

# LC02H&LC29H&LC79H 系列

## AGNSS 应用指导

**GNSS 模块系列**

版本：1.0

日期：2023-06-29

状态：受控文件



上海移远通信技术股份有限公司（以下简称“移远通信”）始终以为客户提供最及时、最全面的服务为宗旨。如需任何帮助，请随时联系我司上海总部，联系方式如下：

上海移远通信技术股份有限公司  
上海市闵行区田林路 1016 号科技绿洲 3 期（B 区）5 号楼 邮编：200233  
电话：+86 21 5108 6236 邮箱：[info@quectel.com](mailto:info@quectel.com)

或联系我司当地办事处，详情请登录：<http://www.quectel.com/cn/support/sales.htm>。

如需技术支持或反馈我司技术文档中的问题，请随时登陆网址：  
<http://www.quectel.com/cn/support/technical.htm> 或发送邮件至：[support@quectel.com](mailto:support@quectel.com)。

## 前言

移远通信提供该文档内容以支持客户的产品设计。客户须按照文档中提供的规范、参数来设计产品。同时，您理解并同意，移远通信提供的参考设计仅作为示例。您同意在设计您目标产品时使用您独立的分析、评估和判断。在使用本文档所指导的任何硬软件或服务之前，请仔细阅读本声明。您在此承认并同意，尽管移远通信采取了商业范围内的合理努力来提供尽可能好的体验，但本文档和其所涉及服务是在“可用”基础上提供给您。移远通信可在未事先通知的情况下，自行决定随时增加、修改或重述本文档。

## 使用和披露限制

### 许可协议

除非移远通信特别授权，否则我司所提供硬软件、材料和文档的接收方须对接收的内容保密，不得将其用于除本项目的实施与开展以外的任何其他目的。

### 版权声明

移远通信产品和本协议项下的第三方产品可能包含受移远通信或第三方材料、硬软件和文档版权保护的相关资料。除非事先得到书面同意，否则您不得获取、使用、向第三方披露我司所提供的文档和信息，或对此类受版权保护的资料进行复制、转载、抄袭、出版、展示、翻译、分发、合并、修改，或创造其衍生作品。移远通信或第三方对受版权保护的资料拥有专有权，不授予或转让任何专利、版权、商标或服务商标权的许可。为避免歧义，除了正常的非独家、免版税的产品使用许可，任何形式的购买都不可被视为授予许可。对于任何违反保密义务、未经授权使用或以其他非法形式恶意使用所述文档和信息的违法侵权行为，移远通信有权追究法律责任。

### 商标

除另行规定，本文档中的任何内容均不授予在广告、宣传或其他方面使用移远通信或第三方的任何商标、商号及名称，或其缩略语，或其仿冒品的权利。

### 第三方权利

您理解本文档可能涉及一个或多个属于第三方的硬软件和文档（“第三方材料”）。您对此类第三方材料的使用应受本文档的所有限制和义务约束。

移远通信针对第三方材料不做任何明示或暗示的保证或陈述，包括但不限于任何暗示或法定的适销性或特定用途的适用性、平静受益权、系统集成、信息准确性以及与许可技术或被许可人使用许可技术相关的不侵犯任何第三方知识产权的保证。本协议中的任何内容都不构成移远通信对任何移远通信产品或任何其他硬软件、设备、工具、信息或产品的开发、增强、修改、分销、营销、销售、提供销售或以其他方式维持生产的陈述或保证。此外，移远通信免除因交易过程、使用或贸易而产生的任何和所有保证。

## 隐私声明

为实现移远通信产品功能，特定设备数据将会上传至移远通信或第三方服务器（包括运营商、芯片供应商或您指定的服务器）。移远通信严格遵守相关法律法规，仅为实现产品功能之目的或在适用法律允许的情况下保留、使用、披露或以其他方式处理相关数据。当您与第三方进行数据交互前，请自行了解其隐私保护和数据安全政策。

## 免责声明

- 1) 移远通信不承担任何因未能遵守有关操作或设计规范而造成损害的责任。
- 2) 移远通信不承担因本文档中的任何因不准确、遗漏、或使用本文档中的信息而产生的任何责任。
- 3) 移远通信尽力确保开发中功能的完整性、准确性、及时性，但不排除上述功能错误或遗漏的可能。除非另有协议规定，否则移远通信对开发中功能的使用不做任何暗示或法定的保证。在适用法律允许的最大范围内，移远通信不对任何因使用开发中功能而遭受的损害承担责任，无论此类损害是否可以预见。
- 4) 移远通信对第三方网站及第三方资源的信息、内容、广告、商业报价、产品、服务和材料的可访问性、安全性、准确性、可用性、合法性和完整性不承担任何法律责任。

版权所有 ©上海移远通信技术股份有限公司 2023，保留一切权利。

**Copyright © Quectel Wireless Solutions Co., Ltd. 2023.**

## 文档历史

文档信息	
标题	LC02H&LC29H&LC79H 系列 AGNSS 应用指导
副标题	GNSS 模块系列
文档类别	应用指导
文档状态	受控文件

## 修订记录

版本	日期	变更表述
-	2023-04-14	文档创建
1.0	2023-06-29	受控版本

# 目录

文档历史 .....	3
目录 .....	4
表格索引 .....	6
图片索引 .....	7
<b>1 引言 .....</b>	<b>8</b>
1.1. Host EPO 和 Flash EPO 的区别 .....	9
1.2. AGNSS 要求.....	10
<b>2 EPO 文件下载 .....</b>	<b>11</b>
2.1. 从服务器下载 EPO 文件 .....	11
2.2. EPO 文件格式 .....	12
2.2.1. EPO 文件格式—单 GPS.....	13
2.2.2. EPO 文件格式：单 BDS/单 Galileo.....	14
2.2.3. EPO 文件格式：GPS + GLONASS .....	16
2.3. EPO 文件类型 .....	16
2.4. EPO 文件下载流程 .....	17
2.5. EPO 文件有效期.....	18
<b>3 AGNSS 实现方法.....</b>	<b>20</b>
3.1. Flash EPO 实现 AGNSS .....	20
3.1.1. 二进制协议 .....	20
3.1.2. EPO 数据传输协议 .....	22
3.1.2.1 EPO 数据传输协议的伪代码.....	22
3.1.3. Flash EPO 流程.....	26
3.2. Host EPO 实现 AGNSS .....	27
3.2.1. Host EPO 的推荐流程 .....	27
3.2.2. 发送 EPO 的示例代码.....	28
<b>4 AGNSS 相关消息.....</b>	<b>31</b>
4.1. PAIR001 PAIR_ACK .....	31
4.2. PAIR010 PAIR_REQUEST_AIDING.....	32
4.3. PAIR470 PAIR_EPO_GET_STATUS.....	33
4.4. PAIR471 PAIR_EPO_SET_DATA .....	34
4.5. PAIR472 PAIR_EPO_ERASE_FLASH_DATA.....	35
4.6. PAIR590 PAIR_TIME_SET_REF_UTC .....	36
4.7. PAIR600 PAIR_LOC_SET_REF .....	37
<b>5 使用 QGNSS 下载 EPO 数据 .....</b>	<b>39</b>
5.1. 使用 QGNSS 下载 Flash EPO .....	39
5.2. 使用 QGNSS 下载 Host EPO.....	40
<b>6 AGNSS 实现示例.....</b>	<b>42</b>
6.1. Flash EPO 实现 AGNSS .....	42
6.2. Host EPO 实现 AGNSS .....	43

7	附录 A 参考文档及术语缩写 .....	45
8	附录 B 特殊符号 .....	47

## 表格索引

表 1: 适用模块 .....	8
表 2: Host EPO 和 Flash EPO 的区别 .....	9
表 3: AGNSS 相关命令 .....	10
表 4: 下载 EPO 文件的 URL.....	11
表 5: EPO 数据的 SVID 范围 .....	12
表 6: EPO 文件类型 .....	16
表 7: 二进制协议字段说明 .....	21
表 8: 二进制格式起始消息 .....	21
表 9: 二进制格式 EPO 数据消息 .....	21
表 10: 二进制格式结束消息 .....	21
表 11: 参考文档 .....	45
表 12: 术语和缩写 .....	45
表 13: 特殊符号 .....	47

## 图片索引

图 1: EPO 文件格式 (单 GPS) .....	13
图 2: EPO 文件的几段格式 .....	13
图 3: EPO 文件格式 (单 BDS/单 Galileo) .....	14
图 4: Galileo EPO 数据头.....	15
图 5: BDS EPO 数据头 .....	15
图 6: EPO 文件格式 (GPS + GLONASS) .....	16
图 7: EPO 文件相关下载流程.....	17
图 8: 二进制协议消息结构.....	20
图 9: Flash EPO 的 AGNSS 流程 .....	26
图 10: Host EPO 的建议流程 .....	28
图 11: 用于设置 Flash EPO 的 QGNSS 界面.....	39
图 12: 下载 Flash EPO 文件 .....	40
图 13: 用于设置 Host EPO 的 QGNSS 界面.....	41
图 14: 下载 Host EPO 文件 .....	41



# 1 引言

移远通信 GNSS 模块支持 EPO AGNSS 功能，此功能可缩短接收机首次定位时间。本文档主要介绍 EPO 文件下载、AGNSS 实现、AGNSS 相关的 PAIR 命令以及如何通过 QGNSS 工具下载 EPO 数据。

表 1：适用模块

模块系列	型号
LC02H	LC02H (BC)
	LC29H (AA)
LC29H	LC29H (BA)
	LC29H (BS)
	LC29H (CA)
	LC29H (DA)
	LC29H (EA)
LC79H	LC79H (AL)

## 1.1. Host EPO 和 Flash EPO 的区别

Host EPO 和 Flash EPO 都可以缩短 GNSS 接收机的首次定位时间，但它们之间的差异使其各自适用于不同的应用场景。

Host EPO（也称为实时 AGNSS）允许接收机在 RAM 中存储最多 6 小时的辅助数据，通过[第 4 章 AGNSS 相关消息](#)中列出的 NMEA PAIR 命令，可将这些辅助数据发送到接收机。对于 Host EPO，因数据保存在 RAM 区，在重启模块之后数据将丢失，所以需要在每次开机后重新注入。

Flash EPO 允许接收机在 Flash 中存储 3 天、7 天或 14 天的辅助数据，这些辅助数据通过二进制协议发送到接收机。Flash EPO 使接收机能够在信息过期之前重复使用存储在 Flash 中的所有辅助信息。EPO 文件的有效期限见[第 2.5 章 EPO 文件有效期](#)。

表 2: Host EPO 和 Flash EPO 的区别

类型	Flash EPO	Host EPO
存储空间	Flash	RAM
存储容量	3 天、7 天或 14 天的辅助数据	6 小时的辅助数据
协议	二进制协议	NMEA 协议

