

AG35-Quecopen

DFOTA 操作说明

LTE 系列

版本：AG35-Quecopen_DFOTA 操作说明

日期：2018-09-11

状态：临时文件



上海移远通信技术股份有限公司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
上海市徐汇区虹梅路 1801 号宏业大厦 7 楼 邮编：200233
电话：+86 21 51086236 邮箱：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，可随时登陆如下网址：
<http://www.quectel.com/cn/support/technical.htm>
或发送邮件至：support@quectel.com

前言

上海移远通信技术股份有限公司提供该文档内容用以支持其客户的产品设计。客户须按照文档中提供的规范、参数来设计其产品。由于客户操作不当而造成的人身伤害或财产损失，本公司不承担任何责任。在未声明前，上海移远通信技术股份有限公司有权对该文档进行更新。

版权申明

本文档版权属于上海移远通信技术股份有限公司，任何人未经我司允许而复制转载该文档将承担法律责任。

版权所有 ©上海移远通信技术股份有限公司 2018，保留一切权利。
Copyright © Quectel Wireless Solutions Co., Ltd. 2018.

文档历史

修订记录

版本	日期	作者	变更表述
1.0	2018-04-08	Len Zhang	Initial
1.1	2018-04-27	Len Zhang	Added Chapter 3.2.4, 3.3, 5
1.2	2018-05-15	Len Zhang	Added Chapter 2.3
1.3	2018-08-14	Len Zhang	Added key point
1.4	2018-09-11	Len Zhang	Deleted AT and download part

目录

文档历史	3
目录	4
1 概述	5
2 差分包	6
2.1. 制作原始和目标版本	6
2.1.1. 客户在量产前制作 DFOTA 的 targetfiles.zip 并保存加以维护（必读）	6
2.2. 制作差分包	7
2.3. 修改文件权限	8
2.3.1. 手动修改权限（不推荐）	8
2.3.2. 配置文件修改权限（新版本）	9
2.3.3. 自动修改权限（最终版本）	9
3 升级	10
3.1. URC 口配置	错误!未定义书签。
3.1.1. URC 口配置	10
3.1.2. 移动差分包	10
3.1.3. 启动升级程序	11
3.2. APP 升级	11
3.3. 升级概略图	11
4 简单问题排查	13
5 优化及注意点	14
6 附录 A 参考文档及术语缩写	15

1 概述

整个升级过程按照升级区域可以分为两个部分，

1. DFOTA 升级，对应 URC 上报百分比 1%~79%，如+QIND: "FOTA","UPDATING",30
2. 备份区更新，对应 URC 上报百分比 80%~100%。

由于备份区更新是模块内部脚本来完成的，亦不属于 DFOTA 的范畴，在此不做详细说明。

2 差分包

2.1. 制作原始和目标版本

在 DFOTA 升级前，需要事先准备好对应版本的差分包。而版本包中的 `upgrade\targetfiles.zip`，其实就是原始版本的压缩文件，是为做差分包而提供的。但是因为 OpenLinux 的客户需要开发自己的东西，所以就有可能会修改到 `boot.img`、`rootfs`，然后再将修改后的文件作为原始版本烧录进模块。这就产生了 Quectel 提供的 `targetfiles.zip` 和客户修改后的 `targetfiles.zip` 文件有所不同，从而导致客户使用自己的 `targetfiles.zip` 做出来的差分包无法升级的问题。所以客户在 Release 自己版本的同时，一定要更新 `targetfiles.zip`，留作以后做差分包使用。

假设如下版本情况：

`Version1.0\upgrade\targetfiles.zip`，这个是模块中当前烧录的版本（包含 modem bin, kernel image, system 中各个文件），客户在量产前已经制作并加以维护。

`Version2.0\upgrade\targetfiles.zip`，这个当前要升级的目标版本（包含 modem bin, kernel image, system 中各个文件）。

2.1.1. 客户在量产前制作 DFOTA 的 targetfiles.zip 并保存加以维护（必读）

因为 `targetfiles.zip` 原始和目标版本的制作方法完全相同，所以此处统称为客户的 `targetfiles.zip` 制作方法。

1. Quectel 发布的版本包中包含对应二次开发环境需要的配置、工具、SDK，以及对应的 `...\upgrade\targetfiles.zip`。

2. 首先将 Quectel 提供的 `targetfiles.zip` 复制到 ubuntu 系统里并进行解压：

```
unzip -d ./target ./targetfiles.zip //使用该命令将 targetfiles.zip 解压到 target 目录里。
```

