



Unisoc Confidential For hiar

# Android 10.0 OTA 升级指导手册

文档版本 V1.3

发布日期 2020-10-15

**版权所有 © 紫光展锐（上海）科技有限公司。保留一切权利。**

本文件所含数据和信息都属于紫光展锐（上海）科技有限公司（以下简称紫光展锐）所有的机密信息，紫光展锐保留所有相关权利。本文件仅为信息参考之目的提供，不包含任何明示或默示的知识产权许可，也不表示有任何明示或默示的保证，包括但不限于满足任何特殊目的、不侵权或性能。当您接受这份文件时，即表示您同意本文件中内容和信息属于紫光展锐机密信息，且同意在未获得紫光展锐书面同意前，不使用或复制本文件的整体或部分，也不向任何其他方披露本文件内容。紫光展锐有权在未经事先通知的情况下，在任何时候对本文件做任何修改。紫光展锐对本文件所含数据和信息不做任何保证，在任何情况下，紫光展锐均不负任何与本文件相关的直接或间接的、任何伤害或损失。

请参照交付物中说明文档对紫光展锐交付物进行使用，任何人对紫光展锐交付物的修改、定制化或违反说明文档的指引对紫光展锐交付物进行使用造成的任何损失由其自行承担。紫光展锐交付物中的性能指标、测试结果和参数等，均为在紫光展锐内部研发和测试系统中获得的，仅供参考，若任何人需要对交付物进行商用或量产，需要结合自身的软硬件测试环境进行全面的测试和调试。

Unisoc Confidential For hiar

**紫光展锐（上海）科技有限公司**



# 前言

## 概述

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

## 读者对象


本文档的对象为进行 OTA 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。

## 缩略语

缩略语	英文全名	中文解释
OTA	Over the Air	无线软件系统版本升级。
DP	Dynamic Partition	动态分区调整。

## 符号约定

在本文中可能出现下列标志，它所代表的含义如下。

符号	说明
 说明	用于突出重要/关键信息、补充信息和小窍门等。 “说明”不是安全警示信息，不涉及人身、设备及环境伤害。

## 变更信息

文档版本	发布日期	修改说明
V1.0	2019-11-13	初稿
V1.1	2020-01-19	更新模板、优化表达
V1.2	2020-04-18	更新模板、优化表达

文档版本	发布日期	修改说明
V1.3	2020-10-15	新增 UDS710_UDX710 平台 <ul style="list-style-type: none"><li>新增 ud710_2h10 modem_bins 的配置</li><li>新增 UDS710_UDX710 需进行升级分区表</li><li>新增 OTATOOLS 工具</li></ul>

## 关键字

OTA、升级。

Unisoc Confidential For hiar

# 目 录

1 Android 10.0 新功能 DP 介绍 .....	1
1.1 DP 简介 .....	1
1.2 DP 适配升级 .....	1
1.2.1 关键软件开关 .....	1
1.2.2 对 DP 分区升级处理 .....	2
1.3 非 DP 只读系统分区升级 .....	3
2 升级分区说明 .....	4
2.1 SC9863A .....	4
2.2 UDS710_UDX710 .....	6
3 升级包制作步骤 .....	10
3.1 OTA 整体升级包制作步骤 .....	10
3.2 增量升级包制作步骤 .....	11
3.3 通过 PAC 包产生 OTA 升级包 .....	11
3.3.1 OTATOOLS 工具获取 .....	11
3.3.2 具体做包指令 .....	12
3.4 Modem bins 改名映射 .....	12
3.4.1 SC9863A .....	12
3.4.2 SC9832E .....	13
3.4.3 SC7731E .....	13
3.4.4 UMS512(T) .....	14
3.4.5 UDS710_UDX710 .....	14
4 升级验证 .....	16
4.1 本地升级 .....	16
4.2 GOTA .....	17
4.3 第三方 FOTA .....	17

# 1 Android 10.0 新功能 DP 介绍

## 1.1 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 OTA 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 OTA 升级预留空间。
- Super 中的剩余空间可用于所有动态分区。
- OTA 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。

关于 DP 的具体软件原理和实现，请参考《Android 10.0 动态分区介绍》。

## 1.2 DP 适配升级

### 1.2.1 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 OTA 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。



