

2.4 MODULACIÓN POR ANCHO DE PULSO “UNBUFFERED” – PWM

CONFIGURACIÓN DEL PWM “UNBUFFERED” Y SUS LIMITANTES

ÍNDICE

Preparado por: Rangel Alvarado
Estudiante Graduando de Lic. en Ing. Electromecánica
Universidad Tecnológica de Panamá
Panamá, Panamá
“e-mail”: issaiass@cwpanama.net
“web site”: <http://www.geocities.com/issaiass/>

2.4.1	Introducción	220
2.4.2	Características de una PWM tipo “Unbuffered”	221
2.4.3	Registros de Estado y Control de los Canales del TIM en PWM	221
2.4.4	Diagrama de Flujo	224
2.4.5	Código	226
2.4.6	Simulación	232
2.4.7	Conclusión	233
2.4.8	Referencias	234
2.4.9	Problemas Propuestos	234

2.4.1 Introducción

Una herramienta configurable e interesante del temporizador es la Modulación por Ancho de Pulso (PWM). *Cualquier canal del temporizador puede generar una PWM.* Esta técnica se utiliza para generar señales de período fijo y un ancho de pulso variable.

Uno de los usos más comunes para PWM es la reducción de un voltaje analógico. Así, si tenemos un PWM que se mantiene con 50% de ciclo de trabajo¹, el voltaje generado será de la mitad del máximo voltaje aplicado; mientras que si se aplica un PWM de 10% de ciclo de trabajo será la décima parte del voltaje.

Uno de los usos frecuentes para PWM son para control de:

- Velocidad de Motores DC, Servos
- Intensidad de pantalla luminosa de LCDs (pantallas de cristal líquido)
- Corriente en carga de baterías

El objetivo de esta nota es:

- Configurar un PWM de tipo “Unbuffered”: Inicializar un PWM de diferentes anchos de pulsos.
- Explorar las limitaciones de un PWM tipo “Unbuffered”: Una limitación de este tipo de PWM es el cambio entre ciclos de trabajos drásticos, usted aprenderá como detectar y combatir este problema.
- Simular: Utilizar “breakpoints” para visualizar la operación de una PWM.

Nota: Se asume un dominio total de la NT0101, NT0102 y se hace referencia cruzada a la misma.

¹ Relación entre el tiempo de encendido y el período, ver Figura 132.

2.4.2 Características de una PWM tipo "Unbuffered"

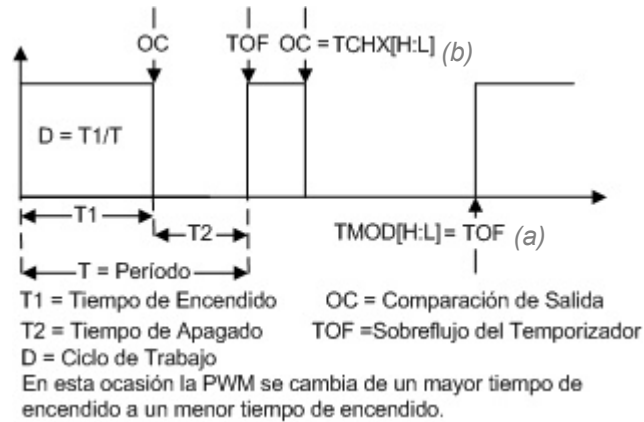


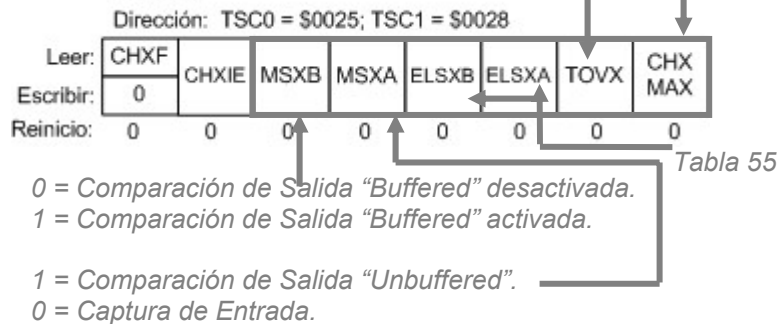
Figura 132 (izquierda). Características de una PWM tipo "Unbuffered". Utilizando el bit de conmutación al sobreflujo del temporizador (TOVX, ver Figura 133) se puede generar una PWM. (a) Período. El período de la onda viene dada por la ecuación 5, pág. 182 (NT0101), controlado por el módulo del temporizador (TMOD[H:L]). (b) Ancho de Pulso. Los registros del canal (TCHX[H:L]) controlan el ancho del pulso y cada uno puede generar una PWM.

Nota: X = 0 ó 1. Canales del Temporizador. Se le dice PWM "unbuffered", pues hay que escribir el nuevo valor del pulso en el canal de temporización.