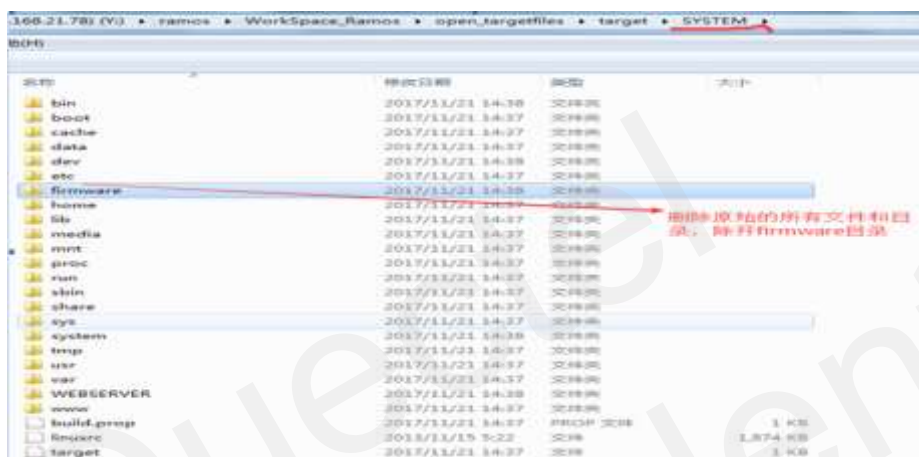
3. 客户用自己开发的 `mdm9607-perf-boot.img` 文件进行替换

使用客户的 `mdm9607-perf-boot.img` 将第 2 步 quectel 的 `targetfiles.zip` 解压后的 `BOOTABLE_IMAGES` 中的 `boot.img`、`recovery.img` 这两个文件替换掉。



4. 替换 rootfs

将 quectel targetfiles 解压后 target\SYSTEM 目录下的文件全部删除,除 firmware 目录,即仅保留 firmware 目录。再将客户二次开始时 SDK 中 ql-ol-rootfs 目录下的文件(firmware 目录除外)copy 到 SYSTEM 目录。



5. 重新打包生成 targetfiles.zip

在 zip 包解压的 target 目录并执行打包命令:

zip -qry0 ../targetfiles.zip ./*

在上层目录就能找到新做的 targetfiles.zip 包了。

注: 此处 zip 命令的“-y”参数不可省略。

2.2. 制作差分包

实测环境: Ubuntu 14.04 64bit

差分工具包: AG35_releasetools

1. 解压差分工具包

tar -xzf AG35_releasetools.tar.gz

2. 工具安装

cd AG35_releasetools

注: 在该路径下, 分别有 quectel_fullota_filelist.txt, quectel_noota_filelist.txt 两个 txt 文件, 其中 quectel_fullota_filelist.txt 文件用来配置那些文件在 FOTA 升级时, 做覆盖式升级, 而 quectel_noota_filelist.txt 文件则刚好相反, 用来配置那些文件不做升级。

拷贝 ./bsdiff 依赖的 libbz2.so.0.* 到系统库目录,

如果 ubuntu 可以联网, 建议使用 apt-get install bsdiff 命令直接安装 bsdiff 即可。

sudo cp -P libbz2.so.0* /lib/x86_64-linux-gnu/

3. 进入到 AG35_releasetools 目录下, 将两个不同发布版本包中 upgrade 目录下的 targetfiles.zip 文件分别放到 v1 和 v2 目录下, 其中 v1 目录放模块当前运行版本的 targetfiles.zip, v2 目录放待升级版本的 targetfiles.zip。
4. 运行 ./update_gen.sh a 命令即可生成相应的差分包
因为该命令会打印很多的 log 信息, 不利于发现制作过程中的异常信息。建议 ./update_gen.sh a > log 方式运行, 这样如果脚本运行有错, console 就可以只打印出 error 信息, 正常的打印则会被定向到 log 文件中。
5. 在该目录下会生成 update.zip 即为最终的差分升级包, 将该差分包放在相应的 http,ftp 服务器上即可。

