

Welcome to [E-XFL.COM](http://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-SOIC (0.154", 3.90mm Width)
Supplier Device Package	8-SOIC
Purchase URL	<a href="https://www.e-xfl.com/product-detail/microchip-technology/pic12hv615t-i-sn">https://www.e-xfl.com/product-detail/microchip-technology/pic12hv615t-i-sn</a>

# PIC12F609/615/617/12HV609/615

**TABLE 2-3: PIC12F609/HV609 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	$\overline{\text{GPPU}}$	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 <sup>(1)</sup>	RP1 <sup>(1)</sup>	RP0	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C	0001 1xxx	18, 116
84h	FSR	Indirect Data Memory Address Pointer								xxxx xxxx	25, 116
85h	TRISIO	—	—	TRISIO5	TRISIO4	TRISIO3 <sup>(4)</sup>	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 <sup>(3)</sup>	0000 0000	20, 116
8Ch	PIE1	—	—	—	—	CMIE	—	—	TMR1IE	---- 0--0	21, 116
8Dh	—	Unimplemented								—	—
8Eh	PCON	—	—	—	—	—	—	$\overline{\text{POR}}$	$\overline{\text{BOR}}$	---- --q $\alpha$	23, 116
8Fh	—	Unimplemented								—	—
90h	OSCTUNE	—	—	—	TUN4	TUN3	TUN2	TUN1	TUN0	---0 0000	41, 116
91h	—	Unimplemented								—	—
92h	—	Unimplemented								—	—
93h	—	Unimplemented								—	—
94h	—	Unimplemented								—	—
95h	WPU <sup>(2)</sup>	—	—	WPU5	WPU4	—	WPU2	WPU1	WPU0	--11 -111	46, 116
96h	IOC	—	—	IOC5	IOC4	IOC3	IOC2	IOC1	IOC0	--00 0000	46, 116
97h	—	Unimplemented								—	—
98h	—	Unimplemented								—	—
99h	—	Unimplemented								—	—
9Ah	—	Unimplemented								—	—
9Bh	—	Unimplemented								—	—
9Ch	—	Unimplemented								—	—
9Dh	—	Unimplemented								—	—
9Eh	—	Unimplemented								—	—
9Fh	ANSEL	—	—	—	—	ANS3	—	ANS1	ANS0	---- 1-11	45, 117

**Legend:** — = Unimplemented locations read as '0', u = unchanged, x = unknown,  $\alpha$  = value depends on condition, shaded = unimplemented

- Note**
- 1: IRP and RP1 bits are reserved, always maintain these bits clear.
  - 2:  $\overline{\text{GP3}}$  pull-up is enabled when MCLRE is '1' in the Configuration Word register.
  - 3: 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.

# PIC12F609/615/617/12HV609/615

**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	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 <sup>(1)</sup>	RP1 <sup>(1)</sup>	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 <sup>(4)</sup>	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	TOIF	INTF	GPIF <sup>(3)</sup>	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 <sup>(2)</sup>	—	—	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 <sup>(7)</sup>	—	—	—	—	—	WREN	WR	RD	---- -000	29
99h	PMCON2 <sup>(7)</sup>	Program Memory Control Register 2 (not a physical register).								---- ----	—
9Ah	PMADRL <sup>(7)</sup>	PMADRL7	PMADRL6	PMADRL5	PMADRL4	PMADRL3	PMADRL2	PMADRL1	PMADRL0	0000 0000	28
9Bh	PMADRH <sup>(7)</sup>	—	—	—	—	—	PMADRH2	PMADRH1	PMADRH0	---- -000	28
9Ch	PMDATL <sup>(7)</sup>	PMDATL7	PMDATL6	PMDATL5	PMDATL4	PMDATL3	PMDATL2	PMDATL1	PMDATL0	0000 0000	28
9Dh	PMDATH <sup>(7)</sup>	—	—	Program Memory Data Register High Byte.						--00 0000	28
9Eh	ADRESL <sup>(5, 6)</sup>	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: 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.

# PIC12F609/615/617/12HV609/615

**TABLE 3-1: SUMMARY OF REGISTERS ASSOCIATED WITH PROGRAM MEMORY**

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
PMCON1	—	—	—	—	—	WREN	WR	RD	---- -000	---- -000
PMCON2	Program Memory Control Register 2 (not a physical register)								---- ----	---- ----
PMADRL	PMADRL7	PMADRL6	PMADRL5	PMADRL4	PMADRL3	PMADRL2	PMADRL1	PMADRL0	0000 0000	0000 0000
PMADRH	—	—	—	—	—	PMADRH2	PMADRH1	PMADRH0	---- -000	---- -000
PMDATL	PMDATL7	PMDATL6	PMDATL5	PMDATL4	PMDATL3	PMDATL2	PMDATL1	PMDATL0	0000 0000	0000 0000
PMDATH	—	—	PMDATH5	PMDATH4	PMDATH3	PMDATH2	PMDATH1	PMDATH0	--00 0000	--00 0000

**Legend:** x = unknown, u = unchanged, — = unimplemented read as '0', q = value depends upon condition.  
Shaded cells are not used by Program Memory module.

# PIC12F609/615/617/12HV609/615

**TABLE 5-1: SUMMARY OF REGISTERS ASSOCIATED WITH GPIO**

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
ANSEL	—	ADCS2 <sup>(1)</sup>	ADCS1 <sup>(1)</sup>	ADCS0 <sup>(1)</sup>	ANS3	ANS2 <sup>(1)</sup>	ANS1	ANS0	-000 1111	-000 1111
CMCON0	CMON	COUT	CMOE	CMPOL	—	CMR	—	CMCH	0000 -0-0	0000 -0-0
INTCON	GIE	PEIE	T0IE	INTE	GPIE	T0IF	INTF	GPIF	0000 0000	0000 0000
IOC	—	—	IOC5	IOC4	IOC3	IOC2	IOC1	IOC0	--00 0000	--00 0000
OPTION_REG	GPPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
GPIO	—	—	GP5	GP4	GP3	GP2	GP1	GP0	--xx xxxx	--u0 u000
TRISIO	—	—	TRISIO5	TRISIO4	TRISIO3	TRISIO2	TRISIO1	TRISIO0	--11 1111	--11 1111
WPU	—	—	WPU5	WPU4	WPU3	WPU2	WPU1	WPU0	--11 1111	--11 -111
T1CON	T1GINV	TMR1GE	TICKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON	0000 0000	uuuu uuuu
CCP1CON <sup>(1)</sup>	P1M	—	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	0-00 0000	0-00 0000
APFCON <sup>(1)</sup>	—	—	—	T1GSEL	—	—	P1BSEL	P1ASEL	---0 --00	---0 --00

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

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

# PIC12F609/615/617/12HV609/615

---

NOTES:

# PIC12F609/615/617/12HV609/615

## 7.2.1 INTERNAL CLOCK SOURCE

When the internal clock source is selected, the TMR1H:TMR1L register pair will increment on multiples of TCY as determined by the Timer1 prescaler.

