AN-1443
应用笔记

One Technology Way • P.O. Box 9106 • Norwood, MA 02062-9106, U.S.A. • Tel: 781.329.4700 • Fax: 781.461.3113 • www.analog.com/cn

通过ADuCM3027/ADuCM3029 SPI总线与SD卡接口

简介

安全数字(SD)卡是便携式设备和嵌入式系统中最常用的闪存器件。SD卡兼具尺寸小、功耗低、使用简单和成本低等优点，堪称理想的存储解决方案。SD卡与大多数设备兼容，因此可以使用任何计算机轻松访问和获取卡中的数据，以便根据应用进一步处理。

本应用笔记将讨论如何使用ADuCM3027/ADuCM3029处理器的串行外设接口(SPI)与SD卡进行接口。

本应用笔记将详细介绍有关SD卡的一般信息（例如，引脚排列、通信标准和设计注意事项），还将讨论SPI基础知识（例如，信号和波形）以及ADuCM3027/ADuCM3029SPI接口的特性。

所需电路板

图1. 所需电路板：EVAL-ADuCM3029 EZ-KIT评估板（左侧），与Arduino兼容的SD卡扩展板（右上方），SD卡（右下方）

目录

修订历史

2017年7月—修订版0：初始版

SD卡

SD卡由SD协会(SDA)于1999年推出，旨在扩展用于便携式设备的多媒体卡(MMC)的能力。从那时起，SD卡便得到广泛应用并已成为行业标准，现在大多数便携式设备都使用SD卡来存储图片、音乐等文件。

SD卡由引脚接口、存储内核、内部寄存器和内部控制器组成，如图2所示。存储内核用于存储数据，具有从1 MB到2 TB的不同存储容量。内部寄存器用于存储SD卡状态。引脚接口用于连接SD卡与使用SD卡的主器件（通常是微控制器）。

卡接口控制器负责管理SD卡的存储内核。该控制器通常处理闪存中数据的写入、读取和擦除操作、错误处理以及闪存耗损均衡。因此，实现SD卡的主微控制器只需通过一系列数据包将命令和数据发送到卡接口控制器即可，而不必管理存储内核。

