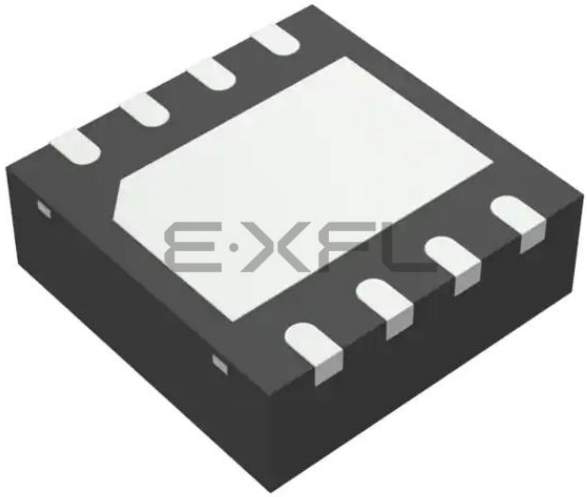


Welcome to E-XFL.COM

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 ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	8-VDFN Exposed Pad
Supplier Device Package	8-DFN (4x4)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic12hv615-i-md

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

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
ISO/TS 16949:2002

Microchip received ISO/TS-16949:2002 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.

TABLE 1-1: PIC12F609/HV609 PINOUT DESCRIPTION

Name	Function	Input Type	Output Type	Description
GP0/CIN+/ICSPDAT	GP0	TTL	CMOS	General purpose I/O with prog. pull-up and interrupt-on-change
	CIN+	AN	—	Comparator non-inverting input
	ICSPDAT	ST	CMOS	Serial Programming Data I/O
GP1/CIN0-/ICSPCLK	GP1	TTL	CMOS	General purpose I/O with prog. pull-up and interrupt-on-change
	CIN0-	AN	—	Comparator inverting input
	ICSPCLK	ST	—	Serial Programming Clock
GP2/T0CKI/INT/COUT	GP2	ST	CMOS	General purpose I/O with prog. pull-up and interrupt-on-change
	T0CKI	ST	—	Timer0 clock input
	INT	ST	—	External Interrupt
	COUT	—	CMOS	Comparator output
GP3/$\overline{MCLR}$/V_{PP}	GP3	TTL	—	General purpose input with interrupt-on-change
	$\overline{MCLR}$	ST	—	Master Clear w/internal pull-up
	V_{PP}	HV	—	Programming voltage
GP4/CIN1-/$\overline{T1G}$/OSC2/ CLKOUT	GP4	TTL	CMOS	General purpose I/O with prog. pull-up and interrupt-on-change
	CIN1-	AN	—	Comparator inverting input
	$\overline{T1G}$	ST	—	Timer1 gate (count enable)
	OSC2	—	XTAL	Crystal/Resonator
	CLKOUT	—	CMOS	F_{OSC}/4 output
GP5/T1CKI/OSC1/CLKIN	GP5	TTL	CMOS	General purpose I/O with prog. pull-up and interrupt-on-change
	T1CKI	ST	—	Timer1 clock input
	OSC1	XTAL	—	Crystal/Resonator
	CLKIN	ST	—	External clock input/RC oscillator connection
V_{DD}	V_{DD}	Power	—	Positive supply
V_{SS}	V_{SS}	Power	—	Ground reference

Legend: AN=Analog input or output CMOS = CMOS compatible input or output HV= High Voltage
ST=Schmitt Trigger input with CMOS levels TTL = TTL compatible input XTAL=Crystal

TABLE 2-4: PIC12F615/617/HV615 SPECIAL FUNCTION REGISTERS SUMMARY BANK 1

Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Page
Bank 1											
80h	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								xxxx xxxx	25, 116
81h	OPTION_REG	$\overline{GPPU}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	19, 116
82h	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	25, 116
83h	STATUS	IRP(1)	RP1(1)	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	18, 116
84h	FSR	Indirect Data Memory Address Pointer								xxxx xxxx	25, 116
85h	TRISIO	—	—	TRISIO5	TRISIO4	TRISIO3(4)	TRISIO2	TRISIO1	TRISIO0	--11 1111	44, 116
86h	—	Unimplemented								—	—
87h	—	Unimplemented								—	—
88h	—	Unimplemented								—	—
89h	—	Unimplemented								—	—
8Ah	PCLATH	—	—	—	Write Buffer for upper 5 bits of Program Counter					---0 0000	25, 116
8Bh	INTCON	GIE	PEIE	T0IE	INTE	GPIE	T0IF	INTF	GPIF(3)	0000 0000	20, 116
8Ch	PIE1	—	ADIE	CCP1IE	—	CMIE	—	TMR2IE	TMR1IE	-00- 0-00	21, 116
8Dh	—	Unimplemented								—	—
8Eh	PCON	—	—	—	—	—	—	$\overline{POR}$	$\overline{BOR}$	---- --qq	23, 116
8Fh	—	Unimplemented								—	—
90h	OSCTUNE	—	—	—	TUN4	TUN3	TUN2	TUN1	TUN0	---0 0000	41, 116
91h	—	Unimplemented								—	—
92h	PR2	Timer2 Module Period Register								1111 1111	65, 116
93h	APFCON	—	—	—	T1GSEL	—	—	P1BSEL	P1ASEL	---0 --00	21, 116
94h	—	Unimplemented								—	—
95h	WPU(2)	—	—	WPU5	WPU4	—	WPU2	WPU1	WPU0	--11 -111	46, 116
96h	IOC	—	—	IOC5	IOC4	IOC3	IOC2	IOC1	IOC0	--00 0000	46, 116
97h	—	Unimplemented								—	—
98h	PMCON1(7)	—	—	—	—	—	WREN	WR	RD	---- -000	29
99h	PMCON2(7)	Program Memory Control Register 2 (not a physical register).								---- ----	—
9Ah	PMADRL(7)	PMADRL7	PMADRL6	PMADRL5	PMADRL4	PMADRL3	PMADRL2	PMADRL1	PMADRL0	0000 0000	28
9Bh	PMADRH(7)	—	—	—	—	—	PMADRH2	PMADRH1	PMADRH0	---- -000	28
9Ch	PMDATL(7)	PMDATL7	PMDATL6	PMDATL5	PMDATL4	PMDATL3	PMDATL2	PMDATL1	PMDATL0	0000 0000	28
9Dh	PMDATH(7)	—	—	Program Memory Data Register High Byte.						--00 0000	28
9Eh	ADRESL(5, 6)	Least Significant 2 bits of the left shifted result or 8 bits of the right shifted result								xxxx xxxx	85, 117
9Fh	ANSEL	—	ADCS2	ADCS1	ADCS0	ANS3	ANS2	ANS1	ANS0	-000 1111	45, 117

