

## Laboratorio 1 - Periféricos Embebidos

Johan Sebastian Mendieta Dilbert

Fecha de presentación 03/06/2025

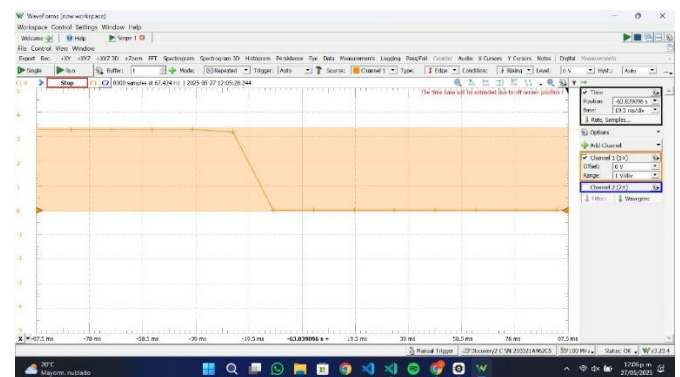
Se implementa el código para la STM32 NUCLEO-L476RG por medio del software STM32CubeMX. El código generado se hace para activar las interrupciones externas, la configuración del reloj del sistema. Dentro del código implementado, se hacen las modificaciones para que el LED ubicado en el pin PA5 haga su encendido solo en el flanco de bajada del botón ubicado en PC13, además, en el código implementado, se hace que el LED quede encendido durante 3 segundos, sin importar las condiciones del presionado del botón en ese tiempo. Además, se implementó en el código el envío del mensaje al momento de iniciar la placa y al momento de hacer la presión del botón en PC13, teniendo el mismo comportamiento del LED.

El código implementado se puede encontrar en el siguiente enlace:

[https://github.com/jomendietad/4100901-Lab\\_1-Embedded\\_Peripherals](https://github.com/jomendietad/4100901-Lab_1-Embedded_Peripherals)

### Primer Punto:

Para este punto se hace uso del Analog Discovery 2, para hacer la conexión del canal del osciloscopio del dispositivo a la salida del botón tanto sin el filtro como con el filtro RC.



Respuesta del botón con el filtro RC



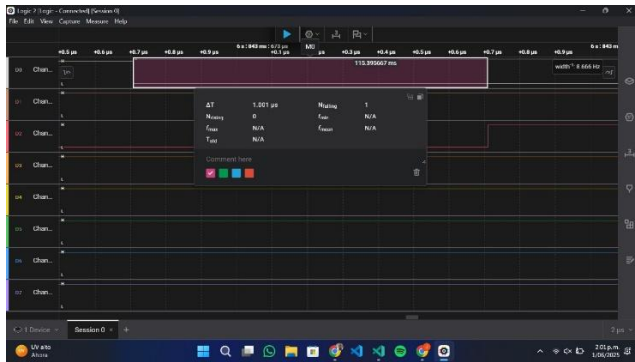
Respuesta del botón sin el filtro RC

Se observa que con el filtro RC el tiempo de establecimiento del nivel lógico es de aproximadamente 19 ms; mientras que sin el filtro RC su establecimiento fue en aproximadamente 2 ms.

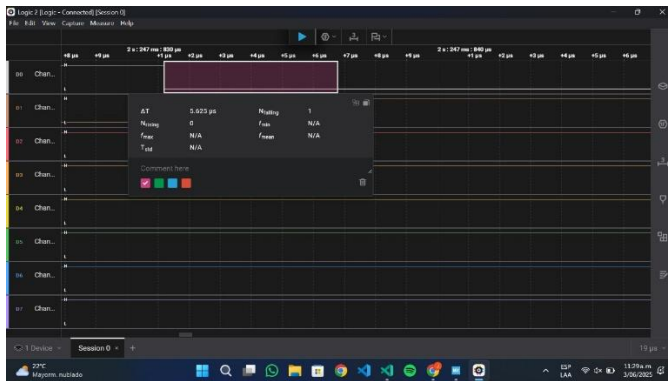
La diferencia notoria entre ambos es que el botón con el filtro RC tiene una respuesta más estable en el tiempo; mientras que sin el filtro RC, la respuesta es más rápida, pero a la vez, más inestable, produciendo que el nivel lógico pueda cambiar en cuestión de milisegundos.

