LS550G (00)

GNSS Protocol Specification

GNSS Products

Version: 1.0

Date: 2025-11-03

Status: Released

At Quectel, our aim is to provide timely and comprehensive services to our customers. If you require any assistance, please contact our headquarters:

Quectel Wireless Solutions Co., Ltd.
Building 5, Shanghai Business Park Phase III (Area B), No.1016 Tianlin Road, Minhang District, Shanghai 200233, China
Tel: +86 21 5108 6236
Email: info@quectel.com

Or our local offices. For more information, please visit:
http://www.quectel.com/support/sales.htm.

For technical support, or to report documentation errors, please visit:
http://www.quectel.com/support/technical.htm.
Or email us at: support@quectel.com.

Legal Notices

We offer information as a service to you. The provided information is based on your requirements and we make every effort to ensure its quality. You agree that you are responsible for using independent analysis and evaluation in designing intended products, and we provide reference designs for illustrative purposes only. Before using any hardware, software or service guided by this document, please read this notice carefully. Even though we employ commercially reasonable efforts to provide the best possible experience, you hereby acknowledge and agree that this document and related services hereunder are provided to you on an “as available” basis. We may revise or restate this document from time to time at our sole discretion without any prior notice to you.

Use and Disclosure Restrictions

License Agreements

Documents and information provided by us shall be kept confidential, unless specific permission is granted. They shall not be accessed or used for any purpose except as expressly provided herein.

Copyright

Our and third-party products hereunder may contain copyrighted material. Such copyrighted material shall not be copied, reproduced, distributed, merged, published, translated, or modified without prior written consent. We and the third party have exclusive rights over copyrighted material. No license shall be granted or conveyed under any patents, copyrights, trademarks, or service mark rights. To avoid ambiguities, purchasing in any form cannot be deemed as granting a license other than the normal non-exclusive, royalty-free license to use the material. We reserve the right to take legal action for noncompliance with abovementioned requirements, unauthorized use, or other illegal or malicious use of the material.

Trademarks

Except as otherwise set forth herein, nothing in this document shall be construed as conferring any rights to use any trademark, trade name or name, abbreviation, or counterfeit product thereof owned by Quectel or any third party in advertising, publicity, or other aspects.

Third-Party Rights

This document may refer to hardware, software and/or documentation owned by one or more third parties ("third-party materials"). Use of such third-party materials shall be governed by all restrictions and obligations applicable thereto.

We make no warranty or representation, either express or implied, regarding the third-party materials, including but not limited to any implied or statutory, warranties of merchantability or fitness for a particular purpose, quiet enjoyment, system integration, information accuracy, and non-infringement of any third-party intellectual property rights with regard to the licensed technology or use thereof. Nothing herein constitutes a representation or warranty by us to either develop, enhance, modify, distribute, market, sell, offer for sale, or otherwise maintain production of any our products or any other hardware, software, device, tool, information, or product. We moreover disclaim any and all warranties arising from the course of dealing or usage of trade.

Privacy Policy

To implement module functionality, certain device data are uploaded to Quectel's or third-party's servers, including carriers, chipset suppliers or customer-designated servers. Quectel, strictly abiding by the relevant laws and regulations, shall retain, use, disclose or otherwise process relevant data for the purpose of performing the service only or as permitted by applicable laws. Before data interaction with third parties, please be informed of their privacy and data security policy.

Disclaimer

a) We acknowledge no liability for any injury or damage arising from the reliance upon the information.
b) We shall bear no liability resulting from any inaccuracies or omissions, or from the use of the information contained herein.
c) While we have made every effort to ensure that the functions and features under development are free from errors, it is possible that they could contain errors, inaccuracies, and omissions. Unless otherwise provided by valid agreement, we make no warranties of any kind, either implied or express, and exclude all liability for any loss or damage suffered in connection with the use of features and functions under development, to the maximum extent permitted by law, regardless of whether such loss or damage may have been foreseeable.
d) We are not responsible for the accessibility, safety, accuracy, availability, legality, or completeness of information, advertising, commercial offers, products, services, and materials on third-party websites and third-party resources.

About the Document

Document Information	
Title	**LS550G (00) GNSS Protocol Specification**
Subtitle	GNSS Products
Document Type	GNSS Protocol Specification
Document Status	Released

Revision History

Version	Date	Description
-	2024-12-05	Creation of the document
1.0	2025-11-03	First official release

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 10
2.2.1. RMC 10
2.2.2. GGA 12
2.2.3. GSV 14
2.2.4. GSA 16
2.2.5. VTG 18
2.2.6. GLL 19
2.2.7. ZDA 21
2.2.8. GNS 22
2.2.9. GST 24
2.2.10. GRS 25
2.2.11. RLM 27
2.3. PQTM Messages 28
2.3.1. PQTMCOLD 29
2.3.2. PQTMWARM 29
2.3.3. PQTMHOT 30
2.3.4. PQTMGNSSSTART 31
2.3.5. PQTMGNSSSTOP 32
2.3.6. PQTMCFGFIXRATE 32
2.3.7. PQTMCFGMSGRATE 34
2.3.8. PQTMCFGCNST 36
2.3.9. PQTMCFGSIGNAL 38
2.3.10. PQTMCFGSBAS 40
2.3.11. PQTMPVT 41
2.3.12. PQTMDOP 43
2.3.13. PQTMVEL 44
2.3.14. PQTMLS 45
2.3.15. PQTMUTC 46
2.3.16. PQTMTIMEGPS 47
2.3.17. PQTMCFGSLAS 48
2.3.18. PQTMSLAS 49
2.3.19. PQTMSLASMT 51
2.3.20. PQTMCFGNAVMODE 52
2.3.21. PQTMPL 55

2.3.22. PQTMEPE 56
2.3.23. PQTMVER 57
2.3.24. PQTMQVER 57
2.3.25. PQTMVERNO 59
2.3.26. PQTMSN 60
2.3.27. PQTMSRR 61
2.3.28. PQTMSAVEPAR 61
2.3.29. PQTMRESTOREPAR 62
2.3.30. PQTMBKP 63
2.3.31. PQTMDEBUGON 64
2.3.32. PQTMDEBUGOFF 64
2.3.33. PQTMCFGGEOFENCE 65
2.3.34. PQTMGEOFENCESTATUS 67
2.3.35. PTQMCFGPPS 68
2.3.36. PQTMCFGUART 70
2.3.37. PQTMCFGODO 72
2.3.38. PQTMRESETODO 73
2.3.39. PQTMODO 74
2.3.40. PQTMCFGNMEADP 75
2.3.41. PQTMCFGNMEATID 76
2.3.42. PQTMCFGAIC 78
2.3.43. PQTMJAMMINGSTATUS 79
2.3.44. PTQMCFGLPM 79
2.3.45. PQTMCFGDGNSS 82
2.3.46. PQTMSAVENAVDATA 83
2.3.47. PQTMCLRNAVDATA 84

3 RTCM Protocol 85

4 Appendix A References 86
5 Appendix B GNSS (NMEA) Numbering 89
6 Appendix C Special Characters 90

Table Index

Table 1: Supported Protocols 7
Table 2: Structure of NMEA Protocol Messages 8
Table 3: NMEA Talker ID 9
Table 4: Error Codes 28
Table 5: Supported Messages 35
Table 6: Modes and Altitude Limitations 53
Table 7: Relationship Between Height and Output Status (Excluding Balloon Mode) 54
Table 8: Stationary Mode Operational Parameters 54
Table 9: Supported RTCM3 Messages 85
Table 10: Related Document 86
Table 11: Terms and Abbreviations 86
Table 12: GNSS Satellites (NMEA) Numbering 89
Table 13: Special Characters 90

1 Introduction

Quectel LS550G (00) GNSS module supports concurrent reception of GPS, GLONASS, Galileo, BDS, and QZSS constellations. Tracking of GPS L1 C/A, GLONASS L1, Galileo E1, BDS B1I and B1C, and QZSS L1 C/A frequency bands provides fast and accurate acquisition and makes this module an ideal solution 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 Quectel (**PQTM** commands). To report GNSS information, the module supports message output in NMEA 0183 standard protocol format and RTCM protocol format.

LS550G (00) supports the following protocols:

Table 1: Supported Protocols

Protocol	Type
NMEA 0183 V4.11	Output, ASCII, standard
	Input/output, ASCII, proprietary
RTCM 10403.3	Output, binary, standard

NOTE

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

2 NMEA Protocol

2.1. Structure of NMEA Protocol Messages

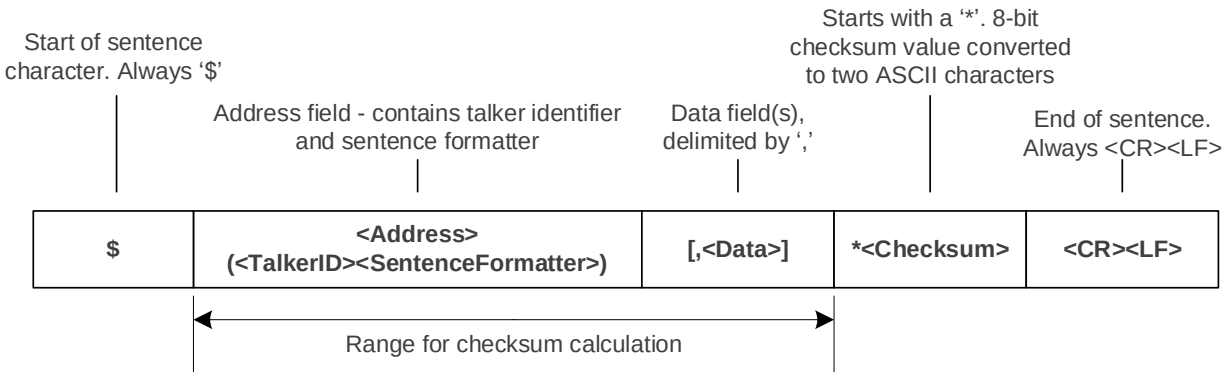

Figure 1: Structure of NMEA Protocol Messages

Table 2: Structure of NMEA Protocol Messages

Field	Description
$	Start of the sentence (Hex 0x24).
<Address>	**In Standard Messages:** In standard messages, this field consists of a two-character talker identifier (TalkerID) and a three-character sentence formatter (SentenceFormatter). The talker identifier identifies the type of talker. For more information on the Talker ID, see *Table 3: NMEA Talker ID*. The sentence formatter identifies the data type and the string format of the successive fields. **In Proprietary Messages:** In proprietary messages, this field consists of the proprietary character **P** followed by a three-character Manufacturer's Mnemonic Code, used to identify the TALKER issuing a proprietary sentence, and any additional characters as required.

