



# Android 15 sys\_partition.fex 使用指南

版本号: 1.1  
发布日期: 2025.3.18

## 版本历史

版本号	日期	制/修订人	内容描述
1.0	2024.10.09	AWA1579	初始版本文档
1.1	2025.3.18	XAA0312	更新 GRF 固件修改分区描述



# 目 录

<b>1 前言</b>	<b>1</b>
1.1 文档简介	1
1.2 目标读者	1
1.3 适用范围	1
1.4 名词解释	1
<b>2 分区表属性说明</b>	<b>2</b>
<b>3 各个分区的作用与说明</b>	<b>3</b>
3.1 bootloader 分区	3
3.2 env 分区	3
3.3 boot 分区	3
3.4 init_boot 分区	3
3.5 vendor_boot 分区	3
3.6 super 分区	4
3.7 misc 分区	4
3.8 vbmeta 分区	4
3.9 vbmeta_system 分区	4
3.10 vbmeta_vendor 分区	4
3.11 frp 分区	5
3.12 treadinghead 分区	5
3.13 metadata 分区	5
3.14 private 分区	5
3.15 dtbo 分区	5
3.16 media_data	6
3.17 Reserve0	6
3.18 pstore	6
3.19 UDISK 分区	6
<b>4 A/B 分区</b>	<b>7</b>
<b>5 分区表的使用</b>	<b>8</b>
5.1 分区整体框架	8
5.2 分区划分的注意事项	8
5.3 分区修改说明	8
5.3.1 增加分区	8
5.3.2 添加分区的镜像	8
5.3.3 量产保护分区	9
<b>6 FAQ</b>	<b>10</b>

6.1 分区大小与预期配置的不一致	10
6.2 无法挂载 UDISK 或某一其他分区	10
6.3 分区数量的限制	10
6.4 OTA 升级失败	11
6.5 烧写镜像不成功	11
6.6 android15 GRF 固件分区表修改	11



# 1 前言

## 1.1 文档简介

本文介绍分区表的配置方法，方便客户进行二次开发。

## 1.2 目标读者

SDK 平台负责人、方案开发负责人。

## 1.3 适用范围

Allwinner Android 15 平台。

## 1.4 名词解释

下文中单位 K 表示 KB，即 1024 字节；M 表示 1024K；G 表示 1024M。

分区的单位：扇区（一个扇区占据 512 字节），分区的个数最多为 120，超过 120 个无法烧写固件。

分区的大小：为了安全和效率考虑，分区大小最好保证为 64K 字节的整数倍，即：128(扇区) = 64K。

## 2 分区表属性说明

分区属性	作用	配置选项	性质
name	分区名	自定义	必选
size	分区大小（单位：默认为扇区，可支持 B/K/M/G 容量单位）	自定义	必选
downloadfile	烧写的镜像	自定义	可选
keydata	量产保护数据选项	1: 量产保护数据。	可选
verify	量产校验的选项	1: 量产校验; 0: 量产不校验 (默认)	可选

## 3 各个分区的作用与说明

### 3.1 bootloader 分区

Boot 数据存放分区，用于保存 boot 启动后需要的数据，如图片、logo、字体库等素材内容。

分区建议：默认 32M。

### 3.2 env 分区

环境变量分区，用于保存环境变量、启动参数等数据。

分区建议：默认 256K。

### 3.3 boot 分区

这个分区上有 Android 的引导程序，包括 kernel 和 recovery-ramdisk。该分区设备用于引导系统启动。

Android13 开始 ramdisk 转移至 init\_boot 分区。

分区建议：根据内核及 ramdisk 大小，分区需要 64M。

### 3.4 init\_boot 分区

Android13 及以上的 GMS 要求有 init\_boot 分区，用于存放原 boot 分区中的 ramdisk 文件。目的是让 boot.img 仅存放内核核心模块与 Android 版本解耦。

