

2.12 MÓDULO DE AUTORECUPERACIÓN – AWU PARA MICROCONTROLADORES DE LA SERIE Q

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

2.12.1	Introducción	308
2.12.2	Registro de Control del Módulo de Autorecuperación	309
2.12.3	Diagrama de Flujo	310
2.12.4	Código	312
2.12.5	Conclusión	319
2.12.6	Referencias	319
2.12.7	Problemas Propuestos	319

2.12.1 Introducción

Aclaratoria: La siguiente Nota Técnica fue utilizada del sitio de la Cía. Motorola, AN2310; y fue ligeramente modificada para implementar el módulo AWU.

Las aplicaciones más comunes de equipos portátiles como controles remotos, teclados inalámbricos; equipos de sistemas especiales como detectores de humo, recaen bajo el mismo lecho: utilidad de baterías. El utilizar baterías necesariamente significa administrar el consumo de las mismas para que dure tanto su vida de utilidad como la durabilidad de un producto en el mercado.

La familia HC08, serie Q, posee un módulo especial que despierta del modo de bajo consumo (STOP), y bastando con unos cuantos microamperios, mantienen en operación al microcontrolador.

La nota abarca la configuración del modo de bajo consumo (STOP) y de autorecuperación (AWU), en más detalle:

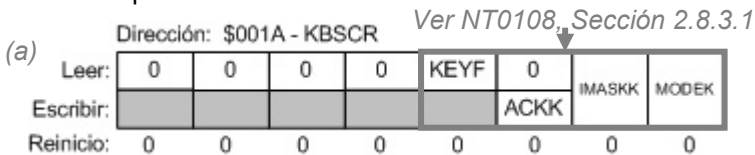
- Configuración del Módulo “Auto Wake Up”: Configuración del módulo de autorecuperación e instrucción de bajo consumo, indispensable para aplicaciones de baterías.
- Código: El código es basado, en el AN2310, el cual simula una alarma casera por medio de un LED y cuyo sensor (entrada) es simulado por un potenciómetro.
- Simulación: El simulador WinIDE presenta una falla del registro CONFIG1, solo configura para deshabilitar el watchdog, por consiguiente, esta opción no es estudiada.

Nota: Se asume un dominio neto de la NT0108 y NT0111, los registros asociados son muy similares y se explica por diferencia.

2.12.2 Registro de Control del Módulo de Autorecuperación

2.12.2.1 Registros de Control del Módulo AWU

El módulo de autorecuperación genera una interrupción periódica que retira al microcontrolador fuera del modo de parada (STOP), sin necesidad de una interrupción externa.



1 = Interrupciones del Módulo de Autorecuperación, pendientes.
0 = No hay interrupciones a procesar.

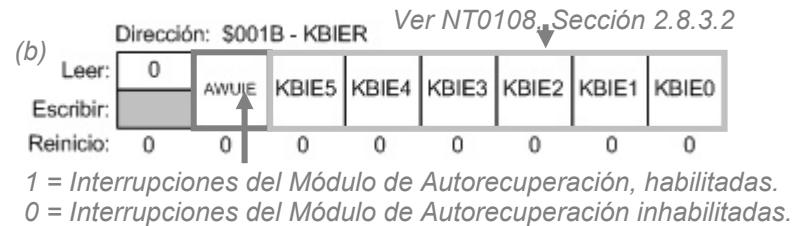


Figura 176. Registros del Módulo de Autorecuperación. (a) Registro de Estado y Control de Teclado. Ver sección 2.8.3.1 de la NT0108. (b) Registro de Control de Interrupción de Teclado. El bit AWUIE habilita las interrupciones del módulo lo que permite al microcontrolador generar una interrupción y despertarse del modo STOP. (c) Registro del Puerto A. El bit AWUL se levanta (1 lógico) si hay una interrupción a procesar.

