

Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "<u>Embedded -</u> <u>Microcontrollers</u>"

Details

E·XFI

Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	64MHz
Connectivity	I ² C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, HLVD, POR, PWM, WDT
Number of I/O	35
Program Memory Size	8KB (4K x 16)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	2.3V ~ 5.5V
Data Converters	A/D 30x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-VQFN Exposed Pad
Supplier Device Package	44-QFN (8x8)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f43k22t-i-ml

Email: info@E-XFL.COM

Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong

PIC18(L)F2X/4XK22

FIGURE 1: 28-PIN PDIP, SOIC, SSOP DIAGRAM





2.5.3 LP, XT, HS MODES

The LP, XT and HS modes support the use of quartz crystal resonators or ceramic resonators connected to OSC1 and OSC2 (Figure 2-6). The mode selects a low, medium or high gain setting of the internal inverter-amplifier to support various resonator types and speed.

LP Oscillator mode selects the lowest gain setting of the internal inverter-amplifier. LP mode current consumption is the least of the three modes. This mode is best suited to drive resonators with a low drive level specification, for example, tuning fork type crystals.

XT Oscillator mode selects the intermediate gain setting of the internal inverter-amplifier. XT mode current consumption is the medium of the three modes. This mode is best suited to drive resonators with a medium drive level specification.

HS Oscillator mode offers a Medium Power (MP) and a High Power (HP) option selectable by the FOSC<3:0> bits. The MP selections are best suited for oscillator frequencies between 4 MHz and 16 MHz. The HP selection has the highest gain setting of the internal inverter-amplifier and is best suited for frequencies above 16 MHz. HS mode is best suited for resonators that require a high drive setting.

FIGURE 2-6: QUARTZ CRYSTAL OPERATION (LP, XT OR HS MODE)



Note 1: Quartz crystal characteristics vary according to type, package and manufacturer. The user should consult the manufacturer data sheets for specifications and recommended application.

- **2:** Always verify oscillator performance over the VDD and temperature range that is expected for the application.
- **3:** For oscillator design assistance, refer to the following Microchip Application Notes:
 - AN826, "Crystal Oscillator Basics and Crystal Selection for rfPIC[®] and PIC[®] Devices" (DS00826)
 - AN849, "Basic PIC[®] Oscillator Design" (DS00849)
 - AN943, "Practical PIC[®] Oscillator Analysis and Design" (DS00943)
 - AN949, "Making Your Oscillator Work" (DS00949)



CERAMIC RESONATOR OPERATION (XT OR HS MODE)



3: An additional parallel feedback resistor (RP) may be required for proper ceramic resonator operation.

3.0 POWER-MANAGED MODES

PIC18(L)F2X/4XK22 devices offer a total of seven operating modes for more efficient power management. These modes provide a variety of options for selective power conservation in applications where resources may be limited (i.e., battery-powered devices).

There are three categories of power-managed modes:

- Run modes
- Idle modes
- Sleep mode

These categories define which portions of the device are clocked and sometimes, what speed. The Run and Idle modes may use any of the three available clock sources (primary, secondary or internal oscillator block). The Sleep mode does not use a clock source.

The power-managed modes include several powersaving features offered on previous PIC[®] microcontroller devices. One of the clock switching features allows the controller to use the secondary oscillator (SOSC) in place of the primary oscillator. Also included is the Sleep mode, offered by all PIC microcontroller devices, where all device clocks are stopped.

3.1 Selecting Power-Managed Modes

Selecting a power-managed mode requires two decisions:

- Whether or not the CPU is to be clocked
- The selection of a clock source

The IDLEN bit (OSCCON<7>) controls CPU clocking, while the SCS<1:0> bits (OSCCON<1:0>) select the clock source. The individual modes, bit settings, clock sources and affected modules are summarized in Table 3-1.

Modo	osco	CON Bits	Module	Clocking	Augilable Clask and Ossillator Source					
MODE	IDLEN ⁽¹⁾	SCS<1:0>	CPU	Peripherals	Available Clock and Oscillator Source					
Sleep	0	N/A	Off	Off	None – All clocks are disabled					
PRI_RUN	N/A	00	Clocked	Clocked	Primary – LP, XT, HS, RC, EC and Internal Oscillator Block ⁽²⁾ . This is the normal full-power execution mode.					
SEC_RUN	N/A	01	Clocked	Clocked	Secondary – SOSC Oscillator					
RC_RUN	N/A	1x	Clocked	Clocked	Internal Oscillator Block ⁽²⁾					
PRI_IDLE	1	00	Off	Clocked	Primary – LP, XT, HS, HSPLL, RC, EC					
SEC_IDLE	1	01	Off	Clocked	Secondary – SOSC Oscillator					
RC_IDLE	1	1x	Off	Clocked	Internal Oscillator Block ⁽²⁾					

TABLE 3-1: POWER-MANAGED MODES

3.1.1 CLOCK SOURCES

The SCS<1:0> bits allow the selection of one of three clock sources for power-managed modes. They are:

- the primary clock, as defined by the FOSC<3:0> Configuration bits
- the secondary clock (the SOSC oscillator)
- the internal oscillator block

3.1.2 ENTERING POWER-MANAGED MODES

Switching from one power-managed mode to another begins by loading the OSCCON register. The SCS<1:0> bits select the clock source and determine which Run or Idle mode is to be used. Changing these bits causes an immediate switch to the new clock source, assuming that it is running. The switch may also be subject to clock transition delays. Refer to **Section 2.11 "Clock Switching"** for more information.