Field	Description
<Data>	Data fields, delimited by the data field delimiter **‘,’**. Variable length (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 sentence (Hex 0x0D 0x0A).

Table 3: NMEA Talker ID

GNSS Constellation Configuration	Talker ID (NMEA 0183 V4.11)
GPS	GP
GLONASS	GL
Galileo	GA
BDS	GB
QZSS	GQ
Combination of Multiple Satellite Systems	GN

NMEA Checksum Sample Code:

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

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

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

```
    return result;
}
```

2.2. Standard Messages

This chapter explains the standard NMEA 0183 V4.11 messages supported by the module.

2.2.1. RMC

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

Type:

Output

Synopsis:

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

Parameter:

Field	Format	Unit	Example	Description
$	Character	-	$	Each NMEA message starts with **$**.
<TalkerID>	String, 2 characters	-	GN	Talker identifier. See *Table 3: NMEA Talker ID*.
RMC	String, 3 characters	-	RMC	Recommended Minimum Specific GNSS Data.
<UTC>	hhmmss.sss	-	020724.000	Position fix UTC. hh: Hours (00–23) mm: Minutes (00–59) ss: Seconds (00–60) sss: Decimal fraction of seconds
<Status>	Character	-	A	Positioning system status. A = Data valid. V = Navigation receiver warning.
<Lat>	ddmm.mmmmmm	-	2516.148021	Latitude. dd: Degrees (00–90) mm: Minutes (00–59) mmmmmm: Decimal fraction of

Field	Format	Unit	Example	Description
				minutes Empty if invalid.
<N/S>	Character	-	N	North-south direction. N = North S = South Empty if invalid.
<Lon>	dddmm.mmmmmm	-	11020.046222	Longitude. ddd: Degrees (000–180) mm: Minutes (00–59) mmmmmm: Decimal fraction of minutes Empty if invalid.
<E/W>	Character	-	E	East-west direction. E = East W = West Empty if invalid.
<SOG>	Numeric	Knot	0.01	Speed over ground. Variable length. Empty if invalid.
<COG>	Numeric	Degree	000.00	Course over ground. Variable length. Maximum value: 359.99. Empty if invalid.
<Date>	ddmmyy	-	191224	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. N = No fix. Satellite system not used

Field	Format	Unit	Example	Description
				for position fixing, or fix not valid.
<NavStatus>	Character	-	V	Navigational status. Not supported. Always “V”. (Navigational status not valid).
<Checksum>	Hexadecimal	-	*0C	Checksum.
<CR><LF>	Character	-	-	Carriage return and line feed.

Example:

```
$GNRMC,020724.000,A,2516.148021,N,11020.046222,E,0.01,000.00,191224,,,D,V*0C
```

2.2.2. GGA

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

Type:

Output

Synopsis:

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

Parameter:

Field	Format	Unit	Example	Description
$	Character	-	$	Each NMEA message starts with **$**.
<TalkerID>	String, 2 characters	-	GN	Talker identifier. See *Table 3: NMEA Talker ID*.
GGA	String, 3 characters	-	GGA	Global Positioning System Fix Data.
<UTC>	hhmmss.sss	-	020724.000	Position fix UTC. hh: Hours (00–23) mm: Minutes (00–59) ss: Seconds (00–60) sss: Decimal fraction of seconds
<Lat>	ddmm.mmmmmm	-	2516.148021	Latitude. dd: Degrees (00–90)

Field	Format	Unit	Example	Description
				mm: Minutes (00–59) mmmmmm: Decimal fraction of minutes Empty if invalid.
<N/S>	Character	-	N	North-south direction. N = North S = South Empty if invalid.
<Lon>	dddmm.mmmmmm	-	11020.046222	Longitude. ddd: Degrees (000–180) mm: Minutes (00–59) mmmmmm: Decimal fraction of minutes Empty if invalid.
<E/W>	Character	-	E	East-west direction. E = East W = West Empty if invalid.
<Quality>	Numeric, 1 digit	-	2	GPS quality indicator. 0 = Fix not available or invalid 1 = GPS SPS Mode, fix valid 2 = Differential GPS, SPS Mode, or Satellite Based Augmentation System (SBAS), fix valid 6 = Estimated (dead reckoning) mode.
<NumSatUsed> [1)]	Numeric, 2 digits	-	63	Number of satellites in use.
<HDOP>	Numeric	-	0.91	Horizontal dilution of precision. Empty if invalid.
<Alt>	Numeric	Meter	179.796	Altitude above mean-sea-level (geoid).
M	Character	-	M	Unit of <Alt>. “M” = Meter.
<Sep>	Numeric	Meter	-20.075	Geoidal 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). Empty if invalid.

Field	Format	Unit	Example	Description
M	Character	-	M	Unit of **<Sep>**. "M" = Meter.
<DiffAge>	-	-	-	Differential GPS data age. Not supported.
<DiffStation>	-	-	-	Differential reference station ID. Not supported.
<Checksum>	Hexadecimal	-	*51	Checksum.
<CR><LF>	Character	-	-	Carriage return and line feed.

Example:

```
$GNGGA,020724.000,2516.148021,N,11020.046222,E,2,63,0.91,179.796,M,-20.075,M,,*51
```

NOTE

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

2.2.3. GSV

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

Type:

Output

Synopsis:

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

Parameter:

Field	Format	Unit	Example	Description
$	Character	-	$	Each NMEA message starts with **$**.
<TalkerID>	String	-	GP	Talker identifier. See *Table 3: NMEA Talker ID*.
GSV	String	-	GSV	GNSS Satellites in View.
<TotalNumSen>	Numeric	-	3	Total number of sentences. Range: 1–9.
<SenNum>	Numeric	-	1	Sentence number. Range: 1– **<TotalNumSen>**.
<TotalNumSat>	Numeric	-	11	Total number of satellites in view.
Start of repeat block. Repeat times: 1–4.				
<SatID>	Numeric	-	25	Satellite ID. See *Table 12: GNSS Satellites (NMEA) Numbering*.
<SatElev>	Numeric	Degree	43	Satellite elevation. Range: 00–90. Empty if invalid.
<SatAz>	Numeric	Degree	055	Satellite azimuth, with true north as the reference plane. Range: 000–359. Empty if invalid.
<SatCN$_0$>	Numeric	dB-Hz	39	Satellite C/N$_0$. Range: 00–99. Null when not tracking.
End of repeat block.				
<SignalID>	Numeric	-	1	GNSS signal ID. See *Table 12: GNSS Satellites (NMEA) Numbering*.
<Checksum>	Hexadecimal	-	*67	Checksum.
<CR><LF>	Character	-	-	Carriage return and line feed.

Example:

```
$GPGSV,3,1,11,25,43,055,39,31,43,273,45,26,34,201,41,23,18,156,35,1*67
$GPGSV,3,2,11,21,08,275,36,29,08,112,30,12,07,038,39,10,59,175,44,1*62
$GPGSV,3,3,11,32,58,026,43,50,55,144,33,28,59,319,43,1*55
$GLGSV,2,1,07,73,66,017,40,80,12,028,28,75,10,213,28,72,15,329,,1*71
$GLGSV,2,2,07,74,62,222,45,71,55,009,32,70,49,101,37,1*4B
```

```
$GAGSV,2,1,05,19,41,328,,21,59,159,,29,56,041,29,27,46,060,31,7*76
$GAGSV,2,2,05,33,20,229,,7*4D
$GBGSV,4,1,15,25,82,311,17,08,60,189,44,39,59,354,48,09,59,298,42,1*79
$GBGSV,4,2,15,41,50,330,45,38,49,179,39,14,44,123,45,23,34,230,40,1*71
$GBGSV,4,3,15,24,33,039,43,57,31,275,,31,26,212,,06,63,326,41,1*78
$GBGSV,4,4,15,13,66,209,41,33,70,101,39,16,62,336,17,1*40
$GQGSV,1,1,04,03,12,161,20,02,48,106,34,07,55,145,31,04,63,119,30,1*65
```

NOTE

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

2.2.4. GSA

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

Type:

Output

Synopsis:

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

Parameter:

Field	Format	Unit	Example	Description
$	Character	-	$	Each NMEA message starts with **$**.
<TalkerID>	String	-	GN	Talker identifier. See *Table 3: NMEA Talker ID*.
GSA	String	-	GSA	GNSS DOP and Active Satellites.
<Mode>	Character	-	A	Selection of 2D or 3D fix. A = Automatic, allowed to automatically switch to 2D or 3D mode.
<FixMode>	Numeric	-	3	Fix mode. 1 = Fix not available 2 = 2D 3 = 3D

Field	Format	Unit	Example	Description
Start of repeat block. Repeat times: 12.				
<SatID>	Numeric	-	10	ID numbers of satellites used in solution. See *Table 12: GNSS Satellites (NMEA) Numbering*. Empty if invalid.
End of repeat block.				
<PDOP>	Numeric	-	1.16	Position dilution of precision. Maximum value: 99.99. Note that this field is 99.99 in case of an invalid value.
<HDOP>	Numeric	-	0.88	Horizontal dilution of precision. Maximum value: 99.99. Note that this field is 99.99 in case of an invalid value.
<VDOP>	Numeric	-	0.75	Vertical dilution of precision. Maximum value: 99.99. Note that this field is 99.99 in case of an invalid value.
<SystemID>	Numeric	-	1	GNSS system ID. See *Table 12: GNSS Satellites (NMEA) Numbering*.
<Checksum>	Hexadecimal	-	*04	Checksum.
<CR><LF>	Character	-	-	Carriage return and line feed.

