

Primer proyecto

Kevin Castro Pérez

Jacob Zumbado Cubero

Instituto Tecnológico de Costa Rica

CE-1104 Fundamentos de Sistemas Computacionales

Prof. Luis Alberto Chavarría Zamora

30 de octubre de 2024

Introducción:

Este proyecto representa la construcción de una máquina de Pinball que permite explorar la integración entre hardware y software mediante el uso de una Raspberry Pi Pico W, consolidando conocimientos fundamentales en sistemas embebidos y programación para interfaz gráfica.

Diagrama general de la maqueta:

Diagrama eléctrico:

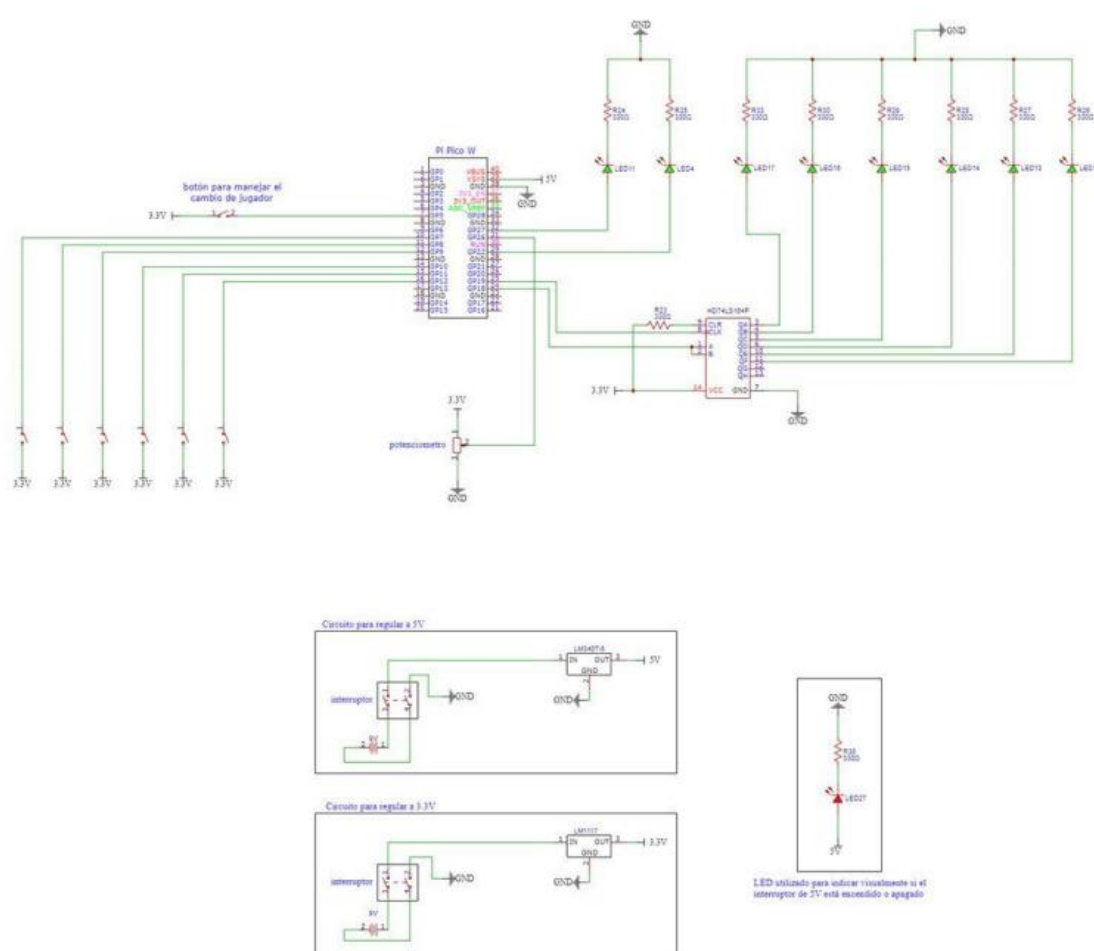


Imagen de la maqueta en construcción:



Funcionalidades:

La maqueta es capaz de detectar si un balón (o cualquier objeto redondo de material conductor) entra dentro de alguna de las zonas de anotación, este evento provoca reacciones como encender los leds. También se puede conectar a una red de internet y comunicarse con la interfaz gráfica. También cuenta con un lanzador mecánico y leds para la retro alimentación.

Detección de la anotación:

```
while True:
    """cuando el boton es presionado, lee el potenciómetro e imprime el valor(falta meter la logica para que diga cuando
    un jugador esta seleccionado) y tambien manda el paquete "b'cambio'" """

    if boton_cambio.value() == 1:
        leer_potenciómetro()
        # Crea un socket
        s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        try:
            s.connect((host, port))
            s.send(b'cambio')
        except OSError as e:
            print("Error de conexión:", e)
        finally:
            s.close() # Cierra el socket manualmente
            time.sleep(0.5) # Anti-rebote
            print("Si, funciona Tílin")

    """cuando el boton es presionado, envia un paquete con la información "b'100'" esta debe ser la leida en el .listen()
    """
    if boton_100.value() == 1:
        boton_pulsado(lista_100)
        led_jugador_1.off()
        time.sleep(1)
        # Crea un socket
        s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        try:
            s.connect((host, port))
            s.send(b'100')
        except OSError as e:
            print("Error de conexión:", e)
        finally:
            s.close() # Cierra el socket manualmente
            time.sleep(0.5) # Anti-rebote
            print("Si, funciona Tílin")
```

Mediante el uso de un *while* se leen todos los botones para verificar si alguno fue accionado, cuando se detecta que alguno fue presionado, enciende los leds y envía un mensaje mediante el *Socket* indicando cual botón fue presionado.