2.4.3 Registros de Estado y Control de los Canales del TIM en PWM

2.4.3.1 Registros de Estado y Control de los Canales en PWM – TSC0, TSC1

- 0 = 100% de ciclo de trabajo inhabilitado.
 - 1 = Si TOVX es cero, fuerza 100% de ciclo de trabajo.
 - 0 = Conmutación al sobreflujo del temporizador inactiva.
 - 1 = Conmuta al sobreflujo del temporizador, ver Figura 132(a)
- X = 0 ó 1



Nota: Si ELSX[B:A] ≠ 0:0, la combinación MSXA es válida.
MS1B no disponible, solo lee un cero.
CHXMAX tiene efecto en el siguiente período.

Tabla 55. Modulación por Ancho de Pulso "Unbuffered"

PWM [UnBuffered], TOVX = 1	MSXB	MSXA	ELSXB	ELSXA
Conmuta en la comparación de salida	0	1	0	1
Borra en la comparación de salida	0	1	1	0
Impone uno en la comparación de salida	0	1	1	1

X = 0 ó 1. Canal de Temporización.

MSXB = Modo de Selección del Canal X, Bit B. ["Mode Select Bit B"].

MSXA = Modo de Selección del Canal X, Bit A. ["Mode Select Bit A"].

ELSXB = Nivel de Selección del Canal X, Bit B. ["Edge Level Select Bit B"].

ELSXA = Nivel de Selección del Canal X, Bit A. ["Edge Level Select Bit A"].

Nota: No programe la PWM para la primera combinación, puede que genere resultados erróneos al trabajar como PWM, ver Figura 134, problemáticas.

Figura 133 (inferior-izquierda). Registro de Estado y Control del Canal del TIM en PWM. Configure según la tabla 55 para la acción de la PWM. Para mayor detalle de los registros ver NT0101, sección 2.1.3 y NT0102, sección 2.2.2.

2.4.3.2 Problemas y Solución para Cambios de una PWM “Unbuffered” de un Ancho de Pulso Mayor a Menor.

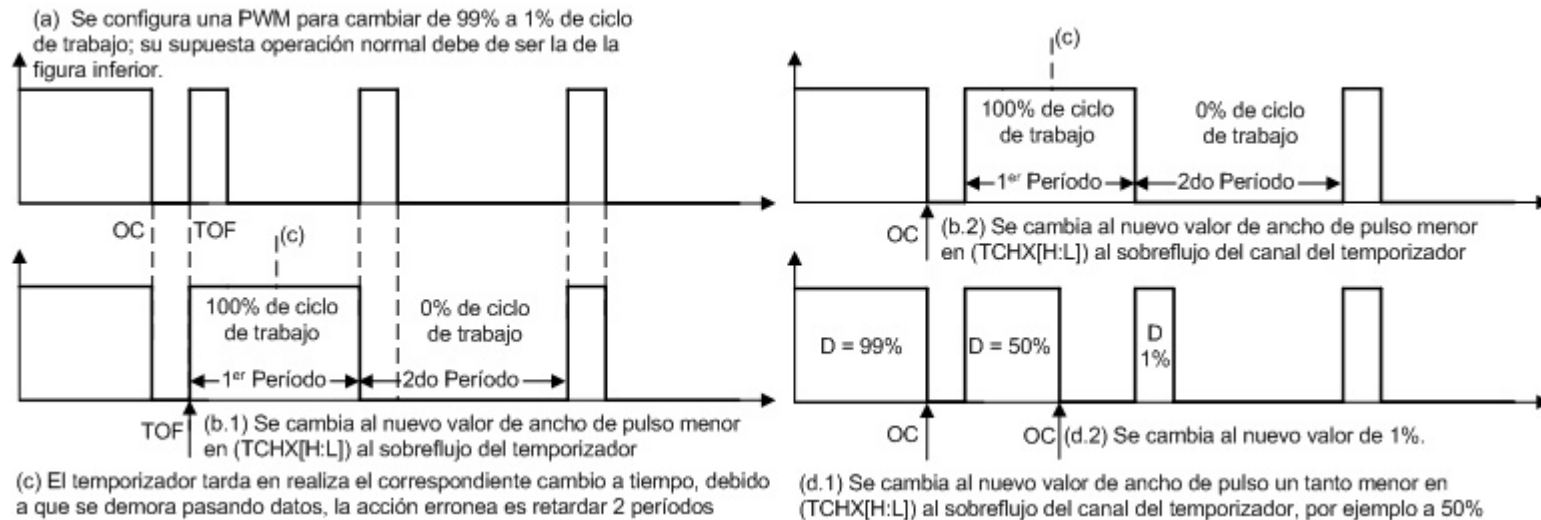


Figura 134. Limitación para la PWM “Unbuffered”. (a) Acción esperada. La PWM cambia drásticamente de un ancho de pulso mayor a menor. (b) Problemas para cambios en (b.1) Sobreflujo del Temporizador. (b.2) Comparación de Salida. Para ambos casos, el temporizador tarda en configurar la PWM. (c) Deficiencia de esta PWM. El temporizador genera dos ciclos de $D = 100\%$ y 0% para luego tomar acción. (d) Solución al problema. (d.1) Cambio de 99% a 50%. Detecte el cambio brusco y cambie a un valor intermedio. (d.2) Cambio de 50% a 1%. Al cambiar a un ancho de pulso intermedio, luego, cambie al valor mínimo deseado. Conclusión: Un cambio de 99% a 1% no debe ser realizado por este modo de PWM.

