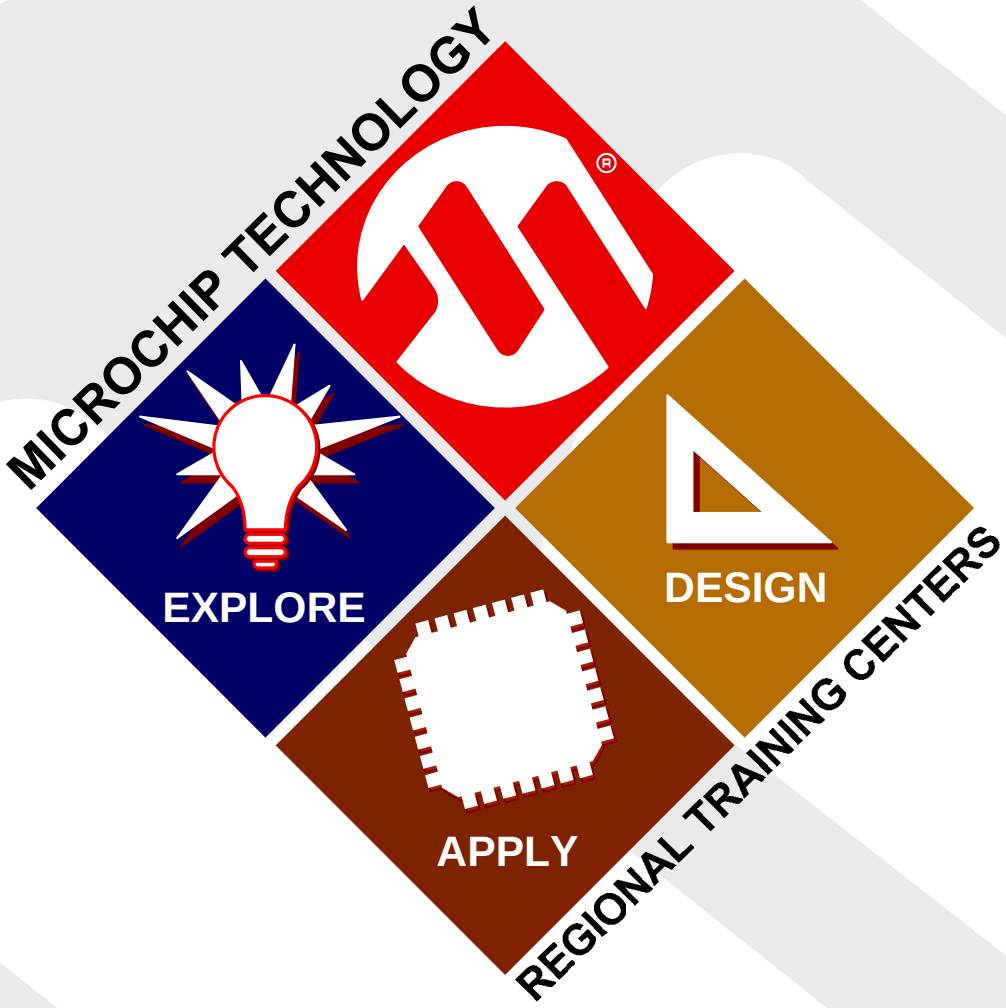


202 PRC

PIC18 Peripheral Configuration and MPLAB® C18 Compiler Programming Techniques

Hands-On Exercises

Table of Contents

202 PRC

PIC18 Peripheral Configuration and MPLAB® C18 Compiler Programming Techniques

202 PRC

PIC18 Peripheral Configuration and MPLAB® C18 Programming Techniques

Lab 1 - I/O and Interrupts

Description:

This function configures RB0 as external interrupt pin. Initially RB0 is in high state (when NOT pressed). Interrupt is enabled for external (RB0) interrupt. Interrupt occurs at the falling edge on RB0. Increments the counter on every time the key is pressed. The counter value is displayed on the LCD. Also in interrupt bouncing of key is avoided (De bounce)

Project settings:

Configuration word:

Select appropriate clock (HS in the current example)
Configure PORTB<4:0> as digital IO on reset
Disable watch dog timer (enable if required)
If using ICD, Disable WDT, Low Voltage programming

Language tool suite:

C18 if writing in C
ASM for assembly programming.

Hardware Configuration:

Ensure jumper J6 is open (RB0 will not work if jumper is connected)

NOTE: All delays in this project are calculated based on 10.0000MHz crystal

HINT: C Syntax for Setting / Clearing an Individual Bit

```
<RegisterName>bits.<BitName> = <0 or 1>;
INTCONbits.GIE = 1;
```

HINT: C Syntax for Writing a Value to an Entire Register

```
<RegisterName> = <value>;
INTCON = 0x90;
```

TRISB REGISTER (PORTB Data Direction Register)

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0
bit 7							bit 0

bit 7-0 **TRISB7—TRISB0** PORTB Data Direction Register

0 = Port pin is an output
1 = Port pin is an input

INTCON REGISTER—Interrupt Control Register

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF
bit 7							bit 0

bit 7 **GIE/GIEH:** Global Interrupt Enable bit
When IPEN = 0:
1 = Enables all unmasked interrupts
0 = Disables all interrupts
When IPEN = 1:
1 = Enables all high priority interrupts
0 = Disables all interrupts

bit 6 **PEIE/GIEL:** Peripheral Interrupt Enable bit
When IPEN = 0:
1 = Enables all unmasked peripheral interrupts
0 = Disables all peripheral interrupts
When IPEN = 1:
1 = Enables all low priority peripheral interrupts
0 = Disables all low priority peripheral interrupts

bit 5 **TMR0IE:** TMR0 Overflow Interrupt Enable bit
1 = Enables the TMR0 overflow interrupt
0 = Disables the TMR0 overflow interrupt

