

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	20MHz
Connectivity	-
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	5
Program Memory Size	1.75KB (1K x 14)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	64 x 8
Voltage - Supply (Vcc/Vdd)	2.3V ~ 5.5V
Data Converters	A/D 4x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	8-SOIC (0.154", 3.90mm Width)
Supplier Device Package	8-SOIC
Purchase URL	<a href="https://www.e-xfl.com/product-detail/microchip-technology/pic12f1501t-i-sn">https://www.e-xfl.com/product-detail/microchip-technology/pic12f1501t-i-sn</a>

## 3.5 Stack

All devices have a 16-level x 15-bit wide hardware stack (refer to Figures 3-4 through 3-7). The stack space is not part of either program or data space. The PC is PUSHed onto the stack when `CALL` or `CALLW` instructions are executed or an interrupt causes a branch. The stack is POPed in the event of a `RETURN`, `RETLW` or a `RETFIE` instruction execution. `PCLATH` is not affected by a `PUSH` or `POP` operation.

The stack operates as a circular buffer if the `STVREN` bit is programmed to '0' (Configuration Words). This means that after the stack has been PUSHed sixteen times, the seventeenth `PUSH` overwrites the value that was stored from the first `PUSH`. The eighteenth `PUSH` overwrites the second `PUSH` (and so on). The `STKOVF` and `STKUNF` flag bits will be set on an Overflow/Underflow, regardless of whether the Reset is enabled.

**Note 1:** There are no instructions/mnemonics called `PUSH` or `POP`. These are actions that occur from the execution of the `CALL`, `CALLW`, `RETURN`, `RETLW` and `RETFIE` instructions or the vectoring to an interrupt address.

### 3.5.1 ACCESSING THE STACK

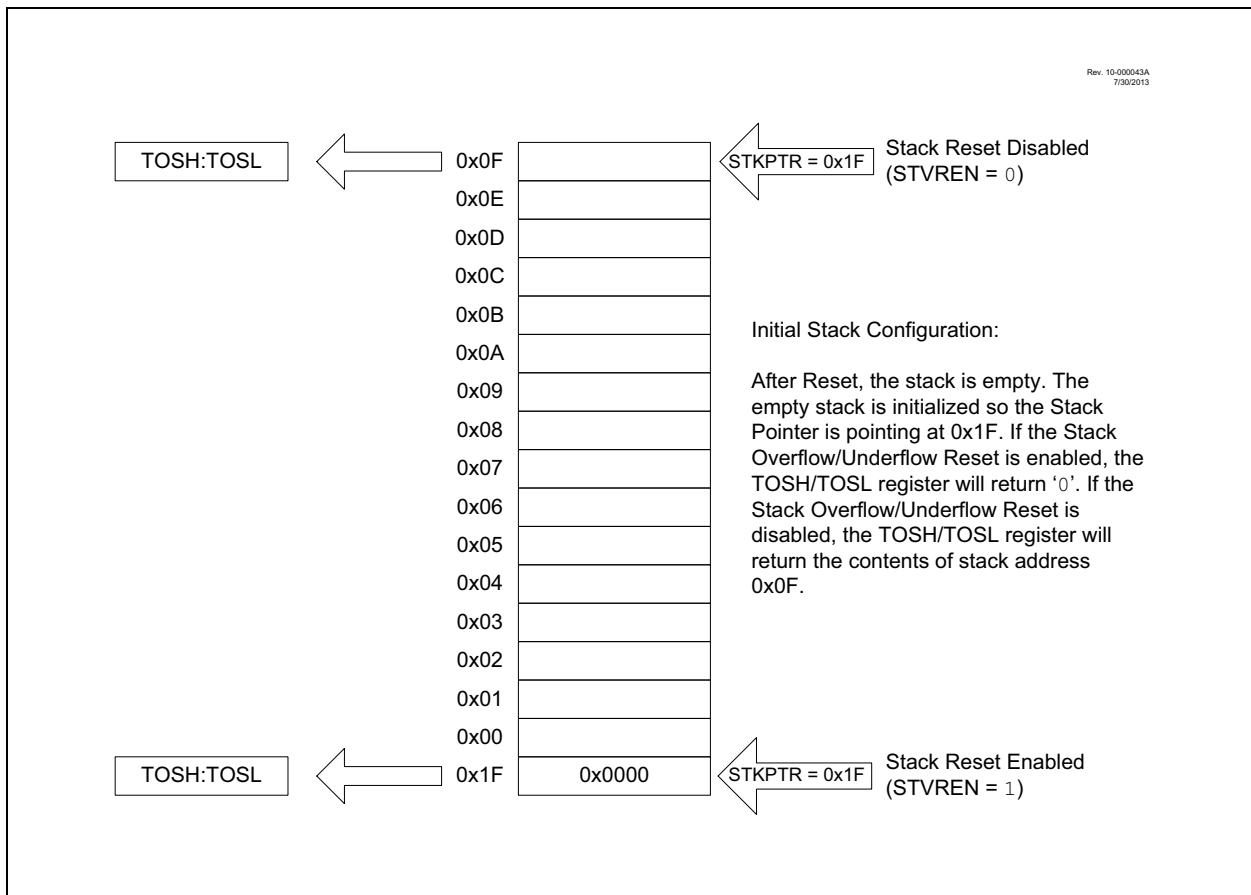
The stack is available through the `TOSH`, `TOSL` and `STKPTR` registers. `STKPTR` is the current value of the Stack Pointer. `TOSH:TOSL` register pair points to the TOP of the stack. Both registers are read/writable. `TOS` is split into `TOSH` and `TOSL` due to the 15-bit size of the PC. To access the stack, adjust the value of `STKPTR`, which will position `TOSH:TOSL`, then read/write to `TOSH:TOSL`. `STKPTR` is 5 bits to allow detection of overflow and underflow.

**Note:** Care should be taken when modifying the `STKPTR` while interrupts are enabled.

During normal program operation, `CALL`, `CALLW` and Interrupts will increment `STKPTR` while `RETLW`, `RETURN`, and `RETFIE` will decrement `STKPTR`. At any time `STKPTR` can be inspected to see how much stack is left. The `STKPTR` always points at the currently used place on the stack. Therefore, a `CALL` or `CALLW` will increment the `STKPTR` and then write the PC, and a return will unload the PC and then decrement the `STKPTR`.

Reference Figure 3-4 through Figure 3-7 for examples of accessing the stack.

**FIGURE 3-4: ACCESSING THE STACK EXAMPLE 1**



## 9.0 WATCHDOG TIMER (WDT)

The Watchdog Timer is a system timer that generates a Reset if the firmware does not issue a `CLRWDT` instruction within the time-out period. The Watchdog Timer is typically used to recover the system from unexpected events.

The WDT has the following features:

- Independent clock source
- Multiple operating modes
  - WDT is always on
  - WDT is off when in Sleep
  - WDT is controlled by software
  - WDT is always off
- Configurable time-out period is from 1 ms to 256 seconds (nominal)
- Multiple Reset conditions
- Operation during Sleep

**FIGURE 9-1: WATCHDOG TIMER BLOCK DIAGRAM**



# PIC12(L)F1501

## 10.4 User ID, Device ID and Configuration Word Access

Instead of accessing program memory, the User ID's, Device ID/Revision ID and Configuration Words can be accessed when  $CFG5 = 1$  in the PMCON1 register. This is the region that would be pointed to by  $PC<15> = 1$ , but not all addresses are accessible. Different access may exist for reads and writes. Refer to Table 10-2.

