

Receptor, Transmisor y Decodificador de Código Morse con UART

Santiago González Camiscia

santiagogonzalezcamiscia@impatrq.com

Matias Leonel Gonzalez Paurailly

matiasgonzalezpaurailly@impatrq.com

Keller Mateo

mateokeller@impatrq.com

E.E.S.T. N°7 - "TRQ"

Resumen— En este proyecto se presentará el desarrollo de un receptor, transmisor y decodificador de código morse utilizando el protocolo de comunicación UART (Universal Asynchronous Receiver-Transmitter) utilizando la Raspberry Pi Pico, un microcontrolador versátil y accesible.

I. INTRODUCCIÓN

El código morse es un sistema de comunicación que utiliza una serie de puntos y rayas para representar letras y números. Aunque ha sido reemplazado en gran medida por tecnologías de comunicación más modernas, el código morse es aún utilizado en múltiples sistemas de aviación como el sistema de ADF (Automatic Direction Finder) o Marker del sistema ILS (Instrumental Landing System).

Nuestro objetivo al momento de realizar este proyecto es posibilitar un fácil entendimiento de qué dicen estos mensajes, usados en muchos de estos sistemas, mediante la utilización de la poderosa Raspberry Pi Pico y sus muchas funciones que permiten fácilmente adaptarla y personalizarla para funcionar en distintos ambientes.

