

U-Boot MTD Block Device Design

发布版本：1.3

作者邮箱：jason.zhu@rock-chips.com

日期：2019.08

文件密级：内部资料

前言

概述

U-Boot 下 MTD block device 设计介绍。

读者对象

本文档（本指南）主要适用于以下工程师：

技术支持工程师

软件开发工程师

产品版本

修订记录

日期	版本	作者	修改说明
2019-05-20	V1.0	Jason Zhu	初始版本
2019-06-18	V1.1	Jason Zhu	修改分区支持,更新 step by step 章节
2019-08-27	V1.2	Jason Zhu	更正 config 错误
2019-08-27	V1.3	Jason Zhu	增加 SPL MTD block 说明

U-Boot MTD Block Device Design

1 引用参考

2 术语

3 简介

4 设计

4.1 MTD block 设计

4.2 多设备挂接设计

4.3 分区表设计

4.4 新增 CONFIG

4.5 驱动挂接框图

4.6 SPL MTD block 设计

5 step by step

5.1 U-Boot

5.2 SPL

1 引用参考

[1]. 《Rockchip-Developer-Guide-UBoot-nextdev-CN.md》

[2]. 《Rockchip-Developer-Guide-Uboot-mmc-device-driver-analysis.md》

2 术语

MTD : Memory Technology Device 即内存技术设备。

3 简介

设计 MTD block 层，兼容目前 block 设备接口。

4 设计

4.1 MTD block 设计

设计 mtd bread & bwrite & berase 三个函数，通过 desc->devnum 来区分挂载的不同 MTD 设备，这样上层可以直接调用 blk_dread & blk_dwrite & blk_derase 操作 MTD 设备。代码位于 drivers/mtd/mtd_blk.c。

4.2 多设备挂载设计

对于 block 设备，会依据 if_type 与 devnum 来找挂载在 block 设备下的驱动。对于挂载在 block 设备下的 MTD 驱动，定义 if_type 为 IF_TYPE_MTD。devnum 在设备驱动在与 block 层 bind 时传递。
例：

```
static int rockchip_nandc_bind(struct udevice *udev)
{
    ...
    blk_create_devicef(udev, "mtd_blk", "blk", IF_TYPE_MTD,
                      devnum, 512, 0, &bdev);
    ...
}
```

devnum 为不同设备，目前 nand 设备有 nand , spi nand, spi nor 分别为 0 , 1 , 2。

MTD block 不同设备间切换：

```
mtd dev <devnum>
```

读写擦除接口挂载：

```
ulong mtd_dread(struct udevice *udev, lbaint_t start,
                lbaint_t blkcnt, void *dst)
{
    struct blk_desc *desc = dev_get_uclass_platdata(udev);

    if (desc->devnum == BLK_MTD_NAND) {
        /* nand驱动*/
    } else if (desc->devnum == BLK_MTD_SPI_NAND) {
        /* spi nand驱动 */
    } else if (desc->devnum == BLK_MTD_SPI_NOR) {
```

```

        /* spi nor驱动 */
    }
}

ulong mtd_dwrite(struct udevice *udev, lbaint_t start,
                lbaint_t blkcnt, const void *src)
{
    struct blk_desc *desc = dev_get_uclass_platdata(udev);

    if (desc->devnum == BLK_MTD_NAND) {
        /* nand驱动*/
    } else if (desc->devnum == BLK_MTD_SPI_NAND) {
        /* spi nand驱动 */
    } else if (desc->devnum == BLK_MTD_SPI_NOR) {
        /* spi nor驱动 */
    }
}

ulong mtd_derase(struct udevice *udev, lbaint_t start,
                lbaint_t blkcnt)
{
    struct blk_desc *desc = dev_get_uclass_platdata(udev);

    if (desc->devnum == BLK_MTD_NAND) {
        /* nand驱动*/
    } else if (desc->devnum == BLK_MTD_SPI_NAND) {
        /* spi nand驱动 */
    } else if (desc->devnum == BLK_MTD_SPI_NOR) {
        /* spi nor驱动 */
    }
}
}