Example:

```
$GNGSA,A,3,23,10,32,25,28,12,18,24,15,,,,0.77,0.43,0.64,1*00
$GNGSA,A,3,73,71,74,70,86,80,87,,,,,,0.77,0.43,0.64,2*08
$GNGSA,A,3,33,19,21,04,,,,,,,,,0.77,0.43,0.64,3*09
$GNGSA,A,3,16,06,08,13,23,03,09,32,02,60,59,01,0.77,0.43,0.64,4*09
$GNGSA,A,3,07,03,02,,,,,,,,,,0.77,0.43,0.64,5*06
```

NOTE

If less than 12 satellites are used for navigation, the remaining **<SatID>** fields are left empty. If more than 12 satellites are used, multiple **GSA** sentences containing all satellite IDs will be 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
$	Character	-	$	Each NMEA message starts with **$**.
<TalkerID>	String	-	GN	Talker identifier. See *Table 3: NMEA Talker ID*.
VTG	String	-	VTG	Course Over Ground & Ground Speed.
<COGT>	Numeric	Degrees	000.00	Course over ground, in true north direction. Empty if invalid
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. Empty if invalid
N	Character	-	N	Fixed field: knot.
<SOGK>	Numeric	km/h	0.02	Speed over ground in kilometers per hour. Empty if invalid.
K	Character	-	K	Fixed field: kilometers per hour
<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

Field	Format	Unit	Example	Description
				ground stations or Satellite Based Augmentation System (SBAS). E = Estimated (dead reckoning) mode. N = Data not valid.
<Checksum>	Hexadecimal	-	*25	Checksum.
<CR><LF>	Character	-	-	Carriage return and line feed.

Example:

```
$GNVTG,000.00,T,,M,0.01,N,0.02,K,D*25
```

2.2.6. GLL

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

Type:

Output

Synopsis:

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

Parameter:

Field	Format	Unit	Example	Description
$	Character	-	$	Each NMEA message starts with **$**.
<TalkerID>	String	-	GN	Talker identifier. See *Table 3: NMEA Talker ID*.
GLL	String	-	GLL	Geographic Position – Latitude/Longitude.
<Lat>	ddmm.mmmmmm	-	2516.148111	Latitude. dd: Degrees (00–90) mm: Minutes (00–59) mmmmmm: Decimal fraction of minutes Empty if invalid.
<N/S>	Character	-	N	North-south direction. N = North S = South

Field	Format	Unit	Example	Description
				Empty if invalid.
<Lon>	dddmm.mmmmmm	-	11020.045646	Longitude. ddd: Degrees (000–180) mm: Minutes (00–59) mmmmmm: Decimal fraction of minutes Empty if invalid.
<E/W>	Character	-	E	East-west direction. E = East W = West Empty if invalid.
<UTC>	hhmmss.sss	-	025321.000	Position fix UTC. hh: Hours (00–23) mm: Minutes (00–59) ss: Seconds (00–60) sss: Decimal fraction of seconds
<Status>	Character	-	A	Positioning system status. A = Data valid. V = Data not valid.
<ModeInd>	Character	-	D	Mode indicator. A = Autonomous mode. Satellite system used in non-differential mode 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. N = Data not valid.
<Checksum>	Hexadecimal	-	*4E	Checksum.
<CR><LF>	Character	-	-	Carriage return and line feed.

Example:

```
$GNGLL,2516.148111,N,11020.045646,E,025321.000,A,D*4E
```

2.2.7. ZDA

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

Type:

Output

Synopsis:

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

Parameter:

Field	Format	Unit	Example	Description
$	Character	-	$	Each NMEA message starts with **$**.
<TalkerID>	String	-	GN	Talker identifier. See *Table 3: NMEA Talker ID*.
ZDA	String	-	ZDA	Time & Date. UTC, day, month, year, and local time zone.
<UTC>	hhmmss.sss	-	030652.000	Position fix UTC. hh: Hours (00–23) mm: Minutes (00–59) ss: Seconds (00–60) sss: Decimal fraction of seconds
<Day>	Numeric	-	19	Day of month. Range: 01–31.
<Month>	Numeric	-	12	Month. Range: 01–12.
<Year>	Numeric	-	2024	Year.
<LocalHour>	Numeric	-	-	Not supported.
<LocalMin>	Numeric	-	-	Not supported.
<Checksum>	Hexadecimal	-	*45	Checksum.
<CR><LF>	Character	-	-	Carriage return and line feed.

Example:

```
$GNZDA,030652.000,19,12,2024,,*45
```

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>,<Sep>
,<DiffAge>,<DiffStation>,<NavStatus>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Example	Description
$	Character	-	$	Each NMEA message starts with **$**.
<TalkerID>	String	-	GN	Talker identifier. See *Table 3: NMEA Talker ID*.
GNS	String	-	GNS	GNSS Fix Data.
<UTC>	hhmmss.sss	-	030652.000	Position fix UTC. hh: Hours (00–23) mm: Minutes (00–59) ss: Seconds (00–60) sss: Decimal fraction of seconds
<Lat>	ddmm.mmmmmm	-	2516.148423	Latitude. dd: Degrees (00–90) mm: Minutes (00–59) mmmmmm: Decimal fraction of minutes Empty if invalid.
<N/S>	Character	-	N	North-south direction. N = North S = South Empty if invalid.
<Lon>	dddmm.mmmmmm	-	11020.045615	Longitude. ddd: Degrees (000–180) mm: Minutes (00–59) mmmmmm: Decimal fraction of minutes Empty if invalid.
<E/W>	Character	-	E	East-west direction.

Field	Format	Unit	Example	Description
				E = East W = West Empty if invalid.
<ModeInd> [1)]	Character	-	DAAAAN	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. N = No fix. Satellite system not used for position fixing, or fix not valid.
<NumSatUsed>	Numeric	-	74	Total number of satellites in use. Range: 00–99.
<HDOP>	Numeric	-	0.88	Horizontal dilution of precision. Maximum value: 99.99. Note that this field is 99.99 in case of an invalid value.
<Alt>	Numeric	Meter	172.762	Antenna altitude above mean-sea-level (geoid). Empty if invalid.
<Sep>	Numeric	Meter	-20.076	Geoidal 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). Empty if invalid.
<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).

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

Example:

```
$GNGNS,030652.000,2516.148423,N,11020.045615,E,DAAAAN,74,0.88,172.762,-20.076,,,V*09
```

NOTE

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

2.2.9. GST

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

Type:

Output

Synopsis:

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

Parameter:

Field	Format	Unit	Example	Description
$	Character	-	$	Each NMEA message starts with **$**.
<TalkerID>	String	-	GN	Talker identifier. See *Table 3: NMEA Talker ID*.
GST	String	-	GST	GNSS Pseudorange Error Statistics.
<UTC>	hhmmss.sss	-	030652.000	Position fix UTC. hh: Hours (00–23) mm: Minutes (00–59)

Field	Format	Unit	Example	Description
				ss: Seconds (00–60) sss: Decimal fraction of seconds
<STD_RMS>	Numeric	Meter	2.3	RMS value of the standard deviation of the range inputs to the navigation process. Empty if invalid.
<STD_Major>	Numeric	Meter	2.3	Standard deviation of semi-major axis of error ellipse. Empty if invalid.
<STD_Minor>	Numeric	Meter	1.8	Standard deviation of semi-minor axis of error ellipse. Empty if invalid.
<Orient>	Numeric	Degree	82.4	Orientation of semi-major axis of error ellipse. Empty if invalid.
<STD_Lat>	Numeric	Meter	1.8	Standard deviation of latitude error. Empty if invalid.
<STD_Lon>	Numeric	Meter	2.3	Standard deviation of longitude error. Empty if invalid.
<STD_Alt>	Numeric	Meter	8.8	Standard deviation of altitude error. Empty if invalid.
<Checksum>	Hexadecimal	-	*44	Checksum.
<CR><LF>	Character	-	-	Carriage return and line feed.

Example

```
//3D fix.
$GNGST,030652.000,2.3,2.3,1.8,82.4,1.8,2.3,8.8*44

//No fix.
$GNGST,114934.608,,,,,,,*53
```

2.2.10. GRS

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

Type:

Output

Synopsis:

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

Parameter:

Field	Format	Unit	Example	Description
$	Character	-	$	Each NMEA message starts with **$**.
<TalkerID>	String	-	GN	Talker identifier. See *Table 3: NMEA Talker ID*.
GRS	String	-	GRS	GNSS range residuals.
<UTC>	hhmmss.sss	-	055429.000	Position fix UTC. hh: Hours (00–23) mm: Minutes (00–59) ss: Seconds (00–60) sss: Decimal fraction of seconds
<Mode>	Numeric	-	1	Calculation method. 0 = Residuals were used to calculate the position given in the matching **GGA** or **GNS** sentence 1 = Residuals were recomputed after the **GGA** or **GNS** position was computed
Start of repeat block. Repeat times: 12.				
<Resi>	Numeric	Meter	0.2	Range residuals for satellites used in navigation. Range: -999 to 999. Empty if invalid.
End of repeat block.				
<SystemID>	Numeric	-	1	GNSS system ID. See *Table 12: GNSS Satellites (NMEA) Numbering*.
<SignalID>	Numeric	-	1	GNSS signal ID. See *Table 12: GNSS Satellites (NMEA) Numbering*.
<Checksum>	Hexadecimal	-	*66	Checksum.

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

Example:

```
$GNGRS,055429.000,1,0.2,0.6,-1.2,1.1,-0.4,-5.2,-2.8,-1.1,1.5,,,,1,1*66
$GNGRS,055429.000,1,20.2,-11.7,-9.4,14.1,22.0,-4.0,4.8,,,,,,2,1*69
$GNGRS,055429.000,1,-3.6,-1.1,0.3,5.8,-1.2,42.7,,,,,,,3,7*7F
$GNGRS,055429.000,1,-3.7,-9.2,-1.7,-12.4,0.2,1.3,-10.7,10.1,0.9,-4.1,26.5,-4.8,4,1*4E
$GNGRS,055429.000,1,,,,,,,,,,,,,5,1*6B
```

NOTE

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

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>,<Msg_Code>,<Para>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Example	Description
$	Character	-	$	Each NMEA message starts with **$**.
<TalkerID>	String	-	GA	Talker identifier. See *Table 3: NMEA Talker ID*.
RLM	String	-	RLM	Return Link Message.
<BeaconID>	Hexadecimal	-	A0434CEB630F190	Beacon ID. 15 hex characters (60 bits).

