LC26G (AB)&LC76G&LC86G Series

GNSS Protocol Specification

GNSS Module Series

Version: 1.1

Date: 2022-12-20

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 (AB)&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 (*Chapter 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 (*Chapter 2.3.13* and *2.3.14*). 7. Added the note in $PAIR050, $PAIR062, $PAIR066, $PAIR080 and $PAIR081 (*Chapter 0*, *2.3.13*, *2.3.15* and *2.3.23*). 8. Added $PAIR154, $PAIR155, $PAIR158, $PAIR680, $PAIR681, $PAIR690, $PAIR691, $PAIR730, $PAIR731, $PAIR732, $PAIR733, $PAIR900, $PAIR901, $PAIR902, $PAIR903, $PAIR904, $PAIR905, $PAIR906, $PAIR907, $PAIR908, $PAIR909 commands (*Chapter 2.3.27*, *2.3.28*, *2.3.29*, *2.3.47*, *2.3.48*, *2.3.49*, *2.3.50*, *2.3.51*, *2.3.52*,

Version	Date	Description
		2.3.53, 2.3.54, 2.3.60, 2.3.61, 2.3.62, 2.3.63, 2.3.64, 2.3.65, 2.3.66, 2.3.67, 2.3.68, 2.3.69 and 2.3.70). 9. Added the format and field description of binary data (Chapter 2.3.58). 10. Added PQTM messages (Chapter 2.4).

Contents

Table Index

1 Introduction

Quectel LC26G (AB), LC76G and LC86G series GNSS modules support GPS, GLONASS, Galileo, BDS and QZSS constellations. Concurrent tracking of multi-frequency bands provides fast and accurate acquisition and makes these modules ideal solutions for positioning and navigation in various vertical markets.

This document describes the software commands that are 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.

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 + QZSS L1 C/A
LC76G Series	LC76G (AB)	GPS L1 C/A + GLONASS L1 + Galileo E1 + BDS B1I + QZSS L1 C/A
	LC76G (PA)	GPS L1 C/A + GLONASS L1 + Galileo E1 + BDS B1I + QZSS L1 C/A
	LC76G (PB)	GPS L1 C/A + GLONASS L1 + Galileo E1 + BDS B1I + QZSS L1 C/A
LC86G Series	LC86G (AA)	GPS L1 C/A + Galileo E1 + BDS B1I
	LC86G (AB)	GPS L1 C/A + GLONASS L1 + Galileo E1
	LC86G (LA)	GPS L1 C/A + GLONASS L1 + Galileo E1 + BDS B1I + QZSS L1 C/A

LC26G (AB), LC76G 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), LC76G series and LC86G series module will hereinafter be referred to collectively as “the module/modules” in parts hereof applicable to all modules, and individually as “LC26G (AB)”, “LC76G series” and “LC86G series” in parts hereof referring to the differences between them.

2 NMEA Protocol

2.1. Structure of NMEA Protocol Messages

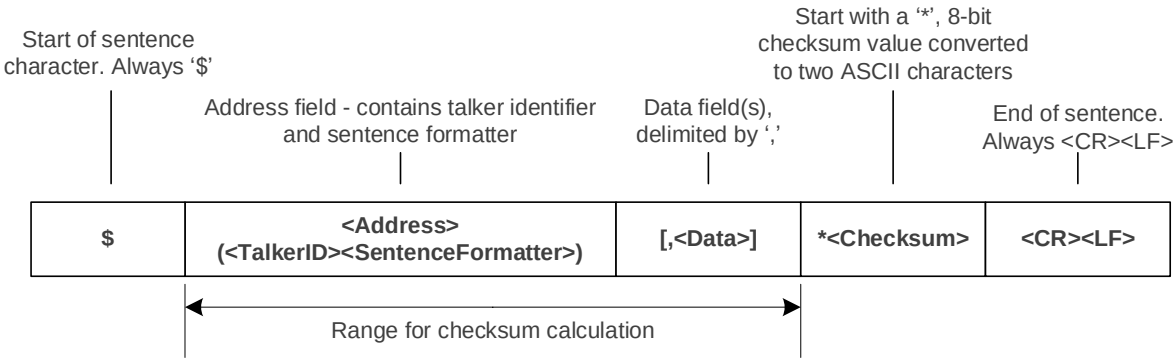

Figure 1: Structure of NMEA Protocol Messages

Table 3: Structure of NMEA Protocol Messages

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

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

Table 4: NMEA Talker ID

GNSS Constellation Configuration	TalkerID (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 NMEA 0183 V4.10 standard 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>
```

Parameters:

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

Field	Format	Unit	Example	Description
				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) mode.

Field	Format	Unit	Example	Description
				F = Float RTK. Satellite system used in RTK mode with floating integers. M = Manual input mode N = No fix. Satellite system not used for position fixing, or fix not valid. R = Real Time Kinematic (RTK). Satellite system used in RTK mode with fixed integers.
<NavStatus>	Character	-	V	Navigational status. Not supported. 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 later 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>
```

Parameters:

Field	Format	Unit	Example	Description
$	Character	-	$	Each NMEA message starts with **$**.

