



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	20MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	33
Program Memory Size	14KB (8K x 14)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	368 x 8
Voltage - Supply (Vcc/Vdd)	4V ~ 5.5V
Data Converters	A/D 8x8b
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Through Hole
Package / Case	40-DIP (0.600", 15.24mm)
Supplier Device Package	40-PDIP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16f77-i-p

Email: info@E-XFL.COM

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

Pin Name	DIP Pin#	PLCC Pin#	QFP Pin#	I/O/P Type	Buffer Type	Description
						PORTD is a bi-directional I/O port or parallel slave port when interfacing to a microprocessor bus.
RD0/PSP0	19	21	38		ST/TTL ⁽³⁾	
RD0				I/O		Digital I/O. Derollel Slove Port dete
RD1/PSP1	20	22	39	1/0	ST/TTI (3)	
RD1	20		00	I/O	01/112	Digital I/O.
PSP1				I/O	(2)	Parallel Slave Port data.
RD2/PSP2	21	23	40		ST/TTL ⁽³⁾	
PSP2				1/O		Parallel Slave Port data.
RD3/PSP3	22	24	41		ST/TTL ⁽³⁾	
RD3				I/O		Digital I/O.
PSP3 RD4/RSD4	27	30	2	1/0	ST/TTI (3)	Parallel Slave Port data.
RD4	21	50	2	I/O	SI/TIL /	Digital I/O.
PSP4				I/O	(-)	Parallel Slave Port data.
RD5/PSP5	28	31	3	1/0	ST/TTL ⁽³⁾	
PSP5				1/O 1/O		Parallel Slave Port data.
RD6/PSP6	29	32	4	., -	ST/TTL ⁽³⁾	
RD6				I/O		Digital I/O.
PSP6			-	I/O	ot (3)	Parallel Slave Port data.
RD7/PSP7 RD7	30	33	5	I/O	SI/IIL	Digital I/O.
PSP7				I/O		Parallel Slave Port data.
					(2)	PORTE is a bi-directional I/O port.
RE0/RD/AN5	8	9	25	1/0	ST/TTL ⁽³⁾	
				1/0		Digital I/O. Read control for parallel slave port.
AN5				I		Analog input 5.
RE1/WR/AN6	9	10	26		ST/TTL ⁽³⁾	
RE1 WR				1/O		Digital I/O. Write control for parallel slave port
AN6				I		Analog input 6.
RE2/CS/AN7	10	11	27		ST/TTL ⁽³⁾	
RE2				I/O		Digital I/O.
AN7						Analog input 7.
Vss	12,31	13,34	6,29	Р	_	Ground reference for logic and I/O pins.
Vdd	11,32	12,35	7,28	Р	_	Positive supply for logic and I/O pins.
NC	—	1,17,2	12,13,		_	These pins are not internally connected. These pins should
Legend: L = input		0, 40 0 = 0	33, 34 utput	<u>ا</u> /() = input/outor	P = power

TABLE 1-3: PIC16F74 AND PIC16F77 PINOUT DESCRIPTION (CONTINUED)

- = Not used TTL = TTL input ST = Schmitt Trigger input

Note 1: This buffer is a Schmitt Trigger input when configured as an external interrupt.

2: This buffer is a Schmitt Trigger input when used in Serial Programming mode.

3: This buffer is a Schmitt Trigger input when configured as general purpose I/O and a TTL input when used in the Parallel Slave Port mode (for interfacing to a microprocessor bus).

4: This buffer is a Schmitt Trigger input when configured in RC Oscillator mode and a CMOS input otherwise.

PIC16F7X

FIGURE 2-3:

PIC16F74/73 REGISTER FILE MAP

Ą	File ddress		File Address		File Address		File Address
Indirect addr.(*)	00h	Indirect addr.(*)	80h	Indirect addr.(*)	100h	Indirect addr.(*)	180h
TMR0	01h	OPTION REG	81h	TMR0	101h	OPTION REG	181h
PCL	02h	PCI	82h	PCL	102h	PCL	182h
STATUS	03h	STATUS	83h	STATUS	103h	STATUS	183h
FSR	04h	FSR	84h	FSR	104h	FSR	184h
PORTA	05h	TRISA	85h		105h		185h
PORTB	06h	TRISB	86h	PORTB	106h	TRISB	186h
PORTC	07h	TRISC	87h		107h		187h
PORTD ⁽¹⁾	08h	TRISD ⁽¹⁾	88h		108h		188h
PORTE ⁽¹⁾	09h	TRISE ⁽¹⁾	89h		109h		189h
PCLATH	0Ah	PCLATH	8Ah	PCLATH	10Ah	PCLATH	18Ah
INTCON	0Bh	INTCON	8Bh	INTCON	10Bh	INTCON	18Bh
PIR1	0Ch	PIE1	8Ch	PMDATA	10Ch	PMCON1	18Ch
PIR2	0Dh	PIE2	8Dh	PMADR	10Dh		18Dh
TMR1L	0Eh	PCON	8Eh	PMDATH	10Eh		18Eh
TMR1H	0Fh		8Eh	PMADRH	10Fh		18Fh
T1CON	10h		90h		110h		190h
TMR2	11h		91h				
T2CON	12h	PR2	92h				
SSPBUF	13h	SSPADD	93h				
SSPCON	14h	SSPSTAT	94h				
CCPR1L	15h		95h				
CCPR1H	16h		96h				
CCP1CON	17h	-	97h				
RCSTA	18h	TXSTA	98h				
TXREG	19h	SPBRG	99h				
RCREG	1Ah		9Ah				
CCPR2L	1Bh		9Bh				
CCPR2H	1Ch		9Ch				
CCP2CON	1Dh		9Dh				
ADRES	1Eh		9Eh				
ADCON0	1Fh	ADCON1	9Fh		4.001		1406
	20h		A0h		120n		TAUN
			, (011				
General		General					
Purpose Register		Purpose Register		accesses		accesses	
OC Dutoo				20h-7Fh		A0h - FFh	155h
96 Bytes		96 Bytes			16Fh		1EF11
					170n		IFUN
	7Eh		FFb		17Fh		1FFh
Bank 0	/ 1 11	Bank 1		Bank 2		Bank 3	
 Unimpleme * Not a phys a 1: These reg 	ented data ical registe isters are i	memory location er. not implemented	s, read as on 28-pin	'0'. devices.			

