

Welcome to [E-XFL.COM](https://www.e-xfl.com)

What is "[Embedded - Microcontrollers](#)"?

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

Applications of "[Embedded - Microcontrollers](#)"

Details

Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	4MHz
Connectivity	-
Peripherals	Brown-out Detect/Reset, POR, WDT
Number of I/O	13
Program Memory Size	1.75KB (1K x 14)
Program Memory Type	OTP
EEPROM Size	-
RAM Size	80 x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 6V
Data Converters	-
Oscillator Type	External
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Through Hole
Package / Case	18-DIP (0.300", 7.62mm)
Supplier Device Package	18-PDIP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16c621-04e-p

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.

Information contained in this publication regarding device applications and the like is intended through suggestion only and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. No representation or warranty is given and no liability is assumed by Microchip Technology Incorporated with respect to the accuracy or use of such information, or infringement of patents or other intellectual property rights arising from such use or otherwise. Use of Microchip's products as critical components in life support systems is not authorized except with express written approval by Microchip. No licenses are conveyed, implicitly or otherwise, under any intellectual property rights.

Trademarks

The Microchip name and logo, the Microchip logo, KEELOQ, MPLAB, PIC, PICmicro, PICSTART, PRO MATE and PowerSmart are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.


FilterLab, microID, MXDEV, MXLAB, PICMASTER, SEEVAL and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

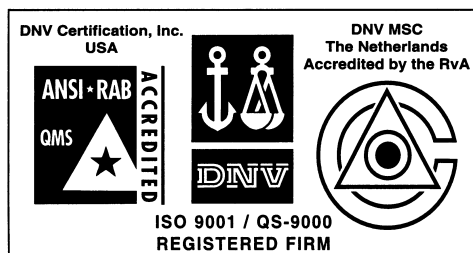
Accuron, Application Maestro, dsPIC, dsPICDEM, dsPICDEM.net, ECONOMONITOR, FanSense, FlexROM, fuzzyLAB, In-Circuit Serial Programming, ICSP, ICEPIC, microPort, Migratable Memory, MPASM, MPLIB, MPLINK, MPSIM, PICC, PICkit, PICDEM, PICDEM.net, PowerCal, PowerInfo, PowerMate, PowerTool, rLAB, rPIC, Select Mode, SmartSensor, SmartShunt, SmartTel and Total Endurance are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

Serialized Quick Turn Programming (SQTP) is a service mark of Microchip Technology Incorporated in the U.S.A.

All other trademarks mentioned herein are property of their respective companies.

© 2003, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

 Printed on recycled paper.



Microchip received QS-9000 quality system certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona in July 1999 and Mountain View, California in March 2002. The Company's quality system processes and procedures are QS-9000 compliant for its PICmicro® 8-bit MCUs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, non-volatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001 certified.

PIC16C62X

NOTES:

4.0 MEMORY ORGANIZATION

4.1 Program Memory Organization

The PIC16C62X has a 13-bit program counter capable of addressing an 8K x 14 program memory space. Only the first 512 x 14 (0000h - 01FFh) for the PIC16C620(A) and PIC16CR620, 1K x 14 (0000h - 03FFh) for the PIC16C621(A) and 2K x 14 (0000h - 07FFh) for the PIC16C622(A) are physically implemented. Accessing a location above these boundaries will cause a wrap-around within the first 512 x 14 space (PIC16C(R)620(A)) or 1K x 14 space (PIC16C621(A)) or 2K x 14 space (PIC16C622(A)). The RESET vector is at 0000h and the interrupt vector is at 0004h (Figure 4-1, Figure 4-2, Figure 4-3).

