

1.6 PROGRAMACIÓN DEL MICROCONTROLADOR – “SOFTWARE”

INICIACIÓN AL ENTORNO DE DESARROLLO EN ENSAMBLADOR WINIDE

ÍNDICE

Preparado por: Rangel Alvarado	1.6.1 <i>Introducción</i>	91
Estudiante Graduando de Lic. en Ing. Electromecánica	1.6.2 <i>Descarga de ICS08Z</i>	92
Universidad Tecnológica de Panamá	1.6.3 <i>Prueba de Comunicación y Diagnóstico</i>	93
Panamá, Panamá	1.6.4 <i>Compilando – CASM08Z</i>	96
“e-mail”: issaiass@cwpanama.net	1.6.5 <i>Simulando – ICS08Z</i>	97
“web site”: http://www.geocities.com/issaiass/	1.6.6 <i>Diagrama de Flujo</i>	98
	1.6.7 <i>Código</i>	99
	1.6.8 <i>Conclusión</i>	102
	1.6.9 <i>Referencias</i>	103
	1.6.10 <i>Problemas Propuestos</i>	104

1.6.1 Introducción

El documento detalla el inicio a la programación de microcontroladores en lenguaje ensamblador.

Se inicia con la descarga del “software” desde “internet” en el sitio de la compañía pemicro¹, luego de su instalación se hace una prueba básica de funcionamiento de la tarjeta y finalmente se muestra un programa ejemplo en lenguaje ensamblador que ayuda a familiarizarse con comandos básicos de programación.

El propósito general de este documento es demostrar las diferentes partes del entorno de desarrollo (IDE) para microcontroladores HC08 como lo son compilación, simulación y reglas básicas de la programación del lenguaje ensamblador.

¹ <http://www.pemicro.com/>

1.6.2 Descarga de ICS08Z

En esta sección refiérase a la figura 67.

(1) Regístrese en: https://www.pemicro.com/login_new_user_form.cfm

P&E New User Form

(2) Llene los campos

Note: If your Email address is already in our system, you do not need to fill out this form. Instead simply request for a new password to be sent to your existing email address [with this simple form](#).

(All fields but Company are required)

First Name

Last Name

Company

Phone

E-Mail

Check to receive occasional newsletters from P&E

Password (minimum 6 characters)

Verify Password

Upon submit, you will get an Email with an activation code to activate your account

P&E User Login

[Create New Account](#) or [Forgot Password?](#)

Email

Password Case-sensitive

(4) Luego, revise su correo electrónico para verificar su activación y navegue hacia la dirección: <https://www.pemicro.com/login.cfm> e inserte su nombre de usuario y contraseña.

(5) Seguido, acceda a la dirección:

http://www.pemicro.com/products/product_view.cfm?product_ID=147&menu_id=details

(6) En la sección "Downloads" Descargue

No-Cost Software	ICS08GPGTZ v. 1.53 (13271 KB)
Product Details	ICS08GPGT Software for 68HC908GP32/GT16/GT8
Product Images	ICS08GRZ v. 1.22 (13121 KB)
Add-ons	ICS08GR Software for 68HC908GR4/8
(a) Downloads	ICS08JBZ v. 1.22 (13580 KB)
Documentation	ICS08JB Software for 68HC908JB8
(b) FAQs	ICS08JLZ v. 1.53 (12337 KB)
Forums	ICS08JL Software for 68HC908JL3/JK3/JK1
Volume Pricing	

Figura 67. Descarga de WinIDE. Prosiga con los pasos 1, 2 y 3 para registrarse y el paso 4 para acceder como usuario del sitio. Además, navegue a la dirección en 5 y presione el botón del paso 6(a); esto llevará su explorador a la sección de IDEs.

Finalmente, siga el paso 6(b) y baje el software de su correspondiente microcontrolador, p.e. IC08JLZ v.153.

1.6.3 Prueba de Comunicación y Diagnóstico

En esta sección refiérase a las figuras 68 y 69.

