



**DESARROLLO E IMPLEMENTACIÓN DE UN CONTROLADOR DE VUELO Y
ESTACIÓN EN TIERRA PARA UN UAV DE ALA FIJA**

Proyecto de Grado

Programa de Ingeniería electrónica Universidad de los Andes

Autor:
Yesid Camilo Almanza Gamba

Profesora Asesora:
Alba Ávila Bernal

Junio 2024

Índice

Índice de figuras

Índice de cuadros

1 Abstract

This thesis addresses the critical need for efficient data management and operational control in fixed-wing UAVs, targeting the gaps in current commercial flight controllers. The developed flight controller resolves issues related to data storage, retrieval, and bidirectional communication for flight preprogramming both in-flight and on the ground. Key technical features include support for up to 16 actuators, an alarm buzzer, and real-time telemetry to a ground station. The controller operates in two modes: auto-stabilized and fly-by-wire, and monitors critical variables such as temperature, pressure, altitude, yaw, pitch, roll, and GPS coordinates in real-time. The controller was tested in a fixed-wing UAV. The tests demonstrated the system's reliability and effectiveness in maintaining stable flight conditions and accurate data logging. It is important to highlight that all these improvements were embedded into a single system, making it innovative. A noteworthy aspect is the shift from the traditional STM32 architecture to the ESP32-S3, enhancing the system's capabilities. This innovation significantly enhances UAV data management, operational efficiency, and safety, impacting UAV operators and researchers by providing a robust, reliable, and efficient flight control system.

Resumen

Esta tesis aborda la necesidad crítica de una gestión eficiente de datos y control operativo en UAVs de ala fija, enfocándose en las brechas de los controladores de vuelo comerciales actuales. El controlador de vuelo desarrollado resuelve problemas relacionados con el almacenamiento de datos, recuperación y comunicación bidireccional para la preprogramación de vuelos tanto en vuelo como en tierra. Las características técnicas clave incluyen soporte para hasta 16 actuadores, un buzzer de alarma y telemetría en tiempo real a una estación terrestre. El controlador opera en dos modos: autoestabilizado y fly-by-wire, y monitorea en tiempo real variables críticas como temperatura, presión, altitud, yaw, pitch, roll y coordenadas GPS. El controlador fue probado en un UAV de ala fija. Las pruebas demostraron la fiabilidad y efectividad del sistema para mantener condiciones de vuelo estables y un registro de datos preciso. Es importante destacar que todas estas mejoras fueron embebidas en un solo sistema, lo que lo hace novedoso. Un aspecto notable es el cambio de la arquitectura tradicional STM32 a la ESP32-S3, mejorando las capacidades del sistema. Esta innovación mejora significativamente la gestión de datos, la eficiencia operativa y la seguridad de los UAVs, impactando a operadores e investigadores de UAVs al proporcionar un sistema de control de vuelo robusto, fiable y eficiente.

2 Agradecimientos

En primer momento quisiera agradecer a Juana Valentina Camacho, quien ha sido la persona que me ha motivado, ayudado y apoyado en todo este proceso. Seguidamente quisiera agradecer a mi asesora Alba Avila Bernal, quien me brindo un excelente apoyo y ayuda durante este proceso investigativo.

También quisiera agradecer al personal de la Universidad que me ayudo a lo largo de este proyecto, especialmente al técnico del laboratorio de circuitos impresos, Alejandro Monroy Ocasiones. A William y al profesor Felipe Forero por su asesoría en el ambito de comunicaciones.

Quisiera agradecer a mi familia, a mi padre, a mi madre y a mis abuelos por haberme dado todo el amor y ayuda durante estos 4 años de universidad. Agradezco a mi amigo Jorge Luis Montes por haberme ayudado en cualquier problema que se me presentará durante el proceso de desarrollo de la tesis y a lo largo de la universidad.

Finalmente y no menos importante a la iniciativa estudiantil RAS Uniandes, de la cual fui miembro por 4 años y con la que pude desarrollar este proyecto. Seguidamente mi más fervoroso agradecimiento a todas las personas que ayudaron y colaboraron en el proyecto **RAS-UAV Uniandes**. A Jorge Humberto Roca por haberme asesorado en todo el proceso del proyecto; a Simon Pineda por haberme ayudado a desarrollar la interfaz, a Rene Fonseca y Juan David Alfonso por haber resuelto todo el cableado eléctrico de la aeronave; y a Sara Camila Sanchez, Felipe Gutierrez y a David Rincón por haber realizado las mejoras mecánicas y estructurales de la aeronave.

Finalmente, un agradecimiento a los piloto de vuelo Guillermo Vásquez y Johan Barriga quienes nos ayudaron a operar y pilotar el UAV durante la prueba de vuelo.

"Le vent se lève! . . . il faut tenter de vivre!"
("El viento se levanta!... ¡hay que intentar vivir!")
Paul Valéry

3 Introducción

Este proyecto se enfoca en la investigación y desarrollo de un controlador de vuelo destinado a ser implementado en vehículos aéreos no tripulados (UAVs), en este caso particular un UAV de ala fija. La relevancia de este proyecto radica en el papel crucial que desempeña el controlador de vuelo, siendo este dispositivo el encargado del monitoreo, sensado y control de la aeronave [?].

Los sistemas aéreos no tripulados han sido objeto de desarrollo desde los años 80s. Su evolución ha pasado por varias denominaciones. En la década de 1930 se empezó a utilizar el término “drone”, que traducido literalmente del inglés significa “zángano” y fue de uso extendido hasta la década de los 50. En los años 60 apareció la denominación RPV (Remotely Piloted Vehicle), es decir, vehículo pilotado a distancia. Durante los 80, la Autoridad de Aviación Civil norteamericana introdujo cambios conceptuales al concepto RPV, aplicando la denominación “Remotely Operated Aircraft” (ROA), sustituyendo las palabras “vehículo” por “aeronave” y “pilotado” por “operado” [?].

En Colombia, los UAS son utilizados en diversas aplicaciones. La Fuerza Aérea los emplea para seguridad; en agricultura, se utilizan para fumigar cultivos; en fotogrametría, para estudios topográficos; y en mercadeo, para programas publicitarios, entre otros [?]. De acuerdo a una entrevista realizada a Andrés Gomez, ingeniero electrónico que participó en el desarrollo del Vtool de la Fuerza Aérea Colombiana *coelum*; la mayoría de los controladores de vuelo de los UAV utilizados son comerciales e internacionales. Una evaluación del mercado de empresas como SpeedyBee, Ardupilot y ,ha permitido identificar que, aunque pueden ser adquiridos en el mercado nacional, presentan las siguientes características:

- ◊ Manejan un número de actuadores de 1 a 11
- ◊ Tienen un consumo de 300 mA/h
- ◊ Pesan aproximadamente 10 g
- ◊ El control de la nave puede realizarse solo en el aire o solo en tierra.
- ◊ La operación es ciega, lo que indica que el operador de vuelo solo maneja el transmisor pero no tiene conocimiento de la aeronave, por ejemplo, ángulos de giro, altitud, presión atmosférica, entre otros.
- ◊ Tienen una sola unidad de recolección de movimiento inercial.
- ◊ Los modelos comerciales, como el Speedy Bee, cuentan con un sistema de recopilación de datos encriptados.

El controlador de vuelo propuesto posee características diferenciadoras frente a un controlador de vuelo comercial. Primero, busca ampliar significativamente la capacidad de actuadores disponibles, lo que resultará en una mayor maniobrabilidad y versatilidad del UAV. Por otra parte, da prioridad a la integración de dispositivos especializados incluyendo; altímetro, barómetro, gps y sensores iniciales. Estos proporcionarán datos esenciales como la orientación, velocidad angular, aceleración, longitud, latitud, y posición geográfica. Esto mejorará la seguridad, precisión, confiabilidad y rendimiento general del UAV. Por último, se incluye una pantalla de visualización en el UAV, que permitirá verificar el correcto funcionamiento de la posición GPS, los ángulos de orientación y la altura del dispositivo previamente a una misión de vuelo.

El objetivo de este tesis es proponer un controlador de vuelo, implementarlo en un UAV de ala fija real y realizar pruebas de vuelo para evaluar su rendimiento. Esta investigación tiene como propósito principal mejorar la instrumentación de los UAVs de ala fija, lo cual puede tener un impacto significativo en diversas aplicaciones industriales.

Al culminar este proyecto, se logró monitorear y supervisar los ángulos de giro de la aeronave en tiempo real (Yaw Pitch, Roll), así como la altitud, posición GPS y presión atmosférica. Además, junto con los datos de geolocalización y altitud se logró reconstruir la trayectoria seguida por la aeronave durante la prueba de vuelo. Esto puede ser de gran utilidad para industrias en donde se requiera monitoreo constante de aeronaves no tripuladas.

4 Estado del Arte

En los últimos años, la investigación en controladores de vuelo para UAVs (Vehículos Aéreos No Tripulados) ha avanzado considerablemente, centrando sus esfuerzos en el desarrollo de sistemas integrados y eficientes. Una de las áreas más destacadas ha sido la implementación de sistemas de control en tiempo real (RTOS, por sus siglas en inglés), que facilitan una gestión eficiente tanto de las tareas críticas como de las no críticas dentro del controlador de vuelo. Sistemas como el propuesto por Rico et al., hacen uso de arquitecturas de hardware modulares y sistemas operativos híbridos que combinan la planificación First Come First Serve (FCFS) y Earliest Deadline First (EDF) para optimizar la ejecución de tareas, incrementando así la robustez y fiabilidad del controlador [?] [?].

Otra área clave de investigación es la integración de sensores redundantes y la mejora en la precisión de las lecturas mediante técnicas de amortiguación mecánica. Por ejemplo, en el sistema URpilot, los sensores se distribuyen en una arquitectura de placa flexible para reducir el ruido y las vibraciones mecánicas, lo cual mejora la estimación de la actitud del UAV. Además, se están desarrollando métodos avanzados para la fusión de datos provenientes de múltiples unidades de medida inercial (IMUs) y para la detección de fallos en tiempo real, aspectos críticos para asegurar la fiabilidad del vuelo en condiciones adversas [?].

En términos de hardware, se ha logrado una optimización significativa mediante el uso de microcontroladores basados en la arquitectura ARM Cortex-M, como el STM32F767 utilizado en la placa URpilot. Este microcontrolador ofrece un alto rendimiento y capacidades de procesamiento en tiempo real, fundamentales para las aplicaciones profesionales de los UAVs. La incorporación de redundancia a nivel de sensores en esta arquitectura también mejora la precisión y fiabilidad de las mediciones críticas para el control de vuelo [?].

Recientemente, se ha propuesto un RTOS simplificado y específico para aplicaciones de UAV, con el objetivo de aumentar la velocidad de ejecución de tareas, minimizar la latencia y facilitar el uso para los programadores. Este sistema emplea un planificador híbrido en el cual las tareas críticas en tiempo real son controladas por un marco FCFS, mientras que las tareas no críticas son gestionadas por un marco EDF con prioridad dinámica. Esta configuración permite que las tareas no críticas puedan ser manejadas como tareas en tiempo real si su prioridad dinámica alcanza un valor máximo, asegurando una mayor flexibilidad y eficiencia en la gestión de las tareas del UAV [?].

5 Planteamiento del problema

En la actualidad, los controladores de vuelo enfrentan desafíos significativos en la gestión y control de Vehículos Aéreos No Tripulados (UAVs) debido a la complejidad y las exigencias de las operaciones en tiempo real. La investigación reciente en controladores de vuelo ha avanzado considerablemente, destacándose la implementación de sistemas de control en tiempo real (RTOS) que optimizan la gestión de tareas críticas y no críticas, mejorando la robustez y fiabilidad del sistema. Asimismo, la integración de sensores redundantes y técnicas de amortiguación mecánica han mejorado la precisión de las lecturas y la estimación de la actitud del UAV, asegurando la fiabilidad del vuelo en condiciones adversas. El uso de microcontroladores avanzados como el ARM Cortex-M ha permitido una optimización significativa en términos de rendimiento y capacidades de procesamiento en tiempo real.

A pesar de los avances tecnológicos, actualmente los controladores de vuelo presentan algunas deficiencias. En primera instancia, tienen un número de actuadores limitado que pueden controlar. Además, presentan una falta de indicadores visuales y auditivos para verificar el correcto funcionamiento del dispositivo previamente al vuelo. De igual forma, estos dispositivos no tienen incorporado un sistema de telemetría, sino que este se adiciona como un modulo externo. Además, se hace necesario explorar microcontroladores que tengan mayor poder de procesamiento que los que se usan en la actualidad.

Por tanto, se quiere optimizar y mejorar el sistema aéreo no tripulado mediante el desarrollo de un controlador de vuelo que ofrezca mayor fiabilidad y efectividad para mantener condiciones de vuelo estables, así como un monitoreo constante de la aeronave. Esto mediante la implementación de un mecanismo de redundancia en el sistema de movimiento inercial, así como indicadores de funcionamiento durante el vuelo. Adicionalmente, se cuenta con un sistema de telemetría que le permite al controlador a su vez ser un estación en tierra. Esto incluye la implementación de sistemas de control avanzados, integración de sensores de alta precisión, y el uso de microcontroladores de última generación, asegurando así una operación óptima para los UAVs en diversos entornos y condiciones de vuelo.

6 Objetivos

6.1 Objetivo General

Desarrollar un controlador de vuelo para UAVs de ala fija que priorice las condiciones de almacenamiento y recuperación de datos de vuelo. El controlador deberá soportar hasta 16 actuadores, incorporar indicadores de funcionamiento, y permitir la comunicación bidireccional para la programación de vuelos desde el aire o la tierra. El controlador también debe tener la capacidad de ser una Ground Control Station (CGS) para obtener datos en tierra.

6.2 Objetivos Específicos

1. Diseño del Controlador:

Diseñar un controlador de vuelo que cumpla con los requisitos para ser aplicado en un UAV de ala fija. Este proceso incluirá la selección y disposición de componentes electrónicos en la PCB, seguido de una fase de implementación que implicará la verificación del diseño a través del uso del dispositivo en condiciones reales de operación.

2. Diseño de Firmware e Interfaces de Vuelo:

Desarrollar el Firmware necesario para el controlador de vuelo diseñado, incluyendo algoritmos de control, sistemas de comunicación con los sensores y actuadores del UAV, así como el software de las interfaces de vuelo para el monitoreo del sistema.

3. Implementación del Controlador de Vuelo en un UAV de Ala Fija:

Instalar y operar el controlador de vuelo diseñado en un UAV de ala fija real. Se llevará a cabo la integración física del controlador en la aeronave, realizando la conexión con los sensores y actuadores específicos del UAV de ala fija, se realizará una prueba de vuelo para verificar el funcionamiento del controlador en condiciones reales de operación.

7 Marco teórico

Un sistema aéreo no tripulado está compuesto por (ver figura ??): una o varias aeronaves, denominadas Vehículos Aéreos No Tripulados (UAV), sus cargas útiles, las estaciones de control en tierra (y, a menudo, otras estaciones remotas), sistemas de lanzamiento y recuperación de aeronaves, sistemas de soporte, comunicación, transporte y el componente humano requerido para su operación [?].