Cuando el microcontrolador entra en modo de parada (STOP) y si las interrupciones del módulo AWU están habilitadas el módulo habilita un *oscilador interno* inherente y exclusivo del mismo a 32 kHz; el módulo empieza una cuenta interna preajutable por el registro de configuraciones uno (1), bit COPRS (ver NT0111), cuyos períodos de autorrecuperación son de aproximadamente 16 ms ($512 \times f_{OSC INT}$) y 512 ms ($16384 \times f_{OSC INT}$) a 5V de operación. Vencido el tiempo, se da una interrupción que se memoriza en el registro del puerto, bit AWUL; este bit se puede borrar por medio de una señal de reconocimiento (ver NT0108), mediante el bit ACKK.

2.12.2.1 Configuración del Módulo de Autorecuperación

Habilite el modo de autorrecuperación para salir del modo de bajo consumo en los períodos más largos posibles, habilite el pin de reinicio e igualmente habilite los modos de LVI a 5V, la instrucción STOP y desactive el "watchdog". Nota: Ayudese con la NT0111 y la NT0108.

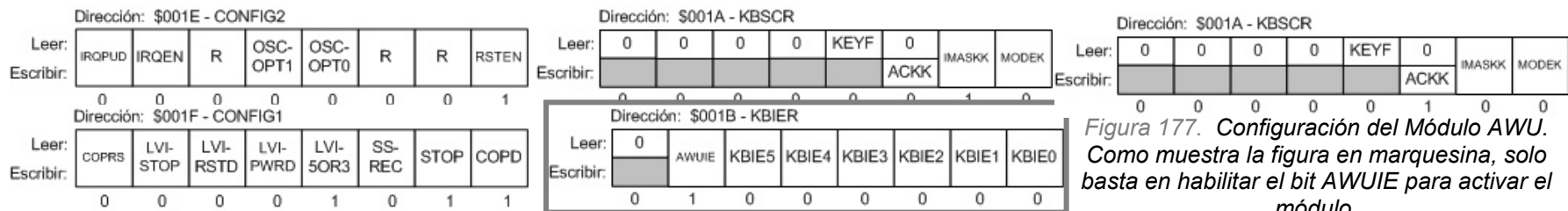
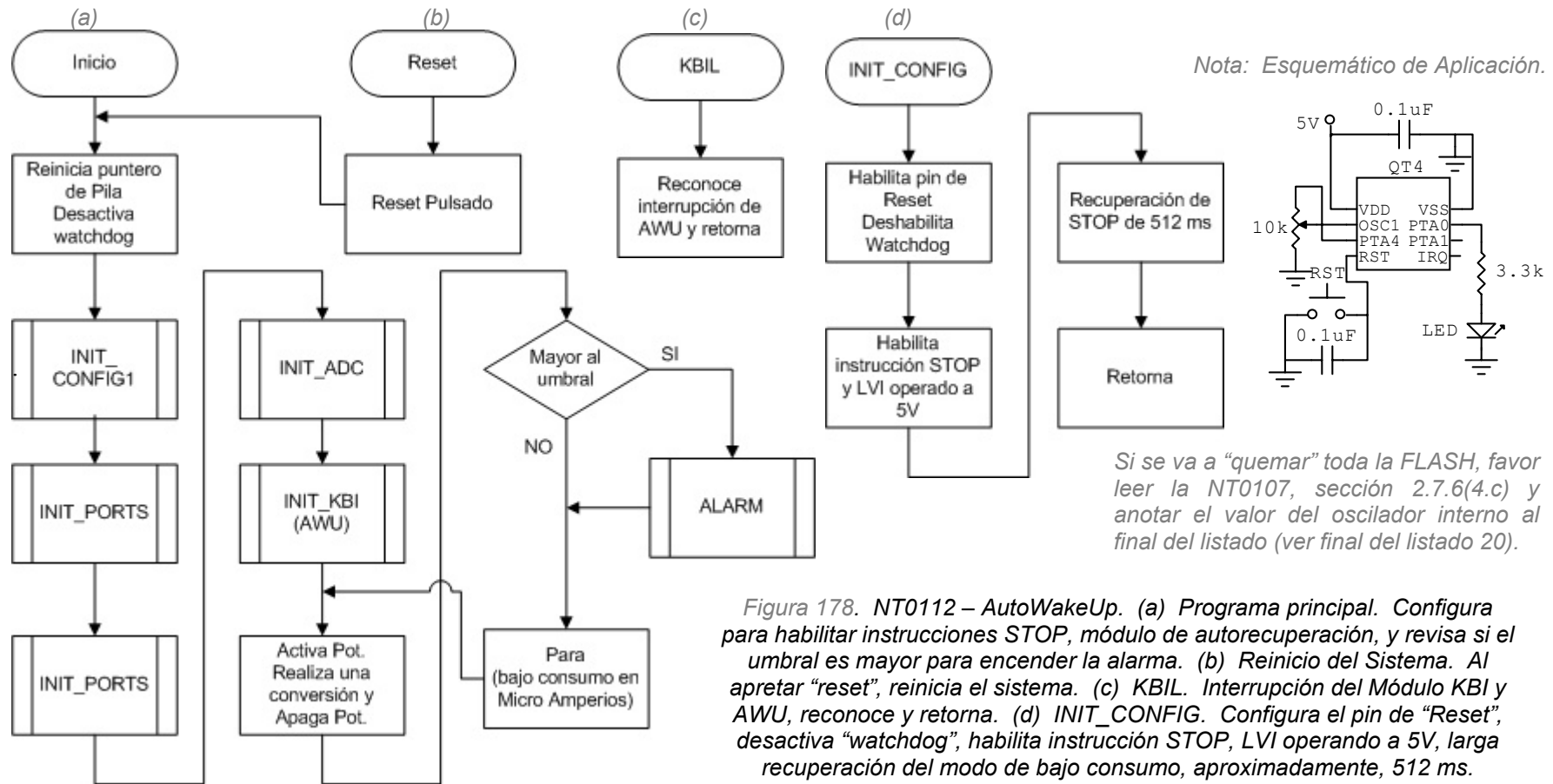


