

# **Monografía sobre el desarrollo de juego Simon en un STM32.**

Teoría de Circuitos Electrónicos  
Ingeniería en Computación

Docente:  
**Norberto Caudet**

Alumno:  
**Franco Sgro**

Ciclo Lectivo:  
**Segundo Cuatrimestre, 2023**

<b>1. Introducción.....</b>	<b>4</b>
<b>1.1 Contextualización.....</b>	<b>4</b>
<b>1.2 Objetivos:.....</b>	<b>4</b>
1.2.1 Interfaz de Usuario:.....	4
1.2.2 Generación de Secuencias:.....	4
1.2.3 Niveles de Dificultad:.....	5
1.2.4 Repetición de Secuencia:.....	5
1.2.5 Temporizador:.....	5
1.2.6 Validación de Respuesta:.....	5
1.2.7 Incremento de Dificultad:.....	5
1.2.8 Seteo de Fecha:.....	5
1.2.9 Funcionalidad de Reinicio:.....	5
1.2.10 Velocidad del Juego:.....	5
<b>2. Marco Teórico.....</b>	<b>6</b>
<b>2.1 Descripción del Juego Simon: Reglas y Funcionamiento.....</b>	<b>6</b>
<b>2.2 Microcontrolador STM32: Breve Introducción y su Aplicación en el Proyecto.....</b>	<b>6</b>
<b>2.3 Proyectos Similares: Revisión Breve de Implementaciones Previas de Juegos Análogos.....</b>	<b>7</b>
2.3.1 Proyecto "Simon Says Arduino":.....	7
2.3.2 "Raspberry Pi Simon Game":.....	7
2.3.3 "Simon Game with PIC Microcontroller":.....	7
<b>3. Diseño y Desarrollo.....</b>	<b>8</b>
<b>3.1 Esquemático, Disposición de Componentes y Consideraciones de Diseño.....</b>	<b>8</b>
<b>3.2 Componentes Electrónicos: Descripción de LEDs, Botones y Elementos Esenciales....</b>	<b>9</b>
3.2.1 LEDs (Diodos Emisores de Luz):.....	9
3.2.2 Botones:.....	9
3.2.3 Resistencias:.....	9
3.2.4 Transistores:.....	9
3.2.5 Display Alfanumérico de 2 Filas y 16 Columnas:.....	9
3.2.6 Potenciómetro:.....	10
<b>3.3 Funcionalidad del STM32: Papel del Microcontrolador en el Proyecto.....</b>	<b>10</b>
3.3.1 Gestión de Periféricos y Entradas:.....	10
3.3.2 Generación de Secuencias Aleatorias:.....	10
3.3.3 Control de Niveles de Dificultad:.....	10
3.3.4 Temporización y Validación de Respuestas:.....	11
<b>3.4 Recursos y Herramientas Utilizados: Proteus y STM32IDE.....</b>	<b>11</b>
3.4.1 Proteus:.....	11
3.4.1 STM32IDE:.....	11
<b>4. Programación.....</b>	<b>12</b>
<b>4.1 Flujo del Código y Estados Principales en el Loop del Main.....</b>	<b>12</b>
4.1.1 Estado 0 (Permanente):.....	12
4.1.2 Estado 1 (Programación de Fecha y Hora):.....	12
4.1.3 Estado 2 (Programación de Dificultad):.....	12
4.1.4 Estado 3 (Juego "Simon" en Proceso):.....	12
4.1.5 Estado 4 (Fin del Juego - GAME OVER):.....	12
<b>4.2 Funciones Importantes utilizadas en los Estados Principales.....</b>	<b>13</b>
4.2.1 Función Display_DiaHora(void):.....	13

4.2.2 Función Teclado(void):.....	13
4.2.3 Función Validador_dia(void):.....	13
4.2.4 Función ConversorAD(void):.....	13
4.2.5 Función actualizarReloj(void):.....	13
4.2.6 Función generarSecuencia(uint8_t secuencia[], uint8_t largo):.....	13
4.2.7 Función mostrarSecuencia(const uint8_t secuencia[], uint8_t largo):.....	13
4.2.8 Función compararTiempo(uint32_t tiempoInicio):.....	14
4.2.9 Función leerInputUsuario(uint8_t secuenciaUsuario[], uint8_t largo):.....	14
<b>5. Conclusiones.....</b>	<b>15</b>
<b>5.1 Verificación de los objetivos planteados.....</b>	<b>15</b>
5.1.1 Diseño e Implementación del Juego "Simon".....	15
5.1.2 Integración del Microcontrolador STM32F103C8.....	15
5.1.3 Niveles de Dificultad y Configuraciones Adicionales.....	15
5.1.4 Validación de Respuestas y Progresión del Juego.....	15
5.1.5 Capacidad de Modificar Fecha y Hora.....	15
<b>5.2 Puntos clave descubiertos durante el proyecto.....</b>	<b>15</b>
5.2.1 Desafíos en la Gestión del Tiempo.....	15
5.2.2 Ajuste de Velocidad y Dificultad.....	16
5.2.3 Interfaz de Usuario Intuitiva.....	16
5.2.4 Comparación Simulador vs. Hardware Real.....	16
5.2.5 Consideración del Uso de Memoria.....	16
<b>6. Recomendaciones.....</b>	<b>17</b>
<b>6.1 Mejoras y Expansiones.....</b>	<b>17</b>
6.1.1 Almacenamiento de Puntajes en la Memoria del STM32:.....	17
6.1.2 Optimización del Uso de Memoria:.....	17
6.1.3 Pruebas en Hardware Real:.....	17
<b>7. Referencias.....</b>	<b>18</b>
7.1 Getting started with STM32, STM32base.Org.....	18
7.2 STM32 Reference Manual. STMicroelectronics,.....	18
7.3 Documentación de Proteus. Labcenter Electronics,.....	18
<b>8. Código Fuente:.....</b>	<b>19</b>
8.1 Repositorio en Github.....	19

