

CURSO DE MICROCONTROLADORES PIC16F87X (...y V)

Fernando Remiro Domínguez
 Profesor de Sistemas Electrónicos
 IES. Juan de la Cierva
www.terra.es/personall/fremiro

MODULO TIMER0

El módulo TIMER0 es un temporizador/contador con las siguientes características:

- El temporizador / contador dispone de 8 bits
- Puede escribirse y leerse
- Preescaler programable por software de 8 bits
- Puede trabajar con el reloj interno o con una señal de reloj externa
- Dispone de una interrupción por desbordamiento al pasar de FFh a 00h
- Selección de flanco ascendente o descendente para el flanco del reloj externo

En la Figura 1 se muestra un diagrama de bloques del TIMER0 y el preescaler que comparte con el WDT. Puede encontrarse información adicional sobre el módulo Timer0 en el manual

de referencia de los microcontroladores de la escala media (DS33023).

El modo temporizador se selecciona poniendo a cero el bit T0CS (registro OPTION_REG <5>). En el modo temporizador, el módulo Timer0 se incrementa en cada ciclo de instrucción (sin el preescaler). Si el registro TMR0 se escribe, el incremento se inhibe durante los siguientes dos ciclos de instrucción. EL usuario puede trabajar teniendo en cuenta esto y ajustando el valor a cargar en el TMR0.

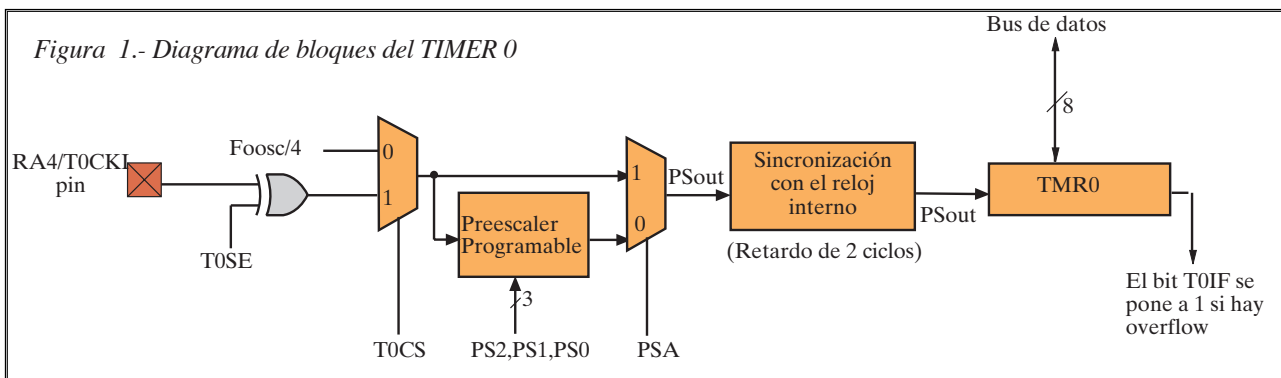
El modo contador se selecciona poniendo a uno el bit T0CS (registro OPTION_REG <5>). El modo contador, Timer0 se incrementa en cada flanco de subida o de bajada de la señal que le llega por RA4/T0CK1. El flanco de incremento se determina por el bit T0SE (registro OPTION_REG

<4>). Poniéndose a cero T0SE se selecciona el flanco ascendente. El preescaler se comparte exclusivamente entre el TIMER0 y el WDT. Y además no es de lectura/escritura.

INTERRUPCIÓN DEL TIMER0

La interrupción de TMR0 se produce cuando el registro TMR0 se desborda al pasar de FFh a 00h. Este desbordamiento pone a uno el bit T0IF (INTCON<2>). La interrupción puede enmascarse poniendo a cero el bit T0IE (INTCON <5>). EL bit T0IF debe ponerse a cero por software al finalizar la rutina de atención a la interrupción del desbordamiento de TMR0. La interrupción de TMR0 no saca al microcontrolador del estado de SLEEP, debido a que el temporizador está desactivado durante el modo SLEEP

Figura 1.- Diagrama de bloques del TIMER 0



Notas:

1. El preescaler es compartido con el WDT (En la Figura 2 se puede ver un diagrama de bloques detallado)
2. T0CS, T0SE, PSA, PS2:PS0 (OPTION_REG<5:0>)

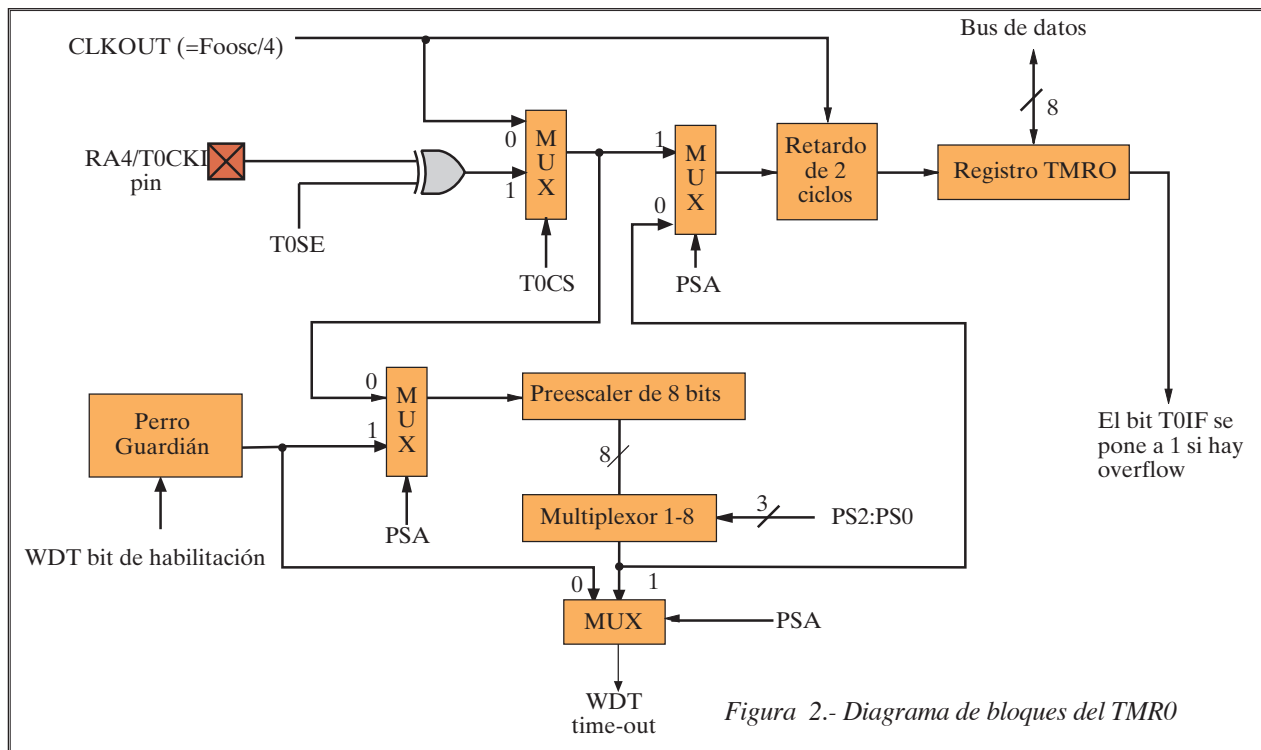


Figura 2.- Diagrama de bloques del TMR0

TABLA DE REGISTROS ASOCIADOS AL TIMER0											
Dirección	Nombre	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Valor en POR,BOR	Valor en el resto de Reset
01h, 101h	TMR0	Módulo del registro del TMR0								xxxx xxxx	uuuu uuuu
0bh, 8Bh											
10Bh, 18Bh	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	0000 000x
81h, 181h	OPTION_REG	RBPU	INIEDG	TOCS	TOSE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
85h	TRISA	---	---	Registro de direccionamiento de datos del PORTA				--11 1111	--11 1111		

Leyenda: x = desconocido, u = inalterado; - = no implementado se lee como '0'. Las celdas sombreadas no son usadas por el TIMER0

UTILIZACIÓN DEL TIMER0 CON UN RELOJ EXTERNO

Cuando no se utiliza el preescaler, la entrada de reloj externa es igual a la salida del preescaler. La sincronización de TOCKI con los relojes de fase interior se acopla, a la salida del preescaler en los ciclos Q2 y Q4 de los relojes de fase internos. Por consiguiente, es necesario que TOCKI está a nivel alto por al menos durante 2Tosc (y un pequeño retardo de 20ns) y a nivel bajo por lo menos 2Tosc (y un retardo RC de 20ns).

