

# ECx00U&EGx00U 系列

## QuecOpen FOTA 应用指导

**LTE Standard 模块系列**

版本：1.0

日期：2021-11-16

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软硬件、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 2021，保留一切权利。

**Copyright © Quectel Wireless Solutions Co., Ltd. 2021.**

# 文档历史

## 修订记录

版本	日期	作者	变更表述
-	2020-12-02	Fei XUE/ Jensen FANG	文档创建
1.0	2021-11-16	Fei XUE/ Jensen FANG	受控版本

## 目录

文档历史 .....	3
目录 .....	4
表格索引 .....	5
图片索引 .....	6
<b>1 引言 .....</b>	<b>7</b>
1.1. 适用模块 .....	7
<b>2 FOTA 升级流程 .....</b>	<b>8</b>
<b>3 FOTA 升级包 .....</b>	<b>9</b>
3.1. 制作升级包 .....	9
3.1.1. 制作差分升级包 .....	10
3.1.2. 制作整包升级包 .....	11
3.2. 查看升级包信息 .....	11
3.3. 差分升级配置 .....	12
3.3.1. XML 文件配置说明 .....	12
3.3.2. 制作中间升级包 .....	15
<b>4 FOTA API .....</b>	<b>17</b>
4.1. 头文件 .....	17
4.2. API 详解 .....	17
4.2.1. ql_fota_get_pack_name .....	17
4.2.1.1. ql_errcode_fota_e .....	17
4.2.2. ql_fota_image_verify .....	19
4.2.3. ql_fota_image_verify_without_setflag .....	19
4.2.4. ql_fota_file_reset .....	20
4.2.5. ql_fota_get_result .....	20
4.2.5.1. ql_fota_result_e .....	20
4.2.6. ql_power_reset .....	21
4.2.6.1. ql_ResetMode .....	21
<b>5 示例 .....</b>	<b>22</b>
5.1. 代码演示 .....	22
5.2. API 调用流程 .....	22
5.3. 测试验证 .....	22
5.4. FOTA 升级异常日志分析 .....	24
<b>6 附录 参考文档及术语缩写 .....</b>	<b>27</b>

## 表格索引

表 1: 适用模块 .....	7
表 2: 参考文档 .....	27
表 3: 术语缩写 .....	27

## 图片索引

图 1：升级包制作工具目录.....	9
图 2：升级包制作工具帮助信息 .....	10
图 3：差分升级包制作执行日志 .....	10
图 4：整包升级包制作执行日志 .....	11
图 5：下载升级进度和状态的日志-A .....	23
图 6：下载升级进度和状态的日志-B .....	24

# 1 引言

移远通信 LTE Standard ECx00U 系列和 EGx00U 模块支持 QuecOpen®方案；QuecOpen®是基于 RTOS 的嵌入式开发平台，可简化 IoT 应用的软件设计和开发过程。有关 QuecOpen®的详细信息，请参考文档 [1]。

本文档主要介绍在 QuecOpen®方案下，ECx00U 系列和 EGx00U 模块的 FOTA 升级功能，包括 FOTA 升级流程、升级包制作、FOTA 相关 API 以及示例。

## 1.1. 适用模块

表 1：适用模块

模块系列	模块
ECx00U	EC200U 系列
	EC600U 系列
EGx00U	EG500U-CN
	EG700U-CN



## 2 FOTA 升级流程

在 QuecOpen 方案下，EC200U 系列、EC600U 系列、EG500U-CN 和 EG700U-CN 模块支持差分升级和整包升级，可仅升级内核、仅升级用户应用程序，或者同时升级内核和用户应用程序，不可升级 Bootloader。具体升级流程如下：

1. 制作升级包并上传至服务器。升级包制作详情请参考第 3 章。
2. 模块根据服务器链接地址下载升级包并写入到模块的文件系统中，下载方式支持 HTTP、FTP 等。
3. 校验升级包，判断当前升级包文件的有效性。
4. 若升级包校验成功，需重启模块，系统在模块重启之后自动进行 FOTA 升级。
5. 升级完成后可以查询升级结果，也可删除升级包文件以释放文件系统空间。

### 备注

1. 升级包下载完成后，一般存放在模块的文件系统中，因此升级包大小不能超过文件系统可用空间大小且需预留一定空间。升级包也可以存放在模块外接专用 SPI Flash 或者 SD 卡中。
2. 当升级包存储在内置 Flash 或外置 Flash 时，存储路径支持设置子目录；当存储在 SD 卡时，存储路径只支持设置根目录。
3. FOTA 升级时，Modem 分区剩余空间需不小于 64 KB，UFS 分区剩余空间需不小于 16 KB。