2.4.3.3 Configuración de Comparación de Salida en Modo "Unbuffered"

Configure el Temporizador para una PWM "Unbuffered" de 400 Hz de frecuencia en el Canal 0, prescalar de dos (PS = 2), borrado en comparación de salida y con un ciclo de trabajo de 10%. Habilite tanto la interrupción del canal como la del temporizador.

$$TMOD[H : L] = \frac{t \cdot f_{xtal}}{2^{2+PS}} = \frac{(1/400) \cdot 4.9152 \times 10^6}{2^{2+2}} = 768_{10} = 300_{16}$$

$$TCH0[H : L] = D \times TMOD[H : L] = 0.1 \times 768 = 76_{10} = 4C_{16}$$

Ecuación 6. Cálculo del Ancho de Pulso

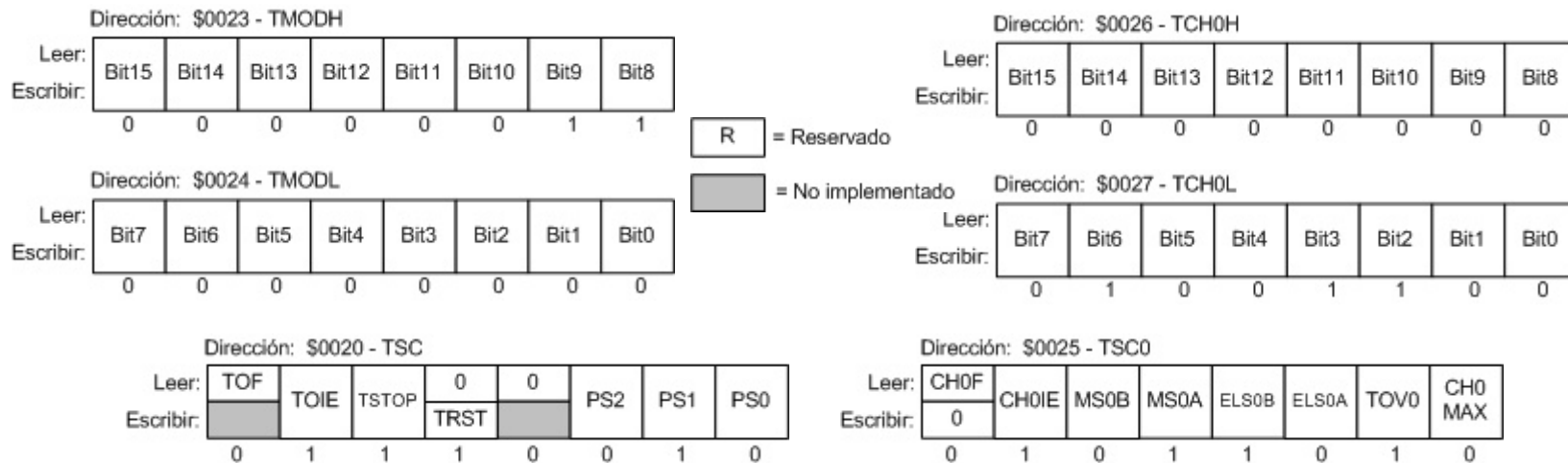


Figura 135. Configuración de una PWM "Unbuffered". El temporizador, canal 0, se utiliza para generar una PWM que existe en el pin PTD4. El registro TMOD[H:L] es el período de la onda, mientras que el ancho del pulso es dado por el registro TCH0[H:L]. Para generar la acción de PWM se debe de habilitar el bit TOVX y habilitar alguna de las combinaciones de la tabla 55 a excepción de la primera combinación la cual puede generar resultados erróneos en uso de una PWM.

2.4.4 Diagrama de Flujo

El siguiente programa hace conmutar un LED conectado al puerto PTD4 a un período de 400 Hz realizando PWM del 10% al 90% y viceversa.

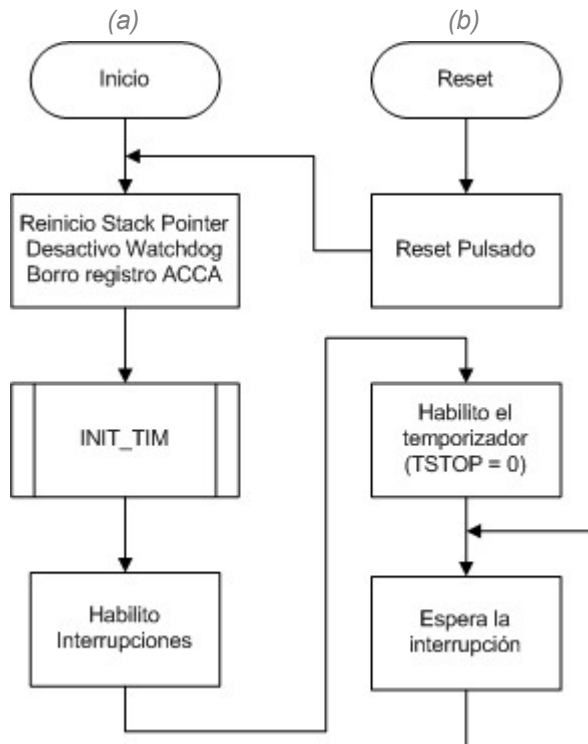


Figura 136. NT0104 – “Timer” – PWM. (a) Programa Principal. Inicializa el temporizador en modo de generación de una PWM tipo “unbuffered”. (b) Reinicio del sistema. Al presionar “reset”, el sistema reinicia automáticamente.