La arquitectura de un Sistema Aéreo No Tripulado (UAS) integra múltiples componentes esenciales para su funcionamiento eficiente y seguro. Los UAVs son las plataformas aéreas del sistema, equipadas con sistemas de propulsión, navegación, y control, así como diversas cargas para misiones específicas, como cámaras, sensores térmicos o equipos de recolección de datos ambientales. La estación de control en tierra (GCS) es el núcleo operativo del UAS, donde los operadores supervisan y dirigen el vuelo del UAV, analizan la información recibida y toman decisiones críticas durante la misión. Los sistemas de comunicación, que pueden incluir enlaces satelitales y de radiofrecuencia, aseguran una transmisión de datos continua y fiable entre el UAV y la estación de control, permitiendo un control preciso y seguro del vehículo en diversas condiciones operativas [?].

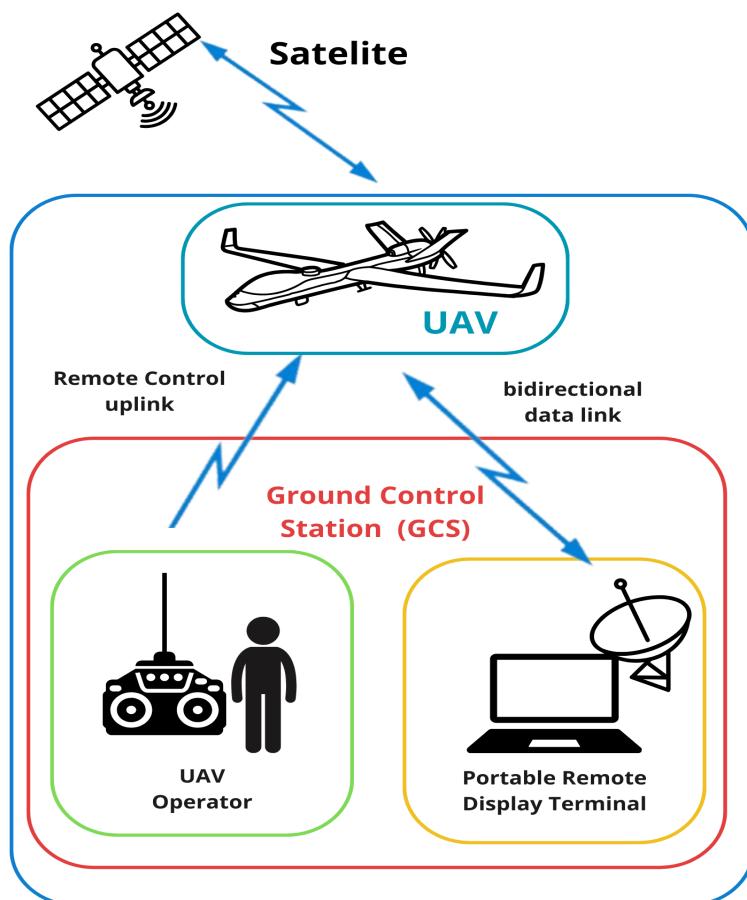


Figura 1: Sistema aéreo no tripulado (UAS)

El subsistema de comunicación es crucial en esta arquitectura, ya que asegura la transmisión bidireccional de datos entre el UAV y la GCS. Este sistema puede incluir enlaces de radiofrecuencia, satelitales y otros

métodos de comunicación redundantes para garantizar la conectividad continua y la seguridad operativa. Una condición importante a tener en cuenta es que los UAS se diseñan para ser operados sin tripulación a bordo de la aeronave. Esto impone la demanda de un control o una unidad de toma de decisiones que debe asumir esta responsabilidad. La combinación de estos elementos permite a los UAS realizar misiones complejas y prolongadas, proporcionando una plataforma flexible y robusta para una amplia gama de aplicaciones [?].

7.1 UAV

Un Vehículo Aéreo No Tripulado (UAV), comúnmente conocido como drone, es una aeronave sin tripulación humana a bordo que requiere un sistema de control eficiente para llevar a cabo una variedad de tareas. Estas tareas pueden incluir misiones de vigilancia, mapeo, entrega de carga, inspección, entre otras [?].

7.1.1. UAV de ala fija

Los Vehículos Aéreos No Tripulados (UAV) de ala fija, también conocidos como drones de ala fija, se caracterizan por sus alas sin movimiento rotacional y su incapacidad para volar verticalmente [?]. A diferencia de los drones multirotor, estos UAVs ofrecen ventajas significativas, como mayor seguridad al no tener hélices giratorias constantes, una mayor eficiencia en el vuelo gracias a su capacidad para planear con el aire, mayor autonomía de vuelo debido a un menor consumo de batería, y la capacidad de cubrir áreas extensas [?]. Además, son resistentes a condiciones climáticas adversas y tienen un costo más económico en comparación con otros tipos de drones. Sin embargo, requieren un espacio amplio para el despegue y el aterrizaje, lo que los hace ideales para aplicaciones que involucran exploración extensa y operaciones en terrenos variados [?].



Figura 2: Dron de ala fija MQ-9 Reaper Hunter [?]

7.2 Ground Control Station (GCS):

La Ground Control Station (GCS) es una parte integral de los Sistemas de Aeronaves No Tripuladas (UAS) que desempeña un papel crucial en la operación y control del UAV. Según la referencia de la Fuerza Aérea Colombiana [?], la GCS se compone de dos componentes principales:

7.2.1. Interfaz de la GCS:

Terminal de Visualización Remota Portátil: Esta interfaz proporciona a los operadores una plataforma para monitorizar y controlar el UAV. Incluye pantallas y dispositivos de entrada que permiten la visualización en tiempo real de los datos de vuelo y la telemetría del UAV, así como la capacidad de enviar comandos al UAV. La interfaz es esencial para la gestión de misiones y la toma de decisiones durante el vuelo.

:

7.2.2. Operador del UAV:

El operador es responsable de la supervisión directa y el control del UAV a través de la GCS. Utiliza dispositivos de control remoto y la interfaz de la GCS para dirigir el vuelo, realizar ajustes en tiempo real y responder a

cualquier eventualidad que pueda surgir durante la operación del UAV. El operador aéreo debe estar entrenado y ser competente en el manejo de los sistemas y procedimientos del UAS.

7.3 Controlador de Vuelo

Un controlador de vuelo consiste esencialmente en un circuito electrónico con distintos componentes electrónicos tales como micro-chips, condensadores, capacitores, resonadores, inductores, entre otros. Este dispositivo se comporta en esencia como el cerebro de un drone, puesto que tiene un procesador que computa todos los datos referentes a la aeronave y ejecuta acciones de control en el vuelo. Además, posee un software que supervisa las acciones del drone. Las funciones del controlador pueden ser divididas en tres categorías principales: percepción, control y comunicación [?].

En el ámbito de la percepción, el controlador de vuelo se enlaza con una serie de sensores que proveen información vital, como la altura, orientación y velocidad del dron. Estos sensores incluyen unidades de medición inercial (IMU) para calcular la velocidad angular y la aceleración, barómetros para la altitud y sensores de distancia para detectar obstáculos. El controlador procesa y combina esta información sensorial para generar acciones de control eficaces y precisas [?].

Por otro lado, en el ámbito de control el controlador de vuelo efectúa las acciones de movimiento del dispositivo, regulando el movimiento de los distintos actuadores del UAV, tales como motores y servomotores para regular las dinámicas de vuelo de la aeronave. Es importante tener en cuenta que en la creación de un controlador de vuelo para UAV de ala fija, se debe diseñar un sistema capaz de gestionar de manera precisa y coordinada los alerones, timón y cola de la aeronave. Esto implica la necesidad de un control preciso de las superficies de control del UAV, como los alerones para el balanceo, el timón para el giro y la cola para el control de cabeceo, para garantizar una navegación estable y maniobras precisas en vuelo.

Por su parte, en el ámbito de comunicaciones, el controlador de vuelo es capaz de comunicarse en tierra con la estación en base. De tal forma que se pueda efectuar una comunicación bidireccional entre el operador de vuelo y el UAV. Además, requiere la integración de un sistema de posicionamiento por satélite (GPS) para conocer la posición en tiempo real del UAV.

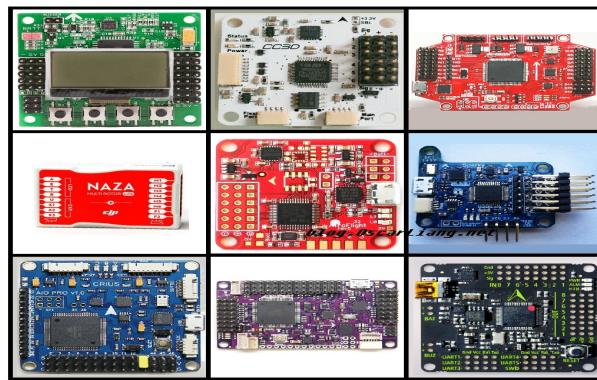


Figura 3: Controladores de Vuelo [?]

7.4 Ejes iniciales del avión

Los ejes de simetría de un avión se conocen como: *yaw*, *pitch* y *roll*. Estos términos se refieren a las tres dimensiones en las que una aeronave en vuelo puede rotar. Estos ejes, que se mueven y rotan con el vehículo, son fundamentales para el control y la maniobrabilidad de una aeronave. [?]

7.4.1. Yaw o eje vertical:

Este eje va de arriba abajo, perpendicular a los otros dos ejes y paralelo al fuselaje. El movimiento alrededor de este eje se denomina *yaw*, y un movimiento positivo en esta dirección mueve la nariz de la aeronave hacia la derecha, siendo el timón la principal superficie de control. [?]

7.4.2. Pitch o eje transversal:

Extendiéndose de izquierda a derecha desde la perspectiva del piloto y paralelo a las alas, este eje controla el movimiento de **pitch**. Un movimiento positivo eleva la nariz de la aeronave y baja la cola, siendo controlado principalmente por las elevaciones. [?]

7.4.3. Roll o eje longitudinal:

Este eje recorre la aeronave de cola a nariz, en la dirección normal de vuelo. El movimiento alrededor de este eje, conocido como *roll*, implica levantar un ala mientras se baja la otra. Los alerones son los principales controladores de este movimiento, aunque el timón también puede influir secundariamente. [?]

Los movimientos alrededor de estos ejes son producidos por torques o momentos, que en las aeronaves se generan intencionalmente mediante el movimiento de superficies de control. Las elevaciones en la cola horizontal controlan el *pitch*, el timón en la cola vertical controla el *yaw*, y los alerones en las alas, que se mueven en direcciones opuestas, controlan el *roll*. En el caso de las naves espaciales, estos movimientos suelen ser producidos por un sistema de control de reacción que utiliza pequeños propulsores para aplicar empuje asimétrico. [?]

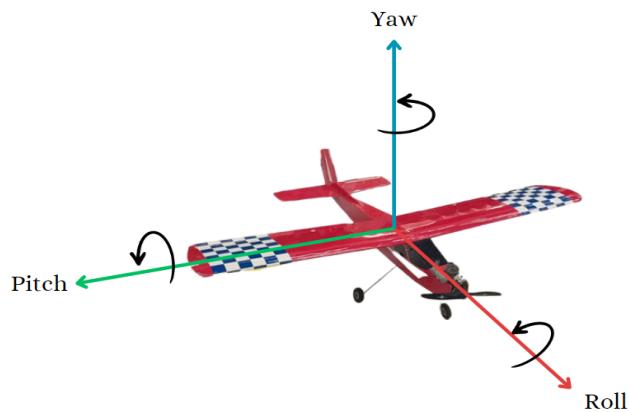


Figura 4: Movimientos Iniciales de un UAV: Yaw, Pitch, Roll

7.5 Main control unit (MCU)

Una MCU, o unidad de control principal, es una plataforma integrada que alberga un procesador, memoria y múltiples pines para la interacción con el entorno. En UAVs, la MCU es crucial para la lectura de señales de los sensores, como GPS y giroscopios, y para el envío de comandos a los actuadores del vehículo, permitiendo así el vuelo autónomo y la navegación precisa. [?]

7.6 Unidad de Medida Inercial (IMU)

Una Unidad de Medida Inercial (IMU) es un dispositivo electrónico que mide y reporta la velocidad angular y la aceleración de un objeto para determinar su orientación y velocidad de movimiento. Principalmente, la IMU se utiliza en la navegación de aeronaves, UAVs, misiles y satélites, operando de manera independiente y libre de interferencias de sistemas externos. Está compuesta por acelerómetros, giróscopos y magnetómetros, los cuales, mediante el uso de algoritmos de fusión de sensores, ayudan a proporcionar datos garantizados sobre la actitud, posición y velocidad, lo que resulta en una alta fiabilidad del sistema contra posibles fallos de los sensores. [?]

7.7 Superficies de control

Las superficies de control son componentes aerodinámicos móviles esenciales en la aviación, cuya manipulación permite al piloto modificar la aerodinámica del avión para lograr el desplazamiento deseado sobre sus ejes y seguir la trayectoria de vuelo establecida. Existen tres tipos principales de superficies de control: ailerones, elevadores y el timón de dirección. Cada una de estas superficies es responsable del control del movimiento en torno a un eje específico del avión.

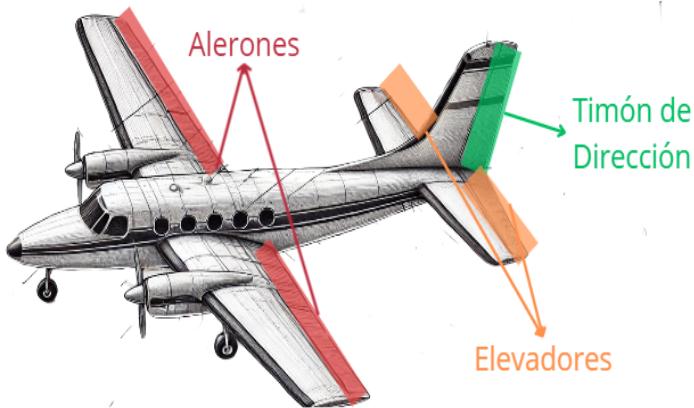


Figura 5: Superficies de control en una aeronave

7.8 Sistema de Posicionamiento Global (GPS)

El GPS (Sistema de Posicionamiento Global) permite la ubicación geográfica de cualquier objeto en el mundo mediante la triangulación de señales en una red de satélites. Este sistema es crucial en la navegación precisa y el control de vuelo de los UAVs (Vehículos Aéreos No Tripulados), ya que les permite conocer su posición exacta en tiempo real. [?]

