

## 1.分区适配

我们使用 USB 烧录工具，烧录 android P 的固件  
aml\_upgrade\_package.img。

从 kernel 启动的 log 打印，获取 **android P** 的分区表如下：

```
[mmcblk0p01]          bootloader offset  
0x00000000000000, size 0x000000400000
```

```
[mmcblk0p02]          reserved offset  
0x0000002400000, size 0x0000004000000
```

```
[mmcblk0p03]          cache  offset 0x0000006c00000,  
size 0x0000046000000
```

```
[mmcblk0p04]          env   offset 0x000004d400000,  
size 0x0000008000000
```

```
[mmcblk0p05]          logo  offset 0x000004e400000,  
size 0x0000008000000
```

```
[mmcblk0p06]          recovery offset 0x000004f400000,  
size 0x0000001800000
```

```
[mmcblk0p07]          misc  offset 0x0000051400000,  
size 0x0000008000000
```

```
[mmcblk0p08]          dtbo  offset 0x0000052400000,  
size 0x0000008000000
```

```
[mmcblk0p09]          cri_data offset 0x0000053400000,  
size 0x0000008000000
```

```
[mmcblk0p10]          param  offset 0x0000054400000,  
size 0x0000001000000
```

|                     |                     |                        |
|---------------------|---------------------|------------------------|
| [mmcblk0p11]        | boot                | offset 0x000055c00000, |
| size 0x000001000000 |                     |                        |
| [mmcblk0p12]        | rsv                 | offset 0x000057400000, |
| size 0x000001000000 |                     |                        |
| [mmcblk0p13]        | metadata            | offset                 |
| 0x000058c00000,     | size 0x000001000000 |                        |
| [mmcblk0p14]        | vbmata              | offset                 |
| 0x00005a400000,     | size 0x000000200000 |                        |
| [mmcblk0p15]        | tee                 | offset 0x00005ae00000, |
| size 0x000002000000 |                     |                        |
| [mmcblk0p16]        | vendor              | offset 0x00005d600000, |
| size 0x000014000000 |                     |                        |
| [mmcblk0p17]        | odm                 | offset 0x000071e00000, |
| size 0x000008000000 |                     |                        |
| [mmcblk0p18]        | system              | offset 0x00007a600000, |
| size 0x000050000000 |                     |                        |
| [mmcblk0p19]        | product             | offset 0x0000cae00000, |
| size 0x000008000000 |                     |                        |
| [mmcblk0p20]        | data                | offset 0x0000d3600000, |
| size 0x0002d0800000 |                     |                        |

使用相同的方法，烧录 android Q 的固件  
aml\_upgrade\_package.img。

我们获取 **android Q** 的分区表如下：

[mmcblk0p01] bootloader offset 0x000000000000, size 0x000000400000

[mmcblk0p02] reserved offset 0x000002400000, size 0x000004000000

[mmcblk0p03] cache offset 0x000006c00000, size 0x000032000000

[mmcblk0p04] env offset 0x000039400000, size 0x000000800000

[mmcblk0p05] logo offset 0x00003a400000, size 0x000000800000

[mmcblk0p06] recovery offset 0x00003b400000, size 0x000001800000

[mmcblk0p07] tee offset 0x00003d400000, size 0x000002000000

[mmcblk0p08] factory offset 0x00003fc00000, size 0x000000800000

[mmcblk0p09] misc offset 0x000040c00000, size 0x000000200000

[mmcblk0p10] dtbo offset 0x000041600000, size 0x000000200000

[mmcblk0p11] cri\_data offset 0x000042000000, size 0x000000800000

[mmcblk0p12] param offset 0x000043000000, size 0x000001000000

|              |          |                                            |
|--------------|----------|--------------------------------------------|
| [mmcblk0p13] | boot     | offset 0x000044800000, size 0x000001000000 |
| [mmcblk0p14] | rsv      | offset 0x000046000000, size 0x000001000000 |
| [mmcblk0p15] | metadata | offset 0x000047800000, size 0x000001000000 |
| [mmcblk0p16] | vbmata   | offset 0x000049000000, size 0x000000200000 |
| [mmcblk0p17] | super    | offset 0x000049a00000, size 0x000064000000 |
| [mmcblk0p18] | data     | offset 0x0000ae200000, size 0x0002f5c00000 |