Entry to the power-managed Idle or Sleep modes is triggered by the execution of a SLEEP instruction. The actual mode that results depends on the status of the IDLEN bit.

Depending on the current mode and the mode being switched to, a change to a power-managed mode does not always require setting all of these bits. Many transitions may be done by changing the oscillator select bits, or changing the IDLEN bit, prior to issuing a SLEEP instruction. If the IDLEN bit is already configured correctly, it may only be necessary to perform a SLEEP instruction to switch to the desired mode.

Note 1: IDLEN reflects its value when the SLEEP instruction is executed.

2: Includes HFINTOSC and HFINTOSC postscaler, as well as the LFINTOSC source.

TABLE 5-2:	REGISTER FILE SUMMARY FOR PIC18(L)F2X/4XK22 DEVICES (CONTINUED)
------------	---

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	<u>Value on</u> POR, BOR
F3Ah	ANSELC	ANSC7	ANSC6	ANSC5	ANSC4	ANSC3	ANSC2	_	—	1111 11
F39h	ANSELB	—	_	ANSB5	ANSB4	ANSB3	ANSB2	ANSB1	ANSB0	11 1111
F38h	ANSELA	—	_	ANSA5	_	ANSA3	ANSA2	ANSA1	ANSA0	1- 1111

Legend: x = unknown, u = unchanged, — = unimplemented, q = value depends on condition

Note 1: PIC18(L)F4XK22 devices only.

2: PIC18(L)F2XK22 devices only.

3: PIC18(L)F23/24K22 and PIC18(L)F43/44K22 devices only.

4: PIC18(L)F26K22 and PIC18(L)F46K22 devices only.

6.3.1 TABLAT – TABLE LATCH REGISTER

The Table Latch (TABLAT) is an 8-bit register mapped into the SFR space. The Table Latch register is used to hold 8-bit data during data transfers between program memory and data RAM.

6.3.2 TBLPTR – TABLE POINTER REGISTER

The Table Pointer (TBLPTR) register addresses a byte within the program memory. The TBLPTR is comprised of three SFR registers: Table Pointer Upper Byte, Table Pointer High Byte and Table Pointer Low Byte (TBLPTRU:TBLPTRH:TBLPTRL). These three registers join to form a 22-bit wide pointer. The low-order 21 bits allow the device to address up to 2 Mbytes of program memory space. The 22nd bit allows access to the device ID, the user ID and the Configuration bits.

The Table Pointer register, TBLPTR, is used by the TBLRD and TBLWT instructions. These instructions can update the TBLPTR in one of four ways based on the table operation. These operations on the TBLPTR affect only the low-order 21 bits.

6.3.3 TABLE POINTER BOUNDARIES

TBLPTR is used in reads, writes and erases of the Flash program memory.

When a TBLRD is executed, all 22 bits of the TBLPTR determine which byte is read from program memory directly into the TABLAT register.

When a TBLWT is executed the byte in the TABLAT register is written, not to Flash memory but, to a holding register in preparation for a program memory write. The holding registers constitute a write block which varies depending on the device (see Table 6-1). The 3, 4, or 5 LSbs of the TBLPTRL register determine which specific address within the holding register block is written to. The MSBs of the Table Pointer have no effect during TBLWT operations.

When a program memory write is executed the entire holding register block is written to the Flash memory at the address determined by the MSbs of the TBLPTR. The 3, 4, or 5 LSBs are ignored during Flash memory writes. For more detail, see **Section 6.6** "**Writing to Flash Program Memory**".

When an erase of program memory is executed, the 16 MSbs of the Table Pointer register (TBLPTR<21:6>) point to the 64-byte block that will be erased. The Least Significant bits (TBLPTR<5:0>) are ignored.

Figure 6-3 describes the relevant boundaries of TBLPTR based on Flash program memory operations.

TABLE 6-1: TABLE POINTER OPERATIONS WITH TBLRD AND TBLWT INSTRUCTIONS

Example	Operation on Table Pointer
TBLRD* TBLWT*	TBLPTR is not modified
TBLRD*+ TBLWT*+	TBLPTR is incremented after the read/write
TBLRD*- TBLWT*-	TBLPTR is decremented after the read/write
TBLRD+* TBLWT+*	TBLPTR is incremented before the read/write

FIGURE 6-3: TABLE POINTER BOUNDARIES BASED ON OPERATION



7.3 Reading the Data EEPROM Memory

To read a data memory location, the user must write the address to the EEADR register, clear the EEPGD control bit of the EECON1 register and then set control bit, RD. The data is available on the very next instruction cycle; therefore, the EEDATA register can be read by the next instruction. EEDATA will hold this value until another read operation, or until it is written to by the user (during a write operation).

The basic process is shown in Example 7-1.

7.4 Writing to the Data EEPROM Memory

To write an EEPROM data location, the address must first be written to the EEADR register and the data written to the EEDATA register. The sequence in Example 7-2 must be followed to initiate the write cycle.

The write will not begin if this sequence is not exactly followed (write 55h to EECON2, write 0AAh to EECON2, then set WR bit) for each byte. It is strongly recommended that interrupts be disabled during this code segment.

