

ESCUELA DE INGENIERÍA DE TELECOMUNICACIÓN Y ELECTRÓNICA



TRABAJO FIN DE GRADO

Desarrollo de un entorno hardware-software para el
control de Drones

Titulación: Grado en Ingeniería en Tecnologías de la
Telecomunicación.

Mención: SISTEMAS DE TELECOMUNICACIÓN.

Autor/a: ANDRÉS PÉREZ ORTEGA.

Tutor/a: VÍCTOR A. ARAÑA PULIDO.

Tutor/a: FRANCISCO JOSÉ CABRERA ALMEIDA.

Fecha: 07/2024

Agradecimientos

A mis padres y a mi hermana, por apoyarme durante el transcurso de mi carrera y siempre confiar en mí, enseñándome que siempre tengo que estudiar lo que a mí me gusta. Al resto de mi familia, y en especial a mi abuela por repetirme siempre “Con fe y con ganas como la gente de la Habana “, para animarme a terminar la carrera.

A mis compañeros de carrera, Sergio, Paula y Juanki, por estar siempre ahí cuando los necesito, quiero agradecerles todas las tardes (y noches) que he pasado con ellos disfrutando, que sin ellos tampoco hubiese sido posible. Gracias por siempre sacarme una sonrisa.

A una de la persona que no me esperaba encontrarme este año, Antonio, que sin él esto no hubiese sido posible. Gracias por ser como eres y por enseñarme tanto estos últimos 3 meses.

Por último, pero no menos importante agradecer a mis tutores Víctor y Francis, por toda la ayuda y oportunidades al ofrecerme este TFG.

Índice

Índice de figuras.....	8
Índice de tablas.....	10
1. Introducción.....	15
1.1. Antecedentes.....	15
1.2. Objetivos.....	17
1.3. Estructura de la memoria.....	19
2. Sistemas UAV.....	21
2.1. Introducción.....	21
2.2. Tipos de drones según su estructura.....	22
2.3. Clasificación de Drones.....	23
2.4. Ángulos de navegación.....	26
3. Tecnología LoRa.....	28
3.1. Introducción.....	28
3.2. Long Range (LoRa).....	29
3.3. Parámetros y conceptos fundamentales.....	29
3.3.1. Modulación LoRa.....	29
3.3.2. Spreading factor.....	31
3.3.3. Frecuencia y ancho de banda.....	32
3.4. LoRaWAN.....	35
3.5. Comparativa con otras tecnologías inalámbricas.....	36
4. Software.....	38
4.1. Introducción.....	38
4.2. MAVLink.....	39
4.3. Paquete MAVLink.....	39

4.4.	Pymavlink.....	42
4.5.	DroneKit.....	42
4.6.	Configuración de la comunicación del Chip ESP32	43
4.7.	Funcionamiento del código.....	46
4.8.	Mission Planner y configuración Pixhawk vía Mission Planner	51
4.9.	Software In The Loop	56
4.10.	Pruebas simuladas	57
4.10.1.	Interfaz de las pruebas.....	57
4.10.2.	Prueba del rombo.....	58
4.10.3.	Prueba circular.....	60
5.	Hardware.....	62
5.1.	Introducción	62
5.2.	Arquitectura del Sistema	63
5.3.	Componentes.....	65
5.3.1.	Controladora de vuelo Pixhawk	65
5.3.2.	Raspberry Pi.....	67
5.3.3.	LoRa LILYGO TTGO T-Beam ESP32.....	68
5.3.4.	Estación Terrena	70
5.3.5.	Emisora Fly Sky FS-TH9X.....	71
5.3.6.	Receptor FRSKY D4R-II.....	72
5.3.7.	Antena 3DR Radio Set.....	73
5.3.8.	Antena GPS 3DR uBlox con brújula	74
5.3.9.	GoPro Hero 3.....	76
5.3.10.	Gimbal.....	77
5.3.11.	Batería Externa	78
5.3.12.	Batería Lipo	78

5.3.13. Motores y Hélices	80
6. Pruebas de campo	81
6.1. Introducción	81
6.2. Primera Prueba de Campo	82
6.3. Segunda Prueba de Campo.....	83
6.4. Tercera Prueba de Campo	86
7. Conclusiones	89
7.1. Resultados y revisión de objetivos.....	89
7.2. Líneas futuras.....	90
Bibliografía	92
Presupuesto	96
P.1. Introducción	96
P.2. Costes materiales.....	96
P.2.1. Materiales hardware.....	97
P.2.2. Materiales Software	98
P.3. Amortización del inmovilizado material	98
P.4. Amortización del material hardware.....	100
P.5. Trabajo tarifado por tiempo empleado	101
P.6. Aplicación de impuestos y coste total.....	102

Índice de figuras

Figura 1: Diagrama de bloques del prototipo.....	18
Figura 2: Estructura según su ala [9].....	22
Figura 3: Tipos de drones según su Hélice [9].....	23
Figura 4: Tipos de clase según la normativa AESA [10].	25
Figura 5: Ángulo de navegación, <i>Yaw</i> [11].....	26
Figura 6: Ángulo de navegación, <i>Pitch</i> [11].	27
Figura 7: Ángulo de navegación, <i>Roll</i> [11].....	27
Figura 8: Ángulo de navegación, <i>Throttle</i> [11].....	27
Figura 9: Representación de pulsos de <i>chirp</i> ascendentes y descendentes [13].	30
Figura 10: Torre de protocolo de LoRa [16].	35
Figura 11: Comparativa entre el rango y el ancho de banda de LoRa respecto a otras tecnologías similares [17].	36
Figura 12: Paquete <i>MAVLink</i> [21].....	39
Figura 13: Diagrama de conexión de pines del módulo ESP32 TTGO T-Beam [25].	44
Figura 14: Diagrama del funcionamiento del <i>chip</i> ESP32.....	46
Figura 15: Diagrama 1 del flujo del funcionamiento del código en Python.	48
Figura 16: Diagrama 2 del flujo del funcionamiento del código en Python.	49
Figura 17: Diagrama 3 del flujo del funcionamiento del código en Python.	51
Figura 18: Configuración del puerto y la velocidad de transmisión.	52
Figura 19: Configuración del firmware.....	53
Figura 20: Calibración del acelerómetro.	53
Figura 21: Calibración del radio control.....	54
Figura 22: Selección de modos de vuelos.....	55
Figura 23: Geo Fence.	55
Figura 24: Ejecución del código en la consola 1.....	57
Figura 25: Simulación QGroundControl, Rombo.	59
Figura 26: Ejecución del código en la consola 2.....	59
Figura 27: Simulación QGroundControl, Circular.	60
Figura 28: Ejecución del código en la consola 2.....	61
Figura 29: Componentes del dron.....	63
Figura 30: Diagrama de la arquitectura del Sistema.....	64

Figura 31: Parte frontal de la Pixhawk.....	65
Figura 32: Partes laterales de la Pixhawk.....	65
Figura 33: Raspberry Pi 4, modelo b [28].	67
Figura 34: Módulo LoRa LILYGO.....	69
Figura 35: Estación base.	70
Figura 36: Emisora Fly Sky FS-TH9X.....	71
Figura 37: Receptor FRISKY D4R-II.	73
Figura 38: Antena 3DR Radio Set, 433 Mhz.....	74
Figura 39: GPS[29]	75
Figura 40: Antena GPS 3DR uBlox con brújula.....	76
Figura 41: Cámara GoPro Hero 3.....	77
Figura 42: Gimbal.....	77
Figura 43: Batería Lipo, TATUU.	79
Figura 44: Motor y Hélices.	80
Figura 45: Primera prueba de campo, Mission Planner.	83
Figura 46: Segunda prueba de campo, QGroundControl.....	85
Figura 47: Tercera prueba de campo a 4,3 km.....	86
Figura 48: Tercera prueba de campo a 500 metros.....	87
Figura 49: Ruta circular con los <i>waypoints</i> spline.....	87

Índice de tablas

Tabla 1: Relación SF y capacidad del receptor [14].....	32
Tabla 2: Especificaciones de Bandas de Frecuencia y Parámetros de LoRa en Diferentes Regiones [15].	34
Tabla 3: Comparativa del rango y potencia transmitida entre diferentes tecnologías inalámbricas.....	37
Tabla 4: Partes de mensajes y <i>bytes</i> de paquetes <i>MAVLink</i> [21].	40
Tabla 5: Características Pixhawk [27].	66
Tabla 6: Características Raspberry Pi 4 [28].	68

Resumen

El uso de drones o UAVs (*Unmanned Aerial Vehicles*) es uno de los campos que mayor evolución está teniendo dentro del campo de la robótica aérea. Este sector abarca infinidad de posibilidades y se puede utilizar en múltiples tareas. Esta es la razón por la que su interés, tanto a nivel usuario como a nivel empresarial, haya experimentado un elevado crecimiento y el futuro de estas tecnologías se presente lleno de posibilidades.

Este TFG (Trabajo Fin de Grado) se basa en la búsqueda de un diseño innovador, capaz de diferenciarse de las herramientas convencionales como los softwares Mission Planner o QGround control. Además, se pretende diseñar un sistema de comunicación independiente, flexible y open-source entre la estación base y el dron, mediante la tecnología LoRa (*Long Range*). El desarrollo de estas herramientas va a facilitar la integración del operativo de drones en plataformas encargadas de dar apoyo en las tareas de vigilancia, en los seguimientos del frente de fuego y en la ubicación de los bomberos que están combatiendo el incendio en tiempo real.

Los resultados obtenidos durante el desarrollo y las pruebas de simulación validaron la funcionalidad del sistema en su conjunto. La implementación de la tecnología LoRa permite que los UAVs operen de manera completamente autónoma, ejecutando misiones predefinidas sin necesidad de intervención humana continua. Esto no solo optimiza la eficiencia, sino que también mejora la seguridad al reducir la dependencia de la supervisión en tiempo real. Esta capacidad de realizar misiones autónomas de manera eficiente y segura resalta el potencial de la tecnología LoRa en el campo de la robótica aérea y sus aplicaciones en situaciones críticas. El sistema desarrollado no solo ofrece una solución flexible, sino que también abre la puerta a futuras mejoras y aplicaciones en el ámbito aéreo.

Palabras Clave: UAVs (*Unmanned Aerial Vehicles*), Dron, LoRa (*Long Range*).

Abstract

The use of drones or UAVs (Unmanned Aerial Vehicles) is one of the most rapidly evolving fields within aerial robotics. This sector encompasses a wide range of possibilities and can be applied to multiple tasks. This is why its interest, both at the user and corporate level, has experienced significant growth, and the future of these technologies appears full of possibilities.

This TFG (Final Degree Project) is based on the search for an innovative design capable of differentiating itself from conventional tools such as Mission Planner or QGroundControl software. Additionally, it aims to design an independent, flexible, and open-source communication system between the base station and the drone using LoRa (Long Range) technology. The development of these tools will facilitate the integration of drone operations into platforms tasked with supporting surveillance tasks, tracking the fire front, and locating firefighters combating the fire in real-time.

The results obtained during the development and simulation tests validated the functionality of the system as a whole. The implementation of LoRa technology allows UAVs to operate completely autonomously, executing predefined missions without the need for continuous human intervention. This not only optimizes efficiency but also improves safety by reducing reliance on real-time supervision. This capability to perform autonomous missions efficiently and safely highlights the potential of LoRa technology in the field of aerial robotics and its applications in critical situations. The developed system not only offers a flexible solution but also opens the door to future improvements and applications in the aerial domain.

Keywords: UAVs (*Unmanned Aerial Vehicles*), Drone, LoRa (*Long Range*).

Acrónimos

Siglas	Descripción
AESA	Agencia Estatal de Seguridad Aérea
CGS	Computer Ground Station
CNAF	Cuadro Nacional de Atribución de Frecuencias
CSS	Chirp Spread Spectrum
DRI	Direct Remote Identification
ESC	Electronic Speed Controllers
IDeTIC	Instituto para el Desarrollo Tecnológico y la Innovación en Comunicaciones
IMU	Inertial Measurement Unit
IoT	Internet Of The Things
ISM	Industrial Scientific and Medical
LGPL	Lesser General Public License
LPWA	Low Power Wide Area
LPWAN	Low Power Wide Area Network
MAC	Media Access Control
NB-IoT	Narrowband Internet of Things
OSI	Open Systems Interconnection
PSK	Phase Shift Keying
Rc	Chip Rate
RF	Radio Frequency
RS	Symbol Rate

SeLiF	Desarrollo de nuevas técnicas para detección y seguimiento de líneas de fuego en incendios forestales basado en sensores térmicos móviles de despliegue rápido
SITL	Software In The Loop
SF	Spreading Factor
SNR	Signal-to-Noise Ratio
TCP	Transmission Control Protocol
UAV	Unmanned Aerial Vehicle
Wi-Fi	Wireless Fidelity

Capítulo 1

1. Introducción

1.1. Antecedentes

En los últimos años, el uso de vehículos aéreos no tripulados o UAV (*Unmanned Aerial Vehicles*) ha crecido exponencialmente, impactando significativamente en diversos ámbitos de la sociedad. Estos dispositivos, comúnmente conocidos como drones, inicialmente se empleaban exclusivamente en aplicaciones militares, pero su versatilidad ha permitido su expansión hacia múltiples sectores, incluyendo la agricultura, la logística, la vigilancia y la gestión de emergencias [1]. La capacidad de los drones para realizar tareas de manera autónoma y eficiente ha proporcionado soluciones innovadoras para la automatización de procesos y la optimización de recursos.

Particularmente, la gestión de incendios forestales se ha beneficiado enormemente de los avances en tecnología de drones. España, uno de los países más afectados por incendios en el sur de Europa, ha visto un promedio de más de 12,000 incendios anuales en la última década, resultando en la quema de aproximadamente 99,000 hectáreas de bosques cada año[2]. A pesar de los sistemas existentes de detección y seguimiento de incendios, como redes de vigilancia terrestre y sistemas de teledetección espacial, estos presentan limitaciones significativas en términos de costos, efectividad y tiempos de respuesta [3].

Las redes de vigilancia terrestre fija requieren altos costos tanto de instalación como de mantenimiento y generan un fuerte impacto visual [4]. Las redes de vigilancia terrestre móvil dependen en gran medida de la experiencia de los profesionales encargados, lo que introduce subjetividad en la localización del incendio y una elevada dificultad de detección en condiciones atmosféricas desfavorables. Por otro lado, la vigilancia aérea enfrenta preocupaciones relacionadas con la seguridad del piloto ante incendios o condiciones meteorológicas adversas. Este problema puede ser mitigado mediante el UAVs, aunque su reducido tamaño puede hacerlos más inestables. Los sistemas de teledetección espacial, aunque útiles, requieren largos tiempos de procesamiento que limitan su utilidad en operaciones en tiempo real [5].

El aumento constante de incendios forestales y su impacto devastador han impulsado la necesidad de desarrollar soluciones tecnológicas avanzadas para la detección temprana, el monitoreo y la gestión eficiente de estos eventos. En este contexto, La División de Ingeniería de Comunicaciones del IDETIC (Instituto para el Desarrollo Tecnológico y la Innovación en Comunicaciones), ha estado a la vanguardia en la investigación y el desarrollo de aplicaciones para drones desde 2011. Entre sus proyectos más destacados se encuentra el "Desarrollo de nuevas técnicas para detección y seguimiento de líneas de fuego en incendios forestales basado en sensores térmicos móviles de despliegue rápido" (SeLiF) [6].

La plataforma Gesfogo, por su parte, ha propuesto el uso de visores termográficos portátiles para la detección y seguimiento en tiempo real de incendios forestales. Estos visores, equipados con tecnología avanzada de geolocalización y conectados a motores para rotación, permiten una monitorización precisa y continua del entorno, facilitando la identificación rápida de focos de incendio y apoyando la toma de decisiones en situaciones de emergencia [7].

A pesar de las soluciones actuales, es evidente que la tecnología de comunicaciones juega un papel crucial en la eficacia de estas aplicaciones. Tradicionalmente, los sistemas de telemetría para drones se basan en tecnologías como Wi-Fi (*Wireless Fidelity*) y RF (*Radio Frequency*) para la transmisión de datos. Sin embargo, estas soluciones presentan limitaciones en términos de alcance y estabilidad de la conexión, especialmente en entornos adversos como los que se encuentran durante un incendio forestal [8].

1.2. Objetivos

El principal objetivo de este TFG consiste en desarrollar e integrar un entorno de comunicación alternativo entre la estación base y la aeronave, permitiendo que ésta vuele de forma autónoma sin la necesidad de un piloto o radio control, con el propósito de utilizarlo en la detección temprana y seguimiento de incendios.

En segundo lugar, se pretende gestionar de manera alternativa el control del dron a través de una plataforma externa. Este enfoque busca diversificar las opciones de control, brindando flexibilidad y redundancia en la unidad de pilotaje de la aeronave no tripulada.

Este entorno propone la implementación de un sistema hardware-software independiente del estándar de telemetría y control del dron. Es importante destacar que la integración existente de la plataforma de Gesfogo y el uso de sistemas independientes de los estándares, justifica el trabajo realizado en el desarrollo de un entorno propio que permita una mayor flexibilidad y adaptabilidad en la gestión de drones. Esta iniciativa busca mejorar la eficiencia en la transmisión de datos entre la estación base y la aeronave, fortaleciendo así la capacidad de comunicación y respuesta del sistema.

Durante la elaboración de este TFG, se han instalado dos placas Raspberry Pi y dos módulos LoRa para la comunicación y transmisión de datos entre la estación base y el dron. A continuación, se detalla la configuración y funcionamiento de cada componente, adaptado a las especificaciones del proyecto.

La estación base está equipada con una Raspberry Pi, a la que se le ha incluido un módulo LoRa y un módem WiFi. Esta configuración permite que la Raspberry Pi transmita datos utilizando tanto el protocolo TCP (*Transmission Control Protocol*) a través de la conexión WiFi como la tecnología LoRa para una comunicación de mayor alcance. Por su parte, el dron también está equipado con una Raspberry Pi y un módulo LoRa. Ésta Raspberry Pi está programada para ejecutar el código del receptor, ya sea

mediante el protocolo TCP o la tecnología LoRa, dependiendo de la disponibilidad y la calidad de la conexión en cada momento.

En la Figura 1, se presenta el diagrama de bloques del prototipo, donde se muestra una visión general de los principales módulos que integran el prototipo y su relación con el dron experimental.

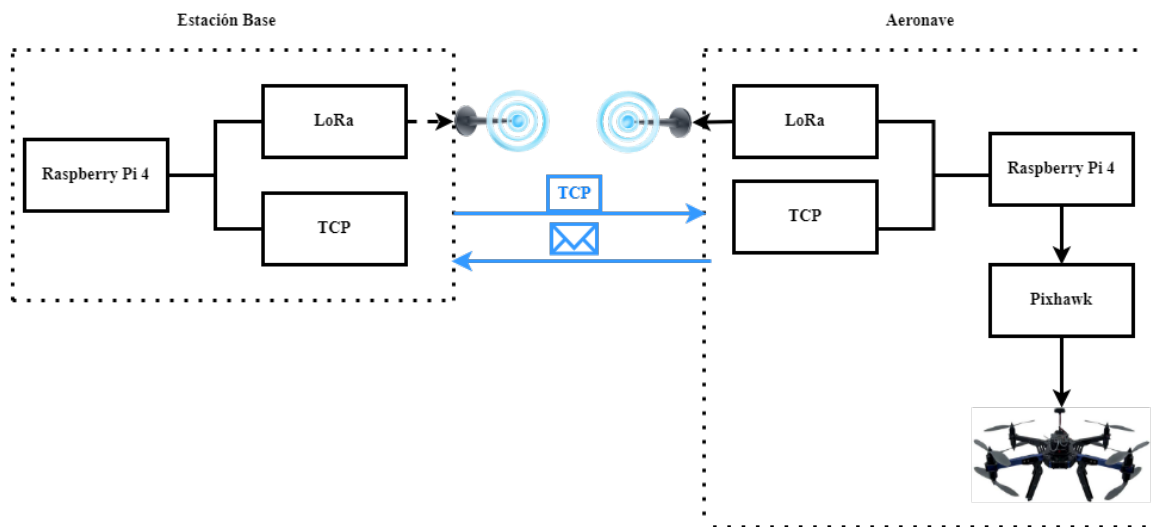


Figura 1: Diagrama de bloques del prototipo.

Esta configuración permite diversificar las opciones de control del dron a través de una plataforma externa, proporcionando una mayor flexibilidad y redundancia en la operación de la aeronave no tripulada.

Para alcanzar estos objetivos, se han utilizado varios softwares clave en diferentes etapas del proyecto:

- **Mission Planner:** este software se utilizó para la calibración inicial del dron, incluyendo la configuración de los sensores y el radio control. Además, se empleó Mission Planner para realizar una misión sencilla con una ruta en forma de cuadrado que incluía cuatro waypoints. La misión se diseñó para evaluar la precisión y estabilidad del dron durante un vuelo autónomo básico.

- **QGroundControl:** esta herramienta se utilizó para la monitorización de telemetría en tiempo real y la visualización de rutas durante los vuelos reales. Además, QGroundControl fue esencial en las pruebas de simulación y en el entorno SITL (*Software In The Loop*).
- **SITL (*Software In The Loop*):** esta plataforma fue fundamental para las simulaciones de vuelo del dron. SITL emula el comportamiento del dron en un entorno virtual, permitiendo realizar pruebas exhaustivas y asegurando que el sistema funcionara correctamente antes de llevar a cabo las pruebas de campo.

1.3. Estructura de la memoria

La memoria de este TFG se organiza en función de seis capítulos, cuya temática se describe de forma breve en los siguientes puntos:

- **Capítulo 1:** en este capítulo se detallan los motivos y antecedentes que justifican el desarrollo de este trabajo. Se proporciona una visión general de la problemática de los incendios forestales y el potencial de los UAVs en su gestión.
- **Capítulo 2:** se introducen los conceptos fundamentales sobre los vehículos aéreos no tripulados. Aquí se describen las distintas clasificaciones de drones, sus aplicaciones, ventajas y limitaciones.
- **Capítulo 3:** este capítulo se centra en la tecnología LoRa. Se describen los conceptos fundamentales de LoRa, su configuración y su integración en el sistema del dron.
- **Capítulo 4:** se abordan los conceptos fundamentales del software utilizado en el proyecto. Se describe el software de planificación y control de misiones, como Mission Planner, y otros programas relevantes para la operación de los drones.
- **Capítulo 5:** se detalla el hardware utilizado en el proyecto. Se describen los componentes del dron, incluyendo el controlador de vuelo Pixhawk, la Raspberry Pi, el módulo LoRa, y otros elementos críticos para la operación eficiente y segura del UAV en misiones.

- **Capítulo 6:** en este capítulo se detallan las pruebas de campo realizadas para validar el sistema. Se describen los procedimientos seguidos, los resultados obtenidos y el análisis de la eficacia del dron en la detección y seguimiento de incendios forestales.

Además, se incluye la bibliografía empleada y el presupuesto que conlleva la ejecución de este TFG.

