC>	hhmmss.sss	-	040143.000	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 = Invalid data
<ModeInd>	Character	-	D	Mode indicator. A = Autonomous mode. Satellite system used in non-differential mode in position fix.

Field	Format	Unit	Example	Description
				D = Differential mode. Satellite system used in differential mode in position fix. Corrections from ground stations or Satellite Based Augmentation System (SBAS). E = Estimated (dead reckoning) mode. N = Data not valid.
<Checksum>	Hexadecimal	-	*46	Checksum
<CR><LF>	Character	-	-	Carriage return and line feed.

LC76G (AB) Example:

```
$GNGLL,3149.334166,N,11706.941670,E,040143.000,A,D*46
```

2.2.7. ZDA

Time and date. UTC, day, month, year and local time zone.

Type:

Output

Synopsis:

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

Parameter:

Field	Format	Unit	Example	Description
$	Character	-	$	Each NMEA message starts with **$**.
<TalkerID>	String, 2 characters	-	GN	Talker identifier. See *Table 4: NMEA Talker ID*.
ZDA	String, 3 characters	-	ZDA	Time and Date. UTC, day, month, year and local time zone.
<UTC>	hhmmss.sss	-	055054.000	Position fix UTC. hh: Hours (00–23) mm: Minutes (00–59) ss: Seconds (00–59) sss: Decimal fraction of seconds
<Day>	Numeric	-	19	Day of month. Range: 01–31.

Field	Format	Unit	Example	Description
<Month>	Numeric	-	09	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	-	*4A	Checksum
<CR><LF>	Character	-	-	Carriage return and line feed.

LC76G (AB) Example:

```
$GNZDA,055054.000,19,09,2022,,*4A
```

2.2.8. GNS

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

Type:

Output

Synopsis:

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

Parameter:

Field	Format	Unit	Example	Description
$	Character	-	$	Each NMEA message starts with **$**.
<TalkerID>	String, 2 characters	-	-	Talker identifier. See *Table 4: NMEA Talker ID*.
GNS	String, 3 characters	-	GNS	GNSS Fix Data.
<UTC>	hhmmss.sss	-	053106.000	Position fix UTC. hh: Hours (00–23) mm: Minutes (00–59) ss: Seconds (00–59) sss: Decimal fraction of seconds

Field	Format	Unit	Example	Description
<Lat>	ddmm.mmmmmm	-	3149.334190	Latitude. dd: Degrees (00–90) mm: Minutes (00–59) mmmmmm: Decimal fraction of minutes Note that this field is empty in case of an invalid value.
<N/S>	Character	-	N	North-south direction. N = North S = South Note that this field is empty in case of an invalid value.
<Lon>	dddmm.mmmmmm	-	11706.948654	Longitude. ddd: Degrees (000–180) mm: Minutes (00–59) mmmmmm: Decimal fraction of minutes Note that this field is empty in case of an invalid value.
<E/W>	Character	-	E	East-west direction. E = East W = West Note that this field is empty in case of an invalid value.
<ModeInd> [1)]	Character	-	DANN	Mode indicator. A = Autonomous mode. Satellite system used in non-differential mode in position fix D = Differential mode. Satellite system used in differential mode for position fixing. Corrections from ground stations or Satellite Based Augmentation System (SBAS) E = Estimated (dead reckoning) mode N = No fix. Satellite system not used for position fixing, or fix not valid
<NumSatUsed>	Numeric	-	16	Total number of satellites in use. Range: 0–99.

Field	Format	Unit	Example	Description
<HDOP>	Numeric	-	0.63	Horizontal dilution of precision. Maximum value: 99.00. Note that this field is empty in case of an invalid value.
<Alt>	Numeric	Meter	51.287	Antenna altitude above the mean-sea-level (geoid). Note that this field is empty in case of an invalid value.
M	Character	-	M	Unit of <Alt>. "M" = Meter.
<Sep>	Numeric	Meter	-0.335	Geoid separation (the difference between the earth ellipsoid surface and the mean-sea-level (geoid) surface defined by the reference datum used in the position solution). Note that this field is empty in case of an invalid value.
M	Character	-	M	Unit of <Sep>. "M" = Meter.
<DiffAge>	-	-	-	Differential GPS data age. Not supported.
<DiffStation>	-	-	-	Differential reference station ID. Not supported.
<NavStatus>	Character	-	V	Navigational status indicator. Always "V" (Navigational status not valid).
<Checksum>	Hexadecimal	-	*05	Checksum.
<CR><LF>	Character	-	-	Carriage return and line feed.

LC76G (AB) Example:

```
$GNGNS,053106.000,3149.334190,N,11706.948654,E,DANN,16,0.63,51.287,M,-0.335,M,,,V*05
```

NOTE

1) **<ModeInd>** is a variable length field. The first character indicates the use of GPS satellites, the second character indicates the use of GLONASS satellites, and the third character indicates the use of Galileo satellites, the fourth character indicates the use of BDS satellites.

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:

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

Parameter:

Field	Format	Unit	Example	Description
$	Character	-	$	Each NMEA message starts with **$**.
<TalkerID>	String, 2 characters	-	GN	Talker identifier. See *Table 4: NMEA Talker ID*.
GST	String, 3 characters	-	GST	GNSS Pseudorange Error Statistics.
<UTC>	hhmmss.sss	-	123624.000	Position fix UTC. hh: Hours (00–23) mm: Minutes (00–59) ss: Seconds (00–59) sss: Decimal fraction of seconds
<RMS_D>	Numeric	Meter	6.3	RMS value of the standard deviation of the range inputs to the navigation process.
<MajorD>	Numeric	Meter	2.5	Standard deviation of semi-major axis of error ellipse.
<MinorD>	Numeric	Meter	2.4	Standard deviation of semi-minor axis of error ellipse.
<Orient>	Numeric	Degree	88.4	Orientation of semi-major axis of error ellipse.
<LatD>	Numeric	Meter	2.4	Standard deviation of latitude error.
<LonD>	Numeric	Meter	2.5	Standard deviation of longitude error.
<AltD>	Numeric	Meter	5.9	Standard deviation of altitude error.

Field	Format	Unit	Example	Description
<Checksum>	Hexadecimal	-	*43	Checksum.
<CR><LF>	Character	-	-	Carriage return and line feed.

LC76G (AB) Example:

```
$GNGST,123624.000,6.3,2.5,2.4,88.4,2.4,2.5,9.2*43
```

2.2.10. GRS

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

Type:

Output

Synopsis:

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

Parameter:

Field	Format	Unit	Example	Description
$	Character	-	$	Each NMEA message starts with **$**.
<TalkerID>	String, 2 characters	-	GN	Talker identifier. See *Table 4: NMEA Talker ID*.
GRS	String, 3 characters	-	GRS	GNSS Range Residuals.
<UTC>	hhmmss.sss	-	125524.000	Position fix UTC. hh: Hours (00–23) mm: Minutes (00–59) ss: Seconds (00–59) sss: Decimal fraction of seconds
<Mode>	Numeric	-	1	Calculation method. 0 = Residuals were used to calculate the position given in the matching **GGA** or **GNS** sentence 1 = Residuals were recomputed after the **GGA** or **GNS** position was computed

Field	Format	Unit	Example	Description
Start of repeat block. Repeat times: 12.				
<Resi>	Numeric	Meter	-0.4	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 13: GNSS Satellites (NEMA) Numbering*. Note that this parameter is only available in messages in line with NMEA 0183 V4.10 or newer versions.
<SignalID>	Numeric	-	1	GNSS signal ID. See *Table 13: GNSS Satellites (NEMA) Numbering*. Note that this parameter is only available in messages in line with NMEA 0183 V4.10 or newer versions.
<Checksum>	Hexadecimal	-	*42	Checksum.
<CR><LF>	Character	-	-	Carriage return and line feed.

LC76G (AB) Example:

```
$GNGRS,125524.000,1,-0.4,-0.7,0.5,-4.6,0.2,1.1,-2.2,-0.6,-1.1,9.2,-2.1,3.1,1,1*42
$GNGRS,125524.000,1,-11.4,,,,,,,,,,,1,1*52
$GNGRS,125524.000,1,19.4,-5.0,11.4,6.3,-118,3.3,-7.5,,,,,,2,1*79
$GNGRS,125524.000,1,-5.6,4.6,21.1,,,,,,,,,,3,7*51
$GNGRS,125524.000,1,-5.8,-10.3,2.4,2.1,-2.3,20.0,-25.1,-9.1,,,,,4,1*7A
```

NOTE

1. The satellite order in a **GRS** sentence should match the order of satellite ID numbers in a **GSA** sentence. If the range residual exceeds ±99.9 meters, the decimal part is truncated, resulting in an integer.
2. The calculation method is: Range Residual = Calculated Range - Measured Range.

2.2.11. RLM

Return Link Message. The receiver will detect the Galileo Search and Rescue (SAR) Return Link Message when the **RLM** function is enabled.

Type:

Output

Synopsis:

```
$<TalkerID>RLM,<BeaconID>,<UTC>,<Meg_Code>,<Para>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Example	Description
$	Character	-	$	Each NMEA message starts with **$**.
<TalkerID>	String, 2 characters	-	GN	Talker identifier. See *Table 4: NMEA Talker ID*.
RLM	String, 3 characters	-	RLM	Return Link Message
<BeaconID>	Hexadecimal	-	9A22BE296 30F010	Beacon ID 15 hex characters (60 bits).
<UTC>	hhmmss.sss	-	055054.000	Position fix UTC. hh: Hours (00–23) mm: Minutes (00–59) ss: Seconds (00–59) sss: Decimal fraction of seconds
<Meg_Code>	Hexadecimal	-	F	Message code, a hex character (4 bits). Identifies the Type of **RLM** Message Service. 0 = Reserved for future RLM services. 1 = Acknowledgement Service RLM 2 = Command Service RLM 3 = Message Service RLM 4–E = Reserved for future RLM services F = Test Service RLM (currently used only by the Galileo Program)
<Para>	Numeric	-	5402	The data parameters provided by RLS. Short message contains 4 hex characters (16 bits) and long message contains 24 hex characters (96 bits).
<Checksum>	Hexadecimal	-	*3B	Checksum

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

LC76G (AB) Example:

```
$GARLM,9A22BE29630F010,125713.000,F,5402*3B
```

2.3. PQTM Messages

This chapter explains the **PQTM** messages (proprietary NMEA messages defined by Quectel) supported by Quectel LC26G (AB), LC26G-T (AA), LC76G series and LC86G series modules.

Table 5: Error Codes

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

2.3.1. PQTMCFGMSGRATE

Sets/gets 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.

Field	Format	Unit	Description
<Rate>	Numeric	-	Message output rate. 0 = Not output. N = Output once per N position fix(es). Range of N, see *Table 6: Supported Messages* for details. Range of N: 0–20.
<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 $PQTMEPE to once per position fix:
$PQTMCFGMSGRATE,W,PQTMEPE,1,2*1D
$PQTMCFGMSGRATE,OK*29

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

Table 6: Supported Messages

Message Name	Message Output Rate Range (N)
$PQTMEPE	1–20
$PQTMANTENNASTATUS	1-20
$PQTMPVT	1–20
$PQTMGEOFENCESTATUS	1–20

$PQTMPL	1–20
$PQTMDOP	1–20
$PQTMODO	1–20
$PQTMJAMMINGSTATUS	1–20
$PQTMLS	1–20

NOTE

1. When configuring the output rate of a **$PQTM** message, use **<MsgVer>** field to specify the message version, otherwise an error will be returned.
2. If the default value is not given for any parameter in a Set command, you can query it with the corresponding Get command, provided that the default setting has not been changed by the Set command. If the default setting has been changed by the Set command, contact Quectel Technical Support (support@quectel.com) to get the default setting, if necessary.

2.3.2. PQTMEPE

Outputs the estimated positioning error.

Type:

Output

Synopsis:

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

Parameter:

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

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

Example:

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

2.3.3. PQTMSAVEPAR

Saves the configurations set via **$PQTM** commands to 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
```

2.3.4. PQTMRESTOREPAR

Restores all parameters set via **$PQTM** to default values.