```

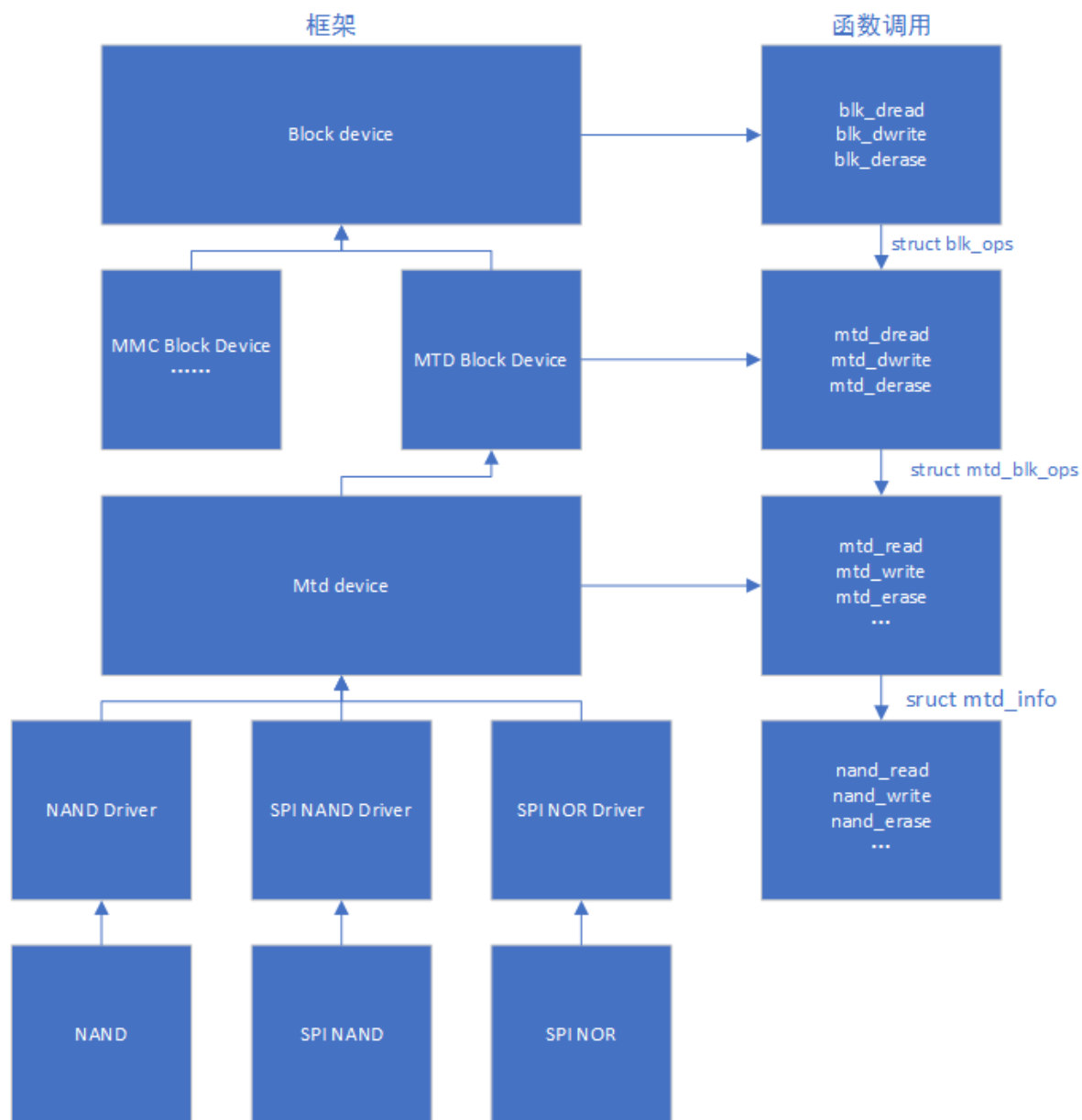
4.3 分区表设计

兼容 GPT 分区表，注意 nand flash 与 spi flash 尾部需要保留 4 个 blocks 用于保存坏块表。

4.4 新增 CONFIG

增加 CONFIG_MTD_BLK、CONFIG_CMD_MTD_BLK，支持 mtd block device。

4.5 驱动挂接框图



4.6 SPL MTD block 设计

SPL MTD block 层可以统一 SPL 下的 nand & spi nand & spi nor 的驱动调用，上层有统一的接口读写设备，达到精简代码的效果。

框架代码：

```
./common/spl/spl_mtd_blk.c
./drivers/mtd/mtdcore.c
./drivers/mtd/mtd_blk.c
./drivers/mtd/mtd_uboot.c
./drivers/mtd/mtd-uclass.c
```

config 配置：

```
CONFIG_SPL_MTD_SUPPORT=y
```

分区表支持：

```
CONFIG_SPL_LIBDISK_SUPPORT=y
CONFIG_SPL_EFI_PARTITION=y
CONFIG_PARTITION_TYPE_GUID=y
```

启动顺序配置：

```
// rkxxx-u-boot.dtsi
chosen {
    u-boot,spl-boot-order = &sfc, &nandc, &emmc;
};
```

启动顺序源码：

```
// arch/arm/mach-rockchip/spl-boot-order.c
void board_boot_order(u32 *spl_boot_list)
{
    .....
    boot_device = spl_node_to_boot_device(node);
    .....
}

static int spl_node_to_boot_device(int node)
{
    struct udevice *parent;

    if (!uclass_get_device_by_of_offset(UCLASS_SPI, node, &parent))
        return BOOT_DEVICE_MTD_BLK_SPI_NAND;
    ....
}
```

读接口：

```
unsigned long blk_dread(struct blk_desc *block_dev, lbaint_t start,
                        lbaint_t blkcnt, void *buffer)
```

5 step by step

5.1 U-Boot

1. 对应的 defconfig 添加

```
CONFIG_MTD_BLK=y
CONFIG_CMD_MTD_BLK=y
```

其他 nand 的配置可以参考<https://10.10.10.29/#/c/android/rk/u-boot/+75116/>。

2. 更新支持 mtd 的 laoder , rk3308 补丁地址<https://10.10.10.29/#/c/rk/rkbin/+75644/>。
3. 编译 uboot , 例如编译 rk3308

```
./make.sh rk3308
```

