

Documento de Diseño

“Phoenix-C1” — Control de motor AC

Proyecto ELO301

Fecha	Revisión	Autor	Comentarios
03/10/2025	1.0	Rodrigo Sierra, Sebastián Pinochet, Gabriel Arcaya	Primera versión del documento de diseño.

Índice general

1. Introducción	3
2. Funcionalidades identificadas	3
3. Elementos identificados	3
4. Diseño de hardware propuesto	4
4.1. Diagrama de bloques	4
4.2. Detalle por bloque	5
5. Diseño de firmware propuesto	6
5.1. Detalle de drivers	6
5.2. Diagrama de flujo	6
5.3. Detalle por módulo	7
6. Pruebas	8
6.1. Prueba sobre elementos	8
6.2. Prueba sobre funcionalidades	8
7. Cronograma	9
8. Costos	9

Índice de imágenes

1. Diagrama de bloques	5
2. Diagrama de Flujo para el firmware	7

1. Introducción

El proyecto aborda la mejora del sistema de control de motores en un robot de competencia de la categoría RoboCup SSL. La arquitectura vigente integra en una única placa maestra la CPU, los giroscopios, los *drivers* de potencia, el módulo de radio y la gestión de energía, configuración que ha demostrado ser frágil: la falla o el desplazamiento de un componente compromete la operatividad completa de la placa. Para mitigar este riesgo, se propone una estrategia de modularización en la cual cada motor dispone de una placa de control independiente. Esta aproximación eleva la confiabilidad, facilita el mantenimiento en competencia y habilita la adopción progresiva de esquemas de control avanzados.

En una primera etapa, se priorizará el uso de un *encoder* magnético para implementar un lazo de control de velocidad simple y confiable. En paralelo, se dejarán previstas las conexiones físicas (sensado de corrientes y voltajes de fase, referencias de posición y sincronía, e interfaces de comunicación pertinentes) con el fin de habilitar, mediante una futura actualización de *firmware*, la implementación de control orientado por campo (FOC) sobre motores síncronos de imanes permanentes (PMSM), mejorando el desempeño dinámico y la eficiencia del actuador.

2. Funcionalidades identificadas

A partir del análisis del marco de referencia y de las correcciones al documento de especificaciones, se identifican las siguientes funcionalidades requeridas:

- Sensado de velocidad y/o posición mediante *encoder* (preferentemente magnético).
- Conmutación trifásica mediante *driver* de compuerta para MOSFET.
- Control básico para el accionamiento de un PMSM con posibilidad de evolución a FOC.
- Interfaces de comunicación SPI para enlace con una unidad central.
- Capacidades de diagnóstico: monitoreo de corrientes de fase y tensión de alimentación.

3. Elementos identificados

Para materializar las funcionalidades anteriores, se identifican los siguientes elementos de diseño y hardware:

- **Controlador principal:** microcontrolador de la familia STM32 para lazo de control y comunicaciones.
- **Etapas de potencia:** puente trifásico con MOSFET y *driver* de compuerta dedicado.
- **Sensado:** resistencias *shunt* para corriente y *encoder* magnético para posición/velocidad.
- **Alimentación y protección:** convertidores DC–DC para rieles lógicos y salvaguardas frente a sobre-corriente/sobretensión.
- **Comunicación:** conectividad externa mediante SPI hacia la CPU principal.
- **Firmware:** algoritmo de control para PMSM con ruta de actualización hacia FOC y parametrización en campo.

4. Diseño de hardware propuesto

4.1. Diagrama de bloques

El sistema se organiza en dos dominios: *control y sensado* (lógica a +3,3 V) y *potencia* (bus +V puede ser entre 10V a 40V). El microcontrolador **STM32G431CBU6** coordina la conmutación del inversor trifásico y la adquisición de señales. La etapa de potencia la componen el **gate driver DRV8323S** y seis **MOSFET NTMFS5C612NLT1G** en tres medio-puentes que alimentan un **PMSM**. La corriente de fase se mide con un **resistor shunt** de baja impedancia en el retorno. Un **AS5600** entrega la posición del rotor por bus I²C.