Capítulo 2

2. Sistemas UAV

2.1. Introducción

Se conoce como UAV o dron a una aeronave que puede realizar diferentes tareas en vuelo sin necesidad de ser manejada por un piloto humano. De esta manera, el usuario puede modificar el movimiento del dron a través de un protocolo de comunicación compartido entre el dron y una estación terrena de control. Además, estos vehículos aéreos suelen clasificarse en función de su estructura aerodinámica en dos tipos principales:

1. **Tipo avión:** estas aeronaves tienen una estructura de ala fija y están orientadas principalmente hacia usos militares debido a su capacidad para cubrir grandes distancias y tiempos de vuelo prolongados.
2. **Tipo multicoptero:** estas aeronaves tienen una estructura de ala rotatoria y se utilizan en ámbitos muy diversos, desde aplicaciones comerciales y recreativas hasta operaciones industriales y de investigación, debido a su maniobrabilidad y facilidad de despegue y aterrizaje vertical.

2.2. Tipos de drones según su estructura

Drones de ala fija

Los drones de ala fija se asemejan a la estructura tradicional de un avión, con un cuerpo central alargado y alas fijas que proporcionan la sustentación necesaria para mantenerse en el aire, como observamos en la Figura 2(a).

Drones de ala rotatoria

Conocidos también como multirrotores, estos drones utilizan múltiples hélices impulsadas por motores situados en sus brazos. Son extremadamente versátiles, capaces de realizar despegues y aterrizajes verticales, y pueden mantenerse en un punto fijo en el aire. Este tipo de drones se emplea en una variedad de aplicaciones, incluyendo filmación y fotografía aérea, inspección industrial, vigilancia, búsqueda y rescate, y espectáculos de luces, como indica en la Figura 2(b).



(a) Dron de ala fija



(b) Dron de ala rotatoria

Figura 2: Estructura según su ala [9].

2.3. Clasificación de Drones

En lo que respecta a los drones tipo multicoptero, es importante señalar que existe una clasificación en función del número de elementos rotatorios que presentan, como observamos en la Figura 3. Además, según la normativa de la AESA (Agencia Estatal de Seguridad Aérea) en España, los drones se clasifican en diferentes categorías basadas en su peso y capacidades. En este apartado, se describen ambas clasificaciones y se presenta el tipo de dron utilizado en este proyecto.

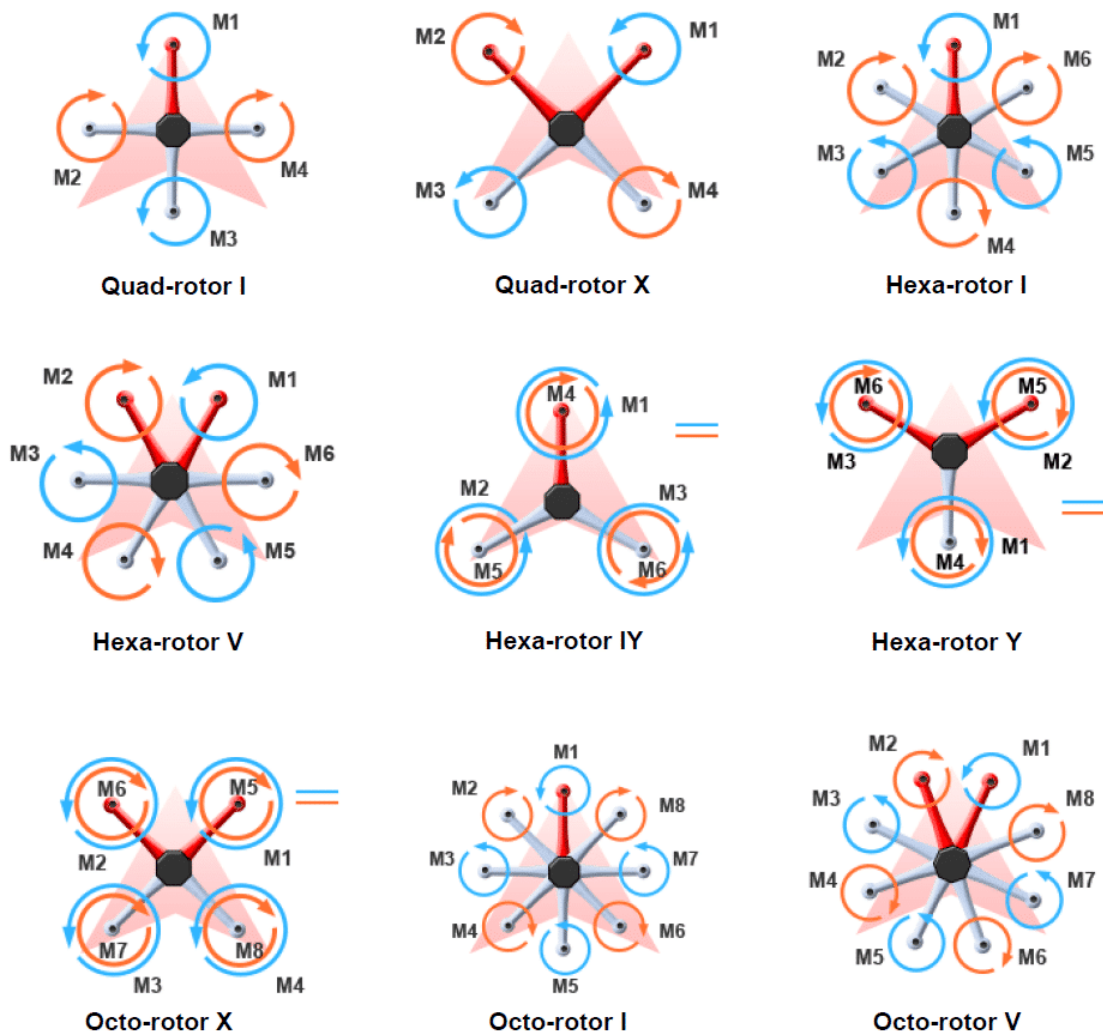


Figura 3: Tipos de drones según su Hélice [9].

Tricóptero

- **Descripción:** este dron cuenta con tres brazos, cada uno con una hélice. Las hélices delanteras giran en direcciones opuestas y la trasera actúa como servomotor para proporcionar estabilidad.
- **Aplicaciones:** es uno de los menos utilizados, generalmente en aplicaciones especializadas que requieren configuraciones específicas de vuelo.

Cuadricóptero

- **Descripción:** el tipo más común, compuesto por cuatro brazos, cada uno con un motor. Dos hélices giran en sentido horario y las otras dos en sentido antihorario.
- **Aplicaciones:** ideal para una amplia gama de usos, tanto recreativos como profesionales, incluyendo fotografía aérea, videografía y misiones de reconocimiento.

Hexacóptero

- **Descripción:** con seis brazos y motores, estos drones ofrecen una mayor estabilidad y seguridad en vuelo.
- **Aplicaciones:** ampliamente utilizado en trabajos profesionales que requieren una mayor capacidad de carga y redundancia en el sistema de propulsión, como inspecciones industriales y levantamientos topográficos.

Octocóptero

- **Descripción:** este dron tiene ocho brazos, cada uno con un motor, lo que le proporciona mayor potencia.
- **Aplicaciones:** ideal para misiones que requieren transportar equipos pesados, como cámaras de alta resolución y sensores avanzados. Sin embargo, su mayor peso puede limitar su maniobrabilidad en espacios reducidos.


Clasificación según la normativa de drones de AESA

La AESA (Agencia Estatal de Seguridad Aérea) en España clasifica los drones en diferentes categorías basadas en su peso y capacidades, con el objetivo de regular su uso y garantizar la seguridad tanto del espacio aéreo como de las personas.

A partir del 1 de enero de 2024, todos los drones en la Categoría Abierta deben estar equipados con un DRI (*Direct Remote Identification*), que transmita información como el número de registro del operador, el número de serie del dron, la posición geográfica, la altura, y la velocidad del dron, así como la posición del piloto remoto y el estado de emergencia del UAS[10].

La normativa AESA distingue tres categorías operacionales de drones: Categoría Abierta, Categoría Específica, y Categoría Certificada. Dentro de la Categoría Abierta, se subdividen las siguientes subcategorías: A1, A2 y A3, como indica en la Figura 4.

Principales requisitos a partir 1/1/2024
Categoría abierta



UAS			Operación		Formación
Clase	DRI	MTOM	Subcat.	Restricciones operacionales	Requisitos a pilotos
Construcción privada	✗	< 250 g	A1	<ul style="list-style-type: none"> No se recomienda volar por encima de personas No está permitido el vuelo sobre reuniones de personas 	Familiarización con el manual de usuario facilitado por el fabricante del UAS
Legacy < 250g	✗				
C0	✗				
C1	✓	< 900 g		<ul style="list-style-type: none"> No volar por encima de ninguna persona no participante No está permitido el vuelo sobre reuniones de personas 	<ul style="list-style-type: none"> Familiarización con el manual de usuario facilitado por el fabricante del UAS Prueba de superación de formación en línea
C2	✓	< 4 kg	A2	<ul style="list-style-type: none"> No volar por encima de ninguna persona no participante 30 m de cualquier persona no participante 5 m de distancia si dispone de modo de baja velocidad 	<ul style="list-style-type: none"> Familiarización con el manual de usuario Prueba de superación de formación en línea Declaración de formación autopráctica Certificado de Competencia de Piloto a Distancia
C3	✓	< 25 kg	A3	<ul style="list-style-type: none"> No volar cerca de personas Distancia de 150 m respecto de: <ul style="list-style-type: none"> Zonas residenciales Zonas comerciales Zonas industriales Zonas recreativas 	<ul style="list-style-type: none"> Familiarización con el manual de usuario facilitado por el fabricante del UAS Prueba de superación de formación en línea
C4	✗				
Construcción privada	✗				
Legacy > 250g	✗				

Figura 4: Tipos de clase según la normativa AESA [10].

En este proyecto, se ha utilizado un octocóptero, clasificado dentro de la Categoría Abierta subcategoría A2, según la normativa de AESA. Este dron ha sido equipado y configurado para cumplir con todas las regulaciones establecidas, asegurando su operación segura y eficiente en todas las misiones planificadas.

2.4. Ángulos de navegación

Los movimientos del dron quedan determinados por los ángulos de Euler (*yaw*, *pitch*, *roll*) y el *throttle* que controla la altitud del dron. Los ángulos de navegación se presentan en las próximas figuras, y se usan normalmente para definir la posición de aeronaves, satélites o embarcaciones. Estos movimientos cumplen las siguientes características:

- ***Yaw***: También llamado guiñada, es el movimiento angular sobre el eje vertical perpendicular a la aeronave, y permite conocer el rumbo de esta, como indica la Figura 5.

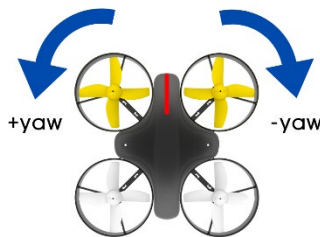


Figura 5: Ángulo de navegación, *Yaw* [11].

- ***Pitch***: También llamado cabeceo, es el movimiento angular sobre el rumbo de la aeronave, que puede marcar su inclinación por delante o por detrás, indicando si se encuentra en ascenso o descenso, como indica la Figura 6.



Figura 6: Ángulo de navegación, *Pitch* [11].

- **Roll:** También llamado alabeo, es el movimiento angular sobre el rumbo de la aeronave, que marca la inclinación a izquierda o derecha de las alas o los rotores, y permite a la aeronave sortear obstáculos o estabilizar su vuelo frente a condiciones ambientales adversas, como indica la Figura 7.



Figura 7: Ángulo de navegación, *Roll* [11].

- **Throttle:** También llamado acelerador, es el movimiento hacia arriba y hacia abajo en el eje Z. El acelerador positivo vuela más alto, y el acelerador negativo hará que vuela más bajo, como indica la Figura 8.

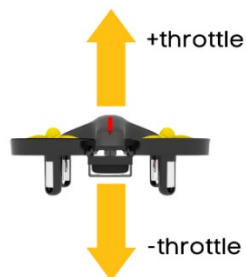


Figura 8: Ángulo de navegación, *Throttle* [11].

Capítulo 3

3. Tecnología LoRa

3.1. Introducción

La historia de LoRa comienza en Francia en el año 2009, cuando Nicolás Sornin y Oliver Sellaer deciden desarrollar una nueva tecnología de modulación de baja potencia y largo alcance. Tras un año de investigación, en 2010 se une al equipo François Sforza y los tres integrantes deciden fundar la empresa Cycleo [12].

En sus inicios, la idea de los desarrolladores era realizar una nueva tecnología orientada al mundo de la industria y la medición de valores como gas o agua. En 2012, la compañía Semtech decide adquirir Cycleo convencida del potencial de las tecnologías de largo alcance y baja potencia, visualizando los múltiples usos y aplicaciones posibles. Gracias a esta compra se mejora la tecnología, finalizan el desarrollo de *chips* y *gateways* necesarios, además de crear el protocolo de red conocido inicialmente como LoRaMAC.

Finalmente, en el año 2015 se fundó la asociación sin ánimo de lucro LoRa Alliance cuyo objetivo es el desarrollo y expansión de la tecnología LoRa, asentándose como uno de los pilares fundamentales de la implementación de IoT (*Internet Of The Things*). Además, se decide renombrar el protocolo de red y crear un estándar llamado LoRaWAN.

Hoy en día, esta alianza sigue trabajando en mejorar la tecnología, buscando nuevas aplicaciones de uso y tratando de que LoRa llegue a todo el mundo.

3.2. Long Range (LoRa)

LoRa es una tecnología de comunicación inalámbrica de nivel físico que permite el envío y recepción de información punto a punto utilizando un radio de cobertura muy elevado, cubriendo como máximo 20 km de alcance gracias al uso de la tecnología CSS (*Chirp Spread Spectrum*) con opciones para distintos SF (*Spreading Factor*) y ancho de banda para optimizar la modulación cumpliendo con requisitos de rango y datos. Muchos sistemas utilizan la modulación por desplazamiento en frecuencia debido a que es muy eficiente para evitar un alto consumo de potencia.

3.3. Parámetros y conceptos fundamentales

3.3.1. Modulación LoRa

La mayoría de las tecnologías inalámbricas usan la modulación PSK (*Phase Shift Keying*), debido a que una de las principales cualidades de esta modulación es su bajo consumo, siendo muy eficiente energéticamente. La tecnología LoRa aporta una gran innovación en este aspecto, ya que utiliza una modulación de espectro ensanchado basada en CSS. Este método se basa en el empleo de pulsos de *chirps* mediante los cuales la frecuencia de la portadora cambia de forma gradual, ascendiendo o descendiendo. Así, el espectro ocupa un mayor ancho de banda, consiguiendo que la modulación sea muy resistente al ruido del canal y al desvanecimiento, aunque se utilicen potencias bajas. De esta manera, se logra que la modulación sea robusta en cuanto a interferencias, permitiendo comunicaciones a largas distancias y siendo muy eficiente energéticamente.

Este tipo de modulaciones se han usado desde hace tiempo en comunicaciones militares y espaciales, pero LoRa es la primera implementación de uso comercial que utiliza esta técnica, como observamos en la Figura 9.

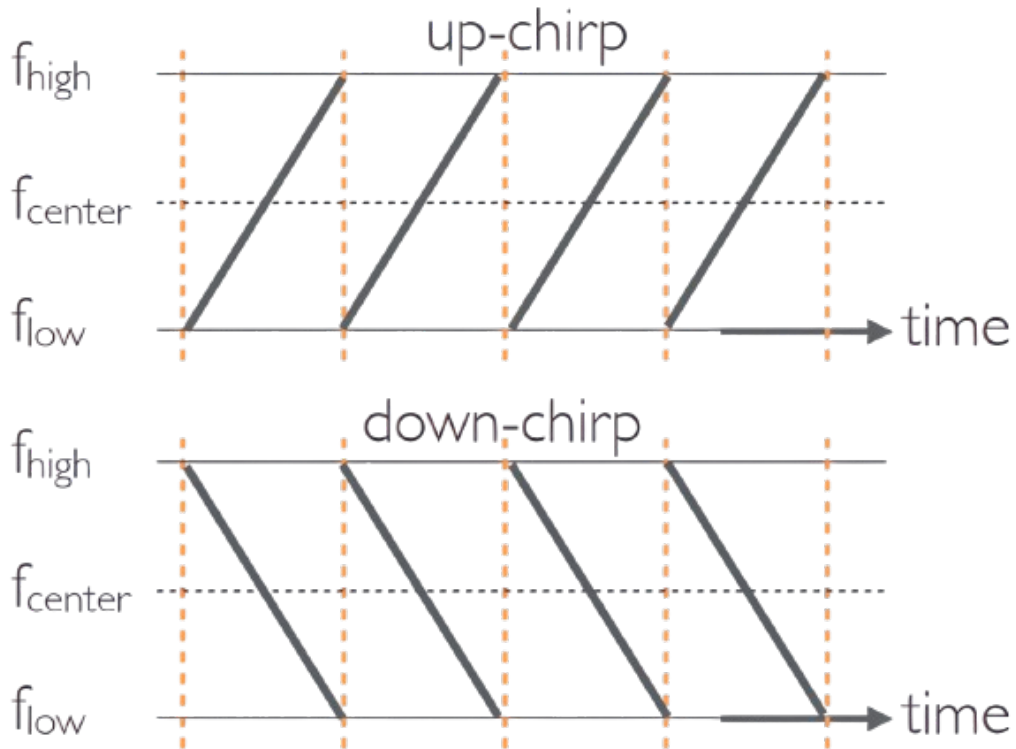


Figura 9: Representación de pulsos de *chirp* ascendentes y descendentes [13].

Cada pulso de *chirp* se descompone en *chips*, que son las unidades fundamentales de información. Estos *chips* se corresponden con los símbolos que aparecen en la palabra de código, la cual está vinculada a la secuencia binaria para expandir el espectro. La RC (Chip Rate) se define como el número de oscilaciones por segundo de la señal de código, y tiene una estrecha relación con el ancho de banda, como indica la ecuación 1.

$$BW = R_c \tag{Ec,1}$$

Donde BW es el ancho de banda.

3.3.2. Spreading factor

La modulación de espectro ensanchado CSS, utiliza un SF para difundir la información. El SF representa el parámetro de propagación del *chirp*, que define cuántos chirps se envían por segundo. Hay seis posibles valores de SF que varían desde SF7 hasta SF12. El SF afecta tanto al tiempo en el aire del símbolo como a la sensibilidad del receptor.

Cuando se aumenta el SF, siendo SF12 el valor máximo, también crece el tiempo en el aire del símbolo, así como el consumo energético. De esta forma, los mensajes se esparcen más en el tiempo, lo que provoca que la comunicación sea más lenta al reducirse la velocidad de transferencia. Sin embargo, al aumentar el SF, la inmunidad de los mensajes al ruido será mayor y aumentará la sensibilidad en el receptor, por lo cual se consiguen comunicaciones a mayor distancia.

Para la tecnología LoRa, un *chirp* se fragmenta en 2^{SF} pasos o chips. Por otro lado, la tasa de símbolo, RS (*Symbol Rate*), es la velocidad a la cual se transmite la información y se vincula con la tasa de chip a través del SF. El SF determina la cantidad de bits empleados para codificar un símbolo, como observamos en la ecuación 2.

$$R_S = \frac{BW}{2^{SF}} \quad (\text{Ec},2)$$

Como se ha comentado anteriormente, el SF determina la SNR, (*Signal-to-Noise Ratio*), afectando a la sensibilidad del receptor como se puede apreciar la ecuación 3.

$$S = -174 + 10 \log_{10}(BW) + NF + SNR \quad (\text{Ec},3)$$

Siendo S la sensibilidad del receptor, NF la figura de ruido del receptor y SNR la relación señal a ruido.

En la siguiente tabla se puede apreciar cómo afectan los diferentes valores posibles de SF a la relación señal-ruido, sensibilidad en el receptor y rango. Para la realización de la Tabla 1 se ha supuesto que el ancho de banda es de 125 kHz, NF es igual a 6 dB y la comunicación se realiza en un entorno rural libre de obstáculos.

<i>Spreading factor</i>	SNR	Sensibilidad del receptor	Rango
7	-7,5 dB	-123 dBm	2 Km
8	-10 dB	-126 dBm	4 Km
9	-12,5 dB	-129 dBm	6 Km
10	-15 dB	-132 dBm	8 Km
11	-17,5 dB	-133 dBm	10 Km
12	-20 dB	-136 dBm	12+ Km

Tabla 1: Relación SF y capacidad del receptor [14].

Los valores mostrados en la tabla anterior son orientativos, pudiendo fluctuar dependiendo de las condiciones de la comunicación. En el caso del rango, los valores con SF máximo pueden llegar a alcanzar los 15 Km en condiciones óptimas. Sin embargo, si el entorno es urbano la comunicación más lejana no superaría los 5 Km de rango utilizando el SF12.

3.3.3. Frecuencia y ancho de banda

En cuanto a las frecuencias, LoRa trabaja en las bandas de radio industriales, científicas y médicas, conocidas como las bandas ISM (*Industrial Scientific and Medical*). Estas frecuencias son libres y no tienen licencia, pudiendo operar a frecuencias por debajo de 1 GHz como 433 MHz, 868 MHz y 915 MHz. Dependiendo de la zona del mundo en la que se encuentre la comunicación sólo se podrá operar en una determinada banda estando otras frecuencias restringidas para su uso, en Europa la banda destinada a

estas comunicaciones es 868 MHz, mientras que en Estados Unidos y otros países se opera a 915 MHz.

Adicionalmente, cada región implementa un cuadro de frecuencia de sub-banda para generar canales de transmisión. A su vez, cada sub-banda se compone de una serie de frecuencias llamadas canales. La utilización de sub-bandas es vital para segregar redes dentro de una misma zona mediante el uso de diferentes frecuencias. También se determina la cantidad de tiempo que un solo dispositivo puede usar el canal.

En el caso de España, el CNAF (Cuadro Nacional de Atribuciones de Frecuencias), se encarga de regular el uso de las bandas de frecuencia y la potencia que se puede transmitir dentro de estas, como observamos en la Tabla 2.

La LoRa Alliance trabaja en colaboración con los diferentes estamentos regionales para confeccionar un cuadro de comunicaciones global con las diversas normativas y reglamentos.

Otro parámetro importante en la tecnología LoRa es el ancho de banda, del cual existen 3 posibles valores, que son 125 kHz, 250 kHz y 500 kHz. El ancho de banda influye en la cantidad de información que se puede transmitir en una determinada cantidad de tiempo. Cuanto mayor sea el ancho de banda, aumentará la tasa de transferencia de datos y se reducirá el tiempo de transmisión. Las relaciones existentes entre los parámetros implicados en la tecnología LoRa están indicadas en las ecuaciones 1 y 2.

Existe una estrecha relación entre el ancho de banda y el SF respecto a la velocidad de las transmisiones. Dependiendo del tipo de comunicación a realizar, se configurarán estos parámetros de un modo u otro. El aumento del ancho de banda provoca una comunicación más rápida, ya que disminuye el tiempo de bit y aumenta la capacidad del canal. Por el contrario, el aumento del spreading factor conlleva una comunicación más lenta debido al aumento del tiempo de bit, pero consiguiendo una mayor robustez frente al ruido.

	Europa	Norteamérica	China	Corea	Japón	India
Banda de Frecuencia	867-869MHz	902-928MHz	470-510MHz	920-925MHz	920-925MHz	865-867MHz
Canales	10	64 + 8	Por definir por el comité técnico	Por definir por el comité técnico	Por definir por el comité técnico	Por definir por el comité técnico
Ancho de Banda Subida	125/250kHz	125/500kHz				
Ancho de Banda Bajada	125kHz	500kHz				
Potencia de Transmisión Subida	+14dBm	+20dBm típico (+30dBm permitido)				
Potencia de Transmisión Bajada	+14dBm	+27dBm				
Factor de Dispersión Subida	07-dic	07-oct				
Tasa de Datos	250bps-50kbps	980bps-21.9kbps				
Estimación de Enlace Subida	155dB	154dB				
Estimación de Enlace Bajada	155dB	157dB				

Tabla 2: Especificaciones de Bandas de Frecuencia y Parámetros de LoRa en Diferentes Regiones [15].

