



Welcome to [E-XFL.COM](https://www.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	40MHz
Connectivity	I <sup>2</sup> C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, HLVD, POR, PWM, WDT
Number of I/O	25
Program Memory Size	4KB (2K x 16)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	2V ~ 5.5V
Data Converters	A/D 10x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SOIC (0.295", 7.50mm Width)
Supplier Device Package	28-SOIC
Purchase URL	<a href="https://www.e-xfl.com/product-detail/microchip-technology/pic18lf2221-i-so">https://www.e-xfl.com/product-detail/microchip-technology/pic18lf2221-i-so</a>

# PIC18F2221/2321/4221/4321 FAMILY

---

## Table of Contents

1.0	Device Overview .....	9
2.0	Guidelines for Getting Started with PIC18F Microcontrollers .....	25
3.0	Oscillator Configurations .....	29
4.0	Power-Managed Modes .....	39
5.0	Reset .....	47
6.0	Memory Organization .....	59
7.0	Flash Program Memory .....	79
8.0	Data EEPROM Memory .....	89
9.0	8 x 8 Hardware Multiplier .....	95
10.0	Interrupts .....	97
11.0	I/O Ports .....	111
12.0	Timer0 Module .....	129
13.0	Timer1 Module .....	133
14.0	Timer2 Module .....	139
15.0	Timer3 Module .....	141
16.0	Capture/Compare/PWM (CCP) Modules .....	145
17.0	Enhanced Capture/Compare/PWM (ECCP) Module .....	153
18.0	Master Synchronous Serial Port (MSSP) Module .....	167
19.0	Enhanced Universal Synchronous Asynchronous Receiver Transmitter (EUSART) .....	211
20.0	10-Bit Analog-to-Digital Converter (A/D) Module .....	233
21.0	Comparator Module .....	243
22.0	Comparator Voltage Reference Module .....	249
23.0	High/Low-Voltage Detect (HLVD) .....	253
24.0	Special Features of the CPU .....	259
25.0	Instruction Set Summary .....	279
26.0	Development Support .....	329
27.0	Electrical Characteristics .....	333
28.0	Packaging Information .....	373
	Appendix A: Revision History .....	385
	Appendix B: Device Differences .....	386
	Appendix C: Conversion Considerations .....	387
	Appendix D: Migration from Baseline to Enhanced Devices .....	387
	Appendix E: Migration From Mid-Range to Enhanced Devices .....	388
	Appendix F: Migration From High-End to Enhanced Devices .....	388
	Index .....	389
	The Microchip Web Site .....	399
	Customer Change Notification Service .....	399
	Customer Support .....	399
	Reader Response .....	400
	PIC18F2221/2321/4221/4321 Product Identification System .....	401

# PIC18F2221/2321/4221/4321 FAMILY

## REGISTER 3-2: OSCCON: OSCILLATOR CONTROL REGISTER

R/W-0	R/W-1	R/W-0	R/W-0	R <sup>(1)</sup>	R-0	R/W-0	R/W-0
IDLEN	IRCF2	IRCF1	IRCF0	OSTS	IOFS	SCS1	SCS0
bit 7							bit 0

bit 7 **IDLEN:** Idle Enable bit

1 = Device enters an Idle mode when a **SLEEP** instruction is executed

0 = Device enters Sleep mode when a **SLEEP** instruction is executed

bit 6-4 **IRCF<2:0>:** Internal Oscillator Frequency Select bits

111 = 8 MHz (INTOSC drives clock directly)

110 = 4 MHz

101 = 2 MHz

100 = 1 MHz<sup>(3)</sup>

011 = 500 kHz

010 = 250 kHz

001 = 125 kHz

000 = 31 kHz (from either INTOSC/256 or INTRC directly)<sup>(2)</sup>

bit 3 **OSTS:** Oscillator Start-up Time-out Status bit<sup>(1)</sup>

1 = Oscillator Start-up Timer (OST) time-out has expired; primary oscillator is running

0 = Oscillator Start-up Timer (OST) time-out is running; primary oscillator is not ready

bit 2 **IOFS:** INTOSC Frequency Stable bit

1 = INTOSC frequency is stable

0 = INTOSC frequency is not stable

bit 1-0 **SCS<1:0>:** System Clock Select bits

1x = Internal oscillator block

01 = Secondary (Timer1) oscillator

00 = Primary oscillator

**Note 1:** Reset state depends on state of the IESO Configuration bit.

**2:** Source selected by the INTSRC bit (OSCTUNE<7>), see text.

**3:** Default output frequency of INTOSC on Reset.

### 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

# PIC18F2221/2321/4221/4321 FAMILY

## 4.4.1 PRI\_IDLE MODE

This mode is unique among the three low-power Idle modes, in that it does not disable the primary device clock. For timing sensitive applications, this allows for the fastest resumption of device operation with its more accurate primary clock source, since the clock source does not have to “warm-up” or transition from another oscillator.

PRI\_IDLE mode is entered from PRI\_RUN mode by setting the IDLEN bit and executing a `SLEEP` instruction. If the device is in another Run mode, set IDLEN first, then clear the SCS bits and execute `SLEEP`. Although the CPU is disabled, the peripherals continue to be clocked from the primary clock source specified by the FOSC<3:0> Configuration bits. The OSTS bit remains set (see Figure 4-7).

When a wake event occurs, the CPU is clocked from the primary clock source. A delay of interval T<sub>CSD</sub> (parameter 38, Table 27-10) is required between the wake event and when code execution starts. This is required to allow the CPU to become ready to execute instructions. After the wake-up, the OSTS bit remains set. The IDLEN and SCS bits are not affected by the wake-up (see Figure 4-8).

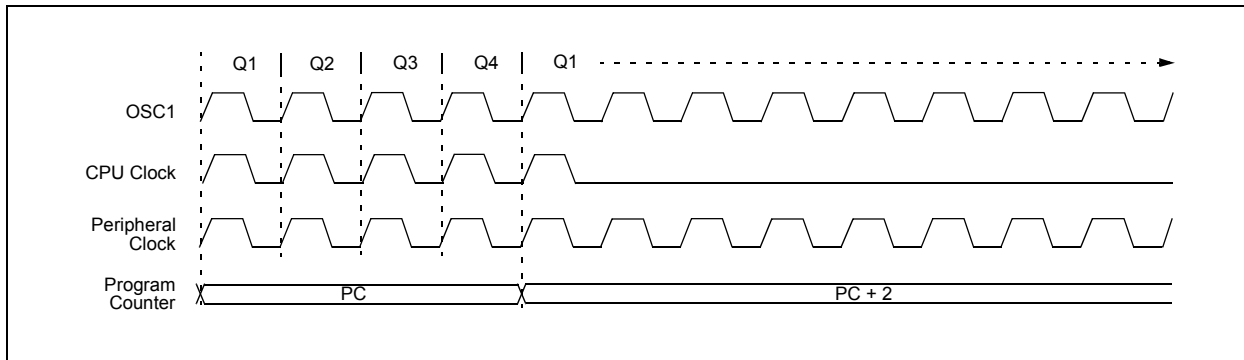
## 4.4.2 SEC\_IDLE MODE

In SEC\_IDLE mode, the CPU is disabled but the peripherals continue to be clocked from the Timer1 oscillator. This mode is entered from SEC\_RUN by setting the IDLEN bit and executing a `SLEEP` instruction. If the device is in another Run mode, set the IDLEN bit first, then set the SCS<1:0> bits to ‘01’ and execute `SLEEP`. When the clock source is switched to the Timer1 oscillator, the primary oscillator is shut down, the OSTS bit is cleared and the T1RUN bit is set.

