MC20&MC30 GNSS Protocol Specification

GSM/GPRS/GNSS Module Series

Rev. MC20&MC30_GNSS_Protocol_Specification_V1.2

Date: 2018-08-29

Status: Released

Our aim is to provide customers with timely and comprehensive service. For any assistance, please contact our company headquarters:

Quectel Wireless Solutions Co., Ltd.
7th Floor, Hongye Building, No.1801 Hongmei Road, Xuhui District, Shanghai 200233, China
Tel: +86 21 5108 6236
Email: info@quectel.com

Or our local office. 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 to: support@quectel.com

GENERAL NOTES

QUECTEL OFFERS THE INFORMATION AS A SERVICE TO ITS CUSTOMERS. THE INFORMATION PROVIDED IS BASED UPON CUSTOMERS' REQUIREMENTS. QUECTEL MAKES EVERY EFFORT TO ENSURE THE QUALITY OF THE INFORMATION IT MAKES AVAILABLE. QUECTEL DOES NOT MAKE ANY WARRANTY AS TO THE INFORMATION CONTAINED HEREIN, AND DOES NOT ACCEPT ANY LIABILITY FOR ANY INJURY, LOSS OR DAMAGE OF ANY KIND INCURRED BY USE OF OR RELIANCE UPON THE INFORMATION. ALL INFORMATION SUPPLIED HEREIN IS SUBJECT TO CHANGE WITHOUT PRIOR NOTICE.

COPYRIGHT

About the Document

History

Revision	Date	Author	Description
1.0	2016-07-22	Ziv LIAO	Initial
1.1	2017-08-24	Simon HU	1. Added the following new AT commands: AT+QGNSSTS/AT+QGNSSEPO/AT+QGREFLOC/ AT+QGEPOAID/AT+QGEPOF. 2. Added the following new PMTK commands: PMTK285/PMTK306/PMTK308/PMTK458/ PMTK461/PMTK607/PMTK707. 3. Added the following new SDK commands: 4. PQBAUD/PQECEF/PQODO/PQPZ90/PQVEL/ PQJAM/PQRLM/PQGEO. 5. Modified description of GSA/GSV/PMTK161/ PMTK225/PMTK353/PMTK886/PQEPE/PQGLP.
1.2	2018-08-29	Simon HU	1. Added MC30 as the applicable module of the document. 2. Updated the longitude format in RMC and GGA messages. (Chapter 2.1 and 2.3) 3. Updated the description of parameter <item>, example, and added notes for AT+QGNSSRD. (Chapter 3.2) 4. Updated notes for AT+QGNSSCMD and AT+QGNSSTS. (Chapter 3.3 and 3.4) 5. Added the value 2 of parameter <mode>, notes and example for AT+QGNSSEPO. (Chapter 3.5) 6. Added example for AT+QGREFLOC. (Chapter 3.6) 7. Updated the notes and added example for AT+QGEPOAID. (Chapter 3.7) 8. Added example for AT+QGEPOF. (Chapter 3.8)

Contents

Table Index

1 Introduction

MC20&MC30 are multi-purpose modules which integrate a high performance GNSS engine and a quad-band GSM/GPRS engine. The GNSS engine is a single receiver integrating GPS and BeiDou systems. It supports multiple positioning and navigation systems including autonomous GPS, BeiDou, SBAS (including WAAS, EGNOS, MSAS and GAGAN), and QZSS. With the embedded GNSS function, MC20&MC30 can help customers get accurate coordinates, high-precision time, etc., and thus is ideal for use in wearable devices, vehicle and personnel tracking, and more fields.

This document describes the software aspects of MC20&MC30 modules. It supports NMEA 0183 standard commands, and also can be controlled and configured via MTK NMEA extended packet.

The document is applicable to the following modules:

- MC20 module which includes the following variants:

  a) OC: MC20CA-04-STD (supports BT3.0)
  b) OC: MC20ECA-04-BLE (supports BT4.0)
  c) OC: MC20CA-04-TTS (supports text-to-speech)

- MC30 module (supports GNSS+Cell ID+Wi-Fi hybrid positioning)

1.1. Differences between Two Application Modes

The internal GSM and GNSS engines of MC20/MC30 can work as a whole unit (**All-in-one** solution) or work relatively independently (**Stand-alone** solution) according to customers' demands.

In **All-in-one** solution, MC20/MC30 works as a whole unit. The GNSS part can be regarded as a peripheral of the GSM part. This allows for convenient communication between GSM and GNSS parts, such as AT command sending for GNSS control, GNSS part firmware upgrading, and EPO data download.

In **Stand-alone** solution, GSM and GNSS parts work independently, and thus have to be controlled separately.

When working in **All-in-one** or **Stand-alone** solution, there are some differences for MC20/MC30 to acquire NMEA output data, or send PMTK/SDK command. The details are listed below.

Table 1: Differences between All-in-one Solution and Stand-alone Solution

Item	All-in-one	Stand-alone
NMEA Output Data Acquisition	Acquire via sending **AT+QGNSSRD** command	Acquire directly
PMTK Command Sending	Send via **AT+QGNSSCMD** command	Send directly
SDK Command Sending	Send via **AT+QGNSSCMD** command	Send directly

NOTE

AT commands are effective only when the module is in **All-in-one** solution.

2 Standard NMEA Packet Protocol

MC20&MC30 support standard NMEA 0183 messages, and the following tables show the structure of these messages.

2.1. --RMC

RMC-Recommended Minimum Position Data (including position, velocity and time).

Example:
$GPRMC,015606.000,A,3150.7584,N,11712.0491,E,0.00,231.36,280715,,,A*67<CR><LF>
$GNRMC,084629.000,A,3150.7822,N,11711.9323,E,0.00,119.00,240715,,,D*7C<CR><LF>
$BDRMC,020547.000,A,3150.7813,N,11711.9212,E,0.37,229.71,280715,,,A*7C<CR><LF>

Field	Description
$	Each NMEA message starts with '$'
--RMC	Message ID
UTC Time	Time in format 'hhmmss.sss'
Data Valid	'V'=Invalid 'A'=Valid
Latitude	Latitude in format 'ddmm.mmmm' (degrees and minutes)
N/S	'N'=North 'S'=South
Longitude	Longitude in format 'dddmm.mmmm' (degrees and minutes)
E/W	'E'=East 'W'=West
Speed	Speed over ground in knots
COG	Course over ground in degree
Date	Date in format 'ddmmyy'
Magnetic Variation	Magnetic variation in degree, not being output

E/W	Magnetic variation E/W indicator, not being output
Positioning Mode	'N'=No fix 'A'=Autonomous GNSS fix 'D'=Differential GNSS fix
*	End character of data field
Checksum	Hexadecimal checksum
<CR><LF>	Each NMEA message ends with 'CR' and 'LF'

NOTE

For more details about talker ID, please refer to ***Table 9***.

2.2. --VTG

VTG-Track Made Good and Ground Speed.

Example:
$GPVTG,227.15,T,,M,0.00,N,0.00,K,A*3E<CR><LF>
$GNVTG,19.11,T,,M,0.16,N,0.30,K,A*1F<CR><LF>
$BDVTG,229.71,T,,M,0.37,N,0.68,K,A*29<CR><LF>

Field	Description
$	Each NMEA message starts with '$'
--VTG	Message ID
COG (T)	Course over ground (true) in degree
T	Fixed field, true
COG(M)	Course over ground (magnetic), not being output
M	Fixed field, magnetic
Speed	Speed over ground in knots
N	Fixed field, knots
Speed	Speed over ground in km/h

K	Fixed field, km/h
Positioning Mode	‘N’=No fix ‘A’=Autonomous GNSS fix ‘D’=Differential GNSS fix
*	End character of data field
Checksum	Hexadecimal checksum
<CR><LF>	Each NMEA message ends with ‘CR’ and ‘LF’

NOTE

For more details about talker ID, please refer to ***Table 9***.

2.3. --GGA

GGA-Global Positioning System Fix Data, is the essential fix data which provides 3D location and accuracy data.

Example:
$GPGGA,015606.000,3150.7584,N,11712.0491,E,1,5,2.28,265.0,M,0.0,M,,*65<CR><LF>
$GNGGA,083354.000,3150.7790,N,11711.9289,E,1,8,2.85,53.2,M,0.0,M,,*4B<CR><LF>
$BDGGA,020547.000,3150.7813,N,11711.9212,E,1,3,3.65,55.3,M,0.0,M,,*4C<CR><LF>

Field	Description
$	Each NMEA message starts with ‘$’
--GGA	Message ID
UTC Time	Time in format ‘hhmmss.sss’
Latitude	Latitude in format ‘ddmm.mmmm’ (degrees and minutes)
N/S	‘N’=North ‘S’=South
Longitude	Longitude in format ‘dddmm.mmmm’ (degrees and minutes)
E/W	‘E’=East ‘W’=West
Fix Status	‘0’=Invalid ‘1’=GNSS fix

	'2'=DGPS fix '6'=Estimated (dead reckoning) Mode
Number of SV	Number of satellites being used (0~26)
HDOP	Horizontal dilution of precision
Altitude	Altitude in meters according to WGS84 ellipsoid
M	Fixed field, meter
Geoid Separation	Height of Geoid (means sea level) above WGS84 ellipsoid, meter
M	Fixed field, meter
DGPS Age	Age of DGPS data in seconds, empty if DGPS is not used
DGPS Station ID	DGPS station ID, empty if DGPS is not used
*	End character of data field
Checksum	Hexadecimal checksum
<CR><LF>	Each NMEA message ends with 'CR' and 'LF'

NOTE

For more details about talker ID, please refer to ***Table 9***.

2.4. --GSA

GSA-GNSS DOP and Active Satellites, which provides details on the fix and includes the number of satellites being used in the current solution and the DOP.

Example:
$GPGSA,A,3,03,17,11,23,193,,,,,,,,3.72,2.85,2.39*3C<CR><LF>
$GNGSA,A,3,23,09,17,03,01,193,,,,,,,1.23,0.74,0.99*28<CR><LF>
$GLGSA,A,3,68,66,82,67,81,,,,,,,,1.25,0.71,1.03*10<CR><LF>
$GAGSA,A,3,07,,,,,,,,,,,,1.25,0.71,1.03*16<CR><LF>
$BDGSA,A,2,10,11,07,,,,,,,,,,3.79,3.65,1.00*18<CR><LF>

Field	Description
$	Each NMEA message starts with '$'

--GSA	Message ID
Mode	Auto selection of 2D or 3D fix 'M'=Manual, forced to switch 2D/3D mode 'A'=Allowed to automatically switch 2D/3D mode
Fix Status	'1'=No fix '2'=2D fix '3'=3D fix
Satellite Used 1	Satellite used on channel 1
Satellite Used 2	Satellite used on channel 2
Satellite Used 3	Satellite used on channel 3
Satellite Used 4	Satellite used on channel 4
Satellite Used 5	Satellite used on channel 5
Satellite Used 6	Satellite used on channel 6
Satellite Used 7	Satellite used on channel 7
Satellite Used 8	Satellite used on channel 8
Satellite Used 9	Satellite used on channel 9
Satellite Used 10	Satellite used on channel 10
Satellite Used 11	Satellite used on channel 11
Satellite Used 12	Satellite used on channel 12
PDOP	Position dilution of precision
HDOP	Horizontal dilution of precision
VDOP	Vertical dilution of precision
*	End character of data field
Checksum	Hexadecimal checksum
<CR><LF>	Each NMEA message ends with 'CR' and 'LF'

NOTE

For more details about talker ID, please refer to ***Table 9***.

