



COMILLAS

UNIVERSIDAD PONTIFICIA

ICAI

GRADO EN INGENIERÍA EN TECNOLOGÍAS DE TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

DESARROLLO DE UN DRON AUTÓNOMO PARA EL TRANSPORTE DE MATERIALES SEMICONDUCTORES EN UNA FÁBRICA

Autor: Patricia Sánchez Martín

Director: Dr. Leonard Franklin Register

Madrid

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título
**DESARROLLO DE UN DRON AUTÓNOMO PARA EL TRANSPORTE DE
MATERIALES SEMICONDUCTORES EN UNA FÁBRICA**

en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el

curso académico 2023/24 es de mi autoría, original e inédito y

no ha sido presentado con anterioridad a otros efectos.

El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido

tomada de otros documentos está debidamente referenciada.



Fdo.: Patricia Sánchez Martín

Fecha: 30/06/2024

Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO



Fdo.: Dr. Leonard Franklin Register

Fecha: 30/06/2024



COMILLAS

UNIVERSIDAD PONTIFICIA

ICAI

GRADO EN INGENIERÍA EN TECNOLOGÍAS DE TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

DEVELOPMENT OF AN AUTONOMOUS DRONE FOR TRANSPORTING SEMICONDUCTOR MATERIALS IN A FACTORY

Autor: Patricia Sánchez Martín

Director: Dr. Leonard Franklin Register

Madrid

Agradecimientos

Gracias a todas las personas que me han acompañado durante el año de intercambio en *The University of Texas at Austin*.

En especial, me gustaría agradecer a mis compañeros Ryan Folmar, Brandon Vu, Nicholas Cagle, Noah Najvar y Srishaan Iyengar.

DESARROLLO DE UN DRON AUTÓNOMO PARA EL TRANSPORTE DE MATERIALES SEMICONDUCTORES EN UNA FÁBRICA

Autor: Sánchez Martín, Patricia.

Director: Register, Dr. Leonard Franklin.

Entidad Colaboradora: Texas Instruments.

RESUMEN DEL PROYECTO

Este proyecto se centra en el desarrollo de un sistema de dron autónomo dedicado al transporte de materiales semiconductores dentro de una fábrica. Concretamente, dicho sistema surge de las necesidades específicas de rendimiento y eficiencia operativas de la empresa *Texas Instruments*. A lo largo del proyecto, se seleccionaron y ensamblaron los componentes de hardware hasta el montaje de un dron desde cero, y se desarrolló el software necesario para su vuelo autónomo como prueba de concepto.

Palabras clave: dron autónomo, estaciones de prueba de materiales semiconductores, sistema de telemetría, Texas Instruments.

1. Introducción

Dadas sus propiedades eléctricas únicas, los materiales semiconductores han ganado en los últimos años un rol determinante en la industria tecnológica. Estos materiales forman las bases de la electrónica moderna, al estar presentes en la fabricación de componentes claves como transistores, diodos y circuitos integrados. En este contexto, la empresa Texas Instruments juega un papel dominante, al desarrollar chips de semiconductores para el uso en aplicaciones variadas, incluyendo el procesamiento digital de señales y tecnología analógica.

En la fabricación de materiales semiconductores, es imperativo un manejo preciso y delicado de los mismos. Tradicionalmente, dichos materiales son transportados por tierra, lo cual consume valioso espacio y provoca retrasos logísticos. Los drones, que utilizan espacio vertical en un entorno industrial, pueden aumentar la eficiencia en el transporte de materiales, mejorando la productividad y reduciendo los cuellos de botella.

Por lo tanto, este proyecto centrado en Texas Instruments, tiene como objetivo mejorar la eficiencia de sus procesos, así como establecer su liderazgo en innovación logística dentro de la industria.

2. Definición del proyecto

El proyecto implica el desarrollo de un sistema de dron inteligente para el transporte autónomo de materiales semiconductores dentro de una fábrica de Texas Instruments. El estado actual de la logística carece de la implementación de drones inteligentes para esta tarea, por lo que se presenta una oportunidad para innovar.

Para ser exitoso, este proyecto tiene como objetivo proporcionar una evaluación ante la posibilidad de la implementación real de un sistema de drones autónomos, la cual sirva como orientación para las posibles futuras inversiones en ello por parte de Texas Instruments. Además, el proyecto se alinea con la tendencia global de promover la automatización en

entornos industriales, lo que promete mejoras en la productividad, seguridad y sostenibilidad.

Los objetivos clave del proyecto son los siguientes:

- Eficiencia y optimización de espacio. Los drones permiten utilizar espacio vertical inutilizado en una fábrica para agilizar el transporte de materiales.
- Reducción de costes y riesgos logísticos. La automatización del transporte de materiales minimiza riesgos y costes asociados al manejo de los mismos.
- Mejora de productividad. El uso de drones permite un flujo continuo de operaciones y disminuye el riesgo de congestión.
- Sostenibilidad y seguridad. El uso de drones se trata de una alternativa de transporte más sostenible y segura frente a otros vehículos autónomos.
- Liderazgo en materia de innovación. Es importante posicionar a Texas Instruments como líder en el ámbito de innovación logística.

3. Descripción del sistema

Este proyecto implica la investigación y selección de componentes de hardware, el desarrollo de software para control autónomo, y la integración de ambos para alcanzar los objetivos especificados por Texas Instruments dentro del cronograma temporal y presupuesto definidos.

La parte de hardware del sistema del dron autónomo incluye una selección de módulos clave divididos en cuatro subsistemas que conforman el sistema de dron completo.

- Controlador de vuelo. Constituye el cerebro del dron, y es el responsable de interpretar señales y ajustar la velocidad de los motores para mantener la estabilidad. Este sistema está compuesto por un microcontrolador de la familia STM32 y sensores (acelerómetro, giroscopio y barómetro). Juntos, permiten al dron ajustar su posición y altitud en tiempo real.
- Estructura y propulsión. La estructura del dron está compuesta por un chasis que proporciona estabilidad y soporte a todos los componentes. Por su parte, la propulsión del dron está constituida por cuatro motores brushless (conectados a controladores electrónicos de velocidad ESCs) y hélices. Estos componentes trabajan juntos para crear el empuje y maniobrabilidad necesaria para el vuelo del dron.
- Sistema de alimentación. Este sistema incluye una batería LiPo de tres celdas, su conector XT60 y su cargador, y están encargados de suministrar energía a todos los componentes eléctricos del dron.
- Sistema de telemetría. Este sistema está formado por una Arduino UNO Clon y el sistema de comunicación APC220 para la transmisión de comandos desde la estación base al dron.

Por su parte, el software establece un punto de partida teórico para adquirir la navegación autónoma del dron y su mayor eficiencia operativa. Principalmente, en cuanto al software cabe destacar lo siguiente:

- Pseudocontrolador de vuelo autónomo. Basado en el software de un controlador de vuelo open-source [1], con modificaciones en la forma de obtención de las instrucciones de movimiento del dron.
- Procesamiento de datos de sensores. Procesamiento en tiempo real de las medidas de sensores para mantener en todo momento un control preciso del dron.

- Sistema de telemetría. Emplea una pareja de transceptores APC220 para la comunicación inalámbrica entre el sistema del dron y la estación de tierra, garantizando la transmisión de datos y la recepción de comandos en tiempo real.

En la ilustración siguiente se resume la arquitectura del hardware y software desarrollado para el proyecto.

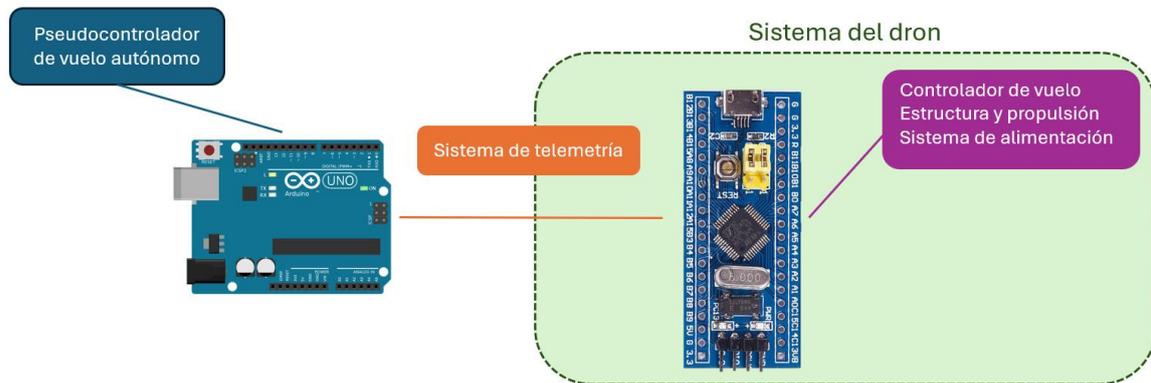


Ilustración 1- Arquitectura del proyecto desarrollado

4. Resultados

A lo largo del desarrollo del proyecto se afrontaron varios retos debido a la dificultad de integrar múltiples subsistemas, y a que surgieron problemas inesperados. Las fases de evaluación y test incluyeron la verificación de componentes individuales, subsistemas y, finalmente, del sistema completo del dron. Algunos de los resultados clave del proyecto son los siguientes:

- I. Las pruebas iniciales de los componentes del sistema confirmaron el correcto funcionamiento de todas las piezas de hardware utilizadas (como el microcontrolador STM32F103C8T6, los sensores y los ESCs). Un paso más allá, las pruebas de los subsistemas resultaron en una integración exitosa, a pesar de que se afrontaron problemas menores por el camino.
- II. Análogamente, la integración del sistema completo conllevó dificultades considerables, en particular con la estabilidad y equilibrio. Fueron necesarios ajustes tanto de software como de hardware para mejorar el rendimiento, estabilidad y precisión del vuelo.
- III. Por último, se comprobó que el sistema de telemetría lograba una comunicación efectiva entre el dron y la estación de control.
- IV. En general, a pesar de los desafíos, el proyecto cumplió los objetivos de desarrollar el hardware del dron y la prueba de concepto del software. Esto demuestra el potencial para un mayor desarrollo en la investigación acerca de la implementación de drones en entornos industriales.

En la imagen siguiente se muestra el sistema del dron montado desde cero.

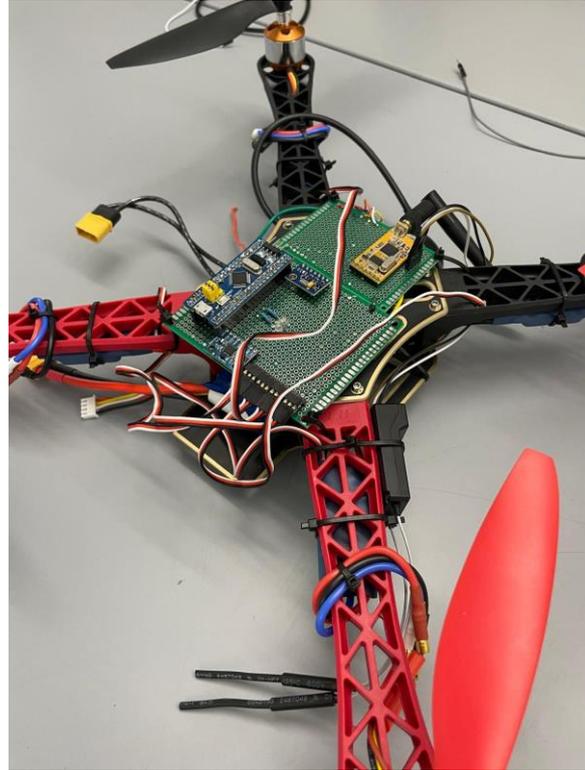


Ilustración 2- Sistema del dron

5. Conclusiones

El desarrollo de un sistema inteligente de dron autónomo para Texas Instruments supuso varios desafíos, especialmente debido a las limitaciones presupuestarias que impidieron la compra de un dron programable para uso profesional desde un principio. Otras complicaciones en el proyecto vinieron por problemas imprevistos (como la rotura de un ESC, una hélice y un motor durante colisiones) y una pronunciada curva de aprendizaje en el ensamblaje del hardware y la programación del sistema de vuelo autónomo. A pesar de estos obstáculos, de los cuales principalmente se aprendió a prepararse mejor frente a los imprevistos, se cumplieron los objetivos principales de Texas Instruments, que expresó su satisfacción con los resultados obtenidos en el proyecto.

En cuanto al trabajo futuro, se han identificado varias áreas de mejora, que incluyen: aumentar la capacidad de la batería para mejorar la autonomía del dron, la integración de un sistema de posicionamiento interno más sofisticado, y la implementación de un sistema de diagnóstico de datos del vuelo del dron en tiempo real. En materia de seguridad, cabe destacar que es imperativa la integración de sistemas robustos de manejo de riesgos y conocidos por los operadores humanos encargados de manejar el dron.

6. Referencias

- [1] Brokking, R. (n.d.). YMFC-32 Auto Downloads. Retrieved from http://www.brokking.net/ymfc-32_auto_downloads.html

DEVELOPMENT OF AN AUTONOMOUS DRONE FOR TRANSPORTING SEMICONDUCTOR MATERIALS IN A FACTORY

Author: Sánchez Martín, Patricia.

Supervisor: Register, Dr. Leonard Franklin.

Collaborating Entity: Texas Instruments.

ABSTRACT

This project focuses on the development of an autonomous drone system dedicated to transporting semiconductor materials within a factory. In particular, this system arises from the specific performance and operational efficiency needs of the company *Texas Instruments*. Throughout the project, the hardware components were selected and assembled until a drone was consolidated from scratch, and the software necessary for its autonomous flight was developed as proof of concept.

Keywords: autonomous drone, semiconductor materials' test stations, telemetry system, Texas Instruments.

1. Introduction

Given their unique electrical properties, semiconductor materials have gained a decisive role in the technology industry during recent years. These materials are the basis of modern electronics, as they are present in the manufacture of key components such as transistors, diodes and integrated circuits. In this context, Texas Instruments plays a dominant role, as it develops semiconductor chips for using in various applications, including digital signal processing and analogue technology.

In the manufacture of semiconductor materials, precise and delicate handling of the materials is imperative. Traditionally, these materials are transported by land, which consumes valuable space and causes logistics delays. Drones, which can use vertical space in an industrial environment, can increase the efficiency of material transportation, improving productivity and reducing bottlenecks.

Therefore, this project focused on Texas Instruments aims to improve the efficiency of its processes, as well as establish its leadership in logistics innovation within the industry.

2. Project Definition

The project involves the development of an intelligent drone system for the autonomous transport of semiconductor materials within a Texas Instruments factory. The current state of logistics lacks the implementation of intelligent drones for this task, so an opportunity to innovate is presented.

To be successful, this project aims to provide an assessment of the possibility of the real implementation of an autonomous drone system, which serves as a guide for possible future investments in it by Texas Instruments. In addition, the project aligns with the global trend of promoting automation in industrial environments, which promises improvements in productivity, safety and sustainability.

The key objectives of the project are the following:

- Efficiency and space optimization. Drones allow the use of unutilized vertical space in a factory to speed up the transport of materials.
- Reduction of logistic costs and risks. The automation of the transport of materials minimizes risks and costs associated with the handling of said materials.
- Improved productivity. The use of drones allows for a continuous flow of operations and reduces the risk of congestion.
- Sustainability and safety. In addition, the use of drones is a sustainable and safe transportation alternative.
- Leadership in innovation. It is important to position Texas Instruments as a leader in the field of logistic innovation.

3. System Description

This project involves the research and selection of hardware components, the development of software for autonomous control, and the integration of both to achieve the objectives specified by Texas Instruments within the defined timeline and budget.

The hardware part of the autonomous drone system includes a selection of key modules divided into four subsystems that make up the complete drone system.

- Flight controller. This is the brain of the drone and is responsible for interpreting signals and adjusting the speed of the motors to maintain stability. This system is composed of a microcontroller from the STM32 family and sensors (accelerometer, gyroscope and barometer). Together, they allow the drone to adjust its position and altitude real-time.
- Structure and propulsion. The structure of the drone is composed of a chassis that provides stability and support to all the components. The propulsion of the drone is made up of four brushless motors (connected to electronic speed controllers ESCs) and propellers. These components work together to create the thrust and manoeuvrability necessary for the flight of the drone.
- Power system. This system includes a three-cell LiPo battery, its XT60 connector and its charger, and is responsible for supplying power to all the drone's electrical components.
- Telemetry system. This system is made up of an Arduino UNO Clone and the APC220 communication system for transmitting commands from the base station to the drone.

On the other hand, the software establishes a theoretical starting point for acquiring autonomous navigation of the drone and its greater operational efficiency. Mainly, regarding the software, the following should be highlighted:

- Pseudo autonomous flight controller. Based on the software of an open-source flight controller [1], with modifications in the source for obtaining the drone's movement instructions.
- Sensor data processing. Real-time processing of sensor measurements to always maintain precise control of the drone.
- Telemetry system. It uses a pair of APC220 transceivers for wireless communication between the drone system and the ground station, ensuring data transmission and command reception real-time.

The following illustration summarizes the architecture of the hardware and software developed for the project.

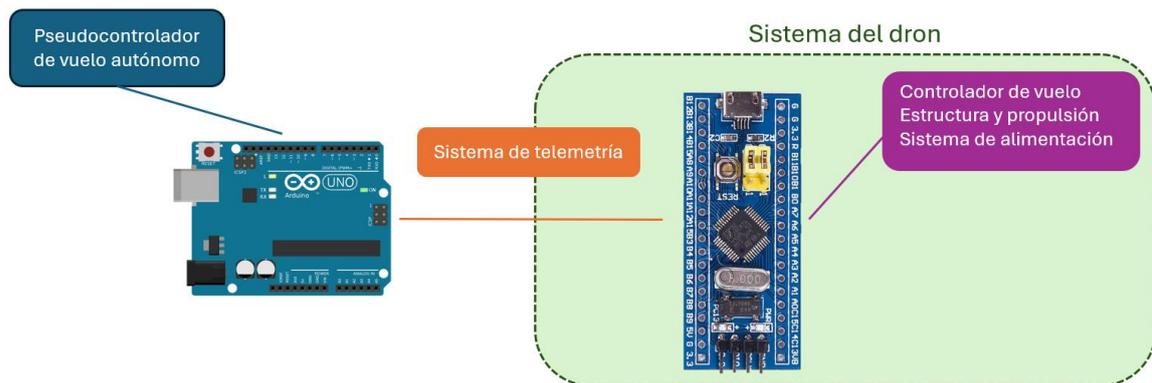


Illustration 1- The project's architecture

4. Results

Throughout the development of the project, several challenges were faced due to the difficulty of integrating multiple subsystems and unexpected issues arising. The evaluation and testing phases included verification of individual components, subsystems, and eventually the entire drone system. Some of the key outcomes of the project are as follows.

- I. Initial testing of the system's components confirmed the proper functioning of all of its hardware parts (such as the STM32F103C8T6 microcontroller, sensors, and ESCs). Going a step further, testing of the subsystems resulted in successful integration, despite minor issues being encountered along the way.
- II. Similarly, the integration of the entire system faced considerable difficulties, particularly with stability and balance. Both software and hardware adjustments had to be made to improve flight performance, stability, and accuracy.
- III. Finally, the telemetry system was found to achieve effective communication between the drone and the control station.
- IV. Overall, despite the challenges faced, the project met the objectives of developing the drone's hardware and software's proof of concept. This demonstrates the potential for further development of research into the implementation of drones in industrial environments.

The image below shows the drone system assembled from scratch.

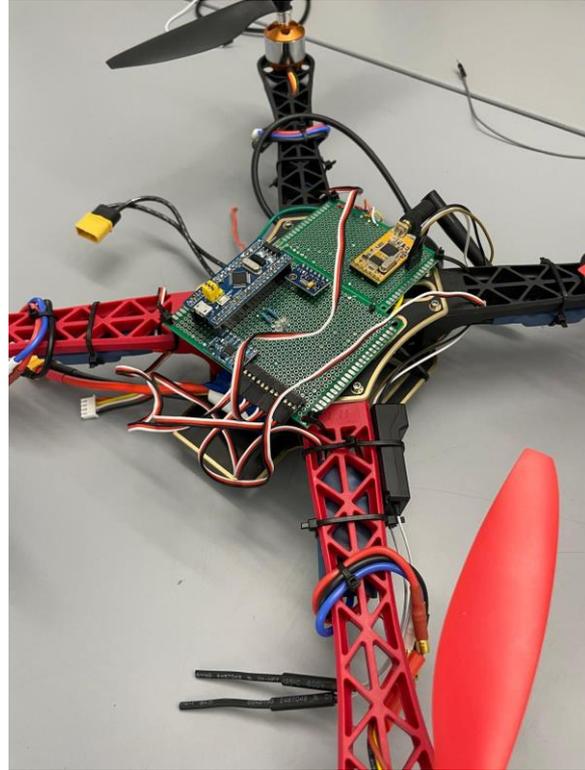


Illustration 2- Drone system

5. Conclusions

The development of an intelligent autonomous drone system for Texas Instruments faced several challenges, especially due to budget constraints that prevented the purchase of a professional drone from the beginning. Other complications in the project came from unforeseen problems (such as the breakage of an ESC, a propeller, and a motor during collisions) and a steep learning curve in the assembly of the hardware and the programming of the autonomous flight system. Despite these obstacles, of which one of the main learnings was how to better prepare for unforeseen events, the main objectives of Texas Instruments were met, who expressed their satisfaction with the results obtained in the project.

Regarding future work, several areas for possible improvements have been identified, including improving the battery technology to enhance the drone's autonomy, the integration of a more sophisticated internal positioning system, and implementing real-time diagnostics of the drone's flight data. In terms of safety, it should be noted that the integration of robust risk management systems known by the human operators in charge of handling the drone is imperative.

6. References

- [1] Brokking, R. (n.d.). YMFC-32 Auto Downloads. Retrieved from http://www.brokking.net/ymfc-32_auto_downloads.html

Índice de la memoria

Capítulo 1. Introducción	8
1.1 Motivación del proyecto.....	10
Capítulo 2. Estado de la Cuestión	12
2.1 Estado de la cuestión exclusiva de patentes	12
2.1.1 Planificación de ruta global de un vehículo de superficie no tripulado basada en un algoritmo A* mejorado.....	13
2.1.2 Robot de reparto premium BellaBot.....	14
2.1.3 Seguimiento de posición mediante varios receptores.....	15
2.2 Patentes.....	17
2.2.1 Vuelos de drones totalmente autónomos	17
2.2.2 Estabilización y navegación de un dron autónomo.....	18
2.2.3 Navegación autónoma de drones basada en computer vision.....	18
2.3 Conclusión.....	19
Capítulo 3. Definición del Trabajo	20
3.1 Justificación.....	20
3.1.1 Posibles desarrollos futuros	20
3.2 Objetivos iniciales	21
3.3 Objetivos adaptados	22
3.3.1 Hardware.....	22
3.3.2 Software.....	24
3.4 Metodología y desglose de tareas.....	25
3.4.1 Hardware.....	25
3.4.2 Software.....	26
3.5 Planificación y Estimación Económica	28
3.5.1 Planificación temporal	28
3.5.2 Estimación económica.....	29
Capítulo 4. Hardware: Componentes del sistema	32
4.1 Controlador de Vuelo	32
4.1.1 Microcontrolador STM32F103C8T6	32
4.1.2 Sensores.....	37

4.1.3	Transmisor y Receptor	42
4.1.4	Resistencias	49
4.1.3	LEDs	50
4.1.4	Placa de Prototipo.....	52
4.2	Estructura y Propulsión	55
4.2.1	Chasis	55
4.2.2	Controladores de velocidad	57
4.2.3	Motores.....	59
4.2.4	Hélices	62
4.3	Sistema de alimentación.....	65
4.3.1	Batería.....	65
4.3.2	Conector de Batería	67
4.3.3	Cargador de Batería.....	68
4.4	Sistema de Telemetría	69
4.4.1	Controlador autónomo	70
4.4.2	Comunicación Inalámbrica.....	72
4.5	Implementación	75
4.5.1	Sistema del dron	76
4.5.2	Sistema de telemetría.....	77
Capítulo 5. Software: Controlador de Vuelo autónomo		79
5.1	controlador de vuelo open-source	79
5.1.1	Control PID.....	81
5.1.2	Recepción de instrucciones no autónomas a través del transmisor y receptor de control remoto.....	82
5.2	Pseudocontrolador autónomo y ejemplos de caso de funcionamiento	84
5.2.1	Operador humano	84
5.2.2	Arduino UNO Clon.....	85
5.2.3	Dron.....	86
5.3	Propuesta para lograr autonomía mediante la comunicación con el Sistema de telemetría	87
5.3.1	Microcontrolador STM32F103C8T6	88
5.3.2	Arduino UNO Clon.....	92
5.4	Código para vuelo autónomo en la Arduino UNO Clon	94
5.4.1	Implementación	94

5.4.2	<i>Ejemplo de código de implementación</i>	95
5.4.3	<i>Interfaz gráfica de usuario</i>	96
5.5	Búsqueda de la ruta óptima en una fábrica.....	97
5.5.1	<i>Plano de una fábrica de Texas Instruments</i>	98
5.5.2	<i>Algoritmo estado-enlace: Dijkstra</i>	99
Capítulo 6. Análisis de Resultados		100
6.1	Evaluación de componentes hardware	101
6.1.1	<i>Método de evaluación</i>	101
6.1.2	<i>Resultados de las pruebas</i>	102
6.1.3	<i>Gestión de problemas</i>	102
6.2	Evaluación de subsistemas de hardware.....	102
6.2.1	<i>Microcontrolador STM32F103C8T6</i>	103
6.2.2	<i>Sistema de alimentación</i>	104
6.2.3	<i>Controlador remoto</i>	105
6.2.4	<i>ESCs</i>	106
6.2.5	<i>Sensores</i>	108
6.2.6	<i>Giroscopio- acelerómetro MPU-6050</i>	108
6.2.7	<i>Barómetro MS5611</i>	109
6.2.8	<i>Comunicación con el controlador de vuelo autónomo Arduino UNO Clon</i>	109
6.3	Evaluación del sistema completo	110
6.3.1	<i>Vuelos de prueba y problemas encontrados</i>	110
6.3.2	<i>Problemas persistentes de estabilidad</i>	112
6.4	Análisis del software	112
6.4.1	<i>Evaluación del controlador de vuelo open source</i>	112
6.4.2	<i>Integración del sistema de telemetría</i>	113
6.4.3	<i>Desarrollo del pseudocontrolador de vuelo autónomo</i>	113
Capítulo 7. Conclusiones y Trabajos Futuros		115
7.1	Conclusiones generales	115
7.2	Limitaciones afrontadas	116
7.3	Recomendaciones para trabajos futuros	116
7.3.1	<i>Investigación en tecnologías de baterías</i>	117
7.3.2	<i>Sistema de navegación sofisticado</i>	117

7.3.3 Mecanismos de carga y descarga de carga útiles.....	117
7.3.4 Sistemas escalables	118
7.3.5 Análisis de datos en tiempo real.....	118
7.3.6 Programas de entrenamiento y mejora de la interfaz de usuario	118
7.3.7 Desarrollo de un algoritmo de navegación.....	118
7.4 Más recomendaciones en materia de seguridad	119
7.4.1 Evaluación y gestión de riesgos	119
7.4.2 Protocolos de seguridad operacional.....	119
7.4.3 Redundancia y respaldo de sistemas.....	119
7.4.4 Monitoreo y diagnóstico en tiempo real.....	119
7.4.5 Mantenimiento preventivo	120
7.4.6 Integración de sistemas de detección y evitación.....	120
7.4.7 Políticas de seguridad y cumplimiento normativo	120
7.4.8 Capacitación y concienciación del personal.....	120
Capítulo 8. Bibliografía.....	121
ANEXO I: ALINEACIÓN DEL PROYECTO CON LOS ODS	125

Índice de figuras

Figura 1- Potador de materiales semiconductores proporcionado por Texas Instruments....	9
Figura 2- Ejecución simultánea de A* empezando por los puntos inicial y final de la ruta [4]	14
Figura 3- Detección de obstáculos de Bella Bot [6].....	15
Figura 4- Diagrama de pines de la Blue Pill [11].....	33
Figura 5- Conexiones eléctricas entre la Blue Pill y el adaptador FTDI [12].....	35
Figura 6- Velocidades de los motores para throttle, roll, pitch y yaw [20].....	44
Figura 7- Transmisor i6X y receptor IA6B [22].....	48
Figura 8- Control de throttle, roll, pitch y yaw [23].....	49
Figura 9- Información sobre el sistema dada por las LEDs [1].....	52
Figura 10- Montaje de la PCB.....	54
Figura 11- Placa con distribución de energía integrada.....	56
Figura 12- ESCs utilizados con los cuatro motores.....	58
Figura 13- Dirección de giro de los cuatro motores en un cuadricóptero [27].....	61
Figura 14- Hélices de plástico reforzado con nylon.....	63
Figura 15- Hélices de fibra de vidrio [28].....	64
Figura 16- 303S / 2200mAh / 30C LiPo Battery [30].....	66
Figura 17- Carga completa de la batería.....	69
Figura 18- Arduino UNO Clon [32].....	71
Figura 19- Transceptor en el sistema del dron.....	75
Figura 20- Diagrama de pines del sistema del dron [1].....	76
Figura 21- Sistema final de hardware adoptado.....	77
Figura 22- Conexión entre Arduino UNO y módulo transceptor APC220 [35].....	78
Figura 23. Centro de control de tierra.....	84
Figura 24- Sistema de telemetría.....	85
Figura 25- Sistema del dron.....	86
Figura 26- Plano de fábrica de Texas Instruments.....	98
Figura 27- Programa de test open-source [1].....	100

Figura 28- Prueba con LEDs	103
Figura 29- Prueba de alimentación [1]	105
Figura 30- Prueba del control remoto [1]	106
Figura 31- Prueba de vibración de motores [1]	107
Figura 32- Prueba de comunicación con sensores [1]	108
Figura 33- Prueba del acelerómetro [1]	109
Figura 34- Prueba del barómetro [1]	109
Figura 35- Se quemó el transistor central de un ESC durante la realización de pruebas de vuelo	111

Índice de tablas

Tabla 1- Especificaciones del entorno de operación del dron	24
Tabla 2- Cronograma temporal del proyecto.....	29
Tabla 3- Coste económico total del proyecto	31
Tabla 4- Funciones del centro de control de tierra	85
Tabla 5- Funciones del sistema de telemetría.....	86
Tabla 6- Funciones del sistema del dron	87

Capítulo 1. INTRODUCCIÓN

Los materiales semiconductores son componentes con propiedades eléctricas únicas que los sitúan entre los conductores y los aislantes. Son vitales en la industria tecnológica porque son utilizados en una multitud de aplicaciones electrónicas, tal como la fabricación de componentes como transistores, diodos y circuitos integrados, los cuales son la base de toda la electrónica moderna desde ordenadores hasta smartphones. Además, en la actualidad, son esenciales en el ámbito de la innovación y los avances tecnológicos. Por ejemplo, los semiconductores tienen un rol crucial en la fotoconductividad, por lo que tienen una gran importancia en el campo de las tecnologías renovables gracias a su aplicación en la fabricación de paneles solares [2].