When read access is initiated on an address outside the parameters listed in Table 10-2, the PMDATH:PMDATL register pair is cleared, reading back '0's.

**TABLE 10-2: USER ID, DEVICE ID AND CONFIGURATION WORD ACCESS (CFG5 = 1)**

Address	Function	Read Access	Write Access
8000h-8003h	User IDs	Yes	Yes
8006h	Device ID/Revision ID	Yes	No
8007h-8008h	Configuration Words 1 and 2	Yes	No

### EXAMPLE 10-4: CONFIGURATION WORD AND DEVICE ID ACCESS

```
* This code block will read 1 word of program memory at the memory address:
*   PROG_ADDR_LO (must be 00h-08h) data will be returned in the variables;
*   PROG_DATA_HI, PROG_DATA_LO

BANKSEL PMADRL           ; Select correct Bank
MOVLW   PROG_ADDR_LO     ;
MOVWF   PMADRL           ; Store LSB of address
CLRF    PMADRH           ; Clear MSB of address

BSF     PMCON1,CFG5      ; Select Configuration Space
BCF     INTCON,GIE       ; Disable interrupts
BSF     PMCON1,RD        ; Initiate read
NOP     ; Executed (See Figure 10-2)
NOP     ; Ignored (See Figure 10-2)
BSF     INTCON,GIE       ; Restore interrupts

MOVF    PMDATL,W         ; Get LSB of word
MOVWF   PROG_DATA_LO     ; Store in user location
MOVF    PMDATH,W         ; Get MSB of word
MOVWF   PROG_DATA_HI     ; Store in user location
```

# PIC12(L)F1501

## 11.1 Alternate Pin Function

The Alternate Pin Function Control (APFCON) register is used to steer specific peripheral input and output functions between different pins. The APFCON register is shown in Register 11-1. For this device family, the following functions can be moved between different pins.

- T1G
- CLC1
- NCO1
- CWG1A
- CWG1B

These bits have no effect on the values of any TRIS register. PORT and TRIS overrides will be routed to the correct pin. The unselected pin will be unaffected.

## 11.2 Register Definitions: Alternate Pin Function Control

**REGISTER 11-1: APFCON: ALTERNATE PIN FUNCTION CONTROL REGISTER**

R/W-0/0	R/W-0/0	U-0	U-0	R/W-0/0	U-0	R/W-0/0	R/W-0/0
CWG1BSEL	CWG1ASEL	—	—	T1GSEL	—	CLC1SEL	NCO1SEL
bit 7							bit 0

**Legend:**

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

u = Bit is unchanged

x = Bit is unknown

-n/n = Value at POR and BOR/Value at all other Resets

'1' = Bit is set

'0' = Bit is cleared

- bit 7      **CWG1BSEL:** Pin Selection bit  
1 = CWG1B function is on RA4  
0 = CWG1B function is on RA0
- bit 6      **CWG1ASEL:** Pin Selection bit  
1 = CWG1A function is on RA5  
0 = CWG1A function is on RA2
- bit 5-4    **Unimplemented:** Read as '0'
- bit 3      **T1GSEL:** Pin Selection bit  
1 = T1G function is on RA3  
0 = T1G function is on RA4
- bit 2      **Unimplemented:** Read as '0'
- bit 1      **CLC1SEL:** Pin Selection bit  
1 = CLC1 function is on RA4  
0 = CLC1 function is on RA2
- bit 0      **NCO1SEL:** Pin Selection bit  
1 = NCO1 function is on RA5  
0 = NCO1 function is on RA1

# PIC12(L)F1501

## 11.4 Register Definitions: PORTA

### REGISTER 11-2: PORTA: PORTA REGISTER

U-0	U-0	R/W-x/x	R/W-x/x	R-x/x	R/W-x/x	R/W-x/x	R/W-x/x
—	—	RA5	RA4	RA3	RA2	RA1	RA0
bit 7							bit 0

#### Legend:

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

bit 7-6 **Unimplemented:** Read as '0'

bit 5-0 **RA<5:0>:** PORTA I/O Value bits<sup>(1)</sup>

1 = Port pin is  $\geq V_{IH}$

0 = Port pin is  $\leq V_{IL}$

**Note 1:** Writes to PORTA are actually written to corresponding LATA register. Reads from PORTA register is return of actual I/O pin values.

### REGISTER 11-3: TRISA: PORTA TRI-STATE REGISTER

U-0	U-0	R/W-1/1	R/W-1/1	U-1	R/W-1/1	R/W-1/1	R/W-1/1
—	—	TRISA5	TRISA4	— <sup>(1)</sup>	TRISA2	TRISA1	TRISA0
bit 7							bit 0

#### Legend:

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

bit 7-6 **Unimplemented:** Read as '0'

bit 5-4 **TRISA<5:4>:** PORTA Tri-State Control bit

1 = PORTA pin configured as an input (tri-stated)

0 = PORTA pin configured as an output

bit 3 **Unimplemented:** Read as '1'

bit 2-0 **TRISA<2:0>:** PORTA Tri-State Control bit

1 = PORTA pin configured as an input (tri-stated)

0 = PORTA pin configured as an output

**Note 1:** Unimplemented, read as '1'.

# PIC12(L)F1501

**TABLE 12-1: SUMMARY OF REGISTERS ASSOCIATED WITH INTERRUPT-ON-CHANGE**

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ANSELA	—	—	—	ANSA4	—	ANSA2	ANSA1	ANSA0	99
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	64
IOCAF	—	—	IOCAF5	IOCAF4	IOCAF3	IOCAF2	IOCAF1	IOCAF0	103
IOCAN	—	—	IOCAN5	IOCAN4	IOCAN3	IOCAN2	IOCAN1	IOCAN0	103
IOCAP	—	—	IOCAP5	IOCAP4	IOCAP3	IOCAP2	IOCAP1	IOCAP0	103
TRISA	—	—	TRISA5	TRISA4	— <sup>(1)</sup>	TRISA2	TRISA1	TRISA0	98

**Legend:** — = unimplemented location, read as '0'. Shaded cells are not used by interrupt-on-change.

**Note 1:** Unimplemented, read as '1'.

**TABLE 14-2: SUMMARY OF REGISTERS ASSOCIATED WITH THE TEMPERATURE INDICATOR**

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on page
FVRCON	FVREN	FVRRDY	TSEN	TSRNG	CDAFVR<1:0>	ADFVR<1:0>			107

**Legend:** Shaded cells are unused by the temperature indicator module.



## REGISTER 15-2: ADCON1: ADC CONTROL REGISTER 1

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0	R/W-0/0	R/W-0/0
ADFM	ADCS<2:0>			—	—	ADPREF<1:0>	
bit 7							bit 0

### Legend:

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

- bit 7      **ADFM:** ADC Result Format Select bit  
 1 = Right justified. Six Most Significant bits of ADRESH are set to '0' when the conversion result is loaded.  
 0 = Left justified. Six Least Significant bits of ADRESL are set to '0' when the conversion result is loaded.
- bit 6-4      **ADCS<2:0>:** ADC Conversion Clock Select bits  
 000 = Fosc/2  
 001 = Fosc/8  
 010 = Fosc/32  
 011 = FRC (clock supplied from an internal RC oscillator)  
 100 = Fosc/4  
 101 = Fosc/16  
 110 = Fosc/64  
 111 = FRC (clock supplied from an internal RC oscillator)
- bit 3-2      **Unimplemented:** Read as '0'
- bit 1-0      **ADPREF<1:0>:** ADC Positive Voltage Reference Configuration bits  
 00 = VRPOS is connected to VDD  
 01 = Reserved  
 10 = VRPOS is connected to external VREF+ pin<sup>(1)</sup>  
 11 = Reserved

**Note 1:** When selecting the VREF+ pin as the source of the positive reference, be aware that a minimum voltage specification exists. See **Section 27.0 "Electrical Specifications"** for details.