3.4. LoRaWAN

LoRaWAN es un protocolo abierto de red de área amplia LPWA, (*Low Power Wide Area*) diseñado para la conexión inalámbrica de dispositivos de baja potencia permitiendo que se comuniquen con aplicaciones conectadas a internet, más conocido como IoT. Mientras que LoRa representa la capa física de la red habilitando el enlace de comunicación de largo alcance, LoRaWAN es la capa de control de acceso al medio MAC, (*Media Access Control*) a través de la cual LoRa opera y que define el protocolo de comunicación y la arquitectura del sistema para la red, como observamos en la Figura 10.

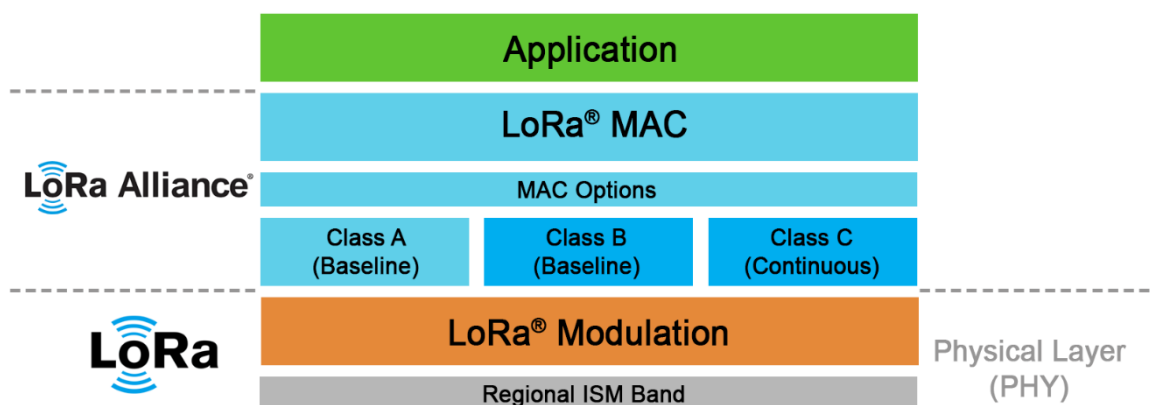


Figura 10: Torre de protocolo de LoRa [16].

En este trabajo se utiliza LoRa en lugar de LoRaWAN por varias razones clave. LoRa permite una implementación más simple y directa de las comunicaciones punto a punto entre el dron y la estación base, reduciendo la complejidad y el consumo energético. Además, proporciona un control total sobre el sistema de comunicación, permitiendo ajustes flexibles y pruebas en un entorno de desarrollo. La latencia es menor en LoRa, lo cual es crucial para el control de drones. Finalmente, la escala del proyecto no requiere la amplia cobertura de LoRaWAN, haciendo de LoRa la opción más adecuada.

3.5. Comparativa con otras tecnologías inalámbricas

Las tecnologías de transmisión inalámbrica pivotan en tres ejes, consumo energético, distancia y capacidad de transmisión. No existe actualmente una tecnología que cubra a la perfección los tres ejes, si no que se sacrifica alguno para potenciar a los demás. Por ejemplo, en el caso de la telefonía móvil ofrecen muy buen alcance, una gran tasa de transmisión, pero consume mucha energía. Otro ejemplo es el *Bluetooth*, ofrece un consumo energético bajo, buena capacidad de transmisión, pero un alcance muy limitado. Las LPWAN, (*Low Power Wide Area Networks*), son tecnologías que, para tener muy buen alcance y un consumo muy bajo, sacrifican la cantidad de datos que pueden transmitir, dentro de las tecnologías LPWAN se encuentra LoRa. Este tipo de redes está en pleno auge, puesto que, permiten una gran cantidad de conexiones mientras que el precio de implementarlas en bandas sin licencia implica mucho menos capital de inversión que otras alternativas. Desde un aspecto general de comparación entre ancho de banda y alcance, LoRa es la tecnología que mayor rango proporciona utilizando un menor ancho de banda, como se puede apreciar en la siguiente Figura 11.

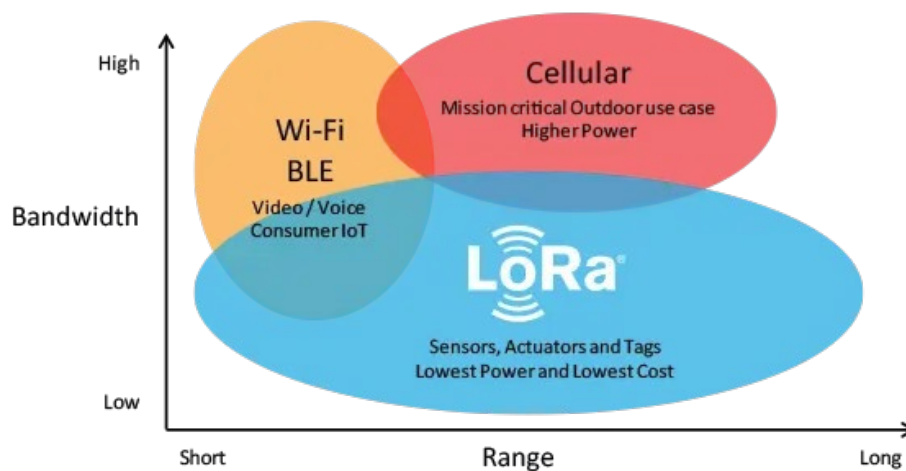


Figura 11: Comparativa entre el rango y el ancho de banda de LoRa respecto a otras tecnologías similares [17].

Si añadimos a la comparación la potencia transmitida como observamos en la Tabla 3, LoRa supera a sus competidores obteniendo un mayor alcance haciendo uso de una menor cantidad de potencia en las transmisiones.

Tecnología	Comunicación inalámbrica	Rango	Potencia transmitida
Bluetooth	Corto alcance	Corto alcance	2.5 mW
Wi-Fi	Corto alcance	Corto alcance	80 mW
3G/4G	Telefonía móvil	Telefonía móvil	5000 mW
LoRa	LPWAN	LPWAN	20 mW

Tabla 3: Comparativa del rango y potencia transmitida entre diferentes tecnologías inalámbricas.

Dentro de las diferentes tecnologías LPWAN, la tecnología LoRa es la más utilizada frente a otras opciones como NB-IoT (*Narrowband Internet of Things*), debido a su rentabilidad y versatilidad. Aunque el ancho de banda de la red LoRa es inferior en comparación con otras tecnologías LPWAN, ofrece una mayor cobertura y alcance, lo que la hace adecuada para un número mayor de aplicaciones. NB-IoT, por su parte, es una tecnología de comunicación que se enfoca en la conexión de dispositivos con baja tasa de transmisión de datos, diseñada para operar en bandas de frecuencia licenciadas, proporcionando una alta capacidad de conexión y eficiencia en el uso de energía. Sin embargo, la implementación de NB-IoT puede resultar más costosa debido a los requisitos de infraestructura y licencias de espectro [18].

Por tanto, se puede concluir que, en la actualidad, para casos donde no se necesite transmitir grandes cantidades de información, la tecnología LoRa es la mejor opción, ofreciendo soluciones eficientes, flexibles y económicas. En aplicaciones donde se requiera una conectividad más robusta y una gestión eficiente del espectro, NB-IoT puede ser la opción adecuada, aunque con un costo mayor.

Capítulo 4

4. Software

4.1. Introducción

El desarrollo de software para la gestión y control de drones ha sido un área de rápido avance en los últimos años, impulsado por la necesidad de mejorar la autonomía, eficiencia y seguridad de estos dispositivos. La historia de los sistemas de control de drones se remonta a principios del siglo XXI, cuando el acceso a hardware accesible y la disponibilidad de software de código abierto comenzaron a transformar el campo de la robótica aérea.

Uno de los hitos importantes en esta evolución fue la creación del protocolo *MAVLink* (*Micro Air Vehicle Link*) en 2009 por Lorenz Meier, entonces estudiante de doctorado en el Swiss Federal Institute of Technology. *MAVLink*, un protocolo de comunicación ligero diseñado específicamente para vehículos aéreos no tripulados, se ha convertido en un estándar en la industria debido a su eficiencia y flexibilidad. Este protocolo facilita la comunicación entre el sistema de control de vuelo y la estación base, permitiendo el intercambio de información de telemetría y comandos de control de manera robusta y confiable [19].

4.2. MAVLink

MAVLink o *Micro Air Vehicle Link* es un protocolo abierto, con licencia LGPL (*Lesser General Public License*), diseñado para la comunicación con todo tipo de UAVs. Como se ha comentado, es compatible con la mayor parte de los controladores de vuelo incluso es compatible con Parrot, aunque éste no lo utiliza como protocolo principal [20].

Este protocolo está formado por un conjunto de mensajes, que van empaquetados en una estructura fija, que indica el tipo de mensaje, su longitud, la identificación del sistema y el componente en enviar el mensaje, el número de secuencia del mismo y un código de comprobación de errores.

De esta forma, el protocolo permite intercambiar todo tipo de información de control velocidad, posición GPS (*Global Positioning System*), modo de vuelo y, también, enviar instrucciones al dron como el cambio de modo, despegue, aterrizaje, etc.

4.3. Paquete MAVLink

Los mensajes de *MAVLink* tienen una cabecera de 6 bytes, dos bytes finales de detección de errores y un *payload* (la información relevante que se transmite en el mensaje) variable de 0 a 255 bytes. La cabecera, que siempre tiene la misma forma, está formada por los siguientes campos, como observamos en la Figura 12.

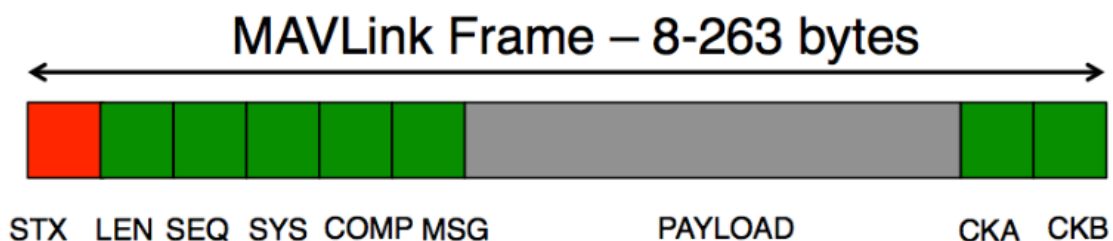


Figura 12: Paquete *MAVLink* [21].

La Tabla 4 describe las partes del mensaje y los bytes:

Partes del mensaje	Byte	Descripción	Valor
Cabecera	0	Indica el inicio de un nuevo mensaje	v1.0: 0xFE (v0.9: 0x55)
Cabecera	1	Indica la longitud del contenido	0 - 255
Cabecera	2	Número de secuencia, de 0 a 255. Cada componente cuenta con una secuencia de envío. Permite detectar pérdida de paquetes	0 - 255
Cabecera	3	ID del sistema de envío. Permite diferenciar diferentes MAVs en la misma red.	1 - 255
Cabecera	4	ID del componente emisor. Permite diferenciar diferentes componentes en el mismo sistema, por ejemplo. e IMU o el piloto automático	0 - 255
Cabecera	5	ID del mensaje (por ejemplo. 0 = heartbeat) - el id define qué significa el contenido y cómo debe ser decodificado.	0 - 255
contenido	6 - (n + 6)	Contenido del mensaje.	(0 - 255) bytes
Checksum	(n+7) to (n+8)	ITU X.25/SAE AS-4 hash, excluido el inicio del mensaje, desde el byte 1..(n+6) Nota: El <i>checksum</i> también incluye <i>MAVLINK_CRC_EXTRA</i> (Número calculado a partir de campos del mensaje. Protege el paquete de ser decodificado de una versión diferente del mismo paquete, pero con diferentes variables)	

Tabla 4: Partes de mensajes y *bytes* de paquetes *MAVLink* [21].

El byte de inicio, cuyo valor hexadecimal es 0xFE, permite detectar la llegada de un nuevo mensaje. Va seguido por el byte de longitud, que indica el tamaño del *payload* del mensaje. A continuación, encontramos el byte de secuencia. Es un identificador del

mensaje. Los dos siguientes bytes hacen referencia al sistema y al componente que envía el mensaje. Son elementales cuando se controlan dos UAV al mismo tiempo, pero no son necesarios en este proyecto. El sexto y último byte de cabecera indica el *ID* del mensaje que se está enviando, y por tanto la forma exacta de su contenido.

Los dos bytes finales constituyen un sistema de detección de errores conocido como *Checksum* o suma de verificación. La idea de este sistema es que el emisor del mensaje envíe el valor calculado de *Checksum* sobre el mensaje original y que el receptor del mensaje calcule por el mismo método el *Checksum* del mensaje recibido, comprobando la integridad de éste al comparar ambos valores. Cuando el *Checksum* calculado no coincide con el enviado, se califica el mensaje como un mensaje corrupto o como ruido. Por tanto, el encapsulamiento del *payload* es siempre de 8 bytes, de forma que el mensaje más largo que puede transmitirse utilizando el protocolo será de 263 bytes.

A la hora de recibir e identificar mensajes, *MAVLink* trabaja con el byte de inicio, la longitud del mensaje y el *Checksum*. En todo momento se encuentra a la espera de recibir el byte de inicio de mensaje. En cuanto recibe un byte con este valor considera que puede estar recibiendo un mensaje, de forma que comprueba que efectivamente sea un mensaje y no ruido. Para ello, lee el byte siguiente, que teóricamente determina la longitud del *payload*. A continuación, calcula el *Checksum* de este teórico *payload*, y lo compara con los dos bytes siguientes. Si coinciden estos dos bytes teóricos de *Checksum* con el que ha calculado, interpreta que realmente ha recibido un mensaje, y decodifica su contenido. En caso contrario, descarta ese teórico inicio de mensaje y considera que es ruido o que ha habido errores en el envío del mensaje. Este funcionamiento es el que da sentido al *byte* de secuencia que aparece en la cabecera de cada mensaje. En todo momento se calculará el ratio de transmisión correcta de mensajes a través de la información sobre mensajes perdidos o descartados que da el *byte* de secuencia. En caso de que se produzcan errores sistemáticos en la transmisión de mensajes por una mala configuración, *MAVLink* enviará un mensaje de alerta informando sobre la inestabilidad de la comunicación.

Cuando *MAVLink* detecta un mensaje y lo acepta según este procedimiento, pasa a decodificarlo. Para ello, en primer lugar, determina qué mensaje está recibiendo, fijándose en la información que le otorga el sexto byte de cabecera, el *ID* del mensaje, para conocer la estructura del *payload*. El *payload* de todos los mensajes de *MAVLink*

está definido, de forma que todos los campos que lo componen son conocidos. Así, se separan todos los campos y se asignan a variables para su posterior uso o almacenamiento.

4.4. Pymavlink

Pymavlink es una librería de Python que facilita al desarrollador la conexión, lectura y envío de mensajes a través de *MAVLink*. Existen otras librerías de un nivel más alto como *droneKit*, que hacen uso de *Pymavlink* para hacer invisible todo el protocolo, facilitando así el desarrollo de código y reduciendo el tiempo necesario para ello.

Pymavlink, en su versión 4.2.1, se puede utilizar con Python 3.5+ y admite las versiones *MAVLink 1* y *MAVLink 2* del protocolo, lo que la hace muy versátil y adaptable a diferentes necesidades de comunicación y control en sistemas UAV. En el contexto de este proyecto, Pymavlink ha permitido la integración efectiva de comunicaciones *MAVLink* para el control y seguimiento del dron, facilitando la implementación de comandos y la recepción de datos en tiempo real [22].

Con *Pymavlink*, se pueden implementar fácilmente scripts personalizados para tareas específicas, como la configuración de *waypoints*, el control de movimientos del dron, y la recopilación de datos de telemetría. Esto ha sido crucial para la configuración de las pruebas de vuelo en simulación y en campo, garantizando que los drones respondan de manera adecuada y segura a los comandos emitidos por la estación de control.

4.5. DroneKit

DroneKit es un proyecto de 3D Robotics, que busca facilitar la comunicación entre el controlador de vuelo y otro dispositivo, como puede ser una CGS (*Computer Ground Station*) o un ordenador de a bordo. Dicho controlador de vuelo debe ser compatible con el protocolo *MAVLink* para que la comunicación pueda llevarse a cabo [23].

Este proyecto ofrece 3 APIs: *DroneKit-Cloud* (un servicio cloud para conectarse al dron y obtener su información), *DroneKit-Python* (orientada a la programación de ordenadores de a bordo) y *DroneKit-Android* (orientada a la creación de aplicaciones para smartphones que funcionen a modo de CGS)[24].

De entre estas tres, la más interesante para este proyecto es la segunda, *DroneKit-Python*, ya que está pensada para la programación de ordenadores a bordo. Esta API para *Python* (que, actualmente, está en su versión 2.0) ofrece la posibilidad de conectarse al dron (tanto a través de cable USB como de telemetría), recibir información del mismo, analizar sus parámetros de vuelo (por ejemplo, su GPS), cargar una misión y guiarlo directamente. Con todo esto, se tiene el control necesario sobre el dron para desarrollar el presente trabajo.

4.6. Configuración de la comunicación del Chip ESP32

El módulo LoRa consta de un transceptor LoRa y el chip ESP32. El transceptor LoRa maneja la modulación y demodulación de señales, mientras que el ESP32 es un controlador integrado que gestiona la lógica de comunicación y ejecución de comandos. La configuración del hardware incluye la correcta asignación de pines para la comunicación SPI (*Serial Peripheral Interface*) entre el ESP32 y el transceptor LoRa.

Los pines más importantes en esta configuración son:

- **RADIO_CS_PIN:** este pin se utiliza para seleccionar el dispositivo con el que se desea comunicar. Activar este pin permite al microcontrolador iniciar la comunicación con el módulo LoRa.
- **RADIO_RST_PIN:** este pin se usa para reiniciar el módulo LoRa. Un reinicio puede ser necesario para asegurar que el módulo esté en un estado conocido y operativo antes de comenzar la transmisión o recepción de datos.
- **RADIO_DIO0_PIN:** este pin es uno de los pines de interrupción del módulo LoRa. Es comúnmente utilizado para indicar eventos como la recepción de un nuevo paquete.

En el código, la función `LoRa.setPins (RADIO_CS_PIN, RADIO_RST_PIN, RADIO_DIO0_PIN)` se utiliza para configurar los pines del módulo LoRa. Esto asegura que el ESP32 sabe qué pines controlar para comunicarse con el módulo LoRa, como observamos en la Figura 13.

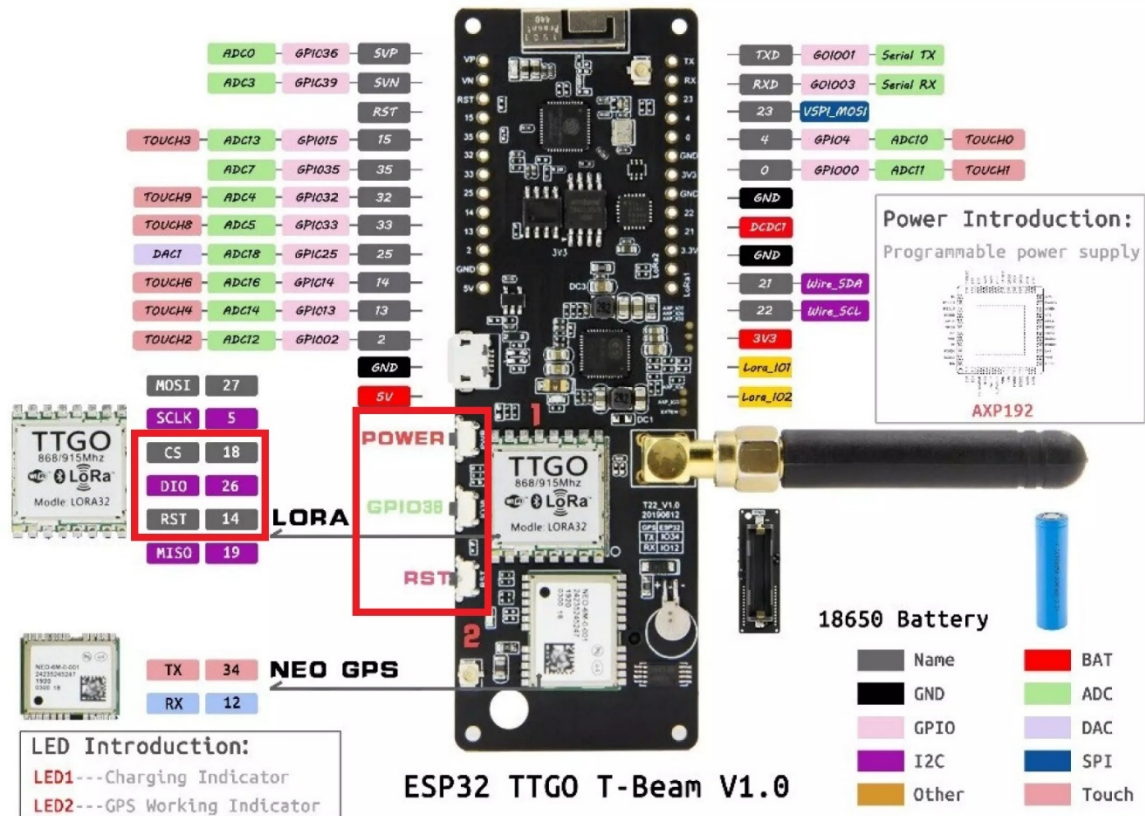


Figura 13: Diagrama de conexión de pines del módulo ESP32 TTGO T-Beam [25].

Envío de Datos

La función `sendLoRaData` es fundamental para la transmisión de datos. Esta función:

1. **Inicia el Paquete:** `LoRa.beginPacket()` configura el módulo LoRa para comenzar un nuevo paquete de transmisión.
2. **Carga los Datos:** la función `LoRa.print(data)` se encarga de cargar la cadena data en el paquete de datos. Este método convierte la cadena `String` en una secuencia de caracteres en formato ASCII y los añade al paquete que será transmitido. Esto permite que los datos, que pueden incluir comandos, mensajes de estado o

cualquier otra información relevante, se envíen de manera correcta y comprensible al receptor.

3. **Envía el Paquete:** *LoRa.endPacket()* finaliza y envía el paquete al aire, utilizando la frecuencia y los parámetros configurados.

Recepción de Datos

Para recibir datos, el ESP32 utiliza un bucle que continuamente verifica la presencia de nuevos paquetes:

1. **Detecta Paquetes:** *LoRa.parsePacket()* analiza el buffer de recepción y detecta la llegada de un nuevo paquete.
2. **Lee los Datos:** si se detecta un paquete, *LoRa.read()* se utiliza en un bucle para leer los datos byte por byte.
3. **Procesa los Datos:** los datos leídos se procesan según la lógica de la aplicación, que puede incluir la actualización de estados o la respuesta con datos adicionales.

