Notas Técnicas de Uso y Aplicación

NT0010

1.10 PUERTOS GENERALES DE ENTRADA Y SALIDA
USO DE PUERTOS PARA GENERAR SEÑALES Y DETECTAR SEÑALES DIGITALES

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/

ÍNDICE

1.10.1 Introducción

Los puertos de entrada – salida son la iniciación a interfase al mundo real. Gracias a ellos podemos: encender leds, controlar motores, leer el estado de un jumper, generar el encendido de una señal a un tiempo determinado, etc.

Por medio de esta nota ud. podrá:

- Entender que es un Puerto de Entrada – Salida y sus registros: Los puertos de entrada salida tiene la capacidad de ser configurados como entradas o salidas. También dependiendo del registro, este tiene la capacidad de manejar LEDs, alto manejo de corriente ó resistores "pull-ups" internos.

- Aprender a generar un PWM por "Software": Un modulador por ancho de pulso o PWM varía el tiempo de encendido o apagado de una señal (ver figura 98) y así poder generar una señal cíclica para atenuar o afirmar el brillo de un LED.

1.10.2 Puertos de Entrada Salida (E/S)

1.10.2.1 Puertos de Entrada y Puertos de Salida

Los puertos de entrada/salida sirven para comunicarse con el mundo real. Cada puerto es bidireccional, es decir, se puede configurar un conjunto de bits como entradas ó como salidas dependiendo de la necesidad del usuario. Por ejemplo, el microcontrolador JL3 posee veintitrés (23) pines de entrada/salida. Si algún pin del puerto se configura como entrada, este puede captar señales lógicas. Si algún pin del puerto se configura como salida, este puede enviar señales lógicas.

1.10.2.2 Registros

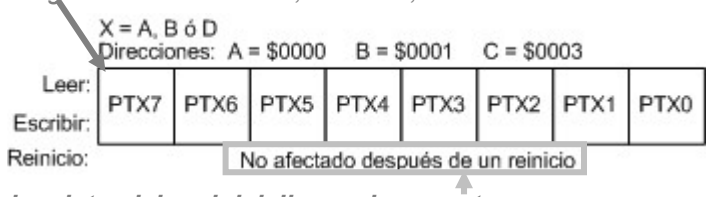
En el caso de los registros de los puertos están configurados en grupos de 8 bits cada uno, de los cuales, el puerto A (PTA) y el registro de direcciones de A (DDRA), el último bit no está disponible.

1.10.2.2.1 Registro de Direccionamiento de Datos del Puerto

Figura 93. Registro de Direccionamiento de Datos del Puerto. Sirven para configurar los pines de los puertos como entradas o salidas. Si algún bit posee un cero (0) el correspondiente pin del puerto es una entrada; si algún bit posee escrito un uno (1) el correspondiente pin del puerto es destinado a señal de salida.

1.10.2.2.2 Registro de Datos del Puerto

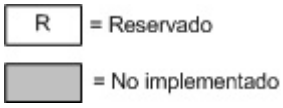
Nota: Bit 7 del registro A no escribible, al leerse, se lee un 0.



Para evitar falsa data, deben inicializarse los puertos
Si se lee ó escribe algún bit del registro con:
1 = el pin del puerto está en estado lógico alto
0 = el pin del puerto está en estado lógico bajo

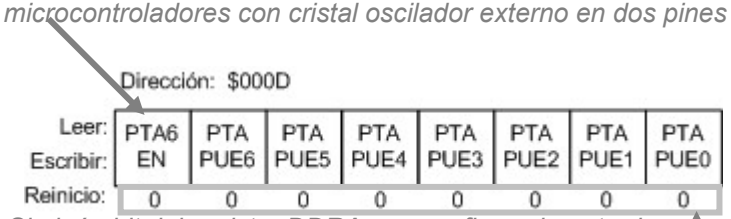
Figura 94. Registro de Datos del Puerto. Sirven para configurar los pines de los puertos para recibir o habilitar señales. Si el correspondiente bit es un cero (0), tiene un estado lógico bajo. Si el correspondiente bit es un uno (1), tiene un estado lógico alto.

