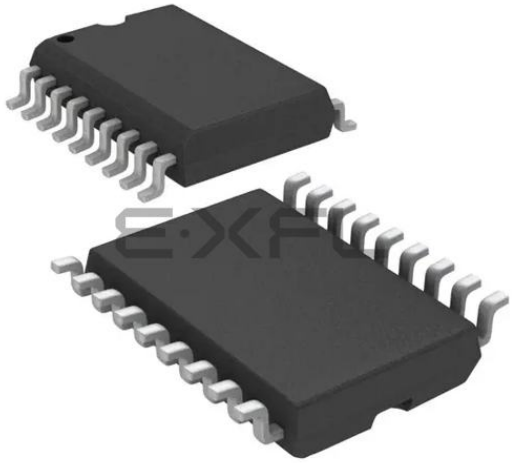


Welcome to E-XFL.COM

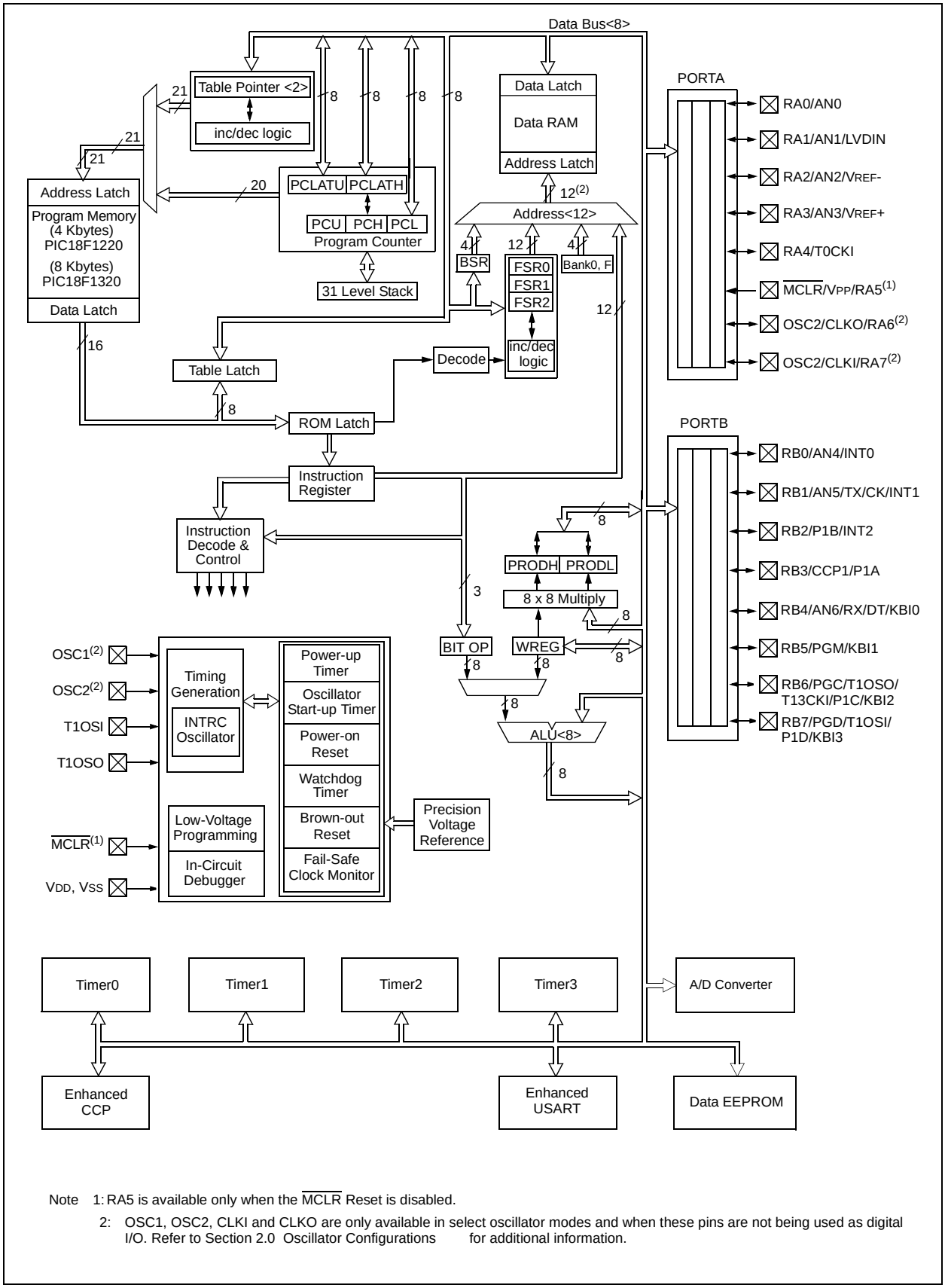
What is "Embedded - Microcontrollers"?

"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 "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	25MHz
Connectivity	UART/USART
Peripherals	Brown-out Detect/Reset, LVD, POR, PWM, WDT
Number of I/O	16
Program Memory Size	4KB (2K x 16)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	4.2V ~ 5.5V
Data Converters	A/D 7x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 150°C (TA)
Mounting Type	Surface Mount
Package / Case	18-SOIC (0.295", 7.50mm Width)
Supplier Device Package	18-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f1220-h-so

FIGURE 1-1: PIC18F1220/1320 BLOCK DIAGRAM



Note 1: RA5 is available only when the $\overline{\text{MCLR}}$ Reset is disabled.

2: OSC1, OSC2, CLKI and CLKO are only available in select oscillator modes and when these pins are not being used as digital I/O. Refer to Section 2.0 Oscillator Configurations for additional information.

3.3.2 SEC_IDLE MODE

In SEC_IDLE mode, the CPU is disabled, but the peripherals continue to be clocked from the Timer1 oscillator. This mode is entered by setting the Idle bit, modifying bits, SCS1:SCS0 = and executing a SLEEP instruction. When the clock source is switched (see Figure 3-5) to the Timer1 oscillator, the primary oscillator is shut down, the OSTS bit is cleared and the T1RUN bit is set.

Note:	The Timer1 oscillator should already be running prior to entering SEC_IDLE mode. If the T1OSCEN bit is not set when the SLEEP instruction is executed, the SLEEP instruction will be ignored and entry to SEC_IDLE mode will not occur. If the Timer1 oscillator is enabled, but not yet running, peripheral clocks will be delayed until the oscillator has started; in such situations, initial oscillator operation is far from stable and unpredictable operation may result.

When a wake event occurs, the peripherals continue to be clocked from the Timer1 oscillator. After a 10 Ps delay following the wake event, the CPU begins executing code, being clocked by the Timer1 oscillator. The microcontroller operates in SEC_RUN mode until the primary clock becomes ready. When the primary clock becomes ready, a clock switchback to the primary clock occurs (see Figure 3-6). When the clock switch is complete, the T1RUN bit is cleared, the OSTS bit is set and the primary clock is providing the system clock. The IDLEN and SCS bits are not affected by the wake-up. The Timer1 oscillator continues to run.

