LC26G&LC76G&LC86G
GNSS Protocol Specification

GNSS Module Series

Version: 1.0.0

Date: 2022-05-13

Status: Preliminary

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&LC76G&LC86G GNSS Protocol Specification**
Subtitle	GNSS Module Series
Document Type	GNSS Protocol Specification
Document Status	Preliminary

Revision History

Version	Date	Description
-	2022-05-13	Creation of the document
1.0.0	2022-05-13	Preliminary

Contents

About the Document 3
Contents 4
Table Index 6

1 Introduction 7

2 NMEA Protocol 8
2.1. Structure of NMEA Protocol Messages 8
2.2. Standard Messages 9
2.2.1. RMC 9
2.2.2. GGA 12
2.2.3. GSV 14
2.2.4. GSA 16
2.2.5. VTG 17
2.2.6. GLL 19
2.3. PAIR Messages 20
2.3.1. Packet Type: 001 PAIR_ACK 21
2.3.2. Packet Type: 002 PAIR_GNSS_SUBSYS_POWER_ON 21
2.3.3. Packet Type: 003 PAIR_GNSS_SUBSYS_POWER_OFF 22
2.3.4. Packet Type: 004 PAIR_GNSS_SUBSYS_HOT_START 23
2.3.5. Packet Type: 005 PAIR_GNSS_SUBSYS_WARM_START 23
2.3.6. Packet Type: 006 PAIR_GNSS_SUBSYS_COLD_START 24
2.3.7. Packet Type: 007 PAIR_GNSS_SUBSYS_FULL_COLD_START 24
2.3.8. Packet Type: 010 PAIR_REQUEST_AIDING 25
2.3.9. Packet Type: 050 PAIR_COMMON_SET_FIX_RATE 26
2.3.10. Packet Type: 051 PAIR_COMMON_GET_FIX_RATE 26
2.3.11. Packet Type: 058 PAIR_COMMON_SET_MIN_SNR 27
2.3.12. Packet Type: 059 PAIR_COMMON_GET_MIN_SNR 28
2.3.13. Packet Type: 062 PAIR_COMMON_SET_NMEA_OUTPUT_RATE 29
2.3.14. Packet Type: 063 PAIR_COMMON_GET_NMEA_OUTPUT_RATE 30
2.3.15. Packet Type: 066 PAIR_COMMON_SET_GNSS_SEARCH_MODE 31
2.3.16. Packet Type: 067 PAIR_COMMON_GET_GNSS_SEARCH_MODE 32
2.3.17. Packet Type: 070 PAIR_COMMON_SET_STATIC_THRESHOLD 33
2.3.18. Packet Type: 071 PAIR_COMMON_GET_STATIC_THRESHOLD 34
2.3.19. Packet Type: 072 PAIR_COMMON_SET_ELEV_MASK 34
2.3.20. Packet Type: 073 PAIR_COMMON_GET_ELEV_MASK 35
2.3.21. Packet Type: 074 PAIR_COMMON_SET_AIC_ENABLE 36
2.3.22. Packet Type: 075 PAIR_COMMON_GET_AIC_STATUS 37
2.3.23. Packet Type: 080 PAIR_COMMON_SET_NAVIGATION_MODE 37
2.3.24. Packet Type: 081 PAIR_COMMON_GET_NAVIGATION_MODE 38
2.3.25. Packet Type: 086 PAIR_COMMON_SET_DEBUGLOG_OUTPUT 39
2.3.26. Packet Type: 087 PAIR_COMMON_GET_DEBUGLOG_OUTPUT 40

