Escuela Técnica Superior de Ingeniería Informática

Ingeniería Informática. Ingeniería de Computadores.

MEMORIA PRÁCTICA II

Asignatura: Sistemas Empotrados y de Tiempo Real I
Profesor: Gabriel Jiménez Moreno

Alumno: Álvaro José Gullón Vega

Índice

1. Objetivos

En esta segunda práctica distinguimos dos objetivos:

1.1. Académico

Los objetivo académicos en este caso, constan de estudiar y entender el funcionamiento de los Displays LCD tradicionales que se usan con los microcontroladores, conocer los aspectos más básicos de la placa de desarrollo de la asignatura y entender las diferencias entre las diversas librerías que proporcionan los fabricantes para el manejo de los periféricos en los microcontroladores tipo Cortex.

1.2. Práctico

Para conseguir estos objetivos académicos, se han propuestos estos objetivos prácticos:

- Mandar texto a cualquier posición en un display tipo HD44780.
- Definir un nuevo dibujo en la memoria del Display y visualizarlo en pantalla
- Manejar los LEDs y el botón de la placa al igual que se hizo en la práctica 1. correspondientes a ese pin y conectadas a VCC o a GND.
- Sustituir en la librería del display LCD, que se proporciona en esta práctica, las llamadas a las funciones GPIO LL por funciones equivalentes GPIO HAL.

2. Introducción

Como hemos mencionado en los objetivos, esta segunda práctica nos acercará a conocer los Display LCD al igual que nos ofrecerá una primera toma de contacto con la placa que usaremos en la asignatura.

Las interfaces visuales más primitivas están basadas en LEDs, un ejemplo de ellos los displays de siete segmentos, los que si agrupamos varios de ellos podemos componer una interfaz visual más completa con un problema que es la cantidad de pines que se requieren para su control. Por otra parte contamos con los displays LCD u OLED, dos tecnologías que han evolucionado mucho los últimos años con las que se construyen un gran número de dispositivos de visualización destinados a su uso con microcontroladores.

Nosotros, en SETR1, nos vamos a centrar en el uso de displays más modestos y primitivos que suelen disponer de dos tipos de interfaces:

- **Paralela:** Consiste en un bus de ocho bits *(D0-D7)* y tres líneas de control *(RS, E, R/W)*. El bus de datos es bidireccional de forma que podemos escribir en el display o podemos leer de él como por el ejemplo el bit de estado.

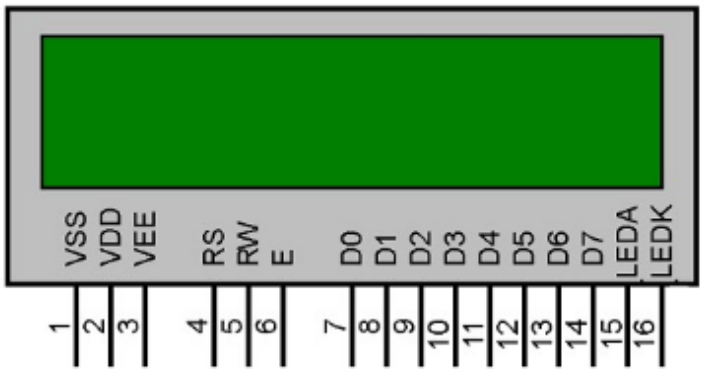

Figura 1: Pinout típico del Display LCD.

- **Serie síncrona:** Usan normalmente puertos *I2C* o *SPI*. Se utilizan dos o tres líneas de conexión con el microcontrolador.

El circuito de control es el cerebro del display, por un lado tiene los pines que sirven de interfaz con el usuario y por otra controla la pantalla de visualización. Casi siempre suelen tener una memoria RAM, excepto los más sofisticados que incluyen de dos en adelante para por ejemplo en la GameBoy organizarlas como memoria principal, memoria de background, memoria de tejas, etc...

La pantalla donde se muestra la información en ocasiones su tamaño es independiente de la capacidad de direccionamiento de la memoria del controlador, y no tienen por qué coincidir. Otro aspecto a tener en cuenta es la resolución de cada carácter, el nuestro tiene 5 columnas por 8 filas de pixel, con 2x16 caracteres y con una anchura total de 6,5 cm.

Para usar el HD44780 se realiza una conexión pin a pin con los GPIO del microcontrolador. Se suele colocar un potenciómetro que lo hace es regular en contraste de la pantalla. Para transferir datos, podemos usar *D0-D7* o *D4-D7*, en este último caso necesitamos dos accesos, primero 4 bits y después los otros 4 bits restantes. La librería software que vamos a usar en esta práctica para el manejo del HD44780 utiliza la interfaz de 4 bits y esperas entre transferencias, sin preguntar por el estado del dispositivo (si está Busy), y por tanto sin utilizar la línea R/W, sólo se hacen escrituras al LCD. Hay muchas librerías en internet para estos LCD y para múltiples plataformas y microcontroladores, ésta que vamos a usar es una de tantas, sin ninguna característica especial.

El software para usar el display HD44780 es muy simple, primero hay una fase de inicialización (función *lcd_reset()* + función *lcd_display_settings(1,0,0)* en la que se configura una serie de características, las más importante son si se va a usar el bus de conexión completo o sólo la parte alta (D0-D7 ó D4-D7) y si la pantalla es de una o dos líneas. En nuestro caso sabremos si la inicialización ha ido bien cuando observemos que se "ven" activas las dos líneas de la pantalla, y no sólo una.

En el HD44780 se pueden hacer las siguientes operaciones:

- **Escritura (función lcd_write(byte, rs):** se transfiere un byte y se indica si es un dato a la memoria o un comando (parámetro rs).
- **Apuntar CGRAM (lcd_cgram_address(uint8_t address)):** cuando se usa esta función se manda un comando que sitúa el puntero de dirección en la CGRAM y apuntando a una determinada dirección.

