

目 录

第 1 章 S3C2410A SD/MMC 卡接口电路	1
第 2 章 S3C2410A SD/MMC 卡块驱动	2
2.1 驱动使用说明	2
2.2 SD/MMC 卡块驱动源文件说明	4

第1章 S3C2410A SD/MMC 卡接口电路

由于 S3C2410A 已经提供了 MMC/SD/SDIO 主机控制器接口，因此设计 SD/MMC 卡接口电路时，只须将这些接口相应地接到 SD 卡卡座就可以了，电路如图 1.1 所示。SDATA0 ~ SDATA3、SDCMD 信号线都使用一个电阻上拉至 3.3V，是为了使本电路可以同时兼容 SD 卡与 MMC 卡。

EINT18 信号线用于检测 SD/MMC 卡是否插入，由于卡的插入是随机的，所以使用 S3C2410A 的一个中断引脚 EINT18 来检测（当卡插入时，发生中断）。对于本电路，当卡未完全插入卡座时，nCD-SD 为高电平（被 R13 拉高）；当卡完全插入卡座时，nCD-SD 被卡座的 CARD_INSERT 引脚拉低。因此，卡完全插入卡座时，将发生一个下降沿中断。

GPH8信号线(网络标号为UCLK)用于检测卡是否有写保护，其检测原理和检测卡是否插入的原理相同。对于本电路，卡写保护时卡座输出高电平，否则输出低电平。

注意：对于不同的卡座，卡插入检测电平和写保护检测电平可能有所不同。

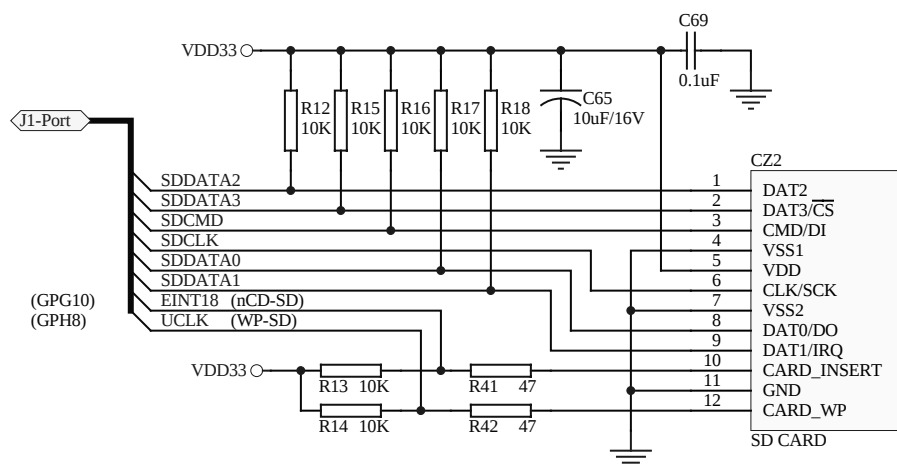


图 1.1 SD 卡接口电路

第2章 S3C2410A SD/MMC 卡块驱动

2.1 驱动使用说明

SD/MMC 卡是一种大容量（最大可达 4GB）、性价比高、体积小、访问接口简单的存储卡。SD/MMC 卡大量应用于数码相机、MP3、手机、大容量存储设备，做为这些便携式设备的存储载体，它还具有低功耗、非易失性、保存数据无需消耗能量等特点。

SD 卡接口向下兼容 MMC（MutliMediaCard 多媒体卡）卡，访问 SD 卡的 SD 协议或 SPI 协议及部分命令也适用于 MMC 卡。

S3C2410A 微控制器内嵌了 MMC/SD/SDIO 主机控制器，只需要在微控制器外部外接少量器件就可以读写 SD/MMC 卡。在 MagicARM2410 实验箱上，S3C2410A 与 SD/MMC 卡座的电路原理图请见图 1.1。

关于 SD/MMC 卡相关规范、读写时序、块驱动的更加详细的说明请见光盘《S3C2410A SD/MMC 卡块驱动使用手册》。

在 Linux 操作系统下，SD/MMC 卡块驱动的软件包括两个模块，如表 2.1 所示。

表 2.1 SD/MMC 卡驱动模块

模块	简介
zlg_fs.ko	块设备通用驱动（支持分区）模块，是 SD 卡、NandFlash 等其它块设备驱动的基础。
mmc2410.ko	S3C2410A SD/MMC 卡底层读写模块，不是驱动程序，不能被 Linux 直接识别，使用时必须先加载 zlg_fs。