PREESCALER

Hay sólo un preescaler disponible que está compartido y

puede asignarse indistintamente al módulo de TIMER0 y el al WDT. La asignación del preescaler al Timer0 hace que no haya ningún preescaler para el WDT, y viceversa. Este preescaler no se puede leer ni escribir (ver la Figura 2).

El bit PSA y PS2:PS0 (OPTION_REG <3:0>) determinan la asignación del preescaler y el rango del preescaler. Cuando se le asigna al módulo del Timer0, todas las instrucciones, que escriben en el registro TMR0 (por ejemplo CLRWF TMR0, MOVWF TMR0, BSF TMR0,x... etc.) ponen a cero el preescaler. Cuando se le asigna al WDT, una instrucción CLRWDT limpia el preescaler junto con el temporizador del

WDT. EL preescaler no se puede leer ni escribir.

Nota.- Escribir en TMR0, cuando el preescaler es asignado a TIMER0, limpia la cuenta del preescaler, pero no cambia la configuración del preescaler.

PROGRAMA EJEMPLO DE FUNCIONAMIENTO DEL TIMER0

Para comprobar el funcionamiento de este programa debemos de montar un circuito como el que se muestra en la Figura 3, en el que se ha conectado un display del tipo cátodo común al PORTB de PIC16F876. Una vez cargado el programa TMR0_2.asm, en el display se irán representado

MICROCONTROLADORES


```

*****
: Programa TMR0_2.ASM                               Fecha : 15 - Abril - 2003
: Estè programa presenta en el display de cátodo común conectado en el PORTB
: los números del 0 al 9, mostrando un dato por segundo (aproximadamente), la
: temporización la realiza utilizando la interrupción por TMR0 y el registro auxiliar CONTA
: Revisión : 0.0                                     Programa para PIC16F87X
: Velocidad del Reloj: 4 MHz                         Reloj Instrucción: 1 MHz = 1 uS
: Perro Guardián : deshabilitado                    Tipo de Reloj : XT
: Protección del código : OFF
*****
List      p=16F876      ;Tipo de procesador
include   "P16F876.INC" ;Definiciones de registros internos
*****
DEFINICION DE VARIABLES *****
W_TEMPORAL equ 0x20      ;Almacena W antes de atender la interrupción
S_TEMPORAL equ 0x21      ;Almacena Status antes de atender la interrupción
DIGITO     equ 0x22      ;numero a representar en display
CONTA      equ 0x23      ;Contador auxiliar para alargar el tiempo a 1 segundo
                        ;entre cada cambio de dígito, ya que los 256 (FF) del Timer,
                        ;aunque se divida por 256, es  $4 \times 256 \times 256 = 262,144$  ms
*****
:
:          ORG 0x00
:          goto INICIO
:          ORG 0x04
*****
***** ATENCION A LA INTERRUPCION *****
INT_TMR0
movwf    W_TEMPORAL      ;Guarda el valor actual de W cuando se interrumpe
swapf    STATUS,W        ;Intercambia nibbles y guarda en W
movwf    S_TEMPORAL      ;Almacena Status pero cambiado el ..
                        ;.. nibble alto por el bajo
clrf     TMR0             ;Inicializa TMR0=00
decfsz   CONTA,f         ;Decrementa CONTA y si =0 incrementa DIGITO
goto     FIN_INT         ;Si cont_aux <> 0 finaliza interrupción
incf     DIGITO,F        ;Incrementa el DIGITO
movlw    D'16'           ;Al cargar TMR0 con 0 hace 256 cuentas con un predivisor de 256
                        ;(programado en OPTION_REG). Como queremos 1sg. ponemos 16 en CONTA
                        ; así obtenemos  $4 \times 256 \times 256 \times 16 = 1,048576$ s
movwf    CONTA           ;inicializa de nuevo CONTA para un nuevo dígito

FIN_INT
swapf    S_TEMPORAL,W    ;Deja Status como estaba antes y en W
movwf    STATUS
swapf    W_TEMPORAL,F    ;Intercambia nibbles del W guardado
swapf    W_TEMPORAL,W    ;Vuelve a intercambiar pero ya en W
bcf     INTCON,TOIF      ;Borra el flag de la interrupción
retfie   ;Retorno de interrupción
*****
***** PROGRAMA PRINCIPAL *****
INICIO
bcf     STATUS,RP0       ;Habilita Banco 1 de registros
bsf     STATUS,RP0
clrf    TRISB            ;direcciona PortB como salidas
movlw   B'00000111'      ;Reloj interno asignado aTMR0 con divisor por 256
movwf   OPTION_REG
bsf     INTCON,GIE       ;habilitación general de las interrupciones
bsf     INTCON,TOIE      ;habilitación de la interrupción por TMR0
bcf     STATUS,RP0       ;RP0=0 habilita Banco0 de registros
clrf    TMR0             ;Inicializa TMR0 a 0
movlw   D'16'           ;Carga CONTA con 16
movwf   CONTA

PONER_0
clrf    DIGITO           ;Carga DIGITO a 0 para iniciar cuenta

LEETABLA
call    TABLA            ;Muestra dígito en el display 7 segmentos (PortB)
movwf   PORTB
movlw   D'9'             ;
subwf   DIGITO,W         ;Compara DIGITO con 9
btfsz   STATUS,Z         ;
goto    LEETABLA        ;Si DIGITO NO = 9, incrementa DIGITO
goto    PONER_0         ;Si DIGITO=9, lo pone a 0 de nuevo.

*****
***** TABLA CON LOS VALORES DE LOS DIGITOS *****
TABLA
movf    DIGITO,W         ;Guarda dígito en acumulador
addwf   PCL,F           ;Suma al contador de programa el dígito
DT 0X3F,0X06,0X5B,0X4F,0X66,0X6D,0X7D,0X07,0X67
*****
:
:          ORG 0x1F00
:          bcf PCLATH,4
:          bcf PCLATH,3
:          goto INICIO
:          ;Selecciona la página 0
:          ;Salto a la dirección de INICIO del programa
*****
:          END

