Telit K3/K7 AGNSS EE Host Application Note

80434NT11836A Rev. 0 – 2020-09-16

TELIT
TECHNICAL
DOCUMENTATION

SPECIFICATIONS ARE SUBJECT TO CHANGE WITHOUT NOTICE

USAGE AND DISCLOSURE RESTRICTIONS

I.   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.

II.   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

III.   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.

IV.   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.

V.   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.

APPLICABILITY TABLE

PRODUCTS

- SL871
- SL871-S
- SL871L
- SL871L-S
- SL869-V2
- SL869-V2S
- SL869L-V2
- SL869L-V2S
- SE868-A
- SE868-AS
- SE868K3-A
- SE868K7-A
- SE868K3-AL
- SE878K7-AL
- SC872-A
- SC874-A

CONTENTS

FIGURE LIST

TABLES LIST

1. INTRODUCTION

1.1. Scope

This document describes the GNSS feature known as Extended Prediction Orbit (EPO) Host supported by Telit's MT33xx based GNSS modules (see Applicability table for the complete list).
The EPO Host feature designed and defined by MediaTek®, are hereby described in support of typical user applications.

1.2. Audience

This document is intended for public distribution to potential customers who are evaluating a GNSS module from the V13 firmware family as listed in the Applicability Table. This document can also be used by the customers interested in designing and implementing a host system that communicates with the Telit's GNSS module families and want to integrate the EPO Host functionality.

1.3. Contact Information, Support

For general contact, technical support services, technical questions and report documentation errors contact Telit Technical Support at:

- TS-EMEA@telit.com
- TS-AMERICAS@telit.com
- TS-APAC@telit.com

Alternatively, use:

https://www.telit.com/contact-us

For detailed information about where you can buy the Telit modules or for recommendations

on accessories and components visit:

http://www.telit.com

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.

1.4.   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.

ℹ️	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.

1.5.　　　Related Documents

[1]　Telit SE868xx-A Family Product User Guide, 1VV0301201
[2]　Telit SL871 Family Product User Guide, 1VV0301170
[3]　Telit SL869x-V2 Family Product User Guide, 1VV0301175
[4]　Telit SC872-A Product User Guide, 1VV0301202
[5]　V13 Software User Guide, 1VV0301162
[6]　V13 Software Authorized User Guide, 1VV0301550
[7]　Telit K3 AGNSS EE Injection Application Note, 80434NT11835A

2. EXTENDED EPHEMERIS

2.1. Overview

Extended Ephemeris (EE) refers to a technology developed by GNSS chip vendors to reduce Time To First Fix (TTFF), particularly in challenging RF environments affecting satellite signal reception.

Telit and GNSS chip vendors collaborate to provide GNSS technology solutions that can be used on Telit GNSS modules to reduce the Time-To-First-Fix (TTFF).

2.2. Background

For a GNSS receiver to provide a navigation solution, it must have satellite orbits data to 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 is 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.
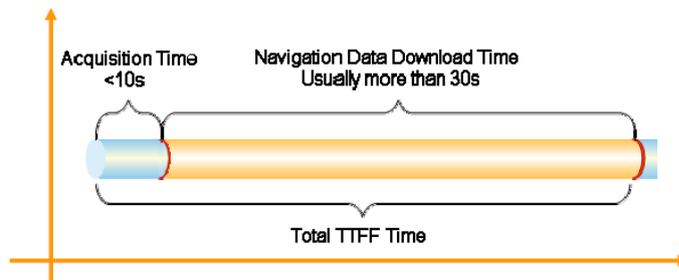
Figure 2-1 TTFF Time Breakdown

EPO technology provides pre-calculated synthetic Extended Ephemeris (EE) data to the GNSS receiver to reduce TTFF. The presence of Extended Ephemeris allows the receiver to substitute it for broadcast ephemeris downloaded from the GNSS satellite and to skip the navigation data collection requirement.

2.3. Telit EPO Technology

Telit EPO technology includes the calculation of synthetic EE, delivery, and use by a supporting GNSS module in order to help a receiver to calculate position solution and reduce TTFF performance.

The EPO is calculated, at a remote source (that is at a vendor's AGPS server), by applying predictive models of satellite motion to actual broadcast ephemeris data and is structured as a set of data segments on combination of satellite constellations which can be GPS only, or GPS + GLONASS.