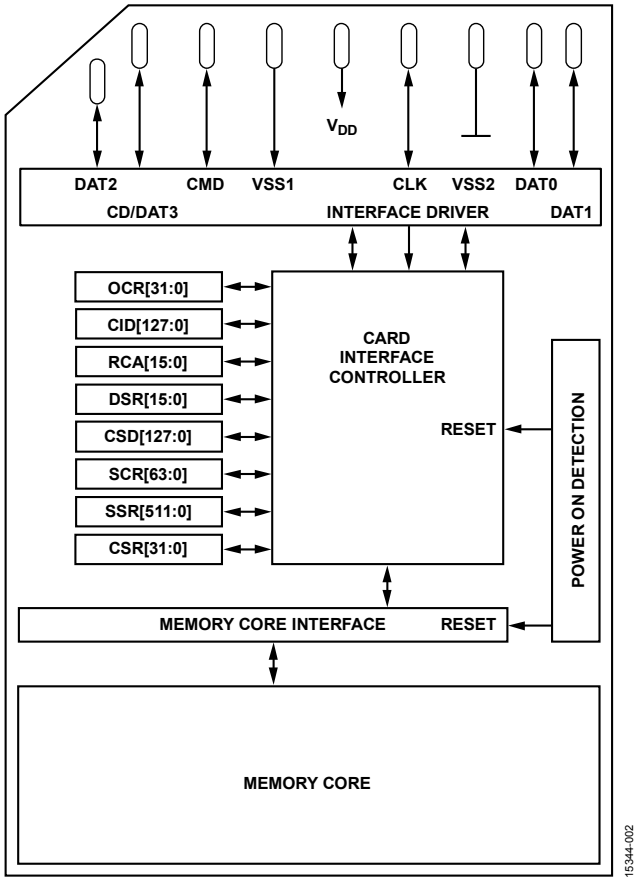

图2. SD卡内部构件

容量和类型

SD卡具有不同的类型、尺寸和容量。SD卡的类型取决于存储容量以及遵循的SD标准。

表1列出了不同类型的SD卡及其容量。

表1. SD卡类型和容量

常见SD卡类型	容量
SD标准容量(SDSC)	1 MB至2 GB
SD高容量(SDHC)	2 GB至32 GB
SD扩展容量(SDXC)	≥32 GB

SD卡还具有不同的外形和尺寸，例如标准尺寸、迷你尺寸和微型尺寸，如图3所示。

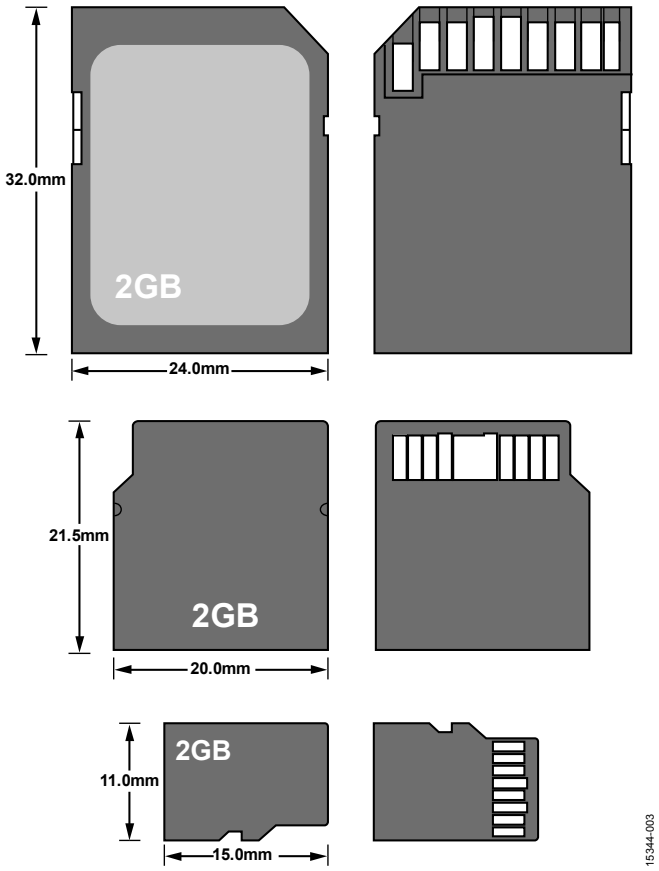

图3. SD卡尺寸

接口和模式

SD卡的引脚接口具有不同的引脚配置，具体取决于使用的通信模式和卡外形。通常，接口由电源线（电源和地）、时钟线、数据线和命令线组成。一些较新的高速卡中实现了低电压差分接口，旨在迎合高速和高带宽传输需求。

与SD卡的通信通常在SD总线模式下完成，这是SDA定义的接口。在这种模式下，接口具有单独的命令线、数据线和时钟线。（后一句话重复，PDF转档问题）SD卡的类型取决于存储容量以及遵循的SD标准。（这句话重复，PDF转档问题）传输格式为专用格式，但如果嵌入式系统不具备SD接口，则很难实现这种格式。

为了支持不具备SD接口的嵌入式系统和微控制器，SDA规范中另增如下定义：SD卡必须支持SPI总线模式。在这种模式下，SD卡可以通过微控制器中广泛使用的SPI进行操作。不过，SPI总线模式仅支持SD卡标准协议的一部分。

图4给出了标准SD卡和微型SD卡的引脚排列。

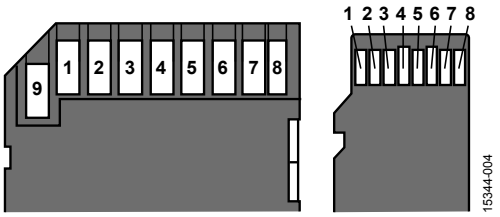

图4. 标准SD卡和微型SD卡的引脚排列

表2给出了标准SD卡的引脚排列说明。

表2. 标准SD卡功能

引脚编号	名称	SD模式	SPI模式
1	CS/DAT3	数据线3	片选
2	CMD/DI	命令线	MOSI
3	VSS1	地	地
4	VDD	电源电压	电源电压
5	CLK	时钟	时钟(SCK)
6	VSS2	地	地
7	DAT0	数据线0	MISO
8	DAT1	数据线1	未用或IRQ
9	DAT2	数据线 2	未用

