# Telit Jupiter

# MT33xx Host EPO Application Note

**80433NT11385A r.1 – 2015-07-20**

Making machines talk.

# APPLICABILITY TABLE

| SL871 |
|---|
| |
| SE868-A |
| SL869-V2 |
| SC872-A |
| SL871-S |
| SE868-AS |
| SL869-V2S |

### Notice

While reasonable efforts have been made to assure the accuracy of this document, Telit assumes no liability resulting from any inaccuracies or omissions in this document, or from use of the information obtained herein. The information in this document has been carefully checked and is believed to be entirely reliable. However, no responsibility is assumed for inaccuracies or omissions. Telit reserves the right to make changes to any products described herein and reserves the right to revise this document and to make changes from time to time in content hereof with no obligation to notify any person of revisions or changes. Telit does not assume any liability arising out of the application or use of any product, software, or circuit described herein; neither does it convey license under its patent rights or the rights of others.

It is possible that this publication may contain references to, or information about Telit products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that Telit intends to announce such Telit products, programming, or services in your country.

### Copyrights

This instruction manual and the Telit products described in this instruction manual may be, include or describe copyrighted Telit material, such as computer programs stored in semiconductor memories or other media. Laws in the Italy and other countries preserve for Telit and its licensors certain exclusive rights for copyrighted material, including the exclusive right to copy, reproduce in any form, distribute and make derivative works of the copyrighted material. Accordingly, any copyrighted material of Telit and its licensors contained herein or in the Telit products described in this instruction manual may not be copied, reproduced, distributed, merged or modified in any manner without the express written permission of Telit. Furthermore, the purchase of Telit products shall not be deemed to grant either directly or by implication, estoppels, or otherwise, any license under the copyrights, patents or patent applications of Telit, as arises by operation of law in the sale of a product.

### Computer Software Copyrights

The Telit and 3rd Party supplied Software (SW) products described in this instruction manual may include copyrighted Telit and other 3rd Party supplied computer programs stored in semiconductor memories or other media. Laws in the Italy and other countries preserve for Telit and other 3rd Party supplied SW certain exclusive rights for copyrighted computer programs, including the exclusive right to copy or reproduce in any form the copyrighted computer program. Accordingly, any copyrighted Telit or other 3rd Party supplied SW computer programs contained in the Telit products described in this instruction manual may not be copied (reverse engineered) or reproduced in any manner without the express written permission of Telit or the 3rd Party SW supplier. Furthermore, the purchase of Telit products shall not be deemed to grant either directly or by implication, estoppels, or otherwise, any license under the copyrights, patents or patent applications of Telit or other 3rd Party supplied SW, except for the normal non-exclusive, royalty free license to use that arises by operation of law in the sale of a product.

# Usage and Disclosure Restrictions

## License Agreements

The software described in this document is the property of Telit and its licensors. It is furnished by express license agreement only and may be used only in accordance with the terms of such an agreement.

### Copyrighted Materials

Software and documentation are copyrighted materials. Making unauthorized copies is prohibited by law. No part of the software or documentation may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language or computer language, in any form or by any means, without prior written permission of Telit

### High Risk Materials

Components, units, or third-party products used in the product described herein are NOT fault-tolerant and are NOT designed, manufactured, or intended for use as on-line control equipment in the following hazardous environments requiring fail-safe controls: the operation of Nuclear Facilities, Aircraft Navigation or Aircraft Communication Systems, Air Traffic Control, Life Support, or Weapons Systems (High Risk Activities"). Telit and its supplier(s) specifically disclaim any expressed or implied warranty of fitness for such High Risk Activities.

### Trademarks

TELIT and the Stylized T Logo are registered in Trademark Office. All other product or service names are the property of their respective owners.

### Third Party Rights

The software may include Third Party Right software. In this case you agree to comply with all terms and conditions imposed on you in respect of such separate software. In addition to Third Party Terms, the disclaimer of warranty and limitation of liability provisions in this License shall apply to the Third Party Right software.

TELIT HEREBY DISCLAIMS ANY AND ALL WARRANTIES EXPRESS OR IMPLIED FROM ANY THIRD PARTIES REGARDING ANY SEPARATE FILES, ANY THIRD PARTY MATERIALS INCLUDED IN THE SOFTWARE, ANY THIRD PARTY MATERIALS FROM WHICH THE SOFTWARE IS DERIVED (COLLECTIVELY “OTHER CODE”), AND THE USE OF ANY OR ALL THE OTHER CODE IN CONNECTION WITH THE SOFTWARE, INCLUDING (WITHOUT LIMITATION) ANY WARRANTIES OF SATISFACTORY QUALITY OR FITNESS FOR A PARTICULAR PURPOSE.

NO THIRD PARTY LICENSORS OF OTHER CODE SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND WHETHER MADE UNDER CONTRACT, TORT OR OTHER LEGAL THEORY, ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE OTHER CODE OR THE EXERCISE OF ANY RIGHTS GRANTED UNDER EITHER OR BOTH THIS LICENSE AND THE LEGAL TERMS APPLICABLE TO ANY SEPARATE FILES, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

## Contents

## Table of Figures

## List of Tables

# 1. Introduction

## 1.1. Scope

This document provides customers with the information about the Host-EPO feature that is furnished by Mediatek® with particular focus on the application support of Telit's MT33xx GNSS modules:

- SL871
- SE868-A
- SL869-V2
- SC872-A
- SL871-S
- SL869-V2S
- SE868-AS

Host-EPO feature is implemented and supported by the Mediatek® in their Extended Prediction Orbit (EPO) data and technology.

