



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 "<u>Embedded -</u> <u>Microcontrollers</u>"

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	3.5КВ (2К х 14)
Program Memory Type	FLASH
EEPROM Size	128 x 8
RAM Size	224 x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 5.5V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Surface Mount
Package / Case	20-SSOP (0.209", 5.30mm Width)
Supplier Device Package	20-SSOP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16f628-04-ss

Email: info@E-XFL.COM

Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong

Table of Contents

1.0	General Description	5
2.0	PIC16F62X Device Varieties	7
3.0	Architectural Overview	9
4.0	Memory Organization	15
5.0	I/O Ports	29
6.0	Timer0 Module	43
7.0	Timer1 Module	46
8.0	Timer2 Module	50
9.0	Comparator Module	53
10.0	Voltage Reference Module	59
11.0	Capture/Compare/PWM (CCP) Module	61
12.0	Universal Synchronous/Asynchronous Receiver/ Transmitter (USART) Module	67
13.0	Data EEPROM Memory	87
14.0	Special Features of the CPU	91
15.0	Instruction Set Summary	. 107
16.0	Development Support	. 121
17.0	Electrical Specifications	. 127
18.0	DC and AC Characteristics Graphs and Tables	. 143
19.0	Packaging Information	. 157

TO OUR VALUED CUSTOMERS

It is our intention to provide our valued customers with the best documentation possible to ensure successful use of your Microchip products. To this end, we will continue to improve our publications to better suit your needs. Our publications will be refined and enhanced as new volumes and updates are introduced.

If you have any questions or comments regarding this publication, please contact the Marketing Communications Department via E-mail at **docerrors@mail.microchip.com** or fax the **Reader Response Form** in the back of this data sheet to (480) 792-4150. We welcome your feedback.

Most Current Data Sheet

To obtain the most up-to-date version of this data sheet, please register at our Worldwide Web site at:

http://www.microchip.com

You can determine the version of a data sheet by examining its literature number found on the bottom outside corner of any page. The last character of the literature number is the version number, (e.g., DS30000A is version A of document DS30000).

Errata

An errata sheet, describing minor operational differences from the data sheet and recommended workarounds, may exist for current devices. As device/documentation issues become known to us, we will publish an errata sheet. The errata will specify the revision of silicon and revision of document to which it applies.

To determine if an errata sheet exists for a particular device, please check with one of the following:

- Microchip's Worldwide Web site; http://www.microchip.com
- Your local Microchip sales office (see last page)
- The Microchip Corporate Literature Center; U.S. FAX: (480) 792-7277

When contacting a sales office or the literature center, please specify which device, revision of silicon and data sheet (include literature number) you are using.

Customer Notification System

Register on our web site at www.microchip.com/cn to receive the most current information on all of our products.

2.1 Clocking Scheme/Instruction Cycle

The clock input (OSC1/CLKIN/RA7 pin) is internally divided by four to generate four non-overlapping quadrature clocks namely Q1, Q2, Q3 and Q4. Internally, the program counter (PC) is incremented every Q1, the instruction is fetched from the program memory and latched into the instruction register in Q4. The instruction is decoded and executed during the following Q1 through Q4. The clocks and instruction execution flow is shown in Figure 2-2.

2.2 Instruction Flow/Pipelining

An "Instruction Cycle" consists of four Q cycles (Q1, Q2, Q3 and Q4). The instruction fetch and execute are pipelined such that fetch takes one instruction cycle while decode and execute takes another instruction cycle. However, due to the pipelining, each instruction effectively executes in one cycle. If an instruction causes the program counter to change, (e.g., GOTO) then two cycles are required to complete the instruction (Example 2-1).

A fetch cycle begins with the program counter (PC) incrementing in Q1.

In the execution cycle, the fetched instruction is latched into the "Instruction Register (IR)" in cycle Q1. This instruction is then decoded and executed during the Q2, Q3, and Q4 cycles. Data memory is read during Q2 (operand read) and written during Q4 (destination write).



FIGURE 2-2: CLOCK/INSTRUCTION CYCLE

EXAMPLE 2-1: INSTRUCTION PIPELINE FLOW



All instructions are single cycle, except for any program branches. These take two cycles since the fetch instruction is "flushed" from the pipeline while the new instruction is being fetched and then executed.

