

PROYECTO FIN DE GRADO

TÍTULO: Coordinación de robots mediante el uso de inteligencia de enjambre

AUTOR: Carlos Berbell Sánchez

TITULACIÓN: Electrónica de Comunicaciones

TUTOR: Javier Malagón Hernández

DEPARTAMENTO: Departamento de Ingeniería Telemática y Electrónica (DTE)

VºBº TUTOR

Miembros del Tribunal Calificador:

PRESIDENTE: Antonio Pérez Yuste

TUTOR: Javier Malagón Hernández

SECRETARIO: Mario Vega Barbas

Fecha de lectura:

Calificación:

El Secretario,

Resumen

Coordinación de robots mediante el uso de inteligencia de enjambre

La inteligencia de enjambre hace uso de las propiedades emergentes que surgen de la interacción de un conjunto de agentes o robots relativamente incapaces, entre ellos, y con el entorno. Dichas propiedades emergentes no están explícitamente programadas, y son el resultado de la operación de múltiples agentes o robots que ejecutan unas reglas simples.

La robótica de enjambre es un campo de investigación centrado en el desarrollo de metodologías, algoritmos, y aplicaciones de los principios de inteligencia de enjambre para el control de sistemas multi-robot. La aplicación de dichos principios dota los sistemas multi-robot de tres características notables: escalabilidad, flexibilidad y robustez.

Por la naturaleza de los sistemas de robótica de enjambre, su desarrollo y validación requiere del uso de un número considerable de robots, o en todo caso, de agentes simulados. Por este motivo, un área de investigación se centra en la creación de plataformas robóticas y simuladores. El presente trabajo está contenido dentro de esa área.

Los objetivos originales del proyecto estaban orientados a la implementación de algoritmos basados en inteligencia de enjambre que permitieran coordinar un pequeño grupo de robots. Estos robots, utilizarían el entorno de Arduino, y un Arduino UNO como controladores. Tras esto, se implementaría un modelo reducido de ellos en el simulador Gazebo.

Las deficiencias encontradas en el hardware original obligaron a reorientar el proyecto y sus objetivos originales. Así, el fin último de este proyecto pasó a ser el de desarrollar una plataforma robótica orientada a la robótica de enjambre, dejando la implementación de algoritmos de robótica de enjambre, y la creación de un modelo simulado a futuros proyectos derivados de este.

Para el desarrollo de la plataforma robótica, en primer lugar se han identificado las necesidades más habituales que se presentan durante el desarrollo de algoritmos de robótica de enjambre. Entre las más relevantes está la necesidad de disponer de un sistema de sensores que permita detectar e identificar otros robots en un entorno cercano, así como distinguirlos de los obstáculos. Dado que no existen módulos comerciales que implementen esta función, una parte del presente trabajo se ha orientado al desarrollo de este sistema.

Además de estos trabajos, para dar soporte a la plataforma robótica en el entorno de desarrollo de Arduino, se ha implementado un *core* de Arduino con soporte nativo de FreeRTOS. Adicionalmente, se ha desarrollado un sistema de carga remota de programas, que permite actualizar el programa ejecutado por las plataformas a través de una conexión Wifi. Dicho sistema está integrado en el *core* de Arduino.

Por último, se ha desarrollado una API de control mínima, que permite comprobar la funcionalidad de la plataforma y que podría servir de base para futuros desarrollos derivados de este proyecto.

Abstract

Multi-robot coordination using swarm artificial intelligence

Swarm intelligence leverages emergent properties that arise from the interaction of a set of relatively incapable agents or robots among themselves and with the environment. These emergent properties are not explicitly programmed; instead, they result from the operation of multiple agents or robots executing simple rules.

Swarm robotics is a research field focused on developing methodologies, algorithms, and applications of swarm intelligence principles for controlling multi-robot systems. Applying these principles provides multi-robot systems with three notable characteristics: scalability, flexibility, and robustness.

Due to the nature of swarm robotics systems, their development and validation require a considerable number of robots, or at the very least, simulated agents. For this reason, one area of research focuses on creating robotic platforms and simulators. This current work falls within that area.

The project's original objectives were aimed at implementing swarm intelligence-based algorithms to coordinate a small group of robots. These robots would use the Arduino environment and an Arduino UNO as controllers. After this, a reduced model of them would be implemented in the Gazebo simulator.

The deficiencies found in the original hardware forced a reorientation of the project and its initial objectives. Thus, the ultimate goal of this project became the development of a robotic platform oriented towards swarm robotics, leaving the implementation of swarm robotics algorithms and the creation of a simulated model for future derivative projects.

For the development of the robotic platform, the most common needs encountered during the development of swarm robotics algorithms were first identified. Among the most relevant is the need for a sensor system that can detect and identify other robots in a nearby environment, as well as distinguish them from obstacles. Since no commercial modules implement this function, a part of this work has focused on developing this system.

In addition to this work, to support the robotic platform in the Arduino development environment, an Arduino *core* with native FreeRTOS support has been implemented. Additionally, a remote program loading system has been developed, which allows updating the program executed by the platforms via a Wi-Fi connection. This system is integrated into the Arduino *core*.

Finally, a minimal control API has been developed, which allows checking the platform's functionality and could serve as a basis for future developments derived from this project.

A mi familia, y
a la memoria de los que ya no están.

Agradecimientos

En primer lugar, quiero agradecer a mis padres el apoyo que me han dado a lo largo de todos estos años. Sin sus recursos, aliento y cariño no habría podido llegar hasta aquí. Sin vosotros, esto solo sería una quimera.

En segundo lugar, quiero agradecer a mi hermana ser un pilar en que el que apoyarme en los momentos difíciles, y una fuente de inspiración. Tu capacidad, perseverancia y valores iluminan mi camino.

En tercer lugar, y muy especialmente, quiero agradecerle a mi tutor, Javier Malagón Hernández, todo el apoyo recibido durante este largo camino. Su incansable guía, y las buenas ideas que siempre me ha dado, han hecho posible este viaje. Pese a todos los problemas que te he causado, nunca te rendiste conmigo. Gracias, de verdad.

En cuarto lugar, quiero agradecer a mis amigos que me hayan aguantado, y animado, durante este proceso. Los senderos parecen más cortos y menos oscuros cuando te acompaña buena gente.

En quinto lugar quiero agradecer a la Escuela y a sus profesores, los esfuerzos que realizan para transmitir a la próxima generación de ingenieros lo que han aprendido. *Si he podido ver más allá es porque me encaramé a hombros de gigantes.*

Por último, quiero agradecer a todos aquellos inventores, ingenieros y científicos, que trabajan en la oscuridad, incansablemente, y sin detenerse, sus contribuciones. Nuestro mundo no sería lo que es sin ellos. *Puede que el presente sea de ellos, pero el futuro me pertenece, Nikola Tesla.*

Índice de contenidos

1. Introducción.....	1
1.1 Marco y motivación del proyecto.....	1
1.2 Objetivos técnicos y académicos.....	2
1.3 Estructura del resto de la memoria.....	3
2. Antecedentes.....	5
2.1 Las tortugas de Walter.....	5
2.2 Fukuda y los robots celulares.....	9
2.3 Hacia una definición formal.....	10
2.4 Definición formal y características.....	11
2.5 Campos de investigación relacionados.....	13
2.6 Taxonomías para la <i>robótica de enjambres</i>	14
2.7 Entornos de desarrollo para <i>robótica de enjambre</i>	19
2.8 El problema del posicionamiento relativo.....	21
3. Especificaciones y restricciones de diseño.....	25
3.1 Requisitos habituales de los algoritmos.....	25
3.2 Alcance del proyecto.....	26
3.3 Requisitos generales.....	26
3.3.1 Requisitos no funcionales.....	27
3.3.2 Requisitos funcionales.....	27
3.4 Especificaciones.....	28
3.5 Verificación de especificaciones y requisitos.....	30
4. Descripción de la solución propuesta.....	31
4.1 Hardware – Robot ARGOS.....	31
4.1.1 Nivel 0 – Placa de los <i>encoders</i> (Encoder_board).....	33
4.1.2 Nivel 1 – Placa de potencia (Power_board).....	38
4.1.2.1 Diseño de la unidad de supervisión de la batería.....	38
4.1.2.2 Diseño del <i>driver</i> de control de motores.....	43
4.1.2.3 Conexión con la placa madre.....	52
4.1.2.4 Diseño de la PCB.....	53
4.1.3 Nivel 3 y 4: Fundamentos del <i>sensor de rango y rumbo</i>	55
4.1.3.1 Definición y descripción de funcionamiento.....	55
4.1.3.2 Algoritmo de cálculo.....	56
4.1.3.3 Diseño preliminar y diagrama de bloques.....	58
4.1.3.4 Proceso de diseño.....	61
4.1.3.5 Parámetros de diseño escogidos.....	70
4.1.4 Nivel 3 – Placa del emisor del sensor de rango y rumbo (<i>R&B_sensor_tx board</i>).....	72
4.1.4.1 Leds emisores de infrarrojos: selección, caracterización y disposición.....	72
4.1.4.2 Amplificadores de transconductancia.....	73
4.1.4.3 Matriz de activación.....	84
4.1.4.4 Generador de onda sinusoidal.....	86
4.1.4.5 Fuente de alimentación.....	99
4.1.4.6 Diseño de la PCB y soporte de los emisores infrarrojos.....	100

4.1.5 Nivel 4 – Placa del receptor del sensor de rango y rumbo (R&B_sensor_rx board).....	102
4.1.5.1 Selección de fotodiodos, caracterización y disposición.....	103
4.1.5.2 Acondicionador de los fotodiodos y multiplexor.....	104
4.1.5.3 Etapas de amplificación.....	110
4.1.5.4 Filtro de acoplamiento entre etapas.....	116
4.1.5.5 Filtro paso-banda.....	117
4.1.5.6 Detector de pico.....	120
4.1.5.7 Fuente de alimentación.....	125
4.1.5.8 Cálculo del ruido total.....	126
4.1.5.9 Cálculo del error total de ganancia.....	127
4.1.5.10 Diseño de la placa y piezas en 3D.....	128
4.1.6 Nivel 5 – Placa de interfaz de usuario (<i>User_interface_board</i>).....	130
4.1.7 Nivel 2 – Placa madre (<i>motherboard</i>).....	133
4.1.7.1 Diagrama de bloques.....	133
4.1.7.2 Distribución de buses.....	134
4.1.7.3 Compatibilidad entre niveles lógicos.....	134
4.1.7.4 Conmutadores de la interfaz UART.....	135
4.1.7.5 Circuito de programación del Atmega2560.....	136
4.1.7.6 Circuito de programación del ESP8266.....	137
4.1.7.7 Conexiones con los sensores.....	138
4.1.7.8 Fuente de alimentación.....	138
4.1.7.9 Diseño de la placa.....	140
4.2 Software.....	141
4.2.1 Arquitectura de la API.....	142
4.2.2 Integración de FreeRTOS con Arduino y e implementación del <i>core</i>	143
4.2.3 Firmware del ESP8266 y sistema de programación remota.....	144
5. Resultados.....	147
5.1 Resultados del ensamblado.....	147
5.2 Pruebas sobre el hardware.....	148
5.3 Pruebas del sensor de rango y rumbo.....	149
5.4 Pruebas del sistema de programación remota.....	150
6. Presupuesto.....	151
6.1 Costo de recursos no específicos.....	151
6.2 Costo de los recursos específicos.....	152
6.3 Costo de recursos humanos.....	152
6.4 Costo total del proyecto.....	152
7. Impacto del proyecto.....	153
8. Conclusiones.....	155
8.1 Conclusiones.....	155
8.2 Trabajos futuros.....	155
9. Referencias.....	157
Anexos.....	161
Anexo A - Esquemas de Argos.....	161
Anexo B - Ensamblado y piezas 3D.....	167

Índice de figuras

Figura 1: Relaciones entre los campos de investigación.....	14
Figura 2: Taxonomía I de Brambilla et. al.....	15
Figura 3: Taxonomía II de Brambilla et. al.....	16
Figura 4: Diagrama de bloques del sensor de R&B diseñado por I. Kelly y A. Martinoli [14].....	22
Figura 5: Posición de los receptores infrarrojos, sensor R&B de I. Kelly y A. Martinoli [14].....	22
Figura 6: Diagrama de bloques de la electrónica de ARGOS.....	32
Figura 7: Encoder óptico incremental en cuadratura [20].....	34
Figura 8: Acondicionador de señal, encoder_board.....	35
Figura 9: Curvas de descarga de una batería para varias temperaturas, y varias condiciones de carga. [30].....	39
Figura 10: Esquema de la BMU.....	40
Figura 11: Puente en H [23].....	43
Figura 12: Modulación por ancho de pulso (PWM) [24].....	44
Figura 13: Esquema de conexión con el microcontrolador, DRV8876.....	45
Figura 14: Driver motor 1.....	46
Figura 15: Valores recomendados.....	46
Figura 16: Control en modo PH/EN, DRV8876.....	46
Figura 17: Opciones de configuración de IMODE, DRV8876.....	47
Figura 18: Modelo térmico del driver.....	50
Figura 19: Encapsulado HTSSOP impedancia térmica unión-ambiente para PCB con 1-oz de grosor de cobre.....	52
Figura 20: Conexión entre la placa de potencia y la placa madre.....	53
Figura 21: Sensor de rango y rumbo.....	55
Figura 22: Posición de los fotodiodos en el módulo - [16].....	56
Figura 23: Relación entre el término de rango, r , y la distancia, d . - [16].....	58
Figura 24: Diagrama de bloques preliminar emisor del sensor R&B.....	60
Figura 25: Diagrama de bloques preliminar receptor del sensor R&B.....	60
Figura 26: Relación entre la irradiancia recibida y la distancia al emisor.....	63
Figura 27: Modelo de error, sensor R&B.....	63
Figura 28: TSAL6400 - Curva I_e/I_f [33].....	72
Figura 29: Amplificador transconductancia.....	73
Figura 30: Modelo de una fuente de corriente real.....	74
Figura 31: Modelo para el cálculo ganancia de lazo, $T(jf)$	76
Figura 32: Resultados de la simulación, i_o	82
Figura 33: Variación del error de ganancia del amplificador de tranconductancia, Monte Carlo N=1001.....	83
Figura 34: Lógica de control del emisor del sensor R&B.....	84
Figura 35: Disposición de sectores, sensor R&B.....	85
Figura 36: Generador de onda sinusoidal.....	86
Figura 37: Características de la onda cuadrada.....	87

Figura 38: Generador de onda cuadrada.....	89
Figura 39: Plantilla de diseño del filtro paso-bajo.....	91
Figura 40: MFF paso-bajo, alimentación bipolar.....	95
Figura 41: $v_c(t)$, onda cuadrada generada.....	97
Figura 42: $v_{EXC}(t)$, señal de excitación.....	97
Figura 43: Análisis de Monte Carlo para el generador de onda cuadrada, N=1000.....	98
Figura 44: Análisis de Monte Carlo para el filtro paso-bajo, N=100000.....	98
Figura 45: Diagrama de bloques del receptor del sensor de rango y rumbo.....	102
Figura 46: Curva I_{ra}/E_e [45].....	104
Figura 47: Acondicionador para los fotodiodos.....	105
Figura 48: Modelo AC, acondicionador del fotodiodo.....	106
Figura 49: Análisis de Monte Carlo para el acondicionador de los fotodiodos, N=1000.....	108
Figura 50: Modelo de ruido del acondicionador del fotodiodo.....	109
Figura 51: Amplificador de sección.....	111
Figura 52: Modelo de ruido del amplificador.....	111
Figura 53: Análisis de Monte Carlo para el amplificador de la primera etapa, N=1000.....	114
Figura 54: Análisis de Monte Carlo para el amplificador de la segunda etapa, N=1000.....	114
Figura 55: Análisis de Monte Carlo para el amplificador de la tercera etapa, N=1000.....	115
Figura 56: Filtro de acoplamiento entre etapas.....	116
Figura 57: Modelo de ruido del filtro paso-alto.....	116
Figura 58: Esquema del filtro MFF paso-banda.....	118
Figura 59: Análisis de Monte Carlo para el filtro paso-banda, N=1000.....	120
Figura 60: Detector de pico.....	121
Figura 61: Fuente de alimentación V_{CC}	125
Figura 62: Fuente de alimentación V_{SS}	125
Figura 63: Fuente de alimentación V_{BIAS}	125
Figura 64: Circuito anti-rebotes.....	130
Figura 65: Conexión del joystick.....	130
Figura 66: Conexión del módulo OLED.....	130
Figura 67: Conexión del zumbador.....	130
Figura 68: Conexión de la IMU 1 - magnetómetro.....	131
Figura 69: Conexión de la IMU 2 - acelerómetro y giroscopio.....	131
Figura 70: Diagrama de bloques de ARGOS.....	133
Figura 71: Conversor de nivel discreto.....	135
Figura 72: Conversor de nivel discreto.....	135
Figura 73: Conversor de nivel para las líneas del SPI y UART.....	135
Figura 74: Selector de modo y conmutación de la interfaz UART.....	136
Figura 75: Circuito de auto-reset del microcontrolador principal - Atmega2560.....	137
Figura 76: Circuito de auto-reset del microcontrolador secundario - ESP8266.....	137
Figura 77: Alimentación de la placa madre.....	138
Figura 78: Arquitectura de la API.....	142
Figura 79: Sistema de control del robot B.....	149

Índice de imágenes

Imagen 1: Chasis 4WD Rover DFRobot.....	1
Imagen 2: Driver de motores.....	1
Imagen 3: Arduino UNO.....	1
Imagen 4: Elmer (Machina Speculatrix).....	5
Imagen 5: Elsie (Machina Docilis).....	5
Imagen 6: Elsie, sin la cubierta de plástico.....	6
Imagen 7: Elección de la mejor fuente de luz.....	7
Imagen 8: Repulsión de una fuente de luz demasiado brillante.....	7
Imagen 9: Búsqueda de una fuente de luz oculta, tras esquivar un obstáculo.....	7
Imagen 10: Prototipos de las células.....	9
Imagen 11: Ejemplo de combinación de las células.....	9
Imagen 12: Foot-bot del proyecto Swarmanoid.....	20
Imagen 13: Hand-bot del proyecto Swarmanoid.....	20
Imagen 14: Eye-bot del proyecto Swarmanoid.....	20
Imagen 15: Mooreboot, la plataforma que implementa el sensor R&B de I. Kelly y A. Martinoli [14]	22
Imagen 16: Posición de los sensores de velocidad en el chasis.....	33
Imagen 17: Caja reductora DFRobot L Shape.....	34
Imagen 18: Caja reductora DFRobot L Shape.....	34
Imagen 19: Capas top y bottom de la PCB.....	36
Imagen 20: Modelo 3D de la encoder_board.....	36
Imagen 21: Disco de código.....	37
Imagen 22: Soporte de la PCB.....	37
Imagen 23: Ensamblado virtual del encoder.....	37
Imagen 24: Ensamblado virtual del encoder.....	37
Imagen 25: Capa top de la Power_board.....	53
Imagen 26: Capa bottom de la Power_board.....	53
Imagen 27: Simulación 3D de la placa terminada.....	54
Imagen 28: Capa top de la R&B_sensor_tx.....	100
Imagen 29: Capa bottom de la R&B_sensor_tx.....	100
Imagen 30: Soporte de los leds IR.....	100
Imagen 31: Simulación 3D de la placa terminada, vista superior.....	101
Imagen 32: Simulación 3D de la placa terminada, vista inferior.....	101
Imagen 33: Capa top de la R&B_sensor_rx.....	128
Imagen 34: Capa bottom de la R&B_sensor_tx.....	128
Imagen 35: Soporte de los fotodiodos.....	129
Imagen 36: Simulación 3D de la placa terminada, vista superior.....	129
Imagen 37: Capa top de la user_interface_board.....	132
Imagen 38: Capa bottom de la user_interface_board.....	132
Imagen 39: Soporte de la antena.....	132

Imagen 40: Simulación 3D de la placa terminada, vista superior.....	132
Imagen 41: Capa top de la motherboard.....	140
Imagen 42: Capa bottom de la motherboard.....	140
Imagen 43: Simulación 3D de la placa terminada, vista superior.....	140
Imagen 44: Modificaciones del archivo main.cpp.....	144
Imagen 45: Enjambre de robots ARGOS.....	147
Imagen 46: Correcciones de la placa madre - versión 0.....	148
Imagen 47: Correcciones de la placa receptor del sensor de rango y rumbo - versión 0.....	148
Imagen 48: Acercamiento.....	149
Imagen 49: Adaptación.....	149
Imagen 50: Seguimiento.....	149
Imagen 51: Sistema de programación remota cargando un programa.....	150
Imagen 52: Interfaz de Argos durante la carga remota.....	150

Lista de acrónimos

- ADC:** Analog-to-Digital Converter
- AHAL:** Arduino Hardware Abstraction Layer
- AO:** Amplificador Operacional
- BMU:** Battery Management Unit
- BSP:** Board Support Package
- DCR:** Discharge Count Register
- DDL:** Device Driver Layer
- DHAL:** Device Hardware Abstraction Layer
- FCC:** Full Charge Capacity
- HAL:** Hardware Abstraction Layer
- HAL:** Hardware Abstraction Layer
- IC:** Initial Capacity
- IDE:** Integrated Development Environment
- IMU:** Inertial Measurement Unit
- LSB:** Least Significant Bit
- LVTTTL:** Low Voltage Transistor to Transistor Logic
- MRS:** Multi-Robot Systems
- PHAL:** Peripheral Hardware Abstraction Layer
- RTOS:** Real Time Operating System
- SE:** Swarm Engineering
- SI:** Swarm Intelligence
- SOC:** State Of Charge
- SR:** Swarm Robotics
- TTL:** Transistor to Transistor Logic
- UART:** Universal Asynchronous Receiver-Transmitter
- USB:** Universal Serial Bus



1. Introducción

1.1 Marco y motivación del proyecto

La robótica de enjambre constituye un campo de estudio orientado al control de sistemas multi-robot utilizando los principios de la inteligencia de enjambres. Esta metodología se basa en las propiedades emergentes que surgen de la interacción de un conjunto de robots, relativamente incapaces, entre sí mismos, y con el entorno.

Dado que las propiedades emergentes no se programan de forma explícita, sino que son el resultado de la interacción entre los agentes del sistema, los cuales ejecutan solo unas reglas simples, estos sistemas exhiben de tres características deseables: escalabilidad, flexibilidad y robustez. Estas características son difíciles de conseguir con otros enfoques, como el control mediante sistemas centralizados. Es por esto, que la robótica de enjambre, y más generalmente, los principios de inteligencia de enjambres constituyen un campo de investigación atractivo y con multitud de potenciales aplicaciones prácticas.

El objetivo original de este proyecto era desarrollar una aplicación de software basada en inteligencia de enjambre para el control conjunto de robots utilizando algunos elementos heredados de otros proyectos: un Arduino UNO, un chasis 4WD Rover, un *driver* de motores, un módulo de comunicaciones Zigbee y un sensor de ultrasonidos. Adicionalmente, se desarrollaría un modelo simulado de esta plataforma en *Gazebo*, lo que permitiría evaluar un algoritmo de robótica de enjambre sobre un enjambre de robots reales, y en un enjambre de robots simulados.



Imagen 1: Chasis 4WD Rover DFRobot

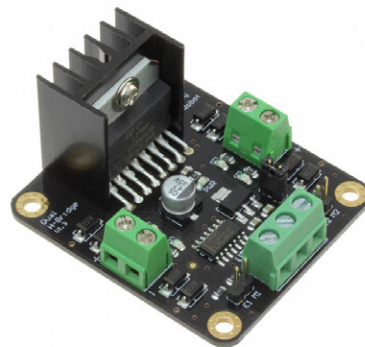


Imagen 2: Driver de motores

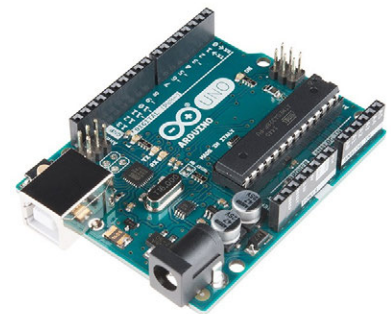


Imagen 3: Arduino UNO

Una de las suposiciones iniciales es que se podría estimar con suficiente precisión la posición de cada robot, conociendo el tiempo de activación de los motores, y su sentido de giro. Si cada robot compartía esta información con el resto, conociendo la posición inicial de todos los robots, cualquier robot podría localizar al resto de robots con esa información. Durante las pruebas preliminares, se comprobó que estas suposiciones eran erróneas. El error en la estimación producida por un solo robot aumentaba muy rápidamente, lo que hacía inviable la aplicación de este concepto en la realidad.

El mismo problema se ha presentado en desarrollos similares [13], y se debe a la acumulación de errores a lo largo del tiempo.

Dado que en los algoritmos basados en inteligencia de enjambre los robots deben ser capaces de interactuar entre sí, es necesario que, al menos, puedan localizarse. Las plataformas de robótica de enjambre utilizadas por los investigadores cuentan con sensores específicos para realizar esta función. Como que en el mercado no existían sensores comerciales que pudieran realizar esta función, se decidió desarrollar uno. Sin embargo, dadas las limitaciones del Arduino UNO en cuanto a número de pines y capacidad de memoria, su uso no era viable para esta aplicación.

Tras sopesar las opciones y limitaciones del hardware, y constatar que la implementación de algoritmos basados en inteligencia de enjambre estaba fuertemente restringida por las capacidades del hardware, se decidió reorientar el proyecto hacia el desarrollo de una plataforma de hardware más completa, que permitiera, en trabajos derivados de este, implementar algoritmos de robótica de enjambre de forma fiable.

Así, el enfoque del proyecto pasó de ser un desarrollo fundamentalmente dirigido al software, a enmarcarse dentro desarrollo de hardware. Dada la necesidad de realizar un diseño robusto y adecuado, la implementación de algoritmos de basados en inteligencia de enjambre, se convirtió en un requisito del hardware en vez de un objetivo alcanzable en el presente proyecto.

Por tanto, en lo que a la implementación de software se refiere, este proyecto solo implementa lo mínimo necesario para demostrar las capacidades de la plataforma, dejando el desarrollo de una API orientada al desarrollo de algoritmos basados en inteligencia de enjambre a futuros proyectistas.

Entre los elementos de software mínimos que se desarrollan en este proyecto están: un *core* de Arduino con soporte de FreeRTOS, que permita la integración del hardware desarrollado en el IDE de Arduino, un sistema de programación remota que permita programar las plataformas a través de Wifi, y un conjunto de test para demostrar el funcionamiento de la plataforma.

1.2 Objetivos técnicos y académicos

Desde un punto de vista técnico, los objetivos de este proyecto son:

- Desarrollar de una plataforma de robótica móvil de bajo costo orientada a la robótica de enjambre.
- Desarrollar una API mínima que permita demostrar la funcionalidad de la plataforma.
- Desarrollar un *core* de Arduino que permita la programación y uso de la plataforma robótica en el entorno de desarrollo de Arduino y que cuente con soporte para FreeRTOS.
- Desarrollo de una herramienta de programación remota que se integre en el entorno de Arduino y que permita la carga de programas a todos los robots del enjambre.

Desde el punto de vista académico, las competencias adquiridas son:

- Desarrollo y planificación de proyectos de forma autónoma.

- Desarrollo de las competencias de investigación.
- Desarrollo de las competencias en la redacción de memorias técnicas.

1.3 Estructura del resto de la memoria

En primer lugar, la exposición de este trabajo comienza con presentando los orígenes de la robótica de enjambre, continuando con los primeros trabajos formales, englobados dentro del campo de la robótica de enjambre. Tras lo cual, se exponen las definiciones de conceptos que han ido surgiendo con la evolución de esta disciplina, y se exploran las relaciones con otros campos de investigación. Para finalizar con se citan algunos trabajos relacionados con este proyecto.

A continuación, se realiza un breve análisis de los casos de uso de esta plataforma, y a partir de estos, se presentan los requisitos, y las especificaciones de diseño formales que este proyecto debe cumplir.

Tras esto, se exponen las soluciones propuestas para dar cumplimiento a las especificaciones. Este apartado se divide en dos bloques: hardware y software. En el bloque de hardware se expone una relación detallada de los desarrollos y diseños derivados de estos, de la electrónica del robot, Dado que este proyecto está orientado hacia el desarrollo de hardware, este bloque ocupa una considerable extensión. El siguiente bloque, correspondiente al software desarrollado, tiene una extensión mucho más modesta y en él se presentan los desarrollos de software realizados. Como ya se ha expuesto, estos desarrollos son mínimos, por lo que no se entra en demasiado detalle.

Después de esto, se presentan las pruebas a las que ha sido sometido el hardware. En este proyecto se han desarrollado dos versiones del hardware. La primera iteración, la Versión 0, contenía numerosos errores, que fueron corregidos en la segunda iteración, esto es la Revisión A. Dada la extensión que ocuparía una relación extremadamente detallada de los errores de la primera revisión, y lo poco que aporta, solo se comentan los resultados las pruebas más relevantes de la segunda iteración del hardware, la Revisión A.

Tras lo cual, se presenta una estimación del costo de este proyecto, teniendo en cuenta los costos materiales no específicos, los costos materiales específicos y los costos en recursos humanos de este proyecto. Seguidamente, se presenta un análisis del impacto del proyecto y su contribución a los *Objetivos para el Desarrollo Sostenible*.

Por último, se presentan las conclusiones que se extraen de este proyecto, y se proponen algunas líneas de desarrollo para futuros trabajos, que permitirían alcanzar los objetivos originales recogidos en el *Anteproyecto*, en proyectos derivados de este.

Adicionalmente, en el Anexo A se exponen los esquemas de la electrónica desarrollada, correspondientes a la Revisión A. En el Anexo B, se presenta una simulación del ensamblado virtual de la plataforma y las piezas diseñadas.

2. Antecedentes

En primer lugar, antes de proceder a definir cuáles son las características de la robótica en enjambre, presentar definiciones, campos de aplicación y desarrollos más destacados; es conveniente explicar algunos trabajos tempranos que si bien no pueden englobarse exactamente dentro de esta disciplina, están relacionados por algunas ideas fundamentales.

Como en muchos casos en ingeniería, sin importar la disciplina, la aplicación de la idea o la creación de un producto funcional, es anterior a la definición formal de los principios que lo gobiernan [63]. En otras palabras, las personas resuelven problemas guiados por su ingenio e intuición, y posteriormente, estas soluciones son mejoradas, a través de análisis que conducen a una mejor comprensión del problema y de las variables que influyen en él. Esto permite mejorar la solución, formalizando el conocimiento y expandiendo su aplicabilidad a otros problemas por resolver.

Todos los campos de la ingeniería han pasado por este proceso que comienza con un conjunto de ideas incompletas y guiadas por los problemas que las inspiraron, y cristaliza en un conjunto de conocimientos y principios formales, verificados usando el método científico, cuando el campo en cuestión alcanza la madurez [64].

En el caso de la robótica en enjambre o *swarm robotics* (en inglés), estas ideas incompletas y tempranas comienzan con los trabajos pioneros en robótica de Grey Walter, tal como se expone en [4].

2.1 Las tortugas de Walter

Entre la pascua de 1948 y las navidades de 1949, Grey Walter desarrollo dos robots, denominados *Elmer* (ELECTroMEchani-cal Robot) o *Machina Speculatrix*, y *Elsie* (Electro Light Sensitive with Internal and External stability) o *Machina Docilis* [6].

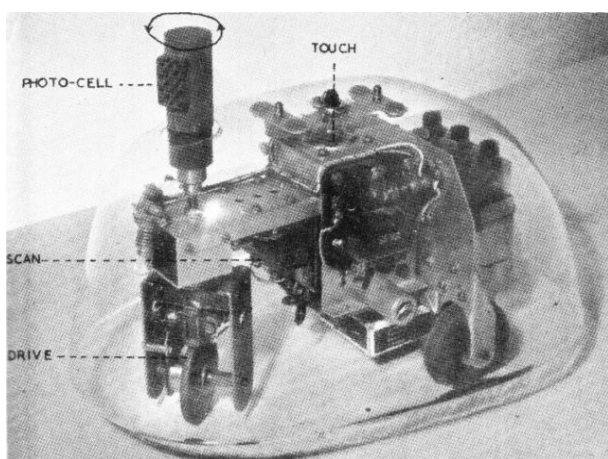


Imagen 4: Elmer (*Machina Speculatrix*)

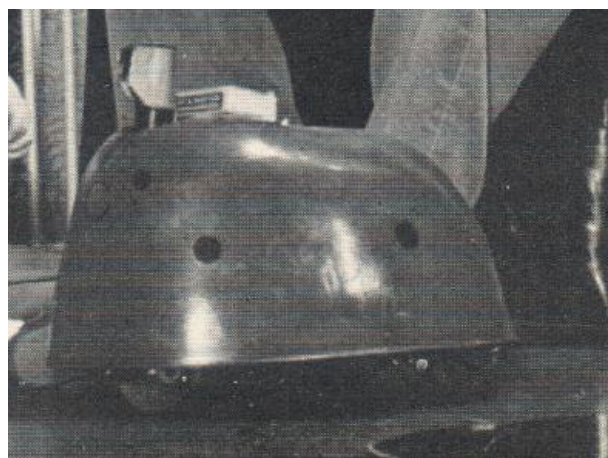


Imagen 5: Elsie (*Machina Docilis*)

En su obra, *Living Brain*, Walter describe la construcción de los robots, y su comportamiento. Tanto Elmer como Elsie tienen un chasis basado en un modelo de triciclo. Esto es, disponen de tres ruedas: la rueda frontal o rueda de dirección, gira verticalmente, provee de tracción y permite

alterar la dirección de la plataforma; y las dos ruedas traseras, de giro libre, permanecen paralelas en todo momento y solo proporcionan dos puntos de apoyo, no proveen de tracción motora.

Adicionalmente, ambos robots llevaban una fotocélula unida solidariamente con el eje de la rueda de dirección, así como una bombilla en la parte frontal. Como se explicará un poco más adelante, esta fotocélula y la bombilla, les permitía generar diversos patrones de conducta en función de la luz que captarían.

Además de esto, ambos robots llevaban una cubierta de plástico que estaba sujeta a un contacto, de forma que si el robot chocaba con algún obstáculo el contacto se cerraba, y esto originaba otro tipo de comportamiento. Esta cubierta, la cual recordaba vagamente al caparazón de una tortuga, les valió a los robots el apodo de “tortugas”.

En la siguiente imagen de Elsie, se puede observar con más claridad el ensamblaje del chasis, la posición del motor de tracción de la rueda de dirección, la posición del motor de dirección, la fotocélula y la bombilla dispuesta en la parte frontal.

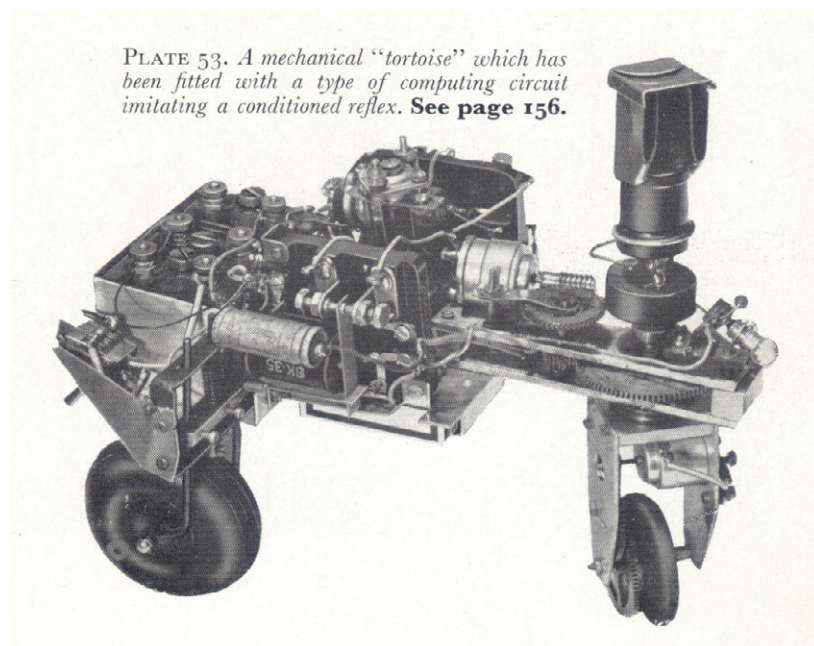


Imagen 6: Elsie, sin la cubierta de plástico

Elmer, el primer robot desarrollado, era puramente reactivo, es decir, no tenía ningún tipo de memoria. Tal como el propio Walter describe, el funcionamiento de Elmer es muy simple: mientras la luz recibida por la fotocélula este por debajo de un umbral determinado, la rueda de dirección gira verticalmente, escaneando posibles fuentes de luz y la tracción del robot funciona a media velocidad. Esto provoca que el robot realice una trayectoria circular. Mientras el motor de dirección de la rueda frontal está activo, la bombilla frontal del robot permanece activada.

Si alguna fuente de luz es percibida, la rotación vertical de la rueda de dirección se detiene, la bombilla frontal se apaga, y la tracción se activa a máxima velocidad, de forma que el robot se dirige hacia la fuente de luz. Cuando la luz recibida por el robot supera un determinado umbral, el robot entran en un estado de “deslumbramiento”, activando la rotación vertical de la rueda de dirección, y alejando al robot de la fuente de luz.

A parte de este comportamiento, si el contacto que detecta que la cubierta del robot ha impactado con algún obstáculo se cierra, el amplificador de la fotocélula se realimenta a través de un condensador, convirtiéndose en un multivibrador y dejando de actuar como amplificador. Esto provoca que el robot oscile entre dos comportamientos: el de escaneo buscando una fuente de luz y el de deslumbramiento, rotando para alejarse de la fuente de luz. El resultado es, que el robot gira y se desplaza a trompicones, moviendo, si es ligero, o esquivando, si es pesado y después de múltiples impactos, el obstáculo detectado.

El controlador de este robot está construido usando elementos discretos disponibles en la época, esto es, relés, resistencias, condensadores y tubos de vacío. No obstante, el comportamiento observado, parafraseando al propio Walter, trasciende la simplicidad de los elementos que forman que el robot.

Walter observó comportamientos previsibles, como tropismo positivo, es decir, atracción, hacia fuentes de luz de intensidad medio, y tropismo negativo, repulsión, hacia fuentes de luz de intensidad alta. Otros patrones, lógicos según el diseño del propio mecanismo, también se produjeron: atracción hacia la fuente de luz más brillante de entre varias opciones, exploración del entorno, liberándose de forma autónoma al encontrar un obstáculo en su camino, etc.



Imagen 7: Elección de la mejor fuente de luz

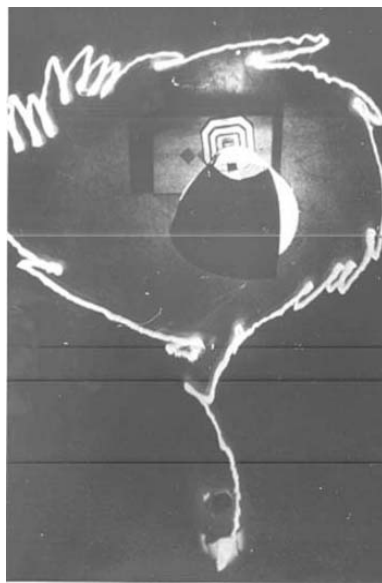


Imagen 8: Repulsión de una fuente de luz demasiado brillante



Imagen 9: Búsqueda de una fuente de luz oculta, tras esquivar un obstáculo

A parte de estos patrones previsibles, algo inesperado ocurrió durante los muchos experimentos que Walter realizaba con los robots: un patrón no previsto se manifestó de forma espontánea.

Al colocar varios robots como Elmer juntos, si no había ninguna fuente de luz, estos tendían a agruparse. Si alguna fuente de luz se situaba cerca del grupo, estos rompían la agrupación y se desplazaban hacia la fuente de luz, para volver a agruparse si se retiraba la luz.

La razón de este aparente comportamiento social era la bombilla frontal, que indicaba que el motor de dirección estaba activado. Cuando dos robots se situaban en la oscuridad, ambos

manténían el motor de dirección activado, rotando la fotocélula mientras giraban, buscando una fuente de luz. Un robot, al detectar la bombilla encendida del otro robot, la tomaba como una fuente de luz y se dirigía hacia ella. Al iniciar la marcha hacia, el robot desconectaba el motor de dirección, apagando la bombilla.

Dado que el otro robot no detectaba ninguna luz, permanecía girando y buscando. Cuando el primer robot alcanzaba al robot que estaba escaneado buscando fuentes de luz, podía impactar contra él o ser repelido por el brillo de la luz frontal. Sea como fuere, activaba el motor dirección y encendía la bombilla frontal. De la misma forma, el otro robot realizaba el mismo patrón, provocando que ambos robots se aproximaran el uno al otro. Es evidente que la aproximación era un tanto caótica, pero el resultado era siempre el mismo, la agrupación de los robots en una zona determinada.

El propio Walter explicaba que ese comportamiento era fortuito, un resultado inesperado del diseño del robot, pues con la bombilla frontal solo se buscaba saber cuándo el motor de dirección estaba activado.

Es por tanto, un comportamiento o propiedad emergente; es decir, una propiedad que surge del diseño del robot y del entorno en el que se le sitúa, pero que no ha sido explícita o deliberadamente diseñada.

Como se describirá un poco más adelante, la robótica en enjambre busca, entre otras cosas, aprovechar las propiedades emergentes de grupos de robots para hacer algo útil.

Esta es la razón por la que el trabajo de Walter es citado como pionero en el campo de la robótica de enjambre, porque fue el primero en implementar, de forma accidental, describir y razonar, sobre la aparición de un comportamiento emergente en un grupo de robots o autómatas extremadamente simples.

Aunque el incipiente trabajo de Walter con los robots es reseñable, no tuvo continuación, pues su campo profesional era la neuropsicología no la robótica o la ingeniería. Habría que esperar a finales de los ochenta, cuando aparecieron los primeros trabajos que buscaban desarrollar nuevos paradigmas para la coordinación de múltiples robots, esta vez, desde el punto de vista de la ingeniería [4], [5].

En los ochenta, el campo de investigación de los autómatas celulares estaba en su máximo apogeo [1]. Durante este tiempo, las ideas de vida artificial, autómatas auto-replicantes y evolución computacional formaban un crisol que entremezclaba numerosos campos del conocimiento: matemáticas, biología, computación, ingeniería, psicología, y por supuesto, robótica.

Es en este contexto donde Toshio Fukuda y Seiya Nakagawa presentan un concepto nuevo, inspirado por la idea de los autómatas celulares, pero con potenciales aplicaciones prácticas: los robots celulares [1].

2.2 Fukuda y los robots celulares

En 1988, Fukuda y Nakagawa publican un artículo [7] titulado: *Approach to the Dynamically Reconfigurable Robotic System*, en la revista *Journal of Intelligent and Robotic Systems*.

En el artículo, se describe un nuevo concepto: un sistema robótico formado por varias células, entiéndase: unidades funcionales autónomas, que son capaces de reconfigurarse para adaptarse a una nueva tarea, al entorno o autorepararse (si alguna de las unidades queda no operativa). Fukuda denomina a este sistema: DRRS, esto es: *Dynamically Reconfigurable Robotic System*.

La investigación presentada es una prueba de concepto, en la que se describe la estructura del sistema, los prototipos de las células que lo forman, la arquitectura de control, unos algoritmos preliminares, y algunos experimentos.

Respecto a la estructura del sistema, Fukuda expone lo siguiente [7]: *el robot estará formado por varias células. Cada una de las células tendrá una cierta inteligencia, y pondrán unirse o separarse unos de otras, dependiendo de la tarea.* Además de esto, las células no son homogéneas, hay tres tipos de células: *células de articulación, células troncales y células de trabajo*, que pueden combinarse para obtener una configuración deseada.

En las siguientes imágenes, se pueden observar los prototipos de las células y una estructura formada por la combinación de estas:

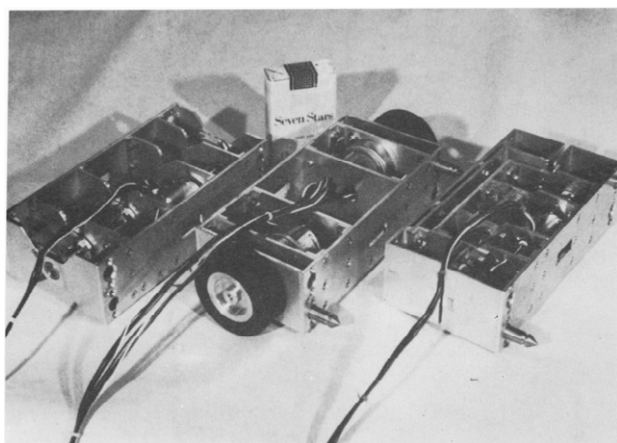


Imagen 10: Prototipos de las células

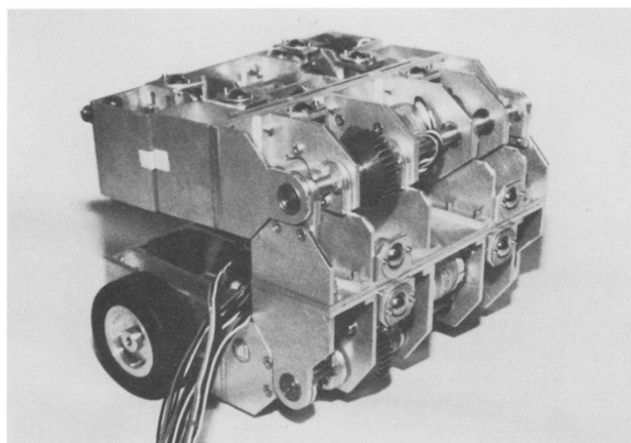


Imagen 11: Ejemplo de combinación de las células

Es evidente que las aplicaciones de estos conceptos en la práctica serían útiles, y de hecho, constituyen en sí mismos, como se verá más adelante, una de las líneas de investigación de la robótica enjambre denominada: *morphogenesis* o *auto-ensamblado*.

Sin embargo, lo más relevante de la investigación de Fukuda es el concepto sobre las capacidades de cada célula y como se relacionan con el resto: las células son estructural y funcionalmente autónomas, y trabajan independientemente de las otras células, pero cooperan con el grupo de células con un objetivo común [7].

Esa visión sobre como un grupo de robots autónomos e independientes puede cooperar con el resto, constituye una de las básicas de la robótica en enjambre. Con todo, el trabajo realizado por Fukuda y Nakagawa no puede considerarse completamente englobado dentro del campo de la

robótica en enjambre, pues entre las células del sistema existía una jerarquía de control centralizada.

La primera definición formal de inteligencia en enjambre y robótica en enjambre, la darían Gerardo Beni y Jing Wang en ese mismo año, 1988, [1] [2] [3] [5], inaugurando un nuevo campo de investigación indisciplinar y considerablemente extenso.

2.3 Hacia una definición formal

Tal como el propio Beni relata en el artículo [1], el término robótica en enjambre surgió a raíz de un intercambio de ideas tras la presentación de una ponencia [8], en el congreso organizado por el *Instituto de Estudios Avanzados* de la OTAN, bajo el título *Ethoexperimental Analysis of Behavior* en Il Ciocco, Italia en 1988.

En la ponencia presentada, entre otros temas, se exponían algunas propiedades que debían cumplir sistemas de sensores distribuidos y auto-organizables, así como las ventajas y aplicaciones que pudieran tener. Entiéndase que estos sistemas no eran sensores fijos, si no sistemas robóticos equipados con sensores, que debían coordinarse de forma descentralizada y operar como unidades autónomas.

Además de esto, Beni y Wang relacionaron estas ideas con la teoría de autómatas celulares, haciendo una distinción fundamental: los *sistemas robóticos auto-organizables* no eran autómatas celulares, pues podrían estar formados por unidades heterogéneas, para cambiar su estado (posición por ejemplo) necesitaban relacionarse con el medio, y las unidades que los formaban operaban de forma asíncrona.

Las cualidades presentadas en la ponencia para los *sistemas robóticos auto-organizables*, constituían en sí mismas un nuevo campo de estudio. Inicialmente, Beni propuso denominar a esta nueva clase de autómatas, robots celulares, pero dado que el término ya había sido aplicado por Fukuda con otro significado [1], un nuevo término fue propuesto: *robótica en enjambre* o *swarm robotics*, en inglés.

No obstante, no sería hasta la publicación del trabajo germinal: *Swarm Intelligence in Cellular Robotic Systems* en 1993, donde Beni y Wang explorarían en profundidad estas ideas y definirían formalmente las propiedades de los sistemas de robótica en enjambre, y así como el concepto de *la inteligencia de enjambre* o *swarm intelligence*, en inglés [1].

Así pues, tal como se recoge en [8], la definición de *inteligencia de enjambre*, dada por Beni y Wang, es:

Sistemas de robots no-inteligentes que exhiben un comportamiento inteligente evidente en la habilidad de producir patrones de materia ordenados en un entorno exterior de forma impredecible.

Esta definición un tanto farragosa, expone simplemente, tres cualidades. Primero, el sistema estará formado por robots no inteligentes, es decir, autómatas, guiados por reglas predecibles. Segundo, los robots deben ser corpóreos, pues deben ser capaces de actuar sobre el entorno. Tercero, los patrones generados por el sistema de robots no deben ser predecibles, en el sentido

computacional del término; es decir, no pueden ser anticipados por ningún tipo de sistema computacional que pueda construirse. Esta última cualidad se presenta en algunos desarrollos de autómatas celulares.

Según expone Beni en [1], con el tiempo, el concepto de *inteligencia en enjambre* ha evolucionado, distinguiéndose del concepto original relacionado con sistemas robóticos para adoptar un significado más amplio, originando nuevos conceptos como la *optimización de enjambre*.

Desde este primer trabajo de Beni y Wang, han sido propuestas otras definiciones más claras y concisas. En concreto, la definición más aceptada y actualmente usada, como se recoge en [3], [4], y [5], es la dada por Erol Sahin.

2.4 Definición formal y características

En 2005, Erol Sahin presentó un artículo titulado: *Swarm Robotics: From Sources of Inspiration to Domains of Application* [2]. En él, al igual que en [1], se señala que el significado de inteligencia de enjambre había cambiado, aplicándose más a estudios relacionados con insectos sociales y algoritmos de optimización que a trabajos de investigación en robótica.

Tomando en consideración la evolución que los términos habían sufrido con el paso del tiempo, Sahin se propuso dar una definición precisa de lo que es la *robótica de enjambre* o *swarm robotics*. Así mismo, con el objetivo de distinguir que trabajos o sistemas deben englobarse dentro del campo de la robótica en enjambre, también expuso un conjunto de características que deben cumplir estos. Estas características los diferencian de otros trabajos que también utilizan *sistemas multi-robot*. Por último, aunque estas ya habían sido expuestas en otros trabajos, recogió cuales son las ventajas que presentan los sistemas de robótica de enjambre.

A continuación, se expone la definición de *robótica en enjambre* y las características que debe cumplir un sistema o trabajo de investigación para englobarse dentro del paradigma de *robótica de enjambre* según Sahin [2]:

Definición de robótica de enjambre (swarm robotics o SR): *la robótica de enjambre es el estudio de como un elevado número de agentes físicamente corpóreos y relativamente simples pueden ser diseñados de forma que un comportamiento colectivo deseable emerja de las interacciones entre los agentes, y de las interacciones entre los agentes y el entorno.*

Características de un sistema de robótica de enjambre:

- *Robots autónomos y físicamente corpóreos:* los robots que forman el sistema deben ser autónomos, es decir, deben ser capaces de procesar la información recibida de forma local, y generar una respuesta, actuando sobre el entorno físicamente.
- *Gran número de robots:* el sistema debe estar formado por un número elevado de robots, o en todo caso, debe estar orientado a la coordinación de un gran número de robots. Aunque por cuestiones de practicidad o costo el número de robots sea reducido, si el sistema es escalable, esto es, permite aumentar el número de robots a coordinar, cumple con esta característica.

- *Pocos grupos de robots heterogéneos: el sistema debe estar compuesto por robots homogéneos, en la medida de lo posible. Cuanto mayor sea la variabilidad de capacidades y diseños entre los robots, menos se aproximará el sistema a las características de un sistema de robótica de enjambre.*
- *Robots relativamente incapaces: los robots que forman el sistema deben ser relativamente incapaces de desempeñar por sí mismos la tarea para la cual el sistema ha sido diseñado. En otras palabras, los robots deben poder beneficiarse de las capacidades de otros robots para desempeñar su función.*
- *Robots con capacidades limitadas: las capacidades de sensado y comunicación deben ser locales, o en todo caso, los robots no deben requerir el acceso a ningún tipo de información global compartida por todos los robots del sistema. Esto garantiza que los algoritmos de control sean descentralizados, pues no habrá posibilidad de establecer un control centralizado.*

Estas características, producen que los sistemas de robótica en enjambre tengan varias ventajas que los hacen superiores a otros enfoques para la coordinación de *sistemas multi-robot*. Sahin expuso las siguientes [2]:

- *Robustez:* dado el elevado número de robots que forman el sistema y que no existe un control centralizado, aunque varios robots queden inoperativos o sean destruidos, el sistema seguirá funcionando con los robots restantes.
- *Flexibilidad:* los sistemas de robótica en enjambre comparten características con los enjambres de insectos y estos son extremadamente adaptables a los cambios de su entorno, por tanto, los sistemas de robótica en enjambre pueden diseñarse para exhibir esa flexibilidad frente a los cambios.
- *Escalabilidad:* una de las características fundamentales de un sistema de robótica en enjambre es que debe ser escalable, por tanto, debe ser capaz de operar con un número de robots variable, lo que, como ya se ha expuesto, lo hace robusto y flexible.

Alternativamente a las definiciones dadas por Beni y por Sahin, se han propuesto otras definiciones de robótica en enjambre. En [5], Senanayake et. al, definen la *robótica de enjambre* como: *la aplicación de los principios de inteligencia de enjambre al control de sistemas multi-robot*. Esta definición, hace referencia a la definición de *inteligencia de enjambre* dada por Beni y Wang en [8]. Sin embargo, esta definición de *inteligencia de enjambre* está orientada a sistemas de robots físicos que pueden interactuar con el entorno.

La realidad, es que la evolución del término *inteligencia de enjambre* trasciende el campo de la robótica, y se aplica a otros problemas como los *algoritmos de optimización meta-heurísticos* basados en *inteligencia de enjambre* [4]. Por tanto, es necesario exponer una última definición de *inteligencia de enjambre* que sea aplicable a estos trabajos.

Esta definición se encuentra en la obra *Swarm Intelligence: From Natural to Artificial Systems* de Eric Bonabeau, Marco Dorigo, y Guy Theraulaz. Tal como se recoge en el artículo [10], que resume los puntos principales del libro, señalado en [5], la definición de *inteligencia de enjambre* es:

Definición de inteligencia de enjambre (swarm intelligence o SI): es cualquier intento de diseñar algoritmos o dispositivos, que resuelvan problemas de forma distribuida inspirados por el comportamiento colectivo de colonias de insectos sociales como hormigas, termitas, abejas, avispas y otras sociedades animales.

Antes de abandonar este apartado que contiene las definiciones fundamentales, se debe recoger aquí la definición de *ingeniería de enjambres*, dada por Brambilla et. al en [3] y que expone claramente las aspiraciones de la ingeniería aplicada a la *robótica de enjambre*.

Definición de ingeniería de enjambre (swarm engineering o SE): es la aplicación sistemática del conocimiento científico y técnico para modelar, especificar requisitos, diseñar, realizar, verificar, validar, operar y mantener un sistema basado en inteligencia de enjambre.

A partir de este punto, y por abreviar, se utilizarán las siglas SR para referirse a el término *robótica de enjambre (swarm robotics)*, y SI, para referirse a el término *inteligencia de enjambre (swarm intelligence)*.

Una vez aclaradas cuales son las definiciones principales y las características fundamentales de la *robótica de enjambre*, y más generalmente, de la *inteligencia de enjambre*; es necesario señalar, aunque sea brevemente, cuáles son los campos de investigación relacionados. Esto permitirá introducir de forma más clara las taxonomías que se han desarrollado para clasificar las diferentes líneas de investigación.

2.5 Campos de investigación relacionados

La *robótica de enjambre* constituye un campo de estudio interdisciplinar y transversal, que originalmente deriva de los *sistemas multi-robot* (en inglés: *multi-robot systems, MRS*), los que a su vez, son una extensión de los estudios realizados en *robot móviles* [3].

Actualmente, dado que la *inteligencia de enjambre* se ha establecido como un campo de investigación en sí mismo, la *robótica de enjambre* está contenida entre la intersección entre este, y los *sistemas multi-robot*. Dentro de la *inteligencia de enjambre* no solo se incluyen las líneas de investigación con aplicación directa a la robótica; también se incluyen los trabajos de estudio y análisis del comportamiento de diferentes animales sociales como hormigas, abejas, que puedan servir de inspiración para el desarrollo de algoritmos, tal como se expone en [3].

Por otro lado, una parte de los trabajos relacionados con *inteligencia de enjambre* están orientados al desarrollo de algoritmos de optimización [3]. Estos algoritmos, englobados bajo el nombre de algoritmos de *optimización meta-heurísticos inspirados en la naturaleza* por Nedjah et. al en [3], o simplemente, *optimización de enjambre* por Beni en [1], utilizan los principios de la *inteligencia de enjambre* para operar.

Así pues, si se tiene en cuenta lo expuesto, los diferentes campos de investigación y sus relaciones se pueden expresar de forma gráfica como conjuntos:

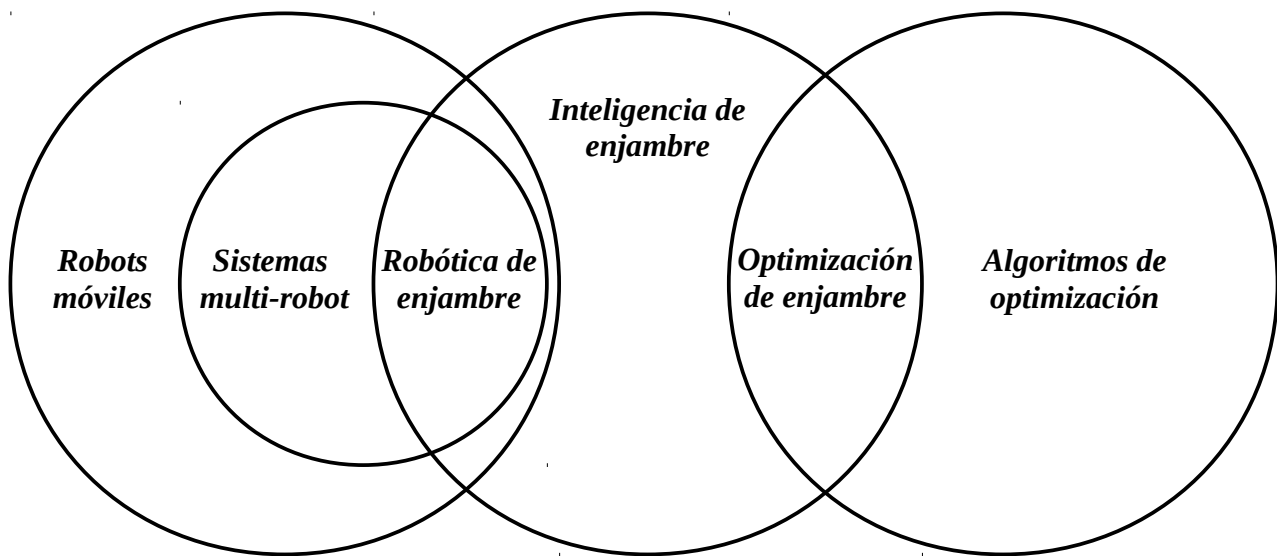


Figura 1: Relaciones entre los campos de investigación

Este diagrama permite clasificar con facilidad a que campo de investigación pertenece un determinado trabajo, no obstante, dentro de cada conjunto o intersección, hay diferentes líneas de investigación.

Este proyecto está orientado a la *robótica de enjambre*, la cual, aun siendo esta un subconjunto de la *inteligencia de enjambre*, presenta una variedad de trabajos y enfoques considerable. Al igual que el significado de la *SI* ha evolucionado, el interés por algunas líneas de investigación ha variado. Por tanto, con el objetivo de identificar cuáles son las líneas de investigación o categorías que reciben más o menos atención, se han propuesto diferentes taxonomías para clasificar las investigaciones.

2.6 Taxonomías para la *robótica de enjambres*

En *Review of methodologies and tasks in swarm robotics towards standardization*, [4], Nadia Nedjaha y Luneque Silva, recogen las principales taxonomías que se han propuesto para clasificar los trabajos de investigación.

Uno de los primeros trabajos referentes a la taxonomía, [4], se encuentra en el artículo publicado en 1993 por Gregory Dudek et. al., bajo el título: *A Taxonomy for Swarm Robots* [11]. En él se aborda la clasificación de los trabajos en función de siete parámetros fundamentales: tamaño del enjambre, rango de comunicación de los agentes, topología de la comunicaciones, ancho de banda de las comunicaciones, capacidad de reconfiguración espacial, unidad de procesamiento, y heterogeneidad de los agentes que forman el enjambre.

Esta taxonomía da mucha importancia a algunos parámetros como el rango de comunicaciones de los robots, la topología de las comunicaciones o el tipo de unidad de procesamiento, que fueron relevantes en los trabajos iniciales, más centrados en el hardware de los *MRS*. Conforme el campo fue madurando, los nuevos trabajos abordaron otros temas más relacionados con el control y la jerarquía de los *SR*.

Así, en 2001, Iocchi et. al. presentaron otra taxonomía [4] que evaluaba los trabajos de *robótica en enjambre* según cuatro características de los robots: nivel de cooperación, nivel de consciencia, tipo de coordinación y tipo de organización.

Si bien esta taxonomía evaluaba aspectos más abstractos de la *SR* que la taxonomía propuesta por Dudek et. al., seguía estando más orientada a la clasificación de los trabajos desarrollados con robots y hardware, que en aquellos enfocados al desarrollo de algoritmos de control y metodologías para los sistemas basados en *SI*.

No sería hasta el año 2007, cuando Erol Şahin y Levent Bayındır presentaron la primera taxonomía que incluía los trabajos relacionados con el modelado, y las metodologías de diseño de sistemas basados en *SI*. Esta taxonomía, clasificaba los trabajos en las siguientes categorías: modelado, diseño de comportamiento (metodologías de diseño), estudios analíticos, tipos de comunicaciones y problemas o casos de uso habituales [4].

La taxonomía propuesta por Şahin et. al. mejoraba las anteriores y era más concisa, pero desde el punto de vista de la ingeniería, era un tanto difusa. Como ya se ha expuesto, en cualquier disciplina orientada a la ingeniería, los conocimientos y metodologías tienden a formalizarse según el campo en cuestión alcanza la madurez. Esto implica responder a dos preguntas fundamentales: para qué, es decir, que problema resuelve o a que problema se aplica; y cómo, cual es el proceso o la metodología de diseño o de qué forma se implementa.

Así pues, en 2012, Manuele Brambilla et. al. presentaron un artículo con el título *Swarm robotics: a review from the swarm engineering perspective* [3], en el que se exponía una taxonomía más orientada a la ingeniería. Actualmente, esta es la taxonomía más aceptada en la literatura [4], y por tanto, es la que se seguirá en este documento.

La taxonomía de Brambilla et. al. se recoge en el siguiente esquema:

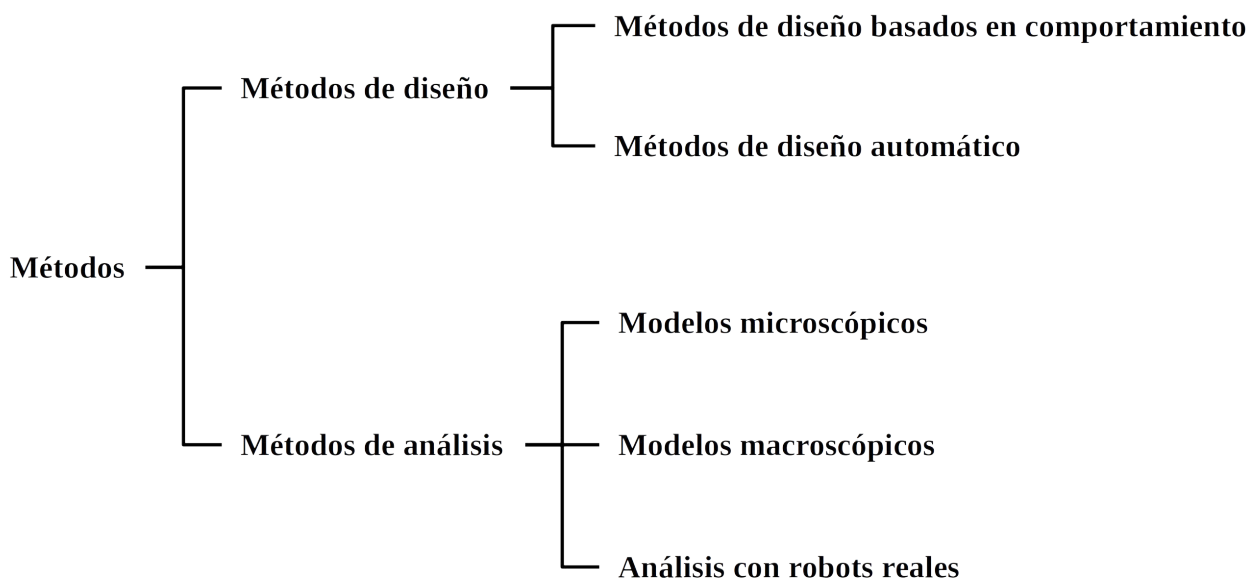


Figura 2: Taxonomía I de Brambilla et. al.

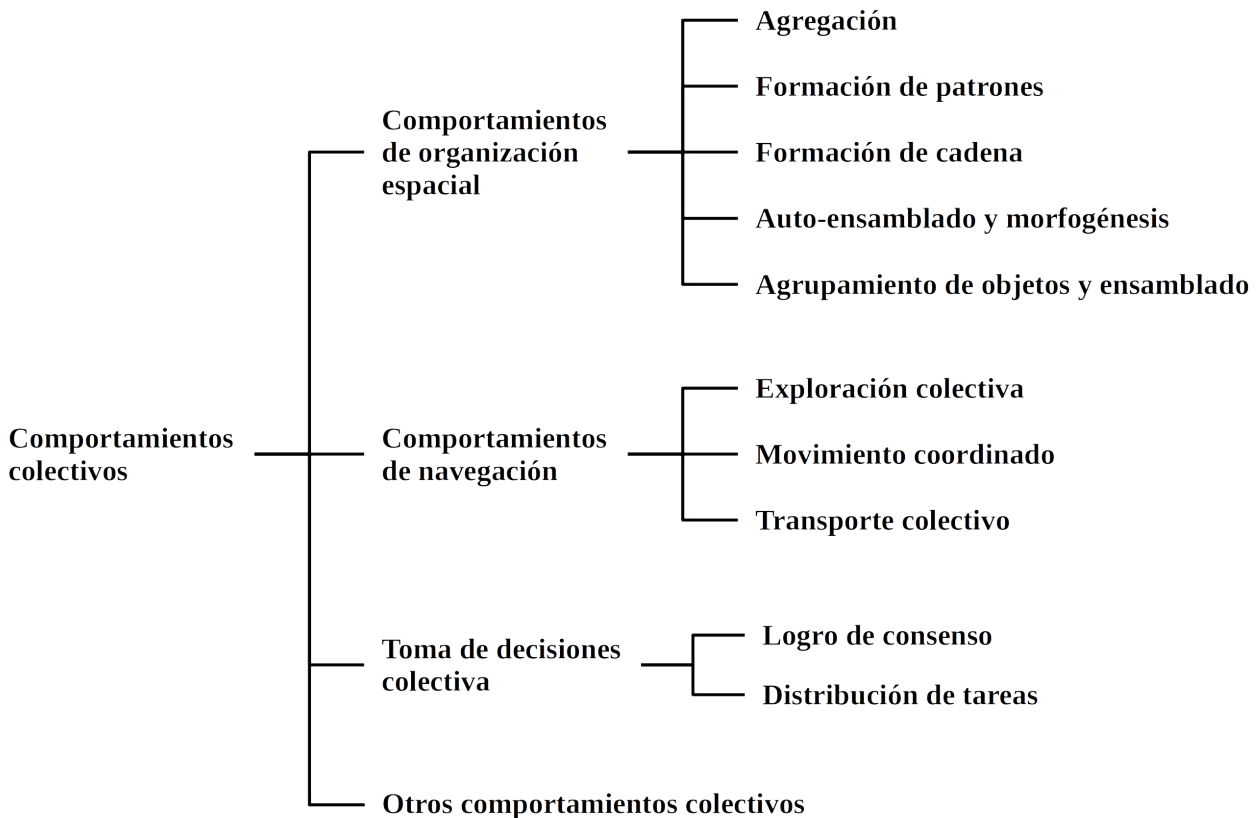


Figura 3: Taxonomía II de Brambilla et. al.

Los trabajos se evalúan según dos parámetros:

- **Métodos:** cuáles son los métodos de diseño, esto es, la metodología que se ha utilizado para diseñar los algoritmos basados en SI que ejecutan los agentes o robots; y métodos de análisis, es decir, que tipo de metodología se ha seguido para validar los algoritmos diseñados.

