

Diseño e implementación de un sistema de adquisición de señales para motores CD con ESP32 y Raspberry Pi

Design and implementation of a signal acquisition system for DC motors using ESP32 and Raspberry Pi

I. Tinoco-Ramírez¹, J. C. Sernaque-Julca¹, G. Quiroz-Compeán^{1,*}

Facultad de Ingeniería Mecánica y Eléctrica, Universidad Autónoma de Nuevo León, 66455, San Nicolás de los Garza, Nuevo León, México.

Resumen

Este trabajo presenta un sistema embebido modular para adquisición de datos y control de motores de corriente directa, basado en una arquitectura distribuida con tres módulos ESP32 y una Raspberry Pi. El sistema monitorea hasta cinco motores en tiempo real, midiendo variables eléctricas (voltaje y corriente) y variables mecánicas (posición angular); además de estimar velocidad y aceleración angulares. La comunicación se realiza mediante protocolo SPI en topología maestro-esclavo: dos ESP32 esclavos, uno intermedio con doble rol y la Raspberry Pi como maestro central encargado del almacenamiento de datos y generación de señales de control en lazo abierto. Las variables anteriores se adquieren a través de los puertos disponibles; además, tiene la opción de un ADC Delta-Sigma de 24 bits para medir par. Con un tiempo de muestreo de 2 ms, el sistema ofrece alta precisión y respuesta rápida. Fue implementado y evaluado en condiciones experimentales, demostrando confiabilidad, modularidad y potencial para aplicaciones de control en lazo cerrado.

Palabras Clave: Sistemas embebidos, control de motores CD, adquisición de señales, comunicación SPI.

Abstract

This work presents a modular embedded system for data acquisition and control of direct current motors, based on a distributed architecture using three ESP32 modules and a Raspberry Pi. The system monitors up to five motors in real time, measuring electrical variables (voltage and current) and mechanical variables (angular position); it also estimates angular velocity and acceleration. Communication is carried out via the SPI protocol in a master-slave topology: two slave ESP32s, one intermediate module with a dual role, and the Raspberry Pi as the central master responsible for data storage and open-loop control signal generation. The aforementioned variables are acquired through available ports; as an option, a 24-bit Delta-Sigma ADC is used for torque measurement. With a sampling time of 2 ms, the system provides high precision and fast response. It was implemented and evaluated under experimental conditions, demonstrating reliability, modularity, and potential for closed-loop control applications.

Keywords: Embedded systems, DC motor control, signal acquisition, SPI communication.

1. Introducción

El desarrollo de sistemas de adquisición de datos robustos y precisos constituye un pilar fundamental en aplicaciones de control de motores, robótica e identificación de sistemas dinámicos. En particular, los sistemas multi-motor requieren soluciones que garanticen sincronización temporal, alta frecuencia de muestreo y capacidad de procesamiento distribuido, especialmente cuando se trabaja con señales mecánicas y

eléctricas que exhiben dinámicas rápidas y no lineales. En el estudio realizado por Qiang *et al.* se muestra que, para sistemas de medición y control distribuidos (como sistemas multi-motor), es esencial implementar sincronización temporal de alta precisión (Qiang *et al.*, 2024). Sin embargo, muchos sistemas comerciales de adquisición son costosos, poco adaptables, lentos o requieren configuraciones complejas (Knapp y Bloom, 2022). En este marco, los microcontroladores modernos como

* Autor para correspondencia: griselda.quirozcm@uanl.edu.mx

Correo electrónico: isaac.tinocorz@uanl.edu.mx (Isaac Tinoco-Ramírez), juan.sernaquej@uanl.edu.mx (Juan Carlos Sernaque-Julca), griselda.quirozcm@uanl.edu.mx (Griselda Quiroz-Compeán)

Historial del manuscrito: recibido el 12/07/2025, última versión-revisada recibida el 03/11/2025, aceptado el 25/10/2025, publicado el 25/03/2026. **DOI:** <https://doi.org/10.29057/icbi.v14iEspecial.15495>



el ESP32 han ganado relevancia gracias a sus capacidades de procesamiento en tiempo real, conectividad, múltiples interfaces de comunicación (SPI, I2C, UART) y su bajo costo. A su vez, las plataformas embebidas como la Raspberry Pi ofrecen un entorno robusto para el procesamiento, visualización y almacenamiento de grandes volúmenes de datos, siendo una opción ideal para sistemas de adquisición distribuidos.

En este trabajo se presenta el diseño de un sistema de adquisición de señales para motores de corriente directa (CD), empleando microcontroladores ESP32 como nodos de captura de datos y una Raspberry Pi como unidad central de almacenamiento y generación de señales de control en lazo abierto. El sistema permite la medición de variables eléctricas (voltaje y corriente) mediante sensores INA219 y variables mecánicas (posición) a partir de encoders rotatorios, así como la estimación de velocidad y aceleración angulares. Además, el sistema asegura la transmisión eficiente de datos cada 2 ms mediante un protocolo maestro-esclavo basado en SPI. El desarrollo de este sistema busca ofrecer una herramienta flexible y modular, de bajo costo y precisa, adaptable a distintos entornos académicos o industriales para estudios de identificación de parámetros, control en tiempo real o análisis de desempeño de motores CD.

