



Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

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

Details

Product Status	Active
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	24
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)	2V ~ 5.5V
Data Converters	A/D 5x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Through Hole
Package / Case	28-DIP (0.300", 7.62mm)
Supplier Device Package	28-SPDIP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18lf2431-i-sp

Email: info@E-XFL.COM

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

2.5 External Oscillator Pins

Many microcontrollers have options for at least two oscillators: a high-frequency primary oscillator and a low-frequency secondary oscillator (refer to **Section 3.0 "Oscillator Configurations"** for details).

The oscillator circuit should be placed on the same side of the board as the device. Place the oscillator circuit close to the respective oscillator pins with no more than 0.5 inch (12 mm) between the circuit components and the pins. The load capacitors should be placed next to the oscillator itself, on the same side of the board.

Use a grounded copper pour around the oscillator circuit to isolate it from surrounding circuits. The grounded copper pour should be routed directly to the MCU ground. Do not run any signal traces or power traces inside the ground pour. Also, if using a two-sided board, avoid any traces on the other side of the board where the crystal is placed.

Layout suggestions are shown in Figure 2-4. In-line packages may be handled with a single-sided layout that completely encompasses the oscillator pins. With fine-pitch packages, it is not always possible to completely surround the pins and components. A suitable solution is to tie the broken guard sections to a mirrored ground layer. In all cases, the guard trace(s) must be returned to ground.

In planning the application's routing and I/O assignments, ensure that adjacent port pins and other signals in close proximity to the oscillator are benign (i.e., free of high frequencies, short rise and fall times, and other similar noise).

For additional information and design guidance on oscillator circuits, please refer to these Microchip Application Notes, available at the corporate web site (www.microchip.com):

- AN826, "Crystal Oscillator Basics and Crystal Selection for rfPIC[™] and PICmicro[®] Devices"
- AN849, "Basic PICmicro[®] Oscillator Design"
- AN943, "Practical PICmicro[®] Oscillator Analysis and Design"
- AN949, "Making Your Oscillator Work"

2.6 Unused I/Os

Unused I/O pins should be configured as outputs and driven to a logic low state. Alternatively, connect a 1 k Ω to 10 k Ω resistor to Vss on unused pins and drive the output to logic low.

FIGURE 2-3: SUGGESTED PLACEMENT OF THE OSCILLATOR CIRCUIT



3.7 Clock Sources and Oscillator Switching

Like previous PIC18 devices, the PIC18F2331/2431/ 4331/4431 devices include a feature that allows the system clock source to be switched from the main oscillator to an alternate low-frequency clock source. PIC18F2331/ 2431/4331/4431 devices offer two alternate clock sources. When enabled, these give additional options for switching to the various power-managed operating modes.

Essentially, there are three clock sources for these devices:

- Primary oscillators
- · Secondary oscillators
- Internal oscillator block

The **primary oscillators** include the External Crystal and Resonator modes, the External RC modes, the External Clock modes and the internal oscillator block. The particular mode is defined on POR by the contents of Configuration Register 1H. The details of these modes are covered earlier in this chapter.

The **secondary oscillators** are those external sources not connected to the OSC1 or OSC2 pins. These sources may continue to operate even after the controller is placed in a power-managed mode.

PIC18F2331/2431/4331/4431 devices offer only the Timer1 oscillator as a secondary oscillator. This oscillator, in all power-managed modes, is often the time base for functions such as a Real-Time Clock (RTC).

Most often, a 32.768 kHz watch crystal is connected between the RC0/T1OSO/T1CKI and RC1/T1OSI/CCP2/FLTA pins. Like the LP Oscillator mode circuit, loading capacitors are also connected from each pin to ground.

The Timer1 oscillator is discussed in greater detail in **Section 13.2 "Timer1 Oscillator**".

In addition to being a primary clock source, the **internal oscillator block** is available as a power-managed mode clock source. The INTRC source is also used as the clock source for several special features, such as the WDT and Fail-Safe Clock Monitor.

The clock sources for the PIC18F2331/2431/4331/4431 devices are shown in Figure 3-8. See **Section 13.0** "**Timer1 Module**" for further details of the Timer1 oscillator. See **Section 23.1** "**Configuration Bits**" for Configuration register details.

3.7.1 OSCILLATOR CONTROL REGISTER

The OSCCON register (Register 3-2) controls several aspects of the system clock's operation, both in full-power operation and in power-managed modes.

The System Clock Select bits, SCS<1:0>, select the clock source that is used when the device is operating in power-managed modes. The available clock sources are the primary clock (defined in Configuration Register 1H), the secondary clock (Timer1 oscillator) and the internal oscillator block. The clock selection has no effect until a SLEEP instruction is executed and the device enters a power-managed mode of operation. The SCS bits are cleared on all forms of Reset.