La configuración y programación de los módulos LoRa asociados con el ESP32 se implementaron para permitir una comunicación bidireccional, operando de manera similar a una máquina de estados, como vemos en la Figura 14. Esta implementación es fundamental para el envío y recepción de comandos y *waypoints* necesarios para las misiones autónomas del dron.

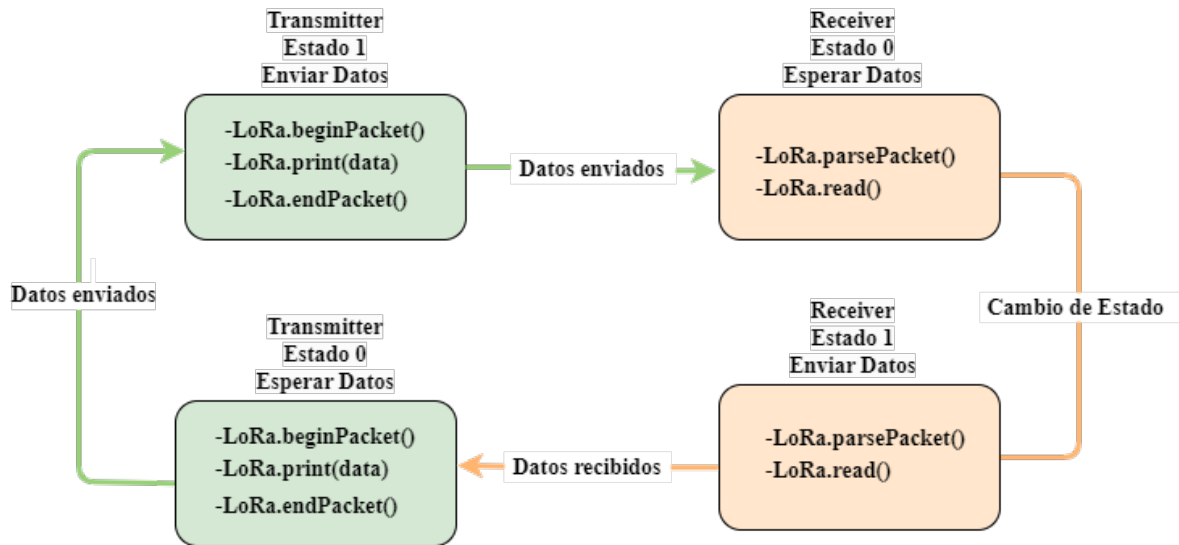


Figura 14: Diagrama del funcionamiento del *chip* ESP32.

El diagrama representa el ciclo continuo de estados en los que operan tanto el transmisor como el receptor. El transmisor comienza en un estado preparado para enviar datos (Estado 1). Una vez que los datos están listos, el transmisor los envía al receptor y cambia su estado a esperar la confirmación (Estado 0). El receptor, inicialmente en un estado de espera (Estado 0), recibe los datos, procesa el paquete, y luego cambia su estado para enviar una confirmación (Estado 1). El transmisor espera esta confirmación antes de enviar nuevos datos, asegurando una comunicación fluida y efectiva.

4.7. Funcionamiento del código

Una vez programado el ESP32 con los módulos LoRa, se procede a explicación detalladamente de tres diagramas de flujo, los cuales representan diferentes aspectos del control de un dron utilizando los protocolos de comunicación TCP y LoRa, así como el uso de los modos *GUIDED* y *AUTO* en el dron, gracias a la librería *dronekit* y el protocolo *MAVLink*. Además, se analizarán las particularidades del código implementado para estos escenarios.

El primer diagrama de flujo, como vemos en la Figura 15, describe cómo se establece la comunicación entre el controlador y el dron utilizando TCP y LoRa. Al inicio del proceso, se presenta al usuario la opción de seleccionar el tipo de conexión. Se presenta un menú interactivo que permite elegir entre TCP y LoRa. Una vez realizada la selección, si el usuario elige TCP, se crea un *socket* TCP que actúa en la capa de transporte del modelo OSI (*Open Systems Interconnection*), asegurando una comunicación confiable entre el cliente y el servidor. Este socket se conecta al servidor mediante la dirección IP y el puerto especificado. Tras establecer la conexión, se envía un mensaje inicial "*Hello Drone*" al servidor y se espera una respuesta "*Hello Andres*" para confirmar la comunicación bidireccional.

En el caso de que el usuario seleccione LoRa, se abre un puerto serie (*/dev/lor0*) que opera en la capa de enlace de datos. El puerto serie se configura una velocidad de transmisión de 115200 baudios, adecuada para asegurar la correcta transmisión de datos. El siguiente paso es la selección y envío del archivo de *waypoints*. Se listan todos los archivos .txt en el directorio de misiones y el usuario selecciona el archivo deseado. Luego, los *waypoints* se leen del archivo y se envían a la Raspberry Pi del receptor, es decir al dron. Una vez enviados todos los *waypoints*, se envía un mensaje de confirmación al servidor para indicar que la transmisión se ha completado exitosamente.

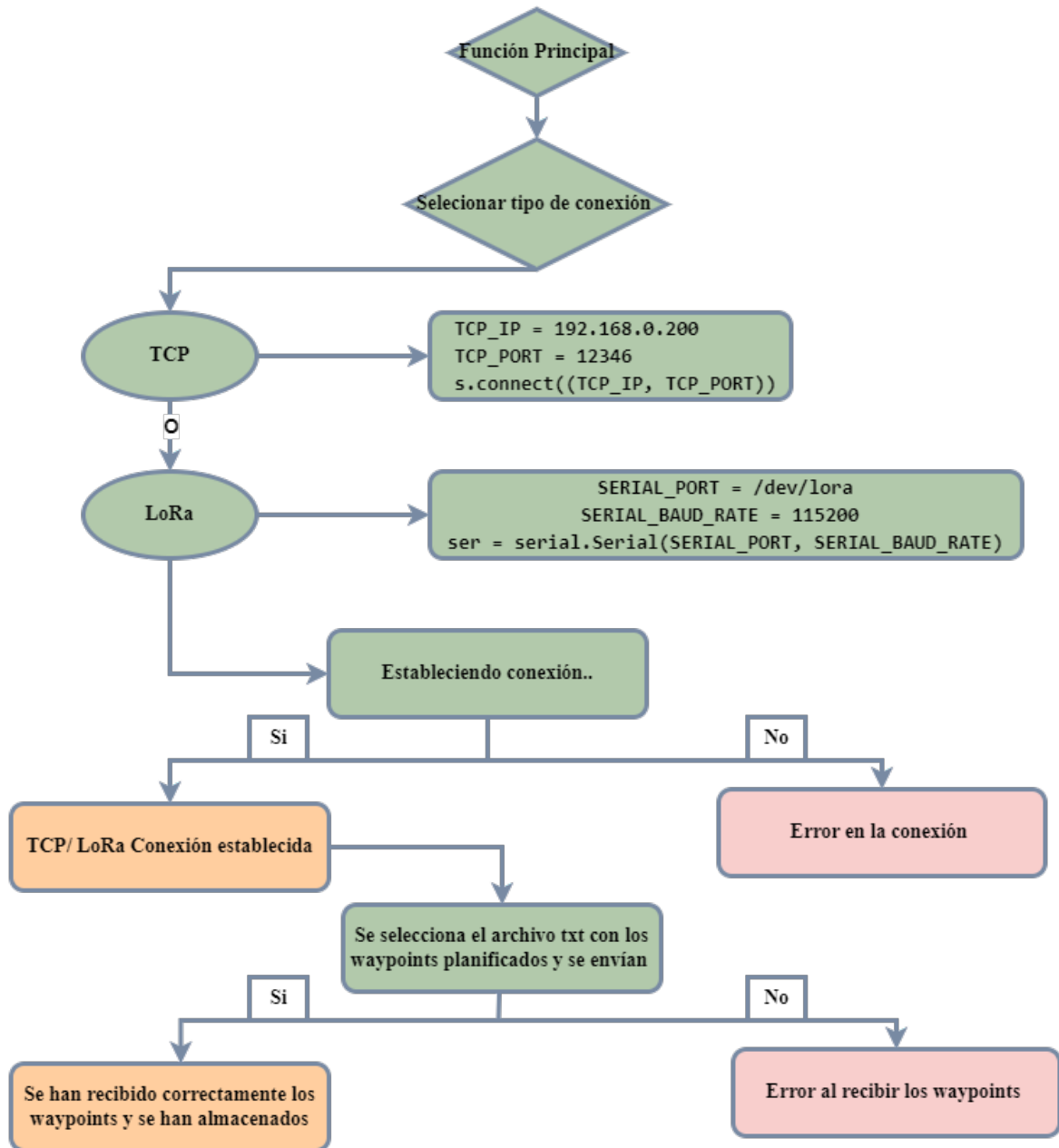


Figura 15: Diagrama 1 del flujo del funcionamiento del código en Python.

El segundo diagrama de flujo, como observamos en la Figura 16, se centra en el control del dron en modo *GUIDED* utilizando la librería *dronekit*. Este modo permite un control manual y directo del dron durante el vuelo. El proceso comienza preguntando al usuario si desea iniciar el vuelo. Si la respuesta es afirmativa, se procede a establecer una conexión con el controlador de vuelo Pixhawk. La conexión con Pixhawk se realiza especificando el puerto y la tasa de baudios. Una vez establecida la conexión, se configura el dron en modo *GUIDED* y se arma el dron.

Una vez armado el dron, se envía una confirmación al servidor. El despegue se realiza ordenando al dron despegar hasta una altura especificada. La navegación a cada *waypoint* se realiza dirigiendo el dron a cada uno de los *waypoints*, calculando la distancia al objetivo y ajustando la trayectoria en tiempo real. Finalmente, se ordena al dron aterrizar utilizando el comando *VehicleMode* ("LAND"). Este proceso asegura que el dron siga las instrucciones precisas y complete la misión de manera controlada y segura.

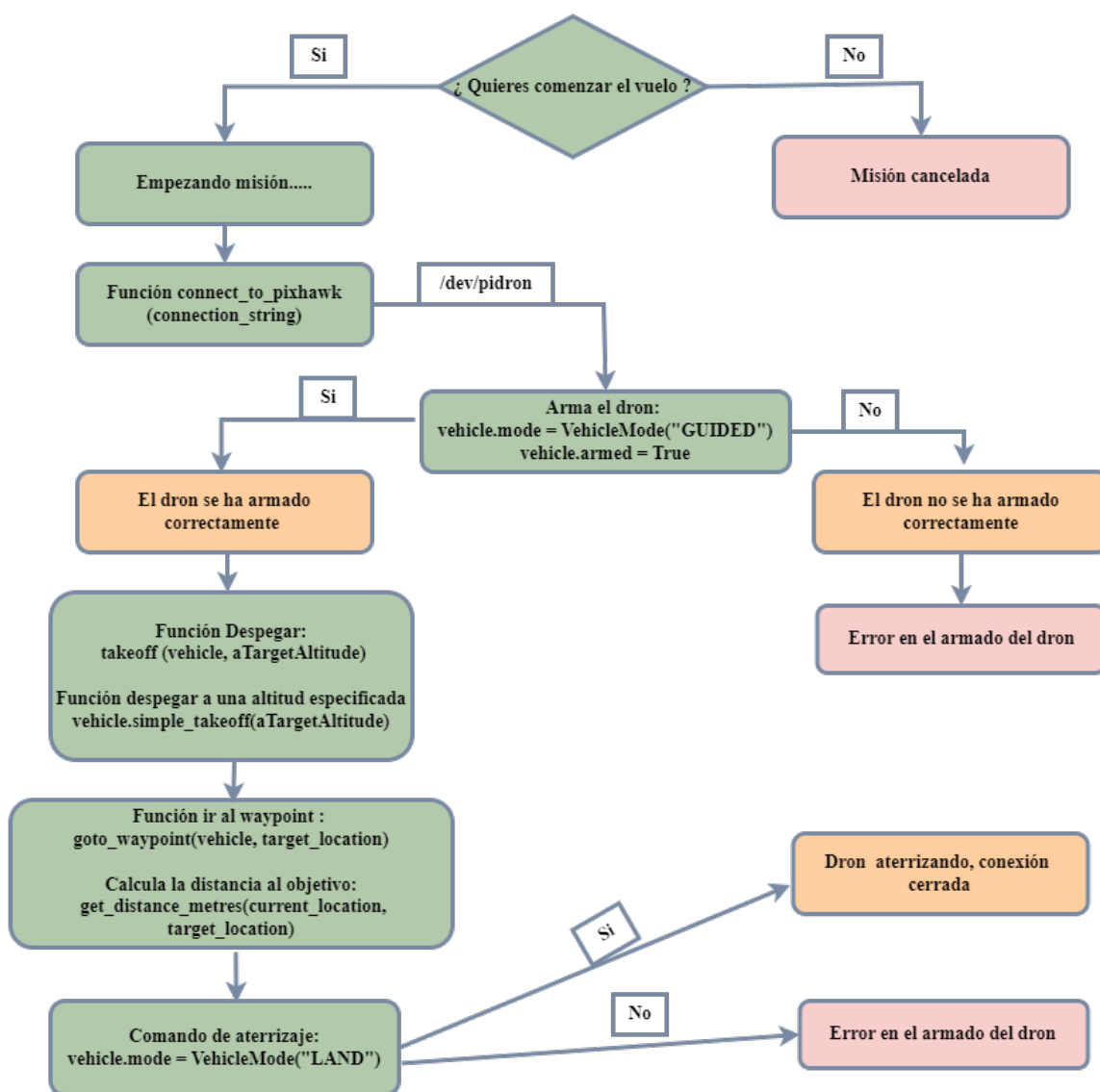


Figura 16: Diagrama 2 del flujo del funcionamiento del código en Python.

El tercer diagrama de flujo, como observamos en la Figura 17, describe el proceso de control del dron en modo *AUTO*, utilizando tanto *MAVLink* como *dronekit*. En este modo, el dron sigue una misión programada sin intervención manual. Al igual que en el modo *GUIDED*, se pregunta al usuario si desea iniciar el vuelo. Si la respuesta es afirmativa, se establece una conexión con el dron.

El dron se arma y cambia al modo *AUTO*, utilizando la función *vehicle.armed* y el comando *MAV_CMD_NAV_TAKEOFF* para despegar. En el modo *AUTO*, se utiliza *MAVLink* para establecer el ROI y los *waypoints spline*, mediante los comandos *MAV_CMD_DO_SET_ROI* y *MAV_CMD_NAV_SPLINE_WAYPOINT*.

Durante la misión, el dron sigue los *waypoints* definidos y, al finalizar, aterriza utilizando el comando *MAV_CMD_NAV_LAND*. Este modo es ideal para misiones de larga duración o donde la ruta está claramente definida y no se requiere intervención humana.

En el modo *GUIDED*, el código utiliza únicamente *droneKit* para gestionar la misión. *DroneKit* abstrae muchos de los detalles de *MAVLink*, proporcionando una interfaz de alto nivel para el control del dron. Los comandos *MAVLink* subyacentes, como *MAV_CMD_COMPONENT_ARM_DISARM*, se envían automáticamente cuando se utilizan funciones como *vehicle.armed = True*.

En el modo *AUTO*, el código combina el uso de *droneKit* y comandos *MAVLink* directos. *DroneKit* se utiliza para establecer la conexión y enviar comandos de vuelo básicos, mientras que los comandos *MAVLink* directos, como *MAV_CMD_NAV_TAKEOFF*, *MAV_CMD_DO_SET_ROI* y *MAV_CMD_NAV_LAND*, se utilizan para una mayor precisión y control sobre la misión predefinida del dron. Este enfoque proporciona una mayor flexibilidad y capacidad de programación avanzada para misiones autónomas complejas.

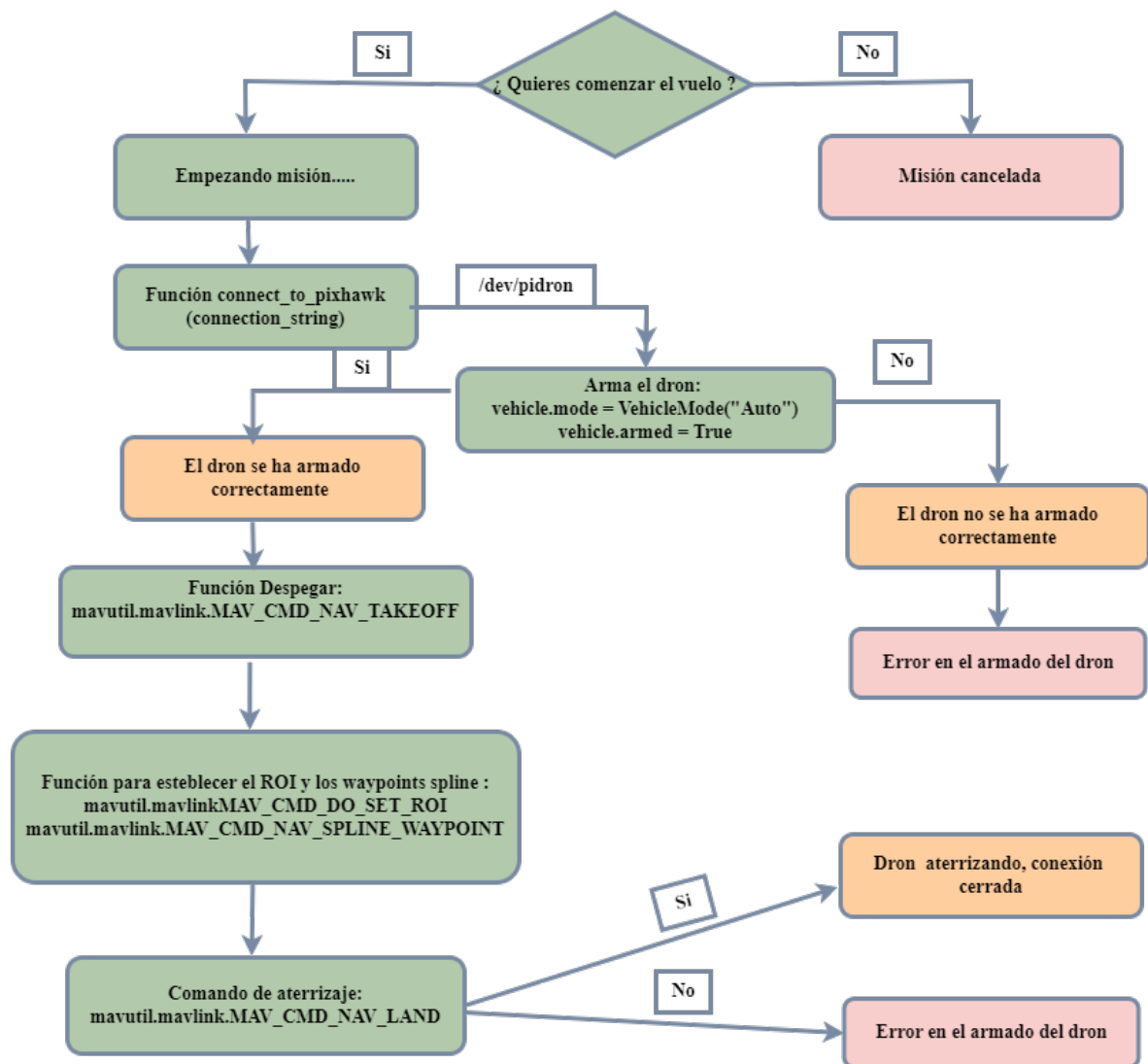


Figura 17: Diagrama 3 del flujo del funcionamiento del código en Python.

4.8. Mission Planner y configuración Pixhawk vía Mission Planner

Mission Planner es una plataforma de software utilizada principalmente para la planificación, control y análisis de misiones para UAVs basadas en el protocolo *MAVLink*. Desarrollada por Michael Osborne, esta plataforma de código abierto ha ganado una gran popularidad en la comunidad de drones debido a su capacidad para

ofrecer una amplia gama de funciones avanzadas, su facilidad de uso y su compatibilidad con numerosos controladores de vuelo, especialmente aquellos basados en ArduPilot. Sin el uso de Mission Planner, la configuración del controlador de vuelo debería realizarse mediante línea de comandos, lo cual es significativamente más complejo y propenso a errores.

La configuración de un controlador de vuelo Pixhawk es esencial para asegurar un funcionamiento correcto y seguro del dron. Para realizar esta tarea, utilizamos el software Mission Planner, que implica varios pasos clave para asegurar que todos los parámetros del dron estén correctamente configurados para un vuelo seguro y eficiente. A continuación, se detalla un proceso paso a paso de manera más detallada.

Una vez instalado Mission Planner, se conecta el Pixhawk al portátil usando un cable USB. Luego, seleccionamos el puerto COM correspondiente al Pixhawk y se establece la velocidad de transmisión a 115200 baudios, como vemos en la Figura 18.

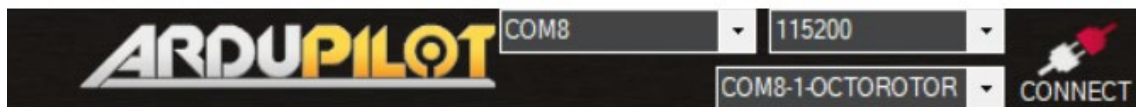


Figura 18: Configuración del puerto y la velocidad de transmisión.

El siguiente paso fue la instalación del firmware adecuado. En Mission Planner, vamos a la pestaña "*Initial Setup*" y seleccionamos "*Install Firmware*" como vemos en la Figura 19. Elegimos el tipo de vehículo en nuestro caso la versión Copter V4.5.4 OFFICIAL para octacópteros y seguimos las instrucciones para cargar el firmware correspondiente en el Pixhawk.

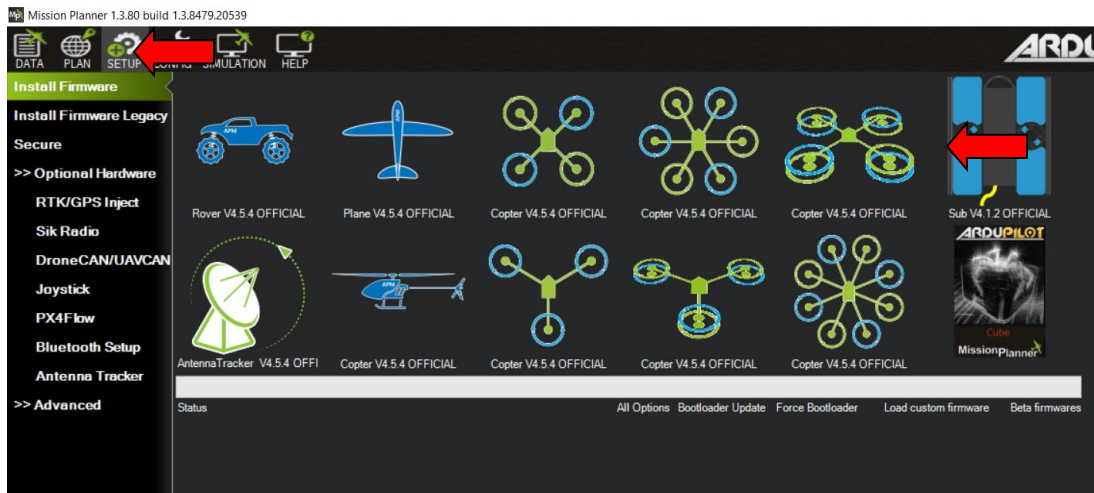


Figura 19: Configuración del firmware.

