

**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL****TAREA DE SISTEMAS EMBEBIDOS – TEÓRICO****TAREA #2**

Comunicación Entre Microcontroladores  
Atmega328p - Pic16f887

"Implementación de un Juego Serpiente Clásico de  
Nokia en Microcontroladores "

**Grupo # 3****Integrantes:**

Cristhian Daniel Bastidas García

[cdbastid@espol.edu.ec](mailto:cdbastid@espol.edu.ec)

Christopher Jeanpierre Lema Fariñas

[cjlema@espol.edu.ec](mailto:cjlema@espol.edu.ec)

Karla Priscilla López Mendoza

[klopez@espol.edu.ec](mailto:klopez@espol.edu.ec)

**Paralelo: 2****Docente:**

Ing. Solis Mesa Ronald David

**PAO II 2025-2026**

## Objetivo General

Desarrollar un sistema embebido interactivo basado en microcontroladores que implemente el clásico juego Serpiente estilo Nokia, integrando control mediante botones, visualización en matriz LED y retroalimentación sonora para ofrecer una experiencia de juego completa y escalable en diferentes niveles de dificultad.

## Objetivos Específicos

- Diseñar e implementar la interfaz de usuario mediante botones para controlar la dirección de la serpiente en la matriz LED.
- Programar la lógica del juego Serpiente con tres niveles de dificultad (fácil, medio y difícil), ajustando la velocidad y la cantidad de elementos en pantalla.
- Integrar señales sonoras mediante un buzzer para indicar eventos importantes como la comida consumida, el inicio del juego, la victoria y la derrota.
- Desarrollar funciones para la visualización de mensajes y animaciones en la matriz LED utilizando el controlador MAX7219.
- Garantizar la detección de colisiones y el manejo adecuado del crecimiento de la serpiente durante el juego.
- Optimizar el código para un funcionamiento eficiente en microcontroladores PIC y AVR, asegurando la estabilidad y respuesta en tiempo real.

## Descripción Técnica

El juego implementado es una versión embebida del clásico juego Serpiente, inspirado en el modelo de Nokia. Utiliza una matriz LED 8x8 para la visualización gráfica y un buzzer para la retroalimentación sonora. El sistema está controlado por un microcontrolador PIC16F887 y un microcontrolador AVR ATmega328P, que gestionan la lógica del juego y la interfaz de usuario.

La matriz LED 8x8, controlada mediante el MAX7219, muestra la serpiente, la comida y mensajes al usuario. La comunicación con el MAX7219 se realiza mediante transmisión serial bit a bit. Los botones de control permiten al jugador cambiar la dirección de la serpiente en las cuatro direcciones básicas, y están configurados con resistencias pull-up internas para evitar ruido. El buzzer proporciona señales sonoras para eventos como el inicio del juego, el consumo de comida, la victoria y la derrota, mejorando la experiencia interactiva.

En cuanto a la lógica del juego, la serpiente está representada por un arreglo de segmentos con coordenadas X e Y en la matriz. En cada ciclo, la cabeza se mueve en la dirección actual y el cuerpo sigue la posición anterior de la cabeza. El juego cuenta con tres niveles de dificultad: en el

nivel fácil la velocidad es lenta, con un retardo de 400 milisegundos entre movimientos y una comida en pantalla; en el nivel medio la velocidad es media, con un retardo de 200 milisegundos y una comida en pantalla; y en el nivel difícil la velocidad es rápida, con un retardo de 100 milisegundos y dos comidas simultáneas en pantalla para aumentar la dificultad.

El sistema detecta colisiones de la cabeza con el cuerpo de la serpiente, y si ocurre una colisión, se muestra un mensaje de “Game Over” y se reproduce la melodía de derrota. Al consumir la comida, la serpiente crece en longitud y se genera una nueva comida en una posición aleatoria no ocupada. La matriz LED desplaza mensajes para indicar el nivel seleccionado y mensajes finales, utilizando una tabla de caracteres para representar letras y símbolos. La matriz se actualiza continuamente para reflejar el estado actual del juego.