FIGURE 3-5: TIMING TRANSITION FOR ENTRY TO SEC_IDLE MODE

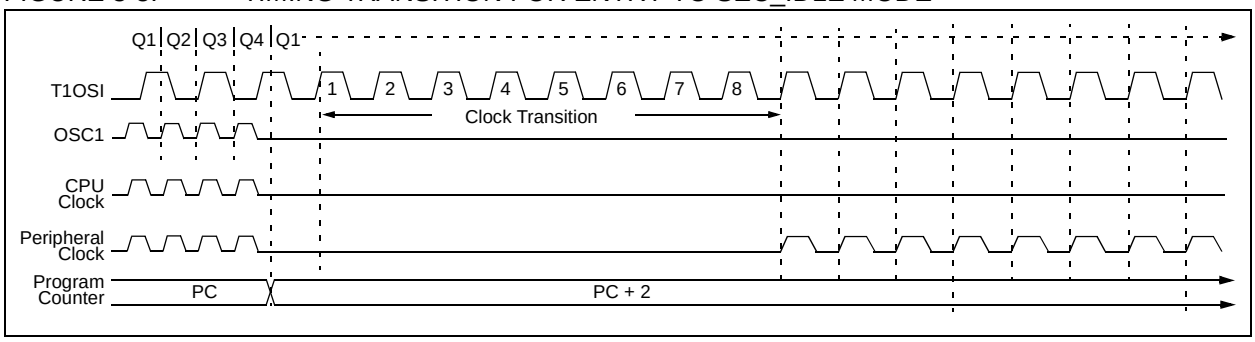
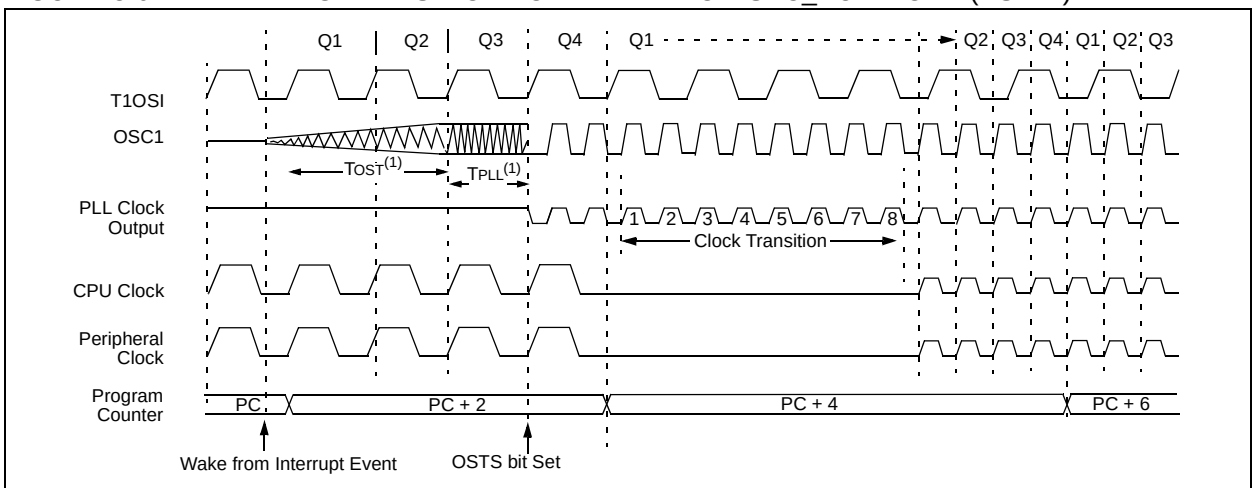


FIGURE 3-6: TIMING TRANSITION FOR WAKE FROM SEC_RUN MODE (HSPLL)



Note 1: TOST = 1024 TOSC; TPLL = 2 ms (approx). These intervals are not shown to scale.

5.9.2 SPECIAL FUNCTION REGISTERS

The Special Function Registers (SFRs) are registers used by the CPU and peripheral modules for controlling the desired operation of the device. These registers are implemented as static RAM. A list of these registers is given in Table 5-1 and Table 5-2.

The SFRs can be classified into two sets: those associated with the core function and those related to the peripheral functions. Those registers related to the core are described in this section, while those related to the operation of the peripheral features are described in the section of that peripheral feature.

The SFRs are typically distributed among the peripherals whose functions they control.

The unused SFR locations will be unimplemented and read as `0` s.

TABLE 5-1: SPECIAL FUNCTION REGISTER MAP FOR PIC18F1220/1320 DEVICES

Address	Name	Address	Name	Address	Name	Address	Name
FFFh	TOSU	FDFh	INDF2(2)	FBFh	CCPR1H	F9Fh	IPR1
FFEh	TOSH	FDEh	POSTINC2(2)	FBEh	CCPR1L	F9Eh	PIR1
FFDh	TOSL	FDDh	POSTDEC2(2)	FBDh	CCP1CON	F9Dh	PIE1
FFCh	STKPTR	FDCh	PREINC2(2)	FBCh		F9Ch	
FFBh	PCLATU	FDBh	PLUSW2(2)	FBBh		F9Bh	OSCTUNE
FFAh	PCLATH	FDAh	FSR2H	FBAh		F9Ah	
FF9h	PCL	FD9h	FSR2L	FB9h		F99h	
FF8h	TBLPTRU	FD8h	STATUS	FB8h		F98h	
FF7h	TBLPTRH	FD7h	TMR0H	FB7h	PWM1CON	F97h	
FF6h	TBLPTRL	FD6h	TMR0L	FB6h	ECCPAS	F96h	
FF5h	TABLAT	FD5h	T0CON	FB5h		F95h	
FF4h	PRODH	FD4h		FB4h		F94h	
FF3h	PRODL	FD3h	OSCCON	FB3h	TMR3H	F93h	TRISB
FF2h	INTCON	FD2h	LVDCON	FB2h	TMR3L	F92h	TRISA
FF1h	INTCON2	FD1h	WDTCON	FB1h	T3CON	F91h	
FF0h	INTCON3	FD0h	RCON	FB0h	SPBRGH	F90h	
FEFh	INDF0(2)	FCFh	TMR1H	FAFh	SPBRG	F8Fh	
FEEh	POSTINC0(2)	FCEh	TMR1L	FAEh	RCREG	F8Eh	
FEDh	POSTDEC0(2)	FCDh	T1CON	FADh	TXREG	F8Dh	
FECh	PREINC0(2)	FCCh	TMR2	FACh	TXSTA	F8Ch	
FEBh	PLUSW0(2)	FCBh	PR2	FABh	RCSTA	F8Bh	
FEAh	FSR0H	FCAh	T2CON	FAAh	BAUDCTL	F8Ah	LATB
FE9h	FSR0L	FC9h		FA9h	EEADR	F89h	LATA
FE8h	WREG	FC8h		FA8h	EEDATA	F88h	
FE7h	INDF1(2)	FC7h		FA7h	EECON2	F87h	
FE6h	POSTINC1(2)	FC6h		FA6h	EECON1	F86h	
FE5h	POSTDEC1(2)	FC5h		FA5h		F85h	
FE4h	PREINC1(2)	FC4h	ADRESH	FA4h		F84h	
FE3h	PLUSW1(2)	FC3h	ADRESL	FA3h		F83h	
FE2h	FSR1H	FC2h	ADCON0	FA2h	IPR2	F82h	
FE1h	FSR1L	FC1h	ADCON1	FA1h	PIR2	F81h	PORTB
FE0h	BSR	FC0h	ADCON2	FA0h	PIE2	F80h	PORTA

Note 1: Unimplemented registers are read as `0` .

