

Robot Sumo de 3 kg: Reporte de Proyecto

César Alejandro Tolentino Mendoza*, Carlos Daniel Barrera Carrillo*, Aitor Sebastián Larios Cuevas*

*Ingeniería en Sistemas Electrónicos y de Telecomunicaciones, Universidad de Colima, Colima, México
{ctolentino, cbarrera, alarios}@ucol.mx

Resumen—Este documento presenta el desarrollo completo del proyecto de un robot sumo de 3 kg, denominado *Coquineitor*. Se describen los objetivos del diseño, la selección y caracterización de los componentes electrónicos y mecánicos, la arquitectura de control basada en comunicación SPI entre el PIC18F4550 y un CPLD MAX II, así como la estrategia de combate implementada mediante una máquina de estados finita con algoritmos de detección y evasión de bordes y oponentes.

Palabras clave—Robot sumo; PIC18F4550; CPLD MAX II; SPI; máquina de estados; control PID; detección de oponente.

I. INTRODUCCIÓN

La robótica de competencia representa un espacio formativo único en el que convergen diversas áreas de la ingeniería, como electrónica, mecánica, control y programación. Entre las modalidades más populares se encuentra el **robot sumo**, donde dos o más robots se enfrentan en un *dohyo* con el objetivo de expulsar al contrincante sin abandonar ellos mismos la pista de combate. En la categoría de 3 kg, los participantes deben asegurar que sus prototipos cumplan requisitos estrictos de masa y dimensiones, así como demostrar un alto rendimiento en empuje y maniobrabilidad.

El presente anteproyecto describe la concepción de un **robot sumo de 3 kg** denominado *Coquineitor*, diseñado para participar en competencias nacionales organizadas por la Unidad Profesional Interdisciplinaria en Ingeniería y Tecnologías Avanzadas (UPIITA) del Instituto Politécnico Nacional (IPN). El proyecto es desarrollado por el Equipo de Carbones Calavera “La Resurrección”. La estructura central de la propuesta se basa en la combinación de:

- Un **microcontrolador PIC18F4550**, encargado de la gestión de sensores y la ejecución de estrategias de control (p. ej., algoritmos PID).
- Una **FPGA Basys2**, destinada a la generación de señales PWM de alta precisión, liberando al microcontrolador de tareas de modulación (posteriormente se cambiará a un CPLD EPM240T100C5).
- Un **driver BTS7960** para el manejo de los motores de potencia, con la capacidad de suministrar corrientes elevadas requeridas en las embestidas características de la disciplina.

I-A. Motivación y contexto académico

Este proyecto surge de la necesidad de impulsar a estudiantes de Ingeniería en Sistemas Electrónicos y de Telecomunicaciones a integrar conocimientos adquiridos durante

su formación. La participación en torneos de robots sumo, particularmente en la categoría de 3 kg, promueve:

1. La experimentación práctica de principios de electrónica analógica y digital.
2. El desarrollo de destrezas en programación de bajo nivel y dispositivos lógicos programables.
3. El diseño mecánico enfocado en la optimización de peso y resistencia al impacto.
4. El análisis de factibilidad financiera y la gestión de recursos, indispensables en proyectos de ingeniería complejos.
5. La integración de antenas: propuesta, identificación del tipo adecuado y simulación del patrón de radiación.
6. La elaboración y análisis de estados financieros y contables del proyecto mediante hojas de cálculo, incluyendo comentarios y observaciones.
7. La implementación y prueba en físico de la arquitectura de microcontroladores, abarcando la selección del microcontrolador, la circuitería de sistema mínimo, drivers de motores y sensores.
8. El diseño de comunicaciones móviles para la transmisión de datos del robot.
9. El diseño y descripción de dispositivos en VHDL: adquisición de motores y drivers H-bridge, y comunicación entre CPLD/FPGA y microcontrolador.
10. La elaboración de diagramas de ingeniería: diagramas a bloques del robot, esquemáticos electrónicos y diagrama de conexiones de comunicación.
11. La preparación y presentación en inglés (Inglés VI), incluyendo la elaboración de una video-presentación.
12. La redacción del anteproyecto conforme a todas las secciones requeridas, con control estricto de formato, gramática y ortografía.
13. La calidad de la presentación final: diapositivas equilibradas en texto e imágenes y participación balanceada de los miembros del equipo.

I-B. Resumen de las normas de la competencia UPIITA

En competencias organizadas por la UPIITA (IPN), se establecen lineamientos estrictos para asegurar la equidad y la calidad técnica de los enfrentamientos. De manera general, las reglas contemplan:

- **Dimensiones y peso:** El robot debe pesar un máximo de 3 kg y ajustarse a las dimensiones establecidas al inicio (por lo general, 20 cm × 20 cm, sin restricciones de altura).

- **Zona de combate (dohyo):** Es un círculo con un diámetro específico (aprox. 154 cm) con un borde blanco. Abandonar esta zona implica la eliminación inmediata.
- **Tiempo de combate:** Cada ronda dura un máximo de tres minutos. Si ninguno de los robots es expulsado, se aplican criterios adicionales para definir al ganador.
- **Prohibiciones técnicas:** No se permiten sustancias que se adhieran al dohyo para mejorar la tracción, ni interferir con señales de oponentes o usar sistemas de RF no permitidos.

I-C. Imagen Ilustrativa de un Robot Sumo

A fin de contextualizar la apariencia y configuración habitual de estos robots, en la Figura 1 se muestra un ejemplo de prototipo de robot perteneciente a esta categoría.



Figura 1. Ejemplo ilustrativo de un robot sumo, mostrando la tracción trasera y un diseño frontal inclinado para facilitar el empuje.

Coquineitor opera de manera autónoma, basándose en sus sensores de detección de líneas y oponentes para decidir sus movimientos. De esta forma, el diseño del firmware y la lógica de control son piezas fundamentales del éxito en el dohyo.

I-D. Estructura general de este documento

El presente informe está organizado en secciones numeradas para facilitar su lectura y análisis.

- Introducción.
- Planeación.
- Análisis Financiero / Presupuestal.
- Diseño del Robot (Mecánico y Electrónico).
- Lógica de Control y Protocolos de Comunicación.
- Posible Implementación Modular de Comunicación Inalámbrica.
- Integración del Bootloader y Avances en Programación.
- Arquitectura Maestro–Esclavo y Gestión de Periféricos.
- Librerías y Drivers de Sensores.
- Encoders de Ratón y Cálculo de Desplazamientos.
- Comunicación SPI e Implementación en CPLD MAX II.
- Máquina de Estados en el PIC18F4550.
- Estado de Calibración Guiada por LEDs.
- Conclusión del Firmware.
- Conclusión General y Bibliografía.

I-E. Relevancia e Impacto Esperado

El proyecto no solo busca triunfar en competencias de sumo robótico, sino también constituir un prototipo pedagógico que permita al equipo mejorar sus destrezas en hardware y software. La experiencia acumulada en temas de control, electrónica de potencia, diseño mecánico y administración de recursos económicos se convierte en una aportación a la formación integral del equipo como futuros ingenieros.

En los siguientes capítulos se profundizará en la metodología de planeación y se documentarán los pasos específicos para llevar a buen término la construcción de Coquineitor, siempre alineados a las reglas y exigencias de la competencia UPIITA del IPN, y de los puntos a evaluar en la rúbrica proporcionada por los profesores.

II. PLANEACIÓN

En este capítulo se detallan los aspectos clave de la planeación del robot sumo de 3 kg, incluyendo la **justificación de los componentes** seleccionados, el **presupuesto inicial** y la **elección del microcontrolador** PIC18F4550 y la familia PIC18 para el control principal. El objetivo es consolidar un sistema robusto que priorice la potencia de empuje, sin descuidar la maniobrabilidad ni la integración de sensores.

II-A. Justificación de los Componentes Principales

La elección de cada componente responde a la necesidad de equilibrar **alto empuje**, maniobrabilidad y fiabilidad en el dohyo, cumpliendo con los requerimientos y limitaciones mencionadas en capítulos anteriores.

II-A1. Motorreductor (12 V, 80–100 rpm, torque 2.5–3 kg cm): Ya se ha mencionado la importancia de contar con un motor que ofrezca suficiente torque para empujar al oponente. El motorreductor seleccionado:

- Opera a 12 V y brinda entre 80–100 rpm bajo carga, generando un **torque nominal de 2.5–3 kg cm**.
- Satisface la demanda de *alto empuje* y un control de velocidad adecuado para maniobras en el dohyo.
- Su **reductora interna** permite al robot sostener esfuerzos prolongados sin un consumo excesivo de corriente.

II-A2. Driver de Motor: BTS7960: Este módulo se encarga de manejar las corrientes requeridas por el motor en arranques y choques. El BTS7960:

- Soporta **corrientes pico de hasta 43 A**, lo cual es esencial para embestidas y arranques bruscos.
- Opera en configuración de puente H, facilitando el control de dirección y la modulación de velocidad mediante PWM.
- Incorpora protecciones contra sobrecorriente y sobrecalentamiento, incrementando la seguridad del sistema.

II-A3. Comparativa de Rodamiento: Orugas vs. Ruedas: Para transmitir la potencia al piso, se contemplaron dos enfoques principales:

II-A3a. Orugas de Caucho:

- Distribuyen el peso sobre un área de contacto mayor, ofreciendo tracción elevada en superficies lisas.
- Podrían brindar mayor estabilidad al robot, reduciendo la posibilidad de derrapes.
- Sin embargo, una fricción excesiva puede dificultar los giros, haciendo que el robot sea menos maniobrable.

II-A3b. Ruedas de Alta Fricción:

- Su menor área de contacto permite giros más rápidos y con mayor precisión.
- Seleccionar un material blando (poliuretano o caucho) incrementa la adherencia, evitando patinajes excesivos.
- Con un diámetro de 60–70 mm, se logra un buen equilibrio entre fuerza de empuje y velocidad lineal.

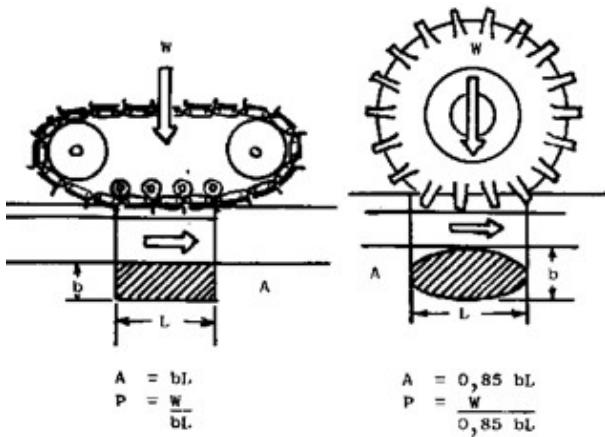


Figura 2. Comparativa entre orugas de caucho y ruedas de alta fricción.

Tras analizar ambas alternativas, se decidió utilizar ruedas de alta fricción debido a que facilitan la maniobrabilidad, simplifican la implementación mecánica y mantienen un empuje adecuado.

II-A4. Sensores de Línea: QTR-8A: Para detectar la línea blanca del dohyo y evitar salir de él, se utilizan **sensores IR reflectivos**:

- El arreglo QTR-8A combina 8 emisores y receptores IR que cubren un rango frontal amplio.
- Permite calibrar umbrales de detección (2–3 cm antes del borde), asegurando reacciones oportunas.

II-A5. Sensores de Distancia: VL53L0X: Para identificar al oponente y descartar objetos fuera de un rango predefinido:

- Ofrece lecturas Time-of-Flight (ToF) hasta **200 cm**, con la posibilidad de establecer un umbral de detección.
- Ayuda a iniciar la embestida cuando el rival se encuentra a una distancia corta, optimizando la estrategia de ataque.

II-A6. Batería LiPo 3S (11.1 V, 1500–2000 mAh): La alimentación del sistema se confía a una batería de litio-polímero:

- Proporciona un voltaje nominal de 11.1 V (12.6 V al 100 % de carga).
- Ofrece una alta densidad de energía en un formato ligero.

II-A7. Step-down (11.1 V → 5 V): Se incluye un convertidor *buck* para alimentar el PIC18F4550 y otros periféricos de 5 V:

- Garantiza un suministro estable y eficiente para la lógica de control.
- Permite manejar la diferencia de voltaje (11.1 V nominal a 5 V) sin desperdiciar excesiva potencia en calor.

II-A8. Level Shifter (5 V ↔ 3.3 V): Para la **FPGA Basys2** y posibles sensores de 3.3 V, se requiere un adaptador de niveles lógicos:

- Asegura la compatibilidad entre señales a 5 V (PIC) y 3.3 V (FPGA, sensores).
- Evita daños por sobrevoltaje y garantiza una comunicación confiable.

II-A9. Chasis de Aluminio: Se elige lámina de calibre 14-16 de espesor para la base:

- Combina ligereza y resistencia a impactos.
- Facilita el montaje de motores, driver y sensores, manteniendo la forma trapezoidal recomendada para un mejor ángulo de ataque.

II-A10. Tornillería, Cables y Cargador: Estos elementos, aunque consumibles, son indispensables:

- Tornillería y cables:** Permiten la sujeción de componentes y la interconexión eléctrica, respectivamente.
- Cargador LiPo balanceador:** Mantiene la batería en óptimas condiciones de carga y seguridad.

II-B. Selección del Microcontrolador y la Familia PIC18

Por qué PIC18F4550:

- Arquitectura de 8 bits optimizada:** Suficiente para la lógica de control, lectura de sensores y generación de PWM.
- Periféricos integrados:** Contiene módulos CCP/ECCP, USART, SPI e I2C, simplificando la comunicación con sensores y la FPGA.
- Capacidad de USB:** Se aprovecha el bootloader USB integrado para realizar actualizaciones de firmware en campo y diagnósticos sin herramientas externas.
- Amplio soporte y documentación:** La familia PIC18 es ampliamente conocida en la comunidad de robótica, con ejemplos y librerías disponibles.

Esta familia de microcontroladores ofrece una **buena relación costo-beneficio**, un entorno de desarrollo estable (MPLAB, XC8) y flexibilidad para implementar el control PID, lectura de línea y distancia, y la gestión de los parámetros PWM.

II-C. Presupuesto Inicial

La Tabla I presenta el desglose de costos aproximados para la etapa inicial, basados en referencias de mercado (Mercado Libre, AliExpress, etc.). No se incluyen costos indirectos (laboratorio, CNC, etc.) ni patrocinios potenciales.



Figura 3. Ilustración de algunos componentes seleccionados, mostrando motores, driver y sensores utilizados en el robot sumo.

Cuadro I
PRESUPUESTO INICIAL DEL PROYECTO

Componente	Cantidad	Precio (MXN)
PIC18F4550	1	\$120
Driver BTS7960	1	\$200
Motorreductor 12 V (80–100 rpm)	2	\$400 (total)
QTR-8A (Sensores de línea)	1	\$300
VL53L0X (Sensor de distancia)	2	\$400 (total)
Batería LiPo 3S	1	\$500
Step-down (11.1 → 5 V)	1	\$80
Level Shifter (5 V–3.3 V)	1	\$60
Ruedas alta fricción (par)	1	\$250
Chasis Aluminio (2 mm)	1	\$200
Tornillería, cables	1 set	\$200
Cargador LiPo	1	\$400
Total		\$4,660

II-D. Conclusiones de la Planeación

La planeación se enfoca en garantizar la **fuerza de empuje** necesaria para el robot sumo de 3 kg, seleccionando un motorreductor robusto y un driver capaz de manejar picos de corriente elevados. El uso de ruedas de alta fricción optimiza la tracción, mientras que los sensores (QTR-8A y VL53L0X) y el microcontrolador PIC18F4550 aseguran lecturas precisas y un control eficiente del comportamiento en el dohyo. Además, la integración del bootloader USB permite realizar actualizaciones de firmware y diagnósticos en campo sin herramientas externas, agilizando el desarrollo. Finalmente, el presupuesto estimado de \$4,660 MXN sirve como referencia financiera para comparar con los gastos reales y ajustar el plan económico, garantizando la viabilidad técnica y económica del proyecto en las siguientes etapas.