Field	Format	Unit	Example	Description
<UTC>	hhmmss.sss	-	034613.000	Position fix UTC. hh: Hours (00–23) mm: Minutes (00–59) ss: Seconds (00–60) sss: Decimal fraction of seconds
<Msg_Code>	Hexadecimal	-	F	Message code, a hex character (4 bits). Identifies the type of RLM Message Service. 0 = Reserved for future RLM services. 1 = Acknowledgement Service RLM 2 = Command Service RLM 3 = Message Service RLM 4–E = Reserved for future RLM services F = Test Service RLM (currently used only by the Galileo Program)
<Para>	Hexadecimal	-	4B02	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	-	*37	Checksum.
<CR><LF>	Character	-	-	Carriage return and line feed.

Example:

```
$GARLM,A0434CEB630F190,034613.000,F,4B02*37
```

2.3. PQTM Messages

This chapter explains the **PQTM** messages (proprietary NMEA messages defined by Quectel) supported by the LS550G (00) module.

Table 4: Error Codes

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

Field	Format	Unit	Description
			2 = Failed execution. 3 = Unsupported command.

2.3.1. PQTMCOLD

Performs a cold start, which restarts the GNSS engine without using some location information, including position and ephemeris data.

Type:

Command

Synopsis:

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

Parameter:

None

Result:

- If successful, the module is restarted and no message is sent as a reply.
- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 4: Error Codes*.

Example:

```
$PQTMCOLD*1C
```

2.3.2. PQTMWARM

Performs a warm start, which restarts the GNSS engine with the valid position, time, and almanac data. However, the ephemeris data is invalid, therefore, the receiver must download the updated ephemeris data before achieving a fix.

Type:

Command

Synopsis:

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

Parameter:

None

Result:

- If successful, the module is restarted and no message is sent as a reply.
- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 4: Error Codes*.

Example:

```
$PQTMWARM*11
```

2.3.3. PQTMHOT

Performs a hot start, which restarts the GNSS engine with the valid position, time, ephemeris, and almanac data, enabling the fastest location acquisition.

Type:

Command

Synopsis:

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

Parameter:

None

Result:

- If successful, the module is restarted and no message is sent as a reply.
- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 4: Error Codes*.

Example:

```
$PQTMHOT*4B
```

2.3.4. PQTMGNSSSTART

Starts GNSS engine.

Type:

Command

Synopsis:

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

Parameter:

None

Result:

- If successful, the module returns:

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

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 4: Error Codes*.

Example:

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

NOTE

The interval between sending **PQTMGNSSSTART** and **PQTMGNSSSTOP** should be at least 2 seconds.

2.3.5. PQTMGNSSSTOP

Stops GNSS engine.

Type:

Command

Synopsis:

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

Parameter:

None

Result:

- If successful, the module returns:

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

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 4: Error Codes*.

Example:

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

NOTE

The interval between sending **PQTMGNSSSTART** and **PQTMGNSSSTOP** should be at least 2 seconds.

2.3.6. PQTMCFGFIXRATE

Sets/gets the position fix rate.

Type:

Set/Get

Synopsis:

```
//Set:
$PQTMCFGFIXRATE,W,<Interval>*<Checksum><CR><LF>
//Get:
$PQTMCFGFIXRATE,R*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Interval>	Numeric	Millisecond	Position fix interval. The interval is fixed at 1000 ms, corresponding to a positioning frequency of 1 Hz.

Result:

- If successful, the module returns:

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

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 4: Error Codes*.

Example:

```
//Set position fix rate:
$PQTMCFGFIXRATE,W,1000*59
$PQTMCFGFIXRATE,OK*27
//Get position fix rate:
$PQTMCFGFIXRATE,R*71
$PQTMCFGFIXRATE,OK,1000*0A
```

2.3.7. PQTMCFGMSGRATE

Sets/gets message output rate on the current port.

Type:

Set/Get

Synopsis:

```
//Set:
$PQTMCFGMSGRATE,W,<MsgName >,<Rate>[,<MsgVer>]*<Checksum><CR><LF>
//Get:
$PQTMCFGMSGRATE,R,<MsgName>[,<MsgVer>]*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<MsgName>	String	-	Message name. See *Table 5: Supported Messages* for details.
<Rate>	Numeric	-	Message output rate. 0 = Output disabled. N = Output once every N position fix(es). Range of N: 0–20. See *Table 5: Supported Messages* for details.
<MsgVer>	Numeric	-	Message version. Optional. This field can be omitted when the configuration message is standard NMEA 0183 message.

Result:

- If successful, the module returns:

```
//Response to Set command:
$PQTMCFGMSGRATE,OK*<Checksum><CR><LF>
//Response to Get command:
$PQTMCFGMSGRATE,OK,<MsgName>,<Rate>[,<MsgVer>]*<Checksum><CR><LF>
```

- If successful, the module returns:

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

For details about **<ErrCode>**, see *Table 4: Error Codes*.

Example:

```
//Set the output rate of PQTMEPE to once per position fix:
$PQTMCFGMSGRATE,W,PQTMEPE,1,2*1D
$PQTMCFGMSGRATE,OK*29

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

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

//Get the output rate of GGA message:
$PQTMCFGMSGRATE,R,GGA*12
$PQTMCFGMSGRATE,OK,GGA,1*59
```

Table 5: Supported Messages

Message Name	Message Output Rate Range (N)
PQTMEPE	1–20
PQTMPVT	1–20
PQTMPL	1–20
PQTMODO	1–20
PQTMVEL	1–20
PQTMDOP	1–20
PQTMGEOFENCESTATUS	1–20
PQTMJAMMINGSTATUS	1–20
PQTMTIMEGPS	1–20
PQTMLS	1–20
PQTMUTC	1–20
PQTMSLAS	1–20
PQTMSLASMT	1–20

Message Name	Message Output Rate Range (N)
RMC	1–20
GGA	1–20
GSV	1–20
GSA	1–20
VTG	1–20
GLL	1–20
ZDA	1–20
GNS	1–20
GST	1–20
GRS	1–20
RLM	1–20

NOTE

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

2.3.8. PQTMCFGCNST

Sets/gets the constellation configuration.

Type:

Set/Get

Synopsis:

```
//Set:
$PQTMCFGCNST,W,<GPS>,<GLONASS>,<Galileo>,<BDS>,<QZSS>,<Reserved>*<Checksum><CR><LF>
```

```
//Get:
$PQTMCFGCNST,R*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<GPS>	Numeric	-	Enable/disable GPS. 0 = Disable 1 = Enable
<GLONASS>	Numeric	-	Enable/disable GLONASS. 0 = Disable 1 = Enable
<Galileo>	Numeric	-	Enable/disable Galileo. 0 = Disable 1 = Enable
<BDS>	Numeric	-	Enable/disable BDS. 0 = Disable 1 = Enable
<QZSS>	Numeric	-	Enable/disable QZSS. 0 = Disable 1 = Enable
<Reserved>	Numeric	-	Reserved.

Result:

- If successful, the module returns:

```
//Response to Set command:
$PQTMCFGCNST,OK*<Checksum><CR><LF>
//Response to Get command:
$PQTMCFGCNST,OK,<GPS>,<GLONASS>,<Galileo>,<BDS>,<QZSS>,<Reserved>*<Checksum><CR><LF>
```

- If failed, the module returns:

```
$PQTMCFGCNST,ERROR,*<Checksum><CR><LF>
```

For details about **<ErrCode>**, see *Table 4: Error Codes*.

Example:

```
//Set the constellation configuration:
$PQTMCFGCNST,W,1,1,1,1,0,0*2B
$PQTMCFGCNST,OK*78
```

```
//Get the constellation configuration:
$PQTMCFGCNST,R*2E
$PQTMCFGCNST,OK,1,1,1,1,0,0*78
```

NOTE

Supported constellation configurations:

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

2.3.9. PQTMCFGSIGNAL

Sets/gets the GNSS signal mask.

Type:

Set/Get

Synopsis:

```
//Set:
$PQTMCFGSIGNAL,W,<GPS_Sig>,<GLO_Sig>,<GAL_Sig>,<BDS_Sig>,<QZS_Sig>,<Reserved>*<Checksum><CR><LF>
//Get:
$PQTMCFGSIGNAL,R*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<GPS_Sig>	Hexadecimal	-	GPS signal mask. Bit 0 = L1 C/A

Field	Format	Unit	Description
<GLO_Sig>	Hexadecimal	-	GLONASS signal mask. Bit 0 = G1 C/A
<GAL_Sig>	Hexadecimal	-	Galileo signal mask. Bit 0 = E1
<BDS_Sig>	Hexadecimal	-	BDS signal mask. Bit 0 = B1I (can only be enabled) Bit 3 = B1C
<QZS_Sig>	Hexadecimal	-	QZSS signal mask. Bit 0 = L1 C/A
<Reserved>	Hexadecimal	-	Reserved.

Result:

- If successful, the module returns:

```
//Response to Set command:
$PQTMCFGSIGNAL,OK*<Checksum><CR><LF>
//Response to Get command:
$PQTMCFGSIGNAL,OK,<GPS_Sig>,<GLO_Sig>,<GAL_Sig>,<BDS_Sig>,<QZS_Sig>,<Reserved>*<Checksum><CR><LF>
```

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 4: Error Codes*.

Example:

```
//Set GNSS signal mask:
$PQTMCFGSIGNAL,W,01,01,01,01,01,00*3E
$PQTMCFGSIGNAL,OK*6C

//Get GNSS signal mask:
$PQTMCFGSIGNAL,R*3A
$PQTMCFGSIGNAL,OK,01,01,01,01,01,00*6D
```

NOTE

1. The current command can only configure the BDS B1C frequency band switch, other frequency bands cannot be disabled.

2. For each field, a bit value of 0 indicates disabled, and 1 indicates enabled.

2.3.10. PQTMCFGSBAS

Sets/gets the SBAS function state.

Type:

Set/Get

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<State>	Numeric	-	SBAS function state. 0 = Disabled 1 = Enabled

Result:

- If successful, the module returns:

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

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 4: Error Codes*.

Example:

```
//Disable SBAS function:
$PQTMCFGSBAS,W,0*3E
$PQTMCFGSBAS,OK*71
```

//Get the SBAS function state:
$PQTMCFGSBAS,R*27
$PQTMCFGSBAS,OK,0*6D

//Enable SBAS function:
$PQTMCFGSBAS,W,1*3F
$PQTMCFGSBAS,OK*71
//Get the SBAS function state:
$PQTMCFGSBAS,R*27
$PQTMCFGSBAS,OK,1*6C

2.3.11. PQTMPVT

Outputs the PVT (GNSS only) result.