2. Marco teórico

Los sistemas de adquisición de datos (DAQ, por las siglas en Inglés de *Data Acquisition*) constituyen un componente fundamental en el control moderno de sistemas electromecánicos, particularmente en aplicaciones que requieren monitoreo y control en tiempo real de múltiples actuadores (Choi *et al.*, 2024). La adquisición precisa y sincronizada de señales eléctricas y mecánicas permite una caracterización dinámica detallada de sistemas como los motores CD.

La arquitectura propuesta en este trabajo sigue el paradigma de sistemas distribuidos, donde múltiples unidades de procesamiento (en este caso, microcontroladores ESP32) colaboran para adquirir y procesar datos de manera coordinada (Chang *et al.*, 2025). Esta aproximación permite escalar el sistema según el número de motores a controlar, manteniendo una alta frecuencia de muestreo gracias a la distribución de la carga computacional. Además, al descentralizar el procesamiento, se mejora la tolerancia a fallos y se reduce la latencia en la recolección de datos. El diseño e implementación del sistema propuesto parte de la medición y acondicionamiento de las señales eléctricas y mecánicas; garantizando la comunicación sincronizada entre los ESP32 y la Raspberry Pi. Además se considera la validación del sistema de adquisición de señales estimulando a los motores con señales apropiadas. Lo anterior, se detalla a continuación.

2.1. Medición y acondicionamiento de señales

La medición precisa de variables eléctricas en motores de CD requiere técnicas especializadas debido a la presencia de ruido conmutado, caídas de tensión en cables y variaciones rápidas en las señales, por eso, el uso de sensores integrados como el INA219 representa una solución eficiente y compacta, al combinar amplificación diferencial, filtrado anti-aliasing y conversión analógico-digital en un único módulo (Setiawan *et al.*, 2023). Estos sensores permiten medir corriente y voltaje

con alta precisión y bajo costo, lo cual es clave para aplicaciones de control en tiempo real. Para asegurar una correlación temporal precisa entre las mediciones eléctricas y mecánicas, se emplean mecanismos de sincronización basados en interrupciones periódicas, lo que minimiza el *jitter* de adquisición entre diferentes nodos.

En cuanto a la medición de variables mecánicas, los encoders incrementales permiten obtener información de posición angular con buena resolución. Sin embargo, la obtención de velocidad y aceleración a partir de estas señales requiere diferenciación numérica, lo que amplifica el ruido y limita la precisión si no se aplican técnicas de filtrado adecuadas (Petrella *et al.*, 2020).

2.2. Comunicación sincronizada

La elección de protocolos de comunicación debe considerar tanto los requisitos como las restricciones de tiempo real, considerando esto, la combinación de SPI para la comunicación entre microcontroladores e I²C para la lectura de sensores es una práctica común en sistemas embebidos heterogéneos, donde SPI ofrece mayor velocidad de transferencia y menor latencia, mientras que I²C facilita la conexión de múltiples dispositivos en un bus compartido con menor número de líneas (Zhuang, 2024).

La sincronización temporal precisa es un desafío clave en sistemas distribuidos, especialmente cuando los datos adquiridos son utilizados para tareas de identificación, control o monitoreo coordinado. El enfoque basado en interrupciones periódicas implementado en este sistema, con una frecuencia de 500 Hz, se inspira en los principios de sistemas de tiempo real, donde la predictibilidad y la baja variabilidad temporal son prioritarias frente al rendimiento bruto. También se han incorporado mecanismos de verificación de integridad de datos mediante check bytes y esquemas de retransmisión, siguiendo buenas prácticas para garantizar comunicaciones confiables en entornos industriales o de laboratorio con condiciones electromagnéticas adversas (Benjamin, 2021).

2.3. Señales para pruebas experimentales

Para validar la capacidad del sistema propuesto, se diseñaron pruebas experimentales que consisten en la adquisición sincronizada de señales eléctricas (voltaje y corriente) y mecánicas (posición, velocidad y aceleración) durante la operación de motores de CD bajo diferentes condiciones de carga y alimentación. Además, se utilizaron perfiles de entrada diseñados específicamente para excitar adecuadamente la dinámica de los motores. Estas señales permiten evaluar la respuesta del sistema y verificar que el sistema de adquisición mantiene la integridad de las mediciones durante transitorios rápidos.

3. Metodología

El desarrollo del sistema de adquisición de señales se llevó a cabo en tres etapas principales: diseño e implementación del hardware, desarrollo del firmware y validación experimental del sistema. Dichas etapas se detallan a continuación.

