

# Laboratorio de Estructuras Computacionales

## Laboratorio 1

*Profesor:* Samuel Andres Cifuentes

*Monitor:* Tomas Jimenez Alvarez

---

### Objetivos

- Medir el tiempo de un bit y el tiempo total de un paquete de bits en una comunicación serie, utilizando diferentes tasas de baudios.
- Explorar la configuración del reloj en un microcontrolador, investigando diferentes configuraciones del HCLK, HSI Y PLL.
- Analizar la estabilidad y precisión de las diferentes configuraciones del reloj en términos de la transmisión de datos y la ejecución de tareas de temporización.
- Adquirir una comprensión más profunda de cómo la configuración del reloj afecta al rendimiento general del microcontrolador, incluyendo la velocidad de comunicación serie y la temporización de tareas.
- Trabajar y familiarizarse con los diferentes dispositivos de medición para calcular tiempos de señales y comprender cómo se relacionan estos tiempos con la configuración del reloj y la velocidad de transmisión de datos.

### Consulta previa

1. ¿Cuál es la diferencia entre el reloj interno HSI, el reloj interno MSI RC y el reloj externo HSE en términos de frecuencia y estabilidad?
2. ¿Qué es la comunicación serial y cómo afecta el baud rate en esta?
3. ¿Qué es el PLL (Phase-Locked Loop) y cómo se usaría si tengo un oscilador interno de alta velocidad HSI = 16 MHz y deseo obtener una frecuencia interna del reloj HCLK = 70 MHz ?

### Equipos a utilizar

- Tarjeta Núcleo-L476RG.
- Analizador lógico.
- Protoboard.
- LEDS (3).
- Resistencias (para protección de los LEDs).

## Procedimiento

Realizar la siguiente práctica siguiendo el debido procedimiento paso a paso en cada punto del laboratorio.

1. **Configuración del LD2 para alternar su nivel cada 1000 milisegundos:** Para lograr este objetivo, se deben realizar cuatro configuraciones diferentes, cada una de las cuales implica modificar la configuración del reloj del sistema. Estas configuraciones son necesarias debido a que la función HAL DELAY depende del contador de tiempo SysTick, que se ejecuta aproximadamente cada milisegundo y está relacionado directamente con la configuración del reloj. A continuación se describen una serie de pasos que deberá aplicar para la configuración del reloj:

Como primer paso debemos iniciar un proyecto en el STM32CubeIDE:

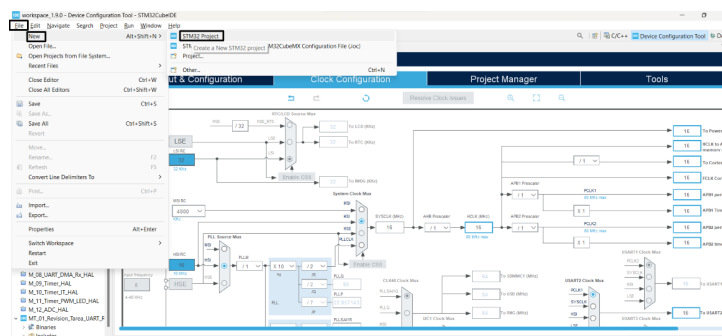


Figure 1: Iniciar proyecto en STM32CubeIDE.

Ahora debemos seleccionar la tarjeta que vamos a utilizar, que para nuestro caso será la tarjeta NUCLEO-L476RG:

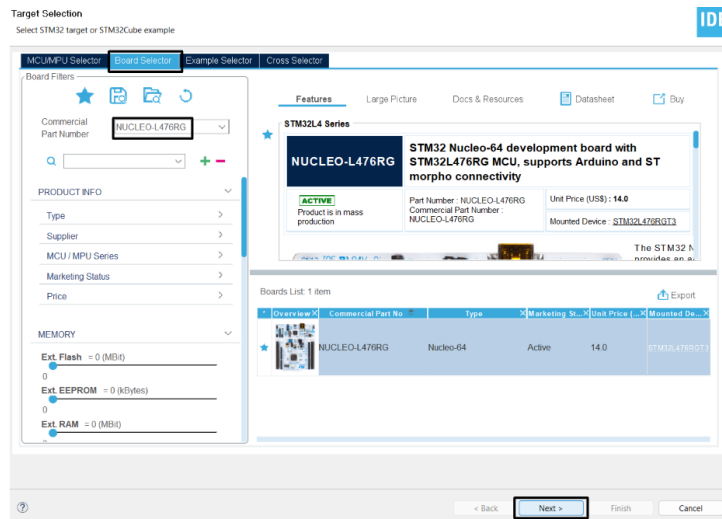
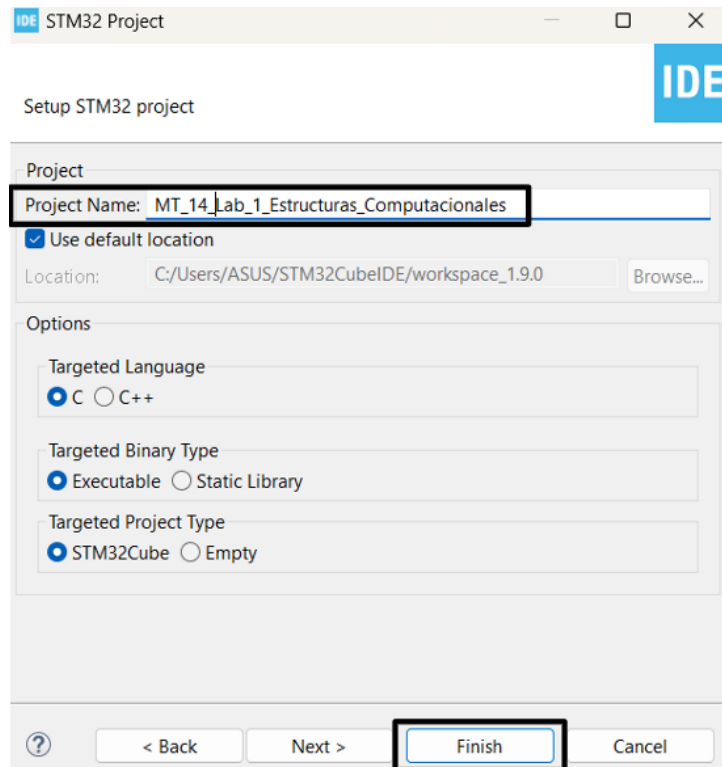