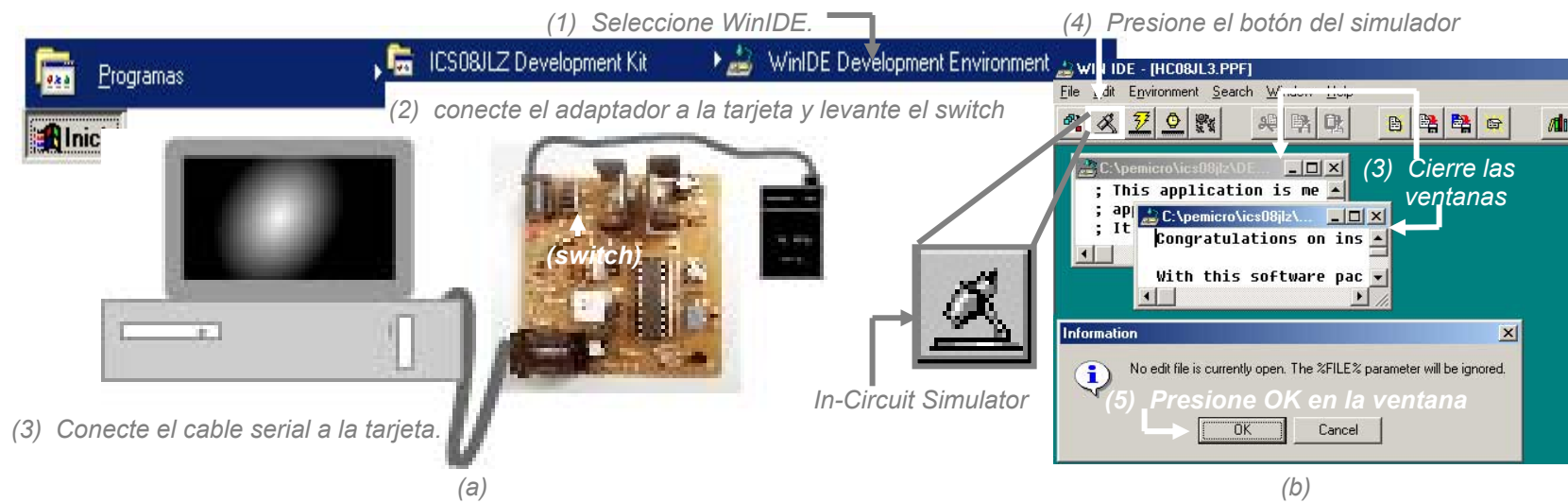
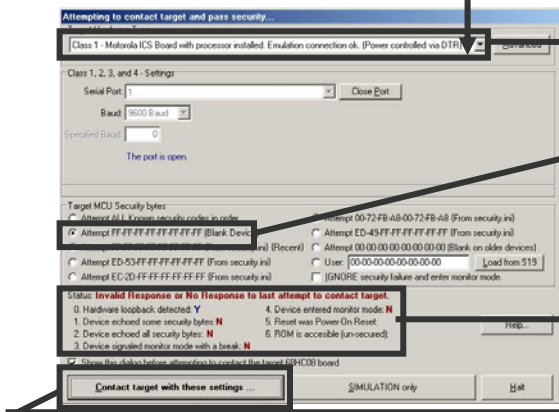


Figura 68. Prueba de Comunicación. (a) Configure su equipo. Inicie WinIDE; luego conecte a la tarjeta el adaptador, el cable serial y enciéndala. (b) Inicio de simulación. Cierre todas las ventanas y luego presione el botón de simulación. En seguida, en la ventana de información, presione OK.

(1) Seleccione el tipo de hardware.



(a)

Class 1 - Motorola ICS Board with processor installed. Emulation connection ok. (Power controlled via DTR)

(2) Ubique el código reciente

Attempt FF-FF-FF-FF-FF-FF-FF-FF (Blank Device)

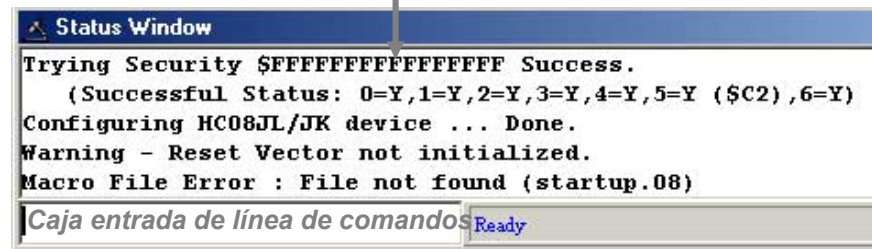
Status: **Invalid Response or No Response to last attempt to contact target.**

0. Hardware loopback detected: Y	4. Device entered monitor mode: N
1. Device echoed some security bytes: N	5. Reset was Power-On Reset:
2. Device echoed all security bytes: N	6. ROM is accesible (un-secured):
3. Device signaled monitor mode with a break: N	

Error: (4) Si el dispositivo no se conecta, apague, luego encienda la tarjeta e intente nuevamente el paso tres (3).