Figura 177. Configuración del Módulo AWU. Como muestra la figura en marquesina, solo basta en habilitar el bit AWUIE para activar el módulo.

2.12.3 Diagrama de Flujo

El programa configura el microcontrolador QT4 en el modo de bajo consumo y verifica si el umbral, dado por el potenciómetro, es suficiente para que se active una señal de alarma visualizada por un LED, de lo contrario, sigue durmiendo en el modo STOP y despierta después de ciertos ciclos.



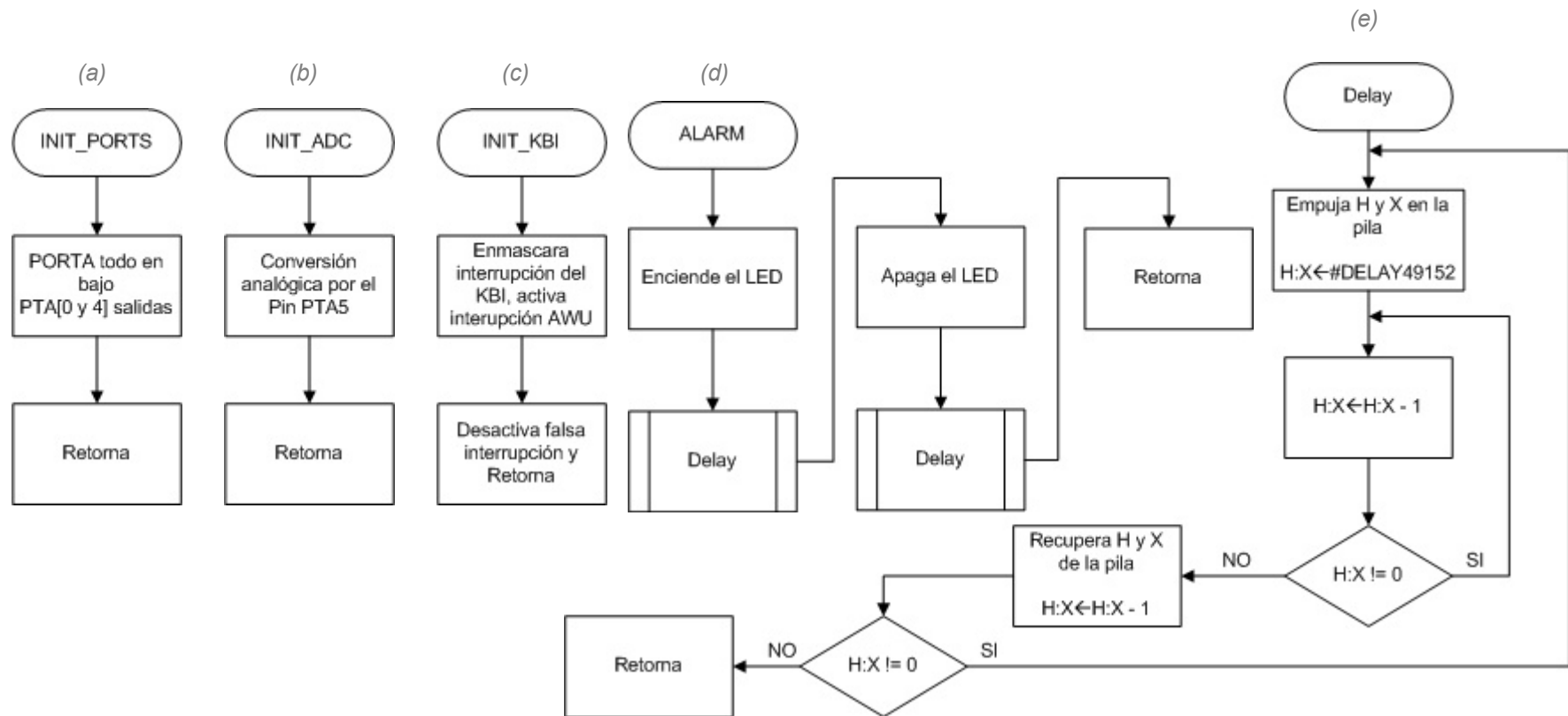


Figura 179. NT0112 – “AutoWakeUp” (continuación, subrutinas). (a) INIT_PORTS. Inicializa el puerto A en bajo y luego las salidas del potenciómetro y el LED. (b) INIT_ADC. Configura para conversión en el puerto A5. (c) INIT_KBI. Habilita la interrupción del módulo de autorecuperación. (d) ALARM. Envía una señal de activación y desactivación al LED simulando una alarma, cuya conexión pudiese ser un piezoeléctrico para llamar la atención. (e) Delay. Subrutina de retardo programable utilizada por la subrutina ALARM, espera de base de 1 ms.

2.12.4 Código

```

=====
; ARCHIVO      : NT0112 - AutoWakeUp - 19 07 04.asm
; PROPÓSITO   : Alarma de nivel. Mientras que el ADC adquiere el resultado
;              : de un transductor (simulado por un potenciómetro); una
;              : alarma suena (simulada por un LED) intermitentemente
;              : avisando que se ha sobrepasado un límite de seguridad.
;
;              :