Field	Format	Unit	Example	Description
<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) 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 3 = GPS PPS Mode, fix valid 4 = Real Time Kinematic (RTK) System used in RTK mode with

Field	Format	Unit	Example	Description
				fixed integers 5 = Float RTK. Satellite system used in RTK mode, floating integers 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). 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>
```

Parameters:

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 9: 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.

Field	Format	Unit	Example	Description
End of repeat block.				
<SignalID>	Numeric	-	1	GNSS signal ID. See *Table 9: GNSS Satellites* (NEMA) Numbering. Note that this parameter is only available in messages in line with NMEA 0183 V4.10 and later versions.
<Checksum>	Hexadecimal	-	*6D	Checksum.
<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 sentence, and DOP values.

Type:

Output

Synopsis:

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

<CR><LF>

Parameters:

Field	Format	Unit	Example	Description
$	Character	-	$	Each NMEA message starts with **$**.
<TalkerID>	String, 2 characters	-	GN	Talker identifier. See *Table 4: NMEA Talker ID*.
GSA	String, 3 characters	-	GSA	GNSS DOP and Active Satellites.
<Mode>	Character	-	A	Selection of 2D or 3D fix. M = Manual, forced to operate in 2D or 3D mode A = Automatic, allowed to automatically switch to 2D/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 9: 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: 99.00. Note that this field is empty in case of an invalid value.
<HDOP>	Numeric	-	0.48	Horizontal dilution of precision. Maximum value: 99.00. Note that this field is empty in case of an invalid value.
<VDOP>	Numeric	-	0.52	Vertical dilution of precision. Maximum value: 99.00. Note that this field is empty in case of an invalid value.
<SystemID>	Numeric	-	1	GNSS system ID. See *Table 9: GNSS Satellites* (NEMA) Numbering.

Field	Format	Unit	Example	Description
				Please note that this parameter is only available in messages in line with NMEA 0183 V4.10 or later 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. If more than 12 satellites are used for navigation, only the IDs of the first 12 are output.

2.2.5. VTG

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

Type:

Output

Synopsis:

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

Parameters:

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	Degrees	0.00	Course over ground, in true north course

Field	Format	Unit	Example	Description
				direction. Note that this field is empty in case of an invalid value.
T	Character	-	T	Fixed field: true.
<COGM>	Numeric	Degrees	-	Course over ground (magnetic). Not supported.
M	Character	-	M	Fixed field: magnetic.
<SOGN>	Numeric	Knots	0.01	Speed over ground in knots. Note that this field is empty in case of an invalid value.
N	Character	-	N	Fixed field: knot.
<SOGK>	Numeric	km/h	0.02	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 D = Differential mode E = Estimated (dead reckoning) mode F = Float RTK. Satellite system used in real time kinematic mode with floating integers M = Manual input mode N = No fix. Satellite system not used for position fixing, or fix not valid R = Real Time Kinematic. Satellite system used in RTK mode with fixed integers
<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>
```

Parameters:

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.

Field	Format	Unit	Example	Description
<UTC>	hhmmss.sss	-	040143.000	Position 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 D = Differential mode E = Estimated (dead reckoning) mode F = Float RTK. Satellite system used in real time kinematic mode with floating integers M = Manual input mode N = No fix. Satellite system not used for position fixing, or fix not valid R = Real Time Kinematic. Satellite system used in RTK mode with fixed integers.
<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>
```

Parameters:

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&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.
<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
```

NOTE

Due to firmware limitation, LC26G (AB), LC76G and LC86G series modules do not support local time outputting.

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>
```

Parameters:

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
<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.

Field	Format	Unit	Example	Description
<ModeInd>[1)]	Character	-	DANN	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) mode F = Float RTK. Satellite system used in RTK mode with floating integers M = Manual input mode N = No fix. Satellite system not used for position fixing, or fix not valid R = Real Time Kinematic (RTK). Satellite system used in RTK mode with fixed integers.
<NumSatUsed>	Numeric	-	16	Total number of satellites in use. Range: 0–99.
<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.

Field	Format	Unit	Example	Description
<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. [1] **<ModeInd>** is a variable length field. The first character indicates the use of GPS satellites, the second character indicates the use of GLONASS satellites, and the third character indicates the use of Galileo satellites. The fourth character indicates the use of BDS satellites, the fifth character indicates the use of QZSS satellites, and the sixth character indicates the use of NavIC (IRNSS) satellites.
2. LC76G (AB) with LC76GABNR02A01, LC26G (AB) with LC26GABNR02A01, and LC86G (LA) with LC86GLANR02A01 versions and their previous versions do not support **GNS** message outputting.

2.2.9. GST

GNSS Psuedorange 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>
```

Parameters:

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 Psuedorange Error Statistics.
<UTC>	hhmmss.sss	-	123624.000	UTC time of the **GGA** or **GNS** fix associated with this sentence.
<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	Degrees	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.
<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>
```

Parameters:

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	Residual calculation mode. 0 = Residuals were used to calculate the position given in the matching GGA or GNS sentence 1 = Residuals were recomputed after the GGA or GNS position was computed
Start of repeat block. Repeat 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 9: GNSS Satellites* (NEMA) Numbering. Note that this parameter is only available in messages in line with NMEA 0183 V4.10 or later versions.
<SignalID>	Numeric	-	1	GNSS signal ID. See *Table 9: GNSS Satellites* (NEMA) Numbering. Note that this parameter is only available in messages in line with

Field	Format	Unit	Example	Description
				NMEA 0183 V4.10 or later 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
```

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>
```

Parameters:

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	-	9A22BE29630F010	Beacon of RLM. 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 Galieo 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
<CR><LF>	Character	-	-	Carriage return and line feed.

LC76G (AB) Example:

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

NOTE

LC76G (AB) with LC76GABNR02A01, LC26G (AB) with LC26GABNR02A01, LC86G (LA) with LC86GLANR02A01 versions and their previous versions do not support outputting **RLM** message.

2.3. PAIR Messages

This chapter explains PAIR messages (proprietary NMEA messages defined by the chipset supplier) supported by the modules. For LC86G series, to ensure successful execution of the commands in this chapter, send them once or multiple times until **$PAIR001** is returned.

2.3.1. Packet Type: 001 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>
```

Parameters:

Field	Format	Unit	Description
<CommandID>	Numeric	-	Type of command/packet to be acknowledged.
<Result>	Numeric	-	0 = Command has been successfully sent 1 = Command is being processed. Please wait for the result. 2 = Command sending failed. 3 = **<commandID>** is not supported. 4 = Command parameter error. Out of range/Some parameters were lost/Checksum error. 5 = MNL service is busy. You can try again soon.

Example:

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

2.3.2. Packet Type: 002 PAIR_GNSS_SUBSYS_POWER_ON

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

Type:

Command

Synopsis:

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

Parameter:

None

Result:

Returns **$PAIR001** message.

Example:

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

NOTE

After powering off the GNSS system with **$PAIR003*39**, LC86G no longer supports using this command to power on the GNSS system.

2.3.3. Packet Type: 003 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. For details about Standby mode, see *documents [1]*, *[2]* and *[3] Hardware Design*.

Type:

Command

Synopsis:

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

Parameter:

None

Result:

Returns **$PAIR001** message.

Example:

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

NOTE

1. For LC86G series, if it receives **$PAIR382,1*2E** before receiving **$PAIR003*39**, and **$PAIR001,382,0*32** is returned correctly, then it can still receive other commands after entering Standby mode. Otherwise, any other commands will not be responded.
2. LC26G (AB) and LC76G series can receive commands after sending **$PAIR003*39**.

2.3.4. Packet Type: 004 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
```

2.3.5. Packet Type: 005 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 fix a 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
```

2.3.6. Packet Type: 006 PAIR_GNSS_SUBSYS_COLD_START

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

Type:

Command

Synopsis:

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

Parameter:

None

Result:

Returns **$PAIR001** message.

Example:

```
$PAIR006*3C
$PAIR001,006,1*3C
$PAIR001,006,0*3D
```

2.3.7. Packet Type: 007 PAIR_GNSS_SUBSYS_FULL_COLD_START

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

2.3.8. Packet Type: 010 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>
```

Parameters:

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
```

NOTE

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

2.3.9. Packet Type: 050 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 is greater than 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. LC76G (PA) and LC76G (PB) modules do not support this command. The position fix rate remains at 1 Hz.
3. If the default value is not given for any parameter in a Set command, you can query it with the corresponding Get command provided that the default setting has not been changed by Set command.

If the default setting had been changed by Set command, contact Quectel Technical Support (support@quectel.com) to get the default setting if necessary.

2.3.10. Packet Type: 051 PAIR_COMMON_GET_FIX_RATE

Gets the position fix interval.

Type:

Get

Synopsis:

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

Parameter:

None

Result:

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

Query result message format:

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

Parameter included in the result:

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

Example:

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

2.3.11. Packet Type: 058 PAIR_COMMON_SET_MIN_SNR

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

Type:

Set

Synopsis:

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

Parameter:

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

Result:

Returns **$PAIR001** message.

Example:

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

2.3.12. Packet Type: 062 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>

Parameters:

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

Field	Format	Unit	Description
			4 = NMEA_SEN_RMC 5 = NMEA_SEN_VTG 6 = NMEA_SEN_ZDA 7 = NMEA_SEN_GRS 8 = NMEA_SEN_GST 9 = NMEA_SEN_GNS
<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. Default value: 1.

Result:

Returns **$PAIR001** message.

Example:

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

NOTE

LC76G (AB) with LC76GABNR02A01S, LC26G (AB) with LC26GABNR02A01S, and LC86G (LA) with LC86GLANR02A01S versions and their previous versions do not support **GNS** message outputting.

2.3.13. Packet Type: 059 PAIR_COMMON_GET_MIN_SNR

Gets the minimum SNR of satellites in use.

