SD卡使用手册

# 第1章 SD 软件包使用手册

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 机、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SPI 协议及部分命令也适用于MMC 卡。

## 1.1 SD/MMC 卡的外部物理接口

SD 和MMC 卡的外形和接口触点如图1 所示。其中SD 卡的外形尺寸为：24mm x 32mm x 2.1mm （普通）或24mm x 32mm x 1.4mm （薄SD 存储卡），MMC 卡的外形尺寸为24mm x 32mm x 1.4mm 。

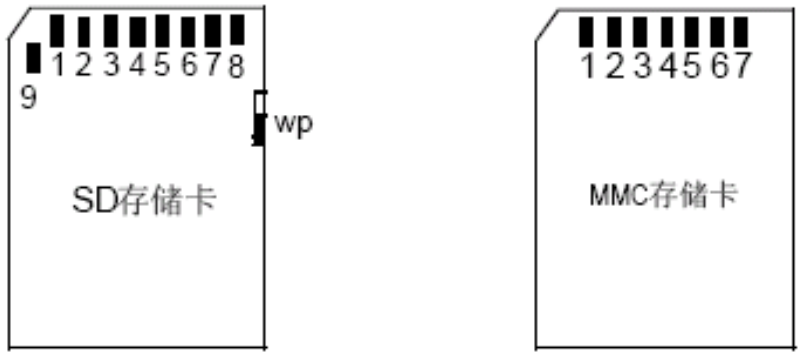


图 1 SD 卡和 MMC 卡的形状和接口（上视图）

表1 为SD/MMC 卡各触点的名称及作用，其中MMC 卡只使用了1 ~ 7 触点。

表1 SD/MMC 卡的焊盘分配

| 引脚 | SD 模式 | | | SPI 模式 | | |
|---|---|---|---|---|---|---|
| | 名称1 | 类型 | 描述 | 名称 | 类型 | 描述 |
| 1 | CD/DAT32 | I/O/PP3 | 卡的检测/数据线[Bit 3] | CS | I | 片选（低电平有效） |
| 2 | CMD | PP4 | 命令／响应 | DI | I5 | 数据输入 |
| 3 | VSS1 | S | 电源地 | VSS | S | 电源地 |
| 4 | VDD | S | 电源 | VDD | S | 电源 |
| 5 | CLK | I | 时钟 | SCLK | I | 时钟 |
| 6 | VSS2 | S | 电源地 | VSS2 | S | 电源地 |
| 7 | DAT0 | I/O/PP | 数据线[Bit 0] | DO | O/PP | 数据输出 |
| 8 | DAT1 | I/O/PP | 数据线[Bit 1] | RSV | | |
| 9 | DAT2 | I/O/PP | 数据线[Bit 2] | RSV | | |

注：1. S：电源；I：输入；O：推挽输出；PP：推挽I/O。

2. 扩展的DAT 线（DAT1 ~ DAT3 ）在上电后处于输入状态。它们在执行SET_BUS_WIDTH 命令后作为DAT 线操作。当不使用DAT1 ~ DAT3 线时，主机应使自己的DAT1~DAT3 线处于输入模式。

这样定义是为了与MMC 卡保持兼容。

3. 上电后，这条线为带50KΩ上拉电阻的输入线（可以用于检测卡是否存在或选择SPI 模式）。用户可以在正常的数据传输中用SET_CLR_CARD_DETECT（ACMD42 ）命令断开上拉电阻的连接。MMC 卡的该引脚在SD 模式下为保留引脚，在SD 模式下无任何作用。

4. MMC 卡在SD 模式下为：I/O/PP/OD。

5. MMC 卡在SPI 模式下为：I/PP。

由表1 可见，SD 卡和MMC 卡在不同的通信模式下，各引脚的功能也不相同。这里的通信模式是指微控制器（主机）访问卡时使用的通信协议，分为两种：SD 模式及SPI 模式。