注: Input parameters such as: <a --all >, 客户输入 a 参数即可。

如果在放文件, 或者执行命令时候出现权限不允许情况, 可以在该目录下执行 chmod 添加权限。

差分包制作完成后, 就可以在 console 下 (OpenLinux 客户) 进行 DFOTA 升级了。

2.3. 修改文件权限

在使用 FOTA 升级时, 有客户反映, 升级前后文件权限不一致。

如: 升级前文件权限为 755, 而在升级后, 文件的权限为 644

这种情况经常会导致用户的 app 执行失败, 为了解决类似问题, 目前提供以下三种解决方式。最新的差分包制作工具已经支持方式 2 和 3。

2.3.1. 手动修改权限 (不推荐)

手动修改文件权限的方法由于操作比较暴力, 所以不推荐后续使用。

```
script.Print("Patching remaining system files...")
for item in deferred_patch_list:
    fn, tf, sf, size, _ = item
    script.ApplyPatch("/"+fn, "-", tf.size, tf.sha1, sf.sha1, "patch/"+fn+".p")
script.SetPermissions("/system/build.prop", 0, 0, 0644)
script.SetPermissions("/system/etc/launch_adbd", 0, 0, 0744)
script.SetPermissionsRecursive("/system/test_folder/", 0, 0, 0644, 0755)

script.AddToZip(target_zip, output_zip, fota)
WriteMetadata(metadata, output_zip)

def main(argv):
```

如上图所示, 在 ota_from_target_files 文件的相应位置添加修改文件权限的命令即可。命令解释如下:

script.SetPermissions("/system/etc/launch_adbd", 0, 0, 0744)

该命令表示修改/etc/launch_adbd 文件的权限为 744, userid 是 0 (root), groupid 是 0 (root)

script.SetPermissionsRecursive("/system/test_folder/", 0, 0, 0644, 0755)

该命令表示修改 test_folder 文件夹的权限为 755, 文件夹下的文件权限为 644, userid 是 0 (root), groupid

是 `0` (root)

2.3.2. 配置文件修改权限（新版本）

该方法省去了用户修改脚本的工作，直接通过配置文件来实现文件权限修改的功能。

具体配置方法如下：

`quectel_chmod_filelist.txt` 文件是用来配置哪些文件或路径需要被修改为什么样的权限或 user，group 默认以“/”开始的被认为是有效配置，其他则可以认为是注释行。

以“/”开始，以“/”结束的行会被解析为文件夹

如下图所示：

```
1 this line is comment
2 #default
3 #example
4 #   file or folder      file_permission   userid   groupid   folder_permission
5 #   default            755          0        3         755
6 # /home/test_folder/   755          1        0         644
7 # /usr/bin/test_bin    644          2        2
8 # /usr/bin/test_bin1   742          3        1
9 # /usr/bin/test_bin2
```

其中参数的具体含义如图中的第 4 行所示

除文件名或者路径以外的其他参数都可以被省略，省略即使用默认配置

默认的 `file_permission` 权限为 755，`userid` 为 0，`groupid` 为 0，`folder_permission` 为 755

脚本使用配置文件中参数的原则是：有则用之，多则弃之，少则默认之。

2.3.3. 自动修改权限（最终版本）

该功能实现后，查分包制作工具就可以根据文件权限的属性，自动判断相关文件是否需要修改权限，完全实现了自动化配置。

具体策略如下：

定义：升级前的版本包为源文件包，升级后的版本包为目标文件包

同一文件，在源文件包中存在，在目标文件包中也存在，如果源文件包中的文件权限与目标文件包中的文件权限一致，不做权限的修改。

同一文件，在源文件包中存在，在目标文件包中也存在，如果源文件包中的文件权限与目标文件包中的文件权限不一致，目标文件包中的权限优先。

对于只存在于目标文件包中的文件，使用目标文件权限。

总结为：目标文件权限优先

3 升级

目前 open 客户只支持 cmdline 方式升级。在升级前，确保升级所需的差分包已正确存放在本地。

3.1. CMDLINE 升级

因为最终的差分包都会被保存为"/usrdata/cache/fota/ipth_package.bin"，所以 cmdline 升级方式其实也是先下载差分包，再将差分包保存为"/usrdata/cache/fota/ipth_package.bin"，如果想要查看升级进度，还需要配置 recoveryfs 中的 AT 口进行进度汇报。而在下载或者手动 push 差分包时，必须保证"usrdata"分区有足够的空间，即 **usrdata 分区的可用空间必须大于差分包的大小**，否则会导致差分包下载或 push 失败。