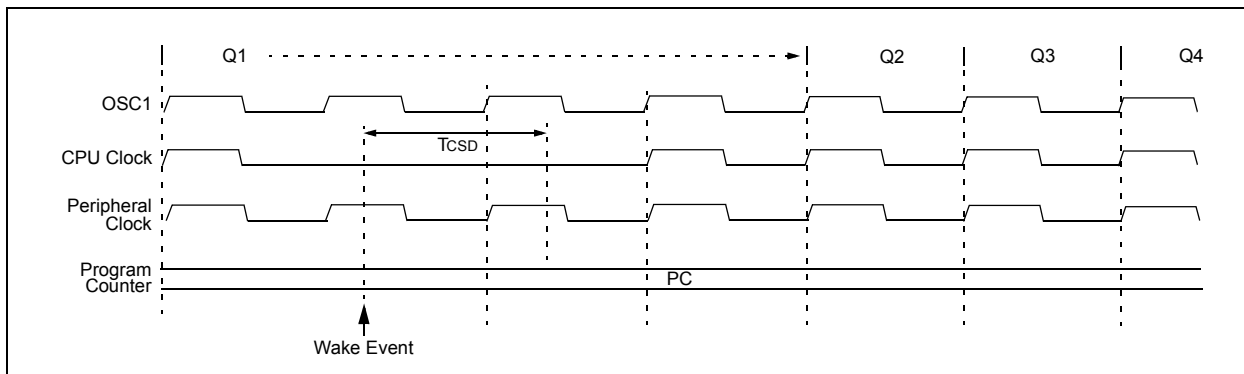
When a wake event occurs, the peripherals continue to be clocked from the Timer1 oscillator. After an interval of T<sub>CSD</sub> following the wake event, the CPU begins executing code being clocked by the Timer1 oscillator. The IDLEN and SCS bits are not affected by the wake-up; the Timer1 oscillator continues to run (see Figure 4-8).

**Note:** The Timer1 oscillator should already be running prior to entering SEC\_IDLE mode. If the T1OSCEN bit is not set when writing the SCS<1:0> bits, 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.

**FIGURE 4-7: TRANSITION TIMING FOR ENTRY TO IDLE MODE**

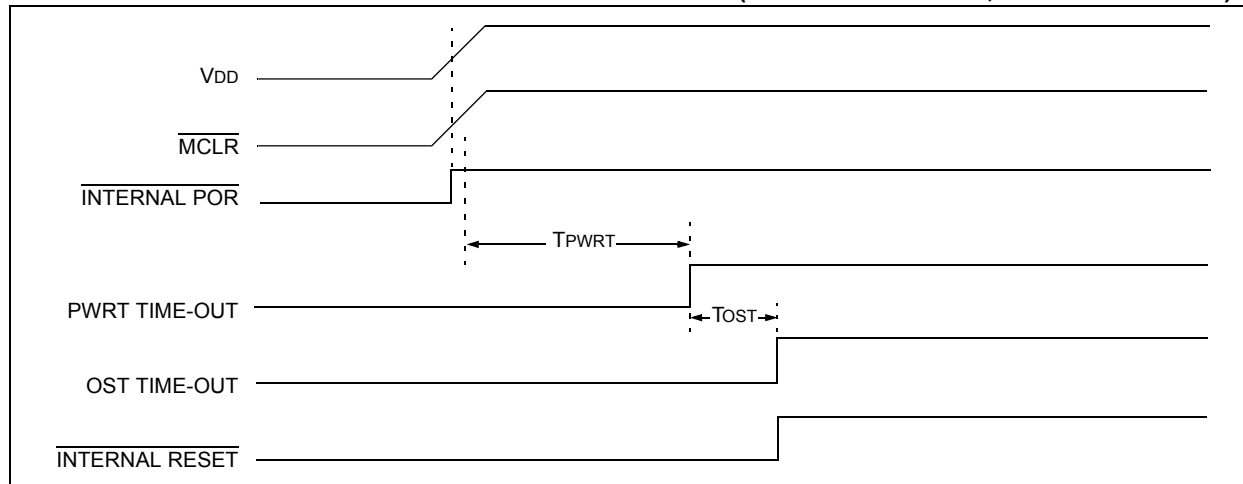


**FIGURE 4-8: TRANSITION TIMING FOR WAKE FROM IDLE TO RUN MODE**

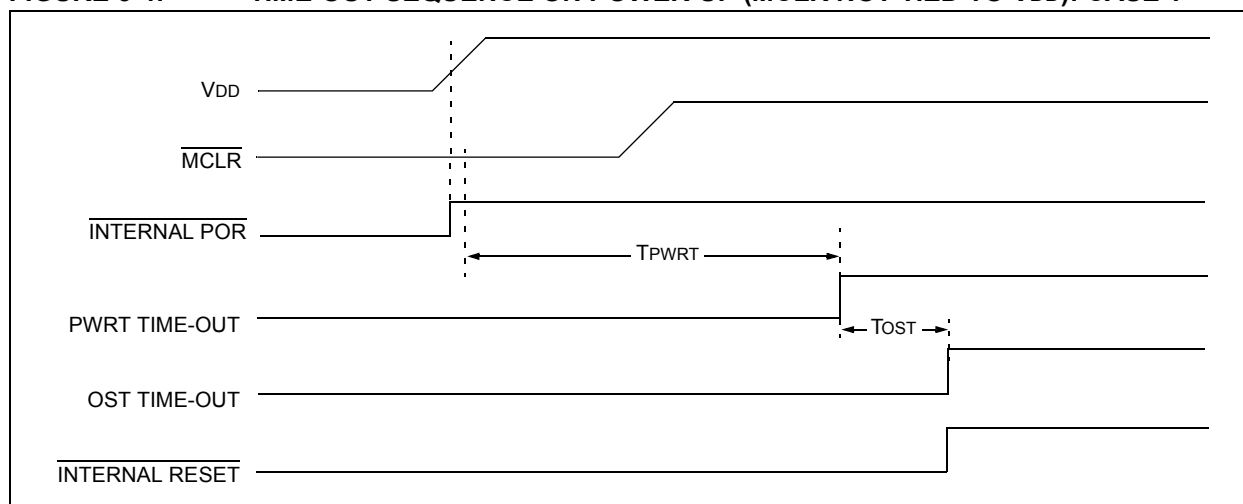


# PIC18F2221/2321/4221/4321 FAMILY

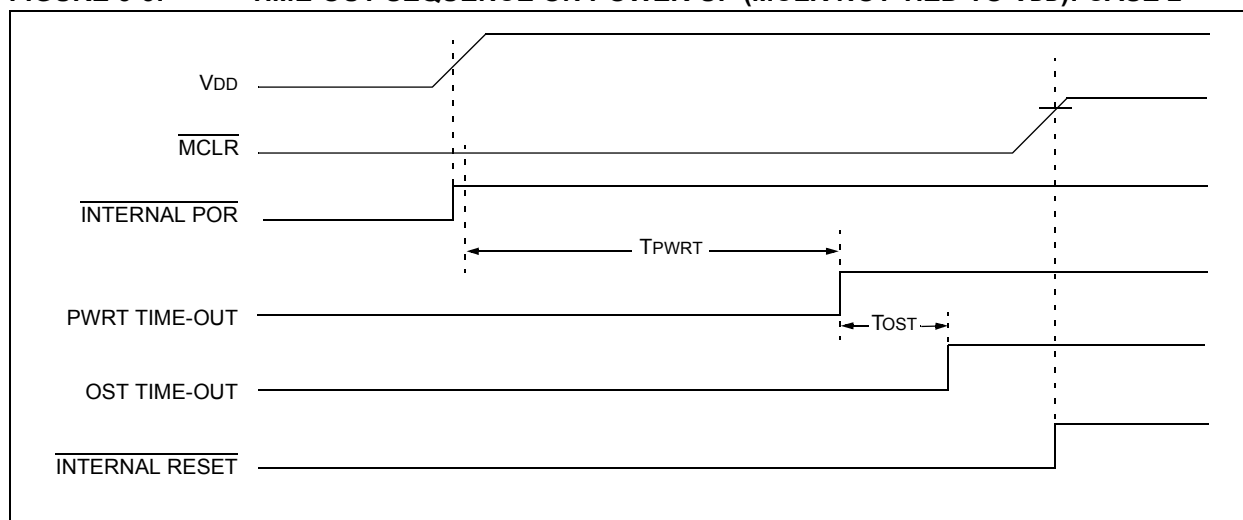
**FIGURE 5-3: TIME-OUT SEQUENCE ON POWER-UP ( $\overline{\text{MCLR}}$  TIED TO  $V_{DD}$ ,  $V_{DD}$  RISE <  $T_{PWRT}$ )**