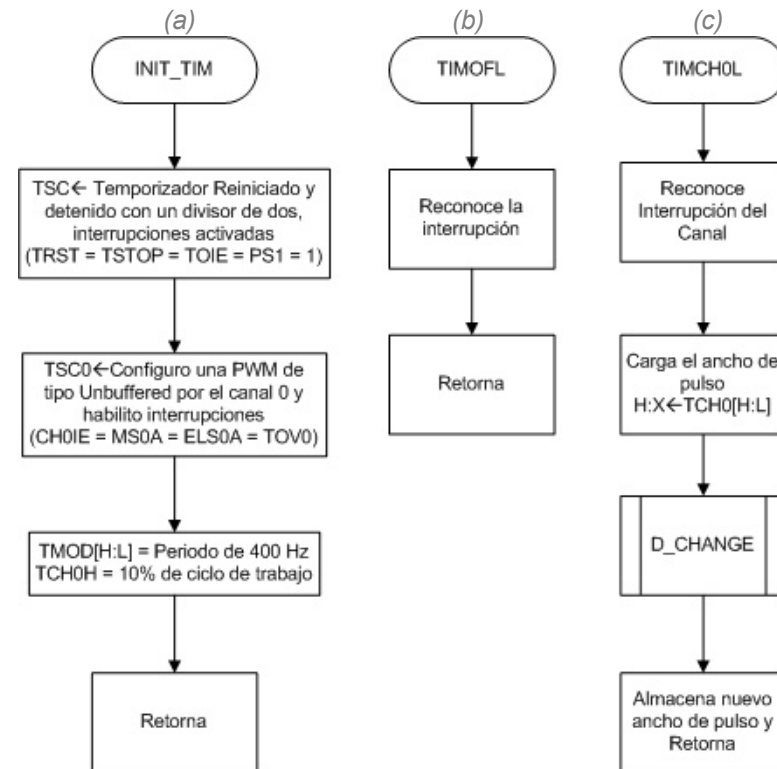
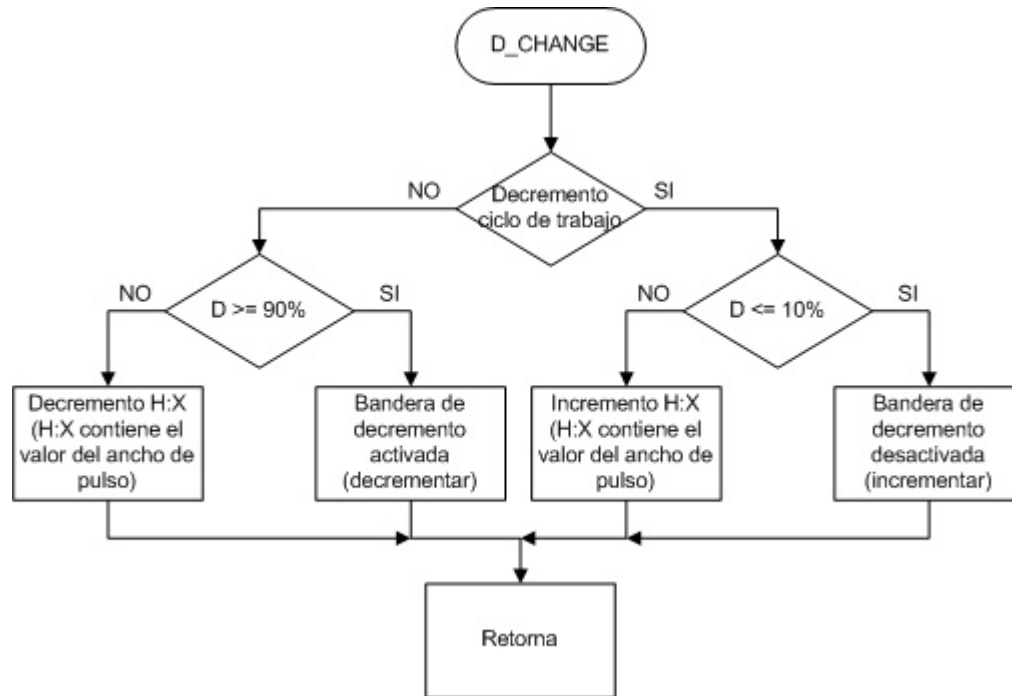


Figura 137. NT0104 – “Timer” – “Output Compare” – Subrutinas. (a) Inicialización de una PWM “Unbuffered”. Inicializa el temporizador en modo PWM “unbuffered” a 400 Hz de frecuencia y 10% de ciclo de trabajo inicial. (b) Interrupción del Temporizador. Se utiliza la misma interrupción para ver el sobreflujo del temporizador y el cambio en el estado del pin. (c) Interrupción del Canal. Cambia de ciclo de trabajo cíclicamente de 10% a 90%.