## 16.1 Output Voltage Selection

The DAC has 32 voltage level ranges. The 32 levels are set with the DACR<4:0> bits of the DACxCON1 register.

The DAC output voltage can be determined by using Equation 16-1.

## 16.2 Ratiometric Output Level

The DAC output value is derived using a resistor ladder with each end of the ladder tied to a positive and negative voltage reference input source. If the voltage of either input source fluctuates, a similar fluctuation will result in the DAC output value.

The value of the individual resistors within the ladder can be found in Table 27-14.

## 16.3 DAC Voltage Reference Output

The unbuffered DAC voltage can be output to the DACxOUTn pin(s) by setting the respective DACOEn bit(s) of the DACxCON0 register. Selecting the DAC reference voltage for output on either DACxOUTn pin automatically overrides the digital output buffer, the weak pull-up and digital input threshold detector functions of that pin.

Reading the DACxOUTn pin when it has been configured for DAC reference voltage output will always return a '0'.

**Note:** The unbuffered DAC output (DACxOUTn) is not intended to drive an external load.

## 16.4 Operation During Sleep

When the device wakes up from Sleep through an interrupt or a Watchdog Timer time-out, the contents of the DACxCON0 register are not affected. To minimize current consumption in Sleep mode, the voltage reference should be disabled.

## 16.5 Effects of a Reset

A device Reset affects the following:

- DACx is disabled.
- DACx output voltage is removed from the DACxOUTn pin(s).
- The DACR<4:0> range select bits are cleared.

### EQUATION 16-1: DAC OUTPUT VOLTAGE

***IF DACEN = 1***

$$DACx\_output = \left( (V_{SOURCE+} - V_{SOURCE-}) \times \frac{DACR[4:0]}{2^5} \right) + V_{SOURCE-}$$

**Note:** See the DACxCON0 register for the available VSOURCE+ and VSOURCE- selections.

## 19.3 Timer1 Prescaler

Timer1 has four prescaler options allowing 1, 2, 4 or 8 divisions of the clock input. The T1CKPS bits of the T1CON register control the prescale counter. The prescale counter is not directly readable or writable; however, the prescaler counter is cleared upon a write to TMR1H or TMR1L.

## 19.4 Timer1 Operation in Asynchronous Counter Mode

If control bit T1SYNC of the T1CON register is set, the external clock input is not synchronized. The timer increments asynchronously to the internal phase clocks. If the external clock source is selected then the timer will continue to run during Sleep and can generate an interrupt on overflow, which will wake-up the processor. However, special precautions in software are needed to read/write the timer (see Section 19.4.1 “Reading and Writing Timer1 in Asynchronous Counter Mode”).

**Note:** When switching from synchronous to asynchronous operation, it is possible to skip an increment. When switching from asynchronous to synchronous operation, it is possible to produce an additional increment.

### 19.4.1 READING AND WRITING TIMER1 IN ASYNCHRONOUS COUNTER MODE

Reading TMR1H or TMR1L while the timer is running from an external asynchronous clock will ensure a valid read (taken care of in hardware). However, the user should keep in mind that reading the 16-bit timer in two 8-bit values itself, poses certain problems, since the timer may overflow between the reads.

For writes, it is recommended that the user simply stop the timer and write the desired values. A write contention may occur by writing to the timer registers, while the register is incrementing. This may produce an unpredictable value in the TMR1H:TMR1L register pair.

## 19.5 Timer1 Gate

Timer1 can be configured to count freely or the count can be enabled and disabled using Timer1 gate circuitry. This is also referred to as Timer1 Gate Enable.

Timer1 gate can also be driven by multiple selectable sources.

### 19.5.1 TIMER1 GATE ENABLE

The Timer1 Gate Enable mode is enabled by setting the TMR1GE bit of the T1GCON register. The polarity of the Timer1 Gate Enable mode is configured using the T1GPOL bit of the T1GCON register.

When Timer1 Gate Enable mode is enabled, Timer1 will increment on the rising edge of the Timer1 clock source. When Timer1 Gate Enable mode is disabled, no incrementing will occur and Timer1 will hold the current count. See Figure 19-3 for timing details.

**TABLE 19-3: TIMER1 GATE ENABLE SELECTIONS**

T1CLK	T1GPOL	T1G	Timer1 Operation
↑	0	0	Counts
↑	0	1	Holds Count
↑	1	0	Holds Count
↑	1	1	Counts

### 19.5.2 TIMER1 GATE SOURCE SELECTION

Timer1 gate source selections are shown in Table 19-4. Source selection is controlled by the T1GSS<1:0> bits of the T1GCON register. The polarity for each available source is also selectable. Polarity selection is controlled by the T1GPOL bit of the T1GCON register.

**TABLE 19-4: TIMER1 GATE SOURCES**

T1GSS	Timer1 Gate Source
00	Timer1 Gate pin (T1G)
01	Overflow of Timer0 (T0_overflow) (TMR0 increments from FFh to 00h)
10	Comparator 1 Output (C1OUT_sync) <sup>(1)</sup>
11	Reserved

**Note 1:** Optionally synchronized comparator output.

## 19.6 Timer1 Interrupt

The Timer1 register pair (TMR1H:TMR1L) increments to FFFFh and rolls over to 0000h. When Timer1 rolls over, the Timer1 interrupt flag bit of the PIR1 register is set. To enable the interrupt on rollover, you must set these bits:

- TMR1ON bit of the T1CON register
- TMR1IE bit of the PIE1 register
- PEIE bit of the INTCON register
- GIE bit of the INTCON register

The interrupt is cleared by clearing the TMR1IF bit in the Interrupt Service Routine.