2.5. --GSV

GSV-GNSS Satellites in View. One GSV sentence can only provide data for at most 4 satellites, so several sentences might be required for full information. Since GSV includes satellites that are not used as part of the solution, GSV sentence contains more satellites than GGA does.

Example:
$GPGSV,3,1,11,193,69,099,30,17,62,354,36,06,47,272,,03,40,054,30*4E<CR><LF>
$GPGSV,3,2,11,02,13,255,,01,12,055,19,23,11,102,25,11,05,074,24*75<CR><LF>
$GPGSV,3,3,11,24,03,303,,47,,,,32,,,21*4D<CR><LF>
$GLGSV,3,1,11,69,48,142,39,68,43,058,51,83,40,049,51,84,40,334,43*64<CR><LF>
$GLGSV,3,2,11,74,30,271,15,73,17,218,19,75,13,324,30,70,07,184,*6E<CR><LF>
$GLGSV,3,3,11,85,06,296,34,82,02,092,21,67,02,023,*56<CR><LF>
$BDGSV,2,1,06,08,63,015,30,11,59,350,32,12,39,081,26,14,29,184,21*68<CR><LF>
$GAGSV,1,1,03,07,61,012,41,02,,,29,20,,,34*51<CR><LF>
$BDGSV,2,2,06,13,07,295,,01,,,26*50<CR><LF>

Field	Description
$	Each NMEA message starts with '$'
--GSV	Message ID
Number of Message	Number of messages, total number of GPGSV messages being output (1~4)
Sequence Number	Sequence number of this entry (1~4)
Satellites in View	Total satellites in view
Satellite ID 1	Satellite ID
Elevation 1	Elevation in degree (0~90)
Azimuth 1	Azimuth in degree (0~359)
SNR 1	Signal to noise ration in dB-Hz (0~99), empty if not tracking
Satellite ID 2	Satellite ID
Elevation 2	Elevation in degree (0~90)
Azimuth 2	Azimuth in degree (0~359)
SNR 2	Signal to noise ration in dB-Hz (0~99), empty if not tracking
Satellite ID 3	Satellite ID

Elevation 3	Elevation in degree (0~90)
Azimuth 3	Azimuth in degree (0~359)
SNR 3	Signal to noise ration in dB-Hz (0~99), empty if not tracking
Satellite ID 4	Satellite ID
Elevation 4	Elevation in degree (0~90)
Azimuth 4	Azimuth in degree (0~359)
SNR 4	Signal to noise ration in dB-Hz (0~99), empty if not tracking
*	End character of data field
Checksum	Hexadecimal checksum
<CR><LF>	Each NMEA message ends with 'CR' and 'LF'

NOTE

For more details about talker ID, please refer to ***Table 9***.

2.6. --GLL

GLL-Geographic Latitude and Longitude, which contains position information, time of position fix and status.

Example:
$GPGLL,3150.7584,N,11712.0491,E,015606.000,A,A*5C<CR><LF>
$GNGLL,3150.7790,N,11711.9289,E,083354.000,A,A*4D<CR><LF>
$BDGLL,3150.7813,N,11711.9212,E,020547.000,A,A*49<CR><LF>

Field	Description
$	Each NMEA message starts with '$'
--GLL	Message ID
Latitude	Latitude in format 'ddmm.mmmm' (degrees and minutes)
N/S	'N'=North 'S'=South

Longitude	Longitude in format ‘dddmm.mmmm’ (degrees and minutes)
E/W	‘E’=East ‘W’=West
UTC Time	Time in format ‘hhmmss.sss’
Data Valid	‘V’=Invalid ‘A’=Valid
Positioning Mode	‘N’=No fix ‘A’=Autonomous GNSS fix ‘D’=Differential GNSS fix
*	End character of data field
Checksum	Hexadecimal checksum
<CR><LF>	Each NMEA message ends with ‘CR’ and ‘LF’

NOTE

For more details about talker ID, please refer to ***Table 9***.

3 GNSS AT Commands

The commands below are used to control or configure the internal GNSS engine of MC20&MC30. These commands are effective only in **All-in-one** solution.

Table 2: Overview of GNSS AT Commands for MC20&MC30

Command	Description
AT+QGNSSC	GNSS module power control
AT+QGNSSRD	Read GNSS navigation information
AT+QGNSSCMD	Send commands to GNSS module
AT+QGNSSTS	Get Time Synchronization Status for GNSS Module
AT+QGNSSEPO	Enable/Disable EPO™ Function
AT+QGREFLOC	Set Reference Location Information for QuecFastFix Online
AT+QGEPOAID	Trigger EPO™ Function
AT+QGEPOF	EPO™ File Operation

3.1. AT+QGNSSC GNSS Module Power Control

The command is used to control the power supply of GNSS module.

AT+QGNSSC GNSS Module Power Control	
Test Command **AT+QGNSSC=?**	Response **+QGNSSC:** (list of supported **<mode>**s) **OK**
Read Command **AT+QGNSSC?**	Response **+QGNSSC: <mode>**

	OK
Write Command **AT+QGNSSC=<mode>**	Response **OK** If there is any error, response: **+CME ERROR: <err>**

Parameter

<mode>	0	Power off GNSS module
	1	Power on GNSS module

NOTE

In **Stand-alone** solution, the power supply of GNSS is controlled by an external circuit rather than GPS_VCC_EN pin. In such case, **AT+QGNSSC** command cannot be used and thus can be ignored.

Example

```
AT+QGNSSC?                  //Query GNSS power status
+QGNSSC: 0                  //GNSS powered off

OK
AT+QGNSSC=1                 //Power on GNSS

OK
```

3.2. AT+QGNSSRD Read GNSS Navigation Information

The command is used to read the GNSS navigation information.

AT+QGNSSRD Read GNSS Navigation Information	
Test Command **AT+QGNSSRD=?**	Response **+QGNSSRD:** (list of supported **<item>**s) **OK**
Read Command **AT+QGNSSRD?**	Response **+QGNSSRD:** (information of all supported **<item>**s) **OK**

Write Command **AT+QGNSSRD=<item>**	Response **+QGNSSRD:** (information of **<item>**) **OK** If there is any error, response: **+CME ERROR: <err>**

Parameter

<item>	"NMEA/GGA": Get GGA sentence "NMEA/GLL": Get GLL sentence "NMEA/GSA": Get GSA sentence "NMEA/GSV": Get GSV sentence "NMEA/RMC": Get RMC sentence "NMEA/VTG": Get VTG sentence "NMEA/EPE": Get PQEPE URC "NMEA/ECEF": Get PQECEF URC "NMEA/PZ90": Get PQPZ90 URC "NMEA/VEL": Get PQVEL URC "NMEA/GEO": Get PQGEO URC "NMEA/JAM": Get PQJAM URC "NMEA/RLM": Get PQRLM URC

NOTES

1. The command can only be executed 3 seconds after GNSS is powered on, otherwise **+CME ERROR: 7103** will be returned.
2. For more details about the URCs, please refer to ***Chapter 5***.

Example

```
AT+QGNSSRD?                          //Query GNSS NMEA sentence
+QGNSSRD: $GNRMC,034035.000,A,3150.8617,N,11711.9038,E,3.02,183.45,240516,,,A*75
$GNVTG,183.45,T,,M,3.02,N,5.59,K,A*20
$GNGGA,034035.000,3150.8617,N,11711.9038,E,1,4,1.50,40.9,M,0.0,M,,*44
$GPGSA,A,3,26,21,,,,,,,,,,,1.75,1.50,0.91*0A
$GLGSA,A,3,82,70,,,,,,,,,,,1.75,1.50,0.91*1C
$GPGSV,3,1,12,16,67,308,,26,58,021,16,23,40,307,,31,40,088,*7F
$GPGSV,3,2,12,08,17,199,,09,14,320,,21,10,086,14,14,10,153,*73
$GPGSV,3,3,12,22,09,226,,193,06,165,,32,03,154,,29,01,034,*45
$GLGSV,3,1,09,81,44,073,,79,40,041,,82,38,145,15,80,36,323,*66
$GLGSV,3,2,09,70,30,290,16,69,26,225,,78,12,078,,88,09,027,*64
$GLGSV,3,3,09,71,05,334,*5B
```

```
$GNGLL,3150.8617,N,11711.9038,E,034035.000,A,A*4C

OK
AT+QGNSSRD="NMEA/RMC"                    //Query RMC information
+QGNSSRD: $GNRMC,034036.000,A,3150.8612,N,11711.9045,E,2.74,178.00,240516,,,A*7C

OK
AT+QGNSSRD="NMEA/GSA"                    //Query GSA information
+QGNSSRD: $GPGSA,A,3,26,21,,,,,,,,,,,1.76,1.50,0.91*09

OK
```

3.3. AT+QGNSSCMD Send Commands to GNSS Module

The command is used to send commands to GNSS module, which allows customers to optionally use some functions to meet application demands.

AT+QGNSSCMD Send Commands to GNSS Module	
Test Command **AT+QGNSSCMD=?**	Response **+QGNSSCMD: (0,1),"cmdString"** **OK**
Write Command **AT+QGNSSCMD=<cmdType>,<cmdString>**	Response **OK** If there is any error, response: **+CME ERROR: <err>**

Parameter

<cmdType>	0	NMEA style command
	1	Hex style command
<cmdString>	Command string	

NOTES

1. Currently only **<cmdType>**=0 is supported.
2. The command can only be executed 3 seconds after GNSS is powered on, otherwise it will not be able to reach GNSS.

Example

```
AT+QGNSSCMD=0,"$PMTK605*31"          //Query GNSS version information
OK

+QGNSSCMD: $PMTK705,AXN_5.10_3333_17062100,0001,MC20-GNSS,1.0*27
```

3.4. AT+QGNSSTS Get Time Synchronization Status for GNSS Module

The command is used to get time synchronization status for GNSS module, and the time plays a very important role in EPO™ function.

AT+QGNSSTS Get Time Synchronization Status for GNSS Module	
Test Command **AT+QGNSSTS=?**	Response **+QGNSSTS: <status>** **OK**
Read Command **AT+QGNSSTS?**	Response **+QGNSSTS: <status>** **OK**

Parameter

<status>	0	Time is not synchronized
	1	Time is synchronized successfully

NOTES

1. Exact time is very important to EPO™ function. So customers must ensure the time is valid before using EPO™ function.
2. The time synchronization status queried by **AT+QGNSSTS?** will be updated in either of the following cases:
   - The time is synchronized via NITZ.
   - GNSS gets fixed in **All-in-one** solution.

Example

```
AT+QGNSSTS=?                 //Test command
+QGNSSTS: (0,1)
```

```
OK
AT+QGNSSTS?          //Read time synchronization status
+QGNSSTS: 1          //Time is synchronized successfully

OK
```

3.5. AT+QGNSSEPO Enable/Disable EPO™ Function

The command is used to enable or disable EPO™ function.

AT+QGNSSEPO Enable/Disable EPO™ Function	
Test Command **AT+QGNSSEPO=?**	Response **+QGNSSEPO:** (list of supported **<mode>**s) **OK**
Read Command **AT+QGNSSEPO?**	Response **+QGNSSEPO: <mode>,<account_id>** **OK**
Write Command When **<mode>**=0 or 1, **AT+QGNSSEPO=<mode>[,<account_id>]** When **<mode>**=2, **AT+QGNSSEPO=<mode>,<address>[,<port>]**	Response **OK** If there is any error, response: **+CME ERROR: <err>**

Parameter

<mode>	0	Disable EPO™ function
	1	Enable EPO™ function
	2	Configure the EPO server address and port
<account_id>	2	Set account ID for EPO™ function
<address>	String type. EPO server address. It could be domain name or IP address.	
<port>	1-65535.	Integer type. EPO server port. The default value is 80 when it is omitted.