Texas Instruments, es una empresa prometedora en la industria de los semiconductores, y está especializada en el desarrollo y fabricación de chips y semiconductores destinados a una variedad de aplicaciones tecnológicas. Entre ellas se encuentra la innovación y suministro para dispositivos que abarcan desde calculadoras, pasando por dispositivos móviles, hasta soluciones industriales a gran escala e incluso de automoción. Asimismo, Texas Instruments contribuye e innova en otros campos clave como el procesamiento digital de señales y la tecnología analógica.

En el contexto de la fabricación de materiales semiconductores, es esencial tener en cuenta la fragilidad y las exigentes condiciones de manejo que requieren éstos. Tradicionalmente, el transporte de estos materiales tan delicados es realizado a nivel de suelo, lo cual no solo consume espacio escaso y valioso, sino que también puede resultar en retrasos logísticos debido a la congestión. Por tanto, en este proceso donde la eficiencia y precisión son críticas, el aprovechamiento del espacio aéreo en instalaciones industriales (a menudo subutilizado), se presenta como una oportunidad para mejorar la eficiencia logística (optimizar procesos y reducir ineficiencias).

A pesar de que algunas entidades han empezado a agilizar la transferencia de materiales mediante el uso de vehículos terrestres autónomos, el uso de drones para un transporte vertical de materiales semiconductores tiene el potencial de transformar radicalmente el escenario presentado en el párrafo anterior donde se dan ineficiencias. Al permitir un flujo continuo y eficiente de materiales transportados por drones, se puede aumentar la productividad reduciendo cuellos de botella y tiempos muertos.

En la imagen siguiente se muestra el contenedor que utiliza Texas Instruments para el transporte seguro de grupos de semiconductores desde estaciones de prueba distintas de una de sus fábricas.



Figura 1- Potador de materiales semiconductores proporcionado por Texas Instruments

Además, hay estudios que apuntan a que la integración de sistemas autónomos en entornos industriales, en específico aeronaves no tripuladas (UAV), puede reducir los costes operativos de forma considerable, ya que dicha automatización minimiza errores humanos y maximiza la utilización de recursos [3].

Por estas razones, resulta imprescindible comenzar a desarrollar e implementar tecnologías de drones dentro de entornos industriales. Específicamente, en instalaciones de fabricación de semiconductores (que es en lo que se centra este proyecto de la mano de Texas Instruments); no solo para mejorar la eficiencia de procesos, sino para situar a la empresa como líder en innovación logística dentro del sector industrial.

1.1 MOTIVACIÓN DEL PROYECTO

En resumen, la iniciativa original de Texas Instruments de integrar drones en sus fábricas para el transporte de materiales semiconductores de una estación de prueba a otra se motiva por la necesidad de la optimización de las siguientes características:

- **Mejorar la eficiencia y optimización del espacio.** En las instalaciones de fabricación el espacio vertical es a menudo un recurso subutilizado.
- **Reducir costes operativos y riesgos.** La automatización del transporte de materiales sensibles con drones minimiza el riesgo de daños durante su manejo, lo cual puede reducir los costes asociados con su manejo manual.
- **Incremento de la productividad.** Los drones son capaces de operar de manera continua y en condiciones de congestión que se dan en los métodos de transporte convencionales.
- **Sostenibilidad y seguridad.** Los drones contribuyen a un entorno de trabajo más seguro y menos contaminante, porque representan una alternativa limpia y segura respecto a los vehículos motorizados tradicionales empleados en instalaciones de producción.
- **Innovación y liderazgo en el sector.** Esta tecnología innovadora puede servir como modelo para optimizar operaciones internas y de cadena de suministro, para otras empresas dentro o fuera de la industria.

En conclusión, los drones suponen una solución ágil para el transporte vertical porque permiten una reconfiguración rápida y eficaz del flujo de trabajo sin tener que interrumpir las operaciones del suelo (donde se encuentran los operadores humanos). Además, la

implementación de esta tecnología no solo se alinea con las tendencias globales de la industria hacia la automatización, seguridad y sostenibilidad, sino que coloca a la empresa a la vanguardia de la innovación en procesos de manufacturación y establece un precedente para futuras integraciones autónomas en la industria.

Capítulo 2. ESTADO DE LA CUESTIÓN

En este capítulo se desarrolla la investigación realizada sobre la tecnología existente que atañe al problema intriducido en el capítulo 1. Esto incluye información sobre patentes y no patentes, así como interpretaciones de la tecnología en su contexto de uso en el proyecto. Dado que el objetivo último del proyecto es el diseño de un sistema inteligente para un dron, la investigación presentada a continuación se centra en la autonomía de drones.

En cuanto a la investigación no incluida en patentes que podría ser útil, se encuentra un algoritmo de gestión de rutas, una visualización del entorno en tres dimensiones, y un método para rastrear el dron usando múltiples receptores. En cuanto a la investigación de patentes, se encuentran sistemas de dron totalmente autónomos, técnicas para estabilizar un dron autónomo durante el vuelo y la navegación de drones autónomos basada en computer vision.

Todos los documentos e información consultada acerca de sistemas robóticos autónomos se tuvieron en cuenta en todas las decisiones del proyecto, así como qué diseños evitar por ser perjudiciales para un entorno industrial. Las implementaciones existentes para este fin brindan información sobre los componentes necesarios en el software, los innecesarios (descubiertos en la aplicación práctica) y los flujos de diseño fundamentales para maximizar la eficiencia.

2.1 ESTADO DE LA CUESTIÓN EXCLUSIVA DE PATENTES

Se investigaron fuentes que no son patentes en la búsqueda de información relevante en relación con autonomía y seguimiento de drones mediante búsquedas en bases de datos e Internet. A continuación, se enumeran varios artículos que cuentan con información valiosa para diseñar un sistema de drones autónomos viable.

2.1.1 PLANIFICACIÓN DE RUTA GLOBAL DE UN VEHÍCULO DE SUPERFICIE NO TRIPULADO BASADA EN UN ALGORITMO A* MEJORADO

Este artículo describe un algoritmo de búsqueda de ruta mejorado desarrollado con el propósito de navegar grandes barcos a través de cuerpos de agua [4].

1. En primer lugar, se crea una cuadrícula bidimensional que representa un mapa del área por el que pueden moverse los barcos. Cada mosaico en el que se divide el área está coloreado de blanco o negro.
 - a. Un mosaico blanco representa un espacio que el barco puede ocupar.
 - b. Un mosaico negro representa un obstáculo que el barco debe evitar.
2. A partir de este mapa, se crea un grafo donde:
 - a. Los nodos o vértices son los mosaicos blancos.
 - b. Las aristas representan dos nodos adyacentes del grafo; con peso 10 los que están directamente al lado y con peso 14 si están conectados diagonalmente.
3. Posteriormente, se ejecuta el algoritmo A* sobre el grafo para encontrar la ruta más corta (a través de los mosaicos blancos), desde el mosaico que representa el punto de partida hasta el mosaico que representa el punto final.

El algoritmo A* está basado en Dijkstra con la heurística agregada de encontrar un orden más eficiente en el que explorar los nodos. Dado que el punto final se conoce de antemano, esta heurística para llegar a la ruta óptima se calcula como la distancia euclidiana entre el nodo explorado y el punto final.

Para optimizar aún más el algoritmo, se utiliza una búsqueda bidireccional en la que los puntos de inicio y fin sirven como puntos de destino entre sí; es decir, A* se ejecuta simultáneamente comenzando desde el punto inicial y final en paralelo, encontrándose finalmente en algún punto intermedio. Esto se puede observar en la imagen siguiente sacada del artículo.

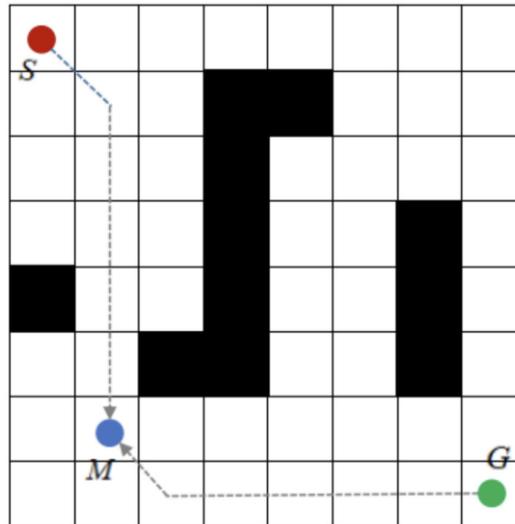


Figura 2- Ejecución simultánea de A* empezando por los puntos inicial y final de la ruta [4]

2.1.2 ROBOT DE REPARTO PREMIUM BELLABOT

Idealmente, cualquier sistema inteligente autónomo necesita un sensor o cámara que pueda detectar su entorno a la perfección, para así poder seguir su camino sin evitar choques con objetos inesperados en un entorno dinámico, mientras se mueve desde un punto inicial A hasta un punto final B. Con este objetivo en mente, *Pudu Robotics* desarrolló un robot camarero de restaurante que denomino *BellaBot*, cuyo éxito se basa en la utilización de una cámara de profundidad 3*RGBD para detectar y evitar objetos cercanos que podrían causar colisiones [5].

Dicha cámara 3*RGBD (la cámara de percepción tridimensional más potente hasta la fecha), otorga al robot inteligente una imagen en tres dimensiones de su entorno a tiempo real, lo cual le permite recibir retroalimentación continua de ésta para evitar choques con clientes y objetos del restaurante (como mesas y sillas), así como prevenir derrames y daños a la comida que transporta.

Como se ve en la imagen siguiente, los sensores 3D equipados en *BellaBot* y su chasis modular, le permite frenar en todos los ángulos y retroceder o modificar su ruta tan pronto se encuentra con algún obstáculo demasiado cerca. Además, cuenta con un diseño interactivo que hace muy fácil para un humano detectar el estado del robot e incluso

interactuar con él si fuese necesario, a través de un menú de opciones y botones intuitivos y fáciles de usar.



Figura 3- Detección de obstáculos de Bella Bot [6]

Por último, existe una plataforma denominada *PUDU CLOUD* que ofrece un gran soporte de servicio inteligente en la nube para robots Pudu como BellaBot. Esto permite un servicio de restauración óptimo; ya que permite que todos ellos estén conectados por la nube para su fácil gestión y mantenimiento.

2.1.3 SEGUIMIENTO DE POSICIÓN MEDIANTE VARIOS RECEPTORES

Uno de los componentes del proyecto a realizar se ocupa del seguimiento de la posición del dron, así como el envío de instrucciones a través de algún protocolo. En el artículo [7], se propone el seguimiento de la posición de un dron mediante el uso de varios dispositivos Bluetooth Low Energy (BLE).

El sistema propuesto en el artículo emplea varios receptores BLE para estimar la posición del dron objetivo mediante el cálculo de la intensidad de la señal recibida (Received Signal Strength Indicator, RSSI). Los primeros componentes de este sistema incluyen los siguientes:

1. **Recepción de señales BLE.** Hay cuatro receptores BLE, que captan las señales RSSI emitidas por el dron objetivo cuya posición se desea detectar. La fuerza de las señales recibidas se utiliza para calcular la distancia y el ángulo hacia el dron objetivo.
2. **Cálculo de la posición del dron basado en distancia y ángulo.**
 - a. **Distancia.** La intensidad de la señal RSSI decrece con la distancia entre cada receptor BLE y el dron objetivo. El cálculo de la distancia se basa en el modelo de propagación de la señal, que relaciona la pérdida inversa de señal con respecto a la distancia.
 - b. **Ángulo.** Se utiliza un método de triangulación para determinar el ángulo de llegada de las señales. Para ello, con tan solo tres receptores BLE es posible trazar líneas de posición desde cada uno de ellos hacia el dron en un plano bidimensional en el que la intersección entre ellas indica la posición del dron en el plano 2D.
3. **Algoritmos de seguimiento.** En el artículo se propone la implementación de dos algoritmos:
 - a. **Algoritmo de estimación básica.** La precisión de este algoritmo depende de la calibración de los receptores, además de dónde estén situados. Este algoritmo usa los datos de distancia y ángulo de cada receptor BLE para estimar iterativamente la posición del dron a través de técnicas de interpolación y triangulación.
 - b. **Algoritmo de memoria de movimiento.** Este algoritmo mejora su precisión al realimentarse con información sobre la trayectoria previa del dron objetivo. Gracias a que el algoritmo predice la posición futura del dron basándose en movimientos anteriores, puede ajustar sus estimaciones para ofrecer un seguimiento más preciso en tiempo real, especialmente en entornos donde es probable que las señales sufran interferencias. El algoritmo se puede implementar de dos formas:
 - i. **Combined Distance and Quality Assessment (CDQA).** Este algoritmo combina la predicción de la distancia con la evaluación

de la calidad de las señales recibidas, para ofrecer una estimación de posición más precisa.

- ii. **Adaptive Distance Correction Algorithm (ADCA)**. Este algoritmo tiene como objeto corregir las predicciones de distancia en tiempo real para tener en cuenta las variaciones en la calidad de la señal y el entorno físico en el que se encuentra el dron.

Finalmente, el artículo evalúa el rendimiento del sistema utilizando dos métricas: número promedio de movimientos necesarios para ubicar el dron; y la tasa de éxito de dicho seguimiento.

2.2 PATENTES

Esta sección del capítulo está dedicada a la discusión de la investigación del estado del arte relacionada con patentes. Concretamente, las siguientes patentes describen soluciones parciales para un algoritmo autónomo que pueda guiar a un dron de una estación de prueba a otra.

2.2.1 VUELOS DE DRONES TOTALMENTE AUTÓNOMOS

La patente [8] describe un sistema para drones autónomos que únicamente necesita la orden de empezar a volar para posteriormente buscar un tipo de lugar de aterrizaje, que ha de definirse a priori. Dicho lugar de aterrizaje puede ser cualquier cosa que elija el operador humano: la palma de la mano, o una plataforma de algún color.

La capacidad de encontrar un tipo de lugar de aterrizaje, en lugar de una ubicación concreta, otorga al sistema gran versatilidad. Por ejemplo, si se le ha indicado al dron que aterrice en plataformas pintadas de verde (en vez de seguir una ruta fija predeterminada), éste podría detectar plataformas verdes ocupadas y continuar su búsqueda hasta encontrar un lugar seguro donde aterrizar.

Sin embargo, hay parte de esta patente que convendría evitar. Por ejemplo, el sistema patentado corta sus sistemas de comunicación durante el vuelo para ahorrar energía. Esto no

se desea para el proyecto a desarrollar porque es conveniente que el sistema externo sea capaz de modificar la ruta del dron en tiempo real.

2.2.2 ESTABILIZACIÓN Y NAVEGACIÓN DE UN DRON AUTÓNOMO

La patente [9] discute métodos, sistemas y técnicas computacionales que, mediante el uso de sensores, permiten controlar la estabilidad de vuelo de un dron inmediatamente después de despegar. Inicialmente, la patente describe como el dron navega basado en su posición relativa a un objeto de referencia que se establece antes del despegue. Posteriormente, y tras recibir retroalimentación de la imagen del entorno capturada por los sensores, el dron es capaz de determinar el tamaño y distancia a objetos cercanos para modificar su trayectoria correctamente.

En relación con la estabilidad del vuelo, la patente describe cómo el dron determina la altura de vuelo óptima basada en condiciones del entorno. Por otro lado, profundiza en la navegación de un dron autónomo en entornos donde el GPS es inoperativo, el entorno informático de red que conecta servidores y dispositivos cliente, y la arquitectura del sistema utilizada para almacenar datos relevantes.

2.2.3 NAVEGACIÓN AUTÓNOMA DE DRONES BASADA EN COMPUTER VISION

Por último, la patente [10] analiza métodos para la navegación autónoma de drones, basada en distintos tipos de sensores y computer vision. Las ideas principales presentadas sobre un sistema inteligente de vuelo implican usar un tipo de navegación justo cuando el dron despegue y, después de un tiempo predeterminado, cambiar a un segundo tipo de navegación basada en la visión del dron. Un ejemplo de esto se presenta en la patente: tras despegar y realizar un reconocimiento exhaustivo de la cara de una persona, el dron se estabiliza y se dirige a una ubicación determinada a priori utilizando su visión.

Una vez el dron ha capturado suficientes imágenes e información de su entorno a través de sensores, la patente describe cómo se han de procesarse dichos datos. El sistema descrito en la patente emplea una red de dispositivos cliente y servidores de mensajería para intercambiar los datos y comandos necesarios para la navegación autónoma. Los

dispositivos clientes (como teléfonos móviles y ordenadores), se comunican con los servidores mediante APIs (Application Programming Interfaces).

2.3 CONCLUSIÓN

El estado de la cuestión presentado anteriormente sobre la tecnología existente sobre autonomía de drones podría ser utilizado en el contexto del problema propuesto por Texas Instruments: el diseño de un sistema de navegación de dron autónomo adaptado a la entrega de materiales semiconductores dentro de un entorno industrial. Esta investigación incluyó la búsqueda de varios artículos académicos y patentes, cuyas tecnologías proporcionaron importantes ideas para muchos aspectos del problema de diseño planteado.

Idealmente, el sistema tendría que combinar y refinar todas estas tecnologías para satisfacer las demandas únicas de eficiencia, fiabilidad y seguridad impuestas por Texas Instruments, por lo que queda un amplio margen para la innovación; como la optimización en la gestión de la carga del dron, el aumento de la precisión de navegación y la garantía de los más altos estándares de seguridad operativa.

Capítulo 3. DEFINICIÓN DEL TRABAJO

3.1 JUSTIFICACIÓN

Dado el estado del arte presentado en el capítulo anterior, queda claro que en la actualidad no hay ningún sistema, desarrollado e implementado, que emplee drones inteligentes específicamente creados para hacer llegar materiales semiconductores desde una estación de prueba a otra dentro de una fábrica.

Por ende, este proyecto proporcionará a Texas Instruments una guía respecto a su futura investigación e inversión en sistema de drones autónomos para realizar tareas logísticas en sus instalaciones.

- Si la recomendación final del proyecto fuese positiva, es probable que Texas Instruments decida invertir tiempo y recursos (tanto humanos como económicos) en profundizar en el tema y estudiar su viabilidad real.
- En cambio, unas conclusiones negativas probablemente llevarían a Texas Instruments a explorar objetivos diferentes, o bien a emplear una metodología diferente hacia el mismo objetivo.

En resumidas cuentas, a pesar de que el alcance y objetivos del proyecto están limitados a los intereses particulares de innovación tecnológica de Texas Instruments, merece la pena destacar cómo el problema del nulo aprovechamiento del espacio aéreo en entornos industriales, por muy relevante que sea, nunca se ha abordado en el contexto específico del transporte de materiales semiconductores como medio para reducir muchas ineficiencias logísticas como cuellos de botella en el transporte por tierra de materiales.

3.1.1 POSIBLES DESARROLLOS FUTUROS

En el futuro, esta línea de investigación podría tener implicaciones mucho más amplias, que excedieran los intereses de Texas Instruments a corto plazo. Por ejemplo, si

Texas Instruments lograrse desarrollar exitosamente soluciones innovadoras para el transporte eficiente de materiales en sus fábricas, esto podría llevar a mejoras en la productividad y eficiencia en la industria a nivel global.

En un mundo ideal, si el proyecto fuese un éxito total y se llegase a implementar el sistema de forma segura y sin fallos, esto contribuiría a hacer las fábricas más eficientes y seguras. Además, la investigación realizada por Texas Instruments podría conducir a avances en otros campos e industrias parecidas. Esto es debido a que la demanda de drones inteligentes capaces de realizar tareas autónomamente es muy alta en todos los sectores tecnológicos.

Asimismo, dada la creciente importancia de prácticas industriales sostenibles, es imperativo abordar el problema del desaprovechamiento del espacio aéreo en fábricas, desde un punto de vista sostenible. Dichas implicaciones se discuten más a fondo en el ANEXO I.

Para concluir este apartado, aunque el objeto de este proyecto sea abordar la eficiencia del transporte en el contexto de las fábricas de semiconductores para las necesidades particulares de la empresa Texas Instruments, el proyecto tiene el potencial de generar un impacto positivo en un contexto más amplio y general.

3.2 OBJETIVOS INICIALES

En una primera reunión con el representante de Texas Instruments, se definieron objetivos muy amplios y ambiciosos que posteriormente se fueron adaptando al presupuesto y otras limitaciones reales del proyecto.

- En primer lugar, la función principal del dron debía ser la de transportar materiales semiconductores entre diversas estaciones de prueba autónomamente. Es prioritario que el dron opere de la forma más segura posible, sin ningún tipo de intervención humana. Esto sería conseguido con un microcontrolador que determinase el movimiento del dron según un algoritmo apropiado. Sin embargo, teniendo en cuenta que el sistema autónomo podría fallar y constituir un peligro para tanto los

trabajadores de la fábrica, como para la integridad de los valiosos materiales semiconductores, el dron debía estar equipado con un sistema de anulación para forzarlo a tener un aterrizaje de emergencia.

- En línea con esto, el segundo elemento del proyecto sería un sistema de monitorización vía sensores que proporcionase diagnósticos sobre el dron, incluyendo, pero no limitándose a, la batería restante, ubicación y velocidad. Estos diagnósticos permitirían a un operador humano determinar cualquier anomalía sobre la actividad del dron, y tomar su control manual si fuese necesario.
- Por último, el tercer elemento del proyecto consistiría en el mecanismo de carga que sujetase el material semiconductor al dron.

3.3 OBJETIVOS ADAPTADOS

En reuniones posteriores con el representante de Texas Instruments, se acotó el alcance del proyecto y se especificaron detalles concretos con el fin de adaptar la solución al problema específico que Texas Instruments deseaba abordar en una de sus fábricas.

3.3.1 HARDWARE

El hardware del dron es un componente crítico del proyecto, pues debe cumplir con una serie de requisitos técnicos y operativos que garanticen su efectividad en el transporte de materiales semiconductores dentro de la fábrica. A continuación, se detallan los objetivos específicos que se acordaron con Texas Instruments para el hardware.

3.3.1.1 Velocidad del dron

En primer lugar, para que el sistema sea considerado un éxito por Texas Instruments, el dron ha de desplazarse a más de 1 m/s, que representa de forma aproximada la velocidad actual a la que los humanos transportan los materiales semiconductores por la vía terrestre.

3.3.1.2 Tamaño del dron

Las dimensiones físicas del dron han de ser aproximadamente 1 ft². Esto es para que sea lo suficientemente grande para transportar la carga de materiales semiconductores, y lo suficientemente ligero y pequeño para que pueda caber entre las aberturas de 2 ft en la fábrica, ya que es imperativo que el dron no vuele sobre las máquinas y utilice los caminos entre ellas.

3.3.1.3 Control del dron

El dron debe ser capaz de volar sin acción humana por medio de instrucciones autónomas del controlador de vuelo.

3.3.1.4 Batería del dron

Idealmente, la batería debe durar varias horas para minimizar el tiempo de inactividad debido a una batería insuficiente. Se considera que una autonomía de 10 minutos con carga es un éxito como prueba de concepto. Sin embargo, para su aplicación real habría que mejorar la autonomía significativamente.

3.3.1.5 Tamaño de la carga

Los portadores de semiconductores de Texas Instruments vienen en dos tamaños: 9 in² y 13 in². Con el objetivo de concretizar el alcance del proyecto, se decidió centrar el sistema únicamente en los portadores de 9 in².

3.3.1.6 Peso de la carga

El peso de la carga del portador contenedor de materiales semiconductores puede oscilar entre 1 y 5 libras. Para el alcance de este proyecto, el objetivo es diseñar un sistema con el mínimo de una libra en mente. Esto es porque cuanto más peso, mayor consumo de la batería y estrés en los motores.

3.3.1.7 Accionamiento manual

En caso de errores en el sistema autónomo de vuelo, así como de alguna de las otras métricas como batería restante o estado de los motores, un operador humano debería de

poder hacerse con el control manual del dron para realizar un aterrizaje de emergencia seguro.

En la tabla a continuación se resumen las especificaciones operativas que fueron clave para establecer los objetivos de hardware concretos del proyecto.

Parámetro	Especificación
Dimensiones	1 ft ²
Tamaño del portador de carga	9 in ²
Peso máximo de carga	1 lb
Velocidad	> 1 m/s
Duración de la Batería	20 minutos (sin carga), 10 minutos (con carga)
Entorno de Operación	Temperatura ambiente: 40-100 °F Altura de vuelo: 10 - 13 ft Ancho del camino de la instalación: 2 ft, no volar sobre las máquinas, usar caminos entre ellas de un ancho aproximado de 2 ft
Otras características	Control de vuelo manual

Tabla 1- Especificaciones del entorno de operación del dron

3.3.2 SOFTWARE

Por su parte, el software del dron es fundamental para garantizar su capacidad de navegación autónoma y su funcionamiento eficiente en un entorno industrial de Texas

Instruments. Los objetivos de software se centran en el desarrollo de algoritmos que aseguren la seguridad y precisión en el transporte de los materiales semiconductores.

3.3.2.1 Dimensiones de la ruta de navegación

Se espera que el controlador de vuelo autónomo sea capaz de asegurar una navegación segura en un espacio bidimensional con aperturas de 2 pies de ancho. La fábrica cuenta con una altura que garantiza una ruta segura a cierta altura, por lo que no es necesario preocuparse por una tercera dimensión en el algoritmo que calcule las rutas óptimas.

3.3.2.2 Seguridad y redundancia

El software también debe incluir medidas de seguridad para garantizar que el dron opere de manera confiable y segura. Esto podría incluir el desarrollo e implementación de sistemas de redundancia para los componentes críticos, o la capacidad de realizar aterrizajes de emergencia en caso de fallos del sistema.

3.4 METODOLOGÍA Y DESGLOSE DE TAREAS

La metodología y desglose de tareas constituyen una parte esencial de este capítulo, porque proporcionan una guía estructurada y en detalle sobre cómo se debe llevar a cabo cada fase del proyecto. Por ende, esta sección describe los métodos y procesos que se utilizan para alcanzar los objetivos del proyecto, así como proponer un desglose más detallado de las tareas individuales necesarias para su realización. Asimismo, con la determinación clara de la metodología y tareas, se optimiza la eficiencia en la consecución de los objetivos, y se establecen las bases para un seguimiento y control efectivos del progreso del proyecto.

3.4.1 HARDWARE

Para este proyecto, que supone el desarrollo de un dron autónomo para el transporte de materiales semiconductores en una fábrica, se requiere una cuidadosa selección y configuración del hardware. Los objetivos claves incluyen la selección de componentes de hardware compatibles entre sí y de alta calidad, el ensamblaje del hardware y la realización de pruebas para verificar su confiabilidad y funcionamiento.

3.4.1.1 Investigación y selección de componentes

Dadas las necesidades del dron para cumplir con las especificaciones deseadas por Texas Instruments, se deben evaluar alternativas de componentes y seleccionar los componentes óptimos basados en los requisitos técnicos del proyecto, así como en el presupuesto disponible.

3.4.1.2 Adquisición de componentes

Esta tarea involucra la realización de pedidos y la recepción de estos tras su encargo y envío. Se tratará de que el tiempo de entrega sea el mínimo posible.

3.4.1.3 Montaje del hardware

La primera fase del montaje del hardware involucra el chasis y la estructura del dron. Posteriormente, han de instalarse el resto de los elementos poco a poco (como sensores, ESCs y motores). Además, hay que montar y conectar el sistema de alimentación.

3.4.1.4 Pruebas iniciales

Las pruebas iniciales incluyen la verificación de las conexiones eléctricas y la calibración de los ESCs y rotores. Básicamente, la verificación del funcionamiento básico del dron.