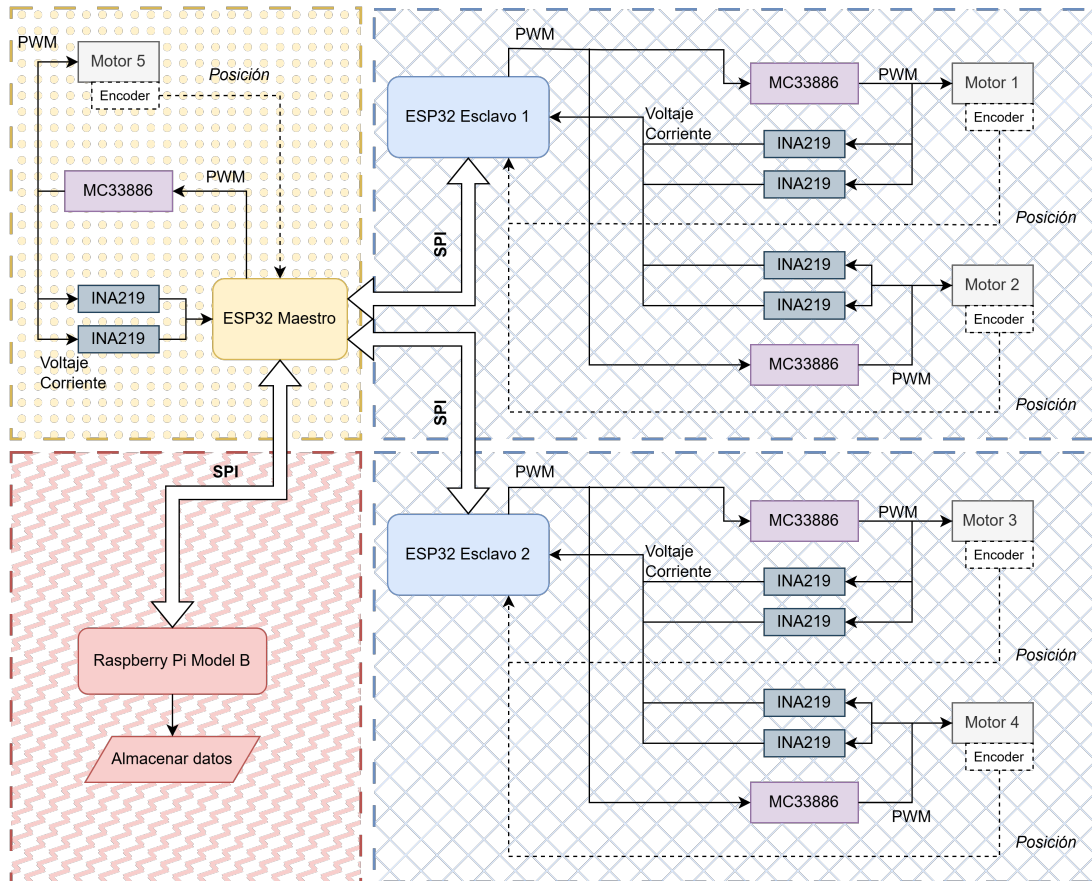


Figura 1: Diagrama de funcionamiento de la tarjeta de adquisición diseñada para la medición de señales eléctricas y mecánicas de motores de CD.

3.1. Diseño e implementación del hardware

El diseño propuesto se desarrolló para una maqueta experimental que incorpora cinco motorreductores: cuatro de 10 RPM y uno de 5 RPM. Cada motor está equipado con un encoder incremental de cuadratura con una resolución de 21 pasos por vuelta y opera a un voltaje nominal de 12 V. Para el driver, se emplea un módulo basado en el chip MC33886, el cual permite trabajar en un rango de voltaje de 5 V a 28 V, con capacidad de corriente de hasta 5 A y permite el control unidireccional de dos motores o bidireccional de uno solo. En este diseño, se utiliza un módulo por motor, operando en modo bidireccional. Además, el módulo incorpora protecciones contra sobrecorriente, sobretensión y cortocircuito a tierra.

Para la medición de señales eléctricas, se utiliza el módulo INA219, que integra un amplificador diferencial de alta precisión y un convertidor analógico-digital (ADC, por las siglas en Inglés de *Analog-to-Digital Converter*) de 12 bits. Este módulo permite medir simultáneamente corriente, voltaje y potencia (en buses de hasta 26 V y 3.2 A), y se comunica mediante el protocolo I²C. Dado que se requiere medir ambas direcciones de giro de cada motor, se emplean dos módulos INA219 por motor.

El sistema cuenta con tres microcontroladores ESP32, dos de ellos operan como esclavos, encargándose del procesamiento de datos provenientes de los sensores INA219 (cuatro por cada ESP32) mediante I²C. Cada ESP32 controla y monitorea dos motores, incluyendo la lectura de encoders y botones de límite (dos por motor) que indican los extremos del rango de

movimiento permitido. El tercer ESP32 actúa en modo dual esclavo/maestro. Como maestro, recopila los datos de los otros dos ESP32 y adquiere localmente los datos correspondientes al quinto motor. Como esclavo, se comunica con una Raspberry Pi (que opera como maestro SPI), a la cual transmite todos los datos adquiridos de los cinco motores y desde la cual recibe las señales de control en lazo abierto. La Figura 1 muestra un diagrama general de las conexiones entre los diferentes módulos de la tarjeta.

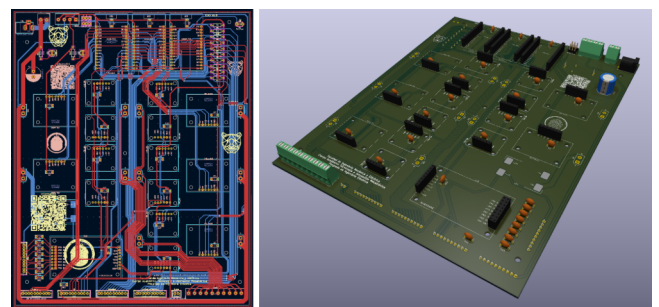


Figura 2: Izquierda: PCB de la tarjeta de adquisición diseñada en KiCAD. Debido a la cantidad de componentes, la tarjeta fue diseñada de doble capa, en donde las líneas rojas representan a la capa frontal y las azules a la capa trasera. Derecha: Renderizado de la PCB en KiCAD.

