

SD Partition MAP 분석

<http://www.mangoboard.com/>

<http://cafe.naver.com/embeddedcrazyboys>

Crazy Embedded Laboratory

Document History

Revision	Date	Change note

1. S5PV310 SD Partition MAP 분석.....4

1. S5PV310 SD Partition MAP 분석

S5PV310 소스에서 u-boot 소스에서

"/arch/arm/include/asm/arch-exynos/movi_partition.h"

코드와

```
#ifndef __MOVI_PARTITION_H_
#define __MOVI_PARTITION_H_

#define eFUSE_SIZE          (1 * 512)      // 512 Byte eFuse, 512 Byte reserved

#define MOVI_BLKSIZE        (1<<9) /* 512 bytes */

/* partition information */
#if defined(CONFIG_SECURE_BOOT) || defined(CONFIG_SECURE_BL1_ONLY)
#define PART_SIZE_FWBL1     (8 * 1024)
#else
#define PART_SIZE_FWBL1     0
#endif
#define PART_SIZE_BL1       (16 * 1024)
#ifdef CONFIG_TRUSTZONE
#define PART_SIZE_UBOOT     (328 * 1024)
#else
#define PART_SIZE_UBOOT     (512 * 1024)
#endif
#define PART_SIZE_KERNEL    (5 * 1024 * 1024)// (5 * 1024 * 1024)//CRZ 4->5MB
#define PART_SIZE_ROOTFS    (26 * 1024 * 1024)
#define PART_SIZE_TZSW      (160 * 1024)

#define MOVI_FWBL1_BLKCNT   (PART_SIZE_FWBL1 / MOVI_BLKSIZE)
#define MOVI_BL1_BLKCNT    (PART_SIZE_BL1 / MOVI_BLKSIZE)
#define MOVI_ENV_BLKCNT     (CONFIG_ENV_SIZE / MOVI_BLKSIZE) /* 16KB */
#define MOVI_UBOOT_BLKCNT   (PART_SIZE_UBOOT / MOVI_BLKSIZE) /* 328KB */
#define MOVI_ZIMAGE_BLKCNT (PART_SIZE_KERNEL / MOVI_BLKSIZE) /* 4MB */
#define MOVI_ROOTFS_BLKCNT  (PART_SIZE_ROOTFS / MOVI_BLKSIZE) /* 26MB */
#define MOVI_TZSW_BLKCNT    (PART_SIZE_TZSW / MOVI_BLKSIZE) /* 160KB */

#define MOVI_UBOOT_POS      ((eFUSE_SIZE / MOVI_BLKSIZE) + MOVI_FWBL1_BLKCNT +
MOVI_BL1_BLKCNT)
```

```
"arch/arm/cpu/armv7/exynos/movi_partition.c"
```

```
init_raw_area_table 함수를 참조
```

```
[icanjji@crz-server113 u-boot]$ grep -ri init_raw_area_table *
arch/arm/cpu/arm11/s5p6450/movi_partition.c:int init_raw_area_table(block_dev_desc_t * dev_desc, int
location)
arch/arm/cpu/armv7/s5pv210/movi_partition.c:int init_raw_area_table(block_dev_desc_t * dev_desc, int
location)
arch/arm/cpu/armv7/exynos/movi_partition.c:int init_raw_area_table(block_dev_desc_t * dev_desc, int
location)
Binary file arch/arm/cpu/armv7/exynos/libexynos.o matches
Binary file arch/arm/cpu/armv7/exynos/movi_partition.o matches
arch/arm/cpu/armv7/s5pv310/movi_partition.c:int init_raw_area_table(block_dev_desc_t * dev_desc, int
location)
build.out:cmd_movi.c:115: warning: implicit declaration of function 'init_raw_area_table'
Binary file common/cmd_movi.o matches
Binary file common/libcommon.o matches
common/cmd_movi.c:      init_raw_area_table(&mmc->block_dev, location);
drivers/mmc/mmc.c:extern int init_raw_area_table (block_dev_desc_t * dev_desc, int location);
drivers/mmc/mmc.c:      init_raw_area_table(&mmc->block_dev, 0);
Binary file drivers/mmc/mmc.o matches
Binary file drivers/mmc/libmmc.o matches
System.map:c3e05f74 T init_raw_area_table
Binary file u-boot matches
u-boot.map:          0xc3e05f74          init_raw_area_table
```

```
[icanjji@crz-server113 u-boot]$ fin "*.c" get_mmc_part_info
int get_mmc_part_info(char *device_name, int part_num, int *start, int *count, unsigned
    get_mmc_part_info(dev_num, 2, &start, &count, &pid);
    get_mmc_part_info(dev_num, 3, &start, &count, &pid);
    get_mmc_part_info(dev_num, 4, &start, &count, &pid);
```

```

get_mmc_part_info(dev_num, 1, &start, &count, &pid);
./common/cmd_fastboot.c
int get_mmc_part_info(char *device_name, int part_num, unsigned long long *block_start,
ock_count, unsigned char *part_Id)
    rv = get_mmc_part_info(argv[2], 1, &(partInfo[0].block_start), &(partInfo[0].bl
    rv = get_mmc_part_info(argv[2], 2, &(partInfo[1].block_start), &(partInfo[1].bl
    rv = get_mmc_part_info(argv[2], 3, &(partInfo[2].block_start), &(partInfo[2].bl
    rv = get_mmc_part_info(argv[2], 4, &(partInfo[3].block_start), &(partInfo[3].bl
./common/cmd_mmc_fdisk.c

```

PARTITION	START SECTOR	END SECTOR	SECTOR COUNT	SIZE (BYTE)
MBR	0	1	1	512
BL1	1	33	32	16384
U-BOOT	33	1057	1024	524288
TRUSTZONE	1057	1377	320	163840
U-BOOT ENV	1377	1409	32	16384
KERNEL	1409	11649	10240	5242880
RAMDISK	11649	64897	53248	27262976
RESERVED	64897	137160	72263	36998656
SYSTEM	137160	762000	624840	319918080
DATA	762000	1386840	624840	319918080
CACHE	1386840	2011680	624840	319918080
FAT32	2011680	15407640	13395960	6858731520