3.4.1.5 Optimización y ajustes

Por último, hay que probar a volar el sistema completo y resolver los problemas que surjan de estabilidad. Ello puede conllevar realizar ajustes en la distribución del peso en la estructura del dron.

3.4.2 SOFTWARE

El objetivo en este caso es el desarrollo y adaptación del software necesario para que el dron sea capaz de volar de forma autónoma, mediante el uso de uno o más controladores de vuelo adecuados. El desglose de tareas se desarrolla a continuación.

3.4.2.1 Estudio del controlador de vuelo

En primer lugar, hay que investigar las opciones de controladores de vuelo open-source y escoger uno adecuado. Es importante que el hardware adquirido (en especial el microcontrolador) sea compatible con él, así como que sea posible descargar a un entorno de desarrollo dónde poder volcar el software del controlador en el microcontrolador del dron para realizar pruebas y modificaciones. También es importante el estudio del software escogido para entender cómo funciona cada uno de los drivers que lo componen.

3.4.2.2 Modificación del controlador de vuelo

En segundo lugar, hay que adaptar el código open source para permitir la recepción de comandos a través del sistema de telemetría externo, así como implementar las funciones de control autónomo.

3.4.2.3 Desarrollo del pseudocontrolador de vuelo en el segundo microcontrolador

El segundo microcontrolador forma parte del sistema de telemetría y en él se han de desarrollar las funciones para enviar comandos autónomos al microcontrolador del dron.

3.4.2.4 Implementación del sistema de telemetría

Para ello, hay que seleccionar un método de comunicación entre ambos microcontroladores y establecer una conexión fiable y robusta entre ellos. Esto depende de los componentes hardware escogidos y sus especificaciones concretas.

3.4.2.5 Integración y pruebas del sistema desarrollado

Hay que realizar muchas pruebas de vuelo para perfeccionar y ajustar iterativamente los comandos autónomos. Además, se debe evaluar el rendimiento del dron en diferentes escenarios de prueba.

3.5 PLANIFICACIÓN Y ESTIMACIÓN ECONÓMICA

Para llevar a cabo el desarrollo del proyecto de un dron autónomo para el transporte de materiales semiconductores en una fábrica de Texas Instruments, es vital contar con una planificación temporal detallada, así como una estimación económica precisa.

Por un lado, la planificación temporal establece un cronograma claro de las actividades a realizar, lo que permite una gestión eficiente del tiempo y los recursos. Por su parte, la estimación económica ofrece una visión general a los costes asociados al proyecto, asegurando que éste se mantenga dentro del presupuesto asignado, y de que se optimice el uso de los recursos disponibles.

3.5.1 PLANIFICACIÓN TEMPORAL

Como se ha mencionado anteriormente, la planificación temporal es fundamental para organizar y coordinar las distintas fases del proyecto presentadas en el apartado 3.4. El objetivo de dicha planificación es garantizar que cada etapa del proyecto se complete de manera oportuna, y asegurar que el trabajo se realiza dentro del plazo establecido. Por tanto, esta sección presenta un cronograma detallado que abarca desde la adquisición de componentes, pasando por la fase inicial del diseño, hasta la implementación y pruebas finales del dron autónomo.

Mes	Actividades de hardware	Actividades de software
Enero	<ul style="list-style-type: none"> Investigación de opciones y alternativas en el mercado. Selección y adquisición de componentes. 	<ul style="list-style-type: none"> Elección y estudio de un controlador de vuelo open-source.
Febrero	<ul style="list-style-type: none"> Recepción y verificación de los componentes comprados. Montaje inicial del chasis. 	<ul style="list-style-type: none"> Modificación inicial del código del controlador de vuelo. Desarrollo del pseudocontrolador de vuelo autónomo.
Marzo	<ul style="list-style-type: none"> Integración de componentes electrónicos. 	<ul style="list-style-type: none"> Implementación del sistema de telemetría.

	<ul style="list-style-type: none"> • Instalación de motores y sistema de propulsión. 	<ul style="list-style-type: none"> • Pruebas de comunicación y ajustes iniciales.
Abril	<ul style="list-style-type: none"> • Pruebas de vuelo y resolución de problemas. 	<ul style="list-style-type: none"> • Mejoras en el sistema completo. • Evaluación del rendimiento.
Mayo	<ul style="list-style-type: none"> • Ajustes y optimización final. 	<ul style="list-style-type: none"> • Ajustes finales.

Tabla 2- Cronograma temporal del proyecto

3.5.2 ESTIMACIÓN ECONÓMICA

Por su parte, la estimación económica es otra herramienta muy útil para evaluar los costes totales del proyecto. En esta sección se desglosan los costes de desarrollo (principalmente compuestos por gastos en hardware). La estimación económica proporciona una visión clara de la inversión requerida, así como permitir a Texas Instruments tomar decisiones detalladas en el futuro acerca de la asignación de recursos para la continuidad del proyecto. Además, la precisión en la estimación económica es esencial para evitar sobrecostes y asegurar la viabilidad financiera del proyecto.

Para la adquisición de las piezas, se contaba con un límite presupuestario de \$1000, que corrían a cuenta de The University of Texas at Austin. En primera instancia, se hizo un encargo por esta vía de todos los componentes de hardware necesarios para el proyecto. Sin embargo, este proceso implicaba realizar una solicitud de compra (mediante la formalización de un Excel) y obtener la aprobación expresa del tutor del proyecto, por lo que conllevaba bastante tiempo. Por esta razón, según iba avanzando el proyecto, se tuvieron que comprar piezas a cuenta personal (por ejemplo, reemplazos de componentes que se rompían y que debían reemplazarse lo antes posible).

En la tabla siguiente se puede ver el presupuesto total utilizado y qué piezas se compraron con dicho dinero.

Componente	Unidades	Vía de adquisición	Precio unitario (\$)	Coste total (\$)
Microcontrolador STM32F103C8T6	3	DigiKey	7.00	21.00
MPU-6050 (giroscopio-acelerómetro)	1	Amazon	6.50	6.50
MS5611 (barómetro)	1	Amazon	11.00	11.00
FlySky transmisor FS-i6x con receptor IA6B	1	Amazon	55.00	55.00
Juego de resistencias de película metálica de cerámica (600 PCS)	1	Amazon	6.00	6.00
PCB de prototipos de fibra de vidrio	1	Amazon	7.50	7.50
LEDs de 3 mm, rojo, amarillo, verde (150 PCS)	1	Amazon	7.00	7.00
Marco de tamaño 450 con placa de distribución de energía integrada	1	Speedy FPV	18.00	18.00
Motor de 1000kV / 10x4.5 props / combo ESC (4 PCS)	1	Speedy FPV	22.00	22.00
Motor de 1000kV / 10x4.5 props / combo ESC	1	Amazon	18.00	18.00
3S / 2200mAh / 30C batería LiPo	1	Amazon	30.00	30.00
Conectores de batería XT60	1	Amazon	12.00	12.00
Cargador de batería lipo 2S/3S	1	Amazon	12.00	12.00
Clon de Arduino UNO	1	Amazon	17.00	17.00
Módulos RF inalámbricos APC220 con convertidor USB	1	DFRobot	40.00	40.00

Adaptador FTDI	1	Amazon	6.00	6.00
Hélices CW/CCW 10x4.5	1	Amazon	12.00	12.00
			TOTAL	301.00

Tabla 3- Coste económico total del proyecto

Capítulo 4. HARDWARE: COMPONENTES DEL SISTEMA

La elección y adquisición de las piezas de hardware que conforman el dron constituye la primera etapa realizada en este proyecto compuesto. En las secciones siguientes, se detallan las especificaciones de dichas piezas y cómo todas ellas son necesarias para conformar el sistema final del dron, dadas las especificaciones detalladas en el capítulo 3.

4.1 CONTROLADOR DE VUELO

El controlador de vuelo representa el cerebro del dron. Su tarea es la gestión y coordinación de todas las funciones y componentes necesarios para asegurar un vuelo estable y seguro. Este sistema está encargado de interpretar las señales (autónomas o no) enviadas por un operador humano, de procesar las medidas de los sensores en tiempo real y de ejecutar los algoritmos de control básicos para ajustar la velocidad de los motores.

El núcleo de este sistema es el microcontrolador STM32F103C8T6, que procesa las medidas de los sensores y manda instrucciones a los motores del dron. Dichos sensores incluyen un acelerómetro, un giroscopio y un barómetro. Por otro lado, un transmisor y receptor de control remoto permiten a un operador humano controlar el dron de forma inalámbrica. Finalmente, hay componentes adicionales como resistencias, LEDs y una placa de prototipo (PCB) para el ensamblaje del circuito.

4.1.1 MICROCONTROLADOR STM32F103C8T6

El microcontrolador STM32F103C8T6 (conocido como *Blue Pill*) es un microcontrolador basado en el núcleo Arm® Cortex®-M3 que incorpora el núcleo RISC de 32 bits. Pertenece a una familia de microcontroladores caracterizada por su alta eficiencia y rendimiento en múltiples aplicaciones, y opera en un rango de frecuencias que llega a alcanzar los 72 MHz. Gracias a su arquitectura, proporciona una rápida capacidad de procesamiento, adecuada para tareas complejas simultáneas en tiempo real [11].

de diversas tareas simultáneamente, como el control de motores a la vez que se leen medidas de los sensores.

2. Capacidades de Memoria.

La Blue Pill ofrece opciones de 64 KB o 128 KB de memoria flash y 20 KB de memoria SRAM. Esto es mayor capacidad de memoria que la Arduino UNO, que cuenta con 32 KB de memoria flash y 2 KB de SRAM. Su gran capacidad de memoria permite a la Blue Pill almacenar programas más pesados y manejar una mayor cantidad de datos en tiempo real.

3. Periféricos y conectividad.

- a. Mientras que la Arduino UNO presenta grandes limitaciones en cuestión de interfaces de comunicación y número de periféricos, la Blue Pill ofrece una variedad mayor de funcionalidades (USB, UARTs, SPIs e I²Cs). Dichas interfaces y periféricos permiten una mejor conexión con sensores y actuadores.
- b. La Blue Pill cuenta con dos ADCs de doce bits y varios temporizadores. Esto implica que es capaz de realizar tareas complejas como la lectura precisa de sensores analógicos y el control preciso del tiempo para PWM. Ambas de estas tareas son fundamentales para el control de motores de un dron.

4. Consumo y gestión de energía.

Modos de ahorro de energía. La Blue Pill optimiza el consumo de energía para aplicaciones alimentadas por batería mediante modos avanzados de gestión de energía. Esto es un factor crítico para optimizar la eficiencia energética del sistema del dron y así maximizar el tiempo de vuelo.

4.1.1.2 Programación de la Blue Pill

En primera instancia, fue necesario utilizar un adaptador FTDI para programar la Blue Pill. A través de este método se instaló un bootloader para poder programar la Blue Pill directamente a través de un cable USB, lo cual es mucho más simple y rápido.

4.1.1.2.1 Programación inicial con FTDI

Inicialmente, se utilizó un adaptador FTDI para programar la Blue Pill. Como se ve en la imagen siguiente, este método requiere conectar el adaptador FTDI a los pines de la UART1 (PA9 y PA10 para TX y RX, respectivamente), y tener una línea +VCC y GND. Además, hay que configurar el puente BOOT0 en la posición 1 (jumpers amarillos con una línea a +VCC en la figura 5) para así poner el dispositivo en modo de programación, y permitir la carga del bootloader o programa.

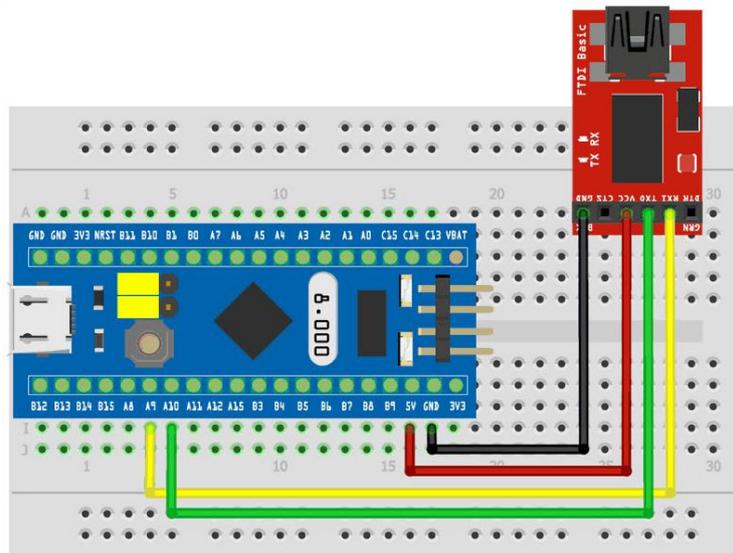


Figura 5- Conexiones eléctricas entre la Blue Pill y el adaptador FTDI [12]

4.1.1.2.2 Programación a través de USB

Posteriormente, para una programación más rápida y directa sin la necesidad de hardware adicional, se configura el bootloader del STM32 que permite la programación de la placa a través de su puerto USB. Para ello, se siguieron los pasos detallados en un artículo [13], y los cuales se resumen a continuación.

1. En primer lugar, es necesario instalar un bootloader que permita la comunicación USB. Este proceso se lleva a cabo usando el software *Flash Loader Demonstrator* de ST, que carga el bootloader en la placa. También hay que descargar el driver para Maple, para que la Blue Pill pueda ser programada a través de un puerto de la IDE de Arduino.

2. Una vez instalado, hay que configurar los puentes en la placa para activar el modo de programación USB (descrito en la sección anterior en la figura 5).

Se optó por este método no solo porque facilita enormemente el proceso de desarrollo, sino porque era necesario liberar la UART1 ocupada por el módulo FTDI para la conexión de los transceptores.

4.1.1.2.3 Configuración de la IDE de Arduino

El último paso para programar la Blue Pill fue configurar la IDE de Arduino. Para ello, hay que seguir los pasos detallados a continuación.

1. Primero se descarga la versión 1.8.3 de la IDE de Arduino para que sea compatible con el controlador de vuelo open-source utilizado [1]. Esto se hace a través de la página oficial de Arduino [14].
2. Después, hay que permitir a la IDE reconocer y programar placas STM32. Para ello en *Archivo* → *Preferencias*, se pega la URL del gestor de placas para las STM32 en el campo *Gestor de URLs Adicionales de Tarjetas*.
3. Además del paso anterior, hay que instalar los paquetes necesarios para añadir soporte a las placas STM32 en la IDE; en *Herramientas* → *Placa* → *Gestor de Tarjetas* y buscando *STM32*.
4. Habiendo completado los dos pasos anteriores correctamente, en *Herramientas* → *Placa*, ya se puede seleccionar el modelo específico de placa: *STM32F103C8T6*. También hay que seleccionar el puerto *Maple Mini* correspondiente; en *Herramientas* → *Puerto*.
5. A partir de este momento ya es posible escribir código, compilarlo (*Verificar*) y programar la placa (*Subir*).

4.1.2 SENSORES

4.1.2.1 Acelerómetro

Los acelerómetros son un tipo de sensor encargado en medir la aceleración, es decir, la tasa de variación de la velocidad. Los acelerómetros modernos utilizan principios físicos para convertir las fuerzas de aceleración en señales eléctricas que pueden ser analizadas. Por lo tanto, este dispositivo es esencial para detectar cambios en la orientación y el movimiento del dron en el que está integrado.

En cuanto a su principio de funcionamiento, un acelerómetro trabaja detectando la aceleración a través de la medición de la fuerza que actúa sobre una masa interna. Dicha masa se encuentra suspendida dentro del dispositivo y se desplaza cuando el sensor sufre una aceleración. Este desplazamiento se traduce en una señal eléctrica que puede ser analizada para determinar la magnitud y la dirección de la aceleración.

Por lo tanto, es posible modelar acelerómetro físicamente como una masa m suspendida por un resorte con una constante elástica k y un amortiguador con coeficiente de amortiguamiento c . La fuerza de aceleración externa F_a aplicada a esta masa produce un desplazamiento lateral (x). Por todo esto, la ecuación de movimiento de la masa es la siguiente:

$$kx + c \frac{\delta x}{\delta t} m \frac{\delta^2 x}{\delta t^2} = F_a$$

Según la segunda ley de Newton, la fuerza de aceleración externa y la aceleración a a la cual la masa queda sometida es la siguiente: $F_a = m \cdot a$

Reemplazando F_a con la ecuación de movimiento hallada anteriormente, es simple resolver la aceleración buscada a :

$$a = \frac{1}{m} (kx + c \frac{\delta x}{\delta t} m \frac{\delta^2 x}{\delta t^2})$$

Cabe destacar que en un sistema ideal sin amortiguamiento ($c = 0$) y con desplazamiento inicial nulo, la aceleración medida por el acelerómetro es directamente proporcional a la aceleración externa aplicada.

Este principio puede implementarse utilizando diferentes tecnologías, destacando el efecto piezoeléctrico, capacitivo o piezoresistivo.

- Los **acelerómetros piezoeléctricos**, utilizan cristales piezoeléctricos que generan una carga eléctrica cuando son sometidos a una fuerza. Dicha carga generada es proporcional a la fuerza de aceleración que hace que se compriman o estiren, permitiendo así medirla con buena precisión. Esta tecnología es muy eficaz para medir vibraciones y choques [15].
- Los **acelerómetros capacitivos** miden cambios en la capacitancia existente entre estructuras micromecánicas compuestas por dos placas dentro del dispositivo. Cuando una fuerza mueve dichas estructuras, la capacitancia cambia, permitiendo así medir la aceleración. Este tipo de acelerómetro es común en dispositivos MEMS (*Micro-Electro-Mechanical Systems*) y es especialmente indicado para sistemas en los cuales es vital una detección precisa de movimientos y orientación.
- Los **acelerómetros piezoresistivos** utilizan materiales cuya resistencia eléctrica cambia cuando son sometidos a una deformación causada por la aceleración (estrés mecánico). Esta variación en la resistencia es medida para determinar así la aceleración [16].

4.1.2.1.1 Acelerómetro Capacitivo

En este tipo de acelerómetros, el principio básico de funcionamiento es que el desplazamiento de la masa dentro del dispositivo causada por una fuerza de aceleración cambia la capacitancia entre las dos placas internas del acelerómetro. A su vez, dicha variación se convierte en una señal eléctrica. La capacitancia C entre dos placas paralelas genéricas separadas una distancia d y con un área A , dada la permitividad del material dieléctrico entre ellas (ϵ), se denota como:

$$C = \frac{\epsilon A}{d}$$

Cuando una fuerza de aceleración se aplica, cambia la distancia entre las placas, lo cual altera la capacitancia. Dicha variación puede medirse con precisión y convertirse a una señal eléctrica de voltaje. De forma análoga, la variación de capacitancia ΔC es proporcional al cambio en la distancia entre las placas Δd y puede ser medida para obtener la aceleración aplicada.

4.1.2.2 Giroscopio

Los giroscopios son dispositivos utilizados para medir y mantener la orientación y estabilidad. Esto es esencial para mantener la estabilidad durante el vuelo de un dron.

Cuando una rueda o rotor está girando, mantiene su orientación (momento angular) a menos que una fuerza externa (torque) actúe sobre él. Por ende, el principio de funcionamiento de un giroscopio se basa en la conservación del momento angular, cambios en el cual permiten detectar cambios en la orientación.

Entre los componentes principales que componen un giroscopio mecánico se incluye un rotor (soportado por aros ligeros conocidos como gimballes), que gira a alta velocidad. Los gimballes permiten que el rotor mantenga su orientación independientemente de la orientación del soporte externo, lo cual proporciona tres grados de libertad rotacional [17].

El momento angular L de un giroscopio depende del momento de inercia del rotor I y de la velocidad angular ω :

$$L = I \cdot \omega$$

Cuando se aplica un torque τ perpendicular a su eje de rotación, el giroscopio experimenta un cambio en la orientación de dicho eje. La velocidad a la que esto sucede (Ω_p) viene dada por la siguiente ecuación:

$$\Omega_p = \frac{\tau}{L}$$

Mediante la detección de cambios en la orientación del rotor, los giroscopios miden la velocidad angular. Existen varios tipos de giroscopios, en función de sus principios de operación y aplicaciones:

- Los **giroscopios mecánicos** se basan en la utilización de un rotor giratorio para mantener la orientación. Son los más utilizados hoy en día y destacan por su uso en navegación de aeronaves y misiles.
- Los **giroscopios ópticos** utilizan haces de luz y fibra óptica (en vez de partes móviles) para medir la rotación, lo cual los hace extremadamente duraderos y especialmente indicados para aplicaciones en sistemas modernos de navegación espacial.
- Los **giroscopios de rodamiento de gas** utilizan gas presurizado para suspender el rotor, lo cual reduce la fricción y permite una operación más silenciosa y precisa.
- Los **giroscopios vibratorios MEMS** funcionan con una masa vibrante interna que vibra a una frecuencia específica. Cuando el giroscopio sufre una rotación, la masa se desvía (debido a la fuerza de Coriolis). Dicha desviación es determinada por medio de sensores capacitivos, piezoeléctricos o piezoresistivos (integrados dentro del chip MEMS), y convertida en una señal eléctrica proporcional a la velocidad angular.

4.1.2.3 MPU-6050: acelerómetro y giroscopio

El MPU-6050 es un sensor que integra un acelerómetro y un giroscopio en un solo dispositivo utilizando tecnología MEMS (Micro-Electro-Mechanical Systems). Su capacidad para proporcionar datos precisos de aceleración y velocidad angular, junto con su tamaño pequeño y consumo de energía reducido, lo hace perfecto para mejorar la estabilidad, el control y la eficiencia del dron.

En este proyecto, el acelerómetro-giroscopio MPU-6050 se integra con el microcontrolador de vuelo mediante el protocolo I²C. Esto permite a la Blue Pill leer de forma constante y continua las medidas de movimiento y orientación, para ajustar la posición y estabilidad del dron en tiempo real.

4.1.2.3.1 Tecnología MEMS

Micro-Electro-Mechanical Systems (MEMS) es una tecnología que combina elementos mecánicos, eléctricos y electrónicos en un único chip de silicio. Por ende, esta tecnología hace posible la creación de sistemas muy pequeños que pueden llevar a cabo funciones complejas. En cuanto a sus características, destacan las siguientes:

- **Miniaturización.** Esta tecnología permite la fabricación de dispositivos complejos que pueden integrarse en espacios muy pequeños [18].
- **Integración.** La tecnología MEMS implica la combinación de múltiples funciones en un único chip, mejorando su eficiencia y reduciendo el tamaño y coste simultáneamente.
- **Precisión.** Los dispositivos MEMS pueden incluir sensores y actuadores con alta precisión y sensibilidad.
- **Bajo consumo de energía.** Ideal para aplicaciones portátiles que requieren eficiencia energética.

4.1.2.3.2 Acelerómetro del MPU-6050

El acelerómetro del MPU-6050 es de tipo capacitivo; es decir, mide la aceleración por medio de la detección de cambios en la capacitancia debido al movimiento de una masa interna suspendida. Los cambios en capacitancia son convertidos en señales eléctricas proporcionales a la aceleración aplicada. Los acelerómetros capacitivos tienen fama de ser muy precisos y estables, lo cual lo hacía ideal para su aplicación en el dron.

4.1.2.3.3 Giroscopio del MPU-6050

El giroscopio del MPU-6050 es de tipo vibratorio MEMS. Utiliza la vibración a una frecuencia específica de una masa interna para determinar la velocidad angular: cuando el giroscopio rota, la masa vibrante experimenta una fuerza de Coriolis que la desvía de su trayectoria. Acto seguido, la deflexión de la masa se mide y se transforma en una señal eléctrica que representa la velocidad angular.

4.1.2.4 Barómetro MS5611

El barómetro MS5611 es un sensor de presión que se utiliza en el controlador de vuelo de este proyecto para el mantenimiento de la altitud del dron. Este barómetro mide las variaciones minúsculas en la presión atmosférica para determinar el cambio de altitud. Es un dispositivo extremadamente sensible, ya que es capaz de detectar cambios de altitud con una precisión de hasta 10 cm [19]. Por lo tanto, es vital para el mantenimiento de la estabilidad y seguridad del dron durante el vuelo. Sin embargo, es importante proteger este sensor de la luz directa del Sol, ya que esta puede interferir en las medidas de presión.

En este proyecto, el barómetro MS5611 se integra con el microcontrolador de vuelo (al igual que el acelerómetro-giroscopio MPU-6050) mediante el protocolo I²C. Esto permite a la Blue Pill leer de forma constante y continua las medidas de presión, para ajustar la altitud del dron en tiempo real.

4.1.3 TRANSMISOR Y RECEPTOR DE CONTROL REMOTO

El conjunto de transmisor y receptor de control remoto permite la comunicación bidireccional entre el dron y el operador humano en la estación base. El transmisor, que cuenta con un par de joysticks, envía los comandos del operador humano al dron. Análogamente, el receptor a bordo del dron recibe dicha información y se la entrega al microcontrolador para que ajuste la velocidad de los motores.

El transmisor es el dispositivo utilizado por el operador humano para enviar comandos de movimiento al dron. Puede variar en términos de diseño, número de canales y tecnología de transmisión. Por otro lado, el receptor es el dispositivo a bordo del dron que recibe las señales del transmisor y las traduce en acciones.

En cuanto a las tecnologías de comunicación, la mayoría de los controles remotos de drones operan utilizando frecuencias propias de las ondas de radio (típicamente entre las bandas de 2.4 GHz o 5.8 GHz), ya que proporcionan un buen equilibrio entre alcance y resistencia a interferencias.

A pesar de que el objetivo final era que el dron funcionase autónomamente, se decidió comprar este par de componentes para probar que el dron funcionaba correctamente con el controlador de vuelo. Una vez, probado que el dron había sido montado correctamente y que era seguro, se podría prescindir de este elemento para el manejo de las instrucciones autónomas.

4.1.3.1 Manejo del dron en vuelo

En primer lugar, el throttle (acelerador) controla directamente la velocidad de los motores del dron, lo cual afecta a la altura y velocidad vertical del dron. Al incrementar el throttle, la velocidad de todos los motores aumenta simultáneamente, lo que causa que el dron ascienda hacia arriba. Análogamente, al disminuir el throttle, se reduce la velocidad de todos los motores, lo cual resulta en que el dron baje su altura. El throttle es fundamental para el despegue, aterrizaje y el mantenimiento de una altura deseada durante el vuelo.

En segundo lugar, el roll (alabeo) es el responsable de inclinar el dron de lado a lado, por lo que es esencial para el control lateral del mismo. Roll a la derecha implica que los motores en el lado izquierdo del dron giren a mayor velocidad; esto genera un inclinamiento del dron a la derecha, así como un desplazamiento lateral a la derecha. Por otro lado, roll a la izquierda tiene el efecto contrario: los motores del lado derecho giran a mayor velocidad para lograr un inclinamiento y desplazamiento lateral del dron a la izquierda. El control del roll es crucial para maniobrar a lo largo de los ejes horizontales, lo cual permite al dron ajustar su posición lateral para, por ejemplo, evitar obstáculos.

En tercer lugar, el pitch (cabeceo) es el responsable de que el dron se incline hacia adelante o hacia atrás. Pitch hacia adelante causa que los motores traseros giren a mayor velocidad que los delanteros, inclinando el dron hacia adelante, lo cual deriva en un movimiento hacia adelante. Por otro lado, el pitch hacia atrás resulta en que los motores delanteros giren a mayor velocidad, inclinando el dron hacia atrás, causando un movimiento hacia atrás.

Por ultimo, se encuentra el yaw (guiñada), que se encarga de rotar el dron alrededor de su eje vertical (hacia arriba o abajo), para cambiar la dirección a la que apunta el dron. En cuanto a su funcionamiento, el yaw hacia la izquierda aumenta la velocidad de los motores que giran en sentido horario y disminuye la velocidad de los que giran en sentido antihorario; esto resulta en una rotación hacia la izquierda. De forma complementaria, el yaw a la derecha aumenta la velocidad de los motores que giran en sentido antihorario y disminuye la velocidad de los que giran en sentido horario, lo cual resulta en una rotación del dron hacia la derecha. En cuanto a sus aplicaciones, el control del yaw es fundamental para ajustar la orientación correcta de las cámaras y/o sensores.

La combinación de estos cuatro controles permite a un operador humano la navegación precisa del dron en todas las direcciones en un espacio tridimensional. Cada uno de los controles interactúa con los otros para estabilizar el dron en la realización de maniobras, haciendo posible un vuelo eficiente, seguro y efectivo.