分区建议：默认 8M。

### 3.5 vendor\_boot 分区

用于存放内核及 ramdisk 中差异化功能模块，包括外设 ko 等。

分区建议：默认 32M。

## 3.6 super 分区

该分区存放了 system、product 以及 vendor 等镜像，升级时可以调整镜像大小，属于动态分区。其中 system 分区用于存放 android 框架系统，里面包含了 Android 用户库文件和预先安装的系统应用等，擦除了这个分区就会删除掉整个 Android 系统；vendor 分区包含所有不可分发给 Android 开源项目 (AOSP) 的二进制文件；product 分区用于存放产品专用的配置和应用，以便定制自己的系统。

分区建议：需要根据方案的具体情况来确定，需要大于 system+vendor+product 内容，默认 2G。

### 📖 说明

需要关注 OTA 升级对其产生的影响，在 ota 升级时，新版本的 Android 系统往往会比旧版本大，因此，需要预留一定空间以便日后可进行升级。

## 3.7 misc 分区

存放系统启动参数等，用于 boot 启动后读取并作出相应的动作。

分区建议：默认 16M。

## 3.8 vbmeta 分区

用于存储 dm-verity 加密数据以及全盘加密、文件加密相关的数据。

分区建议：128K。

## 3.9 vbmeta\_system 分区

存储与 system 分区完整性校验相关的数据，会被公钥签名，公钥会放置 vbmeta 分区。

分区建议：64K。

## 3.10 vbmeta\_vendor 分区

存储与 vendor 分区完整性校验相关的数据，会被公钥签名，公钥会放置 vbmeta 分区。

分区建议：64K。

## 3.11 frp 分区

Factory Reset Protection (frp) 分区，恢复出厂设置保护，防止用户信息在手机丢失后外泄。

分区建议：512K。

## 3.12 treadahead 分区

用于做系统优化的分区。

分区建议：默认 64M。

## 3.13 metadata 分区

vold、加密、ota、gsi 等模块使用，用于存放加密数据及用于 overlayfs 功能。

分区建议：默认 16M。

## 3.14 private 分区

私有分区，非标准分区。用户存放系统序列码等信息，并且使用恢复出厂设置或者量产等操作无法清除该内容，产品出厂后会在该分区写入内容。出厂后一般不允许用户修改该分区内容。

分区建议：因保存的是方案特定的私有数据，其大小取决于带保存的私有数据的大小，默认 16M。

## 3.15 dtbo 分区

用于存放 dtbo 数据，GMS 要求必备的分区，大小为 2M。

## 3.16 media\_data

media\_data 分区是用于 OEMDataPacket 工具，将一些需要打包的内容放到固件里，工具会根据内容自动扩展该分区大小。

分区建议：默认为 16M。

## 3.17 Reserve0

Reserve0 分区是用于定制化功能预留，比如将开机图片导入到分区中，开机时显示该图片。

分区建议：默认为 32K。

## 3.18 pstore

pstore 分区是提供给 pstore 功能用来记录 panic 日志以及其他死机信息的分区。

分区建议：默认为 32M。

## 3.19 UDISK 分区

用户数据 (userdata) 存储分区。系统总分区大小减去已分配的大小 (即所有未分配的剩余的空间) 都为该分区所用。UDISK 分区挂载为 data 分区，用于安装应用及存放用户数据。

分区建议：必须保证 sys\_partition.fex 的最后一个分区为 UDISK 分区，无需配置分区大小。

## 4 A/B 分区

Android 11 以后默认启用虚拟 A/B 分区，虚拟分区主要用于 super 分区中的 system、vendor、product 等动态分区，其他非动态分区依然需要进行物理 A/B 分区。主要涉及内核系统相关的分区需要改为 A/B 双分区，如：

```
;----->bootloader resource
[partition]
name = bootloader_a
size = 32M
downloadfile = "boot-resource.fex"
[partition]
name = bootloader_b
size = 32M
```

默认需要 A/B 分区的包括：bootloader、env、boot、vendor\_boot、vbmeta、vbmeta\_system、vbmeta\_vendor、dtbo。

其中，默认 downloadfile 仅需要烧写 a 分区。

## 5 分区表的使用

### 5.1 分区整体框架

分区表整体框架分为三个物理盘，第一个物理盘是分区属性 0x8000；第二个物理盘是分区属性 0xC100；第三个物理盘是分区属性 0xc200。对于 nand 驱动，把分区抽象成一个目录文件，相同属性的目录文件组合成一个物理盘。