(3) Seleccione "Contact Target with this settings"

(5) Ventana de conexión satisfactoria, el dispositivo ha entrado al modo de simulación



(b)



(6) Seleccione su dispositivo p.e., JL3 (28 pines)


(7) En la caja de comandos escriba "chipmode" si cambiará de dispositivo.



(8) Presione OK

(9) Escriba en la caja de entrada de línea de comandos las instrucciones DDRD 80, presione "enter" y luego PORTD 80, presione "enter" y PORTD 0, observe el estado del LED verde.

Figura 69. Entrada al simulador. (a) Ventana de contactar objetivo y seguridad. Presione "Contact target with this settings..." y espere a entrar a modo monitor, si vuelve a aparecer la ventana, apague y encienda la tarjeta nuevamente; ejecute el paso tres (3). (b) Cambio de dispositivo. En la caja de texto de entrada de línea de comandos, escriba "chipmode", presione "enter" y finalmente seleccione el tipo de microcontrolador de trabajo.

- (a) Inicie WinIDE.
- (b) Conecte la fuente y el cable serial, luego encienda la tarjeta.
- (c) Cierre todas las ventanas y luego presione el botón del simulador [].
- (d) Presione OK y luego de aparecer la ventana de contacto del dispositivo presione "Contact target with this settings...".
- (e) Si el dispositivo recae nuevamente en la ventana de contacto, apague la tarjeta y enciéndala nuevamente, repita el paso (d).
- (f) Al entrar al simulador, si desea cambiar de dispositivo escriba "chipmode" para cambiar de dispositivo (JK1/JK3 = 20 pines; JL3 = 28 pines), presione OK.
- (g) Finalmente seleccionado el dispositivo, escriba las siguientes instrucciones:
- | | | |
|--------------|-----------|---|
| <i>ddrd</i> | <i>80</i> | (PIN7 del PUERTO D es salida) |
| <i>portd</i> | <i>80</i> | (PIN7 del PUERTO D, led verde, encendido) |
| <i>portd</i> | <i>0</i> | (PIN7 del PUERTO D, led verde, apagado) |

Repita las últimas dos instrucciones para ver nuevamente los cambios del LED en la tarjeta.

- (h) Cierre la ventana del simulador.

1.6.4 Compilando – CASM08Z

En esta sección refiérase a las figuras 70 y 71.

- (1) Inicie WinIDE (figura 70(1)). (2) Abra el archivo en ensamblador (File/Open)



- (3) Presione F4 o el botón del compilador (4) Espere el final de la compilación

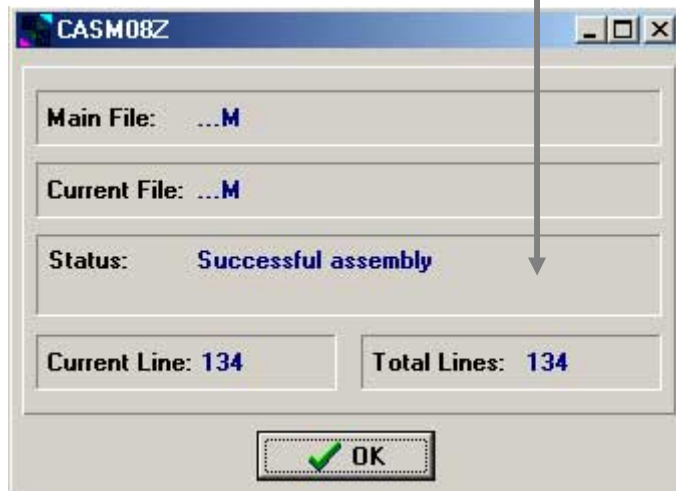



Figura 70. Compilación de Programas. Acceda WinIDE y abra el archivo a compilar, luego presione F4 o el botón del compilador y espere a que termina la acción.

