

Instituto Tecnológico de Costa Rica

Escuela de Ingeniería en Computadores

Curso: Fundamentos de Sistemas Computacionales

Segundo Proyecto “Light Hunt”

1 semestre 2025

Prof. Milton Villegas Lemus

Asistente: Asly Barahona

Estudiantes: María José Solano Cordero,2024314761

Emmanuel García Quiros,2025065962

Fecha de Entrega:20 de junio del 2025

**Introducción**

En la siguiente documentación presentamos la ampliación de nuestro proyecto “Light Hunt”. Esta ampliación consiste en el diseño e implementación de un circuito lógico combinatorio que muestre la puntuación del jugador según un sistema exceso 3 circulatorio en cuatro leds, así como la conexión inalámbrica de la Raspberry Pi pico W con la computadora, para crear una interfaz de rankings de puntajes y usuarios.

Objetivo del Proyecto:

Implementar un sistema empotrado interactivo que utilice leds, compuertas lógicas y conexiones inalámbricas con una Raspberry Pi Pico w para mejorar el juego creado en el proyecto 1

Importancia del proyecto:

Este proyecto nos permite un desarrollo de habilidades en electrónica mediante los componentes utilizados y sus respectivas soldaduras, aplicación de conocimientos sobre circuitos lógicos y su elaboración, programación de una Raspberry Pi Pico/W que nos permitió controlar el juego y su conexión inalámbrica a una computadora, aplicando conocimientos teóricos en un proyecto práctico.

Componentes y sus distintos usos en el proyecto:

Se usaron 4 leds AD688 rojos 5mm, que nos ayudó a mostrar la puntuación del jugador en exceso 3

Para el funcionamiento del circuito lógico combinatorio se utilizaron integrados de compuertas lógicas AND 74LS08, OR 74LS32, NOT 74LS04, XOR 74LS86, que nos ayudaron a enviar señales a los LEDS encargados de mostrar la puntuación exceso 3.

Nivel de completitud del proyecto:

**Conclusiones**

**Recomendaciones**

**Análisis de Resultados**

**Bibliotecas utilizadas**

En este proyecto fueron utilizadas cinco Bibliotecas que son fundamentales para el desarrollo correcto, vamos a conocer un poco de ellas:

* Network

Esta biblioteca consiste en un conjunto de módulos que permiten la comunicación a través de redes, como Wi-Fi y Ethernet. Estos módulos proporcionan funcionalidades para conectar dispositivos a redes, enviar y recibir datos, y realizar diversas operaciones de red. Esta biblioteca se utilizó para la conexión inalámbrica entre la Raspberry y la computadora. Tienen funciones como:

**network.WLAN:** Permite la conexión a redes Wi-Fi. Incluye métodos para buscar redes, conectarse a una red específica, obtener información de la conexión, etc.

**network.STA\_IF**: Modo Estación (Station) para conectarse a un punto de acceso Wi-Fi.

**network.AP\_IF**: Modo Punto de Acceso (Access Point) para crear una red Wi-Fi a la que otros dispositivos puedan conectarse



Figura 1. Ejemplo de uso de network

* Socket

La librería socket en MicroPython permite comunicaciones en red mediante sockets TCP/IP, similar a la librería socket de Python estándar, pero con funciones más ligeras para dispositivos embebidos

**socket.socket():**Crea un nuevo socket.

**socket.connect((host, port)):**Conecta el socket a un servidor (modo cliente).

**socket.bind((host, port)):**Asocia el socket a una dirección (modo servidor).

* Machine

El propósito de esta Biblioteca es proporcionar acceso directo al hardware, incluyendo GPIO, PWM, ADC, UART, 12C, SPI y otras interfaces. Acá la utilizamos para acceder a los pines de la Raspberry Pi Pico/W y poder realizar el funcionamiento correcto.