**FIGURE 5-4: TIME-OUT SEQUENCE ON POWER-UP ( $\overline{\text{MCLR}}$  NOT TIED TO  $V_{DD}$ ): CASE 1**



**FIGURE 5-5: TIME-OUT SEQUENCE ON POWER-UP ( $\overline{\text{MCLR}}$  NOT TIED TO  $V_{DD}$ ): CASE 2**



# PIC18F2221/2321/4221/4321 FAMILY

## EXAMPLE 7-3: WRITING TO FLASH PROGRAM MEMORY (CONTINUED)

```

PROGRAM_MEMORY
    BCF     INTCON, GIE           ; disable interrupts
    MOVLW   55h                  ; required sequence
    MOVWF   EECON2               ; write 55h
    MOVLW   0AAh
    MOVWF   EECON2               ; write AAh
    BSF     EECON1, WR           ; start program (CPU stall)
    NOP
    BSF     INTCON, GIE           ; re-enable interrupts
    DECFSZ  COUNTER_HI           ; loop until done
    GOTO    PROGRAM_LOOP
    BCF     EECON1, WREN         ; disable write to memory
    
```

### 7.5.2 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.

### 7.5.3 UNEXPECTED TERMINATION OF WRITE OPERATION

If a write is terminated by an unplanned event, such as loss of power or an unexpected Reset, the memory location just programmed should be verified and reprogrammed if needed. If the write operation is interrupted by a  $\overline{\text{MCLR}}$  Reset or a WDT Time-out Reset during normal operation, the user can check the WRERR bit and rewrite the location(s) as needed.

### 7.5.4 PROTECTION AGAINST SPURIOUS WRITES

To protect against spurious writes to Flash program memory, the write initiate sequence must also be followed. See **Section 24.0 “Special Features of the CPU”** for more detail.

## 7.6 Flash Program Operation During Code Protection

See **Section 24.5 “Program Verification and Code Protection”** for details on code protection of Flash program memory.

**TABLE 7-2: REGISTERS ASSOCIATED WITH PROGRAM FLASH MEMORY**

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on page
TBLPTRU	—	—	bit 21	Program Memory Table Pointer Upper Byte (TBLPTR<20:16>)					55
TBPLTRH	Program Memory Table Pointer High Byte (TBLPTR<15:8>)								55
TBLPTRL	Program Memory Table Pointer Low Byte (TBLPTR<7:0>)								55
TABLAT	Program Memory Table Latch								55
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	55
EECON2	EEPROM Control Register 2 (not a physical register)								57
EECON1	EEPGD	CFGFS	—	FREE	WRERR	WREN	WR	RD	57
IPR2	OSCFIP	CMIP	—	EEIP	BCLIP	HLVDIP	TMR3IP	CCP2IP	58
PIR2	OSCFIF	CMIF	—	EEIF	BCLIF	HLVDIF	TMR3IF	CCP2IF	58
PIE2	OSCFIE	CMIE	—	EEIE	BCLIE	HLVDIE	TMR3IE	CCP2IE	58

**Legend:** — = unimplemented, read as ‘0’. Shaded cells are not used during Flash/EEPROM access.

# PIC18F2221/2321/4221/4321 FAMILY

## 10.0 INTERRUPTS

The PIC18F2221/2321/4221/4321 family devices have multiple interrupt sources and an interrupt priority feature that allows most interrupt sources to be assigned a high-priority level or a low-priority level. The high-priority interrupt vector is at 0008h and the low-priority interrupt vector is at 0018h. High-priority interrupt events will interrupt any low-priority interrupts that may be in progress.

There are ten registers which are used to control interrupt operation. These registers are:

- RCON
- INTCON
- INTCON2
- INTCON3
- PIR1, PIR2
- PIE1, PIE2
- IPR1, IPR2

It is recommended that the Microchip header files supplied with MPLAB® IDE be used for the symbolic bit names in these registers. This allows the assembler/compiler to automatically take care of the placement of these bits within the specified register.

In general, interrupt sources have three bits to control their operation. They are:

- **Flag bit** to indicate that an interrupt event occurred
- **Enable bit** that allows program execution to branch to the interrupt vector address when the flag bit is set
- **Priority bit** to select high priority or low priority

The interrupt priority feature is enabled by setting the IPEN bit (RCON<7>). When interrupt priority is enabled, there are two bits which enable interrupts globally. Setting the GIEH bit (INTCON<7>) enables all interrupts that have the priority bit set (high priority). Setting the GIEL bit (INTCON<6>) enables all interrupts that have the priority bit cleared (low priority). When the interrupt flag, enable bit and appropriate global interrupt enable bit are set, the interrupt will vector immediately to address 0008h or 0018h, depending on the priority bit setting. Individual interrupts can be disabled through their corresponding enable bits.

When the IPEN bit is cleared (default state), the interrupt priority feature is disabled and interrupts are compatible with PIC® mid-range devices. In Compatibility mode, the interrupt priority bits for each source have no effect. INTCON<6> is the PEIE bit, which enables/disables all peripheral interrupt sources. INTCON<7> is the GIE bit, which enables/disables all interrupt sources. All interrupts branch to address 0008h in Compatibility mode.

When an interrupt is responded to, the global interrupt enable bit is cleared to disable further interrupts. If the IPEN bit is cleared, this is the GIE bit. If interrupt priority levels are used, this will be either the GIEH or GIEL bit. High-priority interrupt sources can interrupt a low-priority interrupt. Low-priority interrupts are not processed while high-priority interrupts are in progress.

The return address is pushed onto the stack and the PC is loaded with the interrupt vector address (0008h or 0018h). Once in the Interrupt Service Routine, the source(s) of the interrupt can be determined by polling the interrupt flag bits. The interrupt flag bits must be cleared in software before re-enabling interrupts to avoid recursive interrupts.

The “return from interrupt” instruction, `RETFIE`, exits the interrupt routine and sets the GIE bit (GIEH or GIEL if priority levels are used), which re-enables interrupts.

For external interrupt events, such as the INTx pins or the PORTB input change interrupt, the interrupt latency will be three to four instruction cycles. The exact latency is the same for one or two-cycle instructions. Individual interrupt flag bits are set, regardless of the status of their corresponding enable bit or the GIE bit.

<b>Note:</b> Do not use the <code>MOVFF</code> instruction to modify any of the interrupt control registers while <b>any</b> interrupt is enabled. Doing so may cause erratic microcontroller behavior.
---

# PIC18F2221/2321/4221/4321 FAMILY

**TABLE 11-4: SUMMARY OF REGISTERS ASSOCIATED WITH PORTB**

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on page
PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	58
LATB	PORTB Data Latch Register (Read and Write to Data Latch)								58
TRISB	PORTB Data Direction Register								58
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	55
INTCON2	RBP $\bar{U}$	INTEDG0	INTEDG1	INTEDG2	—	TMR0IP	—	RBIP	55
INTCON3	INT2IP	INT1IP	—	INT2IE	INT1IE	—	INT2IF	INT1IF	55
ADCON1	—	—	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0	57

**Legend:** — = unimplemented, read as '0'. Shaded cells are not used by PORTB.



# PIC18F2221/2321/4221/4321 FAMILY

## 12.3 Prescaler

An 8-bit counter is available as a prescaler for the Timer0 module. The prescaler is not directly readable or writable; its value is set by the PSA and T0PS<2:0> bits (T0CON<3:0>) which determine the prescaler assignment and prescale ratio.

Clearing the PSA bit assigns the prescaler to the Timer0 module. When it is assigned, prescale values from 1:2 through 1:256 in power-of-2 increments are selectable.

When assigned to the Timer0 module, all instructions writing to the TMR0 register (e.g., `CLRF TMR0`, `MOVWF TMR0`, `BSF TMR0`, etc.) clear the prescaler count.

**Note:** Writing to TMR0 when the prescaler is assigned to Timer0 will clear the prescaler count but will not change the prescaler assignment.

### 12.3.1 SWITCHING PRESCALER ASSIGNMENT

The prescaler assignment is fully under software control and can be changed “on-the-fly” during program execution.

## 12.4 Timer0 Interrupt

The TMR0 interrupt is generated when the TMR0 register overflows from FFh to 00h in 8-bit mode, or from FFFFh to 0000h in 16-bit mode. This overflow sets the TMR0IF flag bit. The interrupt can be masked by clearing the TMR0IE bit (INTCON<5>). Before re-enabling the interrupt, the TMR0IF bit must be cleared in software by the Interrupt Service Routine.