NOTES

1. The parameter **<account_id>** only supports 2. It can be omitted and 2 will be its default value when it is omitted.
2. The EPO™ function should be enabled after the time is synchronized successfully.
3. **<mode>**=2 is only supported by MC20 module with OC **MC20CA-04-STD** and **MC20ECA-04-BLE**.
4. If the EPO server is not configured by **<mode>**=2, the MTK sever will be accessed by default.
5. If a third-party EPO server needs to be used, **AT+QGNSSEPO=2,<address>[,<port>]** should be executed before executing **AT+QGNSSEPO=1**.
6. The parameters **<address>** and **<port>** configured by **<mode>**=2 will be discarded after executing **AT+QGNSSEPO=0**. If a third-party EPO server needs to be used, the two parameters should be configured again before executing **AT+QGNSSEPO=1**.

Since **<mode>**=2 is only supported by MC20 module with OC **MC20CA-04-STD** and **MC20ECA-04-BLE**, the examples of this command for MC20 and MC30 modules are shown respectively below.

Example (MC20 Module)

```
AT+QGNSSEPO=?                 //Test command
+QGNSSEPO: (0,2)

OK
AT+CREG?;+CGREG?              //Check network status
+CREG: 0,1

+CGREG: 0,1

OK
AT+QGNSSTS?                   //Read time synchronization status
+QGNSSTS: 1                   //Time is synchronized successfully

OK
AT+QGNSSEPO=2,“fota.quectel-service.com”,81   //This is an optional procedure. If the command is
                                                used, the domain name of Quectel in this
                                                command is only for testing purpose and
                                                customers can use their own domain name.
OK
AT+QGNSSEPO=1                 //Enable EPO™ function
OK
AT+QGNSSEPO?                  //Read EPO™ status
+QGNSSEPO: 1,2

OK
```

Example (MC30 Module)

```
AT+QGNSSEPO=?                 //Test command
+QGNSSEPO: (0,1)

OK
AT+CREG?;+CGREG?              //Check network status
+CREG: 0,1

+CGREG: 0,1

OK
AT+QGNSSTS?                   //Read time synchronization status
+QGNSSTS: 1                   //Time is synchronized successfully

OK

AT+QGNSSEPO=1                 //Enable EPO™ function
OK
AT+QGNSSEPO?                  //Read EPO™ status
+QGNSSEPO: 1,2

OK
```

3.6. AT+QGREFLOC Set Reference Location Information for QuecFastFix Online

The command is used to set reference location information for QuecFastFix Online function.

AT+QGREFLOC Set Reference Location Information for QuecFastFix Online	
Test Command **AT+QGREFLOC=?**	Response **+QGREFLOC: <ref_latitude>,<ref_longitude>** **OK**
Read Command **AT+QGREFLOC?**	Response **+QGREFLOC: <ref_latitude>,<ref_longitude>** **OK**
Write Command **AT+QGREFLOC=<ref_latitude>,<ref_longitude>**	Response **OK**

	If there is any error, response: **+CME ERROR: <err>**

Parameter

<ref_latitude>	Latitude information of the reference location
<ref_longitude>	Longitude information of the reference location

NOTES

1. The range of **<ref_latitude>** is -90°~90° north latitude, and the range of **<ref_longitude>** is -180°~180° east longitude. The input format of the parameter should retain 6 decimal places, and the unit is degree.
2. The command works for QuecFastFix Online function and should be set before executing **AT+QGNSSEPO=1**.

Example

```
AT+QGREGLOC=?                 //Test command
+QGREFLOC: <ref_latitude>,<ref_longitude>

OK
AT+QGREFLOC=31.507985,117.119750
OK
```

3.7. AT+QGEPOAID Trigger EPO™ Function

The command is used to trigger EPO™ function.

AT+QGEPOAID Trigger EPO™ Function	
Test Command **AT+QGEPOAID=?**	Response **OK**
Execution Command **AT+QEPOAID**	Response **OK** If there is any error, response: **+CME ERROR: <err>**

NOTE

If GNSS is powered on already, this command can be used to trigger EPO™ function after executing **AT+QGNSSEPO=1**.

Example

```
AT+QGNSSC=1                 //Power on GNSS
OK
AT+CREG?;+CGREG?            //Check network status
+CREG: 0,1

+CGREG: 0,1

OK
AT+QGNSSTS?                 //Inquire time synchronization status
+QGNSSTS: 1

OK
AT+QGNSSEPO=1
OK
AT+QGEPOAID
OK
```

3.8. AT+QGEPOF EPO™ File Operation

The command is used to operate EPO™ related files, including deleting related files, getting file size and querying validity period of EPO™ files.

AT+QGEPOF EPO™ File Operation	
Test Command **AT+QGEPOF=?**	Response **+QGEPOF:** (list of supported **<mode>**s)**,** (list of supported **<index>**s) **OK**
Write Command **AT+QGEPOF=<mode>,<index>**	Response If **<mode>** is 0, response: **+QGEPOF: <size_a>,<size_b>,<size_c>** **OK** If **<mode>** is 1, response:

	OK If **<mode>** is 2, response: **+QGEPOF: <time>** **OK** If there is any error, response: **+CME ERROR: <err>**

Parameter

<mode>	Operation mode	
	0	Get EPO™ file size
	1	Delete EPO™ file
	2	Query validity period of EPO™ files
<index>	EPO™ file selection	
	1	Select the EPO™ file containing 6 hours of data
	2	Select the EPO™ file containing the first 3 days of data
	3	Select the EPO™ file containing the second 3 days of data
	255	Select the above 3 files
<size_a>	Integer value. Positive numbers indicate the file size, and negative numbers indicate failed file operation.	
	0-4032	Size of the EPO™ file containing 6 hours of data
	-9	File not found
	-16	File access denied
	-19	Path not found
	Other negative values	Other failed file operation
<size_b>	Integer value. Positive numbers indicate the file size, and negative numbers indicate failed file operation.	
	0-48384	Size of the EPO™ file containing the first 3 days of data
	-9	File not found
	-16	File access denied
	-19	Path not found
	Other negative values	Other failed file operation
<size_c>	Integer value. Positive numbers indicate the file size, and negative numbers indicate failed file operation.	
	0-48384	Size of the EPO™ file containing the second 3 days of data
	-9	File not found
	-16	File access denied

	-19	Path not found
	Other negative values	Other failed file operation
<time>	String type. The queried validity period of EPO™ files The format is “yyyy-MM-dd hh:mm:ss” (UTC time), and the characters indicate year, month, day, hour, minute and second respectively.	
	-9	File not found
	-16	File access denied
	-19	Path not found
	Other negative values	Other failed file operation

NOTES

1. If the EPO™ files are deleted, there is a need to trigger EPO™ function again. For more details, please refer to ***document [5]***.
2. If **<mode>** is 2, **<index>** can be omitted.
3. Commands **AT+QGEPOF=1,1**, **AT+QGEPOF=1,2** and **AT+QGEPOF=1,3** are only for testing purpose. Please use **AT+QGEPOF=1,255** in the real application.

Example

```
AT+QGEPOF=?              //Test command
+QGEPOF: (0-2),(1-3,255)

OK
AT+QGEPOF=0,255          //Get EPO™ file size
+QGEPOF: -9,48384,48384

OK
AT+QGEPOF=2              //Query validity period of EPO™ files
+QGEPOF: 2017-02-19 00:00:00

OK
AT+QGEPOF=1,255          //Delete all EPO™ files.
OK
AT+QGEPOF=0,255
+QGEPOF: -9,-9,-9

OK
AT+QGEPOF=2
+QGEPOF: -9

OK
```

4 MTK NMEA Packet Protocol

This chapter introduces the MTK NMEA packet protocol, which is a set of extension messages of standard NMEA packet protocol. These messages are used to control and configure the internal GNSS engine of MC20&MC30. The following tables show the structure of MTK NMEA packet.

4.1. Packet Type: 010 PMTK_SYS_MSG

This message is used to automatically output system messages through GNSS module.

Data Field: None Example: $PMTK010,001*2E<CR><LF>	
Field	**Description**
$	Each NMEA message starts with ‘$’
PMTK	MTK proprietary message
Packet Type	010
Message	System message ‘0’=Unknown ‘1’=Startup ‘2’=Notification for the host aiding EPO ‘3’=Notification for the transition to normal mode is successfully done
*	End character of data field
Checksum	Hexadecimal checksum
<CR><LF>	Each NMEA message ends with ‘CR’ and ‘LF’

4.2. Packet Type: 011 PMTK_TXT_MSG

This message is used to automatically output system messages through GNSS module.

Data Field:
None
Example:
$PMTK011,MTKGPS*08<CR><LF>

Field	Description
$	Each NMEA message starts with '$'
PMTK	MTK proprietary message
Packet Type	011
Message	MTKGPS
*	End character of data field
Checksum	Hexadecimal checksum
<CR><LF>	Each NMEA message ends with 'CR' and 'LF'

4.3. Packet Type: 001 PMTK_ACK

Acknowledgement of PMTK command. In order to inform the sender whether the receiver has received the packet, and an acknowledge packet PMTK_ACK should be returned after the receiver receives a packet.

Some commands will cause the GNSS module to restart or change the baud rate. There is no PMTK_ACK for those commands as listed below.

- PMTK_CMD_HOT_START
- PMTK_CMD_WARM_START
- PMTK_CMD_COLD_START
- PMTK_CMD_FULL_COLD_START
- PMTK_SET_NMEA_BAUDRATE

Data Field:
$PMTK001,Cmd,Flag
Example:

$PMTK001,869,3*37<CR><LF>	
Field	**Description**
$	Each NMEA message starts with ‘$’
PMTK	MTK proprietary message
Packet Type	001
Cmd	The packet type that the acknowledge responds
Flag	‘0’=Invalid packet ‘1’=Unsupported packet type ‘2’=Valid packet, but action failed ‘3’=Valid packet, action succeeded
*	End character of data field
Checksum	Hexadecimal checksum
<CR><LF>	Each NMEA message ends with ‘CR’ and ‘LF’

4.4. Packet Type: 101 PMTK_CMD_HOT_START

This message is used to perform the hot start of the GNSS module (use all available data in the NV store). Normally hot start means the GNSS module was powered down for less than 3 hours (RTC must be alive) and its ephemeris is still valid. As there is no need for downloading ephemeris, it is the fastest startup method.

Data Field: None Example: $PMTK101*32<CR><LF>	
Field	**Description**
$	Each NMEA message starts with ‘$’
PMTK	MTK proprietary message
Packet Type	101
*	End character of data field
Checksum	Hexadecimal checksum

<CR><LF>	Each NMEA message ends with ‘CR’ and ‘LF’

4.5. Packet Type: 102 PMTK_CMD_WARM_START

This message is used to perform the warm start of the GNSS module. Warm start means the GNSS module has approximate information on time, position and coarse data of satellite positions. But it needs to download ephemeris until it can get a fix. Using this message will force a warm restart on the GNSS module without using the ephemeris data in NV.

Data Field: None Example: $PMTK102*31<CR><LF>	
Field	**Description**
$	Each NMEA message starts with ‘$’
PMTK	MTK proprietary message
Packet Type	102
*	End character of data field
Checksum	Hexadecimal checksum
<CR><LF>	Each NMEA message ends with ‘CR’ and ‘LF’

4.6. Packet Type: 103 PMTK_CMD_COLD_START

This message is used to perform cold start of the GNSS module. Using this message will force a cold restart on the GNSS module without using any prior location information, including time, position, almanacs and ephemeris data.