FIGURE 4-1: PROGRAM MEMORY MAP AND STACK FOR THE PIC16C620/PIC16C620A/ PIC16CR620A

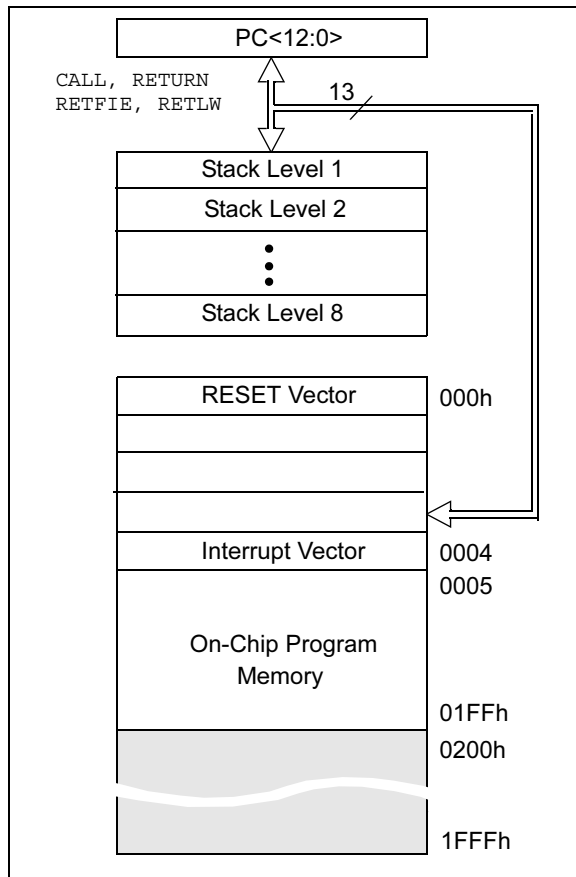


FIGURE 4-2: PROGRAM MEMORY MAP AND STACK FOR THE PIC16C621/PIC16C621A

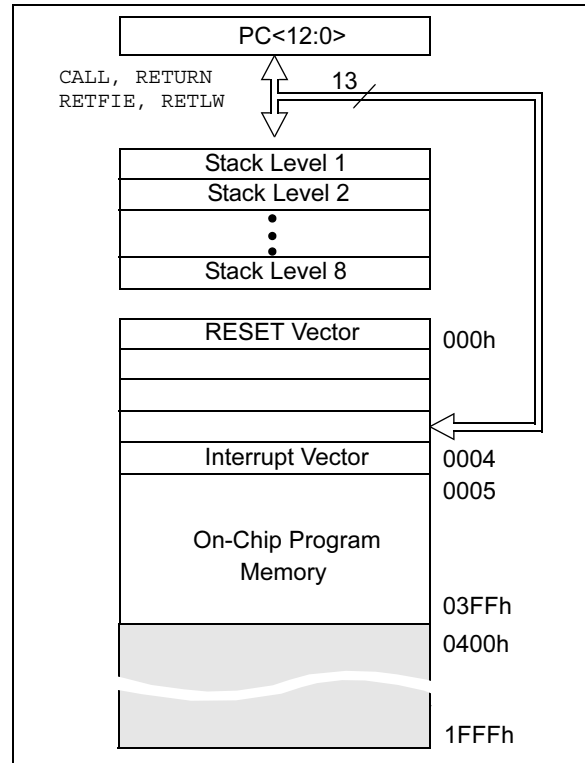


FIGURE 4-3: PROGRAM MEMORY MAP AND STACK FOR THE PIC16C622/PIC16C622A

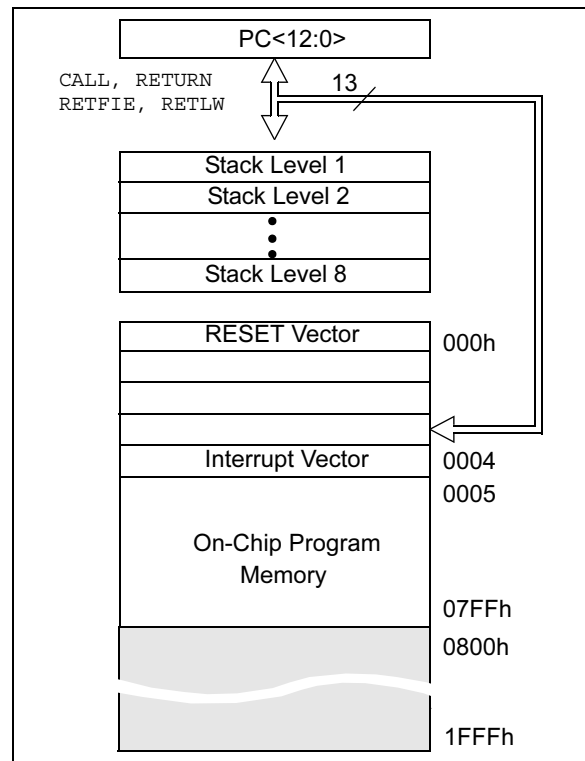


FIGURE 4-4: DATA MEMORY MAP FOR THE PIC16C620/621

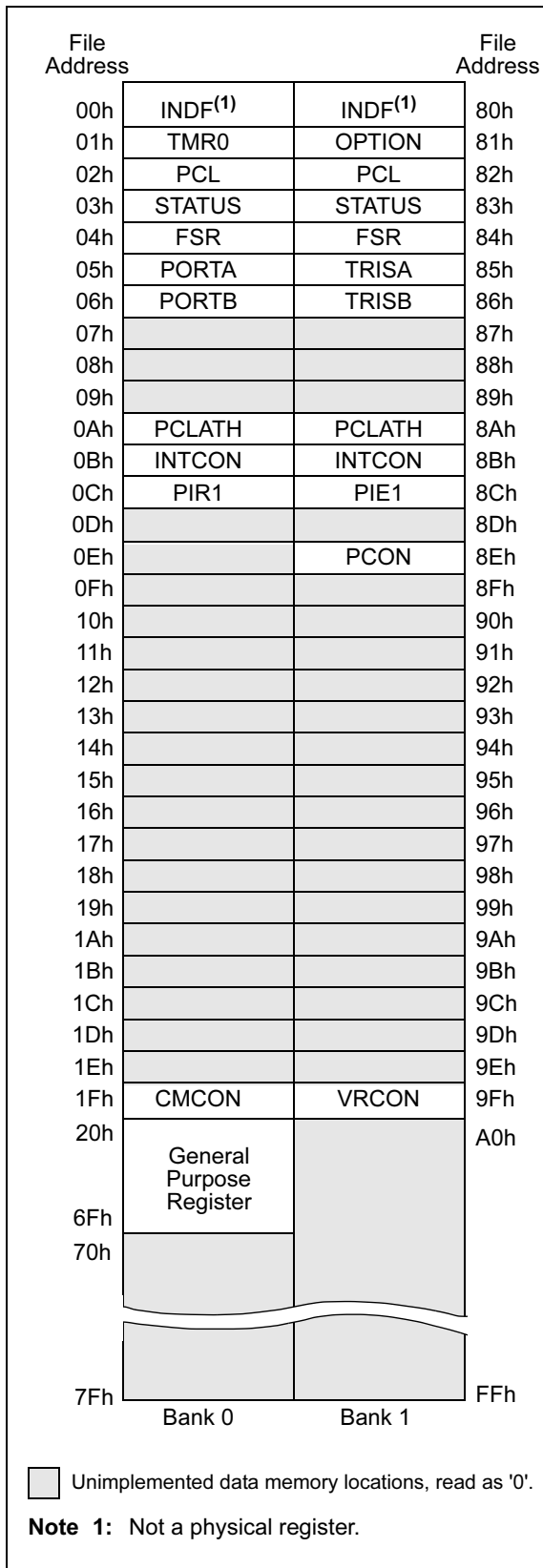
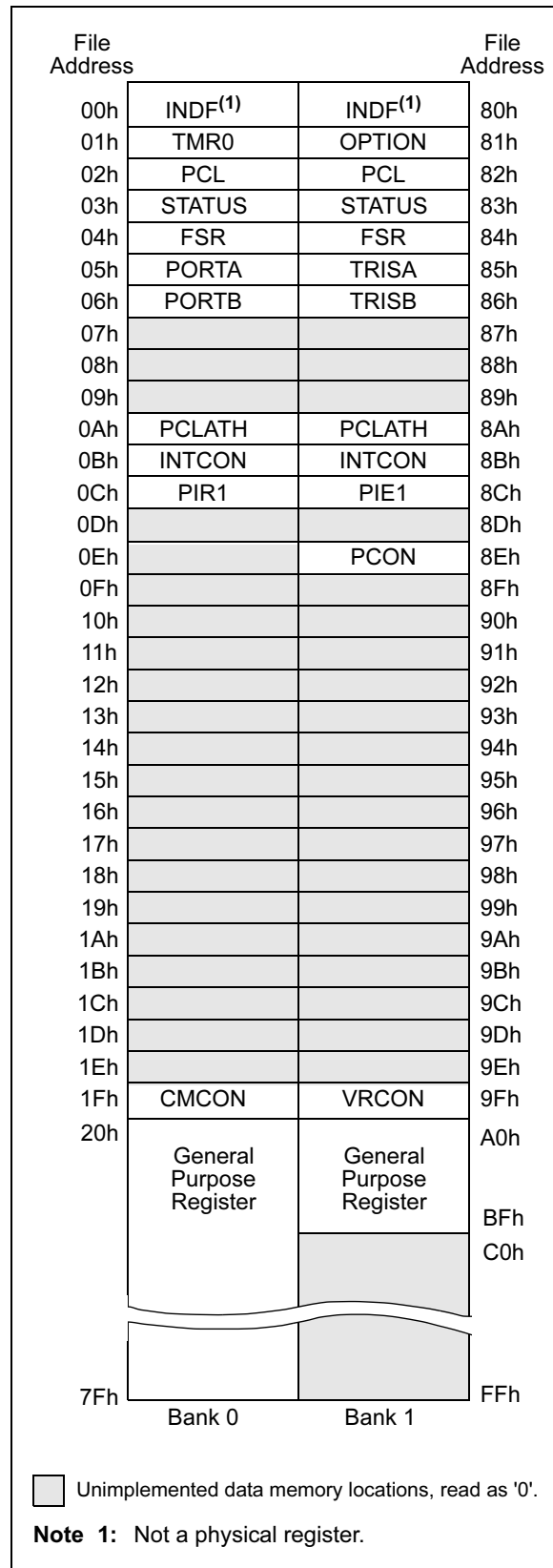


FIGURE 4-5: DATA MEMORY MAP FOR THE PIC16C622



4.2.2.2 OPTION Register

The OPTION register is a readable and writable register, which contains various control bits to configure the TMR0/WDT prescaler, the external RB0/INT interrupt, TMR0 and the weak pull-ups on PORTB.