## 7.2.2 EXTERNAL CLOCK SOURCE

When the external clock source is selected, the Timer1 module may work as a timer or a counter.

When counting, Timer1 is incremented on the rising edge of the external clock input T1CKI. In addition, the Counter mode clock can be synchronized to the microcontroller system clock or run asynchronously.

If an external clock oscillator is needed (and the microcontroller is using the INTOSC without CLKOUT), Timer1 can use the LP oscillator as a clock source.

In Counter mode, a falling edge must be registered by the counter prior to the first incrementing rising edge after one or more of the following conditions:

- Timer1 is enabled after POR or BOR Reset
- A write to TMR1H or TMR1L
- T1CKI is high when Timer1 is disabled and when Timer1 is re-enabled T1CKI is low. See Figure 7-2.

## 7.3 Timer1 Prescaler

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

## 7.4 Timer1 Oscillator

A low-power 32.768 kHz crystal oscillator is built-in between pins OSC1 (input) and OSC2 (output). The oscillator is enabled by setting the T1OSCEN control bit of the T1CON register. The oscillator will continue to run during Sleep.

The Timer1 oscillator is shared with the system LP oscillator. Thus, Timer1 can use this mode only when the primary system clock is derived from the internal oscillator or when in LP oscillator mode. The user must provide a software time delay to ensure proper oscillator start-up.

TRISIO5 and TRISIO4 bits are set when the Timer1 oscillator is enabled. GP5 and GP4 bits read as '0' and TRISIO5 and TRISIO4 bits read as '1'.

**Note:** The oscillator requires a start-up and stabilization time before use. Thus, T1OSCEN should be set and a suitable delay observed prior to enabling Timer1.

## 7.5 Timer1 Operation in Asynchronous Counter Mode

If control bit  $\overline{T1SYNC}$  of the T1CON register is set, the external clock input is not synchronized. The timer continues to increment asynchronous to the internal phase clocks. The timer will continue to run during Sleep and can generate an interrupt on overflow, which will wake-up the processor. However, special precautions in software are needed to read/write the timer (see **Section 7.5.1 "Reading and Writing Timer1 in Asynchronous Counter Mode"**).

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

**Note:** In asynchronous counter mode or when using the internal oscillator and T1ACS=1, Timer1 can not be used as a time base for the capture or compare modes of the ECCP module (for PIC12F615/617/HV615 only).

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

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

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

# PIC12F609/615/617/12HV609/615

## 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<sup>(1, 2)</sup>  
 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<sup>(2)</sup>  
 1 = Enabled  
 0 = Disabled
- bit 3-0    **VR<3:0>:** Comparator Voltage Reference CVREF Value Selection bits ( $0 \leq VR<3:0> \leq 15$ )  
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.  
**Note 2:** When CMVREN is low and the FVREN bit is low, the CVREF signal should provide Vss to the comparator.



# PIC12F609/615/617/12HV609/615

---

## 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.

# PIC12F609/615/617/12HV609/615

## 10.3 A/D Acquisition Requirements

For the ADC to meet its specified accuracy, the charge holding capacitor (CHOLD) must be allowed to fully charge to the input channel voltage level. The Analog Input model is shown in Figure 10-4. The source impedance ( $R_S$ ) and the internal sampling switch ( $R_{SS}$ ) impedance directly affect the time required to charge the capacitor CHOLD. The sampling switch ( $R_{SS}$ ) impedance varies over the device voltage ( $V_{DD}$ ), see Figure 10-4.

**The maximum recommended impedance for analog sources is 10 k $\Omega$ .** As the source impedance is decreased, the acquisition time may be decreased. After the analog input channel is selected (or changed),

an A/D acquisition must be done before the conversion can be started. To calculate the minimum acquisition time, Equation 10-1 may be used. This equation assumes that 1/2 LSB error is used (1024 steps for the ADC). The 1/2 LSB error is the maximum error allowed for the ADC to meet its specified resolution.

### EQUATION 10-1: ACQUISITION TIME EXAMPLE

*Assumptions: Temperature = 50°C and external impedance of 10k $\Omega$  5.0V VDD*

$$\begin{aligned} T_{ACQ} &= \text{Amplifier Settling Time} + \text{Hold Capacitor Charging Time} + \text{Temperature Coefficient} \\ &= T_{AMP} + T_C + T_{COFF} \\ &= 2\mu s + T_C + [(Temperature - 25^\circ C)(0.05\mu s/^\circ C)] \end{aligned}$$

*The value for  $T_C$  can be approximated with the following equations:*

$$V_{APPLIED} \left( 1 - \frac{1}{2047} \right) = V_{CHOLD} \quad ;[1] \text{ } V_{CHOLD} \text{ charged to within } 1/2 \text{ lsb}$$

$$V_{APPLIED} \left( 1 - e^{-\frac{T_C}{RC}} \right) = V_{CHOLD} \quad ;[2] \text{ } V_{CHOLD} \text{ charge response to } V_{APPLIED}$$

$$V_{APPLIED} \left( 1 - e^{-\frac{T_C}{RC}} \right) = V_{APPLIED} \left( 1 - \frac{1}{2047} \right) \quad ;\text{combining [1] and [2]}$$

*Solving for  $T_C$ :*

$$\begin{aligned} T_C &= -CHOLD(R_{IC} + R_{SS} + R_S) \ln(1/2047) \\ &= -10pF(1k\Omega + 7k\Omega + 10k\Omega) \ln(0.0004885) \\ &= 1.37\mu s \end{aligned}$$

*Therefore:*

$$\begin{aligned} T_{ACQ} &= 2\mu s + 1.37\mu s + [(50^\circ C - 25^\circ C)(0.05\mu s/^\circ C)] \\ &= 4.67\mu s \end{aligned}$$

**Note 1:** The reference voltage ( $V_{REF}$ ) has no effect on the equation, since it cancels itself out.

**2:** The charge holding capacitor (CHOLD) is not discharged after each conversion.

**3:** The maximum recommended impedance for analog sources is 10 k $\Omega$ . This is required to meet the pin leakage specification.

# PIC12F609/615/617/12HV609/615

---

## 11.4.2 START-UP CONSIDERATIONS

When any PWM mode is used, the application hardware must use the proper external pull-up and/or pull-down resistors on the PWM output pins.

**Note:** When the microcontroller is released from Reset, all of the I/O pins are in the high-impedance state. The external circuits must keep the power switch devices in the OFF state until the microcontroller drives the I/O pins with the proper signal levels or activates the PWM output(s).

