

2.10 MÓDULO DE BASE DE TIEMPO – TBM GENERACIÓN DE BASE DE TIEMPO (SOLO PARA GP32)

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

2.10.1	Introducción	295
2.10.2	Registro de Control del Módulo de Base de Tiempo	296
2.10.3	Diagrama de Flujo	297
2.10.4	Código	298
2.10.5	Simulación	302
2.10.6	Conclusión	303
2.10.7	Referencias	303
2.10.8	Problemas Propuestos	303

2.10.1 Introducción

El módulo de base de tiempo no es un temporizador, y además, es independiente del módulo temporizador del sistema, lo que permite liberar los temporizadores para ejecutar otras tareas. El módulo, permite generar interrupciones a ratas periódicas seleccionadas por el usuario; las posibles funciones del Módulo de Base de Tiempo o TBM pudieran ser:

- Diagnóstico de dispositivos periódicamente: Preguntar si algún dispositivo está o no funcionando bien.
- Servir periféricos: refrescar LCD, disparar eventos, a un intervalo de tiempo fijo.
- Atender una agenda de eventos: controlar el tiempo de ejecución de un código, p.e., si tuviera un código que enviara información y otro que recibiera, ud. puede administrar cuando pueden ocurrir estos eventos.
- Salida de modos de bajo consumo como espera (“wait”) o parada (“stop”).
Nota: La salida del modo parada (“stop”) se realizará si ud. habilita el oscilador en este modo (registro de configuraciones dos, bit OSCSTOPEN).

Con el solo propósito de demostrar tal captura funcionamiento, se describe en este documento a:

- Configurar el TBM: Se configuras dichos registros para generar una señal periódica dependiendo de la selección del divisor de la tabla 65.
- Generar una interrupción: Dentro del mismo registro, configurará una interrupción y conmutará el LED de la tarjeta conectado al puerto PTA6.
- Simular: ver por medio de “breakpoints” donde se encuentran ubicados los registros correspondientes en el mapa de memoria al simular y observar su acción.

2.10.2 Registro de Control del Módulo de Base de Tiempo

2.10.2.1 Registro de Control del TBM

El módulo de generación de base de tiempo genera frecuencias de tiempo que dependen del cristal y el divisor de la tabla 65. El módulo inicia la cuenta si el bit TBON está activado (1 lógico). El bit TBIF puede ser borrado si TACK = 1. Ver sección 2.10.2.2 para la operación de interrupción.

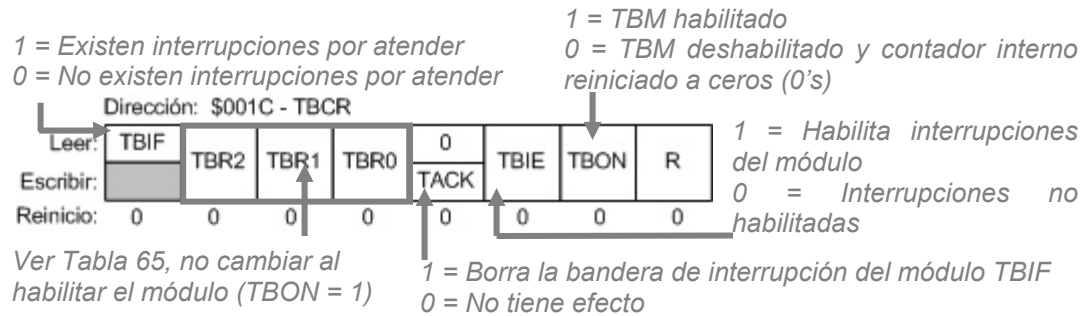


Figura 170 (izquierda). Registro de Control de Tiempo Base. El registro controla todas las actividades del TBM y genera interrupciones fijas seleccionadas por el divisor de la tabla y la ecuación 7.

Nota: La primera interrupción ocurre casi a la mitad del período e interrupciones subsecuentes a su período fijado.