;              : Configurar el QT4 en modo STOP y demostrar la capacidad del
;              : AWU (Auto Wake Up Module), el cual permite recuperarse del
;              : modo de Bajo Consumo.
;              : 1 - Terminal variable del Potenciómetro conectado al PTA5
;              : 2 - Potenciómetro activado por el pin PTA4
;              : 3 - Alarma (LED) en terminal PTA0
;              : 4 - Reset en terminal PTA3
;
;              :
;              : P.D.: Realizar las conexiones adicionales del esquemático.
;
; ACLARACIÓN: Esta nota se basa completamente en el AN2310 -
;              : MC60HC908QT4/D - Low Power Application y es ligeramente
;              : modificada para la explicación del Módulo Auto Wake Up.
;
; REFERENCIA:
;              : MCU68HC908QY4, ...QT1 Datasheet
;              : http://www.freescale.com/files/microcontrollers/doc/
;              : data\_sheet/MC68HC908QY4.pdf
;              : Págs. 49 @ 53 - Módulo de Autorecuperación
;
; LENGUAJE    : IN-LINE ASSEMBLER
;
;-----
; HISTORIAL
; DD MM AA
; 19 07 04 Creado.
; 15 09 04 Modificado.
;
;-----
; Pasos para iniciar la AWU:
;
;              :
;              : 1 - Enmascarar interrupciones por teclado (IMASKK = 1)
;              : 2 - Habilitar la interrupción del módulo de autorecuperación
;              :     KBIER_AWUIE = 1 (Bit 6 del registro)
;              : 3 - Reconocer la falsa interrupción (ACKK = 1)
;              : 4 - Borrar la máscara de Teclado (IMASKK = 0)
;
;-----

