

RECONOCIMIENTO DE TRAMAS I²C CON ANALIZADOR LÓGICO

CARABALLO CARDENAS ANDRES FELIPE, UNIVERSIDAD MILITAR NUEVA GRANADA
[\(est.andres.caraballo@unimilitar.edu.co\)](mailto:(est.andres.caraballo@unimilitar.edu.co))

Resumen En esta práctica de laboratorio se utilizó el analizador lógico para observar y entender las señales que se generan en las tramas del protocolo I²C. Para ello, se empleó el microcontrolador Raspberry Pi Pico 2W y se conectaron dispositivos como el acelerómetro y una pantalla OLED.

Durante el análisis fue posible identificar varios aspectos clave como las direcciones en hexadecimal de cada dispositivo que permiten comprobar su presencia en el bus, las confirmaciones ACK y NACK, útiles para verificar si la comunicación fue aceptada o rechazada y la distribución de roles dentro del bus: la Raspberry Pi Pico 2W actuando como maestro y los demás dispositivos como esclavos, así mismo se estudió el comportamiento de las señales respecto al reloj (SCL) y la línea de datos (SDA). Allí se comprendió que, cuando la línea permanece en nivel alto (gracias a las resistencias *pull-up*), no se transmite información activa. En cambio, cuando la línea baja a nivel lógico bajo es porque el maestro o el esclavo están forzando un valor para enviar un mensaje.

I. INTRODUCCIÓN

Para el desarrollo de esta práctica de laboratorio se hizo uso de diferentes códigos proporcionados por el docente a través de su repositorio en GitHub. Estos programas se fueron ejecutando paso a paso conforme al desarrollo en cada uno de los incisos de la guía. De esta manera, fue posible recolectar y analizar las señales generadas, utilizando el analizador lógico como instrumento principal.

II. PROCEDIMIENTO

En esta sección se describe el procedimiento para el desarrollo de cada uno de los puntos propuestos en la guía de laboratorio.

a. Conexiones y preparación

En este primer punto se realizaron las conexiones de la pantalla OLED al microcontrolador Raspberry Pi Pico 2W con el objetivo de analizar su comunicación mediante el protocolo I²C. Para ello, se emplearon los pines dedicados al bus I²C del microcontrolador: SCL GP27 y SDA GP26.

Estos pines se conectaron directamente a los correspondientes de la pantalla OLED, asegurando que cada línea cumpliera su función., una vez establecida la conexión, se incorporó el analizador lógico digital, el cual dispone de 8 canales y un pin de tierra. La configuración se realizó de la siguiente manera:

- El canal CH0 se conectó a la línea de reloj (SCL) de la pantalla OLED.
- El canal CH1 se conectó a la línea de datos (SDA).
- La conexión de tierra del analizador se enlazó con el pin GND de la pantalla OLED.

De esta forma, se armó el montaje para la captura y análisis de las señales, tal como se observa en el circuito adjunto.

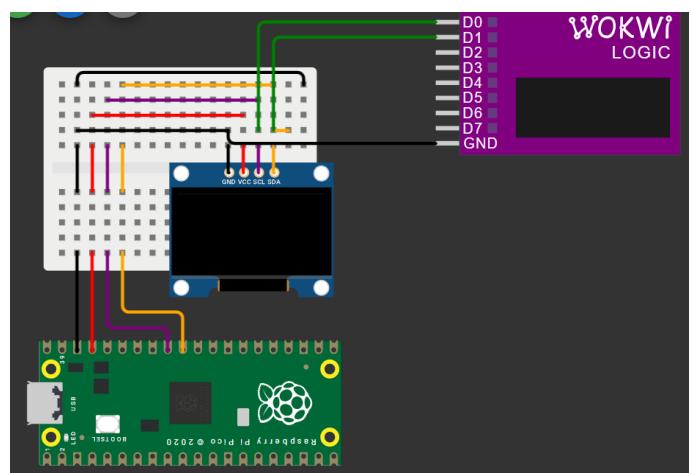


Figura 1. Montaje del circuito entre Raspberry Pi Pico 2W, pantalla OLED y analizador lógico

Al analizar el montaje internamente se muestra de la siguiente manera.

