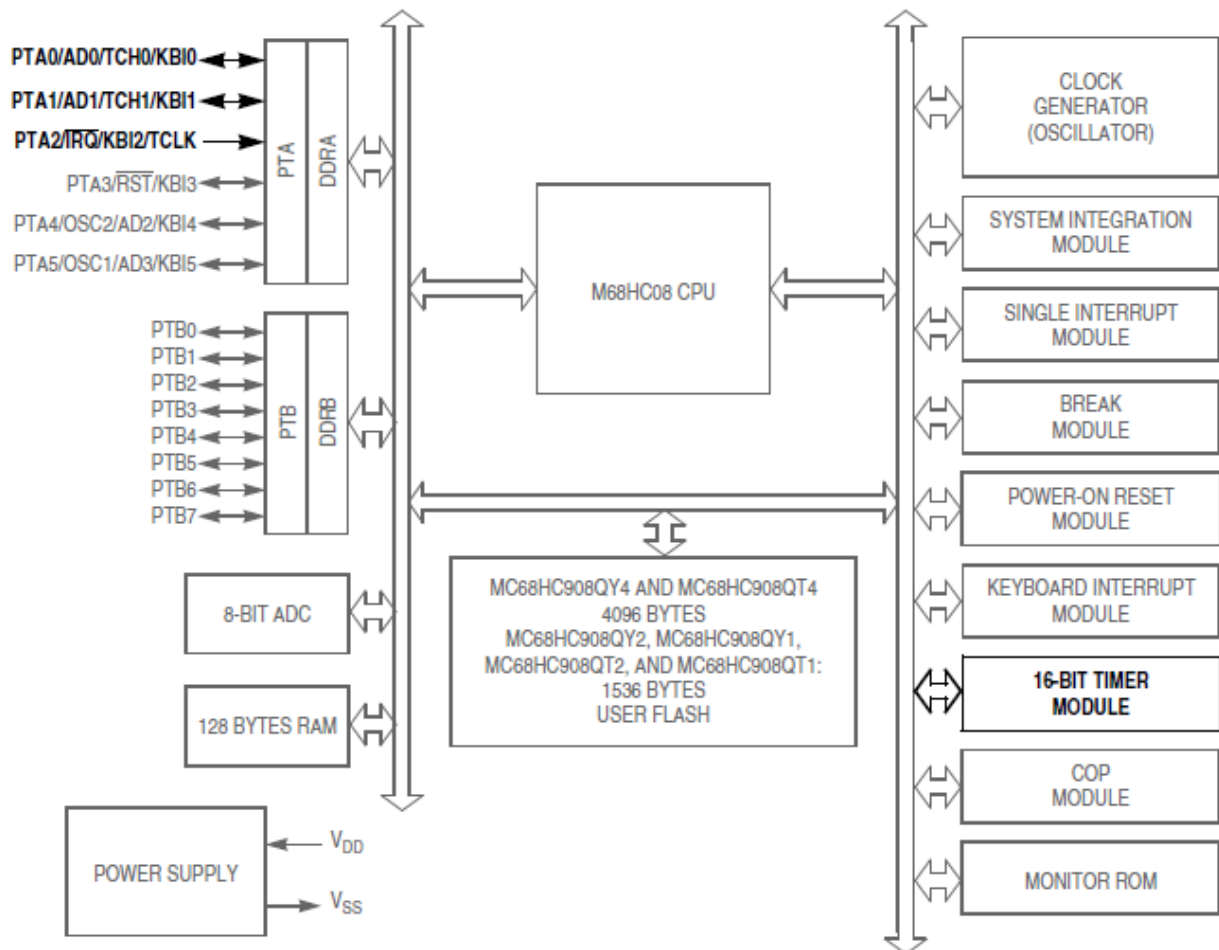


## MODULO TIM

El módulo TIM (Modulo de Interfaz de Tiempo) contiene un temporizador de dos canales que proporciona un tiempo de referencia con posibilidad de 3 funciones:

- a. Input Capture
- b. Output Compare
- c. PWM

El modulo TIM comparte sus pines con los pines de propósito general del puerto A:



A continuación describiremos la función PWM del módulo TIM.