For Telit EPO EE, utilized by the GNSS modules referred by this document (see applicability table), each segment contains synthetic ephemeris for a 6-hour time period. Thus, the extended ephemeris data for one day consists of 4 segments.

2.3.1. EPO Calculation by Servers

The Telit EPO server obtains ephemeris information from world-wide GNSS tracking stations and calculates EE data for all healthy GNSS satellites and store the data at an on-demand basis.

The EE-EPO data has following characteristics:

- The EE data calculation and update are done at a predetermined schedule that is adequate to provide valid data at any time.
- The server calculates a sufficient number of data blocks to produce EE and the data is stored in files which can be accessed over a network connection.
- The EE data blocks are packaged into files according to different prediction intervals. Each of the files represents a different prediction interval such as one-day, three-days, seven-days, 14 days, and so on which is the applicable length of time (in days) over which the EE within the file can be used by the receiver.

2.3.2.   EPO Management by Host Devices

To use EE-EPO data, a host device (for example a microcontroller) must be capable of establishing an HTTP connection and retrieving data files from an EE-EPO server.

The host device must also be capable to provide segments of the EPO file to the GNSS receiver module (operating as a client) over the communication link between the host and the GNSS device (that is a serial port).

2.3.3.   EPO Usage Setup

To reduce the TTFF a receiver supporting EE EPO Host technology must have an accurate initial time and an initial estimate of position, which can either be derived from a previous navigation or provided by a ost device.

During normal operating conditions, a GNSS receiver is capable to achieve fix solutions that can provide verified times and accurate positions. This timing information is retained during a standby operation mode by storing position in Battery-Backup RAM (BBRAM) and updated time in the Real-Time Clock (RTC). At start-up, the module recovers this retained data. In case it's an EPO segment provided by a Host, use this information to reduce the TTFF.

A GNSS receiver that does not have an accurate initial time and an initial estimate of position can receive this information through a Host; this latter can provide this information from another source (for example cellular network) and can share this information with the receiver by using an appropriate protocol. After providing this data the Host can inject an EPO segment to reduce the TTFF.

2.4.   Telit EPO Infrastructure

Figure 2-2, illustrates the overall data flow for Telit EE-EPO.
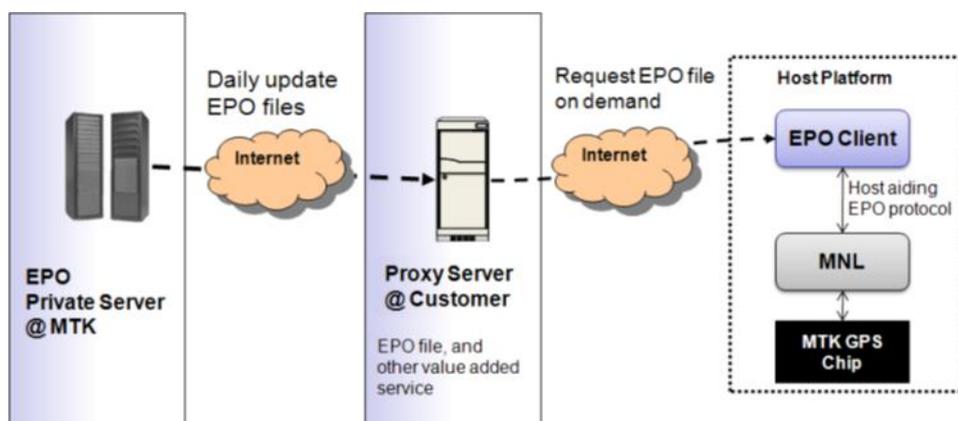
Figure 2-2 Telit EE EPO Data Flow

Refer to the above illustration for the overview of Telit EE EPO workflow, the subsections below provide more details.

.

2.4.1. MediaTek EPO Server

The MediaTek's Extended Ephemeris (EE) data is generated at an EPO private server using one of their proprietary predictive models with respect to the predictive 30-day EE interval. The time frame for each new EE data is each UTC day.

For a 30-day EE interval, a number of EE segments covering the next 30 days are generated. The EE data is compressed, formatted, and published as a set of EE files. All the satellites most recent ephemerides are used to calculate the EE data. Thus, all the EE within the files is new (less than one hour old) when they are published at the beginning of the UTC day.

2.4.2. Extended Ephemeris Files

Currently, the EE EPO data generated at the MediaTek server, can be stored in EE EPO data files that can have a different validity period and a different constellation setup.