### 备注

单 GPS EPO 文件和 GPS + GLONASS EPO 文件的 EPO 数据能在 Flash 中保存的时间最长为 14 天，单 Galileo EPO 文件最长保存时间为 7 天，单 BDS EPO 文件最长保存时间为 3 天。如果发送 30 天的单 GPS EPO 文件或 GPS + GLONASS EPO 文件，则只会存储前 14 天的 EPO 数据。

## 1.2. AGNSS 要求

主机需要向 GNSS 接收机提供参考时间、参考位置和 EPO 数据。主机提供的信息必须满足以下要求，以便 GNSS 接收机更好地使用 EPO 数据：

- 参考时间与实际时间误差应在 3 秒内，且必须为 UTC 时间。
- 参考位置与接收机实际位置的误差应在 30 公里以内。请注意，如果接收机的天空视野有限，则需要提高参考位置的精度。
- EPO 数据需有效。

通过注入辅助数据，可以缩短接收机的首次定位时间。接收机可以使用任一种辅助数据（参考时间、参考位置和 EPO 数据），但三种辅助数据都不是必需的，只有当所有的辅助数据都注入时，才能达到最佳 AGNSS 效果。禁止使用过期或无效的数据，否则可能会导致更长的 TTFF。

主机可以通过下表中列出的消息将参考时间、参考位置和 EPO 数据发送到 GNSS 接收机。有关这些消息的详情，请参阅[第 4 章 AGNSS 相关消息](#)。

表 3：AGNSS 相关命令

数据包类型	数据内容
\$PAIR471	单颗卫星的 GPS/GLONASS/Galileo/BDS EPO 数据
\$PAIR590	参考 UTC 时间
\$PAIR600	参考位置

## 2 EPO 文件下载

移远通信不提供任何有关 EPO 文件的服务级别协议。用户需要自行下载 EPO 数据到服务器并发送到设备，同时要确保 EPO 数据的可用性。

### 2.1. 从服务器下载 EPO 文件

表 4：下载 EPO 文件的 URL

EPO 类型	GNSS 类型	EPO 文件 URL	文件名称
统一 QEPO	GPS only	<a href="http://wpepodownload.mediatek.com/QGPS.DAT?vendorinfo">http://wpepodownload.mediatek.com/QGPS.DAT?vendorinfo</a>	单个名称: QGPS.DAT
统一 QEPO	GPS + GLONASS	<a href="http://wpepodownload.mediatek.com/QG_R.DAT?vendorinfo">http://wpepodownload.mediatek.com/QG_R.DAT?vendorinfo</a>	单个名称: QG_R.DAT
统一 QEPO	BDS only	<a href="http://wpepodownload.mediatek.com/QBD2.DAT?vendorinfo">http://wpepodownload.mediatek.com/QBD2.DAT?vendorinfo</a>	单个名称: QBD2.DAT
统一 QEPO	Galileo only	<a href="http://wpepodownload.mediatek.com/QGA.DAT?vendorinfo">http://wpepodownload.mediatek.com/QGA.DAT?vendorinfo</a>	单个名称: QGA.DAT
EPO	GPS only	<a href="http://wpepodownload.mediatek.com/EPO_GPS_3_X.DAT?vendorinfo">http://wpepodownload.mediatek.com/EPO_GPS_3_X.DAT?vendorinfo</a>	X = 1~10 EPO_GPS_3_1.DAT 至 EPO_GPS_3_10.DAT
EPO	GPS + GLONASS	<a href="http://wpepodownload.mediatek.com/EPO_GR_3_X.DAT?vendorinfo">http://wpepodownload.mediatek.com/EPO_GR_3_X.DAT?vendorinfo</a>	X = 1~10 EPO_GR_3_1.DAT 至 EPO_GR_3_10.DAT
EPO	BDS only	<a href="http://wpepodownload.mediatek.com/EPO_BDS_3.DAT?vendorinfo">http://wpepodownload.mediatek.com/EPO_BDS_3.DAT?vendorinfo</a>	EPO_BDS_3.DAT
EPO	Galileo only	<a href="http://wpepodownload.mediatek.com/EPO_GAL_X.DAT?vendorinfo">http://wpepodownload.mediatek.com/EPO_GAL_X.DAT?vendorinfo</a>	X = 3 或 7 EPO_GAL_3.DAT 或 EPO_GAL_7.DAT

以下是完整的 URL 示例：

http://wpepodownload.mediatek.com/QGPS.DAT?vendor=AAA&project=BBB&device\_id=CCC

- 查询字符串以 “?” 开头，以 “&” 分隔。
  - “vendor” 和 “project”（示例中的 AAA 和 BBB）的值由移远通信发布。请联系移远通信技术支持获取该值。
- “device\_id”（示例中的 CCC）的值由客户指定，必须保证每台设备唯一。如果设备有 IMEI，则推荐使用 IMEI。

## 备注

1. 最多有 10 个文件，因为仅 GPS 或 GPS + GLONASS EPO 文件可能包含最多 30 天的预测。30 天 EPO 分割如下：  
\_1，第 1 天到第 3 天，  
\_2，第 4 到 6 天，  
.....  
\_10，第 28 至 30 天。
2. 主机使用同一个 “device\_id”，每天访问 AGNSS 服务器的次数不应超过 20 次，否则将访问失败。

## 2.2. EPO 文件格式

本章主要介绍 EPO 文件的格式。

不同星系 EPO 文件的 SVID 号如下所示。

表 5：EPO 数据的 SVID 范围

GNSS 类型	PRN	EPO 数据 SVID
GPS	1~32	1~32
GLONASS	1~24	65~88
Galileo	1~36	101~136
BDS	1~54, 55~63	201~254, 190~198

### 2.2.1. EPO 文件格式：单 GPS

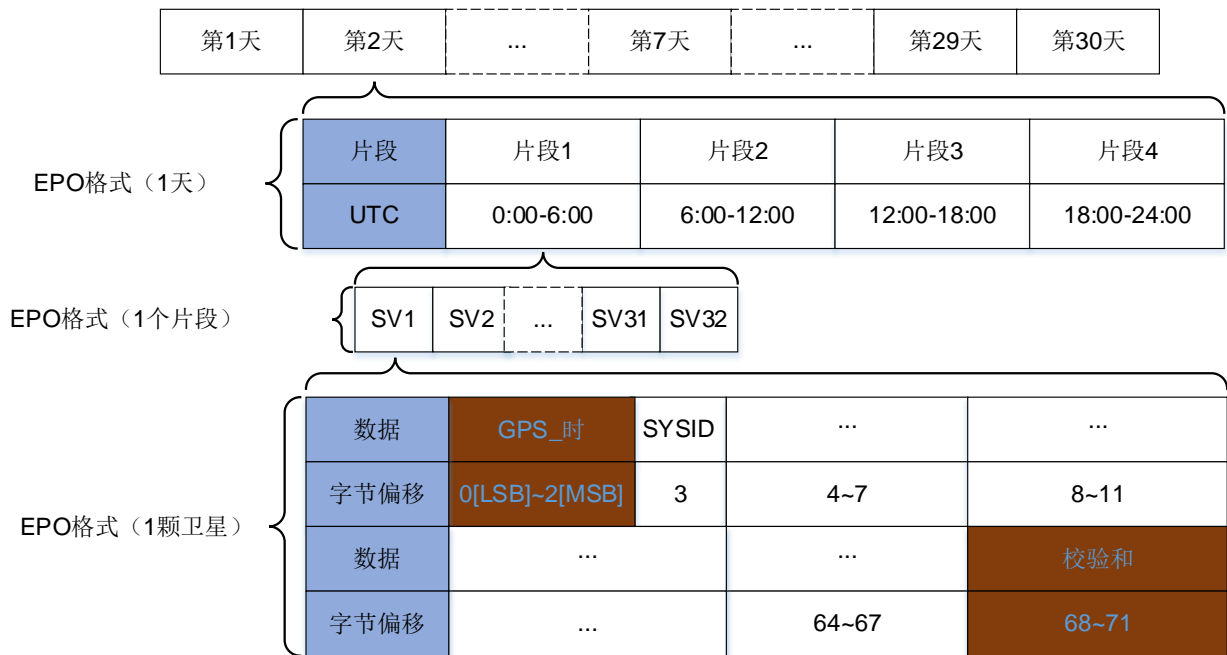


图 1：EPO 文件格式（单 GPS）

$$\text{GPS\_秒} = \text{GPS\_时} \times 3600$$

$$\text{GPS\_周} = \text{GPS\_秒} / 604800$$

$$\text{GPS 周内秒} = \text{GPS\_秒} \% 604800$$

EPO 文件包含 GPS 时间（GPS\_周、GPS\_时和 GPS\_秒）。GPS 时间的最大单位是 GPS 周，大约从 1980 年 1 月 5 日到 6 日凌晨开始计算。

下图说明了 EPO 文件的几个片段的格式。

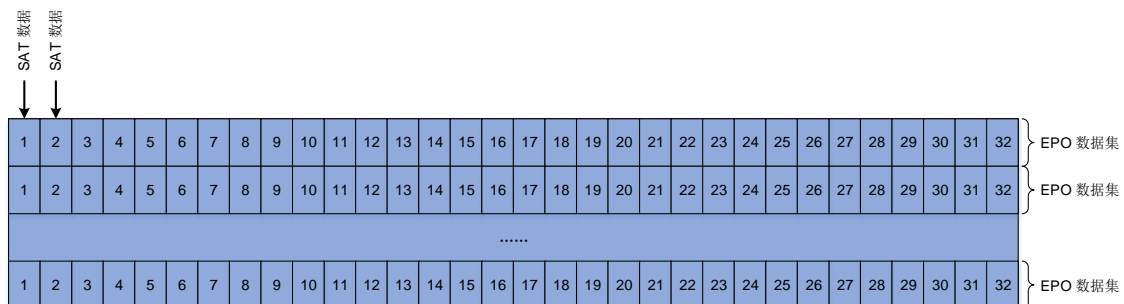


图 2：EPO 文件的几段格式

EPO 文件的基本单位是 SAT 数据，每个 SAT 数据的大小为 72 字节。一个 EPO 数据集包含 32 个 SAT 数据，所以一个 EPO 数据集大小为 2304 字节。每个 EPO 文件包含多个 EPO 数据集，因此文件大小必须是 2304 字节的倍数。EPO 数据集的有效期为 6 小时，因此，1 天内包含 4 个 EPO 数据集。

### 2.2.2. EPO 文件格式：单 BDS/单 Galileo

单 Galileo EPO 数据由 72 字节的数据头和 3 天或 7 天的基本 EPO 数据组成。单 BDS EPO 数据仅由 72 字节的数据头和 3 天的基本 EPO 数据组成。两者都没有固定大小。