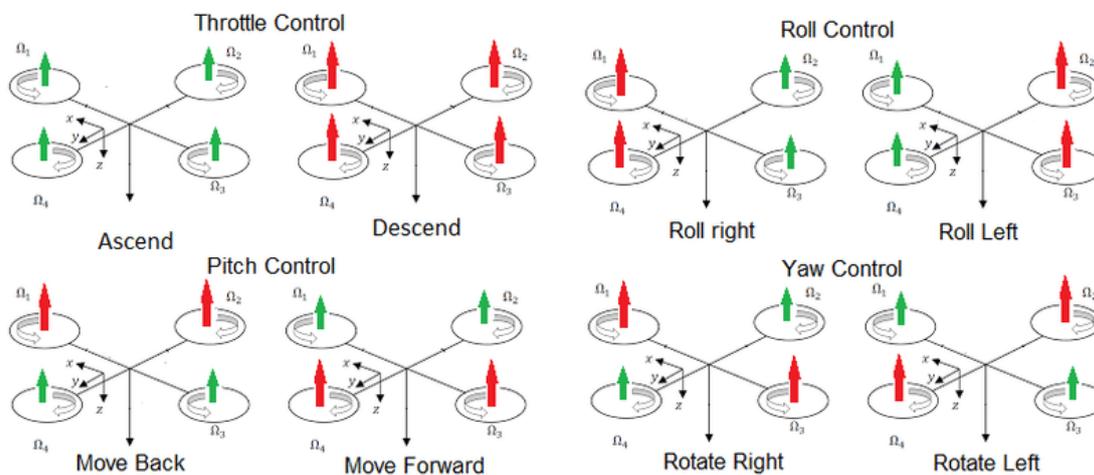


Figura 6- Velocidades de los motores para throttle, roll, pitch y yaw [20]

4.1.3.2 Transmisores y receptores multicanal

Un transmisor y receptor multicanal, en el contexto de los drones, hace referencia a un sistema de radiocontrol (RC) que hace uso de múltiples canales de comunicación para enviar y recibir diversos comandos de forma simultánea. Así, cada canal es capaz de

controlar una función específica del dron. En cuanto a su funcionamiento básico, este se divide en las labores llevadas a cabo por el transmisor (la parte del control remoto en manos del operador humano), y por las llevadas a cabo por el receptor (dispositivo en el dron que capta las señales enviadas por el transmisor).

- En cuanto a la transmisión de señales, el transmisor envía señales específicas a través de cada canal utilizando una frecuencia de radio determinada. Posteriormente, estas señales han de ser moduladas y codificadas para asegurar su correcta transmisión e interpretación por el receptor en el dron.
- Análogamente, en cuanto a la recepción y decodificación de dichas señales transmitidas captadas por el receptor, cada canal del receptor está sintonizado para recibir y decodificar las señales de su correspondiente canal en el transmisor. Una vez que las señales son decodificadas, son convertidas en comandos eléctricos que se envían al microcontrolador para que llegue a los diferentes sistemas actuadores del dron, como los controladores de velocidad electrónicos (ESCs).

4.1.3.3 Modulación por posición de pulso

La modulación por ancho de pulso (PPM) es el método más utilizado en la comunicación por medio de señales de radio. Es particularmente relevante en el campo de la operación de drones, por lo que merece la pena detallar su funcionamiento a continuación.

PPM es un tipo de modulación que codifica la información de múltiples canales en una señal única mediante la variación de la posición de los pulsos relativos a un marco fijo. En otras palabras, una señal PPM condensa la información de múltiples canales (throttle, roll, pitch y yaw) en una secuencia de pulsos incluidos en un único flujo de señal. La duración total de un marco PPM suele ser fija; por ejemplo, de 20 ms. Al contrario que la modulación de ancho de pulso (PWM), que requiere un cable separado para cada canal, PPM permite enviar todos los datos de los múltiples canales mediante un único cable, lo cual reduce significativamente la complejidad y necesidades de cableado.

En cuanto a su estructura, un marco PPM incluye un pulso de inicio, seguido por pulsos más cortos, cada uno representando un canal. La ‘posición del pulso’ se refiere al tiempo transcurrido entre el final de un pulso y el comienzo del siguiente, dentro de un marco de tiempo establecido. Cada uno de estos intervalos codifica un canal de control específico; por ejemplo, si dicha posición del pulso es más larga, esto indica un valor más alto para el comando de dicho canal. En cuanto a los beneficios de PPM para el control remoto de drones se encuentra:

- La simplicidad, ya que para el controlador de vuelo solo es necesario un cable para transmitir una única señal.
- Actualizaciones más rápidas, ya que los datos de todos los canales se transmiten de forma sincronizada y secuencial dentro de un mismo marco.
- Interferencias reducidas, ya que con menos cables hay menos posibilidades de interferencia entre señales de componentes eléctricos presentes en el dron.

En resumen, en el contexto de los drones, PPM es muy útil para comunicar instrucciones precisas a través del control remoto al microcontrolador. El controlador de vuelo interpreta las posiciones de pulso para determinar la velocidad deseada de los motores, el ángulo de inclinación o la dirección de rotación, entre otros.

4.1.3.4 Transmisor FlySky FS-i6x y receptor FS-IA6B

Las características importantes para el vuelo del dron que había que tener en cuenta en relación con este sistema eran las siguientes:

- **Número de canales.** Los transmisores varían en función del número de canales que ofrecen. Como es natural, un mayor número de canales permiten más opciones y grados de control sobre el dron.
- **Distancia de Alcance.** El alcance efectivo de dichos sistemas puede variar desde unos pocos cientos de metros hasta varios kilómetros en sistemas para pilotos más experimentados.

- **Duración de la batería.** Tanto para el transmisor como para el receptor es esencial una buena duración de la batería, especialmente para operaciones largas.
- **Características adicionales.** Existen varios transmisores avanzados que ofrecen funcionalidades extra como telemetría en tiempo real.

En cuanto a las alternativas populares en el mercado, se encuentran los siguientes sistemas transmisor-receptor de control remoto [21]:

- **Spektrum DX Series.** Este sistema es famoso por su fiabilidad. Está muy usado en la comunidad del aeromodelismo, ya que ofrecen múltiples modelos con diferentes características, como número de canales variable. Sin embargo, la otra cara de la moneda a un sistema tan confiable y con gran soporte es su alto coste.
- **FrSky Taranis.** Tiene una gran relación calidad-precio, es robusto, y ofrece la posibilidad de programación extensa a través de su software open-source. A pesar de su flexibilidad y posibilidad de personalización, la curva de aprendizaje para dicha programación es más empinada que para otros sistemas, por lo que no es la opción óptima para proyectos en los que el tiempo y la experiencia son factores limitantes.
- **FlySky FS Series.** Este sistema constituye una gran opción para proyectos principiantes o que tengan un presupuesto ajustado, debido a que ofrecen una funcionalidad básica adecuada para la mayoría de los usuarios no profesionales.
- **DJI Controllers.** Estos controles ofrecen una integración avanzada con funciones como control de cámara, así como ofrecer características de vuelo inteligente. Sin embargo, la compatibilidad entre éste y otros sistemas tiende a ser limitada y tienen un coste bastante alto.

Finalmente, se decidió utilizar el transmisor FS-i6X y el receptor FS-IA6B de FlySky, por varias razones que se desarrollan a continuación.

- En primer lugar y debido a que el coste era un factor altamente limitante en el proyecto, se decidió optar por un sistema de control remoto que proporcionase una buena relación calidad-precio. Esto fue crucial para mantener bajo el coste el proyecto sin comprometer la funcionalidad necesaria por el controlador de vuelo.

- En segundo lugar, el transmisor FS-i6X tiene una interfaz de usuario muy intuitiva, la cual facilita la lectura de la pantalla con el fin de adaptar fácilmente el controlador a diferentes configuraciones de vuelo o cambios en el comportamiento del dron.
- Por otro lado, una vez elegido el transmisor, la decisión del receptor era fácil, ya que la compatibilidad entre el transmisor FS-i6X y el receptor FS-IA6B asegura una comunicación continuada y fiable entre ambos aparatos.
- El sistema de control remoto cuenta con un rango de operación que excede las dimensiones de una fábrica
- En cuanto al número de canales, el transmisor proporciona hasta un máximo de 10 canales, lo cual supera la necesidad del proyecto de controlar los cuatro aspectos básicos de vuelo del dron (throttle, roll, pitch y yaw).



Figura 7- Transmisor i6X y receptor IA6B [22]

En conclusión, se escogieron estos dos componentes para el sistema de control remoto porque consigue un equilibrio entre funcionalidad, coste y facilidad para su operación. Reforzando lo anterior, el transmisor FS-i6X y el receptor FS-IA6B no solo cumplen con los requisitos técnicos del proyecto, sino que además aportan mucha flexibilidad para integrar el controlador de vuelo autónomo.

La imagen en la página siguiente ilustra cómo funciona la operación del transmisor FlySky FS-i6X.

- El throttle se maneja con el stick izquierdo, moviéndolo o bien hacia arriba o hacia abajo. Como ya se ha comentado, este control ajusta la velocidad de todos los motores simultáneamente, lo que afecta a la altura y velocidad vertical del dron.
- El otro componente que se controla con el stick izquierdo es el yaw, pero moviéndolo de derecha a izquierda. El yaw es crucial para determinar la dirección a la que se desea que apunte el dron.
- El pitch se controla con el eje vertical del stick derecho. Hace que el dron se incline y se mueva para adelante o hacia atrás.
- Por último, el roll se controla moviendo el stick derecho a la derecha o a la izquierda y permite el movimiento lateral, haciendo que el dron se incline hacia el lado derecho o izquierdo.

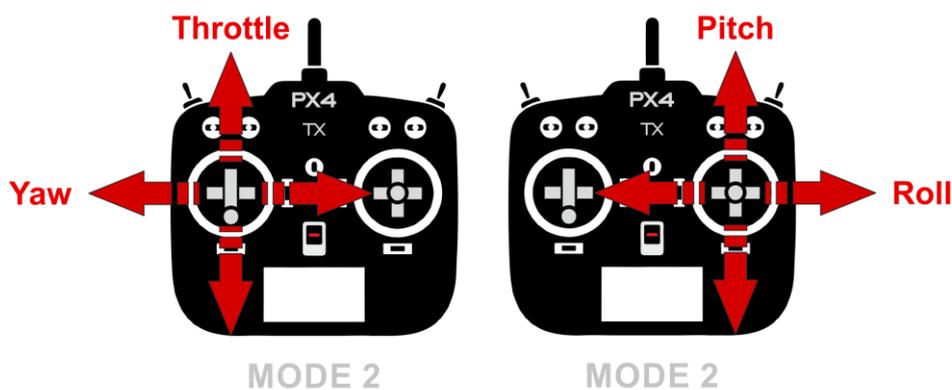


Figura 8- Control de throttle, roll, pitch y yaw [23]

4.1.4 RESISTENCIAS

El uso de resistencias en el proyecto garantiza que el sistema eléctrico funciona eficientemente y protege al microcontrolador y otros componentes sensibles manteniéndolos en rangos de tensión adecuados. Las resistencias tienen un coste muy bajo, con los beneficios siguientes:

- En primer lugar, se suelen utilizar para crear divisores de tensión que permiten al microcontrolador principal leer los niveles de tensión de la batería. Esto es importante para la monitorización de la energía restante del dron, lo cual evita que éste opere con la batería demasiado descargada.
- En segundo lugar, limitan la corriente que fluye a través de componentes eléctricos como LEDs y sensores, para evitar dañarlos con corrientes demasiado elevadas.
- Por último, a veces se usan para formar circuitos de control, asegurando que las señales que llegan al microcontrolador son estables y precisas.

4.1.4.1 Divisor de tensión

Éste es otro de los componentes esenciales para este proyecto. Un divisor de tensión se utiliza para reducir el voltaje de la batería a un nivel seguro para que sea manejado por el microcontrolador que contiene el controlador de vuelo.

Típicamente, dicho circuito está compuesto por dos resistencias en serie conectadas entre el voltaje de entrada (la batería) y tierra. El punto que conecta las dos resistencias es donde se toma la tensión de salida. La relación entre la tensión de entrada y la de salida es determinada por el valor de las dos resistencias, de esta forma:

$$V_{out} = V_{in} \cdot \frac{R_2}{R_1 + R_2}$$

En este caso, como el objetivo es dividir el voltaje de la batería por 11 (para asegurar que el microcontrolador recibiese una señal de voltaje segura y estable), se utilizaron resistencias de 1kΩ y 10kΩ.

4.1.3 LEDs

Para este proyecto, los *light-emitting diodes* (LEDs) jugaron un rol importante, sirviendo como dispositivos para visualmente comunicar el estado y los diversos modos de operación del dron. Esto es, proporcionan información útil tanto durante la configuración como durante el vuelo del dron en tiempo real.

Los dos LEDs de 3mm utilizados en el sistema fueron de color rojo y verde. El más importante para el proyecto fue el rojo, que daba información sobre el estado de la batería y de los sensores, entre otros. El LED verde se instaló también, pero daba información sobre el estado del GPS y el modo de vuelo del dron en el controlador de vuelo. Ninguno de estos elementos se implementó en la solución final del sistema propuesto.

Algunas de las principales señales visuales sobre los diversos estados y funciones del dron proporcionadas por los LEDs son las siguientes:

- **Calibración de sensores.** Durante la calibración del acelerómetro, el LED rojo parpadea en secuencias específicas para indicar el estado de dicho proceso.
- **Alertas de batería baja.** El LED rojo parpadea para indicar al operador humano que es crítico que la batería sea recargada.

- **Diagnóstico de errores.** En caso de no recibir respuesta de los sensores o del transceptor, el LED rojo parpadea de forma que se pueda identificar exactamente qué componente está fallando. Además, también indica otro tipo de errores, como por ejemplo que el stick de throttle no está en su posición más baja en el inicio del despegue.

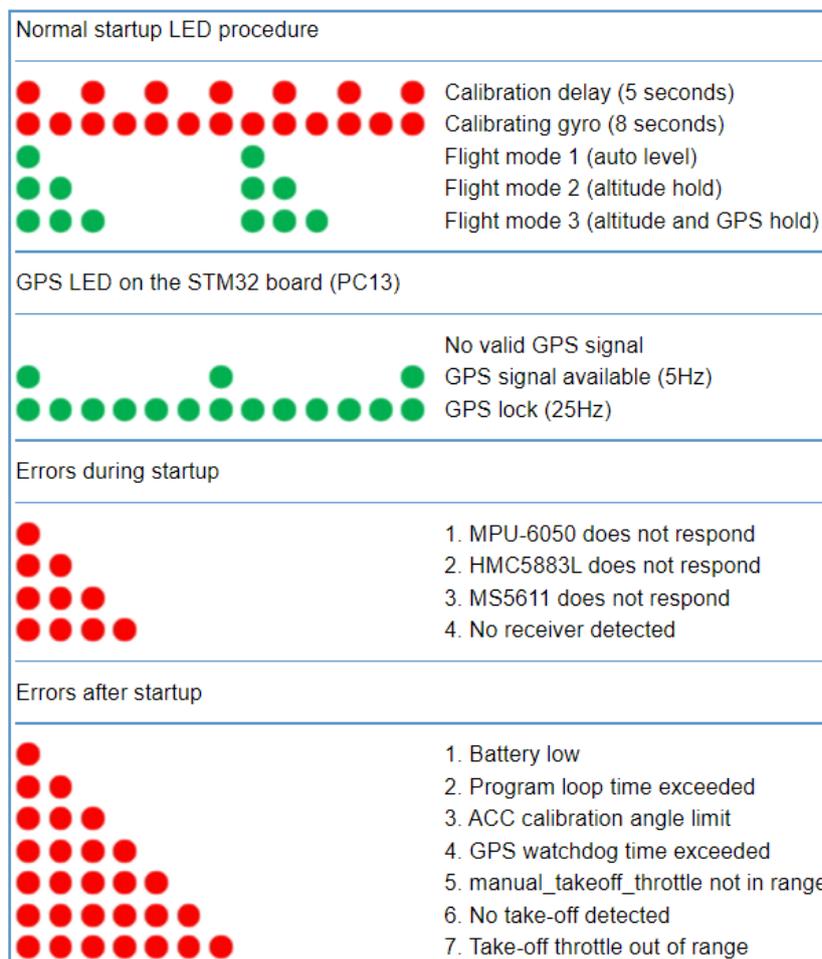


Figura 9- Información sobre el sistema dada por las LEDs [1]

4.1.4 PLACA DE PROTOTIPO

La placa de prototipo de fibra de vidrio, conocida comúnmente como *protoboard* o PCB (*Printed Circuit Board*), es un componente esencial para cualquier proyecto que incluya el diseño y construcción de circuitos electrónicos experimentales. En la construcción

del dron para este proyecto, se utilizaron dos PCBs de distintos tamaños conectadas entre sí, para el montaje y conexión de todos los componentes eléctricos necesarios.

Se compró un set que contenía PCBs de diversos tamaños. Se decidió utilizar una PCB grande de tamaño 7cm x 9cm, y otra más pequeña de 5cm x 7cm, ambas con un grosor de 1,6mm. Esto es debido a que alojar el módulo de telemetría en la PCB pequeña simplificó mucho el diseño del circuito.

La PCBs conjunta actúa como plataforma en la que se realiza el montaje y la conexión de todos los componentes electrónicos del dron. La conexión entre componentes y el correcto suministro de energía a todos ellos es realizada a través de cables, directamente soldados en la parte de atrás de la PCB. Algunos de los componentes que se alojan en la PCB son los siguientes:

- **Microcontrolador STM32F103C8T6.** El cerebro del dron en el que se aloja el software del controlador de vuelo
- **Sensores.** Cruciales para la estabilidad del dron, ya que proporcionan datos sobre su orientación
- **Resistencias.** Usadas para proteger el resto de los componentes de corrientes demasiado elevadas
- **LEDs.** Utilizados para notificar al usuario de batería baja, calibración correcta de sensores, etc.
- **Transceptor APC220.** Utilizado para llevar a cabo la comunicación inalámbrica entre el microcontrolador STM32F103C8T6 y la Arduino UNO). Montaje de la protoboard

Los pasos a seguir durante el proceso de montaje de la PCB son los siguientes:

1. Primero, hay que diseñar el esquema del circuito. Entre otras cosas, se decidió usar buses para conexiones comunes (por ejemplo, para la alimentación y tierra), lo cual simplificó mucho el diseño.

1. En segundo lugar, se colocan los componentes en la protoboard según el diseño establecido, y se aseguran con soldadura.
2. Tras el montaje en las dos PCBs, se unen entre sí con un par de piezas de cabezal macho soldadas en dos de los extremos de las PCBs.
3. Después, es esencial realizar una serie de pruebas para verificar que no hay cortocircuitos, y que las conexiones entre el microcontrolador STM32F103C8T6, los diversos sensores, y el transceptor APC220 son correctas.
4. Después, las PCBs se integran en la estructura del dron mediante abrazaderas de plástico que las sujetan sobre el chasis superior del dron.
5. Por último, hay que conectar los componentes electrónicos de la PCB con:
 - a. **Sistema de propulsión.** Esto se hace mediante conectores hembra-macho entre el circuito de la PCB y los ESCs.
 - b. **Sistema de alimentación.** A través de o bien los ESCs, o el conector USB del microcontrolador STM32F103C8T6.

Tras seguir todos estos pasos y la finalización del proceso de soldadura, se utilizan abrazaderas de plástico sobre el chasis superior del dron.

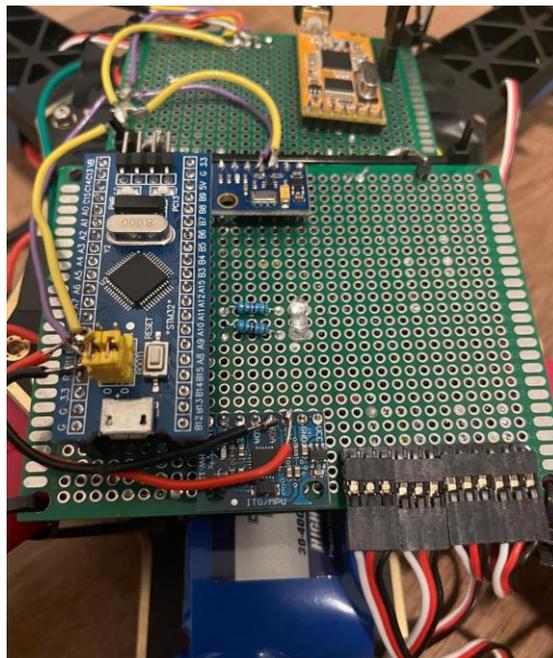


Figura 10- Montaje de la PCB

4.2 ESTRUCTURA Y PROPULSIÓN

El sistema de estructura y propulsión del dron es fundamental para el rendimiento y estabilidad del dron durante el vuelo. Por un lado, el chasis proporciona integridad estructural que soporta el resto de los componentes, los ESCs regulan la velocidad de los motores, los motores generan la fuerza de empuje, y las hélices transforman el empuje y potencia de los motores en movimientos controlados. El rendimiento, durabilidad y capacidad de respuesta del dron a condiciones de vuelo diversas dependen significativamente de una adecuada integración y equilibrio de estos cuatro elementos.

4.2.1 CHASIS

El chasis de un dron es crucial para su estructura y operación, ya que afecta directamente a la estabilidad, durabilidad y rendimiento del mismo. Algunas de las funciones y beneficios del chasis que se tuvieron en cuenta para la decisión de qué componentes comprar fueron los siguientes:

- **Estabilidad.** Una característica de un chasis bien diseñado es proporcionar una base estable y sólida para toda la electrónica del dron.
- **Rigidez.** La rigidez del chasis minimiza las vibraciones que pueden interferir con la estabilidad de vuelo o con las lecturas de los sensores.
- **Durabilidad.** Para un buen chasis, la resistencia a impactos es esencial; más aún para la construcción desde cero de un dron. Materiales tales como la fibra de carbono y plástico reforzado son adecuados para lograr este objetivo.
- **Gestión de calor.** El chasis debe de contar con un diseño modular que permita una adecuada ventilación y disipación del calor, para evitar el sobrecalentamiento de los componentes electrónicos dispuestos sobre él.

4.2.1.1 Placa con distribución de energía integrada

Para el proyecto, se decidió comprar y utilizar un chasis con una placa de tamaño 450, con 450mm de ancho y 55mm de alto y un peso de 270g [24]. Dicha placa tiene una distribución de energía integrada, lo cual proporciona grandes comodidades a la hora de la

integración con el resto de los componentes del dron En los siguientes párrafos se profundiza en las características del chasis mencionadas.

- El tamaño de 450 (medida en milímetros) se refiere a la distancia diagonal entre los motores opuestos. Se escogió un chasis relativamente grande debido a la capacidad de carga que debía tener el dron, establecida por Texas Instruments.
- En cuanto al material, la placa está hecha de materiales que combinan resistencia y ligereza: fibra de carbono (valiosa por su alta rigidez frente a baja densidad) y plástico reforzado. Esta combinación de materiales es ideal, ya que permite tener un dron ligero sin comprometer su fuerza estructural para portar cargas diversas.
- El hecho de que la placa cuente con una distribución de energía integrada facilita mucho la gestión del cableado relacionado con la distribución de energía. Además, reduce el riesgo de conexiones sueltas o mal configuradas.



Figura 11- Placa con distribución de energía integrada

4.2.1.2 Diseño del chasis

En cuanto a su diseño, los brazos del chasis tienen que montarse por separado de la pieza principal. Esto cuenta con varias ventajas, como que, si se rompe alguno de los brazos, éste puede ser fácilmente sustituido sin necesidad de sustituir todo el chasis. Por otro lado,

el diseño incluye orificios perforados y soportes para motores estándar, lo cual facilita la instalación y alineación precisa de éstos.

Como se ve en la figura 11, las placas tienen componentes metálicos. Se sospechó que esto podría estar causando cortocircuitos con la parte inferior de la PCB (que se colocó encima de la placa superior). Por ello, se cubrió dicha placa con cinta americana negra opaca antes de asegurar sobre ella la PCB soldada.

4.2.2 CONTROLADORES DE VELOCIDAD

Los controladores de velocidad electrónicos (ESCs) son elementos vitales en cualquier dron, ya que proporcionan un control preciso sobre la velocidad y dirección de cada motor. Las funciones y beneficios de los ESCs son los siguientes:

- **Control preciso del motor.** La forma en la que funcionan los ESCs es convirtiendo señales PWM del microcontrolador en comandos que ajustan de forma precisa la velocidad de los motores de forma conjunta (lo cual afecta al throttle), o de forma individualizada (para los controles de pitch, roll y yaw).
- **Protección y seguridad.** Los ESCs incluyen mecanismos de seguridad, como por ejemplo corte de bajo voltaje (LVC) y protección frente al sobrecalentamiento y cortocircuitos.
- **Calibración y sincronización.** La calibración de los ESCs es llevada a cabo para asegurar que todos los motores respondan de manera uniforme, sincronizada y estable a las señales recibidas del controlador de vuelo. Específicamente, el proceso de calibración asegura que todos los ESCs interpretan las señales PWM recibidas del microcontrolador de manera coherente.

En cuanto a las características que se buscan en ESCs para formar parte del proyecto se encuentran las siguientes:

- **Compatibilidad con los motores y la batería.** Los ESCs que se presentan en la figura 12 están diseñados para soportar 1000kV y una batería LiPo de 3 celdas. Esta

capacidad es suficiente para soportar los requisitos de los motores sin que estos se recalienten.

- **Capacidad de corriente.** Los ESCs deben ser capaces de gestionar corrientes elevadas durante maniobras intensas como el despegue. Cada uno de los cuatro ESCs



Figura 12- ESCs utilizados con los cuatro motores

tiene una capacidad de corriente nominal de 30A, es decir, adecuada para los motores instalados.

4.2.2.1 Implementación en el sistema del dron

Como se ha explicado anteriormente, los ESCs representan elementos esenciales en el sistema de propulsión del dron, ya que su garantizan un control preciso y seguro de los cuatro motores. Por lo tanto, la correcta instalación y calibración de los ESCs es muy importante, y su proceso se detalla a continuación.

1. **Conexión al controlador de vuelo.** Los cuatro ESCs se conectan a cuatro pines de salida digitales del microcontrolador STM32F103C8T6. Concretamente, se utilizan los pines B6, B7, B8 y B9 para los motores 1, 2, 3 y 4 respectivamente.
2. **Distribución de energía.** La energía que reciben los ESCs proviene de la placa de distribución de energía integrada del chasis. Cabe destacar que solo el primer ESC

conecta su línea de +5V al microcontrolador para suministrar energía al mismo. Los otros tres ESCs solo conectan sus líneas de señal y tierra. Es por esta razón por la que cuando se programa el microcontrolador mediante conexión USB a un ordenador, hay que desconectar la línea de +5V para que no haya conflicto de suministro de alimentación.

3. **Calibración de los ESCs.** La calibración de los ESCs se llevaba a cabo realizando en orden las siguientes instrucciones [25]:
 - i. Encender el Transmisor FlySky FS-i6x y poner el stick izquierdo arriba del todo (lo cual implica throttle al máximo).
 - ii. Alimentar todos los componentes del dron conectando la batería.
 - iii. Esperar a que los pitidos de calibración de los ESCs.
 - iv. Bajar el stick izquierdo al mínimo (lo que representa el throttle mínimo).

4.2.3 MOTORES

Evidentemente, los motores son críticos en este proyecto. Funcionan transformando energía eléctrica en movimiento rotacional, lo cual proporciona la fuerza necesaria para el despegue y el resto de las maniobras del dron mientras vuela. Las consideraciones para la selección de motores son las siguientes:

- **KV rating.** Esto se refiere al número de revoluciones por minuto por voltio, y es clave en la decisión sobre qué motores comprar. Generalmente, un motor de 1000kV proporciona una buena combinación entre velocidad y empuje, por lo que son ideales para drones de tamaño medio, lo que es el caso de este proyecto.
- **Capacidad de corriente.** Es imperativo que los motores seleccionados puedan manejar y operar de manera eficiente la corriente suministrada por los ESCs bajo las condiciones de carga esperadas durante el vuelo, sin que se produzca sobrecalentamiento.
- **Compatibilidad con las hélices.** Para maximizar la eficiencia y el empuje, el tamaño del motor debe ser compatible con las hélices.

4.2.3.1 Función de los motores en el dron

En primer lugar, el empuje que generan los motores a través de la energía eléctrica es la fuerza necesaria para que el dron despegue (que depende únicamente del throttle). Dicho empuje depende de tanto la velocidad de rotación de los motores como de las hélices utilizadas. En segundo lugar, los motores actúan de la mano de los ESCs y del controlador de vuelo para ajustar los otros tres controles del dron: roll, pitch y yaw. Esto es porque al modificar la velocidad de los motores del dron individualmente, es posible inclinar al mismo en diferentes direcciones, o rotarlo sobre su eje vertical. Por último, los motores han de responder velozmente a las señales del controlador de vuelo para mantener un vuelo estable y suave. Los motores brushless (sin escobillas), son indicados para esto debido a su rápida y precisa respuesta. Finalmente, se escogieron motores brushless de 1000kV debido a su alta eficiencia, durabilidad y capacidad para proporcionar el empuje y precisión para realizar vuelos estables y precisos.

4.2.3.2 Motores brushless

Algunas de las características de los motores sin escobillas frente a los motores con escobillas es que son muy eficientes, tienen mayor durabilidad, y poseen mayor potencia y rendimiento:

- Los motores brushless son más eficientes, energéticamente hablando, porque no tienen fricción interna causada por las escobillas.
- Además, al carecer de escobillas que sufren desgaste con el tiempo, tienen una vida útil más larga, además de necesitar menor mantenimiento.
- Por último, los motores brushless de 1000kV son capaces de generar una velocidad de rotación de 1000 revoluciones por minuto por voltio aplicado. Esta es una gran velocidad que no compromete la eficiencia del sistema, y es ideal para drones de tamaño medio.

4.2.3.3 Conexión de los motores a los ESCs y cambio del sentido de giro

Cabe aclarar que los cuatro motores quedaron configurados de forma que se formase un cuadricóptero; es decir, dos motores girando en sentido horario y dos en sentido antihorario. Esta disposición permite equilibrar las fuerzas de torsión que actúan sobre el dron.