The CCP1M<1:0> bits of the CCP1CON register allow the user to choose whether the PWM output signals are active-high or active-low for each PWM output pin (P1A and P1B). The PWM output polarities must be selected before the PWM pin output drivers are enabled. Changing the polarity configuration while the PWM pin output drivers are enable is not recommended since it may result in damage to the application circuits.

The P1A and P1B output latches may not be in the proper states when the PWM module is initialized. Enabling the PWM pin output drivers at the same time as the Enhanced PWM modes may cause damage to the application circuit. The Enhanced PWM modes must be enabled in the proper Output mode and complete a full PWM cycle before configuring the PWM pin output drivers. The completion of a full PWM cycle is indicated by the TMR2IF bit of the PIR1 register being set as the second PWM period begins.

## 11.4.3 OPERATION DURING SLEEP

When the device is placed in sleep, the allocated timer 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, it will continue from this state.

# PIC12F609/615/617/12HV609/615

## 11.4.4 ENHANCED PWM AUTO-SHUTDOWN MODE

The PWM mode supports an Auto-Shutdown mode that will disable the PWM outputs when an external shutdown event occurs. Auto-Shutdown mode places the PWM output pins into a predetermined state. This mode is used to help prevent the PWM from damaging the application.

The auto-shutdown sources are selected using the ECCPASx bits of the ECCPAS register. A shutdown event may be generated by:

- A logic '0' on the INT pin
- Comparator
- Setting the ECCPASE bit in firmware

A shutdown condition is indicated by the ECCPASE (Auto-Shutdown Event Status) bit of the ECCPAS register. If the bit is a '0', the PWM pins are operating normally. If the bit is a '1', the PWM outputs are in the shutdown state. Refer to Figure 1.

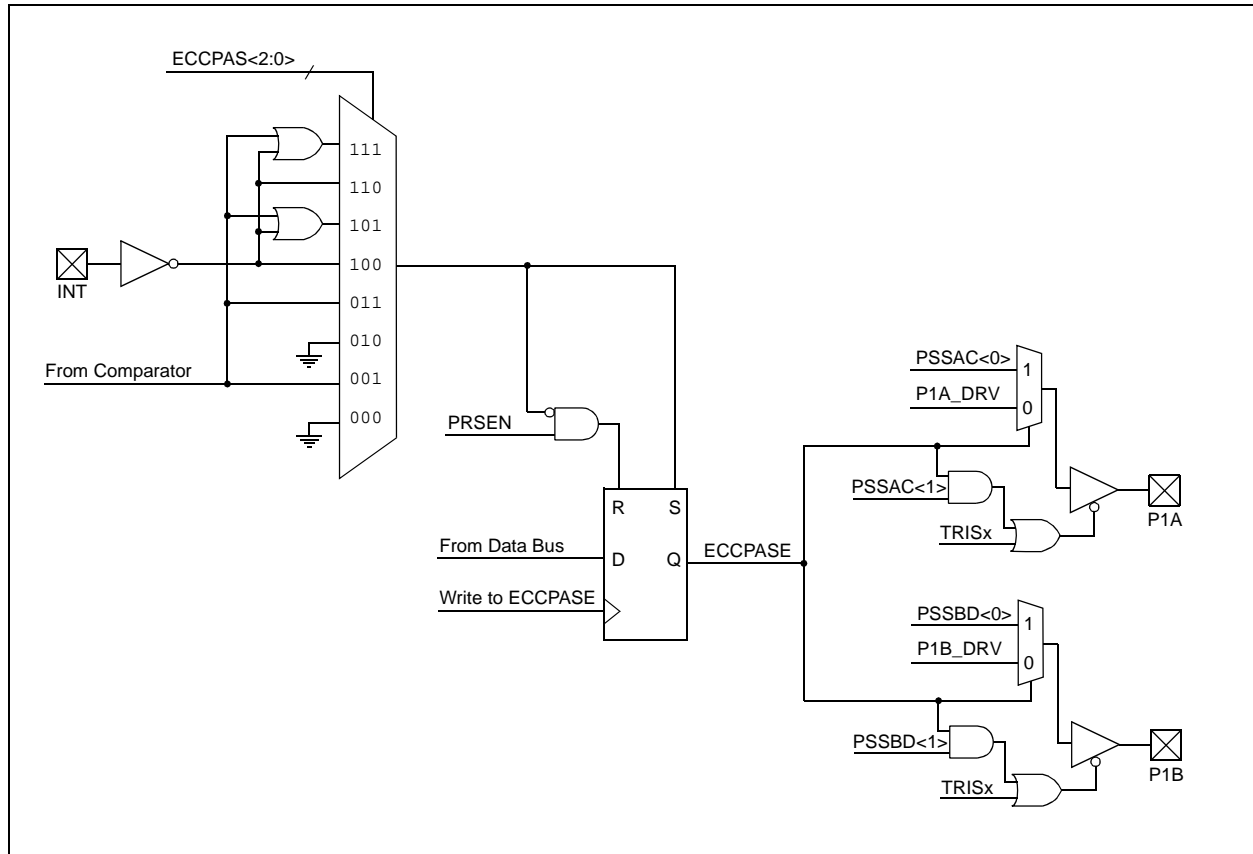
When a shutdown event occurs, two things happen:

The ECCPASE bit is set to '1'. The ECCPASE will remain set until cleared in firmware or an auto-restart occurs (see **Section 11.4.5 "Auto-Restart Mode"**).

The enabled PWM pins are asynchronously placed in their shutdown states. The state of P1A is determined by the PSSAC bit. The state of P1B is determined by the PSSBD bit. The PSSAC and PSSBD bits are located in the ECCPAS register. Each pin may be placed into one of three states:

- Drive logic '1'
- Drive logic '0'
- Tri-state (high-impedance)

**FIGURE 11-10: AUTO-SHUTDOWN BLOCK DIAGRAM**



# PIC12F609/615/617/12HV609/615

## 14.0 INSTRUCTION SET SUMMARY

The PIC12F609/615/617/12HV609/615 instruction set is highly orthogonal and is comprised of three basic categories:

- **Byte-oriented** operations
- **Bit-oriented** operations
- **Literal and control** operations

Each PIC16 instruction is a 14-bit word divided into an **opcode**, which specifies the instruction type and one or more **operands**, which further specify the operation of the instruction. The formats for each of the categories is presented in Figure 14-1, while the various opcode fields are summarized in Table 14-1.

Table 14-2 lists the instructions recognized by the MPASM™ assembler.

For **byte-oriented** instructions, 'f' represents a file register designator and 'd' represents a destination designator. The file register designator specifies which file register is to be used by the instruction.

The destination designator specifies where the result of the operation is to be placed. If 'd' is zero, the result is placed in the W register. If 'd' is one, the result is placed in the file register specified in the instruction.

