

EG915Q Series&EG916Q-GL GNSS Application Note

LTE Standard Module Series

Version: 1.0

Date: 2024-11-14

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.

Copyright © Quectel Wireless Solutions Co., Ltd. 2024. All rights reserved.

About the Document

Revision History

Version	Date	Author	Description
-	2023-10-27	Blake GAN/ Byron ZHOU	Creation of the document
1.0	2024-11-14	Byron ZHOU	First official release

Contents

About the Document.....	3
Contents	4
Table Index.....	7
1 Introduction	8
1.1. All-in-One Solution and Stand-Alone Solution.....	8
1.2. Procedures for Enabling /Disabling GNSS Function	9
1.3. Supported NMEA Sentence Types.....	10
1.4. EPO Assistance Introduction	10
2 Description of GNSS AT Commands.....	13
2.1. AT Command Introduction.....	13
2.1.1. Definitions.....	13
2.1.2. AT Command Syntax	13
2.2. Declaration of AT Command Examples	14
2.3. AT Command Description.....	14
2.3.1. AT+QGPSCFG Configure GNSS	14
2.3.1.1. AT+QGPSCFG="outport" Configure Output Port of NMEA Sentences	15
2.3.1.2. AT+QGPSCFG="nmeasrc" Enable/Disable Acquisition of NMEA Sentences through AT+QGPSGNMEA.....	16
2.3.1.3. AT+QGPSCFG="gpsnmeatype" Configure Output Type of GPS or GNSS Multi-constellation NMEA Sentences.....	17
2.3.1.4. AT+QGPSCFG="glonassnmeatype" Configure Output Type of GLONASS or GNSS Multi-constellation NMEA Sentences.....	18
2.3.1.5. AT+QGPSCFG="galileonmeatype" Configure Output Type of Galileo NMEA Sentences.....	19
2.3.1.6. AT+QGPSCFG="beidoumeatype" Configure Output Type of BDS NMEA Sentences.....	20
2.3.1.7. AT+QGPSCFG="gnssconfig" Configure Supported GNSS Constellations.....	21
2.3.1.8. AT+QGPSCFG="autogps" Enable/Disable Automatic Running of GNSS	22
2.3.1.9. AT+QGPSCFG="gnssota" Configure URL for Downloading GNSS Firmware Package.....	22
2.3.2. AT+QGPSOTA GNSS Firmware Upgrade	23
2.3.3. AT+QGPSDEL Delete Assistance Data	24
2.3.4. AT+QGPS Enable or Disable GNSS Feature.....	25
2.3.5. AT+QGPSEND Disable GNSS Function	26
2.3.6. AT+QGPSLOC Acquire Positioning Information	27
2.3.7. AT+QAGPS Enable/Disable EPO Assistance	29
2.3.8. AT+QAGPSTIME Inject EPO Time.....	30
2.3.9. AT+QAGPSDATA Inject EPO Data File	31
2.3.10. AT+QAGPSAID Inject EPO Data File Directly.....	32
2.3.11. AT+QGNSSCMD Send Commands to GNSS Engine	33

2.3.12.	AT+QAGPSLOC Send Reference Location to GNSS Engine	34
2.3.13.	AT+QAGPSCFG Set URL for Downloading EPO Files.....	35
2.3.14.	AT+QGPSGNMEA Acquire NMEA Sentences.....	36
2.4.	AT Command Examples	39
2.4.1.	Enable/Disable GNSS.....	39
2.4.2.	Send Reference Location to GNSS Engine.....	39
2.4.3.	Send Commands to GNSS Engine.....	39
2.4.4.	Set URL for Downloading EPO Data Files	40
2.4.5.	Inject EPO Data File Directly	40
2.4.6.	Operation Procedures for EPO Assistance	41
2.5.	Summary of Error Codes	42
3	PAIR Message	43
3.1.	PAIR001 PAIR_ACK.....	43
3.2.	PAIR068 PAIR_COMMON_SET_HDOP_THRESHOLD	44
3.3.	PAIR069 PAIR_COMMON_GET_HDOP_THRESHOLD.....	45
3.4.	PAIR074 PAIR_COMMON_SET_AIC_ENABLE.....	46
3.5.	PAIR075 PAIR_COMMON_GET_AIC_STATUS	46
3.6.	PAIR080 PAIR_COMMON_SET_NAVIGATION_MODE.....	47
3.7.	PAIR081 PAIR_COMMON_GET_NAVIGATION_MODE	48
3.8.	PAIR391 PAIR_TEST_JAMMING_DETECT.....	49
3.9.	PAIR400 PAIR_DGPS_SET_MODE	51
3.10.	PAIR401 PAIR_DGPS_GET_MODE.....	51
3.11.	PAIR410 PAIR_SBAS_ENABLE	52
3.12.	PAIR411 PAIR_SBAS_GET_STATUS.....	53
3.13.	PAIR412 PAIR_SBAS_GET_SAT_INFO	54
3.14.	PAIR420 PAIR_SLAS_ENABLE.....	55
3.15.	PAIR421 PAIR_SLAS_GET_STATUS	55
3.16.	PAIR430 PAIR_RTCM_SET_INPUT_VERSION	56
3.17.	PAIR431 PAIR_RTCM_GET_INPUT_VERSION.....	57
3.18.	PAIR432 PAIR_RTCM_SET_OUTPUT_MODE.....	58
3.19.	PAIR433 PAIR_RTCM_GET_OUTPUT_MODE	59
3.20.	PAIR434 PAIR_RTCM_SET_OUTPUT_ANT_PNT	59
3.21.	PAIR435 PAIR_RTCM_GET_OUTPUT_ANT_PNT.....	60
3.22.	PAIR436 PAIR_RTCM_SET_OUTPUT_EPHEMERIS	61
3.23.	PAIR437 PAIR_RTCM_GET_OUTPUT_EPHEMERIS.....	62
3.24.	PAIR490 PAIR_EASY_ENABLE	63
3.25.	PAIR491 PAIR_EASY_GET_STATUS.....	63
3.26.	PAIR530 PAIR_EPH_GET_STATUS.....	64
3.27.	PAIR531 PAIR_EPH_SET_DATA	66
3.28.	PAIR532 PAIR_EPH_GET_DATA.....	67
3.29.	PAIR533 PAIR_EPH_CLEAR.....	69
3.30.	PAIR534 PAIR_EPH_NOTIFY_ENABLE	69
3.31.	PAIR535 PAIR_EPH_NOTIFY	70
3.32.	PAIR650 PAIR_LOW_POWER_ENTRY_RTC_MODE	71

3.33.	PAIR680 PAIR_GLP_ENABLE.....	72
3.34.	PAIR681 PAIR_GLP_GET_STATUS.....	73
3.35.	PAIR690 PAIR_PERIODIC_SET_MODE.....	73
3.36.	PAIR691 PAIR_PERIODIC_GET_MODE	75
3.37.	PAIR720 PAIR_DTM_ENABLE	76
3.38.	PAIR721 PAIR_DTM_GET_STATUS.....	76
3.39.	PAIR730 PAIR_FLP_ENABLE	77
3.40.	PAIR731 PAIR_FLP_GET_STATUS.....	78
3.41.	PAIR732 PAIR_ALP_ENABLE	79
3.42.	PAIR733 PAIR_ALP_GET_STATUS.....	80
3.43.	PAIR752 PAIR_PPS_SET_CONFIG_CMD	81
3.44.	PAIR890 PAIR_GEOFENCE_SET_CONFIG.....	82
3.45.	PAIR891 PAIR_GEOFENCE_GET_CONFIG	84
3.46.	PAIRSPF	85
4	PQ Message.....	87
4.1.	PQBAUD	87
4.2.	PQEPE	89
4.3.	PQECEF.....	90
4.4.	PQODO.....	92
4.5.	PQPZ90.....	94
4.6.	PQGLP	96
4.7.	PQVEL	97
4.8.	PQJAM	99
4.9.	PQRLM.....	101
4.10.	PQGEO	103
5	RTCM Protocol	107
6	Appendix References	109

Table Index

Table 1: Differences Between All-in-One Solution and Stand-Alone Solution	9
Table 2: Types of AT Commands	13
Table 3: Summary of Error Codes	42
Table 4: Supported RTCM3 Messages.....	107
Table 5: Related Documents.....	109
Table 6: Terms and Abbreviations	109

1 Introduction

Quectel LTE Standard EG915Q series and EG916Q-GL modules integrate the multi-GNSS engine which supports GPS, BDS, Galileo, GLONASS and QZSS constellations and EPO (Extended Prediction Orbit) assistance technology. The high-performance GNSS engine, which is suitable for low-cost and precise positioning applications and location tracking without network assistance, makes EG915Q series and EG916Q-GL modules widely used in fields such as turn-by-turn navigation, asset tracking, personnel tracking, location-based games, as well as home and fleet management.

This document describes the GNSS function of EG915Q series and EG916Q-GL modules. The modules support the NMEA 0183 standard protocol and RTCM protocol, and can be controlled and configured by AT commands, PAIR messages, and PQ messages.

1.1. All-in-One Solution and Stand-Alone Solution

The modules integrate both LTE and GNSS engines, which can work as a single unit (all-in-one solution) or work independently (stand-alone solution) according to your demands.

When the all-in-one solution is enabled, LTE part and GNSS engine work as a single unit. The GNSS engine can be regarded as a peripheral of the LTE part. The LTE part can internally supply power to the GNSS engine via the LDO, without an external power supply. If the LTE part is disabled, the GNSS engine does not work. This enables seamless communication between LTE part and GNSS engine via AT commands for GNSS control, GNSS firmware upgrades, and EPO data downloads.

When the stand-alone solution is enabled, LTE part and GNSS engine work independently of each other. Therefore, they should be controlled separately by MCU.

The acquisition of NMEA output data, and sending of PAIR or PQ messages, vary depending on the enabled solution. The differences between the all-in-one solution and stand-alone solution are listed in the following table:

Table 1: Differences Between All-in-One Solution and Stand-Alone Solution

Item	All-in-One Solution	Stand-Alone Solution
NMEA Output Data Acquisition	<ul style="list-style-type: none"> Acquire via AT+QGPSNMEA Acquire directly 	Acquire directly
PAIR Message Sending	Send via AT+QGNSSCMD	Send directly
PQ Message Sending	Send via AT+QGNSSCMD	Send directly

NOTE

- AT commands are effective only on modules with enabled all-in-one solution.
- The unencapsulated PAIR and PQ messages are effective only on modules with enabled stand-alone solution. The PAIR and PQ messages can be used on modules with enabled all-in-one solution by encapsulating into AT commands. For details about the encapsulation format, see **Chapter 2.3.11** for details.
- If GNSS_VBCKP and GNSS_PWR_EN are used for power supply, by default, stand-alone solution is enabled after powering on. You can switch to all-in-one solution using **AT+QGPS=1**, or switch back to stand-alone solution with **AT+QGPS=0**.

1.2. Procedures for Enabling /Disabling GNSS Function

The GNSS engine of the modules supports location calculation without any network assistance. Procedures for enabling and disabling GNSS are shown below:

Step 1: Configure GNSS parameters through **AT+QGPSCFG**. This step can be omitted if default settings are used.

Step 2: Enable the GNSS function through **AT+QGPS**.

Step 3: Once the GNSS function is enabled and position is fixed successfully, obtain the positioning information by using any of the following methods:

- NMEA sentences are output to "usbntmea" port by default and can be obtained by reading the port.
- Obtain positioning information such as latitude, longitude, altitude, GNSS positioning mode, time, and number of satellites, directly through **AT+QGPSLOC**.

Step 4: Disable the GNSS function through **AT+QGPSEND**.

NOTE

For details about the above AT commands, refer to **Chapter 2.3**.

1.3. Supported NMEA Sentence Types

The default NMEA sentences of the modules are compatible with NMEA 0183 protocol V4.11. Talker IDs listed below are available to differentiate NMEA sentences of different satellite systems.

Talker ID of GPS NMEA sentences is "GP":

- GPGLSV – GPS satellites in view (number of satellites in view and satellite ID numbers, etc.).

Talker ID of GLONASS NMEA sentences is "GL":

- GLGLSV – GLONASS satellites in view (number of satellites in view and satellite ID numbers, etc.).

Talker ID of Galileo NMEA sentences is "GA":

- GAGSV – Galileo satellites in view (number of satellites in view and satellite ID numbers, etc.).

Talker ID of BDS NMEA sentences is "GB":

- GBGLSV – BDS satellites in view (number of satellites in view and satellite ID numbers, etc.).

Talker ID of GNSS multi-constellation NMEA sentences is "GN":

- GNGGA – Global positioning system fix data, such as time and position
- GNRMC – Recommended minimum specific GNSS data
- GNGSA – GNSS DOP and active satellites
- GNVTG – Course over ground and ground speed
- GNGLL – Geographic position – latitude/longitude
- GNGNS – GNSS positioning data

1.4. EPO Assistance Introduction

EPO assistance technology enhances GNSS performance. It simplifies the GNSS positioning process by providing essential information to the GNSS engine, including ephemeris data, almanac data, ionospheric data, UTC, satellite health status, and approximate time. When the EPO assistance technology is activated, the TTFF (Time To First Fix) is reduced by 18–30 seconds (or more in harsh environments with weak satellite signals). The assistance data obtained from the EPO assistance web

server needs to be updated before it expires.

Before using this feature, ensure that valid EPO assistance data is available. You must download the latest EPO binary file, which contains the data from the EPO assistance web server, from the following URL:

URL format:

[http://elpo.airoha.com/\[FILENAME\]?vendor=QUECTEL&project=vSfK9CYTCe1iK7GyDf4FzFriVgcgsbNGjA1luetT-z8&device_id=\[DEVICE_ID\]](http://elpo.airoha.com/[FILENAME]?vendor=QUECTEL&project=vSfK9CYTCe1iK7GyDf4FzFriVgcgsbNGjA1luetT-z8&device_id=[DEVICE_ID])

NOTE

The URL is the server address of the manufacturer of GNSS chipset inside the modules.

[FILENAME] indicates the name of EPO assistance data file.

- "EPO_GR_3_%d.DAT": Name of GPS and GLONASS assistance data file, which provides data for 3–14 days. The range of %d: 1–5.

"EPO_GR_3_1.DAT"	3 days
"EPO_GR_3_2.DAT"	4–6 days
"EPO_GR_3_3.DAT"	7–9 days
"EPO_GR_3_4.DAT"	10–12 days
"EPO_GR_3_5.DAT"	13–14 days
- "EPO_GPS_3_%d.DAT": Name of GPS assistance data file, which provides data for 3–14 days. Range of %d: 1–5.

