



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	UART/USART
Peripherals	Brown-out Detect/Reset, LVD, POR, PWM, WDT
Number of I/O	16
Program Memory Size	8KB (4K x 16)
Program Memory Type	FLASH
EEPROM Size	128 x 8
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	4.2V ~ 5.5V
Data Converters	A/D 4x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	18-SOIC (0.295", 7.50mm Width)
Supplier Device Package	18-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f1330t-i-so

TABLE 1-1: DEVICE FEATURES

Features	PIC18F1230	PIC18F1330
Operating Frequency	DC – 40 MHz	DC – 40 MHz
Program Memory (Bytes)	4096	8192
Program Memory (Instructions)	2048	4096
Data Memory (Bytes)	256	256
Data EEPROM Memory (Bytes)	128	128
Interrupt Sources	17	17
I/O Ports	Ports A, B	Ports A, B
Timers	2	2
Power Control PWM Module	6 Channels	6 Channels
Serial Communications	Enhanced USART	Enhanced USART
10-Bit Analog-to-Digital Module	4 Input Channels	4 Input Channels
Resets (and Delays)	POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST), MCLR (optional), WDT	POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST), MCLR (optional), WDT
Programmable Low-Voltage Detect	Yes	Yes
Programmable Brown-out Reset	Yes	Yes
Instruction Set	75 Instructions; 83 with Extended Instruction Set enabled	75 Instructions; 83 with Extended Instruction Set enabled
Packages	18-Pin PDIP 18-Pin SOIC 20-Pin SSOP 28-Pin QFN	18-Pin PDIP 18-Pin SOIC 20-Pin SSOP 28-Pin QFN

TABLE 1-2: PIC18F1230/1330 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number			Pin Type	Buffer Type	Description
	PDIP, SOIC	SSOP	QFN			
RB0/PWM0 RB0 PWM0	8	9	9	I/O O	TTL —	PORTB is a bidirectional I/O port. Digital I/O. PWM module output PWM0.
RB1/PWM1 RB1 PWM1	9	10	10	I/O O	TTL —	Digital I/O. PWM module output PWM1.
RB2/INT2/KBI2/CMP2/ T1OSO/T1CKI RB2 INT2 KBI2 CMP2 T1OSO ⁽²⁾ T1CKI ⁽²⁾	17	19	23	I/O I I I O I	TTL ST TTL Analog — ST	Digital I/O. External interrupt 2. Interrupt-on-change pin. Comparator 2 input. Timer1 oscillator output. Timer1 clock input.
RB3/INT3/KBI3/CMP1/ T1OSI RB3 INT3 KBI3 CMP1 T1OSI ⁽²⁾	18	20	24	I/O I I I I	TTL ST TTL Analog Analog	Digital I/O. External interrupt 3. Interrupt-on-change pin. Comparator 1 input. Timer1 oscillator input.
RB4/PWM2 RB4 PWM2	10	11	12	I/O O	TTL —	Digital I/O. PWM module output PWM2.
RB5/PWM3 RB5 PWM3	11	12	13	I/O O	TTL —	Digital I/O. PWM module output PWM3.
RB6/PWM4/PGC RB6 PWM4 PGC	12	13	15	I/O O I	TTL — ST	Digital I/O. PWM module output PWM4. In-Circuit Debugger and ICSP™ programming clock pin.
RB7/PWM5/PGD RB7 PWM5 PGD	13	14	16	I/O O O	TTL — —	Digital I/O. PWM module output PWM5. In-Circuit Debugger and ICSP programming data 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

Note 1: Placement of FLTA depends on the value of Configuration bit, FLTAMX, of CONFIG3H.

2: Placement of T1OSI and T1OSO/T1CKI depends on the value of Configuration bit, T1OSCMX, of CONFIG3H.

2.2 Power Supply Pins

2.2.1 DECOUPLING CAPACITORS

The use of decoupling capacitors on every pair of power supply pins, such as VDD, VSS, AVDD and AVSS, is required.

Consider the following criteria when using decoupling capacitors:

- **Value and type of capacitor:** A 0.1 μF (100 nF), 10-20V capacitor is recommended. The capacitor should be a low-ESR device, with a resonance frequency in the range of 200 MHz and higher. Ceramic capacitors are recommended.
- **Placement on the printed circuit board:** The decoupling capacitors should be placed as close to the pins as possible. It is recommended to place the capacitors on the same side of the board as the device. If space is constricted, the capacitor can be placed on another layer on the PCB using a via; however, ensure that the trace length from the pin to the capacitor is no greater than 0.25 inch (6 mm).
- **Handling high-frequency noise:** If the board is experiencing high-frequency noise (upward of tens of MHz), add a second ceramic type capacitor in parallel to the above described decoupling capacitor. The value of the second capacitor can be in the range of 0.01 μF to 0.001 μF . Place this second capacitor next to each primary decoupling capacitor. In high-speed circuit designs, consider implementing a decade pair of capacitances as close to the power and ground pins as possible (e.g., 0.1 μF in parallel with 0.001 μF).
- **Maximizing performance:** On the board layout from the power supply circuit, run the power and return traces to the decoupling capacitors first, and then to the device pins. This ensures that the decoupling capacitors are first in the power chain. Equally important is to keep the trace length between the capacitor and the power pins to a minimum, thereby reducing PCB trace inductance.

2.2.2 TANK CAPACITORS

On boards with power traces running longer than six inches in length, it is suggested to use a tank capacitor for integrated circuits, including microcontrollers, to supply a local power source. The value of the tank capacitor should be determined based on the trace resistance that connects the power supply source to the device, and the maximum current drawn by the device in the application. In other words, select the tank capacitor so that it meets the acceptable voltage sag at the device. Typical values range from 4.7 μF to 47 μF .

2.2.3 CONSIDERATIONS WHEN USING BOR

When the Brown-out Reset (BOR) feature is enabled, a sudden change in VDD may result in a spontaneous BOR event. This can happen when the microcontroller is operating under normal operating conditions, regardless of what the BOR set point has been programmed to, and even if VDD does not approach the set point. The precipitating factor in these BOR events is a rise or fall in VDD with a slew rate faster than 0.15V/ μs .

An application that incorporates adequate decoupling between the power supplies will not experience such rapid voltage changes. Additionally, the use of an electrolytic tank capacitor across VDD and VSS, as described above, will be helpful in preventing high slew rate transitions.

If the application has components that turn on or off, and share the same VDD circuit as the microcontroller, the BOR can be disabled in software by using the SBOREN bit before switching the component. Afterwards, allow a small delay before re-enabling the BOR. By doing this, it is ensured that the BOR is disabled during the interval that might cause high slew rate changes of VDD.

Note: Not all devices incorporate software BOR control. See Section 5.0 “Reset” for device-specific information.
--

4.0 POWER-MANAGED MODES

PIC18F1230/1330 devices offer a total of seven operating modes for more efficient power management. These modes provide a variety of options for selective power conservation in applications where resources may be limited (i.e., battery-powered devices).

There are three categories of power-managed modes:

- Run modes
- Idle modes
- Sleep mode

These categories define which portions of the device are clocked and sometimes, what speed. The Run and Idle modes may use any of the three available clock sources (primary, secondary or internal oscillator block); the Sleep mode does not use a clock source.