Posteriormente, procedemos a la calibración del hardware. Primero, la IMU (*Inertial Measurement Unit*), que incluye el acelerómetro y el giroscopio, es crucial para obtener la orientación y movimientos del dron. En la pestaña "Mandatory Hardware", seleccionamos "Accel Calibration" y seguimos las instrucciones para colocar el dron en diferentes posiciones, como vemos en la Figura 20, logrando una calibración precisa. Luego, calibramos el compás seleccionando "Compass" y moviendo el dron en todas las direcciones hasta completar la calibración.



Figura 20: Calibración del acelerómetro.

A continuación, calibramos la radio seleccionando "Radio Calibration". La calibración de la emisora es necesaria para que la aeronave reconozca los límites de movimiento de los controles. En este paso, se mueven todos los controles hasta sus posiciones extremas, asegurando que la controladora de vuelo reconozca los valores numéricos de cada control, como vemos en la Figura 21.



Figura 21: Calibración del radio control.

Finalmente, calibramos los ESCs (*Electronic Speed Controllers*) seleccionando "*ESC Calibration*" y siguiendo las instrucciones para calibrar los motores del dron. La correcta calibración de todos estos componentes es esencial para asegurar un vuelo estable y controlado.

Con el hardware calibrado, configuramos los vuelos y parámetros específicos. En la pestaña "*Flight Modes*", el modo *STABILIZE*, que controla la inclinación de la aeronave para evitar el vuelco; *LOITER*, que mantiene altura y posición cuando no se utiliza la emisora; *AUTO*, que permite programar vuelos desde la estación de tierra (Mission Planner); y *RTL (Return to Launch)*, ver Figura 22, que regresa a la posición de despegue automáticamente. Luego, configuramos los Failsafes en la pestaña "*Initial Setup*", seleccionando "*Failsafe*" y ajustando los failsafes para batería baja, pérdida de señal RC (*Radio Control*), entre otros.

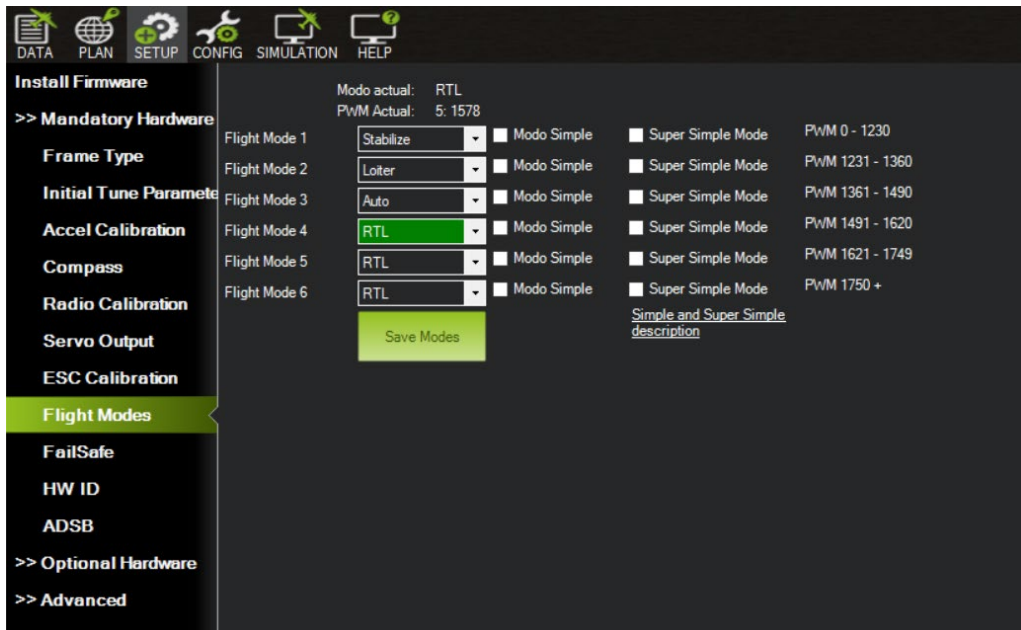


Figura 22: Selección de modos de vuelos.

Por seguridad, se establece un “Geo Fence” para limitar la altura y distancia de vuelo, asegurando que el dron no vuele fuera de los parámetros establecidos. Si se superan estos límites, el dron aterrizará automáticamente en el lugar de despegue. Los parámetros establecidos son una altura máxima de 100 metros, una distancia máxima de 300 metros y una altitud de retorno de 15 metros, ver Figura 23.

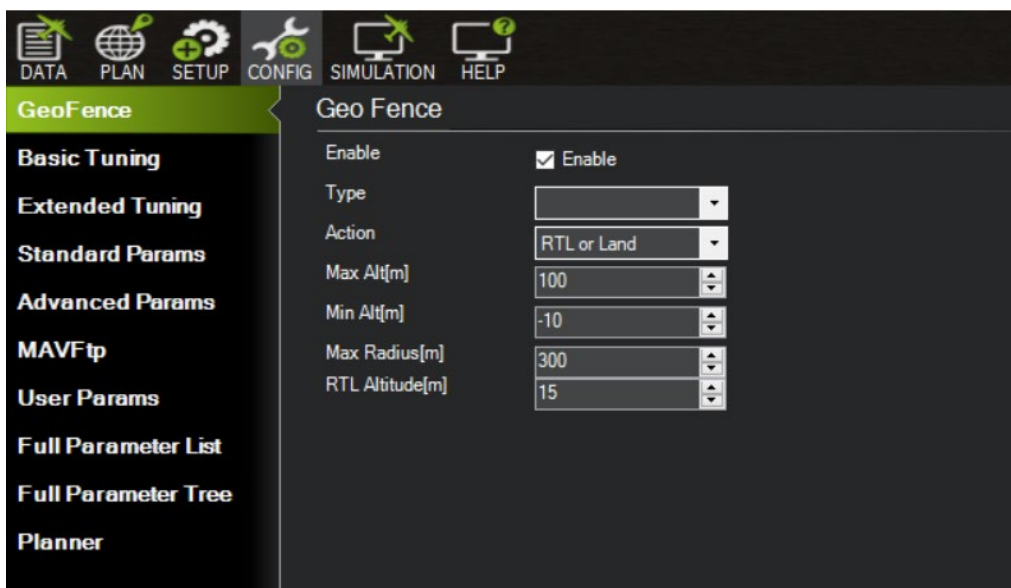


Figura 23: Geo Fence.

La configuración de la controladora de vuelo mediante Mission Planner es un paso esencial para garantizar el correcto funcionamiento del dron. Este proceso asegura que todos los sistemas estén calibrados y configurados adecuadamente, permitiendo un vuelo estable y seguro [26].

4.9. Software In The Loop

En este apartado se documenta las pruebas realizadas para asegurar el correcto funcionamiento del sistema. Se han utilizado escenarios simples, pero suficientemente válidos para comprobar que se obtienen los resultados deseados. En las pruebas, no se utiliza el dron real, sino un simulador llamado SITL. Este proporciona un entorno que emula el comportamiento de un dron real, permitiendo la conexión mediante QGroundControl para observar y controlar el movimiento y el modo de operación del dron.

El uso de SITL es importante en el desarrollo y prueba de sistemas de control para drones, proporcionando una plataforma que simula el comportamiento real del dron. Esto permite evaluar y ajustar los sistemas en un entorno seguro y controlado, sin los riesgos y costos asociados a los vuelos reales. SITL emula tanto el hardware como el software del dron, permitiendo la ejecución de misiones planificadas y el envío de comandos, como si se estuviera operando un dron físico.

Para configurar el entorno de simulación, se siguen varios pasos esenciales. Primero, se instala y configura SITL en el equipo de desarrollo. SITL es parte del paquete de software ArduPilot, que se puede descargar desde su repositorio oficial. Este *software* permite emular diferentes tipos de drones, incluyendo cuadricópteros, hexacópteros y aviones. La configuración de SITL implica especificar las características del dron que se va a emular y asegurar que el *software* esté correctamente integrado con *MAVProxy*.

MAVProxy es una herramienta de línea de comandos que proporciona una interfaz flexible para enviar y recibir datos *MAVLink*, el protocolo de comunicación estándar utilizado por la mayoría de los sistemas de control de drones. Al utilizar *MAVProxy*, se pueden simular diferentes escenarios de vuelo y monitorear el comportamiento del

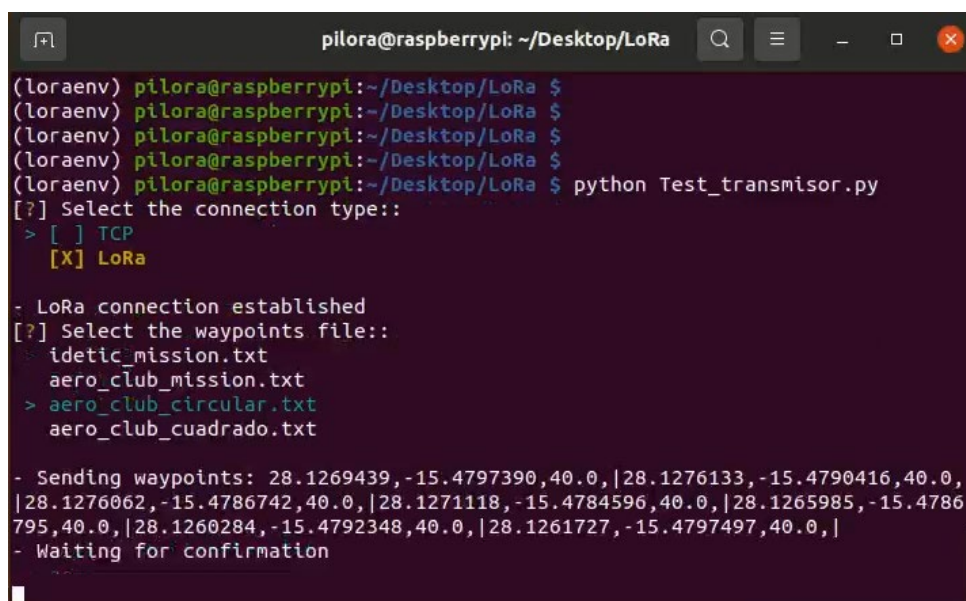
dron en tiempo real. *MAVProxy* facilita una integración fluida entre SITL y QGroundControl, asegurando que todos los componentes del sistema de simulación funcionen de manera coherente y efectiva.

Para cada una de las pruebas se explicará qué es lo que se va a hacer, cuál es su objetivo, cómo llevarla a cabo y los resultados obtenidos.

4.10. Pruebas simuladas

4.10.1. Interfaz de las pruebas

En la Figura 24, se observa la ejecución de un *script Python* en un terminal de consola. Este proceso comienza con la selección del tipo de conexión que se desea utilizar. El sistema presenta un selector que permite elegir entre varias opciones de conexión, como TCP o LoRa, lo que facilita la configuración inicial de las pruebas. Una vez seleccionada la opción deseada, el sistema procede a establecer dicha conexión. La confirmación de que la conexión se ha establecido con éxito se muestra mediante el mensaje "*LoRa connection established*" o "*TCP connection established*". Este paso es importante, ya que asegura que los dispositivos están listos para intercambiar datos.



```
pilora@raspberrypi: ~/Desktop/LoRa
(loraenv) pilora@raspberrypi:~/Desktop/LoRa $
(loraenv) pilora@raspberrypi:~/Desktop/LoRa $
(loraenv) pilora@raspberrypi:~/Desktop/LoRa $
(loraenv) pilora@raspberrypi:~/Desktop/LoRa $
(loraenv) pilora@raspberrypi:~/Desktop/LoRa $ python Test_transmisor.py
[?] Select the connection type::
> [ ] TCP
  [X] LoRa

- LoRa connection established
[?] Select the waypoints file::
  idetic_mission.txt
  aero_club_mission.txt
  > aero_club_circular.txt
  aero_club_cuadrado.txt

- Sending waypoints: 28.1269439,-15.4797390,40.0,|28.1276133,-15.4790416,40.0,
|28.1276062,-15.4786742,40.0,|28.1271118,-15.4784596,40.0,|28.1265985,-15.4786
795,40.0,|28.1260284,-15.4792348,40.0,|28.1261727,-15.4797497,40.0,|
- Waiting for confirmation
```

Figura 24: Ejecución del código en la consola 1.

El siguiente paso involucra la selección del archivo que contiene los *waypoints*, es decir, las coordenadas específicas que el dron debe seguir durante su vuelo. En la consola, se listan varios archivos disponibles y el usuario puede elegir el archivo correspondiente a la misión deseada. Este archivo contiene una serie de puntos predefinidos que conforman una ruta específica. Tras seleccionar el archivo, el *script* lee las coordenadas contenidas en él y las envía al dron a través de la conexión establecida. Estas coordenadas incluyen información de latitud, longitud y altitud para cada *waypoint*. La transmisión de estos datos es fundamental para definir la trayectoria que el dron seguirá de manera autónoma.

Una vez enviados los *waypoints*, el sistema espera una confirmación de recepción por parte del dron. Este mensaje de confirmación es vital para asegurar que todas las coordenadas han sido recibidas correctamente y que el dron está listo para iniciar la misión. Con los *waypoints* confirmados, el sistema está preparado para comenzar la misión. Al recibir la orden de inicio de vuelo, el dron se arma y comienza a seguir la ruta predefinida, moviéndose de un *waypoint* a otro hasta completar la misión.

4.10.2. Prueba del rombo

En la primera simulación, se verificó el funcionamiento correcto de los *waypoints* transmitidos mediante el simulador SITL. Se simuló que los datos de los *waypoints* ya habían llegado y se probó que el código de *droneKit* ejecutara correctamente estos waypoints, asegurando que el dron se armara y volara en la ruta planificada en forma de rombo. El uso de QGroundControl en esta simulación se pudo observar en tiempo real cómo el dron seguía los *waypoints* con precisión, a su vez poder corregir errores y asegurar que los *waypoints* eran precisos, minimizando la posibilidad de errores en futuras pruebas reales. Esta simulación fue crucial para validar el código y preparar el sistema para las pruebas de campo, como vemos en la Figura 25.

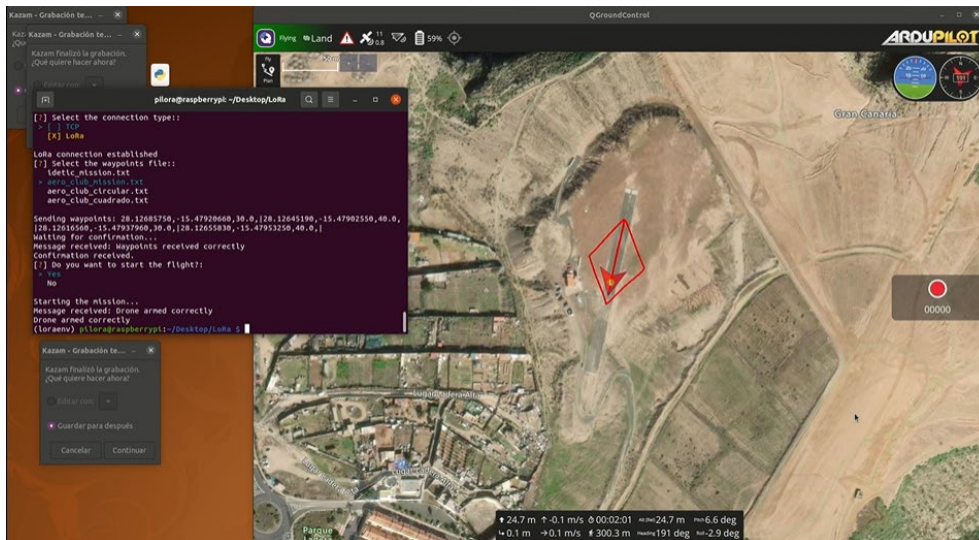


Figura 25: Simulación QGroundControl, Rombo.

Con los waypoints confirmados, el siguiente paso es decidir si se desea iniciar la misión. El sistema pregunta al usuario ¿*“Do you want to start the flight?”*”, ofreciendo la opción de iniciar o no la misión. Al elegir *“Yes”*, el dron se arma y comienza a seguir la ruta predefinida, lo cual se indica en la consola con el mensaje *“Starting the mission...”*. La consola también muestra mensajes de estado como *“Drone armed correctly”* o *“Drone not armed correctly”* para confirmar que el dron está preparado para el vuelo. Una vez que el dron se arma se cierra la conexión de los módulos loRa.

```

pilora@raspberrypi: ~/Desktop/LoRa
[?] Select the connection type::
> [ ] TCP
  [X] LoRa

LoRa connection established
[?] Select the waypoints file::
  idetic_mission.txt
> aero_club_mission.txt
  aero_club_circular.txt
  aero_club_cuadrado.txt

Sending waypoints: 28.12685750,-15.47920660,30.0,|28.12645190,-15.47902550,40.0,|
|28.12616560,-15.47937960,30.0,|28.12655830,-15.47953250,40.0,|
Waiting for confirmation..
Message received: Waypoints received correctly
Confirmation received.
[?] Do you want to start the flight?:
> Yes
  No

Starting the mission..
Message received: Drone armed correctly
Drone armed correctly
(loraenv) pilora@raspberrypi:~/Desktop/LoRa $

```

Figura 26: Ejecución del código en la consola 2.

4.10.3. Prueba circular

La segunda prueba consistió en un vuelo circular, donde se implementó la función ROI (*Region Of Interest*). En esta prueba, el dron debía mantener su cámara centrada en un punto específico durante todo el vuelo circular. Además, cada vez que se alcanzaba un *waypoint*, el módulo LoRa del dron, simulado en el PC, enviaba el siguiente *waypoint* al receptor, asegurando la continuidad del vuelo, como vemos en la Figura 27.

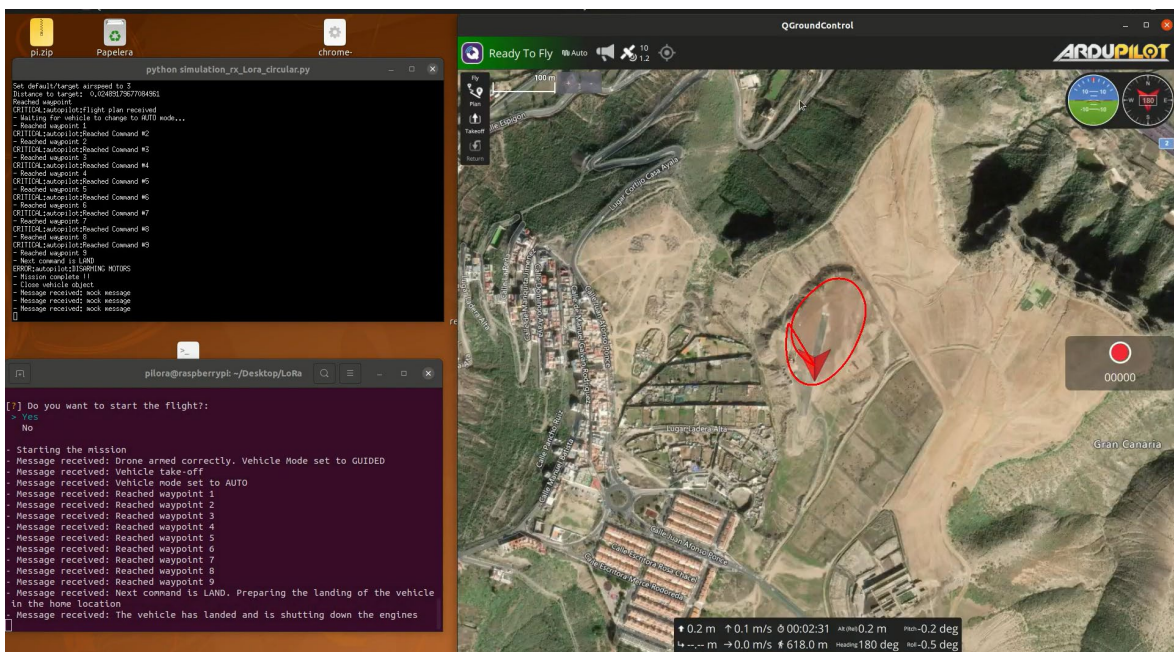
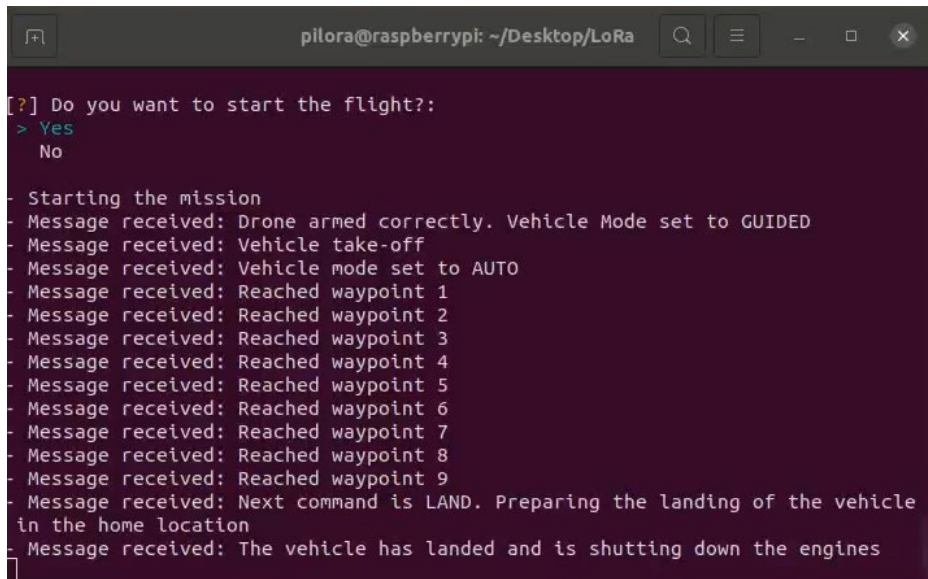


Figura 27: Simulación QGroundControl, Circular.

Además, una mejora significativa en esta prueba fue la implementación de mensajes de estado detallados durante el vuelo. En Figura 28, se observa cómo, después de iniciar la misión, el dron alcanza los *waypoints* sucesivamente, con mensajes como "*Reached waypoint 1*" que indican que la comunicación se mantiene durante todo el vuelo. Al finalizar la misión, se recibe un mensaje de aterrizaje correcto ("*land correctly*") y se concluye la simulación. Esto asegura la continuidad del vuelo y la precisión en la ejecución de los *waypoints* transmitidos mediante LoRa.

En las pruebas anteriores, como la del rombo, la conexión se cerraba una vez que el dron se armaba. Sin embargo, en esta prueba circular, se mejoró la comunicación

manteniéndola abierta durante todo el vuelo y enviando actualizaciones constantes sobre el estado del dron y su posición. Este avance no solo mejoró la precisión de la misión, sino que también aumentó la confianza en el sistema de comunicación LoRa.



```
pilora@raspberrypi: ~/Desktop/LoRa
[?] Do you want to start the flight?:
> Yes
No

- Starting the mission
- Message received: Drone armed correctly. Vehicle Mode set to GUIDED
- Message received: Vehicle take-off
- Message received: Vehicle mode set to AUTO
- Message received: Reached waypoint 1
- Message received: Reached waypoint 2
- Message received: Reached waypoint 3
- Message received: Reached waypoint 4
- Message received: Reached waypoint 5
- Message received: Reached waypoint 6
- Message received: Reached waypoint 7
- Message received: Reached waypoint 8
- Message received: Reached waypoint 9
- Message received: Next command is LAND. Preparing the landing of the vehicle
in the home location
- Message received: The vehicle has landed and is shutting down the engines
```

Figura 28: Ejecución del código en la consola 2.