For **bit-oriented** instructions, 'b' represents a bit field designator, which selects the bit affected by the operation, while 'f' represents the address of the file in which the bit is located.

For **literal and control** operations, 'k' represents an 8-bit or 11-bit constant, or literal value.

One instruction cycle consists of four oscillator periods; for an oscillator frequency of 4 MHz, this gives a normal instruction execution time of 1 μs. All instructions are executed within a single instruction cycle, unless a conditional test is true, or the program counter is changed as a result of an instruction. When this occurs, the execution takes two instruction cycles, with the second cycle executed as a NOP.

All instruction examples use the format '0xhh' to represent a hexadecimal number, where 'h' signifies a hexadecimal digit.

### 14.1 Read-Modify-Write Operations

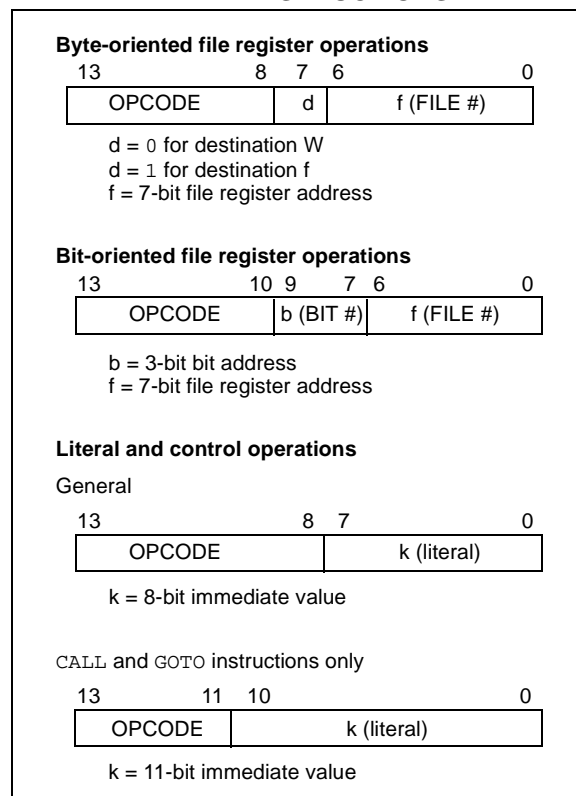
Any instruction that specifies a file register as part of the instruction performs a Read-Modify-Write (RMW) operation. The register is read, the data is modified, and the result is stored according to either the instruction or the destination designator 'd'. A read operation is performed on a register even if the instruction writes to that register.

For example, a CLRF GPIO instruction will read GPIO, clear all the data bits, then write the result back to GPIO. This example would have the unintended consequence of clearing the condition that set the GPIF flag.

TABLE 14-1: OPCODE FIELD DESCRIPTIONS

Field	Description
f	Register file address (0x00 to 0x7F)
W	Working register (accumulator)
b	Bit address within an 8-bit file register
k	Literal field, constant data or label
x	Don't care location (= 0 or 1). The assembler will generate code with x = 0. It is the recommended form of use for compatibility with all Microchip software tools.
d	Destination select; d = 0: store result in W, d = 1: store result in file register f. Default is d = 1.
PC	Program Counter
$\overline{TO}$	Time-out bit
C	Carry bit
DC	Digit carry bit
Z	Zero bit
$\overline{PD}$	Power-down bit

FIGURE 14-1: GENERAL FORMAT FOR INSTRUCTIONS



# PIC12F609/615/617/12HV609/615

**TABLE 14-2: PIC12F609/615/617/12HV609/615 INSTRUCTION SET**

Mnemonic, Operands	Description	Cycles	14-Bit Opcode				Status Affected	Notes	
			MSb	LSb					
<b>BYTE-ORIENTED FILE REGISTER OPERATIONS</b>									
ADDWF	f, d	Add W and f	1	00	0111	dfff	ffff	C, DC, Z	1, 2
ANDWF	f, d	AND W with f	1	00	0101	dfff	ffff	Z	1, 2
CLRF	f	Clear f	1	00	0001	1fff	ffff	Z	2
CLRWF	–	Clear W	1	00	0001	0xxx	xxxx	Z	
COMF	f, d	Complement f	1	00	1001	dfff	ffff	Z	1, 2
DECf	f, d	Decrement f	1	00	0011	dfff	ffff	Z	1, 2
DECFSZ	f, d	Decrement f, Skip if 0	1(2)	00	1011	dfff	ffff		1, 2, 3
INCF	f, d	Increment f	1	00	1010	dfff	ffff	Z	1, 2
INCFSZ	f, d	Increment f, Skip if 0	1(2)	00	1111	dfff	ffff		1, 2, 3
IORWF	f, d	Inclusive OR W with f	1	00	0100	dfff	ffff	Z	1, 2
MOVF	f, d	Move f	1	00	1000	dfff	ffff	Z	1, 2
MOVWF	f	Move W to f	1	00	0000	1fff	ffff		
NOP	–	No Operation	1	00	0000	0xx0	0000		
RLF	f, d	Rotate Left f through Carry	1	00	1101	dfff	ffff	C	1, 2
RRF	f, d	Rotate Right f through Carry	1	00	1100	dfff	ffff	C	1, 2
SUBWF	f, d	Subtract W from f	1	00	0010	dfff	ffff	C, DC, Z	1, 2
SWAPF	f, d	Swap nibbles in f	1	00	1110	dfff	ffff		1, 2
XORWF	f, d	Exclusive OR W with f	1	00	0110	dfff	ffff	Z	1, 2
<b>BIT-ORIENTED FILE REGISTER OPERATIONS</b>									
BCF	f, b	Bit Clear f	1	01	00bb	bfff	ffff		1, 2
BSF	f, b	Bit Set f	1	01	01bb	bfff	ffff		1, 2
BTFSC	f, b	Bit Test f, Skip if Clear	1(2)	01	10bb	bfff	ffff		3
BTFSS	f, b	Bit Test f, Skip if Set	1(2)	01	11bb	bfff	ffff		3
<b>LITERAL AND CONTROL OPERATIONS</b>									
ADDLW	k	Add literal and W	1	11	111x	kkkk	kkkk	C, DC, Z	
ANDLW	k	AND literal with W	1	11	1001	kkkk	kkkk	Z	
CALL	k	Call Subroutine	2	10	0kkk	kkkk	kkkk		
CLRWDT	–	Clear Watchdog Timer	1	00	0000	0110	0100	$\overline{TO}, \overline{PD}$	
GOTO	k	Go to address	2	10	1kkk	kkkk	kkkk		
IORLW	k	Inclusive OR literal with W	1	11	1000	kkkk	kkkk	Z	
MOVLW	k	Move literal to W	1	11	00xx	kkkk	kkkk		
RETFIE	–	Return from interrupt	2	00	0000	0000	1001		
RETLW	k	Return with literal in W	2	11	01xx	kkkk	kkkk		
RETURN	–	Return from Subroutine	2	00	0000	0000	1000		
SLEEP	–	Go into Standby mode	1	00	0000	0110	0011	$\overline{TO}, \overline{PD}$	
SUBLW	k	Subtract W from literal	1	11	110x	kkkk	kkkk	C, DC, Z	
XORLW	k	Exclusive OR literal with W	1	11	1010	kkkk	kkkk	Z	