我们把 android P 与 android Q 的分区表进行对比：

| name       | Android P | Android Q | name       |
|------------|-----------|-----------|------------|
| bootloader | 4M        | 4M        | bootloader |
| reserved   | 64M       | 64M       | reserved   |
| cache      | 1120M     | 800M      | cache      |
| env        | 8M        | 8M        | env        |
| logo       | 8M        | 8M        | logo       |
| recovery   | 24M       | 24M       | recovery   |
| misc       | 8M        | 32M       | tee        |
| dtbo       | 8M        | 8M        | factory    |
| cri_data   | 8M        | 2M        | misc       |
| param      | 16M       | 2M        | dtbo       |
| boot       | 16M       | 8M        | cri_data   |
| rsv        | 16M       | 16M       | param      |
| metadata   | 16M       | 16M       | boot       |
| vbmeta     | 2M        | 16M       | rsv        |
| tee        | 32M       | 16M       | metadata   |
| vendor     | 320M      | 2M        | vbmeta     |
| odm        | 128M      | 1600M     | super      |
| system     | 1280M     |           |            |
| product    | 128M      |           |            |
| data       | NULL      | NULL      | data       |

其中 **android Q** 分区 **cache tee factory misc dtbo super** 分区大小发生了变化，需要适配如下：

cache 800M -> 1120M

misc 2M -> 8M

dtbo 2M -> 8M

factory 0M -> 8M (新增分区，调整分区顺序)

tee 32M -> 32M (分区顺序发生了变化，需要调整分区顺序)

super 1920M -> 1864M (vendor odm system product + 3\*8M) = (factory + super + 8M) 相邻分区有 **8M** 的间隔。

适配后分区如下：

| name       | Android P | Android Q | name       |
|------------|-----------|-----------|------------|
| bootloader | 4M        | 4M        | bootloader |
| reserved   | 64M       | 64M       | reserved   |
| cache      | 1120M     | 1120M     | cache      |
| env        | 8M        | 8M        | env        |
| logo       | 8M        | 8M        | logo       |
| recovery   | 24M       | 24M       | recovery   |
| misc       | 8M        | 8M        | misc       |
| dtbo       | 8M        | 8M        | dtbo       |
| cri_data   | 8M        | 8M        | cri_data   |
| param      | 16M       | 16M       | param      |
| boot       | 16M       | 16M       | boot       |
| rsv        | 16M       | 16M       | rsv        |
| metadata   | 16M       | 16M       | metadata   |
| vbmeta     | 2M        | 2M        | vbmeta     |
| tee        | 32M       | 32M       | tee        |
| vendor     | 320M      | 8M        | factory    |
| odm        | 128M      | 1864M     | super      |
| system     | 1280M     |           |            |
| product    | 128M      |           |            |
| data       | NULL      | NULL      | data       |

修改 patch 如下：

### <1> common 修改:

---

a/arch/arm/boot/dts/amlogic/partition\_mbox\_normal\_dynamic.dtsi

+++

b/arch/arm/boot/dts/amlogic/partition\_mbox\_normal\_dynamic.dtsi

@@ -14,16 +14,16 @@

```

        parts = <15>;
        part-0 = <&logo>;
        part-1 = <&recovery>;
-       part-2 = <&tee>;
-       part-3 = <&factory>;
-       part-4 = <&misc>;
-       part-5 = <&dtbo>;
-       part-6 = <&cri_data>;
-       part-7 = <&param>;
-       part-8 = <&boot>;
-       part-9 = <&rsv>;
-       part-10 = <&metadata>;
-       part-11 = <&vbmeta>;
+       part-2 = <&misc>;

```

```

+      part-3 = <&dtbo>;
+      part-4 = <&cri_data>;
+      part-5 = <&param>;
+      part-6 = <&boot>;
+      part-7 = <&rsv>;
+      part-8 = <&metadata>;
+      part-9 = <&vbmeta>;
+      part-10 = <&tee>;
+      part-11 = <&factory>;
      part-12 = <&super>;
      part-13 = <&cache>;
      part-14 = <&data>;
@@ -50,12 +50,12 @@
    };
    misc:misc{
        pname = "misc";
-       size = <0x0 0x200000>;
+       size = <0x0 0x800000>;
        mask = <1>;
    };
    dtbo:dtbo{
        pname = "dtbo";
-       size = <0x0 0x200000>;
+       size = <0x0 0x800000>;
        mask = <1>;
    };
    cri_data:cri_data
@@ -93,13 +93,13 @@
    super:super
    {
        pname = "super";
-       size = <0x0 0x64000000>;
+       size = <0x0 0x74800000>;
        mask = <1>;
    };
    cache:cache
    {
        pname = "cache";
-       size = <0x0 0x32000000>;
+       size = <0x0 0x46000000>;
        mask = <2>;
    };