Note: To achieve a 1:1 prescaler assignment for TMR0, assign the prescaler to the WDT (PSA = 1).

REGISTER 4-2: OPTION REGISTER (ADDRESS 81H)

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
bit 7							bit 0

- bit 7 **RBPU: PORTB Pull-up Enable bit**
 1 = PORTB pull-ups are disabled
 0 = PORTB pull-ups are enabled by individual port latch values
- bit 6 **INTEDG: Interrupt Edge Select bit**
 1 = Interrupt on rising edge of RB0/INT pin
 0 = Interrupt on falling edge of RB0/INT pin
- bit 5 **T0CS: TMR0 Clock Source Select bit**
 1 = Transition on RA4/T0CKI pin
 0 = Internal instruction cycle clock (CLKOUT)
- bit 4 **T0SE: TMR0 Source Edge Select bit**
 1 = Increment on high-to-low transition on RA4/T0CKI pin
 0 = Increment on low-to-high transition on RA4/T0CKI pin
- bit 3 **PSA: Prescaler Assignment bit**
 1 = Prescaler is assigned to the WDT
 0 = Prescaler is assigned to the Timer0 module
- bit 2-0 **PS<2:0>: Prescaler Rate Select bits**

Bit Value	TMR0 Rate	WDT Rate
000	1 : 2	1 : 1
001	1 : 4	1 : 2
010	1 : 8	1 : 4
011	1 : 16	1 : 8
100	1 : 32	1 : 16
101	1 : 64	1 : 32
110	1 : 128	1 : 64
111	1 : 256	1 : 128

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

4.2.2.4 PIE1 Register

This register contains the individual enable bit for the comparator interrupt.

REGISTER 4-4: PIE1 REGISTER (ADDRESS 8CH)

U-0	R/W-0	U-0	U-0	U-0	U-0	U-0	U-0
—	CMIE	—	—	—	—	—	—
bit 7							bit 0

- bit 7 **Unimplemented:** Read as '0'
- bit 6 **CMIE:** Comparator Interrupt Enable bit
1 = Enables the Comparator interrupt
0 = Disables the Comparator interrupt
- bit 5-0 **Unimplemented:** Read as '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

4.2.2.5 PIR1 Register

This register contains the individual flag bit for the comparator interrupt.

Note: Interrupt flag bits get set when an interrupt condition occurs, regardless of the state of its corresponding enable bit or the global enable bit, GIE (INTCON<7>). User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

REGISTER 4-5: PIR1 REGISTER (ADDRESS 0CH)

U-0	R/W-0	U-0	U-0	U-0	U-0	U-0	U-0
—	CMIF	—	—	—	—	—	—
bit 7							bit 0

- bit 7 **Unimplemented:** Read as '0'
- bit 6 **CMIF:** Comparator Interrupt Flag bit
1 = Comparator input has changed
0 = Comparator input has not changed
- bit 5-0 **Unimplemented:** Read as '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

PIC16C62X

9.1 Configuration Bits

The configuration bits can be programmed (read as '0') or left unprogrammed (read as '1') to select various device configurations. These bits are mapped in program memory location 2007h.

The user will note that address 2007h is beyond the user program memory space. In fact, it belongs to the special test/configuration memory space (2000h – 3FFFh), which can be accessed only during programming.

REGISTER 9-1: CONFIGURATION WORD (ADDRESS 2007h)

CP1	CP0 (2)	CP1	$\overline{\text{CP0}}^{(2)}$	$\overline{\text{CP1}}$	CP0 (2)		BODEN	CP1	CP0 (2)	$\overline{\text{PWRT}}^{\text{E}}$	WDTE	F0SC1	F0SC0
bit 13													bit 0

bit 13-8, 5-4: **CP<1:0>**: Code protection bit pairs (2)
 Code protection for 2K program memory
 11 = Program memory code protection off
 10 = 0400h-07FFh code protected
 01 = 0200h-07FFh code protected
 00 = 0000h-07FFh code protected

Code protection for 1K program memory
 11 = Program memory code protection off
 10 = Program memory code protection off
 01 = 0200h-03FFh code protected
 00 = 0000h-03FFh code protected

Code protection for 0.5K program memory
 11 = Program memory code protection off
 10 = Program memory code protection off
 01 = Program memory code protection off
 00 = 0000h-01FFh code protected

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

bit 6 **BODEN**: Brown-out Reset Enable bit (1)
 1 = BOR enabled
 0 = BOR disabled

bit 3 **PWRT^E**: Power-up Timer Enable bit (1, 3)
 1 = PWRT disabled
 0 = PWRT enabled

bit 2 **WDTE**: Watchdog Timer Enable bit
 1 = WDT enabled
 0 = WDT disabled

bit 1-0 **FOSC1:FOSC0**: Oscillator Selection bits
 11 = RC oscillator
 10 = HS oscillator
 01 = XT oscillator
 00 = LP oscillator

- Note 1:** Enabling Brown-out Reset automatically enables Power-up Timer (PWRT) regardless of the value of bit $\overline{\text{PWRT}}^{\text{E}}$. Ensure the Power-up Timer is enabled anytime Brown-out Detect Reset is enabled.
- 2:** All of the CP<1:0> pairs have to be given the same value to enable the code protection scheme listed.
- 3:** Unprogrammed parts default the Power-up Timer disabled.

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

PIC16C62X

9.2.3 EXTERNAL CRYSTAL OSCILLATOR CIRCUIT

Either a prepackaged oscillator can be used or a simple oscillator circuit with TTL gates can be built. Prepackaged oscillators provide a wide operating range and better stability. A well-designed crystal oscillator will provide good performance with TTL gates. Two types of crystal oscillator circuits can be used; one with series resonance or one with parallel resonance.

Figure 9-3 shows implementation of a parallel resonant oscillator circuit. The circuit is designed to use the fundamental frequency of the crystal. The 74AS04 inverter performs the 180° phase shift that a parallel oscillator requires. The 4.7 kΩ resistor provides the negative feedback for stability. The 10 kΩ potentiometers bias the 74AS04 in the linear region. This could be used for external oscillator designs.

FIGURE 9-3: EXTERNAL PARALLEL RESONANT CRYSTAL OSCILLATOR CIRCUIT

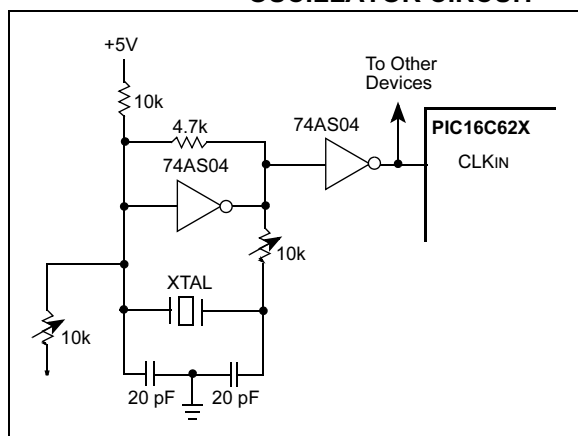
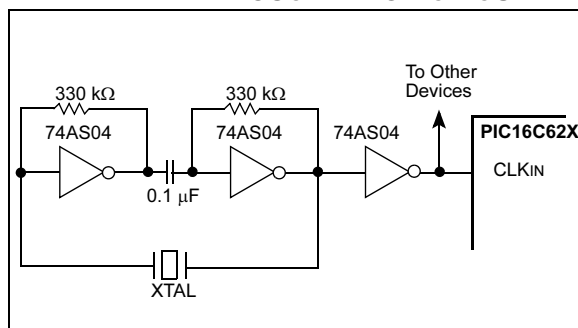


Figure 9-4 shows a series resonant oscillator circuit. This circuit is also designed to use the fundamental frequency of the crystal. The inverter performs a 180° phase shift in a series resonant oscillator circuit. The 330 kΩ resistors provide the negative feedback to bias the inverters in their linear region.

