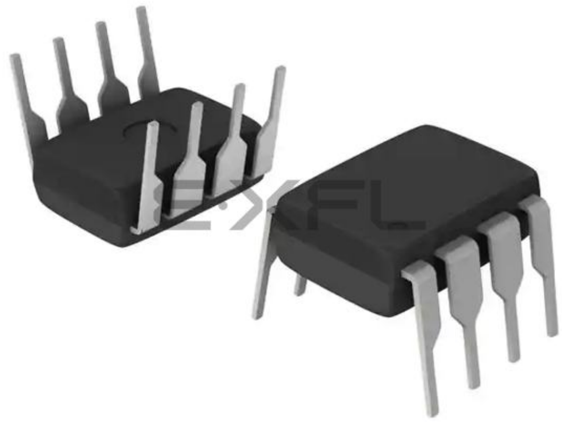


Welcome to E-XFL.COM

## 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 | 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) | 2V ~ 5V |
| Data Converters | A/D 4x10b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 125°C (TA) |
| Mounting Type | Through Hole |
| Package / Case | 8-DIP (0.300", 7.62mm) |
| Supplier Device Package | 8-PDIP |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/pic12hv615-e-p |

## 2.0 MEMORY ORGANIZATION

### 2.1 Program Memory Organization

The PIC12F609/615/617/12HV609/615 has a 13-bit program counter capable of addressing an 8K x 14 program memory space. Only the first 1K x 14 (0000h-03FFh) for the PIC12F609/615/12HV609/615 is physically implemented. For the PIC12F617, the first 2K x 14 (0000h-07FFh) is physically implemented. Accessing a location above these boundaries will cause a wrap-around within the first 1K x 14 space for PIC12F609/615/12HV609/615 devices, and within the first 2K x 14 space for the PIC12F617 device. The Reset vector is at 0000h and the interrupt vector is at 0004h (see Figure 2-1).

**FIGURE 2-1: PROGRAM MEMORY MAP AND STACK FOR THE PIC12F609/615/12HV609/615**

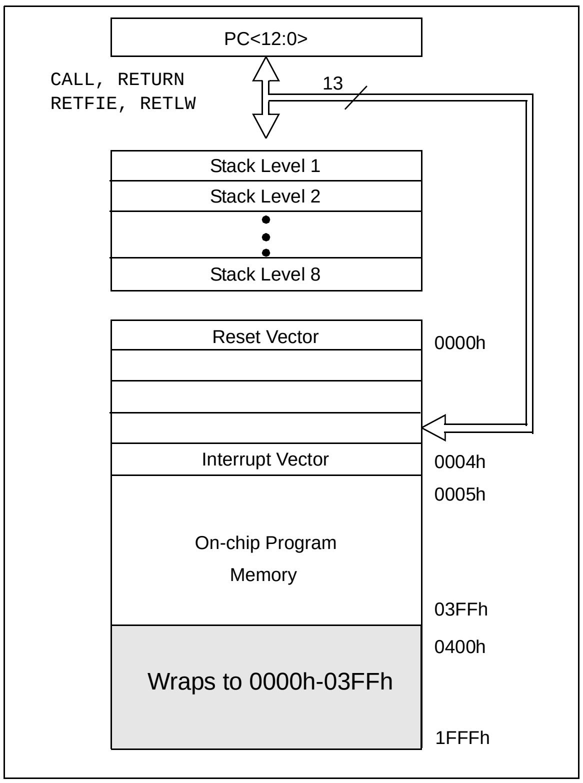
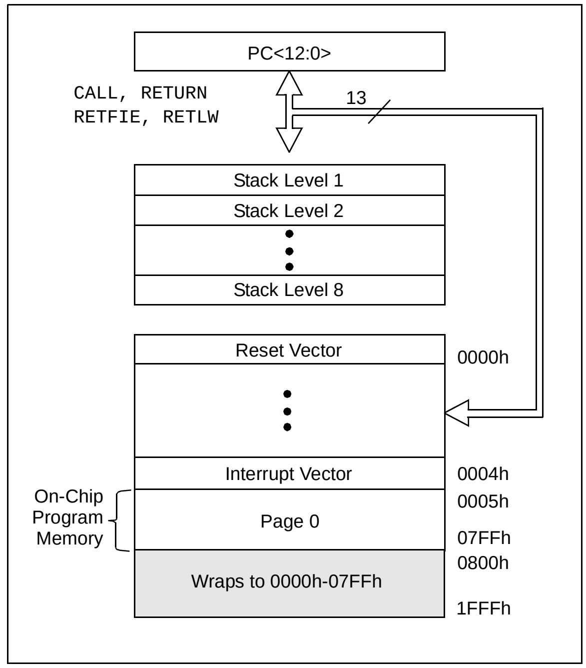



**FIGURE 2-2: PROGRAM MEMORY MAP AND STACK FOR THE PIC12F617**




### 2.2 Data Memory Organization

The data memory (see Figure 2-3) is partitioned into two banks, which contain the General Purpose Registers (GPR) and the Special Function Registers (SFR). The Special Function Registers are located in the first 32 locations of each bank. Register locations 40h-7Fh in Bank 0 are General Purpose Registers, implemented as static RAM. For the PIC12F617, the register locations 20h-7Fh in Bank 0 and A0h-EFh in Bank 1 are general purpose registers implemented as Static RAM. Register locations F0h-FFh in Bank 1 point to addresses 70h-7Fh in Bank 0. All other RAM is unimplemented and returns '0' when read. The RP0 bit of the STATUS register is the bank select bit.

<u>RP0</u>

`0` → Bank 0 is selected

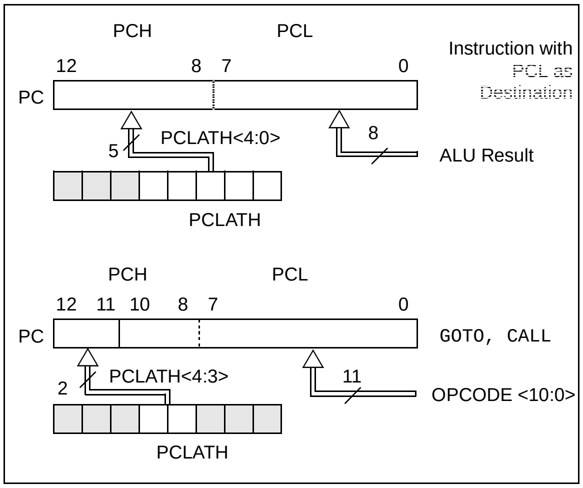
`1` → Bank 1 is selected

| **Note:** | The IRP and RP1 bits of the STATUS register are reserved and should always be maintained as '0's. |
|---|---|

## 2.3 PCL and PCLATH

The Program Counter (PC) is 13 bits wide. The low byte comes from the PCL register, which is a readable and writable register. The high byte (PC<12:8>) is not directly readable or writable and comes from PCLATH. On any Reset, the PC is cleared. Figure 2-5 shows the two situations for the loading of the PC. The upper example in Figure 2-5 shows how the PC is loaded on a write to PCL (PCLATH<4:0> $\rightarrow$ PCH). The lower example in Figure 2-5 shows how the PC is loaded during a `CALL` or `GOTO` instruction (PCLATH<4:3> $\rightarrow$ PCH).