- Note 1:** When an I/O register is modified as a function of itself (e.g., MOVF GPIO, 1), the value used will be that value present on the pins themselves. For example, if the data latch is '1' for a pin configured as input and is driven low by an external device, the data will be written back with a '0'.
- 2:** If this instruction is executed on the TMR0 register (and where applicable, d = 1), the prescaler will be cleared if assigned to the Timer0 module.
- 3:** If the Program Counter (PC) is modified, or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a NOP.

# PIC12F609/615/617/12HV609/615

## 14.2 Instruction Descriptions

### **ADDLW**      **Add literal and W**

**Syntax:**      [ *label* ] ADDLW    *k*  
**Operands:**     $0 \leq k \leq 255$   
**Operation:**     $(W) + k \rightarrow (W)$   
**Status Affected:**    C, DC, Z  
**Description:**    The contents of the W register are added to the eight-bit literal 'k' and the result is placed in the W register.

### **BCF**            **Bit Clear f**

**Syntax:**      [ *label* ] BCF    *f*,*b*  
**Operands:**     $0 \leq f \leq 127$   
                   $0 \leq b \leq 7$   
**Operation:**     $0 \rightarrow (f<b>)$   
**Status Affected:**    None  
**Description:**    Bit 'b' in register 'f' is cleared.

### **ADDWF**        **Add W and f**

**Syntax:**      [ *label* ] ADDWF    *f*,*d*  
**Operands:**     $0 \leq f \leq 127$   
                   $d \in [0,1]$   
**Operation:**     $(W) + (f) \rightarrow (\text{destination})$   
**Status Affected:**    C, DC, Z  
**Description:**    Add the contents of the W register with 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'.

### **BSF**            **Bit Set f**

**Syntax:**      [ *label* ] BSF    *f*,*b*  
**Operands:**     $0 \leq f \leq 127$   
                   $0 \leq b \leq 7$   
**Operation:**     $1 \rightarrow (f<b>)$   
**Status Affected:**    None  
**Description:**    Bit 'b' in register 'f' is set.

### **ANDLW**        **AND literal with W**

**Syntax:**      [ *label* ] ANDLW    *k*  
**Operands:**     $0 \leq k \leq 255$   
**Operation:**     $(W) .\text{AND.} (k) \rightarrow (W)$   
**Status Affected:**    Z  
**Description:**    The contents of W register are AND'ed with the eight-bit literal 'k'. The result is placed in the W register.

### **BTFSC**         **Bit Test f, Skip if Clear**

**Syntax:**      [ *label* ] BTFSC    *f*,*b*  
**Operands:**     $0 \leq f \leq 127$   
                   $0 \leq b \leq 7$   
**Operation:**    skip if  $(f<b>) = 0$   
**Status Affected:**    None  
**Description:**    If bit 'b' in register 'f' is '1', the next instruction is executed.  
                  If bit 'b' in register 'f' is '0', the next instruction is discarded, and a NOP is executed instead, making this a two-cycle instruction.

### **ANDWF**        **AND W with f**

**Syntax:**      [ *label* ] ANDWF    *f*,*d*  
**Operands:**     $0 \leq f \leq 127$   
                   $d \in [0,1]$   
**Operation:**     $(W) .\text{AND.} (f) \rightarrow (\text{destination})$   
**Status Affected:**    Z  
**Description:**    AND the W register with 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'.

# PIC12F609/615/617/12HV609/615

<b>RETFIE</b>	<b>Return from Interrupt</b>
Syntax:	[ <i>label</i> ] RETFIE
Operands:	None
Operation:	TOS → PC, 1 → GIE
Status Affected:	None
Description:	Return from Interrupt. Stack is POPed and Top-of-Stack (TOS) is loaded in the PC. Interrupts are enabled by setting Global Interrupt Enable bit, GIE (INTCON<7>). This is a two-cycle instruction.
Words:	1
Cycles:	2
<u>Example:</u>	<pre>RETFIE</pre> <p>After Interrupt</p> <pre>PC = TOS GIE = 1</pre>

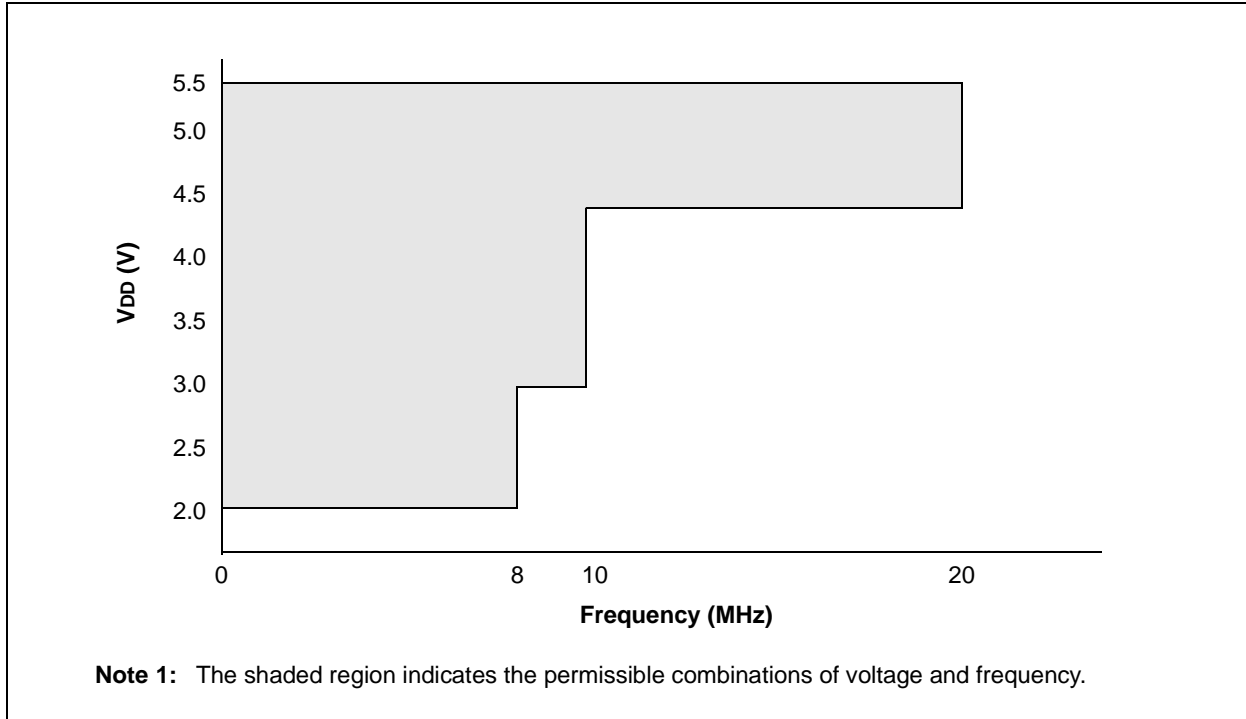
<b>RETLW</b>	<b>Return with literal in W</b>
Syntax:	[ <i>label</i> ] RETLW k
Operands:	0 ≤ k ≤ 255
Operation:	k → (W); TOS → PC
Status Affected:	None
Description:	The W register is loaded with the eight-bit literal 'k'. The program counter is loaded from the top of the stack (the return address). This is a two-cycle instruction.
Words:	1
Cycles:	2
<u>Example:</u>	<pre>CALL TABLE;W contains                 ;table offset                 ;value GOTO DONE TABLE • • ADDWF PC ;W = offset RETLW k1 ;Begin table RETLW k2 ; • • RETLW kn ;End of table DONE</pre> <p>Before Instruction W = 0x07</p> <p>After Instruction W = value of k8</p>