IMPORTANTE: Para eliminar rebotes, configure primero los registros de datos de puertos y luego los registros de direccionamiento.



1.10.2.2.3 Registro de Habilitación de Resistencias "Pull-Ups"

Sin efecto en microcontroladores con cristal oscilador externo en dos pines



Si algún bit del registro DDRA; es configurado entrada:
1 = Activa el resistor 30k de pull-up al estado lógico alto
0 = Desactiva el resistor de pull-up

Figura 95. Registro de Habilitación de Resistencias "Pull-Ups". Configuran los pines del puerto A para conectarlos a un estado lógico alto habilitando las resistencias de "pull-ups" de 30k internas. Un uno (1) habilita los resistores de "pull-ups", mientras que un cero (0) desconecta los resistores internos.

1.10.2.2.4 Registro de Control del Puerto D

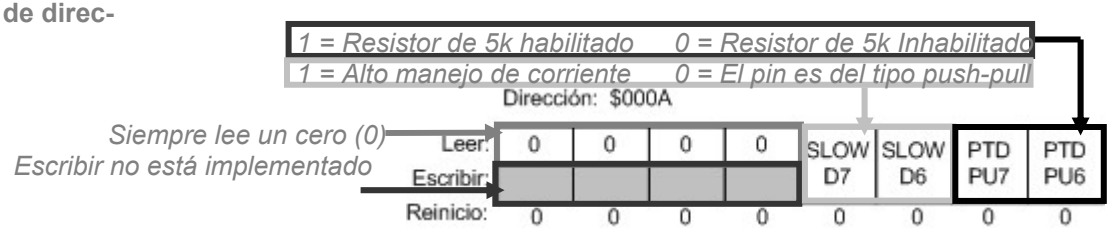


Figura 96. Registro de Control del Puerto D. Habilita ó deshabilita la capacidad de alto manejo de corriente de los puertos SLOWD7, SLOWD6; habilita ó deshabilita los resistores internos de "pull-ups" a 5k del puerto.

1.10.2.3 Ejemplo de Configuración de Registros

(1) Configure PTA[3:6] como salidas en estado alto y PTA[0:2] como entradas.
(2) Habilite el resistor de “pull-up” para el PTA0.
(3) Habilite el PTD6 con capacidad de alto manejo de corriente y al PTD7 con su resistor de “pull-up”.

(1)

Dirección: $0000

Leer:	0	PTA6	PTA5	PTA4	PTA3	PTA2	PTA1	PTA0
Escribir:								
	0	1	1	1	1	0	0	0

Dirección: $0004

Leer:	0	DDRA 6	DDRA 5	DDRA 4	DDRA 3	DDRA 2	DDRA 1	DDRA 0
Escribir:								
	0	1	1	1	1	0	0	0

Se recomienda inicializar primero los puertos y luego definir si son entradas o salidas.

(2)

Dirección: $000D

Leer:	PTA6 EN	PTA PUE6	PTA PUE5	PTA PUE4	PTA PUE3	PTA PUE2	PTA PUE1	PTA PUE0
Escribir:								
	0	0	0	0	0	0	0	1

Nota: El resistor de “pull-up” solo funciona si el puerto es una entrada.

(3)

Dirección: $0003

Leer:	PTD7	PTD6	PTD5	PTD4	PTD3	PTD2	PTD1	PTD0
Escribir:								
Reinicio:	0	0	0	0	0	0	0	0

Se recomienda siempre inicializar los puertos.

Dirección: $0007

Leer:	DDRD 7	DDRD 6	DDRD 5	DDRD 4	DDRD 3	DDRD 2	DDRD 1	DDRD 0
Escribir:								
Reinicio:	0	1	0	0	0	0	0	0

Dirección: $000A

Leer:	0	0	0	0	SLOW D7	SLOW D6	PTD PU7	PTD PU6
Escribir:								
Reinicio:	0	0	0	0	0	1	1	0

Figura 97. Configuración de los Puertos de Entrada/Salida. (1) Configuración de los Puertos como Entradas y Salidas. (2) Habilitación del Resistor de “Pull-up”. (3) Configuración de “Pull-ups” y manejo de corriente en el Puerto D.

