



# INFORME DE LABORATORIO 2

**Autores:** *María Del Mar Arbeláez Sandoval, Julián Mauricio Sánchez Ceballos*

*Laboratorio de Electrónica Digital 3  
Departamento de Ingeniería Electrónica y de Telecomunicaciones  
Universidad de Antioquia*

## Resumen

En este laboratorio, se implementó un juego de tres en línea (tic-tac-toe) utilizando una Raspberry Pi Pico programada en lenguaje C, con el propósito de estudiar y manipular GPIOs para el control de entradas y salidas digitales, como botones y LEDs. Se diseñó un sistema embebido que permite a dos jugadores interactuar en tiempo real mediante interruptores para ingresar posiciones en un tablero de 3x3 LEDs bicolor (rojo y verde). El sistema incluye pulsadores para verificar las jugadas y gestionar la lógica del juego. A través de esta práctica, se profundizó en conceptos clave de electrónica y programación de sistemas embebidos, como la gestión eficiente de recursos limitados y el uso de interrupciones básicas como el callback y polling.

**Palabras clave:** sistema embebido, Raspberry Pi Pico, firmware, software, GPIOs, C.

## INTRODUCCIÓN

Para este laboratorio, se implementó un juego de tres en línea mediante el uso de una Raspberry Pi Pico, programada en C mediante el uso del SDK, esto con el objetivo de familiarizarse con el manejo de GPIOs (General Purpose Input/Output) de la Raspberry Pi Pico para controlar entradas y salidas digitales, como botones, LEDs.

Para esto es necesario diseñar un sistema embebido capaz de gestionar interacciones de dos jugadores en tiempo real mediante un arreglo de interruptores por el cual mediante cada jugador ingresa la posición, un tablero de 3x3 LEDs bicolor para indicar el estado del juego, un

pulsador de verificación. Junto con el diseño de toda la lógica necesaria para gestionar el juego.

Este tipo de laboratorios permiten entender la importancia de la configuración correcta y eficaz en sistemas embebidos y como se pueden aplicar conocimientos de lógica digital y programación para implementar soluciones en el mundo real, tales como sistemas de control o juegos.

La implementación del juego de tres en línea permitió experimentar con conceptos claves de la ingeniería electrónica (como lo son la administración de potencia usando transistores), como la gestión eficiente de recursos limitados (GPIOs), el uso de interrupciones y polling para la captura de eventos y la implementación de algoritmos basados en lógica condicional y de estado.

## ESQUEMÁTICO

En primer lugar se diseña el hardware del tablero y los diferentes pulsadores y botones del sistema, el esquemático del tablero

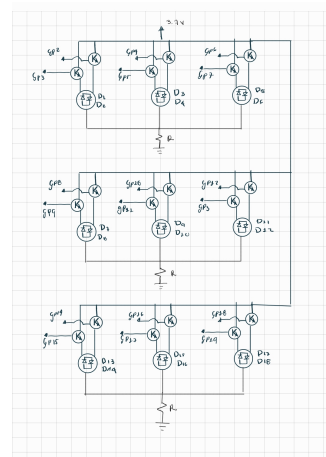


Figura 0-1: Esquemático del tablero de LEDs.

Para la polarización de este tablero se optó por una fuente separada de 3.7V con tierras debidamente acopladas con la Raspberry Pi Pico, aunque en el SDK permite configurar el pad de los GPIOs para que entreguen la corriente suficiente para los LEDs, la justificación de elegir esta configuración y no la del SDK es evitar cualquier accidente que pueda ocurrir y hacer daños tanto en el computador con el cual se programa la tarjeta y la misma tarjeta de desarrollo.

