

Práctica 3

Teclado y pantalla LCD

1. Objetivos

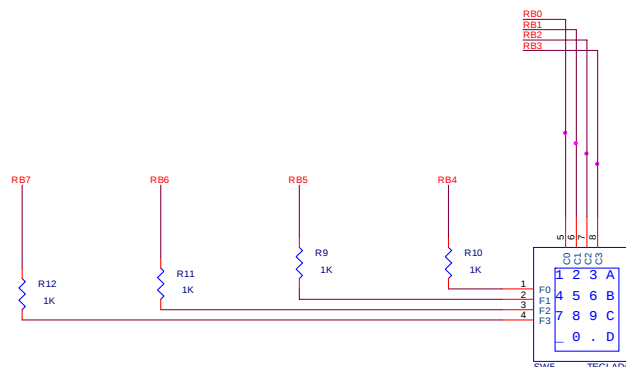
El objetivo de esta sesión es que el alumno profundice en los elementos periféricos de todo sistema empustrado. Para ello se estudiará el teclado matricial de 4x4 teclas y la pantalla LCD de 2x16 caracteres. Así mismo se aumentará la complejidad de los programas al tener que implementar un pequeño menú de opciones.

2. Introducción

En esta sesión se introducen dos elementos periféricos nuevos como son el teclado matricial y la pantalla LCD. Con el teclado matricial dispondremos de 16 teclas mientras que la pantalla LCD nos ofrece una potente herramienta de salida al poder mostrar texto en dos líneas de 16 caracteres cada una. El funcionamiento de cada elemento se muestra a continuación.

2.1 Teclado matricial de 4x4 teclas

Los teclados matriciales son muy utilizados en sistemas empustrados, ya que ofrecen el control de varias teclas con pocas señales; en concreto, con n señales se pueden controlar hasta 2^n teclas. Hoy se utilizará el teclado matricial de 4x4 teclas (16) que se encuentra en la tarjeta EduMIC y cuyo circuito se muestra en la siguiente figura:



Un teclado matricial está compuesto por filas y columnas de cables, de manera que al pulsar una tecla se pone en contacto una fila con una columna. En la figura anterior es posible ver que las columnas del teclado matricial están conectadas a los 4 bits bajos del puerto B, mientras que las filas están conectadas a los 4 bits altos. Para detectar la pulsación de una tecla es muy sencillo: en las filas escribiremos un cero al tiempo que leemos las columnas, si hay alguna tecla pulsada en la columna correspondiente se leerá un cero, mientras que en las columnas que no se pulse se leerá un uno (siempre y cuando se tengan resistencias de *pull-up*). Con esto ya sabríamos en qué columna se ha pulsado una tecla. Esta misma operación se repite cambiando filas por columnas de manera que obtendría el valor de la fila que contiene la tecla pulsada. Con la fila y la columna ya se puede saber de qué tecla se trata. El algoritmo de lectura de tecla sería el siguiente:

1. Las 4 líneas más bajas de PORTB son las columnas y los 4 bits más altos las filas de teclado.
2. Activar las resistencias de puesta a uno del puerto B (*pull-ups*), esto se puede realizar poniendo a cero el bit 7 (#RBPU) del registro INTCON2.
3. Programar los cuatro bits bajos de B como salida y los altos como entrada. A continuación escribir un cero en PORTB.
4. Leer los bits bajos del puerto B, el bit que sea '0' corresponde a la fila donde se encuentra la tecla que esté siendo pulsada. Si todos son '1' no se está pulsando ninguna tecla.

- Para averiguar las filas se repite desde el paso 2 pero cambiando parte alta por baja, obteniendo la columna.

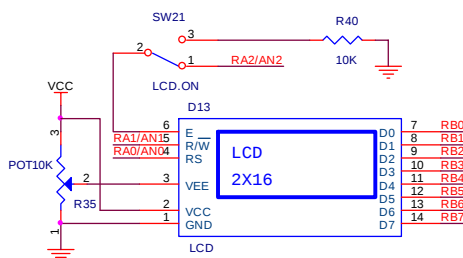
Los pasos anteriores sirven para leer la pulsación de una tecla que se esté pulsando, pero este algoritmo sin más no sirve para implementar la lectura de varias pulsaciones que se estén realizando. Por ejemplo, si metemos la rutina anterior en un bucle repetitivo se leerá miles de veces la misma pulsación de tecla y no sabremos si es que realmente se ha pulsado mil veces o es que hemos mantenido pulsada la tecla durante un tiempo. Por lo tanto, antes de leer una nueva tecla es necesario detectar que se ha soltado la tecla que estaba pulsada.

Por otro lado, los interruptores del teclado son mecanismos imperfectos, ya que al pulsar hay un periodo en el cual se está produciendo el contacto mecánico y la conexión está indefinida. Durante este tiempo, que puede durar unos milisegundos, se producen muchos rebotes que equivalen a pulsar muy rápidamente la tecla. Para evitarlo deberemos dejar suficiente espacio de tiempo entre una lectura de tecla y otra. Estos detalles se resumen en el siguiente algoritmo que sirve para leer una serie de teclas:

- Leer tecla (según algoritmo anterior).
- Si lo que se ha leído corresponde con una tecla la apuntamos y seguimos, sino volvemos a 1.
- Esperamos un tiempo para evitar rebotes, por ejemplo, 10 ms utilizando las funciones de retraso.
- Lee tecla.
- Si lo que se lee sigue siendo la tecla anterior volvemos a 4.
- Volver a empezar en 1.

