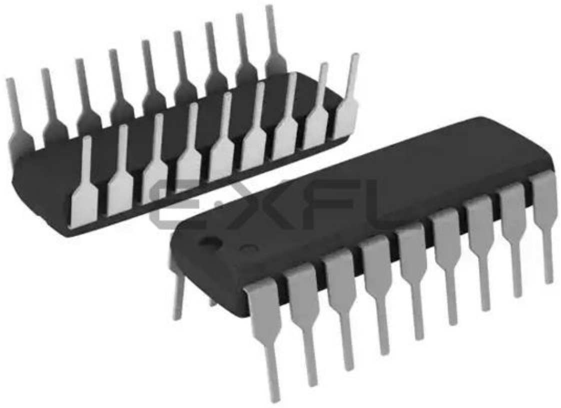


Welcome to E-XFL.COM

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	4MHz
Connectivity	UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	16
Program Memory Size	3.5KB (2K x 14)
Program Memory Type	FLASH
EEPROM Size	128 x 8
RAM Size	224 x 8
Voltage - Supply (Vcc/Vdd)	2V ~ 5.5V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°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/pic16lf628-04i-p

FIGURE 3-2: DATA MEMORY MAP OF THE PIC16F627 AND PIC16F628

File Address	Bank 0	File Address	Bank 1	File Address	Bank 2	File Address	Bank 3
00h	Indirect addr.(1)	80h	Indirect addr.(1)	100h	Indirect addr.(1)	180h	Indirect addr.(1)
01h	TMR0	81h	OPTION	101h	TMR0	181h	OPTION
02h	PCL	82h	PCL	102h	PCL	182h	PCL
03h	STATUS	83h	STATUS	103h	STATUS	183h	STATUS
04h	FSR	84h	FSR	104h	FSR	184h	FSR
05h	PORTA	85h	TRISA	105h		185h	
06h	PORTB	86h	TRISB	106h	PORTB	186h	TRISB
07h		87h		107h		187h	
08h		88h		108h		188h	
09h		89h		109h		189h	
0Ah	PCLATH	8Ah	PCLATH	10Ah	PCLATH	18Ah	PCLATH
0Bh	INTCON	8Bh	INTCON	10Bh	INTCON	18Bh	INTCON
0Ch	PIR1	8Ch	PIE1	10Ch		18Ch	
0Dh		8Dh		10Dh		18Dh	
0Eh	TMR1L	8Eh	PCON	10Eh		18Eh	
0Fh	TMR1H	8Fh		10Fh		18Fh	
10h	T1CON	90h					
11h	TMR2	91h					
12h	T2CON	92h	PR2				
13h		93h					
14h		94h					
15h	CCPR1L	95h					
16h	CCPR1H	96h					
17h	CCP1CON	97h					
18h	RCSTA	98h	TXSTA				
19h	TXREG	99h	SPBRG				
1Ah	RCREG	9Ah	EEDATA				
1Bh		9Bh	EEADR				
1Ch		9Ch	EECON1				
1Dh		9Dh	EECON2(1)				
1Eh		9Eh					
1Fh	CMCON	9Fh	VRCON	11Fh			
20h–6Fh	General Purpose Register 80 Bytes	A0h–EFh	General Purpose Register 80 Bytes	120h–14Fh	General Purpose Register 48 Bytes		
				150h–16Fh		1EFh	
70h–7Fh	16 Bytes	F0h–FFh	accesses 70h-7Fh	170h–17Fh	accesses 70h-7Fh	1F0h–1FFh	accesses 70h - 7Fh

■ Unimplemented data memory locations, read as '0'.

Note 1: Not a physical register.

TABLE 3-2: SPECIAL FUNCTION REGISTERS SUMMARY BANK 1

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR Reset(1)	Details on Page
Bank 1											
80h	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								xxxx xxxx	25
81h	OPTION	$\overline{\text{RBPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	20
82h	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	25
83h	STATUS	IRP	RP1	RP0	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C	0001 1xxx	19
84h	FSR	Indirect data memory address pointer								xxxx xxxx	25
85h	TRISA	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	1111 1111	29
86h	TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	1111 1111	34
87h	—	Unimplemented								—	—
88h	—	Unimplemented								—	—
89h	—	Unimplemented								—	—
8Ah	PCLATH	—	—	—	Write buffer for upper 5 bits of program counter					---0 0000	25
8Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	21
8Ch	PIE1	EEIE	CMIE	RCIE	TXIE	—	CCP1IE	TMR2IE	TMR1IE	0000 -000	22
8Dh	—	Unimplemented								—	—
8Eh	PCON	—	—	—	—	OSCF	—	$\overline{\text{POR}}$	$\overline{\text{BOD}}$	---- 1-0x	24
8Fh	—	Unimplemented								—	—
90h	—	Unimplemented								—	—
91h	—	Unimplemented								—	—
92h	PR2	Timer2 Period Register								1111 1111	50
93h	—	Unimplemented								—	—
94h	—	Unimplemented								—	—
95h	—	Unimplemented								—	—
96h	—	Unimplemented								—	—
97h	—	Unimplemented								—	—
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	69
99h	SPBRG	Baud Rate Generator Register								0000 0000	69
9Ah	EEDATA	EEPROM data register								xxxx xxxx	87
9Bh	EEADR	—	EEPROM address register							xxxx xxxx	87
9Ch	EECON1	—	—	—	—	WRERR	WREN	WR	RD	---- x000	87
9Dh	EECON2	EEPROM control register 2 (not a physical register)								--------	87
9Eh	—	Unimplemented								—	—
9Fh	VRCON	VREN	VROE	VRR	—	VR3	VR2	VR1	VR0	000- 0000	59

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

Note 1: For the Initialization Condition for Registers Tables, refer to Table 14-7 and Table 14-8 on page 98.

TABLE 3-4: SPECIAL FUNCTION REGISTERS SUMMARY BANK 3

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR Reset(1)	Details on Page
Bank 3											
180h	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								xxxx xxxx	25
181h	OPTION	$\overline{RBPU}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	20
182h	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	25
183h	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	19
184h	FSR	Indirect data memory address pointer								xxxx xxxx	25
185h	—	Unimplemented								—	—
186h	TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	1111 1111	34
187h	—	Unimplemented								—	—
188h	—	Unimplemented								—	—
189h	—	Unimplemented								—	—
18Ah	PCLATH	—	—	—	Write buffer for upper 5 bits of program counter					---0 0000	25
18Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	21
18Ch	—	Unimplemented								—	—
18Dh	—	Unimplemented								—	—
18Eh	—	Unimplemented								—	—
18Fh	—	Unimplemented								—	—
190h	—	Unimplemented								—	—
191h	—	Unimplemented								—	—
192h	—	Unimplemented								—	—
193h	—	Unimplemented								—	—
194h	—	Unimplemented								—	—
195h	—	Unimplemented								—	—
196h	—	Unimplemented								—	—
197h	—	Unimplemented								—	—
198h	—	Unimplemented								—	—
199h	—	Unimplemented								—	—
19Ah	—	Unimplemented								—	—
19Bh	—	Unimplemented								—	—
19Ch	—	Unimplemented								—	—
19Dh	—	Unimplemented								—	—
19Eh	—	Unimplemented								—	—
19Fh	—	Unimplemented								—	—

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

Note 1: For the Initialization Condition for Registers Tables, refer to Table 14-7 and Table 14-8 on page 98.