The power-managed modes include several power-saving features offered on previous PIC® devices. One is the clock switching feature, offered in other PIC18 devices, allowing the controller to use the Timer1 oscillator in place of the primary oscillator. Also included is the Sleep mode, offered by all PIC devices, where all device clocks are stopped.

4.1 Selecting Power-Managed Modes

Selecting a power-managed mode requires two decisions: if the CPU is to be clocked or not and the selection of a clock source. The IDLEN bit (OSCCON<7>) controls CPU clocking, while the SCS1:SCS0 bits (OSCCON<1:0>) select the clock source. The individual modes, bit settings, clock sources and affected modules are summarized in Table 4-1.

4.1.1 CLOCK SOURCES

The SCS1:SCS0 bits allow the selection of one of three clock sources for power-managed modes. They are:

- the primary clock, as defined by the FOSC3:FOSC0 Configuration bits
- the secondary clock (the Timer1 oscillator)
- the internal oscillator block (for RC modes)

4.1.2 ENTERING POWER-MANAGED MODES

Switching from one power-managed mode to another begins by loading the OSCCON register. The SCS1:SCS0 bits select the clock source and determine which Run or Idle mode is to be used. Changing these bits causes an immediate switch to the new clock source, assuming that it is running. The switch may also be subject to clock transition delays. These are discussed in **Section 4.1.3 “Clock Transitions and Status Indicators”** and subsequent sections.

Entry to the power-managed Idle or Sleep modes is triggered by the execution of a SLEEP instruction. The actual mode that results depends on the status of the IDLEN bit.

Depending on the current mode and the mode being switched to, a change to a power-managed mode does not always require setting all of these bits. Many transitions may be done by changing the oscillator select bits, or changing the IDLEN bit, prior to issuing a SLEEP instruction. If the IDLEN bit is already configured correctly, it may only be necessary to perform a SLEEP instruction to switch to the desired mode.

TABLE 4-1: POWER-MANAGED MODES

Mode	OSCCON Bits		Module Clocking		Available Clock and Oscillator Source
	IDLEN<7> ⁽¹⁾	SCS1:SCS0<1:0>	CPU	Peripherals	
Sleep	0	N/A	Off	Off	None – All clocks are disabled
PRI_RUN	N/A	00	Clocked	Clocked	Primary – LP, XT, HS, HSPLL, RC, EC and Internal Oscillator Block ⁽²⁾ . This is the normal full power execution mode.
SEC_RUN	N/A	01	Clocked	Clocked	Secondary – Timer1 Oscillator
RC_RUN	N/A	1x	Clocked	Clocked	Internal Oscillator Block ⁽²⁾
PRI_IDLE	1	00	Off	Clocked	Primary – LP, XT, HS, HSPLL, RC, EC
SEC_IDLE	1	01	Off	Clocked	Secondary – Timer1 Oscillator
RC_IDLE	1	1x	Off	Clocked	Internal Oscillator Block ⁽²⁾

Note 1: IDLEN reflects its value when the SLEEP instruction is executed.

2: Includes INTOSC and INTOSC postscaler, as well as the INTRC source.

10.0 I/O PORTS

Depending on the device selected and features enabled, there are up to five ports available. Some pins of the I/O ports are multiplexed with an alternate function from the peripheral features on the device. In general, when a peripheral is enabled, that pin may not be used as a general purpose I/O pin.

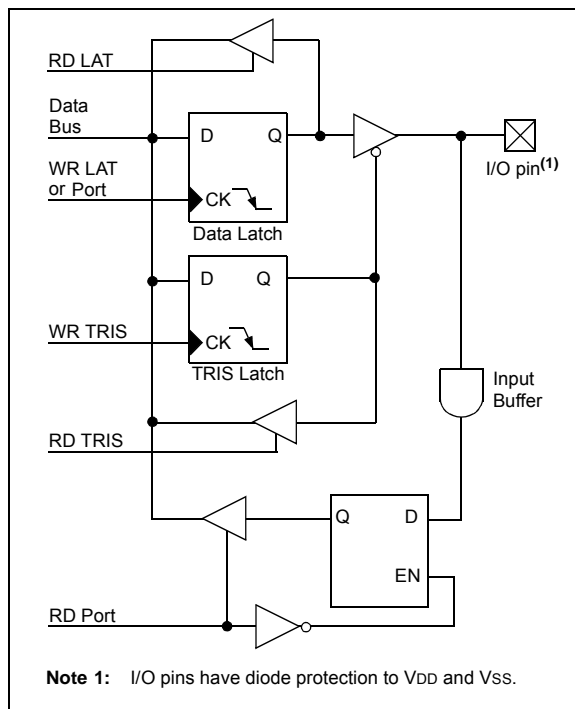
Each port has three registers for its operation. These registers are:

- TRIS register (Data Direction register)
- PORT register (reads the levels on the pins of the device)
- LAT register (Output Latch register)

The Output Latch (LAT register) is useful for read-modify-write operations on the value that the I/O pins are driving.

A simplified model of a generic I/O port, without the interfaces to other peripherals, is shown in Figure 10-1.

FIGURE 10-1: GENERIC I/O PORT OPERATION



10.1 PORTA, TRISA and LATA Registers

PORTA is an 8-bit wide, bidirectional port. The corresponding Data Direction register is TRISA. Setting a TRISA bit (= 1) will make the corresponding PORTA pin an input (i.e., put the corresponding output driver in a high-impedance mode). Clearing a TRISA bit (= 0) will make the corresponding PORTA pin an output (i.e., put the contents of the output latch on the selected pin).

Reading the PORTA register reads the status of the pins, whereas writing to it, will write to the port latch.

The Output Latch (LATA) register is also memory mapped. Read-modify-write operations on the LATA register read and write the latched output value for PORTA.

Pins RA6 and RA7 are multiplexed with the main oscillator pins; they are enabled as oscillator or I/O pins by the selection of the main oscillator in the Configuration register (see **Section 20.1 “Configuration Bits”** for details). When they are not used as port pins, RA6 and RA7 and their associated TRIS and LAT bits are read as '0'.

The RA0 pin is multiplexed with one of the analog inputs, one of the external interrupt inputs, one of the interrupt-on-change inputs and one of the analog comparator inputs to become RA0/AN0/INT0/KBI0/CMP0 pin.

The RA1 pin is multiplexed with one of the analog inputs, one of the external interrupt inputs and one of the interrupt-on-change inputs to become RA1/AN1/INT1/KBI1 pin.

Pins RA2 and RA3 are multiplexed with the Enhanced USART transmission and reception input (see **Section 20.1 “Configuration Bits”** for details).

The RA4 pin is multiplexed with the Timer0 module clock input, one of the analog inputs and the analog VREF+ input to become the RA4/T0CKI/AN2/VREF+ pin.

The Fault detect input for PWM \overline{FLTA} is multiplexed with pins RA5 and RA7. Its placement is decided by clearing or setting the FLTAMX bit of Configuration Register 3H.

Note: On a Power-on Reset, RA0, RA1, RA4 and RA5 are configured as analog inputs and read as '0'. RA2 and RA3 are configured as digital inputs.