Para conectar los motores a los ESCs y fijar el sentido de giro de todos ellos, se han de seguir los siguientes pasos [26]:

1. **Identificar y conectar los cables de los componentes entre sí.** Cada ESC tiene tres cables de entrada, mientras que cada motor tiene tres cables de salida. Estos cables suelen estar identificados con colores, aunque depende del fabricante. En primera instancia, no hay necesidad de seguir un orden específico, ya que el sentido de giro de los motores (CW o CCW) se puede invertir posteriormente.
2. **Fijar las conexiones.** Para asegurar que las conexiones estén firmemente aseguradas, se usan conectores de bala.
3. **Prueba inicial.** Después de conectar los cuatro motores a sus correspondientes ESCs, se encienden los motores y se identifica si giran en sentido horario (CW) o antihorario (CCW).
4. **Invertir el sentido de giro de los motores si es necesario.** Para ello, simplemente se han de intercambiar dos de los tres cables que conectan el motor al ESC.

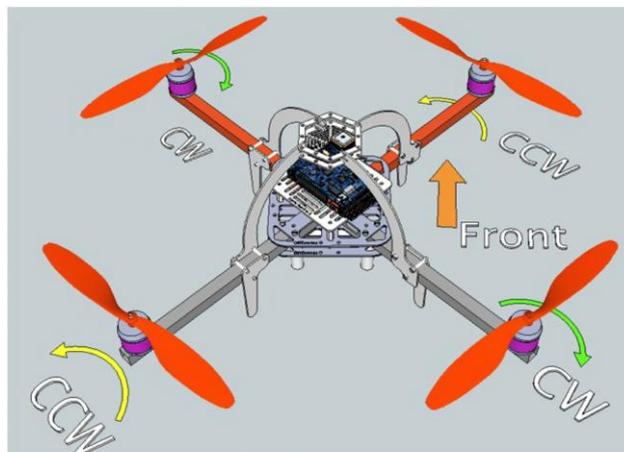


Figura 13- Dirección de giro de los cuatro motores en un cuadricóptero [27]

En cuanto a su montaje, hubo que asegurarse que los motores estuviesen correctamente alineados, para así evitar vibraciones no deseadas durante el vuelo.

Por último, antes del primer vuelo se realizaron una serie de pruebas, junto con la calibración de los ESCs, para verificar que los cuatro motores, tanto individualmente como conjuntamente, respondían correctamente a las instrucciones del controlador de vuelo.

4.2.4 HÉLICES

Las hélices son las palas giratorias que proporcionan empuje para el vuelo del dron. Específicamente, el rol de las hélices en el dron es convertir la potencia rotacional generada por los motores en empuje vertical, lo cual permite al dron que se eleve, descienda, o que se desplace en diferentes direcciones. Su diseño y tamaño tienen un impacto importante en el rendimiento y eficiencia del dron, por las razones que se detallan a continuación.

- **Diseño y material.** Las hélices pueden estar compuestas de diversos materiales:
 - Las hélices de plástico son más baratas y fáciles de reemplazar si se rompen, pero son menos duraderas y resistentes.
 - Las hélices de nylon combinan durabilidad y coste eficiente, ya que están hechas de un material más resistente al plástico estándar, lo cual las otorga un mejor rendimiento bajo tensión.
 - Las hélices de fibra de carbono son las más ligeras y robustas, pero más caras.
- **Tamaño y paso.** En cuanto al tamaño de la hélice, éste se mide en pulgadas y hace referencia a diámetro total de la misma cuando está en rotación. Por otro lado, el paso de la hélice es la distancia que ésta avanzaría en el transcurso de una revolución completa si estuviera en un medio sólido. Mayor paso implica mayor empuje, pero por el contrario requieren mayor potencia por parte del motor.
- **Número de palas.** Las hélices pueden tener dos, tres o incluso más palas. Cuantas más palas, el vuelo será más suave y estable, pero el sistema se vuelve menos eficiente por la acrecentada resistencia de aire.
- **Configuración y sentido de giro.** En el caso de un dron con cuatro motores, dos hélices giran en sentido horario (CW) mientras que las otras dos lo hacen en sentido

antihorario (CCW). Esta configuración cancela las fuerzas de torsión sobre el dron, lo cual hace que se estabilice.

Vista la cantidad de elementos a considerar, la selección de hélices más adecuadas para el proyecto depende fundamentalmente del tipo de motores escogidos, del objetivo de que el dron sea lo más ligero posible y de que el coste total se mantenga el más bajo posible. Por ello, se compraron hélices de dos palas de 10 x 4.5 pulgadas, que ofrecen un buen balance entre empuje y eficiencia. En cuanto al material del que están hechas, las hélices son de plástico reforzado con nylon, lo cual supone una opción rentable pero duradera y resistente.



Figura 14- Hélices de plástico reforzado con nylon

Sin embargo, tras una de las colisiones, una de las hélices se rompió y se reemplazó con una hélice del mismo tamaño, paso y número de palas. La única diferencia era el material, ya que la nueva hélice era de compuesto reforzado de fibra de vidrio. Cabe destacar

que, además, está hélice venía equilibrada de fábrica, por lo que no hubo que seguir el proceso de balanceo de rotores descrito en el apartado 5.2.4.1.



Figura 15- Hélices de fibra de vidrio [28]

En cuanto a su montaje, fue fundamental que las hélices se instalasen con el sentido de giro adecuado. Además, hubo que balancear las hélices para evitar vibraciones que pudiesen afectar a las lecturas de los sensores y a la estabilidad general del vuelo. Dicho proceso se explica a continuación.

4.2.4.1 Balanceo de hélices y rotores

El término ‘rotor’ hace referencia al conjunto completo de componentes giratorios de un sistema motriz que permita el vuelo: hélice y motor. Los rotores se pueden clasificar como de empuje (que proporciona la fuerza necesaria para levantar el dron), o de control (que ayuda a estabilizar y maniobrar el dron).

Los rotores o hélices desequilibradas pueden causar vibraciones excesivas. Esto puede afectar negativamente al rendimiento de los sensores, provocando lecturas erráticas o desgaste prematuro, así como respuestas inestables del dron. El proceso del balanceo de un único rotor es realizado utilizando cinta adhesiva como medio de ajuste, y siguiendo los pasos presentados a continuación [29].

1. **Fase de preparación.** Se montan las hélices en los motores y se ajustan muy bien utilizando alicates.
2. **Prueba inicial.** Se enciende el dron y se sujeta firmemente el brazo de la hélice que se está balanceando. Se aumenta el throttle progresivamente con el transmisor FS-i6x, y se observan las vibraciones del rotor.
3. **Aplicación de cinta adhesiva.** Se coloca un cuadrado pequeño de cinta adhesiva (en este caso, cinta negra americana) cerca de la punta de dicha hélice, en el borde opuesto a la vibración.
4. **Repetición de la prueba.** Se vuelven a probar las vibraciones y estabilidad del rotor. Hay que continuar ajustando el tamaño y posición de la cinta adhesiva hasta que las vibraciones se reduzcan al mínimo

Este proceso hay que repetirlo para los cuatro rotores, hasta que las cuatro hélices estén balanceadas correctamente. Durante el proceso iterativo, se hizo uso de un programa de prueba en el microcontrolador [1], que permitía controlar cada motor de forma individualizada y medir la vibración del rotor. El objetivo es que las vibraciones estén por debajo de 100, y que disminuyan a medida que aumenta el throttle.

4.3 SISTEMA DE ALIMENTACIÓN

El sistema de alimentación de un dron es fundamental para su operación, ya que suministra la energía necesaria para hacer funcionar los motores y el resto de los componentes eléctricos. Ha de garantizar un rendimiento óptimo y la mayor autonomía posible para el dron. El sistema de alimentación del proyecto está compuesto por tres componentes: la batería, el conector de la batería, y el cargador de la batería.

4.3.1 BATERÍA

La batería es el componente principal del sistema de alimentación, ya que es la directamente responsable de suministrar energía a todos los demás sistemas del dron. La elección adecuada de la batería es importante para asegurar una buena combinación entre autonomía de vuelo, buen rendimiento y seguridad operativa.

Para este proyecto, se seleccionó una batería de polímero de litio (LiPo). Este tipo de baterías son muy utilizadas en el mundo de los drones porque tienen alta densidad energética y pueden suministrar corrientes muy altas, manteniéndose relativamente ligeras. El peso del dron fue un factor muy importante para este proyecto, por lo que una batería de este tipo supone la mejor opción.

4.3.1.1 Especificaciones de la batería escogida

La batería escogida para el proyecto es una batería LiPo de tres celdas (3S) con una capacidad de 2200mAh y una tasa de descarga de 30C. Ofrece una combinación óptima entre voltaje, capacidad y tasa de descarga.



Figura 16- 303S / 2200mAh / 30C LiPo Battery [30]

Más específicamente, las razones para su elección se desglosan a continuación.

- Cada celda de una batería LiPo tiene un voltaje nominal de 3.7V. Por ende, una batería de 3S tiene un voltaje total de $3 \cdot 3.7V = 11.1V$. Este voltaje es adecuado para alimentar todos los componentes electrónicos del dron, en especial los motores.
- La capacidad de 2200mAh denota la cantidad de carga que se puede almacenar en la batería. Esta capacidad fue suficiente para proporcionar una autonomía de vuelo de entre 15 y 20 minutos, dependiendo de la carga que llevase el dron y de otras condiciones de vuelo.

- En cuanto a la tasa de descarga, 30C implica que la batería puede suministrar una corriente 30 veces su capacidad: $2200\text{mAh} \cdot 30 = 66\text{A}$. Lo más importante es que esta tasa de descarga sea suficiente para suministrar la energía necesaria a los motores durante maniobras intensas como el despegue.

4.3.1.2 Consideraciones de seguridad

La gestión adecuada de la batería (carga, descarga y almacenamiento) es esencial para mantener su rendimiento y seguridad al largo plazo.

- En cuanto al almacenamiento, las baterías LiPo deben almacenarse a una tensión intermedia de aproximadamente 3.8V por celda para prolongar su vida útil.
- Por otro lado, es importante utilizar cargadores específicos para baterías LiPo que incorporen características de balanceo de celdas, para mantener la uniformidad de voltaje entre las celdas. Además, hay que evitar situaciones peligrosas como sobrecargas o descargas profundas.
- Finalmente, es necesario inspeccionar de forma regular las baterías para identificar posibles daños internos y prevenir riesgos, como por ejemplo daños físicos o fugas.

4.3.2 CONECTOR DE BATERÍA

El conector de batería también forma parte del sistema de alimentación porque es el elemento que proporciona una conexión segura y eficiente entre la batería y el resto del sistema del dron.

4.3.2.1 Conector XT60

En este proyecto, se utiliza el conector XT60 para conectar la batería LiPo de 3S y 2200mAh a la placa de distribución de energía integrada del chasis. Sus características y beneficios se exponen a continuación.

- En primer lugar, el conector XT60 está diseñado para el manejo de corrientes altas de hasta 60A sin sobrecalentarse o perder eficiencia, lo cual es común para la

demanda energética que suponen los motores funcionando a plena capacidad en un dron.

- En segundo lugar, en cuanto a su seguridad y confiabilidad, el conector XT60 tiene un diseño robusto a prueba de chispas; esto asegura una conexión firme y segura entre la batería y el sistema del dron que minimiza el riesgo de que se suelte accidentalmente debido a vibraciones o movimientos bruscos durante un vuelo. Además, está fabricado con materiales de alta calidad, por lo que es muy resistente al desgaste que suponen múltiples ciclos de desconexión.
- Por último, los conectores XT60 son muy fáciles de conectar y desconectar, lo cual facilita el reemplazo rápido de la batería sin tener que cambiar todo el sistema de alimentación.

4.3.3 CARGADOR DE BATERÍA

Finalmente, el cargador de la batería mantiene la batería del dron en buen estado y garantiza su seguridad y eficiencia operativa. Para este proyecto se selecciona un cargador para baterías LiPo de 2S/3S. Su elección se basa en los factores siguientes.

- En primer lugar, el cargador elegido es compatible con la batería LiPo de 3S y 2200mAh del dron.
- En segundo lugar, el cargador asegura que las baterías se carguen de forma segura, ya que está equipado con múltiples protecciones; por ejemplo, protección contra sobrecalentamiento (que está relacionado con riesgos de incendio o explosión asociados con baterías LiPo), sobrecarga y cortocircuito.
 - Contiene indicadores LED que proporcionan una forma visual de monitorizar el estado de la carga que muestran claramente cuándo la batería está completamente cargada o de si hay algún problema durante la carga.

- Incluye la capacidad de balancear las celdas de la batería LiPo; es decir, asegura que las tres celdas alcancen el mismo nivel de voltaje. Esto es importante para cuidar del rendimiento y salud de la batería a largo plazo.

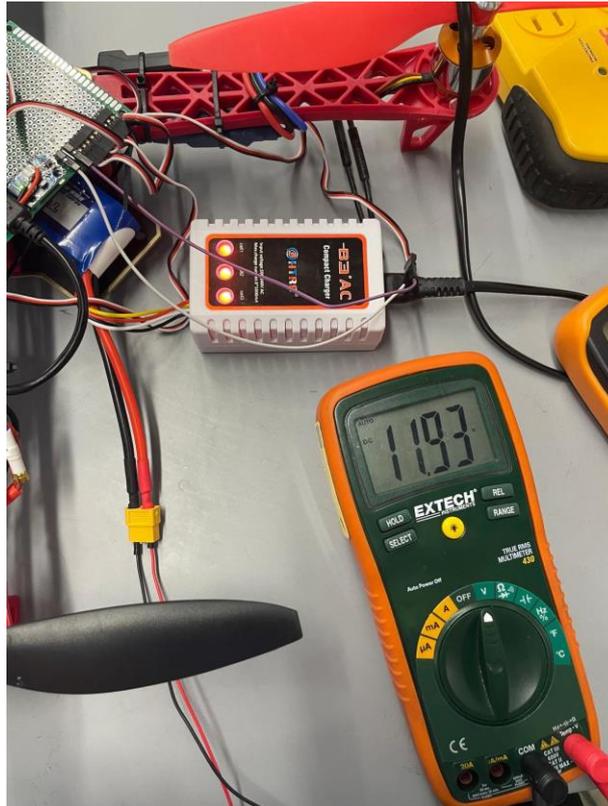


Figura 17- Carga completa de la batería

4.4 SISTEMA DE TELEMETRÍA

La telemetría es un elemento esencial en los sistemas de control de drones porque permite la transmisión de datos entre el dron y la estación base en la que se encuentran los operadores humanos en tiempo real.

En un principio, el plan era utilizar el sistema de telemetría para enviar datos diagnósticos sobre el dron, como la velocidad, altura y batería restante, mediante un par de transceptores APC220. Dichos datos se habrían de mostrar en una placa LCD conectada a una Arduino

UNO. Por otro lado, las instrucciones autónomas se debían enviar al microcontrolador STM32F103C8T6 directamente a través de una red Wi-Fi.

Sin embargo, debido a un conflicto de pines en el microcontrolador (que no tenía suficientes puertos seriales disponibles), y a la limitación de la Arduino UNO de no poder enviar y recibir datos simultáneamente, se decidió priorizar el envío de instrucciones autónomas entre las dos tareas mencionadas en el párrafo anterior. Es por esta razón por la que en las secciones siguientes se explica cómo el par de transceptores APC220 y el controlador Arduino UNO se utilizan en el proyecto para transmitir las instrucciones autónomas pertinentes al dron. Por su parte, el desarrollo de un sistema de diagnóstico se ha dejado en manos de mejoras futuras.

4.4.1 CONTROLADOR AUTÓNOMO

La Arduino UNO es un microcontrolador muy versátil y con una amplia comunidad de soporte basado en el microchip ATmega328P [31]. La elección de una clonación del microcontrolador Arduino UNO se justifica principalmente por las razones que se comentan a continuación.

- **Compatibilidad y flexibilidad.** El microcontrolador Arduino UNO es compatible con una variedad de sensores y módulos, lo cual implica que sea fácil integrarlo en el sistema del dron.
- **Comunidad y soporte.** El hecho de que haya una extensa comunidad de usuarios de Arduino (que implica que hay abundantes recursos, tutoriales y soporte técnico), resulta favorable para la programación de la placa y la solución de problemas a la hora de programar el controlador de vuelo autónomo.
- **Bajo coste.** Por último, la clonación de Arduino UNO es una opción muy económica, ya que dicha placa cuenta con las mismas prestaciones y funcionalidad que una Arduino UNO, pero a un coste más bajo.

En este proyecto, un microcontrolador Arduino UNO Clon de la marca ELEGOO se utiliza para alojar el microcontrolador de vuelo autónomo, según las necesidades de los operadores humanos trabajando en la fábrica, así como transmitir dichas instrucciones autónomas al dron.



Figura 18- Arduino UNO Clon [32]

El software de dicho sistema se desarrolla más en detalle en el capítulo 5, pero consta de dos partes bien diferenciadas.

1. Transforma en instrucciones concretas el deseo del operador humano de que el dron transporte una caja de semiconductores desde un punto de inicio hasta una estación de prueba destino.
2. Envía dichos comandos procesados al microcontrolador central STM32F103C8T6 situado en el dron, que se asegura de que las instrucciones autónomas sean seguidas con precisión por medio de los actuadores correspondientes.

4.4.1.1 Programación de la Arduino UNO clon con un portátil y la Arduino IDE

En primer lugar, se descargó en el portátil (desde la página oficial de Arduino [4]) la última versión del entorno de desarrollo integrado para Arduino (Arduino IDE) hasta el momento (versión 2.3.2). El IDE de Arduino es una herramienta esencial para escribir, compilar y cargar programas en la placa usada Arduino UNO Clon de ELEGOO.

4.4.1.1 Conexión, configuración y programación

En cuanto a la conexión, es tan simple como utilizar un cable USB para conectar la Arduino UNO Clon al portátil. Para configurar el IDE, hay que seleccionar Arduino UNO como tipo de placa (pulsando *Herramientas* → *Placa* → *Arduino UNO*) y seleccionar el puerto del portátil al que la placa está conectada en el portátil (*Herramientas* → *Puerto*).

Para programar la placa, en primer lugar, hay que escribir el código en el espacio proporcionado para ello en el IDE, y compilarlo para asegurar que no hay errores (pulsando el botón *Verificar*). Una vez verificado que el código es correcto, se presiona el botón *Subir* para cargar el código compilado. Los LEDs de la placa dejan de parpadear cuando este proceso es completado con éxito.

Una vez correctamente cargado el código, se puede usar el *Serial Monitor* con la misma velocidad en baudios que la especificada en el código para recibir y enviar mensajes, así como realizar la depuración del programa.

4.4.2 COMUNICACIÓN INALÁMBRICA

Los transceptores APC220 operan utilizando ondas de radio en la frecuencia de 433 MHz y son ideales para proyectos en los que la fiabilidad y autonomía son factores importantes, porque facilitan la transmisión de datos a través de distancias considerables. Concretamente, estos dispositivos pueden enviar información de forma bidireccional hasta 1000 metros. Además, el hecho de que utilicen una frecuencia de 433 MHz los hace menos susceptibles a sufrir interferencias de otros dispositivos que operan en bandas de frecuencia más saturadas, como por ejemplo la de 2.4 GHz.

4.4.2.1 Alternativas

En el mercado existen varias alternativas para la transmisión inalámbrica de datos, como los módulos ZigBee, NRF24L01 y Bluetooth.

- Los módulos ZigBee, a pesar de ser una opción robusta para muchas aplicaciones de comunicación, presentan ciertas limitaciones:

- **Rango y escalabilidad.** Su alcance típico no supera los 100 metros en entornos abiertos, lo cual limita su aplicación en muchos proyectos con requisitos de alcances de operación más altos.
- **Velocidad de transmisión y capacidad de datos.** La velocidad de transmisión de ZigBee no excede los 250 kbps, lo cual no es suficiente para el control a tiempo real de un dron.
- **Interferencias y confiabilidad.** ZigBee opera en la banda de libre uso de 2.4 GHz, la cual está muy saturada por diversos dispositivos, lo cual aumenta el riesgo de interferencias.
- **Costo y complejidad.** Aunque ZigBee ofrece una configuración de red flexible y es capaz de manejar múltiples nodos, tiene una implementación compleja y costosa.
- NRF24L01 son unos módulos populares para la comunicación inalámbrica de corto alcance. No obstante, se encontraron las limitaciones siguientes al evaluar su idoneidad para este proyecto [33]:
 - **Alcance.** Típicamente puede alcanzar hasta 100 metros, pero requiere una línea de vista sin obstrucciones. Esto es insuficiente para un dron destinado a operar en una fábrica.
 - **Interferencia.** Al igual que ZigBee, operan en la banda de 2.4 GHz, lo cual presenta el mismo problema acerca del riesgo de interferencias.
 - **Complejidad y configuración.** La configuración y la gestión de la red con módulos NRF24L01 es bastante compleja, especialmente cuando es necesario el manejo de múltiples nodos y asegurar una comunicación fiable y estable prolongada.
- El Bluetooth presenta ventajas en aplicaciones de corto alcance y bajo consumo de energía. Sin embargo, se consideró que no era adecuado para este proyecto por las siguientes razones:
 - **Alcance.** El alcance de un módulo de Bluetooth es de aproximadamente 10 metros (versión clásica). Dicho alcance se ve incrementado hasta 100 metros para la versión Bluetooth Low Energy (BLE). Como ya se ha mencionado

anteriormente, este alcance suele no ser suficiente para muchas operaciones de un dron.

- **Ancho de banda y velocidad.** Con respecto al resto de módulos comentados, Bluetooth cuenta con un ancho de banda y velocidad de transmisión más bajo. Esta es una gran limitación para la tarea de controlar los movimientos del dron de forma continua y a tiempo real.
- **Interferencia.** Al igual que ZigBee y NRF24L01, al operar en la banda de 2.4 GHz, el Bluetooth es susceptible a interferencias de otros dispositivos.

4.4.2.2 Elección de los Transceptores APC220

Tras evaluar diversas alternativas, se concluyó que los transceptores APC220 constituían la opción óptima para el sistema de telemetría. A continuación, se destacan varias razones clave por las que se decidió comprarlos.

- **Alcance extendido.** Los transceptores APC220 cuentan con un alcance de hasta 1 km. Esto supera con creces los otros tres módulos considerados. Este alcance es importante para mantener una comunicación estable y fiable con el dron, incluso a largas distancias.
- **Frecuencia y robustez.** Operan en la banda de 433 MHz, que está menos congestionada que la banda de 2.4 GHz. Como ya se ha mencionado, esto es un gran punto a favor porque reduce las interferencias y mejora significativamente la fiabilidad y estabilidad de la comunicación. Además, esta frecuencia es capaz de penetrar obstáculos, lo cual para un dron destinado a operar en un entorno dinámico dentro de una fábrica es muy positivo [34].
- **Simplicidad de uso.** Este módulo es conocido por ser fácil de usar y de configurar. Por ejemplo, no requieren ningún tipo de módulo adicional ni una compleja configuración de red. Su integración con la Arduino UNO Clon y la comunicación unidireccional hacia el microcontrolador es sencilla de instalar, directa y eficiente.

- **Fiabilidad y robustez.** Los transceptores APC220 ofrecen una alta fiabilidad en la transmisión de datos. Esto es un factor muy importante para asegurar que las instrucciones autónomas se transmitan sin pérdidas y en tiempo real.

En resumidas cuentas, los transceptores APC220, gracias a su alcance extendido, baja interferencia y facilidad de uso, son la mejor opción para garantizar un sistema de telemetría confiable y robusto para el sistema del dron.

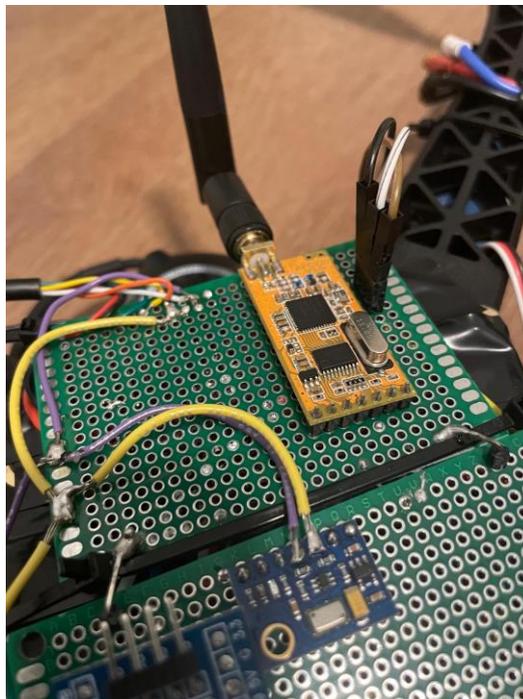


Figura 19- Transceptor en el sistema del dron

4.5 IMPLEMENTACIÓN

La implementación del sistema del dron y del sistema de telemetría son componentes cruciales en el desarrollo del proyecto. A continuación, se detallan ambos sistemas y sus configuraciones correspondientes.

4.5.1 SISTEMA DEL DRON

El sistema del dron está compuesto por varios componentes electrónicos que trabajan en conjunto para proporcionar estabilidad y control al dron. La siguiente imagen muestra el esquema de conexión del dron. Cada componente está interconectado para asegurar conexiones seguras y estables, especialmente en los puntos de soldadura, para evitar fallos durante el vuelo.

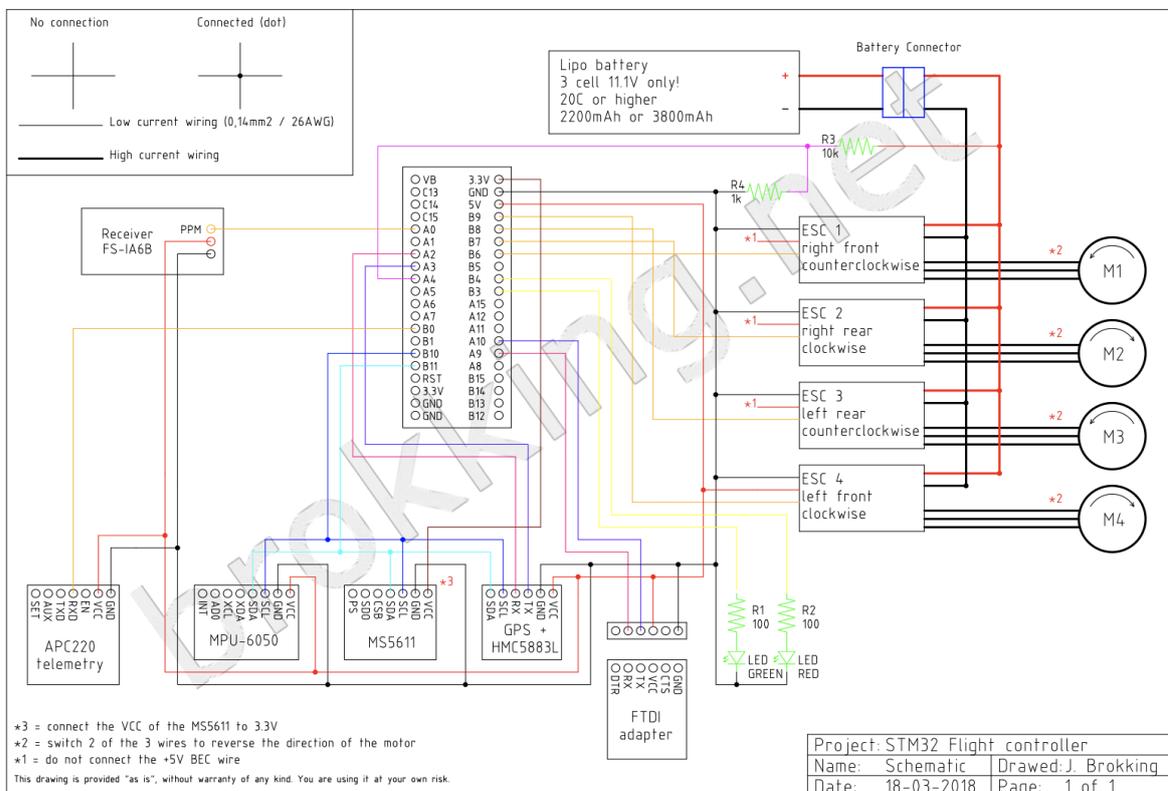


Figura 20- Diagrama de pines del sistema del dron [1]

Sin embargo, en el diseño final se prescinde del adaptador FTDI (ya que se logró programar el microcontrolador STM32F103C8T6 mediante una conexión USB), y se coloca

en su lugar un transceptor APC220, ya que dichos pines (A9 y A10) son para la comunicación serial por medio de la UART1.