NOTES:

4.3 PORTC and the TRISC Register

PORTC is an 8-bit wide, bi-directional port. The corresponding data direction register is TRISC. Setting a TRISC bit (= '1') will make the corresponding PORTC pin an input (i.e., put the corresponding output driver in a Hi-Impedance mode). Clearing a TRISC bit (= '0') will make the corresponding PORTC pin an output (i.e., put the contents of the output latch on the selected pin).

PORTC is multiplexed with several peripheral functions (Table 4-5). PORTC pins have Schmitt Trigger input buffers.

When enabling peripheral functions, care should be taken in defining TRIS bits for each PORTC pin. Some peripherals override the TRIS bit to make a pin an output, while other peripherals override the TRIS bit to make a pin an input. Since the TRIS bit override is in effect while the peripheral is enabled, read-modify-write instructions (BSF, BCF, XORWF) with TRISC as destination should be avoided. The user should refer to the corresponding peripheral section for the correct TRIS bit settings, and to Section 13.1 for additional information on read-modify-write operations.

FIGURE 4-5:

PORTC BLOCK DIAGRAM (PERIPHERAL OUTPUT OVERRIDE)



3: Peripheral OE (output enable) is only activated if peripheral select is active.

Name	Bit#	Buffer Type	Function
RC0/T1OSO/T1CKI	bit0	ST	Input/output port pin or Timer1 oscillator output/Timer1 clock input.
RC1/T1OSI/CCP2	bit1	ST	Input/output port pin or Timer1 oscillator input or Capture2 input/Compare2 output/PWM2 output.
RC2/CCP1	bit2	ST	Input/output port pin or Capture1 input/Compare1 output/PWM1 output.
RC3/SCK/SCL	bit3	ST	RC3 can also be the synchronous serial clock for both SPI and I^2C modes.
RC4/SDI/SDA	bit4	ST	RC4 can also be the SPI Data In (SPI mode) or Data I/O (I ² C mode).
RC5/SDO	bit5	ST	Input/output port pin or Synchronous Serial Port data output.
RC6/TX/CK	bit6	ST	Input/output port pin or USART Asynchronous Transmit or Synchronous Clock.
RC7/RX/DT	bit7	ST	Input/output port pin or USART Asynchronous Receive or Synchronous Data.

TABLE 4-5: PORTC FUNCTIONS

Legend: ST = Schmitt Trigger input

TABLE 4-6: SUMMARY OF REGISTERS ASSOCIATED WITH PORTC

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other RESETS
07h	PORTC	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	xxxx xxxx	uuuu uuuu
87h	TRISC	PORTC	PORTC Data Direction Register							1111 1111	1111 1111

Legend: x = unknown, u = unchanged



FIGURE 4-10: PARALLEL SLAVE PORT READ WAVEFORMS



TABLE 4-11: REGISTERS ASSOCIATED WITH PARALLEL SLAVE PORT

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other RESETS
08h	PORTD	Port data I	Port data latch when written: Port pins when read								uuuu uuuu
09h	PORTE	_			—		RE2	RE1	RE0	xxx	uuu
89h	TRISE	IBF	OBF	IBOV	PSPMODE	_	PORTE D	Data Direct	tion Bits	0000 -111	0000 -111
0Ch	PIR1	PSPIF ⁽¹⁾	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
8Ch	PIE1	PSPIE ⁽¹⁾	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
9Fh	ADCON1	_	—			_	PCFG2	PCFG1	PCFG0	000	000

Legend: x = unknown, u = unchanged, - = unimplemented, read as '0'. Shaded cells are not used by the Parallel Slave Port.

Note 1: Bits PSPIE and PSPIF are reserved on the PIC16F73/76; always maintain these bits clear.

6.0 TIMER1 MODULE

The Timer1 module is a 16-bit timer/counter consisting of two 8-bit registers (TMR1H and TMR1L), which are readable and writable. The TMR1 Register pair (TMR1H:TMR1L) increments from 0000h to FFFFh and rolls over to 0000h. The TMR1 Interrupt, if enabled, is generated on overflow, which is latched in interrupt flag bit TMR1IF (PIR1<0>). This interrupt can be enabled/disabled by setting/clearing TMR1 interrupt enable bit TMR1IE (PIE1<0>).

Timer1 can operate in one of two modes:

- As a timer
- · As a counter

The operating mode is determined by the clock select bit, TMR1CS (T1CON<1>).

In Timer mode, Timer1 increments every instruction cycle. In Counter mode, it increments on every rising edge of the external clock input.

Timer1 can be enabled/disabled by setting/clearing control bit TMR1ON (T1CON<0>).

Timer1 also has an internal "RESET input". This RESET can be generated by either of the two CCP modules as the special event trigger (see Sections 8.1 and 8.2). Register 6-1 shows the Timer1 Control register.

When the Timer1 oscillator is enabled (T1OSCEN is set), the RC1/T1OSI/CCP2 and RC0/T1OSO/T1CKI pins become inputs. That is, the TRISC<1:0> value is ignored and these pins read as '0'.

Additional information on timer modules is available in the PICmicro[™] Mid-Range MCU Family Reference Manual (DS33023).