The TRISA register controls the direction of the PORTA pins, even when they are being used as analog inputs. The user must ensure the bits in the TRISA register are maintained set when using them as analog inputs.

EXAMPLE 10-1: INITIALIZING PORTA

```
CLRF    PORTA    ; Initialize PORTA by
                ; clearing output
                ; data latches
CLRF    LATA     ; Alternate method
                ; to clear output
                ; data latches
MOVLW   07h     ; Configure A/D
MOVWF   ADCON1  ; for digital inputs
MOVWF   07h     ; Configure comparators
MOVWF   CMCON   ; for digital input
MOVLW   0CFh    ; Value used to
                ; initialize data
                ; direction
MOVWF   TRISA   ; Set RA<7:6,3:0> as inputs
                ; RA<5:4> as outputs
```

10.2 PORTB, TRISB and LATB Registers

PORTB is an 8-bit wide, bidirectional port. The corresponding Data Direction register is TRISB. Setting a TRISB bit (= 1) will make the corresponding PORTB pin an input (i.e., put the corresponding output driver in a high-impedance mode). Clearing a TRISB bit (= 0) will make the corresponding PORTB pin an output (i.e., put the contents of the output latch on the selected pin).

The Output Latch register (LATB) is also memory mapped. Read-modify-write operations on the LATB register read and write the latched output value for PORTB.

EXAMPLE 10-2: INITIALIZING PORTB

```
CLRF    PORTB    ; Initialize PORTB by
                  ; clearing output
                  ; data latches
CLRF    LATB      ; Alternate method
                  ; to clear output
                  ; data latches
MOVLW   0Fh      ; Set RB<4:0> as
MOVWF   ADCON1    ; digital I/O pins
                  ; (required if config bit
                  ; PBDEN is set)
MOVLW   0CFh      ; Value used to
                  ; initialize data
                  ; direction
MOVWF   TRISB     ; Set RB<3:0> as inputs
                  ; RB<5:4> as outputs
                  ; RB<7:6> as inputs
```

Each of the PORTB pins has a weak internal pull-up. A single control bit can turn on all the pull-ups. This is performed by clearing bit, $\overline{\text{RPU}}$ (INTCON2<7>). The weak pull-up is automatically turned off when the port pin is configured as an output. The pull-ups are disabled on a Power-on Reset.

Note: On a Power-on Reset, PORTB is configured as digital inputs except for RB2 and RB3.

RB2 and RB3 are configured as analog inputs when the T1OSCMX bit of Configuration Register 3H is cleared. Otherwise, RB2 and RB3 are also configured as digital inputs.

Pins RB0, RB1 and RB4:RB7 are multiplexed with the Power Control PWM outputs.

Pins RB2 and RB3 are multiplexed with external interrupt inputs, interrupt-on-change input, the analog comparator inputs and the Timer1 oscillator input and output to become RB2/INT2/KBI2/CMP2/T1OSO/T1CKI and RB3/INT3/KNBI3/CMP1/T1OSI, respectively.

When the interrupt-on-change feature is enabled, only pins configured as inputs can cause this interrupt to occur (i.e., any RB2, RB3, RA0 and RA1 pin configured as an output is excluded from the interrupt-on-change comparison). The input pins (RB2, RB3, RA0 and RA1) are compared with the old value latched on the last read of PORTA and PORTB. The “mismatch” outputs of these pins are ORed together to generate the RB Port Change Interrupt with Flag bit, RBIF (INTCON<0>).

This interrupt can wake the device from Sleep mode, or any of the Idle modes. The user, in the Interrupt Service Routine, can clear the interrupt in the following manner:

- Any read or write of PORTB (except with the MOVFF (ANY), PORTB instruction).
- 1 Tcy
- Clear flag bit, RBIF.

A mismatch condition will continue to set flag bit, RBIF. Reading PORTB and waiting 1 Tcy will end the mismatch condition and allow flag bit, RBIF, to be cleared. Additionally, if the port pin returns to its original state, the mismatch condition will be cleared.

The interrupt-on-change feature is recommended for wake-up on key depression operation and operations where PORTA and PORTB are used for the interrupt-on-change feature. Polling of PORTA and PORTB is not recommended while using the interrupt-on-change feature.

PIC18F1230/1330

FIGURE 12-1: TIMER0 BLOCK DIAGRAM IN 8-BIT MODE

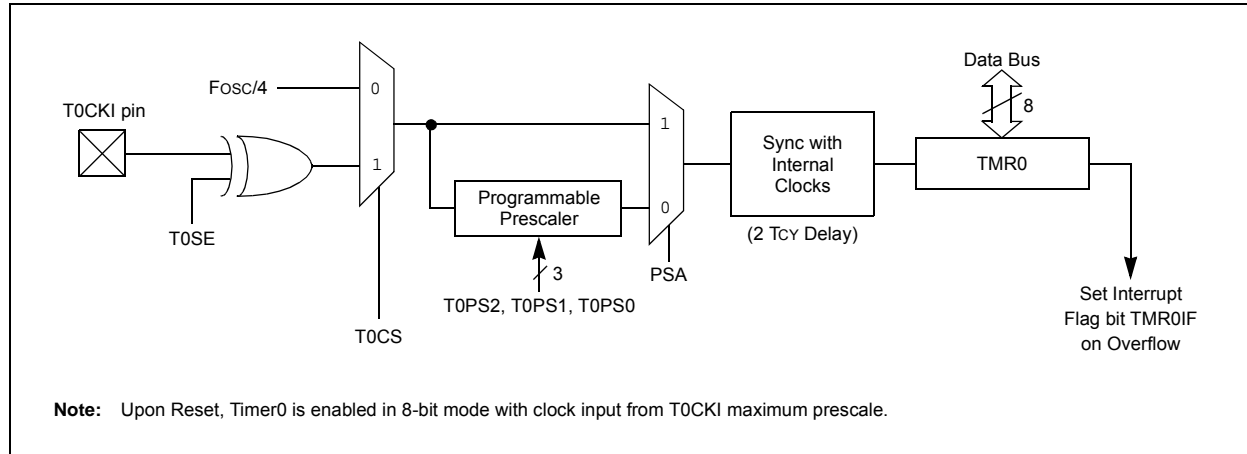
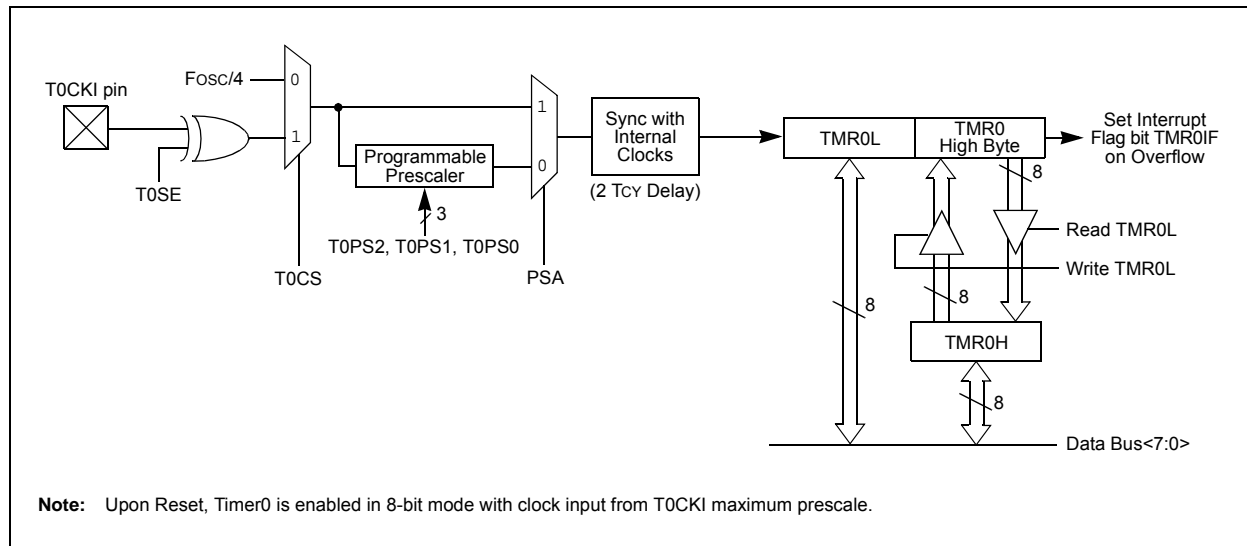