```

```

;=====
;
;           Definiciones del Usuario
;=====
DELAY98304 equ $012C           ; Constante de Retardo a 9.8304 MHz
NULL       equ 0T              ; Valor nulo para inicializar el puerto
BIT0       equ 0T              ; Puerto A, Bit 0
RSTEN      equ 0T              ; CONFIG2, Bit de Habilitación de Reset, Bit 0
PORTA0     equ 0T              ; PORTA, Bit 0 del Registro
PORTA4     equ 4T              ; PORTA, Bit 4 del Registro
AWUIE      equ 6T              ; KBSCR, bit de recuperación de modos stop
; habilitado
COCO       equ 7T              ; ADSCR, Bit de Conversión Completa, Bit 7
MS200      equ 200T            ; A retardar 200 ms ó 0.2 seg
MS100      equ 100T           ; A retardar 100 ms ó 0.1 seg
CH1        equ %00000010      ; ADSCR, Bit de Canal 0, Bit 1 ON
CH0        equ %00000001      ; ADSCR, Bit de Canal 0, Bit 0 ON
ADC3       equ CH1|CH0        ; ADSCR, PTA5 = ADC
ADIV1      equ %01000000      ; ADIV1, Bit 6 ON, ADC a 0.8 MHz
DDRA4      equ %00010000      ; DDRA, Bit 4 ON
DDRA0      equ %00000001      ; DDRA, Bit 0 ON
LVI5OR3    equ %00001000      ; CONFIG1, Bit de Operación del LVI, LVI @
; 5V
STOP       equ %00000010      ; CONFIG1, Bit de Habilitación de modo
; STOP, BIT 1 ON
COPD       equ %00000001      ; CONFIG1, Bit de Inhabilitación de Watchdog,
; BIT 0 ON
IMASKK     equ %00000010      ; KBSCR, Bit de Máscara del Registro
; KBSCR, Bit 1 ON
ACKK       equ %00000100      ; KBSCR, Bit de Reconocimiento de
; Interrupción, Bit 2 ON

;=====
;
;           Mapa de Memoria del Microcontrolador
;=====
;=====
;
;           Registro de E/S
;=====
PORTA       equ $0000          ; Registro del Puerto A
DDRA       equ $0004          ; Registro de Direccionamiento del Puerto A

;=====
;
;           Registro de Configuraciones
;=====
CONFIG2     equ $001E          ; Vectores de configuración 2
CONFIG1     equ $001F          ; Vectores de configuración 1

;=====
;
;           Módulo de Interrupción por Teclado
;=====
KBSCR       equ $001A          ; Registro de Estado y Control del KBI
KBIER      equ $001B          ; Registro de Control de Interrupciones de
; Teclado

