

qspi读写镜像

一.流程说明

- 1.初始化qspi
- 2.从qspi中读取main.img的64字节头部
- 3.分析头部信息，提取main.img镜像长度和最终加载地址等信息
- 4.将main.img从qspi读到指定内存地址
- 5.将main.img从第4步指定的内存地址写到指定的qspi地址中
- 6.将main.img从第5步指定的qspi地址读到第3步分析头部得到的最终加载运行地址中

二.代码

```
/* 步骤4的指定内存地址 */
#define YESHEN_MEM_ADDR 0x30000000

/* 步骤4的初始qspi存放镜像地址 */
#define YESHEN_QSPI_IMAGE_OFFS 0x60000

/* 步骤5的指定qspi地址 */
#define YESHEN_QSPI_IMAGE_OFFS_COPY 0x200000

/* spi flash结构体 */
struct spi_flash {
    struct spi_slave *spi;

    const char *name;

    /* Total flash size */
    u32 size;
    /* Write (page) size */
    u32 page_size;
    /* Erase (sector) size */
    u32 sector_size;

    int (*read)(struct spi_flash *flash, u32 offset,
                size_t len, void *buf);
    int (*write)(struct spi_flash *flash, u32 offset,
                 size_t len, const void *buf);
    int (*erase)(struct spi_flash *flash, u32 offset,
                 size_t len);
```

```

    int      (*poll_read_status)(struct spi_flash *flash);
};

/* 初始化flash */
struct spi_flash *spi_flash_probe(unsigned int bus, unsigned int cs,
    unsigned int max_hz, unsigned int spi_mode);

/* 从flash位置offset位置处开始擦除len字节的长度 */
static inline int spi_flash_erase(struct spi_flash *flash, u32 offset,
    size_t len)
{
    return flash->erase(flash, offset, len);
}

/* 从flash的offset位置处读取长度len字节的数据到buf中 */
static inline int spi_flash_read(struct spi_flash *flash, u32 offset,
    size_t len, void *buf)
{
    return flash->read(flash, offset, len, buf);
}

/* 将buf中的数据读取len字节长度然后写到flash的offset位置处 */
static inline int spi_flash_write(struct spi_flash *flash, u32 offset,
    size_t len, const void *buf)
{
    return flash->write(flash, offset, len, buf);
}

static void yeshen_qspi_load_image_test(void)
{
    struct image_header *header;
    s32 ret;

    /* 初始化qspi */
    struct spi_flash *flash;
    flash = spi_flash_probe(CONFIG_SPL_SPI_BUS, CONFIG_SPL_SPI_CS,
        CONFIG_SF_DEFAULT_SPEED, CONFIG_SF_DEFAULT_MODE);

    if (!flash) {
        puts("SPI probe failed.\n");
        hang_yeshen();
    }

    /* 使用YESHEN_MEM_ADDR作为main.img头部的临时加载地址 */
    header = (struct image_header *) (YESHEN_MEM_ADDR);

    /* 从qspi中读取main.img的64字节头部 */
    ret = spi_flash_read(yeshen_flash, YESHEN_QSPI_IMAGE_OFFSETS, 0x40,
        (void *) header);

    if (ret < 0)

```

```

        yeshen_puts("read 64bytes head wrong\n");

    /* 分析头部信息,提取main.img镜像长度和最终加载地址等信息到spl_image中 */
    spl_parse_image_header(header);
    printf("main.img size: %d\nmain.img load_addr: %p\n",
           spl_image.size, spl_image.load_addr);

    /* step1将main.img从qspi读到YESHEN_MEM_ADDR */
    ret = spi_flash_read(yeshen_flash, YESHEN_QSPI_IMAGE_OFFS,
                         spl_image.size, YESHEN_MEM_ADDR);
    if(ret < 0)
        yeshen_puts("step1 wrong\n");

    /* step2将main.img从ddr的YESHEN_MEM_ADDR地址写到qspi的YESHEN_QSPI_IMAGE_OFFS_COPY地址中 */
    ret = spi_flash_write(yeshen_flash, YESHEN_QSPI_IMAGE_OFFS_COPY,
                          spl_image.size, YESHEN_MEM_ADDR);
    if(ret < 0)
        yeshen_puts("step2 wrong\n");

    /* step3将main.img从qspi的YESHEN_QSPI_IMAGE_OFFS_COPY地址读到ddr最终加载运行地址中 */
    ret = spi_flash_read(yeshen_flash, YESHEN_QSPI_IMAGE_OFFS_COPY,
                         spl_image.size, (void *)spl_image.load_addr);
    if(ret < 0)
        yeshen_puts("step3 wrong\n");
}

```