



CV184X Flash 分区工具使用手册

Version: 1.0.2

Release date: 2025-09-04

©2025 北京晶视智能科技有限公司
本文件所含信息归北京晶视智能科技有限公司所有。
未经授权，严禁全部或部分复制或披露该等信息。

目录

1	声明	2
2	概述	3
2.1	概述	3
3	Nand Flash 分区	4
3.1	关于 mtd	4
3.2	分区表	4
3.3	烧写规则	5
3.3.1	SV 分区	5
3.3.2	FIP 分区	5
3.3.3	其他分区	6
3.4	NAND 分区对齐	6
4	EMMC 分区方式	7
4.1	分区方式	7
5	Nor flash 分区	8
5.1	NOR 分区对齐	8
6	分区修改	9
6.1	分区表配置	9
6.2	分区表解读	10
6.3	修改分区	10

修订记录

Revision	Date	Description
1.0.0	2025/03/27	Initial
1.0.1	2025/06/26	Update Partition configuration
1.0.2	2025/09/04	Update Partition modify

1 声明



法律声明

本数据手册包含北京晶视智能科技有限公司（下称“晶视智能”）的保密信息。未经授权，禁止使用或披露本数据手册中包含的信息。如您未经授权披露全部或部分保密信息，导致晶视智能遭受任何损失或损害，您应对因之产生的损失/损害承担责任。

本文件内信息如有更改，恕不另行通知。晶视智能不对使用或依赖本文件所含信息承担任何责任。本数据手册和本文件所含的所有信息均按“原样”提供，无任何明示、暗示、法定或其他形式的保证。晶视智能特别声明未做任何适销性、非侵权性和特定用途适用性的默示保证，亦对本数据手册所使用、包含或提供的任何第三方的软件不提供任何保证；用户同意仅向该第三方寻求与此相关的任何保证索赔。此外，晶视智能亦不对任何其根据用户规格或符合特定标准或公开讨论而制作的可交付成果承担责任。

联系我们

地址 北京市海淀区丰豪东路 9 号院中关村集成电路设计园（ICPARK）1 号楼

深圳市宝安区福海街道展城社区会展湾云岸广场 T10 栋

电话 +86-10-57590723 +86-10-57590724

邮编 100094（北京）518100（深圳）

官方网站 <https://www.sophgo.com/>

技术论坛 <https://developer.sophgo.com/forum/index.html>

2 概述

2.1 概述

本文件主要是描述 Cvitek 之 SDK 如何对 Flash 进行分区的规划.

Flash 根据 Cvitek SDK 现有版本共可分为三大类型 SPINOR / SPINAND/ EMMC.

后面章节会根据不同 Flash Type 进行说明

3 Nand Flash 分区

3.1 关于 mtd

内存技术设备即 MTD (Memory Technology Device, 缩写为 MTD), 是 Linux 系统中设备文件系统的一个类别, 主要用于闪存的应用, 是一种闪存转换层 (Flash Translation Layer, FTL)。创造 MTD 子系统的主要目的是提供一个介于闪存硬件驱动程序与应用程序之间的抽象层。

3.2 分区表

CVITEK 方案 Flash 分区以 xml 格式定义, 以 boards/default/partition/partition_spinand_page_2k.xml 为例:

```
<physical_partition type="spinand">
  <partition label="fip" size_in_kb="2560" file="fip.bin"/>
  <partition label="BOOT" size_in_kb="8192" file="boot.spinand"/>
  <partition label="MISC" size_in_kb="384" file="logo.jpg" />
  <partition label="ENV" size_in_kb="128" file="" />
  <partition label="ROOTFS" size_in_kb="71680" file="rootfs.spinand" />
  <partition label="SYSTEM" size_in_kb="20480" file="system.spinand" mountpoint="" type=
  ↳"ubifs" />
  <partition label="CFG" size_in_kb="4096" file="cfg.spinand" mountpoint="/mnt/cfg" type=
  ↳"ubifs" />
  <partition label="DATA" file="" mountpoint="/mnt/data" type="ubifs" />
</physical_partition>
```

以 2KB page size 128KB blocksize 的 NAND flash 为例: 由 xml 文件上数据, 将各分区大小换算成 block 大小后 (公式: block 个数 = 分区大小 / 单一 block 大小), 如下所示:

Partition	Start block offset	Number of blocks	Binary files
FIP	0	20	fip.bin
BOOT	24	64	boot.spinand
MISC	顺排（遇坏块则跳过）	3	logo.jpg
ENV	顺排（遇坏块则跳过）	1	Null（无内容）
ENV_BAK	顺排（遇坏块则跳过）	1	Null（无内容）
ROOTFS	顺排（遇坏块则跳过）	560	rootfs.spinand
SYSTEM	顺排（遇坏块则跳过）	160	system.spinand
CFG	顺排（遇坏块则跳过）	32	cfg.spinand
DATA	顺排（遇坏块则跳过）	Don' t Care	Null（无内容）

3.3 烧写规则

3.3.1 SV 分区

SV 分区跟处理器绑定，用户只需选择 2KB 或 4KB page size 的 bin 并将其烧写，请勿自行修改分区内容。以 2KB page size 为例，请将 block 0, 1, 2, 3 都烧写 182x_sv_2k_pg.bin，如遇坏块则略过该 block。如果前四个 block 都是坏块，请勿使用该 SPI NAND flash。

3.3.2 FIP 分区

FIP 分区包含两个部分：处理器相关的 Bootloader(无开源)，u-boot。CVITEK 编译流程会自动将两者打包成一个 fip.bin。

nand 烧录时烧录逻辑：从 block 0~19 之间依序挑选好块，总共烧写两份，第一份会写在 block 0~9，第二份会烧写在 block 10~19，互为备份。

fip.bin 本身烧入进 spinand 大概会使用到 3~4 个 blocks，但因 spinand 特性问题，block 可能会出现坏块状况，所以剩余未使用的 block 预留用于出现坏块时使用。

例一

没有坏块的话，将第一份 fip.bin 烧写至 block 0, 1, 2, 3, 4；第二份 fip.bin 烧写至 block 9, 10, 11, 12, 13。

例二

若 block 4, 11 为坏块，请将第一份 fip.bin 烧写至 block 0, 1, 2, 3, 5；第二份 fip.bin 烧写至 block 9, 10, 12, 13, 14

3.3.3 其他分区

照分区表配置，依序烧写，遇到坏块则略过，跳下一个好块再烧写。

3.4 NAND 分区对齐

分区大小需按照 nand flash 的 erase size 对齐，erase size 可查阅 nand flash 的 spec 手册，当前有 128/256k erase size，若无法获取到 nand flash 的 erase size 大小可默认按照 256k 大小对齐

4 EMMC 分区方式

4.1 分区方式

分区方式是通过 Linux Kernel 的 cmdline partition 的方式进行分区管理而不是 elf partition；因此需要设置如下选项：备注：默认已设置如下选项，用户不需要自行设置

```
CONFIG_BLK_CMDLINE_PARSER=y
CONFIG_PARTITION_ADVANCED=y
CONFIG_CMDLINE_PARTITION=y
CONFIG_EFI_PARTITION=n
```

详细可以参考下列信息：<https://www.kernel.org/doc/Documentation/block/cmdline-partition.txt>

5 Nor flash 分区

5.1 NOR 分区对齐

分区大小需按照 nor flash 的 erase size 对齐，驱动支持 4/64k erase size，当使能 SDK 的 CONFIG_USE_4K_ERASE_SIZE_FOR_JFFS2 选项时，分区大小需按照 4k 对齐反之则会按照 64k 对齐

6 分区修改

6.1 分区表配置

CV184x SDK 把分区表中的各分区添加为 `build/Kconfig.partitions` 配置项，不同 EVB defconfig 档案记录了对应 Flash 的各分区默认配置值。