在具体通信过程中，主机只能选择其中一种通信模式。通信模式的选择对于主机来说

是透明的。卡将会自动检测复位命令的模式（即自动检测复位命令使用的协议），而且要求

以后双方的通信都按相同的通信模式进行。所以，在只使用一种通信模式的时候，无需明

白另一种模式。下面先简单介绍这两种模式。

### 1.1.1 SD 模式

在SD 模式下，主机使用SD 总线访问SD 卡，其拓朴结构如图2 所示。由图可见，SD 总线上不仅可以挂接SD 卡，还可以挂接MMC 卡。

主机

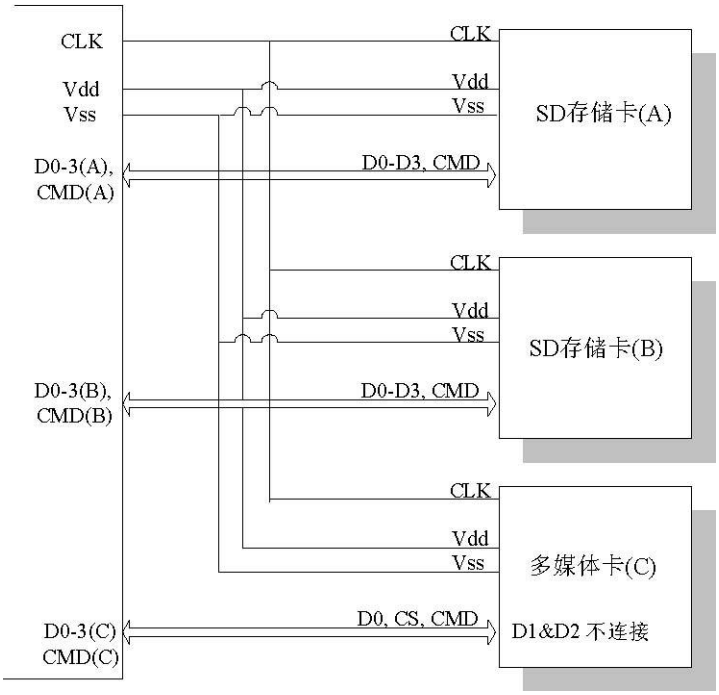


图2 SD 存储卡系统（SD 模式）的总线拓扑结构

SD 总线上的信号线的详细功能描述如表2 所示。

表2 SD 总线信号线功能描述

| 信号线 | 功能描述 |
|---|---|
| CLK | 主机向卡发送的用于同步双方通信的时钟信号 |
| CMD | 双向的命令／响应信号 |
| DAT0 ~ DAT3 | 4 个双向的数据信号（MMC 卡只有DAT0 信号线） |
| VDD | 电源正极，一般电压范围为2.7 ~ 3.6V |
| VSS1、VSS2 | 电源地 |

SD 存储卡系统（SD 模式）的总线拓扑结构为：一个主机（如微控制器）、多个从机（卡）和同步的星形拓扑结构（参考图2）。所有卡共用时钟CLK、电源和地信号。而命令线（CMD）和数据线（DAT0 ~ DAT3 ）则是卡的专用线，即每张卡都独立拥有这些信号线。请注意，MMC 卡只能使用1 条数据线DAT0 。

### 1.1.2 SPI 模式

在SPI 模式下，主机使用SPI 总线访问卡，当今大部分微控制器本身都带有硬件SPI 接口，所以使用微控制器的SPI 接口访问卡是很方便的。微控制器在卡上电后的第1 个复位命令就可以选择卡进入SPI 模式或SD 模式，但在卡上电期间，它们之间的通信模式不能更改为SD 模式。

卡的SPI 接口与大多数微控制器的SPI 接口兼容。卡的SPI 总线的信号线如表3 所示。

表3 SD 卡与MMC 卡的SPI 接口描述

| 信号线 | 功能描述 |
| --- | --- |
| CS | 主机向卡发送的片选信号 |
| CLK | 主机向卡发送的时钟信号 |
| DataIn | 主机向卡发送的单向数据信号 |
| DataOut: | 卡向主机发送的单向数据信号 |