这两个模块是上层与下层的关系，它们之间的调用关系如图 2.1 所示。Linux 内核只与 zlg_fs 打交道，zlg_fs 则调用 mmc2410 在底层完成 SD/MMC 卡的访问操作。

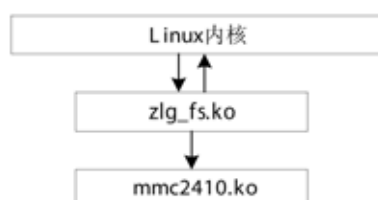


图 2.1 SD/MMC 卡块驱动各模块之间的关系

zlg_fs 模块对应的源文件及简单介绍如表 2.2 所示，mmc2410 模块对应的源文件及简单介绍如表 2.3 所示。

表 2.2 zlg_fs 模块对应的源文件

源文件	简介
zlg_fs.c	块设备通用驱动（支持分区）接口，是 SD 卡、NandFlash 等其它块设备驱动的基础。

表 2.3 mmc2410 模块对应的源文件

源文件	简介
sd.c	zlg_fs.ko 模块与 mmc2410.ko 模块之间的接口。
sddriver.c	为 sd.c 提供初始化 SD/MMC 卡、读写 SD/MMC 卡扇区的函数。
sdcmd.c	SD/MMC 卡命令层，实现卡的各种命令以及主机与卡之间的数据流控制。
sdhal.c	硬件抽象层，包括相关硬件、寄存器初始化，设置读写 SD/MMC 卡时钟等函数。

如果将图 2.1 细化到源文件，就可以清晰地看到各个源文件之间的关系了，如图 2.2 所示。从该图可见，sd.c 在 zlg_fs.c 与 sddriver.c 之间的起到桥梁作用，这种结构使能 zlg_fs.c 与设备无关。

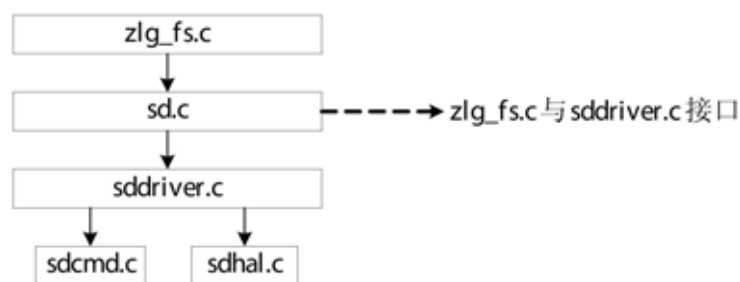


图 2.2 SD/MMC 卡驱动对应各源文件之间的关系

在使用 SD/MMC 卡块驱动时，必须先加载 zlg_fs 模块，然后再加载 mmc2410 模块。而卸载顺序则相反，先卸载 mmc2410 模块，再卸载 zlg_fs 模块。

zlg_fs 支持 4 个块设备，分别为 zlg_fsa、zlg_fsb、zlg_fsc 和 zlg_fsd。每个块设备最多支持 16 个分区，设备分区由其后的数字来区分，例如：zlg_fsa 的第 1 个分区为 zlg_fsa1，第 2 个分区为 zlg_fsa2，以此类推，第 15 个分区为 zlg_fsa15。

在使用任何一个设备分区之前，必须为整个设备创建节点，然后再为分区创建节点。例如使用 zlg_fsa1，必须进行如程序清单 2.1 所示的操作。

程序清单 2.1 为块设备创建节点

```
mknod /dev/zlg_fsa b 252 0
mknod /dev/zlg_fsa1 b 252 1
```

(1)

程序清单 2.1 中的 b 表示块设备，252 为主设备号，0 和 1 和从设备号。

节点创建完毕后，还需要将使用的块设备或者分区 mount 到文件系统的某个目录，才能对该块设备（如 SD 卡）进行操作。例如，将程序清单 2.1(1)对应的分区 mount 到/tmp/sd 目录下的指令为：

```
mount /dev/zlg_fsa1 /tmp/sd
```