FIGURE 9-4: EXTERNAL SERIES RESONANT CRYSTAL OSCILLATOR CIRCUIT



9.2.4 RC OSCILLATOR

For timing insensitive applications the "RC" device option offers additional cost savings. The RC oscillator frequency is a function of the supply voltage, the resistor (R_{EXT}) and capacitor (C_{EXT}) values, and the operating temperature. In addition to this, the oscillator frequency will vary from unit to unit due to normal process parameter variation. Furthermore, the difference in lead frame capacitance between package types will also affect the oscillation frequency, especially for low C_{EXT} values. The user also needs to take into account variation due to tolerance of external R and C components used. Figure 9-5 shows how the R/C combination is connected to the PIC16C62X. For R_{EXT} values below 2.2 kΩ, the oscillator operation may become unstable or stop completely. For very high R_{EXT} values (e.g., 1 MΩ), the oscillator becomes sensitive to noise, humidity and leakage. Thus, we recommend to keep R_{EXT} between 3 kΩ and 100 kΩ.

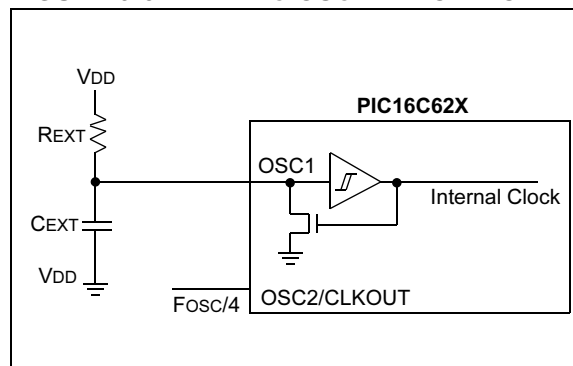
Although the oscillator will operate with no external capacitor (C_{EXT} = 0 pF), we recommend using values above 20 pF for noise and stability reasons. With no or small external capacitance, the oscillation frequency can vary dramatically due to changes in external capacitances, such as PCB trace capacitance or package lead frame capacitance.

See Section 13.0 for RC frequency variation from part to part due to normal process variation. The variation is larger for larger R (since leakage current variation will affect RC frequency more for large R) and for smaller C (since variation of input capacitance will affect RC frequency more).

See Section 13.0 for variation of oscillator frequency due to V_{DD} for given R_{EXT}/C_{EXT} values, as well as frequency variation due to operating temperature for given R, C and V_{DD} values.

The oscillator frequency, divided by 4, is available on the OSC2/CLKOUT pin, and can be used for test purposes or to synchronize other logic (Figure 3-2 for waveform).

FIGURE 9-5: RC OSCILLATOR MODE



9.3 RESET

The PIC16C62X differentiates between various kinds of RESET:

- Power-on Reset (POR)
- $\overline{\text{MCLR}}$ Reset during normal operation
- $\overline{\text{MCLR}}$ Reset during SLEEP
- WDT Reset (normal operation)
- WDT wake-up (SLEEP)
- Brown-out Reset (BOR)

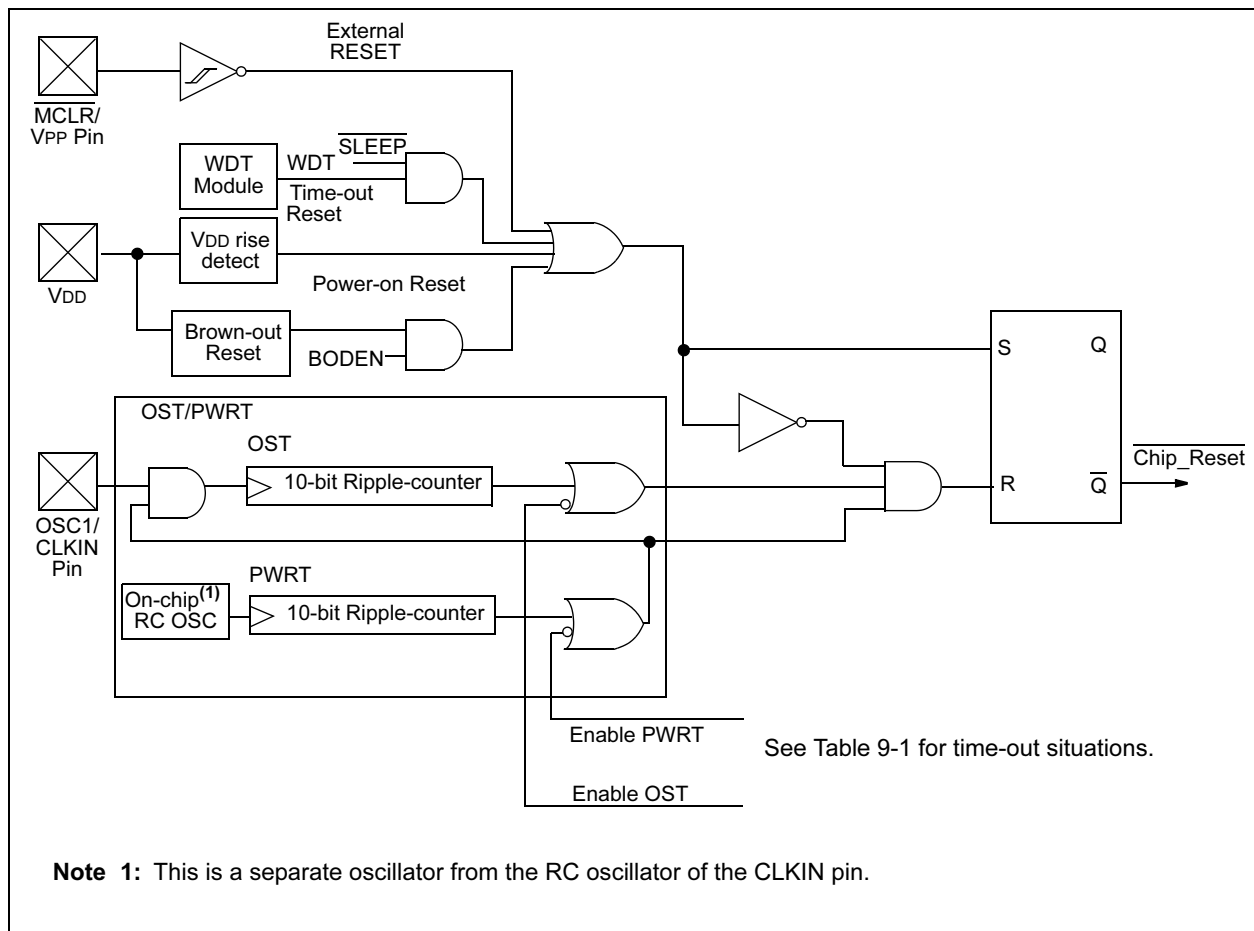
Some registers are not affected in any RESET condition. Their status is unknown on POR and unchanged in any other RESET. Most other registers are reset to a "RESET state" on Power-on Reset,

$\overline{\text{MCLR}}$ Reset, WDT Reset and $\overline{\text{MCLR}}$ Reset during SLEEP. They are not affected by a WDT wake-up, since this is viewed as the resumption of normal operation. $\overline{\text{TO}}$ and $\overline{\text{PD}}$ bits are set or cleared differently in different RESET situations as indicated in Table 9-2. These bits are used in software to determine the nature of the RESET. See Table 9-5 for a full description of RESET states of all registers.

