Reporte 1

Universidad de Guanajuato

Campus Irapuato-Salamanca

División de Ingenierías

Materia:

Informática Industrial

Nombre reporte:

Funciones menú para LCD

Alumna: María Guadalupe Espinosa Sánchez

Profesor: M.I. David Balcázar Torres

Librerías para Menu

#include <LiquidCrystal.h>

Esta biblioteca permite a una placa Arduino controlar las pantallas LiquidCrystal (LCD). Permite crear un objeto que representa al display LCD y que contiene todas las operaciones “de bajo nivel” para que nos resulte fácil la programación de este dispositivo. La pantalla se puede controlar por medio de 4 u 8 líneas de datos.

Esta librería contiene diferentes propiedades las cuales ayudaran a la realización del menú se muestran a continuación*:

begin()

Descripción:

Inicializa la interfaz de la pantalla LCD, y especifica las dimensiones (anchura y altura) de la pantalla. begin () debe ser llamado antes de cualquier otro comando de la biblioteca LCD.

clear()

Descripción:

Borra la pantalla LCD y posiciona el cursor en la esquina superior izquierda.

home()

Descripción:

Pone el cursor en la esquina superior izquierda de la pantalla LCD. Es decir, utiliza esa ubicación en la salida de texto subsiguiente a la pantalla. Para borrar también la pantalla, utilice la función clear() en su lugar.

setCursor()

Descripción;

Coloca el cursor del LCD; es decir, establece la ubicación en la que se mostrará el texto escrit a continuación en la LCD.

write()

Descripción:

Escribe un carácter en la pantalla LCD.

print()

Descripción:

Imprime un texto en la LCD

cursor() y noCursor()

Descripción:

cursor(); Muestra el cursor del LCD: como un guión bajo (línea) en la posición a la que se escribirá el siguiente carácter.

noCursor(): Muestra el cursor de forma normal.

blink() y noBlink()

Descripción:

blink(): Muestra el cursor LCD parpadeante.

noBlink(): Muestra el cursor fijo.

Si se utiliza en combinación con la función cursor (), el resultado dependerá de la pantalla en particular.

display() y noDisplay()

Descripción:

display(): Enciende la pantalla LCD, después de haber sido apagada con noDisplay().

noDisplay(): Apaga la pantalla LCD, después de haber sido encendida con display().

Esto, en ambos casos, restaurará el texto (y el cursor) que estaba en la pantalla.

scrollDisplayLeft() y scrollDisplayRight()

Descripción:

scrollDisplayLeft(): Desplaza el contenido de la pantalla (texto y el cursor) un espacio hacia la izquierda.

scrollDisplayRight(): Desplaza el contenido de la pantalla (texto y el cursor) un espacio hacia la derecha

autoscroll() y noAutoscroll()

Descripción:

autoscroll(): Activa el desplazamiento automático de la pantalla LCD. Esto hace que cada salida de caracteres de la pantalla empuje los caracteres anteriores por un espacio. Si la dirección del texto actual es de izquierda a derecha (el valor predeterminado), la pantalla se desplaza hacia la izquierda; si la dirección actual es de derecha a izquierda, la pantalla se desplaza a la derecha. Esto tiene el efecto de la salida de cada nuevo carácter en la misma ubicación en la pantalla LCD.

noAutoscroll(): Desactiva el desplazamiento automático de la pantalla LCD.

leftToRight() y rightToLeft()

Descripción:

leftToRight(): Establece la dirección del texto escrito para la pantalla LCD de izquierda a derecha, el valor predeterminado. Esto significa que los caracteres escritos posteriores en la pantalla irán de izquierda a derecha, pero no afecta el texto previamente escrito.

rightToLeft(): Establece la dirección del texto escrito para la pantalla LCD de derecha a izquierda (por defecto es de izquierda a derecha. Esto significa que los caracteres escritos posteriores en la pantalla irán de derecha a izquierda, pero no afecta el texto previamente escrito.

createChar()

Descripción:

Crea un carácter personalizado (gylph) para su uso en la pantalla LCD. Hasta ocho caracteres de 5x8 píxeles son compatibles (numerados de 0 a 7). La apariencia de cada carácter se especifica mediante una serie de ocho bytes, uno para cada fila. Los cinco bits menos significativos de cada byte determinan los píxeles de la fila. Para mostrar un carácter personalizado en la pantalla, write() su número.

NB: Cuando se hace referencia al carácter personalizado "0", si no es en una variable, necesita para su emisión como un byte, de lo contrario el compilador genera un error.

#include <EEPROM.h>

Las plaquitas de Arduino cuentan con una zona de memoria de un tipo conocido como EEPROM (Electrically Erasable Read Only Memory, Memoria de Sólo Lectura, Borrable Eléctricamente). Aunque el nombre pueda no parecer muy afortunado, se trata de una memoria no volátil, es decir, cuyo contenido no se pierde cuando se desconecta la alimentación. Sin embargo, se puede borrar y reescribir mediante las adecuadas señales eléctricas. Podemos considerar estas memorias como diminutísimos discos duros (la capacidad varía de unas versiones de Arduino a otras, pero oscila entre 1 y 4 Kb).

Esta librería contiene diferentes propiedades las cuales ayudaran a la realización del menú y se muestran a continuación *:

LA PROPIEDAD EEPROM []

Es una matriz que se refiere a las celdas (unidades donde se almacena la información; en la práctica, cada celda contiene un byte) de la EEPROM de nuestra plaquita. Se trata de una matriz que puede ser leída y/o escrita,

apuntando, con el índice, a la celda en la que queremos escribir, o de la que queremos leer.

EL MÉTODO read()

Se usa como forma de leer una posición de memoria de la EEPROM. La sintaxis general obedece a EEPROM.read(posicion);. Observa que, como esta clase no se instancia en objeto, referenciamos el método (lo haremos así con todos) directamente a través del nombre de la clase.

EL MÉTODO write()

Este es complementario del anterior. Nos permite escribir un byte específico en una posición de la memoria EEPROM. La sintaxis general obedece al esquema EEPROM.write(posición, valor);.

EL MÉTODO update()

Su funcionamiento y uso es el mismo que el anterior. La diferencia es que este método sólo escribe el valor en la celda indicada, si el que hay es diferente, no si ya es igual. La sintaxis es la misma que la del método anterior.

EL MÉTODO put()

Este método se emplea cuando se hace necesario escribir en la memoria datos que exceden del límite de un solo carácter.

EL MÉTODO get()

Este es complementario del anterior. Se usa para leer un dato de la EEPROM, de cualquier tamaño. Especificamos la dirección de inicio, y nombre de la variable donde se almacenará la lectura. De este modo, según el tipo de variable, el método “sabe” cuantas posiciones de memoria consecutivas debe leer.

*No todas serán utilizadas para el menú