## 3 FOTA 升级包

### 3.1. 制作升级包

本文将模块当前运行的版本称为“旧版本”，目标版本称为“新版本”，制作升级包时使用的固件包称为“基础包”。

QuecOpen SDK 中提供 FOTA 升级包制作工具 `dtools`，位于 `tools/win32` 目录下，如下图所示：

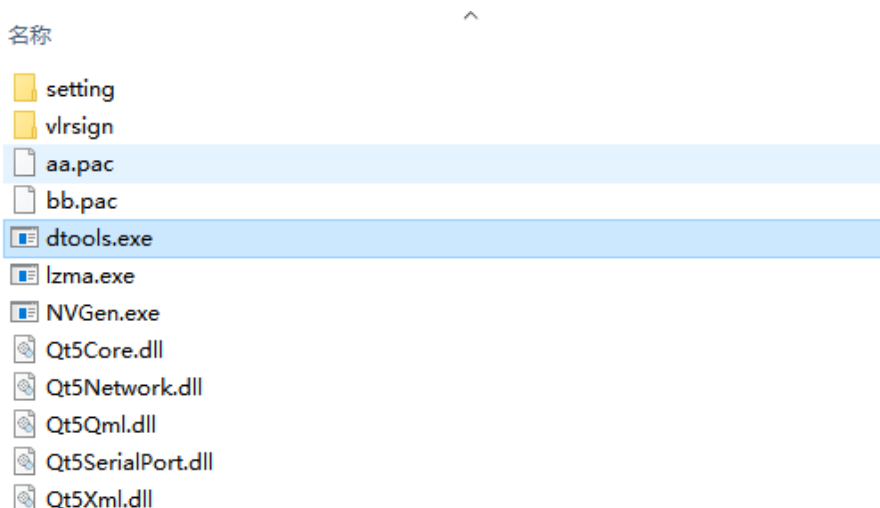


图 1：升级包制作工具目录

升级包制作工具 `dtools` 的主要文件如下：

- 旧版本的固件包（固件包名称可自定义），如 `aa.pac`；
- 新版本的固件包（固件包名称可自定义），如 `bb.pac`；
- 升级工具生成的 FOTA 升级包（固件包名称可自定义），如 `output.pack`；
- 用于制作 FOTA 升级包的程序：`dtools.exe`；
- 用于保存制作 FOTA 升级包的配置信息：文件夹 `setting`。

可通过在 Windows 命令窗口执行 `dtools.exe fotacreate2 [-? /-h /-help]` 查询 `dtools` 工具的使用方法，如下图。

```
PS > .\tools\win32> .\dtools.exe fotacreate2
Usage: .\tools\win32\dtools.exe [options] fotacreate2 output
create fota pack, format version 2

Options:
  -?, -h, --help           Displays this help.
  -v, --version            Displays version information.
  -d, --debug <debug>     debug level, e/w/i/d/v
  --oldversion <oldver>    old version name, information to be
                           embedded into fota
  --newversion <newver>    new version name, information to be
                           embedded into fota
  --force                  create diff for same content, for debug
                           only
  --single-pac <pac.xml>   single pac, without reference
  --pac <old,new.xml>      pac diff
  --lod <old,new,block_size,bundle> lod diff

Arguments:
  fotacreate2
  output                  output image file
```

图 2：升级包制作工具帮助信息

### 3.1.1. 制作差分升级包

制作差分升级包时，需准备旧版本固件包和新版本固件包。本章节以旧版本固件包 *aa.pac* 和新版本固件包 *bb.pac* 为例，具体步骤如下：

- 步骤1：** 将 *aa.pac*、*bb.pac* 固件包拷贝至 SDK 的 *tools/win32* 目录下。
- 步骤2：** 打开 *dtools.exe* 程序。
- 步骤3：** 使用 *dtools* 的 *fotacreate2* 作为升级包生成工具。进入 *tools/win32* 目录，打开 Windows 命令窗口，执行 ***dtools.exe fotacreate2 --pac aa.pac,bb.pac,setting\fota8910.xml output.pack -d v*** 开始制作差分升级包。

```
PS > .\tools\win32> .\dtools.exe fotacreate2 --pac aa.pac,bb.pac,setting\fota8910.xml output.pack -d v
patch AP at 0x60010000: 1955372 -> 2134816
patch PS at 0x604a0000: 3407872 -> 3407872
phys_start = 0x4a0000
phys_size = 0x340000
eb_size = 0x10000
pb_size = 0x200
geom = 0x37
used LB count: 6397
block_size = 500
block_count = 6527
tiny file max = 404
small file max = 101000
large file max = 25048000
file "bcpu_gsm.bin.000" size 12016
file "cp.bin.000" size 40360
file "cp.bin.001" size 41792
file "cp.bin.002" size 43904
file "cp.bin.003" size 43320
file "cp.bin.004" size 44296
file "cp.bin.005" size 43224
file "cp.bin.006" size 43696
file "cp.bin.007" size 43944
file "cp.bin.008" size 45288
file "cp.bin.009" size 42904
```

图 3：差分升级包制作执行日志

#### 备注

1. 制作差分升级包时使用的包需与旧版本固件包保持一致。
2. 若在 Windows 命令窗口执行命令制作升级包时未输入 ***-d v***，终端窗口将不会有日志输出。