Seguidamente, para la entrada de las posiciones, indicador de turno y la confirmación del movimiento, el esquemático de este apartado se muestra a continuación:

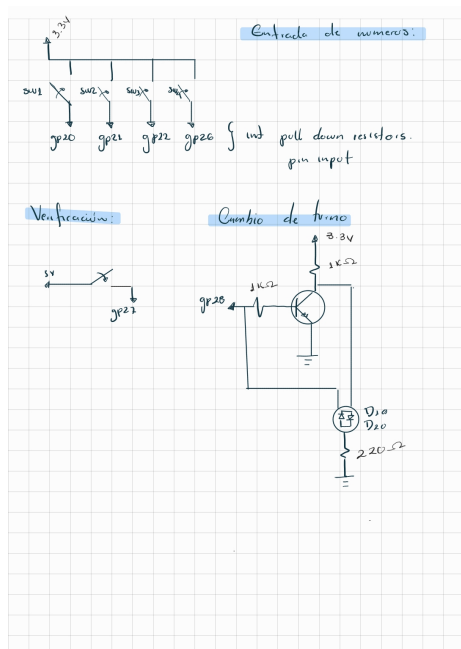


Figura 0-2: Esquemático de la entrada de posiciones, confirmación de movimiento y LED indicador de turno.

La polarización de este circuito se realiza directamente en la Raspberry Pi Pico, debido a que, teóricamente, no hay corriente que entre o salga de de la tarjeta de desarrollo. La configuración para estas entradas se realiza mediante el SDK, configurando todos los pines de entradas como resistencias pull-down. Finalmente, el indicador de turno, como el tablero de posiciones se polariza mediante una fuente externa.

## SOFTWARE

### Estructura de las Funciones Implementadas

El desarrollo del programa está organizado a través de diversas funciones que permiten la interacción del sistema con los jugadores mediante el manejo de los GPIOs para la lectura de botones y el control de los LEDs. A continuación, se presenta una descripción general de cada una de estas funciones:

- **init\_read\_gpio:** Esta función inicializa los pines de la Raspberry Pi Pico que se utilizan para leer las entradas de los jugadores. Estos pines permiten capturar las posiciones seleccionadas en el juego.
- **init\_leds:** Inicializa los pines conectados a los LEDs que indican el estado del juego. Estos incluyen los LEDs que muestran la posición de los jugadores, el turno actual, y si se ha ingresado una posición inválida.
- **set\_led:** Permite encender un LED en la posición indicada, diferenciando si el LED pertenece al jugador 0 o al jugador 1.
- **check\_bitmask:** Verifica si un bit específico en una máscara binaria está encendido, lo cual es útil para comprobar qué posiciones en el tablero han sido ocupadas por los jugadores.
- **read\_binary\_input:** Realiza la lectura binaria de la posición ingresada por el jugador mediante los botones.
- **button\_isr:** Función de interrupción asociada al botón de confirmación de jugada. Esta función se activa cuando el jugador presiona dicho botón. Este prende una bandera que indica que se prendió el botón.
- **change\_player:** Cambia el turno entre los jugadores, alternando entre el jugador 0 y el jugador 1.
- **reset\_game:** Reinicia el estado del juego, apagando todos los LEDs, aleatorizando el primer jugador, y preparando el sistema para una nueva partida.
- **check\_winner:** Verifica si alguno de los jugadores ha completado una línea de tres en el tablero y, en ese caso, declara un ganador.

- **process\_game:** Esta función coordina el flujo del juego, desde la lectura de las posiciones ingresadas hasta la validación de un ganador o el cambio de turno.
- **process\_time:** Procesa el tiempo durante el cual el botón está presionado, si este es mayor de 5 segundos, se reinicia el juego.
- **choose\_player:** Al inicio del juego, esta función determina aleatoriamente qué jugador comenzará la partida.

## Definición e Inicialización de GPIOs

### Máscaras de Pines

- **Pines de lectura de botones:**

```
READ\_PINS = 0x0E300000
(Pines 21, 22, 26, 27)
```

Utilizados para ingresar la posición de los jugadores.

- **Pines de LEDs del juego:**

```
LEDS\_PIN = 0x001FFFFC (Pines 2-20)
```

Controlan las posiciones seleccionadas por los jugadores.

