

Proyecto No. 2: Videojuego Portable

Daniel Fuentes – 17083, Jorge Lorenzana – 17302

Departamento de Ingeniería Mecatrónica, Universidad del Valle de Guatemala

Guatemala, Guatemala

fue17083@uvg.edu.gt

lor17302@uvg.edu.gt

Abstract – En este artículo se detalla como se realizó un videojuego portable en la plataforma Tiva™C Series TM4C123G. Se muestra el análisis que se realizó para llevar a cabo el videojuego. Y se describe el procedimiento necesario para que sea un proyecto funcional. Este documento puede interpretarse como una guía para el desarrollo de una aplicación en la plataforma Tiva™C.

I. INTRODUCCIÓN

Este documento es el procedimiento que se llevó a cabo para realizar un videojuego portable. Cabe mencionar que es un proyecto que requiere de conocimientos de programación en lenguaje de C++ y de electrónica básica. El videojuego será un juego de ritmo donde el usuario tendrá que presionar un botón para la nota musical que corresponde. El videojuego tendrá gráficos de 8-bits al igual que la música. También contará con dos modos de juego, modo de un jugador y modo de dos jugadores.

II. OBJETIVOS

- Implementar el conocimiento adquirido a través del curso de Electrónica Digital 2.
- Implementar la pantalla ILI9341 para desplegar gráficos desde la plataforma.
- Implementar el módulo SD para el almacenamiento de archivos gráficos de gran tamaño.
- Incorporar sonido de 8 bits utilizando señales PWM.

III. PROCEDIMIENTO

A. Pantallas

Primero, se definieron las pantallas que el videojuego tendría. Esté constaría de un menú principal donde se pudiera escoger el modo de juego que se deseaba. Como segunda pantalla, se daría la opción de escoger un personaje, este no afectaría en la jugabilidad. La tercera pantalla sería el menú para escoger la canción que se quería jugar. Por último, la pantalla donde se estaría jugando.

B. Arte gráfico y auditivo

Se creó todo el arte que se estaría utilizando (fondos de pantalla, personajes, animaciones, etc.) y se convirtió en archivos de texto, utilizando el programa “lcd-image-converter”, a una lista de valores hexadecimales para que estos fueran interpretados por la Tiva™C. También se buscó la música que se utilizaría y se realizaron los arrays respectivos para cada canción.

C. Hardware

Teniendo en cuenta los aspectos anteriores, se decidió que utilizar como hardware del videojuego. Se optó por utilizar una pantalla ILI9341 con módulo SD incluido para desplegar y almacenar el arte gráfico, un buzzer pasivo para reproducir los sonidos de 8 bits y cuatro botones para realizar cada uno de los controles. Luego se realizaron los esquemáticos necesarios para las conexiones de los mismos.

D. Software

Definido esto, se crearon funciones para desplegar las imágenes que estarían contenidas en la SD. Para ello se conectó el módulo SD por medio de SPI, por medio de código se seleccionó el archivo de texto que se quería desplegar y a partir de sus caracteres se obtuvieron valores hexadecimales que son interpretados como colores de pixeles. Luego se programaron los *labels* de cada menú y su respectiva función. Para acceder a cada una de ellas se utilizarían botones que, haciendo uso de pines digitales de la tiva, iban a poder seleccionar entre las opciones disponibles en la pantalla. Por último, se programó una función que utilizaría los *arrays* de las notas y de la duración de cada una para reproducir los sonidos en el buzzer.

E. Jugabilidad

Desde el principio se tuvo definido como sería la jugabilidad, así que haciendo uso de los *arrays* de la música se programó la caída de las teclas en pantalla, las cuales guiarían al usuario a presionar los botones respectivos en su control. Cuando el usuario acertara a una nota, este sumaría puntos de lo contrario, perdería puntos.

