



INFORME DE LABORATORIO

Autores: José Moreno, Valentina Restrepo

Laboratorio de Electrónica Digital III

Departamento de Ingeniería Electrónica y Telecomunicaciones

Universidad de Antioquia

Abstract

This report presents a comprehensive account of the iterative development process of a "reaction" game. The project encompasses various stages, including coding in C, implementation on Raspberry Pi Pico W, basic electronic circuit design, and logical inference. Through this endeavor, the project aims to demonstrate the integration of programming skills, hardware utilization, and logical problem-solving.

Introducción

El proyecto consiste en un circuito que a través de una raspberry pico pi w permite el control sobre tres LEDs, cuatro botones y cuatro displays de 7 segmentos, todos con sus respectivas resistencias de pull-down. El control se da a través de la lógica de un juego de reacción, donde luego de una secuencia de presentación de los tres LEDs, se enciende uno de los tres (escogido de forma aleatoria) en un rango de tiempo menor o igual a 10 segundos y el botón asociado a este LED debe pulsarse lo más rápido posible, el objetivo de la práctica es medir ese tiempo de respuesta y mostrarlo en los displays de 7 segmentos.

Con un enfoque en la programación en C, diseño de circuitos electrónicos simples y lógica de control, este juego ofrece una experiencia dinámica que permite explorar elementos como la programación en C, programación de microcontroladores, desarrollo lógico y análisis de información de manera asertiva.

Marco Teórico

Microcontrolador

La Raspberry Pi Pico W es un microcontrolador de bajo costo basado en el chip RP2040. Cuenta con un procesador ARM Cortex-M0+ con una frecuencia de reloj de hasta 133 MHz y una memoria flash integrada de hasta 16 MB. La Pico W ofrece una amplia gama de capacidades de E/S, incluyendo 30 pines GPIO, que admiten varios protocolos de comunicación como UART, SPI, I2C y PWM. Además, cuenta con un sistema de temporizador de 4 canales y un conversor analógico-digital (ADC) de 3 canales. La programación de la Pico W se puede realizar utilizando el SDK de Raspberry Pi Pico en lenguaje C/C++ o mediante MicroPython. [1]

Evitamos utilizar los pines reservados para protocolos de comunicación como UART, SPI o I2C para los GPIO del proyecto porque estos pines tienen funciones específicas asignadas que podrían interferir con la funcionalidad de los pines GPIO. Al reservar los pines GPIO exclusivamente para el control de dispositivos de entrada y salida, como LEDs y botones, evitamos conflictos y garantizamos un funcionamiento confiable del proyecto.

Display 7 segmentos

Para lograr una visualización adecuada en cuatro displays de 7 segmentos, se emplea la técnica de multiplexación. Esto implica alternar rápidamente

entre los displays activándolos secuencialmente a través de pulsos de activación. La frecuencia de multiplexación, determinada por el tiempo de activación de cada display y el tiempo de retardo entre cada cambio, debe ser lo suficientemente alta para evitar el parpadeo perceptible y, al mismo tiempo, lo suficientemente baja para permitir una visualización clara y sin distorsiones. Además, se debe optimizar el código para minimizar el tiempo de procesamiento entre la activación de cada display, asegurando así una transición fluida y una actualización rápida de los números. Esto puede incluir el uso de temporizadores de hardware y técnicas de control de tiempo eficientes para garantizar un rendimiento óptimo del sistema.

Resistencias pull-down

Se utilizan comúnmente en circuitos electrónicos para garantizar un estado lógico definido cuando un botón o interruptor no está activado. Para los botones de entrada conectados a la Raspberry Pi Pico W, una resistencia de pull-down se conecta entre el pin GPIO del microcontrolador y tierra (GND). Cuando el botón no está presionado, la resistencia de pull-down asegura que el pin GPIO esté en un estado de baja lógica (0), ya que la resistencia proporciona una ruta de baja resistencia a tierra. Cuando el botón se presiona, se conecta directamente al pin GPIO, lo que establece el estado del pin a un nivel de alta lógica (1). Esto permite que el microcontrolador detecte fácilmente el cambio de estado del botón y tome las acciones correspondientes en el software.