## 1.2. Contact Information, Support

For general contact, technical support, to report documentation errors and to order manuals, contact Telit Technical Support Center (TTSC) at:

TS-EMEA@telit.com

TS-AMERICAS@telit.com

TS-APAC@telit.com

Alternatively, use:

http://www.telit.com/en/products/technical-support-center/contact.php

For detailed information about where you can buy the Telit modules or for recommendations on accessories and components visit:

http://www.telit.com

To register for product news and announcements or for product questions contact Telit Technical Support Center (TTSC).

Our aim is to make this guide as helpful as possible. Keep us informed of your comments and suggestions for improvements.

Telit appreciates feedback from the users of our information.

## 1.3. Text Conventions

STOP ***Danger – This information MUST be followed or catastrophic equipment failure or bodily injury may occur.***

⚠ ***Caution or Warning – Alerts the user to important points about integrating the module, if these points are not followed, the module and end user equipment may fail or malfunction.***

ℹ **Tip or Information – Provides advice and suggestions that may be useful when integrating the module.**

All dates are in ISO 8601 format, i.e. YYYY-MM-DD.

## 1.4. Related Documents

- Telit Jupiter SGEE-EPO Application Note
- SL871, SE868-A, SL869-V2, and SC872-A (Flash based) Families Software User Guide
- SL871-S, SL869-V2S, and SL868-AS (ROM based) Families Software User Guide

# 2. Background Information

In order for a GNSS receiver to provide a navigation solution, it must have satellite orbit data so that it can accurately determine the expected location of each visible GPS satellite. If the receiver does not have this data, or if the data it has is too old, it must collect the data from the satellite after its signal has been acquired. Even under the best circumstances, where initial position and time are known, this process can take up to 35 seconds, which in turn can mean a Time To First Fix (TTFF) of 35 seconds or more. In typical circumstances the TTFF can be even longer, as the receiver must collect ephemeris for three or four satellites in order to navigate.

The Host-EPO feature can be utilized when a fast TTFF is chosen in an application scenario and a host-client architecture is implemented in a user platform.

With the host device where Host-EPO feature is implemented, the host can provides the system-time, reference location, and synthetic Extended Ephemeris (EE) data to the GNSS receiver at client. The presence of EE data allows the receiver to substitute it for broadcast ephemeris downloaded from the GNSS satellite and to skip the navigation data collection requirement.

## 2.1. Infrastructure Overview

In a host-client architecture, a HOST, such as a mobile device with a networking capability, is responsible to establish an HTTP connection to and access a remote server-generated extended ephemeris (SGEE) server, retrieve the data, and store it in the local storage at the host. These tasks are executed on a scheduled basis automatically. A CLIENT functionally consists of GNSS module, whose primary function is to acquire and track satellite signals, and produce position fixes.

The host decides when to provide assistance data, including transfer of the EE data to the client over the communication link between the host and the client (i.e. a serial port). Typical scenarios for such Host-EPO include the time instants when a power-on or restart event occurred at the client.

A typical host-client infrastructure for a GNSS receiver being a client device can be illustrated with the following diagram where an EE centred, aided-GPS enabled architecture is formed. The data flow can be summarized as the following: the application running at the host provides the functionality to access a remote EE server, retrieve the server-generated extended ephemeris (SGEE) data, and transfer the data to the library running at the client as assistance data.

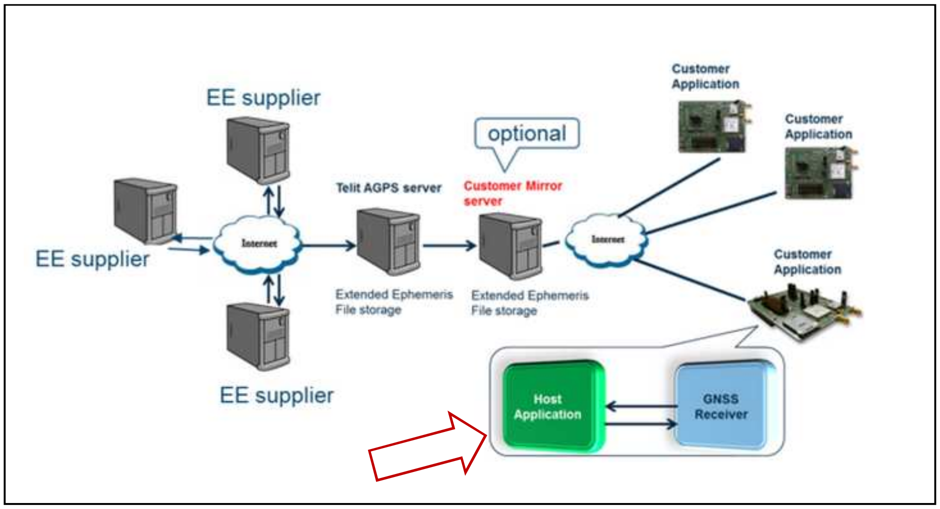


**Figure 1: GNSS receiver in a host-client architecture**

## 2.2. EPO File and Segment Structure

### 2.2.1. SAT Data Layout in EPO File

EPO file refers to the data file that is downloaded every time the host accesses a remote EE server and performs a download.