La tarjeta de adquisición fue diseñada en KiCAD v8.0, una suite de software libre para el diseño de circuitos electrónicos (Figura 2). Debido a la alta densidad de conexiones, se optó por una tarjeta de doble capa. El grueso de las pistas cumple con los estándares establecidos para el rango de corriente que

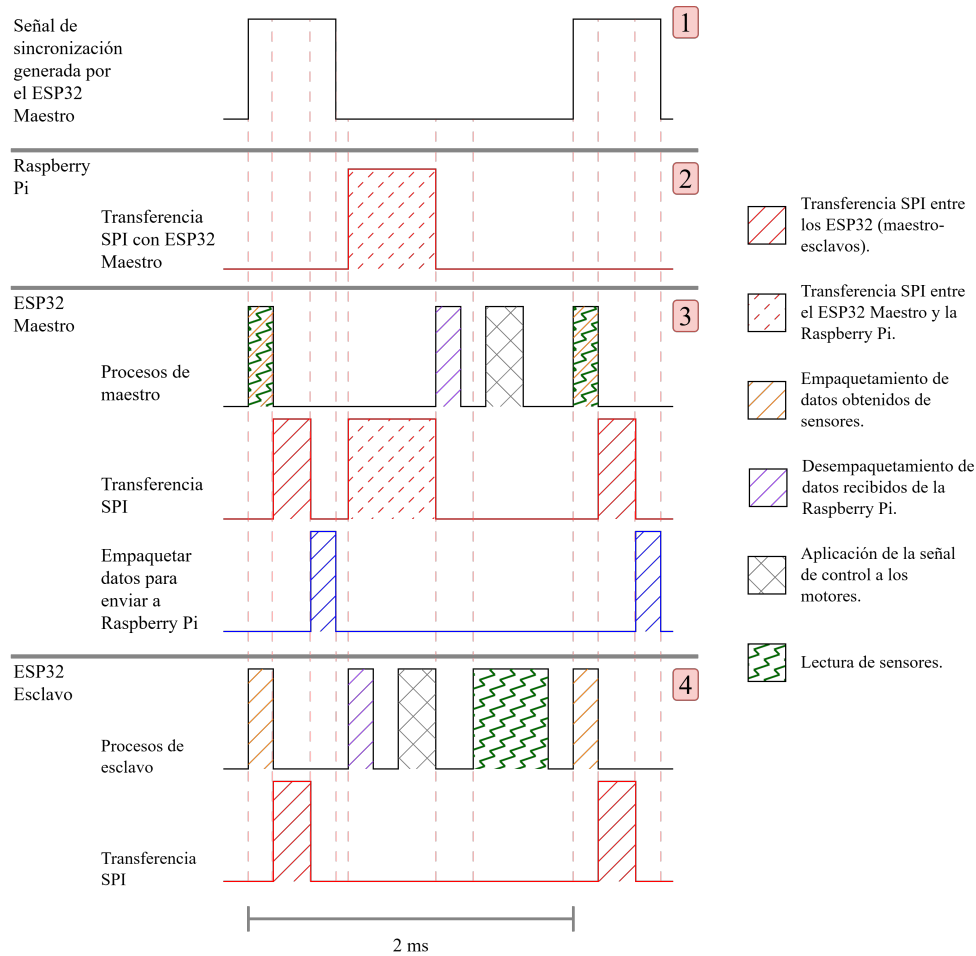


Figura 3: Proceso de comunicación sincronizado detallado por fases del sistema de adquisición. (1) Señal de sincronización generada por el ESP32. (2) Proceso de transferencia llevado a cabo en la Raspberry Pi. (3) Procesos realizados en el ESP32 Maestro para la comunicación. (4) Procesos realizados en los ESP32 Esclavos para la comunicación.

requieren los motores. El diseño se planteó de forma modular, permitiendo la sustitución rápida de componentes sin necesidad de desoldarlos. Para ello, se incorporaron conectores rápidos para la Raspberry Pi, encoders, botones de límite y sensores adicionales como acelerómetros y giroscopios (por I²C), así como un conector para sensores analógicos, conectado a un espacio reservado para un ADC Delta-Sigma. El renderizado 3D de la tarjeta puede observarse en la Figura 2 y la versión final ensamblada en la Figura 4.

3.2. Desarrollo del firmware de lectura y comunicación

Como se mencionó en la sección anterior, la comunicación entre los microcontroladores se lleva a cabo mediante el protocolo SPI, mientras que la comunicación con algunos sensores (como los módulos INA219 y, en caso de utilizarse, acelerómetros y giroscopios) se realiza a través del protocolo I²C. El proceso de sincronización del sistema de adquisición puede visualizarse en la Figura 3, donde se detalla el flujo de tareas implementado en el firmware.

La comunicación SPI entre los ESP32 esclavos y el ESP32 esclavo/maestro se realiza en paquetes de 64 bytes, esto debido a la restricción de la biblioteca SPI utilizada, el ESP32 esclavo/maestro envía a la Raspberry Pi dos paquetes consecutivos de 64 bytes cada uno, conteniendo la información adquirida.