Nota: Realizar la siguiente conexión adicional.

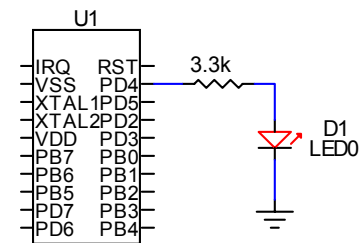


Figura 138. NT0104 – “Timer” – “Output Compare” – Subrutinas – Continuación. D_CHANGE es una subrutina que tiene como entrada en H:X el registro a cambiar el ancho de pulso y de salida en H:X el ancho de pulso incrementado o decrementado si no se ha llegado a sus valores límites de ciclo de trabajo.

2.4.5 Código

```

=====
; ARCHIVO      : NT0104 - Timer - PWM - 04 07 04.asm
; PROPÓSITO   : Generar una PWM de 10% a 90% cíclica de 10kHz de frecuencia.
;              - Observar la configuración de la PWM.
;              - Explorar las limitaciones de una PWM tipo "Unbuffered"
;              P.D.: Agregar una resistencia limitadora al PTD4 de 3.3k
;              con un LED con su cátodo a tierra.
;
; NOTA        :
;              1 - Observar el resultado de generar un PWM cíclico
;              2 - Observar el estado lógico del pin en OC y en TOF.
;
; REFERENCIA:
;              Advanced Information of MCU68HC908JK1, JK3, JL3...
;              http://www.freescale.com/files/microcontrollers/
;              doc/data_sheet/MC68HC08JL3.pdf
;              Pág. 111 - Descripción Funcional de una PWM, sección 10.5.4
;              Pág. 112 - Descripción de una PWM Unbuffered,
;              sección 10.5.4.1
;              Pág. 114 - Inicialización de una PWM, sección 10.5.4.3
;
; LENGUAJE    : IN-LINE ASSEMBLER
;
;-----
; HISTORY
; DD MM AA
; 03 03 03 Created.
; 06 09 04 Last Modification.
;
;-----
;Pasos para iniciar una PWM Unbuffered:
;
; 1 - Definir del registro TSC o registro de
; control del TIM:
;     ¿Necesito o no interrupciones? (TOIE)
;     ¿Cuál que valor de divisor ajustar? (PS[2:0])
; 2 - Definir el registro TMOD[H:L] como módulo máximo de conteo
;     TMOD[H:L] = t*fxtal/(2^(2+PS))
; 3 - Definir el ancho de pulso TCHX[H:L] = D*TMOD[H:L]
; 4 - Configurar el registro TSCX registro de
; control del canal del TIM:
;     4 - a ¿Necesito o no interrupciones? (CHXIE)
;     4 - b TOVX = 1 (Fuerza el pin a cambiar en el sobreflu-
; jo del TIM)
;     4 - c ¿Inicio en bajo o en alto? (MSXB,MSXA,ELAXB,ELAXA)
; Adicionalmente
; 5 - Configurar los vector interrupción a utilizar
;     $FFF2 - TIMOF   $FFF4 - TCH1   $FFF6 - TCH0
; 6 - Escribir el código de la interrupción y su retorno (rti)
;
; 7 - Habilitar Interrupciones globales (cli)
; 8 - Habilitar el temporizador (TSTOP = 0)
=====

```

NT0104

Rev. 1 del 06.08.05

```

$SET      ICS08      ; ICS08 = 1, Vamos a simular en la pastilla
                ; la velocidad de simulación es menor en la
                ; PC.
;SETNOT   ICS08      ; ICS08 = 0, Vamos a programar la pastilla
                ; la aplicación debe correr en tiempo real
$IF ICS08
HZ400     equ $001F   ; 400Hz de período, TMOD[H:L]
D10%     equ $0003   ; 10% de ciclo de trabajo, TCH0[H:L]
D90%     equ $001B   ; 90% de ciclo de trabajo, TCH0[H:L]
$ELSEIF
HZ400     equ $0300   ; 400Hz de período, TMOD[H:L]
D10%     equ $004C   ; 10% de ciclo de trabajo, TCH0[H:L]
D90%     equ $02B3   ; 90% de ciclo de trabajo, TCH0[H:L]
$ENDIF
                ; Fin de la compilación condicional

;=====
;
;                      Definiciones del Usuario
;=====
;
COPD      equ 0T      ; Bit 0 del registro CONFIG1
DOWN      equ 0T      ; Bit 0, registro virtual en RAM
;HZ400    equ $0300   ; 400Hz de período, TMOD[H:L]
;D10%     equ $004C   ; 10% de ciclo de trabajo, TCH0[H:L]
;D90%     equ $02B3   ; 90% de ciclo de trabajo, TCH0[H:L]
BIT5      equ 5T      ; TSC, Bit de Inicio de Conteo, Bit 5 ON
CH0F      equ 7T      ; TSC0, Bit de Salida Comparada
TOF       equ 7T      ; TSC, Bit de sobreflujo del temporizador, Bit
                ; 7, ON
TRST      equ %00010000 ; TSC, Bit de Reinicio de conteo, Bit 4 ON
TSTOP     equ %00100000 ; TSC, Bit de pausa, Bit 5 ON
TOIE      equ %01000000 ; TSC, Bit de Interrupción habilitada, Bit 6, ON
PS1       equ %00000010 ; Preescalar del Temporizador, Bit 1
CH0IE     equ %01000000 ; Habilito interrupciones, canal 0
MS0A      equ %00010000 ; Output Compare Tipo Unbuffered
ELS0A     equ %00001000 ; Bit de Selección de flanco, activa...
                ; la señal en uno lógico al comparar
TOV0      equ %00000010 ; TSC0, Bit de Cambio de estado al
                ; sobreflujo del temporizador

;=====
;
;                      Mapa de Memoria del Microcontrolador
;=====
;
;                      Registro de Configuraciones
;=====
CONFIG1    equ $001F   ; Vectores de configuración

