MAX-M5Q

Receiver Description

Including Protocol Specification

GPS/GNSS

locate, communicate, accelerate

Abstract

The Receiver Description Including Protocol Specification describes the firmware features (of the MT3333 chipset), the specifications and configurations of the u-blox MAX-M5Q high performance multi-GNSS positioning modules.

The Receiver Description provides an overview and conceptual details of the supported features. The Protocol Specification includes details of the NMEA messages and PMTK commands, and serves as a reference manual.

www.u-blox.com

Document Information	
Title	MAX-M5Q Receiver Description
Subtitle	Including Protocol Specification
Document type	Manual
Document number	FTX-SW-13001
Document status	Objective Specification

Document status information	
Objective Specification	This document contains target values. Revised and supplementary data will be published later.
Advance Information	This document contains data based on early testing. Revised and supplementary data will be published later.
Preliminary	This document contains data from product verification. Revised and supplementary data may be published later.
Released	This document contains the final product specification.

This document applies to the following products:

Name	Type number	ROM/FLASH version	PCN reference
MAX-M5Q	MAX-M5Q-0-00	FLASH FW320G-UBX.3135	N/A

This document and the use of any information contained therein, is subject to the acceptance of the u-blox terms and conditions. They can be downloaded from www.u-blox.com.
u-blox makes no warranties based on the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice.
u-blox reserves all rights to this document and the information contained herein. Reproduction, use or disclosure to third parties without express permission is strictly prohibited. Copyright © 2013, u-blox AG.
u-blox® is a registered trademark of u-blox Holding AG in the EU and other countries. Easy™ and AlwaysLocate™ are registered trademarks of MediaTek Inc. ARM® is the registered trademark of ARM Limited in the EU and other countries.

Preface

How to use this Manual

This **Receiver Description including Protocol Specification** describes messages, configurations and functionalities of the u-blox MAX-M5Q positioning modules and relevant firmware features.

This manual has a modular structure. It is not necessary to read it from the beginning to the end. To help in finding needed information, a brief section overview is provided below:

1. **Receiver Description**: This section describes the software aspects of system features and brief configuration of MAX-M5Q positioning technology.
2. **NMEA Protocol**: This section describes the NMEA protocol applied in MAX-M5Q, which allows for proprietary, manufacturer-specific messages to be added provides the information necessary for a successful design.

The following symbols highlight important information within the manual:

☞ An index finger points out key information pertaining to module integration and performance.

⚠ **A warning symbol indicates actions that could negatively influence or damage the module.**

Questions

If you have any questions:

Read this manual carefully.

Contact our information service on the homepage http://www.u-blox.com

Read the questions and answers on our FAQ database on the homepage http://www.u-blox.com

Technical Support

Worldwide Web

Our website (www.u-blox.com) is a rich pool of information. Product information, technical documents and helpful FAQ can be accessed 24h a day.

By E-mail

If you have technical problems or cannot find the required information in the provided documents, contact the nearest of the u-blox Technical Support offices by email. Use our service pool email addresses rather than any personal email address of our staff. This ensures that we process your request as soon as possible. You will find the contact details at the end of the document.

Helpful Information when Contacting Technical Support

When contacting Technical Support, have the following information ready:

Receiver type (e.g. MAX-M5Q-0-00), Datacode (e.g. 172100.0100.000) and Firmware version (e.g. 320G)

Receiver configuration

Clear description of your question or the problem together with a Workbench logfile

A short description of the application

Your complete contact details

Contents

1 Receiver Description

1.1 Overview

The Receiver Description Including Protocol Specification is an important resource for integrating and configuring u-blox MAX-M5Q positioning chips and modules. This document has a modular structure and it is not necessary to read it from the beginning to the end. There are two main sections: The Receiver Description and the Protocol Specification.

The MAX-M5Q Receiver Description describes the software aspects of system features and configuration of MT3333-based positioning technology. The structure of the Receiver Description is according to areas of functionality, with links provided to the corresponding NMEA messages, which are described in the Protocol Specification.

The Protocol Specification is a reference describing the software messages used by the u-blox MAX-M5Q GNSS receiver and is organized by the specific NMEA messages.

This document provides general information on u-blox MAX-M5Q GNSS receivers. Refer to the *MAX-M5Q Data Sheet* [1] and/or the *MAX-M5Q Hardware Integration Manual* [2] for possible restrictions or limitations.

1.2 GNSS Configuration

The u-blox all-in-one hybrid MAX-M5Q GNSS receivers can autonomously acquire and track satellites from multiple Global Navigation Satellite Systems (GNSS) and utilize them in positioning. The MAX-M5Q multi-GNSS receivers can be configured to operate in following constellation options:

- Support GPS and GLONASS simultaneously, which is the default operation mode
- GPS only , SBAS (e.g. WAAS, EGNOS, MSAS) and QZSS L1 signals, centered on 1575.42MHz L1 frequency
- GLONASS only, L1 signals, centered on 1602.00MHz L1 frequency

PMTK commands are used to configure the MAX-M5Q receiver into the required mode of operation. The receiver will respond to such a request with a PMTK message if it can support the requested configuration.

For GNSS mode configuration details, see command $PMTK353 described in section 2.4.32.

1.2.1 GLONASS

The Russian GLONASS satellite system is an alternative system to the US-based Global Positioning System (GPS). The u-blox MAX-M5Q module is capable of receiving and processing GLONASS L1 OF satellite signals using the same hardware. It can receive and track GLONASS and GPS signals simultaneously.

GLONSS has a number of significant differences when compared to GPS. In most cases, u-blox MAX-M5Q receivers operate in a very similar manner when they are configured to use GLONASS signals. However, some aspects of receiver output differ:

- “GP” is the GPS talker identifier. When GLONASS constellation is used in positioning, NMEA messages will use the GLONASS talker identifier “GN” or “GL” (see NMEA Protocol Configuration in section 2.2).
- Positioning accuracy when using only GLONASS satellites may be very slightly worse than when using only GPS satellites. On high latitudes (north or south), GLONASS accuracy is better than that of GPS due to the orbital position of the satellites.
- As GLONASS uses a time base aligned directly to UTC, GLONASS receivers are affected by leap seconds, when the UTC time base is occasionally re-calibrated. As a consequence, users should be prepared for the receiver to restart itself if GLONASS signals are being tracked when a leap second occurs.

GPS receivers are unaffected by leap second changes as their time base (GPS time) is independent of leap seconds. GPS satellites periodically transmit information that allows the receiver to calculate UTC.

1.2.2 QZSS

QZSS is a GNSS operated by Japan Aerospace Exploration Agency (JAXA). It is intended as an enhancement to GPS that increases availability and positional accuracy. This can be achieved by the QZSS system transmitting GPS-compatible signals in the GPS bands.

NMEA messages will show the QZSS satellites only if configured accordingly (see command $PMTK351 described in section 2.4.30).

☞ QZSS is disabled in MAX-M5Q default mode. Configuration of QZSS function by command $PMTK352 is described in section 2.4.31.

1.3 SBAS (Satellite Based Augmentation Systems)

SBAS (Satellite Based Augmentation System) is an augmentation technology for GPS, which calculates GPS integrity and correction data with RIMS (Ranging and Integrity Monitoring Stations) on the ground and uses geostationary satellites to broadcast GPS integrity and correction data to GPS users. The correction data is transmitted on the GPS L1 frequency (1575.42 MHz), and therefore no additional receiver is required to make use of the correction and integrity data.

Currently, there are no operational augmentation systems for any GNSS other than GPS. Consequently, this section only addresses GPS.

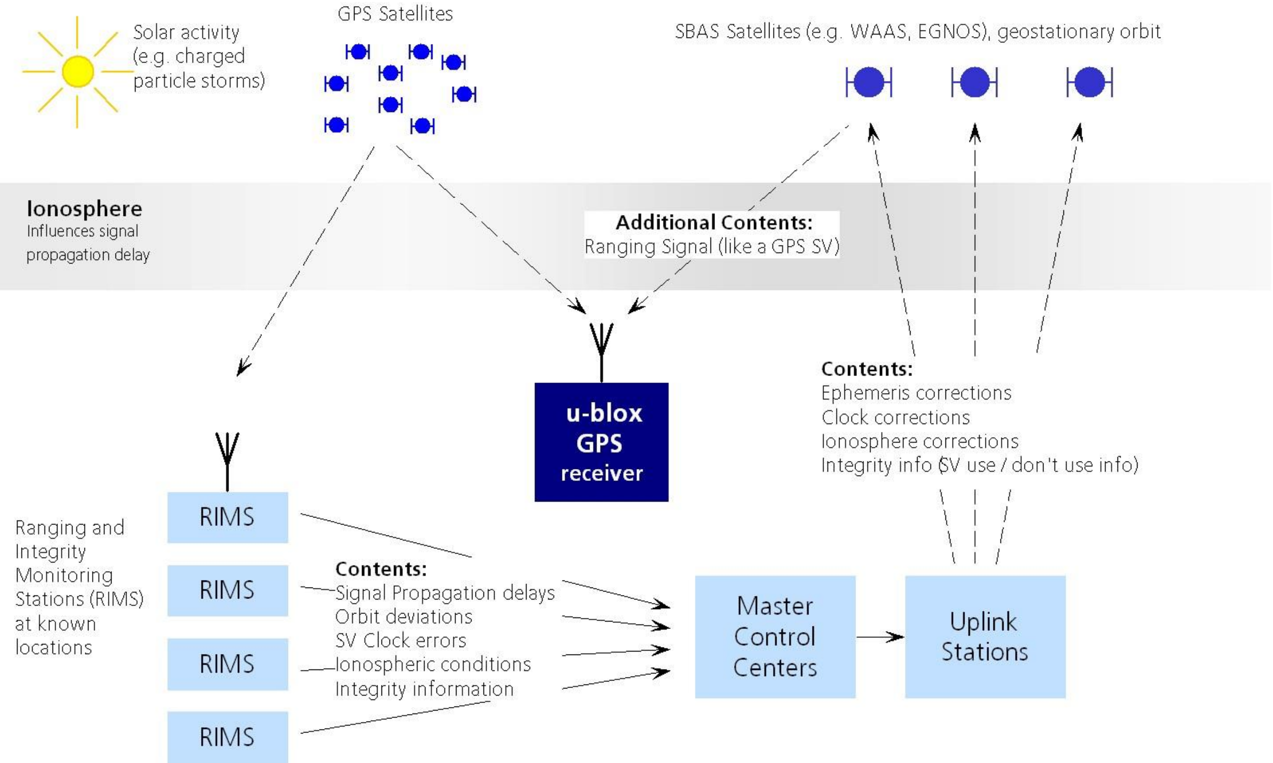

Figure 1: SBAS Principle

There are several compatible SBAS systems available or in development all around the world:

- WAAS (Wide Area Augmentation System) for North America has been in operation since 2003.
- MSAS (Multi-Functional Satellite Augmentation System) for Asia has been in operation since 2007.
- EGNOS (European Geostationary Navigation Overlay Service) has been in operation since 2009.
- GAGAN (GPS Aided Geo Augmented Navigation), developed by the Indian government is at the time of writing in test mode.

When SBAS is enabled, the user benefits from additional satellites for ranging (navigation). u-blox MAX-M5Q GPS technology uses the available SBAS Satellites for navigation just like GPS satellites, if the SBAS satellites offer this service.

For more information on SBAS and associated services, refer to

- RTCA/DO-229D (MOPS). Available from www.rtca.org
- gps.faa.gov for information on WAAS.
- www.esa.int for information on EGNOS.
- www.essp-sas.eu for information about European Satellite Services Provider (ESSP) and the EGNOS operations manager.
- www.isro.org for information on GAGAN

SBAS is disabled in MAX-M5Q in default mode. Configuration of SBAS function by command $PMTK313 is described in section 2.4.25.

1.4 Clocks and Time

1.4.1 Oscillators

The receiver is dependent on a local oscillator (normally a TCXO or Crystal oscillator) for both the operation of its radio parts and for timing within its signal processing. MAX-M5Q GNSS modules are in TCXO (26 MHz Master Clock) versions. The TCXO allows accelerated weak signal acquisition, enabling faster start and reacquisition times.

1.4.2 Real-Time Clock (RTC)

u-blox MAX-M5Q receivers contain circuitry to support a **Real-Time Clock**, which (if correctly fitted and powered) keeps time while the receiver is otherwise powered off. The RTC is driven by a 32.768 kHz oscillator, which makes use of an internal RTC crystal. If the main supply voltage fails and a battery is connected to V_BCKP, parts of the receiver switch off, but the RTC still runs and provides a timing reference for the receiver. When the receiver powers up, it attempts to use the real time clock to initialize receiver local time and in most cases this leads to appreciably faster first fixes.

