

**Project Log 3**  
**Project Title:** USB MicroSD Card Reader  
**EEE G512** Embedded System Design  
October 2018

**Submitted to:**  
Dr. Devesh Samaiya

**Submitted by:**  
Joy Parikh | 2016A3PS0136P  
Rutwik Narendra Jain | 2015A3PS0726P

### 3 Accessing SD Card using SPI on LPC2148

The SD/MMC are the universal low cost, high speed (MMC clock up to 20 MHz, SD clock up to 25 MHz), data storage media. The bus architecture of SD/MMC can be chosen in one of the two mode modes, SD/MMC or SPI.

In SPI mode the following signals are included in the bus architecture:

1. CS: Host to card Chip Select signal
2. CLK: Host to card clock signal
3. MOSI: (Master Out Slave In) Host to card single bit data signal
4. MISO: (Master In Slave Out) Card to host single bit data signal

In order to configure SPI interface on the LPC2148 SSP port, the design consideration includes,

- **GPIO setting** The SPI pins, CLK, CS, MOSI, MISO need to be configured through pin select and GPIO registers, PINSEL1, IODIR0, and IOSET0, before configuring the SPI interface.
- **SPI clock pre-scale and clock rate** Based on the APB clock (PCLK) setting in APB Divider Control register (APBDIV), the clock pre-scale can be set through SSP Clock pre-scale Register (SSPCPSR), and the clock rate can be controlled in the SSP Control 0 Register (SSPCR0).
- **SPI frame format and data size** The SPI format and data size can be configured through setting the proper clock polarity bit (CPOL) and clock phase bit (CPHA) and data size field (DSS) in the SSP control registers (SSPCR0). The data size is set to 8 bits/frame, and both CPOL and CPHA bits are set to zero.
- **SPI enable/disable** The SSP port should be disabled before the GPIO pin setting, clock pre-scale setting, frame format configuration, and enabled after all the configuration is done to ensure a clear start.

The SPI related APIs include:

- void SPI\_Init ( void );  
Initializing SPI interface through configuring GPIO, VPBDIV, SSP port registers.

- void SPI\_Send ( unsigned char \* data\_pointer, unsigned int data\_length );  
Sending a block of data based on the data pointer and the length of the data block.
- void SPI\_Receive ( unsigned char \*data\_pointer, unsigned int data\_length );  
Receiving a block of data based on the data pointer and the length of the data block.
- unsigned char SPI\_ReceiveByte( void );  
Receiving one byte of data, the return value of the API is the received data. This API is primarily used to obtain the command response at different phases.

### 3.1 Basic SD/MMC commands

CMD0 GO\_IDLE\_STATE, reset the card to idle state

CMD1 SEND\_OP\_COND, ask the card in idle state to send their operation conditions contents in the response on the MISO line. Any negative response indicates the media card cannot be initialized correctly.

CMD16 SET\_BLOCKLEN, set the block length (in bytes) for all the following block commands, both read and write. In the sample program, the data length is set to 512 bytes.

CMD17 READ\_SINGLE\_BLOCK, read a block of data that its size is determined by the SET\_BLOCKLEN command.

CMD24 WRITE\_BLOCK, write a block of data that its size is determined by the SET\_BLOCKLEN command.

## References

- [1] AN10406 Application Note: Accessing SD/MMC Card Using SPI on LPC2000, Rev. 03, January 2007
- [2] ARM Keil Documentation, USBMem- Mass Storage Device,  
Available online: <http://www.keil.com/support/man/docs/mcb2140>
- [3] UM10139 LPC214x Manual, Rev. 4, NXP Semiconductors, 23 April 2012.
- [4] SDCard Interfacing with LPC2148 using SPI Module - WikiNote
- [5] SparkFun LPC2148 Bootloader Tutorial <https://www.sparkfun.com/tutorials/94>
- [6] AN10711 USB secondary ISP bootloader: Application Note, Rev. 02, 15 July 2008.