IV. FIGURAS

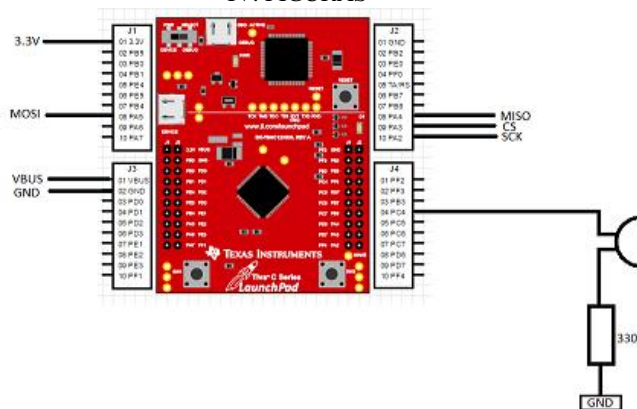


Figura 1. Esquemático de la Tiva™C.

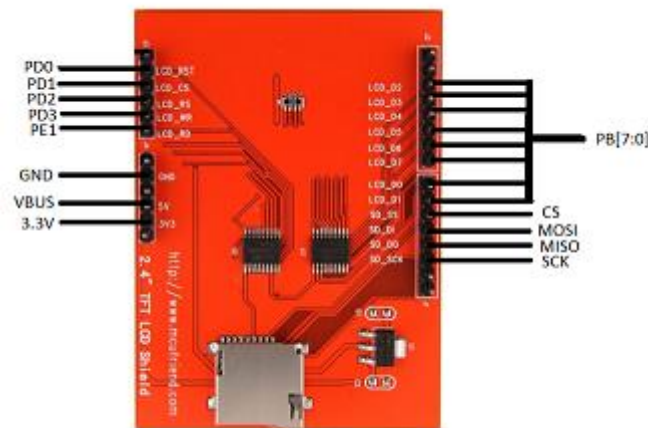


Figura 2. Esquema de la pantalla ILI9341.

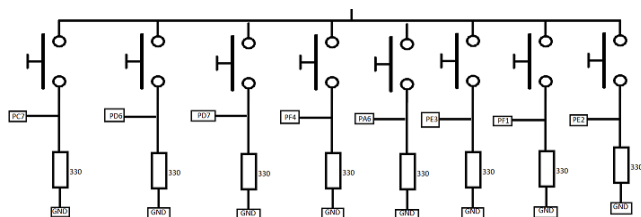


Figura 3. Esquema de los botones para los controles.

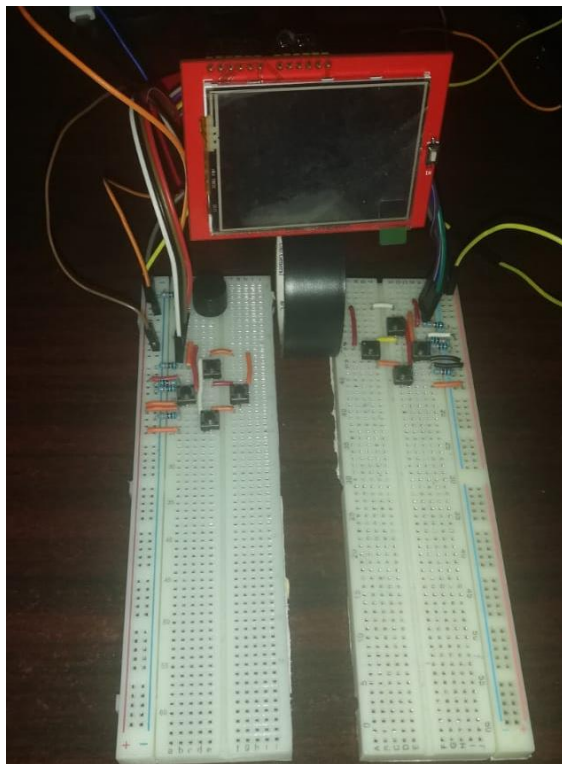


Figura 4. Circuito (vista frontal)

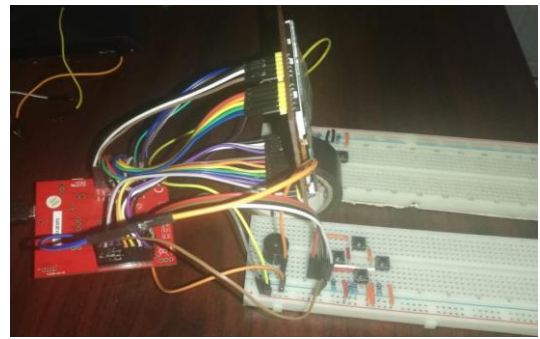


Figura 5. Circuito (vista lateral)

V. MATERIALES

A. TivaTMC TM4C123G

Es un microcontrolador con una memoria de programación de 256kB. Tiene 40 pines que se distribuyen en analógicos, digitales, pwm, tierra, 3.3V, 5V, etc. También cuenta con varios protocolos de comunicación.

