6-2-2022

Álvaro Fraidias y David Ros

780366 y 780584

Práctica 3

Diseño de una plataforma independiente

Contenido

[1. Resumen ejecutivo 2](#_Toc95079736)

[2. Introducción 2](#_Toc95079737)

[3. Objetivos 2](#_Toc95079738)

[4. Juego completo y pantalla LCD 2](#_Toc95079739)

[4.1 Pantalla bienvenida 3](#_Toc95079740)

[4.2 Pantalla tablero 4](#_Toc95079741)

[4.2.1 Método pintar\_tablero() 4](#_Toc95079742)

[4.2.2 Método pintar\_numeros\_tablero() y pintar\_candidatos() 5](#_Toc95079743)

[4.2.3 Método pintar\_información\_salir\_sudoku() 6](#_Toc95079744)

[4.2.3 Método pintar\_tiempo\_juego() y pintar\_tiempo\_calculos() 7](#_Toc95079745)

[4.3 Pantalla despedida 7](#_Toc95079746)

[4.4 Pantalla errores 7](#_Toc95079747)

[4.5 Cambios respecto práctica 2 8](#_Toc95079748)

[5. Plataforma autónoma 9](#_Toc95079749)

[6. Conclusiones 11](#_Toc95079750)



# Resumen ejecutivo

En este documento se desarrolla la última parte de las prácticas de la asignatura de Proyecto Hardware, donde se implementa el juego del sudoku en la pantalla LCD de la placa de forma que el usuario haciendo uso de los botones y del 8led puede ver por la pantalla LCD cómo se desarrolla el juego.

Por otro lado, se carga el código en la memoria flash de la placa para obtener una plataforma autónoma.

Los apartados que forman esta práctica son:

* Utilización de la pantalla LCD.
* Obtener una plataforma autónoma.

# Introducción

Esta práctica va a utilizar el código que ha sido diseñado en anteriores prácticas, pero en este caso se va a utilizar la pantalla LCD de la placa para que el usuario pueda visualizar el juego. Además, se obtendrá una plataforma autónoma, que sea independiente del ordenador del usuario, con la que se pueda jugar directamente.

# Objetivos

Los principales objetivos de esta práctica son:

* Conseguir una plataforma autónoma, con la que se pueda jugar directamente.
* Aprender a utilizar la pantalla LCD para visualizar el tablero.
* Aprender a cargar el código en la memoria flash de la placa.

# Juego completo y pantalla LCD

Para finalizar la implementación del juego sudoku en la placa Embest S3CEV40 vamos a diseñar el código necesario para utilizar la pantalla LCD y dotar así al sistema de una interfaz gráfica.

Para implementar dicho código debemos añadir al proyecto desarrollado en la práctica anterior (práctica 2) las clases “tp.c”, “tp.h”, “lcd.c” y “lcd.h”. En las clases “lcd.c” y “lcd.h” el profesor nos ha proporcionado todo el código necesario para controlar la pantalla lcd por lo que la implementación será mucho más sencilla.

A la hora de diseñar las diferentes pantallas que deberá tener el sistema hemos implementado varios métodos que se encargan de realizar dichas funcionalidades. Los métodos y pantallas son los siguientes:

## 4.1 Pantalla bienvenida

En esta primera pantalla simplemente se mostrarán las instrucciones básicas para poder jugar al sudoku. Para implementar dicha pantalla simplemente llamamos al método “pintar\_pantalla\_inicio()” donde, previamente se limpia la pantalla LCD con los métodos “Lcd\_Clr()” y “Lcd\_Active\_Clr()”. Una vez borrada la pantalla se escribe por pantalla la información que queremos mostrar con el método “Lcd\_DspAscII8x16()” cuyos parámetros son:

* Coordenada x de la pantalla LCD.
* Coordenada y de la pantalla LCD.
* Color a imprimir en la pantalla LCD.
* Texto a mostrar en la pantalla LCD.

Para finalizar debemos ejecutar el método “Lcd\_Dma\_Trans()” para que se muestre en la pantalla. De caso contario no se mostrará nada.

El método explicado anteriormente es el siguiente:

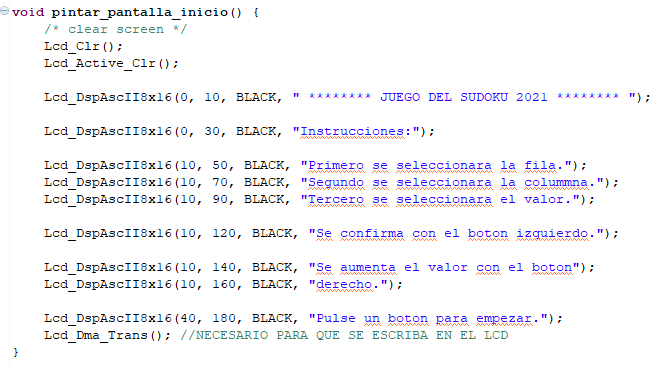


Ilustración 1: Método para pintar la pantalla principal.