### 3.1.2. 制作整包升级包

制作整包升级包时，仅需准备新版本固件包。本章以新版本固件包 *bb.pac* 为例，步骤如下：

**步骤1：** 将 *bb.pac* 固件包拷贝 SDK 中 *tools/win32* 目录下。

**步骤2：** 打开 *dtools.exe* 程序。

**步骤3：** 使用 *dtools* 的 *fotacreate2* 作为升级包生成工具。进入 *tools/win32* 目录下，打开 Windows 命令窗口，执行 ***dtools fotacreate2 --single-pac bb.pac,setting\fota8910.xml out.pac -d v*** 开始制作整包升级包。

```
PS D:\tools\win32> .\dtools fotacreate2 --single-pac bb.pac,setting\fota8910.xml out.pac -d v
patch AP at 0x60010000: 2134816
patch PS at 0x604a0000: 3407872
phys_start = 0x4a0000
phys_size = 0x340000
eb_size = 0x10000
pb_size = 0x200
geom = 0x37
used LB count: 6397
block_size = 500
block_count = 6527
tiny file max = 404
small file max = 101000
large file max = 25048000
file "bcpu_gsm.bin.000" size 12016
file "cp.bin.000" size 40360
file "cp.bin.001" size 41792
file "cp.bin.002" size 43904
file "cp.bin.003" size 43320
file "cp.bin.004" size 44296
file "cp.bin.005" size 43224
file "cp.bin.006" size 43696
file "cp.bin.007" size 43944
file "cp.bin.008" size 45288
```

图 4：整包升级包制作执行日志

#### 备注

若在 Windows 命令窗口执行命令制作升级包时未输入 **-d v**，终端窗口将不会有日志输出。

### 3.2. 查看升级包信息

FOTA 升级包制作完成后，可通过 *dtools* 查询工具查看 FOTA 升级包信息。在 Windows 命令窗口执行 ***dtools.exe fotainfo[-? /-h /-help]*** 查询 *dtools* 工具的使用方法，如下图所示：

```
PS D:\Quectel_Doc\Documents\Quectel_tools\uinsoc_fota_tools\win32> .\dtools.exe fotainfo
Usage: D:\Quectel_Doc\Documents\Quectel_tools\uinsoc_fota_tools\win32\dtools.exe [options] fotainfo pack

show fota pack information

Options:
  -?, -h, --help      Displays this help.
  -v, --version        Displays version information.
  -d, --debug <debug> debug level, e/w/i/d/v

Arguments:
  fotainfo
  pack                fota pack file name
PS D:\Quectel_Doc\Documents\Quectel_tools\uinsoc_fota_tools\win32>
```

以升级包 *fota.pac* 为例，查看步骤如下：

**步骤 1：**将制作完成的升级包 *fota.pac* 拷贝至 *tools/win32* 目录下。

**步骤 2：**打开 *dtools.exe* 程序。

**步骤 3：**使用 *dtools* 的 **fotainfo** 查看升级包的信息；进入 *tools/win32* 目录下，打开 Windows 命令窗口，执行 **.\dtools.exe fotainfo .\fota.pac** 开始查看升级包信息。升级包信息具体如下图所示。

```

fotainfo
pack                fota pack file name
PS D:\Quectel_Doc\Documents\Quectel_tools\winsoc_fota_tools\win32> .\dtools.exe fotainfo .\fota.pac
FOTA 02 FOTA升级包制作工具的版本号
headerSize = 568 FOTA升级包头部大小
tmpFileSize = 65536
bulkCount = 12 待升级部分个数
blockCount = 128
verNameSize = 0
oldVerName = ""
newVerName = ""
bulk #0, type 0 (FLASH patch) 待升级部分编号以及操作类型
  flashName = "SFL1"
  base = 0x10000
  oldSize = 2240352
  oldCrc = 0x179de11c
  bundleCount = 1
  dataSize = 403238
bulk #1, type 0 (FLASH patch)
  flashName = "SFL1"
  base = 0x250000
  oldSize = 5632
  oldCrc = 0xf8e97ef1
  bundleCount = 1
  dataSize = 649
bulk #2, type 2 (plain file patch)
  fname = "/modem/nvm/ims_delta_nv.bin" 需要升级的预置文件名称
  oldCrc = 0xa963319c
  oldSize = 5246
  newSize = 3361
  dataSize = 419 当前待升级部分文件大小
bulk #3, type 2 (plain file patch)
  fname = "/modem/nvm/rf_nv.bin"
  oldCrc = 0x46f1198c
  
```

### 3.3. 差分升级配置

#### 3.3.1. XML 文件配置说明

位于 *tools/win32/setting* 目录下的 *fota8910.xml* 文件可针对模块的 AP 侧或 PS 域、不同的 NV 项、预置文件以及 App 侧进行差分升级配置。如无需修改，使用默认设置即可；如需修改可按照该文件中的注释信息进行修改。该文件可按照四个部分进行差分升级配置，具体如下：

## 1. Pacflash 对以 Flash 形式下载的数据做差分升级

配置属性解释如下：

<i>id</i>	文件 ID
<i>flash</i>	指定内置 flash（SFL1）或外置 flash（SFL2）
<i>blocksize</i>	默认为 0x10000
<i>bundleblock</i>	默认为 0
<i>method</i>	指定差分升级方式。 <i>diff</i> : 进行差分（默认值） <i>ignore</i> : 忽略 <i>remove</i> : 移除 <i>replace</i> : 代替

举例：

```
<pacflash id="AP" flash="SFL1" blocksize="0x10000" bundleblock="0"method="diff"/>
```