Legend: – = Unimplemented locations read as '0', u = unchanged, x = unknown, q = value depends on condition, shaded = unimplemented

Note 1: IRP and RP1 bits are reserved, always maintain these bits clear.

2: GP3 pull-up is enabled when MCLRE is '1' in the Configuration Word register.

3: $\overline{MCLR}$ and WDT Reset does not affect the previous value data latch. The GPIF bit will clear upon Reset but will set again if the mismatch exists.

4: TRISIO3 always reads as '1' since it is an input only pin.

5: Read only register.

6: PIC12F615/617/HV615 only.

7: PIC12F617 only.

2.2.2.4 PIE1 Register

The PIE1 register contains the Peripheral Interrupt Enable bits, as shown in Register 2-4.

Note: Bit PEIE of the INTCON register must be set to enable any peripheral interrupt.

REGISTER 2-4: PIE1: PERIPHERAL INTERRUPT ENABLE REGISTER 1

U-0	R/W-0	R/W-0	U-0	R/W-0	U-0	R/W-0	R/W-0
—	ADIE(1)	CCP1IE(1)	—	CMIE	—	TMR2IE(1)	TMR1IE
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

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

bit 6 **ADIE:** A/D Converter (ADC) Interrupt Enable bit(1)
1 = Enables the ADC interrupt
0 = Disables the ADC interrupt

bit 5 **CCP1IE:** CCP1 Interrupt Enable bit(1)
1 = Enables the CCP1 interrupt
0 = Disables the CCP1 interrupt

bit 4 **Unimplemented:** Read as '0'

bit 3 **CMIE:** Comparator Interrupt Enable bit
1 = Enables the Comparator interrupt
0 = Disables the Comparator interrupt

bit 2 **Unimplemented:** Read as '0'

bit 1 **TMR2IE:** Timer2 to PR2 Match Interrupt Enable bit(1)
1 = Enables the Timer2 to PR2 match interrupt
0 = Disables the Timer2 to PR2 match interrupt

bit 0 **TMR1IE:** Timer1 Overflow Interrupt Enable bit
1 = Enables the Timer1 overflow interrupt
0 = Disables the Timer1 overflow interrupt

Note 1: PIC12F615/617/HV615 only. PIC12F609/HV609 unimplemented, read as '0'.

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
```

3.4 Writing the Flash Program Memory

A word of the Flash program memory may only be written to if the word is in an unprotected segment of memory.

Flash program memory must be written in four-word blocks. See Figure 3-2 and Figure 3-3 for more details. A block consists of four words with sequential addresses, with a lower boundary defined by an address, where PMADRL<1:0> = 00. All block writes to program memory are done as 16-word erase by four-word write operations. The write operation is edge-aligned and cannot occur across boundaries.

To write program data, it must first be loaded into the buffer registers (see Figure 3-2). This is accomplished by first writing the destination address to PMADRL and PMADRH and then writing the data to PMDATL and PMDATH. After the address and data have been set up, then the following sequence of events must be executed:

1. Write 55h, then AAh, to PMCON2 (Flash programming sequence).
2. Set the WR control bit of the PMCON1 register.

All four buffer register locations should be written to with correct data. If less than four words are being written to in the block of four words, then a read from the program memory location(s) not being written to must be performed. This takes the data from the program location(s) not being written and loads it into the PMDATL and PMDATH registers. Then the sequence of events to transfer data to the buffer registers must be executed.

To transfer data from the buffer registers to the program memory, the PMADRL and PMADRH must point to the last location in the four-word block (PMADRL<1:0> = 11). Then the following sequence of events must be executed:

1. Write 55h, then AAh, to PMCON2 (Flash programming sequence).
2. Set control bit WR of the PMCON1 register to begin the write operation.

The user must follow the same specific sequence to initiate the write for each word in the program block, writing each program word in sequence (000, 001, 010, 011). When the write is performed on the last word (PMADRL<1:0> = 11), a block of sixteen words is automatically erased and the content of the four-word buffer registers are written into the program memory.

After the "BSF PMCON1,WR" instruction, the processor requires two cycles to set up the erase/write operation. The user must place two NOP instructions after the WR bit is set. Since data is being written to buffer registers, the writing of the first three words of the block appears to occur immediately. The processor will halt internal operations for the typical 4 ms, only during the cycle in which the erase takes place (i.e., the last word of the sixteen-word block erase). This is not Sleep mode as the clocks and peripherals will continue to run. After the four-word write cycle, the processor will resume operation with the third instruction after the PMCON1 write instruction. The above sequence must be repeated for the higher 12 words.

3.5 Protection Against Spurious Write

There are conditions when the device should not write to the program memory. To protect against spurious writes, various mechanisms have been built in. On power-up, WREN is cleared. Also, the Power-up Timer (64 ms duration) prevents program memory writes.

The write initiate sequence and the WREN bit help prevent an accidental write during brown-out, power glitch or software malfunction.

3.6 Operation During Code-Protect

When the device is code-protected, the CPU is able to read and write unscrambled data to the program memory. The test mode access is disabled.

3.7 Operation During Write Protect

When the program memory is write-protected, the CPU can read and execute from the program memory. The portions of program memory that are write protected can be modified by the CPU using the PMCON registers, but the protected program memory cannot be modified using ICSP mode.

An example of the complete four-word write sequence is shown in Example 3-2. The initial address is loaded into the PMADRH and PMADRL register pair; the eight words of data are loaded using indirect addressing.

EXAMPLE 3-2: WRITING TO FLASH PROGRAM MEMORY