表3给出了微型SD卡的引脚排列说明。

表3. 微型SD卡功能

引脚编号	名称	SD模式	SPI模式
1	DAT2	数据线2	未用
2	DAT3/CS	数据线3	片选
3	CMD	命令线	MOSI
4	VDD	电源电压	电源电压
5	CLK	时钟	时钟(SCK)
6	VSS	地	地
7	DAT0	数据线0	MISO
8	DAT1	数据线 1	未用或 IRQ

SD卡SPI协议

SPI总线模式下使用的SD协议与SD总线模式下使用的协议略有不同。以SPI总线模式与SD卡进行通信是一种简单的命令响应协议，由主器件（微控制器）发送命令帧来启动。SD卡接收到命令帧后，会根据主机微控制器发送的命令决定发送响应帧或错误帧进行响应。

发送至SD卡的命令帧为6字节结构。命令帧始终以01这两位开始，后跟6位命令编号。

初始字节数据包后跟一个从大到小格式的4字节参数。最后一个字节由7位循环冗余校验(CRC)和1个停止位地址组成（见图5）。

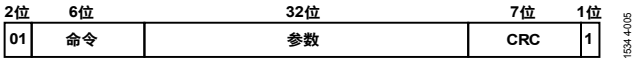

图5. 命令帧格式

根据所发送命令帧的不同，SD卡会以不同的响应帧做出响应。在SPI总线模式下，只有三种响应可供使用：R1、R3和R7（见图6）。

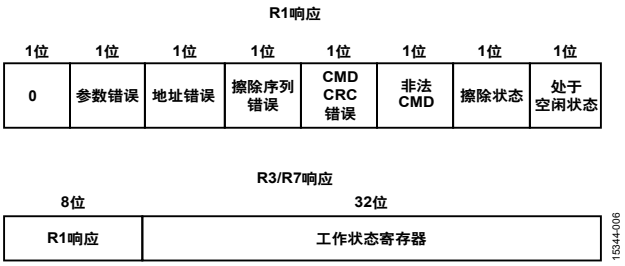

图6. 响应帧格式

接下来，SD卡会在每个命令之后发送R1响应令牌，但SEND_STATUS命令除外。该命令的长度为单字节，最高有效位(MSB)始终设为0。其他位用于错误指示；如果为1，则表示存在错误。错误标志的含义定义如下：

- 空闲状态：卡处于空闲状态和正在运行初始化过程。
- 擦除复位：因接收到退出擦除序列命令而在执行前清除擦除序列。
- 非法命令：检测到非法命令。
- 通信CRC错误：最后一个命令的CRC校验失败。
- 擦除序列错误：擦除序列命令中出错。
- 地址错误：命令中使用了与块长度不符的地址。

- 参数错误：命令的参数（例如，地址或块长度）超出了该卡所允许的范围。

SD卡的SPI模式下使用的命令仅仅是SD模式下使用的命令的一部分。表4详细列出了SPI模式下的命令。该命令集仅限于初始化卡、检索一些重要的详细信息，然后在存储卡中读写数据块。

表4. SPI模式命令集

命令	说明
CMD0	复位卡
CMD8	请求获取当前工作状态
CMD55	特定应用命令(ACMD)的前导命令
ACMD41	启动卡初始化过程
CMD58	请求访问工作状态寄存器(OCR)
CMD16	更改块长度
CMD17	读取数据块
CMD24	写入数据块
CMD32	设置要删除的起始块
CMD33	设置要删除的结束块
CMD38	启动由CMD32和CMD33设置的块擦除

