

Manejo de un módulo LCD con microcontrolador PIC

Los módulos de cristal líquido se pueden conectar fácilmente a microcontroladores, microprocesadores o computadoras. Además, permiten que los proyectos tengan una óptima presentación y funcionalidad.

EDISON DUQUE C.

Cuando se trabaja en diseño de circuitos electrónicos, es frecuente encontrarse con la necesidad de visualizar un mensaje, que tiene que ver con el estado de la máquina a controlar, con instrucciones para el operario, o si es un instrumento de medida, mostrar el valor registrado. En la mayoría de los casos, recurrimos a los displays de siete segmentos, pero estos además de no mostrar caracteres alfanuméricos ni ASCII, tienen un elevado consumo de corriente y son un poco dispendiosos de manejar, cuando se requiere hacer multiplexaje.

Los módulos de cristal líquido o LCD, solucionan estos inconvenientes y presentan algunas ventajas, como un menor consumo de corriente, no hay que preocuparse por hacer multiplexaje, no hay que hacer tablas especiales con los caracteres que se desea mostrar, se pueden conectar fácilmente con microprocesadores o microcontroladores y además, los proyectos adquieren una óptima presentación y funcionalidad.

Este proyecto consiste en manejar un módulo LCD, utilizando como elemento de control un microcontrolador PIC. En principio, se nombran las características más importantes de los módulos, luego se muestra la forma de conectarlos con el microcontrolador y se hacen programas simples para escribir mensajes en la pantalla. Para terminar, se presenta como ejemplo, la construcción de un termómetro digital con visualización en un módulo LCD.

Displays de cristal líquido o LCD

Antes de mostrar la forma de conectar estos módulos con el microcontrolador, haremos un pequeño recuento de las principales características que ellos tienen, las cuales nos

servirán para entender mejor los programas y los diagramas que se muestran más adelante:

- Los módulos LCD se encuentran en diferentes presentaciones, por ejemplo (2x16 líneas por 16 caracteres), 2x20, 4x20, 4x40, etc. La forma de utilizarlos y sus interfaces son similares, por eso, los conceptos vistos aquí se pueden emplear en cualquiera de ellos. En nuestro caso, trabajamos con un display de 2x16, ya que es de bajo costo, se consigue fácilmente en el comercio y tiene un tamaño suficiente para la mayoría de las aplicaciones.
- La figura 1 muestra dos tipos de configuración de pines que se encuentran comúnmente, aunque cambian su ubicación, estos conservan las mismas funciones. Algunos módulos LCD tienen luz posterior o “backlight”, para mejorar su visualización, ésta se maneja a través de dos pines que normalmente se conectan a +5V y a tierra. Para evitar

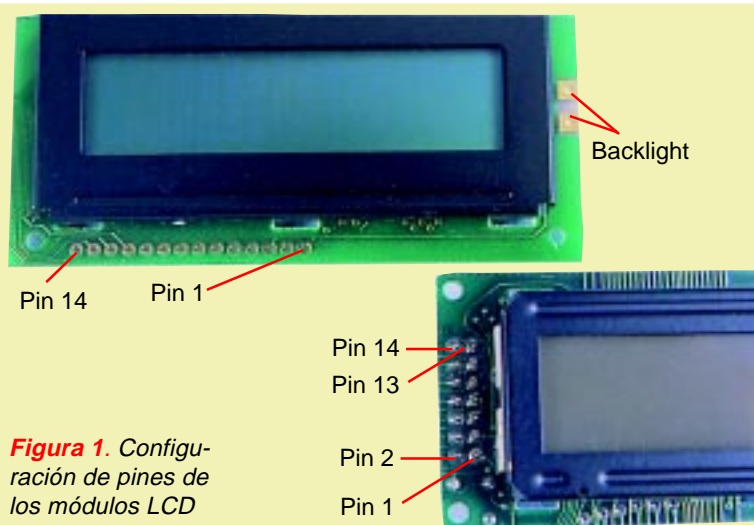


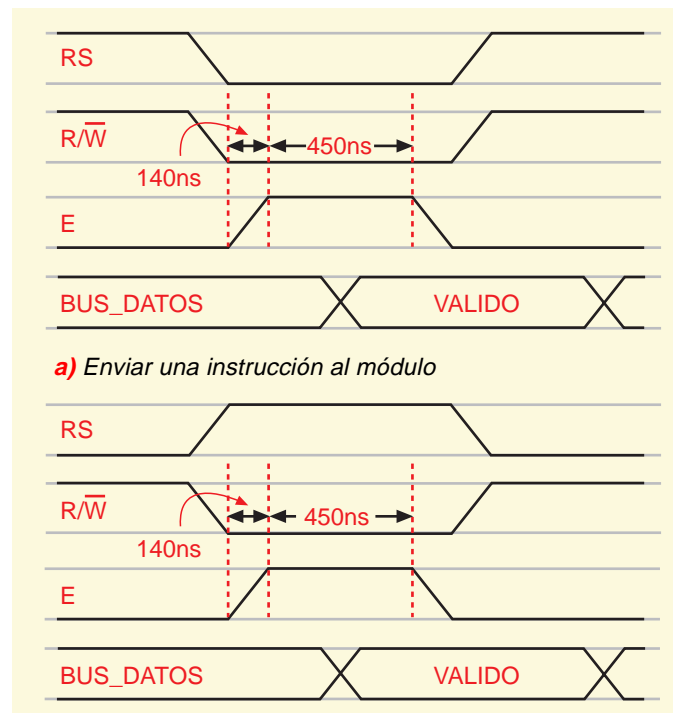
Figura 1. Configuración de pines de los módulos LCD

