NT1008

3.8 PANTALLAS DE CRISTAL LÍQUIDO DE CARACTERES – LCD

IMPLEMENTACIÓN DE ESCRITURAS EN PANTALLAS DE 16 X 4

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/

ÏNDICE

3.8.1 Introducción

Una LCD es la abreviatura de Pantalla de Cristal Líquido ("Liquid Cristal Display") y consta de una tecnología pasiva, de bajo consumo, y que generalmente no pasan del umbral de los 5 mA y se alimenta con un voltaje de 5 V. Las pantallas de cristal líquido son módulos inteligentes, que de por sí están construidos por un circuito de control y memorias para desplegar caracteres de tipo ASCII entre los que se dan números, letras y porqué no, caracteres definidos por el usuario. Las mismas, utilizan la luz ambiental para proyectar con entereza sus caracteres por medio de la polarización de cristales.

Aplicaciones comunes de una LCD son: pantallas de teléfonos públicos, faxes, fotocopiadoras, calculadoras y cualquier dispositivo en el cual se necesite "visualizar" un resultado o desplegar menús al usuario. Esta pantalla, en particular, consta de un arreglo de 16 X 4, lo que quiere decir que es de 16 columnas por 4 líneas de escritura. Pero también hay pantallas de cristal líquido está conformada por una ó dos líneas de 8, 16, 20, 24 ó 40 caracteres configurables algunas en matriz de 5x11 pixels[1] c/u o 5 X 8 pixels.

Como se notará más adelante, esta pantalla en particular se maneja de forma paralela, y son las más populares, pues, el precio es bajo. Por otro lado, pueden adquirirse pantallas de manejo serialmente. En este caso, nuestra pantalla tiene 16 líneas de entrada las cuales cada una tiene un significado específico.

[1] Se puede hacer la analogía de un píxel como un punto de la imagen.

3.8.2 Materiales

1. Microcontrolador: JL3 / JK3 / GP32
2. Potenciómetro: 10k @ 20k, Jameco Part No: 43001
3. Display LCD: LCD 16 X 4, Jameco Part No: 210761
4. Tarjeta de Desarrollo TD68HC908JL3 o similar y su microcontrolador
5. Plantilla de proyectos: “Breadboard” JE27, Jameco Part No: 20811
6. Fuente de Poder
7. Alambres AWG 20
8. Pelador de Alambres

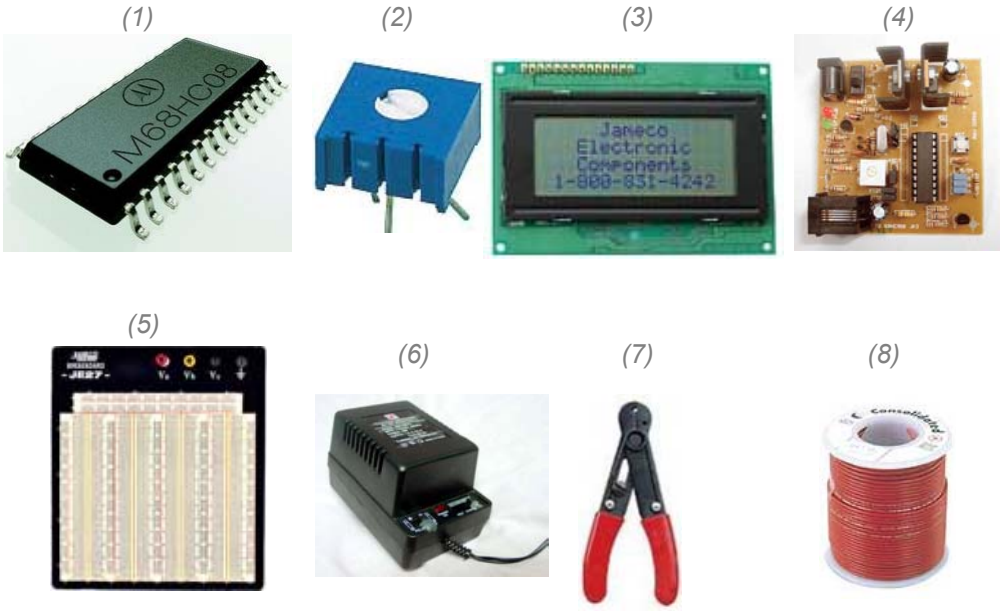


Figura 204. Listado de Materiales a Utilizar para la Experiencia de LCD

3.8.3 Descripción Física del "Hardware"

Para el siguiente párrafo, ayudarse por medio de la figuras y tablas de esta sección. La pantalla LCD consta de poco hardware para realizar interfase. Físicamente, la LCD contiene una parte de la pantalla la cual es visible, pero la misma puede desplazarse y ver la información del otro lado de la pantalla, consiguiendo así una parte visual real y otra virtual. La parte de la LCD encargada de esto se llama DDRAM ("Display Data RAM") y es la encargada de almacenar los caracteres desplegados con un máximo de 80 caracteres. Por otro lado, el usuario puede desplegar caracteres por medio de la CGRAM ("Character Generador RAM") en donde cada carácter es un arreglo de 5 bits por fila del caracter. Acceder la LCD se logra teniendo control del registro de datos (RS = 1) o escritura del carácter y el modo de comandos (RS = 0) para el cual se tiene acceso a los comandos que se tiene la LCD (ver sección 3.8.4).

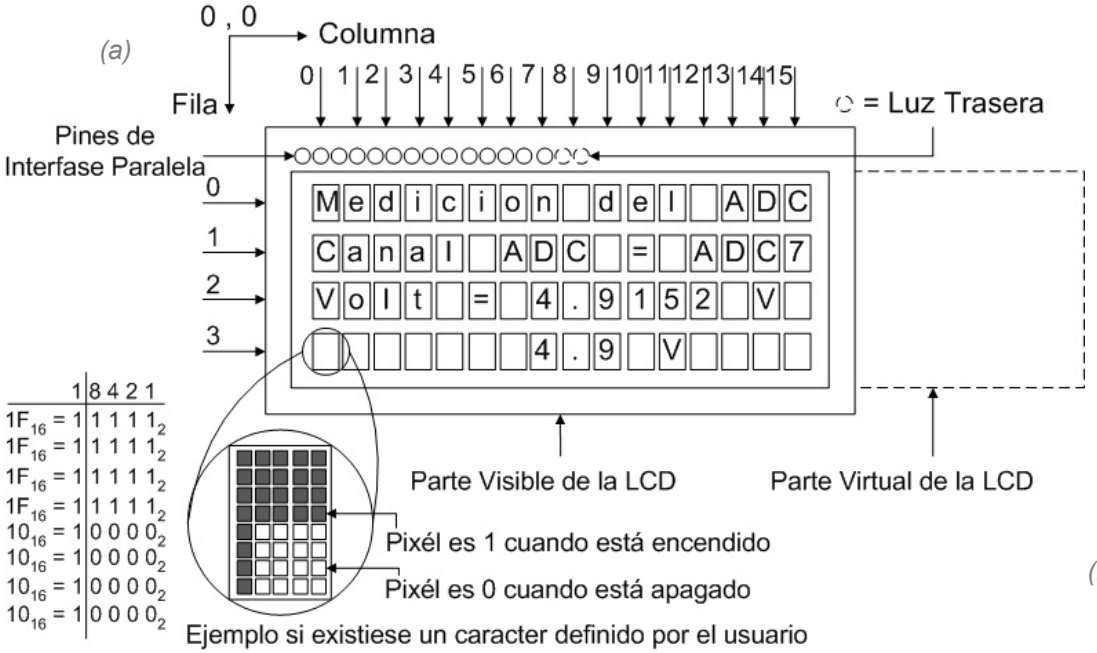


Tabla 76. Descripción de Pines de la LCD

Pin no.	Pin	Descripción
1	GND	Tierra física (0V)
2	VCC	Alimentación (+5V)
3	VO	Voltaje de Contraste
4	RS	Selecciona comando o escritura
5	R/W	Leer (1) o escribir (0)
6	E	Reconoce instrucción
7	D0	Bus de Datos
8	D1	
9	D2	
10	D3	
11	D4	
12	D5	
13	D6	
14	D7	
15	LED A	Luz trasera de LEDs (opcional)
16	LED K	

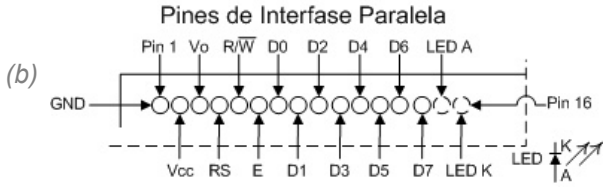


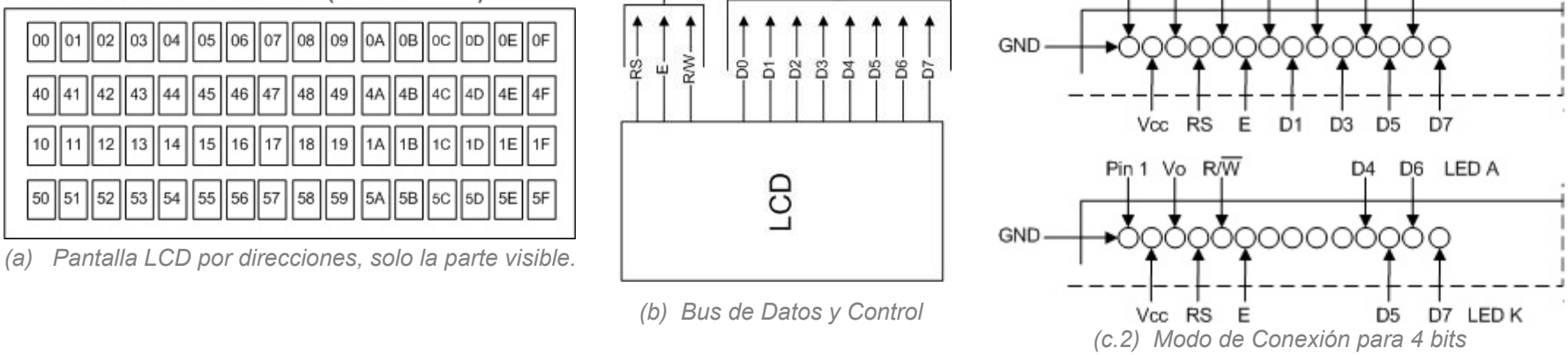
Figura 205. Descripción Física de una LCD. (a) Pantalla LCD 16 X 4. La pantalla está formada por un arreglo de filas y columnas de las cuales existen el área visible y el área virtual. Internamente, existen caracteres creados por defecto como letras y números, pero el usuario puede crear sus propios caracteres. (b) Pines de Interfase Paralela. La LCD está formada por el bus de datos D[0:7] que es el caracter a enviar y el bus de control, controlado por los pines RS, R/W y E, que seleccionan cuando y como se escribe o envía un comando a la pantalla.