REGISTER 6-1: T1CON: TIMER1 CONTROL REGISTER (ADDRESS 10h)

	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	_	—	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR10N
	bit 7							bit 0
bit 7-6	Unimplem	nented: Rea	ad as '0'					
bit 5-4	T1CKPS1	:T1CKPS0:	Timer1 Inpu	ut Clock Pres	scale Select I	bits		
	11 = 1:8 P	rescale valu	he					
	10 = 1:4 P	rescale valu	Je					
	01 = 1:2 P 00 = 1:1 P	rescale vali rescale vali	re Te					
bit 3	T10SCEN	I: Timer1 Os	scillator Ena	ble Control k	oit			
	1 = Oscilla	ator is enabl	ed					
	0 = Oscilla	ator is shut-o	off (the oscill	ator inverter	is turned off	to eliminate	power draii	ר)
bit 2	T1SYNC:	Timer1 Exte	ernal Clock I	nput Synchr	onization Co	ntrol bit		
	TMR1CS :	<u>= 1:</u>						
	1 = Do not	t synchroniz	e external c	lock input				
	0 = Synch	ronize exter	nal clock inp	out				
	TMR1CS :	<u>= 0:</u>			I I 			
		ignorea. Tin	neri uses th		JCK when TW	$ \mathbf{R} ^{1}\mathbf{CS}=0.$		
Dit 1	IMR1CS:	Timer1 Clo	ck Source S	elect bit				
	1 = Extern 0 = Interna	al clock from	m pin RC0/T sc/4)	1050/110	(I (on the risi	ng edge)		
bit 0	TMR10N:	Timer1 On	bit					
	1 = Enable	es Timer1						
	0 = Stops	Timer1						
	Legend:							
	R = Reada	able bit	W = V	Nritable bit	U = Unin	nplemented	bit, read as	'0'
	- n = Value	e at POR re	set '1' = l	Bit is set	'0' = Bit i	s cleared	x = Bit is ι	Inknown

6.1 **Timer1 Operation in Timer Mode**

Timer mode is selected by clearing the TMR1CS (T1CON<1>) bit. In this mode, the input clock to the timer is Fosc/4. The synchronize control bit T1SYNC (T1CON<2>) has no effect, since the internal clock is always in sync.

6.2 **Timer1 Counter Operation**

Timer1 may operate in Asynchronous or Synchronous mode, depending on the setting of the TMR1CS bit.

When Timer1 is being incremented via an external source, increments occur on a rising edge. After Timer1 is enabled in Counter mode, the module must first have a falling edge before the counter begins to increment.



6.3 **Timer1 Operation in Synchronized Counter Mode**

Counter mode is selected by setting bit TMR1CS. In this mode, the timer increments on every rising edge of clock input on pin RC1/T1OSI/CCP2, when bit T1OSCEN is set, or on pin RC0/T1OSO/T1CKI, when bit T1OSCEN is cleared.

If TISYNC is cleared, then the external clock input is synchronized with internal phase clocks. The synchronization is done after the prescaler stage. The prescaler stage is an asynchronous ripple counter.

In this configuration, during SLEEP mode, Timer1 will not increment even if the external clock is present, since the synchronization circuit is shut-off. The prescaler, however, will continue to increment.



FIGURE 6-2: TIMER1 BLOCK DIAGRAM

6.4 Timer1 Operation in Asynchronous Counter Mode

If control bit T1SYNC (T1CON<2>) is set, the external clock input is not synchronized. The timer continues to increment asynchronous to the internal phase clocks. The timer will continue to run during SLEEP and can generate an interrupt on overflow, which will wake-up the processor. However, special precautions in software are needed to read/write the timer (Section 6.4.1).

In Asynchronous Counter mode, Timer1 cannot be used as a time-base for capture or compare operations.

6.4.1 READING AND WRITING TIMER1 IN ASYNCHRONOUS COUNTER MODE

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

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

Reading the 16-bit value requires some care. The example code provided in Example 6-1 and Example 6-2 demonstrates how to write to and read Timer1 while it is running in Asynchronous mode.

EXAMPLE 6-1: WRITING A 16-BIT FREE-RUNNING TIMER

; All	interrupts	are disabled	
CLRF	TMR1L	; Clear Low byte, Ensures no rollover into TMR1H	
MOVLW	HI_BYTE	; Value to load into TMR1H	
MOVWF	TMR1H, F	; Write High byte	
MOVLW	LO_BYTE	; Value to load into TMR1L	
MOVWF	TMR1H, F	; Write Low byte	
; Re-	enable the i	nterrupt (if required)	
CONTI	NUE	; Continue with your code	

EXAMPLE 6-2: READING A 16-BIT FREE-RUNNING TIMER

; All int	errupts a	re	disabled
MOVF TN	R1H, W	;	Read high byte
MOVWF TN	PH		
MOVF TN	R1L, W	;	Read low byte
MOVWF TN	PL		
MOVF TN	R1H, W	;	Read high byte
SUBWF TN	PH, W	;	Sub 1st read with 2nd read
BTFSC ST	ATUS,Z	;	Is result = 0
GOTO CO	NTINUE	;	Good 16-bit read
; TMR1L m	y have r	01	led over between the read of the high and low bytes.
; Reading	the high	a	nd low bytes now will read a good value.
MOVF TN	R1H, W	;	Read high byte
MOVWF TN	PH		
MOVF TN	R1L, W	;	Read low byte
MOVWF TN	PL	;	Re-enable the Interrupt (if required)
CONTINUE		;	Continue with your code

7.0 TIMER2 MODULE

Timer2 is an 8-bit timer with a prescaler and a postscaler. It can be used as the PWM time-base for the PWM mode of the CCP module(s). The TMR2 register is readable and writable, and is cleared on any device RESET.

The input clock (Fosc/4) has a prescale option of 1:1, 1:4 or 1:16, selected by control bits T2CKPS1:T2CKPS0 (T2CON<1:0>).