7.9 Indicadores de vuelo

Los indicadores de vuelo proporcionan información fundamental sobre las condiciones y la orientación del UAV, permitiendo a los operadores realizar ajustes necesarios para asegurar una operación de vuelo en condiciones seguras. En este proyecto, hemos implementado los siguientes indicadores de vuelo:

- ◊ **Anemómetro:** Este dispositivo mide la velocidad del aire en relación con el UAV. Es crucial para determinar la velocidad relativa del UAV y ajustar la velocidad de vuelo para mantener un control adecuado y eficiente.
- ◊ **Horizonte Artificial:** Este indicador muestra la orientación del UAV en relación con el horizonte terrestre. Es fundamental para mantener el equilibrio y la estabilidad durante el vuelo, proporcionando información sobre el ángulo de inclinación y el cabeceo del UAV.
- ◊ **Altímetro:** El altímetro mide la altitud del UAV sobre el nivel del mar. Esta información es vital para mantener el UAV a una altura segura y para cumplir con las regulaciones de espacio aéreo.
- ◊ **Indicador de Dirección:** Este dispositivo muestra la dirección del vuelo en relación con el norte geográfico. Es esencial para la navegación y para seguir una ruta de vuelo planificada.
- ◊ **Indicador de Altitud:** Similar al altímetro, este indicador proporciona información sobre la altitud, pero está calibrado para mostrar variaciones más precisas y rápidas en la altitud del UAV, lo cual es útil para maniobras precisas y ajustes rápidos.

7.10 Normativa seguida en el desarrollo del proyecto

En el desarrollo del proyecto se siguieron los conceptos y esquemas propuestos en la Circular 328-AN/190 de la OACI [?]. Conforme a la normativa, se implementaron sistemas de mando y control (C2) fiables, asegurando la capacidad del piloto remoto para operar el UAV de manera segura y eficiente en todo momento.

Se tomaron en cuenta tanto los estándares existentes para aeronaves tripuladas como los específicos para abordar las diferencias operativas, legales y de seguridad entre aeronaves tripuladas y no tripuladas. El proyecto garantizó que siempre hubiera un piloto responsable de la operación del UAV, manteniendo la capacidad del piloto remoto para intervenir y gestionar el vuelo cuando fuera necesario, cumpliendo así con las reglas del aire del Estado y el espacio aéreo correspondiente.

Asimismo, se estudió el término *aeronave pilotada a distancia* (RPA) para reflejar el estatus de estas aeronaves como pilotadas por un piloto remoto con licencia, ubicado en una estación externa a la aeronave, quien monitorea y gestiona el vuelo en todo momento. Esto aseguró una capacidad operativa comparable a la de las aeronaves tripuladas, permitiendo una intervención y control inmediatos del vuelo.

El proyecto también consideró la posible expansión de los roles de las RPA, aprovechando sus beneficios naturales como largas duraciones de vuelo, capacidades operativas encubiertas y costos operativos reducidos, aplicables a áreas como la seguridad, la agricultura y el análisis ambiental. Todo esto se realizó siguiendo los estándares y regulaciones definidos para garantizar la integración segura y efectiva de las RPA en el ecosistema de la aviación civil.

8 Metodología

La metodología para el desarrollo del controlador de vuelo para un UAV de ala fija se puede observar en la figura ??, en esta figura se observa la estructuración del proyecto en varias etapas, desde la investigación inicial hasta la prueba de vuelo. Comienza con la investigación y la creación de un diagrama funcional, seguido del desarrollo del hardware, el enclosure electrónico, el firmware y la interfaz de usuario. El hardware incluye el diseño y la fabricación electrónica del controlador de vuelo, mientras que el enclosure electrónico es el encargado de dar soporte y protección al circuito. Finalmente, el firmware abarca la adquisición de datos, telecomunicaciones y control de actuadores, y la interfaz se encarga de la visualización de dichos datos. Tanto el firmware como la interfaz de usuario se revisan para asegurar su correcto funcionamiento. Finalmente, se realizan pruebas locales y de vuelo para validar el sistema completo.

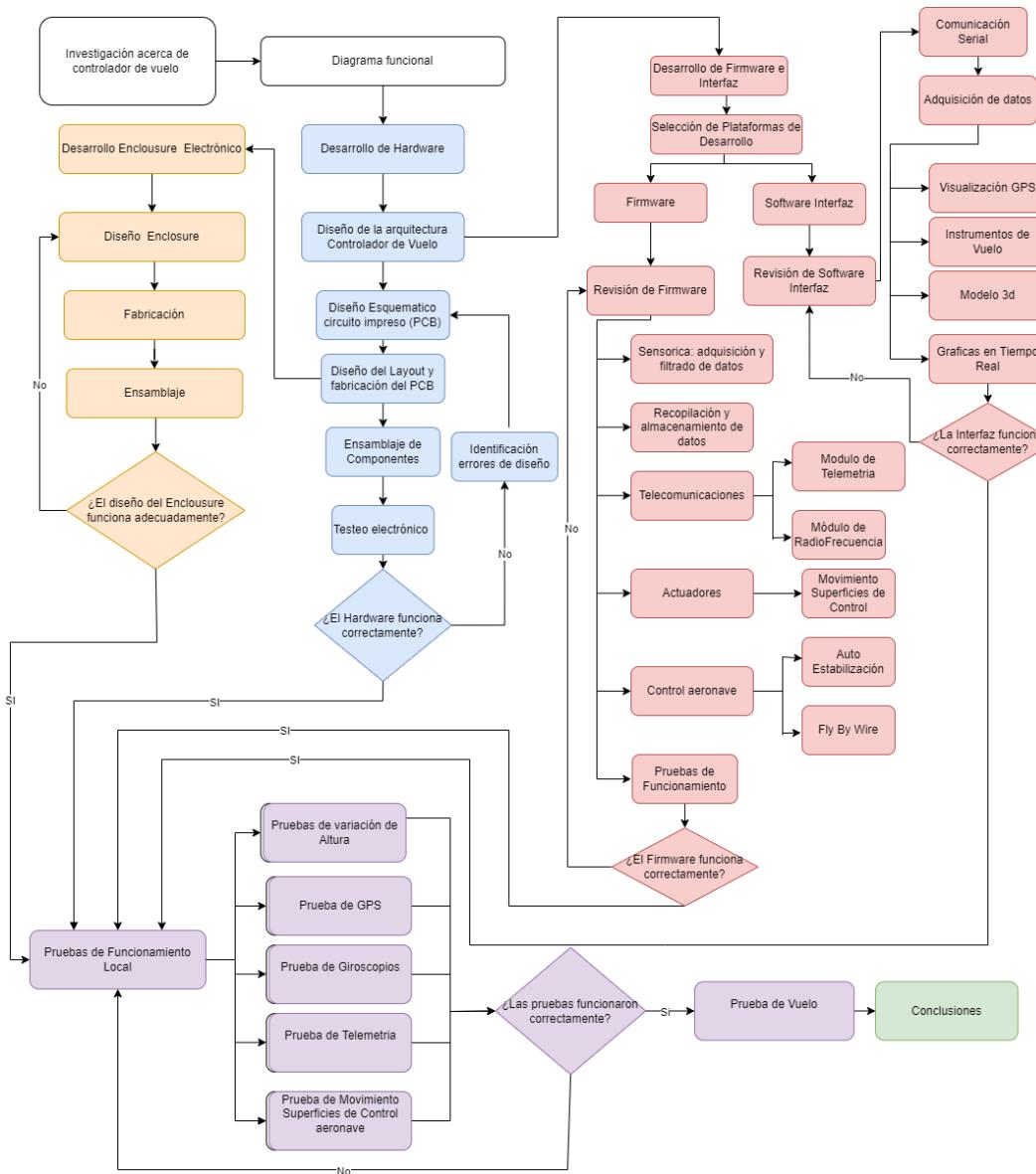


Figura 6: Diagrama Metodología Implementada

9 Diagrama Funcional

En la figura ?? se muestra el diagrama funcional de un controlador de vuelo de UAV de ala fija, este incluye los siguientes componentes esenciales: un barómetro para el cálculo de la presión atmosférica, una unidad de movimiento inercial para calcular la orientación, y un módulo GPS para la localización. Un procesador central que analiza los datos de estos sensores y ejecuta los algoritmos de control. Así mismo una forma de controlar los servomotores, que forman parte del sistema de control de actuadores de vuelo, implementando las órdenes del procesador para manejar los alerones, timón, elevadores y el sistema de propulsión. La unidad de alimentación eléctrica provee la energía necesaria para todos los componentes. Un módulo de telemetría y un módulo de enlace por radiofrecuencia que permiten la transmisión y recepción de datos, facilitando la supervisión y el control remotamente. Finalmente, una unidad de visualización de datos críticos en tiempo real y una unidad de almacenamiento que registra la información relevante del vuelo.

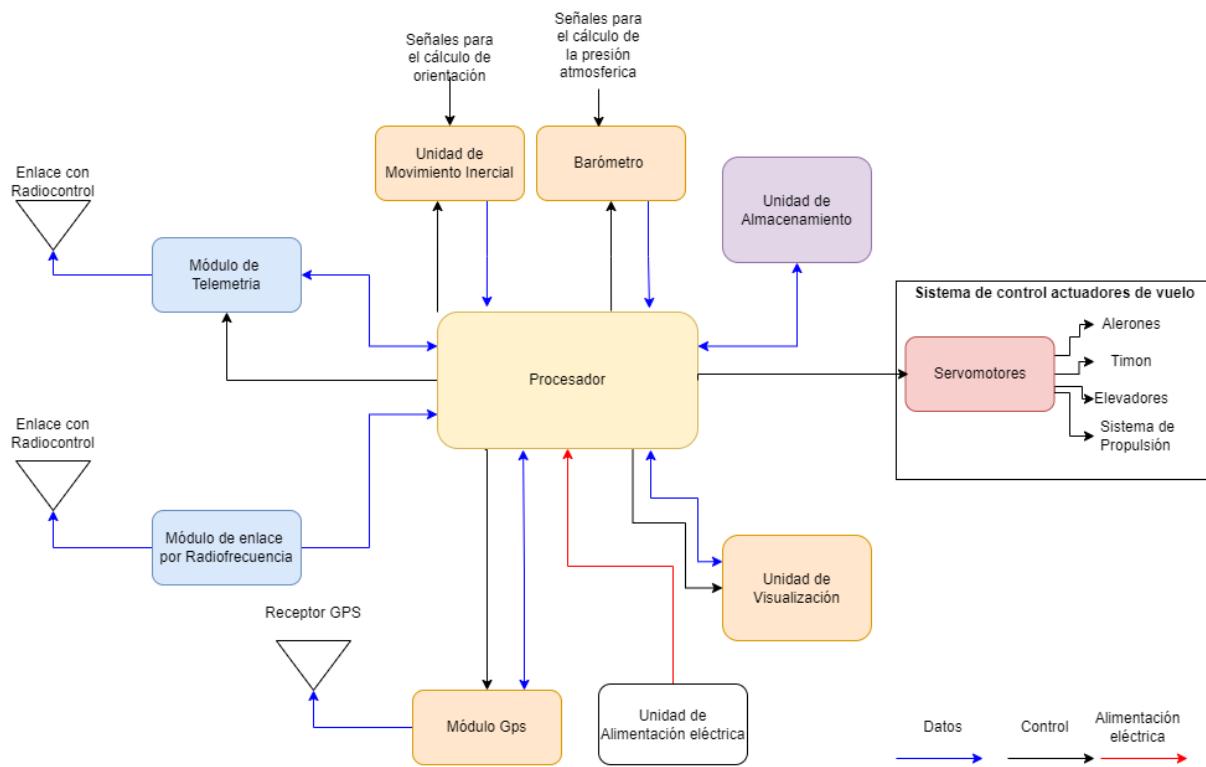


Figura 7: Diagrama de bloques del Controlador de Vuelo para UAV de ala fija

10 Desarrollo de Hardware

10.1 Arquitectura Controlador de Vuelo

Durante la fase inicial del proyecto, se realizó una revisión bibliográfica para definir los componentes esenciales en un controlador de vuelo. Basándome en esta investigación, se diseñó un esquema general de la arquitectura que seguiría el controlador de vuelo, tal y como se muestra en la Figura ??.

La unidad de procesamiento en la arquitectura del controlador de vuelo está a cargo del microcontrolador **ESP32-S3**, que proporciona un procesamiento potente y una gran flexibilidad en la comunicación con otros componentes del sistema. Este microcontrolador de doble núcleo opera a una velocidad de hasta 240 MHz y cuenta con conectividad Wi-Fi y Bluetooth 5.0. Maneja la intercomunicación mediante diversos protocolos estandarizados como I2C, UART y SPI, permitiendo una interacción eficiente con múltiples dispositivos periféricos.

Es relevante mencionar que, en la mayoría de las arquitecturas estudiadas, se utiliza el microcontrolador STM32 Nucleo F767ZI, que, aunque es una opción de alto desempeño con un núcleo ARM Cortex-M7 a 216 MHz, 32 bits, y varias interfaces de comunicación, no cuenta con las capacidades integradas de velocidad, Wi-Fi y Bluetooth que ofrece el ESP32-S3.

Un aspecto crítico es que se incluyen dos Unidades de Medición Inercial (**IMUs**), una actuando como primaria y la otra como secundaria o de respaldo. Este esquema de redundancia aumentará la fiabilidad del sistema. En caso de que falle una **IMU**, la otra puede tomar el control inmediatamente, y así continuar la recolección de datos sobre orientación y movimiento, que son parámetros críticos para la estabilidad y navegación del UAV.

Los **IMUs** a utilizar serán modelos de alta precisión que proporcionan datos esenciales para la orientación, aceleración y velocidades angulares, siendo estos necesarios para las operaciones de vuelo autónomo del UAV y la estabilización del vehículo.

La Figura ?? muestra el diagrama de la arquitectura del controlador de vuelo propuesto, donde se pueden observar los diferentes componentes y su interconexión. Este diagrama destaca la disposición del **ESP32-S3** en el centro, gestionando la comunicación con los sensores, actuadores y unidades de almacenamiento a través de los protocolos **I2C**, **UART**, y **SPI**. La redundancia en las **IMUs** permite que se tenga un sistema de respaldo que pueda reorientar a la aeronave en un caso crítico (sistema de redundancia), lo que junto con la incorporación de sensores adicionales como el **GPS**, **Barómetro**, y el **sistema de almacenamiento** aseguran un sistema robusto y fiable.

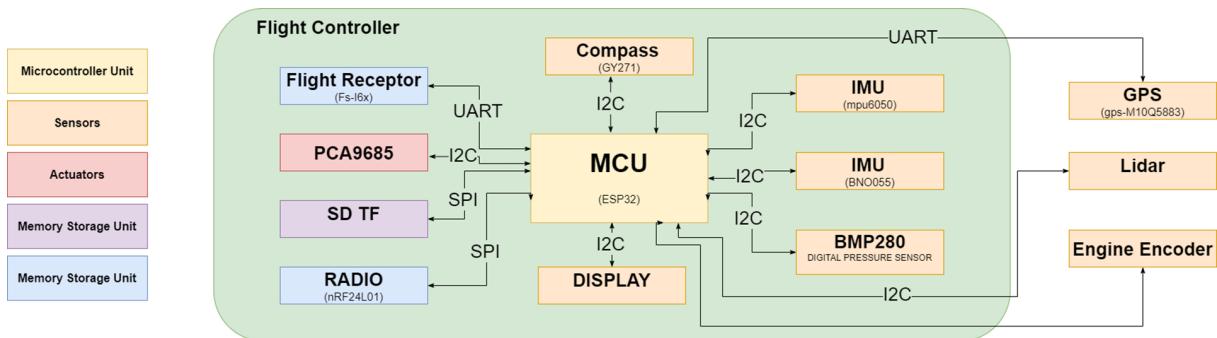


Figura 8: Arquitectura del controlador de vuelo diagrama de Caja Gris

Para ver de forma más detallada como sería esta arquitectura y como se integraría con los distintos módulos, se desarrollo el siguiente diagrama de Caja Blanca (véase ??), en el que se puede observar con más detalle cuales serán los componentes principales en la estructura del dispositivo. De este diagrama todos los dispositivos que están encapsulados por el recuadro verde, indican que están inmersos en el sistema mientras que todos los que se encuentran por fuera son módulos adicionales o que se encuentran en otra parte del Sistema.

