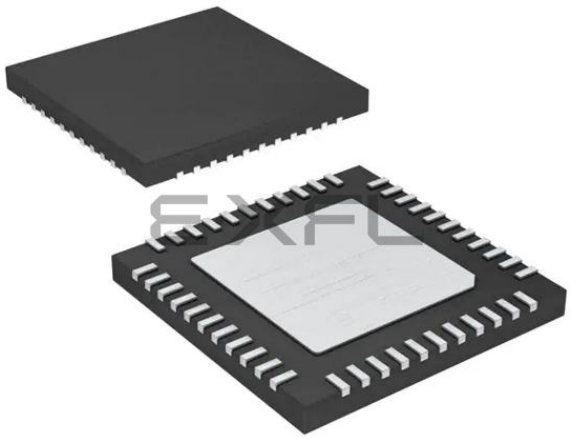


Welcome to E-XFL.COM

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	Obsolete
Core Processor	PIC
Core Size	8-Bit
Speed	40MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LVD, Power Control PWM, QEI, POR, PWM, WDT
Number of I/O	36
Program Memory Size	16KB (8K x 16)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	768 x 8
Voltage - Supply (Vcc/Vdd)	4.2V ~ 5.5V
Data Converters	A/D 9x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-VQFN Exposed Pad
Supplier Device Package	44-QFN (8x8)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f4431t-i-ml

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Trademarks

ISBN: 978-1-60932-490-2

Microchip received ISO/TS-16949:2002 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
ISO/TS 16949:2002

TABLE 1-2: PIC18F2331/2431 PINOUT I/O DESCRIPTIONS

Pin Name	Pin Number		Pin Type	Buffer Type	Description
	SPDIP, SOIC	QFN			
$\overline{MCLR}$/VPP	1	26			Master Clear (input) or programming voltage (input).
$\overline{MCLR}$			I	ST	Master Clear (Reset) input. This pin is an active-low Reset to the device.
VPP			P		High-voltage ICSP™ programming enable pin.
OSC1/CLKI/RA7	9	6			Oscillator crystal or external clock input.
OSC1			I	ST	Oscillator crystal input or external clock source input. ST buffer when configured in RC mode; CMOS otherwise.
CLKI			I	CMOS	External clock source input. Always associated with pin function OSC1. (See related OSC1/CLKI, OSC2/CLKO pins.)
RA7			I/O	TTL	General purpose I/O pin.
OSC2/CLKO/RA6	10	7			Oscillator crystal or clock output.
OSC2			O	—	Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode.
CLKO			O	—	In RC mode, OSC2 pin outputs CLKO, which has 1/4 the frequency of OSC1 and denotes the instruction cycle rate.
RA6			I/O	TTL	General purpose I/O pin.
					PORTA is a bidirectional I/O port.
RA0/AN0	2	27			
RA0			I/O	TTL	Digital I/O.
AN0			I	Analog	Analog Input 0.
RA1/AN1	3	28			
RA1			I/O	TTL	Digital I/O.
AN1			I	Analog	Analog Input 1.
RA2/AN2/VREF-/CAP1/INDX	4	1			
RA2			I/O	TTL	Digital I/O.
AN2			I	Analog	Analog Input 2.
VREF-			I	Analog	A/D reference voltage (low) input.
CAP1			I	ST	Input Capture Pin 1.
INDX			I	ST	Quadrature Encoder Interface index input pin.
RA3/AN3/VREF+/CAP2/QEA	5	2			
RA3			I/O	TTL	Digital I/O.
AN3			I	Analog	Analog Input 3.
VREF+			I	Analog	A/D reference voltage (high) input.
CAP2			I	ST	Input Capture Pin 2.
QEA			I	ST	Quadrature Encoder Interface Channel A input pin.
RA4/AN4/CAP3/QEB	6	3			
RA4			I/O	TTL	Digital I/O.
AN4			I	Analog	Analog Input 4.
CAP3			I	ST	Input Capture Pin 3.
QEB			I	ST	Quadrature Encoder Interface Channel B input pin.

Legend: TTL = TTL compatible input; CMOS = CMOS compatible input or output; ST = Schmitt Trigger input with CMOS levels; I = Input; O = Output; P = Power

NOTES:

TABLE 6-2: REGISTER FILE SUMMARY (PIC18F2331/2431/4331/4431)

File Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR
TOSU	—	—	—	Top-of-Stack Upper Byte (TOS<20:16>)					---0 0000
TOSH	Top-of-Stack High Byte (TOS<15:8>)								0000 0000
TOSL	Top-of-Stack Low Byte (TOS<7:0>)								0000 0000
STKPTR	STKFUL	STKUNF	—	SP4	SP3	SP2	SP1	SP0	00-0 0000
PCLATU	—	—	bit 21(3)	Holding Register for PC<20:16>					---0 0000
PCLATH	Holding Register for PC<15:8>								0000 0000
PCL	PC Low Byte (PC<7:0>)								0000 0000
TBLPTRU	—	—	bit 21(3)	Program Memory Table Pointer Upper Byte (TBLPTR<20:16>)					--00 0000
TBLPTRH	Program Memory Table Pointer High Byte (TBLPTR<15:8>)								0000 0000
TBLPTRL	Program Memory Table Pointer Low Byte (TBLPTR<7:0>)								0000 0000
TABLAT	Program Memory Table Latch								0000 0000
PRODH	Product Register High Byte								xxxx xxxx
PRODL	Product Register Low Byte								xxxx xxxx
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x
INTCON2	$\overline{\text{RBPU}}$	INTEDG0	INTEDG1	INTEDG2	—	TMR0IP	—	RBIP	1111 -1-1
INTCON3	INT2IP	INT1IP	—	INT2IE	INT1IE	—	INT2IF	INT1IF	11-0 0-00
INDF0	Uses contents of FSR0 to address data memory – value of FSR0 not changed (not a physical register)								N/A
POSTINC0	Uses contents of FSR0 to address data memory – value of FSR0 post-incremented (not a physical register)								N/A
POSTDEC0	Uses contents of FSR0 to address data memory – value of FSR0 post-decremented (not a physical register)								N/A
PREINC0	Uses contents of FSR0 to address data memory – value of FSR0 pre-incremented (not a physical register)								N/A
PLUSW0	Uses contents of FSR0 to address data memory – value of FSR0 offset by W (not a physical register)								N/A
FSR0H	—	—	—	—	Indirect Data Memory Address Pointer 0 High				---- xxxx
FSR0L	Indirect Data Memory Address Pointer 0 Low Byte								xxxx xxxx
WREG	Working Register								xxxx xxxx
INDF1	Uses contents of FSR1 to address data memory – value of FSR1 not changed (not a physical register)								N/A
POSTINC1	Uses contents of FSR1 to address data memory – value of FSR1 post-incremented (not a physical register)								N/A
POSTDEC1	Uses contents of FSR1 to address data memory – value of FSR1 post-decremented (not a physical register)								N/A
PREINC1	Uses contents of FSR1 to address data memory – value of FSR1 pre-incremented (not a physical register)								N/A
PLUSW1	Uses contents of FSR1 to address data memory – value of FSR1 offset by W (not a physical register)								N/A
FSR1H	—	—	—	—	Indirect Data Memory Address Pointer 1 High Byte				---- 0000
FSR1L	Indirect Data Memory Address Pointer 1 Low Byte								xxxx xxxx
BSR	—	—	—	—	Bank Select Register				---- 0000
INDF2	Uses contents of FSR2 to address data memory – value of FSR2 not changed (not a physical register)								N/A
POSTINC2	Uses contents of FSR2 to address data memory – value of FSR2 post-incremented (not a physical register)								N/A
POSTDEC2	Uses contents of FSR2 to address data memory – value of FSR2 post-decremented (not a physical register)								N/A
PREINC2	Uses contents of FSR2 to address data memory – value of FSR2 pre-incremented (not a physical register)								N/A
PLUSW2	Uses contents of FSR2 to address data memory – value of FSR2 offset by W (not a physical register)								N/A
FSR2H	—	—	—	—	Indirect Data Memory Address Pointer 2 High Byte				---- 0000
FSR2L	Indirect Data Memory Address Pointer 2 Low Byte								xxxx xxxx
STATUS	—	—	—	N	OV	Z	DC	C	---x xxxx
TMR0H	Timer0 Register High Byte								0000 0000
TMR0L	Timer0 Register Low Byte								xxxx xxxx
T0CON	TMR0ON	T016BIT	T0CS	T0SE	PSA	T0PS2	T0PS1	T0PS0	1111 1111

Legend: x = unknown, u = unchanged, - = unimplemented, q = value depends on condition. Shaded cells are unimplemented.

Note 1: RA6 and associated bits are configured as port pins in RCIO, ECIO and INTIO2 (with port function on RA6) Oscillator modes only and read '0' in all other oscillator modes.

2: RA7 and associated bits are configured as port pins in INTIO2 Oscillator mode only and read '0' in all other modes.

3: Bit 21 of the PC is only available in Test mode and Serial Programming modes.

4: These registers and/or bits are not implemented on the PIC18F2331/2431 devices and read as '0'.

5: The RE3 port bit is only available for PIC18F4331/4431 devices when the MCLRE fuse (CONFIG3H<7>) is programmed to '0'; otherwise, RE3 reads '0'. This bit is read-only.

Example 9-3 shows the sequence to do a 16 x 16 unsigned multiply. Equation 9-1 shows the algorithm that is used. The 32-bit result is stored in four registers, RES<3:0>.

EQUATION 9-1: 16 x 16 UNSIGNED MULTIPLICATION ALGORITHM

$$
\begin{aligned}
RES<3:0> &= ARG1H:ARG1L \bullet ARG2H:ARG2L \\
&= (ARG1H \bullet ARG2H \bullet 2^{16}) + \\
&\quad (ARG1H \bullet ARG2L \bullet 2^{8}) + \\
&\quad (ARG1L \bullet ARG2H \bullet 2^{8}) + \\
&\quad (ARG1L \bullet ARG2L)
\end{aligned}
$$

EXAMPLE 9-3: 16 x 16 UNSIGNED MULTIPLY ROUTINE

```
   MOVF   ARG1L, W
   MULWF  ARG2L         ; ARG1L * ARG2L ->
                        ; PRODH:PRODL
   MOVFF  PRODH, RES1   ;
   MOVFF  PRODL, RES0   ;
;
   MOVF   ARG1H, W
   MULWF  ARG2H         ; ARG1H * ARG2H ->
                        ; PRODH:PRODL
   MOVFF  PRODH, RES3   ;
   MOVFF  PRODL, RES2   ;
;
   MOVF   ARG1L, W
   MULWF  ARG2H         ; ARG1L * ARG2H ->
                        ; PRODH:PRODL
   MOVF   PRODL, W      ;
   ADDWF  RES1, F       ; Add cross
   MOVF   PRODH, W      ; products
   ADDWFC RES2, F       ;
   CLRF   WREG          ;
   ADDWFC RES3, F       ;
;
   MOVF   ARG1H, W      ;
   MULWF  ARG2L         ; ARG1H * ARG2L ->
                        ; PRODH:PRODL
   MOVF   PRODL, W      ;
   ADDWF  RES1, F       ; Add cross
   MOVF   PRODH, W      ; products
   ADDWFC RES2, F       ;
   CLRF   WREG          ;
   ADDWFC RES3, F       ;
```

Example 9-4 shows the sequence to do a 16 x 16 signed multiply. Equation 9-2 shows the algorithm used. The 32-bit result is stored in four registers, RES<3:0>. To account for the sign bits of the arguments, each argument pair's Most Significant bit (MSb) is tested and the appropriate subtractions are done.