**FIGURE 2-5: LOADING OF PC IN DIFFERENT SITUATIONS**




### 2.3.1 MODIFYING PCL

Executing any instruction with the PCL register as the destination simultaneously causes the Program Counter PC<12:8> bits (PCH) to be replaced by the contents of the PCLATH register. This allows the entire contents of the program counter to be changed by writing the desired upper 5 bits to the PCLATH register. When the lower 8 bits are written to the PCL register, all 13 bits of the program counter will change to the values contained in the PCLATH register and those being written to the PCL register.

A computed `GOTO` is accomplished by adding an offset to the program counter (`ADDWF PCL`). Care should be exercised when jumping into a look-up table or program branch table (computed `GOTO`) by modifying the PCL register. Assuming that PCLATH is set to the table start address, if the table length is greater than 255 instructions or if the lower 8 bits of the memory address rolls over from 0xFF to 0x00 in the middle of the table, then PCLATH must be incremented for each address rollover that occurs between the table beginning and the target location within the table.

For more information refer to Application Note AN556, "*Implementing a Table Read*" (DS00556).

### 2.3.2 STACK

The PIC12F609/615/617/12HV609/615 Family has an 8-level x 13-bit wide hardware stack (see Figure 2-1). The stack space is not part of either program or data space and the Stack Pointer is not readable or writable. The PC is PUSHed onto the stack when a `CALL` instruction is 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. This means that after the stack has been PUSHed eight times, the ninth push overwrites the value that was stored from the first push. The tenth push overwrites the second push (and so on).

| | |
|---|---|
| **Note 1:** | There are no Status bits to indicate stack overflow or stack underflow conditions. |
| **2:** | There are no instructions/mnemonics called PUSH or POP. These are actions that occur from the execution of the `CALL, RETURN, RETLW` and `RETFIE` instructions or the vectoring to an interrupt address. |

## 2.4 Indirect Addressing, INDF and FSR Registers

The INDF register is not a physical register. Addressing the INDF register will cause indirect addressing.

Indirect addressing is possible by using the INDF register. Any instruction using the INDF register actually accesses data pointed to by the File Select Register (FSR). Reading INDF itself indirectly will produce 00h. Writing to the INDF register indirectly results in a no operation (although Status bits may be affected). An effective 9-bit address is obtained by concatenating the 8-bit FSR and the IRP bit of the STATUS register, as shown in Figure 2-6.

A simple program to clear RAM location 40h-7Fh using indirect addressing is shown in Example 2-1.

**EXAMPLE 2-1: INDIRECT ADDRESSING**

```
         MOVLW   0x40    ;initialize pointer
         MOVWF   FSR     ;to RAM
NEXT     CLRF    INDF    ;clear INDF register
         INCF    FSR     ;inc pointer
         BTFSS   FSR,7   ;all done?
         GOTO    NEXT    ;no clear next
CONTINUE                 ;yes continue
```

### 3.3 Reading the Flash Program Memory

To read a program memory location, the user must write two bytes of the address to the PMADRL and PMADRH registers, and then set control bit RD (PMCON1<0>). Once the read control bit is set, the program memory Flash controller will use the second instruction cycle after to read the data. This causes the second instruction immediately following the "`BSF PMCON1,RD`" instruction to be ignored. The data is available in the very next cycle in the PMDATL and PMDATH registers; it can be read as two bytes in the following instructions. PMDATL and PMDATH registers will hold this value until another read or until it is written to by the user (during a write operation).

**EXAMPLE 3-1: FLASH PROGRAM READ**

```
    BANKSEL PM_ADR             ; Change STATUS bits RP1:0 to select bank with PMADRL
    MOVLW   MS_PROG_PM_ADDR    ;
    MOVWF   PMADRH             ; MS Byte of Program Address to read
    MOVLW   LS_PROG_PM_ADDR    ;
    MOVWF   PMADRL             ; LS Byte of Program Address to read
    BANKSEL PMCON1             ; Bank to containing PMCON1
    BSF     PMCON1, RD         ; PM Read

    NOP                        ; First instruction after BSF PMCON1,RD executes normally

    NOP                        ; Any instructions here are ignored as program
                               ; memory is read in second cycle after BSF PMCON1,RD
                               ;
    BANKSEL PMDATL             ; Bank to containing PMADRL
    MOVF    PMDATL, W          ; W = LS Byte of Program PMDATL
    MOVF    PMDATH, W          ; W = MS Byte of Program PMDATL
```

### 7.10 ECCP Special Event Trigger (PIC12F615/617/HV615 only)

If a ECCP is configured to trigger a special event, the trigger will clear the TMR1H:TMR1L register pair. This special event does not cause a Timer1 interrupt. The ECCP module may still be configured to generate a ECCP interrupt.

In this mode of operation, the CCPR1H:CCPR1L register pair effectively becomes the period register for Timer1.

Timer1 should be synchronized to the FOSC to utilize the Special Event Trigger. Asynchronous operation of Timer1 can cause a Special Event Trigger to be missed.

In the event that a write to TMR1H or TMR1L coincides with a Special Event Trigger from the ECCP, the write will take precedence.

For more information, see **Section 11.0 “Enhanced Capture/Compare/PWM (With Auto-Shutdown and Dead Band) Module (PIC12F615/617/HV615 only)”**.

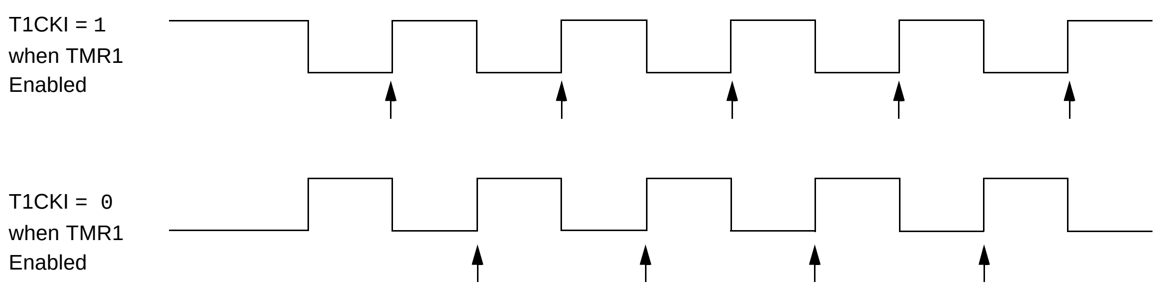
### 7.11 Comparator Synchronization

The same clock used to increment Timer1 can also be used to synchronize the comparator output. This feature is enabled in the Comparator module.