**Note:** The TMR1H:TMR1L register pair and the TMR1IF bit should be cleared before enabling interrupts.

## 19.7 Timer1 Operation During Sleep

Timer1 can only operate during Sleep when setup in Asynchronous Counter mode. In this mode, an external crystal or clock source can be used to increment the counter. To set up the timer to wake the device:

- TMR1ON bit of the T1CON register must be set
- TMR1IE bit of the PIE1 register must be set
- PEIE bit of the INTCON register must be set
- T1SYNC bit of the T1CON register must be set
- TMR1CS bits of the T1CON register must be configured

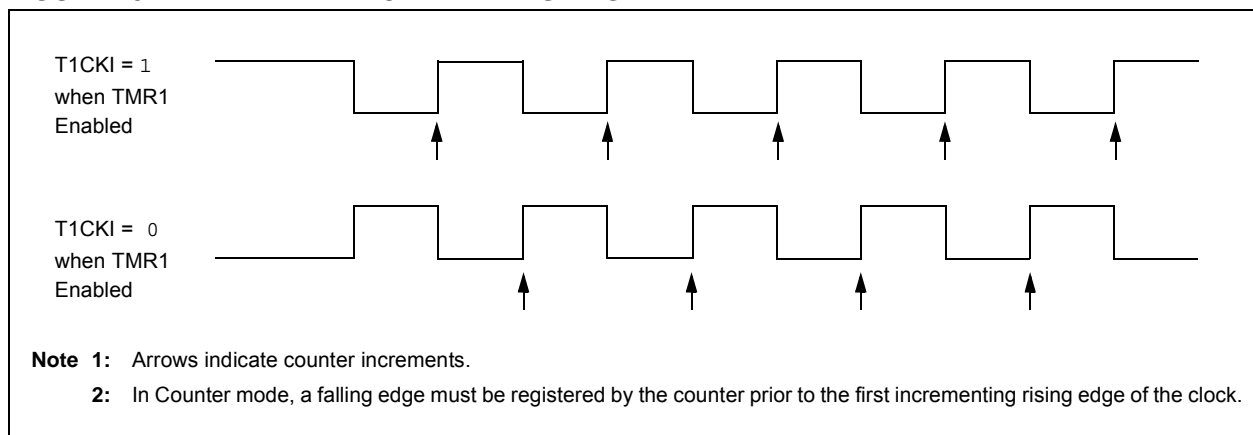
The device will wake-up on an overflow and execute the next instructions. If the GIE bit of the INTCON register is set, the device will call the Interrupt Service Routine.

Timer1 oscillator will continue to operate in Sleep regardless of the T1SYNC bit setting.

## 19.7.1 ALTERNATE PIN LOCATIONS

This module incorporates I/O pins that can be moved to other locations with the use of the alternate pin function register, APFCON. To determine which pins can be moved and what their default locations are upon a Reset, see Section 11.1 “Alternate Pin Function” for more information.

**FIGURE 19-2: TIMER1 INCREMENTING EDGE**



## 22.6 Register Definitions: CLC Control

### REGISTER 22-1: CLCxCON: CONFIGURABLE LOGIC CELL CONTROL REGISTER

R/W-0/0	R/W-0/0	R-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
LCxEN	LCxOE	LCxOUT	LCxINTP	LCxINTN	LCxMODE<2:0>		
bit 7						bit 0	

#### Legend:

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

- bit 7      **LCxEN:** Configurable Logic Cell Enable bit  
 1 = Configurable logic cell is enabled and mixing input signals  
 0 = Configurable logic cell is disabled and has logic zero output
- bit 6      **LCxOE:** Configurable Logic Cell Output Enable bit  
 1 = Configurable logic cell port pin output enabled  
 0 = Configurable logic cell port pin output disabled
- bit 5      **LCxOUT:** Configurable Logic Cell Data Output bit  
 Read-only: logic cell output data, after LCxPOL; sampled from lcx\_out wire.
- bit 4      **LCxINTP:** Configurable Logic Cell Positive Edge Going Interrupt Enable bit  
 1 = CLCxIF will be set when a rising edge occurs on lcx\_out  
 0 = CLCxIF will not be set
- bit 3      **LCxINTN:** Configurable Logic Cell Negative Edge Going Interrupt Enable bit  
 1 = CLCxIF will be set when a falling edge occurs on lcx\_out  
 0 = CLCxIF will not be set
- bit 2-0    **LCxMODE<2:0>:** Configurable Logic Cell Functional Mode bits  
 111 = Cell is 1-input transparent latch with S and R  
 110 = Cell is J-K flip-flop with R  
 101 = Cell is 2-input D flip-flop with R  
 100 = Cell is 1-input D flip-flop with S and R  
 011 = Cell is S-R latch  
 010 = Cell is 4-input AND  
 001 = Cell is OR-XOR  
 000 = Cell is AND-OR

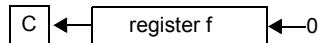
# PIC12(L)F1501

---

## LSLF Logical Left Shift

---

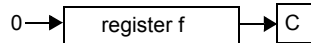
Syntax: [ *label* ] LSLF f {,d}  
Operands:  $0 \leq f \leq 127$   
 $d \in [0,1]$   
Operation:  $(f<7>) \rightarrow C$   
 $(f<6:0>) \rightarrow \text{dest}<7:1>$   
 $0 \rightarrow \text{dest}<0>$   
Status Affected: C, Z  
Description: The contents of register 'f' are shifted one bit to the left through the Carry flag. A '0' is shifted into the LSb. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is stored back in register 'f'.



## LSRF Logical Right Shift

---

Syntax: [ *label* ] LSRF f {,d}  
Operands:  $0 \leq f \leq 127$   
 $d \in [0,1]$   
Operation:  $0 \rightarrow \text{dest}<7>$   
 $(f<7:1>) \rightarrow \text{dest}<6:0>$ ,  
 $(f<0>) \rightarrow C$ ,  
Status Affected: C, Z  
Description: The contents of register 'f' are shifted one bit to the right through the Carry flag. A '0' is shifted into the MSb. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is stored back in register 'f'.