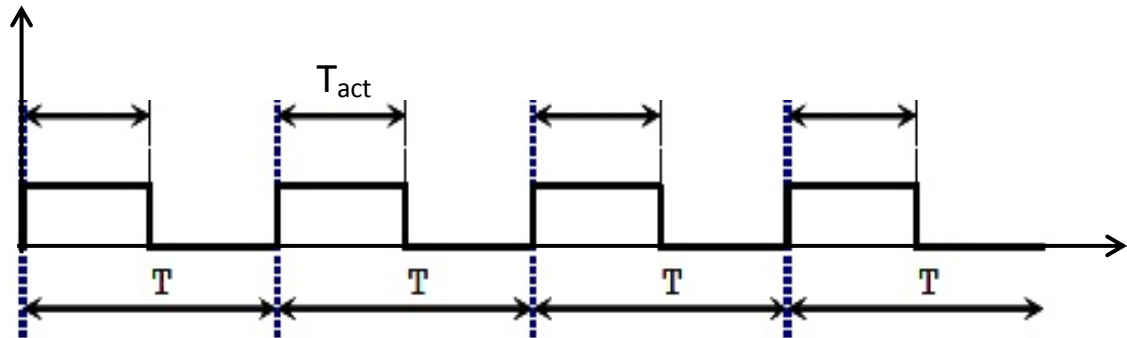
### PWM (Modulación por Ancho de Pulso)

La modulación de ancho de pulso se usa para generar una forma de onda con un periodo fijo y el ciclo de servicio variable, el ciclo de servicio es la relación existente entre el tiempo en el cual la señal se mantiene en un estado alto "1" y el tiempo en el que la señal se mantiene en un estado bajo "0". De acuerdo a esto, podemos decir que en la modulación PWM se pueden presentar diferentes frecuencias y resoluciones. La frecuencia está definida por el periodo o



## Técnicas Digitales II – PWM

duración de un ciclo completo de la señal ( $T=T_H+T_L$ ), y la resolución es definida por el número de pasos discretos del ciclo de servicio que se pueden poner a "1".



$$D = \frac{T_{act}}{T}$$

El ciclo de trabajo de una señal periódica es el ancho relativo de su parte positiva en relación con el período. Expresado matemáticamente:

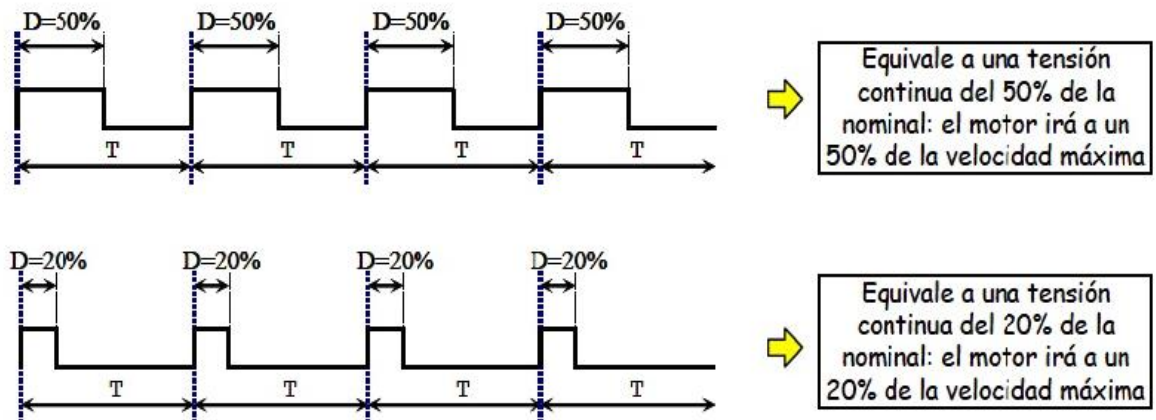
$D$  : es el ciclo de trabajo.