Additionally, the WREN bit in EECON1 must be set to enable writes. This mechanism prevents accidental writes to data EEPROM due to unexpected code execution (i.e., runaway programs). The WREN bit should be kept clear at all times, except when updating the EEPROM. The WREN bit is not cleared by hardware.

After a write sequence has been initiated, EECON1, EEADR and EEDATA cannot be modified. The WR bit will be inhibited from being set unless the WREN bit is set. Both WR and WREN cannot be set with the same instruction.

At the completion of the write cycle, the WR bit is cleared by hardware and the EEPROM Interrupt Flag bit, EEIF, is set. The user may either enable this interrupt or poll this bit. EEIF must be cleared by software.

7.5 Write Verify

Depending on the application, good programming practice may dictate that the value written to the memory should be verified against the original value. This should be used in applications where excessive writes can stress bits near the specification limit.

EXAMPLE 7-1: DATA EEPROM READ

MOVLW	DATA_EE_ADDR	;	
MOVWF	EEADR	;	Data Memory Address to read
BCF	EECON1, EEPGD	;	Point to DATA memory
BCF	EECON1, CFGS	;	Access EEPROM
BSF	EECON1, RD	;	EEPROM Read
MOVF	EEDATA, W	;	W = EEDATA

EXAMPLE 7-2:	DATA EEPROM WRITE

	MOVLW MOVWF	DATA_EE_ADDR_LOW EEADR	Data M	emory Address to write
	MOVLW	DATA_EE_ADDR_HI		
	MOVWF	EEADRH		
	MOVLW	DATA_EE_DATA		
	MOVWF	EEDATA	Data M	emory Value to write
	BCF	EECON1, EEPGD	Point	to DATA memory
	BCF	EECON1, CFGS	Access	EEPROM
	BSF	EECON1, WREN	Enable	writes
	BCF	INTCON, GIE	Disabl	e Interrupts
	MOVLW	55h		
Required	MOVWF	EECON2	Write	55h
Sequence	MOVLW	0AAh		
	MOVWF	EECON2	Write	0AAh
	BSF	EECON1, WR	Set WR	bit to begin write
	BSF	INTCON, GIE	Enable	Interrupts
			User c	ode execution
	BCF	EECON1, WREN	Disabl	e writes on write complete (EEIF set)

					<u> </u>			
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	
OSCFIE	C1IE	C2IE	EEIE	BCL1IE	HLVDIE	TMR3IE	CCP2IE	
bit 7							bit 0	
Legend:								
R = Readable	bit	W = Writable	bit	U = Unimpler	nented bit, rea	d as '0'		
-n = Value at I	POR	'1' = Bit is set		'0' = Bit is cle	ared	x = Bit is unk	nown	
bit 7	OSCFIE: Osc 1 = Enabled	cillator Fail Inte	rrupt Enable I	oit				
bit 6 C1IE: Comparator C1 Interrupt Enable bit 1 = Enabled 0 = Disabled								
bit 5	C2IE: Compa 1 = Enabled 0 = Disabled	arator C2 Intern	upt Enable bit	t				
bit 4	t 4 EEIE: Data EEPROM/Flash Write Operation Interrupt Enable bit 1 = Enabled 0 = Disabled							
bit 3	BCL1IE: MSSP1 Bus Collision Interrupt Enable bit 1 = Enabled 0 = Disabled							
bit 2	HLVDIE: Low 1 = Enabled 0 = Disabled	v-Voltage Detec	t Interrupt En	able bit				
bit 1	TMR3IE: TM 1 = Enabled 0 = Disabled	R3 Overflow In	terrupt Enable	e bit				
bit 0	CCP2IE: CCI 1 = Enabled 0 = Disabled	P2 Interrupt En	able bit					

REGISTER 9-10: PIE2: PERIPHERAL INTERRUPT ENABLE (FLAG) REGISTER 2

10.6 PORTE Registers

Depending on the particular PIC18(L)F2X/4XK22 device selected, PORTE is implemented in two different ways.

10.6.1 PORTE ON 40/44-PIN DEVICES

For PIC18(L)F2X/4XK22 devices, PORTE is a 4-bit wide port. Three pins (RE0/P3A/CCP3/AN5, RE1/P3B/ AN6 and RE2/CCP5/AN7) are individually configurable as inputs or outputs. These pins have Schmitt Trigger input buffers. When selected as an analog input, these pins will read as '0's.

The corresponding data direction register is TRISE. Setting a TRISE bit (= 1) will make the corresponding PORTE pin an input (i.e., disable the output driver). Clearing a TRISE bit (= 0) will make the corresponding PORTE pin an output (i.e., enable the output driver and put the contents of the output latch on the selected pin).

TRISE controls the direction of the REx pins, even when they are being used as analog inputs. The user must make sure to keep the pins configured as inputs when using them as analog inputs.

The Data Latch register (LATE) is also memory mapped. Read-modify-write operations on the LATE register read and write the latched output value for PORTE.

Note: On a Power-on Reset, RE<2:0> are configured as analog inputs.

The fourth pin of PORTE ($\overline{\text{MCLR}}/\text{VPP}/\text{RE3}$) is an input only pin. Its operation is controlled by the MCLRE Configuration bit. When selected as a port pin (MCLRE = 0), it functions as a digital input only pin; as such, it does not have TRIS or LAT bits associated with its operation. Otherwise, it functions as the device's Master Clear input. In either configuration, RE3 also functions as the programming voltage input during programming.