The EPO files offered for download can have different variations on the server, depending on the constellation types and validity lengths.

- Different EPO files provide a selection of constellations and validity lengths. The constellations can include GPS only, GPS + GLONASS, etc., and the validity lengths can include six (6) hours, 1 day, 7 days, etc.

- EPO file may contain a basic transfer block of EPO data, or multiple blocks. The basic transfer block is segment of EPO data that has validity of six (6) hours.

**(1) GPS-Only EPO File**

A GPS-only EPO file layout illustrated by the following diagram.

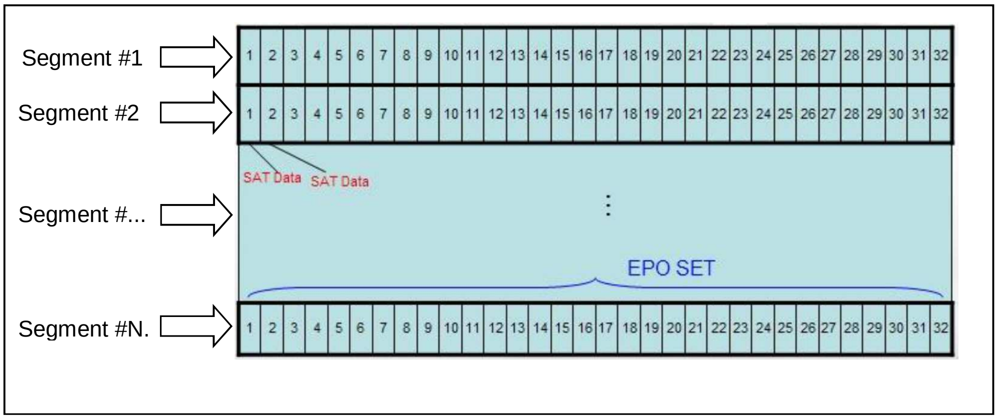


**Figure 2: EPO file format and segments**

Notice the terms that are interchangeable: an EPO SET = an EPO segment.

Refer to the diagram above.

- Each SAT Data is identified with numerical values 1 through 32 as the SVIDs.

- An EPO file consists of one or more “bigger” data blocks, referred as “segment” in this document. Each segment contains EPO data for a fixed number of satellites, based on the constellation that the file contains.

For a GPS only EPO file, the number of satellites in a segment is 32 (as shown above).

**(2) GPS + GLONASS EPO File**

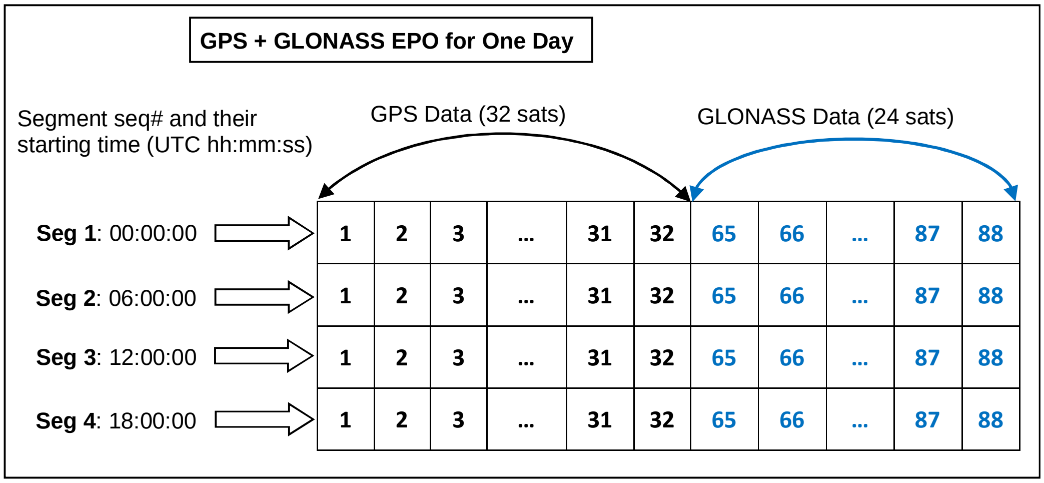
For a GPS + GLONASS EPO file, the number of satellites in a segment becomes 56 (as shown below).

### 2.2.2. EPO Segments

EPO segments are generated based on the time epoch of four (4) time periods a day in UTC time; each period has a start time.

The diagram below illustrates satellite data, for GPS EPO and GLONASS EPO in a one-day UTC time, from 00:00:00 through 23:59:59; each box represents a SAT Data with its SVID number.

On each day (UTC time), the segment #1 starts at UTC 00:00:00, and the segment #2 starts at 06:00:00, following the expiration of segment #1 EPO. The segment #3 and and #4 follow the same rule.