II. DESCRIPCIÓN FUNCIONAL

A. Diagrama en bloques

1) Idea Original

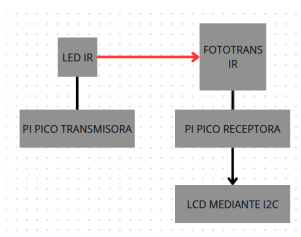


Fig. 1 Diagrama en bloques del circuito original

2) Modelo final debido a la falta de componentes

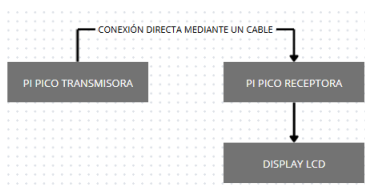


Fig. 2 Diagrama en bloques del circuito alternativo

B. Diagrama de código

1) Pico Transmisora

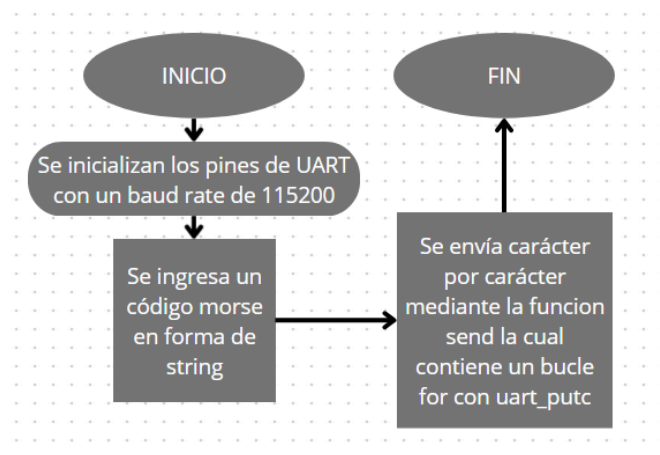


Fig. 3 Diagrama de código del microcontrolador transmisor

2) Pico Receptora - Función General

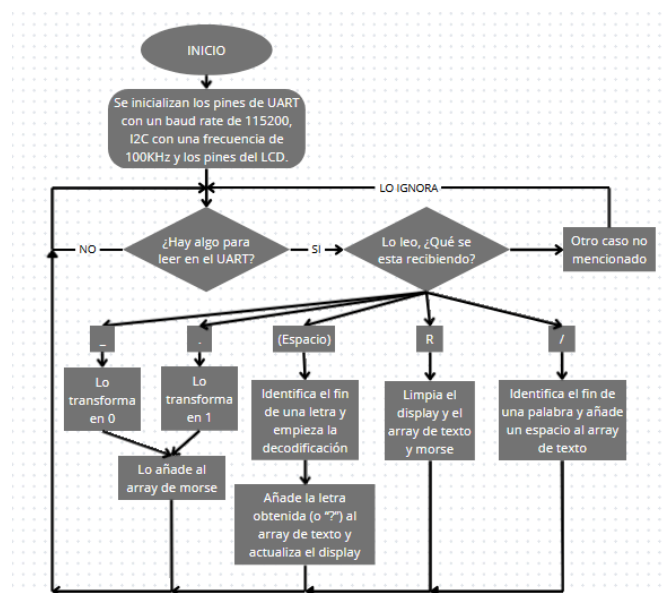


Fig. 4 Diagrama de código general del microcontrolador receptor

3) Pico Receptora - Función de decodificación

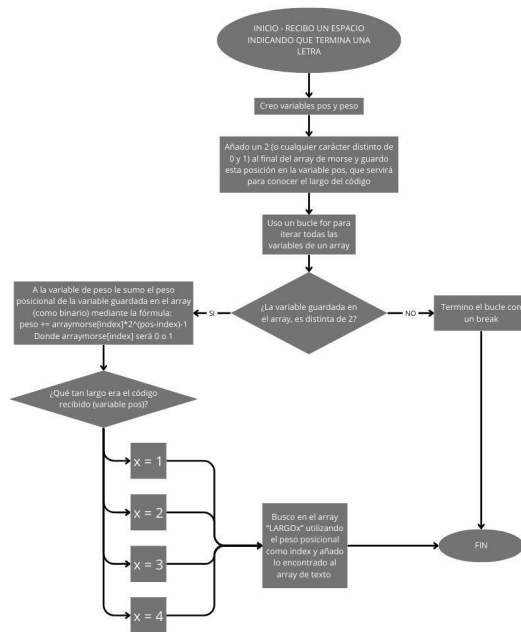


Fig. 5 Diagrama de código de la función de decodificación

C. DESCRIPCIÓN DE CIRCUITOS

- 1) **Conexión UART básica:** Para establecer una conexión UART entre dos microcontroladores, llamémoslos A y B, se deben realizar las siguientes conexiones:
 - RX-A > TX-B,
 - RX-B > TX-A
 - Los GND de ambos microcontroladores entre sí.

Esta configuración asegura una comunicación bidireccional entre ambos microcontroladores.

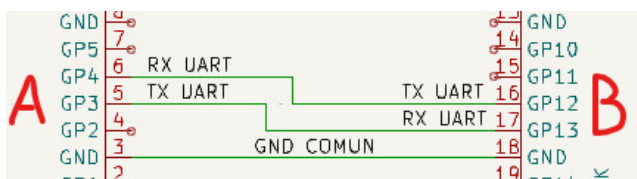


Fig. 6 Conexión básica de UART entre dos microcontroladores

- 2) **Conexión UART mediante leds y fototransistores:** este método utiliza la tecnología UART adaptada a un sistema de transmisión óptica. Un LED emite señales de luz moduladas que representan los datos UART, actuando como un transmisor. Estas señales de luz son recibidas por un fototransistor, que permite el paso de corriente en sincronía con las ondas recibidas. De esta manera, se establece una conexión que funciona como un “conductor inalámbrico”, utilizando luz para transmitir información.