3.8.4 Descripción Interna y Conexiones

3.8.4.1 La DDRAM, el Bus de Datos y de Control

Esta nota de aplicación trata de una LCD de 16 X 4, pero la ventaja de las LCDs es que el software es compatible con cualquier LCD de driver Hitachi HD44780, que es el que comúnmente se utiliza. La DDRAM de la LCD de este caso tiene sus inicios de líneas en 00_{16}, 40_{16}, 10_{16} y 50_{16} para las líneas 0, 1, 2, 3. Estas líneas, en el hardware interno empiezan en la posición 80_{16}, así, la línea 3, será escrita a la LCD como 90_{16}.

Para la LCD entender si se quiere escribir (R/W = 0) o leer (R/W = 1) se necesita cambiar de estado algunas líneas de control, como se muestra en la tabla 76. Luego se necesita un pulso de activación en E (E = 1) cuyo tiempo de retardo aproximado de encendido se da en la última columna de la tabla 76. Nuestra pantalla se ha trabajado en modo de 8 bits, pero también se puede trabajar en modo de 4 bits, para esto el envío de información es diferente; primeramente, solo se utilizan los pines D[7:4] y se envían los bits más significativos y luego los menos significativos.



(a) Pantalla LCD por direcciones, solo la parte visible.

(b) Bus de Datos y Control

(c.2) Modo de Conexión para 4 bits

Figura 206. Direcciones, Buses de Datos y Control de la LCD. (a) Direcciones de la DDRAM. Cada caracter en la DDRAM tiene una posición específica en donde cada línea internamente empieza por 80_{16} más su desfase por fila (80_{16} + 00, fila 0). (b) Bus de Datos y Control. El bus de datos es el encargado a depositar el caracter a desplegar o el comando a ejecutar, mientras el bus de control decide el cambio entre comando o escritura. (c) Modo de Trabajo. (c.1) Modo de 8 bits. Se utilizan todos los puertos de la pantalla para controlarla. (c.2) Modo de 4 bits. En este modo se envían primero los bits más significativos y luego los más significativos.

3.8.4.2 Juego de Comandos/Instrucciones de la LCD

Tabla 77. Juego de Instrucciones Necesarias para la LCD

Set de Instrucciones	RS	R/W	D7	D6	D5	D4	D3	D2	D1	D0	Descripción (1/0)	Delay (us)
Borrar Pantalla	0	0	0	0	0	0	0	0	0	1	Borra toda la pantalla.	1530
Regresar al origen	0	0	0	0	0	0	0	0	1	X	Posiciona cursor en el extremo superior izquierdo.	1530
Modo de Ingreso Fijado	0	0	0	0	0	0	0	1	I/D	SH	I/D Incrementa o decrementa cursor; SH Mueve al otro extremo la pantalla.	40
Control de Despliegue de ENCENDIDO/APAGADO	0	0	0	0	0	0	1	D	C	B	D Display, C Cursor, B Parpadeo ON/OFF.	
Cursor o Cambio de Despliegue	0	0	0	0	0	1	S/C	R/L	X	X	S/C Mueve pantalla o cursor, R/L izquierda o derecha	
Función Fijada	0	0	0	0	1	DL	N	F	X	X	DL Longitud de la Data (8/4bits), N número de líneas (1/2), F Fuente [5X(11/8)].	
Fijar la Dirección en el CGRAM	0	0	0	1	AC 5	AC 4	AC 3	AC 2	AC 1	AC 0	Dirección del CGRAM en el AC (crear carácter en…).	
Fijar la Dirección en el DDRAM	0	0	1	AC 6	AC 5	AC 4	AC 3	AC 2	AC 1	AC 0	Dirección a posicionar en la pantalla.	
Leer Dirección y Bandera de Ocupado	0	1	BF	AC 6	AC 5	AC 4	AC 3	AC 2	AC 1	AC 0	Si hay una operación interna o no, puede ser conocida leyendo el bit BF, el contenido del AC, también puede ser leído.	0
Escribir Data en RAM	1	0	D7	D6	D5	D4	D3	D2	D1	D0	Escribe un carácter de la DDRAM ó CGRAM	40
Leer Data de la RAM	1	1	D7	D6	D5	D4	D3	D2	D1	D0	Leer un carácter de la (DDRAM ó CGRAM)	

D[0:7]: Bits de Data, X: No importa, AC: Contador de Direcciones, BF: Bandera de Ocupado, DDRAM: "Display Data RAM", CGRAM: "Character Generator RAM"

A manera de práctica se implementó el modo de 8 bits el cual utiliza más pines de control del microcontrolador. Por conveniencia, el pin de lectura/escritura R/W se ha anclado a tierra o masa (ver figura 207) por lo que entre instrucción se debe retardar el tiempo necesario dado por la tabla para que el manejador ("driver") pueda distinguir entre una instrucción y otra siguiente.

3.8.5 Esquemático de Aplicación

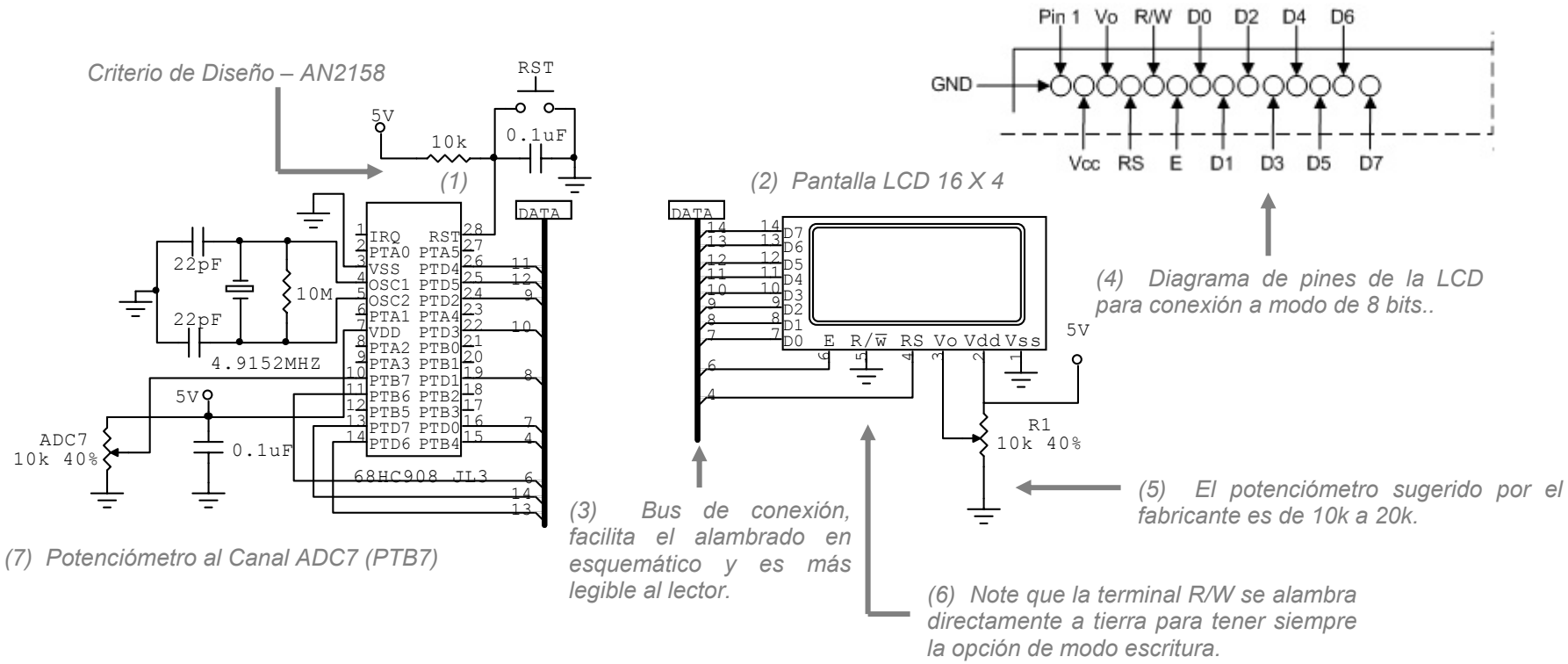


Figura 207. Esquema de Aplicación para LCD. Si ud. posee la tarjeta de desarrollo solo conecte la pantalla a los puertos del microcontrolador y levante el "jumper" del extremo inferior izquierdo conectado al puerto PTD7. Esquema de conexión de una pantalla LCD en modo de 8 bits para despliegue de información. Para conexión a modo de 4 bits observe el esquema de la figura 206(c.2).

3.8.6 Diagrama de Flujo

El siguiente programa utiliza el ADC Canal 7 par convertir el valor digital a un valor numérico y desplegarlo en la LCD como se muestra en la figura 205(a). Para esta nota técnica ud. necesita de las notas básicas NT0009, NT0010, NT0011, NT0012 y Apéndices B, D y K.

