

Proton+ PIC18 SPI™ Macros

Table of Contents

Introduction	2
Version vs. Devices	3
CloseSPI	4
CloseSPI1	4
CloseSPI2	4
DataReadySPI	5
DataReadySPI1	5
DataReadySPI2	5
OpenSPI	6
OpenSPI1	6
OpenSPI2	6
ReadSPI	7
ReadSPI1	7
ReadSPI2	7
WriteSPI	8
WriteSPI1	8
WriteSPI2	8
Example of Use	8

Proton+ PIC18 SPI™ Macros

Introduction

The following routines are provided for devices with a single SPI™ peripheral:
Single SPI™ Peripheral Macros:

Macro	Description
CloseSPI	Disable the SSP module used for SPI™ communications.
DataReadySPI	Determine if a new value is available from the SPI buffer.
OpenSPI	Initialise the SSP module used for SPI communications.
ReadSPI	Read a byte from the SPI bus.
WriteSPI	Write a byte to the SPI bus.

The following routines are provided for devices with multiple SPI peripherals:
Multiple SPI™ Peripheral Macros:

Macro	Description
CloseSPIx	Disable the SSPx module used for SPI™ communications.
DataReadySPIx	Determine if a new value is available from the SPIx buffer.
OpenSPIx	Initialise the SSPxx module used for SPI communications.
ReadSPIx	Read a byte from the SPIx bus.
WriteSPIx	Write a byte to the SPIx bus.

Based on the different control registers, configuration bits and their positions in the control register, all PIC18 devices are divided into following different versions. Wherever required, separate Macros have been designed to support these versions, so before calling the Macros care has to be taken to know the version of the configured device and to call the appropriate Macro with correct number of arguments.

Proton+ PIC18 SPI™ Macros

Below is the table to find the SPI version for the chosen device:

Version vs. Devices

Version name	Device number
SPI_V1	18C242, 18C252, 18C442, 18C452, 18F242, 18F252, 18F442, 18F452, 18F248, 18F258, 18F448, 18F458, 18F2439, 18F2539, 18F4439, 18F4539, 18C601, 18C801, 18C658, 18C858, 18F6620, 18F6720, 18F8620, 18F8720, 18F6520, 18F8520, 18F2220, 18F2320, 18F4220, 18F4320, 18F2420, 18F2520, 18F4420, 18F4520, 18F2423, 18F2523, 18F4423, 18F4523, 18F2455, 18F2550, 18F4455, 18F4550, 18F2480, 18F2580, 18F4480, 18F4580, 18F2410, 18F2510, 18F2515, 18F2610, 18F4410, 18F4510, 18F4515, 18F4610, 18F2525, 18F2620, 18F4525, 18F4620, 18F2585, 18F2680, 18F4585, 18F4680, 18F2682, 18F2685, 18F4682, 18F4685, 18F2221, 18F2321, 18F4221, 18F4321, 18F6310, 18F6410, 18F8310, 18F8410, 18F6390, 18F6490, 18F8390, 18F8490, 18F6585, 18F6680, 18F8585, 18F8680, 18F6525, 18F6621, 18F8525, 18F8621, 18F63J11, 18F64J11, 18F65J11, 18F83J11, 18F84J11, 18F85J11, 18F63J90, 18F64J90, 18F65J90, 18F83J90, 18F84J90, 18F85J90, 18F23K20, 18F24K20, 18F25K20, 18F43K20, 18F44K20, 18F45K20, 18F13K50, 18LF13K50, 18F14K50, 18LF14K50, 18F13K22, 18F14K22, 18LF13K22, 18LF14K22, 18F66J90, 18F67J90, 18F86J90, 18F87J90
SPI_V2	18F24J10, 18F25J10, 18F66J60, 18F66J65, 18F67J60, 18F86J60, 18F86J65, 18F87J60
SPI_V3	18F6527, 18F6622, 18F6627, 18F6722, 18F8527, 18F8622, 18F8627, 18F8722, 18F44J10, 18F45J10, 18F65J10, 18F65J15, 18F66J10, 18F66J15, 18F67J10, 18F85J10, 18F85J15, 18F86J10, 18F86J15, 18F87J10, 18F96J60, 18F96J65, 18F97J60, 18F66J11, 18F66J16, 18F67J11, 18F86J11, 18F86J16, 18F87J11, 18F65J50, 18F66J50, 18F66J55, 18F67J50, 18F85J50, 18F86J50, 18F86J55, 18F87J50
SPI_V4	18F2331, 18F2431, 18F4331, 18F4431
SPI_V5	18F24J11, 18F25J11, 18F26J11, 18F44J11, 18F45J11, 18F46J11, 18F24J50, 18F25J50, 18F26J50, 18F44J50, 18F45J50, 18F46J50, 18LF24J11, 18LF25J11, 18LF26J11, 18LF44J11, 18LF45J11, 18LF46J11, 18LF24J50, 18LF25J50, 18LF26J50, 18LF44J50, 18LF45J50, 18LF46J50

Proton+ PIC18 SPI™ Macros

CloseSPI

CloseSPI1

CloseSPI2

Macro: Disable the SSPx module.