Tabla 65. Divisor de Frecuencias

TBR2	TBR1	TBR0	Divisor*
0	0	0	32768
0	0	1	8192
0	1	0	2048
0	1	1	128
1	0	0	64
1	0	1	32
1	1	0	16
1	1	1	8

*Divisor utilizado en la Ecuación 7

$$f_{TBM} = f_{XTAL} / Divisor \quad \text{Ecuación 7. Frecuencia TBM}$$

f_{TBM} = Frecuencia que genera el Módulo de Base de Tiempo.

f_{XTAL} = Frecuencia del Cristal u Oscilador Externo.

Divisor = Divisor de frecuencia de la tabla 65.

R = Reservado

[No implementado] = No implementado

2.10.2.2 Vector de Interrupción del Módulo de Base de Tiempo

Tabla 66 Interrupción del Módulo de Tiempo

Bandera	Máscara	Dirección	Vector
TBIF	TBIE	FFDC	Vector de Base de Tiempo (Alto)
		FFDD	Vector de Base de Tiempo (Bajo)

Cuando el contador interno del módulo sobrepasa un límite fijado por la tabla y la ecuación 7 la bandera TBIF es levantada (1 lógico) avisando que se ha desbordado la cuenta y si las interrupciones son habilitadas (TBIE = 1) genera una rutina de atención.

2.10.2.3 Inicialización del Módulo

Utilizando un cristal de 4.9152MHz, inicialice el TBM para generar una frecuencia de 600 Hz y habilite las interrupciones. Nota: TBR[2:0] = 001.

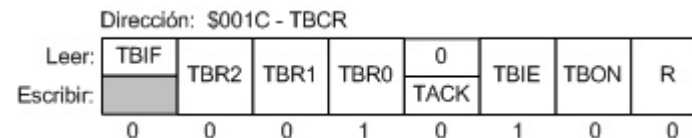


Figura 171 (inferior-derecha). Inicialización del Módulo TBM

2.10.3 Diagrama de Flujo

El siguiente programa configura el TBM a seiscientos hertz (600 Hz) y hace conmutar a una rata de trescientos hertz (300 Hz) un LED conectado al puerto A 6 del microcontrolador.