2: This is not a physical register.

7.0 DATA EEPROM MEMORY

The data EEPROM is readable and writable during normal operation over the entire VDD range. The data memory is not directly mapped in the register file space. Instead, it is indirectly addressed through the Special Function Registers (SFR).

There are four SFRs used to read and write the program and data EEPROM memory. These registers are:

EECON1
EECON2
EEDATA
EEADR

The EEPROM data memory allows byte read and write. When interfacing to the data memory block, EEDATA holds the 8-bit data for read/write and EEADR holds the address of the EEPROM location being accessed. These devices have 256 bytes of data EEPROM with an address range from 00h to FFh.

The EEPROM data memory is rated for high erase/write cycle endurance. A byte write automatically erases the location and writes the new data (erase-before-write). The write time is controlled by an on-chip timer. The write time will vary with voltage and temperature, as well as from chip to chip. Please refer to parameter D122 (Table 22-1 in Section 22.0 Electrical Characteristics) for exact limits.

7.1 EEADR

The address register can address 256 bytes of data EEPROM.

7.2 EECON1 and EECON2 Registers

EECON1 is the control register for memory accesses.

EECON2 is not a physical register. Reading EECON2 will read all 0 s. The EECON2 register is used exclusively in the memory write and erase sequences.

Control bit, EEPGD, determines if the access will be to program or data EEPROM memory. When clear, operations will access the data EEPROM memory. When set, program memory is accessed.

Control bit, CFGS, determines if the access will be to the Configuration registers or to program memory/data EEPROM memory. When set, subsequent operations access Configuration registers. When CFGS is clear, the EEPGD bit selects either program Flash or data EEPROM memory.

The WREN bit enables and disables erase and write operations. When set, erase and write operations are allowed. When clear, erase and write operations are disabled the WR bit cannot be set while the WREN bit is clear. This mechanism helps to prevent accidental writes to memory due to errant (unexpected) code execution.

Firmware should keep the WREN bit clear at all times, except when starting erase or write operations. Once firmware has set the WR bit, the WREN bit may be cleared. Clearing the WREN bit will not affect the operation in progress.

The WRERR bit is set when a write operation is interrupted by a Reset. In these situations, the user can check the WRERR bit and rewrite the location. It is necessary to reload the data and address registers (EEDATA and EEADR), as these registers have cleared as a result of the Reset.

Control bits, RD and WR, start read and erase/write operations, respectively. These bits are set by firmware and cleared by hardware at the completion of the operation.

The RD bit cannot be set when accessing program memory (EEPGD = 1). Program memory is read using table read instructions. See Section 6.1 Table Reads and Table Writes regarding table reads.

Note:	Interrupt flag bit, EEIF in the PIR2 register, is set when write is complete. It must be cleared in software.

¤ 2002-2015 Microchip Technology Inc.

REGISTER 9-2: INTCON2: INTERRUPT CONTROL REGISTER 2

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

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as 0
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
1 = Bit is set	0 = Bit is cleared	

bit 7 $\overline{\text{RBPU}}$: 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

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

REGISTER 9-5: PIR2: PERIPHERAL INTERRUPT REQUEST REGISTER 2

R/W-0/0	U-0	U-0	R/W-0/0	U-0	R/W-0/0	R/W-0/0	U-0
OSCFIF			EEIF		LVDIF	TMR3IF	
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as 0
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
1 = Bit is set	0 = Bit is cleared	

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

bit 6-5 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 Unimplemented: Read as 0

bit 2 LVDIF: Low-Voltage Detect Interrupt Flag bit
1 = A low-voltage condition occurred (must be cleared in software)
0 = The device voltage is above the Low-Voltage Detect trip point

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 Unimplemented: Read as 0

¤ 2002-2015 Microchip Technology Inc.

9.3 PIE Registers

The PIE registers contain the individual enable bits for the peripheral interrupts. Due to the number of peripheral interrupt sources, there are two Peripheral Interrupt Enable registers (PIE1, PIE2). When IPEN = 0, the PEIE bit must be set to enable any of these peripheral interrupts.

REGISTER 9-6: PIE1: PERIPHERAL INTERRUPT ENABLE REGISTER 1

U-0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
	ADIE	RCIE	TXIE		CCP1IE	TMR2IE	TMR1IE
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as 0
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
1 = Bit is set	0 = Bit is cleared	