Type:

Get

Synopsis:

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

Parameter:

None

Result:

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

Query result message format:

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

Parameter included in the result:

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

Example:

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

2.3.14. Packet Type: 063 PAIR_COMMON_GET_NMEA_OUTPUT_RATE

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

Type:

Get

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<Type>	Numeric	-	Type of standard NMEA sentence. -1 = Return the output rates of all types of standard NMEA sentences. 0 = NMEA_SEN_GGA 1 = NMEA_SEN_GLL 2 = NMEA_SEN_GSA 3 = NMEA_SEN_GSV 4 = NMEA_SEN_RMC 5 = NMEA_SEN_VTG 6 = NMEA_SEN_ZDA 7 = NMEA_SEN_GRS 8 = NMEA_SEN_GST 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 per every N position fix(es) Range: 1–20. Default value: 1.

Example:

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

2.3.15. Packet Type: 066 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_Enabl
```

ed>,<Reserved>*<Checksum><CR><LF>

Parameters:

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
<Reserved>	Numeric	-	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), LC86G (LA) and LC76G series:
   - 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) and LC86G (AB) do not support this command. Currently only the default satellite constellation configuration is supported. Details are as follows:

- LC86G (AA): GPS + Galileo + BDS
- LC86G (AB): GPS + GLONASS + Galileo

2.3.16. Packet Type: 067 PAIR_COMMON_GET_GNSS_SEARCH_MODE

Gets the GNSS search mode.

Type:

Get

Synopsis:

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

Parameter:

None

Result:

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

Query result message format:

```
$PAIR067<GPS_Enabled>,<GLONASS_Enabled>,<Galileo_Enabled>,<BDS_Enabled>,<QZSS_Enabled>,<Reserved>*,<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
<Reserved>	Numeric	-	Always “0”

Example:

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

2.3.17. Packet Type: 070 PAIR_COMMON_SET_STATIC_THRESHOLD

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

Type:

Set

Synopsis:

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

Parameter:

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

Result:

Returns **$PAIR001** message.

Example:

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

2.3.18. Packet Type: 071 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

2.3.19. Packet Type: 072 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.

2.3.20. Packet Type: 073 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
```

2.3.21. Packet Type: 074 PAIR_COMMON_SET_AIC_ENABLE

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

Type:

Set

Synopsis

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

Parameter:

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

Result:

Returns **$PAIR001** message.

Example:

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

2.3.22. Packet Type: 075 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
```

2.3.23. Packet Type: 080 PAIR_COMMON_SET_NAVIGATION_MODE

Sets navigation mode.

Type:

Set

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<NavMode>	Numeric	-	Navigation mode. 0 = Normal mode. 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 = Reserved. 4 = Reserved.

	5 = Drone mode: Used for drone applications with equivalent dynamic range and vertical acceleration at different flight phases. (For example, hovering, cruising). 6 = Reserved. 7 = Swimming mode: Used for swimming purposes to smooth the trajectory and improve the accuracy of distance calculation.

Result:

Returns **$PAIR001** message.

Example:

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

NOTE

When the **<NavMode>** = 5, the decimal places of the **<Lat>** and **<Lon>** field in **RMC**, **GGA**, **GLL** and **GNS** messages change from six to seven except LC76GABNR02A01S, LC26GABNR02A01S and LC86GLANR02A01S versions and their previous versions.

2.3.24. Packet Type: 081 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 = Reserved. 4 = Reserved. 5 = Drone mode: Used for drone applications with equivalent dynamic range and vertical acceleration at different flight phases. (For example, hovering, cruising) 6 = Reserved. 7 = Swimming mode: Used for swimming purpose to smooth the trajectory and improve the accuracy of distance calculation.

Example:

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

2.3.25. Packet Type: 086 PAIR_COMMON_SET_DEBUGLOG_OUTPUT

Enables/disables debug log output in binary format.

Type:

Set

Synopsis

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

Parameter:

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

Result:

Returns **$PAIR001** message.

Example:

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

2.3.26. Packet Type: 087 PAIR_COMMON_GET_DEBUGLOG_OUTPUT

Queries the debug log output setting.

Type:

Get

Synopsis

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

Parameter:

None

Result:

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

Query result message format:

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

Parameter included in the result:

Field	Format	Unit	Description
<Status>	Numeric	-	Debug log output setting. 0 = Disabled 1 = Enabled

Example:

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

2.3.27. Packet Type: 154 PAIR_COMMON_SET_RLM_OUTPUT_ENABLE

Enables/disables outputting of **RLM** message.

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 periodically (1 Hz).

Example:

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

NOTE

LC76G (AB) with LC76GABNR02A01S, LC26G (AB) with LC26GABNR02A01S, LC86G (LA) with LC86GLANR02A01S versions and their previous versions do not support this message.

2.3.28. Packet Type: 155 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
```

2.3.29. Packet Type: 158 PAIR_COMMON_SET_B1C_ENABLE

Enable/Disable tracking of BDS B1C 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
```

NOTE

1. LC26G (AB) module currently can not support this message.
2. For LC76G series and LC86G series, LC76GPANR02A02S, LC76GPBNR02A02S, LC86GLANR02A02S versions and their later versions support this message when the satellite constellation configuration is GPS + BDS (+QZSS).