Data Field: None Example: $PMTK103*30<CR><LF>	
Field	**Description**

$	Each NMEA message starts with ‘$’
PMTK	MTK proprietary message
Packet Type	103
*	End character of data field
Checksum	Hexadecimal checksum
<CR><LF>	Each NMEA message ends with ‘CR’ and ‘LF’

4.7. Packet Type: 104 PMTK_CMD_FULL_COLD_START

This message is essentially a cold restart, but additionally clear system and user configuration at re-start. That is, reset the GNSS module to the factory status. Full cold start means the GNSS module has no information on last location. It needs to search the full time and frequency space, and also all possible satellite numbers before it can get a fix.

Data Field: None Example: $PMTK104*37<CR><LF>	
Field	**Description**
$	Each NMEA message starts with ‘$’
PMTK	MTK proprietary message
Packet Type	104
*	End character of data field
Checksum	Hexadecimal checksum
<CR><LF>	Each NMEA message ends with ‘CR’ and ‘LF’

4.8. Packet Type: 161 PMTK_CMD_STANDBY_MODE

This message is used to enter into standby mode for power saving and is not supported in **All-in-one** solution.

Data Field: $PMTK161,Type Example: $PMTK161,0*28<CR><LF> Response: $PMTK001,161,3*36<CR><LF>	
Field	**Description**
$	Each NMEA message starts with '$'
PMTK	MTK proprietary message
Packet Type	161
Type	'0'=Stop mode
*	End character of data field
Checksum	Hexadecimal checksum
<CR><LF>	Each NMEA message ends with 'CR' and 'LF'

4.9. Packet Type: 183 PMTK_LOCUS_QUERY_STATUS

This message is used to query LOCUS logging status and is not supported in **All-in-one** solution.

Data Field: None Example: $PMTK183*38<CR><LF> Response: $PMTK001,183,3*3A<CR><LF>	
Field	**Description**
$	Each NMEA message starts with '$'
PMTK	MTK proprietary message

Packet Type	183
*	End character of data field
Checksum	Hexadecimal checksum
<CR><LF>	Each NMEA message ends with ‘CR’ and ‘LF’

Return:

Example: $PMTKLOG,456,0,11,31,2,0,0,0,3769,46*48<CR><LF>	
Field	Description
$	Each NMEA message starts with ‘$’
PMTK	MTK proprietary message
Packet Type	LOG
Serial#	Logging serial number: 0~65535
Type	Logging type-0: Overlap, 1: Fullstop
Mode	Logging mode-0x08: Interval logger
Content	Logging contents of configuration
Interval	Logging interval setting (valid when interval mode is selected)
Distance	Logging distance setting (valid when distance mode is selected)
Speed	Logging speed setting (valid when speed mode is selected)
Status	Logging status-1: Stop logging, 0: Logging
Number	Logging number of data record
Percent	Logging life used percentage (0%~100%)
*	End character of data field
Checksum	Hexadecimal checksum
<CR><LF>	Each NMEA message ends with ‘CR’ and ‘LF’

4.10. Packet Type: 184 PMTK_LOCUS_ERASE_FLASH

This message is used to erase logger flash and is not supported in **All-in-one** solution.

Data Field: $PMTK184,Type Example: $PMTK184,1*22<CR><LF> Response: $PMTK001,184,3*3D<CR><LF>	
Field	**Description**
$	Each NMEA message starts with '$'
PMTK	MTK proprietary message
Packet Type	184
Type	'1'=Erase all logger internal flash data
*	End character of data field
Checksum	Hexadecimal checksum
<CR><LF>	Each NMEA message ends with 'CR' and 'LF'

4.11. Packet Type: 185 PMTK_LOCUS_STOP_LOGGER

This message is used to stop or start logging data and is not supported in **All-in-one** solution.

Data Field: $PMTK185,Stutas Example: $PMTK185,1*23<CR><LF> Response: $PMTK001,185,3*3C<CR><LF>	
Field	**Description**
$	Each NMEA message starts with '$'
PMTK	MTK proprietary message

Packet Type	185
Status	'0'=Start logging '1'=Stop logging
*	End character of data field
Checksum	Hexadecimal checksum
<CR><LF>	Each NMEA message ends with 'CR' and 'LF'

4.12. Packet Type: 220 PMTK_SET_POS_FIX

This message is used to set position fix interval.

Data Field: $PMTK220, Interval Example: $PMTK220,1000*1F<CR><LF> Response: $PMTK001,220,3*30<CR><LF>	
Field	**Description**
$	Each NMEA message starts with '$'
PMTK	MTK proprietary message
Packet Type	220
Interval	Position fix interval (msec). Range:100~10000
*	End character of data field
Checksum	Hexadecimal checksum
<CR><LF>	Each NMEA message ends with 'CR' and 'LF'

4.13. Packet Type: 223 PMTK_SET_AL_DEE_CFG

This message is used to configure DEE.

Data Field: $PMTK223,SV,SNR,Extension threshold,Extension gap Example: $PMTK223,1,30,180000,60000*3C<CR><LF> Response: $PMTK001,223,3*33<CR><LF>	
Field	**Description**
$	Each NMEA message starts with '$'
PMTK	MTK proprietary message
Packet Type	223
SV	Range: 1~4 (Default value: 1)
SNR	Range: 25~30 (Default value: 30)
Extension Threshold	Range: 40000~180000 (Default value: 180000)
Extension Gap	Range: 0~3600000 (Default value: 60000)
*	End character of data field
Checksum	Hexadecimal checksum
<CR><LF>	Each NMEA message ends with 'CR' and 'LF'

4.14. Packet Type: 225 PMTK_SET_PERIODIC_MODE

This message is used to enter into periodic mode for power saving and is not supported in **All-in-one** solution.

Data Field: $PMTK225,Type,Running time,Sleep time,Second running time,Second sleep time Example: $PMTK225,8*23<CR><LF> Response: $PMTK001,225,3*35<CR><LF>

Field	Description
$	Each NMEA message starts with '$'
PMTK	MTK proprietary message
Packet Type	225
Type	'0'=Back to normal mode '1'=Periodic Backup mode '2'=Periodic Standby mode '8'=AlwaysLocate™ Standby mode '9'=AlwaysLocate™ Backup mode
Run Time	'0': Disable >='1000': Enable (Range: 1000~518400000)
Sleep Time	(Range: 1000~518400000)
Second Run Time	'0': Disable >='1000': Enable (Range: 1000~518400000)
Second Sleep Time	(Range: 1000~518400000)
*	End character of data field
Checksum	Hexadecimal checksum
<CR><LF>	Each NMEA message ends with 'CR' and 'LF'

NOTE

The unit of run time or sleep time is msec. The second runtime should be more than the first run time when the first run time is a non-zero value.

4.15. Packet Type: 256 PMTK_SET_TIMING_PRODUCT

This message is used to enable or disable the timing of product mode (Default: off).

Data Field:
$PMTK256,Enable
Example:
$PMTK256,0*2F<CR><LF>
Response:
$PMTK001,256,3*31<CR><LF>

Field	Description
$	Each NMEA message starts with ‘$’
PMTK	MTK proprietary message
Packet Type	256
Enable	‘0’=Disable ‘1’=Enable
*	End character of data field
Checksum	Hexadecimal checksum
<CR><LF>	Each NMEA message ends with ‘CR’ and ‘LF’

4.16. Packet Type: 285 PMTK_SET_PPS_CONFIG

This message is used to set PPS type.

Data Field:
$PMTK285,Type,PPSPulseWidth
Example:
$PMTK285,4,100*38<CR><LF>
Response:
$PMTK001,285,3*3F<CR><LF>

Field	Description
$	Each NMEA message starts with ‘$’
PMTK	MTK proprietary message
Packet Type	285
Type	‘0’=Disable ‘1’=After the first fix ‘2’=3D fix only ‘3’=2D/3D fix only ‘4’=Always
PPSPulseWidth	2~998 (Unit: ms)
*	End character of data field

Checksum	Hexadecimal checksum
<CR><LF>	Each NMEA message ends with ‘CR’ and ‘LF’

4.17. Packet Type: 286 PMTK_SET_AIC_ENABLED

This message is used to enable or disable AIC function. It is suggested to set cold start command first and then PMTK command.

Data Field:
$PMTK286,Enable
Example:
$PMTK286,0*22<CR><LF>
Response:
$PMTK001,286,3*3C<CR><LF>

Field	Description
$	Each NMEA message starts with ‘$’
PMTK	MTK proprietary message
Packet Type	286
Enable	‘0’=Disable ‘1’=Enable
*	End character of data field
Checksum	Hexadecimal checksum
<CR><LF>	Each NMEA message ends with ‘CR’ and ‘LF’

4.18. Packet Type: 301 PMTK_API_SET_DGPS_MODE

This message is used to configure the source mode of DGPS correction data.

Data Field:
$PMTK301,Mode
Example:
$PMTK301,2*2E<CR><LF>
Response:

$PMTK001,301,3*32<CR><LF>	
Field	**Description**
$	Each NMEA message starts with '$'
PMTK	MTK proprietary message
Packet Type	301
Mode	DGPS data source mode. '0'=No DGPS source '1'=RTCM (Not Supported) '2'=SBAS (Including WAAS/EGNOS/GAGAN/MSAS)
*	End character of data field
Checksum	Hexadecimal checksum
<CR><LF>	Each NMEA message ends with 'CR' and 'LF'

NOTE

Currently mode 1 is not supported.

4.19. Packet Type: 306 PMTK_API_SET_MIN_SNR

This message is used to set the minimum SNR of used satellites. If the minimum SNR threshold value is set, the chip would not use the satellite whose SNR is smaller than it.

Example: $PMTK306,15*1F<CR><LF>	
Field	**Description**
$	Each NMEA message starts with '$'
PMTK	MTK proprietary message
Packet Type	306
MIN_SNR	Minimum SNR threshold of used satellites. Valid range: 9~37
*	End character of data field

Checksum	Hexadecimal checksum
<CR><LF>	Each NMEA message ends with ‘CR’ and ‘LF’

4.20. Packet Type: 308 PMTK_API_SET_DR_LIMIT

This message is used to set the number of estimated fix when entering the tunnel.

Example:
$PMTK308,0*25<CR><LF> => Disable the estimated fix when entering the tunnel.
$PMTK308,3*26<CR><LF> => Keep outputting 3 fix when entering the tunnel.

Field	Description
$	Each NMEA message starts with ‘$’
PMTK	MTK proprietary message
Packet Type	308
DR_LIMIT	Number of estimated fix. Valid range: 0~500
*	End character of data field
Checksum	Hexadecimal checksum
<CR><LF>	Each NMEA message ends with ‘CR’ and ‘LF’

4.21. Packet Type: 311 PMTK_API_SET_ELEV_MASK

This message is used to set satellite elevation mask.

Data Field:
$PMTK311,Type
Example:
$PMTK311,5*28<CR><LF>
Response:
$PMTK001,311,3*33<CR><LF>

Field	Description

$	Each NMEA message starts with '$'
PMTK	MTK proprietary message
Packet Type	311
Satellite Elevation Mask	(Range: 0~90º)
*	End character of data field
Checksum	Hexadecimal checksum
<CR><LF>	Each NMEA message ends with 'CR' and 'LF'

NOTE

The satellite elevation mask is recommended to be set not more than 10 degrees. As with the increase of satellite elevation mask, the number of satellites involved in positioning will decrease.