cmdline 的具体升级过程可划分为以下三步：

3.1.1. URC 口配置

在升级过程中，如果需要实时查看升级进度，那么就需要配置相应的 URC PORT，具体方法如下：

1. 挂载 recoveryfs

```
mtdnum=`cat /proc/mtd | grep -w recoveryfs | awk -F["d":"] '{print $2}'`  

ubiattach -m $mtdnum -d 3 /dev/ubi_ctrl  

mkdir -p /tmp/mount_recovery  

mount -t ubifs /dev/ubi3_0 /tmp/mount_recovery -o bulk_read
```
2. 配置 URC PORT

```
rm /tmp/mount_recovery/sbin/usb/boot_hsusb_composition  

ln -s /sbin/usb/compositions/recovery_9607 \  

/tmp/mount_recovery/sbin/usb/boot_hsusb_composition
```

3.1.2. 移动差分包

移动差分包时，确保原有路径已经存在，并且该路径下无老的差分包和 log 文件，最后确保文件写入 flash。

过程如下：

```
mkdir -p /usrdata/cache/fota
rm -rf /usrdata/cache/fota/*
mv 本地差分包 /usrdata/cache/fota/ipth_package.bin
sync "/sys/bus/msm_subsys/devices/subsys1/restart_level"
```

3.1.3. 启动升级程序

在 console 中，FOTA 升级程序的启动有两种方式：

1. reboot-recovery
2. sync && sys_reboot recovery (在重启前增加 sync 操作，防止文件没有完全写入 flash 导致 fota 升级失败)

这两种启动方式没有本质的区别。

该命令执行后，就可以启动 FOTA 升级程序

注：步骤 3.1.1 可以省略，因为不配置 URC PORT 对升级程序没有影响。

步骤 3.1.2 亦可以省略，如果” /usrdata/cache/fota/ipth_package.bin” 文件存在并且文件正确的情况，我们可以直接使用 reboot-recovery 命令直接进行 FOTA 升级。

步骤 3.1.3 肯定不能省略（不然怎么升级）。

3.2. APP 升级

我们在系统里面添加了一键升级方法的 APP 命令，具体命令为：qlfotapp，相对于前两种升级方法，该方法的目的就是要一键化，简单化，使用方法如下：

1. qlfotapp ./location_file

使用命令+本地差分文件即可进入 FOTA 升级。

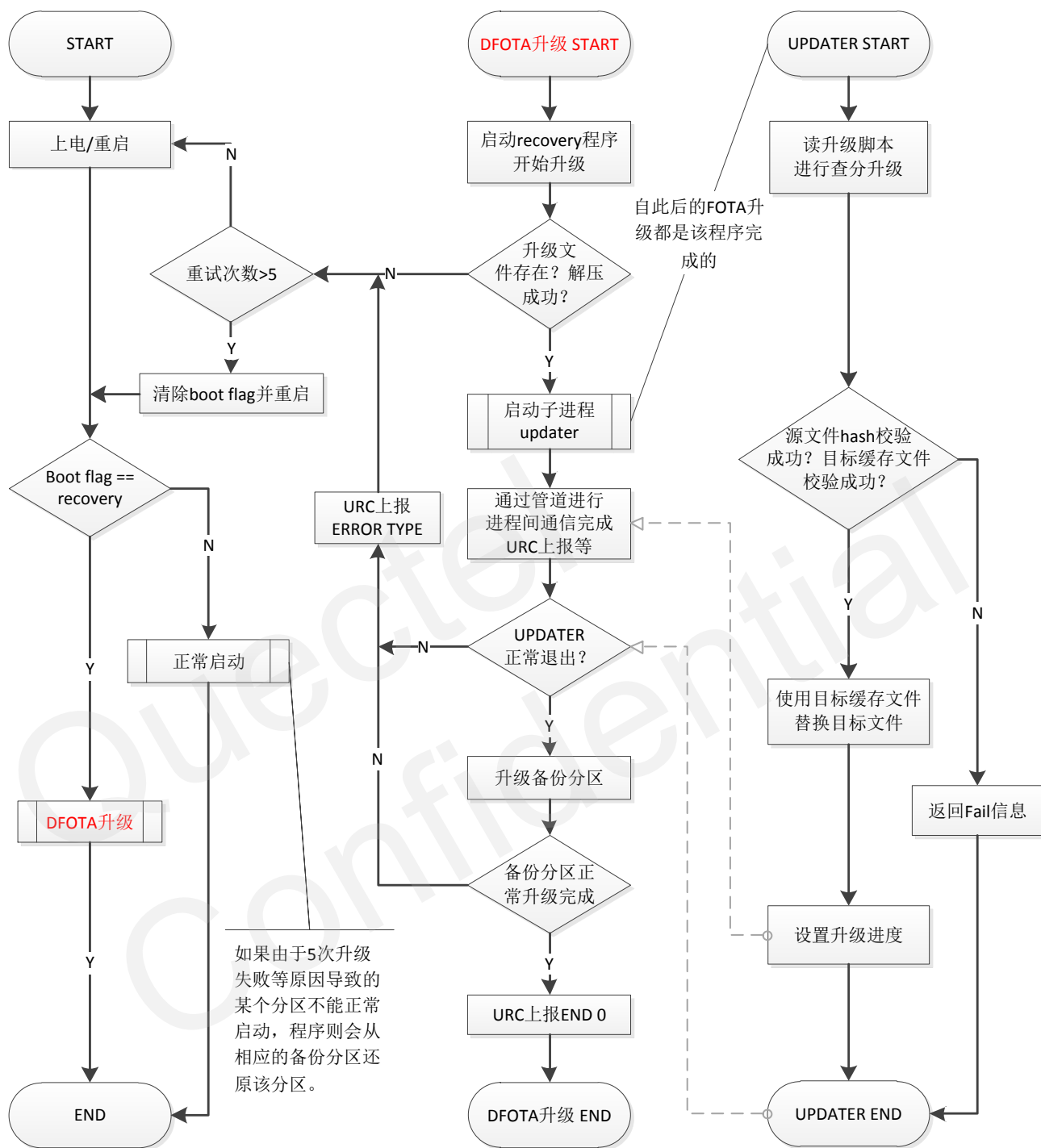
以上命令执行完后，如果差分包存在，系统就会自动重启进入 fota 升级，此后的过程就和 cmdline 升级没有什么两样。