"EPO_GPS_3_1.DAT"	3 days
"EPO_GPS_3_2.DAT"	4–6 days
"EPO_GPS_3_3.DAT"	7–9 days
"EPO_GPS_3_4.DAT"	10–12 days
"EPO_GPS_3_5.DAT"	13–14 days
- "EPO_GAL_3.DAT" and "EPO_GAL_7.DAT": Names of Galileo assistance data files, which provide data for 3 days and 7 days, respectively.
- "EPO_BDS_3.DAT": Name of BDS assistance data file, which provides data for 3 days.

[DEVICE_ID] indicates the device ID. Use the IMEI of the module plus a random number as the device ID to ensure that it is globally unique. Each *[DEVICE_ID]* can access the EPO data server 20 times per day.

Update EPO assistance data regularly. You can query the status of EPO data files through **AT+QAGPSDATA?** before updating the data.

The operation procedures of EPO assistance technology are as follows:

Step 1: Enable GNSS feature via **AT+QGPS**.

Step 2: Enable EPO assistance through **AT+QAGPS=1** because this feature is disabled by default.

Step 3: Query and confirm the validity of the EPO data file via **AT+QAGPSDATA?**.

- If the data is valid, the current data will be used.
- If the data is invalid, perform **Step 4–Step 7**.

Step 4: Download the EPO assistance data file to the modules according to the URL address listed above.

Step 5: Send reference location to GNSS engine via **AT+QAGPSLOC**.

Step 6: Inject the correct EPO time into the GNSS engine via **AT+QAGPSTIME**.

Step 7: Inject a valid EPO data file into the GNSS engine via **AT+QAGPSDATA**.

NOTE

For details about the above AT commands, refer to **Chapter 2.3**.

2 Description of GNSS AT Commands

2.1. AT Command Introduction

2.1.1. Definitions

- **<CR>** Carriage return character.
- **<LF>** Line feed character.
- **<...>** Parameter name. Angle brackets do not appear on the command line.
- **[...]** Optional parameter of a command or an optional part of TA information response. Square brackets do not appear on the command line. When an optional parameter is not given in a command, the new value equals its previous value or the default settings, unless otherwise specified.
- **Underline** Default setting of a parameter.

2.1.2. AT Command Syntax

All command lines must start with **AT** or **at** and end with **<CR>**. Information responses and result codes always start and end with a carriage return character and a line feed character: **<CR><LF><response><CR><LF>**. In tables presenting commands and responses throughout this document, only the commands and responses are presented, and **<CR>** and **<LF>** are deliberately omitted.

Table 2: Types of AT Commands

Command Type	Syntax	Description
Test Command	AT+<cmd>=?	Test the existence of the corresponding command and return information about the type, value, or range of its parameter.
Read Command	AT+<cmd>?	Check the current parameter value of the corresponding command.
Write Command	AT+<cmd>=<p1>[,<p2>[,<p3>[...]]]	Set user-definable parameter value.
Execution Command	AT+<cmd>	Return a specific information parameter or perform a specific action.

2.2. Declaration of AT Command Examples

The AT command examples in this document are provided to help you learn about the use of the AT commands introduced herein. The examples, however, should not be taken as Quectel's recommendations or suggestions about how to design a program flow or what status to set the module into. Sometimes multiple examples may be provided for one AT command. However, this does not mean that there is a correlation among these examples, or that they should be executed in a given sequence.

2.3. AT Command Description

The commands introduced in this chapter are used to control or configure the internal GNSS engine of the module. These commands are effective only if all-in-one solution is enabled.

2.3.1. AT+QGPSCFG Configure GNSS

This command queries and configures various GNSS settings, including the output port and output types of NMEA sentences, etc.

AT+QGPSCFG Configure GNSS	
Test Command AT+QGPSCFG=?	Response +QGPSCFG: "outport", (list of supported <out_port>s) +QGPSCFG: "nmeasrc", (list of supported <NMEA_src>s) +QGPSCFG: "gpsnmeatype", (list of supported <GPS_NMEA_type>s) +QGPSCFG: "glonassnmeatype", (list of supported <GLONASS_NMEA_type>s) +QGPSCFG: "galileonmeatype", (list of supported <Galileo_NMEA_type>s) +QGPSCFG: "beidoumeatype", (list of supported <BeiDou_NMEA_type>s) +QGPSCFG: "gnssconfig", (list of supported <GNSS_config>s) +QGPSCFG: "autogps", (list of supported <autoGPS>s) +QGPSCFG: "gnssota", <OTA_URL> OK

2.3.1.1. AT+QGPSCFG="outport" Configure Output Port of NMEA Sentences

This command configures the output port of NMEA sentences and the baud rate of the port.

AT+QGPSCFG="outport" Configure Output Port of NMEA Sentences	
<p>Write Command</p> <p>AT+QGPSCFG="outport"[,<outport>[,<baud_rate>]]</p>	<p>Response</p> <p>If the optional parameters are omitted, query the current setting.</p> <p>+QGPSCFG: "outport",<out_port>[,<baud_rate>]</p> <p>OK</p> <p>If any of the optional parameters is specified, set the output port of NMEA sentences.</p> <p>OK</p> <p>Or</p> <p>ERROR</p> <p>If there is any error related to ME functionality:</p> <p>+CME ERROR: <errcode></p>
Maximum Response Time	300 ms
Characteristics	<p>The command takes effect immediately.</p> <p>The configurations are saved automatically.</p>

Parameter

<out_port>	<p>String type. Output port of NMEA sentences.</p> <p>"none" Close NMEA sentence output</p> <p>"usbnmea" Output through USB NMEA port</p> <p>"uartdebug" Output through Debug UART port</p>
<baud_rate>	<p>Integer type. Baud rate of Debug UART port. Valid when the NMEA sentence is output through Debug UART port. Default value: 115200. Unit: bps.</p>
<errcode>	<p>Integer type. Error code. See Chapter 2.5 for details.</p>

NOTE

If the "uartdebug" port is used to output NMEA sentences, it cannot be used to print logs.

2.3.1.2. AT+QGPSCFG="nmeasrc" Enable/Disable Acquisition of NMEA Sentences through

AT+QGPSTNMEA

This command enables or disables acquisition of NMEA sentences through **AT+QGPSTNMEA**.

AT+QGPSCFG="nmeasrc" Enable/Disable Acquisition of NMEA Sentences through AT+QGPSTNMEA

Write Command AT+QGPSCFG="nmeasrc"[,<NMEA_src>]	Response If the optional parameter is omitted, query the current setting. +QGPSCFG: "nmeasrc",<NMEA_src> OK If the optional parameter is specified, set whether to enable or disable acquisition of NMEA sentences through AT+QGPSTNMEA . OK Or ERROR If there is any error related to ME functionality: +CME ERROR: <errcode>
Maximum Response Time	300 ms
Characteristics	The command takes effect immediately. The configuration is saved automatically.

Parameter

<NMEA_src>	Integer type. Enable or disable the acquisition of NMEA sentences through AT+QGPSTNMEA . If enabled, after AT+QGPSTNMEA is executed, NMEA sentences are output through the AT port as a return value. 0 Disable 1 Enable
<errcode>	Integer type. Error code. See Chapter 2.5 for details.

2.3.1.3. AT+QGPSCFG="gpsnmeatype" Configure Output Type of GPS or GNSS Multi-constellation NMEA Sentences

This command configures the output type of GPS or GNSS multi-constellation NMEA sentences.

AT+QGPSCFG="gpsnmeatype" Configure Output Type of GPS or GNSS Multi-constellation NMEA Sentences	
Write Command AT+QGPSCFG="gpsnmeatype"[,<GPS_NMEA_type>]	<p>Response</p> <p>If the optional parameter is omitted, query the current setting. +QGPSCFG: "gpsnmeatype",<GPS_NMEA_type></p> <p>OK</p> <p>If the optional parameter is specified, set the output type of GPS NMEA sentences.</p> <p>OK</p> <p>Or</p> <p>ERROR</p> <p>If there is any error related to ME functionality: +CME ERROR: <errcode></p>
Maximum Response Time	300 ms
Characteristics	The command takes effect after the module is rebooted. The configuration is saved automatically.

Parameter

<GPS_NMEA_type>	Integer type. Output type of GPS or GNSS multi-constellation NMEA sentences in XOR format.
0	Disable
1	GNGGA
2	GNRMC
4	GPGSV
8	GNGSA
16	GNVTG
31	All of the above five types of sentences
32	GNGLL
63	All of the above six types of sentences
<errcode>	Integer type. Error code. See Chapter 2.5 for details.

2.3.1.4. AT+QGPSCFG="glonassnmeatype" Configure Output Type of GLONASS or GNSS Multi-constellation NMEA Sentences

This command configures the output type of GLONASS or GNSS multi-constellation NMEA sentences.

AT+QGPSCFG="glonassnmeatype" Configure Output Type of GLONASS or GNSS Multi-constellation NMEA Sentences

Write Command AT+QGPSCFG="glonassnmeatype"[, <GLONASS_NMEA_type>]	Response If the optional parameter is omitted, query the current setting. +QGPSCFG: "glonassnmeatype",<GLONASS_NMEA_type> OK If the optional parameter is specified, set the output type of GLONASS or GNSS multi-constellation NMEA sentences. OK Or ERROR If there is any error related to ME functionality: +CME ERROR: <errcode>
Maximum Response Time	300 ms
Characteristics	The command takes effect after the module is rebooted. The configuration is saved automatically.

Parameter

<GLONASS_NMEA_type>	Integer type. Output type of GLONASS or GNSS multi-constellation NMEA sentences in XOR format. <u>0</u> Disable 1 GLGSV 2 GNGNS
<errcode>	Integer type. Error code. See Chapter 2.5 for details.

2.3.1.5. AT+QGPSCFG="galileonmeatype" Configure Output Type of Galileo NMEA Sentences

This command configures the output type of Galileo NMEA sentences.

AT+QGPSCFG="galileonmeatype" Configure Output Type of Galileo NMEA Sentences	
Write Command AT+QGPSCFG="galileonmeatype"[,<Galileo_NMEA_type>]	Response If the optional parameter is omitted, query the current setting. +QGPSCFG: "galileonmeatype",<Galileo_NMEA_type> OK If the optional parameter is specified, set the output type of Galileo NMEA sentences. OK Or ERROR If there is any error related to ME functionality: +CME ERROR: <errcode>
Maximum Response Time	300 ms
Characteristics	The command takes effect after the module is rebooted. The configuration is saved automatically.

Parameter

<Galileo_NMEA_type>	Integer type. Output type of Galileo NMEA sentences in XOR format. <u>0</u> Disable 1 GAGSV
<errcode>	Integer type. Error code. See Chapter 2.5 for details.

2.3.1.6. AT+QGPSCFG="beidoumeatype" Configure Output Type of BDS NMEA Sentences

This command configures the output type of BDS NMEA sentences.

AT+QGPSCFG="beidoumeatype" Configure Output Type of BDS NMEA Sentences	
Write Command AT+QGPSCFG="beidoumeatype"[, <BeiDou_NMEA_type>]	Response If the optional parameter is omitted, query the current setting. +QGPSCFG: "beidoumeatype",<BeiDou_NMEA_type> OK If the optional parameter is specified, set the output type of BDS NMEA sentences. OK Or ERROR If there is any error related to ME functionality: +CME ERROR: <errcode>
Maximum Response Time	300 ms
Characteristics	The command takes effect after the module is rebooted. The configuration is saved automatically.

Parameter

<BeiDou_NMEA_type>	Integer type. Output type of BDS NMEA sentences in XOR format. <u>0</u> Disable 1 GBGSV
<errcode>	Integer type. Error code. See Chapter 2.5 for details.

2.3.1.7. AT+QGPSCFG="gnssconfig" Configure Supported GNSS Constellations

This command configures the GNSS constellations supported by the module.

AT+QGPSCFG="gnssconfig" Configure Supported GNSS Constellations	
Write Command AT+QGPSCFG="gnssconfig"[,<GNSS_config>]	<p>Response</p> <p>If the optional parameter is omitted, query the current setting. +QGPSCFG: "gnssconfig",<GNSS_config></p> <p>OK</p> <p>If the optional parameter is specified, set the supported GNSS constellations. OK</p> <p>Or ERROR</p> <p>If there is any error related to ME functionality: +CME ERROR: <errcode></p>
Maximum Response Time	300 ms
Characteristics	The command takes effect after the module is rebooted. The configuration is saved automatically.

Parameter

<GNSS_config>	Integer type. Supported GNSS constellations. GPS is always ON.
0	GPS
1	GPS + GLONASS + BDS + Galileo
2	GPS + GLONASS + BDS
3	GPS + GLONASS + Galileo
4	GPS + GLONASS
5	GPS + BDS + Galileo
6	GPS + Galileo
7	GPS + BDS
<errcode>	Integer type. Error code. See Chapter 2.5 for details.

2.3.1.8. AT+QGPSCFG="autogs" Enable/Disable Automatic Running of GNSS

This command enables/disables the automatic running of GNSS after the module is rebooted.

AT+QGPSCFG="autogs" Enable/Disable Automatic Running of GNSS	
Write Command AT+QGPSCFG="autogs"[,<autoGPS>]	Response If the optional parameter is omitted, query the current setting. +QGPSCFG: "autogs",<autoGPS> OK If the optional parameter is specified, enable or disable automatic running of GNSS: OK Or ERROR If there is any error related to ME functionality: +CME ERROR: <errcode>
Maximum Response Time	300 ms
Characteristics	The command takes effect after the module is rebooted. The configuration is saved automatically.

Parameter

<autoGPS>	Integer type. Enable or disable automatic running of GNSS after the module is powered on. <u>0</u> Disable 1 Enable
<errcode>	Integer type. Error code. See Chapter 2.5 for details.

2.3.1.9. AT+QGPSCFG="gnssota" Configure URL for Downloading GNSS Firmware Package

This command configures the URL for downloading the GNSS firmware package.

AT+QGPSCFG="gnssota" Configure URL for Downloading GNSS Firmware Package	
Write Command AT+QGPSCFG="gnssota"[,<OTA_URL>]	Response If the optional parameter is omitted, query the current setting.

	+QGPSCFG: "gnssota",<OTA_URL> OK If the optional parameter is specified, set the URL for downloading GNSS firmware package. OK Or ERROR If there is any error related to ME functionality: +CME ERROR: <errcode>
Maximum Response Time	300 ms
Characteristics	The command takes effect immediately. The configuration is saved automatically.

Parameter

<OTA_URL>	String type. The URL for downloading GNSS firmware package or for loading the GNSS firmware upgrade package from the local file system. It supports HTTP server files or local files.
<errcode>	Integer type. Error code. See Chapter 2.5 for details.