Since Timer0 is shut down in Sleep mode, the TMR0 interrupt cannot awaken the processor from Sleep.

TABLE 12-1: REGISTERS ASSOCIATED WITH TIMER0

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on page
TMR0L	Timer0 Register Low Byte								56
TMR0H	Timer0 Register High Byte								56
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	55
T0CON	TMR0ON	T08BIT	T0CS	T0SE	PSA	T0PS2	T0PS1	T0PS0	56
TRISA	RA7 <sup>(1)</sup>	RA6 <sup>(1)</sup>	RA5	RA4	RA3	RA2	RA1	RA0	58

**Legend:** Shaded cells are not used by Timer0.

**Note 1:** PORTA<7:6> and their direction bits are individually configured as port pins based on various primary oscillator modes. When disabled, these bits read as ‘0’.

# PIC18F2221/2321/4221/4321 FAMILY

---

NOTES:

# PIC18F2221/2321/4221/4321 FAMILY

The CCPRxH 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 CCPRxH and 2-bit latch match TMR2, concatenated with an internal 2-bit Q clock or 2 bits of the TMR2 prescaler, the CCPx pin is cleared.

The maximum PWM resolution (bits) for a given PWM frequency is given by the equation:

## EQUATION 16-3:

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

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

**TABLE 16-4: 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

### 16.4.3 PWM AUTO-SHUTDOWN (CCP1 ONLY)

The PWM auto-shutdown features of the Enhanced CCP module are also available to CCP1 in 28-pin devices. The operation of this feature is discussed in detail in **Section 17.4.7 “Enhanced PWM Auto-Shutdown”**.

Auto-shutdown features are not available for CCP2.

### 16.4.4 SETUP FOR PWM OPERATION

The following steps should be taken when configuring the CCP module for PWM operation:

1. Set the PWM period by writing to the PR2 register.
2. Set the PWM duty cycle by writing to the CCPRxL register and CCPxCON<5:4> bits.
3. Make the CCPx pin an output by clearing the appropriate TRIS bit.
4. Set the TMR2 prescale value, then enable Timer2 by writing to T2CON.
5. Configure the CCPx module for PWM operation.

# PIC18F2221/2321/4221/4321 FAMILY

## 18.3.5 MASTER MODE

The master can initiate the data transfer at any time because it controls the SCK. The master determines when the slave (Processor 2, Figure 18-2) is to broadcast data by the software protocol.

In Master mode, the data is transmitted/received as soon as the SSPBUF register is written to. If the SPI operation is only going to receive, the SDO output could be disabled (programmed as an input). The SSPSR register will continue to shift in the signal present on the SDI pin at the programmed clock rate. As each byte is received, it will be loaded into the SSPBUF register as if a normal received byte (interrupts and status bits appropriately set). This could be useful in receiver applications as a "Line Activity Monitor" mode.

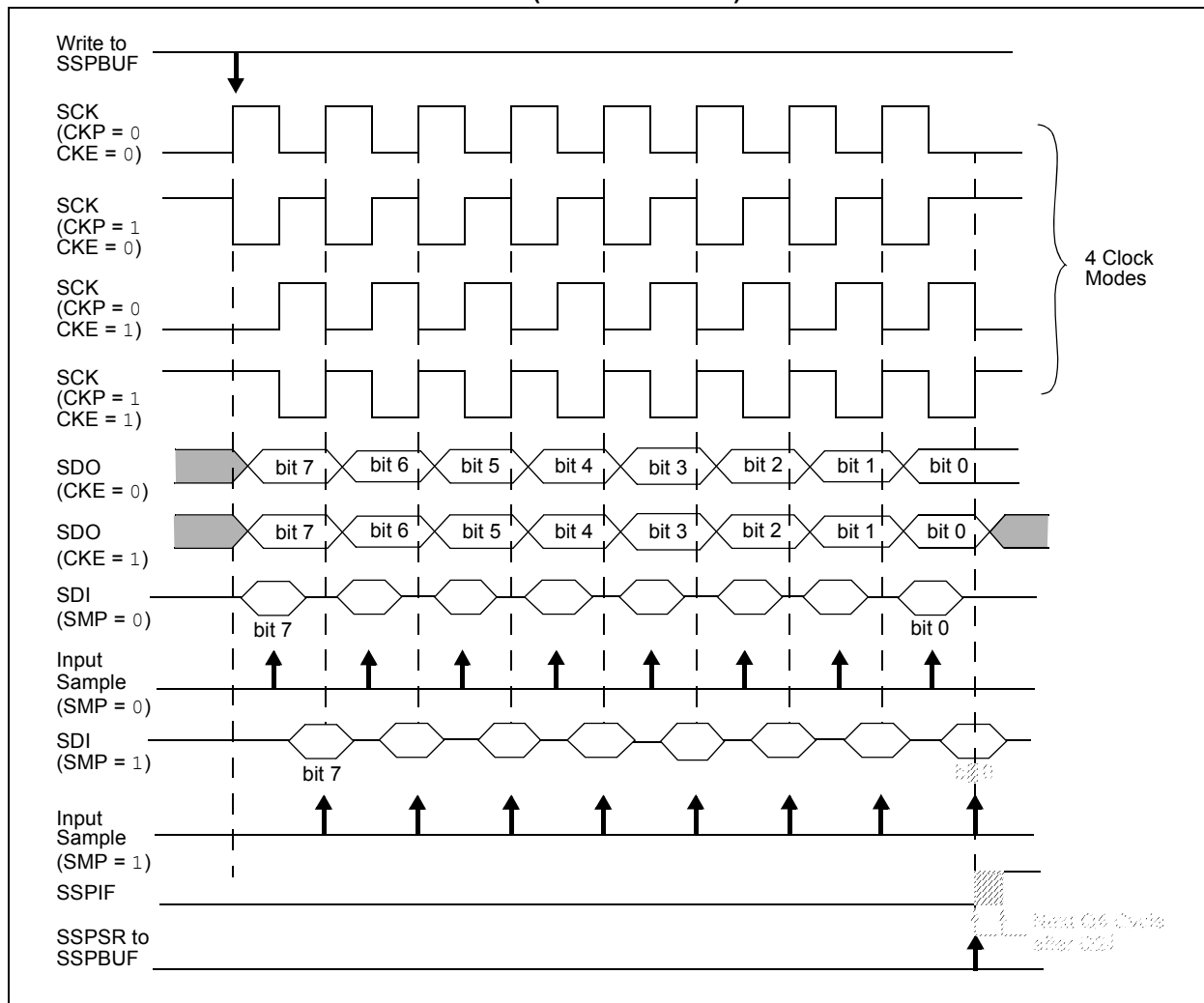
The clock polarity is selected by appropriately programming the CKP bit (SSPCON1<4>). This then, would give waveforms for SPI communication as shown in Figure 18-3, Figure 18-5 and Figure 18-6, where the MSB is transmitted first. In Master mode, the SPI clock rate (bit rate) is user-programmable to be one of the following:

- $F_{osc}/4$  (or  $T_{cy}$ )
- $F_{osc}/16$  (or  $4 \cdot T_{cy}$ )
- $F_{osc}/64$  (or  $16 \cdot T_{cy}$ )
- $\text{Timer2 output}/2$

This allows a maximum data rate (at 40 MHz) of 10.00 Mbps.

Figure 18-3 shows the waveforms for Master mode. When the CKE bit is set, the SDO data is valid before there is a clock edge on SCK. The change of the input sample is shown based on the state of the SMP bit. The time when the SSPBUF is loaded with the received data is shown.

**FIGURE 18-3: SPI MODE WAVEFORM (MASTER MODE)**



# PIC18F2221/2321/4221/4321 FAMILY

## 18.3.8 OPERATION IN POWER-MANAGED MODES

In SPI Master mode, module clocks may be operating at a different speed than when in full power mode. In the case of Sleep mode, all clocks are halted.

In Idle modes, a clock is provided to the peripherals. That clock should be from the primary clock source, the secondary clock (Timer1 oscillator at 32.768 kHz) or the INTOSC source. See **Section 3.7 “Clock Sources and Oscillator Switching”** for additional information.