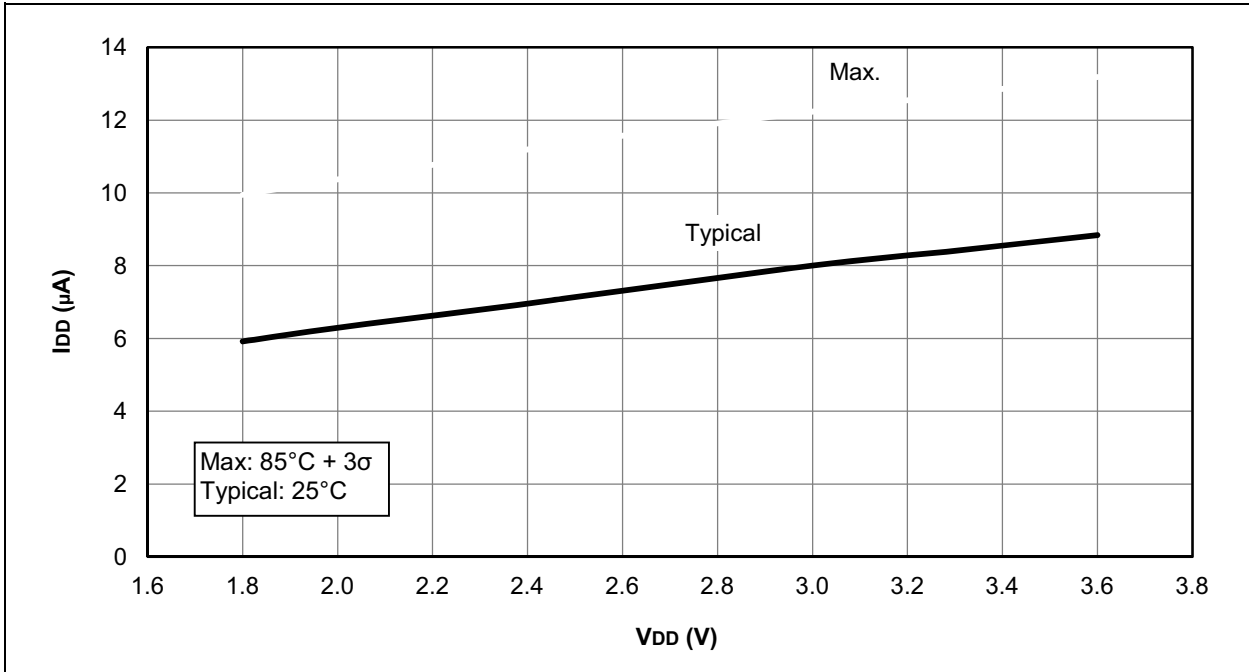
## MOVF Move f

---

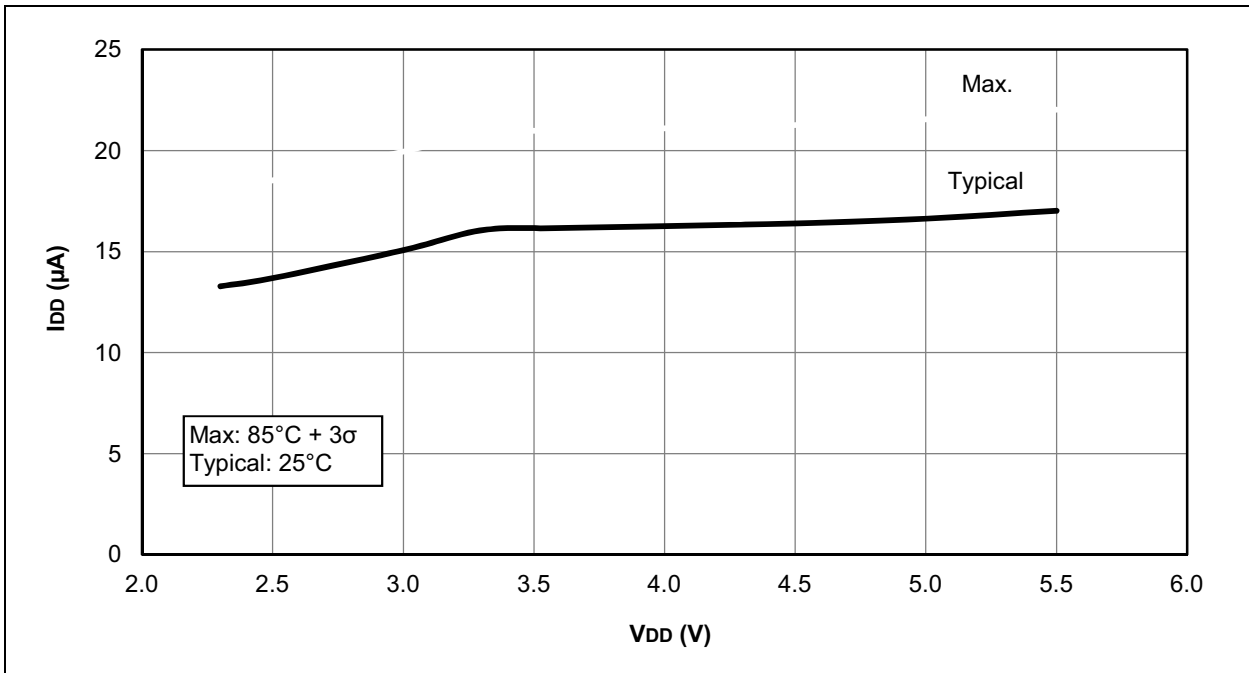
Syntax: [ *label* ] MOVF f,d  
Operands:  $0 \leq f \leq 127$   
 $d \in [0,1]$   
Operation:  $(f) \rightarrow (\text{dest})$   
Status Affected: Z  
Description: The contents of register f is moved to a destination dependent upon the status of d. If d = 0, destination is W register. If d = 1, the destination is file register f itself. d = 1 is useful to test a file register since status flag Z is affected.  
Words: 1  
Cycles: 1  
Example: MOVF FSR, 0

After Instruction  
W = value in FSR register  
Z = 1

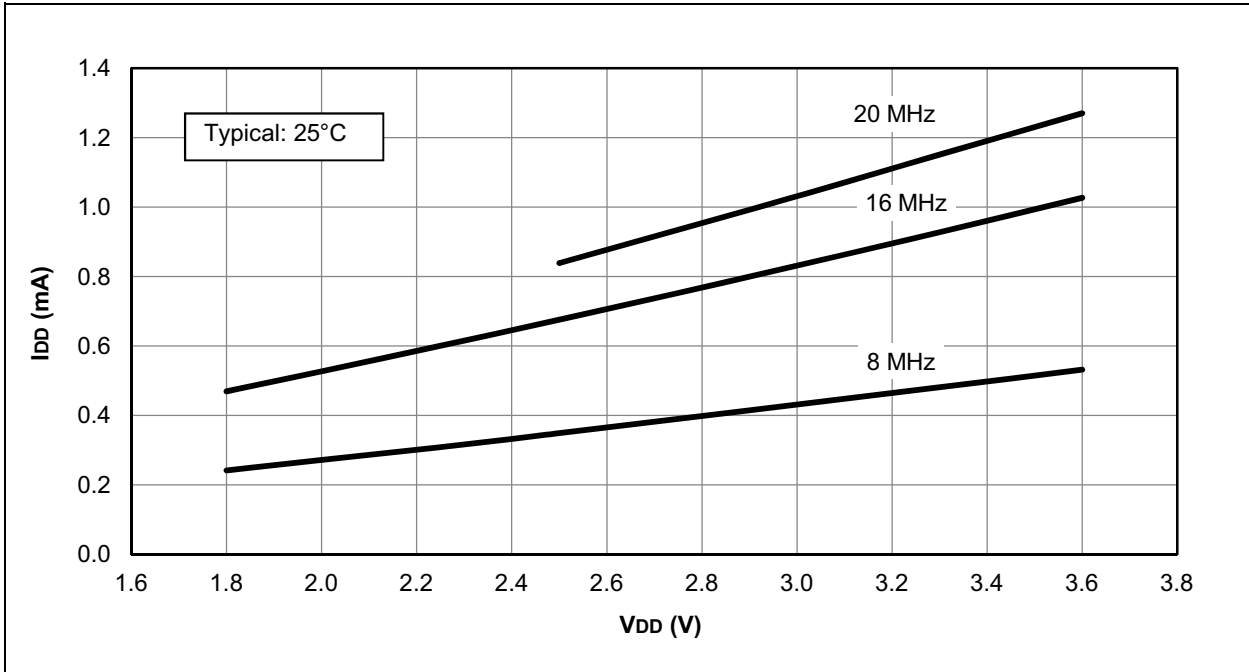
**FIGURE 28-1: I<sub>DD</sub>, EXTERNAL CLOCK (ECL), LOW-POWER MODE, F<sub>osc</sub> = 32 kHz, PIC12LF1501 ONLY**