Las melodías específicas indican el inicio del juego, la victoria y la derrota, mientras que sonidos cortos y largos señalan eventos como la comida consumida y el fin del juego. La generación de sonidos se realiza mediante la modulación del buzzer con frecuencias definidas para cada nota musical.

## Simulación en Proteus

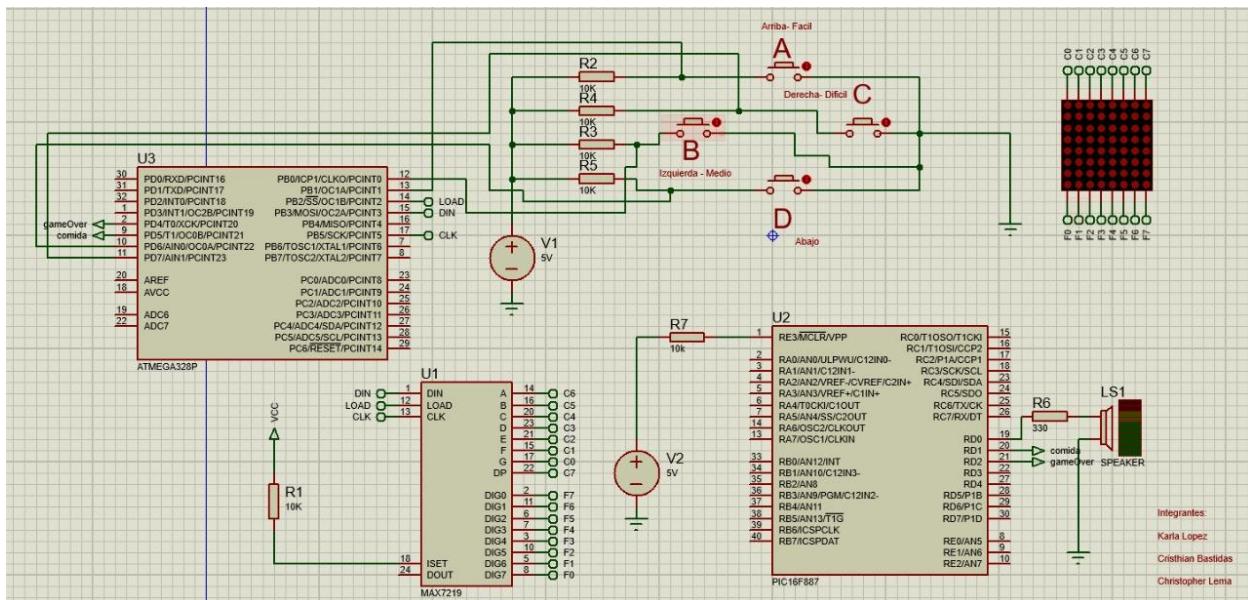


Figura 1. Simulación en Proteus

## Explicación del Código para Cada Microcontrolador

El proyecto utiliza dos microcontroladores principales: el ATmega328P y el PIC16F887, cada uno con funciones específicas para el correcto funcionamiento del juego Serpiente.

El código del ATmega328P está diseñado para manejar la interfaz visual y la interacción del

usuario en el juego Serpiente. Entre sus funciones principales se encuentra la inicialización y configuración de los pines de entrada y salida, donde se definen los pines para controlar la matriz LED y leer los botones de dirección. La función encargada de enviar datos al controlador MAX7219 transmite la información necesaria para actualizar la matriz LED, mostrando la serpiente, la comida y los mensajes en pantalla.

La lógica del juego está implementada en funciones que controlan el movimiento de la serpiente, la detección de colisiones y el crecimiento al consumir comida. Estas funciones actualizan continuamente la matriz LED para reflejar el estado actual del juego. Además, el ATmega328P detecta la pulsación de botones para cambiar la dirección de la serpiente y envía señales digitales a través de líneas específicas para indicar eventos importantes, como cuando la serpiente consume comida o cuando termina el juego. También incluye funciones para desplazar texto en la matriz LED, mostrando mensajes informativos como la selección de nivel o el fin del juego. En resumen, el ATmega328P gestiona toda la parte visual y de control del juego, asegurando que la experiencia del usuario sea fluida y responsiva.