En ambas pruebas, el entorno de simulación permitió verificar la funcionalidad del código y la correcta interacción entre los componentes del sistema. La simulación en SITL resultó ser una herramienta invaluable para asegurar que los scripts y las configuraciones del dron funcionaran como se esperaba antes de realizar las pruebas en el mundo real. Esto no solo optimizó el tiempo y los recursos, sino que también minimizó los riesgos asociados con los vuelos de prueba.

Capítulo 5

5. Hardware

5.1. Introducción

Desde el siglo pasado, la necesidad de utilizar vehículos de navegación no tripulados ha ido incrementando con el paso del tiempo es por esto surge la necesidad del uso de controladores de navegación que cumplen con funciones como proveer algoritmos de guía o navegación y en algunos casos dependiendo la aplicación que se requiera, establecer una comunicación con estaciones fijas de control como computadoras e incluso dispositivos móviles para el monitoreo de éstos, este módulo que integra las funciones descritas es conocido como autopiloto.

En el desarrollo de este TFG, se ha configurado un dron con una estructura y componentes específicos para cumplir con los objetivos de control y operación planteados. A continuación, se describen en detalle las diferentes partes del dron, su función y su importancia dentro del sistema, como se observa en la ver Figura 29.

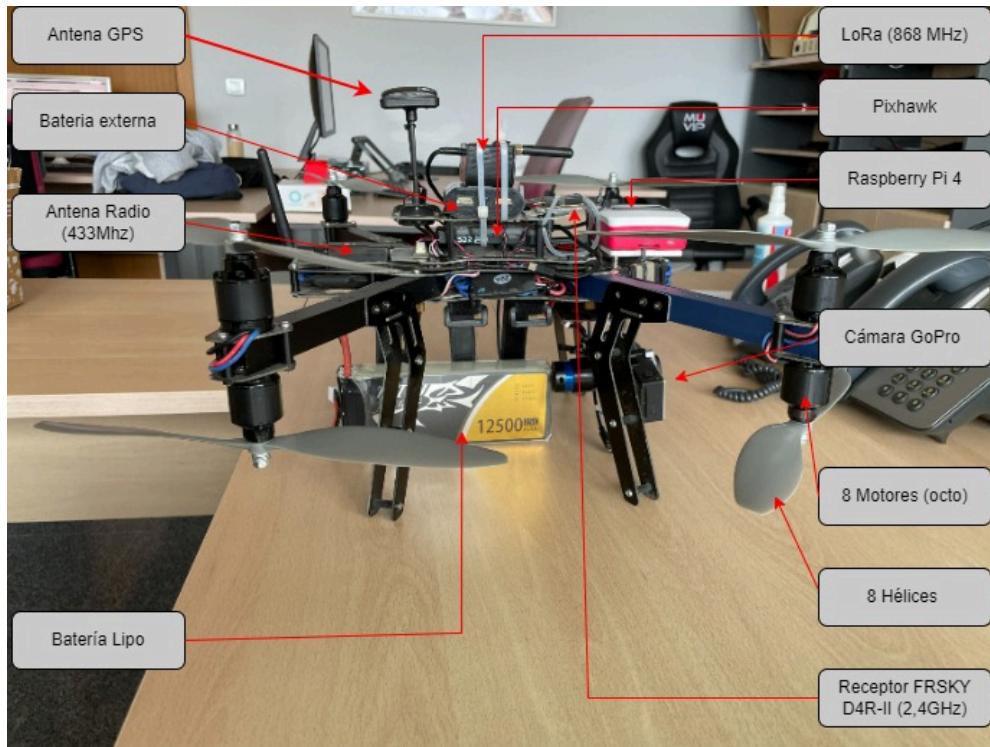


Figura 29: Componentes del dron.

5.2. Arquitectura del Sistema

El diagrama que observamos en la Figura 30 ,ilustra cómo los datos y comandos se envían desde la estación base al dron utilizando tanto TCP como LoRa, dependiendo la disponibilidad de la conexión. La arquitectura está diseñada para asegurar una comunicación eficiente y robusta, permitiendo el control autónomo del dron. A continuación, se describen los componentes y su función dentro del sistema.

En la estación base, compuesta por una pantalla conectada a la Raspberry Pi y a un módulo LoRa o un Módem WiFi, donde se ejecutan los scripts en Python que controlan la comunicación y envían los comandos al dron.

En el dron, la Raspberry Pi recibe los datos y comandos enviados desde la estación base. Ésta Raspberry Pi puede recibir datos tanto a través de TCP como de LoRa, dependiendo de cuál sea la mejor opción en cada momento.

La Raspberry Pi del dron retransmite los comandos a la controladora de vuelo Pixhawk, que es la encargada de ejecutar las acciones necesarias para la misión. La Pixhawk ejecuta el software ArduPilot, que controla los motores y otros componentes del dron.

Las librerías Dronekit y Pymavlink se utilizan para programar y controlar las misiones del dron. Estas librerías permiten una interacción fluida entre la Raspberry Pi y la Pixhawk, facilitando el control de vuelo y la ejecución de misiones.

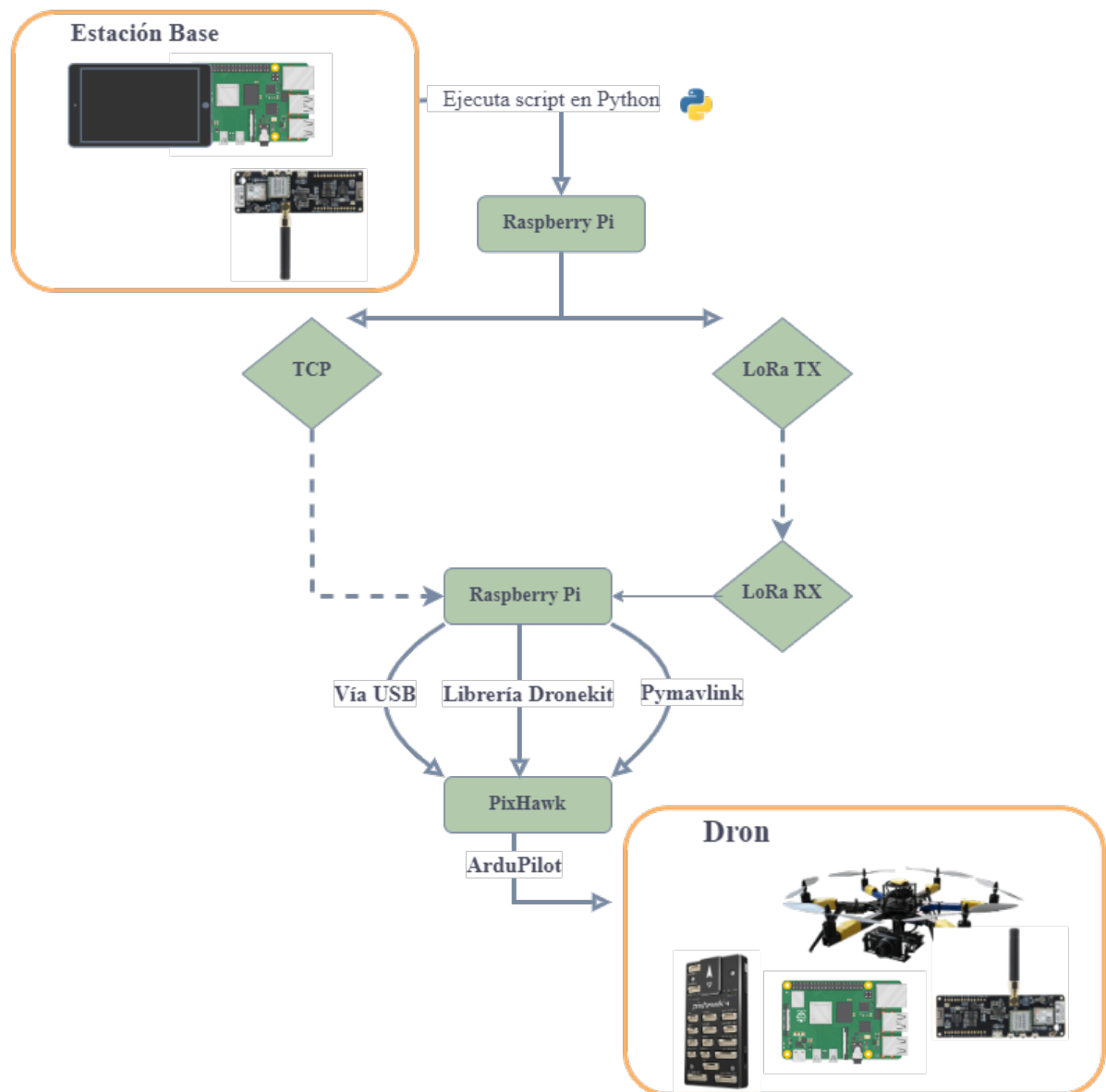


Figura 30: Diagrama de la arquitectura del Sistema.

5.3. Componentes

5.3.1. Controladora de vuelo Pixhawk

La Pixhawk es la controladora de vuelo del dron. Este componente es responsable de la estabilización, navegación y ejecución de las misiones del dron. Integra diversos sensores y módulos que permiten el control preciso y autónomo del dron en diferentes condiciones de vuelo, como observamos en la Figura 31.



Figura 31: Parte frontal de la Pixhawk.

En el controlador de vuelo Pixhawk también se puede encontrar botones en las partes laterales, como observamos en la Figura 32, los cuales son de gran importancia al momento de ser utilizados en conexión con el octocóptero.



Figura 32: Partes laterales de la Pixhawk

En la Figura 32 se observa las partes de los costados de la controladora Pixhawk

- 1) Botón de reinicio de entrada/salida
- 2) Tarjeta SD
- 3) Botón de reinicio de gestión de vuelo
- 4) Puerto Micro-USB

La PixHawk es un dispositivo desarrollado por PX4 de bajo costo y con características competitivas como observamos en la Tabla 5, que lo hacen una excelente opción en términos de relación calidad-precio frente a otros pilotos automáticos. Este dispositivo tiene amplio uso en la comunidad que se dedica al estudio de los vehículos no tripulados. Aunque este piloto automático, mediante el uso de un *software* de telemetría, permite realizar aplicaciones autónomas, no tiene la capacidad física de realizar aplicaciones que requieran un alto procesamiento o el uso de sensores externos. Por ello, utilizando una SBC (*Single Board Computer*) como la Raspberry Pi, se puede utilizar el piloto automático como central de comunicación, dejando como unidad central de procesamiento a ésta Raspberry Pi.

Característica	Especificaciones
Procesador	STM32F765: 32-bit ARM® Cortex®-M7,
Principal FMU	216MHz, 2MB memory, 512KB RAM.
Procesador IO	STM32F100: 32-bit ARM® Cortex®-M3, 24MHz, 8KB SRAM.
Sensores Integrados	Accel/Gyro: ICM-20689 , ICM-20602. Magnetómetro: IST8310. Barómetro: MS5611.
Sistema de Alimentación	Power module input (4.8V to 5.4V). USB power input (4.8V to 5.4V).
Interfaces	14 PWM outputs, 5 UARTs, 3 I2C buses, 2 CAN buses, 2 SPI buses, I2S, 1x Ethernet, 3x Analog inputs, MicroSD card.
Peso	Peso: 15.8g.
Dimensiones	Dimensiones: 50x50x15.5mm.

Tabla 5: Características Pixhawk [27].

5.3.2. Raspberry Pi

La Raspberry Pi 4 es el componente que contendrá toda la lógica para la recepción de datos de los sensores, detección de patrones de eventos, la gestión de los *waypoints* y la ejecución de acciones en respuesta a los mismos. El modelo que se utilizará será una Raspberry Pi 4 Modelo B, como vemos en la Figura 33.

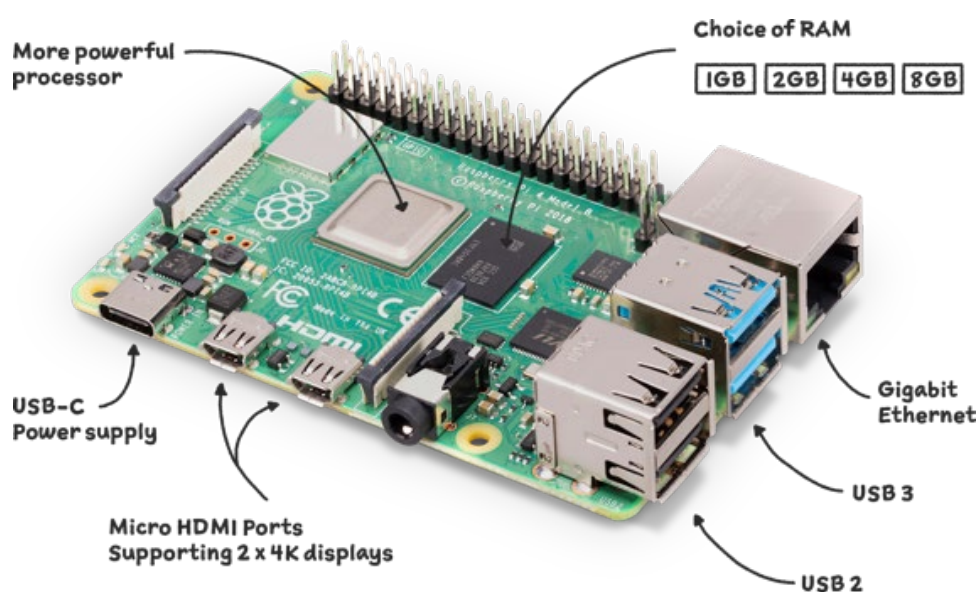


Figura 33: Raspberry Pi 4, modelo b [28].

La Raspberry Pi 4 actúa como el ordenador de a bordo del dron, ejecutando scripts y programas que complementan las funciones de la Pixhawk. Este miniordenador, sin todo el hardware envolvente, permite implementar algoritmos avanzados de control y procesamiento de datos, mejorando significativamente las capacidades del dron, cuyas especificaciones se muestran en la Tabla 6.

Se le añadió una carcasa especialmente diseñada para la Raspberry Pi y se integró en el dron para protegerla en caso de caídas y proyecciones, así como en situaciones de variaciones importantes de temperatura.

Característica	Especificación
Procesador	Broadcom BCM2711, Quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
Memoria	2GB, 4GB o 8GB LPDDR4-3200 SDRAM
Conectividad	2.4 GHz y 5.0 GHz IEEE 802.11ac wireless, Bluetooth 5.0, BLE
Puertos USB	2 puertos USB 3.0, 2 puertos USB 2.0
Video y Sonido	2 puertos micro HDMI (hasta 4Kp60 soportado), puerto MIPI DSI display, puerto MIPI CSI camera, conector de 4 polos estéreo y video compuesto
Almacenamiento	Slot para tarjeta microSD para cargar sistema operativo y almacenamiento de datos
Red	Gigabit Ethernet, 802.3at PoE-capable (requiere PoE HAT)
GPIO	40-pin header GPIO estándar
Alimentación	5V DC vía USB-C connector (mínimo 3A)

Tabla 6: Características Raspberry Pi 4 [28].

5.3.3. LoRa LILYGO TTGO T-Beam ESP32

La LILYGO® TTGO T-Beam es una placa de desarrollo que integra un microcontrolador ESP32 junto con un módulo de comunicación LoRa y un módulo GPS NEO-6M, como vemos en la Figura 34. Esta combinación de componentes hace que la placa sea ideal para aplicaciones de IoT que requieren comunicación de larga distancia y bajo consumo de energía.

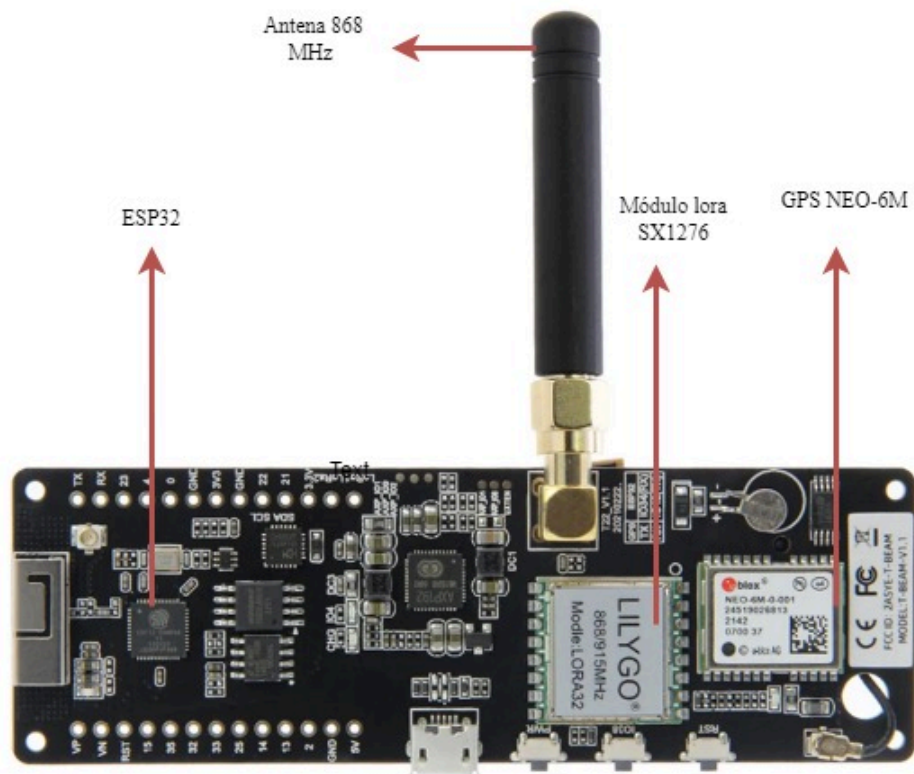


Figura 34: Módulo LoRa LILYGO.

El ESP32 es un microcontrolador de bajo costo y alto rendimiento que proporciona conectividad *Wi-Fi* y *Bluetooth*, permitiendo la comunicación inalámbrica con otros dispositivos. El módulo LoRa integrado en la placa utiliza el chip SX1276 y opera en la frecuencia de 868MHz. LoRa permite la transmisión de datos a largas distancias, con lo cual es ideal para la comunicación entre la estación base y el dron en áreas remotas. Aunque la placa incluye un módulo GPS NEO-6M, en este proyecto no se utiliza este GPS específico, ya que se emplea el GPS integrado en el dron.

La principal función de la LILYGO TTGO T-Beam en este proyecto es conectar la estación base con el dron a través de LoRa, permitiendo la transmisión y recepción de datos de manera eficiente. Esto incluye el envío de comandos, funciones y *waypoints* necesarios para las misiones autónomas del dron. La robustez de la comunicación LoRa garantiza que el dron pueda recibir y ejecutar las instrucciones de manera precisa, para operar de manera autónoma y eficaz en tareas críticas como la detección y seguimiento de incendios forestales.

5.3.4. Estación Terrena

En el proyecto, la estación base juega un papel fundamental para la gestión y control del dron mediante la tecnología LoRa. La estación base está compuesta por una Raspberry Pi 4 conectada a una pantalla táctil, que permite la interacción con el sistema. Esta configuración está montada sobre una plataforma impresa en 3D, proporcionando un soporte estable y personalizado para todos los componentes.

Además, la estación base incluye el módulo LILYGO® TTGO T-Beam ESP32, que opera en la frecuencia de 868 MHz utilizando el módulo LoRa SX1276. Este módulo se encarga de establecer la comunicación inalámbrica de largo alcance con el dron, permitiendo la transmisión y recepción de datos. La Raspberry Pi también está equipada con una batería portátil, asegurando que el sistema tenga una fuente de alimentación autónoma y portátil, cómo vemos en la Figura 35.



Figura 35: Estación base.

La estación base cumple con varias funciones esenciales en el proyecto. Principalmente, se utiliza para enviar los *waypoints* al dron, los cuales son las

coordenadas de los puntos de ruta que el dron debe seguir durante su vuelo. También permite seleccionar el tipo de conexión a utilizar, ya sea TCP o LoRa, dependiendo de la calidad y disponibilidad de la conexión en cada momento.

5.3.5. Emisora Fly Sky FS-TH9X

Debido a que nuestro dispositivo va a ser móvil, necesitamos enviarle órdenes a distancia sin ningún tipo de cables. Para conseguirlo se debe implementar una interfaz de control en un dispositivo a distancia con el que se comunicará mediante un sistema inalámbrico.

El mando RC utilizado para el control remoto del dron pertenece a la empresa FlySky y el modelo utilizado es el que se presenta en la Figura 36.

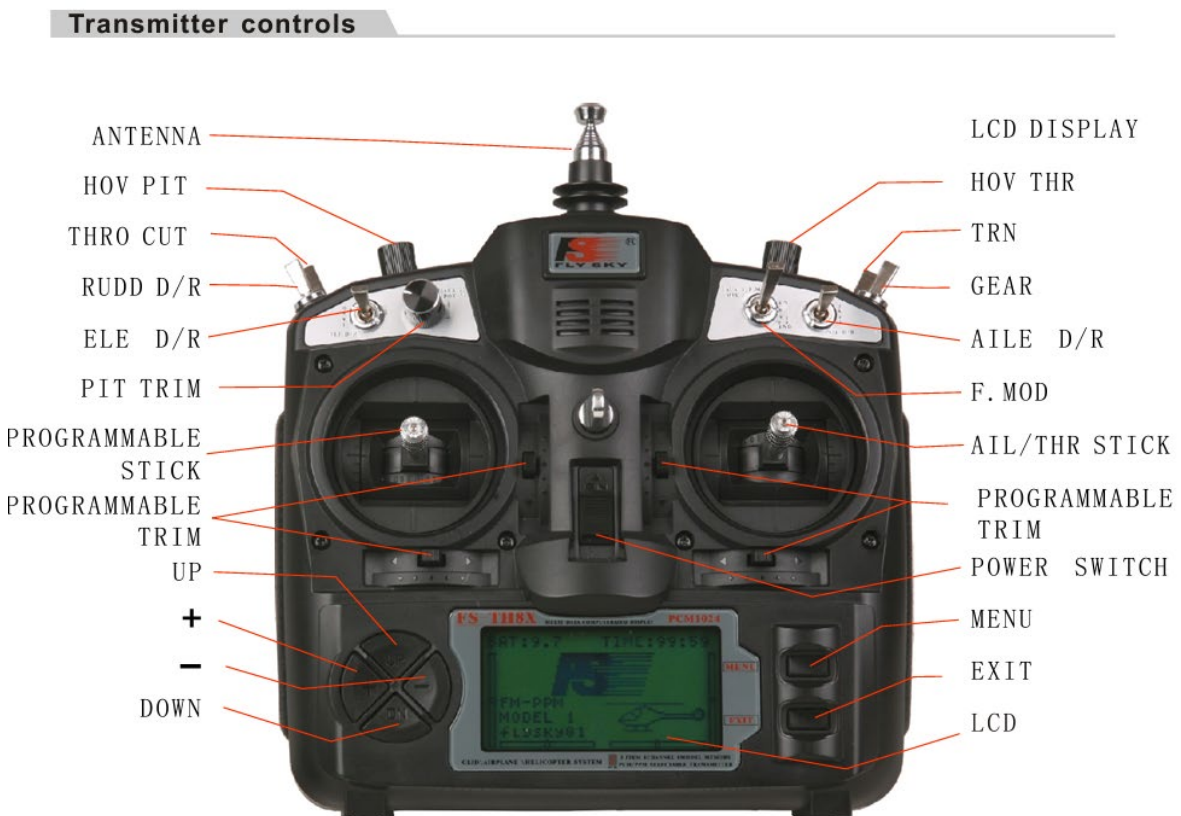


Figura 36: Emisora Fly Sky FS-TH9X.