Terminal	Símbolo	Nombre y Función
1	Vss	Tierra, 0V
2	Vdd	Alimentación +5V
3	V _o	Ajuste de Voltaje de contraste
4	\overline{RS}	Selección Control/Dato
5	R/W	Lectura/escritura en LCD
6	E	Habilitación
7	D0	D0 Bit menos significativo
8	D1	D1
9	D2	D2
10	D3	D3
11	D4	D4
12	D5	D5
13	D6	D6
14	D7	D7 Bit más significativo

Tabla 1. Función de los pines del módulo LCD

que se presenten altas temperaturas, debido a la luz posterior, estos pines se deben manejar de manera pulsante (encendiendo y apagando), con una frecuencia de aproximadamente 60 Hz.

- Los pines de conexión de estos módulos incluyen un bus de datos de 8 bits, un pin de habilitación (E), un pin de selección, que indica que el dato es una instrucción o un caracter del mensaje (RS) y un pin que indica si se va a escribir o leer en el módulo LCD (R/ \overline{W}). La tabla 1 describe la función de cada uno de ellos.
- Según la operación que se desee realizar sobre el módulo de cristal líquido, los pines de control E, RS y R/ \overline{W} deben tener un estado determinado. Además, debe tener en el bus de datos un código que indique un caracter para mostrar en la pantalla o una instrucción de control. En la figura 2 se muestra el diagrama de tiempos que se debe cumplir para manejar el módulo.
- El módulo LCD responde a un conjunto especial de instrucciones, estas deben ser enviadas por el microcontrolador o sistema de control al display, según la operación que se requiera. Estas instrucciones se emplean en los ejemplos que realizaremos más adelante, en ellos se explica la forma de utilizarlas. En la figura 3 se muestran las instrucciones del módulo.
- La interface entre el microcontrolador y el display de cristal líquido se puede hacer con el bus de datos trabajando a 4 u 8 bits. Las señales de control trabajan de la misma forma en cualquiera de los dos casos, la diferencia se establece en el momento de iniciar el sistema, ya que existe una instrucción que permite establecer dicha configuración. Estas conexiones se explican más adelante en forma detallada.



b) Escribir un caracter en la pantalla

Figura 2. Diagrama de tiempo de las señales del módulo

- Los caracteres que se envían al display se almacenan en la memoria RAM del módulo. Existen posiciones de memoria RAM, cuyos datos son visibles en la pantalla y otras que no son visibles, estas últimas se pueden utilizar para guardar caracteres que luego se desplazan hacia la parte visible. En la figura 4 se muestran las direcciones de memoria visibles y no visibles, que conforman las dos líneas de caracteres del módulo.
- Es importante anotar que solamente se pueden mostrar caracteres ASCII de 7 bits, por lo tanto algunos caracteres especiales no se pueden ver (se debe tener a la mano una tabla de los caracteres ASCII para conocer los datos que son prohibidos) . Por otra parte, se tiene la opción de crear caracteres especiales (creados por el programador), y almacenarlos en la memoria RAM que posee el módulo.

[illegible]

Figura 3a. Instrucciones de los módulos LCD

Significado de las abreviaturas

I/D	= 1 Incrementa
	= 0 Decrementa
S	= 1 Desplaza el mensaje en la pantalla
	= 0 Mensaje fijo en la pantalla
D	= 1 Encender (activar) la pantalla
	= 0 Apagar la pantalla (desactivar)
C	= 1 Activar cursor
	= 0 Desactivar cursor
B	= 1 Parpadea caracter señalado por el cursor
	= 0 No parpadea el caracter
S/C	= 1 Desplaza pantalla
	= 0 Mueve cursor
RL	= 1 Desplazamiento a la derecha
	= 0 Desplazamiento a izquierda
DL	= 1 Datos de ocho bits
	= 0 Datos de cuatro bits
BF	= 1 Durante operación interna del módulo
	= 0 Finalizada la operación interna

Figura 3b. Instrucciones de los módulos LCD

Interface con microcontrolador a 8 bits

El proyecto consiste en conectar el módulo de cristal líquido a un microcontrolador PIC16C84, utilizando el bus de datos a 8 bits. En este caso, se emplea el PIC16C84 por su facilidad de reprogramación y fácil manejo. En la figura 5 se muestra el diagrama de conexiones para este caso.

Para estos ejercicios en particular, sólo nos interesa escribir datos en la pantalla (no hacer lectura); por lo tanto el pin de selección de lectura/escritura (R/W) en el display, se conecta a tierra. El puerto B del microcontrolador se utiliza como bus de datos, y el puerto A se encarga de generar las señales de control.

En el oscilador del PIC16C84 se emplea un cristal de 4 MHz, por lo tanto tenemos ciclos de instrucción de un microsegundo. Para el módulo LCD, se emplea un potenciómetro de 5KΩ, conectado entre +5V y tierra, para controlar el contraste de la pantalla.