# 1. Introducción

La presente investigación se inicia con una meticulosa introducción al juego Simon, un dispositivo electrónico de entretenimiento que desafía las habilidades de memoria y atención de los usuarios. La elección del microcontrolador STM32 como componente central de este proyecto se fundamenta en consideraciones técnicas y prácticas. La conjunción de estos dos elementos, el juego Simon y el microcontrolador STM32, ha sido estratégicamente seleccionada con el objetivo de explorar las posibilidades de implementación de un sistema electrónico innovador.

## 1.1 Contextualización

La relevancia de abordar el juego Simon se encuentra en su inherente complejidad cognitiva, demandando no solo la capacidad de recordar patrones visuales y auditivos, sino también la destreza en la ejecución precisa de dichos patrones. Este desafío, enmarcado en un contexto lúdico, proporciona un terreno propicio para la exploración de las capacidades de programación y control electrónico que ofrece el microcontrolador STM32.

La elección del microcontrolador STM32 se sustenta en sus características avanzadas, tales como su arquitectura de bajo consumo de energía, capacidad de procesamiento eficiente, y versatilidad en la gestión de interfaces y periféricos. Estas cualidades posicionan al STM32 como una plataforma idónea para la implementación de proyectos electrónicos complejos y de alto rendimiento, como el emulador del juego Simon propuesto en esta investigación.

La justificación para la combinación de estos dos elementos radica en la sinergia entre la complejidad del juego Simon y las capacidades técnicas del microcontrolador STM32. El juego aporta una dimensión cognitiva y de entretenimiento, mientras que el STM32 ofrece la infraestructura electrónica necesaria para la ejecución y control del juego. Así, la amalgama de estas dos entidades promete no solo un ejercicio técnico y programático significativo, sino también la creación de un dispositivo electrónico recreativo y desafiante.

## 1.2 Objetivos:

La presente investigación tiene como propósito el diseño e implementación del juego "Simon" mediante la utilización del microcontrolador STM32F103C8. La finalidad primordial es gestionar diversas acciones funcionales dentro del contexto del juego electrónico propuesto. A continuación, se detallan las consignas fundamentales que regirán el desarrollo de este proyecto:

### 1.2.1 Interfaz de Usuario:

Objetivo: Diseñar una interfaz visual que refleje la secuencia de luces mediante el uso de LEDs, asociando los sonidos correspondientes a cada color.

Componentes Involucrados: LEDs para representar los colores (rojo, verde, amarillo y azul).

### 1.2.2 Generación de Secuencias:

Objetivo: Implementar en el microcontrolador la capacidad de generar secuencias aleatorias de luces y sonidos, las cuales constituirán la pauta que los jugadores deberán replicar.

Componentes Involucrados: Microcontrolador STM32F103C8.

### **1.2.3 Niveles de Dificultad:**

Objetivo: Definir niveles de dificultad mediante la gestión de dos variables: el tiempo disponible para recrear la secuencia y la duración del encendido de LEDs al mostrar la secuencia.

Componentes Involucrados: Pad de navegación con Switch B para cambiar entre grados de dificultad.

### **1.2.4 Repetición de Secuencia:**

Objetivo: Permitir a los jugadores ingresar sus respuestas replicando la secuencia generada por el microcontrolador. Utilizar botones para la interacción del jugador.

Componentes Involucrados: Pad de navegación con botones asociados a la respuesta del jugador.

### **1.2.5 Temporizador:**

Objetivo: Establecer un temporizador que limite el tiempo para que los jugadores respondan después de que el microcontrolador indique la secuencia.

Componentes Involucrados: Reloj del microcontrolador y pantalla alfanumérica de 2x16 columnas.

### **1.2.6 Validación de Respuesta:**

Objetivo: Implementar la validación de la secuencia ingresada por el jugador. En caso de respuesta incorrecta, finalizar el juego, mostrar la duración de la partida y crear un ranking de los mejores tiempos en la mayor cantidad de niveles completados.

Componentes Involucrados: Lógica de validación en el microcontrolador.

### **1.2.7 Incremento de Dificultad:**

Objetivo: Aumentar la longitud y complejidad de la secuencia generada a medida que los jugadores progresan.

Componentes Involucrados: Algoritmo de generación de secuencias basado en el nivel de dificultad elegido.

### **1.2.8 Seteo de Fecha:**

Objetivo: Configurar la fecha para permitir el almacenamiento y visualización del top 2 de récords en la memoria EEPROM del microcontrolador.

Componentes Involucrados: Teclado de navegación y pantalla alfanumérica.

### **1.2.9 Funcionalidad de Reinicio:**

Objetivo: Agregar la capacidad de reiniciar el juego en cualquier momento para brindar a los jugadores la oportunidad de intentar nuevamente.

Componentes Involucrados: Switch D del pad de navegación.

### **1.2.10 Velocidad del Juego:**

Objetivo: Permitir la variación de la velocidad del juego mediante el uso de un potenciómetro.