Procedimiento Experimental y Resultados

Pinout

Se comenzó con el planteamiento del pinout del proyecto (véase tabla 1), conociendo la disponibilidad de los GPIO 0 a 27, pero para evitar problemas y conflictos por protocolos, se descarta el uso los siguientes pines:

- Pines 0 (Tx) y 1 (Rx) dedicados al UART.
- Pines 8 (SDA) y 9 (SCL) para comunicación I2C.
- Pines 10, 11 y 12 para la comunicación SPI.

Componente	GPIO
LED Azul	2
LED Amarillo	3
LED Rojo	4
Botón 1	5
Botón 2	6
Botón 3	7
Botón 4 (start)	13
A	14
B	15
C	16
D	17
E	18
F	19
G	20
Display 1	21
Display 2	22
Display 3	26
Display 4	27
LCD Nokia data pin	11
LCD Nokia Clock pin	18
LCD Nokia CS pin	8
LCD Nokia D/C pin	13
LCD Nokia reset	8

Tabla 1. Pinout del proyecto

Tras haber añadido el display LCD Nokia 5110, se emplearon algunos GPIO que en su momento no se consideraron, pues este periférico hace uso de la interfaz SPI.

Simulación

Se realizó una simulación del montaje del proyecto en la plataforma Tinkercad haciendo uso del arduino uno disponible en la misma, ya que al llevarse a la práctica solo haría falta cambiar las conexiones a los pines de la raspberry pico pi w.

Antes de poner las resistencias en serie entre el cátodo del LED y VCC, se calculó el valor del resistor necesario para cada uno de los tres LEDs con la ecuación 1, pues habrán tres colores diferentes (azul, amarillo, rojo) y cada uno requiere una potencia diferente.

Partiendo de la Ley de Ohm y con un $V_{fuente} = 3.3V$, se llega a:

$$R = \frac{V_{fuente} - V_{LED}}{I} \quad (1)$$

LED	Voltage (V)	Corriente (A)	Resistencia teórica (Ω)	Resistencia práctica (Ω)
Rojo estándar	1.5	0.015	120	1k
Rojo brillante	2	0.02	65	1k
Amarillo estándar	1.8	0.015	100	1k
Amarillo brillante	2	0.02	65	1k
Azul brillante	2	0.02	65	1k

Tabla 2. Cálculo de resistencias para LEDs

Para efectos prácticos se implementan resistencias de valores más comerciales que no sugieren problema alguno con la luminosidad de cada LED.

Para los cuatro botones, se implementaron resistencias físicas de pull-down de $10k\Omega$, esto para transmitir una entrada en alto cuando se pulsa cada botón.

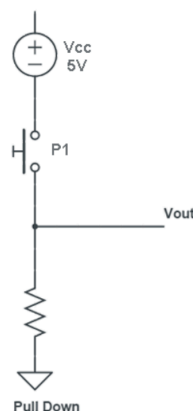


Figura 1. Resistencia de pull-down

Para la simulación, se tomaron cuatro displays multiplexados de cátodo común tal como se muestra en la figura 2, pues en Tinkercad no estaba la opción de los cuatro displays manejados directamente a través de GPIO sino por I2C y ese no es el objetivo de esta práctica.

Se usaron transistores NPN y las resistencias tanto para cada segmento como para cada transistor son de $1k\Omega$.

No se implementó código, ya que debido a la complejidad del mismo, la estructura debe ser modular y Tinkercad no tiene esta opción.

