

INGENIERÍA ELECTRÓNICA

Estructura básica de programación usando un sistema microcontrolado: Juego de Ping-Pong

ELECTRONIC ENGINEERING

Basic programming structure using a microcontroller system: Ping-Pong game

Andrés F. Restrepo-Alvarez*§, Diego Fernando-Navas*

**Universidad del Valle, Santiago de Cali, Colombia*

§ andres.restrepo@correounivalle.edu.co

diego.navas@correounivalle.edu.co

Recibido: 5 de Julio de 2012 - Aceptado: Enero 30 de 2014

Resumen

En este trabajo se presenta el diseño e implementación de un juego de ping-pong sobre una matriz de LED's 7x5 basado en un sistema microcontrolado. Su ejecución involucra todos los procesos típicos en la lógica del desarrollo de software, lo que hace que esta aplicación sea considerada como una herramienta pedagógica para la enseñanza de contenidos básicos de programación. Se presenta el diseño desde la óptica de la ingeniería, se detalla el diagrama de las diferentes etapas del sistema y la estructura del algoritmo implementado en el microcontrolador. La aplicación permite incrementar la complejidad de programación conforme aumentan las condiciones y restricciones de funcionamiento, generando mayor capacidad de análisis y alternativas de solución.

Palabras clave: *Herramienta pedagógica, juego de Ping-Pong, programación, sistema microcontrolado.*

Abstract

In this paper is presented the design and implementation of a ping-pong game on a 7x5 LED matrix based on a microcontroller system. Its implementation involves all processes in typical software development logic, which makes this application to be considered as a pedagogical tool for teaching basic programming content. The design from the perspective of engineering is presented, furthermore is detailed the diagram of the different stages of the system and the structure of the algorithm implemented in the microcontroller. The application allows increase the complexity of programming in so far that increase the conditions and restrictions of operation, generating greater capacity for analysis and alternative solutions.

Keywords: *Pedagogical tool, Ping-Pong game, programming, microcontroller system.*

1. Introducción

Los microprocesadores tienen diversos campos de aplicación, desde grandes sistemas de control industrial (Rahman *et al.*, 2006, Huangsheng *et al.*, 2007), dispositivos de medición (Bucci *et al.*, 2006, Zhang, 2010), hasta pequeños artículos de entretenimiento (Mahmood *et al.*, 2011). Enfocándose en la última aplicación, se diseñó e implementó, tomando como referente resultados positivos obtenidos previamente en otros estudios (Debusse *et al.*, 2008), un juego de ping-pong simple sobre una matriz de LED's 7x5 (filas x columnas) como una estrategia de aprendizaje basada en el juego (Pao-Nan *et al.*, 2011), que sirve como apoyo para mejorar las habilidades de programación de los estudiantes de diferentes áreas de estudio, en especial, estudiantes universitarios de ingeniería de primeros semestres. En este trabajo se presenta el diseño e implementación del sistema microcontrolado desde la óptica de la ingeniería; la evaluación del sistema como estrategia de aprendizaje está actualmente en la etapa preliminar donde se definen los indicadores y los participantes, los resultados se presentarán en publicaciones posteriores.

El desarrollo es práctico, modular y no solo es divertido, sino que dado que involucra una etapa de solución del problema e implementación (Ismail *et al.* 2010), éste puede ser usado como herramienta pedagógica para la enseñanza de contenidos básicos de programación (Caro *et al.*, 2011, Chao *et al.*, 2010). La aplicación desarrollada

permite modificar su estructura y complejidad dependiendo de las condiciones y restricciones que le sean asignadas, utilizando e interactuando con algunos módulos que se encuentran embebidos comúnmente en un microcontrolador (Atmel Corporation, 1999), como por ejemplo, el manejo de interrupciones, puertos de entrada/salida (E/S), temporizadores/contadores, entre otros.

2. Estructura general del sistema planteado

Inicialmente se definen algunas tareas que tienen como objetivo plantear una solución al problema de diseño abordado, estas tareas son: (1) definición de entradas y salidas del sistema completo y consideración de etapas futuras; (2) construcción de etapa de acople entre los diferentes componentes que van conectados al microcontrolador y consideración de etapas futuras; (3) acople entre programación (software) e implementación física (Hardware); (4) Uso de herramientas de emulación para la verificación del correcto funcionamiento (verificación de resultados esperados).