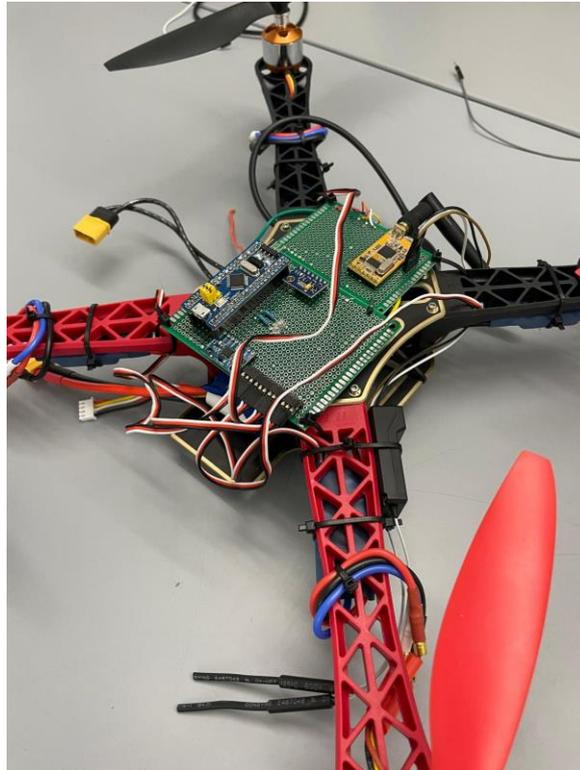


Figura 21- Sistema final de hardware adoptado

4.5.2 SISTEMA DE TELEMETRÍA

El sistema de telemetría está compuesto por los siguientes componentes:

- Un transceptor APC220 que permite el envío de datos entre la estación de control y el dron.
- La Arduino UNO Clon que actúa como estación de control para los operadores humanos, para enviar comandos autónomos al dron para ajustar su comportamiento durante el vuelo.

Para la conexión entre sí, hacen falta cuatro conexiones:

- **Alimentación.** Las líneas de tierra y de +5V del transceptor APC220 se conectan en sus respectivos pines de alimentación presentes en la Arduino UNO Clon.

- **Transmisión serie.** Se utilizan los pines 2 y 3 de la Arduino UNO Clon para recepción (RX) y transmisión (TX). Hay que conectar el pin 2 con el pin TX del transceptor, y el pin 3 con el pin RX del transceptor.

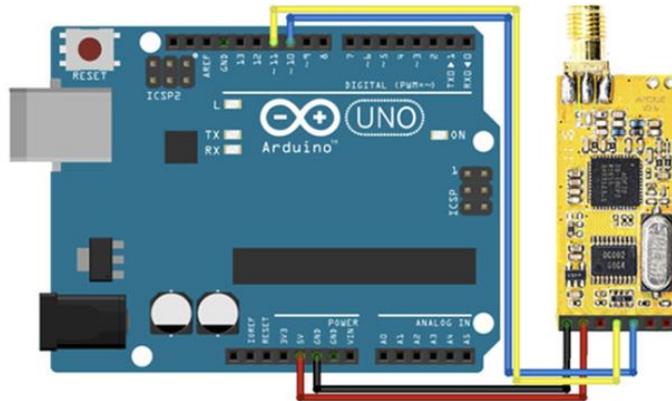


Figura 22- Conexión entre Arduino UNO y módulo transceptor APC220 [35]

Capítulo 5. SOFTWARE: CONTROLADOR DE VUELO AUTÓNOMO

El desarrollo de un dron autónomo representa un desafío teórico y práctico considerable. A pesar de que el hardware estaba correctamente montado y que cada componente funcionaba de forma individual, no se logró que el dron volara de manera autónoma debido a la falta de tiempo para refinar el software y realizar más pruebas. Es por esto por lo que este capítulo deja atrás el hardware desarrollado en el capítulo anterior, y se centra en un marco teórico del software necesario para que el dron sea capaz de volar autónomamente.

Para empezar, se estudió con detalle el controlador de vuelo open-source [1] instalado en el microcontrolador STM32F103C8T6 a bordo del dron para realizar pruebas de vuelo. Dicho software está programado en C++. El objetivo era modificar su funcionamiento para que, en lugar de recibir instrucciones manuales (a través del transmisor y receptor), se pudiera controlar los movimientos del dron usando el sistema de telemetría (con el transceptor conectado al Arduino UNO Clon).

Asimismo, se había de desarrollar un pseudocontrolador de vuelo (para alojarse en la Arduino UNO Clon) que definiera las instrucciones específicas que debían enviarse al dron para guiar su comportamiento, así como el momento adecuado para hacerlo. Dicho pseudocontrolador actúa, por tanto, como intermediario entre un operador humano y el dron, porque gestiona la comunicación y el control del dron a través del sistema de telemetría, pero debe contar con una GUI intuitiva para que la opere un humano.

5.1 CONTROLADOR DE VUELO OPEN-SOURCE

El controlador de vuelo es el núcleo del sistema del dron. Se utiliza un controlador basado en código abierto porque gestiona todas las tareas críticas necesarias para que el dron vuele, desde la recepción de comandos de vuelo, hasta el control preciso de los motores [1]. A

continuación, se proporciona una visión general del sistema y sus funciones específicas; su archivo principal y el resto de los drivers involucrados.

- ***Flight_Controller.ino***. Este es el archivo principal del controlador de vuelo que coordina la inicialización y el bucle *main* principal del sistema, donde se integran el resto de drivers y que se explican a continuación.
- ***calibration.ino***. La calibración es esencial para asegurar que las lecturas de los sensores (giroscopio y acelerómetro) sean precisas y confiables. El proceso de calibración implica tomar múltiples lecturas de los sensores, promediar dichas lecturas, y ajustar cualquier offset detectado.
- ***Barometer.ino***. Este fichero se enfoca en leer los datos de presión y de temperatura para calcular la altura del dron basada en esas lecturas.
- ***gyro.ino***. Gestiona la comunicación con el acelerómetro-giroscopio MPU-6050 y la lectura de datos de dicho sensor. Para ello, se utiliza el protocolo de comunicación I²C.
- ***vertical_acceleration_calculations.ino***. Este módulo calcula la aceleración vertical del dron utilizando una memoria rotativa para obtener una medida lo más precisa posible de la aceleración a lo largo del tiempo. Su función principal es medir y procesar la aceleración vertical del dron para asegurar la estabilidad en vuelo.
- ***LED_control.ino***. Este fichero maneja la iluminación de los LEDs del dron, para indicar distintos estados operativos y errores.
- ***calculate_pid.ino***. Implementa la lógica de ecuaciones de control proporcional-integral-derivativo (PID) para ajustar la respuesta del dron a las condiciones de vuelo detectadas por los sensores, para lograr la estabilización del dron durante su operación.
- ***timer_setup.ino***. Configura los temporizadores para la lectura de entradas PPM y la generación de salidas PWM.
- ***input_capture_mode_handlers.ino***. Código de la interrupción del temporizador que lee las entradas PPM, para calcular el tiempo entre pulsos PPM y poder determinar los comandos precisos de vuelo.

- *start_stop_takeoff.ino*. Gestiona las secuencias de inicio y parada del motor para el despegue y aterrizaje, así como la anulación del proceso de despegue por alguna razón de seguridad.

Nuevamente, cabe destacar que hay varios ficheros que implementaban otro tipo de funciones (como control por GPS), que no se deseaba incluir en la solución propuesta para el dron autónomo. Específicamente, dichos ficheros son los siguientes: *change_settings.ino*, *read_compass.ino*, *read_gps.ino* y *send_telemetry_dat.ino*.

5.1.1 CONTROL PID

Una de las tareas más importantes para el microcontrolador STM32F103C8T6 es controlar las velocidades de cada uno de los cuatro motores usando un bucle de control proporcional-integral-derivativo (PID). Esto es crucial para mantener la estabilidad del dron durante el vuelo.

La ecuación de control PID utilizada en controlador de vuelo es la siguiente [36]:

$$u(t) = K_p \cdot e(t) + K_i \cdot \int e(t) \cdot dt + K_d \cdot \frac{de(t)}{dt}$$

Donde:

- $u(t)$ es la salida de control.
- $e(t)$ es el error entre los valores deseados y los reales.
- K_p , K_i , K_d son las ganancias proporcional, integral y derivativa, respectivamente.

Estas variables determinan el tiempo de respuesta y la estabilidad del sistema.

- Una ganancia K_p más alta mejora el tiempo de respuesta, pero puede disminuir la estabilidad.
- Por otra parte, K_i y K_d más altas mejoran la estabilidad, pero pueden ralentizar la respuesta.

Hubo que ajustar estas ganancias en los primeros vuelos del dron para mejorar su estabilidad. Como se puede ver en la figura siguiente, para roll y pitch las ganancias son $K_p = 1$, $K_i = 0$ y $K_d = 23$. Para yaw, las ganancias son $K_p = 3.9$, $K_i = 0$ y $K_d = 0.3$.

```
float pid_p_gain_roll = 1;           // Roll P-controller (default = 1.3).
float pid_i_gain_roll = 0;           // Roll I-controller (default = 0.04).
float pid_d_gain_roll = 23;          // Roll D-controller (default = 18.0).
int pid_max_roll = 400;              // Max output of the PID-controller (+/-).

float pid_p_gain_pitch = pid_p_gain_roll; //Pitch P-controller.
float pid_i_gain_pitch = pid_i_gain_roll; //Pitch I-controller.
float pid_d_gain_pitch = pid_d_gain_roll; //Pitch D-controller.
int pid_max_pitch = pid_max_roll;

float pid_p_gain_yaw = 3.9;          //Yaw P-controller (default = 4.0).
float pid_i_gain_yaw = 0;            //Yaw I-controller (default = 0.02).
float pid_d_gain_yaw = 0.3;         //Yaw D-controller (default = 0.0).
int pid_max_yaw = 400;
```

El dron emplea tres bucles de control PID separados para gestionar de forma independiente el roll, pitch y yaw. Los valores deseados para estas orientaciones los establece el sistema de control externo. Por ejemplo, si un operador humano inclina el stick derecho hacia la izquierda, aumenta el ángulo de roll deseado. Por su parte, los valores actuales se obtienen a partir del giroscopio-acelerómetro (MPU-6050) y barómetro (MS5611). A partir de ahí, el controlador PID calcula los ajustes necesarios para minimizar el error $e(t)$.

El microcontrolador calcula las salidas PID, que se utilizan para ajustar la potencia entregada a cada motor. Dichos ajustes se realizan sumando o restando las salidas PID en función de la posición y dirección del motor para lograr la orientación deseada.

5.1.2 RECEPCIÓN DE INSTRUCCIONES NO AUTÓNOMAS A TRAVÉS DEL TRANSMISOR Y RECEPTOR DE CONTROL REMOTO

El sistema para la recepción de instrucciones no autónomas en el dron está basado en la comunicación entre el transmisor y receptor del control remoto, utilizando señales PPM.

El controlador de vuelo del dron recibe instrucciones del transmisor de control remoto a través del módulo receptor conectado al dron. Las principales instrucciones se transmiten a través de varios canales siendo los principales canales de interés los siguientes:

- Throttle (canal 3)
- Roll (canal 1)
- Pitch (canal 2)
- Yaw (canal 4)

A continuación, se explica cómo se decodifican dichas señales y cómo se integran en el control del movimiento del dron.

5.1.2.1 Decodificación de señales PPM

Las señales PPM se reciben y decodifican en el microcontrolador. Cada canal de la señal PPM corresponde a una instrucción específica de throttle, roll, pitch o yaw. La función *handler_channel_1* en el driver *input_capture_mode_handlers.ino* utiliza interrupciones para capturar los tiempos de subida y bajada de las señales PPM, permitiendo así medir con precisión los tiempos de los pulsos y determinar los valores exactos para cada canal.

5.1.2.2 Generación de señales PWM para los ESCs

Una vez decodificadas las señales PPM, se han de generar señales PWM para controlar los ESCs, que son los responsables de ajustar la velocidad de los motores del dron. La configuración de los temporizadores para generar las señales PWM se realiza en el fichero *timer_setup.ino*. La duración de los pulsos PWM determina la velocidad de los motores. Por ejemplo, para aumentar el valor de throttle, se alarga la duración del pulso PWM.

5.1.2.3 Integración del bucle de control PID

El bucle de control PID se integra en el bucle principal del controlador de vuelo para ajustar los valores de PWM según las desviaciones detectadas entre las orientaciones deseadas y las actuales. Los cálculos necesarios se realizan en el driver *calculate_pid.ino*.

Los valores PID calculados (pid_output_pitch , pid_output_roll y pid_output_yaw) se usan para ajustar las señales PWM que se envían a los ESCs (que controlan los motores), para asegurar que el dron responda a las instrucciones recibidas del transmisor remoto, siempre manteniendo la estabilidad.

5.2 PSEUDOCONTROLADOR AUTÓNOMO Y EJEMPLOS DE CASO DE FUNCIONAMIENTO

En los diagramas siguientes se muestra la interacción entre el operador humano, la Arduino UNO Clon y el dron. Establecen las bases para lo que debería ser el controlador de vuelo autónomo. Algunas de las funciones se desarrollan en las secciones siguientes, y otras se dejan indicadas de forma teórica (ya que su desarrollo se sale del alcance de este proyecto), y han de ser desarrolladas en fases posteriores.

5.2.1 OPERADOR HUMANO

El operador humano ha de ser capaz de realizar las funciones que se muestran a continuación.

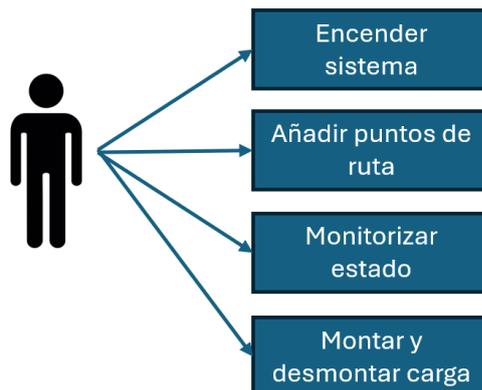


Figura 23. Centro de control de tierra

Función	Descripción
Encender el sistema	Se pone en funcionamiento todo el sistema del dron y se conecta a él automáticamente el sistema de telemetría.
Añadir puntos de ruta	Definir los puntos inicial y final para que el dron sepa dónde tiene que ir.
Monitorizar el estado del dron	Supervisar en tiempo real el estado del dron y atender a situaciones de peligro que puedan surgir durante el vuelo.
Montar y desmontar la carga	Manejar físicamente la carga útil del dron, ya sea montándola en el punto inicial o desmontándola en el punto de destino.

Tabla 4- Funciones del centro de control de tierra

5.2.2 ARDUINO UNO CLON

Por su parte, el sistema de telemetría que aloja el controlador de vuelo autónomo debe de cubrir una serie de funciones clave relacionadas con optimización de ruta y envío de instrucciones para el control de los cuatro motores.

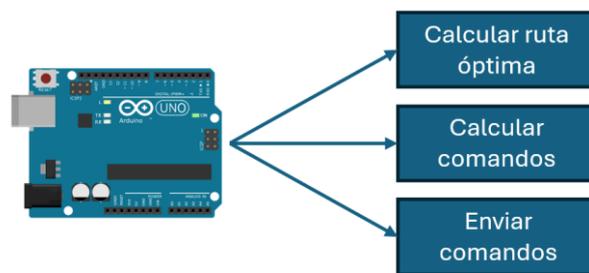


Figura 24- Sistema de telemetría

Función	Descripción
Calcular la ruta óptima	Dados los puntos inicial y final de la ruta, se ha de determinar la ruta óptima mediante algún algoritmo (Dijkstra, por ejemplo), para ahorrar tiempo y energía.

Calcular los comandos correspondientes a la ruta	Dicha ruta ha de traducirse en comandos específicos que el dron pueda interpretar.
Enviar los comandos al dron	Se transmiten dichos comandos de vuelo al controlador de vuelo del dron mediante el transceptor APC220.

Tabla 5- Funciones del sistema de telemetría

5.2.3 DRON

Por último, el dron es el tercer sistema clave que debe de realizar los vuelos que transporten los materiales semiconductores de una estación de prueba a otra autónomamente, de la manera más segura y eficiente posible.

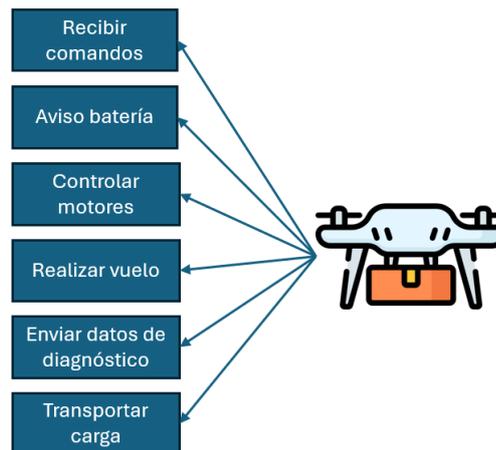


Figura 25- Sistema del dron

Función	Descripción
Recibir comandos	Recibe comandos por medio de la comunicación serial con el transceptor APC220 conectado a la Arduino UNO Clon.
Aviso de batería	Enviar alertas sobre el nivel de batería periódicamente.
Controlar motores directamente	Ajustar la velocidad y dirección de los motores según los comandos recibidos.

Realizar vuelo fijado por la estación de tierra	Ejecutar la misión de vuelo siguiendo los comandos recibidos.
Enviar datos de diagnóstico	Proveer información en tiempo real sobre métricas claves del vuelo, como altura y velocidad, por ejemplo.
Transportar carga de materiales semiconductores	Llevar la carga (materiales semiconductores) bien sujeta y segura desde el punto inicial de la ruta hasta el punto final especificado (que representan do estaciones de prueba diferentes).

Tabla 6- Funciones del sistema del dron

5.3 PROPUESTA PARA LOGRAR AUTONOMÍA MEDIANTE LA COMUNICACIÓN CON EL SISTEMA DE TELEMETRÍA

En las secciones anteriores se detalló cómo el controlador de vuelo open-source [1] está programado para trabajar con las instrucciones manuales recibidas a través del transmisor-receptor de control remoto. Además, se describió una multitud de funciones que idealmente debería implementar el sistema de transporte autónomo para ser lo más eficiente y seguro posible.

En esta sección, se propone una modificación del sistema para permitir la recepción de instrucciones autónomas a través del sistema de telemetría. Dicha propuesta constituye un paso importante hacia la plena autonomía del sistema. Mediante el uso del microcontrolador STM32F103C8T6 y la Arduino UNO Clon, se puede conseguir más flexibilidad en el control del dron para que ejecute movimientos autónomos.

La propuesta en cuestión incluye modificaciones simples en el software del controlador de vuelo open-source [1] para aceptar comandos directamente del transeptor, omitiendo así la necesidad de que un operador humano opere el transmisor de control remoto para controlar el dron. Los comandos pueden enviarse a través de la comunicación serial UART1 del microcontrolador STM32F103C8T6, utilizando el módulo de telemetría

APC220. La Arduino UNO Clon actúa como un controlador de vuelo autónomo, enviando comandos específicos al dron para que realice movimientos autónomos predeterminados.

5.3.1 MICROCONTROLADOR STM32F103C8T6

En esta sección se desarrolla cómo se modificó ligeramente el controlador de vuelo open-source [1] para permitir que la Blue Pill reciba instrucciones autónomas directamente del transceptor. Es decir, el objetivo de dichos cambios es que el controlador de vuelo ajuste los valores de los cuatro canales (throttle, roll, pitch y yaw) sin depender de la lectura de las señales PPM que aporta el receptor de control remoto manual.

5.3.1.1 Recepción de comandos a través de la UART

Para lograr la autonomía del dron mediante el sistema de telemetría, se implementan modificaciones en el controlador open-source [1] que permitan bypass el transmisor remoto y controlar directamente el throttle, roll, pitch y yaw a través del transceptor conectado a la UART1 del microcontrolador STM32F103C8T6.

En el código a continuación, se muestra cómo se inicializa la comunicación UART y cómo se leen los caracteres recibidos para determinar qué comando hay que ejecutar.

1. En primer lugar, se inicializa la comunicación serial en el puerto *Serial1* con una velocidad de 9600 baudios (bits por segundo). Es importante que esta velocidad de recepción coincida con la velocidad de transmisión del dispositivo conectado a *Serial1*, que en este caso es el transceptor del sistema de telemetría.
2. En la segunda línea se verifica si el transceptor ha enviado algún dato al microcontrolador; es decir, se comprueba si hay datos disponibles para leer del buffer de *Serial1* si *Serial1.available()* devuelve un valor mayor que 0.
3. En caso de verificarse la condición, en la siguiente línea se lee un byte de datos desde *Serial1* y se almacena en la variable *recievedChar*.
4. Después, dicho carácter se imprime en el monitor serial del IDE (con fines de verificación y depuración).

5. A continuación, se introduce una breve pausa de 100 milisegundos para asegurar que todos los datos hayan sido recibidos correctamente, antes de proceder a su procesamiento.
6. El bucle *while* denota que mientras haya datos disponibles en *Serial1*, se lee cada byte y se almacena en la variable *loop2_counter*. El propósito de esto es asegurar que no queden datos sin leer.
7. Finalmente, se establece la variable *disable_throttle* en 1. Esto se hace como medida de seguridad para poner el dron en un estado seguro durante el procesamiento de comandos.

```
Serial1.begin(9600);  
  
if (Serial1.available()) {  
  recievedChar = Serial1.read();  
  Serial.println(recievedChar);  
  delay(100);  
  while (Serial1.available() > 0) loop2_counter = Serial1.read();  
  disable_throttle = 1;  
}
```

5.3.1.2 Control directo de los motores

Una vez recibido un comando a través del sistema de telemetría, el sistema puede ajustar directamente los valores de *throttle*, *roll*, *pitch* y *yaw*, mediante asignación de valores específicos a sus canales correspondientes. En el código a continuación se muestra, a modo de ejemplo, cómo se puede ajustar el *throttle* para controlar individualmente cada uno de los cuatro motores.

Si se recibe el carácter *1*, se configura el *channel_3* (correspondiente al throttle) a 1200, y se ajustan los valores de los registros de comparación para cada canal del temporizador 4.

- ***channel_3 = 1200;*** establece el valor del canal 3 en 1200, lo que representa el ancho del pulso PWM para los ESCs.
- ***TIMER4_BASE->CCR1 = channel_3;*** establece el valor PWM del canal 1 del temporizador 4 en 1200, lo que implica que el motor 1 comienza a girar a una

velocidad análoga a que el stick del throttle del transmisor del controlador remoto estuviese más o menos por la mitad.

- **TIMER4_BASE->CCR2 = 1000;** establece el valor PWM del canal 2, temporizador 4 en 1000, lo que mantiene el motor 2 en reposo.
- Con los motores 3 y 4 pasa igual que con el motor 2.

```
if (recievedChar == '1') {
    channel_3 = 1200;
    TIMER4_BASE->CCR1 = channel_3;
    TIMER4_BASE->CCR2 = 1000;
    TIMER4_BASE->CCR3 = 1000;
    TIMER4_BASE->CCR4 = 1000;
}

if (recievedChar == '2') {
    channel_3 = 1200;
    TIMER4_BASE->CCR1 = 1000;
    TIMER4_BASE->CCR2 = channel_3;
    TIMER4_BASE->CCR3 = 1000;
    TIMER4_BASE->CCR4 = 1000;
}

if (recievedChar == '3') {
    TIMER4_BASE->CCR1 = 1000;
    TIMER4_BASE->CCR2 = 1000;
    TIMER4_BASE->CCR3 = channel_3;
    TIMER4_BASE->CCR4 = 1000;
}

if (recievedChar == '4') {
    TIMER4_BASE->CCR1 = 1000;
    TIMER4_BASE->CCR2 = 1000;
    TIMER4_BASE->CCR3 = 1000;
    TIMER4_BASE->CCR4 = channel_3;
}

if (recievedChar == '5') {
    TIMER4_BASE->CCR1 = channel_3;
    TIMER4_BASE->CCR2 = channel_3;
    TIMER4_BASE->CCR3 = channel_3;
    TIMER4_BASE->CCR4 = channel_3;
}
```

En este caso, se asigna un valor de 1200 al *channel_3* para aumentar el throttle, lo cual se traduce en un aumento en la duración del pulso PWM (incrementando así la velocidad del motor correspondiente). Dependiendo del carácter recibido (1, 2, 3, 4 o 5), se establece

el valor PWM correspondiente a uno o varios motores, permitiendo su control independiente o simultáneo. Esto fue esencial para realizar pruebas y diagnósticos, y apunta a cómo implementar el control autónomo del dron a través del sistema de telemetría.

5.3.1.3 Bucle PID y estabilización del dron

Los valores recibidos y asignados a los canales son utilizados en el bucle PID para ajustar los parámetros de control de vuelo. Como se mencionó antes, este bucle de control se encarga del cálculo de las correcciones necesarias en función de las lecturas de los sensores y de los valores recibidos por el sistema de telemetría de control autónomo.

El bucle PID se integra en el código principal del controlador de vuelo open-source [1], y es el responsable de ajustar continuamente los valores definitivos de *roll*, *pitch* y *yaw* para mantener siempre la estabilidad del dron.

Para desglosar el código siguiente, se explican los cálculos de roll (inclinación lateral). Ocurre lo mismo para pitch (inclinación adelante/atrás) y yaw (rotación horizontal).

1. En primer lugar, se calcula el error de roll (*pid_error_temp*), como la diferencia entre el valor medido de roll y el valor objetivo deseado (*setpoint*).
2. Después, se realiza el cálculo del componente integral del roll acumulando el error del paso anterior multiplicado por la ganancia integral (*pid_i_gain_roll*).
3. Sin embargo, en las siguientes líneas hay que limitar el valor acumulado del término integral para evitar la saturación.
4. Para el cálculo del componente derivativo del roll, se calcula la salida PID combinando los siguientes términos: el proporcional (*pid_p_gain_roll*), el integral (*pid_i_mem_roll*) y el derivativo (*d_gain*).
5. De forma similar al componente integral, se limita la salida PID total para evitar sobrecarga o comportamiento inestable.
6. Por último, se actualiza el último error de roll (*pid_last_roll_d_error*).

```
void calculate_pid(void) {  
    // Cálculos de Roll  
    pid_error_temp = gyro_roll_input - pid_roll_setpoint;
```

```
pid_i_mem_roll += pid_i_gain_roll * pid_error_temp;
if (pid_i_mem_roll > pid_max_roll) pid_i_mem_roll = pid_max_roll;
else if (pid_i_mem_roll < pid_max_roll * -1) pid_i_mem_roll = pid_max_roll *
-1;

pid_output_roll = pid_p_gain_roll * pid_error_temp + pid_i_mem_roll +
pid_d_gain_roll * (pid_error_temp - pid_last_roll_d_error);
if (pid_output_roll > pid_max_roll) pid_output_roll = pid_max_roll;
else if (pid_output_roll < pid_max_roll * -1) pid_output_roll = pid_max_roll
* -1;

pid_last_roll_d_error = pid_error_temp;

// Cálculos de Pitch
pid_error_temp = gyro_pitch_input - pid_pitch_setpoint;
pid_i_mem_pitch += pid_i_gain_pitch * pid_error_temp;
if (pid_i_mem_pitch > pid_max_pitch) pid_i_mem_pitch = pid_max_pitch;
else if (pid_i_mem_pitch < pid_max_pitch * -1) pid_i_mem_pitch =
pid_max_pitch * -1;

pid_output_pitch = pid_p_gain_pitch * pid_error_temp + pid_i_mem_pitch +
pid_d_gain_pitch * (pid_error_temp - pid_last_pitch_d_error);
if (pid_output_pitch > pid_max_pitch) pid_output_pitch = pid_max_pitch;
else if (pid_output_pitch < pid_max_pitch * -1) pid_output_pitch =
pid_max_pitch * -1;

pid_last_pitch_d_error = pid_error_temp;

// Cálculos de Yaw
pid_error_temp = gyro_yaw_input - pid_yaw_setpoint;
pid_i_mem_yaw += pid_i_gain_yaw * pid_error_temp;
if (pid_i_mem_yaw > pid_max_yaw) pid_i_mem_yaw = pid_max_yaw;
else if (pid_i_mem_yaw < pid_max_yaw * -1) pid_i_mem_yaw = pid_max_yaw * -1;

pid_output_yaw = pid_p_gain_yaw * pid_error_temp + pid_i_mem_yaw +
pid_d_gain_yaw * (pid_error_temp - pid_last_yaw_d_error);
if (pid_output_yaw > pid_max_yaw) pid_output_yaw = pid_max_yaw;
else if (pid_output_yaw < pid_max_yaw * -1) pid_output_yaw = pid_max_yaw * -
1;

pid_last_yaw_d_error = pid_error_temp;
}
```

5.3.2 ARDUINO UNO CLON

El siguiente código es otra pieza fundamental para permitir el vuelo autónomo del dron, al transmitir instrucciones autónomas a través de un módulo de comunicación serial a través de la pareja de transceptores.

Al implementar esta configuración, se establece un sistema básico para la comunicación entre el controlador autónomo y el controlador de vuelo open-source [1]. Esto es un pequeño pero significativo paso hacia el vuelo completamente autónomo y automatizado del dron; es decir, sin la necesidad de intervención humana manual directa.

1. En primer lugar, es necesario incluir la librería *SoftwareSerial*. La razón para hacer esto es porque los pines 0 (RX) y 1 (TX) de la Arduino UNO Clon están ocupados por la conexión hardware serial integrada con el portátil a través de la conexión USB. Dicha conexión es vital para la programación y depuración de la placa a través de la IDE de Arduino. Debido a esta limitación, se utiliza *SoftwareSerial* con el fin de crear un puerto serial ‘virtual’ que permita la comunicación con otros dispositivos sin interferir con la conexión principal de hardware serial.
2. A continuación, se declara una variable de tipo carácter para almacenar los datos recibidos del monitor serial.
3. En tercer lugar, se definen los pines 2 y 3 como pines de recepción (RX) y transmisión (TX), respectivamente.
4. Para la configuración de la comunicación serial entre la Arduino UNO Clon y la Blue Pill, se inicializa una instancia de *SoftwareSerial* en los pines definidos en el paso anterior.
5. La función *setup()*:
 - a. Inicializa la comunicación serial con el monitor serie a una velocidad de 115200 baudios.
 - b. Inicializa la comunicación serial con el módulo de comunicación serial (transceptor) en los pines RX y TX a una velocidad de 9600 baudios.
6. Finalmente, la función *loop()*:
 - a. Comprueba si hay datos disponibles para leer desde el monitor serie (a través de la IDE de Arduino).
 - b. De ser así, lee el carácter disponible y lo almacena en la variable *c*.
 - c. A continuación, envía dicho carácter leído al módulo de comunicación serial, y lo imprime en el monitor serie para verificar que todo va bien.