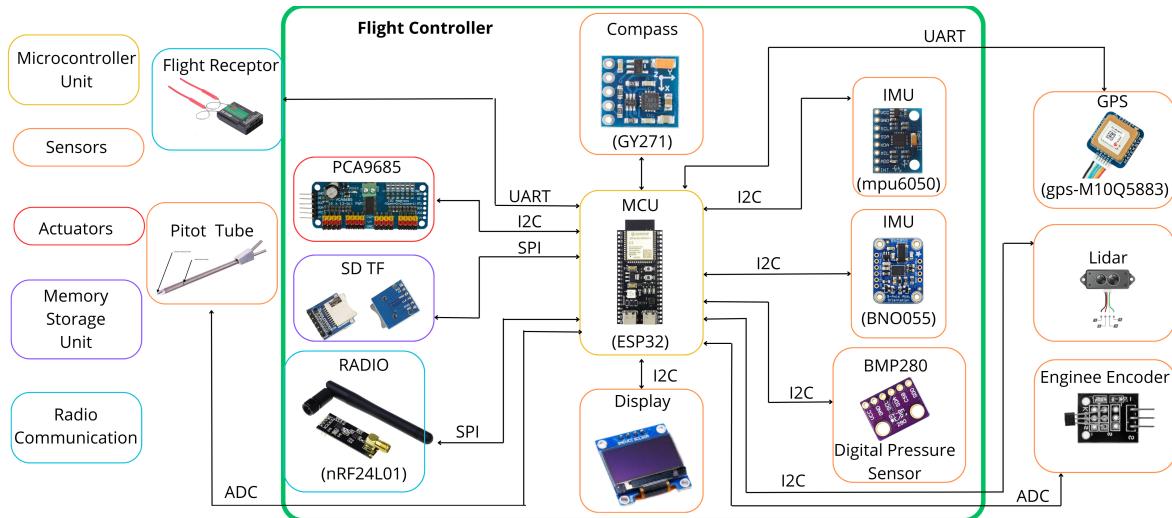


Figura 9: Arquitectura del controlador de vuelo diagrama de Caja Blanca

10.2 Circuito Impreso

Una vez finalizada la etapa de diseño y habiendo terminado de definir los componentes que conformaran el sistema, se procede a diseñar un circuito impreso **PCB**. A lo largo de este proyecto se realizaron en total 2 iteraciones del circuito para mejorar la estabilidad y eficiencia del circuito. El primer circuito fabricado se puede observar en ??, ?? y ?. Para esta primera iteración se utilizó como **MCU** un **ESP32 DevKit v1**, el cual presentaba ciertas deficiencias relacionadas con el protocolo I2C. En esta versión, el módulo **PCA9685** tenía un voltaje de alimentación lógico de **5V**, mientras que todos los demás voltajes de alimentación de los módulos de I2C se encontraban alimentados con **3.3V**. Esto generaba errores con los buses de **SDA** y **SCL**. Este error se puede abordar de dos formas. Agregando resistencias de **Pull-Up** de **4.7 kΩ**, o alimentando toda la parte lógica a **3.3V**. Se decidió optar por la segunda opción.

Otro problema relacionado con la **PCA9685** fue el de alimentación del pin **V+** o el voltaje de entrada del módulo. Puesto que inicialmente se tenía este dispositivo alimentado a una tensión de **+5v**, la cual al momento de realizar pruebas de movimiento de los servomotores generaba un consumo elevado de corriente que causaba un overshoot de voltaje y ocasionaba un reset en el dispositivo.

10.2.1. Diseño del Esquemático Final

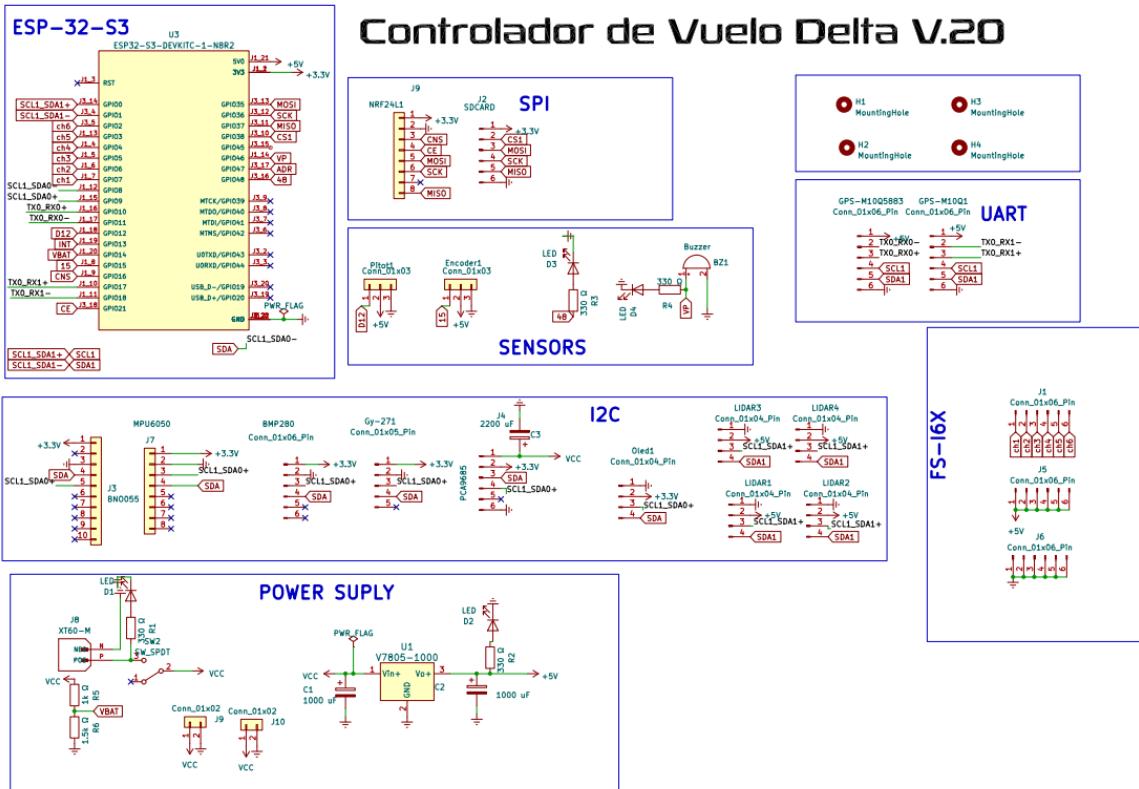
El software donde se diseñaron las dos versiones del circuito fue **KICad** versión **8.0**. Este software fue seleccionado, puesto que es **Open Source** y hay un gran número de huellas y componentes creados por la comunidad.

Para el apartado de diseño del circuito impreso se hace necesario diseñar el esquemático a seguir, el cual consiste en la selección y asociación de **símbolos** y **huellas** con el ruteo eléctrico de todos y cada uno de los componentes del sistema.

El esquemático se dividió en varias partes (véase ??), en este apartado tenemos 7 secciones principales. La primera de ellas localizada en la parte superior izquierda llamada **ESP32-S3** que muestra todas las conexiones

relacionadas con el MCU. Seguidamente se observa el apartado de SPI, en el que se encuentran los distintos módulos configurados bajo este protocolo, los cuales son el **micro SD-CARD Module**, seguido del módulo transmisor **NRF24L1**. En la parte superior derecha se muestra el apartado de **UART** en el que se tiene la posibilidad de tener 2 dispositivos bajo este protocolo, el **GPS** y otro adicional que se requiera. En la mitad se tienen 2 secciones dedicadas a los **sensores** y módulos **I2C**. En la parte inferior derecha está el apartado dedicado al **Power Suply** que es la etapa encargada de gestionar la alimentación del dispositivo. En esta etapa se resalta el hecho de que la entrada debe ser de una batería externa de **7.4 v** y regula a una salida de **5V** para todo el sistema. Finalmente, en la parte inferior derecha se tiene el **FS-16X** el cual corresponde a los pines donde se va a conectar el receptor externo **Flysky**.

Cabe resaltar que se crearon 2 footprints para el dispositivo. El de los módulos de la **PCA9585** y el del **BNO055** los cuales se realizaron en base a la documentación oficial de estos módulos obtenidos directamente de la tienda **Adafruit Learning System**



PCB design (2.0)

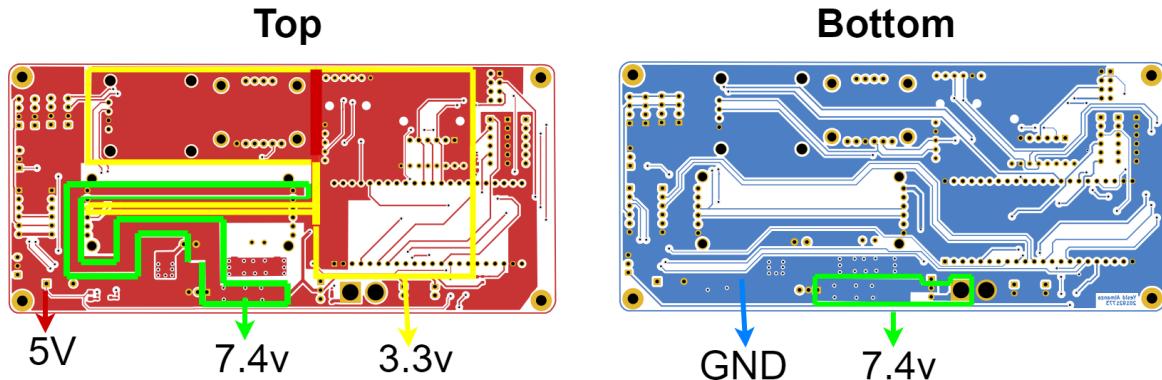


Figura 11: Layout Circuito Impreso 2.0

Para el diseño del layout de la PCB se tuvo en consideración la normativa IPC 2221, la cual establece estándares esenciales para el diseño de placas de circuitos impresos. En la Figura ??, se muestran los parámetros clave de esta normativa, incluyendo el ancho de pista, el ancho de separación y la distribución de componentes. Estas directrices aseguran un diseño robusto y seguro, considerando aspectos como la corriente que pueden soportar las pistas, las distancias de separación necesarias según el voltaje, y una distribución eficiente de los componentes para minimizar interferencias y optimizar el rendimiento del circuito. [?]

IPC 2221

Ancho de Pista

Ancho de pista (mm)	Corriente (Amperios)
0.25	0.3
0.38	0.4
0.5	0.7
0.8	1 A

Ancho de Separación

Voltaje	Distancia
0-15	0.2 mm
15-50	0.4mm-1mm
50-120	8-10 mm o más

Distribución de Componentes



Figura 12: Normativa IPC 2221 [?]

Una vez finalizado el apartado de diseño del layout se procedió a buscar los distintos modelos 3d de los distintos componentes para poder situarlos en un ensamblaje y observar las dimensiones finales del circuito impreso con los distintos módulos. El tamaño final del **pcb** fue de **71.5 mm** de ancho, **156 mm** de largo **21.6 mm** de alto. La distribución de los componentes se puede ver a continuación.

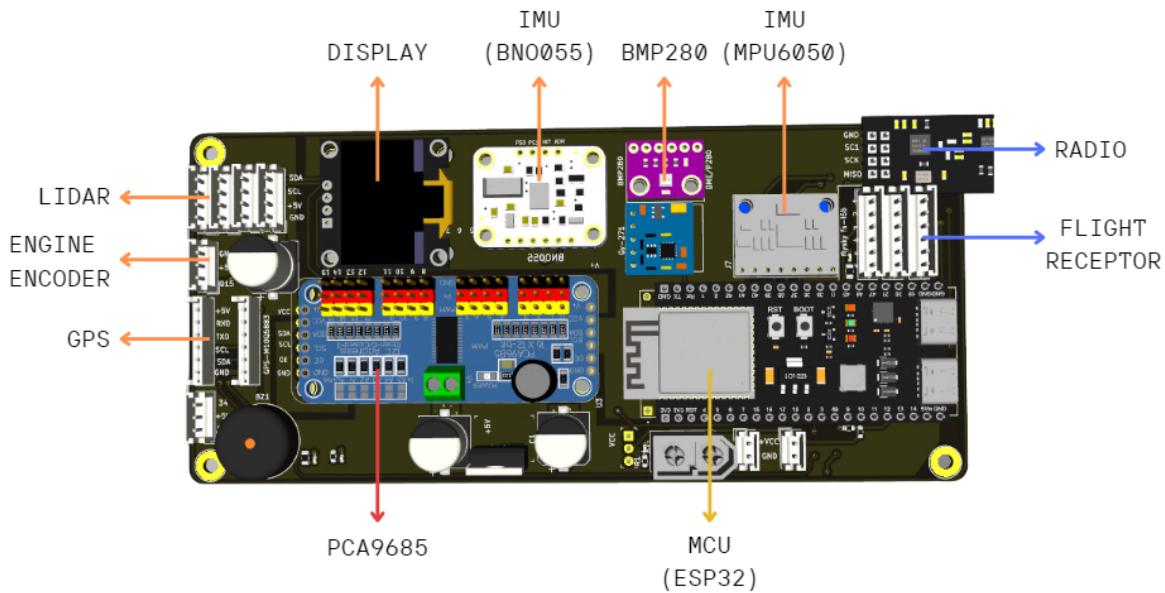


Figura 13: Modelo 3d Circuito Impreso

10.2.3. Fabricación del PCB

El **Printed Circuit Board (PCB)** fue fabricado en el laboratorio de circuitos impresos de la Universidad de los Andes. Se fabricaron tres de los mismos circuitos en caso de que hubiera algún fallo con alguno de los PCBs. Estos PCBs se fabricaron usando un sustrato de FR4, conocido por su excelente estabilidad térmica y propiedades dieléctricas. Además, el PCB cuenta con tecnología True Hole, lo que garantiza la continuidad eléctrica entre capas.

Para asegurar la calidad y fiabilidad, la fabricación siguió el estándar IPC de fabricación de circuitos impresos, garantizando que el PCB cumple con los requisitos estándares de calidad y rendimiento.

10.2.4. Montaje del PCB

La PCB fue ensamblada teniendo en cuenta los grupos de componentes en ambas capas. Todos los módulos están en la capa superior a excepción del módulo de lectura SD que se encuentra en la capa inferior. Se tomó esta decisión en particular para ahorrar espacio y mejorar la accesibilidad a los componentes significativos durante el mantenimiento y el ensamblaje.