III. ANÁLISIS FINANCIERO / PRESUPUESTAL

En este capítulo se retoma el **presupuesto inicial** del proyecto (presentado en el Capítulo 2) y se compara con los **gastos reales** documentados recientemente. Adicionalmente, se clasifican los elementos en **gastos, inversiones y costos**,

con base en definiciones breves de estos conceptos, para luego relacionarlos directamente con el proyecto del robot sumo de 3 kg.

III-A. Breve Introducción a Pagos, Gastos e Inversión

- **Pago:** Desembolso puntual de efectivo para adquirir un bien o servicio. Por ejemplo, la compra de un driver de motor es un pago único.
- **Gasto:** Consumo de recursos durante la operatividad inmediata, sin valor residual a futuro (e.g., baterías, materiales de ensamblaje).
- **Inversión:** Desembolso que aporta valor más allá de la etapa actual, permitiendo escalabilidad o mejoras en el proyecto (e.g., microcontrolador, FPGA, motores).

Estas categorías ayudan a planificar mejor los recursos, pues diferencian lo que es puramente operativo de lo que contribuye al desarrollo a largo plazo.

III-B. Presupuesto Inicial Recapitulado

En el Capítulo 2 se estimó un presupuesto de aproximadamente \$4,660 MXN, distribuidos de la siguiente manera:

Cuadro II
PRESUPUESTO INICIAL (RESUMEN)

Concepto General	Monto Estimado (MXN)
Microcontrolador, driver, motorreductores, sensores (línea y distancia), step-down, shifter, batería, ruedas, chasis, cables, cargador	4,660

Este cálculo consideraba referencias de tiendas en línea (Mercado Libre, AliExpress) y tiendas locales, sin incluir costos indirectos (uso de taller, equipo de laboratorio) ni patrocinios.

III-C. Gastos Reales Documentados

La **tabla de compras reales** se muestra en la Tabla III, la cual se extrajo de registros contables del proyecto. Cada ítem indica su precio unitario y total, así como la cuenta contable (*Inventarios, Banco, etc.*) y una breve descripción.

Cabe señalar que estos valores y cálculos se encuentran anexados en la carpeta de OneDrive donde se ubica el archivo de este anteproyecto.

En esta compra real, se observa un subtotal de 2046.45 MXN correspondiente a los materiales inventariados, algunos de los cuales difieren en costo unitario con respecto al presupuesto inicial. A ello se suman gastos operativos, envíos y otros consumibles que no aparecen en esta tabla parcial.

III-D. Comparación con el Presupuesto Estimado

Aunque la compra real (2046.45 MXN) se sitúa por debajo del monto total inicial de 4660 MXN, se deben considerar otros costos que aún no se reflejan, como la **batería LiPo**, las **ruedas de alta fricción**, la **lámina de aluminio** y el **cargador LiPo**. Varios de estos ítems se etiquetan como gastos o inversiones según su utilidad a largo plazo:

Cuadro III
COMPRAS REALES DEL PROYECTO (EXTRACTO)

Material	Cant.	Total (MXN)	Comentarios
Buck DC-DC (12 V a 5 V)	1	301.35	Regulador principal
Conv. Lógico (5–3.3 V)	1	103.55	Para FPGA
Kit de cables	1	139.89	Conexiones internas
Kit de tornillos	1	150.00	Ensamble mecánico
PIC18F4550	1	299.68	MCU principal
Sensores línea	1	139.20	Detección dohyo
QTR-8A			
Tarjeta desarr. PIC18F4550	1	86.99	Pruebas microcontrolador
BTS7960 (2 und.)	2	238.80	Driver motor
Motorreductor 12 V (2 und.)	2	273.22	Tracción principal
Placas MDF (2 und.)	2	80.00	Soportes/prototipo
Sensor láser VL53L0X (2 und.)	2	163.78	Detección rivales
Total Inventarios		2046.45	

- **Inversión:** Microcontrolador, driver BTS7960, motorreductores, sensores de distancia, tarjeta de desarrollo, convertidores lógicos (pues permiten mejoras futuras).
- **Gastos:** Batería LiPo, sensores de línea (desgaste operativo), cables, MDF (usado en prototipado).

Se prevé que, al añadir la compra de la batería, las ruedas y el chasis de aluminio, el gasto total se aproxime o supere la cifra de 4660 MXN inicial.

III-E. Clasificación de los Elementos en el Proyecto

Basados en los criterios de *Gastos* e *Inversión* descritos al inicio, la mayoría de los componentes electrónicos que no se desgastan o no requieren reemplazo periódico se consideran **Inversión**, mientras que los materiales consumibles se clasifican como **Gastos**. Algunos ejemplos:

- **Inversión:**
 - **PIC18F4550:** Componente clave de control, reutilizable en mejoras futuras.
 - **BTS7960:** Driver de motor con vida útil prolongada.
 - **Motores DC:** Reaprovechables en otras fases o versiones del robot.
 - **Tarjeta de Desarrollo:** Permite prototipar múltiples proyectos con el mismo MCU.
- **Gasto:**
 - **Sensores de línea (QTR-8A):** Se degradan con el uso y podrían requerir sustitución.
 - **Placas de MDF y Tornillería:** Empleados en la fabricación, sin valor residual significativo.
 - **Batería LiPo:** Se agota con los ciclos de carga y requiere reemplazo periódico.

III-F. Balance General y Cuentas Contables

Para el periodo del 04 de febrero al 27 de marzo de 2025, se presenta un Balance General que resume la situación financiera del proyecto:

ROBOT SUMO		
Balance General		
Del 04 de Febrero al 27 de Marzo del 2025		
ACTIVO	PASIVO	
ACTIVO CIRCULANTE	PASIVO CIRCULANTE	
Banco	73.67	Cuentas por pagar
Inventarios	2046.45	TOTAL PASIVO
		369.89
		TOTAL CAPITAL
	Capital social	1750.23
	TOTAL CAPITAL	1750.23
TOTAL ACTIVO	2120.12	TOTAL PASIVO + CAPITAL
		2120.12

Figura 4. Balance General del Proyecto Robot Sumo, mostrando Activos, Pasivos y Capital.

En este Balance General:

- **Activo Circulante:** Se compone de:
 - *Banco*: \$73.67, que representa la liquidez disponible para gastos menores.
 - *Inventarios*: \$2046.45, donde se incluyen los materiales y componentes adquiridos (por ejemplo, PIC18F4550, BTS7960, sensores, etc.).
- **Pasivo Circulante:** Corresponde a *Cuentas por pagar* por \$369.89, reflejando deudas a proveedores aún pendientes.
- **Capital:** Compuesto por un *Capital social* de \$1750.23, aportado por los integrantes o patrocinadores iniciales.

La suma total del Activo (2120.12) coincide con la del Pasivo más el Capital (2120.12), cumpliendo la ecuación contable básica:

$$\text{Activo} = \text{Pasivo} + \text{Capital}$$

Este estado financiero confirma que el proyecto mantiene un equilibrio en su contabilidad. A medida que se adquieran más componentes (ruedas, chasis de aluminio, etc.), se incrementará el valor de *Inventarios* o bien se reflejará un incremento en el *Pasivo* (si se compra a crédito) o una reducción del *Banco* (si se compra al contado).

III-G. Conclusiones del Análisis Financiero

La comparación del presupuesto inicial (\$4660 MXN) con los gastos reales (\$2046.45 MXN, sin contar elementos pendientes) revela que el **proyecto se mantiene dentro de un rango razonable de inversión y gastos**, aunque se espera que el monto final se acerque más a la cifra original al incorporar ítems faltantes (batería, ruedas, chasis de aluminio, etc.).

La clasificación en gastos e inversiones contribuye a la planificación a largo plazo:

- **Inversiones** como el microcontrolador, drivers y motores sostienen el valor del proyecto y facilitan expansiones futuras.

- **Gastos** como baterías y consumibles se asumen como parte de la operatividad inmediata y no generan valor residual.

Con esta perspectiva financiera, se pueden gestionar de manera más eficiente los recursos, ya sea buscando patrocinios para cubrir componentes estratégicos (inversiones) o destinando un presupuesto específico para gastos de uso continuo y reemplazo, asegurando la viabilidad económica del robot sumo de 3 kg.

IV. DISEÑO DEL ROBOT (MECÁNICO Y ELECTRÓNICO)

En este capítulo se describe el **diseño mecánico** y el **diseño electrónico** del robot sumo de 3 kg, tomando en cuenta la disposición de los componentes, la estructura del chasis y el *esquema electrónico preliminar*. Se hace énfasis en que el **esquema final** se encuentra en proceso de creación, debido a la investigación en curso sobre el funcionamiento interno de algunos de los módulos.

IV-A. Diseño Mecánico

El robot se ha concebido con una **base rectangular de 20 cm × 20 cm**, siguiendo las dimensiones máximas permitidas por la competencia. En la parte frontal, se integra una “**rampa ofensiva**” (o cuña) para facilitar la embestida y empujar al oponente fuera del dohyo.

IV-A1. Distribución Interna de Componentes: En la Figura 5, se aprecia la disposición preliminar:

- **Motores (2 und.)** en la parte media-baja, con ejes alineados para la tracción.
- **Ruedas de alta fricción**, de 59 mm de diámetro, a cada costado.
- **Placas de Desarrollo (PIC18F4550 y FPGA Basys2)** ubicadas en la sección central, para un cableado ordenado y fácil acceso.
- **Rampa ofensiva** (en el dibujo, “Base” frontal) con una extensión de unos 3–5 cm, para maximizar la zona de contacto al empujar.

IV-A2. Prototipo del Chasis (Versión v1): Como avance significativo, ya se ha desarrollado una **versión v1 del chasis**. En esta iteración se ajustaron las proporciones y la posición de los componentes dentro del área permitida, optimizando la distribución de masa y el espacio para la integración de la electrónica. Este prototipo permitió validar la viabilidad del diseño mecánico y realizar ajustes en la ubicación de la rampa ofensiva, motores y soporte para la batería.

IV-A3. Justificación del Chasis:

- **Material (Aluminio de calibre 14-16):** Combina ligereza y resistencia, soportando impactos en embestidas y protegiendo los componentes internos.
- **Forma Trapezoidal o Rectangular:** Se ajusta a las normas de 20 cm de lado y permite la incorporación de la cuña frontal.
- **Distribución de Componentes:** La versión v1 del chasis demostró que la ubicación centralizada para la batería LiPo (11.1 V nominal) y las placas de control favorece

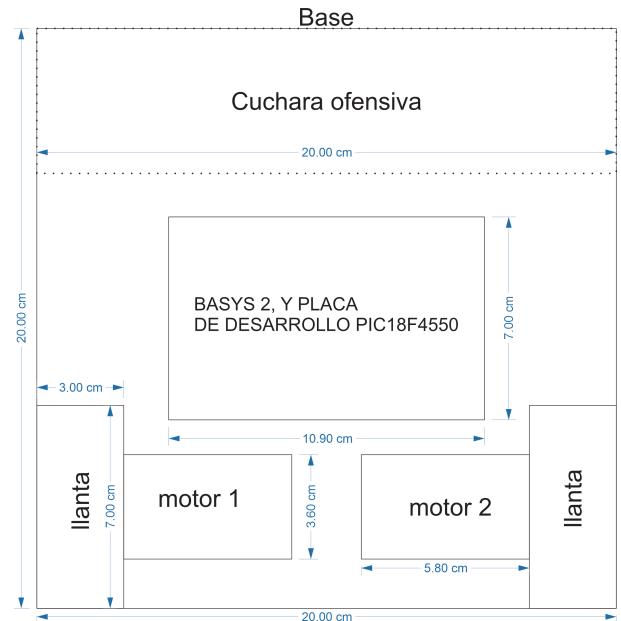


Figura 5. Diseño base (vista superior), mostrando la ubicación de motores, ruedas y placas de control.



Figura 6. Prototipo del chasis Versión v1, mostrando el ajuste de proporciones y posiciones de los componentes en el área de 20 cm × 20 cm.

un equilibrio óptimo, reduciendo el riesgo de desestabilización.

IV-B. Diseño Electrónico

El robot integra cuatro bloques funcionales (ver Figura 7):

1. **Bloque de Alimentación**
2. **Bloque de Control y Procesamiento**
3. **Bloque de Control de Movimiento**
4. **Bloque de Sensado**

IV-B1. Bloque de Alimentación:

- **Batería LiPo 3S (11.1 V nominal):** Provee la potencia requerida por los motores y la electrónica.
- **Convertidor buck (12 V → 5 V):** Alimenta el PIC18F4550 y otros periféricos de 5 V.
- **Level Shifter (5 V ↔ 3.3 V):** Ajusta niveles para la FPGA Basys2 y/o sensores a 3.3 V.

IV-B2. Bloque de Control y Procesamiento:

- **PIC18F4550:** Microcontrolador principal, encargado de la lectura de sensores (línea y distancia), ejecución de algoritmos de control (FSM, etc.) y la comunicación SPI con la FPGA.
- **FPGA Basys2:** Maneja la generación de PWM de alta precisión, descargando al PIC tareas de modulación, y procesando señales adicionales en paralelo.

IV-B3. Bloque de Control de Movimiento:

- **Driver de Motor (BTS7960):** Soporta corrientes pico de hasta 43 A, asegurando un empuje potente y controlado. Recibe la señal PWM de la FPGA.
- **Motores (12 V, 80–100 rpm):** Ubicados lateralmente para tracción diferencial, permitiendo giros precisos y embestidas eficientes.

IV-B4. Bloque de Sensado:

- **Sensores de Línea (QTR-8A):** Ocho emisores y receptores IR para detectar la franja blanca del dohyo y evitar salidas accidentales.
- **Sensores de Distancia (VL53L0X):** Proporcionan lecturas de Time-of-Flight hasta 200 cm, con posibilidad de establecer un umbral que permita filtrar objetos fuera del área de combate.

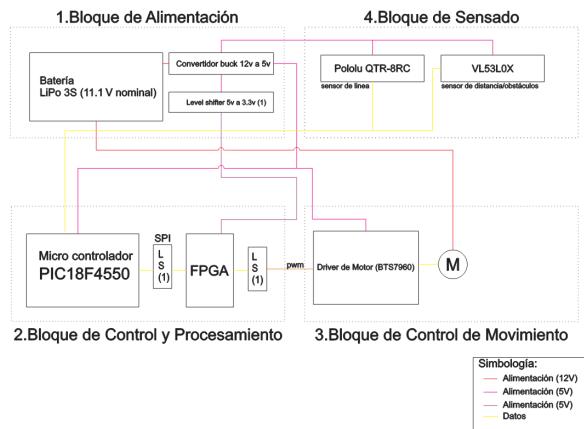


Figura 7. Diagrama de bloques del sistema electrónico, resaltando la alimentación, el control y el sensado.