- `super_partition_groups=group_unisoc`  
进行 `super` 分区内镜像组名定义。
- `super_group_unisoc_group_size=4299161600`  
先前定义的镜像组的大小。
- `super_group_unisoc_partition_list=system vendor product`  
镜像组包含的镜像序列。

## 1.2.2 对 DP 分区升级处理

### 1.2.2.1 DP 镜像处理

在打包过程中，专门引入新的类“DynamicPartitionsDifference”来处理动态分区里面各镜像组以及镜像组内镜像的伸缩等。

其最终要生成一个文件“dynamic\_partitions\_op\_list”，此文件在升级包中升级脚本被调用，并由升级程序使用“update\_dynamic\_partitions()”函数来解释执行。“dynamic\_partitions\_op\_list”文件中包含所有对 `super` 动态分区中的各镜像组等增减、伸缩，以及镜像组中各镜像的增减、伸缩动作。

### 1.2.2.2 产生 op\_list 步骤

**步骤 1** 添加“删除所有镜像组动作 `remove_all_groups`”。如果是差分包前后 `group` 无变化，则无此动作。

**步骤 2** 比较新旧版本的镜像组（`super_partition_groups`）是否有增减、伸缩，并产生相应增减动作：

“`add_group`”，“`resize_group`”，“`remove_group`”，具体比较规则是：

- 源版本有某 `group` 而目标版本没有某 `group`，则添加“`remove_group(group_name)`”动作。
- 源版本没有某 `group` 而目标版本有某 `group`，则添加“`add_group(group_name)`”动作。
- 源版本和目标版本都有某 `group`，且其变小，则使用“`resize_group(group_name)`”缩小（`shrink`）该 `group`。
- 原版本和目标版本都有某 `group`，且其变大，则使用“`resize_group(group_name)`”扩大（`grow`）该 `group`。

**步骤 3** 对某镜像组，检查其子镜像的增减伸缩变化，产生镜像对应的操作动作：“`add`”、“`move`”、“`remove`”、“`resize`”，具体比较产生规则：

- 源版本无某子镜像，而目标版本有，则添加“`add(sub-partition_name)`”动作。
- 源版本有某子镜像，而目标版本无，则添加“`remove(sub-partition_name)`”动作。
- 源版本和目标版本都有某子镜像，且变小，则使用“`resize(sub-partition_name)`”来缩小（`shrink`）其大小。
- 源版本和目标版本都有某子镜像，且变大，则使用“`resize(sub-partition_name)`”来缩小（`grow`）其大小。
- 源版本和目标版本都有某子镜像，但分属不同的镜像组，则使用“`move(sub-partition_name, group_name)`”来将其移动至目标 `group`。

**步骤 4** 产生各镜像的升级动作。

其差异计算仍旧采取块升级比较算法，镜像的块升级动作包含在“`sub-partition_name.transfer.list`”文件中，新增数据放在“`sub-partition_name.new.dat`”中，`patch` 数据放在“`sub-partition_name.patch.dat`”中。

而具体操作的设备则变为由先前的"add"、"move"、"remove"子镜像操作产生的虚拟块设备，获取该设备名称由“map\_partition("sub-partition\_name")”，最终产生的子镜像升级动作为：

```
block_image_update(map_partition("sub-partition_name"), package_extract_file("sub-partition_name.transfer.list"), "sub-partition_name.new.dat", "sub-partition_name.patch.dat")
```

---结束

### 1.2.2.3 具体升级动作简介

对镜像组和子镜像相关的操作如下，其主要是使用 MetadataBuilder 类（来自 liblp）对 metadata 进行计算重组，使用 DeviceMapper 类（来自 libdm）对虚拟块设备进行计算重组。

```
{ "resize", PerformOpResize},  
  { "remove", PerformOpRemove},  
  { "add", PerformOpAdd},  
  { "move", PerformOpMove},  
  { "add_group", PerformOpAddGroup},  
  { "resize_group", PerformOpResizeGroup},  
  { "remove_group", PerformOpRemoveGroup},  
  { "remove_all_groups", PerformOpRemoveAllGroups},
```

而获取设备节点的“map\_partition("sub-partition\_name")”动作则调用 DeviceMapper 类通过 sub-partition\_name 获取设备节点。