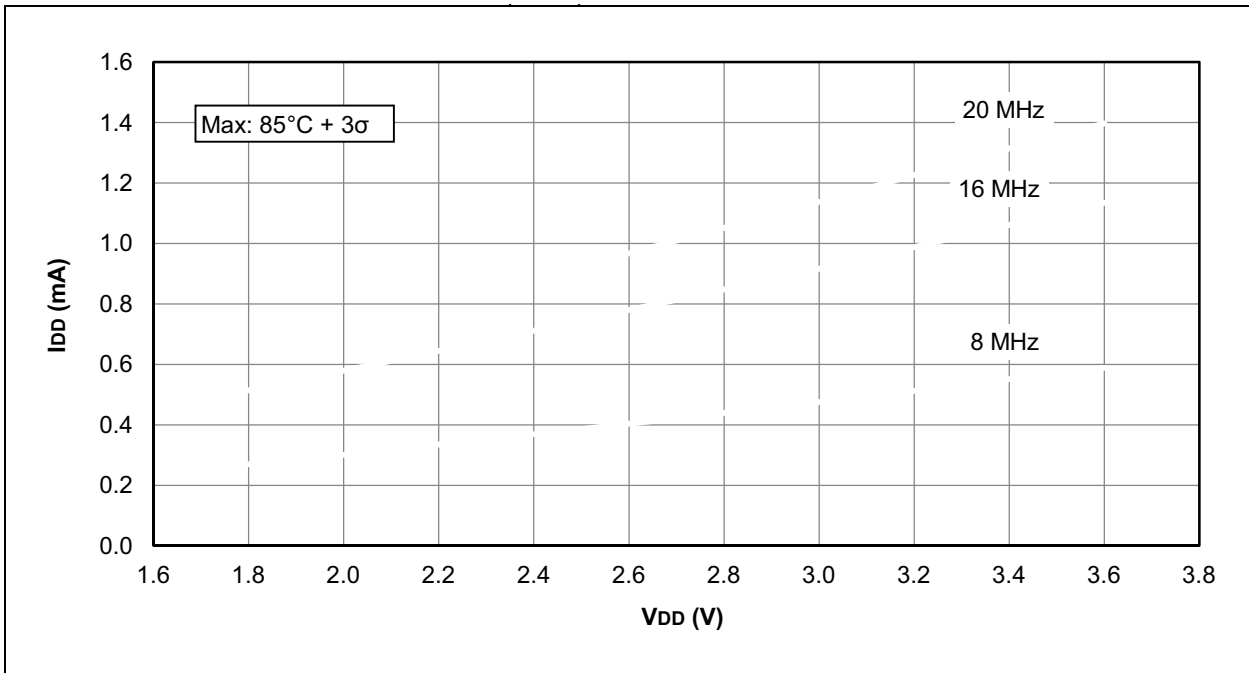
**FIGURE 28-2: I<sub>DD</sub>, EXTERNAL CLOCK (ECL), LOW-POWER MODE, F<sub>osc</sub> = 32 kHz, PIC12F1501 ONLY**



**FIGURE 28-9: I<sub>DD</sub> TYPICAL, EXTERNAL CLOCK (ECH), HIGH-POWER MODE, PIC12LF1501 ONLY**



**FIGURE 28-10: I<sub>DD</sub> MAXIMUM, EXTERNAL CLOCK (ECH), HIGH-POWER MODE, PIC12LF1501 ONLY**





# PIC12(L)F1501

FIGURE 28-53: COMPARATOR HYSTERESIS, NORMAL POWER MODE (CxSP = 1, CxHYS = 1)

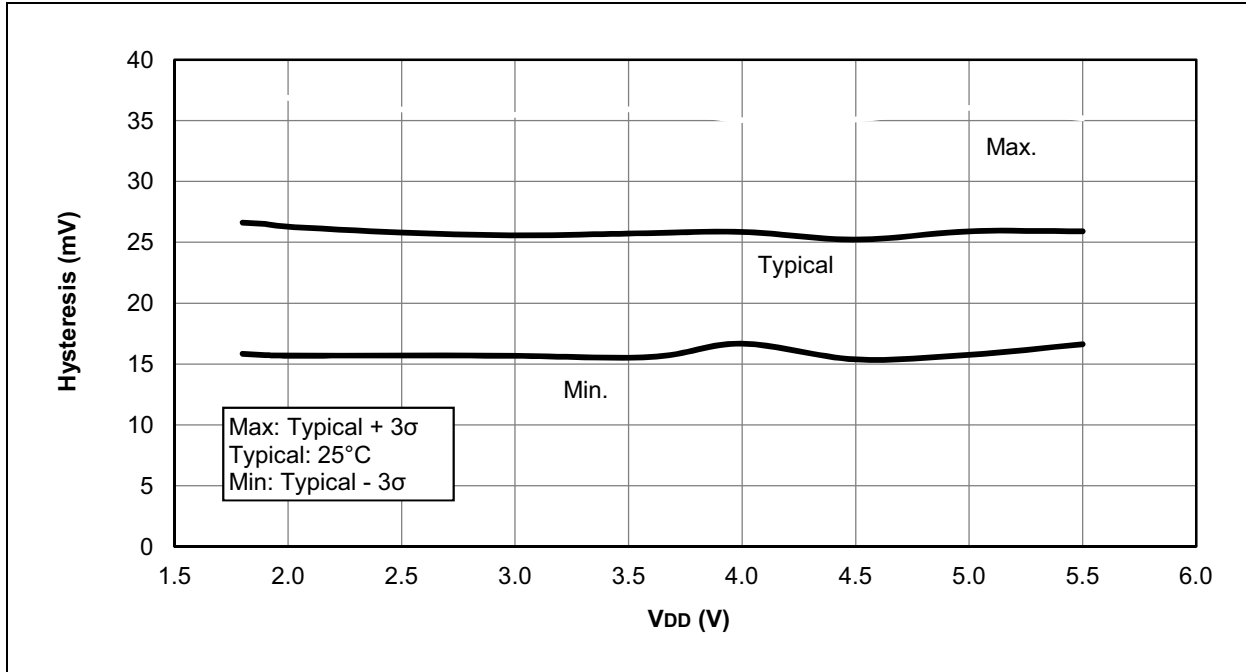
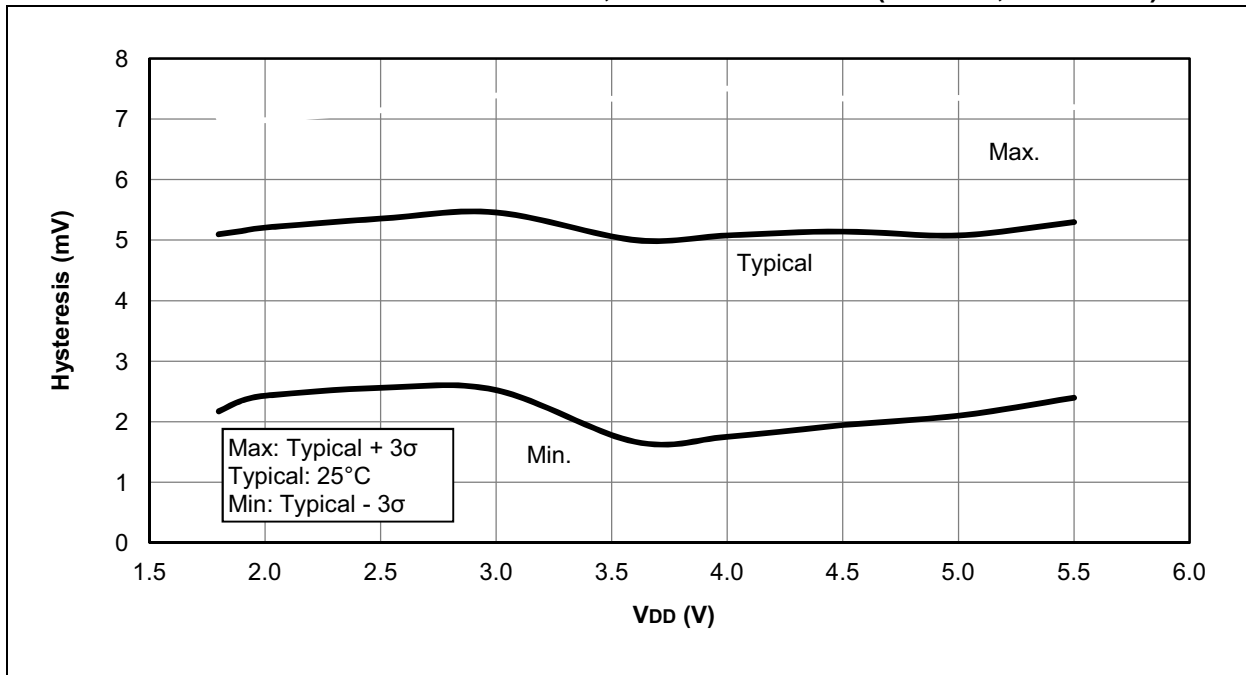