EQUATION 9-2: 16 x 16 SIGNED MULTIPLICATION ALGORITHM

$$
\begin{aligned}
RES<3:0> &= ARG1H:ARG1L \bullet ARG2H:ARG2L \\
&= (ARG1H \bullet ARG2H \bullet 2^{16}) + \\
&\quad (ARG1H \bullet ARG2L \bullet 2^{8}) + \\
&\quad (ARG1L \bullet ARG2H \bullet 2^{8}) + \\
&\quad (ARG1L \bullet ARG2L) + \\
&\quad (-1 \bullet ARG2H<7> \bullet ARG1H:ARG1L \bullet 2^{16}) + \\
&\quad (-1 \bullet ARG1H<7> \bullet ARG2H:ARG2L \bullet 2^{16})
\end{aligned}
$$

EXAMPLE 9-4: 16 x 16 SIGNED MULTIPLY ROUTINE

```
    MOVF   ARG1L, W
    MULWF  ARG2L         ; ARG1L * ARG2L ->
                         ; PRODH:PRODL
    MOVFF  PRODH, RES1   ;
    MOVFF  PRODL, RES0   ;
;
    MOVF   ARG1H, W
    MULWF  ARG2H         ; ARG1H * ARG2H ->
                         ; PRODH:PRODL
    MOVFF  PRODH, RES3   ;
    MOVFF  PRODL, RES2   ;
;
    MOVF   ARG1L,W
    MULWF  ARG2H         ; ARG1L * ARG2H ->
                         ; PRODH:PRODL
    MOVF   PRODL, W      ;
    ADDWF  RES1, F       ; Add cross
    MOVF   PRODH, W      ; products
    ADDWFC RES2, F       ;
    CLRF   WREG          ;
    ADDWFC RES3, F       ;
;
    MOVF   ARG1H, W      ;
    MULWF  ARG2L         ; ARG1H * ARG2L ->
                         ; PRODH:PRODL
    MOVF   PRODL, W      ;
    ADDWF  RES1, F       ; Add cross
    MOVF   PRODH, W      ; products
    ADDWFC RES2, F       ;
    CLRF   WREG          ;
    ADDWFC RES3, F       ;
;
    BTFSS  ARG2H, 7      ; ARG2H:ARG2L neg?
    BRA    SIGN_ARG1     ; no, check ARG1
    MOVF   ARG1L, W      ;
    SUBWF  RES2          ;
    MOVF   ARG1H, W      ;
    SUBWFB RES3
;
SIGN_ARG1
    BTFSS  ARG1H, 7      ; ARG1H:ARG1L neg?
    BRA    CONT_CODE     ; no, done
    MOVF   ARG2L, W      ;
    SUBWF  RES2          ;
    MOVF   ARG2H, W      ;
    SUBWFB RES3
;
CONT_CODE
    :
```

REGISTER 10-2: INTCON2: 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

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

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 enable bit. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt. This feature allows for software polling.

TABLE 11-7: PORTD I/O SUMMARY

Pin	Function	TRIS Setting	I/O	I/O Type	Description
RD0/T0CKI/ T5CKI	RD0	0	O	DIG	LATD<0> data output.
		1	I	ST	PORTD<0> data input.
	T0CKI(1)	1	I	ST	Timer0 alternate clock input.
	T5CKI(1)	1	I	ST	Timer5 alternate clock input.
RD1/SDO	RD1	0	O	DIG	LATD<1> data output.
		1	I	ST	PORTD<1> data input.
	SDO(1)	0	O	DIG	SPI data out; takes priority over port data.
RD2/SDI/SDA	RD2	0	O	DIG	LATD<2> data output.
		1	I	ST	PORTD<2> data input.
	SDI(1)	1	I	ST	SPI data input (SSP module).
	SDA(1)	0	O	DIG	I^2C™ data output (SSP module); takes priority over port data.
		1	I	I^2C	I^2C data input (SSP module).
RD3/SCK/SCL	RD3	0	O	DIG	LATD<3> data output.
		1	I	ST	PORTD<3> data input.
	SCK(1)	0	O	DIG	SPI clock output (SSP module); takes priority over port data.
		1	I	ST	SPI clock input (SSP module).
	SCL(1)	0	O	DIG	I^2C clock output (SSP module); takes priority over port data.
		1	I	I^2C	I^2C clock input (SSP module); input type depends on module setting.
RD4/$\overline{FLTA}$	RD4	0	O	DIG	LATD<4> data output.
		1	I	ST	PORTD<4> data input.
	$\overline{FLTA}$(2)	1	I	ST	Fault Interrupt Input Pin A.
RD5/PWM4	RD5	0	O	DIG	LATD<5> data output.
		1	I	ST	PORTD<5> data input.
	PWM4(3)	0	O	DIG	PWM Output 4; takes priority over port data.
RD6/PWM6	RD6	0	O	DIG	LATD<6> data output.
		1	I	ST	PORTD<6> data input.
	PWM6	0	O	DIG	PWM Output 6; takes priority over port data.
RD7/PWM7	RD7	0	O	DIG	LATD<7> data output.
		1	I	ST	PORTD<7> data input.
	PWM7	0	O	DIG	PWM Output 7; takes priority over port data.