## 2. Pacsffs 对以文件系统形式下载的数据做差分升级

可对挂载在 Modem 分区的 *nvm* 目录下 *nv.bin* 文件中的 NV 项进行差分升级，也可以指定挂载分区。

配置属性解释如下：

<i>id</i>	文件 ID
<i>ebsize</i>	<i>partinfo_xx.json</i> 中指定的 <i>erase_block</i> 大小
<i>pbsize</i>	<i>partinfo_xx.json</i> 中指定的 <i>logic_block</i> 大小
<i>mount</i>	<i>partinfo_xx.json</i> 中指定的挂载分区
<i>method</i>	指定差分升级方式。 <i>diff</i> : 进行差分（默认值） <i>ignore</i> : 忽略 <i>remove</i> : 移除 <i>replace</i> : 代替

举例：

```
<pacsffs id="PS" ebsize="0x10000" pbsize="0x200" mount="/modem"method="diff"/><file  
name="nvm/audio_calib.bin" method="ignore"/>
```

### 3. Pacnv 对 Running NV 文件做差分升级

决定是否删除 Running NV 文件以及是否恢复 Fix NV 文件。

配置属性解释如下：

<i>nvitem id</i>	NV 项 ID，在 <i>nvm_8910.c</i> 中可找到 ID 与 NV 文件的对应关系
<i>running nv</i>	Running NV 的路径和文件名
<i>cleanrunning onchange</i>	清除 Running NV 操作；具体参数如下： <i>always</i> : 无论什么情况下，FOTA 升级时总是清除此 Running NV <i>never</i> : 无论什么情况下，FOTA 升级均不会清除此 Running NV <i>any</i> : 若某一 Running NV 对应的 Fix NV 有修改，删除此 Running NV

举例：

```
<nvitem id="0x1ba" runningnv="/modemnvm/bt_sprd.bin">
    <cleanrunning onchange="0x1ba"/>
</nvitem>
```

如上所述，“0x1ba”对应的 Fix NV 修改时（升级前后两个版本的"/modem/nvm/bt\_sprd.bin"不同），会将"/modemnvm/bt\_sprd.bin"删除。

### 备注

Fix NV 是原始 NV 文件，在模块运行时不允许修改该文件。Running NV 文件是 Fix NV 文件的备份文件，可以被修改。升级包制作工具通过对比 Fix NV 文件制作 FOTA 升级包。Fix NV 文件存放在 Modem 分区的 *nvm* 目录下，Running NV 文件存放在 */modemnvm* 目录下。可从 *nvm\_8910.c* 文件中获取 NV 项 ID 信息。

### 4. Paccpio 对预置文件做差分升级

配置属性解释如下：

<i>id</i>	文件 ID（模块的预置文件 ID 是 PREPACK）
<i>method</i>	指定差分升级方式。 <i>diff</i> : 进行差分（默认值） <i>ignore</i> : 忽略 <i>replace</i> : 代替

举例：

```
<paccpio id="PREPACK" method="diff"> </paccpio>
```

### 3.3.2. 制作中间升级包

FOTA 升级时，若制作的升级包过大，将可能出现 FOTA 校验失败，进而导致 FOTA 升级失败或模块变砖。为了避免 UFS 和 Modem 分区剩余空间不足的情况，模块支持通过如下两种方法进行 FOTA 分步升级：

#### 方法 1：增加中间版本，制作中间升级包

在旧版本包（以 *aa.pac* 为例）与新版本包（以 *bb.pac* 为例）之间，选择一个中间版本（以 *cc.pac* 为例）。首先制作 *aa.pac* 与 *cc.pac* 之间的差分包，将模块固件升级到中间版本。然后再制作 *cc.pac* 与 *bb.pac* 之间的差分包，升级到目标版本，进而完成 FOTA 升级。

#### 方法 2：修改预置文件的升级方式，制作中间升级包

1. 制作升级包时，忽略预置文件以减少升级包的大小。在 XML 文件中，增加 *ignore* 的方式对预置文件进行差分升级，详情请参考第 3.3.1 章。示例如下：

```
</pacnv>
<paccpio id="PREPACK" method="diff">
  <file name="user/firm" method="ignore"/>
  <file name="user/boot" method="ignore"/>
</paccpio>
</pacdiff>
```

修改 XML 文件后，制作中间升级包，下载并进行 FOTA 升级，将模块固件升级到中间版本。

#### 备注

修改 XML 文件前，可通过 **fotainfo** 工具查看旧版本固件包的信息，从而确定需要升级的预置文件。

2. 升级到中间版本后，再基于中间版本与目标版本制作第二个升级包。首先修改 XML 文件，使用 *replace* 的方式对之前忽略的预置文件进行差分升级，然后制作升级包，下载完成后进行 FOTA 升级，从而将模块固件升级至目标版本。

```
</pacnv>
<paccpio id="PREPACK" method="diff">
  <file name="user/firm" method="replace"/>
  <file name="user/boot" method="replace"/>
</paccpio>
```



## 备注

1. 若 UFS 分区剩余空间不足，可使用上述任一方式进行分步升级。
2. Modem 分区文件升级不可分开操作。因此，若 Modem 分区空间不足，建议使用方法一。
3. UFS 分区剩余空间大小可通过调用 `int64 ql_fs_free_size(const char *path)` 获取。Modem 分区剩余空间大小可以通过 FOTA 相关日志获取，详情参考第 5.4 章。

# 4 FOTA API

## 4.1. 头文件

FOTA API 的头文件为 `ql_api_fota.h`，位于 `components\ql-kernel\inc` 目录下。若无特别说明，本文档所提到的接口以及宏定义均在该目录下。