Componentes Involucrados: Potenciómetro dispuesto para tal fin.

## **2. Marco Teórico**

### **2.1 Descripción del Juego Simon: Reglas y Funcionamiento**

El juego "Simon" constituye un dispositivo electrónico de entretenimiento concebido para poner a prueba las habilidades cognitivas y de atención de los usuarios. Desarrollado por Ralph H. Baer y Howard J. Morrison, su introducción en 1978 marcó un hito en la intersección entre la electrónica y el entretenimiento lúdico. El propósito fundamental del juego radica en desafiar la memoria secuencial de los participantes, exigiéndoles repetir con precisión secuencias de luces y sonidos generados por el dispositivo.

Las reglas básicas del juego Simon son fundamentales para comprender su dinámica. El dispositivo consta de cuatro botones o sectores, cada uno asociado a un color específico: rojo, verde, amarillo y azul. La secuencia de juego inicia con la iluminación de un color aleatorio, seguido de un tono auditivo correspondiente. El participante debe replicar la secuencia presionando los botones en el mismo orden en que fueron presentados por el dispositivo.

A medida que el juego progresa, la complejidad se intensifica. La secuencia generada por el dispositivo se alarga y la precisión requerida para su replicación aumenta, desafiando progresivamente la capacidad de retención y ejecución del jugador. La omisión de un solo paso o la incorrecta reproducción de la secuencia resulta en la finalización del juego.

El juego Simon, por ende, amalgama elementos visuales y auditivos para estimular la memoria secuencial y la concentración de los participantes. Su presencia en la cultura del entretenimiento electrónico evidencia su relevancia histórica y su capacidad para ofrecer desafíos cognitivos cautivantes.

### **2.2 Microcontrolador STM32: Breve Introducción y su Aplicación en el Proyecto**

El microcontrolador STM32, desarrollado por STMicroelectronics, constituye una entidad emblemática en el ámbito de la ingeniería electrónica y sistemas embebidos. Esta unidad de procesamiento se erige como una solución integral, destacando por su arquitectura avanzada, eficiencia energética y versatilidad en la gestión de interfaces y periféricos.

En su esencia, el STM32 se fundamenta en la arquitectura ARM Cortex-M, caracterizada por su capacidad de procesamiento eficiente y su flexibilidad para adaptarse a una diversidad de aplicaciones. Su presencia ha proliferado en sectores como la automatización industrial, dispositivos médicos, sistemas de control, y en particular, en proyectos de electrónica de consumo y entretenimiento.

La aplicabilidad del microcontrolador STM32 en el contexto del proyecto de emulación del juego "Simon" radica en su capacidad para orquestar diversas acciones funcionales de manera precisa y eficaz. La gestión de interfaces, periféricos y la capacidad de procesamiento del STM32 se alinean con los requerimientos específicos de la interfaz de usuario, la generación de secuencias aleatorias, el control de niveles de dificultad, y demás funcionalidades delineadas en las consignas del proyecto.

La elección del STM32 se sustenta en su capacidad para ejecutar algoritmos complejos, interactuar con múltiples dispositivos de entrada y salida, y proporcionar una plataforma robusta para el

desarrollo de sistemas embebidos avanzados. Su presencia en este proyecto no solo enfatiza la dimensión técnica del mismo, sino también la aspiración de alcanzar niveles óptimos de eficiencia y rendimiento en la implementación del juego "Simon".

En conclusión, la breve introducción al microcontrolador STM32 destaca su relevancia en el marco del proyecto, delineando las características distintivas que hacen de este componente una elección estratégica. La capacidad del STM32 para integrar y coordinar las complejas operaciones requeridas en el juego "Simon" representa un componente crítico para el logro exitoso de los objetivos planteados en este proyecto.

## **2.3 Proyectos Similares: Revisión Breve de Implementaciones Previas de Juegos Análogos**

La exploración de proyectos anteriores que han abordado la implementación de juegos similares al propuesto, como el juego "Simon", ofrece una perspectiva valiosa para comprender enfoques previos y posiblemente identificar mejores prácticas. La sinergia entre el entretenimiento interactivo y la ingeniería electrónica ha dado lugar a diversas iniciativas que comparten la premisa fundamental de desafiar la memoria secuencial y la capacidad de ejecución de los jugadores. A continuación, se presenta una revisión breve de algunos proyectos análogos:

### **2.3.1 Proyecto "Simon Says Arduino":**

Descripción: Este proyecto, basado en la plataforma Arduino, implementa una versión del juego "Simon" con LEDs y botones. La secuencia se genera aleatoriamente, y los participantes deben replicarla correctamente para avanzar de nivel. La revisión de este proyecto permite apreciar cómo se ha abordado la interfaz de usuario y la lógica de juego utilizando una plataforma diferente.

### **2.3.2 "Raspberry Pi Simon Game":**

Descripción: En este proyecto, se utiliza una Raspberry Pi como núcleo de control. Además de LEDs y botones, se integra una pantalla táctil para mejorar la interactividad. La revisión de esta implementación aporta insights sobre cómo la elección de hardware influye en la complejidad y las capacidades del juego.

### **2.3.3 "Simon Game with PIC Microcontroller":**

Descripción: Un proyecto que utiliza un microcontrolador PIC para emular el juego "Simon". La particularidad radica en la implementación de una memoria EEPROM para almacenar récords y configuraciones del juego. La revisión de esta implementación arroja luz sobre cómo se pueden incorporar funcionalidades adicionales, como el almacenamiento persistente de datos, en proyectos de esta naturaleza.