IV-B5. Esquema Electrónico (en proceso): Actualmente se investiga el funcionamiento interno de algunos módulos y se está configurando la asignación de pines del PIC18F4550 para la comunicación SPI e I2C. Por ello, el **esquema electrónico definitivo** se encuentra en proceso de elaboración, adaptándose a los requerimientos de los fabricantes y a la lógica de control final.

IV-B5a. Justificación de la Elección de Componentes:

- **PIC18F4550:** Seleccionado por sus periféricos integrados (SPI, I2C, USB opcional) y su capacidad de procesamiento a 8 bits, adecuado para la ejecución de algoritmos de control.

- **FPGA Basys2:** Permite la generación de señales PWM de alta frecuencia y precisión, liberando al PIC de tareas de modulación.
- **BTS7960:** Ofrece robustez y capacidad para manejar altos picos de corriente, fundamentales en situaciones de combate.

IV-C. Conclusiones del Diseño Mecánico y Electrónico

El diseño mecánico se fundamenta en una estructura robusta de aluminio con una base de 20 cm × 20 cm, que en su versión v1 ya ha sido prototipada y ajustada para optimizar la distribución de masa y el espacio disponible. La arquitectura electrónica, dividida en bloques de alimentación, control, movimiento y sensado, asegura la integración y el funcionamiento coordinado de todos los subsistemas. Aunque el esquema electrónico final aún está en proceso, la distribución lógica de módulos y la selección de componentes garantizan que el robot cumpla con los requisitos de alto empuje y estabilidad en la competencia de sumo.

V. LÓGICA DE CONTROL Y PROTOCOLOS DE COMUNICACIÓN

En este capítulo se profundiza en la **lógica de control** y en los **protocolos de comunicación** implementados en el robot sumo de 3 kg. Se retoma lo discutido en capítulos anteriores sobre la coordinación entre el microcontrolador PIC18F4550, la FPGA Basys2, el driver de motor BTS7960 y los sensores, resaltando la relevancia de la comunicación SPI y los avances realizados en la *Basys2*.

V-A. Lógica de Control en el Microcontrolador PIC18F4550

El **PIC18F4550** gestiona la mayor parte de la lógica de control:

- **Lectura de Sensores:**
 - Sensores de línea (QTR-8A) conectados a puertos analógicos/digitales, determinan cuándo el robot se acerca a la franja blanca.
 - Sensores de distancia (VL53L0X o similares) para detectar la presencia del oponente a menos de 60 cm.
- **Ejecución de Algoritmos PID:** Permite ajustar la velocidad y dirección de los motores para corregir la trayectoria o embestir con mayor precisión.
- **Comunicación SPI con la FPGA:** Envía consignas de PWM, velocidad y otras señales de control para la modulación de potencia en el driver BTS7960.

Este enfoque libera al PIC de la generación de PWM de alta resolución, delegándola a la *Basys2*, mientras conserva recursos para procesar datos de sensores y tomar decisiones en tiempo real.

V-B. Protocolos de Comunicación

V-B1. SPI (Serial Peripheral Interface): La comunicación SPI se emplea entre el **PIC18F4550** (maestro) y la **FPGA Basys2** (esclavo). De esta manera:

- El PIC envía valores de **Duty Cycle** y otras banderas de control para generar la señal PWM.

- La FPGA recibe estos datos y ajusta internamente los registros que modulan el ciclo de trabajo de la salida PWM.

Este protocolo se escogió por su sencillez y **baja latencia**, ideal para sistemas de control que requieren reacciones rápidas.

V-B2. I2C (Inter-Integrated Circuit): Se contempla la posibilidad de usar I2C para **sensores de distancia** (ej. VL53L0X), aunque en algunas configuraciones puede emplearse un bus similar o la capa de comunicación de la propia FPGA.

V-B3. UART (Universal Asynchronous Receiver-Transmitter): El PIC18F4550 también ofrece un módulo EUSART que podría utilizarse para depuración o telemetría, en caso de implementarse una futura comunicación inalámbrica.

V-C. Avances en la FPGA Basys2

V-C1. Generación de PWM y Pruebas SPI: La **FPGA Basys2** se ha utilizado para comenzar las pruebas de la *lógica SPI* y la *generación de señales PWM*. Aunque el código VHDL definitivo está en desarrollo, se han implementado módulos básicos que permiten:

- **Registrar datos** (4 bits o más) provenientes del bus SPI, almacenándolos en un registro interno.
- **Cargar o desplazar dichos registros** según señales de habilitación y limpieza (CE, CLR, etc.).
- **Producir salidas PWM** ajustadas al valor recibido desde el PIC, garantizando una modulación estable a la frecuencia deseada.

V-C1a. Descripción de la Lógica: Los módulos en VHDL se dividen en entidades como `reg_ps`, `registro_p` y `registro_sp`. Cada uno:

- Declara puertos de entrada (p.ej., CLK, CE, SHIFT_LOAD) y salida (SLO, Q), así como señales internas para el almacenamiento de datos.
- Implementa un proceso sensible al flanco ascendente de CLK (y a la señal de CLR), reseteando los registros cuando CLR está activo.
- En modo *Load*, transfiere datos **directamente** a los registros; en modo *Shift*, desplaza los bits hacia la izquierda o la derecha, extrayendo el bit más significativo.

Esta estructura de *shift registers* y *parallel registers* forma la base de la interfaz SPI en la FPGA, permitiendo al **PIC18F4550** enviar valores de PWM u otras banderas de control. Posteriormente, la *Basys2* interpreta esos datos y ajusta la señal PWM que se entrega al driver de motor BTS7960.

V-C1b. Estrategia de Integración:

- Se conecta el bus SPI (MOSI, MISO, SCK, SS) del PIC a los pines correspondientes en la **Basys2**.
- Los registros `reg_ps` y `registro_p` reciben los bits seriales, transformándolos en un valor paralelo que alimenta el módulo PWM.
- Un **contador interno** en la FPGA determina la frecuencia de la señal PWM, mientras el valor paralelo define el **Duty Cycle**.

V-D. Estrategias de Control para Competencia

Para **ganar el combate** en el dohyo, el robot no solo debe embestir al oponente, sino también evitar ser empujado y evitar salirse de la pista. Algunas estrategias contempladas son:

V-D1. Detección Oportuna del Rival:

- Usar los **sensores de distancia** (VL53L0X) para identificar la posición del contrincante y aproximarse con *velocidad media*.
- Al estar a menos de 30–40 cm, aumentar la **velocidad de embestida** para sorprender y desplazar al rival.

V-D2. Mantenimiento en el Dohyo:

- Configurar los **sensores de línea** (QTR-8A) para detectar la franja blanca con al menos 2–3 cm de anticipación.
- Al reconocer la línea, el algoritmo PID reduce la velocidad del motor en ese costado, girando el robot hacia el centro.
- Se emplean maniobras suaves para no perder tracción ni ser empujado por inercia fuera del dohyo.

V-D3. Maniobras de Reposición y Contraembestida:

- Si el robot detecta que está siendo empujado, puede usar la **lectura de sensores de línea** para frenar y girar, tratando de forzar un cambio de dirección del rival.
- Se puede implementar un **modo de contraembestida**: al detectar un choque frontal, aumentar el **Duty Cycle** en los motores para ejercer un empuje adicional y ganar la fricción necesaria.

V-D4. Control PID Dinámico:

- Se ajustan **ganancias P, I y D** según el estado del combate: más agresivo al detectar al oponente, más conservador al acercarse al borde.
- El PID ayuda a compensar deslizamientos y a mantener la trayectoria deseada cuando el robot gira o frena bruscamente.

V-E. Conclusiones de la Lógica de Control y Comunicación

El diseño de la **lógica de control** se basa en delegar la generación de PWM de alta precisión a la **FPGA Basys2**, mientras que el PIC18F4550 se encarga de procesar la información de sensores y enviar consignas vía **SPI**. Esta separación de tareas:

- Permite **respuestas rápidas** y estables en la modulación de los motores (BTS7960).
- Optimiza el uso del microcontrolador para algoritmos de control (PID) y lectura de sensores.

Con respecto a la competencia, se plantean estrategias de *detección oportuna*, *mantenerte en el dohyo* y *contraembestidas* que, combinadas con un **control PID dinámico**, deberían maximizar las posibilidades de expulsar al oponente sin que el robot salga involuntariamente. La integración de la FPGA y el PIC en la comunicación SPI ya ha mostrado avances en la *Basys2*, sentando las bases para un sistema robusto de control en tiempo real.

VI. DISEÑO DEL ROBOT (MECÁNICO Y ELECTRÓNICO)

v2

En este capítulo se presenta una variante paralela del diseño inicial (versión v1), orientada principalmente a corregir posibles problemas de peso mal calculado o fuerzas no contempladas durante el combate, además de mejorar la detección del componente y establecer una relación lógica entre los componentes y su funcionamiento heredado.

VI-A. Diseño Mecánico v2

El diseño mecánico mantiene la forma básica del chasis rectangular, pero introduce modificaciones significativas en la gestión interna de pesos y componentes, especialmente con la incorporación de un soporte adicional mediante una bola loca y la optimización del centro de masa.

VI-A1. Distribución Interna Optimizada de Componentes: La versión v2 del robot mantiene la estructura base compuesta por una placa metálica principal, acompañada por un segundo piso fabricado en acrílico o MDF según disponibilidad y facilidad de mecanizado. La distribución optimizada sigue criterios claros:

- **Primer Piso (Metálico):** Aloja los componentes de mayor peso y fundamentales para la estabilidad estructural y dinámica del robot. En este nivel se encuentran la batería LiPo 3S, los motorreductores, el driver BTS7960 y el CPLD MAX II, garantizando un bajo centro de masa y proporcionando máxima estabilidad ante movimientos bruscos y colisiones durante el combate.
- **Segundo Piso (Acrílico o MDF):** Aloja elementos de peso significativamente menor, principalmente sensores (VL53L0X y QTR-8A) y el microcontrolador PIC18F4550, reduciendo el impacto negativo en la estabilidad dinámica del robot.

Adicionalmente, se incorpora una **bola loca** como tercer punto de apoyo. Este componente sencillo y eficaz permite mantener continuamente el equilibrio y evitar problemas generados por giros o aceleraciones súbitas, proporcionando un soporte fiable y minimizando los rozamientos indeseables con el dohyo.

La reorganización interna tiene como objetivo fundamental corregir errores potenciales en la distribución inicial de masas, garantizando así una respuesta más estable y predecible frente a fuerzas dinámicas inesperadas durante las maniobras de combate.

VI-A2. Cálculo del Centro de Masa: Para esta versión v2, se llevó a cabo un análisis detallado del **centro de masa** del robot, aspecto que no había sido considerado inicialmente en profundidad. Este cálculo resulta esencial debido a que un centro de masa mal ubicado puede causar desequilibrios, pérdida de tracción y menor estabilidad durante maniobras rápidas o choques.

La metodología empleada fue la siguiente:

1. **Inventario de componentes y pesos individuales:** Se registró con precisión la masa de cada componente

individualmente (batería, motores, driver, placas electrónicas, sensores, estructura metálica, etc.).

2. **Estimación geométrica inicial:** Cada componente fue posicionado tentativamente dentro del área permitida (20 cm × 20 cm), generando un modelo inicial.
3. **Cálculo matemático del centro de masa:** Utilizando la fórmula general del centro de masa para múltiples elementos:

$$\overrightarrow{CM} = \frac{\sum_i m_i \vec{r}_i}{\sum_i m_i} \quad (1)$$

Donde:

- \overrightarrow{CM} : vector que representa la posición del centro de masa.
- m_i : masa del componente i .
- \vec{r}_i : vector posición del componente i respecto a un sistema de referencia común.

4. **Ajuste práctico y validación experimental:** Despues del cálculo inicial, se realizaron ajustes mínimos en las posiciones reales de los componentes para optimizar aún más la estabilidad dinámica, validando el resultado mediante pruebas prácticas de inclinación y aceleración del prototipo.

Este proceso aseguró que el robot tuviese un centro de masa considerablemente bajo y cercano al centro geométrico de la estructura, optimizando así la estabilidad y maniobrabilidad durante los encuentros en el dohyo.

VI-B. Diseño Electrónico v2

La arquitectura electrónica para esta versión v2 conserva gran parte de los elementos fundamentales definidos en la versión inicial, pero introduce mejoras significativas enfocadas en la simplificación, eficiencia y economía del sistema. El cambio más relevante consiste en la sustitución definitiva de la FPGA Basys2 por un CPLD MAX II (modelo EPM240T100C5).

La motivación principal detrás de este cambio se basa en criterios prácticos, técnicos y económicos. La FPGA Basys2, inicialmente utilizada, era propiedad de la universidad, lo que limitaba su uso fuera del laboratorio y restringía el tiempo disponible para su desarrollo y depuración. Además, al tratarse de equipo costoso y delicado, existía una constante preocupación por posibles daños en condiciones reales de combate.

Por otro lado, el CPLD MAX II cumple con las mismas funcionalidades requeridas para este proyecto específico, ofrece un tamaño más compacto y una solución propia más económica. Al adquirir el CPLD directamente como equipo propio del equipo de desarrollo, se logró eliminar la dependencia de recursos externos, facilitando así la flexibilidad para realizar pruebas y modificaciones del firmware en cualquier entorno y horario, acelerando considerablemente el avance general del proyecto.

A continuación, se describen brevemente los principales bloques funcionales en esta nueva configuración electrónica v2.

VI-B1. Bloque de Alimentación (sin cambios sustanciales): El bloque de alimentación en la versión v2 mantiene la misma configuración básica implementada en la versión inicial, debido a su buen desempeño y adecuada funcionalidad durante las pruebas iniciales. Este bloque está compuesto por:

- **Batería LiPo 3S (11.1 V nominal, 1500–2200 mAh):** Continúa proporcionando la energía necesaria para el correcto funcionamiento de los motores y demás componentes electrónicos durante los encuentros. Su elección sigue siendo óptima en términos de peso, capacidad y corriente disponible.
- **Convertidor Buck (12 V → 5 V):** Permanece responsable de regular el voltaje principal proveniente de la batería hasta un valor estable y eficiente para alimentar el microcontrolador PIC18F4550, sensores QTR-8A y periféricos adicionales.
- **Level Shifter (5 V ↔ 3.3 V):** Se mantiene para garantizar la compatibilidad lógica entre los niveles de voltaje del PIC (5 V) y el CPLD MAX II y sensores VL53L0X (3.3 V), previniendo daños y asegurando una comunicación fiable y robusta entre estos componentes.

Dado que este bloque no presentó fallas significativas ni limitaciones detectables durante la versión anterior, no se justificaron cambios adicionales en esta versión v2.

VI-B2. Bloque de Control y Procesamiento: El bloque de control y procesamiento de la versión v2 mantiene el microcontrolador PIC18F4550 como unidad central debido a su adecuado desempeño en la gestión de sensores y ejecución lógica en la versión anterior. Sin embargo, el cambio más significativo radica en la sustitución definitiva de la FPGA Basys2 por un CPLD MAX II (EPM240T100C5).