```

REGISTRO T1CON: Registro de control del TIMER1 (dirección 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	TMR1ON
Bit 7							Bit 0

bit 7-6: No implementados: Se lee como "0"

bit 5-4: TICKPS1:TICKPS0: bit de selección del preescaler de la señal de reloj del TIMER1
 11 = valor del preescaler 1:8
 10 = valor del preescaler 1:4
 01 = valor del preescaler 1:2
 00 = valor del preescaler 1: 1

bit 3 : T1OSCEN: bit de habilitación del oscilador del TIMER1. Cuando se emplea un oscilador externo, hay que poner este bit a 1. El TMR1 puede trabajar a una frecuencia totalmente independiente de la del sistema.
 1 = Habilita el oscilador
 0 = Deshabilita el oscilador
 Nota.- El oscilador y la resistencia de desconectan para reducir el consumo

bit 2: #T1SYNC: bit de control de sincronización de la señal de entrada.
 Con TMR1CS = 1
 1= No sincroniza la entrada de reloj externa
 0 = Sincroniza la entrada de reloj externa
 Con TMR1CS = 0
 En esta condición se ignora. El TIMER1 utiliza el reloj interno cuando TMRICS=0

bit 1 TMR1CS: bit de selección de la fuente de reloj del TIMER1
 1 = Reloj externo por el pin RC0/T1OSO/T1CK1 (flanco ascendente)
 0 = Reloj interno (FOSC/4)

bit 0: TMR1ON: TIMER1. activo. Hace entrar o no en funcionamiento el TIMER1.
 1 = Habilita el TIMER1
 0 = Deshabilita el TIMER1

El modo de trabajo viene determinado por la fuente de los impulsos de reloj, es decir, la señal de reloj puede ser externa o interna, se selecciona con el bit TMR1CS del registro T1CON<1>, cuando este bit está a nivel bajo se selecciona el reloj el interno del micro-controlador (Fosc/4) y cuando está a uno se selecciona el modo contador y cuenta los impulsos que le llegan a través del pin RC0/TICK1. Además como se verá mas adelante el TIMER1 tiene la posibilidad de reinicializarse, a partir del módulo CCP.