Dentro de los **métodos de diseño**, encontramos dos categorías:

- **Métodos de diseño basado en comportamiento:** en el que los algoritmos se diseñan de forma heurística, pudiendo inspirarse en trabajos relacionados con la biología o la entomología para obtener un comportamiento similar al de algunos insectos sociales.
- **Métodos de diseño automáticos:** el diseño de los algoritmos utiliza algún tipo de técnica englobada dentro de la programación evolutiva. En líneas generales, los algoritmos tienen un conjunto de parámetros de los que depende el comportamiento del agente o robot, y estos parámetros se modifican de forma iterativa siguiendo algún tipo de algoritmo hasta que los agentes alcanzan el comportamiento deseado. Algunos ejemplos de estos métodos de diseño son los algoritmos genéticos o el aprendizaje por refuerzo.

En cuanto a los **métodos de análisis**, se distinguen tres categorías:

- **Modelos microscópicos:** en estos se utilizan versiones simuladas de los robots o agentes, las cuales interaccionan dentro de un entorno simulado, lo que permite observar el comportamiento y verificar si cumple con lo esperado. Las versiones simuladas pueden variar en el nivel de detalle, lo cual puede afectar al resultado del análisis. No obstante,

debido a que no requiere gran cantidad de recursos, la gran mayoría de los algoritmos de SR se han validado de esta forma.

- **Modelos macroscópicos:** estos métodos de análisis utilizan modelos matemáticos para capturar los aspectos fundamentales del algoritmo que guía el comportamiento de los robots. De esta forma, se convierten estos detalles a un conjunto de ecuaciones que permiten predecir el comportamiento del conjunto de robots como un todo. El principal problema de estos métodos de análisis es que a menudo requieren realizar simplificaciones y suposiciones que no se cumplen en la práctica, como la ausencia de ruido en los sensores o de errores en los actuadores, que pueden afectar considerablemente al resultado.
- **Análisis con robots reales:** utilizando un conjunto de robots físicos, no simulados, se ejecutan el algoritmo a analizar y se observa su comportamiento en un entorno controlado. Tal como se expone en [3], su principal utilidad es que permite distinguir entre algoritmos aplicables en la práctica, los cuales pueden tolerar problemas como el ruido o los errores, de aquellos que solo pueden funcionar en un entorno simulado (casi perfecto) porque se basan en suposiciones irreales.

El otro parámetro que evalúa la taxonomía de Brambila et. al. es el **comportamiento colectivo**.

- **Comportamiento colectivo:** se define como el conjunto de comportamientos básicos que un enjambre puede desarrollar [3]. Uno de los objetivos fundamentales de la *ingeniería de enjambres*, es el desarrollo de un *framework* o marco común, que permita estandarizar la implementación de diferentes comportamientos colectivos. Así, un hipotético *ingeniero de enjambres*, podría utilizar los comportamientos colectivos como si fueran bloques de construcción, [3], lo que permitiría la implementación de aplicaciones basados en SR de forma sistemática para solucionar problemas complejos reales.

Los comportamientos colectivos pueden dividirse en cuatro categorías básicas:

- **Comportamientos de organización espacial:** son aquellos que se centran en cómo organizar y distribuir robots y objetos en espacio [3]. Dentro de esta categoría, se distinguen los siguientes:
 - **Agregación:** el objetivo es de este comportamiento es la agrupación de todos los agentes o robots en una determinada zona del espacio.
 - **Formación de patrones:** este comportamiento tiene por objetivo organizar la disposición de los robots en el espacio de forma regular y repetitiva, formando un patrón determinado.
 - **Formación de cadena:** en este comportamiento los robots tienen que posicionarse de forma que conecten dos puntos, formando una cadena.
 - **Auto-ensamblado y morfogénesis:** el autoensamblado es el proceso por el cual un enjambre de robots se conecta entre ellos para físicamente. La morfogénesis el proceso por el cual un enjambre de robots se autoensambla siguiendo un patrón particular, combinándose en una estructura que permita resolver una tarea determinada.

- **Agrupamiento de objetos y ensamblado:** el agrupamiento de objetos es el comportamiento en el que un enjambre de robots agrupa un conjunto de objetos distribuidos por el espacio en un determinado punto. El ensamblado es el proceso en el que un grupo de robots, conecta o une físicamente, un conjunto de objetos anteriormente separados. Ambos comportamientos son la base de la construcción autónoma de estructuras.
- **Comportamientos de navegación:** son aquellos que se centran en como coordinar el movimiento de un enjambre de robots [3]. Se distinguen los siguientes:
 - **Exploración colectiva:** estos comportamientos permiten distribuir un enjambre de robots por una zona desconocida, de forma que se pueda establecer una red de comunicaciones entre los robots, y utilizar los mismos para permitir la navegación de otros robots.
 - **Movimiento coordinado:** en este comportamiento los robots se mueven de forma coordinada siguiendo un determinado patrón, lo que permite, por ejemplo mejorar las capacidades de sensado del enjambre. La diferencia entre este comportamiento y el comportamiento de *formación de patrones*, es que este comportamiento los robots están en movimiento, mantenido una determinada formación según se desplazan.
 - **Transporte colectivo:** en este comportamiento, un grupo de robots se coordina para desplazar un objeto de un punto a otro, que es demasiado pesado para ser movido por un solo robot, de un punto a otro.
- **Toma de decisiones colectivas:** un enjambre de robots es a todos los efectos un sistema distribuido, y por tanto, cada uno de los robots puede tener un estado diferente al resto. Esto imposibilita que los robots elijan de forma espontánea tomar la misma decisión. Los comportamientos de toma de decisiones colectivas se aplican a cómo organizar los robots para puedan influenciarse en la toma conjunta de decisiones [3].
 - **Logro de consenso:** son los comportamientos usados para permitir al enjambre de robots alcanzar un consenso, eligiendo entre diferentes alternativas, con el objetivo de maximizar el rendimiento del enjambre.
 - **Distribución de tareas:** en SR, los enjambres son homogéneos, es decir, están formados por robots idénticos, pero eso no implica que todos realicen la misma tarea. Dentro de un enjambre los robots pueden realizar diferentes tareas; puede haber robots que sirvan de nodo de comunicación (permaneciendo inmóviles), otros que exploren, y otros que agrupen objetos. Los comportamientos de distribución de tareas se encargan de cómo distribuir dinámicamente los robots entre las diferentes tareas que el enjambre ejecuta, de forma que el rendimiento del enjambre sea máximo.
- **Otros comportamientos colectivos:** además de los comportamientos colectivos expuestos, se han desarrollado algunos otros: detección colectiva de fallos, regulación del tamaño del enjambre, o interacción humano-enjambre [3].

Como se expuso en la introducción de este proyecto, el objetivo original del mismo era el de crear una *plataforma de desarrollo* que permita la programación y validación de algoritmos de *robótica de enjambre*, utilizando un simulador (*Gazebo*) y un pequeño enjambre de robots reales. Así pues, utilizando la taxonomía de Brambilla et. al., este proyecto estaría englobado dentro de las categorías de métodos de análisis microscópicos y análisis con robots reales.

Antes de finalizar este capítulo, es por tanto necesario mencionar otros trabajos similares, en los que se ha implementado un entorno de desarrollo para la programación y validación de algoritmos basados en SI, utilizando simuladores y robots reales.

2.7 Entornos de desarrollo para *robótica de enjambre*

La *robótica de enjambre* ha sido un campo de investigación muy prolífico durante casi dos décadas. Durante ese tiempo, debido a la constante mejora en la tecnología, y a los diferentes requisitos de los equipos de investigación, se han desarrollado numerosas plataformas robóticas y simuladores destinados a *robótica de enjambre*.

Como se ha señalado en el apartado anterior, los análisis con robots físicos constituyen la forma más adecuada de discriminar entre algoritmos aplicables en la realidad de aquellos algoritmos quiméricos que solo pueden funcionar en una perfecta realidad simulada. Por este motivo, la mayoría de investigadores aspiran a validar sus algoritmos en robots físicos. Sin embargo, la realidad es que el desarrollo de una plataforma física requiere un equipo multidisciplinar y tiene un elevado costo [4], que no todas las instituciones pueden asumir.

Así pues, para aquellos equipos que pueden permitirse el desarrollo, se presentan dos opciones: adaptar una plataforma comercial de investigación en robótica general a los requisitos de la SR, o desarrollar de cero una plataforma robótica destinada a la SR. La misma cuestión se plantea respecto a los simuladores, pero en este caso, la necesidad de simular un gran número de robots suele requerir el desarrollo de un simulador específico.

En cualquier caso, los investigadores suelen intentar adaptar sus trabajos para utilizar simuladores y plataformas ya desarrollados. Lo más habitual es que los requisitos de los algoritmos requieran implementar una plataforma robótica propia, mientras que para la simulación, se implemente el modelo en alguno de los simuladores destinados robótica general o robótica de enjambre.

Existen pocos proyectos en los que se haya realizado ambas tareas: diseñar una plataforma robótica propia partiendo de cero, y desarrollar un simulador especializado en SR. Un ejemplo de este tipo lo encontramos en el proyecto *Swarmanoid*.

El proyecto *Swarmanoid*, desarrollado de forma conjunta por investigadores de varias universidades europeas [12] y liderado por Marco Dorigo, del grupo *IRIDIA* de la *Université Libre* de Bruselas, utiliza tres tipos diferentes de robots denominados: *foot-bot*, *hand-bot* y *eye-bot*.



Imagen 12: Foot-bot del proyecto Swarmanoid

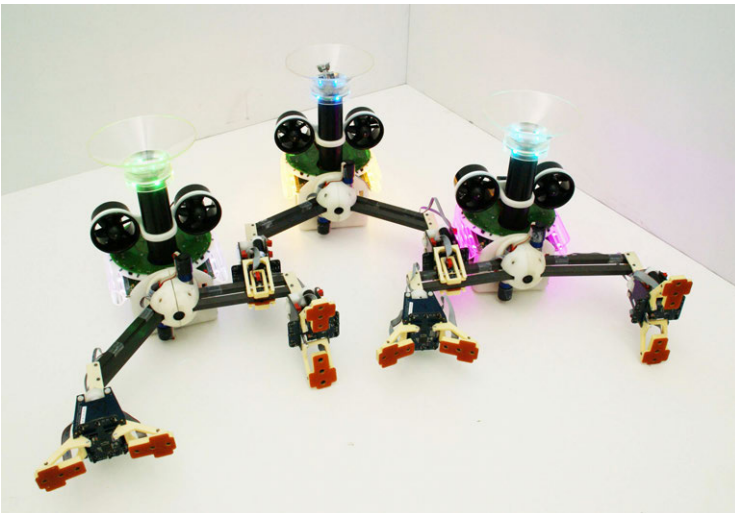


Imagen 13: Hand-bot del proyecto Swarmanoid



Imagen 14: Eye-bot del proyecto Swarmanoid

Los *foot-bot* derivan de la plataforma *S-bot* del proyecto *Swarm-bots*, del cual es continuación el proyecto *Swarmanoid*. Los *foot-bots* utilizan un sistema de tracción diferencial para desplazarse por el suelo. Los *hand-bot* implementan dos manipuladores que le permiten sujetar objetos y escalar; sin embargo, carecen de locomoción autónoma, debiendo ser desplazados mediante la cooperación de los *foot-boots*. Los *eye-bots* son robots voladores autónomos; utilizan cuatro propulsores eléctricos para desplazarse, pudiendo sujetarse a una estructura metálica situada en el techo mediante un actuador magnético. La función principal de los *eye-bots*, es la de proporcionar información que permitan coordinar a los *foot-bots* y *hand-bots*.

Además de los tres tipos de plataformas robóticas, el equipo de *Swarmanoid* desarrolló un simulador: *ARGoS*. Este simulador se creó para evitar la principal limitación de otros simuladores

no específicos de SR: la falta de escalabilidad, es decir, la incapacidad de los simuladores de robótica general para simular la interacción de un gran número de robots [12].

Respecto del hardware de los robots, todos ellos implementan algunos elementos comunes como el procesador o el sistema de comunicaciones. No es el objetivo de este apartado analizar en detalle estos elementos, sin embargo, si se debe señalar un componente fundamental que soluciona un problema recurrente en el desarrollo de robots para SR: el *sistema de alcance y rumbo relativos*.

2.8 El problema del posicionamiento relativo

Un aspecto básico de la mayoría de algoritmos de SR, es que los robots deben conocer la posición relativa de los robots que se encuentran cerca de ellos. Conocer esta información dentro de un entorno simulado es una cuestión trivial, pues todos los robots comparten un marco de referencia común y conocen su *pose*, esto es, sus coordenadas y su orientación, respecto de este.

Sin embargo, en un entorno real los robots no comparten ningún marco de referencia común. Una solución trivial a este problema sería situar los robots en posiciones conocidas al inicio. Una vez se inicie la ejecución del experimento, los robots utilizarían algún sistema de odometría y/o medición inercial para calcular su posición respecto de la posición inicial. Dado que las posiciones iniciales son conocidas, cada robot podría calcular la posición del resto de robots.

Este es un ejemplo de una idea que funciona dentro de un entorno simulado libre de errores, pero que llevada a la práctica no produce los resultados esperados. Ningún sistema de odometría o medición inercial está libre de errores, ni si quiera aquellos que utilizan métodos *fusionado de sensores* (filtros de Kalman, por ejemplo) para reducir el error. El resultado es que los errores de posición se acumulan con el tiempo, provocando que la medición de posición relativa se vuelva completamente inservible por la gran desviación que presenta. Un ejemplo de este problema se presenta en [13].

Otra solución sería utilizar un sistema de GPS para hallar las posiciones absolutas de los robots, lo que permitiría que cada robot conociera la posición del resto de robots. El problema con esta idea es la exactitud; los sistemas GPS presentan un error de varios metros, y a la escala a la que funcionan los sistemas de SR, hace que sea inaplicable.

La solución a este problema es que la mayoría de robots destinados a SR, incluido los desarrollados en este proyecto, implementen es un sistema de medición de posición relativa basado en infrarrojos.

Este sistema conocido en la literatura como *range and bearing sensing*, esto es, *sensado de rango y rumbo* fue desarrollado inicialmente por Ian Kelly y Alcherio Martinoli [14]. Posteriormente, Jim Pugh y Alcherio Martinoli realizaron varias mejoras en el algoritmo de cálculo, lo que permitió aumentar considerablemente precisión [15].

Para implementar este sistema, todos los robots llevan un emisor de infrarrojos, el cuál emite una señal modulada, y un conjunto de receptores infrarrojos dispuestos de forma radial. Para conocer la posición relativa de un robot respecto de otro, uno de los robots activa el emisor infrarrojo,

emitiendo una señal que todos los robots que tenga visión directa de él, recibirán. La señal lleva un identificador que permite al resto de robots identificar que robot está emitiendo.

Conociendo la intensidad de la señal recibida por cada uno de los receptores infrarrojos, se puede calcular el rango (la distancia), y el rumbo (el ángulo relativo) del robot que está emitiendo la señal. Además, dado que la señal infrarroja esta modulada, y lo que se mide es la intensidad de dicha componente, el sistema es inmune a la variación en las condiciones de iluminación.

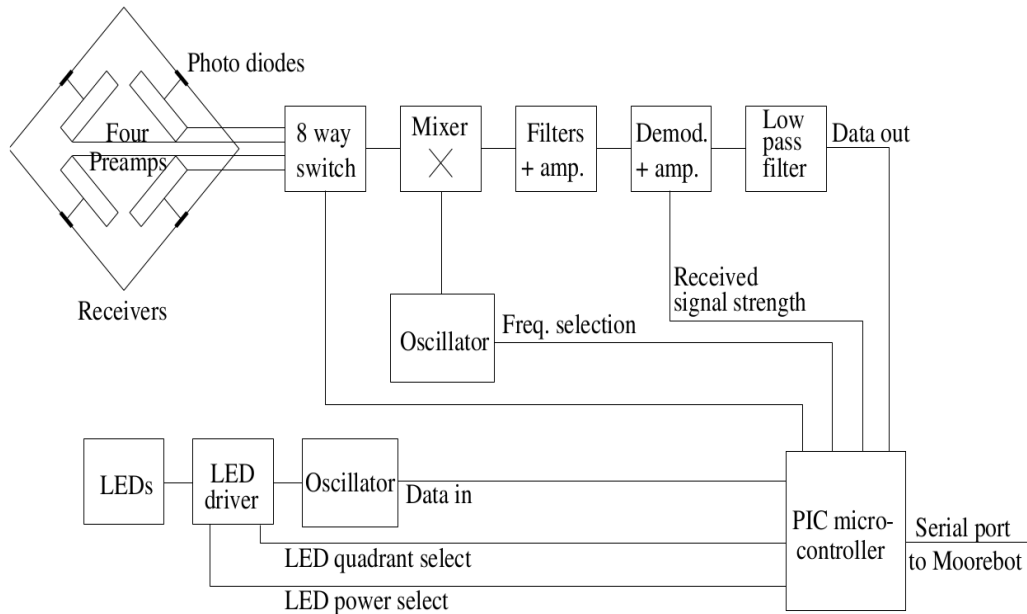


Figura 4: Diagrama de bloques del sensor de R&B diseñado por I. Kelly y A. Martinoli [14]

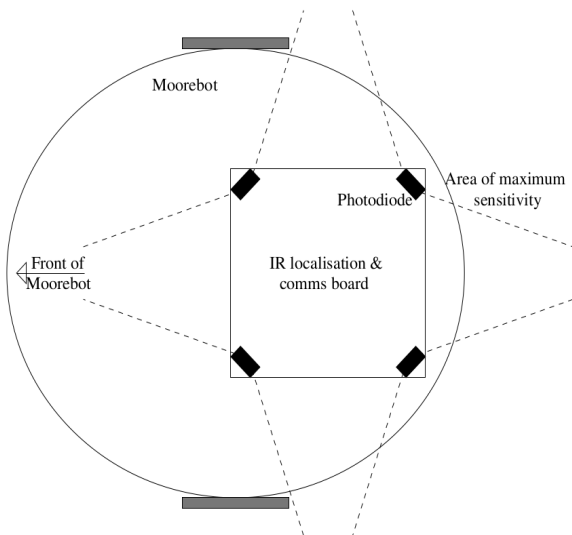


Figura 5: Posición de los receptores infrarrojos, sensor R&B de I. Kelly y A. Martinoli [14]



Imagen 15: Mooreboot, la plataforma que implementa el sensor R&B de I. Kelly y A. Martinoli [14]

Posteriormente, en el año 2009, Jim Pugh, Alcherio Martinoli, et. al. presentaron una nueva revisión destinada a la plataforma Khepera III [16], que mejoraba el diseño original incluyendo más receptores de infrarrojos. Esto aumentó la exactitud y la frecuencia de operación.

Por último, se debe mencionar el trabajo de James F. Roberts et. al. [17], el cual se basa en los trabajos de Pugh y Martinoli, y que simplifica el sistema eliminando la necesidad de establecer un canal de comunicación infrarrojo para identificar a los robots. En su lugar, se utiliza un sistema de turnos, que permite coordinar la emisión y escucha de cada robot.

El *sistema de sensado de rango y rumbo* desarrollado en este proyecto se basa en los principios presentados Jim Pugh, Alcherio Martinoli, et. al en [16], así como en las ideas desarrolladas por James F. Roberts et. al. en [17].

3. Especificaciones y restricciones de diseño

El objetivo principal de este proyecto es la creación de una plataforma robótica que permita validar algoritmos y conceptos relacionados con robótica de enjambre.

Por tanto, y antes de proceder a enumerar las especificaciones y restricciones de diseño de este proyecto, es necesario citar los requisitos más habituales que presentan los algoritmos aplicados a *robótica de enjambre*.

3.1 Requisitos habituales de los algoritmos

Se tomarán como caso de uso dos algoritmos: el algoritmo de agregación presentado por Onur Soysal y Erol Sahin en [18], y el algoritmo de movimiento coordinado desarrollado por Ali E. Turgut et. al. en [19].

En el caso del primer algoritmo, el algoritmo de agregación de Soyal et. al., tiene los siguientes requisitos:

- Plataforma móvil con tracción diferencial.
- Sensor infrarrojo que distinga entre robots y obstáculos.
- Cada robot debe ir equipado con una fuente sonora y un *array* de micrófonos que permita detectar la dirección de origen. Puede implementarse con un sensor de ultrasonidos y una estrategia de búsqueda adecuada.

En el caso del segundo algoritmo, el algoritmo de movimiento coordinado presentado por Turgut et. al., se requieren los siguientes elementos:

- Plataforma móvil con tracción diferencial.
- Sensor virtual de rumbo, que permita conocer la orientación de todos los robots del enjambre. Su implementación requiere:
 - Brújula digital.
 - Sistema de comunicaciones entre robots.
- Sensor infrarrojo que distinga entre robots y obstáculos.
- Sistema de programación remota.

Se han seleccionado estos dos algoritmos, porque sus requisitos se presentan con frecuencia en la mayoría de algoritmos aplicados a *robótica de enjambre*. No obstante, tal como se expuso en el capítulo anterior, desarrollos novedosos de algoritmos requerirán adaptar el entorno, esto es, el hardware del robot. Es por tanto necesario que el hardware desarrollado tenga cierta flexibilidad.

3.2 Alcance del proyecto

El proceso de análisis de un algoritmo utilizando robots reales se realiza ejecutando experimentos sucesivos y observando si el comportamiento del enjambre produce el resultado esperado. Este análisis puede ser cualitativo, esto es, cuando solo importa si el resultado coincide con lo esperado, sin importar ninguna métrica concreta; o cuantitativo, es decir, cuando el algoritmo debe cumplir un conjunto de métricas de rendimiento.

Para este último tipo de análisis se requiere capturar diversas variables del estado del robot, como por ejemplo la posición y orientación (pose) de cada uno de los robots del enjambre, o su trayectoria. Si bien cada robot tendrá la capacidad para estimar alguno de estos parámetros, la estimación puede ser errónea. Es por tanto necesario disponer de un sistema externo que capture estos datos, utilizando por ejemplo, un sistema de cámaras para monitorizar la posición y trayectoria de cada plataforma.

Dada la complejidad de estos sistemas de captura de datos, su implementación queda fuera del alcance de este proyecto, siendo el enfoque utilizado para el análisis de algoritmos estrictamente cualitativo.

Por tanto, este proyecto se centrará exclusivamente en:

- Desarrollo de una plataforma robótica móvil orientada a la ejecución y desarrollo de algoritmos de *robótica de enjambre*.

3.3 Requisitos generales

A continuación se expondrán los requisitos generales del proyecto; estos se dividen en dos: funcionales y no funcionales. Los requisitos no funcionales recogen condicionantes o restricciones que se deben tener en cuenta en el proyecto, pero que no implican ningún tipo de trabajo o implementación derivado de los mismos. Los requisitos funcionales recogen las cualidades del proyecto que implican algún tipo de desarrollo derivado de los mismos.

A partir de estos requisitos generales, se elaborarán unas especificaciones que el proyecto debe cumplir. Es por tanto necesario, que todos los requisitos generales cuenten con un identificador único que permita referirse a los mismos con precisión.

Así pues, los requisitos no funcionales tendrán un identificador con la forma: *RNF-x*, donde *x* será sustituido por un índice numérico asignado de forma consecutiva. Del mismo modo, a los requisitos funcionales, se les asignará un identificador con la forma: *RF-x*, donde *x* será sustituido por un índice numérico asignado de forma consecutiva.

3.3.1 Requisitos no funcionales

Son los siguientes:

- **RNF-1:** se usará el IDE de Arduino como entorno de programación.
- **RNF-2:** el software de las plataformas se programará en C++, con los límites impuestos por el ecosistema de Arduino.
- **RNF-3:** la plataforma robótica debe utilizar el chasis: 4WD Rover de DFRobot, reutilizado de un proyecto anterior.
- **RNF-4:** solo se construirán 4 robots, ya que solo se dispone de 4 chasis 4WD Rover de DFRobot.
- **RNF-5:** todas las herramientas y recursos utilizados deben ser de código abierto y diseño abierto.

3.3.2 Requisitos funcionales

A parte de los requisitos expuestos como requisitos habituales de los algoritmos, se han añadido requisitos adicionales con el objetivo de aumentar las prestaciones y la flexibilidad de la plataforma.

- Los requisitos funcionales son los siguientes:
- **RF-1:** la plataforma robótica debe ser capaz de estimar su posición relativa respecto de su posición inicial.
- **RF-2:** la plataforma robótica debe ser capaz de localizar en el espacio la posición relativa del resto de plataformas situadas a una distancia mínima de 1.5 metros.
- **RF-3:** la plataforma robótica debe ser capaz de obtener la actitud relativa o rumbo, esto es, el ángulo de orientación del resto de plataformas respecto de sí misma o de una referencia compartida.
- **RF-4:** la plataforma robótica debe disponer de un sistema que le permita comunicarse con otras plataformas y con un ordenador de propósito general.
- **RF-5:** la plataforma robótica debe disponer de una interfaz de usuario que disponga de al menos un botón de parada de emergencia.
- **RF-6:** la plataforma robótica debe disponer de una unidad que monitorice el estado de la batería y permita informar al usuario.
- **RF-7:** la plataforma robótica debe disponer de sensores que permitan localizar obstáculos en su entorno cercano.
- **RF-8:** la plataforma robótica debe disponer de algún sensor que permita detectar marcas o indicaciones situadas en el suelo.

- **RF-9:** la plataforma robótica debe disponer de un sistema de programación remota, a fin de que el programa que ejecutan pueda actualizarse sin necesidad de conectarla físicamente a un ordenador.
- **RF-10:** la plataforma robótica debe disponer de sistemas que permitan minimizar el consumo de energía si alguno de los sistemas implementados no están en uso.
- **RF-11:** la plataforma robótica debe disponer de una interfaz que permita ampliar su dotación básica de sensores o actuadores.
- **RF-12:** la plataforma robótica debe poder controlarse usando el modelo de tracción diferencial con solo dos variables: velocidad lineal y velocidad angular.
- **RF-13:** la plataforma robótica debe poder medir la cantidad de potencia que cada motor está aplicando.

A partir de estos requisitos funcionales, y como consecuencia de los mismos, se establecen las especificaciones que la plataforma robótica debe cumplir.

3.4 Especificaciones

Las especificaciones se dividen en dos tipos: especificaciones de hardware y especificaciones de software. Del mismo modo que en el caso de los requisitos generales, las especificaciones están identificadas de forma inequívoca, con el fin de poder realizar referencias futuras a las mismas.

Las especificaciones de hardware serán identificadas de la forma *EH-x*, donde *x* es un código numérico asignado de forma consecutiva. Las especificaciones de software serán identificadas con de la forma *ES-x*, donde *x* es un código numérico asignado de forma consecutiva.

Así pues, las especificaciones que el hardware, entendiéndose los robots que serán diseñados y construidos, debe implementar son:

- **EH-1:** todas las ruedas de la plataforma llevarán *encoders*.
- **EH-2:** la plataforma robótica dispondrá de una IMU (unidad de medición inercial) con un acelerómetro, giroscopio y brújula digital.
- **EH-3:** la plataforma robótica dispondrá de un *sensor de rango y rumbo* basado en infrarrojos que será capaz de obtener la posición relativa del resto de plataformas en un radio máximo de 4,3 metros, y de detectar obstáculos por reflexión de haces de infrarrojos.
- **EH-4:** la plataforma dispondrá de un microcontrolador principal, un Atmega2560, que controlará todos los actuadores, sensores e interfaces.
- **EH-5:** la plataforma dispondrá de un microcontrolador secundario, un ESP8266, con conectividad WIFI y capacidad de programación a través SPI.

- **EH-6:** la plataforma robótica dispondrá de una interfaz de usuario con los siguientes elementos: *display* gráfico oled monocromo de 126x64 pixeles y 0.96 pulgadas, botón de parada de emergencia, *joystick* con cinco botones y zumbador.
- **EH-7:** la plataforma robótica dispondrá de un módulo dedicado a monitorizar el estado de la batería, que permitirá obtener una estimación de la carga, el tiempo de funcionamiento restante, el voltaje y la corriente que entrega al sistema.
- **EH-8:** la plataforma robótica dispondrá de un sensor de distancias por ultrasonidos, que podrá ser utilizado para detectar emisiones de ultrasonidos de otros sensores.
- **EH-9:** la plataforma robótica dispondrá de un sensor de reflexión infrarrojo situado en la parte inferior.
- **EH-10:** las plataformas robóticas dispondrán de un sistema de programación remota, a fin de que el programa que ejecutan pueda actualizarse sin necesidad de conectarlas físicamente a un ordenador.
- **EH-11:** la alimentación de los sensores y actuadores de la plataforma podrá ser habilitada o deshabilitada.
- **EH-12:** la plataforma robótica dispondrá de una interfaz auxiliar basada en I2C, y de una toma de potencia auxiliar conectada al bus de alimentación de la batería.
- **EH-13:** la plataforma robótica dispondrá de un *driver* por cada motor con sensado de corriente.

En cuanto a las especificaciones que el software debe cumplir, son las siguientes:

- **ES-1:** se modificará la HAL de Arduino para incluir los servicios de un RTOS, en concreto de FreeRTOS.
- **ES-2:** se desarrollará un módulo para el *firmware* del microcontrolador auxiliar (ESP8266) que implemente el cliente para el sistema de programación remota o OAP (*over air programming*).
- **ES-3:** se desarrollará un servidor que permita la carga del programa de forma remota en los robots físicos utilizando el microcontrolador auxiliar (ESP8266) como programador.
- **ES-4:** se desarrollará un módulo para el *firmware* del microcontrolador auxiliar (ESP8266) que implemente un servidor de Telnet para la retransmisión de los datos generados por el puerto serie del microcontrolador principal.
- **ES-5:** se creará un *core de Arduino* personalizado que integre las herramientas de programación remota.
- **ES-6:** se implementará una API básica que permita testear su funcionamiento.

3.5 Verificación de especificaciones y requisitos

En los apartados anteriores se han presentado los requisitos funcionales que constituyen el conjunto de problemas que este proyecto debe resolver. Tras esto, se han presentado las especificaciones, es decir, el conjunto de características que este proyecto desarrolla con el objetivo de cumplir con los requisitos, esto es, de solucionar el conjunto de problemas que representan.

Por tanto, a partir de este punto, todos los requisitos funcionales deberían tener asociado al menos una especificación que garantice su cumplimiento. Así pues, con el objetivo de verificar esta afirmación, a continuación se presentará una matriz de requisitos-especificaciones.

En ella se representan en la primera columna los identificadores de los requisitos funcionales, ordenados de forma creciente, y en la segunda el identificador de la especificación que garantiza su cumplimiento.

ID del requisito funcional	ID de la especificación asociada
RF-1	EH-1, EH-4
RF-2	EH-3, EH-4, EH-5, ES-6
RF-3	EH-2, EH-4, EH-5, ES-6
RF-4	EH-4, EH-5, ES-1, ES-6
RF-5	EH-4, EH-6, ES-6
RF-6	EH-4, EH-7, ES-1, ES-2, ES-14
RF-7	EH-3, EH-4, EH-5, ES-6
RF-8	EH-4, EH-9, ES-6
RF-9	EH-4, EH-5, ES-2, ES-3
RF-10	EH-4, EH-11
RF-11	EH-12
RF-12	EH-4, EH-5, ES-6
RF-13	EH-13

Tal como se observa en la tabla, no existe ningún requisito funcional sin al menos una especificación asociada. Por tanto, las especificaciones planteadas son suficientes para cumplir con los objetivos de este proyecto.

4. Descripción de la solución propuesta

La solución propuesta de este proyecto se divide en dos partes: hardware, es decir, el diseño de los robots orientados al desarrollo de algoritmos de robótica en enjambre; y software, esto es, la API mínima usada para su testeo, la creación y la integración de FreeRTOS en un core de Arduino, y el sistema de programación remota desarrollado.

4.1 Hardware – Robot ARGOS

En este apartado se desarrollará la parte de hardware del proyecto; en concreto, el diseño de la electrónica de control de sensores y actuadores; así como las piezas mecánicas auxiliares, necesarias para implementar las funciones que la plataforma robótica destina al análisis de algoritmos de robótica en enjambre.

La plataforma robótica recibe el nombre de ARGOS, como referencia al titán Argos Panoptes, el cual, según la mitología, tenía cien ojos. Debido al diseño del *sensor de rango y rumbo*, el cual dispone de un anillo de emisores y receptores IR que recuerdan un conjunto de ojos apuntando en todas direcciones, el autor consideró que era un nombre adecuado.

Por tanto, para evitar repetir el término “plataforma robótica” o “el o los robots”, a partir de este momento se establece que el término ARGOS también podrá ser usado para referirse a los robots desarrollados en este proyecto.

Para el desarrollo del robot, se usarán cuatro chasis 4WD Rover de la empresa DFRobot como sistema de tracción de los robots. Si bien podría haber diseños más adecuados, dado que una parte del diseño de los robots está condicionado por esta circunstancia, el presente apartado se centrará mayoritariamente en el desarrollo de la electrónica.

Desde el punto de vista de la ingeniería electrónica, un robot no es más que un conjunto de sensores, acondicionadores de señal, controladores, y actuadores. Por tanto, a partir de los requisitos de hardware, se elabora un diagrama de bloques preliminar en el que se muestran todos los elementos de los que dispone la plataforma.

Se ha optado por dividir la implementación de todo el hardware en cinco placas de circuito impreso, las cuales se dispondrán como una pila o torre, interconectando cada una de ellas mediante pines y headers. A parte de estas cinco placas, se ha desarrollado una placa auxiliar donde se situará la fotobarrera y el acondicionador de los *encoders*.

Si bien esta necesidad surgió a lo largo del proceso de diseño, puesto que no era posible incluir todos los componentes electrónicos en una sola placa, se considera que esta división es adecuada para exponer el proceso de desarrollo del hardware, pues cada placa tiene una función definida.

La electrónica de ARGOS se divide en seis placas o niveles. Comenzando por el nivel superior, es decir, por la placa que está situada en lo alto de la torre, estas son:

- **Nivel 5:** interfaz de usuario (también es donde se sitúa la IMU).
- **Nivel 4:** receptor IR del *sensor de rango y rumbo*.

Descripción de la solución propuesta

- **Nivel 3:** emisor IR del sensor de rango y rumbo.
- **Nivel 2:** placa madre.
- **Nivel 1:** placa de potencia.
- **Nivel 0:** acondicionadores de señal de los *encoders*.

A continuación, se muestra el diagrama de bloques de la electrónica de control de ARGOS:

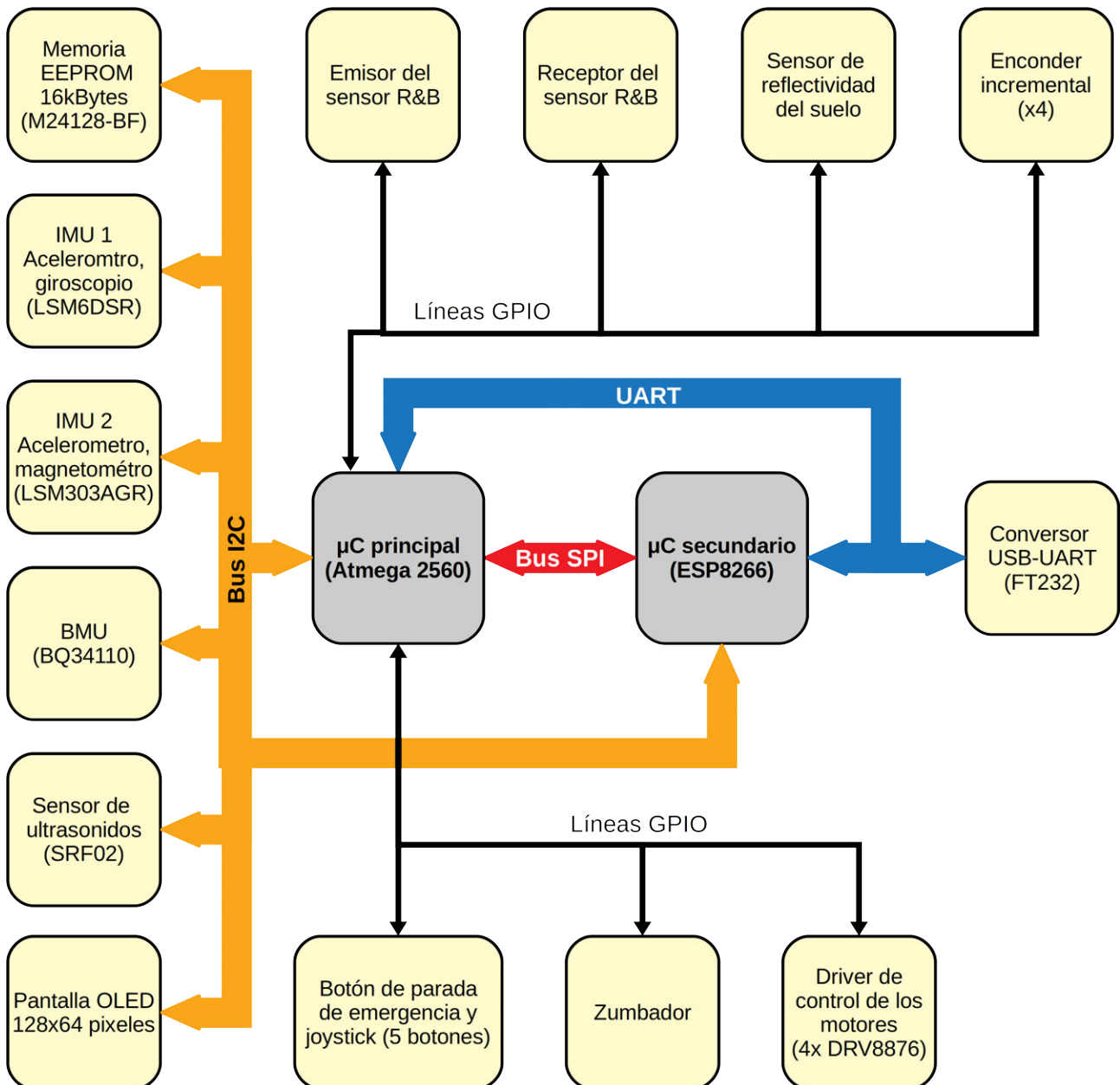


Figura 6: Diagrama de bloques de la electrónica de ARGOS

Este diagrama de bloques no incluye el módulo de programación del Atmega ni los conmutadores de las líneas de UART, que serán presentados en el apartado referente a la placa madre. El desarrollo de esta exposición comenzará por la placa del nivel más bajo, la placa del nivel 0, que contiene los *encoders* y los acondicionadores de señal de estos. Tras esto, se expondrá el proceso de diseño de la placa potencia, situada en el nivel 1, para proseguir después con las placas del

sensor de rango y rumbo (niveles 3 y 4), situadas sobre esta. A continuación de estas, se presentará el diseño de la placa de interfaz de usuario (nivel 5); y para finalizar, se presentará el diseño de la placa madre (nivel 2).

4.1.1 Nivel 0 - Placa de los encoders (Encoder_board)

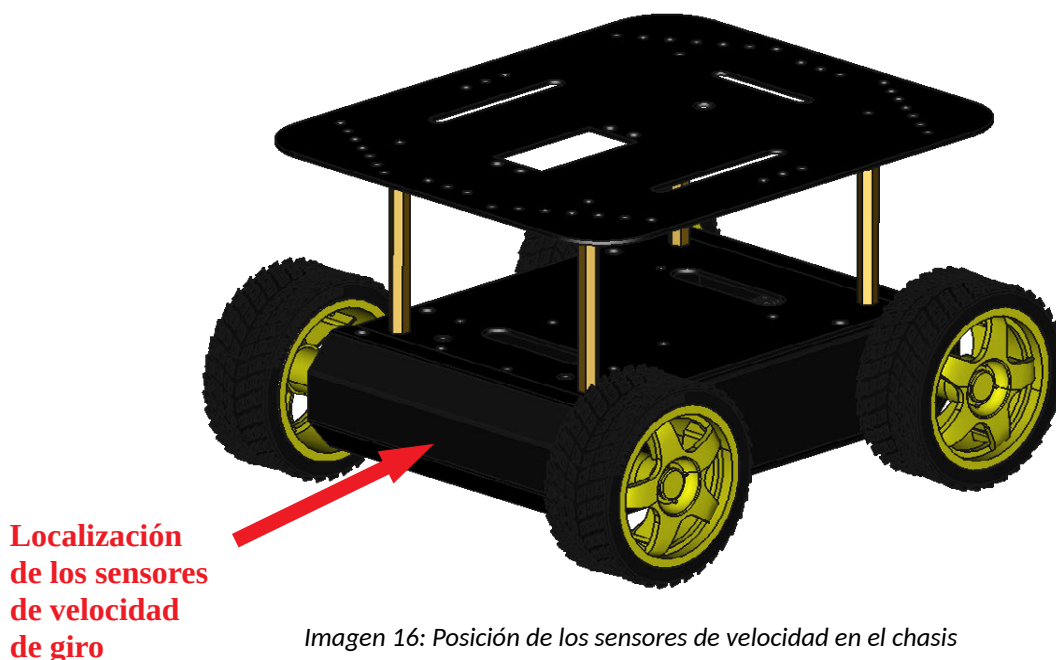
La especificación que satisface es: *EH-1*.

El chasis del que se dispone tiene cuatro ruedas, y cuatro motores, pero el modelo de robot móvil que se usará será el de una plataforma con tracción diferencial, esto es, con dos ruedas unidas a dos motores que pueden ser controladas de forma independiente, y una tercera rueda de giro libre.

Para implementar este modelo con cuatro motores, las ruedas de cada lado del robot deberán girar de forma sincronizada, idealmente, a la misma velocidad. Con todo, esto introducirá errores en el modelo cinemático del robot, puesto que si el robot realiza un giro, todas las ruedas sufrirán un cierto grado de deslizamiento.

Con todo, para el control de lazo cerrado de la pose (posición y orientación) en una plataforma con tracción diferencial, se requiere poder controlar la velocidad de los motores de forma precisa, más aún en este caso concreto. Y para poder implementar un control en lazo cerrado de la velocidad de giro de cada uno de los motores del chasis, es necesario conocer su velocidad o frecuencia angular.

Si bien existen varias opciones para implementar sensores que midan este parámetro, esta parte del diseño está limitado fundamentalmente por la necesidad de reutilizar el chasis 4WD Rover. El chasis tiene el siguiente diseño:



Descripción de la solución propuesta

Los sensores de velocidad de giro, deberán situarse en el habitáculo que ocupan los motores, el cuál forma una estructura cerrada que no permite añadir accesorios voluminosos. Cada motor está acoplado en una caja reductora, la cual dispone de un eje doble que permite añadir un sensor de velocidad.

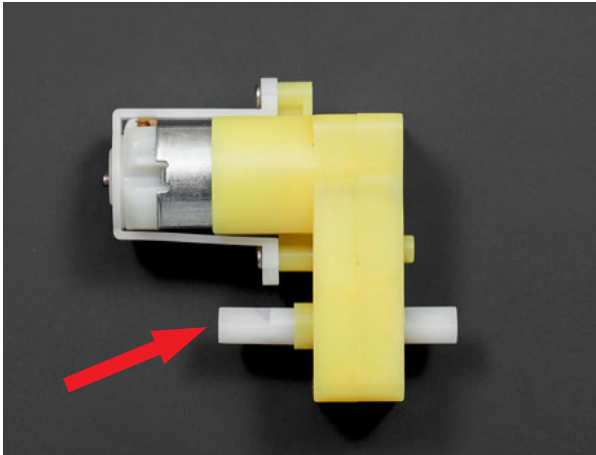


Imagen 17: Caja reductora DFRobot L Shape

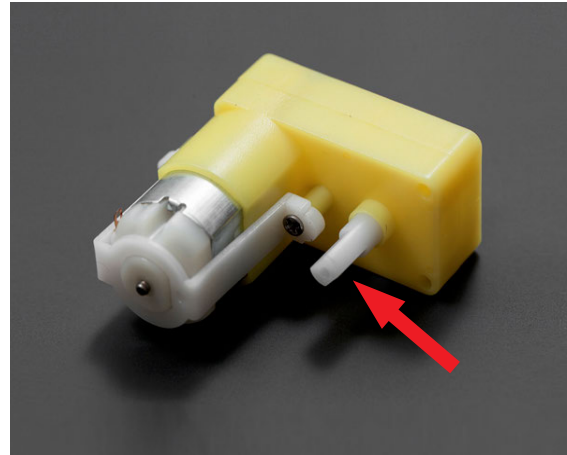


Imagen 18: Caja reductora DFRobot L Shape

Respecto del sensor de velocidad, dadas las limitaciones de espacio, se decidió que el sensor más adecuado era un *encoder* incremental óptico en cuadratura. A continuación, se resume brevemente el principio de funcionamiento de un *encoder* óptico en cuadratura.

Un *encoder* óptico está formado por tres elementos: un disco de código, que en el caso de un *encoder* incremental es simplemente un disco con ranuras que se acopla al eje cuya velocidad o posición va a ser medida, una fotobarrera, formada por un emisor led IR y un fototransistor o fotodiodo, y un acondicionador de señal, que convierte las señales producidas por la fotobarrera en señales digitales.

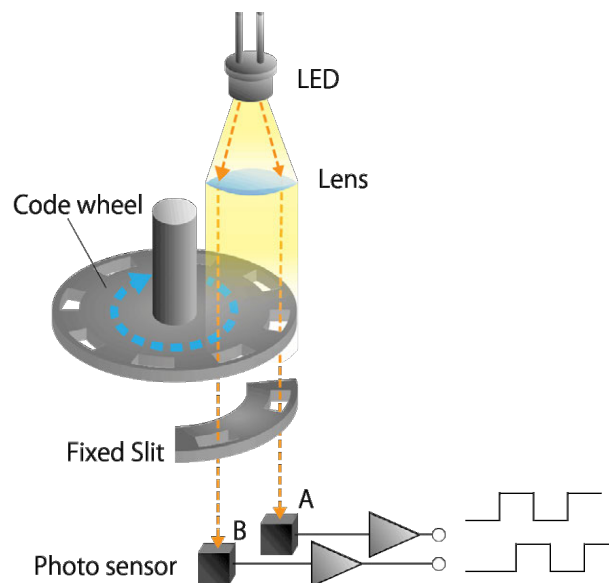


Figura 7: Encoder óptico incremental en cuadratura [20]

Conociendo el número de ranuras de las que dispone la rueda de codificación y la frecuencia de los pulsos generados a la salida del acondicionador se puede calcular la frecuencia angular del eje. Sin

embargo, si bien este sistema de medida permite conocer el valor absoluto de la velocidad angular del motor, no permite conocer el sentido de giro. Para poder implementar un control en lazo cerrado de la velocidad angular que sea estable, es necesario que el valor de velocidad incluya el sentido de giro, es decir, que pueda tomar valores positivos y negativos.

Para lograr esto, se añade una segunda fotobarrera, que incluye un nuevo emisor IR y otro fototransistor, junto con su respectivo acondicionador. Esto proporciona al *encoder* dos canales de salida: canal A, y canal B.

La nueva fotobarrera se sitúa de forma que las señales generadas por el *encoder* estén en cuadratura, esto es, que la señal del canal A este desfasada 90° respecto de la señal del canal B. De esta forma, tomando como referencia para la velocidad la señal del canal A, si durante el flanco de bajada del canal A, el canal B está a nivel alto, significa que el disco del *encoder* está girando en una dirección determinada. Si por el contrario, en el flanco de bajada del canal A, el canal B está a nivel alto, el disco del *encoder* está girando en dirección contraria.

Una vez aclarado esto, se abordará el diseño. Por las limitaciones de espacio, se decidió usar una fotobarrera que incluyera dos canales (dos fototransistores), y que fuera de montaje superficial. El modelo elegido por ser el más económico disponible y que presentaba los dos requisitos mencionados, fue el TCUT1600X01 del fabricante Vishay Semiconductors.

Los esquemas completos se pueden consultar en el apéndice A, pero por comodidad, el circuito acondicionador de señal se presenta a continuación:

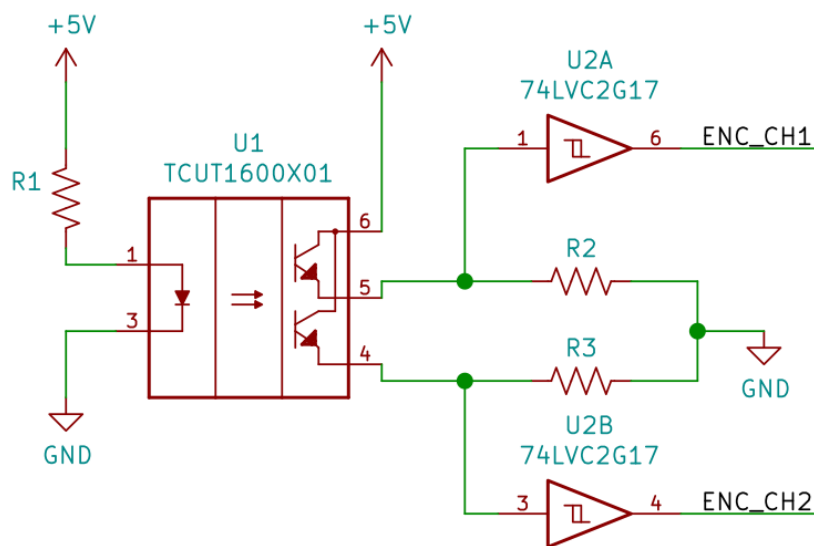


Figura 8: Acondicionador de señal, encoder_board

La fotobarrera tiene un solo led IR y dos fototransistores. Los fototransistores operarán en la zona de saturación, de forma que cuando la luz pasa por una de las ranuras del *encoder*, el fototransistor entra en conducción, produciendo que la tensión entre el emisor y GND sea cercana a los 5V. De forma análoga, cuando el disco del *encoder* bloquea la luz, el transistor permanece en corte, haciendo que la tensión entre el emisor y GND sea de 0V.

Descripción de la solución propuesta

Para elegir el valor de R1, se fija que la corriente de operación del emisor IR sea de 10mA. Se elige este valor para garantizar cierta inmunidad frente al ruido que pueda acoplarse al acondicionador del fototransistor, ya que una corriente menor requeriría aumentar el valor de las resistencias R2 y R3. Así pues, tomando la función de transferencia del diodo del *datasheet* del TCUT1600X01 [21].

$$V_f(10\text{mA}) \simeq 1,1\text{V} \rightarrow R_1 = \frac{V_{CC} - V_f}{I_f} = \frac{5\text{V} - 1,1\text{V}}{10\text{mA}} = 390\ \Omega \rightarrow R_1 = 300\ \Omega (\text{valor normalizado } 5\%)$$

Con esa corriente de diodo fijada, utilizando la curva que relaciona la corriente del diodo con la corriente de colector, se obtiene:

$$I_c(I_f) = I_c(10\text{mA}) = 1\text{mA} \rightarrow R_2 = R_3 = \frac{V_{CC} - V_{CESat}}{I_c} = \frac{5\text{V} - 0,4\text{V}}{1\text{mA}} = 4,6\text{k}\Omega \rightarrow R_2 = R_3 = 4,7\text{k}\Omega (\text{valor normalizado } 5\% \text{ tolerancia})$$

Todas las resistencias serán de 1/8W, y encapsulado SMD 0805 (para facilitar el montaje manual).

A la salida de cada fototransistor, conectado a su emisor, se sitúa un *schmitt trigger*, que tiene dos funciones: evitar que la lógica de entrada del microcontrolador entre en estado meta-estable por recibir una señal fuera de los niveles lógicos TTL, y proporcionar una señal de baja impedancia que sea inmune al ruido generado por los motores. El *schmitt trigger* elegido es el 74LVC2G17, porque sus niveles lógicos son compatibles los del Atmega2560, tiene dos canales, encapsulado smd SOT-23-6 y es el más económico.

La placa del *encoder* se conectará a la placa madre utilizando una manguera de 4 hilos, la cual transportará las tensiones de alimentación: Vcc, Gnd, y las señales correspondientes al canal A y al canal B. La PCB diseñada en KiCad se muestra en las siguientes figuras:

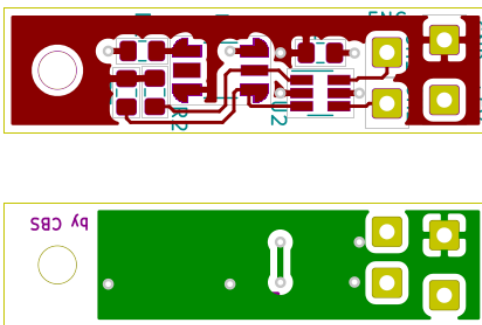


Imagen 19: Capas top y bottom de la PCB

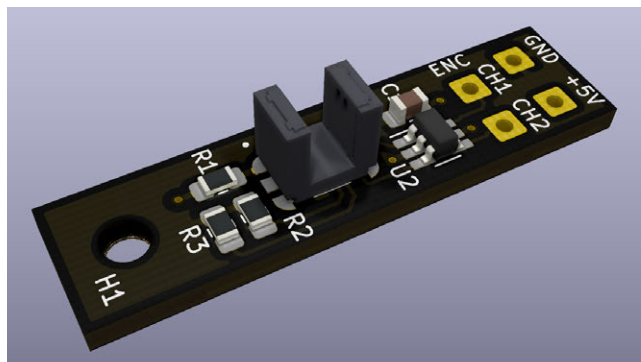


Imagen 20: Modelo 3D de la encoder_board

Por último, para finalizar el desarrollo del hardware del *encoder*, se diseñan dos piezas para su impresión 3D: el disco de código del *encoder*, y el soporte de la PCB.

Para el diseño del disco de código se debe tener en cuenta el espacio que hay en el habitáculo de los motores, quedando su diámetro limitado a 23,6mm. El número de ranuras máximo está limitado por las capacidades de la tecnología FDM de impresión 3D, el cual, después de algunas iteraciones, quedó fijado a 25 ranuras.

En cuanto al soporte de la PCB, se utilizan los tornillos que unen la caja de reducción al chasis para sujetarlo. Tanto el diseño del disco del *encoder* y del soporte se realizaron utilizando los esquemas mecánicos de la caja de reducción, sin embargo, presentar estos detalles aquí no aporta mucho. Por tanto, solo se presentarán los resultados.

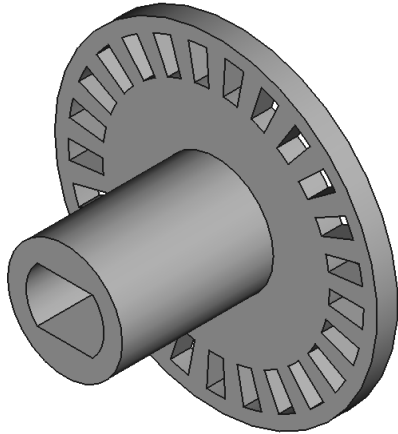


Imagen 21: Disco de código

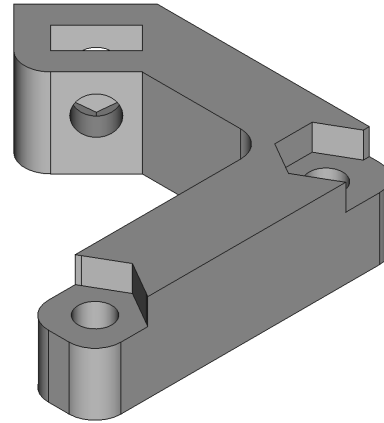


Imagen 22: Soporte de la PCB

El ensamblado virtual de estas piezas junto con la caja reductora y la PCB se observan en la siguiente figura:

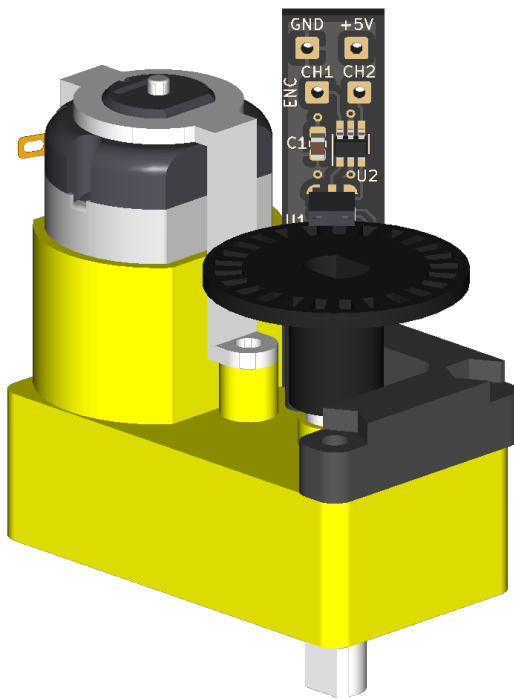


Imagen 23: Ensamblado virtual del encoder

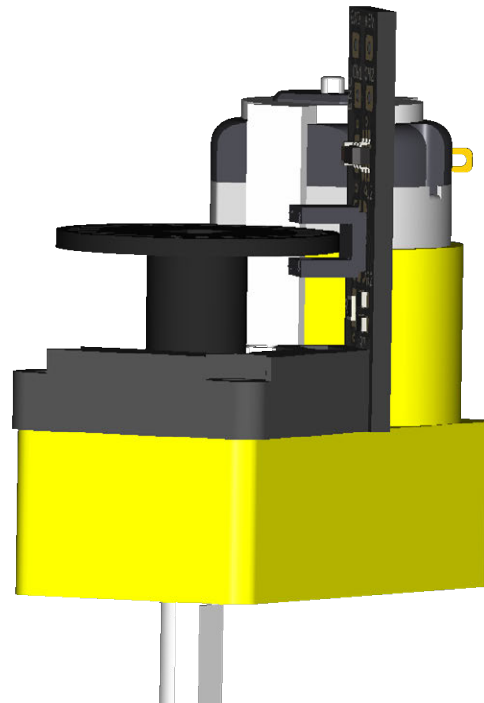


Imagen 24: Ensamblado virtual del encoder

En el apéndice B se pueden consultar los detalles del ensamblado de las piezas en el chasis. En los archivos que acompañan a este proyecto, las piezas están archivadas bajo los nombres: *rueda_encoder_CPR-25_v3.FCStd* y *soporte_lateral_encoder_TCUT1600X01_REV_A.FCStd*.

4.1.2 Nivel 1 – Placa de potencia (Power_board)

Las especificaciones que satisface son: EH-7, EH-11, EH-12, EH-13.

En la placa de potencia se sitúan dos bloques o subsistemas: la *unidad de supervisión de la batería* o *BMU*, (*Battery Managemet Unit*), y los cuatro *drivers* que controlan los motores del chasis.

4.1.2.1 Diseño de la unidad de supervisión de la batería

Para poder medir el estado de carga de la batería, se requiere conocer al menos dos variables: carga actual de la batería, en julios o amperios-hora, y carga total consumida por el robot, la cual es la integral de la corriente instantánea que ha abandonado la batería desde que se conectó el robot.

Entonces, el estado de carga de la batería (SOC), esto es, la cantidad de energía restante almacenada dentro de la batería, expresada en tanto por ciento, se calcula como:

$$DCR = \int_0^t i_{BAT} dt$$

donde: DCR: registro del contador de descarga (*Discharge Count Register*).

$$RC = IC - DCR$$

IC: capacidad inicial (*Initial Capacity*)
RC: capacidad restante (*Remaining Capacity*)

$$SOC (\%) = \frac{RC}{FCC} \cdot 100$$

FCC: capacidad total a plena carga (*Full Charge Capacity*)
SOC: estado de carga, (*State Of Charge*)

Para obtener el parámetro DCR, se debe disponer de un sistema que muestree la corriente instantánea que el robot está consumiendo y acumule este valor. El valor de capacidad inicial, IC, es la cantidad de energía que hay en la batería justo tras conectar la alimentación del sistema. Idealmente, si la batería acaba de cargarse, IC será igual a la capacidad total a plena carga, FCC, de la batería.

Este robot no implementa un cargador integrado, por lo que el usuario deberá utilizar un cargador externo. Esto implica, que la unidad de supervisión de la batería no puede conocer con exactitud cuándo se ha cargado la batería, y cuanta carga hay en la misma. Evidentemente se podría implementar una solución que “obligue” al usuario a informar de que la batería acaba de ser cargada. Sin embargo, incluso en este caso, la cantidad de energía acumulada en la batería podría ser inferior a la FCC, debido al propio cargador o al envejecimiento de las celdas.

Así pues, para estimar la carga inicial de la batería, IC, se debe recurrir al perfil de descarga de la batería. La curva o perfil de descarga de la batería relaciona la tensión en los bornes de la batería con el estado de carga, SOC. Es dependiente de la tecnología de celdas que se use, la temperatura, y sobre todo, única para cada batería. Por tanto, para realizar la estimación de la IC a partir de esta, antes se debe trazar el perfil de descarga para esa batería.

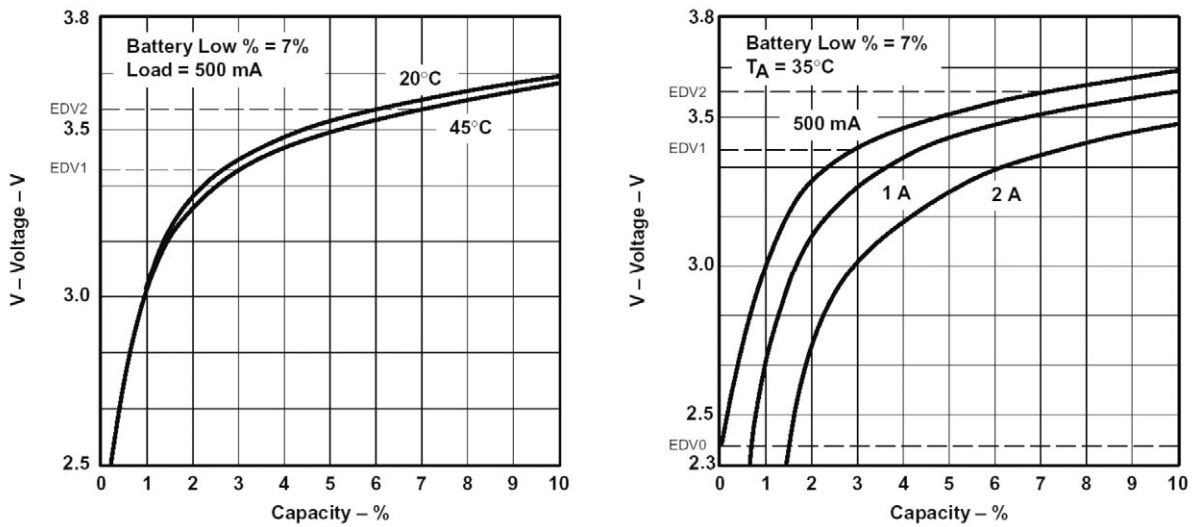


Figura 9: Curvas de descarga de una batería para varias temperaturas, y varias condiciones de carga. [30]

Una cuestión que puede surgir es: si es posible obtener el SOC a partir de la curva de descarga de la batería, utilizando solo la tensión de la batería, ¿para qué complicar el sistema midiendo la corriente y acumulando este valor?. La respuesta es que es posible, y de hecho, esta forma de estimar el estado de carga de la batería se ha usado desde los albores de la electrónica. Sin embargo, el problema que presenta es la inexactitud.

Todas las baterías tienen una resistencia interna, modelada normalmente como una resistencia en serie con una celda de tensión. Así pues, la tensión de la batería no solamente depende del SOC, también depende de la corriente que este circulando por la misma. El perfil de descarga de la batería, normalmente, se obtiene cuando la corriente a través de la batería es nula, o en todo caso, constante.

Entonces, si la batería está entregando una cantidad de corriente elevada, y se usa el perfil de descarga de la batería obtenido con una corriente de descarga nula, para obtener una estimación del SOC, se obtendrá un SOC que no representa el estado de carga real de la batería, y de hecho, puede ser muy inferior al real.

La batería original que llevaban los chasis, heredadas de otro proyecto, era una batería de *Ni-MH* de 6V de 2600mAh. Debido al envejecimiento, la capacidad había disminuido considerablemente. Por ese motivo, se sustituyeron las baterías por baterías *LiPo* de dos elementos y 2600mAh, las cuales proporcionan una tensión de 8,4V a plena carga, y de 6,2V, cuando las celdas llegan al mínimo de carga operativa.

El microcontrolador principal de ARGOS, junto con algunos componentes adicionales, podría implementar las funciones descritas para obtener el estado carga de la batería; sin embargo, dada la ubicuidad de los sistemas electrónicos alimentados por baterías, en el mercado hay numerosas soluciones más adecuadas a precios asequibles.

Descripción de la solución propuesta

Los requisitos que debe cumplir el módulo que se encargue de supervisar el estado de la batería son:

- Soporte de múltiples tecnologías de baterías (*NiHM*, *LiPo*, etc.)
- Sistema de muestreo de la corriente y cálculo del *DRC*.
- Sistema de medida de la tensión de la batería.
- Cálculo del *IC* integrado, mediante perfil de descarga de la batería.

Estas características las encontramos en el circuito integrado *BQ34110* del fabricante *Texas Instruments*, [28]. Este integrado implementa un pequeño microcontrolador y acondicionadores de señal, para realizar el muestro de corriente mediante una resistencia de sensado, medición de tensión de la batería y estimación del estado de la batería. Para comunicarse con el microcontrolador principal dispone de una interfaz *I2C*.

El diseño de la unidad de supervisión de la batería se basa en la aplicación típica presentada en el *datasheet* [28]. El esquema completo se recoge en el Anexo A, no obstante, aquí se reproducirá el esquema de la unidad de supervisión de la batería.

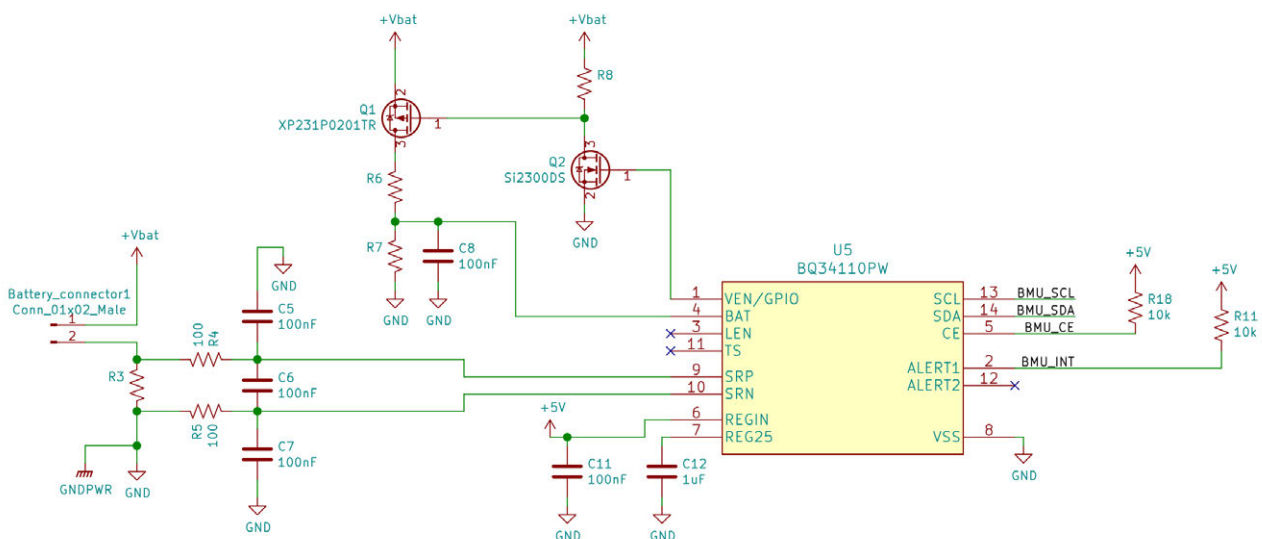


Figura 10: Esquema de la BMU

El condensador C_{11} es un condensador de desacople para el pin de alimentación del regulador interno de tensión. El condensador C_{12} es para la estabilización del regulador interno de 2,5V. Los valores de C_{11} y C_{12} son los recomendados por el fabricante en el *datasheet*.

Las resistencias R_{18} y R_{11} son resistencias de *pull-up* para el *chip select* del bus *I2C* y para el pin de interrupción *ALERT1*, respectivamente. Ambos pines son activos a nivel bajo, y el valor de las resistencias de *pull-up* se fija a 10k.

Para medir la corriente se utiliza la resistencia R_3 , la cual está en serie con la batería. Las resistencias R_4 y R_5 , junto con los condensadores C_5 y C_7 forman un filtro cuyo objetivo es limpiar el ruido en modo común y en modo diferencial de la señal de sensado de R_3 , a la entrada de los pines

SRP y SNP. Los valores de R_4 , R_5 , C_5 , C_6 , C_7 son los recomendados por el fabricante en la aplicación típica [29].