4.22. Packet Type: 313 PMTK_API_SET_SBAS_ENABLED

This message is used to enable or disable the searching of a SBAS satellite. SBAS (Satellite-Based Augmentation System) is a system that supports wide-area or regional augmentation through the use of geostationary satellite-broadcast messages. The geostationary satellite-broadcast GNSS integrity and correction data are composed of multiple ground stations which are located at accurately surveyed points.

Data Field:
$PMTK313,Enable
Example:
$PMTK313,1*2E<CR><LF>
Response:
$PMTK001,313,3*31<CR><LF>

Field	Description
$	Each NMEA message starts with '$'
PMTK	MTK proprietary message
Packet Type	313
Enable	'0'=Disable '1'=Enable

*	End character of data field
Checksum	Hexadecimal checksum
<CR><LF>	Each NMEA message ends with ‘CR’ and ‘LF’

4.23. Packet Type: 314 PMTK_API_SET_NMEA_OUTPUT

This message is used to set NMEA sentence output frequencies. There are totally 19 data fields that present output frequencies for the 19 supported NMEA sentences individually.

Supported Frequency Settings:

- 0 - Disabled or not supported sentence
- 1 - Output once every one position fix
- 2 - Output once every two position fixes
- 3 - Output once every three position fixes
- 4 - Output once every four position fixes
- 5 - Output once every five position fixes

Data Field:
None
Example:
The module only outputs RMC once every one position fix.
$PMTK314,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0*29<CR><LF>
Response:
$PMTK001,314,3*36<CR><LF>

Field	Description
$	Each NMEA message starts with ‘$’
PMTK	MTK proprietary message
Packet Type	314
0 GLL	GLL interval - Geographic position, latitude and longitude
1 RMC	RMC interval - Recommended minimum specific GNSS sentence
2 VTG	VTG interval - Course over ground and ground speed
3 GGA	GGA interval - GNSS fix data

4 GSA	GSA interval - GNSS DOPS and active satellites
5 GSV	GSV interval - GNSS satellites in view
6 GRS	GRS interval – GNSS range residuals
7 GST	GST interval – GNSS pseudorange error statistics
8 Reserved	Always 0
9 Reserved	Always 0
10 Reserved	Always 0
11 Reserved	Always 0
12 Reserved	Always 0
13 Reserved	Always 0
14 Reserved	Always 0
15 Reserved	Always 0
16 Reserved	Always 0
17 ZDA	ZDA interval - Time and date
18 MCHN	PMTKCHN interval - GNSS channel status
*	End character of data field
Checksum	Hexadecimal checksum
<CR><LF>	Each NMEA message ends with ‘CR’ and ‘LF’

The following messages are be used to restore the system default setting:

Example: $PMTK314,-1*04<CR><LF>	
Field	**Description**
$	Each NMEA message starts with ‘$’
PMTK	MTK proprietary message
Packet Type	314

Restore	Always -1
*	End character of data field
Checksum	Hexadecimal checksum
<CR><LF>	Each NMEA message ends with ‘CR’ and ‘LF’

4.24. Packet Type: 351 PMTK_API_SET_SUPPORT_QZSS_NMEA

The receiver supports new NMEA format for QZSS. The command allows users to enable or disable QZSS NMEA format. QZSS NMEA format is disabled by default and is not supported in **All-in-one** solution.

Data Field:
$PMTK351,Enable
Example:
$PMTK351,1*28<CR><LF>
Response:
$PMTK001,351,3*37<CR><LF>

Field	Description
$	Each NMEA message starts with ‘$’
PMTK	MTK proprietary message
Packet Type	351
QZSS_Enable	‘0’=Disable ‘1’=Enable
*	End character of data field
Checksum	Hexadecimal checksum
<CR><LF>	Each NMEA message ends with ‘CR’ and ‘LF’

4.25. Packet Type: 352 PMTK_API_SET_STOP_QZSS

QZSS is regional positioning service and is used to enable or disable QZSS function. It is enabled by default and is not supported in **All-in-one** solution.

Data Field: $PMTK352,Enable Example: $PMTK352,0*2A<CR><LF> Response: $PMTK001,352,3*34<CR><LF>	
Field	**Description**
$	Each NMEA message starts with '$'
PMTK	MTK proprietary message
Packet Type	352
QZSS_Enable	'0'=Enable '1'=Disable
*	End character of data field
Checksum	Hexadecimal checksum
<CR><LF>	Each NMEA message ends with 'CR' and 'LF'

4.26. Packet Type: 353 PMTK_API_SET_GNSS_SEARCH_MODE

This command is used to configure the receiver to start searching satellite system.

Data Field: $PMTK353,GPS_Enable,Reserved,Reserved,Reserved,BEIDOU_Enable Example: $PMTK353,1,0,0,0,1*2B<CR><LF>: Search GPS+BeiDou Response: $PMTK001,353,3,1,0,0,0,1,17*03<CR><LF>	
Field	**Description**
$	Each NMEA message starts with '$'

PMTK	MTK proprietary message
Packet Type	353
GPS_Enable	'0'=Disable (DO NOT search GPS satellites) '1'or non-ZERO: search GPS satellites
Reserved	Always 0
Reserved	Always 0
Reserved	Always 0
BEIDOU_Enable	'0'=Disable '1'or non-ZERO: search BeiDou satellites
*	End character of data field
Checksum	Hexadecimal checksum
<CR><LF>	Each NMEA message ends with 'CR' and 'LF'

NOTES

1. Actually BeiDou only mode is only for testing purpose. Please use GPS+BeiDou in the real application.
2. For more details about talker ID, please refer to ***Table 9***.

4.27. Packet Type: 386 PMTK_API_SET_STATIC_NAV_THD

This message is used to set the speed threshold for static navigation. If the actual speed is below the threshold, the output position will keep the same and output speed will be zero. This function is disabled if the threshold is set to 0.

Data Field:
$PMTK386,Speed_threshold
Example:
$PMTK386,0.3*3E<CR><LF>
Response:
$PMTK001,386,3*3D<CR><LF>

Field	Description
$	Each NMEA message starts with '$'

PMTK	MTK proprietary message
Packet Type	386
Speed Threshold	0~2m/s
*	End character of data field
Checksum	Hexadecimal checksum
<CR><LF>	Each NMEA message ends with ‘CR’ and ‘LF’

4.28. Packet Type: 400 PMTK_API_Q_FIX_CTL

This message is used to query the rate of position fixing activity.

Refer to PMTK_API_SET_FIX_CTL for setting the rate.

Refer to PMTK_DT_FIX_CTL for the result of the query.

Data Field: None Example: $PMTK400*36<CR><LF>	
Field	**Description**
$	Each NMEA message starts with ‘$’
PMTK	MTK proprietary message
Packet Type	400
*	End character of data field
Checksum	Hexadecimal checksum
<CR><LF>	Each NMEA message ends with ‘CR’ and ‘LF’

4.29. Packet Type: 401 PMTK_API_Q_DGPS_MODE

This message is used to query the setting of DGPS mode.

Refer to PMTK_API_SET_DGPS_MODE for setting the DGPS mode.

Refer to PMTK_DT_DGPS_MODE for the result of the query.

Data Field: None Example: $PMTK401*37<CR><LF>	
Field	**Description**
$	Each NMEA message starts with '$'
PMTK	MTK proprietary message
Packet Type	401
*	End character of data field
Checksum	Hexadecimal checksum
<CR><LF>	Each NMEA message ends with 'CR' and 'LF'

4.30. Packet Type: 413 PMTK_API_Q_SBAS_ENABLED

This message is used to query the setting of SBAS.

Refer to PMTK_API_SET_SBAS_ENABLE for SBAS setting.

Refer to PMTK_DT_SBAS_ENABLED for the result of the query.

Data Field: None Example: $PMTK413*34<CR><LF>	
Field	**Description**
$	Each NMEA message starts with '$'

PMTK	MTK proprietary message
Packet Type	413
*	End character of data field
Checksum	Hexadecimal checksum
<CR><LF>	Each NMEA message ends with ‘CR’ and ‘LF’

4.31. Packet Type: 414 PMTK_API_Q_NMEA_OUTPUT

This message is used to query the current NMEA sentence output frequencies.

Refer to PMTK_API_SET_NMEA_OUTPUT for the frequencies setting.
Refer to PMTK_DT_NMEA_OUTPUT for the result of the query.

Data Field: None Example: $PMTK414*33<CR><LF>	
Field	**Description**
$	Each NMEA message starts with ‘$’
PMTK	MTK proprietary message
Packet Type	414
*	End character of data field
Checksum	Hexadecimal checksum
<CR><LF>	Each NMEA message ends with ‘CR’ and ‘LF’

4.32. Packet Type: 458 PMTK_API_GET_POS_XYZ

This message is used to return the WGS84 ECEF XYZ Cartesian position vector (metres) with an estimated 1-sigma accuracy.

Example: $PMTK458*3B<CR><LF>	
Field	**Description**
$	Each NMEA message starts with '$'
PMTK	MTK proprietary message
Packet Type	458
*	End character of data field
Checksum	Hexadecimal checksum
<CR><LF>	Each NMEA message ends with 'CR' and 'LF'

4.33. Packet Type: 461 PMTK_API_GET_VEL_XYZ

This message is used to return the WGS84 ECEF XYZ Cartesian velocity vector (m/s) with an estimated 1-sigma accuracy.

Example: $PMTK461*31<CR><LF>	
Field	**Description**
$	Each NMEA message starts with '$'
PMTK	MTK proprietary message
Packet Type	461
*	End character of data field
Checksum	Hexadecimal checksum
<CR><LF>	Each NMEA message ends with 'CR' and 'LF'

4.34. Packet Type: 500 PMTK_DT_FIX_CTL

This message is the response to PMTK_API_Q_FIX_CTL.

Data Field: $PMTK500,Fix interval Example: $PMTK500,1000,0,0,0,0*1A<CR><LF>	
Field	**Description**
$	Each NMEA message starts with '$'
PMTK	MTK proprietary message
Packet Type	500
Fix Interval	Position fix interval (msec). Range: 100~10000
Reserved	Always 0
Reserved	Always 0
Reserved	Always 0
Reserved	Always 0
*	End character of data field
Checksum	Hexadecimal checksum
<CR><LF>	Each NMEA message ends with 'CR' and 'LF'

4.35. Packet Type: 501 PMTK_DT_DGPS_MODE

This message is the response to PMTK_API_Q_DGPS_MODE.

Data Field: $PMTK501,Mode Example: $PMTK501,1*2B<CR><LF>	
Field	**Description**

$	Each NMEA message starts with ‘$’
PMTK	MTK proprietary message
Packet Type	501
Mode	DGPS data source mode ‘0’=No DGPS source ‘1’=RTCM ‘2’=SBAS
*	End character of data field
Checksum	Hexadecimal checksum
<CR><LF>	Each NMEA message ends with ‘CR’ and ‘LF’

4.36. Packet Type: 513 PMTK_DT_SBAS_ENABLED

This message is the response to PMTK_API_Q_SBAS_ENABLED.

Data Field: $PMTK513,Enable Example: $PMTK513,1*28<CR><LF>	
Field	**Description**
$	Each NMEA message starts with ‘$’
PMTK	MTK proprietary message
Packet Type	513
Enable	‘0’=Disable ‘1’=Enable
*	End character of data field
Checksum	Hexadecimal checksum
<CR><LF>	Each NMEA message ends with ‘CR’ and ‘LF’

4.37. Packet Type: 514 PMTK_DT_NMEA_OUTPUT

This message is the response to PMTK_API_Q_NMEA_OUTPUT.