Legend: DIG = Digital level output; TTL = TTL input buffer; ST = Schmitt Trigger input buffer; x = Don't care (TRIS bit does not affect port direction or is overridden for this option).

Note 1: RC3 is the alternate pin for T0CKI/T5CKI; RC4 is the alternate pin for SDI/SDA; RC5 is the alternate pin for SCK/SCL; RC7 is the alternate pin for SDO.

2: RC1 is the alternate pin for $\overline{FLTA}$.

3: RB5 is the alternate pin for PWM4.

TABLE 11-8: SUMMARY OF REGISTERS ASSOCIATED WITH PORTD

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page:
PORTD	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	57
LATD	LATD Data Output Register								57
TRISD	PORTD Data Direction Register								57

TABLE 15-1: REGISTERS ASSOCIATED WITH TIMER5

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	54
IPR3	—	—	—	PTIP	IC3DRIP	IC2QEIP	IC1IP	TMR5IP	56
PIE3	—	—	—	PTIE	IC3DRIE	IC2QEIE	IC1IE	TMR5IE	56
PIR3	—	—	—	PTIF	IC3DRIF	IC2QEIF	IC1IF	TMR5IF	56
TMR5H	Timer5 Register High Byte								57
TMR5L	TImer5 Register Low Byte								57
PR5H	Timer5 Period Register High Byte								57
PR5L	Timer5 Period Register Low Byte								57
T5CON	T5SEN	$\overline{\text{RESEN}}$	T5MOD	T5PS1	T5PS0	$\overline{\text{T5SYNC}}$	TMR5CS	TMR5ON	56
CAP1CON	—	CAP1REN	—	—	CAP1M3	CAP1M2	CAP1M1	CAP1M0	59
DFLTCON	—	FLT4EN	FLT3EN	FLT2EN	FLT1EN	FLTCK2	FLTCK1	FLTCK0	59

Legend: — = unimplemented. Shaded cells are not used by the Timer5 module.

TABLE 16-2: REGISTERS ASSOCIATED WITH CAPTURE, COMPARE AND TIMER1

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	54
PIR1	—	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	57
PIE1	—	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	57
IPR1	—	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	57
TRISC	PORTC Data Direction Register								57
TMR1L	Timer1 Register Low Byte								55
TMR1H	Timer1 Register High Byte								55
T1CON	RD16	T1RUN	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	TMR1CS	TMR1ON	55
CCPR1L	Capture/Compare/PWM Register 1 Low Byte								56
CCPR1H	Capture/Compare/PWM Register 1 High Byte								56
CCP1CON	—	—	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	56
CCPR2L	Capture/Compare/PWM Register 2 Low Byte								56
CCPR2H	Capture/Compare/PWM Register 2 High Byte								56
CCP2CON	—	—	DC2B1	DC2B0	CCP2M3	CCP2M2	CCP2M1	CCP2M0	56
PIR2	OSCFIF	—	—	EEIF	—	LVDIF	—	CCP2IF	57
PIE2	OSCFIE	—	—	EEIE	—	LVDIE	—	CCP2IE	57
IPR2	OSCFIP	—	—	EEIP	—	LVDIP	—	CCP2IP	57

Legend: — = unimplemented, read as '0'. Shaded cells are not used by Capture, Compare and Timer1.

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 following equation:

EQUATION 16-3:

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

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

16.5.3 SETUP FOR PWM OPERATION

The following steps should be taken when configuring the CCP1 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 CCPR1L register and CCP1CON<5:4> bits.
3. Make the CCP1 pin an output by clearing the TRISC<2> bit.
4. Set the TMR2 prescale value and enable Timer2 by writing to T2CON.
5. Configure the CCP1 module for PWM operation.

TABLE 16-3: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS AT 40 MHz

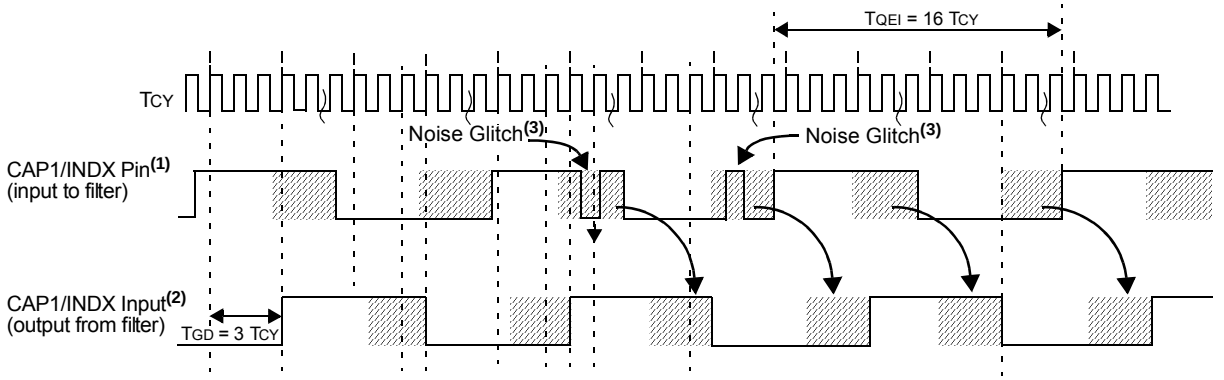
PWM Frequency	2.44 kHz	9.77 kHz	39.06 kHz	156.25 kHz	312.50 kHz	416.67 kHz
Timer Prescaler (1, 4, 16)	16	4	1	1	1	1
PR2 Value	FFh	FFh	FFh	3Fh	1Fh	17h
Maximum Resolution (bits)	10	10	10	8	7	6.58