1.10.3 Diagrama de Flujo

El siguiente programa simula con la tarjeta un efecto atenuador en el LED PTD7.

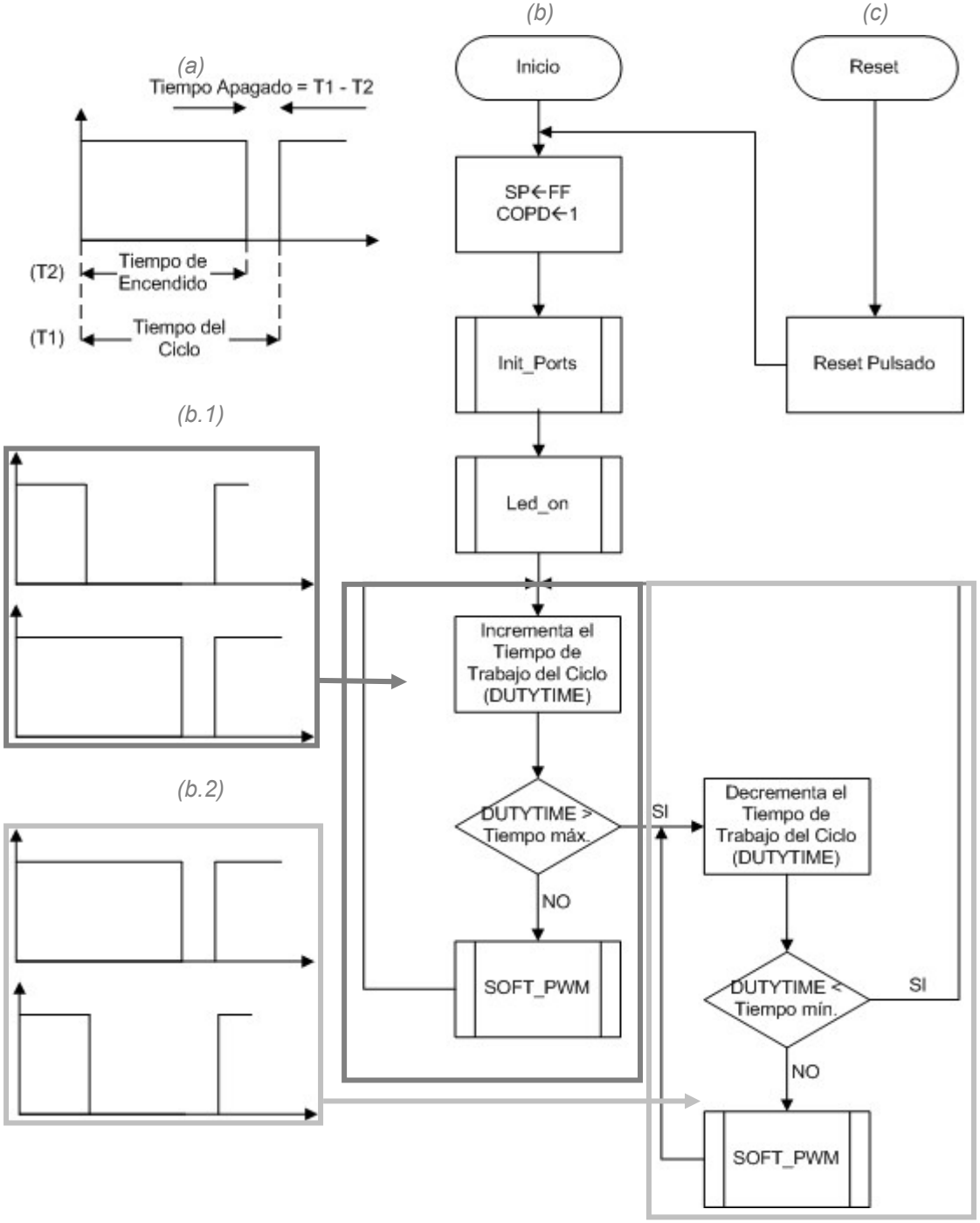


Figura 98. NT0010 – Puertos E-S. (a) Forma de Onda de Salida. Muestra las características de una onda simple, un tren de pulsos. (b) Programa Principal. (b.1) Incremento del ciclo de trabajo. La sección del código diagrama trabaja la onda incrementando el tiempo de encendido. (b.2) Decremento del ciclo de trabajo. La sección del diagrama trabaja la onda decrementando el tiempo de encendido. (c) Reinicio del sistema.