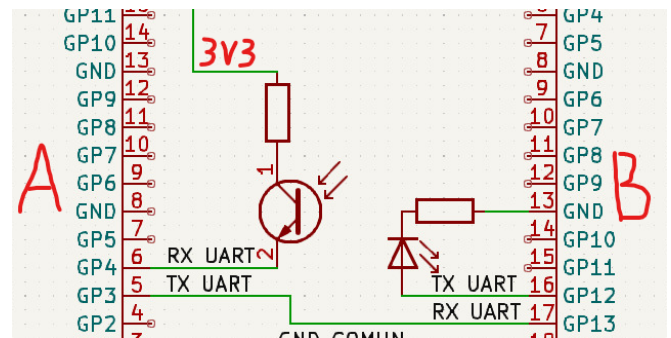


Fig. 7 Conexión de UART con LED y FOTOTRANSISTOR en vez de un cable

III. ALCANCE LOGRADO

Durante este proyecto logramos desarrollar un sistema completo para transmitir, recibir y decodificar mensajes en código Morse utilizando la Raspberry Pi Pico. Durante el proceso fuimos aprendiendo a cómo utilizar el programa de comunicaciones de UART mediante videos en Youtube, foros de internet[1] y/o distintas páginas de Google[2][3] e incluso nos permitió explorar con más profundidad la capacidad de este microcontrolador.

Descubrimos y aprendimos a utilizar funciones de distintos pines de la Raspberry Pi Pico que anteriormente desconocíamos, lo que nos llevó a investigar e informarnos aún más sobre las capacidades del microcontrolador. Además, creemos que este sistema puede ser fácilmente modificado y mejorado. Por ejemplo, podría ser editado para trabajar con otros métodos de ingreso de datos, ampliando así su aplicabilidad en diferentes contextos y proyectos futuros.

IV. CONCLUSIONES

Luego de numerosas pruebas, ajustes y algunos tropiezos, logramos que el display mostrara correctamente las letras introducidas mediante código Morse. Inicialmente, la idea era integrar un LED infrarrojo en el circuito junto a un fototransistor que funcionara como un modo de comunicación entre transmisor y receptor, pero debido a varios inconvenientes, optamos por una solución más práctica que simplificó tanto la implementación como la ejecución del proyecto. Este enfoque nos permitió alcanzar los objetivos planteados de manera más eficiente y rápida para poder realizar las pruebas iniciales, aunque no se descarta la posibilidad y simpleza de aún implementar este mecanismo en un futuro.

V. ANEXOS

A. Referencias

- [1] Foro de reddit que ayudó con el receptor:
https://www.reddit.com/r/raspberrypipico/comments/uc9v1r/issue_with_rea_dingwriting_uart_when_programming/
- [2] Documentación de Raspberry Pi acerca de UART:
https://www.raspberrypi.com/documentation/pico-sdk/hardware.html#group_hardware_uart
- [3] Proyecto de Wokwi acerca de UART:
<https://wokwi.com/projects/397890693498146817>

B. ESQUEMÁTICOS

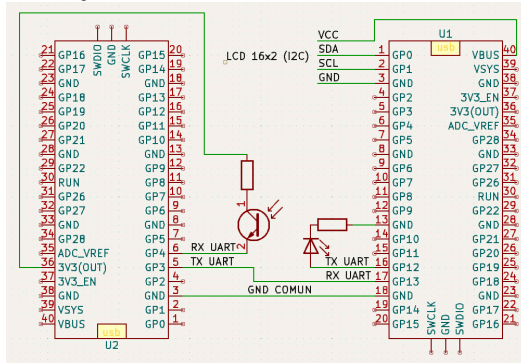


Fig. 8 idea original

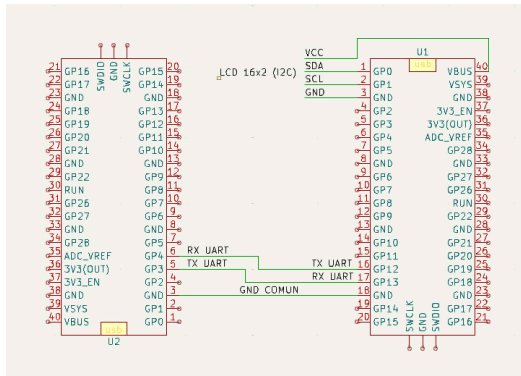


Fig. 9 idea alternativa

C. OTROS

- 1) Link del repositorio de Github:
<https://github.com/pitutsas/proyectomorse>
- 2) Link de la página del proyecto:
<https://pitutsas.github.io>