El valor de R_3 se debe elegir en función de la corriente máxima que vaya a consumir el robot, y del rango máximo de tensión que aceptan las entradas SRP y SNP. Según el fabricante, el rango de tensión máxima entre las entradas SRP y SNP es de 125mV. En cuanto a la corriente máxima, en el apartado dedicado al diseño de la placa madre se expone el consumo de cada placa, Tabla 8, no obstante, aquí se adelanta que el consumo de corriente pico de ARGOS será de 11,8A. En base a estos datos, el valor de R_3 es:

$$R_{SENSE} = R_3 = \frac{|VSR|}{I_{PBAT}} = \frac{125 \text{ mV}}{11,8 \text{ A}} = 10,59 \text{ m}\Omega$$

$$P_{MAX_R_{SENSE}} = P_{MAX_R_3} = |VSR| \cdot I_{PBAT} = 1,475 \text{ W}$$

$$R_3 = 10 \text{ m}\Omega \pm 1\% 75 \text{ ppm}/^\circ\text{C}, 2 \text{ W} \text{ (normalizando a la serie E96, 1\% de tolerancia)}$$

Las resistencias R_6 y R_7 forman el divisor de tensión que se usará para medir la tensión entre los terminales de la batería. Para evitar que el divisor de tensión siempre esté conectado a los terminales de la batería consumiendo corriente, el BQ34110 dispone de una función que permite habilitar el divisor de tensión solo cuando se va a realizar la medida. Para implementar esta función se utilizan los transistores Q_1 y Q_2 .

Cuando se va a realizar la medida, el BQ34110 pone su pin VEN a nivel alto, esto es, 5V. Como la tensión en la puerta de Q_2 supera la tensión umbral de la puerta, se pone en conducción, operando en la zona óhmica, y haciendo que la tensión en la puerta de Q_1 caiga hasta prácticamente 0V. En ese momento, la tensión en puerta de Q_1 supera la tensión umbral de puerta, llevándolo a conducción en la zona óhmica, quedando establecido un camino de baja resistencia entre V_{BAT} y el divisor de tensión.

En cuanto a las resistencias del divisor de tensión, R_6 y R_7 , sus valores deben escogerse de forma que la tensión de entrada en el pin V_O sea de máximo 900mV, según el *datasheet* [28]. Además, sus valores deben elegirse de forma que el error de ratio entre ellas sea mínimo. Utilizando la fórmula del divisor de tensión, si consideramos que el $R_{DS(ON)}$ de Q_1 es nulo:

$$V_O = V_{BAT} \cdot \frac{R_7}{R_7 + R_6} \rightarrow R_6 = R_7 \cdot \frac{V_{BAT} + V_O}{V_O} \rightarrow \frac{R_6}{R_7} = \frac{V_{BAT} + V_O}{V_O} = \frac{8,4 \text{ V} - 900 \text{ mV}}{900 \text{ mV}} = 8,33$$

Utilizando *resrat*, un programa para seleccionar resistores que garantiza que el ratio de error entre ellos sea el menor posible, para la serie E96 de tolerancia del 1%, los resistores elegidos con un ratio de error de -0,36% son:

$$R_7 = 16,5 \text{ k}\Omega \pm 1\%$$

$$R_8 = 137 \text{ k}\Omega \pm 1\%$$

Para la selección del transistor Q_1 , se debe tener en cuenta el error introducido por la resistencia de drenador a fuente, R_{DS} :

$$V_{o_R} = V_{BAT} \cdot \frac{R_7}{R_6 + R_7 + R_{DS(ON)MAX}}$$

$$V_{o_I} = V_{BAT} \cdot \frac{R_7}{R_6 + R_7}$$

$$\varepsilon = \frac{V_{o_R} - V_{o_I}}{V_{o_I}}$$

Despejando $R_{DS(ON)MAX}$, se obtiene:

$$R_{DS(ON)MAX} = (R_6 + R_7) \cdot \frac{-\varepsilon}{\varepsilon + 1}$$

Para un error del -0,01%, $\varepsilon = -0,0001$, el transistor Q_1 deberá tener una $R_{DS(ON)MAX}$ de:

$$R_{DS(ON)MAX} = (16,5 \text{ k}\Omega + 137 \text{ K}\Omega) \cdot \frac{-(-0,0001)}{-0,0001 + 1} = 15,3\Omega$$

El transistor Q_1 se seleccionó porque su $R_{DS(ON)}$ es $3,5\Omega$ para la tensión de polarización de puerta, $V_{GS} = -5\text{V}$ [30]. El transistor Q_2 se seleccionó por su baja tensión umbral de puerta y por su precio, además de por simplificar el stock de componentes, ya que se utiliza en varias de las placas de este diseño.

El valor de la resistencia R_8 se eligió para garantizar que Q_1 operará en la zona óhmica. Utilizando la curva de salida Q_1 recogida en el *datasheet* [31] y la recta de carga, para una tensión mínima de batería de 5V, se obtiene que $R_{DS(ON)} = 80\text{m}\Omega$, por lo que, la tensión en la puerta de Q_2 cuando Q_1 este en conducción es de $4,2\mu\text{V}$, lo que garantiza que Q_1 está en conducción.

Por último, el condensador C_8 forma un filtro paso-bajo con las resistencias R_6 y R_7 , cuya frecuencia de corte es de unos 108Hz. Su función es reducir el ruido de la señal de tensión de la batería.

4.1.2.2 Diseño del *driver* de control de motores

Para comenzar el diseño, lo primero que se debe establecer son las características de los motores a controlar. Según el fabricante, DFRobot, las características del conjunto caja reductora-motor [22] son las siguientes:

- Rango de voltaje operativo: 3V a 7,5V
- Voltaje nominal: 6V
- Máxima corriente sin carga (3V): 140mA
- Máxima corriente sin carga (6V): 170mA
- Velocidad sin carga (3V): 90rpm
- Velocidad sin carga (6V): 160rpm
- Máxima corriente estática: 2,8A

Para controlar el sentido de giro se utiliza un puente en H, y para controlar la tensión aplicada, se usa control PWM. Un puente en H está formado por cuatro transistores, normalmente transistores bipolares o mosfet, los cuales operarán entre la zona de conducción y corte, comportándose como interruptores.

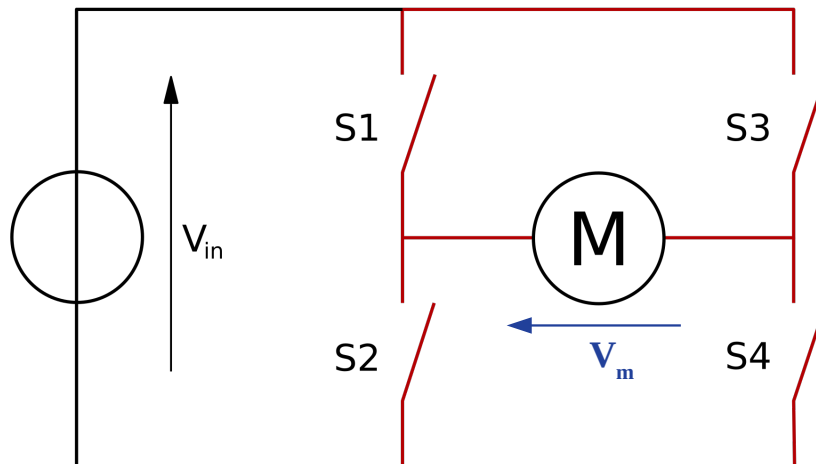


Figura 11: Puente en H [23]

El funcionamiento es el siguiente: si los transistores S1 y S3 o S2 y S4 están abiertos al mismo tiempo, la tensión entre los terminales del motor es 0V, por lo que no se produce movimiento. Si se activan S1 y S4, mientras S2 y S3 permanecen en corte, la tensión en los terminales del motor, V_m , será igual a V_{in} , por lo que el motor girará en una dirección determinada. De forma análoga, si se activan S3 y S2, mientras S1 y S4 permanecen en corte, la tensión entre los terminales del motor, V_m , será igual a $-V_{in}$, por lo que el motor girará en la dirección contraria.

Descripción de la solución propuesta

Para controlar la tensión aplicada, se utiliza una técnica denominada *modulación por ancho de pulso* (PWM), en la que se utiliza una señal cuadrada de ciclo de trabajo variable para controlar el tiempo que cada transistor permanece en conducción.

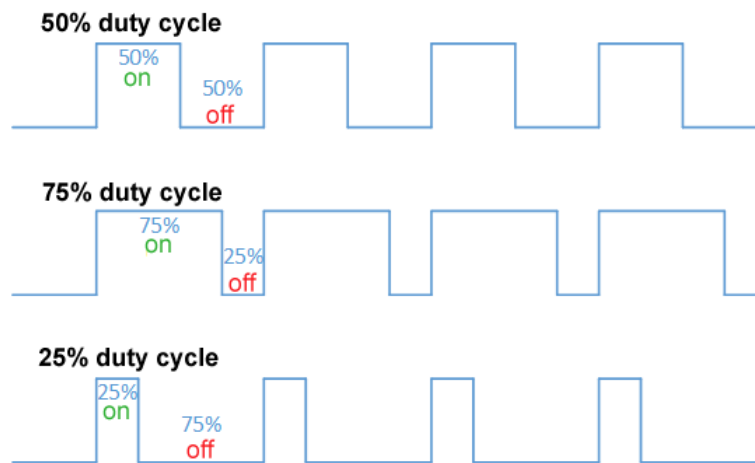


Figura 12: Modulación por ancho de pulso (PWM) [24]

Entonces, la tensión aplicada en los terminales del motor será igual al valor medio de la señal que controla la conmutación de los transistores. Obviando las tensiones de saturación (para los transistores bipolares) o la resistencia de drenador a fuente (para los MOSFET), la tensión media aplicada en los terminales del motor es:

$$V_m = V_{IN} \cdot \frac{D(\%)}{100} \quad (1) \quad \text{donde: } V_{IN}: \text{ es la tensión de alimentación del puente en H.}$$

V_m : es la tensión en los terminales del motor.

D : es el ciclo de trabajo en tantos por cien de la señal PWM.

La frecuencia de conmutación de la señal PWM estará limitada por el dispositivo de conmutación seleccionado.

Ya que cada motor debe ser controlado de forma independiente, se precisarán cuatro puentes en H. Debido a que el consumo de los motores es elevado, la alimentación de los puentes en H irá conectada directamente al bus de la batería. Dado que la tensión máxima de la batería es de 8,4V y la tensión máxima de los motores según el fabricante es de 7,5V, aunque habrá ciertas pérdidas de tensión en los elementos de conmutación del puente, deberá limitarse el ciclo de trabajo de la señal PWM a un máximo del 89%, para no dañar los motores.

Por otro lado, con el objetivo de reducir al máximo el tamaño de la electrónica y permitir apilar en niveles las PCB, se decidió buscar un *driver* en encapsulado SMD, que contuviera el puente en H y algún sistema para el sensado de la corriente (especificación EH-13).

El *driver* seleccionado es el *DRV8876* del fabricante *Texas Instruments*, el cual presenta las siguientes características [25]:

- Tensión de alimentación (V_{VM}): 4,5V a 37V.
- Corriente de salida máxima ($I_{OUT\ PEAK}$): 3,5A

- Resistencia drenador a fuente de cada transistor ($R_{DS(ON)}$): 350m Ω
- Limitación de corriente de salida, con detección de fallo por sobrecorriente y sobretemperatura.
- Sensado de corriente de salida.
- Modo de ultra bajo consumo.

La corriente de salida máxima está sujeta al diseño térmico de la PCB, pero suponiendo que este se realiza correctamente, el margen de seguridad respecto a la corriente estática de los motores es del 20%. No obstante, los motores solo alcanzarán el consumo dado por la corriente estática en el arranque o si están bloqueados.

Aprovechando la protección por sobrecorriente incluida en el *driver*, se establecerá un límite de seguridad del 25% respecto $I_{OUT\ PEAK}$, quedando la corriente máxima que los motores pueden consumir fijada a 2,625A.

El esquema simplificado de conexión con el microcontrolador se muestra a continuación:

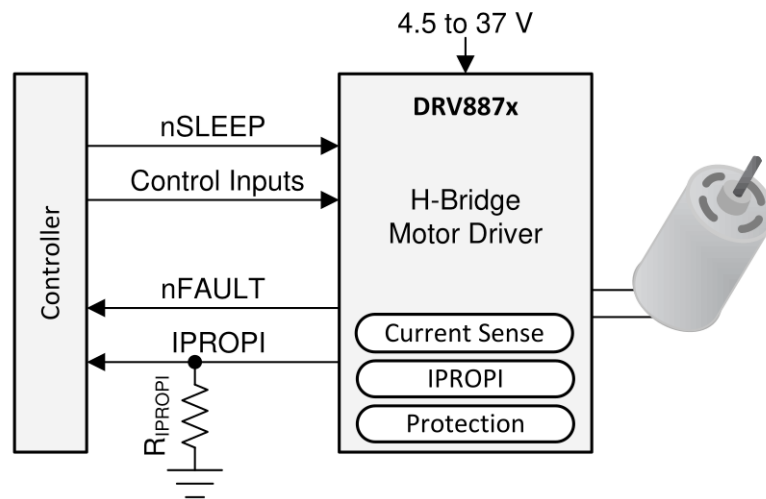


Figura 13: Esquema de conexión con el microcontrolador, DRV8876

La entrada *nSLEEP*, activa a nivel bajo, permite desactivar el *driver*, poniéndolo en un estado de bajo consumo, lo que cumple con la especificación EH-11.

Además, la salida *nFAULT*, permite al microcontrolador detectar cuando se activado la protección por sobrecorriente o sobretemperatura del *driver* . La salida *IPROPI* proporciona una tensión proporcional a la corriente consumida por el motor; cumpliendo con la especificación EH-13.

Respecto al diseño, el *datasheet* [25] presenta una aplicación típica y unos valores recomendados para los componentes. Los esquemas completos se pueden consultar en el Anexo A, no obstante, aquí se reproduce el diseño para el *driver* del motor 1, que se tomará como referencia:

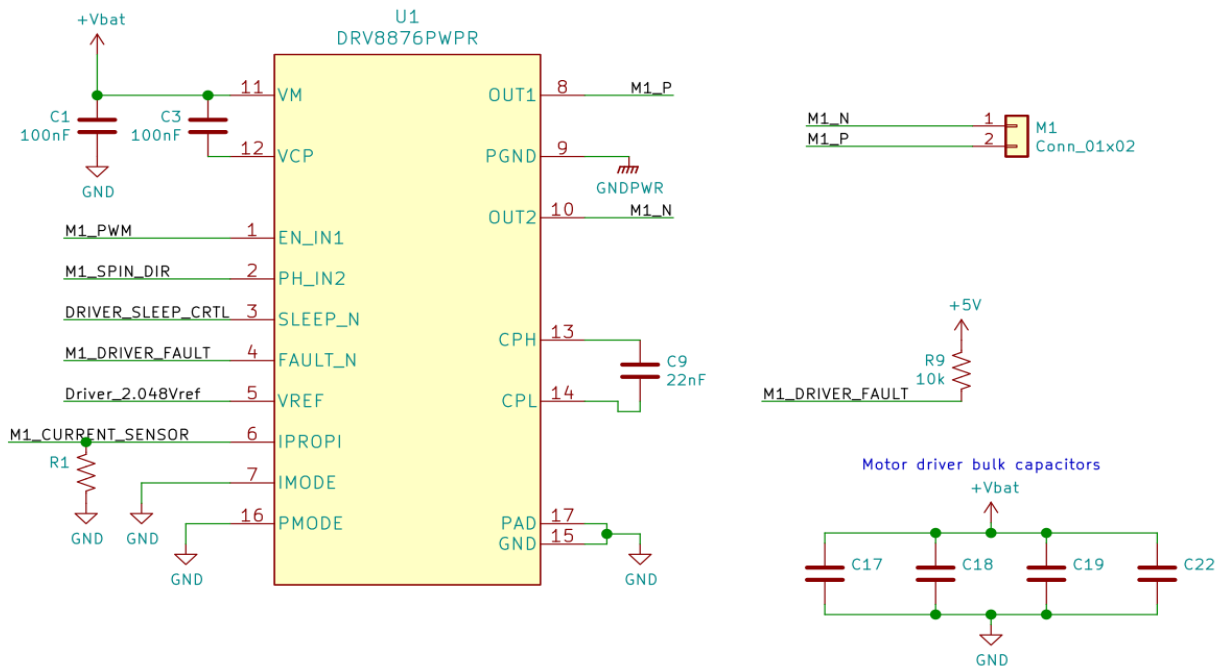


Figura 14: Driver motor 1

COMPONENT	PIN 1	PIN 2	RECOMMENDED
C _{VM1}	VM	GND	0.1- μ F, low ESR ceramic capacitor, VM-rated.
C _{VM2}	VM	GND	Bulk Capacitance, VM-rated.
C _{VCP}	VCP	VM	X5R or X7R, 100-nF, 16-V ceramic capacitor
C _{FLY}	CPH	CPL	X5R or X7R, 22-nF, VM-rated ceramic capacitor
R _{IMODE}	IMODE	GND	See Current Regulation .
R _{PMODE}	PMODE	GND	See Control Modes .
R _{nFAULT}	VCC	nFAULT	Pullup resistor, I _{OD} \leq 5-mA
R _{IPROPI}	IPROPI	GND	See Current Sensing .

Figura 15: Valores recomendados

La conexión del pin *PMODE* permite seleccionar el modo de control del puente en H entre tres modos: *PH/EN*, *PWM* y *Independent Half-Bridge*. Se seleccionó el modo *PH/EN*, que activa los transistores S2 y S3 del puente cuando la señal PWM está a nivel bajo. Esto proporciona un camino de baja impedancia por el que la bobina del motor puede descargar su corriente, disminuyendo el ruido eléctrico generado, y las chispas en el conmutador del rotor.

Para seleccionar el modo *PH/EN*, el pin *PMODE* debe conectarse a nivel bajo, esto es, *GND*.

nSLEEP	EN	PH	OUT1	OUT2	DESCRIPTION
0	X	X	Hi-Z	Hi-Z	Sleep, (H-Bridge Hi-Z)
1	0	X	L	L	Brake, (Low-Side Slow Decay)
1	1	0	L	H	Reverse (OUT2 \rightarrow OUT1)
1	1	1	H	L	Forward (OUT1 \rightarrow OUT2)

Figura 16: Control en modo *PH/EN*, DRV8876

La conexión del pin *IMODE* permiten elegir la respuesta del *driver* cuando se produce un evento de sobrecorriente. Hay cuatro opciones:

IMODE STATE		IMODE FUNCTION		nFAULT Response
		Current Chopping Mode	Overcurrent Response	
Quad-Level 1	R _{IMODE} = GND	Fixed Off-Time	Automatic Retry	Overcurrent Only
Quad-Level 2	R _{IMODE} = 20kΩ to GND	Cycle-By-Cycle	Automatic Retry	Current Chopping and Overcurrent
Quad-Level 3	R _{IMODE} = 62kΩ to GND	Cycle-By-Cycle	Outputs Latched Off	Current Chopping and Overcurrent
Quad-Level 4	R _{IMODE} = Hi-Z	Fixed Off-Time	Outputs Latched Off	Overcurrent Only

Figura 17: Opciones de configuración de IMODE, DRV8876

La configuración elegida es la primera que aparece en la Figura 17, con el pin *IMODE* conectado a *GND*. Bajo esta configuración, cuando se produce un evento de sobrecorriente, el *driver* pondrá los transistores *S1* y *S3* en corte, y activará los transistores *S2* y *S4* durante un tiempo fijo de 25µs, proporcionando un camino de baja impedancia al motor. Cuando el tiempo haya expirado, restaurará el valor anterior que tenían las salidas. Se elige este modo porque mantiene la corriente por debajo del umbral seleccionado sin requerir la intervención del microcontrolador.

Respecto a la selección de componentes, se utilizarán los valores recomendados por el fabricante para los condensadores de desacople y para el condensador de la bomba de carga (conectado entre *CPL* y *CPH*). Los valores de las resistencias de *pull-up* y *pull-down* se fijan a 10kΩ.

La tensión generada en el pin *IPROPI*, es proporcional a la corriente que atraviesa los transistores activos del puente, de acuerdo con la siguiente ecuación:

$$V_{PROPI} = (I_{LS1} + I_{LS2}) \cdot A_{IPROPI} \cdot R_{IPROPI} \quad (2) \quad \text{donde: } I_{LS1}: \text{ es la corriente que circula por S1 o S3.}$$

I_{LS2}: es corriente que circula por *S2* o *S4*.

A_{IPROPI}: es la ganancia de corriente, fijada a 1000µA/A.

R_{IPROPI}: es la resistencia conectada entre el pin *IPROPI* y *GND*.

Para el sensado de corriente y la protección de sobretensión se debe proporcionar una tensión de referencia en el pin *V_{REF}*. Cuando la tensión *V_{PROPI}* supere la tensión de *V_{REF}* se activará la protección de sobrecorriente, por tanto *V_{REF}* establecerá la máxima tensión generada por el pin *I_{PROPI}*. Así pues, se utilizará el mismo valor de tensión de referencia que se usa en el ADC del microcontrolador principal de la placa madre, fijando *V_{REF}* a 2.048V.

Ahora que el valor de tensión del pin *V_{REF}* es conocido, se selecciona el valor de la resistencia *R_{IPROPI}* en función de la corriente máxima que activa la protección por sobrecorriente: *I_{TRIP}* = 2,625A

$$R_{IPROPI} = \frac{V_{REF}}{I_{TRIP} \cdot A_{IPROPI}} = \frac{2,048 \text{ V}}{2,625 \text{ A} \cdot 1000 \mu\text{A/A}} = 780,19 \Omega \rightarrow$$

$$R_1 = R_2 = R_{12} = R_{13} = R_{IPROPI} = 787 \Omega \pm 1\% \quad (\text{normalizando a la serie E96, 1\% de tolerancia})$$

El último componente que queda por seleccionar es el condensador *C_{BULK}*. Cualquier elemento conductor, es decir, cualquier cable o pista en un circuito, tiene una determinada resistencia y una inductancia parásita. Estos valores suelen ser tan pequeños que en la mayoría de aplicaciones ni se consideran.

En este caso, debido al propio principio del control PWM y a la operación del puente en H, se producirán grandes picos de corriente cuando se energice el motor, cada vez que la señal de control PWM este a nivel alto. Estos picos de corriente, que se darán con la misma frecuencia que la señal PWM, generarán oscilaciones transitorias en la tensión de las líneas de alimentación de los *drivers*, debido principalmente a la inductancia parásita de las mismas.

Estas oscilaciones, que son a efectos prácticos ruido eléctrico, se acoplarán a todos los elementos conectados a esas líneas de alimentación, provocando errores en los sensores y otros sistemas del robot. Para evitar esto es, se utiliza el condensador C_{BULK} , el cual debe situarse lo más cerca posible del pin de alimentación del *driver*. Cuando se produzca una gran demanda de corriente, el condensador C_{BULK} se descargará parcialmente, proporcionando la carga necesaria y evitando que se produzca un gran pico de corriente en la pista de alimentación, evitando que se generen las oscilaciones mencionadas.

La selección de este valor no es sencilla porque no hay una guía o referencia. Depende de la inductancia de las pistas, y de la impedancia de carga, parámetros que este caso no son fácilmente estimables. En el *datasheet* se surge la selección empírica, así pues, C_{BULK} se fija a 1000uF.

$$C_{17}=C_{18}=C_{19}=C_{22}=C_{BULK}=1000\mu F$$

Tras las pruebas de este subsistema se comprobó que no se producía ruido cuando todos los motores funcionaban a plena carga, por lo que el valor es suficiente para la aplicación. Debido a las limitaciones de espacio, el condensador elegido tiene una tensión máxima de 10V, por lo que la tensión máxima de la batería queda fijada a 8,4V con un margen de seguridad del 16%.

Por último, antes de finalizar con la presentación del diseño de este subsistema, se debe abordar el diseño térmico. Este *driver* se presenta en un encapsulado de montaje superficial y la disipación del calor residual se realiza a través de un *pad* térmico, situado en su parte inferior. Ese *pad*, que estará soldado a la PCB, junto con el diseño de los planos de cobre conectados a él determinará que la *temperatura en la unión* del componente se mantenga dentro de su rango de operación.

Para realizar un dimensionado adecuado de la potencia que va a disipar el *driver*, se necesita conocer la corriente que van a consumir los motores durante un uso normal. La corriente estática máxima de los motores, proporcionada por el fabricante DFRobot, no representa un consumo en uso normal, puesto que es la corriente que consumirá el motor cuando esté completamente detenido, y por tanto, no haya fuerza contraelectromotriz.

En un motor ideal, si no hay carga, la corriente que circula por el rotor cuando alcanza su régimen permanente es 0A. Cuando se añade alguna carga a ese motor, y por tanto, ejerce un trabajo, la corriente que circula por dicho motor es proporcional al trabajo generado.

Así pues, la única forma de tener una corriente aproximada para dimensionar el *driver* es hallarla experimentalmente. Para ello, se realizarán dos pruebas. La primera consiste en medir la corriente que consumen los motores en vacío, es decir, sin que las ruedas toquen el suelo. La segunda, se cargan 400 gramos de peso sobre el chasis, que simula el peso de la batería y de la electrónica, y

se pone el chasis en el suelo, de forma que la rueda desplace la plataforma; a continuación se mide la corriente consumida por los motores.

Para que la medida sea precisa se utilizarán dos polímetros, uno para medir la corriente (el cual introducirá una resistencia adicional) y otro para medir la tensión en los terminales del motor. El valor de la fuente de alimentación se aumentará hasta que el valor de tensión leído en los terminales del motor coincida el esperado (V_m). Las pruebas se repiten en dos motores distintos, obteniéndose estos resultados:

Corriente consumida con carga (0,4kg)		
V_m (V)	M1 I_{rms} (mA)	M2 I_{rms} (mA)
3	440	653
4,5	710	760
6	800	815

Corriente consumida en vacío		
V_m (V)	M1 I_{rms} (mA)	M2 I_{rms} (mA)
3	247	259
4,5	205	311
6	220	366

Como se puede observar, la corriente consumida en vacío difiere de la corriente especificada por el fabricante del chasis, el motivo más probable es la falta de lubricación de la caja de reducción.

Para calcular la potencia máxima que debe disipar el driver, se toma como referencia la corriente máxima consumida por el motor 2 cuando se utiliza una tensión de 6V y este está bajo carga. Esta corriente es representativa, por ser el máximo que consume un solo motor sometido a una carga equivalente al peso total de la electrónica y la batería, mientras desplaza el chasis.

Además, dado que para implementar el modelo de una plataforma con tracción diferencial siempre habrá al menos dos motores activos, en el uso normal la corriente consumida será menor. No obstante, y se añadirá un factor de seguridad del 50%. Así:

$$I_{OUT_MAX} = 815 \text{ mA} \cdot 1,5 = 1,22 \text{ A}$$

Para calcular la potencia disipada por el *driver*, se calcularán las tres componentes más influyen en esta, señaladas en [26]:

$$P_{VM} = V_{VM} \cdot I_{VM} = V_{BAT} \cdot I_{VM}$$

$$P_{SW} = P_{SW_RISE} + P_{SW_FALL} = \frac{1}{2} \cdot V_{BAT} \cdot I_{OUT_MAX} \cdot f_{PWM} \cdot (t_{RISE} + t_{FALL})$$

$$P_{RDS} = I_{OUT_MAX}^2 \cdot (R_{DS(ON)_HS} + R_{DS(ON)_LS})$$

La potencia total, P_{TOT} , que el *driver* debe disipar es:

$$P_{TOT} = P_{VM} + P_{SW} + P_{RDS} \quad (3) \quad \text{donde: } P_{VM}: \text{ es la potencia disipada por la lógica del } driver.$$

P_{SW} : es la potencia disipada por pérdidas de conmutación.

P_{RDS} : es la potencia disipada por la resistencia de drenador a fuente de los MOSFET que forman el puente en H .

Descripción de la solución propuesta

El valor de corriente en modo activo, I_{VM} , se obtiene del *datasheet*, seleccionado su valor típico de 3mA. Los valores de t_{RISE} y t_{FALL} también se obtienen del *datasheet*, siendo ambos iguales a 150ns. Los valores de la resistencia de drenador a fuente, R_{DS} , se obtienen del *datasheet*, tomando como punto de operación una *temperatura en la unión*, T_J , de 85°C, siendo su valor igual a 430mΩ. Por último, se fija la frecuencia de conmutación de la señal PWM a 15kHz para una operación silenciosa, que son 5kHz por debajo del límite impuesto por el *driver*.

$$P_{VM} = 7,4 V \cdot 3 mA = 22,2 mW$$

$$P_{SW} = P_{SW_RISE} + P_{SW_FALL} = \frac{1}{2} \cdot 7,4 V \cdot 815 mA \cdot 15 kHz \cdot (150 ns + 150 ns) = 13,6 mW$$

$$R_{RDS} = (1,22 A)^2 \cdot (430 m\Omega + 430 m\Omega) = 1,28 W$$

$$P_{TOT} = 22,2 mW + 13,6 mW + 1,28 W = 1,32 W$$

El modelo térmico del *driver* se presenta a continuación:

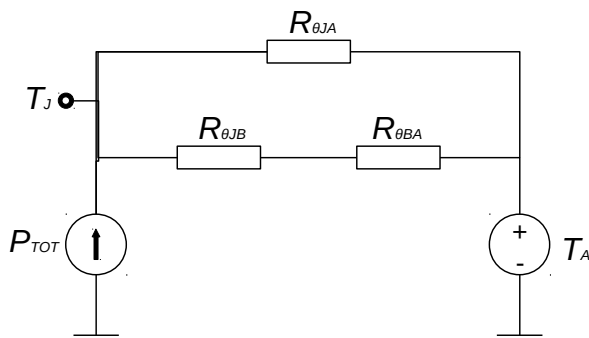


Figura 18: Modelo térmico del driver

donde:

- $R_{\theta JA}$: resistencia térmica de unión a ambiente.
- $R_{\theta JB}$: resistencia térmica de unión a PCB.
- $R_{\theta BA}$: resistencia térmica de PCB a ambiente.
- T_j : temperatura en la unión
- T_a : temperatura ambiente.

Utilizando este modelo, primero se considerará el caso en el que la potencia se disipa únicamente a través del encapsulado del *driver*, esto es, la resistencia térmica $R_{\theta JA}$, la cual es de 44,3°C/W, según el *datasheet*. Se establecerá la temperatura del ambiente a 25°C:

$$T_J = P_{TOT} \cdot R_{\theta JA} + T_A = 1,32 W \cdot 44,3 ^\circ C/W + 25 ^\circ C = 83,5 ^\circ C$$

Esta temperatura en la unión está por debajo de la temperatura máxima de operación en la unión, (150°C), con un margen de seguridad del 43%, por lo que se demuestra que el diseño para las condiciones de uso normal es seguro.

Pese a todo, se debe comprobar cómo se comportaría el diseño para las condiciones menos favorables que pueden presentarse, esto es, cuando se alcance la corriente máxima que activa la protección por sobrecorriente: $I_{TRIP} = 2,625 A$. Con esta corriente, la potencia disipada es:

$$R_{RDS} = (2,625 A)^2 \cdot (430 m\Omega + 430 m\Omega) = 5,92 W$$

$$P_{TOT} = 22,2 mW + 13,6 mW + 5,92 W = 5,96 W$$

Si solos se tiene en cuenta la disipación de potencia a través del encapsulado, la temperatura alcanzada en la unión sería:

$$T_J = P_{TOT} \cdot R_{\theta JA} + T_A = 5,96 \text{ W} \cdot 44,3 \text{ }^\circ\text{C/W} + 25 \text{ }^\circ\text{C} = 289 \text{ }^\circ\text{C}$$

Esta temperatura está muy por encima de la temperatura máxima de operación en la unión (150°C), y produciría que se activara la protección térmica del *driver*. No obstante, en este análisis no se ha tenido en cuenta la disipación de calor a través de la PCB. Si se tiene en cuenta este mecanismo de disipación, la temperatura de la unión es:

$$T_J = P_{TOT} \cdot [R_{\theta JA} \parallel (R_{\theta JB} + R_{\theta BA})] + T_A$$

El valor de la resistencia térmica $R_{\theta BA}$ depende fundamentalmente del diseño de la PCB. Para garantizar que la disipación de calor sea la mejor posible, los *drivers* están dispuestos de forma radial, y cuenta con dos superficies de disipación directamente conectadas al *pad* térmico. La primera, situada en la capa superior, forma un triángulo con un área aproximada de un cuarto del área total de la PCB, y la segunda, es el plano de masa de la capa inferior de la PCB, conectado al *pad* térmico por 15 vías de 8 milésimas de pulgada de diámetro, situadas debajo del *pad* térmico.

El cálculo de la resistencia térmica de PCB a ambiente, $R_{\theta BA}$, no es trivial, y para obtener un valor preciso, se requeriría una simulación de elementos finitos con alguna de las herramientas que existen para este fin. Esos trabajos se salen del alcance de este proyecto, así que usarán los modelos presentados en [27] para obtener una estimación. Para ello, se tendrán en cuenta los siguientes parámetros: el grosor de la PCB es de 1,65mm, el grosor del cobre en las pistas es de 1oz, el grosor del cobre en las vías, dado por el fabricante de la PCB, es 18μm, la PCB tiene forma circular con diámetro de 10cm, y se considera que todo el plano de masa inferior es un disipador. No se tendrá en cuenta la resistencia térmica lateral del cobre. Bajo estas condiciones, el valor de $R_{\theta BA}$ es:

$$R_{\theta JB} = 19,7 \text{ }^\circ\text{C/W}$$

Y la temperatura en la unión es:

$$\begin{aligned} T_J &= P_{TOT} \cdot [R_{\theta JA} \parallel (R_{\theta JB} + R_{\theta BA})] + T_A \\ T_J &= 5,96 \text{ W} \cdot (44,3 \text{ }^\circ\text{C/W} \parallel (20,5 \text{ }^\circ\text{C/W} + 19,7 \text{ }^\circ\text{C/W})) + 25 \text{ }^\circ\text{C} = 5,96 \text{ W} \cdot 21,1 \text{ }^\circ\text{C/W} + 25 \text{ }^\circ\text{C} \\ T_J &= 150,7 \text{ }^\circ\text{C} \end{aligned}$$

Este valor se sale de las condiciones de operación seguras y además, debe tomarse como un valor optimista, dadas las simplificaciones realizadas al calcular $R_{\theta BA}$. No obstante, el aumento de temperatura no se producirá de forma instantánea, pues tanto el *driver* como la placa tienen una capacitancia térmica determinada. Desde que se produzca el evento de sobrecorriente hasta el punto en el que se alcanza una temperatura de operación en la unión fuera de los límites de operación, pasará una cierta cantidad de tiempo.

Estimar la duración de este intervalo requiere conocer la capacitancia térmica o la impedancia térmica del encapsulado del *driver*, y de la PCB. Nuevamente, la estimación de estos valores requeriría una simulación con herramientas especializadas que se sale del alcance de este proyecto.

Sin embargo, en el *datasheet* del *driver* se presenta un ejemplo para el diseño de la PCB, y se expone una simulación de la impedancia térmica para el diseño propuesto. El diseño de la PCB propuesto por el fabricante es menos favorable que el diseño aquí realizado, por lo que se puede asumir que la impedancia térmica real de este diseño será más favorable. A continuación, se muestra la impedancia térmica simulada para el diseño propuesto por el fabricante:

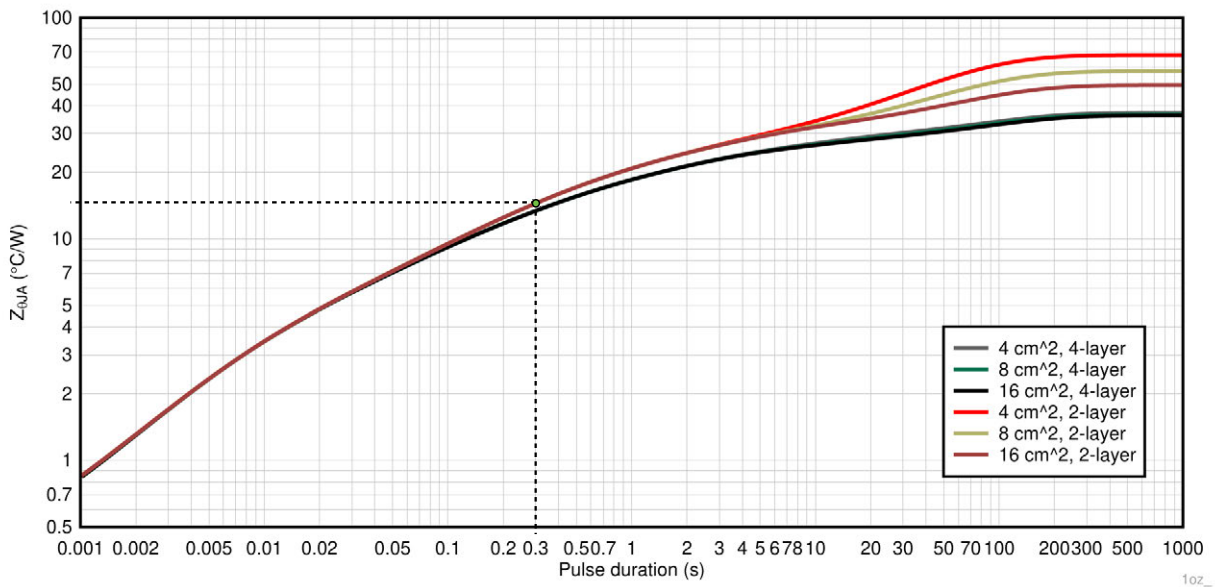


Figura 19: Encapsulado HTSSOP impedancia térmica unión-ambiente para PCB con 1-oz de grosor de cobre

Esta gráfica muestra que la impedancia térmica está dominada por la duración del pulso de potencia, mientras la duración no sea demasiado elevada. Así pues, si la excursión de potencia se produce durante un corto periodo de tiempo, la impedancia térmica unión-ambiente será menor que la resistencia térmica unión-ambiente.

Por tanto, si fijamos una ventana de 0,3 segundos como la duración máxima para un evento de sobrecorriente, lo cual se puede garantizar utilizando la generada por el pin *nFAULT*, la impedancia térmica y la temperatura en una unión será:

$$T_J = P_{TOT} \cdot Z_{\theta JA} + T_A = 5,96 W \cdot 14 ^\circ C / W + 25 ^\circ C = 108 ^\circ C$$

Como se puede observar, esta temperatura está por debajo de la temperatura de unión, con un factor de seguridad del 28%. Por tanto, el controlador del *driver*, deberá vigilar las entradas *nFAULT* y en caso de producirse un evento, comprobar su duración. Si esta fuera mayor de 0,3s, el controlador del *driver* debería desconectarlo e informar al usuario del evento.

4.1.2.3 Conexión con la placa madre

La placa de potencia y la placa madre son el conjunto mínimo que ARGOS requiere para funcionar. Debido a esto, el bus de +5V depende del regulador de tensión de la placa madre, y la placa madre depende de la alimentación de la placa de potencia, la cual proporciona a esta y al resto de placas una conexión con el bus de la batería, V_{BAT} .

Además de estas conexiones, la placa de potencia dispone de cuatro líneas para indicar el fallo en alguno de los *drivers*; cuatro líneas para el sensado de la corriente consumida por los *drivers*; ocho líneas de control para la dirección y ciclo de trabajo de los *drivers*; tres líneas para el bus I2C; una línea para la interrupción por parte de la unidad de supervisión de la batería y una línea para la habilitación de los *drivers*. A parte de estas, hay dos líneas de masa, una dedicada al retorno de masa de la placa madre (GND), y otra destinada al retorno de masa de las placas del sensor de rango y rumbo (PWRGND).

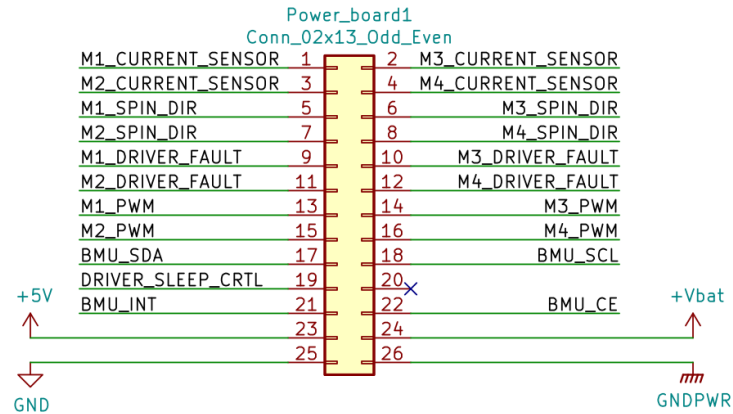


Figura 20: Conexión entre la placa de potencia y la placa madre

Todas estas líneas se conectan a la placa madre a través de un *socket* de 26 pines situado en el centro de la placa, el cual encaja con un *header* de montaje superficial situado en la parte inferior de la placa madre.

4.1.2.4 Diseño de la PCB

A continuación, se muestran las capas *top* y *bottom* de la PCB. En ellas se puede observar la disposición de los planos de masa que forma la superficie de disipación.

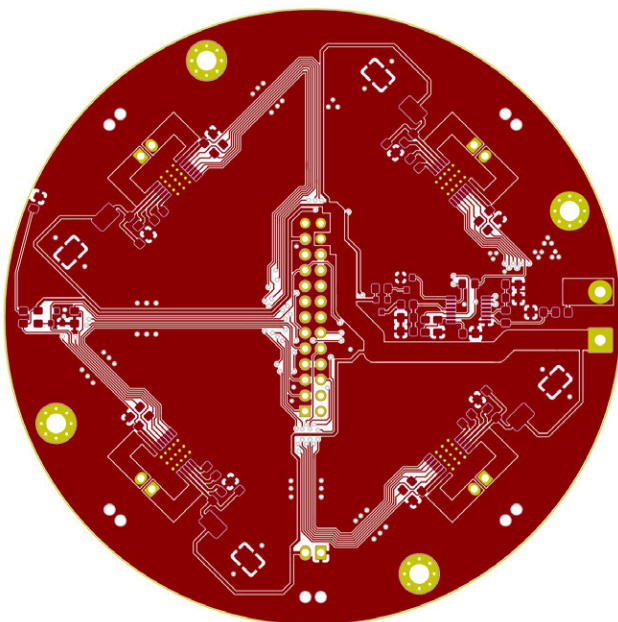


Imagen 25: Capa top de la Power_board

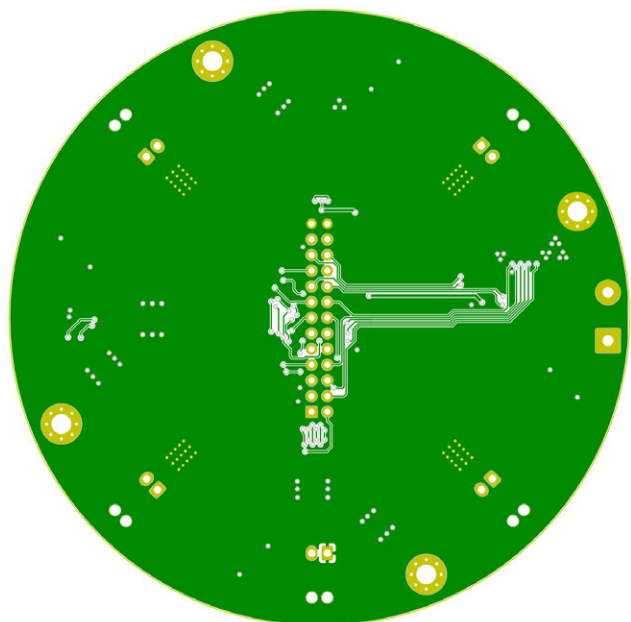


Imagen 26: Capa bottom de la Power_board

Descripción de la solución propuesta

El resultado final puede verse en la siguiente imagen:

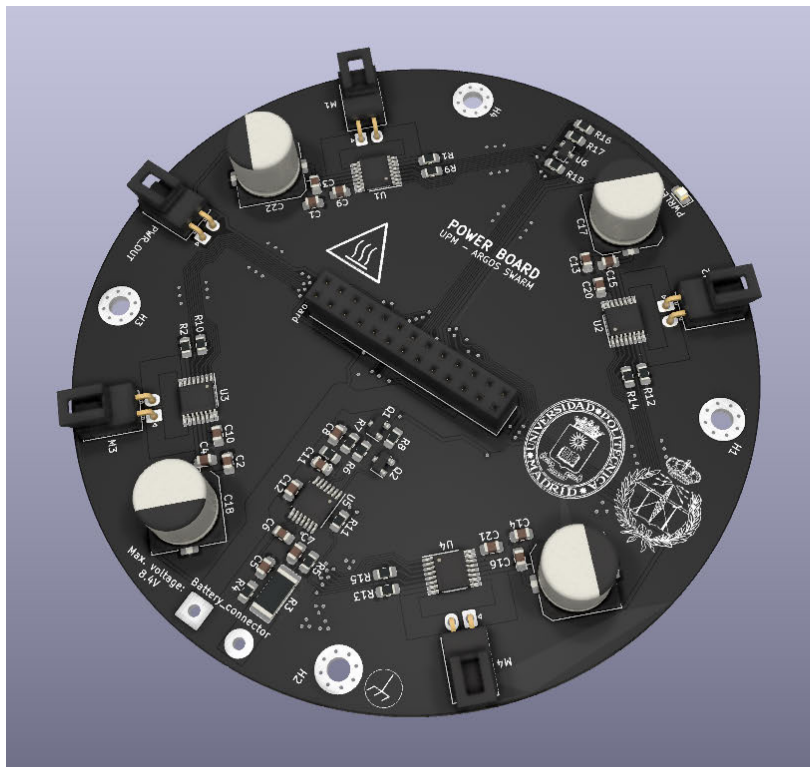


Imagen 27: Simulación 3D de la placa terminada

4.1.3 Nivel 3 y 4: Fundamentos del *sensor de rango y rumbo*

Antes de presentar el diseño de las placas que forman parte del *sensor de rango y rumbo*, es necesario exponer brevemente cual es la teoría de operación del mismo, y que partes lo forman.

4.1.3.1 Definición y descripción de funcionamiento

Un *sensor de rango y rumbo* es un sistema que permite a un robot conocer la distancia (d) y ángulo relativos (ϕ), esto es, el *rango* y el *rumbo*, del resto de robots respecto de sí mismo.

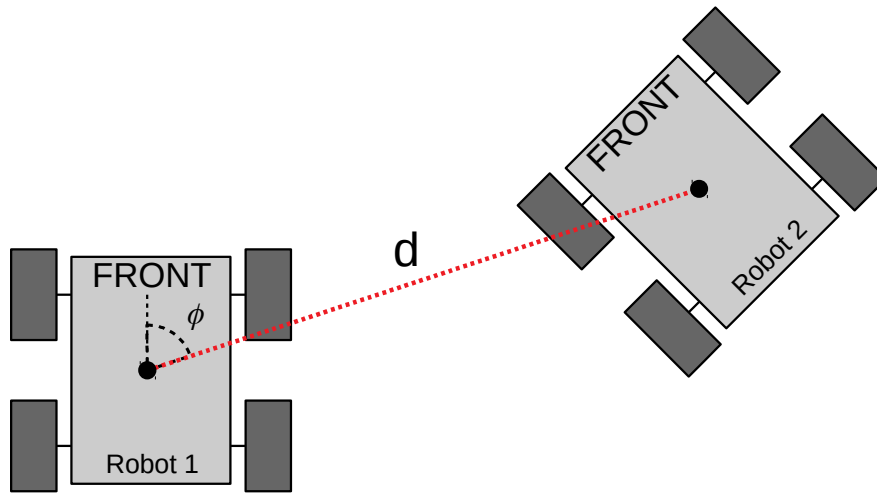


Figura 21: Sensor de rango y rumbo

Existen numerosas formas de implementar este tipo de sistema, no obstante, por el bajo costo, menor complejidad, y mejor frecuencia de refresco, los sistemas basados en infrarrojos son la opción más habitual implementadas en los sistemas multi-robot.

El principio de operación es el siguiente: si se dispone un conjunto de receptores infrarrojos dispuestos de forma radial y equidistante, los cuales entregan una señal proporcional a la intensidad de luz infrarroja recibida, y además se dispone de una fuente emisora de luz infrarroja de intensidad constante; con un algoritmo de cálculo adecuado, es posible obtener la distancia y ángulo que dicha fuente forma con el receptor.

Esta idea, junto con un sistema de identificación que permita conocer que robot del sistema es el responsable de la emisión, permite calcular a un robot las posiciones del resto de robots del sistema que se encuentren dentro del rango operativo del sensor y con los que tenga visión directa.

Tal como se recoge en el capítulo 2, se han desarrollado varios trabajos basados en este principio. En este proyecto, se utilizará el algoritmo de cálculo presentado por Pugh, Martinoli, et. al. en el artículo "A fast On-Board Relative Position Module for Multi-Robot Systems" [16].

4.1.3.2 Algoritmo de cálculo

A parte del algoritmo, en dicho artículo se presenta el diseño del módulo. Este dispone de dieciséis led emisores IR, y ocho fotodiodos, dispuestos de forma radial y equidistante.

El algoritmo de cálculo se utiliza para calcular el rango y el rumbo relativo de la fuente emisora se basa en la observación de que el ángulo de la fuente emisora produce en el fotodiodo receptor una variación de la intensidad de acuerdo con el siguiente modelo:

$$r' = r \cos(\theta) \quad (4) \quad \text{donde: } r': \text{ es la intensidad medida por el fotodiodo.}$$

r : es intensidad medida el fotodiodo, si la fuente emisora esta estuviera alineada con el fotodiodo.

θ : es el ángulo que forma la fuente emisora con el módulo IR.

En la siguiente figura se observa la disposición de los fotodiodos y el ángulo θ :

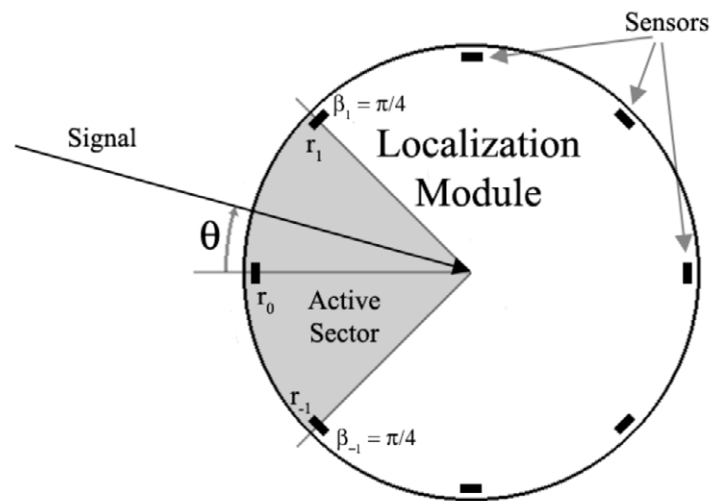


Figura 22: Posición de los fotodiodos en el módulo - [16]

Para realizar el cálculo, se seleccionan los n fotodiodos que reciban la mayor intensidad, siendo n una constante del algoritmo, que como máximo será igual a la mitad de los fotodiodos. Estos fotodiodos seleccionados formarán un sector (parte gris de la figura). Las medidas producidas por los fotodiodos que formen un ángulo positivo respecto del centro del sector se identificarán como $r'_1 \dots r'_m$, y los que forman un ángulo negativo, se identificarán como $r'_{-1} \dots r'_{-m}$.

Así, r'_i será:

$$r'_i = r \cos(\theta - \beta) \quad (5) \quad \text{donde: } \beta_i = -\beta_{-i} \text{ son los ángulos que forman los fotodiodos respecto del centro del sector.}$$

Entonces:

$$r'_i + r'_{-i} = r \cos(\theta - \beta_i) + r \cos(\theta + \beta_i) = 2r \cos(\theta) \cos(\beta_i)$$

$$r'_i - r'_{-i} = r \cos(\theta - \beta_i) - r \cos(\theta + \beta_i) = 2r \sin(\theta) \sin(\beta_i)$$

Ahora, dado que el ángulo β_i es conocido, se pueden despejar las expresiones $r \cos(\theta)$ y $r \sin(\theta)$:

$$a = \frac{\sum_{i=0/1}^n r'_i + r'_{-i}}{\sum_{i=1}^n 2 \cos(\beta_i)} = r \cos(\theta)$$

$$b = \frac{\sum_{i=0/1}^n r'_i - r'_{-i}}{\sum_{i=1}^n 2 \sin(\beta_i)} = r \operatorname{sen}(\theta)$$

A partir de esto:

$$\theta = \arctan\left(\frac{b}{a}\right), \quad r = \sqrt{a^2 + b^2}$$

Conociendo θ , y el ángulo central del sector activo, se puede calcular el rumbo de la fuente emisora, ϕ .

$$\phi = \text{ángulo central del sector} + \theta$$

En el artículo, se expone que tras varias pruebas con diferentes valores de n , el valor óptimo que mejores resultados produce es $n = 3$, es decir, el cálculo se hará con los tres fotodiodos que más intensidad reciban. Esto simplifica las anteriores expresiones a:

$$r'_{-1} = r \cos\left(\theta + \frac{\pi}{4}\right), \quad r'_0 = r \cos(\theta), \quad r'_1 = r \cos\left(\theta - \frac{\pi}{4}\right)$$

$$a = \frac{r'_1 + r'_{-1} + 2r'_0}{2 \cos\left(\frac{\pi}{4}\right) + 2}, \quad b = \frac{r'_1 - r'_{-1}}{2 \sin\left(\frac{\pi}{4}\right)}$$

Con estas expresiones, se calcula el término de rango y el ángulo de rumbo:

$$\phi = \text{ángulo central del sector} + \arctan\left(\frac{b}{a}\right), \quad r = \sqrt{a^2 + b^2}$$

El término de rango, r , no es la distancia, es la intensidad que recibiría un diodo que estuviera alineado exactamente con la fuente emisora. Para obtener la distancia, se utiliza la curva que relaciona la intensidad recibida con la distancia, trazada experimentalmente situando la fuente emisora a diferentes distancias de un fotodiodo alineado con esta.

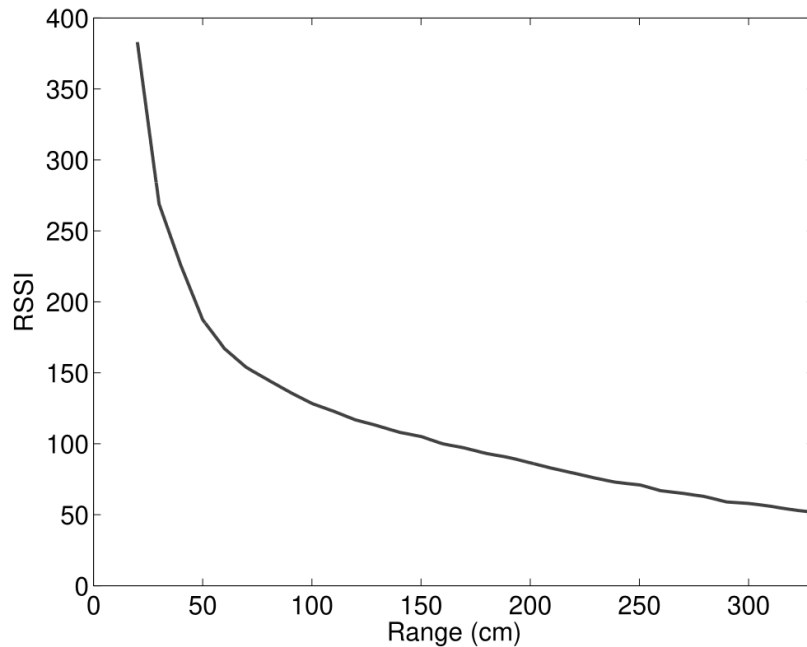


Figura 23: Relación entre el término de rango, r , y la distancia, d . - [16]

A partir de esta curva, se crea una tabla de *look-up*, que permite obtener una estimación de la distancia conociendo el término de rango r , esto es, la intensidad recibida.

4.1.3.3 Diseño preliminar y diagrama de bloques

Una de las variables de mérito de estos sensores son la distancia máxima operativa del sensor, el error de distancia, y el error de ángulo, también llamado, error de rumbo. Otra, es la frecuencia de refresco, es decir, la cantidad de veces por segundo que un robot recibe una actualización de las posiciones del resto de robots. Dado que para conocer la posición de cada robot, cada uno de los robots del conjunto debe turnarse para emitir, la frecuencia de refresco depende del número de robots del sistema.

Como se verá más adelante, la función que relaciona la intensidad recibida por uno de los fotodiodos con la distancia es inversamente proporcional al cuadrado de la distancia que separa la fuente emisora del receptor. Para reducir el error, la señal recibida debe amplificarse considerablemente, y debido a la relación cuadrática, para distancias cercanas, tenderá a saturar los amplificadores, y para distancias lejanas, se perderá resolución.

Para solucionar estos problemas, se pueden adoptar dos estrategias diferentes: la fuente emisora puede enviar señales de intensidad variable, o se puede disponer de un conjunto de amplificadores que aumenten el nivel de la señal recibida por partes, lo que evitaría la saturación y la pérdida de resolución.

En el módulo desarrollado por Pugh, Martinoli, et. al. se utiliza la primera estrategia: el emisor infrarrojo puede emitir con tres niveles de intensidad. El problema de esta estrategia es que o bien debe fijarse un intervalo de operación limitado, o los receptores deben medir durante tres intervalos, uno para cada nivel de intensidad del emisor, aumentando el tiempo y muestreo y reduciendo la frecuencia de operación.

Para un intervalo de operación fijo, limitado a un solo nivel de intensidad del emisor, Pugh, Martinoli, et. al. lograron una frecuencia de refresco máxima de 250Hz para un par de robots. Debido a las de procesamiento, establecieron un límite máximo de 30Hz en la frecuencia de refresco. Así, un grupo de N robots recibiría actualizaciones de posición con una frecuencia de 30Hz/N. Para 10 robots, esta sería de $30\text{Hz}/10 = 3\text{Hz}$.

La frecuencia de muestreo del diseño presentado por Jim Pugh, Alcherio Martinoli, et. al está limitada por la duración del paquete que contenía la información sobre que robot había emitido la señal infrarroja, el cual tenía una duración de 2ms, aproximadamente. Además, este valor de frecuencia de muestreo debería ser menor si se muestra los tres rangos de intensidad que puede emitir el emisor.

De estos 2ms, solo $400\mu\text{s}$ son necesarios para realizar la medida de intensidad, así que, si se elimina la información que identifica a los emisores del canal infrarrojo, y se utiliza algún otro sistema para transmitirla, la frecuencia máxima de operación para dos robots pasaría de 250Hz a 1250Hz. Esto es precisamente lo que se expone en el trabajo "2.5D Infrared Range and Bearing System for Collective Robotics" presentado por James F. Roberts et. al. en [17].

En este trabajo, se presenta un nuevo sensor, que utiliza un módulo de radio a 2.4Ghz para transmitir el identificador de cada robot, lo que permite elevar la frecuencia de refresco a $N/1000\text{Hz}$ con un grupo de N robots. Para 10 robots, la frecuencia de refresco sería 100Hz, es decir, unas 50 veces la obtenida con módulo desarrollado por Jim Pugh, Alcherio Martinoli, et. al.

Además de esto, el módulo presentado por James F. Roberts et. al. presenta otra mejora, utiliza varios amplificadores, conectados en cascada para aumentar el rango máximo de operación del sensor.

Utilizando los principios expuestos por Jim Pugh, Alcherio Martinoli, et. al. y las ideas desarrolladas por James F. Roberts et. al, se propuso diseñar un sistema similar. El sistema diseñado está compuesto por dos módulos: un módulo emisor IR y un módulo receptor.

El módulo emisor IR utiliza 16 leds IR de alta potencia para enviar una onda continua de 10kHz, cuando se activa la transmisión. Además, este módulo está dividido en ocho sectores que pueden activarse de forma individual, lo que permite detectar obstáculos en el entorno cercano utilizando la señal IR reflejada por estos.

El diagrama de bloques preliminar es el siguiente:

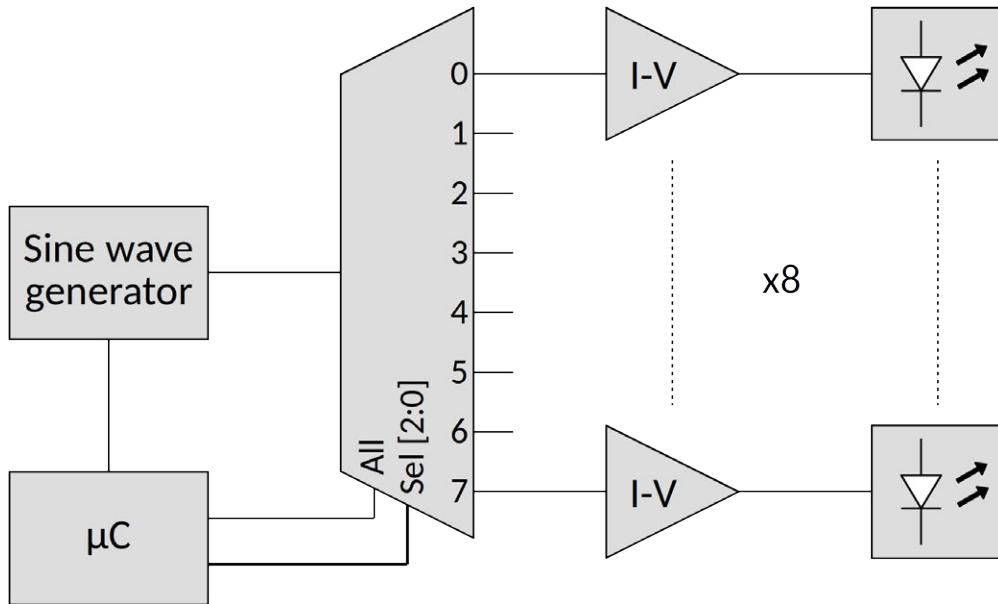


Figura 24: Diagrama de bloques preliminar emisor del sensor R&B

El módulo receptor tiene ocho fotodiodos dispuestos de forma radial. Estos fotodiodos recibirán la señal emitida por el emisor, y tras procesar esta, el módulo entregara a su salida una tensión proporcional a la intensidad pico recibida.

El diagrama de bloques preliminar es el siguiente:

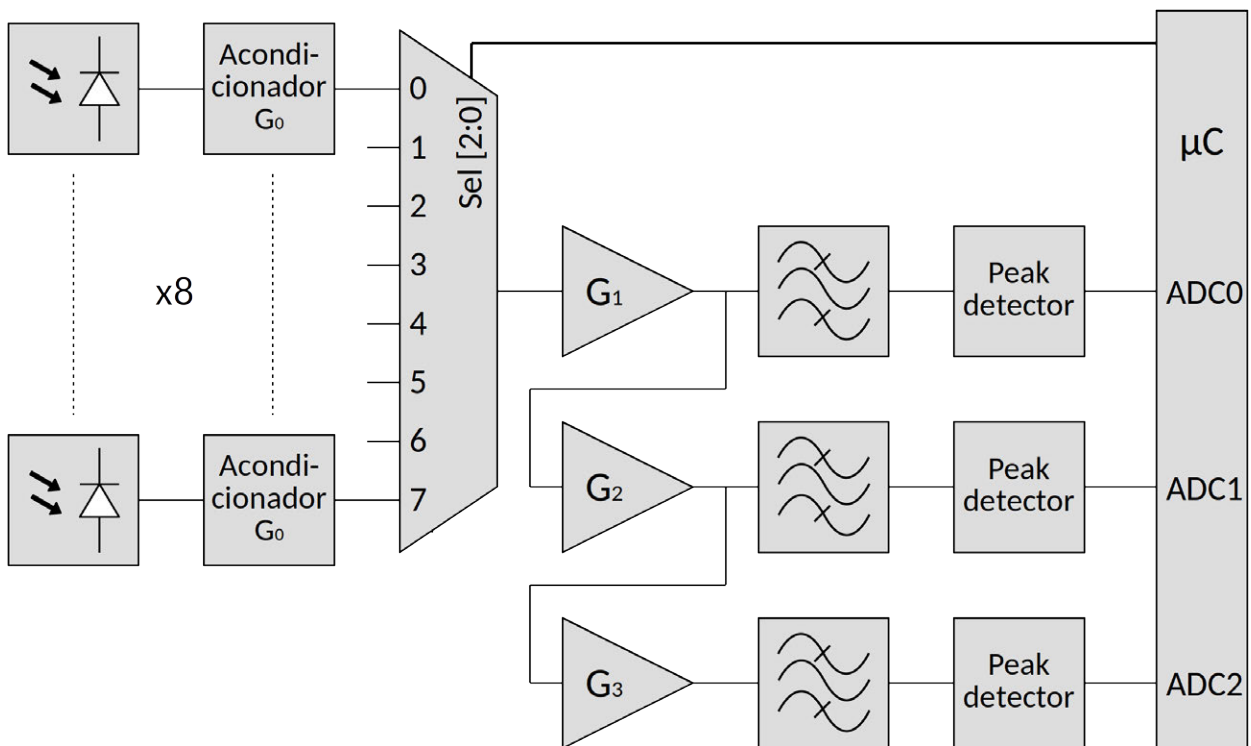


Figura 25: Diagrama de bloques preliminar receptor del sensor R&B

4.1.3.4 Proceso de diseño

Para diseñar los circuitos de los diferentes módulos expuestos en el diagrama de bloques, se seguirá el siguiente proceso: en primer lugar, se hallará un modelo de la función de transferencia del sensor, que relacione la distancia entre el emisor y el receptor con la intensidad recibida.

A continuación, utilizando esta función, se establecerán las ecuaciones que relacionan los diferentes parámetros que limitan este diseño con la distancia medida (ruido térmico, resolución del ADC, etc).

Por último, se aplicarán estas ecuaciones para hallar una configuración que satisfaga los requisitos de precisión. Es importante señalar este último paso, junto con el diseño de los bloques como amplificadores, filtros, etc, es un proceso iterativo, si bien se expone de forma lineal, por las limitaciones en cuanto a extensión de este documento.

En primer lugar, es necesario conocer cuál es la función de transferencia que modela el comportamiento del sistema. En los artículos presentados no se especifica cual ha sido el proceso de diseño de los módulos, ni tampoco se intenta hallar una función de transferencia que relacione de forma analítica la intensidad o irradiancia, r , recibida por un fotodiodo con la distancia al emisor.

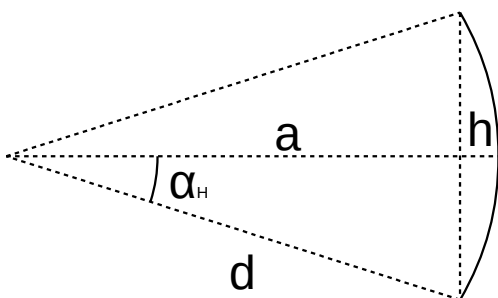
Sin embargo, conocer esta función, o al menos una aproximación a esta, es importante para el diseño de los circuitos, pues de ella dependen las ganancias de los diferentes amplificadores del receptor, así como los límites de ruido térmico, entre otros parámetros.

Para obtener un modelo de la función de transferencia, se tendrá en cuenta que un diodo led radia su energía luminosa de forma dirigida, debido a la lente que forma el encapsulado. En concreto, el patrón de propagación se puede modelar como un casquete esférico. La irradiancia o intensidad en la superficie del casquete esférico será:

$$I_c = \frac{P}{S} \quad (6) \quad \text{donde: } P \text{ es la potencia emitida.}$$

S es la superficie del casquete esférico.

La superficie del casquete se puede calcular, conociendo el ángulo de apertura del casquete esférico. En la hoja de datos del led, el fabricante especifica el perfil de radiación del led, dando el *half-angle* como el punto en el que la irradiancia medida es la mitad de la irradiancia pico emitida. Tomando este valor como el ángulo de apertura del casquete esférico:



$$\begin{cases} S = 2\pi dh \\ h = d - a = d - d \cos \alpha_H = d(1 - \cos \alpha_H) \end{cases} \rightarrow$$

$$S = 2\pi d^2(1 - \cos \alpha_H) \quad (7)$$

Entonces:

$$I_c = \frac{P}{S} = \frac{P}{2\pi d^2(1 - \cos \alpha_h)} \quad (8) \quad \text{donde: } d \text{ es la distancia al led emisor.}$$

α_h es el half-angle de led
 I_c es la irradiancia

La constante $\frac{P}{2\pi d^2(1 - \cos \alpha_h)}$ solo depende de las características del led y de la potencia que emita. Además, esta constante representa la intensidad radiante emitida por el led, la cual se define como:

$$I_{e\Omega} = \frac{P}{\Omega} \quad (9)$$

La definición de ángulo sólido, Ω , es análoga a la definición de radian, pero para un casquete esférico:

$$\Omega = \frac{S}{d^2} \quad (10) \quad \text{donde: } S \text{ es la superficie del casquete esférico.}$$

d es el radio del casquete esférico

Así:

$$\Omega = \frac{2\pi d^2(1 - \cos \alpha_h)}{d^2} = 2\pi(1 - \cos \alpha_h)$$

Entonces:

$$I_c = \frac{P}{2\pi(1 - \cos \alpha_H)d^2} = \frac{P}{\Omega d^2} = \frac{I_{e\Omega}}{d^2}$$

I_c es la irradiancia, esto es, la intensidad en la superficie del casquete esférico, medida en W/m². La irradiancia recibida, I_r , por un fotodiodo a una distancia d será igual a I_c , por lo que, si se conoce esta y la intensidad radiante emitida por el led, $I_{e\Omega}$, la distancia entre el emisor y el receptor es:

$$d = \sqrt{\frac{I_{e\Omega}}{I_r}} \quad (11)$$

La intensidad radiante emitida por el led es proporcional a la corriente que lo atraviesa, y está recogida en la hoja de datos del mismo. La irradiancia recibida, I_r , se puede conocer, ya que un fotodiodo polarizado en inversa entrega una corriente proporcional a la irradiancia recibida. La sensibilidad, esto es, el ratio entre la irradiancia recibida y la corriente inversa o fotocorriente está recogida en la hoja de datos del fotodiodo.