La breve revisión de estos proyectos proporciona un contexto enriquecedor para el desarrollo del juego "Simon" con el microcontrolador STM32 en este proyecto específico. La diversidad de enfoques y elecciones de hardware revela la flexibilidad y adaptabilidad de la premisa central del juego, permitiendo su implementación a través de distintas plataformas y tecnologías. Este análisis previo orienta la toma de decisiones en la selección de componentes y estrategias de implementación para optimizar el rendimiento y la experiencia del usuario en el presente proyecto.

### 3. Diseño y Desarrollo

#### 3.1 Esquemático, Disposición de Componentes y Consideraciones de Diseño

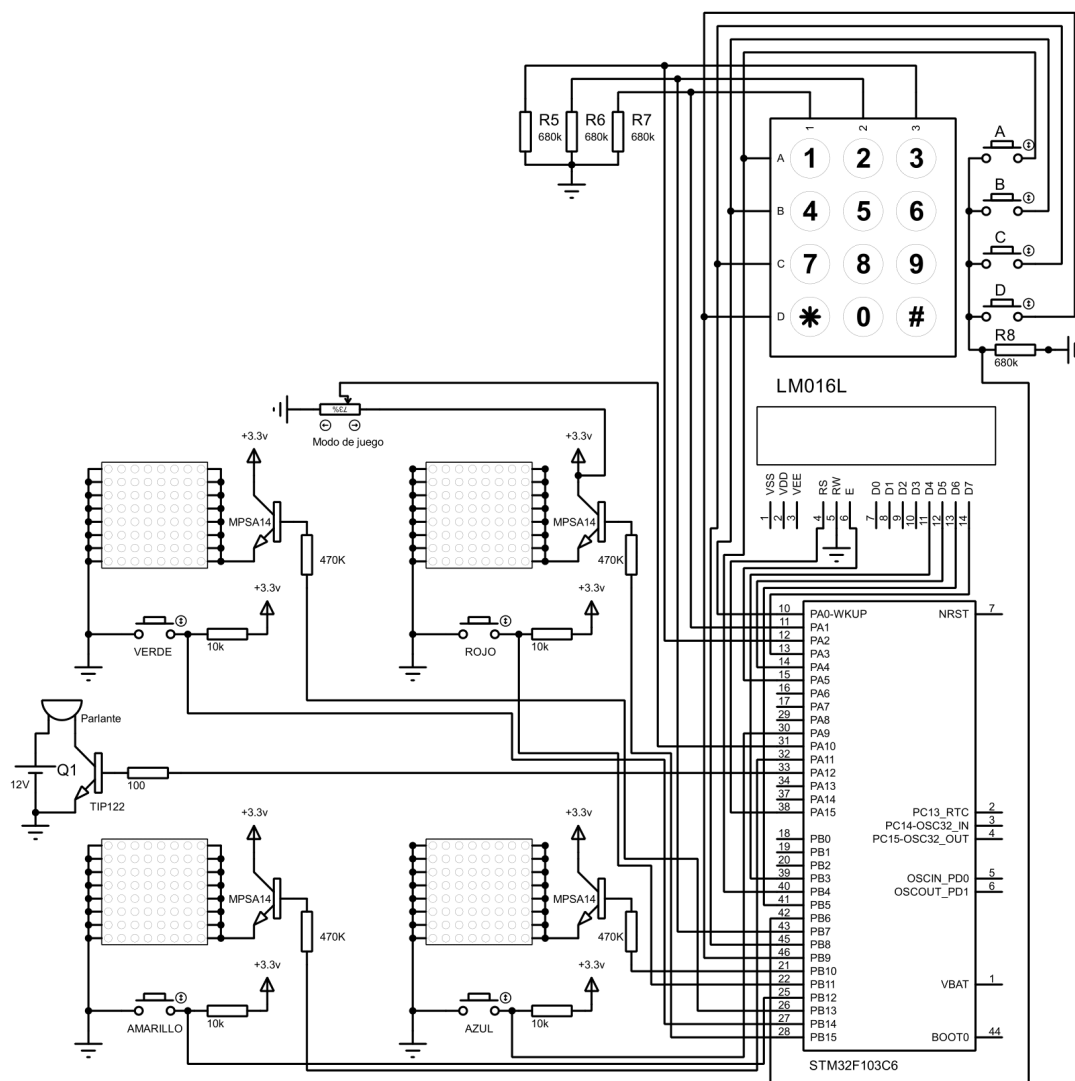


Figura 1. Esquemático del circuito electrónico.

El diseño del circuito que sustenta la implementación del juego "Simon" en el microcontrolador STM32F103C8 ha sido una fase crítica del proyecto, fundamentada en la armonización precisa de componentes electrónicos para garantizar un rendimiento óptimo y coherente. La planificación y disposición cuidadosa de cada elemento contribuyen a la eficiencia operativa y a la experiencia del usuario.

El esquemático del circuito aborda la representación gráfica y detallada de las conexiones entre los componentes clave, delineando la interconexión de LEDs, botones, el display alfanumérico, el potenciómetro y el microcontrolador STM32F103C8. La representación precisa de cada conexión y la definición clara de los pines asignados a cada función son elementos esenciales del esquemático.