<b>RETURN</b>	<b>Return from Subroutine</b>
Syntax:	[ <i>label</i> ] RETURN
Operands:	None
Operation:	TOS → PC
Status Affected:	None
Description:	Return from subroutine. The stack is POPed and the top of the stack (TOS) is loaded into the program counter. This is a two-cycle instruction.

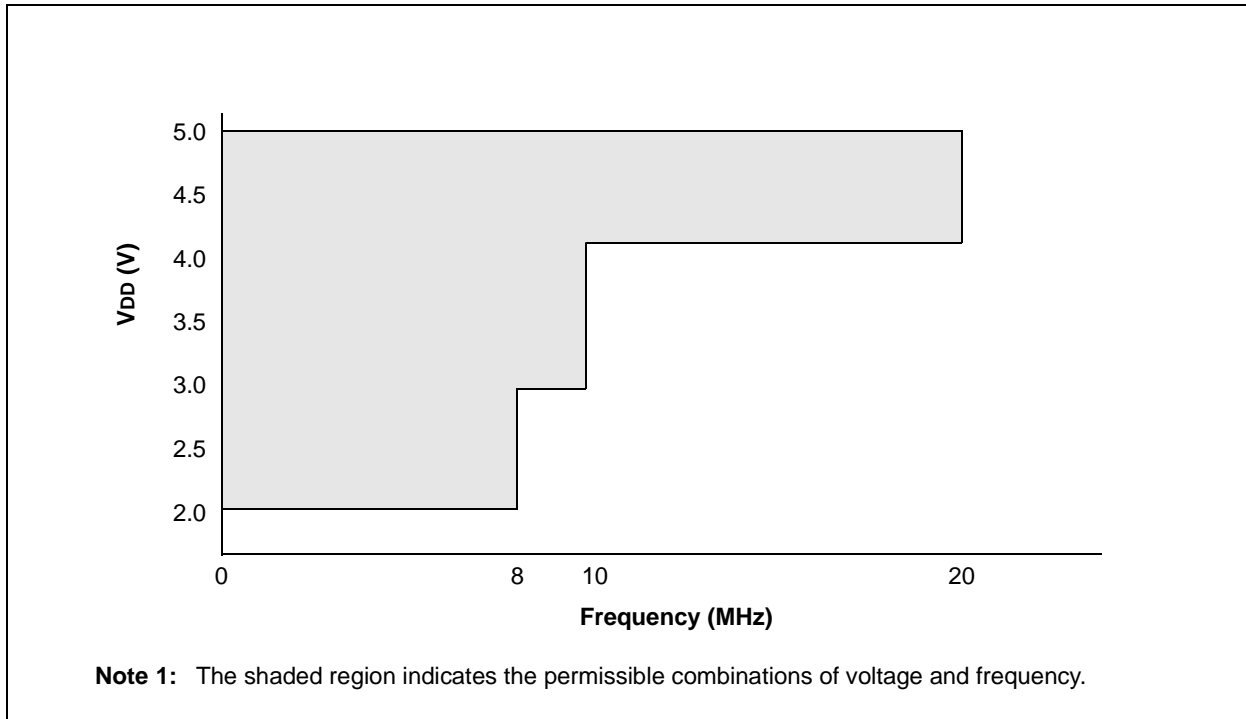


# PIC12F609/615/617/12HV609/615

**FIGURE 16-1: PIC12F609/615/617 VOLTAGE-FREQUENCY GRAPH,  $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$**

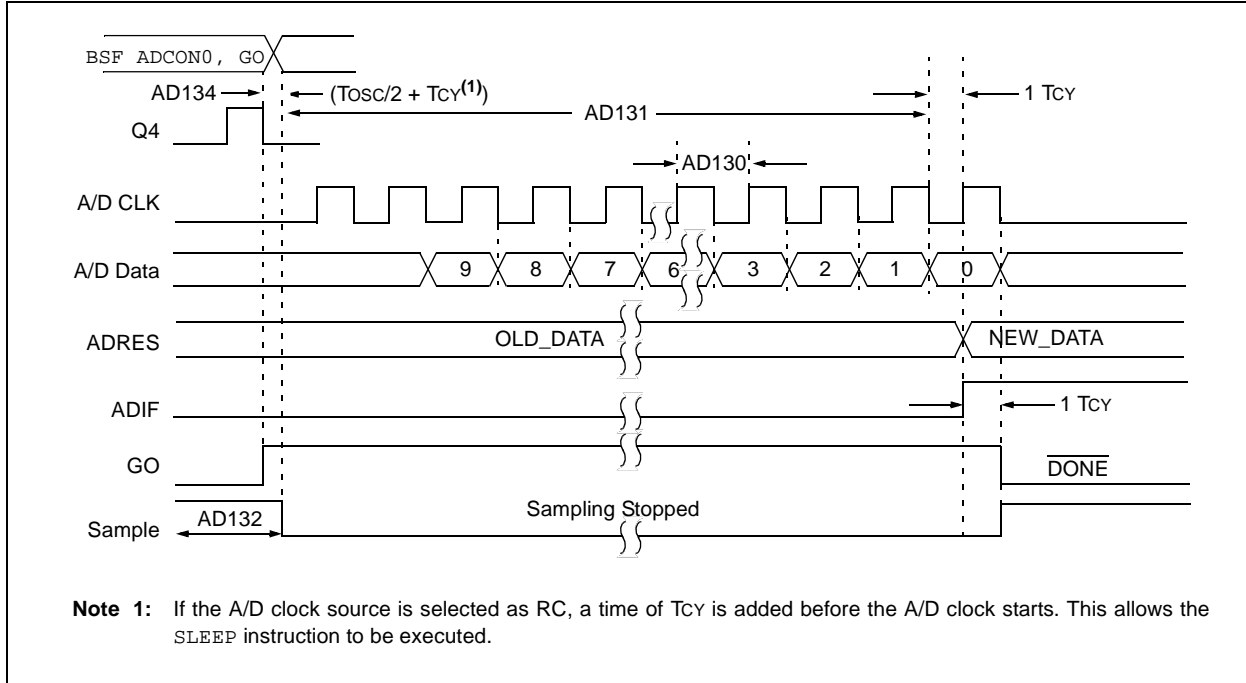


**FIGURE 16-2: PIC12HV609/615 VOLTAGE-FREQUENCY GRAPH,  $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$**



# PIC12F609/615/617/12HV609/615

**FIGURE 16-11: PIC12F615/617/HV615 A/D CONVERSION TIMING (SLEEP MODE)**

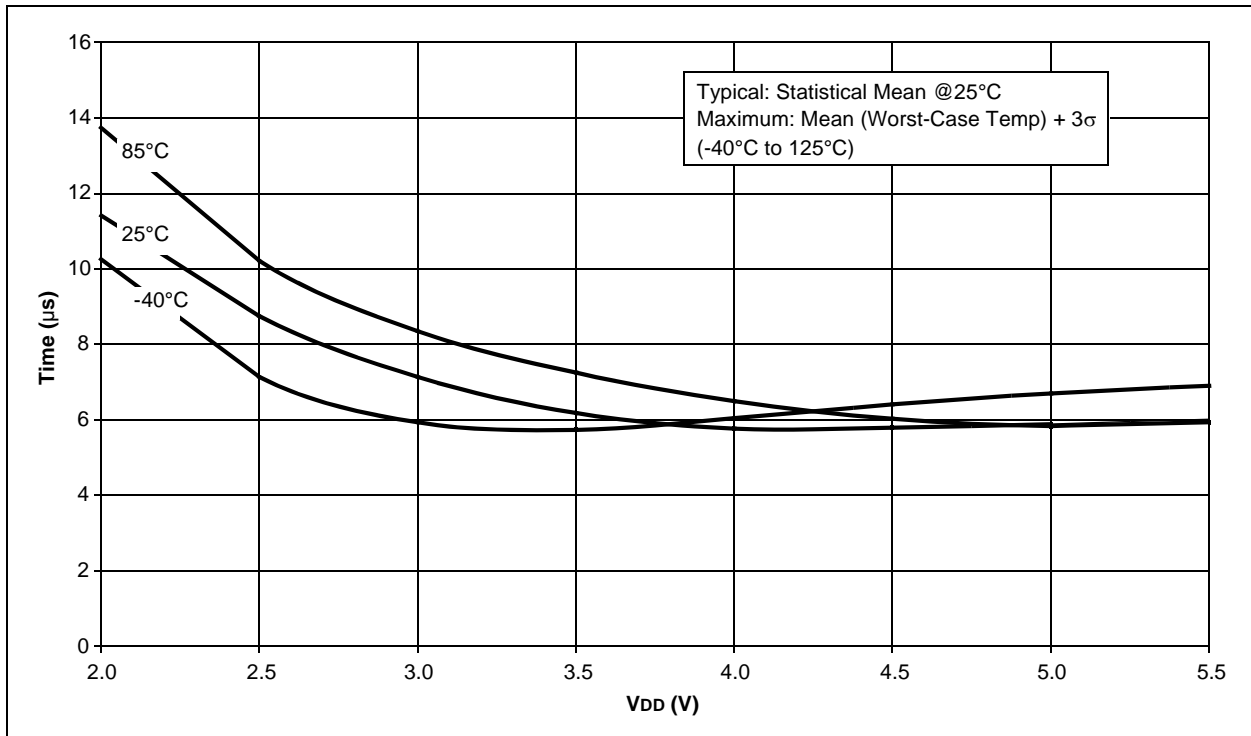


# PIC12F609/615/617/12HV609/615

FIGURE 17-39: SCHMITT TRIGGER INPUT THRESHOLD  $V_{IN}$  vs.  $V_{DD}$  OVER TEMPERATURE



FIGURE 17-40: TYPICAL HFINTOSC START-UP TIMES vs.  $V_{DD}$  OVER TEMPERATURE



# PIC12F609/615/617/12HV609/615

## INDEX

### A

A/D	
Specifications.....	164, 165
Absolute Maximum Ratings .....	143
AC Characteristics	
Industrial and Extended .....	156
Load Conditions .....	155
ADC	
Acquisition Requirements .....	86
Associated registers.....	88
Block Diagram.....	79
Calculating Acquisition Time.....	86
Channel Selection.....	80