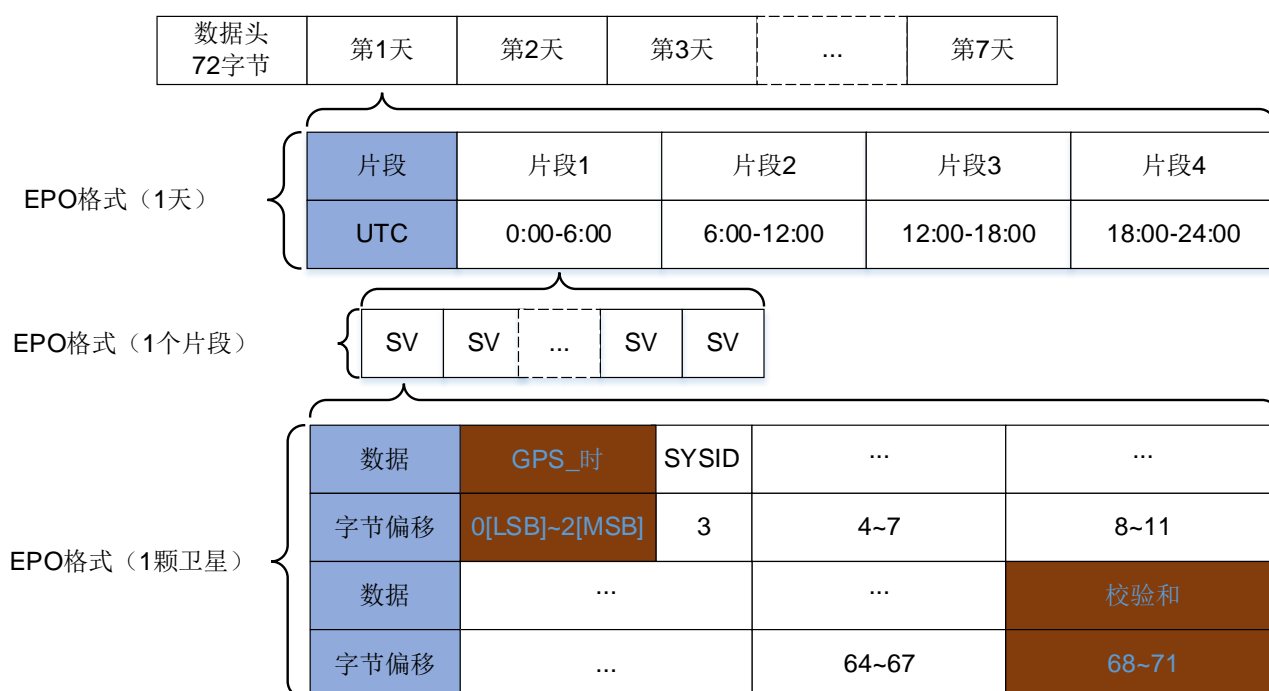


图 3：EPO 文件格式（单 BDS/单 Galileo）

72 字节的数据头包含用于计算 SV 的可用位掩码。当 SV 可用位掩码位置为 1 时，表示卫星可用，所在位数表示 SV。

可参考[图 4：Galileo EPO 数据头](#)方式解析 Galileo EPO 文件数据。

- Galileo 的 SYSID：FE。
- SV 可用位掩码：09 67 94 5D DF。
- 可用 SV 总数：22。
- 可用的 SV：1、2、3、4、5、7、8、9、11、12、13、15、19、21、24、25、26、27、30、31、33、36。

数据	***	SYSID	SV 有效位掩码 (低字节)	***	***
字节偏移	0~2	3	4[LSB]~7[MSB]	8~11	12~15
示例		FE	DF 5D 94 67		

数据	SV 有效位掩码 (高字节)	***	***	***
字节偏移	16[LSB]~19[MSB]	20~23	24~27	18~31
示例	09 00 00 00			

图 4: Galileo EPO 数据头

可参考[图 5: BDS EPO 数据头](#)方式解析 BDS EPO 文件数据。

- BDS 的 SYSID: FF。
- SV 可用位掩码: 3F FF BF FC 3F FF。
- 可用 SV 总数: 41。
- 可用 SV: 1、2、3、4、5、6、7、8、9、10、11、12、13、14、19、20、21、22、23、24、25、26、27、28、29、30、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46。

数据	***	SYSID	SV 有效位掩码 (低字节)	***	***
字节偏移	0~2	3	4[LSB]~7[MSB]	8~11	12~15
示例		FF	FF 3F FC BF		

数据	SV 有效位掩码 (高字节)	***	***	***
字节偏移	16[LSB]~19[MSB]	20~23	24~27	18~31
示例	FF 3F 00 00			

图 5: BDS EPO 数据头



### 2.2.3. EPO 文件格式：GPS + GLONASS

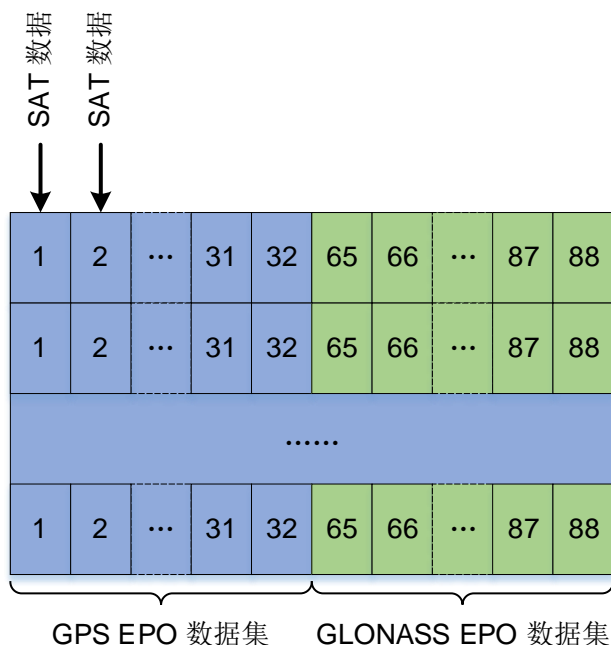


图 6: EPO 文件格式 (GPS + GLONASS)

EPO 文件的基本单位是 SAT 数据，每个 SAT 数据的大小为 72 字节。在 GPS + GLONASS EPO 文件中，一个 EPO 数据集包含 56 个 SAT 数据，所以 EPO 数据集大小为 4032 字节。每个 EPO 文件包含多个 EPO 数据集。因此文件大小必须是 4032 字节的倍数。EPO 数据集的有效期为 6 小时，因此，一天会有 4 个 EPO 数据集。

## 2.3. EPO 文件类型

EPO 数据可以以文件形式下载。可以根据数据连接的可用性和应用中存储空间情况选择最合适的文件类型进行下载。请参考[表 4: 下载 EPO 文件的 URL](#)和[表 6: EPO 文件类型](#)选择要下载的文件类型。

表 6: EPO 文件类型

EPO 类型	GNSS 类型	描述
EPO	单 GPS	3~14 天的预测轨道（星历）。 分成 5 个文件，每个文件包含 3 天的信息。
EPO	单 Galileo	3 天或 7 天预测轨道（星历）。

EPO 类型	GNSS 类型	描述
EPO	单 BDS	3 天的预测轨道（星历）。
EPO	GPS + GLONASS	3~14 天的预测轨道（星历）。 分成 5 个文件，每个文件包含 3 天的信息。
统一 QEPO	单 GPS	6 小时的预测轨道（星历）。 单个文件包含最新可用的 GPS EPO 数据。
统一 QEPO	GPS + GLONASS	6 小时的预测轨道（星历）。 单个文件包含最新可用的 GPS + GLONASS EPO 数据。
统一 QEPO	单 BDS 或单 Galileo	6 小时的预测轨道（星历）。 单个文件包含最新可用的 BDS 或 Galileo EPO 数据。