For GPS constellation (32 GPS satellites), the EPO file size is 2304 bytes per segment; each segment covers a six-hour period of a day.

Table 2-1, details the EE EPO data validity period and the EE EPO file size.

EPO Validity Period	EPO Data Size GPS-Only	EPO Data Size GPS + GLONASS
One six-hour	2304 (2KB+)	4032 (4KB)
One day	9216 (9KB)	16128 (16KB)
Three day	27648 (27KB)	48384 (48KB)
Seven day	64512 (63KB)	112896 (110KB+)
14 day	129024 (126KB)	225792 (220KB+)
30 day	276480 (270KB)	483840 (470KB+)

Table 2-1 EPO File Validity versus Size

This information provides the essential data on some fundamental considerations of use of EE EPO in positioning applications, such as the frequency of download from the EE EPO server, the data file size of the download, and the memory requirement (storage type and the size) in the host device.

2.4.3. Telit EPO Server

The Telit EPO server retrieves EPO data files from the MediaTek's EPO server several times a day to ensure that the most recently updated files are available to Telit customers. Although, Telit provides no specifications with regard to availability and quality of service, it maintains redundant servers for added reliability of service.

The server URL is http://epo.telit.com. A username/password is required for access, which is provided by Telit once the customer has signed a Telit Service Agreement.

2.4.4. Customer Mirror Server

The customer mirror server retrieves EPO data files from the Telit EPO server and makes them available for the customer's deployed devices. Telit provides an EE Mirror Server Setup Application Note containing instructions for setting up this server. The customer can set up one or more additional mirror servers for purposes of redundancy and/or load balance.

2.4.5. Customer Device

The Host application within the customer device is responsible for downloading the EPO data file from the customer mirror server, storing this data in a suitable storage, and aiding the GNSS receiver to reduce the TTFF.

The aiding is performed by relaying to the GNSS receiver an accurate initial time (the provided time shall be within 3 seconds with reference to UTC), and initial estimate of position (the accuracy of the location shall be within 30 km from the true position) and an EPO segment with EE data for the current time slot.

2.5. Telit EPO File Structure

The EE EPO data is stored in EE EPO data files, those files can have a different validity period and a different constellation setup. An EE EPO data file may contain a basic transfer block of EPO data, or multiple blocks, depending on the validity period. The basic transfer block is a segment of EPO data (that can also be referred as EPO SET) that has validity of six hours.

2.5.1. EPO File Details

Every EPO file have one or more segments of EPO data depending on its validity period; the first segment, known as Head segment, have a starting time that depends on the validity length. 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.

EPO Validity Period	GNSS Search Mode	File Name	Segment Count
6 Hours	GPS ONLY	EPO_GPS_6H	1
1 Day	GPS ONLY	EPO_GPS_1	4
1 Day	GPS + GLO	EPO_GR_1	4
3 Day	GPS ONLY	EPO_GPS_3	12
3 Day	GPS + GLO	EPO_GR_3	12
5 Day	GPS ONLY	EPO_GPS_5	20
5 Day	GPS + GLO	EPO_GR_5	20
7 Day	GPS ONLY	EPO_GPS_7	28
7 Day	GPS + GLO	EPO_GR_7	28
14 Day	GPS ONLY	EPO_GPS_14	56

Table 2-2 EPO File Naming Convention

Error! Reference source not found., defines the names used in Telit EPO Server to identify the EPO files for different validity periods and GNSS Search Mode and their segment count. These EPO files are updated periodically with an update schedule based on their validity period:

- Validity period between 6 hour and 5 days files are updated at a 6-hour interval.
- Validity period between 7 days and 14 days files are updated at a 24-hour interval.

EPO files are updated before their starting times, 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 not earlier than its update period.

2.5.2. EPO Segment Structure

EPO segments are the basic transfer block for EE EPO data files, each of them providing satellite orbits predictions for a period of six (6) hours. The EE EPO data related to a day is thus splitted in 4 EPO segments, each of them covering a span of 6 hours, starting from 00:00:00 through 23:59:59.