El diagrama general del juego se muestra en la Figura 1. El juego consta de: (1) dos jugadores que por medio de un control manual controlan el movimiento de las raquetas (izquierda, derecha): Ellos se ubican en la primer y última columna representados con tres LED's encendidos permanentemente que simulan las raquetas; (2) un interruptor que define el inicio del juego o la reanudación: Para dar inicio al juego se presiona el pulsador "Inicio", momento en el cual comienza

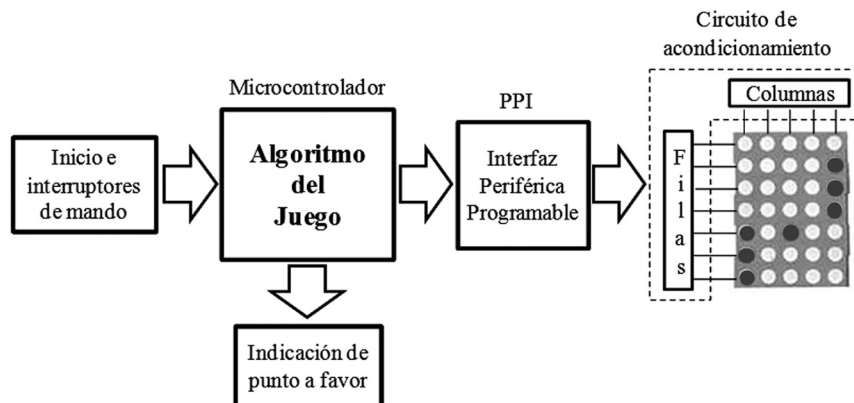


Figura 1. Diagrama general del sistema implementado.

el movimiento de la pelota. Después de obtener un punto se debe presionar nuevamente “Inicio” para reanudar el juego; (3) la matriz de LED's donde se desarrolla el juego: El juego se visualiza en una matriz de LED's de 7x5 que constituye el campo de juego por donde se desplaza la pelota con una componente de aleatoriedad en su trayectoria (la pelota rebota en los lados superior e inferior de la cancha), su desplazamiento se simula por medio del encendido secuencial de un led sobre la matriz; (4) un diodo emisor de luz para la indicación de un punto a favor: Por fuera del campo de juego, se tiene un diodo led para cada jugador. Este indica cuando se presenta un punto a favor el cual se logra cuando la raqueta del oponente no intercepta la trayectoria de la pelota.

2.1 Definición de entradas y salidas del microcontrolador

En términos generales el juego consta de dos jugadores que por medio de un control manual cambian el movimiento de las raquetas (izquierda, derecha), un interruptor que define el inicio del juego o la reanudación, la matriz de LED's donde se desarrolla el juego y un diodo emisor de luz para la indicación de un punto a favor.

Las entradas definidas para el sistema son: 2 controles manuales (2 bits c/u), inicio o reanudación del juego (1 bit) e interrupciones para detección de movimiento en los jugadores (2 bits). Por su parte, las salidas definidas son: señales para visualización en la matriz de LED's (7 bits para las filas y 3 bits para las columnas) e indicación de punto a favor (2 bits).

3. Diseño e implementación del sistema

El microcontrolador seleccionada para la implementación es el AT89C52 de Atmel (Atmel Corporation, 1999), operando con un oscilador de 24 MHz para obtener un ciclo de máquina de 0,5 μ sg. En la definición de las entradas se especifica el uso de las interrupciones externas del microcontrolador INT0 e INT1 para los jugadores, ambas interrupciones activadas por flanco descendente y con el objetivo de detectar la acción de desplazamiento de las raquetas de cada jugador.

Dadas las limitaciones en la capacidad de manejo de señales digitales de E/S de los microcontroladores de bajo costo como el seleccionado, se requiere el uso de un dispositivo adicional que tenga la capacidad de manejar un mayor número de señales. El dispositivo que permite realizar esta mejora conservando el uso del microcontrolador seleccionado es la interfaz periférica programable (PPI) 82C55A de Intel (Intel, 1995). Este dispositivo se conecta con el puerto cero (P0) del microcontrolador, el cual se configura como puerto de salida del sistema. A continuación se relacionan las entradas y salidas con los pines del microcontrolador y de la PPI: jugador 1 (2 bits de entrada; arriba P1.0 y abajo P1.1; interrupción /INT0), jugador 2 (2 bits de entrada; arriba P1.2 y abajo P1.3 del microcontrolador; interrupción /INT1), inicio o reanudación de juego (1 bit de entrada P1.7 del microcontrolador), decodificación de columnas de la matriz (3 bits de salida; PA2-PA0 del puerto A de la PPI), filas de la matriz (7 bits de salida; PB6-PB0 del puerto B de la PPI) y finalmente la indicación de punto a favor (2 bits de salida; PC7 y PC6 del puerto C de la PPI)

El sistema constantemente se encuentra mostrando las posiciones de las raquetas y la pelota sobre la matriz de LED's, se debe considerar también que la pelota rebota en los lados superior e inferior de la cancha y la única forma de conseguir un punto es que la raqueta del oponente no intercepte la pelota.