执行 `defconfig` 命令选定 EVB 时，先解析 `build/Kconfig.partitions` 文件，再加载 `defconfig` 档案中的默认配置，最后生成 `xml` 文件形式的分区表，保存为 `build/partitions.xml`。以 `cv1842hp_wevb_0014a_spinor` 为例：

```
$ source build/envsetup_soc.sh

$ defconfig cv1842hp_wevb_0014a_spinor

$ file build/boards/cv184x/cv1842hp_wevb_0014a_spinor/partition/partition_spinor.xml
build/boards/cv184x/cv1842hp_wevb_0014a_spinor/partition/partition_spinor.xml: symbolic link to .
→ ../../../../partitions.xml

$ cat build/partitions.xml
<?xml version="1.0" ?>
<physical_partition type="spinor">
<!--This is an auto-generated file. DO NOT EDIT manually.-->
<partition label="fip" size_in_kb="2048" readonly="false" file="fip_spl.bin" mountpoint="" type="
→"/>
<partition label="2nd" size_in_kb="1024" readonly="false" file="yoc.bin" mountpoint="" type=""/
→>
<partition label="BOOT" size_in_kb="3072" readonly="false" file="boot.spinor" mountpoint=""
→type=""/>
<partition label="MISC" size_in_kb="128" readonly="false" file="logo.jpg" mountpoint="" type="
→"/>
<partition label="ENV" size_in_kb="64" readonly="false" file="" mountpoint="" type=""/>
<partition label="ENV_BAK" size_in_kb="64" readonly="false" file="" mountpoint="" type=""/>
<partition label="ROOTFS" size_in_kb="4352" readonly="false" file="rootfs.spinor" mountpoint="
→" type=""/>
<partition label="DATA" size_in_kb="1024" readonly="false" file="data.spinor" mountpoint="
→mnt/data" type="jffs2"/>
</physical_partition>
```

6.2 分区表解读

分区表的标签含义如下：

1. physical_partition type: flash 种类。
2. partition label: 分区名称。
3. size_in_kb: 分区大小 (以 KB 为单位)。
4. file: 所指向的 image file 名称。
5. type: (在 partition label 栏位中) 文件系统格式。
6. mountpoint: 分区挂载路径。

CV184x SDK 会生成以下的 image file, 每个档案代表不同的分区, 分列如下:

- FIP : Bootloader/U-Boot 分区
- CV184X CA3 采用 FSBL+UBOOT 架构, 最后打包后也复用 (FIP) 檔名, 方便后续使用。
- 2nd(双系统) : Yun on Processor (YOC) 所在分区
- BOOT : Partition for Linux Kernel 分区
- MISC : Boot Logo 分区
- ROOTFS : root file system 分区
- SYSTEM: CVITEK libraries 所在分区
- DATA : 使用这资料分区

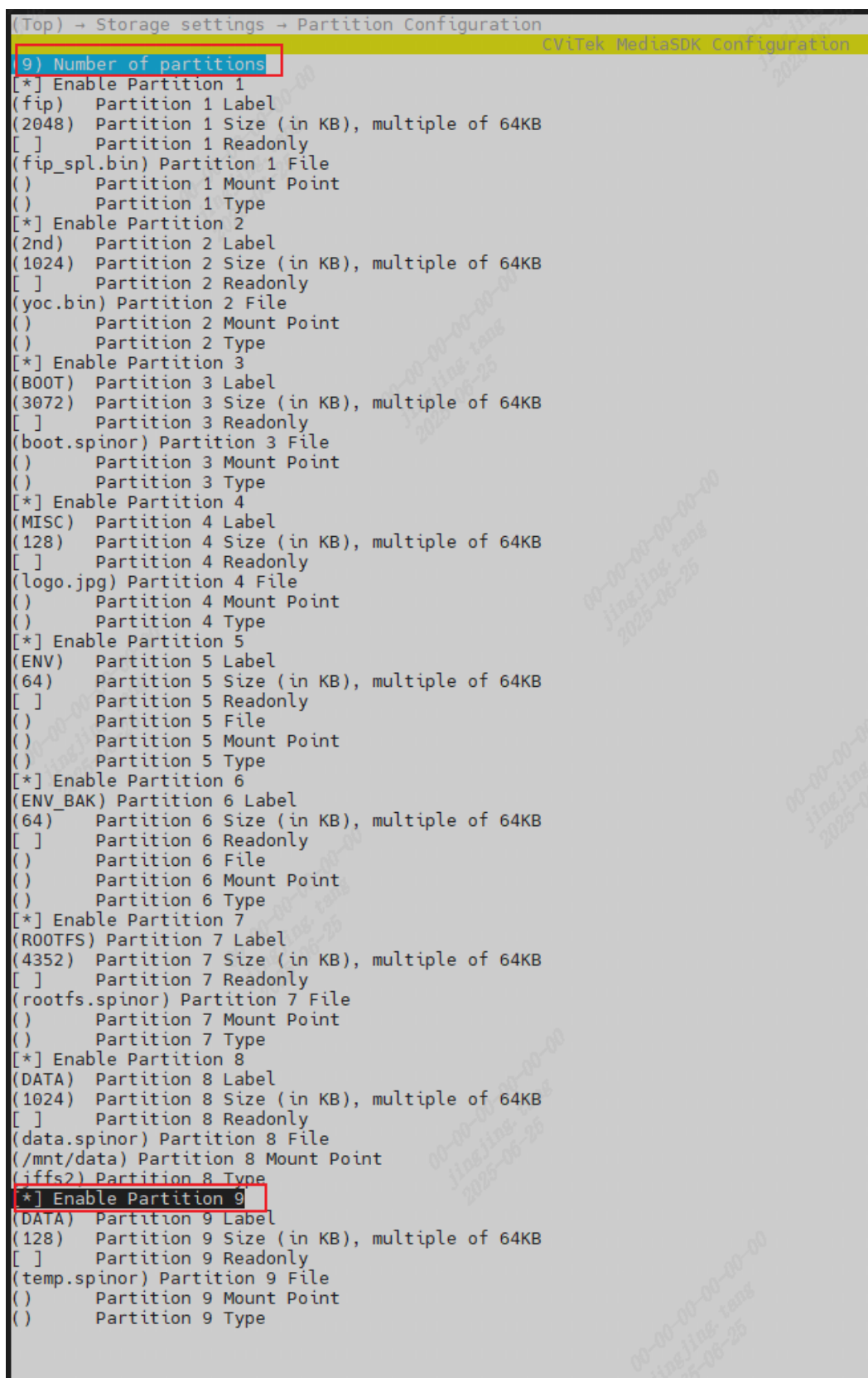
注: 2nd 分区为双系统特有分区, 仅在双系统环境下存在

6.3 修改分区

- 修改分区表

在选定 EVB 后, 支持通过 menuconfig 命令以图形界面的方式修改 build/partitions.xml 分区表。

1. 需要修改已有分区的大小, 可以直接修改对应配置项的值, 分区大小对齐参阅前面章节;
2. 当需要增减分区时, 先修改分区的数量, 然后, 取消勾选不需要的分区, 或者设置新的分区, 如下图所示, 修改完后保存设定, 退出菜单界面就会自动更新到 build/partitions.xml 分区表;
3. 当最后一个分区的大小删除不填写时, 生成分区表时会自动计算剩余大小作为最后一个分区的大小。



例如，上图保存修改后，更新如下：

```
$ cat build/partitions.xml
```

(下页继续)

(续上页)

```
<?xml version="1.0" ?>
<physical_partition type="spinor">
<!--This is an auto-generated file. DO NOT EDIT manually.-->
<partition label="fip" size_in_kb="2048" readonly="false" file="fip_spl.bin" mountpoint="" type="
→"/>
<partition label="2nd" size_in_kb="1024" readonly="false" file="yoc.bin" mountpoint="" type=""/
→>
<partition label="BOOT" size_in_kb="3072" readonly="false" file="boot.spinor" mountpoint=""
→type=""/>
<partition label="MISC" size_in_kb="128" readonly="false" file="logo.jpg" mountpoint="" type="
→"/>
<partition label="ENV" size_in_kb="64" readonly="false" file="" mountpoint="" type=""/>
<partition label="ENV_BAK" size_in_kb="64" readonly="false" file="" mountpoint="" type=""/>
<partition label="ROOTFS" size_in_kb="4352" readonly="false" file="rootfs.spinor" mountpoint="
→" type=""/>
<partition label="DATA" size_in_kb="1024" readonly="false" file="data.spinor" mountpoint="/
→mnt/data" type="jffs2"/>
<partition label="DATA" size_in_kb="128" readonly="false" file="temp.spinor" mountpoint=""
→type=""/>
</physical_partition>
```

· 指定分区表

需要更改为指定分区表，只需要修改 EVB 配置目录中的分区表软连接 build/boards/cv184x/cv1842hp_wevb_0014a_spinor/partition/partition_spinor.xml