The Timer2 module has an 8-bit period register, PR2. Timer2 increments from 00h until it matches PR2 and then resets to 00h on the next increment cycle. PR2 is a readable and writable register. The PR2 register is initialized to FFh upon RESET.

The match output of TMR2 goes through a 4-bit postscaler (which gives a 1:1 to 1:16 scaling inclusive) to generate a TMR2 interrupt (latched in flag bit TMR2IF, (PIR1<1>)).

Timer2 can be shut-off by clearing control bit TMR2ON (T2CON<2>) to minimize power consumption.

Register 7-1 shows the Timer2 control register.

Additional information on timer modules is available in the PICmicro[™] Mid-Range MCU Family Reference Manual (DS33023).

7.1 Timer2 Prescaler and Postscaler

The prescaler and postscaler counters are cleared when any of the following occurs:

- a write to the TMR2 register
- a write to the T2CON register
- any device RESET (POR, MCLR Reset, WDT Reset or BOR)

TMR2 is not cleared when T2CON is written.

7.2 Output of TMR2

The output of TMR2 (before the postscaler) is fed to the SSP module, which optionally uses it to generate shift clock.



8.0 CAPTURE/COMPARE/PWM MODULES

Each Capture/Compare/PWM (CCP) module contains a 16-bit register which can operate as a:

- 16-bit Capture register
- 16-bit Compare register
- PWM Master/Slave Duty Cycle register

Both the CCP1 and CCP2 modules are identical in operation, with the exception being the operation of the special event trigger. Table 8-1 and Table 8-2 show the resources and interactions of the CCP module(s). In the following sections, the operation of a CCP module is described with respect to CCP1. CCP2 operates the same as CCP1, except where noted.

8.1 CCP1 Module

Capture/Compare/PWM Register1 (CCPR1) is comprised of two 8-bit registers: CCPR1L (low byte) and CCPR1H (high byte). The CCP1CON register controls the operation of CCP1. The special event trigger is generated by a compare match and will clear both TMR1H and TMR1L registers.

8.2 CCP2 Module

Capture/Compare/PWM Register1 (CCPR1) is comprised of two 8-bit registers: CCPR1L (low byte) and CCPR1H (high byte). The CCP2CON register controls the operation of CCP2. The special event trigger is generated by a compare match; it will clear both TMR1H and TMR1L registers, and start an A/D conversion (if the A/D module is enabled).

Additional information on CCP modules is available in the PICmicro[™] Mid-Range MCU Family Reference Manual (DS33023) and in Application Note AN594, "Using the CCP Modules" (DS00594).

TABLE 8-1: CCP MODE - TIMER RESOURCES REQUIRED

CCP Mode	Timer Resource
Capture	Timer1
Compare	Timer1
PWM	Timer2

TABLE 8-2:INTERACTION OF TWO CCP MODULES

CCPx Mode	CCPy Mode	Interaction
Capture	Capture	Same TMR1 time-base.
Capture	Compare	Same TMR1 time-base.
Compare	Compare	Same TMR1 time-base.
PWM	PWM	The PWMs will have the same frequency and update rate (TMR2 interrupt). The rising edges are aligned.
PWM	Capture	None.
PWM	Compare	None.

	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R-0	R-0	R-x
	SPEN	RX9	SREN	CREN	—	FERR	OERR	RX9D
	bit 7							bit 0
bit 7	SPEN: Se 1 = Serial 0 = Serial	rial Port Ena port enabled port disabled	ble bit I (configures d	RC7/RX/DT a	and RC6/TX	C/CK pins a	s serial por	t pins)
bit 6	RX9 : 9-bit 1 = Select: 0 = Select:	Receive Ena s 9-bit recep s 8-bit recep	able bit tion tion					
bit 5	SREN: Sin Asynchron Don't care Synchronc 1 = Enable 0 = Disable This bit is o Synchronc Don't care	ngle Receive nous mode: hous mode - M es single rec es single rec cleared after hous mode - S	Enable bit <u>Master:</u> eive eive reception is <u>slave:</u>	complete.				
bit 4	CREN: Co Asynchron 1 = Enable 0 = Disable Synchronc 1 = Enable 0 = Disable	ntinuous Re lous mode: es continuou es continuou ous mode: es continuou es continuou	ceive Enable s receive ls receive s receive unt ls receive) bit il enable bit (CREN is clea	ared (CRE	N overrides	SREN)
bit 3	Unimplem	ented: Rea	d as '0'					
bit 2	FERR: Fra 1 = Framir 0 = No fra	uming Error b ng error (can ming error	bit be updated	by reading R	CREG regis	ter and rec	ceive next v	alid byte)
bit 1	OERR : Ov 1 = Overru 0 = No ove	verrun Error I in error (can errun error	oit be cleared b	by clearing bit	CREN)			
bit 0	RX9D: 9th Can be pa	bit of Recei rity bit (parity	ved Data / to be calcu	ated by firmw	vare)			
	Legend:							
	R = Reada	able bit	W = W	/ritable bit	U = Unim	plemented	bit, read as	s 'O'

'1' = Bit is set

'0' = Bit is cleared

REGISTER 10-2: RCSTA: RECEIVE STATUS AND CONTROL REGISTER (ADDRESS 18h)

- n = Value at POR reset

x = Bit is unknown

12.0 SPECIAL FEATURES OF THE CPU

These devices have a host of features intended to maximize system reliability, minimize cost through elimination of external components, provide power saving operating modes and offer code protection. These are:

- Oscillator Selection
- RESET
 - Power-on Reset (POR)
 - Power-up Timer (PWRT)
 - Oscillator Start-up Timer (OST)
 - Brown-out Reset (BOR)
- Interrupts
- Watchdog Timer (WDT)
- SLEEP
- Code Protection
- ID Locations
- In-Circuit Serial Programming

These devices have a Watchdog Timer, which can be enabled or disabled, using a configuration bit. It runs off its own RC oscillator for added reliability.

