

SPI™ Master Library Module (Interrupt-driven)

1.	Introduction	2
2.	Module Features	2
3.	List of Component Modules.....	3
4.	Using the Library Module in a Project	3
5.	List of Shared Parameters	4
	<i>Shared Data Bytes.....</i>	<i>4</i>
	<i>Shared Functions.....</i>	<i>4</i>
	<i>Shared Macros.....</i>	<i>4</i>
6.	Functions.....	5
7.	Macros.....	7
8.	Error and Status Flags	9

1. Introduction

The SPIMInt is a general-purpose library module. It configures the MSSP/SSP/BSSP module in the Master mode and helps in communicating with the SPITM Slave.

The module code is linkable and relocatable, which provides the user the facility to use it without modifications.

It provides the interrupt-based operation and has its own Tx & Rx buffer, which provides maximum benefit of parallel processing.

By using this Module, one can write his application to interact with any of the SPI Slaves like EEPROM, ADC, Digital Potentiometer, LCD, etc.

The module allows the user to concentrate more on his application's development by providing these library functions.

2. Module Features

It supports following features: -

- It provides simple and primitive functions to communicate with the SPI Slave.
- User defined length of the Tx & Rx Buffer.
- Interrupt driven transmission and reception.
- It generates Error flags on the occurrence of an error. All error conditions are passed through the 'SPIMIntStatus' Register.

3. List of Component Modules

<code>SPIMInt.P16.ex.txt</code>	This is an example file developed to demonstrate the use of the library functions for the PIC16 family.
<code>SPIMInt.P18.ex.txt</code>	This is an example file developed to demonstrate the use of the library functions for the PIC18 family.
<code>SPIMInt.asm</code>	This is the SPI Master code implementation file. <u>One needs to include this file in their project.</u>
<code>16SPIMI.asm</code>	This is the SPI Master code implementation file for the PIC16 family. The <code>SPIMInt.asm</code> file will include this file if the PIC16 family processor is used.
<code>18SPIMI.asm</code>	This is the SPI Master code implementation file for the PIC18 family. The <code>SPIMInt.asm</code> file will include this file if the PIC18 family processor is used.
<code>SPIMInt.inc</code>	This file contains the definitions of all the shared parameters and the macros. <u>One needs to include this in the Assembly file</u> where the library functions and macros are called. This file takes care of the definitions of all the Extern Global parameters, so one can directly call the library routines in their program.
<code>P16xxx.inc</code>	This is the general purpose processor definition file for the PIC16 family
<code>P18xxx.inc</code>	This is the general purpose processor definition file for the PIC18 family

4. Using the Library Module in a Project

Please follow the steps below to use this library module in your project.

1. Use the Application Maestro™ software to configure the module as required.
2. At the 'Generate Files' step, save the output to the directory where your project code resides.
3. Launch the MPLAB® IDE, and open the project's workspace.
4. Verify that the Microchip language tool suite is selected (*Project>Select Language Toolsuite*).
5. In the Workspace view, right-click on the "Source Files" node. Select the "Add Files" option. Select the file `SPIMInt.asm` and click **OK**.
6. Now right-click on the "Linker Scripts" node and select "Add Files". Add the appropriate linker file (`.lkr`) for the project's target microcontroller.
7. Add any other files that the project may require. Save and close the project.
8. In your main source (assembler) file, add `include` directive at the head of the code listing to include the file `SPIMInt.inc`. By doing so, all files required to make the generated code work in your project will be included by reference when you build the project.
9. To use the module in your application, invoke the functions or the macros as needed.

5. List of Shared Parameters

Shared Data Bytes

VSPIMIntStatus	It is the Error/Status register. The details of each bit of this register is explained in Section 8
----------------	--

Shared Functions

SPIMIntInit	It is used for the Synchronous Serial Port Initialization It initializes the Port according to the options opted through the Application Maestro.
SPIMIntPut	It is used for transmitting a byte on the SPI Bus.
SPIMIntSetGetCount	It is used for specifying the number of bytes to be received.
SPIMIntGet	It is used for reading the received byte.
SPIMIntISR	It is called from the interrupt handler.
SPIMIntDiscardRxBuf	It is used for discarding the buffer.

