PmodSF Programmer's Reference Manual

Revision: December 15, 2011

1300 NE Henley Court, Suite 3
Pullman, WA 99163
(509) 334 6306 Voice | (509) 334 6300 Fax

Introduction

This document describes the programming interface to the PmodSF library that is included as part of the PmodLib library. It describes the capabilities of the PmodSF library and all the API functions used to access its features.

The PmodSF library provides functions operating the PmodSF 16/128 serial flash device on Cerebot PIC32 based microcontrollers. The PmodSF is based on the ST M25P16/M25P128 serial flash memory chip and utilizes SPI for communication. Detailed operating information can be found in the M25P16/M25P128 and Digilent PmodSF reference manuals.

PmodSF Basic API Functions

void PmodSFInit(SpiChannel chn,uint32_t pbClock,uint32_t bitRate)

Parameters
SpiChannel chn - Spi channel
uint32_t pbClock - peripheral bus clock in Hz
uint32_t bitRate - bit rate desired in Hz
Return Value
none

Opens the desired SPI channel in 8-bit mode as a master, enables the slave select bit, and sets the desired bit rate as a function of pbClock/bitRate. Examples of peripheral bus bit rate combinations are available in the PIC32 Family Reference Manual.

void PmodSFWriteStatusRegister(SpiChannel chn,uint8_t statusReg)

Parameters
Input: SpiChannel chn - Spi channel
uint8_t statusReg - status register bits (Appendix A: Table 1)

Return Value
none

Writes configuration bits to the status register on the specified SPI channel. See Table 1: PMODSF Status Register Format for register values. For a complete description see Write Status Register (WRSR) in the M25P16/M25P128 reference manual.

void PmodSFBlockWhileWriteInProgress(SpiChannel chn)

Parameters
SpiChannel chn - Spi Channel

Return Value
none

During a write operation the Write In Progress(WIP) bit is set, operations that write to the PmodSF will be ignored while this bit is set. Calling this function blocks further operations until the WIP cleared condition is met.

void PmodSFWriteEnable(SpiChannel chn)

Parameters
SpiChannel chn - Spi Channel

Return Value
none

Enables writing to flash by setting the Write Enable Latch(WEL) bit on the selected channel. For a complete description see Write Enable (WREN) in the M25P16/M25P128 reference manual.

uint8_t PmodSFReadStatusRegister(SpiChannel chn)

Parameters
SpiChannel chn - Spi Channel

Return Value
uint8_t - status register bits (Appendix A: Table 1)

Reads the value of the status register. See appendix A Table 1: PMODSF Status Register Format for register values. For a complete description see Read Status Register (RDSR) in the M25P16/M25P128 reference manual .

void PmodSFWriteDisable(SpiChannel chn)

Parameters
SpiChannel chn - Spi Channel

Return Value
none

Disable writing to flash by resetting the Write Enable Latch(WEL) bit on the selected channel. For a complete description see Write Disable (WRDI) in the M25P16/M25P128 reference manual.

uint32_t PmodSFReadID(SpiChannel chn)

Parameters
SpiChannel chn - Spi Channel

Return Value
uint32_t - Bits 0 - 7: Memory Capacity
Bits 8 - 15: Memory Type
Bits 16- 23 : Manufacturer ID

The Read Identification (RDID) reads the 24-bit device identification to into a 32 bit unsigned integer. See appendix A Table 2: Manufacturer ID/Memory Type/Memory Capacity. For a complete description see Read Identification (RDID)in the M25P16/M25P128 reference manual.

uint8_t PmodSFDeepPowerDownRelease(SpiChannel chn)

Parameters
SpiChannel chn - Spi Channel

Return Value
uint8_t electronic signature 14h

Release deep power down on the selected channel, old style electronic signature 14h is returned. For a complete description see Release from Deep Power-down and Read Electronic Signature (RES)in the M25P16 reference manual.

Notes:
(1) Blocks while Write In Progress bit is set prior to performing operation.
(2) This function available for the PmodSF-16 only.

void PmodSFDeepPowerDown(SpiChannel chn)

Parameters
SpiChannel chn - Spi Channel

Return Value
none

Performs a deep power down on the selected channel. For a complete description see Release from Deep Power-down (DP) in the M25P16 reference manual.