Configuration.....	80
Configuring Interrupt .....	83
Conversion Clock.....	80
Conversion Procedure .....	83
Internal Sampling Switch (Rss) Impedance.....	86
Interrupts.....	81
Operation .....	82
Operation During Sleep .....	82
Port Configuration .....	80
Reference Voltage (VREF).....	80
Result Formatting.....	82
Source Impedance.....	86
Special Event Trigger.....	82
Starting an A/D Conversion .....	82
ADC (PIC12F615/617/HV615 Only) .....	79
ADCON0 Register.....	84
ADRESH Register (ADFM = 0) .....	85
ADRESH Register (ADFM = 1) .....	85
ADRESL Register (ADFM = 0).....	85
ADRESL Register (ADFM = 1).....	85
Analog Input Connection Considerations.....	68
Analog-to-Digital Converter. See ADC	
ANSEL Register (PIC12F609/HV609) .....	45
ANSEL Register (PIC12F615/617/HV615) .....	45
APFCON Register.....	24
Assembler	
MPASM Assembler.....	140

### B

Block Diagrams	
(CCP) Capture Mode Operation .....	90
ADC .....	79
ADC Transfer Function .....	87
Analog Input Model.....	68, 87
Auto-Shutdown .....	101
CCP PWM.....	94
Clock Source.....	37
Comparator .....	67
Compare .....	92
Crystal Operation .....	39
External RC Mode.....	40
GP0 and GP1 Pins.....	47
GP2 Pins.....	48
GP3 Pin.....	49
GP4 Pin.....	50
GP5 Pin.....	51
In-Circuit Serial Programming Connections.....	125
Interrupt Logic .....	119
MCLR Circuit.....	111
On-Chip Reset Circuit.....	110