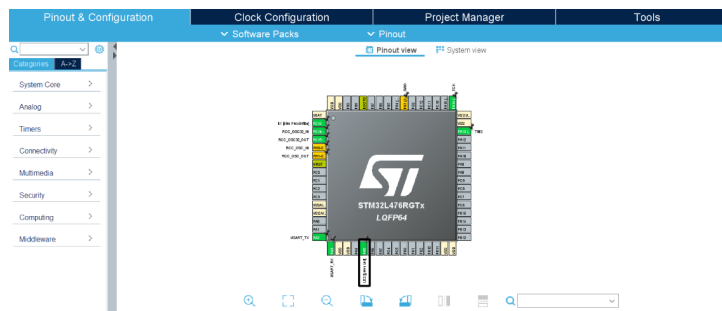
Figure 2: Selección de tarjeta.

Seleccionada la tarjeta asignar un nombre al proyecto, escoger el lenguaje en C, el tipo de proyecto objetivo como STM32Cube e inicializar todos los periféricos al seleccionar "Finish":



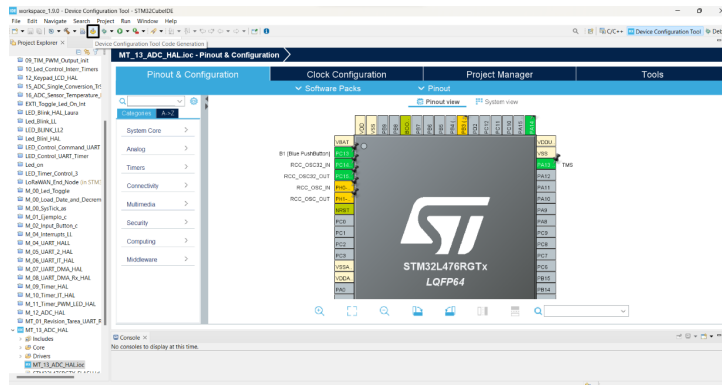
**Figure 3:** Configuración del proyecto.

¡Listo! Deberíamos estar viendo la interfaz gráfica de STM32CubeIDE que nos ayudará a configurar las opciones que necesitamos. Al seleccionar iniciar todos los periféricos se iniciará el Led 2 (PA5) de nuestra tarjeta automáticamente.



**Figure 4:** Interfaz gráfica de STM32.

Por último, generaremos el código con las configuraciones realizadas por la HAL presionando sobre el "Device Configuration Tool Code Generation"



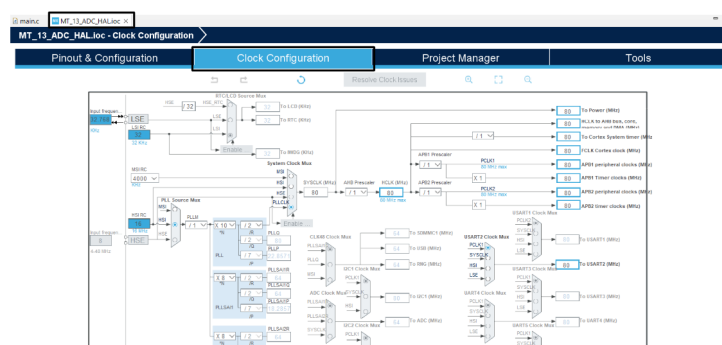
**Figure 5:** Generación de código.

Con cada uno de los pasos completados con éxito, se debió generar una plantilla de código en la que debemos aplicar los conocimientos de programación para alternar el nivel del LD2.



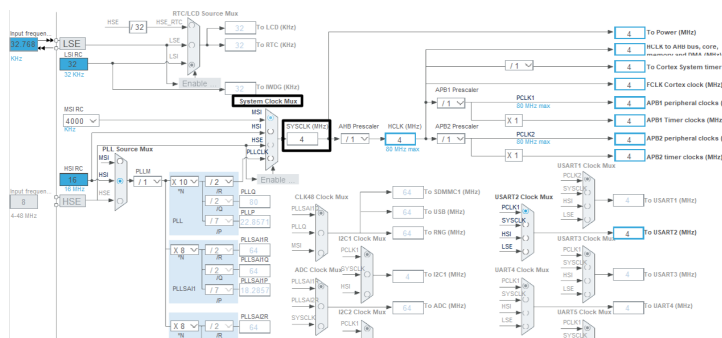
**Figure 6:** Toggle del Led 2.