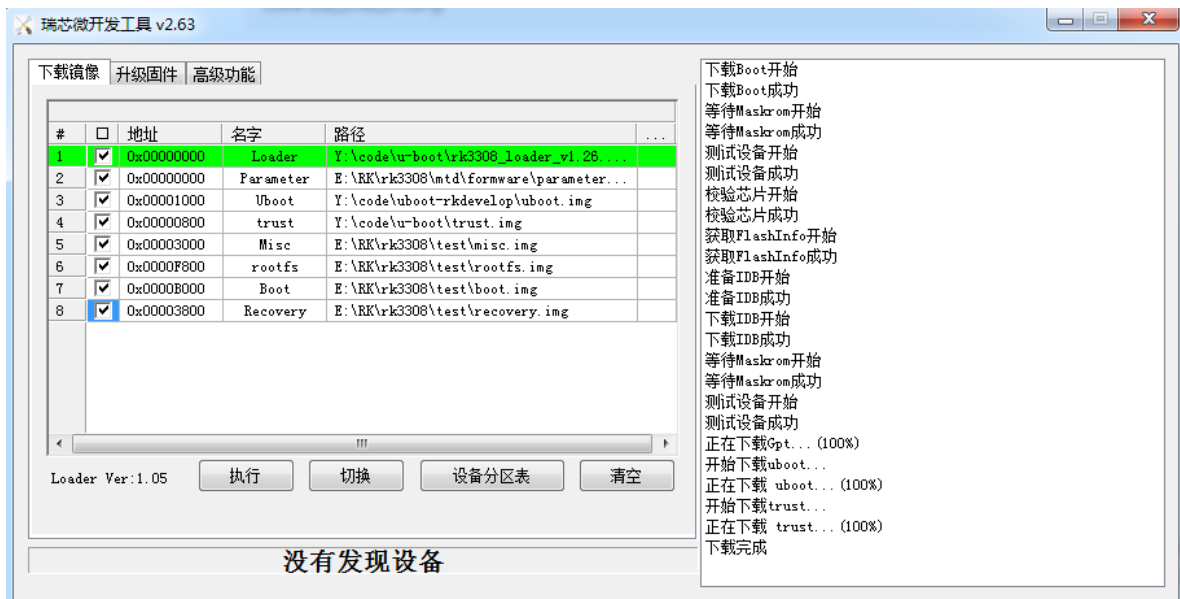
4. 更改支持 GPT 的 parameter.txt , 例如：

```

FIRMWARE_VER:8.1
MACHINE_MODEL:RK3308
MACHINE_ID:007
MANUFACTURER: RK3308
MAGIC: 0x5041524B
ATAG: 0x00200800
MACHINE: 3308
CHECK_MASK: 0x80
PWR_HLD: 0,0,A,0,1
TYPE: GPT
CMDLINE:mtdparts=rk29xxnand:0x00000800@0x00001000(uboot),0x00000800@0x00000800(t
rust),0x00000800@0x00003000(misc),0x00007800@0x00003800(recovery),0x00004800@0x0
00B000(boot),0x00020000@0x0000F800(rootfs),-@0x0002F800(data:grow)

```

5. 烧写固件



6. 成功启动 log

```

.....
U-Boot 2017.09-02976-g47b3c04-dirty (Jun 19 2019 - 17:02:46 +0800)
.....
Device 0: nand_base: Could not find valid JEDEC parameter page; aborting //正常错误打印
Vendor: 0x2207 Rev: V1.00 Prod: MTD //MTD设备初始化
Type: Hard Disk
Capacity: 255.5 MB = 0.2 GB (523264 x 512)
... is now current device
Bootdev: mtd 0 //Bootdev为MTD设备
PartType: EFI //使用GPT分区
.....
Starting kernel ...
.....

```

```
[ 0.000000] kernel command line: storagemedia=mtd
androidboot.storagemedia=mtd androidboot.mode=normal mtdparts=rk-
nand:0x200000@0x400000(uboot),0x200000@0x600000(trust),0x100000@0x800000(misc),0
xc00000@0x900000(recovery),0x900000@0x1500000(boot),0x2a00000@0x1e00000(rootfs),
0x1a00000@0x4800000(oem),-@0x6200000(userdata:grow) androidboot.slot_suffix=
androidboot.serialno=c3d9b8674f4b94f6 rootwait
earlycon=uart8250,mmio32,0xff0c0000 swiotlb=1 console=ttyFIQ0 ubi.mtd=5
root=ubi0:rootfs rootfstype=ubifs snd_aloop.index=7 //mtdparts为调整过的分区表,
单位为Byte
//ubi.mtd指定分区中rootfs的位置
.....
```

5.2 SPL

1.config 配置, 参考<https://10.10.10.29/#/c/android/rk/u-boot/+/79335/>

2.uboot 编译, 例如编译 rk3308

```
./make.sh rk3308
```

3.编译 pre-loader

```
./make.sh spl-s ../rkbin/RKBOOT/RK3308MINIALL_WO_FTL.ini
```

4.下载编译参考上章节