Para la implementación de este sistema, se debe tener en cuenta que la función de transferencia no es lineal. Si se representa la irradiancia recibida en función de la distancia, la curva resultante es:

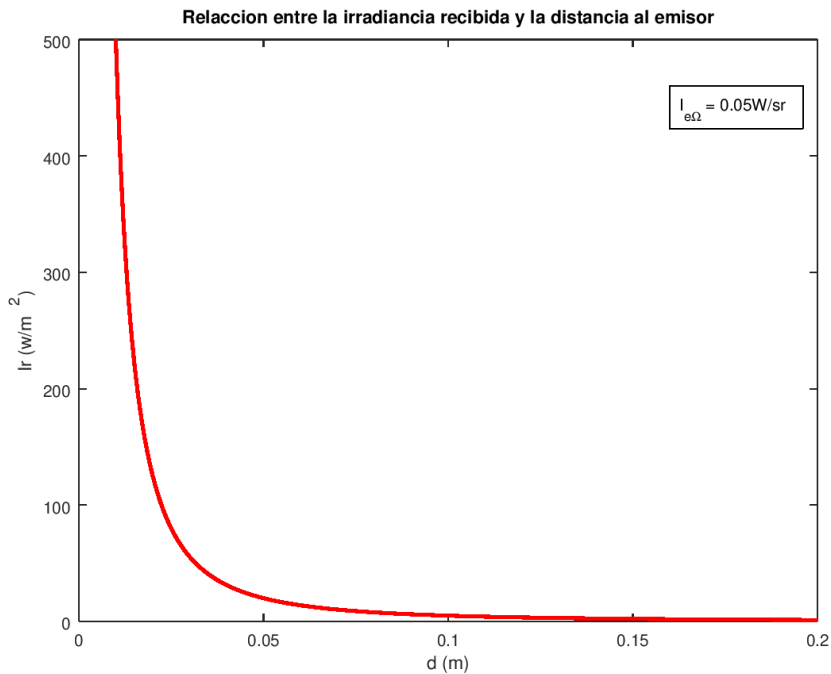


Figura 26: Relación entre la irradiancia recibida y la distancia al emisor

En esta curva, se observa que para valores de distancia grandes, la irradiancia recibida varía mucho menos que para valores pequeños. Por tanto, el error de cuantificación del ADC va a ser mayor en las distancias más lejanas.

Así, si se desea mantener una precisión determinada, será necesario ampliar la función de transferencia por secciones, de forma que el ADC no introduzca un error demasiado grande. Por otro lado, ningún sistema es perfecto, y por tanto, se presentarán errores de ganancia y ruido que han de ser tenidos en cuenta. Estos errores serán los que limiten el alcance y la precisión máxima que este sistema puede desarrollar.

Para estimar como afectan los errores de ganancia y de ruido, se deben relacionar estos con el valor de salida del sensor, es decir, la distancia medida. Dado que la función de transferencia para esta no es lineal, se proyectarán los errores a la salida del módulo emisor sobre la intensidad radiante, $I_{e\Omega}$, y a la entrada del módulo receptor, sobre la irradiancia recibida, I_r . A continuación, se muestra el modelo del sistema:

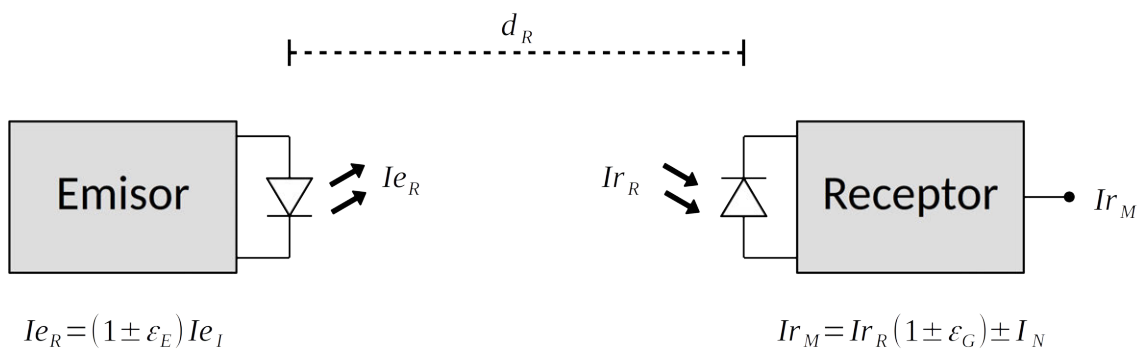


Figura 27: Modelo de error, sensor R&B

donde :

- I_{e_R} : es la intensidad radiante pico real emitida por el emisor.
- I_{e_I} : es la intensidad radiante pico ideal emitida, es decir, la que el emisor emitiría si no hubiera errores.
- ε_E : es el error de amplitud de la intensidad radiante.
- I_{r_R} : es la irradiancia pico real recibida, es decir, el valor real de irradiancia que incide sobre los fotodiodos.
- I_{r_M} : es la intensidad pico medida por el receptor. Es la que se utilizará en el algoritmo para calcular la distancia del emisor.
- I_N : es el error de ruido, cuyo valor es la suma de varios errores introducidos por el receptor, como el ruido térmico, el error de cuantificación del ADC, etc.
- ε_G : error de ganancia del receptor.

Los errores ε_E y ε_G pueden ser compensados por la tabla de look-up, sin embargo, el término I_N no puede compensarse, e influirá considerablemente en las medidas. Aunque se utilizará la tabla de look-up para relacionar la irradiancia medida con la distancia, para obtener una estimación del error de posición debido a I_N , se usará la función de transferencia. Así, la distancia medida es:

$$d_M = \sqrt{\frac{I_{e_I}}{I_{r_M}}} = \sqrt{\frac{I_{e_R}(1+\varepsilon_E)}{I_{r_R}(1+\varepsilon_G) \pm I_N}} = \sqrt{\frac{I_{e_R}}{I_{r_R}} \frac{1}{(1 \pm \varepsilon_E)[(1 \pm \varepsilon_G) \pm \frac{I_N}{I_{r_R}}]}}$$

La distancia medida sin ruido, esto es, si el término I_N fuera nulo, es:

$$d_M|_{noiseless} = \sqrt{\frac{I_{e_I}}{I_{r_M}|_{noiseless}}} = \sqrt{\frac{I_{e_R}}{I_{r_R}(1 \pm \varepsilon_E)(1 \pm \varepsilon_G)}}$$

Entonces, el error relativo de posición debido al ruido es:

$$\varepsilon_N = \frac{d_M - d_M|_{noiseless}}{d_M|_{noiseless}} = \sqrt{1 \pm \varepsilon_G} \left[\frac{1}{\sqrt{(1 \pm \varepsilon_G) \pm \frac{I_N}{I_{r_R}}}} - \frac{1}{\sqrt{(1 \pm \varepsilon_G)}} \right] = \frac{1}{\sqrt{1 \pm \frac{I_N}{I_{r_R}(1 \pm \varepsilon_G)}}} - 1$$

Ahora, para eliminar I_{r_M} , se tienen en cuenta que según la función de transferencia:

$$d_R = \sqrt{\frac{I_{e_R}}{I_{r_R}}} = \sqrt{\frac{I_{e_I}(1 \pm \varepsilon_E)}{I_{r_R}}} \rightarrow I_{r_R} = \frac{I_{e_I}(1 \pm \varepsilon_E)}{d_R^2}$$

$$\varepsilon_N = \frac{1}{\sqrt{1 \pm \frac{I_N d_R^2}{I_{e_I}(1 \pm \varepsilon_E)(1 \pm \varepsilon_G)}}} - 1 \quad (12)$$

El error será mayor cuando el denominador sea menor, y el denominador será menor cuando $I_N < 0$, $(1 \pm \varepsilon_E) < 1$ y $(1 \pm \varepsilon_G) < 1$. Entonces, el error máximo debido al ruido del receptor es:

$$\varepsilon_{Nmax} = \frac{1}{\sqrt{1 - \frac{I_N d_R^2}{I_{e_I}(1 - \varepsilon_E)(1 - \varepsilon_G)}}} - 1 \quad (13)$$

Este error define el error máximo que puede presentarse entre medidas consecutivas, esto es, la precisión. Como se observa, la influencia del término $\frac{I_N}{I_{e_I}(1 - \varepsilon_E)(1 - \varepsilon_G)}$ aumenta de forma cuadrática con la distancia. Así, para un error máximo, ε_{Nmax} , y un valor de I_N determinados, existirá un valor de distancia para el que el error, $|\varepsilon_N|$ está por debajo de $|\varepsilon_{Nmax}|$. Despejando d_R , se obtiene la distancia final mínima del trozo de la función de transferencia, que cumple con que $|\varepsilon_N| < |\varepsilon_{Nmax}|$ para las condiciones más desfavorables, esto es, $I_N < 0$, $(1 \pm \varepsilon_E) < 1$ y $(1 \pm \varepsilon_G) < 1$.

$$d_{R-F}|_{min} = \sqrt{\frac{I_{e_I}(1 - \varepsilon_E)(1 - \varepsilon_G)}{I_N} \left(1 - \frac{1}{(1 + \varepsilon_{Nmax})^2} \right)} \quad (14)$$

Esta ecuación establece que para aumentar la distancia final, se puede disminuir I_N , o bien, aumentar la intensidad radiante emitida por los leds, I_{e_I} . El módulo receptor convertirá la intensidad pico de la irradiancia recibida en una señal de tensión, de acuerdo a la expresión:

$$V_{O(j)} = I_{IN} \prod_{k=0}^j G_k \quad (15) \quad \text{donde: } I_{IN} \text{ es la irradiancia pico a la entrada del receptor.}$$

G_k es la ganancia de cada etapa k.
 V_{jO} es la tensión pico a la salida de la etapa j.

Así, la tensión de ruido a la salida del módulo para la etapa j es:

$$V_{N(j)} = I_{N(j)} \prod_{k=0}^j G_k \quad \rightarrow \quad I_{N(j)} = \frac{V_{N(j)}}{\prod_{k=0}^j G_k}$$

Entonces, para la etapa o sección j de la función de transferencia, la distancia final mínima que garantiza que la precisión cumple $|\varepsilon_N| < |\varepsilon_{Nmax}|$, es:

$$d_{R-F(j)}|_{min} = \sqrt{\frac{I_{e_I}(1 - \varepsilon_E) \prod_{k=0}^j G_k (1 - \varepsilon_{G_k})}{V_{N(j)}} \left(1 - \frac{1}{(1 + \varepsilon_{Nmax(j)})^2} \right)} \quad (16)$$

Descripción de la solución propuesta

Esta ecuación determinará el límite superior de la distancia operativa mínima, para un conjunto de ganancias, un valor de ruido a la salida ($V_{N(j)}$), y una precisión máxima ($\varepsilon_{Nmax(j)}$). Para todas las condiciones de operación, el límite superior de distancia operativa es:

$$d_{R_F(j)} = \sqrt{\frac{Ie_I(1 \pm \varepsilon_E) \prod_{k=0}^j G_k(1 \pm \varepsilon_{G_k})}{\pm V_{N(j)}} \left(1 - \frac{1}{(1 \pm \varepsilon_{N(j)})^2}\right)} \quad (17)$$

El límite inferior de la distancia operativa se obtendrá cuando la irradiancia medida alcance el fondo de escala del ADC.

$$\left. \begin{aligned} d_{R_0} &= \sqrt{\frac{Ie_R}{Ir_{R|max}}} \\ Ie_R &= Ie_I(1 \pm \varepsilon_E) \\ Ir_M &= Ir_R(1 \pm \varepsilon_G) \pm I_N \\ Ir_{M|max} &= \frac{V_{SFV}}{G} = \frac{V_{ref}(1 - 2^{-ADC-N})}{G} \end{aligned} \right\} \rightarrow d_{R_0} = \sqrt{\frac{Ie_I G(1 \pm \varepsilon_E)(1 \pm \varepsilon_G)}{V_{ref}(1 - 2^{-ADC-N}) \pm V_N}}$$

Ahora, para la sección j de la función de transferencia, el límite inferior de la distancia de operación es:

$$d_{R_0(j)} = \sqrt{\frac{Ie_I(1 \pm \varepsilon_E) \prod_{k=0}^j G_k(1 \pm \varepsilon_{G_k})}{V_{ref}(1 - 2^{-ADC-N}) \pm V_{N(j)}}} \quad (18) \quad \text{donde: } V_{ref} \text{ es la tensión de referencia del ADC.}$$

ADC_N es el número de bits del ADC.

Las ecuaciones (17) y (18) determinan los límites de distancia para una sección de la función de transferencia. Para aumentar el alcance del sensor, se deberán añadir tantas secciones o trozos como sea necesario, mientras el error introducido por el ruido lo permita. Las secciones posteriores deben solaparse ligeramente para evitar que existan huecos, es decir, valores para la irradiancia recibida que no puedan relacionarse con la función de transferencia.

Así, para cualquier sección excepto la primera, existirá una distancia de solapamiento entre secciones. Para una sección j , la distancia de solapamiento con la sección anterior, $j-1$, es:

$$d_{S(j)} = d_{R_F(j-1)} - d_{R_0(j)} \quad (19)$$

La distancia de solapamiento entre secciones debe ser siempre positiva para todas las fuentes de error posibles. Sustituyendo términos y operando, se obtiene la siguiente expresión:

$$d_{S(j)} = \underbrace{\sqrt{Ie_I(1 \pm \varepsilon_E) \prod_{k=0}^j G_k(1 \pm \varepsilon_{G_k})}}_A \left(\underbrace{\sqrt{\frac{1}{\pm V_{N(j-1)}} \left[1 - \frac{1}{(1 \pm \varepsilon_{N(j-1)})^2}\right]}}_B - \sqrt{\frac{G_j(1 \pm \varepsilon_{G_j})}{V_{ref}(1 - 2^{-ADC-N}) \mp V_{N(j)}}} \right) \quad (20)$$

El término A puede cambiar el tamaño de la distancia de solapamiento, pero no puede cambiar el signo, es decir, no puede provocar que se “desolapen” las secciones. Esto es significativo porque implica que cualquier variación en la intensidad radiante emitida, en la ganancia de etapas anteriores, o que pueda expresarse como alguna de las anteriores, no creará huecos entre las secciones de la función de transferencia.

El término B, por otro lado, puede provocar que el signo de la distancia de solapamiento cambie, y por tanto, se produzcan huecos entre las secciones de la función de transferencia. Dado que todos los términos, excepto ε_{G_j} , han sido tenidos en cuenta al fijar la distancia final y la distancia inicial para la secciones de la función de transferencia, el término B solo puede cambiar de signo si $\varepsilon_{G_j} > 0$ y alcanza un valor suficientemente grande.

Intuitivamente, esto equivale a que la ganancia real de la etapa j aumente, y dado que según la ecuación (18), la distancia inicial de la sección j , aumentará, esto acabará produciendo que supere a la distancia final de la sección $j-1$, “desolapando” las secciones de la función de transferencia. Para evitar que esto ocurra, debe diseñarse cada etapa de forma que el error máximo en la ganancia, este por debajo de un valor determinado. Para obtener este valor, se despeja ε_{G_j} del término B, imponiendo la condición de que este sea siempre positivo:

$$\left(\sqrt{\frac{1}{\pm V_{N(j-1)}} \left[1 - \frac{1}{(1 \pm \varepsilon_{N(j-1)})^2} \right]} - \sqrt{\frac{G_j(1 \pm \varepsilon_{G_j})}{V_{ref}(1 - 2^{-ADC-N}) \mp V_{N(j)}}} \right) > 0$$

Ahora, teniendo en cuenta que el radicando de la raíz derecha debe ser positivo para que la raíz tenga solución real, y que las peores condiciones de solapamiento se darán cuando la distancia inicial de la sección j sea mínima, lo que se producirá cuando $\pm V_{N(j)} < 0$, y la distancia final de la sección $j-1$ sea mínima, lo que se producirá cuando $\pm V_{N(j-1)} > 0$ y por tanto, $\pm \varepsilon_N = \pm \varepsilon_{Nmax} > 0$; se obtiene que el valor del error de ganancia para la sección j debe estar en el intervalo:

$$-1 < \varepsilon_{G_j} < \frac{V_{ref}(1 - 2^{-ADC-N}) - V_{N(j)}}{G_j V_{N(j-1)}} \left[1 - \frac{1}{(1 + \varepsilon_{Nmax(i-1)})^2} \right] - 1 \quad (21)$$

Una vez expuestas las ecuaciones que relacionan las diferentes fuentes de error con el error de distancia, y que modelan los límites de funcionamiento del sensor, se pasará a explicar el proceso seguido para calcular las ganancias de cada etapa, y fijar los valores de distancia iniciales y finales de cada sección de la función de transferencia.

En primer lugar, las ecuaciones (16) y (18) establecen los límites superiores e inferiores de la distancia de operación para cada sección de la función de transferencia. Estas ecuaciones proporcionan el valor de distancia real, el cual tiene en cuenta el error de amplitud de la intensidad radiante, el error de ganancia de cada etapa, así como el error introducido por el ruido del receptor. El error de amplitud de la intensidad radiante y el error de ganancia de cada etapa son errores sistemáticos, por lo que utilizando la tabla de look-up, pueden ser corregidos. El error de ganancia de cada etapa solo debe cumplir con el requisito impuesto por la ecuación (21) y debe ser realizable, es decir, debe ser superior a 0.

Por otro lado, los errores debido al ruido son errores no sistemáticos, es decir, no pueden ser corregidos porque se presentan de forma aleatorio, aunque si pueden ser acotados. En la ecuación (18), el error introducido por el ruido del receptor no es significativo para el proceso de diseño, pues si se tiene en cuenta el requisito de la ecuación (21), no se producirá hueco entre las secciones de la función de transferencia.

Sin embargo, en la ecuación (16), el error de ruido si es significativo, pues es el error que impondrá el límite superior de operación de la sección. Así pues, para seleccionar las ganancias, se puede anular el error de amplitud de la intensidad radiante y el error de ganancia de cada etapa en las ecuaciones (16) y (18), así como el error de ruido en la ecuación (18). Con estas condiciones, las ecuaciones de diseño que proporcionan los límites ideales para la sección j de la función de transferencia son:

$$d_{I_F(j)}|_{min} = \sqrt{\frac{Ie_I \prod_{k=0}^j G_k}{V_{N(j)}} \left(1 - \frac{1}{(1 + \epsilon_{Nmax(j)})^2}\right)} \quad (22)$$

$$d_{I_0(j)} = \sqrt{\frac{Ie_I \prod_{k=0}^j G_k}{V_{ref} (1 - 2^{-ADC-N})}} \quad (23)$$

El proceso de diseño, se resume en los siguientes pasos:

1. Se establece la precisión máxima que va a tener cada sección de la función de transferencia, es decir, se elige un valor para $\epsilon_{Nmax(j)}$. Como en este punto aún es desconocido que longitud de operación va a tener cada sección y cuantas secciones habrá, se puede fijar un valor único para cada todas las secciones. Además, se especifica cuál va a ser el alcance mínimo y el alcance máximo del sensor.
2. A partir de las hojas de datos del fotodiodo, se obtiene el valor de sensibilidad, S que relaciona la irradiancia recibida (W/m^2) con la fotocorriente. Además, se establece un valor de ganancia, G_0 , para el acondicionador del fotodiodo. Este valor debe ser lo más alto posible, pues de ello depende la relación señal a ruido del sensor, sin embargo, y como se verá más adelante existe una relación de compromiso entre la ganancia G_0 , el rango dinámico del sensor, y la frecuencia de operación. Además, se calcula la intensidad radiante emitida por los leds IR del emisor.
3. Se calcula la densidad espectral de ruido (ruido térmico) a la salida del receptor, y se deja en función de las ganancias de cada etapa, desconocidas por el momento.
4. Se elige un ancho de banda realizable para los filtros, idealmente, cuanto más pequeño mejor, pues disminuirá el ruido a la salida del receptor.
5. Se computa una estimación para todos los errores no sistemáticos, es decir, para el resto de fuentes que contribuyen al ruido del receptor, a parte del ruido térmico, a la salida. Ejemplos de estas son: el ruido de cuantificación del ADC, el error de aproximación del detector de pico,

el ruido acoplado por el rizado de tensión de alimentación del acondicionador de los fotodiodos, etc.

6. Utilizando la ecuación (23), se asigna un valor para la ganancia de la primera etapa G_1 , que garantice que $d_{L_0(1)}$ es igual al alcance mínimo especificado en el paso 1.
7. Utilizando las ganancias G_0, G_1 , la densidad espectral de ruido, el ancho de banda de los filtros, y los valores calculados de las contribuciones de los errores no sistemáticos calculados en el paso 5, se calcula el valor pico del ruido total a la salida. Dado que las fuentes de ruido no están correladas, el ruido total se computa como la suma RMS de todas ellas. Además, puede establecerse que en su mayoría, presentan distribuciones gaussianas, por lo que el valor pico con una probabilidad del 97%, según [32], es:

$$V_{N(1)} = \left(\sqrt{V_{O_NT}^2 + V_{O_ADC}^2 + V_{O_D.PICO}^2 + \dots} \right) \cdot 6,6$$

8. Utilizando el valor de $V_{N(1)}$, y los parámetros calculados en los pasos anteriores, se utiliza la ecuación (22) para calcular el límite superior de la sección 1, $d_{L_F(1)}$.
9. Utilizando la ecuación (23), se selecciona una ganancia para la sección j de forma que se cumpla:

$$d_{R_F(j-1)} > d_{R_0(j)}$$

10. Utilizando las ganancias G_0, G_1, \dots, G_j , la densidad espectral de ruido, el ancho de banda de los filtros, y los valores calculados de las contribuciones de los errores no sistemáticos calculados en el paso 5, se calcula el valor pico del ruido total a la salida de la etapa j .

$$V_{N(j)} = \left(\sqrt{V_{O_NT}^2 + V_{O_ADC}^2 + V_{O_D.PICO}^2 + \dots} \right) \cdot 6,6$$

11. Utilizando el valor de $V_{N(j)}$, y los parámetros calculados en los pasos anteriores, se utiliza la ecuación (22) para calcular el límite superior de la sección j , $d_{L_F(j)}$.
12. Utilizando la ecuación (21), se comprueba que el error de ganancia de la sección j es realizable, esto es, que tiene un valor suficientemente grande como para diseñar un amplificador que lo cumpla, y que $\epsilon_{G_j} > 0$. De no cumplirse estos requisitos, se vuelve al paso 9, y se selecciona una ganancia más pequeña para la sección j , lo que aumenta la distancia de solapamiento.
13. Si el límite superior de la sección j es mayor al alcance máximo del sensor fijado en el paso 1, se pasa al paso siguiente, de lo contrario, se vuelve al paso 9 y se añade una sección $j+1$ nueva.
14. Tomando el ancho de banda, las estimaciones de ruido térmico, las ganancias para cada etapa, y los errores máximos de ganancia de cada etapa como requisitos, se diseña cada uno de los bloques del receptor. Si durante el proceso se encuentra que algún bloque no puede cumplir con los requisitos, se vuelve al paso 4 y se comienza de nuevo el proceso de diseño, disminuyendo los requisitos de aquel o aquellos parámetros que generen problemas.

Este proceso, como se adelante al comienzo de la sección es iterativo, y debe seguirse hasta que el diseño se haya completado. Para simplificar la selección de las ganancias de cada etapa, se ha

creado una tabla de cálculo que permite obtener los diferentes parámetros. Esta hoja de cálculo, se encuentra en la ruta *Hardware/SWR-UPM/Diseño_sensor_R&B*, bajo el nombre *calculo_parametros_sensor_R&B.ods*, en los ficheros que acompañan a este documento.

4.1.3.5 Parámetros de diseño escogidos

Con el objetivo de presentar el diseño de forma lineal, a continuación se adelantarán los valores de algunos parámetros que serán calculados en los apartados siguientes. Esto permitirá establecer los valores de ganancia de cada etapa, y los errores máximos de ganancia.

Parámetros:	
Iei (W/sr)	5,00E-02
Error Ie (%)	0,00
S_PD (Am ² /W)	8,00E-06
G0	7150
G1	3,82
G2	20
G3	10
H01	1
H02	1
H03	1
Ancho de banda del filtro (Hz)	1600
ADC Vref	2,048
ADC N bits	10
Errores DC	
Error detector de pico (Vp)	8,20E-03
Densidad espectral de ruido por módulo	
Acondicionador del fotodiodo (V/Hz ^{0.5})	1,09E-08
Amplificador de la primera etapa (V/Hz ^{0.5})	1,19E-08
Amplificador de la segunda etapa (V/Hz ^{0.5})	1,21E-08
Amplificador de la tercera etapa (V/Hz ^{0.5})	1,21E-08
Ruido de la fuente de polarización de los diodos	
Rizado de alimentación (V)	5,00E-06
Requisitos	
Precisión mínima rango 1 (%)	5
Precisión mínima rango 2 (%)	2,5
Precisión mínima rango 3 (%)	5

Tabla 1: Parámetros de diseño del sensor de rango y rumbo

Con estos parámetros las contribuciones de error, y el error pico a la salida de cada etapa son los siguientes:

Densidad espectral de ruido a la salida de la etapa	
D.E.R. Rango 1 (V/Hz ^{0.5})	6,18E-08
D.E.R. Rango 2 (V/Hz ^{0.5})	1,26E-06
D.E.R. Rango 3 (V/Hz ^{0.5})	1,26E-05

Tabla 2: Densidad espectral de ruido a la salida de cada etapa

Ruido debido a los errores DC	
Error detector pico (Vrms)	1,24E-03
Error ADC (Vrms)	5,77E-04
Error rizado Vbias (Vrms)	7,58E-07
Ruido pico calculado a la salida de cada etapa	
Ruido_pico rango 1 (V)	9,04E-03
Ruido_pico rango 2 (V)	9,06E-03
Ruido_pico rango 3 (V)	1,07E-02

Tabla 3: Otras contribuciones de error y error RMS a la salida de cada etapa

Con estos valores, aplicando la ecuación (21), se obtienen los errores máximos de ganancia:

Error máximo de ganancia	
Error máximo admisible G2 (%)	4,14E+00
Error máximo admisible G3 (%)	7,66E+00

Tabla 4: Error relativo de ganancia máximo admisible para los amplificadores de etapa

Entonces, las distancias superiores e inferiores para cada rango, así como la tensión mínima a la salida de cada etapa es:

Rango 1	
d_0 (cm):	7,32
d_f (cm):	33,42
Vo_min (V)	0,0978

Rango 2	
d_0 (cm):	32,75
d_f (cm):	107,51
Vo_min (V)	0,1890

Rango 3	
d_0 (cm):	103,61
d_f (cm):	435,61
Vo_min (V)	0,1152

Tabla 5: Distancias iniciales y finales de cada etapa

Como se puede observar, existe un solapamiento entre las distancias inferiores de cada rango y las distancias superiores de los que los preceden; además, el error máximo de ganancia es realizable. Por tanto, para el conjunto de ganancias presentadas, ancho de banda, etc, el diseño es válido.

Los valores presentados serán los que se utilizarán en el diseño del emisor del sensor R&B, y del receptor del sensor R&B.

A continuación, se expondrá el diseño del emisor y del receptor del *sensor de rango y rumbo*. Los cálculos para computar los errores de ganancia, densidades espectrales de ruido de cada bloque, etc, si bien no son complejos, sí son tediosos. Así pues, y debido a las limitaciones de extensión de este trabajo, se mostrarán solo los desarrollos más fundamentales, y los resultados que justifiquen los parámetros elegidos.

4.1.4 Nivel 3 – Placa del emisor del sensor de rango y rumbo (R&B_sensor_tx board)

Las especificaciones que satisface son: EH-3, EH-11.

A continuación, se expondrá el proceso de diseño de la placa que forma el emisor del sensor de rango y rumbo. Para facilitar el desarrollo de la exposición, se comenzará por la selección y caracterización de los leds emisores de infrarrojos, y se terminara con el diseño del generador de onda sinusoidal. Tras esto, se presentara el diseño de la PCB y piezas auxiliares, y este apartado concluirá con un breve análisis para validar el diseño, comparando los valores teóricos con los medidos en las simulaciones y en el prototipo terminado.

4.1.4.1 Leds emisores de infrarrojos: selección, caracterización y disposición

Para detectar obstáculos por reflexión de infrarrojos, el módulo emisor de infrarrojos debe disponer de ocho sectores que puedan ser activados individualmente. En el centro de cada sector se situará el fotodiodo que recibirá la señal reflejada. Además, con objeto de mejorar el alcance del sensor, se utilizarán dos leds IR por sector, así pues, se dispondrán dieciséis leds de forma radial y equidistante. Entonces, para que los patrones de radiación de cada led se solapen con los de sus vecinos, el *half-angle* de cada led debe ser de al menos $\pm 22,5^\circ$.

A parte del requisito del *half-angle*, el led elegido debe operar en la longitud de onda en la que la sensibilidad del fotodiodo receptor (BPV22NF) es máxima, la cual es de 940nm. También debe tener un patrón de radiación lo más uniforme posible, y se debe buscar que la intensidad radiante sea máxima. Todas estas características se encuentran en el led infrarrojo TSAL6400 del fabricante Vishay Semiconductors.

Según el *datasheet* del led IR TSAL6400 [33], este tiene un *half-angle* de $\pm 25^\circ$, una longitud de onda pico de 940nm, una intensidad radiante típica de 50mW/sr, y una corriente directa máxima no pulsada de 100mA.

Para generar una señal infrarroja cuya intensidad varíe de forma sinusoidal, se explota el hecho de que la intensidad radiante generada por led es directamente proporcional a la corriente que lo atraviesa, tal como se expone en [34]. Así, a partir de la curva *Radiant Intensity vs. Forward Current* presentada en el *datasheet*, se obtiene la ganancia G_D :

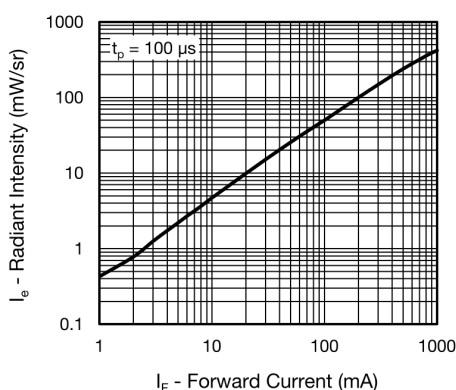


Figura 28: TSAL6400 - Curva I_e/I_f [33]

$$y - y_0 = m(x - x_0) \rightarrow G_{DI} = m = \frac{(y - y_0)}{(x - x_0)}$$

$$\begin{cases} x = 100 \\ y = 50 \\ x_0 = 20 \\ y_0 = 10 \end{cases} \rightarrow G_{DI} = m = \frac{(50 - 10)}{(100 - 20)} = 0.5 \frac{W}{sr \cdot A}$$

Cada led solapará su patrón de emisión con su vecino, por tanto:

$$G_D = 2 \cdot G_{DI} = 1 \frac{W}{sr \cdot A} \quad (24)$$

Para controlar la corriente de excitación de los leds infrarrojos se utilizará un amplificador de transconductancia.

4.1.4.2 Amplificadores de transconductancia

Se dispondrá de ocho amplificadores de transconductancia, que controlarán la corriente que circula por cada par de leds que forman un sector, los cuales estarán conectados en serie. Estos amplificadores están representados en el diagrama de bloques preliminar mostrado en la *Figura 24*, como los amplificadores con ganancia *I-V*.

Dado que los leds infrarrojos solo permiten el paso de la corriente en un sentido, para generar una señal de intensidad radiante que varíe de forma sinusoidal, la corriente que atraviesa los leds nunca debe cambiar de sentido. Por tanto, deberá existir una corriente constante que circule a través de led para mantener el diodo polarizado de forma directa, lo que implica que el amplificador de transconductancia se comportará como un amplificador de clase A.

Teniendo esto en cuenta, el amplificador de transconductancia se implementará utilizando una *fuerza de corriente de Howland* con algunas modificaciones. Estas modificaciones son: se utilizará un transistor bipolar como etapa de salida, lo que permite aumentar la corriente de excitación de los leds; y se colocará un interruptor controlado por tensión, que permita activar o desactivar la excitación a los leds emisores de infrarrojos. El esquema se presenta a continuación:

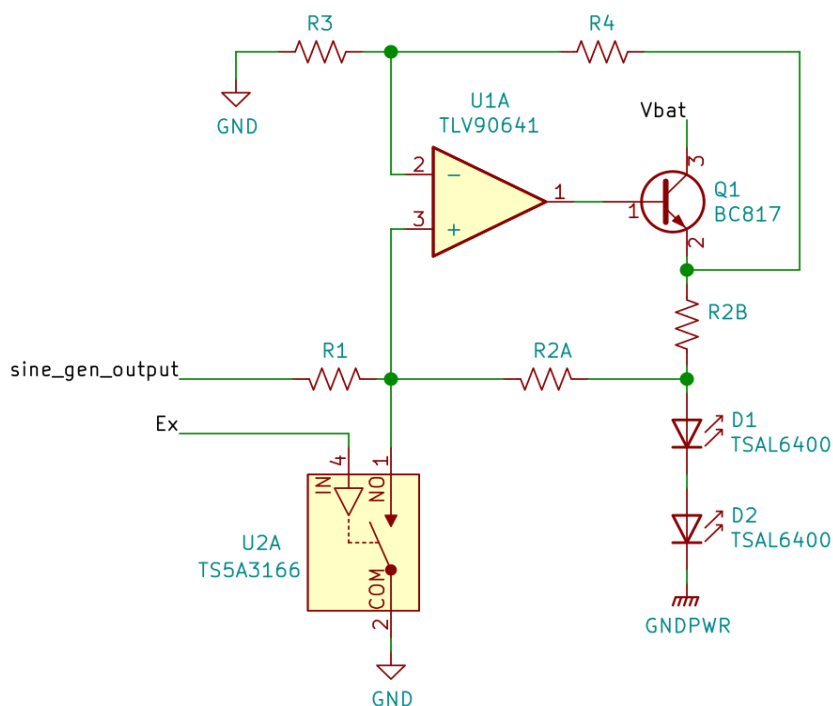


Figura 29: Amplificador transconductancia

Toda fuente de corriente real, tiene una resistencia de salida, la cual provoca que la corriente de salida real difiera de la corriente de salida teórica. El modelo de una fuente de corriente dependiente (de una tensión) y real es:

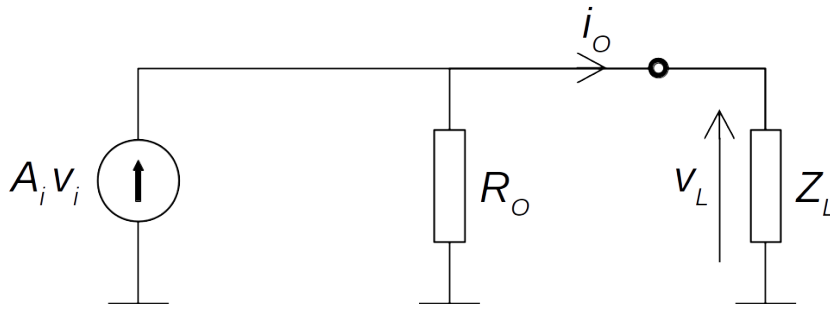


Figura 30: Modelo de una fuente de corriente real

$$i_o = A_i v_i - \frac{v_L}{R_O} \quad (25)$$

$$Z_L = \frac{v_L}{i_o} \quad (26)$$

La impedancia de salida Z_L representa la suma de la impedancia de los leds infrarrojos D_1 y D_2 . Para calcular las ecuaciones de diseño, que relacionan el valor de los componentes y el valor de tensión de entrada, v_i , con el valor de la corriente de salida, i_o , se supondrá que el amplificador operacional es ideal y que la realimentación del amplificador operacional es negativa, y por tanto, las entradas del AO cumplen $v_p = v_n$. La realimentación negativa del AO podrá demostrarse cuando se calcule la ganancia de lazo, $T(jf)$. Así, para obtener las ecuaciones de diseño, se aplica la fórmula del divisor de tensión y la Ley de corrientes de Kirchhoff:

$$\begin{cases} v_o = v_p \left(1 + \frac{R_4}{R_3}\right) \\ i_o = \frac{v_o - v_L}{R_{2B}} - \frac{v_p - v_L}{R_{2A}} \\ \frac{v_i}{R_1} = \frac{v_p}{R_1} + \frac{v_p - v_L}{R_{2A}} \end{cases}$$

Operando, se obtiene la siguiente expresión cuyos términos pueden relacionarse con los de la ecuación (26):

$$i_o = v_i \underbrace{\left(\frac{R_{2A}}{R_{2A} + R_1} \right) \left[\frac{1}{R_{2B}} \left(1 + \frac{R_4}{R_3} \right) + \frac{1}{R_{2A}} \right]}_{A_i} + v_L \underbrace{\left(\frac{R_1}{R_{2A} + R_1} \right) \left[\frac{1}{R_{2B}} \left(1 + \frac{R_4}{R_3} \right) + \frac{1}{R_{2A}} \right] - \left(\frac{1}{R_{2A}} + \frac{1}{R_{2B}} \right)}_{-\frac{1}{R_O}} \quad (27)$$

Reordenando los términos y simplificado, se obtiene la ganancia de la fuente de corriente, A_i , y la resistencia de salida R_O :

$$A_i = \frac{R_3 (R_{2A} + R_{2B}) + R_4 R_{2A}}{(R_{2A} + R_1) R_3 R_{2B}} \quad (28)$$

$$R_O = \frac{(R_1 + R_{2A}) R_3 R_{2B}}{R_3 (R_{2A} + R_{2B}) - R_1 R_4} \quad (29)$$

Una fuente de corriente ideal tiene una resistencia de salida infinita, esto es: $R_O = \infty$. Según la ecuación (29), para que esto ocurra, el denominador debe ser nulo. Por tanto, si se cumple que:

$$R_3(R_{2A}+R_{2B})-R_1R_4=0 \quad \rightarrow \quad \boxed{\frac{R_{2A}+R_{2B}}{R_1}=\frac{R_4}{R_3}} \quad (30)$$

se dice que el puente de resistencias esta balanceado. Aplicando la ecuación (30) a la expresión de la ganancia de la fuente de corriente, y operando, se obtiene:

$$\boxed{A_i|_{P.B.}=\frac{R_4}{R_3R_{2B}}} \quad (31)$$

Del mismo modo, bajo estas condiciones, la ecuación (26) se simplifica a:

$$\boxed{i_o|_{P.B.}=v_i\frac{R_4}{R_3R_{2B}}} \quad (32)$$

Hasta ahora, se ha considerado que el amplificador operacional es ideal, y que las resistencias tendrán los valores exactos para que se cumpla relación (30). Evidentemente, esto es una simplificación, y cualquier desviación de esta introducirá errores o desviaciones en el comportamiento esperado.

Comenzando por el caso de las resistencias, aunque se seleccionen valores normalizados que cumplan exactamente con la ecuación (30), las resistencias tienen una tolerancia, la cual introducirá un error en el ratio de la relación (30). Para diferenciar los valores ideales de los valores normalizados, se señalarán los valores normalizados con un apostrofe. Así, si se utilizan los valores normalizados de los componentes en la ecuación (30) y se introduce un factor que agrupe los efectos del error de ratio del puente, esta se puede reescribir como:

$$\boxed{(1-\varepsilon_D)\frac{R_{2A}'+R_{2B}'}{R_1'}=\frac{R_4'}{R_3'}} \quad (33)$$

Donde ε_D , es el factor de desbalanceo del puente de resistencias. Sustituyendo la ecuación (33) en en la expresión de la resistencia de salida (29):

$$\boxed{R_O'=\frac{(R_1'+R_{2A}')R_{2B}'}{\varepsilon_D(R_{2A}'+R_{2B}')}} \quad (34)$$

Aunque los valores de las resistencias normalizados coincidan con los ideales, debido a la tolerancia de las resistencias, el término ε_D no será nulo, luego, la resistencia de salida tendrá un valor finito. Esto provocará que la corriente de salida, i_o , se vea influenciada por el valor de la impedancia de carga, Z_L . Combinando las ecuaciones (25) y (26), se obtiene una ecuación que recoge este hecho:

$$i_o=\frac{A_i}{1+\frac{Z_L}{R_O}}v_i \quad (35)$$

Descripción de la solución propuesta

Todas las ecuaciones anteriores se basan en el supuesto de que el AO tiene realimentación negativa. Sin embargo, si se observa la Figura 29 se ve que, aparte del camino de realimentación negativa formado por las resistencias R_3 y R_4 , existe un camino de realimentación positiva, formado por R_{2A} , R_{2B} , Z_L , y R_1 . Por tanto, es necesario determinar bajo qué condiciones existirá realimentación negativa. Esto permitirá asegurar que el amplificador de transconductancia es estable, y también permitirá conocer su respuesta en frecuencia, así como los requisitos de ancho de banda del AO.

Teniendo en cuenta el modelo de pequeña señal para el transistor bipolar expuesto en [36], para hallar la ganancia de lazo, $T(jf)$, se aplicará el método presentado en [35]. Para ello, en primer lugar se anulan la entrada y todas las fuentes de tensión y corriente independientes. En segundo lugar, se desconecta la fuente de tensión dependiente del AO de la salida del mismo, y se inyecta una señal v_t .

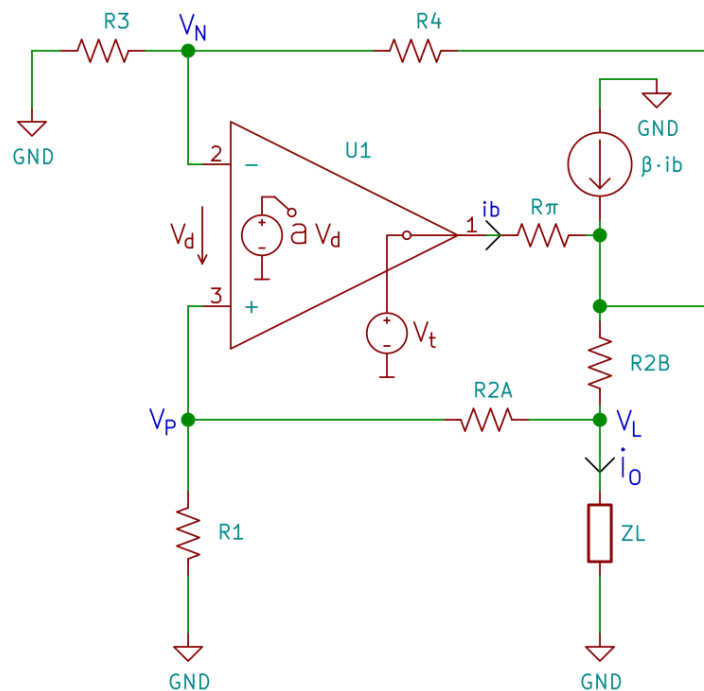


Figura 31: Modelo para el cálculo ganancia de lazo, $T(jf)$

Dado que el AO usado será un amplificador operacional CMOS, sus entradas serán transistores mosfet, por lo que la impedancia de entrada será del orden de $G\Omega$.

Esto permite considerar las entradas como un circuito abierto. Entonces, aplicando el método de nudos, se obtiene el siguiente sistema de ecuaciones:

$$\begin{cases} 0 = \beta i_b + i_b + \frac{v_L - v_O}{R_{2B}} + \frac{0 - v_O}{R_3 + R_4} \\ i_b = \frac{v_t - v_O}{r_\pi} \\ 0 = \frac{-v_L}{R_1 + R_{2A}} + \frac{v_O - v_L}{R_{2B}} - \frac{v_L}{Z_L} \\ r_\pi = \frac{\beta}{g_m} = \frac{V_T}{I_{CQ}} \end{cases} \quad \text{La ganancia de lazo se calcula cómo: } T = -\frac{a v_d}{v_t}$$

V_T es aproximadamente 25mV a 25°C, y I_{CQ} es el punto de operación de la corriente de colector del transistor Q_1 , como se verá más adelante, $I_{CQ} = 52,5\text{mA}$. Operando y teniendo las siguientes simplificaciones: $\beta \gg 1 \rightarrow \beta + 1 \approx \beta$, $R_{2B} \ll (R_3 + R_4) \rightarrow R_{2B} \parallel (R_3 + R_4) \approx R_{2B}$, $(R_1 + R_{2A}) \gg 1$ y $Z_L \gg 1 \rightarrow (R_1 + R_{2A}) \parallel Z_L \approx 1$ y $gm \cdot R_{2B} \gg 0 \rightarrow 1/gm \cdot R_{2B} \approx 0$, la ganancia de lazo es:

$$T = a \left[\frac{R_3}{R_3 + R_4} - \frac{R_1 Z_L}{R_{2B} + (R_1 + R_{2A})(R_{2B} + Z_L)} \right]$$

Donde a , es la ganancia de lazo abierto del amplificador operacional. Para un amplificador compensado por polo dominante:

$$a(jf) = \frac{a_0}{1 + \frac{jf}{f_b}} \quad (36) \quad \text{donde: } a_0: \text{ganancia DC del AO.} \\ f_b: \text{frecuencia de corte del polo dominante del AO.}$$

Combinando esta expresión con la anterior, se obtiene la ganancia de lazo:

$$\begin{cases} T(jf) = \frac{T_0}{1 + \frac{jf}{f_b}} \\ T_0 = a_0 \left[\frac{R_3}{R_3 + R_4} - \frac{R_1 Z_L}{R_{2B} + (R_1 + R_{2A})(R_{2B} + Z_L)} \right] \end{cases} \quad (37)$$

Para que el amplificador sea estable, se debe que cumplir que:

$$\angle T(jf_x) < -180^\circ \quad \text{para la frecuencia } f_x \text{ tal que } |T(jf_x)| = 1$$

El polo del AO introducirá un desfase de -90° por lo que mientras $T_0 > 0$, la fase en la frecuencia unitaria de la ganancia de lazo, f_x , será siempre menor a -180° , por lo que el amplificador será estable. En otras palabras, mientras $T_0 > 0$ el amplificador tendrá realimentación negativa y será estable.

Observando la ecuación (37), se puede comprobar que el signo T_0 depende del valor de la impedancia de salida y de las resistencias R_1 , R_{2A} y R_{2B} . De hecho, esta ecuación implica que si se elimina R_1 , esto es, se deja en circuito abierto, $R_1 = \infty$, por lo que $T_0 \rightarrow -\infty < 0$ lo que causaría que el

amplificador de transconductancia dejara de ser estable. Este es el motivo por el cual se situó el interruptor analógico que controla la activación y desactivación de los sectores entre V_p y masa. Si se activa el interruptor, $R_1 = 0 \rightarrow T_0 > 0$, y por tanto, el amplificador estará realimentado negativamente y será estable, y además, se cumplirá que $V_N = V_P = 0V$, por lo que, a través de R_1 no debe circular ninguna corriente: $i_0 = 0$.

Tal como se recoge en [35], la ganancia de lazo cerrado, A , esto es, la ganancia real del amplificador de transconductancia, está dada por la ecuación:

$$A = \frac{A_0}{1 + \frac{1}{T}} + \frac{a_{ft}}{1+T} \quad (38) \quad \text{donde: } A_0: \text{ es la ganancia ideal de lazo cerrado:}$$

$$A_0 = \lim_{a \rightarrow \infty} \frac{i_O}{v_i}$$

T : es la ganancia de lazo.

a_{ft} : es la ganancia feed-forward.

$$a_{ft} = \lim_{a \rightarrow 0} \frac{i_O}{v_O}$$

Para esta configuración, el valor de la ganancia *feed-forward* es despreciable, por lo que la ganancia de lazo cerrado se expresa como:

$$A = \frac{A_0}{1 + \frac{1}{T}} \quad (39)$$

La ganancia ideal de lazo cerrado relaciona v_i y i_O cuando la ganancia de lazo abierto, a , tiende a infinito. En estas condiciones, se cumple que $V_P = V_N$. Las ecuaciones (28), (29), y (34), se han deducido bajo el supuesto de que el AO es ideal ($a = \infty$), y que debido a la realimentación negativa $V_P = V_N$, por lo que pueden utilizarse para calcular A_0 . Para ello, teniendo en cuenta la ecuación (35):

$$A_0 = \frac{i_o}{v_i} = \frac{A_i}{1 + \frac{Z_L}{R_O}} \quad (40)$$

Entonces, sustituyendo este resultado en (39) y combinando la ecuación (37) con este, se obtiene la expresión completa de la ganancia de lazo cerrado del amplificador de transconductancia:

$$A(jf) = \frac{A_i}{\left(1 + \frac{Z_L}{R_O}\right) \left(1 + \frac{1}{T_0}\right) \left(1 + \frac{jf}{f_b(1+T_0)}\right)} \quad (41)$$

Si la impedancia de carga, Z_L , es puramente resistiva, la respuesta en frecuencia de la ganancia del amplificador de transconductancia, $A(jf)$, solo está limitada por un polo cuya frecuencia crítica es:

$$f_{p1} = f_b(1+T_0)$$

Así, para un amplificador operacional compensado por polo dominante, cuya frecuencia de ganancia unitaria, f_t , o producto de ganancia-ancho de banda (GBP), está relacionada con su frecuencia de polo, f_b , por la expresión:

$$GBP = f_t = a_0 f_b \quad (42)$$

Entonces, dado que la frecuencia crítica del polo, f_{p1} , es igual a la frecuencia de corte a -3 dB del amplificador de transconductancia, el ancho de banda se relaciona con el GBP del AO como:

$$GBP = \frac{a_0 BW}{1 + T_0} = \frac{BW}{\frac{1}{a_0} + \frac{T_0}{a_0}} \quad \text{Como } a_0 \gg 1 \quad GBP = \frac{BW}{\frac{T_0}{a_0}}$$

$$GBP = \frac{BW}{\frac{R_3}{R_3 + R_4} - \frac{R_1 Z_L}{R_{2B} + (R_1 + R_{2A})(R_{2B} + Z_L)}} \quad (43)$$

Esta ecuación permite calcular el GBP que debe tener el AO para un determinado ancho de banda del amplificador de transconductancia.

Utilizando el modelo de pequeña señal para el diodo presentado en [36], la impedancia de carga Z_L puede aproximarse como una resistencia de pequeña señal, r_d . Tomando dos puntos de la curva *Forward Current vs. Forward Voltage*, de la hoja de datos del emisor TSAL6400, la impedancia de carga Z_L es:

$$\begin{cases} r_d = \frac{1,35V - 1,1V}{100mA - 2,5mA} \approx 2,6\Omega & \rightarrow & Z_L = 2 \cdot 2,6\Omega = 5,2\Omega \\ Z_L = 2 \cdot r_d \end{cases}$$

Con este resultado, ya es posible calcular el valor de la ganancia de lazo, $T(jf)$ y de la ganancia de lado cerrado $A(jf)$ del amplificador de transconductancia.

Una vez expuesto todo lo anterior, el error relativo de ganancia del amplificador operacional, ε_{ATC} , esto es, el error tomando en cuenta todas las desviaciones introducidas por la normalización de los valores de las resistencias, la tolerancia, y las limitaciones de ancho de banda y ganancia del AO respecto, del valor ideal es:

$$\varepsilon_{ATC} = \frac{|A(jf)' - A_i|_{P.B.}}{A_i|_{P.B.}} \quad (44)$$

Ahora, antes de proceder a la selección de la ganancia del amplificador de transconductancia, y de los valores de componentes, se debe analizar el rango de salida del dinámico del AO. El rango dinámico de salida del AO está limitado por la tensión de alimentación, y por la corriente de cortocircuito del AO.

La tensión de alimentación limitará la excursión máxima que puede tener la salida respecto de los valores de los raíles de alimentación del AO, en este caso, se utilizará un AO *rail-to-rail*, por lo que

Descripción de la solución propuesta

la excursión de salida alcanzará un valor muy cercano a los valores de tensión de la alimentación, con una diferencia del orden de decenas de mili voltios.

Respecto de la corriente de cortocircuito del AO, esta es la corriente máxima que el AO puede proporcionar a su salida. En esta aplicación se utiliza el transistor Q_1 para aumentar la corriente de salida del amplificador de transconductancia por encima del valor de la corriente de cortocircuito del AO.

Dado que el amplificador de transconductancia opera como un amplificador en clase A, el valor medio de la señal sinusoidal no es nulo. Aprovechando esta circunstancia, para simplificar el diseño y abaratar el costo, la alimentación de esta placa será unipolar. Así, el generador sinusoidal deberá entregar la señal sinusoidal con un valor medio suficiente como para evitar el recortado de la señal, y con un amplitud lo más grande posible para aprovechar todo el rango dinámico de los AO.

La tensión de alimentación será de 5V. Fijando un valor mínimo de 100mV para la señal de excitación sinusoidal, la señal sinusoidal tendrá las siguientes características:

$$v_{EXC}(t) = 2 \cos(2\pi \cdot 10^4 t) V + 2,1 V \quad (45)$$

Amplitud: $A_{EXC} = 2V$.

Valor medio: $[V_{EXC}(t)] = 2,1V$

Frecuencia: 10kHz

Entonces, para garantizar que los leds emisores operan en un rango de corriente seguro, se fija que la corriente media de salida sea de aproximadamente, 50mA, un 50% menos que el valor máximo de corriente no pulsada, 100mA. Así, eligiendo que la irradiancia pico de la señal sinusoidal emitida sea de 50 mW/sr, utilizando ganancia del emisor IR, dada en la ecuación (24), la ganancia del amplificador de transconductancia debe ser:

$$G_{ATC} = \frac{I_E}{G_D \cdot A_{EXC}} = \frac{50 \frac{mW}{sr}}{1 \frac{W}{A sr} \cdot 2V} = 0,025 \frac{A}{V} \rightarrow \boxed{G_{ATC} = 0,025 \Omega^{-1}}$$

La corriente media de salida es:

$$[i_O] = G_{ATC} \cdot [V_{EXC}(t)] = 0,025 \Omega^{-1} \cdot 2,1 V = 52,5 mA$$

Los valores máximos y mínimos de la corriente de salida son:

$$i_{O|_{max}} = G_{ATC} \cdot V_{EXC}(t)|_{max} = 0,025 \Omega^{-1} \cdot 4,1 V = 102,5 mA$$

$$i_{O|_{min}} = G_{ATC} \cdot V_{EXC}(t)|_{min} = 0,025 \Omega^{-1} \cdot 0,1 V = 2,5 mA$$

Utilizando la curva *Forward Current vs. Forward Voltage* de la hoja de datos de los leds emisores [33], el valor máximo de tensión en la carga, V_L , es:

$$v_{l|_{max}} = 2 \cdot \text{funcionDiodo}(i_{O|_{max}}) = 2 \cdot \text{funcionDiodo}(102,5 mA) = 2 \cdot 1,3 V = 2,6 V$$

Ahora, para aproximar la resistencia de salida del amplificador de transconductancia a su valor ideal ($R_o = \infty$), se seleccionarán los valores de las resistencias de forma que cumplan la relación (30), esto es, que el puente de resistencias este balanceado. Así, la ganancia del amplificador de transconductancia estará dada por la ecuación (31):

$$G_{ATC} = A_i|_{P.B.} \rightarrow G_{ATC} = \frac{R_4}{R_3 R_{2B}}$$

Reordenando la expresión anterior:

$$\frac{R_4}{R_3} = R_{2B} G_{ATC}$$

El valor de la resistencia R_{2B} debe seleccionarse de forma que sea lo suficientemente grande como para que los valores de resistencia parásita de las pistas puedan ser despreciables, y lo suficientemente pequeño como para que la salida del AO permanezca dentro de su rango dinámico. Para $R_{2B} = 10\Omega$, la tensión a la salida del AO será:

$$V_{O(AO)}|_{max} = V_{BE} + i_O|_{max} \cdot R_{2B} + v_l|_{max} = 0,7V + 102,5mA \cdot 10\Omega + 2,6V = 4,325V$$

Este valor esta varias centenas de mili voltios por debajo del valor de tensión del rail positivo, 5V, por lo que será realizable con un AO *rail-to-rail*. El ratio entre R_4 y R_3 es:

$$\frac{R_4}{R_3} = R_{2B} \cdot G_{ATC} = 10\Omega \cdot 0,025\Omega^{-1} = 0,25$$

Utilizando el programa *resrat* para minimizar el error ratio en la selección de resistencias normalizadas a la serie E-96 (tolerancia 1%), se eligen los valores de R_4 y R_3 . Para el valor exacto de ratio 0,25: $R_4 = 2,55k\Omega$ y $R_3 = 10,2k\Omega$. Para calcular R_1 y R_{2A} se utiliza la relación (30), asignando valores normalizados de la serie 1%. Los valores normalizados de las resistencias para la serie E-96 son:

$$\begin{array}{lll} R_1 = 52,3k\Omega \pm 1\% & R_{2B} = 10\Omega \pm 1\% & R_4 = 2,55k\Omega \pm 1\% \\ R_{2A} = 13k\Omega \pm 1\% & R_3 = 10,2k\Omega \pm 1\% & \end{array}$$

Con estos valores, ya es posible calcular la magnitud de ganancia de lazo, T_o , y por tanto, obtener una estimación del GBP del AO. La frecuencia de la señal de excitación es de 10kHz, tomando una década de margen, el ancho de banda mínimo que el amplificador de transconductancia debe tener es de 100kHz. Utilizando la ecuación (43), el GBP mínimo es:

$$GBP|_{min} = \frac{100kHz}{\frac{10,2k\Omega}{10,2k\Omega + 2,55k\Omega} - \frac{52,3k\Omega \cdot 5,2\Omega}{10\Omega + (52,3k\Omega + 13k\Omega)(10\Omega + 5,2\Omega)}} = 190,1kHz$$

Teniendo en cuenta esto, que el amplificador operacional debe ser *rail-to-rail*, compensando por polo dominante, y que debe tener entradas CMOS, se elige el amplificador operacional TLV9064 del fabricante *Texas Instruments*. Este integrado empaqueta cuatro amplificadores operacionales

Descripción de la solución propuesta

en un mismo encapsulado, lo que reducirá el espacio requerido en el diseño. Sus características para las condiciones de operación del diseño, según la hoja de características [37], son:

Voltaje de alimentación recomendado $[(V+) - (V-)]$: 1,8V a 5,5V.

Rango de voltaje de salida: -V a +V.

Ganancia DC típica, a_0 : 130dB $\rightarrow 3,16 \cdot 10^6$ V/V.

Ganancia DC mínima, $a_{0\text{min}}$: 104dB $\rightarrow 158 \cdot 10^3$ V/V.

GBP: 10MHz.

Tipo de etapa de entrada: CMOS.

Voltaje de offset a la entrada típico, V_{OS} : 300uV.

Corriente de salida en cortocircuito, I_{SC} : ± 50 mA.

Corriente quiescente máxima, I_{Qmax} : 750uA.

La influencia de parámetros como la corriente de bias o la tensión de offset han sido analizados, no obstante, dado que su influencia es despreciable, dicho análisis no se reproducirá aquí.

El transistor Q_1 elegido es el BC817-40 [37], el cual tiene una corriente máxima de colector, I_{CQ} , de 500mA, una ganancia de corriente mínima, β , de 250 y una tensión base-emisor, V_{BE} , máxima de 1,2V, la cual aunque es mayor que la prevista, mantiene la salida del AO dentro del rango de operación. Estas características son suficientes para este diseño.

En este punto, todos los elementos del diseño han sido seleccionados, por lo que utilizando los macromodelos del AO TLV9064, transistor BC817-40 y led TSAL6400 proporcionados por los fabricantes, se realizará una simulación en LTSpice para corroborar el análisis teórico. El archivo de simulación puede encontrarse en la ruta *Hardware/SWR-UPM/Diseño_sensor_R&B/Simulaciones/RyB_emisor_sensor/Amplificador_transconductancia/transitorio/atc_transitorio.asc*, en los archivos que acompañan a este proyecto. Para una señal de excitación igual a (45), con componentes con valores nominales, la gráfica de la corriente de salida, i_o , obtenida es:

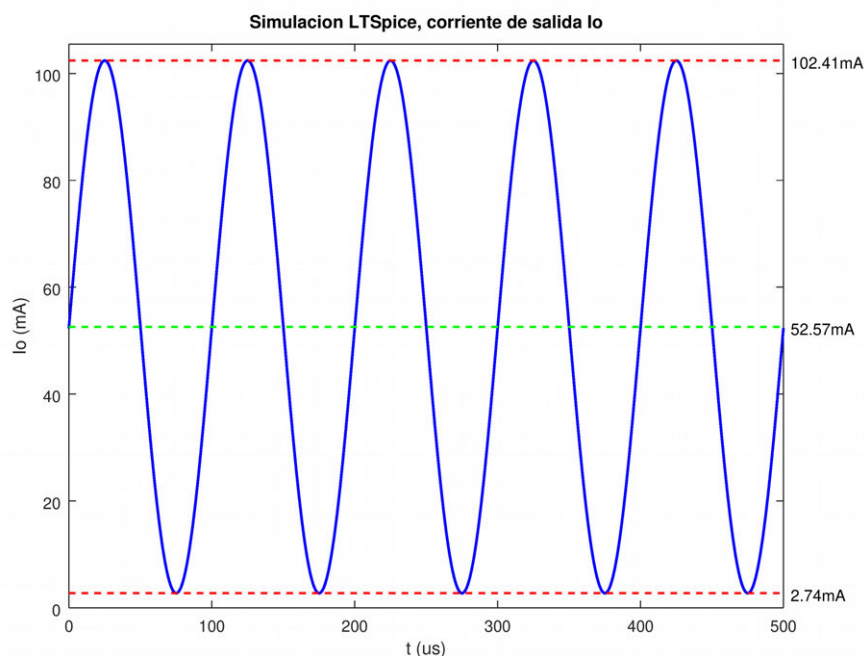


Figura 32: Resultados de la simulación, i_o

Como se puede observar, el valor máximo: 102,75mA, mínimo: 2,74mA y el valor medio 52,57mA de la corriente de salida, i_o , concuerdan con lo esperado.

Respecto al efecto de la tolerancia de los componentes sobre la ganancia, se han realizado dos análisis para estimar el error relativo de ganancia, ϵ_{ATC} .

En primer lugar, se ha realizado un análisis de error por el método de Monte Carlo, utilizando la función *mc* de la que dispone *LTspice* y una hoja de cálculo. El archivo de simulación, y los detalles de este procedimiento pueden consultarse en el fichero presente en la ruta: *Hardware/SWR-UPM/Diseño_sensor_R&B/Simulaciones/RyB_emisor_sensor/Amplificador_transconductancia/AMC_tolerancia*, en los archivos que acompañan a este proyecto. La variación del error de ganancia, ϵ_{ATC} , para una frecuencia de 10kHz, con una muestra de 1001 simulaciones se presenta a continuación:

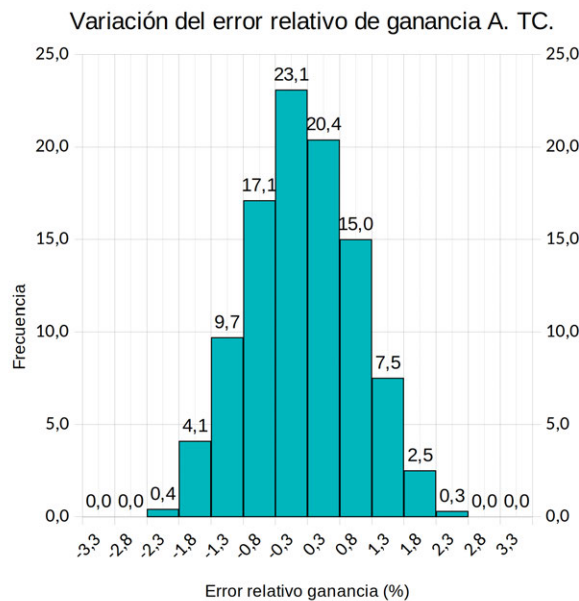


Figura 33: Variación del error de ganancia del amplificador de tranconductancia, Monte Carlo N=1001

A la vista de estos datos, para el 99,4% de las muestras, el error relativo de ganancia, ϵ_{ATC} , está contenido en el intervalo:

$$\epsilon_{ATC}(j10kHz)|_{MC,99,4\%} \in (-1,8\%, 2,3\%) \quad (46)$$

En segundo lugar, se realizado un análisis de error para el peor caso (*Worst Case Analysis*), asignando a el valor de cada componente el valor máximo y mínimo que puede darse debido la tolerancia, para todas las combinaciones posibles (33 combinaciones posibles). El archivo de simulación, y los detalles de este procedimiento pueden consultarse en el fichero ubicado en la ruta: *Hardware/SWR-UPM/Diseño_sensor_R&B/Simulaciones/RyB_emisor_sensor/Amplificador_transconductancia/AWC_tolerancia*. Los resultados son:

$$\epsilon_{ATC}(j10kHz)|_{WC} \in (-3,3\%, 2,7\%) \quad (47)$$

4.1.4.3 Matriz de activación

Dada la gran cantidad de periféricos que la plataforma ARGOS implementa, las líneas de control del microcontrolador principal son un recurso escaso. Con el objetivo de minimizar el número de líneas necesarias para controlar la activación de los sectores, se optó por implementar un circuito combinacional de control, el cual ahorraría tres líneas de control.

Cada sector dispone de un interruptor controlado por tensión, como se puede ver en la Figura 29. Estos interruptores reciben una señal de control, *Ex*, que permite activar o desactivar la emisión del correspondiente sector. Cuando *Ex* está a nivel alto, el sector está desactivado, y cuando *Ex* está a nivel bajo, el sector está activo, es decir, emitiendo. Esto permite utilizar cada sector como un sensor de reflexión por infrarrojos.

