

SPI™ Slave 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.....	6
8.	Error and Status Flags	9

1. Introduction

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

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 Master.

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 Master.
- 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 'SPISIntStatus' Register.

3. List of Component Modules

<code>SPISInt.P16.ex.txt</code>	This is an example file developed to demonstrate the use of the library functions for the PIC16 family.
<code>SPISInt.P18.ex.txt</code>	This is an example file developed to demonstrate the use of the library functions for the PIC18 family.
<code>SPISInt.asm</code>	This is the SPI Slave code implementation file. <u>One needs to include this file in their project.</u>
<code>16SPISInt.asm</code>	This is the SPI Slave code implementation file for the PIC16 family. The <code>SPISInt.asm</code> file will include this file if the PIC16 family processor is used.
<code>18SPISInt.asm</code>	This is the SPI Slave code implementation file for the PIC18 family. The <code>SPISInt.asm</code> file will include this file if the PIC18 family processor is used.
<code>SPISInt.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 Extern Global parameters, so one can directly call the library routines in their program.
<code>P16xxx.inc</code>	General purpose processor definition file for the PIC16 family.
<code>P18xxx.inc</code>	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 `SPISInt.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 the `include` directive at the head of the code listing to include the file `SPISInt.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 macros as needed.

5. List of Shared Parameters

Shared Data Bytes

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

Shared Functions

SPISIntInit	It is used for the Synchronous Serial Port Initialization It initializes the Port according to the options opted through the Application Maestro software.
SPISIntISR	It is called from the interrupt handler.
SPISIntDiscardRxBuf	It is used for discarding the buffer.

Shared Macros

mSPISIntSetClockIdleState	This sets the Idle state of the Clock line, 'Hi' (High) or 'Lo' (Low).
mSPISIntSetTransmitOnClockEdge	This sets the Clock edge at which the data is to be Transmitted, 'IdleToActive' or 'ActiveToIdle'.
mSPISIntPut	It is used for transmitting a byte on the SPI Bus. (This macro is included only if, in Application Maestro option Transmit Buffer Size , opted is non-zero).
mSPISIntGet	It is used for reading the received byte. (This macro is included only if, in the Application Maestro option Receive Buffer Size , opted is non-zero).
mSPISIntSaveReceivedByte	Sets a flag 'Save' to indicate that the received byte should be saved. (This macro is included only if, in the Application Maestro option Receive Buffer Size , opted is a non-zero).
mSPISIntDiscardReceivedByte	Clears the flag 'Save' to indicate that the received byte should not be saved. (This macro is included only if, in the Application Maestro option Receive Buffer Size , opted is a non-zero) .
mSPISIntDisable	Disables the Synchronous Serial Port.
mSetSPISIntHighPriority	This sets the interrupt priority of SSP as High.
mSetSPISIntLowPriority	This sets the interrupt priority of SSP as Low.

6. Functions

Function	SPISIntInit
Preconditions	The TRIS bits of the SDO pin should be made output. The TRIS bit of the pins SCK and SDI should be made input. The TRIS bit of the Slave Select pin (if used) should be made input.
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	15 Cycles by the PIC16 family 12 Cycles by the PIC18 family

Function	SPISIntISR		
Preconditions	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 bytes.		
Input	None		
Output	While transmission is going on, if the buffer is full then, the bit <code>vSPISIntStatus<SPISTxBufFull></code> is set. While reception is going on, if buffer is empty then, the bit <code>vSPISIntStatus<SPISRxBufEmpty></code> 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	100 Cycles	92 Cycles
	Tx Buf =1, Rx Buf >1	65 Cycles	60 Cycles
	Tx Buf >1, Rx Buf =1	60 Cycles	54 Cycles
	Tx Buf =0, Rx Buf >1	62 Cycles	58 Cycles
	Tx Buf >1, Rx Buf =0	56 Cycles	49 Cycles
	Tx Buf =1, Rx Buf =1	25 Cycles	22 Cycles
	Tx Buf =0, Rx Buf =1	22 Cycles	20 Cycles
	Tx Buf =1, Rx Buf =0	21 Cycles	17 Cycles

Function	SPISIntDiscardBuf
Preconditions	The bit <code>'vSPISIntStatus<SPISRxBufEmpty>'</code> should be '0'.
Overview	This function flushes the buffer.
Input	None
Output	None
Side Effects	Bank selection bits and 'W' register 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	mSPISIntSetClockIdleState		
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 Clock pin Idle State as High, mSPISIntSetClockIdleState Hi		
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	mSPISIntSetTransmitOnClockEdge		
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 mSPISIntSetTransmitOnClockEdge IdleToActive		
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	mSPISIntPut		
Overview	Preconditions: In the Application Maestro option the Transmit Buffer size opted should be a non-zero and the function 'SPISIntInit' should have been called. 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 the buffer then, the bit 'vSPISIntStatus<SPISTxBufEmpty>' is cleared. If the buffer gets full then, the bit 'vSPISIntStatus<SPISTxBufFull>' is set.		
Side Effects	Bank selection bits and 'W' register are changed		
Stack Requirement	1 level deep		
Maximum T-Cycles taken		PIC16 family	PIC18 family
	Tx Buf > 1	42 Cycles	38 Cycles
	Tx Buf = 1	3 Cycles	3 Cycles

Macro	mSPISIntGet		
Overview	Preconditions: In the Application Maestro option Receive Buffer size , opted should be a non-zero and the bit <code>vSPISIntStatus<SPISRxBufEmpty></code> should return a '0'. This function reads the byte received.		
Input	None		
Output	'W' Register will have the read byte. The bit ' <code>vSPISIntStatus<SPISRxBufFull></code> ' is cleared. If the buffer gets empty then, the bit ' <code>vSPISIntStatus<SPISRxBufEmpty></code> ' is set.		
Side Effects	Bank selection bits and 'W' register are changed		
Stack Requirement	1 level deep		
Maximum T-Cycles taken	Rx Buf > 1	PIC16 family 45 Cycles	PIC18 family 35 Cycles
	Rx Buf = 1	3 Cycles	3 Cycles
Macro	mSPISIntSaveReceivedByte		
Overview	Sets a flag 'Save' to indicate that the received byte should be saved. (This macro is included only if in the Application Maestro option, the Receive Buffer Size opted is a non-zero) .		
Input	None		
Output	'SPISIntStatus'		
Side Effects	Bank selection bits are changed.		
Stack Requirement	None		
Maximum T-Cycles taken	1 Cycle		
Macro	mSPISIntDoNotSaveReceivedByte		
Overview	Clears the flag 'Save' to indicate that the received byte should not be saved. (This macro is included only if in the Application Maestro option, the Receive Buffer Size opted is a non-zero) .		
Input	None		
Output	'SPISIntStatus'		
Side Effects	Bank selection bits are changed.		
Stack Requirement	None		
Maximum T-Cycles taken	1 Cycle		
Macro	MSPISIntDisable		
Overview	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	mSetSPISIntHighPriority (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	mSetSPISIntLowPriority (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 'W' Register. The individual errors/statuses are unique. Please refer the list below for the information.

SPISSave	This indicates that the received byte should be saved in the buffer.
SPIS TxBufFull	This indicates that the transmit buffer is full.
SPIS TxBufEmpty	This indicates that the transmit buffer is empty.
SPIS RxBufFull	This indicates that the receive buffer is full.
SPIS RxBufEmpty	This indicates that the receive buffer is empty.
SPIS RxBufOverFlow	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.