On each day (UTC time), 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 #4 follow the same rule.
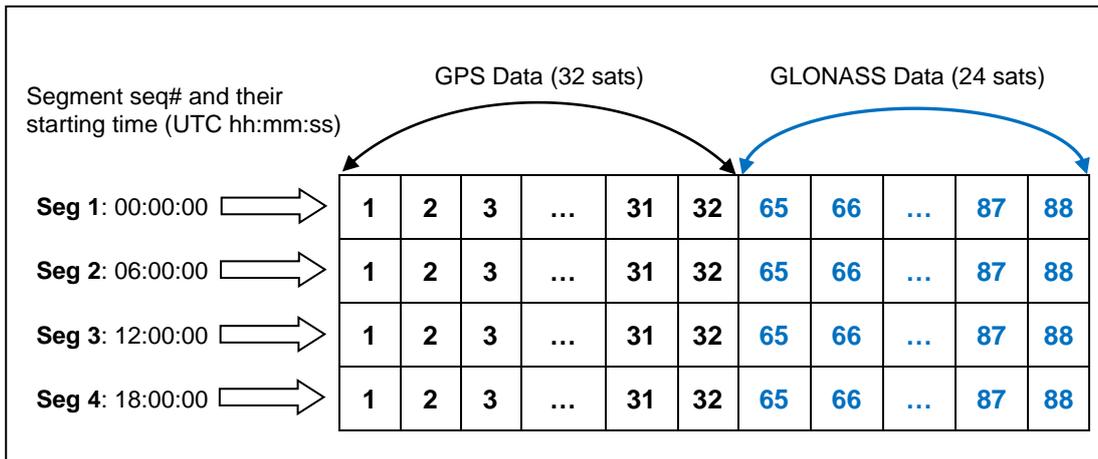
Figure 2-3 One Day UTC EPO Segment Data (GPS + GLONASS)

Only a valid EPO segment can be used for aiding in Host mode. A segment is considered valid only if it can be used to produce a position and that is true only if the following two conditions are met:

- The current time is not prior the starting time of the segment
- The current time is before expiration of the segment.

As a result, each segment of EPO data has six 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.

2.5.3. EPO SAT Data

The basic unit of an EPO segment is the SAT data, which corresponds to the predicted satellite orbits data for one satellite in the GPS constellation (or any other constellation in the GNSS system). The size of an EPO segment depends on the GNSS Search Mode:

- GPS only EPO files will have segments with 32 SAT data elements.
- GPS+GLO EPO files will have segments with 88 SAT data elements.

Parameter	Byte Index	Word Count
GPS Hour	[0] .. [2]	0
SVID	[3]	0
Data	[4] .. [7]	1
Data	[8] .. [11]	2
…	…	…

Parameter	Byte Index	Word Count
Data	[64] .. [67]	16
Checksum	[68] .. [71]	17

Table 2-3 SAT Data Structure

Table 2-3, defines the structure of a SAT Data. 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).

The timing data needs to be decoded to be used by the host in the Host-EPO process. Starting from the GPS_Hour, recorded on the first 3 bytes LSB first, the GPS time information is decoded as follows:

- GPS_Secs = GPS_Hour * 3600
- GPS_Week_Number = GPS_Secs / 604800
- GPS_Time_Of_Week = GPS_Hour % 604800

The SVID, which is encapsulated on the fourth byte, identifies the satellite whose predictions are included for each constellation. To be used by the host in the Host-EPO process this must be decoded as follows:

- For GPS satellites the SVID field matches the PRN number of the satellite:

    SAT_ID = SVID

- For GLO satellites the SVID field matches satellite's frequency slot number:

    SAT_ID = SVID + 64

If the SVID is decoded as 0, the SVID is flagged as unhealthy by the EPO file and not sent to the receiver.

3. HOST EPO PROTOCOL

3.1. Overview

The Host-EPO procedure is supported by a set of messages that are defined by MediaTek proprietary message protocol. Each message start with the "$" character, which is then followed by the proprietary address field string that uses the Manufacturer's Mnemonic Code registered by MediaTek with the NMEA, which is "MTK".

The proprietary address field form is:

$PMTKxxx

Where, xxx represents a Sentence Identifier in decimal format referred to as PktType.

3.2. Proprietary NMEA Output Messages

3.2.1. PMTK001 – Acknowledge

The GNSS receiver sends the PMTK001 message in response to a PMTK packet command. The host receives $PMTK001 messages in response to the $PMTK740, $PMTK741 and $PMTK721 messages.

An example of this message is:

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

Field	Example	Description
Message Header	PMTK001	Acknowledge
Cmd	604	Acknowledged MTK Packet
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

Table 3-1 PMTK001 – Acknowledge Message Structure

3.3. Proprietary NMEA Input Messages

3.3.1. PMTK740 – Set Reference Time

PMTK740 message is used to provide an accurate initial UTC time (not the user local time) to the GNSS receiver, the time accuracy shall be within three seconds.