MODO DE FUNCIONAMIENTO DEL TIMER1 COMO TEMPORIZADOR

Este modo se selecciona poniendo a cero el bit TMR1CS (T1CON <1>). En este modo la señal de reloj es el reloj interno del microcontrolador FOSC/4. En este modo de trabajo el bit T1SYNC (T1CON <2>) no tiene ningún

efecto ya que el reloj interno está siempre sincronizado.

MODO DE FUNCIONAMIENTO DEL TIME1 COMO CONTADOR

En este modo puede trabajar como contador síncrono o asíncrono . Cuando el TIMER1 se está incrementando según le llegan los impulsos externos, los incrementos ocurren en los flancos de subida. Después de que el TIMER1 se ha configurado como contador, debe producirse un flanco de bajada antes de empezar a contar.

MODO DE FUNCIONAMIENTO DEL TIMER1 COMO CONTADOR SÍNCRONO

Para seleccionar este modo se pone a uno el bit TMR1CS (T1CON <1>). En este modo el contador se incrementa en cada flanco ascendente de la señal de reloj que se introduce por el pin RC0/T1OSO/TICK1 cuando el bit

T1OSCEN está a uno, y por el pin RC1/TIOSI/CCP2, cuando el bit T1OSCEN está a cero.

Si T1SYNC se pone a cero, entonces la entrada de reloj externa se sincroniza con los relojes de fase interiores. La sincronización se hace después de la fase del preescaler. En el preescaler la fase de la señal de reloj es por lo tanto asíncrona.

En este modo de trabajo, durante el modo SLEEP el TIMER1 no se incrementa aún cuando la señal de reloj externa esté presente. El preescaler sin embargo continua incrementándose.

MODO DE FUNCIONAMIENTO DEL TIMER1 COMO CONTADOR ASÍNCRONO

Cuando el bit de control T1SYNC (T1CON <2>) se pone a uno, la señal de reloj externa no se sincroniza. El contador sigue realizando la cuenta de forma asíncrona respecto a la fase de la señal de reloj interna. El contador