Example

```
AT+QGPSCFG="gnssota","http://192.0.2.2:21/TEST/AG3352_BINPKG.bin"
```

```
OK //Set the HTTP download path. "AG3352_BINPKG.bin" is the GNSS firmware upgrade package to be downloaded.
```

```
AT+QGPSCFG="gnssota","FILE:1269300"
```

```
OK //"FILE:" indicates loading of the GNSS firmware upgrade package from the local file system for GNSS upgrade. "1269300" is the size of the local GNSS firmware upgrade package in bytes.
```

2.3.2. AT+QGPSOTA GNSS Firmware Upgrade

This command upgrades GNSS firmware. Before executing this command, execute **AT+QGPS=1** to enable GNSS function, and then execute **AT+QGPSCFG="gnssota"[,<OTA_URL>]** to set the download path of the target GNSS firmware file.

After executing **AT+QGPSOTA**, the target GNSS firmware package is downloaded from the HTTP server or the local file is sent to the module for the upgrade.

AT+QGPSOTA GNSS Firmware Upgrade

Read Command AT+QGPSOTA?	Response +QGPSOTA: <OTA_state> OK
Execution Command AT+QGPSOTA	Response OK
Maximum Response Time	300 ms
Characteristics	The command takes effect immediately.

Parameter

<OTA_state>	Integer type. Current GNSS OTA status.
0	No GNSS OTA performed or GNSS OTA completed
1	GNSS OTA to be completed

NOTE

1. If the GNSS firmware package is downloaded from an HTTP server, after **AT+QGPSOTA** is executed, the download and upgrade processes are initiated automatically. If the upgrade fails due to a power interruption, the download and upgrade processes will automatically resume after the module is rebooted and reconnected to the network.
2. If a local file is used for the upgrade, after executing **AT+QGPSOTA**, you need to send the GNSS firmware package to the module through a serial tool such as QCOM. The upgrade process will begin when the module receives the firmware package. For more details about QCOM tool, see **document [1]**. If the power is interrupted during the upgrade, the upgrade will not resume after the module is rebooted.

2.3.3. AT+QGPSDEL Delete Assistance Data

This command deletes assistance data to perform a GNSS cold start. The command can only be executed when GNSS is disabled. After deleting the assistance data through this command, you can trigger a cold start through **AT+QGPS**. Hot/warm start can also be performed if no assistance data or partial data are deleted.

AT+QGPSDEL Delete Assistance Data

Test Command AT+QGPSDEL=?	Response +QGPSDEL: (list of supported <delete_type>s) OK
-------------------------------------	--

Write Command AT+QGPSDEL=<delete_type>	Response OK Or ERROR If there is any error related to ME functionality: +CME ERROR: <errcode>
Maximum Response Time	300 ms
Characteristics	The command takes effect immediately. The configuration is not saved.

Parameter

<delete_type>	Integer type. Type of GNSS assistance data to be deleted. 0 Delete all assistance data. Perform a cold start after starting GNSS. 1 Do not delete any data. Perform a hot start if the conditions are met after starting GNSS. 2 Delete some related data. Perform a warm start if the conditions are met after starting GNSS. 3 Delete the EPO assistance data injected into GNSS engine.
<errcode>	Integer type. Error code. See Chapter 2.5 for details.

2.3.4. AT+QGPS Enable or Disable GNSS Feature

The command enables or disable GNSS feature.

AT+QGPS Enable or Disable GNSS	
Test Command AT+QGPS=?	Response +QGPS: (list of supported <GNSS_state>s) OK
Read Command AT+QGPS?	Response +QGPS: <GNSS_state> OK
Write Command AT+QGPS=<GNSS_state>	Response OK Or ERROR If there is any error related to ME functionality:

	+CME ERROR: <errcode>
Maximum Response Time	300 ms
Characteristics	The command takes effect immediately. The configuration is not saved.

Parameter

<GNSS_state>	Integer type. GNSS state. 0 Disable (Switch to stand-alone solution) 1 Enable (Switch to all-in-one solution)
<errcode>	Integer type. Error code. See Chapter 2.5 for details.

NOTE

Other GNSS related actions can be performed only after GNSS is enabled successfully through **AT+QGPS=1** and a NMEA sentence is output from NMEA sentence output port.

2.3.5. AT+QGPSEND Disable GNSS Function

This command disables GNSS function. After the GNSS is enabled through **AT+QGPS=1**, you can forcibly terminate positioning by using **AT+QGPSEND**.

AT+QGPSEND Disable GNSS	
Test Command AT+QGPSEND=?	Response OK
Read Command AT+QGPSEND?	Response OK Or ERROR
Execution Command AT+QGPSEND	Response OK Or ERROR If there is any error related to ME functionality: +CME ERROR: <errcode>
Maximum Response Time	300 ms
Characteristics	This command takes effect immediately.

Parameter

<errcode> Integer type. Error code. See **Chapter 2.5** for details.

2.3.6. AT+QGPSLOC Acquire Positioning Information

This command acquires positioning information. Before executing this command, enable GNSS via **AT+QGPS=1**. If GNSS fails to fix a position, **+CME ERROR: <errcode>** is returned to indicate the corresponding issue.

AT+QGPSLOC Acquire Positioning Information	
Test Command AT+QGPSLOC=?	Response +QGPSLOC: <UTC>,<latitude>,<longitude>,<hdop>,<altitude>,<fix>,<COG>,<spkm>,<spkn>,<date>,<nsat> OK
Write Command AT+QGPSLOC=<mode>	Response +QGPSLOC: <UTC>,<latitude>,<longitude>,<hdop>,<altitude>,<fix>,<COG>,<spkm>,<spkn>,<date>,<nsat> OK Or ERROR If there is any error related to ME functionality : +CME ERROR: <errcode>
Maximum Response Time	300 ms
Characteristics	The command takes effect immediately. The configuration is not saved.

Parameter

<mode> Integer type. Latitude and longitude display format.

0 **<latitude>,<longitude>** format: ddmm.mmmmmN/S,dddmm.mmmmmE/W

1 **<latitude>,<longitude>** format: ddmm.mmmmmmm,N/S,dddmm.mmmmmmm,E/W

2 **<latitude>,<longitude>** format: (-)dd.ddd,(-)ddd.ddd

<UTC> String type (without double quotation marks). UTC time. Format: hhmmss.ss. (Quoted from GNGGA statement.)

<latitude> String type (without double quotation marks). Latitude.

If **<mode>** is 0:
 Format: ddmm.mmmmm N/S (Quoted from GNGGA statement.)

dd Degree. Range: 00–89
mm.mmmm Minute. Range: 00.0000–59.9999
N/S North/South

If **<mode>** is 1:

Format: ddm.mmmmm,N/S (Quoted from GNGGA statement.)

dd Degree. Range: 00–89
mm.mmmmm Minute. Range: 00.000000–59.999999
N/S North /South

If **<mode>** is 2:

Format: (-)dd.ddddd (Quoted from GNGGA statement.)

dd.ddddd Degree. Range: -89.99999–89.99999
- South

<longitude> String type (without double quotation marks). Longitude.

If **<mode>** is 0:

Format: dddmm.mmmmE/W (Quoted from GNGGA statement.)

ddd Degree. Range: 000–179
mm.mmmm Minute. Range: 00.0000–59.9999
E/W East/West

If **<mode>** is 1:

Format: dddmm.mmmmm,E/W (Quoted from GNGGA statement.)

ddd Degree. Range: 000–179
mm.mmmmm Minute. Range: 00.000000–59.999999
E/W East/West

If **<mode>** is 2:

Format: (-)ddd.ddddd (Quoted from GNGGA statement.)

ddd.ddddd Degree. Range: -179.99999–179.99999
- West

<hdop> Double type. Horizontal dilution of precision. (Quoted from GNGGA statement.)

<altitude> Double type. Altitude of the antenna away from the sea level. Accurate to one decimal place. Unit: meter. (Quoted from GNGGA statement.)

<fix> Integer type. GNSS positioning mode. (Quoted from GNGSA statement.)

2 2D positioning
3 3D positioning

<COG> String type (without double quotation marks). Course Over Ground based on true north.

Format: ddd.mm (Quoted from GNVTG statement.)

ddd Degree. Range: 000–359
mm Minute. Range: 00–99

<spkm> Double type. Speed over ground. (Quoted from GNVTG statement.)

Format: xxxx.x. Unit: km/h. Accurate to one decimal place.

<spkn>	Double type. Speed over ground. (Quoted from GNVTG statement.) Format: xxxx.x. Unit: knots. Accurate to one decimal place.
<date>	String type (without double quotation marks). UTC time when fixing position. Format: ddmmyy. (Quoted from GNRMC statement.) dd Day mm Month yy Year
<nsat>	Integer type. Number of active satellites. Keep the value at two digits, and add 0 if the leading digit is insufficient. (Quoted from GNGGA statement.)
<errcode>	Integer type. Error code. See Chapter 2.5 for details.

2.3.7. AT+QAGPS Enable/Disable EPO Assistance

This command enables or disables EPO assistance.

AT+QAGPS Enable/Disable EPO Assistance	
Test Command AT+QAGPS=?	Response +QAGPS: (list of supported <assisted_enable> s) OK
Read Command AT+QAGPS?	Response +QAGPS: <assisted_enable> OK
Write Command AT+QAGPS=<assisted_enable>	Response OK Or ERROR If there is any error related to ME functionality: +CME ERROR: <errcode>
Maximum Response Time	300 ms
Characteristics	The command takes effect immediately. The configuration is not saved.

Parameter

<assisted_enable>	Integer type. Enable or disable EPO assistance. 0 Disable 1 Enable
<errcode>	Integer type. Error code. See Chapter 2.5 for details.

2.3.8. AT+QAGPSTIME Inject EPO Time

This command injects an EPO time to GNSS engine. Before using this command, enable the EPO assistance feature through **AT+QAGPS=1**. After the feature is enabled, the GNSS engine requests EPO time and assistance data file. Inject EPO time with this command before injecting an EPO data file.

AT+QAGPSTIME Inject EPO Time	
Test Command AT+QAGPSTIME=?	Response +QAGPSTIME: 0,<assisted_time>,(list of supported <UTC>s),(list of supported <force>s),<uncrtn> OK
Write Command AT+QAGPSTIME=<method>,<assisted_time>,<UTC>,<force>,<uncrtn>	Response OK Or ERROR If there is any error related to ME functionality: +CME ERROR: <errcode>
Maximum Response Time	300 ms
Characteristics	The command takes effect immediately. The configurations are not saved.

Parameter

<method>	Integer type. Method of injecting time. <u>0</u> Inject EPO time manually
<assisted_time>	String type. Current UTC time. Format: "yyyy/mm/dd,hh:mm:ss". For example, "2019/01/05,08:30:30".
<UTC>	Integer type. Time type. <u>0</u> GPS time (Not supported currently) <u>1</u> UTC time
<force>	Integer type. Whether to force the GNSS subsystem to accept the injected time. <u>0</u> Not force (Not supported currently) <u>1</u> Force
<uncrtn>	Integer type. Uncertainty of time. It indicates the time difference between the time of sending a request to the SNTP server and the time of receiving a response from the SNTP server. Default value: 3500. Unit: millisecond. (This parameter is not supported currently and you can configure it as any number from 0 to 65536).
<errcode>	Integer type. Error code. See Chapter 2.5 for details.

2.3.9. AT+QAGPSDATA Inject EPO Data File

This command injects an EPO assistance data file to GNSS engine. Before executing this command, enable EPO assistance through **AT+QAGPS=1**, then store the valid EPO data file in UFS of the module through **AT+QFUPL** and inject the EPO time to GNSS engine through **AT+QAGPSTIME**. After executing this command successfully, you can delete the EPO data file from UFS through **AT+QFDEL**. Whether the EPO data file has been injected successfully can be queried through **AT+QAGPSDATA?**. For details of **AT+QFUPL** and **AT+QFDEL**, see *document [2]*.

AT+QAGPSDATA Inject EPO Data File	
Test Command AT+QAGPSDATA=?	Response +QAGPSDATA: <assisted_data_filename> OK
Read Command AT+QAGPSDATA?	Response +QAGPSDATA: <satellite_type>,<assisted_data_durtim e>,<injected_data_time> [...] OK Or ERROR If the error is related to ME functionality: +CME ERROR: <errcode>
Write Command AT+QAGPSDATA=<assisted_data_fil ename>	Response OK Or ERROR If there is any error related to ME functionality: +CME ERROR: <errcode>
Maximum Response Time	1 s
Characteristics	The command takes effect immediately. The configuration is not saved.

Parameter

<assisted_data_filename>	String type. Name of the EPO data file to be injected, for example, "EPO_GPS_3_1.DAT". See Chapter 1.4 for details.
<satellite_type>	Integer type. Satellite type. 0 GPS

	1 GLONASS
	2 Galileo
	3 BDS
<assisted_data_durtime>	Integer type. Validity period of injected EPO data file. Unit: minute. 0 No EPO file or expired EPO file 1–20160 Validity period of the EPO file
<injected_data_time>	String type (without double quotation marks). Start time of the validity period of the injected EPO data file. Format: yyyy/mm/dd,hh:mm:ss, for example, 2016/01/03,15:34:50. This parameter is fixed to 1980/01/05,23:59:42 if there is no EPO file or EPO file is expired.
<errcode>	Integer type. Error code. See Chapter 2.5 for details.

2.3.10. AT+QAGPSAID Inject EPO Data File Directly

The command injects an EPO data file to the module directly. This command includes functions for downloading EPO assistance data files, injecting EPO reference time, and injecting EPO assistance data files.

AT+QAGPSAID Inject EPO Data File Directly	
Test Command AT+QAGPSAID=?	Response +QAGPSAID: <epodata_inject> OK
Write Command AT+QAGPSAID=<epodata_inject>	Response OK If there is any error related to ME functionality: +CME ERROR: <errcode>
Maximum Response Time	300 ms
Characteristics	The command takes effect immediately. The configuration is not saved.

Parameter

<epodata_inject>	Integer type. Inject the EPO data file directly. 1 Enable injecting the EPO data file directly
<errcode>	Integer type. Error code. See Chapter 2.5 for details.

NOTE

1. For the URL downloading EPO files, refer to **AT+QAGPSCFG** (*Chapter 2.3.13*).
2. After the direct EPO data injection function is enabled through **AT+QAGPSAID=1**, wait 20 seconds before querying the injection result via **AT+QAGPSDATA?** to avoid sending other commands within 20 seconds of issuing this command. Otherwise, the injection may fail.

2.3.11. AT+QGNSSCMD Send Commands to GNSS Engine

This command sends commands to the GNSS engine, allowing users to optionally use some functions for application demands.