由于该命令目前仅支持部分版本，所以在使用前，请确认所使用的版本是否存在 qlfotapp 命令。

注：初始版本的 qlfotapp 命令存在触发 fota 升级失败的问题，解决方法请参考 3.1.3

3.3. 升级概略图

下图是 FOTA 升级的概略图：



4 简单问题排查

在 DFOTA 升级过程中，一般来说，升级时间是和差分包的大小成正比的。对于通过 `http(s)/ftp` 方式下载的差分包，最好在升级前检查一下差分包的完整性，建议使用 `md5sum` 命令检查。

在升级过程中，由于各种原因，如 `flash` 的稳定性，高低温环境，随机断电等各种无法预知情况的发生，因此我们不能保证每一次升级都是 100%成功。因而对于整个升级过程，软件会尝试 5 次，如果 5 次均升级失败，则整个升级过程失败，这时会触发还原，版本会被还原为备份分区的版本。

如果程序在升级备份区时失败，模块会自动重启并进入正常状态，但此时的备份区是没有升级成功的。

在升级过程中，URC 除了正常的上报，如升级的百分比，`FOTA START`，`FOTA END 0` 以外，如果发生异常，导致升级失败，URC 也会上报相应异常的 URC 值，上报格式如下：

```
+QIND: "FOTA","START"
+QIND: "FOTA","UPDATING",<percent>
+QIND: "FOTA","UPDATING",<percent>
...
+QIND: "FOTA","END",<err>
```

此处列举出一些常见的 URC 上报值所代表的意义。

0	:	升级成功
1	:	升级备份区失败，常见于 <code>flash</code> 空间不足，导致相应 <code>ubi</code> 文件生成失败
501	:	升级开始，该 URC 一般是差分包不能正常解压，或者解压所需的空間不足导致
502	:	升级中，该 URC 一般是升级程序异常退出导致
510	:	升级中，文件校验失败
511	:	升级中，升级时所需的空間不足
OK	:	命令执行成功（仅适用 AT 升级）
ERROR	:	命令格式错误（仅适用 AT 升级）

5 优化及注意点

1. FOTA 升级后，对客户增加的文件，存在权限被更改的可能，后续会在制作差分工具中增加配置权限 list 的文件，便于客户配置。
2. FOTA 升级前，open 客户需要手动下载相应的差分包并保存在本地，并确保差分包的正确性。

6 附录 A 参考文档及术语缩写

表 1: 参考文档

序号	文档名称	备注
[1]	Quectel_AG35_DFOTA_User_Guide_V1.0	
[2]	KBA_QuecOpen_差分升级包制作方法_V1.0	