

Práctica Experimental Propuesta

JUEGO SIMÓN

Introducción

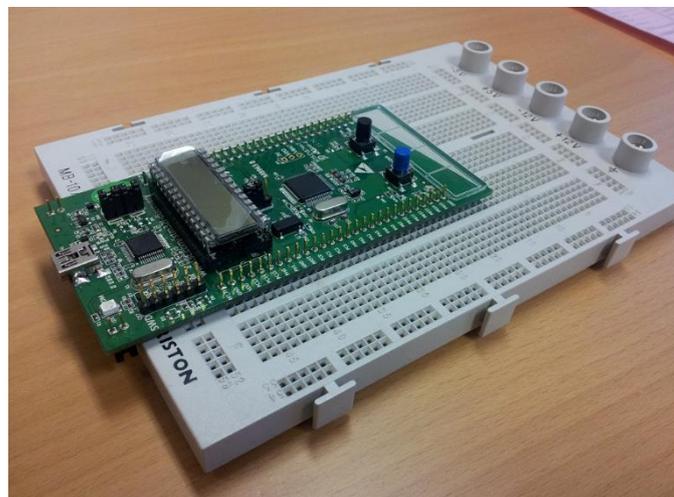
Este documento describe una práctica experimental a ser realizada por el alumno, de forma que, partiendo de lo visto en el material de clase y en los ejercicios (y soluciones) propuestos, consiga los objetivos expuestos. Los objetivos de esta práctica son:

- Aprender las técnicas básicas de programación de microcontroladores
- Aprender a diseñar un conjunto hardware-software para cubrir los requisitos de una aplicación
- Aprender a depurar y verificar el funcionamiento del conjunto hardware-software desarrollado
- Ofrecer un reto asequible pero que estimule la capacidad de diseño e innovación del estudiante

La práctica se va a dividir en una serie de hitos, a ser realizados de forma secuencial. En cada uno de los hitos se indica los Temas que se han debido estudiar antes de acometer la realización de dicho hito.

Para realizar las prácticas, será aconsejable que inserte la placa de desarrollo del curso (la STM32L-Discovery), en una placa de pruebas (protoboard), de tal forma que no se cortocircuiten los pines de los puertos P1 y P2, ni éstos entre sí. De esta forma, podrá conectar cables a los distintos pines, para conectar resistencias, potenciómetros, etc.

Es importante mantener los dos jumpers de la parte inferior de la placa conectados, por lo que se recomienda una instalación como figura en las fotografías adjuntas. Si no es posible hacerlo de esa forma, entonces retire los jumpers, inserte la placa en la protoboard, y conecte los dos pines que estaban con los jumpers en la propia protoboard.





Además, se recomienda que el alumno ponga el JUMPER J1 (situado justo debajo del LCD) en la posición OFF (conectando los dos pines que están más alejados del LCD).

Desarrollo de la Práctica

Las prácticas consisten en el desarrollo e implementación de una versión del juego Simón. El juego original consiste en memorizar una secuencia aleatoria de luces y sonidos que el jugador debe reproducir en el orden correcto y en un tiempo determinado¹. Si no conoce el juego, le recomendamos usar alguno de los simuladores gratuitos disponibles en Internet, como por ejemplo: <http://www.memo-juegos.com/juegos-simon/juego-de-simon-online-gratis>

En la versión a implementar sustituiremos las luces por caracteres en la pantalla LCD. Se usará un carácter distinto para cada sonido/pulsador, y se dispondrán los pulsadores en línea (para poder relacionarlos con los caracteres) como muestra el diagrama de bloques de la figura. Si se quiere utilizar luces en lugar del LCD, entonces habrá que desconectar el LCD y utilizar los pines que tiene dedicados la STM32L-DISCOVERY para el LCD. No es una opción que se recomiende al principio, pero puede ser una mejora a realizar una vez resuelto el resto de la práctica.

El juego comenzará cuando el jugador pulse un botón. En ese momento el juego mostrará uno de los caracteres, emitirá un sonido, apagará la pantalla y esperará a que el jugador pulse el botón correspondiente. Si el jugador acierta el botón, el juego muestra el carácter de nuevo emitiendo su sonido, y añade un carácter a la secuencia con su sonido correspondiente, apaga la pantalla y espera a que el jugador reproduzca la secuencia de botones. El juego continúa hasta que el jugador se equivoque o hasta que se alcance el tamaño de secuencia máxima, que será 6 inicialmente, y posteriormente será ajustable mediante un potenciómetro.