Por otro lado, cuando el emisor opera como baliza de infrarrojos, para que otros robots puedan localizarlo y calcular su posición, todos los sectores deben estar activos (emitiendo). Así pues, la lógica de control debe disponer de un decodificador con habilitación, que permita la selección del sector activo, y la desactivación cuando todos los sectores deben estar desactivados; y un conjunto de puertas NOR, cuyas entradas sean las salidas del decodificador y una línea de control que habilite todos los sectores. La lógica de control implementada es:

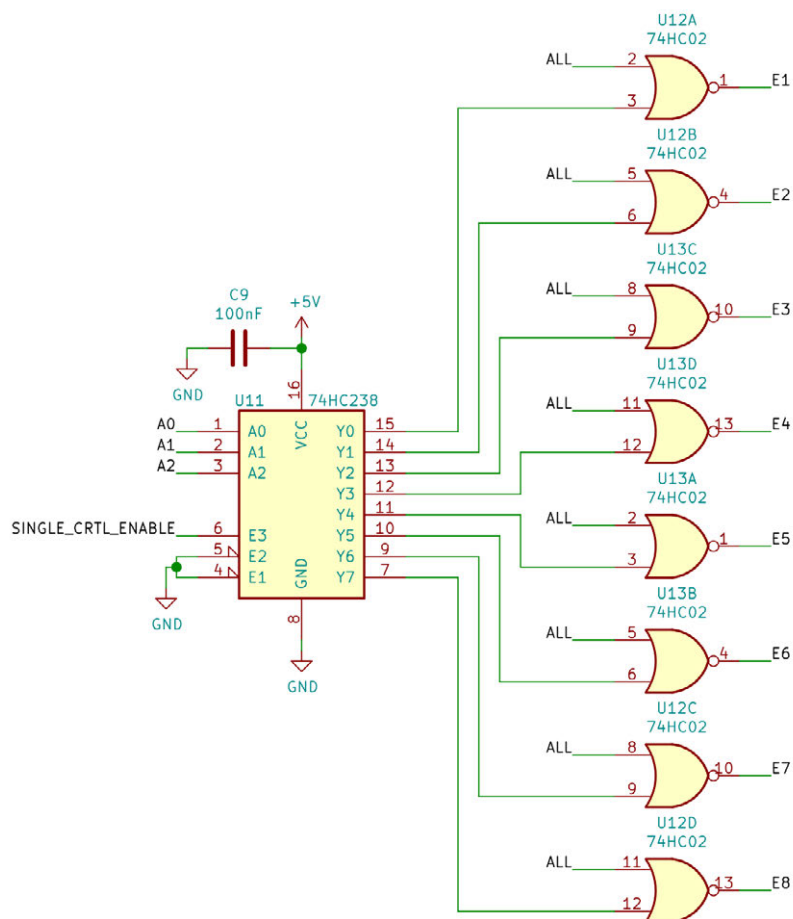


Figura 34: Lógica de control del emisor del sensor R&B

Las líneas A0, A1, A2 establecen que sector va a activarse, la línea SINGLE_CRTL_ENABLE, habilita el decodificador, 74HC238, y la línea ALL, cuando está a nivel alto, activa todos los sectores. La función lógica se resume en la siguiente tabla:

Entradas del módulo					Líneas de activación internas								Sectores activos
ALL	SINGLE_CRTL_ENABLE	A0	A1	A2	E1	E2	E3	E4	E5	E6	E7	E8	
0	0	X	X	X	1	1	1	1	1	1	1	1	Ninguno
0	1	0	0	0	0	1	1	1	1	1	1	1	Sector 1
0	1	0	0	1	1	0	1	1	1	1	1	1	Sector 2
0	1	0	1	0	1	1	0	1	1	1	1	1	Sector 3
0	1	0	1	1	1	1	1	0	1	1	1	1	Sector 4
0	1	1	0	0	1	1	1	1	0	1	1	1	Sector 5
0	1	1	0	1	1	1	1	1	1	0	1	1	Sector 6
0	1	1	1	0	1	1	1	1	1	1	0	1	Sector 7
0	1	1	1	1	1	1	1	1	1	1	1	0	Sector 8
1	X	X	X	X	0	0	0	0	0	0	0	0	Todos

Tomando la planta del robot, los sectores están dispuestos de la siguiente forma:

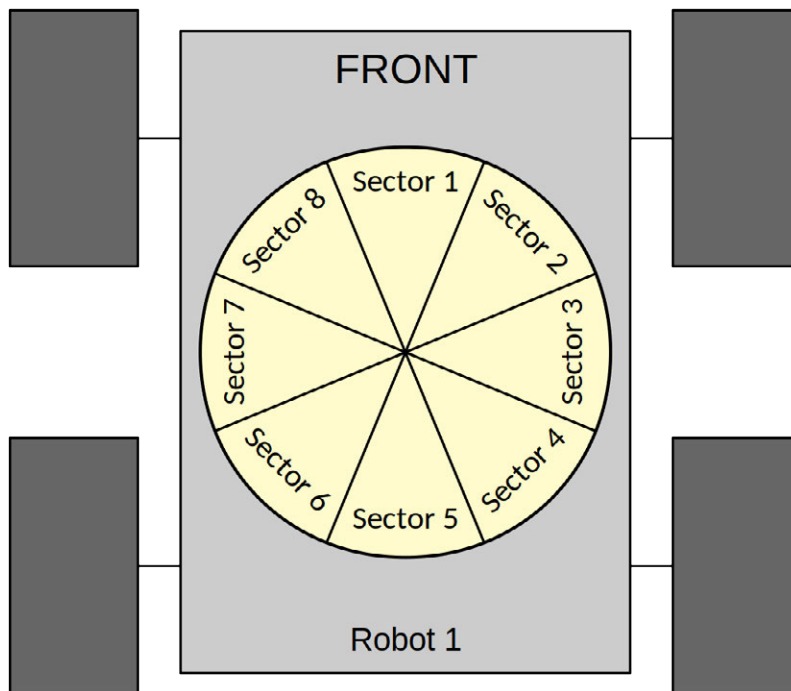


Figura 35: Disposición de sectores, sensor R&B

4.1.4.4 Generador de onda sinusoidal

El generador de onda sinusoidal debe generar una señal de excitación que cumpla la ecuación (45), reproducida a continuación:

$$v_{EXC}(t) = 2 \cos(2\pi \cdot 10^4 t) V + 2,1 V$$

Para generar una onda sinusoidal hay diversas opciones. La más analógica, es un oscilador sinusoidal, sin embargo, los osciladores no garantizan con exactitud que la señal generada tenga una amplitud determinada. Por tanto, si se usa un oscilador, para garantizar la exactitud de la amplitud de la señal, se debe diseñar además un control automático de ganancia. Tanto el oscilador como el control automático de ganancia están sometidos a los efectos de las tolerancias y coeficientes térmicos de los componentes, por lo que su diseño es complejo, y para garantizar cierta precisión, requiere de componentes con tolerancias muy bajas, lo que aumenta el costo.

La tecnología digital, por otro lado, es mucho menos sensible a las variaciones y tolerancias. La mejor opción, sería utilizar generador DDS (*Direct Digital Synthesis*). Un DDS es un componente que incorpora una memoria, un controlador y un DAC en un mismo encapsulado, y que permite generar una señal arbitraria a partir de los datos almacenados en la memoria. Dada su arquitectura, las velocidades de muestreo son elevadas, por lo que los requisitos del filtro de reconstrucción se rebajan considerablemente.

El hardware de este proyecto se diseñó a principios del año 2020, y debido a la crisis de abastecimiento de chips, conseguir un DDS era extremadamente difícil, y si algún distribuidor disponía de stock, los precios eran de al menos un orden de magnitud superior al precio normal. Por ese motivo, se descartó el uso de un DDS. Otras soluciones similares, como utilizar un microcontrolador y un DAC para implementar un DDS discreto, también se descartaron por los mismos motivos: falta de stock, y precios inasumibles.

Así pues, y dado que no había muchas más opciones, se optó por un diseño poco ortodoxo para el generador sinusoidal. Este diseño aprovecha que el generador sinusoidal debe producir una señal con un valor medio no nulo.

La idea se expone en el siguiente diagrama de bloques:

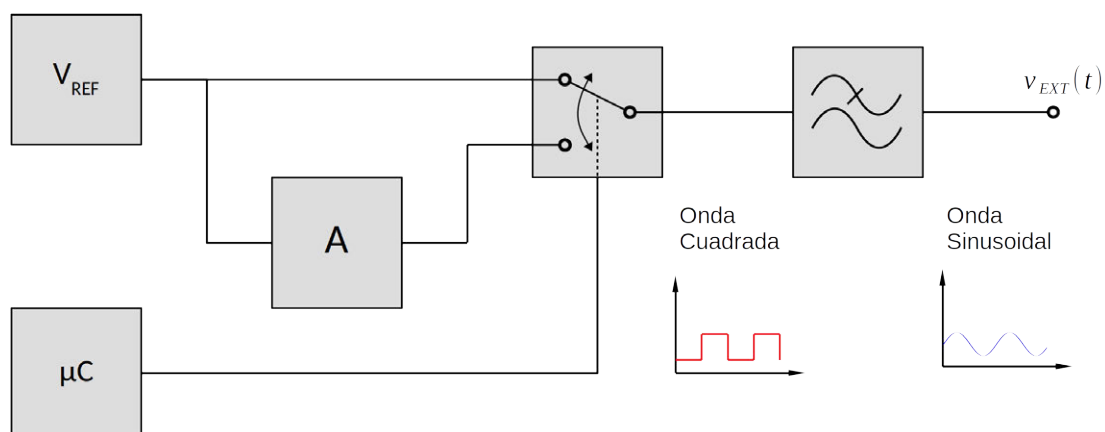


Figura 36: Generador de onda sinusoidal

Para generar la señal sinusoidal se usa un filtro paso-banda, el cual filtra todas las componentes en frecuencia de la onda cuadrada salvo la componente DC y el primer armónico. Dado que el valor de la componente DC está dado por el valor medio de la onda cuadrada, el cual depende del nivel mínimo de la onda cuadrada y de la amplitud de la misma, y el valor del primer armónico depende de la amplitud de la onda cuadrada, con los valores adecuados de V_{REF} , y del atenuador A , se puede generar una señal sinusoidal con las características deseadas.

Además, para garantizar que la frecuencia es estable, la frecuencia del conmutador que conforma la onda cuadrada está controlada por uno de los *timers* del microcontrolador, el cual, en última instancia, depende de la frecuencia del cristal de cuarzo del oscilador del microcontrolador. Dado que el cristal de cuarzo tiene unas excelentes características exactitud y estabilidad, la frecuencia de la onda cuadrada, y por tanto, la frecuencia de la señal sinusoidal generada, también retendrá dichas cualidades.

Antes de presentar el diseño de los circuitos que implementan esta solución, se expondrán las ecuaciones que determinan las características que debe cumplir la señal cuadrada y la función de transferencia del filtro paso-banda, para generar una señal de excitación con la forma:

$$v_{EXT}(t) = A_{EXC} \cos(2\pi \cdot f_{EXC} t) + M_{EXC} \quad (48) \quad \text{donde: } A_{EXT}: \text{ amplitud de la señal sinusoidal.}$$

M_{EXC} : valor medio de la señal de excitación.

f_{EXC} : frecuencia de la señal sinusoidal.

La señal cuadrada tiene las siguientes características:

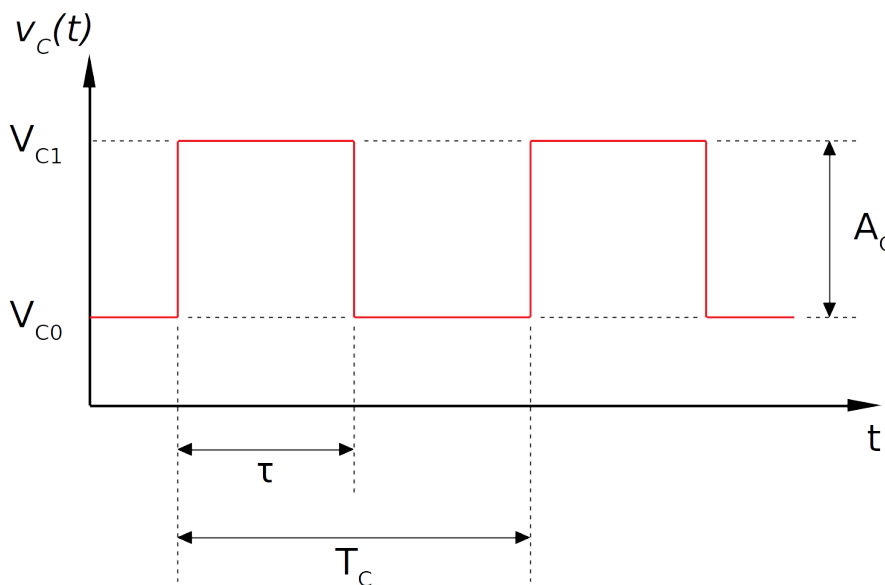


Figura 37: Características de la onda cuadrada

Esta señal puede componerse como la suma de dos funciones: una señal de valor constante e igual a V_{C0} , y una tren de pulsos cuadrados, con un ancho de pulso τ y un periodo T_C .

$$v_C(t) = v_{TPC}(t) + v_{C0}(t)$$

Utilizando la tabla de transformadas recogida en [39], los coeficientes del desarrollo en series de Fourier para estas dos señales son:

$$v_{C0}(t) = V_{C0} \rightarrow \text{Coeficientes del DSF} \equiv \begin{cases} a_0 = V_{C0} \\ a_k = 0, \quad \forall k \neq 0 \end{cases}$$

$$v_{TPC}(t) = \begin{cases} A_C, & |t| < \frac{\tau}{2} \\ 0, & \frac{\tau}{2} < |t| < \frac{T_C}{2} \end{cases} \rightarrow \text{Coeficientes del DSF} \equiv \begin{cases} a_0 = \frac{A_C \tau}{T_C} \\ a_k = \frac{A_C}{\pi k} \text{sen}\left(\frac{\pi k \tau}{T_C}\right), \quad \forall k \neq 0 \end{cases}$$

Si se considera que la señal V_{C0} es periódica, con periodo T_C , se pueden combinar linealmente los coeficientes del DSF. Además, teniendo en cuenta que el ancho de pulso τ , puede relacionarse con el ciclo de trabajo (D) de la señal cuadrada como: $\tau = DT$, los coeficientes del DSF para la señal $v_c(t)$ son:

$$v_c(t), \text{ coeficientes del DSF} \equiv \begin{cases} a_0 = V_{C0} + A_C D \\ a_k = \frac{A_C}{\pi k} \text{sen}(\pi k D), \quad \forall k \neq 0 \end{cases} \quad (49)$$

Idealmente, el filtro paso-bajo solo conservará la componente continua, y el armónico fundamental. Así, la señal de excitación solo tendrá tres componentes en frecuencia, cuyos coeficientes, si la ganancia el filtro paso-bajo es H_0 , son:

$$v_{EXC}(t), \text{ coeficientes del DSF} \equiv \begin{cases} a_0 = H_0(V_{C0} + A_C D) \\ a_1 = \frac{H_0 A_C}{\pi} \text{sen}(\pi D) \\ a_{-1} = \frac{H_0 A_C}{\pi} \text{sen}(\pi D) \end{cases}$$

Aplicando la ecuación de síntesis del DSF a los coeficientes de la señal $v_{EXC}(t)$, se obtiene:

$$v_{EXT}(t) = \frac{2 H_0 A_C}{\pi} \cos\left(\frac{2 \pi}{T_C} t\right) + H_0(V_{C0} + A_C D) \quad (50)$$

Identificando términos con la expresión (48), se establece una relación entre la ganancia del filtro paso-banda y las características de la onda cuadrada, con el valor de amplitud de la señal sinusoidal y el valor medio de la señal de excitación, M_{EXC} :

$$\begin{cases} A_{EXC} = \frac{2 H_0 A_C}{\pi} \\ M_{EXC} = H_0(V_{C0} + A_C D) \\ f_{EXC} = \frac{2 \pi}{T_C} = f_C \end{cases} \quad (51)$$

Si se observa el diagrama de bloques de la Figura 36, se puede establecer que la relación entre V_{REF} , el valor del atenuador A y las características de la onda cuadrada, $v_c(t)$ son:

$$\begin{cases} V_{C0} = V_{REF} A \\ A_C = V_{REF} (1 - A) \end{cases}$$

Sustituyendo este resultado en las ecuaciones dadas en (51), y dado que el ciclo de trabajo, D , solo afecta al valor medio de la señal de excitación, este se puede fijar arbitrariamente al 50% ($D=0,5$), se obtienen las siguientes ecuaciones de diseño:

$$\begin{cases} A_{EXC} = \frac{2V_{REF}}{\pi} H_0 (1 - A) \\ M_{EXC} = \frac{V_{REF}}{2} H_0 A \end{cases} \quad (52)$$

Resolviendo este sistema de ecuaciones, se obtienen los valores de ganancia para el filtro paso-bajo, H_0 , y atenuación, A . Para una tensión de referencia $V_{REF} = 2,048V$, un valor de amplitud $A_{EXC} = 2V$ y un valor medio de $M_{EXC} = 2,1V$, H_0 y A son:

$$\begin{cases} A = \frac{4M_{EXC} - A_{EXC}\pi}{4M_{EXC} + A_{EXC}\pi} \\ H_0 = \frac{4M_{EXC} + A_{EXC}\pi}{4V_{REF}} \end{cases} \rightarrow \begin{cases} A = \frac{21 - 5\pi}{21 + 5\pi} \\ H_0 = \frac{525 + 125\pi}{512} \end{cases} \quad (53)$$

Una vez hallados estos parámetros se presenta el diseño del generador de onda cuadrada. La función realizada es equivalente a la presentada en la Figura 36:

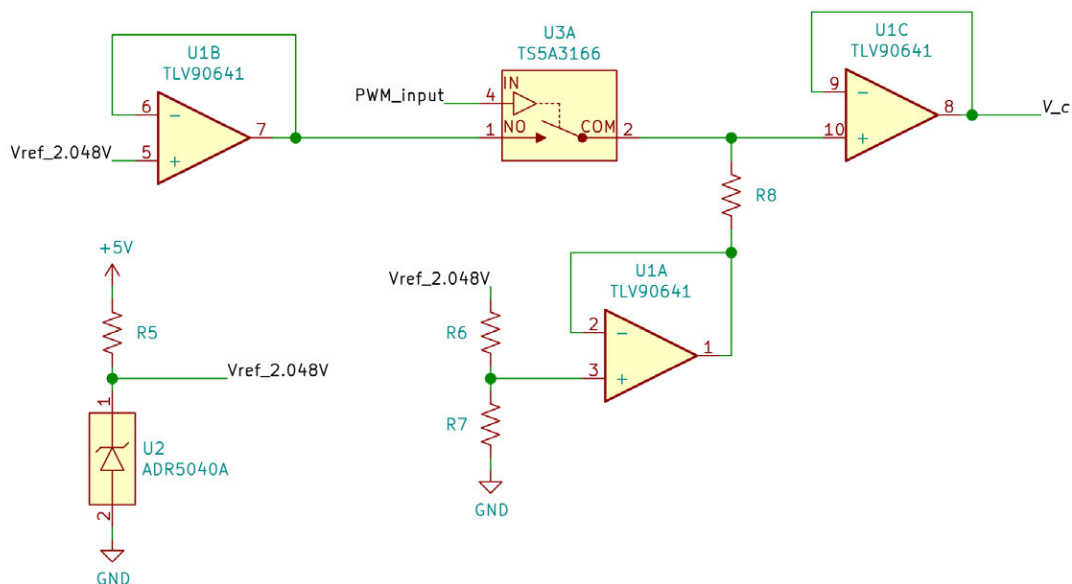


Figura 38: Generador de onda cuadrada

Descripción de la solución propuesta

Los amplificadores operacionales $U1B$, $U1A$ y $U1C$ en configuración seguidor de tensión aíslan la impedancia del divisor de tensión, la tensión de referencia y la impedancia del conjunto de la entrada del filtro. Los AO son los TLV9064, los mismos usados en los amplificadores de transconductancia. En configuración de seguidor de tensión, el ancho de banda es igual al GBP, es decir, 10MHz, por lo que cumplen sobradamente los requisitos para este uso. También se analizó el impacto del *slew-rate*, y es suficiente para esta aplicación.

El funcionamiento del generador de onda cuadrada es el siguiente: cuando el interruptor analógico controlado por tensión, $U3A$, recibe una señal de nivel bajo del *timer*, la tensión en la salida v_c es la misma tensión que hay a la salida del puente de resistencias formado R_6 y R_7 . Cuando el interruptor analógico controlado por tensión, $U3A$, recibe una señal de nivel alto del *timer*, el voltaje de salida, v_c , es igual al voltaje la referencia de tensión. Este es el mecanismo que genera la onda cuadrada.

La atenuación A , debe ser igual a la atenuación del divisor de tensión formado R_6 y R_7 :

$$A = \frac{R_6}{R_6 + R_7} \rightarrow \frac{R_7}{R_6} = \frac{1}{A} - 1 = \frac{1}{\frac{21 - 5\pi}{21 + 5\pi}} - 1 = \frac{10\pi}{21 - 5\pi}$$

Utilizando *resrat*, los resistores seleccionados para la serie E-96 (tolerancia del 1%), con un error de ratio del 0,19%, son:

$$R_6 = 63,4 \text{ k}\Omega \pm 1\% \quad R_7 = 10,7 \text{ k}\Omega \pm 1\%$$

La resistencia R_8 , junto con la resistencia de conducción, $r_{ds(on)}$ del interruptor $UA3$, forman un divisor de tensión, el cual introduce un error en el valor de la tensión V_{C1} , afectando a la amplitud y al valor medio de la señal $v_c(t)$. El interruptor analógico controlador por tensión elegido es el TS5A3166 del fabricante *Texas Instruments*. Uno de los motivos de su elección es su baja $r_{ds(on)}$, del orden de 0,7 Ω para las condiciones de operación [40].

Así, para reducir esta fuente de error, conviene que el valor de R_8 , sea el más bajo posible; sin embargo, un valor bajo aumenta el consumo de corriente. Así pues, se seleccionó el valor de 510 Ω para R_8 , el cual introduciría un error del orden de 0,16% en el valor de V_{C1} , sin producir un consumo excesivo de corriente. Dado que este no es un valor crítico, se escogió un resistor de la serie E-24:

$$R_8 = 510 \Omega \pm 5\%$$

Respecto a la referencia de tensión, se eligió la referencia ADR5040A del fabricante *Analog Devices*, la misma usada en otras placas del robot. Se seleccionó esta por sus buenas características de exactitud: 0.2% (máximo) y bajo coeficiente de temperatura: 10ppm/ $^{\circ}\text{C}$ (típico) [41]. Para polarizar esta referencia de tensión, la hoja de datos específica que la corriente de operación mínima a 25 $^{\circ}\text{C}$ es de 50 μA .

Entonces, fijando una corriente de operación de 80 μA , el valor de la resistencia, normalizada para la serie E-96, es:

$$R_5 = \frac{5V - 2,048V}{80\mu A} = 36,9k\Omega \rightarrow R_6 = 37,4k\Omega \pm 1\%$$

A continuación, se expondrán los detalles del diseño del filtro paso-bajo. Ya se conocer la ganancia del filtro en la banda de paso, H_0 , pero aún falta conocer el tipo de filtro, la frecuencia de corte, la atenuación mínima en la banda de atenuación, y el orden. El filtro debe eliminar todos los armónicos tras el armónico fundamental, de forma que el espectro de la señal de excitación sea lo más parecido al espectro de una señal sinusoidal pura con una componente de continua ($f=0$).

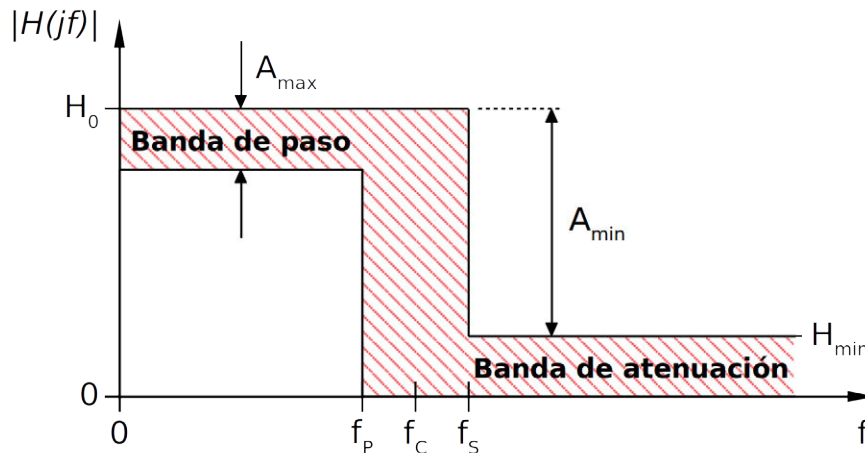


Figura 39: Plantilla de diseño del filtro paso-bajo

El parámetro que mide cual es la distorsión introducida por los armónicos de una señal sinusoidal respecto a su espectro ideal es el THD: *Total Harmonic Distortion*, el cual se define como:

$$THD = \frac{\sqrt{V_2^2 + V_3^2 + V_4^2 + \dots + V_N^2}}{V_1} \quad (54) \quad \text{donde: } V_1: \text{ es el valor RMS del armónico fundamental.}$$

V_2, V_3, V_4, V_n : es el valor RMS del resto de armónicos.

Para una seña sinusoidal pura, esto es, sin distorsión, el THD = 0. El objetivo del siguiente desarrollo es encontrar una relación entre la atenuación mínima de la banda de atenuación del filtro, A_{min} , y el THD. Para ello, según la expresión (49) y aplicando la ecuación de síntesis del desarrollo en series de Fourier, las expresiones para los armónicos de la onda cuadrada para k distinto de 0 son:

$$v_k(t) = \frac{2A_c}{\pi k} \operatorname{sen}\left(\frac{\pi}{2}k\right) \cos(2\pi f_c k t)$$

El armónico fundamental es $k=1$, para los armónicos pares, esto es, $k=2, 4, 6, \dots$ el seno es nulo, por lo que, todos los armónicos pares (excluyendo el término de continua en $k=0$), son nulos. Así, modificando el índice k para tener en cuenta solo los armónicos impares: $k = 2n+1$.

$$v_k(t) = \frac{2A_c}{\pi(2n+1)} \operatorname{sen}\left(\frac{\pi}{2}(2n+1)\right) \cos(2\pi f_c (2n+1)t) = \frac{2A_c}{\pi(2n+1)} \operatorname{sen}\left(\pi n + \frac{\pi}{2}\right) \cos(2\pi f_c (2n+1)t)$$

Descripción de la solución propuesta

Aplicando la relación trigonométrica: $\text{sen}(\alpha + \beta) = \text{sen}(\alpha)\cos(\beta) + \cos(\alpha)\text{sen}(\beta)$

$$v_k(t) = \frac{2A_c}{\pi(2n+1)} (-1)^n \cos(2\pi f_c(2n+1)t)$$

Siendo H_N el valor de ganancia del filtro para cada armónico N , el valor RMS del armónico n -ésimo es:

$$H_n = |H(j(2n+1)f_c)| \rightarrow V_n = \frac{2A_c H_n}{\pi(2n+1)\sqrt{2}} \quad (55)$$

Ahora, teniendo en cuenta la definición de THD, siendo el armónico fundamental $n=0$, y el resto de armónicos $n>0$:

$$THD = \frac{\sqrt{\sum_{n=1}^{\infty} V_n^2}}{V_0} = \frac{\sqrt{\sum_{n=1}^{\infty} \left(\frac{2A_c H_n}{\pi(2n+1)\sqrt{2}} \right)^2}}{\frac{2A_c H_0}{\pi\sqrt{2}}} = \frac{1}{H_0} \sqrt{\sum_{n=1}^{\infty} \left(\frac{H_n}{2n+1} \right)^2}$$

Entonces, si para la banda de atenuación del filtro, la ganancia del filtro para el tercer armónico, esto es, $n=1$, es al menos H_{min} y se cumple:

$$H_{min} \geq H_1 \geq H_2 \geq \dots \geq H_n$$

Se puede aproximar el valor mínimo del THD como:

$$THD_{min} = \frac{1}{H_0} \sqrt{H_{min}^2 \sum_{n=1}^{\infty} \left(\frac{1}{2n+1} \right)^2} = \frac{H_{min}}{H_0} \sqrt{\sum_{n=1}^{\infty} \left(\frac{1}{2n+1} \right)^2}$$

Resolviendo la serie telescópica, se obtiene:

$$THD_{min} = \frac{H_{min}}{H_0} \sqrt{\frac{\pi^2}{8} - 1} \quad (56)$$

Para que la señal sinusoidal tenga un valor de THD mínimo, la atenuación del filtro para la banda de atenuación es:

$$\boxed{A_{min} = \frac{H_0}{H_{min}} = \frac{1}{THD_{min}} \sqrt{\frac{\pi^2}{8} - 1}} \quad \boxed{A_{min}|_{dB} = 20 \log \left(\frac{1}{THD_{min}} \sqrt{\frac{\pi^2}{8} - 1} \right)} \quad (57)$$

Para la banda de paso del filtro, la atenuación tendrá un valor que garantice que es igual al error de ganancia del filtro paso-bajo, ϵ_{GF} :

$$\epsilon_{GF} = \frac{|H(jf_c)| - H_0}{H_0} = \frac{A_{max} - H_0}{H_0} \rightarrow \boxed{A_{max} = \frac{1}{1 + \epsilon_{GF}}} \quad \boxed{A_{min}|_{dB} = 20 \log \left(\frac{1}{1 + \epsilon_{GF}} \right)} \quad (58)$$

La frecuencia de corte debe estar entre la frecuencia del armónico fundamental, $k=1$ y la frecuencia de la banda de atenuación, que es igual a la frecuencia del tercer armónico ($k=3$).

$$\boxed{f_S = 3f_{o.cuadrada}} \quad \boxed{f_{EXC} = f_{o.cuadrada} = f_P < f_C < f_S} \quad (59)$$

Entonces, los parámetros de la plantilla del filtro para un THD de 0,5% y un $\varepsilon_{GF} = -0,5\%$, utilizando las ecuaciones (57), (58) y (59) son:

$$THD = 0,005 \quad y \quad \varepsilon_{GF} = -0,005 \rightarrow$$

$$A_{min} = \frac{1}{0,005} \sqrt{\frac{\pi^2}{8} - 1} = 200 \sqrt{\frac{\pi^2}{8} - 1} \rightarrow A_{min}|_{dB} = 20 \log \left(200 \sqrt{\frac{\pi^2}{8} - 1} \right) = 39,7 \text{ dB}$$

$$A_{max} = \frac{1}{1 - 0,005} = \frac{200}{199} \quad (60)$$

$$\begin{cases} f_P = 10 \text{ kHz} \\ f_S = 3 \cdot 10 \text{ kHz} = 30 \text{ kHz} \end{cases} \rightarrow 10 \text{ kHz} \leq f_C < 30 \text{ kHz}$$

Para implementar el filtro paso-bajo se usa un filtro de Butterworth, ya que no presenta rizado en la banda de paso y simplifica el diseño. En [42] se presenta una aproximación a la ganancia de la función de transferencia del filtro paso-bajo de Butterworth:

$$|H(jf)| = \frac{H_0}{\sqrt{1 + \varepsilon^2 \left(\frac{f}{f_P} \right)^{2n}}} \quad (61) \quad \text{donde: } H_0: \text{ ganancia de la banda de paso, } f = 0 \text{ Hz.}$$

f_P : frecuencia de corte de la banda de paso del filtro.
 ε : constante relacionada con la A_{max} .
 n : orden del filtro.

Utilizando la expresión anterior, la atenuación introducida por el filtro, para una frecuencia f es:

$$A(jf) = \frac{H_0}{|H(jf)|} = \sqrt{1 + \varepsilon^2 \left(\frac{f}{f_P} \right)^{2n}} \quad A(jf)|_{dB} = 20 \log \left(\sqrt{1 + \varepsilon^2 \left(\frac{f}{f_P} \right)^{2n}} \right) \quad (62)$$

Si se considera que la banda de paso abarca desde $f = 0$ Hz hasta la frecuencia de corte de la banda de paso del filtro, f_P , en este punto la atenuación del filtro debe ser A_{max} , entonces, el valor de la constante ε^2 es:

$$A_{max} = A(jf_P) = \sqrt{1 + \varepsilon^2 \left(\frac{f_P}{f_P} \right)^{2n}} = \sqrt{1 + \varepsilon^2} \rightarrow \varepsilon^2 = A_{max}^2 - 1 \quad (63)$$

La frecuencia de corte de la banda de paso del filtro, f_P , no es la frecuencia de corte del filtro a -3dB, f_C . Para la frecuencia de corte del filtro, f_C , a -3dB, la atenuación del filtro debe ser igual a $\sqrt{2}$, imponiendo esta condición en la expresión (62), se obtiene una relación entre f_P y f_C :

$$f_C = \frac{f_P}{\sqrt[n]{\varepsilon}} = \frac{f_P}{\sqrt[2n]{\varepsilon^2}} \quad (64)$$

Para los parámetros de la plantilla del filtro, dados en las expresiones (60), se halla ε^2 ecuación (63):

$$\varepsilon^2 = A_{max}^2 - 1 = \left(\frac{200}{199}\right)^2 - 1 = \frac{399}{36901} \rightarrow$$

Para garantizar que el $THD_{min} = 0,5\%$, se debe cumplir que en la banda de atenuación del filtro:

$$A_{min} < A(jf_s) \rightarrow 39,7 \text{ dB} < A(jf_s) = 20 \log \left(\sqrt{1 + \frac{399}{36901} \left(\frac{30 \text{ kHz}}{10 \text{ kHz}}\right)^{2n}} \right)$$

En este caso, para $n=8$, esto es para filtro de octavo orden:

$$39,7 \text{ dB} < A(jf_s) = 20 \log \left(\sqrt{1 + \frac{399}{36901} \left(\frac{30 \text{ kHz}}{10 \text{ kHz}}\right)^{16}} \right) = 44 \text{ dB}$$

Entonces, la frecuencia de corte del filtro, utilizando la relación (64):

$$f_c = \frac{f_p}{\sqrt[2n]{\varepsilon^2}} = \frac{10 \text{ kHz}}{\sqrt[16]{\frac{399}{36901}}} = 13,27 \text{ kHz}$$

Utilizando esta frecuencia de corte y este orden del filtro, se diseñó e implemento un filtro, que si bien cumplía con la plantilla de diseño, debido a las tolerancias de los componentes y limitaciones de GBP de los AO, presentaba un error de ganancia máxima del -9% para el peor caso.

Tras diversas iteraciones, se llegó a una relación de compromiso entre el error máximo de ganancia, la frecuencia de corte y el THD_{min} . Así para $f_c=18\text{kHz}$, y un filtro de octavo orden, A_{max} es:

$$A_{min} = A(jf_s) = 20 \log \left(\sqrt{1 + \left(\frac{30 \text{ kHz}}{18 \text{ kHz}}\right)^{16}} \right) = 35,5 \text{ dB}$$

Esta atenuación para la banda de atenuación es inferior a la requerida para garantizar un THD_{min} de 0,5%, pero no se consiguió un mejor ajuste. Por otro lado, la atenuación en la banda de paso, A_{min} sera superior a la requerida. Así el nuevo THD_{min} , utilizando la ecuación (57) es:

$$THD_{min} = \frac{1}{A_{min}} \sqrt{\frac{\pi^2}{8} - 1} = \frac{1}{10^{\frac{35,5 \text{ dB}}{20}}} \sqrt{\frac{\pi^2}{8} - 1} = 8,11 \cdot 10^{-3} \rightarrow THD_{min}(\%) = 0,81 \%$$

Para implementar el filtro, se recurrió a las tablas de diseño normalizadas para un filtro paso-bajo de Butterworth dividido en secciones de segundo orden, en concreto, las recogidas en [43]. Así, hacen falta cuatro secciones de segundo orden, cuyas frecuencias de corte, f_0 , y factor de calidad, Q , son:

$$\begin{array}{ll} \text{Sección 1} & \left\{ \begin{array}{l} f_{01} = 18 \text{ kHz} \\ Q_1 = 0.510 \end{array} \right. & \text{Sección 2} & \left\{ \begin{array}{l} f_{02} = 18 \text{ kHz} \\ Q_2 = 0.601 \end{array} \right. \\ \text{Sección 3} & \left\{ \begin{array}{l} f_{03} = 18 \text{ kHz} \\ Q_3 = 0.900 \end{array} \right. & \text{Sección 4} & \left\{ \begin{array}{l} f_{04} = 18 \text{ kHz} \\ Q_4 = 2.563 \end{array} \right. \end{array} \quad (65)$$

Para implementar cada sección, se utiliza filtros MFF: *multiple-feedback filter* paso-bajo. Se elige esta topología por dos motivos: solo requiere un AO y la sensibilidad de la frecuencia de corte, y el factor Q, para las variaciones en los valores de los componentes, es más baja que otras topologías como la KRC.

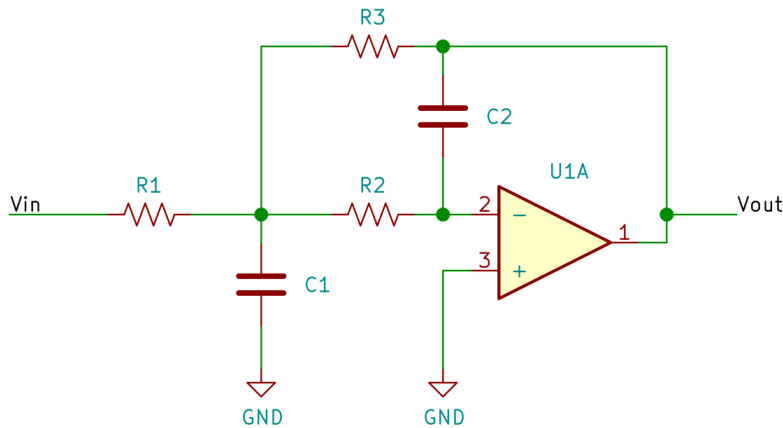


Figura 40: MFF paso-bajo, alimentación bipolar

Las relaciones los diferentes parámetros del filtro se presentan en [44]. Si se asigna el valor de C_1 y C_2 de forma que: $C_1 = kC_2$, las ecuaciones de diseño para los valores de componentes son:

$$R_3 = \frac{1 + \sqrt{1 - 4Q^2(1 + H_0)}/k}{2\omega_0 Q C_2} \quad R_1 = \frac{R_3}{H_0} \quad R_2 = \frac{1}{\omega_0^2 R_3 C_1 C_2}$$

La primera sección, tendrá una ganancia igual a la ganancia del filtro paso-bajo total, calculada en (57), por lo que: $H_{01} = H_0$. El resto de secciones tendrán ganancia unitaria. Utilizado las ecuaciones de diseño, y teniendo en cuenta los valores de frecuencia de corte y factor Q (65), se hallan los valores de los componentes para cada sección. Los valores presentados han sido normalizados para los componentes con la menor tolerancia posible. En el caso de los condensadores, el dieléctrico elegido para todos, excepto el C_1 de la cuarta sección, es NPO (COG), el cual presenta un coeficiente de variación térmica muy bajo.

- Sección 1:

$$R_1 = 8,87 \text{ k}\Omega \pm 0,1\% \quad R_2 = 489 \Omega \pm 0,1\% \quad R_3 = 16 \text{ k}\Omega \pm 0,1\% \\ C_1 = 10 \text{ nF} \pm 2\% \quad C_2 = 1 \text{ nF} \pm 2\%$$

- Sección 2:

$$R_1 = R_3 = 13,7 \text{ k}\Omega \pm 0,1\% \quad R_2 = 576 \Omega \pm 0,1\% \quad C_1 = 10 \text{ nF} \pm 2\% \quad C_2 = 1 \text{ nF} \pm 2\%$$

- Sección 3:

$$R_1 = R_3 = 7,87 \text{ k}\Omega \pm 0,1\% \quad R_2 = 1 \text{ k}\Omega \pm 0,1\% \quad C_1 = 10 \text{ nF} \pm 2\% \quad C_2 = 1 \text{ nF} \pm 2\%$$

- Sección 4:

$$R_1 = R_3 = 2,15 \text{ k}\Omega \pm 0,1\% \quad R_2 = 649 \Omega \pm 0,1\% \quad C_1 = 56 \text{ nF} \pm 5\% \quad C_2 = 1 \text{ nF} \pm 2\%$$

La topología del filtro presentada en Figura 40 es para un AO con alimentación bipolar. Como se expuso con anterioridad, esta placa solo dispone de alimentación unipolar, esto es, de 0V a 5V. Para que el filtro pueda funcionar con este tipo de alimentación, se aprovechará que el valor medio de la señal cuadrada, $v_c(t)$, no es nulo. Utilizando los términos de la ecuación (48), el valor medio de la señal, es decir, la componente DC a la entrada del filtro es M_{EXC}/H_0 .

Para alterar la posición del nivel de referencia de la señal de salida de cada sección del filtro, se puede introducir una tensión a través de la entrada V_p del AO, la cual en la Figura 40 está conectada a GND. Para calcular el efecto que esta tensión tendría en la salida, se aplica el teorema de superposición. Para la componente de continua, los condensadores C_1 y C_2 son circuitos abiertos. Para la tensión introducida por V_p , el amplificador tiene la topología de un amplificador no inversor, y para tensión introducida por la entrada del filtro, el amplificador tiene la topología de un amplificador inversor. Así, el valor medio de tensión, esto es, la componente DC, a la salida para la sección n-ésima es:

$$V_{OUT(n)}|_{DC} = V_{IN(n-1)}|_{DC} \cdot (-H_{0(n)}) + V_{P(n)} \left(\frac{R_{3(n)}}{R_{1(n)}} + 1 \right) \rightarrow V_{P(n)} = \frac{V_{OUT(n)}|_{DC} + V_{IN(n-1)}|_{DC} H_{0(n)}}{\frac{R_{3(n)}}{R_{1(n)}} + 1}$$

La primera sección tiene una ganancia, $H_{0(1)}$ igual a la ganancia total del filtro, H_0 , (57):

$$V_{P(1)} = \frac{V_{OUT(1)}|_{DC} + V_{IN(0)}|_{DC} H_{0(1)}}{\frac{R_{3(1)}}{R_{1(1)}} + 1} = \frac{M_{EXC} + \frac{M_{EXC}}{H_0} H_{0(1)}}{\frac{R_{3(1)}}{R_{1(1)}} + 1} = \frac{2 M_{EXC}}{\frac{R_{3(1)}}{R_{1(1)}} + 1} = \frac{2 \cdot 2,1V}{\frac{16k\Omega}{8,87k\Omega} + 1} = \frac{37254}{24870} V = 1,498 V$$

El resto de secciones tienen ganancia unitaria, $H_0 = 1$, luego $R_3 = R_1$, y por tanto:

$$V_{P(2)} = V_{P(3)} = V_{P(4)} = \frac{V_{OUT}|_{DC} + V_{IN}|_{DC}}{\frac{R_3}{R_1} + 1} = \frac{M_{EXC} + M_{EXC}}{\frac{R_3}{R_1} + 1} = \frac{2 M_{EXC}}{2} = M_{EXC} = 2,1 V$$

La placa dispone de un bloque, formado por un AO (U1D) y cuatro resistencias (R_5 , R_6 , R_7 y R_8) que genera dichas tensiones. Sus valores se pueden consultar en los esquemas situados en el Anexo A. Dado que su diseño es trivial y por no alargar más esta exposición, los detalles no se incluirán aquí.

A continuación, se presentará una simulación en el dominio del tiempo, en la que se muestra la señal cuadrada generada, $v_c(t)$, a la entrada del filtro paso-bajo, y la señal de excitación $v_{EXC}(t)$, a la salida del generador sinusoidal. Los detalles de esta simulación pueden consultarse en la ruta *Hardware/SWR-UPM/Diseño_sensor_R&B/Simulaciones/Generador_sinusoidal/transitorio_simulacion_completa*, en los archivos que acompañan a este proyecto.

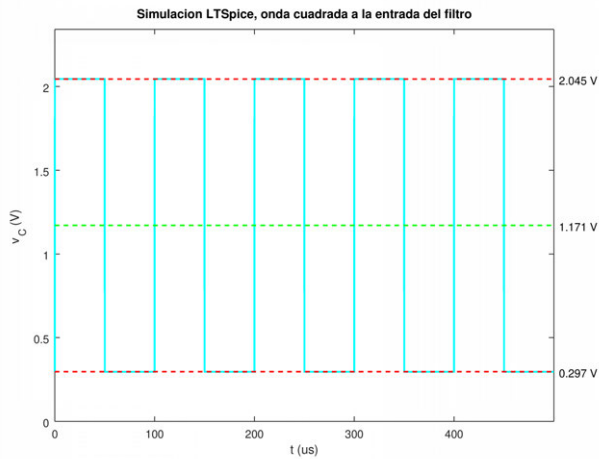


Figura 41: $v_c(t)$, onda cuadrada generada

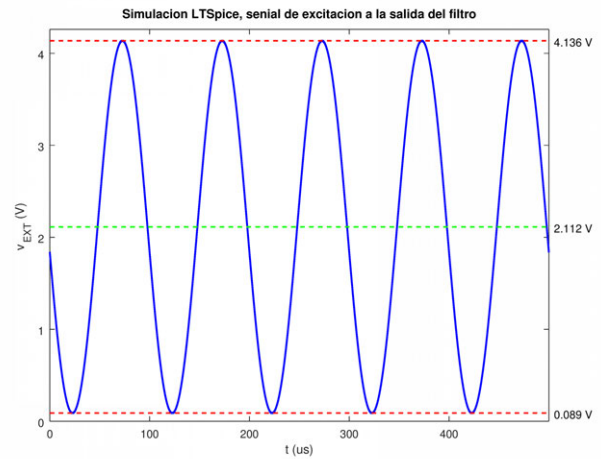


Figura 42: $v_{exc}(t)$, señal de excitación

Respecto del error de amplitud de la señal de excitación, este está relacionado con el error de amplitud de la onda cuadrada, $v_c(t)$, así como con el error de ganancia del filtro paso-bajo. Para obtener una estimación de este error, se realizarán dos tipos de análisis: un análisis para el peor caso (*worst case analysis*) y un análisis por el método de Monte Carlo.

Además, para obtener una estimación de la amplitud de la onda cuadrada, debe realizarse una simulación en el dominio del tiempo, mientras que para obtener una estimación de la ganancia del filtro, debe realizarse una simulación de respuesta en frecuencia. Así pues, para calcular el error total de amplitud de la señal sinusoidal generada, se presentarán los resultados de error para ambas simulaciones, y después se combinarán los resultados

Los análisis y detalles de las simulaciones para el generador de onda cuadrada y para el filtro pueden consultarse en el fichero *Hardware/SWR-UPM/Diseño_sensor_R&B/Simulaciones/RyB_emisor_sensor/Generador_sinusoidal*.

Para el generador de onda cuadrada, el error relativo de amplitud, ϵ_{AC} , afectará a la amplitud de la señal sinusoidal. Para el peor caso, el error está en el intervalo:

$$\epsilon_{A_c|_{WC}} \in (-0,7\%, 0,3\%) \quad (66)$$

En la figura Figura 43 se muestra el resultado del análisis de Monte Carlo para la amplitud de la onda cuadrada. Para el 99.3% de las muestras, el error está contenido en el intervalo:

$$\epsilon_{A_c|_{MC,99.3\%}} \in (-0,6\%, 0,2\%) \quad (67)$$

Respecto al error de ganancia del filtro paso-bajo, realizar un análisis completo de peor caso, para todas la combinaciones posibles de los valores de los componentes es inviable, ya que en total hay 20 componentes, lo que implica que deberían ejecutarse $2^{20} = 1048576$ simulaciones para tener en cuenta todas las posibles combinaciones. Además, dividir el filtro en secciones o en partes más pequeñas tampoco sería útil, porque la función de transferencia del circuito resultante no sería representativa de las características del filtro. Debido a esto, para el filtro paso-bajo solo se presentará una estimación del error basada en el método de Monte Carlo para 100000 muestras.

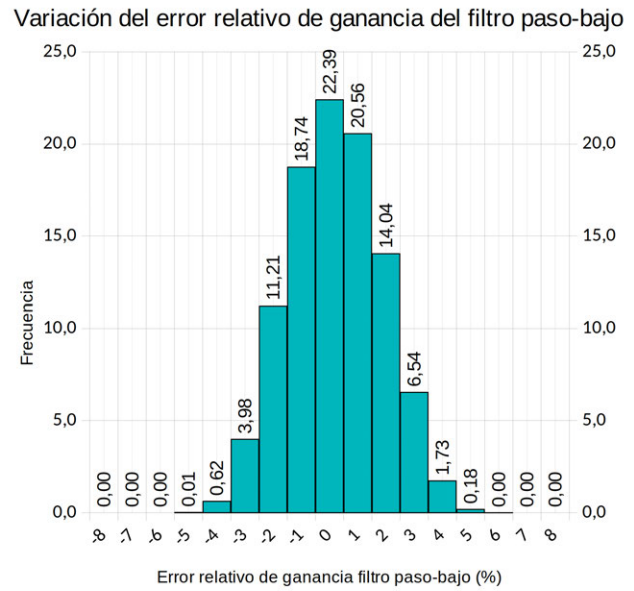
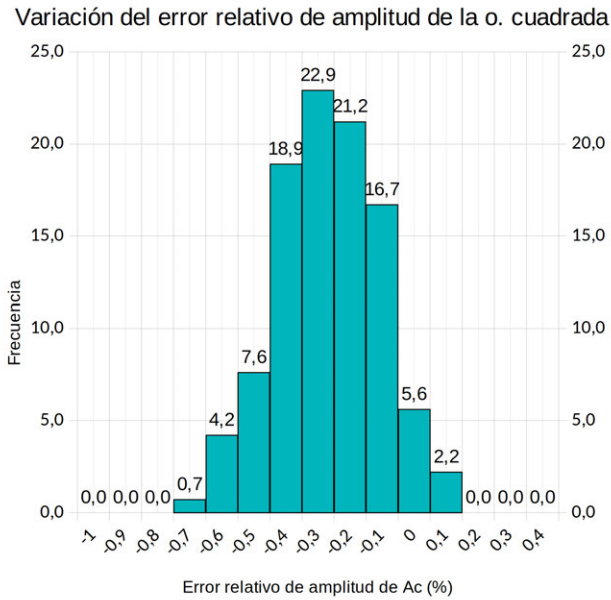


Figura 43: Análisis de Monte Carlo para el generador de onda cuadrada, N=1000

Figura 44: Análisis de Monte Carlo para el filtro paso-bajo, N=100000

Utilizando el resultado del análisis de Monte Carlo para el error de ganancia del filtro paso-bajo mostrado en la Figura 44, para el 97.46% de las muestras el error está contenido en el intervalo:

$$\mathcal{E}_H|_{MC,97.46\%} \in (-3\%, 4\%) \tag{68}$$

El receptor calcula la distancia al emisor utilizando la misma tabla para todos los robots, por lo que si el intervalo de error de la intensidad radiante es amplio, dos robots pueden estar a la misma distancia del receptor y, debido al error de la intensidad radiante emitida, aparecer desplazados. Este fenómeno afectaría al uso en la formación de patrones y movimiento coordinado, por tanto, es importante caracterizarlo.

Con los datos (46), (47), (66), (67) y (68), se puede realizar una estimación del intervalo de error de la intensidad radiante emitida por el módulo. Esta estimación se basa en la suposición los leds infrarrojos no tienen ninguna tolerancia, y que por tanto, la intensidad radiante emitida para una determinada corriente es siempre exactamente la misma. Dado que ningún componente es perfecto, esta suposición es incorrecta, pero dado que no hay forma de estimar dicha tolerancia, se supondrá que son perfectos. Así, la intensidad radiante emitida puede expresarse como:

$$I_{e\Omega}(1+\varepsilon_e) = A_C(1+\varepsilon_{A_C})H_0(1+\varepsilon_{GF})\frac{2}{\pi}G_{ATC}(1+\varepsilon_{ATC})G_D \tag{69}$$

Igualando los términos de error y despejando el error de la intensidad radiante:

$$\varepsilon_e = (1+\varepsilon_{A_C})(1+\varepsilon_{GF})(1+\varepsilon_{ATC}) - 1 \tag{70}$$

Se distinguirán dos tipos de error: máximo, para el resultado de calcular el error utilizando las estimaciones para el peor caso posible, y típico, para el resultado obtenido a partir de las estimaciones basadas en el método de Monte Carlo. Así, utilizando la expresión (70):

$$\boxed{\varepsilon_e|_{max} \in (-6,9\%, 7,1\%)} \quad \boxed{\varepsilon_e|_{typ} \in (-5,3\%, 6,6\%)} \quad (71)$$

Las principales contribuciones a este error se deben al error en la ganancia del amplificador de transconductancia, y al error en la ganancia del filtro paso-bajo.

4.1.4.5 Fuente de alimentación

Como se ha mencionado con anterioridad, aparte de una conexión el bus de alimentación de la batería (+V_{BAT}), esta placa solo dispone de una tensión regulada de +5V. Para cumplir con la especificación EH-11, y para asegurar que las alimentaciones sean las más adecuadas, tanto la placa emisora como la receptora del *sensor de rango y rumbo* llevan sus propios reguladores de tensión. El consumo pico de cada uno de los bloques del emisor para el raíl de +5V se detalla a continuación:

- Amplificadores de transconductancia: 11mA.
- Generador sinusoidal: 12,8mA
- Matriz de activación, referencia de tensión y led indicador de estado: 4mA

Corriente total pico raíl +5V: 27,8mA

El regulador de +5V está conectado al raíl VBAT, así como los colectores de los transistores de salida de los amplificadores de transconductancia. Así, pues, el consumo pico del raíl de Vbat será:

Corriente total pico raíl +V_{BAT}: 850mA (72)

La corriente eficaz que debe alimentar la placa a través de VBAT es:

Corriente RMS total pico raíl +V_{BAT}: 507mAef

Con estos parámetros, se selecciona el regulador NCP718ASN500T1G, del fabricante ONSEMI, el dispone de entrada de habilitación, una corriente de salida máxima de 800mA y unas excelentes características de ruido.

Además, con el objetivo de reducir las posibilidades de acoplamiento o ruido debido a la gran cantidad de corriente que circula a través del raíl de +V_{BAT}, se ha incluido una línea de retorno de masa solo para los emisores infrarrojos. Esta línea de retorno de masa (POWER_GND en el esquema), se une a la masa de la batería en la placa de potencia. El resto de componentes de la placa comparte otra línea de masa con la placa madre (GND en los esquemas).

4.1.4.6 Diseño de la PCB y soporte de los emisores infrarrojos

El diseño de la PCB se muestra a continuación:

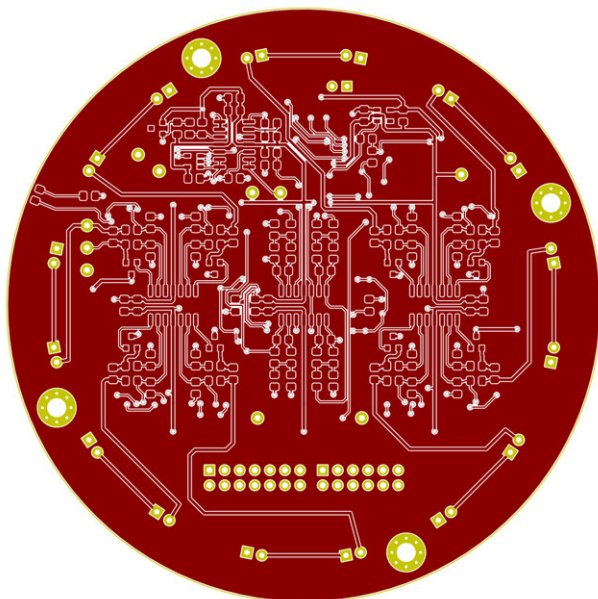


Imagen 28: Capa top de la R&B_sensor_tx

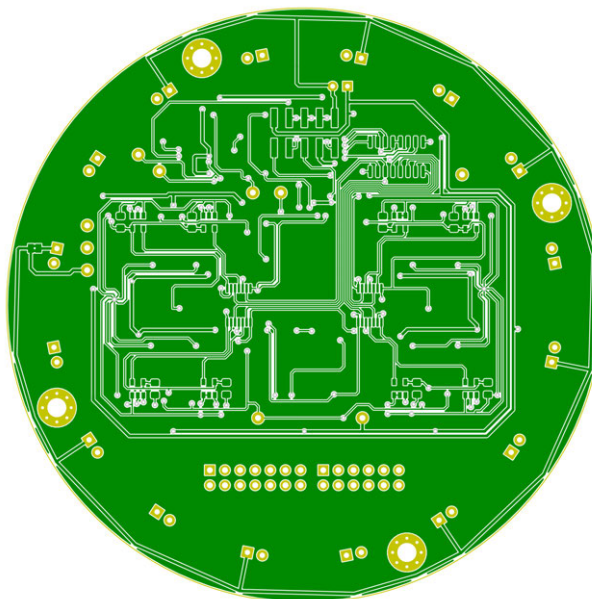


Imagen 29: Capa bottom de la R&B_sensor_tx

Para reducir el *crosstalk* entre los leds emisores y los fotodiodos de la placa receptora, cuando el sensor esté funcionando como sensor de reflexión, se han diseñado dos piezas para imprimir en 3D. La pieza que se sitúa en la placa emisora dispone de dieciséis soportes para los leds IR. Cada soporte tiene a cada lado unas aletas que evitan que la señal enviada por los emisores infrarrojos pueda llegar directamente a los fotodiodos. Esto reduce al mínimo el *crosstalk*, y maximiza el alcance.

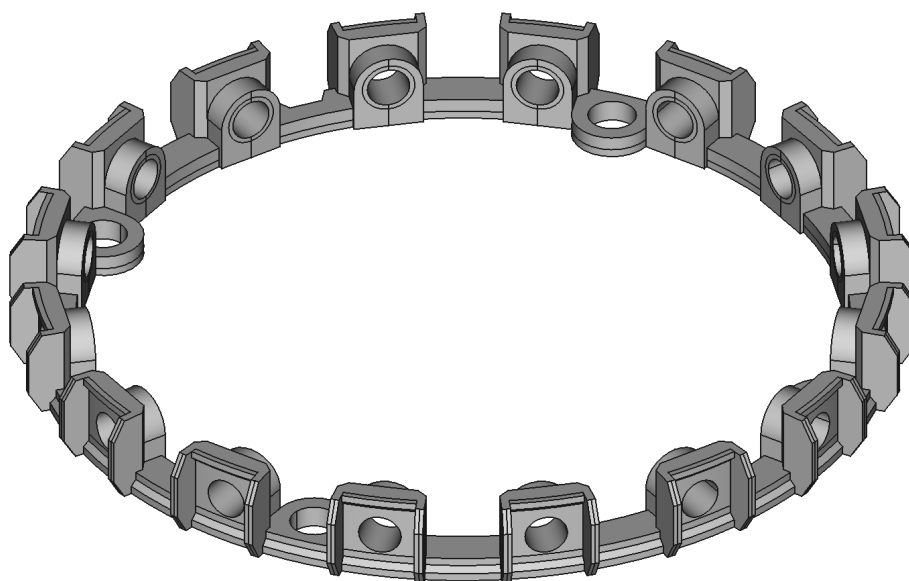


Imagen 30: Soporte de los leds IR

En los archivos que acompañan a este proyecto, esta pieza está archivada bajo el nombre: *soporte_emisores_IR_rediseño_v1_chaflan_30.FCStd*.

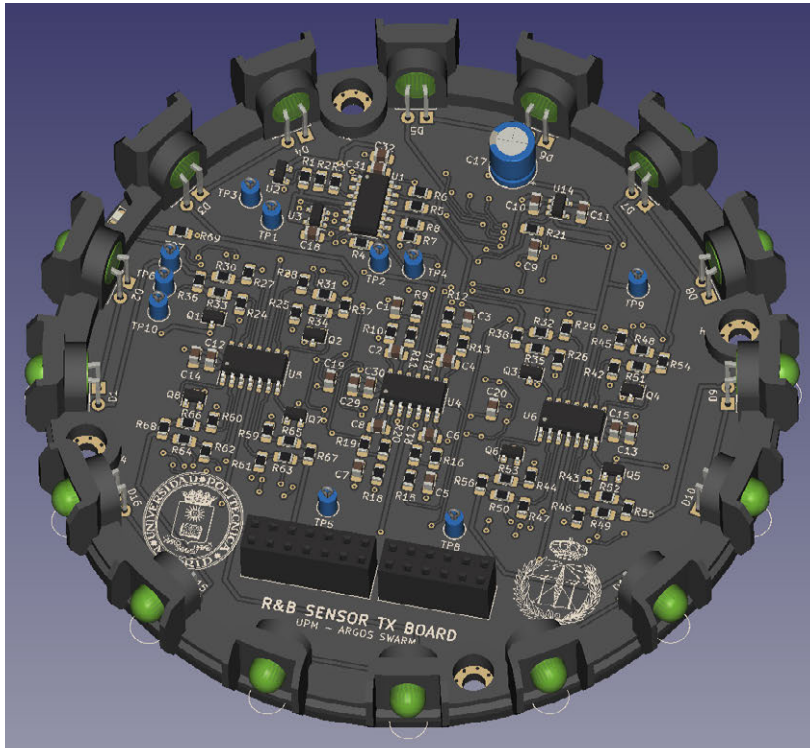


Imagen 31: Simulación 3D de la placa terminada, vista superior

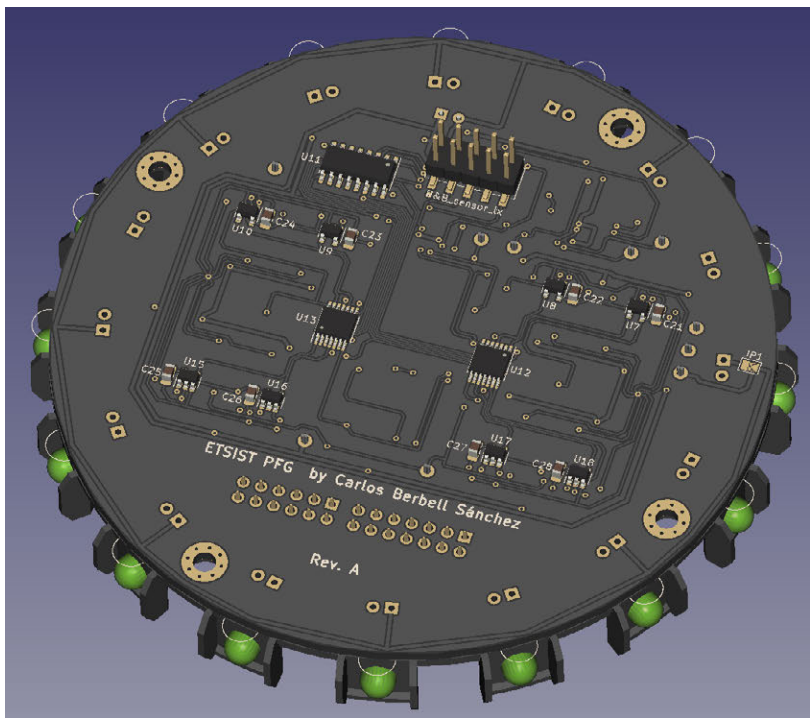


Imagen 32: Simulación 3D de la placa terminada, vista inferior

4.1.5 Nivel 4 - Placa del receptor del sensor de rango y rumbo (R&B_sensor_rx board)

Las especificaciones que satisface son: *EH-3*, *EH-11*.

El siguiente diagrama de bloques es la versión completa del diagrama de bloques preliminar presentado en la Figura 25.

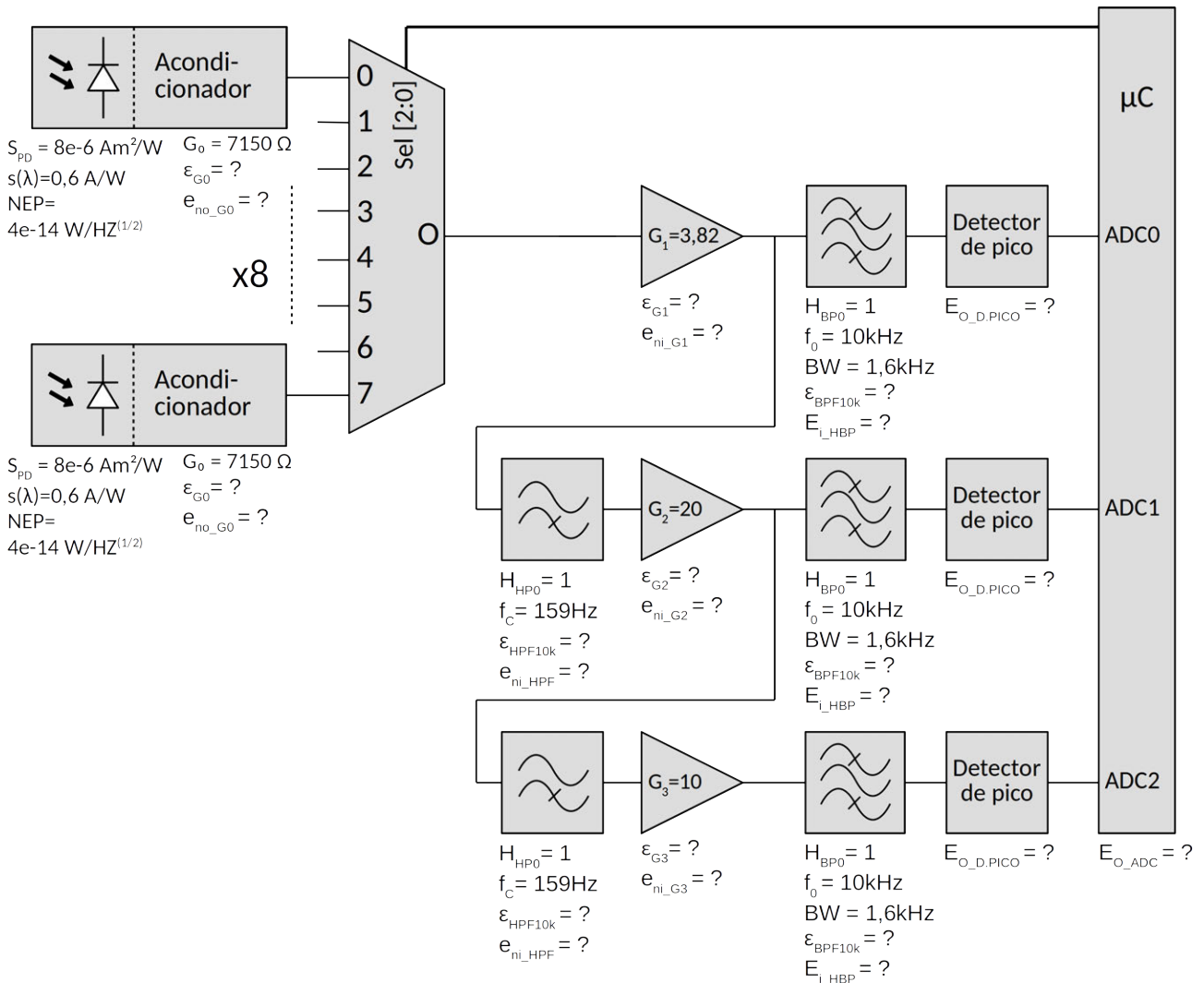


Figura 45: Diagrama de bloques del receptor del sensor de rango y rumbo

Este diagrama recoge los elementos que intervienen en el procesamiento de la señal recibida por los fotodiodos. Los fotodiodos, presentados como bloques separados en el diagrama preliminar, Figura 25, se han incluido dentro del bloque del acondicionador, ya que a efectos de cálculo de ruido y ganancia forman una unidad.

En este diagrama están representadas todos los parámetros importantes de cada bloque. Los más obvios son las ganancias, indicadas con la letra *G*, y los valores de módulo de respuesta en frecuencia de los filtros a la frecuencia crítica, indicados con la letra *H*. A parte de estos, mediante la letra *epsilon*, ϵ , se han indicado los errores relativos que presentan estos valores respecto de sus valores nominales indicados en el diagrama. Dado que los valores de error relativo, ϵ , dependen de

la selección de componentes, estos son desconocidos a priori, por lo que se usa el signo de interrogación para remarcar este hecho.

Como se mencionó en 4.1.3.4, el ruido es el factor principal que limita el alcance del sensor, por lo que se debe calcular su efecto en la salida. Para ello, en el diagrama se han incluido varios parámetros

que permiten caracterizar el ruido de los bloques. Los términos que empiezan por e_{nx} indican la densidad espectral de ruido (W/\sqrt{Hz}), dependiendo del índice x este puede referirse a la densidad espectral de ruido a la salida (e_{no}) o la densidad espectral de ruido a la entrada (e_{ni}). En la mayoría de bloques la densidad espectral de ruido está proyectada en la entrada. Los términos que empiezan por E_x indican la contribución de ruido del módulo en voltaje eficaz, esto es útil para aquellos módulos cuya contribución al ruido sea independiente del ancho de banda del sistema.

Por último, en el acondicionador se han incluido los parámetros de sensibilidad del fotodiodo para la intensidad radiante recibida: S_{PD} , y dos parámetros que influyen en el ruido generado por el fotodiodo: la sensibilidad espectral absoluta, $S(\lambda)$, y la potencia equivalente de ruido, NEP .

Respecto al funcionamiento, los detalles principales han sido presentados en 4.1.3.3. La única modificación significativa respecto al diagrama de bloques preliminar son los filtros paso-alto entre etapas. El nivel de señal a la salida del acondicionador es de una magnitud comparable a la tensión de offset generada por los amplificadores. Para evitar la pérdida de rango dinámico debido a la amplificación de la tensión de offset, los amplificadores están acoplados en AC, a través de un filtro paso-alto, lo que elimina el efecto de la tensión de offset de la etapa anterior de amplificación.

A continuación, se expondrán los detalles del diseño, comenzando por la entrada de la señal, esto es el fotodiodo. Tras esto, se presentaran el resto de módulos, y dado que su diseño es el mismo para cada etapa, los detalles de la implementación se resumirán. Por último, y antes de presentar el diseño de la PCB y piezas en 3D, se realizará un breve análisis que relacione los errores de ganancia del emisor y receptor con los errores de posición del sensor.

4.1.5.1 Selección de fotodiodos, caracterización y disposición

Del mismo modo que los diodos infrarrojos tienen un parámetro que caracteriza el ángulo patrón de radiación, los fotodiodos tienen un parámetro que caracteriza la sensibilidad relativa respecto del ángulo: *angle of half sensitivity*. Este parámetro mide el ángulo a partir del cual la sensibilidad del fotodiodo es la mitad de la máxima sensibilidad, alcanzada cuando la fuente de emisión está totalmente perpendicular a la superficie del fotodiodo.

Este sensor dispone de ocho fotodiodos, situados en el centro de cada uno de los sectores controlables individualmente, y dispuestos de forma equidistante. Los ocho fotodiodos deben cubrir el área completa del sensor, por lo que el *angle of half sensitivity* mínimo debe ser de $360^\circ/8 = 45^\circ$. Cualquier fotodiodo que tenga un *Angle of half sensitivity* de al menos 45° podrá considerarse en el diseño.

Además, el fotodiodo elegido debe alcanzar la sensibilidad máxima para la misma longitud de onda de los emisores IR, en concreto, 940nm. Otro parámetro que influye en la elección es la

responsividad, un parámetro que relaciona la fotocorriente generada por el fotodiodo con la potencia luminosa recibida. Dado que no todos los *datasheet* lo recogen, lo más adecuado es utilizar otro parámetro relacionado con este: la corriente de luz inversa (*reverse light current*), que es la fotocorriente generada por el fotodiodo polarizado en inversa para una irradiancia determinada. Cuanto mayor sea este valor, mayor será la responsividad del fotodiodo.

Teniendo en cuenta estos parámetros, se seleccionó el fotodiodo BPV22NF del fabricante *Vishay Semiconductors*. Según la hoja de datos del BPV22NF [45], este tiene un *angle of half sensitivity* de 60°, la sensibilidad máxima se alcanza cuando la longitud de onda es de 940nm, y la corriente de luz inversa típica es de 85µA, la más grande entre todos los candidatos analizados.