```

data:data

注意：如果是 **64** 位系统，请修改

**arch/arm64/boot/dts/amlogic/partition\_mbox\_normal\_dynamic.dtsi**

**<2> device/amlogic/xxxxxx**

Boardconfig.mk 修改

```
@@ -96,11 +96,11 @@ BOARD_ROOT_EXTRA_FOLDERS += odm
endif
```

```
ifeq ($(PRODUCT_USE_DYNAMIC_PARTITIONS), true)
```

```
-BOARD_SUPER_PARTITION_SIZE := 1677721600
```

```
+BOARD_SUPER_PARTITION_SIZE := 1954545664
```

```
BOARD_BUILD_SUPER_IMAGE_BY_DEFAULT := true
```

```
BOARD_SUPER_PARTITION_GROUPS :=
```

```
amlogic_dynamic_partitions
```

```
-
```

```
BOARD_AMLOGIC_DYNAMIC_PARTITIONS_SIZE := 1677721600
```

```
+BOARD_AMLOGIC_DYNAMIC_PARTITIONS_SIZE := 1954545664
```

```
BOARD_AMLOGIC_DYNAMIC_PARTITIONS_PARTITION_LIST :=
system vendor product odm
```

```
BOARD_EXT4_SHARE_DUP_BLOCKS := true
```

```
@@ -146,7 +146,7 @@ BOARD_PREBUILT_DTBOIMAGE :=
```

```
device/amlogic/$(PRODUCT_DIR)-kernel/dtbo.img
```

```
else
```

```
BOARD_PREBUILT_DTBOIMAGE :=
```

```
out/target/product/$(PRODUCT_DIR)/$(PRODUCT_DIR)-
```

```
kernel/dtbo.img
```

```
endif
```

```
-BOARD_DTBOIMG_PARTITION_SIZE := 2097152
```

```
+BOARD_DTBOIMG_PARTITION_SIZE := 8388608
```

```
BOARD_KERNEL_CMDLINE += androidboot.dtbo_idx=0
```

至此，我们分区表适配完成，可以编译新的固件，确认分区表对应关系如下：

| android P  |                 |                  | android Q  |                 |                  |
|------------|-----------------|------------------|------------|-----------------|------------------|
| bootloader | 0x000000000000  | 0x000000400000   | bootloader | 0x000000000000  | 0x000000400000   |
| reserved   | 0x000000240000  | 0x000000400000   | reserved   | 0x000000240000  | 0x000000400000   |
| cache      | 0x0000006c0000  | 0x000000460000   | cache      | 0x0000006c0000  | 0x000000460000   |
| env        | 0x0000004d40000 | 0x000000080000   | env        | 0x0000004d40000 | 0x000000080000   |
| logo       | 0x0000004e40000 | 0x000000080000   | logo       | 0x0000004e40000 | 0x000000080000   |
| recovery   | 0x0000004f40000 | 0x000000180000   | recovery   | 0x0000004f40000 | 0x000000180000   |
| misc       | 0x0000005140000 | 0x000000080000   | misc       | 0x0000005140000 | 0x000000080000   |
| dtbo       | 0x0000005240000 | 0x000000080000   | dtbo       | 0x0000005240000 | 0x000000080000   |
| cri_data   | 0x0000005340000 | 0x000000080000   | cri_data   | 0x0000005340000 | 0x000000080000   |
| param      | 0x0000005440000 | 0x000000100000   | param      | 0x0000005440000 | 0x000000100000   |
| boot       | 0x00000055c0000 | 0x000000100000   | boot       | 0x00000055c0000 | 0x000000100000   |
| rsv        | 0x0000005740000 | 0x000000100000   | rsv        | 0x0000005740000 | 0x000000100000   |
| metadata   | 0x00000058c0000 | 0x000000100000   | metadata   | 0x00000058c0000 | 0x000000100000   |
| vbmeta     | 0x0000005a40000 | 0x000000020000   | vbmeta     | 0x0000005a40000 | 0x000000020000   |
| tee        | 0x0000005ae0000 | 0x000000020000   | tee        | 0x0000005ae0000 | 0x000000020000   |
| vendor     | 0x0000005d60000 | 0x000000140000   | factory    | 0x0000005d60000 | 0x000000080000   |
| odm        | 0x00000071e0000 | 0x000000080000   | super      | 0x0000005e60000 | 0x0000007480000  |
| system     | 0x0000007a60000 | 0x000000500000   | data       | 0x000000d360000 | 0x0000002d080000 |
| product    | 0x000000cae0000 | 0x000000080000   |            |                 |                  |
| data       | 0x000000d360000 | 0x0000002d080000 |            |                 |                  |