bit 4 **INT0IE:** INT0 External Interrupt Enable bit
1 = Enables the INT0 external interrupt
0 = Disables the INT0 external interrupt

bit 3 **RBIE:** RB Port Change Interrupt Enable bit
1 = Enables the RB port change interrupt
0 = Disables the RB port change interrupt

bit 2 **TMR0IF:** TMR0 Overflow Interrupt Flag bit
1 = TMR0 register has overflowed (must be cleared in software)
0 = TMR0 register did not overflow

bit 1 **INT0IF:** INT0 External Interrupt Flag bit
1 = The INT0 external interrupt occurred (must be cleared in software)
0 = The INT0 external interrupt did not occur

bit 0 **RBIF:** RB Port Change Interrupt Flag bit
1 = At least one of the RB7:RB4 pins changed state (must be cleared in software)
0 = None of the RB7:RB4 pins have changed state

Note: A mismatch condition will continue to set this bit. Reading PORTB will end the mismatch condition and allow the bit to be cleared.

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

NOTE: Only highlighted bits are applicable to this exercise.

Lab 1

INTCON2 REGISTER—Interrupt Control Register 2

R/W-1	R/W-1	R/W-1	R/W-1	U-0	R/W-1	U-0	R/W-1
$\overline{\text{RBPU}}$	INTEDG0	INTEDG1	INTEDG2	-	TMR0IP	-	RBIP
bit 7							bit 0

bit 7 **RBPU:** PORTB Pull-up Enable bit
1 = All PORTB pull-ups are disabled
0 = PORTB pull-ups are enabled by individual port latch values

bit 6 **INTEDG0:** External Interrupt 0 Edge Select bit
1 = Interrupt on rising edge
0 = Interrupt on falling edge

bit 5 **INTEDG1:** External Interrupt 1 Edge Select bit
1 = Interrupt on rising edge
0 = Interrupt on falling edge

bit 4 **INTEDG2:** External Interrupt 2 Edge Select bit
1 = Interrupt on rising edge
0 = Interrupt on falling edge

bit 3 **Unimplemented:** Read as ‘0’

bit 2 **TMR0IP:** TMR0 Overflow Interrupt Priority bit
1 = High priority
0 = Low priority

bit 1 **Unimplemented:** Read as ‘0’

bit 0 **RBIP:** RB Port Change Interrupt Priority bit
1 = High priority
0 = Low priority

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

Note: Interrupt flag bits are set when an interrupt condition occurs, regardless of the state of its corresponding enable bit or the global enable bit. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt. This feature allows for software polling.

NOTE: Only highlighted bits are applicable to this exercise.

Lab 2 - A/D Converter

Description:

This code measures voltage from a POT provided on PIC DEM 2 PLUS board
ADC of PIC18F4520 is used to sample voltage tapped from POT.
Measured voltage is displayed on LCD (0 to 5V)

The center tap of R16 is connected to RA0; the result after ADC is multiplied by a scaling factor and the multiplied result is formatted and displayed on LCD.

Project settings:

Configuration word:

Select appropriate clock (HS in the current example)
Disable watch dog timer (enable if required)
If using ICD, Disable WDT, Low Voltage programming

Language tool suite:

C18 if writing in C
ASM for assembly programming.

Hardware Configuration:

N/A

NOTE: All delays in this project are calculated based on 10.0000MHz crystal

ADCON0 REGISTER—A/D Control Register 0

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
-	-	CHS3	CHS2	CHS1	CHS1	GO/$\overline{\text{DONE}}$	ADON
bit 7							bit 0

bit 7-6 **Unimplemented:** Read as '0'

bit 5-2 **CHS3:CHS0:** Analog Channel Select bits

0000 = Channel 0 (AN0) (The potentiometer on the PICDEM 2 Plus is connected to this one)
0001 = Channel 1 (AN1)
0010 = Channel 2 (AN2)
0011 = Channel 3 (AN3)
0100 = Channel 4 (AN4)
0101 = Channel 5 (AN5)**(1,2)**
0110 = Channel 6 (AN6)**(1,2)**
0111 = Channel 7 (AN7)**(1,2)**
1000 = Channel 8 (AN8)
1001 = Channel 9 (AN9)
1010 = Channel 10 (AN10)
1011 = Channel 11 (AN11)
1100 = Channel 12 (AN12
1101 = Unimplemented**(2)**
1110 = Unimplemented**(2)**
1111 = Unimplemented**(2)**

Note 1: These channels are not implemented on 28-pin devices.
2: Performing a conversion on unimplemented channels will return a floating input measurement.

bit 1 **GO/$\overline{\text{DONE}}$:** A/D Conversion Status bit

When ADON = 1:
1 = A/D conversion in progress
0 = A/D Idle

NOTE: The GO/DONE bit is not set during initialization. It is only set when configuration is complete and you are ready to begin a conversion. Before setting this bit, an acquisition delay is required (it is already provided in the lab code). See the data sheet for details.

bit 0 **ADON:** A/D On bit

1 = A/D converter module is enabled (Turn on the ADC module)
0 = A/D converter module is disabled

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

Lab 2

ADCON1 REGISTER—A/D Control Register 1

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
-	-	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0
bit 7							bit 0

bit 7-6 **Unimplemented:** Read as '0'

bit 5 **VCFG1:** Voltage Reference Configuration bit (V_{REF-} source)
1 = V_{REF-} (AN2)
0 = V_{SS} (Use ground as negative reference)

bit 4 **VCFG0:** Voltage Reference Configuration bit (V_{REF+} source)
1 = V_{REF+} (AN3)
0 = V_{DD} (Use supply as positive reference)

bit 3-0 **PCFG3:PCFG0:** A/D Port Configuration Control bits:

PCFG3: PCFG0	AN12	AN11	AN10	AN9	AN8	AN7	AN6	AN5	AN4	AN3	AN2	AN1	AN0
0000	A	A	A	A	A	A	A	A	A	A	A	A	A
0001	A	A	A	A	A	A	A	A	A	A	A	A	A
0010	A	A	A	A	A	A	A	A	A	A	A	A	A
0011	D	A	A	A	A	A	A	A	A	A	A	A	A
0100	D	D	A	A	A	A	A	A	A	A	A	A	A
0101	D	D	D	A	A	A	A	A	A	A	A	A	A
0110	D	D	D	D	A	A	A	A	A	A	A	A	A
0111	D	D	D	D	D	A	A	A	A	A	A	A	A
1000	D	D	D	D	D	D	A	A	A	A	A	A	A
1001	D	D	D	D	D	D	D	A	A	A	A	A	A
1010	D	D	D	D	D	D	D	D	A	A	A	A	A
1011	D	D	D	D	D	D	D	D	D	A	A	A	A
1100	D	D	D	D	D	D	D	D	D	D	A	A	A
1101	D	D	D	D	D	D	D	D	D	D	D	A	A
1110	D	D	D	D	D	D	D	D	D	D	D	D	A
1111	D	D	D	D	D	D	D	D	D	D	D	D	D

A = Analog Input **D = Digital I/O**

(Only AN0 needs to be an analog input for the potentiometer, all other pins should be digital I/O.)

Note 1: The POR value of the PCFG bits depends on the value of the PBADEN configuration bit. When PBADEN = 1, PCFG<3:0> = 0000; when PBADEN = 0, PCFG<3:0> = 0111.

2: AN5 through AN7 are available only on 40/44-pin devices.

ADCON2 REGISTER—A/D Control Register 2

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	-	ACQT2	ACQT1	ACQT0	ADCS2	ADCS1	ADCS0
bit 7							bit 0

bit 7 **ADFM:** A/D Result Format Select bit
1 = Right justified
0 = Left justified (We won't be using the two LSbs in ADRESL)

bit 6 **Unimplemented:** Read as '0'

bit 5-3 **ACQT2:ACQT0:** A/D Acquisition Time Select bits
111 = 20 TAD
110 = 16 TAD
101 = 12 TAD (Based on formulas in data sheet)
100 = 8 TAD
011 = 6 TAD
010 = 4 TAD
001 = 2 TAD
000 = 0 TAD**(1)**

bit 2-0 **ADCS2:ADCS0:** A/D Conversion Clock Select bits
111 = FRC (clock derived from A/D RC oscillator)**(1)**
110 = FOSC/64
101 = FOSC/16
100 = FOSC/4
011 = FRC (clock derived from A/D RC oscillator)**(1)**
010 = FOSC/32
001 = FOSC/8 (Based on formulas in data sheet)
000 = FOSC/2

Note 1: If the A/D FRC clock source is selected, a delay of one TCY (instruction cycle) is added before the A/D clock starts. This allows the SLEEP instruction to be executed before starting a conversion.

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

Lab 3 - Real Time Clock with TMR1

Description:

A Real time clock (RTC) is designed using Timer1 (Explained in RTC.C), time is displayed on 16x1 LCD provided on PICDEM2 board.

A watch Crystal of frequency 32.768 KHz is connected to T1OSO & T1OSI pins.Timer1 increments at the rate of approx 30.51 micro seconds. Timer1 is 16 bit wide; hence it takes 2 seconds for timer to overflow from zero. To get 1 second resolution for RTC application Timer1 counter registers TMR1H:TMR1L are loaded with half of the value i.e., 0x8000. e.g. `TMR1 = 0x8000;`

Interrupt is enabled for timer1 overflow conditions, three counters Hours, Minutes and Seconds are incremented at the interrupt and time elapsed is displayed on LCD.

Using Switches S2 and S3 Time can be set and allow running from set value.

Time on LCD is refreshed for every second.

Timer 1 is configured to run on every rising edge of clock i.e., synchronous mode.

Project settings:

Configuration word:

Select appropriate clock (HS in the current example)
Disable watch dog timer (enable if required)
If using ICD, Disable WDT, Low Voltage programming

Language tool suite:

C18 if writing in C
ASM for assembly programming.

Hardware Configuration:

N/A

NOTE: All delays in this project are calculated based on 10.0000MHz crystal

INTCON REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF
bit 7							bit 0

See page 5 of 202 PRC handout for details. Bits 7 & 6 will need to be set for TMR1 interrupts to work.

T1CON REGISTER—Timer 1 Control Register

R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
RD16	T1RUN	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	TMR1CS	TMR1ON
bit 7							bit 0

bit 7 **RD16:** 16-bit Read/Write Mode Enable bit
1 = Enables register read/write of Timer1 in one 16-bit operation
0 = Enables register read/write of Timer1 in two 8-bit operations

bit 6 **T1RUN:** Timer1 System Clock Status bit
1 = Device clock is derived from Timer1 oscillator
0 = Device clock is derived from another source (This is the default value)

bit 5-4 **T1CKPS1:T1CKPS0:** Timer1 Input Clock Prescale Select bits
11 = 1:8 Prescale value
10 = 1:4 Prescale value
01 = 1:2 Prescale value
00 = 1:1 Prescale value

bit 3 **T1OSCEN:** Timer1 Oscillator Enable bit
1 = Timer1 oscillator is enabled
0 = Timer1 oscillator is shut off
The oscillator inverter and feedback resistor are turned off to eliminate power drain.