```
        ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
        ; This write routine assumes the following:
        ;     A valid starting address (the least significant bits = '00')
        ;     is loaded in ADDRH:ADDRL
        ;     ADDRH, ADDRL and DATADDR are all located in data memory
        ;
        BANKSEL  PMADRH
        MOVF     ADDRH,W      ; Load initial address
        MOVWF    PMADRH       ;
        MOVF     ADDRL,W      ;
        MOVWF    PMADRL       ;
        MOVF     DATAADDR,W   ; Load initial data address
        MOVWF    FSR          ;
LOOP    MOVF     INDF,W       ; Load first data byte into lower
        MOVWF    PMDATL       ;
        INCF     FSR,F        ; Next byte
        MOVF     INDF,W       ; Load second data byte into upper
        MOVWF    PMDATH       ;
        INCF     FSR,F        ;
        BANKSEL  PMCON1
        BSF      PMCON1,WREN  ; Enable writes
        BCF      INTCON,GIE   ; Disable interrupts (if using)
        BTFSC    INTCON,GIE   ; See AN576
        GOTO     $-2
        ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
        ;                          Required Sequence
        MOVLW    55h          ; Start of required write sequence:
        MOVWF    PMCON2       ; Write 55h
        MOVLW    0AAh         ;
        MOVWF    PMCON2       ; Write 0AAh
        BSF      PMCON1,WR    ; Set WR bit to begin write
        NOP                   ; Required to transfer data to the buffer
        NOP                   ; registers
        ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
        BCF      PMCON1,WREN  ; Disable writes
        BSF      INTCON,GIE   ; Enable interrupts (comment out if not using interrupts)
        BANKSEL  PMADRL
        MOVF     PMADRL, W
        INCF     PMADRL,F     ; Increment address
        ANDLW    0x03         ; Indicates when sixteen words have been programmed
        SUBLW    0x03         ;   0x0F = 16 words
                              ;   0x0B = 12 words
                              ;   0x07 = 8 words
                              ;   0x03 = 4 words
        BTFSS    STATUS,Z     ; Exit on a match,
        GOTO     LOOP         ; Continue if more data needs to be written
```

REGISTER 5-5: WPU: WEAK PULL-UP GPIO REGISTER

U-0	U-0	R/W-1	R/W-1	U-0	R/W-1	R/W-1	R/W-1
—	—	WPU5	WPU4	—	WPU2	WPU1	WPU0
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 7-6 **Unimplemented**: Read as '0'

bit 5-4 **WPU<5:4>:** Weak Pull-up Control bits
1 = Pull-up enabled
0 = Pull-up disabled

bit 3 **WPU<3>:** Weak Pull-up Register bit(3)

bit 2-0 **WPU<2:0>:** Weak Pull-up Control bits