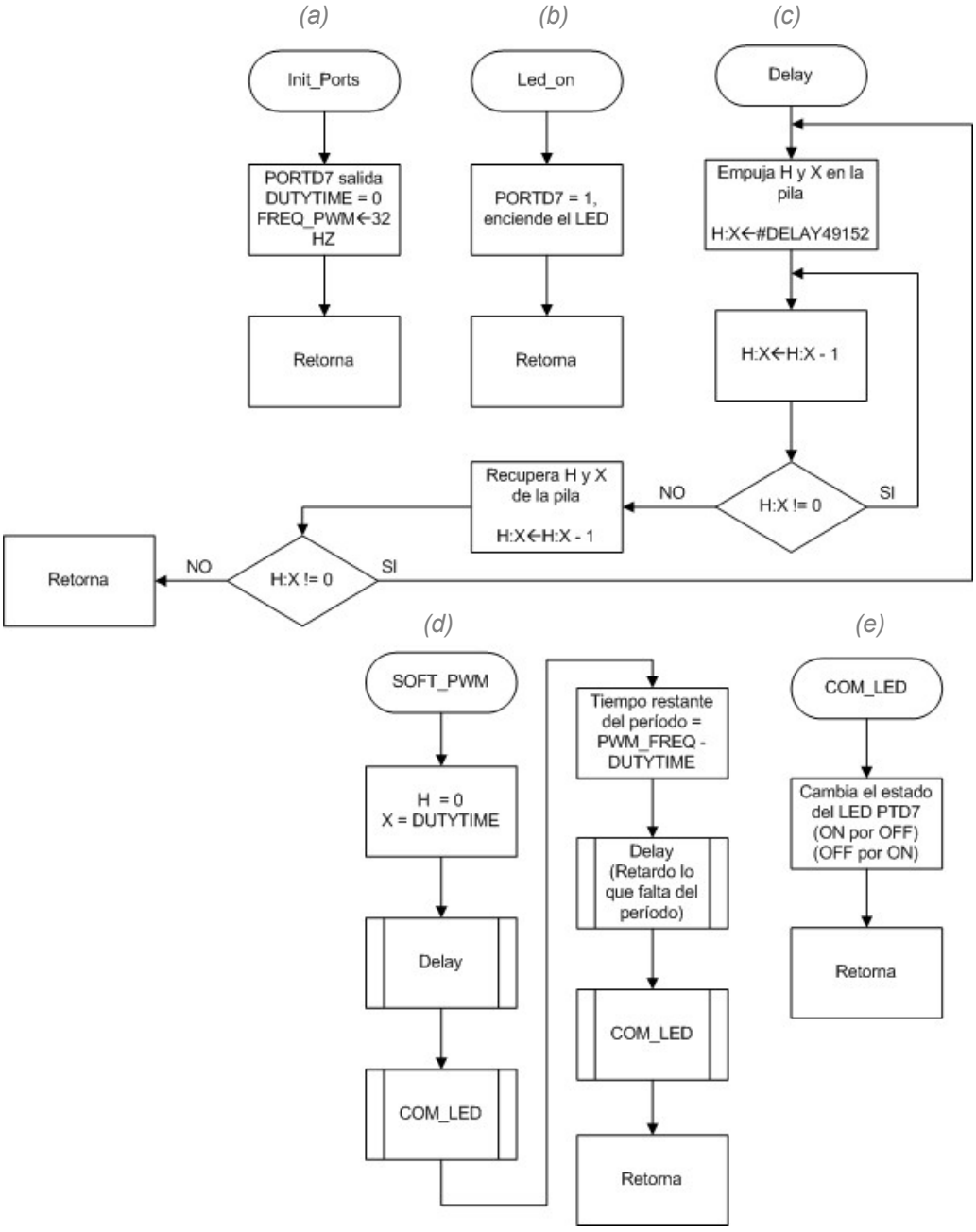


Figura 99. NT0010 – Puertos E-S - Subrutinas. (a) Inicializa Puertos. Inicializa el Puerto como salida y variables globales como el tiempo de encendido y la frecuencia. (b) Encendido del LED. Enciende el LED PTD7. (c) Retardo programable. Ejecuta un retardo entre 1 ms y 65 seg., tipo programable. (d) PWM por "Software". Ejecuta las acciones de las figuras 98(b.1) y 98(b.2), incrementa o decrementa el ciclo de trabajo. (e) Conmutar LED. Conmuta el estado del LED PTD7, si el LED estaba encendido, lo apaga; por otro lado, si estaba apagado, lo enciende.