bit 2 **T1SYNC:** Timer1 External Clock Input Synchronization Select bit
When TMR1CS = 1:
1 = Do not synchronize external clock input
0 = Synchronize external clock input
When TMR1CS = 0:
This bit is ignored. Timer1 uses the internal clock when TMR1CS = 0.

bit 1 **TMR1CS:** Timer1 Clock Source Select bit
1 = External clock from pin RC0/T1OSO/T13CKI (on the rising edge)
0 = Internal clock (Fosc/4)

bit 0 **TMR1ON:** Timer1 On bit
1 = Enables Timer1
0 = Stops Timer1

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

NOTE: In Synchronous mode Timer1 counter increments on every rising edge of the clock pulse. Timer1 will not increment in SLEEP mode since the synchronization circuit is shut off. The Prescaler will however continue to increment

Lab 3

PIE1 REGISTER—Peripheral Interrupt Enable Register 1

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PSPIE(1)	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE
bit 7							bit 0

bit 7 **PSPIE:** Parallel Slave Port Read/Write Interrupt Enable bit**(1)**
1 = Enables the PSP read/write interrupt
0 = Disables the PSP read/write interrupt
Note 1: This bit is unimplemented on 28-pin devices and will read as '0'.

bit 6 **ADIE:** A/D Converter Interrupt Enable bit
1 = Enables the A/D interrupt
0 = Disables the A/D interrupt

bit 5 **RCIE:** EUSART Receive Interrupt Enable bit
1 = Enables the EUSART receive interrupt
0 = Disables the EUSART receive interrupt

bit 4 **TXIE:** EUSART Transmit Interrupt Enable bit
1 = Enables the EUSART transmit interrupt
0 = Disables the EUSART transmit interrupt

bit 3 **SSPIE:** Master Synchronous Serial Port Interrupt Enable bit
1 = Enables the MSSP interrupt
0 = Disables the MSSP interrupt

bit 2 **CCP1IE:** CCP1 Interrupt Enable bit
1 = Enables the CCP1 interrupt
0 = Disables the CCP1 interrupt

bit 1 **TMR2IE:** TMR2 to PR2 Match Interrupt Enable bit
1 = Enables the TMR2 to PR2 match interrupt
0 = Disables the TMR2 to PR2 match interrupt

bit 0 **TMR1IE:** TMR1 Overflow Interrupt Enable bit
1 = Enables the TMR1 overflow interrupt
0 = Disables the TMR1 overflow interrupt

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

PIR1 REGISTER—Peripheral Interrupt Request Register 1 (Flags)

R/W-0	R/W-0	R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
PSPIF(1)	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF
bit 7							bit 0

bit 7 **PSPIF:** Parallel Slave Port Read/Write Interrupt Flag bit**(1)**
1 = A read or a write operation has taken place (must be cleared in software)
0 = No read or write has occurred
Note 1: This bit is unimplemented on 28-pin devices and will read as '0'.

bit 6 **ADIF:** A/D Converter Interrupt Flag bit
1 = An A/D conversion completed (must be cleared in software)
0 = The A/D conversion is not complete

bit 5 **RCIF:** EUSART Receive Interrupt Flag bit
1 = The EUSART receive buffer, RCREG, is full (cleared when RCREG is read)
0 = The EUSART receive buffer is empty

bit 4 **TXIF:** EUSART Transmit Interrupt Flag bit
1 = The EUSART transmit buffer, TXREG, is empty (cleared when TXREG is written)
0 = The EUSART transmit buffer is full

bit 3 **SSPIF:** Master Synchronous Serial Port Interrupt Flag bit
1 = The transmission/reception is complete (must be cleared in software)
0 = Waiting to transmit/receive

bit 2 **CCP1IF:** CCP1 Interrupt Flag bit
Capture mode:
1 = A TMR1 register capture occurred (must be cleared in software)
0 = No TMR1 register capture occurred
Compare mode:
1 = A TMR1 register compare match occurred (must be cleared in software)
0 = No TMR1 register compare match occurred
PWM mode:
Unused in this mode.

bit 1 **TMR2IF:** TMR2 to PR2 Match Interrupt Flag bit
1 = TMR2 to PR2 match occurred (must be cleared in software)
0 = No TMR2 to PR2 match occurred

bit 0 **TMR1IF:** TMR1 Overflow Interrupt Flag bit (Read this bit to see if TMR1 caused the interrupt)
1 = TMR1 register overflowed (must be cleared in software)
0 = TMR1 register did not overflow

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

Lab 4 - PWM

Description:

CCP module is configured to PWM mode. A/D module is used to generate ADC value.
ADC value depends on the POT (R16 – RA0) position. Each time ADC is performed and ADRESH is loaded to Period control register PR2 and half of ADRESH is loaded to Duty Cycle control register CCPR1L which gives 50% duty cycle.

Configured Duty Cycle and Period is displayed on LCD. An oscilloscope can be used to view the PWM signal

Project settings:

Configuration word:

Select appropriate clock (HS in the current example)
Disable watch dog timer (enable if required)
If using ICD, Disable WDT, Low Voltage programming

Language tool suite:

C18 if writing in C
ASM for assembly programming.

Hardware Configuration:

NOTE: All delays in this project are calculated based on 10.0000MHz crystal

Useful Formulas From Data Sheet

PWM Period = [(PR2) + 1] • 4 • T_{OSC} • (TMR2 Prescale Value)

PWM Duty Cycle = (CCPR1L:CCP1CON<5:4>) • T_{OSC} • (TMR2 Prescale Value)

Where:

T_{OSC} = System oscillator period = 1/F_{OSC} = 1/10MHz = 100ns
PR2 = Timer 2 Period Register
TMR2 Prescale Value = Option selected in T2CON register
CCPR1L:CCP1CON<5:4> = Duty Cycle bits (10-bits in two registers: CCPR1L and CCP1CON)

Example:

1) **PWM Period** = [(PR2) + 1] • 4 • TOSC • (TMR2 Pre scale Value)
For PR2 = 0x80, CCPR1L: CCP1CON<5:4> = 380, TOSC = 1/10.0000 MHz, TMR2 presale = 4.
PWM period = 206.4 micro Seconds
PWM Frequency = 1/ PWM Period = 4.902 KHz.

