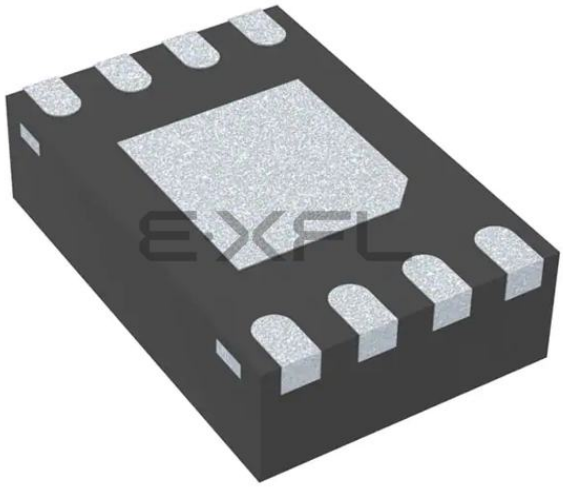


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	8MHz
Connectivity	-
Peripherals	POR, WDT
Number of I/O	4
Program Memory Size	768B (512 x 12)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	23 x 8
Voltage - Supply (Vcc/Vdd)	2V ~ 5.5V
Data Converters	A/D 2x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	8-VFDFN Exposed Pad
Supplier Device Package	8-DFN (2x3)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic10f222t-i-mc

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

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

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

Trademarks

Printed on recycled paper.

ISBN: 9781620775912

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
═ ISO/TS 16949 ═

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

PIC10F220/222

6-Pin, 8-Bit Flash Microcontrollers

Device Included In This Data Sheet:

- PIC10F220
- PIC10F222

High-Performance RISC CPU:

- Only 33 Single-Word Instructions to Learn
- All Single-Cycle Instructions Except for Program Branches which are Two-Cycle
- 12-bit Wide Instructions
- 2-Level Deep Hardware Stack
- Direct, Indirect and Relative Addressing modes for Data and Instructions
- 8-bit Wide Data Path
- 8 Special Function Hardware Registers
- Operating Speed:
  - 500 ns instruction cycle with 8 MHz internal clock
  - 1 μs instruction cycle with 4 MHz internal clock

Special Microcontroller Features:

- 4 or 8 MHz Precision Internal Oscillator:
  - Factory calibrated to ±1%
- In-Circuit Serial Programming™ (ICSP™)
- In-Circuit Debugging (ICD) Support
- Power-On Reset (POR)
- Short Device Reset Timer, DRT (1.125 ms typical)
- Watchdog Timer (WDT) with Dedicated On-Chip RC Oscillator for Reliable Operation
- Programmable Code Protection
- Multiplexed $\overline{MCLR}$ Input Pin
- Internal Weak Pull-Ups on I/O Pins
- Power-Saving Sleep mode
- Wake-up from Sleep on Pin Change

Low-Power Features/CMOS Technology:

- Operating Current:
  - < 175 μA @ 2V, 4 MHz
- Standby Current:
  - 100 nA @ 2V, typical
- Low-Power, High-Speed Flash Technology:
  - 100,000 Flash endurance
  - > 40-year retention
- Fully Static Design
- Wide Operating Voltage Range: 2.0V to 5.5V
- Wide Temperature Range:
  - Industrial: -40°C to +85°C
  - Extended: -40°C to +125°C

Peripheral Features:

- 4 I/O Pins:
  - 3 I/O pins with individual direction control
  - 1 input only pin
  - High current sink/source for direct LED drive
  - Wake-on-change
  - Weak pull-ups
- 8-bit Real-Time Clock/Counter (TMR0) with 8-bit Programmable Prescaler
- Analog-to-Digital (A/D) Converter:
  - 8-bit resolution
  - 2 external input channels
  - 1 internal input channel dedicated

Device	Program Memory	Data Memory	I/O	Timers 8-bit	8-Bit A/D (ch)
	Flash (words)	SRAM (bytes)			
PIC10F220	256	16	4	1	2
PIC10F222	512	23	4	1	2

NOTES:

NOTES:

4.0 MEMORY ORGANIZATION

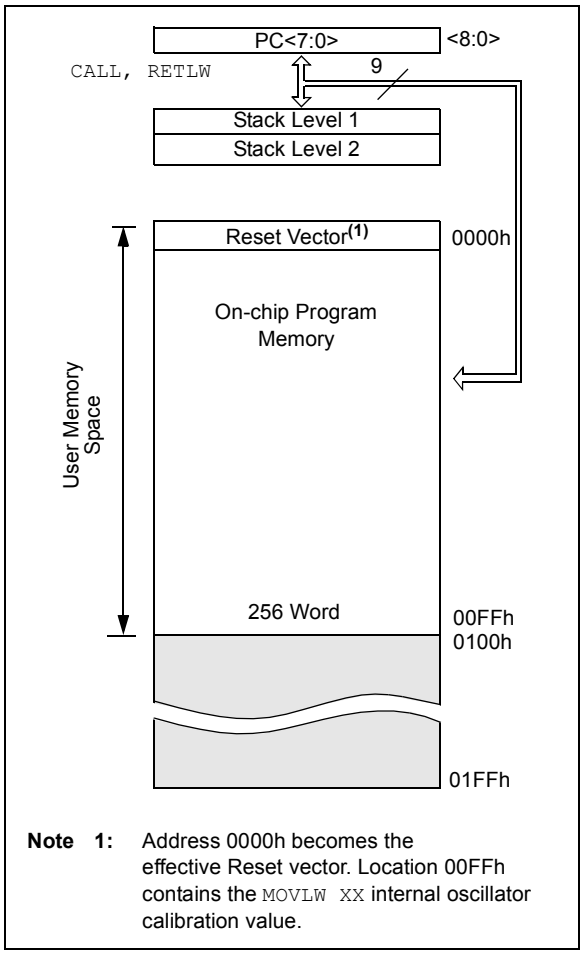
The PIC10F220/222 memories are organized into program memory and data memory. Data memory banks are accessed using the File Select Register (FSR).

4.1 Program Memory Organization for the PIC10F220

The PIC10F220 devices have a 9-bit Program Counter (PC) capable of addressing a 512 x 12 program memory space.

Only the first 256 x 12 (0000h-00FFh) for the PIC10F220 are physically implemented (see Figure 4-1). Accessing a location above these boundaries will cause a wrap-around within the first 256 x 12 space (PIC10F220). The effective Reset vector is at 0000h, (see Figure 4-1). Location 00FFh (PIC10F220) contains the internal clock oscillator calibration value. This value should never be overwritten.