- d. Introduce una pausa de 1 segundo antes de repetir todo el bucle de nuevo.

```
#include <SoftwareSerial.h>
char c = "";

#define RX_PIN 2
#define TX_PIN 3

SoftwareSerial mySerial(RX_PIN, TX_PIN);

void setup() {
  Serial.begin(115200);
  mySerial.begin(9600);
}

void loop() {
  if (Serial.available()) {
    c = Serial.read();
    mySerial.print(c);
    Serial.print(c);
  }
  delay(1000);
}
```

5.4 CÓDIGO PARA VUELO AUTÓNOMO EN LA ARDUINO UNO CLON

En esta sección se describe cómo se puede implementar la parte de telemetría del pseudocontrolador de vuelo utilizando la placa Arduino UNO Clon. ‘Pseudo’ porque habría que realizar muchísimas pruebas para su formalización. Este controlador tiene instrucciones preestablecidas que permiten que el dron realice movimientos específicos sin la intervención continua de un operador humano.

5.4.1 IMPLEMENTACIÓN

El desarrollo de un pseudocontrolador de vuelo autónomo en la Arduino UNO Clon se encarga de enviar instrucciones de vuelo al microcontrolador principal del dron (STM32F103C8T6) a través de la pareja de transceptores. La Arduino UNO Clon genera señales que emulan las señales PPM que el controlador de vuelo open-source [1] normalmente recibe del receptor remoto. Dichas señales han de enviarse a una frecuencia constante de 50 Hz, para imitar de la mejor forma posible la comunicación continua entre el controlador de vuelo del dron y el receptor de control remoto.

5.4.2 EJEMPLO DE CÓDIGO DE IMPLEMENTACIÓN

Para realizar un movimiento específico, como elevar el dron medio metro, se seguiría una secuencia de comandos predefinidos. Por ejemplo:

1. Para inicializar el despegue, se establece el valor de throttle a 1100.
2. Para aumentar la elevación progresivamente, se incrementa el throttle a 1200 en saltos pequeños durante 2 segundos.
3. Por último, para mantener la altura, se pone el throttle a 1300 y se mantiene dicho valor durante 3 segundos.

Cabe destacar que estos valores específicos y el momento justo para enviarlos para que el dron ejecute una serie de movimientos autónomos habrían de determinarse con numerosas pruebas de vuelo, utilizando los valores enviados a través del controlador remoto como referencia. Esto implicaría realizar ajustes finos y calibraciones continuas para asegurar que el dron respondiera de forma precisa y segura a los comandos autónomos, y así replicar exitosa y confiablemente las acciones que un operador humano llevaría a cabo manualmente a través del transmisor y receptor de control remoto. El siguiente código ilustra cómo se podrían implementar funciones de este tipo en el pseudocontrolador de la Arduino UNO Clon.

- La primera función se utiliza para enviar una secuencia de comandos de control de throttle al dron. Específicamente, se trata de controlar la altura del dron mediante el ajuste de la potencia de los motores en tres etapas. Se coloca un retraso de 2 segundos entre cada ajuste, para mantener la estabilidad.
- La segunda función es útil para mover el dron. Se usa para enviar una secuencia de comandos de control de la inclinación lateral, de la inclinación adelante/atrás y del giro del dron (roll, pitch y yaw, respectivamente), con un retraso entre cada ajuste para mantener la estabilidad.

```
void subir(int throttle1, int throttle2, int throttle3, int tiempo) {  
  mySerial.print(throttle1);  
  delay(2000);  
  mySerial.print(throttle2);
```

```
delay(2000);  
mySerial.print(throttle3);  
delay(tiempo * 1000);  
}  
  
void avanzar(int pitch, int roll, int yaw, int tiempo) {  
  mySerial.print(pitch);  
  delay(tiempo * 1000);  
  mySerial.print(roll);  
  delay(tiempo * 1000);  
  mySerial.print(yaw);  
  delay(tiempo * 1000);  
}
```

Una vez definidas las funciones para el movimiento del dron, solo queda enviarlas según el movimiento específico que deba realizar el dron; esto se traduce en una secuencia de miles de estas funciones.

En el código siguiente (a modo de ejemplo básico), se interpretan los caracteres que se reciben por el puerto serial a través del monitor serial de la IDE de Arduino de un portátil. Su estructura modular permite que se escale fácilmente, así como dejar un gran margen para la personalización de los movimientos autónomos (según las necesidades específicas en cada momento).

- Cuando se recibe el carácter *s*, el dron ejecuta la secuencia de elevarse medio metro mediante la función *subir*.
- En cambio, cuando se recibe el carácter *a*, el dron avanza hacia adelante mediante la función *avanzar*.

```
if (recievedChar == 's') {  
  subir(1100, 1200, 1300, 3); // Instrucción para subir medio metro  
}  
  
if (recievedChar == 'a') {  
  // Instrucción para avanzar un metro hacia adelante  
  avanzar(1200, 1000, 1100, 2);  
}
```

5.4.3 INTERFAZ GRÁFICA DE USUARIO

La interfaz gráfica de usuario (GUI) ha de permitir al usuario definir movimientos específicos para el dron, o incluso definir rutas completas para misiones específicas. El

pseudocontrolador de vuelo autónomo ha de interpretar dichas órdenes y traducirlas en microinstrucciones, que posteriormente serán enviadas a través de la Arduino UNO Clon al controlador del vuelo del dron para ajustar la velocidad de los motores según sea necesario.

Actualmente, la GUI consiste en el monitor serial de la IDE de Arduino, donde es posible enviar caracteres para el control del dron. En el futuro, la idea es desarrollar una aplicación frontend mucho más amigable para el usuario, que incluso integre varias funciones; como realizar un aterrizaje de emergencia, o anular inmediatamente una misión para ir a al punto de carga de batería más cercano. Esta evolución permitiría una interacción más segura e intuitiva para el operador humano, facilitando el control autónomo en diversas situaciones operativas del día a día del dron en un entorno específico (como una fábrica de Texas Instruments).

5.5 BÚSQUEDA DE LA RUTA ÓPTIMA EN UNA FÁBRICA

En esta sección se desarrolla brevemente otro elemento clave del sistema autónomo: la determinación de una ruta óptima para la misión del dron de transportar materiales semiconductores desde una estación de prueba en un punto de la fábrica hasta otra estación de prueba ubicada en un lugar diferente.

Este componente es vital para maximizar la eficiencia del dron al navegar por un entorno complejo como una fábrica de testeo de materiales semiconductores de Texas Instruments. Al implementar un algoritmo de búsqueda de rutas, el dron puede realizar misiones de transporte de manera más rápida y segura, optimizando así el flujo de trabajo y reduciendo el tiempo y energía empleados en su operación del día a día.

5.5.1 PLANO DE UNA FÁBRICA DE TEXAS INSTRUMENTS

A continuación, se muestra un plano proporcionado por Texas Instruments, que representa una fábrica dividida en varias zonas para diferentes tipos de Equipos de Prueba Automática (ATE). También se puede visualizar un área de *staging* (donde se preparan y almacenan los componentes semiconductores antes de ser probados).

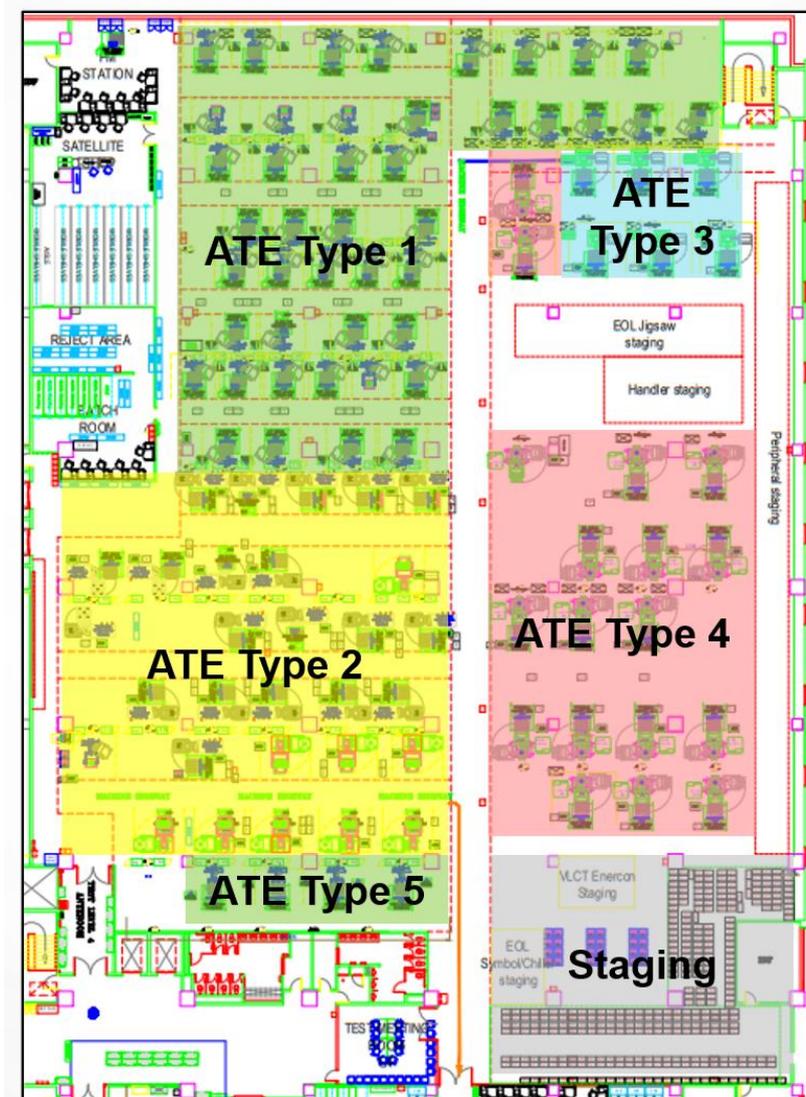


Figura 26- Plano de fábrica de Texas Instruments

El objetivo último es que el dron navegue eficientemente a través de esta disposición para transportar los materiales semiconductores entre las diferentes estaciones de prueba que actúan como puntos de inicio y/o destino para el dron. La planificación de la ruta óptima

debe considerar la disposición de las distintas áreas de prueba en la fábrica, para minimizar el tiempo de viaje y mantener la seguridad en todo momento.

5.5.2 ALGORITMO ESTADO-ENLACE: DIJKSTRA

Para determinar la ruta óptima entre dos estaciones de prueba, se propone utilizar el algoritmo de Dijkstra. Este algoritmo es adecuado para encontrar el camino más corto entre dos puntos en un grafo, lo cual es esencial para la navegación autónoma del dron.

5.5.2.1 Aplicación en el dron

- **Cálculo de rutas en tiempo real.** El controlador de vuelo autónomo del dron (alojado en la Arduino UNO Clon) será capaz de calcular la ruta óptima en tiempo real para cada misión de transporte que se le asigne desde la estación de tierra.
- **Adaptabilidad.** La modificación del código del algoritmo podría permitir al dron adaptarse a los cambios en el entorno, como nuevas obstrucciones o cambios en la disposición de la fábrica.

Capítulo 6. ANÁLISIS DE RESULTADOS

En este capítulo, se discuten los resultados relevantes relacionados con la evolución y mejora del sistema de vuelo del dron autónomo. Además, ofrece un análisis crítico del desempeño del sistema, así como todas las lecciones aprendidas durante el proceso.

Cabe mencionar que el desarrollo y la implementación de este proyecto resultaron ser significativamente más desafiantes de lo esperado inicialmente. Principalmente, esto fue debido a que suponía la integración de múltiples subsistemas y a la resolución de muchos problemas imprevistos que surgieron continuamente a lo largo del proceso. A continuación, se muestra una visión general del software de prueba utilizado (también programado en C++), y de los resultados obtenidos.

El programa de test open-source utilizado [1] para verificar los componentes y subsistemas del dron incluye diversas funcionalidades que pueden visualizarse en la imagen siguiente.

```
=====
                        YMFC-32 quadcopter setup tool
=====
a = Read the receiver input pulses
b = I2C scanner to detect any I2C sensors attached
c = Read the raw gyro values
d = Read the raw accelerometer values
e = Check the IMU angles
f = Test the LEDs
g = Read the battery voltage input
h = Check barometer
i = Check GPS
j = Check HMC5883L compass
=====
1 = Check motor 1 (front right, counter clockwise direction)
2 = Check motor 2 (rear right, clockwise direction)
3 = Check motor 3 (rear left, counter clockwise direction)
4 = Check motor 4 (front left, clockwise direction)
5 = Check all motors
=====
```

Figura 27- Programa de test open-source [1]

En cuanto al hardware, el testeo del sistema fue llevado a cabo en tres fases:

1. **Testeo de componentes individuales.** En primer lugar, cada componente de hardware individual comprado se probó para garantizar su funcionalidad básica.
2. **Testeo de subsistemas.** Más adelante, el testeo de subsistemas fue mayoritariamente exitoso, aunque se afrontaron diversos problemas menores que llevó tiempo arreglar.
3. **Testeo del dron en su conjunto.** Sin embargo, se encontraron grandes dificultades a la hora de integrar todo el sistema completo. Además, la incertidumbre acerca de qué exactamente causaba fallos en el sistema en repetidas ocasiones, retrasó considerablemente el desarrollo del proyecto con respecto a lo inicialmente planeado.

6.1 EVALUACIÓN DE COMPONENTES HARDWARE

Previo al montaje de los subsistemas de hardware presentados en el capítulo 4, cada componente incluido en el diseño ha de ser probado para asegurar que funciona correctamente. Estos componentes incluyen los siguientes: microcontrolador STM32F103C8T6, sensores, ESCs, motores, y el módulo de telemetría. Después de dichas pruebas, se determinó que cada componente individual de hardware funcionaba según se esperaba y que podían ser utilizados en el sistema completo del dron.

6.1.1 MÉTODO DE EVALUACIÓN

Para cada componente, se comenzó comprobando que se alimentaba correctamente y que se encendía. La forma para esto variaba según el componente; a veces era un LED que se encendía (como para el microcontrolador STM32F103C8T6), o midiendo la diferencia de tensión entre sus pines de salida (como para la batería). Una vez confirmada esta simple prueba, se llevaron a cabo pruebas más específicas para cada componente.

- Componentes como el microcontrolador o los transeptores requerían poder programar dicho módulo. Por tanto, con la creación de un programa muy simple (como un LED que parpadea a un intervalo regular), se pudo confirmar la funcionalidad de todos ellos.

- Para los módulos clasificados como sensores o instrumentos de medida, se testaron confirmando que cada uno de ellos era capaz de medir información relativa a ellos de forma precisa, o bien cambios en dicha información. Muchos de estos tests básicos requerían que el módulo estuviera conectado al microcontrolador.

6.1.2 RESULTADOS DE LAS PRUEBAS

Los resultados de los tests básicos indicaron que cada uno de los componentes hardware comprados funcionaban correctamente y que podían ser utilizados en el sistema del dron.

6.1.3 GESTIÓN DE PROBLEMAS

Estos primeros tests sirvieron para familiarizarse con la funcionalidad específica de cada componente. De todas formas, se contaba con un plan B para modificar aquellos componentes que no cumplieran los estándares originales establecidos. Esto incluía reemplazar componentes que fuesen inexactos, defectuosos, o que no alcanzaran su rendimiento en fases posteriores de integración. Gracias a la investigación previa realizada en componentes de hardware, era sencillo seleccionar otra opción adecuada y proceder a su compra (ya corriese por cuenta de la universidad o por cuenta personal).

Otro problema que se contempló era que pudiesen romperse los componentes más adelante. En ese caso, habría que reemplazarlo por la misma pieza, u otra que cumpliera el mismo trabajo y que se decidiese que era mejor (por ejemplo, por temas de tiempos de entrega).

6.2 *EVALUACIÓN DE SUBSISTEMAS DE HARDWARE*

Los subsistemas del dron que fueron evaluados incluyen: la distribución de energía, el microcontrolador STM32F103C8T6 (como cerebro del controlador de vuelo) y la comunicación inalámbrica con la pareja de transceptores. Surgieron muchos problemas que se fueron resolviendo poco a poco hasta eventualmente poder integrar todos estos subsistemas en el sistema completo.

6.2.1 MICROCONTROLADOR STM32F103C8T6

El microcontrolador principal STM32F103C8T6 está situado en la PCB del dron y es responsable de recibir los datos de los sensores, controlar los ESCs, y recibir los comandos del microcontrolador autónomo. El testeo de este módulo imprescindible se realizó verificando que se podía volcar en él los programas software open-source [1] para funcionar como el controlador de vuelo y de prueba de componentes.

Por ejemplo, se realizaron los tests más básicos presentes en el programa de prueba, los cuales resultaron exitosos. Por ejemplo, se verificó que el microcontrolador STM32F103C8T6 era capaz de encender cada uno de los LEDs soldados en la PCB durante tres segundos por separado.

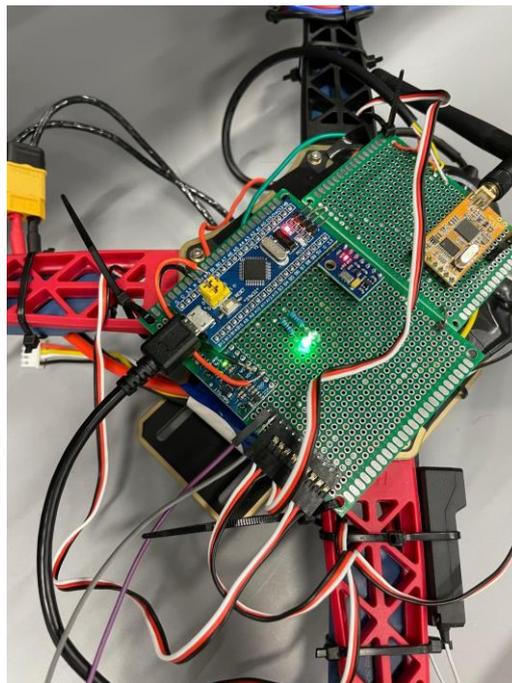


Figura 28- Prueba con LEDs

Por otro lado, originalmente se compraron tres microcontroladores STM32F103C8T6. El primero que se soldó dio problemas porque no era capaz de recibir la energía de la batería cuando ésta se conectaba al sistema del dron. Por esta razón se reemplazó con otro que se soldó de forma más precisa. El tercer microcontrolador se utilizó para realizar otro tipo de pruebas con la IDE de Arduino.

6.2.2 SISTEMA DE ALIMENTACIÓN

La estructura del chasis del dron incluía una red de distribución de energía integrada; lo cual incluía almohadillas de metal en las que se podía soldar cables para transferir energía.

Por su parte, los ESCs venían con conectores de energía XT60 incorporados, que debían ser quitados para soldarlos a la red de energía integrada. Hubo que cortar, pelar y soldar dos cables de 12 AWG en un extremo a la estructura del dron, y en el otro a un conector XT60. A su vez, un conector XT60 se conectaba a la batería LiPo de tres celdas.

La prueba de la red de distribución de energía se realizó (teniendo conectada la batería a través del conector XT60) conectando los dos cables de un ohmímetro a los puntos de energía positiva y negativa del dron. Esto se repitió para las 5 secciones y se determinó que la energía se distribuía uniformemente en todas ellas. Además, coincidían con el voltaje que indicaba la batería.

Sin embargo, mientras se transmite ese voltaje, puede pasar que las conexiones soldadas no permitan que la corriente fluya de manera eficiente. A pesar de ello, las pruebas en el sistema de propulsión en los apartados siguientes indicaron que esto no era un problema significativo, aunque algo que se debía vigilar y resolver si resultaba ser un problema grave en pruebas posteriores.

Sí que surgió un problema que hubo que resolver rápidamente: la soldadura se rompió de uno de los cables 12 AWG que alimentaban el dron. Se solucionó proporcionando una mejor integridad estructural a las conexiones de soldadura para que no se desprendiesen en el futuro.

Una vez se había verificado que se podía volcar el código open-source en el microcontrolador STM32F103C8T6 correctamente, se verificó el voltaje de la batería mediante el programa de prueba, lo que fue satisfactorio. Esto se determinó gracias a la salida del monitor serial, que confirmó que el voltaje entregado por la batería a las cinco secciones de la estructura del dron era de 11.4V.

```
Reading the battery voltage.  
You can exit by sending a q (quit).  
Voltage = 11.4V  
Voltage = 11.4V  
Voltage = 11.4V  
Voltage = 11.4V  
Voltage = 11.4V
```

Figura 29- Prueba de alimentación [1]

6.2.3 CONTROLADOR REMOTO

El sistema de controlador remoto fue una herramienta fundamental para el proyecto durante la realización de pruebas y ajustes del dron, ya que permitía enviar comandos para controlar los cuatro parámetros del vuelo mediante los dos sticks del transmisor. Específicamente, el transmisor utilizado envía señales PPM que son recibidas y decodificadas por el receptor del dron para ajustar los valores de los motores a través de los ESCs.

El programa de prueba permitía verificar que todos los pulsos de entrada que representaban los canales del controlador remoto se recibían correctamente. Esto se puede observar en la figura siguiente, y cómo los valores de throttle, roll, pitch y yaw cambiaban en tiempo real si se modificaba la posición de los sticks del transmisor de control remoto.

```
Reading receiver input pulses.  
You can exit by sending a q (quit).  
Start:0 Roll:--+1501 Pitch:--+1500 Throttle:vvv1008 Yaw:--+1500  
Start:0 Roll:--+1501 Pitch:--+1500 Throttle:vvv1008 Yaw:--+1500  
Start:0 Roll:--+1500 Pitch:--+1500 Throttle:vvv1008 Yaw:--+1500  
Start:0 Roll:--+1500 Pitch:--+1500 Throttle:vvv1008 Yaw:--+1500  
Start:0 Roll:--+1500 Pitch:--+1500 Throttle:vvv1008 Yaw:--+1500
```

Figura 30- Prueba del control remoto [1]

6.2.4 ESCs

Los ESCs son esenciales para el funcionamiento de los cuatro motores brushless del dron. Incluyen varios grupos de cables que los conectan a componentes clave y que les aportan la alimentación que necesitan.

- Dos cables de alimentación, que se conectan a la estructura del dron con distribución de energía integrada.
- Tres conectores tipo bala, que se conectan al motor y determinan su sentido de giro.
- Tres cables (rojo, negro, blanco), que se conectan a un receptor mediante una línea de alimentación de 5V, una línea de tierra y una línea de señal. En este caso, el receptor de cada una de las cuatro líneas de señal (para los cuatro ESCs) debía ser un pin del microcontrolador STM32F103C8T6 que enviase la señal única de PWM a cada uno de los cuatro pines. Cabe resaltar que los ESC además requieren una señal válida y calibración para funcionar correctamente.

Para determinar que el microcontrolador era capaz de controlar los ESC correctamente, se usa el programa de prueba que utilizaba el pin A0 para leer la entrada PPM del receptor del controlador remoto. Esta señal se descompone en los cuatro canales: throttle, roll, pitch y yaw. Posteriormente, el microcontrolador STM32F103C8T6 realiza los cálculos PID pertinentes para determinar los valores a enviar por medio de PWM a cada uno de los cuatro motores (pasando por los ESC).

Para emparejar el transmisor y receptor con los ESC se siguieron las instrucciones del fabricante. Este fue un proceso largo de prueba y error que implicó tener que descifrar los pitidos que hacían los ESC.

1. En primer lugar, una vez los ESCs están bien alimentados y conectados al resto de componentes, suena una secuencia de pitidos exacta para indicar que están correctamente emparejados con el receptor y transmisor.
2. Programar los ESCs implica calibrarlos para que conozcan los valores máximos y mínimos de PWM que envía el controlador de vuelo.
3. Una vez hecho esto, un humano puede encender los motores usando los sticks el transmisor de control remoto.

Se logró el objetivo y todos los motores se activaron al mismo tiempo y respondían correctamente a los movimientos del stick del transmisor del controlador remoto. Por lo tanto, se determinó que el sistema de propulsión funcionaba, por medio de los ESCs, y que las señales PWM que recibían eran válidas.

En la imagen siguiente se puede observar que el programa de prueba permite controlar cada uno de los motore individualmente, y medir la vibración presente en ellos. Hubo que balancear los rotores para lograr reducirlas al mínimo (sección 4.2.4.1). Además, el software era capaz de detectar si el throttle no está en la posición más baja al inicio de la prueba (esto es una medida de seguridad).

```
Check motor 1 (front right, counter clockwise direction).  
You can exit by sending a q (quit).  
Throttle is not in the lowest position.  
Throttle value is: 1094  
Waiting for 10 seconds:.....9557038  
11752131  
2  
3  
3
```

Figura 31- Prueba de vibración de motores [1]

6.2.5 SENSORES

Todos los sensores implicados en la solución final utilizan el protocolo I²C. Se probó que dicho protocolo funcionaba correctamente con el programa de prueba. En un primer momento justo después de soldar la PCB y fijarla en la parte de arriba de la estructura del dron, no se detectaron problemas de comunicación.

```
Starting the I2C scanner.  
The YMFC-32 needs the following devices:  
0x1E = HMC5883L compass  
0x68 = MPU-6050 gyro/accelerometer  
0x77 = MS5611 barometer  
They should appear in the list below.  
  
Scanning address 1 till 127...  
  
I2C device found at address 0x1E  
I2C device found at address 0x68  
I2C device found at address 0x77  
done
```

Figura 32- Prueba de comunicación con sensores [1]

Más adelante, después de realizar varias pruebas de vuelo, se observó que la comunicación I²C entre el microcontrolador STM32F103C8T6 y el barómetro MS5611 había sido dañada. En vez de levantar la placa protoboard, se decidió soldar un par de cables por encima punto a punto para resolver este problema. Para probar que cada sensor medía correctamente, también se utilizó el programa de prueba.

6.2.6 GIROSCOPIO- ACELERÓMETRO MPU-6050

Se conectaron los pines del reloj y datos del MPU-6050 con los pines correspondientes del microcontrolador STM32F103C8T6 y se utilizó el programa de prueba para enviar los datos del sensor al monitor en serie de la IDE de Arduino. Para probar que

las lecturas que tomaba el sensor eran correctas, se movió el sensor más lenta y rápidamente, y se comprobó que se detectaban dichos cambios.

```
For supportReading the raw accelerometer data.  
You can exit by sending a q (quit).  
ACC_x = -255 ACC_y = 5 ACC_z = 3662  
ACC_x = -244 ACC_y = 2 ACC_z = 3679  
ACC_x = -224 ACC_y = -1 ACC_z = 3635  
ACC_x = -247 ACC_y = 12 ACC_z = 3687  
ACC_x = -285 ACC_y = 14 ACC_z = 3689  
ACC_x = -222 ACC_y = 13 ACC_z = 3708  
ACC_x = -241 ACC_y = -23 ACC_z = 3718
```

Figura 33- Prueba del acelerómetro [1]

6.2.7 BARÓMETRO MS5611

Similarmente para el barómetro, se conectaron los pines respectivos a él con el microcontrolador STM32F103C8T6 para probar que las lecturas que tomaba el sensor eran correctas. Se movió el giroscopio en todas las direcciones de rotación.

```
Calibrating the gyro.....  
X calibration value:-135  
Y calibration value:146  
Z calibration value:25  
Gyro_x = 1 Gyro_y = 4 Gyro_z = 5  
Gyro_x = -4 Gyro_y = -3 Gyro_z = 0  
Gyro_x = -5 Gyro_y = 0 Gyro_z = 3  
Gyro_x = 0 Gyro_y = -4 Gyro_z = -1  
Gyro_x = 2 Gyro_y = 4 Gyro_z = -1  
Gyro_x = 1 Gyro_y = -1 Gyro_z = 2  
Gyro_x = -2 Gyro_y = 7 Gyro_z = -1  
Gyro_x = 4 Gyro_y = 2 Gyro_z = -1  
Gyro_x = 1 Gyro_y = -1 Gyro_z = 1
```

Figura 34- Prueba del barómetro [1]

6.2.8 COMUNICACIÓN CON EL CONTROLADOR DE VUELO AUTÓNOMO ARDUINO UNO CLON

El sistema de telemetría está compuesto por la pareja de transceptores APC220. El transceptor conectado a la Arduino UNO Clon actúa como transmisor, mientras que el

transceptor conectado al microcontrolador STM32F103C8T6 actúa como receptor. De esta forma, la Arduino UNO Clon puede actuar como intermediaria que envía instrucciones autónomas al microcontrolador STM32F103C8T6.

De todas formas, para verificar este sistema, se probó la comunicación bilateral.