Buses y direcciones de señal:

- I²C bidireccional (**ENC_SDA**, **ENC_SCL**): el MCU actúa como maestro y el codificador AS5600 como esclavo; el flujo útil de datos es sensor→MCU.
- SPI con roles diferenciados:
 - **DRV8323S** (MCU *maestro*): **SPI_MOSI** (MCU→DRV), **SPI_MISO** (DRV→MCU), **SPI_SCK** (MCU→DRV), **GD_CS** (MCU→DRV, activo-bajo). Uso: configuración, telemetría y diagnóstico de fallas.
 - **Conector de referencia de velocidad** (MCU *esclavo*): **REF_MOSI** (Maestro externo→MCU), **REF_MISO** (MCU→Maestro externo), **REF_SCK** (Maestro externo→MCU), **REF_CS** (Maestro externo→MCU, activo-bajo). Uso: recepción de consignas y, opcionalmente, envío de telemetría bajo petición.
- PWM unidireccional (**MOTOR1**, **MOTOR2**, **MOTOR3**) desde el MCU hacia el DRV8323S para la modulación de compuertas (p.ej., SVPWM).
- GPIO/EXTI: líneas **INT1** y **INT2** reservadas para interrupciones de eventos externos; salida **LED** para señalización de estados.
- **Dominio de potencia**: el *Power Supply Connector* entrega el bus +V a la etapa de potencia y a la entrada del *Power Management (Buck)*, que regula a +3,3 V para lógica, sensores y señales de control, separando los dominios de *potencia* y *lógica*.

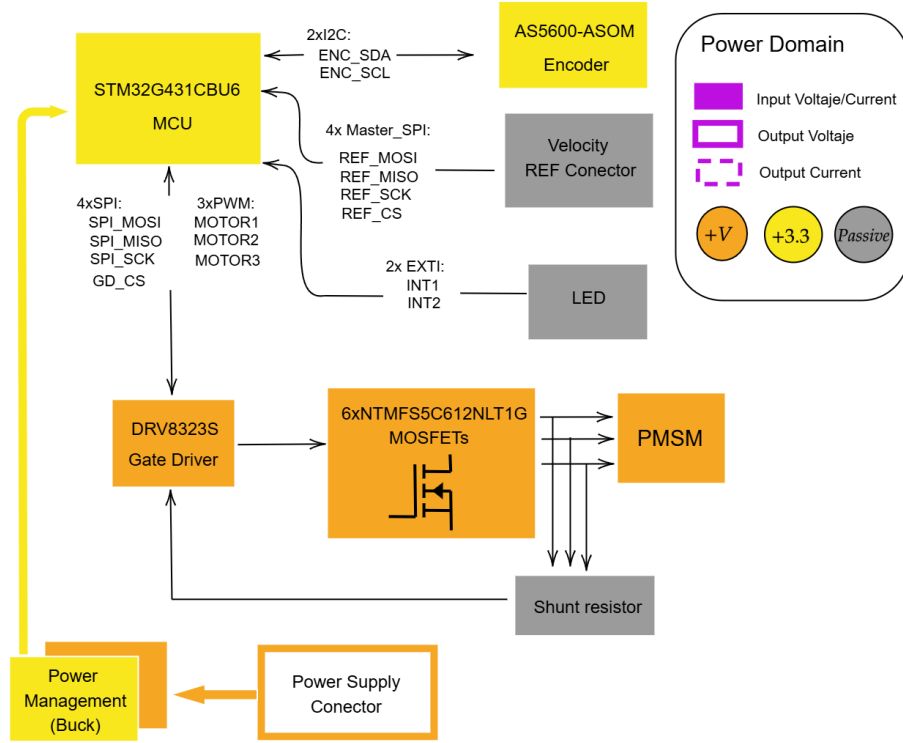


Illustration 1: Diagrama de bloques

4.2. Detalle por bloque

Bloque 1: Control y sensado (MCU, AS5600 e interfaces) El STM32G431CBU6 genera tres señales PWM (MOTOR1{3}) para accionar el inversor a través del DRV8323S. Mantiene **dos enlaces SPI con roles diferenciados**: (i) *MCU como maestro* hacia el DRV8323S (SPI_MOSI, SPI_MISO, SPI_SCK, GD_CS) para configuración y diagnóstico; (ii) *MCU como esclavo* en el **bus de referencia de velocidad** (REF_MOSI (entrada al MCU), REF_MISO (salida del MCU), REF_SCK (entrada), REF_CS (entrada activa-baja)), donde un maestro externo envía consignas. La posición/velocidad del rotor se adquiere por un bus I²C bidireccional con el AS5600 (ENC_SDA, ENC_SCL), operando el MCU como maestro. Además, se disponen dos líneas de interrupción externa (INT1, INT2) para eventos asincrónicos y una salida LED para señalización de estado. Todo el bloque funciona a +3,3 V provistos por el regulador *buck*.