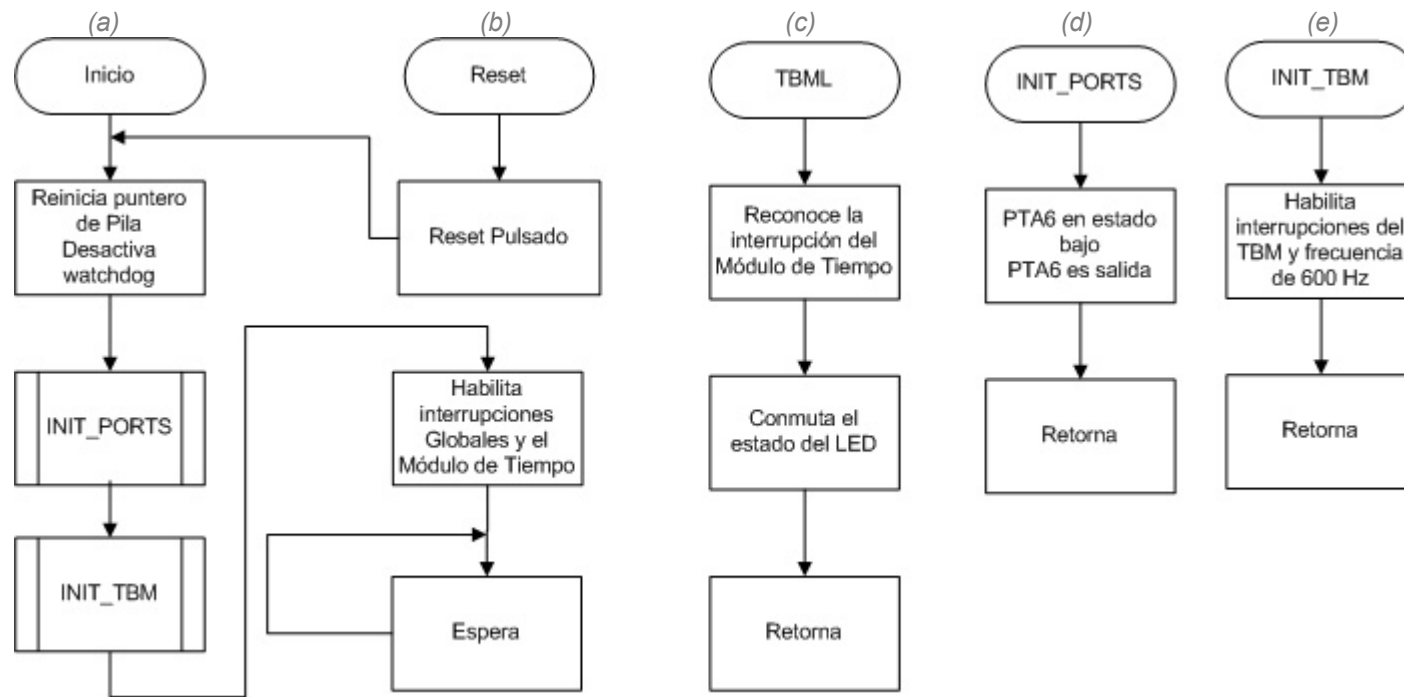


Figura 172. NT0110 – TBM. (a) Programa Principal. Inicializa el módulo de base de tiempo a seiscientos hertz, utiliza el PTA6 como salida y habilita tanto las interrupciones globales como las del módulo de tiempo. (b) Reinicio del Sistema. Al presionar “Reset”, el sistema es reiniciado sin importar su estado actual. (c) Interrupción del Módulo de Base de Tiempo. Conmuta el estado del LED PTA6 y estando el TBM a seiscientos hertz (600 Hz), genera una frecuencia de trescientos hertz (300 Hz). (d) Inicialización de Puertos. Puerto A6 es una salida en estado bajo. (e) Inicialización del Módulo de Base de Tiempo. Inicializa el módulo para que con un cristal de 4.9152 MHz, se genere una frecuencia de 600 Hz.


```
=====
;
;           Mapa de Memoria del Microcontrolador
;
=====
;
;           Registro de E/S
;
=====
PORTA      equ $0000      ; Dirección, Puerto A
DDRA       equ $0004      ; Registro de Direccionamiento, Puerto A
;
=====
;
;           Módulo de Base de Tiempo
;
=====
TBCR       equ $001C      ; Registro de Control del Módulo de Base de
;                               ; Tiempo
;
=====
;
;           Registro de Configuraciones
;
=====
CONFIG1    equ $001F      ; Vectores de configuración 1
;
=====
;
;           Memoria FLASH
;
=====
FLASH_START equ $8000      ; Puntero - Mem.FLASH
;
=====
;
;           Vectores de Usuario
;
=====
TBMH       equ $FFDC      ; Vector de Módulo de Base de Tiempo
RESET_VEC  equ $FFFE      ; Puntero del RESET
;
=====
;
; OBJETIVO : Inicio de Codif. del Ensam-
;           blador en Memoria FLASH.
;
=====
org FLASH_START      ; Inicio Mem. FLASH
```

```

;=====
; OBJETIVO   : Genera una interrupción pe-
;             : riódica cada 600 Hz y la hace
;             : ver en el puerto A6
;=====
START
    rsp                ; Inic.Stack = $00ff
    bset COPD,CONFIG1 ; Desactiva watchdog
    clra               ; Borra A
    clrx               ; Borra X
    jsr INIT_PORTS    ; Inicializa Puertos
    jsr INIT_TBM      ; Inicializa Módulo de Base de Tiempo
    cli                ; Habilita Interrupciones
    bset BIT1,TBCR    ; Habilita el TBM

ESPERA
    wait              ; Espera la interrupción.
    bra ESPERA        ; Salta al modo de bajo consumo

;=====
; INIT_PORTS : Inicializa variables y regis-
;             : tros.
; OBJETIVO   : Inicializa los registros de
;             : direccionamiento.
;             : PORTA6 = LOW
;             : PORTA6 = OUTPUT
; ENTRADA    : Ninguna
; SALIDA     : Ninguna
; REGISTROS  :
; AFECTADOS  : DDRA, PORTA
;=====
INIT_PORTS
    bclr PTA6,PORTA    ; Puerto A 6 en bajo
    bset DDRA6,DDRA    ; Puerto A 6 es salida
    rts                ; retorna

;=====
; INIT_TBM   : Inicializa el módulo de base
;             : de tiempo
; OBJETIVO   : Inicializa el módulo para ge-
;             : nerar una frecuencia de:
;             : ftbm = fxtal/divisor
;             : ftbm = 4915200/8192 = 600 Hz
;             :
; ENTRADA    : Ninguna
; SALIDA     : Ninguna
; REGISTROS  :
; AFECTADOS  : TBCR
;=====
INIT_TBM
    mov #{TBR0|TBIE},TBCR ; TBM = 600 Hz e interrupciones
                           ;habilitadas
    rts                    ; Retorno de la subrutina.