La disposición física de los componentes en la placa de circuito impreso (PCB) es un aspecto crucial del diseño. La colocación estratégica de los LEDs, botones y demás elementos contribuye a la legibilidad del circuito, minimiza interferencias y facilita la implementación práctica. La ubicación del microcontrolador y su conexión con los demás componentes se ha planificado considerando la eficiencia en la disposición de la PCB.

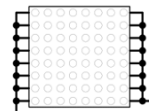
## 3.2 Componentes Electrónicos: Descripción de LEDs, Botones y Elementos Esenciales

Los componentes electrónicos seleccionados para la implementación del juego "Simon" desempeñan roles fundamentales en la estructura y funcionalidad del sistema. Cada elemento ha sido cuidadosamente elegido para contribuir a la generación de una interfaz interactiva y desafiante. A continuación, se presenta una descripción detallada de los componentes clave involucrados en el proyecto:

### 3.2.1 LEDs (Diodos Emisores de Luz):

**Función:** Representan visualmente los colores asociados al juego (rojo, verde, amarillo y azul).

**Aplicación en el Proyecto:** La iluminación secuencial de los LEDs genera patrones visuales que los jugadores deben replicar con precisión.



### 3.2.2 Botones:

**Función:** Constituyen los interruptores de entrada mediante los cuales los jugadores interactúan con el sistema.

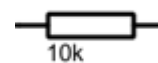
**Aplicación en el Proyecto:** Los botones son cruciales para la replicación de las secuencias generadas, así como para la navegación y selección de opciones en el menú de programación.



### 3.2.3 Resistencias:

**Función:** Limitan la corriente que fluye a través de los LEDs, protegiéndolos y asegurando una operación segura.

**Aplicación en el Proyecto:** Cada LED está asociado a una resistencia para controlar la intensidad luminosa y prevenir daños por corrientes excesivas.



### 3.2.4 Transistores:

**Función:** Actúan como interruptores controlados electrónicamente para gestionar la corriente que atraviesa los LEDs.

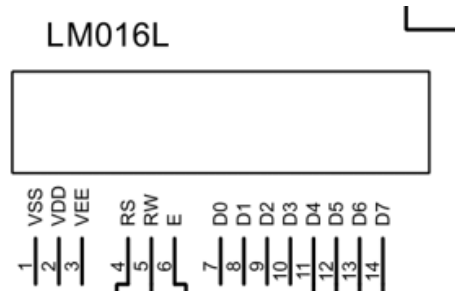
**Aplicación en el Proyecto:** Los transistores permiten controlar la activación y desactivación de los LEDs de manera eficiente y sincronizada con el juego.



### 3.2.5 Display Alfanumérico de 2 Filas y 16 Columnas:

**Función:** Proporciona información visual sobre el modo de juego, el nivel del jugador y la cuenta regresiva para responder.

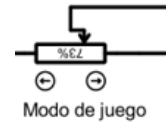
**Aplicación en el Proyecto:** El display contribuye a la interfaz de usuario, ofreciendo información contextual y facilitando la interacción jugador-dispositivo.



### 3.2.6 Potenciómetro:

Función: Permite variar la velocidad del juego ajustando la resistencia eléctrica.

Aplicación en el Proyecto: El potenciómetro añade una dimensión de personalización al juego, permitiendo a los jugadores ajustar la velocidad según sus preferencias.



Estos componentes, integrados de manera sinérgica, conforman la infraestructura electrónica necesaria para la realización eficaz del juego "Simon". Su selección y aplicación estratégica se alinean con los objetivos del proyecto, garantizando un funcionamiento coherente y confiable del sistema.

## 3.3 Funcionalidad del STM32: Papel del Microcontrolador en el Proyecto

La elección del microcontrolador STM32 en el marco de este proyecto no es fortuita, sino que responde a consideraciones estratégicas que se alinean con la complejidad y los requerimientos específicos del juego "Simon" a emular. La funcionalidad del STM32 en esta empresa es central y abarca diversas áreas cruciales para el desempeño integral del sistema electrónico propuesto.

### 3.3.1 Gestión de Periféricos y Entradas:

El STM32, con su arquitectura avanzada, facilita la gestión eficiente de los periféricos esenciales para la operación del juego "Simon". La interfaz de usuario, compuesta por LEDs, botones, display alfanumérico y potenciómetro, requiere una coordinación precisa, y el STM32 se erige como el maestro orquestador que asegura una interacción fluida entre estos componentes.

### 3.3.2 Generación de Secuencias Aleatorias:

La capacidad del STM32 para ejecutar algoritmos complejos es esencial para la generación de secuencias aleatorias de luces y sonidos, una funcionalidad crítica en la dinámica del juego "Simon". La implementación de un algoritmo eficaz de generación de patrones, coordinada por el STM32, garantiza la variabilidad y desafío necesarios para el entretenimiento del jugador.

### 3.3.3 Control de Niveles de Dificultad:

El STM32 desempeña un papel clave en la gestión de los niveles de dificultad del juego. La capacidad de ajustar parámetros como el tiempo disponible para recrear la secuencia y la duración de la iluminación de LEDs requiere una intervención precisa del microcontrolador. El STM32 proporciona la capacidad de controlar estos aspectos, adaptando la experiencia del juego a diferentes grados de habilidad.

### 3.3.4 Temporización y Validación de Respuestas:

La función de temporización, fundamental para limitar el tiempo de respuesta del jugador, y la validación de las secuencias ingresadas se encuentran dentro del dominio de responsabilidad del STM32. La gestión de tiempos y la evaluación de respuestas se ejecutan de manera eficiente gracias a las capacidades de procesamiento del microcontrolador.

