AN12777

在 LPC55S6x 上访问双 SD 卡

Rev. 0 — 16 March, 2020 Application Note

目录

1 介绍

1.1 概览

SDMMC 卡接口可在所有 LPC55S6x/LPC55S2x/LPC552x 器件上使用。

SDMMC 卡接口支持使用相同的 SDMMC 外设连接两个设备（SD0 和 SD1）。

本应用笔记介绍如何基于 LPCXprsso55S69 SDK 来使用相同的 SDMMC 外设访问双 SD 卡。

1.2 SDMMC 块

SD/MMC 控制器接口包含以下主要功能块：

- 总线接口单元（BIU）：为寄存器和数据的读/写提供 AHB 和 DMA 接口。
- 卡接口单元（CIU）：处理卡协议并提供时钟管理。
- 内部 MCI DMA 控制器：AHB 总线主控 DMA 控制器。图 1 显示了框图。

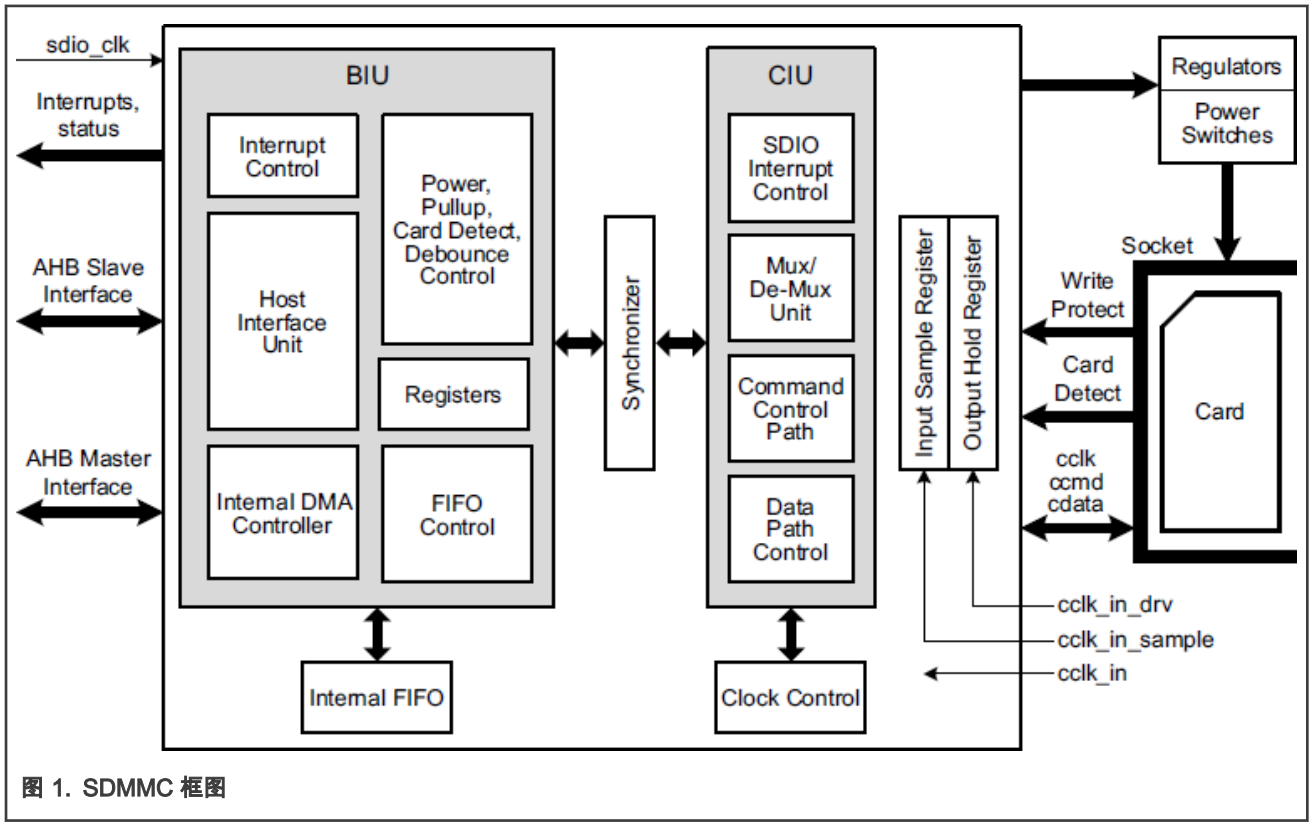


图 1. SDMMC 框图

1.3 SDMMC 引脚描述

表 1 描述了 SDMMC 的可用引脚功能。通常，SDn_CLK, SDn_CARD_DET_N, SDn_CMD, 以及 SDn_D (n= 0, 1) 引脚需要连接到设备，其他引脚是可选的。