1.10.4 Código

```
;*****************************************************************
; ARCHIVO      : NT0010 - Puertos E-S - 25 02 04.asm
; PURPOSE      : Ejecuta un conjunto de Pulsos variable a una frecuencia de
;                50 Hz por software.
;                Probar el uso de los pines de E-S del microcontrolador para
;                propósitos generales.
;
;                Nota:  El PWM (Modulación por ancho de pulso) de software
;                implementado por retardo de software no es exacto, tiene una
;                variación del 0.6% MAYOR al estimado.
;
; LENGUAJE     : IN-LINE ASSEMBLER
; ---------------------------------------------------------------------------
; HISTORIAL
;   DD MM YY
;   12  03  04   Creado.
;   29  08  04   Modificado.
;*****************************************************************
$SET            ICS08               ; ICS08 = 1, Vamos a simular en la pastilla
                                    ; la velocidad de simulación es menor en la
                                    ; PC.
;$SETNOT        ICS08               ; ICS08 = 0, Vamos a programar la pastilla
                                    ; la aplicación debe correr en tiempo real
$IF ICS08                           ; Si ICS08 = 1
DELAY49152      equ $0010           ; Constante de Retardo a 4.9152 MHz –
                                    ; Simulación
$ELSEIF                             ; De lo contrario (ICS08 = 0)
DELAY49152      equ $0096           ; Constante de Retardo a 4.9152 MHz –
                                    ; Tiempo Real
$ENDIF                              ; Fin de la compilación condicional

;*****************************************
;              Definiciones del Usuario
;*****************************************
COPD            equ $0000           ; Bit 0, Registro CONFIG1
BIT7            equ 7T              ; Bit 7 del Registro
ONE             equ 1T              ; Añade o sustrae el número 1
BIT7MASK        equ $80             ; Máscara para el Bit 7 del Puerto
DUTYTIMEMAX equ $0019               ; Tiempo máximo  = 31.
DUTYTIMEMIN  equ $0001              ; Tiempo mínimo = 1
HZ32            equ $002            ; 32 Hz de frecuencia

;*****************************************
;          Mapa de Memoria del Microcontrolador
;*****************************************
;*****************************************
;              Registro de E/S
;*****************************************
PORTD           equ $0003           ; Registro del Puerto D
DDRD            equ $0007           ; Registro de Direccionamiento del Puerto D
```

```
;===========================================
;                Registro de Configuraciones
;===========================================
CONFIG1       equ $001F              ; Puntero - Reg. de Configuración

;===========================================
;                        Variables Globales
;===========================================
PWM_FREQ    equ $0080              ; Frecuencia del PWM
DUTYTIME    equ $0081              ; Tiempo cíclico

;===========================================
;                         Memoria FLASH
;===========================================
FLASH_STARTequ $EC00               ; Puntero - Mem.FLASH

;===========================================
;                        Vectores de Usuario
;===========================================
RESET_VEC   equ $FFFE              ; Puntero del RESET

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

;===========================================
; OBJETIVO    : Inicio del programa
;               SP   = STACK POINTER
;               COPD = WATCHDOG
;===========================================
START
        rsp                        ; inic.Stack = $00ff
        bset COPD,CONFIG1          ; desactiva watchdog
        jsr Init_Ports             ; Subr,Inic. PUERTO
        jsr Led_on
```

```
;=============================================
; PROPÓSITO  : Ciclo interminable, prende y
;              apaga un LED a intervalos de
;              tiempos dada por las varia-
;              bles DutyTime y Freq_Pwm.
;              Modulando el brillo del LED
;=============================================
MORE_BRIGHT
        lda DUTYTIME            ; Carga el tiempo cíclico
        add #ONE                ; añade 1
        sta DUTYTIME            ; almacena en la variable
        cmp #DUTYTIMEMAX        ; Compara con el tiempo del ciclo
        bhi LESS_BRIGHT         ; Si es mayor, da menos brillo
        jsr SOFT_PWM            ; Salta a efectuar un PWM
        bra MORE_BRIGHT         ; Dale más brillo
LESS_BRIGHT
        lda DUTYTIME            ; Carga el tiempo cíclico
        sub #ONE                ; resta 1
        sta DUTYTIME            ; acumula en tiempo
        cmp #DUTYTIMEMIN        ; compara con 1
        blo MORE_BRIGHT         ; Si es menor, da más brillo