3.2.2.3 INTCON Register

The INTCON register is a readable and writable register which contains the various enable and flag bits for all interrupt sources except the comparator module. See Section 3.2.2.4 and Section 3.2.2.5 for a description of the comparator enable and flag bits.

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>).

REGISTER 3-3: INTCON REGISTER (ADDRESS: 0Bh, 8Bh, 10Bh, 18Bh)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF
bit 7							bit 0

bit 7 **GIE:** Global Interrupt Enable bit
1 = Enables all unmasked interrupts
0 = Disables all interrupts

bit 6 **PEIE**: Peripheral Interrupt Enable bit
1 = Enables all unmasked peripheral interrupts
0 = Disables all peripheral interrupts

bit 5 **T0IE**: TMR0 Overflow Interrupt Enable bit
1 = Enables the TMR0 interrupt
0 = Disables the TMR0 interrupt

bit 4 **INTE**: RB0/INT External Interrupt Enable bit
1 = Enables the RB0/INT external interrupt
0 = Disables the RB0/INT external interrupt

bit 3 **RBIE**: RB Port Change Interrupt Enable bit
1 = Enables the RB port change interrupt
0 = Disables the RB port change interrupt

bit 2 **T0IF**: TMR0 Overflow Interrupt Flag bit
1 = TMR0 register has overflowed (must be cleared in software)
0 = TMR0 register did not overflow

bit 1 **INTF**: RB0/INT External Interrupt Flag bit
1 = The RB0/INT external interrupt occurred (must be cleared in software)
0 = The RB0/INT external interrupt did not occur

bit 0 **RBIF**: RB Port Change Interrupt Flag bit
1 = When at least one of the RB7:RB4 pins changed state (must be cleared in software)
0 = None of the RB7:RB4 pins have changed state

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

4.0 GENERAL DESCRIPTION

The PIC16F62X are 18-Pin FLASH-based members of the versatile PIC16CXX family of low cost, high performance, CMOS, fully static, 8-bit microcontrollers.

All PICmicro® microcontrollers employ an advanced RISC architecture. The PIC16F62X have enhanced core features, eight-level deep stack, and multiple internal and external interrupt sources. The separate instruction and data buses of the Harvard architecture allow a 14-bit wide instruction word with the separate 8-bit wide data. The two-stage instruction pipeline allows all instructions to execute in a single cycle, except for program branches (which require two cycles). A total of 35 instructions (reduced instruction set) are available. Additionally, a large register set gives some of the architectural innovations used to achieve a very high performance.

PIC16F62X microcontrollers typically achieve a 2:1 code compression and a 4:1 speed improvement over other 8-bit microcontrollers in their class.

PIC16F62X devices have special features to reduce external components, thus reducing system cost, enhancing system reliability and reducing power consumption.

The PIC16F62X has eight oscillator configurations. The single pin ER oscillator provides a low cost solution. The LP oscillator minimizes power consumption, XT is a standard crystal, INTRC is a self-contained internal oscillator. The HS is for High Speed crystals. The EC mode is for an external clock source.

The SLEEP (Power-down) mode offers power savings. The user can wake-up the chip from SLEEP through several external interrupts, internal interrupts, and RESETS.

A highly reliable Watchdog Timer with its own on-chip RC oscillator provides protection against software lock-up.

Table 4-1 shows the features of the PIC16F62X mid-range microcontroller families.

A simplified block diagram of the PIC16F62X is shown in Figure 2.1.

The PIC16F62X series fits in applications ranging from battery chargers to low power remote sensors. The FLASH technology makes customization of application programs (detection levels, pulse generation, timers, etc.) extremely fast and convenient. The small footprint packages make this microcontroller series ideal for all applications with space limitations. Low cost, low power, high performance, ease of use and I/O flexibility make the PIC16F62X very versatile.

4.1 Development Support

The PIC16F62X family is supported by a full featured macro assembler, a software simulator, an in-circuit emulator, a low cost development programmer and a full-featured programmer. A Third Party “C” compiler support tool is also available.

TABLE 4-1: PIC16F62X FAMILY OF DEVICES

		PIC16F627	PIC16F628	PIC16LF627	PIC16LF628
Clock	Maximum Frequency of Operation (MHz)	20	20	4	4
Memory	FLASH Program Memory (words)	1024	2048	1024	2048
	RAM Data Memory (bytes)	224	224	224	224
	EEPROM Data Memory (bytes)	128	128	128	128
Peripherals	Timer Module(s)	TMR0, TMR1, TMR2	TMR0, TMR1, TMR2	TMR0, TMR1, TMR2	TMR0, TMR1, TMR2
	Comparator(s)	2	2	2	2
	Capture/Compare/PWM modules	1	1	1	1
	Serial Communications	USART	USART	USART	USART
	Internal Voltage Reference	Yes	Yes	Yes	Yes
Features	Interrupt Sources	10	10	10	10
	I/O Pins	16	16	16	16
	Voltage Range (Volts)	3.0-5.5	3.0-5.5	2.0-5.5	2.0-5.5
	Brown-out Detect	Yes	Yes	Yes	Yes
	Packages	18-pin DIP, SOIC, 20-pin SSOP	18-pin DIP, SOIC, 20-pin SSOP	18-pin DIP, SOIC, 20-pin SSOP	18-pin DIP, SOIC, 20-pin SSOP

All PICmicro® Family devices have Power-on Reset, selectable Watchdog Timer, selectable code protect and high I/O current capability. All PIC16F62X Family devices use serial programming with clock pin RB6 and data pin RB7.