bit 7 Unimplemented: Read as 0

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

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

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

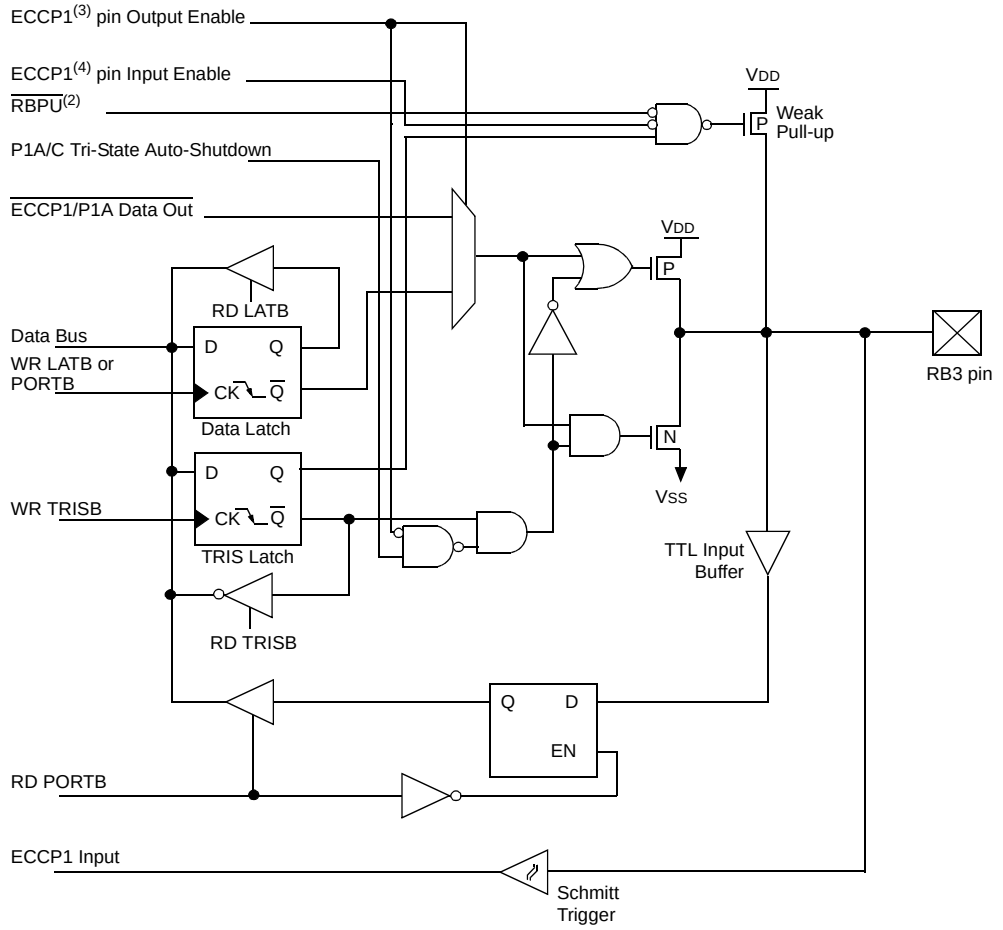
bit 3 Unimplemented: Read as 0

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

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

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

FIGURE 10-10: BLOCK DIAGRA M OF RB3/CCP1/P1A PIN



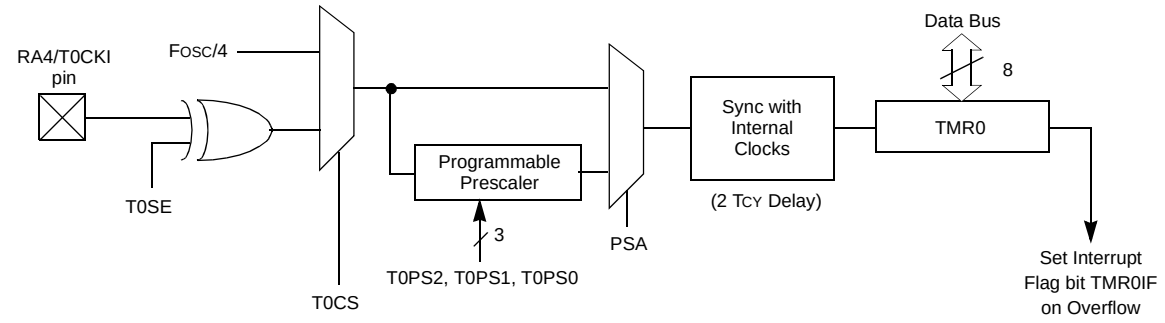
Note 1: I/O pins have diode protection to VDD and VSS.

2: To enable weak pull-ups, set the appropriate TRIS bit(s) and clear the RBPU bit (INTCON2<7>).

3: ECCP1 pin output enable active for any PWM mode and Compare mode, where CCP1M<3:0> = or .

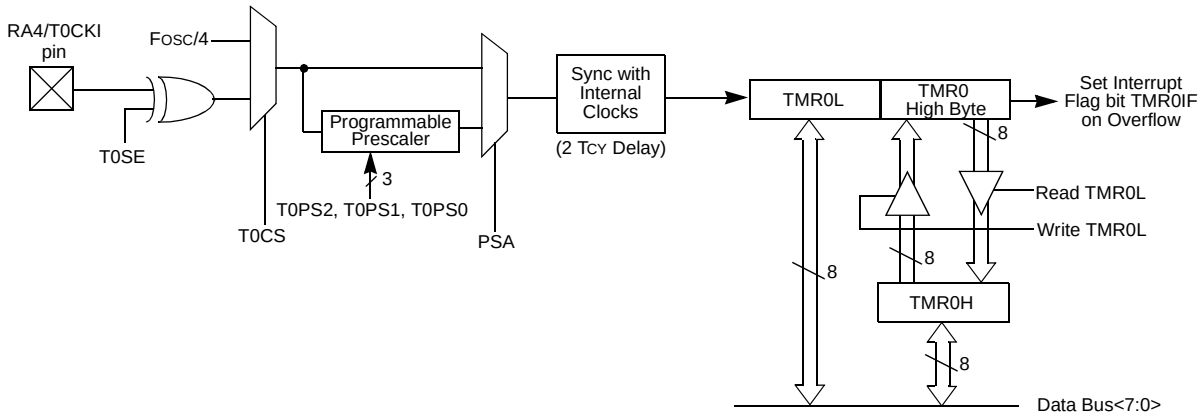
4: ECCP1 pin input enable active for Capture mode only.

FIGURE 11-1: TIMER0 BLOCK DI AGRAM IN 8-BIT MODE



Note: Upon Reset, Timer0 is enabled in 8-bit mode with clock input from T0CKI maximum prescale.

FIGURE 11-2: TIMER0 BLOCK DIAGRAM IN 16-BIT MODE



Note: Upon Reset, Timer0 is enabled in 8-bit mode with clock input from T0CKI maximum prescale.

¤ 2002-2015 Microchip Technology Inc.

12.7 Using Timer1 as a Real-Time Clock

Adding an external LP oscillator to Timer1 (such as the one described in Section 12.2 Timer1 Oscillator , above), gives users the option to include RTC functionality to their applications. This is accomplished with an inexpensive watch crystal to provide an accurate time base and several lines of application code to calculate the time. When operating in Sleep mode and using a battery or supercapacitor as a power source, it can completely eliminate the need for a separate RTC device and battery backup.

The application code routine, RTCisr, shown in Example 12-1, demonstrates a simple method to increment a counter at one-second intervals using an Interrupt Service Routine. Incrementing the TMR1 register pair to overflow, triggers the interrupt and calls the routine, which increments the seconds counter by one; additional counters for minutes and hours are incremented as the previous counter overflow.

Since the register pair is 16 bits wide, counting up to overflow the register directly from a 32.768 kHz clock would take two seconds. To force the overflow at the required one-second intervals, it is necessary to preload it; the simplest method is to set the MSb of TMR1H with a BSF instruction. Note that the TMR1L register is never preloaded or altered; doing so may introduce cumulative error over many cycles.

For this method to be accurate, Timer1 must operate in Asynchronous mode and the Timer1 overflow interrupt must be enabled (PIE1<0> = 1), as shown in the routine, RTCinit. The Timer1 oscillator must also be enabled and running at all times.