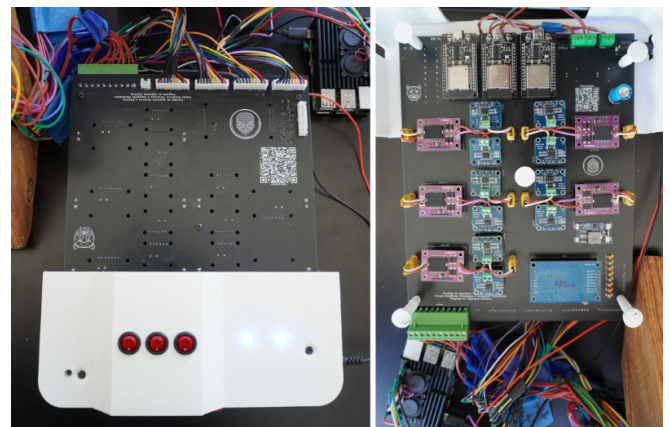


Figura 4: Tarjeta de adquisición de datos fabricada. Izquierda: Vista superior de la tarjeta. Derecha: Vista inferior en donde se pueden apreciar los módulos.

Cada par de paquetes incluye un identificador que permite relacionarlos con el mismo instante de medición. La Raspberry Pi verifica que ambos paquetes pertenezcan al mismo ciclo; si no coinciden, se realiza hasta dos intentos adicionales de verificación. Además, todos los paquetes intercambiados entre los microcontroladores contienen bytes de verificación (*check bytes*) que permiten validar la integridad de los datos recibidos.

La señal de sincronización es generada por el ESP32 que

opera en modo esclavo/maestro (Proceso 1, Figura 3). Esta señal marca el inicio de cada ciclo de adquisición y sincroniza las acciones entre los tres ESP32 y la Raspberry Pi. La generación de esta señal se realiza por medio de interrupciones cada 2 ms (esto es, el sistema opera a una frecuencia de adquisición de 500 Hz), aunque su ciclo de trabajo varía según el tiempo necesario para completar la lectura de los datos provenientes de los esclavos, la adquisición de los datos locales y el empaquetado de los mismos para su posterior transmisión a la Raspberry Pi (Procesos 2 y 3, Figura 3).

Cuando se detecta el flanco de bajada de la señal de sincronización, la Raspberry Pi espera un breve lapso antes de iniciar la comunicación con el ESP32 esclavo/maestro (Proceso 2, Figura 3). Durante esta fase se realiza una doble operación: por un lado, la Raspberry Pi recibe los datos correspondientes al ciclo anterior (es decir, la lectura anterior de los cinco motores), y por otro, transmite los valores de control (PWM) hacia el ESP32 esclavo/maestro. Simultáneamente, los dos ESP32 esclavos desempaquetan los valores de control recibidos previamente y los aplican a sus respectivos motores por medio de los drivers (Proceso 4, Figura 3). Al finalizar la comunicación entre la Raspberry Pi y el ESP32 esclavo/maestro, este último también aplica su señal de control correspondiente. En paralelo, todos los ESP32 (esclavos y maestro) realizan una nueva lectura de sus sensores, la cual quedará almacenada para ser enviada en la próxima ventana de sincronización (Proceso 3 y 4, Figura 3). Este ciclo se repite cada 2 ms de forma continua, aunque la comunicación con la Raspberry Pi se ejecuta únicamente cuando es requerida la adquisición de datos. Los datos recibidos por la Raspberry Pi son almacenados en archivos .csv para su posterior análisis.

3.3. Validación experimental

Para validar el funcionamiento del sistema se realizaron mediciones de los sensores de voltaje, corriente, pulsos de encoders (que derivan en posición), además de velocidad y aceleración al pasarlos por un diferenciador y valor de los botones de límite. Se emplearon señales de excitación persistente, caracterizadas por su riqueza espectral y variabilidad temporal, especialmente útiles en la estimación paramétrica (Cortés, 2012). Estas consisten en funciones multisinusoidales periódicas compuestas con fase aleatoria y frecuencias de valores irracionales, tal como se describe en la siguiente ecuación:

$$v(t_k) = \sum_{j=1}^m a_j \sin(\omega_j t + \varphi_j) \quad (1)$$

donde $m \in \mathbb{N}$ indica el número de componentes sinusoidales, a_j son las amplitudes de las componentes sinusoidales (para $j = 1, \dots, m$), ω_j es la frecuencia de valor irracional, tal que: $0 < \omega_1 < \omega_2 < \dots < \omega_m < \pi$; la fase φ_j debe ser una señal de tipo aleatoria. La ecuación (2) es un ejemplo de función multisinusoidal periódica compuesta, diseñada para cumplir la condición de excitación persistente para los motores utilizados: como señal de control en lazo abierto con la característica de ser de excitación persistente.

$$v(t) = A_1 \sin(\omega_1 t) + A_2 \sin(\omega_2 t) + A_3 \sin(\omega_3 t) + A_4 \sin(\omega_4 t) + A_5 \sin(\omega_5 t), \quad (2)$$

en donde $A_1 = 5$, $A_2 = 3$, $A_3 = 1$, $A_4 = 0.75$, $A_5 = 0.25$, $\omega_1 = 2\pi \times 0.4$, $\omega_2 = 2\pi \times 1$, $\omega_3 = 2\pi \times 5$, $\omega_4 = 2\pi \times 7$, $\omega_5 = 2\pi \times 10$.

La señal de excitación persistente generada a partir de la ecuación (2), fue utilizada como señal de control en lazo abierto durante los experimentos y se puede ver en la Figura 5.