Notes:
(1) Blocks while Write In Progress bit is set prior to performing operation.
(2) This function available for the PmodSF-16 only.

void PmodSFPageProgram(SpiChannel chn,uint32_t numBytes,uint8_t *data,uint32_t address)

Parameters

SpiChannel chn - Spi channel
uint32_t numBytes - number of bytes to write
uint8_t *data - data to write
uint32_t address - 24bit representation of the page address

Return Value

none

The Page Program (PP) instruction programs bytes into a single page of flash memory (changing bits from 1 to 0). For a complete description see Page Program (PP) and Memory Organization, in the M25P16/M25P128 reference manual.

Notes:

(1) Blocks while Write In Progress bit is set prior to performing operation

void PmodSFReadBytes(SpiChannel chn,uint32_t numBytes,uint8_t *data,uint32_t address)

Parameters

SpiChannel chn - Spi channel
uint32_t numBytes - number of bytes to read
uint8_t *data - buffer to store data read
uint32_t address - 24bit representation of the page address

Return Value

none

The Read Data bytes reads N number of bytes from flash memory into a buffer from the specified 24-bit address on the SPI channel selected. For a complete description see Read Data Bytes (READ) and Memory Organization in the M25P16/M25P128 reference manual.

Notes:

(1) Blocks while Write In Progress bit is set prior to performing operation

void PmodSFBulkErase(SpiChannel chn)

Parameters
SpiChannel chn - Spi channel

Return Value
none

Performs a bulk erase of the entire PmodSF setting all bits to 1. For a complete description see Bulk Erase (BE) in the M25P16/M25P128 reference manual.

Notes:
(1) Blocks while Write In Progress bit is set prior to performing operation

void PmodSFClearStatusRegBits(SpiChannel chn,uint8_t bitMask)

Parameters
SpiChannel chn - Spi channel
uint8_t bitMask - bitmask to apply to status register

Return Value
none

Status register is read in, a bitwise OR is performed on the bitmask passed in,
an AND operation is performed on the value of the status register and the bitmask,
this value is then written back to the status register. See appendix A Table 1: PMODSF Status Register Format for register values.

Notes:
(1) Blocks while Write In Progress bit is set prior to performing operation

void PmodSFSetStatusRegBits(SpiChannel chn,uint8_t bitMask)

Parameters
SpiChannel chn - Spi channel
uint8_t bitMask - bitmask to apply to status register

Return Value
none

Status register is read in, an OR operation is performed on the value of the status register, this value is then written back to the status register. See appendix A Table 1: PMODSF Status Register Format for register values.

Notes:
(1) Blocks while Write In Progress bit is set prior to performing operation

void PmodSFSectorErase(SpiChannel chn,uint32_t address)

Parameters

SpiChannel chn - Spi Channel
uint32_t address - 24-bit address contained in sector to erase

Return Value

none

Erases a sector on the PmodSF on the specified channel containing the 24-bit address passed in. For a complete description see Sector Erase (SE) and Memory Organization in the M25P16/M25P128 reference manual.

Notes:

(1) Blocks while Write In Progress bit is set prior to performing operation

void PmodSFSendCommand(SpiChannel chn,uint8_t command)

Parameters

SpiChannel chn - Spi channel
uint8_t command - 8-bit command from Appendix B, Command Macros

Return Value

none

SS is driven low, the 8 bit command is shifted out, 1 dummy byte is shifted in, SS is driven high. See Appendix B, Command Macros for commands and M25P16/M25P128 reference manual for additional details.

PmodSF Additional Information

Appendix A - Registers and Byte Layout

Table 1: PMODSF Status Register Format

7	6	5	4	3	2	1	0
PMODSF_SR_SRWD	0	0	PMODSF_SR_BP2	PMODSF_SR_BP1	PMODSF_SR_BP0	PMODSF_SR_WEL	PMODSF_SR_WIP

**See Appendix B Table 2 for values and descriptions

Table 2: Manufacturer ID/Memory Type/Memory Capacity

	Byte 3	Byte 2	Byte0
Pmod	MF ID	Dev ID Memory Type	 Memory Capacity
PmodSF-16	20h	20h	15h
PmodSF-128	20h	20h	18h

**See Appendix B Table 3 for values and descriptions

Appendix B (Values defined as macros)