FIGURE 12-2: TIMER0 BLOCK DIAGRAM IN 16-BIT MODE



13.4 Timer1 Interrupt

The TMR1 register pair (TMR1H:TMR1L) increments from 0000h to FFFFh and rolls over to 0000h. The Timer1 interrupt, if enabled, is generated on overflow which is latched in interrupt flag bit, TMR1IF (PIR1<0>). This interrupt can be enabled/disabled by setting/clearing Timer1 interrupt enable bit, TMR1IE (PIE1<0>).

13.5 Timer1 16-Bit Read/Write Mode

Timer1 can be configured for 16-bit reads and writes (see Figure 13-2). When the RD16 control bit (T1CON<7>) is set, the address for TMR1H is mapped to a buffer register for the high byte of Timer1. A read from TMR1L will load the contents of the high byte of Timer1 into the Timer1 High Byte Buffer register. This provides the user with the ability to accurately read all 16 bits of Timer1 without having to determine whether a read of the high byte, followed by a read of the low byte, is valid due to a rollover between reads.

A write to the high byte of Timer1 must also take place through the TMR1H Buffer register. Timer1 high byte is updated with the contents of TMR1H when a write occurs to TMR1L. This allows a user to write all 16 bits to both the high and low bytes of Timer1 at once.

The high byte of Timer1 is not directly readable or writable in this mode. All reads and writes must take place through the Timer1 High Byte Buffer register. Writes to TMR1H do not clear the Timer1 prescaler. The prescaler is only cleared on writes to TMR1L.

13.6 Using Timer1 as a Real-Time Clock

Adding an external LP oscillator to Timer1 (such as the one described in **Section 13.2 “Timer1 Oscillator”**), gives users the option to include RTC functionality to their applications. This is accomplished with an inexpensive watch crystal to provide an accurate time base and several lines of application code to calculate the time. When operating in Sleep mode and using a battery or super capacitor as a power source, it can completely eliminate the need for a separate RTC device and battery backup.

The application code routine, *RTCisr*, shown in Example 13-1, demonstrates a simple method to increment a counter at one-second intervals using an Interrupt Service Routine. Incrementing the TMR1 register pair to overflow triggers the interrupt and calls the routine, which increments the seconds counter by one. Additional counters for minutes and hours are incremented as the previous counter overflow.

Since the register pair is 16 bits wide, counting up to overflow the register directly from a 32.768 kHz clock would take 2 seconds. To force the overflow at the required one-second intervals, it is necessary to preload it. The simplest method is to set the MSb of TMR1H with a *BSF* instruction. Note that the TMR1L register is never preloaded or altered; doing so may introduce cumulative error over many cycles.

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

REGISTER 14-3: PWMCON0: PWM CONTROL REGISTER 0

U-0	R/W-1 ⁽¹⁾	R/W-1 ⁽¹⁾	R/W-1 ⁽¹⁾	U-0	R/W-0	R/W-0	R/W-0
—	PWMEN2	PWMEN1	PWMEN0	—	PMOD2	PMOD1	PMOD0
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 **Unimplemented:** Read as '0'

bit 6-4 **PWMEN2:PWMEN0:** PWM Module Enable bits⁽¹⁾

111 = All odd PWM I/O pins enabled for PWM output

110 = PWM1, PWM3 pins enabled for PWM output

10x = All PWM I/O pins enabled for PWM output

011 = PWM0, PWM1, PWM2 and PWM3 I/O pins enabled for PWM output

010 = PWM0 and PWM1 pins enabled for PWM output

001 = PWM1 pin is enabled for PWM output

000 = PWM module disabled; all PWM I/O pins are general purpose I/O

bit 3 **Unimplemented:** Read as '0'

bit 2-0 **PMOD2:PMOD0:** PWM Output Pair Mode bits

For PMOD0:

1 = PWM I/O pin pair (PWM0, PWM1) is in the Independent mode

0 = PWM I/O pin pair (PWM0, PWM1) is in the Complementary mode

For PMOD1:

1 = PWM I/O pin pair (PWM2, PWM3) is in the Independent mode

0 = PWM I/O pin pair (PWM2, PWM3) is in the Complementary mode

For PMOD2:

1 = PWM I/O pin pair (PWM4, PWM5) is in the Independent mode

0 = PWM I/O pin pair (PWM4, PWM5) is in the Complementary mode

Note 1: Reset condition of PWMEN bits depends on the PWMPIN Configuration bit of CONFIG3L.

14.12.2 FAULT INPUT MODE

The FLTAMOD bit in the FLTCONFIG register determines whether the PWM I/O pins are deactivated when they are overridden by a Fault input.

FLTAS bit in the FLTCONFIG register gives the status of the Fault A input.

The Fault input has two modes of operation:

- **Inactive Mode (FLTAMOD = 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 Inactivated 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 Fault status bit (FLTAS) is cleared.

- **Cycle-by-Cycle Mode (FLTAMOD = 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 FLTAS bit is automatically cleared.

14.12.3 PWM OUTPUTS WHILE IN FAULT CONDITION

While in the Fault state (i.e., $\overline{\text{FLTA}}$ input is active), the PWM output signals are driven into their inactive states.

14.12.4 PWM OUTPUTS IN DEBUG MODE

The BRFFEN bit in the FLTCONFIG register controls the simulation of Fault condition when a breakpoint is hit, while debugging the application using an In-Circuit Debugger (ICD). Setting the BRFFEN bit to high enables the Fault condition on breakpoint, thus driving the PWM outputs to inactive state. This is done to avoid any continuous keeping of status on the PWM pin, which may result in damage of the power devices connected to the PWM outputs.

If BRFFEN = 0, the Fault condition on breakpoint is disabled.

Note: It is highly recommended to enable the Fault condition on breakpoint if a debugging tool is used while developing the firmware and the high-power circuitry is used. When the device is ready to program after debugging the firmware, the BRFFEN bit can be disabled.