GPS + GLONASS EPO for One Day
Segment seq# and their starting time (UTC hh:mm:ss)
GPS Data (32 sats)
GLONASS Data (24 sats)

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Seg 1**: 00:00:00 | 1 | 2 | 3 | ... | 31 | 32 | 65 | 66 | ... | 87 | 88 |
| **Seg 2**: 06:00:00 | 1 | 2 | 3 | ... | 31 | 32 | 65 | 66 | ... | 87 | 88 |
| **Seg 3**: 12:00:00 | 1 | 2 | 3 | ... | 31 | 32 | 65 | 66 | ... | 87 | 88 |
| **Seg 4**: 18:00:00 | 1 | 2 | 3 | ... | 31 | 32 | 65 | 66 | ... | 87 | 88 |

**Figure 3: 4 segment data in a one-day UTC time (GPS + GLONASS)**

These four segment in a day, as well as their start time, provides the basis of EPO data in its storage, the transfer to client module, and the use by the navigation library in producing solution results.

In particular, a segment of EPO data can only be considered valid and can be used to produce a position if the following two condition are met:

- The current time is on or after the starting time of the segment, and
- The current time is before expiration of the segment, which is at the six (6) hours following its starting time.

As a result, each segment of EPO data has six (6) hours of validity length, but generally, unless it is used at the starting time, the segment can be used in a shorter time than the provided length.

### 2.2.3. Segments in EPO File

An EPO file consists of one or more segments of EPO data. Each file has a head segment, plus consequential segments if needed.

Head segment: This is the first segment of each EPO file - it is one of four segments that are described in the 2.2.2 EPO Segments, depending on update schedule of the file and the time of a download by the host.

The EPO files at EE server are updated periodically, but each file's update schedule is a little differently based on the validity length, so the makeup of an EPO file with respect to the segments are a little different.

| Sat Type | Validity Length | Head Segment | Total Seg. in a File | File Name |
|---|---|---|---|---|
| GPS only<br>Or<br>GPS +<br>GLONASS | 6 hours | One of four segment depending on the time of download | 1 | EPO_GPS_6h |
| | 1 Day | | 4 | EPO_GPS_1<br>Or<br>EPO_GR_1 |
| | 3 Day | | 12 | EPO_GPS_3<br>Or<br>EPO_GR_3 |
| | 5 Day | | 20 | EPO_GPS_5<br>Or<br>EPO_GR_5 |
| | 7 Day | Seg 1: 00:00:00 | 28 | EPO_GPS_7<br>Or<br>EPO_GR_7 |
| | 14 Day | | 56 | EPO_GPS_14<br>Or<br>EPO_GR_14 |

- EPO files with validity lengths from a 6-hour through 5 days:

These files are updated at a six-hour interval, before the starting times, respectively. At any time of a day when a host accesses the server and performs a download of an EPO file, it has the heading segment with the starting time no more than six hours earlier.

For an EPO file that has six (6) hour validity period (the 6-hour file), it will contain only one segment EPO data, thus it is the smallest EPO file in size.

For an EPO file that has one (1) day validity, the file will contain four (4) segment of data.

- EPO files with validity lengths 7 and 14 days:

These file are updated at 24-hour period with the heading segment set at the segment #1 of each day (in UTC time).

- The validity length of an EPO file determines the number of segments the file contains.

The number of segments contained in an EPO file is proportional to the validity period of an EPO file, and each segment has a validity time period associated with it.

## 2.2.4. GPS Hour Data and SVID

The basic unit of an EPO file is SAT Data, which corresponds to the predicated satellite orbits data for one satellite in the GPS constellation, or any other constellation in the GNSS system.

Based on the spec of the EPO file, the data size of a SAT Data is 72 bytes, which is considered a sequence of 18 WORDs (1 WORD = 4 bytes, LSB first).

| **WORD Count** | **WORD 0** | | **WORD 1** | **WORD 2** | **WORD …** | **WORD 17** |
|---|---|---|---|---|---|---|
| **Byte Index** | [0]…[2] | [3] | [4] … [7] | **……** | **……** | [68] .. [71] |
| **Parameter** | GPS_Hour | SVID | Data | Data | Data… | Checksum |

**Table 1: SAT Data format**

Among the data provided above, the following data will be decoded and used by the host in the Host-EPO process.

- GPS_Hour: from byte[0]…byte[2]

  Gps_Secs = GPS_Hour * 3600

  GPS_Week Number = GPS_Secs / 604800

  GPS TOW = GPS_Secs % 604800

- SVID: from byte[3]

  - GPS satellites are identified by their PRN numbers that range from 1 through 32. The Sat ID numbers (a.k.a. SVID) of GPS constellation are the same as their PRN numbers.

  - GLONASS satellites are identified by their slot numbers that range from 1 through 24. The Sat IDs for GLONASS are assigned by the slot number + 64. Thus Sat IDs range 65 through 88.

The SVID of each SAT Data can be decoded from the byte[3].

If the SVID is decoded as 0, the SVID has been flagged as unhealthy by the EPO file, thus is not expected to be sent across.

## 2.2.5. EPO File Sizes

Based on the spec of the EPO file, the data size of a SAT Data is 72 bytes, thus the EPO file sizes can be calculated according to the validity periods, as the following.

| EPO Validity Period | File Size (bytes)<br>GPS-Only | File Size (bytes)<br>GPS + GLONASS |
|---|---|---|
| One six-hour | 2304 | 4032 |
| One day | 9216 | 16128 |
| Three day | 27648 | 48384 |
| Seven day | 64512 | 112896 |
| 14 day | 129024 | 225792 |
| 30 day | 276480 | 483840 |

**Table 2: EPO file sizes**

Note that the 30-day EPO file is listed here for information purpose. For the current Telit's EPO feature, it is not supported by the module non-volatile storage media.