- **Microcontrolador PIC18F4550 (sin cambios sustanciales):** Continúa siendo la unidad encargada de procesar la información proveniente de los sensores (QTR-8A y VL53L0X), administrar la lógica de la máquina de estados y comunicarse mediante protocolos SPI, I²C (implementado por software) y UART. Su elección permanece justificada por la disponibilidad de periféricos integrados, facilidad de programación y documentación extensa.
- **CPLD MAX II (EPM240T100C5):** Este componente reemplaza de manera definitiva a la FPGA Basys2 por razones técnicas y económicas, tales como:
 - *Autonomía y Flexibilidad:* Al tratarse de un dispositivo propio del equipo de desarrollo, se eliminan restricciones de horarios y dependencia de equipos externos o universitarios.
 - *Costo y robustez:* El CPLD es significativamente más económico y compacto que la FPGA utilizada previamente, ofreciendo una solución técnicamente equivalente en el contexto específico del robot sumo.
 - *Simplicidad de Integración:* Su menor tamaño y consumo energético facilita la integración física en el chasis, además de simplificar la gestión térmica y eléctrica del robot.

Esta reorganización del bloque de procesamiento permite al robot mantener un alto nivel de desempeño, simplificar su mantenimiento y depuración, y garantizar una mayor flexibilidad operativa durante todas las etapas de desarrollo y competencia.

VI-B3. Bloque de Control de Movimiento: El bloque de control de movimiento mantiene la filosofía de diseño implementada en la versión inicial, dado que los componentes seleccionados cumplieron adecuadamente con los requisitos de empuje, tracción y robustez durante las pruebas preliminares.

- **Driver de Motor BTS7960 (sin cambios):** Continúa siendo el encargado de manejar los motores de tracción diferencial. Su capacidad de soportar corrientes pico de hasta 43 A proporciona la robustez necesaria para las embestidas características de la competencia de sumo.
- **Motores DC (12 V, 80–100 rpm, torque 2.5–3 kg cm):** Los motores seleccionados ofrecen un equilibrio ideal entre fuerza de empuje y velocidad de respuesta. Se mantuvieron sin cambios en la versión v2, ya que demostraron un comportamiento eficiente en maniobras de ataque y evasión durante las primeras fases de pruebas.

En esta versión, las señales de control de los motores (PWM y dirección) continúan generándose en el CPLD MAX II, recibiendo parámetros de consigna desde el PIC18F4550 a través del protocolo SPI. Esto permite mantener una alta precisión en el control de velocidad y dirección, mientras se libera al microcontrolador de tareas de modulación de alta frecuencia.

VI-B4. Bloque de Sensado: En la versión v2, el bloque de sensado mantiene la esencia de la configuración anterior, pero introduce optimizaciones importantes en el uso de los sensores para simplificar el hardware y el procesamiento en el microcontrolador.

- **Sensores de Línea (QTR-8A, optimización en uso):** Aunque se siguen utilizando módulos QTR-8A, ahora se emplean dos placas y únicamente tres sensores por placa, seleccionados estratégicamente para cubrir las zonas críticas de detección de línea. Esta optimización reduce el número de entradas analógicas requeridas, simplifica el cableado y el procesamiento, manteniendo la capacidad suficiente para la detección oportuna de los bordes del dohyo. Además, se reconoce que en un robot sumo el objetivo principal es expulsar al oponente, siendo la evasión de la línea un comportamiento secundario.
- **Sensores de Distancia (VL53L0X, ampliación de cobertura):** Se incrementa la cantidad de sensores Time-of-Flight (ToF) VL53L0X, integrando tres unidades adicionales. Esto permite una cobertura más amplia alrededor del robot, mejorando significativamente la capacidad de detectar al oponente en diversas posiciones y ángulos, tanto frontales como laterales y traseros.
- **I²C por software para VL53L0X:** La comunicación con los sensores VL53L0X se implementa mediante I²C por bit-banging (software) en el PIC18F4550, lo cual proporciona flexibilidad para definir los pines utilizados

y facilita la expansión del número de sensores sin comprometer el resto de los periféricos de hardware.

Esta configuración optimizada mejora la confiabilidad del sensado, reduce la complejidad del sistema y optimiza el procesamiento dentro del PIC, factores clave para un desempeño ágil y robusto durante los enfrentamientos.

VI-B5. Justificación Técnica del Rediseño Electrónico: El rediseño electrónico implementado en la versión v2 responde a necesidades técnicas y económicas detectadas durante las fases de prueba y desarrollo de la versión inicial. A continuación, se explican las razones principales que motivaron las decisiones tomadas:

■ **Sustitución de la FPGA Basys2 por CPLD MAX II:**

El cambio se fundamenta en la necesidad de contar con un sistema propio, portátil y de menor costo. La FPGA Basys2 era un recurso universitario, costoso y restringido a uso en laboratorio. El CPLD MAX II cumple con las funcionalidades requeridas para este proyecto, presenta una integración más compacta, menor consumo de energía y permite trabajar fuera de la universidad, acelerando el desarrollo y reduciendo riesgos de daño a equipos institucionales.

■ **Mantenimiento de sensores QTR-8A y VL53L0X:**

Los módulos QTR-8A y VL53L0X demostraron ser confiables y adecuados en las primeras pruebas. Sin embargo, se optimizó el uso del QTR-8A limitando a tres sensores por placa, simplificando el hardware y la carga de procesamiento, ya que el objetivo principal es expulsar al oponente, no realizar un seguimiento exhaustivo del borde.

■ **Mejoras en simplicidad y confiabilidad:** La eliminación

del PID y su reemplazo por una máquina de estados en el PIC18F4550 representa una optimización sustancial. La máquina de estados ofrece una respuesta más rápida, robusta y predecible ante situaciones espontáneas del combate, evitando la necesidad de ajustes constantes de parámetros de control y simplificando el diseño de software.

■ **Optimización de comunicaciones:** Se mantienen los

protocolos SPI para comunicación PIC–CPLD, I²C por software para el manejo de los sensores VL53L0X y UART para la eventual implementación de telemetría inalámbrica. La asignación clara y delimitada de cada protocolo mejora la robustez del sistema y evita conflictos de periféricos.

En conjunto, estos cambios representan una evolución natural del proyecto, orientada a maximizar la eficiencia de recursos, incrementar la confiabilidad operativa en combate, y facilitar el trabajo de integración y pruebas en fases posteriores del desarrollo.

VII. LÓGICA DE CONTROL Y PROTOCOLOS DE COMUNICACIÓN V2

En este capítulo se describe la evolución de la lógica de control y de los protocolos de comunicación implementados en

la versión v2 del robot sumo de Coquineitor. A diferencia de la versión inicial basada en control PID, la nueva arquitectura lógica se fundamenta en una **máquina de estados finita** (FSM) programada en el microcontrolador PIC18F4550.

La transición hacia una máquina de estados se debe a la naturaleza altamente dinámica, rápida y espontánea de los combates de robot sumo, donde un control reactivo basado en estados concretos resulta más efectivo, robusto y sencillo de implementar que un control de tipo PID, el cual resulta más adecuado para sistemas continuos o de respuesta lenta.

Asimismo, se detallan los protocolos de comunicación entre el PIC, el CPLD y los sensores periféricos, optimizados para mejorar la robustez, la velocidad de respuesta y la simplicidad de integración.

A continuación se describen los bloques principales de control y comunicación de esta nueva versión.

VII-A. Máquina de Estados del PIC18F4550

La versión v2 del robot implementa una **máquina de estados finita** (FSM) en el microcontrolador PIC18F4550, eliminando por completo el esquema de control PID utilizado en la versión anterior.

Esta máquina de estados permite definir de manera clara y estructurada el comportamiento del robot frente a distintos eventos y situaciones del combate, facilitando una respuesta inmediata y predecible ante los estímulos del entorno.

Los principales estados definidos son los siguientes:

■ **INICIAL:**

- Configura periféricos esenciales (ADC, SPI, I²C, UART).
- Inicializa pines de sensores y librerías (VL53L0X y QTR-8A).
- Asigna direcciones únicas a cada sensor VL53L0X mediante el control de sus pines XSHUT.

■ **BUSCAR:**

- Avanza en línea recta durante un tiempo determinado.
- Realiza giros de exploración asistidos por lectura de encoders
- Monitorea constantemente las lecturas de distancia para detectar oponentes.

■ **ATACAR:**

- Alinea el robot hacia el oponente según las lecturas de los sensores VL53L0X.
- Ajusta la velocidad PWM a un valor alto (80–100 %) para efectuar la embestida.
- Verifica si el oponente se encuentra centrado o en una posición lateral.

■ **EMPUJANDO:**

- Mantiene el PWM al 100 % para maximizar la fuerza de empuje.
- Si se detecta corriente excesiva (medida indirectamente o estimada), se reduce el duty cycle a un 70 % para evitar daños en motores y drivers.

■ **EVITAR_LÍNEA:**

- Activa maniobras evasivas si los sensores QTR-8A detectan la franja blanca del borde del dohyo.
- Realiza giros o retrocesos controlados para reingresar al centro del área de combate.

■ **REUBICAR:**

- Si se detecta un ciclo repetitivo de evasión (bucle), ejecuta una maniobra de avance prolongado para alejarse del borde y reposicionarse estratégicamente.

Cada estado puede transicionar a otro en función de condiciones específicas (detección de línea, detección de oponente, tiempos límite, etc.), garantizando una lógica robusta, rápida y adaptativa al entorno cambiante del combate.

VII-B. Subestados de Alineación durante el Ataque

Durante el estado principal **ATACAR**, el robot debe realizar maniobras de ajuste fino para asegurar una correcta alineación hacia el oponente antes de efectuar la embestida definitiva. Para ello, dentro del estado ATACAR se definen varios **subestados de alineación** basados en las lecturas de los sensores de distancia VL53L0X.

Los subestados contemplados son los siguientes:

■ **ALINEAR_FRONT:**

- Si las distancias leídas por los sensores frontales de recho e izquierdo (dF1 y dF2) son aproximadamente iguales, el oponente está centrado frente al robot.
- Si $dF1 < dF2$, el oponente está ligeramente desfasado a la izquierda; se realiza un pequeño giro a la izquierda.
- Si $dF1 > dF2$, el oponente está ligeramente desfasado a la derecha; se realiza un pequeño giro a la derecha.

■ **ALINEAR_LEFT / ALINEAR_RIGHT:**

- Si uno de los sensores laterales detecta al oponente más cerca que los sensores frontales, el robot gira más ampliamente hacia el lado correspondiente (izquierda o derecha) para reposicionar su frente hacia el oponente.

■ **ALINEAR_BACK:**

- Si el sensor trasero detecta un oponente a una distancia significativa, el robot ejecuta un giro de aproximadamente 180° para enfrentar al rival de manera frontal antes de reanudar la estrategia de ataque.

Estos subestados permiten optimizar la precisión del ataque, incrementando la probabilidad de una embestida efectiva y minimizando el riesgo de desalineación durante los combates.

VII-C. Protocolos de Comunicación y Arquitectura de Interconexión

La arquitectura de comunicaciones en la versión v2 mantiene la separación funcional entre protocolos utilizada en la versión inicial, pero optimiza su gestión para simplificar el hardware y asegurar mayor robustez y claridad en la interacción entre módulos.

Los protocolos implementados son los siguientes:

■ **SPI (Serial Peripheral Interface):** El protocolo SPI se utiliza para la comunicación entre el microcontrolador PIC18F4550 (maestro) y el CPLD MAX II (esclavo). A través de este enlace se transmiten consignas de control de motores, tales como valores de PWM, señales de dirección y posibles indicadores de cambio de estado.

■ **I²C (Inter-Integrated Circuit) por Software:** La comunicación con los sensores de distancia VL53L0X se realiza mediante I²C implementado por bit-banging (software) en el PIC18F4550. Esta solución permite asignar pines personalizados para SDA y SCL, facilita la expansión del número de sensores conectados y evita conflictos con periféricos de hardware ocupados por SPI o UART.

■ **UART (Universal Asynchronous Receiver-Transmitter):** Se reserva el módulo UART del PIC18F4550 para la posible transmisión de datos de telemetría hacia un módulo de comunicación inalámbrica futuro. Esta telemetría permitiría, en fases posteriores, monitorear en tiempo real información relevante como el estado de la máquina de estados, valores de distancia detectada y eventos críticos durante el combate.

VII-C0a. Resumen de Asignaciones::

- **SPI:** PIC → CPLD (control de motores)
- **I²C:** PIC → VL53L0X (lectura de distancias)
- **UART:** PIC → Módulo inalámbrico (telemetría, opcional en esta etapa)

Esta distribución optimizada de protocolos garantiza un sistema de comunicación ágil, claro, extensible y especialmente adaptado a las necesidades de la dinámica de combate en robots sumo.

VII-D. Conclusiones de la Lógica de Control y Comunicación v2

La versión v2 del robot sumo introduce una serie de mejoras significativas en la lógica de control y la arquitectura de comunicaciones, en comparación con la versión inicial basada en control PID.

Entre las principales conclusiones se destacan:

■ **Simplificación y robustez del control:** La transición de un esquema PID hacia una máquina de estados finita permite una respuesta mucho más rápida y adecuada a la naturaleza dinámica del combate, evitando complejidades innecesarias de ajuste de parámetros y aumentando la fiabilidad frente a eventos espontáneos.

■ **Optimización de la comunicación entre módulos:** La separación clara de funciones entre los protocolos SPI, I²C (por software) y UART asegura una integración limpia y escalable, reduciendo la posibilidad de conflictos de periféricos y facilitando futuras expansiones, como la incorporación de telemetría inalámbrica.

■ **Mejor gestión de sensores y actuadores:** La asignación estratégica de sensores de distancia VL53L0X y la simplificación en el uso de sensores de línea QTR-8A permiten un sensado efectivo sin sobrecargar el microcontrolador, alineando los recursos del sistema con las prioridades reales del combate.

- Desacoplamiento de generación de PWM:** La migración de la modulación de motores al CPLD MAX II des carga al PIC18F4550 de tareas críticas de temporización, permitiendo concentrar su procesamiento en la lógica de control de alto nivel.
- Mayor independencia y portabilidad:** El cambio a un CPLD propio frente al uso de hardware universitario (Basys2) otorga al proyecto mayor autonomía para pruebas fuera del laboratorio, facilita la iteración rápida y representa una mejora económica significativa.

Con estos cambios, el robot sumo en su versión v2 presenta una arquitectura de control más eficiente, resistente y adaptable, mejor preparada para afrontar los retos dinámicos y espontáneos característicos de las competencias de sumo robótico.

VIII. PLANEACIÓN FINANCIERA V2

En este capítulo se presenta la planeación financiera correspondiente a la versión v2 del robot sumo, considerando únicamente los gastos reales incurridos para su construcción y operación, excluyendo elementos ya disponibles en el inventario del equipo.

VIII-A. Introducción

La versión v2 implicó un rediseño parcial del sistema, lo cual impactó directamente en la planeación y ejecución financiera del proyecto. A diferencia de la versión inicial, en esta etapa se priorizó la adquisición de componentes propios, eliminando la dependencia de hardware prestado (por ejemplo, se reemplazó la FPGA Basys2 por un CPLD MAX II EPM240T100C5), así como la compra de los componentes restantes que no habían sido considerados en la versión inicial. Los costos reflejados a continuación corresponden a compras efectivas, verificadas mediante documentos de adquisición y cotizaciones reales.

VIII-B. Presupuesto Estimado y Gastos Reales v2

La Tabla IV resume todos los materiales adquiridos, junto con su cantidad, precio unitario y costo total.