```

```

;=====
;
;                               Registro ADC
;=====
ADSCR      equ $003C           ; Dirección, registro de estado y control del
;                               ; ADC
ADR        equ $003E           ; Registro de Datos del ADC
ADICLK     equ $003F           ; Registro de Reloj de Entrada del ADC

;=====
;
;                               OSCTRIM
;=====
OSCTRIM    equ $FFC0           ; Vector del Registro del Oscilador

;=====
;
;                               Memoria FLASH
;=====
FLASH_START equ $EE00         ; Puntero - Mem.FLASH

;=====
;
;                               Vectores de Usuario
;=====
KBIH       equ $FFE0           ; Auto Wake Up (Alto)
RESET_VEC  equ $FFFE           ; Puntero del RESET

;=====
; OBJETIVO : Genera una interrupción pe-
;           ; ródica cada 600 Hz y la hace
;           ; ver en el puerto A6
;=====
org FLASH_START           ; Inicio Mem. FLASH

```

```

;=====
; OBJETIVO   : La alarma se activa si en el
;              valor registrado de la con-
;              versión su bit más signifi-
;              cativo está habilitado.
;=====
START
    rsp                ; Inic.Stack = $00ff
    jsr INIT_CONFIG    ; Inicializa registros de Configuración
    jsr INIT_PORTS     ; Subr,Inic. PUERTO
    jsr INIT_ADC       ; Inicializa ADC
    jsr INIT_KBI       ; Inicializa AWU (AutoWakeUp)
    cli                ; Habilita interrupciones

READY
    bset PORTA4,PORTA  ; Pot. activado
    mov #ADC3,ADSCR    ; Conversión única en PTA5.
    brclr COCO,ADSCR,* ; Espera la conversión
    bclr PORTA4,PORTA  ; Pot. desactivado
    lda ADR             ; Lee el valor de la conversión
    bpl DORMANT        ; Salta a Dormir si ADR (BIT7) = 1
    jsr ALARM          ; La Alarma avisa (LED conmuta)

DORMANT
    stop               ; Modo de bajo consumo, aprox. 6uA.
    bra READY         ; Nueva conversión

;=====
; INIT_CONFIG : Inicializa registro de confi-
;              guración.
; OBJETIVO   : Habilita pin Reset, LVI a 5 V
;              Recuperación larga de STOP,
;              instrucción STOP y Watchdog
;              inhabilitado.
; ENTRADA   : Ninguna
; SALIDA    : Ninguna
; REGISTROS
; AFECTADOS : CONFIG2, CONFIG1
;=====
INIT_CONFIG
    bset RSTEN,CONFIG2 ; Habilita Pin de Reset
    mov #{LVI5OR3|STOP|COPD},CONFIG1
                                ; LVI a 5V, instrucción STOP
                                ; habilitada y Watchdog inhabilitado
    rts                          ; retorna

```



```

=====
;
; INIT_PORTS : Inicializa variables y re-
;             gistros.
; OBJETIVO   : Inicializa los registros de
;             direccionamiento.
;             PTA = LOW
;             PTA4 y PTA0 = OUTPUT
; ENTRADA    : Ninguna
; SALIDA     : Ninguna
; REGISTROS
; AFECTADOS : PORTA
=====
INIT_PORTS
    mov #NULL,PORTA           ; Inicialmente bajos
    mov #{DDRA4|DDRA0},DDRA   ; PTA0 (LED) y PTA4 (pot)
                                ; son salidas
    rts                       ; retorna
=====
;
; INIT_ADC   : Inicializa el ADC
; OBJETIVO   : Conversión Continua e
;             Interrupciones
; ENTRADA    : Ninguna
; SALIDA     : Ninguna
; REGISTROS
; AFECTADOS : ADSCR, ADICK
=====
INIT_ADC
    mov #ADIV1,ADICK          ; Reloj del ADC = XTAL/2^(ADIV1+2)
                                ; 0.8 MHz de reloj de ADC
    rts                       ; retorna
=====
;
; INIT_KBI   : Inicializa registro de
;             autorecuperación de stop
; OBJETIVO   : Modo de Autorecuperación a
;             instrucción stop habilitado
; ENTRADA    : Ninguna
; SALIDA     : Ninguna
; REGISTROS
; AFECTADOS : KBSCR, KBIER
=====
INIT_KBI
    mov #IMASKK,KBSCR         ; Enmascaro las interrupciones por
                                ; teclado
    bset AWUIE,KBIER         ; Auto Wake Up habilitado
    mov #ACKK,KBSCR           ; Borro la falsa interrupción
    rts                       ; Retorno de la subrutina.
=====