## 3.4 Recursos y Herramientas Utilizados: Proteus y STM32IDE

La realización eficaz del proyecto "Simon" ha implicado la utilización de recursos y herramientas especializadas que han desempeñado un papel fundamental en las etapas de diseño y desarrollo. Entre las herramientas destacadas se encuentran Proteus y STM32IDE, ambas seleccionadas por su idoneidad y eficiencia en el contexto del proyecto.

### 3.4.1 Proteus:

Proteus, un entorno de simulación electrónico, ha sido instrumental en la etapa inicial de diseño del proyecto. Este software permite la modelización virtual de circuitos electrónicos, facilitando la visualización y evaluación de la interacción entre componentes antes de la implementación física. La simulación en Proteus ha posibilitado la verificación y validación de la lógica de juego, así como la correcta integración de los componentes electrónicos. Esto ha contribuido a la detección temprana de posibles problemas y al refinamiento de la implementación antes de la fase de prototipado.



**PROTEUS**

### 3.4.1 STM32IDE:

STM32IDE, un entorno de desarrollo integrado específico para los microcontroladores STM32, ha sido la plataforma elegida para la programación y carga de firmware en el STM32F103C8. Este entorno proporciona una interfaz intuitiva para la escritura, depuración y carga de código en el microcontrolador. La compatibilidad directa con la familia STM32 asegura una integración sin contratiempos, permitiendo un desarrollo eficiente y preciso del software necesario para la funcionalidad del juego "Simon".



La combinación estratégica de Proteus y STM32IDE ha facilitado un enfoque integral en el diseño y desarrollo de este proyecto, permitiendo una transición fluida desde la conceptualización y simulación hasta la implementación práctica en el microcontrolador STM32F103C8. La utilización de estas herramientas ha contribuido significativamente a la eficacia y robustez del proceso de desarrollo, proporcionando un marco integral para la materialización exitosa de las metas establecidas en el proyecto "Simon".

## 4. Programación

### 4.1 Flujo del Código y Estados Principales en el Loop del Main

El código del programa sigue un flujo lógico y estructurado, donde el bucle principal (while (1)) gestiona distintos estados que definen el comportamiento del juego "Simon". A continuación, se proporciona una explicación más detallada de cada uno de los seis estados principales:

#### 4.1.1 Estado 0 (Permanente):

En este estado, el programa actualiza de manera continua el reloj en el display, garantizando que la hora y la fecha se reflejen en tiempo real. Además, monitoriza la entrada del teclado para detectar acciones del usuario. Se verifica si se presionaron las teclas A, B o D. Si es así, el programa transiciona al estado correspondiente (Estados 1, 2 o 3) para ejecutar las operaciones asociadas. La tecla \* regresa al usuario al Estado 0, y la tecla # confirma cambios realizados, si los hay. Este estado sirve como el estado central de monitoreo y control del sistema.

#### 4.1.2 Estado 1 (Programación de Fecha y Hora):

Al presionar la tecla A en el Estado 0, el usuario entra en este estado, donde se muestra en el display la fecha y hora actuales. El programa permite al usuario modificar estos valores mediante la entrada desde el teclado. La tecla \* regresa al usuario al Estado 0 sin realizar cambios. La tecla # confirma los cambios y regresa al Estado 0, actualizando el reloj con los nuevos valores ingresados por el usuario.

#### 4.1.3 Estado 2 (Programación de Dificultad):

Al presionar la tecla B en el Estado 0, el usuario ingresa a este estado, donde se visualizan las opciones de dificultad en el display. Se utiliza el potenciómetro para ajustar el nivel de dificultad. La tecla \* regresa al usuario al Estado 0 sin realizar cambios. La tecla # confirma la dificultad seleccionada y regresa al Estado 0, mostrando el nivel de dificultad actualizado.

#### 4.1.4 Estado 3 (Juego "Simon" en Proceso):

Este estado inicia la ejecución del juego "Simon". El display muestra el nivel actual y genera secuencias de colores y sonidos para que el jugador las reproduzca. El usuario ingresa su respuesta desde el teclado, y el juego evalúa la precisión. Dependiendo de la precisión, el programa ajusta dinámicamente la dificultad y avanza al siguiente nivel. La transición entre niveles y estados se gestiona de manera dinámica, adaptándose al rendimiento del jugador.

#### 4.1.5 Estado 4 (Fin del Juego - GAME OVER):

Este estado se activa cuando el jugador pierde en el juego "Simon". Muestra un mensaje de GAME OVER en el display y espera la interacción del usuario. Las teclas \* o # en este estado devuelven al usuario al Estado 0, reiniciando el juego y ofreciendo la posibilidad de intentar nuevamente.

El código implementa de manera eficiente y modular cada uno de estos estados, asegurando una ejecución fluida del juego y una interacción coherente con el usuario. La estructura modular facilita futuras expansiones y mejoras en el proyecto.

## **4.2 Funciones Importantes utilizadas en los Estados Principales**

En el desarrollo del juego "Simon" en el microcontrolador STM32F103C8, diversas funciones desempeñan roles esenciales para la operatividad del sistema. A continuación, se proporciona una descripción detallada de las funciones más significativas que intervienen en los estados principales del juego:

### **4.2.1 Función Display\_DiaHora(void):**

Esta función despliega en el display la fecha, hora y día de la semana actuales. Utiliza las variables globales date, time y diasem. La lógica interna selecciona la cadena de texto correspondiente al día de la semana en base al valor de diasem y presenta la información de manera estructurada en el display.