Para calcular la sensibilidad del fotodiodo, S_{PD} , se utilizará la tabla gráfica *Reverse Light Current vs. Irradiance* de la hoja de datos del BPV22NF [45]:



Figura 46: Curva I_{ra}/E_e [45]

$$y - y_0 = m(x - x_0) \rightarrow S_{PD} = m = \frac{(y - y_0)}{(x - x_0)}$$

$$\begin{cases} x = 1 \\ y = 80 \\ x_0 = 0,01 \\ y_0 = 0,8 \end{cases} \rightarrow S_{PD} = m = \frac{(80 - 0,8)}{(1 - 0,01)} = 80 \frac{\mu A}{mW/cm^2}$$

$$S_{PD} = 8 \cdot 10^{-6} \frac{Am^2}{W} \quad (73)$$

4.1.5.2 Acondicionador de los fotodiodos y multiplexor

El acondicionador convierte la fotocorriente generada por el fotodiodo a una señal de tensión. Además, dado que el robot debe operar en zonas iluminadas con luz artificial y/o natural, el acondicionador debe eliminar o reducir el efecto de esta iluminación sobre la señal de salida. La iluminación natural se puede modelar como una fuente de intensidad radiante continua, y la iluminación artificial como una fuente con una componente continua, y una componente alterna situada aproximadamente a una frecuencia de 50Hz. Adicionalmente, como se explicó en 4.1.4.2, el emisor del sensor genera una señal con una componente continua.

Por tanto, para suprimir su influencia, el acondicionador debe estar acoplado en AC, eliminando el efecto que las fuentes de intensidad radiante continua generan sobre la señal de salida. La componente alterna generada por la iluminación artificial será eliminada por el filtro paso-banda. A parte de este requisito, el acondicionador debe tener una densidad espectral de ruido lo más reducida que sea posible, ya que el ruido del mismo será amplificado por la cascada de amplificadores junto con la señal recibida, influyendo de forma significativa en la distancia de operación del sensor.

Hay varias formas de implementar este tipo de acondicionador. La más estándar es utilizar un amplificador de transresistencia, es decir, un amplificador que convierte señal de corriente

generada por el fotodiodo en una señal de tensión. Esta configuración es la más usada en aplicaciones como los controles remotos basados en infrarrojos, por dos motivos: tiene un gran ancho de banda, requisito para utilizar una señal cuadrada modulada en frecuencia, y aunque el nivel de ruido de la señal recibida sea alto, la reconstrucción de una señal cuadrada tiene una gran tolerancia al ruido. En este caso, los requisitos de ancho de banda son mínimos, puesto que la señal recibida es una onda continua a una frecuencia fija, y la información va contenida en la amplitud, por lo que la saturación del acondicionador o el ruido son factores mucho más relevantes.

En una fase preliminar al diseño de este módulo se analizó la posibilidad de usar un amplificador de transresistencia, y si bien era posible y mejoraba algunas características como la máxima frecuencia de operación, empeoraba otras características como la máxima intensidad radiante continua que se podía tolerar. Esto, sumado a la mayor complejidad y a la necesidad de usar una AO de bajo ruido, lo que encarece el precio, hizo que se recurriera a una solución más simple.

La solución elegida se presenta a continuación:

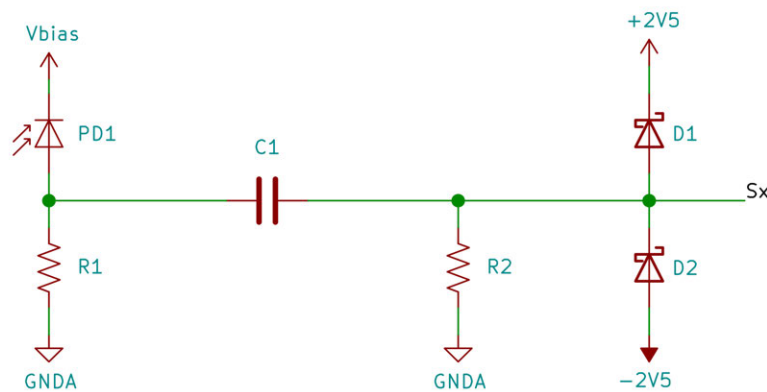


Figura 47: Acondicionador para los fotodiodos

El fotodiodo está polarizado en inversa mediante la tensión V_{bias} . Las resistencias R_1 y R_2 convierten la fotocorriente producida por el fotodiodo en una señal de tensión. El condensador C_1 , junto con las resistencias R_1 y R_2 forman un filtro paso alto, que elimina la componente continua de la fotocorriente. Para la componente continua de la señal, el condensador C_1 es un circuito abierto, por lo que la fotocorriente debido a esta pasará por la resistencia R_1 exclusivamente. Si la fotocorriente debido a la componente continua es muy grande, el fotodiodo podría dejar de estar polarizado en inversa, cuando la tensión en la resistencia R_1 supere a la tensión de V_{bias} . Para evitar esto, V_{bias} debe tener un valor elevado (del orden de decenas de voltios).

La alimentación de esta placa es bipolar, de +2,5V a -2,5V, y tanto el multiplexor como el AO que se sitúa detrás, solo pueden soportar tensiones a su entrada en ese rango. Para evitar que se produzcan daños en estos componentes debido a los transitorios que se generan al variar la fotocorriente causada por la componente continua de intensidad radiante, se utilizan un par de diodos Schottky, D_1 y D_2 , para recortar el nivel de la señal de salida del acondicionador.

Idealmente, a la frecuencia de operación, el condensador C_1 tendrá una impedancia lo suficientemente pequeña que puede considerarse un circuito cerrado. Así, despreciando el efecto

Descripción de la solución propuesta

de la impedancia en la salida, toda la fotocorriente generada por el fotodiodo circulará por las resistencias R_1 y R_2 , por lo que la ganancia ideal del acondicionador es:

$$G_{0i} = \frac{v_o}{i_D} = R_1 \parallel R_2 \quad (74)$$

Esta aproximación permite estimar el valor de las resistencias R_1 y R_2 . Dado que la ganancia del acondicionador, según los parámetros escogidos para el sensor en 4.1.3.5, debe ser de 7150Ω . Si $R_1 = R_2$, los valores para estos componentes son:

$$G_{0i} = \frac{R_1}{2} = \frac{R_2}{2} = 7150\Omega \rightarrow R_1 = R_2 = 2G_{0i} = 2 \cdot 7150\Omega \rightarrow R_1 = R_2 = 14,3k\Omega \pm 1\%$$

La ganancia ideal del acondicionador no tiene en cuenta el impacto de la impedancia de carga en la salida del acondicionador, ni la capacidad en la unión del fotodiodo. A la salida del acondicionador se sitúa el multiplexor analógico, el cual permite seleccionar cuál de los fotodiodos será utilizado en la medida. El multiplexor debe tener una resistencia para el canal activo, r_{on} , baja (de unos pocos ohmios) y una corriente de fugas a la entrada baja (del orden de nA).

Estas características se encuentran en el multiplexor analógico TMUX1108 del fabricante *Texas Instruments*. Según la hoja de datos [45], $r_{on} = 4\Omega$, $I_{D(on)} = I_{S(on)} = 0,01nA$. Además de estos parámetros, la capacidad a entrada y a la salida el multiplexor tendrá un gran impacto en la respuesta en frecuencia del acondicionador; para el canal activo: $C_{S(on)} = C_{D(on)} = 65pF$. Teniendo en cuenta el modelo del fotodiodo presentado en [45], el modelo equivalente en AC del acondicionador es:

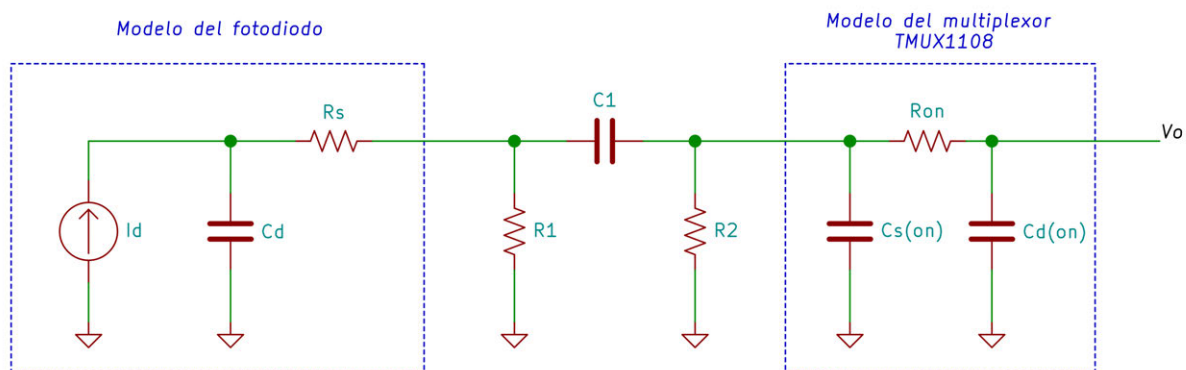


Figura 48: Modelo AC, acondicionador del fotodiodo

El valor de la corriente de fuga es tan pequeño que no ha sido incluido en el modelo. La resistencia del canal del multiplexor, r_{on} , es lo suficientemente pequeña como para considerarla un cortocircuito, por lo que los valores de las capacidades parásitas a la entrada y a la salida del multiplexor, pueden combinarse en un único componente: $C_p = 2 \cdot 65pF = 130pF$. Por otro lado, según la hoja de datos del fotodiodo [45], la capacidad de la unión del fotodiodo es: $C_d = 20pF$ (para una tensión inversa, V_R , igual a $6V$). Dado que la capacidad de C_d es tan pequeña, su impedancia a frecuencias bajas será tan alta, que en una primera aproximación, su efecto puede despreciarse ($C_d = 0$). Ahora, aplicando el Teorema de Norton, y operando, se obtiene la siguiente función de transferencia:

$$\frac{V_o(jf)}{I_D(jf)} = \frac{j2\pi f C_1(R_1+R_2)}{1+j2\pi f(C_p R_2+C_1(R_1+R_2))-4\pi^2 f^2 C_1 C_p R_1 R_2} \cdot R_1 \parallel R_2 \quad (75)$$

El valor de C_1 estará en el rango de nano-faradios. Así pues, conociendo los valores de C_p , R_1 y R_2 , se cumple que $C_p R_2 \ll C_1(R_1+R_2)$, por lo que la ecuación (75), se puede reescribir como:

$$\frac{V_o(jf)}{I_D(jf)} = \frac{j2\pi f C_1(R_1+R_2)}{1+j2\pi f(C_1(R_1+R_2))-4\pi^2 f^2 C_1 C_p R_1 R_2} \cdot R_1 \parallel R_2 \quad (76)$$

La ecuación (76) corresponde a la forma estándar de un filtro paso-banda de segundo orden. Identificando términos, se obtienen las siguientes relaciones:

$$H_0 = R_1 \parallel R_2 \quad f_0 = \frac{1}{2\pi\sqrt{C_p C_1 R_1 R_2}} \quad Q = \frac{\sqrt{C_p C_1 R_1 R_2}}{C_1(R_1+R_2)} \rightarrow BW = \frac{f_0}{Q} = \frac{C_1(R_1+R_2)}{C_p C_1 R_1 R_2} \quad (77)$$

El valor del condensador C_1 no solamente influirá en la impedancia entre R_1 y R_2 , también influirá en el tiempo que tarda el acondicionador en alcanzar un valor estable tras producirse una variación en las condiciones de iluminación. Se elige un valor de $C_1 = 10nF$ por varios motivos: el módulo de la impedancia a una frecuencia de 100kHz y 10kHz varía de 159Ω hasta 1590Ω , y se pueden encontrar condensadores multicapa cerámicos con dieléctrico NPO, que no presentan una disminución de la capacidad efectiva con el voltaje, lo que es relevante porque la tensión de V_{bias} estará en el orden de decenas de voltios.

Así pues, aplicando las relaciones (77), se obtiene, que los parámetros de la respuesta en frecuencia del acondicionador son:

$$H_0 = G_{oi} = 7150 \Omega \quad f_0 = 9761 \text{ Hz} \quad Q = 0,057 \rightarrow BW = 171,2 \text{ kHz}$$

En el fichero localizado en la ruta *Hardware/SWR-UPM/Diseño_sensor_R&B/Simulaciones/RyB_receptor_sensor/Acondicionador_y_multiplexor/AC_valores_nominales*, en los archivos que acompañan a este proyecto, se encuentra una simulación del modelo presentado en la Figura 48.

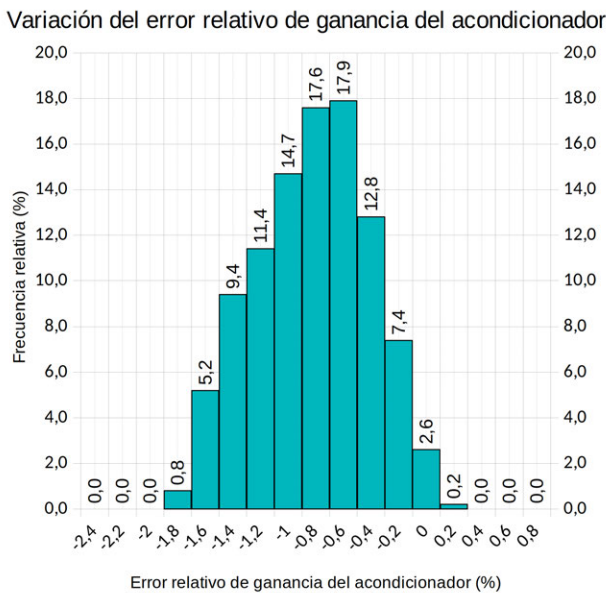
$$H_0|_{sim} = 7098,15 \Omega \quad f_0|_{sim} = 9200 \text{ kHz} \quad |H(j10k)|_{sim} = 7097,82 \Omega \quad BW|_{sim} = 153,231 \text{ kHz}$$

Como se puede observar, los valores difieren de los calculados debido a que no se ha tenido en cuenta el efecto de C_d . Para evitar que el error relativo de ganancia del acondicionador, ϵ_{G_0} , se elige una frecuencia de operación cercana a la frecuencia de crítica del filtro paso bajo, esto es, f_0 . Así pues, se fija la frecuencia de 10kHz como la frecuencia de la onda continua.

En cuanto a la tolerancia de los componentes, dado que las resistencias R_1 y R_2 están en paralelo, el efecto de la tolerancia de las mismas en el valor de G_0 es menor, por lo que se utilizan resistencias de la serie E-96 ($\pm 1\%$ de tolerancia). En cuanto al condensador C_1 , se eligen un condensador con un $\pm 5\%$ de tolerancia, ya que su efecto en la ganancia del acondicionador es menor que el de las resistencia R_1 y R_2 .

$$R_1=R_2=14,3\text{kHz} \pm 1\% \quad C_1=10\text{nF} \pm 5\% \quad \text{Dieléctrico: NP0(C0G)}$$

Conociendo el valor de los componentes, y el modelo de AC del acondicionador, se realiza un análisis del impacto de la tolerancia en el error relativo de ganancia, ϵ_{G0} . Al igual que en los apartados anteriores, se realiza un análisis para el caso peor (*worst case analysis*), y un análisis por el método de Monte Carlo. Las simulaciones correspondientes a ambos se pueden encontrar en los archivos que acompañan a este proyecto, en el directorio: *Hardware/SWR-UPM_memoria/Simulaciones/RyB_receptor_sensor/Acondicionador_y_multiplexor/AWC_tolerancia*, y *Hardware/SWR-UPM_memoria/Simulaciones/RyB_receptor_sensor/Acondicionador_y_multiplexor/AMC_tolerancia*, respectivamente. Los resultados de ambos análisis se exponen a continuación:



$$\epsilon_{G0}|_{WC} \in (-1,8\%, 0,3\%) \quad (78)$$

Utilizando los datos del análisis de Monte Carlo, presentado en la Figura 49, para el 96.4% de las muestras el error está contenido en el intervalo:

$$\epsilon_{G0}|_{MC,96.4\%} \in (-1,6\%, 0\%) \quad (79)$$

Figura 49: Análisis de Monte Carlo para el acondicionador de los fotodiodos, N=1000

En cuanto a la tensión de polarización del fotodiodo, V_{bias} , esta debe garantizar que para las condiciones más adversas de iluminación, el fotodiodo siempre estará polarizado de forma inversa. La iluminación artificial concentra la mayor parte de su potencia de emisión en el espectro visible para el ojo humano, por lo que la cantidad de energía que radia en la longitud de onda cercana a 940nm, contenida en el espectro infrarrojo, es muy pequeña. Por tanto, la mayor fuente de infrarrojos que puede afectar a la operación del sensor es el Sol, pues emite grandes cantidades de energía en todo el espectro electromagnético.

Para obtener una referencia de cuál es la máxima fotocorriente que puede circular por el fotodiodo, se colocó un diodo polarizado de forma inversa con una fuente de tensión de 6V, y en serie con este y la fuente de tensión, un amperímetro. A lo largo de 3 días, cuando la irradiación solar era máxima, se expuso el montaje a la luz solar, variando el ángulo de incidencia de la luz hasta alcanzar la máxima fotocorriente. La fotocorriente máxima fue de:

$$I_{D|_{max}}^{medida} = 1,886 \text{ mA}$$

Utilizando la sensibilidad del fotodiodo, S_{PD} , calculada en (73), se puede estimar que la máxima irradiancia recibida para la espectro en el que el fotodiodo es sensible, fue de 235 W/m^2 . Es muy improbable que los robots operen en condiciones de iluminación tan desfavorables, no obstante, para garantizar la máxima inmunidad frente a esta, se establece que la fotocorriente corriente máxima es de 2mA , correspondiente a una irradiancia de 250W/m^2 . Entonces, fijando una corriente inversa para el fotodiodo mínima de 6V (que garantiza que $C_d = 20\text{pF}$), y despreciando el efecto de la fotocorriente debida a las emisiones de otros robots por estar un par de órdenes de magnitud por debajo, se obtiene que la tensión de polarización, V_{BIAS} , mínima es:

$$\begin{cases} I_D|_{max} = 2\text{mA} \\ V_{BIAS}|_{min} = V_R|_{min} + I_D|_{max} R_1 = 5\text{V} + 2\text{mA} \cdot 14,3\text{k}\Omega = 33,6\text{V} \end{cases} \quad (80)$$

Para proveer a los fotodiodos de dicha tensión, la placa del receptor dispone de un conversor elevador, tal como se mostrará más adelante.

En cuanto al ruido térmico del acondicionador, se utilizó el modelo de ruido térmico para el fotodiodo presentado en [46] para modelar su contribución al ruido. A parte del fotodiodo, otras fuentes de ruido térmico son las resistencias R_1 y R_2 . Los diodos D_1 y D_2 contribuirán principalmente con ruido de disparo (*shot noise*) debido a la corriente de fuga.

La contribución al ruido del multiplexor se desprecia por dos motivos: primero, la resistencia del canal, r_{on} , es muy pequeña por lo que la contribución al ruido total será despreciable, y segundo, el fabricante no especifica ningún parámetro relacionado con los MOSFET que forman el multiplexor o un modelo de ruido, con el que realizar la estimación.

Por otro lado, para obtener una estimación por exceso, en este modelo de ruido supone que el efecto de la capacidad de la unión del fotodiodo, C_d , es despreciable, y que el condensador C_1 tiene una capacidad tan grande que se puede considerar como un cortocircuito. La Figura 50 muestra el modelo de ruido del acondicionador:

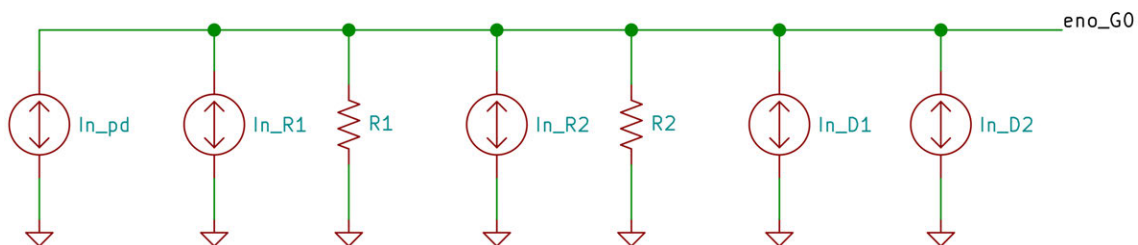


Figura 50: Modelo de ruido del acondicionador del fotodiodo

La densidad de potencia de ruido a la salida es:

$$e_{no_G0}^2 = [(R_1 || R_2) i_{n_R1}]^2 + [(R_1 || R_2) i_{n_R2}]^2 + [(R_1 || R_2) i_{n_pd}]^2 + [(R_1 || R_2) i_{n_D1}]^2 + [(R_1 || R_2) i_{n_D2}]^2$$

$$\begin{cases} i_{n_{R_1}}^2 = i_{n_{R_2}}^2 = \frac{4kT}{(R_1 \parallel R_2)} \\ i_{n_{pd}} = NEP \cdot S(\lambda) \\ i_{n_{D_1}}^2 = i_{n_{D_2}}^2 = 2q I_{LK} \end{cases} \rightarrow \boxed{e_{no_{G0}}^2 = R_1 \parallel R_2 \left[4kT + NEP^2 S(\lambda)^2 (R_1 \parallel R_2) + 4q I_{LK} (R_1 \parallel R_2) \right]} \quad (81)$$

Donde: k es la constante de Boltzman ($1,38 \cdot 10^{-23}$ J/K), T es la temperatura de los componentes en grados Kelvin (300K para 25°C), q es la carga del electrón ($1,602 \cdot 10^{-19}$ C), y I_{LK} es la corriente de fuga de los diodos D_1 y D_2 . Para reducir el espacio del acondicionador y el número de componentes, se eligió un encapsulado que incluía dos diodos Schottky, específicamente, el BAS70-04 del fabricante *OnSemi*. Utilizando la hoja de datos [47] del BAS70-04, para una tensión inversa de 35V y una temperatura de operación de 25°C, la corriente de fuga es: $I_{LK} = 5nA$. Entonces, la densidad de potencia de ruido a la salida del multiplexor es:

$$e_{no_{G0}}^2 = 7150 \Omega \left[4 \cdot 1,38 \cdot 10^{-23} \frac{J}{K} \cdot 300 K + \left(4 \cdot 10^{-14} \frac{W}{\sqrt{Hz}} \right)^2 \left(0,6 \frac{A}{W} \right)^2 \cdot 7150 \Omega + 4 \cdot 1,602 \cdot 10^{-19} C \cdot 5 nA \cdot 7150 \Omega \right] = 1,19089e-16 \frac{V}{Hz}$$

La densidad espectral de ruido a la salida del acondicionador es:

$$\boxed{e_{no_{G0}} = 10,91 \frac{nV}{\sqrt{Hz}}} \quad (82)$$

En el directorio *Hardware/SWR-UPM_memoria/Simulaciones/RyB_receptor_sensor/Acondicionador_y_multiplexor/Analisis_de_ruido*, hay una simulación del ruido generado por el acondicionador del fotodiodo. En esta simulación se aprecia que la densidad espectral de ruido depende de la frecuencia, esto es, no es constante para todas las frecuencias. Con todo, en las cercanías de 10kHz, para el ancho de banda del filtro paso-banda, la densidad espectral de ruido se puede considerar constante:

$$\boxed{e_{no_{G0}}|_{sim} = 10,87 \frac{nV}{\sqrt{Hz}}} \quad (83)$$

Como se puede observar, el valor es muy cercano al obtenido de forma teórica.

4.1.5.3 Etapas de amplificación

Los amplificadores de cada sección se implementan utilizando la topología de un amplificador de voltaje no inversor.

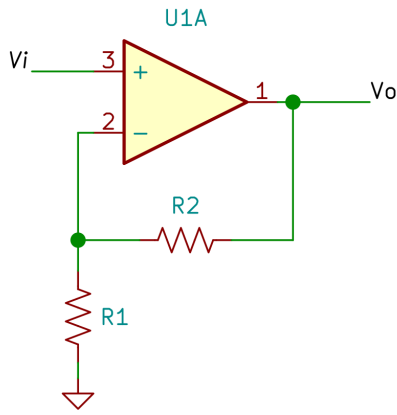


Figura 51: Amplificador de sección

La ganancia ideal en esta configuración se recoge en [48] y es igual a:

$$G_{xi} = \frac{V_o}{V_i} = 1 + \frac{R_2}{R_1} \quad (84)$$

En cuanto a las limitaciones dinámicas del AO, dado que la amplitud de la señal de entrada es pequeña, la limitación principal es el GBP del AO. En [49] se recoge la relación entre el ancho de banda del amplificador, BW, y el GBP del AO:

$$BW = \frac{GBP}{G_{xi}} = GBP \left(\frac{R_1}{R_1 + R_2} \right) \quad (85)$$

Otra limitación importante es el ruido generado por el amplificador, especialmente, para la primera y la segunda etapa de amplificación. A continuación, se presenta el modelo de ruido del amplificador.

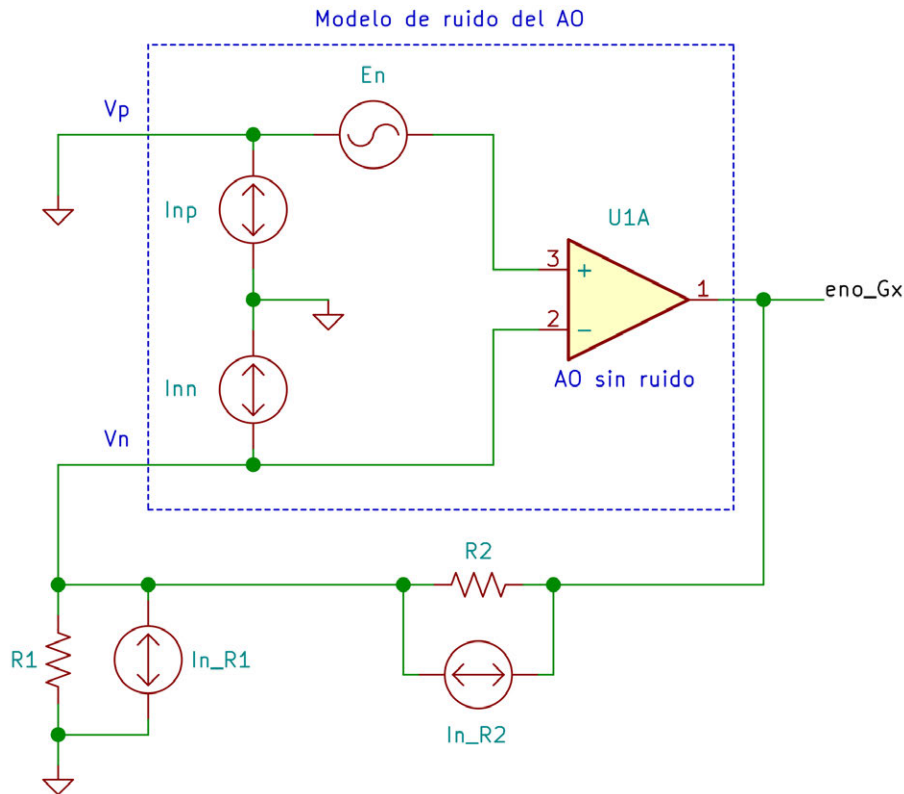


Figura 52: Modelo de ruido del amplificador

Utilizando este modelo, se calcula la densidad de potencia de ruido a la salida: e_{no_Gx} , la cual es:

$$e_{no_Gx}^2 = (i_{nn} R_2)^2 + (i_{nR_1} R_2)^2 + (i_{nR_2} R_2)^2 + \left[e_n \left(1 + \frac{R_2}{R_1} \right) \right]^2$$

$$\begin{cases} i_{n-R_1}^2 = \frac{4kT}{R_1} \\ i_{n-R_2}^2 = \frac{4kT}{R_2} \end{cases} \rightarrow e_{no_Gx}^2 = (i_{nn}^2 + i_{nR_1}^2 + i_{nR_2}^2) R_2^2 + e_n^2 \left(1 + \frac{R_2}{R_1}\right)^2$$

Proyectando la densidad de potencia de ruido a la entrada del amplificador, teniendo en cuenta la ganancia del mismo dada en la ecuación (84), se obtiene la densidad de potencia de ruido a la entrada, e_{ni_Gx} :

$$e_{ni_Gx}^2 = e_n^2 + i_{nn}^2 (R_1 \parallel R_2)^2 + 4kT (R_1 \parallel R_2) \quad (86)$$

Una vez presentadas estas características comunes a todos los amplificadores, se presentan los cálculos de los componentes para garantizar que las ganancias son las especificadas en la Tabla 1. Como se expone en la ecuación (85), el ancho de banda del amplificador está relacionado con la ganancia del mismo. Así, para la máxima ganancia requerida: $G_2 = 20V/V$, el GBP mínimo del AO para que el ancho de banda sea de al menos un orden de magnitud respecto a la frecuencia de operación (10kHz) es:

$$GBP|_{min} = BW \cdot G_{xi} = 100 \text{ kHz} \cdot 20 = 2 \text{ MHz}$$

Los AO utilizados en la placa del emisor, los TLV9064, tienen un GBP de 10MHz [37], por lo que cumple con el requisito para la máxima ganancia requerida. Además, el TLV9064, teniendo una etapa de entrada CMOS, presenta unas características de ruido adecuadas y superiores a otras alternativas: $i_{nn} = i_{np} = 23 \text{ fA}/\sqrt{\text{Hz}}$ y $e_n = 10 \text{ nV}/\sqrt{\text{Hz}}$ [37]. Por tanto, los AO utilizados en los amplificadores serán los TLV9064.

Los valores de los resistores para los amplificadores de cada etapa son:

- **Etapa 1: $G_1 = 3,82 \text{ V/V}$**

$$G_{1i} = 1 + \frac{R_2}{R_1} = 3,82 \rightarrow \frac{R_2}{R_1} = 2,82$$

Los valores de R_1 y R_2 deben cumplir esa relación de ratio. Habitualmente se eligen valores en el rango de kilo ohmios, para que el consumo de los amplificadores este en el rango de mili o micro amperios. Por la expresión, (86), se sabe que los valores de R_1 y R_2 tendrán un impacto en la densidad espectral de ruido, por tanto, cuanto más bajo sean sus valores, mejor será el ruido. Sin embargo, se puede comprobar que mientras los valores de R_1 y R_2 no alcancen valores de decenas o centenas de kilo ohmios, el valor de la densidad espectral de ruido de la expresión (86) está dominado por e_n , es decir, por el ruido del AO. Así pues, los valores R_1 y R_2 serán, normalizados para la serie E-96:

$$R_1 = 3,24 \text{ k}\Omega \quad R_2 = 1,15 \text{ k}\Omega$$

En cuanto a la tolerancia, se puede comprobar que el error relativo de ganancia debido exclusivamente a la tolerancia de los resistores obedece a la siguiente expresión, donde p es la tolerancia de los resistores:

$$\varepsilon_{G_i} = \frac{1}{G_i} \left[\frac{R_2(1 \pm p_1)}{R_1(1 \mp p_2)} + 1 \right] - 1 \quad (87)$$

Asignando arbitrariamente un error máximo debido a la tolerancia de los resistores del 1,5%, y utilizando la expresión (87), se puede comprobar que para resistores del 1% de tolerancia, el error está en el límite aceptable. Debido a esto, se elige para R_2 (el resistor con el valor más grande), una tolerancia máxima de 0,5%. Entonces, los resistores R_1 y R_2 son:

$$R_1 = 3,24 \text{ k}\Omega \pm 0,5\% \quad R_2 = 1,15 \text{ k}\Omega \pm 1\%$$

- **Etapa 2: $G_2 = 20\text{V/V}$**

$$G_{2i} = 1 + \frac{R_2}{R_1} = 20 \rightarrow \frac{R_2}{R_1} = 19$$

Utilizando *resrat*, un programa que busca los valores de los resistores de forma que el error de ratio sea mínimo, y fijando el valor mínimo de R_1 en el rango de kilo ohmios, se eligen los valores de R_1 y R_2 . En cuanto a la tolerancia, utilizando la expresión (87), se comprueba que para una tolerancia del 1%, el error es superior al límite de 1,5%, por lo que se eligen resistores con un 0,5% de tolerancia, los cuales cumplen con el límite de error del 1,5%. Así, los valores de R_1 y R_2 son:

$$R_1 = 21,5 \text{ k}\Omega \pm 0,5\% \quad R_2 = 1,13 \text{ k}\Omega \pm 0,5\%$$

- **Etapa 3: $G_3 = 10\text{V/V}$**

$$G_{3i} = 1 + \frac{R_2}{R_1} = 10 \rightarrow \frac{R_2}{R_1} = 9$$

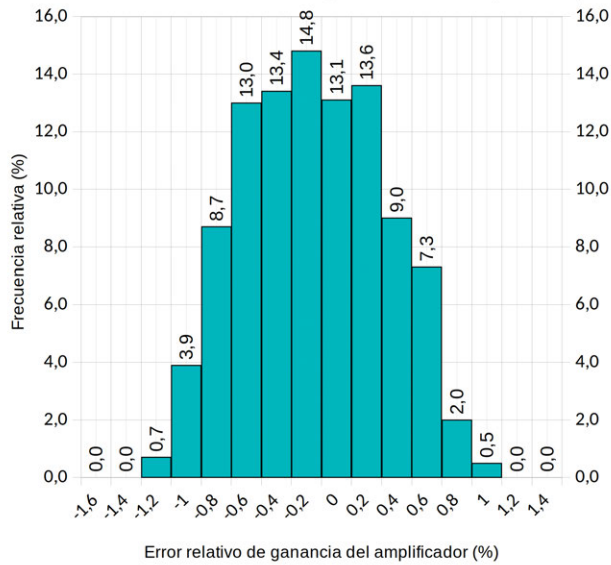
Para hallar los valores que minimizan el error de ratio, se usa *resrat* para elegir los valores. Utilizando la expresión (87), para garantizar que el error de ganancia debido a la tolerancia de los resistores este por debajo del 1,5%, se elige un valor de tolerancia del 0,5% para los resistores. Entonces, los valores de R_1 y R_2 son:

$$R_1 = 10,2 \text{ k}\Omega \pm 0,5\% \quad R_2 = 1,13 \text{ k}\Omega \pm 0,5\%$$

Una vez se han fijado los valores de los componentes, se presenta un análisis de error de ganancia. Durante el diseño, se realizaron análisis teóricos sobre el error de ganancia, pero dado que su presentación consumiría gran cantidad de espacio sin aportar demasiado, solo se expondrán los resultados de los análisis basados en simulación: análisis para el caso peor y análisis basado en el método de Monte Carlo.

Todos estos análisis comparan el valor nominal de ganancia de cada etapa, con el valor del módulo de la respuesta en frecuencia del amplificador para la frecuencia de 10kHz.

Variación del error relativo de ganancia del amp. 1º etapa



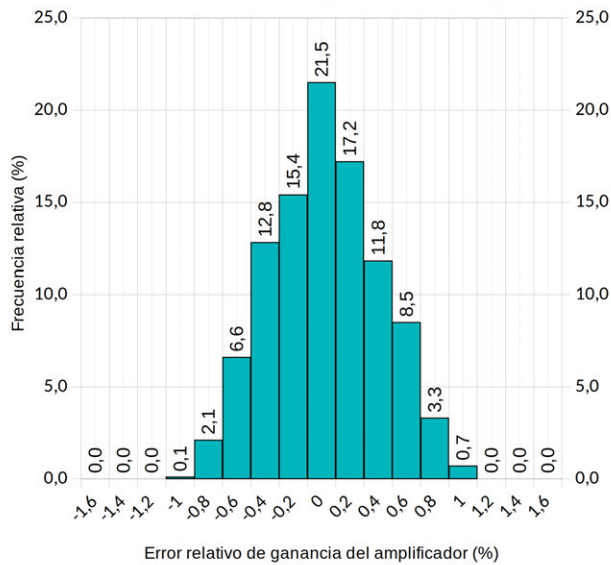
$$\varepsilon_{G1|_{WC}} \in (-1,1\%, 1,2\%) \quad (88)$$

Como se puede observar en los resultados del análisis de Monte Carlo, presentado en la Figura 53, para el 96.8% de las muestras el error está contenido en el intervalo:

$$\varepsilon_{G1|_{MC,96.8\%}} \in (-1\%, 0,8\%) \quad (89)$$

Figura 53: Análisis de Monte Carlo para el amplificador de la primera etapa, N=1000

Variación del error relativo de ganancia del amp. 2º etapa

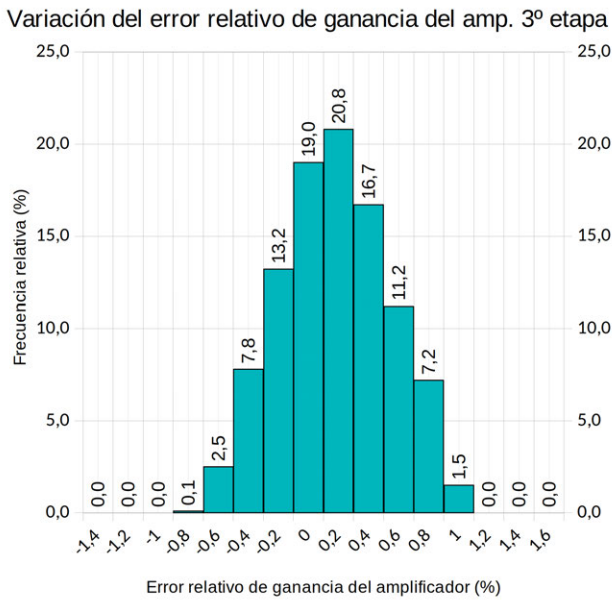


$$\varepsilon_{G2|_{WC}} \in (-0,8\%, 1,1\%) \quad (90)$$

Tal como se muestra en los resultados del análisis de Monte Carlo, presentado en la Figura 54, para el 97.1% de las muestras el error está contenido en el intervalo:

$$\varepsilon_{G2|_{MC,97.1\%}} \in (-0,6\%, 1\%) \quad (91)$$

Figura 54: Análisis de Monte Carlo para el amplificador de la segunda etapa, N=1000



$$\varepsilon_{G3}|_{WC} \in (-0,6\%, 1,2\%) \quad (92)$$

Utilizando los datos del análisis de Monte Carlo, presentado en la Figura 55, para el 97.4% de las muestras el error está contenido en el intervalo:

$$\varepsilon_{G3}|_{MC,97.4\%} \in (-0,4\%, 1\%) \quad (93)$$

Figura 55: Análisis de Monte Carlo para el amplificador de la tercera etapa, N=1000

Respecto al ruido del amplificador, seguidamente, se presenta el cálculo teórico de la densidad espectral de ruido generado por cada amplificador, así como el resultado de las simulaciones. El cálculo teórico se realiza con la ecuación (86).

Las simulaciones y los análisis se pueden consultar en el directorio: *Hardware/SWR-UPM/Simulaciones/RyB_receptor_sensor/Etapas_de_amplificación*, en los archivos que acompañan este proyecto.

El ruido para cada etapa de amplificación es:

$$e_{ni_G1} = 10,68 \frac{nV}{\sqrt{Hz}}$$

$$e_{ni_G1}|_{sim} = 11,94 \frac{nV}{\sqrt{Hz}} \quad (94)$$

$$e_{ni_G2} = 10,85 \frac{nV}{\sqrt{Hz}}$$

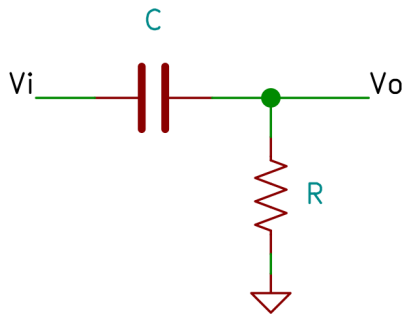
$$e_{ni_G2}|_{sim} = 12,11 \frac{nV}{\sqrt{Hz}} \quad (95)$$

$$e_{ni_G3} = 10,81 \frac{nV}{\sqrt{Hz}}$$

$$e_{ni_G3}|_{sim} = 12,09 \frac{nV}{\sqrt{Hz}} \quad (96)$$

Para las simulaciones, se usa el macromodelo del AO proporcionado por el fabricante, por lo que la densidad de espectral de ruido obtenida será más exacta que la estimada de forma teórica.

4.1.5.4 Filtro de acoplamiento entre etapas



El filtro de acoplamiento entre etapas es un filtro paso-alto de primer orden. Su función de transferencia es:

$$H_{HPF}(jf) = \frac{1}{1 + \frac{f_0}{jf}} \quad (97)$$

donde f_0 : es la frecuencia de corte del filtro.

$$f_0 = \frac{1}{2\pi RC}$$

Figura 56: Filtro de acoplamiento entre etapas

Los componentes elegidos, para una frecuencia de corte, $f_0 = 159\text{Hz}$, son:

$$R = 10\text{k}\Omega \pm 5\% \quad C = 10\text{nF} \pm 5\%$$

El error de ganancia relativa del filtro paso-alto, ϵ_{HPF10k} , se calcula a partir del módulo de la función de transferencia (97):

$$\epsilon_{HPF} = \frac{1}{\sqrt{1 + \left(\frac{f_0}{f}\right)^2}} - 1 \rightarrow \epsilon_{HPF10k} = \frac{1}{\sqrt{1 + \left(\frac{159\text{Hz}}{10\text{kHz}}\right)^2}} - 1 = -1,264 \cdot 10^{-4} \equiv -0,01264\% \quad (98)$$

Como se puede observar, el error relativo de ganancia es muy pequeño en comparación con otros, como el error relativo de ganancia de los acondicionadores o los amplificadores, por lo que su valor es despreciable.

La tolerancia de los componentes no tiene un impacto significativo, pues la frecuencia de corte está alejada casi dos décadas de la frecuencia de la señal sinusoidal (10kHz), por lo que la variación de la frecuencia de corte no produce una variación significativa en el error de ganancia. Por tanto, no se realizarán más análisis para analizar el efecto de la tolerancia.

Respecto del ruido, la única fuente de ruido es el ruido térmico generado por la resistencia, R. El modelo de ruido del filtro es el siguiente:

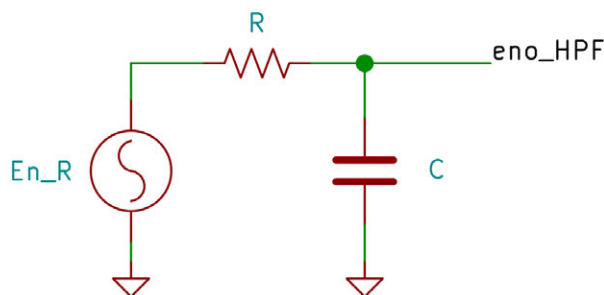


Figura 57: Modelo de ruido del filtro paso-alto

Aplicando la fórmula del divisor de tensión, la densidad espectral de ruido a la salida del filtro es:

$$\left\{ \begin{array}{l} e_{no} = e_{nR} \frac{1}{1 + j 2 \pi f RC} = e_{nR} \frac{1}{1 + \frac{jf}{f_0}} \\ e_R^2 = 4kTR \end{array} \right. \rightarrow \boxed{e_{no_HPF} = \sqrt{4kTR} \frac{1}{\sqrt{1 + \left(\frac{f}{f_0}\right)^2}}} \quad (99)$$

La ecuación (99) muestra que el ruido a la salida depende de la frecuencia, y para el ruido generado por la resistencia, R, el filtro paso-alto se comporta como un filtro paso-bajo, atenuando el ruido por encima de la frecuencia de corte, $f_0 = 159\text{Hz}$. Este es el motivo de que se haya elegido un valor de resistencia tan grande, pese al ruido que introduce. Así, para los valores elegidos para los componentes, en el punto de la frecuencia de operación ($f=10\text{kHz}$), el ruido a la salida del filtro es:

$$e_{no_HPF} = 204,6 \frac{\mu\text{V}}{\sqrt{\text{Hz}}}$$

Este nivel de ruido es muy pequeño en comparación el ruido generado por los amplificadores, basta con comparar el ruido del amplificador de la primera etapa, referido a la entrada, dado en (94) . A la salida del amplificador, el ruido será de $45,6 \text{ nV} / \sqrt{\text{Hz}}$, por lo que el ruido del filtro esta un par de órdenes de magnitud por debajo, por tanto, su efecto puede despreciarse en los cálculos teóricos.

4.1.5.5 Filtro paso-banda

El filtro paso-banda tiene dos funciones: primero, filtrar las componentes en frecuencia generadas por la iluminación artificial, típicamente entre las frecuencias de 50Hz y 100Hz, y segundo, limitar el ruido a la entrada del detector de pico, reduciendo el ancho de banda del sistema.

A la atenuación del filtro paso-banda sobre las componentes generadas por la iluminación, hay que sumar la atenuación introducida por el acondicionador del fotodiodo, la cual, según la simulación, es de 15dB a 100Hz. Para un filtro paso-banda de segundo orden, la atenuación es de 20dB por década, y dado que la frecuencia central del filtro será igual a la frecuencia de operación, $f_0 = 10\text{kHz}$, la atenuación del filtro a 100Hz será de 40dB (dos décadas). Entonces, la atenuación total del filtro y el acondicionador a una frecuencia de 100Hz, será de al menos 55dB. Tras algunas pruebas preliminares, se determinó que este valor es suficiente. No se pudo hallar un valor teórico, ya que no se dispone de datos sobre el tipo de iluminación usada y/o sus componentes espectrales.

A parte de las componentes en frecuencia producto de la iluminación, las señales sinusoidales enviadas por otros robots presentan una cierta distorsión, pero dado que su THD es de 0.81%, y que el filtro paso-banda introducirá una atenuación adicional, no se considerará su efecto.

Además de la frecuencia central del filtro, es necesario determinar el ancho de banda. El ruido será menor cuanto menor sea el ancho de banda. Sin embargo, cuanto menor sea el ancho de banda, mayor será el retardo de grupo y mayor será el impacto de desviación del valor real de f_0 , respecto del valor nominal, debido a la tolerancia de los componentes.

Descripción de la solución propuesta

El retardo de grupo es el tiempo que tarda una señal de banda estrecha en atravesar el filtro a una frecuencia determinada. La señal emitida por cada robot se activa y desactiva a una frecuencia que depende del número de robots del enjambre, por lo que la señal recibida puede modelarse como una señal sinusoidal modulada por una señal cuadrada de frecuencia menor. Así pues, el retardo de grupo representa el tiempo de tránsito de dicha señal cuadrada a través del filtro.

El diseño del filtro se realizó siguiendo un proceso iterativo, empezando por un ancho de banda, BW , de 100Hz y una frecuencia central, f_o , de 10kHz, lo que implica que el factor de calidad del filtro, Q , es 100. Para implementar esta primera versión, se utilizó un filtro *bicua*. Si bien el filtro cumplía con los requisitos para los valores ideales de los componentes, tras normalizarlos a unos valores realizables, el error de ganancia máximo, debido a la tolerancia de los componentes era de un -37,9%, debido a la variación de la frecuencia central del filtro, f_o .

Se realizaron varias iteraciones, aumentando el ancho de banda, hasta que se alcanzó un ancho de banda suficiente como para que las tolerancias de los componentes no produjeran un error de ganancia demasiado alto. Así, el ancho de banda realizable quedó en 1,6kHz, lo que implica un que el factor Q del filtro era de: 6,25. Para valores del factor de calidad Q por debajo de 10, [44], la implementación del filtro se puede realizar con una topología más sencilla que la de un filtro *bicua*, lo que reduce el número de componentes, y por lo tanto, minimiza el efecto de las tolerancias. Se escogió la topología *Multiple Feedback Filter* (MFF) paso-banda, la cual tiene características de sensibilidad muy similares a las del filtro con topología *bicua*.

En la Figura 58 se muestra el esquema del filtro MFF paso-banda [50]. Las ecuaciones de diseño, si se cumple que $C_1 = C_2 = C$ son:

$$R_{1A} = \frac{Q}{2\pi H_{BP0} f_o} \quad R_{1B} = \frac{Q}{2\pi H_{BP0} f_o} \quad R_2 = \frac{2Q}{2\pi f_o C} \quad (100)$$

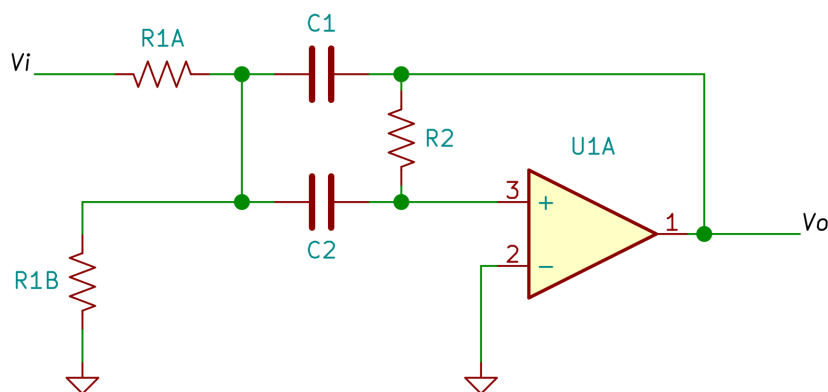


Figura 58: Esquema del filtro MFF paso-banda

Además de estas ecuaciones, se debe tener en cuenta que el GBP del AO producirá una variación en la frecuencia central del filtro, f_o , tal como se expone en [51]:

$$\frac{\Delta f_o}{f} = \frac{-Q f_o}{GBP} \quad (101)$$

Así, para reducir el stock de componentes, si se utilizan los mismos AO usados en los amplificadores, los TLV9064, los cuales tienen un GBP de 10MHz, el error en la frecuencia central del filtro, f_0 , es:

$$\frac{\Delta f_0}{f} = -6,25 \cdot 10^{-3} \equiv -0,625\% \rightarrow \Delta f_0 = 62,5 \text{ Hz}$$

Si bien la variación es pequeña teniendo en cuenta el ancho de banda de 1,6kHz, se predistorsiona el valor de la frecuencia central del filtro para corregir esta desviación. La nueva frecuencia central es de $f_0 = 10063 \text{ Hz}$. Con este valor, si se escoge $C=820 \text{ pF}$, teniendo en cuenta que $H_{BPO} = 1$, y $Q=6,25$, aplicando las ecuaciones de diseño (100) y normalizando los valores obtenidos a las series con menor tolerancia (E-96 y E-192), los valores de los componentes son:

$$R_{1A} = 120 \text{ k}\Omega \pm 0,1\% \quad R_{1B} = 1 \text{ k}\Omega \pm 0,1\% \quad R_{1B}' = 526 \Omega \pm 0,1\% \\ R_2 = 240 \text{ k} \pm 0,1\% \quad C_1 = C_2 = 820 \text{ pF} \pm 1\%$$

Las simulaciones pueden consultarse en la ruta: *Hardware/SWR-UPM/Diseño_sensor_R&B/Simulaciones/RyB_receptor_sensor/Filtro_paso-banda*, en los archivos que acompañan a este proyecto. El retardo de grupo a la frecuencia central del filtro, obtenido con esta configuración es de:

$$t_g = 200 \mu s \quad (102)$$

En cuanto a la atenuación para las componentes debido a la iluminación, para el filtro, y para el filtro y el acondicionador, los valores obtenidos en la simulación son:

$$A_{BP100 \text{ Hz}} = 56 \text{ dB} \quad A_{T100 \text{ Hz}} = A_{BP100 \text{ Hz}} + A_{G0_100 \text{ Hz}} = 56 \text{ dB} + 15 \text{ dB} = 71 \text{ dB} \quad (103)$$

Una vez establecidos los valores de los componentes, se realizó un análisis de tolerancia para el caso peor, y un análisis de tolerancia utilizando el método de Monte Carlo. Los resultados se exponen a continuación:

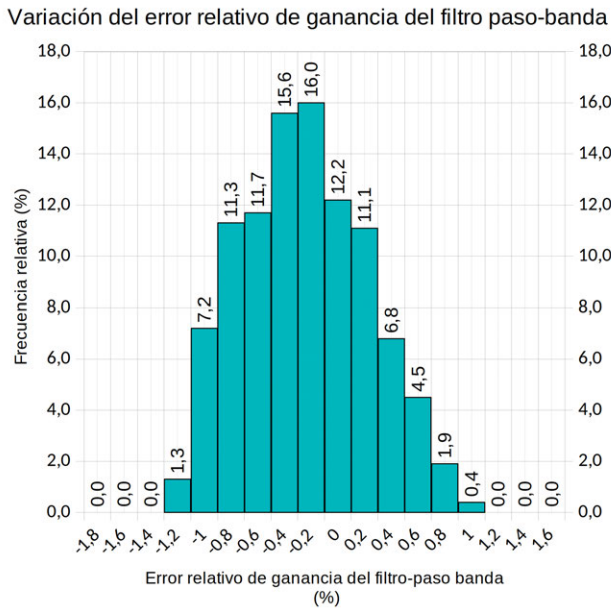


Figura 59: Análisis de Monte Carlo para el filtro paso-banda, N=1000

$$\varepsilon_{BPF\ 10k}|_{WC} \in (-1,5\%, 1,2\%) \quad (104)$$

Según los resultados del análisis de Monte Carlo, presentado en la Figura 59, para el 98.3% de las muestras el error está contenido en el intervalo:

$$\varepsilon_{BPF\ 10k}|_{MC,98.3\%} \in (-1\%, 1\%) \quad (105)$$

El filtro paso-banda, al igual que el resto de bloques, genera una cierta cantidad de ruido que se acoplará a la señal recibida. Como el filtro no está incluido en la etapa de amplificación de la señal, el ruido generado tendrá un efecto mucho menor que el ruido de los amplificadores o el del acondicionador de los fotodiodos. Dado que para el filtro paso-banda, la densidad espectral de ruido a la salida del filtro es una función de la frecuencia, lo más sencillo es utilizar el simulador para integrar el ruido a la salida del filtro, obteniendo una estimación del ruido RMS. Como la ganancia del filtro es unitaria, el ruido proyectado a la entrada del filtro coincide con el ruido a la salida:

$$E_{i_HBP} = E_{o_HBP} = 70,571 \mu V \quad (106)$$

El ruido a la entrada no es el ruido real generado por el filtro a la salida del amplificador, pues este depende de la impedancia de salida del amplificador y de la impedancia de entrada del filtro. Dado que la impedancia de salida del amplificador tendrá un valor pequeño, y la impedancia de entrada del filtro, tendrá un valor grande, el efecto del ruido generado por el filtro en la etapa de amplificación será mínimo. Estos efectos se tendrán en cuenta en el apartado de cálculo de ruido total.

4.1.5.6 Detector de pico

Uno de los bloques que más tiempo requirió para su diseño fue el detector de pico. La versión aquí recogida es la segunda versión, o revisión A del hardware original, como se ha señalado brevemente en la introducción. De todo el diseño, la parte que más modificaciones requirió fue el detector de pico. El diseño original, que puede consultarse en los archivos que acompañan a este proyecto, bajo la ruta: *Hardware/SWR-UPM/Boards/Version_0*, estaba basado en un detector de pico de precisión, en concreto en el diseño recogido en [52].

La frecuencia de operación del *sensor de rango y rumbo* de la versión 0, era de 100kHz, con el objetivo de mejorar la frecuencia de actualización de la posición de los robots del enjambre. Para asegurar que el diseño funcionaba, se realizó un análisis teórico y un conjunto de simulaciones, y dado que todo era correcto, este fue el diseño que se incluyó en la primera versión de las PCB de este robot.

Por desgracia, el análisis teórico y las simulaciones no mostraron un fallo fundamental: la relación entre la fase de la señal sinusoidal cuyo valor pico iba a medirse, y el instante en el que se descargaba el condensador. Así, en las simulaciones el reinicio del condensador siempre se producía cuando la amplitud de la señal a medir era máxima, y mientras esta condición se cumpliera, no había ningún error. Sin embargo, si el condensador se reiniciaba en otro momento, el error de medida era muy importante.

Un análisis más completo demostró que este comportamiento se debía a la respuesta transitoria subamortiguada, causada por un margen de fase bastante reducido. Para solucionar este defecto, se diseñó una red de compensación que aumentaba el margen de fase, y si bien esto mejoró el funcionamiento, el defecto seguía presentándose. Se invirtió gran cantidad de tiempo en diagnosticar el problema, y se realizaron análisis, simulaciones, pruebas y modificaciones sobre el circuito, y después de todo, se decidió abandonar ese diseño porque no se encontró la forma de solucionar el problema en un tiempo razonable.

Se exponen estos hechos, porque el diseño presentado a continuación se desvía de los diseños convencionales de un detector de pico de precisión, y es necesario motivar esta elección. Otras soluciones a este problema, como el uso de ADCs para muestrear la señal y calcular su valor pico, se descartaron por la falta de disponibilidad de componentes, y el precio de los mismos. Por otro lado, el uso de detector de pico con diodos, utilizado en RF se descartó por el error que presentan para señales de pequeña amplitud.

Con todas estas consideraciones, y tomando algunas ideas de las notas de aplicación [53] y [54], se realiza el siguiente diseño para el detector de pico:

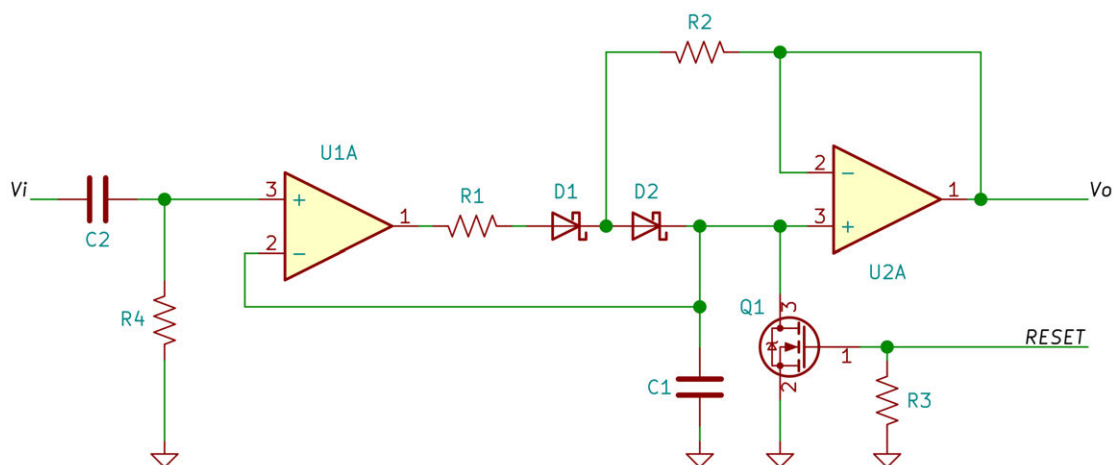


Figura 60: Detector de pico

Empezando por la entrada, la resistencia R_4 y el condensador C_5 forman un filtro paso-alto, con la misma frecuencia de corte que el filtro paso-alto de acoplamiento entre etapas. El objetivo de este filtro es eliminar la componente continua producida por la tensión de offset del AO del filtro paso-banda. U1A es un comparador con salida *push-pull*, y U2A es un amplificador operacional, ambos están alimentados con una tensión bipolar de $\pm 2,5V$.

El funcionamiento de este circuito es el siguiente: si se supone el condensador C1 esta descargado, cuando se sitúa una señal a la entrada del circuito, tras atravesar el filtro, llega a la entrada positiva del comparador. Como el condensador esta descargado, cuando el valor de la señal de entrada supere 0V, el comparador U1A pondrá su salida $V_{CC} = 2,5V$. Esto provocará que el condensador C1 comience a cargarse a través de la resistencia R_1 y los diodos D_1 y D_2 .

Cuando la tensión del condensador alcance y supere ligeramente la tensión de la señal de entrada, el comparador U1A, pondrá su salida a $V_{SS} = -2,5V$. Esto provocará que el diodo D_1 quede polarizado de forma inversa, por lo que el proceso de carga del condensador C_1 se detendrá. Todos los diodos tienen una corriente de fuga que depende de la tensión de polarización inversa. En otras aplicaciones, esta corriente de fuga no supone un problema, sin embargo, en esta aplicación es un factor a tener en cuenta, pues provocaría que el condensador C_1 se descargase.

Para reducir al mínimo la corriente de fuga a través del diodo D_1 , el AO U2A, el cual está configurado como seguidor de tensión, proporciona a D_1 un camino por R_2 para obtener la corriente de fuga de la salida del AO, U2A. Esto produce que en R_2 caiga una tensión extremadamente pequeña, y como D_2 está conectado a V_p , y por el cortocircuito virtual del AO, $V_n=V_p$, D_2 estará polarizado en inversa, con una tensión de polarización inversa muy pequeña, lo que provoca que la corriente de fuga a través de D_2 sea del orden de pico-amperios. Si el comparador y el amplificador operacional elegidos tienen etapas de entrada basadas en MOSFET (CMOS), la corriente de descarga del condensador será tan pequeña, que la única corriente que introducirá un error en la medida será la corriente de fuga a través del transistor Q_1 .

El transistor Q_1 permite descargar el condensador tras una lectura, cuando recibe un pulso positivo a través de la línea de *RESET*.

A parte de la corriente de descarga del condensador C_1 , hay otras fuentes de error que deben tenerse en cuenta. En primer lugar, el comparador no conmutara su salida de forma instantánea, tardará un cierto tiempo, el cual depende de la diferencia de tensión entre su entrada positiva y su entrada negativa. En las hojas de datos de los comparadores, esta diferencia de tensión se denomina tensión de *over-drive*, identificado como V_{OD} , y el tiempo de que tarda en conmutar la salida tras alcanzar la tensión de *over-drive*, es el tiempo de propagación, identificado como t_P .

Así, una vez la tensión del condensador alcance el valor de la tensión pico, y lo supere con una tensión V_{OD} , el condensador continuará cargándose durante el tiempo de propagación del comparador. Esto introducirá un error en la medida que depende del tiempo de propagación del comparador y de la tensión de *over-drive*. En las hojas de datos se recogen estos parámetros para algunas condiciones, no obstante, dado que las condiciones exactas no coinciden con las expuestas en la hoja de datos, los cálculos presentados son una aproximación.

Utilizando la ecuación del condensador, y suponiendo que la tensión de salida del comparador coincide con V_{CC} , la tensión en el condensador, cuya tensión inicial es v_{C0} , transcurrido un tiempo t , para unos diodos cuya tensión de polarización es V_{yDx} , es:

$$v_C = (V_{CC} - V_{yD1} - V_{yD2}) \left(1 - e^{-\frac{t}{R_1 C_1}} \right) + v_{C0} e^{-\frac{t}{R_1 C_1}}$$

A partir de esta expresión, teniendo en cuenta que cuando el conmutador invierta su salida, el condensador debe haber alcanzado la tensión de pico de la señal de entrada más la tensión de *over-drive* del comparador, el error introducido debido al tiempo de propagación es:

$$\varepsilon_{PT} = (V_{CC} - V_{yD1} - V_{yD2}) \left(1 - e^{-\frac{-t_p}{R_1 C_1}} \right) - (v_p + v_{OD}) \left(1 - e^{-\frac{-t_p}{R_1 C_1}} \right) + v_{OD} \quad (107)$$

Cuando el valor de pico de la señal de entrada, v_p , sea mínimo, el error debido al tiempo de propagación será máximo. Así, para el caso extremo en el que $v_p = 0$:

$$\varepsilon_{PT|_{max}} = (V_{CC} - V_{yD1} - V_{yD2}) \left(1 - e^{-\frac{-t_p}{R_1 C_1}} \right) + v_{OD} \left(e^{-\frac{-t_p}{R_1 C_1}} \right) \quad (108)$$

Como ya se ha mencionado, otra fuente de error es la corriente de descarga del condensador. El ADC tardará un tiempo, t_s , en leer el valor de tensión a la salida, y durante ese tiempo, la carga del condensador variará por el efecto de las corrientes de fuga del transistor Q_1 , la entrada del comparador y la entrada del AO. Utilizando la ecuación del condensador, para una corriente de fuga constante, el error introducido por esta en el valor de tensión del condensador es:

$$\varepsilon_F = \frac{I_F}{C_1} t_s \quad (109)$$

Además, ha de tenerse en cuenta que tanto el AO como el comparador presentan un error de offset, modelado como una fuente de tensión en su entrada positiva. Entonces, sumando los errores de las expresiones (108) y (109), y teniendo en cuenta el error de offset, el error total que el detector de pico introducirá en la medida es:

$$\varepsilon_T = V_{OS_AO} + V_{OS_CMP} + \varepsilon_{PT|_{max}} + (I_{P_OA} + I_{P_CMP} + I_{LK}) \frac{t_s}{C_1} \quad (110)$$

A la vista de la expresión (108), se puede comprobar que el tiempo de propagación del comparador, t_p , influirá en el error, por lo que el comparador se elegirá buscando minimizar este parámetro. Otro factor a tener en cuenta, es que al igual que el AO, el comparador tendrá que ser *rail-to-rail*. Estas características las encontramos en el comparador TS3011 del fabricante STMicroelectronics. El amplificador operacional elegido es el TLV9064, ya que esta elección simplifica el *stock* de componentes y el *slew rate* del mismo es suficiente.

En cuanto a los valores de los componentes, los valores de R_1 y C_1 influirán de forma significativa en el error producido por el tiempo de propagación, (108), y en el tiempo que tarda el detector de pico en alcanzar el valor máximo: t_{pks} . Este último parámetro debe estimarse utilizando el simulador; así para obtener los valores de R_1 y C_1 , se asignaron diversos valores, y se calculó el error

Descripción de la solución propuesta

utilizando la ecuación: (108). Tras algunas iteraciones, los valores que producían un error razonable, con un t_{PKS} máximo inferior a 1ms, fueron:

$$R_1=1k\Omega \pm 5\% \quad C_1=10nF \pm 5\%$$

R_2 influye en la tensión inversa de polarización del diodo D_2 , debida a la corriente de fugas del diodo D_1 . Cuanto mayor sea su valor, mayor será la tensión de polarización inversa, y más rápido dejará de conducir D_2 cuando el comparador conmute su salida a V_{SS} . Dada la dificultad de realizar un análisis teórico que justifique su elección, se probaron varios valores en simulación, y se determinó que el valor adecuado es:

$$R_2=100k\Omega \pm 5\%$$

El transistor Q_1 seleccionado es el Si2300DS, un mosfet utilizando en otras placas del robot. Las simulaciones de este bloque pueden consultarse en la ruta: *Hardware/SWR-UPM/Simulaciones/Detector_de_pico*, en los archivos que acompañan a este proyecto.

Para calcular el error total del detector de pico (110), utilizando las hojas de datos del OA [37], TLV9064, y del comparador [55], TS3011, se obtienen los valores de tensión de offset, V_{OS} , y corriente entrada al comparador y AO, I_p . Para obtener el valor de corriente de fuga a través de Q_1 , se utiliza el macromodelo proporcionado por el fabricante. Dado que la corriente de fuga del transistor Q_1 es varios ordenes de magnitud superior a la suma de la corriente I_p para el AO y el comparador, se despreciarán estos valores.

$$v_{OD}=5mV \rightarrow t_s=16ns \quad v_{OD}=5mV \rightarrow t_s=16ns$$

$$V_{OS_CMP}|_{typ}=0,2mV \quad V_{OS_CMP}|_{max}=7mV$$

$$V_{OS_AO}|_{typ}=0,3mV \quad V_{OS_AO}|_{max}=1,6mV$$

$$I_{LK}=2,2nA, V_{GS}=0V \text{ y } V_{DS}=2,048V$$

Así, para el valor típico y máximo de V_{OS} , el error total del detector de pico es:

$$\varepsilon_T|_{typ}=8,2mV \quad \varepsilon_T|_{max}=16,6mV$$

Teniendo en cuenta estos valores, dado que estos son una aproximación por exceso y su contribución al error en la señal es más o menos aleatoria, ya que el valor real depende de la amplitud de la señal de entrada, se puede considerar el error como una fuente de ruido gaussiano. Así, si el error total del detector de pico es el valor máximo, el ruido RMS que introducirá el detector de pico en el camino de la señal puede aproximarse como [32]:

$$E_{O_D.PICO} = \frac{\varepsilon_T|_{typ}}{6,6} = 1,24mV \quad (111)$$

4.1.5.7 Fuente de alimentación

Para alimentar los amplificadores, filtros paso-banda y detectores de pico, se utiliza una tensión bipolar de $\pm 2,5V$. La tensión positiva, $V_{CC}=2,5V$ es producida utilizando un regulador lineal, que disminuye la tensión desde el raíl de tensión de la batería, V_{BAT} , desde los 8,4V hasta los 2,5V.

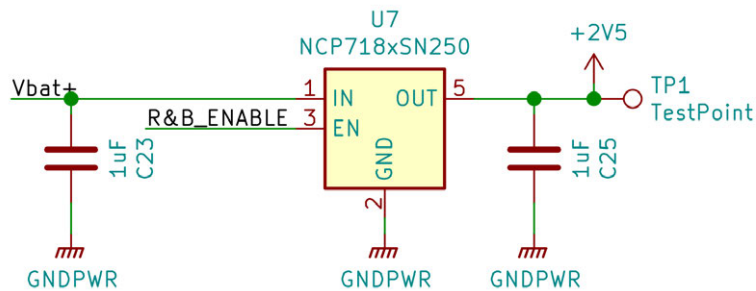


Figura 61: Fuente de alimentación V_{CC}

La tensión negativa es generada utilizando una bomba de carga, ya que la corriente consumida en el raíl de V_{SS} no es muy grande, una bomba de carga proporciona una tensión más limpia que otras soluciones, como un convertidor elevador-reductor. Dado que la bomba de carga elegida tiene una tensión de entrada máxima inferior a la tensión del raíl de alimentación de la batería, V_{BAT} (8,4V), se utiliza un regulador lineal para disminuir está a un valor adecuado (3,3V).

