



A20 sys_partition.fex 分区 表说明

V1.3

2014-06-12

CONFIDENTIAL



Revision History

Version	Date	Changes compared to previous issue
V1.0	2014-01-14	初始版本
V1.1	2014-02-20	修改 1.3 增加分区预留说明
V1.2	2014-03-17	增加修改分区表后的问题与处理
V1.3	2014-6-12	增加分区配置注意点和配置方法，适配 a20 sugar v2.1

CONFIDENTIAL



目录

1. 分区表配置的原则.....	4
1.1 分区的单位.....	4
1.2 分区的大小.....	4
1.3 分区表属性说明.....	4
2. 各个分区表的作用与配置.....	5
2.1 bootloader 分区.....	5
2.2 env 分区.....	5
2.3 boot 分区.....	5
2.4 system 分区.....	5
2.5 recovery 分区.....	5
2.6 databk 分区.....	5
2.7 misc 分区.....	6
2.8 private 分区.....	6
2.9 data 分区.....	6
2.10 cache 分区.....	6
2.11 sysrecovery 分区.....	6
2.12 UDISK 分区.....	6
3. 分区表的配置说明.....	7
3.1 分区整体框架（以下是对于 nand flash 有效）.....	7
3.2 分区划分的注意事项.....	7
3.3 分区修改说明	8
3.3.1 增加分区	8
3.3.2 添加分区的镜像	8
3.3.3 量产校验分区数据	9
3.3.4 量产保护分区	9
4. 修改分区表后的问题与处理.....	10
4.1 分区大小与预期配置的不一致.....	10
4.2 无法挂载 UDISK 或某一其他分区.....	10
4.3 分区数量的限制.....	10
4.4 OTA 升级失败.....	11
4.5 烧写镜像不成功.....	11
5. Declaration.....	12



1. 分区表配置的原则

1.1 分区的单位

分区单位：扇区（一个扇区占据 512 字节）；

分区表示个数最多 $2^{31} * 512 = 2T$ ，即容量大小最多为 2T。

1.2 分区的大小

为了安全和效率考虑，分区大小最好保证为 16M 字节的整数倍；

即：32768(扇区) = 16M。

1.3 分区表属性说明

分区属性	作用	配置选择	性质
name	分区名	自定义	必选
size	分区大小（单位：扇区）	自定义	必选
downloadfile	烧写的镜像	自定义	可选
user_type	掉电保护的配置项（nand flash）	0x8000:掉电不保护； 0xc1000:掉电保护	必选
keydate	量产保护数据选项	0x8000:量产保护数据	可选
verify	量产校验的选项	1:量产校验（默认） 0: 量产不校验	可选



2. 各个分区表的作用与配置

以下说明以 A20 公版(sugar-ref001)为参照标准。

2.1 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：因保存的为素材内容，且数据量不会很大，使用默认 32M 即可。

2.2 env 分区

环境变量分区，用于保存环境变量、启动参数等数据。

注意：SugarV1.2 使用 env 分区作为私有数据存储，但从 SugarV2.0 开始 mac、sn 等私有数据不会保存在这个分区，而是保存在 private 分区，参见 [private 分区](#)。

分区建议：因保存的为系统参数，数据量很小，使用默认 16M 即可。

2.3 boot 分区

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

现有 boot 镜像为 10M 左右，一般不会有大幅度的变化。

分区建议：无重大修改，因此默认使用 16M 即可。

2.4 system 分区

该分区为系统分区，存放 android 操作系统的，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。

各方案需要关注 OTA 升级对其产生的影响，在 ota 升级时，会对该分区进行读写操作，如果升级过程中分区被写满会导致升级失败，因此，在需要预留一定空间以便日后可进行 ota 升级。

分区建议：system 分区用于存放 system.img（路径为 android/out/target/product/方案/system.img），其大小必定要大于 system.img 文件的大小。另外，system 分区的大小严重依赖于需要预装的 apk，预装的 apk 越多，对 system 分区容量的需求就越大。分区大小需要根据方案的具体情况来确定，建议最小 512M。

2.5 recovery 分区

recovery 分区可理解为另一个启动分区（与 boot 分区类似），用户可以启动设备进入 recovery 控制台去执行高级的系统恢复或升级管理等操作。可理解为 window 的 winpe 系统。

分区建议：现有镜像文件 recovery.img 的大小为 10M 左右，日后升级可能会加入更多功能，但不会有大幅度修改，因此建议分区大小为 32M。

2.6 databk 分区

数据备份分区，**非标准分区**。用户保存用户数据分区内容，即将 data 分区数据保存在该分区，**厂商使用，非用户使用**。一般厂商会制作 databk 分区内容，例如会添加第三方软