1.5 Serial Communication Ports Description

1.5.1 UART

Universal Asynchronous Receiver/Transmitter (UART) ports transmit GNSS measurements, monitor status information and configure the receiver. MAX-M5Q modules include one UART interface, which can be used for communication to a host. As shown in Figure 2, the serial ports consist of an RX and a TX line. Neither handshaking signals nor hardware flow control signals are available. These serial ports operate in asynchronous mode. The RX signal is pulled up internally and can be left floating (not connected) when not used.

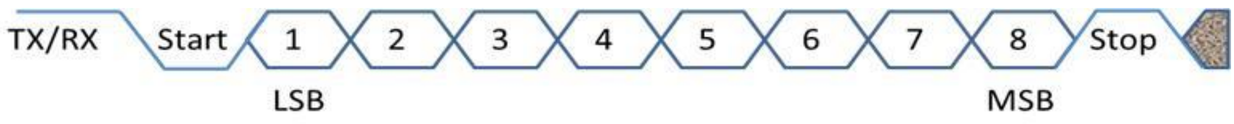

Figure 2: **UART timing**

MAX-M5Q UART supports configurable baud rates. Serial data rates are configurable from 4,800 baud to 921,600 baud by using the $PMTK251,<baud>*<checksum><CR><LF> command. Ensure that the message payload fits in the selected baud rate. The default baud rate is 115,200 baud; protocol is NMEA.

For details of baud rates configuration, see section 2.4.21 (command $PMTK251).

1.6 Receiver Configuration

1.6.1 Configuration Concept

u-blox MAX-M5Q GNSS receivers are configurable with PMTK commands (see section 2.4). The configuration used by the GNSS receiver during normal operation is termed as "Current Configuration". The Current Configuration can be changed during normal operation by sending any supported PMTK commands to the receiver over an I/O port. The receiver will change its Current Configuration immediately after receiving the configuration message. The GNSS receiver always uses only the Current Configuration. If receiver is power-off (without any backup power), it will return to Default Configuration.

☞ You can reset the settings by powering off the module and removing the backup battery supply for at least 30 s.

1.6.2 Receiver Default Configuration

The receiver's Default Configuration is normally determined when the receiver is manufactured. Table 1 lists the default configurations for MAX-M5Q GNSS receivers.

☞ Refer to *MAX-M5Q Data Sheet* [1] for further details.

Features / Function	Default Configuration
Constellation option	GPS + GLONASS hybrid
Baud rate	115,200 baud
NMEA message	GGA, RMC, GSV, GSA, GLL and VTG
PPS	Enabled, 1Hz, 100ms, fix 2D/3D
Locus logger	Enabled, Full & Stop, Basic, AL, Fix Only, 15 sec
DGPS/SBAS	Disabled
QZSS	Disabled
Interference suppression	Disabled

Table 1: Default Configuration for MAX-M5Q

1.7 Power management

u-blox MAX-M5Q technology offers a power-optimized architecture with built-in autonomous power saving functions to minimize power consumption at any given time. The receiver can be used in two operating modes: Continuous mode for best performance or Power Save mode for optimized power consumption. In addition, a high efficiency DC/DC converter is integrated to allow low power consumption even for higher main supply voltages.

u-blox MAX-M5Q modules have the following operating modes:

- Continuous mode for best GPS/GNSS performance
- Power Save mode to optimize power consumption
    - Standby mode
    - Periodic mode
    - AlwaysLocate™ mode
- Backup state

1.7.1 Continuous mode

Continuous Mode uses the acquisition engine at full performance resulting in the shortest possible TTFF and the highest sensitivity. It searches for all possible satellites until the almanac is completely downloaded. The receiver then switches to the tracking engine to lower the power consumption. Thus, a lower tracking current consumption level will be achieved when:

- A valid GPS/GNSS position is obtained
- The entire almanac has been downloaded
- The ephemeris for each satellite in view is valid

1.7.2 Power Save mode

For power sensitive applications, the MAX-M5Q module also supports low power operating modes for reduced power consumption by using the embedded power switch:

☞ For more information about power management PMTK commands, see section 2.4.10 (command $PMTK161), section 2.4.19 (command $PMTK223) and section 2.4.20 (command $PMTK225).

1.7.2.1 Standby mode

In Standby mode the MAX-M5Q receiver stops navigation, the internal processor enters standby state, and the current drain at main supply VCC is reduced to approximately 0.4 mA. Standby mode is entered by sending the PMTK command: $PMTK161,0*28. The host can wake up the module from Standby mode to Full Power mode by sending any byte via the host port.

☞ For detail configuration of Standby mode, see section 2.4.10 (command $PMTK161).

1.7.2.2 Periodic mode

This mode allows MAX-M5Q receivers to autonomously power on/off with reduced fix rate, which reduces the average power consumption, as shown in Figure 3 below. The main power supply VCC is still active, but PMTK commands turn the supply on and off internally. Periodic mode is configured by the command $PMTK225.

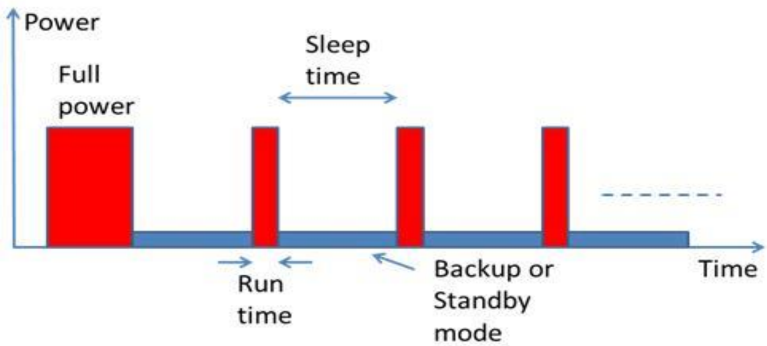

Figure 3: Periodic Mode

☞ Refer to *MAX-M5Q Data Sheet* [1] for more information.

☞ For configuration of periodic mode, see section 2.4.20 (command $PMTK225).

1.7.2.3 AlwaysLocate Mode

AlwaysLocate™ is an intelligent controller of the Periodic mode; the main power supply VCC is still active, but PMTK commands turn the supply on and off internally. Depending on the environment and motion conditions, the MAX-M5Q module can autonomously and adaptively adjust the parameters of the Periodic mode, e.g. on/off ratio and fix rate to achieve a balance in positioning accuracy and power consumption. The average power drain can vary based on conditions; typical average power is 7 mW. Associated profiles are: High and Low Speed, Walking, Outdoor Static and Indoor.

AlwaysLocate™ Mode is configurated by the PMTK command $PMTK225.

☞ Refer to *MAX-M5Q Data Sheet* [1] for more information.

☞ For detail configuration of AlwaysLocate mode, see the section 2.4.20 (command $PMTK225).

The MAX-M5Q module can control the embedded VCC power switch autonomously only after the MAX-M5Q is set to Periodic or to AlwaysLocate™ mode by a PMTK command.

Note that first fix position accuracy can be somewhat degraded in Power Management modes when compared to Full Power operation. The user can improve the position accuracy by taking the 2nd or 3rd fix after waking up.

The user can exit low power modes to Full Power by sending the command $PMTK225,0*2B just after the module wakes up from a previous sleep cycle.

1.7.3 Backup State

Backup State means a low quiescent (approximately 10 µA at V_BCKP) power state where receiver operation is stopped; only the backup supply V_BCKP is powered on while the main supply VCC is switched off by host (or autonomously by MAX-M5Q in Periodic mode and AlwaysLocate™ mode). The host controls waking up from Backup State to Full Power by switching on the VCC supply.

After waking up, the receiver uses all internal aiding, including GNSS time, Ephemeris and Last Position, resulting in the fastest possible TTFF in either hot or warm start modes.

During Backup State, the I/O block is powered off. The suggestion is that the host forces its outputs to a low state or to a high-Z state during the Backup State to minimize small leakage currents (<10 µA) at receiver's input signals.

1.8 Time pulse

The Time pulse (PPS) output signal provides a pulse-per-second output signal for timing purposes. The MAX-M5Q time pulse signal is 1 pulse per second. Pulse length (high state) is 100 ms and it has 1 µs accuracy synchronized at rising edge to full UTC second with nominal GNSS signal levels. The module will output PPS a few seconds after the first fix and after the fix epoch is synchronized to a full second.

The PPS output is valid when navigation is valid and will also continue to "freewheel" after a valid fix is lost by a certain navigation DR timeout of typically 10 seconds. Users can also enable the NMEA $GPZDA message that is sent right after the PPS pulse is sent.

☞ For more information about NMEA $GPZDA message, see section 2.3.2.7.

1.9 Jamming Remover

Jamming Remover in MAX-M5Q receiver is an embedded HW block providing interference suppression that tracks and removes up to 12 pieces CW (Carrier Wave) type signals up to -80 dBm (total power signal levels). By default the interference suppression is disabled and usage requires a command $PMTK286,1*23<CR><LF> to enable.

☞ For more information about Jamming Remover activate command, see section 2.4.22 (command $PMTK286).

Jamming Remover can be used for solving narrow band (CW) EMI problems in the customer's system. For example, it is effective against narrow band clock harmonics. When enabled, Jammer Remover will increase current drain by about 1 mA with a low impact on GNSS performance at modest jamming levels. However, at high jammer levels, -90 to -80 dBm, the RF signal sampling (ADC) starts to be saturated, after which GNSS signal levels start to reduce.

Note that Jamming Remover is not effective against wide band noise (e.g. from host CPU memory bus), which cannot be separated from thermal noise floor. Wide band Jamming signal increases effective noise floor and eventually reduces GNSS signal levels.

1.10 Aiding and Acquisition

1.10.1 Startup Strategies

- **Cold start**: In this startup mode, the receiver has no information about last position, time, velocity, frequency etc. Therefore, the receiver has to search the full time and frequency space, and all possible satellite numbers. If a satellite signal is found, it is tracked to decode ephemeris (18-36 seconds under strong signal conditions), whereas the other channels continue to search satellites. Once there are sufficient numbers of satellites with valid ephemeris, the receiver can calculate position and velocity data. Note that some competitors call this startup mode Factory Startup.

- **Warm start**: In warm start mode, the receiver has approximate information of time, position, and coarse data on Satellite positions (Almanac). In this mode, after power-up, the receiver needs to download ephemeris until it can calculate position and velocity data. As the ephemeris data usually is outdated after four hours, the receiver will typically start with a warm start if it was powered down for over four hours.
- **Hot start:** In hot start, the receiver was powered down only for a short time (four hours or less), so that its ephemeris is still valid. Since the receiver does not need to download ephemeris again, this is the fastest startup method. In the NMEA message, one can force the receiver to reset and clear data, in order to see the effects of maintaining/losing such data between restarts.

☞ For more information about NMEA commands related to different startup mode of MAX-M5Q, see sections 2.4.5 (Hot start), 2.4.6 (Warm start), 2.4.7 (Cold start) and 2.4.8 (Full cold start).

1.10.2 Aiding / Assisted GPS (A-GPS)

Users expect instant position information. With standard GPS this is not always possible because at least four satellites must transmit their precise orbital position data, called ephemeris, to the GPS receiver. Under adverse signal conditions, data downloads from the satellites to the receiver can take minutes, hours or even fail altogether. Assisted GPS (A-GPS) boosts acquisition performance by providing data such as ephemeris, almanac, accurate time and satellite status to the GPS receiver via mobile networks or the Internet. The aiding data enables the receiver to compute a position within seconds, even under poor signal conditions.

Supplying such aiding data and an optional time synchronization signal significantly reduces Time to First Fix (TTFF) and improves acquisition sensitivity.

1.10.2.1 Autonomous Assisted GPS (A-GPS)

Self-Assistance EASY™ usage

The MAX-M5Q module self-assistance uses EASY™ (Embedded Assist System) GPS satellite ephemeris extension, which is embedded in the software without requiring any resources from the host. The EASY™ data is stored on internal flash memory, allows fast TTFF (typ. 3 seconds over 3 days), and is enabled by default.

Allow the receiver to navigate for at least five minutes after initial start and after every three days with good GNSS satellite visibility in order to collect broadcast ephemeris and to process necessary information.

☞ For more information about PMTK command for EASY™ configuration, see section 2.4.17 (command $PMTK869).

1.11 Logger LOCUS usage

The MAX-M5Q module supports an embedded logger function called LOCUS, and when enabled it can log position information to internal flash memory. The default log interval is 15 seconds, which provides typically > 16 h log capacity. The LOCUS can be enabled by the command $PMTK185,0*22.