The Internal Oscillator Select bits, IRCF<2:0>, select the frequency output of the internal oscillator block that is used to drive the system clock. The choices are the INTRC source, the INTOSC source (8 MHz) or one of the six frequencies derived from the INTOSC postscaler (125 kHz to 4 MHz). If the internal oscillator block is supplying the system clock, changing the states of these bits will have an immediate change on the internal oscillator's output. On device Resets, the default output frequency of the internal oscillator block is set at 32 kHz.

The OSTS, IOFS and T1RUN bits indicate which clock source is currently providing the system clock. The OSTS indicates that the Oscillator Start-up Timer has timed out, and the primary clock is providing the system clock in Primary Clock modes. The IOFS bit indicates when the internal oscillator block has stabilized, and is providing the system clock in RC Clock modes. The T1RUN bit (T1CON<6>) indicates when the Timer1 oscillator is providing the system clock in Secondary Clock modes. In power-managed modes, only one of these three bits will be set at any time. If none of these bits are set, the INTRC is providing the system clock, or the internal oscillator block has just started and is not yet stable.

The IDLEN bit controls the selective shutdown of the controller's CPU in power-managed modes. The use of these bits is discussed in more detail in **Section 4.0** "**Power-Managed Modes**"

- Note 1: The Timer1 oscillator must be enabled to select the secondary clock source. The Timer1 oscillator is enabled by setting the T1OSCEN bit in the Timer1 Control register (T1CON<3>). If the Timer1 oscillator is not enabled, then any attempt to select a secondary clock source, when executing a SLEEP instruction, will be ignored.
 - 2: It is recommended that the Timer1 oscillator be operating and stable before executing the SLEEP instruction, or a very long delay may occur while the Timer1 oscillator starts.

4.1.3 CLOCK TRANSITIONS AND STATUS INDICATORS

The length of the transition between clock sources is the sum of two cycles of the old clock source and three to four cycles of the new clock source. This formula assumes that the new clock source is stable.

Three bits indicate the current clock source and its status. They are:

- OSTS (OSCCON<3>)
- IOFS (OSCCON<2>)
- T1RUN (T1CON<6>)

In general, only one of these bits will be set while in a given power-managed mode. When the OSTS bit is set, the primary clock is providing the device clock. When the IOFS bit is set, the INTOSC output is providing a stable, 8 MHz clock source to a divider that actually drives the device clock. When the T1RUN bit is set, the Timer1 oscillator is providing the clock. If none of these bits are set, then either the INTRC clock source is clocking the device, or the INTOSC source is not yet stable.

If the internal oscillator block is configured as the primary clock source by the FOSC<3:0> Configuration bits, then both the OSTS and IOFS bits may be set when in PRI_RUN or PRI_IDLE modes. This indicates that the primary clock (INTOSC output) is generating a stable, 8 MHz output. Entering another power-managed RC mode at the same frequency would clear the OSTS bit.

- Note 1: Caution should be used when modifying a single IRCF bit. If VDD is less than 3V, it is possible to select a higher clock speed than is supported by the low VDD. Improper device operation may result if the VDD/FOSC specifications are violated.
 - 2: Executing a SLEEP instruction does not necessarily place the device into Sleep mode. It acts as the trigger to place the controller into either the Sleep mode or one of the Idle modes, depending on the setting of the IDLEN bit.

4.1.4 MULTIPLE SLEEP COMMANDS

The power-managed mode that is invoked with the SLEEP instruction is determined by the setting of the IDLEN bit at the time the instruction is executed. If another SLEEP instruction is executed, the device will enter the power-managed mode specified by IDLEN at that time. If IDLEN has changed, the device will enter the new power-managed mode specified by the new setting.

4.2 Run Modes

In the Run modes, clocks to both the core and peripherals are active. The difference between these modes is the clock source.

4.2.1 PRI_RUN MODE

The PRI_RUN mode is the normal, full-power execution mode of the microcontroller. This is also the default mode upon a device Reset unless Two-Speed Start-up is enabled (see **Section 23.3 "Two-Speed Start-up"** for details). In this mode, the OSTS bit is set. The IOFS bit may be set if the internal oscillator block is the primary clock source (see **Section 3.7.1 "Oscillator Control Register"**).

4.2.2 SEC_RUN MODE

The SEC_RUN mode is the compatible mode to the "clock switching" feature offered in other PIC18 devices. In this mode, the CPU and peripherals are clocked from the Timer1 oscillator. This gives users the option of lower power consumption while still using a high-accuracy clock source.

SEC_RUN mode is entered by setting the SCS<1:0> bits to '01'. The device clock source is switched to the Timer1 oscillator (see Figure 4-1), the primary oscillator is shut down, the T1RUN bit (T1CON<6>) is set and the OSTS bit is cleared.

Note: The Timer1 oscillator should already be running prior to entering SEC_RUN mode. If the T1OSCEN bit is not set when the SCS<1:0> bits are set to '01', entry to SEC_RUN mode will not occur. If the Timer1 oscillator is enabled, but not yet running, device clocks will be delayed until the oscillator has started. In such situations, initial oscillator operation is far from stable and unpredictable operation may result.

