

# Lab Session 3: PWM

---

**Estimated time:** 1.5h (1 session)

## Descripción

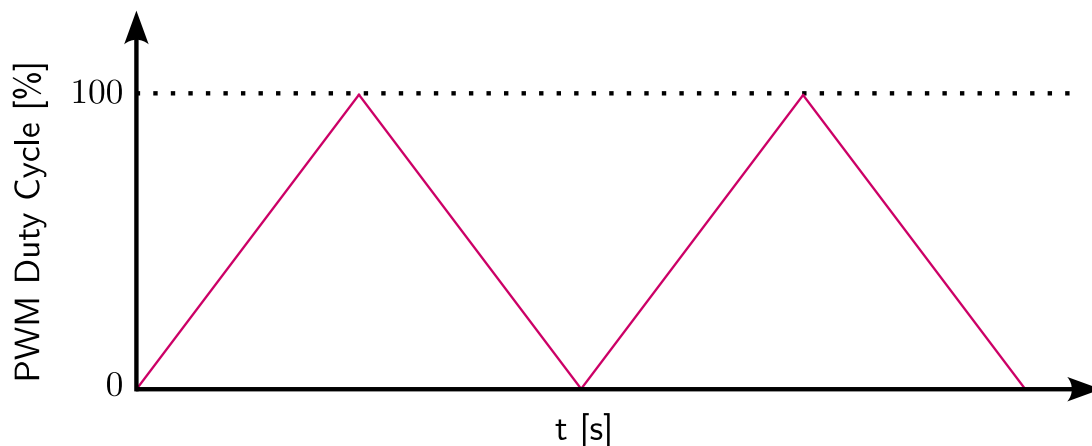
El objetivo de esta práctica es entender la generación y uso de las señales PWM, así como el control de un motor utilizando este tipo de señales. Para esto, se cuenta con un motor de Corriente Continua (CC o DC - Direct Current - en inglés), una controladora de motores (también conocido comúnmente como driver) basada en el puente H L298n, y un M5Core2.

## 1. Trabajando con hardware real

### Parte 1 - Generación de PWM con LED

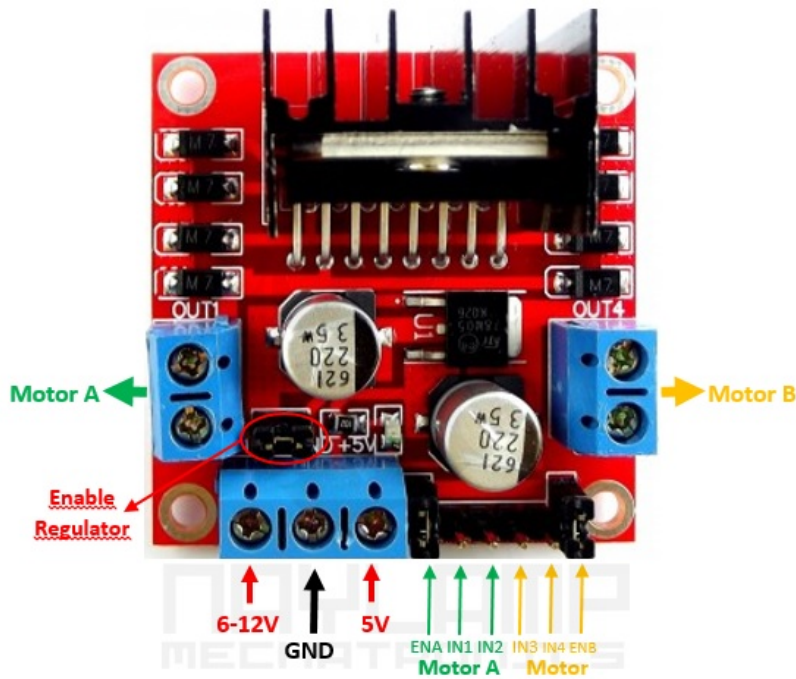
Realizar una prueba de generación de una señal PWM para iluminar más o menos un LED conectado al pin 14 del M5Core2. [Este tutorial](#) explica cómo se puede realizar.

El LED se tiene que encender de forma progresiva desde el estado completamente apagado (PWM de 0%) hasta el estado completamente encendido (PWM de 100%). Una vez en el 100%, deberá apagarse también de forma progresiva (desde un PWM de 100% hasta un PWM de 0%), siguiendo en ambos casos un perfil tipo "rampa", primero creciente y luego decreciente, como el de la siguiente figura:



### Parte 2 - Generación de PWM con motor CC

La siguiente figura muestra la ubicación de los conectores de la controladora:



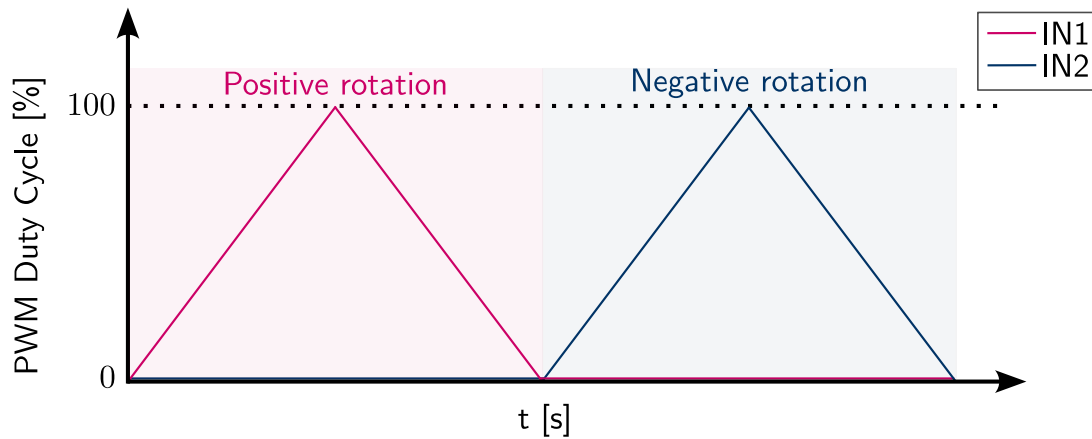
Esta controladora permite controlar hasta 2 motores CC de forma simultánea. En la figura anterior, los puertos en verde se corresponderían con el motor A y los amarillos con el motor B.

Realizar el conexionado del M5Core2 con la controladora de motores (puente H L298N) de la siguiente manera:

- GPIO14 a pin IN1 en la controladora. Esta salida PWM permitirá establecer la potencia del sentido de giro positivo (sentido contrario a las agujas del reloj).
- GPIO13 a pin IN2 en controladora. Esta salida PWM permitirá establecer la potencia del sentido de giro negativo (sentido de las agujas del reloj).
- GND y VS (en controladora) conectado a una fuente de alimentación de 9-12V.
- GND (M5Core2) a GND (Controladora). Es imprescindible conectar las masas de la controladora y del microcontrolador, de lo contrario no se conocería la referencia de las señales PWM.
- Conectar el motor a la controladora en los terminales del motor A.
- Es imprescindible que el pin ENA de la controladora esté alimentado, de lo contrario la salida del motor A estaría deshabilitada (para eso se dispone de un [jumper](#) que permite alimentar el pin ENA con el regulador de 5V que incluye el propio driver).

Un tutorial sencillo sobre el uso de este driver puede encontrarse en [este enlace](#)

Una vez conectado el microcontrolador con el motor, modificar el código del tutorial, para que el motor gire acelerando en el sentido de giro positivo desde un PWM de 0% hasta uno de 100%, y luego decelerando desde el PWM de 100% hasta el 0%. Cuando se haya hecho la aceleración y deceleración en el sentido de giro positivo, se deberá hacer lo mismo pero en sentido negativo de giro y seguir realizando esta secuencia de forma indefinida en el tiempo. Como se indica en la siguiente imagen:



### PREGUNTA:

- ¿Qué ocurre cuando se cambia la frecuencia del PWM? Probar con frecuencias entre 1 y 10 kHz.
- ¿Qué ocurre cuando se cambia la resolución del PWM? Probar con resoluciones entre 8 y 10 bits.
- Si se cambia la resolución, ¿hay que cambiar algo más del código o basta con cambiar el valor de la resolución en la función `ledcSetup`?

## 2. Trabajando en simulación

### Parte 1 - Generación de PWM con LED

El ejercicio se puede realizar exactamente igual en simulación y en real.

### Parte 2 - Generación de PWM con motor CC

Este ejercicio no se puede realizar igual en simulación que en real porque en wokwi no existe el componente de motor de corriente continua. Sin embargo, existen varias formas de trabajar en simulación para hacer el prototipado del código para el motor.

Una forma sencilla sería utilizar 2 LEDs, donde el LED1 simularía el comportamiento del motor cuando gira en sentido horario, mientras el LED2 simularía el comportamiento del motor cuando gira en sentido antihorario. De esta forma, se pueden programar las señales PWM de ambos LEDs de forma similar a las señales PWM que se van a enviar a IN1 e IN2 en la controladora (que harán que el motor gire en sentido horario o antihorario).

### INFO:

- Para comprobar el correcto funcionamiento del código, es muy importante comprobar que ambos LEDs no están encendidos a la vez.

### PISTA:

- Al trabajar con PWM en wokwi, el factor de tiempo real puede disminuir considerablemente, especialmente si se generan varias PWMs y se cambia el ciclo de trabajo en cada paso del loop. Para evitar esto, es aconsejable cambiar el ciclo de trabajo de forma escalonada cada varios pasos del loop o con un tiempo de espera lo suficientemente grande.