1 = Pull-up enabled
0 = Pull-up disabled

Note 1: Global $\overline{GPPU}$ must be enabled for individual pull-ups to be enabled.
2: The weak pull-up device is automatically disabled if the pin is in Output mode (TRISIO = 0).
3: The GP3 pull-up is enabled when configured as $\overline{MCLR}$ in the Configuration Word, otherwise it is disabled as an input and reads as '0'.
4: WPU<5:4> always reads '1' in XT, HS and LP Oscillator modes.

REGISTER 5-6: IOC: INTERRUPT-ON-CHANGE GPIO REGISTER

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	IOC5	IOC4	IOC3	IOC2	IOC1	IOC0
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 7-6 **Unimplemented**: Read as '0'

bit 5-0 **IOC<5:0>:** Interrupt-on-change GPIO Control bit
1 = Interrupt-on-change enabled
0 = Interrupt-on-change disabled

Note 1: Global Interrupt Enable (GIE) must be enabled for individual interrupts to be recognized.
2: IOC<5:4> always reads '1' in XT, HS and LP Oscillator modes.

5.2.4 PIN DESCRIPTIONS AND DIAGRAMS

Each GPIO pin is multiplexed with other functions. The pins and their combined functions are briefly described here. For specific information about individual functions such as the Comparator or the ADC, refer to the appropriate section in this data sheet.

5.2.4.1 GP0/AN0(1)/CIN+/P1B(1)/ICSPDAT

Figure 5-1 shows the diagram for this pin. The GP0 pin is configurable to function as one of the following:

- a general purpose I/O
- an analog input for the ADC(1)
- an analog non-inverting input to the comparator
- a PWM output(1)
- In-Circuit Serial Programming data

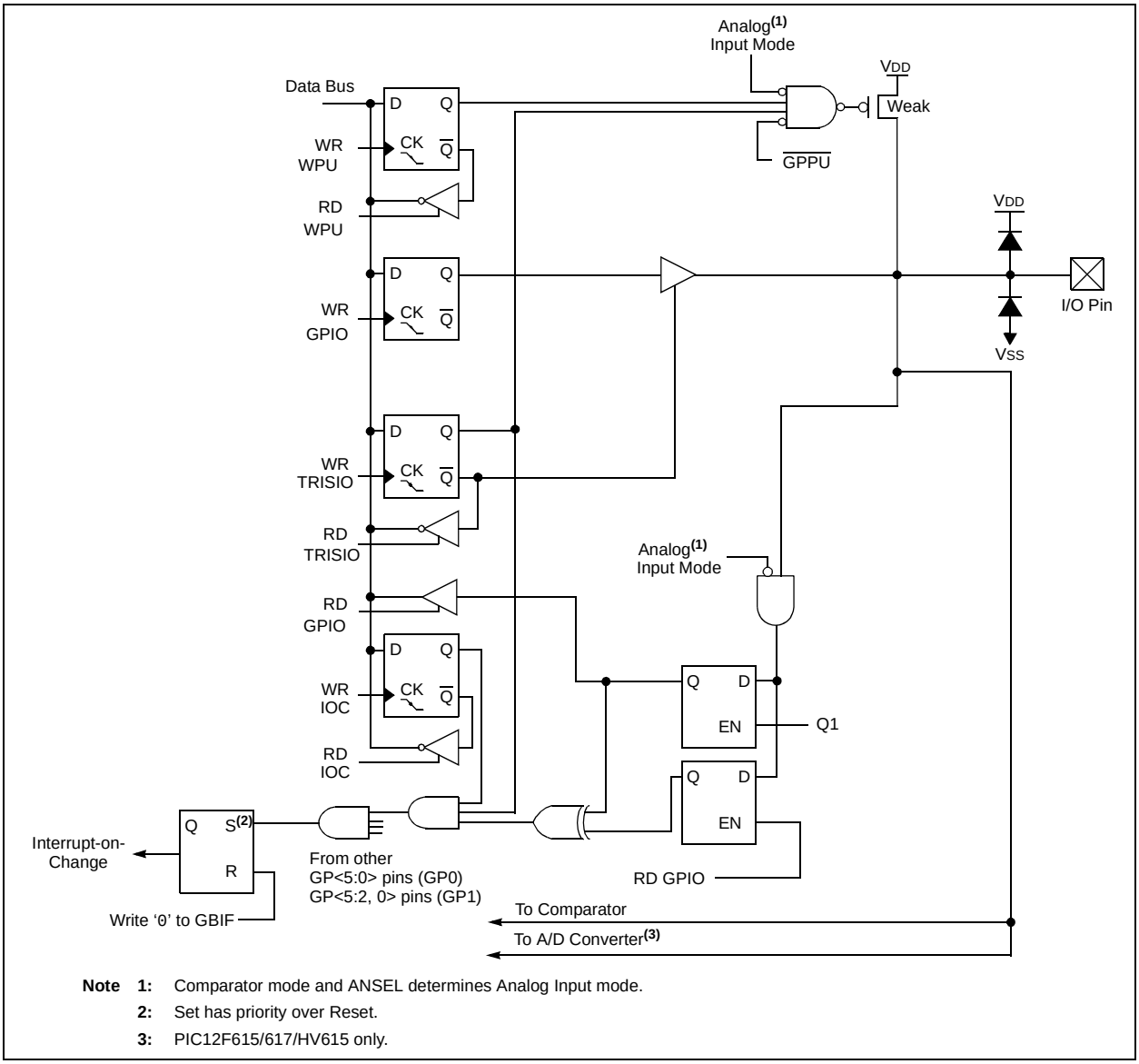
5.2.4.2 GP1/AN1(1)/CIN0-/VREF(1)/ICSPCLK

Figure 5-1 shows the diagram for this pin. The GP1 pin is configurable to function as one of the following:

- a general purpose I/O
- an analog input for the ADC(1)
- an analog inverting input to the comparator
- a voltage reference input for the ADC(1)
- In-Circuit Serial Programming clock

Note 1: PIC12F615/617/HV615 only.

FIGURE 5-1: BLOCK DIAGRAM OF GP<1:0>



Note 1: Comparator mode and ANSEL determines Analog Input mode.
2: Set has priority over Reset.
3: PIC12F615/617/HV615 only.

7.0 TIMER1 MODULE WITH GATE CONTROL

The Timer1 module is a 16-bit timer/counter with the following features:

- 16-bit timer/counter register pair (TMR1H:TMR1L)
- Programmable internal or external clock source
- 3-bit prescaler
- Optional LP oscillator
- Synchronous or asynchronous operation
- Timer1 gate (count enable) via comparator or $\overline{T1G}$ pin
- Interrupt on overflow
- Wake-up on overflow (external clock, Asynchronous mode only)