When using the comparator for Timer1 gate, the comparator output should be synchronized to Timer1. This ensures Timer1 does not miss an increment if the comparator changes.

For more information, see **Section 9.0 “Comparator Module”**.

**FIGURE 7-2: TIMER1 INCREMENTING EDGE**




**Note 1:** Arrows indicate counter increments.

**2:** In Counter mode, a falling edge must be registered by the counter prior to the first incrementing rising edge of the clock.

## 8.0 TIMER2 MODULE (PIC12F615/617/HV615 ONLY)

The Timer2 module is an 8-bit timer with the following features:

- 8-bit timer register (TMR2)
- 8-bit period register (PR2)
- Interrupt on TMR2 match with PR2
- Software programmable prescaler (1:1, 1:4, 1:16)
- Software programmable postscaler (1:1 to 1:16)

See Figure 8-1 for a block diagram of Timer2.

### 8.1 Timer2 Operation

The clock input to the Timer2 module is the system instruction clock (FOSC/4). The clock is fed into the Timer2 prescaler, which has prescale options of 1:1, 1:4 or 1:16. The output of the prescaler is then used to increment the TMR2 register.

The values of TMR2 and PR2 are constantly compared to determine when they match. TMR2 will increment from 00h until it matches the value in PR2. When a match occurs, two things happen:

- TMR2 is reset to 00h on the next increment cycle.
- The Timer2 postscaler is incremented

The match output of the Timer2/PR2 comparator is then fed into the Timer2 postscaler. The postscaler has postscale options of 1:1 to 1:16 inclusive. The output of the Timer2 postscaler is used to set the TMR2IF interrupt flag bit in the PIR1 register.

The TMR2 and PR2 registers are both fully readable and writable. On any Reset, the TMR2 register is set to 00h and the PR2 register is set to FFh.

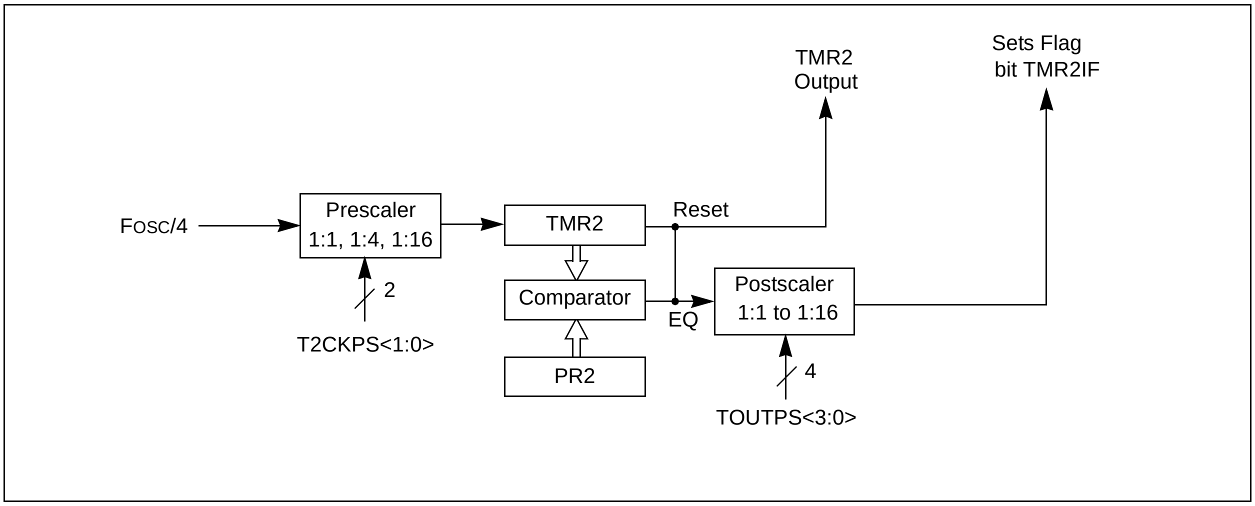
Timer2 is turned on by setting the TMR2ON bit in the T2CON register to a ‘1’. Timer2 is turned off by clearing the TMR2ON bit to a ‘0’.

The Timer2 prescaler is controlled by the T2CKPS bits in the T2CON register. The Timer2 postscaler is controlled by the TOUTPS bits in the T2CON register. The prescaler and postscaler counters are cleared when:

- A write to TMR2 occurs.
- A write to T2CON occurs.
- Any device Reset occurs (Power-on Reset, $\overline{\text{MCLR}}$ Reset, Watchdog Timer Reset, or Brown-out Reset).

| | |
|---|---|
| **Note:** | TMR2 is not cleared when T2CON is written. |

**FIGURE 8-1: TIMER2 BLOCK DIAGRAM**



### 10.2.7 ADC REGISTER DEFINITIONS

The following registers are used to control the operation of the ADC.

**REGISTER 10-1: ADCON0: A/D CONTROL REGISTER 0**

| R/W-0 | R/W-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| ADFM | VCFG | — | CHS2 | CHS1 | CHS0 | GO/$\overline{\text{DONE}}$ | ADON |
| 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 **ADFM:** A/D Conversion Result Format Select bit
1 = Right justified
0 = Left justified

bit 6 **VCFG:** Voltage Reference bit
1 = VREF pin
0 = VDD

bit 5 **Unimplemented:** Read as ‘0’

bit 4-2 **CHS<2:0>: Analog Channel Select bits**
000 = Channel 00 (AN0)
001 = Channel 01 (AN1)
010 = Channel 02 (AN2)
011 = Channel 03 (AN3)
100 = CVREF
101 = 0.6V Reference
110 = 1.2V Reference
111 = Reserved. Do not use.

bit 1 **GO/$\overline{\text{DONE}}$:** A/D Conversion Status bit
1 = A/D conversion cycle in progress. Setting this bit starts an A/D conversion cycle.
This bit is automatically cleared by hardware when the A/D conversion has completed.
0 = A/D conversion completed/not in progress

bit 0 **ADON:** ADC Enable bit
1 = ADC is enabled
0 = ADC is disabled and consumes no operating current

**Note 1:** When the CHS<2:0> bits change to select the 1.2V or 0.6V reference, the reference output voltage will have a transient. If the Comparator module uses this 0.6V reference voltage, the comparator output may momentarily change state due to the transient.