EXAMPLE 12-1: IMPLEMENTING A REAL-TIME CLOCK USING A TIMER1 INTERRUPT SERVICE

```
RTCinit
            MOVLW   0x80              ; Preload TMR1 register pair
            MOVWF   TMR1H             ; for 1 second overflow
            CLRF    TMR1L
            MOVLW   b'00001111'       ; Configure for external clock,
            MOVWF   T1OSC             ; Asynchronous operation, external oscillator
            CLRF    secs              ; Initialize timekeeping registers
            CLRF    mins              ;
            MOVLW   .12
            MOVWF   hours
            BSF     PIE1, TMR1IE      ; Enable Timer1 interrupt
            RETURN
RTCisr
            BSF     TMR1H, 7          ; Preload for 1 sec overflow
            BCF     PIR1, TMR1IF      ; Clear interrupt flag
            INCF    secs, F           ; Increment seconds
            MOVLW   .59               ; 60 seconds elapsed?
            CPFSGT  secs
            RETURN                    ; No, done
            CLRF    secs              ; Clear seconds
            INCF    mins, F           ; Increment minutes
            MOVLW   .59               ; 60 minutes elapsed?
            CPFSGT  mins
            RETURN                    ; No, done
            CLRF    mins              ; clear minutes
            INCF    hours, F          ; Increment hours
            MOVLW   .23               ; 24 hours elapsed?
            CPFSGT  hours
            RETURN                    ; No, done
            MOVLW   .01               ; Reset hours to 1
            MOVWF   hours
            RETURN                    ; Done
```


¤ 2002-2015 Microchip Technology Inc.

15.0 ENHANCED CAPTURE/ COMPARE/PWM (ECCP) MODULE

The Enhanced CCP module is implemented as a standard CCP module with Enhanced PWM capabilities. These capabilities allow for two or four output channels, user-selectable polarity, dead-band control and automatic shutdown and restart and are discussed in detail in Section 15.5 Enhanced PWM Mode .

The control register for CCP1 is shown in Register 15-1.

In addition to the expanded functions of the CCP1CON register, the ECCP module has two additional registers associated with Enhanced PWM operation and auto-shutdown features:

PWM1CON

ECCPAS

REGISTER 15-1: CCP1CON REGISTER FOR ENHANCED CCP OPERATION

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

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as 0
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
1 = Bit is set	0 = Bit is cleared	

bit 7-6 P1M<1:0>: PWM Output Configuration bits

If CCP1M<3:2> = :

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

If CCP1M<3:2> = :

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

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

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

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

bit 5-4 DC1B<1:0>: PWM Duty Cycle Least Significant bits

Capture mode:

Unused.

Compare mode:

Unused.

PWM mode:

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

bit 3-0 CCP1M<3:0>: ECCP1 Mode Select bits

0000 = Capture/Compare/PWM off (resets ECCP module)
0001 = Unused (reserved)
0010 = Compare mode, toggle output on match (ECCP1IF bit is set)
0011 = Unused (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, set output on match (ECCP1IF bit is set)
1001 = Compare mode, clear output on match (ECCP1IF bit is set)
1010 = Compare mode, generate software interrupt on match (ECCP1IF bit is set, ECCP1 pin returns to port pin operation)
1011 = Compare mode, trigger special event (ECCP1IF bit is set; ECCP resets TMR1 or TMR3 and starts an A/D conversion if the A/D module is enabled)
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

FIGURE 15-2: COMPARE MODE OPERATION BLOCK DIAGRAM

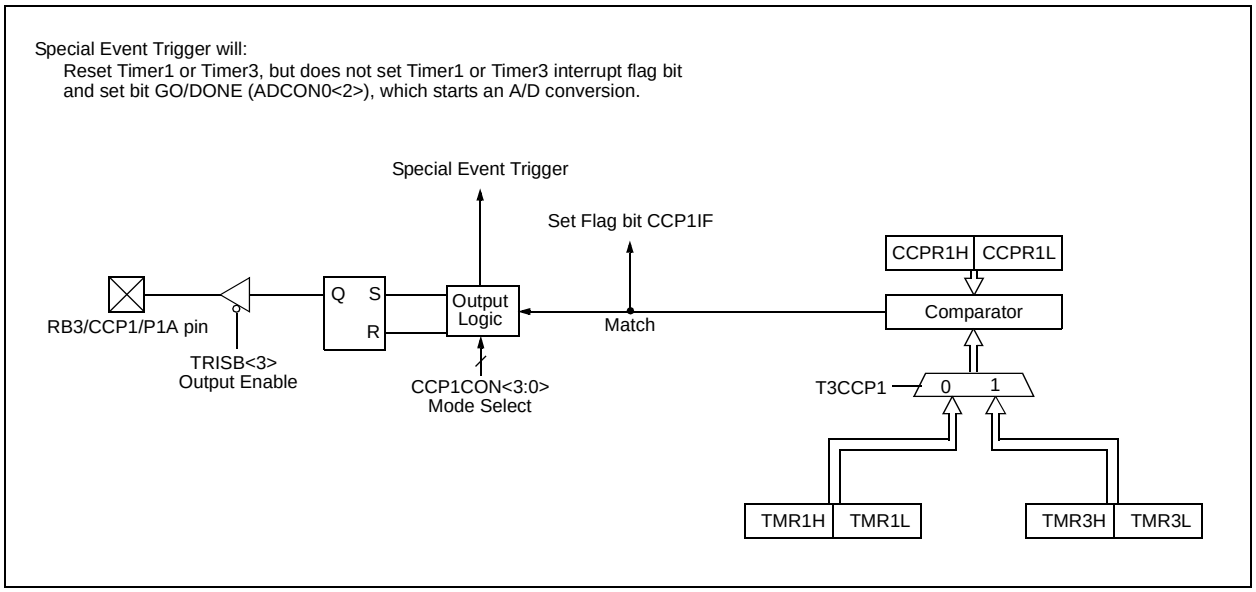


TABLE 15-3: REGISTERS ASSOCIATED WITH CAPTURE, COMPARE, TIMER1 AND TIMER3

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other Resets
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	0000 000u
PIR1		ADIF	RCIF	TXIF		CCP1IF	TMR2IF	TMR1IF	-000 -000	-000 -000
PIE1		ADIE	RCIE	TXIE		CCP1IE	TMR2IE	TMR1IE	-000 -000	-000 -000
IPR1		ADIP	RCIP	TXIP		CCP1IP	TMR2IP	TMR1IP	-111 -111	-111 -111
TRISB	PORTB Data Direction Register								1111 1111	1111 1111
TMR1L	Holding Register for the Least Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	uuuu uuuu
TMR1H	Holding Register for the Most Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	uuuu uuuu
T1CON	RD16	T1RUN	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	TMR1CS	TMR1ON	0000 0000	uuuu uuuu
CCPR1L	Capture/Compare/PWM Register 1 (LSB)								xxxx xxxx	uuuu uuuu
CCPR1H	Capture/Compare/PWM Register 1 (MSB)								xxxx xxxx	uuuu uuuu
CCP1CON	P1M1	P1M0	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	0000 0000	0000 0000
TMR3L	Holding Register for the Least Significant Byte of the 16-bit TMR3 Register								xxxx xxxx	uuuu uuuu
TMR3H	Holding Register for the Most Significant Byte of the 16-bit TMR3 Register								xxxx xxxx	uuuu uuuu
T3CON	RD16		T3CKPS1	T3CKPS0	T3CCP1	$\overline{\text{T3SYNC}}$	TMR3CS	TMR3ON	0-00 0000	u-uu uuuu
ADCON0	VCFG1	VCFG0		CHS2	CHS1	CHS0	GO/$\overline{\text{DONE}}$	ADON		

Legend: x = unknown, u = unchanged, = unimplemented, read as . Shaded cells are not used by Capture and Timer1.