La mayoría de los módulos fueron conectados a través de regletas de inserción, para que en caso de fallo o actualización, los componentes respectivos puedan ser reemplazados sin esfuerzo. Sin embargo, en el módulo SD de lectura, las regletas de inserción no fueron preferidas, ya que estas pueden causar una mala unión mecánica que lleva a la inestabilidad del sistema. La Figura ?? es una imagen de la tarjeta ensamblada, de la que se puede ver claramente la disposición de los diferentes módulos en la tarjeta. La disposición se ha planeado de tal manera que proporciona un ensamblaje fuerte y adecuado con respecto a las necesidades funcionales y de mantenimiento del equipo.

PCB Ensamblada

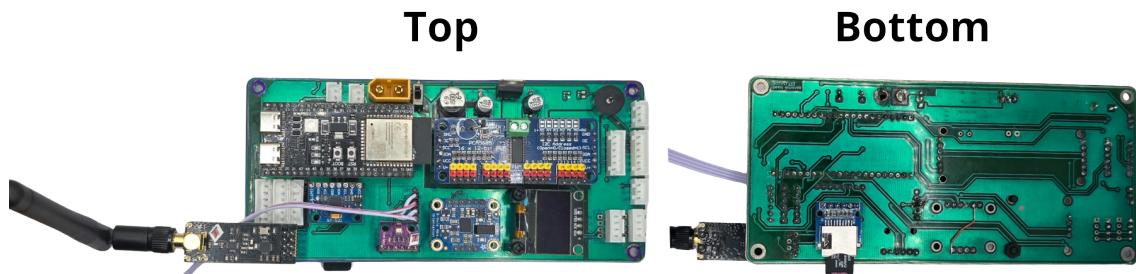


Figura 14: PCB ensamblada

11 Desarrollo Enclosure electrónico

11.1 Diseño Enclosure electrónico

En cuanto al diseño de la enclosure para el circuito impreso, este se diseño utilizando el software comercial de diseño asistido por computadora **Fusion360**. Para el desarrollo final del enclosure se crearon tres modelos durante el proceso de desarrollo. El primero para la versión antigua de la placa de circuito impreso, y los otros dos para el rediseño de la placa. El plano explosionado del modelo de la carcasa y el modelo ensamblado pueden ser vistos en la Figura ???. También, los archivos de las carcasa desarrolladas se pueden encontrar en los anexos (véase ??).

Carcasa PCB

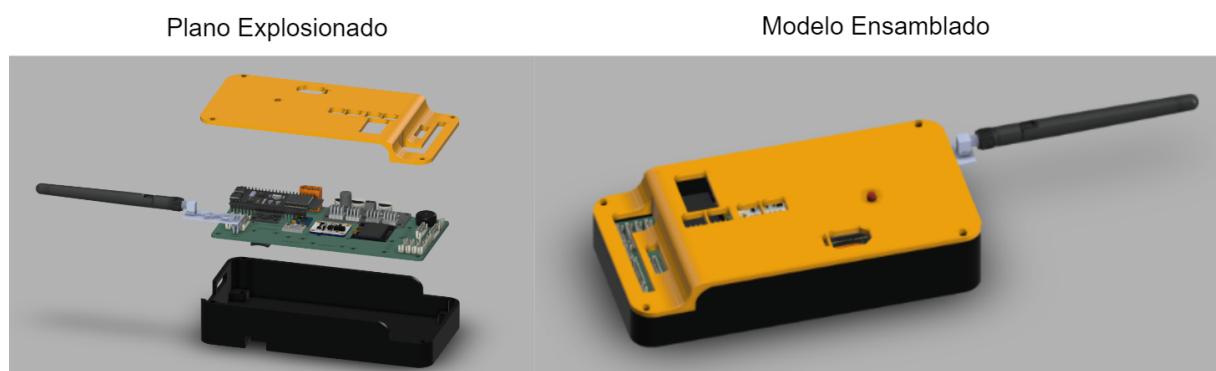


Figura 15: Carcasa PCB modelo Final

11.2 Fabricación Enclosure electrónico

La fabricación del enclosure electrónico se fabricó utilizando una impresora 3D. Se utilizó PETG para la parte inferior y PLA para la parte superior del chasis. Se utilizó PETG para la parte inferior porque es más resistente

que el PLA y puede soportar la tensión mecánica que se asocia con el montaje y la manipulación del usuario, se resalta el uso de insertos metálicos fijados con calor para asegurar correctamente el circuito en la parte inferior de la carcasa. Se utilizó PLA para la parte superior pues esta no requiere de gran resistencia mecánica asociada a su uso y tiene menor densidad. El montaje y producto final se puede observar en ??.

Producto Final



Figura 16: Producto Final ensamblado

12 Desarrollo de Firmware

El desarrollo del firmware se abordó desde varios puntos que se trabajaron en paralelo para garantizar su correcto funcionamiento. El firmware desarrollado fue estructurado considerando que el lenguaje de programación utilizado es C++, optando por la filosofía de programación orientada a objetos. Por lo tanto, se implementaron clases independientes para los diferentes módulos del sistema, como Sensórica, Control y el Receptor FlySky. Estas clases son gestionadas desde el programa principal (main), asegurando una organización modular y facilitando el mantenimiento y la escalabilidad del código. Esta metodología permite una interacción clara y definida entre los distintos componentes del sistema, asegurando que cada módulo funcione de manera autónoma pero coordinada dentro del controlador de vuelo del UAV.

12.1 Entorno de Programación

El firmware desarrollado para el controlador de vuelo fue escrito en C++ utilizando el framework de **Espressif**. Para el desarrollo y gestión del proyecto se empleó PlatformIO, una herramienta multiplataforma que permite programar en diferentes boards de manera eficiente y flexible. El entorno de programación utilizado fue Visual Studio Code, que ofrece una interfaz amigable y numerosas extensiones útiles para el desarrollo de proyectos embebidos.

Inicialmente, el proyecto fue diseñado y programado para una board ESP32. Posteriormente, se migró a una ESP32-S3, una versión más reciente y avanzada de este microcontrolador. Esta migración se realizó para aprovechar las mejoras y características adicionales que ofrece la ESP32-S3, asegurando un rendimiento superior y una mayor capacidad de procesamiento para las tareas del controlador de vuelo.

12.2 Verificación de Conectividad Sensores y Actuadores

Antes de proceder con la sensórica, es fundamental realizar una verificación de los sensores y actuadores conectados al sistema. Este paso incluye un escaneo de las direcciones de los dispositivos I2C, asegurando que todos los componentes estén correctamente conectados al bus I2C. Este escaneo permite identificar qué dispositivos están presentes y si falta alguno, lo que facilita la detección de problemas de conexión antes de inicializar el programa.

12.2.1. Escaneo de direcciones I2C

Para determinar el correcto funcionamiento del bus I2C, se diseñó un software que detecta los dispositivos conectados al bus y cuántos hay. Este procedimiento se ejecuta cada vez que se inicia el dispositivo. A continuación, se presentan los resultados del escaneo de direcciones I2C:

```
Scanning I2C Addresses
00 ...
...
...
28 29 ...
...
30 ...
40 ...
...
...
68 ...
...
76 ...
Scan Completed, 7 I2C Devices found.
I2C devices found.
```

Figura 17: Escaneo I2C de los dispositivos conectados.

Los dispositivos detectados en el bus I2C y sus correspondientes direcciones son:

- ◊ **0x28**: Corresponde al tubo de Pitot.
- ◊ **0x29**: Corresponde al sensor de orientación **BNO055**.
- ◊ **0x40**: Corresponde al controlador de servos PCA9685.
- ◊ **0x76**: Corresponde al sensor de presión barométrica BMP280.
- ◊ **0x68**: Corresponde al acelerómetro y giroscopio **MPU6050**.
- ◊ **0x00**: Corresponde a la pantalla OLED.

12.3 Sensorica y adquisición de datos

Los sensores son uno de los componentes más críticos en el controlador de vuelo, proporcionando los datos necesarios para la navegación y el control de estabilidad del UAV. Estos incluyen dos Unidades de Medición Inerciales **IMUs** para la detección de movimiento y orientación de la aeronave, un **barómetro** para medir la presión atmosférica y la altura de la aeronave. Finalmente un módulo **GPS** para obtener la posición geográfica de la aeronave en todo momento.

12.3.1. Unidades de Medición Inerciales

En el tema de las Unidades de Medición Inerciales IMUs se seleccionan como IMU principal al **BNO055** debido a que es un sensor de alta precisión que integra acelerómetro, giroscopio, magnetómetro y un procesador ARM Cortex-M0 para el cálculo en tiempo real de la orientación y movimiento, ofreciendo datos fiables y precisos con una frecuencia de actualización de hasta 100 Hz [?]. En contraste, el **MPU6050**, aunque menos avanzado, es una opción económica que proporciona seis grados de libertad y capacidades decentes para la medición de movimiento y orientación gracias a su giroscopio y acelerómetro integrados. Se comunican ambos a través del protocolo de comunicación I2C y se puede programar para diferentes sensibilidades. [?]

En caso de que se produzca un fallo con el **BNO055**, el **MPU6050** serviría como un sistema de respaldo eficiente. Aunque no ofrece la misma precisión y el conjunto de características que el **BNO055**, su capacidad para proporcionar seguimiento de movimiento y datos de inclinación lo convierte en un sustituto confiable que puede mantener la funcionalidad del sistema hasta cierto nivel hasta que se resuelva el problema principal.

La comparación entre las especificaciones del sensor **BNO055** y el sensor **MPU6050** revela diferencias significativas en sus capacidades de procesamiento de datos y aplicaciones. El **BNO055**, como se detalla en el Cuadro ??, ofrece cálculos internos que proporcionan la orientación del dispositivo en diferentes formatos, incluidos grados, radianes y cuaterniones. Esto significa que el **BNO055** entrega directamente datos de orientación tridimensional, vector de velocidad angular, vector de aceleración, vector de fuerza del campo magnético y temperatura, sin

necesidad de procesamiento adicional. Este enfoque integrado facilita su uso en aplicaciones donde se requiere una rápida interpretación y utilización de los datos de movimiento.

Por otro lado, el sensor **MPU6050**, según el Cuadro ??, requiere un postprocesamiento de la aceleración angular en cada uno de los ejes del dispositivo para obtener la orientación. Mientras que el **MPU6050** proporciona datos de giroscopio y acelerómetro en bruto, junto con soporte para filtros y funciones programables, el cálculo de la orientación en grados, radianes o cuaterniones debe realizarse externamente. Esto implica un esfuerzo adicional en el procesamiento de datos para lograr el mismo nivel de interpretación inmediata que ofrece el **BNO055**.

En resumen, mientras que el **BNO055** simplifica la integración y el uso con su capacidad de cálculo interno de orientación, el **MPU6050** requiere un procesamiento posterior de los datos para obtener resultados similares, lo que puede ser una consideración importante en el diseño y la implementación de sistemas que dependen de la precisión y rapidez de los datos de orientación.

12.3.1a. **BNO055**

Cuadro I: Especificaciones del Sensor **BNO055**

Característica	Descripción
Orientación Absoluta (Vector Euler, 100Hz)	Datos de orientación tridimensional basados en una esfera de 360°
Orientación Absoluta (Quaternión, 100Hz)	Salida de cuaternion de cuatro puntos para manipulación precisa de datos
Vector de Velocidad Angular (100Hz)	Tres ejes de 'velocidad de rotación' en rad/s
Vector de Aceleración (100Hz)	Tres ejes de aceleración (gravedad + movimiento lineal) en m/s ²
Vector de Fuerza del Campo Magnético (20Hz)	Tres ejes de detección de campo magnético en micro Tesla (uT)
Vector de Aceleración Lineal (100Hz)	Tres ejes de datos de aceleración lineal (aceleración menos gravedad) en m/s ²
Vector de Gravedad (100Hz)	Tres ejes de aceleración gravitacional (menos cualquier movimiento) en m/s ²
Temperatura (1Hz)	Temperatura ambiente en grados Celsius

Cuadro II: Especificaciones del Sensor **MPU6050**

Característica	Descripción
Giroscopio	Salida digital en ejes X, Y, Z con escalas de ±250, ±500, ±1000, y ±2000°/sec
Acelerómetro	Salida digital de 3 ejes con escalas de ±2g, ±4g, ±8g, y ±16g
Voltaje de Trabajo	2.375-3.46V
Voltaje de Lógica I2C	1.71V a VDD
Interface Serial	I2C
ADC	16 bits para giroscopio y acelerómetro
Procesador de Movimiento	DMP integrado para procesamiento de movimiento en 3D y algoritmo de reconocimiento de gestos
Filtros	Filtros pasa bajo programables
Interrupciones Programables	Soporta funciones como reconocimiento de gestos, detección de golpe y movimiento

12.3.1b. Marcos de Referencia para la Medición de Ángulos en el Controlador de Vuelo
 Inicialmente, se estableció el marco de referencia de los ángulos que va a medir el controlador con respecto al marco de referencia del UAV. Los ejes ilustrados representan las orientaciones críticas: yaw (giro alrededor del eje vertical), pitch (inclinación hacia adelante o hacia atrás) y roll (rotación sobre el eje longitudinal) ?? . Estos ángulos son fundamentales para el control de la orientación del UAV, y el controlador de vuelo está diseñado para recopilar y procesar datos sobre cada uno de estos ejes, permitiendo así ajustes precisos en tiempo real para mantener la estabilidad del dispositivo.

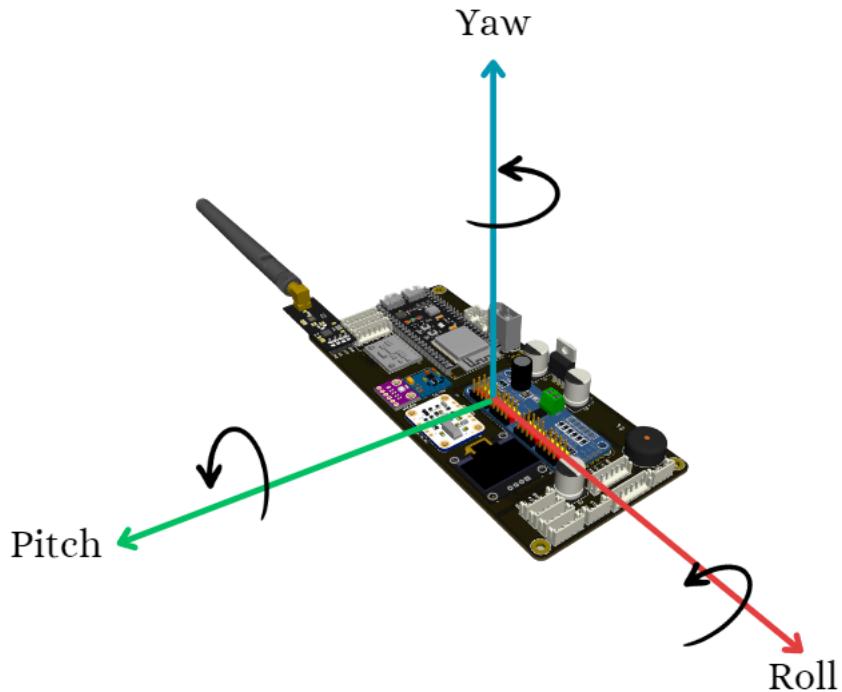


Figura 18: Movimientos Inerciales Yaw, Pitch y Roll en el controlador de vuelo

12.3.1c. *Filtrado y recopilación de datos de movimientos inerciales*

Para este apartado, se leyeron y procesaron datos tomados del sensor **MPU6050**. Se realizó un algoritmo en el que se aplican filtros de Kalman para estimar los ángulos de orientación del dispositivo (Roll, Pitch, Yaw) a partir de las mediciones del sensor. Esto con el objetivo de mejorar la precisión de estas estimaciones al reducir el ruido y combinar de manera óptima ambas fuentes de datos.