☞ For more information concerning configuration of LOCUS function, see sections:
2.4.11 (command $PMTK183), 2.4.12 (command $PMTK184), 2.4.13 (command $PMTK185),
2.4.14 (command $PMTK186), 2.4.15 (command $PMTK187) and 2.4.16 (command $PMTK622).

2 NMEA Protocol

2.1 Protocol Overview

The MAX-M5Q GNSS receiver is based on the MT3333 chipset. The NMEA protocol applied in MAX-M5Q modules is based on NMEA 0183 revision 3.01. Figure 4 shows the structure of a NMEA protocol message.

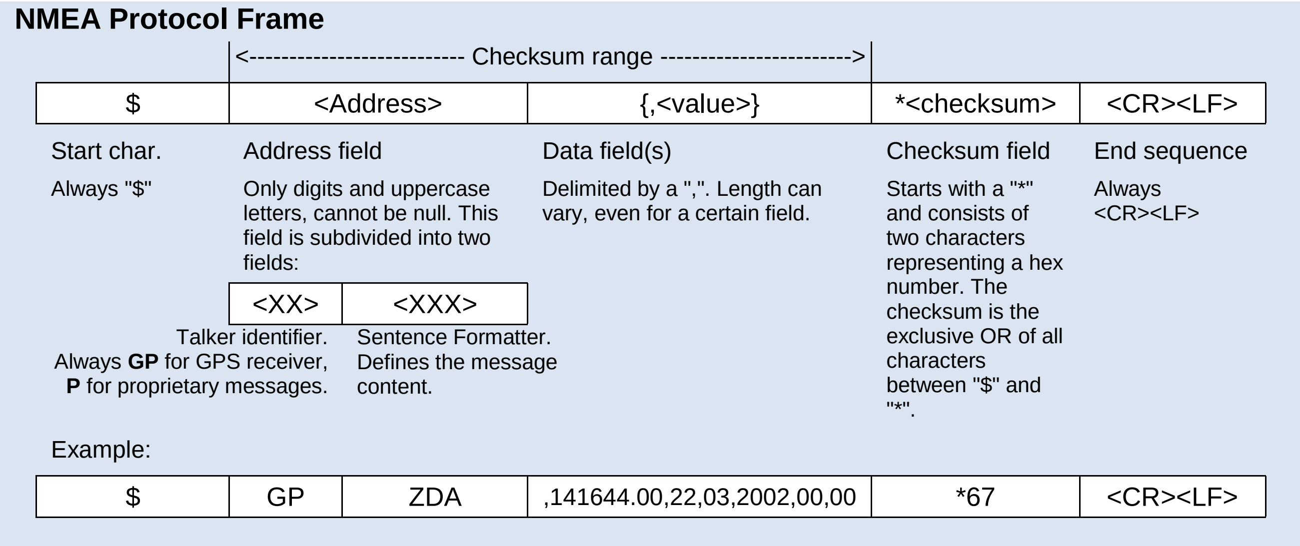

Figure 4: Overview of NMEA protocol message structure

2.2 NMEA Protocol Configuration

The NMEA Protocol on MAX-M5Q receiver can be configured according to the need of customer applications using PMTK messages.

The NMEA standard differentiates between GPS, GLONASS and combined GNSS receivers using a two-letter message identifier, known as the "Talker ID". By default, receivers configured to support GPS and GLONASS use the "GN", "GL" and "GP" talker IDs. Table 2 shows "Talker ID" used in individual operation modes.

Operation mode	Talker ID
GPS + GLONASS hybrid (defualt)	GP, GN, GL
GPS only	GP, GL
GLONASS only	GP, GN, GL

Table 2: Talker ID for different constellation modes

☞ The NMEA specification indicates that the GGA, GLL and VTG messages are GPS specific.

☞ In GPS only mode: the incomplete GLGSV message shown as "GLGSV,1,1,00*65" (SV-info missing) should be ignored.

☞ In GLONASS only mode: the incomplete GPGSV messages with empty SNR-field indicate that GPS satellites are not in track.

2.3 NMEA Messages Overview

The following table shows a list of standard NMEA messages of MAX-M5Q receivers.

Page	Message	Description
15	**GGA**	Global Positioning System Fix Data
16	**GLL**	Latitude and Longitude, with Time of Position Fix and Status
17	**GSA**	GNSS DOP and Active Satellites
18	**GSV**	GNSS Satellites in View
18	**RMC**	Recommended Minimum Data
19	**VTG**	Course over Ground and Ground Speed
20	**ZDA**	Time and Date

Table 3: Standard NMEA messages of MAX-M5Q receiver

2.3.1 NMEA default output messages

The MAX-M5Q receivers support standard NMEA0183 messages. In the default configuration, output messages are GGA, GLL, RMC, GSV, GSA and VTG messages. The receiver can be configured to have user defined set of output messages, by command PMTK314 as described in section 2.4.26.

Output messages are utilized in Workbench 5 (version 5.22, download from http://www.u-blox.com/evk-downloads.html) to visualize the GPS/GNSS data. Default output messages cover most of the applications.

2.3.2 NMEA standard messages

2.3.2.1 GGA - global positioning system fix data

Message	Description	Type	Comment
GGA	Global positioning system fix data	output message	Time and position, together with GPS fixing related data (number of satellites in use, and the resulting HDOP, etc.).

Example:

$GPGGA,114353.000,6016.3245,N,02458.3270,E,1,10,0.81,35.2,M,19.5,M,,*50

Message structure:

$GPGGA,hhmmss.ddd,xxmm.dddd,<N|S>,yyymm.dddd,<E|W>,v,s,d.d,h.h,M,g.g,M,a.a,xxxx*hh<CR><LF>

Field No	Format	Example	Description
0	$	$	Preamble, one byte character
1	xxGGA	GPGGA	GGA Message ID (xx = current Talker ID)
2	hhmmss.ddd	114353.000	UTC time of the fix. hh=hours; mm=minutes; ss=seconds; ddd=decimal part of seconds
3	xxmm.dddd	6016.3245	Latitude coordinates. xx=degrees; mm=minutes; dddd=decimal part of minutes
4	<N|S>	N	Character denoting either N=North or S=South
5	yyymm.dddd	02458.3270	Longitude coordinates. yyy=degrees; mm=minutes; dddd=decimal part of minutes
6	<E/W>	E	Character denoting either E=East or W=West

Field No	Format	Example	Description
7	v	1	Fix valid indicator 1: GPS fix (SPS) 2: DGPS fix 3: PPS fix 4: Real Time Kinematic 5: Float RTK 6: Estimated (dead reckoning) (2.3 feature) 7: Manual input mode 8: Simulation mode
8	s	10	Number of satellites used in position fix
9	d.d	0.81	HDOP - Horizontal Dilution Of Precision.
10	h.h	35.2	Altitude (mean-sea-level, geoid)
11	M	M	Letter M
12	g.g	19.5	Difference between the WGS-84 reference ellipsoid surface and the mean-sea-level altitude.
13	M	M	Letter M
14	a.a	-	-
15	xxxx	-	-
16	hh	50	Two bytes character string, hh is the checksum of the data between preamble and "*"
17	<CR><LF>	-	Two bytes binary data used to identify the end of a message.

2.3.2.2 GLL – Geographic position (Latitude/Longitude)

Message	Description	Type	Comment
GLL	Geographic position (Latitude/Longitude)	output message	Latitude and longitude, UTC time of fix and status

Example:

$GPGLL,6012.5674,N,02449.6545,E,072022.000,A,A*50

Message structure:

$GPGLL,xxmm.dddd,<N|S>,yyymm.dddd,<E|W>,hhmmss.ddd,S,M*hh<CR><LF>

Field No	**Format**	**Example**	**Description**
0	$	$	Preamble, one byte character
1	xxGLL	GPGLL	GLL Message ID (xx = current Talker ID)
2	xxmm.dddd	6012.5674	Latitude coordinates. xx=degrees; mm=minutes; dddd=decimal part of minutes
3	<N|S>	N	Character denoting either N=North or S=South
4	yyymm.dddd	02449.6545	Longitude coordinates. yyy=degrees; mm=minutes; dddd=decimal part of minutes
5	<E|W>	E	Character denoting either E=East or W=West
6	hhmmss.ddd	072022.000	UTC time of the fix. hh=hours; mm=minutes; ss=seconds; ddd=decimal part of seconds
7	S	A	Status indicator. A=valid; V=invalid
8	M	A	Mode indicator. A=autonomous; N=data not valid
9	hh	50	Two bytes character string, hh is the checksum of the data between preamble and "*"
10	<CR><LF>	-	Two bytes binary data used to identify the end of a message.

2.3.2.3 GSA – GNSS DOP and active satellites

Message	Description	Type	Comment
GSA	GNSS DOP and active satellites	output message	GNSS receiver operating mode, satellites used in the navigation solution reported by the GGA sentence and DOP values.

☞ Message starting with "$GNGSA" is output default mode including both GPS and GLONASS constellation satellite data, or in GLONASS only mode.

Example:

$GPGSA,A,3,02,21,30,04,16,05,10,12,31,29,,,1.33,0.81,1.06*02

Example MAX-M5Q parallel mode:

$GNGSA,A,3,26,21,16,22,18,06,19,15,30,03,07,08,1.03,0.55,0.87*1D

$GNGSA,A,3,78,71,80,86,65,79,88,87,72,,,,1.03,0.55,0.87*19

Message structure:

$GPGSA,a,b,xx,xx,xx,xx,xx,xx,xx,xx,xx,xx,xx,xx,p.p,h.h,v.v*hh<CR><LF>

Field No	Format	Example	Description
0	$	$	Preamble
1	xxGSA	GNGSA	GSA Message ID (xx = current Talker ID)
2	a	A	Mode: M = Manual, forced to operate in 2D or 3D mode. A = Automatic, allowed to automatically switch 2D/3D.
3	b	3	Mode: 1: Fix not available, 2: 2D 3: 3D
4	xx	26,21,16,22,18, 06,19,15,30,03, 07,08	ID (PRN) numbers of GPS satellites used in solution
5	p.p	1.03	PDOP, position dilution of precision
6	h.h	0.55	HDOP, horizontal dilution of precision
7	v.v	0.87	VDOP, vertical dilution of precision
8	hh	1D	Two bytes character string, hh is the checksum of the data between preamble and "*"
9	<CR><LF>	-	Two bytes binary data used to identify the end of a message.

2.3.2.4 GSV – GNSS satellites in view

Message	Description	Type	Comment
GSV	GNSS Satellites in view	output message	Number of satellites in view, satellite ID (PRN) numbers, elevation, azimuth and SNR value. The information for four satellites is a maximum per one message, additional messages up to maximum of eight are sent if needed. The satellites are in PRN number order.

Message starting with "$GPGSV" will be output using GPS constellation satellite data.

Message starting with "$GLGSV" will be output using GLONASS constellation satellite data.

Example:

$GPGSV,3,1,11,29,68,228,47,30,59,151,47,31,44,284,45,02,38,062,44*7C

$GLGSV,3,1,09,67,74,147,46,77,65,160,49,68,46,306,48,76,45,047,50*66

$GPGSV,3,3,11,21,05,196,29,16,05,297,28,13,02,021,30*4E

Message structure:

$GPGSV,n,m,ss,xx,ee,aaa,cn,………… ,xx,ee,aaa,cn*hh<CR><LF>

Field No	Format	Example	Description
0	$	$	Preamble
1	xxGSV	GPGSV	GSV Message ID (xx = current Talker ID)
2	n	3	Total number of messages, 1 to 9
3	m	1	Message number, 1 to 9
4	ss	09	Total number of satellites in view
5	xx	67	Satellite ID (PRN) number
6	ee	74	Satellite elevation, degrees 90 max
7	aaa	147	Satellite azimuth, degrees True, 000 to 359
8	cn	46	Signal-to-noise ration (C/No) 00-99 dB-Hz. Value of zero means that the satellite is predicted to be on the visible sky but it isn't being tracked.
9	hh	66	Two bytes character string, hh is the checksum of the data between preamble and "*"
10	<CR><LF>	-	Two bytes binary data used to identify the end of a message.

2.3.2.5 RMC - recommended minimum specific GNSS data

Message	Description	Type	Comment
RMC	Recommended minimum specific GNSS data	output message	The recommended minimum sentence defined by NMEA for GNSS system data.

Message start as "$GNRMC" will be output in default mode including both GPS and GLONASS constellation satellite data, or in GLONASS only mode.