En la figura 6 se muestra el listado del programa. Para este caso, el ejemplo consiste en hacer circular un mensaje en la línea superior de la pantalla. La explicación de los pasos contenidos en él es la siguiente:

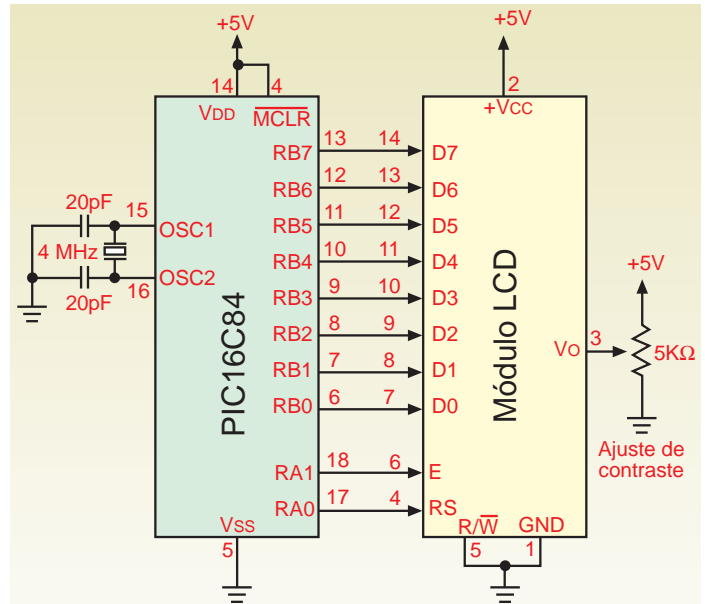


Figura 5. Conexión entre el PIC y el display a 8 bits

1. Se programan los puertos según las conexiones que se tienen en el circuito.
2. Se debe inicializar el módulo LCD. El primer dato que se envía (30H) le dice al módulo que la comunicación es a 8 bits y que se empleará solo una línea de caracteres. El dato se puede deducir con la lista de las instrucciones que se muestra en la figura 3. La rutina llamada CONTROL, se encarga de generar las señales y los tiempos necesarios para que exista una correcta comunicación.
3. El segundo dato (07H), le dice al módulo que el mensaje se va a desplazar en la pantalla.
4. El tercer dato (0CH), hace que se encienda el display.
5. El siguiente paso es entrar en un ciclo que hace una lectura de la tabla donde se encuentra el mensaje y lo lleva a la memoria del módulo LCD. Cuando se termina de enviar todos los caracteres, se inicia el ciclo nuevamente.

Las rutinas CONTROL y DATO emplean las mismas instrucciones. La única diferencia es que cada una le da el nivel lógico adecuado al pin RS, que indica si el dato enviado es un caracter del mensaje (un dato) o una instrucción de control.