FIGURE 4-1: PROGRAM MEMORY MAP AND STACK FOR THE PIC10F220



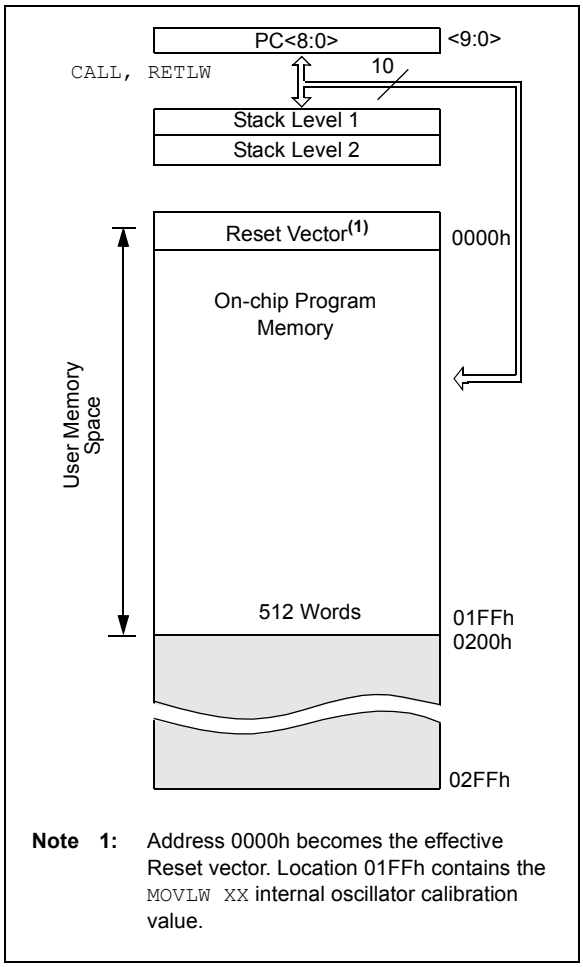
Note 1: Address 0000h becomes the effective Reset vector. Location 00FFh contains the `MOVLW XX` internal oscillator calibration value.

4.2 Program Memory Organization for the PIC10F222

The PIC10F222 devices have a 10-bit Program Counter (PC) capable of addressing a 1024 x 12 program memory space.

Only the first 512 x 12 (0000h-01FFh) for the Mem-High are physically implemented (see Figure 4-2). Accessing a location above these boundaries will cause a wrap-around within the first 512 x 12 space (PIC10F222). The effective Reset vector is at 0000h, (see Figure 4-2). Location 01FFh (PIC10F222) contains the internal clock oscillator calibration value. This value should never be overwritten.

FIGURE 4-2: PROGRAM MEMORY MAP AND STACK FOR THE PIC10F222



Note 1: Address 0000h becomes the effective Reset vector. Location 01FFh contains the `MOVLW XX` internal oscillator calibration value.

REGISTER 4-1: STATUS REGISTER (ADDRESS: 03h)

R/W-0	R/W-0	R/W-0	R-1	R-1	R/W-x	R/W-x	R/W-x
GPWUF	—	—	$\overline{TO}$	$\overline{PD}$	Z	DC	C
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 **GPWUF:** GPIO Reset bit
1 = Reset due to wake-up from Sleep on pin change
0 = After power-up or other Reset

bit 6 **Reserved:** Do not use. Use of this bit may affect upward compatibility with future products.

bit 5 **Reserved:** Do not use. Use of this bit may affect upward compatibility with future products.

bit 4 **$\overline{TO}$:** Time-out bit
1 = After power-up, CLRWDT instruction or SLEEP instruction
0 = A WDT time-out occurred

bit 3 **$\overline{PD}$:** Power-down bit
1 = After power-up or by the CLRWDT instruction
0 = By execution of the SLEEP instruction

bit 2 **Z:** Zero bit
1 = The result of an arithmetic or logic operation is zero
0 = The result of an arithmetic or logic operation is not zero

bit 1 **DC:** Digit carry/$\overline{borrow}$ bit (for ADDWF and SUBWF instructions)
ADDWF:
1 = A carry to the 4th low-order bit of the result occurred
0 = A carry to the 4th low-order bit of the result did not occur
SUBWF:
1 = A borrow from the 4th low-order bit of the result did not occur
0 = A borrow from the 4th low-order bit of the result occurred

bit 0 **C:** Carry/$\overline{borrow}$ bit (for ADDWF, SUBWF and RRF, RLF instructions)

ADDWF:	SUBWF:	RRF or RLF:
1 = A carry occurred	1 = A borrow did not occur	Load bit with LSb or MSb, respectively
0 = A carry did not occur	0 = A borrow occurred	

4.9 Indirect Data Addressing; INDF and FSR Registers

The INDF register is not a physical register. Addressing INDF actually addresses the register whose address is contained in the FSR register (FSR is a *pointer*). This is indirect addressing.

4.9.1 INDIRECT ADDRESSING

- Register file 09 contains the value 10h
- Register file 0A contains the value 0Ah
- Load the value 09 into the FSR register
- A read of the INDF register will return the value of 10h
- Increment the value of the FSR register by one (FSR = 0A)
- A read of the INDR register now will return the value of 0Ah.

Reading INDF itself indirectly (FSR = 0) will produce 00h. Writing to the INDF register indirectly results in a no-operation (although Status bits may be affected).

A simple program to clear RAM locations 10h-1Fh using Indirect addressing is shown in Example 4-1.

EXAMPLE 4-1: HOW TO CLEAR RAM USING INDIRECT ADDRESSING

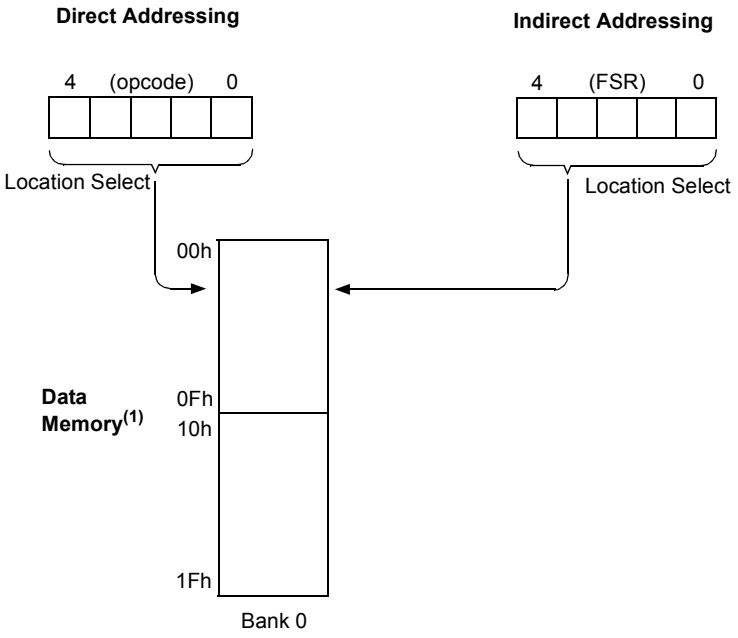
```
        MOVLW   0x10    ;initialize pointer
        MOVWF   FSR     ;to RAM
NEXT    CLRF    INDF    ;clear INDF
                        ;register
        INCF    FSR,F   ;inc pointer
        BTFSC   FSR,4   ;all done?
        GOTO    NEXT    ;NO, clear next
CONTINUE
            :           ;YES, continue
            :
```

The FSR is a 5-bit wide register. It is used in conjunction with the INDF register to indirectly address the data memory area.