```



```

;=====
;                               Registro de Temporizador
;=====
TSC          equ $0020      ; Dirección, registro de estado y control del
                          ; TIM
TCNTH        equ $0021      ; TCNT, Registro almacenador de cuenta del
                          ; módulo, registro alto.
TCNTL        equ $0022      ; TCNT, Registro almacenador de cuenta del
                          ; módulo, registro bajo.
TMODH        equ $0023      ; TMODH, Registro de cuenta del módulo,
                          ; registro alto.
TMODL        equ $0024      ; TMODL, Registro de cuenta del módulo,
                          ; registro bajo.
TSC0         equ $0025      ; Dirección, registro de estado y control del
                          ; TIM Canal 0
TCH0H        equ $0026      ; TCH0H, Registro almacenador de ancho de
                          ; pulso, registro alto.
TCH0L        equ $0027      ; TCH0L, Registro almacenador de ancho de
                          ; pulso, registro bajo.

;=====
;                               Memoria RAM
;=====
UP_DOWN      equ $0080      ; Variable de incremento o decremento
                          ; de ciclo de trabajo

;=====
;                               Memoria FLASH
;=====
FLASH_START  equ $EC00      ; Puntero - Mem.FLASH

;=====
;                               Vectores de Usuario
;=====
TIMOFH       equ $FFF2      ; Vector de Sobreflujo del TIM (Alto)
TIMCH0H      equ $FFF6      ; Vector de Sobreflujo del TIM, Canal 0(Alto)
RESET_VEC    equ $FFFE      ; Puntero del RESET

;=====
; OBJETIVO : Inicio de Codif. del Ensam-
;           ; blador en Memoria FLASH.
;=====
org FLASH_START          ; Inicio Mem. FLASH

```

```

=====
; OBJETIVO : Configura el TIM para generar
;           una PWM cíclica de 10% a 90%
;           de ciclo de trabajo.
=====
START
    rsp                ; Inic.Stack = $00ff
    bset COPD,CONFIG1 ; Desactiva watchdog
    clra               ; Borra A
    clr UP_DOWN       ; Variable de control de
                    ; incremento o decremento
                    ; del ciclo de trabajo
    jsr INIT_TIM      ; Inicializa TIM
    cli               ; Habilita Interrupciones
    bclr BIT5,TSC     ; Inicia el temporizado (TSTOP = 0)
ESPERA
    wait              ; Espera la interrupción.
    bra ESPERA        ; Salta al modo de bajo consumo
    
```

```

=====
; INIT_TIM : Inicializa el TIM
; OBJETIVO : Inicialización del temporizador.
;           PWM de tipo Unbuffered
; ENTRADA : Ninguna
; SALIDA : Ninguna
; REGISTROS
; AFECTADOS : TSC, TCH0H, TCH0L, TSC0, H:X
=====
INIT_TIM
    mov #{TOIE|TRST|TSTOP|PS1},TSC
                    ; Reinicio el temporizador, habilito interrupciones
                    ; PS[2:0] = 2, el temporizador se encuentra detenido
                    ; por defecto.
    mov #{CH0IE|MS0A|ELSOA|TOV0},TSC0
                    ; Habilito Interrupciones del canal y configuro
                    ; un Output Compare Unbuffered, conmuto al sobre
                    ; flujo del temporizador el puerto.
    ldhx #HZ400
                    ; H:X = 400Hz de frecuencia
    sthx TMODH
                    ; Almacena el período de la PWM
    ldhx #D10%
                    ; H:X = 10% de ciclo de trabajo, inicial
    sthx TCH0H
                    ; Ancho de Pulso, Canal 0, 10% de ciclo de trabajo
    rts
                    ; retorna
    
```

```

=====
; D_CHANGE : Cambia el ciclo de trabajo.
; OBJETIVO : Incrementar o decrementar el
;           ciclo de trabajo
; ENTRADA  : Ninguna
; SALIDA   : Ninguna
; REGISTROS
; AFECTADOS : TSC0, TCH0H, TCH0L, H:X
=====
D_CHANGE
    brset DOWN,UP_DOWN,DOWN.D
                                ; Si UP_DOWN = 1, Decremento D
    cphx #D90%                   ; ¿90% de ciclo de trabajo?
    bge SET.DOWN                 ; Mayor o igual, SI, decrementar
    aix #1                       ; Añado 1 a H:X
    bra OUT.D                    ; salir
SET.DOWN
    dec UP_DOWN                  ; Decrementamos
    bra OUT.D                    ; salir
DOWN.D
    cphx #D10%                   ; ¿10% de ciclo de trabajo?
    ble SET.UP                   ; Menor o igual, SI, incrementar
    aix #-1                      ; Sustraigo 1 a H:X
    bra OUT.D                    ; salir
SET.UP
    inc UP_DOWN                  ; Incrementamos
OUT.D rts                       ; retorna