2) **PWM Duty Cycle** = (CCPRXL: CCPXCON<5:4>) • TOSC • (TMR2 Presale Value)
Considering same value for CCPR1L:CCP1CON<5:4> = 380,
PWM Duty Cycle = 152.0 micro Seconds

CCP1CON REGISTER—Enhanced CCP 1 Control Register

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
P1M1	P1M0	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0
bit 7							bit 0

bit 7-6 **P1M1:P1M0:** Enhanced PWM Output Configuration bits

If CCP1M3:CCP1M2 = 00, 01, 10:

xx = P1A assigned as Capture/Compare input/output; P1B, P1C, P1D assigned as port pins

If CCP1M3:CCP1M2 = 11:

00 = Single output: P1A modulated; P1B, P1C, P1D assigned as port pins

01 = Full-bridge output forward: P1D modulated; P1A active; P1B, P1C inactive

10 = Half-bridge output: P1A, P1B modulated with dead-band control; P1C, P1D assigned as port pins

11 = Full-bridge output reverse: P1B modulated; P1C active; P1A, P1D inactive

bit 5-4 **DC1B1:DC1B0**: PWM Duty Cycle bit 1 and bit 0

Capture mode:

Unused.

Compare mode:

Unused.

PWM mode:

These bits are the two LSbs of the 10-bit PWM duty cycle. The eight MSbs of the duty cycle are found in CCPR1L.

bit 3-0 **CCP1M3:CCP1M0**: Enhanced CCP Mode Select bits

0000 = Capture/Compare/PWM off (resets ECCP module)

0001 = Reserved

0010 = Compare mode, toggle output on match

0011 = Capture mode

0100 = Capture mode, every falling edge

0101 = Capture mode, every rising edge

0110 = Capture mode, every 4th rising edge

0111 = Capture mode, every 16th rising edge

1000 = Compare mode, initialize CCP1 pin low, set output on compare match (set CCP1IF)

1001 = Compare mode, initialize CCP1 pin high, clear output on compare match (set CCP1IF)

1010 = Compare mode, generate software interrupt only, CCP1 pin reverts to I/O state

1011 = Compare mode, trigger special event (ECCP resets TMR1 or TMR3, sets CC1IF bit)

1100 = PWM mode; P1A, P1C active-high; P1B, P1D active-high

1101 = PWM mode; P1A, P1C active-high; P1B, P1D active-low

1110 = PWM mode; P1A, P1C active-low; P1B, P1D active-high

1111 = PWM mode; P1A, P1C active-low; P1B, P1D active-low

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

NOTE FOR EXERCISE: Any combination of 11xx may be chosen for bits 3-0 since only P1A will be modulated (others not connected to pins) because bits 7-6 are both clear (see above).

Lab 4

T2CON REGISTER—Timer 2 Control Register 1

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
-	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0
bit 7							bit 0

bit 7 **Unimplemented:** Read as '0'

bit 6-3 **T2OUTPS3:T2OUTPS0**: Timer2 Output Postscale Select bits

0000 = 1:1 Postscale (Postscaler not used by CCP1 in PWM mode)

0001 = 1:2 Postscale

•

•

•

1111 = 1:16 Postscale

bit 2 **TMR2ON**: Timer2 On bit

1 = Timer2 is on (Turn on Timer 2)

0 = Timer2 is off

bit 1-0 **T2CKPS1:T2CKPS0**: Timer2 Clock Prescale Select bits

00 = Prescaler is 1

01 = Prescaler is 4 (Chosen based on formulas in data sheet)

1x = Prescaler is 16

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

Lab 5 - Input Capture

Description:

PWM module is used to generate signal, A/D module is used to load Period control register (PR2) and Duty Cycle control register (CCPR1L). Duty Cycle is always maintained at 50% of period.

CCP module is configured in capture mode to capture at rising edge.
Two consecutive rising edge will give Pulse period and events at rising and immediate falling edge will provide pulse width.

For demonstration, PWM pulse length and CCP capture event are configured at 1:1 ratio, more details on how this is achieved is mentioned in details in following chapters.

Project settings:

Configuration word:

Select appropriate clock
RB3 as CCP2 pin
Disable watch dog timer (enable if required)
If using ICD, Disable WDT, Low Voltage programming

Language tool suite:

C18 if writing in C
ASM for assembly programming.

Hardware Configuration:

RB3 is the input for CCP2.
RC2 is the output of PWM
Connect a wire from RC2 (remove short link from buzzer) to RB3 on PICDEM 2 PLUS board
S2 & S3 for STOP and START for Capture Module.
R16 (POT) is for varying Duty cycle

NOTE: Please tune delay functions appropriately.
This project developed on 10.0000MHz crystal clock (HS)

INTCON REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF
bit 7							bit 0

See page 5 of 202 PRC handout for details. Bits 7 & 6 will need to be set for TMR1 interrupts to work.