In most cases, the speed that the master clocks SPI data is not important; however, this should be evaluated for each system.

If MSSP interrupts are enabled, they can wake the controller from Sleep mode, or one of the Idle modes, when the master completes sending data. If an exit from Sleep or Idle mode is not desired, MSSP interrupts should be disabled.

If the Sleep mode is selected, all module clocks are halted and the transmission/reception will remain in that state until the device wakes. After the device returns to Run mode, the module will resume transmitting and receiving data.

In SPI Slave mode, the SPI Transmit/Receive Shift register operates asynchronously to the device. This allows the device to be placed in any power-managed mode and data to be shifted into the SPI Transmit/Receive Shift register. When all 8 bits have been received, the MSSP interrupt flag bit will be set and if enabled, will wake the device.

## 18.3.9 EFFECTS OF A RESET

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

## 18.3.10 BUS MODE COMPATIBILITY

Table 18-1 shows the compatibility between the standard SPI modes and the states of the CKP and CKE control bits.

**TABLE 18-1: SPI BUS MODES**

Standard SPI Mode Terminology	Control Bits State	
	CKP	CKE
0, 0	0	1
0, 1	0	0
1, 0	1	1
1, 1	1	0

There is also an SMP bit which controls when the data is sampled.

**TABLE 18-2: REGISTERS ASSOCIATED WITH SPI OPERATION**

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on page
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	55
PIR1	PSPIF <sup>(1)</sup>	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	58
PIE1	PSPIE <sup>(1)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	58
IPR1	PSPIP <sup>(1)</sup>	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	58
TRISA	TRISA7 <sup>(2)</sup>	TRISA6 <sup>(2)</sup>	PORTA Data Direction Control Register						58
TRISC	PORTC Data Direction Control Register								58
SSPBUF	MSSP Receive Buffer/Transmit Register								56
SSPCON1	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	56
SSPSTAT	SMP	CKE	D/A	P	S	R/W	UA	BF	56

**Legend:** Shaded cells are not used by the MSSP in SPI mode.

**Note 1:** These bits are unimplemented on 28-pin devices and read as ‘0’.

**2:** PORTA<7:6> and their direction bits are individually configured as port pins based on various primary oscillator modes. When disabled, these bits read as ‘0’.

# PIC18F2221/2321/4221/4321 FAMILY

## 18.4.5 GENERAL CALL ADDRESS SUPPORT

The addressing procedure for the I<sup>2</sup>C bus is such that the first byte after the Start condition usually determines which device will be the slave addressed by the master. The exception is the general call address which can address all devices. When this address is used, all devices should, in theory, respond with an Acknowledge.

The general call address is one of eight addresses reserved for specific purposes by the I<sup>2</sup>C protocol. It consists of all '0's with R/W = 0.

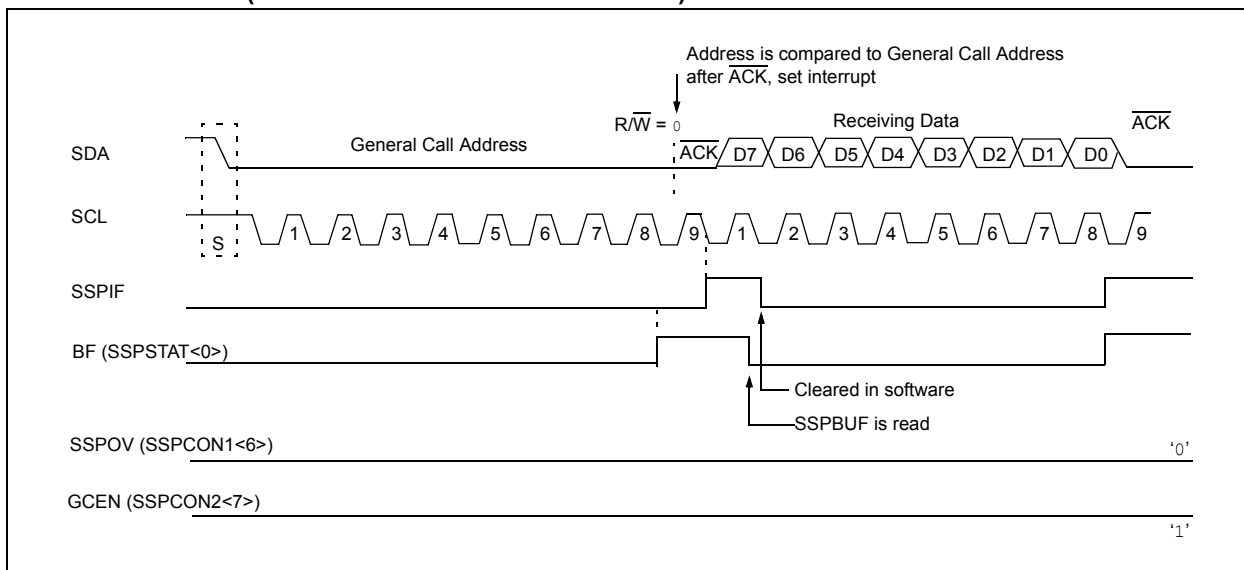
The general call address is recognized when the General Call Enable bit, GCEN, is enabled (SSPCON2<7> is set). Following a Start bit detect, 8 bits are shifted into the SSPSR and the address is compared against the SSPADD. It is also compared to the general call address and fixed in hardware.

If the general call address matches, the SSPSR is transferred to the SSPBUF, the BF flag bit is set (eighth bit) and on the falling edge of the ninth bit ( $\overline{\text{ACK}}$  bit), the SSPIF interrupt flag bit is set.

When the interrupt is serviced, the source for the interrupt can be checked by reading the contents of the SSPBUF. The value can be used to determine if the address was device specific or a general call address.

In 10-bit mode, the SSPADD is required to be updated for the second half of the address to match and the UA bit (SSPSTAT<1>) is set. If the general call address is sampled when the GCEN bit is set, while the slave is configured in 10-Bit Addressing mode, then the second half of the address is not necessary, the UA bit will not be set and the slave will begin receiving data after the Acknowledge (Figure 18-17).

**FIGURE 18-17: SLAVE MODE GENERAL CALL ADDRESS SEQUENCE (7 OR 10-BIT ADDRESSING MODE)**



# PIC18F2221/2321/4221/4321 FAMILY

## 19.1 Baud Rate Generator (BRG)

The BRG is a dedicated 8-bit or 16-bit generator that supports both the Asynchronous and Synchronous modes of the EUSART. By default, the BRG operates in 8-bit mode; setting the BRG16 bit (BAUDCON<3>) selects 16-bit mode.

The SPBRGH:SPBRG register pair controls the period of a free running timer. In Asynchronous mode, bits BRGH (TXSTA<2>) and BRG16 (BAUDCON<3>) also control the baud rate. In Synchronous mode, BRGH is ignored. Table 19-1 shows the formula for computation of the baud rate for different EUSART modes which only apply in Master mode (internally generated clock).

Given the desired baud rate and FOSC, the nearest integer value for the SPBRGH:SPBRG registers can be calculated using the formulas in Table 19-1. From this, the error in baud rate can be determined. An example calculation is shown in Example 19-1. Typical baud rates and error values for the various Asynchronous modes are shown in Table 19-2. It may be advantageous to use the high baud rate (BRGH = 1) or the 16-bit BRG to reduce the baud rate error, or achieve a slow baud rate for a fast oscillator frequency.

Writing a new value to the SPBRGH:SPBRG registers causes the BRG timer to be reset (or cleared). This ensures the BRG does not wait for a timer overflow before outputting the new baud rate.

**Note:** A BRG value of 0 is not supported.

### 19.1.1 OPERATION IN POWER-MANAGED MODES

The device clock is used to generate the desired baud rate. When one of the power-managed modes is entered, the new clock source may be operating at a different frequency. This may require an adjustment to the value in the SPBRG register pair.

### 19.1.2 SAMPLING

The data on the RX pin is sampled three times by a majority detect circuit to determine if a high or a low level is present at the RX pin when SYNC is clear or when BRG16 and BRGH are both not set. The data on the RX pin is sampled once when SYNC is set or when BRGH16 and BRGH are both set.