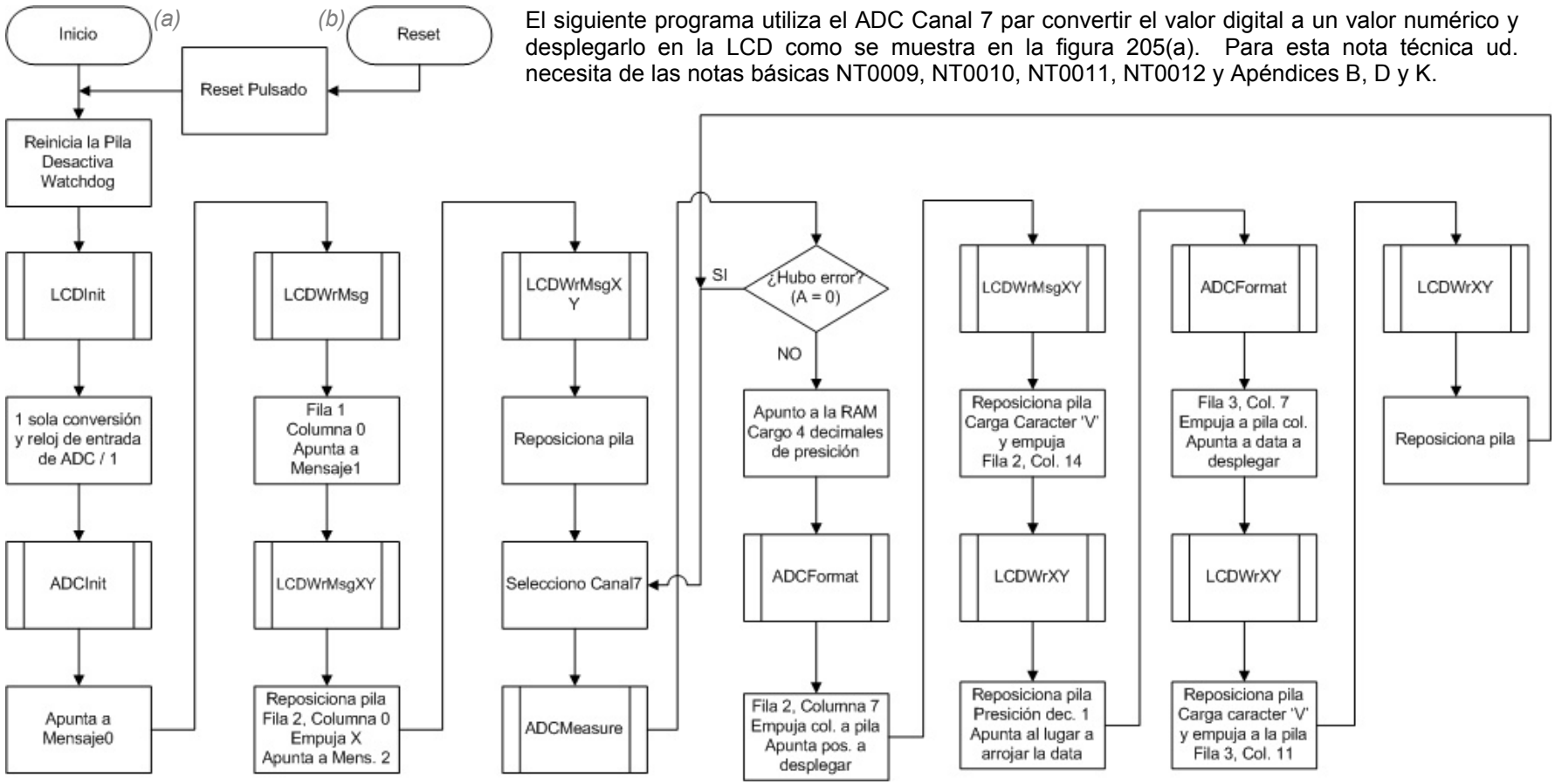


Figura 208. NT1008 – LCD. (a) Programa Principal. El programa despliega mensajes de texto sobre la LCD y finalmente despliega un valor numérico entre los 0 y 5V que representa el valor del convertidor analógico digital tomado directamente del canal ADC7.

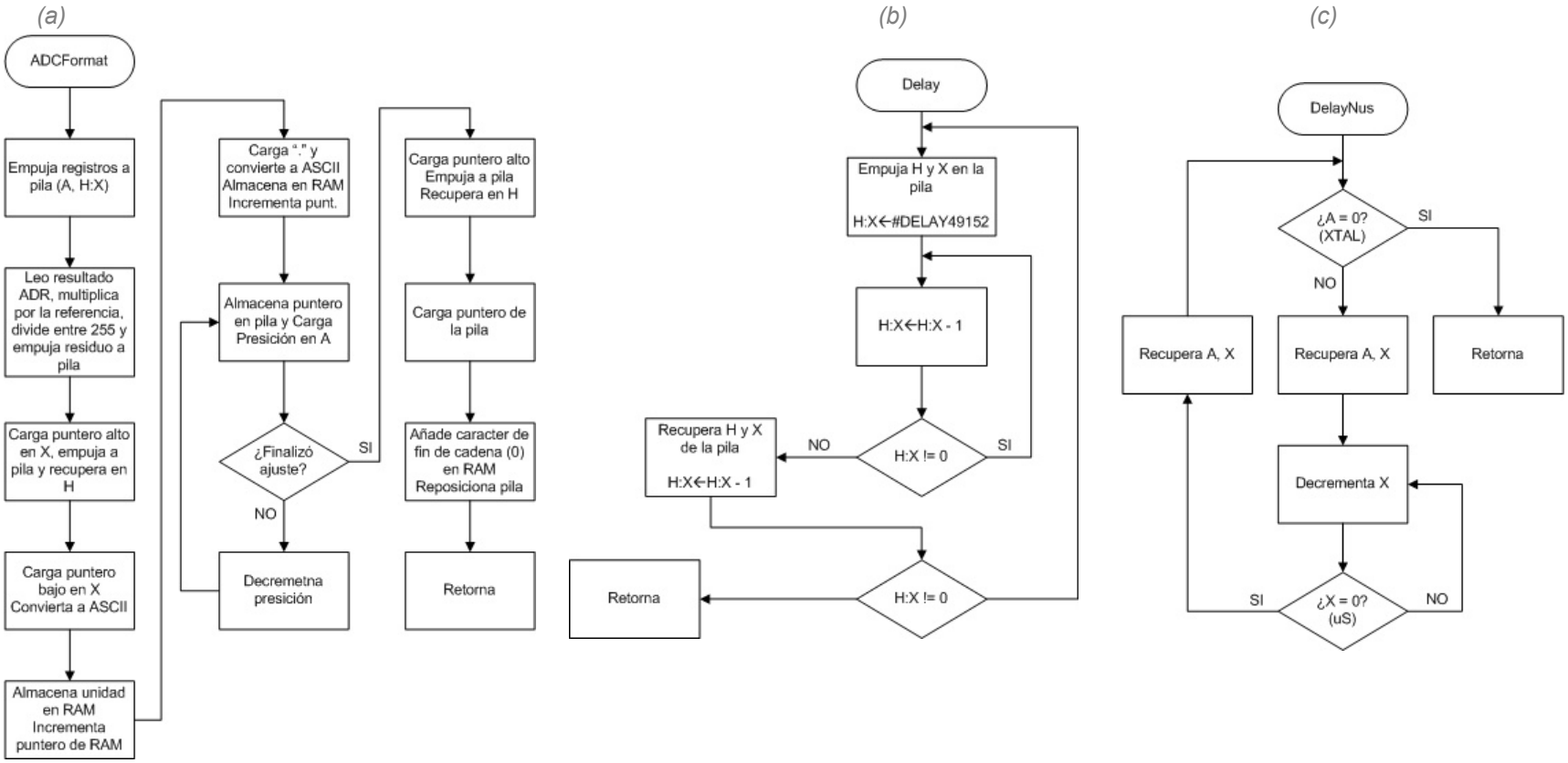


Figura 209. NT1008 – LCD – Subrutinas. (a) ADCFormat. Rutina programable que almacena en algún lugar de la RAM el valor escalado en números ASCII del equivalente voltaje analógico arrojado por el potenciómetro. La misma rutina puede ajustarse para mayor precisión en los números decimales. (b) Delay. Rutina programable de base de tiempo de 1 ms. (c) DelayNus. Rutina progamable que retarda "n" μs.