TABLE 5-1: PORTA FUNCTIONS

Name	Function	Input Type	Output Type	Description
RA0/AN0	RA0	ST	CMOS	Bi-directional I/O port
	AN0	AN	—	Analog comparator input
RA1/AN1	RA1	ST	CMOS	Bi-directional I/O port
	AN1	AN	—	Analog comparator input
RA2/AN2/VREF	RA2	ST	CMOS	Bi-directional I/O port
	AN2	AN	—	Analog comparator input
	VREF	—	AN	VREF output
RA3/AN3/CMP1	RA3	ST	CMOS	Bi-directional I/O port
	AN3	AN	—	Analog comparator input
	CMP1	—	CMOS	Comparator 1 output
RA4/T0CKI/CMP2	RA4	ST	OD	Bi-directional I/O port
	T0CKI	ST	—	External clock input for TMR0 or comparator output. Output is open drain type
	CMP2	—	OD	Comparator 2 output
RA5/$\overline{\text{MCLR}}$/VPP	RA5	ST	—	Input port
	$\overline{\text{MCLR}}$	ST	—	Master clear
	VPP	HV	—	Programming voltage input. When configured as $\overline{\text{MCLR}}$, this pin is an active low RESET to the device. Voltage on $\overline{\text{MCLR}}$/VPP must not exceed VDD during normal device operation
RA6/OSC2/CLKOUT	RA6	ST	CMOS	Bi-directional I/O port.
	OSC2	XTAL	—	Oscillator crystal output. Connects to crystal resonator in Crystal Oscillator mode.
	CLKOUT	—	CMOS	In ER/INTRC mode, OSC2 pin can output CLKOUT, which has 1/4 the frequency of OSC1
RA7/OSC1/CLKIN	RA7	ST	CMOS	Bi-directional I/O port
	OSC1	XTAL	—	Oscillator crystal input
	CLKIN	ST	—	External clock source input. ER biasing pin.

Legend: ST = Schmitt Trigger input HV = High Voltage OD = Open Drain AN = Analog

5.3 I/O Programming Considerations

5.3.1 BI-DIRECTIONAL I/O PORTS

Any instruction which writes, operates internally as a read followed by a write operation. The `BCF` and `BSF` instructions, for example, read the register into the CPU, execute the bit operation and write the result back to the register. Caution must be used when these instructions are applied to a port with both inputs and outputs defined. For example, a `BSF` operation on Bit 5 of PORTB will cause all eight bits of PORTB to be read into the CPU. Then the `BSF` operation takes place on Bit 5 and PORTB is written to the output latches. If another bit of PORTB is used as a bi-directional I/O pin (e.g., Bit 0) and it is defined as an input at this time, the input signal present on the pin itself would be read into the CPU and rewritten to the data latch of this particular pin, overwriting the previous content. As long as the pin stays in the Input mode, no problem occurs. However, if Bit 0 is switched into Output mode later on, the content of the data latch may now be unknown.

Reading a port register, reads the values of the port pins. Writing to the port register writes the value to the port latch. When using read-modify-write instructions (ex. `BCF, BSF`, etc.) on a port, the value of the port pins is read, the desired operation is done to this value, and this value is then written to the port latch.

Example 5-2 shows the effect of two sequential read-modify-write instructions (ex., `BCF, BSF`, etc.) on an I/O port.

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

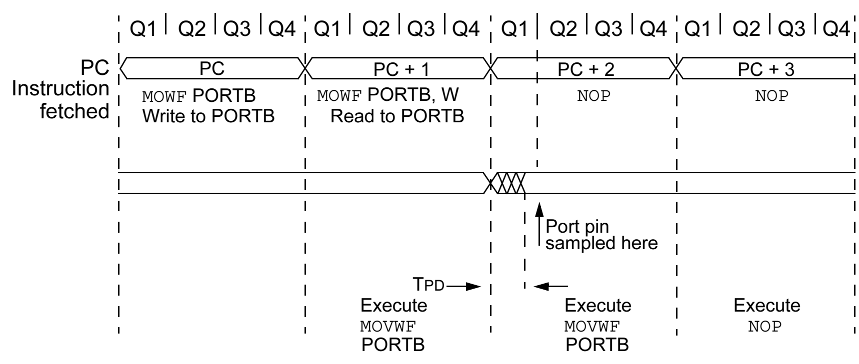
EXAMPLE 5-2: READ-MODIFY-WRITE INSTRUCTIONS ON AN I/O PORT

```
;Initial PORT settings:PORTB<7:4> Inputs
;
;                      PORTB<3:0> Outputs
;PORTB<7:6> have external pull-up and are not
;connected to other circuitry
;
;                      PORT latchPORT Pins
;                      ---------- ----------
    BCF STATUS, RP0    ;
    BCF PORTB, 7       ;01pp pppp 11pp pppp
    BSF STATUS, RP0    ;
    BCF TRISB, 7       ;10pp pppp 11pp pppp
    BCF TRISB, 6       ;10pp pppp 10pp pppp
;
;Note that the user may have expected the pin
;values to be 00pp pppp. The 2nd BCF caused
;RB7 to be latched as the pin value (High).
```

5.3.2 SUCCESSIVE OPERATIONS ON I/O PORTS

The actual write to an I/O port happens at the end of an instruction cycle, whereas for reading, the data must be valid at the beginning of the instruction cycle (Figure 5-16). Therefore, care must be exercised if a write followed by a read operation is carried out on the same I/O port. The sequence of instructions should be such to allow the pin voltage to stabilize (load dependent) before the next instruction which causes that file to be read into the CPU is executed. Otherwise, the previous state of that pin may be read into the CPU rather than the new state. When in doubt, it is better to separate these instructions with a `NOP` or another instruction not accessing this I/O port.

FIGURE 5-16: SUCCESSIVE I/O OPERATION



Note 1: This example shows write to PORTB followed by a read from PORTB.
2: Data setup time = (0.25 T_{CY} - T_{PD}) where T_{CY} = instruction cycle and T_{PD} = propagation delay of Q1 cycle to output valid. Therefore, at higher clock frequencies, a write followed by a read may be problematic.

11.3.2 PWM DUTY CYCLE

The PWM duty cycle is specified by writing to the CCPR1L register and to the CCP1CON<5:4> bits. Up to 10-bit resolution is available: the CCPR1L contains the eight MSbs and the CCP1CON<5:4> contains the two LSbs. This 10-bit value is represented by CCPR1L:CCP1CON<5:4>. The following equation is used to calculate the PWM duty cycle in time:

EQUATION 11-1: PWM DUTY CYCLE

$$\text{PWM duty cycle} = (\text{CCPR1L:CCP1CON<5:4>}) \bullet \text{Tosc} \bullet (\text{TMR2 prescale value})$$

CCPR1L and CCP1CON<5:4> can be written to at any time, but the duty cycle value is not latched into CCPR1H until after a match between PR2 and TMR2 occurs (i.e., the period is complete). In PWM mode, CCPR1H is a read-only register.

The CCPR1H register and a 2-bit internal latch are used to double buffer the PWM duty cycle. This double buffering is essential for glitchless PWM operation.

When the CCPR1H and 2-bit latch match TMR2 concatenated with an internal 2-bit Q clock or 2 bits of the TMR2 prescaler, the CCP1 pin is cleared.