An example of this message is:

$PMTK740,2020,2,20,10,1,11*31<CR><LF>

Field	Example	Description
Message Header	PMTK740	Set Reference Time
Year	2020	UTC time year in 4 digits (valid range: 1980 and after)

Field	Example	Description
Month	2	UTC time month, 1 or 2 digits (valid range: 1 - 12)
Day	20	UTC time day, 1 or 2 digits (valid range: 1 - 31)
Hour	10	UTC time hour, 1 or 2 digits (valid range: 0 – 23)
Minute	1	UTC time minute, 1 or 2 digits (valid range: 0 – 59)
Sec	11	UTC time seconds, 1 or 2 digits (valid range: 0 – 59)

Table 3-2 PMTK740 – Set Reference Time Message Structure

3.3.2. PMTK741 – Set Reference Location

PMTK741 message is used to provide an accurate initial reference location to the GNSS receiver as well as the time of the position, the accuracy of the location shall be within 30 km from the true position.

An example of this message is:

*$PMTK741,24.772816,121.022636,160,2020,2,20,10,1,11*11<CR><LF>*

Field	Example	Description
Message Header	PMTK741	Set Reference Location
Latitude	24.772816	Geodetic latitude, floating point with 6 decimal digits precision (valid range: -90.000000 : +90.000000)
Longitude	121.022636	Geodetic longitude, floating point with 6 decimal digits precision (valid range: -180.000000 : +180.000000)
Altitude	160	Ellipsoidal altitude with reference to WGS84 datum
Year	2020	UTC time year in 4 digits (valid range: 1980 and after)
Month	2	UTC time month, 1 or 2 digits (valid range: 1 - 12)
Day	20	UTC time day, 1 or 2 digits (valid range: 1 - 31)
Hour	10	UTC time hour, 1 or 2 digits (valid range: 0 – 23)
Minute	1	UTC time minute, 1 or 2 digits (valid range: 0 – 59)
Sec	11	UTC time seconds, 1 or 2 digits (valid range: 0 – 59)

Table 3-3 PMTK741 – Set Reference Location Message Structure

3.3.3. PMTK721 – Host EPO Data

PMTK721 message is used to provide GPS/GLONASS EPO data for a single satellite to the GNSS receiver.

An example of this message is:

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

Field	Example	Description
Message Header	PMTK721	Host EPO Data
Sat Id	11	Satellite ID as decoded from SAT data, 1 or 2 digits in hexadecimal
WORD 1	6a043d2f	EPO data as decoded from SAT data, 2 to 8 digits in hexadecimal, LSB first
…	…	…
WORD 16	d52e00	EPO data as decoded from SAT data, 2 to 8 digits in hexadecimal, LSB first

Table 3-4 PMTK721 – Host EPO Data Message Structure

4. HOST EPO PROCESS

4.1. Overview

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 is ready to receive commands in PMTK format. After receiving this start-up message, the host sends the assistance data.