2.3.27. Packet Type: 098 PAIR_COMMON_SET_NMEA_POS_DECIMAL_PRECISION 41
2.3.28. Packet Type: 099 PAIR_COMMON_GET_NMEA_POS_DECIMAL_PRECISION 42
2.3.29. Packet Type: 382 PAIR_TEST_LOCK_SYSTEM_SLEEP 42
2.3.30. Packet Type: 391 PAIR_TEST_JAMMING_DETECT 43
2.3.31. Packet Type: 400 PAIR_DGPS_SET_MODE 45
2.3.32. Packet Type: 401 PAIR_DGPS_GET_MODE 45
2.3.33. Packet Type: 410 PAIR_SBAS_ENABLE 46
2.3.34. Packet Type: 411 PAIR_SBAS_GET_STATUS 47
2.3.35. Packet Type: 432 PAIR_RTCM_SET_OUTPUT_MODE 48
2.3.36. Packet Type: 433 PAIR_RTCM_GET_OUTPUT_MODE 48
2.3.37. Packet Type: 490 PAIR_EASY_ENABLE 49
2.3.38. Packet Type: 491 PAIR_EASY_GET_STATUS 50
2.3.39. Packet Type: 511 PAIR_NVRAM_SAVE_NAVIGATION_DATA 51
2.3.40. Packet Type: 513 PAIR_NVRAM_SAVE_SETTING 52
2.3.41. Packet Type: 650 PAIR_LOW_POWER_ENTRY_RTC_MODE 52
2.3.42. Packet Type: 752 PAIR_PPS_SET_CONFIG_CMD 53
2.3.43. Packet Type: 864 PAIR_IO_SET_BAUDRATE 54
2.3.44. Packet Type: 865 PAIR_IO_GET_BAUDRATE 55
2.3.45. Packet Type: 866 PAIR_IO_SET_FLOW_CONTROL 56
2.3.46. Packet Type: 867 PAIR_IO_GET_FLOW_CONTROL 57

3 Appendix A References 59

4 Appendix B GNSS Numbering 61

Table Index

Table 1: Structure of NMEA Protocol Messages 8
Table 2: NMEA Talker ID 9
Table 3: Terms and Abbreviations 59
Table 4: GNSS Numbering 61

1 Introduction

Quectel LC26G&LC76G&LC86G GNSS modules support GPS, GLONASS, Galileo, BDS and QZSS constellations. Concurrent tracking of GPS L1 C/A, GLONASS L1, BDS B1I, Galileo E1 and QZSS L1 C/A frequency bands provides fast and accurate acquisition and makes these modules ideal solutions for positioning and navigation in various vertical markets.

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

NOTE

Only use the commands listed in this document. Quectel assumes no responsibility if other commands are used.

2 NMEA Protocol

2.1. Structure of NMEA Protocol Messages

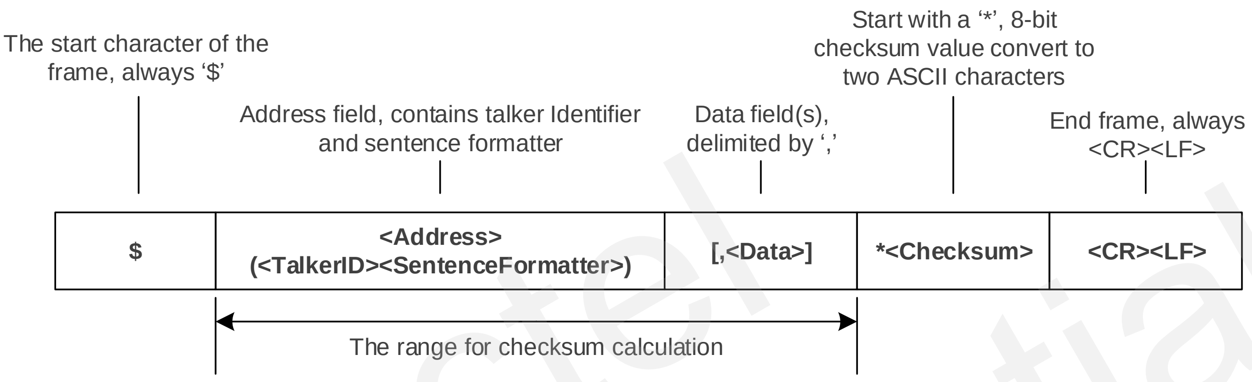
The start character of the frame, always '$'
Address field, contains talker Identifier and sentence formatter
Data field(s), delimited by ','
Start with a '*', 8-bit checksum value convert to two ASCII characters
End frame, always <CR><LF>
$
<Address> (<TalkerID><SentenceFormatter>)
[,<Data>]
*<Checksum>
<CR><LF>
The range for checksum calculation