The FSR<4:0> bits are used to select data memory addresses 00h to 1Fh.

Note:	Do not use banking. FSR <7:5> are unimplemented and read as '1's.

FIGURE 4-6: DIRECT/INDIRECT ADDRESSING



Note 1: For register map detail, see **Section 4.3 "Data Memory Organization"**.

TABLE 5-3: SUMMARY OF PORT REGISTERS

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-On Reset	Value on All Other Resets
N/A	TRISGPIO	—	—	—	—	I/O Control Registers				---- 1111	---- 1111
N/A	OPTION	$\overline{GPWU}$	$\overline{GPPU}$	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
03h	STATUS	GPWUF	—	—	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	q00q quuu(1)
06h	GPIO	—	—	—	—	GP3	GP2	GP1	GP0	---- xxxx	---- uuuu

Legend: Shaded cells not used by PORT registers, read as '0', – = unimplemented, read as '0', x = unknown, u = unchanged, q = depends on condition.

Note 1: If Reset was due to wake-up on pin change, then bit 7 = 1. All other Resets will cause bit 7 = 0.

5.4 I/O Programming Considerations

5.4.1 BIDIRECTIONAL I/O PORTS

Some instructions operate internally as read followed by write operations. The BCF and BSF instructions, for example, read the entire port into the CPU, execute the bit operation and re-write the result. Caution must be used when these instructions are applied to a port where one or more pins are used as input/outputs. For example, a BSF operation on bit 2 of GPIO will cause all eight bits of GPIO to be read into the CPU, bit 2 to be set and the GPIO value to be written to the output latches. If another bit of GPIO is used as a bidirectional I/O pin (say bit 0) and it is defined as an input at this time, the input signal present on the pin itself would be read into the CPU and rewritten to the data latch of this particular pin, overwriting the previous content. As long as the pin stays in the Input mode, no problem occurs. However, if bit 0 is switched into Output mode later on, the content of the data latch may now be unknown.

Example 5-1 shows the effect of two sequential Read-Modify-Write instructions (e.g., BCF, BSF, etc.) on an I/O port.

A pin actively outputting a high or a low should not be driven from external devices at the same time in order to change the level on this pin ("wired-or", "wired-and"). The resulting high output currents may damage the chip.

EXAMPLE 5-1: I/O PORT READ-MODIFY-WRITE INSTRUCTIONS

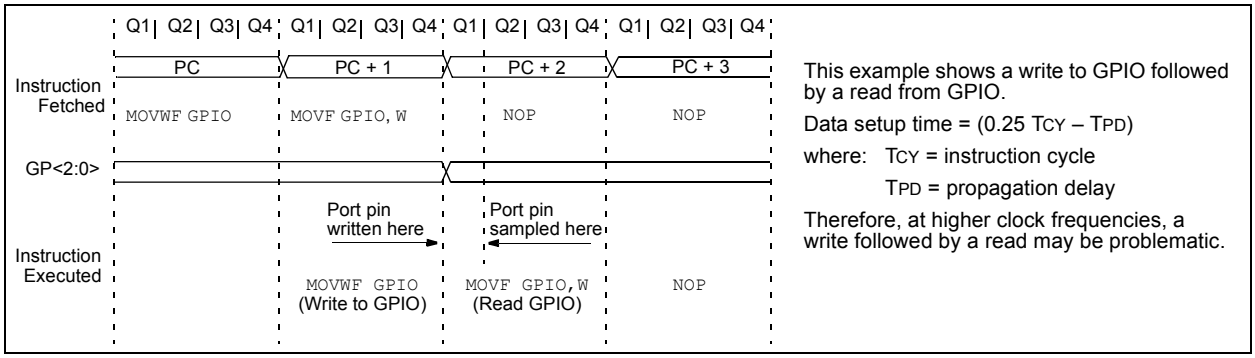
```
;Initial GPIO Settings
;GPIO<3:2> Inputs
;GPIO<1:0> Outputs
;
;                 GPIO latch   GPIO pins
;                 ----------   ----------
 BCF    GPIO,  1 ;---- pp01    ---- pp11
 BCF    GPIO,  0 ;---- pp10    ---- pp11
 MOVLW  007h;
 TRIS   GPIO     ;---- pp10    ---- pp11
;
```

Note: The user may have expected the pin values to be ---- pp00. The second BCF caused GP1 to be latched as the pin value (High).

5.4.2 SUCCESSIVE OPERATIONS ON I/O PORTS

The actual write to an I/O port happens at the end of an instruction cycle, whereas for reading, the data must be valid at the beginning of the instruction cycle (Figure 5-5). Therefore, care must be exercised if a write followed by a read operation is carried out on the same I/O port. The sequence of instructions should allow the pin voltage to stabilize (load dependent) before the next instruction causes that file to be read into the CPU. Otherwise, the previous state of that pin may be read into the CPU rather than the new state. When in doubt, it is better to separate these instructions with a NOP or another instruction not accessing this I/O port.

FIGURE 5-5: SUCCESSIVE I/O OPERATION



This example shows a write to GPIO followed by a read from GPIO.

Data setup time = (0.25 TCY – TPD)

where: TCY = instruction cycle

TPD = propagation delay

Therefore, at higher clock frequencies, a write followed by a read may be problematic.

NOTES:

6.0 TMR0 MODULE AND TMR0 REGISTER

The Timer0 module has the following features:

- 8-bit timer/counter register, TMR0
- Readable and writable
- 8-bit software programmable prescaler
- Internal or external clock select:
  - Edge select for external clock

Figure 6-1 is a simplified block diagram of the Timer0 module.

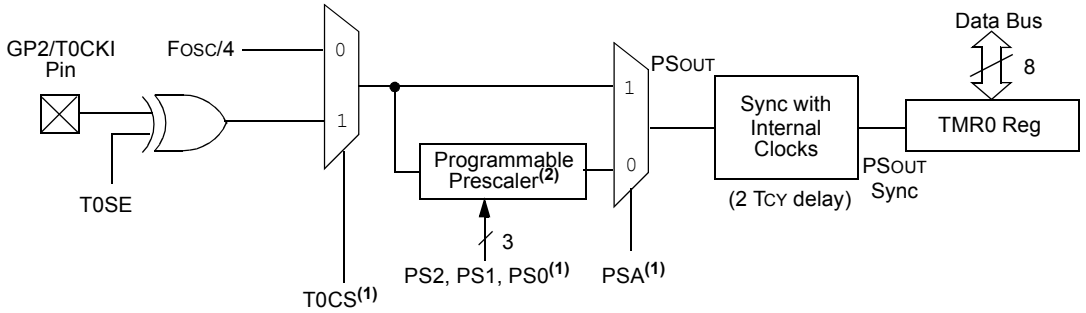
Timer mode is selected by clearing the T0CS bit (OPTION<5>). In Timer mode, the Timer0 module will increment every instruction cycle (without prescaler). If TMR0 register is written, the increment is inhibited for the following two cycles (Figure 6-2 and Figure 6-3). The user can work around this by writing an adjusted value to the TMR0 register.