## 2.4. EPO 文件下载流程

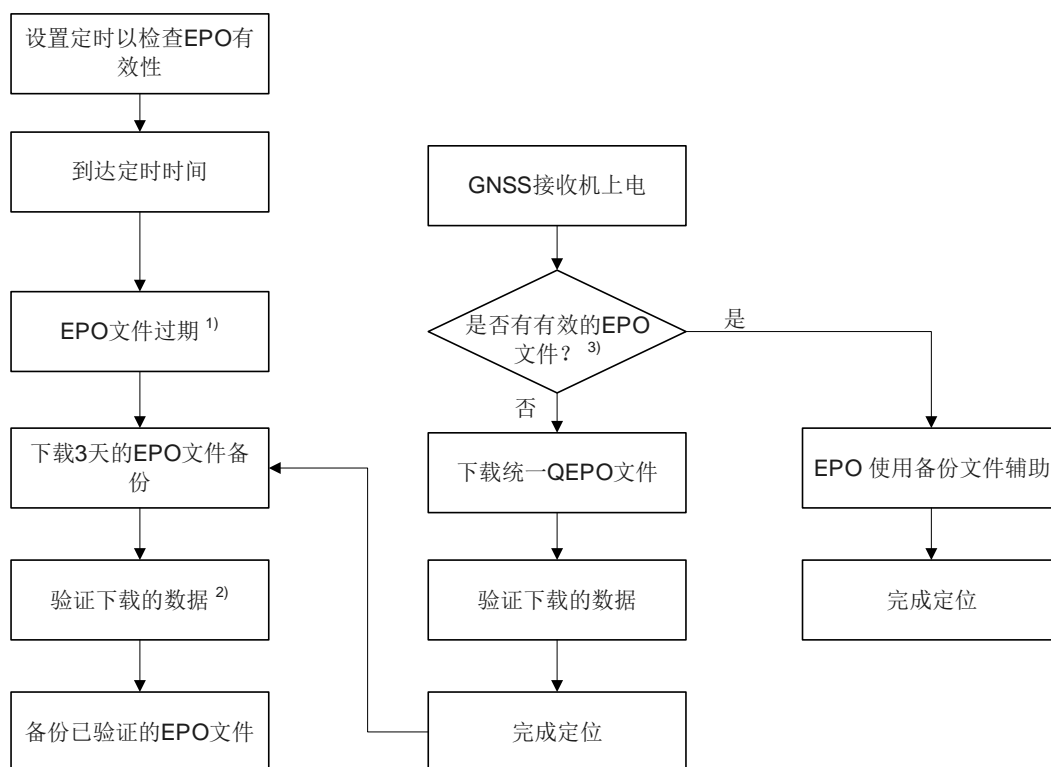


图 7：EPO 文件相关下载流程

### 备注

- 1) 用户必须知道当前的 UTC 时间才能下载有效的 EPO 文件。
- 2) 发送\$PAIR470 命令检查数据是否正确。

3. <sup>3)</sup> 如果设备长时间断电，flash 中存储的 EPO 文件可能会过期。

## 2.5. EPO 文件有效期

EPO 文件有效期与当前 UTC 时间有关。EPO 文件的有效期可以从 EPO 文件的最后一段中得到。参考 [图 1: EPO 文件格式 \(单 GPS\)](#) 或如何计算 EPO 文件有效期的示例 (GPS\_时 + 6)。需要提前 12 小时下载 EPO 文件。以下代码显示了 UTC 时间和 GPS 时间之间的转换。

```
void utc_to_gpstime(kal_uint32 year,           //Input year
                   kal_uint8  mon,           //Input month: 1~12
                   kal_uint8  day,           //Input day: 1~31
                   kal_uint8  hour,          //Input hour: 0~23
                   kal_uint8  min,           //Input minute: 0~59
                   kal_uint8  sec,           //Input second: 0~59
                   kal_int32*  wn,           //Output GPS week number
                   double*     tow)          //Output GPS time of week
{
    kal_int32 iYearsElapsed;                  //Elapsed years since 1980
    kal_int32 iDaysElapsed;                   //Elapsed days since Jan 5/Jan 6, 1980
    kal_int32 iLeapDays;                     //Leap days since Jan 5/Jan 6, 1980
    kal_int32 i;
    //Number of days at the start of each month (ignore leap years).
    kal_uint16 doy[12] = {0, 31, 59, 90, 120, 151, 181, 212, 243, 273, 304, 334};

    iYearsElapsed = year - 1980;
    i = 0;
    iLeapDays = 0;
    while (i <= iYearsElapsed)
    {
        if ((i % 100) == 20)
        {
            if ((i % 400) == 20)
            {
                iLeapDays++;
            }
        }
        else if ((i % 4) == 0)
        {
            iLeapDays++;
        }
        i++;
    }
    /* iLeapDays = iYearsElapsed / 4 + 1; */.
    if ((iYearsElapsed % 100) == 20)
    {
        if (((iYearsElapsed % 400) == 20) && (mon <= 2))
        {
            iLeapDays--;
        }
    }
}
```

```
else if (((iYearsElapsed % 4) == 0) && (mon <= 2))
{
    iLeapDays--;
}
iDaysElapsed = iYearsElapsed * 365 + doy[mon - 1] + day + iLeapDays - 6;
//Convert time to GPS weeks and seconds.
*wn = iDaysElapsed / 7;
*tow = (double)(iDaysElapsed % 7) * 86400 + hour * 3600 + min * 60 + sec;
}
```

## 3 AGNSS 实现方法

本章介绍两种 AGNSS 实现方法：Host EPO 和 Flash EPO。

- 使用 Host EPO 实现 AGNSS  
主机通过 NMEA PAIR 命令（例如\$PAIR471）向 GNSS 接收机发送 EPO 数据。
- 使用 Flash EPO 实现 AGNSS  
EPO 数据通过二进制协议下载到 GNSS 接收机的 Flash 中。

Flash EPO 比 Host EPO 保留数据的时间更长。

### 3.1. Flash EPO 实现 AGNSS

Flash EPO 可以在 Flash 中存储长达 14 天的 EPO 辅助数据，这使接收机能够使用自启动时间以来的可用数据。Flash EPO 使用二进制协议进行通信。因此，应以下文所述二进制格式将辅助数据下载到 GNSS 接收机。详见[第 3.1.2 章 EPO 数据传输协议](#)和[第 3.1.3 章 Flash EPO 流程](#)。

#### 3.1.1. 二进制协议

帧头，固定为0x04 0x24

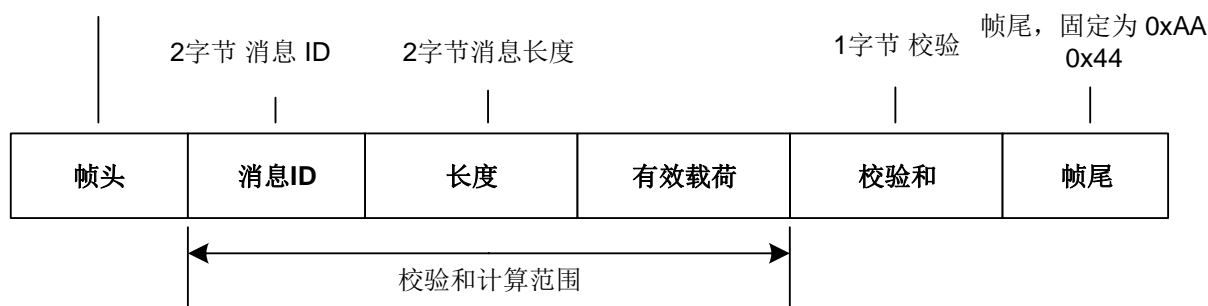


图 8：二进制协议消息结构

表 7：二进制协议字段说明

字段	长度（字节）	描述
帧头	2	固定为 0x2404。 小端。
消息 ID	2	消息 ID。
长度	2	有效载荷的长度。单位：字节。 默认数据包大小：72 字节。 小端。
有效载荷	可变	待传输的有效内容。
校验和	1	校验和是消息中帧头和校验和之间（不包括帧头和校验和）的所有字节的 8 位异或校验码。
帧尾	2	固定为 0x44AA。 小端。

EPO 二进制格式分为起始消息、EPO 数据消息和结束消息。

表 8：二进制格式起始消息

帧头	消息 ID	长度	有效载荷	校验和	帧尾
0x04 0x24	0xB0 0x04	1	'G' – GPS 'R' – GLONASS 'E' – Galileo 'C' – BDS	0x**	0xAA 0x44
2 字节	2 字节	2 字节	1 字节	1 字节	2 字节

表 9：二进制格式 EPO 数据消息

帧头	消息 ID	长度	有效载荷	校验和	帧尾
0x04 0x24	0xB1 0x04	72	EPO 数据	0x**	0xAA 0x44
2 字节	2 字节	2 字节	72 字节	1 字节	2 字节

表 10：二进制格式结束消息

帧头	消息 ID	长度	有效载荷	校验和	帧尾
0x04 0x24	0xB2 0x04	1	'G' – GPS 'R' – GLONASS	0x**	0xAA 0x44

帧头	消息 ID	长度	有效荷载	校验和	帧尾
			'E' – Galileo 'C' – BDS		
2 字节	2 字节	2 字节	1 字节	1 字节	2 字节

### 3.1.2. EPO 数据传输协议

当传输辅助数据时，主机先发送起始消息包，然后将 EPO 数据拆分成数据包发送，最后发送结束消息包。主机在将 EPO 数据传输到 GNSS 接收机时应遵循 EPO 数据传输协议。

#### 3.1.2.1 EPO 数据传输协议的伪代码

GPS + GLONASS 的 EPO 数据传输过程的伪码如下，仅供参考：

```
#define GNSS_APP_BINARY_BINARY_PREAMBLE1      (0x04)
#define GNSS_APP_BINARY_BINARY_PREAMBLE2      (0x24)
#define GNSS_APP_BINARY_BINARY_ENDWORD1       (0xAA)
#define GNSS_APP_BINARY_BINARY_ENDWORD2       (0x44)

#define GNSS_APP_BINARY_BINARY_PREAMBLE_SIZE   (2)
#define GNSS_APP_BINARY_BINARY_CHECKSUM_SIZE   (1)
#define GNSS_APP_BINARY_BINARY_ENDWORD_SIZE   (2)
#define GNSS_APP_BINARY_BINARY_CONTROL_SIZE
    (GNSS_APP_BINARY_BINARY_PREAMBLE_SIZE + \
    GNSS_APP_BINARY_BINARY_CHECKSUM_SIZE + \
    GNSS_APP_BINARY_BINARY_ENDWORD_SIZE)

#define GNSS_APP_BINARY_BINARY_MESSAGE_ID_SIZE (2)
#define GNSS_APP_BINARY_BINARY_PAYLOAD_LENGTH_SIZE (2)
#define GNSS_APP_BINARY_BINARY_PAYLOAD_HEADER_SIZE
    (GNSS_APP_BINARY_BINARY_MESSAGE_ID_SIZE + \
    GNSS_APP_BINARY_BINARY_PAYLOAD_LENGTH_SIZE)

#define GNSS_APP_BINARY_BINARY_MAX_DATA_SIZE (512)
#define GNSS_APP_BINARY_BINARY_MAX_PAYLOAD_DATA_SIZE
    (GNSS_APP_BINARY_BINARY_MAX_DATA_SIZE - \
    GNSS_APP_BINARY_BINARY_CONTROL_SIZE - \
    GNSS_APP_BINARY_BINARY_PAYLOAD_HEADER_SIZE)

typedef enum gnss_app_binary_binary_decode_results {
    GNSS_APP_BINARY_BINARY_DECODE_SUCCESS = 0,
    GNSS_APP_BINARY_BINARY_DECODE_WRONG_PARAMETER = -1,
```

```

    GNSS_APP_BINARY_BINARY_DECODE_WRONG_PREAMBLE = -2,
    GNSS_APP_BINARY_BINARY_DECODE_WRONG_CHECKSUM = -3,
    GNSS_APP_BINARY_BINARY_DECODE_WRONG_ENDWORD = -4,
}gnss_app_binary_binary_decode_results_t;
typedef struct gnss_app_binary_binary_payload {
    uint16_t message_id;
    uint16_t data_size; /* actual size of data in payload data buffer */
    uint8_t data[GNSS_APP_BINARY_BINARY_MAX_PAYLOAD_DATA_SIZE];
}gnss_app_binary_binary_payload_t;

uint8_t gnss_app_binary_calculate_binary_checksum(const
    gnss_app_binary_binary_payload_t* const payload)
{
    uint8_t checksum = 0;
    uint8_t* pheader = NULL;
    uint8_t* pdata = NULL;
    uint16_t i;
    if (NULL == payload) {
        return 0;
    }
    /* The checksum is the 8-bit exclusive OR of all bytes in the payload. */
    pheader = (uint8_t*)payload;
    for (i = 0; i < GNSS_APP_BINARY_BINARY_PAYLOAD_HEADER_SIZE; i++) {
        checksum ^= *pheader;
        pheader++;
    }
    pdata = (uint8_t*)payload->data;
    for (i = 0; i < payload->data_size; i++) {
        checksum ^= *pdata;
        pdata++;
    }
    return checksum;
}

int16_t gnss_app_binary_encode_binary_packet(uint8_t* const buffer, uint16_t
    max_buffer_size, const gnss_app_binary_binary_payload_t* const payload)
{
    uint8_t* pbyte;
    uint16_t required_length;
    if (NULL == buffer || payload == NULL) {
        return -1;
    }
    required_length = payload->data_size + GNSS_APP_BINARY_BINARY_CONTROL_SIZE +
        GNSS_APP_BINARY_BINARY_PAYLOAD_HEADER_SIZE;