Por otro lado, el código del PIC16F887 está enfocado en la generación de señales sonoras para complementar la experiencia del juego. Sus funciones principales definen las frecuencias de las notas musicales y controlan la duración de cada una para reproducir melodías específicas. Entre estas melodías se encuentran la de bienvenida, la de victoria y la de derrota, además de sonidos cortos para eventos como la comida consumida y sonidos largos para el fin del juego.

El PIC16F887 recibe señales digitales desde el ATmega328P a través de líneas de comunicación dedicadas. Cuando detecta la activación de estas señales, inicia la reproducción de la melodía correspondiente. Para evitar que se reproduzcan sonidos múltiples por un solo evento debido a rebotes eléctricos, el PIC implementa un mecanismo que espera a que la señal se libere antes de aceptar una nueva activación. En esencia, el PIC16F887 se encarga de la retroalimentación sonora sincronizada con los eventos del juego, trabajando en conjunto con el ATmega328P para ofrecer una experiencia completa y coordinada.

Ambos microcontroladores trabajan en conjunto para ofrecer una experiencia de juego completa, donde el ATmega328P maneja la parte visual y de control, mientras que el PIC16F887 se encarga de la retroalimentación sonora.

## **Descripción del Esquema de Comunicación entre Microcontroladores**

La comunicación entre el microcontrolador ATmega328P y el PIC16F887 se realiza mediante dos líneas digitales que permiten sincronizar eventos importantes del juego. Una de estas líneas conecta el pin PD5 del ATmega328P con el pin RD1 del PIC16F887, y se utiliza para indicar que el jugador ha consumido una comida. La otra línea conecta el pin PD4 del ATmega328P con el pin RD2 del PIC16F887, señalando el fin de la partida.

El ATmega328P activa estas líneas durante un breve período para enviar una señal de pulso que el PIC16F887 detecta y utiliza para iniciar la reproducción de la melodía correspondiente a cada evento. Esta activación momentánea funciona como una notificación clara y puntual para el PIC, que responde con la retroalimentación sonora adecuada.

Para evitar activaciones múltiples o erróneas causadas por rebotes eléctricos o interferencias, el PIC16F887 incorpora un mecanismo de protección que consiste en esperar a que la línea de señal se libere completamente antes de aceptar una nueva activación. De esta forma, se asegura que cada evento sonoro se reproduzca una única vez por cada señal recibida, garantizando una comunicación robusta y eficiente entre ambos microcontroladores.

Este esquema sencillo y directo permite una coordinación en tiempo real entre la parte visual y sonora del juego, optimizando la experiencia del usuario sin requerir protocolos complejos ni hardware adicional.

### **Enlace a un repositorio de GitHub:**

Link: [https://github.com/KarlaPriscilla/Tarea2\\_Grupo\\_3](https://github.com/KarlaPriscilla/Tarea2_Grupo_3)

### **Conclusiones**

- El desarrollo del juego Serpiente utilizando microcontroladores ATmega328P y PIC16F887 permitió integrar de manera efectiva la visualización en matriz LED y la retroalimentación sonora, logrando una experiencia interactiva completa y funcional.
- La implementación de diferentes niveles de dificultad, ajustando la velocidad y la cantidad de elementos en pantalla, demostró ser una estrategia efectiva para aumentar el desafío y la rejugabilidad del sistema embebido.
- La comunicación simple y directa entre los microcontroladores mediante líneas digitales permitió una sincronización eficiente de eventos, garantizando que las señales visuales y sonoras estén coordinadas en tiempo real sin necesidad de protocolos complejos.

### **Recomendaciones**

- Se recomienda optimizar el código para mejorar la eficiencia y reducir el consumo energético, especialmente en aplicaciones donde la autonomía sea crítica, como dispositivos portátiles o alimentados por baterías.
- Es aconsejable implementar un sistema de detección y manejo de errores más robusto, que permita identificar y corregir fallos en la comunicación entre microcontroladores o en la lectura de botones para mejorar la estabilidad del juego.
- Para futuras versiones, se sugiere ampliar la interfaz de usuario incorporando pantallas más grandes o táctiles, así como añadir efectos sonoros más variados y complejos para enriquecer la experiencia del jugador.