3.2.2.2 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). See Section 6.3.1					

REGISTER 3-2: OPTION REGISTER (ADDRESS: 81h, 181h)

	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	INTED	TOCS	T0SE	PSA	PS2	PS1	PS0
	bit 7	•						bit 0
bit 7	RBPU : PC	ORTB Pull-	up Enable bi	t				
	1 = PORT 0 = PORT	B pull-ups B pull-ups	are disabled are enabled	by individual	port latch value	es		
bit 6	INTEDG:	nterrupt E	dge Select bi	t				
	1 = Interru 0 = Interru	pt on rising pt on fallin	g edge of RB g edge of RB	0/INT pin 30/INT pin				
bit 5	TOCS: TM	R0 Clock \$	Source Selec	t bit				
	1 = Transi 0 = Interna	tion on RA al instructio	4/T0CKI pin on cycle clocl	(CLKOUT)				
bit 4	4 T0SE : TMR0 Source Edge Select bit							
	1 = Increm 0 = Increm	nent on hig nent on low	h-to-low tran /-to-high tran	sition on RA sition on RA	4/T0CKI pin 4/T0CKI pin			
bit 3	PSA: Pres	caler Assi	gnment bit					
	1 = Presca 0 = Presca	aler is assi aler is assi	gned to the V gned to the T	VDT īmer0 modu	le			
bit 2-0	PS2:PS0:	Prescaler	Rate Select	bits				
		Bit Value	TMR0 Rate	WDT Rate				
		000 001 010	1:2 1:4 1:8	1:1 1:2 1:4				
		011	1:16	1:8				

001		•••
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

3.3 PCL and PCLATH

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

FIGURE 3-3: LOADING OF PC IN DIFFERENT SITUATIONS



3.3.1 COMPUTED GOTO

A computed GOTO is accomplished by adding an offset to the program counter (ADDWF PCL). When doing a table read using a computed GOTO method, care should be exercised if the table location crosses a PCL memory boundary (each 256 byte block). Refer to the application note *"Implementing a Table Read"* (AN556).

3.3.2 STACK

The PIC16F62X family has an 8-level deep x 13-bit wide hardware stack (Figure 3-1 and Figure 3-2). The stack space is not part of either program or data space and the stack pointer is not readable or writable. The PC is PUSHed onto the stack when a CALL instruction is executed or an interrupt causes a branch. The stack is POPed in the event of a RETURN, RETLW or a RETFIE instruction execution. PCLATH is not affected by a PUSH or POP operation.

The stack operates as a circular buffer. This means that after the stack has been PUSHed eight times, the ninth push overwrites the value that was stored from the first push. The tenth push overwrites the second push (and so on).

Note 1: There are no STATUS bits to indicate stack overflow or stack underflow conditions.

2: There are no instructions/mnemonics called PUSH or POP. These are actions that occur from the execution of the CALL, RETURN, RETLW and RETFIE instructions, or the vectoring to an interrupt address.

3.4 Indirect Addressing, INDF and FSR Registers

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

Indirect addressing is possible by using the INDF register. Any instruction using the INDF register actually accesses data pointed to by the file select register (FSR). Reading INDF itself indirectly will produce 00h. Writing to the INDF register indirectly results in a nooperation (although STATUS bits may be affected). An effective 9-bit address is obtained by concatenating the 8-bit FSR register and the IRP bit (STATUS<7>), as shown in Figure 3-4.

A simple program to clear RAM location 20h-2Fh using indirect addressing is shown in Example 3-1.

EXAMPLE 3-1: Indirect Addressing

	movlw	0x20	;initialize pointer
	movwf	FSR	;to RAM
NEXT	clrf	INDF	clear INDF register;
	incf	FSR	;inc pointer
	btfss	FSR,4	;all done?
	goto	NEXT	;no clear next
			;yes continue

NOTES:



FIGURE 5-3: BLOCK DIAGRAM OF THE RA3/AN3 PIN









BLOCK DIAGRAM OF THE RA5/MCLR/VPP PIN



FIGURE 5-6:

BLOCK DIAGRAM OF RA6/OSC2/CLKOUT PIN



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
05h	PORTA	RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0	xxxx 0000	xxxu 0000
85h	TRISA	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	1111 1111	1111 1111
1Fh	CMCON	C2OUT	C10UT	C2INV	C1INV	CIS	CM2	CM1	CM0	0000 0000	0000 0000
9Fh	VRCON	VREN	VROE	VRR	—	VR3	VR2	VR1	VR0	000- 0000	000- 0000

 TABLE 5-2:
 SUMMARY OF REGISTERS ASSOCIATED WITH PORTA⁽¹⁾

Legend: - = Unimplemented locations, read as '0', u = unchanged, x = unknown **Note 1:** Shaded bits are not used by PORTA.