Data Field:
None
Example:
$PMTK514,1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0*33<CR><LF>

Field	Description
$	Each NMEA message starts with '$'
PMTK	MTK proprietary message
Packet Type	514
0 GLL	GLL interval - Geographic position, latitude and longitude
1 RMC	RMC interval - Recommended minimum specific GNSS sentence
2 VTG	VTG interval - Course over ground and ground speed
3 GGA	GGA interval - GNSS fix data
4 GSA	GSA interval - GNSS DOPS and active satellites
5 GSV	GSV interval - GNSS satellites in view
6 Reserved	GRS interval – GNSS range residuals
7 Reserved	GST interval – GNSS pseudorange error statistics
8 Reserved	
9 Reserved	
10 Reserved	
11 Reserved	
12 Reserved	
13 Reserved	
14 Reserved	
15 Reserved	

16 Reserved	
17 ZDA	ZDA interval - Time and date
18 Reserved	PMTKCHN interval - GNSS channel status
*	End character of data field
Checksum	Hexadecimal checksum
<CR><LF>	Each NMEA message ends with ‘CR’ and ‘LF’

4.38. Packet Type: 605 PMTK_Q_RELEASE

This message is used to query the firmware release information.

Refer to PMTK_DT_RELEASE for the result of the query.

Data Field: None Example: $PMTK605*31<CR><LF>	
Field	**Description**
$	Each NMEA message starts with ‘$’
PMTK	MTK proprietary message
Packet Type	605
*	End character of data field
Checksum	Hexadecimal checksum
<CR><LF>	Each NMEA message ends with ‘CR’ and ‘LF’

4.39. Packet Type: 607 PMTK_Q_EPO_INFO

This command is used to query the EPO data status stored in the GNSS chip.

Data Field: None Example: $PMTK607*33<CR><LF> Response: PMTK_DT_EPO_INFO	
Field	**Description**
$	Each NMEA message starts with '$'
PMTK	MTK proprietary message
Packet Type	607
*	End character of data field
Checksum	Hexadecimal checksum
<CR><LF>	Each NMEA message ends with 'CR' and 'LF'

4.40. Packet Type: 622 PMTK_Q_LOCUS_DATA

This message is used to dump locus flash data and is not supported in **All-in-one** solution.

Data Field: $PMTK622,Type Example: $PMTK622,1*29<CR><LF> Response: $PMTK001,622,3*36<CR><LF>	
Field	**Description**
$	Each NMEA message starts with '$'
PMTK	MTK proprietary message
Packet Type	622
Type	'1'=Dump partial in used LOCUS flash data.
*	End character of data field
Checksum	Hexadecimal checksum

<CR><LF>	Each NMEA message ends with ‘CR’ and ‘LF’

4.41. Packet Type: 705 PMTK_DT_RELEASE

This message is the response to PMTK_Q_RELEASE.

Data Field: $PMTK705,Release string,Build ID,Product Model (SDK Version) Example: $PMTK705,AXN_5.10_3333_17062100,0001,MC20-GNSS,1.0*27<CR><LF>	
Field	**Description**
$	Each NMEA message starts with ‘$’
PMTK	MTK proprietary message
Packet Type	705
Release String	Firmware release name and version 3318: Mcore_x.x 3329: AXN_x.x 3339: AXN_x.x 3333: AXN_x.x 3337: AXN_x.x
Build ID	Build ID set in CoreBuilder for firmware version control
Product Model	Product model set in CoreBuilder for product identification
SDK Version (Optional)	Showing SDK version if the firmware is used for SDK
*	End character of data field
Checksum	Hexadecimal checksum
<CR><LF>	Each NMEA message ends with ‘CR’ and ‘LF’

4.42. Packet Type: 707 PMTK_DT_EPO_INFO

This response packet contains EPO data status stored in GNSS chip.

Data Field: $PMTK707,Set,FWN,FTOW,LWN,LTOW,FCWN,FCTOW,LCWN,LCTOW*CS<CR><LF> Example: $PMTK707,56,1468,172800,1470,151200,1468,259200,1468,259200*1F<CR><LF> Response: PMTK_DT_EPO_INFO	
Field	**Description**
$	Each NMEA message starts with '$'
PMTK	MTK proprietary message
Packet Type	707
Set	Total number sets of EPO data stored in GNSS chip
FWN	GPS week number of the first set of EPO data stored in GNSS chip
FTOW	GPS week TOW of the first set of EPO data stored in GNSS chip
LWN	GPS week number of the last set of EPO data stored in GNSS chip
LTOW	GPS week TOW of the last set of EPO data stored in GNSS chip
FCWN	GPS week number of the first set of EPO data that are currently used
FCTOW	GPS week TOW of the first set of EPO data that are currently used
LCWN	GPS week number of the last set of EPO data that are currently used
LCTOW	GPS week TOW of the last set of EPO data that are currently used
*	End character of data field
Checksum	Hexadecimal checksum
<CR><LF>	Each NMEA message ends with 'CR' and 'LF'

4.43. Packet Type: 838 PMTK_TEST_ANTI_SPOOFING

This message is used to enable or disable jamming detection function.

Data Field: $PMTK838,CmdType

Example:
$PMTK838,1*2C<CR><LF>
Response:
$PMTK001,838,3,1*2E<CR><LF>
Return:
$PMTKSPF,1*5A => No jamming, healthy status (status 1)
$PMTKSPF,2*59 => Warning status (status 2)
$PMTKSPF,3*58 => Critical status (status 3)

Field	Description
$	Each NMEA message starts with '$'
PMTK	MTK proprietary message
Packet Type	838
CmdType	'0'=Disable jamming detection function '1'=Enable jamming detection function
*	End character of data field
Checksum	Hexadecimal checksum
<CR><LF>	Each NMEA message ends with 'CR' and 'LF'

NOTE

After jamming detection is enabled, the module starts to detect whether there is jamming.

1. If there is no jamming, "$PMTKSPF,1*5A" will be reported to indicate healthy status (status 1).
2. If there is continuous jamming, then the module status will change from 1 to 2 and finally 3.
   - In the case of not being positioned: after jamming detection is enabled, the module status will be 1 at the very beginning, and then change to 2 when jamming is detected. During the process, the module will attempt to fix position. If it still fails in positioning after 200s, the module status will change to 3 finally.
   - In the case of being positioned: after jamming detection is enabled, the module status will be 1 at the very beginning. When jamming is detected, the module status will change to 2 and then 3 consecutively.

4.44. Packet Type: 869 PMTK_EASY_ENABLE

This message is used to enable or disable EASY™ function, and query whether EASY™ is enabled or disabled.

Data Field: $PMTK869,CmdType[,Enabled] Example: $PMTK869,1,1*35<CR><LF> Response: $PMTK001,869,3*37<CR><LF>	
Field	**Description**
$	Each NMEA message starts with '$'
PMTK	MTK proprietary message
Packet Type	869
CmdType	'0'=Query '1'=Set '2'=Result for query operation
Enabled	'0'=Disable '1'=Enable
*	End character of data field
Checksum	Hexadecimal checksum
<CR><LF>	Each NMEA message ends with 'CR' and 'LF'

NOTES

1. If EASY™ is disabled, the receiver returns:
   $PMTK869,2,0,0*2B<CR><LF>
2. If EASY™ is enabled and is not finished yet, the receiver may return:
   $PMTK869,2,1,0*2A<CR><LF>
3. If EASY™ is enabled and is finished after 1 day, the receiver may return:
   $PMTK869,2,1,1*2B<CR><LF>
4. If EASY™ is enabled and is finished after 2 days, the receiver may return:
   $PMTK869,2,1,2*28<CR><LF>
5. If EASY™ is enabled and is finished after 3 days, the receiver may return:
   $PMTK869,2,1,3*29<CR><LF>

4.45. Packet Type: 875 PMTK_PMTKLSC_STN_OUTPUT

This message is used to enable or disable PMTKLSC sentence output and query whether PMTKLSC sentence output is enabled or disabled. This command is not supported in **All-in-one** solution.

Data Field:
$PMTK875,CmdType[,Enabled]
Example:
$PMTK875,1,1*38<CR><LF>: Enable PMTKLSC and PMTKLSCB sentence output
Response:
$PMTKLSC,Parameter1,Parameter2,Parameter3*CS
$PMTKLSB,Parameter1,Parameter2,Parameter3*CS
Where Parameter1: current leap second
Parameter2: leap indicator, 1 means updated from broadcast data
Parameter3: next leap second

Field	Description
$	Each NMEA message starts with '$'
PMTK	MTK proprietary message
Packet Type	875
CmdType	'0'=Query '1'=Set '2'=Result for query operation
Enabled	'0'=Disable '1'=Enable
*	End character of data field
Checksum	Hexadecimal checksum
<CR><LF>	Each NMEA message ends with 'CR' and 'LF'

4.46. Packet Type: 886 PMTK_FR_MODE

This message is used to set navigation mode.

Data Field:
$PMTK886,CmdType
Example:
$PMTK886,3*2B<CR><LF>
Response:
$PMTK001,886,3*36

Field	Description
$	Each NMEA message starts with '$'
PMTK	MTK proprietary message
Packet Type	886
CmdType	'0'=Normal mode: For general purpose '1'=Fitness mode: For running and walking purpose that the low-speed (<5m/s) movement will have more effect on the position calculation. '2'=Aviation mode: For high-dynamic purpose that the large-acceleration movement will have more effect on the position calculation. '3'=Balloon mode: For high-altitude balloon purpose that the vertical movement will have more effect on the position calculation. '4'=Stationary mode: For stationary applications that zero dynamics is assumed.
*	End character of data field
Checksum	Hexadecimal checksum
<CR><LF>	Each NMEA message ends with 'CR' and 'LF'

NOTE

Each mode has its altitude limitation. Please choose the appropriate mode base on the table below. If the test scenario exceeds the limitation, the position calculation will be incorrect.

Mode	Altitude Limitation
Normal mode	10000m
Fitness mode	10000m
Aviation mode	10000m
Stationary mode	10000m
Balloon mode	80000m

5 SDK NMEA Packet Protocol

This chapter introduces the SDK NMEA packet protocol, which is a set of extension messages of standard NMEA packet protocol. These messages are used to control and configure the internal GNSS engine of MC20&MC30. The following table shows the structure of SDK NMEA packet. The SDK NMEA packet is sent through AT commands.

5.1. PQBAUD Change NMEA Port Default Baud Rate

PQBAUD Change NMEA Port Default Baud Rate	
Write Command **$PQBAUD,W,<baudrate>*Checksum<CR><LF>**	Response **$PQBAUD,W,OK*Checksum<CR><LF>** If there is any error, response: **$PQBAUD,W,ERROR*Checksum<CR><LF>**
Reference	

Parameter

<baudrate>	NMEA port baud rate, default value is 115200bps, and can be set to 4800bps,9600bps 14400bps, 19200bps, 38400bps and 57600bps in **Stand-alone** solution

Example

```
$PQBAUD,W,38400*7B      //Change NMEA port default baud rate to 38400bps
$PQBAUD,W,OK*40         //Set OK
```

NOTES

1. The command will be valid in **Stand-alone** solution only.
2. The command will be effective immediately after setting.
3. Parameter is automatically saved.
4. If the baud rate is changed, then there is no response returned in the current baud rate.