Example:

$GPRMC,114353.000,A,6016.3245,N,02458.3270,E,0.01,0.00,121009,,,A*69

Example MAX-M5Q parallel mode:

$GNRMC,105440.000,A,6012.5669,N,02449.6536,E,0.00,0.00,061112,,,D*70

Message structure:

$GPRMC,hhmmss.dd,S,xxmm.dddd,<N|S>,yyymm.dddd,<E|W>,s.s,h.h,ddmmyy,d.d, <E|W>,M*hh<CR><LF>

Field No	Format	Example	Description
0	$	$	Preamble
1	xxRMC	GNRMC	The recommended minimum sentence defined by NMEA for GPS and GLONASS hybrid mode

Field No	Format	Example	Description
2	hhmmss.dd	105440.000	UTC time of the fix. hh=hours; mm=minutes; ss=seconds; dd=decimal part of seconds
3	S	A	Status indicator. A=valid; V=invalid
4	xxmm.dddd	6012.5669	Latitude coordinate. xx=degrees; mm=minutes; dddd=decimal part of minutes
5	<N|S>	N	Character denoting either N=North or S=South.
6	yyymm.dddd	02449.6536	Longitude coordinate. yyy=degrees; mm=minutes; dddd=decimal part of minutes
7	<E|W>	E	Character denoting either E=East or W=West.
8	s.s	0.00	Speed in knots.
9	h.h	0.00	Heading
10	ddmmyy	061112	UTC Date of the fix. dd=day of month; mm=month; yy=year
11	d.d	-	Magnetic variation in degrees, not supported
12	<E|W>	-	Letter denoting direction of magnetic variation. Either E=East or W=West. Not supported
13	M		Mode indicator A=autonomous; N=data not valid; D=DGPS
14	hh	70	Two bytes character string, hh is the checksum of the data between preamble and "*"
15	<CR><LF>	-	Two bytes binary data used to identify the end of a message.

2.3.2.6 VTG – Course over ground and ground speed

Message	Description	Type	Comment
VTG	Course over ground and ground speed	output message	Course and speed

Example:
$GPVTG,0.00,T,,M,0.00,N,0.00,K,A*3D

Message structure:
$GPVTG,h.hh,T,m.m,M,s.ss,N,s.ss,K,M*hh<CR><LF>

Field No	Format	Example	Description
0	$	$	Preamble
1	xxVTG	GPVTG	VTG Message ID (xx = current Talker ID)
2	h.hh	0.00	Heading in degrees.
3	T	T	Letter "T" denoting True heading in degrees.
4	m.m	-	Magnetic heading in degrees.
5	M	M	Letter "M" denoting Magnetic heading in degrees.
6	s.ss	0.00	Speed in knots.
7	N	N	Letter "N" denoting speed in knots.
8	s.ss	0.00	Speed, km/h.
9	K	K	Letter "K" denoting speed in km/h.
10	M	A	Mode indicator. A=autonomous; N=data not valid
11	hh	3D	Two bytes character string, hh is the checksum of the data between preamble and "*"
12	<CR><LF>	-	Two bytes binary data used to identify the end of a message.

2.3.2.7 ZDA – Time and date

Message	Description	Type	Comment
ZDA	Time and date	output message	Current UTC time and date

Example:

$GPZDA,071850.000,31,08,2011,,*55

Message structure:

$GPZDA,hhmmss.ddd,dd,mm,yyyy,xx,yy*hh<CR><LF>

Field No	Format	Example	Description
0	$	$	Preamble
1	xxZDA	GPZDA	ZDA Message ID (xx = current Talker ID)
2	hhmmss.ddd	071850.000	UTC time in hours, minutes, seconds and fractions of a second.
3	dd	31	UTC day of month
4	mm	08	UTC month
5	yyyy	2011	UTC year
6	xx	-	Local zone hours. Not implemented
7	yy	-	Local zone minutes. Not implemented
9	hh	55	Two bytes character string, hh is the checksum of the data between preamble and "*"
10	<CR><LF>	-	Two bytes binary data used to identify the end of a message.

2.4 PMTK commands

PMTK commands are used to change or query settings of the receivers. Table 4 shows the contents of PMTK commands for u-blox MAX-M5Q GNSS receiver based on MT3333 chipset.

Command length:
The maximum length of each packet is restricted to 255 bytes.

Sample Command:
$PMTK000*32<CR><LF>

Contents	Example	Description
Preamble	$	One byte character.
NMEA Talker ID	PMTK	This will identify for the NMEA parser that it will receive proprietary commands. Four bytes character string.
Packet type	000	Three bytes character string. An identifier used to tell the decoder how to decode the packet. From "000" to "999".
DataField	-	The DataField has variable length depending on the packet type. A comma symbol "," must be inserted before each data field to help the decoder process the DataField.
	*	One byte character. The star symbol is used to mark the end of DataField.
CHK1, CHK2	32	Two bytes character string. CHK1 and CHK2 are the checksum of the data between Preamble and "*".
CR, LF	-	Two bytes binary data used to identify the end of a packet.

Table 4: Contents of proprietary PMTK commands

2.4.1 PMTK000 TEST

Message	Type	Description
TEST	Input message	Test the communication between the receiver and host.

Example:
$PMTK000*32<CR><LF>

DataField structure:
None

Field No	Format	Example	Description
0	Four bytes character string	$PMTK	Talker ID.
1	Numeric	000	Packet type.
2	Two bytes character string	32	Checksum of the data between Preamble and "*".
3	<CR><LF>	-	Two bytes binary data used to identify the end of a command.

Return message:
$PMTK001,0,3*30

2.4.2 PMTK001 ACK

Message	Type	Description
ACK	Output message	Acknowledge a PMTK000 command.

Example:
$PMTK001,604,3*32<CR><LF>

DataField structure:
Cmd,Flag

Field No	Format	Example	Description
0	Four bytes character string	$PMTK	Talker ID.

Field No	Format	Example	Description
1	Numeric	001	Packet type.
2	Numeric (Cmd)	604	The command/packet type the acknowledge responds.
3	Numeric (Flag)	3	0: Invalid command / packet. 1: Unsupported command / packet type 2: Valid command / packet, but action failed 3: Valid command / packet, and action succeeded
4	Two bytes character string	32	Checksum of the data between Preamble and "*".
5	<CR><LF>	-	Two bytes binary data used to identify the end of a command.

2.4.3 PMTK010 SYS MSG

Message	Type	Description
SYS MSG	Output message	Output system message

Example:
$PMTK010,001*2E<CR><LF>

DataField structure:
Msg

Field No	Format	Example	Description
0	Four bytes character string	$PMTK	Talker ID.
1	Numeric	010	Packet type.
2	Numeric (Msg)	001	The system message. 0: UNKNOWN 1: STARTUP 2: Notification for the host aiding EPO 3: Notification for the transition to Normal mode is successfully done
3	Two bytes character string	2E	Checksum of the data between Preamble and "*".
4	<CR><LF>	-	Two bytes binary data used to identify the end of a command.

Return message:
$PMTK001,10,1*03

2.4.4 PMTK011 TXT MSG

Message	Type	Description
TXT MSG	Output message	Output system text message

Example:

$PMTK011,MTKGPS*08<CR><LF>

DataField structure:

MTKGPS

Field No	Format	Example	Description
0	Four bytes character string	$PMTK	Talker ID.
1	Numeric	011	Packet type.
2	Character	MTKGPS	-
3	Two bytes character string	08	Checksum of the data between Preamble and "*".
4	<CR><LF>	-	Two bytes binary data used to identify the end of a command.

2.4.5 PMTK101 CMD HOT START

Message	Type	Description
CMD HOT START	Input message	Hot restart: Use all available data in the NV Store.

Example:
$PMTK101*32<CR><LF>

DataField structure:
None

Field No	Format	Example	Description
0	Four bytes character string	$PMTK	Talker ID.
1	Numeric	101	Packet type.
2	Two bytes character string	32	Checksum of the data between Preamble and "*".
3	<CR><LF>	-	Two bytes binary data used to identify the end of a command.

2.4.6 PMTK102 CMD WARM START

Message	Type	Description
CMD WARM START	Input message	Warm restart: Do not use ephemeris at re-start.

Example:
$PMTK102*31<CR><LF>

DataField structure:
None

Field No	Format	Example	Description
0	Four bytes character string	$PMTK	Talker ID.
1	Numeric	102	Packet type.
2	Two bytes character string	31	Checksum of the data between Preamble and "*".
3	<CR><LF>	-	Two bytes binary data used to identify the end of a command.

2.4.7 PMTK103 CMD COLD START

Message	Type	Description
CMD COLD START	Input message	Cold restart: Do not use time, position, almanacs and ephemeris data at re-start.

Example:
$PMTK103*30<CR><LF>

DataField structure:
None

Field No	Format	Example	Description
0	Four bytes character string	$PMTK	Talker ID.
1	Numeric	103	Packet type.
2	Two bytes character string	30	Checksum of the data between Preamble and "*".
3	<CR><LF>	-	Two bytes binary data used to identify the end of a command.

2.4.8 PMTK104 CMD FULL COLD START

Message	Type	Description
CMD FULL COLD START	Input message	Full cold restart: It is essentially a cold restart, but additionally it clears system/user configurations at re-start. That is, reset the receiver to the factory status.

Example:
$PMTK104*37<CR><LF>

DataField structure:
None

Field No	Format	Example	Description
0	Four bytes character string	$PMTK	Talker ID.
1	Numeric	104	Packet type.
2	Two bytes character string	37	Checksum of the data between Preamble and "*".
3	<CR><LF>	-	Two bytes binary data used to identify the end of a command.

2.4.9 PMTK120 CLEAR FLASH AID

Message	Type	Description
CLEAR FLASH AID	Input message	Erase aiding data stored in the flash memory.

Example:
$PMTK120*31<CR><LF>

DataField structure:
None

Field No	Format	Example	Description
0	Four bytes character string	$PMTK	Talker ID.
1	Numeric	120	Packet type.
2	Two bytes character string	31	Checksum of the data between Preamble and "*".
3	<CR><LF>	-	Two bytes binary data used to identify the end of a command.

Return message:
$PMTK001,120,3*33

2.4.10 PMTK161 CMD STANDBY MODE

Message	Type	Description
CMD STANDBY MODE	Input message	Enter standby mode for power saving. In this mode the receiver stops navigation and the internal processor enters standby state. The receiver will wake up as soon as any command or text is sent to the receiver.

Example:
$PMTK161,0*28<CR><LF>

DataField structure:
Type

Field No	Format	Example	Description
0	Four bytes character string	$PMTK	Talker ID.
1	Numeric	161	Packet type.
2	Numeric (Type)	0	0: Stop mode 1: Sleep mode
3	Two bytes character string	28	Checksum of the data between Preamble and "*".
4	<CR><LF>	-	Two bytes binary data used to identify the end of a command.

Return message:
$PMTK001,161,3*36

2.4.11 PMTK183 LOCUS QUERY LOGGING STATUS

Message	Type	Description
LOCUS QUERY LOGGING STATUS	Input message	Query LOCUS logging status.

Example:
Input: $PMTK183*38<CR><LF>

Output: $PMTKLOG,456,0,11,31,2,0,0,0,3769,46*48

Output message structure:

$PMTKLOG, Serial#, Type, Mode, Content, Interval, Distance, Speed, Status, Number, Percent*CH

Field No	Format	Example	Description
0	Seven bytes character string	$PMTKLOG	Talker ID.
1	Numeric (Serial #)	456	Logging serial number: 0~65535
2	Numeric (Type)	0	Logging type 0: Overlap 1: FullStop
3	Numeric (Mode)	11	Logging mode, 0x08: Interval logger //1<<0: AlwaysLocate™ mode (logging with AlwaysLocate™) //1<<1: Fix only mode (logging when 3D-fix only) //1<<2: Normal mode (logging per positioning. e.g. 15 s) //1<<3: Interval mode (logging per pre-setting interval. e.g. 15 s) //1<<4: Distance mode logger (by distance. e.g. 50m) //1<<5: Speed mode (by speed. e.g. 10m/s)
4	Numeric (Content)	31	Logging contents of configuration
5	Numeric (Interval)	2	Logging interval setting (valid when Interval mode selected)
6	Numeric (Distance)	0	Logging distance setting (valid when Distance mode selected)
7	Numeric (Speed)	0	Logging speed setting (valid when Speed mode selected)
8	Numeric (Status)	0	Logging status 1: Stop Logging 2: Logging
9	Numeric (Number)	3769	Logging number of data record
10	Numeric (Percent)	46	Logging life used percentage (0%~100%)
11	CH	48	Two bytes binary data used to identify the end of a command.

Return message:
$PMTK001,183,3*3A

2.4.12 PMTK184 LOCUS ERASE FLASH

Message	Type	Description
LOCUS ERASE FLASH	Input message	Erase logger flash.