5.2 PORTB and TRISB Registers

PORTB is an 8-bit wide bi-directional port. The corresponding data direction register is TRISB. A '1' in the TRISB register puts the corresponding output driver in a Hi-impedance mode. A '0' in the TRISB register puts the contents of the output latch on the selected pin(s).

PORTB is multiplexed with the external interrupt, USART, CCP module and the TMR1 clock input/output. The standard port functions and the alternate port functions are shown in Table 5-3. Alternate port functions override TRIS setting when enabled.

Reading PORTB register reads the status of the pins, whereas writing to it will write to the port latch. All write operations are read-modify-write operations. So a write to a port implies that the port pins are first read, then this value is modified and written to the port data latch.

Each of the PORTB pins has a weak internal pull-up ($\approx 200 \ \mu A \ typical$). A single control bit can turn on all the pull-ups. This is done by clearing the RBPU (OPTION<7>) bit. The weak pull-up is automatically turned off when the port pin is configured as an output. The pull-ups are disabled on Power-on Reset.

Four of PORTB's pins, RB<7:4>, have an interrupt-onchange feature. Only pins configured as inputs can cause this interrupt to occur (i.e., any RB<7:4> pin configured as an output is excluded from the interrupt-onchange comparison). The input pins (of RB7:RB4) are compared with the old value latched on the last read of PORTB. The "mismatch" outputs of RB7:RB4 are OR'ed together to generate the RBIF interrupt (flag latched in INTCON<0>).

This interrupt can wake the device from SLEEP. The user, in the interrupt service routine, can clear the interrupt in the following manner:

- a) Any read or write of PORTB. This will end the mismatch condition.
- b) Clear flag bit RBIF.

A mismatch condition will continue to set flag bit RBIF. Reading PORTB will end the mismatch condition and allow flag bit RBIF to be cleared. This interrupt on mismatch feature, together with software configurable pull-ups on these four pins allow easy interface to a key pad and make it possible for wake-up on key-depression. (See AN552)

Note:	If a change on the I/O pin should occur					
	when a read operation is being executed					
	(start of the Q2 cycle), then the RBIF inter-					
	rupt flag may not get set.					

The interrupt-on-change feature is recommended for wake-up on key depression operation and operations where PORTB is only used for the interrupt-on-change feature. Polling of PORTB is not recommended while using the interrupt-on-change feature.



FIGURE 5-9: BLOCK DIAGRAM OF



9.5 Comparator Outputs

The comparator outputs are read through the CMCON register. These bits are read only. The comparator outputs may also be directly output to the RA3 and RA4 I/O pins. When the CM<2:0> = 110, multiplexors in the output path of the RA3 and RA4/T0CK1 pins will switch and the output of each pin will be the unsynchronized output of the comparator. The uncertainty of each of the comparators is related to the input offset voltage and the response time given in the specifications. Figure 9-3 shows the comparator output block diagram.

The TRISA bits will still function as an output enable/ disable for the RA3 and RA4/T0CK1 pins while in this mode.

- Note 1: When reading the PORT register, all pins configured as analog inputs will read as a '0'. Pins configured as digital inputs will convert an analog input according to the Schmitt Trigger input specification.
 - 2: Analog levels on any pin that is defined as a digital input may cause the input buffer to consume more current than is specified.

FIGURE 9-3: COMPARATOR OUTPUT BLOCK DIAGRAM



11.1 Capture Mode

In Capture mode, CCPR1H:CCPR1L captures the 16-bit value of the TMR1 register when an event occurs on pin RB3/CCP1. An event is defined as:

- · Every falling edge
- Every rising edge
- · Every 4th rising edge
- Every 16th rising edge

An event is selected by control bits CCP1M3:CCP1M0 (CCP1CON<3:0>). When a capture is made, the Interrupt Request Flag bit CCP1IF (PIR1<2>) is set. It must be cleared in software. If another capture occurs before the value in register CCPR1 is read, the old captured value will be lost.