PIC12F609/12HV609 .....	7
PIC12F615/617/12HV615 .....	8
PWM (Enhanced) .....	97
Resonator Operation .....	39
Timer1 .....	57, 58
Timer2 .....	65
TMR0/WDT Prescaler .....	53
Watchdog Timer .....	122
Brown-out Reset (BOR).....	112
Associated Registers.....	113
Specifications .....	160
Timing and Characteristics .....	159

### C

C Compilers	
MPLAB C18.....	140
MPLAB C30.....	140
Calibration Bits.....	109
Capture Module. See Enhanced Capture/Compare/ PWM (ECCP)	
Capture/Compare/PWM (CCP)	
Associated registers w/ Capture .....	91
Associated registers w/ Compare .....	93
Associated registers w/ PWM.....	105
Capture Mode.....	90
CCP1 Pin Configuration .....	90
Compare Mode.....	92
CCP1 Pin Configuration .....	92
Software Interrupt Mode .....	90, 92
Special Event Trigger .....	92
Timer1 Mode Selection.....	90, 92
Prescaler .....	90
PWM Mode.....	94
Duty Cycle .....	95
Effects of Reset .....	96
Example PWM Frequencies and Resolutions, 20 MHz.....	95
Example PWM Frequencies and Resolutions, 8 MHz .....	95
Operation in Sleep Mode.....	96
Setup for Operation .....	96
System Clock Frequency Changes .....	96
PWM Period .....	95
Setup for PWM Operation .....	96
CCP1CON (Enhanced) Register .....	89
Clock Sources	
External Modes.....	38
EC .....	38
HS .....	39
LP .....	39
OST .....	38
RC .....	40
XT.....	39
Internal Modes .....	40
INTOSC.....	40
INTOSCIO .....	40
CMCON0 Register.....	72
CMCON1 Register.....	73
Code Examples	
A/D Conversion .....	83
Assigning Prescaler to Timer0.....	54
Assigning Prescaler to WDT.....	54
Changing Between Capture Prescalers .....	90
Indirect Addressing.....	25

# PIC12F609/615/617/12HV609/615

RLF .....	136	Oscillator Parameters .....	157
RRF .....	136	Oscillator Specifications.....	156
SLEEP .....	136	Oscillator Start-up Timer (OST) Specifications .....	160
SUBLW .....	136	OSCTUNE Register.....	41
SUBWF .....	137	<b>P</b>	
SWAPF .....	137	P1A/P1B/P1C/P1D. See Enhanced Capture/Compare/ PWM (ECCP) .....	97
XORLW .....	137	Packaging .....	195
XORWF .....	137	Marking.....	195
Summary Table.....	130	PDIP Details .....	196
INTCON Register .....	20	PCL and PCLATH.....	25
Internal Oscillator Block		Stack.....	25
INTOSC		PCON Register .....	23, 113
Specifications.....	157, 158	PICSTART Plus Development Programmer.....	142
Internal Sampling Switch (Rss) Impedance .....	86	PIE1 Register .....	21
Internet Address.....	209	Pin Diagram	
Interrupts.....	118	PIC12F609/HV609 (PDIP, SOIC, MSOP, DFN).....	4
ADC .....	83	PIC12F615/617/HV615 (PDIP, SOIC, MSOP, DFN)....	5
Associated Registers .....	120	Pinout Descriptions	
Context Saving.....	121	PIC12F609/12HV609 .....	9
GP2/INT .....	118	PIC12F615/617/12HV615 .....	10
GPIO Interrupt-on-Change.....	119	PIR1 Register .....	22
Interrupt-on-Change.....	44	PMADRH and PMADRL Registers .....	27
Timer0.....	119	PMCON1 and PMCON2 Registers.....	27
TMR1 .....	60	Power-Down Mode (Sleep).....	123
INTOSC Specifications .....	157, 158	Power-on Reset (POR).....	111
IOC Register .....	46	Power-up Timer (PWRT) .....	111
<b>L</b>		Specifications .....	160
Load Conditions .....	155	Precision Internal Oscillator Parameters .....	158
<b>M</b>		Prescaler	
MCLR .....	111	Shared WDT/Timer0.....	54
Internal .....	111	Switching Prescaler Assignment .....	54
Memory Organization.....	11	Program Memory.....	11
Data .....	11	Map and Stack.....	11
Program .....	11	Programming, Device Instructions.....	129
Microchip Internet Web Site .....	209	Protection Against Spurious Write.....	32
Migrating from other PICmicro Devices .....	203	PWM Mode. See Enhanced Capture/Compare/PWM.....	97
MPLAB ASM30 Assembler, Linker, Librarian .....	140	PWM1CON Register.....	105
MPLAB ICD 2 In-Circuit Debugger .....	141	<b>R</b>	
MPLAB ICE 2000 High-Performance Universal In-Circuit Emulator .....	141	Reader Response.....	210
MPLAB Integrated Development Environment Software ..	139	Reading the Flash Program Memory.....	30
MPLAB PM3 Device Programmer .....	141	Read-Modify-Write Operations .....	129
MPLAB REAL ICE In-Circuit Emulator System.....	141	Registers	
MPLINK Object Linker/MPLIB Object Librarian .....	140	ADCON0 (ADC Control 0) .....	84
<b>O</b>		ADRESH (ADC Result High) with ADFM = 0) .....	85
OPCODE Field Descriptions .....	129	ADRESH (ADC Result High) with ADFM = 1) .....	85
Operation During Code Protect.....	32	ADRESL (ADC Result Low) with ADFM = 0).....	85
Operation During Write Protect.....	32	ADRESL (ADC Result Low) with ADFM = 1).....	85
Operational Amplifier (OPA) Module		ANSEL (Analog Select) .....	45
AC Specifications.....	163	APFCON (Alternate Pin Function Register) .....	24
OPTION Register .....	19	CCP1CON (Enhanced CCP1 Control) .....	89
OPTION_REG Register .....	55	CMCON0 (Comparator Control 0) .....	72
Oscillator		CMCON1 (Comparator Control 1) .....	73
Associated registers.....	41, 63	CONFIG (Configuration Word) .....	108
Oscillator Module .....	27, 37	Data Memory Map (PIC12F609/HV609) .....	12
EC .....	37	Data Memory Map (PIC12F615/617/HV615) .....	13
HS.....	37	ECCPAS (Enhanced CCP Auto-shutdown Control) .	102
INTOSC .....	37	EEDAT (EEPROM Data) .....	28
INTOSCIO.....	37	EEDATH (EEPROM Data) .....	28
LP.....	37	GPIO.....	43
RC.....	37	INTCON (Interrupt Control) .....	20
RCIO.....	37	IOC (Interrupt-on-Change GPIO) .....	46
XT .....	37	OPTION_REG (OPTION).....	19