Counter mode is selected by setting the T0CS bit (OPTION<5>). In this mode, Timer0 will increment either on every rising or falling edge of pin T0CKI. The T0SE bit (OPTION<4>) determines the source edge. Clearing the T0SE bit selects the rising edge. Restrictions on the external clock input are discussed in detail in **Section 6.1 "Using Timer0 With An External Clock"**.

The prescaler may be used by either the Timer0 module or the Watchdog Timer, but not both. The prescaler assignment is controlled in software by the control bit PSA (OPTION<3>). Clearing the PSA bit will assign the prescaler to Timer0. The prescaler is not readable or writable. When the prescaler is assigned to the Timer0 module, prescale values of 1:2, 1:4, 1:256 are selectable. **Section 6.2 "Prescaler"** details the operation of the prescaler.

A summary of registers associated with the Timer0 module is found in Table 6-1.

FIGURE 6-1: TIMER0 BLOCK DIAGRAM



Note 1: Bits T0CS, T0SE, PSA, PS2, PS1 and PS0 are located in the OPTION register.

2: The prescaler is shared with the Watchdog Timer (Figure 6-5).

FIGURE 6-2: TIMER0 TIMING: INTERNAL CLOCK/NO PRESCALE

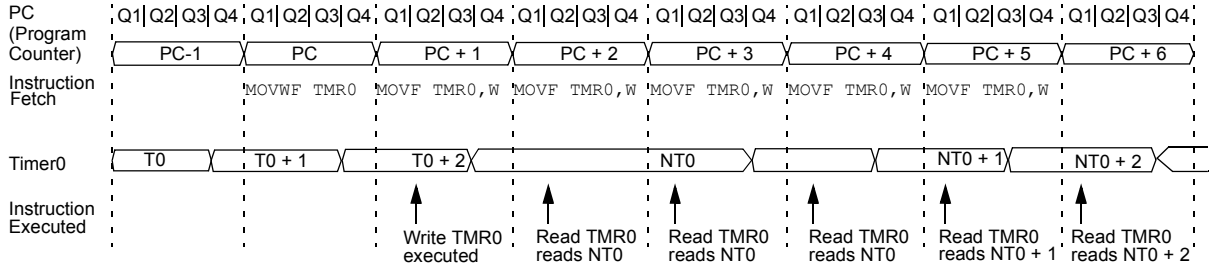


FIGURE 8-2: SIMPLIFIED BLOCK DIAGRAM OF ON-CHIP RESET CIRCUIT

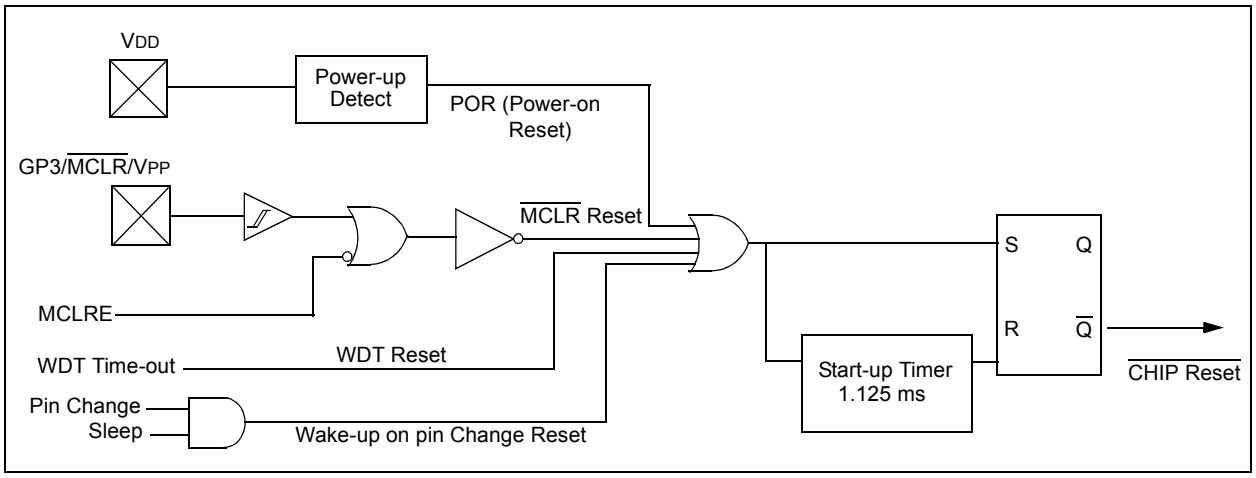


FIGURE 8-3: TIME-OUT SEQUENCE ON POWER-UP (MCLR PULLED LOW)