注

SD0_D 最多支持 8 位数据，SD1_D 最多支持 4 位。两者都支持 1 位模式。

表 1. SDMMC 引脚描述

引脚功能	类型	描述
SD0_CLK，SD1_CLK	O	SD/SDIO/MMC 时钟
SD0_CARD_DET_N，SD1_CARD_DET_N	I	SDIO 卡检测信号插槽 0 表明存在卡
SD0_WR_PRT	I	SDIO 卡书写保护 1 表明写操作正被保护
SD0_CMD, SD1_CMD	O/I	输入/输出命令
SD0_D[7:0], SD1_D[3:0]	O/I	数据线的数据输入/输出 DAT[7:0]
SD0_POW_EN，SD1_POW_EN	O	SD/SDIO/MMC 插槽电源启用
SD1_BACKEND_PWR	O	嵌入式设备的后端电源供应 它控制一台嵌入式设备的后端电源。该位不控制主机控制器的 VDDH。寄存器位使能软件编程。此寄存器上的值控制打开和关闭嵌入式设备。
SD0_CARD_INT_N，SD1_CARD_INT_N	I	卡中断线 该引脚用于指示卡中断，即使在卡的时钟关闭时也会对其进行采样。它连接到 eSDIO 卡中断线，仅针对 eSDIO 定义。

像其他外设一样，可以通过 I/O 控制寄存器（IOCON）来配置 SDMMC 引脚。表 2 列出了 SD0_CLK, SD0_CMD, SD0_Dn，SD1_CLK, SD1_CMD，以及 SD1_Dn 这些引脚的设置。

表 2. SDMMC 建议引脚设置

IOCON bit(s)	类型 D 引脚	类型 A 引脚
10	未使用，设置为 0	模拟开关已打开 (禁用)。设置为 0。
9	控制漏极开路。设置为 0	与类型 D 相同
8	DIGIMODE：设置为 1	与类型 D 相同
7	INVERT：设置为 0	与类型 D 相同
6	SLEW：设置为 1	与类型 D 相同
5:4	Mode：设置为 0	与类型 D 相同

Table continues on the next page...

表 2. SDMMC 建议引脚设置 (continued)

IOCON bit(s)	类型 D 引脚	类型 A 引脚
3:0	FUNC：必须为该外设选择正确的功能	与类型 D 相同
总体评论	SDIO 功能的不错选择	一个潜在的选择。缺少 SLEW 功能可能会降低性能。

注

表 1 中的引脚功能由 表 2 中 IOCON bit 3:0 (FUNC field) 的值配置。 有关详细信息，请参考 *LPC55S6x/LPC55S2x/LPC552x User manual*（文档 UM11126）中 **IOCON pin functions in relation to FUNC values** 章节中的表格。

2 开发及测试环境

为了展示双 SD 卡支持的功能，软件和硬件环境搭建如下。

2.1 软件环境

- 具有 SDMMC 中间件支持的 LPCXpresso55S69 SDK (v2.63)
- KEIL MDK v5.27

串行端口程序（例如 PuTTY），其设置为：

```
115200+8+N+1
```

2.2 硬件环境及搭建

- LPCXpresso55S69 评估板 Rev2.0
- 两张 SD 卡
- 带有 SD 卡卡槽的板子
- 电脑
- Micro USB 线缆以及部分飞线

LPCXpresso55S69 EVK 板设计为支持一张 SD 卡，其上一个 SD 卡插槽连接到 SD0 引脚。并且它在 Arduino 接头上保留了与SD1 相关的引脚，用于连接到另一个 SD 卡。图 2 显示了原理图的截图，相关的引脚用黄色突出显示。