Cátodo Común

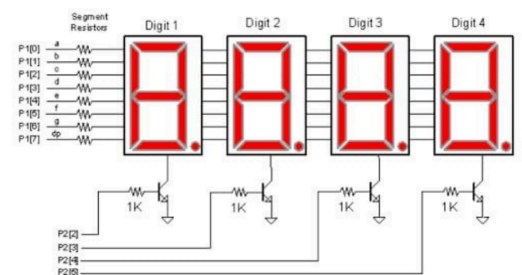


Figura 2. Multiplexación de displays 7 segmentos

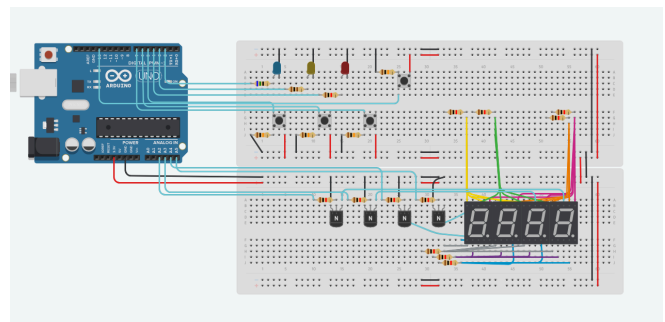


Figura 3. Montaje del circuito en Tinkercad

A través de la simulación del montaje se verificó el funcionamiento de pull-down para los botones y las demás conexiones.

Montaje

Para el montaje sólo se varió el display, pues se usó uno que ya venía multiplexado (referencia TDCR1060m) y siguiendo el planeamiento de pines que se propuso, se conectó tal como está en la figura 2 con transistores NPN 2N3904 de tipo

BJT y con el pinout del mismo que entrega el fabricante (véase figura 4), para este pinout no se tuvieron en cuenta los puntos de los pines 4, 7, 9 y 12, solo los segmentos y los cátodos comunes de cada display (D1, D2, D3 y D4).

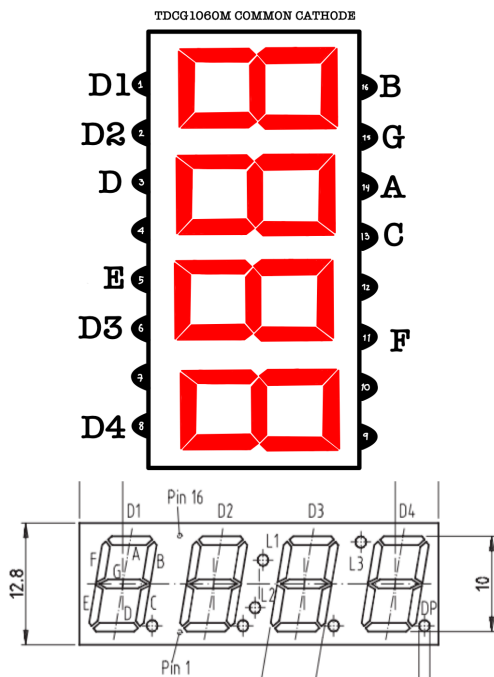


Figura 4. Pinout del display

Los demás elementos se mantuvieron igual que en la simulación.