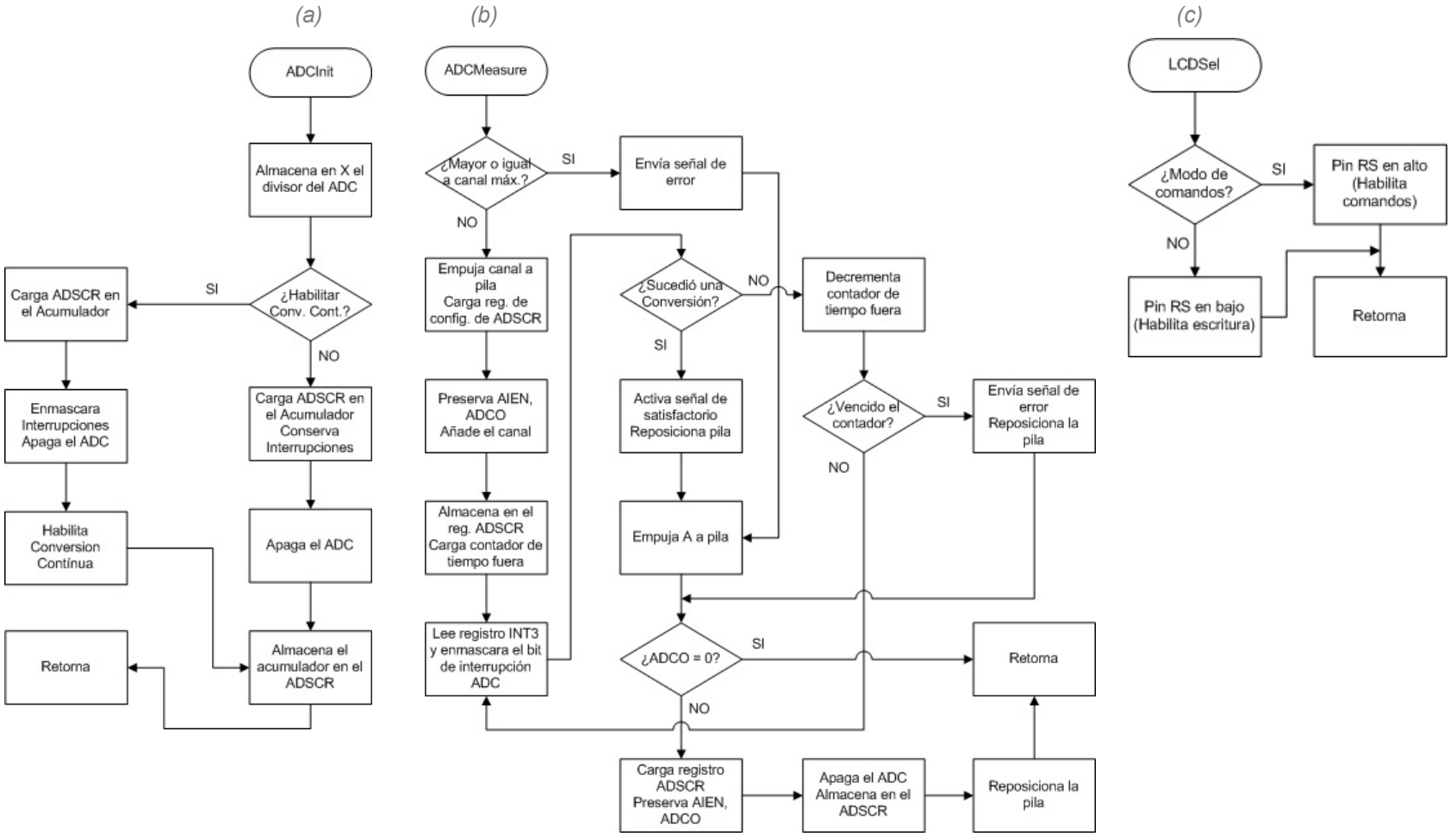


Figura 210. NT1008 – LCD – Subrutinas - Continuación. (a) ADCInit. Subrutina programable encargada de inicializar el ADC, en este caso, inicia en una sola conversión y apaga el convertidor. (b) ADCMeasure. Subrutina decide cuando se debe leer el convertidor. Envía una bandera de estado que avisa si se debe o no leer el convertidor. Para mayor información de los códigos reusables, ver apéndices. Nota: solo se lista las funciones del ADC utilizadas. (c) LCDSel. Rutina que cambia el modo de comandos a escritura o viceversa.

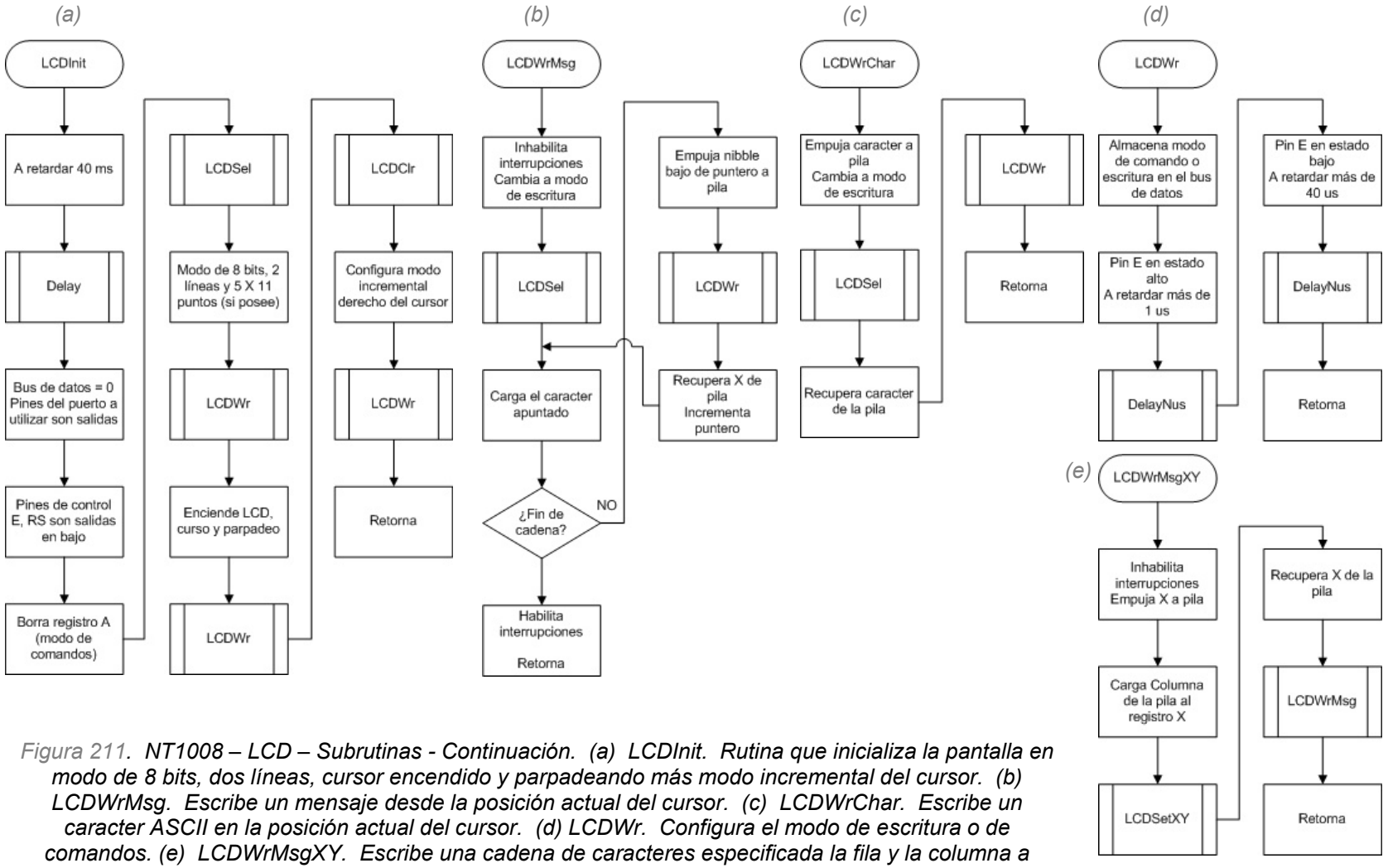


Figura 211. NT1008 – LCD – Subrutinas - Continuación. (a) LCDInit. Rutina que inicializa la pantalla en modo de 8 bits, dos líneas, cursor encendido y parpadeando más modo incremental del cursor. (b) LCDWrMsg. Escribe un mensaje desde la posición actual del cursor. (c) LCDWrChar. Escribe un caracter ASCII en la posición actual del cursor. (d) LCDWr. Configura el modo de escritura o de comandos. (e) LCDWrMsgXY. Escribe una cadena de caracteres especificada la fila y la columna a escribirse.