REGISTER 14-8: FLTCONFIG: FAULT CONFIGURATION REGISTER

R/W-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
BRFFEN	—	—	—	—	FLTAS	FLTAMOD	FLTAEN
bit 7							bit 0

Legend:

R = Readable bit
-n = Value at POR

W = Writable bit
'1' = Bit is set

U = Unimplemented bit, read as '0'
'0' = Bit is cleared
x = Bit is unknown

bit 7	BRFFEN: Breakpoint Fault Enable bit 1 = Enable Fault condition on a breakpoint 0 = Disable Fault condition
bit 6-3	Unimplemented: Read as '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

PIC18F1230/1330

NOTES:

REGISTER 15-2: RCSTA: RECEIVE STATUS AND CONTROL REGISTER

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

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	SPEN: Serial Port Enable bit 1 = Serial port enabled (configures RX/DT and TX/CK pins as serial port pins) 0 = Serial port disabled (held in Reset)
bit 6	RX9: 9-Bit Receive Enable bit 1 = Selects 9-bit reception 0 = Selects 8-bit reception
bit 5	SREN: Single Receive Enable bit <u>Asynchronous mode:</u> Don't care. <u>Synchronous mode – Master:</u> 1 = Enables single receive 0 = Disables single receive This bit is cleared after reception is complete. <u>Synchronous mode – Slave:</u> Don't care.
bit 4	CREN: Continuous Receive Enable bit <u>Asynchronous mode:</u> 1 = Enables receiver 0 = Disables receiver <u>Synchronous mode:</u> 1 = Enables continuous receive until enable bit CREN is cleared (CREN overrides SREN) 0 = Disables continuous receive
bit 3	ADDEN: Address Detect Enable bit <u>Asynchronous mode 9-bit (RX9 = 1):</u> 1 = Enables address detection, enables interrupt and loads the receive buffer when RSR<8> is set 0 = Disables address detection, all bytes are received and ninth bit can be used as parity bit <u>Asynchronous mode 9-bit (RX9 = 0):</u> Don't care.
bit 2	FERR: Framing Error bit 1 = Framing error (can be updated by reading RCREG register and receiving next valid byte) 0 = No framing error
bit 1	OERR: Overrun Error bit 1 = Overrun error (can be cleared by clearing bit CREN) 0 = No overrun error
bit 0	RX9D: 9th bit of Received Data This can be address/data bit or a parity bit and must be calculated by user firmware.

15.2 EUSART Asynchronous Mode

The Asynchronous mode of operation is selected by clearing the SYNC bit (TXSTA<4>). In this mode, the EUSART uses standard Non-Return-to-Zero (NRZ) format (one Start bit, eight or nine data bits and one Stop bit). The most common data format is 8 bits. An on-chip dedicated 8-bit/16-bit Baud Rate Generator can be used to derive standard baud rate frequencies from the oscillator.

The EUSART transmits and receives the LSb first. The EUSART's transmitter and receiver are functionally independent but use the same data format and baud rate. The Baud Rate Generator produces a clock, either x16 or x64 of the bit shift rate depending on the BRGH and BRG16 bits (TXSTA<2> and BAUDCON<3>). Parity is not supported by the hardware but can be implemented in software and stored as the 9th data bit.

In Asynchronous mode, clock polarity is selected with the TXCKP bit (BAUDCON<4>). Setting TXCKP sets the Idle state on CK as high, while clearing the bit sets the Idle state as low. Data polarity is selected with the RXDTP bit (BAUDCON<5>).

Setting RXDTP inverts data on RX, while clearing the bit has no effect on received data.

When operating in Asynchronous mode, the EUSART module consists of the following important elements:

- Baud Rate Generator
- Sampling Circuit
- Asynchronous Transmitter
- Asynchronous Receiver
- Auto-Wake-up on Sync Break Character
- 12-Bit Break Character Transmit
- Auto-Baud Rate Detection

15.2.1 EUSART ASYNCHRONOUS TRANSMITTER

The EUSART transmitter block diagram is shown in Figure 15-3. The heart of the transmitter is the Transmit (Serial) Shift Register (TSR). The Shift register obtains its data from the Read/Write Transmit Buffer register, TXREG. The TXREG register is loaded with data in software. The TSR register is not loaded until the Stop bit has been transmitted from the previous load. As soon as the Stop bit is transmitted, the TSR is loaded with new data from the TXREG register (if available).

Once the TXREG register transfers the data to the TSR register (occurs in one Tcy), the TXREG register is empty and the TXIF flag bit (PIR1<4>) is set. This interrupt can be enabled or disabled by setting or clearing the interrupt enable bit, TXIE (PIE1<4>). TXIF will be set regardless of the state of TXIE; it cannot be cleared in software. TXIF is also not cleared immediately upon loading TXREG but becomes valid in the second instruction cycle following the load instruction. Polling TXIF immediately following a load of TXREG will return invalid results.

While TXIF indicates the status of the TXREG register, another bit, TRMT (TXSTA<1>), shows the status of the TSR register. TRMT is a read-only bit which is set when the TSR register is empty. No interrupt logic is tied to this bit so the user has to poll this bit in order to determine if the TSR register is empty.

Note 1: The TSR register is not mapped in data memory so it is not available to the user.

2: Flag bit TXIF is set when enable bit TXEN is set.

To set up an Asynchronous Transmission:

1. Initialize the SPBRGH:SPBRG registers for the appropriate baud rate. Set or clear the BRGH and BRG16 bits, as required, to achieve the desired baud rate.
2. Enable the asynchronous serial port by clearing bit SYNC and setting bit SPEN.
3. If interrupts are desired, set enable bit TXIE.
4. If 9-bit transmission is desired, set transmit bit TX9. Can be used as address/data bit.
5. Enable the transmission by setting bit TXEN which will also set bit TXIF.
6. If 9-bit transmission is selected, the ninth bit should be loaded in bit TX9D.
7. Load data to the TXREG register (starts transmission).
8. If using interrupts, ensure that the GIE and PEIE bits in the INTCON register (INTCON<7:6>) are set.

The value in the ADRESH:ADRESL registers is not modified for a Power-on Reset. The ADRESH:ADRESL registers will contain unknown data after a Power-on Reset.

After the A/D module has been configured as desired, the selected channel must be acquired before the conversion is started. The analog input channels must have their corresponding TRIS bits selected as inputs. To determine acquisition time, see **Section 16.2 “A/D Acquisition Requirements”**. After this acquisition time has elapsed, the A/D conversion can be started. An acquisition time can be programmed to occur between setting the GO/DONE bit and the actual start of the conversion.

The following steps should be followed to perform an A/D conversion:

1. Configure the A/D module:
 - Configure analog pins, voltage reference and digital I/O (ADCON1)
 - Select A/D input channel (ADCON0)
 - Select A/D acquisition time (ADCON2)
 - Select A/D conversion clock (ADCON2)
 - Turn on A/D module (ADCON0)
2. Configure A/D interrupt (if desired):
 - Clear ADIF bit
 - Set ADIE bit
 - Set GIE bit
3. Wait the required acquisition time (if required).
4. Start conversion:
 - Set GO/DONE bit (ADCON0 register)

5. Wait for A/D conversion to complete, by either:
 - Polling for the GO/DONE bit to be cleared
 - OR
 - Waiting for the A/D interrupt
6. Read A/D Result registers (ADRESH:ADRESL); clear bit ADIF, if required.
7. For next conversion, go to step 1 or step 2, as required. The A/D conversion time per bit is defined as T_{AD} . A minimum wait of 2 T_{AD} is required before the next acquisition starts.