SPI 总线以字节为单位进行数据传输，所有数据令牌都是字节（8 位）的倍数，而且字节通常与CS 信号对齐。SD 卡存储卡系统如图3 所示。

主机

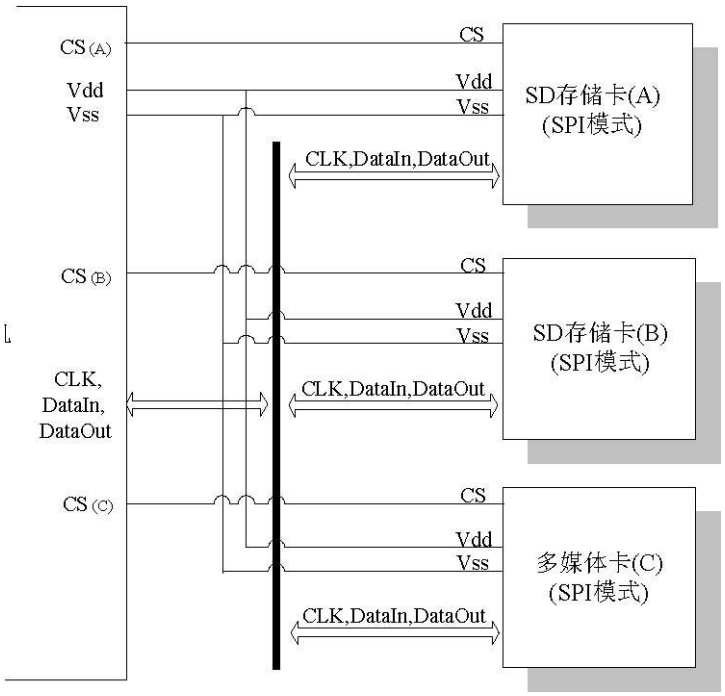


图3 SD 存储卡系统（SPI 模式）的总线拓扑结构

3

当主机外部连接有多张SD 卡或MMC 卡时，主机利用CS 信号线对卡进行寻址。例如：在图3 中，当主机需要向SD 存储卡（A）传输数据或需要从该卡接收数据时，必须将CS(A) 置为低电平（同时其它卡的CS 信号线必须置为高电平）。

CS 信号在SPI 处理（命令、响应和数据）期间必须续持有效（低电平）。唯一例外的情况是在对卡编程的过程。在这个过程中，主机可以使CS 信号为高电平，但不影响卡的编程。

由图3 还可见，当SPI 总线上挂接N 张卡时，需要N 条CS 片选线。

## 1.3 SD 软件包的文件结构及整体构架

本小节介绍本软件包的组成文件以及它们之间的联系。

### 1.3.1 SD 软件包的文件组成

SD 软件包包括的文件如表5 所示。

表5 SD/MMC 卡读写软件包包含的文件

| 文件 | 作用 |
|---|---|
| sdconfig.h | 软件包硬件配置头文件 |
| sdhal.c | 软件包硬件抽象层，实现SPI 接口初始化，SPI 字节的收、发等与SPI 硬件相关的函数 |
| sdhal.h | sdhal.c 头文件 |
| sdcmd.c | 软件包命令层，实现卡的各种命令以及主机与卡之间的数据流控制 |
| sdcmd.h | sdcmd.c 头文件 |
| sddriver.c | 软件包应用层，实现卡的读、写、擦API 函数，该文件还包含一些卡操作函数 |
| sddriver.h | sddriver.c 头文件，包括函数执行错误代码 |
| sdcrc.c | 卡相关的CRC 运算函数。 |
| sdcrc.h | sdcrc.h 头文件 |