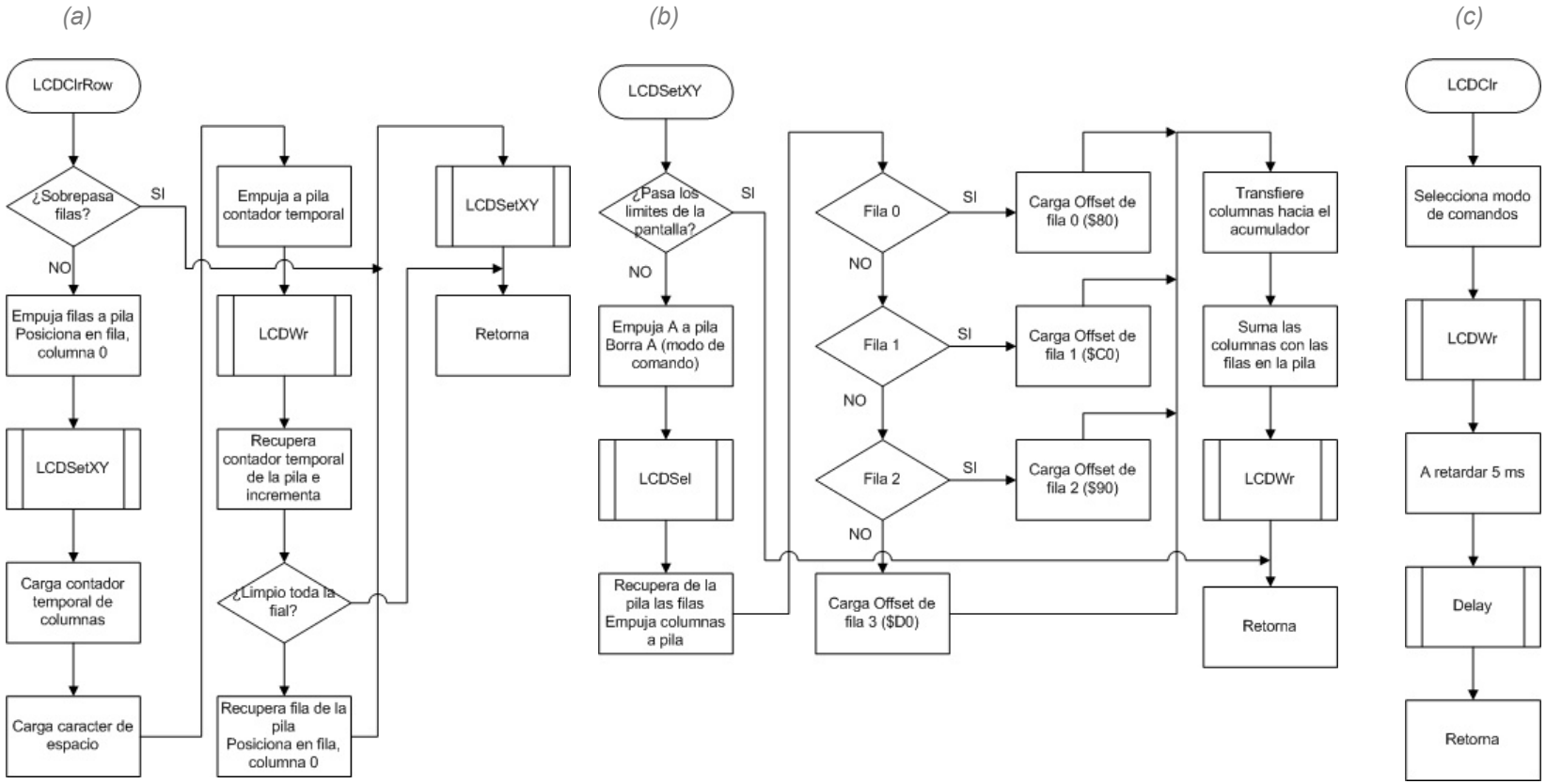


Figura 212. NT1008 – LCD – Subrutinas - Continuación. (a) LCDClrRow. Rutina que borra una fila escribiendo el caracter ASCII ' ' y reposiciona al inicio de la fila . (b) LCDSetXY. Dada las coordenadas, posiciona en la fila y columna correspondiente. (d) LCDClr. Borra completamente la pantalla. Para mayor información sobre las rutinas reusables, ver apéndices.

3.8.7 Código

Se recomienda que si no se tiene experiencia previa, referirse a las notas, NT0010, NT0011, NT0012 y Apéndices B, D y K.

```
;===================================================================
; ARCHIVO      : NT1008 - LCD - 05 11 04.asm
; PROPÓSITO    : Control de una pantalla LCD para envío de caracteres.
;                Despliega la medición del voltaje del canal ADC7 en la LCD.
;
;          00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15
;        +************************************************************
;      00* M  e  d  i  c  i  o  n     d  e  l     A  D  C
;      01* C  a  n  a  l     A  D  C     =     A  D  C  7
;      02* V  o  l  t     =     4  .  9  1  5  2  V
;      03*                      4  .  9  V
;        +************************************************************
;
; NOTA         : Por el momento solo se controlan pantallas de caracteres
;
; REFERENCIA:  NT1008 - LCD - 09 11 04.doc
;
; LENGUAJE     : IN-LINE ASSEMBLER
; -------------------------------------------------------------------
; HISTORIAL
;  DD MM AA
;  13  06  03   Creado.
;  09  11  04   Modificado.
;===================================================================

;===========================================
;              Cabecera de Macros, Const. y Memoria
;===========================================
$include '\MAP\includes.equ'                ; Definiciones de usuario y mapa de memoria
```

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

;=============================================
; OBJETIVO    : Escribe en la pantalla el
;               resultado de la conversión
;               de la medición a rangos de
;               voltaje de 0 V a +5 V
;=============================================
START
        rsp                         ; Inic.Stack = $00ff
        bset BIT0,CONFIG1           ; Desactiva watchdog
        jsr LCDInit                 ; Inicializa pantalla
        clra                        ; 1 sola conversión
        ldx #ADCDIV1                ; Reloj ADC / 1
        jsr ADCInit                 ; Inicializa ADC
        ldhx #LCDMESSAGE00          ; Carga mensaje 0
        jsr LCDWrMsg                ; Escribe
        lda #1T                     ; Fila 1
        clrx                        ; Columna 0
        pshx                        ; Empuja columna
        ldhx #LCDMESSAGE01          ; Carga mensaje 1
        jsr LCDWrMsgXY              ; Escribe en la posición X-Y
        pulx                        ; Reposiciona pila
        lda #2T                     ; Fila 2
        clrx                        ; Columna 0
        pshx                        ; Empuja a pila
        ldhx #LCDMESSAGE02          ; Carga mensaje 2
        jsr LCDWrMsgXY              ; Escribe en la pantalla
        pulx                        ; Reposiciona pila
LCD1008.K
        lda #ADC7                   ; Canal ADC7 = PTB7
        jsr ADCMeasure              ; Mide
        beq LCD1008.K               ; Error?, medir nuevamente
        ldhx #$0080                 ; Posición a almacenar la información
        lda #3T                     ; Número de decimales (4)
        jsr ADCFormat               ; Cambia a ASCII
        lda #2T                     ; Fila 2
        ldx #7T                     ; Columna 7
        pshx                        ; Empuja Columna a pila
        ldhx #$0080                 ; Carga puntero
        jsr LCDWrMsgXY              ; Escribe mensaje
        pulx                        ; Reposiciona pila
        lda #'V'                    ; Carga caracter
        psha                        ; Empuja Caracter
        lda #2T                     ; Fila 2
        ldx #14T                    ; Columna 14
        jsr LCDWrXY                 ; Escribe caracter
        pulx                        ; Reposiciona pila
```

```
        clra                    ; 1 número decimal
        ldhx #$0080             ; Ubica en RAM
        jsr ADCFormat           ; Cambia a ASCII
        lda #3T                 ; Fila 3
        ldx #7T                 ; Columna 7
        pshx                    ; Empuja Columna
        ldhx #$0080             ; Carga posición a desplegar
        jsr LCDWrMsgXY          ; Despliega
        pulx                    ; Reposiciona pila
        lda #'V'                ; Carga V
        psha                    ; Empuja a pila
        lda #3T                 ; Fila 3
        ldx #11T                ; Columna 11
        jsr LCDWrXY             ; Escribe caracter
        pula                    ; posiciona puntero
        bra LCD1008.K           ; Salta a desplegar nuevamente

;===============================================
;               Declaración y definición de funciones
;===============================================
$include '\FUNCTIONS\INCLUDES.inc' ; Incluye Funciones


;===============================================
;             Declaración y Definición de interrupciones
;===============================================
$include '\INTERRUPTS\interrupt.inc'    ; Incluye interrupciones del microcontrolador
```

Listado 55. NT1008 – LCD. Utiliza el convertidor analógico digital en modo de una sola conversión para ver el resultado desplegado en la LCD. Nótese al ejecutar el código, que la precisión de 1 decimal no varía en comparación a la precisión de 4 decimales, esto se debe al error de cuantización del ADC.

```
;=====================================================================
; ARCHIVO      :  USER.inc
; PROPÓSITO    :  Funciones de usuario
; LENGUAJE     :  IN-LINE ASSEMBLER
; ---------------------------------------------------------------------
; HISTORIAL
;   DD MM AA
;   06   09   04   Creado.
;   05   11   04   Modificado.
;=====================================================================

;=============================================
;                    Macros & Constantes
;=============================================
VREFH           equ 5T                  ; Voltaje máximo a medirse (por el momento)
                                        ; la rutina DEBE ser menor a VREF < 10 V

;=============================================
;                    Funciones de Usuario
;=============================================
```

```
;===========================================
; ADCFORMAT :  Cambia el registro ADR a un
;              valor a desplegar en el
;              LCD.
; OBJETIVO  :  Despliega la data en RAM
; ENTRADA   :  A     es la precisión de dí
;              gitos decimales
;              H:X   es el lugar a ubicar
;              la data a desplegar
; SALIDA    :  Ninguna
; REGISTROS
; AFECTADOS :  RAM, H:X, A, SP
;===========================================
ADCFormat
        pshh                    ; Empuja a pila
        pshx                    ; Empuja a pila
        psha                    ; Guarda precisión
        lda ADR                 ; Carga resultado
        ldx #VREFH              ; Carga referencia
        mul                     ; Escala ADC
        pshx                    ; Empuja byte alto a pila
        pulh                    ; Recupera para dividir

        ldx #255T               ; Carga divisor
        div                     ; Divide
        pshh                    ; Empuja a pila
        ldx 4,sp                ; Carga puntero alto
        pshx                    ; Empuja a pila
        pulh                    ; Recupera en H
        ldx 3,sp                ; Carga puntero bajo
        add #'0'                ; Cambia a ASCII
        sta ,x                  ; Almacena en RAM
        aix #1                  ; Siguiente posición en RAM
        lda #'.'                ; Carga punto decimal
        sta ,x                  ; Almacena en RAM
        aix #1                  ; Siguiente posición
        stx 3,sp                ; Almacena en pila
ADCFMT1008.A
        pshh                    ; Empuja a pila
        pulx                    ; Recupera byte bajo
        stx 4,sp                ; Almacena en pila
        pula                    ; Recupera residuo
        ldx #10T                ; Carga X con decimal
        mul                     ; Corre el décimo a entero
        pshx                    ; Empuja a pila
        pulh                    ; Recupera en H para dividir
        ldx #255T               ; A Dividir entre la resolución máxima
        div                     ; Divide
        pshh                    ; Empuja residuo a pila
        ldx 4,sp                ; Carga X con byte alto de RAM
        pshx                    ; Empuja a pila
        pulh                    ; Posicion alta a almacenar
        ldx 3,sp                ; Carga Posición baja de RAM
```

```
        add #'0'                ; Cambia a ASCII
        sta ,x                  ; Almacena en pila
        aix #1                  ; Incrementa en 1 la posición
        stx 3,sp                ; Almacena puntero en pila
        lda 2,sp                ; Carga precisión
        beq ADCOUT1008.D        ; Si es 0 salir
        dec 2,sp                ; Decrementa precisión
        bra ADCFMT1008.A        ; Sigue hasta terminar la precisión
ADCOUT1008.D
        ldx 4,sp                ; Carga puntero alto
        pshx                    ; Empuja a pila
        pulh                    ; Recupera en posición alta
        ldx 3,sp                ; Carga puntero a pila
        clra                    ; Caracter de fin de cadena
        sta ,x                  ; Almacena en RAM
        ais #4                  ; Posiciona valor de retorno
        rts                     ; Retorna

;=============================================
;                   Tablas del Usuario
;=============================================
$include '\USER\TABLES.inc'         ; Tablas del Usuario
```

Listado 56. NT1008 – LCD – USER.inc. Esta parte de programa es una función de usuario que instala en la RAM el resultado del convertidor ADC en ASCII.