## 4.2 Pantalla tablero

Para pintar el tablero debemos ejecutar los siguientes métodos:

* Pintar\_tablero(): pinta las líneas del sudoku.
* Pintar\_numeros\_tablero(): pinta los números y los posibles candidatos en el tablero.
* Pintar\_informacion\_salir\_sudoku(): muestra la información necesaria para poder salir del sudoku.
* Pintar\_tiempo\_juego(): muestra la cantidad de milisegundos transcurridos desde que el usuario empezó la partida.
* Pintar\_tiempo\_calculos(): pinta la cantidad de milisegundos requerida en los cálculos necesarios del sudoku (propagar valores, calcular candidatos…)

El método que ejecuta todos los métodos mencionados anteriormente es el siguiente:

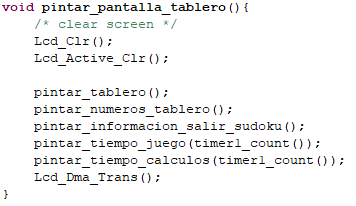


Ilustración 2: Método pintar\_pantalla\_tablero() de la clase lcd.c

### 4.2.1 Método pintar\_tablero()

Este método se encarga de pintar en la pantalla LCD las líneas del tablero del sudoku. Para pintar las líneas se ejecuta un bucle como se muestra a continuación:

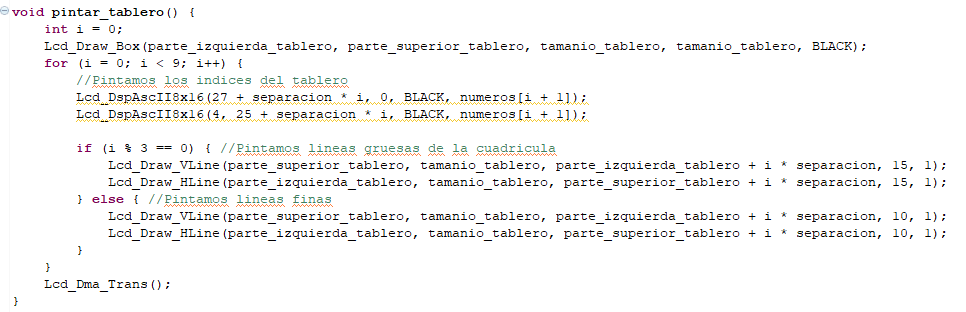


Ilustración : Método pintar\_tablero()

### 4.2.2 Método pintar\_numeros\_tablero() y pintar\_candidatos()

Igual que en el caso anterior debemos hacer dos bucles para recorrer el tablero. En cada iteración vemos el valor de cada celda. En caso de que haya un valor lo escribimos y en caso de que no haya un valor hacemos lo siguiente: obtenemos los candidatos de esa celda y desplazamos 7 posiciones a la izquierda dichos bits para quedarnos únicamente con los candidatos. Como puede haber 9 posibilidades debemos implementar un bucle que se ejecute 9 veces y en cada iteración comparamos mediante una AND lógica los candidatos con la variable auxiliar inicializada como “0x000000001”. En caso de que sea 0 será porque es un candidatos y debemos llamar al método pintar\_candidatos() que explicaremos más adelante.

El método explicado anteriormente es el siguiente:

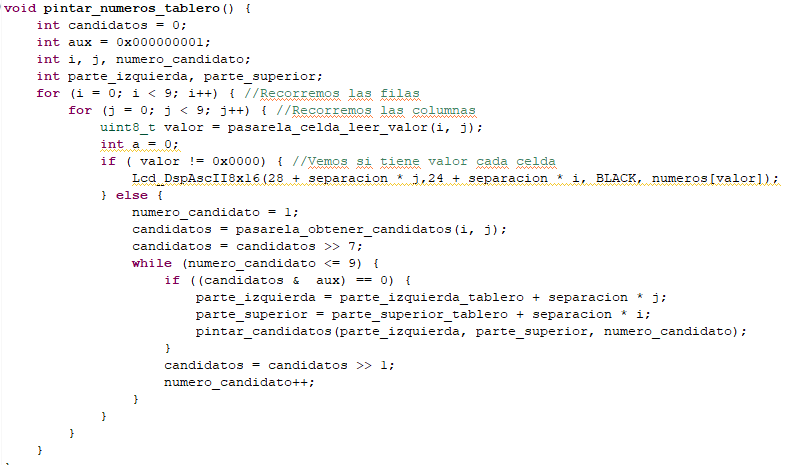


Ilustración : Método pintar\_numeros\_tablero()

El método pintar\_candidatos() obtiene como parámetros la posición x e y del tablero y el número candidato a pintar. Este método simplemente ejecuta un switch para escribir un punto negro en la posición del candidato.