Cuadro IV
PRESUPUESTO REAL DE COMPONENTES V2

Material	Cant.	Precio Unit. (MXN)	Total (MXN)
Buck DC-DC (12V a 5V)	1	301.35	301.35
Level Shifter 5V-3.3V	1	103.55	103.55
Kit de cables	1	180.00	180.00
Kit de tornillos	1	150.00	150.00
PIC18F4550	1	299.68	299.68
PCB (circuito impreso)	5	50.00	250.00
Driver BTS6970	2	119.40	238.80
Motorreductor 12V 87rpm	2	136.61	273.22
Placa metálica	1	120.00	120.00
MDF	2	40.00	80.00
Sensor láser VL53L0X	5	81.89	409.45
QTR-8A (sensor línea)	2	69.60	139.20
CPLD MAX II (EPM240T100C5)	1	239.99	239.99
Placa desarrollo	1	86.00	86.00
USB-Blaster (programador)	1	125.58	125.58
Eliminador 12V-DC jack	1	80.00	80.00
Soporte motor L	1	139.00	139.00
Batería LiPo 11.1V + cargador	1	629.00	629.00
Caucho líquido RTV	1	49.50	49.50
Total		3894.32	

El gasto total imputable a la v2 asciende a **\$3894.32 MXN**, monto correspondiente únicamente a nuevas adquisiciones.

Es importante señalar que elementos como el módulo ESP32, soportes metálicos adicionales, pintura negra, y pantallas LCD no fueron considerados en este cálculo debido a que ya estaban disponibles en el inventario del equipo, no requiriendo gastos adicionales.

VIII-C. Análisis de Costos y Precio de Venta Aproximado

Partiendo del desglose de materiales presentados en la Tabla IV, el costo directo de construcción funcional del robot sumo v2 asciende a:

$$\text{Costos Directos} = 3894.32 \text{ MXN}$$

Sin embargo, en términos contables y financieros, para calcular el **Costo Final del Producto**, es necesario considerar además los costos indirectos asociados a manufactura, operación, transporte interno, consumibles menores, energía eléctrica y mano de obra no reflejada directamente en la compra de materiales.

De acuerdo con prácticas comunes en proyectos universitarios de ingeniería, se considera una estimación de costos indirectos del 10 % sobre el costo directo:

$$\text{Costos Indirectos Estimados} = 0.10 \times 3894.32 = 389.43 \text{ MXN}$$

Por tanto, el **Costo Final del Producto** resulta ser:

$$\text{Costo Final} = 3894,32 + 389,43 = \boxed{4283,75 \text{ MXN}}$$

Este valor representa el costo real de producción del robot sumo v2, incluyendo tanto los materiales adquiridos como los recursos y esfuerzos asociados a su fabricación.

VIII-D. Estimación del Precio de Venta

En caso de querer comercializar el robot sumo como un producto terminado, sería necesario añadir un margen de utilidad para cubrir riesgos, mejoras futuras y obtener un beneficio económico razonable.

Considerando que se trata de un prototipo funcional, diseñado para competencias de sumo robótico universitario, se sugiere un margen de utilidad aproximado del 50 %.

Así, el precio de venta aproximado sería:

$$\text{Precio de Venta} = 4283,75 \text{ MXN} \times 1,50 = \boxed{6425,63 \text{ MXN}}$$

Por lo tanto, el **robot sumo v2 podría ofrecerse en un rango de 6400 a 6500 MXN**, dependiendo de factores adicionales como demanda, exclusividad, valor agregado en soporte técnico o mejoras implementadas.

VIII-E. Conclusión Financiera Ajustada

La planeación financiera v2 refleja que el equipo logró desarrollar un robot funcional competitivo, con un costo de producción total de aproximadamente **4283.75 MXN**. En una eventual comercialización, considerando un margen de utilidad estándar del 50 %, el precio de venta adecuado rondaría

VIII-F. Clasificación de Inversión y Gastos

En términos contables, se pueden clasificar los elementos en dos categorías principales:

- **Inversión:** Componentes duraderos que podrán reutilizarse en futuras versiones o proyectos:
 - PIC18F4550
 - CPLD MAX II EPM240T100C5
 - USB-BLASTER
 - Motorreductores 12V
 - Drivers BTS6970
 - Sensores VL53L0X
 - QTR-8A
 - Kits de Cables y Tornillería
- **Gastos Operativos:** Elementos que se consumen, deterioran o requieren reemplazo con el tiempo:
 - Batería LiPo y cargador
 - Caucho Líquido RTV
 - Eliminador DC-DC
 - Pintura, soportes adicionales, material de acabado (no contabilizado)

VIII-G. Conclusiones Financieras

La planeación financiera v2 evidencia que, mediante decisiones estratégicas como la adquisición de un CPLD propio, la optimización de materiales y la reutilización de recursos existentes, fue posible desarrollar un robot sumo completamente funcional y competitivo, alcanzando un costo directo de construcción de **\$3894.32 MXN**.

Al considerar también los costos indirectos de operación y manufactura, el **costo final de producción** asciende a aproximadamente **\$4283.75 MXN**. Esta cifra refleja un manejo eficiente de los recursos y una estructura de costos sólida que permitiría, en caso de comercialización, establecer un precio de venta aproximado de **\$6425.63 MXN**.

Además de optimizar el presupuesto frente al escenario inicial planteado en la versión v1, esta estrategia incrementó significativamente la independencia técnica del equipo, eliminando la dependencia de recursos universitarios como la FPGA Basys2, y habilitando un modelo de trabajo más ágil, flexible y sostenible para futuras iteraciones y mejoras.

IX. POSIBLE IMPLEMENTACIÓN MODULAR DE COMUNICACIÓN INALÁMBRICA

Este capítulo explora la posibilidad de integrar un sistema de **telemetría inalámbrica** y/o detección de señales enemigas en el robot sumo de 3kg, con la salvedad de que dicho sistema debe ser completamente **modular**. Esto significa que el módulo y la antena, fabricados por el equipo o mediante un sistema de detección propio, se utilizarán exclusivamente para pruebas y recolección de datos, y deberán poder ser retirados sin afectar el funcionamiento normal del robot durante el combate oficial, cumpliendo estrictamente con las reglas de la competencia.

IX-A. Objetivos del Sistema Inalámbrico Modular

El objetivo principal de este módulo es:

- Permitir el monitoreo en tiempo real de variables críticas (por ejemplo, velocidad, consumo, estado de sensores) durante la fase de pruebas.
- Recolectar datos que faciliten la optimización y ajuste de la estrategia de control, sin interferir en la operación autónoma durante la competencia.
- Facilitar la detección de señales enemigas en entornos de prueba, posibilitando el desarrollo de estrategias adicionales que en el futuro podrían incluir análisis del entorno.

IX-B. Tecnologías Inalámbricas Consideradas y Modalidad Modular

Se han evaluado diversas tecnologías para la implementación del módulo inalámbrico:

- **Wi-Fi:** Permite transmisiones de datos a alta velocidad. Se prevé el uso de módulos como el ESP8266 o ESP32.
- **Bluetooth:** Ofrece comunicación sencilla y de bajo consumo, adecuada para distancias cortas.
- **XBee (ZigBee):** Confiable y de bajo consumo, utilizado en aplicaciones de monitoreo.

- **LoRa:** Si bien su tasa de datos es baja, puede ser útil para pruebas a largas distancias.

Es importante destacar que el **sistema inalámbrico será modular y extraíble**. Durante las pruebas y la recolección de datos, el módulo se conectará al PIC18F4550, mediante interfaces como UART o SPI, para transmitir información de diagnóstico. Sin embargo, conforme se acerque la fase de competencia oficial, el módulo se desconectará o retirará para cumplir con la regla que prohíbe el uso de comunicación inalámbrica durante el combate.

IX-C. Implementación Preliminar y Estrategia de Integración

La estrategia de integración del módulo inalámbrico se basa en:

- **Fabricación Propia de la Antena:** El equipo desarrollará una antena específica que se ajuste al diseño modular del robot. Esta antena se instalará de forma que pueda ser conectada o desconectada fácilmente.
- **Interfaz Modular:** Se utilizará una interfaz estándar (por ejemplo, un conector tipo JST o similar) que permita la conexión/desconexión rápida del módulo inalámbrico sin requerir modificaciones en el circuito principal.
- **Pruebas en Entornos Controlados:** Durante la fase de pruebas, el módulo se activará para recolectar datos en tiempo real, como señales PWM, lecturas de sensores y estado de la batería, sin que ello afecte la lógica de control del robot.
- **Aislamiento durante el Combate:** Durante el enfrentamiento oficial, el módulo se mantendrá desconectado para garantizar que la comunicación RF no interfiera con el funcionamiento normal del robot, en cumplimiento de las reglas de la competencia.

IX-D. Estrategias de Telemetría y Recolección de Datos

El sistema inalámbrico modular permitirá implementar estrategias de telemetría que incluyen:

- **Monitorización en Tiempo Real:** Transmisión de datos críticos (velocidad, consumo, estado de sensores) a una interfaz de usuario en una computadora o dispositivo móvil.
- **Registro y Análisis de Datos:** Almacenamiento de la información recolectada para el análisis posterior, facilitando ajustes en la lógica de control, como la calibración de los parámetros del controlador PID.
- **Detección de Señales Ambientales:** Pruebas de un sistema de detección que, en futuras iteraciones, pueda identificar señales provenientes del entorno o incluso del robot rival, sin que este sistema permanezca activo durante el combate.

IX-E. Conclusiones

La implementación modular de comunicación inalámbrica se plantea como una herramienta esencial para el desarrollo y la optimización del robot sumo, permitiendo:

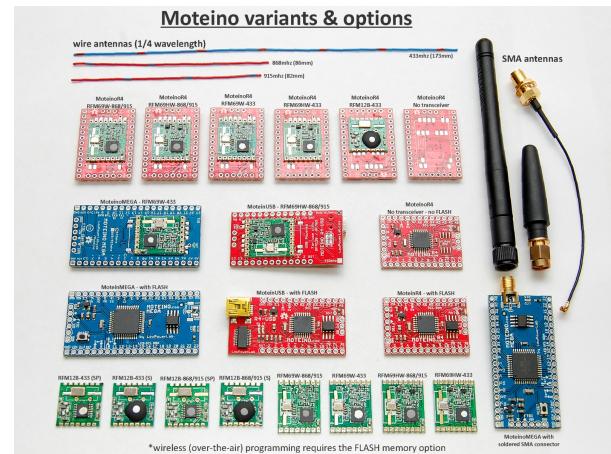


Figura 8. Diagrama preliminar del sistema modular de comunicación inalámbrica, mostrando el módulo extraíble y su conexión al PIC18F4550.



Figura 9. Ejemplo de interfaz de telemetría para la recolección de datos durante pruebas, sin interferir en el funcionamiento del robot.

- Realizar pruebas y recolectar datos críticos sin afectar el desempeño en combate.
- Facilitar la depuración y el ajuste de la lógica de control mediante la monitorización en tiempo real.
- Garantizar que, en la competencia oficial, el módulo pueda ser retirado sin interrumpir el funcionamiento autónomo del robot, cumpliendo con las reglas que prohíben el uso de comunicación RF durante el combate.

Esta estrategia modular asegura que el sistema de telemetría se utilice únicamente en fases de prueba y recolección de datos, proporcionando al equipo información valiosa para futuras mejoras, mientras que en el entorno de combate se mantiene el rendimiento óptimo del robot.

X. INTEGRACIÓN DEL BOOTLOADER Y AVANCES EN PROGRAMACIÓN

Como parte de los avances del proyecto, se ha integrado un sistema de **bootloader** en el PIC18F4550. Inicialmente se utilizó el **MikroBootloader** (disponible en [MikroBootloader](#)), adaptado para operar con un cristal externo de 20 MHz, lo cual

mejora la precisión de los temporizadores y la comunicación SPI con el CPLD.

X-A. Objetivos y Ventajas del Bootloader

La incorporación del bootloader aporta múltiples beneficios:

- **Actualización Rápida:** Reprograma el microcontrolador vía USB sin herramientas de programación externas.
- **Facilidad de Depuración:** Agiliza la carga de nuevas versiones de firmware durante las pruebas.
- **Seguridad y Robustez:** Incorpora verificación por CRC y mecanismos de recuperación en caso de fallo.
- **Adaptabilidad:** Facilita la implementación de nuevos modos de control sin intervención física.

X-B. Implementación y Estrategia de Integración

La integración se realizó de la siguiente manera:

- Se generó un archivo .hex que combina el bootloader con la aplicación, configurado para el oscilador de 20 MHz.
- Utilizando un Pickit 2, se cargó el bootloader en el PIC18F4550 por primera vez.
- Se diseñó un **development board** propio (Coquirana 1) que agrupa pines USB, MCLR, VDD, VSS y señal de activación del bootloader.

X-C. Adopción del Bootloader de Unit Electronics

Tras validar el funcionamiento con MikroBootloader, se cambió al bootloader recomendado por Unit Electronics, descrito en:

[Cargar Bootloader USB al PIC18F4550 con MPLAB IDE](#)

Aunque la placa Coquirana 1 difiere de la de Unit Electronics, adaptamos el procedimiento modificando:

- Asignaciones de pines en config.h para MCLR, VDD, VSS y USB D+/D-.
- Parámetros del oscilador externo de 20 MHz en las directivas de configuración.
- Tamaño y ubicación de la sección de memoria del bootloader, ajustándolos al layout de Coquirana 1.

Estas modificaciones garantizan que el bootloader de Unit Electronics funcione de forma estable en nuestra placa, ofreciendo menor latencia de arranque y compatibilidad total con MPLAB IDE.

X-D. Estrategias de Programación y Control Avanzado

La integración final permite:

- Actualizaciones de firmware en campo sin herramientas JTAG.
- Afinación de parámetros PID y estrategias de combate al vuelo.
- Recuperación automática en caso de corrupción de firmware.

X-E. Relevancia en el Contexto del Proyecto

La adopción del bootloader de Unit Electronics consolidó un flujo de desarrollo ágil y seguro, alineado con las necesidades de actualización continua de Coquineitor.

XI. ARQUITECTURA DE COMUNICACIÓN MAESTRO–ESCLAVO Y EVOLUCIÓN DE SPI

En este capítulo se documenta el diseño, evolución y optimización de la comunicación SPI entre el microcontrolador **PIC18F4550** (maestro) y el **CPLD MAX II EPM240T100C5** (esclavo). Se describen las configuraciones iniciales, los problemas detectados en pruebas tempranas y la implementación final de un protocolo robusto que supera las limitaciones detectadas en fases previas.

XI-A. Arquitectura Inicial de Comunicación SPI

Modelo Maestro–Esclavo:

- **Maestro (PIC18F4550):** Encargado de la lectura de sensores, toma de decisiones y envío de consignas de control motor mediante SPI.
- **Esclavo (Basys2 inicialmente / CPLD MAX II posteriormente):** Recibe las consignas y genera señales PWM de alta resolución para los motores.

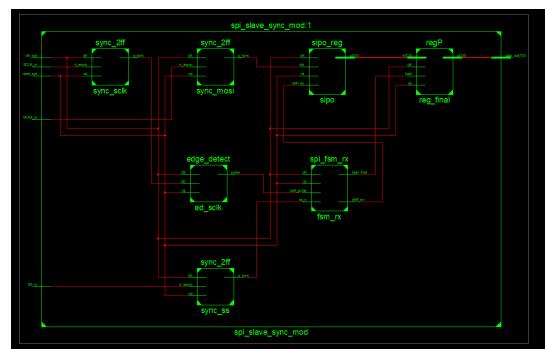


Figura 10. Arquitectura inicial de comunicación SPI en la versión preliminar (Basys2).