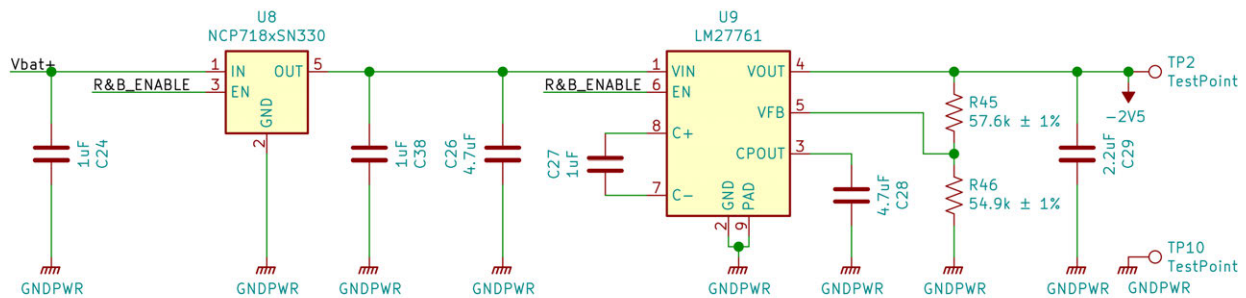


Figura 62: Fuente de alimentación V_{SS}

La tensión de polarización de los fotodiodos, V_{BIAS} , se obtiene utilizando un conversor elevador, el cual aumenta la tensión desde 8,4V, la tensión del raíl V_{BAT} , hasta 34V.

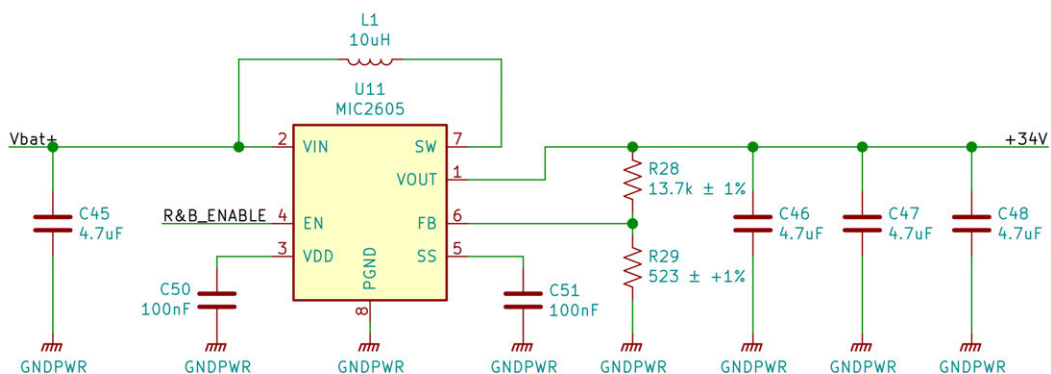


Figura 63: Fuente de alimentación V_{BIAS}

Descripción de la solución propuesta

Para cumplir con la especificación EH-11, la alimentación de la placa puede ser habilitada o deshabilitada a través de la línea R&B_ENABLE, la cual está conectada al microcontrolador principal.

En cuanto al consumo, la corriente máxima consumida por los raíles de alimentación es:

Corriente total pico raíl +2,5V, V_{CC} : 5,9mA

Corriente total pico raíl -2,5V, V_{SS} : 4,6mA

Corriente total pico raíl +34V, V_{BIAS} : 24mA (suponiendo que todos los acondicionadores estén saturados y añadiendo un factor de seguridad del 25%).

La corriente de pico consumida por el convertidor elevador es de 110mA, y la corriente RMS es de 125mAef. Así, la corriente pico total consumida a través del raíl V_{BAT} es:

Corriente total pico raíl V_{BAT} : 120,5mA (112)

La corriente RMS consumida por la placa, aproximando por exceso la corriente RMS de los raíles de V_{CC} y V_{SS} a su valor pico, es:

Corriente total RMS máxima raíl V_{BAT} : 135,5mAef

4.1.5.8 Cálculo del ruido total

A continuación, se calculará el ruido RMS a la salida de cada etapa, de forma que se compruebe que el ruido total coincide con el ruido calculado en Tabla 3. Hay dos formas de realizar este cálculo: de forma teórica, teniendo en cuenta las densidades espectrales de ruido de cada bloque, (82), (83), (94), (95), y (96), y el ancho de banda del filtro paso-banda; o mediante simulación, utilizando los modelos de ruido presentados de cada bloque.

En la hoja de cálculo utilizada en el apartado 4.1.3.5 se realiza el cálculo de forma teórica, tomando la densidad espectral de ruido como constante en el entorno de la frecuencia de operación: 10kHz. Para simplificar la presentación de resultados, y comprobar que este análisis es correcto, aquí se realiza el cálculo del error RMS mediante simulación. Para ello, en la ruta *Hardware/SWR-UPM/Simulaciones/RyB_receptor_sensor/Análisis_de_ruido_completo*, se presenta un modelo completo de ruido del receptor.

Utilizando este modelo, se obtiene que a la entrada del filtro paso-banda, las densidades espectrales de ruido son quasi-constantes y sus valores son:

$$e_{i1} = 61,7 \frac{nV}{\sqrt{Hz}} \quad e_{i2} = 1,26 \frac{\mu V}{\sqrt{Hz}} \quad e_{i3} = 12,61 \frac{nV}{\sqrt{Hz}}$$

Entonces, utilizando el simulador e integrando el ruido desde 10mHz hasta 100MHz, se obtiene que el ruido RMS a la salida de los filtros paso-banda es:

$$V_{O_PBNT1} = 70,64 \mu V \quad V_{O_PBNT2} = 94,43 \mu V \quad V_{O_PBNT3} = 632,94 \mu V \quad (113)$$

El ruido de cuantificación introducido por el ADC, es:

$$V_{O_ADC} = \frac{1 \text{ LSB}}{\sqrt{12}} = \frac{V_{ref}}{2^{ADC_N} \sqrt{12}} = \frac{2,048 \text{ V}}{2^{10} \sqrt{12}} = 577 \mu\text{V} \quad (114)$$

Otra fuente de ruido, que no se ha explicado pero se ha tenido en cuenta en el diseño, es la tensión de rizado producida por el conversor elevador que genera la tensión de V_{BIAS} . El ruido RMS que representa esta fuente de ruido es:

$$V_{O_NBIAS} = 5 \mu\text{V} \quad (115)$$

Entonces, teniendo en cuenta las contribuciones de ruido RMS dadas en (111), (113), (114) y (115), el ruido pico total a la salida es:

$$V_{N(1)} = \left(\sqrt{V_{O_PBNT1}^2 + E_{O_D.PICO}^2 + V_{O_ADC}^2 + V_{O_NBIAS}^2} \right) 6,6 \rightarrow$$

$$V_{N(1)} = \left(\sqrt{(70,64 \mu\text{V})^2 + (1,24 \text{ mV})^2 + (577 \mu\text{V})^2 + (5 \mu\text{V})^2} \right) 6,6 = 9,04 \text{ mV} \quad (116)$$

$$V_{N(2)} = \left(\sqrt{V_{O_PBNT2}^2 + E_{O_D.PICO}^2 + V_{O_ADC}^2 + V_{O_NBIAS}^2} \right) 6,6 \rightarrow$$

$$V_{N(2)} = \left(\sqrt{(94,43 \mu\text{V})^2 + (1,24 \text{ mV})^2 + (577 \mu\text{V})^2 + (5 \mu\text{V})^2} \right) 6,6 = 9,05 \text{ mV} \quad (117)$$

$$V_{N(3)} = \left(\sqrt{V_{O_PBNT3}^2 + E_{O_D.PICO}^2 + V_{O_ADC}^2 + V_{O_NBIAS}^2} \right) 6,6 \rightarrow$$

$$V_{N(3)} = \left(\sqrt{(632,94 \mu\text{V})^2 + (1,24 \text{ mV})^2 + (577 \mu\text{V})^2 + (5 \mu\text{V})^2} \right) 6,6 = 9,95 \text{ mV} \quad (118)$$

Como se puede observar, estos valores son muy similares a los calculados usando la hoja de cálculo, mostrados en la Tabla 3, por lo que se demuestra que los cálculos son correctos.

4.1.5.9 Cálculo del error total de ganancia

Se puede comprobar que los errores de ganancia para el amplificador de la segunda y tercera etapa, dados en (90), (91), (92) y (93), están por debajo de los valores máximos que provocarían que los se produjera un “desolapamiento” de las secciones de la función de transferencia, establecidos en la Tabla 4.

El error relativo de ganancia del receptor para la salida de cada etapa, dependerá de los errores de relativos introducidos por cada bloque sobre la amplitud de la señal. Para cada una de las tres etapas o secciones de la función de transferencia, este error puede calcularse como:

$$\varepsilon_{GT1} = (1 + \varepsilon_{G0})(1 + \varepsilon_{G1})(1 + \varepsilon_{BPF10k}) - 1$$

$$\varepsilon_{GT2} = (1 + \varepsilon_{G0})(1 + \varepsilon_{G1})(1 + \varepsilon_{G2})(1 + \varepsilon_{BPF10k}) - 1$$

$$\varepsilon_{GT3} = (1 + \varepsilon_{G0})(1 + \varepsilon_{G1})(1 + \varepsilon_{G2})(1 + \varepsilon_{G3})(1 + \varepsilon_{BPF10k}) - 1$$

Entonces, si se toman los valores obtenidos mediante a través de un análisis para el caso peor, dados en los intervalos (78), (88), (90), (92), y (104), el error relativo de ganancia máximo para cada etapa, esta contenido en los intervalos:

Descripción de la solución propuesta

$$\boxed{\varepsilon_{GT1}|_{max} \in (-4,3\%, 2,7\%)} \quad \boxed{\varepsilon_{GT2}|_{max} \in (-5,1\%, 3,8\%)} \quad \boxed{\varepsilon_{GT3}|_{max} \in (-5,7\%, 5,1\%)} \quad (119)$$

De forma análoga, tomando los valores obtenidos mediante el análisis de Monte Carlo, dados en los intervalos (79), (89), (91), (93), y (105), como los valores típicos, el error relativo de ganancia típico para cada etapa es:

$$\boxed{\varepsilon_{GT1}|_{typ} \in (-3,5\%, 1,8\%)} \quad \boxed{\varepsilon_{GT2}|_{typ} \in (-4,1\%, 2,8\%)} \quad \boxed{\varepsilon_{GT3}|_{typ} \in (-4,5\%, 3,8\%)} \quad (120)$$

4.1.5.10 Diseño de la placa y piezas en 3D

Para el diseño de la PCB se siguieron todas las directrices para el diseño de sistemas de medida analógicos de bajo ruido: masas separadas para las fuentes de alimentación y para la circuitería analógica, pistas de retorno de masa unidas en un mismo punto para los amplificadores y filtros, dos planos de masa separados uno para la zona que contiene las fuentes de alimentación, y otro la zona que donde se sitúa la circuitería analógica, etc. En especial, se prestó atención a que no se produjeran retornos de masa de las fuentes de alimentación a través de las pistas de masa de la zona analógica, ya que esto provocaría problemas de ruido muy significativos.

Los resultados del diseño se presentan a continuación:

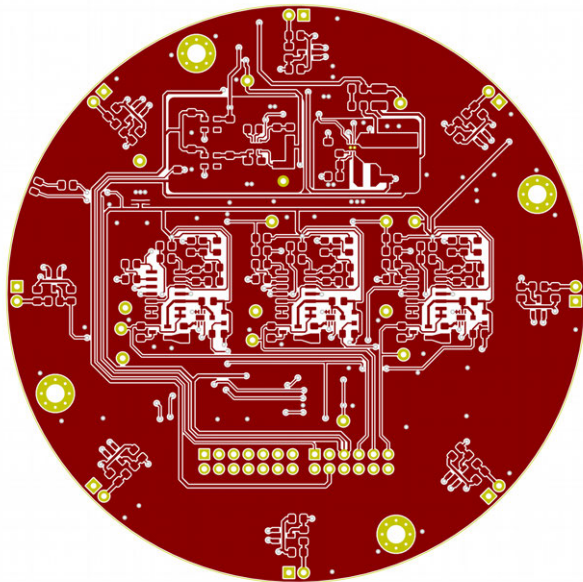


Imagen 33: Capa top de la R&B_sensor_rx

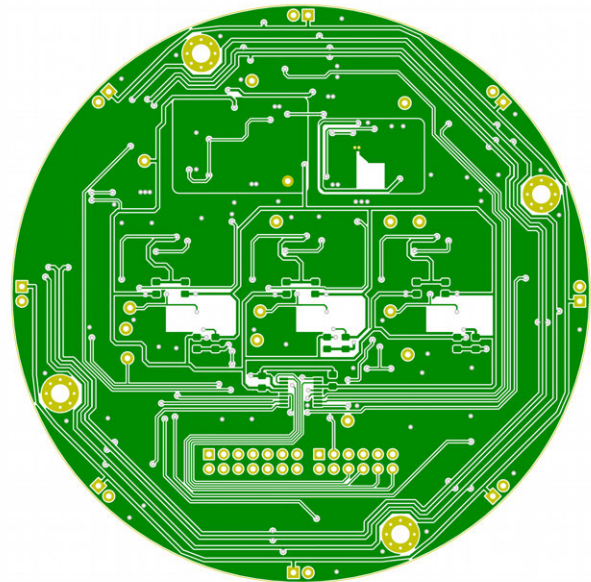


Imagen 34: Capa bottom de la R&B_sensor_tx

Al igual que en el caso de la placa del emisor del *sensor de rango y rumbo*, se diseñó una pieza para imprimir en 3D que actúa de soporte para los fotodiodos, la cual garantiza que tanto los fotodiodos como los emisores IR están en el mismo plano.

Esta pieza está archivada bajo el nombre: *soporte_receptores_IR_rediseño_v1_reforzado.FCStd*, en los archivos que acompañan a este proyecto.

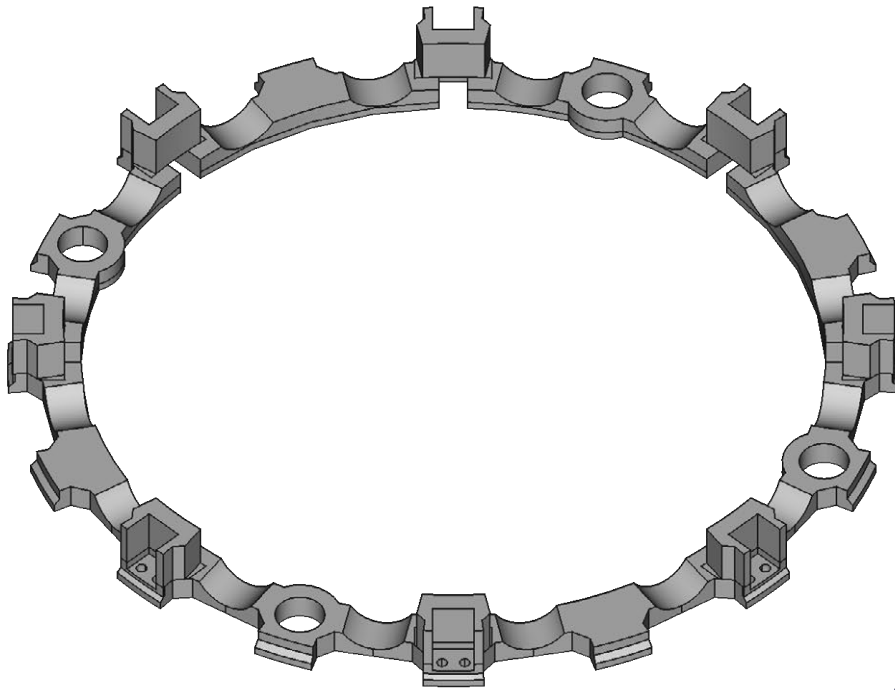


Imagen 35: Soporte de los fotodiodos

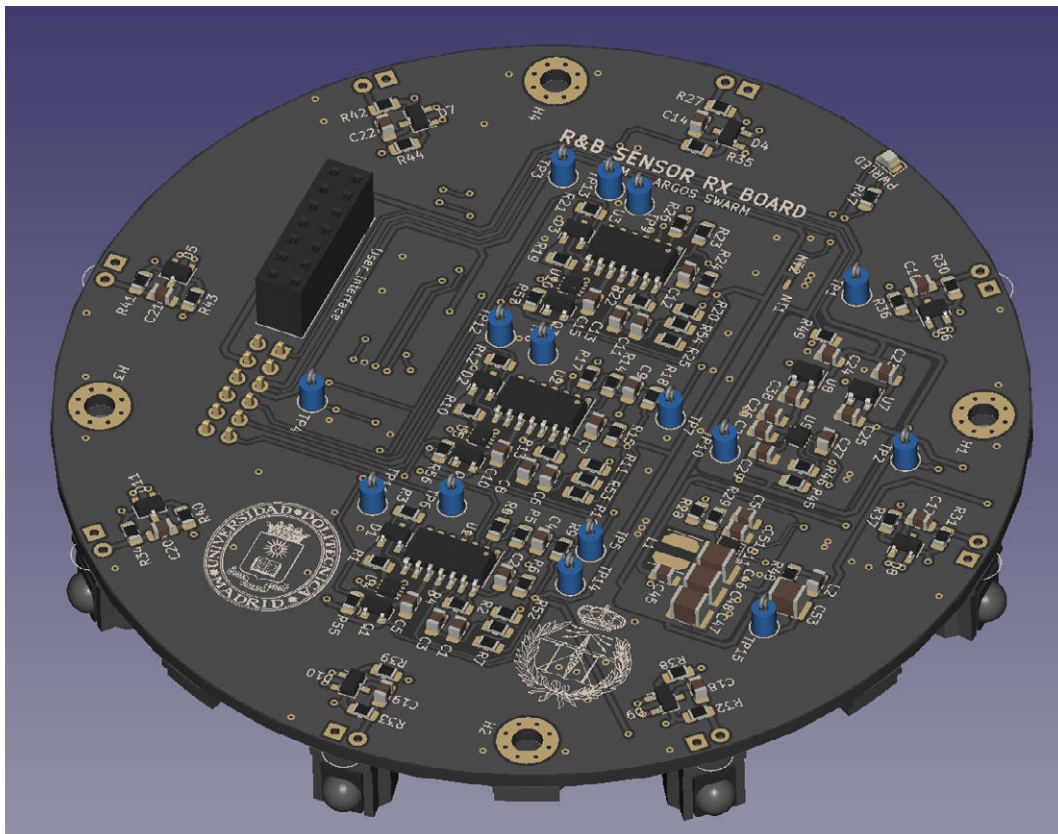


Imagen 36: Simulación 3D de la placa terminada, vista superior

4.1.6 Nivel 5 - Placa de interfaz de usuario (*User_interface_board*)

Especificaciones que satisface: *EH-2*, *EH-6*

La placa de interfaz de usuario está situada en la parte superior del robot y dispone de una pantalla grafica OLED de 0.96 pulgadas, un botón de parada de emergencia, un joystick de cinco botones, y un zumbador. Esta interfaz permite implementar un sistema de menús simple, e informar al usuario de algunos parámetros: estado de la batería, conexión con el enjambre, etc.

Para evitar malgastar los recursos del microcontrolador principal con rutinas para gestionar los rebotes de los pulsadores, se implementó un circuito “anti-rebotes” por hardware, el cual utiliza un *schmitt-trigger* para garantizar que la señal generada por cada pulsador está dentro de los niveles lógicos del microcontrolador principal.

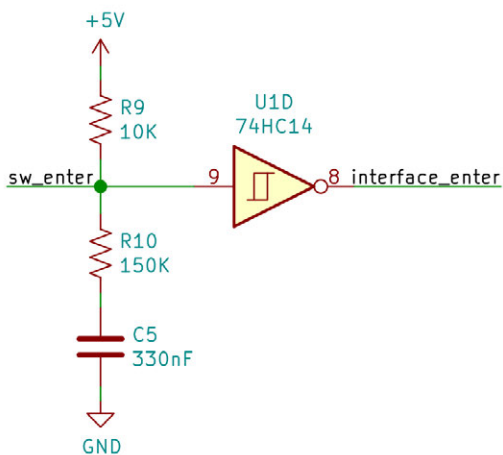


Figura 64: Circuito anti-rebotes

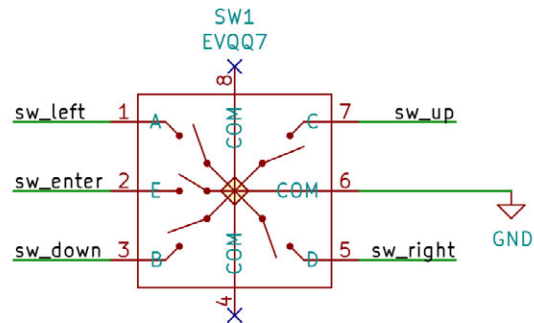


Figura 65: Conexión del joystick

La pantalla OLED está conectada al microcontrolador principal y al microcontrolador secundario a través del bus I2C, tal como se expone en el diagrama de bloques de la Figura 6. Esto permite al microcontrolador secundario utilizar la pantalla OLED para informar al usuario sobre el progreso, durante la programación remota del robot. El zumbador permite avisar al usuario de condiciones de operación que requieren atención inmediata, como que el nivel de la batería sea demasiado bajo.

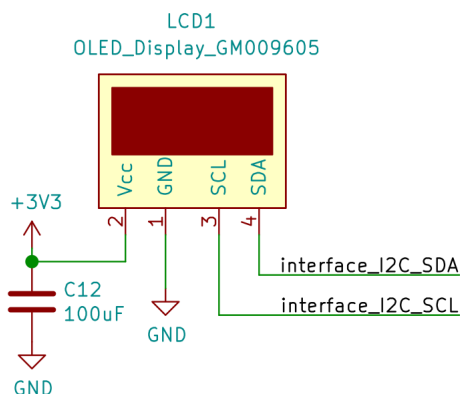


Figura 66: Conexión del módulo OLED

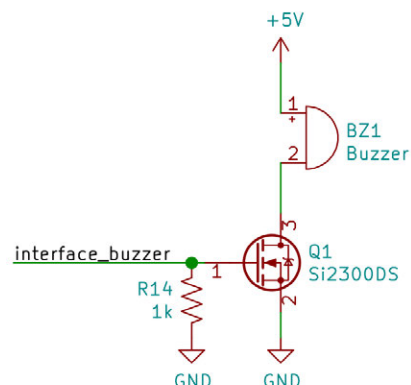


Figura 67: Conexión del zumbador

A parte de la interfaz de usuario, en esta placa se sitúan los dos circuitos integrados que forman la IMU de 9 ejes: acelerómetro, giroscopio y magnetómetro. En la versión 0 del diseño, la IMU estaba formada únicamente por un acelerómetro y un magnetómetro en un mismo encapsulado, pero durante las pruebas de la primera versión se comprobó que, pese a la calibración del magnetómetro, cualquier masa metálica grande producía un desviación significativa en el curso del robot. Por ese motivo, se incluyó un giroscopio.

El coste de incluir solo un giroscopio era mayor que el coste de incluir una segunda IMU que agrupara un acelerómetro y un giroscopio, por lo que se añadió una segunda IMU, dejando la primera IMU que contenía un acelerómetro y un magnetómetro solo como magnetómetro. Ambas IMU están conectadas al microcontrolador principal a través del bus I2C y dos líneas de interrupción, una para cada IMU.

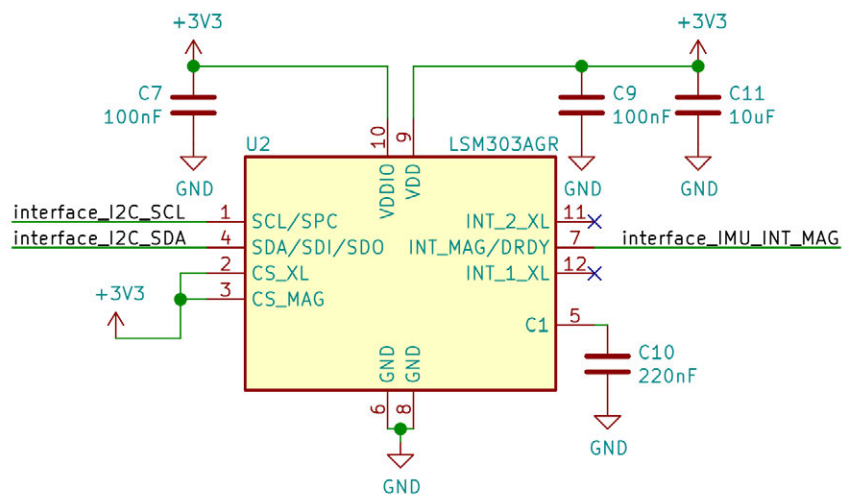


Figura 68: Conexión de la IMU 1 - magnetómetro

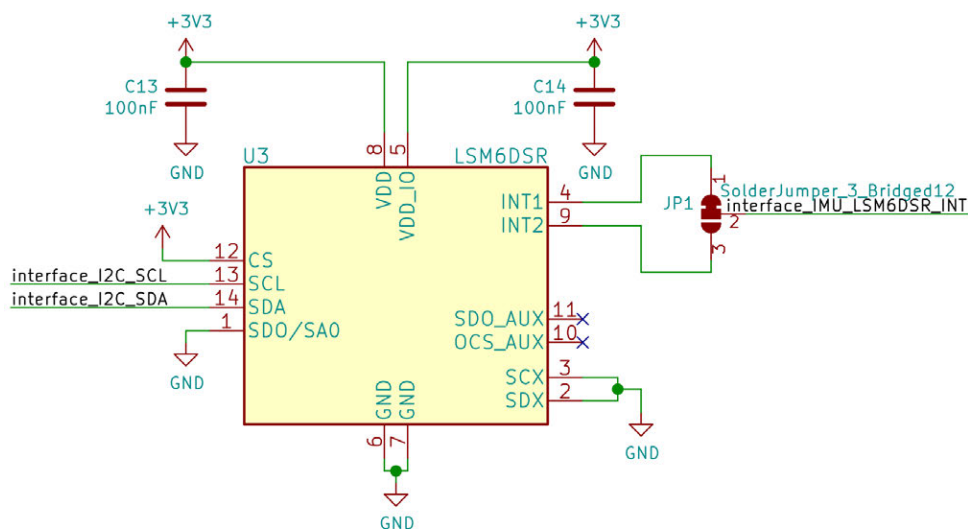


Figura 69: Conexión de la IMU 2 - acelerómetro y giroscopio

El diseño de la PCB de la interfaz de usuario se muestra a continuación:

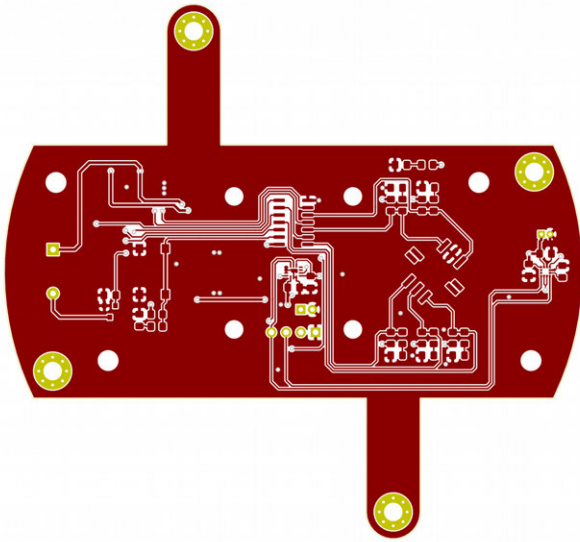


Imagen 37: Capa top de la user_interface_board

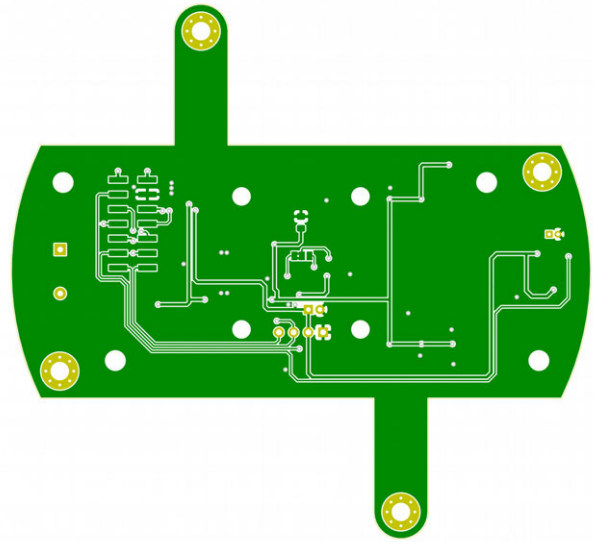


Imagen 38: Capa bottom de la user_interface_board

Adicionalmente, esta placa también sirve de soporte para la antena de 2.4GHz. Para acoplarla a la placa se ha diseñado una pieza que se sujeta a los mismos tornillos que aseguran el módulo OLED a la PCB. Esta pieza se encuentra archivada con el nombre: *soporte_antena_rev_A.FCStd*.

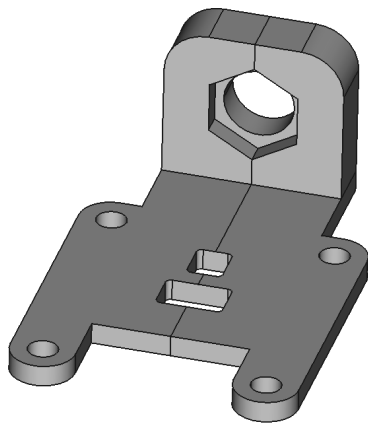


Imagen 39: Soporte de la antena

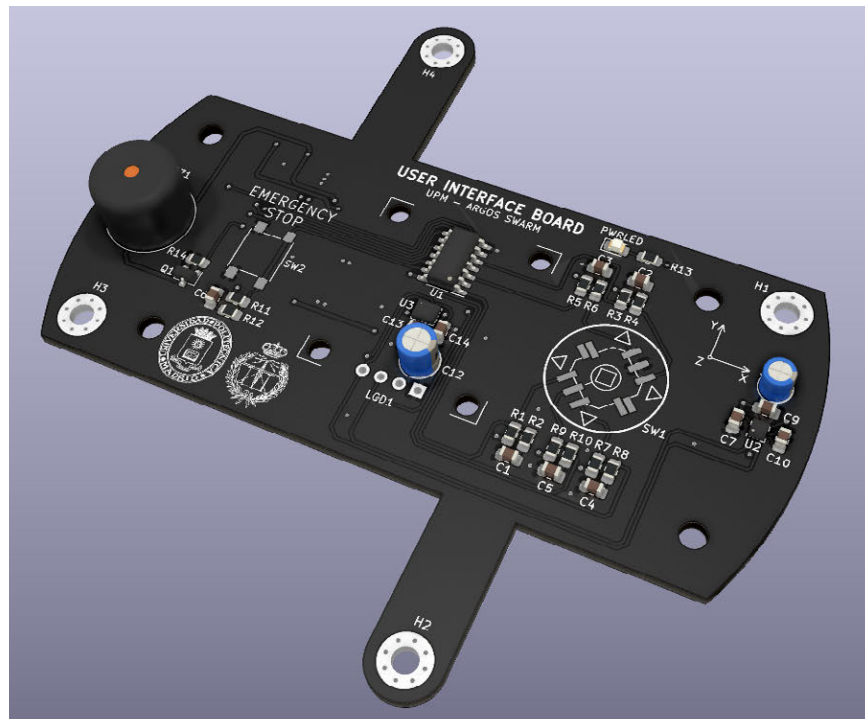


Imagen 40: Simulación 3D de la placa terminada, vista superior

4.1.7 Nivel 2 - Placa madre (motherboard)

Especificaciones que satisface: EH-4, EH-5, EH-10, EH-12.

La placa madre del robot es la placa donde están los microcontroladores que controlan todas las funciones del robot. A esta placa llegan todas las conexiones desde el resto de placas, y es en esta placa donde está la interfaz USB a Serie que permite programar los robots mediante una conexión por cable y acceder al puerto serie de cada microcontrolador.

4.1.7.1 Diagrama de bloques

El diagrama de bloques, presentado de forma preliminar en la Figura 6, se reproduce a continuación:

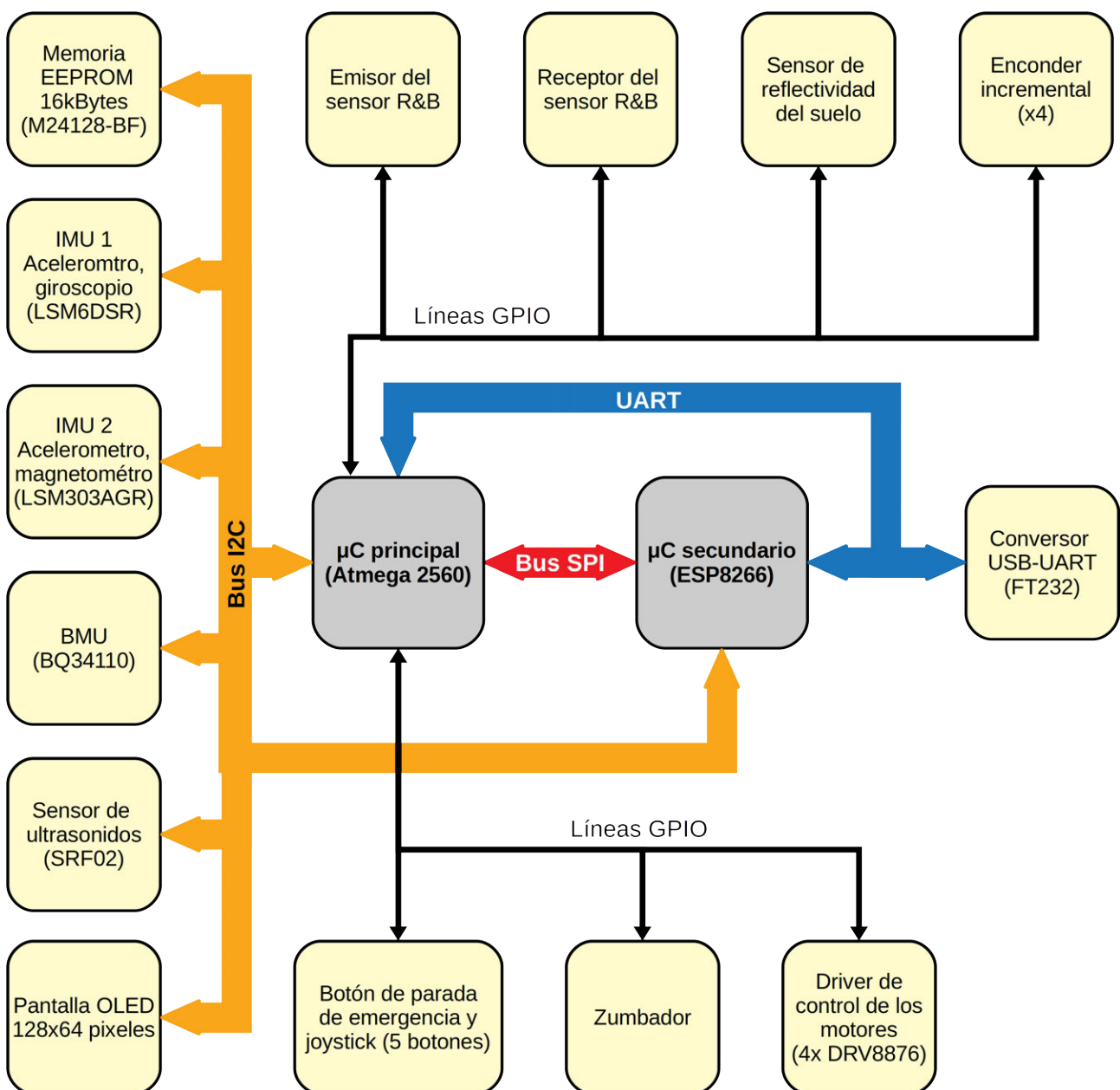


Figura 70: Diagrama de bloques de ARGOS

4.1.7.2 Distribución de buses

Tal como se observa en la Figura 70, los periféricos de ARGOS están conectados al microcontrolador principal a través de un bus I2C, y de algunas líneas de entrada y salida de propósito general (GPIO). A parte del bus I2C, la placa madre dispone de dos buses adicionales que interconectan el microcontrolador principal con el microcontrolador secundario: un bus SPI, y una interfaz UART.

Como ya se ha explicado en el apartado 4.1.6, el bus I2C está conectado al microcontrolador secundario, ESP8266, solo para permitir a este mostrar el progreso de la carga remota del programa en la pantalla de la interfaz de usuario. El bus I2C es un bus monomaestro y esta controlado por el microcontrolador principal, Atmega2560. Como se explicará a continuación, durante la carga remota de un programa, el microcontrolador principal esta controlador por el *bootloader*, por lo que no puede operar sobre el bus I2C, y por tanto, el ESP8266 asume el control del bus. Una vez la carga del programa ha finalizado, el ESP8266 deja de controlar el bus I2C.

A diferencia del bus I2C, el bus SPI se utiliza exclusivamente para permitir la intercomunicación del microcontrolador principal con el microcontrolador secundario. Esto permite al microcontrolador principal ejecutar procedimientos remotos en el microcontrolador secundario, lo que en última instancia, proporciona al Atmega2560 servicios de comunicaciones, por ejemplo. Dada la estructura del bus SPI, se requiere una línea de GPIO adicional para señalar que el ESP8266 solicita la lectura de datos.

La interfaz UART tiene varias funciones: en el entorno Arduino el puerto UART se utiliza habitualmente para interactuar con los programas (para depurarlos, por ejemplo), y por otro lado, también se utiliza como interfaz para cargar programas en el microcontrolador principal o en el microcontrolador secundario. A diferencia del bus I2C o el bus SPI, la interfaz UART es mono-maestro y mono-esclavo, lo que implica, que para interconectar los dos microcontroladores y la interfaz serie, como se muestra en la Figura 70, será necesario conmutar las líneas de la interfaz UART entre cada uno de los dispositivos.

4.1.7.3 Compatibilidad entre niveles lógicos

El Atmega2560 utiliza niveles lógicos TTL, mientras que el ESP8266 utiliza niveles lógicos LVTTTL. Tal como se recoge en la hoja de datos del ESP8266 [57], la máxima tensión que soporta cada entrada es de 3,6V, y dado que las salidas del Atmega2560, el cuál esta alimentado a 5V, a nivel alto producen una tensión de 5V, conectar ambos directamente ocasionaría daños al ESP8266.

Por esto motivo, en las líneas que comparte el ESP8266 y el Atmega2560, es necesario utilizar circuitos conversores de nivel. Algunas líneas, como la líneas de interrupción que permiten informar al ESP8266 de que tiene el Atemga2560 tiene datos que procesar o viceversa, son líneas unidireccionales, y dado que su frecuencia de operación es baja, la lógica de conversión de nivel puede implementarse con componentes discretos, esto es, resistencias y transistores.

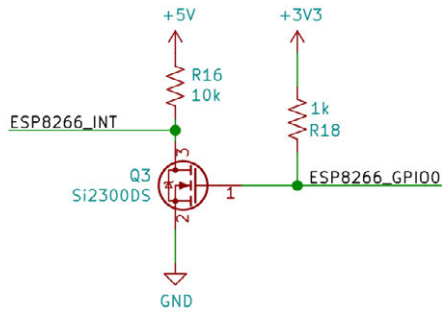


Figura 71: Conversor de nivel discreto

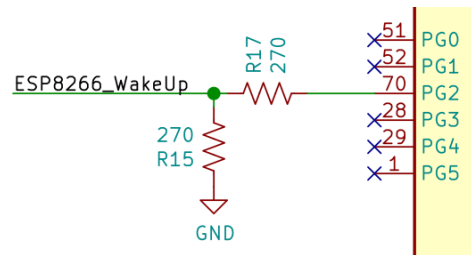


Figura 72: Conversor de nivel discreto

Sin embargo, otras interconexiones, como las líneas del bus SPI o las líneas de la interfaz UART, tienen requisitos de frecuencia de operación más estrictos, y su implementación con componentes discretos ocuparía un área de la PCB considerable, por lo que para estas, se utilizó un conversor de nivel integrado:

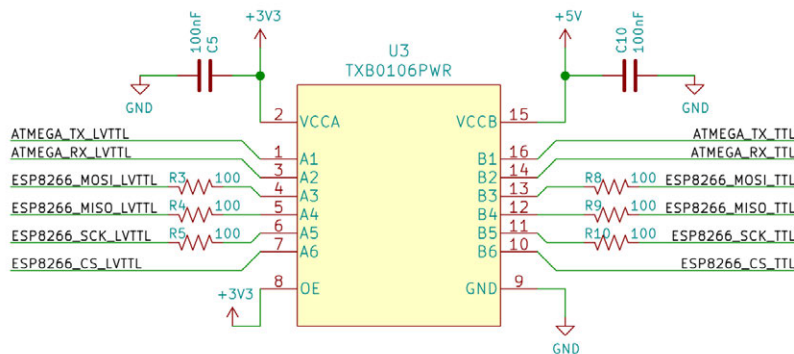


Figura 73: Conversor de nivel para las líneas del SPI y UART

Por último, las resistencias de *pull-up* del bus I2C están conectadas a 3,3V, y como las líneas del bus operan en *open-drain* y los niveles lógicos del Atmega2560 son compatibles con los niveles lógicos LVTTL, no es necesario implementar ningún tipo de conversor de nivel.

4.1.7.4 Conmutadores de la interfaz UART

La interfaz UART es compartida por el Atmega2560 (microcontrolador principal), el ESP8266 (microcontrolador secundario) y el por la interfaz USB a Serie, basada en un integrado FTDI232. Además de las líneas de RX y TX de la interfaz UART, los microcontroladores comparten con el chip de interfaz USB a Serie las líneas de RTS y DTS, utilizadas en el protocolo RS232; las cuales permiten activar el modo de programación del microcontrolador principal y del microcontrolador secundario.

Para establecer la forma en la que se interconectan los puertos UART de los microcontroladores y de la interfaz USB a Serie, la placa dispone de un conmutador de tres posiciones, a saber:

- **ESP8266_prog**: conecta el puerto UART del ESP8266 a la interfaz USB a Serie, lo que permite actualizar el firmware del ESP8266 a través de esta, y acceder al puerto UART del mismo.

Descripción de la solución propuesta

- **Atmega_prog:** conecta el puerto UART del Atmega2560 a la interfaz USB a Serie, lo que permite cargar programas a través de esta, y comunicarse con puerto UART del microcontrolador principal.
- **Remote_prog:** desconecta las líneas UART de la interfaz USB a Serie, e interconecta las líneas UART entre el Atmega2560 y el ESP8266. Esto permite la programación remota del microcontrolador principal utilizando el microcontrolador secundario, y además, permite acceder al puerto serie del microcontrolador principal utilizando el servidor Telnet del ESP8266.

Para realizar estas funciones, las líneas UART del ESP8266 y el Atmega2560 están conectados mediante un interruptor digital. Así mismo, las líneas UART, RTS y DTS del FTDI232 están conectadas a través de un multiplexor de dos posiciones con las líneas de cada uno de los microcontroladores. Tanto el interruptor digital como el multiplexor se habilitan o deshabilitan según el modo seleccionado con el conmutador de tres posiciones.

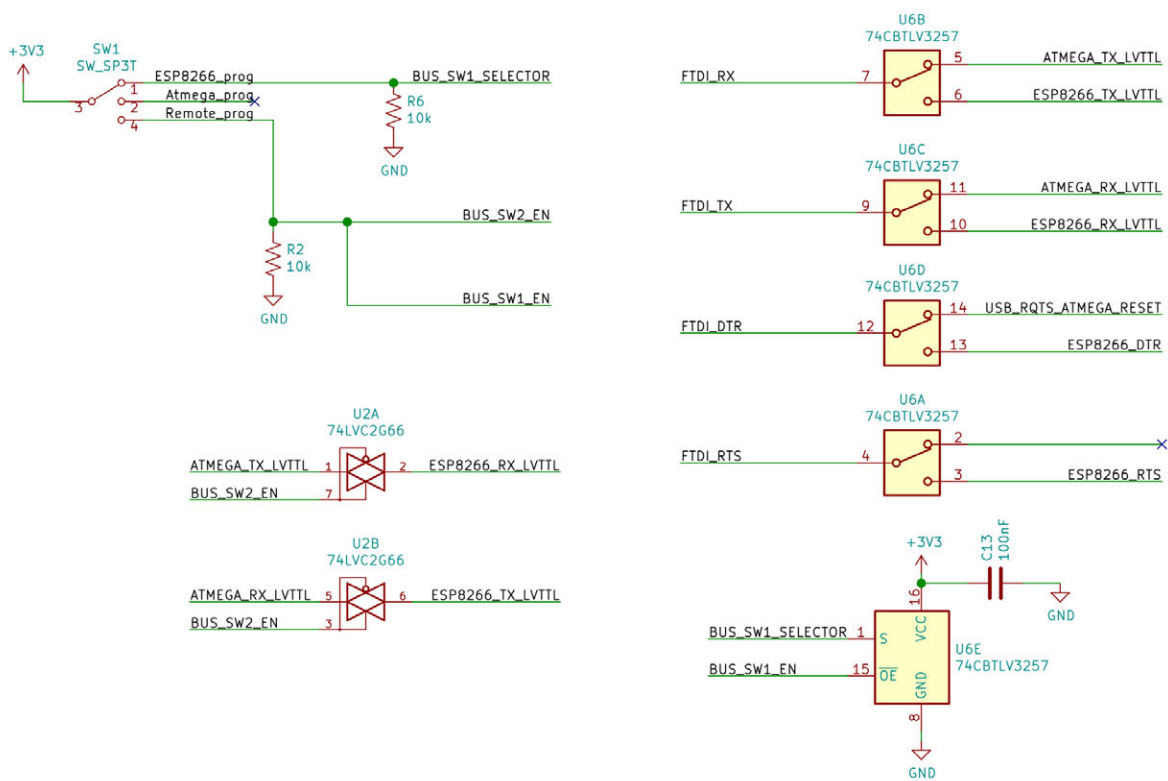


Figura 74: Selector de modo y conmutación de la interfaz UART

4.1.7.5 Circuito de programación del Atmega2560

El Atmega2560 se puede programar a través de varias interfaces: JTAG, SPI, y UART (si tiene un *bootloader* cargado en la memoria). La placa base dispone de todas ellas, aunque habitualmente solo se usan dos: la interfaz SPI, para cargar el *bootloader* en el microcontrolador, y la interfaz UART, para cargar programas al microcontrolador de forma remota o mediante cable, a través el puerto USB.

El *bootloader* es un programa que reside en una zona de la memoria flash del microcontrolador. Cuando el Atmega2560 recibe por primera vez alimentación, o tras un pulso en la línea de RESET,

el código que primero se ejecuta es el del *bootloader*. Entonces, el *bootloader*, comprueba la interfaz UART durante un cierto tiempo, y si no recibe un comando específica que indica que se va a cargar un programa, resume la operación con el último programa almacenado en la memoria flash. De lo contrario, comienza la carga del código máquina programa en la memoria flash, y posteriormente, lo ejecuta.

Este pulso en la línea de RESET debe ser generado por la interfaz USB-serie o por el ESP8266. Adicionalmente, la placa base dispone de un pulsador para reiniciar el microcontrolador de forma manual. El circuito que se encarga de realizar esta función, es el circuito de auto-reset:

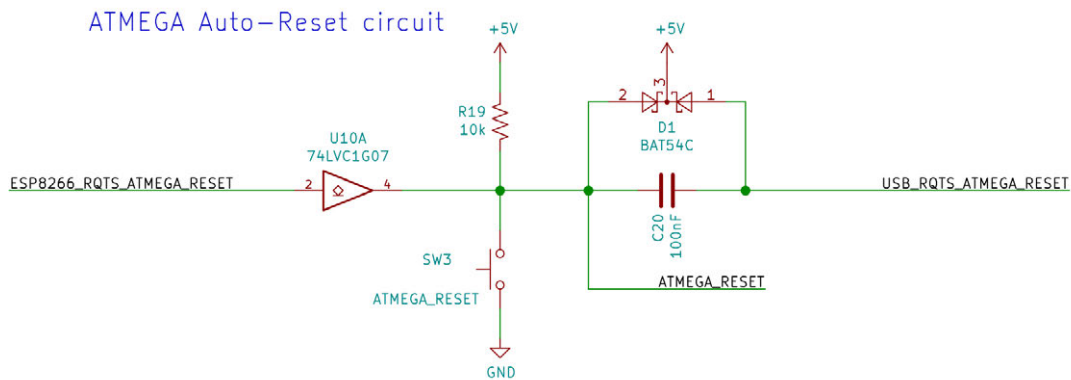


Figura 75: Circuito de auto-reset del microcontrolador principal - Atmega2560

4.1.7.6 Circuito de programación del ESP8266

De forma análoga, para comenzar el proceso de programación del ESP8266, este debe resetearse. Tras el *reset*, dependiendo del estado en que se encuentren los pines *GPIO2*, *GPIO15* y *GPIO0*, el ESP8266 entrará en diversos modos de arranque [58]. Para la operación entre el modo de ejecución normal, y el modo de carga de programas usando el *bootloader*, *GPIO2* debe estar a nivel alto y *GPIO15* a nivel bajo. Entonces, cuando *GPIO0* este a nivel alto, el microcontrolador ejecutará el programa almacenado en la ROM, y cuando *GPIO0* este a nivel bajo, el microcontrolador entrará en el modo de carga de programas. Para activar la señal de *reset* y establecer el valor de *GPIO0* adecuado a partir de las líneas de RTS y DTS de la interfaz USB a serie, se usa el siguiente circuito:

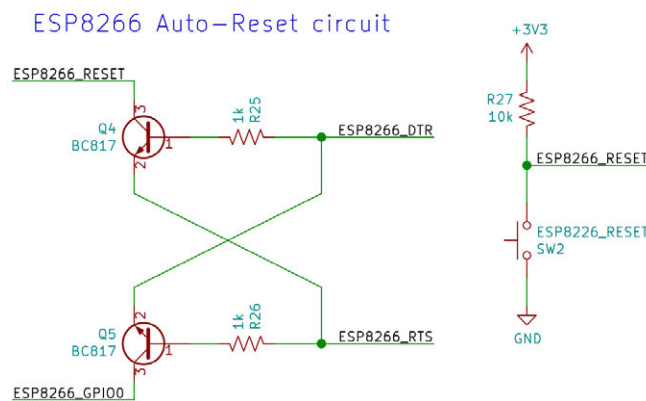


Figura 76: Circuito de auto-reset del microcontrolador secundario - ESP8266

4.1.7.7 Conexiones con los sensores

A esta placa llegan todas las conexiones con el resto de placas de ARGOS, y además, en esta placa se sitúan las conexiones con los *encoders* de los motores, el sensor de reflectividad, el sensor de ultrasonidos, y una conexión auxiliar al bus I2C que permite aumentar la dotación de sensores. Para cumplir con la especificación EH-11, la alimentación del sensor de reflectividad y de los *encoders* puede ser desactivada.

4.1.7.8 Fuente de alimentación

Los reguladores de tensión de la placa base alimentan dos placas adicionales: la placa de interfaz de usuario, y la placa de potencia. Dado que el microcontrolador principal, Atmega2560, requiere +5V, y el microcontrolador auxiliar, ESP8266, requiere +3,3V, la placa base dispone de de dos reguladores lineales para generar esas tensiones.

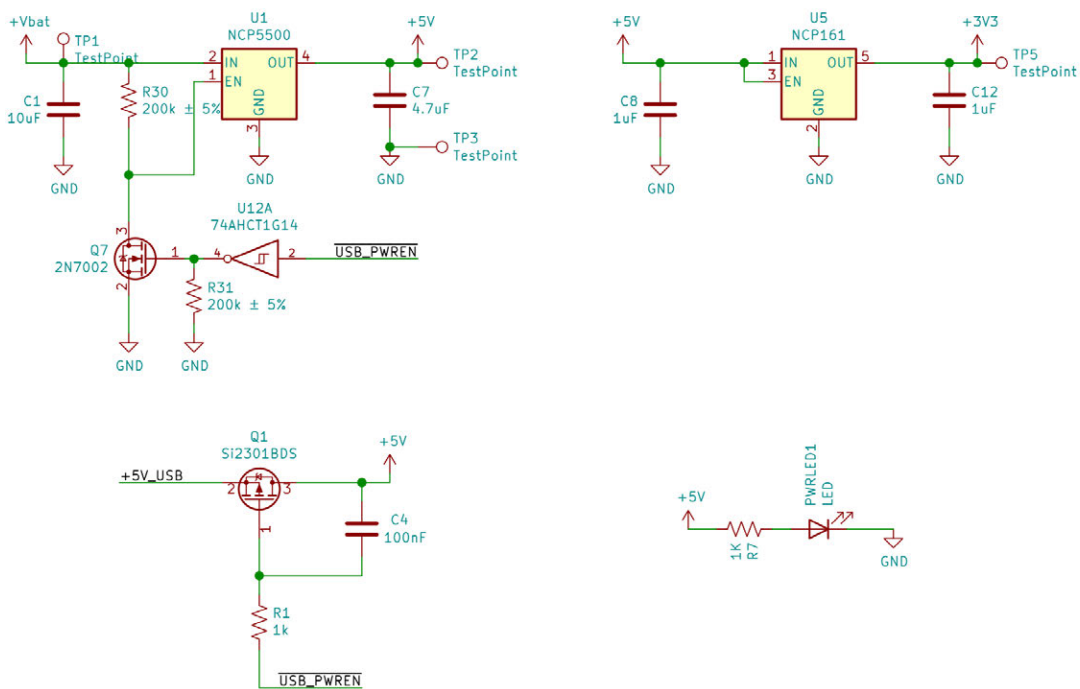


Figura 77: Alimentación de la placa madre

El regulador de tensión de 3,3V, tiene una tensión de entrada máxima de 5,5V, debido a esto, su entrada está conectada al regulador de 5V. Además, la arquitectura de alimentación se diseñó de modo que el regulador de +5V se desconectará si la placa está recibiendo alimentación a través del USB, lo que el FTDI232 indica mediante la línea *USB_PWREN*. Esto permite utilizar de forma segura el robot cuando está conectado a un ordenador mediante USB, y además, el interruptor de alimentación de la batería está conectado.

Además de alimentar la circuitería de la placa madre, los reguladores de tensión alimentan también la placa de potencia y la placa de la interfaz de usuario.

A continuación, se exponen los consumos pico de corriente de los raíles de alimentación para cada placa:

Raíl +3,3V	
Módulo	Consumo de corriente pico (mA)
Placa de Interfaz de usuario	33,5
Placa madre (solo)	187,5

Total	221mA
--------------	-------

Tabla 6: Consumo pico raíl +3,3V

Raíl +5V	
Módulo	Consumo de corriente pico (mA)
Placa de Interfaz de usuario	27
Placa de potencia	8
Placa madre (solo +5V)	77

Total	112mA
--------------	-------

Tabla 7: Consumo pico raíl +5V

Dado que en última instancia, toda la corriente se extrae del raíl de alimentación de la batería, V_{BAT} , teniendo en cuenta los consumos calculados para placa del emisor del *sensor de rango y rumbo* (72), y para la placa del receptor del *sensor de rango y rumbo* (112), el consumo de corriente pico en el raíl de alimentación de la batería es:

Raíl V_{BAT}	
Módulo	Consumo de corriente pico (A)
Placa madre (combinado)	0,333
Placa de potencia	10,5
Placa del emisor del sensor R&B	0,1205
Placa del receptor del sensor R&B	0,85

Total	11,8A
--------------	-------

Tabla 8: Consumo total pico del raíl V_{BAT}

4.1.7.9 Diseño de la placa

Para el diseño de esta placa se han tenido los mismos principios de diseño que para las placas del sensor de rango y rumbo, a saber: plano de masa, pistas de señal analógicas separadas de pistas de alimentación o reloj que pudieran causar interferencias por acoplamiento, etc.

Los resultados se muestran a continuación:

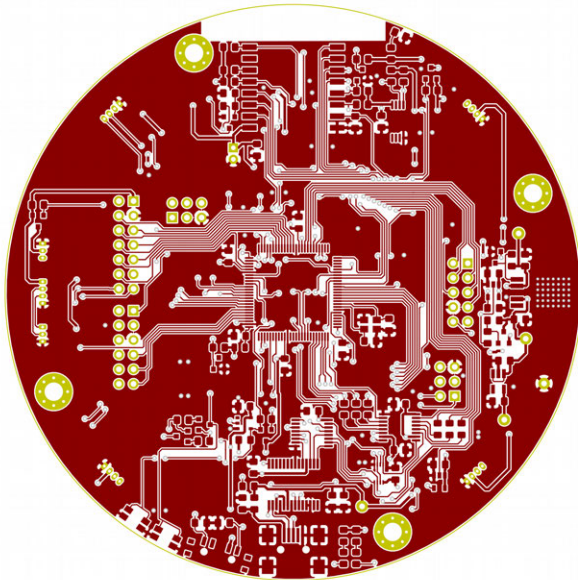


Imagen 41: Capa top de la motherboard

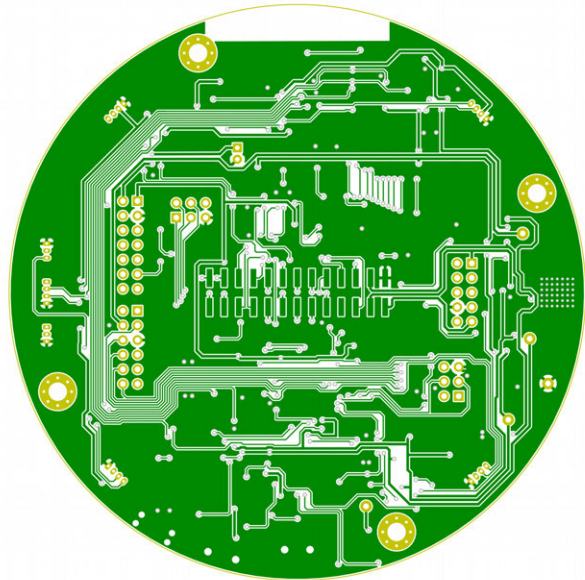


Imagen 42: Capa bottom de la motherboard

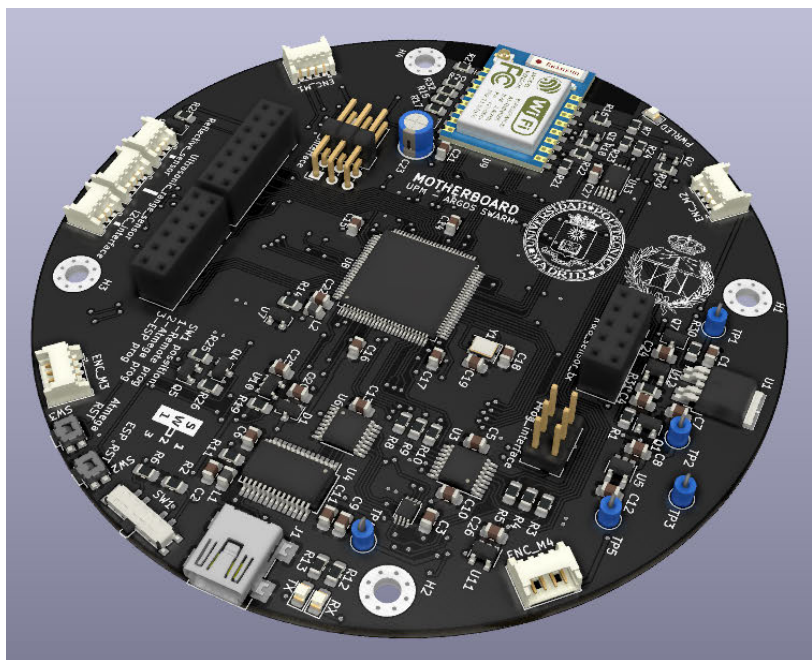


Imagen 43: Simulación 3D de la placa terminada, vista superior

4.2 Software

El software de este proyecto utiliza el ecosistema de prototipado rápido Arduino como base, y tal como se recoge en los requisitos *RNF-1* y *RNF-3*. El ecosistema de Arduino, proporciona un entorno de desarrollo, es decir, una aplicación que incluye las herramientas de programación y compilación para un conjunto de placas de desarrollo. Adicionalmente, también incluye una HAL, esto es, una capa de abstracción de hardware, que oculta los detalles de implementación de las bibliotecas que controlan los diferentes periféricos de los microcontroladores de las placas de desarrollo.

Esto, junto con el gran número de bibliotecas que la comunidad ha desarrollado para este ecosistema, permite que un usuario con poca experiencia pueda desarrollar aplicaciones complejas y hasta cierto punto, portables.

El software de este proyecto, se basa en esa filosofía. Así una de las partes de este proyecto consiste en la implementación de un BSP (*Board Support Package*), es decir, el conjunto de bibliotecas de bajo nivel que den soporte a la placa en el IDE de Arduino, y algunas herramientas que amplíen el *toolchain* disponible en Arduino, en concreto, el sistema de programación remota o OAP (*Over Air Programming*).

Para una añadir una nueva placa de desarrollo al entorno de Arduino, se requiere reimplementar un conjunto de bibliotecas que constituyen la HAL mínima, así como proporcionar el conjunto de herramientas, compilador, enlazador, etc, que forman el BSP de la placa. El microcontrolador principal elegido, Atmega2560, así como es microcontrolador secundario, ESP8266, ya cuenta cuentan con soporte dentro del IDE de Arduino, por lo que solo fue necesario adaptar el BSP existente para extender su funcionalidad.

La plataforma de Arduino está pensada para servir como herramienta de prototipado, y está orientada a la comunidad de aficionados, por lo que no cuenta con muchas características habituales en los entornos de desarrollo para sistemas embebidos modernos. Una de sus principales limitaciones es que no tiene ningún tipo de soporte para RTOS, esto es, sistemas operativos en tiempo real. Esto, junto con la forma en la que esta implementado la HAL, implica que para utilizar un RTOS debe adaptarse parte de la HAL.

En el apartado de hardware de este capítulo, se ha presentado toda la dotación de sensores y actuadores que llevan los robots físicos. Algunos de ellos, como el *sensor de rango y rumbo*, tienen requisitos de tiempo de ejecución que no pueden cumplirse si no se utiliza un sistema operativo que soporte la multitarea apropiativa.

Además de esta limitación, las aplicaciones que serán usadas en la evaluación de los algoritmos de robótica de enjambre deben poder ejecutarse sin realizar ninguna modificación, tanto en la plataforma real, como en la plataforma simulada.

4.2.1 Arquitectura de la API

En la siguiente figura expone la arquitectura de la API que se ha implementado para dar soporte al hardware en Arduino:

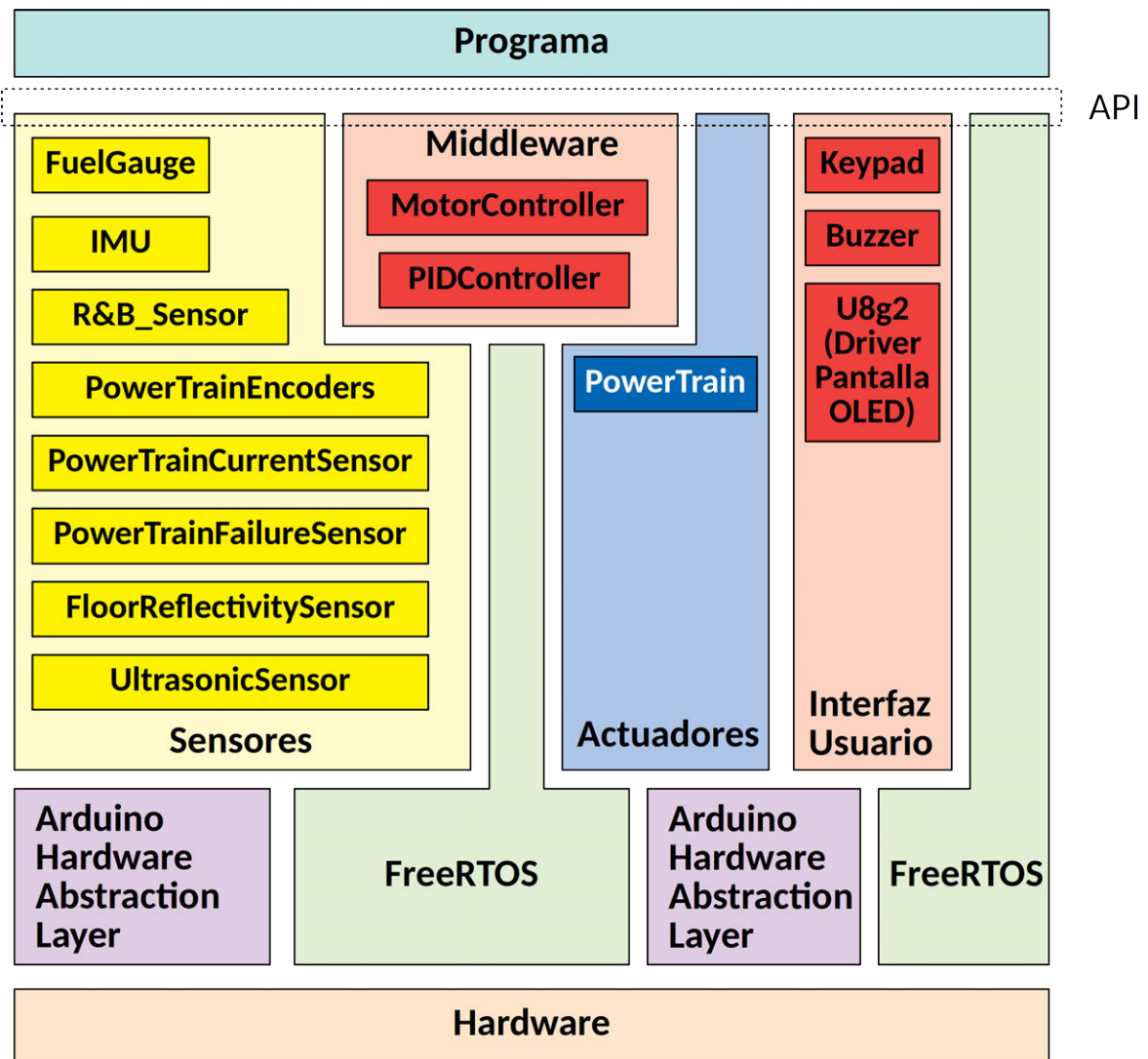


Figura 78: Arquitectura de la API

Empezando por la parte inferior de la pila, sobre el hardware se sitúan dos capas: *Arduino HAL* y *FreeRTOS*. La capa *Arduino HAL* utiliza las bibliotecas proporcionadas por Arduino para abstraer los detalles de implementación de más bajo nivel del hardware. La capa *FreeRTOS* proporciona los servicios de un sistema operativo en tiempo real, algunos de estos servicios son: ejecución de *callbacks* de forma periódica, ejecución concurrente de tareas, etc.

En un sistema embebido *baremetal*, esto es, un sistema que no utiliza un RTOS para funcionar, no se pueden producir problemas de concurrencia porque solo hay un hilo de ejecución. En un sistema basado en RTOS, como en este caso, se presentan problemas de concurrencia debido a que varias tareas se ejecutan en paralelo y compiten por el acceso a los diferentes recursos de hardware. Para solucionar este problema, se utilizan *mutex* para cada periférico. Así, si por ejemplo si el módulo *UltrasonicSensor* está utilizando el I2C para realizar una medida tomará el *mutex*

asociado al puerto I2C, de forma que si su ejecución se interrumpe y otra tarea desea acceder al puerto I2C, no podrá hacerlo porque el *mutex* del I2C esta cogido por otra tarea.

Los módulos están dividido en función del tipo de tarea que realizan. Así, hay tres capas que contienen los módulos asociados a las siguientes funciones: sensores, actuadores e interfaz de usuario. Adicionalmente, los módulos que implementan servicios que no dependen directamente del hardware están contenidos en la capa *middleware*.

La capa *middleware* contiene el control de velocidad de los motores, *motorController*, y el *PIDController*, que implementa un controlador PID, que es utilizado por el control de velocidad de los motores, y que puede ser instanciado para otras funciones.

La capa de interfaz de usuario contiene los recursos asociados a la placa de interfaz de usuario, excluyendo lo relacionado con la IMU.

La IMU, y el resto de sensores: *FuelGage* (medición del estado de la batería), *R&B_sensor* (el módulo que controla el hardware del *sensor de rango y rumbo*), el *UltrasonicSensor* que controla el sensor de ultrasonidos, el *FloorReflectivitySensor*, que controla el sensor de reflectividad del suelo, junto con el resto de sensores del tren de potencia van en la capa de sensores.

Por último, los actuadores, esto es, el módulo, *PowerTrain*, que controla los *drivers* del de los motores está situado en la capa de actuadores.

Los ficheros de estos archivos están la biblioteca *Argos_API*, localizada en la ruta *Software/Argos_API*, de los archivos que acompañan a este proyecto. Además de estos, la biblioteca también tiene un conjunto de ejemplos que demuestran algunas capacidades de la plataforma.