On transitions from SEC_RUN mode to PRI_RUN, the peripherals and CPU continue to be clocked from the Timer1 oscillator while the primary clock is started. When the primary clock becomes ready, a clock switch back to the primary clock occurs (see Figure 4-2). When the clock switch is complete, the T1RUN bit is cleared, the OSTS bit is set and the primary clock is providing the clock. The IDLEN and SCS bits are not affected by the wake-up; the Timer1 oscillator continues to run.

Register	Applicable Devices		Power-on Reset, Brown-out Reset	MCLR Resets WDT Reset RESET Instruction Stack Resets	Wake-up via WDT or Interrupt		
IPR2	2331	2431	4331	4431	11 -1-1	11 -1-1	uu -u-u
PIR2	2331	2431	4331	4431	00-0	00-0	uu -u-u
PIE2	2331	2431	4331	4431	00-0	00-0	uu -u-u
IPR1	2331	2431	4331	4431	-111 1111	-111 1111	-uuu uuuu
PIR1	2331	2431	4331	4431	-000 0000	-000 0000	-uuu uuuu ⁽¹⁾
	2331	2431	4331	4431	-000 0000	-000 0000	-uuu uuuu ⁽¹⁾
PIE1	2331	2431	4331	4431	0000 0000	0000 0000	սսսս սսսս
	2331	2431	4331	4431	-000 0000	-000 0000	-uuu uuuu
OSCTUNE	2331	2431	4331	4431	00 0000	00 0000	uu uuuu
TRISE ⁽⁶⁾	2331	2431	4331	4431	111	111	uuu
TRISD	2331	2431	4331	4431	1111 1111	1111 1111	սսսս սսսս
TRISC	2331	2431	4331	4431	1111 1111	1111 1111	սսսս սսսս
TRISB	2331	2431	4331	4431	1111 1111	1111 1111	սսսս սսսս
TRISA ⁽⁵⁾	2331	2431	4331	4431	1111 1111 (5)	1111 1111 (5)	uuuu uuuu ⁽⁵⁾
PR5H	2331	2431	4331	4431	1111 1111	1111 1111	սսսս սսսս
PR5L	2331	2431	4331	4431	1111 1111	1111 1111	սսսս սսսս
LATE ⁽⁶⁾	2331	2431	4331	4431	xxx	uuu	uuu
LATD	2331	2431	4331	4431	xxxx xxxx	uuuu uuuu	uuuu uuuu
LATC	2331	2431	4331	4431	xxxx xxxx	սսսս սսսս	սսսս սսսս
LATB	2331	2431	4331	4431	xxxx xxxx	սսսս սսսս	սսսս սսսս
LATA ⁽⁵⁾	2331	2431	4331	4431	xxxx xxxx ⁽⁵⁾	uuuu uuuu ⁽⁵⁾	uuuu uuuu ⁽⁵⁾
TMR5H	2331	2431	4331	4431	xxxx xxxx	uuuu uuuu	սսսս սսսս
TMR5L	2331	2431	4331	4431	xxxx xxxx	սսսս սսսս	սսսս սսսս
PORTE ⁽⁶⁾	2331	2431	4331	4431	xxxx	xxxx	uuuu
PORTD	2331	2431	4331	4431	xxxx xxxx	uuuu uuuu	นนนน นนนน
PORTC	2331	2431	4331	4431	xxxx xxxx	นนนน นนนน	นนนน นนนน
PORTB	2331	2431	4331	4431	xxxx xxxx	uuuu uuuu	นนนน นนนน
PORTA ⁽⁵⁾	2331	2431	4331	4431	xx0x 0000 (5)	uu0u 0000 (5)	uuuu uuuu ⁽⁵⁾

TABLE 5-3: INITIALIZATION CONDITIONS FOR ALL REGISTERS (CONTINUED)

Legend: u = unchanged, x = unknown, - = unimplemented bit, read as '0', q = value depends on condition. Shaded cells indicate conditions do not apply for the designated device.

Note 1: One or more bits in the INTCONx or PIRx registers will be affected (to cause wake-up).

2: When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the PC is loaded with the interrupt vector (0008h or 0018h).

3: When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the TOSU, TOSH and TOSL are updated with the current value of the PC. The STKPTR is modified to point to the next location in the hardware stack.

4: See Table 5-2 for Reset value for specific condition.

5: Bits 6 and 7 of PORTA, LATA and TRISA are enabled depending on the oscillator mode selected. When not enabled as PORTA pins, they are disabled and read '0'.

6: Bit 3 of PORTE and LATE are enabled if MCLR functionality is disabled. When not enabled as the PORTE pin, they are disabled and read as '0'. The 28-pin devices do not have only RE3 implemented.

6.1.1 PROGRAM COUNTER

The Program Counter (PC) specifies the address of the instruction to fetch for execution. The PC is 21 bits wide and contained in three 8-bit registers. The low byte, known as the PCL register, is both readable and writable. The high byte (PCH register) contains the PC<15:8> bits and is not directly readable or writable.