La PPI se codifica a partir de la dirección 4000H, en esta se asigna el puerto A, en 4001H el puerto B, el puerto C en 4002H, y palabra de control en la dirección 4003H, la PPI se configura en este caso para que todos puertos queden configurados como salida.

3.1 Barrido de columnas para visualizar los jugadores y la pelota

Para encender un conjunto de LED's sobre la matriz se debe hacer un “barrido de columnas”, esto quiere decir habilitar secuencialmente cada columna (PA2 – PA0, puerto A de la PPI hacia el decodificador de 3 a 8) al mismo tiempo que se pone un dato en las 7 líneas de las filas correspondientes (PB6 – PB0, puerto B de la PPI), si la columna está habilitada y

se pone un “0” sobre una de las filas, se enciende el led correspondiente, la distribución de filas y columnas se puede ver en la Figura 2. Se tiene con lo anterior, que cuando se habilita la columna 1, se pone sobre las filas el dato correspondiente a la posición del jugador 1, las columnas 2,3 y 4 son para el desplazamiento de la pelota y por último en la columna 5 se pone el dato correspondiente a la posición del jugador 2.

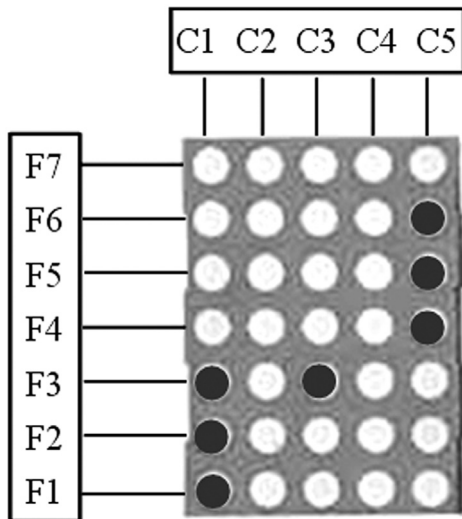


Figura 2. Distribución de filas y columnas de la matriz 7x5.

La habilitación de las columnas se realiza por medio de un decodificador de 3 a 8 conectado a los bits menos significativos del puerto A de la PPI, y las filas se conectan directamente en los 7 bits menos significativos del puerto B de la PPI. Por otra parte la indicación de los puntos a favor se visualiza en

leds colocados en el puerto C, PC7 y PC6 para el jugador 1 y 2 respectivamente, el diagrama completo del esquema de conexión de puertos en la PPI se muestra en la Figura 3. Para visualizar correctamente el juego sobre la matriz, se debe asegurar una suficiente circulación de corriente en los leds que garantice una buena intensidad de luz. Para ello y para la protección de los dispositivos electrónicos, se utilizaron buffers entre las salidas de la PPI y la matriz de LED's. Para las columnas se empleó un buffer inversor 74LS240 y para las filas un buffer no inversor 74LS244.

El programa principal se encarga de hacer constantemente la activación secuencial de cada una de las columnas de la matriz con su respectivo dato en las filas una vez iniciado el juego, esta acción se realiza a una alta velocidad para crear el efecto de encendido de todas las columnas al tiempo. Los parámetros críticos a tener en cuenta en la evolución del programa son la posición de la pelota y la posición de los jugadores, esta última se actualiza a través de la atención a la interrupción de los controles de mando de cada jugador.

3.2 Lectura de los controles de posición de un jugador

El control de posición de cada jugador consta de dos pulsadores (arriba o abajo), los pines conectados a estos pulsadores (P1.0, P1.1 para el jugador 1) sin que sean presionados tienen normalmente un estado lógico “1”, usando una compuerta AND

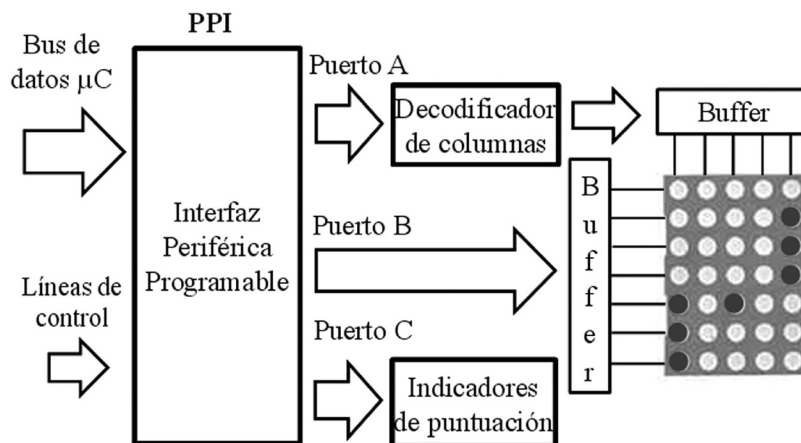


Figura 3. Esquema de conexión de puertos en la PPI.