Example:
$PMTK184*22<CR><LF>

DataField structure:
Type

Field No	Format	Example	Description
0	Four bytes character string	$PMTK	Talker ID.
1	Numeric	184	Packet type.
2	Numeric (Type)	-	-1: erase all logger internal flash data
3	Two bytes character string	22	Checksum of the data between Preamble and "*".
4	<CR><LF>	-	Two bytes binary data used to identify the end of a command.

Return message:
$PMTK001,184,2*3C

2.4.13 PMTK185 LOCUS START/STOP LOGGER

Message	Type	Description
LOCUS START/STOP LOGGER	Input message	Stop or start logging data.

Example:
$PMTK185,1*23<CR><LF>

DataField structure:
Status

Field No	Format	Example	Description
0	Four bytes character string	$PMTK	Talker ID.
1	Numeric	185	Packet type.
2	Numeric (Status)	1	0: Stop logging 1: Start logging
3	Two bytes character string	23	Checksum of the data between Preamble and "*".
4	<CR><LF>	-	Two bytes binary data used to identify the end of a command.

Return message:
$PMTK001,185,3*3C

2.4.14 PMTK186 LOCUS LOG NOW

Message	Type	Description
LOCUS LOG NOW	Input message	Snapshot write log.

Example:
$PMTK186,1*20<CR><LF>

DataField structure:
Type

Field No	Format	Example	Description
0	Four bytes character string	$PMTK	Talker ID.
1	Numeric	186	Packet type.
2	Numeric (Type)	1	1 means snapshot log data.
3	Two bytes character string	20	Checksum of the data between Preamble and "*".
4	<CR><LF>	-	Two bytes binary data used to identify the end of a command.

Return message:
$PMTK001,186,3*3F

2.4.15 PMTK187 LOCUS CONFIG

Message	Type	Description
LOCUS CONFIG	Input message	Configure LOCUS setting by command.

Example:
$PMTK187,1,5*35<CR><LF>

DataField structure:
mode,setting

Field No	Format	Example	Description
0	Four bytes character string	$PMTK	Talker ID.
1	Numeric	187	Packet type.
2	Numeric (mode)	1	1 means interval mode.

Field No	Format	Example	Description
3	Numeric (setting)	5	New setting instead of the original configuration (e.g. change to 5 seconds interval as in the example above)
4	Two bytes character string	35	Checksum of the data between Preamble and "*".
5	<CR><LF>	-	Two bytes binary data used to identify the end of a command.

Return message:
$PMTK001,187,3*3E

2.4.16 PMTK622 LOCUS QUERY DATA

Message	Type	Description
LOCUS QUERY DATA	Input message	Dump Locus flash data.

Input example:
$PMTK622,1*29<CR><LF>

DataField structure:
Type

Field No	Format	Example	Description
0	Four bytes character string	$PMTK	Talker ID.
1	Numeric	622	Packet type.
2	Numeric (Type)	1	0: dump full flash data 1: dump partial in used flash data
3	Two bytes character string	29	Checksum of the data between Preamble and "*".
4	<CR><LF>	-	Two bytes binary data used to identify the end of a command.

Output example:
$PMTKLOX,0,43*6E

$PMTKLOX,1,0,0100010B,1F000000,0F000000,0000100B,00000000,7FFFFFFF,FFFFFFFF,FFFFFFFF,FFFFFFFF,FFFFFFFF,FFFFFFFF,FFFFFFFF,FFFFFFFF,FFFFFFFF,FFFFFF
FF,00FC8C1C,0B37464F,027FD670,42DC9EC6,4113007A,1A37464F,027FD670,42DC9EC6,4113006B*25
$PMTKLOX,1,1,2037464F,027FD670,42DD9EC6,41130050,2437464F,027FD670,42DD9EC6,41130054,28374
64F,027FD670,42DD9EC6,41130058,2B37464F,027FD670,42DD9E
C6,4113005B,2E37464F,027FD670,42DD9EC6,4113005E,3D37464F,027FD670,42DC9EC6,4113004C*59
$PMTKLOX,1,2,4C37464F,027FD670,42DC9EC6,4113003D,5B37464F,027FD670,42DC9EC6,4113002A,6A374
64F,027FD670,42DD9EC6,4113001A,7937464F,027FD670,42DD9E
C6,41130009,8837464F,027FD670,42DD9EC6,411300F8,9737464F,027FD670,42DD9EC6,411300E7*5C
$PMTKLOX,1,3,A637464F,027FD670,42DD9EC6,411300D6,B537464F,027FD670,42DD9EC6,411300C5,C4374
64F,027FD670,42DD9EC6,411300B4,D337464F,027FD670,42DD9E
C6,411300A3,E237464F,027FD670,42DD9EC6,41130092,F137464F,027FD670,42DD9EC6,41130081*59
$PMTKLOX,1,4,0038464F,027FD670,42DD9EC6,4113007F,0F38464F,027FD670,42DC9EC6,41130071,1E3846
4F,027FD670,42DC9EC6,41130060,2D38464F,027FD670,42DC9E
C6,41130053,3C38464F,027FD670,42DC9EC6,41130042,4B38464F,027FD670,42DD9EC6,41130034*58
$PMTKLOX,1,5,5A38464F,027FD670,42DD9EC6,41130025,6938464F,027FD670,42DC9EC6,41130017,78384
64F,027FD670,42DC9EC6,41130006,8738464F,027FD670,42DC9E
C6,411300F9,9638464F,027FD670,42DC9EC6,411300E8,A538464F,027FD670,42DD9EC6,411300DA*5D
$PMTKLOX,1,6,B438464F,027FD670,42DC9EC6,411300CA,FFFFFFFF,FFFFFFFF,FFFFFFFF,FFFFFFFF,FFFFFFFF,FFFFFF
FF,FFFFFFFF,FFFFFFFF,FFFFFFFF,FFFFFFFF,FFFFFF
FF,FFFFFFFF,FFFFFFFF,FFFFFFFF,FFFFFFFF,FFFFFFFF,FFFFFFFF,FFFFFFFF,FFFFFFFF,FFFFFFFF*58
$PMTKLOX,1,7,FFFFFFFF,FFFFFFFF,FFFFFFFF,FFFFFFFF,FFFFFFFF,FFFFFFFF,FFFFFFFF,FFFFFFFF,FFFFFFFF,FFFFFFFF,FFFFFFF
F,FFFFFFFF,FFFFFFFF,FFFFFFFF,FFFFFFFF,FFFFFFFF,FFFFFFFF,FFFFFFFF,FFFFFFFF,FFFFFFFF,FFFFFFFF,FFFFFFFF,FFFFFFFF,FFF
FFFFF*58
…
$PMTKLOX,2*47

Output message structure:
$PMTKLOX,Type

Field No	Format	Example	Description
0	Seven bytes character string	$PMTKLOX	Talker ID.
1	Numeric	622	Packet type.
2	Numeric (Type)	0,43	Type1: LOCUS starts. Output as PMTKLOX,0,n (n is the number PMTKLOX packets will be sent) Type2: LOCUS data. (Data sent by 8-byte HEX sting, at most 24 events). If empty, output as "FFFFFFFF". Commas separate one log item. Type3: LOCUS ends. Output as PMTKLOX,2 UTC: 4 bytes Fix: 1 byte Lat: 4 bytes Lon: 4 bytes Alt: 2 bytes Spd: 2 bytes Sat: 2 bytes Cks: 1 byte
3	Two bytes character string	6E	Checksum of the data between Preamble and "*".
4	<CR><LF>	-	Two bytes binary data used to identify the end of a command.

Return message:
$PMTK001,622,3*36

2.4.17 PMTK869 ENABLE EASY

Message	Type	Description
ENABLE EASY	Input message	Enable or disable EASY™ function. Query if EASY™ is enabled or disabled.

Example:
To query if EASY™ is enabled or disabled: $PMTK869,0*29<CR><LF>
If EASY™ is enabled, the receiver returns: $PMTK869,2,1*36<CR><LF>
If EASY™ is disabled, the receiver returns: $PMTK869,2,0*37<CR><LF>
To Enable EASY™, use: $PMTK869,1,1*35<CR><LF>
To Disable EASY™, use: $PMTK869,1,0*36<CR><LF>

DataField structure:
CmdType,[Enable]

Field No	Format	Example	Description
0	Four bytes character string	$PMTK	Talker ID.
1	Numeric	869	Packet type.
2	Numeric (CmdType)	0	Set or query 0: Query 1: Set 2: Result for Query operation
3	Numeric ([Enable])	-	Enable or Disable 0: Disable 1: Enable
4	Two bytes character string	29	Checksum of the data between Preamble and "*".
5	<CR><LF>	-	Two bytes binary data used to identify the end of a command.

2.4.18 PMTK220 SET POS FIX

Message	Type	Description
SET POS FIX	Input message	Set position fix interval.

Example:
$PMTK220,1000*1F<CR><LF>

DataField structure:
Interval

Field No	Format	Example	Description
0	Four bytes character string	$PMTK	Talker ID.
1	Numeric	220	Packet type.
2	Numeric (Interval)	1000	Position fix interval [msec]. Must be larger than 200.
3	Two bytes character string	1F	Checksum of the data between Preamble and "*".
4	<CR><LF>	-	Two bytes binary data used to identify the end of a command.

Return message:
$PMTK001,220,3*30

2.4.19 PMTK223 SET AlwaysLocate DEE CFG

Message	Type	Description
SET AlwaysLocate DEE CFG	Input message	Set AlwaysLocate™ default configuration.

Example:
$PMTK223,1,25,180000,60000*38<CR><LF>

DataField structure:
SV,SNR,Extension_threshold,Extension_gap

Field No	Format	Example	Description
0	Four bytes character string	$PMTK	Talker ID.
1	Numeric	223	Packet type.
2	Numeric (SV)	1	Default value: 1 Range: 1~4
3	Numeric (SNR)	25	Default value: 30 Range: 25~30
4	Numeric (Extension_threshold)	180000	Default value: 180000 msec Range: 40000~180000
5	Numeric (Extension_gap)	60000	Default value: 60000 msec Range: 0~3600000 Extension gap is the limitation between neighbor DEE
6	Two bytes character string	38	Checksum of the data between Preamble and "*".
7	<CR><LF>	-	Two bytes binary data used to identify the end of a command.

Return message:
$PMTK001,223,3*33

2.4.20 PMTK225 SET PERIODIC MODE

Message	Type	Description
SET PERIODIC MODE	Input message	Set Periodic power saving mode.

In RUN stage, the GNSS receiver measures and calculates positions.

In SLEEP stage, the GNSS receiver may enter two different power saving modes. One is "Periodic Standby Mode", and another is "Periodic Backup Mode". Due to hardware limitations, the maximum power down duration (SLEEP) is 2047 seconds. If the configured "SLEEP" interval is larger than 2047 seconds, the GNSS firmware will automatically extend the interval by software method. However, the GNSS system will be powered on for the interval extension and powered down again after the extension is done.

With mode (type) AlwaysLocate™, you can leave other parameters set to zero, because the wake and sleep times are controlled automatically.

DataField structure:
$PMTK225,Type,Run_time,Sleep_time,Second_run_time,Second_sleep_time*CS<CR><LF>

Field No	Format	Example	Description
0	Four bytes character string	$PMTK	Talker ID.
1	Numeric	225	Packet type.
2	Numeric (Type)	1	Set operation mode of power saving. 0: Back to normal mode 1: Periodc backup mode 2: Periodic standby mode 4: Perpetual backup mode 8: AlwaysLocate™ standby mode 9: AlwaysLocate™ backup mode
3	Numeric (Run_time)	3000	Duration [msec] to fix (or attempt to fix) before switching from running mode back to a minimum power sleep mode. With AlwaysLocate™ you cannot set run time, since sleep and wakeup are controlled automatically. 0: Disable >=1000: Enable Range: 1000~518400000
4	Numeric (Sleep_time)	12000	Interval [msec] to come out of a minimum power sleep mode and start running in order to get a new position fix. With AlwaysLocateTM you cannot set sleep time, since sleep and wakeup are controlled automatically. [Range: 1000~518400000]
5	Numeric (Second_run_time)	18000	Duration [msec] to fix (or attempt to fix) before switching from running mode back to a minimum power sleep mode. With AlwaysLocate™ you cannot set second run time, since sleep and wakeup are controlled automatically. 0: Disable >=1000: Enable Range: Second set both 0 or 1000~518400000
6	Numeric (Second_sleep_time)	72000	Interval [msec] to come out of a minimum power sleep mode and start running in order to get a new position fix. Range: Second set both 0 or 1000~518400000 The second run time should be larger than the first run time when non-zero value.
7	Two bytes character string	16	Checksum of the data between Preamble and "*".
8	<CR><LF>	-	Two bytes binary data used to identify the end of a command.