There are two timers that offer necessary delays on power-up. One is the Oscillator Start-up Timer (OST), intended to keep the chip in RESET until the crystal oscillator is stable. The other is the Power-up Timer (PWRT), which provides a fixed delay of 72 ms (nominal) on power-up only. It is designed to keep the part in RESET while the power supply stabilizes, and is enabled or disabled, using a configuration bit. With these two timers on-chip, most applications need no external RESET circuitry. SLEEP mode is designed to offer a very low current power-down mode. The user can wake-up from SLEEP through external RESET, Watchdog Timer Wake-up, or through an interrupt.

Several oscillator options are also made available to allow the part to fit the application. The RC oscillator option saves system cost while the LP crystal option saves power. Configuration bits are used to select the desired oscillator mode.

Additional information on special features is available in the PICmicro[™] Mid-Range Reference Manual (DS33023).

12.1 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, which can be accessed only during programming.

12.11.1 INT INTERRUPT

External interrupt on the RB0/INT pin is edge triggered, either rising, if bit INTEDG (OPTION_REG<6>) is set, or falling, if the INTEDG bit is clear. When a valid edge appears on the RB0/INT pin, flag bit INTF (INTCON<1>) is set. This interrupt can be disabled by clearing enable bit INTE (INTCON<4>). Flag bit INTF must be cleared in software in the Interrupt Service Routine before re-enabling this interrupt. The INT interrupt can wake-up the processor from SLEEP, if bit INTE was set prior to going into SLEEP. The status of global interrupt enable bit GIE decides whether or not the processor branches to the interrupt vector following wakeup. See Section 12.14 for details on SLEEP mode.

12.11.2 TMR0 INTERRUPT

An overflow (FFh \rightarrow 00h) in the TMR0 register will set flag bit TMR0IF (INTCON<2>). The interrupt can be enabled/disabled by setting/clearing enable bit TMR0IE (INTCON<5>). (Section 5.0)

12.11.3 PORTB INTCON CHANGE

An input change on PORTB<7:4> sets flag bit RBIF (INTCON<0>). The interrupt can be enabled/disabled by setting/clearing enable bit RBIE (INTCON<4>), see Section 4.2.

12.12 Context Saving During Interrupts

During an interrupt, only the return PC value is saved on the stack. Typically, users may wish to save key registers during an interrupt (i.e., W, PCLATH and STA-TUS registers). This will have to be implemented in software, as shown in Example 12-1.

For the PIC16F73/74 devices, the register W_TEMP must be defined in both banks 0 and 1 and must be defined at the same offset from the bank base address (i.e., If W_TEMP is defined at 20h in bank 0, it must also be defined at A0h in bank 1.). The registers, PCLATH_TEMP and STATUS_TEMP, are only defined in bank 0.

Since the upper 16 bytes of each bank are common in the PIC16F76/77 devices, temporary holding registers W_TEMP, STATUS_TEMP and PCLATH_TEMP should be placed in here. These 16 locations don't require banking and, therefore, make it easier for context save and restore. The same code shown in Example 12-1 can be used.

MOVWF	W_TEMP	;Copy W to TEMP register	
SWAPF	STATUS,W	;Swap status to be saved into W	
CLRF	STATUS	;bank 0, regardless of current bank, Clears IRP,RP1,RP0	
MOVWF	STATUS TEMP	;Save status to bank zero STATUS TEMP register	
MOVF	PCLATH, W	;Only required if using pages 1, 2 and/or 3	
MOVWF	PCLATH_TEMP	;Save PCLATH into W	
CLRF	PCLATH	;Page zero, regardless of current page	
:			
:(ISR)		;Insert user code here	
:			
MOVF	PCLATH_TEMP, W	;Restore PCLATH	
MOVWF	PCLATH	;Move W into PCLATH	
SWAPF	STATUS_TEMP,W	;Swap STATUS_TEMP register into W	
		;(sets bank to original state)	
MOVWF	STATUS	;Move W into STATUS register	
SWAPF	W_TEMP,F	;Swap W_TEMP	
SWAPF	W_TEMP,W	;Swap W_TEMP into W	

EXAMPLE 12-1: SAVING STATUS, W, AND PCLATH REGISTERS IN RAM

PIC16F7X

MOVF	Move f
Syntax:	[<i>label</i>] MOVF f,d
Operands:	$\begin{array}{l} 0 \leq f \leq 127 \\ d \in \ [0,1] \end{array}$
Operation:	(f) \rightarrow (destination)
Status Affected:	Z
Description:	The contents of register f are moved to a destination dependant 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.

NOP	No Operation
Syntax:	[label] NOP
Operands:	None
Operation:	No operation
Status Affected:	None
Description:	No operation.

MOVLW	Move Literal to W
Syntax:	[<i>label</i>] MOVLW k
Operands:	$0 \le k \le 255$
Operation:	$k \rightarrow (W)$
Status Affected:	None
Description:	The eight-bit literal 'k' is loaded into W register. The don't cares will assemble as 0's.

RETFIE	Return from Interrupt
Syntax:	[label] RETFIE
Operands:	None
Operation:	$TOS \rightarrow PC$, 1 $\rightarrow GIE$
Status Affected:	None

MOVWF	Move W to f
Syntax:	[label] MOVWF f
Operands:	$0 \le f \le 127$
Operation:	$(W) \to (f)$
Status Affected:	None
Description:	Move data from W register to register 'f'.