Figure 1: Structure of NMEA Protocol Messages

Table 1: Structure of NMEA Protocol Messages

Field	Description
$	Start of the sentence (Hex 0x24).
<Address>	**In Standard Messages:** In NMEA standard messages, this field consists of a two-character talker identifier (TalkerID) and a three-character sentence formatter (SentenceFormatter). The talker identifier serves to define the nature of the data being transmitted. For more information on the TalkerID, see *Table 2: NMEA Talker ID.* The sentence formatter is used to define data format and type. **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.
<Data>	Data fields, delimited by comma (**,**). Variable length (depends on the NMEA message type).

<Checksum>	The checksum field follows the checksum delimiter character *****. The checksum is the 8-bit exclusive OR of all characters in the sentence, including the comma (**,**) delimiter, between but not including the **$** and the ***** delimiters.
<CR><LF>	End of the sentence (Hex 0x0D 0x0A).

Table 2: NMEA Talker ID

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

2.2. Standard Messages

This chapter explains the NMEA 0183 V3.01 and 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>
```

Parameter:

Field	Format	Unit	Example	Description
$	-	-	$	Each NMEA message starts with **$**.
<TalkerID>	String, 2 characters	-	GN	Talker identifier. See *Table 2: NMEA Talker ID.*
RMC	String, 3 characters	-	RMC	Recommended Minimum Specific GNSS Data.
<UTC>	hhmmss.sss	-	040143.000	Position fix UTC: hh: Hours (00–23) mm: Minutes (00–59) ss: Seconds (00–59) sss: Decimal fraction of seconds

<Status>	Character	-	A	Positioning system status:
				A = Data valid
				V = Invalid
				D = Differential

<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	Latitude 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	Longitude 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: 359.9. 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>	-	-	-	The direction of magnetic variation. Not supported.
<ModeInd>	Character	-	D	Mode indicator: A = Autonomous mode. Satellite system used in non-differential mode in position fix. D = Differential mode. Satellite system used in differential mode in position fix. Corrections from ground stations or Satellite Based Augmentation System (SBAS). E = Estimated (dead reckoning) mode. F = Float RTK. Satellite system used in RTK mode with floating integers. M = Manual input mode N = No fix. Satellite system not used in position fix, or fix not valid. R = Real Time Kinematic (RTK). Satellite system used in RTK mode with fixed integers.
<NavStatus>	Character	-	V	Navigational status. Not supported. Always "V" (invalid, The device cannot provide navigation status indication). Please note that this parameter is only available in NMEA V4.10 and above.
<Checksum>	Hexadecimal	-	*0E	Checksum.
<CR><LF>	Character	-	-	Carriage return and line feed.

Take LC76G (AA) for 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,<Se
p>,M,<DiffAge>,<DiffStation>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Example	Description
$	-	-	$	Each NMEA message starts with **$.**
<TalkerID>	String, 2 characters	-	GN	Talker identifier. See *Table 2: NMEA Talker ID.*
GGA	String, 3 characters	-	GGA	Global Positioning System Fix Data.
<UTC>	hhmmss.sss	-	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	Latitude 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	Longitude 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 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	-
<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	-
<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.

Take LC76G (AA) for example:

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

NOTE

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

2.2.3. GSV

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

Type:

Output

Synopsis:

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

Parameter:

Field	Format	Unit	Example	Description
$	-	-	$	Each NMEA message starts with **$**.
<TalkerID>	String, 2 characters	-	GP	Talker identifier.

				See *Table 2: NMEA Talker ID*.
GSV	String, 3 characters	-	GSV	GNSS Satellites in View.
<TotalNumSen>	Numeric	-	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 4: GNSS Numbering*.
<SatElev>	Numeric	Degree	72	Satellite elevation. Range: 00–90.
<SatAz>	Numeric	Degree	076	Satellite azimuth, with true north as the reference plane. Range: 000–359.
<SatCN0>	Numeric	dB-Hz	42	Satellite C/N_0. Range 00–99. Null when not tracking.
End of repeat block.				
<SignalID>	Numeric	-	1	GNSS signal ID. See *Table 4: GNSS Numbering*. Please note that this parameter is only available in NMEA 4.10 and above.
<Checksum>	Hexadecimal	-	*6D	Checksum.
<CR><LF>	Character	-	-	Carriage return and line feed.

Take LC76G (AA) for 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>
```