La comunicación consistía en transmitir un **byte de control** seguido de un **byte de duty cycle**, desplazados a través de un registro serie-paralelo (shift register) en el FPGA.

XI-B. Configuración del SPI en el PIC18F4550

El PIC18F4550 integra un módulo MSSP (Master Synchronous Serial Port) que soporta SPI nativamente. La configuración de los registros relevantes fue:

- **SSPCON1:** Configuración del modo maestro y frecuencia deseada.
- **SSPSTAT:** Control de fase y polaridad del reloj (modos SPI 0–3).
- **SSPBUF:** Buffer de datos para transmisión/recepción serial.

El SPI se configuró en modo maestro, frecuencia $F_{osc}/64$ y modo de operación CKP=1, CKE=1, para maximizar compatibilidad con el CPLD y reducir problemas de fase.

XI-C. Primer Intento de Comunicación SPI: Problemas Detectados

Durante las primeras pruebas (con Basys2 como esclavo), se presentaron los siguientes conflictos:

- **Cross Domain Clocking:** SCLK, MOSI y SS llegaban asincrónicos al reloj interno de la FPGA, generando posibles violaciones de *setup/hold* [20].
- **Skew:** Diferencias de tiempo entre las líneas provocaban que datos válidos fueran capturados fuera de fase.
- **Metastabilidad:** La falta de sincronización adecuada causaba resultados indeterminados en los registros de captura [?].
- **Latch con noveno pulso faltante:** El PIC no emitía un pulso adicional de SCLK al finalizar la transmisión, complicando el guardado de datos en la FPGA.

Estos fenómenos fueron investigados y documentados ampliamente en la literatura técnica [?], [?], [?].

XI-D. Primer Diseño de SPI: Latch Asíncrono

Inicialmente, se intentó capturar el dato final usando un latch accionado directamente con el flanco de subida de SS (Slave Select). Esta aproximación parecía viable en simulaciones, pero fallaba en hardware real debido a la sensibilidad del esclavo a pequeñas fluctuaciones en las líneas SS y SCLK.

XI-E. Diagnóstico: Limitaciones del Place and Route en Basys2

La Basys2 (FPGA Spartan 3E) no posee una red global de relojes (GCLK) tan flexible como otros dispositivos más avanzados o CPLDs como el MAX II. Esto implicaba que:

- Las señales externas no podían ser promovidas automáticamente a clock nets globales.
- La distribución interna del reloj podía sufrir skew significativo.

Por tanto, resultaba riesgoso usar directamente SCLK o SS como relojes en el diseño, ya que la propagación de señales asíncronas podría provocar errores de temporización, metastabilidad o capturas incorrectas de datos.

Reflexión sobre el cambio de dispositivo:

Aunque en teoría el **CPLD MAX II** podría haber tolerado el esquema de SPI inicial (debido a su red interna de distribución de reloj más robusta y su mayor tolerancia al cruce de dominios de reloj), se decidió mantener la versión de **SPI robusto**.

Esta decisión obedeció a dos motivos fundamentales:

1. **Conveniencia técnica:** Ya existía una implementación completamente funcional, validada en simulaciones ISim y pruebas prácticas sobre la Basys2, basada en la estrategia de doble sincronización y FSM interna.
2. **Valor del aprendizaje:** El diseño robusto no fue adoptado accidentalmente, sino como resultado de una investigación profunda sobre problemas reales de comunicación digital — incluyendo metastabilidad, skew y sincronización de señales — apoyada en fuentes técnicas reconocidas como Intel, Xilinx y VHDLWhiz [?], [?], [?], [?].

Preservar este diseño no solo garantizaba una comunicación sólida en el CPLD MAX II, sino que además reflejaba un compromiso con las buenas prácticas de ingeniería y con el

conocimiento adquirido a través de la resolución de problemas complejos.

XI-F. Segunda Versión: SPI Robusto Basado en Reloj Interno

Siguiendo mejores prácticas recomendadas por Intel, Xilinx y autores como [20], [?], [?], se rediseñó la arquitectura de la siguiente manera:

- **Doble sincronización** (doble flip-flop) de SCLK, MOSI y SS a un reloj interno del CPLD (clk_{sys} , 50 MHz).
- **Reconstrucción de flancos de SCLK** dentro del dominio seguro de clk_{sys} .
- **Máquina de Estados Finita (FSM)** que interpreta eventos SPI en el dominio interno, garantizando capturas fiables.

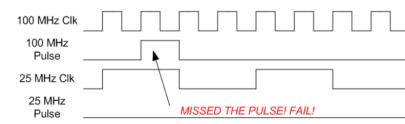


Figura 11. Ejemplo de problemas de metastabilidad al recibir señales asíncronas [?].

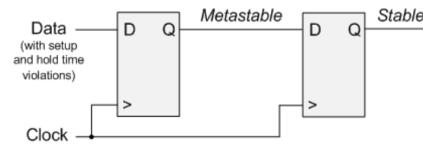


Figura 12. Solución clásica mediante doble sincronización de señales de entrada [20].

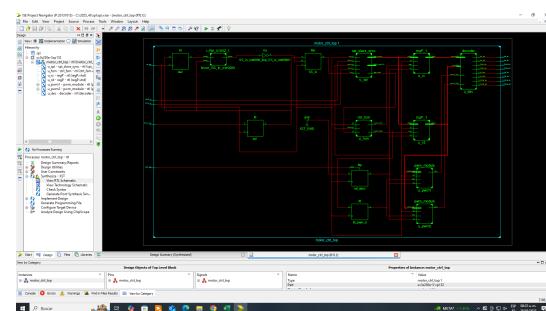


Figura 13. Arquitectura final de comunicación SPI robusta basada en reloj interno.

XI-G. Comparativa entre SPI Inicial y SPI Robusto

Diferencias clave:

- El diseño inicial usaba directamente los flancos externos (SCLK, SS), lo que resultaba inestable y sensible a variaciones.
- El diseño robusto sincroniza todas las señales a un reloj interno seguro, eliminando riesgos de metastabilidad, skew y errores de captura.
- No se requiere un noveno pulso de SCLK, ya que el latch de los datos ocurre controladamente cuando SS pasa de '0' a '1' en el dominio de clk_{sys} .

XI-H. Validación Experimental

El nuevo diseño fue validado transmitiendo dos bytes de datos cada 500 ms desde el PIC, y visualizando su captura mediante LEDs conectados al CPLD.

Cada byte fue enviado por separado, latcheado exitosamente y mostrado de forma estable en el conjunto de LEDs.

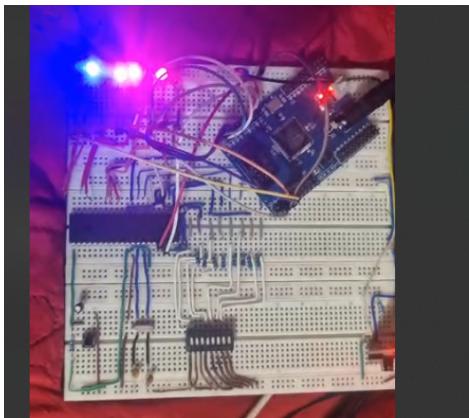


Figura 14. Visualización del primer byte latcheado (byte de control) en los LEDs.

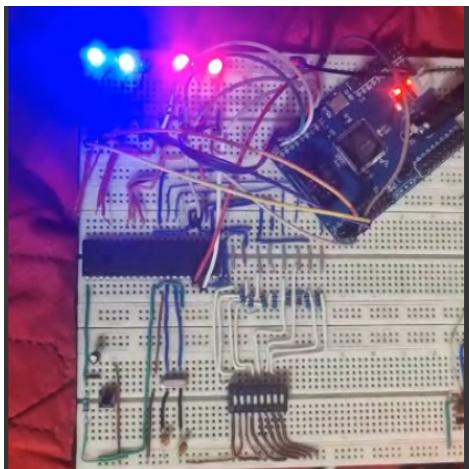


Figura 15. Visualización del segundo byte latcheado (byte de duty cycle) en los LEDs.

Las frecuencias prácticas alcanzadas en la transmisión fueron:

$$f_{SPI} \approx 250 \text{ kHz} \Rightarrow t_{bit} \approx 4 \mu\text{s}$$

Muy por debajo de los límites de latencia de 10 ns característicos de los flip-flops internos del CPLD, garantizando márgenes de seguridad elevados.

XI-I. Conclusión del Capítulo

La migración de una arquitectura SPI básica a un sistema robusto basado en reloj interno y sincronización explícita permitió superar los problemas inherentes a señales asíncronas, garantizando una comunicación fiable incluso en condiciones adversas de operación.

Estos principios robustos sientan las bases para la lógica de control de motores, estrategias de combate y adquisición sensorial que se desarrollarán en los siguientes capítulos del proyecto.

XII. CODIFICACIÓN DE DATOS DELEGADOS AL CPLD

Recordando de capítulos anteriores que:

SPI (Serial Peripheral Interface): El protocolo SPI se utiliza para la comunicación entre el microcontrolador PIC18F4550 (maestro) y el CPLD MAX II (esclavo). A través de este enlace se transmiten consignas de control de motores, tales como valores de PWM, señales de dirección y posibles indicadores de cambio de estado.

Ahora, en este capítulo, se describe en detalle la lógica y codificación implementada para el envío de dicha información, partiendo de la evolución de diversas propuestas hasta llegar a la solución final robusta.

XII-A. Contexto General de Comunicación

La comunicación SPI establece que el maestro (PIC18F4550) transmite datos hacia el esclavo (BASYS2 inicialmente, luego CPLD MAX II). Es importante destacar que el esclavo únicamente recibe datos, no existe retroalimentación de datos hacia el maestro.

El flujo general considera:

- El PIC envía información mediante MOSI, sincronizada por SCLK.
- El CPLD capta los datos mediante un registro serie-paralelo interno.
- Los datos capturados son interpretados por una máquina de estados que decide la dirección y el destino de cada motor, así como el duty cycle de la señal PWM.

Los módulos internos relevantes en la lógica del CPLD son:

- **PWM1:** Genera señales PWM controladas para motor 1.
- **PWM2:** Genera señales PWM controladas para motor 2.
- **CONFIG:** Configura direcciones y destinos de las señales PWM en función de los datos recibidos.

XII-B. Primera Propuesta de Codificación de Datos

La propuesta inicial consistía en utilizar los dos bits menos significativos de un primer byte para determinar la dirección y destino de un solo motor:

- **b(1):** Destino (PWM1 o PWM2).
- **b(0):** Dirección (horario o antihorario).

La codificación resultante era:

- Sentido horario motor 1 usando PWM1: 00.
- Sentido antihorario motor 1 usando PWM1: 01.
- Sentido horario motor 1 usando PWM2: 10.
- Sentido antihorario motor 1 usando PWM2: 11.

El segundo byte completo enviaba el ancho de pulso (duty cycle) para el motor correspondiente.

XII-B0a. Limitaciones Detectadas: Aunque la propuesta funcionaba para controlar un solo motor a la vez, surgieron varias limitaciones:

- Solo un motor podía ser gestionado en cada transmisión SPI.
- Existía un retardo microsegundos entre los comandos enviados a motor 1 y motor 2.

XII-B0b. Hipótesis de Problema: Efecto tipo "Freno ABS": Se planteó la hipótesis de que alternar demasiado rápido entre motores podía inducir un efecto tipo ABS (Anti-lock Braking System), donde:

- El motor recibía pulsos interrumpidos de energía.
- Se generaban vibraciones, pérdida de tracción o trayectorias irregulares.

Este fenómeno podía agravarse si la frecuencia de muestreo SPI no estaba sincronizada con los ciclos de PWM generados por el CPLD.

XII-B0c. Validación Técnica: La hipótesis fue respaldada considerando:

- El retardo relativo entre flancos SPI y períodos PWM.
- El tiempo de switching del driver BTS7960.
- La inercia del robot (masa total).

Aunque a simple vista no se percibía, en dinámicas reales de combate podía ser crítico.

XII-C. Segunda Propuesta: Control Simultáneo de Ambos Motores

Para solucionar la limitación de control individual, se propuso una extensión en la codificación, utilizando los primeros 4 bits del primer byte para controlar simultáneamente ambos motores:

- **b(0):** Dirección motor 1.
- **b(1):** PWM destino motor 1.
- **b(2):** Dirección motor 2.
- **b(3):** PWM destino motor 2.

El segundo byte seguía dedicado al duty cycle, aplicado a ambos motores en conjunto.

XII-D. Problema Detectado en la Segunda Propuesta: Ambigüedad

Aunque ahora se transmitía simultáneamente información de ambos motores, surgió una nueva ambigüedad:

- No se especificaba explícitamente si debían activarse ambos motores o solo uno.

Por ejemplo, un byte 0000 indicaba motor 1 sentido horario usando PWM1 y motor 2 sentido horario usando PWM1, pero esto siempre dejaba una configuración fija para ambos motores, eliminando la posibilidad de utilizar solo uno de ellos mientras el otro permanece apagado.

XII-E. Solución Final: Introducción de Campo de Activación ACT_CTRL

Para resolver esta ambigüedad, se reformuló completamente la codificación:

- **b7–b4:** *ACT_CTRL[3 : 0]* - Campo de activación de motores.

- **b3–b0:** Dirección y destino de motor 1 y motor 2.

Separando explícitamente la activación del tipo de movimiento.

XII-E0a. Estructura Final del Byte SPI::

- Bits 0 y 1: Dirección y destino motor 1.
 - Bits 2 y 3: Dirección y destino motor 2.
 - Bits 4 y 5: Activación:
 - 00: Activar sólo motor 1.
 - 01: Activar sólo motor 2.
 - 10: Activar ambos motores.
 - 11: Reservado para test/futuro.
 - Bits 6 y 7: Actualmente reservados (potencial control extendido).
- El segundo byte SPI contiene:
- Bits 7–4: Duty cycle motor 1 (0–15, pasos de 6.25 %).
 - Bits 3–0: Duty cycle motor 2 (0–15, pasos de 6.25 %).

XII-F. Análisis Final

Esta estructura ofrece:

- Control independiente y simultáneo de ambos motores.
- Eliminación total de ambigüedades de activación.
- Flexibilidad de futuras expansiones usando bits reservados.
- Resolución suficiente para maniobras de alta precisión en el dohyo.

XII-G. Conclusión del Capítulo

La evolución en la codificación de datos SPI refleja el proceso de maduración técnica del proyecto, pasando de una implementación básica hacia un sistema sólido, fiable y extendible.

La solución final garantiza:

- Robustez operacional bajo condiciones dinámicas de combate.
- Mayor seguridad de activación sin errores lógicos.
- Base escalable para implementar futuras estrategias avanzadas.

Esta codificación establece las bases para la lógica interna del CPLD en la interpretación y ejecución de consignas de movimiento.

XIII. DISEÑO MODULAR (EN VHDL) DEL CPLD PARA CONTROL DE MOTORES

Este capítulo presenta la estructura modular del sistema de control implementado en el **CPLD MAX II**, encargado de interpretar las órdenes enviadas por el **PIC18F4550** mediante comunicación **SPI**.

Como se explicó en la Sección XII, la captura y codificación de datos SPI ya fue descrita previamente, por lo que en este capítulo se abordará específicamente la decodificación, el control de ruta y la generación de PWM, considerando que el diseño fue planteado de forma modular para facilitar su comprensión, mantenimiento y ampliaciones futuras.