```



```

    if (max_buffer_size < required_length) {
        return -1;
    }
    memset((void*)buffer, 0, max_buffer_size);
    buffer[0] = GNSS_APP_BINARY_BINARY_PREAMBLE1;
    buffer[1] = GNSS_APP_BINARY_BINARY_PREAMBLE2;
    pbyte = &buffer[2];
    memcpy(pbyte, payload, GNSS_APP_BINARY_BINARY_PAYLOAD_HEADER_SIZE);
    pbyte += GNSS_APP_BINARY_BINARY_PAYLOAD_HEADER_SIZE;
    memcpy(pbyte, payload->data, payload->data_size);
    pbyte += payload->data_size;
    *pbyte++ = gnss_app_binary_calculate_binary_checksum(payload);
    *pbyte++ = GNSS_APP_BINARY_BINARY_ENDWORD1;
    *pbyte = GNSS_APP_BINARY_BINARY_ENDWORD2;
    return required_length;
}

FILE* gnss_epo_file = NULL;
#define GNSS_MAX_EPO_NUMBER (37)
#define GNSS_MAX_RECORD_SIZE (72)
static uint32_t gnss_epo_sv_buf[(GNSS_MAX_EPO_NUMBER *
    GNSS_MAX_RECORD_SIZE) / sizeof(uint32_t)];
int16_t gnss_epo_encode_binary(uint16_t msg_id, char* buffer, uint16_t buffer_size,
    char* data_input, int32_t data_length) {
    gnss_app_binary_binary_payload_t payload;
    int16_t binary_message_length;
    memset((void*)&payload, 0, sizeof(gnss_app_binary_binary_payload_t));
    payload.message_id = msg_id;
    payload.data_size = (uint16_t)data_length;
    memcpy(payload.data, data_input, sizeof(uint8_t) * data_length);
    binary_message_length = gnss_app_binary_encode_binary_packet(buffer,
        buffer_size, &payload);
    return binary_message_length;
}

void gnss_epo_binary_demo() {
    gnss_app_binary_data_t data;
    gnss_app_binary_data_result_t result = { 0 };
    uint32_t* epobuf;
    int32_t i;
    char buffer[500];
    uint16_t length = 0;
    int32_t buffer_size = 0;
    uint8_t segment = 0;

```

```

uint8_t curr_sys_type = 'G'; //type is the GPS
gnss_epo_file = fopen("EPO_GR_3_1.DAT", "rb");
length = gnss_epo_encode_binary(1200, buffer, 512, &curr_sys_type, 1);
gnss_app_uart_send_data(buffer, length);
memset(&gnss_epo_sv_buf, 0, sizeof(gnss_epo_sv_buf));
while (gnss_epo_read_data(&gnss_epo_sv_buf, 32 * GNSS_MAX_RECORD_SIZE, segment *
    (32 + 24) * GNSS_MAX_RECORD_SIZE)) {
    segment++;
    for (i = 0; i < 32; i++) {
        epobuf = (uint32_t*)(gnss_epo_sv_buf + ((i * GNSS_MAX_RECORD_SIZE) /
            4));
        length = gnss_epo_encode_binary(1201, buffer, 512, (char*)epobuf,
            GNSS_MAX_RECORD_SIZE);
        gnss_app_uart_send_data(buffer, length);
    }
    memset(&gnss_epo_sv_buf, 0, sizeof(gnss_epo_sv_buf));
}
length = gnss_epo_encode_binary(1202, buffer, 512, &curr_sys_type, 1);
gnss_app_uart_send_data(buffer, length);

curr_sys_type = 'R'; //type is the GLONASS
length = gnss_epo_encode_binary(1200, buffer, 512, &curr_sys_type, 1);
gnss_app_uart_send_data(&data, &result);
memset(&gnss_epo_sv_buf, 0, sizeof(gnss_epo_sv_buf));
segment = 0;
while (gnss_epo_read_data(&gnss_epo_sv_buf, 37 * GNSS_MAX_RECORD_SIZE, (segment
    * (32 + 24) * GNSS_MAX_RECORD_SIZE) + (32 * GNSS_MAX_RECORD_SIZE))) {
    segment++;
    for (i = 0; i < 24; i++) {
        epobuf = (uint32_t*)(gnss_epo_sv_buf + ((i * GNSS_MAX_RECORD_SIZE) /
            4));
        length = gnss_epo_encode_binary(1201, buffer, 512, (char*)epobuf,
            GNSS_MAX_RECORD_SIZE);
        gnss_app_uart_send_data(buffer, length);
    }
    memset(&gnss_epo_sv_buf, 0, sizeof(gnss_epo_sv_buf));
}
length = gnss_epo_encode_binary(1202, buffer, 512, &curr_sys_type, 1);
gnss_app_uart_send_data(buffer, length);
fclose(gnss_epo_file);
}

```

### 3.1.3. Flash EPO 流程

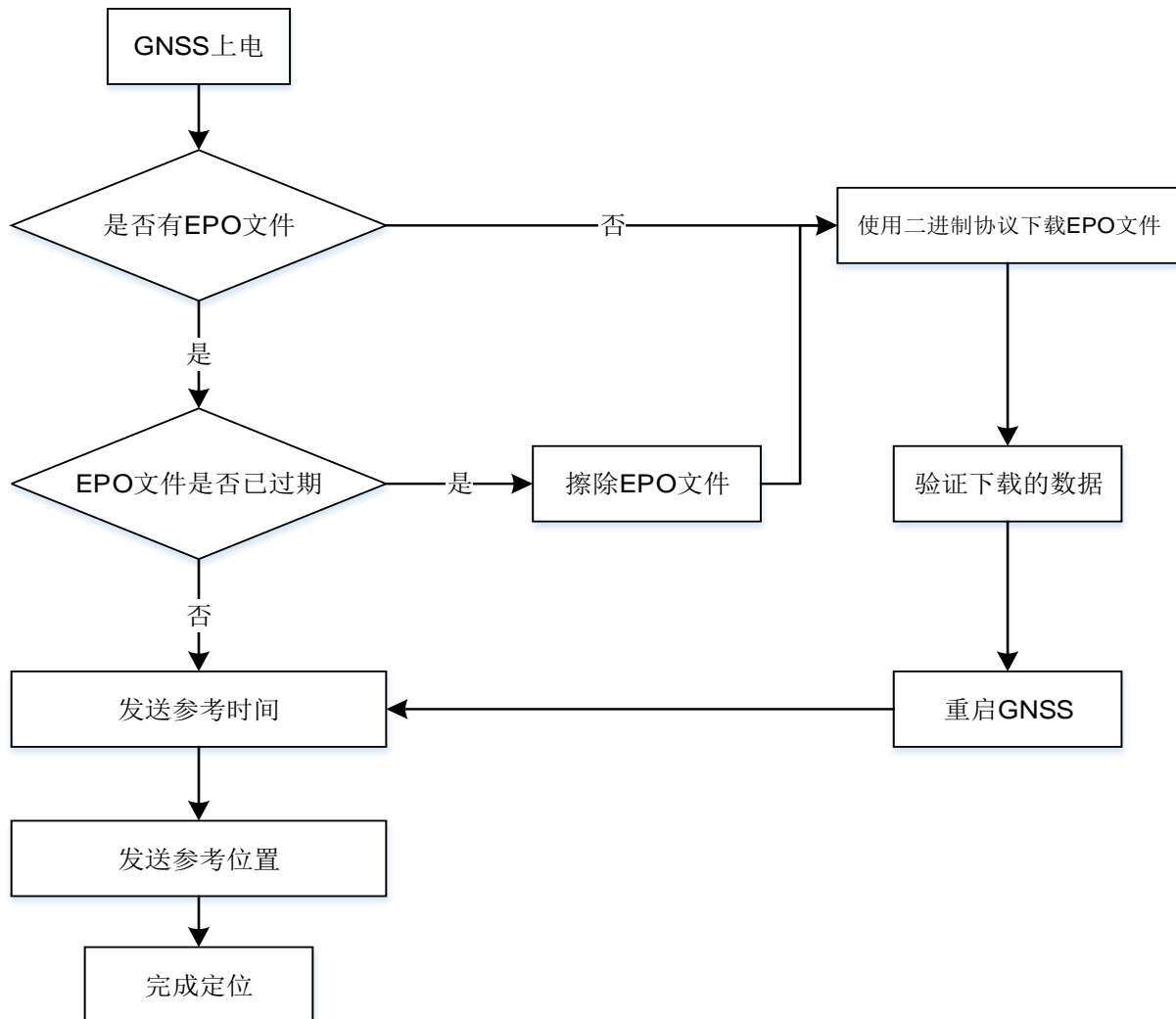


图 9: Flash EPO 的 AGNSS 流程

1. GNSS 模块上电;
2. 用\$PAIR470 命令检查 GNSS 模块 Flash 中是否有 EPO 数据;
3. 如果 Flash 中包含 EPO 数据, 继续下一步检查数据有效性。否则, 下载 EPO 数据到 GNSS 模块并验证下载的数据, 然后重启 GNSS 模块并执行步骤 5;
4. 检查 GNSS 模块 Flash 中的 EPO 文件是否过期;
5. 如果 EPO 文件仍然有效, 转到步骤 6。否则, 使用\$PAIR472 命令擦除过期的 EPO 文件, 并下载新的 EPO 文件;
6. 使用\$PAIR590 命令将参考时间发送到 GNSS 模块;
7. 使用\$PAIR600 命令将参考位置发送到 GNSS 模块;
8. 等待 GNSS 模块定位。

## 3.2. Host EPO 实现 AGNSS

Host EPO 是更为简单的基于文本的 AGNSS 实现方式，它使接收机能够执行快速启动，每次启动时必须将辅助数据发送到接收机。使用 Host EPO 时，接收机只能接收到 6 小时内有效的 1 块辅助数据。

实现 Host EPO 只需用到数个 PAIR 语句，整个数据传输可以在 NMEA 模式下进行。**\$PAIR471**、**\$PAIR590** 和 **\$PAIR600** 的详细信息，请见[第 4 章 AGNSS 相关消息](#)。

### 3.2.1. Host EPO 的推荐流程

模块上电后，当辅助数据和星历均无效时，将发送辅助请求通知其存储的 GNSS 辅助数据已失效。因此，如果模块没有上报辅助请求，建议主机收到系统启动报文后，通过发送**\$PAIR470** 来判断辅助数据是否过期。主机应按[图 10: Host EPO 的建议流程](#)所示的顺序发送辅助数据：