## Segundo Punto:

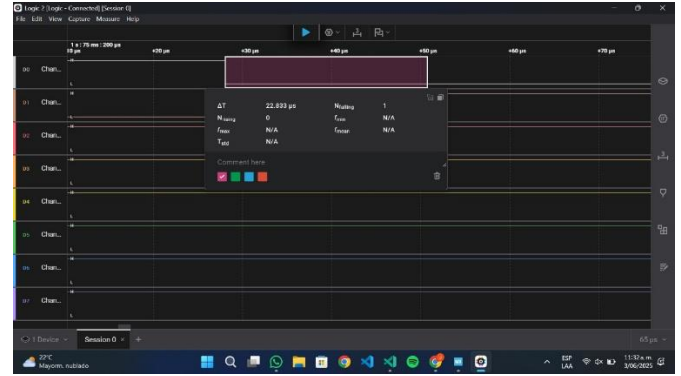
Para este punto se hace uso del Analizador Lógico, que se conecta a la salida del botón y al pin PA5 ligado al LED. Además, en el software de STM32CubeMX se va modificando la frecuencia del sistema dependiendo de lo solicitado.



Tiempo de respuesta a 80MHz (Canal 0: Botón; Canal 2: LED)



Tiempo de respuesta a 16MHz (Canal 0: Botón; Canal 1: LED)



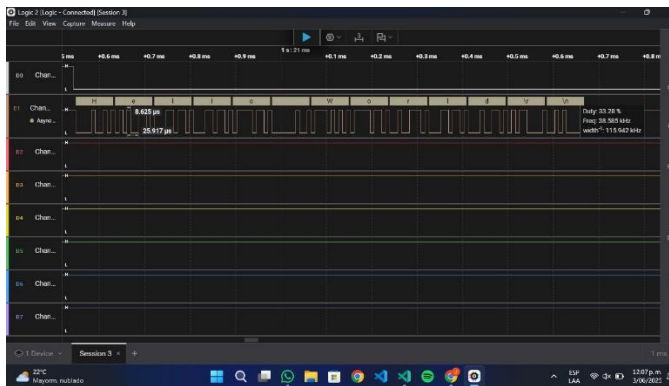
Tiempo de respuesta a 4MHz (Canal 0: Botón; Canal 1: LED)

Se observa que para una frecuencia de 4MHz, el tiempo de respuesta es de aproximadamente 22.833 us; para la frecuencia de 16MHz, el tiempo de respuesta es de aproximadamente 5.625 us; y para una frecuencia de 80MHz, el tiempo de respuesta es de aproximadamente 1.001 us.

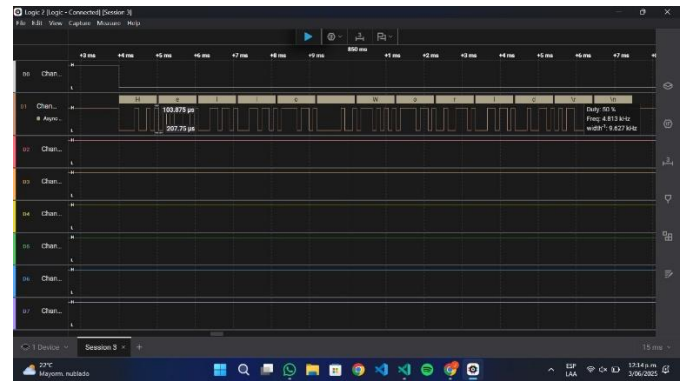
Según estos resultados, se puede observar que, a una mayor frecuencia del sistema, el tiempo de respuesta del sistema es más rápido, siendo esto bastante útil para diferentes tipos de sistemas, como lo son los sistemas en tiempo real.