FIGURE 28-54: COMPARATOR HYSTERESIS, LOW-POWER MODE (CxSP = 0, CxHYS = 1)



## 29.6 MPLAB X SIM Software Simulator

The MPLAB X SIM Software Simulator allows code development in a PC-hosted environment by simulating the PIC MCUs and dsPIC DSCs on an instruction level. On any given instruction, the data areas can be examined or modified and stimuli can be applied from a comprehensive stimulus controller. Registers can be logged to files for further run-time analysis. The trace buffer and logic analyzer display extend the power of the simulator to record and track program execution, actions on I/O, most peripherals and internal registers.

The MPLAB X SIM Software Simulator fully supports symbolic debugging using the MPLAB XC Compilers, and the MPASM and MPLAB Assemblers. The software simulator offers the flexibility to develop and debug code outside of the hardware laboratory environment, making it an excellent, economical software development tool.

## 29.7 MPLAB REAL ICE In-Circuit Emulator System

The MPLAB REAL ICE In-Circuit Emulator System is Microchip's next generation high-speed emulator for Microchip Flash DSC and MCU devices. It debugs and programs all 8, 16 and 32-bit MCU, and DSC devices with the easy-to-use, powerful graphical user interface of the MPLAB X IDE.

The emulator is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with either a connector compatible with in-circuit debugger systems (RJ-11) or with the new high-speed, noise tolerant, Low-Voltage Differential Signal (LVDS) interconnection (CAT5).

The emulator is field upgradable through future firmware downloads in MPLAB X IDE. MPLAB REAL ICE offers significant advantages over competitive emulators including full-speed emulation, run-time variable watches, trace analysis, complex breakpoints, logic probes, a ruggedized probe interface and long (up to three meters) interconnection cables.

## 29.8 MPLAB ICD 3 In-Circuit Debugger System

The MPLAB ICD 3 In-Circuit Debugger System is Microchip's most cost-effective, high-speed hardware debugger/programmer for Microchip Flash DSC and MCU devices. It debugs and programs PIC Flash microcontrollers and dsPIC DSCs with the powerful, yet easy-to-use graphical user interface of the MPLAB IDE.

The MPLAB ICD 3 In-Circuit Debugger probe is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with a connector compatible with the MPLAB ICD 2 or MPLAB REAL ICE systems (RJ-11). MPLAB ICD 3 supports all MPLAB ICD 2 headers.

## 29.9 PICkit 3 In-Circuit Debugger/Programmer

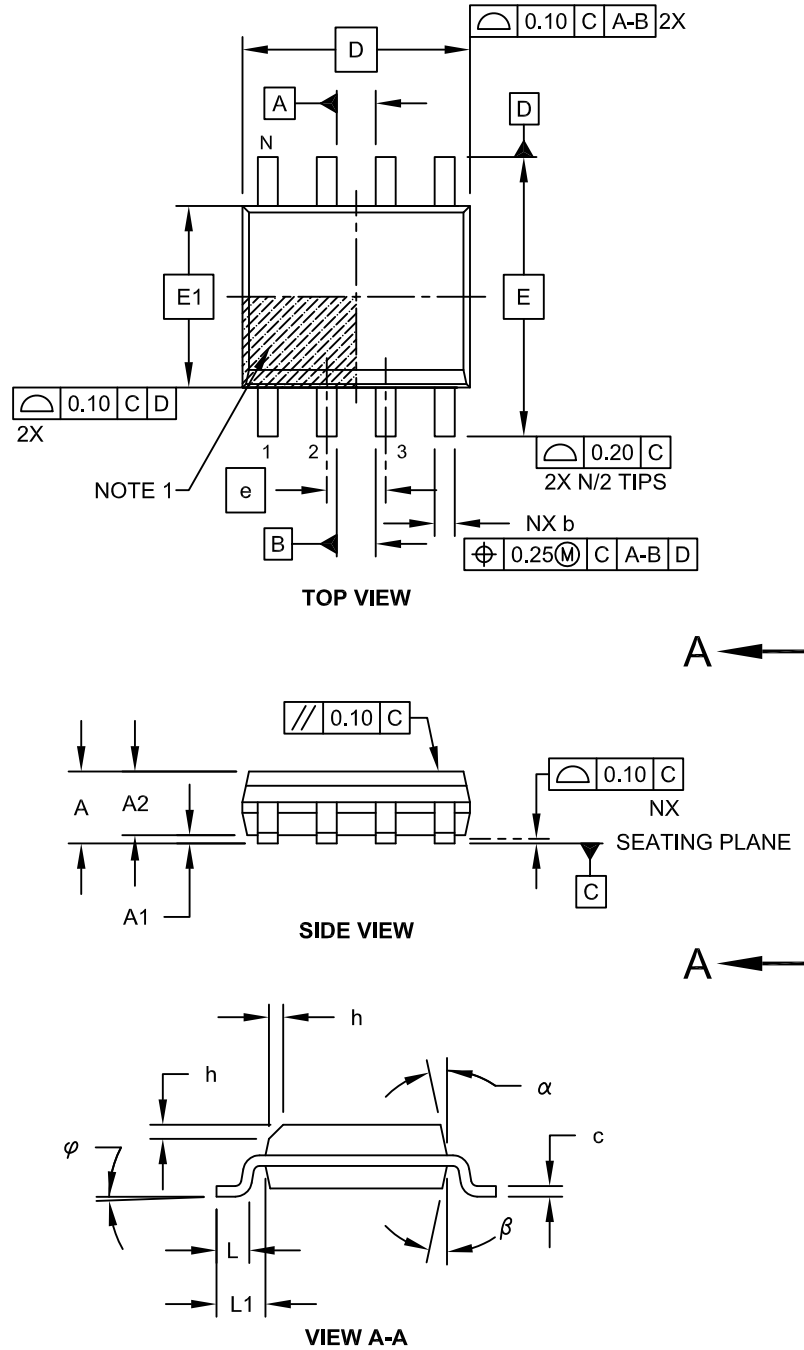
The MPLAB PICkit 3 allows debugging and programming of PIC and dsPIC Flash microcontrollers at a most affordable price point using the powerful graphical user interface of the MPLAB IDE. The MPLAB PICkit 3 is connected to the design engineer's PC using a full-speed USB interface and can be connected to the target via a Microchip debug (RJ-11) connector (compatible with MPLAB ICD 3 and MPLAB REAL ICE). The connector uses two device I/O pins and the Reset line to implement in-circuit debugging and In-Circuit Serial Programming™ (ICSP™).