El filtro de Kalman es un enfoque matemático que se utiliza ampliamente en aplicaciones de control y navegación para estimar el estado de sistemas dinámicos a partir de mediciones ruidosas. Este algoritmo se basa en dos fases principales: predicción y actualización. Estas se utilizan iterativamente para mejorar la estimación del estado del sistema con cada nueva medición.

Fase de Predicción

En la fase de predicción, el filtro de Kalman utiliza un modelo del sistema para predecir el próximo estado, basándose en el estado estimado actual y en el conocimiento de cómo evoluciona el sistema con el tiempo. Esta predicción también incluye una estimación de la incertidumbre (o error) asociada con el estado predicho. La incertidumbre se modela como una matriz de covarianza, que cuantifica la confianza en la predicción del estado. Para esto se tiene que:

$$\begin{aligned}\hat{x}_{k|k-1} &= A\hat{x}_{k-1|k-1} + Bu_k \\ P_{k|k-1} &= AP_{k-1|k-1}A^T + Q\end{aligned}$$

En donde:

- ◊ $\hat{x}_{k|k-1}$ es la estimación del estado en el tiempo k , dada la información hasta el tiempo $k-1$.
- ◊ A es la matriz del modelo de transición del estado.
- ◊ $\hat{x}_{k-1|k-1}$ es la estimación del estado en el tiempo $k-1$, dada toda la información hasta el tiempo $k-1$.
- ◊ B es la matriz del modelo de control.
- ◊ u_k es el vector de control en el tiempo k .
- ◊ $P_{k|k-1}$ es la covarianza del error de predicción en el tiempo k .
- ◊ Q es la covarianza del ruido del proceso.

En esta fase de predicción, se actualizará el ángulo de Roll basado en la velocidad angular previa, ajustada por el sesgo estimado, y el paso de tiempo. Además, se actualizan las matrices de covarianza de error para reflejar la nueva incertidumbre después de la predicción.

Fase de Actualización

En la fase de actualización, se ajusta la estimación del estado con base en la nueva medición:

$$\begin{aligned} K_k &= P_{k|k-1} H^T (H P_{k|k-1} H^T + R)^{-1} \\ \hat{x}_{k|k} &= \hat{x}_{k|k-1} + K_k (z_k - H \hat{x}_{k|k-1}) \\ P_{k|k} &= (I - K_k H) P_{k|k-1} \end{aligned}$$

En donde:

- ◊ K_k es la ganancia de Kalman en el tiempo k .
- ◊ H es la matriz del modelo de observación.
- ◊ R es la covarianza del ruido de observación.
- ◊ z_k es la medición real en el tiempo k .
- ◊ I es la matriz identidad, con dimensiones adecuadas para la operación.

En esta fase de actualización, se calcula la discrepancia entre la estimación del ángulo a partir del acelerómetro y la predicción del ángulo de Roll. Además, se calculan las ganancias de Kalman, que determinan cómo se combinan la predicción y la medición para obtener una nueva estimación. Por otro lado, se actualizan el ángulo estimado de Roll y el sesgo utilizando las ganancias de Kalman. Por último, se ajustan las matrices de covarianza de error para reflejar la disminución en la incertidumbre después de la actualización.

Estas ecuaciones se aplican de forma iterativa para cada nueva medición, mejorando así la estimación del estado del sistema a lo largo del tiempo.

En las Figuras ?? y ??, se muestran las gráficas obtenidas al implementar este algoritmo. En estas gráficas podemos ver la diferencia de los datos obtenidos mediante el **MPU6050** antes del proceso de filtrado (*Raw*) y después de aplicar el filtro de Kalman (*Filtered*).

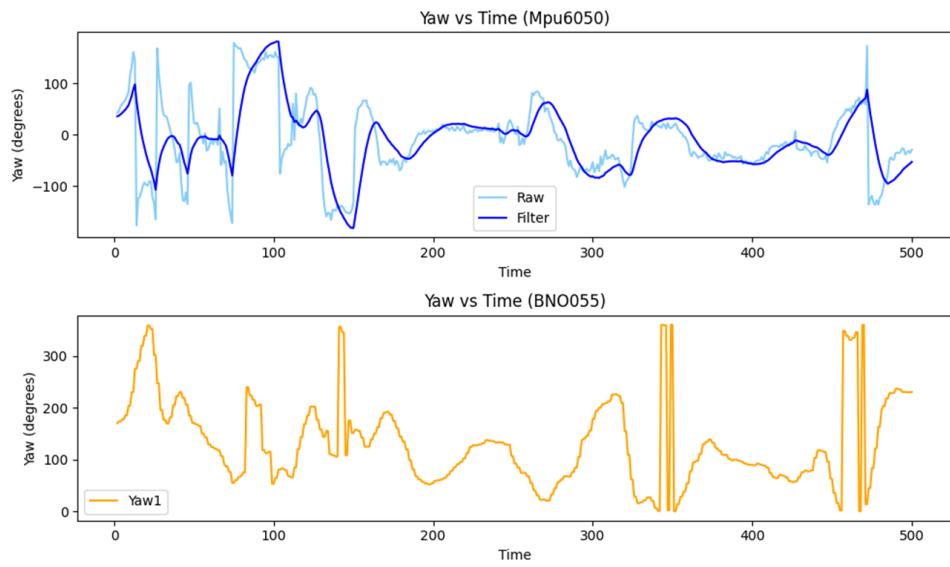


Figura 19: Ángulos de Giro Pitch obtenidos con **MPU6050** y **BNO055**.

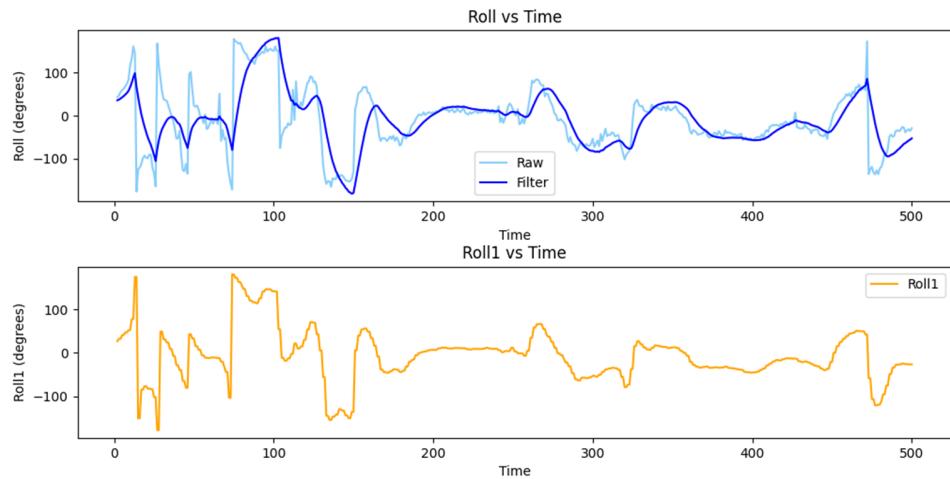


Figura 20: Ángulos de Giro Roll obtenidos con **MPU6050** y **BNO055**.

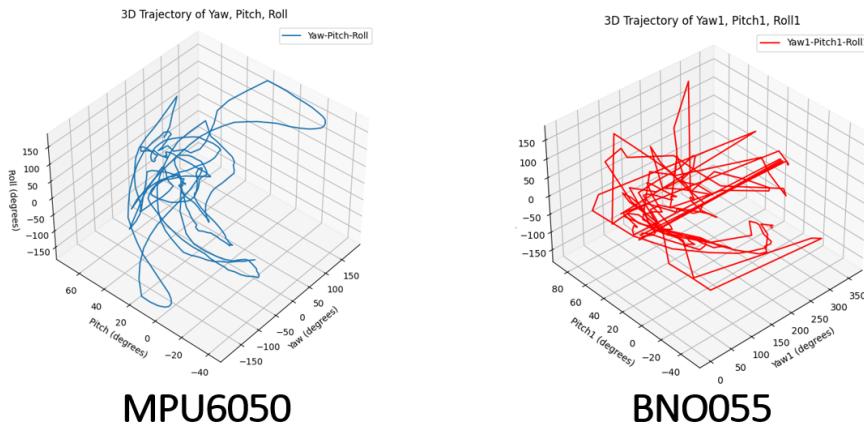


Figura 21: Ángulos de Giro **MPU6050** y **BNO055**.

12.3.1d. Protocolo de Inicialización y Recepción de Datos del BNO055

1. Detección del Sensor:

- ◊ El sensor **BNO055** se inicializa y se intenta establecer comunicación en la dirección I2C predeterminada. Si no se detecta el sensor, se muestra un mensaje de error en el monitor serie.

2. Calibración del Sensor:

- ◊ Se entra en un bucle de calibración donde se verifica continuamente el estado de calibración del sensor. Durante este proceso, se muestra el estado de calibración en una pantalla OLED.

3. Contador de 5 Segundos:

- ◊ Despues de la calibración, se muestra un contador de 5 segundos en la pantalla OLED, permitiendo que el usuario coloque el dispositivo en una posición de referencia estable.

4. Cálculo y Visualización de los Offsets:

- ◊ Se calculan los offsets de los ángulos (roll, pitch y yaw) en la posición de referencia y se muestran en la pantalla OLED.

12.3.1e. Recepción de Datos del BNO055

1. Lectura de Datos de Orientación:

- ◊ Se leen los valores de orientación (yaw, pitch y roll) del sensor y se ajustan estos valores restando los offsets calculados previamente.

2. Lectura de Datos del Acelerómetro:

- ◊ Se leen los valores de aceleración en los ejes X, Y y Z.

3. Lectura de Datos del Magnetómetro:

- ◊ Se leen los valores del campo magnético en los ejes X, Y y Z.

4. Cálculo del Valor de la Brújula:

- ◊ Se calcula el valor del heading utilizando los datos del magnetómetro y luego se aplica un filtro de Media Exponencialmente Ponderada (EMA) para suavizar el valor del heading.

12.3.1f. Cálculo del Heading

El heading (rumbo) se calcula utilizando los componentes X e Y del campo magnético:

$$\text{heading_rad} = \text{atan2}(my, mx)$$

$$\text{heading_deg} = \text{heading_rad} \times \frac{180.0}{\pi}$$

Si el valor calculado del heading es negativo, se ajusta sumándole 360 grados para obtener un valor positivo:

```
if (heading_deg < 0) then heading_deg+ = 360
```

12.3.1g. Suavizado del Heading con EMA

Para reducir la oscilación en los valores de heading, se aplica un filtro de Media Exponencialmente Ponderada (EMA). La fórmula para actualizar el valor suavizado es:

$$\text{smoothedHeading} = \alpha \times \text{newHeading} + (1 - \alpha) \times \text{oldHeading}$$

Donde α es el factor de suavizado. Para manejar la discontinuidad alrededor de 360 grados, se ajusta la diferencia de los valores de heading:

$$\text{diff} = \text{newHeading} - \text{oldHeading}$$

$$\text{if } (\text{diff} > 180) \text{ then } \text{oldHeading}+ = 360$$

$$\text{else if } (\text{diff} < -180) \text{ then } \text{oldHeading}- = 360$$

Finalmente, se ajusta el valor suavizado para mantenerse dentro del rango de 0 a 360 grados:

$$\text{if } (\text{smoothedHeading} \geq 360) \text{ then } \text{smoothedHeading}- = 360$$

$$\text{else if } (\text{smoothedHeading} < 0) \text{ then } \text{smoothedHeading}+ = 360$$

Este método asegura que los valores de heading se mantengan dentro del rango de 0 a 360 grados y se suavicen adecuadamente, reduciendo las fluctuaciones en la señal.

12.3.2. Barómetro, Altímetro y Termómetro

Un barómetro es un instrumento capaz de medir la presión atmosférica, proporcionando datos esenciales para la predicción del tiempo y la altitud. En el contexto de sensores modernos, como el **BMP280**, estas funcionalidades se amplían significativamente. El BMP280 no solo actúa como un barómetro, sino que también integra un altímetro y un termómetro. Esto permite obtener medidas precisas de presión, altitud y temperatura en un solo dispositivo compacto. La capacidad de medir la presión atmosférica es crucial para determinar la altitud en aplicaciones de navegación y control de un UAV de ala fija, mientras que la medición de la temperatura ayuda a saber la temperatura del controlador de vuelo.

12.3.2a. Inicialización del BMP280

Al iniciar el sensor BMP280, se configura la presión atmosférica de referencia para **Bogotá** en **1028 hPa**. Luego, se lee la altura inicial basada en esta referencia. Si la inicialización del sensor falla, se imprime un mensaje de error en el monitor serial. El proceso de inicialización garantiza que las lecturas de altura sean precisas y consistentes con la altitud específica de Bogotá.

12.3.2b. Aplicación del EMA (Exponential Moving Average)

Para suavizar las lecturas de presión atmosférica, altura y temperatura interna del dispositivo, se está aplicando un algoritmo de Media Móvil Exponencial (EMA). Este algoritmo es útil para reducir el ruido en los datos y proporcionar valores más estables y representativos. El EMA calcula el promedio ponderado de los datos recientes, donde los datos más recientes tienen un peso mayor en el cálculo del promedio. Esto se realiza para cada una de las mediciones (presión, altura y temperatura) obtenidas del BMP280.

La fórmula utilizada para el cálculo del EMA es la siguiente:

$$\text{EMA} = (\alpha \cdot \text{currentReading}) + ((1 - \alpha) \cdot \text{previousEMA})$$

Donde:

- ◊ currentReading es la lectura actual del sensor.
- ◊ previousEMA es el valor del EMA anterior.
- ◊ α es el factor de suavizado, un valor entre 0 y 1 que determina el peso de la lectura actual en el cálculo del EMA.

12.3.2c. Proceso de Lectura de Datos

1. **Presión Atmosférica:** Se obtiene directamente del sensor y se aplica el EMA para suavizar las lecturas.
2. **Altura:** Se calcula en función de la presión atmosférica y se ajusta utilizando la altura inicial leída durante la inicialización. También se aplica el EMA para obtener una lectura más estable.
3. **Temperatura Interna:** Se mide directamente desde el BMP280 y se procesa mediante el EMA para reducir el ruido en las lecturas.