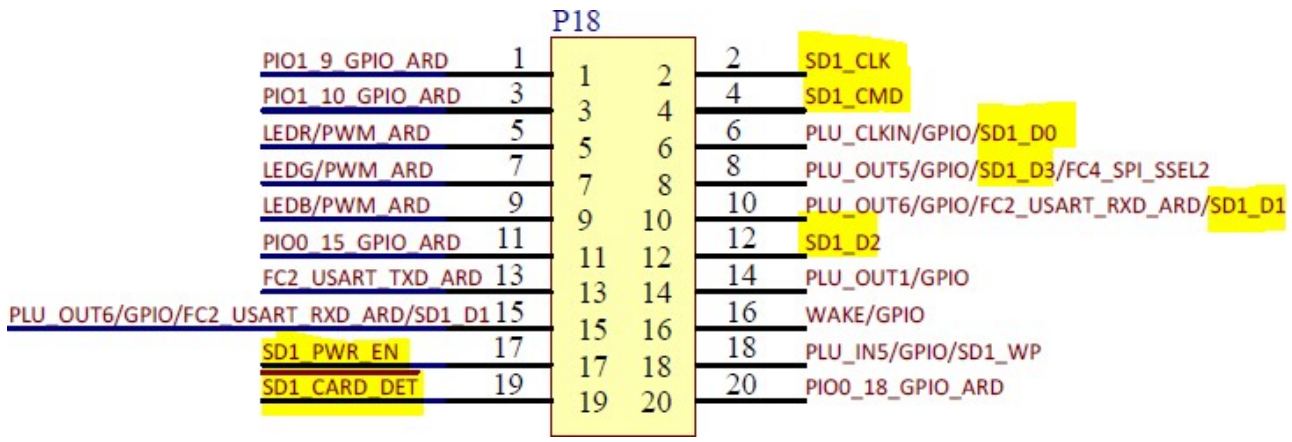


图 2. LPCXpresso55S69 EVK 上保留的 SD1 引脚

要在具有 SD1 引脚的另一板上驱动 SD 卡，也应共享电源和地。它们也保留在 LPCXPrsso55S69 评估板的 Arduino 接头上，在图 3 中以黄色突出显示。

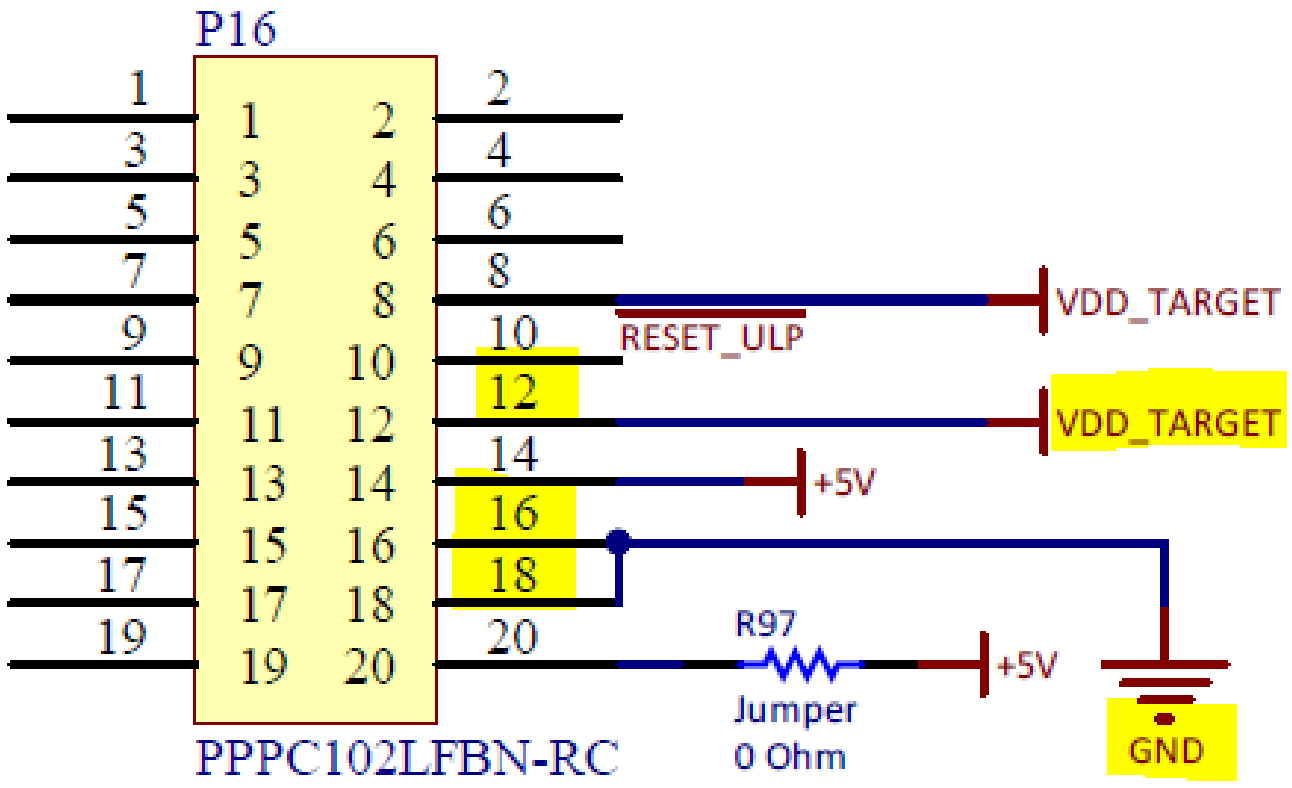


图 3. LPCXpresso55S69 EVK 上保留的电源和 GND 引脚

图 4 展示了 LPCXprsso55S69 评估板上 Arduino 接头实际连接 SD1 的照片截图。使用飞线，可以很容易地连接到另一块带有 SD 卡槽的板子，以便访问。