Note: On a Power-on Reset, RE3 is enabled as a digital input only if Master Clear functionality is disabled.

EXAMPLE 10-5: INITIALIZING PORTE

CLRF	PORTE	;	Initialize PORTE by
		;	clearing output
		;	data latches
CLRF	LATE	;	Alternate method
		;	to clear output
		;	data latches
CLRF	ANSELE	;	Configure analog pins
		;	for digital only
MOVLW	05h	;	Value used to
		;	initialize data
		;	direction
MOVWF	TRISE	;	Set RE<0> as input
		;	RE<1> as output
		;	RE<2> as input

10.6.2 PORTE ON 28-PIN DEVICES

For PIC18F2XK22 devices, PORTE is only available when Master Clear functionality is disabled (MCLR = 0). In these cases, PORTE is a single bit, input only port comprised of RE3 only. The pin operates as previously described.

10.6.3 RE3 WEAK PULL-UP

The port RE3 pin has an individually controlled weak internal pull-up. When set, the WPUE3 (TRISE<7>) bit enables the RE3 pin pull-up. The RBPU bit of the INT-CON2 register controls pull-ups on both PORTB and PORTE. When RBPU = 0, the weak pull-ups become active on all pins which have the WPUE3 or WPUBx bits set. When set, the RBPU bit disables all weak pull-ups. The pull-ups are disabled on a Power-on Reset. When the RE3 port pin is configured as MCLR, (CON-FIG3H<7>, MCLRE=1 and CONFIG4L<2>, LVP=0), or configured for Low Voltage Programming, (MCLRE=x and LVP=1), the pull-up is always enabled and the WPUE3 bit has no effect.

10.6.4 PORTE OUTPUT PRIORITY

Each PORTE pin is multiplexed with other functions. The pins, their combined functions and their output priorities are briefly described here. For additional information, refer to the appropriate section in this data sheet.

When multiple outputs are enabled, the actual pin control goes to the peripheral with the higher priority. Table 10-4 lists the PORTE pin functions from the highest to the lowest priority.

Analog input functions, such as ADC, comparator and SR latch inputs, are not shown in the priority lists.

These inputs are active when the I/O pin is set for Analog mode using the ANSELx registers. Digital output functions may control the pin when it is in Analog mode with the priority shown below.

13.1 Timer2/4/6 Operation

The clock input to the Timer2/4/6 module is the system instruction clock (Fosc/4).

TMRx increments from 00h on each clock edge.

A 4-bit counter/prescaler on the clock input allows direct input, divide-by-4 and divide-by-16 prescale options. These options are selected by the prescaler control bits, TxCKPS<1:0> of the TxCON register. The value of TMRx is compared to that of the Period register, PRx, on each clock cycle. When the two values match, the comparator generates a match signal as the timer output. This signal also resets the value of TMRx to 00h on the next cycle and drives the output counter/postscaler (see Section 13.2 "Timer2/4/6 Interrupt").

The TMRx and PRx registers are both directly readable and writable. The TMRx register is cleared on any device Reset, whereas the PRx register initializes to FFh. Both the prescaler and postscaler counters are cleared on the following events:

- a write to the TMRx register
- a write to the TxCON register
- Power-on Reset (POR)
- Brown-out Reset (BOR)
- MCLR Reset
- Watchdog Timer (WDT) Reset
- Stack Overflow Reset
- Stack Underflow Reset
- RESET Instruction

Note: TMRx is not cleared when TxCON is written.

13.2 Timer2/4/6 Interrupt

Timer2/4/6 can also generate an optional device interrupt. The Timer2/4/6 output signal (TMRx-to-PRx match) provides the input for the 4-bit counter/postscaler. This counter generates the TMRx match interrupt flag which is latched in TMRxIF of the PIR1/PIR5 registers. The interrupt is enabled by setting the TMRx Match Interrupt Enable bit, TMRxIE of the PIE1/PIE5 registers. Interrupt Priority is selected with the TMRxIP bit in the IPR1/IPR5 registers.

A range of 16 postscale options (from 1:1 through 1:16 inclusive) can be selected with the postscaler control bits, TxOUTPS<3:0>, of the TxCON register.

13.3 Timer2/4/6 Output

The unscaled output of TMRx is available primarily to the CCP modules, where it is used as a time base for operations in PWM mode. The timer to be used with a specific CCP module is selected using the CxTSEL<1:0> bits in the CCPTMRS0 and CCPTMRS1 registers.

Timer2 can be optionally used as the shift clock source for the MSSPx modules operating in SPI mode by setting SSPM<3:0> = 0011 in the SSPxCON1 register. Additional information is provided in Section 15.0 "Master Synchronous Serial Port (MSSP1 and MSSP2) Module".

13.4 Timer2/4/6 Operation During Sleep

The Timer2/4/6 timers cannot be operated while the processor is in Sleep mode. The contents of the TMRx and PRx registers will remain unchanged while the processor is in Sleep mode.

13.5 Peripheral Module Disable

When a peripheral module is not used or inactive, the module can be disabled by setting the Module Disable bit in the PMD registers. This will reduce power consumption to an absolute minimum. Setting the PMD bits holds the module in Reset and disconnects the module's clock source. The Module Disable bits for Timer2 (TMR2MD), Timer4 (TMR4MD) and Timer6 (TMR6MD) are in the PMD0 Register. See Section 3.0 "Power-Managed Modes" for more information.