**TABLE 11-2: SUMMARY OF REGISTERS ASSOCIATED WITH CAPTURE**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR, BOR | Value on all other Resets |
|---|---|---|---|---|---|---|---|---|---|---|
| CCP1CON | P1M | — | DC1B1 | DC1B0 | CCP1M3 | CCP1M2 | CCP1M1 | CCP1M0 | `0-00 0000` | `0-00 0000` |
| CCPR1L | Capture/Compare/PWM Register 1 Low Byte | | | | | | | | `xxxx xxxx` | `uuuu uuuu` |
| CCPR1H | Capture/Compare/PWM Register 1 High Byte | | | | | | | | `xxxx xxxx` | `uuuu uuuu` |
| INTCON | GIE | PEIE | T0IE | INTE | GPIE | T0IF | INTF | GPIF | `0000 0000` | `0000 0000` |
| PIE1 | — | ADIE(1) | CCP1IE(1) | — | CMIE | — | TMR2IE(1) | TMR1IE | `-00- 0-00` | `-00- 0-00` |
| PIR1 | — | ADIF(1) | CCP1IF(1) | — | CMIF | — | TMR2IF(1) | TMR1IF | `-00- 0-00` | `-00- 0-00` |
| T1CON | T1GINV | TMR1GE | T1CKPS1 | T1CKPS0 | T1OSCEN | $\overline{\text{T1SYNC}}$ | TMR1CS | TMR1ON | `0000 0000` | `uuuu uuuu` |
| TMR1L | Holding Register for the Least Significant Byte of the 16-bit TMR1 Register | | | | | | | | `xxxx xxxx` | `uuuu uuuu` |
| TMR1H | Holding Register for the Most Significant Byte of the 16-bit TMR1 Register | | | | | | | | `xxxx xxxx` | `uuuu uuuu` |
| TRISIO | — | — | TRISIO5 | TRISIO4 | TRISIO3 | TRISIO2 | TRISIO1 | TRISIO0 | `--11 1111` | `--11 1111` |

**Legend:** - = Unimplemented locations, read as '0', u = unchanged, x = unknown. Shaded cells are not used by the Capture.

**Note 1:** For PIC12F615/617/HV615 only.

### 11.3.4 OPERATION IN SLEEP MODE

In Sleep mode, the TMR2 register will not increment and the state of the module will not change. If the CCP1 pin is driving a value, it will continue to drive that value. When the device wakes up, TMR2 will continue from its previous state.

### 11.3.5 CHANGES IN SYSTEM CLOCK FREQUENCY

The PWM frequency is derived from the system clock frequency. Any changes in the system clock frequency will result in changes to the PWM frequency. See **Section 4.0 "Oscillator Module"** for additional details.

### 11.3.6 EFFECTS OF RESET

Any Reset will force all ports to Input mode and the CCP registers to their Reset states.

### 11.3.7 SETUP FOR PWM OPERATION

The following steps should be taken when configuring the CCP module for PWM operation:

1. Disable the PWM pin (CCP1) output drivers by setting the associated TRIS bit.
2. Set the PWM period by loading the PR2 register.
3. Configure the CCP module for the PWM mode by loading the CCP1CON register with the appropriate values.
4. Set the PWM duty cycle by loading the CCPR1L register and DC1B bits of the CCP1CON register.
5. Configure and start Timer2:
   - Clear the TMR2IF interrupt flag bit of the PIR1 register.
   - Set the Timer2 prescale value by loading the T2CKPS bits of the T2CON register.
   - Enable Timer2 by setting the TMR2ON bit of the T2CON register.
6. Enable PWM output after a new PWM cycle has started:
   - Wait until Timer2 overflows (TMR2IF bit of the PIR1 register is set).
   - Enable the CCP1 pin output driver by clearing the associated TRIS bit.

**NOTES:**

## 12.4 Interrupts

The PIC12F609/615/617/12HV609/615 has 8 sources of interrupt:

- External Interrupt GP2/INT
- Timer0 Overflow Interrupt
- GPIO Change Interrupts
- Comparator Interrupt
- A/D Interrupt (PIC12F615/617/HV615 only)
- Timer1 Overflow Interrupt
- Timer2 Match Interrupt (PIC12F615/617/HV615 only)
- Enhanced CCP Interrupt (PIC12F615/617/HV615 only)
- Flash Memory Self Write (PIC12F617 only)

The Interrupt Control register (INTCON) and Peripheral Interrupt Request Register 1 (PIR1) record individual interrupt requests in flag bits. The INTCON register also has individual and global interrupt enable bits.

The Global Interrupt Enable bit, GIE of the INTCON register, enables (if set) all unmasked interrupts, or disables (if cleared) all interrupts. Individual interrupts can be disabled through their corresponding enable bits in the INTCON register and PIE1 register. GIE is cleared on Reset.

When an interrupt is serviced, the following actions occur automatically:

- The GIE is cleared to disable any further interrupt.
- The return address is pushed onto the stack.
- The PC is loaded with 0004h.

The Return from Interrupt instruction, `RETFIE`, exits the interrupt routine, as well as sets the GIE bit, which re-enables unmasked interrupts.

The following interrupt flags are contained in the INTCON register:

- INT Pin Interrupt
- GPIO Change Interrupt
- Timer0 Overflow Interrupt

The peripheral interrupt flags are contained in the special register, PIR1. The corresponding interrupt enable bit is contained in special register, PIE1.

The following interrupt flags are contained in the PIR1 register:

- A/D Interrupt
- Comparator Interrupt
- Timer1 Overflow Interrupt
- Timer2 Match Interrupt
- Enhanced CCP Interrupt

For external interrupt events, such as the INT pin or GPIO change interrupt, the interrupt latency will be three or four instruction cycles. The exact latency depends upon when the interrupt event occurs (see Figure 12-8). The latency is the same for one or two-cycle instructions. Once in the Interrupt Service Routine, the source(s) of the interrupt can be determined by polling the interrupt flag bits. The interrupt flag bit(s) must be cleared in software before re-enabling interrupts to avoid multiple interrupt requests.

| | |
|---|---|
| **Note 1:** | Individual interrupt flag bits are set, regardless of the status of their corresponding mask bit or the GIE bit. |
| **2:** | When an instruction that clears the GIE bit is executed, any interrupts that were pending for execution in the next cycle are ignored. The interrupts, which were ignored, are still pending to be serviced when the GIE bit is set again. |

For additional information on Timer1, Timer2, comparators, ADC, Enhanced CCP modules, refer to the respective peripheral section.