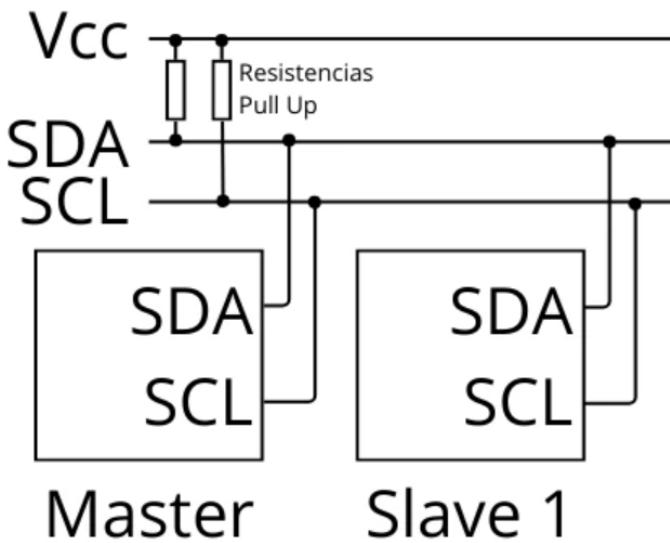


Figura 2. Diagrama del circuito internamente con el protocolo I²C

Como se aprecia en la figura anterior, el dispositivo que actúa como maestro es la Raspberry Pi Pico 2W, el cual se encarga de controlar la señal de reloj e iniciar la comunicación. Por su parte, la pantalla OLED cumple el rol de esclavo, respondiendo a las instrucciones a través de la línea de datos. Ambos dispositivos comparten el bus I²C, conformado por dos líneas principales: SDA (datos) y SCL (reloj).

En este protocolo, las líneas trabajan bajo el principio open-drain, lo que significa que ningún dispositivo genera directamente un "1". En su lugar, cuando no hay transmisión activa, las resistencias *pull-up* mantienen las líneas en nivel lógico alto (1). Cuando el maestro o el esclavo necesitan enviar información, bajan la línea a nivel lógico bajo (0).

Posteriormente, en la aplicación Logic 2, se configuró la frecuencia de muestreo en 2 MS/s y se estableció un tiempo de captura de 10 segundos. Con estos ajustes, el programa quedó listo para iniciar la simulación.

b. Elementos de la trama y cómo verlos en la señal

En esta sección se presenta una breve explicación sobre cómo visualizar e identificar cada uno de los elementos de la trama una vez que la señal ha sido capturada. Los aspectos para observar son los siguientes:

1. Bus en reposo (IDLE): SCL y SDA en alto (pull-ups).
2. Bit de Start: transición SDA: alto→bajo mientras SCL se mantiene alto. Marca el inicio de la transacción.

3. Dirección (7 bits) + R/W: se transmiten 8 pulsos de SCL. El maestro establece cada bit en SDA antes del flanco de subida de SCL. ADDR = 0x3C = 0b 0111100 (MSB primero) y R/W = 0 indica escritura. El octeto completo enviado es 0x78 = (0x3C << 1) | 0.
4. ACK/NACK (9. bit): en el noveno pulso, el maestro libera SDA. ACK → el esclavo fuerza SDA a 0. NACK → SDA queda en 1 (alto).
5. Bit de Stop: transición SDA: bajo→alto mientras SCL está alto. Finaliza la transacción de información.

c. Pruebas ACK - NACK

En este punto se realizaron pruebas con **ACK** y **NACK**, comenzando por el caso de **ACK**, se ejecutó el primer código de prueba, el cual realiza un *ping* I²C al bus para comprobar si la pantalla OLED, cuya dirección es **0x3C**, está conectada.

La función principal de este script es verificar la presencia del dispositivo en el bus:

- Si la dirección 0x3C responde, se imprime ACK, lo que indica que la pantalla OLED está conectada y disponible.
- En caso contrario, se imprime NACK, lo que significa que ningún dispositivo está asociado a esa dirección en el bus.

Por consiguiente, se procede a ejecutar el código en cuestión, pero antes se inicia la aplicación logic 2 para realizar la captura de la señal, la cual ya había sido configurada previamente, como se mencionó en el inciso a. Una vez puesta en marcha la aplicación y ejecutado el código, la señal es registrada mediante el analizador lógico, obteniendo la captura que se muestra a continuación.