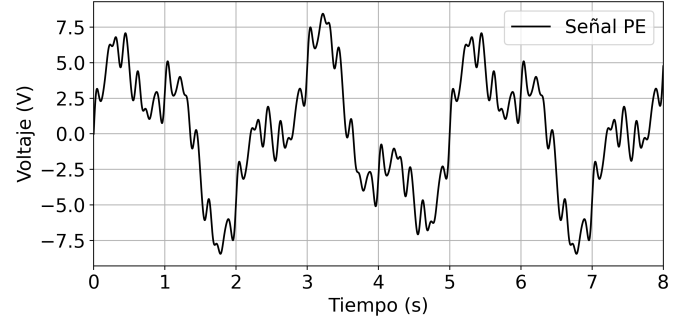


Figura 5: Señal de excitación persistente diseñada para los experimentos.

4. Resultados

La evaluación del desempeño de la tarjeta de adquisición se realizó utilizando la maqueta experimental de un mecanismo de cadena abierta con un motor en cada articulación y ejecutando dos experimentos. En el primero, se tomaron 10 mediciones individuales por cada uno de los cinco motores, resultando en un total de 50 conjuntos de datos. En el segundo experimento, los cinco motores se operaron de forma simultánea, adquiriéndose también 10 mediciones del sistema en conjunto. En ambos casos, se empleó una señal de excitación persistente como entrada PWM en lazo abierto para cada motor.

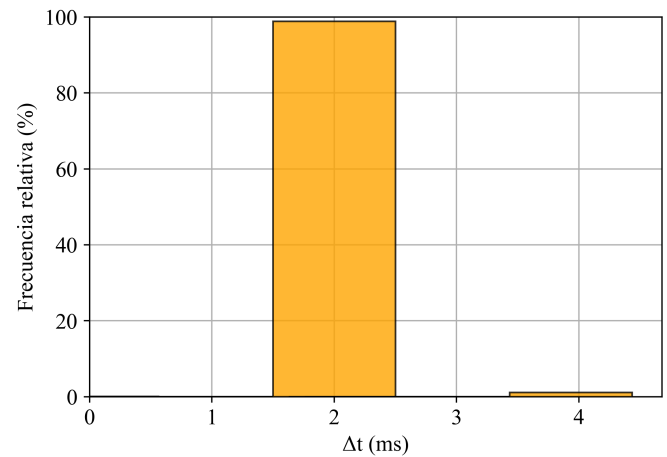


Figura 6: Histograma de frecuencia relativa del motor 1.

Como métricas de desempeño de la comunicación del sistema se calcularon: la frecuencia media de muestreo, el *jitter*, la diferencia promedio entre muestras consecutivas (Δt), así como los valores mínimo y máximo de Δt para cada conjunto. También se evaluó el porcentaje de muestras cuya desviación superó el 2 % respecto a la frecuencia media. En la Tabla 1 se presentan los resultados promediados de las diez mediciones individuales por motor. Para ilustrar el desempeño, se generó un histograma de la frecuencia relativa de Δt correspondiente al motor 1,

Tabla 1: Métricas de desempeño de la comunicación del sistema de adquisición propuesto.

Motor	Frec. media (Hz)	Frec. mínima (Hz)	Frec. máxima (Hz)	Δt medio (ms)	Δt mín. (ms)	Δt máx. (ms)	Jitter medio (ms)	Jitter máx. (ms)	% > 2 %	%máx > 2 %
1	494.757	489.234	497.883	2.021	0.000	4.000	0.209	0.300	1.173	2.301
2	494.875	488.159	500.125	2.021	0.000	4.000	0.199	0.314	1.140	2.526
3	495.246	489.594	498.753	2.019	0.000	4.000	0.201	0.295	1.088	2.226
4	495.911	489.953	499.126	2.017	0.000	4.000	0.184	0.290	0.938	2.151
5	498.041	493.095	500.000	2.008	0.000	4.000	0.134	0.243	0.520	1.500

como se puede ver en la Figura 6. También en la Figura 7 se encuentran las cinco variables principales de interés de los experimentos para el caso del motor 5, en la gráfica del voltaje se incluye la misma señal de excitación persistente de la Figura 5 de manera comparativa.