11.1.1 CCP PIN CONFIGURATION

In Capture mode, the RB3/CCP1 pin should be configured as an input by setting the TRISB<3> bit.

Note:	If the RB3/CCP1 is configured as an out-			
	put, a write to the port can cause a capture			
	condition.			

TABLE 11-2:CAPTURE MODE OPERATION
BLOCK DIAGRAM



11.1.2 TIMER1 MODE SELECTION

Timer1 must be running in Timer mode or Synchronized Counter mode for the CCP module to use the capture feature. In Asynchronous Counter mode, the capture operation may not work.

11.1.3 SOFTWARE INTERRUPT

When the Capture mode is changed, a false capture interrupt may be generated. The user should keep bit CCP1IE (PIE1<2>) clear to avoid false interrupts and should clear the flag bit CCP1IF following any such change in Operating mode.

11.1.4 CCP PRESCALER

There are four prescaler settings, specified by bits CCP1M3:CCP1M0. Whenever the CCP module is turned off, or the CCP module is not in Capture mode, the prescaler counter is cleared. This means that any RESET will clear the prescaler counter.

Switching from one capture prescaler to another may generate an interrupt. Also, the prescaler counter will not be cleared, therefore the first capture may be from a non-zero prescaler. Example 11-1 shows the recommended method for switching between capture prescalers. This example also clears the prescaler counter and will not generate the "false" interrupt.

EXAMPLE 11-1: CHANGING BETWEEN CAPTURE PRESCALERS

CLRF	CCP1CON	;Turn CCP module off
MOVLW	NEW_CAPT_PS	;Load the W reg with
		; the new prescaler
		; mode value and CCP ON
MOVWF	CCP1CON	;Load CCP1CON with this
		; value

11.2 Compare Mode

In Compare mode, the 16-bit CCPR1 register value is constantly compared against the TMR1 register pair value. When a match occurs, the RB3/CCP1 pin is:

- Driven High
- Driven Low
- · Remains Unchanged

The action on the pin is based on the value of control bits CCP1M3:CCP1M0 (CCP1CON<3:0>). At the same time, interrupt flag bit CCP1IF is set.

FIGURE 11-1: COMPARE MODE OPERATION BLOCK DIAGRAM



11.2.1 CCP PIN CONFIGURATION

The user must configure the RB3/CCP1 pin as an output by clearing the TRISB<3> bit.

Note: Clearing the CCP1CON register will force the RB3/CCP1 compare output latch to the default low level. This is not the data latch.

12.1 USART Baud Rate Generator (BRG)

The BRG supports both the Asynchronous and Synchronous modes of the USART. It is a dedicated 8-bit baud rate generator. The SPBRG register controls the period of a free running 8-bit timer. In Asynchronous mode bit BRGH (TXSTA<2>) also controls the baud rate. In Synchronous mode bit BRGH is ignored. Table 12-1 shows the formula for computation of the baud rate for different USART modes which only apply in Master mode (internal clock).

Given the desired baud rate and Fosc, the nearest integer value for the SPBRG register can be calculated using the formula in Table 12-1. From this, the error in baud rate can be determined.

Example 12-1 shows the calculation of the baud rate error for the following conditions:

Fosc = 16 MHz Desired Baud Rate = 9600 BRGH = 0

SYNC = 0

EXAMPLE 12-1: CALCULATING BAUD RATE ERROR

Desired Baud rate = Fosc / (64(X + 1))

9600 = 16000000 / (64(+1))X X = $\hat{1}25.042^{\circ}$

Calculated Baud Rate = 16000000 / (64(25 + 1)) = 9615

Error = (<u>Calculated Baud Rate = Desired Baud Rate</u>) Desired Baud Rate

> = (9615 - 9600)/ 9600 = 0.16%

It may be advantageous to use the high baud rate (BRGH = 1) even for slower baud clocks. This is because the Fosc/(16(X + 1)) equation can reduce the baud rate error in some cases.

Writing a new value to the SPBRG register, causes the BRG timer to be RESET (or cleared), this ensures the BRG does not wait for a timer overflow before outputting the new baud rate.