## 4.2. API 详解

### 4.2.1. ql\_fota\_get\_pack\_name

该函数用于获取升级包的路径信息。

- 函数原型

```
ql_errcode_fota_e ql_fota_get_pack_name(char *p_pac_file_name,int* length);
```

- 参数

*p\_pac\_file\_name:*

[In] 升级包路径信息的存储地址。

*length:*

[In] 升级包路径信息存储地址的长度。

- 返回值

详情请参考第 4.2.1.1 章。

#### 4.2.1.1. ql\_errcode\_fota\_e

FOTA 错误码由相关的组件 ID 与标准错误码共同组成，其中组件 ID 为高 16 位，标准错误码为低 16 位。FOTA 的组件 ID 为 0x8C00；QL\_FOTA\_ERRCODE\_BASE = 0x8C00<<16。

FOTA 错误码枚举信息定义如下：

```
typedef enum
{
    QL_FOTA_SUCCESS                = 0,
    QL_FOTA_EXECUTE_ERR           = 502|QL_FOTA_ERRCODE_BASE,
    QL_FOTA_IMAGE_VERIFY_READY_ERR = 510|QL_FOTA_ERRCODE_BASE
    QL_FOTA_IMAGE_VERIFY_ERR      = 547|QL_FOTA_ERRCODE_BASE

    QL_FOTA_FLAG_SET_ERR          = 548|QL_FOTA_ERRCODE_BASE
    QL_FOTA_BUSY_ERR             = 549|QL_FOTA_ERRCODE_BASE
    QL_FOTA_POINT_NULL_ERR        = 550|QL_FOTA_ERRCODE_BASE
    QL_FOTA_PARAM_INVALID         = 551|QL_FOTA_ERRCODE_BASE
    QL_FOTA_PACKPATH_INVALID      = 552|QL_FOTA_ERRCODE_BASE
    QL_FOTA_ADAPTION_RESET_ERR    = 553|QL_FOTA_ERRCODE_BASE
    QL_FOTA_ADAPTION_SAVE_ERR     = 554|QL_FOTA_ERRCODE_BASE
    QL_FOTA_ADAPTION_DELETE_ERR   = 555|QL_FOTA_ERRCODE_BASE
    QL_FOTA_ADAPTION_LOAD_ERR     = 556|QL_FOTA_ERRCODE_BASE
    QL_FOTA_ADAPTION_LENGTH_ERR   = 557|QL_FOTA_ERRCODE_BASE
} ql_errcode_fota_e;
```

● 成员

成员	描述
QL_FOTA_SUCCESS	执行成功
QL_FOTA_EXECUTE_ERR	执行异常
QL_FOTA_IMAGE_VERIFY_READY_ERR	FOTA 升级包校验并设置升级就绪标志位操作失败
QL_FOTA_IMAGE_VERIFY_ERR	FOTA 升级包校验操作失败
QL_FOTA_FLAG_SET_ERR	FOTA 升级标志设置失败
QL_FOTA_BUSY_ERR	FOTA 繁忙
QL_FOTA_POINT_NULL_ERR	空指针异常
QL_FOTA_PARAM_INVALID	无效参数
QL_FOTA_PACKPATH_INVALID	输入的升级包存储路径无效
QL_FOTA_ADAPTION_RESET_ERR	重置升级包路径信息为默认值失败
QL_FOTA_ADAPTION_SAVE_ERR	保存升级包路径信息的配置文件失败
QL_FOTA_ADAPTION_DELETE_ERR	删除升级包路径信息的配置文件失败
QL_FOTA_ADAPTION_LOAD_ERR	读取升级包路径信息的配置文件失败

QL\_FOTA\_ADAPTION\_LENGTH\_ERR

升级包存储路径的缓存大小错误

### 4.2.2. ql\_fota\_image\_verify

该函数用于校验模块文件系统中存储的升级包信息并设置 FOTA 升级就绪标志位。若升级包有效，函数执行成功并将 FOTA 升级标志位设置为 QL\_FOTA\_READY 状态，然后调用 *ql\_power\_reset()* 重启模块后开始升级固件；若升级包无效，则返回错误码。

- 函数原型

```
ql_errcode_fota_e ql_fota_image_verify(char* PackFileName);
```

- 参数

*PackFileName*:

[In] 升级包在模块文件系统中的存储路径。默认值：UFS:fota.pac。

- 返回值

详情参考第 4.2.1.1 章。

### 4.2.3. ql\_fota\_image\_verify\_without\_setflag

该函数用于校验模块文件系统中存储的升级包信息。若升级包有效，函数执行成功；若升级包无效，返回错误码。

- 函数原型

```
ql_errcode_fota_e ql_fota_image_verify_without_setflag(char* PackFileName);
```

- 参数

*PackFileName*:

[In] 升级包在模块文件系统中的存储路径。默认值：UFS:fota.pac。

- 返回值

详情参考第 4.2.1.1 章。

### 备注

该函数仅用于校验升级包，并不会设置 FOTA 升级就绪标志位。如需校验成功后立即重启模块进行 FOTA 升级，则建议直接调用 *ql\_fota\_image\_verify()*。