Maximum PWM resolution (bits) for a given PWM frequency:

EQUATION 11-2: MAXIMUM PWM RESOLUTION

$$\text{PWM Resolution} = \frac{\log\left(\frac{\text{Fosc}}{\text{Fpwm x TMR2 Prescaler}}\right)}{\log(2)} \text{ bits}$$

Note:	If the PWM duty cycle value is longer than the PWM period, the CCP1 pin will not be cleared.

For an example on the PWM period and duty cycle calculation, see the PICmicro™ Mid-Range Reference Manual (DS33023).

11.3.3 SET-UP FOR PWM OPERATION

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

1. Set the PWM period by writing to the PR2 register.
2. Set the PWM duty cycle by writing to the CCPR1L register and CCP1CON<5:4> bits.
3. Make the CCP1 pin an output by clearing the TRISB<3> bit.
4. Set the TMR2 prescale value and enable Timer2 by writing to T2CON.

TABLE 11-4: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS AT 20 MHz

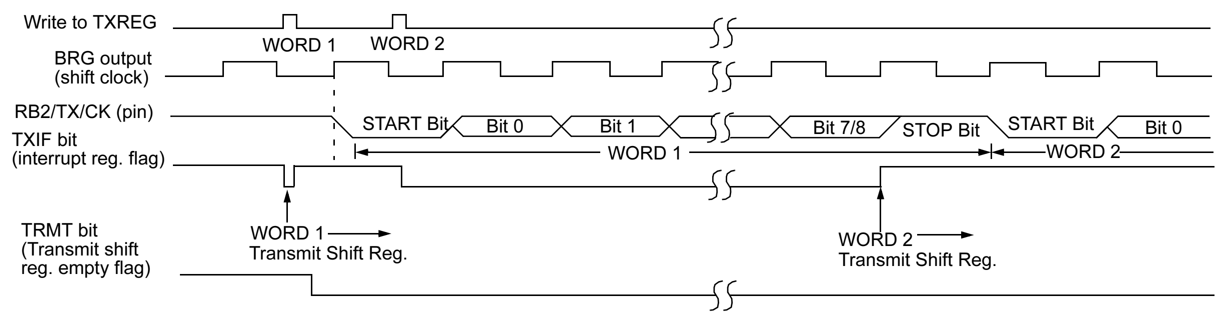
PWM Frequency	1.22 kHz	4.88 kHz	19.53 kHz	78.12 kHz	156.3 kHz	208.3 kHz
Timer Prescaler (1, 4, 16)	16	4	1	1	1	1
PR2 Value	0xFF	0xFF	0xFF	0x3F	0x1F	0x17
Maximum Resolution (bits)	10	10	10	8	7	6.5

TABLE 11-5: REGISTERS ASSOCIATED WITH PWM AND TIMER2

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR	Value on all other RESETS
0Bh/8Bh/ 10Bh/18Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	EEIF	CMIF	RCIF	TXIF	—	CCP1IF	TMR2IF	TMR1IF	0000 -000	0000 -000
8Ch	PIE1	EEIE	CMIE	RCIE	TXIE	—	CCP1IE	TMR2IE	TMR1IE	0000 -000	0000 -000
87h	TRISB	PORTB Data Direction Register								1111 1111	1111 1111
11h	TMR2	Timer2 module's register								0000 0000	0000 0000
92h	PR2	Timer2 module's period register								1111 1111	1111 1111
12h	T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	uuuu
15h	CCPR1L	Capture/Compare/PWM register1 (LSB)								xxxx xxxx	uuuu uuuu
16h	CCPR1H	Capture/Compare/PWM register1 (MSB)								xxxx xxxx	uuuu uuuu
17h	CCP1CON	—	—	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0	--00 0000	--00 0000

Legend: x = unknown, u = unchanged, - = unimplemented read as '0'. Shaded cells are not used by PWM and Timer2.

FIGURE 12-7: ASYNCHRONOUS TRANSMISSION (BACK TO BACK)



Note 1: This timing diagram shows two consecutive transmissions.

TABLE 12-6: REGISTERS ASSOCIATED WITH ASYNCHRONOUS TRANSMISSION

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR	Value on all other RESETS
0Ch	PIR1	EEIF	CMIF	RCIF	TXIF	—	CCP1IF	TMR2IF	TMR1IF	0000 -000	0000 -000
18h	RCSTA	SPEN	RX9	SREN	CREN	ADEN	FERR	OERR	RX9D	0000 -00x	0000 -00x
19h	TXREG	USART Transmit Register								0000 0000	0000 0000
8Ch	PIE1	EEIE	CMIE	RCIE	TXIE	—	CCP1IE	TMR2IE	TMR1IE	0000 -000	0000 -000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

Legend: x = unknown, - = unimplemented locations read as '0'.
Shaded cells are not used for Asynchronous Transmission.

REGISTER 13-2: EECON1 REGISTER (ADDRESS: 9Ch)

U-0	U-0	U-0	U-0	R/W-x	R/W-0	R/S-0	R/S-x
—	—	—	—	WRERR	WREN	WR	RD
bit 7							bit 0

bit 7-4 **Unimplemented:** Read as '0'

bit 3 **WRERR**: EEPROM Error Flag bit
1 = A write operation is prematurely terminated (any $\overline{MCLR}$ Reset, any WDT Reset during normal operation or BOD Reset)
0 = The write operation completed

bit 2 **WREN**: EEPROM Write Enable bit
1 = Allows write cycles
0 = Inhibits write to the data EEPROM

bit 1 **WR**: Write Control bit
1 = Initiates a write cycle. (The bit is cleared by hardware once write is complete. The WR bit can only be set (not cleared) in software.
0 = Write cycle to the data EEPROM is complete

bit 0 **RD**: Read Control bit
1 = Initiates an EEPROM read (read takes one cycle. RD is cleared in hardware. The RD bit can only be set (not cleared) in software).
0 = Does not initiate an EEPROM read

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

14.12 In-Circuit Serial Programming

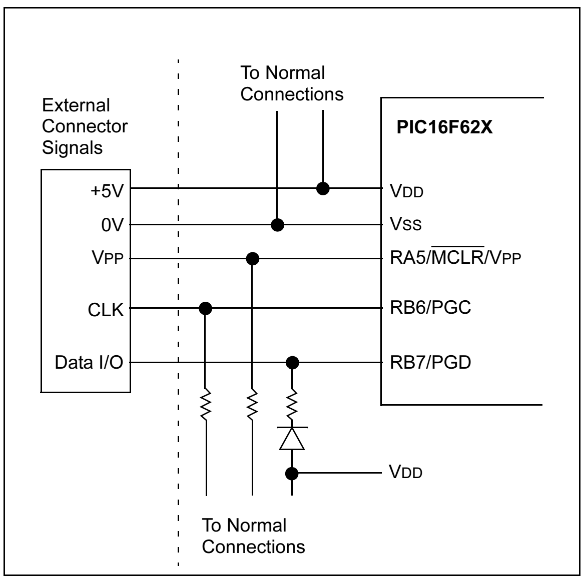
The PIC16F62X microcontrollers can be serially programmed while in the end application circuit. This is simply done with two lines for clock and data, and three other lines for power, ground, and the programming voltage. This allows customers to manufacture boards with unprogrammed devices, and then 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 $\overline{MCLR}$ (V_{PP}) pin from V_{IL} to V_{IHH} (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 Programming Specifications.