文件分配表(FAT)文件系统

常用的SD卡默认以FAT文件系统格式化。FAT文件系统是一种适用范围广泛的传统文件系统，几乎受所有PC、移动设备和嵌入式系统的支持。该文件系统只需要一个小型稳健的轻量级驱动程序提供支持即可，这在微控制器或嵌入式系统中可以轻松实现。由于该基本文件系统广受支持，因此对于在FAT格式化的存储卡上保存的任何文件，几乎在所有PC上都可以查看和修改。

SD卡规范定义了不同容量的SD卡中可以使用的FAT文件系统类型（更多信息，请参阅参考文献部分）。

SD卡实现

ADuCM3027/ADuCM3029 SPI

ADuCM3027/ADuCM3029微控制器具有三个SPI接口（SPI0、SPI1和SPIH），可用于与各种SPI兼容器件（例如，高速传感器和存储器件）进行通信。每个SPI端口都有四个硬件片选信号，分别用于控制四个SPI兼容器件。此外，SPI总线外设还包括可编程波特率、时钟相位和时钟极性以及不同的硬件流控制机制，而且既可以用作SPI主器件，也可以用作SPI从器件。

这三个SPI在编程和模型方面完全相同，只是连接的内部总线接口有所不同。SPIH外设与高性能的高级外设总线(APB)相连，时钟速率与处理器时钟相同。SPI0和SPI1与主APB相连。ADuCM3027/ADuCM3029微控制器的许多外设都使用APB，这会导致有更多的模块需要仲裁，延迟的不确定性也会更大。

因此，在较高数据速率下，SPIH更为高效，传输数据的延迟也更低。

硬件实现

下面列出了用于演示SD卡与ADuCM3027/ADuCM3029处理器接口的硬件：

1. EVAL-ADuCM3029 EZ-KIT®评估板
2. 与Arduino®兼容的数据记录扩展板
3. 4 GB SanDisk® SD卡

EVAL-ADuCM3029 EZ-KIT评估板

EVAL-ADuCM3029 EZ-KIT评估板是一套适用于ADuCM3027/ADuCM3029处理器的评估系统（见图7）。该评估板包含丰富的板载组件，可用于评估ADuCM3027/ ADuCM3029微控制器。EVAL-ADuCM3029 EZ-KIT还包含EI3接口和Arduino接口。这两个接口可用于连接各种子板和扩展板，从而为EVAL-ADuCM3029 EZ-KIT评估板实现功能扩展。

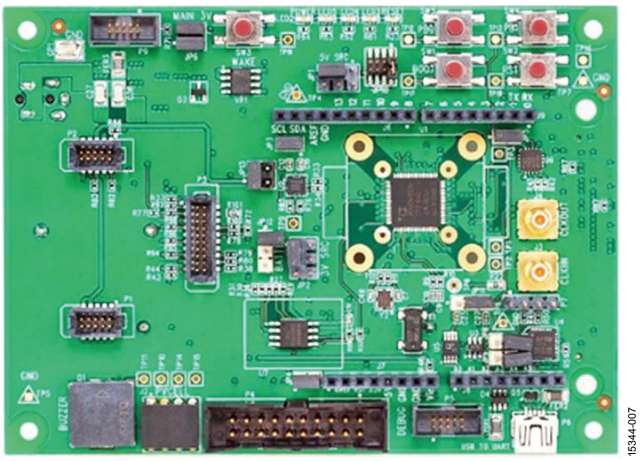

图7. EVAL-ADuCM3029 EZ-KIT评估板

与Arduino兼容的数据记录扩展板

本演示中使用的数据记录扩展板是大多数Arduino记录项目所常用的扩展板（见图8）。记录扩展板的SPI与ADuCM3027/ADuCM3029微控制器的SPIH端口相连。