- (a) Inicie WinIDE, como se muestra en la figura 70(1).
 (b) Cargue el siguiente archivo con nombre *NT0006 – Software – 01 02 04.asm*.
 (c) Presione F4 o el botón del compilador [].
 (d) Espere el final de la compilación.

El trabajo directo de un compilador es convertir las instrucciones de un lenguaje y traducirlas a otro lenguaje, en este caso, a un lenguaje de máquina.

1.6.5 Simulando – ICS08Z

En esta sección refiérase a la figura 71

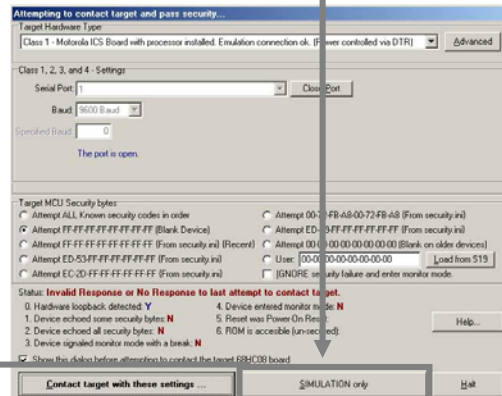
(1) Presione el botón de simulación



Solo simulación: No necesita tener la tarjeta conectada a la PC.



(2) En la ventana de contactar el dispositivo, presionar el botón de solo simulación.



(a)

Reinicio

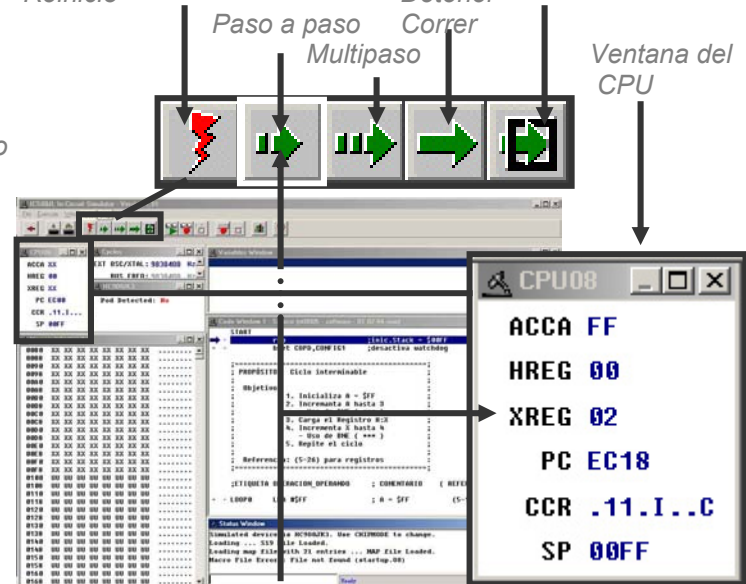
Paso a paso

Detener

Multipaso

Correr


Ventana del CPU



(3) Presione el botón de simulación paso a paso varias veces y vea el cambio en la ventana del CPU.

(b)

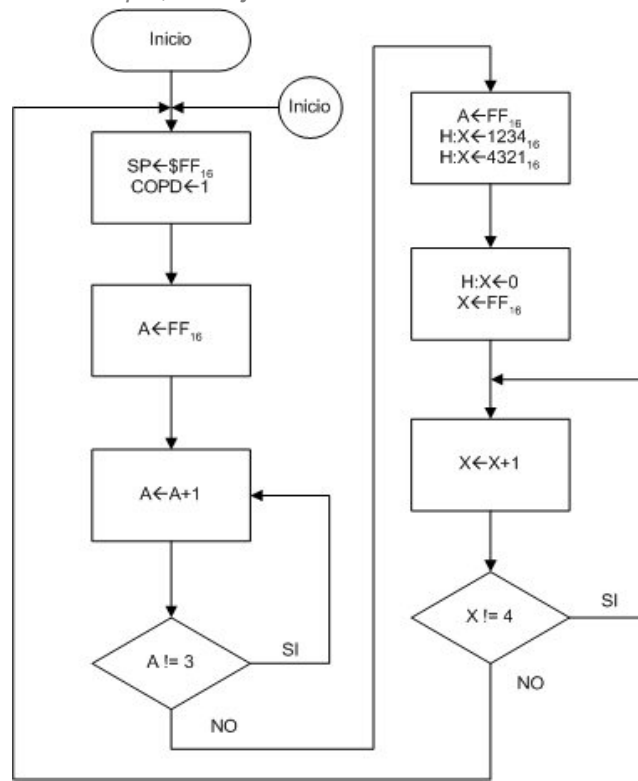
Figura 71. Simulación del Programa. (a) Contacto de solo simulación. Presione F6 o el botón de simulación y en la ventana de contacto del objetivo, presione “Simulation only”. (b) Entorno de Simulación. Ejecute la instrucción de paso a paso y conforme el programa avance observe que acción directa tiene sobre la ventana del CPU.

