

TAREA DE SISTEMAS EMBEBIDOS - TEÓRICO

TAREA #3

Grupal

COMUNICACIÓN ENTRE MICROCONTROLADORES

ATMEGA328P - PIC16F887

1. Objetivo General

- Desarrollar un sistema de juego interactivo de 3 niveles de dificultad que combine salidas visuales mediante una matriz LED 8x8 controlada por el microcontrolador ATmega328P, y salidas auditivas mediante un sistema de reproducción de melodías con el microcontrolador PIC16F887, y entradas mediante pulsadores, empleando comunicación entre ambos dispositivos y simulando el funcionamiento completo en Proteus, con el propósito de reforzar habilidades en programación y control de sistemas embebidos. El sistema deberá ser programado en C para ambos microcontroladores.

2. Objetivos Específicos

- Diseñar e implementar un sistema de control de una matriz LED 8x8 con el microcontrolador ATmega328P para desplegar caracteres, símbolos y animaciones del juego.
- Programar un sistema de reproducción de melodías utilizando el PIC16F887, capaz de interpretar comandos recibidos desde el ATmega328P.
- Integrar botones físicos como entradas para interactuar con el juego (selección, acción, reinicio, etc.).
- Implementar 3 niveles de dificultad ya sea modificando la velocidad, lógica o complejidad del juego.
- Establecer un canal de comunicación entre los microcontroladores (por ejemplo, UART) que permita la sincronización entre efectos visuales y sonoros.
- Simular el juego de manera completa en el entorno Proteus, integrando los dos microcontroladores, la matriz LED y el sistema de audio.
- Documentar todo el proceso de diseño, desarrollo y simulación en un informe técnico, incluyendo repositorio con evidencias y código.

3. Tarea

Cada grupo deberá desarrollar un sistema de juego interactivo programado en lenguaje C con 3 niveles de dificultad (fácil, medio, difícil) que cambien dinámicamente algún parámetro del juego (velocidad de animaciones, número de intentos, lógica de botones, etc.), donde el ATmega328P controlará una matriz LED 8x8 para mostrar los símbolos y dinámicas del juego; además de utilizar pulsadores como entradas que permitan al usuario interactuar con el juego, mientras que el PIC16F887 será responsable de reproducir melodías relacionadas con los eventos del juego (inicio, victoria, error, etc.). Ambos microcontroladores deben estar conectados mediante comunicación serial para intercambiar datos. El sistema deberá ser simulado y validado completamente en Proteus.

4. Instrucciones

a) Simulación en Proteus:

Diseñar el circuito completo en Proteus, incluyendo:

- ATmega328P
- PIC16F887
- Matriz LED 8x8 (puede usarse MAX7219 o conexión directa con multiplexado)
- Pulsadores conectados al ATmega328P
- Buzzer o altavoz controlado por el PIC
- Conexión serial entre microcontroladores

b) Lenguaje y Entorno de Desarrollo:

- **ATmega328P:** Programado en C con Visual Studio Code + PlatformIO.
- **PIC16F887:** Programado en C con MikroC for PIC.

c) Matriz LED 8x8 con ATmega328P:

- El programa debe implementar funciones para mostrar en la matriz letras, números o símbolos relacionados al juego.
- Implementar efectos de desplazamiento o animación.
- El comportamiento visual debe estar vinculado a los eventos del juego y sincronizado con los sonidos generados por el PIC16F887.

d) Sistema de Reproducción de Melodías:

- Programar el PIC16F887 para que reciba comandos por UART y, según estos, reproduzca melodías con el buzzer.
- Las melodías debe estar vinculado a los eventos del juego y sincronizado con los efectos visuales de la matriz LED 8x8 controlados por el ATmega328P.

e) Uso de Pulsadores:

- Implementar la lectura de pulsadores físicos desde el ATmega328P como entradas.
- Los botones deben controlar el flujo del juego (iniciar, seleccionar, reiniciar, etc.).
- El sistema debe ser capaz de actuar según la interacción del usuario.

f) Niveles del juego:

- Nivel 1 (fácil): menor velocidad o menor cantidad de tareas para el usuario.
- Nivel 2 (medio): lógica estándar del juego.
- Nivel 3 (difícil): mayor velocidad, tiempo limitado o menor margen de error.
- Visualizar en la matriz LED el nivel actual brevemente al cambiarlo (por ejemplo, mostrar L1, L2, L3).

5. Entregables

Cada grupo deberá entregar un archivo PDF con el siguiente contenido:

- Portada (nombres, paralelo, docente, nombre del proyecto).
- Objetivo general y específicos (en base al juego que desarrolló el grupo).
- Descripción técnica completa del juego (incluyendo una explicación detallada de los niveles).
- Capturas de la simulación en Proteus.

- Explicación del código para cada microcontrolador (ATmega328P y PIC16F887).
- Descripción del esquema de comunicación entre microcontroladores.
- Enlace a un repositorio de GitHub que contenga:
 - Carpeta /ATmega-328P/ con el código fuente en C.
 - Carpeta /PIC16F887/ con el código fuente en MikroC.
 - Carpeta /Proteus/ con el archivo .pdsprj y componentes usados.
 - Carpeta /Video/ con un video mostrando el funcionamiento de la simulación.
 - Archivo PDF final de la tarea.
- Conclusiones y recomendaciones.



File/Folder	Commit Message	Time
ATmega-328P	Update schematic and codes for PIC16F887 and ATmega...	8 months ago
PIC16F887	Update schematic and codes for PIC16F887 and ATmega...	8 months ago
Proteus	Update schematic and video	8 months ago
Video	Update schematic and video	8 months ago
Tarea Taller Grupal G2.pdf	Document	8 months ago

Figura 1. Ejemplo repositorio de GitHub

Nota:

Cada integrante del grupo deberá realizar al menos un *commit* en el repositorio de GitHub. Esta medida se implementa con el fin de verificar la participación activa y colaboración de todos los miembros del equipo en el desarrollo de la tarea.

Se revisará el historial de *commits* como evidencia del trabajo colaborativo. Los grupos que no cumplan con este requisito podrían ser penalizados en la calificación final del trabajo.

6. Criterios de Evaluación

Criterio	Puntaje
Simulación funcional en Proteus	10
Control de matriz LED 8x8 con ATmega328P	15
Comunicación serial entre ATmega328P y PIC16F887	15
Reproducción de melodías en PIC16F887	15
Integración de pulsadores funcionales	5
Implementación de 3 niveles en el juego	5
Repositorio GitHub (mínimo 1 <i>commit</i> por integrante)	10
Informe final PDF	15
Video explicativo del funcionamiento	10
Total	100