CCP2CON REGISTER—Capture, Compare, PWM 2 Control Register

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
-	-	DC2B1	DC2B0	CCP2M3	CCP2M2	CCP2M1	CCP2M0
bit 7							bit 0

bit 7-6 **Unimplemented:** Read as '0'

bit 5-4 **DCxB1:DCxB0**: PWM Duty Cycle bit 1 and bit 0 for CCP Module x

Capture mode:
Unused.
Compare mode:
Unused.
PWM mode:
These bits are the two LSbs (bit 1 and bit 0) of the 10-bit PWM duty cycle. The eight MSbs (DCx9:DCx2) of the duty cycle are found in CCPRxL.

bit 3-0 **CCPxM3:CCPxM0**: CCP Module x Mode Select bits

0000 = Capture/Compare/PWM disabled (resets CCP module)
0001 = Reserved
0010 = Compare mode, toggle output on match (CCPxIF bit is set)
0011 = Reserved
0100 = Capture mode, every falling edge
0101 = Capture mode, every rising edge
0110 = Capture mode, every 4th rising edge
0111 = Capture mode, every 16th rising edge
1000 = Compare mode: initialize CCP pin low; on compare match, force CCP pin high (CCPIF bit is set)
1001 = Compare mode: initialize CCP pin high; on compare match, force CCP pin low (CCPIF bit is set)
1010 = Compare mode: generate software interrupt on compare match (CCPxIF bit is set, CCP pin reflects I/O state)
1011 = Compare mode: trigger special event, reset timer, start A/D conversion on CCP2 match (CCPxIF bit is set)
11xx = PWM mode

Lab 5

T3CON REGISTER—Timer 3 Control Register 1

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
RD16	T3CCP2	T3CKPS1	T3CKPS0	T3CCP1	$\overline{\text{T3SYNC}}$	TMR3CS	TMR3ON
bit 7							bit 0

bit 7 **RD16:** 16-bit Read/Write Mode Enable bit

1 = Enables register read/write of Timer3 in one 16-bit operation

0 = Enables register read/write of Timer3 in two 8-bit operations

bit 6,3 **T3CCP2:T3CCP1:** Timer3 and Timer1 to CCPx Enable bits

Careful! Non-contiguous Bits

1x = Timer3 is the capture/compare clock source for the CCP modules

01 = Timer3 is the capture/compare clock source for CCP2;
Timer1 is the capture/compare clock source for CCP1

00 = Timer1 is the capture/compare clock source for the CCP modules

bit 5-4 **T3CKPS1:T3CKPS0**: Timer3 Input Clock Prescale Select bits

11 = 1:8 Prescale value

10 = 1:4 Prescale value

01 = 1:2 Prescale value

00 = 1:1 Prescale value

bit 2 **T3SYNC:** Timer3 External Clock Input Synchronization Control bit

(Not usable if the device clock comes from Timer1/Timer3.)

When TMR3CS = 1:

1 = Do not synchronize external clock input

0 = Synchronize external clock input (Default Value)

When TMR3CS = 0:

This bit is ignored. Timer3 uses the internal clock when TMR3CS = 0.

bit 1 **TMR3CS:** Timer3 Clock Source Select bit

1 = External clock input from Timer1 oscillator or T13CKI (on the rising edge after the first falling edge)

0 = Internal clock (FOSC/4) (Default Value)

bit 0 **TMR3ON:** Timer3 On bit

1 = Enables Timer3

0 = Stops Timer3

PIE2 REGISTER—Peripheral Interrupt Enable 2 Register

R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
OSCFIE	CMIE	-	EEIE	BCLIE	HLVDIE	TMR3IE	CCP2IE
bit 7							bit 0

bit 7 **OSCFIE:** Oscillator Fail Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 6 **CMIE:** Comparator Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 5 **Unimplemented:** Read as '0'

bit 4 **EEIE:** Data EEPROM/Flash Write Operation Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 3 **BCLIE:** Bus Collision Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 2 **HLVDIE:** High/Low-Voltage Detect Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 1 **TMR3IE:** TMR3 Overflow Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 0 **CCP2IE:** CCP2 Interrupt Enable bit
1 = Enabled
0 = Disabled

Lab 5

PIR2 REGISTER—Peripheral Interrupt Request Register 2 (Flags)

R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
OSCFIF	CMIF	-	EEIF	BCLIF	HLVDIF	TMR3IF	CCP2IF
bit 7							bit 0

bit 7 **OSCFIF:** Oscillator Fail Interrupt Flag bit
1 = Device oscillator failed, clock input has changed to INTOSC (must be cleared in software)
0 = Device clock operating

bit 6 **CMIF:** Comparator Interrupt Flag bit
1 = Comparator input has changed (must be cleared in software)
0 = Comparator input has not changed

bit 5 **Unimplemented:** Read as '0'

bit 4 **EEIF:** Data EEPROM/Flash Write Operation Interrupt Flag bit
1 = The write operation is complete (must be cleared in software)
0 = The write operation is not complete or has not been started

bit 3 **BCLIF:** Bus Collision Interrupt Flag bit
1 = A bus collision occurred (must be cleared in software)
0 = No bus collision occurred

bit 2 **HLVDIF:** High/Low-Voltage Detect Interrupt Flag bit
1 = A high/low-voltage condition occurred (direction determined by VDIRMAG bit, HLVDCON<7>)
0 = A high/low-voltage condition has not occurred

bit 1 **TMR3IF:** TMR3 Overflow Interrupt Flag bit
1 = TMR3 register overflowed (must be cleared in software)
0 = TMR3 register did not overflow

bit 0 **CCP2IF:** CCPx Interrupt Flag bit
Capture mode:
1 = A TMR1/3 register capture occurred (must be cleared in software)
0 = No TMR1/3 register capture occurred
Compare mode:
1 = A TMR1 register compare match occurred (must be cleared in software)
0 = No TMR1 register compare match occurred
PWM mode:
Unused in this mode.

Lab 6 - MSSP in I²C Mode

Description:

A TC74 Serial Digital Thermal Sensor is used to measure ambient temperature. The PIC and TC74 communicate using the MSSP module. The TC74 is connected to the SDA & SCL I/O pins of the PIC and functions as a slave. Measured temperature is displayed in the LCD.