### 12.4.1 GP2/INT INTERRUPT

The external interrupt on the GP2/INT pin is edge-triggered; either on the rising edge if the INTEDG bit of the OPTION register is set, or the falling edge, if the INTEDG bit is clear. When a valid edge appears on the GP2/INT pin, the INTF bit of the INTCON register is set. This interrupt can be disabled by clearing the INTE control bit of the INTCON register. The INTF bit must be cleared by software in the Interrupt Service Routine before re-enabling this interrupt. The GP2/INT interrupt can wake-up the processor from Sleep, if the INTE bit was set prior to going into Sleep. See **Section 12.7 “Power-Down Mode (Sleep)”** for details on Sleep and Figure 12-9 for timing of wake-up from Sleep through GP2/INT interrupt.

| | |
|---|---|
| **Note:** | The ANSEL register must be initialized to configure an analog channel as a digital input. Pins configured as analog inputs will read ‘0’ and cannot generate an interrupt. |

## 15.11 PICkit 2 Development Programmer/Debugger and PICkit 2 Debug Express

The PICkit™ 2 Development Programmer/Debugger is a low-cost development tool with an easy to use interface for programming and debugging Microchip's Flash families of microcontrollers. The full featured Windows® programming interface supports baseline (PIC10F, PIC12F5xx, PIC16F5xx), midrange (PIC12F6xx, PIC16F), PIC18F, PIC24, dsPIC30, dsPIC33, and PIC32 families of 8-bit, 16-bit, and 32-bit microcontrollers, and many Microchip Serial EEPROM products. With Microchip's powerful MPLAB Integrated Development Environment (IDE) the PICkit™ 2 enables in-circuit debugging on most PIC® microcontrollers. In-Circuit-Debugging runs, halts and single steps the program while the PIC microcontroller is embedded in the application. When halted at a breakpoint, the file registers can be examined and modified.

The PICkit 2 Debug Express include the PICkit 2, demo board and microcontroller, hookup cables and CDROM with user's guide, lessons, tutorial, compiler and MPLAB IDE software.

## 15.12 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.

## 15.13 Demonstration/Development Boards, Evaluation Kits, and Starter Kits

A wide variety of demonstration, development and evaluation boards for various PIC MCUs and dsPIC DSCs allows quick application development on fully functional systems. Most boards include prototyping areas for adding custom circuitry and provide application firmware and source code for examination and modification.

The boards support a variety of features, including LEDs, temperature sensors, switches, speakers, RS-232 interfaces, LCD displays, potentiometers and additional EEPROM memory.

The demonstration and development boards can be used in teaching environments, for prototyping custom circuits and for learning about various microcontroller applications.

In addition to the PICDEM™ and dsPICDEM™ demonstration/development board series of circuits, Microchip has a line of evaluation kits and demonstration software for analog filter design, KEELOQ® security ICs, CAN, IrDA®, PowerSmart battery management, SEEVAL® evaluation system, Sigma-Delta ADC, flow rate sensing, plus many more.

Also available are starter kits that contain everything needed to experience the specified device. This usually includes a single application and debug capability, all on one board.

Check the Microchip web page (www.microchip.com) for the complete list of demonstration, development and evaluation kits.

## 16.1 DC Characteristics: PIC12F609/615/617/12HV609/615-I (Industrial) PIC12F609/615/617/12HV609/615-E (Extended)

| DC CHARACTERISTICS | | | Standard Operating Conditions (unless otherwise stated)<br>Operating temperature -40°C ≤ TA ≤ +85°C for industrial<br>-40°C ≤ TA ≤ +125°C for extended | | | | |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Sym** | **Characteristic** | **Min** | **Typ†** | **Max** | **Units** | **Conditions** |
| | VDD | **Supply Voltage** | | | | | |
| D001 | | PIC12F609/615/617 | 2.0 | — | 5.5 | V | FOSC < = 4 MHz |
| D001 | | PIC12HV609/615 | 2.0 | — | —(2) | V | FOSC < = 4 MHz |
| D001B | | PIC12F609/615/617 | 2.0 | — | 5.5 | V | FOSC < = 8 MHz |
| D001B | | PIC12HV609/615 | 2.0 | — | —(2) | V | FOSC < = 8 MHz |
| D001C | | PIC12F609/615/617 | 3.0 | — | 5.5 | V | FOSC < = 10 MHz |
| D001C | | PIC12HV609/615 | 3.0 | — | —(2) | V | FOSC < = 10 MHz |
| D001D | | PIC12F609/615/617 | 4.5 | — | 5.5 | V | FOSC < = 20 MHz |
| D001D | | PIC12HV609/615 | 4.5 | — | —(2) | V | FOSC < = 20 MHz |
| D002* | VDR | **RAM Data Retention Voltage(1)** | 1.5 | — | — | V | Device in Sleep mode |
| D003 | VPOR | **VDD Start Voltage** to ensure internal Power-on Reset signal | — | VSS | — | V | See **Section 12.3.1 "Power-on Reset (POR)"** for details. |
| D004* | SVDD | **VDD Rise Rate** to ensure internal Power-on Reset signal | 0.05 | — | — | V/ms | See **Section 12.3.1 "Power-on Reset (POR)"** for details. |

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** This is the limit to which VDD can be lowered in Sleep mode without losing RAM data.