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
```

2.3.5. PQTMVERNO

Queries the firmware version information.

Type:

Command

Synopsis:

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

Parameter:

None

Result:

- If successful, the module returns:

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

Parameters included in the result:

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

- If failed, the module returns:

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

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

Example:

```
$PQTMVERNO*58
$PQTMVERNO,LC76GABNR02A01S,2022/09/14,11:47:03*3D
```

2.3.6. PQTMCFGANTENNA

Sets/gets the antenna working mode.

Type:

Set/Get

Synopsis:

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

Parameter:

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

Result:

- If successful, the module returns:

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

- If failed, the module returns:

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

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

Example:

```
//Set the antenna working mode to Automatic mode：
$PQTMCFGANTENNA,W,0,0*7E
$PQTMCFGANTENNA,OK*2D

//Get the antenna working mode:
$PQTMCFGANTENNA,R*7B
$PQTMCFGANTENNA,OK,0,0*2D
```

NOTE

1. This command is only supported for LC86G series module.
2. For LC86G series, when connected to a passive antenna, **<Mode>** can only be set to 2.

2.3.7. PQTMANTENNASTATUS

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

Type:

Output

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<MsgVer>	Numeric	-	Message version.

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

Example:

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

NOTE

This message is only supported for LC86G series module.

2.3.8. PQTMPVT

Outputs the PVT (GNSS only) result.

Type:

Output

Synopsis:

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

Parameter:

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

Field	Format	Unit	Description
<TOW>	Numeric	Millisecond	Time of week.
<Date>	YYYYMMDD	-	UTC date. YYYY: Year MM: Month DD: Day of month
<Time>	hhmmss.sss	-	UTC time. hh: Hours. Range: 00–23. mm: Minutes. Range: 00–59. ss: Seconds. Range: 00–59. sss: Decimal fraction of seconds.
<Quality>	Numeric	-	GPS quality indicator. 0 = Fix not available or invalid 1 = GPS SPS Mode, fix valid 2 = Differential GPS, SPS Mode, or Satellite Based Augmentation. System (SBAS), fix valid 6 = Estimated (dead reckoning) mode Note that this field is the same as **<Quality>** in **GGA**.
<FixMode>	Numeric	-	Fix mode. 0 = No fix 1 = Reserved 2 = 2D fix 3 = 3D fix
<NumSatUsed>	Numeric	-	Number of satellites in use.
<LeapS>	Numeric	Second	Leap seconds. Note that this field is empty in case of an invalid value.
<Lat>	Numeric	Degree	Latitude. Note that this field is empty in case of an invalid value.
<Lon>	Numeric	Degree	Longitude. Note that this field is empty in case of an invalid value.
<Alt>	Numeric	Meter	Altitude above mean-sea-level. Note that this field is empty in case of an invalid value.
<Sep>	Numeric	Meter	Geoidal separation (the difference between the WGS84 Earth ellipsoid surface and the mean-sea-level surface). Note that this field is empty in case of an invalid value.
<VelN>	Numeric	m/s	North velocity. Note that this field is empty in case of an invalid value.
<VelE>	Numeric	m/s	East velocity. Note that this field is empty in case of an invalid value.

Field	Format	Unit	Description
<VelD>	Numeric	m/s	Down velocity. Note that this field is empty in case of an invalid value.
<Spd>	Numeric	m/s	Ground speed. Note that this field is empty in case of an invalid value.
<Heading>	Numeric	Degree	Heading. Range: 0.00–360.00. Note that this field is empty in case of an invalid value.
<HDOP>	Numeric	-	Horizontal dilution of precision. Note that the value is 99.99 in case of an invalid value.
<PDOP>	Numeric	-	Position (3D) dilution of precision. Note that the value is 99.99 in case of an invalid value.

Example:

```
//No fix:
$PQTMPVT,1,1012,19800105,235943.012,0,0,00,,,,,,,,,,99.99,99.99*46

//3D fix:
$PQTMPVT,1,459596000,20230505,073938.000,1,3,31,18,31.82176510,117.11534360,99.357,-0.337,-0.004,0.000,-0.003,0.004,96.91,0.51,0.93*74
```

2.3.9. PQTMCFGGEOFENCE

Sets/gets the geofence feature.

Type:

Set/Get

Synopsis:

```
//Set:
$PQTMCFGGEOFENCE,W,<Index>,<Status>[,<Res>,<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.

Field	Format	Unit	Description
<Status>	Numeric	-	Geofence function status. 0 = Disabled 1 = Enabled
<Res>	Numeric	-	Reserved. Always "0".
<Shape>	Numeric	-	Geofence shape. 0 = Circle defined by the center and the radius 1 = Circle defined by the center and a point on the circle 2 = Triangle 3 = Quadrangle (such as square, rectangle, trapezium.)
<Lat0>	Numeric	Degree	Latitude of the first point.
<Lon0>	Numeric	Degree	Longitude of the first point.
<Lat1/Radius>	Numeric	Degree/Meter	If the geofence shape is a circle with a certain radius, this value will be the radius of the circle, otherwise, this value will be the latitude of the second point.
<Lon1>	Numeric	Degree	Longitude of the second point.
<Lat2>	Numeric	Degree	Latitude of the third point.
<Lon2>	Numeric	Degree	Longitude of the third point.
<Lat3>	Numeric	Degree	Latitude of the fourth point.
<Lon3>	Numeric	Degree	Longitude of the fourth point.

Result:

- If successful, the module returns:

```
//Response to Set command:
$PQTMCFGGEOFENCE,OK*<Checksum><CR><LF>
//Response to Get command:
$PQTMCFGGEOFENCE,OK,<Index>,<Status>[,<Res>,<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
```

NOTE

1. Latitude range is [-90, +90], where negative values indicate south latitude. Longitude range is [-180, +180], where negative values indicate west longitude.
2. The geofence shape's points should be set either in a clockwise or counterclockwise sequence.
3. If the geofence shape is a circle, the first point represents the center.
4. If you set latitude and longitude to more than six decimal places when setting geofence coordinates, the values will be rounded to and fixed at six decimal places.

2.3.10. PQTMGEOFENCESTATUS

Outputs the geofence status.

Type:

Output

Synopsis:

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

Parameter:

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

Field	Format	Unit	Description
End of repeat block.			

Example:

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

NOTE

1) The unknown state of **<StateN>** occurs in two cases:

- Geofence function status is disabled;
- The module's position is not fixed.

2.3.11. PQTMCFGPPS

Sets/gets the 1PPS feature configuration.

Type:

Set/Get

Synopsis:

```
//Set:
$PQTMCFGPPS,W,<Index>,<Enable>[,<Duration>,<Mode>,<Res1>,<Interval>]*<Checksum><CR><LF>
//Get:
$PQTMCFGPPS,R,<Index>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Index>	Numeric	-	PPS index. Always "1". 1 = 1PPS
<Enable>	Numeric	-	Enable/disable PPS output. 0 = Disable 1 = Enable
<Duration>	Numeric	Millisecond	Pulse duration. Range: 1–999. Default value: 100.
<Mode>	Numeric	-	Fix mode for PPS output. 1 = Always 2 = 2D fix 3 = 3D fix

Field	Format	Unit	Description
			4 = After the first fix
<Res1>	Numeric	-	Reserved. Always "0".
<Interval> [1)]	Numeric	-	PPS generated interval. 0 = Every time 1 = Odd time 2 = Even time Note that the odd time and even time in **<Interval>** can only be configured successfully under Always and 3D in **<Mode>**.

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>,<Res1>,<Interval>]*<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,1,1,100,1,0,0*72
$PQTMCFGPPS,OK*21

//Get:
$PQTMCFGPPS,R,1*6A
$PQTMCFGPPS,OK,1,1,100,1,0,0*21
```

NOTE

1. When **<Enable>** is set to 0 (disable), the fields after **<Enable>** must be omitted.
2. [1)] **<Interval>** field can only be configured on LC26G-T (AA) module; for other modules, this field is reserved and the value is always “0“.

2.3.12. 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 version.)
<TOW>	Numeric	Millisecond	Time of week. Null if invalid.
<PUL>	Numeric	%	Probability of uncertainty level per epoch.
<Res1>	Numeric	-	Reserved. Always "1".
<Res2>	Numeric	-	Reserved. Always "1".
<PL_PosN>	Numeric	mm	Protection level of north position. Null if invalid.
<PL_PosE>	Numeric	mm	Protection level of east position. Null if invalid.
<PL_PosD>	Numeric	mm	Protection level of down position. Null if invalid.
<PL_VelN>	Numeric	mm/s	Protection level of north velocity. Null if invalid.
<PL_VelE>	Numeric	mm/s	Protection level of east velocity. Null if invalid.
<PL_VelD>	Numeric	mm/s	Protection level of down velocity. Null if invalid.
<Res3>	Numeric	-	Always null.
<Res4>	Numeric	-	Always null.
<PL_Time>	Numeric	Nanosecond	Protection level of time. Null if invalid.

Example:

```
//No fix:
$PQTMPL,1,,5.00,1,1,,,,,,,,*2E

//3D fix:
$PQTMPL,1,206535000,5.00,1,1,4338,3766,15170,244,236,212,,,36*12
```

2.3.13. PQTMLS

Outputs leap second forecast information.

Type:

Output

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<MsgVer>	Numeric	-	Message version. 1 = Version 1 (Always 1 for this version.)
<TOW>	Numeric	Millisecond	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 F = SBAS (not supported)
<WN>	Numeric	-	UTC reference week number.
<LS>	Numeric	s	Current number of leap seconds since the beginning of GPS time (January 6, 1980). It reflects how far ahead GPS time is compared to UTC time. Galileo has the same number of leap seconds as GPS. BDS has 14 fewer leap seconds than GPS. GLONASS follows UTC time, so no leap seconds.

Field	Format	Unit	Description
<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 F = SBAS (not supported) The field value is invalid if **<Flag>** = 0.
<Reserved>	-	-	Reserved. Always null.
<WNLSF>	Numeric	-	Week number of the new leap second. The field value is invalid if **<Flag>** = 0.
<DN>	Numeric	-	The day of the week when the new leap second takes effect. GPS and Galileo: 1 to 7 from Sunday to Saturday; BDS: 0 to 6 from Sunday to Saturday. The field value is invalid if **<Flag>** = 0.
<LSF>	Numeric	s	Leap second count after future leap second changes. The field value is invalid if **<Flag>** = 0.