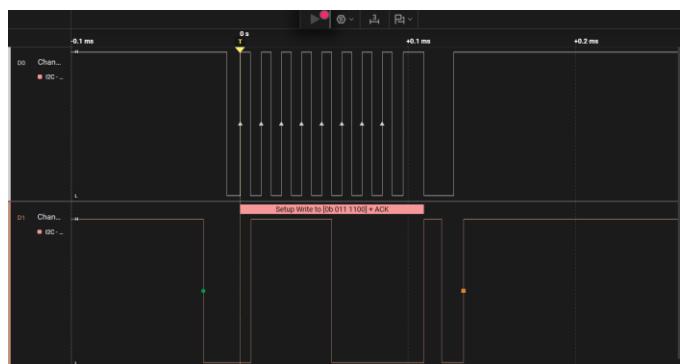


Figura 3. Señal resultante ACK

Una vez obtenida la señal, se utilizan los marcadores de la aplicación para identificar los elementos más importantes de la trama: el bit de inicio, los datos (en este caso, la dirección hexadecimal de la pantalla OLED), el bit de escritura y finalmente el bit de stop. Estos resultados se presentan a continuación.



Figura 4. Señal resultante ACK con marcadores



Figura 5. Señal resultante ACK con medición de frecuencia SCL

Al medir con cursos se obtiene el valor de frecuencia SCL de 76.5 KHz.

B. Provocar un NACK (dirección incorrecta)



Figura 6. Señal resultante NACK

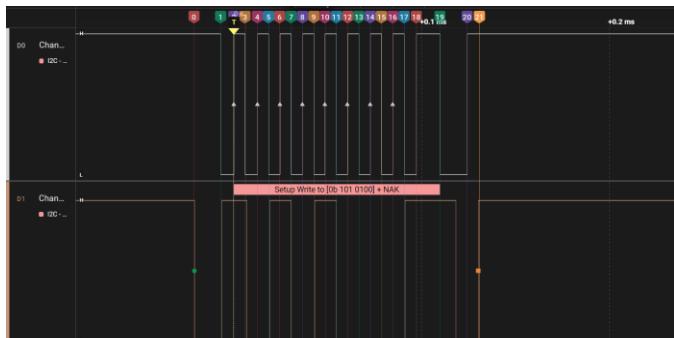


Figura 7. Señal resultante NACK con marcadores



Figura 8. Señal resultante NACK con medición de frecuencia SCL

Al medir con cursos se obtiene el valor de frecuencia SCL de 77.2KHz

d. Prueba: descubrir la ADDR correcta

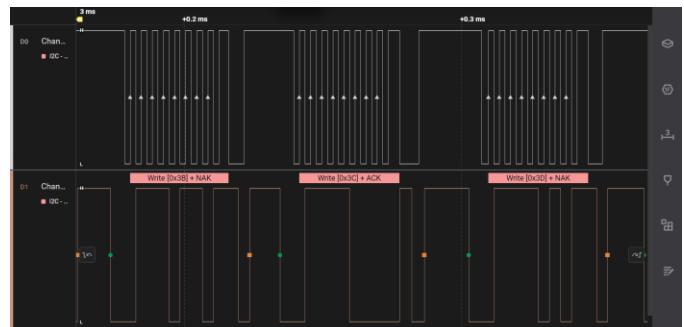


Figura 8. Señal resultante en búsqueda del ADDR correcto

Como se aprecia en la captura anterior el bus I²C mostrando la verificación de direcciones. Se observa que la dirección 0x3C responde con ACK, confirmando la conexión de la pantalla OLED, mientras que las direcciones 0x3B y 0x3D responden con NAK, indicando que no corresponden a ningún dispositivo en el bus.

e. Análisis de códigos hex para la pantalla OLED SSD1306

- Comando de encendido (0xAF)

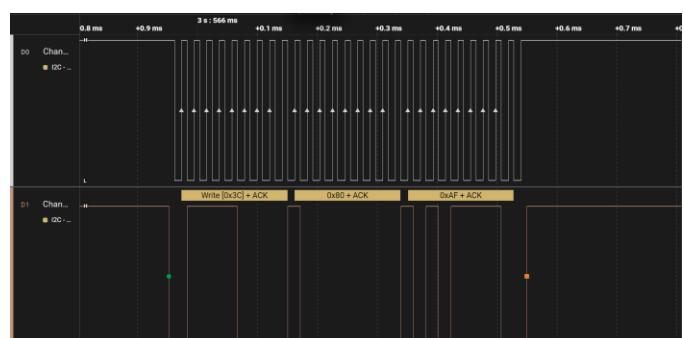


Figura 9. Señal resultante del encendido (0xAF)

En la captura se observa la comunicación entre el maestro (Raspberry Pi Pico 2W) y la pantalla OLED como esclavo a través del bus I²C. El maestro envía la dirección 0x3C seguida de un bit de escritura, a lo que el esclavo responde con un ACK, confirmando que está presente en el bus.