TABLE 16-4: REGISTERS ASSOCIATED WITH PWM AND TIMER2

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	54
PIR1	—	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	57
PIE1	—	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	57
IPR1	—	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	57
TRISC	PORTC Data Direction Register								57
TMR2	Timer2 Register								55
PR2	Timer2 Period Register								55
T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	55
CCPR1L	Capture/Compare/PWM Register 1 Low Byte								56
CCPR1H	Capture/Compare/PWM Register 1 High Byte								56
CCP1CON	—	—	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	56
CCPR2L	Capture/Compare/PWM Register 2 Low Byte								56
CCPR2H	Capture/Compare/PWM Register 2 High Byte								56
CCP2CON	—	—	DC2B1	DC2B0	CCP2M3	CCP2M2	CCP2M1	CCP2M0	56

Legend: x = unknown, u = unchanged, - = unimplemented, read as '0'. Shaded cells are not used by PWM and Timer2.

FIGURE 17-14: NOISE FILTER TIMING DIAGRAM (CLOCK DIVIDER = 1:1)



Note 1: Only the CAP1/INDX pin input is shown for simplicity. Similar event timing occurs on the CAP2/QEA and CAP3/QEB pins.

2: Noise filtering occurs in the shaded portions of the CAP1 input.

3: Filter's group delay: TGD = 3 TCY.

17.4 IC and QEI Shared Interrupts

The IC and QEI submodules can each generate three distinct interrupt signals; however, they share the use of the same three interrupt flags in register, PIR3. The meaning of a particular interrupt flag at any given time depends on which module is active at the time the interrupt is set. The meaning of the flags in context are summarized in Table 17-7.

When the IC submodule is active, the three flags (IC1IF, IC2QEIF and IC3DRIF) function as interrupt-on-capture event flags for their respective input capture channels. The channel must be configured for one of the events that will generate an interrupt (see **Section 17.1.7 "IC Interrupts"** for more information).

When the QEI is enabled, the IC1IF interrupt flag indicates an interrupt caused by a velocity measurement event, usually an update of the VELR register. The IC2QEIF interrupt indicates that a position measurement event has occurred. IC3DRIF indicates that a direction change has been detected.

TABLE 17-7: MEANING OF IC AND QEI INTERRUPT FLAGS

Interrupt Flag	Meaning	
	IC Mode	QEI Mode
IC1IF	IC1 Capture Event	Velocity Register Update
IC2QEIF	IC2 Capture Event	Position Measurement Update
IC3DRIF	IC3 Capture Event	Direction Change

17.5 Operation in Sleep Mode

17.5.1 3x INPUT CAPTURE IN SLEEP MODE

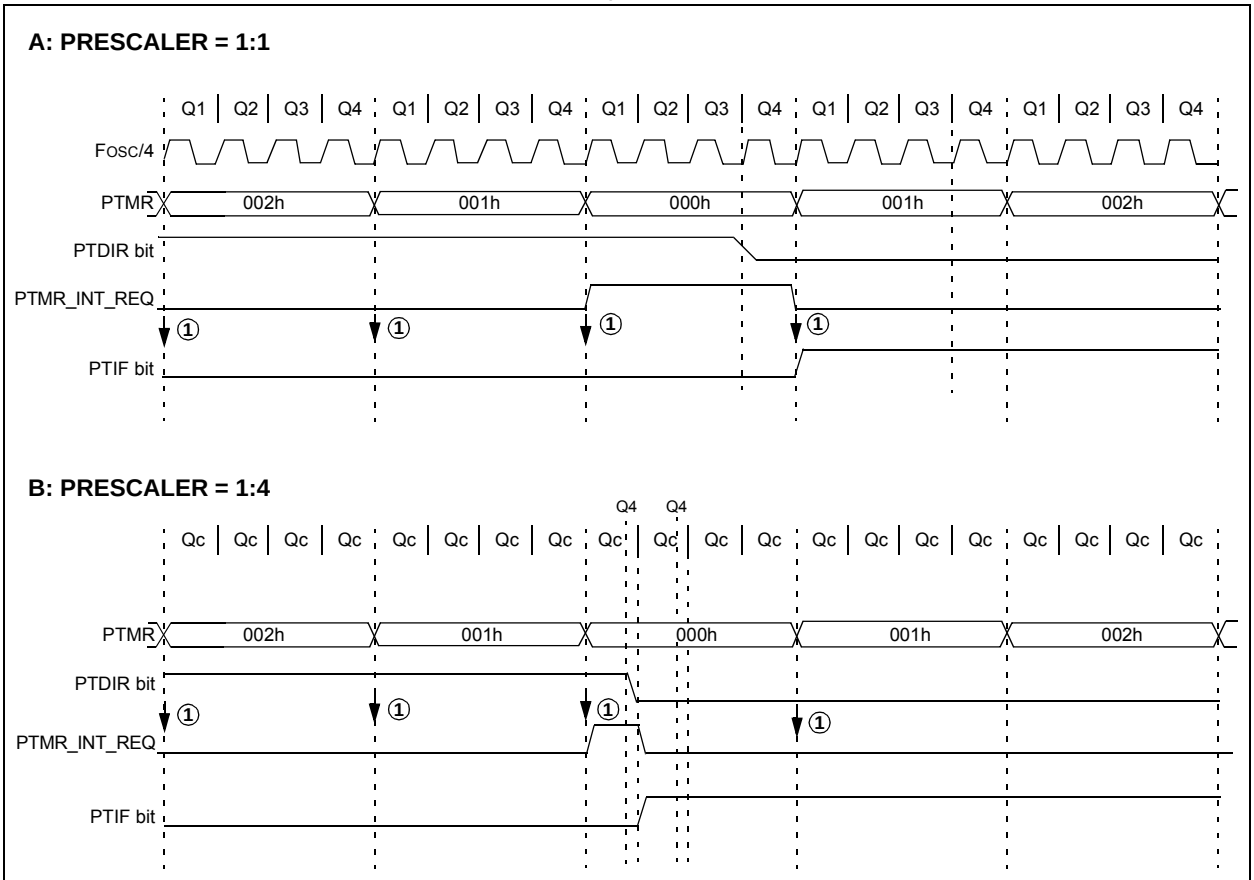
Since the input capture can operate only when its time base is configured in a Synchronous mode, the input capture will not capture any events. This is because the device's internal clock has been stopped and any internal timers in Synchronous modes will not increment. The prescaler will continue to count the events (not synchronized).

