

Sistemas Basados en Microprocesador

Bloque 2 – Práctica 4

Interfaces de usuario: Display LCD

Parte II

Introducción y objetivos de la práctica.

Una vez conocidos los procedimientos de inicialización y configuración del LCD en la práctica 3, el objetivo de esta práctica es completar su gestión mediante el desarrollo de distintas funciones que permitan escribir mensajes. Es necesario conocer y recordar la disposición de la memoria gráfica del LCD y los comandos necesarios para escribir y modificar la visualización de uno o varios grupos de píxeles.

Como en prácticas anteriores se busca obtener una serie de ficheros (*.c, *.h) que permitan el uso del LCD en prácticas posteriores. Por ello es de suma importancia que todo el código desarrollado esté lo suficientemente documentado y verificado.

Documentación

- mbed App Board documentation and schematics:
- <https://os.mbed.com/components/mbed-Application-Board/>
- https://os.mbed.com/media/uploads/chris/mbed-014.1_b.pdf
- Display datasheet <https://www.newhavendisplay.com/specs/NHD-C12832A1Z-FSW-FBW-3V3.pdf>
- Display controller <http://www.lcd-module.de/eng/pdf/zubehoer/st7565r.pdf>
- CMSIS driver specification
<http://www.keil.com/pack/doc/CMSIS/Driver/html/index.html>
- CMSIS SPI Interface
http://www.keil.com/pack/doc/CMSIS/Driver/html/group_spi_interface_gr.html
- Aplicación GLCD Font Creator (Mikroelektronika: <https://www.mikroe.com/supporting-software>).
- Aplicación FontEditor <https://www.min.at/prinz/o/software/pixelfont/>
- Fichero de fuente en Moodle.

Ejemplo 1.

Cree un proyecto (P4_1) en Keil uVision para la tarjeta de desarrollo, utilizando CMSIS, basado en el proyecto de la práctica anterior P3_4. Debe añadir el siguiente código e incluir en su proyecto el fichero Arial12x12.h que podrá encontrar en Moodle. Además incluya al final de la función main() una llamada a la función **LCD_symbolToLocalBuffer_L1(uint8_t symbol)** (antes de copiar los datos a la memoria del LCD), utilizando como parámetro para dicha función el carácter 'H'.

```

void symbolToLocalBuffer_L1(uint8_t symbol){

    uint8_t i, value1, value2;
    uint16_t offset=0;

    offset=25*(symbol - ' ');

    for (i=0; i<12; i++){

        value1=Arial12x12[offset+i*2+1];
        value2=Arial12x12[offset+i*2+2];

        buffer[i/*+positionL1*/]=value1;
        buffer[i+128/*+positionL1*/]=value2;
    }
    //positionL1=positionL1+Arial12x12[offset];
}

```

Una vez visualice el carácter 'H' responda a las siguientes preguntas:

- Teniendo en cuenta la información existente en el fichero Arial12x12.h, ¿Cuántos bytes se utilizan para conformar un carácter en este tipo de letra?
- ¿En qué posición se ha escrito el carácter? Podría escribirse en otro lugar en el LCD, piense cómo sería el procedimiento.
- Utilizando este tipo de letra, ¿Cuántas líneas de texto se podrían visualizar en todo el LCD?
- En el fichero del tipo de letra, el primer byte de cada carácter, no contiene información del valor de los píxeles que constituyen la letra. Haga una rápida inspección de algunas líneas e identifique la utilidad de este primer dato.

Ejemplo 2.

Cree un proyecto (P4_2) basado en el anterior que escriba una línea de texto en el LCD:

Prueba de texto L1

Para ello será necesario realizar llamadas, con cada uno de los caracteres de la cadena, a la función `LCD_symbolToLocalBuffer_L1`. Modifique dicha función para que la anchura ocupada por cada letra no sea fija y el texto se ajuste como lo haría en un editor de textos. Mejore también la copia de datos a la memoria del LCD, con el fin de que solo se actualicen los datos correspondientes a esta línea.

Ejemplo 3.

Tomando como base el resultado del ejercicio anterior, cree otra función denominada `LCD_symbolToLocalBuffer_L2(uint8_t symbol)`, que permita visualizar en la línea inferior del LCD el texto:

Prueba de texto L2

Por último y utilizando el resultado del ejemplo 2 y el ejemplo 3 realice una función que responda al siguiente prototipo y que permita escribir en ambas líneas del LCD:

```
void symbolToLocalBuffer(uint8_t line,uint8_t symbol)
```

Ejemplo 4.

Estudie el prototipo de la función de librería `sprintf()`. Utilizando esta función escriba en la línea 1 y 2 del LCD, el valor de dos variables, una de tipo entero (`uint32_t`) que vale 1234 y otra de tipo float que vale 3.14159, de la siguiente manera:

Prueba valor1: 1234
Prueba valor2: 3.14159

En este caso será necesario distribuir el código desarrollado para el proyecto en los ficheros: `main.c`, `lcd.h` y `lcd.c`. Los ficheros correspondientes al lcd contendrán la definición de las variables globales utilizadas y las funciones que permiten inicializar y configurar el lcd y las funciones que permiten escribir en las dos líneas. De esta forma será posible reutilizar el código en prácticas posteriores.

Ejemplo 5 (opcional).

Acceda a la web del fabricante Mikroelektronika y descargue el software GLCD Font Editor. Una vez instalado, es posible crear un Font desde cero, o utilizar y personalizar una fuente del sistema.

Cree una fuente de caracteres grandes (18-24 puntos), y desarrolle la función que permite usarla para escribir en el LCD una cadena de caracteres.