Example:

```
$PQTMLS,1,191362000,1,2293,18,0,1,,137,7,18*1C
```

NOTE

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

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

2.3.14. PQTMJAMMINGSTATUS

Reports 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 version.)
<Status>	Numeric	-	Jamming detection status. 0 = Unknown. 1 = No jamming, healthy status. 2 = Warning status. 3 = Critical status.

Example:

```
$PQTMJAMMINGSTATUS,1,0*46
```

NOTE

For LC76G (AB), the JAM_IND pin level is low when the jamming statuses detected by the module are warning and critical, and the pin outputs a low-level signal only when jamming is detected. For more information, see *document [3] hardware design*.

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

Example:

```
$PQTMDOP,1,299950000,1.04,0.89,0.53,0.74,0.50,,*7A
```

2.3.16. PQTMCFGODO

Sets/gets the odometer feature configuration.

Type:

Set/Get

Synopsis:

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

Parameter:

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

Result:

- If successful, the module returns:

```
//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:

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

2.3.17. 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.

2.3.18. PQTMODO

Outputs the odometer information.

Type:

Output

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<MsgVer>	Numeric	-	Message version. 1 = Version 1 (Always 1 for this version.)
<Time>	hhmmss.sss	-	UTC time. hh: Hour (00–23) mm: Minute (00–59) ss: Second (00–59) sss: Decimal fraction of seconds

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

Example:

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

NOTE

<Dist> in **$PQTMODO** represents the sum of **<InitDist>** value set in **$PQTMCFGODO** and accumulated mileage. The accumulated distance starts from 0 m and resets to 0 m after a power outage or when cleared with **$PQTMRESETODO**. If **<InitDist>** value in the **$PQTMCFGODO** is modified, the actual **<Dist>** output in **$PQTMODO** reflects the sum of the accumulated distance and the new **<InitDist>** value, as shown below:
<Dist> = Accumulated Distance + **<InitDist>**.

2.4. PAIR Messages

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

2.4.3. PAIR003: PAIR_GNSS_SUBSYS_POWER_OFF

Powers off the GNSS system, including DSP, RF, PE and clock. For LC26G (AB) and LC76G series, the CPU core will enter Standby mode after this command is sent. For LC86G series, the CPU core will enter Sleep mode after this command is sent.

Type:

Command

Synopsis:

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

Parameter:

None

Result:

Returns **$PAIR001** message.

Example:

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

2.4.4. PAIR004: PAIR_GNSS_SUBSYS_HOT_START

Performs a hot start (uses all available data in the NVRAM). Normally a hot start means that the GNSS module has been powered down for less than 2 hours (RTC must be alive) with its ephemeris still valid. Therefore, there is no need to download 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
$PAIR001,004,1*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 almanacs data, and thus needs to download an ephemeris before it can get a valid position.

Type:

Command

Synopsis:

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

Parameter:

None

Result:

Returns **$PAIR001** message.

Example:

```
$PAIR005*3F
$PAIR001,005,1*3F
$PAIR001,005,0*3E
```

2.4.6. PAIR006: PAIR_GNSS_SUBSYS_COLD_START

Performs a cold start, which means that will be erased location information stored in the receiver, including time, position, almanacs and ephemeris data.

Type:

Command

Synopsis:

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

Parameter:

None

Result:

Returns **$PAIR001** message.

Example:

```
$PAIR006*3C
$PAIR001,006,1*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 default settings. Upon a full cold start, the module will erase all stored data from the previous position. Therefore, it needs to search over the full frequency spectrum for all visible satellites before acquiring a valid position.

Type:

Command

Synopsis:

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

Parameter:

None

Result:

Returns **$PAIR001** message.

Example:

```
$PAIR007*3D
$PAIR001,007,1*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 (including roll-over).
<TOW>	Numeric	Second	Time of week.

Example:

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

2.4.9. PAIR011: PAIR_INDICATION_SYSTEM_MESSAGE

Outputs GNSS system notification. The GNSS system automatically outputs this message. Please do not actively send it to the GNSS system.

Type:

Output

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<Type>	Numeric	-	1 = Notification for GNSS system startup

Example:

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

2.4.10. PAIR050: PAIR_COMMON_SET_FIX_RATE

Sets position fix interval.

Type:

Set

Synopsis:

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

Parameter:

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

Result:

Returns **$PAIR001** message.

Example:

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

NOTE

1. If the set frequency exceeds 1 Hz, only **RMC**, **GGA** and **GNS** massages will be output at the set frequency, whereas **VTG**, **GLL**, **ZDA**, **GRS** and **GST** messages will not be output, and **GSA** and **GSV**

messages will be output at 1 Hz.
2. LC26G-T (AA), LC76G (PA), LC76G (PB) and LC86G (PA) modules do not support this command. The position fix rate remains at 1 Hz.

2.4.11. PAIR051: PAIR_COMMON_GET_FIX_RATE

Queries 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
```

2.4.12. PAIR058: PAIR_COMMON_SET_MIN_SNR

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

Type:

Set

Synopsis:

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

Parameter:

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

Result:

Returns **$PAIR001** message.

Example:

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

2.4.13. PAIR059: PAIR_COMMON_GET_MIN_SNR

Gets the minimum SNR threshold 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 of satellites in use. Range: 9–37. Default value: 9.

Example:

```
$PAIR059*36
$PAIR001,059,0*37
$PAIR059,9*23
```

2.4.14. PAIR062: PAIR_COMMON_SET_NMEA_OUTPUT_RATE

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

Type:

Set

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<Type>	Numeric	-	Type of standard NMEA sentence. -1 = Reset the output rates of all types of sentences to default values. 0 = NMEA_SEN_GGA 1 = NMEA_SEN_GLL 2 = NMEA_SEN_GSA 3 = NMEA_SEN_GSV 4 = NMEA_SEN_RMC 5 = NMEA_SEN_VTG 6 = NMEA_SEN_ZDA 7 = NMEA_SEN_GRS 8 = NMEA_SEN_GST 9 = NMEA_SEN_GNS
<OutputRate>	Numeric	-	Message outputting rate setting. 0 = Disabled or not supported N = Output a message once every N position fix(es) Range of N: 0–20.

Field	Format	Unit	Description
			For **GGA**, **GLL**, **GSA**, **GSV**, **RMC** and **VTG**, **<OutputRate>** is 1 by default. For **ZDA**, **GRS**, **GST** and **GNS**, **<OutputRate>** is 0 by default.

Result:

Returns **$PAIR001** message.

Example:

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

NOTE

GGA, **GLL**, **GSA**, **GSV**, **RMC** and **VTG** messages are output by default.

2.4.15. PAIR063: PAIR_COMMON_GET_NMEA_OUTPUT_RATE

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

Type:

Get

Synopsis:

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

Parameter:

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

Field	Format	Unit	Description
			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 outputting rate setting. 0 = Disabled or not supported N = Output message once every N position fix(es) Range: 1–20. Default value: 1.

Example:

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

2.4.16. PAIR066: PAIR_COMMON_SET_GNSS_SEARCH_MODE

Sets the GNSS search mode. The setting is valid when the NVRAM data are valid. The module reboots when it receives this command.

Type:

Set

Synopsis:

```
$PAIR066,<GPS_Enabled>,<GLONASS_Enabled>,<Galileo_Enabled>,<BDS_Enabled>,<QZSS_Enabled>,<Res>*<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
<Res>	Numeric	-	Reserved. Always "0".

Result:

Returns **$PAIR001** message.

Example:

```
//Search for GPS + GLONASS + Galileo + BDS satellites:
$PAIR066,1,1,1,1,0,0*3A
$PAIR001,066,0*3B
```

NOTE

1. QZSS is always enabled by default.
2. GNSS search modes supported by LC26G (AB), LC26G-T (AA), LC76G series and LC86G (LA, PA):
   - GPS only
   - GPS + QZSS
   - GPS + GLONASS
   - GPS + GLONASS+ QZSS
   - GPS + Galileo
   - GPS + Galileo + QZSS

- GPS + BDS
- GPS + BDS+ QZSS
- GPS + GLONASS + Galileo + BDS
- GPS + GLONASS + Galileo + BDS+ QZSS

3. LC86G (AA) is configured as GPS + Galileo + BDS using **$PAIR066,1,1,1,1,0,0*3A**, due to the internal mechanism of the software. However, the GLONASS satellite information will not appear in the output log and GLONASS satellites will not be available in the search channel. Similarly, configure GPS + Galileo + BDS + QZSS using **$PAIR066,1,1,1,1,1,0*3B**.
   GNSS search modes supported by LC86G (AA):
   - GPS only
   - GPS + QZSS
   - GPS + Galileo
   - GPS + Galileo + QZSS
   - GPS + BDS
   - GPS + BDS+ QZSS
   - GPS + Galileo + BDS
   - GPS + Galileo + BDS + QZSS
4. LC86G (AB) is configured as GPS + GLONASS + Galileo using **$PAIR066,1,1,1,1,0,0*3A**, due to the internal mechanism of the software. However, the BDS satellite information will not appear in the output log and BDS satellites will not be available in the search channel. Similarly, configure GPS + GLONASS + Galileo + QZSS using **$PAIR066,1,1,1,1,1,0*3B**.
   GNSS search modes supported by LC86G (AB):
   - GPS only
   - GPS + QZSS
   - GPS + GLONASS
   - GPS + GLONASS+ QZSS
   - GPS + Galileo
   - GPS + Galileo + QZSS
   - GPS + GLONASS + Galileo
   - GPS + GLONASS + Galileo + QZSS

2.4.17. PAIR067: PAIR_COMMON_GET_GNSS_SEARCH_MODE

Queries 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>,<Res>*,<Checksum><CR><LF>
```

Parameters included in the result:

Field	Format	Unit	Description
<GPS_Enabled>	Numeric	-	0 = Disabled (Do not search for GPS satellites) 1 = Search for GPS satellites
<GLONASS_Enabled>	Numeric	-	0 = Disabled (Do not search for GLONASS satellites) 1 = Search for GLONASS satellites
<Galileo_Enabled>	Numeric	-	0 = Disabled (Do not search for Galileo satellites) 1 = Search for Galileo satellites
<BDS_Enabled>	Numeric	-	0 = Disabled (Do not search for BDS satellites) 1 = Search for BDS satellites
<QZSS_Enabled>	Numeric	-	0 = Disabled (Do not search for QZSS satellites) 1 or other non-zero values = Search for QZSS satellites
<Res>	Numeric	-	Reserved. Always "0".

Example:

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

NOTE

1. For LC86G (AA), when you query GPS + GLONASS + BDS constellations, the module will output **$PAIR067,1,1,1,1,0,0*3B**; similarly, when you query GPS + Galileo + BDS + QZSS, it will output **$PAIR067,1,1,1,1,1,0*3A**.
2. For LC86G (AB), when you query GPS + GLONASS + Galileo constellations, it will output **$PAIR067,1,1,1,1,0,0*3B**; similarly, when you query GPS + GLONASS + Galileo + QZSS, it will output **$PAIR067,1,1,1,1,1,0*3A**.

2.4.18. PAIR070: PAIR_COMMON_SET_STATIC_THRESHOLD

Sets the static navigation speed threshold. If the actual speed is below the threshold, the output position remains unchanged and the output speed is equal to 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	Static navigation speed threshold. Range: 0–20. Default value: 0.