Bloque 2: Potencia y conversión de energía (DRV8323S, MOSFETs, PMSM y alimentación) El DRV8323S recibe las señales PWM del MCU y conmuta seis MOSFET NTMFS5C612NLT1G dispuestos en tres medio-puentes para conformar un **inversor trifásico** que alimenta el PMSM. El *gate driver* gestiona la habilitación, los tiempos muertos y la protección, y expone el estado por SPI y/o pines de falla dedicados. El **Power Supply Connector** introduce el bus +V a la etapa de potencia y al **Power Management (Buck)**, que regula a +3,3 V para la electrónica de control, manteniendo separados los dominios de potencia y lógica.

Bloque 3: Sensado de corriente (hardware presente; deshabilitado en esta versión) El trayecto de **sensado de corriente** mediante *resistor shunt* y acondicionamiento analógico se encuentra *implementado a nivel de hardware* y cableado al DRV8323S/ADC del MCU. No obstante, **no será utilizado por el firmware inicial**: por restricciones de tiempo no se implementará control *FOC* ni lazos de corriente (ni i_d ni i_q). La primera versión ejecutará **control simple de velocidad** basado únicamente en realimentación del encoder. El sensado de corriente y el *FOC* quedarán planificados como **actualización futura** del firmware, reutilizando el hardware existente sin cambios en la placa.

5. Diseño de firmware propuesto

5.1. Detalle de drivers

El firmware se organiza en controladores de bajo nivel asociados a cada elemento del sistema:

- **Driver de comunicaciones para referencia:** módulo de recepción de consignas desde el exterior. Entrega una bandera de *referencia válida* y el valor de consigna.
- **Temporización:** temporizador/SysTick para retardos y planificación de tareas; se utiliza un sondeo con espera de 0,5 ms cuando no hay referencia disponible.
- **Encoder:** adquisición de posición/velocidad del motor a través del sensor de posición. Expone la velocidad filtrada al lazo de control.
- **Driver PWM (TIMx):** generación de tres salidas PWM para el inversor trifásico. Permite habilitar/deshabilitar el puente y actualizar el ciclo útil en tiempo real.
- **Driver del gate driver (DRV8323S):** interfaz de configuración y diagnóstico del controlador de compuertas; aplica las órdenes de encendido/apagado y supervisa el estado de fallas.
- **Driver de GPIO de estado:** control de indicadores (LED) y líneas de interrupción si corresponde.

5.2. Diagrama de flujo

El flujo funcional se resume en los siguientes pasos:

1. **Inicio:** inicialización de temporizadores, GPIO, comunicaciones, encoder, PWM y *gate driver* (salidas deshabilitadas).
2. **Recepción de referencia:** si no existe una referencia válida, se espera 0,5 ms y se repite el sondeo.
3. **Lectura de velocidad:** una vez disponible la referencia, se obtiene la velocidad del motor desde el encoder.
4. **Cálculo en STM32:** se ejecuta el cómputo de variables de control y se determina el ciclo útil de PWM para las tres fases.
5. **Accionamiento:** se actualizan las salidas PWM; el *gate driver* DRV8323S conmuta los MOSFET del inversor.
6. **Ejecución y realimentación:** el motor gira, se cierra el bucle mediante la nueva lectura del encoder y se regresa al paso de cálculo.