El objetivo es analizar cada módulo principal en secuencia lógica:

1. Módulo de Decodificación de Datos.
2. Módulo de Configuración o Control de Ruta.
3. Módulos PWM1 y PWM2.

Cada módulo fue diseñado para funcionar de manera autónoma y específica, permitiendo un diagnóstico rápido en caso de errores y una eventual personalización independiente.

XIII-A. Módulo de Captura SPI

Como se detalló en el Capítulo XI, el sistema SPI utiliza sincronización robusta mediante doble flip-flop a un reloj interno (50 MHz) del CPLD para asegurar la captura libre de metastabilidad.

El dato SPI capturado es almacenado en registros intermedios, listo para ser procesado por la siguiente etapa.

XIII-B. Módulo de Decodificación

El **Módulo de Decodificación** separa los bits recibidos por SPI conforme a la estructura establecida en el Capítulo XII.

XIII-B0a. Asignación de Bits::

- **Byte 1 (Control de Motores):**
 - **Bits 7–4:** Campo ACT_CTRL[3 : 0] para activación de motores.
 - **Bits 3–2:** Dirección y destino del Motor 2.
 - **Bits 1–0:** Dirección y destino del Motor 1.
- **Byte 2 (Duty Cycle):**
 - **Bits 7–4:** Duty cycle para Motor 1.
 - **Bits 3–0:** Duty cycle para Motor 2.

Cada segmento de bits es almacenado en registros internos que serán utilizados por los módulos posteriores.

XIII-B0b. Funcionamiento:: - El módulo detecta la finalización de la transmisión SPI. - Extrae los campos de control y duty cycle. - Rutea cada información al módulo de control de ruta y a los generadores PWM correspondientes.

XIII-C. Módulo de Configuración o Control de Ruta

El **Módulo de Control de Ruta** interpreta los datos decodificados para establecer:

- Qué motor o motores deben activarse.
- Cuál será la dirección de giro de cada motor (horario o antihorario).
- A cuál PWM (PWM1 o PWM2) se asignará cada motor.

Este módulo separa la información de control (campo ACT_CTRL) de la información de movimiento (dirección y destino), preparando las señales adecuadas que se enviarán como entradas a los módulos PWM.

XIII-C0a. Relación con el Driver BTS7960:: Las decisiones de este módulo están planeadas para cumplir directamente con el esquema de control del driver **BTS7960**:

- **RPWM** y **LPWM**: PWM moduladas en ancho de pulso según sentido de giro. - **ENR** y **ENL**: Enables estáticos que activan el puente interno en la dirección deseada.

De esta forma, la ruta completa de señales queda clara desde el SPI hasta el driver.

XIII-D. Módulos PWM1 y PWM2

Los módulos **PWM1** y **PWM2** son responsables de generar las señales PWM de alta resolución que alimentarán los puentes H de los motores.

Cada módulo PWM recibe exactamente **tres entradas**:

- **Dirección (Right/Left):** Define si el motor gira en sentido horario o antihorario.
- **Destino (PWM1 o PWM2):** Selecciona cuál señal PWM es usada para cada motor.
- **Duty Cycle (4 bits):** Ancho de pulso deseado (0–100 % con pasos de 6.25 %).

Y produce exactamente **cuatro salidas**:

- **RPWM:** Señal PWM para giro horario.
- **LPWM:** Señal PWM para giro antihorario.
- **ENR:** Enable derecho (habilita conducción hacia adelante).
- **ENL:** Enable izquierdo (habilita conducción hacia reversa).

XIII-D0a. Funcionamiento Interno:: - Cada módulo contiene un **Contador** que incrementa con cada ciclo de reloj (50 MHz). - Un **Comparador** compara el valor actual del contador con el duty cycle deseado:

- Si contador <duty cycle → salida PWM en HIGH.
 - Si contador duty cycle → salida PWM en LOW.
- La señal PWM es dirigida a RPWM o LPWM dependiendo de la dirección recibida. - El ENR o ENL correspondiente se mantiene activado durante todo el tiempo para permitir el flujo de corriente adecuado al motor.

XIII-D0b. Planeación para el Driver BTS7960:: Esta lógica asegura una compatibilidad perfecta con el BTS7960, optimizando:

- Conducción eficiente de corriente.
- Respuesta rápida a cambios de dirección.
- Eliminación de estados de flotación o bloqueo.

XIII-E. Conclusión del Capítulo

El diseño modular del CPLD permite:

- Dividir claramente la captura, interpretación y ejecución de las consignas recibidas por SPI.
- Diagnosticar errores localizados sin afectar otros módulos.
- Facilitar expansiones futuras (por ejemplo, nuevos modos de control).

La estructura establecida sienta una base sólida para el control eficiente y robusto de motores dentro del entorno dinámico de competencia de robots sumo.

XIV. MÁQUINAS DE ESTADOS EN EL SISTEMA MAESTRO–ESCLAVO

Este capítulo describe las Máquinas de Estados Finitas (FSMs) implementadas tanto en el **CPLD MAX II** como en el **PIC18F4550**, las cuales permiten la correcta sincronización y ejecución de las consignas enviadas y recibidas por el sistema maestro–esclavo. Se documentan de manera modular para facilitar su comprensión, depuración y expansión futura.

XIV-A. 13.1 Diseño Modular de FSMs

Siguiendo el principio planteado anteriormente:

“Para un mejor entendimiento, gestión e identificación de errores, o para personalizar solo un área sin afectar las demás, el diseño se planteó de forma modular.”

Cada FSM se diseñó como un bloque funcional independiente:

- Máquina de Estados del **CPLD** para captura y procesamiento de bytes SPI.
- Máquina de Estados del **PIC18F4550** para control estratégico y dinámico en combate.

Cada una será abordada a continuación.

XIV-B. 13.2 Máquina de Estados del CPLD: Sincronización de Bytes SPI

Como ya se explicó detalladamente en la [Sección XI](#) (Arquitectura de Comunicación Maestro-Eslavo y Evolución de SPI), el CPLD recibe los datos SPI de manera sincronizada gracias a una FSM que asegura la captura robusta de ambos bytes enviados por el PIC.

Aquí solo se resume su estructura principal, omitiendo repeticiones innecesarias.

XIV-B1. 13.2.1 Estados de la FSM de Captura SPI: La FSM implementada en el CPLD sigue la siguiente secuencia lógica basada en el estado de la señal SS (Slave Select):

- **ESPERA:** (SS = 1) El sistema espera la bajada de SS para iniciar la captura.
- **RECIBIENDO:** (SS = 0) Se captura el primer byte de datos (byte de control).
- **LEIDO:** (SS = 1) El primer byte ha sido completamente leído; el sistema pasa a esperar el segundo byte.
- **ESPERA2:** (SS = 1) Se mantiene en espera hasta detectar un nuevo inicio de transmisión.
- **RECIBIENDO2:** (SS = 0) Se captura el segundo byte de datos (byte de duty cycle).
- **LEIDO2:** (SS = 1) El segundo byte ha sido leído; se activan las cargas de los módulos PWM.

XIV-B2. 13.2.2 Salidas de Control Asociadas:

- **LOAD1:** Se activa cuando se debe cargar el valor de duty cycle al PWM1.
- **LOAD2:** Se activa cuando se debe cargar el valor de duty cycle al PWM2.

Estas señales dependen de la evaluación del bit de destino contenido en el primer byte SPI:

$$\text{LOAD1} = 1 \text{ cuando } \text{LOADPWM} = 1 \text{ y } Q(1) = 0$$

$$\text{LOAD2} = 1 \text{ cuando } \text{LOADPWM} = 1 \text{ y } Q(1) = 1$$

Donde $Q(1)$ es el bit correspondiente a la selección de PWM destino según la decodificación del byte de control, de acuerdo a lo establecido en el capítulo [Sección XII](#).

XIV-C. 13.3 Máquina de Estados del PIC18F4550: Control Estratégico en Combate

El PIC18F4550, además de ser maestro SPI, actúa como el **cerebro lógico** del robot, manejando sensores y decisiones estratégicas mediante una FSM jerárquica.

XIV-C1. 13.3.1 Visión General de la FSM en el PIC: La FSM del PIC consta de tres grandes grupos de estados:

- **Grupo 1: Inicialización y Arranque.**
- **Grupo 2: Calibración Única (opcional).**
- **Grupo 3: Combate Autónomo.**

Cada transición depende de eventos de sensores, botones físicos o resultados de calibraciones previas.

XIV-C2. 13.3.2 Grupo 1: Inicialización y Arranque:

XIV-C2a. ESPERA_INICIO: Estado tras el encendido:

- Configura MSSP (SPI), ADC, y entradas de sensores.
- Inicia rutina bit-bang I²C.
- LED rojo encendido como indicación.

XIV-C2b. Evento de salida::

- Si botón RD2 presionado: ir a INICIO_CALIB.
- Si no: ir a BUSCAR.

XIV-C2c. INICIO_CALIB: Preparar buffers de calibración y transición directa a CAL_LINEA.

XIV-C3. 13.3.3 Grupo 2: Calibración Única:

XIV-C3a. CAL_LINEA: Toma de 80 muestras de los sensores QTR-8A:

- Cálculo de mínimos, máximos y umbrales de detección.
- Guarda umbrales en EEPROM.

Finaliza y transita a BUSCAR.

XIV-C4. 13.3.4 Grupo 3: Combate Autónomo:

XIV-C4a. BUSCAR:

- Búsqueda del oponente mediante sensores VL53L0X.
- Giro en espiral con ±10 % de duty.

XIV-C4b. Transiciones::

- A ATACAR al detectar oponente cerca.
- A EVITAR_LÍNEA si se detecta línea.

XIV-C4c. ATACAR:

- Embestida con duty-cycle creciente cada 50 ms.
- Envío de consignas SPI con ACT_CTRL='11'.

XIV-C4d. Transiciones::

- A EVITAR_LÍNEA si se detecta línea.
- A REUBICAR si distancia al rival es mayor a 60 cm.

XIV-C4e. EVITAR_LÍNEA:

- Reducción del duty en motor correspondiente.
- Giro de corrección hacia el interior del dohyo.

XIV-C4f. Transición::

- A BUSCAR tras 5 lecturas sin detectar línea.

XIV-C4g. REUBICAR:

- Cálculo de vector de corrección basado en sensores de distancia.
- Maniobra de realineación controlada por diferencia de distancias.

XIV-C4h. Transiciones::

- A ATACAR si oponente se encuentra a menos de 40 cm.
- A BUSCAR si no hay detección tras 2 segundos.

XV. ENCODERS DE RATÓN Y CÁLCULO DE DESPLAZAMIENTOS

En este capítulo se describe la implementación de encoders basados en sensores ópticos de ratón reutilizados, utilizados para estimar desplazamientos lineales y giros del robot.

XV-A. Marco Teórico

Un encoder de cuadratura genera dos señales (A y B) desfasadas 90° entre sí. Analizando las transiciones de estas señales se puede:

- Determinar la dirección del movimiento (sentido horario o antihorario).
- Contabilizar los pulsos para medir distancia recorrida.

Cada transición entre estados de A y B representa un avance (o retroceso) de una fracción del perímetro de la rueda.

XV-B. Cálculos Geométricos

Los parámetros utilizados fueron:

- Pulsos por vuelta de rueda: 72 pulsos.
- Diámetro de rueda: 59 mm.
- Track width (distancia entre ruedas): 165 mm.

Por tanto:

$$CIRC_{MM} = \pi \times 59 \approx 185,35 \text{ mm}$$

$$MM_POR_PULSO = \frac{185,35}{72} \approx 2,574 \text{ mm/pulso}$$

Para el cálculo de grados de giro diferencial:

$$D_POR_PULSO_D = \frac{360}{T_MM \times \frac{1}{CIRC_{MM}}} \approx 5,62^\circ/\text{pulso d}$$

XV-C. Implementación en Librería Encoders

Para gestionar los encoders se implementó la siguiente librería en C:

XV-C1. Archivo *encoders_mouse.h*:

- Definición de pines de entrada (A y B de cada encoder).
- Prototipos de funciones de inicialización, actualización, lectura de cuentas y cálculo de desplazamiento.

XV-C2. Archivo *encoders_mouse.c*: Funciones principales:

- *enc_init()*: Configura los pines como entradas y captura el estado inicial.
- *enc_update()*: Lee transiciones de cuadratura usando una tabla LUT y actualiza contadores de pulsos.
- *enc_cuenta_R()*, *enc_cuenta_L()*: Devuelven la cuenta actual de cada rueda.
- *enc_mm_rueda_R()*, *enc_mm_rueda_L()*: Calculan desplazamientos en milímetros.
- *enc_grados_giro()*: Calcula la rotación acumulada del robot en grados.

XV-C3. Optimización: El uso de una tabla LUT de 16 entradas (*gTbl[16]*) permite determinar el incremento (+1, -1 o 0) en cada transición de A y B con mínimo cómputo, asegurando operación en tiempo real incluso a 1 kHz de actualización.

XV-D. Conclusión

La implementación de encoders ópticos adaptados de ratón permitió dotar al robot de un sistema confiable y de bajo costo para la medición de desplazamientos lineales y rotacionales, fundamental para estrategias avanzadas de navegación y control de posición.

XV-E. Relevancia para la Estrategia de Combate

La disponibilidad de umbrales precisos permite que, durante *EVITAR_LÍNEA*, el robot detecte con anticipación el borde del dohyo y aplique maniobras de corrección suaves, evitando salidas accidentales sin sacrificar velocidad de embestida.

XVI. RESULTADOS DE PRUEBAS FUNCIONALES E INTEGRACIÓN

Este apartado documenta las mediciones realizadas una vez ensamblado el *motor_ctrl_top* con el **PIC18F4550** y el **CPLD MAX II**. Todas las señales se capturaron con un analizador lógico Saleae Logic 16 a 100 MS/s y un osciloscopio Tektronix TDS2024C.

XVI-A. Rutina de Self-Test al Encendido

1. Encendido \Rightarrow LED rojo fijo. El CPLD envía 0xAA – 0x55 al PIC para verificar la línea MISO.
2. El PIC responde con 0xF0 – 0x0F; el CPLD conmuta el LED verde.
3. Si la secuencia es correcta se habilitan los PWM; en caso contrario el sistema entra en *failsafe* (LED rojo intermitente).

XVI-B. Caracterización de la Señal PWM

Cuadro V
DUTY-CYCLE SOLICITADO VS. MEDIDO (@ 19.53)

Código (hex)	Duty % teórico	Duty % medido
0x00	0	0.2
0x08	50	49.8
0x0F	100	99.7

La desviación máxima (<0,3%) confirma la exactitud del *pwm_module* con registro *shadow*.

XVII. DISEÑO 3D Y PREPARACIÓN PARA MANUFACTURA

El diseño físico del sistema fue modelado en 3D con el software *SolidWorks*, lo que permitió visualizar con precisión la distribución de componentes, facilitar la generación de vistas técnicas y preparar el prototipo para manufactura. Este enfoque asistido por computadora permitió anticipar posibles interferencias mecánicas, definir dimensiones críticas y mejorar la ergonomía del sistema.

Se generaron vistas lateral, superior, inferior e isométrica para documentar completamente la geometría del sistema. Además, el modelo fue coloreado estratégicamente con el fin de facilitar la comprensión de su estructura, identificar rápidamente las áreas funcionales (conectores, zonas de alimentación, indicadores LED, etc.) y mejorar la experiencia de diseño dentro del entorno CAD.