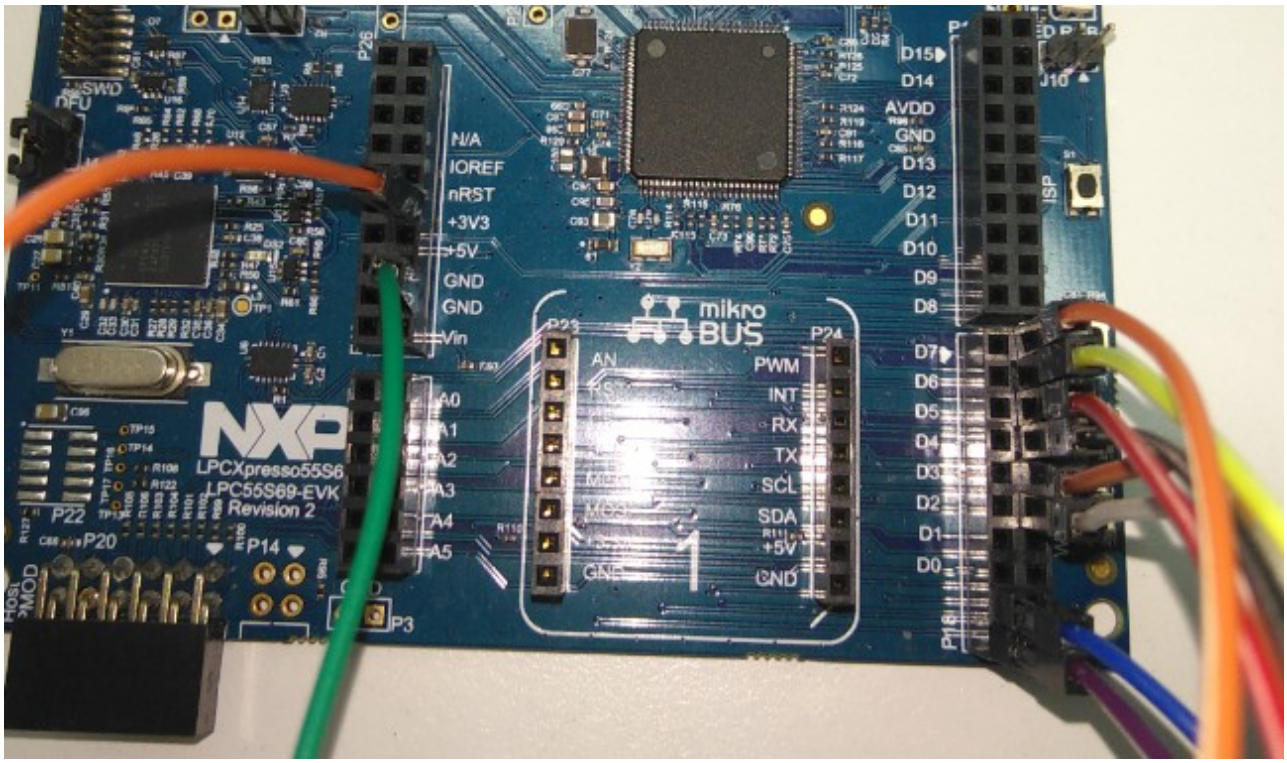


图 4. LPCXpresso55S69 EVK 上 Arduino 接头的 SD1 连接

3 软件实现

3.1 基本思想

软件实现基于 LPCXpresso55S69 SDK。 软件实现的基本思想如下。

两个设备共享一个 SDMMC 控制器，尽管它们有两组独立的引脚。 因此，对两个设备的访问过程必须是串行的，不能并行。这意味着只有在前一台完成之后，下一台设备上的事务才可以开始。否则，可能会相互干扰。 这个关键点限制了软件的实现。 应用程序的开发人员也应该注意这一点。