### 5.2 分区划分的注意事项

1. 对于 nand flash，用户不能改变物理盘的分区属性，规定第一个物理盘都是 0x8000，第二个物理盘都是 0xC100，因此一个物理盘不允许插入不同属性的分区在里面。
2. 对于 nand flash，物理盘现阶段是 3 个，禁止用户随意改变。
3. 对于 4G nand flash，UDISK 分区要预留 1G，但对于  $\geq 8G$  nand flash 由于有足够空间，这里可以忽略。
4. 分区个数  $< 24$  (nand flash)。
5. UDISK 分区一定要是最后一个分区。
6. 分区名字不能改变，(除了用户新增加分区)。

### 5.3 分区修改说明

#### 5.3.1 增加分区

用户增加若干的分区，首先明确知道增加分区是什么属性的，如果不是掉电保护 ( $user\_type = 0x8000$ )，请把分区放在第一个物理盘最后的位置；如果是掉电保护 ( $user\_type = 0xC100$ )，请把分区放在第二个物理盘最后位置；UDISK 分区用作 data 分区，如需要增加掉电保护，请加上  $user\_type=0xc200$ 。

#### 5.3.2 添加分区的镜像

量产时候，把对应镜像烧录到对应的分区。

例如 boot 分区，boot.fex 代表量产的时候该分区要下载的镜像。

```
[partition]
name    = fenqu
size    = 32768
downloadfile = "fenqu.fex"
```

当用户添加新分区，并且量产的时候，烧录自定义镜像包，请按以下步骤来修改。

步骤一：添加分区，详细请根据增加分区说明来完成；

步骤二：把对应镜像包拷贝到 longan/out/pack\_out 下，打包的时候会在这个目录下找素材生成最终的固件。

#### 📖 说明

android/\$OUT 目录下的 fenqu.img 会自动链接到 longan/out/pack\_out/fenqu.fex，无需额外拷贝。

### 5.3.3 量产保护分区

量产时候，该分区量产不丢失数据。

例如以下分区，keydata 是量产保护数据属性，0x8000 是保护数据，其他值无效。

注意：如果用户要保护的分区数据过大，存在量产失败的可能，主要原因在于堆没有足够大，现阶段堆大小是 128M。

```
[partition]
name    = xxx
size    = 32768
user_type = 0x8000
keydata = 0x1
```

## 6 FAQ

### 6.1 分区大小与预期配置的不一致

在用户修改分区大小后，查看机器的真实大小，会发现跟预期的大小不一致。

此现象可分为以下两种情况。

注意：以下说明预期为在 `sys_partition.fex` 中分配大小，在真实机器中使用 `df`，`fdisk` 查看信息。

第一种：例如在 `system` 分区可能是预期 512m，查看到 `df` 真实的大小可能相差近 10m，使用 `fdisk -l` 查看，与预期大小相匹配。此为正常情况，受 `ext4` 文件系统源数据的影响，获取的是有效的数据空间，这个会比真正的 `size` 略小。

第二种：例如预期 `Udisk` 分区与 `df` 查看到的相差巨大，此情况一般正常。因会存在保留分区，因此，当直接在 `sys_partition.fex` 中直接累加用户分区后还需要其大小 1/4 的保留块。

### 6.2 无法挂载 UDISK 或某一其他分区

用户修改分区大小或增添分区后，`UDISK` 或其他分区挂载不成功，此问题分为两种情况。

第一种：无分区盘符的节点信息。此情况一般是由于用户分区后，预留剩下的 `UDSIK` 的容量不足，导致不生成相应的节点，使得挂载不成功。这时可以使用 `cat /proc/partition` 命令查看分区信息，如不存在 `UDISK` 对应的盘符，则需要先确定是否存在分区数量限制的情况，请参考相关说明处理。如非数量限制问题，则增加预留给 `UDISK` 的容量。

