Application Report
SLAA281A−November 2005−Revised January 2006

Interfacing the MSP430 with MMC / SD Flash Memory Cards

Stefan Schauer, Christian Speck

MSP430

ABSTRACT

This application report and the associated source code files demonstrate the implementation of a serial peripheral interface (SPI) between the MSP430F161x microcontroller and an MMC or SD Flash Memory Card used in SPI mode. The provided information can be used with any MSP430 device with a hardware SPI interface.

1 Hardware Description

The MSP430F161x is used to communicate with the MMC or SD card via the SPI interface. SPI is a fast and efficient protocol that allows for simultaneous bi-directional data transfer. Serial data is transmitted and received by the MSP430 using he USART module in SPI mode. The hardware interconnection for the master–slave configuration operating on a single supply voltage is shown in Figure 1.

The associated MSP430F161x pins P5.3 and P5.4 are configured as GPIO to control the Chip-Select pin and read the memory card detect signal. The USART1 hardware peripheral of the MSP430F161x is configured in the 3-pin SPI mode. Pins P5.1, P5.2, and P5.3 of the MSP430 provide the SIMO1, SOMI1, and UCK1 interface to the MMC card.

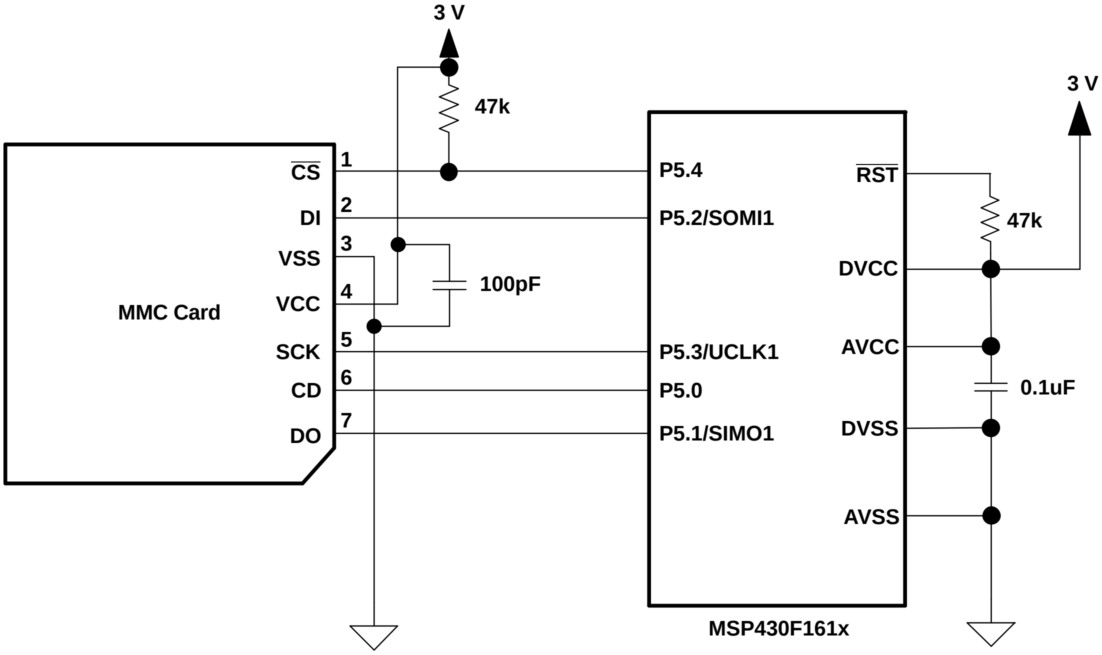

Figure 1. Connection between MSP430 and MMC Card

2 Software Description

The code associated with this application note is designed as a driver set for communication between a MSP430 and a MMC or SD card via the SPI bus as illustrated in the preceding hardware section. The code is written modular and can be reused easily. Note that only a subset of the available card commands are used based on the limitations of the SPI interface and the secure functions of the SD Card.