因此，为了更简单，清楚地显示此功能，双 SD 卡上的事务在参考代码中以轮询和阻塞模式实现。

此外，当前的 SDK 仅提供接口一张 SD 卡（SD0）的函数。 为了更好地与 SDK 兼容，将在新文件中创建支持另一张 SD 卡（SD1）的新函数，而不是修改现有函数和文件。 因此，可以在不干扰现有文件的情况下可以将这些包含新函数的文件添加到 SDK 中。

最后，虽然如上所述，SDMMC 外设是共享的，但寄存器中 SD0 和 SD1 仍然存在一些不同的配置：

- PWREN 寄存器用于使能电源。
- CLKENA 寄存器用于使能时钟。
- CDETECT 寄存器用于检测卡的。
- CTYPE 寄存器用于在 1 位，4 位和 8 位模式之间切换卡类型。
- 发送命令时，必须分别对应于 SD0 或 SD1 将 CMD 寄存器的 20-16 位设置为 0 或 1。

对 PWREN, CLKENA, CDETECT, 以及 CTYPE 寄存器的控制已在 SDK 的 HAL 级别上实现。有关这些寄存器的介绍，请参考 *LPC55S6x/LPC55S2x/LPC552x User manual*（文档 UM11126）。在双 SD 支持的软件实现中，只需要调用支持 SD1 的 API ，例如 SD 1 初始化。 但是，向 SD1 发送命令的功能并没有在 SDK 中的传递函数中实现。 因此需要添加 SD1 的传递函数。表 3 描述了 CMD 寄存器中的相关字段。

表 3. CMD 寄存器中 SD0 或 SD1 的命令字段

位	标志	值	描述	重置值
20:16	CARD_NUMBER	0 or 1	指定执行当前命令的 SD 卡卡号	0

基于上述两个 SD 卡设备（SD0 and SD1）的不同寄存器配置，添加了使能 SD1 的新函数。

3.2 在 SDK 上为 SD1 使能添加函数

本节将介绍基于 SDK 中 SDMMC 的 SW 层如何添加用于 SD1 使能函数的要点：HAL 层，主机层，协议层和板级。 有关更多信息，请参考 AN12777SW。

1. **HAL 层**

   SDMMC 接口的 HAL 层在 SDK 中被抽象到了文件 `fsl_sdif.c` 和 `fsl_sdif.h` 之中，其中包含了 SDMMC 寄存器的设置。

   正如基本思想中所提到的，除了通过 CMD 寄存器向 SD1 发送命令外，SD0 和 SD1 的所有其他不同寄存器配置均已在 SDK 中实现。表 4 列出了 SD1 的函数应用程序接口。

表 4. SDK 中 SD1 HAL 的 API

函数名称	描述	评论
`SDIF_EnableCard1Clock()`	使能/禁用 SD 卡 1 时钟	在 `fsl_sdif.h` 中提供
`SDIF_EnableCard1Power()`	使能/禁用 SD 卡 1 电源	在 `fsl_sdif.h` 中提供
`SDIF_SetCard1BusWidth()`	设置 SD 卡 1 数据总线宽度	在 `fsl_sdif.c` 中提供
`SDIF_DetectCard1Insert()`	检测 SD 卡 1 插入状态	在 `fsl_sdif.h` 中提供

   因此，需要创建一个函数，以通过 CMD 寄存器将命令发送到 SD1。参考现有为 SD 0 提供的名为 `SDIF_SetCommandRegister()`函数， 新函数被命名为 `SDIF1_SetCommandRegister()`其中 SD 1 被指定为当前卡号（有关定义参考 表 3）。相关代码如 图 5 所示。黄色突出显示的是相应位的设置。

```
//send command to sd card 1
base->CMD = cmdIndex | SDIF_CMD_CARD_NUMBER(1) | SDIF_CMD_START_CMD_MASK;
```

图 5. 设置 CMD 寄存器中 SD0 或 SD1 的命令字段

   更多详细信息，请阅读 `fsl_sdif1.c` 的新文件，包括 HAL 层的所有新功能。

2. **主机层**

   归类为中间件的主机层是 SDMMC HAL 和协议层之间的接口。 基本的系统操作是在此层完成的，比如传输功能，卡检测功能，卡电源切换功能。此层中的参考代码是参考在路径 \middleware\sdmmc\port\sdif\polling\下 fsl_sdmmc_host.c 文件

   在名为 `SDMMC1HOST_TransferFunction()`.的 SD1 新的传递函数中，添加了代码行以指定 SD1 然后以阻塞模式进行传递。参见 图 6 中用黄色标亮的位置。

```
content->command->flags |= SDIF_CMD_CARD_NUMBER(1);//specify SD card 1

if (kStatus_Success != SDIF_TransferBlocking(base, &dmaConfig, content))
{
    error = kStatus_Fail;
}
```