When the specified capture event occurs, the CAPx interrupt will be set. The Capture Buffer register will be updated upon wake-up from sleep to the current TMR5 value. If the CAPx interrupt is enabled, the device will wake-up from Sleep. This effectively enables all input capture channels to be used as the external interrupts.

17.5.2 QEI IN SLEEP MODE

All QEI functions are halted in Sleep mode.

FIGURE 18-7: PWM TIME BASE INTERRUPT, CONTINUOUS UP/DOWN COUNT MODE



Note 1: Interrupt flag bit, PTIF, is sampled here (every Q1).

18.11.3 PWM OUTPUT PIN RESET STATES

The PWMPIN Configuration bit determines the PWM output pins to be PWM output pins or digital I/O pins, after the device comes out of Reset. If the PWMPIN Configuration bit is unprogrammed (default), the PWMEN<2:0> control bits will be cleared on a device Reset. Consequently, all PWM outputs will be tri-stated and controlled by the corresponding PORT and TRIS registers. If the PWMPIN Configuration bit is programmed low, the PWMEN<2:0> control bits will be set, as follows, on a device Reset:

- PWMEN<2:0> = 101 if device has 8 PWM pins (PIC18F4331/4431 devices)
- PWMEN<2:0> = 100 if device has 6 PWM pins (PIC18F2331/2431 devices)

All PWM pins will be enabled for PWM output and will have the output polarity defined by the HPOL and LPOL Configuration bits.

18.12 PWM Fault Inputs

There are two Fault inputs associated with the PWM module. The main purpose of the input Fault pins is to disable the PWM output signals and drive them into an inactive state. The action of the Fault inputs is performed directly in hardware so that when a Fault occurs, it can be managed quickly and the PWM outputs are put into an inactive state to save the power devices connected to the PWMs.

The PWM Fault inputs are $\overline{\text{FLTA}}$ and $\overline{\text{FLTB}}$, which can come from I/O pins, the CPU or another module. The $\overline{\text{FLTA}}$ and $\overline{\text{FLTB}}$ pins are active-low inputs so it is easy to "OR" many sources to the same input. $\overline{\text{FLTB}}$ and its associated logic are not implemented on PIC18F2331/2431 devices.

The FLTCONFIG register (Register 18-8) defines the settings of $\overline{\text{FLTA}}$ and $\overline{\text{FLTB}}$ inputs.

Note:	The inactive state of the PWM pins are dependent on the HPOL and LPOL Configuration bit settings, which define the active and inactive state for PWM outputs.

18.12.1 FAULT PIN ENABLE BITS

By setting the bits, FLTAEN and FLTBEN in the FLTCONFIG register, the corresponding Fault inputs are enabled. If both bits are cleared, then the Fault inputs have no effect on the PWM module.

18.12.2 MFAULT INPUT MODES

The FLTAMOD and FLTBMOD bits in the FLTCONFIG register determine the modes of PWM I/O pins that are deactivated when they are overridden by Fault input.

The FLTAS and FLTBS bits in the FLTCONFIG register give the status of Fault A and Fault B inputs.

Each of the Fault inputs have two modes of operation:

- **Inactive Mode (FLTxMOD = 0)**

  This is a Catastrophic Fault Management mode. When the Fault occurs in this mode, the PWM outputs are deactivated. The PWM pins will remain in Inactivate mode until the Fault is cleared (Fault input is driven high) and the corresponding Fault Status bit has been cleared in software. The PWM outputs are enabled immediately at the beginning of the following PWM period, after the Fault Status bit (FLTxS) is cleared.

- **Cycle-by-Cycle Mode (FLTxMOD = 1)**

  When the Fault occurs in this mode, the PWM outputs are deactivated. The PWM outputs will remain in the defined Fault states (all PWM outputs inactive) for as long as the Fault pin is held low. After the Fault pin is driven high, the PWM outputs will return to normal operation at the beginning of the following PWM period and the FLTxS bit is automatically cleared.