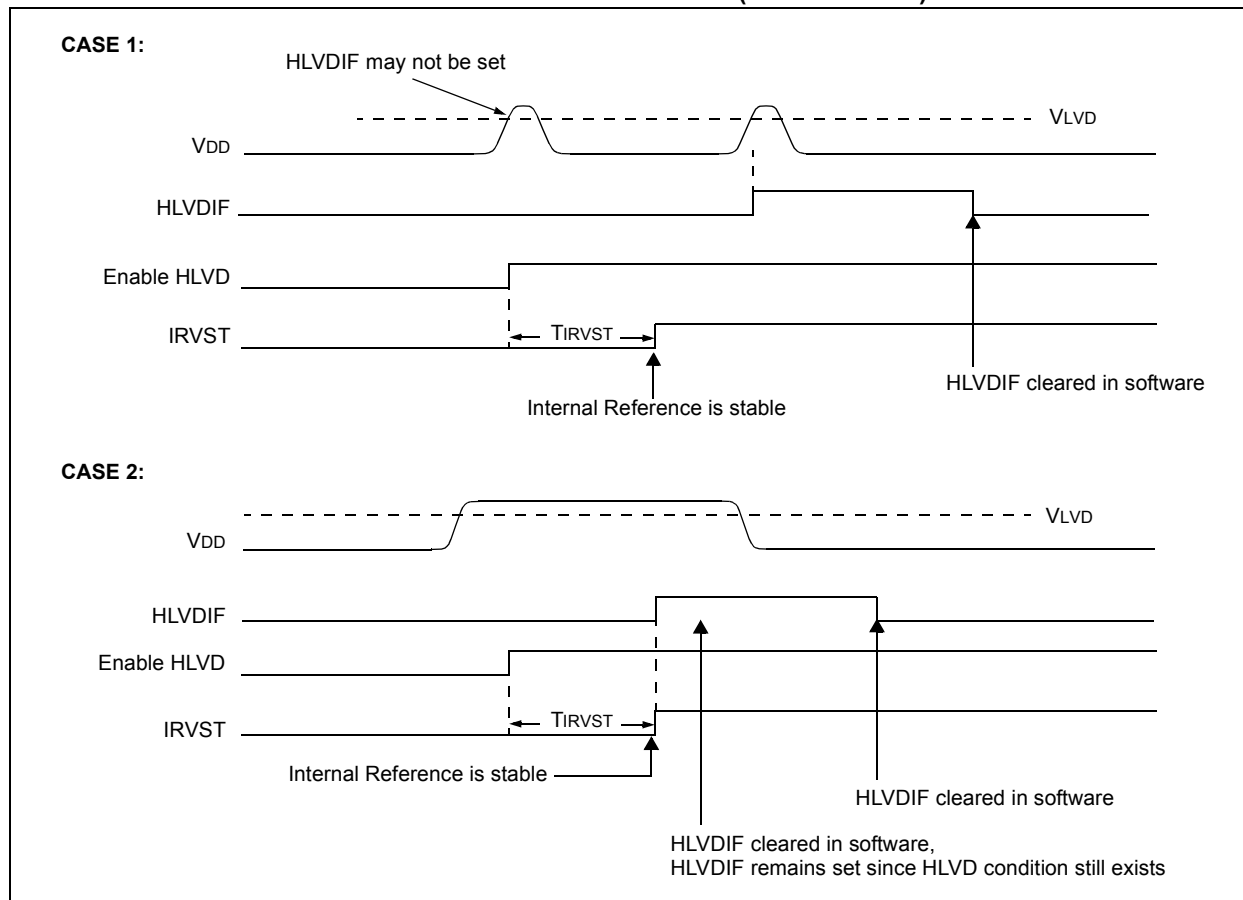
**TABLE 19-1: BAUD RATE FORMULAS**

Configuration Bits			BRG/EUSART Mode	Baud Rate Formula
SYNC	BRG16	BRGH		
0	0	0	8-bit/Asynchronous	$F_{osc}/[64 (n + 1)]$
0	0	1	8-bit/Asynchronous	$F_{osc}/[16 (n + 1)]$
0	1	0	16-bit/Asynchronous	
0	1	1	16-bit/Asynchronous	$F_{osc}/[4 (n + 1)]$
1	0	x	8-bit/Synchronous	
1	1	x	16-bit/Synchronous	

**Legend:** x = Don't care, n = value of SPBRGH:SPBRG register pair

# PIC18F2221/2321/4221/4321 FAMILY

**FIGURE 23-3: HIGH-VOLTAGE DETECT OPERATION (VDIRMAG = 1)**

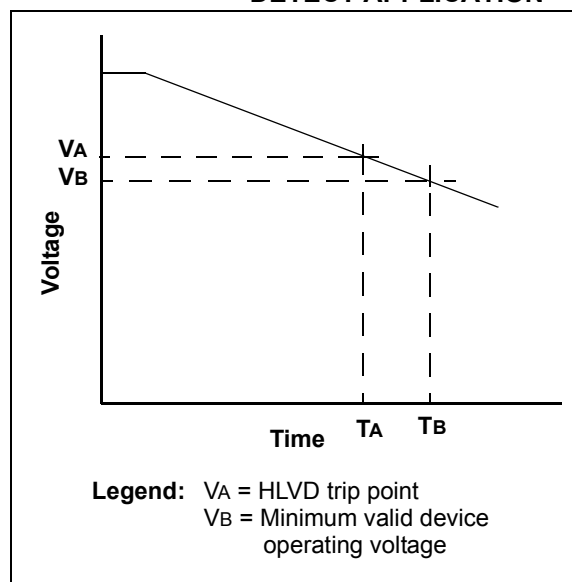


## 23.5 Applications

In many applications, the ability to detect a drop below or rise above a particular threshold is desirable. For example, the HLVD module could be periodically enabled to detect a Universal Serial Bus (USB) attach or detach. This assumes the device is powered by a lower voltage source than the USB when detached. An attach would indicate a high-voltage detect from, for example, 3.3V to 5V (the voltage on USB) and vice versa for a detach. This feature could save a design a few extra components and an attach signal (input pin).

For general battery applications, Figure 23-4 shows a possible voltage curve. Over time, the device voltage decreases. When the device voltage reaches voltage  $V_A$ , the HLVD logic generates an interrupt at time  $T_A$ . The interrupt could cause the execution of an ISR, which would allow the application to perform “house-keeping tasks” and perform a controlled shutdown before the device voltage exits the valid operating range at  $T_B$ . The HLVD, thus, would give the application a time window, represented by the difference between  $T_A$  and  $T_B$ , to safely exit.

**FIGURE 23-4: TYPICAL LOW-VOLTAGE DETECT APPLICATION**





# PIC18F2221/2321/4221/4321 FAMILY

## REGISTER 24-2: CONFIG2L: CONFIGURATION REGISTER 2 LOW (BYTE ADDRESS 300002h)

U-0	U-0	U-0	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1
—	—	—	BORV1 <sup>(1)</sup>	BORV0 <sup>(1)</sup>	BOREN1 <sup>(2)</sup>	BOREN0 <sup>(2)</sup>	PWRTEN <sup>(2)</sup>
bit 7			bit 0				

bit 7-5 **Unimplemented:** Read as '0'

bit 4-3 **BORV<1:0>:** Brown-out Reset Voltage bits<sup>(1)</sup>

11 = Minimum setting

.

.

.

00 = Maximum setting

bit 2-1 **BOREN<1:0>:** Brown-out Reset Enable bits<sup>(2)</sup>

11 = Brown-out Reset enabled in hardware only (SBOREN is disabled)

10 = Brown-out Reset enabled in hardware only and disabled in Sleep mode (SBOREN is disabled)

01 = Brown-out Reset enabled and controlled by software (SBOREN is enabled)

00 = Brown-out Reset disabled in hardware and software

bit 0 **PWRTEN:** Power-up Timer Enable bit<sup>(2)</sup>

1 = PWRT disabled

0 = PWRT enabled

**Note 1:** See **Section 27.1 “DC Characteristics”** for the specifications.

**2:** The Power-up Timer is decoupled from Brown-out Reset, allowing these features to be independently controlled.

### Legend:

R = Readable bit

P = Programmable bit

U = Unimplemented bit, read as '0'

-n = Value when device is unprogrammed

u = Unchanged from programmed state

# PIC18F2221/2321/4221/4321 FAMILY

## 24.5.2 DATA EEPROM CODE PROTECTION

The entire data EEPROM is protected from external reads and writes by two bits: CPD and WRTD. CPD inhibits external reads and writes of data EEPROM. WRTD inhibits internal and external writes to data EEPROM. The CPU can always read data EEPROM under normal operation, regardless of the protection bit settings.