AT+QGNSSCMD Send Commands to GNSS Engine	
Test Command AT+QGNSSCMD=?	Response +QGNSSCMD: (list of supported <cmdType>s),<cmdString> OK
Write Command AT+QGNSSCMD=<cmdType>,<cmdString>	Response If a PAIR message is sent: OK +QGNSSCMD: \$PAIR001,<MessageID>,<Result>*<Checksum> If a PQ message is sent: OK +QGNSSCMD: <result_string> If there is any error related to ME functionality: +CME ERROR: <errcode>
Maximum Response Time	300 ms
Characteristics	The command takes effect immediately. The configurations are not saved.

Parameter

<cmdType>	Integer type. Type of <cmdString>. 0 NMEA style 1 Hex style (currently not supported)
<cmdString>	String type. Message string (PAIR or PQ message). See <i>Chapter 3</i> or <i>Chapter 4</i> for details.

<MessageID>	Integer type. Type of PAIR message/packet to be acknowledged. See Chapter 3.1 for details.
<Result>	Integer type. Execution result of PAIR message. See Chapter 3.1 for details.
<Checksum>	Checksum. See Chapter 3 and Chapter 4 for details.
<result_string>	String type (without double quotation marks). Execution result of PQ message. See Chapter 4 for details.
<errcode>	Integer type. Error code. See Chapter 2.5 for details.

NOTE

1. After a PAIR message is executed using **AT+QGNSSCMD=<cmdType>,<cmdString>** and an **OK** is responded, the URC **+QGNSSCMD: \$PAIR001,<MessageID>,<Result>*<Checksum>** is reported, indicating the execution result. The execution result varies depending on **<MessageID>**. For detailed information, refer to **Chapter 3**.
2. After a PQ message is executed using **AT+QGNSSCMD=<cmdType>,<cmdString>** and an **OK** is responded, the URC **+QGNSSCMD: <result_string>** is reported, indicating the execution result. The **<result_string>** varies depending on **<cmdString>**. For detailed information, refer to **Chapter 4**.

2.3.12. AT+QAGPSLOC Send Reference Location to GNSS Engine

The command sends reference location to GNSS engine.

AT+QAGPSLOC Send Reference Location to GNSS Engine	
Test Command AT+QAGPSLOC=?	Response +QAGPSLOC: <ref_latitude>,<ref_longitude> OK
Write Command AT+QAGPSLOC=<ref_latitude>,<ref_longitude>	Response OK If there is any error related to ME functionality: +CME ERROR: <errcode>
Maximum Response Time	300 ms
Characteristics	The command takes effect immediately. The configurations are not saved.

Parameter

<ref_latitude>	String type (without double quotation marks). Latitude of the reference location. Range: -90 – 90. Unit: degree.
<ref_longitude>	String type (without double quotation marks). Longitude of the reference location. Range: -180 – 180. Unit: degree.
<errcode>	Integer type. Error code. See Chapter 2.5 for details.

NOTE

The input format of **<ref_latitude>** and **<ref_longitude>** should retain at least 6 decimal places.

2.3.13. AT+QAGPSCFG Set URL for Downloading EPO Files

The command sets the URL for downloading EPO files, after which the EPO data file can be directly injected to the module via **AT+QAGPSAID**.

AT+QAGPSCFG Set URL for Downloading EPO Files	
Test Command AT+QAGPSCFG=?	Response +QAGPSCFG: <URL>,<File_name>,<vendorID>,<project_ID>,<device_ID> OK
Read Command AT+QAGPSCFG?	Response +QAGPSCFG: <URL> OK Or ERROR If there is any error related to ME functionality: +CME ERROR: <errcode>
Write Command AT+QAGPSCFG=<URL>,<File_name>,<vendorID>,<project_ID>,<device_ID>	Response OK If there is any error related to ME functionality: +CME ERROR: <errcode>
Maximum Response Time	300 ms
Characteristics	The command takes effect immediately. The configurations are saved automatically.

Parameter

<URL>	String type. Domain name. Fixed to "http://elpo.airoha.com".
<File_name>	String type. EPO file name to be downloaded. See Chapter 1.4 for details.
<vendorID>	String type. Vendor ID. Fixed to "QUECTEL-TEST".
<project_ID>	String type. Project ID. Fixed to "4kn_0DxhWopmK6uRyZHcmsvxGg9sfuPXa3w2StXMDL0".
<device_ID>	String type. Device ID. It is advised to use the IMEI of the module plus a random number as the device ID.
<errcode>	Integer type. Error code. See Chapter 2.5 for details.

NOTE

If you need to configure <vendorID> and <project_ID> to other values, contact Quectel Technical Support for assistance.

2.3.14. AT+QGPSGNMEA Acquire NMEA Sentences

This command acquires NMEA sentences. Before using this command, enable GNSS through **AT+QGPS=1**, and execute **AT+QGPSCFG="nmeasrc",1** to enable acquisition of NMEA sentences through **AT+QGPSGNMEA**.

The NMEA sentence output can be disabled through **AT+QGPSCFG="gpsnmeatype",0**, **AT+QGPSCFG="glonassnmeatype",0**, **AT+QGPSCFG="galileonmeatype",0** or **AT+QGPSCFG="beidoumeatype",0**. This command prevents the module from outputting updated sentences. If you disable sentence output after acquiring NMEA sentences, the module will save the sentences obtained between the time GNSS was enabled and when sentence output was disabled. At this time, you can still retrieve the saved NMEA sentence with **AT+QGPSGNMEA**.

AT+QGPSGNMEA Acquire NMEA Sentences

Test Command AT+QGPSGNMEA=?	Response +QGPSGNMEA: (list of supported <NMEA_type>s) OK
Write Command Query GGA sentence AT+QGPSGNMEA="GGA"	Response [+QGPSGNMEA: <GGA_sentence>] OK If there is any error related to ME functionality: +CME ERROR: <errcode>

Write Command Query RMC sentence AT+QGPSGNMEA="RMC"	Response [+QGPSGNMEA: <RMC_sentence>] OK If there is any error related to ME functionality: +CME ERROR: <errcode>
Write Command Query GSV sentence AT+QGPSGNMEA="GSV"	Response [+QGPSGNMEA: <GSV_sentence>] OK If there is any error related to ME functionality: +CME ERROR: <errcode>
Write Command Query GSA sentence AT+QGPSGNMEA="GSA"	Response [+QGPSGNMEA: <GSA_sentence>] OK If there is any error related to ME functionality: +CME ERROR: <errcode>
Write Command Query VTG sentence AT+QGPSGNMEA="VTG"	Response [+QGPSGNMEA: <VTG_sentence>] OK If there is any error related to ME functionality: +CME ERROR: <errcode>
Write Command Query GNS sentence AT+QGPSGNMEA="GNS"	Response [+QGPSGNMEA: <GNS_sentence>] OK If there is any error related to ME functionality: +CME ERROR: <errcode>
Write Command Query GLL sentence AT+QGPSGNMEA="GLL"	Response [+QGPSGNMEA: <GLL_sentence>] OK If there is any error related to ME functionality: +CME ERROR: <errcode>

Maximum Response Time	300 ms
Characteristics	The command takes effect immediately.

Parameter

<NMEA_type>	String type. NMEA sentence type. "GGA" GGA sentence "RMC" RMC sentence "GSV" GSV sentence "GSA" GSA sentence "VTG" VTG sentence "GNS" GNS sentence "GLL" GLL sentence
<GGA_sentence>	String type (without double quotation marks). GGA sentence.
<RMC_sentence>	String type (without double quotation marks). RMC sentence.
<GSV_sentence>	String type (without double quotation marks). GSV sentence.
<GSA_sentence>	String type (without double quotation marks). GSA sentence.
<VTG_sentence>	String type (without double quotation marks). VTG sentence.
<GNS_sentence>	String type (without double quotation marks). GNS sentence.
<GLL_sentence>	String type (without double quotation marks). GLL sentence.
<errcode>	Integer type. Error code. See Chapter 2.5 for details.

2.4. AT Command Examples

2.4.1. Enable/Disable GNSS

In this example, default parameters are used to enable the GNSS. After the GNSS is enabled, NMEA sentences are output from "usbntmea" port by default. The GNSS can be disabled through **AT+QGPSSEND**.

```
AT+QGPS=1           //Enable GNSS.
OK
//After GNSS is enabled, NMEA sentences are output from "usbntmea" port by default.
AT+QGPSLOC=0        //Acquire positioning information.
+QGPSLOC: 064339.00,2516.1462N,11020.0472E,0.63,160.4,3,241.52,0000.7,0000.4,041124,29
OK
AT+QGPSSEND         //Disable GNSS.
OK
```

2.4.2. Send Reference Location to GNSS Engine

```
AT+QAGPSLOC=?
+QAGPSLOC: <ref_latitude>,<ref_longitude>
OK
AT+QAGPSLOC=25.271863,110.345325
OK
```

2.4.3. Send Commands to GNSS Engine

```
//Send a PAIR message through AT command.
AT+QGNSSCMD=0,"$PAIR420,1*21"
OK
+QGNSSCMD: $PAIR001,420,0*3D

//Send a PQ message through AT command.
AT+QGNSSCMD=0,"$PQEPE,W,0,1*2B"
OK
+QGNSSCMD: $PQEPE,W,OK*02
```


2.4.4. Set URL for Downloading EPO Data Files

```
AT+QAGPSCFG=?
+QAGPSCFG: <URL>,<File_name>,<vendorID>,<project_ID>,<device_ID>

OK
AT+QAGPSCFG="http://elpo.airoha.com","EPO_GPS_3_1.DAT","QUECTEL-TEST","4kn_0DxhWopmK6uRyZHcmsvxGg9sfuPXa3w2StXMDL0","868225060008814123"
OK
```

2.4.5. Inject EPO Data File Directly

```
AT+QGPS=1 //Enable GNSS.
OK
AT+CREG?;+CGREG? //Check network status. See document [3] for the commands.
+CREG: 0,1

+CGREG: 0,1

OK
AT+CCLK? //Get UTC time. See document [3] for this command.
+CCLK: "23/11/25,02:05:30+00"

OK
AT+QAGPSCFG="https://elpo.airoha.com","EPO_GPS_3_1.DAT","QUECTEL-TEST","4kn_0DxhWopmK6uRyZHcmsvxGg9sfuPXa3w2StXMDL0","868225060008814123" //Set the URL for
downloading EPO files.

OK
AT+QAGPSAID=1 //Enable injecting the EPO data file directly.
OK
AT+QAGPSDATA? //Query the status of EPO data files.
+QAGPSDATA: 0,0,2023/11/27,23:59:42
+QAGPSDATA: 1,0,1980/01/05,23:59:42
+QAGPSDATA: 2,0,1980/01/05,23:59:42
+QAGPSDATA: 3,0,1980/01/05,23:59:42

OK
AT+QGPSLOC=0 //Get location information.
+QGPSLOC: 064339.00,2516.1462N,11020.0472E,0.63,160.4,3,241.52,0000.7,0000.4,041124,29

OK
```

2.4.6. Operation Procedures for EPO Assistance

This example shows the operation procedures for EPO assistance. For information about FILE commands used to upload a file to UFS or delete a file from UFS, please see [document \[2\]](#).

```

AT+QGPS=1 //Enable GNSS.
OK
AT+QAGPS=1 //If EPO assistance is disabled, enable it first and
OK then perform the following procedures.

//Query the EPO data file status through AT+QAGPSDATA?, if the file is valid, the current file will be
used; if it is invalid, perform the following procedures.
//You can download the EPO data file to PC (or MCU) from URLs listed in Chapter 1.4.
AT+QAGPSDATA? //Query the EPO data file status.
+QAGPSDATA: 0,0,1980/01/05,23:59:42
+QAGPSDATA: 1,0,1980/01/05,23:59:42
+QAGPSDATA: 2,0,1980/01/05,23:59:42
+QAGPSDATA: 3,0,1980/01/05,23:59:42

OK
AT+QFUPL="EPO_GPS_3_1.DAT",27648 //Upload the EPO file to the module through
//QCOM tool. For more details about QCOM tool,
//see document \[1\].

OK
AT+QAGPSLOC=31.8216915130615,117.115325927734 //Send reference location to GNSS engine.
OK
AT+QAGPSTIME=0,"2023/07/22,08:30:30",1,1,3500 //Inject EPO time to GNSS engine.
OK
AT+QAGPSDATA="EPO_GPS_3_1.DAT" //Inject EPO data file.
OK
AT+QFDEL="EPO_GPS_3_1.DAT" //Delete the EPO data file from UFS.
OK

```

NOTE

When the module is powered down and rebooted, the EPO time needs to be re-injected. The EPO data file is saved in the flash memory of the GNSS engine and does not need to be injected again if it is still valid.

2.5. Summary of Error Codes

<errcode> indicates an error related to GNSS operation. The details about <errcode> are described in the following table.

Table 3: Summary of Error Codes

<errcode>	Description
501	Invalid parameter(s)
502	Operation not supported
503	GNSS subsystem busy
504	Session is ongoing
505	Session not active
506	Operation timeout
507	Function not enabled
508	Time information error
509	EPO not enabled
510	EPO file opening failed
512	Validity time is out of range
513	Internal resource error
514	GNSS locked
515	End by E911
516	Not fixed currently
517	CMUX port is not open
549	Unknown error

3 PAIR Message

This chapter explains the PAIR messages (proprietary NMEA messages defined by the chipset supplier) supported by EG915Q series and EG916Q-GL modules. The PAIR messages can be used when all-in-one solution is enabled by encapsulating them into AT commands.

PAIR Message Format:

```
$PAIR<PacketType>[,<Data>]*<Checksum><CR><LF>
```

Field	Description
\$	Start of NMEA message.
<PacketType>	Three-byte character string, from 000 to 999. An identifier for each PAIR message.
<Data>	This field can be omitted, or multiple fields can be delimited by a data field delimiter ",". Data type varies in different messages.
<Checksum>	Checksum. Calculated by performing an XOR operation on all characters between "\$" and "*" (excluding "\$" and "*"), represented in hexadecimal.
<CR><LF>	End of the NMEA message.

3.1. PAIR001 PAIR_ACK

Acknowledges a PAIR message. **\$PAIR001** is returned to inform the sender that the receiver has received the packet.

Type:

Output

Synopsis:

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

Parameter:

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

Example:

```
$PAIR001,410,4*3A
```

3.2. PAIR068 PAIR_COMMON_SET_HDOP_THRESHOLD

Sets the HDOP threshold. If the HDOP value is larger than this threshold value, the position will not be fixed.