Updates to the PCH register are performed through the PCLATH register. The upper byte is the PCU register and contains the bits, PC<20:16>. This register is also not directly readable or writable. Updates to the PCU register are performed through the PCLATU register.

The contents of PCLATH and PCLATU are transferred to the program counter by any operation that writes to the PCL. Similarly, the upper two bytes of the program counter are transferred to PCLATH and PCLATU by an operation that reads PCL. This is useful for computed offsets to the PC (see **Section 6.1.4.1 "Computed GOTO**").

The PC addresses bytes in the program memory. To prevent the PC from becoming misaligned with word instructions, the Least Significant bit of the PCL is fixed to a value of '0'. The PC increments by two to address sequential instructions in the program memory.

The CALL, RCALL, GOTO and program branch instructions write to the program counter directly. For these instructions, the contents of PCLATH and PCLATU are not transferred to the program counter.

6.1.2 RETURN ADDRESS STACK

The return address stack allows any combination of up to 31 program calls and interrupts to occur. The PC (Program Counter) is pushed onto the stack when a CALL or RCALL instruction is executed, or an interrupt is Acknowledged. The PC value is pulled off the stack on a RETURN, RETLW or a RETFIE instruction. PCLATU and PCLATH are not affected by any of the RETURN or CALL instructions.

The stack operates as a 31-word by 21-bit RAM and a 5-bit Stack Pointer, with the Stack Pointer initialized to 00000b after all Resets. There is no RAM associated with Stack Pointer, 00000b. This is only a Reset value. During a CALL type instruction, causing a push onto the stack, the Stack Pointer is first incremented and the RAM location pointed to by the Stack Pointer is written with the contents of the PC (already pointing to the instruction, causing a pop from the stack, the contents of the RAM location pointed to by the STKPTR are transferred to the PC and then the Stack Pointer is decremented.

The stack space is not part of either program or data space. The Stack Pointer is readable and writable, and the address on the top of the stack is readable and writable through the Top-of-Stack (TOS) Special Function Registers. Data can also be pushed to, or popped from, the stack using the Top-of-Stack SFRs. Status bits indicate if the stack is full, has overflowed or underflowed.

6.1.2.1 Top-of-Stack Access

The top of the stack is readable and writable. Three register locations, TOSU, TOSH and TOSL, hold the contents of the stack location pointed to by the STKPTR register (Figure 6-3). This allows users to implement a software stack if necessary. After a CALL, RCALL or interrupt, the software can read the pushed value by reading the TOSU, TOSH and TOSL registers. These values can be placed on a user-defined software stack. At return time, the software can replace the TOSU, TOSH and TOSL and do a return.

The user must disable the global interrupt enable bits while accessing the stack to prevent inadvertent stack corruption.

6.1.2.2 Return Stack Pointer (STKPTR)

The STKPTR register (Register 6-1) contains the Stack Pointer value, the STKFUL (Stack Full) status bit and the STKUNF (Stack Underflow) status bits. The value of the Stack Pointer can be 0 through 31. The Stack Pointer increments before values are pushed onto the stack and decrements after values are popped off the stack. At Reset, the Stack Pointer value will be zero. The user may read and write the Stack Pointer value. This feature can be used by a Real-Time Operating System (RTOS) for return stack maintenance.

After the PC is pushed onto the stack 31 times (without popping any values off the stack), the STKFUL bit is set. The STKFUL bit is cleared by software or by a POR.

The action that takes place when the stack becomes full depends on the state of the STVREN (Stack Overflow Reset Enable) Configuration bit. (Refer to **Section 23.1 "Configuration Bits**" for a description of the device Configuration bits.) If STVREN is set (default), the 31st push will push the (PC + 2) value onto the stack, set the STKFUL bit and reset the device. The STKFUL bit will remain set and the Stack Pointer will be set to zero.

If STVREN is cleared, the STKFUL bit will be set on the 31st push and the Stack Pointer will increment to 31. Any additional pushes will not overwrite the 31st push and STKPTR will remain at 31.

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 (erasebefore-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 26-1 in **Section 26.0** "**Electrical Characteristics**") for exact limits.

7.1 EEADR

The Address register can address 256 bytes of data EEPROM.

7.2 EECON1 and EECON2 Registers

Access to the data EEPROM is controlled by two registers: EECON1 and EECON2. These are the same registers which control access to the program memory and are used in a similar manner for the data EEPROM.

The EECON1 register (Register 7-1) is the control register for data and program memory access. 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 Flash program or data EEPROM memory.

The WREN bit, when set, will allow a write operation. On power-up, the WREN bit is clear. The WRERR bit is set in hardware when the WREN bit is set and cleared when the internal programming timer expires and the write operation is complete.

Note:	During normal operation, the WRERR bit								
	is read as '1'. This can indicate that a write								
	operation was prematurely terminated by								
	a Reset or a write operation was								
	attempted improperly.								

The WR control bit initiates write operations. The bit cannot be cleared, only set, in software; it is cleared in hardware at the completion of the write operation.