RETLW	Return with Literal in W
Syntax:	[<i>label</i>] RETLW k
Operands:	$0 \le k \le 255$
Operation:	$k \rightarrow (W);$ TOS $\rightarrow PC$
Status Affected:	None
Description:	The W register is loaded with the eight-bit literal 'k'. The program counter is loaded from the top of the stack (the return address). This is a two-cycle instruction.

14.4 MPLINK Object Linker/ MPLIB Object Librarian

The MPLINK object linker combines relocatable objects created by the MPASM assembler and the MPLAB C17 and MPLAB C18 C compilers. It can also link relocatable objects from pre-compiled libraries, using directives from a linker script.

The MPLIB object librarian is a librarian for precompiled code to be used with the MPLINK object linker. When a routine from a library is called from another source file, only the modules that contain that routine will be linked in with the application. This allows large libraries to be used efficiently in many different applications. The MPLIB object librarian manages the creation and modification of library files.

The MPLINK object linker features include:

- Integration with MPASM assembler and MPLAB C17 and MPLAB C18 C compilers.
- Allows all memory areas to be defined as sections to provide link-time flexibility.

The MPLIB object librarian features include:

- Easier linking because single libraries can be included instead of many smaller files.
- Helps keep code maintainable by grouping related modules together.
- Allows libraries to be created and modules to be added, listed, replaced, deleted or extracted.

14.5 MPLAB SIM Software Simulator

The MPLAB SIM software simulator allows code development in a PC-hosted environment by simulating the PICmicro series microcontrollers on an instruction level. On any given instruction, the data areas can be examined or modified and stimuli can be applied from a file, or user-defined key press, to any of the pins. The execution can be performed in single step, execute until break, or trace mode.

The MPLAB SIM simulator fully supports symbolic debugging using the MPLAB C17 and the MPLAB C18 C compilers and the MPASM assembler. The software simulator offers the flexibility to develop and debug code outside of the laboratory environment, making it an excellent multiproject software development tool.

14.6 MPLAB ICE High Performance Universal In-Circuit Emulator with MPLAB IDE

The MPLAB ICE universal in-circuit emulator is intended to provide the product development engineer with a complete microcontroller design tool set for PICmicro microcontrollers (MCUs). Software control of the MPLAB ICE in-circuit emulator is provided by the MPLAB Integrated Development Environment (IDE), which allows editing, building, downloading and source debugging from a single environment.

The MPLAB ICE 2000 is a full-featured emulator system with enhanced trace, trigger and data monitoring features. Interchangeable processor modules allow the system to be easily reconfigured for emulation of different processors. The universal architecture of the MPLAB ICE in-circuit emulator allows expansion to support new PICmicro microcontrollers.

The MPLAB ICE in-circuit emulator system has been designed as a real-time emulation system, with advanced features that are generally found on more expensive development tools. The PC platform and Microsoft[®] Windows environment were chosen to best make these features available to you, the end user.

14.7 ICEPIC In-Circuit Emulator

The ICEPIC low cost, in-circuit emulator is a solution for the Microchip Technology PIC16C5X, PIC16C6X, PIC16C7X and PIC16CXXX families of 8-bit One-Time-Programmable (OTP) microcontrollers. The modular system can support different subsets of PIC16C5X or PIC16CXXX products through the use of interchangeable personality modules, or daughter boards. The emulator is capable of emulating without target application circuitry being present. NOTES:

15.2 **DC Characteristics:** PIC16F73/74/76/77 (Industrial, Extended) PIC16LF73/74/76/77 (Industrial)

DC CHARACTERISTICS		$\begin{array}{llllllllllllllllllllllllllllllllllll$					
Param No.	Sym	Characteristic	Min	Тур†	Мах	Units	Conditions
	VIL	Input Low Voltage					
		I/O ports:					
D030		with TTL buffer	Vss	—	0.15Vdd	V	For entire VDD range
D030A			Vss	—	0.8V	V	$4.5V \le VDD \le 5.5V$
D031		with Schmitt Trigger buffer	Vss	—	0.2Vdd	V	
D032		MCLR, OSC1 (in RC mode)	Vss	_	0.2Vdd	V	(Note 1)
D033		OSC1 (in XT and LP mode)	Vss	—	0.3V	V	
		OSC1 (in HS mode)	Vss	—	0.3Vdd	V	
	Viн	Input High Voltage					
		I/O ports:					
D040		with TTL buffer	2.0	—	Vdd	V	$4.5V \le VDD \le 5.5V$
D040A			0.25Vdd + 0.8V	—	Vdd	V	For entire VDD range
D041		with Schmitt Trigger buffer	0.8Vdd	—	Vdd	V	For entire VDD range
D042		MCLR	0.8Vdd	—	Vdd	V	
D042A		OSC1 (in XT and LP mode)	1.6V	—	Vdd	V	
		OSC1 (in HS mode)	0.7Vdd	_	Vdd	V	
D043		OSC1 (in RC mode)	0.9Vdd	—	Vdd	V	(Note 1)
D070	Ipurb	PORTB Weak Pull-up Current	50	250	400	μA	VDD = 5V, VPIN = VSS
	lı∟	Input Leakage Current (Notes 2	2, 3)				
D060		I/O ports	—	—	±1	μA	Vss ≤ VPIN ≤ VDD, pin at hi-impedance
D061		MCLR, RA4/T0CKI	—	—	±5	μA	$Vss \le VPIN \le VDD$
D063		OSC1			±5	μA	Vss \leq VPIN \leq VDD, XT, HS and LP osc configuration

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/CLKIN pin is a Schmitt Trigger input. It is not recommended that the PIC16F7X be driven with external clock in RC mode.

2: The leakage current on the MCLR pin is strongly dependent on the 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 current sourced by the pin.