#### 4.2.4. ql\_fota\_file\_reset

该函数用于删除 FOTA 升级包。

- 函数原型

```
ql_errcode_fota_e ql_fota_file_reset(bool del_image);
```

- 参数

*del\_image*:

[In] 设置为 *TRUE* 时则删除升级包；设置为 *FALSE* 时则不删除升级包。

- 返回值

详情参考第 4.2.1.1 章。

#### 4.2.5. ql\_fota\_get\_result

该函数用于获取 FOTA 升级结果。

- 函数原型

```
ql_errcode_fota_e ql_fota_get_result(ql_fota_result_e *p_fota_result);
```

- 参数

*p\_fota\_result*:

[In] FOTA 升级结果的指针。详情请参考第 4.2.5.1 章。

- 返回值

详情参考第 4.2.1.1 章。

##### 4.2.5.1. ql\_fota\_result\_e

FOTA 升级结果枚举定义如下：

```
typedef enum
{
    QL_FOTA_FINISHED = 0,
    QL_FOTA_NOT_EXIST,
    QL_FOTA_READY,
    QL_FOTA_STATUS_INVALID,
} ql_fota_result_e
```

- 成员

成员	描述
<code>QL_FOTA_FINISHED</code>	升级完成
<code>QL_FOTA_NOT_EXIST</code>	未检测到最新的 FOTA 升级包
<code>QL_FOTA_READY</code>	检测到最新的 FOTA 升级包，等待系统升级
<code>QL_FOTA_STATUS_INVALID</code>	无效状态

#### 4.2.6. ql\_power\_reset

该函数用于重启模块。FOTA 升级包校验通过后，需主动调用该函数重启模块。重启后模块自动进入升级模式，开始自动升级固件。该函数的详细信息可参考文档 [2]。

- 函数原型

```
ql_errcode_power ql_power_reset(ql_ResetMode reset_mode);
```

- 参数

*reset\_mode:*

[In] 重启方式。请参考第 4.2.6.1 章。

- 返回值

详情请参考文档 [2]。

##### 4.2.6.1. ql\_ResetMode

重启方式枚举定义如下：

```
typedef enum
{
    RESET_QUICK,
    RESET_NORMAL
}ql_ResetMode;
```

- 成员

成员	描述
<code>RESET_QUICK</code>	立即重启
<code>RESET_NORMAL</code>	正常重启

# 5 示例

## 5.1. 代码演示

QuecOpen SDK 中提供示例代码文件 `components\ql-application\http_fota\inc\fota_http_demo.h` 和 `components\ql-application\http_fota\fota_http_demo.c`，分别为 FOTA 升级包下载和升级过程示例。

如需使用示例代码，仅需将代码中的升级包路径替换为 `dtools` 工具生成的升级包所在的远程服务器链接地址。

## 5.2. API 调用流程

- 步骤1：** 下载 FOTA 升级包至模块的文件系统并保存。
- 步骤2：** 调用 `ql_fota_image_verify()`校验升级包，校验成功后设置 FOTA 升级就绪标志位；若调用该函数时参数为空，系统则使用默认路径：UFS:fota.pac。
- 步骤3：** 调用 `ql_power_reset()`重启模块。重启后模块将在 Bootloader 中自动进行 FOTA 升级。
- 步骤4：** FOTA 升级完成后，调用 `ql_fota_get_result()`获取 FOTA 升级结果。
- 步骤5：** FOTA 升级成功后，调用 `ql_fota_file_reset()`删除 FOTA 升级包。

## 5.3. 测试验证

以下测试验证主要针对差分升级。使用旧版本固件包 `aa.pac` 和新版本固件包 `bb.pac` 制作 FOTA 差分包 `output.pack`，并存放至 HTTP 服务器中，固件包名称可自定义。

1. 将旧版本固件包 `aa.pac` 烧录至模块中，烧录完成后可通过 `ql_dev_get_firmware_version()`获取当前版本信息。无论烧录的旧版本固件包是否为 `aa.pac`，FOTA 测试例程都会下载 HTTP 服务器中的差分包 `output.pack`。若烧录的旧版本固件包不是 `aa.pac`，在 FOTA 升级包校验时会返回错误，导致 FOTA 升级失败。
2. 下载差分包 `output.pack`；若下载正常并校验成功，模块会自动重启并自动进行 FOTA 升级。
3. 升级成功后，FOTA 测试例程判断 FOTA 升级是否完成；若完成则自动删除 FOTA 相关的文件。

FOTA 升级示例运行的日志分析如下：

1. 图中红框 1，进入 `http_fota_demo`。如需使用 SD，需对 SD 卡进行挂载。挂载详情请参考文档 [3]。
2. 图中红框 2，表明未检测到 FOTA 升级包。
3. 图中红框 3，显示当前模块的固件版本信息。
4. 图中红框 4，需要下载的 FOTA 升级包的长度。
5. 图中红框 5，开始下载 FOTA 升级包。
6. 图中红框 6，表明完成下载以及下载的 FOTA 升级包长度。
7. 图中红框 7，当前 FOTA 升级包的存储路径。
8. 图中红框 8，升级包下载完成后 UFS 和 Modem 分区剩余空间大小。
9. 图中红框 9，升级包校验信息日志，可与 `fotainfo` 工具获取的固件包信息进行对比。
10. 图中红框 10，制作差分包时使用的基础包与旧版本固件包不一致，FOTA 升级包检验失败。
11. 图中红框 11，FOTA 升级包校验并设置升级就绪标志位操作失败。