Based on this paradigm, the host sends the Host-EPO data on receiving the system start-up message "$PMTK010,001" message at the time a fast TTFF is chosen.
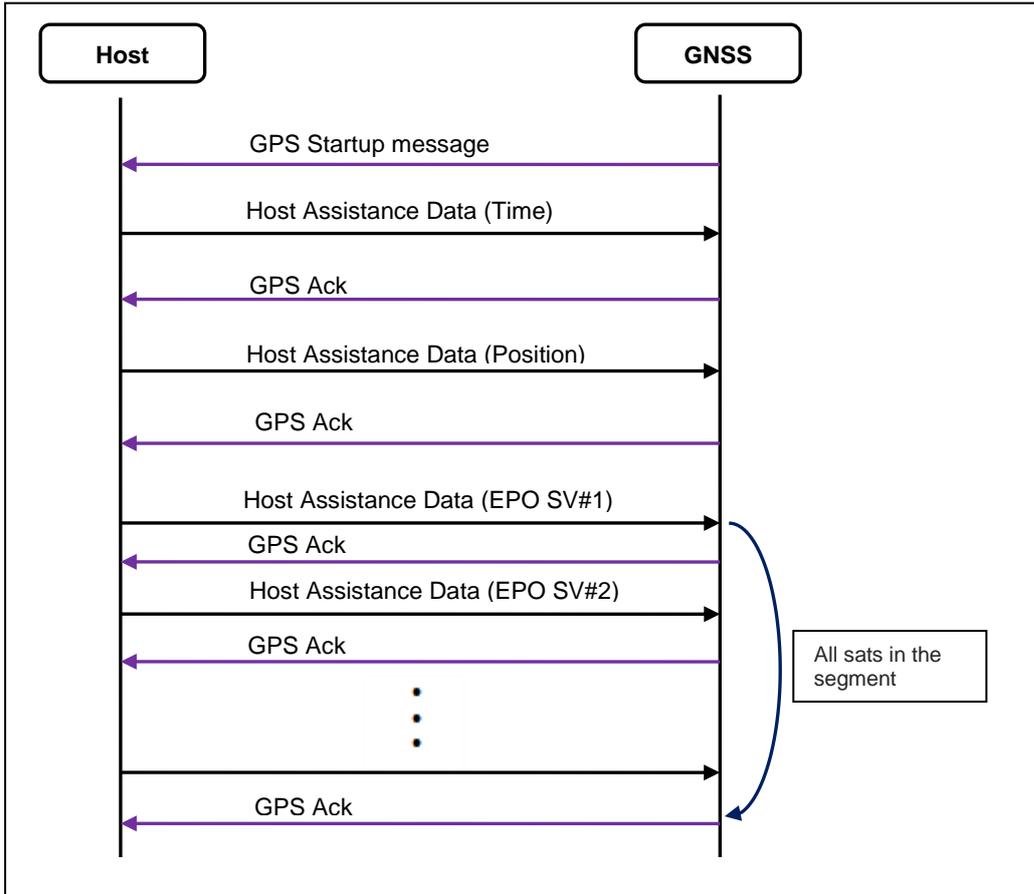
Figure 4-1 Host - EPO Message Flow

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.

4.2. Process 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 (……)                    │
           └───────────────┬───────────────────┘
                           │◄──────────────────────┐
           ┌───────────────▼───────────────────┐   │
           │ Error! Reference source not found. │   │
           │ Build Host EPO data for a single   │   │
           │ Sat in $PMTK721                    │   │
           │ Call sprintf_s ($PMTK721     )     │   │
           └───────────────┬───────────────────┘   │
                           │                        │
           ┌───────────────▼───────────────────┐   │
           │ Error! Reference source not found. │   │
           │ Append Checksum to $PMTK721        │   │
           │ Call strcat_s (……)                 │   │
           └───────────────┬───────────────────┘   │
                           │                        │
           ┌───────────────▼───────────────────┐   │
           │ Error! Reference source not found. │   │
           │ Send Sat Data:                     │   │
           │ Call SendASCIIData_s (……)          │   │
           └───────────────┬───────────────────┘   │
                           │                        │
                    ┌──────▼──────┐                 │
                   ╱               ╲                │
                  ╱ All sat data    ╲──── No ───────┘
                  ╲ sent?           ╱
                   ╲               ╱
                    └──────┬──────┘
                           │
                    ┌──────▼──────┐
                    │     End     │
                    └─────────────┘
```

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

- The format of the pseudo code is based on Windows platform implementations.
- 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.

4.3. Constants Definition

```
/////////////////////////////////////////////////////////////////

/** \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
```

4.4. Function Start Host - EPO

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

```
{

  ……

  {

  SYSTEMTIME systemTime;

  GetSystemTime(&systemTime);   // Provides 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.

4.5. Function UTC to GPS Hour

This function translates the 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);
}
```

4.6.   Function Send Host - EPO Data

This function sends the 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;
  }

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;
```

```c
    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.   ACRONYMS

	Description
ASCII	American Standard Code for Information Interchange
BE	Broadcast Ephemeris
EE	Extended Ephemeris
EPO	Extended Prediction Orbit
GLONASS	Global Navigation Satellite System
GNSS	Global Navigation Satellite System
NMEA	National Marine Electronics Association
OEM	Original Equipment Manufacturer
RTC	Real Time Clock
SRAM	Static Random Access Memory
TTFF	Time To First Fix
TTSC	Telit Technical Support Centre
UART	Universal Asynchronous Receiver Transmitter
USB	Universal Serial Bus
UTC	Co-ordinated Universal Time

6. DOCUMENT HISTORY

Revision	Date	Changes
0	2020-09-16	First issue

SUPPORT INQUIRIES

Link to **www.telit.com** and contact our technical support team for any questions related to technical issues.

www.telit.com

Telit