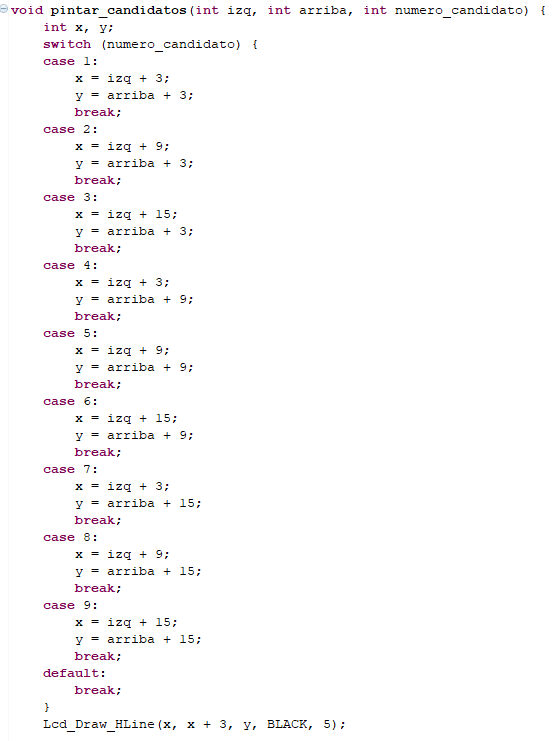


Ilustración : Método pintar\_candidatos()

### 4.2.3 Método pintar\_información\_salir\_sudoku()

Este método simplemente muestra en la pantalla del tablero lo que el usuario debe hacer si quiere salir del juego.

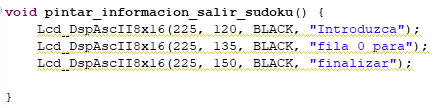


Ilustración : Método pintar\_informacion\_salir\_sudoku()

### 4.2.3 Método pintar\_tiempo\_juego() y pintar\_tiempo\_calculos()

Ambos métodos reciben como parámetro un entero que vendrá marcado por el timer 1.

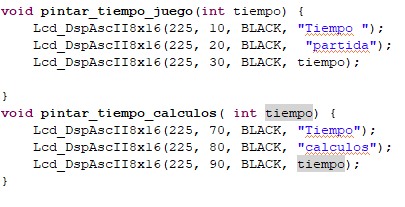


Ilustración : Métodos pintar\_tiempo\_juego() y pintar\_tiempo\_calculos()

## 4.3 Pantalla despedida

Cuando el usuario haya completado el sudoku o salga voluntariamente, se mostrará una pantalla de despedida. Para ello debemos llamar al siguiente método:

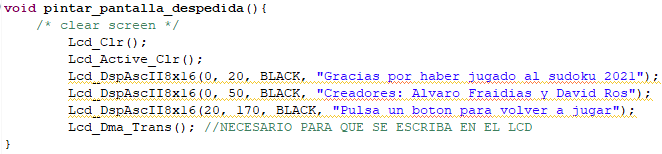


Ilustración : Método pintar\_pantalla\_despedida()

## 4.4 Pantalla errores

Cuando el usuario juegue al sudoku pueden ocurrir varios errores:

* Que el usuario quiera escribir en una celda que es pista.
* Que el usuario quiera escribir un número en una celda cuyos candidatos no coinciden con el número.

Para ello debemos mostrar por pantalla cuando el usuario está haciendo algo no permitido. Para ello ejecutamos los métodos “pintar\_error\_pista\_celda\_tablero()” y “pintar\_error\_candidato\_celda\_tablero()”:

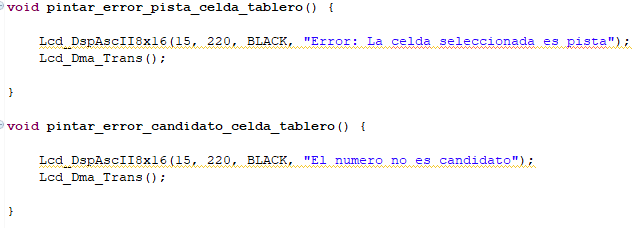


Ilustración : Métodos para pintar en pantalla errores.

Los métodos para pintar la pantalla LCD serán llamados desde la clase “máquina\_estados.c” en el método “maquina\_estados\_sudoku2021”.