Para cada uno de los siguientes numerales deberemos modificar la sección "Clock Configuration" en el archivo .ioc de nuestro proyecto



**Figure 7:** Clock Configuration.

(a) **Configuración con MultiSpeed Internal (MSI) a 4 MHz:** En este caso, se configurará el reloj del sistema para utilizar el oscilador interno de múltiple velocidad (MSI) a una frecuencia de 4 MHz. Esto se logra ajustando el System Clock Mux de acuerdo con esta configuración.



**Nota:** Repetir el procedimiento de la *Figure 8* para cada uno de los siguientes numerales configurándose de acuerdo a las especificaciones de cada punto.

- Anexar la evidencia necesaria en el informe para la sustentación de cada numeral.

a **Instalar el programa Pulse View:** Pulse View es una herramienta de software libre que nos permite el acceso a una amplia gama de dispositivos y decodificadores de protocolos para grabar, analizar, procesar y exportar datos analógicos y lógicos mediante una fácil interfaz gráfica. Para su instalación debemos:

Abrir el sitio oficial de [Sigrok](#) y descargar el instalador de *Pulse View* acorde a su computador.

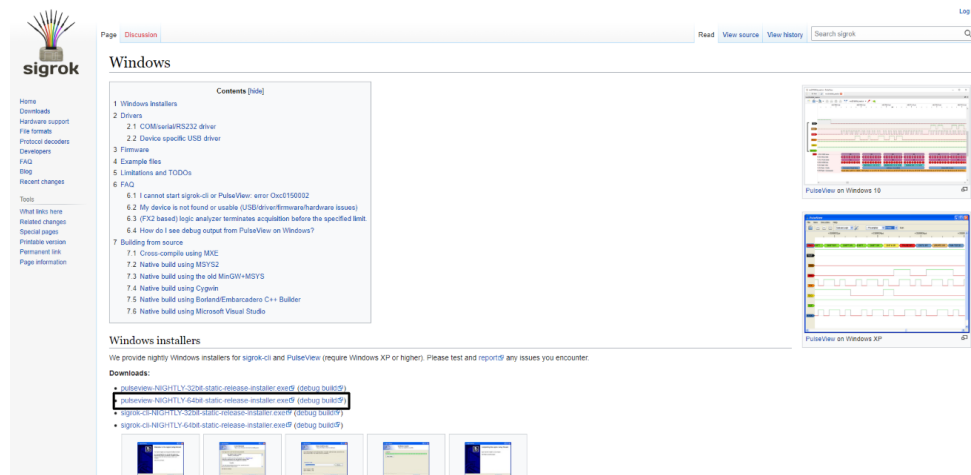


Figure 9: Website Sigrok.

Una vez instalado cerciórese que el programa se haya instalado con éxito.

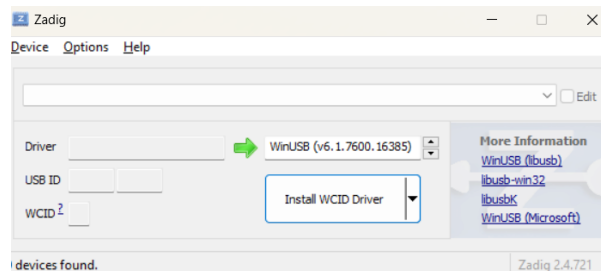
- b **Instalar el programa Zadig:** Es una herramienta de software que permite instalar controladores USB genéricos permitiéndote acceder a diferentes dispositivos por USB.

Abrir el sitio oficial de [Zadig](#) y descargar el instalador de *Zadig* acorde a su computador.



Figure 10: Website Zadig.

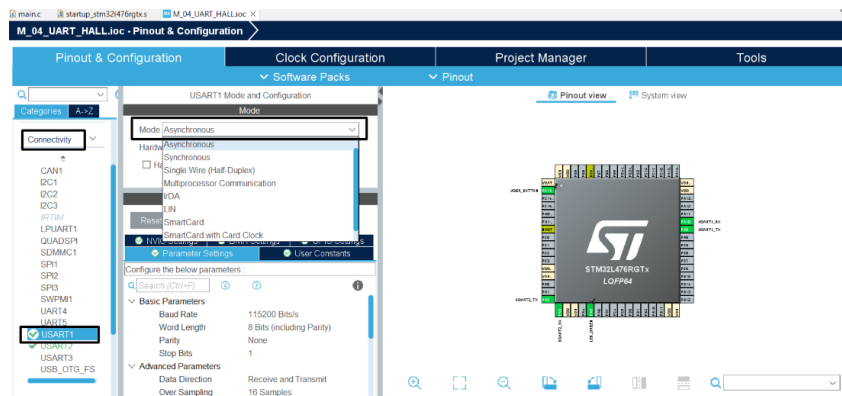
Una vez instalado el programa ábralo y presione sobre *Install WCID Driver*, así se instalará el debido drive para el analizador lógico que veremos más adelante.



**Figure 11:** Install WCID Driver.

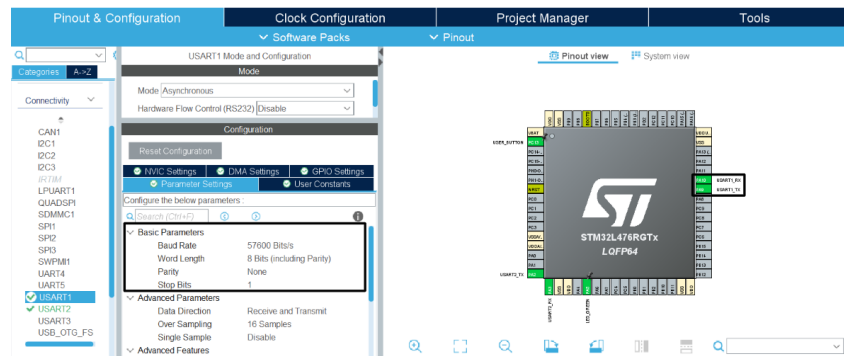
Con los pasos previos realizados, podremos iniciar la segunda parte de la práctica que a su vez consta de dos secciones: medir el tiempo de un sólo bit y medir el tiempo de un tren completo de bits transmitidos por UART.

Iniciaremos con la medición de un solo bit, para esto se debe crear un nuevo proyecto de STM32CubeIDE, explicado en los pasos anteriores. Realizado este paso, debemos iniciar el *USART1* en modo asíncrono de nuestra tarjeta.



**Figure 12:** USART1 Init.

Con el UART activado se debe configurar las siguientes opciones: word length de 8 bits, ningún bit de paridad, un bit de parada y, en un principio, el Baud Rate en 57600, el resto de configuraciones avanzadas se pueden dejar por default.



**Figure 13:** Configuración USART1.

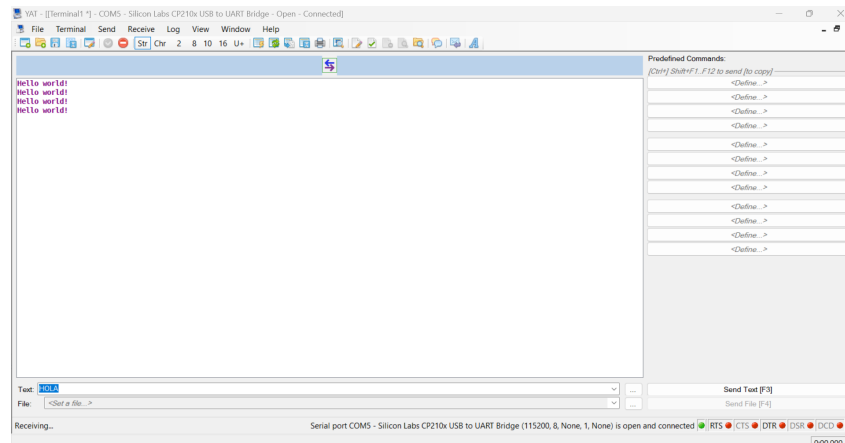
Deben generar el código y aplicar los conocimientos vistos previamente en la clase para transmitir mediante UART el string "Hello world!" con un delay de un segundo.



**Figure 14:** Código transmisión por UART.



Antes de realizar la medición con el analizador lógico, se debe verificar mediante el terminal de preferencia que se esté transmitiendo correctamente.



**Figure 15:** Transmisión por UART.

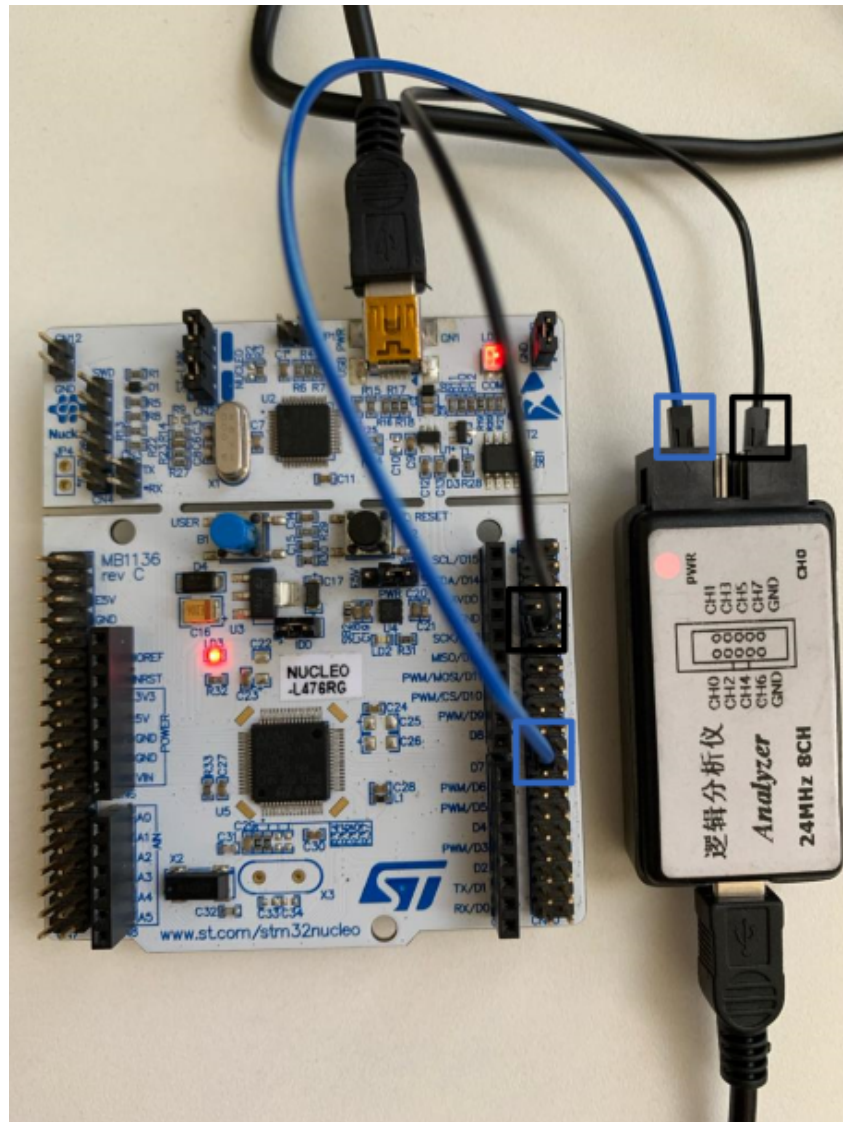
Si todo está correcto, conecte su analizador lógico a su computadora mediante el cable USB, debe observar que se encienda el led "PWR". Además conecte dos cables hembra-hembra, uno a tierra (GND) y otro para la medición de la señal (CH0).



**Figure 16:** Analizador lógico.

**Nota:** Asegúrese que el cable del analizador lógico NO sea sólo de alimentación, este debe estar comunicándose con su computadora.

Siguiendo la misma notación, conecte la otra punta de los cables a la tarjeta NUCLEO-L476RG.



**Figure 17:** Conexión STM-Analizador Lógico.

Ejecute el debugueador de STM para que se empiece a compilar y guardar el programa en su tarjeta.

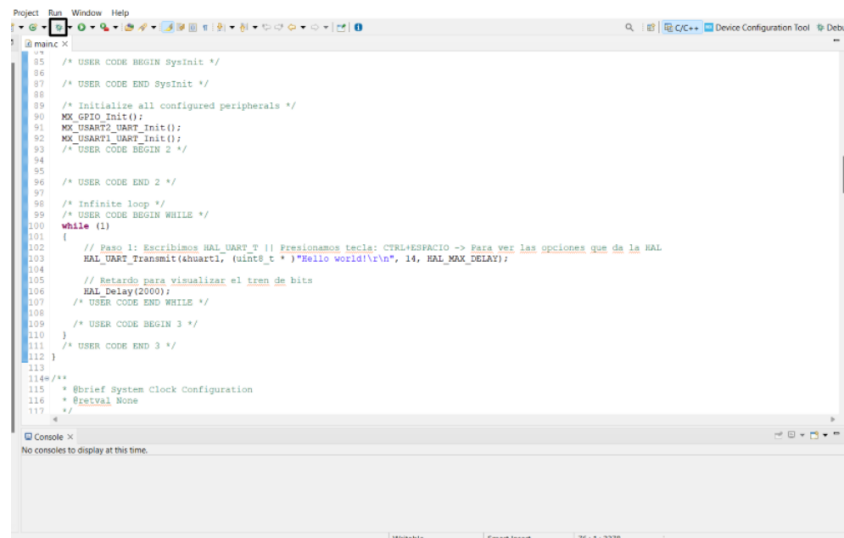


Figure 18: Debug.

**Nota:** En caso de presentar un error al conectarse a la tarjeta mediante el cable, se debe desconectar el analizador lógico, ejecutar el debugueador y conectar nuevamente el analizador lógico.

Si no presenta ningún error de compilación, el programa intercambiará a la ventana de depuración en el que se debe presionar el botón de "Resume" o mediante el atajo de teclado F8; en caso contrario, revisar qué errores se tienen y repetir los pasos anteriores.

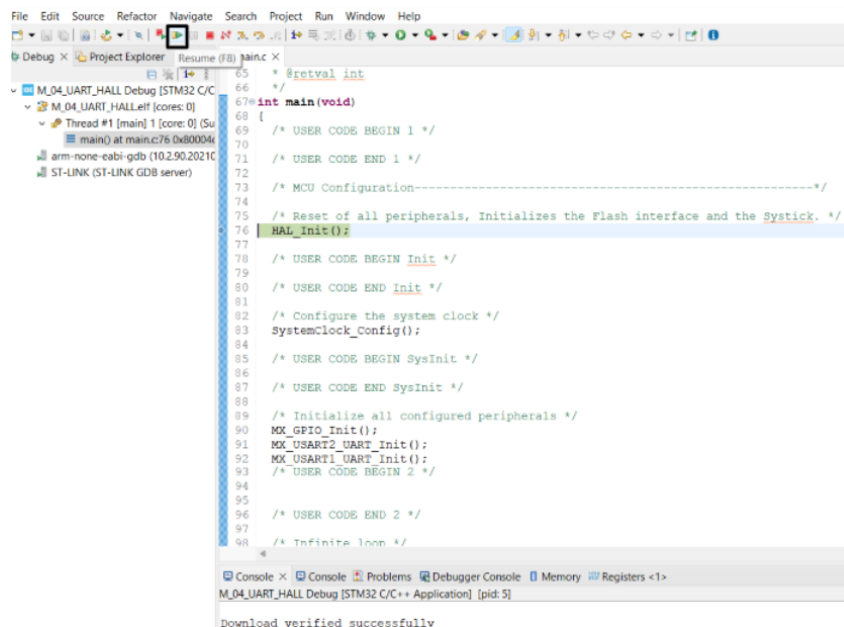


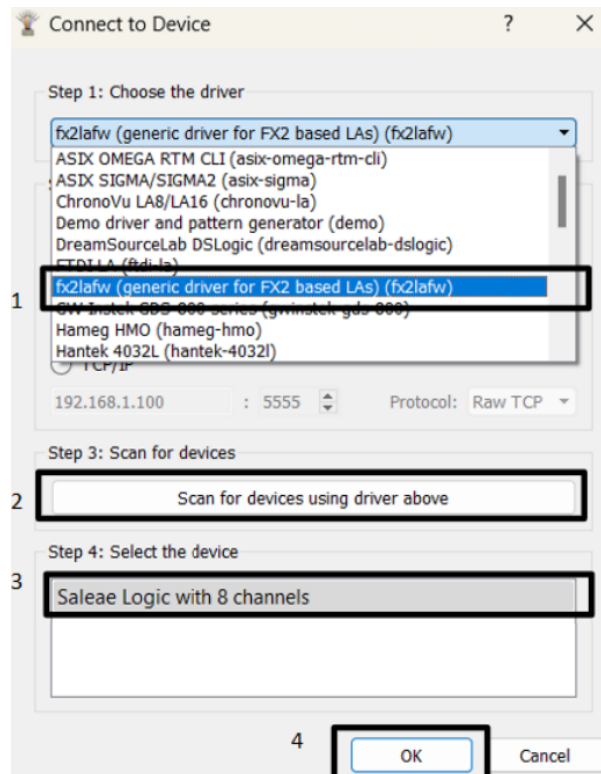
Figure 19: Ventana de depuración.

¡Listo! Tienes el UART de la tarjeta transmitiendo. Como paso siguiente se debe abrir el programa *Pulse View* (previamente instalado) y conectar el analizador lógico con el programa (en caso de no tenerlo conectado). Para conocer el estado en que se encuentra, se debe verificar en la parte superior *No Device*: No está conectado, *Saleae Logic*: Conectado.



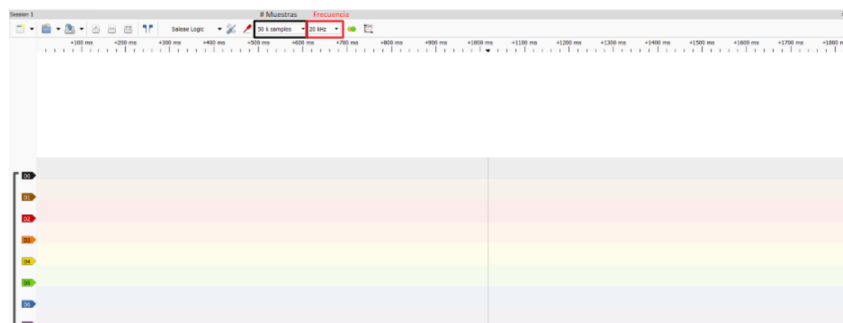
**Figure 20:** Conectividad analizador lógico.

En caso de no tener el analizador lógico se debe presionar sobre *No Device* y se desplegará una lista de opciones. Siga los pasos a continuación: 1. Escoger el Driver: fx2lafw (generic driver for FX2 based LAs) (fx2lafw). 2. Escanear los dispositivos configurados con el Driver seleccionado. 3. Escoger el dispositivo: Saleae Logic with 8 channels. 4. OK.



**Figure 21:** Conexión analizador lógico al Pulse View.

Con el analizador lógico conectado, deberás configurar el número de muestras que tomará y la frecuencia a la que el analizador lógico tomará cada una de estas.



**Figure 22:** Configuración entorno Pulse View.

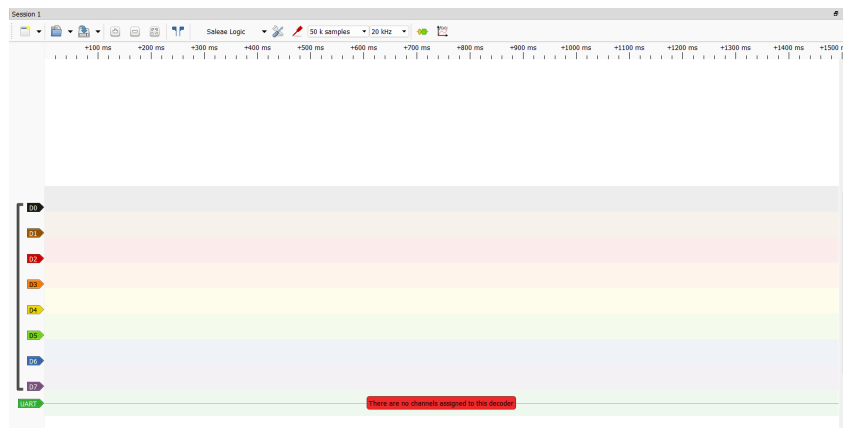
**Tip:** Recuerda tener en cuenta el Teorema de Nyquist.

Con la frecuencia y el número de muestras determinado, se debe configurar el protocolo que se usará, en nuestro caso será el UART.



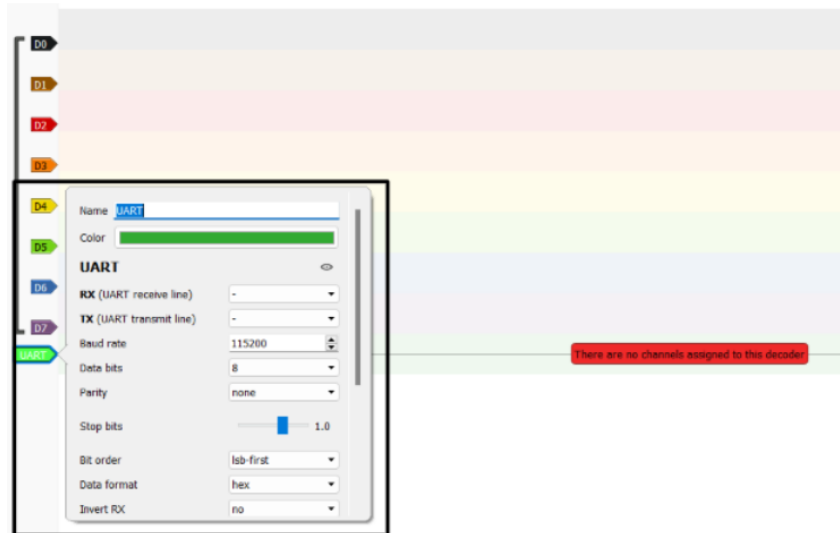
**Figure 23:** Configuración protocolo Pulse View.

Con cada uno de los pasos cumplidos se tendrá una sesión como la que se ilustra en la siguiente imagen:



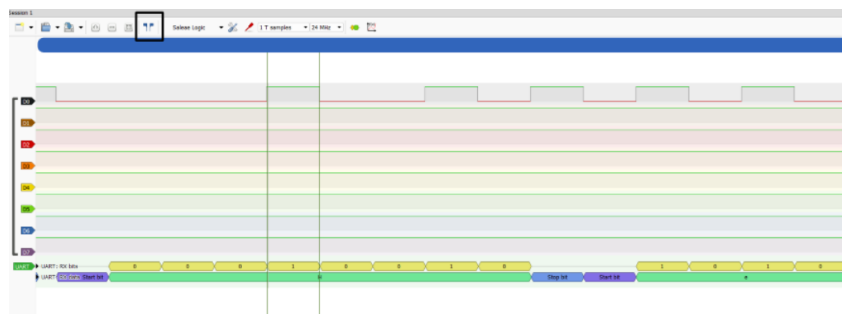
**Figure 24:** Sesión configurada Pulse View.

Por último, se deben modificar los parámetros del UART de la sesión. Esto se realiza pulsando sobre UART, acto seguido, se desplegará un menú de configuración en donde se deberá ajustar cada uno de los parámetros acorde a los de el entorno STM32CubeIDE.



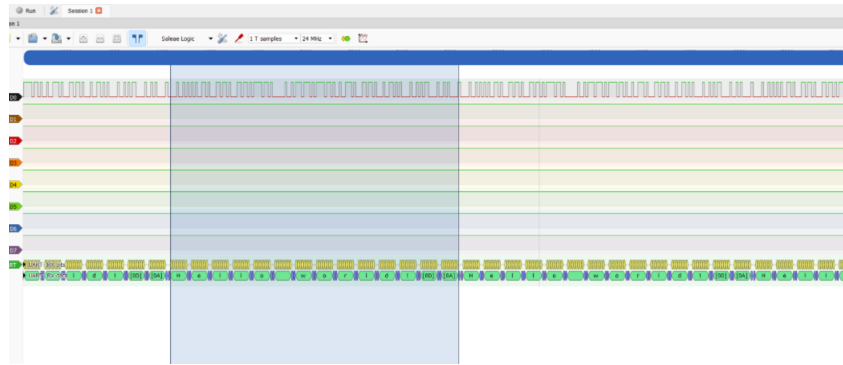
**Figure 25:** Configuración UART Pulse View.

Con cada uno de los pasos completados con éxito, se debe iniciar la lectura del analizador lógico presionando sobre *Run* en la parte superior izquierda de la interfaz. Además, se debe medir el tiempo de cada bit ayudándonos de las diferentes herramientas de medición del programa.



**Figure 26:** Herramientas de medición Pulse View.

Repita los pasos anteriores para la medición de un tren completo de bits. **Recuerde que deberá observar en su Pulse View el mensaje que transmitió mediante UART**, como se observa en la *figura 27*.



**Figure 27:** Medición tren de bits.

Para cada una de las siguientes tasas de baudio encuentre el tiempo del bit y su el tiempo total del tren de bits:

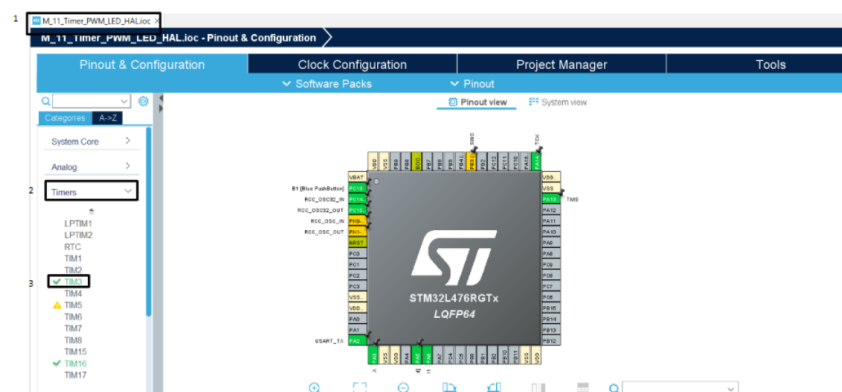
- (a) Baud Rate: 115200
- (b) Baud Rate: 256000
- (c) Baud Rate: 921600

Relacione estos datos en una tabla de resumen. Adicionalmente deberá demostrar estos resultados mediante pantallazos para cada uno de los procedimientos.

3. **PWM Dinámico en Dos LEDs Contrapuestos (Opcional +1):** Implementar una modulación de ancho de pulso (PWM, por sus siglas en inglés) dinámica en dos LEDs, donde el primer LED comenzará con un ciclo de trabajo del 0% (apagado) y se incrementará gradualmente hasta alcanzar el 100% (totalmente encendido), mientras que el segundo LED realizará una operación inversa.

Para realizar esta práctica podrá basarse del siguiente link de repositorio: [https://github.com/Tjimenez1303/Monitoria\\_estructuras\\_Computacionales/tree/main/M\\_11\\_Timer\\_PWM\\_LED\\_HAL](https://github.com/Tjimenez1303/Monitoria_estructuras_Computacionales/tree/main/M_11_Timer_PWM_LED_HAL) y siguiendo estos pasos:

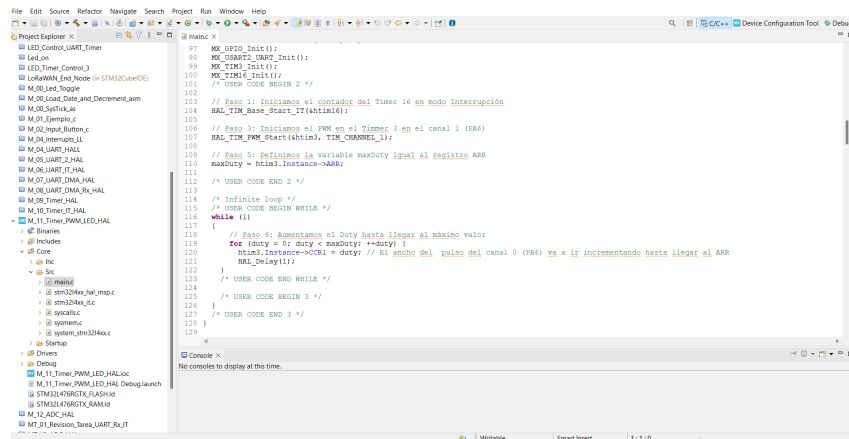
Cree un nuevo provecto e inicialice los timers y canales necesarios para la práctica



**Figure 28:** Inicialización de periféricos.

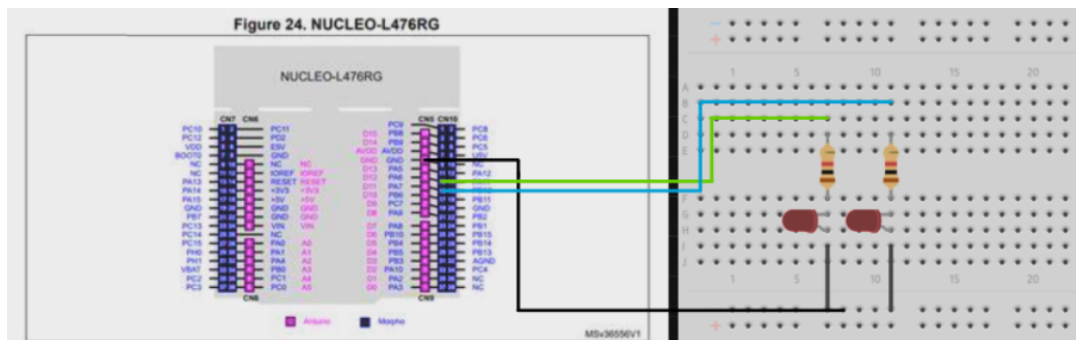
Implemente la lógica necesaria para cumplir el objetivo basándose en el repositorio mencionado anteriormente





**Figure 29: Programación PWM.**

Realice las debidas conexiones de los LEDs a la tarjeta. Puede basarse en la imagen siguiente:



**Figure 30: Conexiones Tarjeta-Protoboard.**

Por último, mida el tiempo de 3 diferentes ciclos útiles para cada Led anexando el analizador lógico a su protoboard siguiendo los pasos ya descritos anteriormente.

## Cuestionario

1. Presentar un informe de manera organizada con los resultados de la práctica uno de estructuras computacionales. El informe se estructura en las siguientes secciones: introducción, marco teórico, análisis de resultados, que incluye capturas de pantalla del analizador lógico, así como tablas pertinentes y pantallazos. Además, deberá escribir unas conclusiones ( mínimo dos conclusiones).”