**GPIO:** Configurar pines como entrada/salida.

**PWM:** Generar señales moduladas por ancho de pulso.

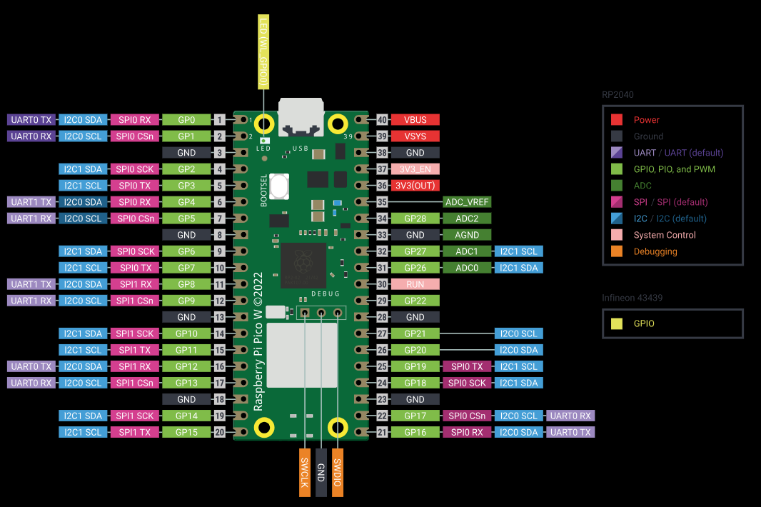


Figura 2. Pines Raspberry Pi Pico/W

* Utime

Para solucionar el problema del congelamiento que provocaba utilizar Time, se decidió utilizar utime la cual tiene como propósito manejar retardos, mediciones de tiempo y conversiones de formato de tiempo.

**sleep(seconds):** Pausa la ejecución durante seconds segundos.

**sleep\_ms(ms)** y **sleep\_us(us):** Retardos en milisegundos y microsegundos.

**ticks\_ms():** Obtiene el tiempo actual en milisegundos (útil para mediciones).

* Random

Esta Biblioteca tiene como propósito generar números aleatorios para patrones o decisiones no deterministas, para este proyecto fue utilizada para la realización de las secuencias de leds.

**randint(a, b):** Entero aleatorio entre a y b (incluidos).

**random():** Float aleatorio en el rango [0.0, 1.0).

**choice(seq):** Selecciona un elemento aleatorio de una secuencia.

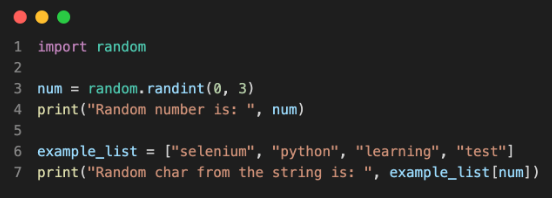


Figura 3. Ejemplo de uso de Random

**Diagramas**

* Diagrama de Arquitectura
* Diagrama de Clases

**Referencias**

*machine — functions related to the hardware — MicroPython latest documentation*. (s. f.). <https://docs.micropython.org/en/latest/library/machine.html>

Raspberry Pi. (2024). *Pico-series microcontrollers - Raspberry Pi documentation*. Raspberrypi.com. <https://www.raspberrypi.com/documentation/microcontrollers/pico-series.html>

Santos, S., & Santos, S. (2020, July 30). *MicroPython: Wi-Fi Manager with ESP32 and ESP8266 | Random Nerd Tutorials*. Random Nerd Tutorials. <https://randomnerdtutorials.com/micropython-wi-fi-manager-esp32-esp8266/#:~:text=Biblioteca%20MicroPython%20de%20WiFiManager,%C2%BFQu%C3%A9%20es%20esto>?

*socket – socket module — MicroPython latest documentation*. (n.d.).

<https://docs.micropython.org/en/latest/library/socket.html>

*time – time related functions — MicroPython latest documentation*. (s. f.). <https://docs.micropython.org/en/latest/library/time.html>