```

```

=====
; TIMOFL   : Interrupción del Temporizador
; OBJETIVO : Reconocer la interrupción
; ENTRADA  : Ninguna
; SALIDA   : Ninguna
; REGISTROS
; AFECTADOS : TSC0, TCH0H, TCH0L, H:X;
=====
TIMOFL
    bclr TOF,TSC                 ; Reconoce la interrupción
    rti                         ; retorna

```

```

;=====
; TIMCH0L      : Interrupción del Canal 0 del
;               TIM.
; OBJETIVO     : Cambiar de ciclo de trabajo
; ENTRADA      : Ninguna
; SALIDA       : Ninguna
; REGISTROS    :
; AFECTADOS    : TSC0, TCH0H, TCH0L, H:X
;=====
TIMCH0L
    bclr CH0F,TSC0      ; Reconoce salida comparada.
    ldhx TCH0H         ; Carga H:X con TCH0[H:L]
    jsr D_CHANGE      ; Incrementa de 10% hasta 90%
                       ; y visceversa ciclicamente
    sthx TCH0H        ; Deposita en TCH1[H:L], cambio "D"
    rti               ; retorna

```

```

;=====
; OBJETIVO     : Inicializa el Vector de Reset
;               Arranque del programa en la
;               memoria Flash y búsqueda de
;               la interrupción del canal del
;               temporizado y de sobreflujo.
;=====
;===== Vectores del TIM, sobreflujo y comparación =====
    org TIMOFH        ; Vector de Sobreflujo del temporizador
    dw TIMOFL         ; Desborde del temporizador
    org TIMCH0H       ; Vector del Canal 0
    dw TIMCH0L        ; Desborde del contador del Canal
;===== Vector de Reinicio de Sistema =====;
    org RESET_VEC     ; Puntero Vec - RESET

```

Listado 13. NT0104 – "Timer" – PWM. El programa responde a inicializar el temporizador, canal 0 para generar una PWM "Unbuffered" de 400 Hz de período, variar el ciclo de trabajo de 10% a 90% y disminuir la intensidad del LED conectado al PTD4.

2.4.6 Simulación

Para el uso de esta sección se asume tener dominio de la NT0006 (Sección 1.6.4), NT0009 (Sección 1.9.5) y NT0102 (Sección 2.2.3).