Note: The EEIF interrupt flag bit (PIR2<4>) is set when the write is complete. It must be cleared in software.

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 7.3** "**Reading the Data EEPROM Memory**" regarding table reads.

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

8.0 FLASH PROGRAM MEMORY

The Flash program memory is readable, writable and erasable during normal operation over the entire VDD range.

A read from program memory is executed on one byte at a time. A write to program memory is executed on blocks of 8 bytes at a time. Program memory is erased in blocks of 64 bytes at a time. A bulk erase operation may not be issued from user code.

While writing or erasing program memory, instruction fetches cease until the operation is complete. The program memory cannot be accessed during the write or erase, therefore, code cannot execute. An internal programming timer terminates program memory writes and erases.

A value written to program memory does not need to be a valid instruction. Executing a program memory location that forms an invalid instruction results in a NOP.

8.1 Table Reads and Table Writes

In order to read and write program memory, there are two operations that allow the processor to move bytes between the program memory space and the data RAM:

- Table Read (TBLRD)
- Table Write (TBLWT)





Table read operations retrieve data from program memory and place it into TABLAT in the data RAM space. Figure 8-1 shows the operation of a table read with program memory and data RAM.

Table write operations store data from TABLAT in the data memory space into holding registers in program memory. The procedure to write the contents of the holding registers into program memory is detailed in **Section 8.5 "Writing to Flash Program Memory"**. Figure 8-2 shows the operation of a table write with program memory and data RAM.

Table operations work with byte entities. A table block containing data, rather than program instructions, is not required to be word-aligned. Therefore, a table block can start and end at any byte address. If a table write is being used to write executable code into program memory, program instructions will need to be word-aligned, (TBLPTRL<0> = 0).



17.1.3.1 Pulse-Width Measurement Timing

Pulse-width measurement accuracy can only be ensured when the pulse-width high and low present on the CAPx input exceeds one TcY clock cycle. The limitations depend on the mode selected:

- When CAPxM<3:0> = 0110 (rising to falling edge delay), the CAPx input high pulse width (TccH) must exceed Tcy + 10 ns.
- When CAPxM<3:0> = 0111 (falling to rising edge delay), the CAPx input low pulse width (TccL) must exceed TcY + 10 ns.
 - Note 1: The Period Measurement mode will produce valid results upon sampling of the second rising edge of the input capture. CAPxBUF values latched during the first active edge after initialization are invalid.
 - 2: The Pulse-Width Measurement mode will latch the value of the timer upon sampling of the first input signal edge by the input capture.

17.1.4 INPUT CAPTURE ON STATE CHANGE

When CAPxM<3:0> = 1000, the value is captured on every signal change on the CAPx input. If all three capture channels are configured in this mode, the three input captures can be used as the Hall effect sensor state transition detector. The value of Timer5 can be captured, Timer5 reset and the interrupt generated. Any change on CAP1, CAP2 or CAP3 is detected and the associated time base count is captured.

For position and velocity measurement in this mode, the timer can be optionally reset (see **Section 17.1.6 "Timer5 Reset"** for Reset options).

FIGURE 17-6: INPUT CAPTURE ON STATE CHANGE (HALL EFFECT SENSOR MODE)



Note 1: TMR5 can be selected as the time base for input capture. The time base can be optionally reset when the Capture Reset Enable bit is set (CAPxREN = 1).

2: Detailed CAPxBUF event timing (all modes reflect the same capture and Reset timing) is shown in Figure 17-4. There are six commutation BLDC Hall effect sensor states shown. The other two remaining states (i.e., 000h and 111h) are invalid in the normal operation. They remain to be decoded by the CPU firmware in BLDC motor application.

PIC18F2331/2431/4331/4431



The PWM time base can be configured for four different modes of operation:

- Free-Running mode
- Single-Shot mode
- Continuous Up/Down Count mode
- Continuous Up/Down Count mode with interrupts for double updates

These four modes are selected by the PTMOD<1:0> bits in the PTCON0 register. The Free-Running mode produces edge-aligned PWM generation. The Continuous Up/Down Count modes produce center-aligned PWM generation. The Single-Shot mode allows the PWM module to support pulse control of certain Electronically Commutated Motors (ECMs) and produces edge-aligned operation. Table 18-1 shows the minimum PWM frequencies that can be generated with the PWM time base and the prescaler. An operating frequency of 40 MHz (FCYC = 10 MHz) and PTPER = 0xFFF is assumed in the table. The PWM module must be capable of generating PWM signals at the line frequency (50 Hz or 60 Hz) for certain power control applications.

Minimum PWM Frequencies vs. Prescaler Value for Fcyc = 10 MIPS (PTPER = 0FFFh)						
Prescale PWM Frequency Edge-Aligned PWM Frequency Center-Aligned						
1:1	2441 Hz	1221 Hz				
1:4	610 Hz	305 Hz				
1:16	153 Hz	76 Hz				
1:64	38 Hz	19 Hz				

TABLE 18-1: MINIMUM PWM FREQUENCY

18.3.5 PWM TIME BASE POSTSCALER

The match output of PTMR can optionally be postscaled through a 4-bit postscaler (which gives a 1:1 to 1:16 scaling inclusive) to generate an interrupt. The postscaler counter is cleared when any of the following occurs:

- Write to the PTMR register
- Write to the PTCON register
- Any device Reset

The PTMR register is not cleared when PTCON is written.