It should be noted that the EPO file size increases noticeably when the GNSS satellite type from GPS only changes to GPS + GLONASS. Because of this jump in file size, the storage capacity to support a 14-day GPS only will shrink to a 7-day GPS + GLONASS EPO.

## 2.3. Host-EPO Messages

### 2.3.1. PMTK Protocol

The message format between the host and the client is the text format.

The data is sent in the text format, not a binary format.

### 2.3.2. Message Types

The following proprietary NMEA messages are used to support the Host-EPO feature.

| Packet Type | Message Header | Description |
|---|---|---|
| 740 | $PMTK740, ... | Reference time (UTC) |
| 741 | $PMTK741, … | Reference location |
| 721 | $PMTK721, … | EPO data |

**Table 3: Message types for Host-EPO**

**(1) Reference time (PMTK740)**

In general, the host shall have a way, independent of GPS time that may be available from the GNSS module output, to know the time. Thus the time information is available when a Host-EPO is chosen as the source of reference time.

The host can use $GPRMC to retrieve UTC date, time, and position information, to keep its information current.

The accuracy of reference UTC time shall be within three (3) seconds.

User needs to be aware that the system time aiding requires UTC time, not the local time of the user. The local time usually contains a number of hours as offset from UTC time.

If the status field, the second field after $GPRMC, is not “A” (Active), do not send this data.

**(2) Reference Location (PMTK741)**

The host can use $GPRMC to retrieve and position information, to keep its location information current.

Refer to the (1) Reference time for the requirement for the reference positon.

The accuracy of the location shall be within 30 km from the true position.

**(3) Host EPO data (PMTK721)**

This message contains the EPO data to be sent to the client, one sentence per satellite data.

**(4) Receiver's Ack Message: PMTK001 (GPS Ack)**

The $PMTK001… message is not a part of definition for the EPO feature and communication protocol, but it will be sent out by the receiver as Acknowledgement message to a $PMTK… message. As a result there will be $PTMK001 messages received by the host in response to the $PMTK740 and $PMTK741 messages.

## 2.4. Flow of Host-EPO

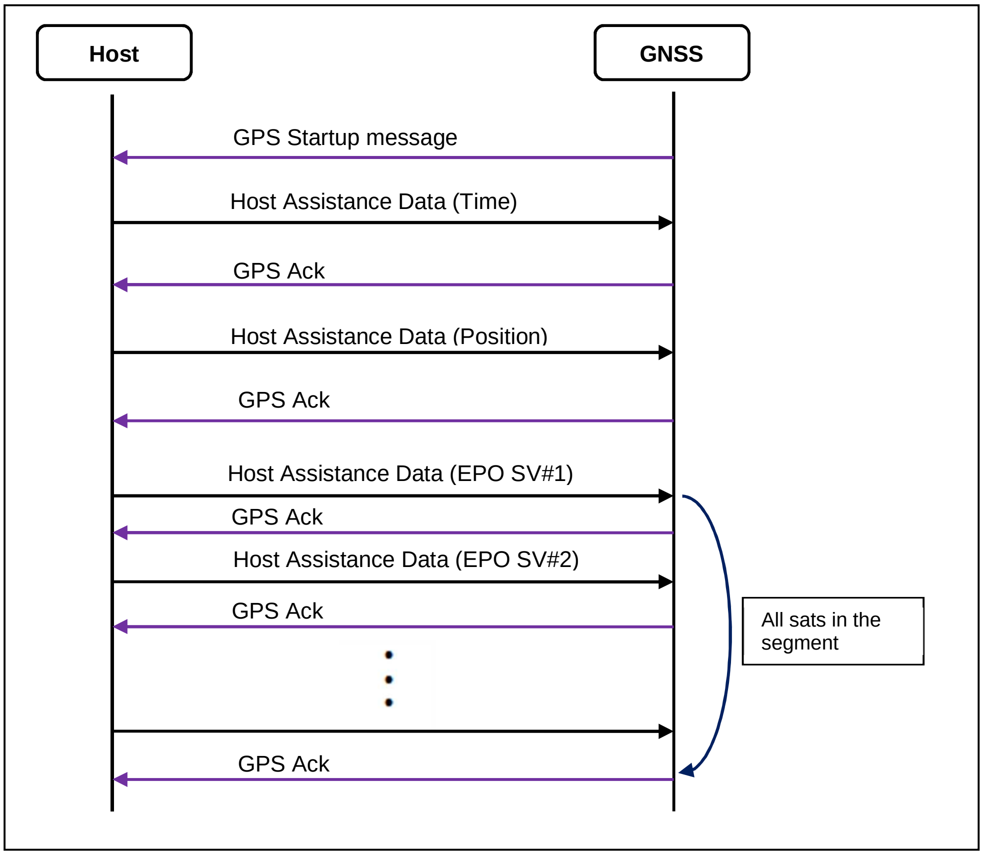

Host
GNSS
GPS Startup message
Host Assistance Data (Time)
GPS Ack
Host Assistance Data (Position)
GPS Ack
Host Assistance Data (EPO SV#1)
GPS Ack
Host Assistance Data (EPO SV#2)
GPS Ack
All sats in the segment
GPS Ack

**Figure 4: Flow of Host-EPO**

Based on the implementation of the firmware on the module, GPS Ack detection is no longer required in the implementation of firmware in the host; the host assistance data messages can be sent across without waiting for the acknowledgment from the client module.