REGISTER 18-8: FLTCONFIG: FAULT CONFIGURATION 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
BRFEN	FLTBS(1)	FLTBMOD(1)	FLTBEN(1)	FLTCON(2)	FLTAS	FLTAMOD	FLTAEN
bit 7							bit 0

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

bit 7 **BRFEN:** Breakpoint Fault Enable bit
- 1 = Enable Fault condition on a breakpoint (i.e., only when PWMPIN = 1)
- 0 = Disable Fault condition

bit 6 **FLTBS:** Fault B Status bit(1)
- 1 = $\overline{\text{FLTB}}$ is asserted:
  if FLTBMOD = 0, cleared by the user;
  if FLTBMOD = 1, cleared automatically at beginning of the new period when $\overline{\text{FLTB}}$ is deasserted
- 0 = No Fault

bit 5 **FLTBMOD:** Fault B Mode bit(1)
- 1 = Cycle-by-Cycle mode: Pins are inactive for the remainder of the current PWM period or until $\overline{\text{FLTB}}$ is deasserted; FLTBS is cleared automatically when $\overline{\text{FLTB}}$ is inactive (no Fault present)
- 0 = Inactive mode: Pins are deactivated (catastrophic failure) until $\overline{\text{FLTB}}$ is deasserted and FLTBS is cleared by the user only

bit 4 **FLTBEN:** Fault B Enable bit(1)
- 1 = Enable Fault B
- 0 = Disable Fault B

bit 3 **FLTCON:** Fault Configuration bit(2)
- 1 = $\overline{\text{FLTA}}$, $\overline{\text{FLTB}}$ or both deactivates all PWM outputs
- 0 = $\overline{\text{FLTA}}$ or $\overline{\text{FLTB}}$ deactivates PWM<5:0>

bit 2 **FLTAS:** Fault A Status bit
- 1 = $\overline{\text{FLTA}}$ is asserted:
  if FLTAMOD = 0, cleared by the user;
  if FLTAMOD = 1, cleared automatically at beginning of the new period when $\overline{\text{FLTA}}$ is deasserted
- 0 = No Fault

bit 1 **FLTAMOD:** Fault A Mode bit
- 1 = Cycle-by-Cycle mode: Pins are inactive for the remainder of the current PWM period or until $\overline{\text{FLTA}}$ is deasserted; FLTAS is cleared automatically
- 0 = Inactive mode: Pins are deactivated (catastrophic failure) until $\overline{\text{FLTA}}$ is deasserted and FLTAS is cleared by the user only

bit 0 **FLTAEN:** Fault A Enable bit
- 1 = Enable Fault A
- 0 = Disable Fault A

Note 1: Unimplemented in PIC18F2331/2431 devices; maintain these bits clear.

2: PWM<7:6> are implemented only on PIC18F4331/4431 devices. On PIC18F2331/2431 devices, setting or clearing FLTCON has no effect.

20.2 EUSART 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 also control the baud rate. In Synchronous mode, bit BRGH is ignored. Table 20-1 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 20-1. From this, the error in baud rate can be determined. An example calculation is shown in Example 20-1. Typical baud rates and error values for the various Asynchronous modes are shown in Table 20-2. 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.

20.2.1 POWER-MANAGED MODE OPERATION

The system clock is used to generate the desired baud rate. However, when a power-managed mode is entered, the clock source may be operating at a different frequency than in PRI_RUN mode. In Sleep mode, no clocks are present and in PRI_IDLE, the primary clock source continues to provide clocks to the Baud Rate Generator. However, in other power-managed modes, the clock frequency will probably change. This may require the value in SPBRG to be adjusted.

If the system clock is changed during an active receive operation, a receive error or data loss may result. To avoid this problem, check the status of the RCIDL bit and make sure that the receive operation is Idle before changing the system clock.

20.2.2 SAMPLING

The data on the RC7/RX/DT/SDO 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.

TABLE 20-1: BAUD RATE FORMULAS

Configuration Bits			BRG/EUSART Mode	Baud Rate Formula
SYNC	BRG16	BRGH		
0	0	0	8-Bit/Asynchronous	FOSC/[64 (n + 1)]
0	0	1	8-Bit/Asynchronous	FOSC/[16 (n + 1)]
0	1	0	16-Bit/Asynchronous	
0	1	1	16-Bit/Asynchronous	FOSC/[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

REGISTER 21-5: ADCHS: A/D CHANNEL SELECT 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
GDSEL1	GDSEL0	GBSEL1	GBSEL0	GCSEL1	GCSEL0	GASEL1	GASEL0
bit 7							bit 0

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

bit 7-6 **GDSEL<1:0>**: Group D Select bits
S/H-2 positive input.
00 = AN3
01 = AN7(1)
1x = Reserved

bit 5-4 **GBSEL<1:0>**: Group B Select bits
S/H-2 positive input.
00 = AN1
01 = AN5(1)
1x = Reserved

bit 3-2 **GCSEL<1:0>**: Group C Select bits
S/H-1 positive input.
00 = AN2
01 = AN6(1)
1x = Reserved

bit 1-0 **GASEL<1:0>**: Group A Select bits
S/H-1 positive input.
00 = AN0
01 = AN4
10 = AN8(1)
11 = Reserved

Note 1: AN5 through AN8 are available only in PIC18F4331/4431 devices.

REGISTER 23-2: CONFIG2L: CONFIGURATION REGISTER 2 LOW (BYTE ADDRESS 300002h)

U-0	U-0	U-0	U-0	R/P-1	R/P-1	R/P-1	R/P-1
—	—	—	—	BORV1	BORV0	BOREN(1)	$\overline{\text{PWRTEN}}$(1)
bit 7							bit 0

Legend:		
R = Readable bit	P = Programmable bit	U = Unimplemented bit, read as '0'
-n = Value when device is unprogrammed		U = Unchanged from programmed state

bit 7-4 **Unimplemented:** Read as '0'

bit 3-2 **BORV<1:0>:** Brown-out Reset Voltage bits
- 11 = Reserved
- 10 = V_{BOR} set to 2.7V
- 01 = V_{BOR} set to 4.2V
- 00 = V_{BOR} set to 4.5V

bit 1 **BOREN:** Brown-out Reset Enable bit(1)
- 1 = Brown-out Reset is enabled
- 0 = Brown-out Reset is disabled

bit 0 **$\overline{\text{PWRTEN}}$:** Power-up Timer Enable bit(1)
- 1 = PWRT is disabled
- 0 = PWRT is enabled

Note 1: Having BOREN = 1 does not automatically override the $\overline{\text{PWRTEN}}$ to '0', nor automatically enables the Power-up Timer.