$T_{act}$  : es el tiempo en que la función es positiva (ancho del pulso).

$T$  : es el período de la función

### Ejemplo de aplicación del PWM

Para variar la velocidad de trabajo en un motor:



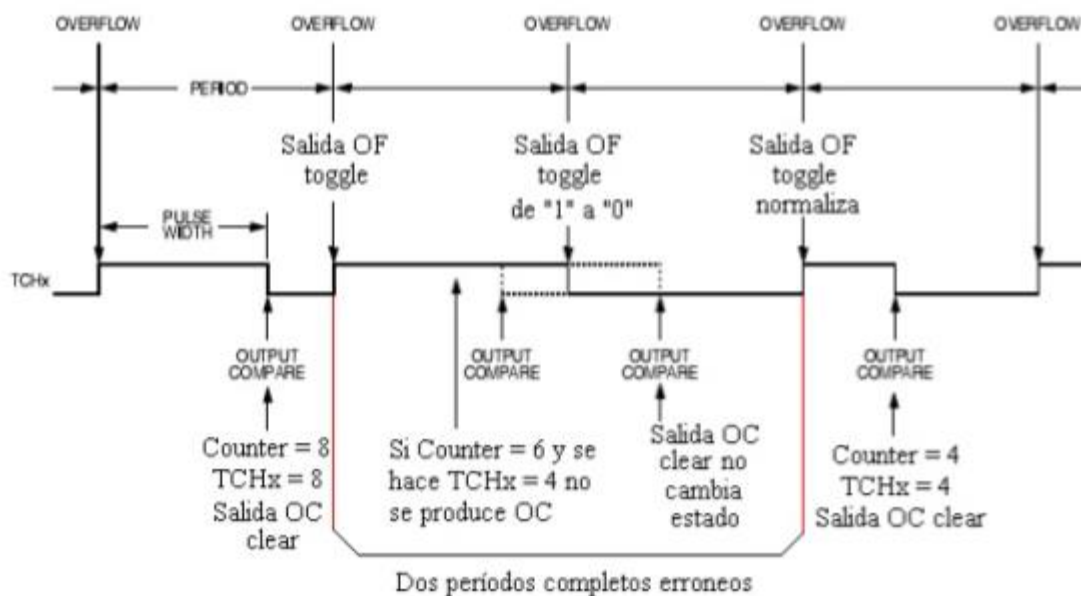
### FORMAS DE PROGRAMAR EL PWM

Existen dos variantes de PWM, las cuales son buffered y unbuffered:

#### PWM Unbuffered

En esta modalidad se puede configurar un PWM por cada canal del timer, y se denomina así ya que para cada cambio de ancho de pulso solo se requiere escribir el nuevo valor en el registro de canal correspondiente.

La desventaja de este método es que, si se cambia el registro de canal a un valor menor y el contador ya lo pasó sin haber alcanzado aún a producir el output compare de ese período, se produce una salida errónea por dos períodos como se muestra en la figura:



La manera de evitar esto es la siguiente:

- Si se desea cambiar el registro de canal a un valor menor, habilitar la interrupción de output compare de canal y en el código de atención de interrupción, configurar el nuevo valor del registro del canal. No configurar aquí el registro de canal si se desea poner un valor mayor porque se puede producir dos output compare en el mismo período.
- Si se desea cambiar a un valor mayor, habilitar la interrupción por overflow y cambiar ahí el valor del registro del canal.

De esta manera, a pesar de la complejidad sumada al configurar dos interrupciones, se puede tener un PWM de funcionamiento estable.