TABLE 12-1: BAUD RATE FORMULA

SYNC	BRGH = 0 (Low Speed)	BRGH = 1 (High Speed)
0	(Asynchronous) Baud Rate = Fosc/(64(X+1))	Baud Rate= Fosc/(16(X+1))
1	(Synchronous) Baud Rate = Fosc/(4(X+1))	NA

Legend: X = value in SPBRG (0 to 255)

TABLE 12-2: REGISTERS ASSOCIATED WITH BAUD RATE GENERATOR

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
98h	TXSTA	CSRC	TX9	TXEN	SYNC		BRGH	TRMT	TX9D	0000 -010	0000 -010
18h	RCSTA	SPEN	RX9	SREN	CREN	ADEN	FERR	OERR	RX9D	x00- 0000	0000 -00x
99h SPBRG Baud Rate Generator Register										0000 0000	0000 0000

Legend: x = unknown, - = unimplemented read as '0'.

Shaded cells are not used by the BRG.

BAUD	Fosc = 20 MHz		SPBRG	16 MHz		SPBRG	, 10 MHz		SPBRG
RATE (K)	KBAUD	ERROR	value (decimal)	KBAUD	ERROR	value (decimal)	KBAUD	ERROR	value (decimal)
0.3	NA			NA			NA		
1.2	1.221	+1.73%	255	1.202	+0.16%	207	1.202	+0.16%	129
2.4	2.404	+0.16%	129	2.404	+0.16%	103	2.404	+0.16%	64
9.6	9.469	-1.36%	32	9.615	+0.16%	25	9.766	+1.73%	15
19.2	19.53	+1.73%	15	19.23	+0.16%	12	19.53	+1.73V	7
76.8	78.13	+1.73%	3	83.33	+8.51%	2	78.13	+1.73%	1
96	104.2	+8.51%	2	NA	_	_	NA	_	_
300	312.5	+4.17%	0	NA		_	NA	_	
500	NA			NA		_	NA	_	
HIGH	312.5	_	0	250	_	0	156.3	_	0
LOW	1.221		255	0.977		255	0.6104	_	255

TABLE 12-4: BAUD RATES FOR ASYNCHRONOUS MODE (BRGH = 0)

BAUD	Fosc = 7.15	909 MHz	SPBRG	5.0688 MHz		SPBRG	4 MHz		SPBRG
RATE (K)	KBAUD	ERROR	value (decimal)	KBAUD	ERROR	value (decimal)	KBAUD	ERROR	value (decimal)
0.3	NA	_	_	0.31	+3.13%	255	0.3005	-0.17%	207
1.2	1.203	+0.23%	92	1.2	0	65	1.202	+1.67%	51
2.4	2.380	-0.83%	46	2.4	0	32	2.404	+1.67%	25
9.6	9.322	-2.90%	11	9.9	+3.13%	7	NA	_	_
19.2	18.64	-2.90%	5	19.8	+3.13%	3	NA	_	_
76.8	NA	_	_	79.2	+3.13%	0	NA	_	_
96	NA	_	_	NA		_	NA	_	_
300	NA	_	_	NA		_	NA	_	_
500	NA	_	_	NA	_	_	NA	_	_
HIGH	111.9	_	0	79.2		0	62.500	_	0
LOW	0.437	—	255	0.3094	—	255	3.906	—	255

BAUD	Fosc = 3.579	9545 MHz	SPBRG	1 MHz		SPBRG	32.768 MHz		SPBRG
RATE (K)	KBAUD	ERROR	value (decimal)	KBAUD	ERROR	value (decimal)	KBAUD	ERROR	value (decimal)
0.3	0.301	+0.23%	185	0.300	+0.16%	51	0.256	-14.67%	1
1.2	1.190	-0.83%	46	1.202	+0.16%	12	NA	_	—
2.4	2.432	+1.32%	22	2.232	-6.99%	6	NA	_	—
9.6	9.322	-2.90%	5	NA	_	_	NA	—	—
19.2	18.64	-2.90%	2	NA			NA	_	_
76.8	NA		—	NA	_		NA	—	—
96	NA		_	NA			NA	_	_
300	NA	_	_	NA		_	NA	_	_
500	NA		_	NA			NA	_	_
HIGH	55.93		0	15.63		0	0.512	_	0
LOW	0.2185	—	255	0.0610		255	0.0020	—	255

13.3 READING THE EEPROM DATA MEMORY

To read a data memory location, the user must write the address to the EEADR register and then set control bit RD (EECON1<0>). The data is available, in the very next cycle, in the EEDATA register; therefore it can be read in the next instruction. EEDATA will hold this value until another read or until it is written to by the user (during a write operation).