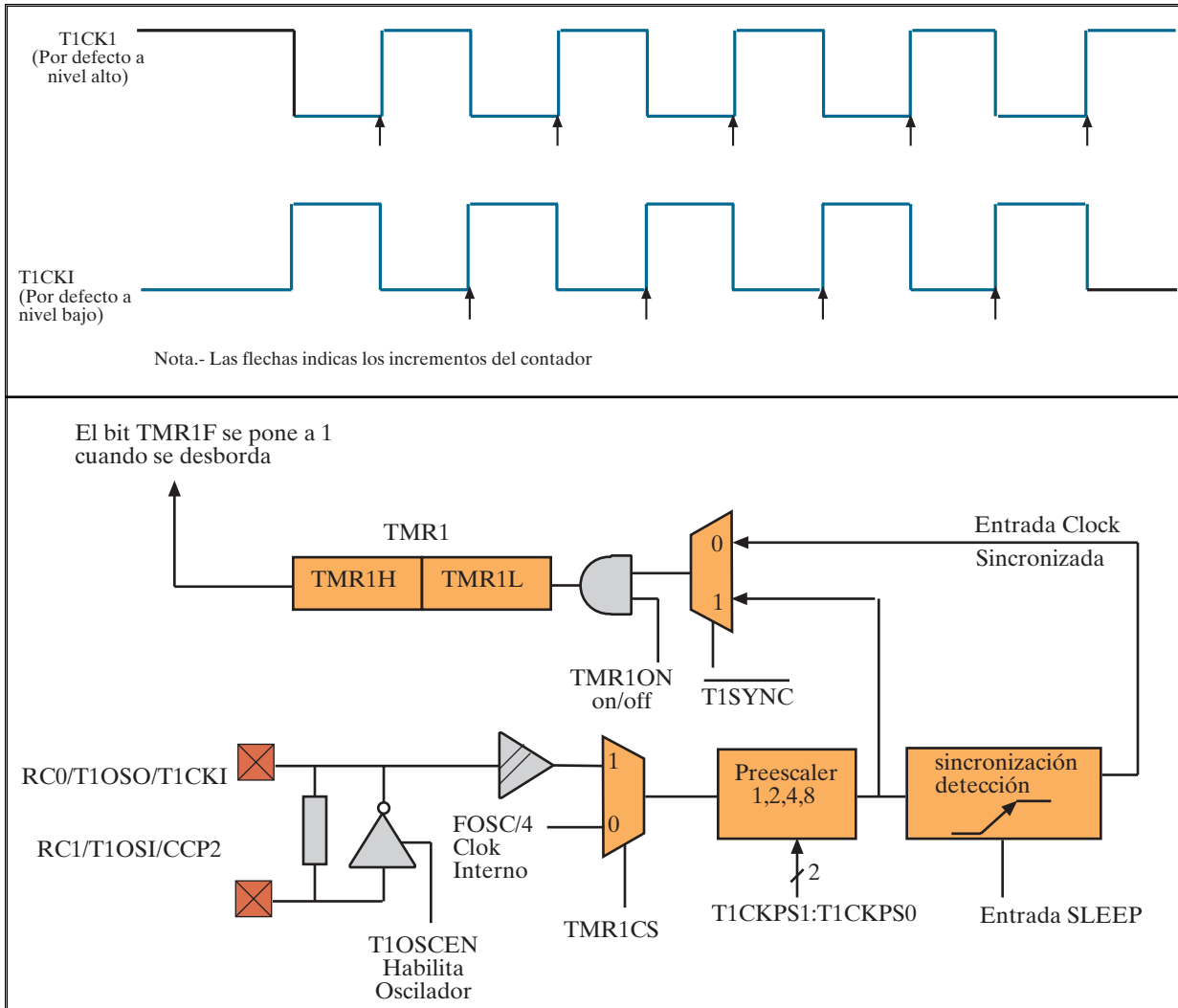


Figura 5.- Diagrama de bloques de TIMER1

Tipo de Oscilador	Frecuencia	C1	C2
LP	32 kHz	33pF	33pF
	100kHz	15 pF	15 pF
	200	15 pF	15 pF
Estos valores son para los siguientes cristales de referencia			
32.768	EpsonC-001R21.768K-A ±20 PPM		
100 kHz	Epson C-2 100.000 KC-p1 ±20 PPM		
200 kHz	STD XTL 200.000 kHz ±20 PPM		
Nota.-			
1. Capacidades mayores aumentan la estabilidad del oscilador, pero también aumentan el start-up.			
2. Cada cristal/resonador tiene sus propias características, el usuario debe consultar los valores apropiados de los componentes externos al cristal.			

Tabla 1.- Selección de los condensadores para el oscilador del TIMER1

continúa la cuenta incluso en el modo SLEEP y puede generar una interrupción por desbordamiento que despierta al procesador. Hay que tener especial cuidado con el software al leer o escribir el contador.

Cuando se trabaja en el modo contador asíncrono, el TIMER1 no puede usarse como base de tiempos

para el módulo CCP (Captura y comparación-PWM)

LECTURA Y ESCRITURA EN EL TIMER1 CUANDO SE TRABAJA EN EL MODO CONTADOR ASÍNCRONO

Se pueden leer los contadores TMR1H y TMR1L mientras la señal externa del contador se está

recibiendo (teniendo cuidado con el hardware). Sin embargo, el usuario debe tener en cuenta que el contador es de 16 bits y se pueden tener ciertos problemas al leer los dos registros de ocho bits, ya que el contador puede desbordarse entre las lecturas.

Para escribir en él, se recomienda que el usuario

simplemente pare el contador y escriba los valores deseados. Cuando se escribe el registro del contador puede haber conflicto mientras este se está incrementando. Esto puede producir un valor imprevisible en el contador.

OSCILADOR DEL TIMER1

Se puede conectar un oscilador a cristal entre los pines T1OSI (entrada) y T1OSO (salida del amplificador). Se debe habilitar poniendo a uno el bit de control TLOSCEN (TICON <3>). El oscilador de bajo consumo puede trabajar hasta 200 kHz. En estas condiciones el oscilador sigue funcionando aunque se fuerce el

(CP1CON<3:0>)), esta señal reestablecerá el TIMER1.