- **Apuntar a la DDRAM (lcd_display_address(uint8_t address)):** si queremos que vuelvan a visualizarse los datos en la pantalla tenemos que hacer que el apuntador vuelva a apuntar a una dirección de la DDRAM.
- **moveToXY(unsigned char row, unsigned char column):** con ella situamos el cursor en la fila/columna de la pantalla que queramos, como es natural hay que usar esa función primero y después mandar el texto que se quiera ver en la pantalla.
- **lcd_print(char string[]):** sirve para visualizar una cadena de caracteres.
- **writeIntegerToLCD(int integer): se utiliza para mostar un entero.**
- **lcd_clock():** es la encargada de subir y bajar la línea de habilitación E.
- **lcd_clear():** nos permite limpiar el contenido de la pantalla de nuestro display.

En microcontroladores complejos, como los ARM Cortex, el uso de librerías es esencial para desarrollar rápidamente. Aunque el núcleo ARM está estandarizado por la empresa ARM, los periféricos y el hardware adicional varían según el fabricante. ARM proporciona una interfaz estándar llamada CMSIS, útil pero limitada, ya que no incluye funciones prácticas para manejar el hardware específico de cada fabricante.

Para facilitar el desarrollo, cada fabricante ofrece sus propias librerías. Por ejemplo, STMicroelectronics proporciona dos:

- **HAL (Hardware Abstraction Layer):** de alto nivel, fáciles de usar y portables, aunque menos eficientes.
- **LL (Low Layer):** de bajo nivel, más cercanas al hardware, optimizadas y potentes, pero más complejas.

Las HAL intentan ser universales dentro de cada fabricante y son comúnmente utilizadas por principiantes. En cambio, las LL ofrecen mayor control y rendimiento, ideales para desarrolladores con más experiencia. En esta práctica se estudiarán ambas aplicadas a los periféricos GPIO.

Los pines GPIO se agrupan en puertos de 16 bits (GPIOA, GPIOB, etc.). Cada pin se identifica por su puerto y número (por ejemplo, PA1 o PC2). Para manipular estos pines se utilizan registros de 32 bits y funciones específicas. Así, tenemos el siguiente grupo de funciones LL para manejar los GPIO:

```
1        LL_GPIO_IsOutputPinSet(GPIO_TypeDef *GPIOx, uint32_t PinMask)
            // lee uno o varios pines de un
2        puerto de salida
3        LL_GPIO_IsInputPinSet(GPIO_TypeDef *GPIOx, uint32_t PinMask) //
            lee uno o varios pines de un
4        puerto de entrada
5        LL_GPIO_SetOutputPin(GPIO_TypeDef *GPIOx, uint32_t PinMask) //
            pone a uno pin/pines
6        LL_GPIO_ResetOutputPin(GPIO_TypeDef *GPIOx, uint32_t PinMask)
            // pone a cero pin/pines
7        LL_GPIO_TogglePin(GPIO_TypeDef *GPIOx, uint32_t PinMask) //
            complementa pin/pines
```

```
1        LL_GPIO_ReadOutputPort(GPIO_TypeDef *GPIOx) // lee un puerto de
             salida entero
2        LL_GPIO_WriteOutputPort(GPIO_TypeDef *GPIOx, uint32_t
             PortValue) // escribe un puerto entero
```

Por último, vamos a tratar con la placa **B-L475E-IOT01A** con el microcontrolador **STM32L475**:

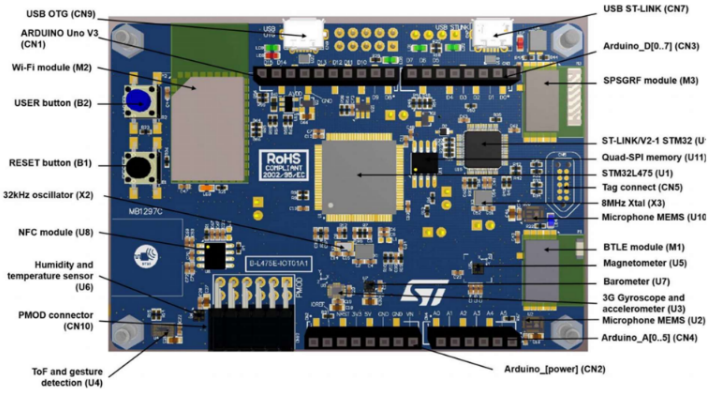

Figura 2: Placa **B-L475E-IOT01A** y elementos que la componen.

Los aspectos más destacables de esta placa que nos conviene conocer es que el depurador ST-LINK vía USB sobre la misma placa que nos permite tres tipos de dispostivos: depurador, unidad de almacenamiento ficticio "masivaz puerto virtual COM. También contiene un conector tipo Arduino.

3. Desarrollo de la práctica

3.1. Fase 1: Usar el display HD44780 y empezar a manejar la placa de la asignatura.

Lo primero que haremos será colocar el shield del display sobre la placa, teniendo en cuenta la orientación de esta para que los pines vayan en su sitio, para que sea más fácil, podemos retirar el .^{em}bellecedor"del botón azul para que no haga de escalón y no esté la placa puesta de aquella manera. Una vez colocada, conectaremos la placa a nuestro equipo usando el USB y conectándolo a la entrada *USB ST-LINK*

3.2. Fase 2: Repetir la Fase 2 y 4 de la práctica 1 con la nueva placa.

3.3. Fase 3: Repetir la fase 1 de esta práctica, pero usando librerías HAL.

4. Contestación de preguntas

5. Conclusiones