Param. No.	Symbol	Characte	eristic	Min	Max	Units	Conditions
100*	Тнідн	Clock high time	100 kHz mode	4.0		μs	Device must operate at a minimum of 1.5 MHz
			400 kHz mode	0.6		μs	Device must operate at a minimum of 10 MHz
			SSP Module	1.5TCY	_		
101*	TLOW	Clock low time	100 kHz mode	4.7		μs	Device must operate at a minimum of 1.5 MHz
			400 kHz mode	1.3		μs	Device must operate at a minimum of 10 MHz
			SSP Module	1.5TCY	—		
102*	Tr	SDA and SCL rise	100 kHz mode	—	1000	ns	
		time	400 kHz mode	20 + 0.1Св	300	ns	CB is specified to be from 10 - 400 pF
103*	TF	SDA and SCL fall	100 kHz mode	—	300	ns	
		time	400 kHz mode	20 + 0.1Св	300	ns	CB is specified to be from 10 - 400 pF
90*	TSU:STA	START condition	100 kHz mode	4.7	_	μs	Only relevant for
		setup time	400 kHz mode	0.6		μs	Repeated START condition
91*	THD:STA	START condition	100 kHz mode	4.0	—	μs	After this period the first
		hold time	400 kHz mode	0.6	—	μs	clock pulse is generated
106*	THD:DAT	Data input hold time	100 kHz mode	0	—	ns	
			400 kHz mode	0	0.9	μs	
107*	TSU:DAT	Data input setup	100 kHz mode	250	—	ns	(Note 2)
		time	400 kHz mode	100	—	ns	
92*	Tsu:sto	STOP condition	100 kHz mode	4.7	—	μs	
		setup time	400 kHz mode	0.6	—	μs	
109*	ΤΑΑ	Output valid from	100 kHz mode	—	3500	ns	(Note 1)
		clock	400 kHz mode	—	—	ns	
110*	TBUF	Bus free time	100 kHz mode	4.7	—	μs	Time the bus must be free
			400 kHz mode	1.3	—	μs	before a new transmission can start
	Св	Bus capacitive loadir	ng	—	400	pF	

TABLE 15-9: I²C BUS DATA REQUIREMENTS

* These parameters are characterized but not tested.

Note 1: As a transmitter, the device must provide this internal minimum delay time to bridge the undefined region (min. 300 ns) of the falling edge of SCL to avoid unintended generation of START or STOP conditions.

2: A Fast mode (400 kHz) I²C bus device can be used in a Standard mode (100 kHz) I²C bus system, but the requirement TsU:DAT ≥ 250 ns must then be met. This will automatically be the case if the device does not stretch the LOW period of the SCL signal. If such a device does stretch the LOW period of the SCL signal, it must output the next data bit to the SDA line TR max. + TSU:DAT = 1000 + 250 = 1250 ns (according to the Standard mode I²C bus specification), before the SCL line is released.

PIC16F7X



FIGURE 16-15: TYPICAL, MINIMUM AND MAXIMUM VOH vs. IOH (VDD = 5V, -40°C TO 125°C)