Project settings:

Configuration word:

Select appropriate clock (HS in the current example)
Disable watch dog timer (enable if required)
If using ICD, Disable WDT, Low Voltage programming

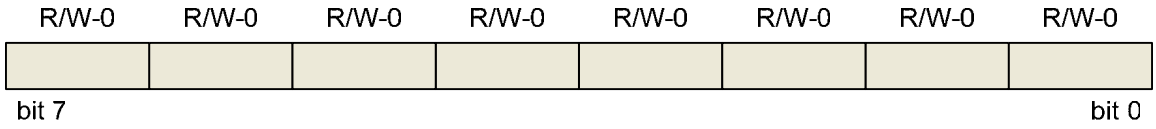
Language tool suite:

C18 if writing in C
ASM for assembly programming.

Hardware Configuration:

NOTE: All delays in this project are calculated based on 10.0000MHz crystal

SSPADD REGISTER—Baud Rate Value (I²C Mode)



In I2C Master Mode this register is used for baud rate generation. In this application, MSSP is configured in I2C master mode with the TC74 as a slave device.

Baud Rate = F_{osc} / (4 * (SSPADD + 1)

F_{osc} = 10.000MHz and **SSPADD** = 0x63 (any desired value—as master you are in control of the baud rate)

SSPCON1 REGISTER—MSSP Control Register 1 (I^2C Mode)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0
bit 7							bit 0

bit 7 **WCOL:** Write Collision Detect bit

In Master Transmit mode:
1 = A write to the SSPBUF register was attempted while the I2C conditions were not valid for a transmission to be started (must be cleared in software)
0 = No collision

In Slave Transmit mode:
1 = The SSPBUF register is written while it is still transmitting the previous word (must be cleared in software)
0 = No collision

In Receive mode (Master or Slave modes):
This is a “don’t care” bit.

bit 6 **SSPOV:** Receive Overflow Indicator bit

In Receive mode:
1 = A byte is received while the SSPBUF register is still holding the previous byte (must be cleared in software)
0 = No overflow

In Transmit mode:
This is a “don’t care” bit in Transmit mode.

bit 5 **SSPEN:** Synchronous Serial Port Enable bit

1 = Enables the serial port and configures the SDA and SCL pins as the serial port pins
0 = Disables serial port and configures these pins as I/O port pins

Note: When enabled, the SDA and SCL pins must be properly configured as input or output.

bit 4 **CKP:** SCK Release Control bit

In Slave mode:
1 = Release clock
0 = Holds clock low (clock stretch), used to ensure data setup time

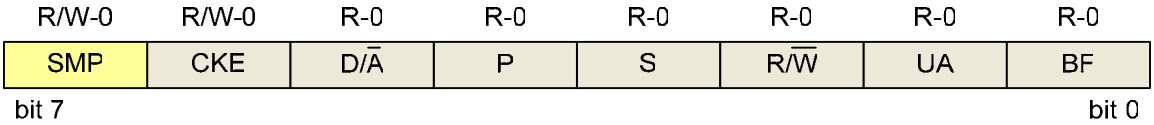
In Master mode:
Unused in this mode.

bit 3-0 **SSPM3:SSPM0:** Synchronous Serial Port Mode Select bits

1111 = I2C Slave mode, 10-bit address with Start and Stop bit interrupts enabled
1110 = I2C Slave mode, 7-bit address with Start and Stop bit interrupts enabled
1011 = I2C Firmware Controlled Master mode (Slave Idle)
1000 = I2C Master mode, clock = FOSC/(4 * (SSPADD + 1))
0111 = I2C Slave mode, 10-bit address
0110 = I2C Slave mode, 7-bit address
Bit combinations not specifically listed here are either reserved or implemented in SPI mode only.

Lab 6

SSPSTAT REGISTER—MSSP Status Register (I^2C Mode)



bit 7 **SMP:** Slew Rate Control bit (In Master or Slave Mode)
1 = Slew rate control disabled for standard speed mode (100 kHz and 1 MHz)
0 = Slew rate control enabled for high-speed mode (400 kHz)

bit 6 **CKE:** SMBus Select bit (In Master or Slave Mode)
1 = Enable SMBus specific inputs
0 = Disable SMBus specific inputs

bit 5 **D/A:** Data/Address bit
In Master mode:
Reserved.
In Slave mode:
1 = Indicates that the last byte received or transmitted was data
0 = Indicates that the last byte received or transmitted was address

bit 4 **P:** Stop bit
1 = Indicates that a Stop bit has been detected last
0 = Stop bit was not detected last
Note: This bit is cleared on Reset and when SSPEN is cleared.

bit 3 **S:** Start bit
1 = Indicates that a Start bit has been detected last
0 = Start bit was not detected last
Note: This bit is cleared on Reset and when SSPEN is cleared.

bit 2 **R/W:** Read/Write Information bit (I^2C mode only)
In Slave mode:
1 = Read
0 = Write
Note: This bit holds the R/W bit information following the last address match. This bit is only valid from the address match to the next Start bit, Stop bit or not ACK bit.
In Master mode:
1 = Transmit is in progress
0 = Transmit is not in progress
Note: ORing this bit with SEN, RSEN, PEN, RCEN or ACKEN will indicate if the MSSP is in Active mode.

bit 1 **UA:** Update Address bit (10-bit Slave mode only)
1 = Indicates that the user needs to update the address in the SSPADD register
0 = Address does not need to be updated

bit 0 **BF:** Buffer Full Status bit
In Transmit mode:
1 = SSPBUF is full
0 = SSPBUF is empty
In Receive mode:
1 = SSPBUF is full (does not include the ACK and Stop bits)
0 = SSPBUF is empty (does not include the ACK and Stop bits)

Lab 6

Lab 7 - EUSART

Description:

USART module is configured to asynchronous mode. Baud Rate of 19200 is selected.
Firmware waits for PC to send a byte; when it receives a byte from PC, firmware will check for following condition;
a) If received byte is 'T', then firmware will read on board temperature sensor and transmits measure temperature to PC over RS232 in the format **"Temp = xxx ° C"**.
b) If received byte is 'S', then firmware will replay to PC with a string **"MICROCHIP MASTER"**.
c) If any other byte is received, firmware will reply with the same byte.

