SISTEMAS ELECTRÓNICOS Y AUTOMÁTICOS
PRACTICAS DE MICROCONTROLADORES PIC

PRÁCTICA 8:
El módulo LCD (II)

- ***Más ejemplos de funcionamiento***

---

1. VISUALIZACIÓN DE MENSAJES FIJOS Y EN MOVIMIENTO

Muchos proyectos requieren visualizar mensajes más o menos largos en la pantalla de un LCD. La librería LCD_MENS.INC describe dos subrutinas para realizar esta tarea de forma muy sencilla:

- Subrutina "**LCD Mensaje**" que visualiza mensajes fijos
- Subrutina "**LCD_Movimiento**" que visualiza mensajes en movimiento.

```
;************************** Librería "LCD_MENS.INC" ******************************
;
;       ======================================================================
;         Del libro "MICROCONTROLADOR PIC16F84. DESARROLLO DE PROYECTOS"
;         E. Palacios, F. Remiro y L. López.
;         Editorial Ra-Ma.  www.ra-ma.es
;       ======================================================================
;
; Librería de subrutinas para el manejo de mensajes a visualizar en un visualizador LCD.

        CBLOCK
        LCD_ApuntaCaracter              ; Indica la posición del carácter a visualizar
                                        ; respecto del comienzo de todos los mensajes,
                                        ; (posición de la etiqueta "Mensajes").
        LCD_ValorCaracter                       ; Código ASCII del carácter a
        ENDC                            ; visualizar.

; Los mensajes tienen que estar situados dentro de las 256 primeras posiciones de la
; memoria de programa, es decir, no pueden superar la dirección 0FFh.

; Subrutina "LCD_Mensaje" ---------------------------------------------------------
;
; Visualiza por pantalla el mensaje apuntado por el registro W.
;
; Los mensajes deben localizarse dentro de una zona encabezada por la etiqueta "Mensajes" y que
; tenga la siguiente estructura:
;
; Mensajes                              ; ¡Etiqueta obligatoria!
;       addwf   PCL,F
; Mensaje0                              ; Posición inicial del mensaje.
;       DT ".. ..", 0x00                ; Mensaje terminado en 0x00.
; Mensaje1
;       ...
;       ...
; FinMensajes
;
; La llamada a esta subrutina se realizará siguiendo este ejemplo:
;
;       movlw   Mensaje0                ; Carga la posición del mensaje.
;       call    LCD_Mensaje             ; Visualiza el mensaje.
;
LCD_Mensaje
        movwf   LCD_ApuntaCaracter      ; Posición del primer carácter del mensaje.
        movlw   Mensajes                ; Halla la posición relativa del primer carácter
        subwf   LCD_ApuntaCaracter,F    ; del mensaje respecto de etiqueta "Mensajes".
        decf    LCD_ApuntaCaracter,F    ; Compensa la posición que ocupa "addwf PCL,F".
LCD_VisualizaOtroCaracter
        movf    LCD_ApuntaCaracter,W
        call    Mensajes                ; Obtiene el código ASCII del carácter apuntado.
        movwf   LCD_ValorCaracter               ; Guarda el valor de carácter.
        movf    LCD_ValorCaracter,F     ; Lo único que hace es posicionar flag Z. En caso
        btfsc   STATUS,Z        ; que sea "0x00", que es código indicador final
        goto    LCD_FinMensaje          ; de mensaje, sale fuera.
LCD_NoUltimoCaracter
        call    LCD_Caracter            ; Visualiza el carácter ASCII leído.
        incf    LCD_ApuntaCaracter,F    ; Apunta a la posición del siguiente carácter
        goto    LCD_VisualizaOtroCaracter; dentro del mensaje.
LCD_FinMensaje
        return                          ; Vuelve al programa principal.
```

```
; Subrutina "LCD_MensajeMovimiento" ----------------------------------------------------
;
; Visualiza un mensaje de mayor longitud que los 16 caracteres que pueden representarse
; en una línea, por tanto se desplaza a través de la pantalla.
;
; En el mensaje debe dejarse 16 espacios en blanco, al principio y al final para
; conseguir que el desplazamiento del mensaje sea lo más legible posible.
;
        CBLOCK
        LCD_CursorPosicion              ; Contabiliza la posición del cursor dentro de la
        ENDC                            ; pantalla LCD

LCD_MensajeMovimiento
        movwf   LCD_ApuntaCaracter      ; Posición del primer carácter del mensaje.
        movlw   Mensajes                ; Halla la posición relativa del primer carácter
        subwf   LCD_ApuntaCaracter,F    ; del mensaje respecto de la etiqueta "Mensajes".
        decf    LCD_ApuntaCaracter,F    ; Compensa la posición que ocupa "addwf PCL,F".
LCD_PrimeraPosicion
        clrf    LCD_CursorPosicion      ; El cursor en la posición 0 de la línea.
        call    LCD_Borra               ; Se sitúa en la primera posición de la línea 1 y
LCD_VisualizaCaracter                   ; borra la pantalla.
        movlw   LCD_CaracteresPorLinea  ; ¿Ha llegado a final de línea?
        subwf   LCD_CursorPosicion,W
        btfss   STATUS,Z
        goto    LCD_NoEsFinalLinea
LCD_EsFinalLinea
        call    Retardo_200ms           ; Lo mantiene visualizado durante este tiempo.
        call    Retardo_200ms
        movlw   LCD_CaracteresPorLinea-1; Apunta a la posición del segundo carácter visualizado
        subwf   LCD_ApuntaCaracter,F    ; en pantalla, que será el primero en la siguiente
        goto    LCD_PrimeraPosicion      ; visualización de línea, para producir el efecto
LCD_NoEsFinalLinea                      ; de desplazamiento hacia la izquierda.
        movf    LCD_ApuntaCaracter,W
        call    Mensajes                ; Obtiene el ASCII del carácter apuntado.
        movwf   LCD_ValorCaracter               ; Guarda el valor de carácter.
        movf    LCD_ValorCaracter,F     ; Lo único que hace es posicionar flag Z. En caso
        btfsc   STATUS,Z        ; que sea "0x00", que es código indicador final
        goto    LCD_FinMovimiento       ; de mensaje, sale fuera.
LCD_NoUltimoCaracter2
        call    LCD_Caracter            ; Visualiza el carácter ASCII leído.
        incf    LCD_CursorPosicion,F    ; Contabiliza el incremento de posición del
                                        ; cursor en la pantalla.
        incf    LCD_ApuntaCaracter,F    ; Apunta a la siguiente posición por visualizar.
        goto    LCD_VisualizaCaracter   ; Vuelve a visualizar el siguiente carácter
LCD_FinMovimiento                       ; de la línea.
        return                          ; Vuelve al programa principal.

;       ================================================================
;         Del libro "MICROCONTROLADOR PIC16F84. DESARROLLO DE PROYECTOS"
;         E. Palacios, F. Remiro y L. López.
;         Editorial Ra-Ma.  www.ra-ma.es
;       ================================================================
```