Host EPO 推荐流程：

1. GNSS 模块启动；
2. 主机发送参考时间；
3. 主机发送参考位置；
4. 主机发送 EPO 数据。

参考时间、参考位置和 EPO 数据必须符合[第 1.2 章 AGNSS 要求](#)中列出的要求。

#### 备注

确保主机在发送每一段 EPO 数据之后都能接收到模块返回的**\$PAIR001** 语句。

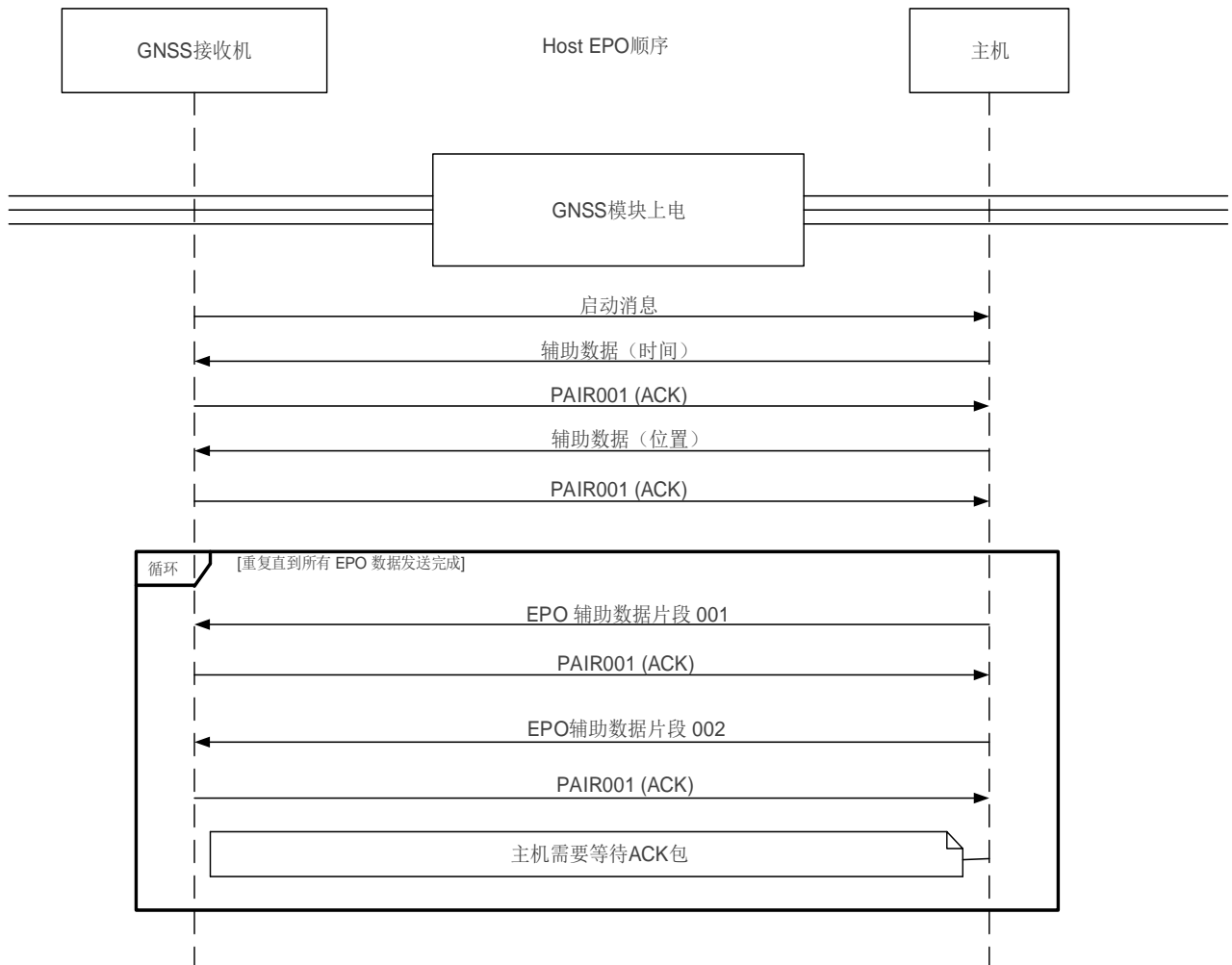


图 10: Host EPO 的建议流程

### 3.2.2. 发送 EPO 的示例代码

以下是向模块发送一段 EPO 数据的参考代码。它说明了如何为 GNSS 接收机构建 PAIR 消息。本例中不包括参考时间和参考位置的 PAIR 消息。

```

#define GNSS_GLONASS_EPO_BASE_ID (64)
#define GNSS_GALILEO_EPO_BASE_ID (100)
#define GNSS_BDS_EPO_BASE_ID (200)

#define MNL_SERVICE_MAX_COMMAND_LEN (352)
#define EPO_DEMO_RECORD_SIZE (72)

typedef enum{
    EPO_DEMO_MODE_GPS,
    EPO_DEMO_MODE_GLONASS,

```

```

    EPO_DEMO_MODE_GALILEO,
    EPO_DEMO_MODE_BEIDOU
}epo_demo_mode_t;

int32_t epo_demo_get_sv_prn(int32_t type, uint8_t *data)
{
    int32_t sv_id, sv_prn = 0;
    sv_id = data[3];

    switch(type) {
        case EPO_DEMO_MODE_GPS:
            sv_prn = sv_id;
            break;
        case EPO_DEMO_MODE_GLONASS:
            sv_prn = sv_id - GNSS_GLONASS_EPO_BASE_ID;
            break;
        case EPO_DEMO_MODE_GALILEO:
            if(sv_id == 255) {
                sv_prn = 255;
            } else {
                sv_prn = sv_id - GNSS_GALILEO_EPO_BASE_ID;
            }
            break;
        case EPO_DEMO_MODE_BDS:
            if(sv_id == 255) {
                sv_prn = 255;
            } else {
                sv_prn = sv_id - GNSS_BDS_EPO_BASE_ID;
            }
            break;
        default:
            sv_prn = 0;
    }
    return sv_prn;
}

void epo_demo_send_data(epo_demo_epo_data_t *data_p, int32_t data_num, int32_t type){

    char temp_buffer[MNL_SERVICE_MAX_COMMAND_LEN] = {0};
    uint8_t data_buffer[EPO_DEMO_RECORD_SIZE] = {0};
    int32_t i;
    int32_t sv_prn = 0;

    for(i = 0; i < data_num; i++) {

```

```

unsigned int *epobuf = (unsigned int *)data_buffer;
epo_demo_epo_fread(data_p, data_buffer, EPO_DEMO_RECORD_SIZE);
sv_prn = epo_demo_get_sv_prn(type, data_buffer);

sprintf((char *) temp_buffer,
        "471,%X,%X,%X,%X,%X,%X,%X,%X,%X,%X,%X,%X,%X,%X,%X,%X,%X,%X,%X",
        (unsigned int)type,
        (unsigned int)sv_prn,
        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]);

gnss_app_send_command_ex(temp_buffer);
memset(temp_buffer, 0, MNL_SERVICE_MAX_COMMAND_LEN);
}
}

```

## 4 AGNSS 相关消息

### 4.1. PAIR001 PAIR\_ACK

确认 PAIR 命令。模块返回\$PAIR001，通知发送方其已收到数据包。

类型：

输出

格式：

```
$PAIR001,<CommandID>,<Result>*<Checksum>
```

参数：

字段	格式	单位	描述
<CommandID>	十进制	-	待确认的命令类型。
<Result>	十进制	-	0 = 命令已成功发送。 1 = 正在处理命令，请等待结果。 2 = 命令发送失败。 3 = <CommandID>不支持。 4 = 命令参数错误，如超出范围、某些参数丢失或校验和错误。 5 = MNL 服务忙。

示例：

```
$PAIR001,0,3*38
```



## 4.2. PAIR010 PAIR\_REQUEST\_AIDING

通知模块中存储的 GNSS 辅助数据已到期。该信息在模块上电时自动输出。

类型:

输出

格式:

\$PAIR010,<Type>,<GNSS\_System>,<WN>,<TOW>\*<Checksum><CR><LF>

参数:

字段	格式	单位	描述
<Type>	十进制	-	待更新的数据类型。 0 = EPO 数据 1 = 时间 2 = 位置
<GNSS_System>	十进制	-	所需的 GNSS 数据类型。 0 = GPS 数据 1 = GLONASS 数据 2 = Galileo 数据 3 = BDS 数据
<WN>	十进制	周	周数（自适应周翻转）。
<TOW>	十进制	秒	周内秒。

示例:

//主机在收到模块输出此消息时发送 GPS EPO 数据:

\$PAIR010,0,0,2044,369413\*33

//主机在收到模块输出此消息时发送参考时间:

\$PAIR010,1,-1\*16

//主机在收到模块输出此消息时发送参考位置:

\$PAIR010,2,-1\*15

备注

GNSS 模块自动输出此消息，请勿手动向模块发送此消息。

### 4.3. PAIR470 PAIR\_EPO\_GET\_STATUS

查询存储在 GNSS 芯片中的 EPO 数据的状态。

类型：

命令

格式：

```
$PAIR470,<System_ID>*<Checksum><CR><LF>
```

参数：

字段	格式	单位	描述
<System_ID>	十进制	-	GNSS 系统 ID。 0 = GPS 1 = GLONASS 2 = Galileo 3 = BDS

结果：

返回\$PAIR001 消息和查询结果。

查询结果消息格式：

```
$PAIR470,<System_ID>,<Set>,<FWN>,<FTOW>,<LWN>,<LTOW>,<FCWN>,<FCTOW>,<LCWN>,<LCTOW>*<Checksum><CR><LF>
```

参数：

字段	格式	单位	描述
<System_ID>	十进制	-	GNSS 系统 ID。 0 = GPS 1 = GLONASS 2 = Galileo 3 = BDS
<Set>	十进制	-	存储在 GNSS 芯片中的 EPO 数据集总数。
<FWN>	十进制	-	设置存储在 Flash 中的第一个 EPO 数据集的 GPS 周数。

字段	格式	单位	描述
<FTOW>	十进制	-	设置存储在 Flash 中的第一个 EPO 数据集的 GPS 周内秒。
<LWN>	十进制	-	设置存储在 Flash 中的最后一个 EPO 数据集的 GPS 周数。
<LTOW>	十进制	-	设置存储在 Flash 中的最后一个 EPO 数据集的 GPS 周内秒。
<FCWN>	十进制	-	目前正在使用的第一个 EPO 数据集的 GPS 周数。
<FCTOW>	十进制	-	目前正在使用的第一个 EPO 数据集的 GPS 周内秒。
<LCWN>	十进制	-	目前正在使用的最后第一个 EPO 数据集的 GPS 周数。
<LCTOW>	十进制	-	目前正在使用的最后第一个 EPO 数据集的 GPS 周内秒。