A simplified block diagram of the on-chip RESET circuit is shown in Figure 9-6.

The $\overline{\text{MCLR}}$ Reset path has a noise filter to detect and ignore small pulses. See Table 12-5 for pulse width specification.

FIGURE 9-6: SIMPLIFIED BLOCK DIAGRAM OF ON-CHIP RESET CIRCUIT



PIC16C62X

9.9 Code Protection

If the code protection bit(s) have not been programmed, the on-chip program memory can be read out for verification purposes.

Note: Microchip does not recommend code protecting windowed devices.

9.10 ID Locations

Four memory locations (2000h-2003h) are designated as ID locations where the user can store checksum or other code identification numbers. These locations are not accessible during normal execution, but are readable and writable during Program/Verify. Only the Least Significant 4 bits of the ID locations are used.

9.11 In-Circuit Serial Programming™

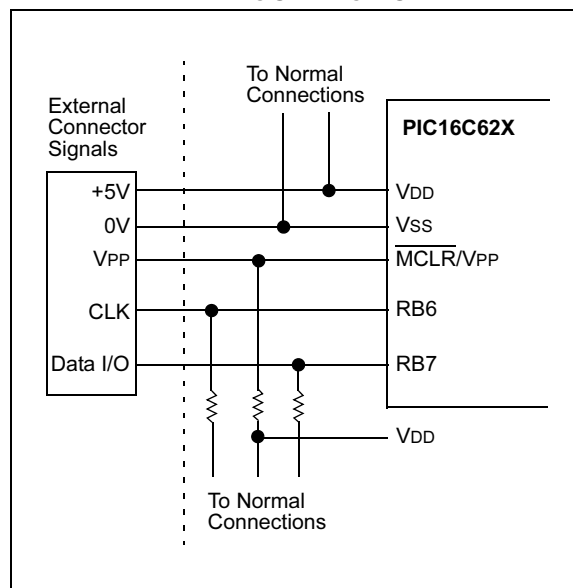
The PIC16C62X microcontrollers can be serially programmed while in the end application circuit. This is simply done with two lines for clock and data and three other lines for power, ground and the programming voltage. This allows customers to manufacture boards with unprogrammed devices and then program the microcontroller just before shipping the product. This also allows the most recent firmware or a custom firmware to be programmed.

The device is placed into a Program/Verify mode by holding the RB6 and RB7 pins low, while raising the MCLR (VPP) pin from VIL to VIH (see programming specification). RB6 becomes the programming clock and RB7 becomes the programming data. Both RB6 and RB7 are Schmitt Trigger inputs in this mode.

After RESET, to place the device into Programming/Verify mode, the program counter (PC) is at location 00h. A 6-bit command is then supplied to the device. Depending on the command, 14-bits of program data are then supplied to or from the device, depending if the command was a load or a read. For complete details of serial programming, please refer to the PIC16C6X/7X/9XX Programming Specification (DS30228).

A typical In-Circuit Serial Programming connection is shown in Figure 9-19.

FIGURE 9-19: TYPICAL IN-CIRCUIT SERIAL PROGRAMMING CONNECTION



PIC16C62X

TABLE 10-2: PIC16C62X INSTRUCTION SET

Mnemonic, Operands	Description	Cycles	14-Bit Opcode				Status Affected	Notes	
			MSb		LSb				
BYTE-ORIENTED FILE REGISTER OPERATIONS									
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
CLRW	-	Clear W	1	00	0001	0000	0011	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
BIT-ORIENTED FILE REGISTER OPERATIONS									
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
LITERAL AND CONTROL OPERATIONS									
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 PORTB, 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'.

- 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.
- If Program Counter (PC) is modified or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a NOP.

PIC16C62X

INCFSZ Increment f, Skip if 0

Syntax:	[<i>label</i>] INCFSZ f,d				
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$				
Operation:	$(f) + 1 \rightarrow (\text{dest})$, skip if result = 0				
Status Affected:	None				
Encoding:	<table border="1"><tr><td>00</td><td>1111</td><td>dfff</td><td>ffff</td></tr></table>	00	1111	dfff	ffff
00	1111	dfff	ffff		
Description:	<p>The contents of register 'f' are incremented. If 'd' is 0 the result is placed in the W register. If 'd' is 1, the result is placed back in register 'f'.</p> <p>If the result is 0, the next instruction, which is already fetched, is discarded. A NOP is executed instead making it a two-cycle instruction.</p>				
Words:	1				
Cycles:	1(2)				
Example	<pre>HERE INCFSZ CNT, 1 GOTO LOOP CONTINUE • • •</pre>				

Before Instruction
PC = address HERE

After Instruction
CNT = CNT + 1
if CNT= 0,
PC = address CONTINUE
if CNT≠ 0,
PC = address HERE +1

IORLW Inclusive OR Literal with W

Syntax:	[<i>label</i>] IORLW k			
Operands:	$0 \leq k \leq 255$			
Operation:	(W) .OR. k \rightarrow (W)			
Status Affected:	Z			
Encoding:	11	1000	kkkk	kkkk
Description:	The contents of the W register is OR'ed with the eight bit literal 'k'. The result is placed in the W register.			
Words:	1			
Cycles:	1			
Example	IORLW 0x35			
	Before Instruction			
	W = 0x9A			
	After Instruction			
	W = 0xBF			
	Z = 1			

IORWF Inclusive OR W with f

Syntax:	[<i>label</i>] IORWF f,d															
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$															
Operation:	(W) .OR. (f) \rightarrow (dest)															
Status Affected:	\bar{Z}															
Encoding:	<table border="1"><tr><td>00</td><td>0100</td><td>dfff</td><td>ffff</td></tr></table>	00	0100	dfff	ffff											
00	0100	dfff	ffff													
Description:	Inclusive OR the W register with register 'f'. If 'd' is 0 the result is placed in the W register. If 'd' is 1 the result is placed back in register 'f'.															
Words:	1															
Cycles:	1															
Example	<pre> IORWF RESULT, 0</pre> <p>Before Instruction</p> <table><tr><td>RESULT</td><td>=</td><td>0x13</td></tr><tr><td>W</td><td>=</td><td>0x91</td></tr></table> <p>After Instruction</p> <table><tr><td>RESULT</td><td>=</td><td>0x13</td></tr><tr><td>W</td><td>=</td><td>0x93</td></tr><tr><td>Z</td><td>=</td><td>1</td></tr></table>	RESULT	=	0x13	W	=	0x91	RESULT	=	0x13	W	=	0x93	Z	=	1
RESULT	=	0x13														
W	=	0x91														
RESULT	=	0x13														
W	=	0x93														
Z	=	1														