A typical in-circuit serial programming connection is shown in Figure 14-18.

FIGURE 14-18: TYPICAL IN-CIRCUIT SERIAL PROGRAMMING CONNECTION



14.13 Low Voltage Programming

The LVP bit of the configuration word, enables the low voltage programming. This mode allows the microcontroller to be programmed via ICSP using only a 5V source. This mode removes the requirement of V_{IHH} to be placed on the $\overline{MCLR}$ pin. The LVP bit is normally erased to '1', which enables the low voltage programming. In this mode, the RB4/PGM pin is dedicated to the programming function and ceases to be a general purpose I/O pin. The device will enter Programming mode when a '1' is placed on the RB4/PGM pin. The HV Programming mode is still available by placing V_{IHH} on the $\overline{MCLR}$ pin.

> **Note 1:** While in this mode, the RB4 pin can no longer be used as a general purpose I/O pin.
>
> **2:** V_{DD} must be 5.0V $\pm$10% during erase/program operations while in low voltage Programming mode.

If Low voltage Programming mode is not used, the LVP bit can be programmed to a '0', and RB4/PGM becomes a digital I/O pin. To program the device, V_{IHH} must be placed onto $\overline{MCLR}$ during programming. The LVP bit may only be programmed when programming is entered with V_{IHH} on $\overline{MCLR}$. The LVP bit cannot be programmed when programming is entered with RB4/PGM.

It should be noted, that once the LVP bit is programmed to 0, High voltage Programming mode can be used to program the device.

 Preliminary

INCF — Increment f

Syntax:	[*label*] INCF f,d
Operands:	$0 \le f \le 127$ $d \in [0,1]$
Operation:	(f) + 1 → (dest)
Status Affected:	Z
Encoding:	00 | 1010 | dfff | ffff
Description:	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'.
Words:	1
Cycles:	1
Example	`INCF REG1, 1`

Before Instruction
REG1 = 0xFF
Z = 0
After Instruction
REG1 = 0x00
Z = 1

INCFSZ — Increment f, Skip if 0

Syntax:	[*label*] INCFSZ f,d
Operands:	$0 \le f \le 127$ $d \in [0,1]$
Operation:	(f) + 1 → (dest), skip if result = 0
Status Affected:	None
Encoding:	00 | 1111 | dfff | ffff
Description:	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'. 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.
Words:	1
Cycles:	1(2)
Example	see below

```
HERE      INCFSZ   REG1, 1
          GOTO     LOOP
CONTINUE  •
          •
          •
```

Before Instruction
PC = address `HERE`
After Instruction
REG1 = REG1 + 1
if CNT = 0,
PC = address `CONTINUE`
if REG1 ≠ 0,
PC = address `HERE +1`

IORLW	Inclusive OR Literal with W			
Syntax:	[*label*] IORLW k			
Operands:	$0 \leq k \leq 255$			
Operation:	(W) .OR. k → (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 = 0			

IORWF	Inclusive OR W with f			
Syntax:	[*label*] IORWF f,d			
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$			
Operation:	(W) .OR. (f) → (dest)			
Status Affected:	Z			
Encoding:	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	`IORWF REG1, 0` Before Instruction REG1 = 0x13 W = 0x91 After Instruction REG1 = 0x13 W = 0x93 Z = 1			

MOVLW	Move Literal to W			
Syntax:	[*label*] MOVLW k			
Operands:	$0 \leq k \leq 255$			
Operation:	k → (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:	[*label*] MOVF f,d			
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$			
Operation:	(f) → (dest)			
Status Affected:	Z			
Encoding:	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 REG1, 0` After Instruction W = value in REG1 register Z = 1			

RRF — Rotate Right f through Carry

Syntax: [*label*] RRF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: See description below

Status Affected: C

Encoding:

00	1100	dfff	ffff

Description: The contents of register 'f' are rotated one bit to the right through the Carry Flag. If 'd' is 0 the result is placed in the W register. If 'd' is 1 the result is placed back in register 'f'.

Words: 1

Cycles: 1

Example: `RRF REG1, 0`

Before Instruction
REG1 = 1110 0110
C = 0
After Instruction
REG1 = 1110 0110
W = 0111 0011
C = 0

SLEEP

Syntax: [*label*] SLEEP

Operands: None

Operation: 00h → WDT,
0 → WDT prescaler,
1 → $\overline{TO}$,
0 → $\overline{PD}$

Status Affected: $\overline{TO}$, $\overline{PD}$

Encoding:

00	0000	0110	0011

Description: The power-down STATUS bit, $\overline{PD}$ is cleared. Timeout STATUS bit, $\overline{TO}$ is set. Watchdog Timer and its prescaler are cleared. The processor is put into SLEEP mode with the oscillator stopped. See Section 14.9 for more details.

Words: 1

Cycles: 1

Example: `SLEEP`

SUBLW — Subtract W from Literal

Syntax: [*label*] SUBLW k

Operands: $0 \le k \le 255$

Operation: k - (W) → (W)

Status Affected: C, DC, Z

Encoding:

11	110x	kkkk	kkkk

Description: The W register is subtracted (2's complement method) from the eight bit literal 'k'. The result is placed in the W register.

Words: 1

Cycles: 1

Example 1: `SUBLW 0x02`

Before Instruction
W = 1
C = ?

After Instruction
W = 1
C = 1; result is positive

Example 2: Before Instruction
W = 2
C = ?

After Instruction
W = 0
C = 1; result is zero

Example 3: Before Instruction
W = 3
C = ?

After Instruction
W = 0xFF
C = 0; result is negative

TABLE 17-1: COMPARATOR SPECIFICATIONS

	Operating Conditions: 3.0V < VDD <5.5V, -40°C < TA < +125°C, unless otherwise stated.						
Param No.	**Characteristics**	**Sym**	**Min**	**Typ**	**Max**	**Units**	**Comments**
D300	Input offset voltage	VIOFF	—	±5.0	±10	mV	
D301*	Input Common mode voltage	VICM	0	—	VDD - 1.5	V	
D302*	Common Mode Rejection Ratio	CMRR	55	—	—	db	
300* 300A	Response Time(1)	TRESP	—	150	400 600	ns ns	16F62X 16LF62X
301	Comparator Mode Change to Output Valid*	TMC2OV	—	—	10	μs	

* These parameters are characterized but not tested.