12.3.3. Sistema de Posicionamiento Global (GPS)

Para asegurar una alta fiabilidad y precisión en la ubicación del UAV, se seleccionó el **GPS M10Q-5883**. Este dispositivo integra el módulo GNSS SAM-M10Q-00B de u-blox, capaz de recibir simultáneamente señales de cuatro constelaciones **satelitales: GPS, GLONASS, Galileo y BeiDou**. [?] Esta capacidad multiconstelación amplía significativamente la cobertura y mejora la disponibilidad de la señal en condiciones adversas, como en entornos urbanos densos o áreas con obstrucciones. Además, el módulo GNSS SAM-M10Q-00B emplea tecnología avanzada que optimiza la sensibilidad y reduce el tiempo de adquisición de señal, garantizando así una ubicación precisa y dinámica incluso en escenarios complicados. Utilizando el protocolo de comunicaciones UART, este sistema transmite datos de posición de manera precisa y confiable, asegurando que el UAV pueda navegar y operar de forma óptima en diversas condiciones.

12.3.3a. Proceso de Inicialización del GPS

Para inicializar el GPS es necesario establecer la comunicación serial con el dispositivo. Esto se logra utilizando el siguiente código, que configura los pines TX2 y RX2 para la comunicación serial:

```

1 const int TX2 = 11;
2 const int RX2 = 10;
3 Serial2.begin(9600, SERIAL_8N1, RX2, TX2);

```

Una vez que el GPS está conectado y se inicializa la comunicación serial, el dispositivo se encenderá con una luz verde. Es importante situar el GPS en un lugar con cielo abierto, ya que tardará unos minutos en obtener la recepción de las señales de los satélites. Una vez que el módulo recibe la señal de los satélites, comenzará a titilar, indicando que se ha establecido correctamente la comunicación con el satélite.

Para contrastar y validar que los valores recibidos por el GPS eran correctos, se utilizó la aplicación GnssLogger de Google (véase la Figura ??). En esta prueba, se compararon los datos obtenidos por el GPS con los

proporcionados por la red WLS (Wireless Location Service) y el GPS del celular. La prueba se realizó en la terraza del quinto piso del edificio Mario Laserna. Se encontró que todos los datos de ubicación se encontraban dentro de un radio de aproximadamente 6 metros, lo que confirma la precisión de los datos recibidos.

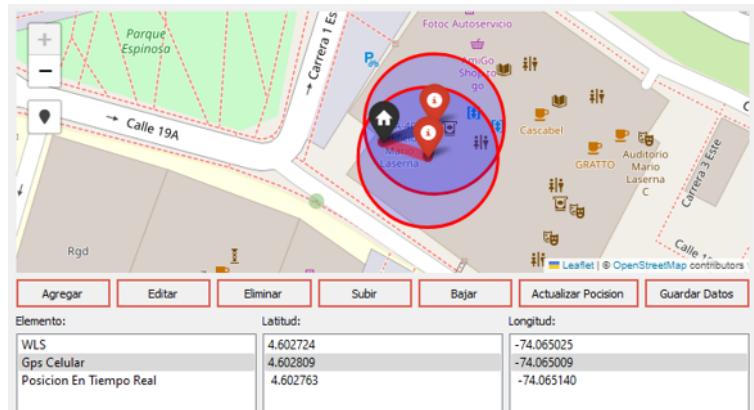


Figura 22: Validación de la precisión del GPS utilizando GnssLogger.

12.4 Recopilación y Almacenamiento de Datos

En el ámbito de las aeronaves, la recopilación y el almacenamiento de datos juegan un papel crucial. Este dispositivo es esencial, ya que permite registrar la información histórica del vuelo, proporcionando un registro detallado de todos los eventos y parámetros operativos. Actúa como una "caja negra" que almacena datos vitales que pueden ser utilizados para análisis post-vuelo, mantenimiento preventivo, y en el desafortunado caso de una eventualidad, para la investigación de incidentes. La capacidad de recopilar y almacenar datos de manera confiable asegura que toda la información relevante del vuelo esté disponible para su revisión.

12.4.1. SD TF

El módulo lector de microSD, también conocido como SD (Secure Digital), es un componente esencial en el dispositivo. Este módulo permite guardar la información y los datos generados por los sensores y los cálculos realizados por la **ESP32-S3**. La microSD funciona mediante un sistema de archivos que permite almacenar y recuperar datos de forma eficiente y confiable. Al inicializar la comunicación con el módulo a través de la **ESP32-S3**, se pueden escribir y leer datos en la tarjeta microSD. La información recopilada se almacenará en formato CSV (Comma-Separated Values), lo que facilita la organización y el análisis posterior de los datos.



Figura 23: Lector de tarjetas MicroSD

Especificaciones Técnicas

- ◊ Modelo: Módulo lector SD TF
- ◊ Interfaz: SPI
- ◊ Voltaje de operación: 3.3V - 5V
- ◊ Compatibilidad: Tarjetas microSD

12.4.2. Conexión y Configuración

El módulo lector SD se comunica con el ESP32-S3 a través del protocolo SPI. A continuación, se detallan las conexiones entre el módulo lector SD y el ESP32-S3:

- ◊ MISO (Master In Slave Out): Conectado al pin MISO del ESP32-S3 **GPIO 37**
- ◊ MOSI (Master Out Slave In): Conectado al pin MOSI del ESP32-S3 **GPIO 35**
- ◊ SCK (Serial Clock): Conectado al pin SCK del ESP32-S3 **GPIO 36**
- ◊ CS (Chip Select): Conectado a un pin GPIO del ESP32-S3 **GPIO 38**

12.5 Telecomunicaciones

En el apartado de telecomunicaciones es importante resaltar la presencia de dos componentes principales que son cruciales para el control y monitoreo del UAV.

El primer componente es el módulo de radiofrecuencia, el cual se encarga de recibir la información del operador de vuelo. Este módulo permite que la aeronave responda a los comandos enviados por el operador, facilitando el control y la maniobrabilidad del UAV. La comunicación por radiofrecuencia es esencial para asegurar que los comandos se transmitan de manera eficiente y sin retrasos, permitiendo una respuesta inmediata del UAV a las instrucciones del operador.

El segundo componente es el módulo de telemetría, que tiene la capacidad de recibir información del UAV en tiempo real. Este módulo proporciona datos vitales sobre el estado y la posición de la aeronave, incluyendo parámetros como la altitud, velocidad, orientación y otros datos críticos de vuelo. La telemetría en tiempo real es fundamental para el monitoreo continuo del UAV.

12.5.1. Módulo de radiofrecuencia Fly-Sky FS-IA6B

En el apartado de radiofrecuencia, se utilizó el transmisor y receptor FS-i6 de la empresa FlySky el cual es un controlador de radio digital de 6 canales diseñado para ofrecer un control preciso y fiable para diversas aplicaciones de modelismo, como drones, aviones, helicópteros y coches RC. Este transmisor utiliza la tecnología AFHDS 2A (Automatic Frequency Hopping Digital System) para asegurar una comunicación sin interferencias y una alta inmunidad al ruido. [?]

Características Principales:

- ◊ **Frecuencia:** 2.4 GHz AFHDS 2A.
- ◊ **Canales:** 6 canales programables.
- ◊ **Pantalla LCD:** Proporciona una interfaz gráfica para la configuración y el monitoreo en tiempo real. [?]
- ◊ **Rango de Operación:** Hasta 1 km en condiciones óptimas. [?]
- ◊ **Compatibilidad:** Compatible con diversos receptores FlySky que soportan AFHDS 2A. [?]
- ◊ **Funciones:** Incluye funciones avanzadas como mezcla de canales, configuración de servos, límite de puntos finales, y modos de vuelo. [?]

Para interconectar este sistema de radio control con el controlador de vuelo, se procedió a realizar un análisis de cómo sería la comunicación del receptor con el dispositivo (véase ??). Se descubrió que cada uno de los 6 canales del dispositivo envía una señal **PWM**, por lo que, si se procesa esta información con la **ESP32-S3** y se decodifica esta señal, es posible transcribir este ancho de pulso en valores para que la **PCA9685** pueda mover los servos al valor deseado.



Figura 24: Conexión del Radio Control con el Controlador de Vuelo

Este sistema se realizó de esta forma ya que es posible decodificar la información del transmisor **Fly-Sky FS-I6B** y poder efectuar una respuesta en los actuadores de la aeronave. Es importante destacar el hecho de al estar leyendo la información y posteriormente enviando esta información a los **servos** se genera un pequeño **delay** que debe tenerse en cuenta al momento de operar la aeronave.

12.5.1a. Proceso de Decodificación

Para decodificar las señales PWM, se utiliza la función `pulseIn` del microcontrolador, que mide la duración del pulso en microsegundos. Esta duración se mapea a un rango de valores especificado mediante la función `map`.

12.5.1b. Medición del Pulso

La función `pulseIn` mide el tiempo durante el cual la señal es alta (nivel lógico alto). Esta duración, t , se expresa en microsegundos (μs). El rango típico de estas duraciones para las señales PWM del receptor FLYSKY-I6B es:

$$t \in [1007 \mu s, 1999 \mu s]$$

12.5.1c. Mapeo de la Señal

Para convertir la duración del pulso a un valor utilizable, se utiliza la función `map`. La función mapea el rango de entrada $[1007, 1999]$ a un rango de salida especificado por los parámetros `minLimit` y `maxLimit`. Estos parámetros descritos anteriormente hacen referencia al valor máximo y mínimo al que pueden llegar cada uno de los servomotores de la aeronave, típicamente este valor es **-30°** y **30°** con respecto al valor de referencia inicial del servo, el cual generalmente es de **90°**. A continuación se muestra la ecuación de mapeo:

$$y = \frac{(\text{maxLimit} - \text{minLimit})}{(1999 - 1007)} \cdot (t - 1007) + \text{minLimit}$$

12.5.1d. Condiciones de Valor por Defecto

Si la duración del pulso t es menor que $100 \mu s$, se asume que la señal es inválida, y se devuelve un valor por defecto, `defaultValue`.

12.5.1e. Lectura de Interruptores

Para leer el estado de un interruptor, se convierte la señal PWM del canal correspondiente a un valor booleano (verdadero o falso). Este proceso implica mapear la señal a un rango de 0 a 100 y determinar si el valor resultante es mayor que 50. La ecuación de mapeo en este caso es:

$$z = \frac{(100 - 0)}{(1999 - 1007)} \cdot (t - 1007) + 0 \quad (1)$$

El resultado se evalúa para determinar el estado del interruptor:

$$\text{estado} = \begin{cases} \text{verdadero} & \text{si } z > 50 \\ \text{falso} & \text{si } z \leq 50 \end{cases} \quad (2)$$

A continuación se muestra un esquema (??) del transmisor FLYSKY-I6B y los valores decodificados de cada uno de los seis canales. Nótese que el **canal 1** hace referencia al movimiento de **roll**, el **canal 2** a **pitch** y el **canal 4** a **yaw** de la aeronave, controlando cada uno un servomotor que representa un movimiento particular de esta. El **canal 3** se encarga de controlar el suministro de combustible al motor. Finalmente, los canales **5** y **6** son interruptores.

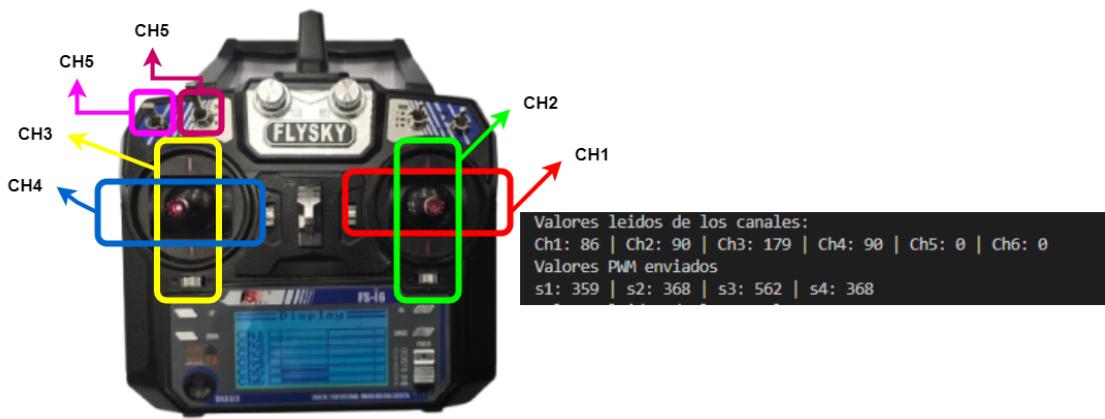


Figura 25: Diagrama Transmisor y decodificado de Señales PWM

12.6 Telemetría

12.6.1. Módulo nRF24L01

El módulo nRF24L01 es un transceptor inalámbrico que opera en la banda de 2.4 GHz, una frecuencia comúnmente utilizada para comunicaciones de corto alcance debido a su equilibrio entre velocidad de transmisión y penetración de obstáculos. Este módulo es conocido por su bajo consumo de energía y su capacidad para manejar comunicaciones de hasta 2 Mbps.

Una de las limitaciones del nRF24L01 es su capacidad para manejar paquetes de datos de hasta 32 bits. Esto significa que los datos transmitidos deben ser empaquetados para evitar la pérdida o corrupción de la información.

12.6.2. Estación en Tierra

La estación en tierra es un circuito igual que el controlador de vuelo que permite la transmisión y recepción de datos entre la aeronave y el equipo en tierra. Esta comunicación se realiza utilizando el módulo nRF24L01, previamente descrito.

La estación en tierra recibe los datos transmitidos por el controlador de vuelo, una vez interpretados, envía un mensaje de confirmación de recepción al controlador de vuelo. Este proceso bidireccional asegura que los datos de telemetría sean correctamente.

El funcionamiento de la comunicación se puede observar en la figura ??, donde se muestra el flujo de mensajes entre la estación en tierra y el controlador de vuelo:

- ◊ **Transmisión:** El controlador de vuelo envía datos a la estación en tierra codificados.
 - ◊ **Recepción:** La estación en tierra recibe los datos y los decodifica.
 - ◊ **Mensaje de Verificación:** La estación en tierra envía un mensaje de verificación de recepción.
 - ◊ **Confirmación:** El controlador de vuelo recibe la confirmación de que los datos fueron recibidos.
- Este ciclo continuo de transmisión y verificación es esencial para mantener la integridad de los datos y garantizar que cualquier ajuste necesario durante el vuelo se base en información precisa y actualizada.