Tick	Level	Description
63770	QOPN/I	[ql_fota_http][ql_fota_http_app_init, 784] http fota demo support!
16013	QOPN/I	[ql_fota_http][fota_http_app_thread, 728] ql sdmmc mount : 0
16018	QOPN/I	[ql_fota_api][ql_fota_get_result, 418] read PackFileName cfg failed!
16021	QOPN/I	[ql_fota_api][ql_fota_get_result, 426] fup_status 0
16021	QOPN/I	[ql_fota_http][fota_http_result_process, 715] fota file not exist
16022	QOPN/I	[ql_fota_http][fota_http_app_thread, 740] current version: EC200UCNABR02A01M08
16069	QOPN/I	[ql_fota_http][fota_http_init, 428] init file name:[UFS:fota.pack]
16079	QOPN/I	[ql_fota_http][fota_http_init, 429] init file size:[0]
16079	QOPN/I	[ql_fota_http][fota_http_init, 430] init file stage:[0]
16080	QOPN/I	[ql_fota_http][fota_http_app_thread, 752] start [1] times download fota package
16080	QOPN/I	[ql_fota_http][fota_http_info_cfg, 101] init file name:[UFS:fota.pack]
16080	QOPN/I	[ql_fota_http][fota_http_info_cfg, 102] init file stage:[0]
16080	QOPN/I	[ql_fota_http][fota_http_info_cfg, 103] init file download:[0]
16080	QOPN/I	[ql_fota_http][fota_http_info_cfg, 104] init file file_size:[0]
16085	QOPN/I	[ql_fota_http][fota_http_info_cfg, 105] init file real file_size:[0]
16085	QOPN/I	[ql_fota_http][fota_http_info_cfg, 106] init file is_show:[1]
16085	QOPN/I	[ql_fota_http][fota_http_info_cfg, 107] init file last_percent:[0]
16085	QOPN/I	[ql_fota_http][fota_http_info_cfg, 108] init file space:[1]
16090	QOPN/I	[ql_fota_http][fota_http_get_fd, 529] over write file [UFS:fota.pack]
35893	QOPN/I	[ql_fota_http][fota_http_event_cb, 160] response code:200 chunk_encode 0
36122	QOPN/I	[ql_fota_http][fota_dload_file_clran, 90] clran write file [UFS:fota.pack] open fd 1027
36123	QOPN/I	[ql_fota_http][fota_http_event_cb, 183] content_length:[406846] totalsize=[406846]
36439	QOPN/I	[ql_fota_http][fota_http_write_file, 250] write [2535] size
36555	QOPN/I	[ql_fota_http][fota_http_write_file, 252] write ret=[2535]
40699	QOPN/I	[ql_fota_http][fota_http_write_file, 250] write [5120] size
47165	QOPN/I	[ql_fota_http][fota_http_write_file, 252] write ret=[5120]
47168	QOPN/I	[ql_fota_http][fota_http_write_file, 266] dload progress:[1%]==total size[406846] file_size[7655] dload size[7655]
47580	QOPN/I	[ql_fota_http][fota_http_write_file, 250] write [5120] size

图 5：下载升级进度和状态的日志-A



fota FUPD	
Level	Description
BOOT/I	FUPDATE: 0
VFSN/T	ram heap start/80951380 size/5024320 pool/80951380
Level	Description
QOPN/I	[ql_FOTA_http][fota_http_write_file, 252] write ret=[1111]
QOPN/I	[ql_FOTA_http][fota_http_write_file, 266] dload progress:===[100%]===total size[406846] file_size[406846] dload size[406846]
QOPN/I	[ql_FOTA_http][fota_http_event_cb, 216] ==http transfer end!!!!
QOPN/I	[ql_FOTA_http][fota_http_evt_request, 635] fota http dload size 406846====End,
QOPN/I	[ql_FOTA_http][fota_http_info_cfg, 101] init file name:[UFS:fota.pack]
QOPN/I	[ql_FOTA_http][fota_http_info_cfg, 102] init file stage:[3]
QOPN/I	[ql_FOTA_http][fota_http_info_cfg, 103] init file download:[406846]
QOPN/I	[ql_FOTA_http][fota_http_info_cfg, 104] init file file_size:[406846]
QOPN/I	[ql_FOTA_http][fota_http_info_cfg, 105] init file real file_size:[406846]
QOPN/I	[ql_FOTA_http][fota_http_info_cfg, 106] init file is_show:[1]
QOPN/I	[ql_FOTA_http][fota_http_info_cfg, 107] init file last_percent:[100]
QOPN/I	[ql_FOTA_http][fota_http_info_cfg, 108] init file space:[1]
QOPN/I	[ql_fota_api][ql_fs_convert_fota_pack_name, 119] fota pack name UFS:fota.pack
QOPN/I	[ql_fota_api][ql_fota_fs_size_judge, 245] free size: ufs=763356/0 modem=16384/0
FUPD/D	pack file size/406846, size/406846, header size/432
FUPD/D	file system block size/500, avail/1795, needed/65536
FUPD/D	bulk 0, type 0
FUPD/D	old flash crc mismatch
QOPN/I	[ql_fota_api][ql_fota_image_verify, 282] FOTA ready failed!
QOPN/I	[ql_FOTA_http][fota_http_download_pacfile, 675] [UFS:fota.pack]package is invalid
QOPN/I	[ql_FOTA_http][fota_http_app_thread, 752] start [2] times download fota package
QOPN/T	[ql_FOTA_http][fota_http_info_cfg, 101] init file name:[UFS:fota.pack]