PIC18(L)F2X/4XK22

14.3.6 PWM RESOLUTION

The resolution determines the number of available duty cycles for a given period. For example, a 10-bit resolution will result in 1024 discrete duty cycles, whereas an 8-bit resolution will result in 256 discrete duty cycles.

The maximum PWM resolution is ten bits when PRx is 255. The resolution is a function of the PRx register value as shown by Equation 14-4.

EQUATION 14-4: PWM RESOLUTION

Resolution =
$$\frac{\log[4(PRx+1)]}{\log(2)}$$
 bits

Note: If the pulse width value is greater than the period the assigned PWM pin(s) will remain unchanged.

TABLE 14-7: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS (Fosc = 32 MHz)

PWM Frequency	1.95 kHz	7.81 kHz	31.25 kHz	125 kHz	250 kHz	333.3 kHz
Timer Prescale (1, 4, 16)	16	4	1	1	1	1
PRx Value	0xFF	0xFF	0xFF	0x3F	0x1F	0x17
Maximum Resolution (bits)	10	10	10	8	7	6.6

TABLE 14-8: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS (Fosc = 20 MHz)

PWM Frequency	1.22 kHz	4.88 kHz	19.53 kHz	78.12 kHz	156.3 kHz	208.3 kHz
Timer Prescale (1, 4, 16)	16	4	1	1	1	1
PRx Value	0xFF	0xFF	0xFF	0x3F	0x1F	0x17
Maximum Resolution (bits)	10	10	10	8	7	6.6

TABLE 14-9: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS (Fosc = 8 MHz)

PWM Frequency	1.22 kHz	4.90 kHz	19.61 kHz	76.92 kHz	153.85 kHz	200.0 kHz
Timer Prescale (1, 4, 16)	16	4	1	1	1	1
PRx Value	0x65	0x65	0x65	0x19	0x0C	0x09
Maximum Resolution (bits)	8	8	8	6	5	5

14.3.7 OPERATION IN SLEEP MODE

In Sleep mode, the TMRx register will not increment and the state of the module will not change. If the CCPx pin is driving a value, it will continue to drive that value. When the device wakes up, TMRx will continue from its previous state.

14.3.8 CHANGES IN SYSTEM CLOCK FREQUENCY

The PWM frequency is derived from the system clock frequency. Any changes in the system clock frequency will result in changes to the PWM frequency. See Section 2.0 "Oscillator Module (With Fail-Safe Clock Monitor)" for additional details.

14.3.9 EFFECTS OF RESET

Any Reset will force all ports to Input mode and the CCP registers to their Reset states.

14.4.7 START-UP CONSIDERATIONS

When any PWM mode is used, the application hardware must use the proper external pull-up and/or pull-down resistors on the PWM output pins.

The CCPxM<1:0> bits of the CCPxCON register allow the user to choose whether the PWM output signals are active-high or active-low for each pair of PWM output pins (PxA/PxC and PxB/PxD). The PWM output polarities must be selected before the PWM pin output drivers are enabled. Changing the polarity configuration while the PWM pin output drivers are enable is not recommended since it may result in damage to the application circuits.

The PxA, PxB, PxC and PxD output latches may not be in the proper states when the PWM module is initialized. Enabling the PWM pin output drivers at the same time as the Enhanced PWM modes may cause damage to the application circuit. The Enhanced PWM modes must be enabled in the proper Output mode and complete a full PWM cycle before enabling the PWM pin output drivers. The completion of a full PWM cycle is indicated by the TMRxIF bit of the PIR1, PIR2 or PIR5 register being set as the second PWM period begins.

Note: When the microcontroller is released from Reset, all of the I/O pins are in the highimpedance state. The external circuits must keep the power switch devices in the Off state until the microcontroller drives the I/O pins with the proper signal levels or activates the PWM output(s).









15.2.3 SPI MASTER MODE

The master can initiate the data transfer at any time because it controls the SCKx line. The master determines when the slave (Processor 2, Figure 15-5) is to broadcast data by the software protocol.

In Master mode, the data is transmitted/received as soon as the SSPxBUF register is written to. If the SPI is only going to receive, the SDOx output could be disabled (programmed as an input). The SSPxSR register will continue to shift in the signal present on the SDIx pin at the programmed clock rate. As each byte is received, it will be loaded into the SSPxBUF register as if a normal received byte (interrupts and Status bits appropriately set).

The clock polarity is selected by appropriately programming the CKP bit of the SSPxCON1 register and the CKE bit of the SSPxSTAT register.

This then, would give waveforms for SPI communication as shown in Figure 15-6, Figure 15-8, Figure 15-9 and Figure 15-10, where the MSB is transmitted first. In Master mode, the SPI clock rate (bit rate) is user programmable to be one of the following:

- Fosc/4 (or Tcy)
- Fosc/16 (or 4 * Tcy)
- Fosc/64 (or 16 * Tcy)
- Timer2 output/2
- Fosc/(4 * (SSPxADD + 1))

Figure 15-6 shows the waveforms for Master mode.

When the CKE bit is set, the SDOx data is valid before there is a clock edge on SCKx. The change of the input sample is shown based on the state of the SMP bit. The time when the SSPxBUF is loaded with the received data is shown.