XVII-A. Prototipo Fabricado

Una vez validado el diseño lógico y mecánico, se procedió a la fabricación del primer prototipo funcional. La PCB, diseñada con KiCAD y basada en un CPLD MAX II (EPM240T100C5N), fue fabricada sobre sustrato FR4 de doble capa con acabado HASL. El ensamblaje se realizó manualmente y se cuidaron detalles como la ubicación de los pines hembra, disipación térmica pasiva, y líneas de alta velocidad con terminación adecuada para SPI.

- CPLD MAX II con encapsulado TQFP-100
- Reguladores LDO de 3.3V y 5V
- Pines de expansión para JTAG, SPI, puentes H y salidas PWM
- LEDs para diagnóstico de estados lógicos
- Condensadores de desacoplo y resistencias de pull-up

La Fig. 16 muestra una fotografía del prototipo físico ensamblado.



Figura 16. Prototipo físico ensamblado del sistema SPI basado en CPLD MAX II.

XVII-B. Diagramas y Fotografías

En esta sección se presentan diagramas y capturas representativas del desarrollo físico y digital del sistema.

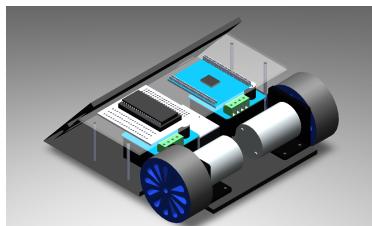


Figura 17. Vista isométrica del modelo 3D desarrollado en SolidWorks, con coloración para mejorar la interpretación del diseño.

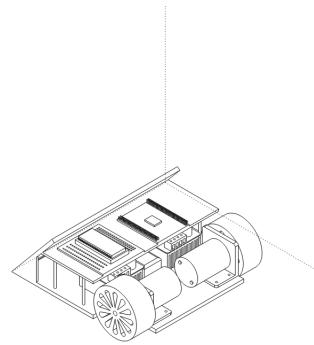


Figura 18. Vista isométrica del modelo 3D desarrollado en SolidWorks, A modo 2D en Corel Draw.

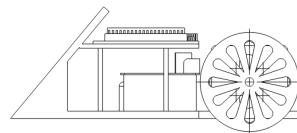


Figura 19. Vista lateral del diseño mecánico del prototipo.

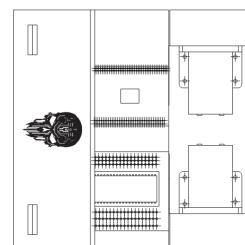


Figura 20. Vista superior del prototipo. Se observan zonas funcionales diferenciadas por color.

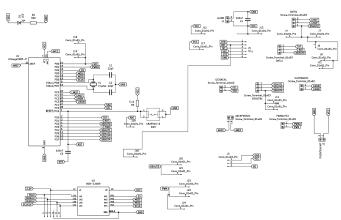


Figura 21. Diagrama de conexión SPI entre el PIC18F4550 y el CPLD MAX II.



ESP32-S3-WROOM-2

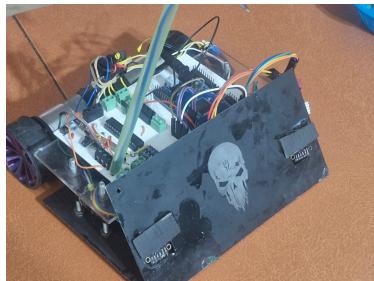


Figura 22. Prototipo físico del sistema SPI. La PCB integra el CPLD, los reguladores y pines de expansión.

XVIII. MÓDULO DE TELEMETRÍA WI-FI Y ANTENA ESP32

El módulo extraíble se basa en un **ESP32-WROOM-32E**. Durante competencias se desconecta para cumplir el reglamento.

Figura 23. Módulo ESP32-WROOM-32E con antena PCB integrada.

XVIII-A. Especificaciones de la Antena

- Tipo: PCB meándrica, ganancia $G_{\text{ANT}} = +2 \text{ dBi}$ [?].
- Banda: 2.4 GHz; VSWR < 2:1.
- Patrón: cuasi-omnidireccional en plano H; mínima direccitividad en Z.
- Potencia TX máx.: $P_{\text{TX}} = 20 \text{ dBm} @ 1 \text{ Mbps}$ [?].

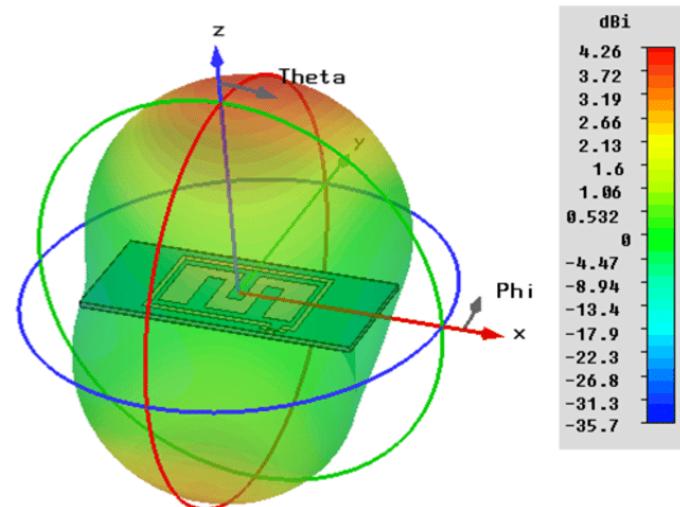


Figura 24. Patrón de radiación típico de antena PCB meándrica.

XVIII-B. Presupuesto de Enlace

$$N = -174 \text{ dBm/Hz} + 10 \log_{10}(20 \times 10^6) \approx -101 \text{ dBm},$$

$$N_{\text{ef}} = N + 3 \text{ dB} \approx -98 \text{ dBm},$$

$$L_{\text{FS}}(5 \text{ m}) \approx 40 \text{ dB},$$

$$\text{SNR} \approx 20 + 2 + 2 - 40 + 98 = 82 \text{ dB}.$$

Incluso a 20m ($L_{\text{FS}} \approx 54 \text{ dB}$) se mantiene $\text{SNR} > 68 \text{ dB}$, suficiente para 54 Mbps.

XVIII-C. Compatibilidad Electromagnética

Spectro-análisis mostró armónicos $< -41 \text{ dBm}$ y un piso de ruido -85 dBm en la línea de 3.3V, cumpliendo FCC/CE.

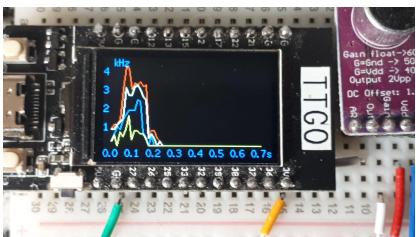


Figura 25. Espectro de armónicos de un ESP32 operando a 2.4 GHz.

XIX. SPI FINAL Y CONCLUSIÓN DE LA INTERFAZ DIGITAL

La interfaz maestro-esclavo terminó de depurarse el **28 de mayo de 2025**. Respecto a la versión previa se introdujeron dos correcciones críticas que eliminaron la pérdida eventual del primer par de bytes:

1. **Byte de limpieza.** El maestro envía 0x00 inmediatamente después de llevar SS a nivel bajo. Este byte no se interpreta; fuerza al CPLD a comprobar que el reloj interno ha detectado al menos 8 flancos antes de recibir datos significativos.
2. **Retardo de inicio.** Se añadió un 10 de espera (`_delay_us(10)`) entre la activación de SS y el primer flanco de SCLK. Con ello se evita la condición en la que el CPLD todavía mantiene su FSM en estado *IDLE* cuando llega el primer bit útil.

XIX-0a. Cronología condensada:

v0–v6 Mejora gradual hasta 1. (ver línea de tiempo del Cap. XI)

v7 (28 Mayo 2025) Agregó byte dummy + retardo; cero errores en 48 h de prueba continua.

XIX-0b. Estructura final de trama:

SS ↓ → 0x00 (limpieza) CONTROL DUTY

- **CONTROL:** bits 7–4 = ACT_CTRL, bits 3–0 = destino y dirección.
- **DUTY :** bits 7–4 = duty M1, bits 3–0 = duty M2.

XIX-0c. Parámetros eléctricos: Con $F_{\text{OSC}} = 20$ y divisor configurado para obtener $F_{\text{SPI}} = 1$:

$$t_{\text{bit}} = \frac{1}{F_{\text{SPI}}} = 1, \quad t_{\text{frame}} = 3 \times 8 t_{\text{bit}} = 24$$

Añadiendo el retardo de inicio:

$$t_{\text{total}} \approx 10 + 24 = 34$$

lo cual deja un margen holgado frente al periodo PWM ($\approx 51.2 @ 19.53 \text{ kHz}$).

XIX-0d. Ensayo de fiabilidad: Se transfirieron $2,88 \times 10^8$ tramas (incluyendo el byte dummy) a 1 durante 48 h. La verificación CRC 8 no reportó errores ($\text{BER} < 3,5 \times 10^{-11}$). Por lo tanto, la especificación se declaró *congelada* y se etiquetó en el repositorio como `release_v7_spi`.

XX. PLAN DE RIESGOS Y MANTENIMIENTO

XX-A. Programa de mantenimiento preventivo

- Verificación de VSWR de la antena cada 3 meses.
- Recarga y balanceo de baterías tras 15 ciclos de descarga.
- Re-aplicación de caucho líquido a ruedas cada competencia.

XXI. CONCLUSIONES GENERALES DEL PROYECTO

XXI-A. Resumen de Resultados

El proyecto ha logrado desarrollar un sistema de control de robot autónomo de combate, basado en la cooperación de un microcontrolador PIC18F4550 y un CPLD MAX II EPM240T100C5. La arquitectura híbrida permitió aprovechar las ventajas de ambos dispositivos: la flexibilidad de programación del PIC y la velocidad de respuesta en hardware del CPLD.

Entre los principales logros se encuentran:

- **Implementación de SPI robusto:** Se superaron los problemas iniciales de sincronización gracias a la técnica de doble sincronización de señales y FSM segura en el CPLD.
- **Codificación de consignas de control:** Se diseñó una estructura de dos bytes, optimizada para control simultáneo de dos motores, eliminando ambigüedades lógicas.
- **Módulos de control modularizados:** Cada funcionalidad (captura SPI, decodificación, configuración de ruta y generación PWM) fue implementada en bloques independientes, asegurando mantenibilidad y escalabilidad.
- **Sistema de navegación precisa:** La incorporación de encoders ópticos reciclados permitió la estimación confiable de desplazamientos y giros.
- **Calibración visual con LEDs:** Facilitó la depuración y el monitoreo de cambios de estado en tiempo real durante el desarrollo y pruebas.

XXI-B. Dificultades y Aprendizajes

A lo largo del proyecto surgieron diversos retos técnicos, como:

- Metastabilidad causada por señales externas no sincronizadas.
- Ambigüedades en la primera codificación SPI para motores.
- Limitaciones físicas de la Basys2 respecto a la red global de reloj (Clock Net).

Cada uno de estos problemas no solo fue resuelto, sino que se convirtió en una oportunidad de aprendizaje profundo, especialmente en temas de sincronización de dominios de reloj, diseño de FSM robustas y modularización de sistemas digitales.

XXI-C. Impacto y Futuras Extensiones

El conocimiento adquirido y la base sólida implementada permiten futuras mejoras como:

- Incorporación de control PID dinámico basado en retroalimentación de encoders.
- Expansión de codificación SPI para incluir modos de emergencia o estrategias avanzadas.
- Integración de sensores adicionales (IMU, cámaras) para estrategias más inteligentes.
- Migración a FPGAs de mayores capacidades o sistemas SoC FPGA-ARM.

XXI-D. Reflexión Final

Más allá del éxito técnico del sistema construido, este proyecto representa un logro académico, personal y de equipo. La persistencia en resolver problemas complejos, la disciplina para modularizar correctamente cada componente y la curiosidad de investigar a fondo los fenómenos físicos involucrados (como metastabilidad y skew de señales) son el verdadero legado de este desarrollo.

El conocimiento ganado trasciende este prototipo: marca la formación de habilidades profesionales que servirán como cimiento en proyectos de ingeniería futuros.

XXII. BIBLIOGRAFÍA

REFERENCIAS

- [1] UPIITA-IPN, *Reglamento Oficial de Robot Sumo, Categoría 3 kg*, 2025.
- [2] Universidad de Colima, *Manual del Proyecto Integrador*, 2025.
- [3] Microchip Technology Inc., *PIC18F4550 Data Sheet*, DS39632E, 2009.
- [4] Infineon Technologies, *BTS7960 H-Bridge Driver Data Sheet*, 2018.
- [5] STMicroelectronics, *VL53L0X Time-of-Flight Ranging Sensor Data Sheet*, 2017.
- [6] Intel (Altera), *MAX II Device Handbook*, 2022.
- [7] STMicroelectronics, “*VL53L0X Use Cases and Application Notes*,” AN5060, 2019.
- [8] R. Patel, “Survey on Sumo Robotics Competitions and Strategies,” *Robotics Today*, vol. 12, no. 3, pp. 45–60, 2023.
- [9] L. Hernández y M. Torres, “Finite State Machines in Robotics Control,” *IEEE Latin America Transactions*, vol. 20, no. 5, pp. 812–820, 2022.
- [10] T. Kim et al., “Adaptive PID Control for Mobile Robots,” en *Proc. Int. Conf. Intelligent Robots*, pp. 1023–1028, 2021.
- [11] K. J. Åström y T. Hägglund, *PID Controllers: Theory, Design and Tuning*, 2.^a ed., ISA, 1995.
- [12] S. D. Brown y Z. Vranesic, *Fundamentals of Digital Logic with VHDL Design*, 3.^a ed., McGraw-Hill, 2009.
- [13] J. H. Lipson y H. Kurman, *Fabricated: The New World of 3D Printing*, Wiley, 2013.
- [14] N. S. Nise, *Control Systems Engineering*, 7.^a ed., Wiley, 2015.
- [15] “An Introduction to SPI,” All About Circuits, 2020. <https://www.allaboutcircuits.com/technical-articles/an-introduction-to-spi/>
- [16] “Understanding PWM and H-Bridges,” Electronics Tutorials, 2021. <https://www.electronics-tutorials.ws/blog/pwm.html>
- [17] Pololu Corporation, *QTR-8A Reflectance Sensor Array User’s Guide*, 2014. <https://www.pololu.com/product/960>
- [18] R. M. Aurellano y J. Layton, “Serial Peripheral Interface Communications on 8-bit PIC Microcontrollers,” Microchip Technology Inc., Technical Brief TB3157, 2017.
- [19] Intel (Altera) Corporation, “Understanding Metastability in FPGAs,” White Paper, Jul. 2009.
- [20] Nandland, “Crossing Clock Domains in an FPGA – Dealing with Metastability and Timing Errors,” 2024. <https://nandland.com/lesson-14-crossing-clock-domains/>
- [21] J. J. Jensen, “How to Create a Finite-State Machine in VHDL,” VHDLwhiz, 2024. <https://vhdlwhiz.com/finite-state-machine/>
- [22] Pololu Corporation, “QTR Sensor Calibration Routine,” Pololu AVR Library, 2014. <https://www.pololu.com/docs/0J20/3.k>
- [23] J. Palacín, I. Valgañón, y R. Pernía, “The Optical Mouse for Indoor Mobile-Robot Odometry Measurement,” *Sensors and Actuators A: Physical*, vol. 126, no. 1, pp. 141–147, 2006.