18.4 PWM Time Base Interrupts

The PWM timer can generate interrupts based on the modes of operation selected by the PTMOD<1:0> bits and the postscaler bits (PTOPS<3:0>).

18.4.1 INTERRUPTS IN FREE-RUNNING MODE

When the PWM time base is in the Free-Running mode (PTMOD<1:0> = 00), an interrupt event is generated each time a match with the PTPER register occurs. The PTMR register is reset to zero in the following clock edge.

Using a postscaler selection other than 1:1 will reduce the frequency of interrupt events.

FIGURE 18-5: PWM TIME BASE INTERRUPT TIMING, FREE-RUNNING MODE



18.6.2 DUTY CYCLE REGISTER BUFFERS

The four PWM Duty Cycle registers are double-buffered to allow glitchless updates of the PWM outputs. For each duty cycle block, there is a Duty Cycle Buffer register that is accessible by the user and a second Duty Cycle register that holds the actual compare value used in the present PWM period.

In Edge-Aligned PWM Output mode, a new duty cycle value will be updated whenever a PTMR match with the PTPER register occurs and PTMR is reset as shown in Figure 18-12. Also, the contents of the duty cycle buffers are automatically loaded into the Duty Cycle registers when the PWM time base is disabled (PTEN = 0).

When the PWM time base is in the Continuous Up/Down Count mode, new duty cycle values will be updated when the value of the PTMR register is zero and the PWM time base begins to count upwards. The contents of the duty cycle buffers are automatically loaded into the Duty Cycle registers when the PWM time base is disabled (PTEN = 0). Figure 18-13 shows the timings when the duty cycle update occurs for the Continuous Up/Down Count mode. In this mode, up to one entire PWM period is available for calculating and loading the new PWM duty cycle before changes take effect.

When the PWM time base is in the Continuous Up/Down Count mode with double updates, new duty cycle values will be updated when the value of the PTMR register is zero and when the value of the PTMR register matches the value in the PTPER register. The contents of the duty cycle buffers are automatically loaded into the Duty Cycle registers during both of the previously described conditions. Figure 18-14 shows the duty cycle updates for Continuous Up/Down Count mode with double updates. In this mode, only up to half of a PWM period is available for calculating and loading the new PWM duty cycle before changes take effect.

18.6.3 EDGE-ALIGNED PWM

Edge-aligned PWM signals are produced by the module when the PWM time base is in the Free-Running mode or the Single-Shot mode. For edge-aligned PWM outputs, the output for a given PWM channel has a period specified by the value loaded in PTPER and a duty cycle specified by the appropriate Duty Cycle register (see Figure 18-12). The PWM output is driven active at the beginning of the period (PTMR = 0) and is driven inactive when the value in the Duty Cycle register matches PTMR. A new cycle is started when PTMR matches the PTPER as explained in the PWM period section.

If the value in a particular Duty Cycle register is zero, then the output on the corresponding PWM pin will be inactive for the entire PWM period. In addition, the output on the PWM pin will be active for the entire PWM period if the value in the Duty Cycle register is greater than the value held in the PTPER register.





FIGURE 18-13: DUTY CYCLE UPDATE TIMES IN CONTINUOUS UP/DOWN COUNT MODE



18.6.5 COMPLEMENTARY PWM OPERATION

The Complementary mode of PWM operation is useful to drive one or more power switches in half-bridge configuration as shown in Figure 18-16. This inverter topology is typical for a 3-phase induction motor, brushless DC motor or a 3-phase Uninterruptible Power Supply (UPS) control applications.

Each upper/lower power switch pair is fed by a complementary PWM signal. Dead time may be optionally inserted during device switching, where both outputs are inactive for a short period (see **Section 18.7 "Dead-Time Generators"**).

In Complementary mode, the duty cycle comparison units are assigned to the PWM outputs as follows:

- · PDC0 register controls PWM1/PWM0 outputs
- PDC1 register controls PWM3/PWM2 outputs
- · PDC2 register controls PWM5/PWM4 outputs
- PDC3 register controls PWM7/PWM6 outputs

PWM1/3/5/7 are the main PWMs that are controlled by the PDCx registers and PWM0/2/4/6 are the complemented outputs. When using the PWMs to control the half bridge, the odd numbered PWMs can be used to control the upper power switch and the even numbered PWMs used for the lower switches.

FIGURE 18-16: TYPICAL LOAD FOR COMPLEMENTARY PWM OUTPUTS



The Complementary mode is selected for each PWM I/O pin pair by clearing the appropriate PMODx bit in the PWMCON0 register. The PWM I/O pins are set to Complementary mode by default upon all kinds of device Resets.