FIGURE 16-2: A/D TRANSFER FUNCTION

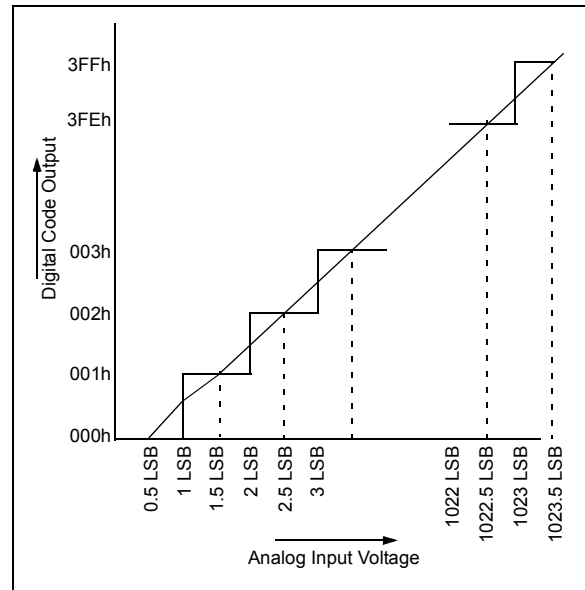
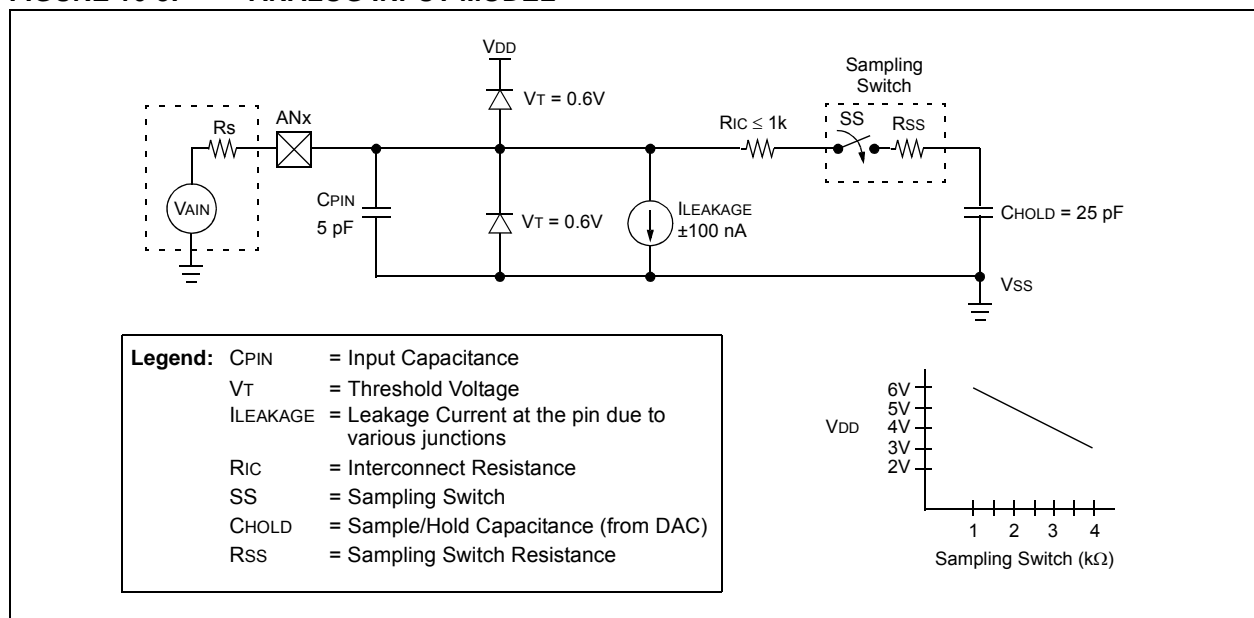


FIGURE 16-3: ANALOG INPUT MODEL



21.0 DEVELOPMENT SUPPORT

The PIC® microcontrollers are supported with a full range of hardware and software development tools:

- Integrated Development Environment
 - MPLAB® IDE Software
- Assemblers/Compilers/Linkers
 - MPASM™ Assembler
 - MPLAB C18 and MPLAB C30 C Compilers
 - MPLINK™ Object Linker/
MPLIB™ Object Librarian
 - MPLAB ASM30 Assembler/Linker/Library
- Simulators
 - MPLAB SIM Software Simulator
- Emulators
 - MPLAB ICE 2000 In-Circuit Emulator
 - MPLAB REAL ICE™ In-Circuit Emulator
- In-Circuit Debugger
 - MPLAB ICD 2
- Device Programmers
 - PICSTART® Plus Development Programmer
 - MPLAB PM3 Device Programmer
 - PICKit™ 2 Development Programmer
- Low-Cost Demonstration and Development Boards and Evaluation Kits

21.1 MPLAB Integrated Development Environment Software

The MPLAB IDE software brings an ease of software development previously unseen in the 8/16-bit microcontroller market. The MPLAB IDE is a Windows® operating system-based application that contains:

- A single graphical interface to all debugging tools
 - Simulator
 - Programmer (sold separately)
 - Emulator (sold separately)
 - In-Circuit Debugger (sold separately)
- A full-featured editor with color-coded context
- A multiple project manager
- Customizable data windows with direct edit of contents
- High-level source code debugging
- Visual device initializer for easy register initialization
- Mouse over variable inspection
- Drag and drop variables from source to watch windows
- Extensive on-line help
- Integration of select third party tools, such as HI-TECH Software C Compilers and IAR C Compilers

The MPLAB IDE allows you to:

- Edit your source files (either assembly or C)
- One touch assemble (or compile) and download to PIC MCU emulator and simulator tools (automatically updates all project information)
- Debug using:
 - Source files (assembly or C)
 - Mixed assembly and C
 - Machine code

MPLAB IDE supports multiple debugging tools in a single development paradigm, from the cost-effective simulators, through low-cost in-circuit debuggers, to full-featured emulators. This eliminates the learning curve when upgrading to tools with increased flexibility and power.