5.2. PQEPE Enable/Disable PQEPE Sentence Output

PQEPE Enable/Disable PQEPE Sentence Output	
Write Command **$PQEPE,W,<mode>,<save>*Checksum<CR><LF>**	Response **$PQEPE,W,OK*Checksum<CR><LF>** If there is any error, response: **$PQEPE,W,ERROR*Checksum<CR><LF>**
URC Message	**$PQEPE,<EPE_hori>,<EPE_vert>*Checksum<CR><LF>** Parameter **<EPE_hori>** Estimated horizontal position error **<EPE_vert>** Estimated vertical position error Example **$PQEPE,5.3050,3.2000*53**
Reference	

Parameter

<mode>	Operation 0 Disable the URC including EPE data 1 Enable the URC including EPE data
<save>	Save operation 0 Parameter is not saved, and ineffective after restart 1 Parameter is saved in flash, and effective after restart

Example

```
$PQEPE,W,1,1*2A          //Enable the URC including EPE data, and save parameters into flash
$PQEPE,W,OK*02           //Set OK
```

NOTE

The command will be effective immediately after setting.

5.3. PQECEF Enable/Disable ECEFPOSVEL Sentence Output

PQECEF Enable/Disable ECEFPOSVEL Sentence Output	
Write Command **$PQECEF,W,<mode>,<save>*ChkSum<CR><LF>**	Response **$PQECEF,W,OK*ChkSum<CR><LF>** If there is any error, response: **$PQECEF,W,ERROR*ChkSum<CR><LF>**
Read Command **$PQECEF,R*ChkSum<CR><LF>**	Response **$PQECEF,R,<mode>*ChkSum<CR><LF>**
URC Message	**$ECEFPOSVEL,<time>,<x>,<y>,<z>,<v_x>,<v_y>,<v_z>*ChkSum<CR><LF>** Parameter **<time>** UTC from the internal real-time clock **<x>** The value of X axis in ECEF **<y>** The value of Y axis in ECEF **<z>** The value of Z axis in ECEF **<v_x>** Velocity component of X axis in ECEF **<v_y>** Velocity component of Y axis in ECEF **<v_z>** Velocity component of Z axis in ECEF Example **$ECEFPOSVEL,052743.000,-1526672.867459,6191083.982801,143008.780911,0,0,0*14**
Reference	

Parameter

<mode>	Operation 0 Disable ECEFPOSVEL sentence output 1 Enable ECEFPOSVEL sentence output
<save>	Save operation 0 Parameter is not saved, and ineffective after restart 1 Parameter is saved in flash, and effective after restart

Example

```
$PQECEF,W,1,1*7F          //Enable ECEFPOSVEL sentence output, and save the parameter into flash
$PQECEF,W,OK*57           //Set OK
$PQECEF,R*7A              //Read mode
$PQECEF,R,1*67            //Read OK, ECEFPOSVEL sentence output is enabled
```

NOTE

The command will be effective immediately after setting.

5.4. PQODO Start/Stop Odometer Reading

PQODO Start/Stop Odometer Reading	
Write Command **$PQODO,W,<mode>[,<initial distance>]*ChkSum<CR><LF>**	Response **$PQODO,W,OK*ChkSum<CR><LF>** If there is any error, response: **$PQODO,W,ERROR*ChkSum<CR><LF>**
Read Command **$PQODO,R*ChkSum<CR><LF>**	Response **$PQODO,R,<mode>*ChkSum<CR><LF>** If there is any error, response: **$PQODO,R,ERROR*ChkSum<CR><LF>**
Query Command **$PQODO,Q*ChkSum<CR><LF>**	Response **$PQODO,Q,<distance>*ChkSum<CR><LF>** If there is any error, response **$PQODO,Q,ERROR*ChkSum<CR><LF>**
Reference	

Parameter

<mode>	Start or stop odometer reading 0 Stop odometer reading and remember the distance value 1 Start odometer reading and initialize the distance according to the **<initial distance>**
<initial distance>	Set the initial distance, range: 0-1e09, unit: meter When **<mode>** is 1, this parameter can be omitted, and its default value is 0 When **<mode>** is 0, this parameter must be omitted
<distance>	Current distance. Unit: meter

Example

```
$PQODO,W,1*23          //Start odometer reading, and initial distance is 0m.
$PQODO,W,OK*16         //Set OK
```

$PQODO,W,1,1000000*3E	//Start odometer reading, and initial distance is 1,000,000m.
$PQODO,W,OK*16	//Set OK
$PQODO,R*3B	//Read mode
$PQODO,R,1*26	//Read OK, odometer reading has already been started
$PQODO,Q*38	//Query the distance value
$PQODO,Q,123.45*0B	//Current distance value is returned

NOTES

1. The command will be effective immediately after setting.
2. After module is restarted, the **PQODO** write command must be executed again to re-start odometer reading.
3. The command is not supported in backup mode.

5.5. PQPZ90 Enable/Disable Switching from WGS84 to PZ-90.11

PQPZ90 Enable/Disable Switching from WGS84 to PZ-90.11	
Write Command **$PQPZ90,W,<mode>,<save>*ChkSum<CR><LF>**	Response **$PQPZ90,W,OK*ChkSum<CR><LF>** If there is any error, response: **$PQPZ90,W,ERROR*ChkSum<CR><LF>**
Read Command **$PQPZ90,R*ChkSum<CR><LF>**	Response **$PQPZ90,R,<mode>*ChkSum<CR><LF>**
URC Message	**$xxDTM,P90,x,xx.xxxx,x,xx.xxxx,x,xxx,W84*hh<CR><LF>** Parameter definition is available in ***Table 8***.
Reference	

Parameter

<mode>	Operation 0 Disable switching from WGS84 to PZ-90.11 1 Enable switching from WGS84 to PZ-90.11
<save>	Save operation 0 Parameter is not saved, and ineffective after restart 1 Parameter is saved in flash, and effective after restart

Example

```
$PQPZ90,W,1,1*79        //Enable switching from WGS84 to PZ-90.11, and save the parameter into flash
$PQPZ90,W,OK*51         //Set OK
$PQPZ90,R*7C            //Read mode
$PQPZ90,R,0*60          //Read OK, switching from WGS84 to PZ-90.11 is enabled
```

NOTES

1. The command will be effective immediately.
2. If switching from WGS84 to PZ-90.11 is enabled and effective, the coordinate values in RMC and GGA sentences will be switched to PZ-90.11 after fixing. Also, a DTM sentence will be displayed to identify the datum used.

5.6. PQGLP Set the Module into GLP Mode

PQGLP Set the Module into GLP Mode	
Write Command **$PQGLP,W,<mode>,<save>*ChkSum<CR><LF>**	Response **$PQGLP,W,OK*ChkSum<CR><LF>** If there is any error, response: **$PQGLP,W,ERROR*ChkSum<CR><LF>**
Read Command **$PQGLP,R*ChkSum<CR><LF>**	Response **$PQGLP,R,<mode>*ChkSum<CR><LF>**
Reference	

Parameter

<mode>	Module operation mode
	0 Normal mode
	1 GLP mode
<save>	Save operation
	0 Parameter is not saved, and ineffective after restart
	1 Parameter is saved in flash, and effective after restart

Example

```
$PQGLP,W,1,1*21         //Change to GLP mode
$PQGLP,W,OK*09          //Set OK
```

$PQGLP,R*24	//Read mode
$PQGLP,R,1*39	//Read OK, GLP mode is enabled

NOTES

1. The command will be effective immediately after setting.
2. For more details about the usage of PQGLP command, please refer to ***document [4]***.

5.7. PQVEL Enable/Disable 3 Ways Velocity Sentence Output

PQVEL Enable/Disable 3 Ways Velocity Sentence Output	
Write Command **$PQVEL,W,<mode>,<save>*ChkSum<CR><LF>**	Response **$PQVEL,W,OK*ChkSum<CR><LF>** If there is any error, response: **$PQVEL,W,ERROR*ChkSum<CR><LF>**
Read Command **$PQVEL,R*ChkSum<CR><LF>**	Response **$PQVEL,R,<mode>*ChkSum<CR><LF>**
URC Message	**$PQVEL,<north_vel>,<east_vel>,<down_vel>*ChkSum<CR><LF>** Parameter **<north_vel>** North velocity **<east_vel>** East velocity **<down_vel>** Down velocity Example **$PQVEL,1.000000,2.000000,-0.000000*42**
Reference	

Parameter

<mode>	Enable/disable 3 ways velocity sentence output
	0 Disable
	1 Enable
<save>	Save operation
	0 Parameter is not saved, and ineffective after restart
	1 Parameter is saved in flash, and effective after restart

Example

```
$PQVEL,W,1,1*25          //Enable 3 ways velocity sentence output
$PQVEL,W,OK*0D           //Set OK

$PQVEL,R*20              //Read mode
$PQVEL,R,1*3D            //Read OK, 3 ways velocity sentence output is enabled
```

NOTE

The command will be effective immediately after setting.

5.8. PQJAM Enable/Disable Jamming Detection Function

PQJAM Enable/Disable Jamming Detection Function	
Write Command **$PQJAM,W,<mode>,<save>*ChkSum<CR><LF>**	Response **$PQJAM,W,OK*ChkSum<CR><LF>** If there is any error, response: **$PQJAM,W,ERROR*ChkSum<CR><LF>**
Read Command **$PQJAM,R*ChkSum<CR><LF>**	Response **$PQJAM,R,<mode>*ChkSum<CR><LF>**
URC Message	**$PMTKSPF, <status>*ChkSum<CR><LF>** Parameter **<status>** Jamming status 1 No jamming, healthy status 2 Warning status 3 Critical status Example **$PMTKSPF,1*5A**
Reference	

Parameter

<mode>	Operation mode 0 Disable jamming detection function 1 Enable jamming detection function

<save>	Save operation 0 Parameter is not saved, and ineffective after restart 1 Parameter is saved in flash, and effective after restart

Example

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

$PQJAM,R*39              //Read operation mode
$PQJAM,R,1*24            //Read OK
```

NOTES

1. The command will be effective immediately after setting.
2. The command is not supported in backup mode.

5.9. PQRLM Enable/Disable Return Link Message Output

PQRLM Enable/Disable Return Link Message Output	
Write Command **$PQRLM,W,<mode>,<save>*ChkSum<CR><LF>**	Response **$PQRLM,W,OK*ChkSum<CR><LF>** If there is any error, response: **$PQRLM,W,ERROR*ChkSum<CR><LF>**
Read Command **$PQRLM,R*ChkSum<CR><LF>**	Response **$PQRLM,R,<mode>*ChkSum<CR><LF>**
URC Message	**$PQRLM,<beacon_id>,<gps_second>,<msg_code>,<para>*ChkSum<CR><LF>** Parameter **<beacon_id>** Beacon ID of RLM **<gps_second>** The GPS second when receiving RLM **<msg_code>** Message code **<para>** The data parameters provided by RLS. (short message contains 4 hex characters, long message contains 24 hex characters) Example **$PQRLM,0a0a0a0a0a0a0a8,955065733,1,8aa1*3E** **$PQRLM,050505050505052,955065709,15,5555555555555**

	55555
Reference	

Parameter

<mode>	Enable/disable return link message output
	0 Disable
	1 Enable
<save>	Save operation
	0 Parameter is not saved, and ineffective after restart
	1 Parameter is saved in flash, and effective after restart

Example

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

$PQRLM,R*2C              //Read mode
$PQRLM,R,1*31            //Read OK
```

NOTE

The command will be effective immediately after setting.