Note 1: Response time measured with one comparator input at (VDD - 1.5)/2 while the other input transitions from VSS to VDD.

TABLE 17-2: VOLTAGE REFERENCE SPECIFICATIONS

	Operating Conditions: 3.0V < VDD < 5.5V, -40°C < TA < +125°C, unless otherwise stated.						
Spec No.	**Characteristics**	**Sym**	**Min**	**Typ**	**Max**	**Units**	**Comments**
D310	Resolution	VRES	VDD/24	—	VDD/32	LSb	
D311	Absolute Accuracy	VRaa	— —	— —	1/4 1/2	LSb LSb	Low Range (VRR = 1) High Range (VRR = 0)
D312*	Unit Resistor Value (R)	VRur	—	2k	—	Ω	
310*	Settling Time(1)	Tset	—	—	10	μs	

* These parameters are characterized but not tested.

Note 1: Settling time measured while VRR = 1 and VR<3:0> transitions from 0000 to 1111.

TABLE 17-3: DC CHARACTERISTICS: PIC16F62X, PIC16LF62X

DC Characteristics			Standard Operating Conditions (unless otherwise stated)				
Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
		Data EEPROM Memory					
D120	ED	Endurance	1M*	10M	—	E/W	25°C at 5V
D121	VDRW	VDD for read/write	VMIN	—	5.5	V	VMIN = Minimum operating voltage
D122	TDEW	Erase/Write cycle time	—	4	8*	ms	
		Program FLASH Memory					
D130	EP	Endurance	1000*	10000	—	E/W	
D131	VPR	VDD for read	Vmin	—	5.5	V	VMIN = Minimum operating voltage
D132	VPEW	VDD for erase/write	4.5	—	5.5	V	
D133	TPEW	Erase/Write cycle time	—	4	8*	ms	

* These parameters are characterized but not tested.