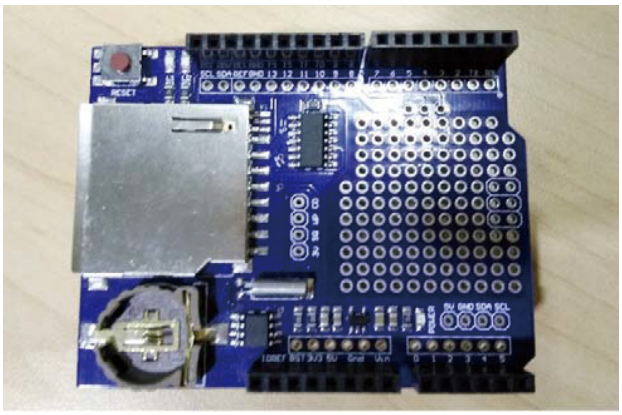

图8. 数据记录扩展板

大部分数据记录扩展板都包含一个互补金属氧化物半导体(CMOS)缓冲器（例如，CD4050），用于充当3.3 V至5 V电压转换器并保护SD卡免受损坏。通常，SD卡的数据输入、串行时钟和片选线会被缓冲，因为它们是卡的输入引脚。不过，当将SD卡与ADuCM3027/ADuCM3029微控制器相连时，该电压转换器则为选配器件，这是因为微控制器的输出与SD卡具有相同的电压电平。

SPIH接口和通用输入/输出(GPIO)端口1共用ADuCM3027/ADuCM3029微控制器的引脚。SPIH、SCLK、MOSI和MISO信号线也使用微控制器引脚（引脚P1_02、引脚P1_03和引脚P1_04），因此这些引脚必须保留未用。

本应用笔记中的应用程序使用GPIO引脚P2_01作为SD卡的片选，而不使用SPIH专用片选（见图9）。

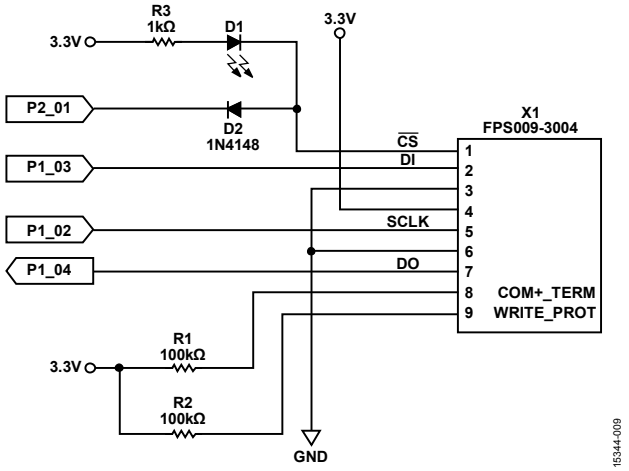

图9. 数据记录扩展板原理图

软件实现

要测试该应用程序，请使用以下软件工具：

- IAR Embedded Workbench 7.60，可访问EVAL-ADuCM3029 EZ-KIT页面下载该软件。
- 适用于IAR的ADuCM302x软件，可访问EVAL-ADuCM3029 EZ-Kit页面进行下载。
- ChaN提供的FatFs库R0.12a，可从FatFs - 通用FAT文件系统模块网站获取。

该应用程序使用FatFs库来处理文件系统调用，以及读取和修改FAT格式化SD卡中的文件。要使用该文件系统库，必须实现硬件抽象层以处理低级硬件相关函数调用。

SPI接口初始化

要初始化ADuCM3027/ADuCM3029 SPIH外设，请按照下述步骤操作：

1. 配置SPIH外设。
2. 将GPIO引脚P2_01配置为片选。
3. 为GPIO1和SPIH共用的引脚配置引脚复用。

配置SPIH外设

SPIH外设配置包括一系列调用设置，让SPIH外设进入就绪状态，准备与SD卡进行通信。要配置SPIH外设，请按照下述步骤操作：

1. 开启SPIH外设。
2. 设置位速率。请注意，SPIH的位速率因微控制器与SD卡间的互操作而异：
   - 当微控制器正在对SD卡进行通信初始化时，位速率必须介于100 kHz和400 kHz之间。
   - 当SD卡配置为SPI模式后，位速率最高可达20 MHz，具体取决于硬件设计和微控制器容量。
3. 设置连续工作模式。在连续模式下，SPI外设可在不取消或中断传输的情况下执行多字节收发操作。