**2:** User defined. Voltage across the shunt regulator should not exceed 5V.

## 16.11 AC Characteristics: PIC12F609/615/617/12HV609/615 (Industrial, Extended)

**FIGURE 16-4: CLOCK TIMING**

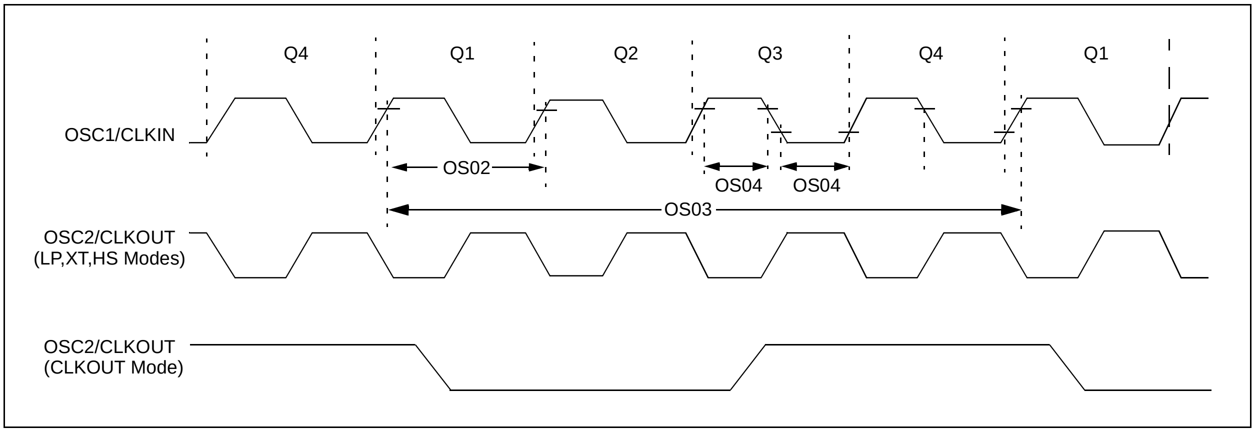



**TABLE 16-1: CLOCK OSCILLATOR TIMING REQUIREMENTS**

**Standard Operating Conditions (unless otherwise stated)**
Operating temperature -40°C ≤ TA ≤ +125°C

| Param No. | Sym | Characteristic | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|
| OS01 | FOSC | External CLKIN Frequency(1) | DC | — | 37 | kHz | LP Oscillator mode |
| | | | DC | — | 4 | MHz | XT Oscillator mode |
| | | | DC | — | 20 | MHz | HS Oscillator mode |
| | | | DC | — | 20 | MHz | EC Oscillator mode |
| | | Oscillator Frequency(1) | — | 32.768 | — | kHz | LP Oscillator mode |
| | | | 0.1 | — | 4 | MHz | XT Oscillator mode |
| | | | 1 | — | 20 | MHz | HS Oscillator mode |
| | | | DC | — | 4 | MHz | RC Oscillator mode |
| OS02 | TOSC | External CLKIN Period(1) | 27 | — | ∞ | μs | LP Oscillator mode |
| | | | 250 | — | ∞ | ns | XT Oscillator mode |
| | | | 50 | — | ∞ | ns | HS Oscillator mode |
| | | | 50 | — | ∞ | ns | EC Oscillator mode |
| | | Oscillator Period(1) | — | 30.5 | — | μs | LP Oscillator mode |
| | | | 250 | — | 10,000 | ns | XT Oscillator mode |
| | | | 50 | — | 1,000 | ns | HS Oscillator mode |
| | | | 250 | — | — | ns | RC Oscillator mode |
| OS03 | TCY | Instruction Cycle Time(1) | 200 | TCY | DC | ns | TCY = 4/FOSC |
| OS04* | TOSH, TOSL | External CLKIN High, External CLKIN Low | 2 | — | — | μs | LP oscillator |
| | | | 100 | — | — | ns | XT oscillator |
| | | | 20 | — | — | ns | HS oscillator |
| OS05* | TOSR, TOSF | External CLKIN Rise, External CLKIN Fall | 0 | — | ∞ | ns | LP oscillator |
| | | | 0 | — | ∞ | ns | XT oscillator |
| | | | 0 | — | ∞ | ns | HS oscillator |

* These parameters are characterized but not tested.
† Data in “Typ” column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** Instruction cycle period (TCY) equals four times the input oscillator time base period. All specified values are based on characterization data for that particular oscillator type under standard operating conditions with the device executing code. Exceeding these specified limits may result in an unstable oscillator operation and/or higher than expected current consumption. All devices are tested to operate at “min” values with an external clock applied to OSC1 pin. When an external clock input is used, the “max” cycle time limit is “DC” (no clock) for all devices.

**FIGURE 16-8: TIMER0 AND TIMER1 EXTERNAL CLOCK TIMINGS**

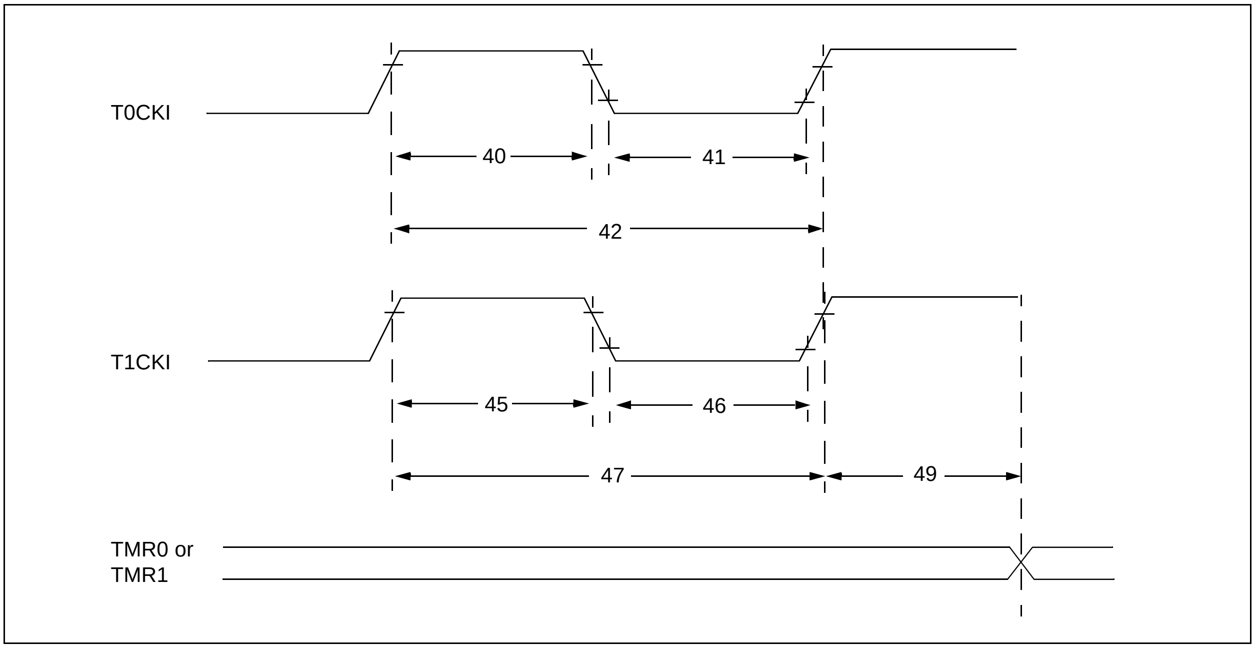



**TABLE 16-5: TIMER0 AND TIMER1 EXTERNAL CLOCK REQUIREMENTS**

**Standard Operating Conditions (unless otherwise stated)**
Operating Temperature -40°C ≤ $T_A$ ≤ +125°C

| Param No. | Sym | Characteristic | | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|---|
| 40* | $T_{T0H}$ | T0CKI High Pulse Width | No Prescaler | 0.5 $T_{CY}$ + 20 | — | — | ns | |
| | | | With Prescaler | 10 | — | — | ns | |
| 41* | $T_{T0L}$ | T0CKI Low Pulse Width | No Prescaler | 0.5 $T_{CY}$ + 20 | — | — | ns | |
| | | | With Prescaler | 10 | — | — | ns | |
| 42* | $T_{T0P}$ | T0CKI Period | | Greater of: 20 or $\frac{T_{CY} + 40}{N}$ | — | — | ns | N = prescale value (2, 4, ..., 256) |
| 45* | $T_{T1H}$ | T1CKI High Time | Synchronous, No Prescaler | 0.5 $T_{CY}$ + 20 | — | — | ns | |
| | | | Synchronous, with Prescaler | 15 | — | — | ns | |
| | | | Asynchronous | 30 | — | — | ns | |
| 46* | $T_{T1L}$ | T1CKI Low Time | Synchronous, No Prescaler | 0.5 $T_{CY}$ + 20 | — | — | ns | |
| | | | Synchronous, with Prescaler | 15 | — | — | ns | |
| | | | Asynchronous | 30 | — | — | ns | |
| 47* | $T_{T1P}$ | T1CKI Input Period | Synchronous | Greater of: 30 or $\frac{T_{CY} + 40}{N}$ | — | — | ns | N = prescale value (1, 2, 4, 8) |
| | | | Asynchronous | 60 | — | — | ns | |
| 48 | $F_{T1}$ | Timer1 Oscillator Input Frequency Range (oscillator enabled by setting bit T1OSCEN) | | — | 32.768 | — | kHz | |
| 49* | $T_{CKEZTMR1}$ | Delay from External Clock Edge to Timer Increment | | 2 $T_{OSC}$ | — | 7 $T_{OSC}$ | — | Timers in Sync mode |

