

# EC2X-QuecOpen

## 配置参数备份更新参考方案

**LTE 系列**

版本：EC2X-QuecOpen\_客户 app 重要配置参数备份更新参考方案\_V1.1

日期：2018-03-07

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](mailto:info@quectel.com)

或联系我司当地办事处，详情请登录：

<http://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://quectel.com/cn/support/technical.htm>

或发送邮件至：[support@quectel.com](mailto: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	2017-09-08	Ramos Zhang	初始版本(Quectel_QuecOpen_App 双备份实施方案建议)
1.1	2018-03-07	Ramos Zhang	更新具体实施方案参考

## 目录

文档历史 .....	2
目录 .....	3
1 引言 .....	4
2 OpenLinux 与客户相关分区介绍 .....	5
2.1. QuectelOpenLinu 与客户相关分区及用途概述 .....	5
3 如何实现客户 app 的备份还原 .....	6
3.1. 客户 app 备份还原机制参考方案流程 .....	6
3.2. 可能出现 app 备份还原同步更新的场景 .....	7
3.3. 测试方法 .....	7
4 参考脚本 .....	8
5 附录 A 参考文档及术语缩写 .....	10

# 1 引言

该参考方案大致原理是：客户的 app（或者重要的配置参数）在 system 分区和 usr\_data 分区各存放一份。每次开机时检查两个分区的 app 是都存在，任何一个分区 app 不存在，则从另外一个分区 copy 过去，如果两个分区的 app 都存在，再检查 app 的版本号(版本号读取可以依赖和 app 放在同一个目录下的版本文件或者其他更好的方法)，如果版本号不同，将版本号高的 app copy 覆盖到另外一个分区。因为这两个分区不会同时损坏，所以必能保障 app 不丢失，且即使 app 做过升级，模块后面发生了文件系统还原，还能保证 app 版本为最新的版本。

客户的一些重要配置参数可以采用相同方案来维护。

## 2 OpenLinux 与客户相关分区介绍

与客户相关的分区是指客户应用程序和配置参数数据存放相关的，可以调整的分区。其他不相关的分区客户一律不能任何修改使用。

### ■ QuectelOpenLinu 与客户相关分区及用途概述

与客户相关分区	默认大小	可用大小	分区格式	分区用途
usr_data	124.25M	约 100M	UBI 文件系统	存放读写用户数据，应用和参数配置
sys_back	58M	禁用	Image	rootfs 卷备份, 可根据实际情况调整大小, 不可使用
system	88.75M	约 19M	UBI 文件系统	rootfs 卷, 可读写分区, 可存放客户 app 和参数配置

注意 1: 如果要集成 quectel fota 升级功能, usr\_data 最少要预留 60M 空间; sys\_back 用来备份 system 分区, 如果 system 分区存放的数据较大, sys\_back 分区要做相应的调整, 调整原则最低要求为客户 system ubi size+6M, 且 system 系统客户在后期使用中不能有大数据存放增加。

注意 2: usr\_data 分区默认在/etc/init.d/find\_partitions.sh 脚本里加载完成, 且如果发现该分区 mount 失败时会将分区重新格式化使用。因此客户存放在该分区的 app 或者数据都有可能丢失的风险 (该文件是开放的, 客户也可以根据自身编程需求调整), 具体实施方法可以参考 find\_partitions.sh 里的代码。

# 3 如何实现客户 app 的备份还原

## 客户 app 备份还原机制参考方案流程

开机时 app 自动检测更新备份还原的方式：在开机运行完 `/etc/init.d/find_partitions.sh` 脚本最后（因为默认在该脚本之后模块所有的分区才加载完成）运行一个自动备份还原检测脚本



`app_auto_backup_restore.sh`

(脚本代码仅供参考)，脚本处理大致流程图如图 1。

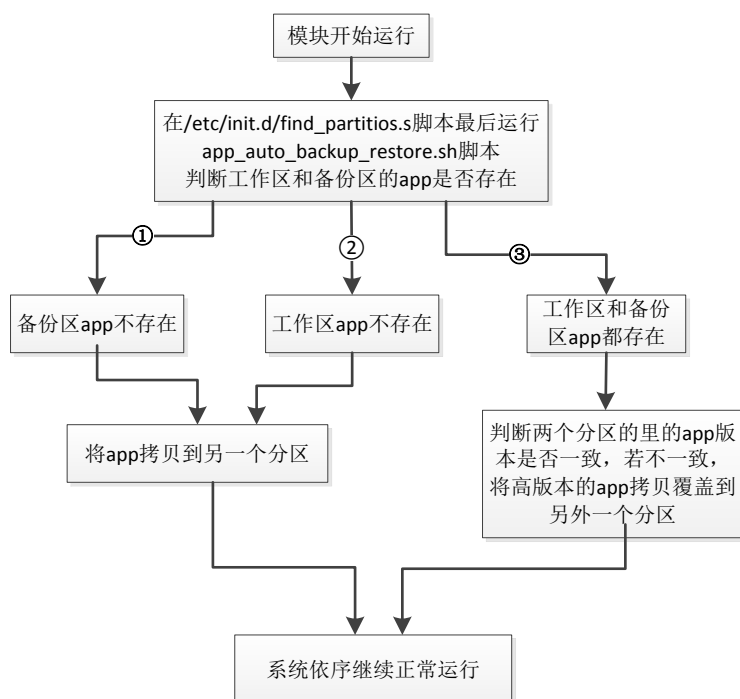


图 1: app 备份还原自检处理流程图

注意 1：由于 Linux 文件操作会存在缓存原理，如果在备份时突然断电可能会导致备份的 app 不完整，而代码却没有“察觉”到，等到下次还原时却无法运行。解决方法是，一定要先 copy 文件，最好要判断 copy 是否真正完成、完整，然后做 sync 动作，最后 copy 版本标记文件。