```

```

;=====
;ALARM      : Simula una alarma por el
;            LED.
;OBJETIVO   : El LED PTA0, conmuta a inter
;            valos de 500 ms ON y 100 ms
;            OFF.
;ENTRADA    : Ninguna
;SALIDA     : Ninguna
;REGISTROS
;AFECTADOS  : PORTA, H:X
;=====

```

```

ALARM
    bset PORTA0,PORTA      ; Enciende el LED
    ldhx #MS200            ; 200 ms
    jsr Delay              ; Retarda 0.2 seg
    bclr PORTA0,PORTA     ; Apaga el LED
    ldhx #MS100           ; 100 ms
    jsr Delay              ; Retarda 0.1 seg
    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 #DELAY98304    ; [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

```

```

;=====
;KBIL      : AutoWakeUp, Interrupción
;OBJETIVO  : Reconoce la interrupción y
;           retorna.
;ENTRADA   : Ninguna
;SALIDA    : Ninguna
;REGISTROS
;AFECTADOS : KBSCR
;=====
KBIL
    mov #ACKK,KBSCR      ; Reconoce la interrupción
                        ; borra AWUL
    rti                 ; retorna

;=====
;OBJETIVO  : Inicializa el Vector de Reset
;           Arranque del programa en la
;           memoria Flash y vector de
;           AWUL simulado.
;=====
;===== AutoWakeUp, Vector de Rehabilitación del sistema =====
    org KBIH            ; Puntero Vec - AWU
    dw KBIL             ; Recuperación del modo stop
;===== Vector de Reinicio de Sistema =====
    org RESET_VEC      ; Puntero Vec - RESET
    dw START           ; al darse reset salta a Start
;===== Contenido del Oscilador Interno =====
    org OSCTRIM        ; Dirección del Oscilador Interno
    fcb $86,$FF       ; Ajustar según el valor leído
                        ; dado por el fabricante

```

Listado 20. NT0112 – AWU. El programa principal inicializa el Módulo de Base de tiempo para seiscientos hertz (600 Hz) de frecuencia y la subrutina hace conmutar el LED PTA6 lo que genera una frecuencia neta del LED de trescientos hertz (300 Hz)

2.12.5 Conclusión

La mayoría de los microcontroladores de la familia HC08 escapan de reinicios del sistema en modo STOP, esta es una característica infinitesimalmente importante si se requiere conservar la vida útil de un producto que trabaje con baterías.

En la aplicación se utiliza el modo “Auto Wake Up” para generar esporádicos retiros del modo de bajo consumo (STOP), y confeccionar una simple alarma casera. Finalmente, recapitulando, el modo AWU, nos permite realizar una aplicación de larga vida y bajo consumo, ideal para productos caseros.

2.12.6 Referencias

2.12.6.1 Información Avanzada sobre el Microcontrolador

(a) http://www.freescale.com/files/microcontrollers/doc/data_sheet/MC68HC908QY4.pdf

Págs. 49 a 53 – Módulo de Autorecuperación (AWU).

Pág. 56 y 57 – Registro de Configuraciones.

2.12.6.2 Manual de Referencia del CPU

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

Págs. 125 – Instrucción BPL.

2.12.6.3 Página “web” sobre esta Nota Técnica

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

2.12.7 Problemas Propuestos

2.12.7.1 Inicialice el modo “Auto Wake Up” o de Autorecuperación en modo STOP y mida el consumo del microcontrolador, compárelo con el consumo cuando corre un programa común y corriente.

Para medir el consumo, referirse al AN2310, pág. 5, “Average Current Calculations”.