- Time base for the Capture/Compare function
- Special Event Trigger (with ECCP)
- Comparator output synchronization to Timer1 clock

Figure 7-1 is a block diagram of the Timer1 module.

7.1 Timer1 Operation

The Timer1 module is a 16-bit incrementing counter which is accessed through the TMR1H:TMR1L register pair. Writes to TMR1H or TMR1L directly update the counter.

When used with an internal clock source, the module is a timer. When used with an external clock source, the module can be used as either a timer or counter.

7.2 Clock Source Selection

The TMR1CS bit of the T1CON register is used to select the clock source. When TMR1CS = 0, the clock source is FOSC/4. When TMR1CS = 1, the clock source is supplied externally.

Clock Source	TMR1CS	T1ACS
FOSC/4	0	0
FOSC	0	1
T1CKI pin	1	x

REGISTER 9-3: VRCON: VOLTAGE REFERENCE CONTROL REGISTER

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CMVREN	—	VRR	FVREN	VR3	VR2	VR1	VR0
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 **CMVREN:** Comparator Voltage Reference Enable bit(1, 2)
1 = CVREF circuit powered on and routed to CVREF input of the Comparator
0 = 0.6 Volt constant reference routed to CVREF input of the Comparator

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

bit 5 **VRR:** CVREF Range Selection bit
1 = Low range
0 = High range

bit 4 **FVREN:** 0.6V Reference Enable bit(2)
1 = Enabled
0 = Disabled

bit 3-0 **VR<3:0>:** Comparator Voltage Reference CVREF Value Selection bits ($0 \leq$ VR<3:0> ≤ 15)
When VRR = 1: CVREF = (VR<3:0>/24) * VDD
When VRR = 0: CVREF = VDD/4 + (VR<3:0>/32) * VDD

Note 1: When CMVREN is low, the CVREF circuit is powered down and does not contribute to IDD current.
2: When CMVREN is low and the FVREN bit is low, the CVREF signal should provide Vss to the comparator.

10.1 ADC Configuration

When configuring and using the ADC the following functions must be considered:

- Port configuration
- Channel selection
- ADC voltage reference selection
- ADC conversion clock source
- Interrupt control
- Results formatting

10.1.1 PORT CONFIGURATION

The ADC can be used to convert both analog and digital signals. When converting analog signals, the I/O pin should be configured for analog by setting the associated TRIS and ANSEL bits. See the corresponding port section for more information.

Note:	Analog voltages on any pin that is defined as a digital input may cause the input buffer to conduct excess current.

10.1.2 CHANNEL SELECTION

The CHS bits of the ADCON0 register determine which channel is connected to the sample and hold circuit.