☞ AlwaysLocate™ Backup is not supported in MAX-M5Q receivers.

Retrun message:
$PMTK001,225,3*35

Example: How to enter Periodic modes

- Periodic Backup mode

$PMTK225,0*2B
$PMTK223,1,25,180000,60000*38
$PMTK225,1,3000,12000,18000,72000*16

- Periodic Standby mode

  $PMTK225,0*2B
  $PMTK223,1,25,180000,60000*38
  $PMTK225,2,3000,12000,18000,72000*15

Example: How to enter AlwaysLocate modes

- AlwaysLocate™ Standby

  $PMTK225,0*2B
  $PMTK225,8*23

2.4.21 PMTK251 SET NMEA BAUD RATE

Message	Type	Description
SET NMEA BAUD RATE	Input message	Set NMEA port baud rate.

Example:
$PMTK251,38400*27<CR><LF>

DataField structure:
Baud_rate

Field No	Format	Example	Description
0	Four bytes character string	$PMTK	Talker ID.
1	Numeric	251	Packet type.
2	Numeric (Baud_rate)	38400	Baud rate setting 0: default setting 4800 9600 14400 19200 38400 57600 115200
3	Two bytes character string	27	Checksum of the data between Preamble and "*".
4	<CR><LF>	-	Two bytes binary data used to identify the end of a command.

2.4.21.1 Setting 5 Hz navigation

For 5 Hz navigation, it is necessary to change the baud rate to 38400 in order to handle the increased message load in serial port.

Command for changing the baud rate to 38400: $PMTK251,38400*27

The next command can only be given after seeing NMEA sentences again.

Command for setting the Fix Rate to 5 Hz: $PMTK300,200,0,0,0,0*2F

2.4.21.2 Setting 10 Hz navigation

For 10 Hz navigation, it is necessary to change the baud rate to 115200 in order to handle the increased message load in serial port.

Command for changing the baud rate to 115200: $PMTK251,115200*1F

The next command can only be given after seeing NMEA sentences again.

Command for setting the Fix Rate to 10 Hz: $PMTK300,100,0,0,0,0*2C

2.4.22 PMTK286 SET AIC CMD

Message	Type	Description
SET AIC CMD	Input message	Enable or disable Active Interference Cancellation (AIC) function.

The Active Interference Cancellation (AIC) feature provides effective narrow-band interference and jamming elimination. The GNSS signals can be recovered from the jammed signals and let the user get better navigation quality.

By default, this feature is disabled.

Example:
$PMTK286,1*23<CR><LF>

DataField structure:
Enabled

Field No	Format	Example	Description
0	Four bytes character string	$PMTK	Talker ID.
1	Numeric	286	Packet type.
2	Numeric (Enabled)	1	0: Disable 1: Enable
3	Two bytes character string	23	Checksum of the data between Preamble and "*".
4	<CR><LF>	-	Two bytes binary data used to identify the end of a command.

Return message:
$PMTK001,286,3*3C

2.4.23 PMTK300 API SET FIX CTL

Message	Type	Description
API SET FIX CTL	Input message	Set fix interval.

Example:

$PMTK300,1000,0,0,0,0*1C<CR><LF> :Set fix interval to 1000 milliseconds

DataField structure:

Fixinterval,0,0,0,0

Field No	Format	Example	Description
0	Four bytes character string	$PMTK	Talker ID.
1	Numeric	300	Packet type.
2	Numeric (Fixinterval)	1000	Unit: milliseconds Range: 100~10000
3	Two bytes character string	1C	Checksum of the data between Preamble and "*".
4	<CR><LF>	-	Two bytes binary data used to identify the end of a command.

Return message:
$PMTK001,300,3*33

2.4.24 PMTK301 API SET DGPS MODE

Message	Type	Description
API SET DGPS MODE	Input message	DGPS correction data source mode.

Example:
$PMTK301,1*2D<CR><LF>

DataField structure:
Mode

Field No	Format	Example	Description
0	Four bytes character string	$PMTK	Talker ID.
1	Numeric	301	Packet type.
2	Numeric (Mode)	1	DGPS data source mode. 0: No DGPS source 1: RTCM 2: WAAS
3	Two bytes character string	2D	Checksum of the data between Preamble and "*".
4	<CR><LF>	-	Two bytes binary data used to identify the end of a command.

Return message:
$PMTK001,301,3*32

2.4.25 PMTK313 API SET SBAS ENABLED

Message	Type	Description
API SET SBAS ENABLED	Input message	Enable/disable search of SBAS satellite.

Example:
$PMTK313,1*2E<CR><LF>

DataField structure:
Enabled

Field No	Format	Example	Description
0	Four bytes character string	$PMTK	Talker ID.
1	Numeric	313	Packet type.
2	Numeric (Enabled)	1	Enable or disable 0: Disable 1: Enable
3	Two bytes character string	2E	Checksum of the data between Preamble and "*".
4	<CR><LF>	-	Two bytes binary data used to identify the end of a command.

Return message:
$PMTK001,313,3*31

☞ SBAS can be used only with 1 Hz (Default) output rate!

2.4.26 PMTK314 API SET NMEA OUTPUT

Message	Type	Description
API SET NMEA OUTPUT	Input message	Set NMEA sentence output frequencies.

Example:
$PMTK314,1,1,1,1,1,5,0,0,0,0,0,0,0,0,0,0,0,1,0*2D<CR><LF>

This command set GLL output frequency to be outputting once every 1 position fix, and RMC to be outputting once every 1 position fix, and so on.

DataField structure:
Type

☞ There are totally 19 DataFields, present individual output frequencies for the 19 supported NMEA sentences.

<u>MAX-M5Q Supported NMEA Sentences</u>

0 NMEA_SEN_GLL, // GPGLL interval - –eographic Position - –atitude longitude
1 NMEA_SEN_RMC, // GPRMC interval - –ecomended Minimum Specific GNSS Sentence
2 EA_SEN_VTG, // GPVTG interval - –ourse Over Ground and Ground Speed
3 EA_SEN_GGA, // GPGGA interval - –PS Fix Data
4 EA_SEN_GSA, // GPGSA interval - –NSS DOPS and Active Satellites

5 NMEA_SEN_GSV, // GPGSV interval - –NSS Satellites in View
17 NMEA_SEN_ZDA, // GPZDA interval – Time & Date

Field No	Format	Example	Description
0	Four bytes character string	$PMTK	Talker ID.
1	Numeric	314	Packet type.
2	Numeric (Type)	1	Supported Frequency Setting for individual 19 NMEA sentences: 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
3	Two bytes character string	2D	Checksum of the data between Preamble and "*".
4	<CR><LF>	-	Two bytes binary data used to identify the end of a command.

You can restore the system default setting via following command:
$PMTK314,-1*04<CR><LF>

Messages GRS, GST, ALM, EPH, DGP, DBG and CHN are not currently supported by MAX-M5Q.

2.4.27 PMTK330 API SET DATUM

Message	Type	Description
API SET DATUM	Input message	Set default datum.

Example:
$PMTK330,0*2E<CR><LF>

DataField structure:
Datum

Field No	Format	Example	Description
0	Four bytes character string	$PMTK	Talker ID.
1	Numeric	330	Packet type.
2	Numeric (Type)	0	0: WGS84 1: TOKYO-M 2: TOKYO-A Support 219 different datums. The total datums list can be found in **Appendix** (Table 5**)**.
3	Two bytes character string	2E	Checksum of the data between Preamble and "*".
4	<CR><LF>	-	Two bytes binary data used to identify the end of a command.

Return message:
$PMTK001,330,3*30

2.4.28 PMTK331 API SET DATUM ADVANCE

Message	Type	Description
API SET DATUM ADVANCE	Input message	Set user defined datum.

Example:
$PMTK331,6377397.155,299.1528128,-148.0,507.0,685.0*16<CR><LF>

DataField structure:
majA,ecc,dX,dY,dZ

Field No	Format	Example	Description
0	Four bytes character string	$PMTK	Talker ID.
1	Numeric	331	Packet type.

Field No	Format	Example	Description
2	Numeric (majA)	6377397.155	User defined datum semi-major axis [m] Range(0~7000000)
3	Numeric (ecc)	299.1528128	User defined datumeccentric [m], Range(0~330)
4	Numeric (dX)	-148.0	User defined datum to WGS84 X axis offset [m]
5	Numeric (dY)	507.0	User defined datum to WGS84 X axis offset [m]
6	Numeric (dZ)	685.0	User defined datum to WGS84 X axis offset [m]
7	Two bytes character string	16	Checksum of the data between Preamble and "*".
8	<CR><LF>	-	Two bytes binary data used to identify the end of a command.

2.4.29 PMTK335 API SET RTC TIME

Message	Type	Description
API SET RTC TIME	Input message	Set RTC UTC time.

Example:
$PMTK335,2007,1,1,0,0,0*02<CR><LF>

DataField structure:
Year,Month,Day,Hour,Min,Sec

Field No	Format	Example	Description
0	Four bytes character string	$PMTK	Talker ID.
1	Numeric	335	Packet type.
2	Numeric (YEAR)	2007	Year
3	Numeric (Month)	1	Month: 1~12
4	Numeric (Day)	1	Day: 1~31
5	Numeric (Hour)	0	Hour: 0~23
6	Numeric (Min)	0	Min: 0~59
7	Numeric (Sec)	0	Sec: 0~59
8	Two bytes character string	02	Checksum of the data between Preamble and "*".
9	<CR><LF>	-	Two bytes binary data used to identify the end of a command.

☞ The command doesn't update the GPS time, which is maintained by GPS receiver. After setting, the RTC UTC time finally may be updated by GPS receiver with more accurate time after 60 seconds.

2.4.30 PMTK351 API SET SUPPORT QZSS NMEA

Message	Type	Description
API SET SUPPORT QZSS NMEA	Input message	Enable or disable QZSS NMEA format.

☞ The MAX-M5Q receivers support the new NMEA format for QZSS. Default is to disable QZSS NMEA format (use NMEA 0183 V3.01).

Example:
$PMTK351,0*29: Disable QZSS NMEA format

$PMTK351,1*28: Enable QZSS NMEA format

DataField structure:
Enabled

Field No	Format	Example	Description
0	Four bytes character string	$PMTK	Talker ID.
1	Numeric	351	Packet type.
2	Numeric (Enabled)	0	0: Disable 1: Enable
3	Two bytes character string	29	Checksum of the data between Preamble and "*".
4	<CR><LF>	-	Two bytes binary data used to identify the end of a command.

2.4.31 PMTK352 API SET STOP QZSS

Message	Type	Description
API SET STOP QZSS	Input message	Enable or disable the QZSS function.

QZSS is a regional positioning service. Default setting is to disable the QZSS function.

Example:

$PMTK352,0*2B : Enable QZSS function

$PMTK352,1*2A : Disable QZSS function

DataField structure:

Enabled

Field No	Format	Example	Description
0	Four bytes character string	$PMTK	Talker ID.
1	Numeric	352	Packet type.
2	Numeric (Enabled)	0	0: Enable 1: Disable
3	Two bytes character string	2B	Checksum of the data between Preamble and "*".
4	<CR><LF>	-	Two bytes binary data used to identify the end of a command.

2.4.32 PMTK353 API SET GNSS SEARCH MODE

Message	Type	Description
API SET GNSS SEARCH MODE	Input message	Configure the constellations used in navigation.

Default setting is both GLONASS and GPS enabled.

Example:

$PMTK353,0,1*36<CR><LF>: Search GLONASS satellites only

$PMTK353,1,0*36<CR><LF>: Search GPS satellites only

$PMTK353,1,1*37: Search GPS and GLONASS satellites

DataField structure:

GPS_Enabled,GLONASS_Enabled

Field No	Format	Example	Description
0	Four bytes character string	$PMTK	Talker ID.
1	Numeric	353	Packet type.
2	Numeric (GPS_Enabled)	0	0: disable (DO NOT search GPS satellites) 1 or non-Zero: search GPS satellites
3	Numeric (GLONASS_Enabled)	1	0: disable (DO NOT search GLONASS satellites) 1 or non-ZERO: search GLONASS satellites
4	Two bytes character string	36	Checksum of the data between Preamble and "*".
5	<CR><LF>	-	Two bytes binary data used to identify the end of a command.

2.4.33 PMTK386 API SET STATIC NAV THD

Message	Type	Description
API SET STATIC NAV THD	Input message	Set the speed threshold for static navigation.

If the actual speed is below the threshold, the output position will stay the same and the output speed will be zero. If the threshold value is set to 0, this function is disabled.