The actual dead time is calculated from the DTCON register as follows:

Dead Time = Dead-Time Value/(Fosc/Prescaler)

Table 18-3 shows example dead-time ranges as a function of the input clock prescaler selected and the device operating frequency.

TABLE 18-3:	EXAMPLE DEAD-TIME
	RANGES

Fosc (MHz)	MIPS	Prescaler Selection	Dead-Time Min	Dead-Time Max
40	10	Fosc/2	50 ns	3.2 μs
40	10	Fosc/4	100 ns	6.4 μs
40	10	Fosc/8	200 ns	12.8 μs
40	10	Fosc/16	400 ns	25.6 μs
32	8	Fosc/2	62.5 ns	4 μs
32	8	Fosc/4	125 ns	8 μ s
32	8	Fosc/8	250 ns	16 μs
32	8	Fosc/16	500 ns	32 μs
25	6.25	Fosc/2	80 ns	5.12 μs
25	6.25	Fosc/4	160 ns	10.2 μs
25	6.25	Fosc/8	320 ns	20.5 μs
25	6.25	Fosc/16	640 ns	41 μs
20	5	Fosc/2	100 ns	6.4 μs
20	5	Fosc/4	200 ns	12.8 μs
20	5	Fosc/8	400 ns	25.6 μs
20	5	Fosc/16	800 ns	51.2 μs
10	2.5	Fosc/2	200 ns	12.8 μs
10	2.5	Fosc/4	400 ns	25.6 μs
10	2.5	Fosc/8	800 ns	51.2 μs
10	2.5	Fosc/16	1.6 μ s	102.4 μs
5	1.25	Fosc/2	400 ns	25.6 μs
5	1.25	Fosc/4	800 ns	51.2 μs
5	1.25	Fosc/8	1.6 μs	102.4 μs
5	1.25	Fosc/16	3.2 μs	204.8 μs
4	1	Fosc/2	0.5 μs	32 μs
4	1	Fosc/4	1 μs	64 μs
4	1	Fosc/8	2 μs	128 μs
4	1	Fosc/16	4 μs	256 μs

18.7.4 DEAD-TIME DISTORTION

- Note 1: For small PWM duty cycles, the ratio of dead time to the active PWM time may become large. In this case, the inserted dead time will introduce distortion into waveforms produced by the PWM module. The user can ensure that dead-time distortion is minimized by keeping the PWM duty cycle at least three times larger than the dead time. A similar effect occurs for duty cycles at or near 100%. The maximum duty cycle used in the application should be chosen such that the minimum inactive time of the signal is at least three times larger than the dead time. If the dead time is greater or equal to the duty cycle of one of the PWM output pairs, then that PWM pair will be inactive for the whole period.
 - Changing the dead-time values in DTCON when the PWM is enabled may result in an undesired situation. Disable the PWM (PTEN = 0) before changing the dead-time value

18.8 Independent PWM Output

Independent PWM mode is used for driving the loads (as shown in Figure 18-19) for driving one winding of a switched reluctance motor. A particular PWM output pair is configured in the Independent Output mode when the corresponding PMOD bit in the PWMCON0 register is set. No dead-time control is implemented between the PWM I/O pins when the module is operating in the Independent PWM mode and both I/O pins are allowed to be active simultaneously. This mode can also be used to drive stepper motors.

18.8.1 DUTY CYCLE ASSIGNMENT IN THE INDEPENDENT PWM MODE

In the Independent PWM mode, each duty cycle generator is connected to both PWM output pins in a given PWM output pair. The odd and even PWM output pins are driven with a single PWM duty cycle generator. PWM1 and PWM0 are driven by the PWM channel which uses the PDC0 register to set the duty cycle, PWM3 and PWM2 with PDC1, PWM5 and PWM4 with PDC2, and PWM7 and PWM6 with PDC3 (see Figure 18-3 and Register 18-4).

PIC18F2331/2431/4331/4431



FIGURE 19-3: SPI MODE TIMING (SLAVE MODE WITH CKE = 0)



20.3.5 BREAK CHARACTER SEQUENCE

The Enhanced USART module has the capability of sending the special Break character sequences that are required by the LIN/J2602 bus standard. The Break character transmit consists of a Start bit, followed by twelve '0' bits and a Stop bit. The Frame Break character is sent whenever the SENDB and TXEN bits (TXSTA<3> and TXSTA<5>) are set while the Transmit Shift register is loaded with data. Note that the value of data written to TXREG will be ignored and all '0's will be transmitted.

The SENDB bit is automatically reset by hardware after the corresponding Stop bit is sent. This allows the user to preload the transmit FIFO with the next transmit byte following the Break character (typically, the Sync character in the LIN/J2602 specification).

Note that the data value written to the TXREG for the Break character is ignored. The write simply serves the purpose of initiating the proper sequence.

The TRMT bit indicates when the transmit operation is active or Idle, just as it does during normal transmission. See Figure 20-9 for the timing of the Break character sequence.