15.5.3 SLAVE TRANSMISSION

When the R/W bit of the incoming address byte is set and an address match occurs, the R/W bit of the SSPxSTAT register is set. The received address is loaded into the SSPxBUF register, and an ACK pulse is sent by the slave on the ninth bit.

Following the ACK, slave hardware clears the CKP bit and the SCLx pin is held low (see **Section 15.5.6 "Clock Stretching"** for more detail). By stretching the clock, the master will be unable to assert another clock pulse until the slave is done preparing the transmit data.

The transmit data must be loaded into the SSPxBUF register which also loads the SSPxSR register. Then the SCLx pin should be released by setting the CKP bit of the SSPxCON1 register. The eight data bits are shifted out on the falling edge of the SCLx input. This ensures that the SDAx signal is valid during the SCLx high time.

The ACK pulse from the master-receiver is latched on the rising edge of the ninth SCLx input pulse. This ACK value is copied to the ACKSTAT bit of the SSPxCON2 register. If ACKSTAT is set (not ACK), then the data transfer is complete. In this case, when the not ACK is latched by the slave, the slave goes Idle and waits for another occurrence of the Start bit. If the SDAx line was low (ACK), the next transmit data must be loaded into the SSPxBUF register. Again, the SCLx pin must be released by setting bit CKP.

An MSSPx interrupt is generated for each data transfer byte. The SSPxIF bit must be cleared by software and the SSPxSTAT register is used to determine the status of the byte. The SSPxIF bit is set on the falling edge of the ninth clock pulse.

15.5.3.1 Slave Mode Bus Collision

A slave receives a Read request and begins shifting data out on the SDAx line. If a bus collision is detected and the SBCDE bit of the SSPxCON3 register is set, the BCLxIF bit of the PIRx register is set. Once a bus collision is detected, the slave goes Idle and waits to be addressed again. User software can use the BCLxIF bit to handle a slave bus collision.

15.5.3.2 7-bit Transmission

A master device can transmit a read request to a slave, and then clock data out of the slave. The list below outlines what software for a slave will need to do to accomplish a standard transmission. Figure 15-18 can be used as a reference to this list.

- 1. Master sends a Start condition on SDAx and SCLx.
- 2. S bit of SSPxSTAT is set; SSPxIF is set if interrupt on Start detect is enabled.
- 3. Matching address with R/W bit set is received by the slave setting SSPxIF bit.
- 4. Slave hardware generates an ACK and sets SSPxIF.
- 5. SSPxIF bit is cleared by user.
- 6. Software reads the received address from SSPxBUF, clearing BF.
- 7. R/\overline{W} is set so CKP was automatically cleared after the ACK.
- 8. The slave software loads the transmit data into SSPxBUF.
- 9. CKP bit is set releasing SCLx, allowing the master to clock the data out of the slave.
- 10. SSPxIF is set after the ACK response from the master is loaded into the ACKSTAT register.
- 11. SSPxIF bit is cleared.
- 12. The slave software checks the ACKSTAT bit to see if the master wants to clock out more data.

Note 1: If the master ACKs the clock will be stretched.

 ACKSTAT is the only bit updated on the rising edge of SCLx (9th) rather than the falling.

- 13. Steps 9-13 are repeated for each transmitted byte.
- 14. If the master sends a not ACK; the clock is not held, but SSPxIF is still set.
- 15. The master sends a Restart condition or a Stop.
- 16. The slave is no longer addressed.

FIGURE 15-31: STOP CONDITION RECEIVE OR TRANSMIT MODE



15.6.10 SLEEP OPERATION

While in Sleep mode, the I²C slave module can receive addresses or data and when an address match or complete byte transfer occurs, wake the processor from Sleep (if the MSSPx interrupt is enabled).

15.6.11 EFFECTS OF A RESET

A Reset disables the MSSPx module and terminates the current transfer.

15.6.12 MULTI-MASTER MODE

In Multi-Master mode, the interrupt generation on the detection of the Start and Stop conditions allows the determination of when the bus is free. The Stop (P) and Start (S) bits are cleared from a Reset or when the MSSPx module is disabled. Control of the I²C bus may be taken when the P bit of the SSPxSTAT register is set, or the bus is Idle, with both the S and P bits clear. When the bus is busy, enabling the SSPx interrupt will generate the interrupt when the Stop condition occurs.

In multi-master operation, the SDAx line must be monitored for arbitration to see if the signal level is the expected output level. This check is performed by hardware with the result placed in the BCLxIF bit.

The states where arbitration can be lost are:

- Address Transfer
- Data Transfer
- A Start Condition
- A Repeated Start Condition
- An Acknowledge Condition