Type:

Set

Synopsis:

```
$PAIR068,<HDOPThreshold>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<HDOPThreshold>	Numeric	-	HDOP threshold.
			0 Disable setting the HDOP threshold
			Other value Set the HDOP threshold

Result:

Returns **\$PAIR001** message.

Example:

```
$PAIR068,0.8*3E
$PAIR001,068,0*35
```

3.3. PAIR069 PAIR_COMMON_GET_HDOP_THRESHOLD

Gets the HDOP threshold.

Type:

Get

Synopsis:

```
$PAIR069*<Checksum><CR><LF>
```

Parameter:

None

Result:

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

Query result message format:

```
$PAIR069,<HDOPThreshold>*<Checksum><CR><LF>
```

Parameter included in the result:

Field	Format	Unit	Description
<HDOPThreshold>	Numeric	-	HDOP threshold.
			<u>0</u> Disable setting the HDOP threshold
			Other value HDOP threshold

Example:

```
$PAIR069*35
$PAIR001,069,0*34
$PAIR069,0.8*3F
```

3.4. PAIR074 PAIR_COMMON_SET_AIC_ENABLE

Enables or disables the active interference cancellation (AIC) function.

Type:

Set

Synopsis:

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

Parameter:

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

Result:

Returns \$PAIR001 message.

Example:

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

3.5. PAIR075 PAIR_COMMON_GET_AIC_STATUS

Gets the status of AIC function.

Type:

Get

Synopsis:

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

Parameter:

None

Result:

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

Query result message format:

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

Parameter included in the result:

Field	Format	Unit	Description
<Status>	Numeric	-	AIC function status. 0 AIC is disabled <u>1</u> AIC is enabled

Example:

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

3.6. PAIR080 PAIR_COMMON_SET_NAVIGATION_MODE

Sets navigation mode.

Type:

Set

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<NavMode>	Numeric	-	Navigation mode. 0 Normal mode <u>1</u> Fitness mode 2 Reserved 3 Balloon mode 4 Stationary mode 5 Drone mode

6	Reserved
7	Swimming mode

Result:

Returns **\$PAIR001** message.

Example:

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

NOTE

1. Normal mode: For general purposes.
2. Fitness mode: For running and walking activities. It ensures more accurate position calculations for low-speed movements (< 5 m/s).
3. Balloon mode: For high-altitude balloon purposes. It ensures more accurate position calculations for vertical movements.
4. Stationary mode: For stationary applications where a zero dynamics is assumed.
5. Drone mode: For drone applications with equivalent dynamics range and vertical acceleration at different flight phases (e.g., hovering, cruising). (Note: The **<latitude>/<longitude>** in NMEA sentences will be correct to 7 decimal places, and **<altitude>** to 3 decimal places automatically.)
6. Swimming mode: For swimming purposes. It smooths the trajectory and improves the accuracy of distance calculation.
7. This message fails in the following situations:
 - If ULP is enabled, it fails to set to one of the other 5 modes except the fitness mode.
 - If the current RF LNA gain is "High gain", only normal mode and drone mode can be configured. Detailed failure information will be displayed in system logs.

3.7. PAIR081 PAIR_COMMON_GET_NAVIGATION_MODE

Gets navigation mode.

Type:

Get

Synopsis:

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

Parameter:

None

Result:

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

Query result message format:

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

Parameter included in the result:

Field	Format	Unit	Description
<NavMode>	Numeric	-	Navigation mode.
			0 Normal mode
			1 Fitness mode
			2 Reserved
			3 Balloon mode
			4 Stationary mode
			5 Drone mode
			6 Reserved
			7 Swimming mode

Example:

```
$PAIR081*33
$PAIR001,081,0*32
$PAIR081,1*2E //The current mode is fitness mode.
```

3.8. PAIR391 PAIR_TEST_JAMMING_DETECT

Enables or disables jamming detection function. The jamming status messages will be transmitted over the GNSS UART port once jamming detection feature is enabled.

Type:

Set

Synopsis:

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

Parameter:

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

Result:

Returns **\$PAIR001** message and periodical **\$PAIRSPF** messages output periodically. For details about **\$PAIRSPF** message, see **Chapter 3.46**.

Example:

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

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

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

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

NOTE

- Once the jamming detection feature is enabled, the module can detect an external jamming interference signal.
 - If there is no jamming, **\$PAIRSPF,1*52** is reported to indicate good status.
 - In case of continuous jamming, the jamming status changes from 1 to 2 and finally to 3.
 - If the jamming detection feature is enabled while the module is still in the acquisition stage, the jamming status is 1. If jamming is detected, the jamming status changes to 2. The module keeps attempting to acquire a valid position during this process; if the anti-jamming repair fails, the jamming status changes to 3.

- 2) After a successful position fix: jamming status is 1 right after jamming detection is enabled, and changes to 2 and 3 consecutively when jamming is detected.
2. This message is currently supported in stand-alone solution.

3.9. PAIR400 PAIR_DGPS_SET_MODE

Sets the DGPS correction data source.

Type:

Set

Synopsis:

```
$PAIR400,<Mode>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Mode>	Numeric	-	DGPS correction data source.
			<u>0</u> No DGPS correction data source
			1 RTCM
			2 SBAS (including WAAS/EGNOS/GAGAN/MSAS)
			3 SLAS

Result:

Returns **\$PAIR001** message.

Example:

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

3.10. PAIR401 PAIR_DGPS_GET_MODE

Gets the DGPS correction data source.

Type:

Get

Synopsis:

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

Parameter:

None

Result:

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

Query result message format:

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

Parameter included in the result:

Field	Format	Unit	Description
<Mode>	Numeric	-	DGPS correction data source.
			0 No DGPS source
			1 RTCM
			2 SBAS (Include WAAS/EGNOS/GAGAN/MSAS)
			3 SLAS

Example:

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

3.11. PAIR410 PAIR_SBAS_ENABLE

Enables or disables SBAS satellite searching.

Type:

Set

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<Enable>	Numeric	-	Enable or disable SBAS satellite searching. 0 Disable <u>1</u> Enable

Result:

Returns **\$PAIR001** message.

Example:

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

NOTE

SBAS is not supported in fitness or swimming navigation mode.

3.12. PAIR411 PAIR_SBAS_GET_STATUS

Gets the status of SBAS satellite search.

Type:

Get

Synopsis:

```
$PAIR411*<Checksum><CR><LF>
```

Parameter:

None

Result:

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

Query result message format:

```
$PAIR411,<Enabled>*<Checksum><CR><LF>
```

Parameter included in the result:

Field	Format	Unit	Description
<Enabled>	Numeric	-	SBAS satellite search status. 0 Disabled <u>1</u> Enabled

Example:

```
$PAIR411*3E
$PAIR001,411,0*3F
$PAIR411,1*23
```

3.13. PAIR412 PAIR_SBAS_GET_SAT_INFO

Gets SBAS satellite information, such as satellite ID, SNR, azimuth, and elevation.

Type:

Get

Synopsis:

```
$PAIR412*<Checksum><CR><LF>
```

Parameter:

None

Result:

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

Query result message format:

```
$PAIR412,<SVid>,<SNR>,<Azim>,<Elev>*<Checksum><CR><LF>
```

Parameter included in the result:

Field	Format	Unit	Description
<SVid>	Numeric	-	Satellite identifier, used to distinguish different satellites
<SNR>	Numeric	-	Signal-to-noise ratio of satellite signal

<Azim>	Numeric	-	Azimuth
<Elev>	Numeric	-	Elevation

Example:

```
$PAIR412*3D
$PAIR001,412,0*3C
$PAIR412,50,42,134,50*0D
```

3.14. PAIR420 PAIR_SLAS_ENABLE

Enables or disables the QZSS SLAS operation.

Type:

Set

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<Enable>	Numeric	-	Enable or disable the QZSS SLAS operation. 0 Disable 1 Enable

Result:

Returns \$PAIR001 message.

Example:

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

3.15. PAIR421 PAIR_SLAS_GET_STATUS

Gets the status of QZSS SLAS operation.

Type:

Get

Synopsis:

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

Parameter:

None

Result:

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

Query result message format:

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

Parameter included in the result:

Field	Format	Unit	Description
<Enabled>	Numeric	-	Status of QZSS SLAS operation. 0 Disabled 1 Enabled

Example:

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

3.16. PAIR430 PAIR_RTCM_SET_INPUT_VERSION

Sets RTCM version for incoming correction data.

Type:

Set

Synopsis:

```
$PAIR430,<Version>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Version>	Numeric	-	RTCM version for incoming correction data. <div> <div>0</div> <div>1</div> </div> <div> <div>RTCM V2.x</div> <div>RTCM V3.x</div> </div>

Result:

Returns **\$PAIR001** message.

Example:

```
$PAIR430,0*21
$PAIR001,430,0*3C
```

3.17. PAIR431 PAIR_RTCM_GET_INPUT_VERSION

Gets RTCM version of incoming correction data.

Type:

Get

Synopsis:

```
$PAIR431*<Checksum><CR><LF>
```

Parameter:

None

Result:

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

Query result message format:

```
$PAIR431,<Version>*<Checksum><CR><LF>
```

Parameter included in the result:

Field	Format	Unit	Description
<Version>	Numeric	-	RTCM version of incoming correction data. 0 RTCM V2.x 1 RTCM V3.x

Example:

```
$PAIR431*3C
$PAIR001,431,0*3D
$PAIR431,0*20
```

3.18. PAIR432 PAIR_RTCM_SET_OUTPUT_MODE

Sets the RTCM output mode.

Type:

Set

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<Mode>	Numeric	-	RTCM output mode (MSM4/MSM7). -1 Output None 0 Output RTCM V3.x with message type MSM4 1 Output RTCM V3.x with message type MSM7

Result:

Returns \$PAIR001 message.

Example:

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

3.19. PAIR433 PAIR_RTCM_GET_OUTPUT_MODE

Gets the RTCM output mode.

Type:

Get

Synopsis:

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

Parameter:

None

Result:

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

Query result message format:

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

Parameter included in the result:

Field	Format	Unit	Description
<Mode>	Numeric	-	RTCM output mode (MSM4/MSM7). <u>-1</u> Output None 0 Output RTCM V3.x with message type MSM4 1 Output RTCM V3.x with message type MSM7

Example:

```
$PAIR433*3E
$PAIR001,433,0*3F
$PAIR433,0*22
```

3.20. PAIR434 PAIR_RTCM_SET_OUTPUT_ANT_PNT

Enables or disables the output of stationary antenna reference point in RTCM format.

Type:

Set

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<Enable>	Numeric	-	Enable or disable the output of stationary antenna reference point in RTCM format. <div> <div>0</div> <div>Disable</div> </div> <div> <div>1</div> <div>Enable</div> </div>

Result:

Returns **\$PAIR001** message.

Example:

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

3.21. PAIR435 PAIR_RTCM_GET_OUTPUT_ANT_PNT

Gets the status of stationary antenna reference point output in RTCM format.

Type:

Get

Synopsis:

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

Parameter:

None

Result:

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

Query result message format:

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

Parameter included in the result:

Field	Format	Unit	Description
<Enable>	Numeric	-	Status of stationary antenna reference point output. 0 Disabled 1 Enabled

Example:

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

3.22. PAIR436 PAIR_RTCM_SET_OUTPUT_EPHEMERIS

Enables or disables the output of satellite ephemeris in RTCM format.

Type:

Set

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<Enable>	Numeric	-	Enables or disables the output of satellite ephemeris in RTCM format. 0 Disable 1 Enable

Result:

Returns **\$PAIR001** message.

Example:

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

3.23. PAIR437 PAIR_RTCM_GET_OUTPUT_EPHEMERIS

Gets the status of satellite ephemeris output in RTCM format.

Type:

Get

Synopsis:

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

Parameter:

None

Result:

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

Query result message format:

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

Parameter included in the result:

Field	Format	Unit	Description
<Enable>	Numeric	-	Status of satellite ephemeris output in RTCM format.
			<u>0</u> Disabled
			1 Enabled

Example:

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

3.24. PAIR490 PAIR_EASY_ENABLE

Enables or disables EASY function.

Type:

Set

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<Enable>	Numeric	-	Enable or disable EASY function. 0 Disable 1 Enable

Result:

Returns **\$PAIR001** message.

Example:

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

3.25. PAIR491 PAIR_EASY_GET_STATUS

Gets the status of EASY function.

Type:

Get

Synopsis:

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

Parameter:

None

Result:

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

Query result message format:

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

Parameter included in the result:

Field	Description
<Enabled>	EASY function status. 0 Disabled 1 Enabled
<Status>	EASY data extension status. 0 Not finished 1 1-day extension finished 2 2-day extension finished 3 3-day extension finished If <Enabled> is set to 0, <Status> is not displayed in the result.

Example:

//If **<Enabled>** is set to 1:

```
$PAIR491*36
```

```
$PAIR001,491,0*37
```

```
$PAIR491,1,0*37
```

//If **<Enabled>** is set to 0:

```
$PAIR491*36
```

```
$PAIR001,491,0*37
```

```
$PAIR491,0*2A
```

3.26. PAIR530 PAIR_EPH_GET_STATUS

Gets the ephemeris status after a specific time.

Type:

Get

Synopsis:

```
$PAIR530,<Constellation>,<Time_interval>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Constellation>	Numeric	-	GNSS constellation type. 0 GPS 1 GLONASS 2 Galileo 3 BDS 4 QZSS
<Time_interval>	Numeric	Seconds	Time. Range: 1–7200 (2 hours).

Result:

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

Query result message format:

```
$PAIR530,<Constellation>,<L1_SV>*<Checksum><CR><LF>
```

Parameter included in the result:

Field	Format	Unit	Description
<Constellation>	Numeric	-	GNSS constellation type. 0 GPS 1 GLONASS 2 Galileo 3 BDS 4 QZSS
<L1_SV>	Hexadecimal	-	L1 satellite signal

Example:

```
//Get the status of GPS ephemeris in the next 1 second.
```

```
$PAIR530,0,1*3D
```

```
$PAIR001,530,0*3D
```

```
$PAIR530,0,DF508A0E*07
```

```
//HEX DF508A0E corresponds to 1101 1111 0101 0000 1000 1010 0000 1110 and the valid L1 SV numbers are 2, 3, 4, 10, 12, 16, 21, 23, 25, 26, 27, 28, 29, 31, 32.
```

3.27. PAIR531 PAIR_EPH_SET_DATA

Sends ephemeris subframe message to GNSS engine.