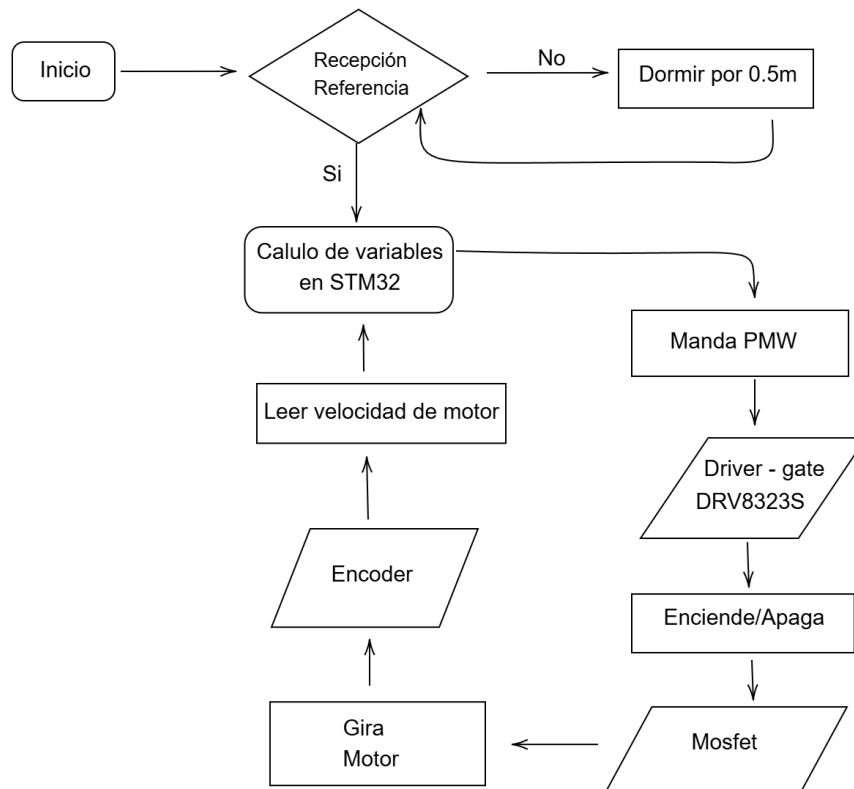


Illustration 2: Diagrama de Flujo para el firmware

5.3. Detalle por módulo

Módulo 1: Adquisición y referencia Este módulo gestiona la llegada de la consigna externa y la medición de la velocidad del motor. Implementa el sondeo con espera de 0,5 ms cuando no hay referencia, convierte la posición del encoder en velocidad utilizable por el control y publica los datos al *scheduler*. Sus principales interfaces son: entrada de *Referencia* (comunicación), entrada de *Encoder* y salida de *Velocidad estimada* más la bandera de *Referencia válida*.

Módulo 2: Control y actuación Este módulo ejecuta el cálculo en el STM32 para obtener el ciclo útil de las tres fases y actualiza el generador PWM. Coordina el encendido/apagado del *gate driver DRV8323S* y, por ende, la conmutación de los MOSFET. Sus interfaces son: entradas de *Referencia* y *Velocidad*, salidas *PWM_A/B/C* hacia el *gate driver* y líneas de *estado/diagnóstico* para supervisión.

6. Pruebas

6.1. Prueba sobre elementos

Prueba 1 Integridad eléctrica y alimentación. Se alimenta el bus $+V \in [10, 40]$ V y se verifica la regulación de +3,3 V en arranque y régimen, el rizado a la frecuencia f_{PWM} y la correcta separación de masas de potencia y lógica; como metas prácticas se considera $V_{3,3}$.

Prueba 2 Interfaces digitales y señalización. Se valida el SPI con el **DRV8323S** (MCU como maestro: lectura/escritura de registros, ID y banderas), el SPI de referencia (MCU como esclavo: **REF_CS** activo-bajo, recepción de consignas), el I²C con el **AS5600** (lectura de ángulo/velocidad), la generación PWM (frecuencia y resolución) y las GPIO/EXTI de estado.

6.2. Prueba sobre funcionalidades

Prueba 1 Arranque y control de velocidad en lazo cerrado *sin FOC*. Con el **DRV8323S** habilitado, se aplican referencias tipo escalón y rampa dentro del rango operativo, registrando referencia, velocidad del encoder, ciclo útil y estado del driver para distintos niveles de $+V$.

Prueba 2 Gestión de fallas y seguridad funcional. Se induce UVLO, se fuerza **nFAULT** y se acciona el *kill-switch* lógico para comprobar deshabilitación inmediata de PWM, registro/notificación de la causa y recuperación controlada.

7. Cronograma

El cronograma se compacta en **cuatro bloques quincenales** más la **entrega** aproximada, cubriendo desde el 13 de octubre hasta el 1 de diciembre de 2025. Las columnas representan ventanas de trabajo con solapamientos controlados.

	13–26 Oct	27 Oct–9 Nov	10–23 Nov	24–30 Nov	1 Dic
Diseño esquemático	X				
Diseño PCB	X	X			
Compra componentes		X			
Compra y envío de PCB		X			
Integración			X	X	
Desarrollo firmware		X	X	X	
Pruebas				X	X
Documentación y entrega				X	X

Tabla 1: Cronograma compacto hasta la entrega del 1 de diciembre de 2025.

8. Costos

Descripción (por <i>driver</i>)	1 unidad [CLP]	100 unidades [CLP]
STM32G431C8U6 (1×)	\$5,667	\$361,800
DRV8323S (gate driver, 1×)	\$3,998	\$248,600
NTMFS5C612NLT (MOSFET, 6×)	\$13,374	\$636,600
AS5600–ASOM (encoder, 1×)	\$3,192	\$208,300
MAX15062BATA+T (dc-dc buck, 1×)	\$2,442	\$189,300
PCB 4×4 cm, 4 capas + envío (1×)	\$10,000	\$1,000,000
Pasivos y reguladores (R, C, etc.)	\$3,000	\$300,000
Total (BOM parcial)	\$41,673	\$2,944,600

Tabla 2: Costos por *componente* en CLP. Los valores provienen de Mouser y consideran 6 MOSFET por driver. Proyección lineal a 100 unidades, con descuentos por volumen.