2. PROGRAMAS EJEMPLO y EJERCICIOS

EJEMPLO 1:

a) El programa **Mensaje_01.asm,** para el PIC16F877A, produce que en la pantalla del módulo LCD se visualiza un mensaje de menos de 16 caracteres grabado en la memoria ROM mediante la directiva DT.

b) Comprueba el funcionamiento del programa en la tarjeta EasyPIC.
c) Modifica el programa de modo que en la pantalla del módulo LCD se visualice tu nombre.

EJEMPLO 2:

a) El programa **Mensaje_02.asm,** realiza la misma tarea que el programa anterior pero utilizando la subrutina **LCD_mensaje** de la librería **LCD_MENS.INC**.

b) Comprueba el funcionamiento del programa en la tarjeta EasyPIC.
c) Modifica el programa de modo que en la pantalla del módulo LCD se visualice tu nombre.

EJEMPLO 3:

a) El programa **Mensaje_03.asm,** para el PIC16F877A, produce que en la pantalla del módulo LCD se visualicen varios mensajes, uno detrás de otro. Cada mensaje permanece visualizado durante 2 segundos (si Fosc=4Mhz, la mitad para Fosc=8MHz). Entre mensaje y mensaje la pantalla se mantiene apagada durante 200ms.

b) Comprueba el funcionamiento del programa en la tarjeta EasyPIC.

EJEMPLO 4:

a) El programa **Mensaje_03B.asm,** realiza la misma tarea que el programa anterior, pero resuelto de una forma más eficaz.

b) Comprueba el funcionamiento del programa en la tarjeta EasyPIC .

c) Modifica el programa de modo que en la pantalla del módulo LCD se visualicen tus propios mensajes

EJEMPLO 5:

a) El programa **Mensaje_07.asm,** para el PIC16F877A, produce que en la pantalla del módulo LCD se visualice un mensaje largo (de más de 16 caracteres) que se va desplazando a lo largo de la pantalla. Se utilizará la subrutina **LCD_MensajeMovimiento** de la librería **LCD_MENS.INC**.

b) Comprueba el funcionamiento del programa en la tarjeta EasyPIC .
c) Modifica el programa de modo que en la pantalla del módulo LCD se visualice la siguiente frase: *"Estudia Ingeniería Industrial en la UMH!!"*

EJEMPLO 6:

a) El programa **Mensaje_08.asm,** para el PIC16F877A, es un programa para el juego de la Quiniela: Al presionar sobre el pulsador conectado al pin RA4 en la pantalla aparecerá rápidamente "1", "X", "2". Cuando suelta el pulsador, permanece el signo seleccionado

b) Comprueba el funcionamiento del programa en la tarjeta EasyPIC .

EJEMPLO 7:

a) El programa **Mensaje_09.asm,** para el PIC16F877A, produce que en la pantalla del módulo LCD se visualice CERRADO O ABIERTO según si un pulsador está presionado o no.

b) Comprueba el funcionamiento del programa en la tarjeta EasyPIC .

EJEMPLO 8:

a) Escribe un programa que realice la siguiente tarea:
Cada vez que se presione alguno de los pulsadores conectado a **PORTB** se visualizará un mensaje diferente de más de 16 caracteres que se irá desplazando a lo largo de la pantalla.

b) Comprueba el funcionamiento del programa en la tarjeta EasyPIC