INDEX

	-	
1	Δ	۱
		•

A/D Co	onversion Status (GO/DONE Bit)	
Acquis	ition Requirements	
ADCO	N0 Register	
ADCO	N1 Register	
ADRES	S Register	
Analog	Port Pins8, 10	, 12, 39
Analog	J-to-Digital Converter	
Associ	ated Registers	
Config	uring Analog Port Pins	
Config	uring the Interrupt	
Config	uring the Module	
Conve	rsion Clock	
Conve	rsion Requirements	139
Conve	rsions	
Conve	rter Characteristics	138
Effects	of a RESET	
Faster	Conversion - Lower Resolution	
	Trade-off	
Interna	al Sampling Switch (Rss) Impedance	
Operat	tion During SLEEP	
Source	Impedance	
Using t	the CCP Trigger	
Absolute Ma	aximum Ratings	119
ACK Pulse		65, 66
ADCON0 R	egister	
GO/DC	ONE Bit	
ADCON1 R	egister	
ADRES Reg	gister	
Analog Port	Pins. See A/D	
Application	Notes	
AN552	(Implementing Wake-up on Key Strokes	
AN552	(Implementing Wake-up on Key Strokes Using PIC16F7X)	
AN552 AN556	(Implementing Wake-up on Key Strokes Using PIC16F7X)	33 26
AN552 AN556 AN578	2 (Implementing Wake-up on Key Strokes Using PIC16F7X) 6 (Implementing a Table Read) 9 (Use of the SSP Module in the I ² C	
AN552 AN556 AN578	 Implementing Wake-up on Key Strokes Using PIC16F7X) Implementing a Table Read) Use of the SSP Module in the I²C Multi-Master Environment) 	33 26 59
AN552 AN556 AN578 AN607	 Implementing Wake-up on Key Strokes Using PIC16F7X) (Implementing a Table Read) (Use of the SSP Module in the I²C Multi-Master Environment) (Power-up Trouble Shooting) 	33 26 59 94
AN552 AN556 AN578 AN607 Assembler	 Implementing Wake-up on Key Strokes Using PIC16F7X) (Implementing a Table Read) (Use of the SSP Module in the I²C Multi-Master Environment) (Power-up Trouble Shooting) 	
AN552 AN556 AN578 AN607 Assembler MPASI	 Implementing Wake-up on Key Strokes Using PIC16F7X) (Implementing a Table Read) (Use of the SSP Module in the I²C Multi-Master Environment) (Power-up Trouble Shooting) M Assembler 	
AN552 AN556 AN578 AN607 Assembler MPASI B	 Implementing Wake-up on Key Strokes Using PIC16F7X) (Implementing a Table Read) (Use of the SSP Module in the I²C Multi-Master Environment) (Power-up Trouble Shooting) M Assembler 	
AN552 AN556 AN578 AN607 Assembler MPASI B	 Implementing Wake-up on Key Strokes Using PIC16F7X) (Implementing a Table Read) (Use of the SSP Module in the I²C Multi-Master Environment) (Power-up Trouble Shooting) M Assembler 	
AN552 AN556 AN578 AN607 Assembler MPASI B Banking, Da	 Implementing Wake-up on Key Strokes Using PIC16F7X) (Implementing a Table Read) (Use of the SSP Module in the I²C Multi-Master Environment) (Power-up Trouble Shooting) M Assembler ata Memory 	
AN552 AN556 AN578 AN607 Assembler MPASI B Banking, Da BF bit	(Implementing Wake-up on Key Strokes Using PIC16F7X)	
AN552 AN556 AN578 AN607 Assembler MPASI B Banking, Da BF bit Block Diagra	(Implementing Wake-up on Key Strokes Using PIC16F7X)	
AN552 AN556 AN578 AN607 Assembler MPASI B Banking, Da BF bit Block Diagra A/D	(Implementing Wake-up on Key Strokes Using PIC16F7X)	
AN552 AN556 AN578 AN607 Assembler MPASI B Banking, Da BF bit Block Diagra A/D Analog	(Implementing Wake-up on Key Strokes Using PIC16F7X)	
AN552 AN556 AN578 AN607 Assembler MPASI B Banking, Da BF bit Block Diagra A/D Analog Captur	(Implementing Wake-up on Key Strokes Using PIC16F7X)	
AN552 AN556 AN578 AN607 Assembler MPASI B Banking, Da BF bit Block Diagra A/D Analog Captur Compa	(Implementing Wake-up on Key Strokes Using PIC16F7X)	
AN552 AN556 AN578 AN607 Assembler MPASI B Banking, Da BF bit Block Diagra A/D Analog Captur Corpsa	(Implementing Wake-up on Key Strokes Using PIC16F7X)	
AN552 AN556 AN578 AN607 Assembler MPASI B Banking, Da BF bit Block Diagra A/D Analog Captur Corpstal		
AN552 AN556 AN578 AN607 Assembler MPASI B Banking, Da BF bit Block Diagra A/D Analog Captur Corpsa Crystal Externa		
AN552 AN556 AN578 AN607 Assembler MPASI B Banking, Da BF bit Block Diagra A/D Analog Captur Compa Crystal Externa		
AN552 AN556 AN578 AN607 Assembler MPASI B Banking, Da BF bit Block Diagra A/D Analog Captur Compa Crystal Externa		
AN552 AN556 AN578 AN607 Assembler MPASI B Banking, Da BF bit Block Diagra A/D Analog Captur Compa Crystal Externa Interrup PIC166		
AN552 AN556 AN578 AN607 Assembler MPASI B Banking, Da BF bit Block Diagra A/D Analog Captur Compa Crystal Externa Interrup PIC166 PIC166		
AN552 AN556 AN578 AN607 Assembler MPASI B Banking, Da BF bit Block Diagra A/D Analog Captur Compa Crystal Externa Interrup PIC166 PIC166 PIC167		
AN552 AN556 AN578 AN607 Assembler MPASI B Banking, Da BF bit Block Diagra A/D Analog Captur Corpstal Externa Interrup PIC166 PIC166 PORT/ R		
AN552 AN556 AN578 AN607 Assembler MPASI B Banking, Da BF bit Block Diagra A/D Analog Captur Compa Crystal Externa Interrup PIC166 PIC166 PIC166 PORT/ R		
AN552 AN556 AN578 AN607 Assembler MPASI B Banking, Da BF bit Block Diagra A/D Analog Captur Compa Crystal Externa Interruu PIC166 PIC166 PIC166 PORT/ R R PORT/		
AN552 AN556 AN578 AN607 Assembler MPASI B Banking, Da BF bit Block Diagra A/D Analog Captur Compa Crystal Externa Interrup PIC166 PIC166 PIC166 PIC166 PORT/ R R PORTR	 Implementing Wake-up on Key Strokes Using PIC16F7X) (Implementing a Table Read) (Use of the SSP Module in the I²C Multi-Master Environment) (Power-up Trouble Shooting) M Assembler M Assembler M Assembler ata Memory ata Memory ata Memory ams Input Model e Mode Operation are //Ceramic Resonator Operation (HS, XT or LP Osc Configuration) al Clock Input Operation (HS Osc Configuration) pt Logic F73 and PIC16F76 F74 and PIC16F77 A A3:RA0 and RA5 Port Pins B3:RB0 Port Pins B7:PB4 Port Pins 	

С

Capture/Compare/PWM (CCP)		
Associated Registers	56	58
Capture Mode	,	55
Prescaler		55
CCP Pin Configuration	55	56
	.55,	50
	٥	11
CCP2		
	a	11
Compare Mode	,	55
Software Interrunt Mode		56
Special Trigger Output		56
Timer1 Mode Selection		56
Example PWM Frequencies and Resolutions		58
Interaction of Two CCP Modules		53
PWM Duty Cycle		57
PW/M Mode		57
PW/M Period		57
Setup for PWM Operation		58
Special Event Trigger and A/D Conversions		56
Timer Resources		53
CCP1 Module		53
CCP2 Module		53
		53
CCPR1L Register		53
CCPvM-3:0> hits		54
CCPvY and CCPvV hite		54
CKE bit		60
CKP bit		61
Code Examples		01
Call of a Subroutine in Page 1 from Page 0		26
Changing Between Capture Prescalers		55
Changing Prescaler Assignment to Timer()		45
Changing Prescaler Assignment to WDT		45
FLASH Program Read		30
Indirect Addressing		27
		21
Reading a 16-bit Free-Rupping Timer		10
Saving STATUS W/ and PCI ATH Pagistors		49
in PAM	4	100
Writing a 16-bit Free-Punning Timer		100
winning a to-bit rice-routining timel	•••••	49

PORTC (Peripheral Output Override)35