4.2.2 Integración de FreeRTOS con Arduino y e implementación del core

Para la integración de FreeRTOS con Arduino, primero se requirió encontrar un *port* para el microcontrolador Atmega2560, ya que FreeRTOS no lo soporta de forma oficial. Para ello, se siguió la documentación de FreeRTOS [59], y se utilizó el *port* contenido en el repositorio [60].

A continuación para dar soporte al hardware de Argos, se partió del *core* o placa de Arduino, del Atmega2560. Para integrar FreeRTOS con Arduino, se ha modificado el archivo *main.cpp*, el cual contiene las llamadas a los métodos *setup()* y *loop()* que la base de los *sketchs* de Arduino. Las llamadas a estas funciones se han incorporado dentro de una tarea de FreeRTOS. De esta forma, el *scheduler* de FreeRTOS se inicia en el punto de entrada del programa, y a continuación, lanza las llamadas a los métodos *setup()* y *loop()* dentro de una tarea. Estas modificaciones pueden observarse en la Imagen 44.

A parte de estas modificaciones, también fue necesario configurar uno de los *timers* del Atmega2560 para generar el *tick* que requiere FreeRTOS.

Adicionalmente, se modificaron las definiciones de los pines de Arduino, contenidas en el archivo *pins_arduino.h*, para dar soporte a los pines adicionales (puerto J del microcontrolador) que utiliza

la placa base de Argos. Además, se modificaron los archivos de configuración: *boards.txt*, *platform.txt* y *programmers.txt* para integrar el sistema de programación remota desarrollado con el IDE de Arduino.

```
int main( void )
{
    //Stack and TCB for the ArduinoSketch task
    static StaticTask_t xArduinoSketchTaskTCB;
    static StackType_t uxArduinoSketchTaskStack[ configMINIMAL_STACK_SIZE ];

    /*Arduino init Timers*/
    init();

    //Arduino's sketch task:
    xTaskCreateStatic( ArduinoSketch, "sket", configMINIMAL_STACK_SIZE, \NULL, 1

    vTaskStartScheduler();

    //It won't reach here
    return 0;
}

//////////////////////////////////// Arduino's sketch task //////////////////////////////////////
void ArduinoSketch(void* param)
{
    //Setup
    setup();

    for(;;)
    {
        //Loop
        loop();

        if (serialEventRun) serialEventRun();
    }
}
```

Imagen 44: Modificaciones del archivo *main.cpp*

La nueva placa se encuentra en la ruta *Software/Arduino Board*, en los ficheros que acompañan a este proyecto. Para utilizarla, basta con copiarla en el directorio *hardware*, de la instalación de Arduino. Los pasos para que la programación remota funcione se detallan en el siguiente apartado.

4.2.3 Firmware del ESP8266 y sistema de programación remota

El microcontrolador secundario, ESP8266, proporciona varios servicios al microcontrolador principal: programación remota, un servidor de Telnet, y un enlace de comunicaciones entre robots basado en ESP-NOW.

El sistema de programación remota permite actualizar el programa del microcontrolador principal de todos los robots del enjambre de forma inalámbrica. Para ello, el microcontrolador secundario, ESP8266, actúa como programador, cargando el código máquina recibido a través de la conexión WiFi.

Para que este sistema funcione, en primer lugar el usuario debe modificar el archivo *WifiCredentials.h* contenido en el fichero del firmware que hay en ruta *Software/ESP8266_firmware*, y añadir las credenciales de acceso a la red Wifi donde se encuentra el ordenador que ejecuta Arduino. A continuación, utilizando el IDE de Arduino, debe cargar el firmware con los credenciales de la red modificados a la placa del robot, utilizando la placa *Generic ESP8266 Module* de Arduino, y seleccionando *Prog_ESP8266* en el conmutador de la placa madre.

Una vez el ESP8266 se haya conectado a la red Wifi, se suscribirá a un grupo *multicast*, en concreto, al grupo 239.0.0.42/8. Cuando el usuario inicia la carga remota de un programa, el toolchain de Arduino compila el programa, y a continuación, ejecuta el script de Python *OAP_remote_prog.py*. Este script es el encargado de desplegar el servidor de carga remota, y de localizar a todos los integrantes del enjambre. Para esto último, el servidor envía un mensaje que contiene dos datos: un identificador, y la dirección IP del servidor, al grupo *multicast* 239.0.0.42/8. Tras esto, el servidor queda a la escucha de conexiones TCP entrantes en el puerto 50042 durante 20 segundos.

Los ESP8266 que están conectados a la red Wifi, reciben el mensaje a través del grupo *multicast*. A continuación, los ESP8266 se conectan a la dirección IP especificada en el mensaje utilizando un *socket* TCP. Entonces para cada conexión TCP abierta, el servidor inicia la transmisión de los fragmentos del código máquina que deben cargarse en el microcontrolador principal de cada robot. Una vez se ha completado la carga en todas las conexiones abiertas, el servidor finaliza. El fichero que contiene el script del servidor, *OAP_remote_prog.py*, está almacenado en el *core* de la placa desarrollada, y además se puede encontrar en la ruta *Software/Over Air Programming*, en los archivos que acompañan a este proyecto.

Respecto del sistema de programación del ESP8266, este utiliza el protocolo STK500v2 [62], el cual permite ordenarle al *bootloader* que reside en la memoria del micro controlador principal, que borre la memoria y comience a almacenar los datos por orden de llegada. Tanto el comportamiento del servidor como del cliente (ESP8266) están controlados por una máquina de estados.

A parte de estas funciones, el ESP8266 también implementa un servidor Telnet, el cual establece una conexión entre el terminal remoto y el puerto UART del ESP8266. Esto, permite conectarse de forma remota con el puerto UART del microcontrolador principal, pudiendo usar este enlace para extraer datos o telemetría, durante la operación de los robots.

5. Resultados

5.1 Resultados del ensamblado

Los cuatro robots ensamblados se muestran en la siguiente imagen:

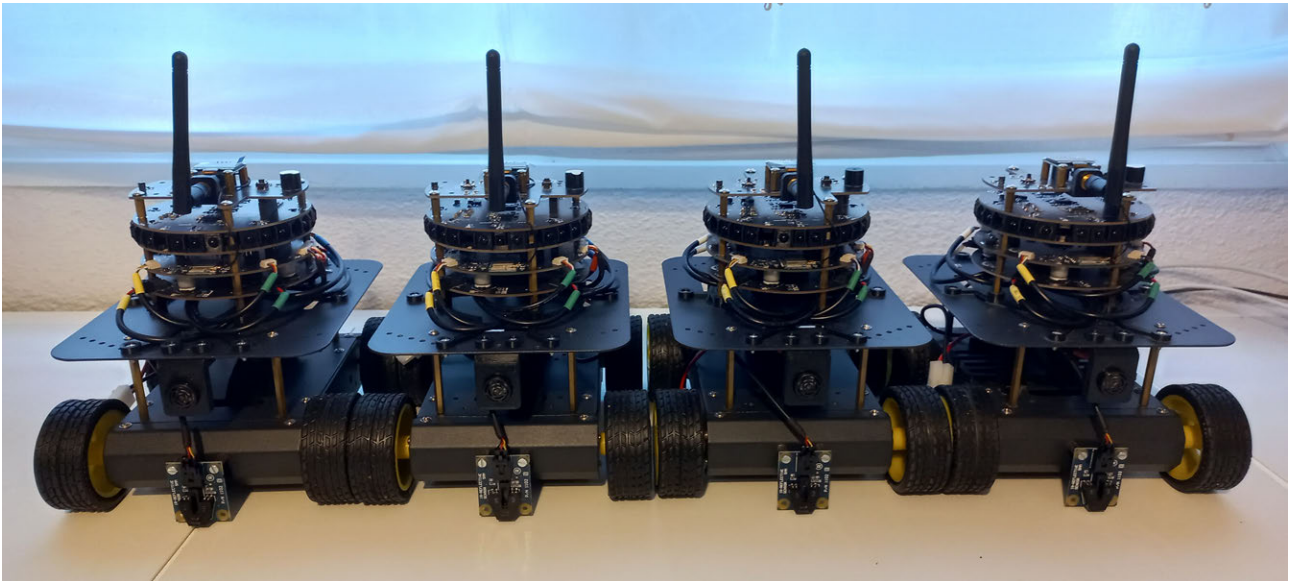


Imagen 45: Enjambre de robots ARGOS

Dado que durante la fase de diseño de las PCB y de las placas 3D se realizaron ensamblados virtuales, los errores de diseño de las piezas 3D fueron corregidos antes de llegar a la fase de producción, por tanto, no se encontraron problemas durante el montaje.

Los archivos correspondientes al diseño del hardware están contenidos en la ruta *Hardware/SWR-UPM* de los archivos de este proyecto. En la carpeta *3d_printed_parts* se encuentran todas las 3d piezas diseñadas, y en la carpeta *Boards*, los archivos de *KiCad* de todas las placas diseñadas.

Respecto al diseño del hardware, los resultados de las pruebas presentadas a continuación corresponden al diseño de la Revisión A, es decir, a la segunda iteración sobre el hardware de los robots. En esta memoria no se han presentado de forma detallada las diferencias entre la versión inicial (Versión 0), y la Revisión A porque esta información ocuparía una extensión considerable, y no es de demasiada utilidad.

No obstante, las correcciones y pruebas sobre la versión 0 ocuparon una considerable cantidad de tiempo, principalmente debido a los problemas encontrados en el diseño del *sensor de rango y rumbo*. A continuación, en la Imagen 46 y la Imagen 47, se muestran las placas originales de la Versión 0 con las PCBs auxiliares que se diseñaron para solucionar los problemas, y garantizar que el hardware de la Revisión A estaba libre de errores.

Las PCBs auxiliares se encuentran archivadas en la ruta *Hardware/SWR-UPM/Boards/rev_A_hot_fix_boards*, en los archivos adjuntos a este proyecto. Algunos resultados de las pruebas, se encuentran recogidos en la carpeta *Hardware/SWR-UPM/Pruebas/Versión_0*.

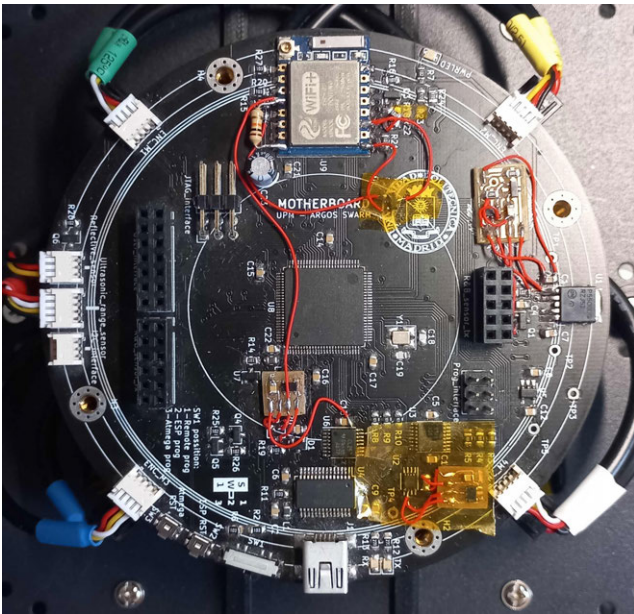


Imagen 46: Correcciones de la placa madre - versión 0

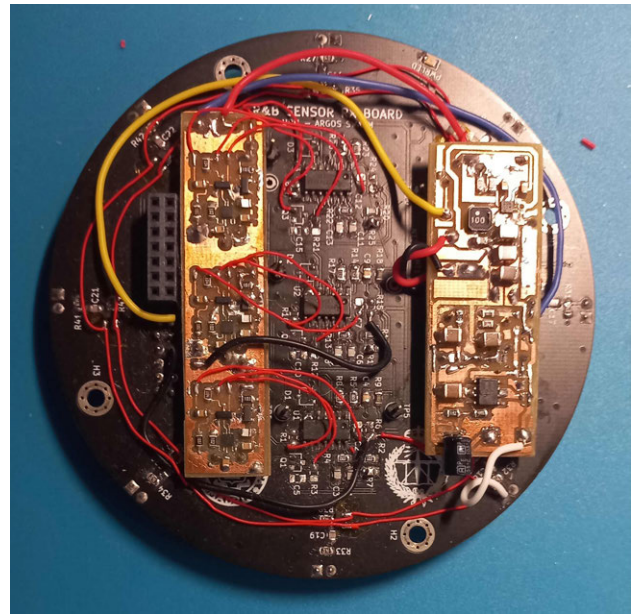


Imagen 47: Correcciones de la placa receptor del sensor de rango y rumbo - versión 0

5.2 Pruebas sobre el hardware

La placa de potencia, la placa madre, y las placas del *sensor de rango y rumbo*, cuenta con puntos de test, que permiten soldar un *header* para obtener diferentes mediciones. Utilizando dichos puntos de test se comprobó en primer lugar, que todas las tensiones de las fuentes de alimentación estaban dentro de los rangos esperables, para las peores condiciones de consumo, no encontrándose desviaciones respecto de los valores previstos.

En segundo lugar, utilizando como referencia los valores de tolerancia máxima para las diferentes ganancias de los bloques que forman la placa emisora y la placa receptora del *sensor de rango y rumbo* presentados en 4.1.4, 4.1.5, se comprobó que todas las ganancias estaban dentro de las tolerancias esperadas, teniendo en cuenta además el error de medida que introduce el polímetro.

En este aspecto, la única desviación más reseñable es el ruido pico medido para la salida de la sección 3 de la placa de recepción, cuando no hay ninguna señal de entrada y el fotodiodo (PD1) permanece cubierto. Tomando el valor del LSB del ADC, y convirtiendo los valores obtenidos del ADC a tensiones, para 1000 muestras, se obtuvo:

Valor medio (V)	0,003818
Desviación típica (mV)	2,190
Valor pico (mV)	12

Tabla 9: Desviación del ruido pico esperado

El valor pico máximo esperado para el ruido, según (118), es de 9,95mV. La diferencia entre el valor esperado, y el valor medido puede deberse a que no se han tenido en cuenta los efectos de la no linealidad del ADC del Atmega2560. En cualquier caso, observando la desviación típica y la media, el efecto de esta desviación se puede considerarse menor.

Además de estas pruebas, para verificar los *drivers* desarrollados para los diferentes periféricos que incorpora el robot eran correctos, y que el funcionamiento de estos periféricos era el esperado, se desarrollaron un conjunto de test, que pueden encontrarse en la carpeta *Software/tests_periféricos*, en los archivos adjuntos a este proyecto. Utilizando estos test, se verificó la funcionalidad de todos los sensores, actuadores, y componentes de la interfaz de usuario, no encontrándose errores.

5.3 Pruebas del sensor de rango y rumbo

Para probar el *sensor de rango y rumbo*, se utilizó un enfoque cualitativo, simulando las condiciones de operación más cercanas a su uso habitual. Para ello, se utilizan dos robots: el primero de ellos, el robot A, está configurado en modo baliza, esto es, utiliza el emisor del *sensor de rango y rumbo* para emitir una señal ininterrumpida, mientras realiza una trayectoria circular. El segundo robot, el robot B, implementa un sistema de control que varía su velocidad lineal y su velocidad angular en función de la distancia y del ángulo que el robot B forma con el robot A.

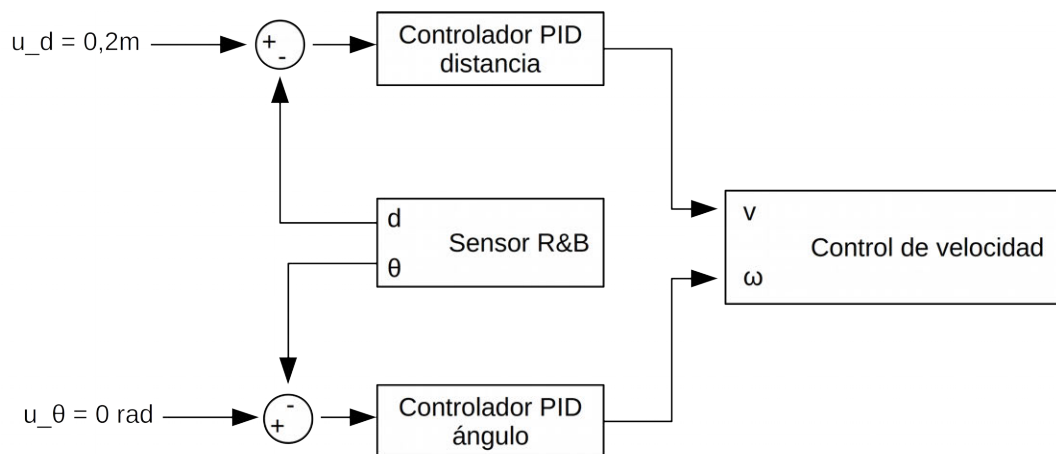


Figura 79: Sistema de control del robot B

De esta forma, el robot B sigue al robot A, adaptando su trayectoria para mantener una distancia constante con este. En las siguientes imágenes se presenta la secuencia de aproximación y seguimiento:

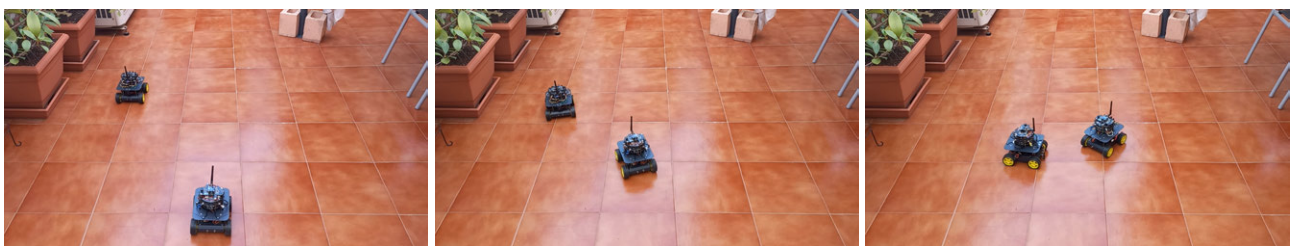


Imagen 48: Acercamiento

Imagen 49: Adaptación

Imagen 50: Seguimiento

El video completo y el programa de la prueba, pueden consultarse en el fichero *Demostraciones*, en los archivos que acompañan a este proyecto. Esta prueba es un demostrador mínimo de las capacidades del *sensor de rango y rumbo*, las cuales cumplen con lo esperado.

5.4 Pruebas del sistema de programación remota

En la siguiente imagen se muestra el sistema de programación cargando un programa en una de las plataformas Argos:

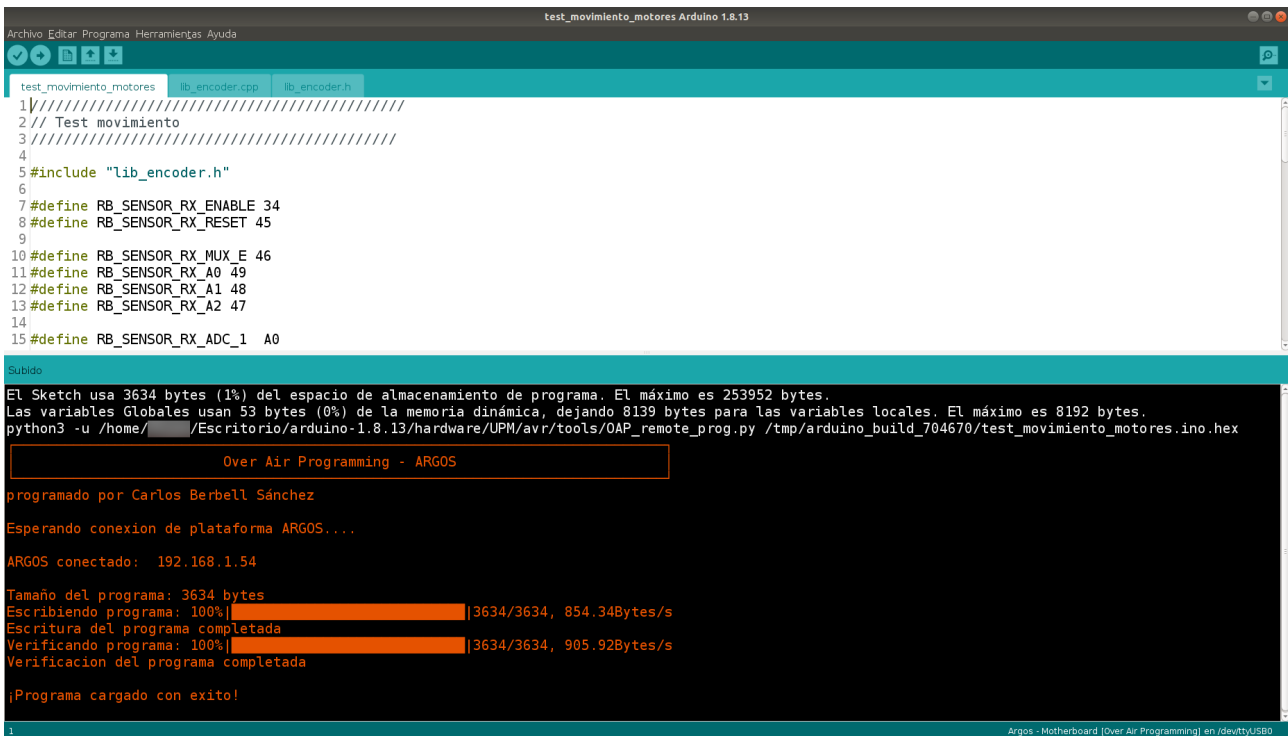


Imagen 51: Sistema de programación remota cargando un programa

Durante el proceso de carga, la pantalla de la interfaz de usuario de la plataforma que recibe el programa muestra el progreso:

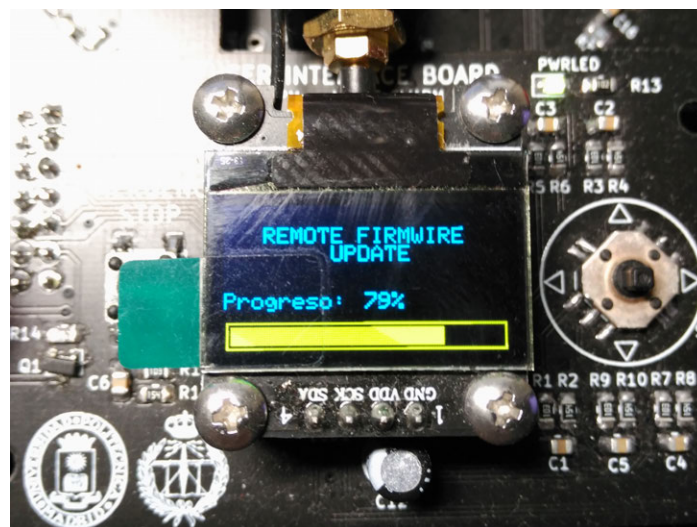


Imagen 52: Interfaz de Argos durante la carga remota

El sistema de carga remota se ha usado ampliamente durante el desarrollo del software, por lo que su funcionamiento se considera validado.

6. Presupuesto

El presupuesto de este proyecto se está dividido en tres partidas: costo de recursos no específicos del proyecto, costos de recursos específicos del proyecto, y costos de los recursos humanos.

Los costos de recursos no específicos son el conjunto de gastos derivados del uso de recursos materiales no adquiridos específicamente para este proyecto: ordenadores, instrumentación, herramientas. Los costos de recursos específicos del proyecto constituyen todos los gastos en materiales, software, herramientas que han sido adquiridos exclusivamente para la realización de este proyecto y no pueden destinarse a otros fines. Los costos de los recursos humanos son aquellos imputables a los salarios de los trabajadores que han sido empleados en este proyecto.

6.1 Costo de recursos no específicos

Para la realización de este proyecto se han empleado los siguientes recursos no específicos: ordenador, fuente de alimentación, multímetro, estación de soldadura, lupa binocular y herramientas varias (destornilladores, alicates, crimpadoras, etc). El coste de estos recursos se calcula teniendo en cuenta que el costo por hora de utilización depende de la vida útil, y del número de horas laborales anuales. En España, para una jornada de 40 horas semanales, el número de horas anuales es 1826 horas.

Recurso	Costo	Vida útil	Costo de utilización por hora
Ordenador	1.300,00 €	6	0,119 €
Fuente de alimentación de laboratorio Korah KA3005D	150,00 €	5	0,014 €
Multímetro Brymen 869	90,00 €	5	0,008 €
Estación de soldadura Yihua 995D	130,00 €	5	0,012 €
Lupa binocular x40 aumentos	100,00 €	5	0,009 €
Impresora 3D Anet A8	160,00 €	5	0,015 €
Herramientas varias	30,00 €	5	0,003 €

Tabla 10: Coste por hora de los recursos no específicos

Recurso	Actividad	Horas	Costo de utilización
Ordenador	Investigación preliminar	25	120,436 €
	Diseño del hardware	720	
	Ensamblado	120	
	Diseño del software	150	
Fuente de alimentación de laboratorio	Depuración del hardware	50	0,685 €
Multímetro Brymen 869	Ensamblado	120	1,396 €
	Depuración del hardware	50	
Estación de soldadura Yihua 995D	Ensamblado	120	2,017 €
	Depuración del hardware	50	
Lupa binocular x40 aumentos	Ensamblado	120	1,552 €
	Depuración del hardware	50	
Impresora 3D Anet A8	Producción de piezas 3D	40	0,584 €
Herramientas varias	Ensamblado	120	0,329 €

Total	127,00 €
--------------	----------

Tabla 11: Utilización y costo total de los recursos no específicos

6.2 Costo de los recursos específicos

En la Tabla 12 se presenta el costo de los recursos específicos, los cuales comprenden los componentes, el costo de fabricación de las PCB, consumibles para el montaje (estaño, manguitos termo retráctiles, cables, etc).

Recurso	Cantidad	Costo por unidad	Costo total
Componentes placa encoders	4	2,20 €	8,80 €
Componentes placa de potencia	4	19,23 €	76,92 €
Componentes placa madre	4	23,42 €	93,68 €
Componentes placa emisor R&B	4	11,19 €	44,76 €
Componentes placa receptor R&B	4	21,26 €	85,04 €
Componentes placa interfaz de usuario	4	5,15 €	20,60 €
Componentes adicionales, 2º prototipo	1	85,13 €	85,13 €
PCBs primer prototipo	4	7,05 €	28,20 €
PCBs primer prototipo	4	7,01 €	28,04 €
Consumibles para el montaje	1	15,00 €	15,00 €
Bobina de plástico para la impresora 3D	1	25,00 €	25,00 €
Total			511,17 €

Tabla 12: Costo total de los recursos específicos

6.3 Costo de recursos humanos

Para calcular el costo de los recursos humanos, se tomará como referencia el salario medio de un ingeniero electrónico junior en España proporcionado por el portal *Jobted* [61], el cual es de 20.700€ por año. Para una jornada de 40 horas, el número de horas trabajadas al año es 1826 horas, por lo que el costo por hora de emplear a un ingeniero junior es de: 11,34€ la hora. Entonces, para cada una de las tareas realizadas, el coste de los recursos humanos es:

Actividad	Horas	Costo de RH
Investigación preliminar	25	283,50 €
Diseño del hardware	720	8.164,80 €
Ensamblado	120	1.360,80 €
Diseño del software	150	1.701,00 €
Depuración del hardware	50	567,00 €
Ensamblado	120	1.360,80 €
Total		13.437,90 €

Tabla 13: Costo de total de recursos humanos

6.4 Costo total del proyecto

Sumando el costo de cada una de las partidas expuesto en las Tabla 11, Tabla 12, Tabla 13, el costo total del proyecto es:

$$\text{Costo total del proyecto} = 14079,07 \text{ €}$$

7. Impacto del proyecto

Uno de los fines que este proyecto persigue es hacer más accesible el campo de la robótica enjambre, proporcionando una solución de bajo costo y que requiera una curva de aprendizaje pequeña, para su utilización en entornos de investigación. Actualmente, tal como se describe en el capítulo de Antecedentes, el campo de investigación de la robótica de enjambre y sus campos de investigación relacionados, ya presentan múltiples aplicaciones prácticas. Por citar algunas de ellas: el despliegue autónomo de redes de sensores, y redes de comunicaciones en entornos aislados.

Dado que este proyecto contribuye a la investigación y desarrollo del campo de la robótica de enjambre, y por tanto, de sus aplicaciones prácticas, este proyecto presenta tres implicaciones fundamentales:

- **Implicaciones medioambientales:** el desarrollo de sistemas autónomos de sensores robustos puede contribuir significativamente a aplicaciones de motorización ambiental en entornos aislados. Esto permitiría detectar la fuente de estas alteraciones ambientales con la mayor celeridad y, actuar con mayor eficacia.
- **Implicaciones económicas:** esta plataforma ofrece una solución de bajo costo para el testeo y validación de algoritmos de robótica de enjambre, lo que permite que los equipos de investigación centren sus recursos económicos en lo más importante: el desarrollo y validación de nuevas ideas, abaratando los costos de investigación y maximizando sus rendimientos.
- **Implicaciones de seguridad:** el disminuir la barrera de entrada económica para la investigación de sistemas de robótica en enjambre debería motivar una investigación más activa en este campo. Dado que los sistemas de robótica de enjambre son por sus propias características: robustos, escalables y resilientes, su generalizada contribuiría a mejorar la seguridad de las sociedades y de los individuos, dependientes de la tecnología y por tanto, vulnerables a las consecuencias de sus fallos.

Así pues, teniendo estas implicaciones, este proyecto aporta valor y contribuye a alcanzar los siguientes ODS (Objetivos de Desarrollo Sostenible):

- **ODS-9- Industria, innovación e infraestructura:** la aplicación de las tecnologías basadas en la robótica de enjambre contribuirán a mejorar la seguridad y fiabilidad de las infraestructuras críticas.
- **ODS-11 - Ciudades y comunidades sostenibles:** el desarrollo de sistemas robustos, escalables y resilientes basados en los principios de la *SI*, supondrá una mejora en los actuales sistemas de supervisión de entornos urbanos, lo que en última instancia, permitirá mejorar la sostenibilidad de las comunidades.
- **ODS 13 - Acción por el clima:** los sistemas de supervisión medioambiental podrían beneficiarse de las tecnologías basadas en *SR*, ya que sus características permiten un despliegue autónomo en áreas aisladas de redes de sensores robustas. Esto permitiría obtener mayor cantidad de datos sobre el nivel de amenaza, y el origen de los actuales problemas climáticos; permitiendo afrontar sus causas con mayor precisión.

8. Conclusiones

8.1 Conclusiones

Los objetivos de este proyecto son: el desarrollo de una plataforma robótica de bajo coste orientada a la implementación de algoritmos de robótica de enjambre, el desarrollo de una API que mínima que permita mostrar su funcionalidad, la implementación de un *core* de Arduino que de soporte a esta plataforma dentro del entorno de Arduino, y por último, el desarrollo de una herramienta de programación remota integrada con el entorno de Arduino.

El primero objetivo ha sido alcanzado con éxito, obteniéndose un hardware versátil con una característica difícil de encontrar en las plataformas comerciales: el *sensor de rango y rumbo*. Además, la aproximación teórica al problema de diseño, y las disertaciones expuestas en el apartado 4.1.3, son una aportación valiosa, ya que el proceso de diseño sistemas similares [16], [17], es fundamentalmente empírico y está poco documentado.

El segundo objetivo también se ha alcanzado, la API mínima desarrollada permite mostrar el funcionamiento básico de la plataforma y constituye un punto de partida para desarrollos más avanzados.

El tercer objetivo se ha alcanzado satisfactoriamente, el desarrollo de un *core* de Arduino para la plataforma ARGOS que integre de forma nativa FreeRTOS, permite la integración de la plataforma en el entorno de Arduino, y sienta las bases el desarrollo de una API potente y más avanzada.

Por último, también se ha alcanzado el último objetivo: el desarrollo de una herramienta de programación remota, que contribuirá a hacer desarrollos más eficientes, eliminando la necesidad de cargar los programas en las plataformas de forma individual.

8.2 Trabajos futuros

Dado que los objetivos originales del proyecto, recogidos en el documento de *Antecedentes*, difieren de los perseguidos en este proyecto por las razones expuesta en el apartado Marco y motivación del proyecto, se proponen las siguientes líneas de desarrollo que continúan con los objetivos iniciales del proyecto, partiendo de la plataforma y las soluciones alcanzadas.

En primer lugar, el uso transparente de la misma API para la programación de la plataforma desarrollada y la plataforma simulada en *Gazebo*, que es uno de los objetivos que aparecen en el *Anteproyecto*, requiere re-implementar el conjunto de módulos que controlan el hardware, para que los programas y algoritmos desarrollados puedan ejecutarse en el entorno virtual de *Gazebo*. Sin embargo, el nivel de abstracción actual de la API, cercano al hardware, complica esta implementación.

Así pues, se sugiriere como trabajo futuro, rediseñar la API utilizando un modelo a capas, que abstraiga los detalles del hardware. Esto contribuiría a simplificar una futura integración con

Gazebo, y mejoraría el estilo de programación, lo que haría más cómodo para el investigador su uso para la implementación de algoritmos de robótica de enjambre.

En segundo lugar, dadas las características de la plataforma y número de plataformas disponibles, los algoritmos de robótica de enjambre que mejor se podrían adaptar en trabajos futuros son aquellos relacionados con la formación de patrones y movimiento coordinado. Así, se surgiere comenzar por la implementación de los algoritmos contenidos estas categorías.

En tercer lugar, y con el objetivo de ampliar las capacidades de la plataforma, se propone como trabajo futuro el desarrollo de un sistema que permita la detección de objetos y su transporte. La implementación de este hardware no requiere de grandes modificaciones, y permitiría implementar algoritmos de agrupamiento de objetos y ensamblado.

En cuarto lugar, si bien las características del *sensor de rango y rumbo* son adecuadas para el número de robots de los que se dispone, la frecuencia de refresco del mismo limita su aplicación a mayor escala. Por tanto, se propone como trabajo futuro, acometer las mejoras necesarias para aumentar la frecuencia de refresco del sensor.

En quinto lugar, otra mejora que podría realizarse sobre el hardware es el diseño de un sistema de carga de batería integrado, y una base de carga que permitan la carga autónoma de los robots. Eso, junto con el sistema de programación remota desarrollado, simplificaría aún más el uso de las plataformas para la investigación.

Por último, y aunque en el presente trabajo centrado en el hardware no se ha expuesto, durante la fase de investigación preliminar, se encontraron algunos condicionantes que complican la integración de *Gazebo* y el entorno de Arduino. En primer lugar, el desarrollo de modelos para *Gazebo* requiere la compilación con grandes bibliotecas que proporcionan a los modelos acceso a toda la infraestructura de *Gazebo*, sin embargo, el sistema de compilación que utiliza Arduino es completamente inadecuado para este fin por su escasa flexibilidad. En segundo lugar, *Gazebo* despliega los robots simulados en diferentes hilos, dentro de su proceso. Esto produce que el estilo de programación de las bibliotecas de Arduino, que hace un uso permanente de variables globales, sea incompatible con *Gazebo*. La solución a este problema es el desarrollo de un *programa ejecutor*, que se conecte al servidor de *Gazebo*, y lance los controladores de los robots simulados en procesos independientes. En la carpeta, *Trabajos futuros*, hay algunos desarrollos preliminares de este concepto que podrían servir como base.

9. Referencias

- [1] - G. Beni, "From swarm intelligence to swarm robotics," in *Proc. Swarm Robot. Workshop*, 2004.
- [2] - E. Sahin, "Swarm robotics: from sources of inspiration to domains of application," in *Proc. Eur. Conf. Artif. Life*, 2005, pp. 1–10.
- [3] - M. Brambilla, E. Ferrante, M. Birattari, and M. Dorigo, "Swarm robotics: A review from the swarm engineering perspective," *Swarm Intell.*, vol. 7, no. 1, pp. 1–41, Mar. 2013.
- [4] - N. Nedjaha, J. A. R. P. da Costa, J. P. B. L. Paixão, and J. L. P. J. da Costa, "Review of methodologies and tasks in swarm robotics towards standardization," *Robot. Auton. Syst.*, vol. 97, pp. 248–261, Nov. 2017.
- [5] - M. Senanayakea, L. L. Ratnayakea, and M. A. U. P. Perera, "Search and tracking algorithms for swarms of robots: A survey," *Robot. Auton. Syst.*, vol. 71, pp. 195–215, Sep. 2015.
- [6] - P. Bladin, "W. Grey Walter, pioneer in the electroencephalogram," *J. Clin. Neurosci.*, vol. 4, no. 1, pp. 135–142, Mar. 1997.
- [7] - T. Fukuda, S. Nakagawa, Y. Kawauchi, and M. Buss, "Approach to the dynamically reconfigurable robotic system with self-organizing capability," in *Proc. IEEE Int. Workshop Intell. Robots Syst.*, 1988, pp. 493–498.
- [8] - G. Beni and J. Wang, "Swarm Intelligence in Cellular Robotic Systems," in *Robots and Biological Systems: Towards a New Bionics?*, P. Dario, G. Sandini, and P. Aebischer, Eds. Berlin, Heidelberg: Springer, 1993, pp. 703–712.
- [9] - G. Beni and J. Wang, "Self-Organizing Sensory Systems," in *Highly Redundant Sensing in Robotic Systems*, J. T. Tou and J. G. Balchen, Eds. Berlin, Heidelberg: Springer, 1988, pp. 1–9.
- [10] - A. Martinoli, "Collective complexity out of individual simplicity: A review of swarm intelligence: From natural to artificial systems, by E. Bonabeau, M. Dorigo, and G. Theraulaz," *Artif. Life*, vol. 7, no. 3, pp. 315–319, 2001.
- [11] - G. Dudek, M. Jenkin, E. Milios, P. Moore, and N. O. S. of Computer Science, "A taxonomy for swarm robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 1993, vol. 1, pp. 58–64.
- [12] - M. Dorigo *et al.*, "Swarmanoid: A novel concept for the study of heterogeneous robotic swarms," in *Proc. Int. Joint Conf. Artif. Intell.*, 2012, pp. 1599–1604.
- [13] - A. Jevtic, "Warm intelligence novel tools for optimization, feature extraction, and multi-agent system modeling", Tesis doctoral, SSR, ETSIT – UPM, Madrid, España, 2011, pp. 101.
- [14] - I. Kelly and A. Martinoli, "A scalable, on-board localization and communication system for indoor multi-robot experiments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2004, vol. 4, pp. 3205–3210.
- [15] - J. Pugh, A. Martinoli, and C. H. D. S. Robotics, "Local range and bearing sensing using infrared transceivers in mobile robotics," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2004, vol. 3, pp. 2481–2486.

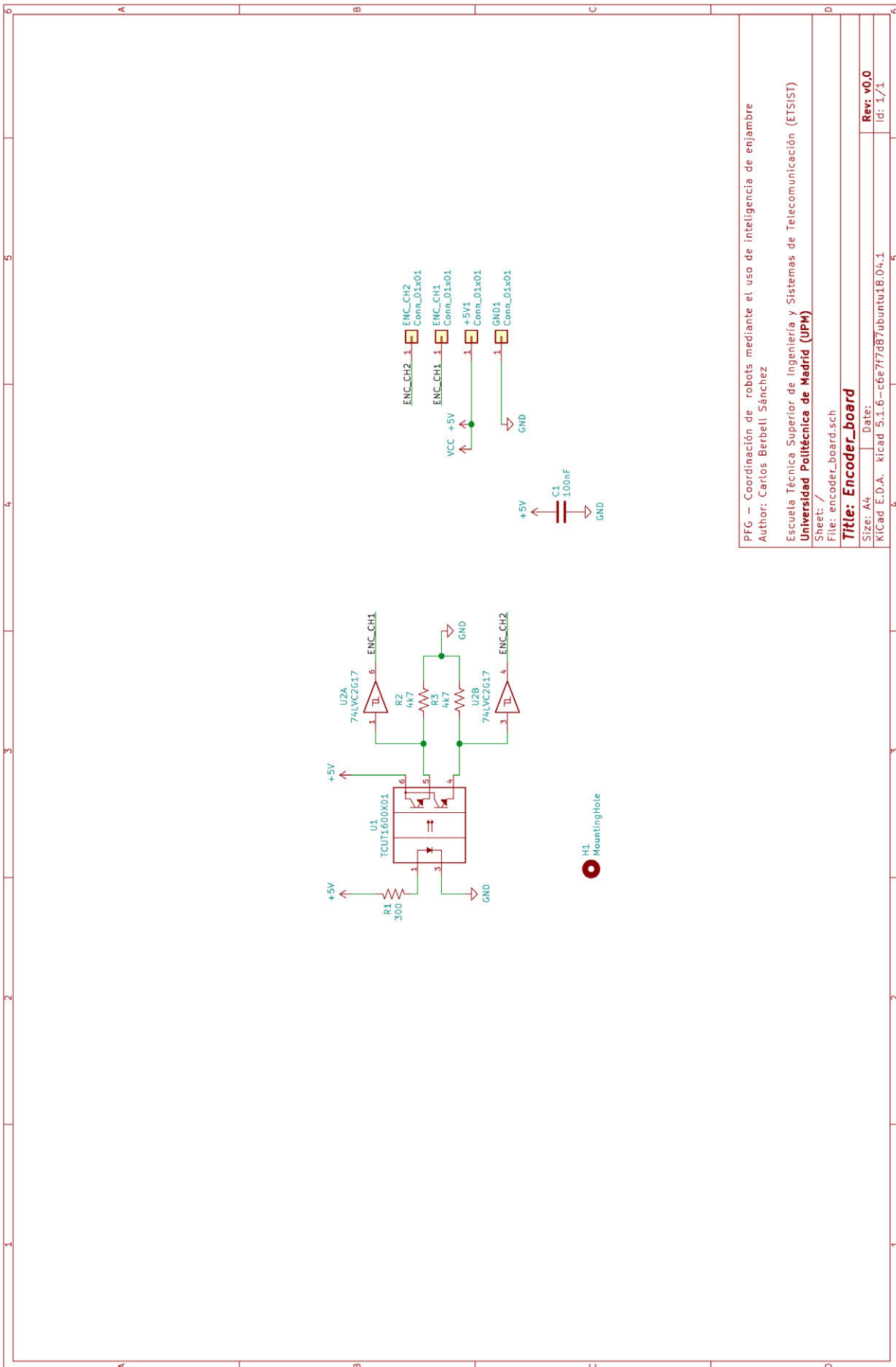
- [16] - J. Pugh, A. Martinoli, and F. Engineering, "A Fast On-Board Relative Positioning Module for Multi-Robot Systems," in *Proc. IEEE/RSJ Int. Conf. Robot. Autom.*, 2009, pp. 4501–4506.
- [17] - J. F. Roberts, J. Pugh, and A. Martinoli, "2.5D Infrared Range and Bearing System for Collective Robotics," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2009, pp. 2379–2384.
- [18] - O. Soysal and E. Sahin, "Probabilistic aggregation strategies in swarm robotic systems," in *Proc. Swarm Intell. Workshop*, 2005, pp. 250–257.
- [19] - A. E. Turgut, H. Çelikkanat, and E. Şahin, "Self-organized flocking in mobile robot swarms," *Swarm Intell.*, vol. 2, no. 2-3, pp. 101–120, Aug. 2008.
- [20] - AKM, "Optical Encoder Tutorial," AKM, [Online]. Available: <https://www.akm.com/global/en/products/rotation-angle-sensor/tutorial/optical-encoder/>. [Consultado el: Jun. 15, 2025].
- [21] - Vishay, "Datasheet TCUT1600X01," [Online]. Available: <https://www.vishay.com/docs/84174/tcut1600x01.pdf>. [Consultado el: Jun. 28, 2025].
- [22] - DFRobot, "DFRobot Gear Motor," [Online]. Available: <https://www.dfrobot.com/product-100.html>. [Consultado el: Jun. 03, 2025].
- [23] - C. Buttay, "Puente H, esquema," 2007. [Online]. Available: <https://commons.wikimedia.org/w/index.php?curid=854051>. (CC BY-SA 3.0). [Consultado el: Jun. 19, 2025].
- [24] - Thewrightstuff, "Modulación por ancho de pulso," 2017. [Online]. Available: <https://creativecommons.org/licenses/by-sa/4.0>. (CC BY-SA 4.0 via Wikimedia Commons). [Consultado el: Jun. 07, 2025].
- [25] - Texas Instruments, "Datasheet DRV8876," [Online]. Available: <https://www.ti.com/lit/ds/symlink/drv8876.pdf>. [Consultado el: Jun. 22, 2025].
- [26] - Texas Instruments, "Calculating Power Dissipation for a H-Bridge or Half Bridge Driver," Application Report SLVA504A, Mar. 2012. [Online]. Available: <https://www.ti.com/lit/an/slva504a/slva504a.pdf>. [Consultado el: Jun. 11, 2025].
- [27] - Texas Instruments, "AN-2020 Thermal Design By Insight, Not Hindsight," Application Report SNVA419C, Sep. 2008. [Online]. Available: <https://www.ti.com/lit/an/snva419c/snva419c.pdf>. [Consultado el: Jun. 25, 2025].
- [28] - Texas Instruments, "Datasheet BQ34110," [Online]. Available: <https://www.ti.com/mx/lit/gpn/bq34110>. [Consultado el: Jun. 04, 2025].
- [29] - Texas Instruments, "Manual técnico de referencia BQ34110," [Online]. Available: <https://www.ti.com/mx/lit/pdf/sluubf7>. [Consultado el: Jun. 29, 2025].
- [30] - Torex Semiconductor, "Datasheet XP231P0201TR," [Online]. Available: <https://product.torexsemi.com/system/files/series/XP231P0201TR.pdf>. [Consultado el: Jun. 16, 2025].
- [31] - Vishay, "Datasheet SI2300," [Online]. Available: <https://www.vishay.com/docs/65701/si2300ds.pdf>. [Consultado el: Jun. 09, 2025].

- [32] - S. Franco, *Operational Amplifier Theory and Applications*. New York, NY, USA: McGraw-Hill, pp. 336.
- [33] - Vishay, "Datasheet TSAL6400," [Online]. Available: <https://www.vishay.com/docs/81011/tsal6400.pdf>. [Consultado el: Jun. 21, 2025].
- [34] - S. King, "Luminous Intensity of an LED as a Function of Input Power." in *International School Bangkok Journal of Physics*, vol. 2, Jun. 2008.
- [35] - S. Franco, *Operational Amplifier Theory and Applications*. New York, NY, USA: McGraw-Hill, pp. 40.
- [36] - A. R. Hambley, *Electronics*. Upper Saddle River, NJ, USA: Pearson Prentice Hall, pp. 163.
- [37] - Texas Instruments, "TLV9064 Datasheet."
- [38] - ON Semiconductor, "BC817 Datasheet," [Online]. Available: <https://www.onsemi.com/download/data-sheet/pdf/bc817-40w-d.pdf>. [Consultado el: Jun. 05, 2025].
- [39] - P. Amo, et. al. *Señales y sistemas, ejercicios*. Madrid: Editorial Ciencia 3, 2004. pp. 154.
- [40] - Texas Instruments, "Datasheet TS5A3166," [Online]. Available: <https://www.ti.com/lit/gpn/TS5A3166>. [Consultado el: Jun. 18, 2025].
- [41] - Analog Devices, "Datasheet ADR5040A," [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/adr5040_5041_5043_5044_5045.pdf. [Consultado el: Jun. 13, 2025].
- [42] - S. Franco, *Operational Amplifier Theory and Applications*. New York, NY, USA: McGraw-Hill, pp. 173.
- [43] - S. Franco, *Operational Amplifier Theory and Applications*. New York, NY, USA: McGraw-Hill, pp. 179.
- [44] - S. Franco, *Operational Amplifier Theory and Applications*. New York, NY, USA: McGraw-Hill, pp. 159.
- [45] - Vishay, "Datasheet BPV22NF," [Online]. Available: <https://www.vishay.com/docs/81509/bpv22nf.pdf>. [Consultado el: Jun. 27, 2025].
- [46] - OSI Optoelectronics, "Photodiode Characteristics and Applications," [Online]. Available: <https://www.osioptoelectronics.com/media/pages/knowledgebase/b954012b64-1675100541/an-photodiode-parameters-and-characteristics.pdf>. [Consultado el: Jun. 06, 2025].
- [47] - Nexperia, "Datasheet BAS70-04," [Online]. Available: <https://assets.nexperia.com/documents/data-sheet/BAS70-04.pdf>. [Consultado el: Jun. 20, 2025].
- [48] - S. Franco, *Operational Amplifier Theory and Applications*. New York, NY, USA: McGraw-Hill, pp. 10
- [49] - S. Franco, *Operational Amplifier Theory and Applications*. New York, NY, USA: McGraw-Hill, pp. 285.

- [50] - S. Franco, *Operational Amplifier Theory and Applications*. New York, NY, USA: McGraw-Hill, pp. 150.
- [51] - S. Franco, *Operational Amplifier Theory and Applications*. New York, NY, USA: McGraw-Hill, pp. 312.
- [52] - S. Franco, *Operational Amplifier Theory and Applications*. New York, NY, USA: McGraw-Hill, , pp. 467.
- [53] - Digi-Key, "Detector de pico 1," Application Note. [Online]. Available: <https://application-notes.digchip.com/003/3-4081.pdf>. [Consultado el: Jun. 14, 2025].
- [54] - EDN, "High-speed peak detector uses ECL comparator," [Online]. Available: <https://www.edn.com/high-speed-peak-detector-uses-ecl-comparator/>. [Consultado el: Jun. 08, 2025].
- [55] - STMicroelectronics, "Datasheet TS3011," [Online]. Available: <https://www.st.com/resource/en/datasheet/ts3011.pdf>. [Consultado el: Jun. 23, 2025].
- [56] - Microchip Technology, "Datasheet Atmega2560," [Online]. Available: <https://ww1.microchip.com/downloads/aemDocuments/documents/OTH/ProductDocuments/DataSheets/ATmega640-1280-1281-2560-2561-Datasheet-DS40002211A.pdf>. [Consultado el: Jun. 10, 2025].
- [57] - Espressif Systems, "Datasheet ESP8266," [Online]. Available: https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf. [Consultado el: Jun. 26, 2025].
- [58] - Espressif Systems, "ESP8266 Documentation," [Online]. Available: <https://docs.espressif.com/projects/esptool/en/latest/esp8266/advanced-topics/boot-mode-selection.html#select-bootloader-mode>. [Consultado el: Jun. 02, 2025].
- [59] - FreeRTOS, "Creating a New FreeRTOS Port," [Online]. Available: <https://www.freertos.org/Documentation/02-Kernel/03-Supported-devices/01-FreeRTOS-porting-guide>. [Consultado el: Jun. 17, 2025].
- [60] - Greiman, "FreeRTOS-Arduino," GitHub, [Online]. Available: <https://github.com/greiman/FreeRTOS-Arduino>. [Consultado el: Jun. 30, 2025].
- [61] - Jobted.es, "Salario ingeniero junior," [Online]. Available: <https://www.jobted.es/salario/ingeniero-electr%C3%B3nico>. [Consultado el: Jun. 12, 2025].
- [62] - Microchip Technology, "AVR068: STK500 Communication Protocol," Application Note.
- [63] - P. Kroes, *Technical Artefacts: Creations of Mind and Matter: A Philosophy of Engineering Design*. Dordrecht, Netherlands: Springer, 2012.
- [64] - W. G. Vincenti, *What Engineers Know and How They Know It: Analytical Studies from Aeronautical History*. Baltimore, MD: The Johns Hopkins University Press, 1990.

Anexos

Anexo A - Esquemas de Argos



PFG — Coordinación de robots mediante el uso de inteligencia de enjambre
 Author: Carlos Berbell Sánchez

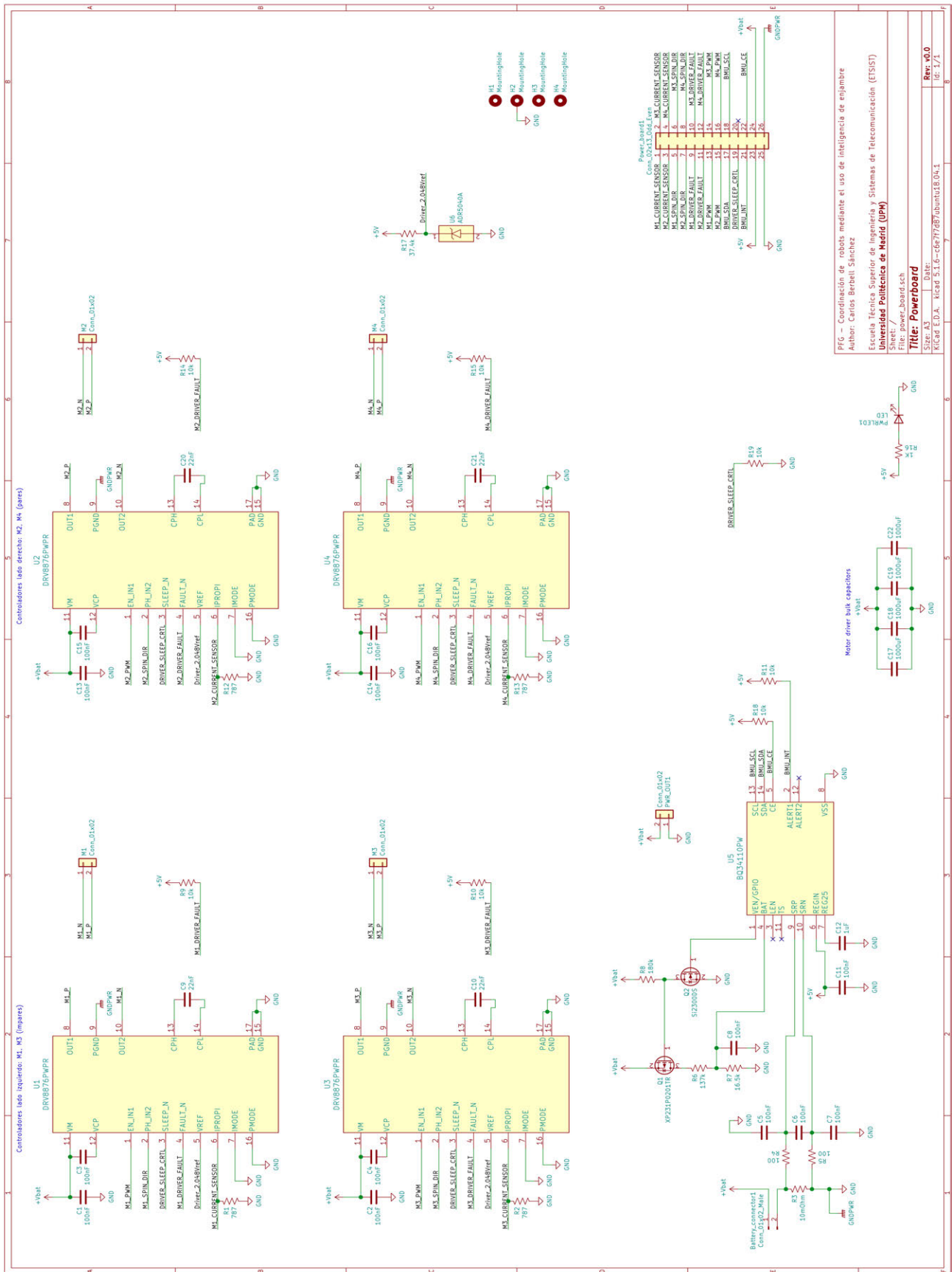
Escuela Técnica Superior de Ingeniería y Sistemas de Telecomunicación (ETSIIST)
Universidad Politécnica de Madrid (UPM)

Sheet: /
 File: encoder_board.sch

Title: Encoder_board

Size: A4 Date:
 KiCad E.D.A. kicad 5.1.6-c6e7f7d87abuntui18.04.1

Rev: v0.0
 Id: 1/1



RFQ - Coordinación de robots mediante el uso de Inteligencia de enjambre
 Autor: Carlos Berbel Sánchez
 Universidad Técnica Superior de Ingeniería y Sistemas de Telecomunicación (ETISIT)
 Sheet: /
 File: power-board.sch
Title: PowerBoard
 Size: A3
 Date:
 KICad E.D.A. Kicad 5.1.0-c6f7f68/ubuntu18.04.1
 Rev: V0.0
 Id: 1/1

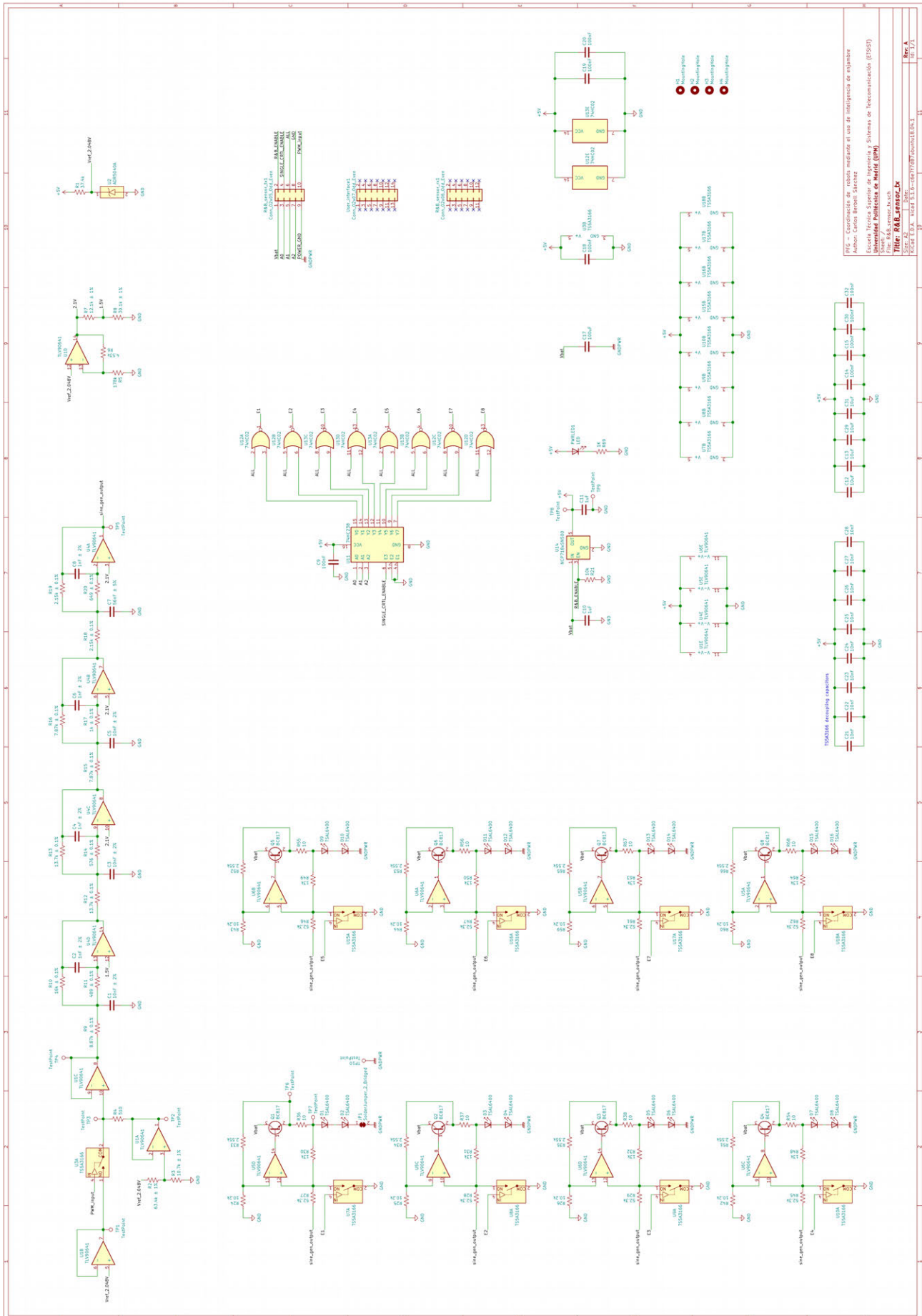
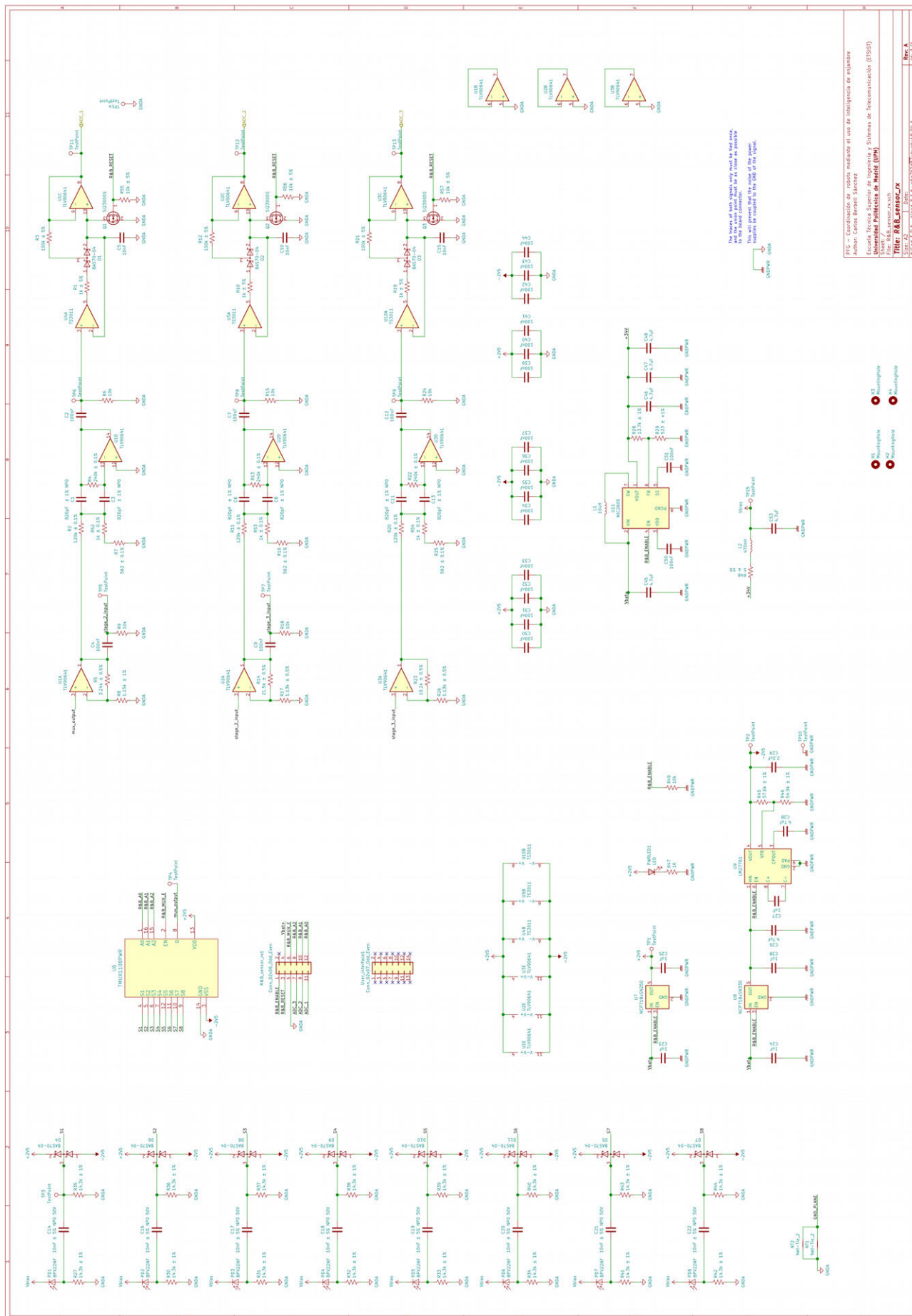
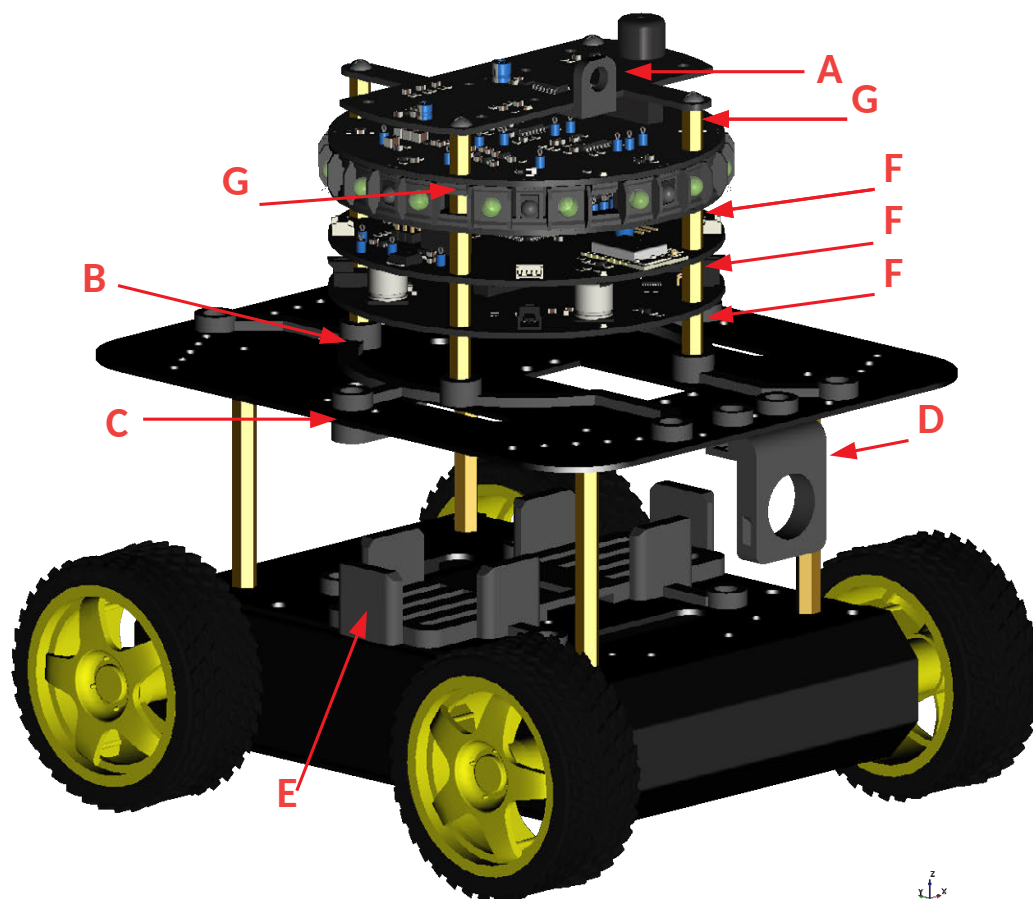
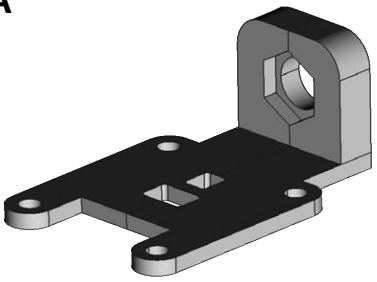
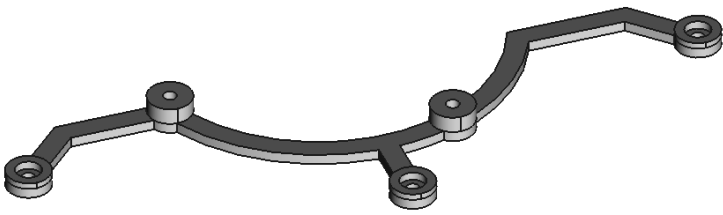
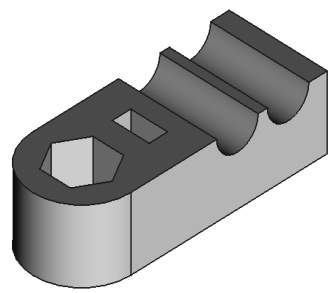
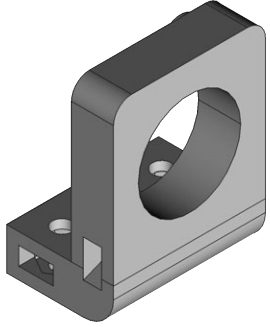



FIG. 2. Configuración de cables mediante el uso de inteligencia de empresa
 Author: Carlos Berini Sánchez
 Universidad Politécnica de Aragón (UPA)
 Title: R&B sensor, s.r.l.
 File: R&B_sensor.sch



Anexo B - Ensamblado y piezas 3D



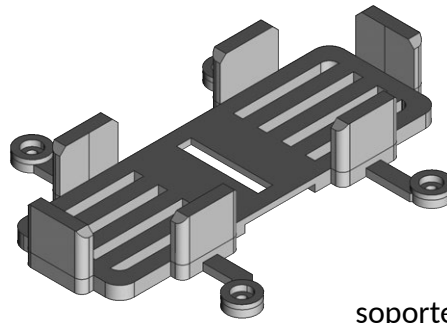
<p>A</p>  <p>soporte_antena_rev_A.FCStd</p>	<p>B</p>  <p>soporte_pcb_s_chasis.FCStd</p>	
<p>C</p>  <p>sujeta_cable_sensores.FCStd</p>	<p>D</p>  <p>soporte_sensor_ultrasonidos.FCStd</p>	<p>F</p>  <p>Separador M3 12mm</p>

G



Separador M3 11mm

E



soporte_bateria_LIPO.FCStd

4.1.2