¤ 2002-2015 Microchip Technology Inc.

15.5 Enhanced PWM Mode

The Enhanced PWM Mode provides additional PWM output options for a broader range of control applications. The module is an upwardly compatible version of the standard CCP module and offers up to four outputs, designated P1A through P1D. Users are also able to select the polarity of the signal (either active-high or active-low). The module s output mode and polarity are configured by setting the P1M1:P1M0 and CCP1M3CCP1M0 bits of the CCP1CON register (CCP1CON<7:6> and CCP1CON<3:0>, respectively).

Figure 15-3 shows a simplified block diagram of PWM operation. All control registers are double-buffered and are loaded at the beginning of a new PWM cycle (the period boundary when Timer2 resets) in order to prevent glitches on any of the outputs. The exception is the PWM Delay register, ECCP1DEL, which is loaded at either the duty cycle boundary or the boundary period (whichever comes first). Because of the buffering, the module waits until the assigned timer resets instead of starting immediately. This means that Enhanced PWM waveforms do not exactly match the standard PWM waveforms, but are instead offset by one full instruction cycle (4 TOSC).

As before, the user must manually configure the appropriate TRIS bits for output.

15.5.1 PWM PERIOD

The PWM period is specified by writing to the PR2 register. The PWM period can be calculated using the equation:

EQUATION 15-1: PWM PERIOD

$$\text{PWM Period} = [(PR2) + 1] \cdot 4 \cdot T_{OSC} \cdot (\text{TMR2 Prescale Value})$$

PWM frequency is defined as 1/[PWM period]. When TMR2 is equal to PR2, the following three events occur on the next increment cycle:

TMR2 is cleared

The CCP1 pin is set (if PWM duty cycle = 0%, the CCP1 pin will not be set)

The PWM duty cycle is copied from CCPR1L into CCPR1H

Note:	The Timer2 postscaler (see Section 13.0 Timer2 Module) is not used in the determination of the PWM frequency. The postscaler could be used to have a servo update rate at a different frequency than the PWM output.

15.5.2 PWM DUTY CYCLE

The PWM duty cycle is specified by writing to the CCPR1L register and to the CCP1CON<5:4> bits. Up to 10-bit resolution is available. The CCPR1L contains the eight MSbs and the CCP1CON<5:4> contains the two LSbs. This 10-bit value is represented by CCPR1L:CCP1CON<5:4>. The PWM duty cycle is calculated by the equation:

EQUATION 15-2: PWM DUTY CYCLE

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

CCPR1L and CCP1CON<5:4> can be written to at any time, but the duty cycle value is not copied into CCPR1H until a match between PR2 and TMR2 occurs (i.e., the period is complete). In PWM mode, CCPR1H is a read-only register.

The CCPR1H register and a 2-bit internal latch are used to double-buffer the PWM duty cycle. This double-buffering is essential for glitchless PWM operation. When the CCPR1H and 2-bit latch match TMR2, concatenated with an internal 2-bit Q clock or two bits of the TMR2 prescaler, the CCP1 pin is cleared. The maximum PWM resolution (bits) for a given PWM frequency is given by the equation:

EQUATION 15-3: PWM RESOLUTION

$$\text{PWM Resolution (max)} = \frac{\log\left(\frac{F_{OSC}}{F_{PWM}}\right)}{\log(2)} \text{ bits}$$

Note:	If the PWM duty cycle value is longer than the PWM period, the CCP1 pin will not be cleared.

15.5.3 PWM OUTPUT CONFIGURATIONS

The P1M1:P1M0 bits in the CCP1CON register allow one of four configurations:

Single Output

Half-Bridge Output

Full-Bridge Output, Forward mode

Full-Bridge Output, Reverse mode

The Single Output mode is the Standard PWM mode discussed in Section 15.5 Enhanced PWM Mode . The Half-Bridge and Full-Bridge Output modes are covered in detail in the sections that follow.

The general relationship of the outputs in all configurations is summarized in Figure 15-4.

TABLE 15-4: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS AT 40 MHz

PWM Frequency	2.44 kHz	9.77 kHz	39.06 kHz	156.25 kHz	312.50 kHz	416.67 kHz
Timer Prescaler (1, 4, 16)	16	4	1	1	1	1
PR2 Value	FFh	FFh	FFh	3Fh	1Fh	17h
Maximum Resolution (bits)	10	10	10	8	7	6.58

REGISTER 16-1: TXSTA: TRANSMIT STATUS AND CONTROL REGISTER

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

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as 0
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
1 = Bit is set	0 = Bit is cleared	

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)

1 = Transmit enabled
0 = Transmit disabled

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 Idle
0 = TSR busy

bit 0 TX9D: 9th bit of Transmit Data

Can be address/data bit or a parity bit.

Note 1: SREN/CREN overrides TXEN in Sync mode.

16.3.2 EUSART ASYNCHRONOUS RECEIVER

The receiver block diagram is shown in Figure 16-5. The data is received on the RB4/AN6/RX/DT/KBI0 pin and drives the data recovery block. The data recovery block is actually a high-speed shifter, operating at x16 times the baud rate, whereas the main receive serial shifter operates at the bit rate or at FOSC. This mode would typically be used in RS-232 systems.