#### PWM Buffered

En esta modalidad, los canales 0 y 1 se encadenan, trabajando en conjunto y presentando la salida por el pin correspondiente al canal 0 (el pin del canal 1 estará disponible como puerto

de entrada-salida). Aquí, los nuevos valores del registro de canal se deben escribir en aquel que no esté activo (el MCU comienza utilizando el registro del canal 0) y estará activo a partir del siguiente overflow, cambio que realiza el MCU; será responsabilidad del programador llevar el control de cuál es el registro activo y cual es aquel que deba modificarse, ya que si se modifica el registro activo, se estaría trabajando en la modalidad unbuffered.

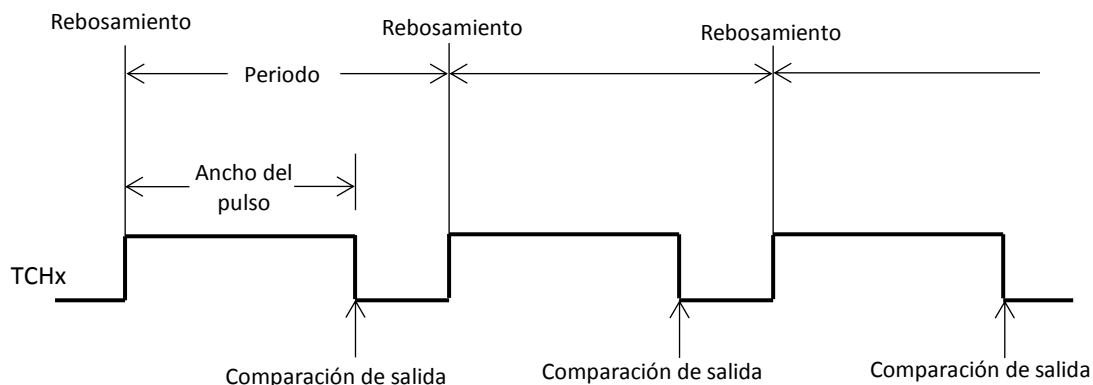
De esta manera, logramos estabilidad de funcionamiento evitando la complejidad de codificar interrupciones, aunque se pierda la funcionalidad de uno de los canales del timer.

### **IMPLEMENTACIÓN DEL PWM EN EL MICROCONTROLADOR**

Para efectuar la operación de modulación por ancho de pulso se debe tener en cuenta que el periodo de la operación PWM se adjudica en los registros Modulo del contador del TIM y en los registros TCHxH:TCHxL se configura el ancho de pulso. La comparación de salida en el registro del canal del TIM determina el ancho de pulso de la señal PWM. El tiempo entre rebosamiento y la comparación de salida corresponde al ancho de pulso.

#### **Período y ancho de pulso de la operación PWM**

El valor de los registros modulo y prescalador de salida determina la frecuencia de salida de la señal PWM. La frecuencia es variable en 256 incrementos. Si TMODH:TMODL = \$00FF (255), los registros contadores de modulo producen un periodo de 256 veces el periodo del reloj interno si el valor del prescalador 000 (configuración del prescalador en el registro TSC).



El valor de los registros del canal del TIM determina el ancho de pulso de la señal PWM. Aquí también se tiene una variación de 256 incrementos. Si, por ejemplo, el valor \$0080 (128), es almacenado en los registros canales del TIM, produce una duración de ciclo de  $(128/256)$  es decir 50%.