EXAMPLE 13-1: DATA EEPROM READ

BSF	STATUS, H	RPO ;	Bank 1
MOVLW	CONFIG_AI	DDR ;	
MOVWF	EEADR	;	Address to read
BSF	EECON1, H	RD ;	EE Read
MOVF	EEDATA, V	vi ;	W = EEDATA
BCF	STATUS, H	RPO ;	Bank 0

13.4 WRITING TO THE EEPROM DATA MEMORY

To write an EEPROM data location, the user must first write the address to the EEADR register and the data to the EEDATA register. Then the user must follow a specific sequence to initiate the write for each byte.

EXAMPLE 13-2: DATA EEPROM WRITE

Required Sequence	BSF BSF MOVLW MOVWF MOVLW MOVWF BSF	STATUS, RP0 EECON1, WREN INTCON, GIE 55h EECON2 AAh EECCN2 EECON1,WR	;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;	Bank 1 Enable write Disable INTs. Write 55h Write AAh Set WR bit begin write
	BSF	INTCON, GIE	;	Enable INTs.

The write will not initiate if the above sequence is not exactly followed (write 55h to EECON2, write AAh to EECON2, then set WR bit) for each byte. We strongly recommend that interrupts be disabled during this code segment. A cycle count is executed during the required sequence. Any number that is not equal to the required cycles to execute the required sequence will cause the data not to be written into the EEPROM.

Additionally, the WREN bit in EECON1 must be set to enable write. This mechanism prevents accidental writes to data EEPROM due to errant (unexpected) code execution (i.e., lost programs). The user should keep the WREN bit clear at all times, except when updating EEPROM. The WREN bit is not cleared by hardware.

After a write sequence has been initiated, clearing the WREN bit will not affect this write cycle. The WR bit will be inhibited from being set unless the WREN bit is set.

At the completion of the write cycle, the WR bit is cleared in hardware and the EE Write Complete Interrupt Flag bit (EEIF) is set. The user can either enable this interrupt or poll this bit. The EEIF bit in the PIR1 registers must be cleared by software.

13.5 WRITE VERIFY

Depending on the application, good programming practice may dictate that the value written to the Data EEPROM should be verified (Example 13-3) to the desired value to be written. This should be used in applications where an EEPROM bit will be stressed near the specification limit.

EXAMPLE 13-3: WRITE VERIFY

```
BSF
         STATUS, RP0 ; Bank 1
   MOVF
         EEDATA, W
   BSF
         EECON1, RD
                      ; Read the
                      ; value written
; Is the value written (in W reg) and
; read (in EEDATA) the same?
   SUBWF EEDATA, W
   BCF STATUS, RPO ; Bank0
   BTFSS STATUS, Z
                      ; Is difference 0?
   GOTO WRITE ERR
                      ; NO, Write error
                      ; YES, Good write
   :
                      ; Continue program
   .
```

13.6 PROTECTION AGAINST SPURIOUS WRITE

There are conditions when the device may not want to write to the data EEPROM memory. To protect against spurious EEPROM writes, various mechanisms have been built in. On power-up, WREN is cleared. Also, the Power-up Timer (72 ms duration) prevents EEPROM write.

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

13.7 DATA EEPROM OPERATION DURING CODE PROTECT

When the device is code protected, the CPU is able to read and write unscrambled data to the Data EEPROM.

FIGURE 14-16: WATCHDOG TIMER BLOCK DIAGRAM



TABLE 14-10: SUMMARY OF WATCHDOG TIMER REGISTERS

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR Reset	Value on all other RESETS
2007h	Config. bits	LVP	BODEN	MCLRE	FOSC2	PWRTE	WDTE	FOSC1	FOSC0	uuuu uuuu	uuuu uuuu
81h	OPTION	RBPU	INTEDG	TOCS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111

Legend: -= Unimplemented location, read as "0", + = Reserved for future use

Note 1: Shaded cells are not used by the Watchdog Timer.