## 1.3 非 DP 只读系统分区升级

对于没有放入 super 分区的非 DP、带有文件系统（ext4）的系统分区来说，基线有 socko 和 odmko 两分区，其延续 Android 9.0 时的升级方法。同样，镜像的块升级动作包含在“sub-partition\_name.transfer.list”文件中，新增数据放在“sub-partition\_name.new.dat”中，patch 数据放在“sub-partition\_name.patch.dat”中。只是操作的设备节点是根据其固有设备节点写定的，最终升级动作为：

```
block_image_update("/dev/block/platform/soc/soc:ap-apb/71400000.sdio/by-name/partition_name", package_extract_file("partition_name.transfer.list"), "partition_name.new.dat", "partition_name.patch.dat")
```



# 2 升级分区说明

目前基线中，所有分区及其作用的简略说明、在 OTA 升级时是否进行升级，请参见下文（分别以 SC9863A 和 UDS710\_UDX710 为例，主要区别在于 modem\_bins 的升级）。

## 2.1 SC9863A

Emmc 物理分区	分区 Name	大小	作用	OTA 时是否进行升级
BOOTArea 1	spl	128K	引导 uboot	是
BOOTArea 2	spl_bak	128K	spl 备份分区	是
RPMB	Replay Protected Memory Block	128K	存放重要安全域数据	否
User Data Area	prodnv	10M	存放数种校准数据	否
User Data Area	miscdata	1M	存放重要系统数据	否
User Data Area	recovery	40M	recovery 分区	是
User Data Area	misc	1M	主系统跟 recovery 模式命令传递分区	否
User Data Area	trustos	6M	tee-tos	是
User Data Area	trustos_bak	6M	tee-tos 备份分区	是
User Data Area	sml	1M	tee-sml	是
User Data Area	sml_bak	1M	tee-sml 备份分区	是
User Data Area	uboot	1M	uboot 分区	是
User Data Area	uboot_bak	1M	uboot 备份分区	是
User Data Area	uboot_log	4M	uboot 运行 log 存储分区	否

Emmc 物理分区	分区 Name	大小	作用	OTA 时是否 进行升级
User Data Area	logo	6M	正常模式 logo	否
User Data Area	fbootlogo	M	fastboot 模式 logo	否
User Data Area	l_fixnv1	2M	modem nv	是
User Data Area	l_fixnv2	2M	modem nv 备份	是
User Data Area	l_runtimenv1	2M	运行中 nv	否
User Data Area	l_runtimenv2	2M	运行中 nv	否
User Data Area	gpsgl	1M	gps bin	是
User Data Area	gpsbd	1M	gps bin	是
User Data Area	wcnmodem	10M	wcn bin	是
User Data Area	persist	2M	FRP-google 账户存储	否
User Data Area	l_modem	25M	lte 主 modem	是
User Data Area	l_deltanv	1M	单软多硬 nv	是
User Data Area	l_gdsp	10M	modem-lte gdsp	是
User Data Area	l_ldsp	20M	modem-lte dsp	是
User Data Area	pm_sys	1M	pmsys	是
User Data Area	teecfg	1M	tee	是
User Data Area	teecfg_bak	1M	tee	是
User Data Area	boot	35M	boot 分区	是
User Data Area	dtbo	8M	dtbo 分区	是

Emmc 物理分区	分区 Name	大小	作用	OTA 时是否进行升级
User Data Area	super	4100M	DP 系统分区	是
User Data Area	cache	150M	cache 分区	否
User Data Area	socko	75M	展锐 ko 存放分区	是
User Data Area	odmko	25M	odm 的 ko 存放分区	是
User Data Area	vbmeta	1M	avb 元数据分区	是
User Data Area	vbmeta_bak	1M	avb 元数据备份分区	是
User Data Area	metadata	16M	Udc 功能使用分区	否
User Data Area	sysdumpdb	10M	sysdump 分区	否
User Data Area	vbmeta_system	1M	avb 对应 super 子分区 system 的元数据分区	是
User Data Area	vbmeta_vendor	1M	avb 对应 super 子分区 vendor/product 的元数据分区	是
User Data Area	userdata	除去上述分区后剩余大小	userdata 分区	否