MOVLW Move Literal to W

Syntax:	[<i>label</i>] MOVLW k			
Operands:	$0 \leq k \leq 255$			
Operation:	$k \rightarrow (W)$			
Status Affected:	None			
Encoding:	11	00xx	kkkk	kkkk
Description:	The eight bit literal 'k' is loaded into W register. The don't cares will assemble as 0's.			
Words:	1			
Cycles:	1			
Example	MOVLW 0x5A			
	After Instruction			
	W = 0x5A			

MOVF		Move f						
Syntax:	[<i>label</i>] MOVF f,d							
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$							
Operation:	(f) \rightarrow (dest)							
Status Affected:	Z							
Encoding:	<table border="1"><tr><td>00</td><td>1000</td><td>dfff</td><td>ffff</td></tr></table>				00	1000	dfff	ffff
00	1000	dfff	ffff					
Description:	The contents of register f is moved to a destination dependent upon the status of d. If d = 0, destination is W register. If d = 1, the destination is file register f itself. d = 1 is useful to test a file register since status flag Z is affected.							
Words:	1							
Cycles:	1							
Example	MOVF FSR, 0							
	After Instruction							
	W = value in FSR							
	register							
	Z = 1							

MOVWF	Move W to f				
Syntax:	[<i>label</i>] MOVWF f				
Operands:	0 ≤ f ≤ 127				
Operation:	(W) → (f)				
Status Affected:	None				
Encoding:	<table><tr><td>00</td><td>0000</td><td>1fff</td><td>ffff</td></tr></table>	00	0000	1fff	ffff
00	0000	1fff	ffff		
Description:	Move data from W register to register 'f'.				
Words:	1				
Cycles:	1				
Example	MOVWF OPTION Before Instruction OPTION = 0xFF W = 0x4F After Instruction OPTION = 0x4F W = 0x4F				

NOP	No Operation			
Syntax:	[<i>label</i>] NOP			
Operands:	None			
Operation:	No operation			
Status Affected:	None			
Encoding:	00	0000	0xx0	0000
Description:	No operation.			
Words:	1			
Cycles:	1			
Example	NOP			

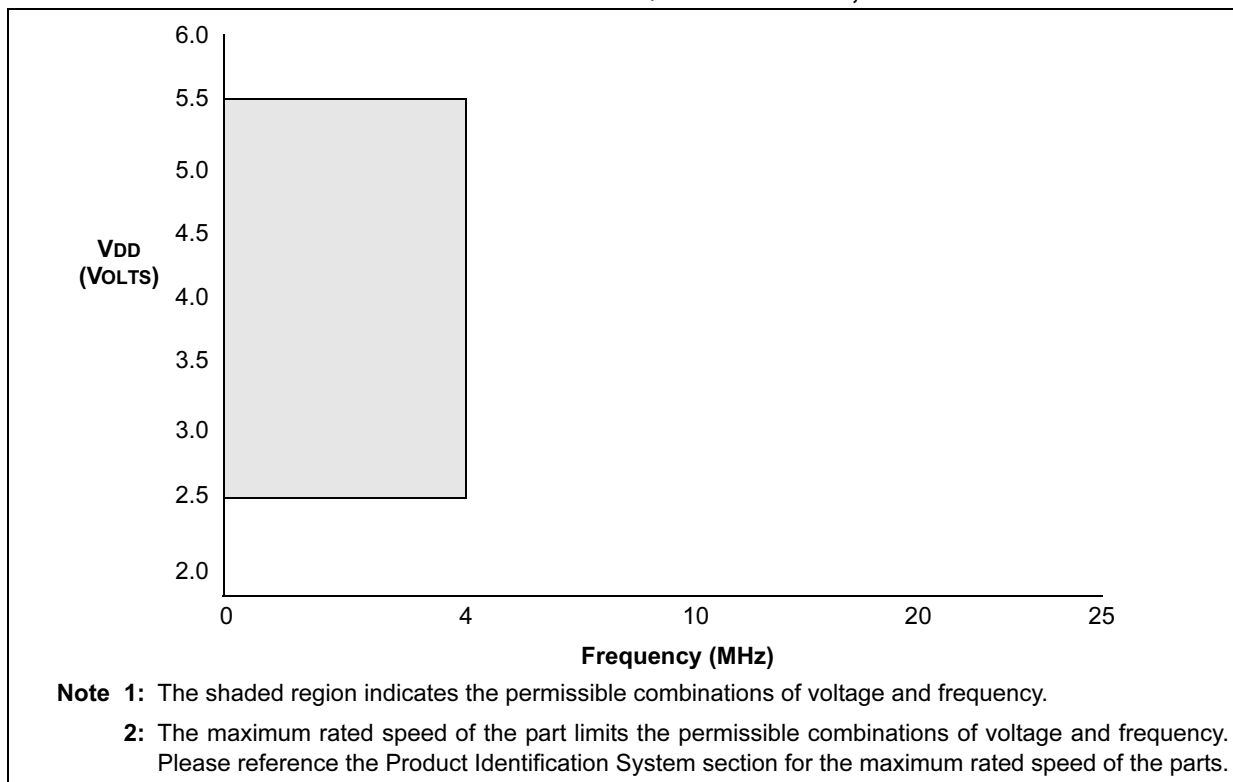
OPTION	Load Option Register				
Syntax:	[<i>label</i>] OPTION				
Operands:	None				
Operation:	(W) → OPTION				
Status Affected:	None				
Encoding:	<table><tr><td>00</td><td>0000</td><td>0110</td><td>0010</td></tr></table>	00	0000	0110	0010
00	0000	0110	0010		
Description:	<p>The contents of the W register are loaded in the OPTION register.</p> <p>This instruction is supported for code compatibility with PIC16C5X products. Since OPTION is a readable/writable register, the user can directly address it.</p>				
Words:	1				
Cycles:	1				
Example	<div>To maintain upward compatibility with future PICmicro[®] products, do not use this instruction.</div>				

PIC16C62X

NOTES:

PIC16C62X

FIGURE 12-9: PIC16LCR62XA VOLTAGE-FREQUENCY GRAPH, $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$



PIC16C62X

12.1 DC Characteristics: PIC16C62X-04 (Commercial, Industrial, Extended) PIC16C62X-20 (Commercial, Industrial, Extended) PIC16LC62X-04 (Commercial, Industrial, Extended)