14.9 Power-Down Mode (SLEEP)

The Power-down mode is entered by executing a SLEEP instruction.

If enabled, the Watchdog Timer will be cleared but keeps running, the PD bit in the STATUS register is cleared, the TO bit is set, and the oscillator driver is turned off. The I/O ports maintain the status they had, before SLEEP was executed (driving high, low, or hi-impedance).

For lowest current consumption in this mode, all I/O pins should be either at VDD, or VSS, with no external circuitry drawing current from the I/O pin and the comparators, and VREF should be disabled. I/O pins that are hi-impedance inputs should be pulled high or low externally to avoid switching currents caused by floating inputs. The TOCKI input should also be at VDD or VSS for lowest current consumption. The contribution from on-chip pull-ups on PORTB should be considered.

The MCLR pin must be at a logic high level (VIHMC).

Note:	It should be noted that a RESET generated									
	by a WDT timeout does not drive MCLR									
	pin low.									

PIC16F62X

INCF	Increment f	INCFSZ	Increment f, Skip if 0			
Syntax:	[<i>label</i>] INCF f,d	Syntax:	[<i>label</i>] INCFSZ f,d			
Operands:	$0 \le f \le 127$ $d \in [0,1]$	Operands:	$0 \le f \le 127$ d \in [0,1]			
Operation:	(f) + 1 \rightarrow (dest)	Operation:	(f) + 1 \rightarrow (dest), skip if result = 0			
Status Affected:	Z	Status Affected:	None			
Encoding:	00 1010 dfff ffff	Encoding:	00 1111 dfff ffff			
Description: Words: Cycles:	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'. 1	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			
Example	INCF REG1, 1		executed instead making it a two-cycle instruction.			
	REG1 = 0xFF	Words:	1			
	Z = 0	Cycles:	1 ⁽²⁾			
	After Instruction REG1 = 0x00 Z = 1	Example	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

-

17.2 DC Characteristics: PIC16F62X (Commercial, Industrial, Extended) PIC16LF62X (Commercial, Industrial)

DC CHAR	RACTERIS	TICS	$\begin{array}{llllllllllllllllllllllllllllllllllll$						
Param. No.	Sym	Characteristic/Device	Min	Тур†	Мах	Unit	Conditions		
	VIL	Input Low Voltage							
D030		I/O ports with TTL buffer	Vss	_	0.8 0.15 Vdd	V V	VDD = 4.5V to 5.5V otherwise		
D031 D032		with Schmitt Trigger input MCLR, RA4/T0CKI,OSC1 (in ER mode)	Vss Vss	_	0.2 Vdd 0.2 Vdd	V V	(Note1)		
D033		OSC1 (in XT and HS) OSC1 (in LP)	Vss Vss	_	0.3 Vdd 0.6 Vdd - 1.0	V V			
	Vih	Input High Voltage							
D040		I/O ports with TTL buffer	2.0V .25 Vdd + 0.8V	_	Vdd Vdd	V V	VDD = 4.5V to 5.5V otherwise		
D041 D042 D043		with Schmitt Trigger input MCLR RA4/T0CKI OSC1 (XT, HS and LP)	0.8 VDD 0.8 VDD 0.7 VDD		Vdd Vdd Vdd	V V V			
D043A		OSC1 (in ER mode)	0.9 Vdd			V	(Note1)		
D070	IPURB	PORTB weak pull-up current	50	200	400	μA	VDD = 5.0V, VPIN = VSS		
	١L	Input Leakage Current ^{(2), (3)}							
D060 D061 D063		I/O ports (Except PORTA) PORTA RA4/T0CKI OSC1, MCLR	_ _ _		$\pm 1.0 \\ \pm 0.5 \\ \pm 1.0 \\ \pm 5.0$	μΑ μΑ μΑ	$\label{eq:VSS} \begin{array}{l} VPIN \leq VDD, \ \text{pin at hi-impedance} \\ VSS \leq VPIN \leq VDD, \ \text{pin at hi-impedance} \\ VSS \leq VPIN \leq VDD \\ VSS \leq VPIN \leq VDD, \ XT, \ HS \ \text{and} \ LP \ \text{osc} \\ configuration \end{array}$		
	Vol	Output Low Voltage	•		•				
D080 D083		I/O ports OSC2/CLKOUT (ER only)		 	0.6 0.6 0.6 0.6	V V V V	IOL=8.5 mA, VDD=4.5V, -40° to +85°C IOL=7.0 mA, VDD=4.5V, +125°C IOL=1.6 mA, VDD=4.5V, -40° to +85°C IOL=1.2 mA, VDD=4.5V, +125°C		
	Vон	Output High Voltage ⁽³⁾			I				
D090 D092		I/O ports (Except RA4) OSC2/CLKOUT (ER only)	Vdd - 0.7 Vdd - 0.7 Vdd - 0.7			V V V	IOH=-3.0 mA, VDD=4.5V, -40° to +85°C IOH=-2.5 mA, VDD=4.5V, +125°C IOH=-1.3 mA, VDD=4.5V, -40° to +85°C		
2002			VDD - 0.7	_	_	v	Iон=-1.0 mA, VDD=4.5V, +125°С		
D150	Vod	Open-Drain High Voltage		_	8.5	V	RA4 pin PIC16F62X, PIC16LF62X*		
		Capacitive Loading Specs on C	Output Pins						
D100*	COSC2	OSC2 pin		—	15	pF	In XT, HS and LP modes when external clock used to drive OSC1.		
D101*	Cio	All I/O pins/OSC2 (in ER mode)		_	50	pF			