Result:

Returns **$PAIR001** message.

Example:

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

2.4.19. PAIR071: PAIR_COMMON_GET_STATIC_THRESHOLD

Gets the static navigation speed threshold.

Type:

Get

Synopsis:

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

Parameter:

None

Result:

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

Query result message format:

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

Parameter included in the result:

Field	Format	Unit	Description
<SpeedThreshold>	Numeric	m/s	Static navigation speed threshold. Range: 0–20. Default value: 0.

Example:

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

2.4.20. PAIR072: PAIR_COMMON_SET_ELEV_MASK

Sets satellite elevation mask.

Type:

Set

Synopsis:

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

Parameter:

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

Result:

Returns **$PAIR001** message.

Example:

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

NOTE

Satellites below the elevation mask cannot be used for positioning.

2.4.21. PAIR073: PAIR_COMMON_GET_ELEV_MASK

Gets satellite elevation mask.

Type:

Get

Synopsis

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

Parameter:

None

Result:

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

Query result message format:

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

Parameter included in the result:

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

Example:

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

2.4.22. PAIR074: PAIR_COMMON_SET_AIC_ENABLE

Enables/disables the active interference cancellation (AIC) function. For details about AIC function, see *documents [1]*, *[2]*, *[3]* and *[4] 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
```

2.4.23. PAIR075: PAIR_COMMON_GET_AIC_STATUS

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

Type:

Get

Synopsis

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

Parameter:

None

Result:

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

Query result message format:

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

Parameter included in the result:

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

Example:

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

2.4.24. PAIR080: PAIR_COMMON_SET_NAVIGATION_MODE

Sets the navigation mode.

Type:

Set

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<NavMode>	Numeric	-	Navigation mode. 0 = Normal mode. For general purposes. 1 = Fitness mode. Used for running and walking purposes, making low-speed movement (< 5 m/s) more impactful on position calculation. 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. Used for stationary applications where a zero-dynamic is assumed. 5 = Drone mode. Used for drone applications with equivalent dynamic range and vertical acceleration at different flight phases (for example, hovering and cruising). 6 = Reserved. 7 = Swimming mode. Used for swimming activities to smooth the

Field	Format	Unit	Description
			trajectory and improve the accuracy of distance calculation.

Result:

Returns **$PAIR001** message.

Example:

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

NOTE

1. LC26G-T (AA) module only supports Normal mode.
2. Each mode has its altitude limitation. Please choose the appropriate mode based on *Table 7: Modes and Altitude Limitations*. If your test scenario exceeds the limitation, the position calculation will be incorrect.
3. When the height is above 80000 meters in Balloon mode, NMEA messages, RTCM messages, debug log, and binary data will not be output. The output of enabled messages resumes when the height is reduced below 80000 meters. Except for Balloon mode, other modes can be divided into three levels according to height. See *Table 8: Relationship Between Height and Output State (Excluding Balloon Mode)* for details.
4. For more information about Stationary mode, see *Table 9: Stationary Mode Details*.

Table 7: Modes and Altitude Limitations

Mode	Altitude Limitation
Normal mode	10000 m
Fitness mode	10000 m
Balloon mode	80000 m
Stationary mode	10000 m
Drone mode	10000 m
Swimming mode	10000 m

Table 8: Relationship Between Height and Output State (Excluding Balloon Mode)

Height	Output State
< 10000 m	Normal position fixing.
10000–50000 m	Positioning status and result cannot be guaranteed.
> 50000 m	NMEA messages, RTCM messages, debug log, and binary data will not be output.

Table 9: Stationary Mode Details

Important Item	Value	Comment
Height limit value	10000 m	1. When all the conditions are met, the longitude and latitude of the module will be fixed: ● Horizontal speed < 5 m/s; ● Driving distance < 50 m; ● Vertical speed < 15 m/s; ● Height change < 250 m. 2. When one of the following conditions is met, the longitude and latitude of the module will be continuously updated: ● Horizontal speed ≥ 5 m/s; ● Vertical speed ≥ 15 m/s; ● Driving distance ≥ 50 m; ● Height change ≥ 250 m.
Speed limit value	490 m/s	
Horizontal speed limit value	5 m/s	
Vertical speed limit value	15 m/s	
Maximum driving distance with constant latitude and longitude under horizontal low speed operation	50 m	
Maximum distance with constant latitude and longitude under vertical low speed operation (i.e., height change)	250 m	

2.4.25. PAIR081: PAIR_COMMON_GET_NAVIGATION_MODE

Queries navigation mode.

Type:

Get

Synopsis:

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

Parameter:

None

Result:

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

Query result message format:

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

Parameter included in the result:

Field	Format	Unit	Description
<NavMode>	Numeric	-	Navigation mode. 0 = Normal mode. For general purposes. 1 = Fitness mode. Used for running and walking purposes, making low-speed movement (< 5 m/s) more impactful on position calculation. 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. Used for stationary applications where a zero-dynamic is assumed. 5 = Drone mode. Used for drone applications with equivalent dynamic range and vertical acceleration at different flight phases (for example, hovering and cruising). 6 = Reserved. 7 = Swimming mode. Used for swimming activities to smooth the trajectory and improve the accuracy of distance calculation.

Example:

```
$PAIR081*33
$PAIR001,081,0*32
$PAIR081,0*2F
```

2.4.26. PAIR086: PAIR_COMMON_SET_DEBUGLOG_OUTPUT

Enables/disables debug log output in binary format.

Type:

Set

Synopsis

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

Parameter:

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

Result:

Returns **$PAIR001** message.

Example:

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

2.4.27. PAIR087: PAIR_COMMON_GET_DEBUGLOG_OUTPUT

Queries the debug log output setting.

Type:

Get

Synopsis

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

Parameter:

None

Result:

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

Query result message format:

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

Parameter included in the result:

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

Example:

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

2.4.28. PAIR098: PAIR_COMMON_SET_NMEA_POS_DECIMAL_PRECISION

Sets the decimal places of NMEA messages.

Type:

Set

Synopsis

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

Parameter:

Field	Format	Unit	Description
<Mode>	Numeric	-	Mode. 0 = Latitude and longitude in 4 digits, altitude in 1 digit. 1 = Latitude and longitude in 5 digits, altitude in 2 digits. 2 = Latitude and longitude in 6 digits, altitude in 3 digits. 3 = Latitude and longitude in 7 digits, altitude in 3 digits.

Result:

Returns **$PAIR001** message.

Example:

```
$PAIR098,0*27
$PAIR001,098,0*3A
```

2.4.29. PAIR099: PAIR_COMMON_GET_NMEA_POS_DECIMAL_PRECISION

Gets the decimal places of NMEA messages.

Type:

Get

Synopsis

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

Parameter:

None

Result:

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

Query result message format:

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

Parameter included in the result:

Field	Format	Unit	Description
<Mode>	Numeric	-	Mode. 0 = Latitude and longitude in 4 digits, altitude in 1 digit. 1 = Latitude and longitude in 5 digits, altitude in 2 digits. 2 = Latitude and longitude in 6 digits, altitude in 3 digits. 3 = Latitude and longitude in 7 digits, altitude in 3 digits.

Example:

```
$PAIR099*3A
$PAIR001,099,0*3B
$PAIR099,0*26
```

2.4.30. PAIR154: PAIR_COMMON_SET_RLM_OUTPUT_ENABLE

Enables/disables outputting of **RLM** message at 1 Hz.

Type:

Set

Synopsis

```
$PAIR154,<Enable>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Enable>	Numeric	-	Enable/disable outputting **RLM** message. 0 = Disable 1 = Enable

Result:

Returns **$PAIR001** message and enable/disable **RLM** message outputting accordingly.

Example:

```
$PAIR154,1*27
$PAIR001,154,0*3B
```

2.4.31. PAIR155: PAIR_COMMON_GET_RLM_OUTPUT_STATUS

Queries **RLM** message output setting.

Type:

Get

Synopsis

```
$PAIR155*<Checksum><CR><LF>
```

Parameter:

None

Result:

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

Query result message format:

```
$PAIR155,<Enable>*<Checksum><CR><LF>
```

Parameter included in the result:

Field	Format	Unit	Description
<Enable>	Numeric	-	**RLM** message output setting. 0 = Disabled 1 = Enabled

Example:

```
$PAIR155*3B
$PAIR001,155,0*3A
$PAIR155,1*26
```

2.4.32. PAIR158: PAIR_COMMON_SET_B1C_ENABLE

Enables/disables tracking of BDS B1C satellite band.

Type:

Set

Synopsis

```
$PAIR158,<Enable>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Enable>	Numeric	-	Enable/disable tracking of BDS B1C band. 0 = Disable 1 = Enable

Result:

Returns **$PAIR001** message.

Example:

```
$PAIR158,1*2B
$PAIR001,158,0*37
```

2.4.33. PAIR159: PAIR_COMMON_GET_B1C_STATUS

Queries tracking of BDS B1C band status.

Type:

Get

Synopsis

```
$PAIR159*<Checksum><CR><LF>
```

Parameter:

None

Result:

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

Query result message format:

```
$PAIR159,<Enable>*<Checksum><CR><LF>
```

Parameter included in the result:

Field	Format	Unit	Description
<Enable>	Numeric	-	BDS B1C band status. 0 = Disabled 1 = Enabled

Example:

```
$PAIR159*37
$PAIR001,159,0*36
$PAIR159,0*2B
```

2.4.34. PAIR382: PAIR_TEST_LOCK_SYSTEM_SLEEP

Enables/disables the locking of Sleep mode. The CPU core will not enter Sleep mode automatically after the command has been sent to the module.

Type:

Set

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<Enabled>	Numeric	-	Enable or disable Sleep mode locking. 0 = Disable 1 = Enable

Result:

Returns **$PAIR001** message.

Example:

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

NOTE

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

2.4.35. PAIR391: PAIR_TEST_JAMMING_DETECT

Enables/disables jamming detection feature. The jamming status messages will be transmitted over the port once jamming detection feature is enabled.

Type:

Set

Synopsis:

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

Parameter:

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

Result:

Returns **$PAIR001** message and enable **$PAIRSPF** message outputting periodically. For details about **$PAIRSPF** message, see *Chapter 2.4.81 PAIRSPF*.

Example:

```
//Unknown status:
$PAIR391,1*2C
$PAIR001,391,0*30
$PAIRSPF,0*53
```

```
//Good status:
$PAIR391,1*2C
$PAIR001,391,0*30
$PAIRSPF,1*52

//Warning status:
$PAIR391,1*2C
$PAIR001,391,0*30
$PAIRSPF,2*51

//Critical status:
$PAIR391,1*2C
$PAIR001,391,0*30
$PAIRSPF,3*50
```

NOTE

Once the jamming detection feature is enabled, the module can detect an external jamming interference signal.

- If there is no jamming, **$PAIRSPF,1*52** 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) If the jamming detection feature has been enabled and the module is still in the acquisition stage, the jamming status will be 1 and it will be changed to 2 if jamming is detected. During this process, the module keeps attempting to acquire a valid position; if the anti-jamming repair fails, the jamming status message will be changed to 3.
  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.

2.4.36. 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, and MSAS)

Result:

Returns a **$PAIR001** message.

Example:

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

2.4.37. 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, and MSAS)

Example:

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

2.4.38. PAIR410: PAIR_SBAS_ENABLE

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

Type:

Set

Synopsis

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

Parameter:

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

Result:

Returns **$PAIR001** message.

Example:

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

NOTE

1. When the navigation mode is Fitness or Swimming mode (see **$PAIR080**), SBAS feature is not supported.
2. SBAS is not supported in low power modes GLP, FLP and ALP mode 1.

2.4.39. 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
```

2.4.40. 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 outputting RTCM3 with message type MSM4 format 1 = Enable outputting RTCM3 with message type MSM7 format

Result:

Returns **$PAIR001** message.