PIC16C62X		Standard Operating Conditions (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for industrial and $0^{\circ}\text{C} \leq T_A \leq +70^{\circ}\text{C}$ for commercial and $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ for extended					
PIC16LC62X		Standard Operating Conditions (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for industrial and $0^{\circ}\text{C} \leq T_A \leq +70^{\circ}\text{C}$ for commercial and $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ for extended Operating voltage VDD range is the PIC16C62X range.					
Param. No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
D001	VDD	Supply Voltage	3.0	—	6.0	V	See Figures 12-1, 12-2, 12-3, 12-4, and 12-5
D001	VDD	Supply Voltage	2.5	—	6.0	V	See Figures 12-1, 12-2, 12-3, 12-4, and 12-5
D002	VDR	RAM Data Retention Voltage ⁽¹⁾	—	1.5*	—	V	Device in SLEEP mode
D002	VDR	RAM Data Retention Voltage ⁽¹⁾	—	1.5*	—	V	Device in SLEEP mode
D003	VPOR	VDD start voltage to ensure Power-on Reset	—	VSS	—	V	See section on Power-on Reset for details
D003	VPOR	VDD start voltage to ensure Power-on Reset	—	VSS	—	V	See section on Power-on Reset for details
D004	SVDD	VDD rise rate to ensure Power-on Reset	0.05*	—	—	V/ms	See section on Power-on Reset for details
D004	SVDD	VDD rise rate to ensure Power-on Reset	0.05*	—	—	V/ms	See section on Power-on Reset for details
D005	VBOR	Brown-out Detect Voltage	3.7	4.0	4.3	V	BOREN configuration bit is cleared
D005	VBOR	Brown-out Detect Voltage	3.7	4.0	4.3	V	BOREN configuration bit is cleared
D010	IDD	Supply Current ⁽²⁾	—	1.8	3.3	mA	FOSC = 4 MHz, VDD = 5.5V, WDT disabled, XT mode, (Note 4)*
			—	35	70	μA	FOSC = 32 kHz, VDD = 4.0V, WDT disabled, LP mode
			—	9.0	20	mA	FOSC = 20 MHz, VDD = 5.5V, WDT disabled, HS mode
D010	IDD	Supply Current ⁽²⁾	—	1.4	2.5	mA	FOSC = 2.0 MHz, VDD = 3.0V, WDT disabled, XT mode, (Note 4)
			—	26	53	μA	FOSC = 32 kHz, VDD = 3.0V, WDT disabled, LP mode
D020	IPD	Power-down Current ⁽³⁾	—	1.0	2.5	μA	VDD=4.0V, WDT disabled
D020	IPD	Power-down Current ⁽³⁾	—	0.7	2	μA	VDD=3.0V, WDT disabled

* 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: This is the limit to which VDD can be lowered without losing RAM data.

Note 2: The supply current is mainly a function of the operating voltage and frequency. Other factors such as I/O pin loading and switching rate, oscillator type, internal code execution pattern, and temperature also have an impact on the current consumption.

The test conditions for all IDD measurements in Active Operation mode are:

OSC1 = external square wave, from rail to rail; all I/O pins tri-stated, pulled to VDD,

MCLR = VDD; WDT enabled/disabled as specified.

Note 3: The power-down current in SLEEP mode does not depend on the oscillator type. Power-down current is measured with the part in SLEEP mode, with all I/O pins in hi-impedance state and tied to VDD or VSS.

Note 4: For RC osc configuration, current through REXT is not included. The current through the resistor can be estimated by the formula: $I_r = V_{DD}/2R_{EXT}$ (mA) with REXT in kΩ.

Note 5: The Δ current is the additional current consumed when this peripheral is enabled. This current should be added to the base IDD or IPD measurement.

PIC16C62X

12.4 DC Characteristics: PIC16C62X/C62XA/CR62XA (Commercial, Industrial, Extended) PIC16LC62X/LC62XA/LCR62XA (Commercial, Industrial, Extended)

PIC16C62X/C62XA/CR62XA			Standard Operating Conditions (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for industrial and $0^{\circ}\text{C} \leq T_A \leq +70^{\circ}\text{C}$ for commercial and $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ for extended				
PIC16LC62X/LC62XA/LCR62XA			Standard Operating Conditions (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for industrial and $0^{\circ}\text{C} \leq T_A \leq +70^{\circ}\text{C}$ for commercial and $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ for extended				
Param. No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
D030 D031 D032 D033	V _{IL}	Input Low Voltage I/O ports					V _{DD} = 4.5V to 5.5V otherwise (Note 1)
		with TTL buffer	V _{SS}	—	0.8V 0.15 V _{DD}	V	
		with Schmitt Trigger input	V _{SS}	—	0.2 V _{DD}	V	
		$\overline{\text{MCLR}}$, RA4/T0CKI, OSC1 (in RC mode)	V _{SS}	—	0.2 V _{DD}	V	
		OSC1 (in XT and HS) OSC1 (in LP)	V _{SS} V _{SS}	— —	0.3 V _{DD} 0.6 V _{DD} - 1.0	V V	
D030 D031 D032 D033	V _{IL}	Input Low Voltage I/O ports					V _{DD} = 4.5V to 5.5V otherwise (Note 1)
		with TTL buffer	V _{SS}	—	0.8V 0.15 V _{DD}	V	
		with Schmitt Trigger input	V _{SS}	—	0.2 V _{DD}	V	
		$\overline{\text{MCLR}}$, RA4/T0CKI, OSC1 (in RC mode)	V _{SS}	—	0.2 V _{DD}	V	
		OSC1 (in XT and HS) OSC1 (in LP)	V _{SS} V _{SS}	— —	0.3 V _{DD} 0.6 V _{DD} - 1.0	V V	
D040 D041 D042 D043 D043A	V _{IH}	Input High Voltage I/O ports					V _{DD} = 4.5V to 5.5V otherwise (Note 1)
		with TTL buffer	2.0V 0.25 V _{DD} + 0.8V	—	V _{DD} V _{DD}	V	
		with Schmitt Trigger input	0.8 V _{DD}	—	V _{DD}	V	
		$\overline{\text{MCLR}}$ RA4/T0CKI	0.8 V _{DD}	—	V _{DD}	V	
		OSC1 (XT, HS and LP) OSC1 (in RC mode)	0.7 V _{DD} 0.9 V _{DD}	— —	V _{DD} V _{DD}	V V	

* 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:** In RC oscillator configuration, the OSC1 pin is a Schmitt Trigger input. It is not recommended that the PIC16C62X(A) be driven with external clock in RC mode.
- 2:** The leakage current on the $\overline{\text{MCLR}}$ pin is strongly dependent on applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.
- 3:** Negative current is defined as coming out of the pin.