Type:

Output

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<MsgVer>	Numeric	-	Message version. 1 = Version 1 (Always 1 for this version.)
<TOW>	Numeric	Millisecond	Time of week.
<Date>	YYYYMMDD	-	UTC date. YYYY: Year MM: Month DD: Day of month
<Time>	hhmmss.sss	-	UTC time. hh: Hours (00–23) mm: Minutes (00–59) ss: Seconds (00–60) sss: Decimal fraction of seconds
<Quality>	Numeric	-	GPS quality indicator.

Field	Format	Unit	Description
			0 = Fix not available or invalid. 1 = GPS SPS Mode, fix valid. 2 = Differential GPS, SPS Mode, or Satellite Based Augmentation System (SBAS), fix valid. 6 = Estimated (dead reckoning) mode. Note that this field is same as **<Quality>** in **GGA**.
<FixMode>	Numeric	-	Fix mode. 0 = No fix. 1 = Reserved. 2 = 2D fix. 3 = 3D fix.
<NumSatUsed>	Numeric	-	Number of satellites in use.
<LeapS>	Numeric	Second	Leap seconds. Empty if invalid.
<Lat>	Numeric	Degree	Latitude. Empty if invalid.
<Lon>	Numeric	Degree	Longitude. Empty if invalid.
<Alt>	Numeric	Meter	Altitude above mean-sea-level. Empty if invalid.
<Sep>	Numeric	Meter	Geoidal separation (the difference between the WGS84 earth ellipsoid surface and the mean-sea-level surface). Empty if invalid.
<VelN>	Numeric	m/s	North velocity. Empty if invalid.
<VelE>	Numeric	m/s	East velocity. Empty if invalid.
<VelD>	Numeric	m/s	Down velocity. Empty if invalid.
<Spd>	Numeric	m/s	Ground speed. Empty if invalid.
<Heading>	Numeric	Degree	Heading. Range: 0.00–360.00. Empty if invalid.
<HDOP>	Numeric	-	Horizontal dilution of precision. 99.99 if this field is invalid.
<PDOP>	Numeric	-	Position (3D) dilution of precision. 99.99 if this field is invalid.

Example:

```
//No fix.
$PQTMPVT,1,1000,20221225,163355.000,0,0,00,,,,,,,,,,99.99,99.99*49
```

//3D fix.
$PQTMPVT,1,68872000,20251026,190734.000,2,3,33,18,25.26921270,110.33406510,180.033,-20.076,0.004,0.001,0.001,0.004,107.01,0.57,0.93*76

2.3.12. PQTMDOP

Outputs dilution of precision.

Type:

Output

Synopsis:

$PQTMDOP,<MsgVer>,<TOW>,<GDOP>,<PDOP>,<TDOP>,<VDOP>,<HDOP>,<NDOP>,<EDOP>*<Checksum><CR><LF>

Parameter:

Field	Format	Unit	Description
<MsgVer>	Numeric	-	Message version. 1 = Version 1 (Always 1 for this message version).
<TOW>	Numeric	Millisecond	Time of week. Empty if invalid.
<GDOP>	Numeric	-	Geometric dilution of precision. 99.99 if this field is invalid.
<PDOP>	Numeric	-	Position (3D) dilution of precision. 99.99 if this field is invalid.
<TDOP>	Numeric	-	Time dilution of precision. 99.99 if this field is invalid.
<VDOP>	Numeric	-	Vertical dilution of precision. 99.99 if this field is invalid.
<HDOP>	Numeric	-	Horizontal dilution of precision. 99.99 if this field is invalid.
<NDOP>	Numeric	-	Not supported. Always null.
<EDOP>	Numeric	-	Not supported. Always null.

Example:

$PQTMDOP,1,181390000,0.83,0.73,0.40,0.54,0.49,,*75

2.3.13. PQTMVEL

Outputs the velocity information.

Type:

Output

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<MsgVer>	Numeric	-	Message version. 1 = Version 1 (Always 1 for this message version).
<Time>	hhmmss.sss	-	UTC time. hh: Hours (0–23) mm: Minutes (0–59) ss: Seconds (0–60) sss: Decimal fraction of seconds
<VelN>	Numeric	m/s	North velocity.
<VelE>	Numeric	m/s	East velocity.
<VelD>	Numeric	m/s	Down velocity.
<GrdSpd>	Numeric	m/s	2D speed.
<Spd>	Numeric	m/s	3D speed.
<Heading>	Numeric	Degree	Heading.
<GrdSpdAcc>	Numeric	m/s	Estimate 2D speed accuracy.
<SpdAcc>	Numeric	m/s	Estimate 3D speed accuracy.
<HeadingAcc>	Numeric	Degree	Estimate heading accuracy.

Example:

```
$PQTMVEL,1,020045.000,0.017,0.005,-0.029,0.018,0.034,103.467,0.373,0.397,124.439*41
```

2.3.14. PQTMLS

Outputs leap second forecast information.

Type:

Output

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<MsgVer>	Numeric	-	Message version. 1 = Version 1 (Always 1 for this version.)
<TOW>	Numeric	Millisecond	Time of week. Empty if invalid.
<LS_Ref>	Hexadecimal	-	Referenced constellation by the current leap second information. 1 = GPS 3 = Galileo 4 = BDS Empty if invalid.
<WN>	Numeric	-	UTC reference week number. Empty if invalid.
<LS>	Numeric	Second	Current number of leap seconds since the beginning of GPS time (January 6, 1980). It reflects how far ahead GPS time is compared to UTC time. Galileo has the same number of leap seconds as GPS. BDS has 14 fewer leap seconds than GPS. GLONASS follows UTC time, so no leap seconds. Empty if invalid.
<Flag>	Numeric	-	Valid marker for future occurrences of leap seconds. 0 = Invalid 1 = Available
<LSF_Ref>	Hexadecimal		Referenced constellation by the leap second forecast information. 1 = GPS 3 = Galileo 4 = BDS

Field	Format	Unit	Description
<Reserved>	-	-	Reserved. Always null.
<WNLSF>	Numeric	-	Week number of the new leap second.
<DN>	Numeric	-	Day of the week when the new leap second takes effect. GPS and Galileo: 1 to 7 from Sunday to Saturday; BDS: 0 to 6 from Sunday to Saturday.
<LSF>	Numeric	Second	Leap second count after future leap second changes.

Example:

```
$PQTMLS,1,295122000,1,2382,18,0,1,,2441,7,18*2A
```

NOTE

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

2.3.15. PQTMUTC

Reports the UTC time, date, leap second parameters and PPS information.

Type:

Output

Synopsis:

```
$PQTMUTC,<MsgVer>,<Time>,<Date>,<TOW>,<Frac>,<Acc>,<GPSLeapSVal>,<GPSLeapS>,<PPSStatus>,<Reserved>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<MsgVer>	Numeric	-	Message version. 1 = Version 1 (Always 1 for this message version.)
<Time>	hhmmss.sss	-	UTC time. hh: Hours. Range: 00–23. mm: Minutes. Range: 00–59. ss: Seconds. Range: 00–60. sss: Decimal fraction of seconds.
<Date>	ddmmyyyy	-	UTC date. yyyy: Year mm: Month dd: Day of month
<TOW>	Numeric	Millisecond	Integer part of time of week. Empty if invalid.
<Frac>	Numeric	Nanosecond	Fractional part of time of week. Empty if invalid.
<Acc>	Numeric	Nanosecond	Time accuracy estimate. Empty if invalid.
<GPSLeapSVal>	Numeric	-	GPS leap seconds validity. 0 = Invalid 2 = Valid
<GPSLeapS>	Numeric	Second	GPS leap seconds. Empty if invalid.
<PPSStatus>	Numeric	-	PPS status. 0 = Invalid 1 = Valid
<Reserved>	Numeric	-	Reserved. Always null.

Example:

```
$PQTMUTC,1,095711.000,03092025,295049000,469438,27,2,18,1,*49
```

2.3.16. PQTMTIMEGPS

Outputs the GPS time.

Type:

Output

Synopsis:

```
$PQTMTIMEGPS,<MsgVer>,<TOW>,<FTOW>,<WN>,<Date>,<Time>,<LeapS>,<Time_Acc>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<MsgVer>	Numeric	-	Message version. 1 = Version 1 (Always 1 for this message version.)
<TOW>	Numeric	Millisecond	Integer part of time of week. Empty if invalid.
<FTOW>	Numeric	Nanosecond	Fractional part of **<TOW>**. Empty if invalid.
<WN>	Numeric	-	GPS week number of the navigation. Empty if invalid epoch.
<Date>	YYYYMMDD	-	UTC date. YYYY: Year MM: Month DD: Day of month
<Time>	hhmmss.sss	-	UTC time. hh: Hours. Range: 00-23. mm: Minutes. Range: 00-59. ss: Seconds. Range: 00-60. sss: Decimal fraction of seconds.
<LeapS>	Numeric	Second	Leap seconds. Empty if invalid.
<Time_Acc>	Numeric	Nanosecond	Time accuracy estimate. Empty if invalid.

Example:

```
$PQTMTIMEGPS,1,92792000,-418074,2358,20250317,014614.000,18,42*47
```

2.3.17. PQTMCFGSLAS

Sets/gets the SLAS (Sub-meter Level Augmentation Service) state.

Type:

Set/Get

Synopsis:

```
//Set:
```

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

Parameter:

Field	Format	Unit	Description
<State>	Numeric	-	SLAS state. 0 = Disabled 1 = Enabled

Result:

- If successful, the module returns:

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

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 4: Error Codes*.

Example:

```
$PQTMCFGSLAS,W,1*31
$PQTMCFGSLAS,OK*7F
```

2.3.18. PQTMSLAS

Outputs the SLAS status data.