When changing channels, a delay is required before starting the next conversion. Refer to **Section 10.2 "ADC Operation"** for more information.

10.1.3 ADC VOLTAGE REFERENCE

The VCFG bit of the ADCON0 register provides control of the positive voltage reference. The positive voltage reference can be either VDD or an external voltage source. The negative voltage reference is always connected to the ground reference.

10.1.4 CONVERSION CLOCK

The source of the conversion clock is software selectable via the ADCS bits of the ANSEL register. There are seven possible clock options:

- FOSC/2
- FOSC/4
- FOSC/8
- FOSC/16
- FOSC/32
- FOSC/64
- FRC (dedicated internal oscillator)

The time to complete one bit conversion is defined as TAD. One full 10-bit conversion requires 11 TAD periods as shown in Figure 10-3.

For correct conversion, the appropriate TAD specification must be met. See A/D conversion requirements in **Section 16.0 "Electrical Specifications"** for more information. Table 10-1 gives examples of appropriate ADC clock selections.

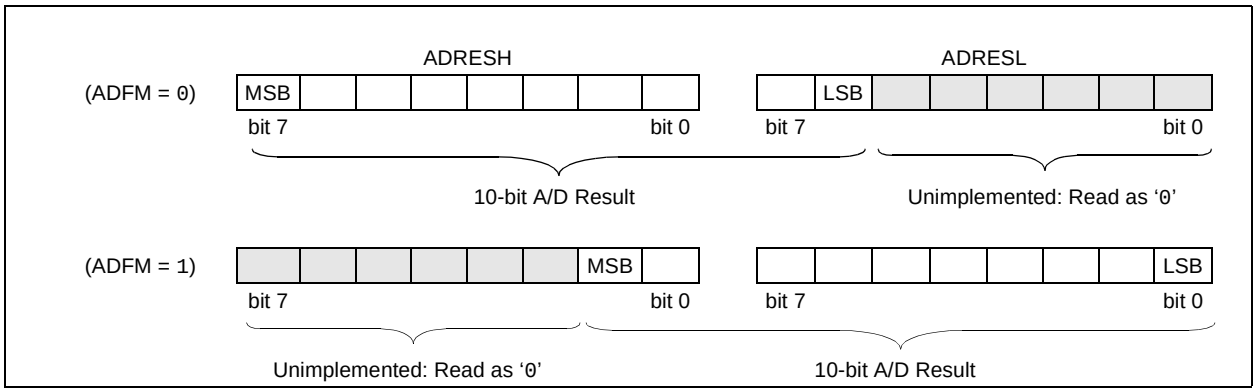
Note:	Unless using the FRC, any changes in the system clock frequency will change the ADC clock frequency, which may adversely affect the ADC result.

10.1.6 RESULT FORMATTING

The 10-bit A/D conversion result can be supplied in two formats, left justified or right justified. The ADFM bit of the ADCON0 register controls the output format.

Figure 10-4 shows the two output formats.

FIGURE 10-3: 10-BIT A/D CONVERSION RESULT FORMAT



10.2 ADC Operation

10.2.1 STARTING A CONVERSION

To enable the ADC module, the ADON bit of the ADCON0 register must be set to a '1'. Setting the GO/$\overline{\text{DONE}}$ bit of the ADCON0 register to a '1' will start the Analog-to-Digital conversion.

Note:	The GO/$\overline{\text{DONE}}$ bit should not be set in the same instruction that turns on the ADC. Refer to **Section 10.2.6 "A/D Conversion Procedure"**.

10.2.2 COMPLETION OF A CONVERSION

When the conversion is complete, the ADC module will:

- Clear the GO/$\overline{\text{DONE}}$ bit
- Set the ADIF flag bit
- Update the ADRESH:ADRESL registers with new conversion result

10.2.3 TERMINATING A CONVERSION

If a conversion must be terminated before completion, the GO/$\overline{\text{DONE}}$ bit can be cleared in software. The ADRESH:ADRESL registers will not be updated with the partially complete Analog-to-Digital conversion sample. Instead, the ADRESH:ADRESL register pair will retain the value of the previous conversion. Additionally, a 2 TAD delay is required before another acquisition can be initiated. Following this delay, an input acquisition is automatically started on the selected channel.