Parameter:

Field	Format	Unit	Example	Description
$	-	-	$	Each NMEA message starts with **$**.
<TalkerID>	String, 2 characters	-	GN	Talker identifier. See *Table 2: NMEA Talker ID*.
GSA	String, 3 characters	-	GSA	GNSS DOP and Active Satellites.
<Mode>	Character	-	A	Auto 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	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 4: GNSS 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 4: GNSS Numbering.* Please note that this parameter is only available in NMEA 4.10 and above.
<Checksum>	Hexadecimal	-	*34	Checksum.
<CR><LF>	Character	-	-	Carriage return and line feed.

Take LC76G (AA) for 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>
```

Parameter:

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

<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 in position fix, 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.

Take LC76G (AA) for example:

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

2.2.6. GLL

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

Type:

Output

Synopsis:

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

Parameter:

Field	Format	Unit	Example	Description
$	-	-	$	Each NMEA message starts with **$**.
<TalkerID>	String, 2 characters	-	GN	Talker identifier. See *Table 2: NMEA Talker ID*.
GLL	String, 3 characters	-	GLL	Geographic Position – Latitude/Longitude.
<Lat>	ddmm.mmmmmm	-	3149.334166	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	Latitude 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	Longitude direction:
				E = East
				W = West
				Note that this field is empty in case of an invalid value.

<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:
				V = Invalid
				A = Autonomous
				D = Differential

<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 in position fix, 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.

Take LC76G (AA) for example:

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

2.3. PAIR Messages

This chapter explains PAIR messages (proprietary NMEA messages defined by the chipset supplier). “**P**” means proprietary message, “**AIR**” means the command defined by the chipset supplier.

2.3.1. Packet Type: 001 PAIR_ACK

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

Type:

Output

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<CommandID>	Numeric	-	The type of command/packet to be acknowledged.
<Result>	Numeric	-	0 = The command has been successfully sent 1 = The command is processing. Please wait for the result. 2 = Command sending failed. 3 = The <commandID> is not supported. 4 = Command parameter error. Out of range/some parameters were lost/checksum error. 5 = The 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 a **$PAIR001** message.

Example:

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

2.3.3. Packet Type: 003 PAIR_GNSS_SUBSYS_POWER_OFF

Powers off the GNSS system, including DSP, RF, PE and Clock after you send this command. When this command is sent, CM4 will be set to the Standby mode. Send relevant command within 300 ms of pulling AP_REQ low.

Type:

Command

Synopsis:

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

Parameter:

None

Result:

Returns a **$PAIR001** message.

Example:

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

NOTE

1. If you send **$PAIR382,1*2E** before sending **$PAIR003*39**, the module can still receive any command in Standby mode. Then send **$PAIR002*38** to wake up the CM4 system and exit Standby mode.
2. If the software version that supports I2C communication, you send the **$PAIR003*39** command, the module can also receive any command in the Standby state.

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, which makes this startup method the fastest.

Type:

Command

Synopsis:

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

Parameter:

None

Result:

Returns a **$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 ephemeris 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 a **$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 a **$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 it can fix a position.

Type:

Command

Synopsis:

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

Parameter:

None

Result:

Returns a **$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 up.

Type:

Output

Synopsis:

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

Parameter:

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

Example:

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

NOTE

GNSS module automatically sends this command. Please do not send it to the GNSS module.

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 a **$PAIR001** message.

Example:

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

NOTE

If the set frequency is greater than 1 Hz, only **RMC** and **GGA** are output at the set frequency, other NMEA sentences keep output at 1 Hz.

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 a **$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 value 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 a **$PAIR001** message.

Example:

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

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

Query result message format:

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

Parameter included in the result:

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

Example:

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

2.3.13. 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>
```