**注意：**这样，我们的 tee 分区，data 分区的 offset 与 size 都相同的了，才能保证用户数据不丢失。

data 分区存储用户数据

tee 分区存储分区加密相关的密钥。

## 2. data 分区加密适配

我们从 android P 升级到 android Q，必须保证两个版本的分区加密方式相同。

### <1> 查看 fstab.amlogic

(device/amlogic/franklin/fstab.system.amlogic)

```
/dev/block/data    /data             ext4      noatime,nosuid,nodev,
nodelalloc,nomblk_io_submit,errors=panic  wait,check,fileencryp
tion=aes-256-xts,quota,formattable,reservedsize=32M
```

串口下命令确认：

```
console:/ # getprop ro.crypto.state
```

## encrypted

```
console:/ #  
console:/ # getprop ro.crypto.type  
file  
console:/ #
```

由于从 android P 开始，data 分区加密是强制性的，所以需要确保 android P 与 android Q 的 data 分区都加密，且加密属性相同，加密方式相同。

## <2> opteeos 跟 hw keymaster

如果 android P 打开了 opteeos 跟 hw keymaster，则 android Q 也必须打开，如果 android P 没有打开，则 android Q 也不能打开。

```
device/amlogic/common  
+++ b/media.mk  
@@ -25,7 +25,7 @@ endif  
ifeq ($(BOARD_WIDEVINE_OEMCRYPTO_LEVEL), 1)  
TARGET_USE_OPTEEOS := true  
TARGET_ENABLE_TA_SIGN := true  
-TARGET_USE_HW_KEYMASTER := true  
+TARGET_USE_HW_KEYMASTER := false  
endif
```

```
console:/mnt/vendor/tee #  
console:/mnt/vendor/tee #  
console:/mnt/vendor/tee # ls  
1C1EFB8EE5372643A5D68C33726C7D57 lost+found  
console:/mnt/vendor/tee #
```

**如何确认：**如果 tee 分区，只有 lost+found，则没有打开 keymaster，如果还有一串数字的目录，则打开了 keymaster 功能。

## 3.新增分区格式化

由于 android Q 上，比 android P 上新增了 factory 分区，升级过程未升级 factory.img，所以必须格式一次，否则无法正常挂载：

## device/amlogic/common/releasetools.py

```
--- a/releasetools.py  
+++ b/releasetools.py
```

```

@@ -224,7 +224,7 @@ package_extract_file("vbmeta.img",
"/dev/block/vbmeta");""")
    info.script.FormatPartition("/data")
    info.script.AppendExtra('set_update_stage("0");')
    info.script.AppendExtra('endif;')
-
+ info.script.FormatPartition("/factory")
    SetBootloaderEnv(info.script, "upgrade_step", "1")
    SetBootloaderEnv(info.script, "force_auto_update", "false")

```

#### 4.参考提交

franklin:

分区表适配，必须的：

<http://scgit.amlogic.com:8080/#/c/92093/>  
<http://scgit.amlogic.com:8080/#/c/93625/>

格式化 **factory** 分区，必须的

是否打开 **hw keymaster**，根据实际情况适配，非必须：

<http://scgit.amlogic.com:8080/#/c/92114/>

#### 5.注意事项

android Q 必须与 android P 使用相同的 key 签名，否则升级过程会签名不正确。

当 make otapackage -j8 查看最后的签名信息：

```

running: java -Xmx2048m -Djava.library.path=out/host/linux-
x86/lib64 -jar out/host/linux-x86/framework/signapk.jar -w
build/target/product/security/testkey.x509.pem
build/target/product/security/testkey.pk8
/tmp/tmpperdnhT.zip out/target/product/franklin/franklin-ota-
eng.zhigang.yu.zip

```

done.

#### build completed successfully (07:01 (mm:ss)) ####

请确认是使用的 **build/target/product/security** 下的 key 还是 **device/amlogic/common/keys** 下 key，是使用的 **releasekey** 还是 **testkey**，除了名字，确保 key 的内容是否完全一样。

**Was this page helpful?**