Dentro de la interfaz dichos códigos son leídos y desencadenan ciertas acciones, tales como:

Reproducir un sonido: para las zonas “normales” se activa un sonido y para la “especial” se activa uno diferente.

Sumar puntos: dependiendo de la zona de anotación varía la cantidad de puntos sumados, siendo la “especial” la que más puntos otorga.

```

def iniciar_socket():

    host = '0.0.0.0'
    port = 12345

    with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
        s.bind((host, port))
        s.listen()

        print("Esperando conexión...")

        while True:
            conn, addr = s.accept()
            with conn:
                print(f"Conectado a {addr}")
                data = conn.recv(1024)

                if data in [b'100', b'150', b'200', b'250']:
                    puntos = int(data.decode())
                    reproducir_sonido_botones()
                    mostrar_mensaje(puntos)

                if data == b"500":
                    puntos = int(data.decode())
                    reproducir_sonido_500()
                    mostrar_mensaje(puntos)

```

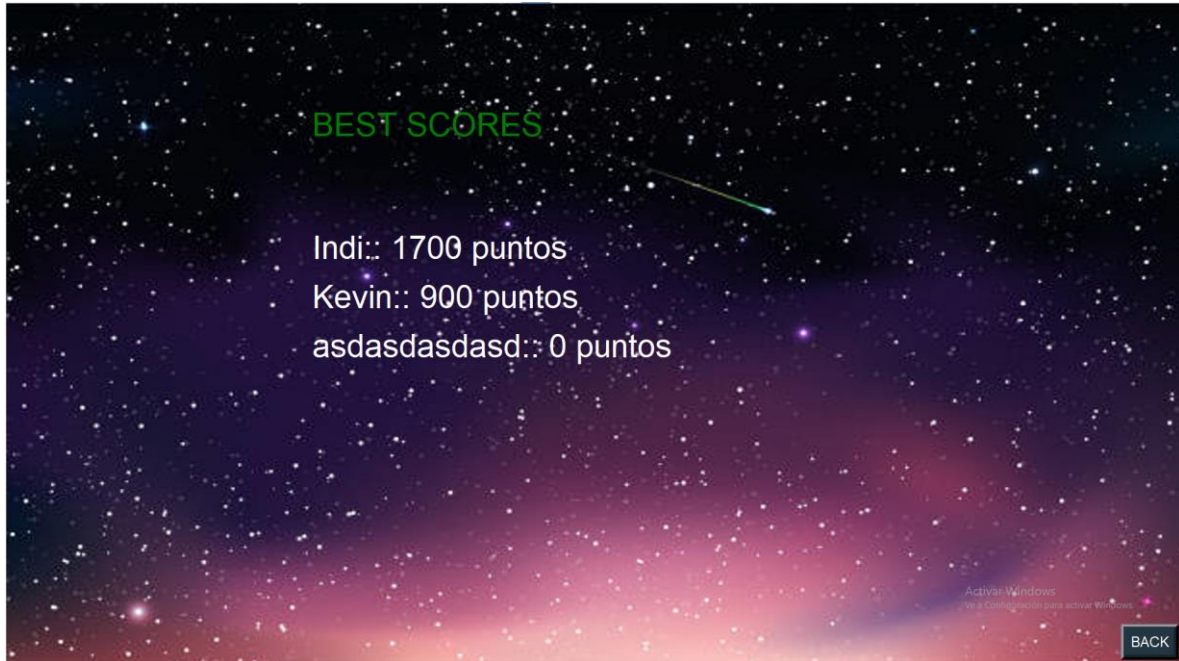
Elección del jugador e imagen de perfil:

La interfaz permite seleccionar una imagen de perfil para el jugador e insertar su nombre.



Mejores puntajes:

El puntaje se muestra al finalizar la partida o presionar el botón dentro de la pantalla principal de la ventana.



La ventana muestra los tres mejores puntajes y el nombre del jugador que los obtuvo.

Conclusiones:

Trabajar con un dispositivo como el Raspberry Pi Pico W es una experiencia didáctica que permitió desarrollar el proyecto, la disponibilidad de tutoriales e información en el internet facilitó el autoaprendizaje para aplicar los conocimientos en el proyecto.

Recomendaciones:

Usar un repositorio en GitHub o GitLab para almacenar el código y documentar los avances del proyecto, facilitando la colaboración entre integrantes.

Bitácora:

Fechas	Integrantes	Actividad	Horas
14-10-2024	Jacob Zumbado	Construcción de la maqueta	6 horas
15-10-2024	Kevin Castro Jacob Zumbado	Cableado de la maqueta	5 horas
16-10-2024	Kevin Castro	Creación del código en Micro Python	4 horas
18-10-2024	Jacob Zumbado	Creación de la interfaz	5 horas
22-10-2024	Jacob Zumbado Kevin Castro	Optimización del cableado	6 horas
24-10-2024	Jacob Zumbado	Detalles a la interfaz gráfica	5 horas
27-10-2024	Kevin Castro	Realización de la documentación	4 horas

Enlace al repositorio:

<https://github.com/Kevin416A12/Proyecto-de-Fundamentos>

Referencias:

<https://core-electronics.com.au/guides/raspberry-pi-pico-w-connect-to-the-internet/>