图 6：下载升级进度和状态的日志-B

## 5.4. FOTA 升级异常日志分析

FOTA 升级包校验成功后手动重启或者调用 API 重启模块，开始进行 FOTA 升级。若升级包校验失败或 FOTA 升级失败，模块将无法启动，并一直处于启动失败循环状态。此时可使用 coolwatcher 工具抓取串口日志，分析 FOTA 升级失败的原因。常见的几种失败原因以及解决方法如下。

### 1. 系统剩余空间不足

异常日志如下所示：

```
FUPD/E : flash new data backup write failed
FUPD/E : plain file new data backup write failed
FUPD/E : plain file write new data failed
FUPD/E : blocked file new data backup write failed
```

分析：

上述日志表明写入文件失败，通常的原因是系统剩余空间不足。

解决方法：

制作中间升级包，进行 FOTA 分步升级，详情请参考第 3.3.2 章。

## 备注

预置文件支持通过 FOTA 进行升级，系统默认支持的最大预置文件大小是 64 KB。

### 2. 文件信息不一致

异常日志如下所示：

```
FUPD/E : plain file read old data size crc mismatch
```

分析：

上述日志表明模块当前运行的固件中待升级的文件和差分包中的文件信息不一致。例如，制作差分包使用的基础包中有 `/prepack/example.txt` 文件，而模块当前运行的固件版本中，该文件内容被修改。此时，升级校验过程中会发现文件信息不一致，无法进行差分升级。

解决方法：

通过忽略或者替换方式升级信息不一致的文件。在 `fota8910.xml` 中将此文件的 `method` 设置为 `ignore` 或者 `replace`。示例如下：

```
<pacdiff>
<paccpio id="PREPACK" method="replace">
<file name="prepack/example.txt" method="replace"/>
</paccpio>
</pacdiff>
```

### 3. Modem 分区剩余空间不足

异常日志如下所示：

```
FUPD/E : blocked file write new data failed
```

分析：

上述日志表明 Modem 分区剩余空间不足导致 FOTA 升级失败。新旧版本固件包中 Modem 分区剩余空间需不小于 64 KB，否则可能会导致升级失败。

解决方法：

如果 Modem 分区剩余空间不少于 64 KB，仍然打印上述异常日志，可在 `dtools` 工具的 `fota8910.xml` 文件中针对 Modem 分区升级使能 `savespace` 功能，减少升级过程中对 Modem 分区的占用。示例如下。

```
<pacsfs id="PS" ebsize="0x10000" pbsize="0x200" mount="/modem" method="diff"
savespace="y">
```

## 备注

`dtools` 工具版本需不低于 v1.0-109 才能使用 `savespace` 功能。

#### 4. 模块旧版本固件包的 AP 侧或 Flash 中的文件和制作差分包使用的基础包中的不一致

异常日志如下所示：

```
FUPD/E : flash old crc mismatch
```

分析：

上述日志表明模块旧版本固件包的 AP 侧（kernel）或者 Flash 中的文件与基础包中对应的文件内容不一致。

解决方法：

更换基础包，重新制作差分包。制作差分升级包时使用的基础包需与旧版本固件包保持一致。

#### 5. 模块旧版本固件包 CP 侧文件和制作差分包使用的基础包中的不一致

异常日志如下所示：

```
FUPD/E : failed to check blocked file size crc
FUPD/E : failed to check blocked file content size crc
```

分析：

上述日志表明模块旧版本固件包 Modem 分区（CP）中的文件与基础包中对应的文件内容不一致。

解决方法：

更换基础包，重新制作差分包。制作差分升级包时使用的基础包需与旧版本固件包保持一致。

## 6 附录 参考文档及术语缩写

表 2：参考文档

文档名称
[1] Quectel_ECx00U&EGx00U 系列_QuecOpen_CSDK_快速开发指导
[2] Quectel_ECx00U&EGx00U 系列_QuecOpen_开关机用户指导
[3] Quectel_ECx00U&EGx00U 系列_QuecOpen_SDMMC_API_参考手册

表 3：术语缩写

术语	英文全称	中文全称
AP	Application Processor	应用处理器
API	Application Program Interface	应用程序接口
App	Application	应用程序
FOTA	Firmware Over-the-Air	固件空中升级
FTP	File Transfer Protocol	文件传输协议
HTTP	HyperText Transfer Protocol	超文本传输协议
ID	Identifier	标识符
IoT	Internet of Things	物联网
LTE	Long-Term Evolution	长期演进
NV	Non-volatile Memory	非易失性存储器
PS	Packet Switch	分组交换
RTOS	Real-Time Operating System	实时操作系统

SD	Secure Digital Card	安全数字卡
SDK	Software Development Kit	软件开发工具包
SPI	Serial Peripheral Interface	串行外设接口