Parameter:

Field	Format	Unit	Description
<Type>	Numeric	-	Type of standard NMEA sentence. -1 = Reset the output rates of all types of sentences to default values. 0 = NMEA_SEN_GGA 1 = NMEA_SEN_GLL 2 = NMEA_SEN_GSA 3 = NMEA_SEN_GSV 4 = NMEA_SEN_RMC 5 = NMEA_SEN_VTG8 = NMEA_SEN_GST
<OutputRate>	Numeric	-	Output rate setting. 0 = Disabled or not supported N = Output once every N position fix(es) Range of N: 0–20. Default value: 1.

Result:

Returns a **$PAIR001** message.

Example:

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

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_VTG8 = NMEA_SEN_GST

Result:

Returns a **$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_VTG8 = NMEA_SEN_GST

<OutputRate>	Numeric	-	Output rate setting. 0 = Disabled or not supported N = Output once every N position fix(es) Range of N: 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_Enabled>,0*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<GPS_Enabled>	Numeric	-	0 = Disable (DO NOT search for GPS satellites) 1 = Search for GPS satellites
<GLONASS_Enabled>	Numeric	-	0 = Disable (DO NOT search for GLONASS satellites) 1 = Search for GLONASS satellites
<Galileo_Enabled>	Numeric	-	0 = Disable (DO NOT search for Galileo satellites) 1 = Search for Galileo satellites
<BDS_Enabled>	Numeric	-	0 = Disable (DO NOT search for BDS satellites) 1 = Search for BDS satellites
<QZSS_Enabled>	Numeric	-	0 = Disable (DO NOT search for QZSS satellites) 1 = Search for QZSS satellites
<Reserved>	Numeric	-	Always "0"

Result:

Returns a **$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 and can be used with any of the following constellation combinations.
2. The following GNSS search modes are supported:
   - GPS only
   - GPS + QZSS
   - GPS + GLONASS
   - GPS + QZSS + GLONASS
   - GPS + Galileo
   - GPS + QZSS + Galileo
   - GPS + BDS
   - GPS + QZSS + BDS
   - GPS + GLONASS + Galileo + BDS
   - GPS + QZSS + GLONASS + BDS + 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 a **$PAIR001** message and the query result.