```
;=====================================================================
; ARCHIVO      :  TABLES.inc
; PROPÓSITO    :  Tablas de búsqueda predefinidas por el usuario
; LENGUAJE     :  IN-LINE ASSEMBLER
; ---------------------------------------------------------------------
; HISTORIAL
;   DD MM AA
;   06   09   04   Creado.
;   11   11   04   Modificado.
;=====================================================================

;=============================================
;                  Tablas de Búsquedas
;=============================================
;=============================================
;                  TABLA DE MENSAJES
;=============================================
LCDMESSAGE00        db        'Medicion del ADC',0
LCDMESSAGE01        db        'Canal ADC = ADC7',0
LCDMESSAGE02        db        'Volt =',0
```

Listado 57. NT1008 – LCD – TABLES.inc. Tabla de mensajes a enviar a la pantalla de cristal líquido

```
;=====================================================================
; ARCHIVO      : ADC.inc
; PROPÓSITO    : Inclusión de la función de inicialización y medición de ADC
; LENGUAJE     : IN-LINE ASSEMBLER
; ---------------------------------------------------------------------
; HISTORIAL
;   DD MM AA
;   30  08  04   Creado.
;   30  08  04   Modificado.
;=====================================================================

;=============================================
;                    Constantes & Macros
;=============================================
ADC_MAX_CH          equ 12T         ; Máximo número de canales del MUX
ADC_TO              equ 15T         ; Tiempo fuera de Conversión

;=============================================
; ADCINIT       : Inicializa el convertidor
; OBJETIVO      : Inicializa el ADC con su di-
;                 visor y tipo de conversión
; ENTRADA       : A  es el tipo de conversión
;                 X  contiene el valor del
;                 divisor del reloj de en-
;                 trada del ADC
; SALIDA        : Ninguna
; REGISTROS
; AFECTADOS     : ADSCR, ADICLK, ACCA, X
;=============================================
ADCInit
        stx ADICLK                  ; Almacena divisor del ADC
        cmp #1                      ; ¿A habilitar conv. continua?
        beq ADC0011.E               ; Si no es igual, apagar ADC
        lda ADSCR                   ; Carga el ADSCR
        and #AIEN                   ; Habilita interrupciones
        ora #ADCOFF                 ; Apaga el ADC
        bra ADC0011.F               ; Salir
ADC0011.E
        lda ADSCR                   ; A = ADSCR
        and #{AIEN|ADCOFF}          ; Preserva bits de int. y canal
        ora #ADCO                   ; añade bit ADCO = 1
ADC0011.F
        sta ADSCR                   ; Almacena en el registro del ADC
        rts                         ; Retorna
```

```
;===========================================
; ADCMEASURE
;                : Decide si el canal es hábil
; OBJETIVO       : Canal disponible o no
; ENTRADA        : A  contiene el canal a medir
; SALIDA         : A  decide si el canal está
;                  disponible o no
; REGISTROS
; AFECTADOS      : ADSCR, ADICLK, ACCA, X
; NOTA           : SALIDA = 0, Señal de error
;                  SALIDA = 1, Lectura Dispo-
;                  nible
;===========================================
ADCMeasure
        cmp #ADC_MAX_CH          ; ¿Dentro del rango de multiplexión?
        bge ADCOUT0011.D         ; NO, salir
        psha                     ; Envía a la pila
        lda ADSCR                ; Carga Registro ADC
        and #{AIEN|ADCO}         ; Enmascara los bits altos
        ora 1,SP                 ; Cambia de canal
        sta ADSCR                ; Almacena en el convertidor
        ldx #ADC_TO              ; Contador de Tiempo Fuera
ADCLOOP0011.B
        lda INT3                 ; leer INT3

        bit #BIT0M               ; Enmascara BIT0
        bne AREAD0011.A          ; Hay interrupción, leer
        brset COCO,ADSCR,AREAD0011.A
                                 ; Si termina la conversión, leer
        decx                     ; ADC_TO = ADC_TO - 1 (TO =
                                 ; TimeOut)
        cpx #0                   ; ¿contador = 0?
        bne ADCLOOP0011.B        ; Salta y pregunta nuevamente
        ais #1                   ; Posiciona la dirección de retorno
ADCOUT0011.D
        clra                     ; Señal de error
        bra ADCOUT0011.C         ; salir del ADC
AREAD0011.A
        lda #1                   ; Resultado Disponible
        ais #1                   ; Posiciona la dirección de retorno
ADCOUT0011.C
        psha                     ; empuja a la pila
        brset BIT5,ADSCR,ADC0011.I ; Si ADCO = 0, salir
        lda ADSCR                ; Carga ADSCR
        and #{AIEN|ADCO}         ; Preserva bits
        ora #ADCOFF              ; apaga el ADC
        sta ADSCR                ; Almacena en el registro
ADC0011.I
        pula                     ; recupera de la pila
        rts                      ; Retorna
```

Listado 58. NT1008 – LCD – ADC.inc. Nota: Solo se listan las funciones utilizadas para el ADC.

```
;*******************************************************************
; ARCHIVO    :  DELAY.inc
; PROPÓSITO  :  Función de retardo de tiempo
; ARGUMENTOS :  H:X = tiempo en milisegundos
; RETORNO    :  Ninguno
; LENGUAJE   :  IN-LINE ASSEMBLER
; -------------------------------------------------------------------
; HISTORIAL
;   DD MM AA
;   22 08 04 Creado.
;   22 08 04 Modificado.
;*******************************************************************

;===============================================
;                  CONSTANTES & MACROS
;===============================================
DELAY4          equ $0079       ; Constante de Retardo a 4.0000 MHz
DELAY49152      equ $0096       ; Constante de Retardo a 4.9152 MHz
DELAY98304      equ $0130       ; Constante de Retardo a 9.8304 MHz
DELAY128        equ $018C       ; Constante de Retardo a 12.800 MHz
DELAY32         equ $03E4       ; Constante de Retardo a 32.000 MHz

;===============================================
; 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
;===============================================
Delay
        cphx #0                 ; [3] Compara con 0
        beq DELAY0009.A         ; [3] Salir si es 0
        pshx                    ; [2] Salva X en la pila
        pshh                    ; [2] Salva H en la pila
        ldhx #DELAY49152        ; [3] Carga constante de bucle fino
Delay0009.B
        aix #-1                 ; [2] Decrementa H:X en 1
        cphx #0                 ; [3] LLegó a cero (0)
        bne Delay0009.B         ; [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
DELAY0009.A
        rts                     ; [4] retorna
```

```
;===========================================
; DELAYNUS    :  Genera un retardo de tiempo
; OBJETIVO    :  Retardo de tiempo, base 1us
; ENTRADA     :  X   = Valor entero del cris-
;                tal externo
;             A   = Tiempo a retardar en us
; SALIDA      :  Ninguna
; REGISTROS
; AFECTADOS   :  A, X
;===========================================
DelayNus
        tsta                    ; [1] Verifica estado de A
        beq DELAY0009.C         ; [3] Salir si es cero
        deca                    ; [1] Decrementa A
        psha                    ; [2] Empuja A a pila
        pshx                    ; [2] Empuja variable X
        dbnzx *                 ; [3] Decrementa hasta 0
        pulx                    ; [2] Recupera variable X
        pula                    ; [2] Recupera A
        bra DelayNus            ; [3] Salta a comprobar estado de A
DELAY0009.C
        Rts                     ; [4] Retorna
```

Listado 59. NT1008 – LCD – DELAY.inc. Funciones de retardo de "software".

```
;=====================================================================
; ARCHIVO      : LCD.inc
; PROPÓSITO    : Librería para pantalla LCD
; LENGUAJE     : IN-LINE ASSEMBLER
; NOTAS        : LA LIBRERÍA LCD.INC UTILIZA LAS FUNCIONES DE LA ;
                 LIBRERÍA DE FUNCIONES DE RETARDO DELAY.INC.
; ---------------------------------------------------------------------
; HISTORIAL
;  DD MM AA
;  05  11  04   Creado.
;  11  11  04   Modificado.
;=====================================================================
; INICIALIZACIÓN
;                : Esta rutina puede utilizarse con cualquier LCD con
;                  driver Hitachi HD44780.
;                Para utilizar esta rutina usted deberá inicializar
;                el módulo de la siguiente manera.  Configurar:
;                1 - Puerto de datos:  LCD_DATA_PORT
;                2 - Reg. de direccionamiento para datos:  LCD_DATA_DDR
;                3 - Pines de salida de datos a utilizar:  LCD_DATA_OUTPUT
;                4 - Puerto de la señal RS:  LCD_RS_PORT
;                5 - Reg. de direccionamiento de la señal RS:  LCD_RS_DDR
;                6 - Pin del puerto para señal RS:  LCD_RS_PIN
;                7 - Puero de la señal E:  LCD_E_PORT
;                8 - Reg. de direccionamiento de E:  LCD_E_DDR
;                9 - Pin del puerto para E:  LCD_E_PIN
;
;  0,0
;   +-> (Columna)
;   |         Pantalla LCD de 16 X 4 Caracteres
;   U              (Caracteres Visibles)
; (Fila)
;
;           00  01  02  03  04  05  06  07  08  09  10  11  12  13  14  15
;          +*****************************************************************
;        00* A   B   C   D   E   F   G   H   I   J   K   L   M   N   O   P
;        01* Q   R   S   T   U   V   W   X   Y   Z   0   1   2   3   4   5
;        02* 6   7   8   9   a   b   c   d   e   f   g   h   i   j   k   l
;        03* m   n   o   p   q   r   s   t   u   v   w   x   y   z   #   !
;          +*****************************************************************
;
;
;
;=====================================================================
```