```

```
=====
;
; TBML      : Módulo de Base de Tiempo
; OBJETIVO  : Reconoce la interrupción y
;            conmuta el estado del LED en
;            el PTA6
; ENTRADA   : Ninguna
; SALIDA    : Ninguna
; REGISTROS
; AFECTADOS : TBCR, PORTA, ACCA
=====
```

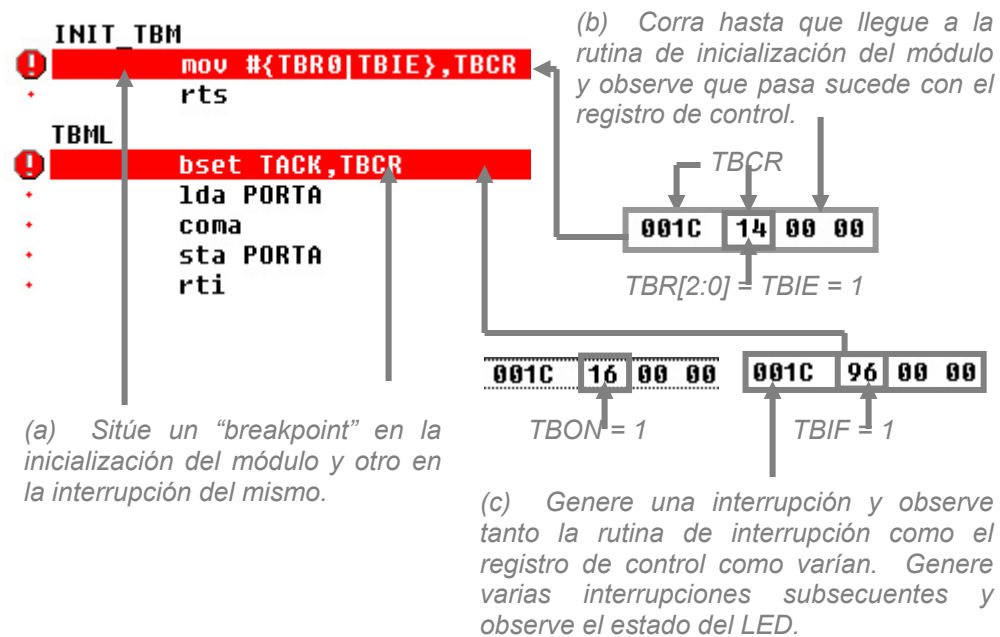
```
TBML
    bset TACK,TBCR          ; Reconoce la interrupción
    lda PORTA              ; Lee el puerto
    coma                   ; cambia 1's por 0's
    sta PORTA              ; PORTA = A
    rti                    ; Retorno de la interrupción.
```

```
=====
; OBJETIVO  : Inicializa el Vector de Reset
;            Arranque del programa en la
;            memoria Flash y búsqueda de la
;            interrupción del módulo de
;            base de tiempo.
=====
```

```
===== Vector del Módulo de Base de Tiempo =====
    org TBMH                ; Vector de Sobreflujo del TBM
    dw TBML                 ; Sobreflujo de Conteo
===== Vector de Reinicio de Sistema =====
    org RESET_VEC          ; Puntero Vec - RESET
    dw START               ; al darse reset salta a Start
```