Table 1: Command Macros		
Command	Macro	Value
Write Enable (WREN)	PMODSF_WRITE_ENABLE	0x06
Write Disable (WRDI)	PMODSF_WRITE_DISABLE	0x04
Read Identification (RDID)	PMODSF_READ_ID	0x9F
Read Status Register (RDSR)	PMODSF_READ_STATUS_REG	0x05
Write Status Register (WRSR)	PMODSF_WRITE_STATUS_REG	0x01
Read Data Bytes (READ)	PMODSF_READ_DATA_BYTES	0x03
Page Program (PP)	PMODSF_PAGE_PGM	0x02
Sector Erase (SE)	PMODSF_SECTOR_ERASE	0xD8
Bulk Erase (BE)	PMODSF_BULK_ERASE	0xC7
Deep Power Down	PMODSF_DEEP_POWER_DOWN	0xB9
Release From Deep Power Down and Read Electronic Signature	PMODSF_RELEASE_FROM_DPD	0xAB

Table 2: Status Register Macros		
Macro	Value	Description
PMODSF_SR_SRWD	0x80	Status Register Write Protect
PMODSF_SR_BP2	0x10	See Protected Area Sizes in Reference Manual
PMODSF_SR_BP1	0x8	See Protected Area Sizes in Reference Manual
PMODSF_SR_BP0	0x4	See Protected Area Sizes in Reference Manual
PMODSF_SR_WEL	0x2	Write Enable Latch Bit
PMODSF_SR_WIP	0x1	Write In Progress Bit
PMODSF_SR_SRWD	0x80	Status Register Write Protect

Table 3: Miscellaneous Macro Values		
Macro	Value	Description
PMODSF_PAGE_LEN	256	Maximum Size of a Page Write is 256 Bytes (0 - 255)
PMODSF_16_MBIT	0x15	PmodSFReadID - Memory Capacity Byte Denoting 16 Mb Flash
PMODSF_128_MBIT	0x18	PmodSFReadID - Memory Capacity Byte Denoting 128 Mb Flash
PMODSD_MEM_CAPACITY_BYTE	0	Byte Position of Memory Capacity Byte in Value Returned From PmodSFReadID
PMODSD_MEM_TYPE_BYTE	1	Byte Position of Memory Type Byte in Value Returned From PmodSFReadID
PMODSD_MFID_BYTE	2	Byte Position of Manufacturer's ID Byte in Value Returned From PmodSFReadID

Appendix C (PmodSF Use Example)

```
/* ------------------------------------------------------------ */
/*            PIC32 Configuration Settings                      */
/* ------------------------------------------------------------ */
#pragma config FPLLMUL = MUL_20, FPLLIDIV = DIV_2, FPLLODIV = DIV_1
#pragma config FWDTEN = OFF
#pragma config POSCMOD = HS, FNOSC = PRIPLL
#pragma config FPBDIV = DIV_2

#include <stdint.h>
#include <plib.h>
#include <pmodlib.h>

#define SYSTEM_CLOCK (80000000L)            //System clock speed (8 MHz Crystal/ FPLLIDIV * FPLLMUL / FPLLODIV)
#define PB_CLOCK (SYSTEM_CLOCK/2)           //Peripheral bus clock
#define SPI_BITRATE 625000                  //Bit rate for SPI port

int main(void)
{
        uint8_t bytesOut[PMODSF_PAGE_LEN];
        uint8_t bytesIn[PMODSF_PAGE_LEN];

        //Set all bytes to 5;
        memset(bytesOut,5,(sizeof(uint8_t) * PMODSF_PAGE_LEN));

        PmodSFInit(SPI_CHANNEL2,PB_CLOCK,SPI_BITRATE); //Init SPI Channel 2

        PmodSFSectorErase(SPI_CHANNEL2,0x0); //Erase all contents of sector 0(sets all bits to 1)

        PmodSFPageProgram(SPI_CHANNEL2,PMODSF_PAGE_LEN,bytesOut,0x100); //Write page to address 0x100

        PmodSFReadBytes(SPI_CHANNEL2,PMODSF_PAGE_LEN,bytesIn,0x100); //Read page from address 0x100

        //Byte values in bytesIn should equal those in bytesOut
}
```