



SIM8500 FOTA升级介绍

模组

芯讯通无线科技(上海)有限公司

上海市长宁区临虹路289号3号楼芯讯通总部大楼

电话: 86-21-31575100

技术支持邮箱: support@simcom.com

官网: www.simcom.com

名称:	SIM8500 FOTA升级介绍
版本:	1.0
日期:	2022.3.8
状态:	已发布

版权声明

本手册包含芯讯通无线科技（上海）有限公司（简称：芯讯通）的技术信息。除非经芯讯通书面许可，任何单位和个人不得擅自摘抄、复制本手册内容的部分或全部，并不得以任何形式传播，违反者将被追究法律责任。对技术信息涉及的专利、实用新型或者外观设计等知识产权，芯讯通保留一切权利。芯讯通有权在不通知的情况下随时更新本手册的具体内容。

本手册版权属于芯讯通，任何人未经我公司书面同意进行复制、引用或者修改本手册都将承担法律责任。

芯讯通无线科技(上海)有限公司

上海市长宁区临虹路289号3号楼芯讯通总部大楼

电话：86-21-31575100

邮箱：simcom@simcom.com

官网：www.simcom.com

了解更多资料，请点击以下链接：

<http://cn.simcom.com/download/list-230-cn.html>

技术支持，请点击以下链接：

<http://cn.simcom.com/ask/index-cn.html> 或发送邮件至 support@simcom.com

版权所有 © 芯讯通无线科技(上海)有限公司 2021，保留一切权利。

关于文档

版本历史

版本	日期	作者	备注
1.0	2022.3.8	刘明敏	第一版
1.01	2022.3.17	刘明敏	优化文档内容

适用范围

本文档适用于 SIMCom SIM8500 系列。

目录

关于文档.....	3
版本历史.....	3
适用范围.....	3
目录.....	4
1 介绍.....	5
1.1 本文目的.....	5
1.2 术语和缩写.....	5
2 FOTA 升级介绍.....	6
3 新功能 DP 介绍.....	6
3.1 DP 简介.....	6
3.2 关键软件开关.....	6
4 升级包介绍与制作.....	7
4.1 update-binary 升级程序.....	7
4.2 updater-script 升级脚本.....	7
4.3 生成升级包过程.....	7
4.4 升级包制作步骤指导.....	8
4.4.1 整体升级包步骤.....	8
4.4.2 差分升级包步骤.....	8
4.4.3 通过 PAC 包产生 FOTA 升级包.....	9
4.4.4 具体做包指令.....	9
5 升级验证.....	10
5.1 本地升级.....	10
5.2 GOTA.....	11
5.3 第三方 FOTA.....	11
6 Recovery log 抓取办法.....	12
6.1 Userdebug 版本.....	12
6.1.1 如果恢复出厂设置或者 FOTA 升级出错了，并且重启后无法开机.....	12
6.1.2 如果恢复出厂设置或者 FOTA 升级完成并重启成功.....	12
6.2 User 版本.....	12

1 介绍

1.1 本文目的

本文主要介绍 Android 10.0 的 FOTA 升级功能开发和验证办法。

本文档的对象为进行 FOTA 功能开发和测试的工程师，以及负责项目版本编制的工程师参考此应用文档，开发者可以很快理解并快速开发相关业务。

1.2 术语和缩写

名称	全称	定义
FOTA	Firmware Over-The-Air Technology	空中无线技术
DP	Dynamic Partition	动态分区调整

2 FOTA 升级介绍

FOTA（Firmware Over-The-Air）即通过空中下载的方式对用户设备进行升级的一种手段。

较新的智能设备的系统升级界面更加人性化，比如检测到新版本提示，显示新版本解决的问题，是否需要自动下载等交互界面，这种升级软件我们通常称之为 FOTA。

广义上来讲，FOTA 升级是一个复杂的系统。包括编译、版本发布、终端版本检测、在终端上检测服务器上最新版本、通过终端版本制作对应差分升级包、终端下载服务器上的升级包、本地升级等一系列过程。升级包放置到服务器和设备的“系统升级”app 检测到升级包并下载到本地的过程主要由 FOTA app 来完成。FOTA 升级会涉及到两部分内容，Recovery 模式和升级包。这两个部分是相对独立的。在升级的过程中，Recovery 模式提供基本的功能，这些功能包括挂载需要用到的分区到特定目录、输出 log 到 log 文件、检查电量是否够、验证升级包是否完整正确、解析出升级程序和升级脚本并执行之。其余升级有关的操作均由升级包中的 update-binary（升级程序）来完成。