以上这些文件构成了本软件包，下面说明由这些文件构成的整体框架。

### 1.3.2 SD 软件包整体框架

考虑到该软件包的可移植性及易用性，将软件包分为3 个层，如图5 所示。图中的实时操作系统并不是必须的，也就是说，本软件包既可以应用于前后台系统（无实时操作系统），也可以应用于实时操作系统中，本软件包提供在前后台系统和实时操作系统μCOS-II 中接口统一的API 函数。

是否使用实时操作系统由本软件包sdconfig.h 文件中的宏定义SD_UCOSII_EN 来使能或禁止。

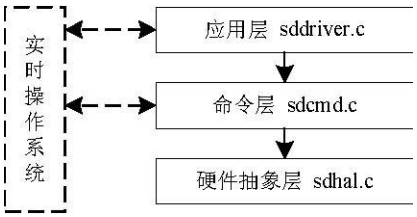


图5 SD/MMC 卡读写软件包结构图

各层的特点如下：

（1）硬件抽象层：读写SD/MMC 卡的硬件条件配置，与硬件相关的函数；

（2）命令层：SD/MMC 卡的相关命令以及卡与主机之间数据流的控制，这一层与实时操作系统相关，与硬件无关；

（3）应用层：向用户应用程序或文件系统提供操作卡的API 函数，这一层由实时操作系统控制。

## 1.4 SD 软件包的硬件配置

SD 软件包的配置只与sdconfig.h 文件相关，配置头文件sdconfig.h 使用户能方便地配置本软件包的相关功能及裁剪某些对用户来说无需用到的函数。该小节提到的所有程序清单都在该文件上。下面阐述该头文件的配置方法。

1. 软件包参数配置

软件包的参数配置如程序清单1 所示，配置选项如下：

（1）是否运行于μCOS-II 中。本软件包既可以运行于前后台系统中，又可以运行于实时操作系统μCOS-II 中。当运行于μCOS-II 中时，宏定义SD_UCOSII_EN 的值应置为1， 否则应置为0。

（2）CRC 校验。由于SD/MMC 卡在SPI 通信模式下可以不需要进行数据传输的CRC 校验，该宏用于使能或禁止本ZLG/SD 软件包的数据传输CRC 校验功能。使能CRC 校验则通信可靠性更高，但CRC 运算也带来传输速度的一些损失，由于本软件包采用查表的方法计算CRC16，所以速度只是略有损失。

（3）SPI 时钟频率。定义SPI 总线的CLK 线的频率，该频率值用于计算读、写、擦操作中的超时时间对应的CLK 个数，这样就将超时时间转换为超时计数。该频率值的单位为：Hz，该值需要用户定义，定义的方法见下面介绍（4. 设置SPI 接口的时钟频率小于400kHz）。

（4）SD/MMC 卡块的长度。定义SD/MMC 卡块的最大长度，当今流行的SD/MMC 卡块的最大长度大部分都支持512 字节。宏定义SD_BLOCKSIZE_NBITS 值为9，对应于$2^9$ = 512 字节（对应于宏定义SD_BLOCKSIZE 的值），SD_BLOCKSIZE_NBITS 与SD_BLOCKSIZE 一定要有这样的对应关系。SD_BLOCKSIZE_NBITS 参数用于固件程序数据计算的方便。用户一般无须改动这两个宏定义的值。

/* 下面函数不常用,如果用户不需要,可置为 0 裁剪指定函数 */

```
#define  SD_ReadMultiBlock_EN      0      /* 是否使能读多块函数 */
#define  SD_WriteMultiBlock_EN     0      /* 是否使能写多块函数 */
#define  SD_EraseBlock_EN          0      /* 是否使能擦卡函数 */
#define  SD_ProgramCSD_EN          0      /* 是否使能写CSD 寄存器函数 */
#define  SD_ReadCID_EN             0      /* 是否使能读CID 寄存器函数 */
#define  SD_ReadSD_Status_EN       0      /* 是否使能读SD Status 寄存器函数 */
#define  SD_ReadSCR_EN             0      /* 是否使能读SCR 寄存器函数 */
```