Example:

$PMTK386,0.7*3A<CR><LF>

DataField structure:

speed_threshold

Field No	Format	Example	Description
0	Four bytes character string	$PMTK	Talker ID.
1	Numeric	386	Packet type.
2	Numeric (speed_threshold)	0.7	0: disable >0: speed threshold in m/s The minimum speed: 0.1 m/s; the maximum speed: 2.0 m/s
3	Two bytes character string	3A	Checksum of the data between Preamble and "*".
4	<CR><LF>	-	Two bytes binary data used to identify the end of a command.

Return message:
$PMTK001,386,3*3D

2.4.34 PMTK400 API Q FIX CTL

Message	Type	Description
API Q FIX CTL	Input message	Query Position fix interval.

Example:
$PMTK400*36<CR><LF>

DataField structure:
None

Field No	Format	Example	Description
0	Four bytes character string	$PMTK	Talker ID.
1	Numeric	400	Packet type.
2	Two bytes character string	36	Checksum of the data between Preamble and "*".
3	<CR><LF>	-	Two bytes binary data used to identify the end of a command.

Return message:
$PMTK500,1000,0,0,0.0,0.0*1A

2.4.35 PMTK401 API Q DGPS MODE

Message	Type	Description
API Q DGPS MODE	Input message	Query DGPS mode.

Example:
$PMTK401*37<CR><LF>

DataField structure:
None

Field No	Format	Example	Description
0	Four bytes character string	$PMTK	Talker ID.
1	Numeric	401	Packet type.
2	Two bytes character string	37	Checksum of the data between Preamble and "*".
3	<CR><LF>	-	Two bytes binary data used to identify the end of a command.

Return message:
$PMTK501,2*28

2.4.36 PMTK413 API Q SBAS ENABLED

Message	Type	Description
API Q SBAS ENABLED	Input message	Query SBAS mode.

Example:
$PMTK413*34<CR><LF>

DataField structure:
None

Field No	Format	Example	Description
0	Four bytes character string	$PMTK	Talker ID.
1	Numeric	413	Packet type.
2	Two bytes character string	34	Checksum of the data between Preamble and "*".
3	<CR><LF>	-	Two bytes binary data used to identify the end of a command.

Return message:
$PMTK513,1*28

2.4.37 PMTK414 API Q NMEA OUTPUT

Message	Type	Description
API Q NMEA OUTPUT	Input message	Query current NMEA sentence output frequencies.

Example:
$PMTK414*33<CR><LF>

DataField structure:
None

Field No	Format	Example	Description
0	Four bytes character string	$PMTK	Talker ID.
1	Numeric	414	Packet type.
2	Two bytes character string	33	Checksum of the data between Preamble and "*".
3	<CR><LF>	-	Two bytes binary data used to identify the end of a command.

Return message:
$PMTK514,0,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0*2F

2.4.38 PMTK430 API Q DATUM

Message	Type	Description
API Q DATUM	Input message	Query default datum.

Example:
$PMTK430*35<CR><LF>

DataField structure:
None

Field No	Format	Example	Description
0	Four bytes character string	$PMTK	Talker ID.
1	Numeric	430	Packet type.
2	Two bytes character string	35	Checksum of the data between Preamble and "*".
3	<CR><LF>	-	Two bytes binary data used to identify the end of a command.

Return message:
$PMTK530,0*28

2.4.39 PMTK431 API Q DATUM ADVANCE

Message	Type	Description
API Q DATUM ADVANCE	Input message	Query user defined datum.

Example:
$PMTK431*34<CR><LF>

DataField structure:
None

Field No	Format	Example	Description
0	Four bytes character string	$PMTK	Talker ID.
1	Numeric	431	Packet type.
2	Two bytes character string	34	Checksum of the data between Preamble and "*".
3	<CR><LF>	-	Two bytes binary data used to identify the end of a command.

Return message:
$PMTK001,431,2*37

2.4.40 PMTK490 API GET USER OPTION

Message	Type	Description
API GET USER OPTION	Input message	Return the current user setting from the flash memory.

Example:
$PMTK490*3F<CR><LF>

DataField structure:
None

Field No	Format	Example	Description
0	Four bytes character string	$PMTK	Talker ID.
1	Numeric	490	Packet type.
2	Two bytes character string	3F	Checksum of the data between Preamble and "*".
3	<CR><LF>	-	Two bytes binary data used to identify the end of a command.

Return message:
$PMTK590,8,1,115200,0,1,1,1,1,1,0,0,0,2,115200*34

2.4.41 PMTK605 Q RELEASE

Message	Type	Description
Q RELEASE	Input message	Query the firmware release information.

Example:
$PMTK605*31<CR><LF>

DataField structure:
None

Field No	Format	Example	Description
0	Four bytes character string	$PMTK	Talker ID.
1	Numeric	605	Packet type.
2	Two bytes character string	31	Checksum of the data between Preamble and "*".
3	<CR><LF>	-	Two bytes binary data used to identify the end of a command.

Return Message:

$PMTK705,AXN_3.20_3333_13050700,0C3F,u-blox3333,*60

2.4.42 PMTK607 Q EPO INFO

Message	Type	Description
Q EPO INFO	Input message	Check EPO Data valid date.

Example:

$PMTK607*33<CR><LF>

DataField structure:

none

Field No	Format	Example	Description
0	Four bytes character string	$PMTK	Talker ID.
1	Numeric	607	Packet type.
2	Two bytes character string	33	Checksum of the data between Preamble and "*".
3	<CR><LF>	-	Two bytes binary data used to identify the end of a command.

Return message:
$PMTK707,56,1565,345600,1567,324000,1565,367200,1565,367200*1E

Explanation:

Number Epoch: 56

First Epoch Week: 1565

First Epoch TOW: 345600

Final Epoch Week: 1567

Final Epoch TOW: 324000

Current Min Epoch Week: 1565

Current Min Epoch TOW: 388800

Current Max Epoch Week: 1565

Current Max Epoch TOW: 388800

2.4.43 PMTK660 Q AVAILABLE GPS SV EPH

Message	Type	Description
Q AVAILABLE GPS SV EPH	Input message	Query valid ephemeris after specified interval.

☞ Host -> MT3333: A PMTK660 command to request the EPH info, together with a time interval parameter (for example, 1800sec).

☞ MT3333 -> Host: Reply 32-bit flags of 32 GPS SV to indicate which EPHs will be available after the specified time interval.

Example:
$PMTK660,1800*17<CR><LF>, indicates which EPHs will be available after 1800 seconds.

DataField structure:
Time_interval

Field No	Format	Example	Description
0	Four bytes character string	$PMTK	Talker ID.
1	Numeric	660	Packet type.
2	Numeric (Time_interval)	1800	Set the time interval for MT3333 to reply 32-bit flags of 32 GPS SV. Range: > 0 and <= 7200 (2 hours).
3	Two bytes character string	17	Checksum of the data between Preamble and "*".
4	<CR><LF>	-	Two bytes binary data used to identify the end of a command.

Return message:
$PMTK001,660,3,d049bedb*44<CR><LF>

The Hex d049bedb means 11010000010010011011111011011011, and the valid GPS SV numbers are 1, 2, 4, 10, 13, 16, 17, 19, 20, 21, 22, 23, 25, 26, 28, 29, 31, 32.

2.4.44 PMTK661 Q AVAILABLE GPS SV ALM

Message	Type	Description
Q AVAILABLE GPS SV ALM	Input message	Query valid almanac after specified interval.

Host -> MT3333: A PMTK661 command to request the almanac info, together with a time interval parameter (for example, 30 days).

Field No	Format	Example	Description
0	Four bytes character string	$PMTK	Talker ID.
1	Numeric	661	Packet type.
2	Numeric (Time_interval)	30	Set the time interval for MT3333 to reply 32-bit flags of 32 GPS SV. Range: > 0
2	Two bytes character string	1C	Checksum of the data between Preamble and "*".
3	<CR><LF>	-	Two bytes binary data used to identify the end of a command.

MT3333 -> Host: Reply 32-bit flags of 32SV to indicate which almanac will be available after the specified time interval.

Example:
$PMTK661,30*1C<CR><LF>, indicates which almanac will be available after 30 days.

DataField structure:
Time_interval

Return message:
$PMTK001,661,3,fbffffff*19<CR><LF>

The Hex fbffffff means 1111101111111111111111111111111 and the valid GPS SV numbers are 1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32.

2.4.45 PMTK740 DT UTC

Message	Type	Description
DT UTC	Input message	Contain current UTC time.

Do not use local time, which has time-zone offset. To have faster TTFF, the accuracy of reference UTC should be less than 3 seconds.

Example:
$PMTK740,2012,9,28,10,29,00*09<CR><LF>: Indicate the current UTC time 2012/Sep/28 10:29:00.

DataField structure:
YYY,MM,DD,hh,mm,ss

Field No	Format	Example	Description
0	Four bytes character string	$PMTK	Talker ID.
1	Numeric	740	Packet type.
2	Numeric (YYY)	2012	Year in 4 digits, range>1980
3	Numeric (MM)	9	Month, range 1~12
4	Numeric (DD)	28	Day, range 1~31
5	Numeric (hh)	10	Hour, range 0~23
6	Numeric (mm)	29	Minute, range 0~59
7	Numeric (ss)	00	Second, range 0~59
8	Two bytes character string	09	Checksum of the data between Preamble and "*".
9	<CR><LF>	-	Two bytes binary data used to identify the end of a command.

2.4.46 PMTK741 DT POS

Message	Type	Description
DT POS	Input message	Contain reference location for the receiver.

☞ To have faster TTFF, the accuracy of the location should be better than 30 km.

Example:

$PMTK741,24.772816,121.022636,160,2012,9,28,10,29,00*29<CR><LF>

DataField structure:

Lat,Long,Alt,YYYY,MM,DD,hh,mm,ss

Field No	Format	Example	Description
0	Four bytes character string	$PMTK	Talker ID.
1	Numeric	741	Packet type.
2	Numeric (Lat)	24.772816	WGS84 geodetic latitude. Suggest to express this value in floating-point with 6 decimal points, Minus: south; Plus: north, Range -90.0~+90.0
3	Numeric (Long)	121.022636	WGS84 geodetic longitude. Suggest to express this value in floating-point with 6 decimal points, Minus: west; Plus: east, Range -180.0~+180.0
4	Numeric (Alt)	160	WGS84 ellipsoidal altitude in meters.
5	Numeric (YYYY)	2012	Year in 4 digits, range>1980
6	Numeric (MM)	9	Month, range 1~12
7	Numeric (DD)	28	Day, range 1~31
8	Numeric (hh)	10	Hour, range 0~23
9	Numeric (mm)	29	Minute, range 0~59
10	Numeric (ss)	00	Second, range 0~59
11	Two bytes character string	29	Checksum of the data between Preamble and "*".
12	<CR><LF>	-	Two bytes binary data used to identify the end of a command.