## 2.2 UDS710\_UDX710

emmc 物理分区	分区 Name	大小	作用	OTA 时是否进行升级
BOOTArea 1	spl	128K	引导 uboot	是
BOOTArea 2	spl_bak	128K	spl 备份分区	是
RPMB	Replay Protected Memory Block	128K	存放重要安全域数据	否

emmc 物理分区	分区 Name	大小	作用	OTA 时是否进行升级
User Data Area	prodnv	10M	存放数种校准数据	否
User Data Area	miscdata	1M	存放重要系统数据	否
User Data Area	recovery	40M	recovery 分区	是
User Data Area	misc	1M	主系统跟 recovery 模式命令传递分区	否
User Data Area	nr_fixnv1	8M	modem nv	是
User Data Area	nr_fixnv2	8M	modem nv 备份	是
User Data Area	nr_runtimenv1	10M	运行中 nv	否
User Data Area	nr_runtimenv2	10M	运行中 nv	否
User Data Area	trustos	6M	tee-tos	是
User Data Area	trustos_bak	6M	tee-tos 备份分区	是
User Data Area	sml	1M	tee-sml	是
User Data Area	sml_bak	1M	tee-sml 备份分区	是
User Data Area	uboot	1M	uboot 分区	是
User Data Area	uboot_bak	1M	uboot 备份分区	是
User Data Area	uboot_log	4M	uboot 运行 log 存储分区	否
User Data Area	logo	8M	正常模式 logo	否
User Data Area	fbootlogo	8M	fastboot 模式 logo	否
User Data Area	l_pmsys	1M	pmsys	是
User Data Area	l_agdsp	6M	modem-lte gdsp	是

emmc 物理分区	分区 Name	大小	作用	OTA 时是否进行升级
User Data Area	gnssmodem	1M	gps bin	是
User Data Area	wcnmodem	10M	wcn bin	是
User Data Area	persist	2M	FRP google 账户存储	否
User Data Area	nr_spl	1M	5G 引导 UBOOT	是
User Data Area	nr_sml	1M	5G sml 文件	是
User Data Area	nr_uboot	1M	5G uboot 分区	是
User Data Area	nr_boot	35M	5G boot 分区	是
User Data Area	nr_pmsys	1M	5G pmsys	是
User Data Area	nr_agdsp	6M	5G lte gdsp	是
User Data Area	nr_modem	40M	5G 主 modem bin	是
User Data Area	nr_v3phy	8M	5G modem bin	是
User Data Area	nr_nrphy	8M	5G modem bin	是
User Data Area	nr_nrdsp1	5M	5G modem bin	是
User Data Area	nr_nrdsp2	5M	5G modem bin	是
User Data Area	nr_deltanv	2M	单软多硬 nv	是
User Data Area	teecfg	1M	tee	是
User Data Area	teecfg_bak	1M	tee	是
User Data Area	boot	35M	boot 分区	是
User Data Area	dtbo	8M	dtbo 分区	是

emmc 物理分区	分区 Name	大小	作用	OTA 时是否进行升级
User Data Area	super	4100M	DP 系统分区	是
User Data Area	cache	150M	cache 分区	否
User Data Area	socko	75M	展锐 ko 存放分区	是
User Data Area	odmko	25M	odm 的 ko 存放分区	是
User Data Area	vbmeta	1M	avb 元数据分区	是
User Data Area	vbmeta_bak	1M	avb 元数据备份分区	是
User Data Area	metadata	16M	Udc 功能使用分区	否
User Data Area	sysdumpdb	10M	sysdump 分区	否
User Data Area	vbmeta_system	1M	avb 对应 super 子分区 system 的元数据分区	是
User Data Area	vbmeta_vendor	1M	avb 对应 super 子分区 vendor/product 的元数据分区	是
User Data Area	userdata	除去上述分区后剩余大小	userdata 分区	否

# 3 升级包制作步骤

## 3.1 OTA 整体升级包制作步骤

步骤 1 下载项目 AP 的代码。

步骤 2 通过以下命令设置编译环境。

```
source build/envsetup.sh
lunch
kheader
```

步骤 3 通过 make 命令全编整个工程。

步骤 4 进入 “device/sprd/sharkle/sp9832e\_1hXX/” 目录（board 对应目录），手动建立 modem\_bins 子目录。