下面给出了用于配置ADuCM3027/ADuCM3029微控制器的SPIH的代码示例：

```
static uint8_t SPIMem[ADI_SPI_MEMORY_SIZE],
static ADI_SPI_HANDLE spih_Dev;

// Open the SPI
adi_spi_Open(SPI_DEV_NUM, SPIMem,
             ADI_SPI_MEMORY_SIZE,
&spih_Dev);

// Set the bit rate
adi_spi_SetBitrate(spih_Dev, 100000);

// Set the continuous mode
adi_spi_SetContinousMode(spih_Dev, true);
```

配置片选

片选使用GPIO引脚（引脚P2_01），而不使用SPI专用片选。使用定制的GPIO引脚可以让用户完全控制片选信号。SD卡和微控制器之间的一些事务需要特别处理片选信号，以便软件适当地控制片选。

要配置片选引脚，请将SPIH片选选项设置为无，然后使用以下代码将GPIO引脚P2_01配置为输出：

```
adi_spi_SetChipSelect (spih_Dev,
   ADI_SPI_CS_NONE);
adi_gpio_OutputEnable (SPI_CS_PORT,
SPI_CS_PIN,
   true);
adi_gpio_SetHigh(SPI_CS_PORT, SPI_CS_PIN);
```

配置微控制器引脚复用器

GPIOx_CFG寄存器是保存ADuCM3027/ADuCM3029微控制器引脚复用器设置的配置寄存器。SPIH使用引脚P1_02、引脚P1_03和引脚P1_04作为SD卡的连接引脚。

要配置SPIH使用的引脚，请设置REG_GPIO1_CFG寄存器中相应的位选项。关于该寄存器的更多信息，请参阅*集成电源管理的ADuCM302x超低功耗ARM Cortex-M3 MCU硬件参考*手册。

下面给出了SPIH使用的端口配置寄存器设置的代码示例：

```
#define SPI0_SCLK_PORTP1_MUX
((uint32_t) ((uint32_t) 1<<4))
#define SPI0_MISO_PORTP1_MUX
((uint32_t) ((uint32_t) 1<<8))
#define SPI0_MOSI_PORTP1_MUX
((uint32_t) ((uint32_t) 1<<6))

*((volatile uint32_t *) REG_GPIO1_CFG) =
SPI0_SCLK_PORTP1_MUX |
SPI0_MISO_PORTP1_MUX |
SPI0_MOSI_PORTP1_MUX;
```

收发数据

配置SPI外设进行SD卡通信后，该接口现在即可收发数据包和启动SD卡。adi_spi_ReadWrite函数用于SD卡的数据收发。该函数需要一个保存数据和SPI器件指针的结构指令。下面给出了本部分所述内容的代码示例：

```
ADI_SPI_TRANSCEIVER spi_xcv_buff;
spi_xcv_buff.pTransmitter = txbuff;
spi_xcv_buff.pReceiver = rxbuff;
spi_xcv_buff.TransmitterBytes = txsize;
spi_xcv_buff.ReceiverBytes = rxsize;
spi_xcv_buff.nTxIncrement = 1;
spi_xcv_buff.nRxIncrement = 1;

adi_spi_ReadWrite(spih_Dev, &spi_xcv_buff);
```

ADI_SPI_TRANSCEIVER是一个结构指令，用于保存事务中使用的缓冲区内容。必须定义用于发送、接收、数据大小和增量的缓冲区。

发送和接收缓冲区为uint8_t数组，大小与发送或接收的数据一致。发送数据大小约为6字节到8字节，接收数据大小从1字节到512字节。

请注意，adi_spi_ReadWrite函数是一个阻塞函数，微控制器需等待当前事务成功后才能执行下一条指令。

SD卡与微控制器接口的示例代码

本应用笔记附有SD卡与ADuCM3027/ADuCM3029接口的示例代码。要查看示例代码，请将项目解压缩并导入IAR Embedded Workbench 7.6（可从EVAL-ADuCM3029 EZ-KIT产品页面下载）。

参考文献

集成电源管理的ADuCM302x超低功耗ARM Cortex-M3 MCU硬件参考。ADI公司，2016年。

SD规范第1部分，物理层精简版规范，版本5.00。SD卡协会。2014.

AN15344sc-0-7/17(0)