Nota.- El evento especial que activa los módulos CCP1 y CCP2 no activará el flag

TMR1IF (PIR1<0>).

El TIMER1 debe configurarse como temporizador o contador síncrono.

Restablecimiento TIMER1 que usa un disparador de CCP.

TIMER1 debe configurarse para temporizador o contador síncrono para aprovechar esta característica. Si el TIMER1 está trabajando en modo contador asíncrono, el reset no puede ser activado.

En el caso de escribir sobre el TIMER1 en el momento del

un disparo de evento especial en CCP1 y CCP2. El registro T1CON se inicializa a 00h después de un Power-on Reset o Brown-out Reset que deja a 1:1 el preescaler. En los demás reset, el registro no es alterado.

TIMER1 PRESCALER

El contador del preescaler se pone a cero al escribir en los registros TMR1H o TMR1L.

RESUMEN DE REGISTROS ASOCIADOS AL TIMER1

En la siguiente tabla se muestran los registros principales que controlan el comportamiento del TIMER1 y la distribución de los bit.

Dirección	Nombre	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Valor en POR,BOR	Valor en el resto de Reset
0Bh,8Bh 10Bh,18Bh	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTE	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
0Bh	PIE1	PSPIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
0Eh	TMR1L	Registro de carga del byte de menor peso del registro de 16 bits de TMR1								xxxx xxxx	uuuu uuuu
0Fh	TMR1H									xxxx xxxx	uuuu uuuu
10h	TICON	---	---	TICKPS1	TICKPS0	TLOSCEN	T1SYNC	TMR1CS	TMR1ON	--xx xxxx	--uu uuuu

Leyenda

x = desconocido, u = inalterado; - = no implementado se lee como '0'. Las celdas sombreadas no son usadas por el TIMER1

Nota 1: Los bits PSPIE y PSPIF están reservados para el PIC16F873/876, mantener estos bits a cero.

modo SLEEP. Está pensado para trabajar con un cristal de hasta 32 kHz. La Tabla 1 muestra el valor de los condensadores para el TIMER1.

El oscilador TIMER1 es idéntico al oscilador LP. El usuario debe proporcionar un tiempo de retardo por software para asegurar la salida apropiada del oscilador.

RESTABLECIMIENTO DEL TIMER1 UTILIZADO LA SALIDA TRIGGER DEL CCP

Si el módulo CCP1 o CCP2 se configuran en modo comparación para generar un "disparo por evento especial" (CCP1M3:CCP1M0 = 1011

disparo del evento especial de CCP1 o CCP2, se tomará el valor anterior.

En el caso de escribir sobre TIMER1 si coincide con un disparo por evento especial de CCP1 o CCP2, se carga el valor anterior.

En este modo de funcionamiento, el par de registros CCPRxH:CCPRxL se reponen para volver a tener el periodo del TIMER1.

RESTABLECIMIENTO DEL PAR DE REGISTRO DEL TIMER1 (TMR1H, TMR1L)

Los registros TMR1H y TMR1L no se inicializan a 00h después de un Power-on Reset o por cualquier otro reset excepto por