This scheme dictates that the whole segment of satellite data will be sent across in the number of messages consecutively. In the case of GPS constellation, there are 32 satellite in each segment, there will be 32 messages to be sent.

# 3. Host-EPO Detailed Information

## 3.1. Start of Host-EPO

On each power up or software reset, the GNSS module, as client, will output a start-up message “$PMTK010,001” to notify the host that it has completed initialization and ready to receive commands in PMTK format. After receiving the start-up message, the host can send the assistance data.

Based on this paradigm, the host shall send the Host-EPO data on receiving the system start-up message “$PMTK010,001” message at the time a fast TTFF is chosen.

## 3.2. Message Format

### 3.2.1. Referent Time (PMTK740)

This sentence contains the system time (UTC time) as time-aiding.

The accuracy of reference UTC time shall be within three (3) seconds.

[Message Format]

$PMTK740,YYYY,MM,DD,hh,mm,ss*CS<CR><LF>

| Name | Unit | Range | Description |
|---|---|---|---|
| $PMTK740 | | | Reference UTC Time |
| YYYY | Year | 1980 and after | UTC time: year in 4 digits |
| MM | Month | 1 - 12 | UTC time: month, 1 or 2 digits |
| DD | Day | 1 – 31 | UTC time: day, in 1 or 2 digits |
| hh | Hour | 0 – 23 | UTC time, hour, in 1 or 2 digits |
| mm | Minute | 0 – 59 | UTC time: minute, in 1 or 2 digits |
| ss | Second | 0 – 59 | UTC time: second, 1 or 2 digits |
| CS | n/a | n/a | 8-bit exclusive-or checksum of all bytes between the ‘$’ and the ‘*’ characters, hexadecimal |

**Table 4: Reference Time message**

[Sample Message]

The packet indicates that the current UTC time 2010/Feb/10 09:00:58.

$PMTK740,2010,2,10,9,0,58*05<CR><LF>

There will be a “$PMTK001,740,3*33” message sent out by the receiver as the Ack to this message.

### 3.2.2. Reference Location (PMTK741)

This message contains reference location to be sent to the GNSS receiver, as well as the time of the position (as reported by the receiver).  It is the age information of the position as known by the host.

The accuracy of the location shall be within 30 km from the true position.

[Message Format]

$PMTK741,Lat,Lon,Alt,YYYY,MM,DD,hh,mm,ss*CS<CR><LF>

| Name | Unit | Range | Description |
|---|---|---|---|
| $PMTK741 | | | Reference location without accuracy information |
| Lat | Degree | -90.0 ~  90.0 | WGS84 geodetic latitude.<br>Express this value in floating-point with 6 decimal digits precision. |
| Lon | Degree | -180.0 ~ 180.0 | WGS84 geodetic longitude.<br>Express this value in floating-point with 6 decimal digits precision. |
| Alt | m | | WGS84 ellipsoidal altitude. |
| YYYY | Year | 1980 and after | UTC time: year in 4 digits |
| MM | Month | 1 - 12 | UTC time: month, 1 or 2 digits |
| DD | Day | 1 – 31 | UTC time: day, in 1 or 2 digits |
| hh | Hour | 0 – 23 | UTC time, hour, in 1 or 2 digits |
| mm | Minute | 0 – 59 | UTC time: minute, in 1 or 2 digits |
| ss | Second | 0 – 59 | UTC time: second, 1 or 2 digits |
| CS | n/a | n/a | 8-bit exclusive-or checksum of all bytes between the ‘$’ and the ‘*’ characters, hexadecimal |

**Table 5: Reference Location message**

[Sample Message]

The message indicates the following information is sent to GNSS receiver in the reference location message:

- Geodetical coordinates: lat=24.772816, lon=121.022636, alt=160m
- Time info: August 1, 2011, 08:00:00

$PMTK741,24.772816,121.022636,160,2011,8,1,08,00,00*12<CR><LF>

There will be a “$PMTK001,741,3*32” message sent out by the receiver as the Ack to this message.

### 3.2.3. Host EPO Data (PMTK721)

This message contains GPS/GLONASS EPO data for a single satellite.