Shared Macros

MSPIMIntDisable	It disables the Synchronous Serial Port.
mSPIMIntSetClockIdleState	This sets the Idle state of the Clock line, 'Hi' (High) or 'Lo' (Low).
mSPIMIntSetTransmitOnClockEdge	This sets the Clock edge at which the data is to be transmitted, 'IdleToActive' or 'ActiveToIdle'.
mSPIMIntSetSampleAtDataOut	This sets at which phase of the Data Out, the Data In should be sampled, 'Mids' (Middle) or 'Ends' (End).
mSetSPIMIntHighPriority	This sets the interrupt priority of the SSP as High.
mSetSPIMIntLowPriority	This sets the interrupt priority of the SSP as Low.

6. Functions

Function	SPIMIntInit
Pre-conditions	The TRIS bits of the SCK pin and the SDO pin should be made output. The TRIS bit of the SDI pin should be made input. The TRIS bit of the Slave Chip Select pin (if any used) should be made output.
Overview	This function is used for initializing the MSSP/SSP/BSSP module. It initializes the module according to the Application Maestro options.
Input	Application Maestro options
Output	None
Side Effects	Bank selection bits and 'W' register are changed
Stack Requirement	1 level deep
Maximum T-Cycles taken	21 Cycles by the PIC16 family 18 Cycles by the PIC18 family

Function	SPIMIntPut		
Pre-conditions	The function 'SPIMIntInit' should have been called.		
Overview	This function sends the byte in the 'W' Reg. over the SPI bus if the bus is free, else saves the byte in the buffer.		
Input	'W' Register.		
Output	If 'W' is written into buffer then, the bit 'vSPIMIntStatus<SPIMTxBufEmpty>' is cleared. If the buffer gets full then, the bit 'vSPIMIntStatus<SPIMTxBufFull>' is set.		
Side Effects	Bank selection bits and 'W' register are changed		
Stack Requirement	1 level deep		
Maximum T-Cycles taken	Tx & Rx Buffer lengths	PIC16 family	PIC18 family
	Tx Buf > 1	49 Cycles	46 Cycles
	Tx Buf = 1	9 Cycles	9 Cycles

Function	SPIMIntGet		
Pre-conditions	The function 'SPIMIntSetGetCount' should have been called and the bit 'vSPIMIntStatus<SPIMRxBufEmpty>' should be read '0'.		
Overview	This function reads the byte received.		
Input	None		
Output	'w' Register. The bit 'vSPIMIntStatus<SPIMRxBufFull>' is cleared. If the buffer gets empty then, the bit 'vSPIMIntStatus<SPIMRxBufEmpty>' is set.		
Side Effects	Bank selection bits and 'w' register are changed		
Stack Requirement	1 level deep		
Maximum T-Cycles taken	Tx & Rx Buffer lengths	PIC16 family	PIC18 family
	Tx Buf > 1	47 Cycles	41 Cycles
	Tx Buf = 1	7 Cycles	7 Cycles

Function	SPIMIntSetGetCount		
Pre-conditions	The function 'SPIMIntStart' should have been called.		
Overview	This function sets the number of bytes to be received.		
Input	'w' Register.		
Output	None.		
Side Effects	Bank selection bits are changed		
Stack Requirement	1 level deep		
Maximum T-Cycles taken	17 Cycles by the PIC16 family 13 Cycles by the PIC18 family		

Function	SPIMIntISR		
Pre-conditions	This function should be called from the interrupt handler.		
Overview	This is the interrupt service routine for SSPIF. This handles the transmission and reception of the bytes.		
Input	None		
Output	While transmission is going on, if the buffer is full then, the bit vSPIMIntStatus<SPIMTxBufFull> is set. While reception is going on, if the buffer is empty then, the bit vSPIMIntStatus<SPIMRxBufEmpty> is set.		
Side Effects	Bank selection bits and 'w' register are changed		
Stack Requirement	2 level deep		
Maximum T-Cycles taken	Tx & Rx Buffer lengths	PIC16 family	PIC18 family
	Tx Buf >1, Rx Buf >1	106 Cycles	99 Cycles
	Tx Buf =1, Rx Buf >1	75 Cycles	67 Cycles
	Tx Buf >1, Rx Buf =1	69 Cycles	63 Cycles
	Tx Buf =1, Rx Buf =1	38 Cycles	31 Cycles