图 6. 设置 CMD 寄存器中 SD0 或 SD1 的命令字段

用于 SD1 卡检测的 SDMMCHOST_DetectCard1InsertByHost() 函数和用于 SD1 电源切换的 SDMMCHOST_PowerOnCard1()/SDMMCHOST_PowerOffCard1() 函数分别是调用已在 SDK 中实现的 API 。

SDMMCHOST_CARD1_DETECT_INSERT_STATUS() 和 SDIF_EnableCard1Power() 已在 SDK 中的 HLA 层中实现。

更多详细信息，请阅读 fsl_sdmmc_host1.c 的新文件，包括 Host 层的所有新功能。

3. **协议层**

   归类为中间件的协议层除了用于被应用程序调用的 MMC 和 SDIO 外还实现 SD 卡的命令协议。根据上述 SD0 和 SD1 的不同寄存器配置，此层需要为 SD1 添加的新功能涉及 SD1 初始化，SD1 时钟和时序设置，SD1 电源切换和 SD1 检测，这些功能是在新文件 fsl_sd1.c 中实现的。 它们调用上面提到的 HAL 层和主机层中应用程序接口函数。请阅读文件 fsl_sd1.c 以了解详细。

   注

   因为 SDK 中 fsl_sd.c 文件中的许多 SD 卡协议函数都定义为 static。fsl_sd1.c 的新文件将包含在 fsl_sd.c 文件的行尾，以便共享减少代码容量。

4. **板级**

   由于 SD1 具有独立于 SD0 的引脚，因此还需要配置与 SD1 相关的引脚。可以通过参考建议设置或 SDK 中 SDMMC 示例里与 SD0 相关的引脚设置。

3.3 为双 SD 卡支持更新 SDK

软件包中提供了为双 SD 卡支持创建的源代码和项目信息文件。 必须将它们合并到 SDK 中才能工作。 执行以下操作更新 SDK 以实现双 SD 卡支持。

- 将随附的 SW 软件包中\drivers\下的 fsl_sdif1.c 和 fsl_sdif1.h 文件复制到 LPCXpresso55S69 SDK 软件包中的 \devices\LPC55S69\drivers\中。
- 将随附的 SW 软件包中\sdmmc\inc\下的 fsl_sd1.h 和 fsl_sdmmc_host1.h 文件复制到 LPCXpresso55S69 SDK 软件包中的\middleware\sdmmc\inc\中。
- 将随附的 SW 软件包中\sdmmc\port\sdif\polling\下的 fsl_sdmmc_host1.c 文件复制到 LPCXpresso55S69 SDK 软件包中的\middleware\sdmmc\port\sdif\polling\中。
- 将随附的 SW 软件包中\sdmmc\src 下的 fsl_sd1.c 文件复制到 LPCXpresso55S69 SDK 软件包中的 \middleware\sdmmc\src\中。

  在 LPCXpresso55S69 SDK 软件包的 fsl_sd.c 文件的行末添加#include fsl_sd1.c
- 将随附的 SW 软件包中包含 keil MDK 项目信息，示例代码和电路板代码层的 dual_sdcards 文件夹复制到 LPCXpresso55S69 SDK 软件包中的\boards\lpcxpresso55s69\demo_apps\。