[Message Format]

$PMTK721,SatId,W[0],W[1],...,W[17]*CS<CR><LF>

| Name | Unit | Range | Description |
|---|---|---|---|
| $PMTK721 | | | GPS/GLONASS EPO data for a single satellite |
| SatId | | GPS: 1 ~ 32<br>GLONASS: 65 ~ 88 | Satellite IDs in hexadecimal |
| W[0] ~ W[17] | | | 18 words of data for a satellite (length 72 bytes). LSB first, and hexadecimal |
| CS | n/a | n/a | 8-bit exclusive-or checksum of all bytes between the ‘$’ and the ‘*’ characters, hexadecimal |

**Table 6: EPO data**

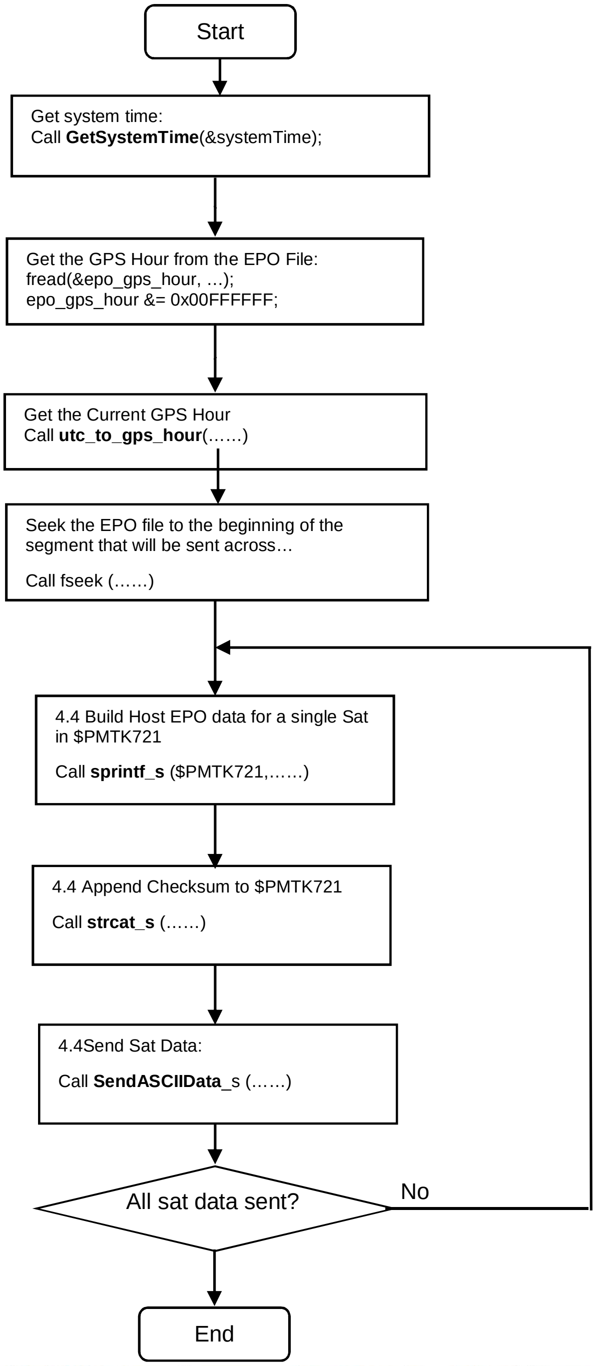
[Sample Message]

This message contains EPO data for satellite PRN 17.

$PMTK721,11,6a043d2f,d52e00,……<CR><LF>

# 4. Pseudo Code for Host-EPO Process

## 4.1. Flow Chart


Start
Get system time:
Call GetSystemTime(&systemTime);
Get the GPS Hour from the EPO File:
fread(&epo_gps_hour, …);
epo_gps_hour &= 0x00FFFFFF;
Get the Current GPS Hour
Call utc_to_gps_hour(……)
Seek the EPO file to the beginning of the segment that will be sent across…
Call fseek (……)
4.4 Build Host EPO data for a single Sat in $PMTK721
Call sprintf_s ($PMTK721,……)
4.4 Append Checksum to $PMTK721
Call strcat_s (……)
4.4Send Sat Data:
Call SendASCIIData_s (……)
All sat data sent?
No
End

The flowchart above illustrates the core flow of sending Host EPO sate data in the Host-EPO process, with the following notes:

Please note that the pseudo code is, in the format, based on the Windows platform implementation. For applications on any other platforms or environment, it is the programmer's responsibility to adopt, design and implement the functionalities for their specific running environment.

This section lists pseudo code for reference purposes.

## 4.2. Definition of Constants

```
///////////////////////////////////////////////////////////////////
/** \brief Definitions of constants
 *
 * #define constants
 */
///////////////////////////////////////////////////////////////////
#define SUCCESS   1
#define FAIL      0

#define BYTES_PER_EPO_SEG   2304
#define NUMB_SAT_PER_PACKET    3
#define NUMB_SEG_PER_DAY       4
#define MTKEPO_RECORD_SIZE     72
#define NUMB_SAT_PER_EPO_SEG    32
```

## 4.3. Start Host-EPO EPO Function

Now the data in the data port will be viewed as Binary Packet format. Please create a thread to transmit / receive binary packets for the data port.

```
{
  ……

  {
  SYSTEMTIME systemTime;
  GetSystemTime(&systemTime);   // GetSystemTime returns UTC time

  send_assistance_data(systemTime.wYear,           systemTime.wMonth,
systemTime.wDay, systemTime.wHour);

  _endthread();
  }
}
```

Please note that the code here is for reference and is it based on the Windows platform only, as it is depends on the programmer’s design on how to use a thread or task to run a Host-EPO update process.