Listado 19. NT0110 – TBM. El programa principal inicializa el Módulo de Base de tiempo para seicientos hertz (600 Hz) de frecuencia y la subrutina hace conmutar el LED PTA6 lo que genera una frecuencia neta del LED de trecientos hertz (300 Hz).

2.10.5 Simulación



(a) Sitúe un "breakpoint" en la inicialización del módulo y otro en la interrupción del mismo.

(a) Inicie WinIDE.

(b) Cargue el archivo NT0110 – TBM – 16 07 04.asm.

(c) Compile.

(d) Entre al simulador.

(e) Añada "breakpoints" según la figura 173(a).

(f) Corra su programa y en la sección de llamado a la rutina de inicialización del módulo de tiempo (INIT_TBM) observe la ejecución de cada instrucción.

(g) Presione una tecla y espere a la llegada del "breakpoint". Observe el estado del LED; ejecute dos o más pasos y observe nuevamente que sucede.

(h) Genere otra interrupción y repita el paso (g).

(i) Si desea "quemar" su pastilla, revisar la NT0009, Sección 1.9.5.

Figura 173. Simulación del Programa Principal. (a) "Breakpoints". Sitúe "breakpoints" para visualizar la configuración del módulo y su interrupción. (b) Rutina de Inicialización de Módulo de Base de Tiempo. Pasos a seguir para inicializar el módulo. (c) Interrupción. Espere a que se genere una interrupción y observe que sucede mientras la ejecuta.

2.10.6 Conclusión

En situaciones donde necesitemos el control con temporizador y no se desee comprometer su uso, este microcontrolador (GP32) provee un módulo independiente que puede realizar actividades esporádicas como el de refrescado de una pantalla o revisión de actividades del sistema.

Para efecto práctico se utiliza el módulo en generación de encendido y apagado de un LED, pero la misma inicialización puede servir para “despertar” el sistema de los modos de bajo consumo como “stop” o “wait” los cuales solo consumen unos cuantos microamperios.

Finalmente, el uso de la simulación, nos involucra más internamente en la operación del módulo y entender la mecánica del sistema. El módulo TBM es exclusivo (hablando de los cuatro microcontroladores cubiertos en la NT0107) del GP32.

2.10.7 Referencias

2.10.8.1 Información Avanzada sobre el Microcontrolador

(a) http://www.freescale.com/files/microcontrollers/doc/data_sheet/MC68HC908GP32.pdf

Págs. 335 a 339 – Módulo de Base de Tiempo (TBM).

2.10.8.2 Manual de Referencia del CPU

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

2.10.8.3 Página “web” sobre esta Nota Técnica

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

2.10.8 Problemas Propuestos

2.10.8.1 Inicialice el Módulo de Base de Tiempo para que con un divisor de 32768 y haga que su LED se encienda a intervalos de un (1) segundo.

2.10.8.2 Del mismo problema anterior, ahora conecte cuatro (4) LEDs y haga que cada uno encienda cada segundo; p.e., el primero, el primer LED el primer segundo, el segundo LED en el segundo dos y repetir sucesivamente.

2.10.8.3 Utilice el Módulo de base de tiempo y haga que la frecuencia varíe, es decir, cambie a todos los divisores posibles de TBR[2:0] de cero (0) hasta siete (2).

Nota: Rápidas frecuencias del TBM por un tiempo prolongado pueden causar un desborde de la pila; p.e., se recomienda para un cristal de 4.9152 MHz no pasar de un TBR[2:0] = 010 (ftbm = 2.4kHz)