Function	SPIMIntDiscardBuf		
Pre-conditions	The function 'SPIMIntPut' is called or the function 'SPIMIntSetGetCount' is called, and the bit 'vSPIMIntStatus<SPIMRxBufEmpty>' is cleared.		
Overview	This function flushes the buffer.		
Input	None		
Output	None.		
Side Effects	Bank selection bits are changed		
Stack Requirement	1 level deep		
Maximum T-Cycles taken	10 Cycles by the PIC16 family 9 Cycles by the PIC18 family		

7. Macros

Macro	<code>mSPIMIntSetClockIdleState</code>
Overview	This Macro is used to specify the Idle State of the Clock pin (SCK).
Input	The Clock pin Idle state: ' Hi ' (for High) ' Lo ' (for Low) Example- To set the Idle State of the Clock pin as High <code>mSPIMIntSetClockIdleState Hi</code>
Output	None
Side Effects	Bank selection bits are changed.
Stack Requirement	None
Maximum T-Cycles taken	3 Cycles by the PIC16 family 1 Cycle by the PIC18 family
Macro	<code>mSPIMIntSetTransmitOnClockEdge</code>
Overview	This Macro is used to specify on what edge of the Clock the transmission should take place.
Input	Transmission at the clock edge: ' IdleToActive ' ' ActiveToIdle ' Example- To transmit on the Clock edge Idle to Active <code>mSPIMIntSetTransmitOnClockEdge IdleToActive</code>
Output	None
Side Effects	Bank selection bits are changed.
Stack Requirement	None
Maximum T-Cycles taken	3 Cycles by the PIC16 family 1 Cycle by the PIC18 family
Macro	<code>mSPIMIntSetSampleAtDataOut</code>
Overview	This Macro is used to specify the sampling phase of the Data In with respect to the Data Out.
Input	Sampling phase with respect to the Data Out: ' Mids ' (Middle) ' Ends ' (End). Example- To sample at mid of the Data Out <code>mSPIMIntSetSampleAtDataOut Mids</code>
Output	None
Side Effects	Bank selection bits are changed.
Stack Requirement	None
Maximum T-Cycles taken	3 Cycles by the PIC16 family 1 Cycle by the PIC18 family

Macro	MSPIMIntDisable
Overview	Pre-conditions- The function 'SPIMIntStop' should have been called and the bit 'vSPIMIntStatus<SPIMBusy>' is '0'. Disables the MSSP/SSP/BSSP module.
Input	None
Output	None
Side Effects	Bank selection bits are changed.
Stack Requirement	None
Maximum T-Cycles taken	3 Cycles by the PIC16 family 1 Cycle by the PIC18 family

Macro	mSetSPIMIntHighPriority (Valid only for the PIC18 family devices).
Overview	This sets the interrupt priority of SSP as High.
Input	None
Output	None
Side Effects	Bank selection bits are changed.
Stack Requirement	None
Maximum T-Cycles taken	1 Cycle by the PIC18 family

Macro	mSetSPIMIntLowPriority (Valid only for the PIC18 family devices).
Overview	This sets the interrupt priority of SSP as Low.
Input	None
Output	None
Side Effects	Bank selection bits are changed.
Stack Requirement	None
Maximum T-Cycles taken	1 Cycle by the PIC18 family

8. Error and Status Flags

All errors/statuses are set as a content of the 'vSPIMIntStatus' Register. The individual errors/statuses are unique. Please refer the list below for the information.

SPIMBusy	This indicates that the SPI Module is busy doing some preassigned work.
SPIMTxBufFull	This indicates that the transmit buffer is full.
SPIMTxBufEmpty	This indicates that the transmit buffer is empty.
SPIMRxBufFull	This indicates that the receive buffer is full.
SPIMRxBufEmpty	This indicates that the receive buffer is empty.
SPIMRxBufOverflow	This indicates that the receive buffer is over flowing.

MPLAB is a registered trademark of Microchip Technology INC. in the U.S.A and other countries. Application Maestro is a trademark of Microchip Technology INC. in the U.S.A and other countries. All other trademarks mentioned herein are property of their respective companies.