### **4.2.2 Función Teclado(void):**

La función Teclado gestiona la entrada desde el teclado matricial. Configura las filas y columnas de la matriz para detectar qué tecla ha sido presionada. Dependiendo de la tecla presionada, asigna un valor a la variable global Tecla. La asignación de valores numéricos y especiales como Enter y Esc facilita la interacción del usuario.

### **4.2.3 Función Validador\_dia(void):**

Validador\_dia establece los límites válidos para el día del mes en función del mes y año actuales. La verificación se realiza considerando si el año es bisiesto y el mes es febrero, así como los días máximos permitidos para los demás meses. Los resultados se almacenan en las variables Limite\_dia\_decena y Limite\_dia\_unidad para su posterior uso.

### **4.2.4 Función ConversorAD(void):**

Esta función realiza la conversión analógico-digital utilizando el convertidor analógico-digital (ADC) del microcontrolador. El resultado, almacenado en AD\_RES, se utiliza para ajustar la dificultad del juego. La relación lineal entre la lectura analógica y la dificultad contribuye a la adaptabilidad del juego a diferentes condiciones.

### **4.2.5 Función actualizarReloj(void):**

actualizarReloj extrae los valores de fecha y hora almacenados en las variables date y time. Realiza validaciones para garantizar que los datos ingresados por el usuario sean coherentes. Si los datos son válidos, actualiza las variables globales dia, mes, anio, hora, minuto y segundo con los valores correspondientes.

### **4.2.6 Función generarSecuencia(uint8\_t secuencia[], uint8\_t largo):**

Genera una secuencia aleatoria de colores y asigna los valores resultantes a la matriz secuencia. La longitud de la secuencia está determinada por el parámetro largo. La generación aleatoria garantiza la diversidad y desafío en las secuencias presentadas al jugador.

### **4.2.7 Función mostrarSecuencia(const uint8\_t secuencia[], uint8\_t largo):**

La función mostrarSecuencia presenta visualmente la secuencia generada en el display. Enciende y apaga LEDs de diferentes colores y emite sonidos asociados a cada color. La velocidad de presentación se ajusta según el nivel y la dificultad del juego, proporcionando una experiencia de juego progresivamente desafiante.

#### **4.2.8 Función compararTiempo(uint32\_t tiempoInicio):**

Compara el tiempo transcurrido desde tiempoInicio con un límite de 3000 milisegundos (3 segundos). Esta función se utiliza en leerInputUsuario para determinar el tiempo disponible para que el jugador reproduzca la secuencia generada. La limitación temporal agrega presión al juego, incentivando respuestas rápidas.

#### **4.2.9 Función leerInputUsuario(uint8\_t secuenciaUsuario[], uint8\_t largo):**

Gestiona la entrada del usuario mientras se espera que reproduzca la secuencia generada. Utiliza el estado de los interruptores (Switches) para asignar valores a la matriz secuenciaUsuario, que representa la respuesta del jugador. La función se detiene si se supera el límite de tiempo de 3 segundos.

Estas funciones constituyen la columna vertebral del juego "Simon" implementado en el microcontrolador STM32F103C8, asegurando su funcionamiento, interacción y capacidad de adaptación a las acciones del usuario.

## **5. Conclusiones**

### **5.1 Verificación de los objetivos planteados**

El proyecto ha logrado cumplir de manera satisfactoria los objetivos establecidos, demostrando la viabilidad de implementar un juego "Simon" en un microcontrolador STM32F103C8. Los objetivos, detallados en la sección de introducción, incluían diseñar e implementar un juego electrónico con funcionalidades específicas. A continuación, se presenta una evaluación detallada:

#### **5.1.1 Diseño e Implementación del Juego "Simon"**

Se logró diseñar e implementar el juego "Simon" de manera completa, incorporando la generación aleatoria de secuencias, la presentación visual y sonora de las mismas, así como la interacción con el jugador a través del teclado matricial.

#### **5.1.2 Integración del Microcontrolador STM32F103C8**

Se ha demostrado con éxito la integración del microcontrolador STM32F103C8 en el desarrollo del juego. Las funciones del STM32, como el control de LEDs, el manejo del teclado matricial y la generación de sonidos, fueron implementadas de manera efectiva.

#### **5.1.3 Niveles de Dificultad y Configuraciones Adicionales**

Se cumplió con la implementación de niveles de dificultad ajustables y configuraciones adicionales a través de un menú de programación. Los usuarios tienen la capacidad de personalizar la velocidad del juego, configurar el reloj del microcontrolador y reiniciar el juego en cualquier momento.

#### **5.1.4 Validación de Respuestas y Progresión del Juego**

La validación de las respuestas de los jugadores se implementó de manera precisa, y la progresión del juego se lleva a cabo de acuerdo con el desempeño del jugador, aumentando la complejidad de las secuencias a medida que avanza.

#### **5.1.5 Capacidad de Modificar Fecha y Hora**

Se logró implementar exitosamente la capacidad de modificar la fecha y hora del reloj del microcontrolador. Esta funcionalidad proporciona flexibilidad y personalización al usuario, permitiéndole ajustar la configuración del juego según sus preferencias.