El diagrama de bloques del sistema a desarrollar se puede ver reflejado en la siguiente figura:

¹ [https://es.wikipedia.org/wiki/Simon_\(juego\)](https://es.wikipedia.org/wiki/Simon_(juego))

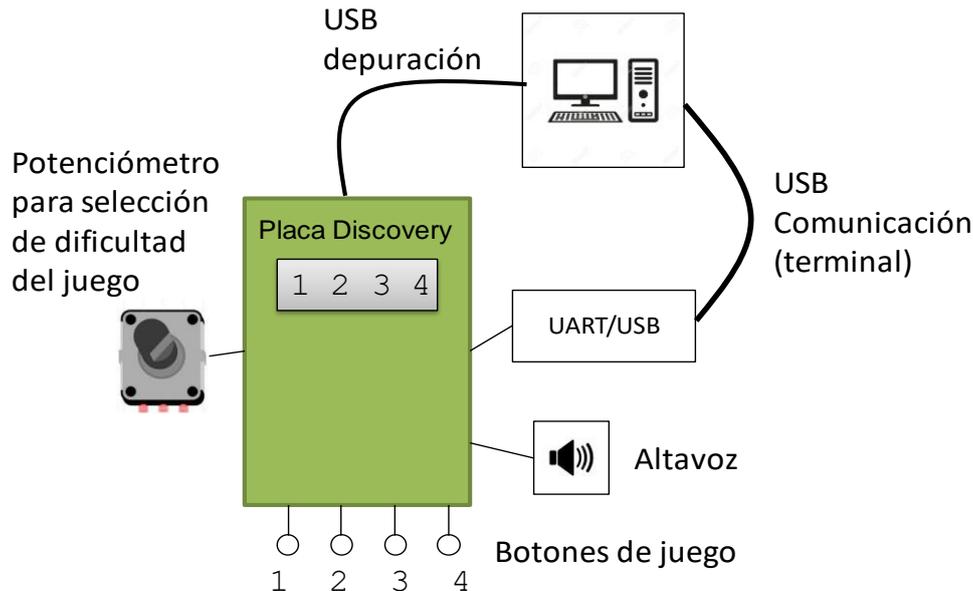


Ilustración 1: Diagrama de Bloques del Proyecto

Para poder seleccionar los sonidos asociados a cada pulsador, en el Anexo 1 de esta memoria se recoge una tabla con las frecuencias de las notas musicales.

Material a adquirir

Para realizar la práctica, el alumno debe disponer del siguiente material:

- Placa STM32L-DISCOVERY y cable USB
- 3 placas de pruebas (protoboards) pequeñas o una placa grande
- 4 pulsadores para inserción en placa (6mmx6mm)
- Altavoz 0.5W 8Ω
- Transistor BC548 o 2N2222A
- Resistencia 2K2 1/4W
- Potenciómetro de 1 vuelta de 5 kΩ.. SE RECOMIENDA QUE EL POTENCIÓMETRO NO SEA MULTIVUELTA
- Cable 0.5mm para protoboard, y Cables DuPont macho-hembra
- Pelacables y cortacables

A la protoboard no hay que suministrarle ninguna fuente de alimentación externa, sino que todas las fuentes de alimentación proceden de la tarjeta STM32L-Discovery, la cual se alimenta con el mismo conector USB con el que se realiza la descarga del programa en el micro. Es decir, cuando conectemos el cable USB a nuestro PC por un lado y, por otro lado, a la tarjeta, estaremos no sólo comunicando la tarjeta y el PC sino también alimentando la tarjeta y todos sus componentes. Para ello tiene un pin de 3V en la parte más alta del conector de expansión ubicado a la izquierda de la placa, y otro de 5V en la parte más alta del conector de expansión del otro lado.

El desarrollo de la práctica se va dividir en varios pasos con hitos intermedios que se describen a continuación. Para facilitar la tarea de asignación de pines inicial, se adjunta una tabla con la asignación de pines que debe hacer en el Anexo 3.

Es recomendable leer todos los pasos y mejoras propuestas al comienzo del proyecto, es decir, antes de empezar con el Hito 1.

Hito 1: Interfaz de jugador, botones (Tema 5 y, opcionalmente, Tema 7)

El objetivo en este primer paso es implementar la lectura de los botones o pulsadores, de forma que permita determinar qué botón ha sido pulsado (vea Anexo 2).

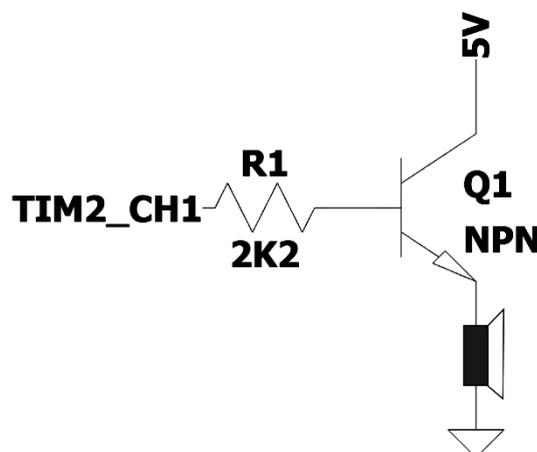
Se deben poder identificar 5 botones distintos (4 de juego y 1 de inicio). Cada botón de juego se asociará a un número que se mostrará en una pantalla LCD, mientras que el botón de inicio hará mostrar el mensaje "INIT" en la pantalla. El mensaje o el número aparece en la pantalla cuando se pulsa el botón correspondiente y se borra cuando se suelta dicho botón.