Example:

```
$PAIR432,1*22
$PAIR001,432,0*3E
```

2.4.41. 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 = Disable outputting RTCM 0 = Enable outputting RTCM 10403.3 with message type MSM4 1 = Enable outputting RTCM 10403.3 with message type MSM7

Example:

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

2.4.42. PAIR434: PAIR_RTCM_SET_OUTPUT_ANT_PNT

Enables/disables outputting stationary antenna reference in RTCM format.

Type:

Set

Synopsis

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

Parameter:

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

Result:

Returns **$PAIR001** message.

Example:

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

2.4.43. PAIR435: PAIR_RTCM_GET_OUTPUT_ANT_PNT

Queries the setting of stationary antenna reference point.

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	-	Stationary antenna reference point (message type 1005). 0 = Disable 1 = Enable

Example:

```
$PAIR435*38
$PAIR001,435,0*39
$PAIR435,0*24
```

2.4.44. PAIR436: PAIR_RTCM_SET_OUTPUT_EPHEMERIS

Enables/disables outputting satellite ephemeris information 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
```

2.4.45. PAIR437: PAIR_RTCM_GET_OUTPUT_EPHEMERIS

Queries the status of outputting satellite ephemeris.

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	-	Enable/disable getting satellite ephemeris. 0 = Disable 1 = Enable

Example:

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

2.4.46. PAIR490: PAIR_EASY_ENABLE

Enables/disables the EASY function.

Type:

Set

Synopsis:

```
$PAIR490,<Enabled>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Enabled>	Numeric	-	EASY function setting. 0 = Disable 1 = Enable

Result:

Returns **$PAIR001** message.

Example:

```
$PAIR490,1*2A
$PAIR001,490,0*36
```

NOTE

EASY is supported only when the position fix rate is 1 Hz.

2.4.47. PAIR491: PAIR_EASY_GET_STATUS

Queries the status of the EASY function.

Type:

Get

Synopsis:

```
$PAIR491*<Checksum><CR><LF>
```

Parameter:

None

Result:

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

Query result message format:

```
$PAIR491,<Enabled>,<Status>*<Checksum><CR><LF>
```

Parameters included in the result:

Field	Format	Unit	Description
<Enabled>	Numeric	-	EASY function setting. 0 = Disabled 1 = Enabled
<Status>	Numeric	-	EASY data extension status. 0 = Not finished 1 = 1-day extension finished 2 = 2-day extension finished 3 = 3-day extension finished If **<Enabled>** is set to 0, **<Status>** will not be displayed in the result.

Example:

```
//If <Enabled> is set to enable:
$PAIR491*36
$PAIR001,491,0*37
$PAIR491,1,0*37

//If <Enabled> is set to disable:
$PAIR491*36
```

```
$PAIR001,491,0*37
$PAIR491,0*2A
```

2.4.48. PAIR511: PAIR_NVRAM_SAVE_NAVIGATION_DATA

Saves current navigation data from RTC RAM into flash.

Type:

Command

Synopsis

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

Parameter:

None

Result:

Returns **$PAIR001** message.

Example:

```
//If fix rate is 1 Hz.
$PAIR511*3F
$PAIR001,511,1*3F
$PAIR001,511,0*3E

//If fix rate exceeds 1 Hz.
$PAIR382,1*2E
$PAIR001,382,0*32
$PAIR003*39
$PAIR001,003,1*39
$PAIR001,003,0*38
$PAIR511*3F
$PAIR001,511,1*3F
$PAIR001,511,0*3E
$PAIR002*38
$PAIR001,002,1*38
$PAIR001,002,0*39
```

NOTE

1. If RTC cannot be powered by the hardware after module power supply is cut off, this command must be sent every time the parameters are modified.
2. If fix rate exceeds 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 rates below 1 Hz.

2.4.49. PAIR513: PAIR_NVRAM_SAVE_SETTING

Saves the current configurations from RTC RAM into flash.

Type:

Command

Synopsis:

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

Parameter:

None

Result:

Returns **$PAIR001** message.

Example:

```
//If fix rate is 1 Hz.
$PAIR513*3D
$PAIR001,513,0*3C

//If fix rate exceeds 1 Hz.
$PAIR382,1*2E
$PAIR001,382,0*32
$PAIR003*39
$PAIR001,003,1*39
$PAIR001,003,0*38
$PAIR513*3D
$PAIR001,513,0*3C
$PAIR002*38
$PAIR001,002,1*38
$PAIR001,002,0*39
```

NOTE

1. If RTC cannot be powered by the hardware after the module power supply is cut off, this command must be sent every time the parameters are modified.
2. If position fix rate exceeds 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 rates below 1 Hz.
3. This command does not support saving configurations set by PPS-related **PAIR** messages.

2.4.50. PAIR514: PAIR_NVRAM_RESTORE_DEFAULT_SETTING

Clears the current configurations and restores the default settings.

Type:

Command

Synopsis:

```
$PAIR514*<Checksum><CR><LF>
```

Parameter:

None

Result:

Returns **$PAIR001** message.

Example:

```
$PAIR382,1*2E
$PAIR001,382,0*32
$PAIR003*39
$PAIR001,003,1*39
$PAIR001,003,0*38
$PAIR514*3A
$PAIR001,514,0*3B
$PAIR002*38
$PAIR001,002,1*38
$PAIR001,002,0*39
```

NOTE

1. This function does not support running time restore when the GNSS system is powered on.

2. To send the command successfully, power off the GNSS system with **$PAIR382,1*2E** and **$PAIR003*39** in sequence before sending this command. After sending **$PAIR514*3A**, send **$PAIR002*38** to re-power the module.

2.4.51. PAIR650: PAIR_LOW_POWER_ENTRY_RTC_MODE

Powers off the GNSS system, except the clock. This command sets the CPU to Backup mode, in which it cannot receive any commands. For details about Backup mode, see *documents [1]*, *[3]* and *[4] hardware designs*.

Type:

Set

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<Second>	Numeric	Second	Duration of Backup mode. Range: 0 and 10–62208000 (2 years); 0 means entering the Backup mode without any timer.

Result:

- If successful, the module will be set to Backup mode and be prevented from receiving any commands.
- If failed, the **$PAIR001** message will be returned.

Example:

```
$PAIR650,1*24
$PAIR001,650,4*3C
```

NOTE

LC26G-T (AA) does not support this command as the module does not support the Backup mode.

2.4.52. PAIR680: PAIR_GLP_ENABLE

Enables/disables GPS Low Power (GLP) mode, which utilizes the adjustment of the duty cycle concept to ensure good performance and low power consumption performance in different signal conditions.

Type:

Set

Synopsis

```
$PAIR680,<Enabled>*<Checksum><CR><LF>
```

Parameter:

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

Result:

Returns **$PAIR001** message.

Example:

```
$PAIR680,1*29
$PAIR001,680,0*35
```

NOTE

1. Requirements for entering GLP mode:
   1) Fix rate: 1 Hz.
   2) Satellite constellation configuration: GPS only.
   3) Navigation mode: fitness mode.
2. When the GLP mode is enabled, some of the features will be disabled automatically, such as SBAS ALP, FLP, and the periodic power saving mode.
3. LC26G-T (AA) does not support this command as the module does not support the GLP mode.

2.4.53. PAIR681: PAIR_GLP_GET_STATUS

Queries GPS Low Power (GLP) mode setting.

Type:

Get

Synopsis

```
$PAIR681*<Checksum><CR><LF>
```

Parameter:

None

Result:

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

Query result message format:

```
$PAIR681,<Enabled>*<Checksum><CR><LF>
```

Parameter included in the result:

Field	Format	Unit	Description
<Enabled>	Numeric	-	GLP mode setting. 0 = Disabled 1 = Enabled

Example:

```
$PAIR681*35
$PAIR001,681,0*34
$PAIR681,1*28
```

NOTE

LC26G-T (AA) does not support this command as the module does not support the GLP mode.

2.4.54. PAIR690: PAIR_PERIODIC_SET_MODE

Sets Periodic Power Saving mode configurations. There are two stages in periodic power saving mode (Run stage and Sleep stage), and they will change periodically according to the setting. In Run stage, the GNSS module measures and calculates the position. In Sleep stage, the GNSS module enters power saving modes.

Type:

Set

Synopsis:

```
$PAIR690,<Mode>,<FirstRun>,<FirstSleep>,<SecondRun>,<SecondSleep>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Mode>	Numeric	-	State of Periodic Power Saving mode. 0 = Disabled 1 = Smart periodic mode enabled 2 = Strict periodic mode enabled
<FirstRun>	Numeric	Second	Running time. Range: 3–518400.
<FirstSleep>	Numeric	Second	Sleeping time. Range: 3–518400.
<SecondRun>	Numeric	Second	Second running time. Range: 0 or 3–518400.
<SecondSleep>	Numeric	Second	Second sleeping time. Range: 0 or 3–518400.

Result:

Returns a **$PAIR001** message.

Example:

```
$PAIR690,1,21,39,48,72*28
$PAIR001,690,0*34
```

NOTE

1. **<FirstRun>:** Interval in seconds after exiting Sleep mode and getting a new position fix.
2. **<FirstSleep>:** Duration of Sleep mode after getting a fix (or attempting to get a fix).
3. **<SecondRun>:** GNSS module will use “second running time” instead of “first running time” setting when there is no signal. The second running time can be “0” only when the second sleeping time is “0”.

4. **<SecondSleep>:** GNSS module will use “second sleeping time” instead of “first sleeping time” setting when there is no signal. The second sleeping time can be “0” only when the second running time is “0”.
5. LC26G-T (AA) does not support this command as the module does not support the Periodic Power Saving mode.

2.4.55. PAIR691: PAIR_PERIODIC_GET_MODE

Queries Periodic Power Saving mode configurations.

Type:

Get

Synopsis:

```
$PAIR691*<Checksum><CR><LF>
```

Parameter:

None

Result:

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

Query result message format:

```
$PAIR691,<Mode>,<FirstRun>,<FirstSleep>,<SecondRun>,<SecondSleep>*<Checksum><CR><LF>
```

Parameters included in the result:

Field	Format	Unit	Description
<Mode>	Numeric	-	State of Periodic Power Saving mode. 0 = Disabled 1 = Smart periodic mode enabled 2 = Strict periodic mode enabled
<FirstRun>	Numeric	Second	Running time. Range: 3–518400.
<FirstSleep>	Numeric	Second	Sleeping time. Range: 3–518400.
<SecondRun>	Numeric	Second	Second running time. Range: 0 or 3–518400.
<SecondSleep>	Numeric	Second	Second sleeping time. Range: 0 or 3–518400.

Example:

```
$PAIR691*34
$PAIR001,691,1*34
$PAIR001,691,0*35
$PAIR691,0,3,12,18,72*14
```

NOTE

LC26G-T (AA) does not support this command as the module does not support the Periodic Power Saving mode.

2.4.56. PAIR730: PAIR_FLP_ENABLE

Enables/disables Fitness Low Power (FLP) mode. When the FLP mode is enabled, some of the features will be automatically disabled, such as SBAS, low power periodic mode, GLP mode, and ALP mode.

Type:

Set

Synopsis

```
$PAIR730,<Enabled>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Enabled>	Numeric	-	Enable/disable FLP mode setting. 0 = Disable 1 = Enable

Result:

Returns **$PAIR001** message.