### **PROGRAMACIÓN DEL PWM**

#### **Inicialización de la función PWM**

Para lograr un correcto funcionamiento de la función PWM:

1. Efectuar las siguientes tareas en el registro TSC:
  - TSTOP = 1
  - TRST = 1
2. En los registros (TMODH:TMODL), escribir el valor requerido para el periodo PWM

## Técnicas Digitales II – PWM

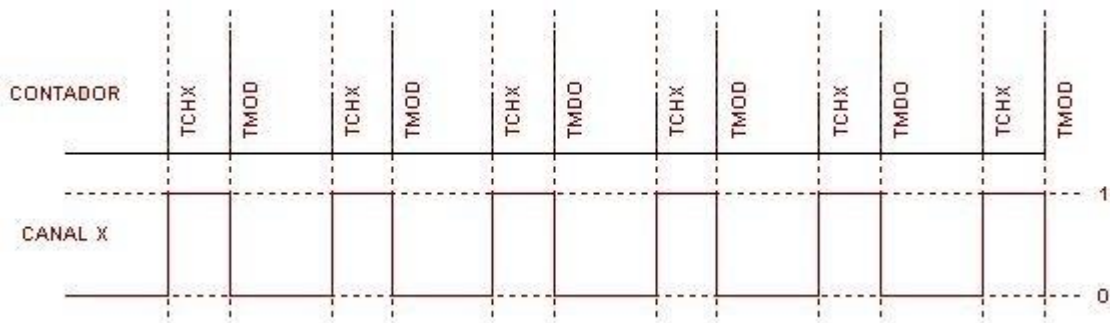
3. En los registros (TCHxH:TCHxL), escribir el valor requerido para el ancho de pulso
4. En el registro TSCx, efectuar las siguientes tareas:
  - Escribir 0:1 (para comparación de salida sin buffer o señales PWM) o escribir 1:0 (para comparación de salida con buffer o señales PWM) en el modo de selección de bit MSxB:MSxA; ver tabla de selección de modos y flancos expuesta anteriormente.
  - $TOVx = 1$
  - Escribir 1:0 (para 0 lógico la comparación de salida) o escribir 1:1 (para 1 lógico la comparación de salida) en el bit de selección para flanco/nivel ELSxB:ELSxA.
5.  $TSTOP = 0$  del registro TSC

El registro TSCR0 es el que se encarga de controlar y monitorear la señal PWM. Cuando se coloca en el bit  $TOVx = 0$  tiene como resultado que la duración del ciclo de salida es 0%.

### Programación

En el modo PWM el canal conmuta en un valor que toma TCHX y conmuta de nuevo en el valor que toma TMOD. Se puede configurar para que el canal en el valor de TCHX conmute a uno o a cero.

En el siguiente ejemplo se configura el canal para que con TCHX conmute a uno y con TMOD conmute a cero.



En este programa cada señal permanecerá un 20% en nivel alto. El canal 0 Conmuta a cero en TCH0 y el canal 1 conmuta a uno en TCH1.

Variando el valor de TCHX y dejando fijo el valor de TMOD se tiene un PWM variable.