1. Primero se probó la comunicación serial Arduino UNO Clon → Microcontrolador STM32F103C8T6. Se hicieron dos programas simples:
 - a. En el primero, se enviaban los caracteres introducidos por el usuario a través del monitor serial de la IDE de Arduino
 - b. En el segundo, se recibían dichos caracteres y se vaciaba el buffer. Si se enviaba un 1, 2 o 3, el LED del microcontrolador se encendía, apagaba o comenzaba a parpadear a un intervalo regular, respectivamente.
2. En segundo lugar, se revertieron los papeles; el transceptor del microcontrolador STM32F103C8T6 enviaba caracteres y la Arduino UNO Clon los recibía y se modificaba el comportamiento del LED.

Además, se probó el alcance de la comunicación serial alejando los dispositivos transceptores e interponiendo obstáculos (por ejemplo, paredes) de por medio, y se observó que el rendimiento y la velocidad de la transmisión apenas se veían afectadas.

6.3 EVALUACIÓN DEL SISTEMA COMPLETO

6.3.1 VUELOS DE PRUEBA Y PROBLEMAS ENCONTRADOS

Se realizaron muchos vuelos de prueba con el dron una vez el sistema estuvo completamente integrado. Muchos de ellos no tuvieron éxito debido a razones variadas, y se fueron corrigiendo problemas a medida que se realizaban más y más pruebas. A continuación, se resumen los principales problemas encontrados y cómo se solucionaron.

1. Las primeras pruebas resultaron en el dron volcándose hacia abajo. Se determinó que la causa más probable fuera debido a que las vibraciones de la PCB corrompían las medidas del giroscopio necesarias para reequilibrar el dron.

2. Tras fijar más la PCB, se descubrió que el dron se inclinaba considerablemente hacia atrás mientras volaba. Se pensó que se trataba principalmente debido a un problema de distribución de peso, ya que el módulo GPS (que no se utilizaba) estaba ubicado en la parte trasera del dron.
3. Una vez se empujó la batería hacia adelante para contrarrestar el peso del módulo de GPS, el dron se estabilizó por delante y por detrás. Sin embargo, se observó inestabilidad en el dron a lo largo del eje de giro. Tras fijar los valores de calibración del acelerómetro y no ver resultados, se decidió modificar las ganancias iniciales para los cálculos de PID en el software. A medida que se aumentaban los valores de ganancia (tras mucha prueba y error), se detectó que el dron se estabilizaba más y más. Durante este proceso se rompió una de las hélices del dron, y tuvo que ser reemplazada por la única que tenía un tiempo de entrega inferior a una semana, a pesar de ser de un material distinto a las otras.
4. Poco después de determinar los valores óptimos para las ganancias en el bucle de control PID, el dron se estrelló debido a un error humano y se rompió uno de los ESCs. Esto resultó en el motor dañándose también, ya que el ESC roto no lo protegió frente a las altas corrientes suministradas por la batería.

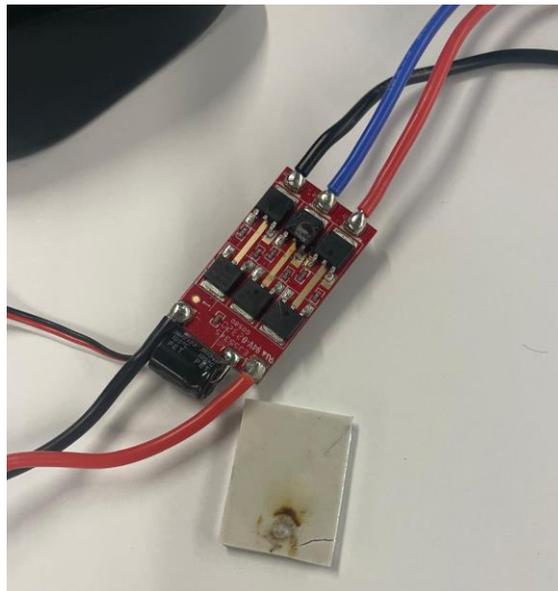


Figura 35- Se quemó el transistor central de un ESC durante la realización de pruebas de vuelo

6.3.2 PROBLEMAS PERSISTENTES DE ESTABILIDAD

Tras recibir los reemplazos para el ESC y el motor, se continuaron experimentando algunos problemas de estabilidad, especialmente al ganar altura. Las posibles causas de dichos problemas se desarrollan a continuación.

- **Material de hélices diferente.** Al romperse una de las hélices durante un vuelo de prueba para determinar los valores de ganancia del bucle de control PID, la que se compró para reemplazarla era de un material de mejor calidad que el de las demás. A pesar de ser de las mismas dimensiones, esto podría haber afectado a la aerodinámica del dron.
- **Descompensación de peso.** Podrían persistir problemas de distribución de peso. Por ejemplo, el receptor del controlador remoto estaba fijado en el lado de uno de los cuatro brazos del dron, sin ningún contrapeso en su diagonal.
- **Condiciones ambientales.** Las últimas pruebas de vuelo se realizaron en el mes de mayo, y el excesivo viento y las condiciones climáticas en el exterior podrían haber afectado a la estabilidad del dron.

6.4 ANÁLISIS DEL SOFTWARE

6.4.1 EVALUACIÓN DEL CONTROLADOR DE VUELO OPEN SOURCE

El software open-source [1] implementado en el microcontrolador STM32F103C8T6 funcionaba correctamente y era capaz de gestionar todas las tareas críticas necesarias para que el dron volase. Sin embargo, contenía algunas funciones obsoletas, por lo que fue de vital importancia utilizar una versión de la IDE de Arduino compatible.

Se logró la integración y modificación del software para permitir la recepción de comandos a través del sistema de telemetría. Entre los resultados clave, destacan los siguientes.

- **Inicialización y configuración.** Se logró compilar, volcar y ejecutar tanto el programa de prueba como el programa de controlador de vuelo sin errores críticos.
- **Control PID.** Se ajustaron los valores de las ganancias del bucle de control PID para mejorar la estabilidad del dron en vuelo, aunque también se identificaron áreas para mejoras adicionales en este aspecto, como un ajuste mucho más fino.

6.4.2 INTEGRACIÓN DEL SISTEMA DE TELEMETRÍA

Se logró la comunicación entre el microcontrolador STM32F103C8T6 y el sistema de telemetría mediante la pareja de transceptores APC220. Esto permitió la comunicación autónoma entre el dron y la estación de control (basada en la Arduino UNO Clon). Además, se probó y verificó con éxito varias funciones de comunicación. Los resultados más destacados se detallan a continuación.

- **Recepción de comandos.** El dron pudo recibir y ejecutar comandos enviados desde la Arduino UNO Clon para ajustar directamente la velocidad de los motores.
- **Pruebas de comunicación.** La comunicación entre la pareja de transceptores APC220 era estable y efectiva a largas distancias e incluso con obstáculos de por medio.

6.4.3 DESARROLLO DEL PSEUDOCONTROLADOR DE VUELO AUTÓNOMO

En tercer lugar, el pseudocontrolador de vuelo autónomo alojado en la Arduino UNO Clon se encargó de enviar secuencias de comandos predefinidos al dron para que ejecutase movimientos autónomos específicos. Aunque este sistema tuvo buenos resultados en las pruebas iniciales realizadas y sirve como prueba de concepto, hace falta mucho más trabajo en esta funcionalidad para lograr una autonomía completa, segura y robusta.

- **Secuencias de movimiento.** Se probaron a enviar secuencias de comandos para controlar la altura (throttle) del dron, con resultados prometedores en las pruebas básicas.

- **Flexibilidad en el control.** La modularidad del código permitió una fácil y rápida modificación y ampliación de las funciones de control, para llevar a cabo el mayor número de pruebas posible.

Capítulo 7. CONCLUSIONES Y TRABAJOS FUTUROS

7.1 CONCLUSIONES GENERALES

El proyecto de desarrollo de un dron autónomo para Texas Instruments afrontó múltiples dificultades, principalmente debido al presupuesto ajustado (por la imposibilidad de adquirir un dron programable para uso profesional

Por ende, al redefinir el proyecto y centrarlo en el hardware, surgieron diversos problemas e imprevistos, como la rotura de un ESC, una hélice y un motor. Además, la curva de aprendizaje en el montaje del hardware, así como de la programación y ajuste del sistema de vuelo autónomo fue más pronunciada de lo inicialmente anticipado.

Una conclusión clave aprendida durante el proyecto es la importancia de prepararse adecuadamente frente a las contingencias. Por ejemplo, si se hubiese anticipado la posibilidad de choques, fallos y daños en componentes críticos, se podría haber comprado piezas de repuesto adicionales desde el principio. Precisamente, dicha falta de piezas de repuesto disponibles provocó retrasos considerables en el cronograma, ya que se perdió tiempo valioso esperando la entrega de reemplazos, que como mínimo eran dos días. Esta experiencia subraya la necesidad de planificar con anticipación y mantener un inventario adecuado de componentes para evitar interrupciones y mantener el flujo continuo del trabajo.

A pesar de estos desafíos, se lograron cubrir los objetivos principales establecidos por Texas Instruments. Las aportaciones realizadas en el trabajo fueron valoradas positivamente por la empresa, lo que sugiere que el proyecto tiene potencial para ser desarrollado en el futuro (a pesar de su complejidad inherente), después de una extensa investigación acerca de la viabilidad de la implementación de éste.

7.2 LIMITACIONES AFRONTADAS

Las principales limitaciones encontradas durante el transcurso del proyecto fueron las siguientes:

- I. **Presupuesto.** La falta de recursos económicos impidió la compra de un dron programable para uso profesional, por lo que hubo que construirlo de cero. Esto limitó considerablemente el tiempo que se podía dedicar a cada tarea, así como la capacidad de realizar pruebas exhaustivas y de alta calidad.
- II. **Tiempo.** El tiempo limitado disponible para el proyecto también afectó a la capacidad para realizar pruebas exhaustivas y ajustes necesarios. La planificación inicial falló en prever todas las necesidades y contingencias del proyecto. Además, los retrasos debido a la necesidad de encargar online piezas de reemplazo para aquellas que se rompían empeoraron esta situación.
- III. **Experiencia.** No menos importante, la experiencia limitada en algunos aspectos técnicos específicos del proyecto hizo más complicada la definición del alcance inicial y la ejecución efectiva de todas las tareas planeadas.
- IV. **Problemas de hardware.** Finalmente, surgieron diversos problemas con el hardware (rotura de ESC, hélices y motores), lo cual causó retrasos en el proyecto y aumentó considerablemente su complejidad.

A pesar de estas limitaciones, se logró desarrollar un prototipo funcional que cumplió con los objetivos básicos del proyecto, demostrando su potencial y abriendo las puertas a futuras investigaciones y desarrollos en esta área.

7.3 RECOMENDACIONES PARA TRABAJOS FUTUROS

Teniendo en cuenta las limitaciones a lo largo del proyecto, y habiendo aprendido de ellas, las recomendaciones que se presentan a continuación proporcionan una ruta clara para el desarrollo y la optimización del sistema de drones autónomos de Texas Instruments.

7.3.1 INVESTIGACIÓN EN TECNOLOGÍAS DE BATERÍAS

En primer lugar, se recomienda una mayor investigación en las tecnologías de baterías que podrían prolongar la autonomía del dron sin añadir mucho peso al sistema. Es decir, mejorar las capacidades operativas del dron mediante la implementación de baterías de mayor densidad, o que gestionen la energía de forma más eficiente. Además, la exploración de sistemas de gestión térmica avanzados podría maximizar la eficiencia y seguridad de las baterías.

7.3.2 SISTEMA DE NAVEGACIÓN SOFISTICADO

Para mejorar la precisión y la fiabilidad del dron en un entorno interior complejo, como una fábrica para la manufacturación de materiales semiconductores, se debe implementar un sistema de navegación más sofisticado que no solo sea autónomo sino inteligente. Esto incluye la idea inicial para el proyecto de un sistema de posicionamiento en interiores (IPS) robusto.

La implementación de sensores adicionales, como sensores de detección y localización por luz (LiDAR) y cámaras de visión artificial, podrían aumentar significativamente la precisión y fiabilidad del posicionamiento dentro de estructuras complejas. El objetivo final es que el dron sea consciente de dónde está en cada momento y tome decisiones en tiempo real en base a ello. Esto le permitiría, por ejemplo, evitar obstáculos o redefinir su ruta si encuentra algún camino congestionado

7.3.3 MECANISMOS DE CARGA Y DESCARGA DE CARGA ÚTILES

Además, se sugiere explorar mecanismos diferentes de fijación y desmontaje de cargas útiles, capaces de manipular portadores de semiconductores de diferentes tamaños sin necesidad de reconfiguración manual.

Además, la implementación de materiales o técnicas de absorción de impactos en el sistema de carga útil también contribuiría a mejorar la seguridad e integridad de los materiales semiconductores durante el transporte.

7.3.4 SISTEMAS ESCALABLES

También en esta línea, se recomienda explorar sistemas escalables que se adapten a cargas útiles más grandes o pequeñas, sin rediseños significativos, para que brinden una mayor flexibilidad operativa para las necesidades específicas de Texas Instruments. Esto podría incluir un diseño modular para permitir adaptaciones a nuevos requerimientos que fueran surgiendo sobre la marcha, o para adaptarse a nuevas tecnologías.

7.3.5 ANÁLISIS DE DATOS EN TIEMPO REAL

Es fundamental el desarrollo de capacidades de análisis de datos de diagnóstico en tiempo real que proporcionen información sobre el rendimiento y la eficiencia del sistema mientras opera. La integración de dichos mecanismos de retroalimentación permitiría a los operadores humanos ajustar los parámetros del sistema de vuelo en función de los datos de diagnóstico en tiempo real, optimizando su funcionalidad. Este enfoque puede abarcar el uso de inteligencia artificial para el análisis y el ajuste proactivo de los parámetros de vuelo.

7.3.6 PROGRAMAS DE ENTRENAMIENTO Y MEJORA DE LA INTERFAZ DE USUARIO

También se recomiendan mejoras en la interfaz gráfica de usuario (GUI) para ayudar en el monitoreo del sistema. Una interfaz mejorada podría incluir gráficos interactivos, alertas en tiempo real y opciones de control más accesibles y mejoradas.

7.3.7 DESARROLLO DE UN ALGORITMO DE NAVEGACIÓN

Otro de los elementos vitales para la mejora de la autonomía y eficiencia del dron que no dio tiempo a implementar es el desarrollo de un algoritmo de Dijkstra para determinar la ruta óptima entre dos estaciones de prueba dentro de la fábrica de Texas Instruments. Dicho algoritmo permitiría a la Arduino UNO Clon calcular la ruta más corta entre dos puntos, mejorando así la productividad del dron y fomentando un entorno de trabajo seguro.

7.4 MÁS RECOMENDACIONES EN MATERIA DE SEGURIDAD

Tanto la seguridad como la práctica de buena ética son esenciales en este proyecto destinado a operar en un entorno con personas y materiales delicados. Por ello, además de todas las recomendaciones previas, se incide en medidas de seguridad adicionales.

7.4.1 EVALUACIÓN Y GESTIÓN DE RIESGOS

Antes de cualquier operación del dron, es importante que se realicen evaluaciones de riesgos exhaustivas, para identificar posibles puntos de falla y establecer medidas preventivas. La implementación de un sistema de gestión de riesgos que evalúe continuamente el entorno operativo del dron, podría contribuir a anticipar y mitigar posibles peligros.

7.4.2 PROTOCOLOS DE SEGURIDAD OPERACIONAL

También es esencial establecer protocolos de seguridad claros y detallados para todas las fases del vuelo. Dichos protocolos deben incluir procedimientos para las situaciones de despegue, vuelo, aterrizaje y emergencias. Es crucial que todos los operadores humanos encargados del manejo del dron estén capacitados en dichos procedimientos, y que se realicen simulacros regulares para asegurar su efectividad en el largo plazo.

7.4.3 REDUNDANCIA Y RESPALDO DE SISTEMAS

También es imperativo la implementación de sistemas redundantes para garantizar la seguridad de todos los trabajadores. Esto incluye sensores duales, sistemas de navegación duplicados y fuentes de alimentación de respaldo. En caso de fallo de algún componente crítico, los sistemas redundantes deben tomar el control, permitiendo al dron continuar su operación o aterrizar de manera segura.

7.4.4 MONITOREO Y DIAGNOSTICO EN TIEMPO REAL

Se recomienda encarecidamente incluir sistemas de diagnóstico más completos y precisos para monitorizar continuamente todos los sistemas del dron durante su vuelo, con

el fin de proporcionar alertas tempranas sobre posibles fallos. Sistemas de diagnóstico avanzados que analicen datos en tiempo real pueden identificar patrones que indiquen problemas inminentes.

7.4.5 MANTENIMIENTO PREVENTIVO

También es esencial desarrollar un programa de mantenimiento preventivo que sea riguroso, para asegurar la longevidad y fiabilidad del dron a largo plazo. Dicho programa debe incluir elementos como: inspecciones regulares, pruebas de mantenimiento y el reemplazo de componentes críticos periódicamente.

7.4.6 SISTEMAS DE DETECCIÓN Y EVITACIÓN DE OBSTÁCULOS

Para aumentar aún más la seguridad del sistema, hay que integrar sistemas de detección y evitación de obstáculos. Estos sistemas deben utilizar tecnología LiDAR, cámaras o sensores ultrasónicos para detectar y evitar automáticamente obstáculos durante el vuelo, para evitar colisiones.

7.4.7 POLÍTICAS DE SEGURIDAD Y CUMPLIMIENTO NORMATIVO

Mantenerse actualizado con las regulaciones y normas de seguridad vigentes, así como las mejores prácticas de la industria es crucial. Es importante que las operaciones del dron,(un UAV que opera dentro de una instalación industrial), cumplan con todas las regulaciones locales e internacionales, y que se implementen políticas de seguridad que aborden aspectos como la privacidad, la seguridad física y el uso de datos sensibles.

7.4.8 CAPACITACIÓN Y CONCIENCIACIÓN DEL PERSONAL

Por último, la capacitación continua del personal es esencial para mantener altos estándares de seguridad. Se recomienda que los operadores humanos reciban formación regular sobre nuevos procedimientos, actualizaciones de sistemas y mejoras en materia de seguridad. Además, fomentar una cultura de seguridad dentro de la organización puede ayudar a que todos los empleados estén comprometidos con la seguridad.

Capítulo 8. BIBLIOGRAFÍA

- [2] Hitachi High-Tech Corporation. (n.d.). *Semiconductors in everyday life*. Hitachi High-Tech. Retrieved from <https://www.hitachi-hightech.com/global/en/knowledge/semiconductor/room/about/life.html>
- [3] Martínez, J., & Rodríguez, A. (2023). *La aplicabilidad de los drones a la logística: Presente y futuro de una tecnología en crecimiento*. ResearchGate. Retrieved from https://www.researchgate.net/publication/370928130_La_aplicabilidad_de_los_drones_a_la_logistica_Presente_y_futuro_de_un_tecnologia_en_crecimiento
- [4] H. Zhang *et al.*, “Global Path Planning of Unmanned Surface Vehicle Based on Improved A-Star Algorithm,” *Sensors*, vol. 23, no. 14, pp. 6647-6665, Jul. 2023.
- [5] Pudu Robotics. *BellaBot Premium Delivery Robot* [Online]. Retrieved from <https://www.pudurobotics.com/product/detail/bellabot>
- [6] PB Tech. (n.d.). *Pudu Premium Series Bella Bot Smart delivery robot Kitten-shaped with AI voice and Expression features for friendly and improved Human Robot interaction experience*. PBTech. Retrieved from <https://www.pbtech.co.nz/product/ROBPDUBB001/Pudu-Premium-Series-Bella-Bot-Smart-delivery-robot>
- [7] J. Shin, “Position Tracking Techniques Using Multiple Receivers for Anti-Drone Systems,” *Sensors*, vol. 21, no. 1, pp. 35, Jan. 21.
- [8] M. A. Hinterman *et al.*, “Fully Autonomous Drone Flights,” U.S. Patent Application US 2023/03500431 A1, Nov. 2, 2023.
- [9] D. G. Fliszar and R. D. Patton, “Stabilization and Navigation of an Autonomous Drone,” U.S. Patent Application US 2023/0350414 A1, Nov. 2, 2023.
- [10] D. G. Fliszar and R. D. Patton, “Autonomous Drone Navigation Based on Vision,” U.S. Patent Application US 2023/0350434 A1, Nov. 2, 2023.
- [11] STM32-base. (n.d.). STM32F103C8T6 - Blue Pill. STM32-base project. Retrieved from <https://stm32-base.org/boards/STM32F103C8T6-Blue-Pill.html>

- [12] Tolocka, E. (2021). *Programming the STM32 Blue Pill with Arduino IDE. Part 3: Using an FTDI adapter*. Profe Tolocka. Retrieved from <https://www.profetolocka.com.ar/2021/04/21/programando-la-stm32-blue-pill-con-arduino-ide-parte-3-uso-de-adaptador-ftdi/>
- [13] Fahad, S. (2021). STM32 Bootloader, STM32F103C Programming using USB Cable. Electronic Clinic. Retrieved from <https://www.electronicclinic.com/stm32-bootloader-stm32f103c-programming-using-usb-cable/>
- [14] Arduino. (n.d.). *Software*. Arduino. Retrieved from <https://www.arduino.cc/en/software>
- [15] Woodford, C. (2023). *Accelerometers*. Explain that Stuff. Retrieved from <https://www.explainthatstuff.com/accelerometers.html>
- [16] Testbook. (2023). *Accelerometer: Definition, Types, Working Principle, & Uses*. Testbook. Retrieved from <https://testbook.com/physics/accelerometer>
- [17] BYJU'S. (n.d.). *Gyroscope: Definition, Types, Working Principle, Applications*. BYJU'S. Retrieved from <https://byjus.com/physics/gyroscope/>
- [18] TDK InvenSense. (n.d.). *MPU-6050*. TDK InvenSense. Retrieved from <https://invensense.tdk.com/products/motion-tracking/6-axis/mpu-6050/>
- [19] TE Connectivity. (n.d.). *MS5611-01BA03 high resolution pressure sensor* [Data sheet]. TE Connectivity. Retrieved from https://www.te.com/commerce/DocumentDelivery/DDEController?Action=showdoc&DocId=Data+Sheet%7FMS5611-01BA03%7FB3%7Fpdf%7FEnglish%7FENG_DS_MS5611-01BA03_B3.pdf%7FCAT-BLPS0036
- [20] Martinez, J., & Rodriguez, A. (2021). Simultaneous Execution of Quad and Plane Flight Modes For Efficient Take-Off of Quad-Plane Unmanned Aerial Vehicles. ResearchGate. Retrieved July 1, 2024, from https://www.researchgate.net/publication/352208829_Simultaneous_Execution_of_Quad_and_Plane_Flight_Modes_For_Efficient_Take-Off_of_Quad-Plane_Unmanned_Aerial_Vehicles
- [21] Alex. (n.d.). *Flysky Radios and Receivers for FPV Quadcopters - Which One Should You Buy?* Unmanned Tech Blog. Retrieved from <https://blog.unmanned.tech/flysky-radios-and-receivers-which-one-should-you-buy/>

- [22] RC Innovations. (n.d.). *FlySky FS-i6X 10ch 2.4Ghz - Remote Control - Includes FS-ia6b Receiver*. RC Innovations. Retrieved from <https://rc-innovations.es/shop/emisora-radio-control-2-4ghz-flysky-fs-i6x-10-canales-telemetry#attr=7151>
- [23] PX4 Dev Team. (2024). *Flying 101*. PX4 User Guide. Retrieved from https://docs.px4.io/v1.11/en/flying/basic_flying.html
- [24] Unmanned Tech. (n.d.). *F450 Quadcopter Frame With Integrated PDB*. Unmanned Tech Shop. Retrieved from <https://www.unmannedtechshop.co.uk/product/f450-quadcopter-frame-with-integrated-pdb/>
- [25] ArduPilot Dev Team. (2024). *Electronic Speed Controller (ESC) Calibration*. ArduPilot Documentation. Retrieved from <https://ardupilot.org/copter/docs/esc-calibration.html>
- [26] Alex. (n.d.). *How To Connect Quadcopter Motors and ESC*. Unmanned Tech Blog. Retrieved from <https://blog.unmanned.tech/how-to-connect-motors-and-esc/>
- [27] DroneCenter. (2015). *Hélices o Propelas*. DroneCenter Blog. Retrieved from <https://dronecenter.blogspot.com/2015/11/helices-o-propelas.html>
- [28] Amazon. (n.d.). *High Precision CNC Machined Parts Kit for DIY Projects*. Amazon. Retrieved from <https://www.amazon.com/gp/aw/d/B0734CF553/>
- [29] Crystal Instruments. (2024). *Rotor Balancing*. Crystal Instruments. Retrieved from <https://www.crystalinstruments.com/rotor-balancing>
- [30] RC Product. (n.d.). *Turnigy 2200mAh 3S 30C Lipo Pack*. RC Product. Retrieved from <https://www.reproduct.in/product/turnigy-2200mah-3s-30c-lipo-pack/>
- [31] Arduino. (n.d.). *Arduino Uno WiFi Rev2 [Datasheet]*. Octopart. Retrieved from <https://octopart.com/datasheet/arduino+uno+wifi+rev2-arduino-116872670>
- [32] Amazon. (n.d.). *Microcontroller Board ATmega328P ATMEGA16U2 Compatible*. Amazon. Retrieved July 1, 2024, from <https://www.amazon.es/Tarjeta-Microcontrolador-ATmega328P-ATMEGA16U2-Compatible/dp/B01M7ZB2B4/>
- [33] Stack Exchange. (2014). *What's the difference between RF options (WiFi, XBee, NRF24L01)*. Arduino Stack Exchange. Retrieved from <https://arduino.stackexchange.com/questions/3270/whats-the-difference-between-rf-options-wifi-xbee-nrf24l01>
- [34] DFRobot. (n.d.). *APC220 Datasheet*. DFRobot. Retrieved July 1, 2024, from https://image.dfrobot.com/image/data/TEL0005/APC220_Datasheet.pdf

- [35] Circuits DIY. (2023). *Interfacing APC220 RF Transceiver Module with Arduino*. Circuits DIY. Retrieved from <https://www.circuits-diy.com/interfacing-apc220-rf-transceiver-module-with-arduino/>
- [36] National Instruments. (2024). *PID Theory Explained*. NI. Retrieved from <https://www.ni.com/es/shop/labview/pid-theory-explained.html>

ANEXO I: ALINEACIÓN DEL PROYECTO CON LOS ODS

Los Objetivos de Desarrollo Sostenible (ODS), fueron adoptados por las Naciones Unidas en 2015, y establecieron un compromiso mundial para afrontar desafíos antes del 2030. Entre sus objetivos principales se encuentran la lucha contra la pobreza, la preservación del medio ambiente y la búsqueda de la paz y bienestar en el mundo.

En concreto, se trata de 17 objetivos interrelacionados que sostienen que para lograr un desarrollo sostenible éstos se deben abordar de manera integral y conjunta. Además, fomentan un enfoque equilibrado centrado en los ámbitos social, económico y ambiental.

Asimismo, a través de los ODS se garantiza un avance conjunto entre todos los países, promoviendo que ningún país se quede atrás en el camino hacia un 2030 más justo y sostenible para todos.

ODS 3: Salud y Bienestar

El proyecto presentado contribuye a este ODS porque los drones autónomos tienen el potencial de realizar tareas peligrosas, mejorando la seguridad de los trabajadores y reduciendo así el riesgo de accidentes laborales.

Además, los drones autónomos también podrían ser utilizados para la entrega rápida de suministros médicos en áreas de difícil acceso, mejorando la atención sanitaria en situaciones de emergencia.

ODS 8: Trabajo Decente y Crecimiento Económico

La investigación, desarrollo e implementación de tecnología de drones inteligentes promueve la creación de empleos relacionados con la alta tecnología, impulsando así el crecimiento económico sostenible. Paralelamente, al automatizar tareas repetitivas, es posible mejorar la eficiencia al liberar recursos humanos para tareas más creativas y estratégicas.

ODS 9: Industria, innovación e Infraestructura

Por otro lado, mediante el desarrollo de drones autónomos capaces de operar en entornos complejos y desafiantes, este proyecto apoya la innovación en la industria de semiconductores. Además, la integración de sistemas de navegación avanzados y el análisis de datos en tiempo real implementado en el sistema de dron autónomo promueve la adopción de tecnologías avanzadas en procesos industriales.

ODS 11: Ciudades y Comunidades Sostenibles

En cuanto a las ciudades y comunidades, la implementación de drones autónomos puede aportar a la gestión eficiente de recursos, ya que facilita el monitoreo y mantenimiento de infraestructuras críticas. Por ejemplo, los drones autónomos pueden ser utilizados para inspeccionar, e incluso reparar, infraestructuras urbanas, reduciendo así el impacto ambiental y mejorando la calidad de vida de todos los ciudadanos.

ODS 13: Acción por el Clima

En cuanto a mitigar los efectos del cambio climático, esto es algo que también se puede lograr por medio de drones autónomos que reduzcan la huella de carbono. El papel de dichos drones por tanto sería recolectar datos precisos sobre el medio ambiente, para posteriormente tomar decisiones informadas para la conservación y gestión sostenible de los recursos naturales.

ODS 17: Alianzas para Lograr los Objetivos

Por último, este proyecto fomenta la colaboración entre diversos organismos: instituciones académicas, el sector privado y organizaciones gubernamentales. Estas alianzas son de vital importancia para compartir conocimientos, recursos y experiencias. Todo ello con el fin de que se logren desarrollar e implementar tecnologías avanzadas de drones, y que ello sea escalable en diferentes contextos y regiones.