Se trata de un sistema de 9 canales. La conexión con el receptor, conectado directamente al piloto automático, es vía radio a 2.4 GHz y proporciona al usuario el gobierno total del vehículo en los modos manuales.

En el caso de misiones programadas, en las que el vehículo realizará su misión de forma totalmente autónoma, el mando no jugará ningún papel, pero, como medida de seguridad y si fuera necesario, el operador puede cambiar el modo de vuelo y tomar el control absoluto del dron, asegurando así la capacidad de intervenir y dirigir el vuelo manualmente si las circunstancias lo requieren.

Además, a través del mando se realizan las acciones de armar (*Arm*) y desarmar (*Disarm*) el vehículo. Esto implica que el control remoto tenga o no tenga control sobre el UAV, respectivamente. Es decir, cuando el vehículo se encuentra desarmado, por mucho que se muevan las palancas del mando, el vehículo no reaccionará a sus órdenes. De forma contraria, en el momento en el que se arma, el vehículo queda expuesto totalmente a los movimientos de los canales del mando de radio control.

5.3.6. Receptor FRSKY D4R-II

El receptor FRSKY D4R-II que opera en la frecuencia de 2.4 GHz es utilizado para recibir las señales de control del transmisor de radio. Su pequeño tamaño (40 x 22,5 x 6 mm) y peso (5,8 g) lo hace muy competitivo, como observamos en la Figura 37. Su alcance operativo es de 1.5 Km. Este receptor permite el control manual del dron, lo cual es esencial para el despegue, aterrizaje y maniobras de emergencia.



Figura 37: Receptor FRSKY D4R-II.

5.3.7. Antena 3DR Radio Set

La Antena 3DR Radio Set, como vemos en la Figura 38, es una herramienta para la comunicación inalámbrica para drones. Este sistema de transmisión y recepción de datos opera en la frecuencia de 433 MHz y es ampliamente utilizado en la telemetría de UAVs: La antena 3DR Radio Set se utilizó para establecer una comunicación en tiempo real entre el dron y la estación base, permitiendo la monitorización continua del estado del dron, incluyendo datos críticos como posición GPS, estado de la batería, velocidad y altitud, lo cual es esencial para garantizar la seguridad y éxito de las operaciones de vuelo.



Figura 38: Antena 3DR Radio Set, 433 Mhz.

5.3.8. Antena GPS 3DR uBlox con brújula

El GPS es un sistema que permite determinar la posición del dron en tiempo real. El sistema fue desarrollado por el Departamento de Defensa de los Estados Unidos con fines militares para proporcionar estimaciones precisas de posición, velocidad y tiempo. Utiliza conjuntamente una red de ordenadores y una constelación de 24 satélites para determinar por triangulación, la altitud, longitud y latitud de cualquier objeto en la superficie terrestre. Cuando se desea determinar la posición, el receptor que se utiliza para ello localiza automáticamente como mínimo tres satélites de la red, como vemos en la Figura 39, de los que recibe unas señales indicando la identificación y la hora del reloj de cada uno de ellos. Con base en estas señales, el aparato sincroniza el reloj del GPS y calcula el tiempo que tardan en llegar las señales al equipo, y de tal modo mide la distancia al satélite mediante el método de trilateración inversa, la cual se basa en determinar la distancia de cada satélite respecto al punto de medición. Conocidas las distancias, se determina fácilmente la propia posición relativa respecto a los satélites. Conociendo además las coordenadas o posición de cada uno de ellos por

la señal que emiten, se obtiene la posición absoluta o coordenadas reales del punto de medición.



Figura 39: GPS[29]

En este proyecto se utiliza la Antena GPS 3DR uBlox con brújula es un componente crucial para garantizar la precisión y estabilidad del dron durante las misiones autónomas. Este dispositivo combina un módulo GPS de alta precisión con una brújula electrónica, como vemos en la Figura 40, proporcionando datos esenciales para la navegación y el control del dron.

El módulo GPS uBlox es reconocido por su alta precisión y fiabilidad en la adquisición de señales satelitales, lo que permite obtener una ubicación exacta en tiempo real. La brújula integrada, también conocida como magnetómetro, proporciona datos de orientación esenciales para la estabilización del dron y para realizar maniobras precisas. La antena GPS uBlox puede operar con frecuencias de actualización altas, típicamente de 5 Hz a 10 Hz, asegurando una rápida adquisición de datos. Este dispositivo se conecta a la controladora de vuelo del dron, como la Pixhawk, a través de interfaces estándar como I2C o *UART*. Además, es compacta y ligera, diseñada para

minimizar el impacto en el peso total del dron sin comprometer su capacidad de carga útil.



Figura 40: Antena GPS 3DR uBlox con brújula.

En cuanto a sus funciones y beneficios, el módulo GPS proporciona coordenadas geográficas exactas, permitiendo al dron seguir rutas planificadas con gran precisión. La brújula integrada asegura que el dron mantenga una orientación correcta en todo momento, lo cual es vital para realizar maniobras y mantener la trayectoria deseada. La combinación de datos GPS y de la brújula ayuda a la controladora de vuelo a mantener la estabilidad del dron, incluso en condiciones de viento o turbulencia. La alta frecuencia de actualización y la precisión del GPS uBlox minimizan los errores de posicionamiento, mejorando la eficiencia de las misiones autónomas.

5.3.9. GoPro Hero 3

La cámara GoPro montada en el dron es utilizada para capturar imágenes y videos durante el vuelo. Esta cámara es esencial para aplicaciones de mapeo, vigilancia y reconocimiento, proporcionando una visualización en tiempo real de la operación del dron, como vemos en la Figura 41.



Figura 41: Cámara GoPro Hero 3.

5.3.10. Gimbal

El gimbal es un componente crucial para la estabilización de cámaras en aplicaciones de drones, especialmente en operaciones que requieren una captura de imágenes y vídeos de alta calidad y precisión. En este proyecto, el gimbal está configurado para ser controlado mediante el radio control del dron, como vemos en la Figura 42, permitiendo al operador mover la cámara a la posición deseada en tiempo real durante el vuelo.



Figura 42: Gimbal.

5.3.11. Batería Externa

La batería externa es utilizada para alimentar componentes adicionales del dron que no pueden ser soportados por la batería principal. Esto incluye dispositivos como la Raspberry Pi y otros sensores que requieren una fuente de energía constante y confiable.

En este proyecto, se utilizan dos baterías externas: una en el dron y otra en la estación base. La batería externa del dron está conectada a la Raspberry Pi y a su vez al módulo LoRa. Esta configuración asegura que la Raspberry Pi, que ejecuta algoritmos avanzados de control y procesamiento de datos, pueda funcionar sin interrupciones. De manera similar, la estación base también está equipada con una batería externa que alimenta su propia Raspberry Pi y demás componentes, garantizando que toda la estación funcione de manera autónoma y eficiente en el campo, sin depender de una fuente de alimentación fija.

5.3.12. Batería Lipo

Es el elemento que aporta la energía al sistema para poder funcionar. En el mercado hay infinidad de tipos de batería atendiendo a especificaciones de voltaje y corriente, qué tipo de material ha sido utilizado para su elaboración, cantidad de energía que se demanda en un instante, capacidad. La que se ha usado en este proyecto es una Batería TATTU 12500 mAh 4s 25c (14.8 voltios), tiene la capacidad de 12500 mAh, su tasa de descarga es de 25C, como vemos en la Figura 43. Son unos valores normales, dentro de la aplicación que le vamos a dar al dron dentro de un rango económico.



Figura 43: Bateria Lipo, TATUU.

Todas las baterías están compuestas por celdas individuales que se conectan en serie. Así, el número de celdas se indicará acompañando el dígito de la letra S. 1S indica una celda, 2S indica dos celdas, y así sucesivamente. Una batería de 4 celdas tendrá $4 \times 3.7 \text{ (V)} = 14,8 \text{ (V)}$ nominales. Cuando se carga, tiene $4 \times 4,2 \text{ (V)} = 16,8 \text{ (V)}$ y cuando se descarga, $4 \times 3,3 \text{ (V)} = 13,2 \text{ (V)}$, como es la que hemos usado. El voltaje afecta de manera directa a la velocidad que podrán alcanzar los motores, así como al peso de la batería. Las baterías con mayor número de celdas pesan más. Por otro lado, la capacidad de la batería, en mAh, determina la cantidad de energía total que será capaz de suministrar la batería, y, por tanto, está directamente relacionada con la duración máxima del vuelo que podrá realizarse. No obstante, aumentar la capacidad de la batería también tiende a aumentar su tamaño y peso, lo que puede afectar negativamente a la eficiencia del vuelo del dron. Por último, la tasa máxima de descarga de la batería, C, determina la corriente máxima que podrá suministrar la batería sin dañarse. Dicho máximo se calcula multiplicando C por la capacidad de la batería. Cuanto mayor sea el consumo de los motores mayor serán los requerimientos de capacidad y de tasa de descarga máxima de la batería. Por regla general, se aconseja emplear baterías con una tasa máxima de descarga considerablemente superior a la estrictamente requerida por los motores para alargar la vida útil de las mismas. No obstante, las baterías con valores C mayores pesan más y suelen ser más caras.

Este tipo de baterías que se encuentran divididas en celdas necesitan de cargadores especiales que realizan cargas equilibradas entre las celdas. Esto significa que no cargan una celda y luego otra ya que puede acarrear una disminución del rendimiento de la batería y de su vida útil.

5.3.13. Motores y Hélices

El dron está equipado con 8 motores y hélices (octocóptero), lo que le proporciona una mayor estabilidad y capacidad de carga. Los motores son responsables de generar la fuerza de empuje necesaria para el vuelo, mientras que las hélices transforman la energía de los motores en movimiento ascendente y control direccional.



a) Motor SunnySky



b) Hélices

Figura 44: Motor y Hélices.

Los motores que se pusieron en el dron son motores asíncronos sin escobillas (*brushless*), como vemos en la Figura 44. Estos motores presentan ventajas importantes frente a otros tipos de motores ya que son más baratos de fabricar, pesan menos y requieren menos mantenimiento. Sin embargo, su control es más complejo y requiere el empleo de ESC. Los motores del dron son el componente que permite el movimiento de las hélices. Comprenden un aspecto fundamental pues por su elección pasa el consumo energético de manera directa.

Capítulo 6

6. Pruebas de campo

6.1. Introducción

Tras la integración del prototipo en el dron experimental, se realizaron diversas pruebas de campo con el objetivo de validar el correcto funcionamiento de los sistemas integrados y evaluar el desempeño del dron en diferentes escenarios de vuelo. Estas pruebas fueron esenciales para asegurar que el sistema desarrollado pudiera operar de manera eficiente y segura. A continuación, se detallan los procedimientos y resultados de cada una de las pruebas realizadas.

Una de las partes importantes de las pruebas de campo fue el proceso de envío y recepción de *waypoints*. Estos *waypoints* son coordenadas geográficas que el dron debe seguir durante su vuelo autónomo. El proceso comienza con la selección de un archivo de texto que contiene las coordenadas de los *waypoints*. Este archivo se carga en el sistema utilizando una interfaz desarrollada para permitir la selección y carga de archivos de manera eficiente.

El archivo de texto contiene una lista de coordenadas en formato de latitud, longitud y altitud. Estas coordenadas son esenciales para definir la ruta que el dron seguirá durante su misión. El sistema de comunicación LoRa se utiliza para transmitir estas coordenadas desde la estación base hasta el dron. El código desarrollado para esta

tarea lee cada línea del archivo de texto, separa las coordenadas y las envía al dron a través del módulo LoRa.

El dron, equipado con su propio módulo LoRa, recibe estas coordenadas y las almacena para su uso durante el vuelo. Una vez que el dron ha recibido todas las coordenadas, se envía un mensaje de confirmación a la estación base indicando que está listo para iniciar la misión. Este proceso de envío y recepción es crucial para asegurar que el dron pueda seguir una ruta predefinida con precisión y sin errores.

6.2. Primera Prueba de Campo

La primera prueba de campo se centró en verificar el funcionamiento básico y manual del dron. Para ello, se realizaron varias calibraciones esenciales, incluyendo la configuración inicial del radio control y la calibración de los sensores del dron. Estas actividades se llevaron a cabo con el objetivo de asegurar que todos los sistemas estuvieran operativos y en condiciones óptimas para vuelos controlados manualmente.

Antes de realizar la misión, se conectó la Pixhawk del dron con el ordenador mediante un transceptor de 433 MHz. La conexión se estableció utilizando el puerto COM y configurando una velocidad de transmisión de 115200 baudios. Esta conexión permitió la comunicación entre la estación base y el dron.

Para la misión, se eligió una ruta sencilla en forma de cuadrado, compuesta por cuatro *waypoints*. La velocidad entre *waypoints* se configuró a 250 centímetros por segundo. Los *waypoints* se cargaron en el dron a través del transceptor. Cargar los *waypoints* implica que el transceptor en la estación base envía los datos al transceptor del dron, que a su vez los transfiere a la Pixhawk para su almacenamiento y ejecución.

En Mission Planner, el proceso de selección y carga de *waypoints* se realiza a través de una interfaz gráfica que permite al usuario definir los puntos de la misión. Los *waypoints* se pueden seleccionar directamente en el mapa de la interfaz, utilizando el botón "*Waypoint*" para agregar nuevos puntos. Una vez seleccionados, se almacenan con el botón "*Write WPs*" y se cargan en el dron utilizando el botón "*Send WPs*".

Los *waypoints* seleccionados para esta misión fueron los siguientes:

- **Waypoint 1:** 28.1263253, -15.4792750, 20.0
- **Waypoint 2:** 28.1262969, -15.4790202, 30.0
- **Waypoint 3:** 28.1265784, -15.4789504, 30.0
- **Waypoint 4:** 28.1266175, -15.4792576, 30.0

La misión cargada fue una ruta en forma de cuadrado, diseñada para evaluar la precisión y estabilidad del dron durante un vuelo autónomo simple. En la Figura 45, se observa cómo el dron ejecutó la misión de manera efectiva, siguiendo los *waypoints* establecidos sin desviaciones significativas. Esta prueba fue fundamental para confirmar la correcta integración entre el *software* y el *hardware* del dron, y para identificar posibles ajustes necesarios antes de avanzar a misiones más complejas.

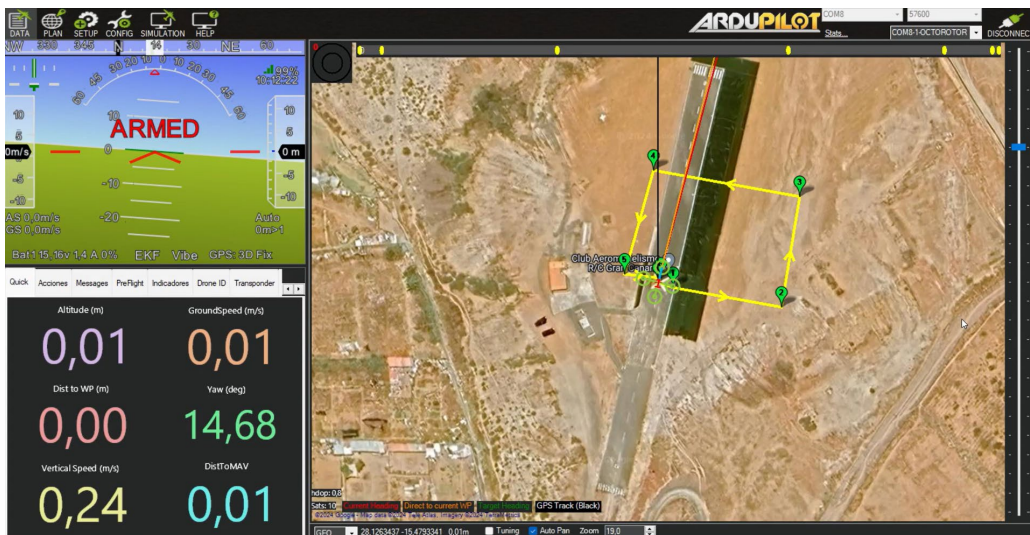


Figura 45: Primera prueba de campo, Mission Planner.

6.3. Segunda Prueba de Campo

La segunda prueba de campo se realizó con un enfoque más avanzado, utilizando el software QGroundControl y la implementación del código de comunicación LoRa para el transmisor y receptor. Esta prueba tuvo como objetivo validar la integración de estos componentes y la precisión en la ejecución de misiones autónomas.

En esta prueba, nos desligamos del uso de Mission Planner y utilizamos nuestra estación base, que consiste en una Raspberry Pi equipada con una interfaz gráfica para la selección y envío de waypoints, como se describe en el Capítulo 4, Figura 24. Para enviar los *waypoints*, se hizo uso de la interfaz mostrada en dicha figura. La misión consistió en un rombo, para lo cual se necesitaron cargar cinco *waypoints*.

Los *waypoints* se almacenaron en un archivo txt, en este caso "*aero_club_mission.txt*". Este archivo contiene las coordenadas de los waypoints, que se pasan al ESP32 a través de la Raspberry Pi. El ESP32, a su vez, los transmite mediante el módulo LoRa. El módulo LoRa del dron recibe estos datos, los pasa al ESP32 del dron, y este se encarga de transferirlos a la Pixhawk para la carga de la misión.

Los waypoints seleccionados para esta misión fueron los siguientes:

- **Waypoint 1:** 28.12685750, -15.47920660, 30.0
- **Waypoint 2:** 28.12645190, -15.47902550, 40.0
- **Waypoint 3:** 28.12616560, -15.47937960, 30.0
- **Waypoint 4:** 28.12655830, -15.47953250, 40.0
- **Waypoint 5:** 28.12654730, -15.47952469, 40.0

La misión cargada fue una ruta en forma de rombo, como observamos en la Figura 46. Durante la ejecución de la misión, se evaluaron varios aspectos clave:

- **Establecimiento de Conexión LoRa:** El código se encargó de establecer la conexión entre el módulo LoRa del dron y el de la estación base.
- **Confirmación de Inicio de Misión:** El dron esperó y procesó la confirmación de inicio de misión. Este proceso fue crucial para asegurar que el dron solo comenzará la misión una vez que se recibiera la confirmación desde la estación base.
- **Armado del Dron:** Se verificó el proceso mediante el cual el dron recibía y procesaba el mensaje de armado correctamente. Este paso involucró cambiar el modo del dron a "*GUIDED*", utilizando funciones de la librería *dronekit* para armar el dron.
- **Comunicación Constante:** Se evaluó la capacidad del sistema para mantener una comunicación constante y efectiva entre el dron y la estación base. Esto se logró utilizando el protocolo LoRa para transmitir datos en tiempo real.

En el proceso de armado y vuelo autónomo, se utilizaron comandos de la librería *droneKit* y *Pymavlink*. Estos comandos incluyen:

1. ***drone.mode = VehicleMode("GUIDED")***: este comando cambia el modo de vuelo del dron a "GUIDED", permitiendo el control autónomo mediante comandos enviados desde la estación base.
2. ***drone.armed = True***: este comando arma el dron, preparándolo para el despegue.
3. ***drone.simple_takeoff(altura)***: este comando ordena al dron despegar y ascender hasta la altura especificada.
4. ***drone.simple_goto(waypoint)***: este comando dirige al dron hacia el waypoint especificado.
5. ***drone.mode = VehicleMode("LAND")***: este comando cambia el modo de vuelo del dron a "LAND", ordenando al dron aterrizar en la ubicación actual.

En este caso el QGroundControl no cargaba la misión del vuelo, si no que se utilizó para para para ver la telemetría a tiempo real y visualizar la ruta durante el vuelo, como observamos en la Figura 46. Los resultados fueron positivos, el sistema LoRa funcionó, manteniendo una comunicación estable y efectiva durante toda la misión. El dron completó la misión con alta precisión, validando así la efectividad del sistema LoRa para la transmisión de datos.

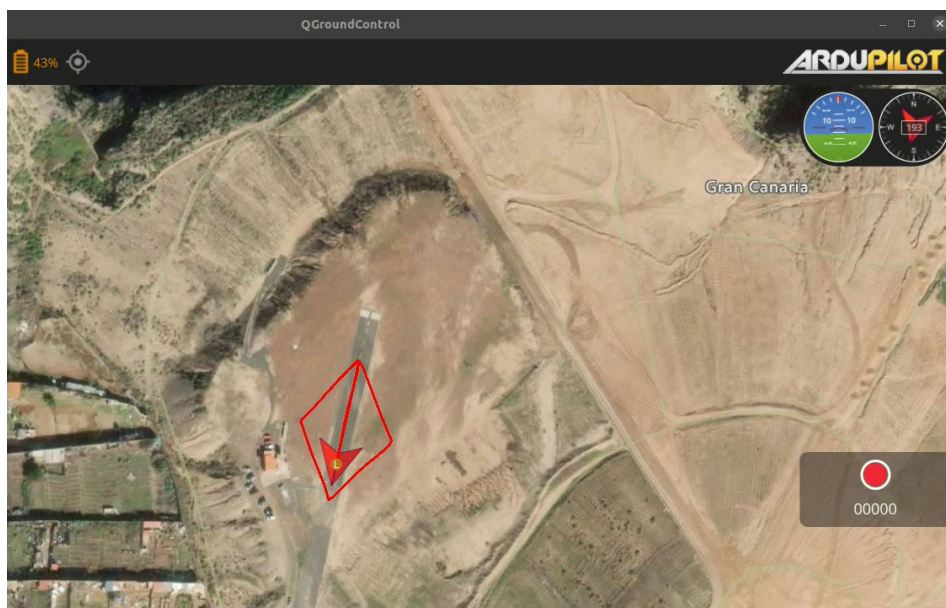


Figura 46: Segunda prueba de campo, QGroundControl.

6.4. Tercera Prueba de Campo

La tercera prueba de campo se enfocó en evaluar la capacidad de transmisión del módulo LoRa a diferentes distancias, además de implementar *waypoints* avanzados y la función ROI. Inicialmente, se intentó realizar la prueba desde el Mirador de la Montaña de Arucas hasta el Club Aeromodelismo R/C Gran Canaria, a una distancia de 4.3 km, como se muestra en la Figura 47. Aunque algunos mensajes llegaban, la conexión no era estable y se perdían datos cruciales. Esto sugirió que, a esta distancia y con las condiciones presentes, el sistema LoRa no proporcionaba la fiabilidad suficiente para las operaciones del dron.



Figura 47: Tercera prueba de campo a 4,3 km.

Se eligió esta localización debido a la visibilidad directa entre los puntos de prueba. Sin embargo, la falta de buenos resultados podría deberse a las antenas utilizadas o a las interferencias en la antena ubicada en el mirador de la montaña de Arucas. Ante estos resultados, se decidió cambiar la localización para mejorar la evaluación del sistema.

Posteriormente, se redujo la distancia de prueba a 500 metros, como se muestra en la Figura 48. En esta configuración, la conexión resultó ser significativamente más estable, con todos los *waypoints* llegando correctamente y sin pérdida de datos. Esto

permitió que el dron se armara y siguiera la ruta planificada de manera efectiva, manteniendo la comunicación continua con la estación base.



Figura 48: Tercera prueba de campo a 500 metros.