2.3.30. Packet Type: 382 PAIR_TEST_LOCK_SYSTEM_SLEEP

Enables/disables the locking of Sleep mode. The CPU core will not enter Sleep mode automatically after the command is sent.

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

1. 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. LC86G module can only receive commands in Sleep mode when it receives **$PAIR382,1*2E** before receiving **$PAIR003*39** and correctly return **$PAIR001,382,0*32**. Otherwise, any other commands can not be received.
3. LC86G series modules can not receive any commands automatically. To avoid that, it is suggested to send **$PAIR382,1*2E** before sending other commands, and ensure that **$PAIR001,382,0*32** is returned correctly.

2.3.31. Packet Type: 391 PAIR_TEST_JAMMING_DETECT

Enables/disables jamming detection. The jamming status messages will be returned when jamming detection 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 2.3.70 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

The module starts jamming detection once the feature is enabled.

1. If there is no jamming, **$PAIRSPF,1*52** will be reported to indicate good status (status 1).
2. In case of continuous jamming, the jamming status will change from 1 to 2 and finally to 3.

1) When no position fix has been completed: module status is 1 right after jamming detection is enabled, and then changes to 2 when jamming is detected. During this process, the module keeps attempting to get a fix; if the anti-jamming repair fails, the jamming status changes to 3.
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.

2.3.32. Packet Type: 400 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 1 = RTCM 2 = SBAS (Including WAAS/EGNOS/GAGAN/MSAS)

Result:

Returns a **$PAIR001** message.

Example:

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

2.3.33. Packet Type: 401 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 1 = RTCM 2 = SBAS (Including WAAS/EGNOS/GAGAN/MSAS)

Example:

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

2.3.34. Packet Type: 410 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

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

2.3.35. Packet Type: 411 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
```

2.3.36. Packet Type: 432 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 1 = Enable outputting RTCM3 with message type MSM7

Result:

Returns **$PAIR001** message.

Example:

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

2.3.37. Packet Type: 433 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

Field	Format	Unit	Description
			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
```

2.3.38. Packet Type: 434 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:

```
$PQTM434,1*36
$PAIR001,432,0*3E
```

2.3.39. Packet Type: 435 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
```

2.3.40. Packet Type: 436 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:

```
$PQTM436,1*34
$PAIR001,436,0*3A
```

2.3.41. Packet Type: 437 PAIR_RTCM_GET_OUTPUT_EPHEMERIS

Queries the setting of 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
```

2.3.42. Packet Type: 490 PAIR_EASY_ENABLE

Enables/disables 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
```

2.3.43. Packet Type: 491 PAIR_EASY_GET_STATUS

Queries the status of 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 = Disable 1 = Enable
<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
```

2.3.44. Packet Type: 511 PAIR_NVRAM_SAVE_NAVIGATION_DATA

Saves current navigation data from RTC RAM to flash.

Type:

Command

Synopsis

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

Parameter:

None

Result:

Returns **$PAIR001** message.

Example:

```
$PAIR511*3F
$PAIR001,511,1*3F
$PAIR001,511,0*3E
```

NOTE

1. If RTC cannot be powered after module power supply is cut off, this command must be sent every time the parameters are modified.
2. In case of fix rates greater than 1 Hz, power off the GNSS system with **$PAIR382,1*2E** and **$PAIR003*39** 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.

2.3.45. Packet Type: 513 PAIR_NVRAM_SAVE_SETTING

Saves the current configurations from RTC RAM to flash.

Type:

Command

Synopsis:

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

Parameter:

None

Result:

Returns **$PAIR001** message.

Example:

```
$PAIR513*3D
$PAIR001,513,0*3C
```

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. In multi-Hz, power off the GNSS system with **$PAIR382,1*2E** and **$PAIR003*39** before sending this command. After sending **$PAIR513*3D**, send **$PAIR002*38** again to power on the module. This limitation does not apply to frequencies below 1 Hz.

2.3.46. Packet Type: 650 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]*, *[2]* and *[3] hardware designs*.

Type:

Set

Synopsis:

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

Parameter:

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

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
```

2.3.47. Packet Type: 680 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 in different environments.

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 is 1 Hz.
2) Satellite constellation configuration is GPS only.
3) Navigation mode is 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.

2.3.48. Packet Type: 681 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
```