- **Pines de LEDs del jugador 0:**

```
LEDS\_PIN\_PLAYER\_0 = 0x000AAAA8
```

- **Pines de LEDs del jugador 1:**

```
LEDS\_PIN\_PLAYER\_1 = 0x00055554
```

- **Pin de LED interno de la Raspberry Pi Pico de error:**

```
INNER\_LED = 25
```

### Variables Importantes

- **Máscara de posiciones seleccionadas (bitmask):** Define las posiciones seleccionadas por ambos jugadores.
- **Estado de los LEDs de los jugadores:**

- **g\_state\_leds\_0:** Estado de los LEDs ocupados por el jugador 0.
- **g\_state\_leds\_1:** Estado de los LEDs ocupados por el jugador 1.

- **Posición seleccionada (position):** Almacena la posición ingresada por el jugador en cada turno.
- **Estado del jugador actual (g\_state\_player):** Indica si el jugador actual es el jugador 0 (valor false) o el jugador 1 (valor true).
- **Bandera del botón de confirmación (button\_pressed):** Se activa cuando el botón es presionado.
- **Momento de la presión del botón (moment):** varía entre 0, 1 y 2 según el momento de la presión del botón: 0, si es un flanco de subida, 1 si es de bajada y 2 si se cumple la condición de pasar por encima de 5 segundos.

## IMPLEMENTACIÓN

En la implementación del juego de tres en línea, se conectaron los botones y LEDs al hardware de la Raspberry Pi Pico, distribuyendo los GPIOs para manejar tanto la entrada de los jugadores como la visualización del estado del juego en el tablero de LEDs. Los interruptores DIP permiten a los jugadores ingresar sus movimientos, mientras que los LEDs bicolors indican las posiciones ocupadas por cada jugador. A continuación, se muestra el montaje físico del sistema, donde se puede observar la distribución del hardware y las conexiones a la Raspberry Pi Pico.

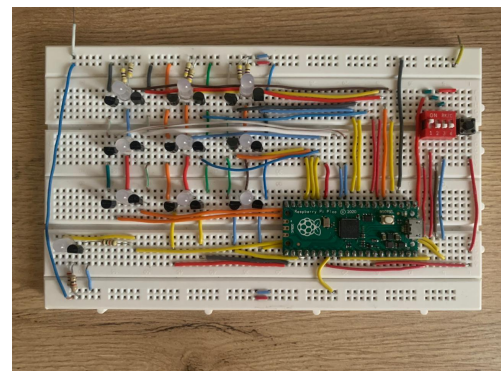


Figura 0-3: Montaje del tres en línea.

## CONCLUSIONES

- Se pudo evidenciar la importancia de la gestión eficiente de recursos que en algunos sistemas suelen ser limitados, como los GPIOs de la Raspberry Pi Pico, aunque los GPIOs, fueron justos para esta práctica, no siempre será así, crear códigos eficientes y diseñar sistemas que utilicen de manera óptima los recursos necesarios y disponibles en cada tarjeta de desarrollo.
- El SDK de Raspberry Pi Pico proporciona una amplia gama de opciones para el control y la configuración de los GPIOs, lo que se convierte en una herramienta muy fuerte a la hora de diseñar firmware para sistemas embebidos sobre este microcontrolador. Además, el desarrollo modular y la implementación de funciones modulares facilitó el desarrollo del sistema, permitiendo mantener un

código limpio y escalable.

- Por último, el laboratorio permitió aplicar y consolidar conocimientos teóricos sobre el uso de GPIOs en la tarjeta de desarrollo Raspberry Pi Pico, proporcionando una visión clara de cómo los sistemas embebidos pueden ser utilizados para aplicaciones del mundo real, como lo son juegos interactivos o sistemas de control.

## BIBLIOGRAFÍA:

- Raspberry Pi Foundation. Raspberry Pi Pico documentation. Recuperado de <https://www.raspberrypi.com/documentation/microcontrollers/pico-series.html#raspberry-pi-pico-and-pico-h>