To set up an Asynchronous Reception:

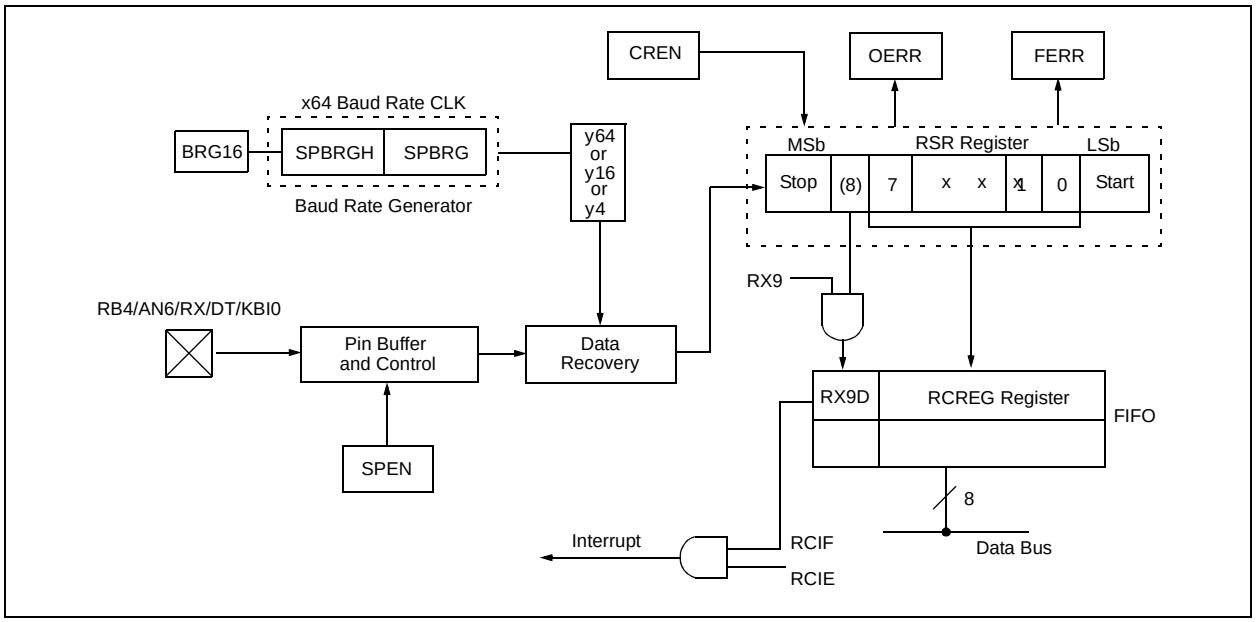
1. Initialize the SPBRGH:SPBRG registers for the appropriate baud rate. Set or clear the BRGH and BRG16 bits, as required, to achieve the desired baud rate.
2. Enable the asynchronous serial port by clearing bit SYNC and setting bit SPEN.
3. If interrupts are desired, set enable bit RCIE.
4. If 9-bit reception is desired, set bit RX9.
5. Enable the reception by setting bit CREN.
6. Flag bit, RCIF, will be set when reception is complete and an interrupt will be generated if enable bit RCIE was set.
7. Read the RCSTA register to get the 9th bit (if enabled) and determine if any error occurred during reception.
8. Read the 8-bit received data by reading the RCREG register.
9. If any error occurred, clear the error by clearing enable bit CREN.
10. If using interrupts, ensure that the GIE and PEIE bits in the INTCON register (INTCON<7:6>) are set.

16.3.3 SETTING UP 9-BIT MODE WITH ADDRESS DETECT

This mode would typically be used in RS-485 systems. To set up an Asynchronous Reception with Address Detect Enable:

1. Initialize the SPBRGH:SPBRG registers for the appropriate baud rate. Set or clear the BRGH and BRG16 bits, as required, to achieve the desired baud rate.
2. Enable the asynchronous serial port by clearing the SYNC bit and setting the SPEN bit.
3. If interrupts are required, set the RCEN bit and select the desired priority level with the RCIP bit.
4. Set the RX9 bit to enable 9-bit reception.
5. Set the ADDEN bit to enable address detect.
6. Enable reception by setting the CREN bit.
7. The RCIF bit will be set when reception is complete. The interrupt will be Acknowledged if the RCIE and GIE bits are set.
8. Read the RCSTA register to determine if any error occurred during reception, as well as read bit 9 of data (if applicable).
9. Read RCREG to determine if the device is being addressed.
10. If any error occurred, clear the CREN bit.
11. If the device has been addressed, clear the ADDEN bit to allow all received data into the receive buffer and interrupt the CPU.

FIGURE 16-5: EUSART RECEIVE BLOCK DIAGRAM



¤ 2002-2015 Microchip Technology Inc.

REGISTER 19-12: DEVID1: DEVICE ID REGISTER 1 FOR PIC18F1220/1320 DEVICES

R	R	R	R	R	R	R	R
DEV2	DEV1	DEV0	REV4	REV3	REV2	REV1	REV0
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as 0
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
1 = Bit is set	0 = Bit is cleared	

bit 7-5 DEV<2:0>: Device ID bits
= PIC18F1220
= PIC18F1320

bit 4-0 REV<4:0>: Revision ID bits
These bits are used to indicate the device revision.

REGISTER 19-13: DEVID2: DEVICE ID REGISTER 2 FOR PIC18F1220/1320 DEVICES

R	R	R	R	R	R	R	R
DEV10	DEV9	DEV8	DEV7	DEV6	DEV5	DEV4	DEV3
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as 0
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
1 = Bit is set	0 = Bit is cleared	

bit 7-0 DEV<10:3>: Device ID bits
These bits are used with the DEV<2:0> bits in the Device ID Register 1 to identify the part number.
= PIC18F1220/1320 devices

Note 1: These values for DEV<10:3> may be shared with other devices. The specific device is always identified by using the entire DEV<10:0> bit sequence.

MOVLW Move literal to W

Syntax: [*label*] MOVLW k

Operands: 0 dk d255

Operation: k o W

Status Affected: None

Encoding:

Description: The 8-bit literal k is loaded into W.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal k	Process Data	Write to W

Example: `MOVLW 0x5A`

After Instruction

W = 0x5A

MOVWF Move W to f

Syntax: [*label*] MOVWF f [,a]

Operands: 0 df d255
a • [0,1]

Operation: (W) o f

Status Affected: None

Encoding:

Description: Move data from W to register f . Location f can be anywhere in the 256-byte bank. If a is 0 , the Access Bank will be selected, overriding the BSR value. If a =1, then the bank will be selected as per the BSR value (default).

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register f	Process Data	Write register f

Example: `MOVWF REG`

Before Instruction

W = 0x4F
REG = 0xFF

After Instruction

W = 0x4F
REG = 0x4F

¤ 2002-2015 Microchip Technology Inc.

22.2 DC Characteristics: Power-Down and Supply Current PIC18F1220/1320 (Industrial) PIC18LF1220/1320 (Industrial) (Continued)

PIC18LF1220/1320 (Industrial)	Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C dTA d+85°C for industrial
PIC18F1220/1320 (Industrial, Extended)	Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C dTA d+85°C for industrial -40°C dTA d+125°C for extended

Param No.	Device	Typ.	Max.	Units	Conditions		
	Supply Current (IDD)(2,3)						
	PIC18LF1220/1320	415	600	µA	-40°C	VDD = 2.0V	FOSC = 4 MHz (PRI_RUN mode, EC oscillator)
		425	600	µA	+25°C		
		435	600	µA	+85°C		
	PIC18LF1220/1320	0.87	1.0	mA	-40°C	VDD = 3.0V	
		0.75	1.0	mA	+25°C		
		0.75	1.0	mA	+85°C		
	All devices	1.6	2.0	mA	-40°C	VDD = 5.0V	
		1.6	2.0	mA	+25°C		
		1.5	2.0	mA	+85°C		
	Extended devices	1.5	2.0	mA	+125°C		
	Extended devices	6.3	9.0	mA	+125°C	VDD = 4.2V	FOSC = 25 MHz (PRI_RUN mode, EC oscillator)
		9.7	10.0	mA	+125°C	VDD = 5.0V	
	All devices	9.4	12	mA	-40°C	VDD = 4.2V	FOSC = 40 MHz (PRI_RUN mode, EC oscillator)
		9.5	12	mA	+25°C		
		9.6	12	mA	+85°C		
	All devices	11.9	15	mA	-40°C	VDD = 5.0V	
		12.1	15	mA	+25°C		
		12.2	15	mA	+85°C		

Legend: Shading of rows is to assist in readability of the table.