## 24.5.3 CONFIGURATION REGISTER PROTECTION

The Configuration registers can be write-protected. The WRTC bit controls protection of the Configuration registers. In normal execution mode, the WRTC bit is readable only. WRTC can only be written via ICSP operation or an external programmer.

## 24.6 ID Locations

Eight memory locations (200000h-200007h) are designated as ID locations, where the user can store checksum or other code identification numbers. These locations are both readable and writable during normal execution through the `TBLRD` and `TBLWT` instructions or during program/verify. The ID locations can be read when the device is code-protected.

## 24.7 In-Circuit Serial Programming

PIC18F2221/2321/4221/4321 family microcontrollers can be serially programmed while in the end application circuit. This is simply done with two lines for clock and data and three other lines for power, ground and the programming voltage. This allows customers to manufacture boards with unprogrammed devices and then program the microcontroller just before shipping the product. This also allows the most recent firmware or a custom firmware to be programmed.

## 24.8 In-Circuit Debugger

When the `DEBUG` Configuration bit is programmed to a '0', the In-Circuit Debugger functionality is enabled. This function allows simple debugging functions when used with MPLAB® IDE. When the microcontroller has this feature enabled, some resources are not available for general use. Table 24-4 shows which resources are required by the background debugger.

**TABLE 24-4: DEBUGGER RESOURCES**

I/O Pins:	RB6, RB7
Stack:	2 levels
Program Memory:	512 bytes
Data Memory:	10 bytes

To use the In-Circuit Debugger function of the microcontroller, the design must implement In-Circuit Serial Programming connections to `MCLR/VPP/RE3`, `VDD`, `Vss`, `RB7` and `RB6`. This will interface to the In-Circuit Debugger module available from Microchip or one of the third party development tool companies.

## 24.9 Single-Supply ICSP Programming

The LVP Configuration bit enables Single-Supply ICSP Programming (formerly known as *Low-Voltage ICSP Programming* or *LVP*). When Single-Supply Programming is enabled, the microcontroller can be programmed without requiring high voltage being applied to the `MCLR/VPP/RE3` pin, but the `RB5/KBI1/PGM` pin is then dedicated to controlling Program mode entry and is not available as a general purpose I/O pin.

While programming, using Single-Supply Programming, `VDD` is applied to the `MCLR/VPP/RE3` pin as in normal execution mode. To enter Programming mode, `VDD` is applied to the `PGM` pin.

- Note 1:** High-voltage programming is always available, regardless of the state of the LVP bit or the PGM pin, by applying `VIHH` to the `MCLR` pin.

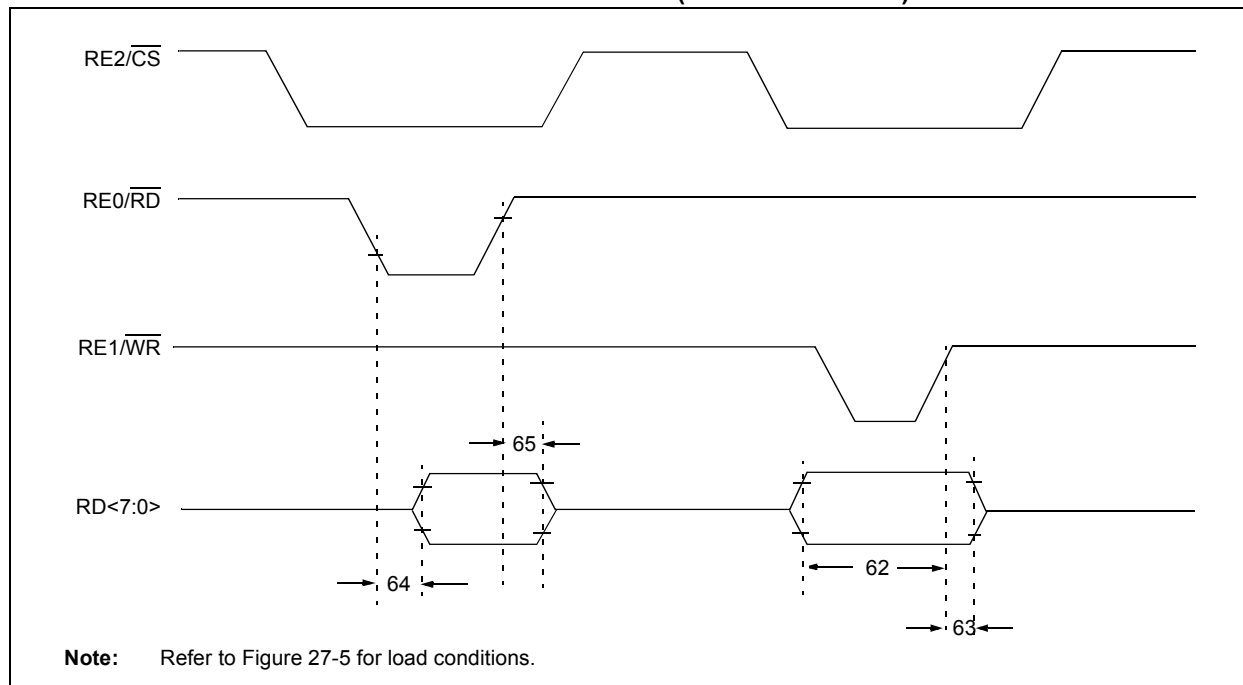
  - By default, Single-Supply ICSP Programming is enabled in unprogrammed devices (as supplied from Microchip) and erased devices.
  - When Single-Supply ICSP Programming is enabled, the `RB5` pin can no longer be used as a general purpose I/O pin.
  - When LVP is enabled, externally pull the `PGM` pin to `Vss` to allow normal program execution.

If Single-Supply ICSP Programming mode will not be used, the LVP bit can be cleared. `RB5/KBI1/PGM` then becomes available as the digital I/O pin, `RB5`. The LVP bit may be set or cleared only when using standard high-voltage programming (`VIHH` applied to the `MCLR/VPP/RE3` pin). Once LVP has been disabled, only the standard high-voltage programming is available and must be used to program the device.

Memory that is not code-protected can be erased using either a block erase, or erased row by row, then written at any specified `VDD`. If code-protected memory is to be erased, a block erase is required. If a block erase is to be performed when using Low-Voltage ICSP Programming, the device must be supplied with `VDD` of 4.5V to 5.5V.