Type:

Set

Synopsis:

```
$PAIR531,<Constellation>,<Signal_ID>,<SV_ID>,<EPH_data>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Constellation>	Numeric	-	GNSS constellation type.
			0 GPS
			1 GLONASS
			2 Galileo
			3 BDS
<Signal_ID>	Numeric	-	4 QZSS
			0 L1
			1 L5 (including GPS L5, Galileo E5a and BDS B2a) (Not supported)
<SV_ID>	Decimal	-	Satellite ID.
			GPS 1–32
			GLONASS 1–24
			Galileo 1–36
			BDS 1–63 (GEO: 1–5, 59–63; MEO: 6–58)
<EPH_data>	Numeric	-	QZSS 1–7
			Assistance ephemeris data. Contact Quectel Technical Support for details.
			GPS (L1): W[0],...,W[23]
			GLONASS (L1): W[0],...,W[15]
			Galileo (L1): W[0],...,W[15],(W[16],...,W[18])
<EPH_data>	Numeric	-	BDS GEO (L1): W[0],...,W[29]
			BDS MEO (L1): W[0],...,W[20]
			QZSS (L1): W[0],...,W[23]

Result:

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

Query result message format:

```
$PAIR531,<Status>* <Checksum><CR><LF>
```

Parameter included in the result:

Field	Format	Unit	Description
<Status>	Numeric	-	Message execution result. 0 Fail 1 Success

Example:

```
$PAIR531,0,0,1,025000,2B072D,F3002F,4BBD3E,06510C,488598,00FFAB,FA8C41,48F752,28BC4B,654D79,F88804,937C14,1969A1,0D4B91,85987C,FFA27C,508DD6,000F27,C7053B,133E1D,319E79,FFAC83,481070*59
$PAIR001,531,0*3C
$PAIR531,1*20
```

3.28. PAIR532 PAIR_EPH_GET_DATA

Gets a single ephemeris subframe message.

Type:

Get

Synopsis:

```
$PAIR532,<Constellation>,<Signal_ID>,<SV_ID>* <Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Constellation>	Numeric	-	GNSS constellation type. 0 GPS 1 GLONASS 2 Galileo 3 BDS 4 QZSS
<Signal_ID>	Numeric	-	0 L1 1 L5 (including GPS L5, Galileo E5a,

BDS B2a) (Not supported)			
<SV_ID>	Decimal	-	Satellite ID.
			GPS 1–32
			GLONASS 1–24
			Galileo 1–36
			BDS 1–63 (GEO: 1-5, 59-63; MEO: 6-58)
			QZSS 1–7

Result:

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

Query result message format:

```
$PAIR532,<Constellation>,<Signal_ID>,<SV_ID>,<EPH_data>*<Checksum><CR><LF>
```

Parameter included in the result:

Field	Format	Unit	Description
<Constellation>	Numeric	-	GNSS constellation type.
			0 GPS
			1 GLONASS
			2 Galileo
			3 BDS
			4 QZSS
<Signal_ID>	Numeric	-	0 L1
			1 L5 (including GPS L5, Galileo E5a, BDS B2a) (Not supported)
<SV_ID>	Decimal	-	Satellite ID.
			GPS 1–32
			GLONASS 1–24
			Galileo 1–36
			BDS 1–63 (GEO: 1–5, 59–63; MEO: 6–58)
			QZSS 1–7
<EPH_data>	Numeric	-	Refer to the format of PAIR531 . See Chapter 3.27 for details.

Example:

```
$PAIR532,0,0,1*23
```

```
$PAIR001,532,0*3F
```

```
$PAIR532,0,0,01,025000,2B072D,F3002F,4BBD3E,06510C,488598,00FFAB,FA8C41,48F752,28BC4
```

```
B,654D79,F88804,937C14,1969A1,0D4B91,85987C,FFA27C,508DD6,000F27,C7053B,133E1D,319E79,FFAC83,481070*6A
```

3.29. PAIR533 PAIR_EPH_CLEAR

Clears the ephemeris data from the critical memory area.

Type:

Set

Synopsis:

```
$PAIR533*<Checksum><CR><LF>
```

Parameter:

None

Result:

Returns **\$PAIR001** message.

Example:

```
$PAIR533*3F
$PAIR001,533,0*3E
```

NOTE

The ephemeris data is invalidated upon a reboot or power-off of the module or GNSS engine.

3.30. PAIR534 PAIR_EPH_NOTIFY_ENABLE

Enables or disables ephemeris update notification.

Type:

Set

Synopsis:

```
$PAIR534,<Enable>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Enable>	Numeric	-	Enable or disable ephemeris update notification. 0 Disable 1 Enable

Result:

Returns **\$PAIR001** message.

Example:

```
$PAIR534,1*25
$PAIR001,534,0*39
```

3.31. PAIR535 PAIR_EPH_NOTIFY

Notifies an ephemeris update to GNSS engine.

Type:

Output

Synopsis:

```
$PAIR535,<Constellation>,<Signal_ID>,<SV>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Constellation>	Numeric	-	GNSS constellation ID. 0 GPS 1 GLONASS 2 Galileo 3 BDS 4 QZSS
<Signal_ID>	Numeric	-	0 L1

			1	L5 (including GPS L5, Galileo E5a, BDS B2a) (Not supported)
<SV>	Hexadecimal	-		64-bit bitmap indicating the SVs whose ephemeris data have been updated.

Example:

```
//Ephemeris of GPS L1 PRN20 and PRN29 is updated.
$PAIR535,0,0,0000000010080000*1C
```

NOTE

This message is automatically sent by the GNSS system when ephemeris update notification is enabled. Do not actively send it to the GNSS system.

3.32. PAIR650 PAIR_LOW_POWER_ENTRY_RTC_MODE

Shut downs GNSS engine and sets it to RTC mode, after which it cannot receive any messages. The GNSS engine can only be reactivated by a timer. Upon waking up, all system resources will re-initialize.

Type:

Set

Synopsis:

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

Parameter:

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

Result:

- If successful, the module enters RTC mode and cannot receive any messages.
- If failed, the **\$PAIR001** message is returned.

Example:

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

3.33. PAIR680 PAIR_GLP_ENABLE

Enables or disables GPS Low Power (GLP) mode which adjusts the duty cycle to ensure good performance and low power consumption of the module in different signal conditions.

Type:

Set

Synopsis:

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

Parameter:

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

Result:

Returns **\$PAIR001** message.

Example:

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

NOTE

- The requirements for entering GLP mode are as follows:
 - Fix rate: 1 Hz.
 - Satellite constellation configuration: GPS only.
 - Navigation mode: fitness mode.
- When the GLP mode is enabled, some of the features will be disabled automatically, such as SBAS, ALP, FLP, and the Periodic Power Saving mode.

3.34. PAIR681 PAIR_GLP_GET_STATUS

Gets GLP mode status.

Type:

Get

Synopsis:

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

Parameter:

None

Result:

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

Query result message format:

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

Parameter included in the result:

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

Example:

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

3.35. PAIR690 PAIR_PERIODIC_SET_MODE

Configures Periodic Power Saving mode setting. Periodic Power Saving mode has two stages (Run stage and Sleep stage), and the module periodically transitions from one stage to the other and vice versa depending on the setting. In Run stage, the GNSS engine measures and calculates the position. In

Sleep stage, the GNSS engine is in power saving mode.

Type:

Set

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<Mode>	Numeric	-	Periodic Power Saving mode setting. 0 Disable 1 Enable smart periodic mode 2 Enable strict periodic mode
<FirstRun>	Numeric	Second	Running time. Range: 3–518400.
<FirstSleep>	Numeric	Second	Sleeping time. Range: 3–518400.
<SecondRun>	Numeric	Second	Second running time. Range: 0 or 3–518400.
<SecondSleep>	Numeric	Second	Second sleeping time. Range: 0 or 3–518400.

Result:

Returns **\$PAIR001** message.

Example:

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

NOTE

1. In smart periodic mode, the GNSS system dynamically increases running time to acquire more navigation data. In strict periodic mode, the GNSS system enters Power Saving mode periodically.
2. **<FirstRun>**: Interval between exiting Sleep stage and getting a new position fix.
3. **<FirstSleep>**: Duration of Sleep stage after getting a fix (or attempting to get a fix).
4. **<SecondRun>**: GNSS module uses **<SecondRun>** instead of **<FirstRun>** setting when there is no signal. The second running time can be "0" only when the second sleeping time is "0".
5. **<SecondSleep>**: GNSS module uses **<SecondSleep>** instead of **<FirstSleep>** setting when there is no signal. The second sleeping time can be "0" only when the second running time is "0".

3.36. PAIR691 PAIR_PERIODIC_GET_MODE

Gets Periodic Power Saving mode configurations.

Type:

Get

Synopsis:

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

Parameter:

None

Result:

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

Query result message format:

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

Parameter included in the result:

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

Example:

```
$PAIR691*34
$PAIR001,691,1*34
$PAIR691,0,21,39,48,72*28
```

3.37. PAIR720 PAIR_DTM_ENABLE

Enables/Disables duty tracking mode (DTM). DTM is a power-saving mechanism that adjusts the duty cycle to ensure good performance and low power consumption in different environments.

Type:

Set

Synopsis:

```
$PAIR720,<Enable>,<Update_Period>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Enable>	Numeric	-	Enable or disable DTM. 0 Disable 1 Enable
<Update_Period>	Numeric	Second	Position update period. Value: 2, 5, 10, 20, 30, 60.

Result:

Returns **\$PAIR001** message.

Example:

```
$PAIR720,1,2*23
$PAIR001,720,0*3E
```

NOTE

When DTM is disabled, the value of **<Update_Period>** is 1.

3.38. PAIR721 PAIR_DTM_GET_STATUS

Gets DTM status and configuration.

Type:

Get

Synopsis:

```
$PAIR721*<Checksum><CR><LF>
```

Parameter:

None

Result:

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

Query result message format:

```
$PAIR721,<Enable>,<Update_Period>*<Checksum><CR><LF>
```

Parameter included in the result:

Field	Format	Unit	Description
<Enable>	Numeric	-	DTM status. 0 Disabled 1 Enabled
<Update_Period>	Numeric	Second	Position update period. Value: 2, 5, 10, 20, 30, 60.

Example:

```
$PAIR721*3E
$PAIR001,721,0*3F
$PAIR721,1,2*3D
```

3.39. PAIR730 PAIR_FLP_ENABLE

Enables or disables Fitness Low Power (FLP) mode. When the FLP mode is enabled, some features will be automatically disabled, such as SBAS, Periodic Power Saving mode, GLP mode, and ALP mode.

Type:

Set

Synopsis:

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

Parameter:

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

Result:

Returns **\$PAIR001** message.

Example:

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

NOTE

The requirements for entering FLP mode are as follows:

1. Fix rate: 1 Hz.
2. Navigation mode: fitness mode.
3. The supported satellite constellation configurations are as follows:
 - 1) GPS + GLONASS + Galileo + BDS (+ QZSS)
 - 2) GPS + GLONASS (+ QZSS)
 - 3) GPS+ BDS (+ QZSS)

3.40. PAIR731 PAIR_FLP_GET_STATUS

Gets FLP mode status.

Type:

Get

Synopsis:

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

Parameter:

None

Result:

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

Query result message format:

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

Parameter included in the result:

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

Example:

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

3.41. PAIR732 PAIR_ALP_ENABLE

Enables or disables Adaptive Low Power (ALP) mode, which ensures good performance and low power consumption in different environments.

Type:

Set

Synopsis:

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

Parameter:

Field	Format	Unit	Description
<Enabled>	Numeric	-	Enable or disable ALP mode. <u>0</u> Disable

1 Enable

Result:

Returns **\$PAIR001** message.

Example:

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

NOTE

1. The requirements for entering ALP mode are as follows:
 - 1) Fix rate: 1 Hz.
 - 2) Navigation mode: normal mode.
 - 3) The supported satellite constellation configurations are as follows:
 - GPS + GLONASS + Galileo + BDS (+ QZSS)
 - GPS + GLONASS (+ QZSS)
2. When the ALP mode is enabled, some of the features will be automatically disabled, such as SBAS, Periodic Power Saving mode, FLP mode, and GLP mode.

3.42. PAIR733 PAIR_ALP_GET_STATUS

Gets ALP mode status.

Type:

Get

Synopsis:

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

Parameter:

None

Result:

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

Query result message format:

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

Parameter included in the result:

Field	Format	Unit	Description
			ALP mode status.
<Enabled>	Numeric	-	0 Disabled
			1 Enabled

Example:

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

NOTE

The requirements for querying ALP mode are as follows:

1. Fix rate: 1 Hz.
2. Navigation mode: normal mode.

3.43. PAIR752 PAIR_PPS_SET_CONFIG_CMD

Configures the Pulse Per Second (PPS) settings.

Type:

Set

Synopsis:

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

Parameter:

Field	Format	Unit	Description
			PPS settings.
<PPSType>	Numeric	-	0 Disable
			1 After the first fix

			2	3D fix only
			3	2D/3D fix only
			4	Always
<PPSPulseWidth>	Numeric	Millisecond	PPS pulse width. Range: 1–999.	

Result:

Returns **\$PAIR001** message.

Example:

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

3.44. PAIR890 PAIR_GEOFENCE_SET_CONFIG

Sets geofencing configuration.

Type:

Set

Synopsis:

```
$PAIR890,<FenceNum>,<ConfLvl>,<Lat1>,<Lon1>,<Rad1>,...,<RadN>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Unit	Description
<FenceNum>	Numeric	-	Geofence count. Range: 0–4. When the value is set to 0, the geofencing function is disabled.
<ConfLvl>	Numeric	-	Confidence level for state evaluation. 0 No requirement 1 1-Sigma (68 %) 2 2-Sigma (95 %) 3 3-Sigma (99.7 %) 4 4-Sigma (99.9999 %) 5 5-Sigma (99.999999 %)
<Lat>	Numeric	Degree	Latitude of geofence circle center.
<Lon>	Numeric	Degree	Longitude of geofence circle center.

<Rad>	Numeric	Meter	Radius of geofence circle.
-------	---------	-------	----------------------------

Result:

- When <FenceNum> = 0, the module returns \$PAIR001 message.
- When <FenceNum> ≠ 0, the module returns \$PAIR001 message and data in hexadecimal format. Hexadecimal data are used in the following example.

Data field

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

Message format

Preamble	Message ID	Length	Data	Checksum	End Word
0x04 0x24	0xDD 0x07	0x0C 0x00	See <u>Payload</u>	0x**	0xAA 0x44

Payload

Byte Offset	Length (byte)	Name	Unit	Description
0	1	Status	-	Geofencing status. 0 Geofencing disabled or not reliable 1 Geofencing enabled
1	1	Fencenum	-	Geofence count.
2	1	State	-	Combined state of all geofences. 0 Outside 1 Unknown 2 Inside
3	4	EachState[4]	-	Geofence state. 0 Outside 1 Unknown

				2	Inside
7	1	Hour	-	Hour of day. Range: 0–23 (UTC).	
8	1	Min	-	Minute of hour. Range: 0–59 (UTC).	
9	1	Sec	-	Second of hour. Range: 0–59 (UTC).	
10	2	Msec	-	Millisecond of second. Range: 0–999 (UTC).	

