

飞凌嵌入式技术服务部

关于 K9GAG08U0D (2G Nand) 升级到 K9LBG08U0D (4G Nand) 的升级说明

Linux3.0.1

如果您的产品使用的是 OK6410 或者 TE6410 的 2G NandFlash, 运行的 Linux3.0 系统, 现在要更换 4G NandFlash, 继续运行 Linux3.0, 同时升级后的 Linux 系统兼容 NandFlash 2G 平台, 下面是您需要关注的内容。

一键烧写文件 mmc.bin 使用我们 2012-09 发布的版本, 该文件支持 4G NandFlash 烧写, 同时也兼容 2G NandFlash 的烧写, mmc.bin 会自动识别您的 Flash 大小。

当然您最关心的是 Uboot, Linux 内核代码和文件系统修改的内容了, 我们提供了文件夹, 里面有需要您关注的文件, 按照下面我们详细说明, 使用新文件覆盖掉您的原有代码文件即可。

注意: 2012 年我们发布了三个版本, 分别为 Linux3.0-2012-02 版本, Linux3.0-2012-05 版本, Linux3.0-2012-09 版本, 其中 Linux3.0-2012-09 版本已经实现了 4G nandFlash 的支持, 同时兼容 NandFlash 2G 版本。以下修改适用于 2012 年 2 月和 5 月发布的版本。

另外需要注意的是, 如果您使用的是飞凌早期发布的 Linux2.6.36 版本, 或者 Linux2.6.28 版本, 如下修改方法可以做一个参考:

Uboot 代码修改

1 common 文件夹修改内容:

uboot1.1.6/common/cmd_nand.c

该文件主要实现 Uboot 操作 NandFlash 的命令集合, 如执行 nand write.uboot 会调用该文件

uboot1.1.6/common/main.c

#用于实现启动菜单显示

uboot1.1.6/common/env_nand.c

#执行 saveenv 命令将调用该文件, 用于存储环境变量到 NandFlash

2 cpu 文件夹修改内容

uboot1.1.6/cpu/s3c64xx/nand.c

该文件是 NandFlash 修改重要文件, 定义 8 位硬件 ECC 功能。

uboot1.1.6/cpu/s3c64xx/nand_cp.c

该文件实现 NandFlash 启动系统关键代码, 比如把 Uboot 代码拷贝到内存运行。

uboot1.1.6/cpu/s3c64xx/usbd-otg-hs.c

用于菜单形式实现 DNW 烧写系统功能, 这部分与 NandFlash 关系不大, 但跟 Uboot 菜单项相关。

3 driver 文件夹修改内容

uboot1.1.6/drivers/nand/nand_base.c

NandFlash 驱动重要文件，实现 NandFlash 的读写功能。

uboot1.1.6/drivers/nand/nand_ids.c

NandFlash ID 定义文件，里面定义了我们目前支持的 NandFlash 类型，当然里面有一些类型是系统自带的。

uboot1.1.6/drivers/nand/nand_util.c

#NandFlash 驱动文件，实现 NandFlash 读写辅助功能。

4 include 文件夹修改内容

uboot1.1.6/include/configs/smdk6410.h

#NandFlash 相关的宏定义

uboot1.1.6/include/linux/mtd/nand.h

定义 NandFlash 页最大空间，OOB 最大空间

uboot1.1.6/include/s3c6410.h

s3c6410 头文件

注意：修改文件后，重新编译前，需要执行 **make clean** 命令

以上是 Uboot 代码 2G 换成 4G 需要修改的地方，下面我们介绍 Linux3.0 内核需要修改的文件。

Linux 内核修改文件

Linux-3.0.1/arch/arm/mach-s3c64xx/mach-smdk6410.c

该文件是 6410 平台重要文件，实现平台设备的添加，初始化工作，
这个文件里面有一个重要地方，就是对 NandFlash 的分区信息，

```
struct mtd_partition ok6410_nand_part[] = {
    {
        .name          = "Bootloader",
        .offset         = 0,
        .size           = (2 * SZ_1M),
        .mask_flags     = MTD_CAP_NANDFLASH,
    },
    {
```

```

        .name          = "Kernel",
        .offset         = (2 * SZ_1M),
        .size           = (5 * SZ_1M),
        .mask_flags     = MTD_CAP_NANDFLASH,
    },
    {
        .name          = "User",
        .offset         = (7 * SZ_1M),
        .size           = (200 * SZ_1M),
    },
    {
        .name          = "File System",
        .offset         = MTDPART_OFS_APPEND,
        .size           = MTDPART_SIZ_FULL,
    }
};

```

注意：

这个结构体把 NandFlash 分成了四个分区，我们 2012-09 发布的文件系统里面，`/etc/init.d/rcS` 脚本把第四个分区挂载到了 `/mnt` 目录下面，您可根据您的需求，如果您的 NandFlash 仍然使用三个分区，可以不更改这个文件。如果更改了这个文件，采用 4 个分区，不要忘了更改 `/etc/init.d/rcS` 文件，把第四个分区挂载到 `/mnt` 目录下面。

更改之前请确认一下您的文件系统里面 `/etc/init.d/rcS` 文件 有没有挂载第四个分区，或者执行 `mount` 命令看看您目前的系统是三个分区还是四个分区，根据您的具体情况来修改。

[Linux-3.0.1/arch/arm/plat-samsung/dev-nand.c](#)

Nandflash 设备描述信息。

[Linux-3.0.1/drivers/mtd/nand/nand_base.c](#)

NandFlash 驱动文件，实现基本的读写操作

[Linux-3.0.1/drivers/mtd/nand/nand_ids.c](#)

NandFlash ID 信息描述文件,与 Uboot 代码中的 `nand_ids.c` 文件类似，但不相同。

[Linux-3.0.1/drivers/mtd/nand/s3c_nand.c](#)

NandFlash 驱动文件，实现 NandFlash 驱动的注册，NandFlash 设备扫描等重要。

[Linux-3.0.1/include/mtd/mtd-abi.h](#)

NandFlash 驱动头文件

注意：修改文件后，重新编译前,需要执行 `make clean` 命令

文件系统：

文件系统部分不需要任何的改动，使用 2G 版本的 Yaffs2 的制作工具 `mkyaffs2image-256DDR` 即可制作适用 4G NandFlash 的 `rootfs.yaffs2`。

注意：完成以上操作，系统成功运行后可以在超级终端或者 DNW 中执行 `df -h` 命令，此时可以看到您的 NandFlash 空间大小了。

Wince6.0

由于只针对 Nand 进行了升级，所以可以将原有(2G nand 版本 BSP)BSP 的修改内容，在最新(4G nand 版本 BSP)BSP 上重新移植一遍即可。不用考虑其他更改。