- (a) Desconecte la tarjeta de desarrollo de la PC y apáguela.
- (b) Siga los pasos de compilación de la sección 1.6.4.
- (c) Presione el botón de simulación [].
- (d) En la ventana de contactar objetivo (figura 71(a)), presione “Simulation only”.
- (e) Con el botón de simulación paso a paso observe el cambio de la ventana del CPU.
- (f) Siga simulando paso a paso para entender la lógica del programa.

1.6.6 Diagrama De Flujo

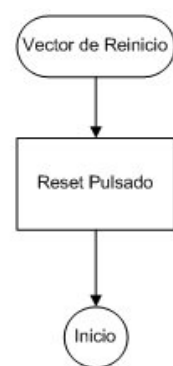
El siguiente programa ejemplifica el manejo de las instrucciones *bne* y asignaciones.

Rutina Principal, manejo de instrucciones básicas



(a)

Reinicia el sistema



(b)

Figura 72. NT0006 – “Software”. (a) Rutina principal de manejo de instrucciones básicas. Saltos y asignaciones. (b) Vectores de Interrupción. Reinicia el sistema en el punto de inicio.

1.6.7 Código

```

=====
; ARCHIVO      : NT0006 - Software - 01 02 04.asm
; PROPÓSITO   : Manejo de estructuras de control y de incremento.
;              Incrementar, Comparar y Saltar ( Ver uso de "BNE" )
;              1. FAMILIARIZARSE CON:
;              - LA ARQUITECTURA INTERNA DEL MICRO
;              - EVALUAR INSTRUCCIONES BASICAS
;
; PROCED.     :
;
; 1. COMPILAR EL ARCHIVO *.ASM
; 2. ACTIVAR EL SIMULADOR - SIN LA TARJETA
; 3. CARGAR EL ARCHIVO *.S19 + RESET
; 4. EN EL MEMORY WINDOW 1 REVISAR
;   - MEMORIA $EC00 = INICIO DEL PROGRAMA
;   - MEMORIA $FFFE = VECTOR DE INICIO
; 5. EJECUTAR PASO A PASO
; 6. LUEGO DE CARGAR EL PROGRAMA
;   OBSERVAR EN EL SIMULADOR LA VENTANA:
;   - MEMORY: LA LOCALIDAD $EC00 Y $FFF8
;   - CPU08: AL EJECUTAR OBSERVAR "A","H","X"
;
; REFERENCIA:
;
; 1. ICS08 Operators Manual.PDF
; 2. ICS08 Addendum - Operators Manual.PDF
; 3. Technical Data of 68HC908JK3 - 9j13r1.PDF
; 4. Advanced Information of 68HC908JK3 - J13r3.PDF
; 5. M68HC08 CPU Reference Manual - Cpu08r2.PDF
;
; LENGUAJE    : IN-LINE ASSEMBLER
;
=====
; HISTORIAL
; DD MM AA
; 01 02 02 Creado.
; 25 08 04 Modificado.
=====
;
; Definiciones del Usuario
;
=====
;El prefijo "$" simboliza un hexadecimal, sin prefijo se asume base 10.
COPD      equ $0          ; (3-31)
;(a.1) Pseudo Operación: "equate", asigna en esta caso un valor numérico de 016