Posteriormente, se transmiten los bytes 0x80 y 0xAF, ambos también reconocidos con ACK por parte del esclavo. Estos datos corresponden a comandos de configuración de la pantalla, donde 0xAF es especialmente importante, ya que indica la instrucción para encender la pantalla OLED.

- Comando de apagar (0xAE)

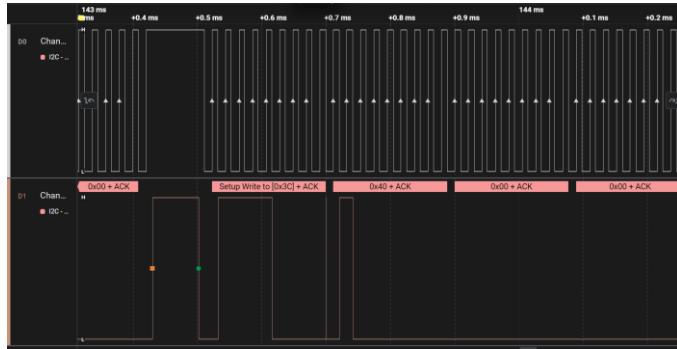


Figura 10. Señal resultante del encendido (0xAE)

En la captura se observa el inicio de la comunicación I²C. El maestro envía la dirección 0x3C con el bit de escritura, a lo cual el dispositivo esclavo (la pantalla OLED) responde con un ACK, confirmando que está presente en el bus.

Después, se realizan varias transmisiones de datos (0x00, 0xD5, 0x80), y en cada caso el esclavo responde nuevamente con ACK, lo que significa que todos los bytes fueron recibidos correctamente.

- Comando de contraste (0x81) (0x80)

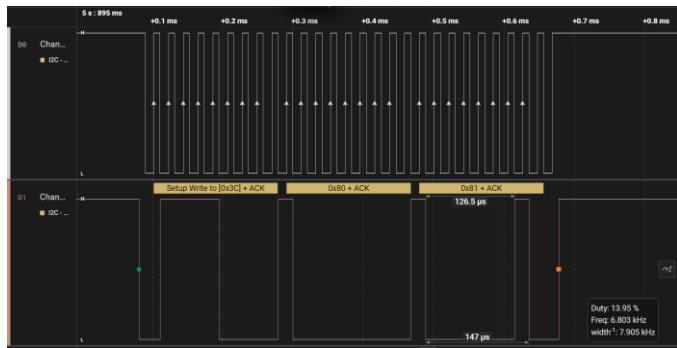


Figura 11. Señal resultante del contraste (0x81) (0x80)

En esta gráfica se observa una transmisión I²C donde el maestro inicia la comunicación enviando la dirección 0x3C con el bit de escritura. El esclavo (pantalla OLED) responde correctamente con un ACK, confirmando que está presente en el bus.

Después, se transmiten los datos 0x80 y 0x81, y en ambos casos el esclavo vuelve a contestar con ACK, lo que significa que está aceptando sin problemas la información enviada.

Además, en la parte superior se aprecia la señal de reloj (SCL), que marca los pulsos de sincronización. Los tiempos de cada pulso rondan entre 12 µs y 21 µs, lo que corresponde a una frecuencia aproximada de 47.6 kHz, con un ciclo útil (duty cycle) cercano al 43%.

- Comando invertir 1/0 (0xA7/0XA6)

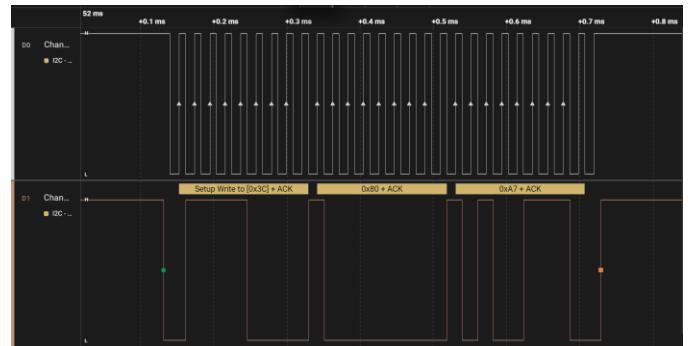


Figura 12. Señal resultante del comando invertir 1/0 (0xA7/0XA6)