### APLICACIÓN PRACTICA DE LA FUNCION PWM

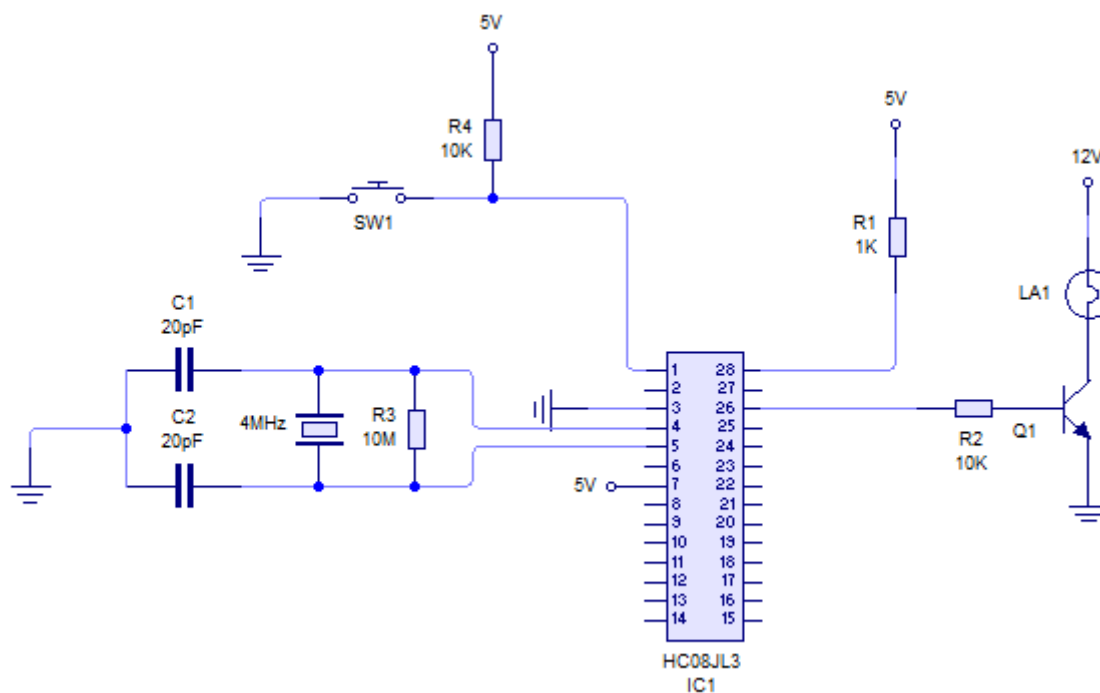
El objetivo de nuestro trabajo es la implementación de la programación del PWM del módulo timer en el control de la intensidad de una lámpara, utilizando un microcontrolador de la familia HC08 Freescale (JL3).

El sistema contará con un pulsador, el cual cada vez que se presione incrementará la intensidad de la lámpara pasando duraciones de ciclo de trabajo PWM de 10%, 20%, 30%, 40% o 50%. Cuando se encuentre en el ciclo de trabajo del 50% y se presione el pulsador, éste provocará reiniciar el ciclo de trabajo a un 10%.

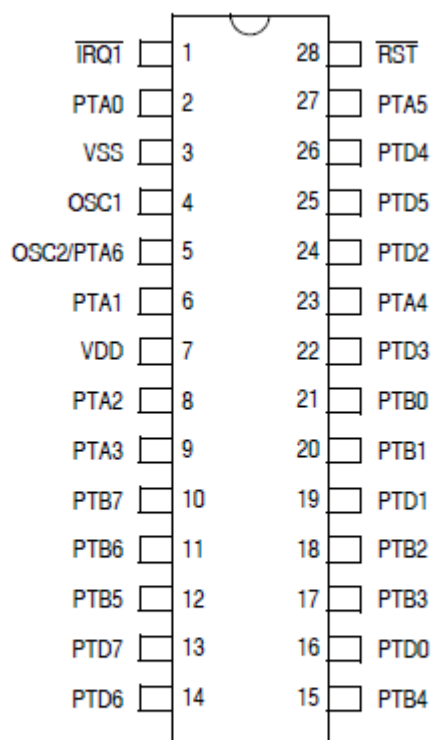


## Técnicas Digitales II – PWM

Esquema circuital para el control de la intensidad lumínica:



Pin out del microcontrolador usado en el proyecto:



**28-PIN ASSIGNMENT  
MC68H(R)C908JL3**

## Técnicas Digitales II – PWM

Como se requiere que con el control del PWM se obtengan distintos porcentajes de ciclo útil, se necesita crear una tabla de la secuencia de los valores correspondientes al registro TCH0L.