## 29.10 MPLAB PM3 Device Programmer

The MPLAB PM3 Device Programmer is a universal, CE compliant device programmer with programmable voltage verification at VDDMIN and VDDMAX for maximum reliability. It features a large LCD display (128 x 64) for menus and error messages, and a modular, detachable socket assembly to support various package types. The ICSP cable assembly is included as a standard item. In Stand-Alone mode, the MPLAB PM3 Device Programmer can read, verify and program PIC devices without a PC connection. It can also set code protection in this mode. The MPLAB PM3 connects to the host PC via an RS-232 or USB cable. The MPLAB PM3 has high-speed communications and optimized algorithms for quick programming of large memory devices, and incorporates an MMC card for file storage and data applications.

## 8-Lead Plastic Small Outline (SN) - Narrow, 3.90 mm Body [SOIC]

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

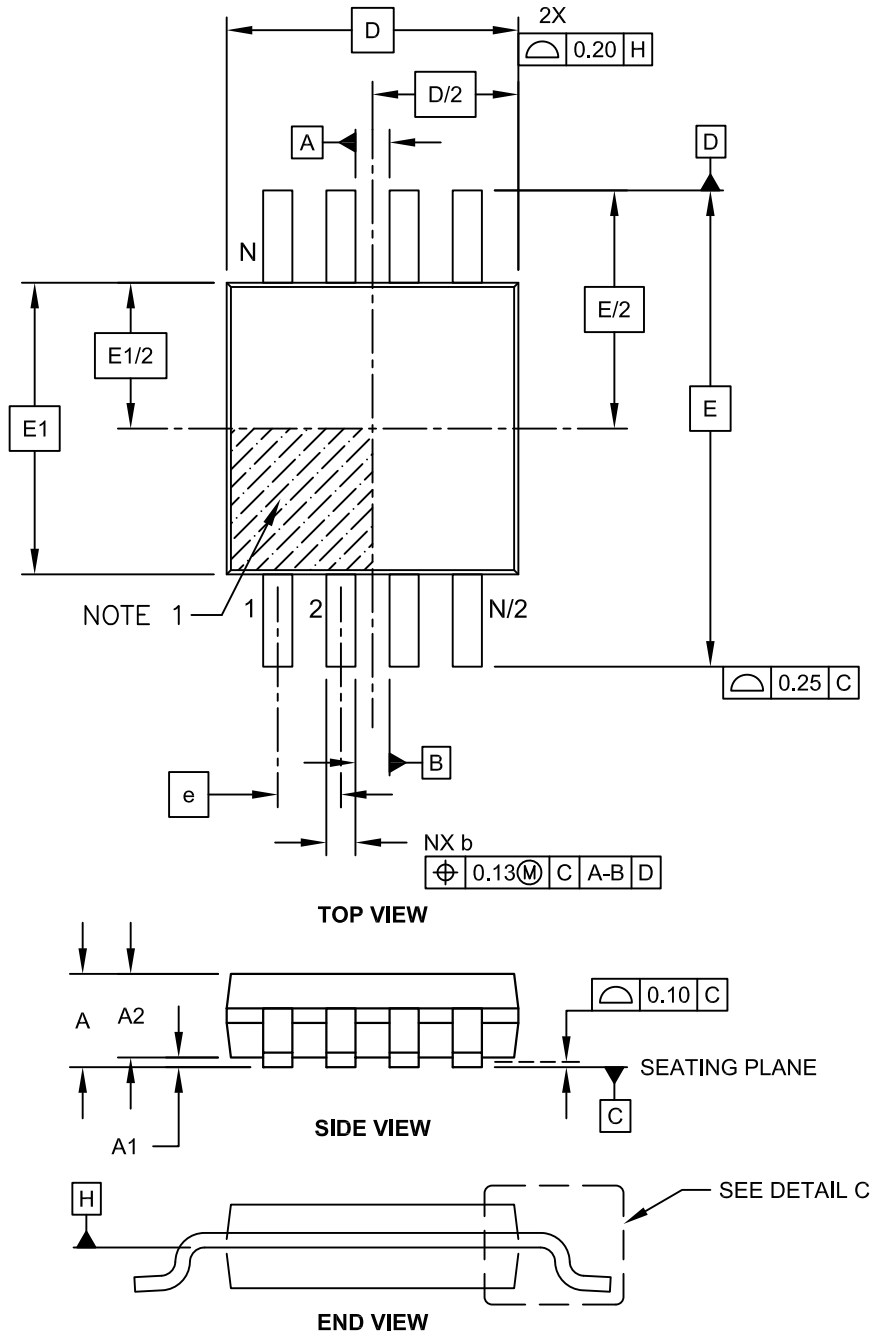


Microchip Technology Drawing No. C04-057C Sheet 1 of 2

# PIC12(L)F1501

## 8-Lead Plastic Micro Small Outline Package (MS) [MSOP]

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

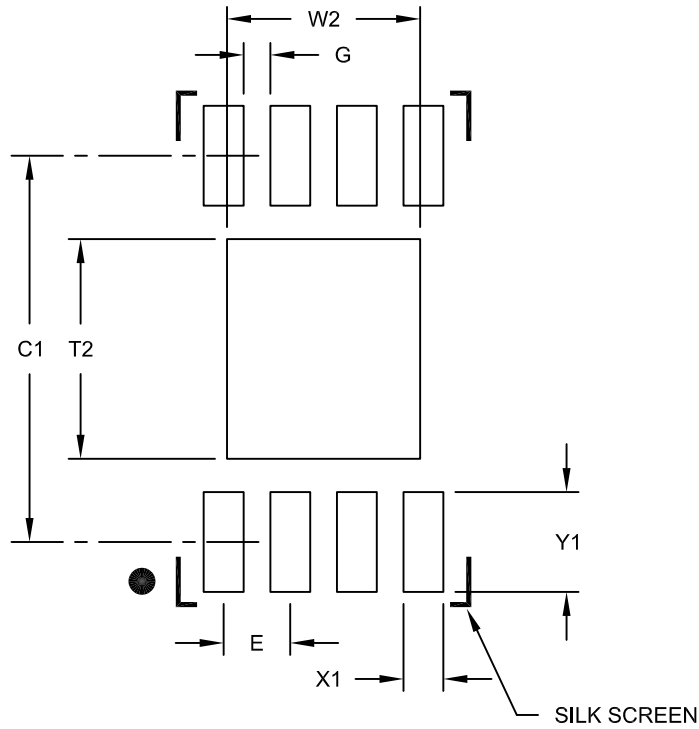


Microchip Technology Drawing C04-111C Sheet 1 of 2

# PIC12(L)F1501

8-Lead Plastic Dual Flat, No Lead Package (MC) - 2x3x0.9mm Body [DFN]

**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	E		0.50 BSC		
Optional Center Pad Width	W2				1.45
Optional Center Pad Length	T2				1.75
Contact Pad Spacing	C1		2.90		
Contact Pad Width (X8)	X1				0.30
Contact Pad Length (X8)	Y1				0.75
Distance Between Pads	G	0.20			

**Notes:**

1. Dimensioning and tolerancing per ASME Y14.5M

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

Microchip Technology Drawing No. C04-2123B