

TPK AUR3852 固件升级调试总结

李东汉

ldh@rock-chips.com

一、前言.....	1
二、概念.....	1
三、FLASH 分布结构.....	2
四、固件升级流程.....	3
五、代码分析.....	4
1.主要的函数原型及其实现功能.....	4
2.升级函数.....	4
六、小结.....	7

一、前言

前段时间和 TPK 的工程师调试 TP 的固件升级功能，有点收获，同时发现其他 TP 的固件升级也是大同小异，在这里将一些过程和调试的心得写出来，希望能对大家以后调试此类问题有所帮助。

二、概念

TP 固件在线升级是指将 TP 原厂调试好的固件编译进我们的平台，在我们机器起来的时候通过与 TP 的通信将原厂的固件升级到其 IC 中。这样做的好处是 TP 原厂调试好 TP 的效果后不需要一个个地去通过他们的工具给 TP 烧写固件，相当于和我们发布的机器固件放在一起，机器起来即可以自动升级。

升级，其实是将固件文件写入到 TP 模组内置的 FLASH 存储设备中（一般用的是 NORFLASH），FLASH 的读写是以页为单位，擦除是以块为单位，一个块中包含若干个页，重要的一点是 FLASH 在写之前一定要先擦除。

三、FLASH 分布结构

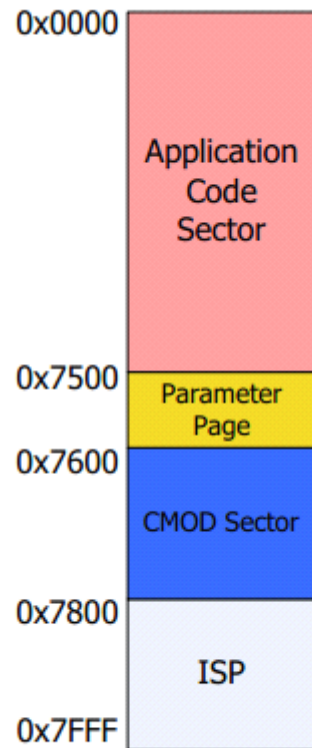
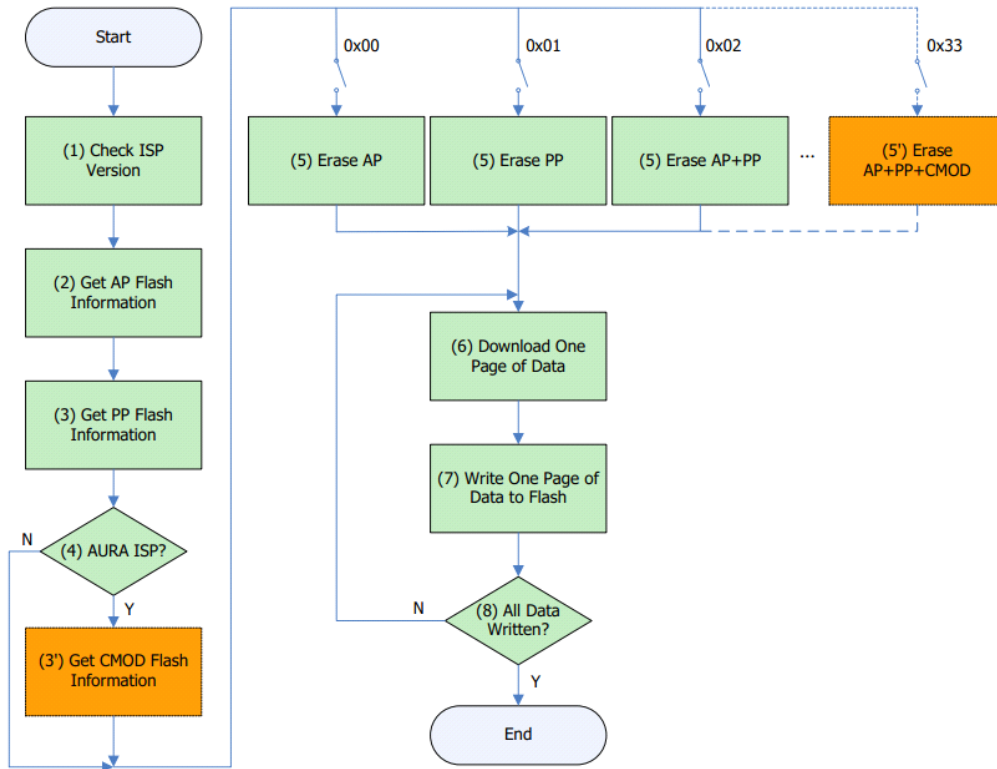


图 1: A3852 新的记忆体布局

有上图我们可以很清晰的看到 TP 内置的 FLASH 使用分布图，我只需要关心的就是上面说的固件是对应上图的 AP 和 PP 区（即地址 0x0000 到 0x75ff），后续的升级其实就是将固件写到该地址区。

四、固件升级流程



因为这个 TP 的固件升级一般情况下都不包括 CMOS 区的升级，所以看此图时可以略去橘黄色的部分，简单的概括就是

(1) 检查当前 TP 里面固件的版本。可以通过其输出的相应字节码判断 TP 里面是否存在固件，固件的版本号，TP 的工作模式。

(2) 获取 AP 区域的信息，包括起止地址，大小

(3) 获取 PP 区域信息同 (2)