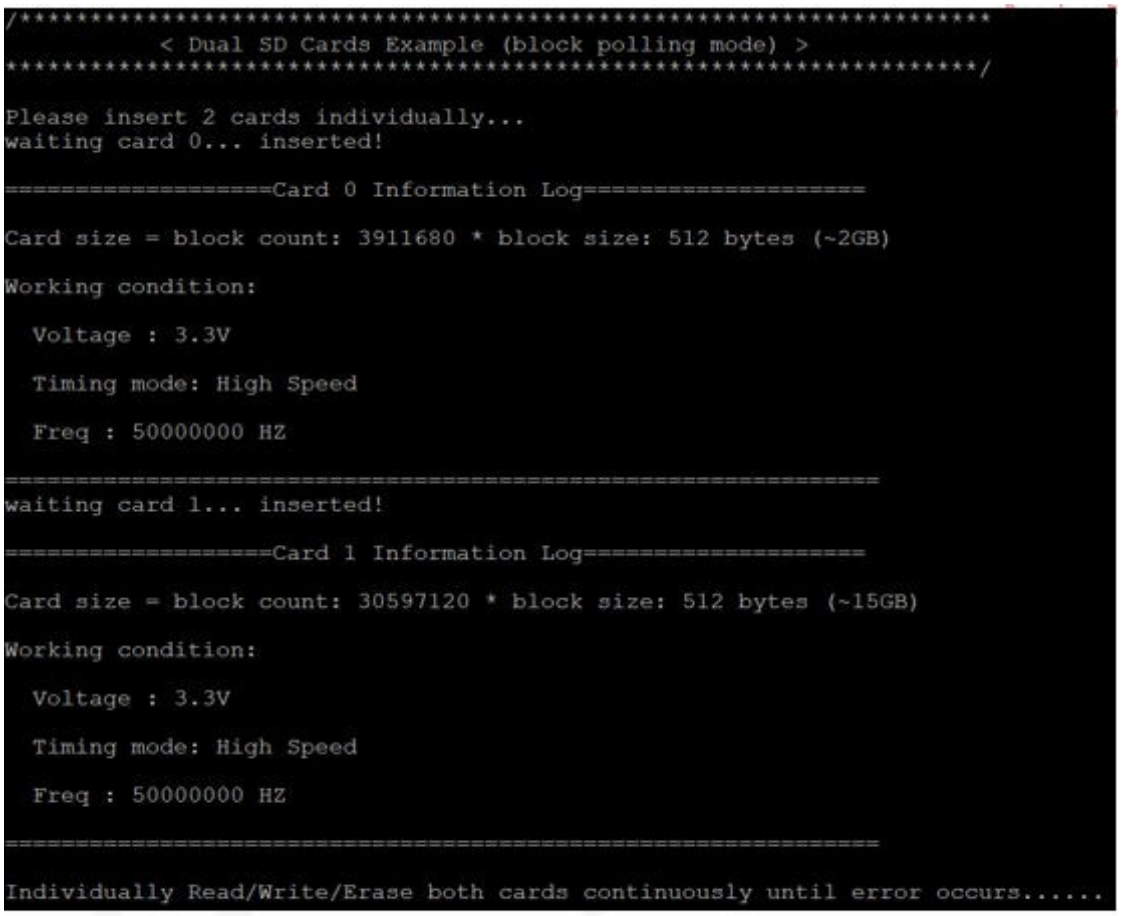
3.4 演示双 SD 卡访问

在\boards\lpcxpresso55s69\demo_apps\dual_sdcards\下，在文件 dual_sdcards.c 中，有一个简单的实例实现，以演示对双 SD 卡的访问。

在该示例中，完成有关 SDMMC 的系统和外围设备的基本配置后，执行以下步骤：

1. 首先检测是否有 SD 卡（SD0）插入插槽。 一旦检测到，该卡将被上电并初始化。
2. 打印出初始化过程中获得的卡的信息。
3. 对另一个卡插槽中的另一张 SD 卡（SD1）执行相同的操作。 成功初始化两张卡后，执行对 SD0 卡的访问。
   a. 向其写入一个数据块并将其读出。
   b. 比较数据是否一致。
   c. 写入/读取/比较多个数据块。
   d. 如果都一致的话，则表明读取成功。否则，就是失败的。

4. 在 SD1 卡上执行相同的访问。
5. 在搭建好前面所提的软件和硬件后，在 SDK 中 `\boards\lpcxpresso55s69\demo_apps\dual_sdcards\cm33_core0\mdk\`打开 `dual_sdcards.uvprojx` 这一项目。在成功构建，下载并运行例程后，将在串行终端上看到双 SD 卡的访问日志，如 图 7 所示。


/***
< Dual SD Cards Example (block polling mode) >
***/
Please insert 2 cards individually...
waiting card 0... inserted!
====================Card 0 Information Log====================
Card size = block count: 3911680 * block size: 512 bytes (~2GB)
Working condition:
Voltage : 3.3V
Timing mode: High Speed
Freq : 50000000 HZ
==
waiting card 1... inserted!
====================Card 1 Information Log====================
Card size = block count: 30597120 * block size: 512 bytes (~15GB)
Working condition:
Voltage : 3.3V
Timing mode: High Speed
Freq : 50000000 HZ
==
Individually Read/Write/Erase both cards continuously until error occurs.....