## Tercer Punto:

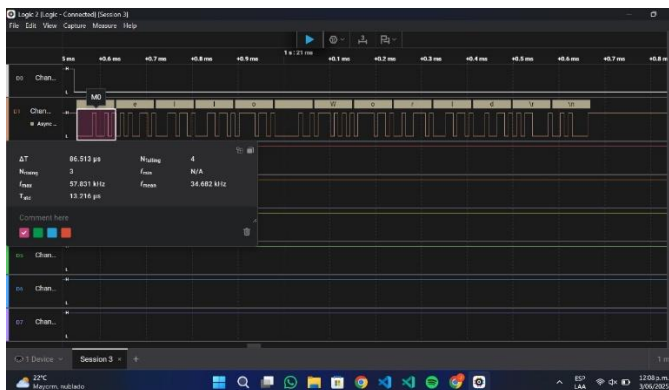
Para este punto se hace uso del Analizador Lógico, que se conecta a la salida del botón y al pin RX en la placa para observar la transmisión del mensaje. Además, en el software se hace la modificación para que el canal 1 pueda observar y comportarse como un lector de un monitor serial, y tanto el código como la configuración de este canal se modifican para cada tasa de baudios solicitada.



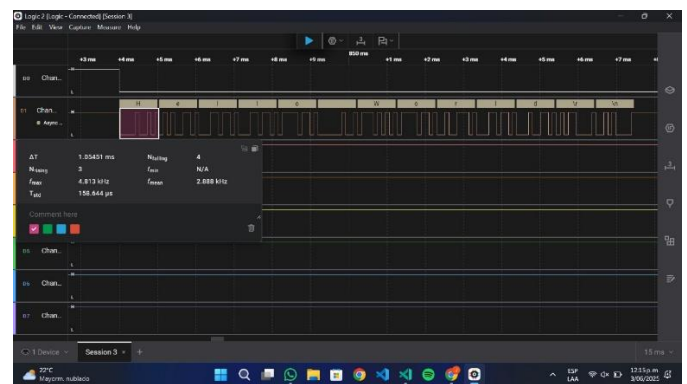
Tiempo de 1 bit a 115200 baudios



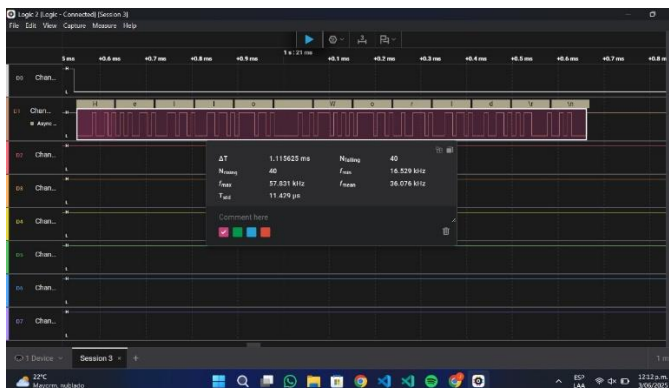
Tiempo de 1 bit a 9600 baudios



Tiempo de 1 byte a 115200 baudios

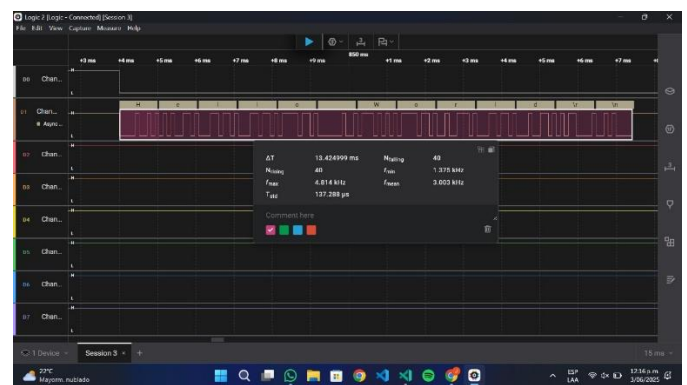


Tiempo de 1 byte a 9600 baudios



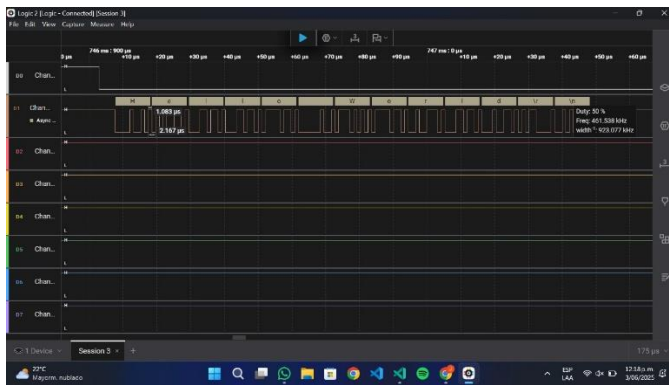
Tiempo del mensaje completo a 115200 baudios

Se puede observar que, a esta tasa de baudios, la transferencia de 1 bit se da a aproximadamente 8.625 us, la de 1 byte (1 carácter) a aproximadamente 86.513 us, y la del mensaje completo a 1.115625 ms aproximadamente.

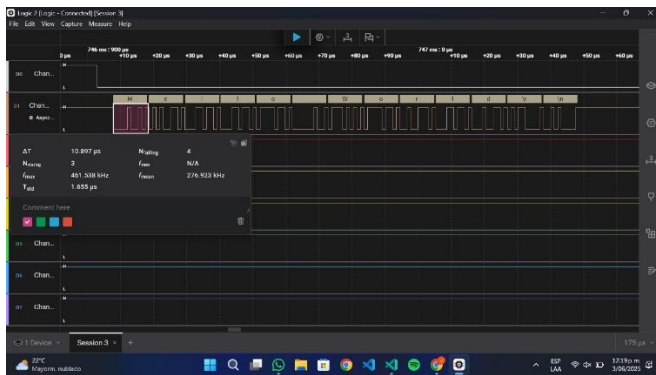


Tiempo del mensaje completo a 9600 baudios

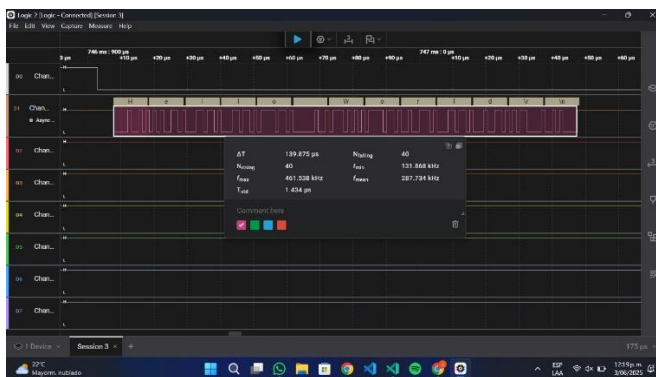