Note:	A device Reset forces all registers to their Reset state. Thus, the ADC module is turned off and any pending conversion is terminated.

10.2.4 ADC OPERATION DURING SLEEP

The ADC module can operate during Sleep. This requires the ADC clock source to be set to the FRC option. When the FRC clock source is selected, the ADC waits one additional instruction before starting the conversion. This allows the SLEEP instruction to be executed, which can reduce system noise during the conversion. If the ADC interrupt is enabled, the device will wake-up from Sleep when the conversion completes. If the ADC interrupt is disabled, the ADC module is turned off after the conversion completes, although the ADON bit remains set.

When the ADC clock source is something other than FRC, a SLEEP instruction causes the present conversion to be aborted and the ADC module is turned off, although the ADON bit remains set.

10.2.5 SPECIAL EVENT TRIGGER

The ECCP Special Event Trigger allows periodic ADC measurements without software intervention. When this trigger occurs, the GO/$\overline{\text{DONE}}$ bit is set by hardware and the Timer1 counter resets to zero.

Using the Special Event Trigger does not assure proper ADC timing. It is the user's responsibility to ensure that the ADC timing requirements are met.

See **Section 11.0 "Enhanced Capture/Compare/PWM (With Auto-Shutdown and Dead Band) Module (PIC12F615/617/HV615 only)"** for more information.

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.

TABLE 11-3: SUMMARY OF REGISTERS ASSOCIATED WITH COMPARE

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
TMR2	Timer2 Module Register								0000 0000	0000 0000
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 Compare.

Note 1: For PIC12F615/617/HV615 only.

11.3 PWM Mode

The PWM mode generates a Pulse-Width Modulated signal on the CCP1 pin. The duty cycle, period and resolution are determined by the following registers:

- PR2
- T2CON
- CCPR1L
- CCP1CON

In Pulse-Width Modulation (PWM) mode, the CCP module produces up to a 10-bit resolution PWM output on the CCP1 pin. Since the CCP1 pin is multiplexed with the PORT data latch, the TRIS for that pin must be cleared to enable the CCP1 pin output driver.

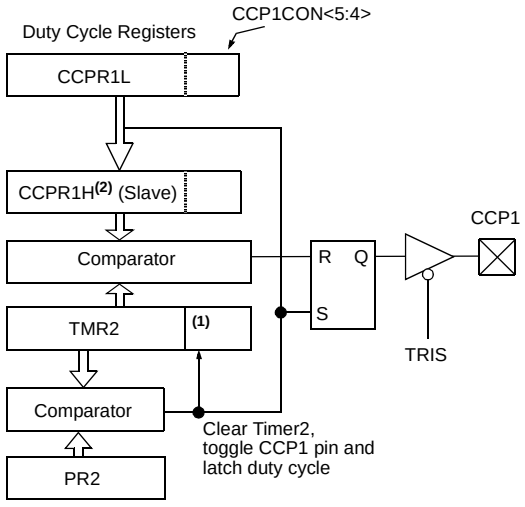
Note:	Clearing the CCP1CON register will relinquish CCP1 control of the CCP1 pin.

Figure 11-3 shows a simplified block diagram of PWM operation.