PROGRAMA EJEMPLO DE FUNCIONAMIENTO DEL TIMER1

Para comprobar el funcionamiento de este programa utilizaremos el circuito de la Figura 3 que es el mismo que hemos utilizado para comprobare el funcionamiento del TIMER0, en el que se ha conectado un display del tipo cátodo común al PORTB de PIC16F876. Una vez cargado el programa TMR1_0.asm, en el display se irán representado los números del 0 al 9 con una cadencia de aproximadamente un segundo.

```

*****
; Programa TMR1_0.ASM                               Fecha : 15 - Abril - 2003
; Este programa presenta en el display de cátodo común conectado en el PORTB
; los números del 0 al 9, mostrando un dato por segundo (aproximadamente), la
; temporización la realiza utilizando la interrupción por TMR1 y el registro auxiliar CONTA
; Revisión : 0.0                                     Programa para PIC16F87X
; Velocidad del Reloj: 4 MHz                         Reloj Instrucción: 1 MHz = 1 uS
; Perro Guardián : deshabilitado                   Tipo de Reloj : XT
; Protección del código : OFF
*****
;
; List p=16F876                                     ;Tipo de procesador
; include "P16F876.INC"                             ;Definiciones de registros internos
*****
; DEFINICION DE VARIABLES *****
W_TEMPORAL equ 0x20                                ;Almacena W antes de atender la interrupción
S_TEMPORAL equ 0x21                                ;Almacena Status antes de atender la interrupción
DIGITO equ 0x22                                     ;numero a representar en display
CONTA equ 0x23                                     ;Contador auxiliar para alargar el tiempo a 1 segundo
;entre cada cambio de digito, ya que los 256 (FF) del Timer,
; aunque se divida por 256, es 4x256x256=262,144 ms
*****
;
; ORG 0x00
; goto INICIO
; ORG 0x04
*****
; ATENCION A LA INTERRUPCION *****
INT_TMR1
movwf W_TEMPORAL ;Guarda el valor actual de W cuando se interrumpe
swapf STATUS,W ;Intercambia nibbles y guarda en W
movwf S_TEMPORAL ;Almacena Status pero cambiado el ..
; nibble alto por el bajo
movlw low ~.62500 ;Carga parte baja del contador con el complemento
movwf TMR1L ;de 62500 en hex.
movlw high ~.62500 ;Carga parte alta del contador con el complemento
movwf TMR1H ;de 62500 en hex.
bcf PIR1.0 ;Desconecta el flag del TMR1
incf PORTB,F ;Incrementa el contador de la Puerta B
decfsz CONTA,f ;Decrementa CONTA y si =0 incrementa DIGITO
goto FIN_INT ;Si cont_aux <> 0 finaliza interrupcion
incf DIGITO,F ;Incrementa el DIGITO
movlw D'2'
movwf CONTA ;inicializa de nuevo cont_aux para un nuevo digito

FIN_INT
swapf S_TEMPORAL,W ;Deja Status como estaba antes y en W
movwf STATUS
swapf W_TEMPORAL,F ;Intercambia nibbles del W guardado
swapf W_TEMPORAL,W ;Vuelve a intercambiar pero ya en W
bcf INTCON,T0IF ;Borra el flag de la interrupción
retfie

*****
; PROGRAMA PRINCIPAL *****
INICIO
bcf STATUS,RP0
bsf STATUS,RP0 ;Habilita Banco1 de registros
clrf TRISB ;direcciona PortB como salidas
movlw b'1111111'
movwf OPTION_REG ;Preescaler d 128 asociado al WDT
bsf PIE1.0 ;habilita interrupción del TMR1
bcf STATUS,RP0 ;Selecciona banco 0
;El TMR1 trabaja con oscilador interno y un preescaler de 1:8. Si se trabaja a una frecuencia
;de 4 MHz, el TMR1 deberá ser cargado con 62500 para que provoque interrupción al de 0.5s b

movlw b'00110001'
movwf TICON
movlw b'11000000'
movwf INTCON ;Habilitación global de interrupciones
movlw D'2' ;Carga CONTA con 2
movwf CONTA

PONER_0
clrf DIGITO ;Carga DIGITO a 0 para iniciar cuenta

LEETABLA
call TABLA
movwf PORTB ;Muestra digito en el display 7 segmentos (PortB)
movlw D'9'
subwf DIGITO,W ;Compara DIGITO con 9
btss STATUS,Z
goto LEETABLA ;Si DIGITO NO = 9, incrementa DIGITO
goto PONER_0 ;Si DIGITO=9, lo pone a 0 de nuevo.

*****
; TABLA CON LOS VALORES DE LOS DIGITOS *****
TABLA
movf DIGITO,W ;Guarda digito en acumulador
addwf PCL,F ;Suma al contador de programa el digito
DT 0x3F,0x06,0x5B,0x4F,0x66,0x6D,0x7D,0x07,0x67
*****
;
; ORG 0x1F00
; bcf PCLATH,4
; bcf PCLATH,3 ;Selecciona la página 0
; goto INICIO ;Salto a la dirección de INICIO del programa
*****
;
; END

```