件等。建议保留该分区，该分区用于固件修改工具。

分区建议：使用该分区，需要考虑 data 分区内数据多少（或预装的内容大小），一般使用 256M 可满足需求。

2.7 misc 分区

存放系统启动参数等，一般用户 boot 启动后读取并作出相应的动作。

分区建议：因保存的为系统参数，数据量很小，使用默认 16M 即可。

2.8 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，一般数据量很小，使用默认 16M 即可。

2.9 data 分区

用户数据分区。这个分区保存着用户数据，安装的应用的数据都在这个分区上，其中也包括从系统分区中进行优化过的 dex 文件等。擦除这个分区相当于恢复出厂设置，当第一次启动设备的时候或者重刷 ROM 之后系统会自动重建这个分区。当执行恢复出厂设置时，就是在擦除这个分区。

分区建议：由于该分区存放安装的应用以及应用的数据等，安装的应用越多以及应用保存的数据越多，对 data 分区容量的需求就越大。建议最小 1G。

2.10 cache 分区

缓存分区。最主要的用处为当进行 ota 升级时，会将相应的命令和升级包存放在该分区使用。但也有部分系统可不使用该分区，直接在如 /mnt/sdcard 分区里经常 ota 升级(需要 recovery 支持，非标准)。

分区建议：如可直接使用 /mnt/sdcard/ 分区进行 ota 升级，可仅仅划分 16M 即可，否则，需要根据规划中 ota 的升级包大小划分分区。如一般的 ota 包在 250M 左右，可划分到 350M，或根据具体的 ota 升级包预留 50% 空间。

2.11 sysrecovery 分区

系统备份分区，**非标准分区**。存放原始固件，用于当系统出现问题时，可使该分区的固件备份包来还原系统。**一键恢复功能需要配置此分区。**

分区建议：根据固件大小预留空间。

2.12 UDISK 分区

用户内置存储分区。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，不需要显式分配大小)都为该分区所用。UDISK 分区可理解为 android 系统的 /sdcard 分区即用户 internal storage，用于存放用户数据，也可将应用程序安装在该位置，可以节省 data 分区的空间。**必须保证 sys_partition.fex 的最后一个分区为 UDISK 分区。**

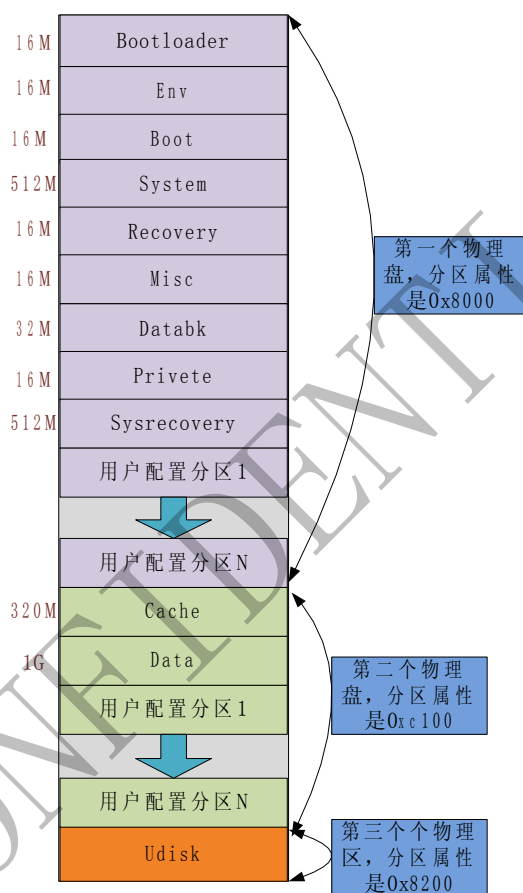
分区建议：根据 flash 容量大小分配剩余的即可。



3. 分区表的配置说明

3.1 分区整体框架（以下是对于 nand flash 有效）

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是 0xC100；第三个物理盘分区属性是隐藏的，这部分用户可以不用关心，默认是 0x8200。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。



3.2 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. 用户可以增加分区，但不能删减。（sysrecovery 分区除外）
4. 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。
5. 分区个数 < 24 （Sugar sdk v2.1）。
6. 如果开启一键恢复功能（通过固件备份包来恢复系统），sysrecovery 分区容量要大于