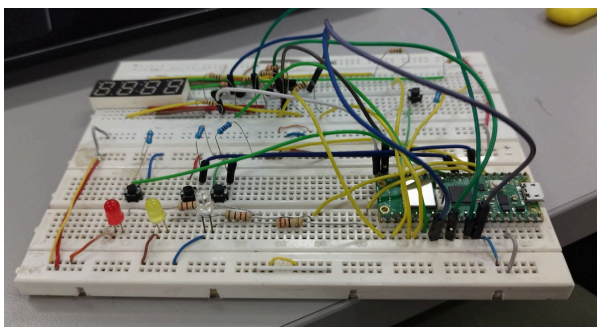


Figura 5. Montaje del circuito

Código

El código para el proyecto se planteó de forma modular y a través de funciones

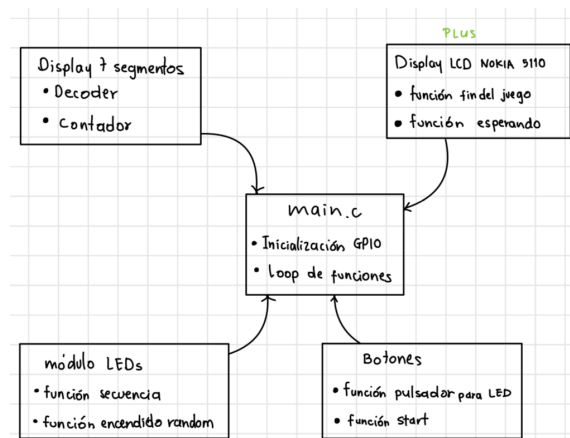


Figura 6. Módulos para la programación

El módulo Display 7 segmentos tendría dos funciones, una para el decoder, en la que reconocería la representación a través de los segmentos de cada número. La otra función se encargaría de servir como cronómetro para la función de medición del tiempo del programa.

El módulo LEDs tendría dos funciones, la primera implementaría la secuencia principal que se da con los LEDs al iniciar el programa con el botón start y la segunda función se encargará de encender alguno de los tres LEDs de forma aleatoria.

El módulo botones tendrá dos funciones, la primera implementa la verificación del pulsado del botón correcto al momento de estar jugando y la segunda función se encarga de inicializar el juego al oprimir el botón start.

El módulo Display Nokia es un plus planteado que tiene dos funciones o estados, se visualizarán diferentes caracteres según la etapa del juego, un caracter para el fin del juego y otro caracter que indique otro estado posible del mismo.

Todos estos módulos se implementarán como funciones en el main.c, donde primero se inician los GPIO a través de los pines de la tabla 1 y luego en un bucle infinito, se ejecutará el juego.

Discusión de resultados

La primera prueba que se llevó a cabo se enfocó en los botones, para verificar su correcto

funcionamiento, pues no fue necesario implementar antirrebotes.

Tras haber finalizado con buenos resultados las pruebas de reconocimiento de los cuatro botones, se enfocaron las pruebas en los contadores, para verificar que los tiempos tomados en el juego sí fueran los correctos, por lo que se comparó el resultado mostrado en el display con un cronómetro tras empezar una prueba del juego y se obtuvieron resultados acordes.

Se probaron las penalizaciones para los casos en los que fueran necesarios (presionar el botón incorrecto o reaccionar luego del tiempo), se hicieron correcciones y se tuvo un buen sistema de penalizaciones.

Finalmente, se probó el código para el display Nokia 5110 que en principio no funcionó, pero tras haber corregido la forma en la que se transmitían los datos y se conectaba al microcontrolador a través de los GPIO comenzó a dar mejores resultados.

Conclusiones

- Caracterizar correctamente el montaje del circuito es fundamental para detectar y corregir errores en tiempos cortos, por lo que se recalcó en el informe la importancia de haber comenzado con la planeación de los pines antes que con cualquier otro elemento del proyecto.
- La implementación de un código modular y por funciones permite escalar y aplicar en otros aspectos las funcionalidades del mismo, permitiendo llamar funciones con diferentes entradas en más de una ocasión.
- El conocimiento de las librerías adecuadas para los diferentes periféricos permitieron abarcar todas las funciones propuestas sin mayor complicación, por lo que las correcciones que se hicieron tomaron menos tiempo de lo esperado.
- Ajustar adecuadamente la frecuencia de los displays de 7 segmentos permitía visualizar

resultados de forma coherente, pues una frecuencia menor o mayor desajustaba el efecto visual de ver el resultado en simultáneo.

Referencias

- [1] Monk, S. (2022). *Raspberry Pi Cookbook*. "O'Reilly Media, Inc."
- [2] HETPRO. (2017). Resistencia de LED, Cómo calcularla en función a la corriente. Disponible en: <https://hetpro-store.com/TUTORIALES/resistencia-de-led/>. [Accedido: Marzo, 2024]