2.3.49. Packet Type: 690 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>
```

Parameters:

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	Run time. Range: 3–518400.
<FirstSleep>	Numeric	Second	Sleep time. Range: 3–518400.
<SecondRun>	Numeric	Second	Second run time. Range: 0 or 3–518400.
<SecondSleep>	Numeric	Second	Second sleep 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 run time” instead of “first run time” setting when there is no signal. The second run time can be “0” only when the second sleep time is “0”.
4. **<SecondSleep>:** GNSS module will use “second sleep time” instead of “first sleep time” setting when there is no signal. The second sleep time can be “0” only when the second run time is “0”.

2.3.50. Packet Type: 691 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	Run time. Range: 3–518400.
<FirstSleep>	Numeric	Second	Sleep time. Range: 3–518400.
<SecondRun>	Numeric	Second	Second run time. Range: 0 or 3–518400.
<SecondSleep>	Numeric	Second	Second sleep time. Range: 0 or 3–518400.

Example:

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

2.3.51. Packet Type: 730 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

Requirements for entering FLP mode:

1. Fix rate is 1 Hz.
2. Navigation mode is fitness mode. LC76GABNR02A01S, LC26GABNR02A01S and LC86GLANR02A01S versions and their previous versions do not need to change the navigation mode to fitness mode.
3. Supported satellite constellation configuration:
   - GPS + GLONASS + Galileo + BDS +(QZSS)
   - GPS + GLONASS +(QZSS)
   - GPS+ BDS +(QZSS)

2.3.52. Packet Type: 731 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	-	Enable/disable FLP mode. 0 = Disable 1 = Enable

Example:

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

NOTE

1. Requirements for querying FLP mode:
1) Fix rate is 1 Hz.
2) Navigation mode is fitness mode.
3) Supported satellite constellation configuration:
   - GPS + GLONASS + Galileo + BDS +(QZSS)
   - GPS + GLONASS +(QZSS)
   - GPS+ BDS +(QZSS)
2. LC76G (AB) with LC76GABNR02A01S, LC26G (AB) with LC26GABNR02A01S, LC86G (LA) with LC86GLANR02A01S versions and their previous versions only need to meet the first requirement for fix rate.

2.3.53. Packet Type: 732 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

Result:

Returns **$PAIR001** message.

Example:

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

NOTE

1. Requirements for entering ALP mode:
1) Fix rate is 1 Hz.
2) Navigation mode is normal mode.
3) Supported satellite constellation configuration:
   - GPS + GLONASS + Galileo + BDS +(QZSS)
   - GPS + GLONASS +(QZSS)
2. LC76GABNR02A01S, LC26GABNR02A01S and LC86GLANR02A01S versions and their previous versions do not support this message.
3. When the ALP mode is enabled, some of the features will be automatically disabled, such as SBAS, low power periodic mode, FLP mode, and GLP mode.

2.3.54. Packet Type: 733 PAIR_ALP_GET_STATUS

Queries adaptive low-power (ALP) mode configuration 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	-	Enable/disable ALP mode. 0 = Disable 1 = Enable

Example:

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

NOTE

1. Requirements for querying ALP mode:
1) Fix rate is 1 Hz.
2) Navigation mode is normal mode.
2. LC76G (AB) with LC76GABNR02A01S, LC26G (AB) with LC26GABNR02A01S, LC86GLANR02A01S versions and their previous versions do not support this message.

2.3.55. Packet Type: 752 PAIR_PPS_SET_CONFIG_CMD

Sets PPS configurations.

Type:

Set

Synopsis:

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

Parameters:

Field	Format	Unit	Description
<PPSType>	Numeric	-	PPS pulse type. 0 = Disable 1 = After the first fix 2 = 3D fix only 3 = 2D/3D fix only 4 = Always
<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
```

2.3.56. Packet Type: 864 PAIR_IO_SET_BAUDRATE

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

Type:

Set

Synopsis:

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

Parameters:

Field	Format	Unit	Description
<PortType>	Numeric	-	Hardware port type. 0 = UART
<PortIndex>	Numeric	-	Hardware port index. 0 = UART0
<Baudrate>	Numeric	bps	Baud rate. 9600

Field	Format	Unit	Description
			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.

2.3.57. Packet Type: 865 PAIR_IO_GET_BAUDRATE

Gets the baud rate of UART interface.

Type:

Get

Synopsis:

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

Parameters:

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 115200 230400 460800 921600

Example:

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

2.3.58. Packet Type: 890 PAIR_GEOFENCE_SET_CONFIG

Sets geofence configuration. For details about geofence, see *documents [1]*, *[2]* and *[3] hardware designs*.

Type:

Set

Synopsis:

```
$PAIR890,<FenceNum>,<ConfLvl>{,<Lat1>,<Lon1>,<Rad1>}*<Checksum><CR><LF>
```

Parameters:

Field	Format	Unit	Description
<FenceNum>	Numeric	-	Number of geofences. Maximum value: 4. When it is set to 0, the geofence function is disabled.
<ConfLvl>	Numeric	-	Confidence level for state evaluation. 0 = No requirement 1 = 1-Sigma (68 %)

Field	Format	Unit	Description
			2 = 2-Sigma (95 %) 3 = 3-Sigma (99.7 %)
Start of repeat block. Repeat times: 1–**<FenceNum>**.			
<Lat>	Numeric	Degree	Latitude of the geofence circle center
<Lon>	Numeric	Degree	Longitude of the geofence circle center
<Rad>	Numeric	Meter	Radius of the geofence circle
End of repeat block.			