示例：

```
$PAIR470,0*25
```

```
$PAIR001,470,0*38
```

```
$PAIR470,0,1,2098,194400,2098,216000,2098,194400,2098,216000*38
```

#### 4.4. PAIR471 PAIR\_EPO\_SET\_DATA

发送包含单颗卫星的 EPO 数据的数据包。

类型：

输入

格式：

```
$PAIR471,<System_ID>,<SV_ID>,<W[0]>,...,<W[17]>*<Checksum><CR><LF>
```

参数：

字段	格式	单位	描述
<System_ID>	十进制	-	GNSS 系统 ID。 0 = GPS 1 = GLONASS 2 = Galileo 3 = BDS

字段	格式	单位	描述
<SV_ID>	十六进制	-	正在发送 EPO 数据的卫星的 PRN 编号。 GPS 范围：1~32。 GLONASS 范围：1~24。 Galileo 范围：1~30。 BDS 范围：1~37。 特殊 255：BDS IONO 数据。 特殊 254：Galileo IONO 数据。
<W[0]~W[17]>	-	-	一个 EPO 段数据的 18 个字（LSB 优先）（共 72 个字节）。

结果：

返回\$PAIR001 消息。

示例：

```
$PAIR471,1,16,56056272,F2BC0244,4F19AE34,F95C534D,FAE67014,4F19AF6B,F96749BD,9F341F2
D,6F4EA9F,77DB4710,66ADAC2,9ADF3B01,8CC8B19C,29D2D20C,FC5B2E94,1000001C,11005000,
748B45F4*0A
$PAIR001,471,0*39
```

## 4.5. PAIR472 PAIR\_EPO\_ERASE\_FLASH\_DATA

擦除存储在 Flash 中的 EPO 数据。

类型：

命令

格式：

```
$PAIR472*<Checksum><CR><LF>
```

参数：

无

结果：

返回\$PAIR001 消息。

示例:

```
$PAIR472*3B
$PAIR001,472,0*3A
```

## 4.6. PAIR590 PAIR\_TIME\_SET\_REF\_UTC

发送参考 UTC 时间到 GNSS 芯片以获得更快的首次定位时间。由于时区偏移，应避免使用当地时间。要获取更快的首次定位时间，参考时间应与实际时间的误差应在 3 秒内，并且必须为 UTC 时间。

类型:

输入

格式:

```
$PAIR590,<YYYY>,<MM>,<DD>,<hh>,<mm>,<ss>*<Checksum><CR><LF>
```

参数:

字段	格式	单位	描述
<YYYY>	十进制	年	UTC 年；最小值：1980。
<MM>	十进制	月	UTC 月；范围：1~12。
<DD>	十进制	日	UTC 日；范围：1~31。
<hh>	十进制	时	UTC 时；范围：0~23。
<mm>	十进制	分	UTC 分；范围：0~59。
<ss>	十进制	秒	UTC 秒；范围：0~59。

结果:

返回\$PAIR001 消息。

示例:

```
$PAIR590,2023,3,29,9,0,58*09
$PAIR001,590,0*37
```

## 4.7. PAIR600 PAIR\_LOC\_SET\_REF

发送参考位置到 GNSS 芯片以获得更快的首次定位时间。参考位置与接收机实际位置的误差应在 30 公里以内。请注意，如果接收机的天空视野有限，则需要提高参考位置的精度。

类型：

输入

格式：

```
$PAIR600,<Lat>,<Lon>,<Height>,<AccMaj>,<AccMin>,<Bear>,<AccVert>*<Checksum><CR><LF>
```

参数：

字段	格式	单位	描述
<Lat>	十进制	度	参考纬度。范围：-90 至 90。-：南；+：北。建议用保留小数点后 6 位的浮点数表示。
<Lon>	十进制	度	参考经度。范围：-180 至 180。-：西；+：东。建议用保留小数点后 6 位的浮点数表示。
<Height>	十进制	米	参考高度。
<AccMaj>	十进制	米	长半轴 RMS 精度。
<AccMin>	十进制	米	短半轴 RMS 精度。
<Bear>	十进制	度	方位角。
<AccVert>	十进制	米	垂直 RMS 精度。

结果：

返回\$PAIR001 消息。

示例：

```
$PAIR600,24.772816,121.022636,175.0,50.0,50.0,0.0,100.0*06
$PAIR001,600,0*3D
```

## 备注

每次 GNSS 模块重启后都需要发送该命令。

# 5 使用 QGNSS 下载 EPO 数据

本章介绍如何通过 QGNSS 工具下载 EPO 数据。有关 QGNSS 的详细信息，请联系移远通信技术支持。

## 5.1. 使用 QGNSS 下载 Flash EPO

使用 QGNSS 工具下载 Flash EPO 的步骤：

1. 运行 QGNSS 工具；
2. 在主界面，点击“AGNSS”→“Assistant GNSS Offline”，如下图：

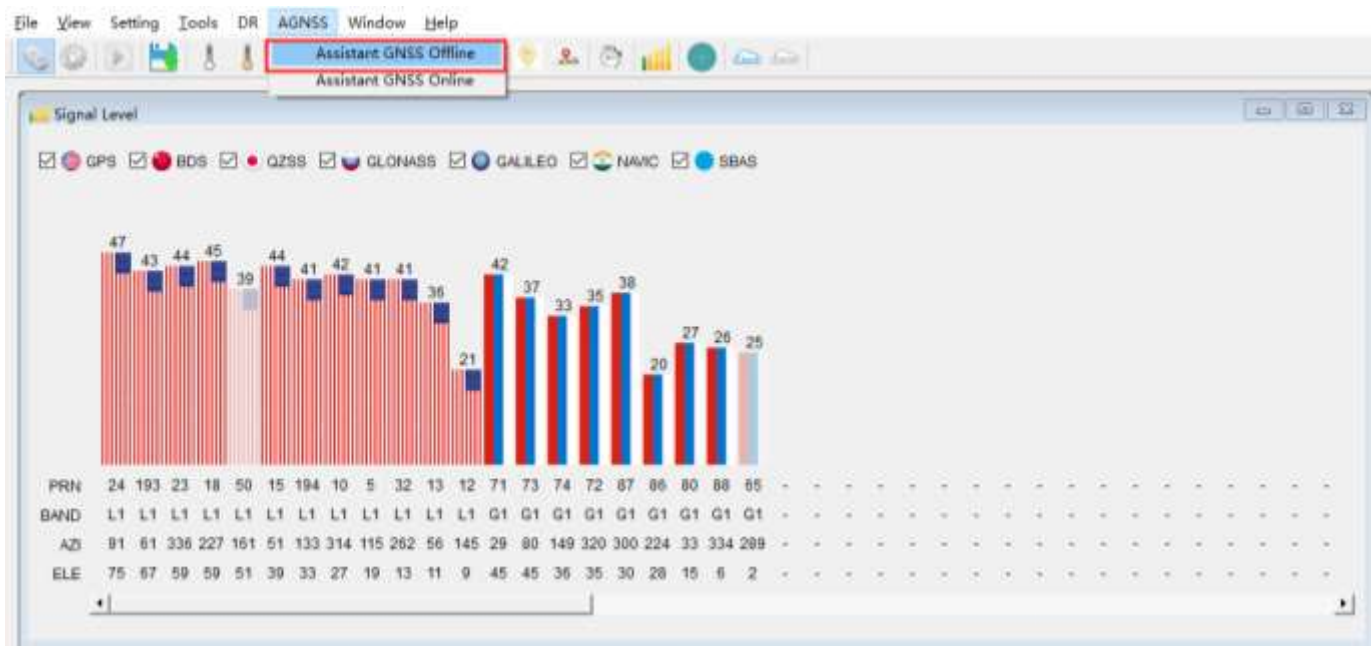


图 11：用于设置 Flash EPO 的 QGNSS 界面

3. 下载 EPO 文件到模块。
  - a) 点击“Connect”按钮连接FTP服务器；
  - b) 选择EPO文件；
  - c) 点击“Download selected file”按钮下载EPO文件到电脑；
  - d) 选择星系类型；



- e) 点击 “...” 按钮选择EPO文件;
- f) 点击 “Download” 按钮下载EPO文件到模块。

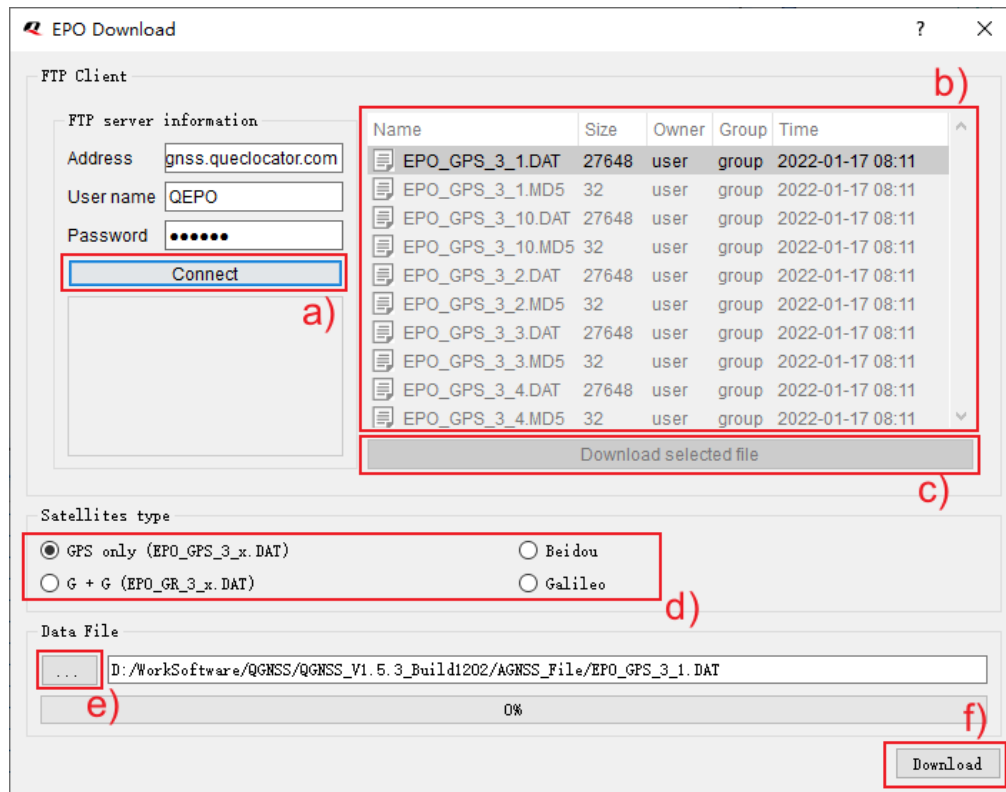


图 12: 下载 Flash EPO 文件

## 5.2. 使用 QGNSS 下载 Host EPO

使用QGNSS工具下载Host EPO的步骤:

1. 运行 QGNSS 工具;
2. 在主界面, 点击 “AGNSS” → “Assistant GNSS Online”, 如下图:

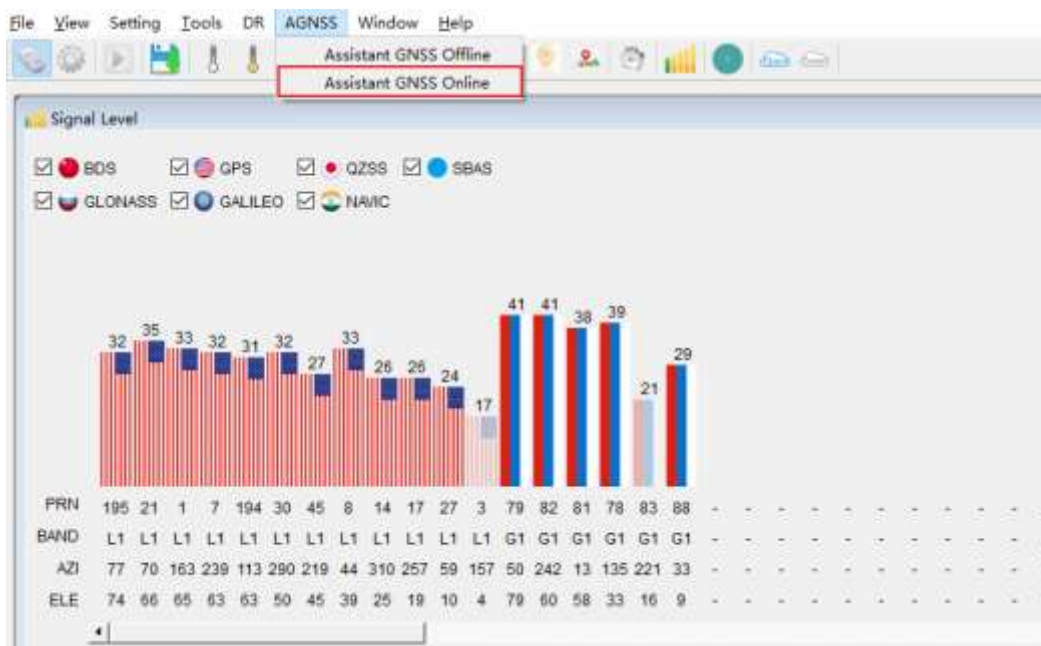


图 13: 用于设置 Host EPO 的 QGNSS 界面

### 3. 配置参数

- 选择“Use Current Position”使用当前位置；
- 选择“Use Current UTC”使用当前 UTC 时间；
- 点击“Transfer”下载 Host EPO 文件至模块。

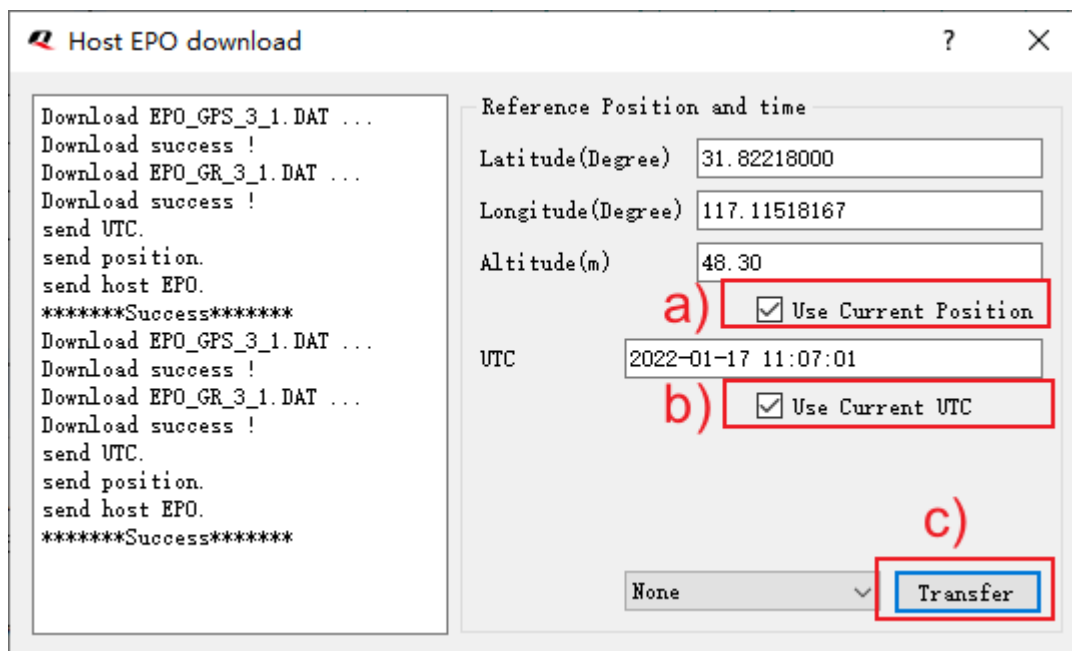


图 14: 下载 Host EPO 文件

# 6 AGNSS 实现示例

本章举例说明如何将 EPO 文件下载到模块。

## 6.1. Flash EPO 实现 AGNSS

蓝色加粗：发送数据

黑色加粗：ACK 信息

//主机发送\$PAIR472\*3B 擦除 Flash 中存储的 EPO 数据:

**\$PAIR472\*3B**

//模块返回\$PAIR001 消息:

**\$PAIR001,472,0\*3A**

//主机以十六进制格式发送 EPO 起始消息:

**04 24 B0 04 01 00 52 E7 AA 44**

//模块返回 ACK 消息:

**04 24 E8 03 04 00 B0 04 00 00 5B AA 44**

//主机以十六进制格式发送 EPO 数据:

**04 24 B1 04 48 00 BF 96 05 41 6E 3A 74 05 C3 21..... AA 44**

//模块返回 ACK 消息:

**04 24 E8 03 04 00 B1 04 00 00 5A AA 44**

//主机以十六进制格式发送 EPO 数据:

**04 24 B1 04 48 00 BF 96 05 42 09 3A 74 0C CA 77..... AA 44**

//模块返回 ACK 消息:

**04 24 E8 03 04 00 B1 04 00 00 5A AA 44**

.....

//主机以十六进制格式发送 EPO 结束数据:

**04 24 B2 04 01 00 52 E5 AA 44**

//模块返回 ACK 消息:

**04 24 E8 03 04 00 B2 04 00 00 59 AA 44**

//主机查询 GNSS 芯片中存储的 EPO 数据状态:

**\$PAIR470,0\*25**

//模块返回\$PAIR001 和\$PAIR470 消息:

**\$PAIR001,470,0\*38**

**\$PAIR470,0,1,2098,194400,2098,216000,2098,194400,2098,216000\*38**

## 6.2. Host EPO 实现 AGNSS

蓝色加粗: 发送数据

黑色加粗: ACK 信息

//主机发送命令\$PAIR002 为 GNSS 系统上电 (详情见[文档 \[1\] 协议规范](#)):

**\$PAIR002\*38**

//模块返回\$PAIR001 消息:

**\$PAIR001,002,0\*39**

//模块自动输出\$PAIR010 消息:

**\$PAIR010,1,-1\*16**

**\$PAIR010,2,-1\*15**

//主机发送参考 UTC 时间命令\$PAIR590:

**\$PAIR590,2023,3,29,9,0,58\*09**

//模块返回\$PAIR001 消息:

**\$PAIR001,590,0\*37**

//主机发送参考位置命令\$PAIR600:

**\$PAIR600,31.822203,117.115219,175.0,50.0,50.0,0.0,100.0\*0F**

//模块返回\$PAIR001 消息:

**\$PAIR001,600,0\*3D**

//主机发送 EPO 数据:

**\$PAIR471,0,1,10596C0,A174051A,1B2EDE67,9F0BB6,17C37A4,1B2EDE22,F85B368E,845FB0C9,6F18C40,23557111,2A4CBD5,A60348AB,FEF7E24,2F236B88,2439FDC6,1000001C,0,4860BF93\*44**

//模块返回\$PAIR001 消息:

**\$PAIR001,471,0\*39**

//主机发送 EPO 数据:

**\$PAIR471,0,2,20596C0,F0740341,1B2EEE36,5F3FAA,17E07E6,1B2E1100,F8C1048F,84A50985,6F1BD33,29192005,D651ED6,A600506D,256C95F,20430E67,C36C910D,1000001C,0,FAC0DA552A 33 43 0D 0A\*3C**

//模块返回\$PAIR001 消息:

**\$PAIR001,471,0\*39**

.....

//主机发送 EPO 数据:

**\$PAIR471,0,7,70596C0,377403C2,1B2E41F1,F757006,C57EB,1B2EBF14,F89F543F,8986E880,6F1E693,DC3908E,DA7BB7,A603A757,8FBA37BF,21C8A8F0,A2247EAD,1000001C,22000000,DDACBBB6\*0B**

//模块返回\$PAIR001 消息:

**\$PAIR001,471,0\*39**

# 7 附录 A 参考文档及术语缩写

表 11: 参考文档

文档名称
[1] Quectel_LC29H 系列&LC79H(AL)_GNSS_协议规范

表 12: 术语和缩写

缩写	英文全称	中文全称
ACK	Acknowledgement	确认消息
AGNSS	Assisted GNSS (Global Navigation Satellite System)	辅助全球导航卫星系统
EPO	Extended Prediction Orbit	扩展轨道预测
GLONASS	Global Navigation Satellite System (Russia)	格洛纳斯导航卫星系统（俄罗斯）
GNSS	Global Navigation Satellite System	全球导航卫星系统
GPS	Global Positioning System	全球定位系统
LSB	Least Significant Bit	最低有效位
MSB	Most Significant Bit	最高有效位
IMEI	International Mobile Equipment Identity	国际移动设备标识
IONO	Ionospheric	电离层
MNL	MTK Navigation Lib	MTK 导航库
RAM	Random Access Memory	随机存取存储器
SYSID	System Identification	系统标识符
SV	Satellite in View	可视卫星

缩写	英文全称	中文全称
SVID	Satellite in View Identification	可视卫星标识符
TOW	Time of Week	周内时间
UART	Universal Asynchronous Receiver/Transmitter	通用异步收发传输器
UTC	Coordinated Universal Time	协调世界时
URL	Uniform Resource Locator	统一资源定位器

## 8 附录 B 特殊符号

表 13：特殊符号

特殊符号	定义
<CR>	回车符
<LF>	换行符
<...>	参数名称（实际命令中不包含尖括号）
[...]	可选参数（实际命令中不包含方括号）
{...}	循环参数（实际命令中不包含大括号）
<u>下划线</u>	参数的默认设置