步骤 5 将展锐发布的对应 AP 版本的 modem bins 按照 “device/sprd/sharkle/common/ModemBuild.mk” 中的规定名字改名后拷贝到 “device/sprd/sharkle/sp9832e\_1hXX/modem\_bins/” 目录下。以下是 SC9832E 平台示例：

```
device/sprd/sharkle/common/ModemBuild.mk
MODEM_COPY_LIST := litemodem ltegdsp ltedsp pmsys ltenvitem wcnmodem ltedeltanv \
a) gnssmodem gnssbdmodem
```

改名对应规则请参考 “3.4 Modem bins 改名映射”。

### 说明

主 modem 以 “.dat” 为后缀的文件也要改为 “.bin” 为文件名后缀的文件。

步骤 6 通过命令 “make otapackage” 编译 OTA 整包 此命令运行完后会产生版本 target 包和 OTA 整包。整包目录：out/target/product/spXXXX/spXXXX-ota-\*.zip。

### 说明

为了以后在版本升级时可以使用差分升级，要保留此版本对应的 target 文件。路径为：  
out/target/product/spXXXX/obj/PACKAGING/target\_files\_intermediates/\*-target\_files-\*.zip

步骤 7 如果要制作版本 PAC 包，请在此时执行命令生成 PAC 包。

### 说明

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。

---结束



## 3.2 增量升级包制作步骤

差分升级包分为升级差分包和降级差分包。降级差分包是从新版本向老版本进行降级，Android 7.0 之后的版本支持该种升级包，由于是回退降级，要擦除 userdata，故此种升级包慎用。