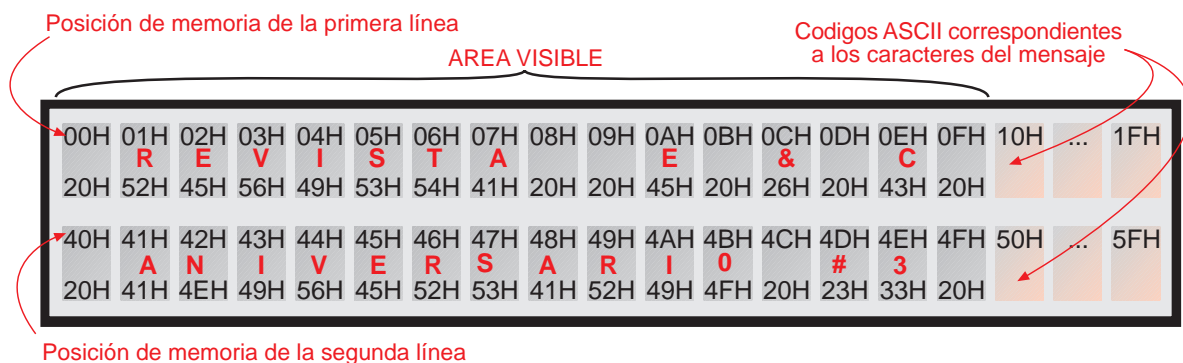


Figura 4. Memoria RAM del módulo de cristal líquido

;ESTE PROGRAMA HACE QUE UN MENSAJE CIRCULE EN LA ;PANTALLA DE UN MODULO LCD							
PC	EQU	2H	;	RETLW	"Y"		
STATUS	EQU	3H	;	RETLW	" "		
PTOA	EQU	5H	;	RETLW	"U"		
PTOB	EQU	6H	;	RETLW	"N"		
R0C	EQU	0CH	;	RETLW	" "		
R0D	EQU	0DH	;	RETLW	"P"		
R13	EQU	13H	;	RETLW	"R"		
Z	EQU	2H	;	RETLW	"O"		
C	EQU	0H	;	RETLW	"S"		
W	EQU	0H	;	RETLW	"P"		
R	EQU	1H	;	RETLW	"E"		
E	EQU	1H	;	RETLW	"R"		
RS	EQU	0H	;	RETLW	"O"		
; ***** PROGRAMA PRINCIPAL *****				RETLW	" "		
	ORG	00	;VECTOR DE RESET	RETLW	" "		
	GOTO	INICIO		RETLW	" "		
	ORG	05H		RETLW	" "		
RETARDO	MOVLW	0FFH		RETLW	0		
	MOVWF	R13		INICIO	MOVLW	0FCH	;PROGRAMACION DE PUERTOS
DECRE	DECFSZ	R13,R			TRIS	PTOA	;SEGUN EL CIRCUITO
	GOTO	DECRE			MOVLW	00H	
	RETLW	0			TRIS	PTOB	
CONTROL	BCF	PTOA,RS	;ESTA RUTINA GENERA LAS	BEGIN	MOVLW	30H	;INICIA MODULO A 8 BITS
	GOTO	DAT02	;SEÑALES DE CONTROL Y ENVIA		CALL	CONTROL	;Y 1 LINEA
DAT0	BSF	PTOA,RS	;EL DATO AL MODULO		MOVLW	07H	;SELECCIONA MODO DE ROTAR
DAT02	BSF	PTOA,E	;CON INTERFACE DE 8 BITS		CALL	CONTROL	;EL MENSAJE
	MOVWF	PTOB			MOVLW	0CH	;ACTIVA DISPLAY
	CALL	RETARDO		MUESTRA	CALL	CONTROL	
	BCF	PTOA,E			MOVLW	0	;INICIA ENVIO DE DATOS
	CALL	RETARDO			MOVWF	R0C	;AL MODULO
TABLA2	RETLW	0		CICLO	MOVF	R0C,W	;HACE BARRIDO DE LA TABLA
	ADDWF	PC,R	;MENSAJE A SER ROTADO		CALL	TABLA2	
	RETLW	"F"			CALL	DAT0	
	RETLW	"E"			MOVLW	09FH	;RETARDO ENTRE CARACTERES
	RETLW	"L"			MOVWF	R0D	
	RETLW	"I"		RETA1	CALL	RETARDO	
	RETLW	"Z"			CALL	RETARDO	
	RETLW	" "			DECFSZ	R0D,R	
	RETLW	"N"			GOTO	RETA1	
	RETLW	"A"			INCF	R0C,R	;SIGUE CON EL PROXIMO
	RETLW	"V"			MOVLW	23H	;CARACTER DEL MENSAJE
	RETLW	"I"			XORWF	R0C,W	;PREGUNTA SI TERMINO EL
	RETLW	"D"			BTFSS	STATUS,Z	;MENSAJE PARA EMPEZAR
	RETLW	"A"			GOTO	CICLO	;OTRA VEZ
	RETLW	"D"			GOTO	MUESTRA	
	RETLW	" "			END		

Figura 6. Programa que hace circular un mensaje en la pantalla (interface a 8 bits)

Interface con microcontrolador a 4 bits

La conexión entre el PIC y el módulo LCD se hará empleando un bus de datos de 4 bits, en este caso, se utilizarán los 4 pines de mayor peso del puerto B (RB4-RB7) del microcontrolador. Las señales de control (RS y E), se generarán con los dos pines de menor peso del puerto B (RB0 y RB1). Con esto, se libera todo el puerto A y los pines RB2 y RB3 del microcontrolador, los cuales se podrían emplear en otras funciones.

El software que se implementará en el microcontrolador, se encarga de mostrar un mensaje en el display. Dicho mensaje ocupa las dos líneas de la pantalla y permanece fijo, a diferencia del ejemplo anterior, en el cual el mensaje se desplazaba.

En la figura 8 se muestra el listado del programa que realiza la tarea descrita anteriormente, este sufre unas modificaciones respecto al ejemplo anterior. Los pasos más importantes son:

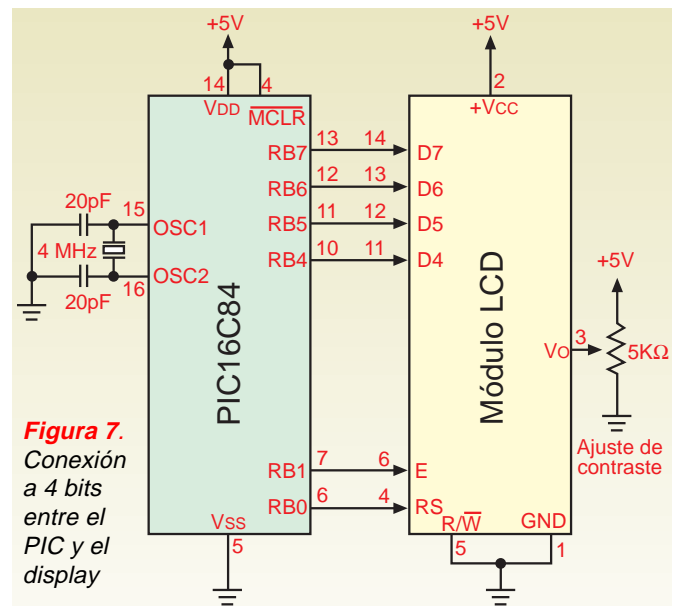


Figura 7. Conexión a 4 bits entre el PIC y el display

```
; ESTE PROGRAMA HACE QUE UN MENSAJE SE REPITA
; EN UN MODULO LCD DE 2 LINEAS CON 16 CARACTERES
```

```
PC      EQU      2H      ;CONTADOR DE PROGRAMA
STATUS  EQU      3H      ;REGISTRO DE ESTADOS
PTOA     EQU      5H      ;PUERTOS
PTOB     EQU      6H
R0C      EQU      0CH      ;
R0D      EQU      0DH      ;
R0E      EQU      0EH      ;
R13      EQU      13H      ;
Z        EQU      2H      ;
C        EQU      0H      ;
W        EQU      0H      ;
R        EQU      1H      ;
E        EQU      1H
RS       EQU      0H
```

```
; ***** PROGRAMA PRINCIPAL *****
```

```
ORG      00      ;Vector de reset
GOTO     INICIO   ;INICIA PROGRAMA PRINCIPAL
ORG      05H
```

```
RETARDO  MOVLW    0FFH
          MOVWF    R13
DECRE     DECFSZ   R13,R
          GOTO     DECRE
          RETLW    0
```

```
CONTROL  BCF      PTOB,RS ;ESTA RUTINA GENERA LAS SEÑALES
          GOTO     DATO2   ;DE CONTROL PARA ESCRIBIR EN EL
DATO      BSF      PTOB,RS ;MODULO LCD
DATO2     BSF      PTOB,E  ;UTILIZA LA INTERFACE A 4 BITS
          MOVWF    R0E
          MOVLW    0FH
          ANDWF    PTOB,R
          MOVF     R0E,W
          ANDLW    0F0H
          IORWF    PTOB,R
          CALL     RETARDO
          BCF      PTOB,E
          CALL     RETARDO
          BSF      PTOB,E
          MOVLW    0FH
          ANDWF    PTOB,R
          SWAPF    R0E,W
          ANDLW    0F0H
          IORWF    PTOB,R
          CALL     RETARDO
          BCF      PTOB,E
          CALL     RETARDO
          RETLW    0
```

```
TABLA     ADDWF    PC,R    ;MENSAJE QUE SE MUESTRA
          RETLW    " "
          RETLW    "R"
          RETLW    "E"
          RETLW    "V"
          RETLW    "I"
          RETLW    "S"
          RETLW    "T"
          RETLW    "A"
          RETLW    " "
          RETLW    " "
          RETLW    "E"
          RETLW    " "
```

```
RETLW    "&"
RETLW    " "
RETLW    "C"
RETLW    " "
RETLW    " " ;MENSAJE DE LA SEGUNDA LINEA
RETLW    " "
RETLW    "A"
RETLW    "N"
RETLW    "I"
RETLW    "V"
RETLW    "E"
RETLW    "R"
RETLW    "S"
RETLW    "A"
RETLW    "R"
RETLW    "I"
RETLW    "O"
RETLW    " "
RETLW    "#"
RETLW    "3"
RETLW    " "
RETLW    0
```

```
INICIO    MOVLW    0FFH ;PROGRAMACION DE PUERTOS
          TRIS     PTOA ;SEGUN EL CIRCUITO
          MOVLW    0CH
          TRIS     PTOB ;
```

```
BEGIN     MOVLW    02H ;INICIA DISPLAY A 4 BITS
          CALL     CONTROL
          MOVLW    28H ;DISPLAY A 4 BITS Y 2 LINEAS
          CALL     CONTROL
          MOVLW    0CH ;ACTIVA DISPLAY
          CALL     CONTROL
          MOVLW    06H ;HACE QUE EL MENSAJE PERMANEZCA
                      ;FIJO (SIN ROTAR)
```

```
          CALL     CONTROL
          MOVLW    01H ;BORRA DISPLAY
          CALL     CONTROL
```

```
MUESTRA   MOVLW    0 ;INICIA CONTADOR DE CARACTERES
          MOVWF    R0C
          MOVF     R0C,W ;HACE BARRIDO DE LA TABLA
          CALL     TABLA
          CALL     DATO
          MOVLW    0FFH ;RETARDO ENTRE CARACTERES
          MOVWF    R0D
```

```
          CALL     RETARDO
          DECFSZ   R0D,R
          GOTO     RETA1
          INCF     R0C,R ;SIGUE CON LA TABLA
          MOVLW    11H
          SUBWF    R0C,W ;PREGUNTA SI ESTA MOSTRANDO EL
          BTFS     STATUS,C ;MENSAJE DE LA SEGUNDA LINEA
          GOTO     CICLO ;
```

```
          MOVLW    11H ;PREGUNTA SI ES LA PRIMERA VEZ
          XORWF    R0C,W ;QUE ENTRA A LA SEGUNDA LINEA
          BTFS     STATUS,Z ;PARA IR A INICIAR EL PUNTERO
          GOTO     LINE2 ;DE LA RAM DEL MODULO LCD
          MOVLW    0C0H ;UBICA PUNTERO DE LA RAM DEL
          CALL     CONTROL ;MODULO LCD EN LA SEGUNDA LINEA
```

```
          MOVLW    21H ;PREGUNTA SI TERMINO LA SEGUNDA
          XORWF    R0C,W ;LINEA PARA IR A INICIAR DE
          BTFS     STATUS,Z ;NUEVO EL MENSAJE O PARA
          GOTO     CICLO ;CONTINUAR EN LA SEGUNDA PARTE
                      ;DEL MENSAJE
          GOTO     BLANK ;VA A REINICIAR EL MENSAJE
          END
```