Query result message format:

```
$PAIR067<GPS_Enabled>,<GLONASS_Enabled>,<Galileo_Enabled>,<BDS_Enabled>,<QZSS_Ena
bled>,0*<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 speed threshold for static navigation. If the actual speed is below the threshold, the output position remains unchanged and the output speed is zero. 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	m/s	Speed threshold. Range: 0–20. Default value: 0.

Result:

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

Query result message format:

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

Parameter included in the result:

Field	Format	Unit	Description
<SpeedThreshold>	Numeric	m/s	Speed threshold. Range: 0–20. 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 a **$PAIR001** message.

Example:

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

NOTE

The satellites below the elevation mask are not used.

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

Query result message format:

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

Parameter included in the result:

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

Example:

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

2.3.21. Packet Type: 074 PAIR_COMMON_SET_AIC_ENABLE

Enables or disables the active interference cancellation (AIC) function. The module turns on the AIC function to reduce the interference of RF signals from Wi-Fi, Bluetooth and 2G, 3G, 4G and 5G networks. By measuring the C/N$_0$, it will be found that the anti-interference signal ability is strong after the AIC function is turned on.

Type:

Set

Synopsis

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

Parameter:

Field	Format	Unit	Description
<Enabled>	Numeric	-	0 = Disable 1 = Enable (default)

Result:

Returns a **$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 AIC(active interference cancellation) function.

Type:

Get

Synopsis

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

Parameter:

None

Result:

Returns a **$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	-	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,<CmdType>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<CmdType>	Numeric	-	Navigation mode. 0 = Normal Mode. For general purposes. 1 = Fitness Mode: For running and walking purposes so that the low-speed (< 5 m/s) movement will have more effect on position calculation. (default). 2 = Reserved. 3 = Reserved. 4 = Stationary Mode: For stationary applications where zero dynamics is assumed. 5 = Drone mode: Used for drone applications with equivalent dynamics range and vertical acceleration on different flight phase. (Ex. hovering, cruising, etc.). 6 = Reserved. 7 = Swimming Mode: For swimming purpose so that it smooths the trajectory and improves the accuracy of distance calculation.

Result:

Returns a **$PAIR001** message.

Example:

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

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

Query result message format:

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

Parameter included in the result:

Field	Format	Unit	Description
<CmdType>	Numeric	-	Navigation mode. 0 = Normal Mode. For general purposes. 1 = Fitness Mode: For running and walking purposes so that the low-speed (< 5 m/s) movement will have more effect on position calculation. 2 = Reserved. 3 = Reserved. 4 = Stationary Mode: For stationary applications where zero dynamics is assumed. 5 = Drone mode: Used for drone applications with equivalent dynamics range and vertical acceleration on different flight phase. (Ex. hovering, cruising, etc.) 6 = Reserved. 7 = Swimming Mode: For swimming purpose so that it smooths the trajectory and improves 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 (default) 1 = Enable

Result:

Returns a **$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 a **$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: 098 PAIR_COMMON_SET_NMEA_POS_DECIMAL_PRECISION

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

Type:

Set

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<Mode>	Numeric	-	The mode of the digits shown in the NMEA position: 0 = Latitude, Longitude: 4; Altitude: 1 1 = Latitude, Longitude: 5; Altitude: 2 2 = Latitude, Longitude: 6; Altitude: 3 (default) 3 = Latitude, Longitude: 7; Altitude: 3

Result:

Returns a **$PAIR001** message.

Example:

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

2.3.28. Packet Type: 099 PAIR_COMMON_GET_NMEA_POS_DECIMAL_PRECISION

Queries the precision of coordinates.

Type:

Get

Synopsis:

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

Parameter:

None

Result:

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

Query result message format:

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

Parameter included in the result:

Field	Format	Unit	Description
<Mode>	Numeric	-	The mode of the digits shown in the NMEA position: 0 = Latitude, Longitude: 4; Altitude: 1 1 = Latitude, Longitude: 5; Altitude: 2 2 = Latitude, Longitude: 6; Altitude: 3 3 = Latitude, Longitude: 7; Altitude: 3

Example:

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

2.3.29. Packet Type: 382 PAIR_TEST_LOCK_SYSTEM_SLEEP

Enables/disables to lock system Standby mode.The system will lock into Standby mode after sending the command. CM4 will entry Standby if application not working. You can send this command to lock / unlock Standby mode for special test senario.

Type:

Set

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<Enabled>	Numeric	-	Enable or disable the lock sleep. 0 = Disable(default) 1 = Enable

Result:

Returns a **$PAIR001** message.

Example:

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

NOTE

1. This configuration will not keep in the flash or RTC RAM. Please send this command every time after the GNSS subsystem or main power reboots.
2. Before the module receives **$PAIR003*39** and enters Standby mode, if it receives **$PAIR382,1*2E** command, the module can still receive command in Standby mode. Otherwise, other commands cannot be received. Sending **$PAIR002*38** will wake up the CM4 system and exit standby mode.

2.3.30. Packet Type: 391 PAIR_TEST_JAMMING_DETECT

Enables or disables jamming detection. And the jamming status messages will be returned when jamming detection is enabled.

Type:

Set/Get

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<CmdType>	Numeric	-	Enable or disable jamming detection. 0 = Disable jamming detection function (default) 1 = Enable jamming detection function

Result:

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

Query result message format:

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

Parameter included in the result:

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

Example:

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

NOTE

1. The module starts jamming detection once the feature is enabled.
   - If there is no jamming, **$PAIRSPF,1*52** will be reported to indicate good status (status 1).
   - If there is continuous jamming, the module status will change from 1 to 2 and finally to 3.
   1) When no fix has been completed: module status is 1 right after the jamming detection is enabled, and then changes to 2 when jamming is detected. During this process, the module keeps

attempting to get a fix; if the repair fails, the module status changes to 3 at last.

2) After a successful fix: module status is 1 right after jamming detection is enabled, and changes to 2 and 3 consecutively when jamming is detected.

2.3.31. Packet Type: 400 PAIR_DGPS_SET_MODE

Sets the source of DGPS correction data.

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) (default)

Result:

Retuns a **$PAIR001** message.

Example:

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

2.3.32. Packet Type: 401 PAIR_DGPS_GET_MODE

Queries the source of DGPS correction data.

Type:

Get

Synopsis:

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

Parameter:

None

Result:

Returns a **$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
```

