

# PWM para el control de giro de un motor DC

Miguel Angel Romero Renteria

Marisol Alejandre Flores

Noemi Hernández Oliva

Instituto Politécnico Nacional

CECyT 2 "Miguel Bernard"

[mromeror2300@alumno.ipn.mx](mailto:mromeror2300@alumno.ipn.mx)

[malejandref@ipn.mx](mailto:malejandref@ipn.mx)

[nhernandezo@ipn.mx](mailto:nhernandezo@ipn.mx)

Referencia de este artículo [1].  
(Noto Sans SemiCondensed Light no. 10)

## RESUMEN

Este trabajo presenta la implementación de la técnica de modulación por ancho de pulso (PWM) para el control de motores de corriente directa (DC) en sistemas embebidos orientados a la robótica móvil. Se utilizó un microcontrolador junto con componentes de bajo costo y simuladores gratuitos como TinkerCAD y Arduino IDE, lo que permitió validar el comportamiento del motor. El enfoque se centró en regular la velocidad, controlar la orientación del giro del motor y ajustar su aceleración de forma adecuada, utilizando señales PWM generadas por el microcontrolador. Se documentó el uso de un puente H para invertir el giro del motor y se analizaron las señales mediante osciloscopio, observando cómo el ciclo de trabajo y la frecuencia afectan directamente la respuesta física del sistema. La propuesta demuestra que es posible lograr un control funcional, estable y replicable en robótica móvil, utilizando electrónica y programación en C/C++.

## ABSTRACT

This work presents the implementation of the Pulse Width Modulation (PWM) technique for controlling direct current (DC) motors in embedded systems oriented toward mobile robotics. A microcontroller was used along with low-cost components and free simulators such as TinkerCAD and Arduino IDE, which allowed the motor's behavior to be validated. The focus was on regulating speed, controlling the motor's rotation direction, and adjusting its acceleration precisely using PWM signals generated by the microcontroller. The use of an H-bridge to reverse the motor's rotation was documented, and the signals were analyzed using an oscilloscope, observing how the duty cycle and frequency directly affect the physical response of the system. The proposal demonstrates that it is possible to achieve functional, stable, and replicable control in mobile robotics using electronics and programming in C/C++.

## 1. Introducción

El control de motores de corriente directa (DC) mediante la técnica de modulación por ancho de pulso (Pulse Width Modulation, por sus siglas en inglés) se ha convertido en una herramienta clave dentro de la robótica móvil. Esta técnica permite regular la velocidad del motor de forma precisa y progresiva, lo que facilita tanto la aceleración como la desaceleración según las necesidades del sistema. Al trabajar con microcontroladores y componentes accesibles como motores DC y puentes H, es posible lograr arranques suaves, evitar picos de corriente y mantener la estabilidad del sistema. Todo esto contribuye no solo a la eficiencia energética, sino también a la vida útil del motor, lo cual es especialmente importante cuando se busca construir robots móviles funcionales y educativos.

En sistemas embebidos, el PWM se utiliza para generar señales digitales que simulan una regulación analógica de velocidad. El ciclo de trabajo define cuánto tiempo la señal permanece en estado alto, y la frecuencia determina que tan rápido se repite ese ciclo. Al aplicar esta técnica para controlar el giro y la dirección del motor —por ejemplo, usando un puente H— se puede invertir el sentido de rotación. Con ayuda de un osciloscopio, es posible visualizar los pulsos y entender cómo la programación compilada al microcontrolador se traduce al movimiento del motor. Este tipo de análisis permite afinar el comportamiento del robot, logrando desplazamientos más precisos.

Estos principios se conectan directamente con lo que otros investigadores han desarrollado en sus diferentes áreas. Por ejemplo, estudios sobre técnicas básicas de PWM —como PWM senoidal, trapezoidal o escalonado— muestran cómo cada variante tiene sus ventajas en términos de distorsión armónica y pérdidas de conmutación (Sahoo et al., 2012; Panda Tripathi, 2006). Este proyecto se enfoca en hacer comparativas que ayudan a decidir técnicas para proyectos de diversos investigadores.

También hay investigaciones que van más allá, explorando inversores multilevel y métodos para minimizar pérdidas de conmutación. En aplicaciones de mayor potencia, estas técnicas

—como las que proponen Al-Fatlawi Al-Mhana (2022) y Miao et al. (2010)— permiten reducir el estrés térmico en los componentes y mejorar el rendimiento general. Aunque no son el foco de este trabajo, sirven como referencia para etapas posteriores o para quienes escalan sus prototipos.

Los esquemas híbridos y las estrategias de eliminación armónica seleccionada han demostrado ser útiles en inversores conectados a red, donde la calidad de la señal y la compatibilidad electromagnética son críticas. En este artículo se trabaja en un entorno educativo, lo cual resulta útil para escalar algunos proyectos basándose en esto.

En cuanto al control de corriente, estudios como el de Zhao Jiang (2017) ofrecen modelos que ayudan a

mantener la estabilidad en sistemas con cargas variables, algo que también aplica cuando se integran motores DC en plataformas móviles que enfrentan distintos tipos de esfuerzo mecánico. Y en el campo de la tracción eléctrica, investigaciones sobre space-vector PWM y control de tensión de enlace DC —como las de Qamar et al. (2021)— muestran cómo optimizar el rendimiento en vehículos eléctricos, aportando criterios que se adaptan a proyectos de menor escala como el que aquí se presenta.

Además, existe una línea de investigación sobre el uso de señales PWM como herramienta de diagnóstico. El trabajo de Yi et al. (2025) demuestra que es posible detectar fallas en motores síncronos de imanes permanentes a partir de la lectura de pulsos PWM. Esta idea se conecta directamente con el uso del osciloscopio en este proyecto, y permite extender los experimentos del proyecto, y permite extender los experimentos hacia el monitoreo predictivo y el análisis de condición en robots móviles educativos.

Todos estos aportes respaldan las decisiones técnicas y metodológicas tomadas en este artículo: desde la elección de frecuencia PWM para Arduino, el uso del puente H para invertir el giro, la instrumentación con osciloscopio para validar la forma de onda. Las comparativas de técnicas PWM más básicas fueron clave para definir lo que se utilizó en este proyecto. Pues ayudan a entender qué tipo de señal resulta más conveniente para lograr un control del motor DC.

## 2. Materiales y Métodos

Para la experimentación se utilizaron los siguientes materiales:

- Microcontrolador Arduino UNO
- Motorreductor de doble eje (Motor DC de 5V)
- Protoboard y cables de conexión
- Software de simulación (Tinkercad)
- Puente H (L293D)
- 2 × resistencia fija de 10k $\Omega$
- 2 × Botones pulsadores

En cuanto a los métodos, se llevaron a cabo las siguientes acciones:

- Simulación electrónica de PWM:
  - Se realizó un circuito en simulador educativo (Tinkercad) para observar el comportamiento de la señal PWM en un osciloscopio.
- Generación de señal PWM para controlar la velocidad del motor:
  - Se programó el código del microcontrolador en Arduino IDE para generar distintas señales con ciclos de trabajo variables, controlando así la velocidad del motor.
- Implementación de lógica de control para invertir el sentido de giro:
  - Se utilizó un puente H para invertir la polaridad aplicada al motor, permitiendo cambiar el sentido de giro desde el código.

- Pruebas de funcionamiento variando el ciclo de trabajo del PWM.
- Observación del comportamiento del motor en respuesta a los cambios de señal. Pruebas de validación

Para verificar el adecuado comportamiento del sistema, realizaron distintas pruebas electrónicas y de programación en un simulador educativo, lo que permitió analizar el comportamiento del sistema PWM antes de llevarlo a conexiones físicas.

### 1. Simulación electrónica inicial

- Antes de llegar a la programación, fue fundamental conocer el funcionamiento electrónico detrás de la tarjeta microcontroladora para utilizar la PWM. A este tipo de pruebas se le denomina "Software-in-the-Loop" (SIL), a partir de las cuales se verifica el funcionamiento del sistema con el uso de simuladores en software, permitiendo ajustar valores previos al desarrollo del sistema físico. Dichas pruebas son las siguientes.

### 2. Simulación de control de motor DC por microcontrolador

- Prueba para determinar el funcionamiento electrónico de Arduino UNO en los pines especiales PWM: En las figuras 1 y 2, se muestran las conexiones de un circuito básico para un motor DC.

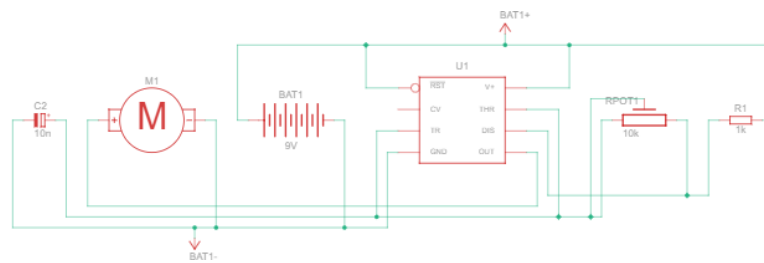


Figure 1: Circuito básico de simulación PWM con un motor DC.

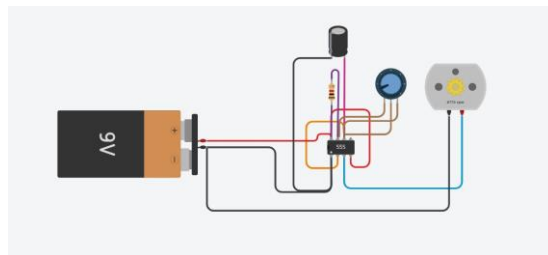


Figure 2: Variación de ciclo de trabajo en simulador educativo.

Después de entender la prueba inicial, el siguiente paso fue aplicar la lógica de programación para girar el motor con el 20%, como se observa en la figura 3. Además, se muestra una forma de arrancar el motor evitando picos de voltaje mediante un ciclo for, como se ve en la figura 4, específicamente en una placa microcontroladora Arduino UNO.

Figure 3: Circuito básico de simulación PWM con un motor DC (Programado).

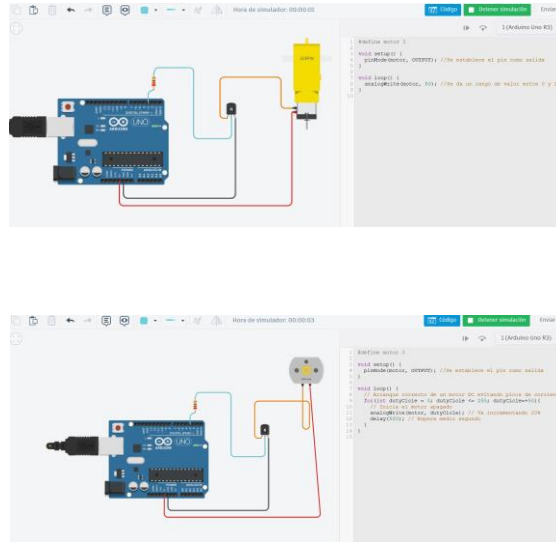


Figure 4: Arranque correcto de un motor DC evitando picos de voltaje.

- Después de realizar pruebas con el PWM, se observó que, para mayor estabilidad, el motor debería estar funcionando al 80%. Esto, en términos de código, se refiere a mandar señales de 200 en analogWrite. Además, se implementó un puente H para la lógica del control de giro de un motor, fundamental para el control de direcciones en robótica móvil; es decir, con la finalidad de poder invertir el sentido de un motor, como se observa en la figura 5.

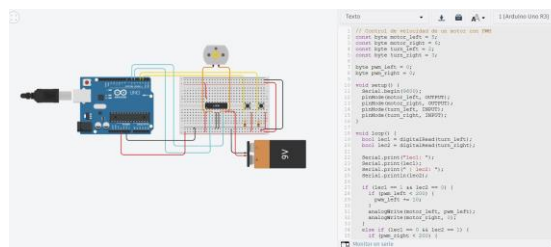
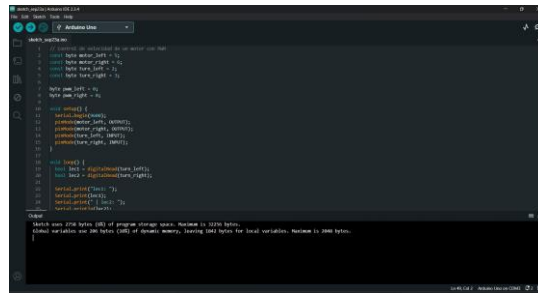


Figure 5: Control de giro de un motor por PWM y puente H.

### 3. Código propuesto y conexiones físicas del Arduino UNO

- Cuando el entorno de trabajo está señalado, se puede pasar a trabajar con conexiones físicas y a la programación del microcontrolador, creando un nuevo sketch en Arduino IDE, como se muestra en la figura 6, y cargando el código anterior a este. Posteriormente, se procede a realizar las conexiones físicas,

como se muestra en la figura 7.



```

1 // Definición de pines de salida para el motor con PWM
2 #define PIN_MOTOR_1 9
3 #define PIN_MOTOR_2 10
4 #define PIN_MOTOR_3 11
5 #define PIN_MOTOR_4 12
6
7 // Definición de constantes para el motor
8 #define FREQ_PWM 1000 // Frecuencia de PWM en Hz
9 #define DUTY_CYCLE 0.5 // Ciclo de trabajo (50%)
10
11 // Definición de constantes para el motor
12 #define PIN_MOTOR_1 9
13 #define PIN_MOTOR_2 10
14 #define PIN_MOTOR_3 11
15 #define PIN_MOTOR_4 12
16
17 // Definición de constantes para el motor
18 #define FREQ_PWM 1000 // Frecuencia de PWM en Hz
19 #define DUTY_CYCLE 0.5 // Ciclo de trabajo (50%)
20
21 // Definición de constantes para el motor
22 #define PIN_MOTOR_1 9
23 #define PIN_MOTOR_2 10
24 #define PIN_MOTOR_3 11
25 #define PIN_MOTOR_4 12
26
27 // Definición de constantes para el motor
28 #define FREQ_PWM 1000 // Frecuencia de PWM en Hz
29 #define DUTY_CYCLE 0.5 // Ciclo de trabajo (50%)
30
31 // Definición de constantes para el motor
32 #define PIN_MOTOR_1 9
33 #define PIN_MOTOR_2 10
34 #define PIN_MOTOR_3 11
35 #define PIN_MOTOR_4 12
36
37 // Definición de constantes para el motor
38 #define FREQ_PWM 1000 // Frecuencia de PWM en Hz
39 #define DUTY_CYCLE 0.5 // Ciclo de trabajo (50%)
40
41 // Definición de constantes para el motor
42 #define PIN_MOTOR_1 9
43 #define PIN_MOTOR_2 10
44 #define PIN_MOTOR_3 11
45 #define PIN_MOTOR_4 12
46
47 // Definición de constantes para el motor
48 #define FREQ_PWM 1000 // Frecuencia de PWM en Hz
49 #define DUTY_CYCLE 0.5 // Ciclo de trabajo (50%)
50
51 // Definición de constantes para el motor
52 #define PIN_MOTOR_1 9
53 #define PIN_MOTOR_2 10
54 #define PIN_MOTOR_3 11
55 #define PIN_MOTOR_4 12
56
57 // Definición de constantes para el motor
58 #define FREQ_PWM 1000 // Frecuencia de PWM en Hz
59 #define DUTY_CYCLE 0.5 // Ciclo de trabajo (50%)
60
61 // Definición de constantes para el motor
62 #define PIN_MOTOR_1 9
63 #define PIN_MOTOR_2 10
64 #define PIN_MOTOR_3 11
65 #define PIN_MOTOR_4 12
66
67 // Definición de constantes para el motor
68 #define FREQ_PWM 1000 // Frecuencia de PWM en Hz
69 #define DUTY_CYCLE 0.5 // Ciclo de trabajo (50%)
70
71 // Definición de constantes para el motor
72 #define PIN_MOTOR_1 9
73 #define PIN_MOTOR_2 10
74 #define PIN_MOTOR_3 11
75 #define PIN_MOTOR_4 12
76
77 // Definición de constantes para el motor
78 #define FREQ_PWM 1000 // Frecuencia de PWM en Hz
79 #define DUTY_CYCLE 0.5 // Ciclo de trabajo (50%)
80
81 // Definición de constantes para el motor
82 #define PIN_MOTOR_1 9
83 #define PIN_MOTOR_2 10
84 #define PIN_MOTOR_3 11
85 #define PIN_MOTOR_4 12
86
87 // Definición de constantes para el motor
88 #define FREQ_PWM 1000 // Frecuencia de PWM en Hz
89 #define DUTY_CYCLE 0.5 // Ciclo de trabajo (50%)
90
91 // Definición de constantes para el motor
92 #define PIN_MOTOR_1 9
93 #define PIN_MOTOR_2 10
94 #define PIN_MOTOR_3 11
95 #define PIN_MOTOR_4 12
96
97 // Definición de constantes para el motor
98 #define FREQ_PWM 1000 // Frecuencia de PWM en Hz
99 #define DUTY_CYCLE 0.5 // Ciclo de trabajo (50%)
100

```

Figure 6: Captura de pantalla del software Arduino IDE después de compilar el código.

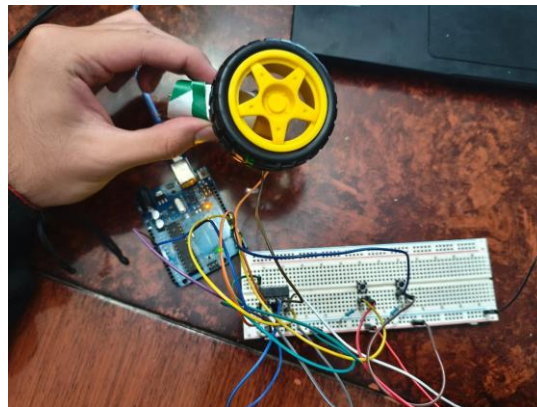


Figure 7: Conexiones físicas del microcontrolador y el motor.

### 3. Análisis de resultados

Durante la experimentación y exploración de la técnica PWM, se observó que, en lugar de enviar un voltaje constante en un estado lógico alto o bajo, este alterna a una frecuencia dada según lo requerido. Esto pudo analizarse con solo conocer los componentes de la simulación del control PWM electrónico, pues al utilizar un temporizador 555 junto con un capacitor, se indica que no hay electrónica analógica de por medio. De este modo, el ciclo de trabajo se reduce al tiempo en que la señal permanece en estado lógico alto; es decir, si se mantiene en estado alto, el voltaje se mantiene constante. Si está la mitad del tiempo encendido y la otra mitad apagado, se observa un 50% de ciclo de trabajo.

Lo mencionado fue más claro al conectar un osciloscopio a la simulación del circuito electrónico, pues como se observa en las figuras 8 y 9, al tener un tiempo por división de 100 microsegundos, se pueden observar claramente los intervalos de tiempo al controlar el motor, y esto se ve reflejado en las RPM de este. Al tener el potenciómetro en su resistencia más baja, las RPM disminuían y los cambios entre estado lógico alto y bajo eran algo considerables. Por otra parte, al tenerlo con su resistencia más alta, el motor giraba con RPM más rápidas y en el osciloscopio no se distinguían los cambios.

El ciclo de trabajo esta dado en la ecuacion (1), mientras que en la (2) corresponde con el periodo de la señal.

$$D = \frac{t}{T} \tag{1}$$

$$T = \frac{1}{f} \tag{2}$$

- $D$ : Ciclo de trabajo
- $t$ : Tiempo en estado lógico alto
- $T$ : Periodo de la señal
- $f$ : Frecuencia de la señal

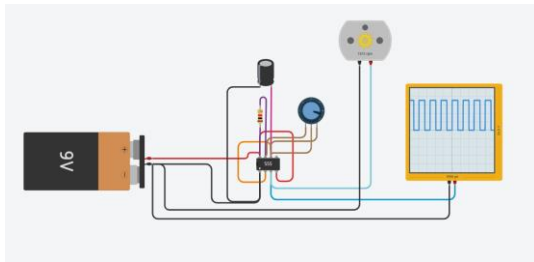


Figure 8: Circuito con osciloscopio (Potenciómetro con baja resistencia)

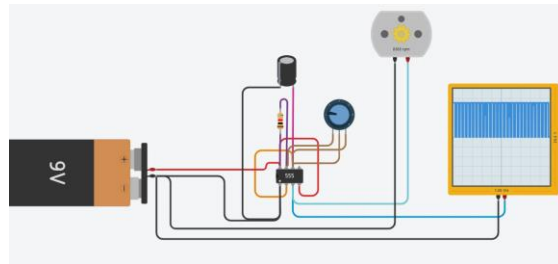


Figure 9: Circuito con osciloscopio (Potenciómetro con alta resistencia)

Además de esto, se observó en el osciloscopio que el Arduino UNO trabaja a frecuencias mayores que las de la simulación electrónica. Por lo tanto, se tuvo que ajustar el tiempo por división del osciloscopio a 750 microsegundos para analizar mejor el comportamiento de las señales, como se observa en la figura 10.

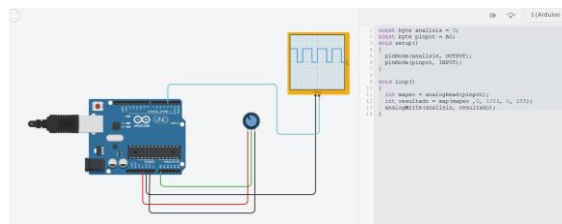


Figure 10: Señales PWM de Arduino UNO y potenciómetro

De igual manera, se utilizó el osciloscopio en el circuito con el puente H para analizar los pulsos PWM que llegan al motor, con el objetivo de controlar su giro e invertirlo. Al girar hacia la izquierda, el osciloscopio marcaba las señales en dirección descendente, lo que indicaba una inversión en la polaridad de los pulsos, y las RPM del motor eran negativas, como se muestra en la figura 11. Esta condición se logró al modificar la lógica de control en el microcontrolador, enviando señales opuestas a los terminales del puente H. Por otra parte, al girar hacia la derecha, las señales PWM se observaban en dirección ascendente, reflejando una polaridad positiva en los pulsos y un incremento en las RPM del motor como se muestra en la figura 12.

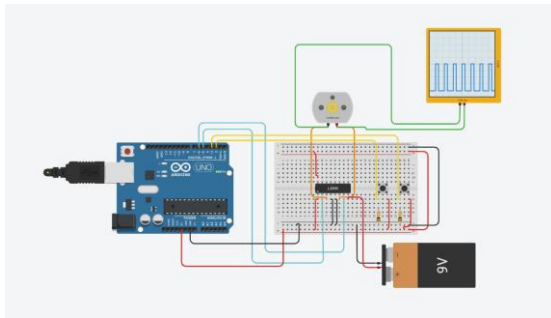


Figure 11: Motor girando hacia la izquierda

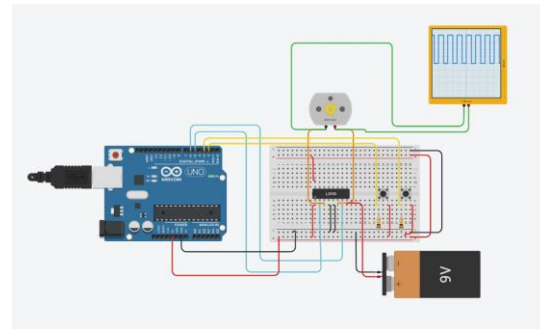


Figure 12: Motor girando hacia la derecha

Por último, se realizaron las pruebas físicas, realizando las conexiones diseñadas en el entorno de simulación TinkerCAD. Se conectó el microcontrolador a los pines correspondientes del puente H. Los pulsadores se integraron en el circuito con la misma disposición utilizada en la simulación (Se utilizaron resistencias de 10 k $\Omega$  conectadas entre los pines de entrada digital y Vcc para configurar los pulsadores en modo pull-up), permitiendo modificar el comportamiento de la lectura de los pulsadores de forma manual. Esto fue importante para que la programación en Arduino IDE se comportara de manera adecuada en el entorno físico, confirmando que las señales PWM generadas por el microcontrolador eran correctamente interpretadas por el puente H y reflejadas en el movimiento del motor como se muestra en las figuras 13 y 14. Además, se observó que la respuesta del microcontrolador ante la interacción con los pulsadores funcionaba adecuadamente, observando cómo se modificaba el ciclo de trabajo y la dirección del giro en tiempo real. La conexión se realizó sobre una protoboard, respetando la distribución de componentes simulada.

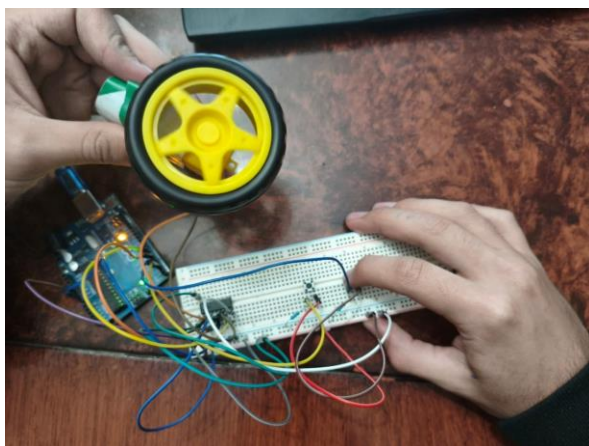


Figure 13: Control de giro de un motor por

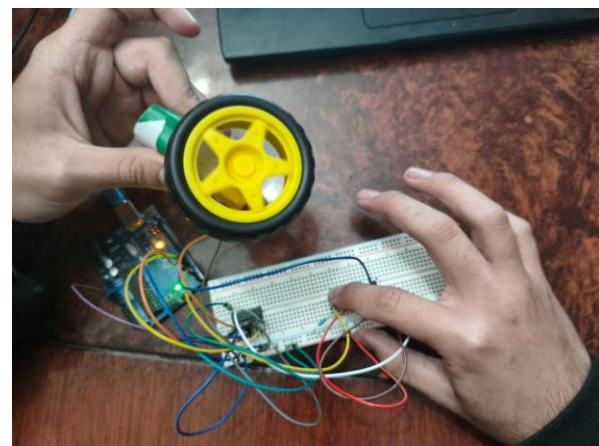


Figure 14: Control de giro de un motor por

#### 4. Conclusiones

El uso de la técnica de modulación por ancho de pulso (PWM) para el control de motores de corriente directa (DC) en sistemas embebidos resultó ser eficiente para aplicaciones en robótica móvil, pues se requiere de pocos componentes para replicar esta técnica, además de poder aplicarse en microcontroladores low cost con pruebas SIL previas a la implementación física.

Cabe resaltar que se comprobó que el PWM permite regular la velocidad del motor de forma adecuada, facilitando tanto el arranque como la inversión de giro mediante el uso de un puente H (L293D), lo cual se observó con apoyo del osciloscopio y la lectura de señales de este.

La implementación del Arduino y componentes de bajo costo permitió construir un buen sistema adaptable a distintas necesidades. Además, se observó que el comportamiento del motor responde a lo definido en el código, como el ciclo de trabajo.

La investigación logró demostrar que es posible controlar un motor DC con precisión y estabilidad utilizando electrónica, programación y simuladores gratuitos como TinkerCAD y Arduino IDE. El enfoque de este trabajo es importante para entender correctamente lo que se aplica en la robótica móvil, para el control de orientación del giro de un motor, el control preciso de su aceleración y la estabilidad general al evitar los picos de voltaje. Al implementar un control por PWM desde un microcontrolador, se logra que el motor responda de forma progresiva y controlada, lo que permite maniobras más suaves y una mejor adaptación a los cambios de orientación o velocidad. Este tipo de control es clave cuando se busca que el robot móvil se mantenga en pie, avance con precisión y reaccione ante las instrucciones programadas.

#### Referencias bibliográficas

- Yi, C.-P., Lin, Y.-J., Ho, P.-J., & Yang, S.-C. (2025). Magnet fault diagnosis for permanent magnet synchronous motor based on flux estimation with PWM voltage measurement. *IEEE Transactions on Industrial Electronics*, 72(2), 2100–2110. <https://doi.org/10.1109/TIE.2024.3426031>
- Sahoo, S. K., Ramulu, A., Batta, S., & Duggal, S. (2012). Performance analysis and simulation of three phase voltage source inverter using basic PWM techniques. In *IET Chennai 3rd International on Sustainable Energy and Intelligent Systems (SEISCON 2012)* (pp. 1–7). <https://doi.org/10.1049/cp.2012.2223>
- Panda, R., & Tripathi, R. K. (2006). A symmetrical hybrid sine PWM switching technique for full bridge inverters. In *2006 India International Conference on Power Electronics* (pp. 345–348). <https://doi.org/10.1109/IICPE.2006.4685395>
- Al-Fatlawi, E. H., & Al-Mhana, T. H. M. (2022). Comparison between 3-level single phase NPC and SI-NPC Multilevel Inverter Based on PD PWM & POD PWM Techniques for Photovoltaic Applications. In *2022 2nd International Conference on Advances in Engineering Science and Technology (AEST)* (pp. 639–644). <https://doi.org/10.1109/AEST55805.2022.10412885>
- Miao, C.-x., Shi, L.-p., & Wang, H.-y. (2010). Research on Switching Loss Minimized PWM Method for Flying

- 
- Capacitor Three-Level Inverters. In 2010 International Conference on Electrical and Control Engineering (pp. 4140–4143). <https://doi.org/10.1109/ICECE.2010.1005>
- Zhang, Y., Xu, D., Yan, C., & Zou, S. (2015). Hybrid PWM Scheme for the Grid Inverter. *IEEE Journal of Emerging and Selected Topics in Power Electronics*, 3(4), 1151–1159. <https://doi.org/10.1109/JESTPE.2015.2451159>
- Zhao, J., & Jiang, Q. (2017). Research on a new control strategy for three-phase current-source PWM rectifier. In 2017 Eighth International Conference on Intelligent Control and Information Processing (ICICIP) (pp. 157–161). <https://doi.org/10.1109/ICICIP.2017.8113934>
- Qamar, H., Qamar, H., & Ayyanar, R. (2021). Performance Evaluation of Space Vector PWM Methods with DC Link Voltage Control for EV/HEV Powertrains. In *IECON 2021 – 47th Annual Conference of the IEEE Industrial Electronics Society* (pp. 1–6). <https://doi.org/10.1109/IECON48115.2021.9589333>

## Referencia del artículo

Romero, M., Alejandre, M. & Hernández, N. (noviembre - diciembre, 2025). PWM para el control de giro de un motor DC. *Boletín UPIITA. año 20, (111) 2025*  
<https://www.boletin.upiita.ipn.mx/index.php/ciencia/1101-cyt-numero-111/2459-pwm-para-el-control-de-giro-de-un-motor-dc>