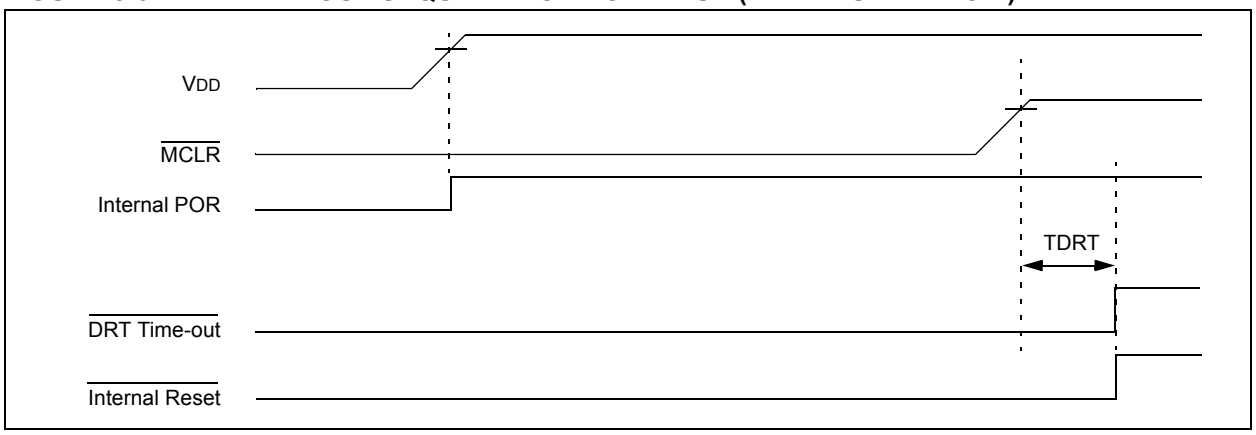
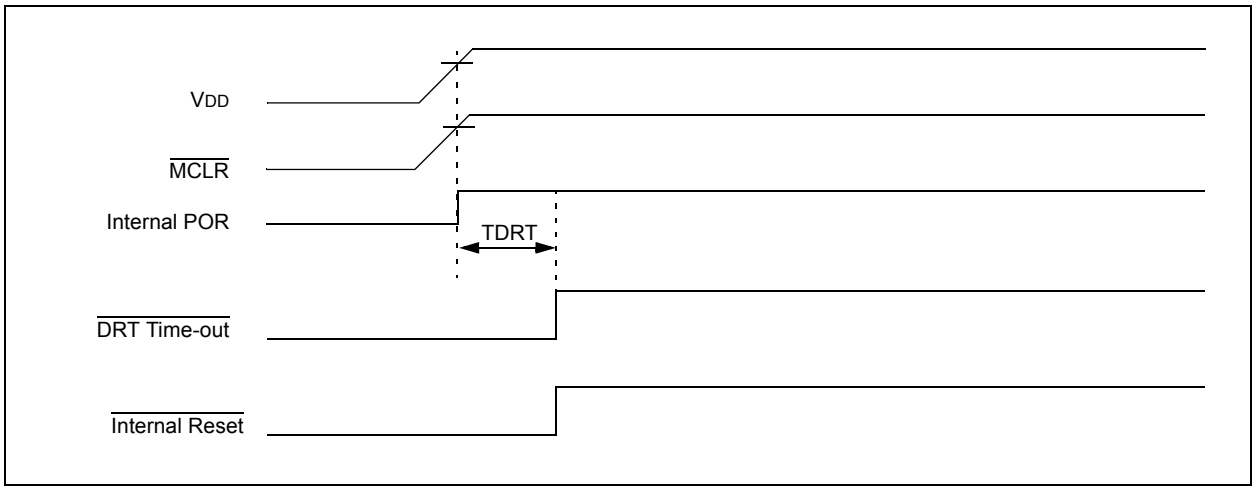


FIGURE 8-4: TIME-OUT SEQUENCE ON POWER-UP (MCLR TIED TO VDD): FAST VDD RISE TIME



BTFSS	Bit Test f, Skip if Set
Syntax:	[*label*] BTFSS f,b
Operands:	$0 \le f \le 31$ $0 \le b < 7$
Operation:	skip if (f<b>) = 1
Status Affected:	None
Description:	If bit 'b' in register 'f' is '1', then the next instruction is skipped. If bit 'b' is '1', then the next instruction fetched during the current instruction execution, is discarded and a NOP is executed instead, making this a two-cycle instruction.

CALL	Subroutine Call
Syntax:	[*label*] CALL k
Operands:	$0 \le k \le 255$
Operation:	(PC) + 1→ Top of Stack; k → PC<7:0>; (Status<6:5>) → PC<10:9>; 0 → PC<8>
Status Affected:	None
Description:	Subroutine call. First, return address (PC + 1) is pushed onto the stack. The eight-bit immediate address is loaded into PC bits <7:0>. The upper bits PC<10:9> are loaded from STATUS<6:5>, PC<8> is cleared. CALL is a two-cycle instruction.

CLRF	Clear f
Syntax:	[*label*] CLRF f
Operands:	$0 \le f \le 31$
Operation:	00h → (f); 1 → Z
Status Affected:	Z
Description:	The contents of register 'f' are cleared and the Z bit is set.

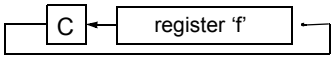
CLRW	Clear W
Syntax:	[*label*] CLRW
Operands:	None
Operation:	00h → (W); 1 → Z
Status Affected:	Z
Description:	The W register is cleared. Zero bit (Z) is set.

CLRWDT	Clear Watchdog Timer
Syntax:	[*label*] CLRWDT k
Operands:	None
Operation:	00h → WDT; 0 → WDT prescaler (if assigned); 1 → $\overline{TO}$; 1 → $\overline{PD}$
Status Affected:	$\overline{TO}$, $\overline{PD}$
Description:	The CLRWDT instruction resets the WDT. It also resets the prescaler, if the prescaler is assigned to the WDT and not Timer0. Status bits $\overline{TO}$ and $\overline{PD}$ are set.

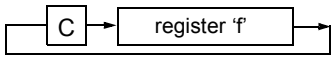
COMF	Complement f
Syntax:	[*label*] COMF f,d
Operands:	$0 \le f \le 31$ $d \in [0,1]$
Operation:	$(\bar{f}) \rightarrow$ (dest)
Status Affected:	Z
Description:	The contents of register 'f' are complemented. If 'd' is '0', the result is stored in the W register. If 'd' is '1', the result is stored back in register 'f'.

RETLW	Return with Literal in W
Syntax:	[*label*] RETLW k
Operands:	$0 \le k \le 255$
Operation:	$k \rightarrow (W)$; $TOS \rightarrow PC$
Status Affected:	None
Description:	The W register is loaded with the eight-bit literal 'k'. The program counter is loaded from the top of the stack (the return address). This is a two-cycle instruction.

RLF	Rotate Left f through Carry
Syntax:	[*label*] RLF f,d
Operands:	$0 \le f \le 31$ $d \in [0,1]$
Operation:	See description below
Status Affected:	C
Description:	The contents of register 'f' are rotated one bit to the left through the Carry Flag. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is stored back in register 'f'.



RRF	Rotate Right f through Carry
Syntax:	[*label*] RRF f,d
Operands:	$0 \le f \le 31$ $d \in [0,1]$
Operation:	See description below
Status Affected:	C
Description:	The contents of register 'f' are rotated one bit to the right through the Carry Flag. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'.



SLEEP	Enter SLEEP Mode
Syntax:	[*label*] SLEEP
Operands:	None
Operation:	00h → WDT; 0 → WDT prescaler; 1 → $\overline{TO}$; 0 → $\overline{PD}$
Status Affected:	$\overline{TO}$, $\overline{PD}$, RBWUF
Description:	Time-out Status bit ($\overline{TO}$) is set. The Power-down Status bit ($\overline{PD}$) is cleared. RBWUF is unaffected. The WDT and its prescaler are cleared. The processor is put into Sleep mode with the oscillator stopped. See section on Sleep for more details.

SUBWF	Subtract W from f
Syntax:	[*label*] SUBWF f,d
Operands:	$0 \le f \le 31$ $d \in [0,1]$
Operation:	(f) – (W) → (dest)
Status Affected:	C, DC, Z
Description:	Subtract (2's complement method) the W register from register 'f'. If 'd' is '0', the result is stored in the W register. If 'd' is '1', the result is stored back in register 'f'.