Result:

- When **<FenceNum>**=0, module returns **$PAIR001** message.
- When **<FenceNum>**≠0, module returns **$PAIR001** message and the binary data which is parsed to hexadecimal format then. Hexadecimal parsed data are used in the following example.

Data Field:

Parameter	Length (byte)	Description
Preamble	2	2-byte word (0x2404).
MessageID	2	Geofencing status (0x07DD).
Length	2	Length of the data field, excluding preamble, message ID, length, checksum and end words fields.
Data	12	Payload.
Checksum	1	The checksum is the 8-bit exclusive OR of all bytes in the packet between but excluding preamble, checksum and end words.
End words	2	2-byte word (0x44AA).

Message format:

Preamble	MessageID	Length	Data	Checksum	End Words
0x04 0x24	0xDD 0x07	0x0C 0x00	See *Table 5: Payload*.	0x**	0xAA 0x44

Table 5: Payload

Byte Offset	Length (byte)	Name	Unit	Description
0	1	Status	-	Geofencing status: 0 = Geofencing disabled or not reliable 1 = Geofencing enabled
1	1	Fencenum	-	Number of geofences
2	1	State	-	Combined state of all geofences: 0 = Outside 1 = Unknown 2 = Inside
3	4	EachState[4]	-	Geofence state 0 = Outside 1 = Unknown 2 = Inside
7	1	Hour	-	Hour of day, 0–23 (UTC)
8	1	Min	-	Minute of hour, 0–59 (UTC)
9	1	Sec	-	Second of hour, 0–59 (UTC)
10	2	Msec	-	Millisecond of second, 0–999 (UTC)

Example:

```
//Enable the geofencing function:
$PAIR890,1,1,25.0567,121.5743,30*20
$PAIR001,890,0*3A
04 24 DD 07 0C 00 01 01 00 00 00 00 00 05 27 39 00 00 CD AA 44

//Disable the geofencing function:
$PAIR890,0*27
$PAIR001,890,0*3A
```

NOTE

1. When **<FenceNum>**=0, the other parameters should be omitted in **$PAIR890**.
2. When **<FenceNum>**≠0, the **$PAIR890** parameter is in the Synopsis. The binary data will be output along with the NMEA message after the command is sent successfully.
3. In binary message, all multi-byte values are ordered in Little Endian format.

2.3.59. Packet Type: 891 PAIR_GEOFENCE_GET_CONFIG

Queries the geofence configuration.

Type:

Get

Synopsis:

```
$PAIR891*<Checksum><CR><LF>
```

Parameter:

None

Result:

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

Query result message format:

```
$PAIR891,<FenceNum>,<ConfLvl>{,<Lat1>,<Lon1>,<Rad1>}*<Checksum><CR><LF>
```

Parameters included in the result:

Field	Format	Unit	Description
<FenceNum>	Numeric	-	Number of geofences. Maximum value: 4 When it is set to 0, the geofence function is disabled.
<ConfLvl>	Numeric	-	Confidence level for state evaluation. 0 = No requirement 1 = 1-Sigma (68 %) 2 = 2-Sigma (95 %) 3 = 3-Sigma (99.7 %)
Start of repeat block. Repeat times: 1–**<FenceNum>**.			
<Lat>	Numeric	Degree	Latitude of the geofence circle center
<Lon>	Numeric	Degree	Longitude of the geofence circle center
<Rad>	Numeric	Meter	Radius of the geofence circle
End of repeat block.			

Example:

```
$PAIR891*3A
$PAIR001,891,0*3B
$PAIR891,1,1,31.8222168,117.1158337,30*2F
```

2.3.60. Packet Type: 900 PAIR_LOCUS_ENABLE

Enables/disables LOCUS to save fix data.

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. LC26G (AB) with LC26GABR02A01S, LC76G (AB) with LC76GABR02A01S, LC76G (PA) with LC76GPAR02A01S, LC76G (PB) with LC76GPBR02A01S, LC86G (LA) with LC86GLAR02A01S versions and their previous versions do not support this message.
2. Saved fix data: UTC time, fix status, longitude, latitude, altitude, ground speed, heading degree, horizontal dilution of precision, and number of satellites used.
3. Same configuration is not allowed to be set after the first successful execution of **$PAIR900**.

2.3.61. Packet Type: 901 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 = Disable 1 = Enable

Example:

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

2.3.62. Packet Type: 902 PAIR_LOCUS_SET_MODE

Sets LOCUS saving mode.

Type:

Set

Synopsis

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

Parameters:

Field	Format	Unit	Description
<Mode>	Hexadecimal	-	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 2.3.64 Packet Type: 904 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 2.3.64 Packet Type: 904 PAIR_LOCUS_SET_THRESHOLD*. Bit 3= 1. Distance-triggered save mode. Record once after the distance threshold is met. For details, see *Chapter 2.3.64 Packet Type: 904 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 2.3.67 Packet Type: 907 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 & 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.

