

The PIC 16F874 Timers

There are a total of four timers that are available in the PIC. These are:

The Watchdog timer (WDT)

The Watchdog timer is a safety device. It has it's own independent RC oscillator and, , when enabled it times out after 7 to 33ms. If it is allowed to timeout, it resets the PIC by performing a watchdog timer reset.

Timer 0 (TMR0)

An 8-bit counter/timer which can be driven by either an internal or external clock. A prescaler allows dividing the input clock using 8 selectable divisors. Some parts of Timer 0 are shared with the Watchdog Timer.

Timer 1 (TMR1)

A 16 bit counter/timer with 4 selectable prescaler divisors. Timer 1 can be driven by an internal or external clock and has the ability to synchronize an external clock to the internal PIC clock.

Timer 2 (TMR2)

An 8 bit counter/timer with a prescaler and a postscaler. The timer rollover point is programmable. This timer may be used to drive the baud clock for serial output.

Timer 0 And The WDT

By inspecting the block diagram of Timer 0, we can see the features that are available. We can use both Timer 0 and the Watchdog concurrently, but there are restrictions!



The Timer 1 Block Diagram

Timer 1 provides a 16-bit register for creating larger counts or divisors than Timer 0. There are similarities between the block diagrams, but as we can see, the options are different for Timer 1. Again, we can only figure this out from the block diagram!



EE2801-L23P04

Timer 2 Characteristics

You should be starting to see a trend. All of these timers provide similar features with slightly different variations. As you might expect, setting up any of these timers is very similar.



The Timer 1 Control Register (T1CON)

Each timer has a control register that is used to setup the behavior of the timer. Referring to the timer block diagram, you can see visually what paths are being set when you put values into a timer control register. In addition to the control register, Timer 1 also has a low and high byte register to set the count (TMR1H and TMR1L).

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			_			
ó	ó	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR10N	R	= Readable bit				
bit7							bit0	U	= vvritable bit = Unimplemented bit,				
									read as ë0i				
	llusiususla	un antad. D						- n	= Value at POR reset				
DIT 7-0	-6: Unimplemented: Read as '0'												
bit 5-4.	T1CKPS	I 1CKPS1: I 1CKPS0: Timer1 Input Clock Prescale Select bits											
	11 - 1.0 10 = 1.4	Prescale va	alue										
	01 = 1 :2	Prescale va	alue										
	00 = 1:1	Prescale va	alue										
bit 3:	T1OSCE	T1OSCEN: Timer1 Oscillator Enable Control bit											
	1 = Oscillator is enabled 0 = Oscillator is shut off (The oscillator inverter is turned off to eliminate power drain)												
bit 2 [.]	$\overline{0} = Costinator is shut on (the ostinator inverter is turned on to enfinite power drain)$												
Dit Z.													
	<u>TMR1CS = 1</u>												
	1 = Do not synchronize external clock input												
	0 – Sync			Kiliput									
	<u>TMR1CS = 0</u>												
	This bit is	s ignored. T	limer1 use	es the inter	nal clock v	vhen TMR	1CS = 0.						
bit 1:	TMR1CS	5: Timer1 C	lock Sour	ce Select b	it The second								
	1 = Exte	rnal clock fr nal clock (E	om pin Ru Tosc/4)	20/11080	/11CKI (0	n the rising	(edge)						
bit 0		V Timer1 C)n hit										
SIL U.	1 = Enat	oles Timer1											
	0 = Stop	s Timer1											
										EE2001-L23P00			

The Other Registers Associated With Timer 0

From the PIC datasheet, we also can find out exactly what other registers are associated with the operation of Timer 1 (which we might have to worry about):

TABLE 6-2:REGISTERS ASSOCIATED WITH TIMER1 AS A TIMER/COUNTER

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit O	Value on: POR, BOR	n: Value on all ot her resets	
0Bh,8Bh, 10Bh, 18Bh	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	0000 000u	
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	
0Eh	TMR1L	Holding re	gister for		XXXX XXXX	uuuu uuuu						
0Fh	TMR1H	Holding re	gister for		XXXX XXXX	uuuu uuuu						
10h	T1CON	ó	ó	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR10N	00 0000	uu uuuu	

Legend: x = unknown, u = unchanged, - = unimplemented read as '0'. Shaded cells are not used by the Timer1 module.

Note 1: Bits PSPIE and PSPIF are reserved on the PIC16F873/874; always maintain these bits clear.

An Example Of Using Timer 1

```
; PIC example program to setup timer 1. This routine will
; set the timer to count seconds (assuming a 4Mhz clock for
; Fosc). The seconds are displayed in binary on the Port D
; LEDs.
; Prof. W. Michalson, ECE Dept, WPI, February 2000
; tmr1.asm ; Version 1.0
include p16f877.inc
        equ 0x20 ;Counter variable
count
                 ; Start program at address 000
        org 0x000
                    ; Required for debugger
        nop
; Initialize PORT D to be all outputs.
       bsf STATUS, RP0 ; Go to BANK 1 by setting
Start
       bcf STATUS, RP1 ; RP1, RP0 = 01.
                 ; Set Port D for output.
        clrf TRISD
                      ; Go back to bank 0!
        bcf STATUS, RPO
        clrf PORTD
                       ; Write Os to Port D.
        clrf count ; Initialize count to 0.
```

An Example Of Using Timer 1 (Continued)

```
; Setup timer 1
         bcf T1CON, TMR1ON ; Turn Timer 1 off.
         bsf T1CON, T1CKPS1 ; Set prescaler for divide
         bsf T1CON, T1CKPS0 ; by 8.
         bcf T1CON, T1OSCEN ; Disable the RC oscillator.
         bcf T1CON, TMR1CS ; Use the Fosc/4 source.
         clrf TMR1L
                           ; Start timer at 0000h
         clrf TMR1H
         bsf T1CON, TMR1ON ; Start the timer
; Wait in a loop until the timer finishes
time1
         btfss PIR1,0 ; Did timer overflow?
         goto time1
                           ; Wait if not.
;Timer overflowed, increment counter and display
                         ; Clear the flag
         bcf PIR1,0
         incf count, F ; Bump the counter
         movf count,W
                           ; Get the count
         movwf PORTD
                           ; Send to Port D.
         bcf T1CON,0
                           ; Turn the timer off.
                            ; Start timer at 8000h
         movlw 80h
         movwf TMR1H
         clrf TMR1L
         bsf T1CON,0
                        : Turn the timer on.
         qoto time1
         End
```