```

```

;=====
;                               Mapa de Memoria del Microcontrolador
;=====
;                               Registro de Configuraciones
;=====

CONFIG1    equ $001F            ; (3-31)
; Inicializacion de Variables y Punteros (3-28)
; Nota:
; A los primeros 256 bytes de la memoria se les llama la
; página directa de memoria.

;=====
;                               Memoria FLASH
;=====
FLASH_START equ $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
;(a.2) Pseudo Operación: "originate", define el inicio del programa en memoria.

;=====
; OBJETIVO   : Inicio del programa
;               SP = STACK POINTER (3-31)
;               COPD = WATCHDOG
;=====
START
                rsp                ; inic.Stack = $00FF
                bset COPD,CONFIG1  ; desactiva watchdog

;=====
; PROPÓSITO : Ciclo interminable
;
; Objetivo   :
; 1. Inicializa A = $FF
; 2. Incrementa A hasta 3
;    - Uso de BNE ( ***)
; 3. Carga el Registro H:X
; 4. Incrementa X hasta 4
;    - Uso de BNE ( ***)
; 5. Repite el ciclo
;
; Referencia : (5-26) para registros
;=====

```

NT0006

Rev. 1 del 06.08.05

```

; (1)          (2)          (3)          (4) – Ver Tabla 40
;ETIQUETA OPERACION_OPERANDO ; COMENTARIO ( REFERENCIA )

LOOP0      LDA #$FF          ; A = $FF          (5-143)
LOOP1      INCA              ; A = A + 1       (5-186)
           CMP #3           ; A = 3 ?????    (5-185)
           BNE LOOP1        ; Subr. de RETARDO (5-118) ***
           LDA #$FF          ; A = $FF = 255  (5-186)
           LDHX #$1234       ; H:X = $1234    (5-144)
           LDHX #$4321       ; H:X = $4321    (5-144)
           LDHX #0           ; H:X = $0000    (5-186)
           LDX #$FF          ; X = $FF        (5-186)
LOOP2      INCX              ; X = X + 1      (5-186)
           CPX #4           ; X = 4 ?????    (5-185)
           BNE LOOP2        ; Subr. de RETARDO (5-184) ***
           JMP LOOP0         ; REPITE EL CICLO (5-186)

```

```

;=====
; OBJETIVO: Búsqueda del Vector de Reset
;           Arranque del programa en la memo
;           ria Flash.
;=====
;===== Vector de Reinicio de Sistema =====
           org   RESET_VEC ; Puntero Vec - RESET
           dw   START      ; al darse reset salta a Start
;(a.3) Pseudo Operación: “define word”, define una constante de dos (2) bytes

```

Listado 1. NT0006 – “Software”. Exploración de las instrucciones condicionales de control de flujo (CMP, BNE), incondicionales de control de flujo (JMP), de incremento (INC) y carga a registros (LDA, LDHX, LDX).

Tabla 40. Estructura de una Línea de Comandos en Ensamblador

Número	Descripción	Acción
1	Etiqueta	El compilador le asigna la dirección de memoria de la ubicación de la instrucción que tenga la etiqueta.
2	Operación	Indica la instrucción a ejecutarse.
3	Operando	Complemento de la operación necesaria para ejecutar una instrucción.
4	Comentario	No interfieren en el programa principal, aclaran la manera de cómo está hecho el programa.

Ejemplo:

```

...           ; Código del programa, después de un “;” siguen comentarios
JMP Suma      ; Salta a la etiqueta suma.
...           ; Código del programa.
Suma  ADD #1   ; La etiqueta suma sabe en que dirección se encuentra la
               ; operación ADD, ADD suma un operando de #1.

```

Tabla 41. Pseudo Operaciones

Etiqueta	Pseudo Operador	Acción
a.1	equ	Asocia un valor binario a una etiqueta
a.2	org	Impone el origen del programa
a.3	dw	Asigna dos bytes el cual vincula la etiqueta

Ejemplos:

- (a.1) COPD equ \$0 ; COPD= 0000 0000, "\$" es el prefijo hexadecimal.
- (a.2) org RAM ; El programa origina en la dirección de RAM (\$80).
- (a.3) dw COPD ; Define una contante numérica en memoria de valor 00₁₆

Tabla 42. Instrucciones Utilizadas bajo el programa de la NT0006

Instrucción	Complemento ejemplo		Acción
rsp			Reinicia la el SP ² pila en FF ₁₆
bset	0,\$001F	COPD,CONFIG1	Bit 0 de la dirección 001F en "1" lógico
lda	#\$0	#COPD	Carga ACCA con el número 00 ₁₆
	\$EC00	FLASH	Carga ACCA con el contenido de EC00
inca			Incrementa ACCA en uno (1)
cmp	#\$04	#NUM4	Compara ACCA con el número 80 ₁₆
	\$04	NUM4	Compara ACCA con el contenido de la dirección 0004 ₁₆
bne	\$EC00	FLASH	Si el valor comparado no es igual salta hacia la etiqueta FLASH (\$EC00)
ldhx	#\$1234		Carga H:X con el número 1234 ₁₆
ldx	#\$FF		Carga X con el número FF ₁₆
incx			Incrementa X en uno (1)
cpx	#4		Compara X con cuatro (4)
jmp	LOOP0		Salta a la etiqueta LOOP0
bne	LOOP0		Si no es igual salta a LOOP0