	-									
R/W-0	R-1	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0			
ABDOVF	RCIDL	DTRXP	CKTXP	BRG16		WUE	ABDEN			
bit 7							bit 0			
Legend:						<i>(</i> - .				
R = Readable b	Dit	W = Writable k	Dit	U = Unimplem	nented bit, read	as '0'				
-n = Value at P	OR	'1' = Bit is set		'0' = Bit is clea	ared	x = Bit is unkn	own			
bit 7	ABDOVF: Aut Asynchronous 1 = Auto-baud 0 = Auto-baud Synchronous Don't care	to-Baud Detect (<u>a mode</u> : I timer overflowe I timer did not ov <u>mode</u> :	Overflow bit ed verflow							
bit 6	RCIDL: Recei	ve Idle Flag bit								
	Asynchronous 1 = Receiver i 0 = Start bit ha Synchronous Don't care	<u>s mode</u> : s Idle as been detecte <u>mode</u> :	d and the rece	eiver is active						
bit 5	DTRXP: Data/	DTRXP: Data/Receive Polarity Select bit								
	Asynchronous mode:									
	1 = Receive data (RXx) is inverted (active-low)									
	u = Receive data (RAX) is not inverted (active-nign) Svnchronous mode:									
	1 = Data (DTx) is inverted (active-low)									
	0 = Data (DTx) is not inverted	(active-high)							
bit 4	CKTXP: Clock	√Transmit Polar	ity Select bit							
	<u>Asynchronous mode</u> : 1 = Idle state for transmit (TXx) is low 0 = Idle state for transmit (TXx) is high									
	Synchronous mode:									
	1 = Data chan 0 = Data chan	iges on the fallin iges on the risin	ng edge of the g edge of the	clock and is sar clock and is san	npled on the ris npled on the fal	ing edge of the c ling edge of the c	lock lock			
bit 3	BRG16: 16-bi 1 = 16-bit Ba 0 = 8-bit Bau	t Baud Rate Ge ud Rate Genera d Rate Generate	nerator bit itor is used (S or is used (SP	PBRGHx:SPBR BRGx)	Gx)					
bit 2	Unimplement	ted: Read as '0'								
bit 1	WUE: Wake-u	ip Enable bit								
	Asynchronous 1 = Receiver i edge. WL 0 = Receiver i Synchronous	mode: s waiting for a f JE will automation s operating norr mode:	alling edge. N cally clear on t nally	o character will he rising edge.	be received bu	t RCxIF will be se	et on the falling			
	Don't care									
bit 0	ABDEN: Auto	-Baud Detect Er	nable bit							
	Asynchronous 1 = Auto-Bau 0 = Auto-Bau Synchronous Don't care	<u>a mode</u> : Id Detect mode Id Detect mode <u>mode</u> :	is enabled (cle is disabled	ears when auto-l	baud is comple	te)				

REGISTER 16-3: BAUDCONX: BAUD RATE CONTROL REGISTER

18.9 Register Definitions: Comparator Control

REGISTER 18-1: CMxCON0: COMPARATOR x CONTROL REGISTER

R/W-0	R-0	R/W-0	R/W-0	R/W-1	R/W-0	R/W-0	R/W-0	
CxON	CxOUT	CxOE	CxPOL	CxSP	CxR	CxCH	<1:0>	
bit 7							bit 0	
								
Legend:								
R = Readable	e bit	W = Writable	bit	U = Unimpler	mented bit, rea	ad as '0'		
-n = Value at	POR	'1' = Bit is set		'0' = Bit is cle	ared	x = Bit is unkr	nown	
bit 7	CxON: Comp 1 = Compara 0 = Compara	arator Cx Enal tor Cx is enable tor Cx is disabl	ole bit ed ed					
bit 6	bit 6 CxOUT: Comparator Cx Output bit $\frac{ f CxPOL = 1 (inverted polarity):}{CxOUT = 0 when CxVIN+ > CxVIN- CxOUT = 1 when CxVIN+ < CxVIN- \frac{ f CxPOL = 0 (non-inverted polarity):}{CxOUT = 1 when CxVIN+ > CxVIN- CxOUT = 1 when CxVIN+ > CxVIN- CxOUT = 1 when CxVIN+ > CxVIN-$							
bit 5	CxOE: Comp 1 = CxOUT is 0 = CxOUT is	arator Cx Outp present on the internal only	out Enable bit e CxOUT pin ⁽¹)				
bit 4	CxPOL: Comparator Cx Output Polarity Select bit 1 = CxOUT logic is inverted 0 = CxOUT logic is not inverted							
bit 3	CxSP: Comp 1 = Cx operat 0 = Cx operat	arator Cx Spee tes in Normal-F tes in Low-Pow	d/Power Sele Power, Higher ver, Low-Spee	ct bit Speed mode d mode				
bit 2	CxR: Compare 1 = CxVIN+ co 0 = CxVIN+ co	rator Cx Refere onnects to CXV onnects to C12	ence Select bit REF output IN+ pin	t (non-inverting	input)			
bit 1-0	CxCH<1:0>: 00 = C12IN0- 01 = C12IN1- 10 = C12IN2- 11 = C12IN3-	Comparator C: pin of Cx conr pin of Cx conr pin of Cx conr pin of Cx conr pin of Cx conr	Channel Sel nects to CxVIN nects to CxVIN nects to CxVIN nects to CxVIN	ect bit - - -				

Note 1: Comparator output requires the following three conditions: CxOE = 1, CxON = 1 and corresponding port TRIS bit = 0.