Type:

Output

Synopsis:

```
$PQTMSLAS,<MsgVer>,<UTC>,<GMS_ID>,<GMS_Lat>,<GMS_Lon>,<UsedSatID>,<Reserved>,<Num
PRC>{,<GNSS_ID>,<SVID>,<PRC>}*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<MsgVer>	Numeric	-	Message version. 1 = Version 1 (Always 1 for this message version.)
<UTC>	hhmmss.sss	-	UTC time. hh: Hours. Range: 00–23. mm: Minutes. Range: 00–59. ss: Seconds. Range: 00–60. sss: Decimal fraction of seconds.
<GMS_ID>	Numeric	-	Monitoring station code. Empty if invalid. Range: 0–63.
<GMS_Lat>	Numeric	Degree	Monitoring station latitude. Empty if invalid.
<GMS_Lon>	Numeric	Degree	Monitoring station longitude. Empty if invalid.
<UsedSatID>	Numeric	-	QZSS satellite ID whose correction data is used. Empty if invalid.
<Reserved>	Numeric	-	Reserved. Always null.
<NumPRC>	Numeric	-	Number of pseudorange corrections.
Start of repeat block. Repeat times: **<NumPRC>**.			
<GNSS_ID>	Numeric	-	GNSS ID. 0 = GPS 1 = GLONASS 2 = Galileo 3 = BDS 4 = QZSS
<SVID>	Numeric	-	Satellite ID.
<PRC>	Numeric	Centimeter	Pseudorange correction.
End of repeat block.			

Example:

```
$PQTMSLAS,1,080251.000,3,36.580,140.550,183,,14,4,193,-496,4,195,-2100,4,194,-524,4,196,-900,0,20,-2572,0,11,-908,0,22,-772,0,16,-2020,0,1,-964,0,14,-1160,0,6,-620,0,19,-516,0,3,-532,0,18,-2632*0D
```

2.3.19. PQTMSLASMT

Outputs the SLAS message type and data.

Type:

Output

Synopsis:

```
$PQTMSLASMT,<MsgVer>,<UTC>,<Parity>,<SatID>,<MsgType>,<DataLength>,<Data>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<MsgVer>	Numeric	-	Message version. 1 = Version 1 (Always 1 for this message version.)
<UTC>	hhmmss.sss	-	UTC time. hh: Hours. Range: 00–23. mm: Minutes. Range: 00–59. ss: Seconds. Range: 00–60. sss: Decimal fraction of seconds.
<Parity>	Numeric	-	Parity checking status. 0 = Passed 1 = Failed
<SatID>	Numeric	-	Satellite ID of the broadcast message data. Empty if invalid.
<MsgType>	Numeric	-	Message type. Empty if invalid. Range: 0–63.
<DataLength>	Numeric	Byte	Data length. Empty if invalid.
<Data>	Hexadecimal	-	Payload. Empty if invalid.

Example:

```
$PQTMSLASMT,1,075242.000,0,186,50,27,539EFFC03DC7593DFBA8BA9BD0BDA7D576379D3DFBDFBD9F9B4000*08
```

2.3.20. PQTMCFGNAVMODE

Sets/gets the navigation mode.

Type:

Set/Get

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<Mode>	Numeric	-	Navigation mode. 0 = Normal mode. 1 = Fitness mode. Used for running and walking purposes, making low-speed movement (< 5 m/s) more impactful on position calculation. 2 = Reserved. 3 = Balloon mode. Used for high-altitude balloon scenario where the vertical movement has a greater impact on the position calculation. 4 = Stationary mode. Used for stationary applications where a zero-dynamic is assumed. 5 = Dynamic flight mode. Used for drone applications with equivalent dynamic range and vertical acceleration at different flight phases (for example, hovering and cruising). 6 = Reserved. 7 = Swimming mode. Used for swimming activities to smooth the trajectory and improve the accuracy of distance calculation.

Result:

- If successful, the module returns:

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

```
$PQTMCFGNAVMODE,OK,<Mode>*<Checksum><CR><LF>
```

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 4: Error Codes*.

Example:

```
//Set navigation mode
$PQTMCFGNAVMODE,W,1*66
$PQTMCFGNAVMODE,OK*28

//Get navigation mode
$PQTMCFGNAVMODE,R*7E
$PQTMCFGNAVMODE,OK,1*35
```

NOTE

1. Each mode has its altitude limitation. Please choose the appropriate mode based on *Table 6: Modes and Altitude Limitations*. If your test scenario exceeds the limitation, the position calculation will be incorrect.
2. At heights above 80000 meters in Balloon mode, NMEA messages, RTCM messages, debug logs, and binary data are not output. The output of enabled messages resumes at heights below 80000 meters. Except for Balloon mode, other modes can be divided into three levels according to height. For details, please see *Table 7: Relationship Between Height and Output Status (Excluding Balloon Mode)*.
3. For more information about Stationary mode, see *Table 8: Stationary Mode Operational Parameters*.
4. When **<Mode>** = 4 (Stationary mode), if the module speed is at a critical value, and the following conditions are met, the longitude and latitude values will remain unchanged:
   - 60 seconds have passed since the last position fix;
   - Module's speed remains below 5 m/s for at least 3 consecutive seconds.
5. When **<Mode>** = 5 (Dynamic flight mode), the UTC time is automatically adjusted to reflect real time.

Table 6: Modes and Altitude Limitations

Mode	Altitude Limitation
Normal mode	10000 m

Fitness mode	10000 m
Balloon mode	80000 m
Stationary mode	10000 m
Dynamic flight mode	10000 m
Swimming mode	10000 m

Table 7: Relationship Between Height and Output Status (Excluding Balloon Mode)

Protocol	Type
< 10000 m	Normal position fix.
10000–50000 m	Positioning status and result cannot be guaranteed.
> 50000 m	NMEA messages, RTCM messages, debug log, and binary data will not be output.

Table 8: Stationary Mode Operational Parameters

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

2.3.21. PQTMPL

Outputs protection level information.

Type:

Output

Synopsis:

```
$PQTMPL,<MsgVer>,<TOW>,<PUL>,<Res1>,<Res2>,<PL_PosN>,<PL_PosE>,<PL_PosD>,<PL_VelN>,<PL_VelE>,<PL_VelD>,<Res3>,<Res4>,<PL_Time>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<MsgVer>	Numeric	-	Message version. 1 = Version 1 (Always 1 for this message version).
<TOW>	Numeric	Millisecond	Time of week. Null if invalid.
<PUL>	Numeric	%	Probability of uncertainty level per epoch.
<Res1>	Numeric	-	Reserved. Always 1.
<Res2>	Numeric	-	Reserved. Always 1.
<PL_PosN>	Numeric	Millimeter	Protection level of north position. Empty if invalid.
<PL_PosE>	Numeric	Millimeter	Protection level of east position. Empty if invalid.
<PL_PosD>	Numeric	Millimeter	Protection level of down position. Empty if invalid.
<PL_VelN>	Numeric	mm/s	Protection level of north velocity. Empty if invalid.
<PL_VelE>	Numeric	mm/s	Protection level of east velocity. Empty if invalid.
<PL_VelD>	Numeric	mm/s	Protection level of down velocity. Empty if invalid.
<Res3>	Numeric	-	Reserved. Always null.
<Res4>	Numeric	-	Reserved. Always null.
<PL_Time>	Numeric	Nanosecond	Protection level of time. Null if invalid.

Example:

```
//No fix:
```

$PQTMPL,1,,5.00,1,1,,,,,,,,*2E

//3D fix:
$PQTMPL,1,180455000,5.00,1,1,14946,15901,44840,246,254,214,,,44*19

2.3.22. PQTMEPE

Outputs the estimated positioning error.

Type:

Output

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<MsgVer>	Numeric	-	Message version. 2 = Version 2 (Always 2 for this version.)
<EPE_North>	Numeric	Meter	Estimated north error.
<EPE_East>	Numeric	Meter	Estimated east error.
<EPE_Down>	Numeric	Meter	Estimated down error.
<EPE_2D>	Numeric	Meter	Estimated 2D position error.
<EPE_3D>	Numeric	Meter	Estimated 3D position error.

Example:

$PQTMEPE,2,1.146,1.348,2.112,1.769,2.755*54

2.3.23. PQTMVER

Outputs the firmware version. The message is output immediately upon module power-up and is output only once.

Type:

Output

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<VerID>	Numeric	-	Version ID. 1 = Firmware version.
<DesStr>	String	-	Version description. Fixed to “MODULE”.
<VerStr>	String	-	Version string.
<BuildDate>	yyyy/mm/dd	-	Firmware build date. yyyy: Year mm: Month dd: Day of month
<BuildTime>	hh:mm:ss	-	Firmware build time. hh: Hours (00–23) mm: Minutes (00–59) ss: Seconds (00–60)

Example:

```
$PQTMVER,1,MODULE,LS550G00AANR01A03S,2025/07/31,18:23:57*30
```

2.3.24. PQTMQVER

Queries the firmware version.

Type:

Command

Synopsis:

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

Parameter:

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

Result:

- If successful, the module returns:

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

Parameters included in the result:

Field	Format	Unit	Description
<DesStr>	String	-	Version description. Fixed to “MODULE”.
<VerStr>	String	-	Version string.
<BuildDate>	yyyy/mm/dd	-	Firmware build date. yyyy: Year mm: Month dd: Day of month
<BuildTime>	hh:mm:ss	-	Firmware build time. hh: Hours (00–23) mm: Minutes (00–59) ss: Seconds (00–60)

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 4: Error Codes*.

Example:

```
$PQTMQVER*08
$PQTMQVER,OK,1,MODULE,LS550G00AANR01A01S,2024/12/05,19:11:26*4F
```

2.3.25. PQTMVERNO

Queries the firmware version information.

Type:

Command

Synopsis:

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

Parameter:

None

Result:

- If successful, the module returns:

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

Parameters included in the result:

Field	Format	Unit	Description
<VerStr>	String	-	Version string.
<BuildDate>	yyyy/mm/dd	-	Firmware build date. yyyy: Year mm: Month dd: Day of month
<BuildTime>	hh:mm:ss	-	Firmware build time. hh: Hours (00–23) mm: Minutes (00–59) ss: Seconds (00–60)

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 4: Error Codes*.

Example:

```
$PQTMVERNO*58
$PQTMVERNO,LS550G00AANR01A01S,2024/12/05,19:11:26*1C
```

2.3.26. PQTMSN

Gets the module SN code.

Type:

Command

Synopsis:

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

Parameter:

None

Result:

- If successful, the module returns:

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

Parameter:

Field	Format	Unit	Description
<Reserved>	Numeric	-	Reserved. Always 1.
<Length>	Numeric	-	SN length. Unit: byte.
<SN>	String	-	Module SN code.

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 4: Error Codes*.

Example:

```
$PQTMSN*05
$PQTMSN,OK,1,16,ESG5241364AUIDE5*3C
```

2.3.27. PQTMSRR

Performs a system reset and reboots the receiver.

Type:

Command

Synopsis:

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

Parameter:

None

Result:

- If successful, the receiver will be restarted and no message will be sent as a reply.
- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 4: Error Codes*.

Example:

```
$PQTMSRR*4B
```

2.3.28. PQTMSAVEPAR

Saves configurations to NVM.

Type:

Command

Synopsis:

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

Parameter:

None

Result:

- If successful, the module returns:

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

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 4: Error Codes*.

Example:

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

2.3.29. PQTMRESTOREPAR

Restores all parameters to default values.

Type:

Command

Synopsis:

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

Parameter:

None

Result:

- If successful, the module returns:

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

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 4: Error Codes*.

Example:

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

2.3.30. PQTMBKP

Shuts down all systems at once to save power (that is, sets the module to the Backup mode). In Backup mode, the module cannot execute **PQTM** commands and the GNSS engine does not provide position-related information.

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

Type:

Command

Synopsis:

```
$PQTMBKP[,<Second>]*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Second>	Numeric	Second	Timer for exiting Backup mode. Range: 0 and 10–62208000 (about 2 years). Use 0 or omit to disable timer.