cuando se pulsa cualquiera de ellos, aparece una transición de 1 a 0 (flanco descendente) en la salida de la compuerta, este flanco descendente se usa para generar la interrupción al microcontrolador (INT0 para el control del jugador 1, e INT1 para el control del jugador 2).

Cuando uno de los pulsadores se presiona, inmediatamente el microcontrolador atiende la interrupción solicitada, y la Rutina de Servicio de Interrupciones (ISR) respectiva se encarga de leer los bits de P1 y así identificar cuál de los pulsadores fue el que se presionó. El esquema del control de posición del jugador 1 se muestra en la Figura 4. Este esquema es similar para el control de posición del jugador 2, solamente que se utiliza la interrupción INT1 y los pines P1.2 y P1.3 del microcontrolador.

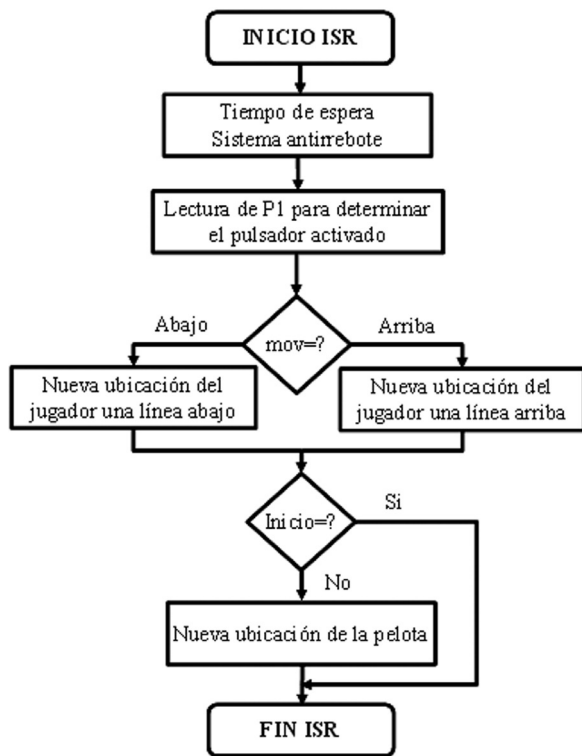


Figura 4. Diagrama de flujo algoritmo de control de posición de jugadores.

Para la lectura de los estados del control manual se debe considerar que durante el cierre de un interruptor se produce una oscilación producto de la acción mecánica en sus contactos que no se puede evitar y que genera estados no deseados en

el sistema (esto que en algunas aplicaciones de electricidad tiene una importancia relativa, cuando se trata de electrónica digital es un problema muy grave). Se busca obtener un único valor estable del estado de los pulsadores ya que normalmente cuando se presiona un pulsador se produce un número indeterminado de pulsos que pueden ser considerados como datos válidos. Para eliminar este problema, existen circuitos antirrebote que pueden ser implementados por hardware (una configuración RC o el uso de compuertas lógicas) o por software (algoritmo de retardo o de promedio). Para este diseño se realizó un sistema antirrebote por software, que consiste básicamente en crear un tiempo de espera para que la señal se estabilice y evitar los cambios indeseados al momento de realizar el pulso.

En la Figura 4 se muestra el diagrama de flujo de la ISR de la interrupción externa 0 que corresponde al jugador 1, este diagrama es similar para la interrupción externa 1 que corresponde al jugador 2. La posición de los jugadores se almacenan en las variables "posicionjug1" y "posicionjug2" respectivamente, estas variables se cargan inicialmente con "11100011" (E3H), los ceros en el dato indican cuales son los leds encendidos. Como se mencionó anteriormente se toman los 7 bits menos significativos para la visualización en la matriz, esto quiere decir que la posición del jugador 1 y 2 inicia en la mitad de las filas y dependiendo de la solicitud de los pulsadores, el dato almacenado en la variable se desplaza a la izquierda o a la derecha. De igual forma la posición de la pelota se almacena en la variable "posicionpelota" y se carga con "11110111" (F7H), el cero en el bit 3 significa que la pelota inicia ubicada en la mitad de las filas.