20.3.5.1 Break and Sync Transmit Sequence

The following sequence will send a message frame header made up of a Break, followed by an Auto-Baud Sync byte. This sequence is typical of a LIN/J2602 bus master.

- 1. Configure the EUSART for the desired mode.
- 2. Set the TXEN and SENDB bits to setup the Break character.
- 3. Load the TXREG with a dummy character to initiate transmission (the value is ignored).
- 4. Write '55h' to TXREG to load the Sync character into the transmit FIFO buffer.
- 5. After the Break has been sent, the SENDB bit is reset by hardware. The Sync character now transmits in the preconfigured mode.

When the TXREG becomes empty, as indicated by the TXIF, the next data byte can be written to TXREG.

20.3.6 RECEIVING A BREAK CHARACTER

The Enhanced USART module can receive a Break character in two ways.

The first method forces configuration of the baud rate at a frequency of 9/13 of the typical speed. This allows for the Stop bit transition to be at the correct sampling location (13 bits for Break versus Start bit and 8 data bits for typical data).

The second method uses the auto-wake-up feature described in Section 20.3.4 "Auto-Wake-up on Sync Break Character". By enabling this feature, the EUSART will sample the next two transitions on RX/DT, cause an RCIF interrupt and receive the next data byte followed by another interrupt.

Note that following a Break character, the user will typically want to enable the Auto-Baud Rate Detect feature. For both methods, the user can set the ABD bit before placing the EUSART in its Sleep mode.

FIGURE 20-9: SEND BREAK CHARACTER SEQUENCE



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 unp	rogrammed	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 unp	programmed	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.

23.5.1 PROGRAM MEMORY CODE PROTECTION

The program memory may be read to, or written from, any location using the table read and table write instructions. The Device ID may be read with table reads. The Configuration registers may be read and written with the table read and table write instructions.

In normal execution mode, the CPn bits have no direct effect. CPn bits inhibit external reads and writes. A block of user memory may be protected from table writes if the WRTn Configuration bit is '0'. The EBTRn bits control table reads. For a block of user memory with the EBTRn bit set to '0', a table read instruction that executes from within that block is allowed to read. A table read instruction that executes from a location outside of that block is not allowed to read, and will result in reading '0's. Figures 23-6 through 23-8 illustrate table write and table read protection. **Note:** Code protection bits may only be written to a '0' from a '1' state. It is not possible to write a '1' to a bit in the '0' state. Code protection bits are only set to '1' by a full chip erase or block erase function. The full chip erase and block erase functions can only be initiated via ICSP or an external programmer.

FIGURE 23-6: TABLE WRITE (WRTn) DISALLOWED



PIC18F2331/2431/4331/4431



Param No.	Symbol	Characteristic	Min	Max	Units	Conditions	
70	TssL2scH, TssL2scL	$\overline{SS} \downarrow$ to SCK \downarrow or SCK \uparrow Input		Тсү		ns	
71	TscH	SCK Input High Time	Continuous	1.25 Tcy + 30	_	ns	
71A			Single byte	40		ns	(Note 1)
72	TscL	SCK Input Low Time	Continuous	1.25 Tcy + 30	_	ns	
72A			Single byte	40		ns	(Note 1)
73A	Тв2в	Last Clock Edge of Byte 1 to the First Cl	ock Edge of Byte 2	1.5 Tcy + 40	_	ns	(Note 2)
74	TscH2diL, TscL2diL	Hold Time of SDI Data Input to SCK E	40	_	ns		
75	TdoR	SDO Data Output Rise Time	PIC18FXX31	—	25	ns	
			PIC18LFXX31	—	45	ns	
76	TdoF	SDO Data Output Fall Time		—	25	ns	
77	TssH2doZ	\overline{SS} \uparrow to SDO Output High-Impedance		10	50	ns	
80	TscH2doV,	SDO Data Output Valid after SCK	PIC18FXX31	—	50	ns	
	TscL2doV	Edge	PIC18LFXX31	—	100	ns	
82	TssL2doV	SDO Data Output Valid after $\overline{ ext{SS}}\downarrow$	PIC18FXX31	—	50	ns	
		Edge	PIC18LFXX31	—	100	ns	
83	TscH2ssH, TscL2ssH	SS ↑ after SCK Edge		1.5 Tcy + 40	—	ns	

TABLE 26-14: EXAMPLE SPI SLAVE MODE REQUIREMENTS (CKE = 1)

Note 1: Requires the use of Parameter 73A.

2: Only if Parameter 71A and 72A are used.

NOTES:

28-Lead Plastic Small Outline (SO) - Wide, 7.50 mm Body [SOIC]

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



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	Е	1.27 BSC		
Contact Pad Spacing	С		9.40	
Contact Pad Width (X28)	Х			0.60
Contact Pad Length (X28)	Y			2.00
Distance Between Pads	Gx	0.67		
Distance Between Pads	G	7 <u>.</u> 40		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

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

Microchip Technology Drawing No. C04-2052A