2.2 El display LCD

El esquema correspondiente al circuito de este display se encuentra en la siguiente figura:



Los displays LCD son mayoritariamente estándar y se controlan de formas muy parecidas, incluso cuando no coincide el número de caracteres. En la figura podemos ver los elementos básicos de un display LCD estándar. Por un lado se tiene el bus de datos D0..D7 que en este caso está conectado al puerto B del PIC. Este bus de datos puede ser de entrada al LCD (para escribir caracteres y enviar instrucciones) o puede ser de salida del LCD (para poder leer el estado por ejemplo). El LCD tiene tres señales de control: E es la de habilitación y está conectada al bit 2 del puerto A del PIC, sirve para habilitar el LCD y como vemos está conectada a un interruptor para deshabilitar permanentemente el LCD; la señal R/W sirve para indicar operación de lectura o escritura; por último se la señal RS es la de sincronismo de datos e instrucciones y está conectada al bit 1 del puerto A. Además el LCD tiene señales para alimentación y una señal (VEE) que sirve para controlar el contraste de la pantalla.

Para simplificar las operaciones de escritura y control en el LCD, se provee un fichero (lcd.h) disponible en la página web de la asignatura. Este fichero contiene varias rutinas muy útiles entre las que destacan las siguientes:

- **LCD_Prepara()** Esta rutina prepara los puertos A y B de forma adecuada y pone las señales de control en un estado de preparado.
- **LCD_Ini()** Esta rutina envía al LCD la secuencia de inicialización. Esta rutina es necesaria al encender el LCD y debe venir siempre después de la rutina anterior.
- **LCD_Comando(char)** Esta rutina permite enviar cualquier comando o instrucción al LCD. Hay numerosas instrucciones, todas ellas se pueden consultar en el manual del LCD que se puede encontrar en la página web de la asignatura. Por ejemplo:
 - o **LCD_Comando(0b00111100)** indica que el bus de datos es de 8 bits, que se van a utilizar las 2 líneas del LCD con una fuente 5x10.

- o `LCD_Comando(0b00001110)` enciende el LCD para que se vea, enciende el cursor y le dice que no parpadee.
- o `LCD_Comando(0b10000000 + pos)` mueve el cursor hasta una posición determinada por el parámetro `pos`; // `Pos = 0x40` para el principio de la segunda fila.
- o `LCD_Comando(0b00000001)` hace un reset del LCD borrando su contenido y posiciona el cursor en la posición `0x00`.
- `LCD_Dato(char)` Si todo se ha iniciado bien, esta instrucción muestra por pantalla el carácter que se le pase en la posición actual del cursor.

En principio basta con estas cuatro funciones para operar con el LCD. Es bastante aconsejable tener cerca el manual del LCD para ver los comandos de control, ya que son muy útiles. Estos comandos permiten apagar y encender la pantalla, mostrar el cursor, llevar el cursor a una determinada posición, limpiar la pantalla, etc.

Es importante que los conmutadores de la placa EduMIC que sirven para elegir Analógico o Digital estén en su posición central (LIBRE) de esta manera los periféricos no interfieren con la operación sobre los bits RA0, RA1 y RA2 que son los que utiliza el LCD.

3. Trabajos a realizar

En todas las sesiones de laboratorio se pide la realización de unos programas o tareas. **Es imprescindible que se tengan preparados, o al menos esbozados, los programas al principio de cada sesión.** Al inicio de la sesión el profesor comprueba de que efectivamente los programas han sido preparados y se pone la nota correspondiente. Tener los programas preparados al inicio de la sesión es muy importante, de lo contrario, es muy difícil que en una misma sesión se puedan terminar los programas habiendo probado su funcionamiento.

Programa: control del teclado y LCD

Se propone realizar un programa que escriba en el LCD todas las teclas pulsadas en el teclado. Para poder trabajar con el LCD recordar que primero es necesario inicializarlo en 3 pasos:

Primero antes de realizar cualquier operación sobre el LCD hay que configurar los puertos con:

`LCD_Prepara()`

Después hay que llamar a la función de inicialización del LCD:

`LCD_Ini()`

Por último que hay que llamar a la función de configuración (`LCD_Comando`) dos veces con los siguientes valores:

`LCD_Comando(0b00111100)`

`LCD_Comando(0b00001110)`

También hay que tener en cuenta que los caracteres enviados al LCD se deben escribir mediante la función `LCD_Dato(char)` que espera que el valor pasado como parámetro sea el correspondiente valor ASCII del carácter a escribir. Antes de llamar a esta función también es necesario configurar los puertos con la función `LCD_Prepara()`. Además el programa debe cambiar de línea automáticamente cuando se complete la primera, y debe limpiar el LCD y volver a la primera posición de la primera línea cuando se llene también la segunda línea. Para ello recordar que el LCD tiene 16 posiciones en cada línea y que para situarse al principio de la segunda línea hay que mover el cursor hasta la posición `0x40`.

Recordar que antes de programar el microcontrolador con el software de programación EduMIC, la palabra de configuración `config1` hay que modificarla para que su valor sea **`config1= '0C00'`**.

Apartado opcional: Control de acceso

Se pide realizar un programa para construir un sistema de control de acceso mediante clave. Inicialmente el programa envía un mensaje que invita a introducir una clave de 4 cifras. Una vez introducida, el programa comprueba si es correcta y muestra en el LCD el resultado. Transcurridos dos segundos, vuelve al inicio y solicita la clave de nuevo.