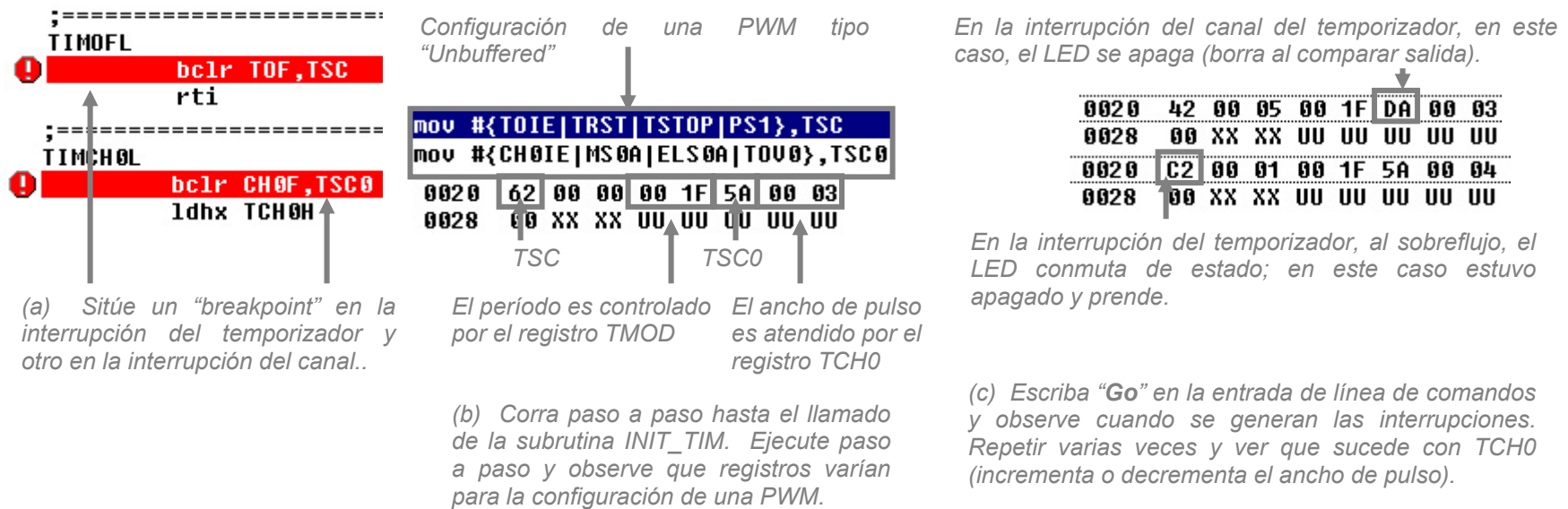


Figura 139. Simulación del Programa Principal. (a) "Breakpoint". Sitúe un "breakpoint" para ver el cambio entre el ancho de pulso y el periodo del temporizador. (b) Registros del Canal del Temporizador. En la rutina de inicialización del canal observe la variación de los registros del temporizador que configuran una PWM tipo "Unbuffered". (c) Desbordes del Canal y el Temporizador. Al desborde del canal, si la interrupción está habilitada, note que el LED de la tarjeta se apagó (borra al comparar salida), ver Tabla 55. Mientras que en la interrupción del temporizador se conmuta el estado del LED. Siga generando nuevas interrupciones del canal y observe que le sucede al ancho de pulso.

- (a) Inicie WinIDE.
- (b) Cargue el archivo NT0104 – “Timer” – PWM - 04 07 04.asm.
- (c) Compile.
- (d) Entre al simulador.
- (e) Añada un (1) “breakpoint” según la figura 139(a).
- (f) Corra su programa paso a paso y en la sección de llamado a la rutina de inicialización del temporizador (INIT_TIM) observe la ejecución de cada instrucción.
- (g) Corra su programa y espere a la llegada del “breakpoint”. Observe el estado del LED, pasa de estar encendido a apagado, ver Tabla 55 y Figura 139(b).
- (h) Nuevamente corra su programa y observe que cuando la cuenta iguala al módulo, sucede una nueva interrupción, compare el estado del LED que ha pasado de estar apagado a encendido, ver Figura 133.
- (j) Si desea “quemar” su pastilla, revisar la NT0009, Sección 1.95.

Nota: Recuerde mover su compilación condicional si desea que su programa corra en tiempo real:

```
; $SET ICS08  
$SETNOT ICS08
```

2.4.7 Conclusión

La PWM opera no muy distinto de una Comparación de Salida (“Output Compare”). El período de la señal está gobernado por el registro módulo (TMOD[H:L]) mientras que el ancho de pulso por el registro del canal (TCHX[H:L]). Sucesivas escrituras sobre el registro que comanda el ancho de pulso en cada período tienen la acción de cambiar el ciclo de trabajo de la onda que es lo mismo al tiempo de encendido y apagado de esta.

Además, la PWM tipo “Unbuffered” tiene la limitante explorada en la sección 2.4.3.3., no se puede cambiar de un ancho de pulso mayor a uno menor, este problema se soluciona cambiando primero a un ancho de pulso intermedio primero y luego al pulso más pequeño.

Para fin demostrativo se hizo una PWM de 400 Hz que varía su ciclo de trabajo de 10% a 90% y viceversa para demostrar que al variar el ancho de pulso a una rata muy alta varía la intensidad de un LED conectado al canal, esto corresponde a variar el voltaje aplicado al LED.

2.4.8 Referencias

2.4.8.1 Información Avanzada sobre el Microcontrolador

(a) http://www.freescale.com/files/microcontrollers/doc/data_sheet/MC68HC08JL3.pdf

Pág. 111 - Descripción Funcional de una PWM, sección 10.5.4

Pág. 112 - Descripción de una PWM "Unbuffered", sección 10.5.4.1

Pág. 114 - Inicialización de una PWM, sección 10.5.4.3

2.4.8.2 Manual de Referencia del CPU

(a) http://www.freescale.com/files/microcontrollers/doc/ref_manual/CPU08RM.pdf

2.4.8.3 Página "web" sobre esta Nota Técnica

(a) <http://www.geocities.com/issaiass/>

2.4.9 Problemas Propuestos

2.4.9.1 Genere una PWM de 10kHz de frecuencia, Canal 0, PS = 0 de 50% de ciclo de trabajo, imponer al comparar salida.

2.4.9.1 Configurar una PWM de 60 Hz de frecuencia, Canal 1, PS = 5 de 80% de ciclo de trabajo, borrar al comparar salida.

2.4.9.1 Configurar una PWM de 60 Hz de frecuencia, Canal 1, PS = 5, y varíe el ciclo de trabajo con un potenciómetro.

Sugerencia: $TCHX[H:L] = (TMOD[H:L]/255) \times ADR$