**步骤 1** 下载 A 版本代码，执行“3.1 OTA 整体升级包制作步骤”中所有步骤，然后保存此版本对应的 target 包 A-target.zip。

**步骤 2** 下载 A 版本代码，执行“3.1 OTA 整体升级包制作步骤”中所有步骤，然后保存此版本对应的 target 包 A-target.zip。

**步骤 3** 执行命令制作差分升级包。“-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”。

– 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
```

– 降级差分包制作

此处须注意 A 版本必须要比 B 版本新，即编译时间靠后。这种差分包需要加入“--downgrade”参数。

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

----结束

## 3.3 通过 PAC 包产生 OTA 升级包

PAC 包直接产生 OTA 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 OTA 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 OTA 升级包所必须的要素，并最终制作出 OTA 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 OTA 整包、差分包、downgrade 包等。

### 3.3.1 OTATOOLS 工具获取

依次执行如下命令即可编译出项目对应 otatool：

1. source build/envsetup.sh
2. lunch 对应项目工程
3. kheader
4. make otatools

编译出的 otatool 在 out/target/product/spXXXX/目录下，由于该目录可能会被频繁的删除并重编，请将 otatool 文件夹（otatools.zip 解压也行）直接挪至其他非工程代码 linux 环境的单独目录使用。

该 otatool 的特点：

- 经一次编译，可长期使用，无需每个版本都重新产生。
- 该 tool 即能用 target 包产生 OTA 升级包，也能用 PAC 包产生 OTA 升级包。

### 3.3.2 具体做包指令

#### 制作 OTA 整包

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

#### 制作 OTA 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	<ul style="list-style-type: none"> <li>• user 版本为 build/target/product/security/release/releasekey</li> <li>• userdebug 版本为 build/target/product/security/testkey</li> </ul>
delta_base-to-target.zip	基准版本到目标版本的差分 OTA 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 OTA 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。

## 3.4 Modem bins 改名映射

### 3.4.1 SC9863A

Modem 原始名称	映射后命名
SC9600_sharkl3_pubcp_modem.dat	litemodem.bin
sharkl3_pubcp_LTEA_DSP.bin	litedsp.bin
sharkl3_pubcp_DM_DSP.bin	ltegdsp.bin
sharkl3_cm4.bin	pmsys.bin

Modem 原始名称	映射后命名
gnssbdmodem.bin	gnssbdmodem.bin
gnssmodem.bin	gnssmodem.bin
PM_sharkl3_cm4.bin	wcnmodem.bin
sharkl3_pubcp_nvitem.bin	ltenvitem.bin

### 3.4.2 SC9832E

Modem 原始名称	映射后命名
SC9600_sharkle_pubcp_Smart_Phone_modem.dat	ltemodem.bin
SharkLE_LTEA_DSP.bin	ltedsp.bin
SHARKLE2_DM_DSP.bin	ltegdsp.bin
PM_sharkle_cm4.bin	wcnmodem.bin
sharkle_cm4.bin	pmsys.bin
sharkle_pubcp_Smart_Phone_nvitem.bin	ltenvitem.bin
gnssbdmodem.bin	gnssbdmodem.bin
gnssmodem.bin	gnssmodem.bin

### 3.4.3 SC7731E

Modem 原始名称	映射后命名
SC9600_pike2_pubcp_uncache_modem.dat	wmodem.bin
PIKE2_DM_DSP.bin	wgdsp.bin
pike2_pubcp_v1_nvitem.bin	wnvitem.bin
delta_nv.bin	wdeltanv.bin
pike2_cm4.bin	pmsys.bin
PM_pike2_cm4.bin	wcnmodem.bin
gnssmodem.bin	gnssmodem.bin
gnssbdmodem.bin	gnssbdmodem.bin

### 3.4.4 UMS512(T)

Modem 原始名称	映射后命名
SC9600_sharkl5pro_pubcp_modem.dat	ltemodem.bin
sharkl5pro_pubcp_AGCP_DSP.bin	lteagdsp.bin
sharkl5pro_pubcp_LTEA_DSP.bin	ltedsp.bin
sharkl5pro_pubcp_DM_DSP.bin	ltegdsp.bin
sharkl5pro_pubcp_CDMA_DSP.bin	ltecdsp.bin
sharkl5pro_pubcp_nvitem.bin	ltenvitem.bin
sharkl5pro_pubcp_DM_DSP.bin	ltdeltanv.bin
sharkl5pro_cm4.bin	pmsys.bin
EXEC_KERNEL_IMAGE.bin	wcnmodem.bin
gnssmodem.bin	gnssmodem.bin

### 3.4.5 UDS710\_UDX710

Modem 原始名称	映射后命名
SC9600_Orca_PSCP_modem_modem.dat	nrmodem.bin
Orca_NR_phy_NR_XC0_DSP.bin	nrdspl.bin
Orca_PSCP_modem_AGCP_DSP-sign.bin	nragsdsp.bin
SC9600_Orca_NR_phy_modem-sign.bin	nrphy.bin
SC9600_Orca_V3_phy_modem-sign.bin	nrsv3phy.bin
Orca_PSCP_modem_Phone_110_nvitem.bin	nrnvitem.bin
Orca_cm4.bin	nrpmsys.bin
Orca_PSCP_modem_AGCP_DSP_ROC1_PLUS_ORCA.bin	lteagdsp.bin
roc1_cm4.bin	ltepmsys.bin
u-boot-spl-16k-sign.bin	nrsp1.bin
u-boot-sign.bin	nruboot.bin
sml-sign.bin	nrsm1.bin
boot-sign.img	nrboot.bin
gnssmodem.bin	gnssmodem.bin
EXEC_KERNEL_IMAGE.bin	wcnmodem.bin

Modem 原始名称	映射后命名
Orca_PSCP_modem_Phone_110_deltanv.bin	nr_deltanv.bin

Unisoc Confidential For hiar

# 4

## 升级验证

展锐提供的 OTA 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 OTA 升级：

- 使用 Google 的 GOTA
- 使用其他三方方案的 FOTA

### 4.1 本地升级

本地升级主要验证 OTA 升级包是否能正常制作、安装，是最基础的升级调试方式。

本地升级验证有以下两种方式。

#### 通过 Android 系统菜单进行升级

步骤 1 将“升级包”放到 SD 卡的根目录下并命名为 update.zip。或者将升级包 update.zip 放到/data/media/0 下。

步骤 2 依次点击设置->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。

---结束

#### 说明

将 SD 卡使用为内部存储后不能再做为普通 SD 卡使用。

#### 直接进入 Recovery 模式升级

步骤 1 将“升级包”放入到 SD 卡或者 cache 分区根目录下。

步骤 2 使手机置于关机状态。

步骤 3 用组合键方式开机进入 Recovery 模式（操作方法：手机关机状态下，按 power 键后再按住音量下键，亮屏后松开 power 键和音量下键，进入 Recovery 模式）。

步骤 4 根据“升级包”所在位置选择相应选项和升级包进行升级。

步骤 5 升级完成后手动选择相应选项进行重启。

---结束

## 4.2 GOTA

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。

## 4.3 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。

Unisoc Confidential For hiar