Dentro de esta subrutina de interrupción se tiene en cuenta la condición de que si no se ha pulsado "Inicio" la pelota también se desplaza junto con el jugador hasta que se dé comienzo al juego.

3.3 Desplazamiento y selección de trayectorias para la pelota

Uno de los puntos importantes a determinar es la velocidad de desplazamiento de la pelota, esta velocidad debe ser mucho menor que la velocidad

en que se hace el barrido de las columnas de la matriz. Teniendo en cuenta esto, se creó la variable “avance” que según su contenido, determina la velocidad en que la pelota avanza de columna en columna, si se desea un avance más lento solamente basta con aumentar el valor de comparación de la variable (avance) para generar un avance más lento.

La trayectoria que sigue la pelota se elige de forma aleatoria de un grupo de trayectorias previamente establecidas por medio del programa. Usando el parámetro rand() se crea un número aleatorio entre 0 y 10, y dependiendo de este, se realiza la selección de la trayectoria a seguir. En esta aplicación se definen básicamente tres trayectorias, las cuales se describen a continuación: trayectoria 0 (Desplazamiento en línea recta horizontal con pendiente igual a cero), trayectoria 1 (Desplazamiento hacia arriba cuando la pelota avanza de una columna a otra) y trayectoria 2 (Desplazamiento hacia abajo cuando la pelota avanza de una columna a otra).

Cuando la pelota avanza (por ejemplo de izquierda a derecha) y llega al contacto del otro jugador, se realiza de nuevo el llamado de la subrutina que

define aleatoriamente una nueva trayectoria para el avance en el otro sentido.

En la Figura 5 se muestra el diagrama de flujo de la definición de trayectoria de la pelota, esta asignación se realiza de manera aleatoria considerando un conjunto de tres posibilidades, en el inicio del algoritmo se hace la aclaración de que se debe haber iniciado el juego para que se pueda ejecutar este algoritmo.

3.4 Determinar punto a favor

Cuando se termina un avance de la pelota ya sea de izquierda a derecha o de derecha a izquierda, se verifica que la raqueta del jugador que recibe logre interceptar el avance de la pelota, conociendo que las variables que almacena la posición de los jugadores son “posicionjug1” y “posicionjug2” respectivamente y que la posición de la pelota se almacena en “posicionpelota”, se realiza una operación lógica de comparación para determinar si hay punto o no.

A manera de ilustración, supóngase que la pelota avanza en la trayectoria 0 (línea recta

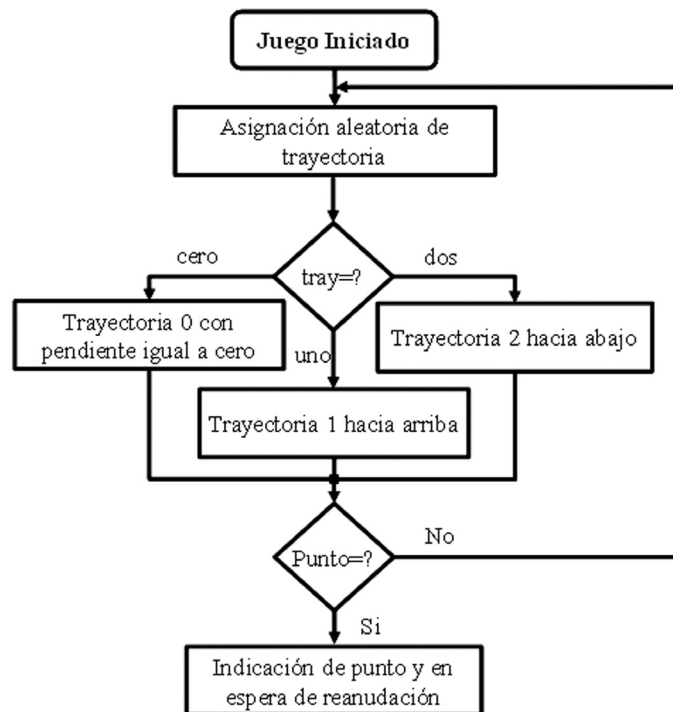


Figura 5. Diagrama de flujo definición de trayectoria de la pelota.

horizontal) de izquierda a derecha y que las variables de posición del jugador 2 y de la pelota son $posicionjug2 = "11100011"$ y $posicionpelota = "11111011"$ respectivamente. Para determinar si hay punto o no, se realiza la operación lógica OR entre las dos variables, en este caso el resultado es diferente a FFH por lo que se determina que no hay puntuación y que la pelota será devuelta. En caso contrario la operación lógica daría como resultado FFH lo cual significa que el jugador 2 no alcanzó a interceptar la pelota y por consiguiente será punto a favor para el jugador 1.