2.3.33. Packet Type: 410 PAIR_SBAS_ENABLE

Enables or disables the search of SBAS satellites. 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 (default)

Result:

Returns a **$PAIR001** message.

Example:

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

NOTE

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

2.3.34. 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 a **$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
```

2.3.35. 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 output RTCM (default) 0 = Enable output RTCM3.x with message type MSM4 1 = Enable output RTCM3.x with message type MSM7

Result:

Returns a **$PAIR001** message.

Example:

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

2.3.36. Packet Type: 433 PAIR_RTCM_GET_OUTPUT_MODE

Queries RTCM output mode.

Type:

Get

Synopsis

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

Parameter:

None

Result:

Returns a **$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 = Disabled output RTCM 0 = Enabled output RTCM3.x with message type MSM4 1 = Enabled output RTCM3.x with message type MSM7

Example:

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

2.3.37. Packet Type: 490 PAIR_EASY_ENABLE

Enables or 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 (default)

Result:

Returns a **$PAIR001** message.

Example:

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

2.3.38. Packet Type: 491 PAIR_EASY_GET_STATUS

Queries the status of EASY function.

Type:

Get

Synopsis:

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

Parameter:

None

Result:

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

Query result message format:

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

Parameters included in the result:

Field	Format	Unit	Description
<Enabled>	Numeric	-	EASY function setting. 0 = Disabled 1 = Enabled
<Status>	Numeric	-	EASY function status. 0 = Not finished 1 = Finished 1-day extension 2 = Finished 2-day extension 3 = Finished 3-day extension

Example:

```
$PAIR491*36
$PAIR001,491,0*37
$PAIR491,1,0*37
```

NOTE

If EASY function is disabled, only the **<Enabled>** value will be returned after executing this command.

2.3.39. 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 a **$PAIR001** message.

Example:

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

NOTE

1. If the hardware cannot supply power to the RTC after the power supply of the module is cut off, this command needs to 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 the **$PAIR511*3F** command, send the **$PAIR002*38** command again to power on the module. While 1 Hz does not have this limitation.

2.3.40. 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 a **$PAIR001** message.

Example:

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

NOTE

1. If the hardware cannot supply power to the RTC after the power supply of the module is cut off, this command needs to 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 the **$PAIR513*3D** command, send the **$PAIR002*38** command again to power on the module. While 1 Hz does not have this limitation.

2.3.41. Packet Type: 650 PAIR_LOW_POWER_ENTRY_RTC_MODE

Shuts down all systems, including GNSS and CM4. When this command is sent, CM4 will be set to the Backup mode, in which it cannot receive any commands. CM4 can be woken up by Restore VCC power. All system resources will re-initialize after wake-up.

Type:

Set

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<Second>	Numeric	Second	Time to exit RTC-Mode. Range: 0 and 10–62208000 (2 years).

Result:

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

Example:

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

NOTE

<Second>=0: enter Backup mode without any timer.

2.3.42. Packet Type: 752 PAIR_PPS_SET_CONFIG_CMD

Sets PPS configurations.

Type:

Set

Synopsis:

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

Parameter:

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

Result:

Returns a **$PAIR001** message.

Example:

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

2.3.43. Packet Type: 864 PAIR_IO_SET_BAUDRATE

Sets the baud rate of UART interface.

Type:

Set

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<PortType>	Numeric	-	HW Port Type: 0 = UART
<PortIndex>	Numeric	-	HW Port Index: 0 = UART0
<Baudrate>	Numeric	bps	Baud rate: 9600 115200 (default) 230400 460800 921600 3000000

Result:

Returns a **$PAIR001** message.

Example:

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

NOTE

Must reboot the module after changing the port baud rate. The change will valid after reboot.

2.3.44. Packet Type: 865 PAIR_IO_GET_BAUDRATE

Gets the baud rate of UART interface.

Type:

Get

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<PortType>	Numeric	-	HW Port 0 = UART
<PortIndex>	Numeric	-	HW Port Index 0 = UART0

Result:

Returns a **$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 3000000

Example:

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

NOTE

Must reboot the module after changing the port baud rate. The change will valid after reboot.

2.3.45. Packet Type: 866 PAIR_IO_SET_FLOW_CONTROL

Sets UART flow control.

Type:

Set

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<PortType>	Numeric	-	HW Port Type 0 = UART
<PortIndex>	Numeric	-	HW Port Index 0 = UART0
<FlowControl>	Numeric	-	Flow control setting. 0 = Disable flow control (default) 1 = Enable SW flow control 2 = Enable HW flow control

Result:

Returns a **$PAIR001** message.

Example:

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

NOTE

Must reboot the module after changing the flow control type. The change will valid after reboot.

2.3.46. Packet Type: 867 PAIR_IO_GET_FLOW_CONTROL

Gets UART flow control setting.

Type:

Get

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<PortType>	Numeric	-	HW Port 0 = UART
<PortIndex>	Numeric	-	HW Port Index 0 = UART0

Result:

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

Query result message format:

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

Parameter included in the result:

Field	Format	Unit	Description
<FlowControl>	Numeric	-	Flow control setting. 0 = Disable flow control 1 = Enable SW flow control 2 = Enable HW flow control

Example:

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

```
$PAIR867,0*2F
```

3 Appendix A References

Table 3: Terms and Abbreviations

Abbreviation	Description
2D	2 Dimension
3D	3 Dimension
ACK	Acknowledgement
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
EASY	Embedded Assist System
EGNOS	European Geostationary Navigation Overlay Service
EPO	Extended Prediction Orbit
GGA	Global Positioning System Fix Data
GLL	Geographic Position – Latitude/Longitude
GLONASS	Global Navigation Satellite System (Russia)
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GSA	GNSS DOP and Active Satellites
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
NVRAM	Non-Volatile Random Access Memory
PAIR	Proprietary Protocol of MTK
PDOP	Position Dilution of Precision
PPS	Pulse Per Second
QZSS	Quasi-Zenith Satellite System
RMC	Recommended Minimum Specific GNSS Data
RMS	Root Mean Square
RTC	Real-time Clock
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 & Date

4 Appendix B GNSS Numbering

Table 4: GNSS Numbering

GNSS Type	System ID	Satellite ID	Signal ID
GPS	1	1–32	1 = L1 C/A
GLONASS	2	65–88	1 = L1
Galileo	3	1–36	7 = E1
BDS	4	1–63	1 = B1I
QZSS	5	193–199	-
SBAS	-	33–64	-