SWAPF	Swap Nibbles in f
Syntax:	[*label*] SWAPF f,d
Operands:	$0 \le f \le 31$ $d \in [0,1]$
Operation:	(f<3:0>) → (dest<7:4>); (f<7:4>) → (dest<3:0>)
Status Affected:	None
Description:	The upper and lower nibbles of register 'f' are exchanged. If 'd' is '0', the result is placed in W register. If 'd' is '1', the result is placed in register 'f'.

10.2 DC Characteristics: PIC10F220/222 (Extended)

DC CHARACTERISTICS			Standard Operating Conditions (unless otherwise specified) Operating Temperature -40×C £ TA £ +125×C (extended)				
Param No.	**Sym**	**Characteristic**	**Min**	**Typ(1)**	**Max**	**Units**	**Conditions**
D001	VDD	**Supply Voltage**	2.0		5.5	V	See **Figure 10-1**
D002	VDR	**RAM Data Retention Voltage(2)**	1.5*		—	V	Device in Sleep mode
D003	VPOR	**VDD Start Voltage** to ensure Power-on Reset	—	Vss	—	V	
D010	IDD	**Supply Current(3)**					
			—	175	275	µA	VDD = 2.0V, Fosc = 4 MHz
			—	0.625	1.1	mA	VDD = 5.0V, Fosc = 4 MHz
			—	250	400	µA	VDD = 2.0V, Fosc = 8 MHz
			—	0.800	1.5	mA	VDD = 5.0V, Fosc = 8 MHz
D020	IPD	**Power-down Current(4)**					
			—	0.1	9	µA	VDD = 2.0V
			—	1	15	µA	VDD = 5.0V
D022	IWDT	**WDT Current(4)**					
			—	1.0	18	µA	VDD = 2.0V
			—	7	22	µA	VDD = 5.0V

* These parameters are characterized but not tested.

Note 1: Data in the Typical (“Typ”) column is based on characterization results at 25°C. This data is for design guidance only and is not tested.

2: This is the limit to which VDD can be lowered in Sleep mode without losing RAM data.

3: The supply current is mainly a function of the operating voltage and frequency. Other factors such as bus loading, bus rate, internal code execution pattern and temperature also have an impact on the current consumption.

a) The test conditions for all IDD measurements in active operation mode are:
All I/O pins tri-stated, pulled to VSS, T0CKI = VDD, $\overline{MCLR}$ = VDD; WDT enabled/disabled as specified.

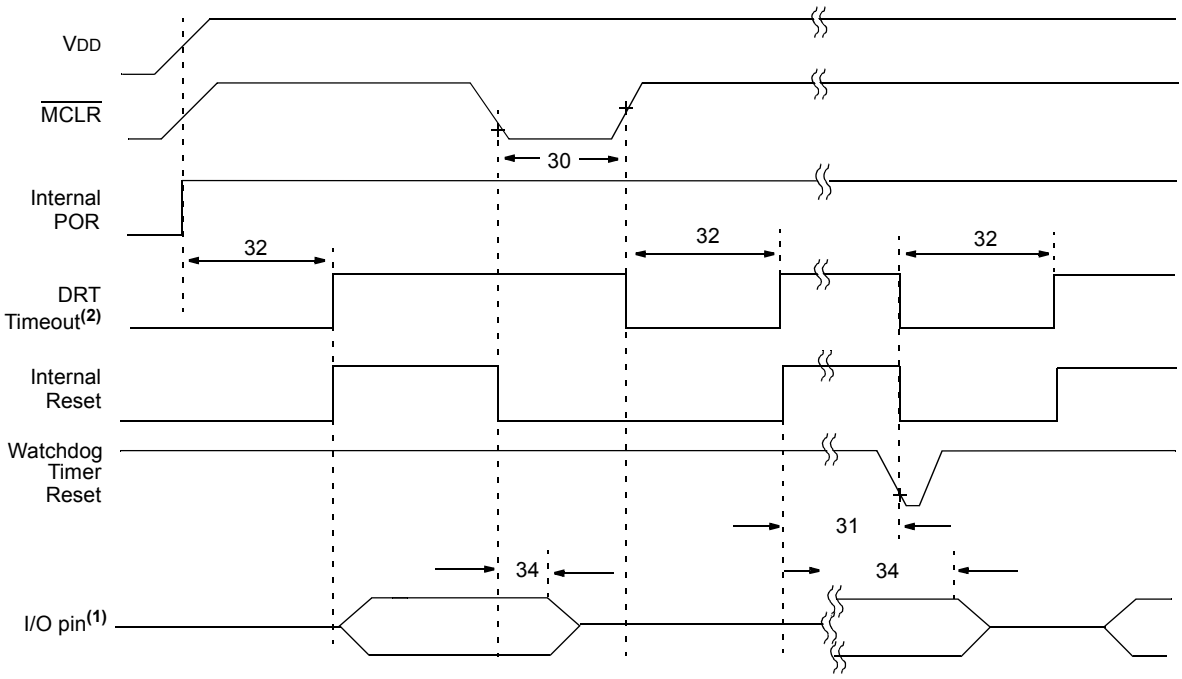
b) For standby current measurements, the conditions are the same, except that the device is in Sleep mode.

4: Power-down current is measured with the part in Sleep mode, with all I/O pins in high-impedance state and tied to VDD or VSS. The peripheral current is the sum of the base IPD and the additional current consumed when the peripheral is enabled.

TABLE 10-1: PULL-UP RESISTOR RANGES

VDD (Volts)	Temperature (°C)	Min	Typ	Max	Units
GP0/GP1					
2.0	-40	73K	105K	186K	Ω
	25	73K	113K	187K	Ω
	85	82K	123K	190K	Ω
	125	86K	132k	190K	Ω
5.5	-40	15K	21K	33K	Ω
	25	15K	22K	34K	Ω
	85	19K	26k	35K	Ω
	125	23K	29K	35K	Ω
GP3					
2.0	-40	63K	81K	96K	Ω
	25	77K	93K	116K	Ω
	85	82K	96k	116K	Ω
	125	86K	100K	119K	Ω
5.5	-40	16K	20k	22K	Ω
	25	16K	21K	23K	Ω
	85	24K	25k	28K	Ω
	125	26K	27K	29K	Ω

FIGURE 10-3: RESET, WATCHDOG TIMER AND DEVICE RESET TIMER TIMING



Note 1: I/O pins must be taken out of High-impedance mode by enabling the output drivers in software.
2: Runs on POR Reset only.