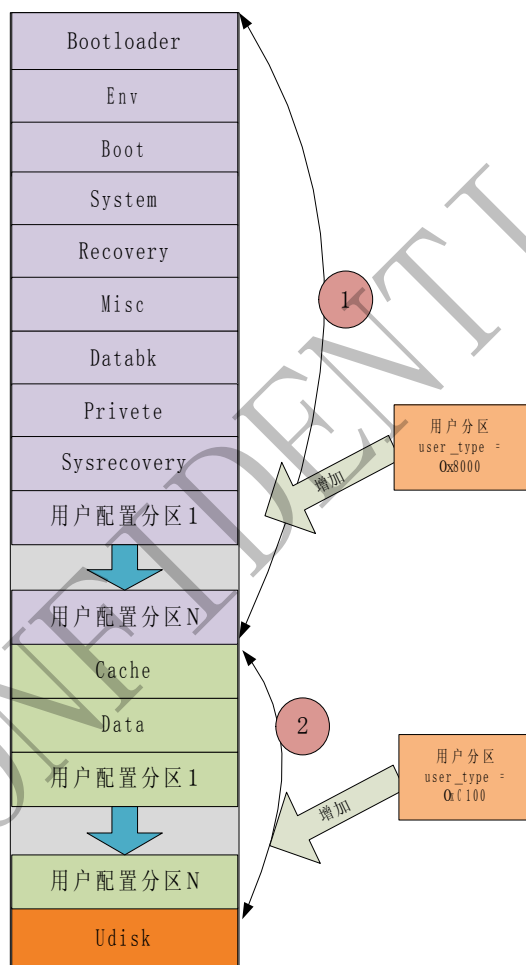
固件大小。

7. Udisk 分区一定要是最后一个分区。
8. 分区名字不能改变。(除了用户新增加分区)

3.3 分区修改说明

3.3.1 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，如步骤一；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，如步骤二；



3.3.2 添加分区的镜像

量产时候，把对应镜像拷贝到对应的分区。

例如 boot 分区，红色字体的区域代表量产的时候该分区要下载的镜像。

[partition]	
name	= boot
size	= 32768
downloadfile	= "boot.fex"



<code>user_type</code>	<code>= 0x8000</code>
------------------------	-----------------------

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 3.3.1 添加分区说明来完成；

步骤二：修改打包脚本 `lichee/tools/pack_brandy/pack`，把对应镜像包拷贝到 `lichee/tools/pack_brandy/chips/out` 下，打包的时候会在这个目录下找素材生成最终的固件。

3.3.3 量产校验分区数据

量产时候，默认是校验数据的。

例如 `sysrecovery` 分区，`verify` 默认是隐藏的，并且变量值是 1，如果想让分区量产不校验，请加上以下参数。见红色区域。

[partition]	
<code>name</code>	<code>= sysrecovery</code>
<code>size</code>	<code>= 1343488</code>
<code>downloadfile</code>	<code>= "sysrecovery.fex"</code>
<code>user_type</code>	<code>= 0x8000</code>
<code>verify</code>	<code>= 0</code>

3.3.4 量产保护分区

量产时候，该分区量产不丢失数据。

例如一下分区，`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]	
<code>name</code>	<code>= xxx</code>
<code>size</code>	<code>= 32768</code>
<code>user_type</code>	<code>= 0x8000</code>
<code>keydata</code>	<code>= 0x8000</code>



4. 修改分区表后的问题与处理

4.1 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。

此现象可分为以下三种情况。

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` 文件系统元数据的影响，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 `data` 分区 1G 或其他大小，查看到 `df` 分区只有约 128m，使用 `fdisk -l` 查看，分区确实与预期相匹配。此情况不正常，请检查编译系统中，对应分区分配的大小是否匹配。如：`BOARD_USERDATAIMAGE_PARTITION_SIZE` 的大小。

第三种：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。

4.2 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，预留剩下的容量于 `UDISK` 的容量不足，导致不生成相应的节点。导致挂载不成功。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。在 `Sugar sdk v1.2` 和 `v2.0` 中，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。

Sugar v2.1 采用软链接，保证 UDISK 分区为最后一个分区即可。

4.3 分区数量的限制

分区数量在 `Sugar sdk v1.2` 和 `v2.0` 版本会有最大 12 个的限制情况，因此，当超出 12 个后，会导致超出部分的分区无法挂在成功。比较容易发现的是 `UDISK` 挂在不成功。

分区数量大小限制可查看源码中 `\lichee\linux-3.4\modules\nand\nfd\nand_lib.h` 的宏 `MAX_PART_COUNT_PER_FTL` 的值(`v2.1` 和 `v2.0` 版本为 12 个，`v2.1` 版本会是 24 个)。如客



户使用的是旧版本代码且限制数量不满足客户需求，可向我司提出相应的补丁。

4.4 OTA 升级失败

修改分区表后，前后版本如使用 ota 升级会失败。即在批量生产时一定要注意或版本升级时需要特别注意的。否则会导致后期用户体验。

4.5 烧写镜像不成功

分区表修改后，当配置的分区大小、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 1.3 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：

- 1、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)。
- 2、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。



5. Declaration

This document [A20 sys_partition.fex 分区表说明书](#) is the original work and copyrighted property of Allwinner Technology (“Allwinner”). Reproduction in whole or in part must obtain the written approval of Allwinner and give clear acknowledgement to the copyright owner.

The information furnished by Allwinner is believed to be accurate and reliable. Allwinner reserves the right to make changes in circuit design and/or specifications at any time without notice. Allwinner does not assume any responsibility and liability for its use. Nor for any infringements of patents or other rights of the third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of Allwinner. This datasheet neither states nor implies warranty of any kind, including fitness for any particular application.

CONFIDENTIAL