4. 设置SPI 接口的时钟频率小于400kHz

该函数主要是在SD/MMC 卡初始化阶段，用于设置SPI 接口的时钟频率小于400kHz， 因为MMC 卡在初始化期间SPI 总线的时钟频率不能高于400kHz ，这样本软件包才能达到兼容MMC 卡的目的。在sdconfig.h中定义

```
#define SPI_CLOCK               400000    //400KHZ
```

如果使用SD卡不使用MMC卡可以定义到更高,由于altera提供的SPI核不能通过软件修改时钟频率，所以如果要定义得更高需要到SOPC BUILDER中重新定义时钟。

需要注意，当今流行的SD/MMC 卡的SPI 接口的时钟频率一般不允许超过25MHz ，所以在定义MCU 访问SD/MMC 卡的时钟频率时，必须注意这一点。

### 1.4.2 SD 软件包提供的API 函数

用户可以利用本软件包提供的API 函数对SD/MMC 卡进行访问，见表6 至表11。

表6 SD_Initialize()

| | |
|---|---|
| 函数名称 | SD_Initialize |
| 函数原型 | INT8U SD_Initialize(void) |
| 功能描述 | 初始化SD/MMC 卡、设置块大小为512 字节，获取卡的相关信息 |
| 函数参数 | 无 |
| 函数返回值 | SD_NO_ERR：初始化成功； > 0：初始化失败（错误码，见表12） |
| 特殊说明和注意点 | 该函数设置了卡的读/写块长度为512 字节 |

表7 SD_ReadBlock ()

| | |
|---|---|
| 函数名称 | SD_ReadBlock |
| 函数原型 | INT8U SD_ReadBlock(INT32U blockaddr, INT8U *recbuf) |
| 功能描述 | 读SD/MMC 卡的一个块 |
| 函数参数 | blockaddr：以块为单位的块地址。例如，卡开始的0 ~ 511 字节为块地址0，512 ~ 1023 字节的块地址为1 recbuf： 接收缓冲区，长度固定为512 字节 |
| 函数返回值 | SD_NO_ERR：读成功； > 0：读失败（错误码，见表12） |
| 特殊说明和注意点 | recbuf 的长度必须是512 字节 |

表8 SD_WriteBlock()

| | |
|---|---|
| 函数名称 | SD_WriteBlock |
| 函数原型 | INT8U SD_WriteBlock(INT32U blockaddr, INT8U *sendbuf) |
| 功能描述 | 写SD/MMC 卡的一个块 |
| 函数参数 | blockaddr：以块为单位的块地址。例如，卡开始的0 ~ 511 字节为块地址0，512 ~ 1023 字节的块地址为1 sendbuf： 发送缓冲区，长度固定为512 字节 |
| 函数返回值 | SD_NO_ERR：写成功； > 0：写失败（错误码，见表12） |
| 特殊说明和注意点 | sendbuf 的长度必须是512 字节 |

表9 SD_ReadMultiBlock()

| | |
|---|---|
| 函数名称 | SD_ReadMultiBlock |
| 函数原型 | INT8U SD_ReadMultiBlock(INT32U blockaddr, INT32U blocknum, INT8U *recbuf) |
| 功能描述 | 读SD/MMC 卡的多个块 |
| 函数参数 | blockaddr：以块为单位的块地址blocknum：块数recbuf：接收缓冲区，长度为512 * blocknum 字节 |
| 函数返回值 | SD_NO_ERR：读成功； > 0：读失败（错误码，见表12） |
| 特殊说明和注意点 | 使用时必须将sdconfig.h 中的宏定义值SD_ReadMultiBlock_EN 置为 1 |