TABLE 22-2: PIC18FXXXX INSTRUCTION SET (CONTINUED)

Mnemonic, Operands	Description	Cycles	16-Bit Instruction Word				Status Affected	Notes	
			MSb		LSb				
BIT-ORIENTED OPERATIONS									
BCF	f, b, a	Bit Clear f	1	1001	bbba	ffff	ffff	None	1, 2
BSF	f, b, a	Bit Set f	1	1000	bbba	ffff	ffff	None	1, 2
BTFSC	f, b, a	Bit Test f, Skip if Clear	1 (2 or 3)	1011	bbba	ffff	ffff	None	3, 4
BTFSS	f, b, a	Bit Test f, Skip if Set	1 (2 or 3)	1010	bbba	ffff	ffff	None	3, 4
BTG	f, d, a	Bit Toggle f	1	0111	bbba	ffff	ffff	None	1, 2
CONTROL OPERATIONS									
BC	n	Branch if Carry	1 (2)	1110	0010	nnnn	nnnn	None	4
BN	n	Branch if Negative	1 (2)	1110	0110	nnnn	nnnn	None	
BNC	n	Branch if Not Carry	1 (2)	1110	0011	nnnn	nnnn	None	
BNN	n	Branch if Not Negative	1 (2)	1110	0111	nnnn	nnnn	None	
BN OV	n	Branch if Not Overflow	1 (2)	1110	0101	nnnn	nnnn	None	
BNZ	n	Branch if Not Zero	1 (2)	1110	0001	nnnn	nnnn	None	
BOV	n	Branch if Overflow	1 (2)	1110	0100	nnnn	nnnn	None	
BRA	n	Branch Unconditionally	2	1101	0nnn	nnnn	nnnn	None	
BZ	n	Branch if Zero	1 (2)	1110	0000	nnnn	nnnn	None	
CALL	n, s	Call subroutine 1st word	2	1110	110s	kkkk	kkkk	None	
		2nd word		1111	kkkk	kkkk	kkkk		
CLRWDT	—	Clear Watchdog Timer	1	0000	0000	0000	0100	\overline{TO} , \overline{PD}	
DAW	—	Decimal Adjust WREG	1	0000	0000	0000	0111	C	
GOTO	n	Go to address 1st word	2	1110	1111	kkkk	kkkk	None	
		2nd word		1111	kkkk	kkkk	kkkk		
NOP	—	No Operation	1	0000	0000	0000	0000	None	
NOP	—	No Operation	1	1111	xxxx	xxxx	xxxx	None	
POP	—	Pop Top of Return Stack (TOS)	1	0000	0000	0000	0110	None	
PUSH	—	Push Top of Return Stack (TOS)	1	0000	0000	0000	0101	None	
RCALL	n	Relative Call	2	1101	1nnn	nnnn	nnnn	None	
RESET		Software Device Reset	1	0000	0000	1111	1111	All	
RETFIE	s	Return from Interrupt Enable	2	0000	0000	0001	000s	GIE/GIEH, PEIE/GIEL	
RETLW	k	Return with Literal in WREG	2	0000	1100	kkkk	kkkk	None	
RETURN	s	Return from Subroutine	2	0000	0000	0001	001s	None	
SLEEP	—	Go into Standby mode	1	0000	0000	0000	0011	\overline{TO} , \overline{PD}	

- Note 1:** When a PORT register is modified as a function of itself (e.g., `MOVF PORTB, 1, 0`), the value used will be that value present on the pins themselves. For example, if the data latch is '1' for a pin configured as input and is driven low by an external device, the data will be written back with a '0'.
- 2:** If this instruction is executed on the TMR0 register (and where applicable, 'd' = 1), the prescaler will be cleared if assigned.
- 3:** If the Program Counter (PC) is modified or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a NOP.
- 4:** Some instructions are two-word instructions. The second word of these instructions will be executed as a NOP unless the first word of the instruction retrieves the information embedded in these 16 bits. This ensures that all program memory locations have a valid instruction.

23.2 DC Characteristics: Power-Down and Supply Current PIC18F1230/1330 (Industrial) PIC18LF1230/1330 (Industrial) (Continued)

PIC18LF1230/1330 (Industrial)		Standard Operating Conditions (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for industrial					
PIC18F1230/1330 (Industrial, Extended)		Standard Operating Conditions (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for industrial $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ for extended					
Param No.	Device	Typ	Max	Units	Conditions		
	Supply Current (I _{DD}) ⁽²⁾						
	PIC18LF1230/1330	0.8	1.83	mA	-40°C	V _{DD} = 2.0V	FOSC = 4 MHz (RC_RUN mode, INTOSC source)
		0.8	1.83	mA	$+25^{\circ}\text{C}$		
		0.8	1.83	mA	$+85^{\circ}\text{C}$		
	PIC18LF1230/1330	1.3	2.93	mA	-40°C	V _{DD} = 3.0V	
		1.3	2.93	mA	$+25^{\circ}\text{C}$		
		1.3	2.93	mA	$+85^{\circ}\text{C}$		
	All devices	2.5	4.73	mA	-40°C	V _{DD} = 5.0V	
		2.5	4.73	mA	$+25^{\circ}\text{C}$		
		2.5	4.73	mA	$+85^{\circ}\text{C}$		
	Extended devices only	2.5	10.0	mA	$+125^{\circ}\text{C}$		
	PIC18LF1230/1330	2.9	7.6	μA	-40°C	V _{DD} = 2.0V	FOSC = 31 kHz (RC_IDLE mode, INTRC source)
		3.1	7.6	μA	$+25^{\circ}\text{C}$		
		3.6	10.6	μA	$+85^{\circ}\text{C}$		
	PIC18LF1230/1330	4.5	10.6	μA	-40°C	V _{DD} = 3.0V	
		4.8	10.6	μA	$+25^{\circ}\text{C}$		
		5.8	14.6	μA	$+85^{\circ}\text{C}$		
	All devices	9.2	15.6	μA	-40°C	V _{DD} = 5.0V	
		9.8	15.6	μA	$+25^{\circ}\text{C}$		
		11.4	35.6	μA	$+85^{\circ}\text{C}$		
Extended devices only	21	179	μA	$+125^{\circ}\text{C}$			

- Note 1:** The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode, with all I/O pins in high-impedance state and tied to V_{DD} or V_{SS} and all features that add delta current disabled (such as WDT, Timer1 oscillator, BOR, etc.).
- 2:** The supply current is mainly a function of operating voltage, frequency and mode. Other factors, such as I/O pin loading and switching rate, oscillator type and circuit, internal code execution pattern and temperature, also have an impact on the current consumption.
The test conditions for all I_{DD} measurements in active operation mode are:
OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to V_{DD} or V_{SS} ;
MCLR = V_{DD} ; WDT enabled/disabled as specified.
- 3:** Low-power Timer1 oscillator selected.
- 4:** BOR and LVD enable internal band gap reference. With both modules enabled, current consumption will be less than the sum of both specifications.