Figura 8. Programa que escribe un mensaje fijo en la pantalla (interface a 4 bits)

1. Se programan los puertos según el circuito.
2. Se debe inicializar el módulo LCD. El primer dato que se envía (02H) le dice al módulo que la comunicación se va a realizar a 4 bits.
3. El segundo dato (28H) ratifica que la comunicación es a 4 bits y que se emplearán las dos líneas de caracteres del display. El dato se puede deducir con la lista de las instrucciones que se muestra en la figura 3. La rutina llamada CONTROL,

- se encarga de generar las señales y los tiempos necesarios para que exista una correcta comunicación. Debe notarse que esta rutina ha cambiado respecto al primer ejemplo.
4. El tercer dato (0CH), hace que se encienda el display.
 5. El cuarto dato (06H), le indica al módulo LCD que el mensaje debe permanecer fijo en la pantalla. Recuerde que en el ejemplo anterior, para desplazarlo se usó el dato 07H.

Pasa a la página 62...

...Viene de la página 9

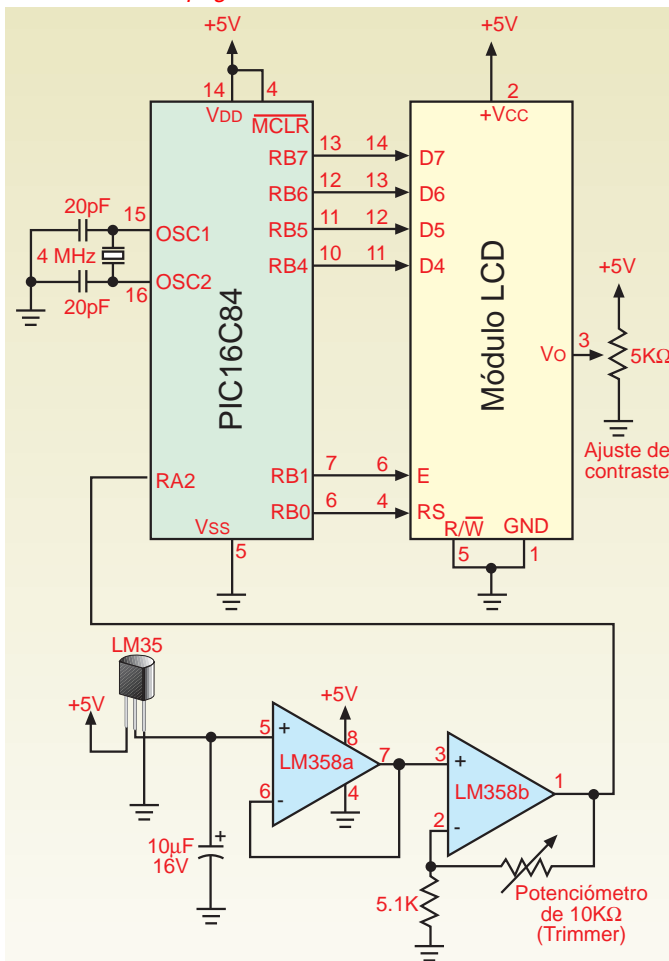


Figura 9. Diagrama esquemático del termómetro digital

6. El siguiente paso, es entrar en un ciclo que hace un borrado de la pantalla (con el comando 01H) y luego una lectura de la tabla donde se encuentra el mensaje, para llevarlo a la memoria del módulo LCD. Esta lectura incluye una verificación del momento en que se llena la primera línea de caracteres, con el fin de ubicar el puntero de la RAM del módulo LCD, en la segunda línea de caracteres. Para esto se envía el comando C0H al display (este dato se puede deducir observando la lista de instrucciones de la figura 3).

Cuando se está escribiendo en la segunda línea de caracteres, se verifica en que momento se llenó el área visible del display (se compara el puntero de la tabla con 21H), en ese momento el programa vuelve a iniciar el ciclo de borrar el display y escribir el mensaje en la pantalla.

Las rutinas CONTROL y DATO han cambiado de manera sustancial, como en este caso la interface es a cuatro bits, se debe enviar primero el *nibble* alto (4 bits de mayor peso) del dato y luego el *nibble* bajo (4 bits de menor peso). En el momento de enviar cada uno de los datos de 4 bits, las señales de control deben comportarse de la misma forma como si fuera un dato completo. Por eso en la rutina se ve que la señal RS conserva el nivel lógico adecuado y la señal E genera los dos pulsos que se requieren (el primero para el *nibble* alto y el segundo para el bajo).

Termómetro digital con LCD

Como ejemplo de la utilización del display de cristal líquido, vamos a implementar un termómetro, que muestra la temperatura medida en grados centígrados, en la pantalla del display. Como elemento de control se utiliza un PIC16C71, debido a que posee el convertidor análogo a digital necesario para leer el sensor de temperatura.