Para esta prueba, se diseñaron otro tipo de *waypoints* llamados *waypoints spline*, lo que permitió que el dron siguiera una ruta más suave, continua, y que el dron se dirigiera con más redondez a esos *waypoints* con una velocidad entre *waypoints* 125 centímetros por segundo, mejorando la eficiencia del vuelo y reduciendo el consumo de energía, como observamos en la Figura 49. Adicionalmente, se configuró el ROI para que la cámara del dron permaneciera apuntando a un punto específico durante todo el vuelo. Los resultados mostraron que el dron pudo mantener el enfoque en la posición designada mientras seguía la ruta, demostrando la integración de las funcionalidades avanzadas del software de control y la capacidad del hardware para soportar operaciones de mayor complejidad.



Figura 49: Ruta circular con los *waypoints spline*.

En la misión circular, se siguió un proceso similar al de misiones anteriores. Utilizamos el archivo "aero_club_circular.txt" y aseguramos que la distancia y todos los demás parámetros fueran correctos. Los waypoints seleccionados para esta misión fueron los siguientes:

- **Waypoint 1:** 28.1269439, -15.4797390, 40.0
- **Waypoint 2:** 28.1276133, -15.4790416, 40.0
- **Waypoint 3:** 28.1276062, -15.4786742, 40.0
- **Waypoint 4:** 28.1271118, -15.4784596, 40.0
- **Waypoint 5:** 28.1265985, -15.4786795, 40.0
- **Waypoint 6:** 28.1260284, -15.4792348, 40.0
- **Waypoint 7:** 28.1261727, -15.4797497, 40.0

Para el armado y vuelo autónomo, utilizamos el modo AUTO. En este modo se emplearon comandos de MAVLink, como el comando MAV_CMD_NAV_TAKEOFF para el despegue, el comando MAV_CMD_NAV_LAND para el aterrizaje, y los comandos MAV_CMD_DO_SET_ROI y MAV_CMD_NAV_SPLINE_WAYPOINT para establecer la región de interés y los waypoints spline, respectivamente. El uso de estos comandos asegura que el dron puede ejecutar la misión de manera completamente autónoma.

Es importante destacar que mientras que el transceptor de 433 MHz permite realizar misiones a distancias menores, el sistema LoRa proporciona un alcance significativamente mayor, permitiendo la transmisión de waypoints a distancias de varios kilómetros. Esta capacidad de largo alcance ofrece una ventaja crucial para misiones que requieren que el dron opere a grandes distancias de la estación base, aumentando la flexibilidad y el potencial operativo del sistema.

Capítulo 7

7. Conclusiones

7.1. Resultados y revisión de objetivos

Tras realizar las pruebas de verificación del prototipo, se puede concluir que los elementos que conforman el sistema son viables y cumplen con los objetivos planteados en este TFG. El objetivo principal era diseñar e implementar un sistema de comunicación autónomo y flexible entre una estación base y un dron, utilizando la tecnología LoRa y protocolos como *MAVLink* o *MAVProxy*. A continuación, se detallan los logros y desafíos enfrentados durante este trabajo.

En segundo lugar, se pretendía gestionar de manera alternativa el control del dron a través de una plataforma externa. Este enfoque buscaba diversificar las opciones de control, brindando flexibilidad y redundancia en la unidad de pilotaje de la aeronave no tripulada. A continuación, se detallan los logros y desafíos enfrentados durante este trabajo.

Uno de los primeros pasos fue la calibración del dron y la configuración de parámetros críticos, incluyendo la IMU. Estos pasos fueron esenciales para asegurar la estabilidad y precisión del dron durante el vuelo. Se utilizaron herramientas como Mission Planner y QGroundControl para establecer rutas y configurar misiones, asegurando que el dron pudiera ejecutar vuelos autónomos de manera fiable.

El uso de entornos de simulación, particularmente SITL, fue fundamental para probar y validar las soluciones desarrolladas antes de llevar a cabo vuelos reales. La

simulación permitió identificar y corregir errores de manera segura, optimizando el proceso de desarrollo y asegurando la funcionalidad del sistema en condiciones controladas.

La implementación de la tecnología LoRa fue uno de los aspectos más innovadores del proyecto. Se programaron los *chips* ESP32 utilizando el entorno Arduino IDE, y se desarrollaron *scripts* en *Python* para gestionar la comunicación en los módulos LoRa. Esto permitió el envío y recepción de *waypoints* y otros comandos, lo cual es crucial para misiones en entornos remotos. Además, se llevaron a cabo pruebas para evaluar el rendimiento de LoRa a diferentes distancias, confirmando su efectividad y adaptabilidad.

Este proyecto no solo demuestra el potencial de la tecnología LoRa para mejorar la autonomía y eficiencia de los UAV, sino que también resalta la importancia de integrar diferentes tecnologías y metodologías para desarrollar soluciones en el campo de la robótica y las telecomunicaciones.

7.2. Líneas futuras

Durante el desarrollo del proyecto y tras la revisión de los resultados, se han identificado varias áreas de mejora y posibles líneas de trabajo futuras que podrían expandir y optimizar el sistema:

- **Mejora de la Comunicación LoRa:** Es necesario optimizar el código para la decodificación de paquetes y bytes, implementando mecanismos de confirmación (*ACK*) y explorando distintas formas de transmisión de datos. Además, se podría investigar el uso de repetidores LoRa para extender el alcance efectivo de la comunicación.
- **Optimización del Consumo de Energía:** La prueba circular reveló que la velocidad de los *waypoints* a 125 cm/s resultaba ser muy lenta y por tanto el consumo de energía elevado, impidiendo completar la misión debido a la baja batería. Se sugiere explorar configuraciones de vuelo más eficientes y baterías

- **Cámaras térmicas para incendios forestales y detección de objetos:** Integrar cámaras térmicas en los drones para la detección y seguimiento de incendios forestales, así como para la identificación de objetos en diversas condiciones ambientales. Esto no solo mejoraría la capacidad del dron para operar en situaciones de emergencia, sino que también expandiría sus aplicaciones en el monitoreo ambiental y la seguridad.
- **Algoritmos colaborativos:** Desarrollar algoritmos colaborativos entre los drones que faciliten la coordinación y optimización de recursos, mejorando la eficiencia en misiones conjuntas.

En conclusión, el trabajo realizado ha sentado las bases para futuras investigaciones y desarrollos en el campo de los drones autónomos, con un enfoque especial en la comunicación y el control eficiente mediante tecnologías como LoRa, *MAVLink* y *droneKit*. Las mejoras y líneas de trabajo propuestas apuntan a un sistema más robusto, flexible y eficiente, capaz de abordar los desafíos futuros en este campo en constante evolución.

Bibliografía

- [1] A. Madridano, S. Campos, A. Al-Ka, F. Garcia, D. Martin, y A. Escalera, «Unmanned aerial vehicle for fire surveillance and monitoring», *RIAI - Revista Iberoamericana de Automatica e Informatica Industrial*, vol. 17, n.º 3, pp. 254-263, 2020, doi: 10.4995/RIAI.2020.11806.
- [2] Agencia de gobierno, «“Avances informativos,” Ministerio Para La Transición Ecológica Y El Reto Demográfico.», El Ministerio para la Transición Ecológica y el Reto Demográfico de España. Accedido: 26 de febrero de 2024. [En línea]. Disponible en: <https://www.miteco.gob.es/es/biodiversidad/temas/incendios-forestales/estadisticas-avances.html>
- [3] Greenpeace, «Incendios forestales | Greenpeace España - ES | Greenpeace España», Greenpeace. Accedido: 29 de febrero de 2024. [En línea]. Disponible en: <https://es.greenpeace.org/es/trabajamos-en/bosques/incendios-forestales/>
- [4] Asociación Nacional de Protección contra el Fuego, «Ingenieros de protección contra incendios». Accedido: 13 de julio de 2024. [En línea]. Disponible en: <https://www.nfpa.org/es/for-professionals/career-growth/fire-protection-engineer>
- [5] La Dirección General del Instituto Geográfico Nacional (IGN) y el organismo autónomo Centro Nacional de Información Geográfica (CNIG), «Teledetección». [En línea]. Disponible en: <http://www.ign.es>
- [6] Desarrollo Tecnológico y la Innovación en Comunicaciones IDeTIC, «SeLiF: Detección y seguimiento de líneas de fuego en incendios forestales basado en sensores térmicos móviles de despliegue rápido.» Accedido: 13 de julio de 2024. [En línea]. Disponible en: <https://idetec.ulpgc.es/idetic/index.php/es/proyecto-selif>
- [7] ««Red integral de prevención y Gestión de incendios Forestales mediante Georreferenciación en Observadores móviles (Red Gesfogo)” Unión europea (COOPERACIÓN)», IP Coord. V. Araña. (ULPGC, MITI, Cabildo GC, Gob. Canarias, Gob. Madeira, Bomberos Cabo Verde) “Red integral de prevención y Gestión de

- incendios». Accedido: 13 de junio de 2024. [En línea]. Disponible en: <https://www.gesfogo.ulpgc.es/index.php/es/>
- [8] «Estas son las limitaciones que tiene el WiFi, ¿cuándo se arreglará?» Accedido: 13 de junio de 2024. [En línea]. Disponible en: <https://www.redeszone.net/noticias/hogar/poner-enchufe-inteligente-frigorifico-utilidad/>
- [9] «Drones Industriales - ATRIA Innovation». Accedido: 10 de julio de 2024. [En línea]. Disponible en: <https://atriainnovation.com/blog/drones-industriales/>
- [10] «En 2024 entra en vigor el mercado de clase de drones/UAS y la identificación a distancia directa | AESA-Agencia Estatal de Seguridad Aérea - Ministerio de Fomento». Accedido: 13 de junio de 2024. [En línea]. Disponible en: <https://www.seguridadaerea.gob.es/es/noticias/en-2024-entra-en-vigor-el-marcado-de-clase-de-dronesuas-y-la-identificacion-distancia>
- [11] «Lesson 1: Leaf Cycle – Robolink Basecamp». Accedido: 11 de julio de 2024. [En línea]. Disponible en: <https://learn.robolink.com/lesson/lesson-1-leaf-cycle/>
- [12] «A Brief History of LoRa: Three Inventors Share Their Story». Accedido: 13 de junio de 2024. [En línea]. Disponible en: <https://blog.semtech.com/a-brief-history-of-lora-three-inventors-share-their-personal-story-at-the-things-conference>
- [13] E. R. Ramos Silvestre, R. I. Fuentes Telleria, y A. Otalora Rodriguez, «Desarrollo de una plataforma de monitoreo para una red de dispositivos IOT empleando tecnología Lorawan», *Journal Boliviano de Ciencias*, vol. 18, n.º 53, pp. 6-25, dic. 2022, doi: 10.52428/20758944.V18I53.371.
- [14] «Spreading Factors | The Things Network». Accedido: 10 de julio de 2024. [En línea]. Disponible en: <https://www.thethingsnetwork.org/docs/lorawan/spreading-factors/>
- [15] «LoRa-Wan – Orus Eficiencia». Accedido: 15 de mayo de 2024. [En línea]. Disponible en: <https://oruseficiencia.es/lora-wan>

- [16] «LoRaWAN - Parte 3 - Arquitectura y tramas - El bosque de silicio». Accedido: 5 de julio de 2024. [En línea]. Disponible en: <https://elbosquedesilicio.es/lorawan3/>
- [17] «Goto IoT | Intro a la tecnología LoRa». Accedido: 10 de julio de 2024. [En línea]. Disponible en: https://www.gotoiot.com/pages/articles/lora_intro/index.html
- [18] «LoRaWAN versus NB-IoT: ¿Cómo se comparan y difieren?» Accedido: 10 de junio de 2024. [En línea]. Disponible en: <https://www.mokosmart.com/es/lorawan-vs-nb-iot-how-do-they-compare-and-differ/>
- [19] P. T. F. F. and M. P. L. Meier, «Micro Air Vehicle Link (MAVLink) in a Nutshell: A Survey», *IEEE Access*, June 2019..
- [20] MAVLink, «Introduction · MAVLink Developer Guide».
- [21] «MAVLink · Erle Robotics Gitbook». Accedido: 10 de julio de 2024. [En línea]. Disponible en: <https://erlerobotics.gitbooks.io/erlerobot/content/es/mavlink/mavlink.html>
- [22] «Pymavlink (Python-mavgen) · MAVLink Developer Guide». Accedido: 13 de junio de 2024. [En línea]. Disponible en: https://mavlink.io/en/mavgen_python/
- [23] «DroneKit». Accedido: 13 de junio de 2024. [En línea]. Disponible en: <https://dronekit.io/>
- [24] «Welcome to DroneKit-Python's documentation!» Accedido: 13 de junio de 2024. [En línea]. Disponible en: <https://dronekit-python.readthedocs.io/en/latest/>
- [25] «LoRa, ESP32, TTGO, TTGO-LoRa32, LoRaWan». Accedido: 15 de julio de 2024. [En línea]. Disponible en: <https://altronics.cl/ttgo-lora32-t3-esp32-915mhz>
- [26] «Compass Calibration — Copter documentation». Accedido: 10 de junio de 2024. [En línea]. Disponible en: <https://ardupilot.org/copter/docs/common-compass-calibration-in-mission-planner.html>
- [27] «Homepage - Pixhawk». Accedido: 15 de julio de 2024. [En línea]. Disponible en: <https://pixhawk.org/>

- [28] «Processors - Raspberry Pi Documentation». Accedido: 15 de julio de 2024. [En línea]. Disponible en: <https://www.raspberrypi.com/documentation/computers/processors.html>
- [29] «Tecnología Multimedia: GPS». Accedido: 15 de julio de 2024. [En línea]. Disponible en: <https://tecmultimed.blogspot.com/2016/08/gps.html>

Presupuesto

P.1. Introducción

En este capítulo se desarrolla el presupuesto de este TFG, que permite valorar los costes que han supuesto los materiales y herramientas utilizados durante su elaboración, además del coste del trabajo realizado. Para ello, los puntos tratados en este capítulo en relación con el cálculo del coste total que supone este trabajo son los siguientes:

- Costes de los materiales.
- Amortización del inmovilizado material.
- Trabajo tarifado por tiempo empleado.
- Aplicación de impuestos y coste total.

P.2. Costes materiales

Los costes de materiales de un proyecto se pueden diferenciar entre fungibles, referidos a todos los componentes adquiridos que están expuestos a un deterioro con el uso durante la elaboración de este, y no fungibles, que se asume que no están expuestos a deterioro durante el periodo de ejecución del proyecto.

En este TFG, al no usar materiales no fungibles implica que únicamente tenemos en cuenta los costes de los materiales fungibles, que en nuestro caso serían:

- **Materiales hardware**, que hacen referencia a componentes *hardware* que son utilizados para la elaboración de este trabajo, sin incluir los equipos de medida que, al ser amortizables, no se incluyen en esta categoría.

- **Materiales software**, que hacen referencia a las licencias de los programas que se emplean en este trabajo, asumiendo un tiempo de uso de un año, motivo por el que no se incluye en la categoría de elementos amortizables.

Además de esta clasificación, se añaden a los costes materiales la categoría de otros conceptos de gastos, donde se evalúan aquellos elementos adquiridos que no pueden englobarse formalmente en ninguna de las dos categorías anteriores, aunque se imputan igualmente en los costes materiales finales del Trabajo Fin de Grado.

P.2.1. Materiales hardware

En el coste de materiales hardware se incluye todo el material que se ha utilizado para el ensamblaje del dron.

Concepto	Precio Unitario (€)	Precio unidad (€)	Total (€)
Portatil	1	1200	1.200,00 €
Dron bloque	1	2200	2.200,00 €
Raspberry Pi 4 Modelo B	2	75	150,00 €
Lora LILYGO	2	43	86,00 €
Cámara GoPro hero 3+	1	250	250,00 €
TOTAL:			3.886,00 €

Tabla P.1: Costes de materiales *hardware*.

P.2.2. Materiales Software

El coste por uso de equipos y software se compone de:

- **Software de libre distribución:** Mission Planner, QGroundControl.
- **Sistema operativo:** Windows 10, Ubuntu.

Concepto	Precio Unitario (€)	Precio unidad (€)	Total (€)
Ubuntu	1	0	0,00 €
Windows 10	1	0	0,00 €
Mission Planner	1	0	0,00 €
QGroundControl	1	0	0,00 €
Arduino iddle	1	0	0,00 €
Visual Studio Code	1	0	0,00 €
TOTAL:			0,00 €

Tabla P.2: Presupuesto de Costes de Equipos y Software.

P.3. Amortización del inmovilizado material

El inmovilizado material son aquellos elementos que son utilizados durante la elaboración de este TFG, pero pueden ser reutilizados para otros proyectos más adelante. De esta manera, los elementos que se engloban en esta categoría son utilizados durante el periodo de tiempo estipulado para el desarrollo del trabajo, que se conoce como **tiempo de vida del producto**, y con el tiempo reducen su valor adquisitivo inicial una cierta cantidad, que es lo que se conoce como **valor residual**.

De este modo, se estipula el coste de amortización para un período de 3 años, utilizando un sistema de amortización lineal o constante. En este sistema, se supone que el inmovilizado material se deprecia de forma constante a lo largo de su vida útil. La cuota de amortización anual se calcula haciendo uso de la siguiente fórmula (P.1):

$$C = \frac{(V_{ad} - V)}{N}$$

(P.1)

Donde:

C: cuota de amortización anual

V_{ad} : valor de la adquisición

V: valor residual.

N: número de años de vida útil de la adquisición.

Siendo el valor residual el valor teórico que se supone tendrá el elemento en cuestión después de su vida útil, teniendo en cuenta los índices de depreciación actual, como vemos en la fórmula (P.2). En el caso del hardware y del software son 3 años (al 10 % de depreciación máximo por año).

$$\text{Valor residual} = \frac{\text{Valor inicial}}{\text{Vida útil en años}}$$

(P.2)

P.4. Amortización del material hardware

Debido a que el trabajo se ha elaborado en un periodo inferior a 3 años, que es el periodo en que se calcula la amortización del material hardware, se realizará una amortización equiparable al período de duración de este. Según esto, se obtienen los gastos expuestos en la tabla (P.3).

Equipos	Unidades	Valor de adquisición	Valor residual	Vida útil (años)	Tiempo de uso (meses)	Total
Portátil	1	1.200	120	5	3	54,00 €
Dron bloque	1	2.200	220	3	3	165,00 €
Raspberry Pi 4 Modelo B	2	75	7,5	3	3	5,63 €
Lora LILYGO	2	43	4,3	3	3	3,23 €
Cámara GoPro hero 3+	1	250	25	3	3	18,75 €
TOTAL:						246,60 €

Tabla P.3: Precios y costes de amortización hardware.

P.5. Trabajo tarifado por tiempo empleado

Para calcular el salario que debe percibir el autor de este TFG, se toma como referencia la Tabla de Contrataciones de Investigadores con cargo a Proyectos de Investigación de la Universidad de Las Palmas de Gran Canaria, como observamos en la figura (P.1), publicada en el BOULPGC el 3 febrero 2023.

MODIFICACIÓN DE LAS RETRIBUCIONES DEL PERSONAL INVESTIGADOR, DEL PERSONAL TÉCNICO Y DEL PERSONAL TÉCNICO DE APOYO CONTRATADO DE LA ULPGC CON CARGO A PROYECTOS, CONVENIOS Y CONTRATOS PARA EL AÑO 2023

TIPO DE PERSONAL	TITULACIÓN MÍNIMA EXIGIDA	CATEGORÍA	DEDICACIÓN (horas semanales)	RETRIBUCIÓN MENSUAL	SEGURIDAD SOCIAL MENSUAL	SEGURIDAD SOCIAL ANUAL	RETRIBUCIÓN BRUTA ANUAL	COSTE TOTAL ANUAL (H+I+L)	COSTE MENSUAL	
PERSONAL TÉCNICO Y DE APOYO	PERSONAL DE APOYO	Técnico superior FP o equivalente (MECES 1)	PACP3 (MECES 1)	TC 37,5	1.077,18	447,25	5.366,97	12.926,16	18.293,13	1.524,43
				TP 20	574,49	334,10	4.009,18	6.893,88	10.903,06	908,59
		Grado o equivalente (MECES 2)	PACP2 (MECES 2)	TC 37,5	1.135,32	539,30	6.471,58	13.623,84	20.095,42	1.674,62
				TP 20	605,51	403,05	4.836,62	7.266,12	12.102,74	1.008,56
		Master o equivalente (MECES 3)	PACP1 (MECES 3)	TC 37,5	1.849,85	612,81	7.353,68	22.198,20	29.551,88	2.462,66
				TP 20	986,58	403,05	4.836,62	11.838,96	16.675,58	1.389,63
	TÉCNICO	Técnico superior FP o equivalente (MECES 1)	TCP5 (MECES 1)	TC 37,5	1.077,18	447,25	5.366,97	12.926,16	18.293,13	1.524,43
				TP 20	574,49	334,10	4.009,18	6.893,88	10.903,06	908,59
		Grado o equivalente (MECES 2)	TCP4 (MECES 2)	TC 37,5	1.397,18	539,30	6.471,58	16.766,16	23.237,74	1.936,48
				TP 20	745,15	403,05	4.836,62	8.941,80	13.778,42	1.148,20
		Master o equivalente (MECES 3)	TCP3 (MECES 3)	TC 37,5	1.601,19	539,30	6.471,58	19.214,28	25.685,86	2.140,49
				TP 20	853,96	403,05	4.836,62	10.247,52	15.084,14	1.257,01

Figura P.1: Tabla de Contrataciones de Investigadores con cargo a Proyectos de Investigación de la Universidad de Las Palmas de Gran Canaria.

Se asume que el autor se enmarca en la clasificación de investigador en proyecto a tiempo parcial, con una titulación de Grado, de modo que su salario mensual es de 745,15 €. Dado que el TFG tiene una duración límite de 3 meses. La tabla (P.4) muestra el sueldo total que debe ganar el autor de este trabajo en función del número de horas dedicadas y el sueldo establecido por la institución mencionada.

En el coste de personal, se incluyen los sueldos base por hora trabajada de la persona que trabajó en este proyecto. En este caso se ha trabajado 25 horas semanales a 4 semanas cada mes en 3 meses (300 horas en total).

Concepto	Cantidad (Ud)	Precio por hora	Total
TÉCNICO GRADO	300	7,45	2235,45

Tabla P.4: Coste de personal.

Por tanto, el salario que percibe el autor de este TFG suma un total de dos mil doscientas treinta y cinco con cuarenta y cinco euros.

P.6. Aplicación de impuestos y coste total

Dado que este TFG se realiza en una institución integrada en la Comunidad Autónoma de Canarias, sobre los costes totales se debe aplicar el Impuesto General Indirecto Canario (I.G.I.C.), que se establece en un 7% sobre los costes asociados, tal y como se indica en la tabla (P.4).

Concepto	Coste
Costes de materiales	3.886,00 €
Amortización del inmovilizado material	246,60 €
Trabajo tarifado por tiempo empleado	2.235,45 €
COSTE TOTALES (sin I.G.I.C.):	6.368,05 €
COSTE TOTALES (CON I.G.I.C.):	6.813,81 €

Tabla P.4: Costes totales del TFG.

El coste de este Trabajo Fin de Grado titulado *Desarrollo de un entorno hardware-software para el control de Drones*, considerando la aplicación del I.G.I.C. sobre el coste final, asciende a un total de *seis mil ochocientos trece con ochenta y un euros*.

Fdo: Andrés Pérez Ortega

19 de julio de 2024

Las Palmas de Gran Canaria