Nota: El símbolo de numeral (#) convierte a un número. El prefijo string (\$) representa un número en hexadecimal.

1.6.8 Conclusión

El lenguaje ensamblador es aquel que se utiliza para representar códigos binarios de una forma más conveniente, es lo más cercano al lenguaje de máquina del microcontrolador.

El documento detalló como descargar el programa de ensamblador para microcontroladores de la familia HC08.

Luego, se cumplió en desarrollar la prueba básica de encendido y apagado un LED desde el programa de simulación, simulando en circuito.

² SP = Puntero de Pila, ver ventana de CPU en la figura 46(b).

Además, el programa WinIDE se utilizó para compilar y simular una aplicación sin necesidad de poseer el microcontrolador conectado a la PC, es decir, estando en el modo de simulación.

Finalmente, un breve diagnóstico del código y operaciones/operandos fueron utilizados y analizados en el transcurso del código del programa.

1.6.9 Referencias

1.6.9.1 Sitio de la casa de “software” para programación de microcontroladores en lenguaje ensamblador y descarga del “software”

- (a) <http://www.pemicro.com/>
- (b) <http://www.pemicro.com/ics08/index.html>

1.6.9.2 Librería de documentación de microcontroladores de 8-bits de Motorola

- (a) <http://www.freescale.com/webapp/DocLibServlet?nodeId=03t3ZGpnLn>

1.6.9.3 Páginas “web” sobre microcontroladores Motorola

- (a) <http://www.mundomicro.com.ar/>
- (b) <http://www.bairesrobotics.com.ar/>
- (c) <http://www.ingdubatti.com.ar/>
- (d) <http://www.hc08.cz/>

1.6.9.4 Concurso de microcontroladores de la familia HC08, serie Q

- (a) <http://www.circuitcellar.com/fi2003/>

1.6.9.5 Manuales de Referencia de los microcontrolador

Los siguientes hipervínculos comienzan con: <http://www.freescale.com/>

- (a) /files/microcontrollers/doc/ref_manual/CPU08RM.pdf
- (b) /files/microcontrollers/doc/ref_manual/ADCRM.pdf
- (c) /files/microcontrollers/doc/data_sheet/HC908JL3AD.pdf
- (d) /files/microcontrollers/doc/data_sheet/MC68HC08JL3.pdf
- (e) /files/microcontrollers/doc/data_sheet/MC68HC908JL3E.pdf
- (f) /files/microcontrollers/doc/data_sheet/MC68HC908QY4.pdf
- (g) /files/microcontrollers/doc/data_sheet/MC68HC908QY4SM.pdf
- (h) /files/microcontrollers/doc/data_sheet/MC68HLC908QY4.pdf
- (i) /files/microcontrollers/doc/data_sheet/MC68HC908GP32.pdf

1.6.9.6 Nota sobre el “Kit” de Desarrollo JK3 Publicado por CEKIT

- (a) http://www.geocities.com/moto_hc08/pdf/NotaTecnica%20Kit%20JK3.pdf

1.6.9.7 Manual de Referencia del CPU

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

Pág. 124, Instrucción BNE.

Pág. 131, Instrucción BSET.

Pág. 140, Instrucción CPX.

Pág. 147, Instrucción INC.

Pág. 148, Instrucción JMP.

Pág. 150, Instrucción LDA.

Pág. 151, Instrucción LDHX.

Pág. 152, Instrucción LDX.

Pág. 169, Instrucción RSP.

1.6.9.8 Página “web” sobre esta Nota Técnica

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

1.6.10 Problemas Propuestos

Básese para confeccionar los programas en el listado 1.

1.6.10.1 Haga que el programa ejemplo salte a LOOP1 siete (7) veces.

1.6.10.2 Haga que su programa nunca salte a LOOP2.