Example:

```
$PAIR730,1*23
$PAIR001,730,0*3F
```

NOTE

1. Requirements for entering FLP mode:

1) Fix rate: 1 Hz.
2) Navigation mode: fitness mode.
3) Supported satellite constellation configuration:
   - GPS + GLONASS + Galileo + BDS (+ QZSS)
   - GPS + GLONASS (+ QZSS)
   - GPS+ BDS (+ QZSS)

2. LC26G-T (AA) does not support this command as the module does not support the FLP mode.

2.4.57. PAIR731: PAIR_FLP_GET_STATUS

Queries FLP mode setting.

Type:

Get

Synopsis

```
$PAIR731*<Checksum><CR><LF>
```

Parameter:

None

Result:

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

Query result message format:

```
$PAIR731,<Enabled>*<Checksum><CR><LF>
```

Parameter included in the result:

Field	Format	Unit	Description
<Enabled>	Numeric	-	FLP mode setting. 0 = Disabled 1 = Enabled

Example:

```
$PAIR731*3F
$PAIR001,731,0*3E
$PAIR731,1*22
```

NOTE

1. Requirements for querying FLP mode:
   1) Fix rate: 1 Hz.
   2) Navigation mode: fitness mode.
   3) Supported satellite constellation configuration:
      - GPS + GLONASS + Galileo + BDS (+ QZSS)
      - GPS + GLONASS (+ QZSS)
      - GPS+ BDS (+ QZSS)
2. LC26G-T (AA) does not support this command as the module does not support the FLP mode.

2.4.58. PAIR732: PAIR_ALP_ENABLE

Enables/disables adaptive low power (ALP) mode, which ensures good performance and low power consumption in different environments.

Type:

Set

Synopsis

```
$PAIR732,<Enabled>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Enabled>	Numeric	-	Enable/disable ALP mode. 0 = Disable 1 = Enable ALP mode 1 2 = Enable ALP mode 2

Result:

Returns **$PAIR001** message.

Example:

```
$PAIR732,1*21
$PAIR001,732,0*3D
```

NOTE

1. Requirements for entering ALP mode:
   1) Fix rate: 1 Hz.
   2) Navigation mode: Normal mode.
2. ALP mode consists of **ALP mode 1** and **ALP mode 2**:
   - **ALP mode 1** maintains lower power consumption while the performance is reduced compared to previous versions.
   - **ALP mode 2** achieves relatively good performance while keeping the power consumption low.
3. Enabling the ALP mode automatically disables some features, such as low power periodic mode, FLP mode, and GLP mode, and reduces the number of tracked satellites.
4. LC26G-T (AA) does not support this command as the module does not support the ALP mode.

2.4.59. PAIR733: PAIR_ALP_GET_STATUS

Queries adaptive low-power (ALP) mode setting.

Type:

Get

Synopsis

```
$PAIR733*<Checksum><CR><LF>
```

Parameter:

None

Result:

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

Query result message format:

```
$PAIR733,<Enabled>*<Checksum><CR><LF>
```

Parameter included in the result:

Field	Format	Unit	Description
<Enabled>	Numeric	-	ALP mode setting. 0 = Disabled 1 = ALP mode 1 enabled 2 = ALP mode 2 enabled

Example:

```
$PAIR733*3D
$PAIR001,733,0*3C
$PAIR733,1*20
```

NOTE

1. Requirements for querying ALP mode:
   1) Fix rate: 1 Hz.
   2) Navigation mode: Normal mode.
2. LC26G-T (AA) does not support this command as the module does not support the ALP mode.

2.4.60. PAIR751: PAIR_PPS_SYNC_NMEA

Synchronizes PPS pulse with NMEA message timestamp.

Type:

Set

Synopsis

```
$PAIR751,<Enabled>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Enabled>	Numeric	-	PPS synchronizes NMEA setting. 0 = Disabled 1 = Enabled

Result:

Returns **$PAIR001** message.

Example:

```
$PAIR751,1*3D
$PAIR001,751,0*38
```

NOTE

After the function is enabled, the time between PPS and NMEA pulses will be fixed at 350 ms.

2.4.61. PAIR752: PAIR_PPS_SET_CONFIG_CMD

Sets PPS configurations.

Type:

Set

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<PPSType> [1)]	Numeric	-	PPS pulse type. 0 = Disable 1 = After the first fix 2 = 3D fix only 3 = 2D/3D fix only 4 = Always 5 = Odd time only (under 3D fix) 6 = Even time only (under 3D fix) 7 = Odd time only (Always) 8 = Even time only (Always)
<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

[1)] Values 5, 6, 7, and 8 in the **<PPSType>** field are only supported on LC26G-T (AA).

2.4.62. PAIR753: PAIR_PPS_SET_TIMING_PRODUCT

Sets timing mode. The timing mode enhances the PPS output timing accuracy.

Type:

Set

Synopsis:

```
$PAIR753,<TimingMode>[,<Par1>,…,<ParN>]*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<TimingMode>	Numeric	-	Timing mode. 0 = Timing mode disabled 2 = Survey-in mode 3 = Position-hold mode 4 = PPS pulse delay
<Par1>,…,<ParN>	Numeric	-	This field varies with the message type.

2.4.62.1. If <TimingMode> = 0

Disables timing mode.

Type:

Set

Synopsis:

```
$PAIR753,0*<Checksum><CR><LF>
```

Result:

Returns **$PAIR001** message.

Example:

```
$PAIR753,0*27
$PAIR001,753,0*3A
```

2.4.62.2. If <TimingMode> = 2

Configures the Survey-in mode. It allows the system to calculate the average position by setting the measurement time/measurement accuracy.

Type:

Set

Synopsis:

```
$PAIR753,2,<SVIN_Time>,<SVIN_Threshold>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<SVIN_Time>	Numeric	Second	Survey-in time. Range: 1–2592000.
<SVIN_Threshold>	Numeric	Meter	Survey-in distance standard deviation threshold. Range: 0.0–100.0.

Result:

Returns **$PAIR001** message.

Example:

```
$PAIR753,2,100,10*15
$PAIR001,753,0*3A
```

NOTE

This command is only supported on LC26G-T (AA) module.

2.4.62.3. If <TimingMode> = 3

Configures the Position-hold mode. The position holding function is realized by inputting a known position.

Type:

Set

Synopsis:

```
$PAIR753,3,<Coordinate>,<ECEF-X/Latitude>,<ECEF-Y/Longitude>,<ECEF-Z/Altitude>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Coordinate>	Numeric	-	Coordinate selection. 0 = ECEF 1 = LLH
<ECEF-X/Latitude>	Numeric	Meter Degree	If **<Coordinate>** = 0, this value means ECEF-X. If **<Coordinate>** = 1, this value means LLH-Latitude.
<ECEF-Y/Longitude>	Numeric	Meter Degree	If **<Coordinate>** = 0, this value means ECEF-Y. If **<Coordinate>** = 1, this value means LLH-Longitude.
<ECEF-Z/Altitude>	Numeric	Meter Degree	If **<Coordinate>** = 0, this value means ECEF-Z. If **<Coordinate>** = 1, this value means LLH-Altitude.

Result:

Returns **$PAIR001** message.

Example:

```
$PAIR753,3,0,-3027171.52,4925534.63,2684798.13*20
$PAIR001,753,0*3A
```

NOTE

This command is only supported on LC26G-T (AA) module.

2.4.62.4. If <TimingMode> = 4

Sets the PPS pulse delay

Type:

Set

Synopsis:

```
$PAIR753,4,<PPSDelay>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<PPSDelay>	Numeric	Nanosecond	PPS pulse delay. Range: -500000000–500000000.

Result:

Returns **$PAIR001** message.

Example:

```
$PAIR753,4,100*3E
$PAIR001,753,0*3A
```

NOTE

This command is only supported on LC26G-T (AA) module.

2.4.63. PAIR754: PAIR_PPS_GET_TIMING_INFO

Gets timing mode information.

Type:

Get

Synopsis:

```
$PAIR754*<Checksum><CR><LF>
```

Result:

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

Query result message format:

```
$PAIR754,<TimingStatus>,<Position-holdStatus>,<SVIN_ElapsedTime>,<ECEF-X>,<ECEF-Y>,<ECEF-Z>,<PositionSTD>,<PPSDelay>*<Checksum><CR><LF>
```

Parameters included in the result:

Field	Format	Unit	Description
<TimingStatus>	Numeric	-	Timing status. 0 = Timing mode disabled

Field	Format	Unit	Description
			1 = Survey-in mode 2 = Position-hold mode
<Position-holdStatus>	Numeric	-	Position-hold mode status. 0 = Disabled 1 = Enabled
<SVIN_ElapsedTime>	Numeric	Second	Survey-in elapsed time.
<ECEF-X>	Numeric	Meter	Position-hold ECEF coordinate in X direction.
<ECEF-Y>	Numeric	Meter	Position-hold ECEF coordinate in Y direction.
<ECEF-Z>	Numeric	Meter	Position-hold ECEF coordinate in Z direction.
<PositionSTD>	Numeric	Meter	Current position standard deviation in Survey-in mode.
<PPSDelay>	Numeric	Nanosecond	PPS pulse delay.

Example:

```
$PAIR754*3C
$PAIR001,754,1*3C
$PAIR001,754,0*3D
$PAIR754,1,0,21,-2472432.15,4828369.57,3343691.40,1.88,100*24
```

NOTE

This command is only supported on LC26G-T (AA) module.

2.4.64. PAIR761: PAIR_PPS_SET_REF_TIMEBASE

Sets PPS reference time base.