3 新功能 DP 介绍

3.1 DP 简介

由于 Android 9.0 及以前系统上 system、vendor、product 等分区预留大小为固定值，而分区大小会随着系统文件增减进行增大或者缩小调整。因此，需要在出厂时预留足够的空间，预留空间不足会造成 FOTA 升级失败。为解决该问题，Android 10.0 增加 DP（Dynamic Partition，动态分区调整）功能，其核心是将 system、vendor、product、other 等包含只读文件系统的系统镜像放置在一个物理分区中，预留一定空间，根据各镜像间彼此消长动态调整其占用的大小，无需单独更改某物理分区。

动态分区主要特点：

- ◆ 只需定义一个 Super 分区和对应大小，super 分区的子分区存储空间可以动态调整。
- ◆ 单个分区镜像不再需要为将来的 FOTA 升级预留空间。
- ◆ super 中的剩余空间可用于所有动态分区。
- ◆ FOTA 更新时允许增加、删除和重定义只读分区大小。
- ◆ 基于 Linux kernel device-Mapper 实现用户空间的分区结构。
- ◆ super 分区的 metadata 记录各动态分区的名称和各自存储范围。
- ◆ Init 第一阶段解析和校验 super 的 metadata，创建虚拟的块设备对应各个动态分区。

3.2 关键软件开关

在使能 DP 功能后，对应的软件开关会被打开，执行 make otapackage 时会将其信息收集至 target 包中的 META/misc_info.txt 文件中，以用作打包时获取 DP 各信息之用，制作 FOTA 包时用到 DP 相关信息包括：

- ◆ `use_dynamic_partitions=true` 开启 DP 功能。
- ◆ `super_metadata_device=super` 将只读镜像的 metadata 放置分区定为 super。
- ◆ `super_block_devices=super` 将只读镜像放置的分区取为 super。
- ◆ `super_super_device_size=4299161600` super 分区大小。
- ◆ `dynamic_partition_list= system vendor product` 进行动态调整的只读镜像序列。

4 升级包介绍与制作

4.1 update-binary 升级程序

升级程序在升级包中/META-INF/com/google/android/update-binary。

Recovery 系统在进行升级时，会将此程序从升级包中解压到/tmp 这个 ramdisk 分区并创建子进程执行它。

update-binary 的来源是源代码中的 updater 程序。其是由很多的实际执行动作组成，包括写分区、解压文件、删除文件、执行程序等。updater 去解释运行升级脚本/META-INF/com/google/android/updater-script，升级脚本中包含本次升级的升级内容。

Android 10 上查看 `bootable/recovery/updater/Android.mk` 可以得知这个程序的源码包含如下文件：

```
LOCAL_SRC_FILES := \
    updater.cpp
```

4.2 updater-script 升级脚本

升级脚本在升级包中：/META-INF/com/google/android/updater-script。

升级脚本中逐条罗列了升级时要执行的动作，包括对升级基准版本和时间戳产品名的校验、各分区的校验、各分区的升级、升级进度显示等。

本脚本是在升级包制作过程中由打包脚本生成的，打包脚本在决定各分区如何升级以及获得各分区数据后再把这些数据的具体升级动作写入到此脚本文件中。

其具体的操作文件是 `build/tools/releasetools/edify_generator.py`。

4.3 生成升级包过程

打包脚本把 target 包拿过来进行加工，最终加工出 ota 升级包。

打包脚本在 `build/tools/releasetools`，主入口是 `ota_from_target_files`。

打包脚本生成升级包的过程：

- ◆ 读取传入脚本的参数、解压 target 包、创建 FOTA 升级包临时文件

- ◆ 链入芯片商自己特有处理脚本 `releasetools.py`
- ◆ 获取 `bootimg` 数据以及信息并产生升级动作
- ◆ 获取 `system` 数据以及信息并产生升级动作
- ◆ 获取 `recovery` 数据以及信息并产生升级动作
- ◆ 运行展锐嵌入脚本 `releasetools.py` 去进行 `modem bins` 和 `uboot bins` 升级动作处理
- ◆ 产生版本间 `metadata` 并写入升级包
- ◆ 制作完的升级包进行签名

4.4 升级包制作步骤指导

4.4.1 整体升级包步骤

以 SIM8500 工程为例：

1. 下载项目 AP 部分的代码
2. 设置编译环境：

```
source build/envsetup.sh
lunch sl8541e_1h10_32b_Natv-userdebug-gms
```