2.3.63. Packet Type: 903 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>	Hexadecimal	-	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 2.3.64 Packet Type: 904 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 2.3.64 Packet Type: 904 PAIR_LOCUS_SET_THRESHOLD*. Bit 3= 1. Distance-triggered save mode. Record once after the distance threshold is met. For details, see *Chapter 2.3.64 Packet Type: 904*

Field	Format	Unit	Description
			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 2.3.67 Packet Type: 907 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
```

2.3.64. Packet Type: 904 PAIR_LOCUS_SET_THRESHOLD

Sets LOCUS mode threshold.

Type:

Set

Synopsis

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

Parameters:

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	Meter second m/s	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.

Field	Format	Unit	Description
			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.

2.3.65. Packet Type: 905 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	Meter second m/s	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
```

2.3.66. Packet Type: 906 PAIR_LOCUS_CLEAR

Clears LOCUS Data.

Type:

Command

Synopsis:

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

Parameter:

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

Result:

Returns **$PAIR001** message.

Example:

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

2.3.67. Packet Type: 907 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.

2.3.68. Packet Type: 908 PAIR_LOCUS_GET_DATA

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

Returns **$PAIR001** message and enables outputting LOCUS data periodically.

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 outputtings 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.

Field	Format	Unit	Description
<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. 2 bytes 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:
$PAIR908,0*27
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

Make sure to execute **$PAIR900,1*2E** before executing **$PAIR908**.

2.3.69. Packet Type: 909 PAIR_LOCUS_GET_RECORD_NUM

Queries 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
$PAIR909,5699*15

2.3.70. 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 2 = Warning status 3 = Critical status

Example:

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

//Good status:
$PAIRSPF,1*52

//Warning status:
$PAIRSPF,2*51

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

2.4. PQTM Messages

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

2.4.1. PQTMCFGMSGRATE

Configures 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>
```

Parameters:

Field	Format	Unit	Description
<MsgName>	String	-	Configuration message name/ID. Currently only **$PQTMEPE** can be configured with this command.
<Rate>	Numeric	-	Message output rate. 0 = Output disabled. 1 = Output once per every position fix. N = Output once per every N position fixes. 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>
```

Parameter included in the result:

Field	Format	Unit	Description
<ErrCode>	Numeric	-	Error code. 1 = Invaild parameters. 2 = Execute failed.

Example:

```
//Set the output rate of $PQTMEPE to once per every 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
```

2.4.2. PQTMEPE

Outputs the estimated positioning error.

Type:

Output

Synopsis:

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

Parameters:

Field	Format	Unit	Description
<MsgVer>	Numeric	-	Message version. Always 2.
<EPE_North>	Numeric	Meter	Estimated north error. n.xxx.
<EPE_East>	Numeric	Meter	Estimated east error. n.xxx
<EPE_Down>	Numeric	Meter	Estimated down error. n.xxx
<EPE_2D>	Numeric	Meter	Estimated 2D position error. n.xxx
<EPE_3D>	Numeric	Meter	Estimated 3D position error. n.xxx

Example:

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

2.4.3. PQTMSAVEPAR

Saves the configurations set via **$PQTM** commands to NVDM.

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>
```

Parameter included in the result:

Field	Format	Unit	Description
<ErrCode>	Numeric	-	Error code. 1 = Invaild parameters. 2 = Execute failed.

Example:

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

2.4.4. 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>
```

Parameter included in the result:

Field	Format	Unit	Description
<ErrCode>	Numeric	-	Error code. 1 = Invaild parameters. 2 = Execute failed.

Example:

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

2.4.5. 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>
```

Parameter included in the result:

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

Example:

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

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 6: 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 7: Related Documents

Document Name
[1] Quectel_LC26G(AB)_Hardware_Design
[2] Quectel_LC76G_Series_Hardware_Design
[3] Quectel_LC86G_Series_Hardware_Design

Table 8: 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
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 Psuedorange Error Statistics
GSV	GNSS Satellites in View
HDOP	Horizontal Dilution of Precision
HW	Hardware
ID	Identifier
MNL	MTK Navigation Lib
MSAS	Multi-functional Satellite Augmentation System
NMEA	NMEA (National Marine Electronics Association) 0183 Interface Standard
NVDM	Non-volatile Data Memory
NVRAM	Non-Volatile Random Access Memory

Abbreviation	Description
PAIR	Proprietary Protocol of MTK
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
UART	Universal Asynchronous Receiver/Transmitter
ULP	Ultra-Low Power
UTC	Coordinated Universal Time
VDOP	Vertical Dilution of Precision
VTG	Course Over Ground and Ground Speed
WAAS	Wide Area Augmentation System
ZDA	Time and Date

5 Appendix B GNSS Satellites (NEMA) Numbering

Table 9: GNSS Satellites (NEMA) Numbering

GNSS Type	System ID	Satellite ID	Signal ID
GPS	1	1–32	1 = L1 C/A
GLONASS	2	65–88	1 = L1
Galileo	3	1–36	7 = E1
BDS	4	1–63	1 = B1I 3 = B1C
QZSS	5	193–199	-
SBAS	-	33–51	-

6 Appendix C Special Characters

Table 10: 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.