Type:

Set

Synopsis:

```
$PAIR761,<RefTime>,<TimeBase>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<RefTime>	Numeric	-	Reference time base. 0 = GNSS 1 = UTC
<TimeBase>	Numeric	-	If **<Ref_Time>** = 0: 0 = GPS 2 = Galileo 3 = BDS If **<Ref_Time>** = 1: 0 = UTC (USNO) derived from GPS time 2 = UTC (European laboratories) derived from Galileo time 3 = UTC (NTSC) derived from BDS time

Result:

Returns **$PAIR001** message.

Example:

```
$PAIR761,1,0*3B
$PAIR001,761,0*3B
```

NOTE

This command is only supported on LC26G-T (AA) module.

2.4.65. PAIR762: PAIR_PPS_GET_REF_TIMEBASE

Gets PPS reference time base.

Type:

Get

Synopsis:

```
$PAIR762*<Checksum><CR><LF>
```

Parameter:

None

Result:

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

Query result message format:

```
$PAIR762,<Status>,<RefTimebase>,<IntDelta>,<FraDelta>*<Checksum><CR><LF>
```

Parameters included in the result:

Field	Format	Unit	Description
<Status>	Numeric	-	UTC-correction data status. 0 = Not ready 1 = Ready
<RefTimebase>	Numeric	-	Reference time base. 0 = GPS 2 = Galileo 3 = BDS 5 = UTC (USNO) derived from GPS time 7 = UTC (European laboratories) derived from Galileo time 8 = UTC (NTSC) derived from BDS time
<IntDelta>	Numeric	Second	UTC leap second difference integer.
<FraDelta>	Numeric	Nanosecond	UTC leap second difference fraction.

Example:

```
$PAIR762*39
$PAIR001,762,0*38
$PAIR762,1,0,18,1.061039*23
```

NOTE

This command is only supported on LC26G-T (AA) module.

2.4.66. PAIR763: PAIR_PPS_GET_CONFIG

Gets the PPS configurations set.

Type:

Get

Synopsis

```
$PAIR763*<Checksum><CR><LF>
```

Parameter:

None

Result:

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

Query result message format:

```
$PAIR763,<Res1>,<Res2>,<Res3>,<PPSType>,<PPSPulseWidth>*<Checksum><CR><LF>
```

Parameters included in the result:

Field	Format	Unit	Description
<Res1>	Numeric	-	Reserved. Always "0".
<Res2>	Numeric	-	Reserved. Always "0".
<Res3>	Numeric	-	Reserved. Always "0".
<PPSType> [1)]	Numeric	-	PPS pulse type. 0 = Disable 1 = After the first fix 2 = 3D fix only 3 = 2D/3D fix only 4 = Always 5 = Odd time only (under 3D fix) 6 = Even time only (under 3D fix) 7 = Odd time only (Always) 8 = Even time only (Always)
<PPSPulseWidth>	Numeric	Millisecond	PPS pulse width. Range: 1–999. Default value: 100.

Example:

```
$PAIR763*38
$PAIR001,763,0*39
$PAIR763,0,0,0,2,100*27
```

NOTE

[1] Values 5, 6, 7, and 8 in the **<PPSType>** field are only supported on LC26G-T (AA).

2.4.67. PAIR764: PAIR_PPS_TRAIM_SET_CONFIG

Sets TRAIM configuration.

Type:

Set

Synopsis:

```
$PAIR764,<Enable>,<AlarmThr>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Enable>	Numeric	-	Enable/disable TRAIM. 0 = Disable 1 = Enable
<AlarmThr>	Numeric	Nanosecond	Time error threshold. Range: 3–200.

Result:

Returns **$PAIR001** message.

Example:

```
$PAIR764,1,100*3F
$PAIR001,764,0*3E
```

NOTE

This command is only supported on LC26G-T (AA) module.

2.4.68. PAIR765: PAIR_PPS_TRAIM_GET_STATUS

Gets TRAIM current status.

Type:

Get

Synopsis

```
$PAIR765*<Checksum><CR><LF>
```

Parameter:

None

Result:

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

Query result message format:

```
$PAIR765,<Enable>,<Warning>,<AlarmThr>,<TimeError>*<Checksum><CR><LF>
```

Parameters included in the result:

Field	Format	Unit	Description
<Enable>	Numeric	-	TRAIM current status. 0 = UTC correction data not ready 1 = UTC correction data ready
<Warning>	Numeric	-	Warning status. 0 = Under alarm 1 = Over alarm 2 = Unknown
<AlarmThr>	Numeric	Nanosecond	Time error threshold.
<TimeError>	Numeric	Nanosecond	Current time error.

Example:

```
$PAIR765*3E
$PAIR001,765,0*3F
$PAIR765,1,0,100,28.0*1A
```

NOTE

This command is only supported on LC26G-T (AA) module.

2.4.69. PAIR864: PAIR_IO_SET_BAUDRATE

Sets the UART port baud rate. For the new configuration to take effect, reboot the module after changing the port baud rate.

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 = UART0
<Baudrate>	Numeric	bps	Baud rate. 9600 19200 38400 57600 115200 230400 460800 921600

Result:

Returns **$PAIR001** message.

Example:

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

NOTE

Default baud rate is recommended. If the actual baud rate is lower than the 115200 bps, messages may be lost.

2.4.70. PAIR865: PAIR_IO_GET_BAUDRATE

Queries the baud rate of UART port.

Type:

Get

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<PortType>	Numeric	-	Hardware port type. 0 = UART
<PortIndex>	Numeric	-	Hardware port index. 0 = UART0

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. 9600 19200 38400 57600 115200 230400 460800 921600

Example:

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

2.4.71. PAIR900: PAIR_LOCUS_ENABLE

Enables/disables the LOCUS feature to save location data. LOCUS is the feature of integrated logging solution that enables automatic storage of log data. The LOCUS data storage capacity is 128 KB. For specific guidelines on how to save LOCUS data, see *Chapter 2.4.73 PAIR902: PAIR_LOCUS_SET_MODE*.

Type:

Set

Synopsis

```
$PAIR900,<Enable>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Enable>	Numeric	-	Enable/disable saving fix data by LOCUS. 0 = Disable 1 = Enable

Result:

Returns **$PAIR001** message.

Example:

```
$PAIR900,1*2E
$PAIR001,900,0*32
```

NOTE

1. Saved location data: UTC time, fix status, longitude, latitude, altitude, ground speed, heading degree, horizontal dilution of precision, and number of satellites used.
2. Same configuration cannot be set after the first successful execution of **$PAIR900**.
3. Disable LOCUS feature before powering off or restarting the module, otherwise the data saved before powering off or restarting will be lost.

2.4.72. PAIR901: PAIR_LOCUS_GET_STATUS

Queries whether LOCUS saves data.

Type:

Get

Synopsis

```
$PAIR901*<Checksum><CR><LF>
```

Parameter:

None

Result:

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

Query result message format:

```
$PAIR901,<Enable>*<Checksum><CR><LF>
```

Parameter included in the result:

Field	Format	Unit	Description
<Enable>	Numeric	-	LOCUS setting. 0 = Disabled 1 = Enabled

Example:

```
$PAIR901*32
$PAIR001,901,0*33
$PAIR901,0*2E
```

2.4.73. PAIR902: PAIR_LOCUS_SET_MODE

Sets LOCUS saving mode.

Type:

Set

Synopsis

```
$PAIR902,<Mode>,<Check_3D_Fix>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Mode>	Decimal	-	Save Mode: Bit 0 = 1. Normal. Record each fix data.

Field	Format	Unit	Description
			Bit 1 = 1. Time-triggered save mode. Record once after the time threshold is met. For details, see *Chapter 2.4.75 PAIR904: PAIR_LOCUS_SET_THRESHOLD*. Bit 2 = 1. Speed-triggered save mode. Record once after the speed threshold is met. For the value of N, see *Chapter 2.4.75 PAIR904: PAIR_LOCUS_SET_THRESHOLD*. Bit 3= 1. Distance-triggered save mode. Record once after the distance threshold is met. For details, see *Chapter 2.4.75 PAIR904: PAIR_LOCUS_SET_THRESHOLD*. Bit 4 = 1. Before entering sleep mode. Record before entering sleep. Bit 5 = 1. User control. Record after user send $PAIR907*34. For details, see *Chapter 2.4.78 PAIR907: PAIR_LOCUS_LOG_NOW*. Note that when the value of each bit is 0, it means that the corresponding data is not recorded.
<Check_3D_Fix>	Numeric	-	Whether it is necessary to check 3D fix or not. 0 = Do not check. 1 = It is necessary to check. If you set this type as 1, system will save the position with 3D fixed.

Result:

Returns **$PAIR001** message.

Example:

```
//Set mode as out of time and out of speed mode. It is necessary to check 3D fix.
$PAIR902,6,1*36
$PAIR001,902,0*30
```

NOTE

LOCUS saving must be disabled before sending this command.

2.4.74. PAIR903: PAIR_LOCUS_GET_MODE

Queries LOCUS saving mode.

Type:

Get

Synopsis

```
$PAIR903*<Checksum><CR><LF>
```

Parameter:

None

Result:

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

Query result message format:

```
$PAIR903,<Mode>,<Check_3D_Fix>*<Checksum><CR><LF>
```

Parameters included in the result:

Field	Format	Unit	Description
<Mode>	Decimal	-	Save Mode: Bit 0 = 1. Normal. Record each fix data. Bit 1 = 1. Time-triggered save mode. Record once after the time threshold is met. For details, see *Chapter 2.4.75 PAIR904: PAIR_LOCUS_SET_THRESHOLD*. Bit 2 = 1. Speed-triggered save mode. Record once after the speed threshold is met. For the value of N, see *Chapter 2.4.75 PAIR904: PAIR_LOCUS_SET_THRESHOLD*. Bit 3= 1. Distance-triggered save mode. Record once after the distance threshold is met. For details, see *Chapter 2.4.75 PAIR904: PAIR_LOCUS_SET_THRESHOLD*. Bit 4 = 1. Before entering sleep mode. Record before entering sleep. Bit 5 = 1. User control. Record after user send $PAIR907*34. For details, see *Chapter 2.4.78 PAIR907: PAIR_LOCUS_LOG_NOW*. Note that when the value of each bit is 0, it means that the corresponding data is not recorded.
<Check_3D_Fix>	Numeric		Whether it is necessary to check 3D fix or not. 0 = Do not check 1 = It is necessary to check. If you set this type as 1, system will save the position with 3D fixed

Example:

```
$PAIR903*30
$PAIR001,903,0*31
```

$PAIR903,6,1*37

2.4.75. PAIR904: PAIR_LOCUS_SET_THRESHOLD

Sets LOCUS mode threshold.

Type:

Set

Synopsis

$PAIR904,<Mode>,<Threshold>*<Checksum><CR><LF>

Parameter:

Field	Format	Unit	Description
<Mode>	Numeric	-	Saving Mode. 0 = Time-triggered save mode 1 = Speed-triggered save mode 2 = Distance-triggered save mode
<Threshold>	Numeric	Second m/s Meter	When **<mode>** = 0, it indicates time threshold. Range: 1-43200. Unit: second. When **<mode>** = 1, it indicates speed threshold. Range: 1-100. Unit: m/s. When **<mode>** = 2, it indicates distance threshold. Range: 1-50000. Unit: meter.

Result:

Returns **$PAIR001** message.

Example:

$PAIR904,1,5*33
$PAIR001,904,0*36

NOTE

1. Make sure that LOCUS saving is disabled before executing **$PAIR902**.
2. Make sure to set the save mode as time-triggered save mode, speed-triggered save mode, or distance-triggered save mode before executing **$PAIR904** to set mode threshold.

2.4.76. PAIR905: PAIR_LOCUS_GET_THRESHOLD

Queries LOCUS mode threshold.