TABLE 10-3: RESET, WATCHDOG TIMER AND DEVICE RESET TIMER – PIC10F220/222

AC CHARACTERISTICS			Standard Operating Conditions (unless otherwise specified) Operating Temperature -40°C ≤ TA ≤ +85°C (industrial) -40°C ≤ TA ≤ +125°C (extended) Operating Voltage VDD range is described in **Section 10.1 "DC Characteristics: PIC10F220/222 (Industrial)"**				
Param No.	**Sym**	**Characteristic**	**Min**	**Typ(1)**	**Max**	**Units**	**Conditions**
30	T_{MCL}	$\overline{MCLR}$ Pulse Width (low)	2* 5*	— —	— —	µs µs	VDD = 5V, -40°C to +85°C VDD = 5.0V
31	TWDT	Watchdog Timer Time-out Period (no prescaler)	10 10	18 18	29 31	ms ms	VDD = 5.0V (Industrial) VDD = 5.0V (Extended)
32	TDRT*	Device Reset Timer Period (standard)	0.600 0.600	1.125 1.125	1.85 1.95	ms ms	VDD = 5.0V (Industrial) VDD = 5.0V (Extended)
34	TIOZ	I/O High-impedance from $\overline{MCLR}$ low	—	—	2*	µs	

* These parameters are characterized but not tested.

Note 1: Data in the Typical ("Typ") column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

FIGURE 10-4: TIMER0 CLOCK TIMINGS

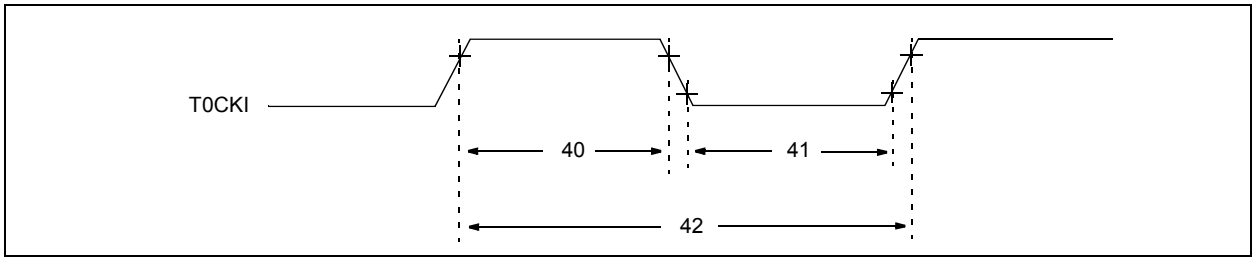


TABLE 10-4: TIMER0 CLOCK REQUIREMENTS

AC CHARACTERISTICS				**Standard Operating Conditions (unless otherwise specified)** Operating Temperature -40°C ≤ TA ≤ +85°C (industrial) -40°C ≤ TA ≤ +125°C (extended)				
Param No.	**Sym**	**Characteristic**		**Min**	**Typ(1)**	**Max**	**Units**	**Conditions**
40	Tt0H	T0CKI High Pulse Width	No Prescaler	0.5 TCY + 20*	—	—	ns	
			With Prescaler	10*	—	—	ns	
41	Tt0L	T0CKI Low Pulse Width	No Prescaler	0.5 TCY + 20*	—	—	ns	
			With Prescaler	10*	—	—	ns	
42	Tt0P	T0CKI Period		20 or TCY + 40* N	—	—	ns	Whichever is greater. N = Prescale Value (1, 2, 4,..., 256)

* These parameters are characterized but not tested.

Note 1: Data in the Typical (“Typ”) column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

FIGURE 11-8: VOL vs. IOL OVER TEMPERATURE (VDD = 3.0V)

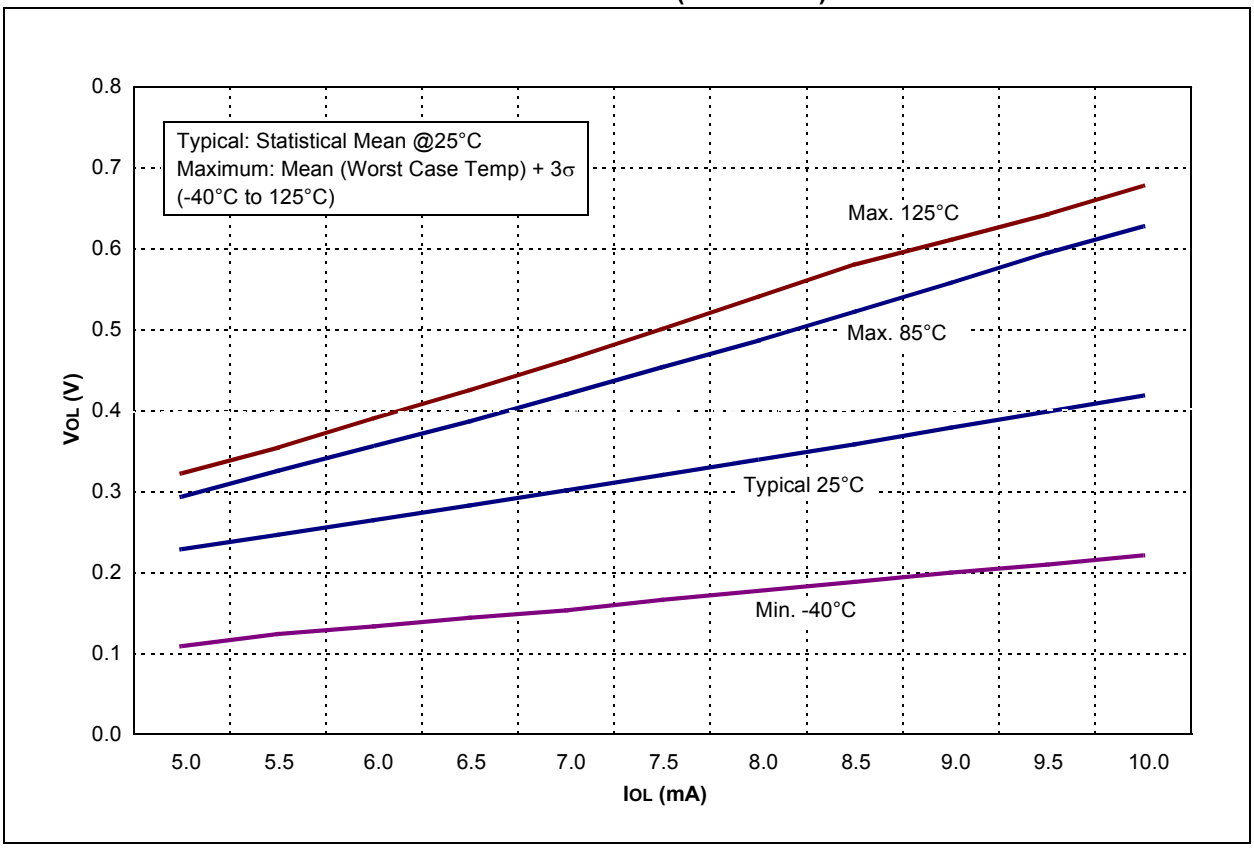
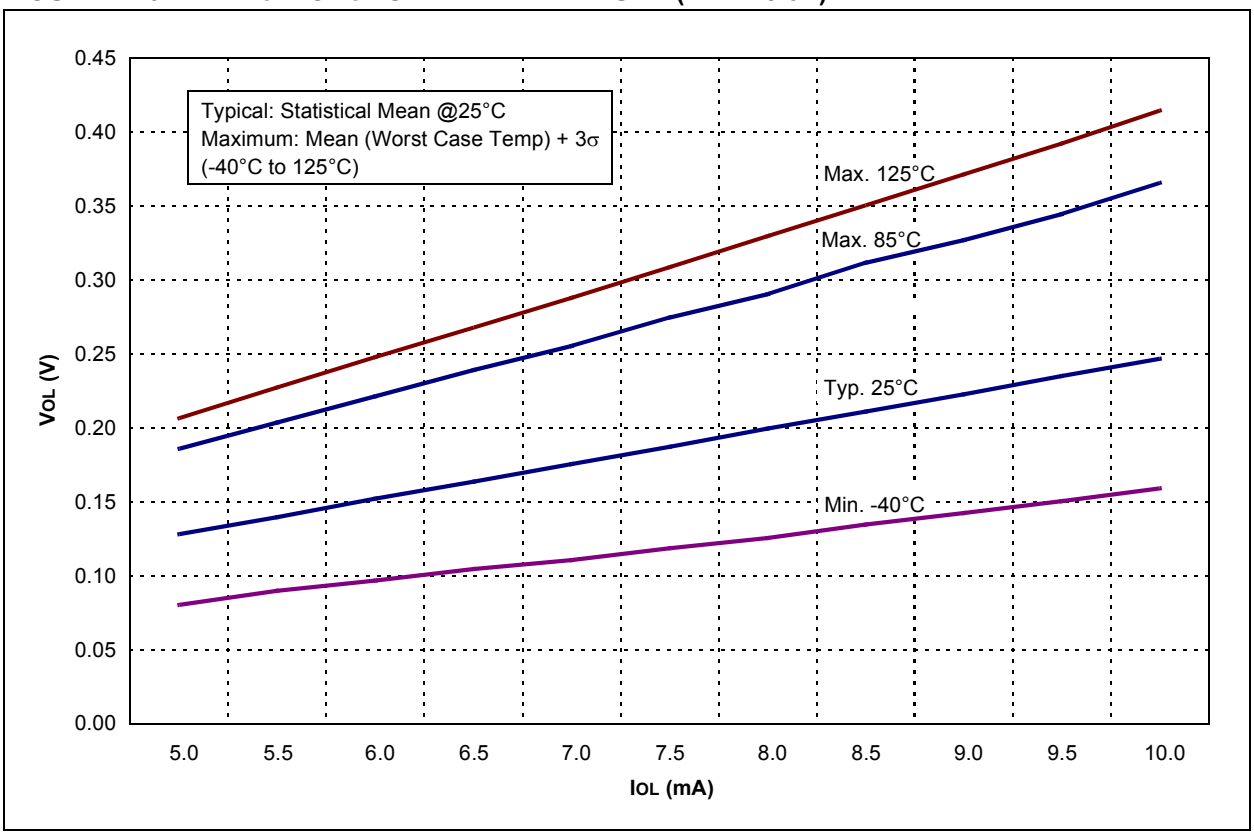