PIC18F1230/1330

FIGURE 23-7: CLKO AND I/O TIMING

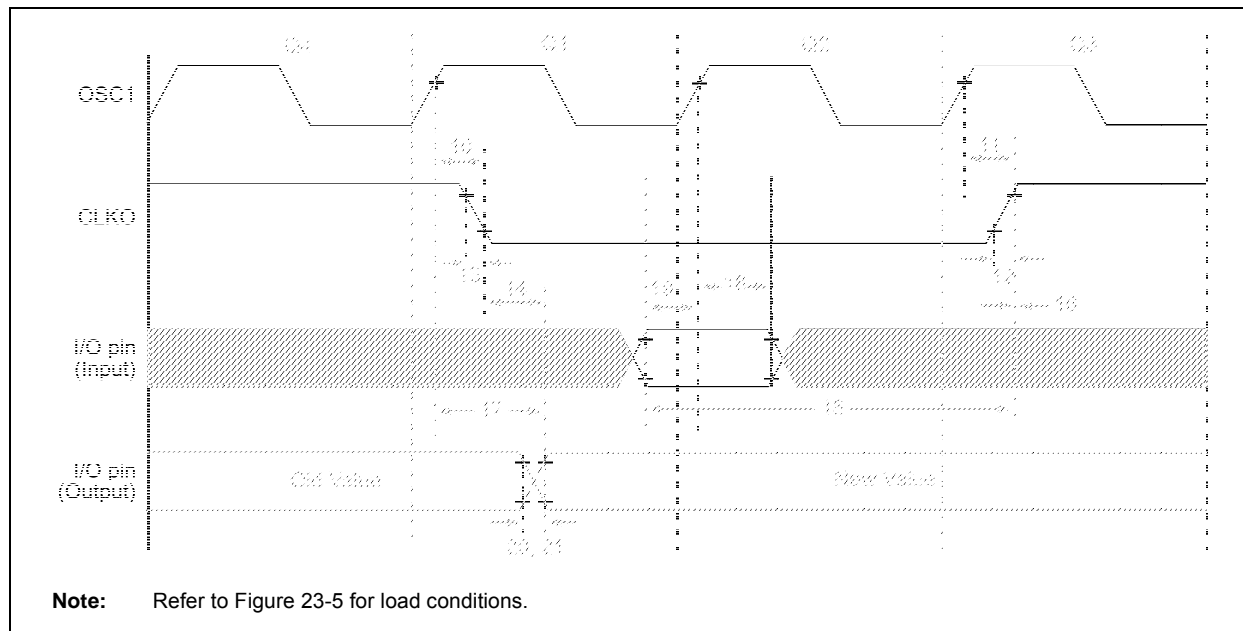


TABLE 23-9: CLKO AND I/O TIMING REQUIREMENTS

Param No.	Symbol	Characteristic	Min	Typ	Max	Units	Conditions
10	TosH2ckL	OSC1 ↑ to CLKO ↓	—	75	200	ns	(Note 1)
11	TosH2ckH	OSC1 ↑ to CLKO ↑	—	75	200	ns	(Note 1)
12	TckR	CLKO Rise Time	—	35	100	ns	(Note 1)
13	TckF	CLKO Fall Time	—	35	100	ns	(Note 1)
14	TckL2ioV	CLKO ↓ to Port Out Valid	—	—	0.5 Tcy + 20	ns	(Note 1)
15	TioV2ckH	Port In Valid before CLKO ↑	0.25 Tcy + 25	—	—	ns	(Note 1)
16	TckH2ioI	Port In Hold after CLKO ↑	0	—	—	ns	(Note 1)
17	TosH2ioV	OSC1 ↑ (Q1 cycle) to Port Out Valid	—	50	150	ns	
18	TosH2ioI	OSC1 ↑ (Q2 cycle) to Port Input Invalid (I/O in hold time)	PIC18FXXXX 100	—	—	ns	
18A			PIC18LFXXXX 200	—	—	ns	VDD = 2.0V
19	TioV2osH	Port Input Valid to OSC1 ↑ (I/O in setup time)	0	—	—	ns	
20	TioR	Port Output Rise Time	PIC18FXXXX —	10	25	ns	
20A			PIC18LFXXXX —	—	60	ns	VDD = 2.0V
21	TioF	Port Output Fall Time	PIC18FXXXX —	10	25	ns	
21A			PIC18LFXXXX —	—	60	ns	VDD = 2.0V
22†	TINP	INTx Pin High or Low Time	Tcy	—	—	ns	
23†	TRBP	RB7:RB4 Change INTx High or Low Time	Tcy	—	—	ns	

† These parameters are asynchronous events not related to any internal clock edges.

Note 1: Measurements are taken in RC mode, where CLKO output is 4 x TOSC.

PIC18F1230/1330

How to Clear RAM (Bank 0) Using Indirect Addressing	65	Data Memory	57
Implementing a Real-Time Clock Using a Timer1 Interrupt Service	115	Access Bank	59
Initializing PORTA	87	and the Extended Instruction Set	67
Initializing PORTB	90	Bank Select Register (BSR)	57
Reading a Flash Program Memory Word	75	General Purpose Registers	59
Saving STATUS, WREG and BSR Registers in RAM	105	Map for PIC18F1230/1330	58
Writing to Flash Program Memory	78–79	Special Function Registers	60
Code Protection	191, 207	DAW	234
Associated Registers	207	DC Characteristics	279
Configuration Register Protection	210	Power-Down and Supply Current	269
Data EEPROM	210	Supply Voltage	268
Program Memory	208	DCFSNZ	235
COMF	232	DECf	234
Comparator	179	DECFSZ	235
Analog Input Connection Considerations	181	Development Support	211
Associated Registers	182	Device Differences	304
Configuration	180	Device Overview	9
Effects of a Reset	181	Details on Individual Family Members	10
Interrupts	180	Features (table)	11
Operation	180	New Core Features	9
Operation During Sleep	181	Other Special Features	10
Outputs	180	Device Reset Timers	43
Reference	180	Oscillator Start-up Timer (OST)	43
Response Time	180	PLL Lock Time-out	43
Comparator Specifications	282	Power-up Timer (PWRT)	43
Comparator Voltage Reference	183	Time-out Sequence	43
Accuracy and Error	185	Direct Addressing	66
Associated Registers	185	E	
Configuring	183	Effect on Standard PIC MCU Instructions	262
Effects of a Reset	185	Effects of Power-Managed Modes on Various Clock Sources	29
Operation During Sleep	185	Electrical Characteristics	265
Computed GOTO	54	Enhanced Universal Synchronous Asynchronous Receiver Transmitter (EUSART). See EUSART.	
Configuration Bits	191	Equations	
Context Saving During Interrupts	105	A/D Acquisition Time	174
Conversion Considerations	305	A/D Minimum Charging Time	174
CPFSEQ	232	Calculating the Minimum Required Acquisition Time	174
CPFSGT	233	PWM Frequency	129
CPFSLT	233	PWM Period for Continuous Up/Down Count Mode	129
Crystal Oscillator/Ceramic Resonator	21	PWM Period for Free-Running Mode	129
Customer Change Notification Service	314	PWM Resolution	129
Customer Notification Service	314	Errata	7
Customer Support	314	EUSART	
D		Asynchronous Mode	157
Data Addressing Modes	65	12-Bit Break Character Sequence	163
Comparing Options with the Extended Instruction Set		Associated Registers, Receive	161
Enabled	68	Associated Registers, Transmit	159
Direct	65	Auto-Wake-up on Sync Break Character	161
Indexed Literal Offset	67	Receiver	160
Instructions Affected	67	Receiving a Break Character	163
Indirect	65	Setting Up 9-Bit Mode with Address Detect	160
Inherent and Literal	65	Transmitter	157
Data EEPROM Memory	81	Baud Rate Generator	
Associated Registers	84	Operation in Power-Managed Modes	151
EEADR Register	81	Baud Rate Generator (BRG)	151
EECON1 and EECON2 Registers	81	Associated Registers	152
Operation During Code-Protect	84	Auto-Baud Rate Detect	155
Protection Against Spurious Write	83	Baud Rate Error, Calculating	152
Reading	83	Baud Rates, Asynchronous Modes	153
Using	84	High Baud Rate Select (BRGH Bit)	151
Write Verify	83	Sampling	151
Writing	83	Synchronous Master Mode	164
		Associated Registers, Receive	166

PIC18F1230/1330

NOTES: