Instituto Tecnológico de Estudios Superiores de Monterrey

Fundamentación de robótica & Manchester Robotics

Manchester Robotics: Challenge 3

Profesores:

Rigoberto Cerino Jiménez

Dr. Mario Martinez

Integrantes

Daniel Castillo López - A01737357

Emmanuel Lechuga Arreola - A01736241

Paola Rojas Domínguez - A01737136

6 de Marzo de 2025

Índice

Resumen

El resumen presenta una síntesis de los aspectos más relevantes del proceso de investigación y desarrollo llevado a cabo, destacando los objetivos, la metodología empleada y los principales resultados obtenidos.

El Challenge 3 tiene como objetivo que implementemos nodos de ROS para controlar la velocidad de un motor de corriente continua (DC). Para ello, se utiliza una ESP 32 como microcontrolador, junto con un puente H y modulación por ancho de pulso (PWM) para regular la potencia suministrada al motor.

Se hace uso de Micro-Ros, una extensión de ROS 2 optimizada para microcontroladores, permitiendo la comunicación eficiente entre sistemas embebidos y una computadora externa. Además, se detalla el funcionamiento del ADC, el puente H y la generación de señales PWM en la ESP 32.

La metodología aplicada incluye la configuración del hardware, la programación de nodos en ROS para la regulación de velocidad y validación del sistema, Finalmente, se presentan los resultados obtenidos, evaluando la precisión del control de velocidad y proponiendo mejoras para optimizar el rendimiento del sistema.

Objetivos

En esta sección se presentan el objetivo general y los objetivos específicos del reto, los cuales están enfocados en el diseño e implementación de un sistema de control para un motor de corriente continua.

- Implementar un sistema de control para un motor de corriente continua utilizando Micro ROS, una ESP32 y un puente H.
- Regular la velocidad del motor mediante modulación por ancho de pulso (PWM).
- Facilitar la integración de sistemas embebidos con ROS 2 para la comunicación eficiente entre dispositivos.
- Analizar el funcionamiento del sistema a través de pruebas y medición de resultados.

Introducción

La introducción presenta el tema de investigación, proporcionando el contexto necesario para comprender la relevancia del reto a realizar. Se ofrece una visión general que facilita la construcción de un esquema mental sobre los conceptos y tecnologías que se abordarán en el desarrollo del reporte.

El control de motores de corriente continua, es esencial en automatización y robótica, permitiendo regular su velocidad y dirección de giro con eficiencia, En este proyecto, se diseñó e implementó un sistema de control a través de Micro-ROS utilizando una ESP32 y un

puente H, aplicando modulación por ancho de pulso (PWM) para regular la potencia suministrada al motor.

Micro-ROS

Micro-ROS (Micro Robot Operating System) es una extensión de ROS 2 diseñada específicamente para funcionar con sistemas embebidos y microcontroladores. Su propósito es llevar las capacidades de ROS a dispositivos con capacidades de procesamiento y memoria reducidas, permitiendo la integración de sensores, actuadores y otros componentes dentro de sistemas robóticos más grandes y complejos. Al aprovechar las ventajas de ROS 2, Micro-ROS facilita la comunicación entre estos dispositivos y sistemas más potentes, por ejemplo computadoras que ejecutan ROS 2.

Uno de los aspectos clave de Micro-Ros, como se mencionó anteriormente, es su capacidad de operar en hardware de bajos recursos, como la ESP32, que suelen contar con apenas unos pocos kilobytes de RAM y almacenamiento. Para hacer esto posible, utiliza DDS-XRCE (Data Distribution Service for eXtremely Resource-Constrained Environments), un protocolo de comunicación que permite reducir el uso de memoria y ancho de banda, garantizando así una transmisión eficiente de datos entre dispositivos embebidos y el resto del sistema ROS 2.

Otra característica fundamental de Micro-ROS es su compatibilidad con sistemas operativos en tiempo real (RTOS). Estos sistemas permiten a Micro-ROS ejecutar tareas con restricciones de tiempo críticas. Además, su modularidad le permite adaptarse a distintas configuraciones y requisitos, asegurando así poder ser utilizado en una variedad de dispositivos.

Micro-ROS se usa en robots móviles, drones, sistemas autónomos y sensores distribuidos en entornos industriales que necesitan interactuar con ROS 2 en sistemas más grandes, como estaciones base o la nube.

Módulos ADC

Un módulo ADC (Analog-to-Digital Converter) es un componente fundamental en sistemas electrónicos ya que permite la conversión de señales analógicas en datos digitales tal y como su nombre lo indica. Dicho de otra forma, toma una señal continua y la convierte en un valor numérico que puede ser procesado por un sistema digital.

La conversión analógica digital ocurre en varias etapas. Primeramente, el muestreo captura valores de la señal analógica en intervalos regulares. Luego, viene la cuantización de esos valores capturados a niveles discretos dentro de un rango predefinido. Finalmente, en la etapa de codificación, cada valor cuantificado se traduce a un código binario para que pueda ser interpretado por dispositivos digitales.

Una de las principales características de un módulo ADC es su resolución, la cual determina la precisión con la que se representa la señal analógica y se mide en bits. Por ejemplo, un

ADC de 8 bits puede representar 256 niveles diferentes (2^8), mientras que uno de 12 bits puede representar 4096 niveles (2^{12}), permitiendo una mayor exactitud en la conversión.

Otro aspecto importante es la frecuencia de muestreo, que define cuántas veces por segundo el ADC toma una muestra de la señal analógica. Si la frecuencia de muestreo es muy baja, se podrían perder detalles de una señal, por otro lado, una frecuencia demasiado alta podría generar una carga innecesaria de datos a procesar.

El rango de voltaje define los valores mínimo y máximo que el ADC puede interpretar. Si el voltaje de entrada supera ese rango, se pueden generar distorsiones en la conversión, por lo cual es crucial tener en cuenta el rango de voltaje.

En la práctica, los módulos ADC tienen varias aplicaciones, desde la medición de temperatura en sensores industriales hasta la captura de audio y video en dispositivos electrónicos.

Puente H

El puente H es un circuito utilizado para controlar el sentido de giro de un motor de corriente continua. Su nombre proviene de la configuración de cuatro transistores que se organizan en la forma de la letra “H”, con el motor ubicado en el centro tal y como se muestra en la figura 1.

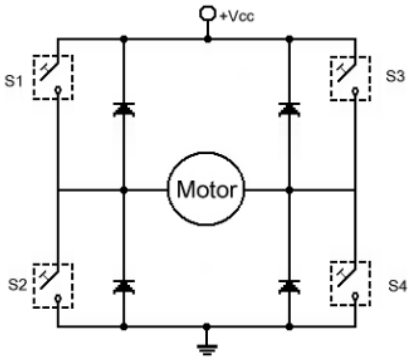

Figura 1. Puente H

Los cuatro interruptores del puente H (S1, S2, S3 y S4) permiten intervenir la polaridad de la tensión aplicada al motor, lo que cambia su dirección de giro tal como se muestra en la tabla y figura 2.

S1	S2	S3	S4	Resultado
1	0	0	1	El motor gira en avance
0	1	1	0	El motor gira en retroceso
0	0	0	0	El motor se detiene bajo su inercia

1	1	0	0	El motor se detiene
0	0	1	1	El motor se detiene
1	0	1	0	Cortocircuito
0	1	0	1	Cortocircuito
1	1	1	1	Cortocircuito

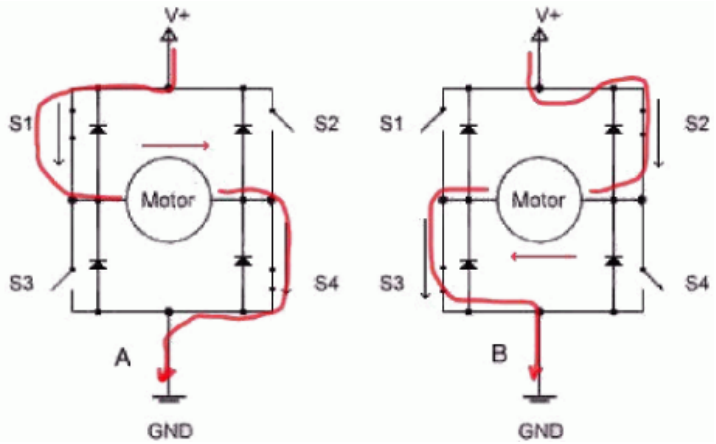

Figura 2. Funcionamiento puente H

El PWM se puede utilizar con el Puente H para controlar la velocidad del motor. En este caso, en lugar de encender o apagar completamente los transistores, se modula su tiempo de activación con PWM ajustando el voltaje promedio aplicado al motor.

Características de un Puente H L-298 (Figura 3)

- Voltaje de alimentación del motor entre 3V y 35V
- Los pines IEA, IN1 e IN2 controlan la salida A.
- Los pines IEB, IN3 e IN4 controlan la salida B.
- Los pines IN1, IN2, e IN3 e IN4, controlan la dirección de giro, respectivamente, de la salida A y B.

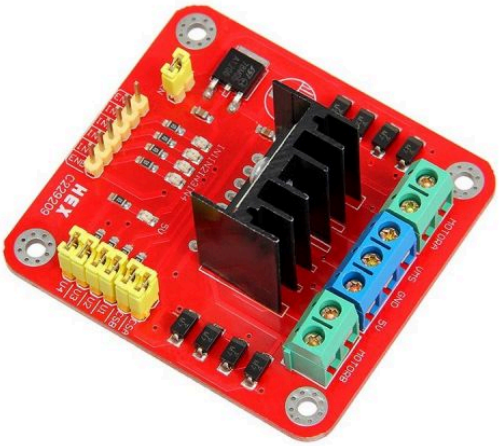

Figura 3. Puente H L-298

PWM

La modulación por ancho de pulso (PWM) es una técnica que permite controlar la potencia entregada a un carga variando el tiempo en que una señal digital permanece activa dentro de un ciclo. Esta variación se conoce como ciclo de trabajo (duty cycle).

La señal de PWM es una onda cuadrada con dos estados posibles: alto (voltaje máximo) y bajo (voltaje cero). Al modificar el tiempo que la señal permanece en estado alto en cada ciclo (duty cycle), se puede controlar el voltaje promedio aplicado a la carga. Podemos observar este comportamiento en la figura 4.

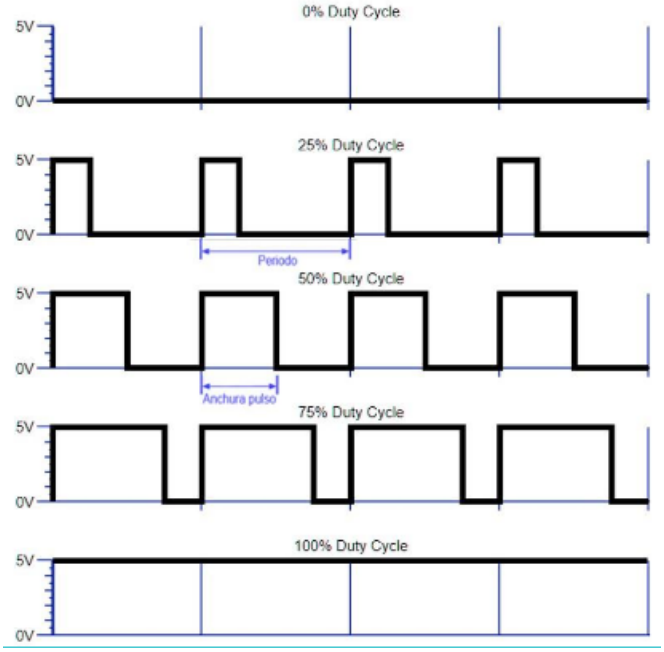

Figura 4. PWM

El voltaje promedio aplicado a la carga está dado por la ecuación:

$$V_{prom} = (V_{in} - 0V) \cdot \frac{Duty\ Cicle}{100}$$

Donde:

V_{in} es el voltaje de alimentación del sistema

V_{prom} es el voltaje promedio aplicado a la carga

ESP32

La ESP32 es un microcontrolador de alto rendimiento desarrollado por Espressif Systems, diseño para aplicaciones de IoT, automatización y control embebido.

Características

- Posee un procesador Dual-core
- Soporta Wi-Fi y Bluetooth
- Incluye un módulo ADC de 12 bits en múltiples canales
- Tiene dos salidas DAC (convertidor digital-analógico) de 8 bits
- Soporte para RTOS, micro-ros, micro-python y programación de C/C++
- Podemos observar el Pinout de la ESP32 en la figura 5.

Figura 5. pines ESP32

Funcionamiento del PWM en la ESP32

La ESP32 incorpora un módulo especializado llamado LEDC (LED Controller), el cual está diseñado para generar señales PWM de manera eficiente. Se puede utilizar para controlar una gran variedad de dispositivos, como motores DC, servomotores, entre otros actuadores. El PWM en una ESP32 se puede configurar al definir tres parámetros clave:

- Frecuencia de la señal, que determina la cantidad de ciclos por segundo que genera la señal.

- Resolución del PWM, que establece el número de niveles posibles para ajustar el duty cycle. En el caso de la ESP32 puede ser de hasta 12 bits.
- Duty Cycle, que define el tiempo durante el cual la señal permanece en estado alto dentro de un periodo completo.

En comparación con otros microcontroladores, la ESP32 tiene ciertas ventajas en términos de generación de PWM. Mientras que un Arduino Uno solo puede generar PWM en 6 pines con una resolución de 8 bits y una frecuencia máxima de aproximadamente 1 kHz, la ESP32 ofrece 16 canales de PWM, una resolución de hasta 16 bits y frecuencias que pueden alcanzar los 40 MHz. Esto permite un control mucho más preciso en aplicaciones de control de motores y audio, además de permitir manejar múltiples dispositivos de manera simultánea sin afectar el rendimiento general del microcontrolador.

Gracias a su capacidad de generar PWM de alta frecuencia con una resolución ajustable, la ESP32 se usa en múltiples aplicaciones prácticas. Uno de los usos más comunes, y el cual se implementa en este reto, es el control de motores DC. Para esto, el PWM se combina con un Puente H, que permite invertir la polaridad de la corriente, controlando así tanto la dirección como la velocidad del motor. Usando PWM, se puede ajustar la velocidad sin necesidad de cambiar el voltaje de alimentación, lo que mejora la eficiencia y prolonga la vida útil del sistema.

Solución del problema

A continuación, se describe la metodología empleada para alcanzar los objetivos del reto, detallando los elementos utilizados, las funciones implementadas y el desarrollo del código de programación.

Figura 6. Grafo del reto

Para lograr el control de velocidad de un motor de corriente continua (DC) mediante Micro-ROS y una ESP32 tal y como se muestra en la figura 6, se siguió una metodología estructurada que incluyó las siguientes etapas:

1. Configuración del Hardware
    a. Se utilizó ESP32 como microcontrolador, un puente H para invertir la dirección del motor y un generador de señales PWM para controlar su velocidad, La conexión se realizó siguiendo los diagramas proporcionados, asegurando la correcta asignación de pines (figura 7):
        i. GPIO 14 y 27: control de dirección del motor.
        ii. GPIO 26: generación de la señal PWM

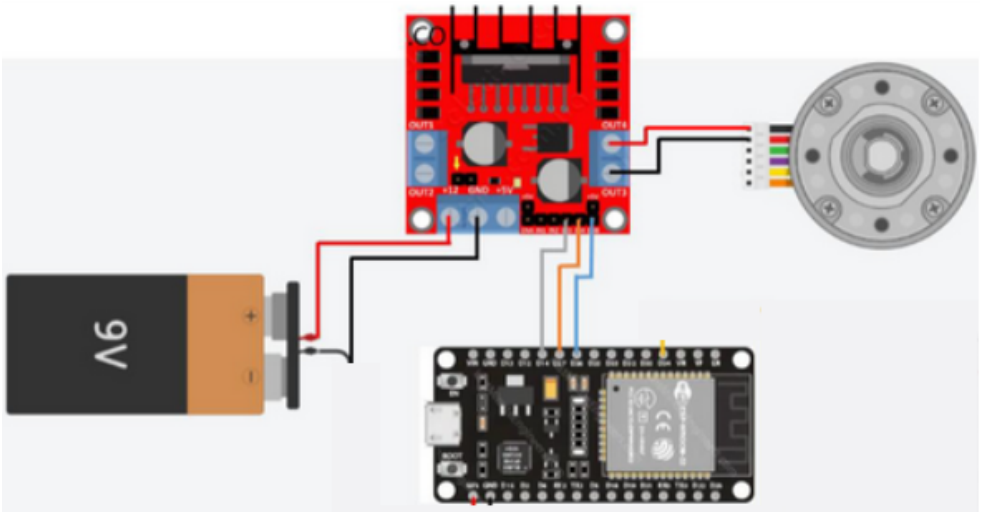

Figura 7. Diagrama de conexión

b. Implementación de Micro-Ros.
    i. Se configuró Micro-ROS en la ESP 32 para permitir la comunicación con una computadora externa que ejecuta ROS 2. Esto permite enviar comandos desde la computadora al microcontrolador para modificar la velocidad del motor dinámicamente.
c. Generación de la señal PWM.
    i. La modulación por ancho de pulso (PWM), se implementó en la ESP32 mediante el módulo LEDC (LED Controller) configurando:
        1. Frecuencia de la señal: 980 hz.
        2. Resolución del duty cycle: 8 bits (0-255).
        3. Canal de PWM y pin de salida.
d. Desarrollo de Nodos ROS 2.
    i. Se crearon nodos en ROS 2 para publicar los comandos de velocidad desde la computadora hacia la ESP 32. Estos nodos publicaban mensajes que eran interpretados por el firmware del microcontrolador para ajustar la señal PWM y controlar el motor en tiempo real.
    ii. El desarrollo del nodo recibe desde la terminal el mensaje del usuario para el valor de pwm, este nodo se suscribe a los mensajes que están llegando y compara si se encuentra en el parámetro establecido de -1 a 1, en base a su valor máximo y mínimo
    iii. poniendo esto a prueba establece a qué dirección se encontrará el movimiento de nuestro motor, siendo menor de 0 a la izquierda y mayor a cero a la derecha, como último se define que si el valor del pwm es cero el motor se tiene que detener
e. Pruebas y Validación.
    i. Se realizaron pruebas para verificar el correo funcionamiento del sistema evaluando:
        1. La capacidad del motor para responder a los cambios de velocidad.

2. La estabilidad de la señal PWM generada.
3. La correcta comunicación entre la ESP 32 y la computadora a través de Micro-ROS.

Las pruebas confirmaron que el sistema lograba regular la velocidad del motor de manera eficiente, aunque se identificaron oportunidades de mejora en la precisión del control.

Resultados

En esta sección se presentan las evidencias obtenidas durante la implementación del sistema, describiendo el significado de cada resultado y su impacto en el desempeño del sistema. Además, se analiza si los resultados alcanzados cumplen con los objetivos planteados en la actividad, permitiendo evaluar su efectividad y posibles áreas de mejora.