Appendix

Map Datum

No	Datum	Region
0	WGS1984	International
1	Tokyo	Japan
2	Tokyo	Mean For Japan, South Korea, Okinawa
3	User Setting	User Setting
4	Adindan	Burkina Faso
5	Adindan	Cameroon
6	Adindan	Ethiopia
7	Adindan	Mali
8	Adindan	Mean for Ethiopia, Sudan
9	Adindan	Senegal
10	Adindan	Sudan
11	Afgooye	Somalia
12	Ain El Abd1970	Bahrain
13	Ain El Abd1970	Saudi Arabia
14	American Samoa1962	American Samoa Islands
15	Anna 1 Astro1965	Cocos Island
16	Antigua Island Astro1943	Antigua(Leeward Islands)
17	Arc1950	Botswana
18	Arc1950	Burundi
19	Arc1950	Lesotho
20	Arc1950	Malawi
21	Arc1950	Mean for Botswana, Lesotho, Malawi, Swaziland, Zaire, Zambia, Zimbabwe
22	Arc1950	Swaziland
23	Arc1950	Zaire
24	Arc1950	Zambia
25	Arc1950	Zimbabwe
26	Arc1960	Mean for Kenya Tanzania
27	Arc1960	Kenya
28	Arc1960	Tamzamia
29	Ascension Island1958	Ascension Island
30	Astro Beacon E 1945	Iwo Jima
31	Astro Dos 71/4	St Helena Island
32	Astro Tern Island (FRIG) 1961	Tern Island
33	Astronomical Station 1952	Marcus Island
34	Australian Geodetic 1966	Australia, Tasmania
35	Australian Geodetic 1984	Australia, Tasmania
36	Ayabelle Lighthouse	Djibouti
37	Bellevue (IGN)	Efate and Erromango Islands
38	Bermuda 1957	Bermuda
39	Bissau	Guuinea-Bissau
40	Bogota Observatory	Colombia
41	Bukit Rimpah	Indonesia(Bangka and Belitung Ids)
42	Camp Area Astro	Antarctica(McMurdi Camp Area)
43	Campo Inchauspe	Argentina
44	Canton Astro1966	Phoenix Island
45	Cape	South Africa
46	Cape Canaveral	Bahamas, Florida
47	Carthage	Tunisia

48	Chatham Island Astro1971	New Zealand(Chatham Island)
49	Chua Astro	Paraguay
50	Corrego Alegre	Brazil
51	Dabola	Guinea
52	Deception Island	Deception Island, Antarctia
53	Djakarta (Batavia)	Indonesia(Sumatra)
54	Dos 1968	New Georgia Islands (Gizo Island)
55	Easter Island 1967	Easter Island
56	Estonia Coordinate System1937	Estonia
57	European 1950	Cyprus
58	European 1950	Egypt
59	European 1950	England, Channel Islands, Scotland, Shetland Islands
60	European 1950	England, Ireland, Scotland, Shetland Islands
61	European 1950	Finland, Norway
62	European 1950	Greece
63	European 1950	Iran
64	European 1950	Italy (Sardinia)
65	European 1950	Italy (SIcily)
66	European 1950	Malta
67	European 1950	Mean for Austria, Belgium,Denmark, Finland, France, W Germany, Gibraltar, Greece, Italy, Luxembourg, Netherlands, Norway, Portuga,l Spain, Sweden, Switzerland
68	European 1950	Mean for Austria, Debnmark,France, W Germany, Netherland, Switzerland
69	European 1950	Mean for Irag, Israel, Jordan, Lebanon, Kuwait, Saudi Arabia, Syria
70	European 1950	Portugal, Spain
71	European 1950	Tunisia,
72	European 1979	Mean for Austria, Finland ,Netherlands ,Norway, Spain, Sweden, Switzerland
73	Fort Thomas 1955	Nevis St Kitts (Leeward Islands)
74	Gan 1970	Republic Of Maldives
75	Geodetic Dataum 1970	New Zealand
76	Graciosa Base SW1948	Azores (Faial, Graciosa, Pico, Sao, Jorge, Terceria)
77	Guam1963	Guam
78	Gunung Segara	Indonesia (Kalimantan)
79	Gux l Astro	Guadalcanal Island
80	Herat North	Afghanistan
81	Hermannskogel Datum	Croatia-Serbia, Bosnia-Herzegoivna
82	Hjorsey 1955	Iceland
83	Hongkong 1963	Hongkong
84	Hu Tzu Shan	Taiwan
85	Indian	Bangladesh
86	Indian	India,Nepal
87	Indian	Pakistan
88	Indian 1954	Thailand
89	Indian 1960	Vietnam (Con Son Island)
90	Indian 1960	Vietnam (Near 16 deg N)
91	Indian 1975	Thailand
92	Indonesian 1974	Indonesian
93	Ireland 1965	Ireland
94	ISTS 061 Astro 1968	South Georgia Islands
95	ISTS 073 Astro 1969	Diego Garcia
96	Johnston Island 1961	Johnston Island
97	Kandawala	Sri Lanka
98	Kerguelen Island 1949	Kerguelen Island
99	Kertau 1948	West Malaysia and Singapore
100	Kusaie Astro 1951	Caroline Islands

101	Korean Geodetic System	South Korea
102	LC5 Astro 1961	Cayman Brac Island
103	Leigon	Ghana
104	Liberia 1964	Liberia
105	Luzon	Philippines (Excluding Mindanao)
106	Luzon	Philippines (Mindanao)
107	M'Poraloko	Gabon
108	Mahe 1971	Mahe Island
109	Massawa	Ethiopia (Eritrea)
110	Merchich	Morocco
111	Midway Astro 1961	Midway Islands
112	Minna	Cameroon
113	Minna	Nigeria
114	Montserrat Island Astro 1958	Montserrat (Leeward Island)
115	Nahrwan	Oman (Masirah Island)
116	Nahrwan	Saudi Arabia
117	Nahrwan	United Arab Emirates
118	Naparima BWI	Trinidad and Tobago
119	North American 1927	Alaska (Excluding Aleutian Ids)
120	North American 1927	Alaska (Aleutian Ids East of 180 degW)
121	North American 1927	Alaska (Aleutian Ids West of 180 degW)
122	North American 1927	Bahamas (Except San Salvador Islands)
123	North American 1927	Bahamas (San Salvador Islands)
124	North American 1927	Canada (Alberta, British Columbia)
125	North American 1927	Canada (Manitoba, Ontario)
126	North American 1927	Canada (New Brunswick, Newfoundland, Nova Scotia, Qubec)
127	North American 1927	Canada (Northwest Territories, Saskatchewan)
128	North American 1927	Canada (Yukon)
129	North American 1927	Canal Zone
130	North American 1927	Cuba
131	North American 1927	Greenland (Hayes Peninsula)
132	North American 1927	Mean for Antigua, Barbados, Barbuda, Caicos Islands, Cuba, Dominican, Grand Cayman, Jamaica, Turks Islands
133	North American 1927	Mean for Belize, Costa Rica, El Salvador, Guatemala, Honduras, Nicaragua
134	North American 1927	Mean for Canada
135	North American 1927	Mean for Conus
136	North American 1927	Mean for Conus (East of Mississippi, River Including Louisiana,Missouri, Minnesota)
137	North American 1927	Mean for Conus (West of Mississippi, Rive Excluding Louisiana, Minnesota, Missouri)
138	North American 1927	Mexico
139	North American 1983	Alaska (Excluding Aleutian Ids)
140	North American 1983	Aleutian Ids
141	North American 1983	Canada
142	North American 1983	Conus
143	North American 1983	Hahawii
144	North American 1983	Mexico, Central America
145	North Sahara 1959	Algeria
146	Observatorio Meteorologico 1939	Azores (Corvo and Flores Islands)
147	Old Egyptian 1907	Egypt
148	Old Hawaiian	Hawaii
149	Old Hawaiian	Kauai
150	Old Hawaiian	Maui
151	Old Hawaiian	Mean for Hawaii, Kauai, Maui, Oahu
152	Old Hawaiian	Oahu
153	Oman	Oman

154	Ordnance Survey Great Britian 1936	England
155	Ordnance Survey Great Britian 1936	England, Isle of Man, Wales
156	Ordnance Survey Great Britian 1936	Mean for England ,Isle of Man, Scotland, Shetland Island, Wales
157	Ordnance Survey Great Britian 1936	Scotland, Shetland Islands
158	Ordnance Survey Great Britian 1936	Wales
159	Pico de las Nieves	Canary Islands
160	Pitcairn Astro 1967	Pitcairn Island
161	Point 58	Mean for Burkina Faso and Niger
162	Pointe Noire 1948	Congo
163	Porto Santo 1936	Porto Santo, Maderia Islands
164	Provisional South American 1956	Bolovia
165	Provisional South American 1956	Chile (Northern Near 19 deg S)
166	Provisional South American 1956	Chile (Southern Near 43 deg S)
167	Provisional South American 1956	Colombia
168	Provisional South American 1956	Ecuador
169	Provisional South American 1956	Guyana
170	Provisional South American 1956	Mean for Bolivia Chile,Colombia, Ecuador, Guyana, Peru, Venezuela
171	Provisional South American 1956	Peru
172	Provisional South American 1956	Venezuela
173	Provisional South Chilean 1963	Chile (Near 53 deg S) (Hito XVIII)
174	Puerto Rico	Puerto Rico, Virgin Islands
175	Pulkovo 1942	Russia
176	Qatar National	Qatar
177	Qornoq	Greenland (South)
178	Reunion	Mascarene Island
179	Rome 1940	Italy (Sardinia)
180	S-42 (Pulkovo 1942)	Hungary
181	S-42 (Pulkovo 1942)	Poland
182	S-42 (Pulkovo 1942)	Czechoslavakia
183	S-42 (Pulkovo 1942)	Lativa
184	S-42 (Pulkovo 1942)	Kazakhstan
185	S-42 (Pulkovo 1942)	Albania
186	S-42 (Pulkovo 1942)	Romania
187	S-JTSK	Czechoslavakia (Prior 1 Jan1993)
188	Santo (Dos) 1965	Espirito Santo Island
189	Sao Braz	Azores (Sao Miguel, Santa Maria Ids)
190	Sapper Hill 1943	East Falkland Island
191	Schwarzeck	Namibia
192	Selvagem Grande 1938	Salvage Islands
193	Sierra Leone 1960	Sierra Leone
194	South American 1969	Argentina
195	South American 1969	Bolivia
196	South American 1969	Brazial
197	South American 1969	Chile
198	South American 1969	Colombia
199	South American 1969	Ecuador
200	South American 1969	Ecuador (Baltra, Galapagos)
201	South American 1969	Guyana
202	South American 1969	Mean for Argentina, Bolivia, Brazil,Chile, Colombia, Ecuador, Guyana, Paraguay, Peru, Trinidad and Tobago, Venezuela
203	South American 1969	Paraguay
204	South American 1969	Peru
205	South American 1969	Trinidad and Tobago
206	South American 1969	Venezuela
207	South Asia	Singapore
208	Tananarive Observatory 1925	Madagascar

209	Timbalai 1948	Brunei, E Malaysia (Sabah Sarawak)
210	Tokyo	Japan
211	Tokyo	Mean for Japan, South Korea, Okinawa
212	Tokyo	Okinawa
213	Tokyo	South Korea
214	Tristan Astro 1968	Tristam Da Cunha
215	Viti Levu 1916	Fiji (Viti Levu Island)
216	Voirol 1960	Algeria
217	Wake Island Astro 1952	Wake Atoll
218	Wake-Eniwetok 1960	Marshall Islands
219	WGS 1972	Global Definition
220	WGS 1984	Global Definition
221	Yacare	Uruguay
222	Zanderij	Suriname

Table 5: Map datums

Abbreviations

Abbreviation	Definition
AIC	Active Interference Cancellation
EASY	Embedded Assist System
EGNOS	European Geostationary Navigation Overlay Service
GAGAN	GPS Aided Geo Augmented Navigation
GLONASS	Russian satellite system
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
MSAS	Multi-Functional Satellite Augmentation System
RTC	Real-Time Clock
SBAS	Satellite Based Augmentation System
QZSS	Quasi-Zenith Satellite System
UART	Universal Asynchronous Receiver/Transmitter
WASS	Wide Area Augmentation System

Table 6: Explanation of abbreviations used

Related documents

[1] MAX-M5Q Data Sheet, Docu. No FTX-HW-13003
[2] MAX-M5Q Hardware Integration Manual, Docu. No FTX-HW-13008

All these documents are available on our homepage (http://www.u-blox.com).

☞ For regular updates to u-blox documentation and to receive product change notifications, register on our homepage (http://www.u-blox.com)

Revision history

Revision	Date	Name	Status / Comments
-	31-May-2013	julu	Objective Specification

Contact

For complete contact information, visit us at www.u-blox.com

u-blox Offices

North, Central and South America

u-blox America, Inc.
Phone: +1 703 483 3180
E-mail: info_us@u-blox.com

Regional Office West Coast:
Phone: +1 408 573 3640
E-mail: info_us@u-blox.com

Technical Support:
Phone: +1 703 483 3185
E-mail: support_us@u-blox.com

Headquarters
Europe, Middle East, Africa

u-blox AG
Phone: +41 44 722 74 44
E-mail: info@u-blox.com
Support: support@u-blox.com

Asia, Australia, Pacific

u-blox Singapore Pte. Ltd.
Phone: +65 6734 3811
E-mail: info_ap@u-blox.com
Support: support_ap@u-blox.com

Regional Office China (Beijing):
Phone: +86 10 68 133 545
E-mail: info_cn@u-blox.com
Support: support_cn@u-blox.com

Regional Office China (Shenzhen):
Phone: +86 755 8627 1083
E-mail: info_cn@u-blox.com
Support: support_cn@u-blox.com

Regional Office India:
Phone: +91 959 1302 450
E-mail: info_in@u-blox.com
Support: support_in@u-blox.com

Regional Office Japan:
Phone: +81 3 5775 3850
E-mail: info_jp@u-blox.com
Support: support_jp@u-blox.com

Regional Office Korea:
Phone: +82 2 542 0861
E-mail: info_kr@u-blox.com
Support: support_kr@u-blox.com

Regional Office Taiwan:
Phone: +886 2 2657 1090
E-mail: info_tw@u-blox.com
Support: support_tw@u-blox.com