        jsr SOFT_PWM            ; Salta a efectuar un PWM
        bra LESS_BRIGHT         ; Dale menos brillo

;=============================================
; INIT_PORTS  : Inicializa variables y regis
;               tros.
; OBJETIVO    : Inicializa los registros de
;               direccionamiento.
;               PORTD7 = OUTPUT
;               Frecuencia del PWM a 32 Hz
; ENTRADA     : Ninguna
; SALIDA      : Ninguna
; REGISTROS
; AFECTADOS   : DDRD, RAM
;=============================================
Init_Ports
        bset BIT7,DDRD          ; Fija PD7 = Salida
        mov #HZ32,PWM_FREQ      ; Carga con 32ms para una freuencia de 31.25
                                ; Hz
        clr DUTYTIME            ; Inicializa Tiempo en 0
        rts                     ; retorna
```

```
;===========================================
; LED_ON       : Enciende el LED del PORTD7
; OBJETIVO     : PORTD7 = ON
; ENTRADA      : Ninguna
; SALIDA       : Ninguna
; REGISTROS
; AFECTADOS    : PORTD
;===========================================
Led_on
        bset BIT7,PORTD             ; BIT 7 = 1 = LED = ON
        rts                         ; retorna

;===========================================
; DELAY        : Genera un retardo de tiempo
; OBJETIVO     : Retardo de tiempo, base 1ms
; ENTRADA      : H:X = Retardo en ms
; SALIDA       : H:X = 0
; REGISTROS
; AFECTADOS    : H:X
; USO          :
;                MIN = H:X = 1T
;                MÁX = H:X = 65535T
;                ldhx #500
;                jsr Delay ; retarda 0.5 seg
;===========================================

Delay
        pshx                        ; [2] Salva X en la pila
        pshh                        ; [2] Salva H en la pila
        ldhx #DELAY49152            ; [3] Carga constante de bucle fino
Delay0
        aix #-1                     ; [2] Decrementa H:X en 1
        cphx #0                     ; [3] Llegó a cero (0)
        bne Delay0                  ; [3] Si no es igual, salta a Delay0
        pulh                        ; [2] Si es igual, recupera H de la pila
        pulx                        ; [2] Recupera X de la pila
        aix #-1                     ; [2] Decrementa H:X en 1
        cphx #0                     ; [3] LLegó a cero (0)
        bne Delay                   ; [3] Si no es igual, salta a Delay
        rts                         ; [4] retorna
```

```
;===============================================
; SOFT_PWM   : Varía la intensidad de un
;              tren de pulsos.
; OBJETIVO   : Realiza el encendido o apaga-
;              do del led modulando el ancho
;              de pulso de encendido y apaga
;              do sin variar la frecuencia.
; ENTRADA    : Ninguna
; SALIDA     : ACCA = PWM_FREQ, H:X = 0
; REGISTROS
; AFECTADOS  : RAM(DUTYTIME, PWM_FREQ)
;              ACCA, H:X
;===============================================
SOFT_PWM
        clrh                    ; Borra H
        ldx DUTYTIME            ; Carga X con el valor del ciclo de trabajo
        jsr Delay               ; Salta a retardar H:X
        jsr COM_LED             ; Conmuta el PTD7
        lda PWM_FREQ            ; Carga ACCA con la frecuencia del PWM
        sub DUTYTIME            ; Resta el ciclo de trabajo
        sta PWM_FREQ            ; Lo almacena en el tiempo restante de freq.
        clrh                    ; Borra H
        ldx PWM_FREQ            ; Carga X con el tiempo restante
        jsr Delay               ; Retarda PWM_FREQ
        jsr COM_LED             ; Conmuta el estado del LED
        lda #HZ32               ; Frecuencia de 31.25 Hz
        sta PWM_FREQ            ; ACCA es almacenado en PWM_FREQ
        rts                     ; retorna

;===============================================
; COM_LED    : Apaga o prende el LED depen-
;              diendo del estado anterior.
; OBJETIVO   : Si el led del puerto D, bit 7
;              estuvo apagado, lo enciende ó
;              visceversa.
; ENTRADA    : Ninguna
; SALIDA     : ACCA = Estado del LED PTD7
; REGISTROS
; AFECTADOS  : ACCA, PORTD
;===============================================
COM_LED
        lda PORTD               ; Carga ACCA con el contenido del Puerto D
        coma                    ; Cambia 1's por 0's y 0's por 1's
        and #BIT7MASK           ; Enmascara el bit 7
        sta PORTD               ; ACCA lo almacena en el puerto D
        rts                     ; retorna
```

```
;=========================================
; OBJETIVO     :  Inicializa el Vector de Reset
;                 Arranque del programa en la
;                 memoria Flash.
;=========================================
;===== Vector de Reinicio de Sistema =======
        org RESET_VEC               ; Puntero Vec - RESET
        dw START                    ; al darse reset salta a Start
```