Para medir la temperatura emplearemos un LM35. Este sensor tiene tres pines: Alimentación, tierra y la salida análoga. Este dispositivo presenta en su salida una variación de 10 mV/°C, por lo tanto el valor de la temperatura se puede obtener directamente, sin necesidad de hacer modificaciones al dato obtenido.

El LM35 puede trabajar en un rango de temperatura entre -55 y 150°C, la fuente de alimentación positiva puede estar entre 4 y 30 Voltios. Además, su precisión es de 0.5°C. Este elemento viene en un encapsulado plástico TO92 y tiene la misma apariencia de un transistor.

El módulo de cristal líquido se conecta al microcontrolador, utilizando interface de 4 bits. En este ejercicio, conectamos la entrada análoga al pin RA2/AN2 del microcontrolador y el voltaje de referencia del convertidor análogo a digital, se configuró para que sea el voltaje de alimentación del PIC (VDD). En la figura 9 se muestra el diagrama esquemático del termómetro digital.

La máxima lectura del convertidor análogo a digital (0FFH) se obtiene cuando la entrada análoga sea superior a 5 voltios. De tal manera, que si hacemos la división de 5 Voltios entre 255 (FFH), obtenemos una relación de aproximadamente 20 mV por unidad. Por ejemplo, si la entrada análoga está en 160 mV, el valor digital que entrega el convertidor es 8 (160/20). Dado lo anterior, la salida del sensor LM35 se debe amplificar por dos, esto se hace con el fin de tener una salida que varíe 20mV/°C. Para así garantizar que el dato que entrega el convertidor análogo a digital, sea directamente el valor de la temperatura medida.

Para realizar esta función, se emplea el amplificador operacional LM358. El primero de ellos se configura como seguidor de tensión para acoplar impedancias, y el segundo se configura como amplificador no inversor de ganancia dos. El ajuste exacto de la ganancia, se hace con un potenciómetro de 10 vueltas (trimmer) en la realimentación del amplificador. De esta forma, sólo se debe hacer la comprobación con un multímetro de buena precisión, para garantizar que el voltaje de salida es el doble del voltaje de entrada del operacional.

En la figura 10 se muestra el listado del programa, los comentarios explican lo que se hace en cada una de las instrucciones. Este incluye todo el programa de la interface a 4 bits con el display de cristal líquido, una sección que se encarga de leer el convertidor análogo a digital, y una sección que convierte el dato hexadecimal obtenido luego de la conversión, a su equivalente en decimal, para ser mostrado en la pantalla. 📺