## 4.5 Cambios respecto práctica 2

Hemos añadido nuevas funcionalidades en las celdas para ver si un valor es pista o dada una celda y un valor, ver si ese valor está en la lista de candidatos. Para ello hemos implementado estos métodos:

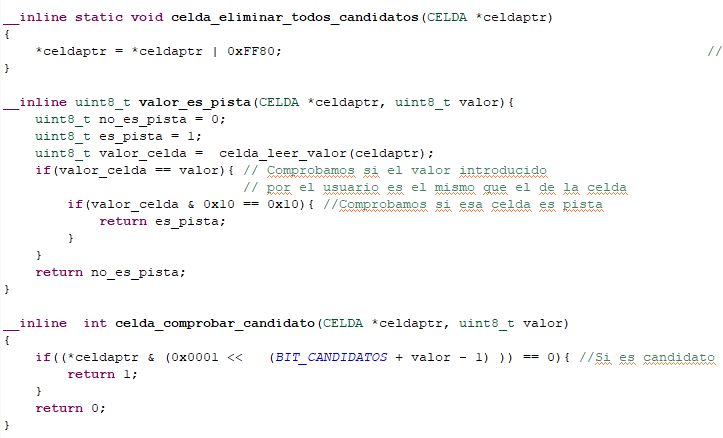


Ilustración : Método celda\_eliminar\_todos\_candidatos() y celda\_comprobar\_candidato().

# Plataforma autónoma

Para que la plataforma que utilizamos sea autónoma debemos cargar nuestro programa en la memoria flash. Antes de cargar nuestro código en la flash hay que realizar ciertas modificaciones para que funcionen las escrituras y para que las direcciones reales cuadren con las que utiliza el linker.

Las modificaciones realizadas han sido las siguientes:

* Cambiar el código del fichero “ld\_script.ld” por el proporcionado en el documento “P3-LEC”:

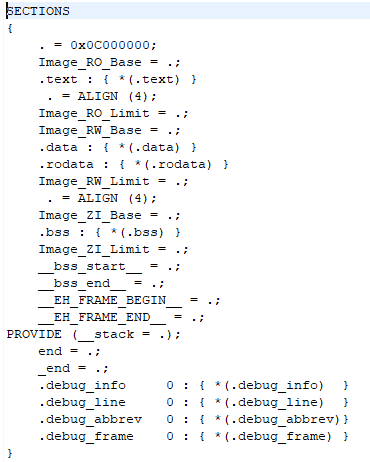


Ilustración . Código original fichero "ld\_script.ld"

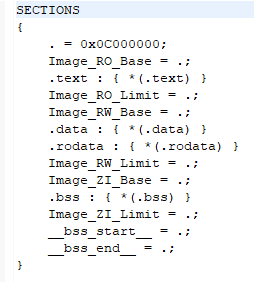


Ilustración . Código modificado fichero “ld\_script.ld”

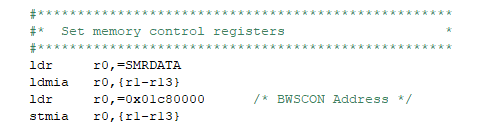
* Cambiar la inicialización del controlador de memoria: 

Ilustración . Inicialización original del controlador de memoria

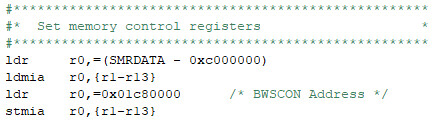


Ilustración . Inicialización modificada del controlador de memoria

* Modificar el código para que copie el contenido de la ROM a la memoria RAM al comienzo de la ejecución:

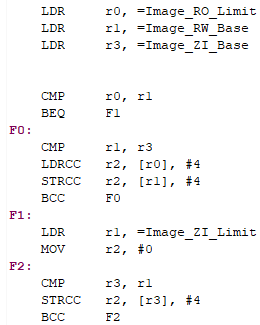


Ilustración . Código original

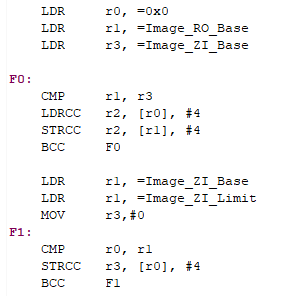


Ilustración . Código modificado

# Conclusiones

Con la práctica 3 finalizada hemos implementado al completo el sudoku con una interfaz gráfica que facilita el juego al usuario.

Gracias a esta práctica y a la asignatura completa de proyecto hardware hemos podido entender mejor cómo funcionan ciertos componentes fuera de la programación como tal, peleando con los componentes físicos y observando que la realidad es distinta a lo establecido.