注意 2：app 版本标记文件要和 apps 目录（存放所有的 app，甚至包含重要参数）分开保存，copy 时先保证 apps 目录下所有文件都 copy 过去，sync 同步最后 copy app 版本标记文件，参考提供的脚本。如果客户重要参数和应用在其他目录存放，可以参考脚本增加对应的两个目录达到同样的目的。

注意 3：互相备份的两个文件夹要放在不同的分区中，因为内核检测到文件系统损坏后会 format 整个分区。

## 可能出现 app 备份还原同步更新的场景

该章节还是以脚本内容来做解说，同时分析实际运行中可能出现的集中场景及特点。该脚本重要参数和应用存放目录在 `/usrapp/apprun/apps` 目录下，备份目录为 `/usrdata/appbackup/apps`，实际运行中可能出现如下 A B C D 四种场景。

- A. 模块第一次下载运行，`/usrdata/appbackup/apps` 备份区是空的，`/usrdata/appbackup/app_ver.txt` 标记也不存在，此脚本会将 `/usrapp/apprun/apps` 文件夹下的内容 copy 到备份区，然后把标记也 copy 过去完成备份。
- B. 模块在后期运行时，假如 `usr_data` 的分区损坏，在 `/etc/init.d/find_partitions.sh` 加载时会重新格式化 `usr_data` 分区，此时备份数据和备份标记丢失，此脚本会将 `/usrapp/apprun/apps` 文件夹下的内容，`/usrapp/apprun/app_ver.txt` 版本号等文件，copy 到 `usr_data` 备份区。
- C. 客户 app 成功对自己的 app 升级后，同时对 `/usrapp/apprun/app_ver.txt` 版本号文件版本号递增。即使不去更新备份区，下次开机时，此脚本会检测到版本号变更后，自动将 `usrapp/apprun/apps` 下的文件 copy 到 `/usrdata/appbackup/apps` 目录，然后 app 版本文件再 copy 过去完成同步。
- D. 客户 app 做了升级，且备份区也更新了。但在后期的使用中模块的 `system` 分区损坏发生了还原，此时 `/usrapp/apprun/apps` 文件夹内容会回到出厂状态，此脚本会判断出来备份区域的版本较高，会将备份区 `/usrdata/appbackup/apps` 文件夹内容同步到 `/usrapp/apprun/apps` 文件夹，并同步版本号。

## 测试方法

1. 删除 `usr_data` 下的 `apps` 和版本标记文件，重新开机（或者直接运行脚本），检查 `system` 分区下是否有同步过去。
2. 同上删除 `system` 分区的 `app` 及版本号控制文件验证。
3. 在 `system` 分区 `apps` 里添加文件，同时将版本标记文件数字增加 1，重新开机（或者直接运行脚本），检查 `system` 分区下是否有同步过去。
4. 同理更新 `usr_data` 分区的 `app`，开机检查 `system` 分区下的是否有同步。
5. 正常运行后，使用 `fastboot` 方式擦除 `system` 分区，开机等系统起来后，检查 `system` 分区里 `app` 是否存在，且和 `usr_data` 分区里的一致。
6. 使用 `fastboot` 擦除 `usr_data` 分区，开机等系统起来后，检查 `usr_data` 分区里 `app` 是否存在，且和 `system` 分区里的一致。



# 4 参考脚本

```
#!/bin/sh

# Copyright (c) 2014, The Linux Foundation. All rights reserved.

#

# example app_auto_backup_restore.sh

#

# if app.bin stored in system partition (/usrapp/app.bin, /usrapp/app_Ver)
# app backup partition is usr_data in (/usrdata/appbackup/app.bin, /usrdata/appbackup/app_Ver)

#

# /usrapp/apprun/apps
# /usrapp/apprun/app_ver.txt
# /usrdata/appbackup/apps
# /usrdata/appbackup/app_ver.txt

Ver1=0
Ver2=0
app1="/usrapp/apprun/apps"
app1_Ver="/usrapp/apprun/app_ver.txt"
#just support 1,2,3,4,5,6,7 .....
app2="/usrdata/appbackup/apps"
app2_Ver="/usrdata/appbackup/app_ver.txt"
# you'd better make sure the usr_data partition is mount ok in here
if [ ! -f $app1_Ver ];then
    Ver1=-1
else
    Ver1=`cat $app1_Ver`
fi
if [ ! -f $app2_Ver ];then
    Ver2=-1
else
    Ver2=`cat $app2_Ver`
fi

if [ "$Ver1" -eq "$Ver2" ];then
    echo "app version same , exit !!!"
    exit 0
fi

echo -n " app version update now!!!"
if [ "$Ver1" -gt "$Ver2" ];then
    echo -n " app update to usr_data partition /usrdata/appbackup/"
```

```
rm -rf $app2
rm -rf $app2_Ver
mkdir -p $app2
cp -rf $app1 $app2
sync
cp -rf $app1_Ver $app2_Ver
sync
fi
if [ "$Ver1" -lt "$Ver2" ];then
    echo -n " app update to system partition /usrapp/apprun/"
    rm -rf $app1
    rm -rf $app1_Ver
    mkdir -p $app1
    cp -r $app2 $app1
    sync
    cp -rf $app2_Ver $app1_Ver
    sync
fi
```

# 5 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