(4) 和 CMOS 的升级相关，可以不理睬

(5) 擦除相关的区域

(6) 下载固件数据，此处是将数据写到缓存，还没有真正写到 flash 中

(7) 写固件数据，将固件真正地写到 flash 中

五、代码分析

1.主要的函数原型及其实现功能

- 1) int touchfirmwareupdate(void) //升级入口函数
- 2) static int aur3852_transfer(u8 buf[], u16 flag, unsigned len) //i2c 通信接口函数
buf[]为所要传输的数据，flag 为读写标识，len 为数据长度
- 3) static int Check_ISP_Version(unsigned char *buf)/ /对应固件升级流程中的（1），获取固件版本
- 4) static int Read_flash_info(u8 type,unsigned char *buf)//获取对应固件区域的 FLASH 信息
- 5) static int Erase_flash(u8 type) //擦除接口
- 6) int Download_One_Page(int pagesize,unsigned char *buf)//下载固件
- 7) int Write_One_Page(u8 pageNum)//写固件

2.升级函数

```
int touchfirmwareupdate(void)
{
    static char *buf;
    struct file *fp;
    mm_segment_t fs;
    loff_t pos;
    int code_length;
    int checksumorig;
    int page_size;
    int start_page_num;//开始页地址
    int end_page_num;//结束页地址
    int total_page_num;//结束页地址
    int i;
    int ret;
    unsigned char ver_buf[5];
    unsigned char databuf[5];
    unsigned char *pdata;
    struct flash_info ap_to_pp;
```

```
/*
```

此段代码对应上述的升级流程（1），在这里主要的功能是检查 TP 当前的处在哪种工作模式下，BOOTLOADER 或 AP，升级固件时必须要在 BOOTLOADER 模式下，TP 正常工作时在 AP 模式下。enter_bootloader()和后面的 exit_ISP()分别是进入和退出 BOOTLOADER 模式

```
*/
```

```
    if(Check_ISP_Version(ver_buf)<0)
    {
        goto init;
    }
    if(ver_buf[0] < BL_MODE)
    {
init:
        ret = enter_bootloader();//enter bootloader
        if(ret < 0)
            goto error;
        if(Check_ISP_Version(ver_buf)<0)
        {
            goto error;
        }
        printk("Already entered bootloader mode!\n");
        printk("bootloader version:0x%02x, 0x%02x, 0x%02x\n", ver_buf[0], ver_buf[1],
ver_buf[2]);

        if(ver_buf[2] < 0x05)
        {
            printk("\nThe bootloader version is too old, it can't upgrade firmware!\n");
            goto exit;
        }
    }
    else
    {
        printk("Already entered bootloader mode!\n");
        if(Check_ISP_Version(ver_buf)<0)
        {
            goto error;
        }
        if(ver_buf[2] < 0x05)
        {
            printk("\nThe bootloader version is too old, it can't upgrade firmware!\n");
            goto exit;
        }
    }
}
```

```

/*获取对应固件的 flash 信息*/
//update ap and pp only
    if(Read_flash_info(TYPE_FLASH_APPP,databuf)<0)//((unsigned char*)&ap_to_pp)<0)
    {
        goto error;
    }
//    printk("ap_to_pp.page_size = 0x%04x,ap_to_pp.start_page = 0x02x,ap_to_pp.end_page =
0x02x\n",ap_to_pp.page_size,
//        ap_to_pp.start_page,ap_to_pp.end_page);
    page_size = databuf[1]<<8|databuf[0];//ap_to_pp.page_size;
    start_page_num = databuf[2];//ap_to_pp.start_page;
    end_page_num = databuf[3];//ap_to_pp.end_page;
    total_page_num = end_page_num - start_page_num +1;
    printk("page_size    =    0x%04x,start_page_num    =    0x%02x,end_page_num    =
0x%02x\n",page_size,
        start_page_num,end_page_num);
/*擦除指定的 flash 区域*/
    if(Erase_flash(TYPE_FLASH_APPP)<0)
    {
        goto error;
    }
/*下载和写入固件，这里容易混淆，下面的写循环是从直接从 0 开始到（total_page_num-1）
的，而我们上面获取到的信息 start_page_num 和循环中的 0 是两个不同的概念，
start_page_num 是页编号,起始编号为 1,而 0 是起始的页地址。
*/
    for (i=0;i<total_page_num;i++)
    {
        if(0< Download_One_Page(page_size,aur3852_data+i*page_size))
        {
            goto error;
        }

        if(0< Write_One_Page(i))
        {
            goto error;
        }
        printk("write the %d page ok\n",i);
    }
/*退出 bootloader 模式*/
    ret = exit_ISP();
    if(ret < 0)
        goto error;
    Check_ISP_Version(ver_buf);
//    wake_unlock(&ts_update_wake_lock);

```

```

    printk("touchpanel AUR3852 firmware update OK\n");
    //kfree(buf);
    __gpio_set_value(TOUCH_RESET_PIN,GPIO_LOW);
    msleep(50);
    printk("touchpanel reset after progaram\n");
    __gpio_set_value(TOUCH_RESET_PIN,GPIO_HIGH);
    msleep(500);
    return 0;

exit:
    ret = exit_ISP();
    if(ret < 0)
        goto error;

//    wake_unlock(&ts_update_wake_lock);
//    kfree(buf);
    __gpio_set_value(TOUCH_RESET_PIN,GPIO_LOW);
    msleep(50);
    __gpio_set_value(TOUCH_RESET_PIN,GPIO_HIGH);
    msleep(500);
    return -1;

error:
//    wake_unlock(&ts_update_wake_lock);
//kfree(buf);
    printk("error here\n");
    __gpio_set_value(TOUCH_RESET_PIN,GPIO_LOW);
    msleep(50);
    __gpio_set_value(TOUCH_RESET_PIN,GPIO_HIGH);
    msleep(500);
    return -1;
}

```

六、小结

将上面的东西写出来的初衷不是因为其技术含量有多高，而是觉得此 TP 的升级过程相对于其他 TP 思路步骤很清晰，很具有典型性。希望能对后续其他 TP 的调试会有些许参考价值。