En la captura se observa el inicio de la comunicación I²C con el dispositivo en la dirección 0x3C, el cual responde con ACK confirmado su presencia. Luego envía un byte de control (0x80) para indicar que los siguientes datos son comandos, seguido del comando 0xA7, que activa la inversión de colores en la pantalla OLED. En cada transmisión, el dispositivo confirma con ACK y finalmente se genera la condición de stop para cerrar la comunicación.

f. Prueba con el acelerómetro

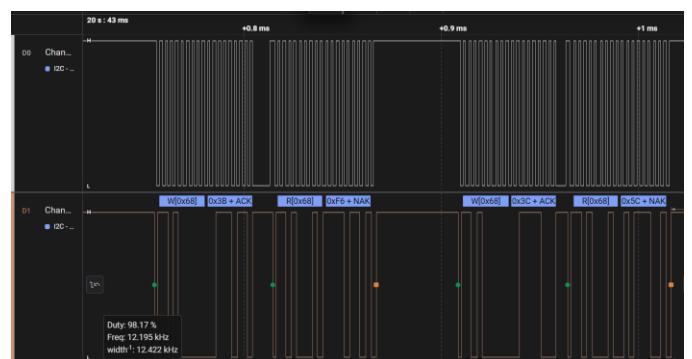


Figura 12. Señal resultante con el acelerómetro

El proceso de comunicación inicia cuando el microcontrolador envía una condición START, seguida de la dirección del acelerómetro en modo escritura. El dispositivo responde con un ACK, confirmando su presencia en el bus. A continuación, el microcontrolador indica el registro interno (0x44) desde el cual desea obtener los datos.

Sin liberar la línea de comunicación, se genera un Repeated START, esta vez en modo lectura. En ese momento, el acelerómetro comienza a transmitir los valores correspondientes a las mediciones de aceleración en sus ejes. Cada byte recibido es confirmado por el microcontrolador mediante un ACK, hasta llegar al último dato, donde responde con un NAK para señalar que no requiere más información. Finalmente, la comunicación concluye con una condición STOP.

III. ANÁLISIS DE RESULTADOS

- Las conexiones entre la Raspberry Pi Pico 2W, la pantalla OLED y el acelerómetro se realizaron adecuadamente, lo que permitió establecer la comunicación sin fallos en el bus I²C
- A través del analizador lógico se confirmaron las direcciones de los periféricos (0x3C para la pantalla OLED y 0x68 para el acelerómetro), verificando su presencia con respuestas ACK.
- El uso de la herramienta *Logic 2* permitió identificar los distintos elementos de cada trama (direcciones, bits de control, datos, ACK/NAK), facilitando la comprensión del protocolo.

IV. CONCLUSIONES

- La práctica permitió comprender de manera clara cómo funciona la comunicación I²C, no solo a nivel teórico, sino también observando directamente las señales en el analizador lógico.
- El uso de herramientas como *Logic 2* facilitó la identificación de cada parte de la trama (direcciones, datos, ACK/NAK), lo que hizo más sencillo interpretar lo que está presente en el bus.
- La experiencia práctica evidenció la importancia de las resistencias pull-up y de la sincronización entre reloj (SCL) y datos (SDA) para garantizar una comunicación estable.
- Se cumplió el objetivo de la práctica en el comprobar la correcta transmisión y recepción de datos en un bus I²C y aplicar en la interacción real con sensores y pantallas.

V. REFERENCIAS

- [1]. NXP Semiconductors. (2021). *I2C-bus specification and user manual*. Recuperado de <https://www.nxp.com/docs/en/user-guide/UM10204.pdf>
- [2]. MicroPython. (s.f.). *class I2C – a two-wire serial protocol*. Documentación oficial. Recuperado de <https://docs.micropython.org/en/latest/library/machine.I2C.html>
- [3]. InvenSense. (2013). *MPU-6000 and MPU-6050 product specification*. TDK InvenSense. Recuperado de <https://invensense.tdk.com/products/motion-tracking/6-axis/mpu-6050/>
- [4]. Adafruit. (2019). *Monochrome 128x64 I2C/SPI OLED display – SSD1306 datasheet*. Adafruit Industries. Recuperado de <https://cdn-shop.adafruit.com/datasheets/SSD1306.pdf>
- [5]. Saleae. (s.f.). *Logic 2 software user guide*. Saleae Inc. Recuperado de <https://support.saleae.com/logicsoftware/logic2>