Type:

Get

Synopsis

```
$PAIR905,<mode>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Mode>	Numeric	-	Saving Mode. 0 = Time-triggered 1 = Speed-triggered 2 = Distance-triggered

Result:

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

Query result message format:

```
$PAIR905,<Threshold>*<Checksum><CR><LF>
```

Parameter included in the result:

Field	Format	Unit	Description
<Threshold>	Numeric	Second m/s Meter	When **<mode>** = 0, it indicates time threshold. Range: 1–43200. Unit: second. When **<mode>** = 1, it indicates speed threshold. Range: 1–100. Unit: m/s. When **<mode>** = 2, it indicates distance threshold. Range: 1-50000. Unit: meter.

Example:

```
$PAIR905,0*2A
$PAIR001,905,0*37
$PAIR905,15*1E
```

2.4.77. PAIR906: PAIR_LOCUS_CLEAR

Clears stored LOCUS data.

Type:

Command

Synopsis:

```
$PAIR906,<Type>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Type>	Numeric	-	Clean type. 0 = Clear recorded data and restore to default setting 1 = Clear recorded data only 2 = Clear user setting and restore to default setting

Result:

Returns **$PAIR001** message.

Example:

```
$PAIR906,0*29
$PAIR001,906,0*34
```

2.4.78. PAIR907: PAIR_LOCUS_LOG_NOW

Saves current fix data in flash.

Type:

Command

Synopsis:

```
$PAIR907*<Checksum><CR><LF>
```

Parameter:

None

Result:

Returns **$PAIR001** message.

Example:

```
$PAIR907*34
$PAIR001,907,0*35
```

NOTE

1. Make sure to set the value of user control (bit 5) as 1 in **<mode>** field of **$PAIR902** before executing this command.
2. Saved fix data: UTC time, fix status, longitude, latitude, altitude, ground speed, heading degree, horizontal dilution of precision, number of satellites used.

2.4.79. PAIR908: PAIR_LOCUS_GET_DATA

Queries LOCUS data. There are two output formats to choose from: standard NMEA message format and PAIR message format.

Type:

Command

Synopsis:

```
$PAIR908,<Type>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Type>	Numeric	-	Response type. 0 = Response in NMEA format 1 = Response in PAIR format

Result:

Enables outputting LOCUS data periodically and returns **$PAIR001** message.

Query result message format:

1. Start reading LOCUS data:

```
$PAIR908,0*<Checksum><CR><LF>
```

2. Read LOCUS data:

```
$PAIR908,1,<Record_Num>,<Record_Size>*<Checksum><CR><LF>
```

Parameters included in the result:

Field	Format	Unit	Description
<Record_Num>	Numeric	-	Total number of recorded LOCUS data.
<Record_Size>	Numeric	-	Size of each LOCUS data recorded.

3. Output LOCUS data according to response type of **<Type>** field:

- If **<Type>** = 0, the message outputs are LOGGA + LORMC. Please refer to **GGA** for LOGGA format and **RMC** for LORMC format.
- If **<Type>** = 1, output message format is as follows:

```
$PAIR908,2,<UTC>,<Fix_Type>,<Lat>,<Lon>,<Height>,<Speed>,<Heading>,<HDOP>,<SatNo>*<Checksum><CR><LF>
```

Parameters included in the result:

Field	Format	Unit	Description
<UTC>	Hexadecimal	-	UTC Second. 4 bytes in LOCUS data.
<Fix_Type>	Hexadecimal	-	Fix quality, refer to Quality Indicator in GGA. 1 byte in LOCUS data.
<Lat>	Hexadecimal	Degree	User latitude in WGS84. 4 bytes in LOCUS data.
<Lon>	Hexadecimal	Degree	User longitude in WGS84. 4 bytes in LOCUS data.
<Height>	Hexadecimal	Meter	User altitude above mean sea level. 2 bytes in LOCUS data.
<Speed>	Hexadecimal	m/s	User ground speed (2-D). 2 bytes in LOCUS data.
<Heading>	Hexadecimal	Degree	User heading of motion. 2 bytes in LOCUS data.
<HDOP>	Hexadecimal	-	Horizontal (2-D) dilution of precision. 2 bytes in LOCUS data.
<SatNo>	Hexadecimal	-	Number of satellites used in navigation solution. 1 byte in LOCUS data.

4. LOCUS data read ends.

```
$PAIR908,3*<Checksum><CR><LF>
```

Example:

```
//If <Type> = 0, LOCUS example:
```

```
$PAIR908,0*27
//LOCUS output:
$PAIR908,0*27
$PAIR908,1,5699,23*24
$LOGGA,033632.000,3148.8100,N,11707.0463,E,1,4,2.45,3.0,M,,M,,*53
$LORMC,033632.000,A,3148.8100,N,11707.0463,E,0.00,0.00,160122,,,A,V*07
…
$LOGGA,033645.000,3148.8108,N,11707.0451,E,1,8,1.14,13.0,M,,M,,*60
$LORMC,033645.000,A,3148.8108,N,11707.0451,E,0.00,0.00,160122,,,A,V*0E
$LOGGA,033649.000,3148.8133,N,11707.0461,E,1,9,1.00,15.0,M,,M,,*65
$LORMC,033649.000,A,3148.8133,N,11707.0461,E,0.00,0.00,160122,,,A,V*09
$PAIR908,3*24
$PAIR001,908,0*3A

//If <Type> = 1, LOCUS example:
$PAIR908,1*26
//LOCUS output:
$PAIR908,0*27
$PAIR908,1,5699,23*24
$PAIR908,2,61E392B9,01,12F65A6D,45CEB007,0021,0000,0000,005F,09*77
$PAIR908,2,61E392BA,01,12F65A67,45CEAFDD,0021,0000,0000,005D,0A*74
…
$PAIR908,2,61E3AC35,02,12FBDCD1,45E9D8ED,0006,0000,8960,003C,1D*0F
$PAIR908,2,61E3AC44,02,12FBDCD1,45E9D8ED,0006,0000,8960,0038,1D*72
$PAIR908,2,61E3AC4E,02,12FBE812,45E9D677,0007,0007,8BBC,0036,1E*75
$PAIR908,3*24
$PAIR001,908,0*3A
```

NOTE

1. Make sure to execute **$PAIR900,1*2F** before executing **$PAIR908**.
2. When **<Type>** = 0, the decimal fraction of **<Lat>** and **<Lon>** is 4 digits and that of **<Hight>** is 1 digit.

2.4.80. PAIR909: PAIR_LOCUS_GET_RECORD_NUM

Queries the total number of recorded LOCUS data.

Type:

Command

Synopsis

```
$PAIR909*<Checksum><CR><LF>
```

Parameter:

None

Result:

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

Query result message format:

```
$PAIR909,<Record_Num>*<Checksum><CR><LF>
```

Parameter included in the result:

Field	Format	Unit	Description
<Record_Num>	Numeric	-	Total number of recorded LOCUS data.

Example:

```
$PAIR909*3A
$PAIR001,909,0*3B
$PAIR909,5243*16
```

2.4.81. PAIRSPF

Outputs jamming status when jamming detection function is enabled.

Type:

Output

Synopsis:

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

Parameter:

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

Field	Format	Unit	Description
			2 = Warning status 3 = Critical status

Example:

```
//Unknown status:
$PAIRSPF,0*53

//No jamming, good status:
$PAIRSPF,1*52

//Warning status:
$PAIRSPF,2*51

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

3 RTCM Protocol

The modules support 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 10: Supported RTCM3 Messages

Message Type	Mode	Message Name
1005	Output	Stationary RTK Reference Station ARP.
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	Output	GPS MSM4.
1077	Output	GPS MSM7.
1084	Output	GLONASS MSM4.
1087	Output	GLONASS MSM7.
1094	Output	Galileo MSM4.
1097	Output	Galileo MSM7.
1114	Output	QZSS MSM4.
1117	Output	QZSS MSM7.
1124	Output	BDS MSM4.
1127	Output	BDS MSM7.

NOTE

1. **$PAIR432** can enable/disable MSM4/MSM7 (1074, 1077, 1084, 1087, 1094, 1097,1114, 1117, 1124, 1127) messages if the corresponding constellation is enabled.
2. **$PAIR434** can enable/disable Stationary RTK Reference Station ARP (1005) message.
3. **$PAIR436** can enable/disable ephemeris (1019, 1020, 1042, 1044, 1046) messages if the corresponding constellation is enabled.

4 Appendix A References

Table 11: Related Documents

Document Name
[1] Quectel_LC26G(AB)_Hardware Design
[2] Quectel_LC26G-T(AA)_Hardware Design
[3] Quectel_LC76G_Series_Hardware_Design
[4] Quectel_LC86G_Series_Hardware_Design

Table 12: Terms and Abbreviations

Abbreviation	Description
2D	2 Dimension
3D	3 Dimension
ACK	Acknowledgement
ALP	Adaptive Low Power
AIC	Active Interference Cancellation
BDS	BDS Navigation Satellite System
C/N_0	Carrier-to-Noise-Density Ratio
COG	Course over Ground
COGM	Course over Ground (in Magnetic North Course Direction)
COGT	Course over Ground (in True North Course Direction)
DGPS	Differential Global Positioning System
DOP	Dilution of Precision

Abbreviation	Description
EASY	Embedded Assist System
ECEF	Earth-Centered, Earth-Fixed
EGNOS	European Geostationary Navigation Overlay Service
EPO	Extended Prediction Orbit
FLP	Fitness Low Power
GAGAN	GPS Aided GEO Augmented Navigation
GGA	Global Positioning System Fix Data
GLL	Geographic Position – Latitude/Longitude
GLONASS	Global Navigation Satellite System (Russia)
GLP	GPS Low Power
GNS	GNSS Fix Data
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GRS	GNSS Range Residuals
GSA	GNSS DOP and Active Satellites
GST	GNSS Pseudorange Error Statistics
GSV	GNSS Satellites in View
HDOP	Horizontal Dilution of Precision
ID	Identifier
LLH	Latitude Longitude Height
MNL	Middleware Navigation Library
MSAS	Multi-functional Satellite Augmentation System
NMEA	NMEA (National Marine Electronics Association) 0183 Interface Standard
NTSC	National Time Service Center

Abbreviation	Description
NVM	Non-Volatile Memory
NVRAM	Non-Volatile Random Access Memory
PAIR	Proprietary Protocol of Airoha
PDOP	Position Dilution of Precision
PPS	Pulse Per Second
QZSS	Quasi-Zenith Satellite System
RAIM	Receiver Autonomous Integrity Monitoring
RMC	Recommended Minimum Specific GNSS Data
RMS	Root Mean Square
RLM	Return Link Message
RLS	Return Link Service
RTC	Real-time Clock
RTCM	Radio Technical Commission for Maritime Services
RTK	Real Time Kinematic
SBAS	Satellite-Based Augmentation System
SNR	Signal-to-noise Ratio
SOG	Speed over Ground
SPS	Standard Positioning Service
SV	Satellites in View
TRAIM	Timing Receiver Autonomous Integrity Monitoring
UART	Universal Asynchronous Receiver/Transmitter
USNO	United States Naval Observatory
UTC	Coordinated Universal Time
VDOP	Vertical Dilution of Precision

Abbreviation	Description
VTG	Course Over Ground and Ground Speed
WAAS	Wide Area Augmentation System
ZDA	Time and Date

5 Appendix B GNSS Numbering

Table 13: GNSS Satellites (NEMA) Numbering

GNSS Type	System ID	Satellite ID	Signal ID
GPS	1	1–32 (33–51 for SBAS)	1 = L1 C/A
GLONASS	2	65–88	1 = L1
Galileo	3	1–36	7 = E1
BDS	4	1–63	1 = B1I 3 = B1C
QZSS	1	193–199	-

6 Appendix C Special Characters

Table 14: 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.