Note 1: The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode, with all I/O pins in high-impedance state and tied to VDD or VSS and all features that add delta current disabled (such as WDT, Timer1 Oscillator, BOR, etc.).

2: The supply current is mainly a function of operating voltage, frequency and mode. Other factors, such as I/O pin loading and switching rate, oscillator type and circuit, internal code execution pattern and temperature, also have an impact on the current consumption.
The test conditions for all IDD measurements in active operation mode are:
OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to VDD;
MCLR = VDD; WDT enabled/disabled as specified.

3: For RC oscillator configurations, current through REXT is not included. The current through the resistor can be estimated by the formula Ir = VDD/2REXT (mA) with REXT in k : .

4: Standard low-cost 32 kHz crystals have an operating temperature range of -10°C to +70°C. Extended temperature crystals are available at a much higher cost.

¤ 2002-2015 Microchip Technology Inc.

22.4.2 TIMING CONDITIONS

The temperature and voltages specified in Table 22-3 apply to all timing specifications unless otherwise noted. Figure 22-5 specifies the load conditions for the timing specifications.

TABLE 22-3: TEMPERATURE AND VOLTAGE SPECIFICATIONS AC

AC CHARACTERISTICS	Standard Operating Conditions (unless otherwise stated) Operating temperature -40℃ dTA d+85℃ for industrial -40℃ dTA d+125℃ for extended Operating voltage VDD range as described in DC spec Section 22.1 and Section 22.3 . LF parts operate for industrial temperatures only.

FIGURE 22-5: LOAD CONDITIONS FOR DE VICE TIMING SPECIFICATIONS

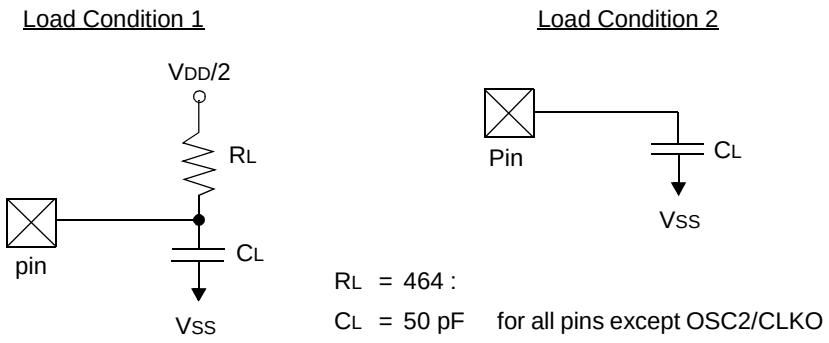


FIGURE 23-15: TYPICAL I_{PD} vs. V_{DD} (+25°C), 125 kHz TO 8 MHz RC_RUN MODE, ALL PERIPHERALS DISABLED

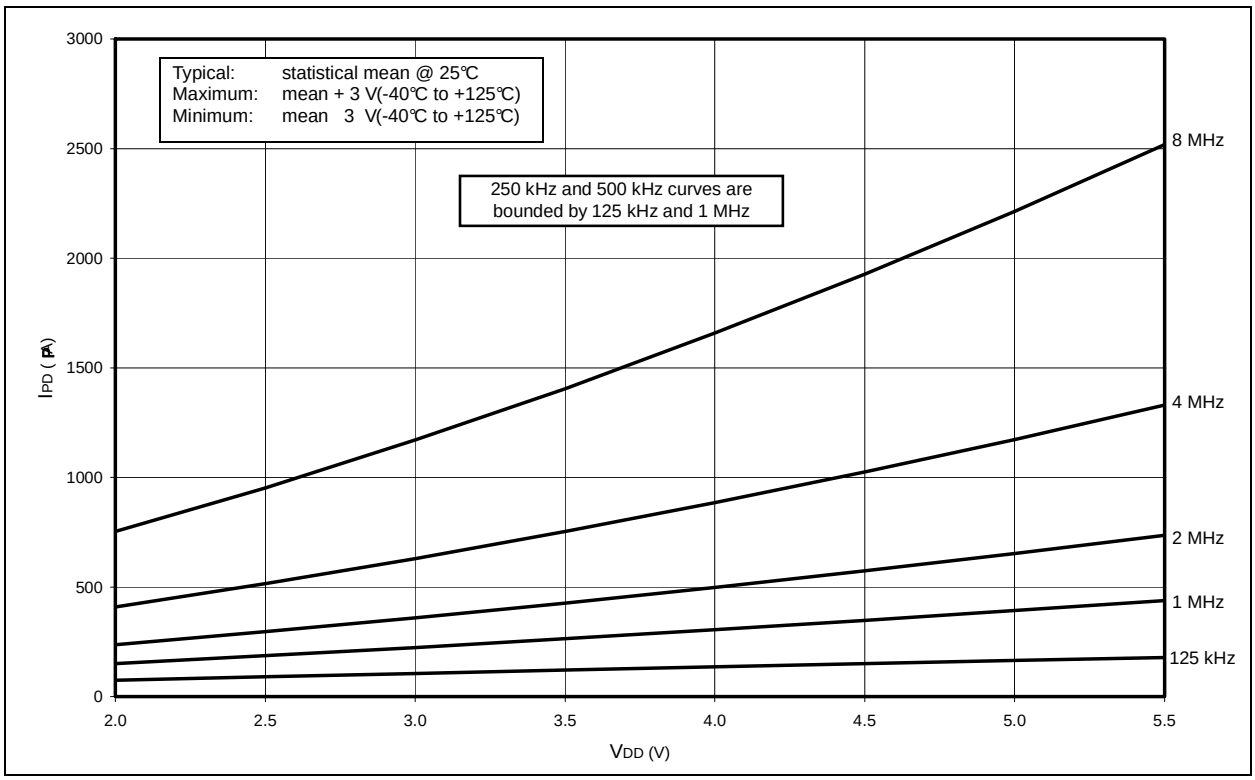


FIGURE 23-16: MAXIMUM I_{PD} vs. V_{DD} (-40°C TO +125°C), 125 kHz TO 8 MHz RC_RUN MODE, ALL PERIPHERALS DISABLED