Cuando ocurre puntuación se indica sobre el led respectivo y la posición del jugador ganador al igual que la pelota se ubica en el centro de las filas, el jugador que va a sacar puede moverse junto con la pelota hasta que se reanude el juego por medio de la señal de "Inicio".

En el caso de que la trayectoria sea diferente a la línea recta horizontal (trayectoria 0), es decir que haya desplazamiento vertical en la posición de la pelota (trayectoria 1 o 2), se debe comparar la posición del jugador con la siguiente posición que ocuparía la pelota según sea la dirección (arriba o abajo) con la que haya llegado a la columna donde se ubica la raqueta. En la Figura 6 se puede ver en forma general el diagrama de flujo planteado del algoritmo del juego.

Por otra parte, la implementación física del sistema propuesto se muestra en la Figura 7, donde se puede identificar en la parte baja los mandos de los jugadores, en la parte superior la matriz de LED's donde se desarrolla el juego, los leds que indican el punto a favor, el microcontrolador y los dispositivos de acople del sistema.

4. Resultados y discusión

La definición de las condiciones de funcionamiento de cualquier aplicación, es una actividad fundamental en el momento de formular problemas. Una buena descripción del problema permite un mejor planteamiento de una posible solución. Considerando las diferentes condiciones de funcionamiento de la aplicación, se ejecutan

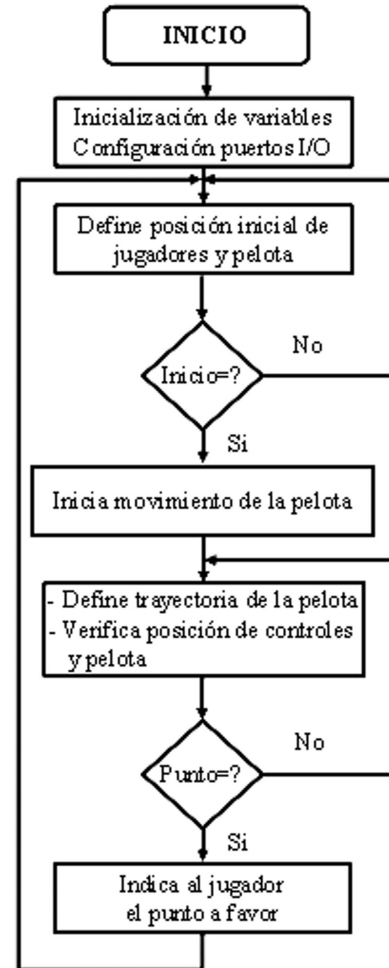


Figura 6. Diagrama de flujo general del algoritmo del juego

cada una de las tareas planteadas inicialmente, para obtener finalmente el juego de ping-pong entre dos competidores, con indicación de punto a favor y selección aleatoria de tres diferentes trayectorias de la pelota.

Durante la etapa de desarrollo de la aplicación, se realizó el proceso de verificación a través de simulaciones usando el software Proteus, este programa permite emular el comportamiento del microcontrolador interactuando con los demás componentes sobre el mismo espacio de trabajo, es decir que permite evaluar la respuesta del sistema completo, el diagrama esquemático empleado para la simulación se muestra en la Figura 8. Este tipo de herramientas software brindan una mayor facilidad en el proceso