```
;===========================================
;                  Constantes & Macros
;===========================================
;===========================================
;              DEFINA EL PUERTO DE DATOS
;===========================================
LCD_DATA_PORT     equ PORTD     ; Puerto de Salida de la Información
LCD_DATA_DDR      equ DDRD      ; Registro de dir. del Puerto de Salida
                                ; de la Información
LCD_DATA_OUTPUT   equ $FF       ; Puertos a Utilizar en el LCD ($FF = Todos
                                ; son utilizados, modo 8bits)

;===========================================
;  DEFINA PUERTO DE REGISTRO/INSTRUCCIÓN LCD
;===========================================
LCD_RS_PORT       equ PORTB     ; Puerto de Registro/Instrucción del
                                ; LCD
LCD_RS_DDR        equ DDRB      ; Registro de dir. del Puerto del RS del LCD
LCD_RS_PIN        equ BIT4      ; Pin del Puerto

;===========================================
;  DEFINA PUERTO HABILITADOR DE INSTRUCCIÓN
;===========================================
LCD_E_PORT        equ PORTB     ; Puerto de Habilitador de Instrucción
                                ; del LCD
LCD_E_DDR         equ DDRB      ; Registro de dir. del Puerto de RS
LCD_E_PIN         equ BIT6      ; Pin del Puerto

;===========================================
; DEFINA MÁXIMO DE FILAS Y COLUMNAS DE SU LCD
;===========================================
LCD_MAX_ROW       equ 4T        ; Máximo de filas
LCD_MAX_COL       equ 16T       ; Máximo de columnas
```

```
;=============================================
;                INSTRUCCIONES DEL LCD
;=============================================
LCD_CLR         equ %00000001     ; Borra pantalla y ubica en posición
                                  ; inicial izq.
LCD_HOME        equ %00000010     ; Regresa a posición inicial
LCD_INC         equ %00000010     ; Incremento de Cursor
LCD_SHIFT       equ %00000001     ; Cambio extremo activado
LCD_ON          equ %00000100     ; Enciende el LCD
LCD_CSR         equ %00000010     ; Enciende el Cursor
LCD_BLINK       equ %00000001     ; Parpadea LCD
LCD_RT          equ %00000100     ; Display Shift Derecho
LCD_SET         equ %00001000     ; Cursor Encendido
LCD_8BIT        equ %00010000     ; LCD de 8 bit
LCD_2LINE       equ %00001000     ; LCD 2 líneas
LCD_5X11        equ %00000100     ; LCD 5X11 Puntos

LCD_CSR_RT      equ %00010100     ; Mueve el cursor a la derecha
LCD_CSR_LT      equ %00010000     ; Mueve el cursor a la izquierda

LCD_MOVE_RT     equ %00011100     ; Mueve la pantalla a la derecha
LCD_MOVE_LT     equ %00011000     ; Mueve la pantalla a la izquierda

;=============================================
;   CONSTANTES DE RETARDO E INHERENCIAS
;=============================================
LCD_RS_HIGH     equ 1T        ; Comando de escritura
LCD_SPACE_CHAR  equ $20       ; Caracter de espacio

LCD_CLR_DLY     equ 5T        ; Retardo de estabilidad de comando CLR

LCD_ROW1        equ 1T        ; Comprobante de fila 1
LCD_ROW2        equ 2T        ; Comprobante de fila 2
LCD_ROW0_OFFSET equ $80       ; Desfase hacia fila 0 ($80 + $00)
LCD_ROW1_OFFSET equ $C0       ; Desfase hacia fila 1 ($80 + $40)
LCD_ROW2_OFFSET equ $90       ; Desfase hacia fila 2 ($80 + $10)
LCD_ROW3_OFFSET equ $D0       ; Desfase hacia fila 3 ($80 + $50)
```

```
;=========================================
; LCDINIT      : Inicializa el módulo LCD
; OBJETIVO     : Configura el LCD en modo de
;                8bits, 2 líneas, matriz de
;                5 X 11 puntos si existe en el
;                LCD, Incremento
; ENTRADA      : Ninguna
; SALIDA       : Ninguna
; REGISTROS
; AFECTADOS    : PORTX, DDRX, A
;=========================================
LCDInit
        clrh                            ; Borra H
        ldx #40T                        ; A retardar 40 ms
        jsr Delay                       ; Retarda
        clr LCD_DATA_PORT               ; DATA     = LOW
        mov #LCD_DATA_OUTPUT,LCD_DATA_DDR
                                        ; DDR_DATA  = OUTPUT
        bset LCD_E_PIN,LCD_E_DDR ; DDR_E    = OUTPUT
        bset LCD_RS_PIN,LCD_RS_DDR
                                        ; DDR_RS    = OUTPUT
        bclr LCD_RS_PIN,LCD_RS_PORT
                                        ; RS       = LOW
        bclr LCD_E_PIN,LCD_E_PORT
                                        ; E        = LOW
        clra                            ; A        = 0
        jsr LCDSel                      ; RS       = LOW
        lda #{$20|LCD_8BIT|LCD_2LINE|LCD_5X11}
                                        ; LCD 8bits, 2 líneas y 5 X 11
        jsr LCDWr                       ; Ejecuta comando
        lda #{$08|LCD_ON|LCD_CSR|LCD_BLINK}
                                        ; Enciende LCD, Cursor, Parpadeo.
        jsr LCDWr                       ; Ejecuta comando
        jsr LCDClr                      ; Borra pantalla
        lda #{$04|LCD_INC}              ; Modo incremental
        jsr LCDWr                       ; Ejecuta comando
        rts                             ; Retorna
```

```
;===============================================
; LCDCLR       : Borra la pantalla
; OBJETIVO     : Borra completamente el LCD
;                y ubica cursor en la parte
;                superior izquierda
; ENTRADA      : Ninguna
; SALIDA       : Ninguna
; REGISTROS
; AFECTADOS    : A, H:X
;===============================================
LCDClr
        lda #0T                 ; RS = LOW
        jsr LCDSel              ; Cambia estado de RS a bajo
        lda #LCD_CLR            ; Comando borrar
        jsr LCDWr               ; Ejecuta comando
        ldhx #LCD_CLR_DLY       ; A retardar 5 ms
        jsr Delay               ; Ejecuta retardo
        rts                     ; Retorna

;===============================================
; LCDCLRROW:  Borra una línea específica
; OBJETIVO     : Llena una línea de caracteres
;                en blanco (ASCII $20) y ubica
;                al principio
; ENTRADA      : A    contiene la línea a
;                eliminar
; SALIDA       : Ninguna
; REGISTROS
; AFECTADOS    : A, X, SP
;===============================================
LCDClrRow
        cmp #LCD_MAX_ROW        ; Compara con el máximo de filas
        bge LCDOUT1008.K        ; ¿Mayor, salir?
        psha                    ; NO, Empuja fila a la pila
        clrx                    ; Columna 0
        jsr LCDSetXY            ; Posiciona en fila-columna 0
        lda #LCD_RS_HIGH        ; A  = 1
        jsr LCDSel              ; RS = HIGH
        clrx                    ; Contador temporal inicializado
LCDLOOP1008.L
        lda #LCD_SPACE_CHAR     ; Carga caracter espacio
        pshx                    ; Empuja contador a pila
        jsr LCDWr               ; Escribe caracter en blanco
        pulx                    ; Recupera contador de pila
        incx                    ; Incrementa contador
        cpx #LCD_MAX_COL        ; ¿Final de línea limpiado?
        blo LCDLOOP1008.L       ; NO, Seguir
        pula                    ; Si, recupera fila de la pila
        clrx                    ; Columna 0
        jsr LCDSetXY            ; Posiciona nuevamente en la línea
LCDOUT1008.K
        rts                     ; Retorna
```

```
;===========================================
; LCDWRCHAR:  Escribe un caracter en la
;             posición actual del cursor
; OBJETIVO    : Escribe un caracter en cual
;             quier posición de la LCD
; ENTRADA     : A    es el caracter a enviar
;             al LCD
; SALIDA      : Ninguna
; REGISTROS
; AFECTADOS : A, SP
;===========================================
LCDWrChar
        psha                    ; Empuja caracter a pila
        lda #LCD_RS_HIGH        ; A ejecutar comando
        jsr LCDSel              ; RS = LOW
        pula                    ; Recupera caracter de pila
        jsr LCDWr               ; Escribe caracter
        rts                     ; Retorna

;===========================================
; LCDWRMSG  : Escribe una cadena de carac-
;             teres
; OBJETIVO    : Escribe una cadena dada por
;             la dirección de una tabla
; ENTRADA     : H:X    contine la dirección
;             de la cadena a enviar
; SALIDA      : Ninguna
; REGISTROS
; AFECTADOS : CCR, A, H:X
;===========================================
LCDWrMsg
        sei                     ; Inhabilita interrupciones
        lda #LCD_RS_HIGH        ; Comando RS = 1
        jsr LCDSel              ; RS = 1
LCDWRITE1008.I
        lda ,x                  ; Carga caracter
        beq LCDOUT1008.J        ; ¿Cero?, salir
        pshx                    ; Empuja a pila
        jsr LCDWr               ; Envía al LCD
        pulx                    ; Recupera de pila
        aix #1                  ; Incrementa puntero
        bra LCDWRITE1008.I      ; Siguiente caracter
LCDOUT1008.J
        cli                     ; Habilita interrupciones
        rts                     ; Retorna
```