B. Pantalla ILI9341

Es una pantalla con una resolución de 320 x 240 pixeles. También cuenta con un módulo SD que se comunica a través de SPI.

C. Buzzer Pasivo

Es un dispositivo que permite convertir una señal eléctrica en una onda de sonido.

D. Push Button

Es un interruptor que se acciona cuando pulsamos sobre él y se desactiva cuando lo soltamos.

E. Energia IDE

Es la plataforma que se utiliza para programar la TivaTMC y está basado en el lenguaje de programación C++.

F. lcd-image-converter

Es el software utilizado para convertir las imágenes a un archivo de texto con una lista de valores hexadecimal. Es necesario para convertir las imágenes a un formato de color R5G6B5.

VI. CÓDIGO

Para organizar las pantallas anteriormente definidas, utilizamos algunas variables que funcionaban como banderas o indicadores para que los botones tuviesen anti-rebote, que le diesen valores a distintas variables para saber si el usuario entro al modo de uno o dos jugadores (modeSelect), saber qué personaje escogió el jugador 1 y el jugador 2 (characterSelect y carácterSelect2), saber qué canción seleccionó el jugador (songSelect) y una que sirve como bandera para entrar al gameplay (game). Es importante mencionar que se utilizó el módulo SD para almacenar las pantallas de fondo y las imágenes de los jugadores y poder recuperarlas para desplegarlas en cada pantalla que se necesite.

Para la jugabilidad, es necesario definir que tenemos tres funciones llamadas "GamePlaySong". Cada una realiza el mismo procedimiento, pero con una canción distinta (únicamente cambia el nombre de los arrays que contienen la melodía y la duración de la nota que utiliza). Lo primero que hace esta función es reiniciar los valores de las variables, dejar caer las cuatro flechas hasta el punto donde le indica al jugador en qué momento debe tocarlas, es la forma de mostrarle al jugador que el juego está a punto de iniciar. Luego, entra en un ciclo que se repite hasta que ya no haya más notas que tocar. Dentro de este, se va guardando el tiempo actual y se verifica que la resta entre el tiempo actual y el tiempo en que se comenzó el juego no sea mayor al tiempo de la duración de la nota actual, de lo contrario, el tiempo previo se vuelve el tiempo actual, se cambia de nota y se escoge de forma aleatoria la nueva flecha que caerá. Es importante mencionar que, durante esto, también se hace la animación de las flechas cayendo.

Para hacer que coincidan las flechas con el momento en el que se cambia de nota, se hizo un cálculo. Primero, sabemos que si fuese instantánea la ejecución de cada línea, las flechas tardarían aproximadamente 1600 ms en llegar a la parte baja, bajando pixel por pixel entonces hicimos una proporción entre esto para obtener el tiempo que tardaría en caer si tuviese que recorrer más de un pixel por ciclo (este tiempo debe ser igual a la duración de cada nota, por lo que nos interesa obtener el número de "saltos" de pixel que debe hacer cada ciclo para caer en el tiempo deseado). Este cálculo de "saltos" se realiza cada vez que se cambia de nota, debido a que cada nota tiene duración diferente. Para revisar si el jugador está presionando o no en el momento adecuado, cada vez que se va a hacer un cambio de nota el debería de estar presionando el botón que indique la flecha (escogido por la variable frand) si el estado del botón coincide en ese momento con el número de la flecha, entonces el jugador gana puntos, sino, este pierde puntos.

Este procedimiento de puntaje también se realiza para los controles del jugador 2 si ese es el caso y la única diferencia entre estos dos modos de juego en cuestión de gameplay es que al final se muestra el puntaje de ambos jugadores (no solo de uno como en el primer jugador), se comparan y se muestra el Sprite del personaje que haya escogido el jugador que obtuvo la mayor cantidad de puntos.

VII. ENLACES

A. *GitHub*

<https://github.com/JALR0701/Proyecto-2>

B. *Video del Funcionamiento*

<https://youtu.be/lopDVXhhvzc>