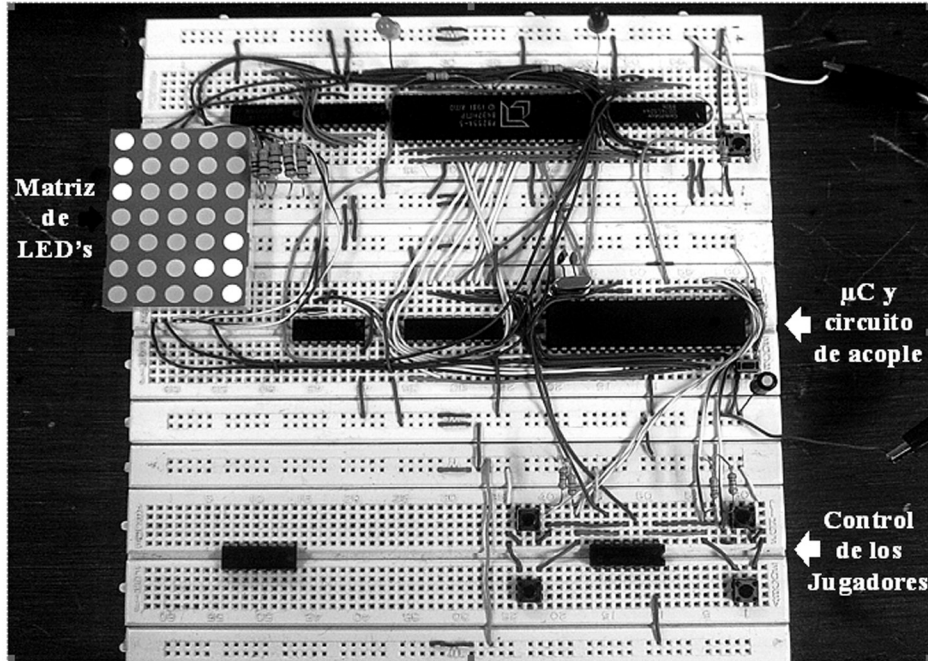


Figura 7. Implementación física del juego de ping-pong.

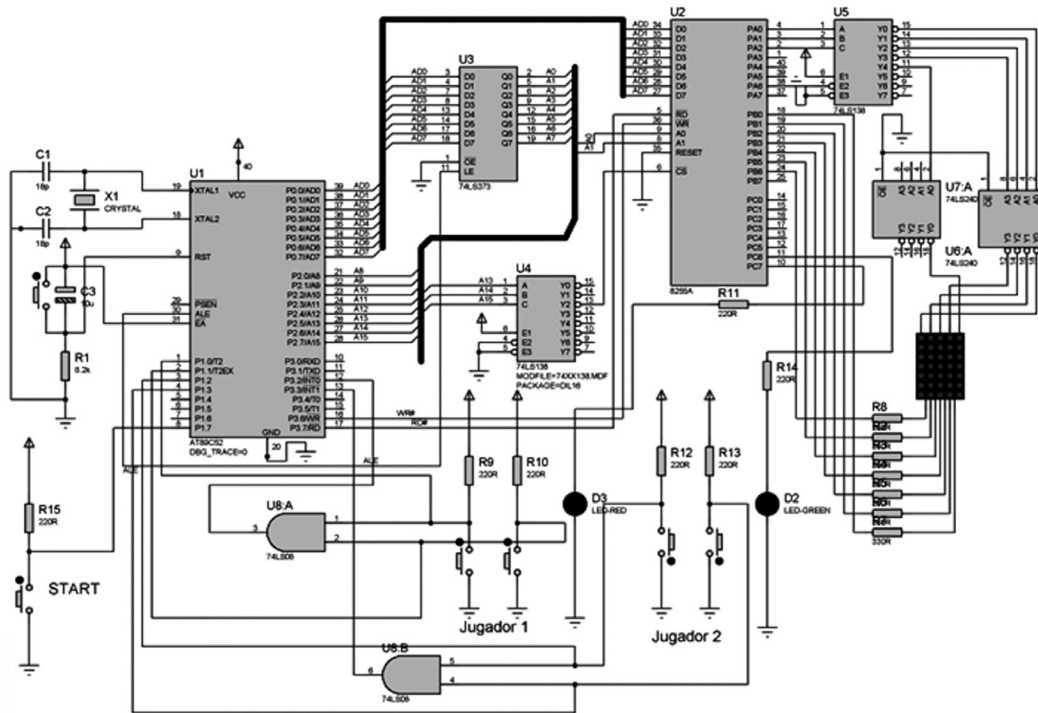


Figura 8. Diagrama esquemático implementado en Proteus.

de diseño y programación para una posterior implementación, ya que permite considerar múltiples restricciones durante la etapa de diseño tanto de hardware como de software según los

requerimientos, dando lugar al desarrollo de múltiples soluciones que depende básicamente de la destreza del programador y del nivel de complejidad impuesto.

Los resultados obtenidos en la simulación fueron validados en el funcionamiento experimental operando la aplicación y verificando el cumplimiento de las condiciones de juego planteadas (punto a favor, trayectorias, etc).

Uno de los mayores inconvenientes que se presentaron fue en la definición de las trayectorias 1 y 2, ya que se deben tomar muchas consideraciones, por ejemplo en las filas de los bordes superior e inferior se debe simular un rebote, esto requiere cambiar la trayectoria de la pelota de manera consecuente para evitar que la pelota se salga del campo de juego. Otro aspecto importante fue tomar tiempos diferentes, uno para hacer el barrido sobre la matriz y otro totalmente diferente para el avance de la pelota, con estos dos problemas solucionados el problema se simplifica a manipular el algoritmo para cumplir con las restricciones de funcionamiento.