```
;===========================================
; LCDWRXY    : Ubica el cursor y escribe
; OBJETIVO   : Ubica cursor en posición X-Y
;              y escribe un caracter
; ENTRADA    : A (PUSH)
;              es el dato a desplegar
;              en la pantalla
;              A    es la fila en número de-
;              cimal
;              X    es la columna en número
;              decimal
; SALIDA     : Ninguna
; REGISTROS
; AFECTADOS  : CCR, SP, A
;===========================================
LCDWrXY
        jsr LCDSetXY            ; Posiciona en X-Y
        lda #LCD_RS_HIGH        ; A  = 1
        jsr LCDSel              ; RS = HIGH
        lda 3,SP                ; Recupera dato
        jsr LCDWr               ; Escribe el dato
        rts                     ; Retorna

;===========================================
; LCDWRMSGXY
;              : Escribe un mensaje en la po-
;              sición X-Y
; OBJETIVO   : Escribe un conjunto de carac-
;              teres en la posición X-Y
; ENTRADA    :
;              A     es la fila del mensaje
;              X (PUSH)
;              es la columna del men-
;              saje
;              H:X   es la dirección del
;              mensaje a escribir
; SALIDA     : Ninguna
; REGISTROS
; AFECTADOS  : X, SP, CCR
;===========================================
LCDWrMsgXY
        sei                     ; Inhabilita Interrupciones
        pshx                    ; Empuja parte baja de puntero a pila
        ldx 4,SP                ; Carga Columna
        jsr LCDSetXY            ; Posiciona en X-Y
        pulx                    ; Recupera parte baja del puntero
        jsr LCDWrMsg            ; Escribe cadena y habilita interrupciones
        rts                     ; Retorna
```

```
;===========================================
; LCDSEL      : Selecciona modo de instruc-
;               ciones o datos
; OBJETIVO    : RS = HIGH ó RS = LOW
; ENTRADA     : A    es el estado booleano a
;               imponer para control
;               de RS
;               A = 1    RS = HIGH
;               A = 0    RS = LOW
; SALIDA      : Ninguna
; REGISTROS
; AFECTADOS   : DDRX, PORTX
;===========================================
LCDSel
        beq LCDOUT1008.A        ; A = 0, Saltar a imponer RS bajo
        bset LCD_RS_PIN,LCD_RS_PORT
                                ; A = 1, Impone RS alto
        bra LCDOUT1008.B        ; Salir
LCDOUT1008.A
        bclr LCD_RS_PIN,LCD_RS_PORT
                                ; Impone RS bajo
LCDOUT1008.B
        rts                     ; Retorna

;===========================================
; LCDWR       : Escribe un caracter
; OBJETIVO    : Muestra un caracter en la
;               pantalla
; ENTRADA     : A    contiene el caracter a
;               desplegar
; SALIDA      : Ninguna
; REGISTROS
; AFECTADOS   : DDRDX, PORTX, A, X
;===========================================
LCDWr
        sta LCD_DATA_PORT       ; Almacena dato en puerto de LCD
        bset LCD_E_PIN,LCD_E_PORT
                                ; Levanta pin Enable
        ldx #5T                 ; XTAL
        lda #10T                ; A retardar +1 us
        jsr DelayNus            ; Retardar +1
        bclr LCD_E_PIN,LCD_E_PORT
                                ; Baja línea enable
        ldx #5T                 ; XTAL
        lda #60T                ; A retardar +40 us
        jsr DelayNus            ; Retarda +40 us y más
        rts                     ; Retorna
```

```
;=============================================
; LCDSETXY   : Cursor de LCD ubicado en
;              Posición de pantalla
; OBJETIVO   : Ubica cursor en posición X-Y
; ENTRADA    : A   es la fila en número de-
;              cimal
;              X   es la columna en número
;              decimal
; SALIDA     : Ninguna
; REGISTROS
; AFECTADOS  : CCR
;=============================================
LCDSetXY
        cmp #LCD_MAX_ROW        ; Compara A con máximo de filas
        bge LCDOUT1008.G        ; Mayor o igual?, salir
        cpx #LCD_MAX_COL        ; Compara X con máximo de columnas
        bge LCDOUT1008.G        ; Mayor o igual?, salir
        psha                    ; Empuja fila a la pila
        clra                    ; A = 0
        jsr LCDSel              ; RS = LOW
        pula                    ; Recupera A
        pshx                    ; Empuja columna a pila
        tsta                    ; Comprueba si es fila 0
        beq LCDOUT1008.C        ; Salir si es 0
        cmp #LCD_ROW1           ; Comprueba si es fila 1
        beq LCDOUT1008.D        ; Ir a offset de fila 1
        cmp #LCD_ROW2           ; Comprueba si es fila 2
        beq LCDOUT1008.E        ; Ir a offset de fila 2
        ldx #LCD_ROW3_OFFSET    ; Carga offset de fila 3
        bra LCDOUT1008.F        ; Salta a posicionar en fila
LCDOUT1008.C
        ldx #LCD_ROW0_OFFSET    ; Carga offset de fila 0
        bra LCDOUT1008.F        ; Salta a posicionar en fila
LCDOUT1008.D
        ldx #LCD_ROW1_OFFSET    ; Carga offset de fila 1
        bra LCDOUT1008.F        ; Salta a posicionar en fila
LCDOUT1008.E
        ldx #LCD_ROW2_OFFSET    ; Carga offset de fila 2
LCDOUT1008.F
        txa                     ; Carga columna
        add 1,sp                ; Suma fila + columna
        jsr LCDWr               ; Posiciona en Fila, repite X veces
        pulx                    ; Posiciona stack
LCDOUT1008.G
        rts                     ; Retorna
```

Listado 60. NT1008 – LCD – LCD.inc. Rutinas reusables para configuración y uso de la pantalla de cristal líquido de caracteres.

```
;=====================================================================
; ARCHIVO      :  VECTORSJL3.inc
; PROPÓSITO    :  Definir el vector de búsqueda de cada interrupción
; LENGUAJE     :  IN-LINE ASSEMBLER
; ---------------------------------------------------------------------
; HISTORIAL
;   DD MM AA
;   22   08   04 Creado.
;   22   08   04 Modificado.
;=====================================================================

;=============================================
;       Vector de Reinicio del Sistema
;=============================================
          org RESET_VEC                 ; Puntero Vec - RESET
          dw START                      ; al darse reset salta a Start
```

Listado 61. NT1008 – LCD – VECTORS.inc. Nota: Solo se listan los vectores usados. Como no existen interrupciones usadas, el vector declarado y su interrupción son los de reinicio del sistema y se usa en el programa principal.

3.8.8 Conclusión

Las pantallas de cristal líquido sirven no solo para desplegar caracteres, sino para tener una disponibilidad con el usuario de acción de lo que puede estar pasando, como en este caso, el despliegue del voltaje medido por el ADC o para desplegar un menú y darle la opción al usuario de lo que pueda suceder.

También se implementaron las rutinas utilitarias de la pantalla las cuales permiten inicializar, posicionar, borrar y desplegar mensajes de una manera más saludable y menos viable a cometer errores. Para mayor referencia de cómo utilizar estas rutinas se pueden encontrar en los apéndices, en la sección de LCD, Apéndice K.

3.8.9 Referencias

3.8.9.1 "Embedded Systems Building Blocks, Second Edition" - "Complete and Ready-to-Use Modules in C"
Autor: Jean J. Labrosse
Recurso: Capítulo 05 – Character LCD Modules

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

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

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

NT0026 – Módulos – Rutinas reusables para programación, Apéndices en Tesis.

3.8.9.5 Displays de Cristal Líquido
(a) www.microbotica.es/web/downl/docs/display.pdf

3.8.9.6 Notas de Aplicación sobre Control de Pantalla de Cristal Líquido
(a) http://www.freescale.com/files/microcontrollers/doc/app_note/AN1287.pdf
(b) http://www.freescale.com/files/microcontrollers/doc/app_note/AN1745.pdf
(c) http://www.freescale.com/files/microcontrollers/doc/app_note/AN1762.pdf
(d) http://ww1.microchip.com/downloads/en/AppNotes/00658a.pdf

3.8.10 Problemas Propuestos

3.8.10.1 Utilice la LCD e inicializarla para modo de 4 bits, sin parpadeo de cursor, sin posicionador de cursor.

3.8.10.2 Utilice un microcontrolador QT4 para controlar una LCD en modo de 8 bits. Nota: Utilice un registro de corrimiento como el de la figura.

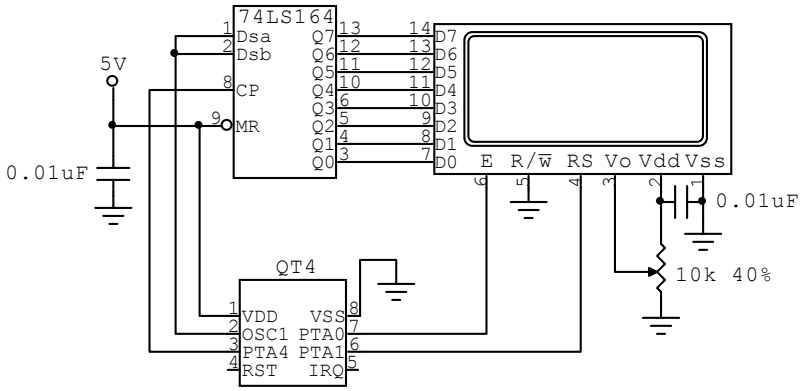


Figura 213. Control de una LCD por medio de un Microcontrolador 68HC908QT4. Esquema para el reto 3.8.9.2.