Listado 7. NT0010 – Puertos E-S. El programa simula un PWM o modulador por ancho de pulso, el cual atenúa y acentúa el LED PTD7. A una frecuencia fija de 31.25 Hz, se varía el ciclo de trabajo de encendido del LED, lo que causa un efecto de luz tenue a luz fuerte (ver Figura 98(b.1) y 98(b.2)).

1.10.5 Corrida o Programado de la Pastilla

(a) Ejecute el programa WinIDE.

(b) Cargue el archivo NT0010 – Puertos E-S – 12 03 04.asm.

(c) Si piensa simular, compile y presione el botón del simulador (ver NT0006).

(d) Si desea programar su pastilla cambie la compilación condicional:

```
;$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
```

(e) Compile y presione el botón del programador (ver NT0009).

(f) Programe su pastilla (NT0009).

(g) Cambie el estado del jumper a aplicación, encienda y reinicie la tarjeta.

(h) Note el efecto de atenuación y afirmación del LED de la tarjeta.

1.10.6 Conclusión

En el microcontrolador, un puerto de entrada es capaz de captar señales lógicas si es configurado como entrada; mientras tanto, si el mismo es configurado como salida, puede generar señales o actuar a un tiempo específico deseado o aproximado. Cualquiera que sea su configuración nos sirve para hacer interfase al mundo que nos rodea.

Para finalizar, se concluye el estudió del uso de los puertos de entrada – salida como interfase al mundo real, en este caso, para generar un PWM que acentúa o atenúa el brillo del LED.

1.10.7 Referencias

1.10.7.1 Información Avanzada sobre el Microcontrolador
(a) http://www.freescale.com/files/microcontrollers/doc/data_sheet/MC68HC08JL3.pdf

Pág. 30 - Registros de Puertos de Entrada – Salida y de Direccionamiento de Datos.
Págs. 137 – 147 – Información de los puertos de Entrada/Salida

1.10.7.2 Manual de Referencia del CPU
(a) http://www.freescale.com/files/microcontrollers/doc/ref_manual/CPU08RM.pdf

1.10.7.3 Página "web" sobre esta Nota Técnica
(a) http://www.geocities.com/issaiass/

1.10.8 Problemas Propuestos

1.10.8.1 Utilice el listado de la nota técnica para variar que solamente parpadee 10 veces.

1.10.8.2 Utilice el listado de la nota técnica para que en vez de variar la intensidad del LED, varíe el tiempo del ciclo (frecuencia). ¿Cuál será el resultado esperado?