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

17.4 Timing Diagrams and Specifications

FIGURE 17-6: EXTERNAL CLOCK TIMING

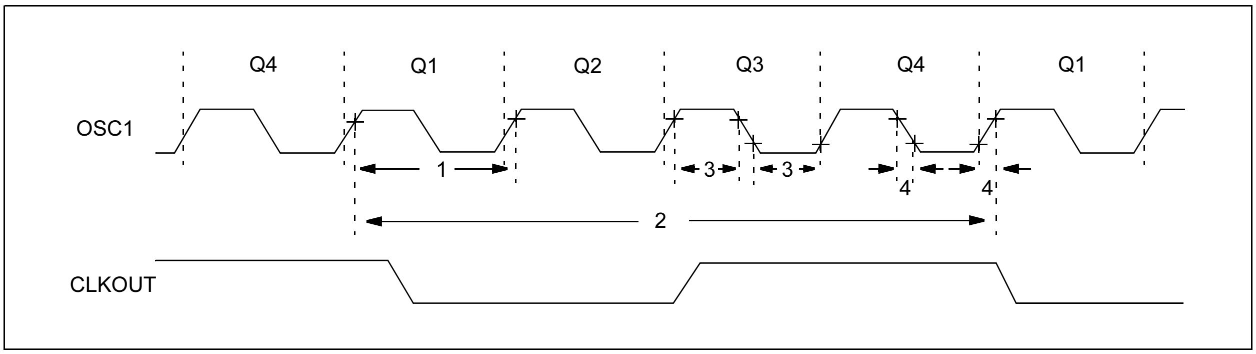


FIGURE 17-10: TIMER0 AND TIMER1 EXTERNAL CLOCK TIMINGS

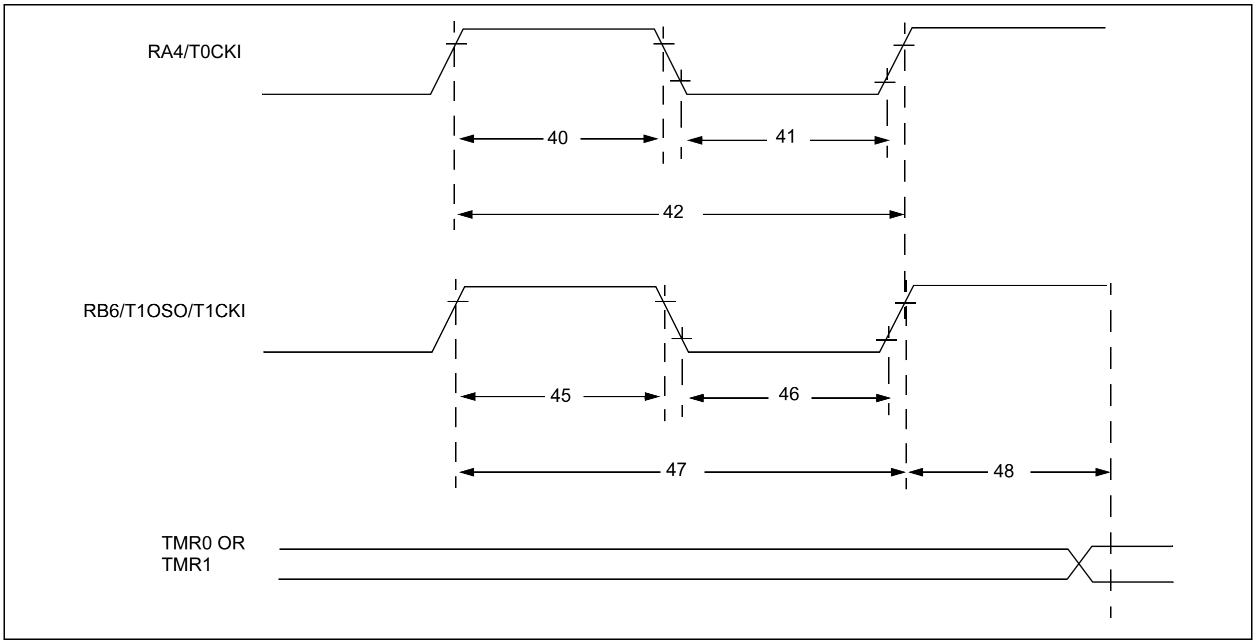


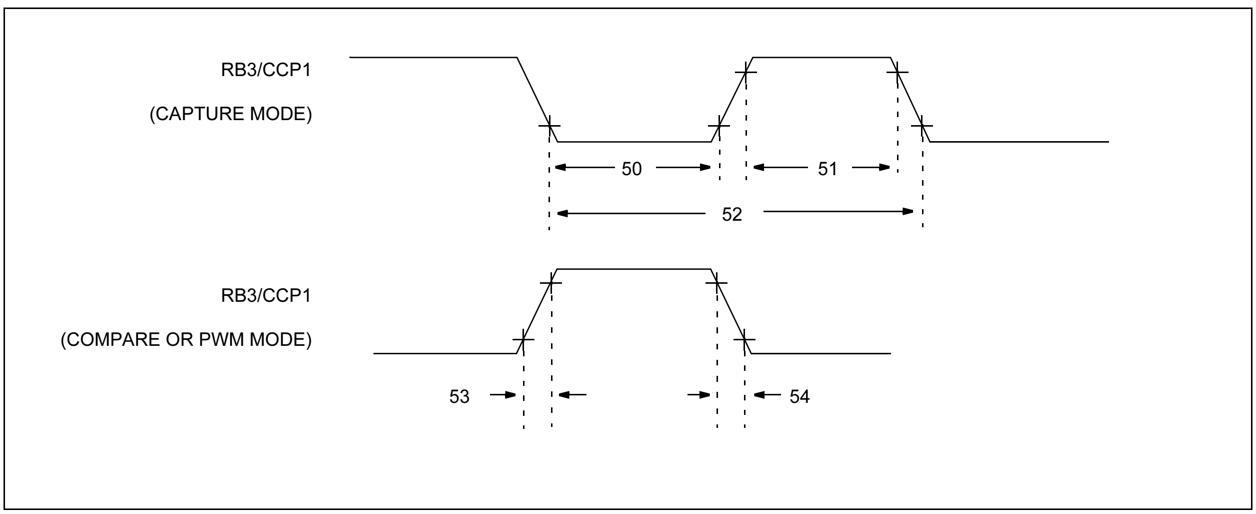
TABLE 17-7: TIMER0 AND TIMER1 EXTERNAL CLOCK REQUIREMENTS

Param No.	Sym	Characteristic			Min	Typ†	Max	Units	Conditions
40*	Tt0H	T0CKI High Pulse Width	No Prescaler		0.5TCY + 20	—	—	ns	
			With Prescaler		10	—	—	ns	
41*	Tt0L	T0CKI Low Pulse Width	No Prescaler		0.5TCY + 20	—	—	ns	
			With Prescaler		10	—	—	ns	
42*	Tt0P	T0CKI Period			Greater of: $\frac{T_{CY} + 40}{N}$	—	—	ns	N = prescale value (2, 4, ..., 256)
45*	Tt1H	T1CKI High Time	Synchronous, No Prescaler		0.5TCY + 20	—	—	ns	
			Synchronous, with Prescaler	16F62X	15	—	—	ns	
				16LF62X	25	—	—	ns	
			Asynchronous	16F62X	30	—	—	ns	
				16LF62X	50	—	—	ns	
46*	Tt1L	T1CKI Low Time	Synchronous, No Prescaler		0.5TCY + 20	—	—	ns	
			Synchronous, with Prescaler	16F62X	15	—	—	ns	
				16LF62X	25	—	—	ns	
			Asynchronous	16F62X	30	—	—	ns	
				16LF62X	50	—	—	ns	
47*	Tt1P	T1CKI input period	Synchronous	16F62X	Greater of: $\frac{T_{CY} + 40}{N}$	—	—	ns	N = prescale value (1, 2, 4, 8)
				16LF62X	Greater of: $\frac{T_{CY} + 40}{N}$	—	—	—	
			Asynchronous	16F62X	60	—	—	ns	
				16LF62X	100	—	—	ns	
	Ft1	Timer1 oscillator input frequency range (oscillator enabled by setting bit T1OSCEN)			DC	—	200	kHz	
48	TCKEZtmr1	Delay from external clock edge to timer increment			2Tosc	—	7Tosc	—	

* 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.

FIGURE 17-11: CAPTURE/COMPARE/PWM TIMINGS



 Preliminary

Note: The graphs and tables provided in this section are for design guidance and are not tested.

FIGURE 18-6: MAXIMUM IDD VS FOSC OVER VDD (LP MODE)