Utilice los mecanismos de depuración disponibles para comprobar el correcto funcionamiento de su solución y, en caso de un funcionamiento erróneo, para encontrar el origen de los errores:

- Puntos de ruptura a lo largo de la ejecución
- Evaluación de variables
- Evaluación de los registros de los periféricos utilizados.

Hito 2: Interfaz de jugador, sonido (Temas 7 y 8)

El objetivo de esta sesión consiste añadir la función de reproducción del sonido al paso 1. Se utilizará un temporizador para generar señales cuadradas de la frecuencia correspondiente a la nota a reproducir, que se conectará a un altavoz, usando el siguiente esquema. Usará la salida asociada al temporizador TIM2, canal 1 configurada para que genere una onda cuadrada de la frecuencia deseada (vea Anexo 1).



La nota musical debe sonar mientras el botón está pulsado. En función del botón seleccionado se reproduce una u otra nota. Sólo se asociarán notas a los botones de juego.

Hito 3: El juego reproduce una secuencia (Temas 6, 7 y 8)

Usar el botón USER de la placa como botón de inicio. Añadir la siguiente función a dicho botón de inicio (a través de una interrupción externa del microcontrolador): cada vez que se pulse el botón el sistema reproduce una secuencia almacenada en memoria combinando los 4 sonidos del Paso 2. El tamaño de la secuencia en este hito será 6 (ampliable en hitos posteriores). Cada sonido estará asociado al carácter 1, 2, 3 o 4 de la pantalla LCD, de forma que se mostrará el carácter correspondiente mientras suene su nota. Cada nota debe durar medio segundo.

La duración de cada nota deberá gestionarse con un temporizador del microcontrolador (no con esperas), y deberá poder interrumpirse en cualquier momento (por ejemplo, pulsando un botón).

Una vez resuelta esta parte del hito, añada la funcionalidad de que un potenciómetro determine el número de posiciones a recorrer de la secuencia almacenada en memoria, que se guardará en una tabla. Antes de iniciar la reproducción se leerá la posición del potenciómetro con ayuda del ADC, seleccionando el número de elementos de la secuencia guardada que se reproducirán. Es suficiente con distinguir 4 casos. Por ejemplo, distinguiendo entre reproducir 6, 7, 8 o 9 elementos de la secuencia guardada. Es decir, tendrá que escoger 4 valores límite de tensión para que, dependiendo de la posición del potenciómetro, se decida un número de elementos de secuencia u otro.

Hito 4: Juego completo (básico y avanzado) (Temas 5 – 10)

Con los logros obtenidos en los hitos previos, ha de implementar el juego básico, el cual podrá ser mejorado implementado el juego avanzado.

Juego básico: Implementar el algoritmo del juego usando la secuencia guardada en el Hito 3, e integrando los Hitos 1 y 2. El juego terminará cuando el jugador se equivoque o cuando se alcance la longitud de secuencia seleccionada con el potenciómetro. Una vez finalizado el juego, se mostrará en el LCD el mensaje "TOT x", siendo x el número de pulsaciones correctas.

Juego avanzado: Añadir una conexión serie asíncrona que conecte la placa con el ordenador. Mediante el uso de un programa de transmisión serie (como, por ejemplo, el Tera Term), se implementará la siguiente funcionalidad:

- El PC le dirá, mediante un carácter, a la placa, el número de rondas de la que constará el juego. A partir de recibir este carácter, la placa considerará que empieza un nuevo juego, que estará compuesto por tantas rondas como se ha indicado, empezando cada ronda mediante la pulsación del botón USER.
- Al acabar cada ronda, la placa mandará al PC el mensaje "RESULTADO RONDA x: y", siendo x el número de ronda, e y el resultado obtenido. El mensaje acabará con un retorno de carro.
- Al acabar el juego, la placa mandará al PC los mensajes siguientes, separados por un retorno de carro:
 - "PUNTUACION MAS ALTA: x", siendo x el resultado más alto obtenido
 - "PUNTUACION MAS BAJA: y", siendo y el resultado más bajo obtenido
 - "MEDIA DEL JUEGO: z", siendo z la media de las puntuaciones, con un decimal