These parameters are characterized but not tested.

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

Note 1: In ER oscillator configuration, the OSC1 pin is a Schmitt Trigger input. It is not recommended that the PIC16F62X be driven with external clock in ER mode.

2: The leakage current on the 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.

17.3 Timing Parameter Symbology

The timing parameter symbols have been created with one of the following formats:

1. TppS2ppS

2. TppS

z. rppo							
Т							
F	Frequency	Т	Time				
Lowercase subscripts (pp) and their meanings:							
рр							
ck	CLKOUT	osc	OSC1				
io	I/O port	tO	TOCKI				
mc	MCLR						
Uppercase	e letters and their meanings:						
S							
F	Fall	Р	Period				
н	High	R	Rise				
I	Invalid (Hi-impedance)	V	Valid				
L	Low	Z	Hi-Impedance				

FIGURE 17-5: LOAD CONDITIONS



Param No.	Sym	Characteristic	Min	Тур†	Max	Units	Conditions
	Fosc	External CLKIN Frequency ⁽¹⁾	DC		4	MHz	XT and ER Osc mode, VDD = 5.0V
			DC	_	20	MHz	HS Osc mode
			DC	—	200	kHz	LP Osc mode
		Oscillator Frequency ⁽¹⁾		_	4	MHz	ER Osc mode, VDD = 5.0V
			0.1	—	4	MHz	XT Osc mode
			1	_	20 200	MHz kHz	HS Osc mode LP Osc mode
			3.65	4	4.28	MHz	INTRC mode (fast), VDD = 5.0V
				37		kHz	INTRC mode (slow)
4	INTRC	Internal Calibrated RC	3.65	4.00	4.28	MHz	VDD = 5.0V
5	ER	External Biased ER Frequency	10 kHz		8 MHz		VDD = 5.0V
1	Tosc	External CLKIN Period ⁽¹⁾	250	_	—	ns	XT and ER Osc mode
			50	_	—	ns	HS Osc mode
			5	—	—	μS	LP Osc mode
		Oscillator Period ⁽¹⁾	250		_	ns	ER Osc mode
			250	—	10,000	ns	XT Osc mode
			50	—	1,000	ns	HS Osc mode
			5			μs	LP Osc mode
				250		ns	INTRC mode (fast)
				27		μs	INTRC mode (slow)
2	Тсу	Instruction Cycle Time	1.0	TCY	DC	ns	Tcy = 4/Fosc
3	TosL, TosH	External CLKIN (OSC1) High External CLKIN Low	100 *		_	ns	XT oscillator, Tosc L/H duty cycle*

TABLE 17-4: EXTERNAL CLOCK TIMING REQUIREMENTS

* These parameters are characterized but not tested.

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

Note 1: Instruction cycle period (Tcy) equals four times the input oscillator time-based 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.

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



FIGURE 18-18: VOH VS IOH OVER TEMP (C) VDD = 5V