Example:

```
//Enable the geofencing function.
$PAIR890,1,1,25.0567,121.5743,30*20
$PAIR001,890,0*3A
04 24 DD 07 0C 00 01 01 00 00 00 00 00 07 14 33 00 00 F6 AA 44

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

NOTE

When **<FenceNum>** = 0, the other parameters should be omitted in **\$PAIR890**. When **<FenceNum>** ≠ 0, parameters in **\$PAIR890** cannot be omitted.

3.45. PAIR891 PAIR_GEOFENCE_GET_CONFIG

Gets geofencing configuration.

Type:

Get

Synopsis:

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

Parameter:

None

Result:

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

Query result message format:

```
$PAIR891,<FenceNum>,<ConfLvl>,<Lat1>,<Lon1>,<Rad1>,...,<RadN>*<Checksum><CR><LF>
```

Parameter included in the result:

Field	Format	Unit	Description
<FenceNum>	Numeric	-	Geofence count. Range: 0–4. When the value is set to 0, the geofencing function is disabled.
<ConfLvl>	Numeric	-	Confidence level for state evaluation. 0 No requirement 1 1-Sigma (68 %) 2 2-Sigma (95 %) 3 3-Sigma (99.7 %) 4 4-Sigma (99.9999 %) 5 5-Sigma (99.999999 %)
<Lat>	Numeric	Degree	Latitude of geofence circle center.
<Lon>	Numeric	Degree	Longitude of geofence circle center.
<Rad>	Numeric	Meter	Radius of geofence circle.

Example:

```
$PAIR891*3A
$PAIR001,891,0*3B
$PAIR891,1,1,25.0567,121.5743,30*21
```

3.46. PAIRSPF

Outputs jamming status when jamming detection function is enabled.

Type:

Output

Synopsis:

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

Parameter:

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

Example:

```
//Unknown status.
```

```
$PAIRSPF,0*53
```

```
//No jamming, good status.
```

```
$PAIRSPF,1*52
```

```
//Warning status.
```

```
$PAIRSPF,2*51
```

```
//Critical status.
```

```
$PAIRSPF,3*50
```

4 PQ Message

This chapter outlines PQ messages which are used to control and configure the internal GNSS engine of EG915Q series and EG916Q-GL modules. The PQ messages can be used when all-in-one solution is enabled by encapsulating them into AT commands.

PQ Message Format:

```
$<Command>[,<Data>]*<Checksum><CR><LF>
```

Field	Description
\$	Start of sentence.
<Command>	PQ message character string.
<Data>	This field can be omitted, or multiple fields can be delimited by a data field delimiter “,”. Data type varies in different messages.
<Checksum>	Checksum. Calculated by performing XOR operation on all characters between "\$" and "*" (excluding "\$" and "*"), represented in hexadecimal.
<CR><LF>	End of sentence.

4.1. PQBAUD

Changes the default baud rate of NMEA port.

Type:

Set

Synopsis:

```
$PQBAUD,W,<baudrate>*Checksum<CR><LF>
```


Parameter:

Field	Format	Unit	Description
<baudrate>	Numeric	bps	<p>NMEA port baud rate, which can be set to the following values:</p> <p>4800</p> <p>9600</p> <p>14400</p> <p>19200</p> <p>38400</p> <p>57600</p> <p>115200</p> <p><u>921600</u></p>

Result:

- If successful, the module returns:

```
$PQBAUD,W,OK*Checksum<CR><LF>
```

- If failed, the module returns:

```
$PQBAUD,W,ERROR*Checksum<CR><LF>
```

Example:

```
//Change NMEA port default baud rate to 38400 bps.
```

```
$PQBAUD,W,38400*7B
```

```
$PQBAUD,W,OK*40
```

NOTE

1. If the baud rate of the module is changed, please modify the baud rate of the client to the same value to achieve normal communication.
2. The message takes effect immediately. The configuration is saved automatically.
3. Currently, when all-in-one solution is enabled, the module only supports NMEA sentence output at a baud rate of 921600 bps. In contrast, when the stand-alone solution is enabled, the module supports NMEA sentence output at any configurable baud rate. Therefore, if a baud rate other than 921600 bps is used when stand-alone solution is enabled, the baud rate must be reset to 921600 bps before switching to all-in-one solution.

4.2. PQEPE

Enables or disables the output of URC including EPE data.

Type:

Set

Synopsis:

```
$PQEPE,W,<mode>,<save>*Checksum<CR><LF>
```

Parameter:

Field	Format	Unit	Description
<mode>	Numeric	-	Enable/Disable the output of URC including EPE data.
			0 Disable the output of URC including EPE data
			1 Enable the output of URC including EPE data
<save>	Numeric	-	Save operation.
			0 Parameters are not saved, and ineffective after the module is rebooted
			1 Parameters are saved in flash, and effective after the module is rebooted

Result:

- If successful, the module returns:

```
$PQEPE,W,OK*Checksum<CR><LF>
```

```
[$PQEPE,<EPE_hori>,<EPE_vert>*Checksum]
```

- If failed, the module returns:

```
$PQEPE,W,ERROR*Checksum<CR><LF>
```

URC format:

```
$PQEPE,<EPE_hori>,<EPE_vert>*Checksum
```

Parameter included in the URC:

Field	Format	Unit	Description
<EPE_hori>	Numeric	-	Estimated horizontal position error.
<EPE_vert>	Numeric	-	Estimated vertical position error.

Example:

```
$PQEPE,W,1,1*2A //Enable the output of URC including EPE data, and save parameters into flash.
$PQEPE,W,OK*02
$PQEPE,5.3050,3.2000*53
```

NOTE

1. The message takes effect immediately.
2. The URC is reported when **<mode>** is set to 1.

4.3. PQECEP

Enables or disables PQECEPPOSVEL sentence output, or gets the status of PQECEPPOSVEL sentence output.

Type:

Set/Get

Synopsis:

```
//Set
$PQECEP,W,<mode>,<save>*Checksum<CR><LF>
//Get
$PQECEP,R*Checksum<CR><LF>
```

Parameter:

Field	Format	Unit	Description
<mode>	Numeric	-	Enable or disable PQECEPPOSVEL sentence output. 0 Disable 1 Enable

			Save operation.
<save>	Numeric	-	0 Parameters are not saved, and ineffective after the module is rebooted
			1 Parameters are saved in flash, and effective after the module is rebooted

Result:

- If successful, the module returns:

```
//Set
$PQCECF,W,OK*Checksum<CR><LF>

[$PQCECFPOSVEL,<time>,<x>,<y>,<z>,<v_x>,<v_y>,<v_z>*Checksum<CR><LF>]
//Get
$PQCECF,R,<mode>*Checksum<CR><LF>
```

- If failed, the module returns:

```
$PQCECF,W,ERROR*Checksum<CR><LF>
```

Parameter included in the result:

Field	Format	Unit	Description
<mode>	Numeric	-	PQCECFPOSVEL sentence output status.
			0 Disabled
			1 Enabled

URC format:

```
$PQCECFPOSVEL,<time>,<x>,<y>,<z>,<v_x>,<v_y>,<v_z>*Checksum<CR><LF>
```

Parameter included in the URC:

Field	Format	Unit	Description
<time>	Numeric	-	Time from the internal real-time clock.
<x>	Numeric	-	Value of X axis in ECEF.
<y>	Numeric	-	Value of Y axis in ECEF.
<z>	Numeric	-	Value of Z axis in ECEF.
<v_x>	Numeric	-	Velocity component of X axis in ECEF.

<v_y>	Numeric	-	Velocity component of Y axis in ECEF.
<v_z>	Numeric	-	Velocity component of Z axis in ECEF.

Example:

```

$PQCECF,W,1,1*7F //Enable PQCECFPOSVEL sentence output, and save parameters into
                  flash.
$PQCECF,W,OK*57

$PQCECFPOSVEL,052743.000,-1526672.867459,6191083.982801,143008.780911,0,0,0*14
$PQCECF,R*7A      //Get status of PQCECFPOSVEL sentence output.
$PQCECF,R,1*67
    
```

NOTE

1. The message takes effect immediately.
2. The URC is reported when **<mode>** is set to 1.

4.4. PQODO

Starts or stops odometer reading; or gets current odometer reading mode; or queries the current distance.

Type:

Set/Get/Query

Synopsis:

```

//Set
$PQODO,W,<mode>,<initial_distance>*Checksum<CR><LF>
//Get current odometer reading mode.
$PQODO,R*Checksum<CR><LF>
//Query the current distance.
$PQODO,Q*Checksum
    
```

Parameter:

Field	Format	Unit	Description
<mode>	Numeric	-	Start or stop odometer reading. 0 Stop odometer reading and record the distance value

- 1 Start odometer reading and initialize the distance according to **<initial_distance>**

<initial_distance>	Format	Unit	Description
Numeric	Meter	Initial distance. Default: 0 m.	

Result:

- If successful, the module returns:

```
//Set
$PQODO,W,OK*Checksum<CR><LF>
//Get
$PQODO,R,<mode>*Checksum<CR><LF>
//Query
$PQODO,Q,<distance>*Checksum
```

- If failed, the module returns:

```
//Set
$PQODO,W,ERROR*Checksum<CR><LF>
//Get
$PQODO,R,ERROR*Checksum<CR><LF>
//Query
$PQODO,Q,ERROR*Checksum<CR><LF>
```

Parameter included in the result:

Field	Format	Unit	Description
<distance>	Numeric	Meter	Current distance. It is returned when the Query Command is successfully executed.

Example:

```
$PQODO,W,1*23 //Start odometer reading, and the initial distance is 0 m.
$PQODO,W,OK*16
$PQODO,W,1,1000000*3E //Start odometer reading, and set the initial distance to 1000000 m.
$PQODO,W,OK*16
$PQODO,R*3B //Get current odometer reading mode .
$PQODO,R,1*26
$PQODO,Q*38 //Query the current distance.
$PQODO,Q,123.45*0B
```

NOTE

1. The message takes effect immediately.
2. After the module is rebooted, the **PQODO** should be executed again to re-read the odometer

value.

3. When the odometer reading function is turned off and turned on again, the odometer will be read from the **<initial_distance>** set last time.

4.5. PQPZ90

Enables or disables switching from WGS84 to PZ-90.11; or get the status of switching from the WGS84 to PZ-90.11.

Type:

Set/Get

Synopsis:

```
//Set
$PQPZ90,W,<mode>,<save>*Checksum<CR><LF>
//Get
$PQPZ90,R*Checksum<CR><LF>
```

Parameter:

Field	Format	Unit	Description
<mode>	Numeric	-	Enable or disable switching from WGS84 to PZ-90.11. 0 Disable 1 Enable
<save>	Numeric	-	Save operation. 0 Parameters are not saved, and ineffective after the module is rebooted 1 Parameters are saved in flash, and effective after the module is rebooted

Result:

- If successful, the module returns:

```
//Set
$PQPZ90,W,OK*Checksum<CR><LF>

[$--DTM,xxx,,l.lllll,a,y.yyyyyy,a,x.xx,W84*hh<CR><LF>]
//Get
$PQPZ90,R,<mode>*Checksum<CR><LF>
```

- If failed, the module returns:

```
$PQPZ90,W,ERROR*Checksum<CR><LF>
```

Parameter included in the result:

Field	Format	Unit	Description
<mode>	Numeric	-	Status of switching from WGS84 to PZ-90.11.
			0 Disabled
			1 Enabled

URC format

```
$--DTM,xxx,,l.lllll,a,y.yyyyyy,a,x.xx,W84*hh<CR><LF>
```

Parameter:

Field	Meaning
--	Talker ID.
xxx	Local datum code. W84 = WGS84 P90 = PZ-90.11
Null field	Sub datum
l.lllll	Latitude offset. Unit: minute.
a	Direction of latitude offset N: North S: South
y.yyyyyy	Longitude offset. Unit: minute.
a	Direction longitude offset E: East W: West
x.xx	Altitude offset. Unit: meter.
W84	Local datum code indicating WGS84. Fixed to W84.
*hh	Checksum

Example:

```
$PQPZ90,W,1,1*79 //Enable switching from WGS84 to PZ-90.11, and save parameters into flash.
$PQPZ90,W,OK*51

$GNDTM,P90,,0.000018,S,0.000088,E,1.04,W84*52
$PQPZ90,R*7C //Get the status of switching from the WGS84 to PZ-90.11.
$PQPZ90,R,0*60
```

NOTE

1. The message takes effect immediately.
2. The URC is reported when **<mode>** is set to 1.

4.6. PQGLP

Sets the module to GLP mode; or gets the current operation mode of the module.

Type:

Set/Get

Synopsis:

```
//Set
$PQGLP,W,<mode>,<save>*Checksum<CR><LF>
//Get
$PQGLP,R*Checksum<CR><LF>
```

Parameter:

Field	Format	Unit	Description
<mode>	Numeric	-	Module operation mode.
			0 Normal mode 1 GLP mode
<save>	Numeric	-	Save operation.
			0 Parameters are not saved, and ineffective after the module is rebooted 1 Parameters are saved in flash, and effective after the module is rebooted

Result:

- If successful, the module returns:

```
//Set
$PQGLP,W,OK*Checksum<CR><LF>
//Get
$PQGLP,R,<mode>*Checksum<CR><LF>
```

- If failed, the module returns:

```
//Set
$PQGLP,W,ERROR*Checksum<CR><LF>
//Get
$PQGLP,R,ERROR*Checksum<CR><LF>
```

Example:

```
$PQGLP,W,1,1*21 //Set GLP mode for module.
$PQGLP,W,OK*09
$PQGLP,R*24 //Get the module operation mode.
$PQGLP,R,1*39
```

NOTE

1. The message takes effect immediately.
2. The GLP mode only supports 1 Hz PVT and fitness mode, and only the GPS L1 frequency band is used.

4.7. PQVEL

Enables or disables the output of velocity data along 3 axes (north, east, and down); or gets the current status of velocity data output.

Type:

Set/Get

Synopsis:

```
//Set
$PQVEL,W,<mode>,<save>*Checksum<CR><LF>
//Get
$PQVEL,R,<mode>*Checksum<CR><LF>
```

Parameter:

Field	Format	Unit	Description
<mode>	Numeric	-	Enable or disable the output of velocity data along 3 axes. 0 Disable 1 Enable
<save>	Numeric	-	Save operation. 0 Parameters are not saved, and ineffective after the module is rebooted 1 Parameters are saved in flash, and effective after the module is rebooted

Result:

- If successful, the module returns:

```
//Set
$PQVEL,W,OK*Checksum<CR><LF>

[$PQVEL,<north_vel>,<east_vel>,<down_vel>*Checksum<CR><LF>]
//Get
$PQVEL,R,<mode>*Checksum<CR><LF>
```

- If failed, the module returns:

```
$PQVEL,W,ERROR*Checksum<CR><LF>
```

Parameter included in the result:

Field	Format	Unit	Description
<mode>	Numeric	-	The current status of velocity data output. 0 Disabled 1 Enabled

URC format:

```
$PQVEL,<north_vel>,<east_vel>,<down_vel>*Checksum<CR><LF>
```

Parameter included in the URC:

Field	Format	Unit	Description
<north_vel>	Numeric	m/s	Velocity along the north axis
<east_vel>	Numeric	m/s	Velocity along the east axis

<down_vel>	Numeric	m/s	Velocity along the down axis
------------	---------	-----	------------------------------

Example:

```
$PQVEL,W,1,1*25 //Enable output of velocity data along three axes.
$PQVEL,W,OK*0D

$PQVEL,-1,0,3*6D
$PQVEL,R*20 //Get the current status of velocity data output.
$PQVEL,R,1*3D
```

NOTE

1. The message takes effect immediately.
2. The URC is reported when **<mode>** is set to 1.

4.8. PQJAM

Enables or disables the jamming detection function, or gets the current status of the jamming detection function.

Type:

Set/Get

Synopsis:

```
//Set
$PQJAM,W,<mode>,<save>*Checksum<CR><LF>
//Get
$PQJAM,R*Checksum<CR><LF>
```

Parameter:

Field	Format	Unit	Description
<mode>	Numeric	-	Enable or disable jamming detection function. 0 Disable 1 Enable
<save>	Numeric	-	Save operation. 0 Parameters are not saved, and ineffective after the module is rebooted 1 Parameters are saved in flash, and effective after the

module is rebooted

Result:

- If successful, the module returns:

```
//Set
$PQJAM,W,OK*Checksum<CR><LF>

[$PAIRSPF,<Status>*Checksum<CR><LF>]
//Get
$PQJAM,R,<mode>*Checksum<CR><LF>
```

- If failed, the module returns:

```
//Set
$PQJAM,W,ERROR*Checksum<CR><LF>
//Get
$PQJAM,R,ERROR*Checksum<CR><LF>
```

Parameter included in the result:

Field	Format	Unit	Description
<mode>	Numeric	-	Current status of jamming detection function. 0 Disabled 1 Enabled

URC format:

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

Parameter included in the URC:

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

Example:

```
$PQJAM,W,1,1*3C //Enable jamming detection function.
$PQJAM,W,OK*14
```

\$PAIRSPF,1*52

\$PQJAM,R*39

//Get the current status of jamming detection function.

\$PQJAM,R,1*24

NOTE

1. The message takes effect immediately.
2. The URC is reported when **<mode>** is set to 1.
3. The message is not supported in RTC mode.

4.9. PQRLM

Enables or disables return link message output; or gets the current status of the return link message output.

Type:

Set/Get

Synopsis:

```
//Set
$PQRLM,W,<mode>,<save>*Checksum<CR><LF>
//Get
$PQRLM,R*Checksum<CR><LF>
```

Parameter:

Field	Format	Unit	Description
<mode>	Numeric	-	Enable or disable return link message output.
			0 Disable 1 Enable
<save>	Numeric	-	Save operation.
			0 Parameters are not saved, and ineffective after the module is rebooted 1 Parameters are saved in flash, and effective after the module is rebooted

Result:

- If successful, the module returns:

```
//Set
$PQRLM,W,OK*Checksum<CR><LF>

[$PQRLM,<beacon_id>,<gps_second>,<msg_code>,<para>*Checksum<CR><LF>]
//Get
$PQRLM,R,<mode>*Checksum<CR><LF>
```

- If failed, the module returns:

```
//Set
$PQRLM,W,ERROR*Checksum<CR><LF>
//Get
$PQRLM,R,ERROR*Checksum<CR><LF>
```

Parameter included in the result:

Field	Format	Unit	Description
<mode>	Numeric	-	The current status of the return link message output. 0 Disabled 1 Enabled

URC format:

```
$PQRLM,<beacon_id>,<gps_second>,<msg_code>,<para>*Checksum<CR><LF>
```

Parameter included in the URC:

Field	Format	Unit	Description
<beacon_id>	Numeric	-	Beacon ID of RLM.
<gps_second>	Numeric	-	UTC timestamp of position.
<msg_code>	Numeric	-	Message code.
<para>	Numeric	-	The data parameters. Short messages contain 4 hex characters; long messages contain 24 hex characters.

Example:

```
$PQRLM,W,1,1*29 //Enable return link message output.
$PQRLM,W,OK*01

$PQRLM,0a0a0a0a0a0a0a8,955065733,1,8aa1*3E
$PQRLM,R*2C //Get the current status of the return link message output.
```

\$PQRLM,R,1*31

NOTE

1. The message takes effect immediately.
2. The URC is reported when **<mode>** is set to 1.

4.10. PQGEO

Configures geo-fence parameters; or gets the geo-fence parameters; or queries the status of the specified geo-fence.

Type:

Set/Get/Query

Synopsis:

```
//Set
$PQGEO,W,<Geo_Num>,<mode>,<shape>,<latitude0>,<longitude0>,<radius0>,<latitude1>,<longitude
1>,<radius1>,<latitude2>,<longitude2>,<radius2>,<latitude3>,<longitude3>,<radius3>*Checksum<CR>
<LF>
//Get geo-fence setting.
$PQGEO,R*Checksum<CR><LF>
//Get the geo-fence status.
$PQGEO,Q,<GEOID>*Checksum <CR><LF>
```

Parameter:

Field	Format	Unit	Description
<Geo_Num>	Numeric	-	Geo-fence count. Range: 0–3. If <Geo_Num> is 0, the parameters after <mode> can be omitted.
<mode>	Numeric	-	Report mode. 0 Do not report when to enter or leave the geo-fence 1 Report when to enter the geo-fence 2 Report when to leave the geo-fence 3 Report when to enter or leave the geo-fence
<shape>	Numeric	-	Fence shape. 0 Circular with center and radius 1 Circular with center and one point on the circle (Not supported currently)

			2 Triangle (Not supported currently)
			3 Quadrangle (Not supported currently)
<latitude>	Numeric	Degree	Latitude of the point which is defined as the center of geo-fence circular region. Format: $\pm dd.dddddd$. Range: -90.000000–90.000000.
<longitude>	Numeric	Degree	Longitude of the point which is defined as the center of geo-fence circular region. Format: $\pm ddd.dddddd$. Range: -180.000000–180.000000.
<radius>	Numeric	Meter	Radius. Range: 1–65534.
<GEOID>	Numeric	-	Geo-fence index. Range: 0–3.

Result:

- If successful, the module returns:

```
//Set
$PQGEO,W,OK*Checksum<CR><LF>

[$PQGEO,<GEOID>,<action>,<Fix_status>,<UTC_date&Time>,<Latitude>,<Longitude>,<MSL_Altitude>,<Speed_Over_Ground>,<Course_Over_Ground>,<HDOP>,<PDOP>,<VDOP>,<GPS_Satellites_in_View>,<GPS_SatellitesUsed>*Checksum<CR><LF>]

//Get
$PQGEO,R,<Geo_Num>,<mode>,<shape>,<latitude0>,<longitude0>,<radius0>,<latitude1>,<longitude1>,<radius1>,<latitude2>,<longitude2>,<radius2>,<latitude3>,<longitude3>,<radius3>*Checksum<CR><LF>

//Query
$PQGEO,Q,<GEOID>,<status>*Checksum<CR><LF>
```

- If failed, the module returns:

```
//Set
$PQGEO,W,ERROR*Checksum<CR><LF>

//Get
$PQGEO,R,ERROR*Checksum<CR><LF>

//Query
$PQGEO,Q,ERROR*Checksum<CR><LF>
```

Parameter included in the result:

Field	Format	Unit	Description
<status>	Numeric	-	The status of current position. It is returned after Query Command is executed successfully. 0 Unknown position 1 Inside the geo-fence

2 Outside the geo-fence

URC format:

```
$PQGEO,<GEOID>,<action>,<Fix_status>,<UTC_date&Time>,<Latitude>,<Longitude>,<MSL_Altitude>,<Speed_Over_Ground>,<Course_Over_Ground>,<HDOP>,<PDOP>,<VDOP>,<GPS_Satellites_in_View>,<GPS_SatellitesUsed>*Checksum<CR><LF>
```

Parameter included in the URC:

Field	Format	Unit	Description
<GEOID>	Numeric	-	Geo-fence index. Range: 0–3.
<action>	Numeric	-	Current action of the module. 0 Outside 1 Unknown 2 Inside
<Fix_status>	Numeric	-	Fix status. 0 No fix 1 2D fix 2 3D fix
<UTC_date&Time>	Numeric	-	UTC time. Format: DyyMMddThhmmss.sss. D: Date yy: Current year - 2000 MM: Months 1–12 dd: Days 1–31 T: Time hh: Hours 0–23 mm: Minutes 0–59 ss.sss: Seconds 00.000–59.999
<Latitude>	-	Degree	Latitude of current position. Format: ±dd.dddddd Range: -90.000000–90.000000
<Longitude>	-	Degree	Longitude of current position. Format: ±ddd.dddddd Range: -180.000000–180.000000
<MSL_Altitude>	-	Meter	Mean sea level (MSL) altitude.
<Speed_Over_Ground>	-	km/h	Speed over ground.
<Course_Over_Ground>	-	Degree	Course over ground. Range: 0–360.00.
<HDOP>	-	-	Horizontal dilution of precision.

<PDOP>	-	-	Position dilution of precision.
<VDOP>	-	-	Vertical dilution of precision.
<GPS_Satellites_in_View>	-	-	GPS satellites in view.
<GPS_SatellitesUsed>	-	-	GPS satellites used.

Example:

```
$PQGEO,W,2,3,0,23.145266,117.211199,400,63.1452,157.2111,700*29
```

```
$PQGEO,W,OK*1F
```

```
$PQGEO,0,0,2,D231218T055108.000,25.269181,110.334106,180.7,1.49,71.43,0.46,0.86,0.72,16,40*4A
```

```
$PQGEO,1,0,2,D231218T055108.000,25.269181,110.334106,180.7,1.49,71.43,0.46,0.86,0.72,16,40*4B
```

```
$PQGEO,R*32
```

```
$PQGEO,R,2,3,0,23.145266,117.211199,400,63.1452,157.2111,700*2C
```

NOTE

1. The message takes effect immediately.
2. The URC is reported when **<mode>** ≠ 0.
3. If **<mode>** is 0 and there are no parameters after **<mode>**, this message disables the geo-fence.

5 RTCM Protocol

The modules support RTCM protocol which is in accordance with *RTCM Standard 10403.3 Differential GNSS Services – Version 3*. This protocol is used to transfer GNSS raw measurement data and is available from <https://www.rtcn.org/>.

Table 4: Supported RTCM3 Messages

Message Type	Type	Message Name
1005	Output	Stationary Antenna Reference Point
1019	Output	GPS Satellite Ephemeris Data
1020	Output	GLONASS Satellite Ephemeris Data
1042	Output	BDS Satellite Ephemeris Data
1044	Output	QZSS Satellite Ephemeris Data
1046	Output	Galileo I/NAV Satellite Ephemeris Data
1074	Output	GPS MSM4
1077	Output	GPS MSM7
1084	Output	GLONASS MSM4
1087	Output	GLONASS MSM7
1094	Output	Galileo MSM4
1097	Output	Galileo MSM7
1114	Output	QZSS MSM4
1117	Output	QZSS MSM7
1124	Output	BDS MSM4
1127	Output	BDS MSM7

NOTE

1. **\$PAIR432** can enable or disable MSM4 or MSM7 message format (1074, 1077, 1084, 1087, 1094, 1097, 1114, 1117, 1124, 1127) if the corresponding constellation is enabled.
2. **\$PAIR434** can enable or disable Stationary Antenna Reference Point message (1005).
3. **\$PAIR436** can enable or disable the output of ephemeris data (1019, 1020, 1042, 1044, 1046) if the corresponding constellation is enabled.

6 Appendix References

Table 5: Related Documents

Document Name
[1] Quectel_QCOM_User_Guide
[2] Quectel_EG800Q&EG91xQ_Series_FILE_Application_Note
[3] Quectel_EG800Q&EG91xQ_Series_AT_Commands_Manual

Table 6: Terms and Abbreviations

Abbreviation	Description
2D	2 Dimension
3D	3 Dimension
AIC	Active Interference Cancellation
ALP	Adaptive Low Power
BDS	BeiDou Navigation Satellite System
CMUX	Connection Multiplexing
DGPS	Differential Global Positioning System
DOP	Dilution of Precision
DTM	Duty Tracking Mode
EASY	Embedded Assist System
ECEF	Earth-Centered, Earth-Fixed
EGNOS	European Geostationary Navigation Overlay Service
EPE	Estimated Position Error

EPO	Extended Prediction Orbit
FLP	Fitness Low Power
GAGAN	GPS Aided Geo Augmented Navigation
Galileo	Galileo Satellite Navigation System
GEO	Geostationary Orbit
GGA	Global Positioning System Fix Data
GLONASS	Global Navigation Satellite System (Russia)
GLP	GPS Low Power
GNS	New GGA Message For GNSS
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GSA	GPS DOP and Active Satellites
GSV	GNSS Satellites in View
HDOP	Horizontal Dilution of Precision
ID	Identifier
MCU	Microcontroller Unit
ME	Mobile Equipment
MEO	Medium Earth Orbit
MNL	MTK Navigation Lib
MSAS	Multi-functional Satellite Augmentation System (Japan)
NMEA	NMEA (National Marine Electronics Association) 0183 Interface Standard
OTA	Over-the-air programming
PAIR	Proprietary Protocol of Airoha
PC	Personal Computer
PDOP	Position Dilution of Precision

PPS	Pulse Per Second
PRN	Pseudo-Random Noise
PVT	Position, Velocity, and Time
QZSS	Quasi-Zenith Satellite System
RF LNA	Radio Frequency Low Noise Amplifier
RLM	Return Link Message
RMC	Recommended Minimum Specific GNSS Data
RTCM	Radio Technical Commission for Maritime Services
SBAS	Satellite-Based Augmentation System
SLAS	Sub-meter Level Augmentation Service
SNR	Signal-to-Noise Ratio
SNTP	Simple Network Time Protocol
TA	Terminal Adapter
TTFF	Time to First Fix
UART	Universal Asynchronous Receiver & Transmitter
UFS	User File System
ULP	Ultra Low Power
URL	Uniform Resource Locator
USB	Universal Serial Bus
UTC	Coordinated Universal Time
VDOP	Vertical Dilution of Precision
VTG	Course Over Ground and Ground Speed
WAAS	Wide Area Augmentation System