第二种：挂载点错误。由于使用的是硬编码，会导致修改分区表后，挂载不成功的情况，遇到上述情况，请先确定系统是否有使用软链接方式作处理，在 `dev/block` 或 `dev/block/by-name` 文件夹下查看是否有对应链接文件（通过 `ls -l` 即可看出），如不存在，则不应该改动分区数量和顺序。如存在，并且只是 `UDISK` 分区挂载不成功，则修改 `/device/softwinner/方案/vold.fstab` 中的 `UDISK` 相应的节点即可。一般使用手动 `mount` 测试分区是否能挂载，即可判断出问题。

系统采用软链接，保证 `UDISK` 分区为最后一个分区即可。

### 6.3 分区数量的限制

Nand flash 介质：

分区数量在 nand flash 介质中有最大 24 个的限制情况，因此，当超出 24 个后，会导致超出部分的分区的无法挂载成功。比较容易发现的是 UDISK 挂载不成功。

## 6.4 OTA 升级失败

修改分区表后，前后的版本如果不同 (分区个数或大小)，则会使得 ota 升级失败。所以在批量生产或者版本升级时需要特别注意这一点，否则会影响后期的用户体验。

## 6.5 烧写镜像不成功

分区表修改后，当配置的分区的数量、属性等都需要严格验证是否正确。当出现烧写不成功时，很可能是由于分区大小配置不正确导致的，如 4.2 节所说，一般使用 4G 的 flash 中，但系统分区 (即除 UDISK 分区) 配置的大小超过 3G 左右的一个临界点 (不同的 flash 会有所不同)，就会出现不可烧写或开机后无法找到 UDISK 的问题。

解决方法：此情况可根据客户需求提出申请对应的补丁 (分区大小限制的补丁)。

修改其他分区容量大小，如减少 cache 分区大小、删减不必要的分区等。一个合适的值需要尝试后获得。

## 6.6 android15 GRF 固件分区表修改

全志平台已加入 GRF 计划；芯片所使用的 Vendor 软件版本与 System 版本可能不一致。sys\_partition.fex 属于 Vendor 软件，需要在对应 Vendor 的 SDK 中进行修改。例如：A523/A527/A133 系列的 Vendor 使用 Android13，则在 Android15+Android13 的 GRF 固件中，Android13 中的 sys\_partition.fex 生效。

同时**必须**同步分区表修改到对应的 android15 sdk，以防止出现 remount 场景异常或其他问题。

注：A733 的 Vendor 冻结在 Android14。




## 著作权声明

版权所有 ©2025 珠海全志科技股份有限公司。保留一切权利。

本文档及内容受著作权法保护，其著作权由珠海全志科技股份有限公司（“全志”）拥有并保留一切权利。

本文档是全志的原创作品和版权财产，未经全志书面许可，任何单位和个人不得擅自摘抄、复制、修改、发表或传播本文档内容的部分或全部，且不得以任何形式传播。

## 商标声明

、、**全志科技**、（不完全列举）均为珠海全志科技股份有限公司的商标或者注册商标。在本文档描述的产品中出现的其它商标，产品名称，和服务名称，均由其各自所有人拥有。

## 免责声明

您购买的产品、服务或特性应受您与珠海全志科技股份有限公司（“全志”）之间签署的商业合同和条款的约束。本文档中描述的全部或部分产品、服务或特性可能不在您所购买或使用的范围内。使用前请认真阅读合同条款和相关说明，并严格遵循本文档的使用说明。您将自行承担任何不当使用行为（包括但不限于如超压，超频，超温使用）造成的不利后果，全志概不负责。

本文档作为使用指导仅供参考。由于产品版本升级或其他原因，本文档内容有可能修改，如有变更，恕不另行通知。全志尽全力在本文档中提供准确的信息，但并不确保内容完全没有错误，因使用本文档而发生损害（包括但不限于间接的、偶然的、特殊的损失）或发生侵犯第三方权利事件，全志概不负责。本文档中的所有陈述、信息和建议并不构成任何明示或暗示的保证或承诺。

本文档未以明示或暗示或其他方式授予全志的任何专利或知识产权。在您实施方案或使用产品的过程中，可能需要获得第三方的权利许可。请您自行向第三方权利人获取相关的许可。全志不承担也不代为支付任何关于获取第三方许可的许可费或版税（专利税）。全志不对您所使用的第三方许可技术做出任何保证、赔偿或承担其他义务。