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**TABLE 16-18: OSCILLATOR PARAMETERS FOR PIC12F615-H (High Temp.)**

| Param No. | Sym | Characteristic | Frequency Tolerance | Units | Min | Typ | Max | Conditions |
|---|---|---|---|---|---|---|---|---|
| OS08 | INTOSC | Int. Calibrated INTOSC Freq.(1) | ±10% | MHz | 7.2 | 8.0 | 8.8 | $2.0V \le V_{DD} \le 5.5V$<br>$-40°C \le T_A \le 150°C$ |

**Note 1:** To ensure these oscillator frequency tolerances, Vdd and Vss must be capacitively decoupled as close to the device as possible. 0.1 µF and 0.01 µF values in parallel are recommended.

**TABLE 16-19: COMPARATOR SPECIFICATIONS FOR PIC12F615-H (High Temp.)**

| Param No. | Sym | Characteristic | Units | Min | Typ | Max | Conditions |
|---|---|---|---|---|---|---|---|
| CM01 | $V_{OS}$ | Input Offset Voltage | mV | — | ±5 | ±20 | $(V_{DD} - 1.5)/2$ |

**FIGURE 17-50: SHUNT REGULATOR VOLTAGE vs. TEMP (TYPICAL)**

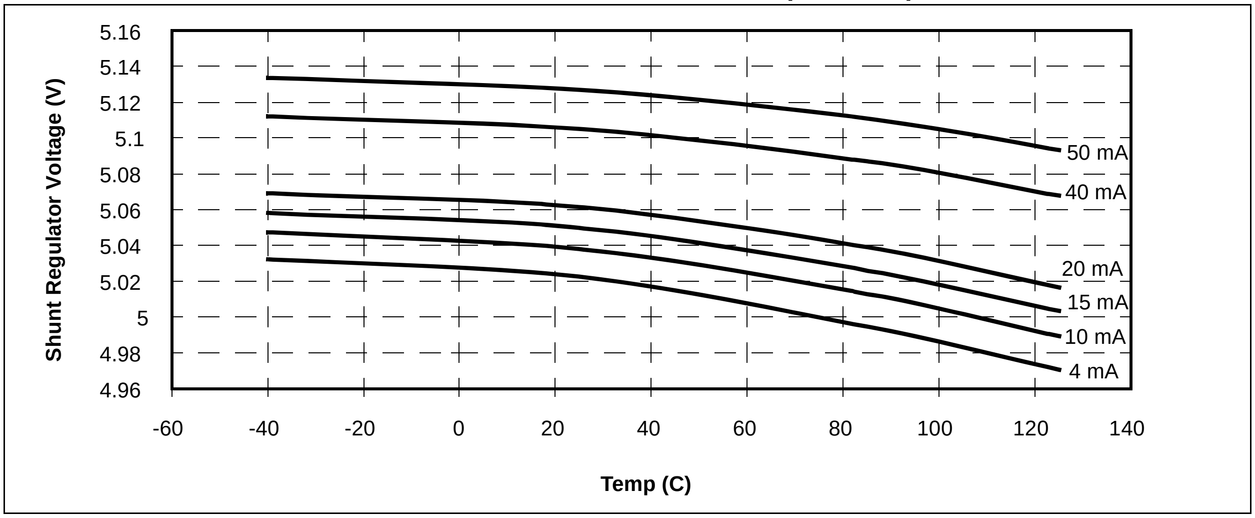
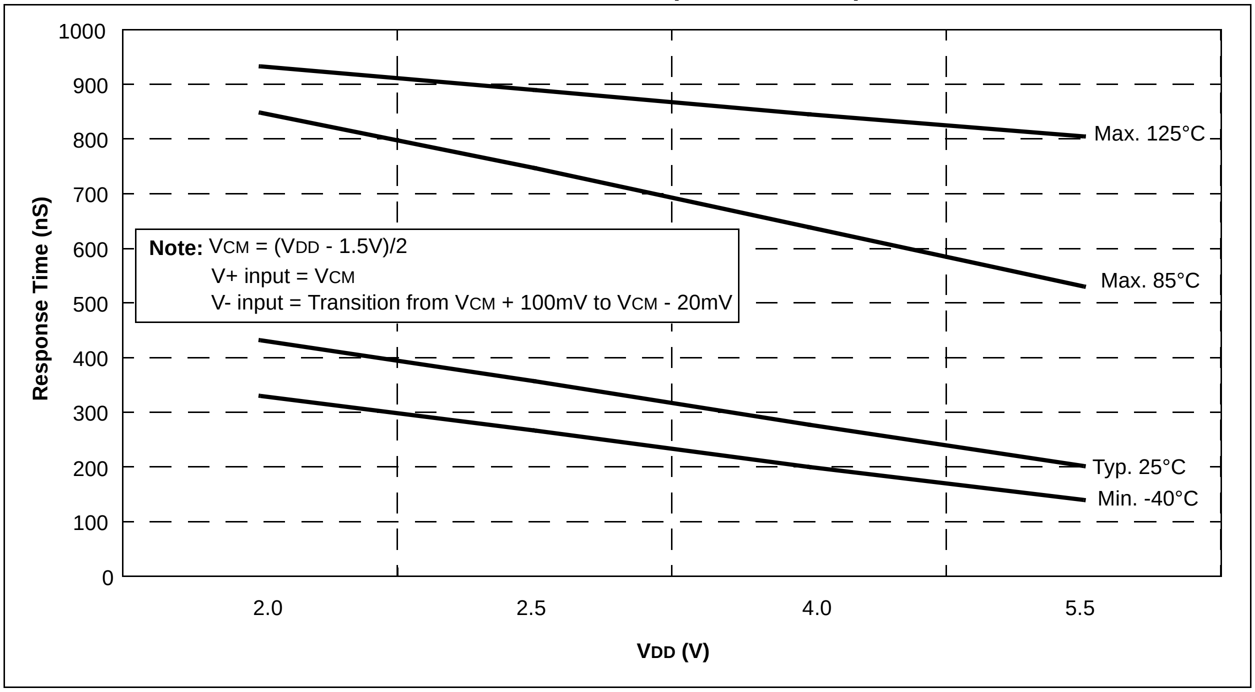
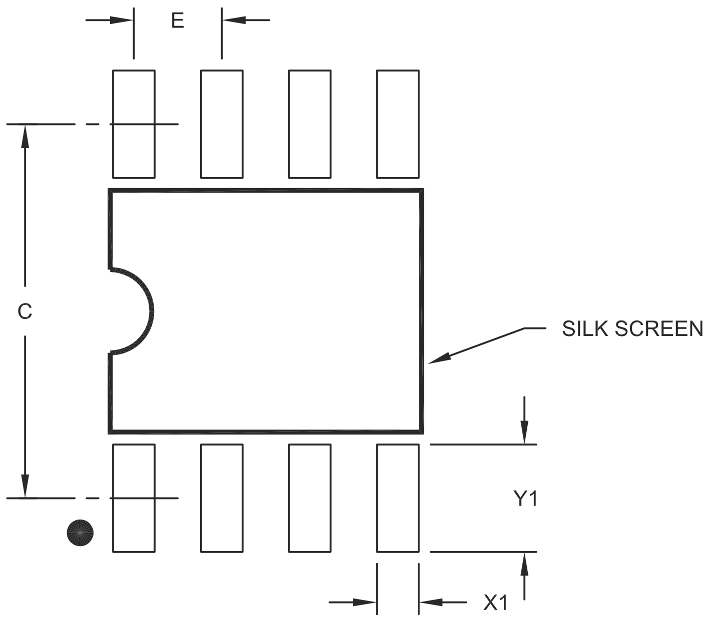