5.10. PQGEO Configure Parameters of Geo-fence

PQGEO Configure Parameters of Geo-fence	
Write Command **$PQGEO,W,<GEO ID>,<mode>,<shape>,<latitude0>,<longitude0>,<latitude1/radius>,<longitude1>,<latitude2>,<longitude2>,<latitude3>,<longitude3>*ChkSum<CR><LF>**	Response **$PQGEO,W,OK*ChkSum<CR><LF>** If there is any error, response: **$PQGEO,W,ERROR*ChkSum<CR><LF>**
Read Command **$PQGEO,R,<GEO ID>*ChkSum<CR><LF>**	Response **$PQGEO,R,<GEO ID>,<mode>,<shape>,<latitude0>,<longitude0>,<latitude1/radius>,<longitude1>,<latitude2>,<longitude2>,<latitude**

	3>,<longitude3>*ChkSum<CR><LF> If there is any error, response: **$PQGEO,R,ERROR*ChkSum<CR><LF>**
Inquire Command **$PQGEO,Q,<GEO ID>*ChkSum<CR><LF>**	Response **$PQGEO,Q,<GEO ID>,<status>*ChkSum<CR><LF>** If there is any error, response: **$PQGEO,Q,ERROR*ChkSum<CR><LF>**
URC Message	**$PQGEO,<GEO ID>,<action>,<Fix status>,<UTC date & Time>,<Latitude>,<Longitude>,<MSL Altitude>,<Speed Over Ground>,<Course Over Ground>,<Fix Mode>,<Reserved1>,<HDOP>,<PDOP>,<VDOP>,<Reserved2>,<GPS Satellites in View>,<GPS Satellites Used>,*ChkSum<CR><LF>** Parameters **<GEO ID>** Geo-fence ID is from 0 to 9. **<action>** The current action of the module 1 Enters a geo-fence 2 Leaves a geo-fence **<Fix status>** Fix status 0 No fix 1 2D fix 2 3D fix **<UTC date & Time>** UTC time Format: DyyMMddThhmmss.sss D: char 'D', refers to date yy: current year-2000 MM: 1-12 dd: 1-31 T: char 'T', refers to time hh: 0-23 mm: 0-59 ss.sss: 00.000-59.9999 **<Latitude>** The latitude of current position Unit: degree Format: ±dd.dddddd Range: -90.000000~90.000000 **<Longitude>** The longitude of current position Unit: degree Format: ±ddd.dddddd Range: -180.000000~180.000000 **<MSL Altitude>** Mean sea level (MSL) altitude

	Unit: meter **<Speed Over Ground>** Speed over ground. Unit: km/h **<Course Over Ground>** Course over ground Unit: degree Range: 0-360.00 **<Fix Mode>** Fix mode 0 No fix 1 Estimated mode 2 Position fixed 3 Position fixed in DGPS mode **<Reserved1>** Reserved1 **<HDOP>** Horizontal dilution of precision **<PDOP>** Position dilution of precision **<VDOP>** Vertical dilution of precision **<Reserved2>** Reserved2 **<GPS Satellites in View>** GPS satellites in view **<GPS Satellites Used>** GPS satellites used Example **$PQGEO,0,1,2,D150506T070127.000,31.856038,117.19711 0,49.4,14.92,0.18,2,,1.11,2.95,2.74,,14,9*5D**
Reference	

Parameter

<GEO ID>	Geo-fence ID, range: 0-9
<mode>	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 If **<mode>** is 0, the parameters after **<mode>** can be omitted.
<shape>	Fence shape 0 Circularity with center and radius 1 Circularity with center and one point on the circle 2 Triangle 3 Quadrangle
<latitude0>	The latitude of a point which is defined as the center of the geo-fence circular region or the first point, Unit: degree Format: ±dd.dddddd, range: -90.000000~90.000000
<longitude0>	The longitude of a point which is defined as the center of the geo-fence circular region or the first point. Unit: degree Format: ±ddd.dddddd, range: -180.000000~180.000000

<latitude1/radius>	When <shape> is 0, this parameter is radius, unit: meter, range: 0-6000000.0 When <shape> is other values, this parameter is latitude1, Unit: degree Format: ±dd.dddddd, range: -90.000000~90.000000 If <shape> is 0, the parameters after <latitude1/radius> must be omitted.
<longitude1>	The longitude of the second point. Unit: degree Format: ±ddd.dddddd, range: -180.000000~180.000000 If <shape> is 1, the parameters after <longitude1> must be omitted.
<latitude2>	The latitude of the third point. Unit: degree Format: ±dd.dddddd, range: -90.000000~90.000000
<longitude2>	The longitude of the third point. Unit: degree Format: ±ddd.dddddd, range: -180.000000~180.000000 If <shape> is 2, the parameters after <longitude2> must be omitted.
<latitude3>	The latitude of the fourth point. Unit: degree Format: ±dd.dddddd, range: -90.000000~90.000000
<longitude3>	The longitude of the fourth point. Unit: degree Format: ±ddd.dddddd, range: -180.000000~180.000000
<status>	The status of current position. 0 Unknown position 1 Inside the geo-fence 2 Outside the geo-fence

Example

```
$PQGEO,W,0,1,0,31.85913,117.1933,500.0*26                    //<shape> is 0.
$PQGEO,W,OK*1F
$PQGEO,W,4,3,1,31.91133,117.1129,31.994856,117.070281*1C     //<shape> is 1.
$PQGEO,W,OK*1F
$PQGEO,W,3,0*34                                              //Delete geo-fence 3.
$PQGEO,W,OK*1F

$PQGEO,R,0*2E
$PQGEO,R,0,1,0,31.859130,117.193300,500.0*13
$PQGEO,R,4*2A
$PQGEO,R,4,3,1,31.911330,117.112900,31.994856,117.070281*29

$PQGEO,Q,0*2D
$PQGEO,Q,0,1*30
$PQGEO,Q,4*29
$PQGEO,Q,4,2*37
```

NOTES

1. If **<mode>** is 0 and there are no parameters after **<mode>**, this command can delete the geo-fence.
2. $PQGEO,R,10*1F command can inquire parameters of all geofences.

3. The command will be effective immediately after setting, and the parameters will be automatically saved into flash.
4. Input the latitude and longitude in sequence in clockwise or counter-clockwise order.

6 Default Configurations

Table 3: Default Configurations

Item	Default
NMEA Port Baud Rate	115200bps
Datum	WGS84
Rate of Position Fixing	1Hz
DGPS Mode	SBAS
SBAS	Enabled
NMEA Output Messages	GGA, RMC, GSA, GSV, VTG and GLL
AIC	On
EASY™	On

7 Appendix A References

Table 4: Related Documents

SN	Document Name	Remark
[1]	Quectel_MC20_Series_Hardware_Design	MC20 Series Hardware Design
[2]	Quectel_MC30_Hardware_Design	MC30 Hardware Design
[3]	Quectel_GNSS_SDK_Commands_Manual	GNSS SDK Commands Manual
[4]	Quectel_GNSS_Low_Power_Mode_Application_Note	GNSS Low Power Mode Application Note
[5]	Quectel_MC20&MC30_GNSS_AGPS_Application_Note	MC20&MC30 GNSS AGPS Application Note

Table 5: Terms and Abbreviations

Abbreviation	Description
AGPS	Assisted Global Positioning System
AIC	Active Interference Cancellation
CS	Commercial Sample
DGPS	Differential Global Positioning System
EASY	Embedded Assist System
GBS	NMEA: GPS Satellite Fault Detection
GGA	NMEA: Global Positioning System Fix Data
GLL	NMEA: Geographic Latitude and Longitude
GLONASS	GLObalnaya NAvigatsionnaya Sputnikovaya Sistema, the Russian Global Navigation Satellite System
GNSS	Global Navigation Satellite System

BeiDou	Chinese Satellite Navigation System
GPS	Global Positioning System
GSA	NMEA: GNSS DOP and Active Satellites
GSV	NMEA: GNSS Satellites in View
HDOP	Horizontal Dilution of Precision
MP	Mass Production
NMEA	National Marine Electronics Association
NITZ	Network Identity and Time Zone
PDOP	Position Dilution of Precision
PMTK	Private Protocol of MTK
RMC	NMEA: Recommended Minimum Position Data
SBAS	Satellite-Based Augmentation System
UTC	Universal Time Coordinated
VDOP	Vertical Dilution of Precision
VTG	NMEA: Track Made Good and Ground Speed
WAAS	Wide Area Augmentation System

Table 6: Structure of NMEA Message

Filed	Length (Bytes)	Description
$	1	Each NMEA message starts with ‘$’
Talker ID	1~2	Talker IDs can be ‘GP’, ‘GN’ and ‘BD’ when the message ID is RMC, VTG, GLL or GGA; Talker IDs can be ‘GP’, ‘GN’, ‘BD’ and ‘GA’ when the message ID is GSA; Talker IDs can be ‘GP’, ‘GL’, ‘BD’ and ‘GA’ when the message ID is GSV, and Talker IDs can be ‘GN’ when the message ID is GBS.
NMEA Message ID	3	NMEA message ID
Data Field	Variable, depends on the	Data fields, delimited by comma ‘,’

	NMEA message type	
*	1	End character of data field
Checksum	2	A hexadecimal number calculated by exclusive OR of all characters between '$' and '*'
<CR><LF>	2	Each NMEA message ends with 'CR' and 'LF'

NOTE

The default output messages of MC20&MC30 are as following six sentences: RMC, VTG, GGA, GSA, GSV and GLL.

Table 7: Structure of MTK NMEA Packet

Filed		Length (Bytes)	Description
$		1	Each NMEA message starts with '$'
Talker ID		1	'P' for proprietary message
NMEA	Data Type	3	Always 'MTK' to indicate MTK proprietary message
Data Filed	Packet Type	3	Packet type, from '000' to '999'
	Packet Data	Variable, depends on the packet type	Data fields, delimited by comma ','
*		1	End character of data field
Checksum		2	A hexadecimal number calculated by exclusive OR of all characters between '$' and '*'
<CR><LF>		2	Each NMEA message ends with 'CR' and 'LF'

Table 8: Datum Sentence Definition

$xxDTM,P90,x,xx.xxxx,x,xx.xxxx,x,xxx,W84*hh<CR><LF>	
Field	**Meaning**
1	Talker ID, refer to ***Table 9***
2	Local datum code (xxx): W84 - WGS84 P90 - PZ-90

3	Local datum sub code (x)
4	Latitude offset in minutes (xx.xxxx)
5	Latitude offset mark (N: +, S: -) (x)
6	Longitude offset in minutes (xx.xxxx)
7	Longitude offset mark (E: +, W: -) (x)
8	Altitude offset in meters.
9	Datum (xxx): W84 - WGS84 P90 - PZ-90
10	Checksum

Table 9: Talker ID Display in Different GNSS System (for NMEA 0183 3.01 Version)

Talker ID	GPS Only	BeiDou Only	GLONASS Only	Galileo Only	GPS+ GLONASS	GPS+ BeiDou	GPS+ Galileo	GPS+GLONASS +Galileo
GGA	GP	BD	GL	GA	GN	GN	GN	GN
RMC	GP	BD	GL	GA	GN	GN	GN	GN
GLL	GP	BD	GL	GA	GN	GN	GN	GN
VTG	GP	BD	GL	GA	GN	GN	GN	GN
GSA	GP	BD	GL	GA	GP+GL	GP+BD	GP+GA	GP+GA+GL
GSV	GP	BD	GL	GA	GP+GL	GP+BD	GP+GA	GP+GA+GL
DTM	GP	BD	GL	GA	GN	GN	GN	GN