- Como se estableció que el registro TMODH:TMODL=\$00FF
- El valor del prescaler del timer se desea configurar en 64
- Frecuencia del oscilador externo 4MHz

Teniendo en cuenta que:

$$1 \text{ CICLO DE MÁQUINA} = 4 \text{ CICLOS DEL OSCILADOR}$$

Entonces, si se considera esto, se puede afirmar que:

$$\text{frecuencia del bus interno} = \frac{\text{Frec. oscilador}}{4}$$

$$1 \text{ ciclo de máquina} = \frac{1}{\text{Frec. Bus Interno}} = \frac{4}{\text{Frec. Oscilador}}$$

El valor del periodo de la señal PWM será:

$$\text{Periodo PWM} = \frac{4}{F_{\text{prescaler}}} = \text{prescaler} \times (\text{TMODH:TMODL})$$

$$\text{Periodo PWM} = \frac{4}{4 \times 10^6} \times 64 \times (255) = 16.32 \text{ ms}$$

Los valores del registro TCH0L serán entonces los siguientes:

$$\text{TCH0L} = \frac{255}{100} \times (\% \text{ servicio})$$

% servicio	Valor calculado	Valor TCH0L
10	25.5	28
20	51	51
30	76.5	77
40	102	102
50	127.5	128

Entonces la tabla quedara de la siguiente manera:

TABLA	DB	\$0, 26T, 51T, 77T, 102T, 128T, \$0
-------	----	-------------------------------------



## Técnicas Digitales II – PWM

Los valores extremos de la tabla (\$0) se establecieron como 0 para que cuando se detecte el ultimo valor de la tabla, al presionar nuevamente el pulsador se inicialice el proceso secuencial de valores. Teniendo en cuenta lo expuesto antes sobre los pasos de inicialización del módulo PWM, el programa final quedaría de la siguiente forma:

```
FLASH EQU $ECE0
RESET EQU $FFFE
T_PWM EQU $FFF6
RAM EQU $0080
COPD EQU 0
IMASK1 EQU 1
MODE1 EQU 0
ACK1 EQU 2
$Include 'JL3REGS.inc'

    ORG RAM
DATO RMB 1
MAXIMO RMB 1
MINIMO RMB 1

    ORG FLASH
TABLA DB $0,26T,52T,77T,102T,128T,$0
INICIO BSET COPD,CONFIG1 ;Inahbilita el COPD

    CLRA ; A=0
    CLRX ;X=0
    CLR DATO ;DATO=0

*Configuracion de los puertos

    MOV #$00,PORTD ;Puerto D borrado
    MOV #$FF,DDRD ;PTD(7:0) OUT

*Configuracion del TIMER

    MOV #$36,TSC
    MOV #$00,TMODH
    MOV #$255T,TMODL
    MOV #$00,TCH0H
```





## Técnicas Digitales II – PWM

```
MOV  #$26T,TCH0L
MOV  #%01101010,TSC0 ;Inicia ciclo de servicio 10%
MOV  #$06,TSC      ;Configuracion PWM
*CONFIGURACION DE LA INTERRUPCION IRQ
BCLR  IMASK1,INTSCR ;Habilita interrupcion de la IRQ
BCLR  MODE1,INTSCR  ;IRQ por flanco descendiente
CLI           ;Habilita interrupciones
SALTO BRA  SALTO      ;Bucle infinito
PWM  PSHH
      BCLR  7,TSC0      ;Interrupcion atendida
      PULH
      RTI
*INTERRUPCION IRQ
IRQ  PSHH              ;Guarda H en el Stack
      BIH  FIN_IRQ      ;Salta a FIN_IRQ si el pin IRQ=1
DIMMER LDA  TABLA,X    ;Carga VALOR(% SERVIVIO)
      BEQ  LIMPIAR      ;Si valor=0, salta
      STA  TCH0L        ;cambia el % de servicio
FIN_IRQ BSET  ACK1,INTSCR ;Borra interrupcion IRQ
      PULH              ;Saca H del Stack
      RTI              ;Salir de Interrupcion
LIMPIAR CLRX           ;X=0
      BRA  FIN_IRQ
ORG  T_PWM
DW           PWM
ORG  $FFFA
DW  IRQ        ;Cuando ocurre interrupcion por IRQ
ORG  RESET
DW  INICIO
```