An example main() function is provided to illustrate proper use of the driver functions. It initializes the MSP430F161x in the 3-pin SPI mode with the function initMMC(). The MSP430 will then poll until an MMC card is detected and try to read the device memory size. Upon the completion of this, the MSP430 prepares a buffer with 512 bytes of data and writes it into two different sectors of the card. Finally, the MSP430 will read the data back that was written to each memory segment.

After uncommenting the code line:

```
//#define withDMA
```

the MSP430F161x DMA module will be used for data transmission between the MSP430 and the MMC card, resulting in higher communication speed and less CPU load.

If the software should be adopted to a different USART or a different device where the USART is connected to other port pins following functions have to be checked for the correct settings:

- void initSPI (void)
- char initMMC (void)
- mmc.h file

3 Function Description

3.1 *char initMMC (void);*

Initializes the Port, SPI and the MMC Card. This routine may need to be adjusted if the library should be adopted to another MSP430 type or to another USART module. Also some of the control signals could be set to a different port if necessary.

Parameter:	None	
Return:	Status	Error/success code

3.2 *char mmc_ping(void);*

Check if MMC card is present.

Parameter:	None	
Return:	Status	Error/success code

3.3 *void mmcSendCmd (const char cmd, unsigned long data, const char crc);*

Send a command to the MMC card.

Parameter:	cmd	Command which should be sent to the MMC card
	data	Data for the command
	crc	Checksum for the command
Return:	None	

3.4 char mmc_GoIdle();

Set the MMC Card in Idle mode to save current.

Parameter:	None	
Return:	Status	Error/success code

3.5 char mmcSetBlockLength (const unsigned long);

Set the MMC block length of count=2^n Byte. Normally this command is not required. The default block length is 512 byte.

Parameter:	Block Length of count=2^n Byte	
Return:	Status	Error/success code

3.6 char mmcReadBlock(const unsigned long address, const unsigned long count, unsigned char *pBuffer);

#define mmcReadSector(sector, pBuffer) mmcReadBlock(sector × 512, 512, pBuffer)

Read a size Byte big block beginning at the address.

Parameter:	address	Start address of data to read on the card
	count	Number of bytes to read
	pBuffer	Pointer to read buffer
Return:	Status	Error/success code

3.7 char mmcWriteBlock (const unsigned long address, const unsigned long count, unsigned char *pBuffer);

#define mmcWriteSector(sector, pBuffer) mmcWriteBlock(sector × 512, 512, pBuffer).

Write a 512 byte big block beginning at the (aligned) address

Parameter:	address	Start address of data to read on the card
	count	Number of bytes to write
	pBuffer	Pointer to write buffer
Return:	Status	Error/success code

3.8 char mmcReadRegister (const char cmd_register, const unsigned char length, unsigned char *pBuffer);

Read the Register arg1 with Length arg2 (into the buffer).

Parameter:	cmd_register	Register to read
	length	Number of bytes to read
	pBuffer	Pointer to read buffer

Return:	Status	Error/success code

3.9 *unsigned long MMC_ReadCardSize(void);*

Read the Card Size from the CSD Register.

Parameter:	None
Return:	Detected Card Size

4 References

1. *MSP430x169 Mixed Signal Microcontroller* Data Sheet (SLAS368)
2. *MSP430x1xx Family User's Guide* (SLAU049)
3. *SanDisk MultiMediaCard Product Manual* (SanDisk, 2001)

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		**Applications**	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DSP	dsp.ti.com	Broadband	www.ti.com/broadband
Interface	interface.ti.com	Digital Control	www.ti.com/digitalcontrol
Logic	logic.ti.com	Military	www.ti.com/military
Power Mgmt	power.ti.com	Optical Networking	www.ti.com/opticalnetwork
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
		Telephony	www.ti.com/telephony
		Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

Mailing Address: Texas Instruments
Post Office Box 655303 Dallas, Texas 75265