Result:

- If successful, the module does not return any response. It either wakes up after the specified time (**<Second>** ≥ 10), or remains in Backup mode if no timer is set.
- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 4: Error Codes*.

Example:

```
//Enter Backup mode without any response and wake up after 66 seconds:
$PQTMBKP,66*6D

//Enter Backup mode without any response and without any timer:
$PQTMBKP,0*5D
```

2.3.31. PQTMDEBUGON

Enables debug log output.

Type:

Command

Synopsis:

```
$PQTMDEBUGON[,<Mask>]*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Mask>	Hexadecimal	-	Debug log mask. Bit value: 0 = Disabled, 1 = Enabled. Bit 0 = Full debug information. Bit 1 = Lite debug information. Bit 4 = Block information. If this field is omitted, the value of **<Mask>** is 0x11.

Result:

- If successful, the module returns:

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

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 4: Error Codes*.

Example:

```
//Enable full debug log output and block information output.
$PQTMDEBUGON*48
$PQTMDEBUGON,OK*60
```

2.3.32. PQTMDEBUGOFF

Disables debug log output.

Type:

Command

Synopsis:

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

Parameter:

None

Result:

- If successful, the module returns:

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

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 4: Error Codes*.

Example:

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

2.3.33. PQTMCFGGEOFENCE

Sets/gets the geofence feature.

Type:

Set/Get

Synopsis:

```
//Set:
$PQTMCFGGEOFENCE,W,<Index>,<Status>[,<Reserved>,<Shape>,<Lat0>,<Lon0>,<Lat1/Radius>,<Lon1>,<Lat2>,<Lon2>,<Lat3>,<Lon3>]*<Checksum><CR><LF>
//Get:
$PQTMCFGGEOFENCE,R,<Index>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Index>	Numeric	-	Geofence index. Range: 0–3.

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

Result:

- If successful, the module returns:

```
//Response to Set command:
$PQTMCFGGEOFENCE,OK*<Checksum><CR><LF>
//Response to Get command:
$PQTMCFGGEOFENCE,OK,<Index>,<Status>[,<Reserved>,<Shape>,<Lat0>,<Lon0>,<Lat1/Radius>,<Lon1>,<Lat2>,<Lon2>,<Lat3>,<Lon3>]*<Checksum><CR><LF>
```

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 4: Error Codes*.

Example:

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

//Get the configuration of the enabled geofence 0:
$PQTMCFGGEOFENCE,R,0*3E
$PQTMCFGGEOFENCE,OK,0,1,0,0,31.451248,117.451245,100.5*4B

//Disable geofence 0:
$PQTMCFGGEOFENCE,W,0,0*27
$PQTMCFGGEOFENCE,OK*74

//Get the configuration of the disabled geofence 0:
$PQTMCFGGEOFENCE,R,0*3E
$PQTMCFGGEOFENCE,OK,0,0*74
```

NOTE

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

2.3.34. PQTMGEOFENCESTATUS

Outputs the geofence status.

Type:

Output

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<MsgVer>	Numeric	-	Message version. 1 = Version 1 (Always 1 for this version.)
<Time>	hhmmss.sss	-	UTC time. hh: Hours. Range: 00–23. mm: Minutes. Range: 00–59. ss: Seconds. Range: 00–60. sss: Decimal fraction of seconds.
Start of repeat block. Repeat times: 4.			
<StateN> 1)	Numeric	-	Geofence N state. Range of N: 0–3. 0 = Unknown. 1 = Inside. 2 = Outside.
End of repeat block.			

Example:

```
$PQTMGEOFENCESTATUS,1,124521.000,1,2,2,2*27
```

NOTE

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

- Geofence function is disabled;
- Module's position is not fixed.

2.3.35. PTQMCFGPPS

Sets/gets the 1PPS feature configuration.

Type:

Set/Get

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<Index>	Numeric	-	PPS index. Always 1. 1 = 1PPS
<Enable>	Numeric	-	Enable/disable PPS output. 0 = Disable 1 = Enable
<Duration>	Numeric	Millisecond	Pulse duration. Range: 1–999. Default value: 100.
<Mode>	Numeric	-	Fix mode for PPS output. 1 = Always 2 = 2D fix 3 = 3D fix 4 = After the first fix
<Polarity>	Numeric	-	Pulse polarity. 0 = Low 1 = High
<Interval>	Numeric	-	PPS generated interval. 0 = Every second 1 = Odd second 2 = Even second Note that the odd second and even second in **<Interval>** can only be configured successfully when **<Mode>** is set to Always and 3D.

Result:

- If successful, the module returns:

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

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 4: Error Codes*.

Example:

```
//Set the 1PPS feature configuration:
$PQTMCFGPPS,W,1,1,100,1,1,0*73
$PQTMCFGPPS,OK*21
```

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

NOTE

When **<Enable>** is set to 0 (disable), the fields after **<Enable>** must be omitted.

2.3.36. PQTMCFGUART

Sets/gets the UART port.

Type:

Set/Get

Synopsis:

```
//Set the current UART port:
$PQTMCFGUART,W,<BaudRate>[,<DataBit>,<Parity>,<StopBit>,<FlowCtrl>]*<Checksum><CR><LF>
//Set a specified UART port:
$PQTMCFGUART,W,<Index>,<BaudRate>[,<DataBit>,<Parity>,<StopBit>,<FlowCtrl>]*<Checksum><CR><LF>
//Get the configuration of the current UART port or a specified UART port:
$PQTMCFGUART,R[,<Index>]*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Index>	Numeric	-	UART port index. 1 = UART1
<BaudRate>	Numeric	bps	UART baud rate. Supported values: 4800 9600 19200 38400 57600 115200 230400 460800 921600

Field	Format	Unit	Description
<DataBit>	Numeric	bit	UART data bits. 7 = 7 bits 8 = 8 bits
<Parity>	Numeric	-	Parity. 0 = No parity 1 = Odd parity 2 = Even parity
<StopBit>	Numeric	bit	Stop bit(s). 1 = 1 stop bit 2 = 2 stop bits
<FlowCtrl>	Numeric	-	Flow control. 0 = None 4 = Software flow control

Result:

- If successful, the module returns:

```
//Response to Set command:
$PQTMCFGUART,OK*<Checksum><CR><LF>
//Response to Get command:
$PQTMCFGUART,OK,<Index>,<BaudRate>,<DataBit>,<Parity>,<StopBit>,<FlowCtrl>*<Checksum><CR><LF>
```

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 4: Error Codes*.

Example:

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

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

//Set all parameters on the current UART port:
$PQTMCFGUART,W,115200,8,0,1,0*11
```

$PQTMCFGUART,OK*60

//Set all parameters on UART port 1:
$PQTMCFGUART,W,1,115200,8,0,1,0*0C
$PQTMCFGUART,OK*60

//Get the configuration of the current UART port:
$PQTMCFGUART,R*36
$PQTMCFGUART,OK,1,115200,8,0,1,0*5F

//Get the configuration of UART port 1.
$PQTMCFGUART,R,1*2B
$PQTMCFGUART,OK,1,115200,8,0,1,0*5F

2.3.37. PQTMCFGODO

Sets/gets the odometer feature configuration.

Type:

Set/Get

Synopsis:

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

Parameter:

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

Result:

- If successful, the module returns:

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

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

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 4: Error Codes*.

Example:

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

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

2.3.38. PQTMRESETODO

Resets the accumulated distance recorded by the odometer.

Type:

Command

Synopsis:

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

Parameter:

None

Result:

- If successful, the module returns:

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

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 4: Error Codes*.

Example:

```
$PQTMRESETODO*09
$PQTMRESETODO,OK*21
```

NOTE

To reset the accumulated distance recorded by the odometer, you can either use **PQTMRESETODO** or power off the module. Disabling the odometer feature with **PQTMCFGODO** while the module is still working will stop distance calculation, but it cannot reset the distance to zero.

2.3.39. PQTMODO

Outputs the odometer information.

Type:

Output

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<MsgVer>	Numeric	-	Message version. 1 = Version 1 (Always 1 for this version.)
<Time>	hhmmss.sss	-	UTC time. hh: Hour (00–23) mm: Minute (00–59) ss: Second (00–60) sss: Decimal fraction of second
<State>	Numeric	-	Odometer status. 0 = Disabled 1 = Enabled
<Dist>	Numeric	Meter	Distance since last reset.

Example:

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

NOTE

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

2.3.40. PQTMCFGNMEADP

Sets/gets the decimal places of NMEA messages.

Type:

Set/Get

Synopsis:

```
//Set:
$PQTMCFGNMEADP,W,<UTC_DP>,<POS_DP>,<ALT_DP>,<DOP_DP>,<SPD_DP>,<COG_DP>*<Checksum><CR><LF>
//Get:
$PQTMCFGNMEADP,R*<Checksum><CR><LF>
```

Parameter:

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

Field	Format	Unit	Description
			0 = No fractional part.
<COG_DP>	Numeric	-	Number of decimal places for COG in standard NMEA messages. Range: 0–3. Default value: 2. 0 = No fractional part.

Result:

- If successful, the module returns:

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

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 4: Error Codes*.

Example:

```
//Set decimal places of NMEA messages:
$PQTMCFGNMEADP,W,3,6,1,2,3,2*35
$PQTMCFGNMEADP,OK*61

//Get decimal places of NMEA messages:
$PQTMCFGNMEADP,R*37
$PQTMCFGNMEADP,OK,3,6,1,2,3,2*66
```

2.3.41. PQTMCFGNMEATID

Sets/gets the NMEA Talker ID.

Type:

Set/Get

Synopsis:

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

```
//Get:
$PQTMCFGNMEATID,R*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Main_TalkerID>	Character	-	Main Talker ID for all standard NMEA messages except **GSV**. 00 = Auto mode. The main talker ID is determined by the GNSS constellation configuration. If it is not “00”, specify a two-character talker ID. Note that ‘*’ character cannot be used.
<GSV_TalkerID>	Numeric	-	**GSV** Talker ID. 0 = Determined by the GNSS constellation configuration 1 = Same value as **<Main_TalkerID>**

Result:

- If successful, the module returns:

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

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 4: Error Codes*.