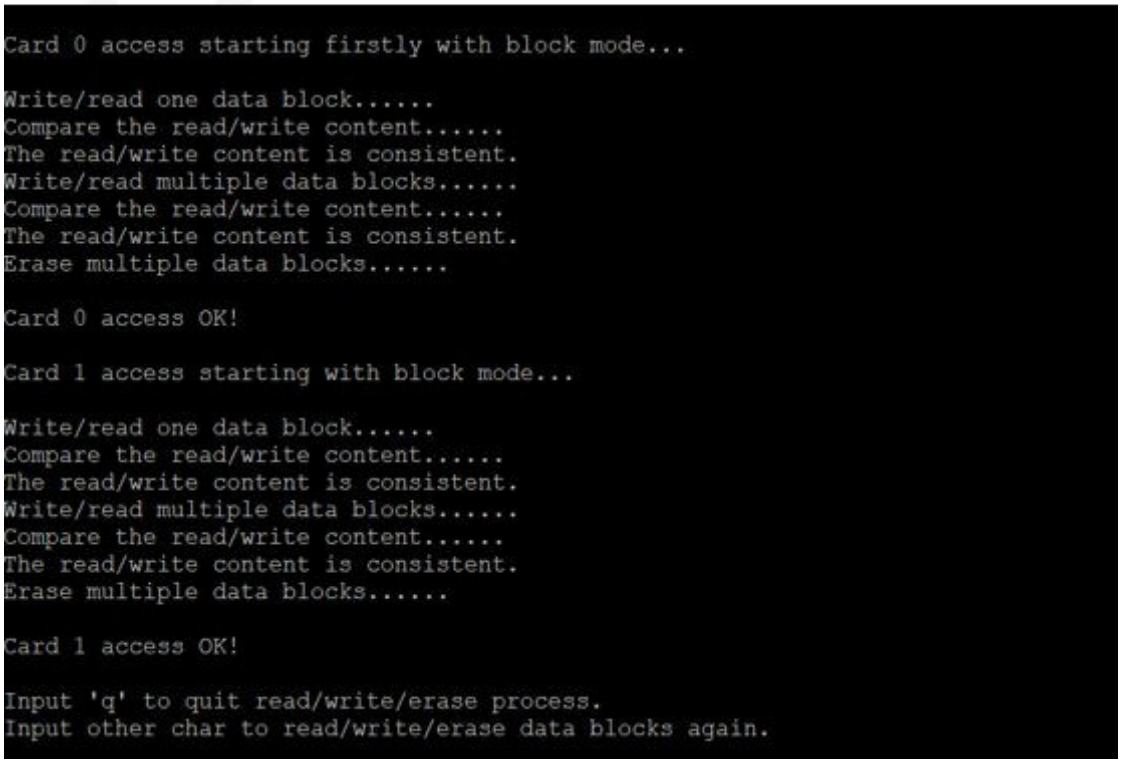

Card 0 access starting firstly with block mode...
Write/read one data block......
Compare the read/write content......
The read/write content is consistent.
Write/read multiple data blocks......
Compare the read/write content......
The read/write content is consistent.
Erase multiple data blocks......
Card 0 access OK!
Card 1 access starting with block mode...
Write/read one data block......
Compare the read/write content......
The read/write content is consistent.
Write/read multiple data blocks......
Compare the read/write content......
The read/write content is consistent.
Erase multiple data blocks......
Card 1 access OK!
Input 'q' to quit read/write/erase process.
Input other char to read/write/erase data blocks again.

图 7. 记录双 SD 卡访问

4 总结

本应用笔记详细介绍了在 LPC55S6x 系列上使用相同的一个 SDMMC 接口实现双 SD 卡支持的方法。 它的参考代码的实现是基于支持 SDMMC 中间件的 LPCXpresso55S69 SDK（其中仅支持一个 SD 卡）。 对应于这个 SDK，LPCXPresso55S69 EVK 板被使用，因为有另一个 SD 卡的引脚被保留很方便连接。

- 介绍 SDDMC 模块和 SD0 和 SD1 的引脚分配。
- 描述了软件和硬件环境，特别是双 SD 卡支持的硬件设置。
- 由于 SDK 已支持其中一张卡，该笔记说明了在基本思想下如何为另一张 SD 卡添加函数。
- 说明如何将软件包合并到 SDK 中，最后通过一个简单的例子展示了结果。

How To Reach Us

Home Page:

nxp.com

Web Support:

nxp.com/support

Limited warranty and liability — Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including “typicals,” must be validated for each customer application by customer's technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: nxp.com/SalesTermsandConditions.

Right to make changes - NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Security — Customer understands that all NXP products may be subject to unidentified or documented vulnerabilities. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer’s applications and products. Customer’s responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer’s applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately. Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP. NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

For more information, please visit: http://www.nxp.com

For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 16 March, 2020

Document identifier: AN12777

arm