这样，操作目录/tmp/sd 就是对 zlg_fsa1 进行操作。如果设备没有分区，则直接操作整个设备即可。例如挂载到系统上的第 1 个块设备没有分区，则使用：

```
mount /dev/zlg_fs /tmp/sd
```

命令将整个设备 mount 到/tmp/sd 目录下。

以上过程比较多，为了方便，使用一个 loadsd 文件进行管理。用于加载 SD/MMC 卡块驱动程序和建立相关节点，如程序清单 2.2 所示。

程序清单 2.2 loadsd 文件内容

```
#!/bin/sh

rm -f /dev/zlg_fsa
rm -f /dev/zlg_fsa1
rmmod mmc2410.ko
rmmod zlg_fs.ko
```

```
mknod /dev/zlg_fsa b 252 0
mknod /dev/zlg_fsa1 b 252 1
insmod zlg_fs.ko
insmod mmc2410.ko
```

该文件使用的前提条件是：zlg_fs、mmc2410 和 loadsd 都在同一目录下。

2.2 SD/MMC 卡块驱动源文件说明

下面对 SD/MMC 卡块驱动源文件中的一些重要文件作一些简单的说明。

1. config.h 文件

该文件主要包含了编写 Linux 驱动时所需要的内核头文件。还有就是需要在此文件中编写“SD/MMC 卡块驱动初始化”函数、编写“卸载 SD/MMC 卡块驱动”函数。

SD/MMC 卡块驱动初始化函数请见：static void drive_init(void)；

卸载 SD/MMC 卡块驱动函数请见：static void drive_clean(void)。

2. sdconfig.h 文件

该文件在当前目录的 sdmmc 目录中，该文件定义了 SD/MMC 卡底层读/写的一些宏定义，用于控制 SD/MMC 卡的读写方法及使用到的硬件接口。该文件的主要内容如程序清单 2.3 所示。

程序清单 2.3 sdconfig.h 文件的主要内容

```
/* disable(1) or disable(0) reading or writing by interrupt */
#define SD_INTERRUPT_EN      1          /* 是(1)否(0)使能中断读写 SD/MMC 卡 */      (1)
#define MMC_MMC_CLOCK_HIGH  20000000   /* SD/MMC 卡最高读写速度 20MHz */      (2)
#define SD_BLOCKSIZE        512        /* SD 卡/MMC 卡块的长度 */      (3)
#define SD_BLOCKSIZE_NBITS  9          /* 2 ^ 9 = 512 */
/* define the interrupt of card inserting */      (4)
#define IRQ_nCD_SD           IRQ_EINT18
#define SD_IRQ_CD             IRQ_nCD_SD
#define GPIO_nCD_SD           (GPIO_MODE_IN | GPIO_PULLUP_DIS | GPIO_G10)
#define SD_GPIO_CD            GPIO_nCD_SD
#define SD_CD_LEVEL_LOW      /* 定义卡插入低电平有效 */
```

程序清单 2.3(1)宏定义指定读写 SD/MMC 卡的方式是以中断方式还是查询方式，当宏定义值为 1 时，读写采用中断方式，这种方式对 Linux 的资源占用少，一般采用这种方式。而宏定义值为 0 时，则采用查询方式，这种方式对 Linux 的资源占用较大。本驱动默认采用中断方式。

程序清单 2.3(2)定义了读写 SD/MMC 卡的最高时钟频率，通常 MMC 卡的读写时钟频率不得高于 20MHz，SD 卡则不得高于 25MHz，为了使驱动兼容两者，故定义读写 SD/MMC 卡的时钟频率最高为 20MHz。

程序清单 2.3(3)定义了 SD/MMC 卡一个数据块的大小,一般 SD/MMC 卡块大小都为 512 字节,这里通常不需要改动,而 SD_BLOCKSIZE_NBITS 宏定义与 512 这个值是相对应的,有 $2^9 = 512$ 的关系,所以通常也不需要改动。

程序清单 2.3(4)定义了卡插入检测使用的中断 GPIO 口,这里定义使用了 GPIO 中的 GPG10,如果用户需要改用其它 GPIO 口,只需在这里做修改。

关于本驱动程序的使用方法请见《MagicARM2410 教学实验开发平台实验指导》中的相关章节。

广州致远电子有限公司

www.zyinside.com