3. 通过 `make -jx` 命令全编整个工程
4. 然后通过命令“`make otapackage`”编译 FOTA 整包 此命令运行完后会在 `out` 目录下得到 FOTA 整包，整包目录：`out/target/product/sl8541eXXXX/sl8541eXXXX-ota-*.zip`。

注意：为了以后在版本升级时可以使用差分升级，同时要保留此版本对应的 `target` 文件。

路径为：`out/target/product/sl8541eXXXX/obj/PACKAGING/target_files_intermediates/sl8541eXXXX-target_files-*.zip`

5. 如果想制作跟 `target` 包对应的 `pac` 包，请在此时执行命令生成 `pac` 包。

```
cp_sign
makepac
```

注意：请严格在执行完 `make otapackage` 后做 `pac` 包，因为 `make otapackage` 命令会对很多 `img` 重新编译，只有在此步骤后做的 `pac` 包才是跟 `target` 包严格对应的！

4.4.2 差分升级包步骤

差分升级包分为升级差分包和降级差分包。

1. 下载代码
2. 版本号改为 A，执行 4.4.1 中所有步骤，然后保存此版本对应的 `target` 包 `A-target.zip`
3. 版本号改为 B，执行 4.4.1 中所有步骤，然后保存此版本对应的 `target` 包 `B-target.zip`
4. 执行命令制作差分升级包。

升级差分包制作：

```
./build/tools/releasetools/ota_from_target_files -k sign_key_dir -i A-target.zip B-target.zip A-B_update.zip
```


“-k”后面参数 sign_key_dir 为实际版本的 key 的放置目录。

在 user 版本是 “build/target/product/security/release/releasekey” ，

在 userdebug 版本是 “build/target/product/security/testkey” 。

降级差分包制作：

```
./build/tools/releasetools/ota_from_target_files --downgrade -k sign_key_dir -i B-target.zip A-target.zip
B-A_downgrade_update.zip
```

降级差分包指从叫新版本反向升级到较老版本，Android 7.0 之后的版本支持该种升级包，由于是回退升级中要擦除 userdata（故此种升级包慎用）。这种差分包只需要加入 “--downgrade” 参数即可，此处须注意 B 版本必须要比 A 版本要新，即编译时间靠后。

4.4.3 通过 PAC 包产生 FOTA 升级包

PAC 包直接产生 FOTA 升级包好处有：

- ◆ 无需进行 target 包编译，节省编译时间。
- ◆ 不会因 target 包遗失而无法产生升级包。
- ◆ 无需另辟很大存储空间进行 target 包和 FOTA 整包存储。

具有该种功能的 OTATOOLS 思路是从 PAC 包中的 imgs 和分区表中解析出制作 FOTA 升级包所必须的要素，并最终制作出 FOTA 升级包。

目前已有的 OTATOOLS 能同时使用 PAC 包和 target 包产生升级包，能产生 FOTA 整包、差分包、downgrade 包等。

4.4.4 具体做包指令

制作 FOTA 整包：

```
./build/make/tools/releasetools/ota_from_pac_files -p product_name -k sign_key -t pac_target full_update.zip
```

制作 FOTA 差分升级包：

```
./build/make/tools/releasetools/ota_from_pac_files -p product_name -k sign_key -b pac_base -t pac_target
delta_base-to-target.zip
```

制作降级差分升级包：

```
./build/make/tools/releasetools/ota_from_pac_files -d -p product_name -k sign_key -b pac_newer_build -t
pac_oldrget delta_update_newer-to-old.zip
```

参数说明

参数名称	参数含义
------	------

sign_key	user 版本为 build/target/product/security/release/releasekey userdebug 版本为 build/target/product/security/testkey
delta_base-to-target.zip	基准版本到目标版本的差分 FOTA 升级包。
full_update.zip	目标版本整包。
full_update_repart.zip	目标版本分区变更的 FOTA 整包。
product_name	base 对应的项目名称，只能是小写，如：sharkl3 sharkle pike2 sharkl5 roc1
pac_base	基准版本 PAC 包，即可为.gz 的压缩格式，也可为.pac 的非压缩格式。
pac_target	目标版本 PAC 包，压缩非压缩皆可。