Se puede observar que, a esta tasa de baudios, la transferencia de 1 bit se da a aproximadamente 103.875 us, la de 1 byte (1 carácter) a aproximadamente 1.05451 ms, y la del mensaje completo a 13.424999 ms aproximadamente.



Tiempo de 1 bit a 921600 baudios



Tiempo de 1 byte a 921600 baudios



Tiempo del mensaje completo a 921600 baudios

Se puede observar que, a esta tasa de baudios, la transferencia de 1 bit se da a aproximadamente 1.083 us, la de 1 byte (1 carácter) a

aproximadamente 10.897 us, y la del mensaje completo a 139.875 us aproximadamente.

En este punto se puede observar que entre hay una relación entre la transferencia de 1 bit, 1 byte y del mensaje completo, teniendo en cuenta que 1 byte son 8 bits (1 carácter), y que el mensaje completo tiene 13 bytes, entonces las proporciones entre cada nivel se mantienen relativamente constantes, solamente que para cada byte hay un bit de inicio y 1 de stop.

Además, entre los resultados obtenidos, se puede deducir que una mayor tasa de baudios conlleva a una transmisión más rápida del mensaje; que puede ser muy bueno, pero hay que tener cuidado, porque esas tasas de baudios más altas no las manejan todos los dispositivos.