Hito 5: Mejoras del Juego

Una vez resuelto el juego avanzado, implemente las siguientes mejoras:

- La generación de las notas mediante ondas cuadradas genera mucha distorsión, de manera que una mejora es la generación de ondas sinusoidales mediante el uso del DAC. Usará el mismo circuito del altavoz, pero conectado a la salida del DAC2. Para generar una onda sinusoidal con el DAC, almacene en memoria una tabla de valores de una onda sinusoidal que pueda leerse a distintas frecuencias. En función de la nota que se quiera reproducir, se debe variar la velocidad con la que se recorre dicha tabla.
- Genere la secuencia aleatoriamente. Para ello, tras el reset del sistema se arrancará un temporizador libre que se consultará cuando el jugador pulse el botón de inicio. El valor del temporizador se usará como semilla para generar la secuencia de longitud máxima.
- Cambie la funcionalidad del potenciómetro. La longitud de secuencia máxima será fija, y la duración de la reproducción de cada elemento de la secuencia se acorta de acuerdo al nivel de dificultad seleccionado con el potenciómetro.
- Añada un tiempo límite para la respuesta del jugador. Si el jugador se demora más de 2 segundos en pulsar algún botón de la secuencia se acaba el juego.
- Reproduzca una melodía al finalizar el juego. La melodía variará dependiendo del resultado (perdedor, ganador o record de juego).

Anexo 1. Frecuencias de las principales notas musicales, Octavas 3 y 4

OCTAVA 3	DO	130,812783 Hz
		138,591315 Hz
	RE	146,832384 Hz
		155,563492 Hz
	MI	164,813778 Hz
	FA	174,614116 Hz
		184,997211 Hz
	SOL	195,997718 Hz
	207,652349 Hz	
LA	220,000000 Hz	
	233,081881 Hz	
SI	246,941651 Hz	
OCTAVA 4	DO	261,625565 Hz
		277,182631 Hz
	RE	293,664768 Hz
		311,126984 Hz
	MI	329,627557 Hz
	FA	349,228231 Hz
		369,994423 Hz
	SOL	391,995436 Hz
	415,304698 Hz	
LA	440,000000 Hz	
	466,163762 Hz	
SI	493,883301 Hz	

Anexo 2. Pulsadores o botones

Los botones del juego se pueden implementar con pulsadores conectados a GPIOs de entrada con resistencia interna de pull-up. Cada pulsador se conectará entre el pin del puerto y masa.

Si se desea una botonera más original pueden pintarse con grafito (lápiz) en una cartulina los 4 botones (dándoles la forma que nos guste). En los extremos de los botones se pueden conectar cables cocodrilo que enlacen cada tecla con su GPIO correspondiente. En este caso, la resistencia interna de pull-up es demasiado pequeña. Deberá configurar el puerto sin pull-up, y poner una externa de mayor tamaño, dependiendo de cómo planea cerrar el circuito.

Si desea cerrar el circuito usando su propio cuerpo necesita una resistencia de pull-up muy alta, al menos de $10M\Omega$. En ese caso debe conectar alguna parte de su cuerpo a masa, usando por ejemplo una pulsera.

Si desea cerrar el circuito usando un cable, deberá conectar un extremo a masa y el otro al botón que se desea tocar. En este caso, una resistencia de $1M\Omega$ es suficiente para detectar el cierre del circuito a través del grafito.

Vea los siguientes enlaces para obtener su inspiración:

- https://leantec.es/blog/49_Como-hacer-m%C3%BAsica-con-Arduino-y-sensores-t%C3%A1ct.html
- <https://www.youtube.com/watch?v=rfQgh7iCcOU>

Anexo 3. Distribución de pines y configuración de Puertos

Nombre	Puerto	Configuración	Descripción
BOTÓN INICIO	PA0	Entrada Digital con Pull-up	tecla
BOTON 1	PA11	Entrada Digital con Pull-up	tecla
BOTON 2	PA12	Entrada Digital con Pull-up	tecla
BOTON 3	PD2	Entrada Digital con Pull-up	tecla
BOTON 4	PH1	Entrada Digital con Pull-up	tecla
TIM2_CH1 (DAC_OUT2)	PA5	Alternate Function	Conexión de Altavoz con control por onda cuadrada (control por onda sinusoidal)
ADC_IN4	PA4	Puerto analógico	Conexión de Potenciómetro
USART1_RX	PB7	Alternate Function	Conexión a PC con puente UART-USB En la configuración por defecto es un LED de la placa, si no se usa el puerto serie, no es necesario cambiar la configuración de este pin
USART1_TX	PB6	Alternate Function	Conexión a PC con puente UART-USB En la configuración por defecto es un LED de la placa, si no se usa el puerto serie, no es necesario cambiar la configuración de este pin