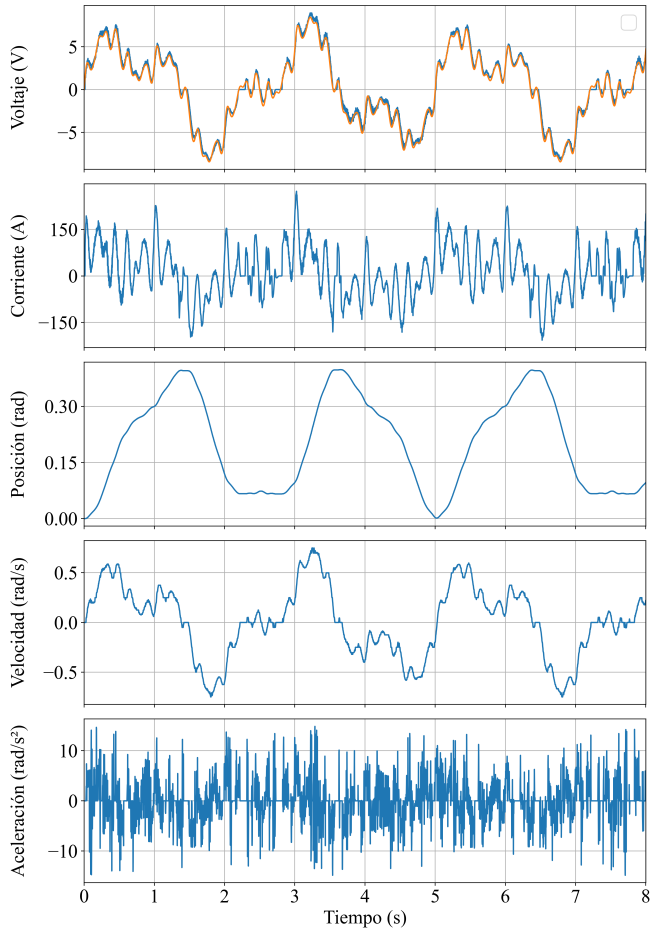


Figura 7: Principales señales obtenidas de un experimento en el motor 5. De arriba a abajo: voltaje, corriente, posición angular, velocidad angular y aceleración angular. En la gráfica de voltaje, la línea azul representa el voltaje promedio, mientras que la línea naranja es la señal de excitación persistente utilizada como entrada.

Además, se evaluó la repetibilidad entre experimentos, para ello se analizaron las señales de voltaje, corriente, posición, velocidad y aceleración, considerando múltiples repeticiones para cada condición experimental. A partir de estas señales, se calcularon estadísticas de las métricas de desempeño que permiten evaluar la calidad, estabilidad y similitud de las mediciones. Las métricas consideradas para esta sección son:

- **Error estándar de la media (SEM).** Se calculó el valor máximo del SEM para cada tipo de señal, lo cual proporciona una medida de la precisión del promedio estimado a partir de múltiples repeticiones.

- **Desviación estándar (STD).** Se reporta el valor promedio de la desviación estándar a lo largo del tiempo para cada señal, lo cual refleja la dispersión general de los datos.
- **Error cuadrático medio (RMSE).** Se calcula el RMSE entre cada repetición y la señal promedio correspondiente, promediando luego este valor sobre todas las repeticiones. Esto proporciona un indicador directo de la similitud temporal entre las repeticiones de cada señal.

Las Tablas 2, 3 y 4 presentan los resultados obtenidos para cada métrica y cada motor.

Tabla 2: Error estándar de la media (SEM) por señal y número de motor.

Motor	Voltaje	Corriente	Posición	Velocidad	Aceleración
1	0.316	21.038	0.664	0.063	4.389
2	0.451	32.212	0.039	0.057	4.510
3	0.263	31.866	0.020	0.043	4.470
4	0.310	12.972	0.050	0.038	4.617
5	0.247	14.047	0.032	0.620	4.449

Tabla 3: Desviación estándar (STD) media por señal y número de motor.

Motor	Voltaje	Corriente	Posición	Velocidad	Aceleración
1	0.222	12.073	0.011	0.011	4.520
2	0.275	13.733	0.002	0.012	5.095
3	0.252	13.929	0.003	0.011	4.659
4	0.271	12.988	0.001	0.011	5.246
5	0.261	13.769	0.002	0.013	5.805

Tabla 4: Error cuadrático medio (RMSE) por señal y número de motor.

Motor	Voltaje	Corriente	Posición	Velocidad	Aceleración
1	0.258	14.486	0.074	0.015	5.574
2	0.324	16.456	0.004	0.018	5.876
3	0.298	17.695	0.004	0.016	5.586
4	0.315	14.764	0.004	0.013	5.965
5	0.299	15.077	0.002	0.026	6.445

5. Discusión

Los resultados obtenidos muestran que el sistema de adquisición propuesto es capaz de capturar señales de voltaje, corriente y posición, además de estimar velocidad y aceleración con una frecuencia de muestreo cercana a los 500 Hz por motor, incluso cuando los cinco motores están activos simultáneamente. En comparación con otros sistemas de adquisición, el desempeño temporal y la estabilidad alcanzados resultan competitivos. Por ejemplo, se desarrolló un sistema basado en Arduino para dinámica vehicular con frecuencias menores a 80 Hz, aunque puede trabajar a mayores frecuencias al leer menos datos (González *et al.*, 2018). De forma similar, se reportó un sistema multisensorial a 140 Hz para detección de fallas en impresión 3D; aunque su arquitectura permite la adquisición de

menos variables (3 a 4 aproximadamente) (Kumar *et al.*, 2022). Esto demuestra que el diseño propuesto permite un desempeño adecuado para aplicaciones que requieren adquisición de datos en tiempo real con múltiples motores (Pană *et al.*, 2022).

El análisis de la frecuencia de muestreo reveló una variabilidad aceptable, con un *jitter* promedio inferior a 0.21 ms y un porcentaje de muestras fuera del 2 % menor al 1.2 % en promedio. El *jitter* promedio observado en este trabajo está dentro de rangos que han sido considerados aceptables en aplicaciones experimentales y de adquisición de datos de características similares. Por ejemplo, estudios en sistemas de adquisición comerciales han estimado precisiones operativas en el rango 0.5 a 3 ms en experimentos biológicos, lo que sitúa nuestro valor por debajo de esas precisiones reportadas (Ortega *et al.*, 2023). Estos valores son apropiados para diferentes aplicaciones de comunicación e indican una estabilidad temporal adecuada, fundamental para garantizar la consistencia en el análisis dinámico de los motores. Cabe destacar que el motor 5 presentó el mejor desempeño en términos de estabilidad temporal, lo que podría estar asociado a una menor carga del mecanismo debido a la posición en donde se encuentra acoplado o menor interferencia en su canal de comunicación SPI.