## 4.4. Get GPS Hour & Start of EPO Segment

This function is to translate the UTC to GPS hour.

```
// translate UTC to GPS_Hour
int
utc_to_gps_hour (int iYr, int iMo, int iDay, int iHr)
{
  int iYearsElapsed;     // Years since 1980
  int iDaysElapsed;      // Days elapsed since Jan 6, 1980
  int iLeapDays;         // Leap days since Jan 6, 1980
  int i;

    // Number of days into the year at the start of each month (ignoring
leap years)
const unsigned short doy[12] =
  0,31,59,90,120,151,181,212,243,273,304,334};

  iYearsElapsed = iYr - 1980;
  i = 0;
  iLeapDays = 0;

  while (i <= iYearsElapsed)
  {
    if ((i % 100) == 20)
    {
      if ((i % 400) == 20)
      {
```

```
        iLeapDays++;
      }
    }
    else if ((i % 4) == 0)
    {
      iLeapDays++;
    }

    i++;
  }

  if ((iYearsElapsed % 100) == 20)
  {
    if (((iYearsElapsed % 400) == 20) && (iMo <= 2))
    {
      iLeapDays--;
    }
  }
  else if (((iYearsElapsed % 4) == 0) && (iMo <= 2))
  {
    iLeapDays--;
  }

  iDaysElapsed = iYearsElapsed * 365 + (int)doy[iMo - 1] + iDay +
iLeapDays - 6;

  // Convert time to GPS weeks and seconds
  return  (iDaysElapsed * 24 + iHr);
}
```

This function is to send Host-EPO data.

void send_assistance_data (int iYr, int iMo, int iDay, int iHr)

```
{
  // open EPO file and read the header (assume EPO file has passed
integrity check)
  if (NULL == (fp = fopen(epoFileName, "rb")))
  {
    return;
  }

  fread(&epo_gps_hour, 4, 1, fp);
  epo_gps_hour &= 0x00FFFFFF;

  // determine the segment to use
  current_gps_hour = utc_to_gps_hour(iYr, iMo, iDay, iHr);
   segment = (current_gps_hour - epo_gps_hour) / 6;

  // seek to the begining of the segment that will be used.
  if (fseek(fp, segment*(MTKEPO_RECORD_SIZE)*(MTKEPO_SV_NUMBER),
SEEK_SET) != 0)
  {
    return;
  }
```

## 4.5. Build Host EPO Message

This portion of code takes out a SAT Data from the Host EPO file and build it in a PMTK proprietry NMEA message.

```
for (i = 0; i < MTKEPO_SV_NUMBER; i++)
{
  memset(epobuf, 0, MTKEPO_RECORD_SIZE);
  // Read one satellite EPO data
  fread(epobuf, MTKEPO_RECORD_SIZE, 1, fp);

  // assume host system is little-endian
  if (i == 0)     //* Only print out the GPS_Hr data of the first Sat!
  {
    unsigned int gpsHr = epobuf[0] & 0x00FFFFFF;
    int gpsSec = gpsHr * 3600;
    int gpsWk = gpsSec / 604800;
    int gpsTOW = gpsSec % 604800;
    // "GPS hour: %d (UTC: %02d:00 %02d/%d/%d)"
    sprintf_s(szDbgBuf, "GPS hr: %d (EPO: Wk=%d, TOW=%d)", gpsHr, gpsWk,
gpsTOW);
    PrintfDbg(szDbgBuf);
  }

  SatID = (epobuf[0] & 0xFF000000) >> 24;
  // if SatID = 0, mean one GPS satellite is unhealth, don't use it.
  if (SatID == 0)
  {
    sprintf_s(szDbgBuf, "Unhealthy sat: %d", i+1);
    PrintfDbg(szDbgBuf);
    continue;
  }
  // verify EPO GLONASS
  else if (SatID < 0 || SatID > 88)
  {
    if (SatID > 88)
    {
      sprintf_s(szDbgBuf, "Sat not in GLONASS range: %d", SatID);
      PrintfDbg(szDbgBuf);
    }
    continue;
  }

  sprintf_s(szCmdBuf,                               sizeof(szCmdBuf),
"$PMTK721,%X,%X,%X,%X,%X,%X,%X,%X,%X,%X,%X,%X,%X,%X,%X,%X,%X,%X,%X",
           SatID,
           epobuf[0], epobuf[1], epobuf[2], epobuf[3],
           epobuf[4], epobuf[5], epobuf[6], epobuf[7],
           epobuf[8], epobuf[9], epobuf[10], epobuf[11],
           epobuf[12], epobuf[13], epobuf[14], epobuf[15],
           epobuf[16], epobuf[17]);
```

UINT8 u8Checksum = 0;

```
 char wd[10] = { 0 };

 u8Checksum = calc_nmea_checksum(&szCmdBuf[1]);

 sprintf_s(wd, sizeof(wd), "*%02X\r\n", u8Checksum);

 strcat_s(szCmdBuf, sizeof(szCmdBuf), wd);

 SendASCIIData((char*)szCmdBuf);


   ......
}
```

# 5. Document History

| Revision | Date | Changes |
|---|---|---|
| 0 | 2015-07-15 | First Issue |
| 1 | 2015-07-17 | Rev 1:<br>Add to 2.3.2: Description of PMTK001<br>Add to 3.2.1, 3.2.2: the reference to PMTK001<br>Change in 2.2 EPO File and Segment Structure to further describe the EPO segment, the definitions, and composition of EPO file with respect to the validity length. |