; ESTE PROGRAMA TOMA EL VALOR ENTREGADO POR EL SENSOR DE ; TEMPERATURA Y LO MUESTRA EN LA PANTALLA DEL MODULO LCD				TABLA	ADDWF	PC,R	; MENSAJE QUE SE MUESTRA
PC	EQU	2H	; CONTADOR DE PROGRAMA		RETLW	" "	; LA TEMPERATURA
STATUS	EQU	3H	; REGISTRO DE ESTADOS		RETLW	"L"	; ACTUAL ES 025'C
PTOA	EQU	5H	; PUERTOS		RETLW	"A"	
PTOB	EQU	6H			RETLW	" "	
ADCON0	EQU	8H	; REGISTROS DEL CONVERTIDOR A/D		RETLW	"T"	
TRISA	EQU	85H			RETLW	"E"	
TRISB	EQU	86H			RETLW	"M"	; CONTINUA EL MENSAJE
ADCON1	EQU	88H				; CONTINUA EL MENSAJE
ADRES	EQU	9H			RETLW	" "	
R0C	EQU	0CH			RETLW	"C"	
R0D	EQU	0DH			RETLW	" "	
R0E	EQU	0EH			RETLW	0	
UNIDAD	EQU	10H		INICIO	BSF	STATUS,RP0	; SELECCIONAR PAGINA 1
DECENA	EQU	11H			MOVLW	0FFH	; CONFIGURA PTOA COMO ENTRADAS
CENTENA	EQU	12H			MOVWF	TRISA	
R14	EQU	14H			BCF	ADCON1,PCFG1	; CONFIGURA ENTRADAS
R1B	EQU	1BH			BCF	ADCON1,PCFG0	; ANALOGAS
RP0	EQU	5H			MOVLW	0CH	; CONFIGURA PTOB
Z	EQU	2H			MOVWF	TRISB	
C	EQU	0H			BCF	STATUS,RP0	; VUELVE A PAGINA 0
W	EQU	0H		BEGIN	MOVLW	02H	; INICIA DISPLAY A 4 BITS
R	EQU	1H			CALL	CONTROL	
PCFG0	EQU	0	; BITS DEL REGISTRO ADCON1		MOVLW	28H	; DISPLAY A 4 BITS Y 2 LINEAS
PCFG1	EQU	1			CALL	CONTROL	
ADIF	EQU	1H	; BITS DEL REGISTRO ADCON0			; RUTINA IDENTICA AL EJEMPLO
GO	EQU	2H				; ANTERIOR
E	EQU	1H				
RS	EQU	0H		LINEA2	MOVLW	0C0H	; UBICA PUNTERO DE LA RAM DEL
					CALL	CONTROL	; MODULO LCD EN LA SEGUNDA LINEA
				LINE2	MOVLW	20H	; PREGUNTA SI TERMINO LA SEGUNDA
					XORWF	R0C,W	; LINEA PARA IR A INICIAR
					BTFSS	STATUS,Z	; DE NUEVO EL MENSAJE 0 PARA
				GOTO	CICLO		; CONTINUAR EN LA SEGUNDA PARTE
							; DEL MENSAJE
							; CONVERSION ANALOGO A DIGITAL
				MIDE	MOVLW	B'01010000'	; CONFIGURA EL CONVERTIDOR
					MOVWF	ADCON0	; VELOCIDAD DE CONVERSION Y
							; ADIF=0 ANTES DE EMPEZAR
					BSF	ADCON0,0	
					MOVLW	0FH	
					MOVWF	R1B	
				DEC	DECFSZ	R1B,R	
					GOTO	DEC	
					BSF	ADCON0,GO	; EMPEZAR LA CONVERSION
					MOVLW	0FH	
					MOVWF	R1B	
				DE2	DECFSZ	R1B,R	
					GOTO	DE2	
					CONSU	NOP	
					BTFSS	ADCON0,ADIF	; PREGUNTA FIN DE CONVERSION
					GOTO	CONSU	; SI NO HA TERMINADO SIGUE
					BCF	ADCON0,0	; ESPERANDO
					BCF	ADCON0,ADIF	; APAGA EL CONVERTIDOR
					MOVF	ADRES,W	; DEBE CONVERTIR EL RESULTADO
					MOVWF	UNIDAD	
					CALL	DECIMAL	; CONVERTIR EL DATO A DECIMAL
					MOVLW	0CBH	; UBICO PUNTERO DE RAM DONDE VA
					CALL	CONTROL	; EL NUMERO LEIDO
					MOVF	CENTENA,W	; PRIMER DIGITO = CENTENAS
					ADDLW	30H	
					CALL	DATO	
					MOVLW	0CCH	; UBICO PUNTERO DE RAM DONDE VA
					CALL	CONTROL	; EL NUMERO LEIDO
					MOVF	DECENA,W	; SEGUNDO DIGITO = DECENAS
					ADDLW	30H	
					CALL	DATO	
					MOVLW	0CDH	; UBICO PUNTERO DE RAM DONDE VA
					CALL	CONTROL	; EL NUMERO LEIDO
					MOVF	UNIDAD,W	; SEGUNDO DIGITO = UNIDADES
					ADDLW	30H	
					CALL	DATO	
					CALL	RETAR2	
					CLRF	ADRES	
					GOTO	MIDE	; TERMINA Y VUELVE A MEDIR
					END		; LA TEMPERATURA
; ***** PROGRAMA PRINCIPAL *****							
	ORG	00	; Vector de reset				
	GOTO	INICIO	; Va a iniciar programa principal				
	ORG	05H					
RETARDO	MOVLW	0FFH					
	MOVWF	R1B					
DECRE	DECFSZ	R1B,R					
	GOTO	DECRE					
	RETLW	0					
RETAR2	MOVLW	0FFH					
	MOVWF	R14					
DECR2	CALL	RETARDO					
	CALL	RETARDO					
	DECFSZ	R14,R					
	GOTO	DECR2					
	RETLW	0					
			; RUTINA QUE CONVIERTE BINARIO EN BCD				
DECIMAL	CLRF	DECENA	; BORRAR REGISTROS DE TRABAJO				
	CLRF	CENTENA					
	MOVLW	D'100'					
OTRA	SUBWF	UNIDAD,R	; RESTAR 100 AL VALOR INICIAL				
	BTFSS	STATUS,C	; VERIFICA EL CARRY				
	GOTO	SUM	; SI ES CERO DEJA DE RESTAR 100				
	INCF	CENTENA	; SI ES 1 INCREMENTA CENTENA				
	GOTO	OTRA	; VOLVER A RESTAR				
SUM	ADDWF	UNIDAD	; SUMARLE 100				
	MOVLW	D'10'					
REPITE	SUBWF	UNIDAD	; RESTAR 10 AL VALOR				
	BTFSS	STATUS,C	; VERIFICA EL CARRY				
	GOTO	SUM1	; SI ES 0 DEJA DE RESTAR				
	INCF	DECENA	; SI ES 1 INCREMENTA DECENA				
	GOTO	REPITE					
SUM1	ADDWF	UNIDAD	; SUMARLE 10 AL VALOR				
	RETLW	0	; EL RESULTADO QUEDA EN CENTENA,				
			; DECENA Y UNIDAD				
CONTROL	BCF	PTOB,RS	; ESTA RUTINA GENERA LAS SEÑALES				
	GOTO	DAT02	; DE CONTROL PARA ESCRIBIR EN EL				
DAT0	BSF	PTOB,RS	; MODULO LCD Y ENTREGA EL DATO				
DAT02	BSF	PTOB,E	; A SER MOSTRADO EN LA PANTALLA				
	MOVWF	R0E	; UTILIZA LA INTERFACE A 4 BITS				
	MOVLW	0FH					
		; ES LA MISMA RUTINA DEL				
		; EJEMPLO ANTERIOR				
	CALL	RETARDO					
	RETLW	0					

Figura 10. Programa del termómetro digital