5 升级验证

Unisoc 提供的 FOTA 方案只支持本地升级，并无实网升级的 FOTA app 以及服务器端服务。

目前两种渠道进行实网 FOTA 升级：

- ◆ 使用 Google 的 GOTA
- ◆ 使用其他三方方案的 FOTA

5.1 本地升级

本地升级主要是验证 FOTA 升级包是否能正常制作、安装，这是最基础的升级调试。

按照 4.4 节制作升级包步骤制作出整包或者差分包后，可以通过如下办法进行本地升级验证：

通过 Android 系统菜单进行升级

1. 将“升级包”放到 sd 卡的根目录下并命名为 update.zip 或将升级包 update.zip push 到/data/media/0 下
2. 设置->关于->系统软件更新，选择内置或者外置存储升级后，就会自动重启并升级。

直接进入 Recovery 模式升级

1. 将“升级包”放入到外置 SD 卡/storage/sdcard0/ 目录下
2. 使手机置于关机状态
3. 用组合键方式开机进入 Recovery 模式（手机上操作方法：关机状态下，同时按 power 键和音量下键，亮屏后松开 power 键和音量下键，进入 Recovery 模式。
4. 根据“升级包”所在位置选择相应选项进入并选择升级包进行升级。
5. 升级完成后手动选择相应选项进行重启。 此处要注意，Android 6.0 以后平台上将 sd 卡使用为内部存储后已经不能再作为本地升级的存储升级包的普通 sd 卡使用，因为其已经被挂载为 ext4 格式，而非 recovery 能识别的 FAT 格式。

使用代码方式升级

在 Frameworks/base/services/core/java/com/android/server/policy/PhoneWindowManager.java 中已增加 ota 升级的接收广播：“com.android.intent.otaupdate”

通过发送广播并在 intent 中传入 ota 文件 path 路径进行升级，广播接收代码如下：

```
filter = new IntentFilter();
filter.addAction("com.android.intent.otaupdate");
context.registerReceiver(mInstallReceiver, filter);

.....

BroadcastReceiver mInstallReceiver = new BroadcastReceiver() {
    @Override
    public void onReceive(Context context, Intent intent) {
        Log.d(TAG, "Receive otaupdate broadcast.");
        String path = intent.getStringExtra("path");
        try{
            File file = new File(path);
            RecoverySystem.installPackage(mContext,file);
        }catch(IOException e){
            e.printStackTrace();
        }
    }
};
```

发送升级广播需客户根据具体需求自己完成。

5.2 GOTA

是使用 Google 服务器进行升级包部署，并使用 gms 版本中内置的客户端进行升级包下载、校验、发起升级的一套实网升级服务。流程如下：

1. 以 Android Partner 身份申请创建 GOTA Deployment 账户。
2. 由账户所有人配置对应的设备列表。
3. 由账户所有人上传升级包。
4. 由账户所有人完成该升级包配置。
5. 设备检测升级包并下载升级。 具体请参考对应的 Google 网址。

5.3 第三方 FOTA

请自行选择并洽谈第三方 FOTA。如何编制升级包、如何部署服务器等，由第三方厂商直接支持。

6 Recovery log 抓取办法

不论是恢复出厂设置还是进行 FOTA 升级，如果出现问题，可以按照如下办法抓取 recovery 的 log。

6.1 Userdebug 版本

6.1.1 如果恢复出厂设置或者 FOTA 升级出错了，并且重启后无法开机

可在 recovery 模式下选择菜单“mount system”菜单并连接 adb，通过如下命令获取 recovery 的 log：

1. adb root
2. adb pull /tmp/recovery.log D:\log
3. D 盘 log 下的 recovery.log 即是所需要的 log

6.1.2 如果恢复出厂设置或者 FOTA 升级完成并重启成功

通过如下命令获取 recovery 的 log：

1. adb root
2. adb pull /cache/recovery D:\log
3. D 盘下面 log 里面的各文件即 recovery log。

6.2 User 版本

由于 user 版本无法连接 adb，也没有 root 权限访问 cache 分区，在没有同版本 userdebug 版本用于 debug 的情况下，可以通过如下办法获取 log。

- ◆ 使用目标版本 pac 包读取手机中的 cache 分区数据并解析出 recovery log。须注意的是手机中版本的分区表与 downloader 加载的 pac 包中的分区表必须严格一致，否则读取的 cache 数据不对，解析不出 log。
- ◆ Android 5.1 以后平台在 recovery 的菜单里有“view recovery log”，选择 cache/recovery/last_log，逐页查看，看到有出错语句的前后几页，拍照作为 log 图片。
- ◆ Android 7.0 以后会将本次升级的 recovery log 也拷贝到 sd 卡一份，user 版本的升级 log 可以直接从 sd 卡取出以供检查。8.1、9.0 的存储目录是 G:\recovery_log（一个目录此处假定 sd 卡在电脑上识别的盘符是 G）。