Figure 11-4 shows a typical waveform of the PWM signal.

For a step-by-step procedure on how to set up the CCP module for PWM operation, see **Section 11.3.7 "Setup for PWM Operation"**.

FIGURE 11-3: SIMPLIFIED PWM BLOCK DIAGRAM

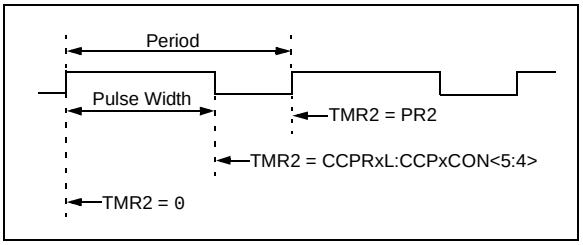


Note 1: The 8-bit timer TMR2 register is concatenated with the 2-bit internal system clock (FOSC), or 2 bits of the prescaler, to create the 10-bit time base.

2: In PWM mode, CCPR1H is a read-only register.

The PWM output (Figure 11-4) has a time base (period) and a time that the output stays high (duty cycle).

FIGURE 11-4: CCP PWM OUTPUT



12.3.4 BROWN-OUT RESET (BOR)

The BOREN0 and BOREN1 bits in the Configuration Word register select one of three BOR modes. One mode has been added to allow control of the BOR enable for lower current during Sleep. By selecting BOREN<1:0> = 10, the BOR is automatically disabled in Sleep to conserve power and enabled on wake-up. See Register 12-1 for the Configuration Word definition.

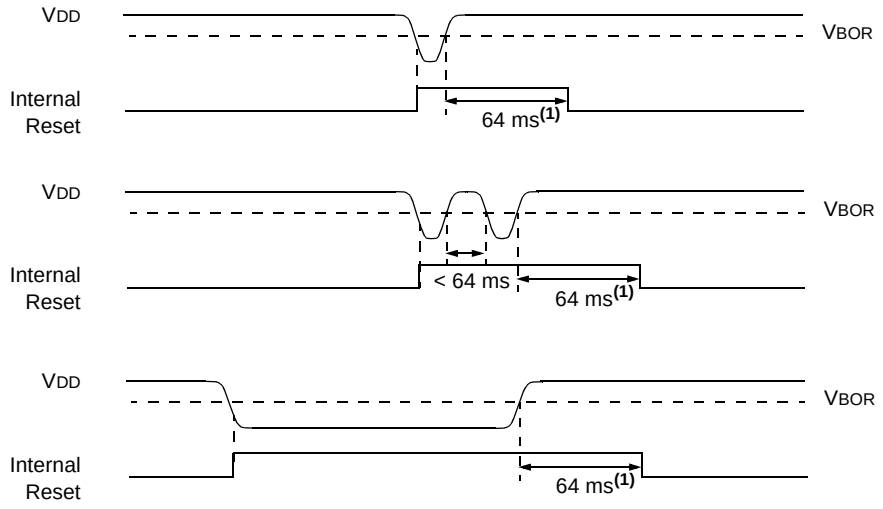
A brown-out occurs when VDD falls below VBOR for greater than parameter TBOR (see **Section 16.0 "Electrical Specifications"**). The brown-out condition will reset the device. This will occur regardless of VDD slew rate. A Brown-out Reset may not occur if VDD falls below VBOR for less than parameter TBOR.

On any Reset (Power-on, Brown-out Reset, Watchdog timer, etc.), the chip will remain in Reset until VDD rises above VBOR (see Figure 12-3). If enabled, the Power-up Timer will be invoked by the Reset and keep the chip in Reset an additional 64 ms.

Note:	The Power-up Timer is enabled by the $\overline{\text{PWRTE}}$ bit in the Configuration Word register.

If VDD drops below VBOR while the Power-up Timer is running, the chip will go back into a Brown-out Reset and the Power-up Timer will be re-initialized. Once VDD rises above VBOR, the Power-up Timer will execute a 64 ms Reset.

FIGURE 12-3: BROWN-OUT SITUATIONS



Note 1: 64 ms delay only if $\overline{\text{PWRTE}}$ bit is programmed to '0'.

FIGURE 12-4: TIME-OUT SEQUENCE ON POWER-UP (DELAYED $\overline{\text{MCLR}}$): CASE 1

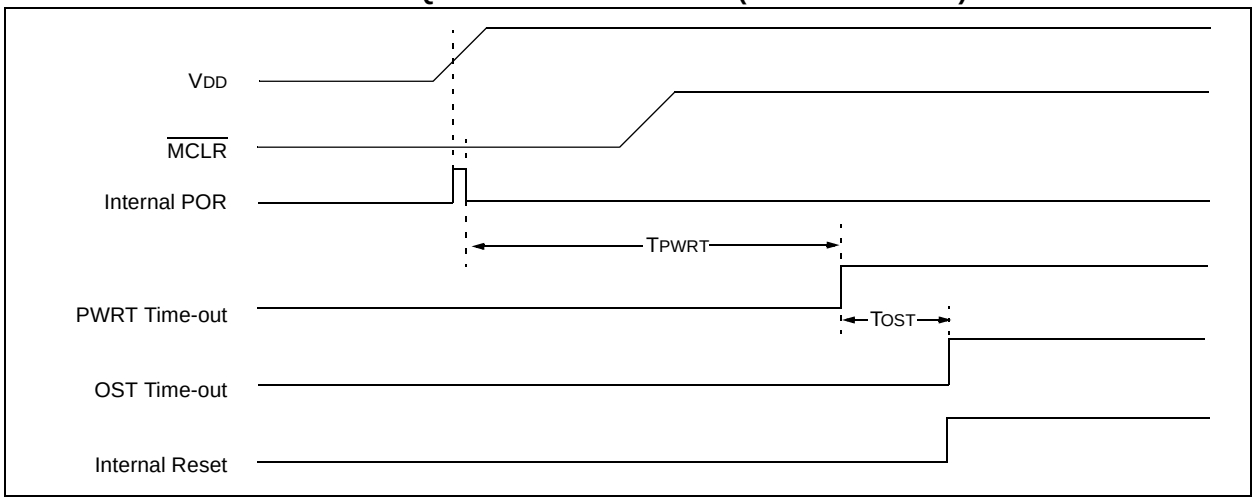


FIGURE 12-5: TIME-OUT SEQUENCE ON POWER-UP (DELAYED $\overline{\text{MCLR}}$): CASE 2

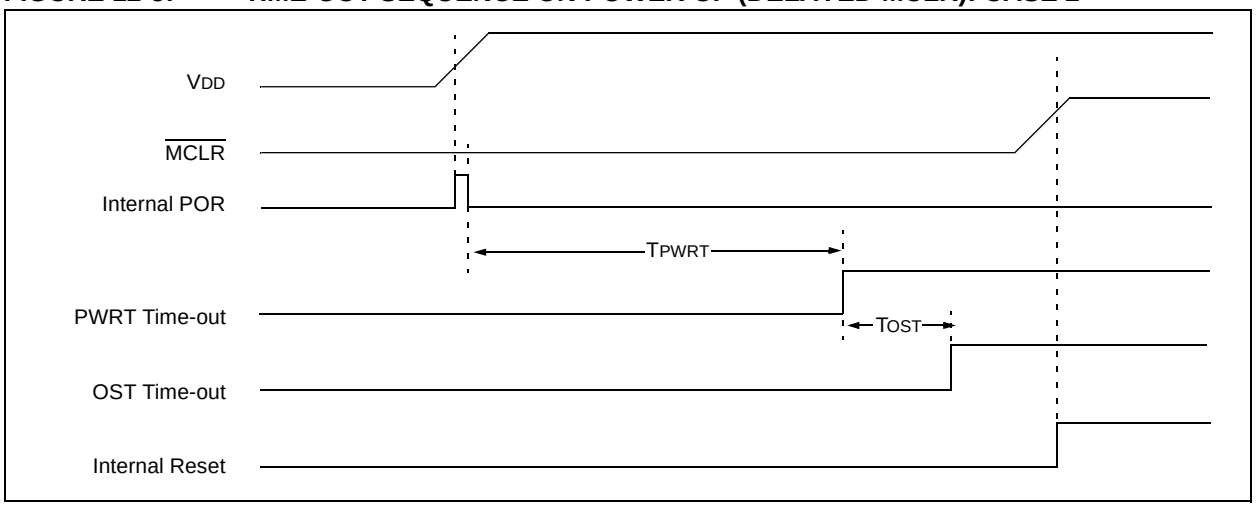
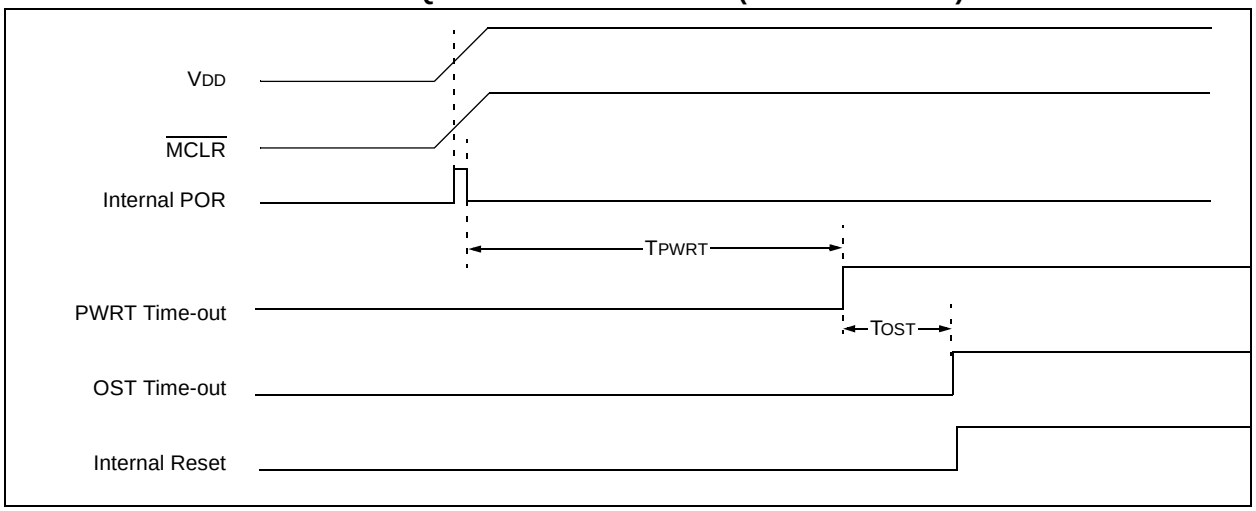


FIGURE 12-6: TIME-OUT SEQUENCE ON POWER-UP ($\overline{\text{MCLR}}$ WITH VDD)



NOTES:

FIGURE 17-52: COMPARATOR RESPONSE TIME (FALLING EDGE)

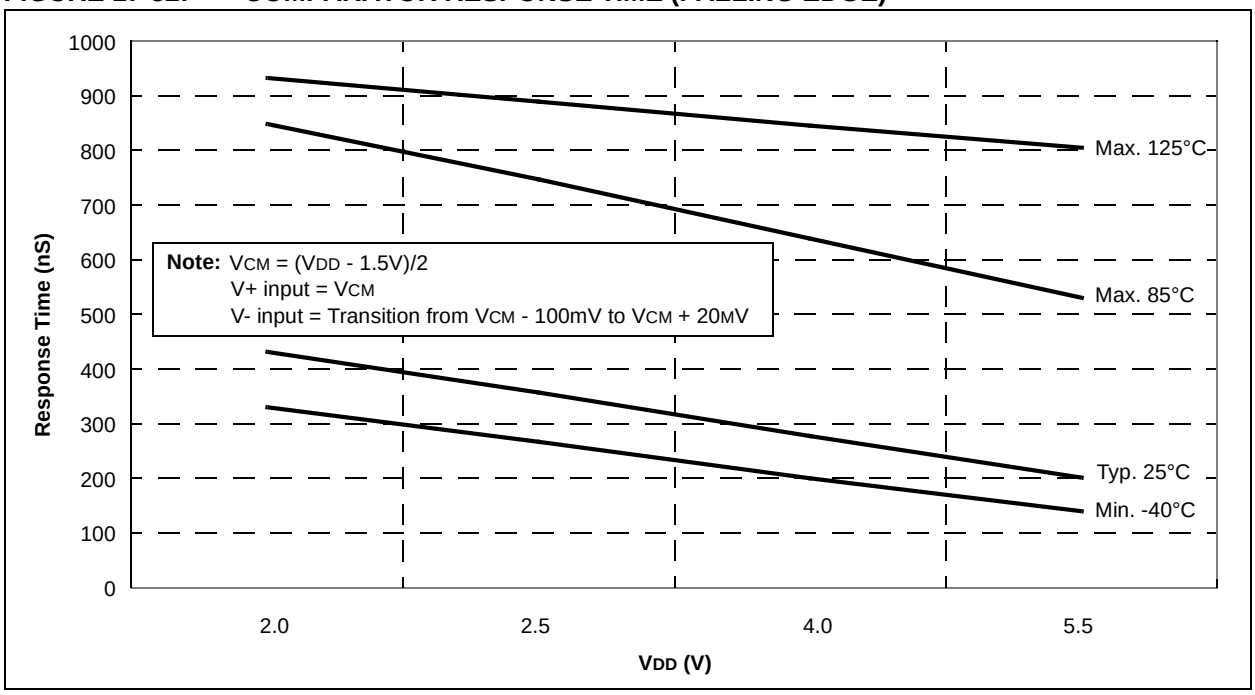


FIGURE 17-53: WDT TIME-OUT PERIOD vs. VDD OVER TEMPERATURE