### **5.2 Puntos clave descubiertos durante el proyecto**

Durante la ejecución del proyecto, se identificaron varios hallazgos y aspectos clave que contribuyen al entendimiento global de la implementación y sus posibles áreas de mejora:

#### **5.2.1 Desafíos en la Gestión del Tiempo**

Se observó que la implementación de un temporizador para la respuesta del jugador agregó un componente adicional de desafío al juego. Sin embargo, la gestión precisa del tiempo resultó ser una tarea crucial para garantizar una experiencia de juego justa y emocionante.

### **5.2.2 Ajuste de Velocidad y Dificultad**

La variabilidad de la velocidad del juego y la dificultad proporciona una experiencia personalizable para los usuarios. Se descubrió que el equilibrio entre desafío y jugabilidad es fundamental para mantener el interés del jugador.

### **5.2.3 Interfaz de Usuario Intuitiva**

La importancia de una interfaz de usuario intuitiva se destacó durante las pruebas. Garantizar que los jugadores comprendan claramente las indicaciones visuales y sonoras contribuye significativamente a la experiencia del juego.

### **5.2.4 Comparación Simulador vs. Hardware Real**

Se observó una diferencia notable entre probar el proyecto en un simulador y en hardware real. La respuesta de los componentes físicos y la interacción del usuario pueden variar, lo que resalta la importancia de realizar pruebas en condiciones reales para una validación completa.

### **5.2.5 Consideración del Uso de Memoria**

Durante el desarrollo, se evidenció la importancia de monitorear el porcentaje de memoria utilizado en el microcontrolador. Esto es crucial para asegurar un funcionamiento estable y evitar problemas de capacidad en proyectos más complejos.

En resumen, el proyecto no solo logró cumplir con los objetivos predefinidos, sino que también proporcionó información valiosa sobre los desafíos y consideraciones clave asociados con el diseño e implementación de juegos electrónicos en un entorno de microcontroladores. Estos hallazgos proporcionan una base sólida para futuras iteraciones y mejoras en el juego "Simon" implementado en el microcontrolador STM32F103C8.



## **6. Recomendaciones**

### **6.1 Mejoras y Expansiones**

Como parte del proceso de mejora continua y expansión del proyecto, se proponen las siguientes recomendaciones:

#### **6.1.1 Almacenamiento de Puntajes en la Memoria del STM32:**

Se sugiere la implementación de un sistema de almacenamiento para los puntajes más altos obtenidos por los jugadores. Utilizando la memoria EEPROM del microcontrolador STM32F103C8, se pueden registrar y recuperar los mejores tiempos y niveles alcanzados. Esto proporcionaría una funcionalidad adicional al juego al permitir a los usuarios rastrear y competir por los mejores puntajes.

La implementación de un mecanismo eficiente de almacenamiento y recuperación de puntajes podría incluir la creación de una estructura de datos que almacene la información relevante, como el tiempo de juego, el nivel alcanzado y cualquier otra métrica pertinente. Además, se podría desarrollar una interfaz de usuario para visualizar y gestionar estos registros almacenados.

Esta mejora no solo enriquecería la experiencia del usuario al agregar un elemento competitivo, sino que también aprovecharía la capacidad de almacenamiento a largo plazo que ofrece la memoria EEPROM del microcontrolador.

#### **6.1.2 Optimización del Uso de Memoria:**

Dado que la memoria en microcontroladores es un recurso limitado, se recomienda realizar una optimización exhaustiva del código para minimizar el uso de memoria. Esto es especialmente crucial cuando se considera la posible expansión del juego con nuevas características. Evaluar y reducir el uso de variables, optimizar las funciones y considerar técnicas de programación eficientes contribuirán a una utilización más efectiva de los recursos del STM32.

#### **6.1.3 Pruebas en Hardware Real:**

Aunque el proyecto fue desarrollado y probado en un simulador, se recomienda realizar pruebas adicionales en hardware real. La transición del entorno de simulación al hardware físico puede revelar posibles desafíos y variaciones que no se detectaron durante la simulación. La verificación en un entorno de hardware real garantizará la robustez y confiabilidad del sistema.

Implementar estas recomendaciones contribuirá significativamente a la calidad, funcionalidad y eficiencia del proyecto, ofreciendo a los usuarios una experiencia más completa y mejorada.

## 7. Referencias

Durante el desarrollo de este proyecto, se consultaron diversas fuentes que proporcionaron información valiosa y orientación en áreas clave. A continuación, se detallan las citas bibliográficas utilizadas:

### 7.1 Getting started with STM32, STM32base.Org

<https://stm32-base.org/guides/getting-started.html>

### 7.2 STM32 Reference Manual. STMicroelectronics,

<https://www.st.com/>

### 7.3 Documentación de Proteus. Labcenter Electronics,

<https://www.labcenter.com/>

Estas referencias proporcionaron información fundamental sobre el microcontrolador STM32, programación de sistemas embebidos, diseño de juegos electrónicos y el uso del simulador Proteus. Se recomienda la exploración detallada de estas fuentes para obtener una comprensión más completa de los conceptos y técnicas involucradas en el desarrollo del proyecto.

## 8. Código Fuente:

A continuación se presenta el código fuente utilizado en el desarrollo de este proyecto. Este código está destinado a ser implementado en un entorno de desarrollo para microcontroladores STM32 y ha sido estructurado para facilitar la comprensión y modificación.

### 8.1 Repositorio en Github

<https://github.com/sgrofranco/STM32Simon>

Este código abarca la lógica principal del juego Simon, la interacción con el microcontrolador STM32F103C8, y las funciones esenciales para el manejo de la interfaz y la entrada del usuario.