表10 SD_WriteMultiBlock ()

| | |
|---|---|
| 函数名称 | SD_WriteMultiBlock |
| 函数原型 | INT8U SD_WriteMultiBlock(INT32U blockaddr, INT32U blocknum, INT8U *sendbuf) |
| 功能描述 | 读SD/MMC 卡的多个块 |
| 函数参数 | blockaddr：以块为单位的块地址blocknum：块数sendbuf：发送缓冲区，长度为512 * blocknum 字节 |
| 函数返回值 | SD_NO_ERR：写成功； > 0：写失败（错误码，见表12） |
| 特殊说明和注意点 | 使用时必须将sdconfig.h 中的宏定义值SD_WriteMultiBlock_EN 置为 1 |

表11 SD_EraseBlock()

| | |
|---|---|
| 函数名称 | SD_EraseBlock |
| 函数原型 | INT8U SD_EraseBlock(INT32U startaddr, INT32U blocknum) |
| 功能描述 | 擦除SD/MMC 卡的多个块 |
| 函数参数 | startaddr：以块为单位的块擦除起始地址blocknum：块数（取值范围1 ~ sds.block_num） |
| 函数返回值 | SD_NO_ERR：擦除成功； > 0：擦除失败（错误码，见表12） |
| 特殊说明和注意点 | 使用时必须将sdconfig.h 中的宏定义值SD_EraseBlock_EN 置为 1 。Startaddr 和 blocknum 建议为 sds.erase_unit 的整数倍，因为有的卡只能以 sds.erase_unit 为单位进行擦除 |

其它函数不常用，这里就不一一列出了。需要用到其它函数的读者可以阅读源码中的函数说明。表6 至表11 函数返回值所代表的含义如表12 所示。

表12 错误代码列表

| 错误码宏定义 | 宏定义值 | 含义 |
|---|---|---|
| SD_NO_ERR | 0x00 | 函数执行成功 |
| SD_ERR_NO_CARD | 0x01 | 卡没有完全插入到卡座中 |
| SD_ERR_USER_PARAM | 0x02 | 用户使用API 函数时，入口参数有错误 |
| SD_ERR_CARD_PARAM | 0x03 | 卡中参数有错误（与本软件包不兼容） |
| SD_ERR_VOL_NOTSUSP | 0x04 | 卡不支持3.3V 供电 |
| SD_ERR_OVER_CARDRANGE | 0x05 | 操作超出卡存储器范围 |
| SD_ERR_UNKNOWN_CARD | 0x06 | 无法识别卡型 |
| SD_ERR_CMD_RESPTYPE | 0x10 | 命令类型错误 |
| SD_ERR_CMD_TIMEOUT | 0x11 | 命令响应超时 |
| SD_ERR_CMD_RESP | 0x12 | 命令响应错误 |
| SD_ERR_DATA_CRC16 | 0x20 | 数据流CRC16 校验不通过 |
| SD_ERR_DATA_START_TOK | 0x21 | 读单块或多块时，数据开始令牌不正确 |
| SD_ERR_DATA_RESP | 0x22 | 写单块或多块时，卡数据响应令牌不正确 |
| SD_ERR_TIMEOUT_WAIT | 0x30 | 写或擦操作时，发生超时错误 |
| SD_ERR_TIMEOUT_READ | 0x31 | 读操作超时错误 |
| SD_ERR_TIMEOUT_WRITE | 0x32 | 写操作超时错误 |
| SD_ERR_TIMEOUT_ERASE | 0x33 | 擦除操作超时错误 |
| SD_ERR_TIMEOUT_WAITIDLE | 0x34 | 初始化卡时，等待卡退出空闲状态超时错误 |
| SD_ERR_WRITE_BLK | 0x40 | 写块数据错误 |
| SD_ERR_WRITE_BLKNUMS | 0x41 | 写多块时，想要写入的块与正确写入的块数不一致 |
| SD_ERR_WRITE_PROTECT | 0x42 | 卡外壳的写保护开关打在写保护位置 |
| SD_ERR_CREATE_SEMSD | 0xA0 | 创建访问卡的信号量失败 |