FIGURE 11-9: VOL vs. IOL OVER TEMPERATURE (VDD = 5.0V)



12.6 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.

12.7 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.

12.8 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.

12.9 PICkit 3 In-Circuit Debugger/ Programmer

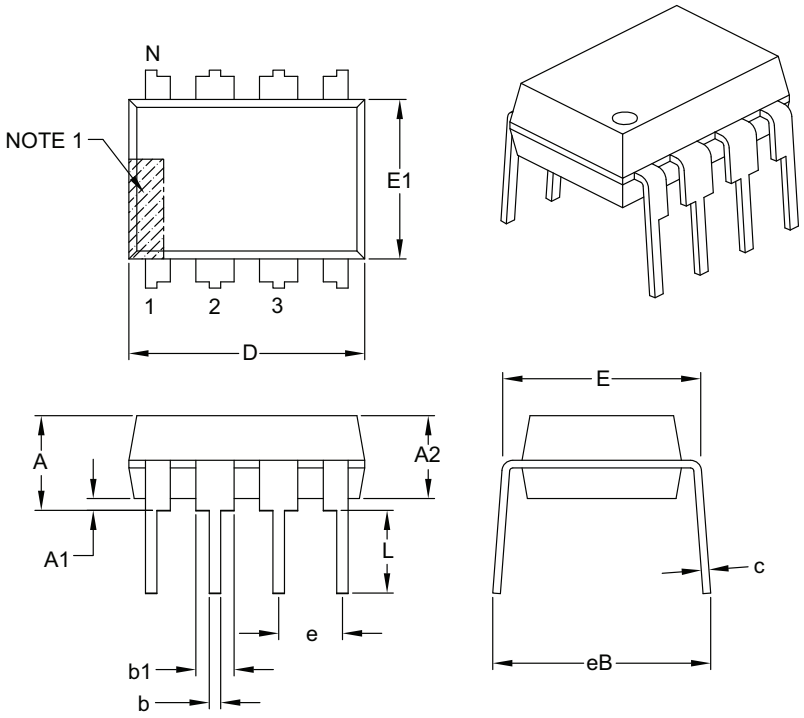
The MPLAB PICkit 3 allows debugging and programming of PIC and dsPIC Flash microcontrollers at a most affordable price point using the powerful graphical user interface of the MPLAB IDE. The MPLAB PICkit 3 is connected to the design engineer's PC using a full-speed USB interface and can be connected 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™).

12.10 MPLAB PM3 Device Programmer

The MPLAB PM3 Device Programmer is a universal, CE compliant device programmer with programmable voltage verification at V_{DDMIN} and V_{DDMAX} 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 Dual In-Line (P) – 300 mil Body [PDIP]

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



Units		INCHES		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N		8	
Pitch	e		.100 BSC	
Top to Seating Plane	A	–	–	.210
Molded Package Thickness	A2	.115	.130	.195
Base to Seating Plane	A1	.015	–	–
Shoulder to Shoulder Width	E	.290	.310	.325
Molded Package Width	E1	.240	.250	.280
Overall Length	D	.348	.365	.400
Tip to Seating Plane	L	.115	.130	.150
Lead Thickness	c	.008	.010	.015
Upper Lead Width	b1	.040	.060	.070
Lower Lead Width	b	.014	.018	.022
Overall Row Spacing §	eB	–	–	.430

Notes:

1. Pin 1 visual index feature may vary, but must be located with the hatched area.
2. § Significant Characteristic.
3. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" per side.
4. Dimensioning and tolerancing per ASME Y14.5M.

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

Microchip Technology Drawing C04-018B