Include: `SPI.inc`

Prototype: For `SPI_V1,V4`
`CloseSPI()`

For `SPI_V2,V3,V5`
`CloseSPI1()`

For `SPI_V2,V3,V5`
`CloseSPI2()`

Remarks: This Macro disables the SSPx module. Pin I/O returns under the control of the appropriate TRIS and LAT registers.

Proton+ PIC18 SPI™ Macros

DataReadySPI

DataReadySPI1

DataReadySPI2

Macro: Determine if the SSPBUFx contains data.

Include: `SPI.inc`

Prototype: For `SPI_V1,V4`
`DataReadySPI`

For `SPI_V2,V3,V5`
`DataReadySPI1`

For `SPI_V2,V3,V5`
`DataReadySPI2`

Remarks: This Macro determines if there is a byte to be read from the SSPBUFx register.

Return Value: 0 if there is no data in the SSPBUFx register
1 if there is data in the SSPBUFx register

Code Example: `While DataReadySPI = 0 : Wend`

Proton+ PIC18 SPI™ Macros

OpenSPI

OpenSPI1

OpenSPI2

Macro: Initialise the SSPx module.

Include: SPI.inc

Prototype:

```
For SPI_V1,V4
    OpenSPI(pSyncMode, pBusMode, pSmpPhase)

For SPI_V2,V3,V5
    OpenSPI1(pSyncMode, pBusMode, pSmpPhase)

For SPI_V2,V3,V5
    OpenSPI2(pSyncMode, pBusMode, pSmpPhase)
```

Arguments: ***pSyncMode***
One of the following values, defined in SPIdefs.inc:

SPI_FOSC_4	SPI Master mode, clock = Fosc / 4
SPI_FOSC_16	SPI Master mode, clock = Fosc / 16
SPI_FOSC_64	SPI Master mode, clock = Fosc / 64
SPI_FOSC_TMR2	SPI Master mode, clock = TMR2 output / 2
SLV_SSON	SPI Slave mode, /SS pin control enabled
SLV_SSOFF	SPI Slave mode, /SS pin control disabled

pBusMode
One of the following values, defined in SPIdefs.inc:

MODE_00	Setting for SPI bus Mode 0,0
MODE_01	Setting for SPI bus Mode 0,1
MODE_10	Setting for SPI bus Mode 1,0
MODE_11	Setting for SPI bus Mode 1,1

pSmpPhase
One of the following values, defined in SPIdefs.inc:

SMPEND	Input data sample at end of data out
SMPMID	Input data sample at middle of data out

Remarks: This Macro sets up the SSPx module for use with a SPIx bus device.

Code Example: `OpenSPI(SPI_FOSC_16, MODE_00, SMPEND)`

Proton+ PIC18 SPI™ Macros

ReadSPI

ReadSPI1

ReadSPI2

Macro: Read a byte from the SPIx bus.

Include: `SPI.inc`

Prototype: For `SPI_V1,V4`
`ReadSPI()`

For `SPI_V1,V2,V3,V4`
`ReadSPI1()`
`ReadSPI2()`

Remarks: This Macro initiates a SPIx bus cycle for the acquisition of a byte of data.

Return Value: This Macro returns a byte of data read during a SPIx read cycle.

Code Example: `Dim Var as Byte`
`Var = ReadSPI()`

Proton+ PIC18 SPI™ Macros

WriteSPI

WriteSPI1

WriteSPI2

Macro: Write a byte to the SPIx bus.

Include: SPI.inc

Prototype: For SPI_V1,V4
`WriteSPI(pDataOut)`

For SPI_V2,V3,V5
`WriteSPI1(pDataOut)`

For SPI_V2,V3,V5
`WriteSPI2(pDataOut)`

Arguments: *pDataOut*
Value to be written to the SPIx bus.

Remarks: This Macro writes a single data byte out.

Return Value: None

Code Example: **Dim Var as Byte**

`WriteSPI(Var)`

Example of Use

The following example demonstrates the use of an SSP macro.

```
Device = 18F452
Xtal = 4
Optimiser_Level = 1
Dead_Code_Remove = On
```

```
Include "SPI.inc"
```

```
Dim SPI_data as Byte
```

Main:

```
OpenSPI(SPI_FOSC_64 , MODE_01 , SMPMID)
WriteSPI($55)
SPI_data = ReadSPI()
DataReadySPI()
CloseSPI()
Stop
```