Respecto a la repetibilidad entre experimentos, las métricas de SEM, STD y RMSE muestran niveles aceptables de variabilidad. Las señales de posición y velocidad presentaron menor dispersión que las de corriente y aceleración, lo cual era esperado debido a la mayor sensibilidad de estas últimas a ruidos de medición y diferencias en la dinámica del motor. El bajo valor del SEM en todas las señales sugiere que el promedio obtenido es representativo, y que el sistema tiene una buena estabilidad entre ejecuciones bajo condiciones similares. En comparación con otros trabajos, los niveles de error observados se encuentran dentro de rangos considerados aceptables. En un trabajo se reporta para una etapa lineal de alta resolución errores dinámicos de posición con STD de 31.7 nm a 20 mm/s, y de 79.7 nm a 1 mm/s (Piščalov *et al.*, 2020). En sistemas de seguimiento electrónico, velocidades estimadas frente al estándar muestran errores RMS de 0.21 m/s; para aceleración los RMSE pueden alcanzar 0.45 m/s² dependiendo del sistema (Schmid y Lames, 2023). Estos valores, aunque en distintas escalas y condiciones, sirven para mostrar que el uso de SEM, STD y RMSE como métricas de repetibilidad y variabilidad es común, y son coherentes con lo que se considera aceptable en la literatura.

6. Conclusiones

Este sistema constituye una plataforma viable para futuras investigaciones relacionadas con la estimación paramétrica, la identificación de modelos dinámicos y el control de motores CD. Gracias a su capacidad de adquisición simultánea, sincronizada y con bajo retardo, puede utilizarse en experimentos de

lazo abierto que requieran precisión temporal para evaluar la respuesta dinámica de múltiples actuadores.

Como trabajo futuro, se puede implementar esquemas de control en lazo cerrado desde la Raspberry Pi, permitiendo así evaluar el desempeño del sistema bajo condiciones de operación reales. Adicionalmente, se puede completar la integración de sensores inerciales como acelerómetros y giroscopios, lo cual permitiría extender el análisis dinámico a sistemas robóticos o mecanismos multieje, donde las variables de orientación y aceleración lineal son relevantes. Esto abriría la posibilidad de desarrollar algoritmos avanzados de control y observación en plataformas embebidas, manteniendo la arquitectura modular, económica y flexible del sistema presentado.

Referencias

- Benjamin, B. (2021). Communication methods for data integrity using delta-sigma data converters. Application Report SBAA106A, Texas Instruments.
- Chang, Y., Wu, F., y Lin, H. (2025). Design and implementation of esp32-based edge computing for object detection. *Sensors*, 25(6):1656.
- Choi, S., Yang, H., Noh, Y., Kim, G., Kwon, E., y Yoo, H. (2024). Fpga-based multi-channel real-time data acquisition system. *Electronics*, 13(15):2950.
- Cortés, F. R. (2012). *MATLAB aplicado a la robótica y mecatrónica*. Alpha Editorial.
- González, A., Olazagoitia, J. L., y Vinolas, J. (2018). A low-cost data acquisition system for automobile dynamics applications. *Sensors*, 18(2):366.
- Knapp, A. y Bloom, A. J. (2022). Easy as piades: A low-cost, ultra-high-resolution data acquisition system using a raspberry pi. *Applications in plant sciences*, 10(3):e11485.
- Kumar, S., Kolekar, T., Patil, S., Bongale, A., Kotecha, K., Zaguia, A., y Prakash, C. (2022). A low-cost multi-sensor data acquisition system for fault detection in fused deposition modelling. *Sensors*, 22(2):517.
- Ortega, J., Niebur, T., Wood, L., Conn, R., y Sponberg, S. (2023). An information theoretic method to resolve millisecond-scale spike timing precision in a comprehensive motor program. *PLOS Computational Biology*, 19(6):1–24.
- Pană, C. F., Manta, L. F., Vladu, I. C., Cismaru, I., Petcu (Besnea), F. L., Cojocaru, D., y Bizdoacă, N. (2022). The design of a smart lower-limb prosthesis supporting people with transtibial amputation—a data acquisition system. *Applied Sciences*, 12(13):6722.
- Petrella, R., Tursini, M., Peretti, L., y Zigliotto, M. (2020). Speed measurement algorithms for low-resolution incremental encoder equipped drives: a comparative analysis. *IEEE Transactions on Industrial Electronics*, 67(8):6586–6595.
- Piščalov, A., Urbonas, E., Višniakov, N., Zabulionis, D., y Kilikevičius, A. (2020). Investigation of position and velocity stability of the nanometer resolution linear motor stage with air bearings by shaping of controller transfer function. *Symmetry*, 12(12):2062.
- Qiang, L., Jing, G., Wenyi, L., Wanjia, G., Yingzi, Z., y Yanjun, H. (2024). An enhanced time synchronization method for a network based on kalman filtering. *Scientific Reports*, 14(1):21271.
- Schmid, M. y Lames, M. (2023). Correction of systematic errors in electronic performance and tracking systems. *Sports Eng.*, 26(30).
- Setiawan, N. R., Ma'arif, A., y Widodo, N. (2023). Dc motor controller using full state feedback. *Control Systems and Optimization Letters*, 1(1):7–11.
- Zhuang, C. (2024). Comparison and selection of commonly used communication protocols in measurement and control instruments. *Highlights in Science, Engineering and Technology*, 81(1):540–546.