Finalmente este tipo de desarrollos permite que el planteamiento y solución de problemas de Ingeniería sea usado como estrategia pedagógica no solo en el área de Ingeniería sino en diferentes campos de estudio. Además, este tipo de aplicaciones con sistemas microcontrolados posibilita la implementación hardware, permitiendo no solo mejorar habilidades de programación (como en el caso de estudiantes de ciencias de la computación), sino también mejorar habilidades de implementación hardware (como en el caso de estudiantes de áreas afines a la electrónica, mecatrónica, etc.).

5. Conclusiones

El uso de las interrupciones permite al microcontrolador asignar prioridades en las actividades y reducir así el impacto que las transferencias de E/S tienen en el sistema, en este caso el programa principal se encarga del muestreo permanente de las posiciones de los jugadores y la pelota, dejando en manos de las interrupciones la acción de mando de las raquetas de ambos jugadores. Esto también permite crear una estructura simplificada de programación como estrategia pedagógica basada en la separación y

asignación de actividades para la solución de un problema en particular.

El uso de las herramientas de software en este caso Proteus permite tener una perspectiva muy confiable sobre la funcionalidad del sistema, brindando una mayor facilidad en el proceso pedagógico de la enseñanza y aplicación de conceptos básicos de diseño y programación para una posterior implementación software y hardware. Además, posibilita abordar problemas de diseño más complejos, como por ejemplo, la definición de las trayectorias de la pelota haciendo que el juego sea lo más real posible.

6. Referencias bibliográficas

ATMEL Corporation (1999). *8-bit Microcontroller with 8K Bytes Flash AT89C52*. [Consulted March 2011]. Available in: http://www.atmel.com/dyn/resources/prod_documents/doc0313.pdf

Bucci, G., Ciancetta, F., Fiorucci, E., Manno, R., & Tedeschini, L. (2006). *A Microcontroller System for Human Mechanical Power Measurement during Squat Jumps*. In Proceedings of the IEEE Instrumentation and Measurement Technology Conference (IMTC), Sorrento, Italy, p. 2215-2220.

Caro, E. O., Prieto, M. M., & Monroy, M. N. (2011). *Tecnologías de la Información y la Comunicación, aprendizaje significativo y uso de mapas conceptuales como estrategia pedagógica para la enseñanza de contenidos básicos de la programación de computadores en lenguaje Java*. Encuentro Nacional de Investigación y Desarrollo (ENID), Bogotá D.C., Colombia. Editorial Universidad Nacional de Colombia.

Chao, M., Qingli, L., Zhongyuan, L., & Yu, J. (2010). *Low cost AVR microcontroller development kit for undergraduate laboratory and take-home pedagogies*. In 2nd International Conference on Education Technology and Computer (ICETC), Shanghai, China, p. V1-35 – V1-38.

Debusse, J. C. W., & Stiller, T. (2008). *Technologies and Strategies for Integrating Object-Oriented*

Analysis and Design Education with Programming. In 19th Australian Conference on Software Engineering (ASWEC), Perth, Western Australia, Australia, p. 97-103.

Huangsheng, X., King, K., & Jani, Y. (2007). *High Performance DC Chopper Speed and Current Control of Universal Motors Using a Microcontroller.* In Conference Record of the IEEE Industry Applications Conference. 42nd Industry Application Society (IAS) Annual Meeting, New Orleans, USA, p. 701-705.

INTEL (1995). *82C55A CMOS Programmable Peripheral Interface.* [Consulted March 2011]. Available in: <http://download.intel.com/design/archives/periphrl/docs/23125604.pdf>

Ismail, M. N., Ngah, N. A., & Umar, I. N. (2010). Instructional strategy in the teaching of computer programming: a need assessment analyses. *Turkish Online Journal of Educational Technology (TOJET)* 9(2), 125-131.

Mahmood, N. H., Salahuddin, Che Ku Long, C. K. M., Ghani, I. A., & Sudirman, R. (2011). *Low cost electronic chess set for chess tournament.* In the IEEE 7th International Colloquium on Signal Processing and its Applications (CSPA), Pulau Pinang Penang, Malaysia, p. 123-126.

Pao-Nan, C., & Hsi-Chi, H. (2011). An alternative learning strategy to support engineering students programming skills: a case study. *Global Journal of Engineering Education* 13(1), 6-11.

Rahman, K. M., & Choudhury, M. A. (2006). *Dead-Time Compensated Pulsewidth Modulator for a 3-Phase VSI Implemented with an AT89C52 Microcontroller.* In International Conference on Electrical and Computer Engineering (ICECE), Dhaka, Bangladesh, p. 466-472.

Zhang, Q. (2010). *Design of Digital Thermometer Based on AT89C52 Single Chip Microcontroller.* In International Conference on Electrical and Control Engineering (ICECE), Wuhan, China, p. 725-727.