PIC16C62X

12.9 Timing Diagrams and Specifications

FIGURE 12-12: EXTERNAL CLOCK TIMING

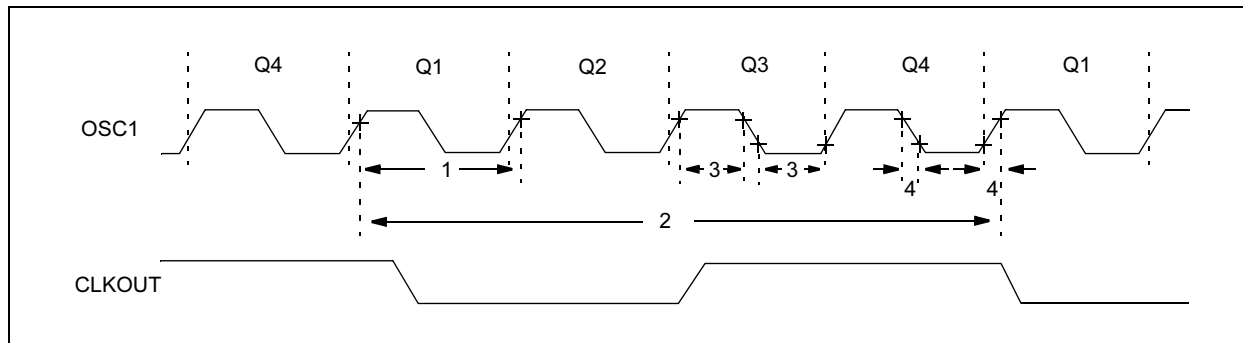


TABLE 12-3: EXTERNAL CLOCK TIMING REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
1A	Fosc	External CLKIN Frequency ⁽¹⁾	DC	—	4	MHz	XT and RC Osc mode, VDD=5.0V
			DC	—	20	MHz	HS Osc mode
			DC	—	200	kHz	LP Osc mode
		Oscillator Frequency ⁽¹⁾	DC	—	4	MHz	RC Osc mode, VDD=5.0V
			0.1	—	4	MHz	XT Osc mode
1	Tosc	External CLKIN Period ⁽¹⁾	1	—	20	MHz	HS Osc mode
			DC	—	200	kHz	LP Osc mode
			250	—	—	ns	XT and RC Osc mode
		Oscillator Period ⁽¹⁾	50	—	—	ns	HS Osc mode
			5	—	—	μs	LP Osc mode
2	Tcy	Instruction Cycle Time ⁽¹⁾	1.0	Fosc/4	DC	μs	Tcys=Fosc/4
3*	TosL, TosH	External Clock in (OSC1) High or Low Time	100*	—	—	ns	XT oscillator, TosC L/H duty cycle
			2*	—	—	μs	LP oscillator, TosC L/H duty cycle
			20*	—	—	ns	HS oscillator, TosC L/H duty cycle
4*	TosR, TosF	External Clock in (OSC1) Rise or Fall Time	25*	—	—	ns	XT oscillator
			50*	—	—	ns	LP oscillator
			15*	—	—	ns	HS oscillator

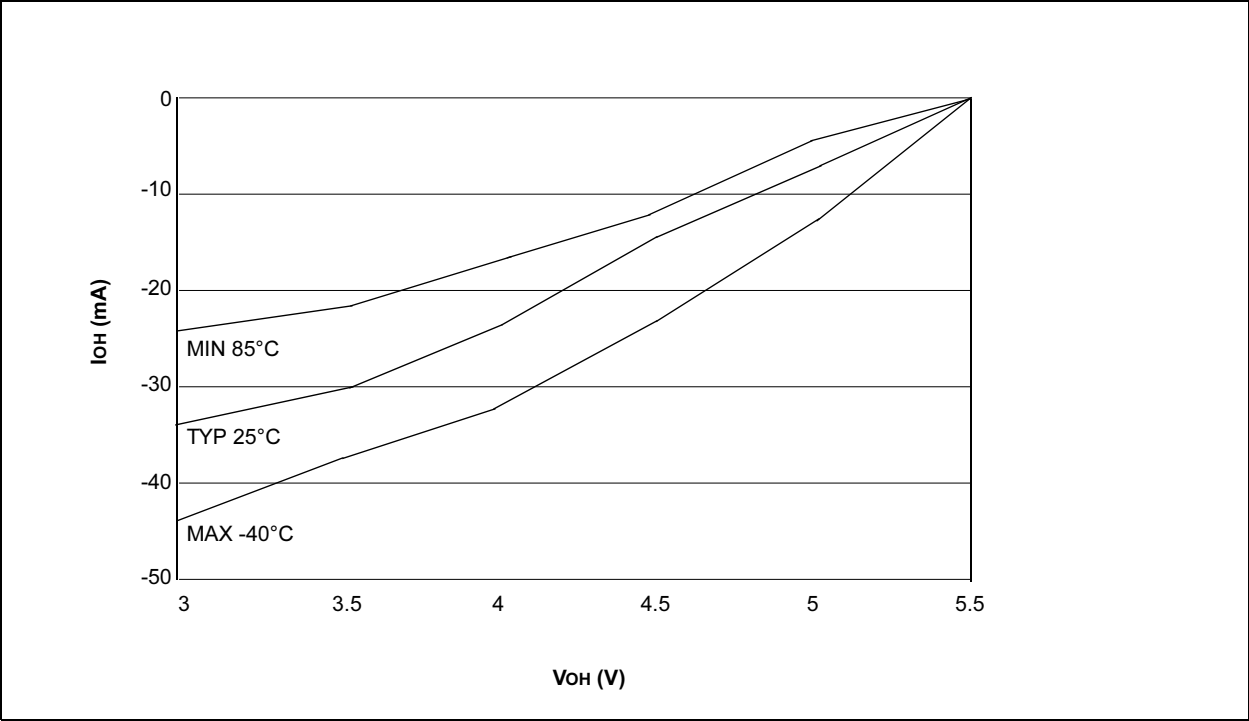
2: * These parameters are characterized but not tested.

3: † 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: 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 the OSC1 pin. When an external clock input is used, the "Max." cycle time limit is "DC" (no clock) for all devices.

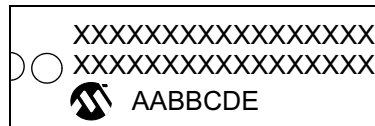
PIC16C62X

FIGURE 13-7: I_{OH} vs. V_{OH} , $V_{DD} = 5.5V$

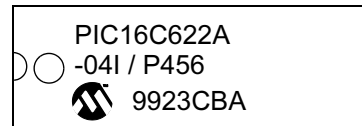


14.1 Package Marking Information

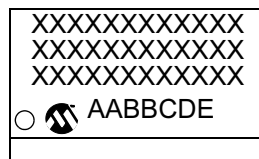
18-Lead PDIP



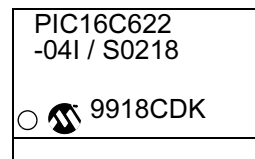
Example



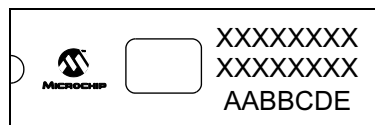
18-Lead SOIC (.300")



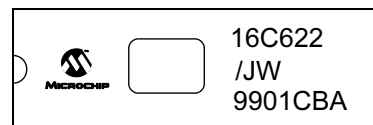
Example



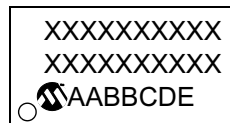
18-Lead Cerdip Windowed



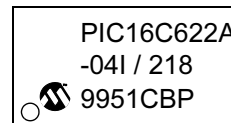
Example



20-Lead SSOP



Example



Legend: XX...X Customer specific information*
 Y Year code (last digit of calendar year)
 YY Year code (last 2 digits of calendar year)
 WW Week code (week of January 1 is week '01')
 NNN Alphanumeric traceability code

Note: In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line thus limiting the number of available characters for customer specific information.

* Standard PICmicro device marking consists of Microchip part number, year code, week code, and traceability code. For PICmicro device marking beyond this, certain price adders apply. Please check with your Microchip Sales Office. For QTP devices, any special marking adders are included in QTP price.