# PIC18F2221/2321/4221/4321 FAMILY

**FIGURE 27-12: PARALLEL SLAVE PORT TIMING (PIC18F4221/4321)**



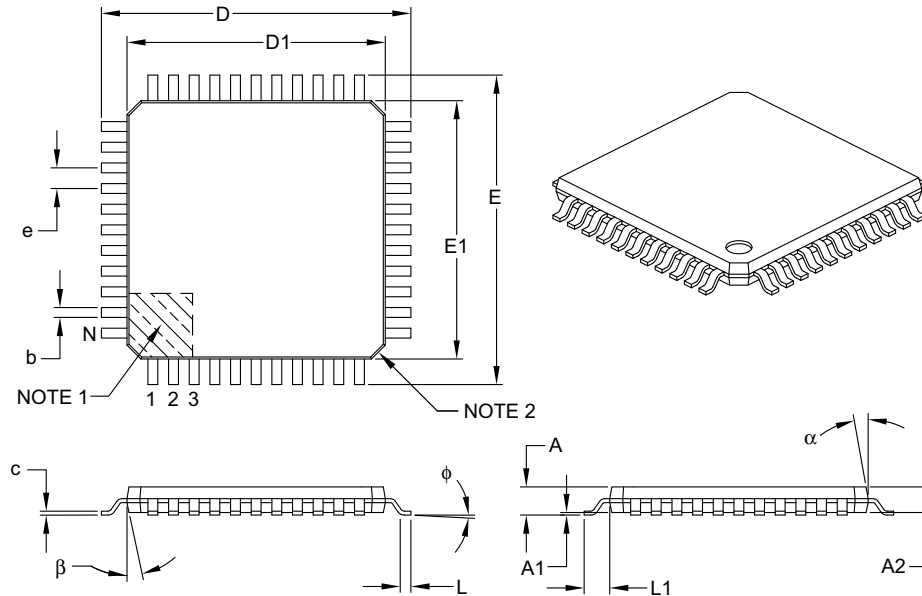
**TABLE 27-13: PARALLEL SLAVE PORT REQUIREMENTS (PIC18F4221/4321)**

Param. No.	Symbol	Characteristic	Min	Max	Units	Conditions
62	TdtV2wrH	Data In Valid before WR ↑ or CS ↑ (setup time)	20	—	ns	
63	TwrH2dtI	WR ↑ or CS ↑ to Data-In Invalid (hold time)	PIC18FXXXX	20	—	VDD = 2.0V
			PIC18LFXXXX	35	—	
64	TrdL2dtV	RD ↓ and CS ↓ to Data-Out Valid	—	80	ns	
65	TrdH2dtI	RD ↑ or CS ↓ to Data-Out Invalid	10	30	ns	
66	TibfINH	Inhibit of the IBF Flag bit being Cleared from WR ↑ or CS ↑	—	3 Tcy		

# PIC18F2221/2321/4221/4321 FAMILY

## 44-Lead Plastic Thin Quad Flatpack (PT) – 10x10x1 mm Body, 2.00 mm [TQFP]

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



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Leads	N	44		
Lead Pitch	e	0.80 BSC		
Overall Height	A	–	–	1.20
Molded Package Thickness	A2	0.95	1.00	1.05
Standoff	A1	0.05	–	0.15
Foot Length	L	0.45	0.60	0.75
Footprint	L1	1.00 REF		
Foot Angle	$\phi$	0°	3.5°	7°
Overall Width	E	12.00 BSC		
Overall Length	D	12.00 BSC		
Molded Package Width	E1	10.00 BSC		
Molded Package Length	D1	10.00 BSC		
Lead Thickness	c	0.09	–	0.20
Lead Width	b	0.30	0.37	0.45
Mold Draft Angle Top	$\alpha$	11°	12°	13°
Mold Draft Angle Bottom	$\beta$	11°	12°	13°

### Notes:

- Pin 1 visual index feature may vary, but must be located within the hatched area.
- Chamfers at corners are optional; size may vary.
- Dimensions D1 and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.25 mm per side.
- Dimensioning and tolerancing per ASME Y14.5M.

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

REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-076B

# PIC18F2221/2321/4221/4321 FAMILY

I <sup>2</sup> C Master Mode (7 or 10-Bit Transmission) .....	202	Timing Diagrams and Specifications .....	354
I <sup>2</sup> C Master Mode (7-Bit Reception) .....	203	Capture/Compare/PWM Requirements	
I <sup>2</sup> C Slave Mode (10-Bit Reception, SEN = 0,		(All CCP Modules) .....	359
ADMSK = 01001) .....	187	CLKO and I/O Requirements .....	356
I <sup>2</sup> C Slave Mode (10-Bit Reception, SEN = 0) .....	188	EUSART Synchronous Receive Requirements .....	369
I <sup>2</sup> C Slave Mode (10-Bit Reception, SEN = 1) .....	193	EUSART Synchronous Transmission Requirements .....	369
I <sup>2</sup> C Slave Mode (10-Bit Transmission) .....	189	Example SPI Mode Requirements	
I <sup>2</sup> C Slave Mode (7-Bit Reception, SEN = 0,		(Master Mode, CKE = 0) .....	361
ADMSK = 01011) .....	185	Example SPI Mode Requirements	
I <sup>2</sup> C Slave Mode (7-Bit Reception, SEN = 0) .....	184	(Master Mode, CKE = 1) .....	362
I <sup>2</sup> C Slave Mode (7-Bit Reception, SEN = 1) .....	192	Example SPI Mode Requirements	
I <sup>2</sup> C Slave Mode (7-Bit Transmission) .....	186	(Slave Mode, CKE = 0) .....	363
I <sup>2</sup> C Slave Mode General Call Address		Example SPI Mode Requirements	
Sequence (7 or 10-Bit Addressing Mode) .....	194	(Slave Mode, CKE = 1) .....	364
I <sup>2</sup> C Stop Condition Receive or Transmit Mode .....	204	External Clock Requirements .....	354
Low-Voltage Detect Operation (VDIRMAG = 0) .....	255	I <sup>2</sup> C Bus Data Requirements (Slave Mode) .....	366
Master SSP I <sup>2</sup> C Bus Data .....	367	I <sup>2</sup> C Bus Start/Stop Requirements (Slave Mode) .....	365
Master SSP I <sup>2</sup> C Bus Start/Stop Bits .....	367	Master SSP I <sup>2</sup> C Bus Data Requirements .....	368
Parallel Slave Port (PIC18F4221/4321) .....	360	Master SSP I <sup>2</sup> C Bus Start/Stop Bits	
Parallel Slave Port (PSP) Read .....	127	Requirements .....	367
Parallel Slave Port (PSP) Write .....	127	Parallel Slave Port Requirements	
PWM Auto-Shutdown (PRSEN = 0,		(PIC18F4221/4321) .....	360
Auto-Restart Disabled) .....	164	PLL Clock .....	355
PWM Auto-Shutdown (PRSEN = 1,		Reset, Watchdog Timer, Oscillator Start-up	
Auto-Restart Enabled) .....	164	Timer, Power-up Timer and	
PWM Direction Change .....	161	Brown-out Reset Requirements .....	357
PWM Direction Change at Near		Timer0 and Timer1 External Clock	
100% Duty Cycle .....	161	Requirements .....	358
PWM Output .....	150	Top-of-Stack Access .....	60
Repeated Start Condition .....	200	TRISE Register	
Reset, Watchdog Timer (WDT), Oscillator Start-up		PSPMODE Bit .....	120
Timer (OST), Power-up Timer (PWRT) .....	357	TSTFSZ .....	319
Send Break Character Sequence .....	227	Two-Speed Start-up .....	259, 271
Slave Synchronization .....	173	Two-Word Instructions	
Slow Rise Time (MCLR Tied to VDD,		Example Cases .....	64
VDD Rise > TPWRT) .....	53	TXSTA Register	
SPI Mode (Master Mode) .....	172	BRGH Bit .....	215
SPI Mode (Slave Mode, CKE = 0) .....	174	<b>V</b>	
SPI Mode (Slave Mode, CKE = 1) .....	174	Voltage Reference Specifications .....	350
Synchronous Reception (Master Mode, SREN) .....	230	<b>W</b>	
Synchronous Transmission .....	228	Watchdog Timer (WDT) .....	259, 269
Synchronous Transmission (Through TXEN) .....	229	Associated Registers .....	270
Time-out Sequence on POR w/PLL Enabled		Control Register .....	269
(MCLR Tied to VDD) .....	53	During Oscillator Failure .....	272
Time-out Sequence on Power-up		Programming Considerations .....	269
(MCLR Not Tied to VDD, Case 1) .....	52	WCOL .....	199, 200, 201, 204
Time-out Sequence on Power-up		WCOL Status Flag .....	199, 200, 201, 204
(MCLR Not Tied to VDD, Case 2) .....	52	WWW Address .....	399
Time-out Sequence on Power-up		WWW, On-Line Support .....	8
(MCLR Tied to VDD, VDD Rise < TPWRT) .....	52	<b>X</b>	
Timer0 and Timer1 External Clock .....	358	XORLW .....	319
Transition for Entry to Idle Mode .....	44	XORWF .....	320
Transition for Entry to SEC_RUN Mode .....	41		
Transition for Entry to Sleep Mode .....	43		
Transition for Two-Speed Start-up			
(INTOSC to HSPLL) .....	271		
Transition for Wake from Idle to Run Mode .....	44		
Transition for Wake from Sleep (HSPLL) .....	43		
Transition from RC_RUN Mode to PRI_RUN Mode .....	42		
Transition from SEC_RUN Mode to			
PRI_RUN Mode (HSPLL) .....	41		
Transition to RC_RUN Mode .....	42		