**FIGURE 17-51: COMPARATOR RESPONSE TIME (RISING EDGE)**



### 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 |
|---|---|




RECOMMENDED LAND PATTERN

| | | Units | MILLIMETERS | | |
|---|---|---|---|---|---|
| | | Dimension Limits | MIN | NOM | MAX |
| Contact Pitch | | E | 1.27 BSC | | |
| Contact Pad Spacing | | C | | 5.40 | |
| Contact Pad Width (X8) | | X1 | | | 0.60 |
| Contact Pad Length (X8) | | Y1 | | | 1.55 |

Notes:

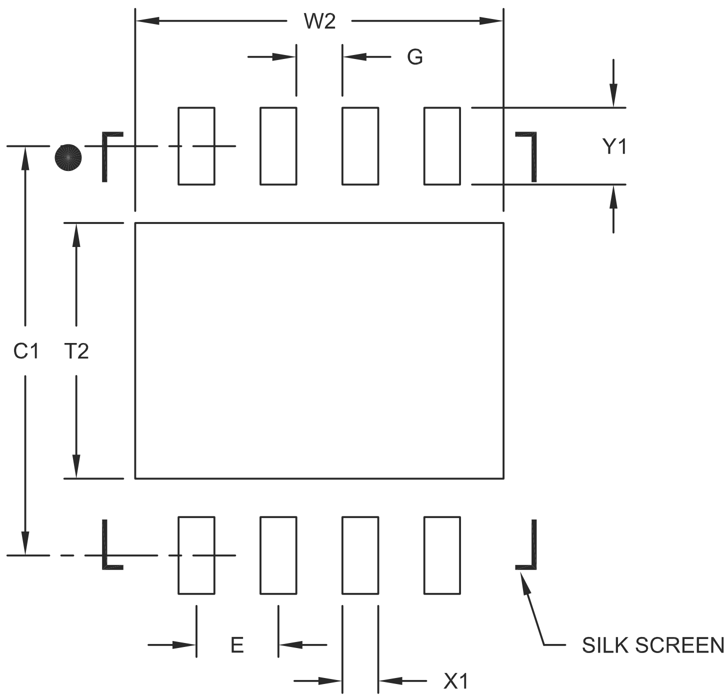
1. Dimensioning and tolerancing per ASME Y14.5M

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

Microchip Technology Drawing No. C04-2057A

## 8-Lead Plastic Dual Flat, No Lead Package (MD) – 4x4x0.9 mm 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 | | 0.80 BSC | |
| Optional Center Pad Width | W2 | | | 3.60 |
| Optional Center Pad Length | T2 | | | 2.50 |
| Contact Pad Spacing | C1 | | 4.00 | |
| Contact Pad Width (X8) | X1 | | | 0.35 |
| Contact Pad Length (X8) | Y1 | | | 0.75 |
| Distance Between Pads | G | 0.45 | | |

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

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

Microchip Technology Drawing No. C04-2131B

## PRODUCT IDENTIFICATION SYSTEM

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.

| PART NO. | X | /XX | XXX |
|---|---|---|---|
| Device | Temperature Range | Package | Pattern |

| | | | |
|---|---|---|---|
| **Device:** | PIC12F609, PIC12F609T[1], PIC12HV609, PIC12HV609T[1], PIC12F615, PIC12F615T[1], PIC12HV615, PIC12HV615T[1], PIC12F617, PIC12F617T[1] | | |
| **Temperature Range:** | H | = -40°C to +150°C | (High Temp)[3] |
| | I | = -40°C to +85°C | (Industrial) |
| | E | = -40°C to +125°C | (Extended) |
| **Package:** | P | = | Plastic DIP (PDIP) |
| | SN | = | 8-lead Small Outline (150 mil) (SOIC) |
| | MS | = | Micro Small Outline (MSOP) |
| | MF | = | 8-lead Plastic Dual Flat, No Lead (3x3) (DFN) |
| | MD | = | 8-lead Plastic Dual Flat, No Lead (4x4)(DFN)[1,2] |
| **Pattern:** | QTP, SQTP or ROM Code; Special Requirements (blank otherwise) | | |

**Examples:**

a) PIC12F615-E/P 301 = Extended Temp., PDIP package, 20 MHz, QTP pattern #301
b) PIC12F615-I/SN = Industrial Temp., SOIC package, 20 MHz
c) PIC12F615T-E/MF = Tape and Reel, Extended Temp., 3x3 DFN, 20 MHz
d) PIC12F609T-E/MF = Tape and Reel, Extended Temp., 3x3 DFN, 20 MHz
e) PIC12HV615T-E/MF = Tape and Reel, Extended Temp., 3x3 DFN, 20 MHz
f) PIC12HV609T-E/MF = Tape and Reel, Extended Temp., 3x3 DFN, 20 MHz
g) PIC12F617T-E/MF = Tape and Reel, Extended Temp., 3x3 DFN, 20 MHz
h) PIC12F617-I/P = Industrial Temp., PDIP package, 20 MHz
i) PIC12F615-H/SN = High Temp., SOIC package, 20 MHz

**Note 1:** T = in tape and reel for MSOP, SOIC and DFN packages only.
**2:** Not available for PIC12F617.
**3:** High Temp. available for PIC12F615 only.