EXAMPLE 19-3: CAPACITANCE CALIBRATION ROUTINE

```
#include "p18cxxx.h"
#define COUNT 25
                                            //@ 8MHz INTFRC = 62.5 us.
#define ETIME COUNT*2.5
                                            //time in uS
#define DELAY for(i=0;i<COUNT;i++)</pre>
#define ADSCALE 1023
                                            //for unsigned conversion 10 sig
bits
#define ADREF 3.3
                                            //Vdd connected to A/D Vr+
#define RCAL .027
                                            //R value is 4200000 (4.2M)
                                            //scaled so that result is in
                                            //1/100th of uA
int main(void)
{
   int i;
   int j = 0;
                                            //index for loop
   unsigned int Vread = 0;
   float CTMUISrc, CTMUCap, Vavg, VTot, Vcal;
//assume CTMU and A/D have been set up correctly
//see Example 25-1 for CTMU & A/D setup
setup();
CTMUCONHbits.CTMUEN = 1;
                                            //Enable the CTMU
CTMUCONLbits.EDG1STAT = 0;
                                            // Set Edge status bits to zero
CTMUCONLbits.EDG2STAT = 0;
   for(j=0;j<10;j++)</pre>
    {
       CTMUCONHbits.IDISSEN = 1;
                                           //drain charge on the circuit
                                            //wait 125us
       DELAY;
       CTMUCONHbits.IDISSEN = 0;
                                            //end drain of circuit
       CTMUCONLbits.EDG1STAT = 1;
                                            //Begin charging the circuit
                                            //using CTMU current source
                                            //wait for 125us
       DELAY;
       CTMUCONLbits.EDG1STAT = 0;
                                           //Stop charging circuit
       PIR1bits.ADIF = 0;
                                           //make sure A/D Int not set
       ADCON0bits.GO=1;
                                            //and begin A/D conv.
       while(!PIR1bits.ADIF);
                                            //Wait for A/D convert complete
       Vread = ADRES;
                                            //Get the value from the A/D
       PIR1bits.ADIF = 0;
                                            //Clear A/D Interrupt Flag
       VTot += Vread;
                                            //Add the reading to the total
   }
   Vavg = (float)(VTot/10.000);
                                            //Average of 10 readings
   Vcal = (float)(Vavg/ADSCALE*ADREF);
                                            //CTMUISrc is in 1/100ths of uA
   CTMUISrc = Vcal/RCAL;
   CTMUCap = (CTMUISrc*ETIME/Vcal)/100;
```

PIC18(L)F2X/4XK22

SUBWFB	Subtract W from f with Borrow						
Syntax:	SI	JBWFB	f {,d {,a	n}}			
Operands:	0 : d a	≤ f ≤ 255 ∈ [0,1] ∈ [0,1]					
Operation:	(f)	– (W) –	$(\overline{C}) \rightarrow de$	st			
Status Affected:	N,	OV, C, E	DC, Z				
Encoding:		0101	10da	fff	f ffff		
Description:	iption: Subtract W and the CARRY flag (borrow) from register 'f' (2's comple- ment method). If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored back in register 'f' (default). If 'a' is '0', the Access Bank is selected If 'a' is '1', the BSR is used to select the GPR bank. If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever f ≤ 95 (5Fh). See Section 25.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed						
Words:	1						
Cvcles:	1						
Q Cycle Activity:							
Q1		Q2	Q	3	Q4		
Decode		Read	Proce	ess	Write to		
	re	gister 'f'	Dat	a	destination		
Example 1:	5	SUBWFB	REG, 1	, 0			
Before Instruc REG W C	tion = = =	19h 0Dh 1	(000)	1 100 0 110	01) 01)		
After Instructio REG W C Z	n = = =	0Ch 0Dh 1 0	(000)	0 110 0 110	00) 01)		
Ν	=	0	; resu	lt is po	sitive		
Example 2:	S	SUBWFB	REG, 0	, 0			
Before Instruc REG W C	tion = = =	1Bh 1Ah 0	(000)	1 101 1 101	.1) .0)		
After Instructic REG W C	n = =	1Bh 00h 1	(000)	1 101	1)		
Ž N	= =	1 0	; resu	lt is ze	ro		
Example 3:	5	SUBWFB	REG, 1	, 0			
Before Instruc REG W C	tion = = =	03h 0Eh 1	(000)	0 001 0 111	.1) .0)		
After Instructio REG W	n = =	F5h 0Eh	(111); ; [2's ((000)	1 010 comp] 0 111	01)		
C Z N	= = =	0 0 1	; resu	lt is ne	egative		

SWAPF	Swap f							
Syntax:	SWAPF f {,d {,a}}							
Operands:	0 ≤ f ≤ 255 d ∈ [0,1] a ∈ [0,1]	0 ≤ f ≤ 255 d ∈ [0,1] a ∈ [0,1]						
Operation:	(f<3:0>) → (f<7:4>) →	→ dest<7:4 → dest<3:0	l>,)>					
Status Affected:	None							
Encoding:	0011	10da	ffff	ffff				
	The upper and lower nubbles of register 'f' are exchanged. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed in register 'f' (default). If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank. If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \le 95$ (5Fh). See Section 25.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Addressing							
Words:	1							
Cycles:	1							
Q Cycle Activity:								
Q1	Q2	Q3	3	Q4				

Decode	Read	Process	Write to
	register 'f'	Data	destination

REG, 1, 0

Example:

SWAPF

Before Instruction REG = 53h After Instruction REG = 35h







© 2010-2016 Microchip Technology Inc.

28-Lead Plastic Shrink Small Outline (SS) - 5.30 mm Body [SSOP]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



	MILLIMETERS			
Dimension	MIN	NOM	MAX	
Contact Pitch E 0.65 BSC				
Contact Pad Spacing	С		7.20	
Contact Pad Width (X28)	X1			0.45
Contact Pad Length (X28)	Y1			1.75
Distance Between Pads	G	0.20		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2073A