REGISTER 23-7: CONFIG5L: CONFIGURATION REGISTER 5 LOW (BYTE ADDRESS 300008h)

U-0	U-0	U-0	U-0	R/C-1	R/C-1	R/C-1	R/C-1
—	—	—	—	CP3(1,2)	CP2(1,2)	CP1(2)	CP0(2)
bit 7							bit 0

Legend:		
R = Readable bit	C = Clearable bit	U = Unimplemented bit, read as '0'
-n = Value when device is unprogrammed		U = Unchanged from programmed state

bit 7-4 **Unimplemented:** Read as '0'

bit 3 **CP3:** Code Protection bit(1,2)
- 1 = Block 3 is not code-protected
- 0 = Block 3 is code-protected

bit 2 **CP2:** Code Protection bit(1,2)
- 1 = Block 2 is not code-protected
- 0 = Block 2 is code-protected

bit 1 **CP1:** Code Protection bit(2)
- 1 = Block 1 is not code-protected
- 0 = Block 1 is code-protected

bit 0 **CP0:** Code Protection bit(2)
- 1 = Block 0 is not code-protected
- 0 = Block 0 is code-protected

Note 1: Unimplemented in PIC18F2331/4331 devices; maintain this bit set.
2: Refer to Figure 23-5 for block boundary addresses.

REGISTER 23-8: CONFIG5H: CONFIGURATION REGISTER 5 HIGH (BYTE ADDRESS 300009h)

R/C-1	R/C-1	U-0	U-0	U-0	U-0	U-0	U-0
CPD(1)	CPB(1)	—	—	—	—	—	—
bit 7							bit 0

Legend:		
R = Readable bit	C = Clearable bit	U = Unimplemented bit, read as '0'
-n = Value when device is unprogrammed		U = Unchanged from programmed state

bit 7 **CPD:** Data EEPROM Code Protection bit(1)
- 1 = Data EEPROM is not code-protected
- 0 = Data EEPROM is code-protected

bit 6 **CPB:** Boot Block Code Protection bit(1)
- 1 = Boot Block is not code-protected
- 0 = Boot Block is code-protected

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

Note 1: Refer to Figure 23-5 for block boundary addresses.

REGISTER 23-13: DEVID1: DEVICE ID REGISTER 1 FOR PIC18F2331/2431/4331/4431 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	P = Programmable bit	U = Unimplemented bit, read as '0'
-n = Value when device is unprogrammed		U = Unchanged from programmed state

bit 7-5 **DEV<2:0>:** Device ID bits
These bits are used with the DEV<10:3> bits in the Device ID Register 2 to identify the part number.
000 = PIC18F4331
001 = PIC18F4431
100 = PIC18F2331
101 = PIC18F2431

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

REGISTER 23-14: DEVID2: DEVICE ID REGISTER 2 FOR PIC18F2331/2431/4331/4431 DEVICES

R	R	R	R	R	R	R	R
DEV10(1)	DEV9(1)	DEV8(1)	DEV7(1)	DEV6(1)	DEV5(1)	DEV4(1)	DEV3(1)
bit 7							bit 0

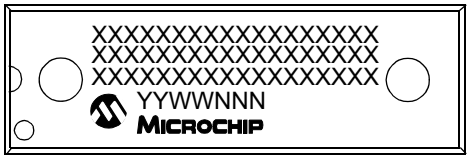
Legend:		
R = Readable bit	P = Programmable bit	U = Unimplemented bit, read as '0'
-n = Value when device is unprogrammed		U = Unchanged from programmed state

bit 7-0 **DEV<10:3>:** Device ID bits(1)
These bits are used with the DEV<2:0> bits in the Device ID Register 1 to identify the part number
0000 0101 = PIC18F2331/2431/4331/4431 devices

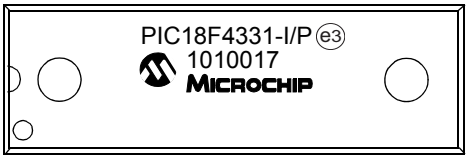
Note 1: These values for DEV<10:3> may be shared with other devices. The specific device is always identified by using the entire DEV<10:0> bit sequence.

27.1 Package Marking Information (Continued)

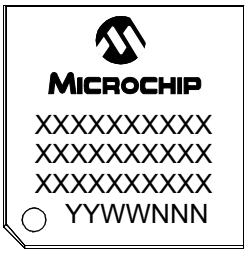
40-Lead PDIP



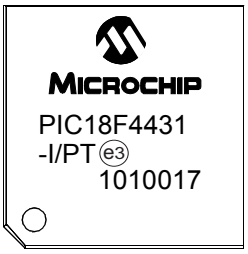
Example



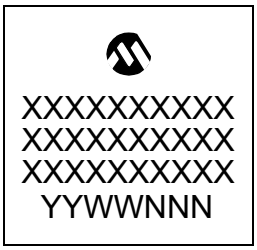
44-Lead TQFP



Example



44-Lead QFN



Example