Tras la implementación del sistema de control de velocidad del motor DC mediante Micro ROS y la ESP32, se obtuvieron los siguientes resultados:

1. Control efectivo del motor:
    a. Se logró modular la velocidad del motor de manera precisa mediante la señal PWM.
    b. Se verificó que el puente H permitió cambiar la dirección de giro correctamente.
2. Comunicación estable con ROS 2:
    a. La ESP32 recibe comandos desde la computadora externa a través de Micro-ROS sin latencias significativas.
    b. Se confirmó la correcta interpretación de los mensajes para modificar la velocidad en tiempo real.
3. Precisión en la generación de PWM.
    a. Se comprobó que la ESP32 pudo generar señales PWM con la frecuencia y resolución esperadas (980 Hz y 8 bits de resolución).
    b. Se observó una respuesta estable del motor ante variaciones en el duty cycle.
4. Áreas de mejora identificadas:
    a. Detectamos que, en bajas velocidades, el motor presentaba pequeñas variaciones en el rendimiento, lo que sugiere la necesidad de un ajuste en la señal PWM o en el uso de un controlador PID.
    b. Recomendamos optimizar la gestión de recursos en la ESP32 para mejorar la eficiencia en la ejecución de Micro-Ros.

Video de evidencia en terminal

Video de implementación física

Conclusiones

Por último, se presentan los logros alcanzados a lo largo del desarrollo del reto, analizando el grado de cumplimiento de los objetivos planteados. Se evalúa si estos fueron alcanzados en su totalidad, identificando las razones detrás de su éxito o, en caso contrario, los factores que pudieron haber limitado su cumplimiento. Asimismo, se proponen posibles mejoras a la metodología utilizada, con el propósito de optimizar los resultados obtenidos. A través de este análisis, se busca profundizar en la comprensión de los procesos involucrados y extraer aprendizajes clave para futuras implementaciones.

El desarrollo de este reto permitió la implementación de un sistema de control de velocidad para un motor de corriente continua utilizando Micro-ROS, una ESP32 y un puente H, A través del uso de modulación por ancho de pulso (PWM), se logró regular la velocidad del motor de manera eficiente y precisa.

Se demostró que Micro-ROS es una herramienta viable para la integración de microcontroladores con ROS 2, facilitando la comunicación entre dispositivos embebidos y una computadora externa. La ESP 32 se desempeñó correctamente en la generación de señales PWM y la ejecución de nodos ROS 2, aunque se identificaron oportunidades de mejora en la estabilidad a bajas velocidades.

Los Objetivos del reto fueron alcanzados satisfactoriamente, permitiéndonos aplicar conocimientos sobre control de motores, comunicación de sistemas embebidos y ROS 2. Para futuras mejoras, se sugiere optimizar la señal PWM con técnicas de control más avanzadas, como la implementación de un controlador PID, así como explorar el uso de filtros para reducir variaciones en la velocidad del motor.

En conclusión, este reto proporcionó una experiencia práctica en la interconexión de sistemas embebidos con ROS, sentando las bases para proyectos más complejos en robótica y automatización.

Referencias

En este apartado se anexan los elementos consultados para el desarrollo del tema de investigación.

Cerino Jiménez, R. (2025, 28 febrero). *Sensores y actuadores_sesión_6_PWM.pdf* [Diapositivas].

www.alldatasheet.es. (s. f.). *L298N datasheet STMICROELECTRONICS*. DUAL FULL-BRIDGE DRIVER. https://www.alldatasheet.es/html-pdf/22440/STMICROELECTRONICS/L298N/3243/2/L298N.html

ESP32 Series Datasheet Version 4.8 https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf

ManchesterRoboticsLtd. (s. f.). *GitHub - ManchesterRoboticsLtd/TE3001B_Intelligent_Robotics_Implementation_2025: In this repository you will find all the resources required for the TE3001B*. GitHub. https://github.com/ManchesterRoboticsLtd/TE3001B_Intelligent_Robotics_Implementation_2025/tree/main

Micro-ROS. (s. f.). micro-ROS. https://micro.ros.org/

ROS Documentation. (s. f.). https://docs.ros.org/