Project settings:

Configuration word:

Select appropriate clock (HS in the current example)
Disable watch dog timer (enable if required)
If using ICD, Disable WDT, Low Voltage programming

Language tool suite:

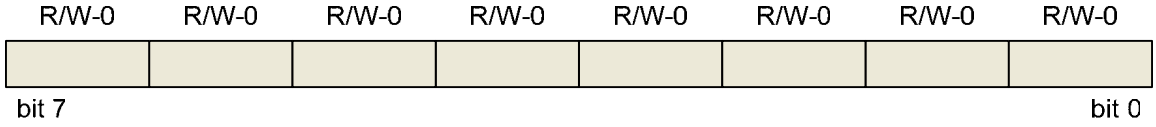
C18 if writing in C
ASM for assembly programming.

Hardware Configuration:

An RS232 cable (D9) from PC to PICDEM2PLUS board

NOTE: Please tune baud rate appropriately. This project developed on 10.0000MHz crystal clock (HS), Baud rate in the current example is 19200.

SPBRG REGISTER—EUSART Baud Rate Value



For a baud rate of 19200 with BRGH = 1 (See TXSTA Register), we use a formula from the data sheet to determine the value required for the SPBRG Register.

BaudRate = F_{OSC} / [16 • (SPBRG + 1)]
19200 = 10MHz / [16 • (SPBRG + 1)]

SPBRG = 31 (decimal)

TXSTA REGISTER—EUSART Transmit Status Register

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-1	R/W-0
CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D
bit 7							bit 0

bit 7 **CSRC:** Clock Source Select bit
Asynchronous mode:
Don't care.
Synchronous mode:
1 = Master mode (clock generated internally from BRG)
0 = Slave mode (clock from external source)

bit 6 **TX9:** 9-bit Transmit Enable bit
1 = Selects 9-bit transmission
0 = Selects 8-bit transmission

bit 5 **TXEN:** Transmit Enable bit
1 = Transmit enabled
0 = Transmit disabled
Note: SREN/CREN overrides TXEN in Sync mode.

bit 4 **SYNC:** EUSART Mode Select bit
1 = Synchronous mode
0 = Asynchronous mode

bit 3 **SENDB:** Send Break Character bit
Asynchronous mode:
1 = Send Sync Break on next transmission (cleared by hardware upon completion)
0 = Sync Break transmission completed
Synchronous mode:
Don't care.

bit 2 **BRGH:** High Baud Rate Select bit
Asynchronous mode:
1 = High speed
0 = Low speed
Synchronous mode:
Unused in this mode.

bit 1 **TRMT:** Transmit Shift Register Status bit
1 = TSR empty
0 = TSR full

bit 0 **TX9D:** 9th bit of Transmit Data
Can be address/data bit or a parity bit.

Lab 7

RCSTA—EUSART Receive Status Register

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R-0	R-x
SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D
bit 7							bit 0

bit 7 **SPEN:** Serial Port Enable bit
1 = Serial port enabled (configures RX/DT and TX/CK pins as serial port pins)
0 = Serial port disabled (held in Reset)

bit 6 **RX9:** 9-bit Receive Enable bit
1 = Selects 9-bit reception
0 = Selects 8-bit reception

bit 5 **SREN:** Single Receive Enable bit
Asynchronous mode:
Don't care.
Synchronous mode – Master:
1 = Enables single receive
0 = Disables single receive
This bit is cleared after reception is complete.
Synchronous mode – Slave:
Don't care.

bit 4 **CREN:** Continuous Receive Enable bit
Asynchronous mode:
1 = Enables receiver
0 = Disables receiver
Synchronous mode:
1 = Enables continuous receive until enable bit CREN is cleared (CREN overrides SREN)
0 = Disables continuous receive

bit 3 **ADDEN:** Address Detect Enable bit
Asynchronous mode 9-bit (RX9 = 1):
1 = Enables address detection, enables interrupt and loads the receive buffer when RSR<8> is set
0 = Disables address detection, all bytes are received and ninth bit can be used as parity bit
Asynchronous mode 9-bit (RX9 = 0):
Don't care.

bit 2 **FERR:** Framing Error bit
1 = Framing error (can be updated by reading RCREG register and receiving next valid byte)
0 = No framing error

bit 1 **OERR:** Overrun Error bit
1 = Overrun error (can be cleared by clearing bit CREN)
0 = No overrun error

bit 0 **RX9D:** 9th bit of Received Data
This can be address/data bit or a parity bit and must be calculated by user firmware.

INTCON REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF
bit 7							bit 0

See page 5 of 202 PRC handout for details. Bits 7 & 6 will need to be set for TMR1 interrupts to work.

PIE1 REGISTER—Peripheral Interrupt Enable Register 1

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PSPIE(1)	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE
bit 7							bit 0

See page 14 of 202 PRC handout for details. Ensure that transmit interrupts are *disabled* and that receive interrupts are *enabled.*

PIR1 REGISTER—Peripheral Interrupt Request Register 1 (Flags)

R/W-0	R/W-0	R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
PSPIF(1)	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF
bit 7							bit 0

See page 15 of 202 PRC handout for details.

202 PRC