Example:

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

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

$PQTMCFGNMEATID,OK,00,0*1C

2.3.42. PQTMCFGAIC

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

Type:

Set/Get

Synopsis:

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

Parameter:

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

Result:

- If successful, the module returns:

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

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 4: Error Codes*.

Example:

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

```
//Get AIC function state:
$PQTMCFGAIC,R*6F
$PQTMCFGAIC,OK,1*2
```

2.3.43. PQTMJAMMINGSTATUS

Reports the jamming detection status.

Type:

Output

Synopsis:

```
$PQTMJAMMINGSTATUS,<MsgVer>,<Status>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<MsgVer>	Numeric	-	Message version. 1 = Version 1 (Always 1 for this message version.)
<Status>	Numeric	-	Jamming detection status. 0 = Unknown 1 = No jamming, healthy status 2 = Warning status 3 = Critical status

Example:

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

NOTE

When LS550G (00) detects Warning or Critical jamming status, the JAM_IND pin outputs low level. For more information, see *document [1] hardware design*.

2.3.44. PTQMCFGLPM

Sets/gets low power mode configuration.

Type:

Set/Get

Synopsis:

```
//Set:
$PQTMCFGLPM,W,<Type>[,<Mode>[,<FirstRun>,<FirstSleep>,<SecondRun>,<SecondSleep>]]*<Checksum><CR><LF>
//Get:
$PQTMCFGLPM,R*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Type>	Numeric	-	Low power mode type. 0 = Normal 1 = ALP 2 = FLP 3 = GLP 4 = Periodic
<Mode>	Numeric	-	If **<Type>** = ALP: 1 = ALP mode 1 2 = ALP mode 2 If **<Type>** = Periodic: 1 = Smart mode 2 = Strict mode If **<Type>** is in other modes, this field and subsequent fields must be omitted.
<FirstRun>	Numeric	Second	Running time. Range: 3–518400. This parameter is used in an environment with good GNSS signals. If **<Type>** is not set to Periodic mode, this field must be omitted.
<FirstSleep>	Numeric	Second	Sleeping time. Range: 3–518400. This parameter is used in an environment with good GNSS signals. If **<Type>** is not set to Periodic mode, this field must be omitted.
<SecondRun>	Numeric	Second	Second running time. Range: 3–518400. This parameter is used in an environment where the module has no GNSS signal. If **<Type>** is not set to Periodic mode, this field must be omitted.
<SecondSleep>	Numeric	Second	Second sleeping time. Range: 3–518400. This parameter is used in an environment where the module has no GNSS signal. If **<Type>** is not set to Periodic mode, this field

Field	Format	Unit	Description
			must be omitted.

Result:

- If successful, the module returns:

```
//Response to Set command:
$PQTMCFGLPM,OK*<Checksum><CR><LF>
//Response to Get command:
$PQTMCFGLPM,OK,<Type>[,<Mode>[,<FirstRun>,<FirstSleep>,<SecondRun>,<SecondSleep>]]*<Checksum><CR><LF>
```

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 4: Error Codes*.

Example:

```
//Set the module to periodic smart mode
$PQTMCFGLPM,W,4,1,20,30,50,70*76
$PQTMCFGLPM,OK*23

//Get low power mode configuration
$PQTMCFGLPM,R*75
$PQTMCFGLPM,OK,4,1,20,30,50,70*25
```

NOTE

1. Requirements for entering ALP mode:
   1) Fix rate: 1 Hz.
   2) Navigation mode: Normal mode.
2. ALP mode consists of **ALP mode 1** and **ALP mode 2**:
   - **ALP mode 1** maintains lower power consumption while the performance is reduced compared to previous versions.
   - **ALP mode 2** achieves relatively good performance while keeping the power consumption low.
3. Requirements for entering GLP mode:
   1) Fix rate: 1 Hz.
   2) Navigation mode: Fitness mode.
   3) Satellite constellation configuration: GPS only or GPS + QZSS.
4. Requirements for entering FLP mode:
   1) Fix rate: 1 Hz.
   2) Navigation mode: Fitness mode.

3) Satellite constellation configuration:
   - GPS + GLONASS + Galileo + BDS (+ QZSS)
   - GPS + GLONASS (+ QZSS)
   - GPS + BDS (+ QZSS)

5. In periodic mode, **<FirstRun>** must be less than or equal to **<SecondRun>** time.
6. **<FirstRun>**: Interval in seconds after exiting Sleep mode and getting a new position fix.
7. **<FirstSleep>**: Duration of Sleep mode after getting a fix (or attempting to get a fix).
8. **<SecondRun>**: GNSS module will use “second running time” instead of “first running time” setting when there is no signal. The second running time can be “0” only when the second sleeping time is “0”.
9. **<SecondSleep>**: GNSS module will use “second sleeping time” instead of “first sleeping time” setting when there is no signal. The second sleeping time can be “0” only when the second running time is “0”.

2.3.45. PQTMCFGDGNSS

Sets/gets the differential data source for DGNSS.

Type:

Set/Get

Synopsis:

```
//Set
$PQTMCFGDGNSS,W,<Source>*<Checksum><CR><LF>
//Get
$PQTMCFGDGNSS,R*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Source >	Numeric	-	Differential data source. 0 = No differential data source 1 = SBAS 2 = SLAS

Result:

- If successful, the module returns:

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

```
$PQTMCFGDGNSS,OK,<Source>*<Checksum><CR><LF>
```

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 4: Error Codes*.

Example:

```
//Set the differential data source for DGNSS:
$PQTMCFGDGNSS,W,1*71
$PQTMCFGDGNSS,OK*3F

//Get the differential data source for DGNSS:
$PQTMCFGDGNSS,R*69
$PQTMCFGDGNSS,OK,1*22
```

2.3.46. PQTMSAVENAVDATA

Saves navigation data to flash memory.

Type:

Command

Synopsis:

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

Parameter:

None

Result:

- If successful, the module returns:

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

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 4: Error Codes*.

Example:

```
$PQTMSAVENAVDATA*50
$PQTMSAVENAVDATA,OK*78
```

2.3.47. PQTMCLRNAVDATA

Erases navigation data stored in flash memory.

Type:

Command

Synopsis:

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

Parameter:

None

Result:

- If successful, the module returns:

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

- If failed, the module returns:

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

For details about **<ErrCode>**, see *Table 4: Error Codes*.

Example:

```
$PQTMCLRNAVDATA*0C
$PQTMCLRNAVDATA,OK*24
```

3 RTCM Protocol

The LS550G (00) GNSS module supports the RTCM protocol that is in accordance with *RTCM Standard 10403.3 Differential GNSS (Global Navigation Satellite Systems) Services - Version 3*. This protocol is used for transferring GNSS raw measurement data and is available from https://www.rtcm.org/.

Table 9: 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

4 Appendix A References

Table 10: Related Document

Document Name
[1] Quectel_LS550G(00)_Hardware_Design

Table 11: Terms and Abbreviations

Abbreviation	Description
2D	Two-dimensional
3D	Three-dimensional
AIC	Active Interference Cancellation
ALP	Adaptive Low Power
ASCII	American Standard Code for Information Interchange
BDS	BeiDou Navigation Satellite System
C/N_0	Carrier-to-Noise-Density Ratio
COG	Course over Ground
COGM	Course over Ground (in Magnetic North Course Direction)
COGT	Course over Ground (in True North Course Direction)
DOP	Dilution of Precision
FLP	Fitness Low Power
Galileo	Galileo Satellite Navigation System (EU)
GLONASS	Global Navigation Satellite System (Russian)
GLP	GPS Low Power

Abbreviation	Description
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
HDOP	Horizontal Dilution of Precision
ID	Identifier
LS	Leap Second
LSF	Leap Second Forecast
NavIC	Indian Regional Navigation Satellite System
NMEA	NMEA (National Marine Electronics Association) 0183 Interface Standard
NVM	Non-Volatile Memory
PDOP	Position Dilution of Precision
PPS	Pulse Per Second
PRC	Pseudorange Correction
PVT	Position, Velocity, and Time
QZSS	Quasi-Zenith Satellite System
RAIM	Receiver Autonomous Integrity Monitoring
RTC	Real-time Clock
RTK	Real-Time Kinematic
SAR	Search and Rescue
SBAS	Satellite-Based Augmentation System
SLAS	Sub-meter Level Augmentation Service
SN	Serial Number
SNR	Signal-to-Noise Ratio
SOG	Speed over Ground
SOGK	Speed over Ground in Kilometers per Hour

Abbreviation	Description
SOGN	Speed over Ground in Knots
SPS	Standard Positioning Service
SV	Satellites in View
TOW	Time of Week
UART	Universal Asynchronous Receiver/Transmitter
UTC	Coordinated Universal Time
VDOP	Vertical Dilution of Precision
WN	Week Number
WGS84	World Geodetic System 1984

5 Appendix B GNSS (NMEA) Numbering

Table 12: GNSS Satellites (NMEA) Numbering

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

NOTE

The table above is only applicable to standard NMEA messages.

6 Appendix C Special Characters

Table 13: 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.