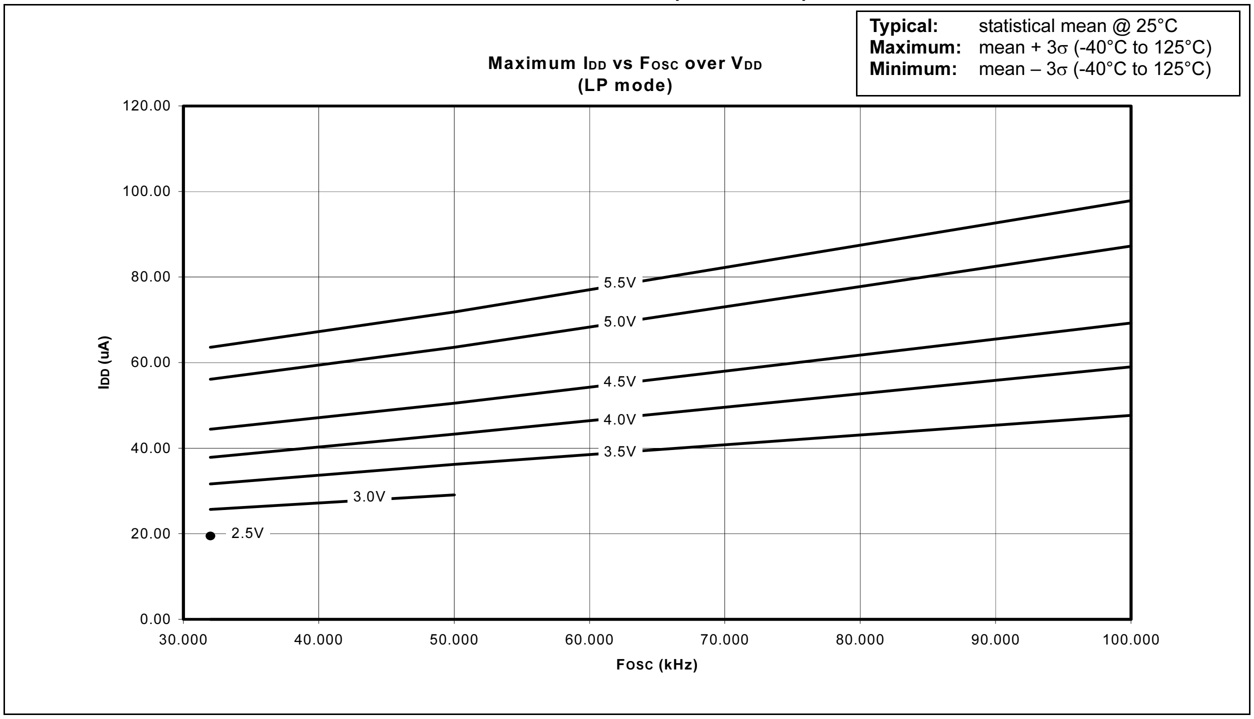
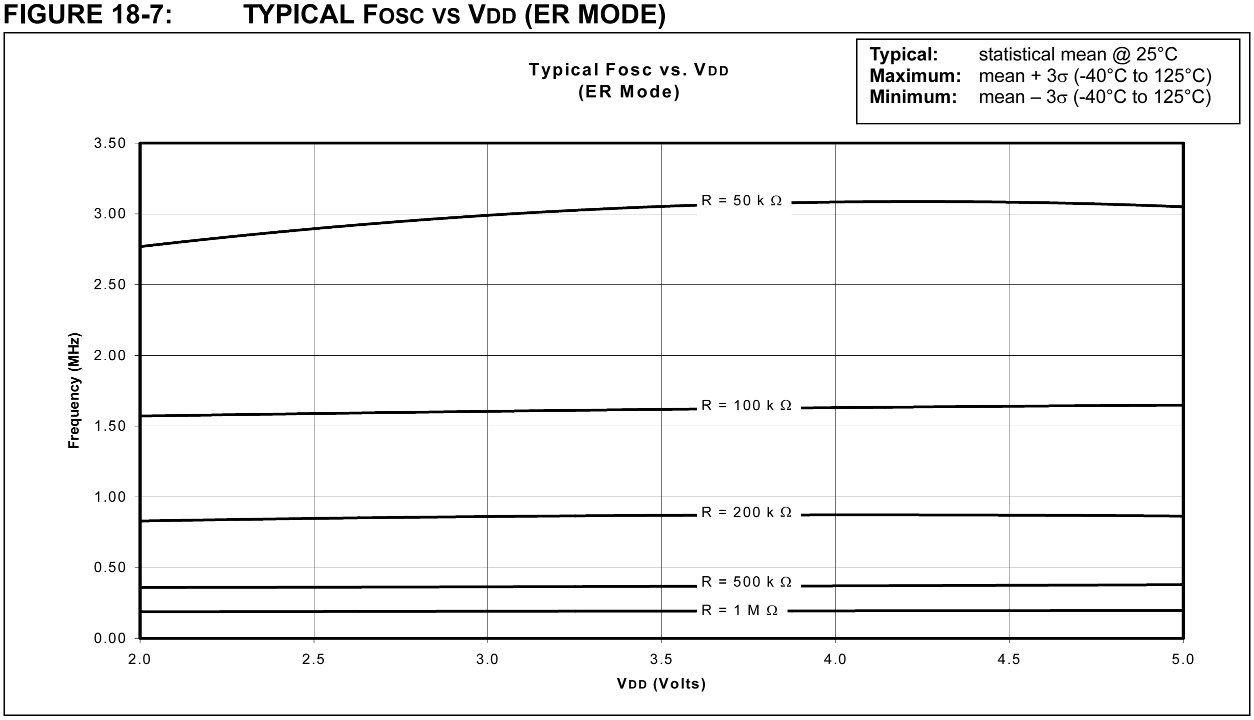
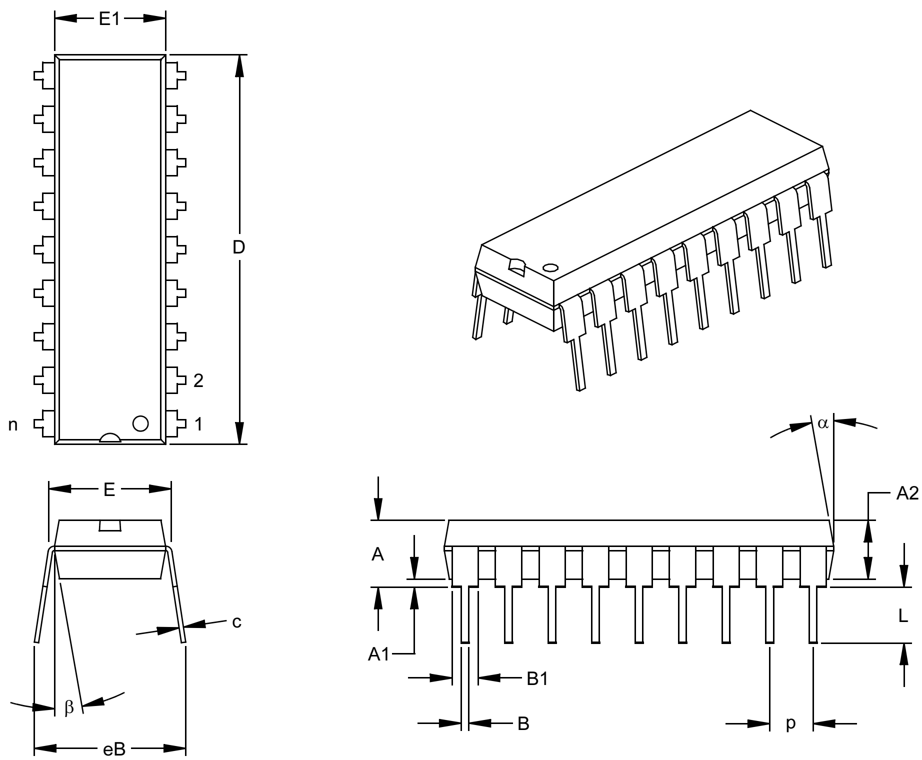


FIGURE 18-7: TYPICAL FOSC VS VDD (ER MODE)



 Preliminary

K04-007 18-Lead Plastic Dual In-line (P) – 300 mil



	Units	INCHES*			MILLIMETERS		
	Dimension Limits	MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n		18			18	
Pitch	p		.100			2.54	
Top to Seating Plane	A	.140	.155	.170	3.56	3.94	4.32
Molded Package Thickness	A2	.115	.130	.145	2.92	3.30	3.68
Base to Seating Plane	A1	.015			0.38		
Shoulder to Shoulder Width	E	.300	.313	.325	7.62	7.94	8.26
Molded Package Width	E1	.240	.250	.260	6.10	6.35	6.60
Overall Length	D	.890	.898	.905	22.61	22.80	22.99
Tip to Seating Plane	L	.125	.130	.135	3.18	3.30	3.43
Lead Thickness	c	.008	.012	.015	0.20	0.29	0.38
Upper Lead Width	B1	.045	.058	.070	1.14	1.46	1.78
Lower Lead Width	B	.014	.018	.022	0.36	0.46	0.56
Overall Row Spacing §	eB	.310	.370	.430	7.87	9.40	10.92
Mold Draft Angle Top	α	5	10	15	5	10	15
Mold Draft Angle Bottom	β	5	10	15	5	10	15

* Controlling Parameter
§ Significant Characteristic

Notes:
Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010” (0.254mm) per side.
JEDEC Equivalent: MS-001
Drawing No. C04-007