Telemetría estación en Tierra y controlador de vuelo

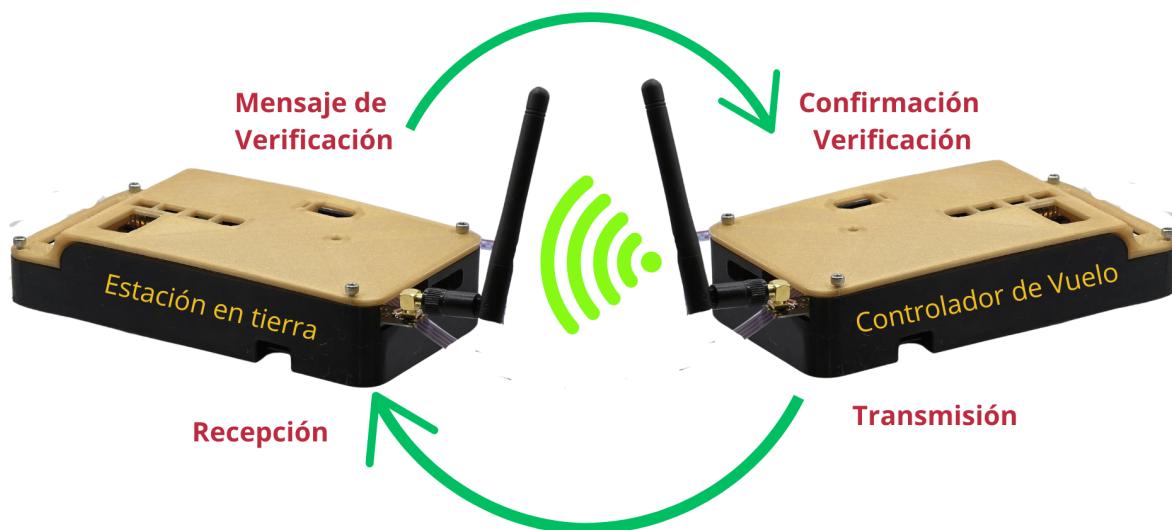


Figura 26: Conexión del Controlador de Vuelo y la estación en Tierra

12.6.3. Arquitectura General del Firmware Controlador de Vuelo y la Estación de Control en Tierra
 La figura ?? representa la arquitectura general del controlador de vuelo y la estación de control en tierra, detallando los protocolos de inicialización y las principales tareas realizadas por cada sistema. A continuación se presenta una explicación general de cada sección:

12.7 Controlador de Vuelo

- ◊ **Protocolo de Inicialización**

- **Escaneo e Inicialización de Dispositivos I2C:** El sistema escanea e inicializa todos los dispositivos I2C conectados al controlador de vuelo.
- **Inicialización del Protocolo SPI y Creación del Archivo de Datos:** Se inicializa el protocolo SPI y se crea un archivo de datos para almacenar los datos de vuelo.
- ◊ **Bucle Principal (Mientras Está Encendido)**
 - **Lectura de Sensores y GPS, Actualización de Variables Globales y Pantalla OLED:** El controlador de vuelo lee datos de varios sensores y del módulo GPS, actualiza las variables globales y la pantalla OLED.
 - **Lectura de Canales del Receptor:** El sistema lee las entradas de los canales del receptor, que contienen los comandos de control de la estación de control en tierra o del operador.
 - **Escritura de Datos en la MicroSD:** Los datos de vuelo se escriben en una tarjeta MicroSD para su registro y análisis posterior.
 - **Envío de Datos (Telemetría):** Los datos de telemetría comprimidos en 32 bits se envían a la estación de control en tierra.
- ◊ **Tarea: Escritura y Movimiento de Actuadores**
 - **Determinación del Modo de Operación Basado en Comandos y Datos de Vuelo:** El controlador de vuelo determina el modo de operación del UAV basado en los comandos de entrada y los datos de vuelo actuales.
 - **Mover Actuadores:** El sistema envía comandos a los actuadores para controlar los movimientos del UAV.

12.8 Estación de Control en Tierra

- ◊ **Protocolo de Inicialización**
 - **Escaneo e Inicialización de Dispositivos I2C:** Similar al controlador de vuelo, la estación de control en tierra escanea e inicializa todos los dispositivos I2C conectados.
 - **Inicialización del Protocolo SPI y Creación del Archivo de Datos:** Se inicializa el protocolo SPI y se crea un archivo de datos para almacenar los datos recibidos.
- ◊ **Bucle Principal (Mientras Está Encendido)**
 - **Lectura de Información de Radio:** La estación de control en tierra lee los datos recibidos vía radio desde el controlador de vuelo.
 - **Decodificación de la Información:** Los datos recibidos se decodifican para su procesamiento.
 - **Escritura de Datos en Serial:** Los datos decodificados se escriben en la interfaz serial para su monitoreo en tiempo real o procesamiento adicional.
 - **Escritura de Datos en la MicroSD:** Los datos recibidos y decodificados también se registran en una tarjeta MicroSD para su análisis posterior.

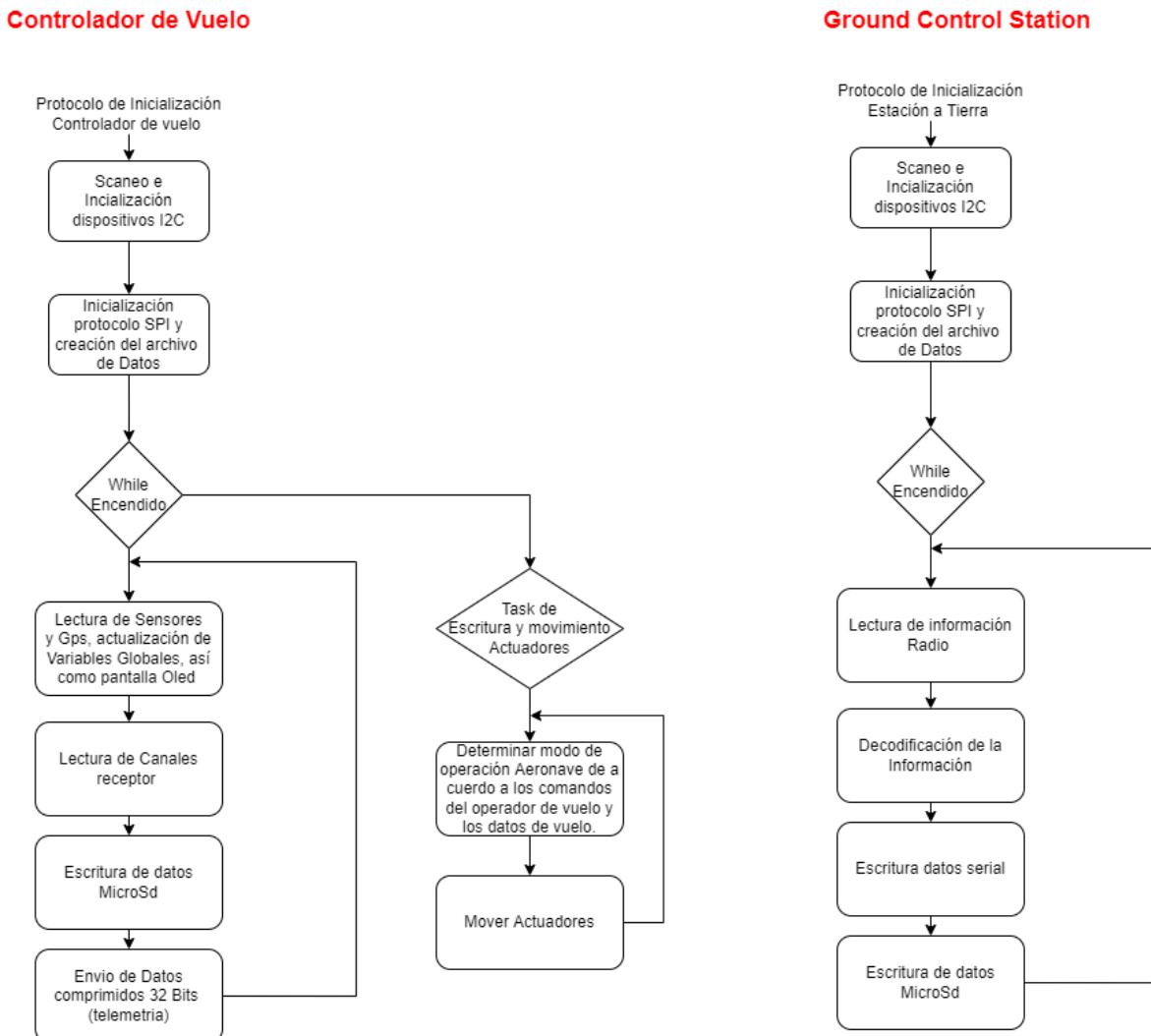


Figura 27: Diagrama general de funcionamiento Controlador y estación a Tierra

12.9 Actuadores

En términos de actuadores, se trabajará con servomotores, ya que son los encargados de mover las superficies de control de la aeronave, incluyendo los alerones, timón de cola, elevadores y el sistema de propulsión eléctrico o de gasolina.

Para manejar estos distintos actuadores, se seleccionó el módulo **PCA9685** debido a su capacidad de controlar hasta 16 servomotores o actuadores simultáneamente mediante señales **PWM**. Este módulo ofrece una gestión precisa y eficiente de los componentes mecánicos de la aeronave. La elección del **PCA9685** permite una optimización del diseño y la funcionalidad del UAV, facilitando la implementación de sistemas de control complejos con una sola placa. Esto reduce la necesidad de múltiples controladores y simplifica la arquitectura del sistema.

12.9.1. Funcionamiento

A la hora de leer los valores del radiotransmisor FlySky, es importante recordar que estamos mapeando la señal que devuelve en un valor angular, lo que permite al controlador determinar el ángulo al que debe fijar el servomotor. Para mover los servomotores, se debe convertir ese ángulo en un ancho de pulso PWM válido. El proceso se describe en el siguiente código:

```

1 #define MIN_PULSE_WIDTH 600
2 #define MAX_PULSE_WIDTH 2600
3
4 int ReguladorServos::pulseWidth(int angle)
5 {
6     // Función que convierte un ángulo entre 0 y 180 a un ancho de pulso (dato que leen los
7     // servos)
8     int pulse_wide, analog_value;
9     pulse_wide = map(angle, 0, 180, MIN_PULSE_WIDTH, MAX_PULSE_WIDTH);
10    analog_value = int(float(pulse_wide) / 1000000 * FREQUENCY * 4096);
11    return analog_value;
12 }
```

Listing 1: Conversión de ángulo a ancho de pulso PWM

Definición de los límites del ancho de pulso:

- ◊ `#define MIN_PULSE_WIDTH 600`: Define el ancho de pulso mínimo en microsegundos.
- ◊ `#define MAX_PULSE_WIDTH 2600`: Define el ancho de pulso máximo en microsegundos.

Función de conversión de ángulo a ancho de pulso:

Definición de los límites del ancho de pulso:

- ◊ `#define MIN_PULSE_WIDTH 600`: Define el ancho de pulso mínimo en microsegundos.
- ◊ `#define MAX_PULSE_WIDTH 2600`: Define el ancho de pulso máximo en microsegundos.

Función de conversión de ángulo a ancho de pulso:

- ◊ La función `pulseWidth(int angle)` toma un ángulo como parámetro, que debe estar en el rango de 0 a 180 grados.
- ◊ Se utiliza la función `map()` para convertir el ángulo en un ancho de pulso proporcional entre los valores mínimos y máximos definidos.
- ◊ La conversión del ancho de pulso a un valor analógico adecuado para el servomotor se realiza mediante el cálculo: `analog_value = int(float(pulse_wide) / 1000000 * FREQUENCY * 4096);`.
- ◊ La función devuelve este valor analógico, que representa el ancho de pulso PWM requerido para mover el servomotor al ángulo deseado.

Este proceso asegura que los ángulos mapeados desde la señal del radiotransmisor se conviertan correctamente en pulsos PWM que los servomotores puedan interpretar.

12.9.2. Movimiento Superficies de Control

Una vez interpretadas las señales del radiocontrol por el receptor y decodificados los ángulos deseados por el operador de vuelo, se asignan valores de movimiento a los servomotores correspondientes a las superficies de control de la aeronave. El canal 1 del receptor controla los ailerones, ilustrados en la figura ?? en color morado, que manejan el roll (balanceo lateral). El canal 2 controla los elevadores , en color negro, encargados del pitch (inclinación longitudinal). El canal 3 está conectado al acelerador, regulando la potencia del motor, mientras que el canal 4 controla el timón de cola para el yaw (giro horizontal), en color verde.

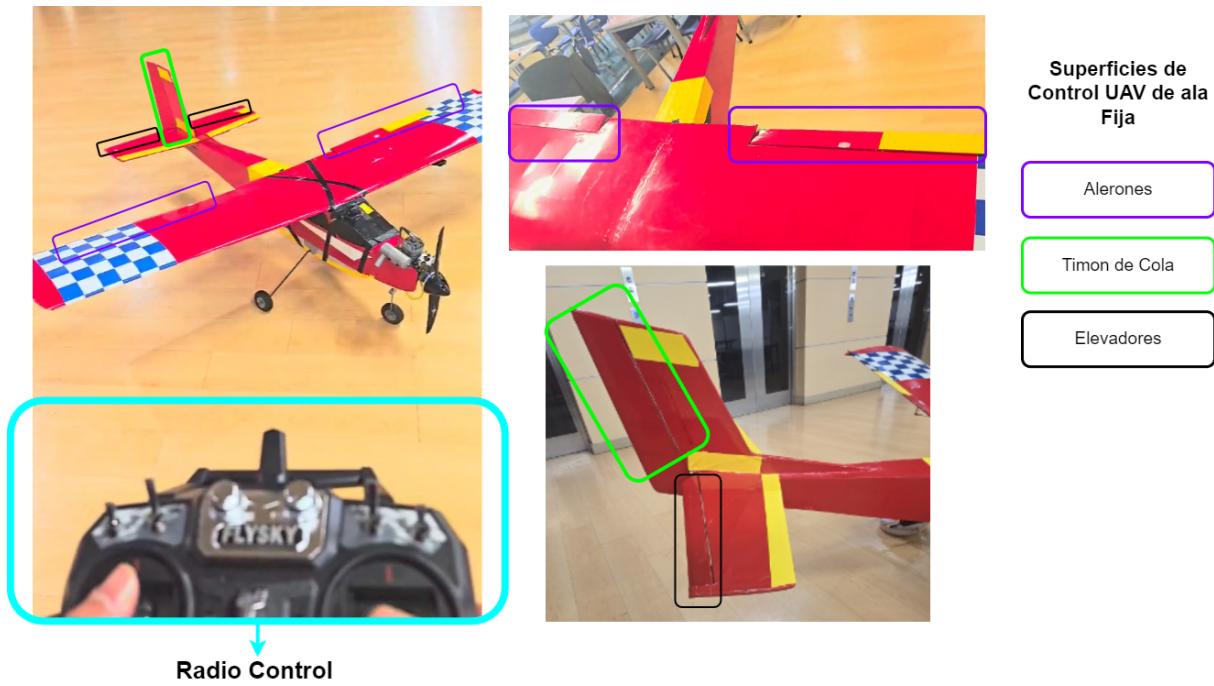


Figura 28: Movimiento general Superficies de Control

12.10 Control

Para el control básico de la aeronave se estableció el siguiente modelo, en el que el operador de vuelo proporciona una señal de referencia para orientar la aeronave. Esta señal de referencia es un valor angular que indica la posición deseada. Este diagrama de PID está simplificado, ya que cada superficie de control tiene su propio PID para alcanzar la señal de referencia (véase imagen ??). Seguidamente, el procesador interpreta la señal de referencia que el operador de vuelo desea establecer y mueve la superficie de control correspondiente a cada señal dada por el operador. Este sistema tiene en cuenta las perturbaciones ambientales y se retroalimenta con los valores obtenidos por las unidades de movimiento iniciales (IMUs).

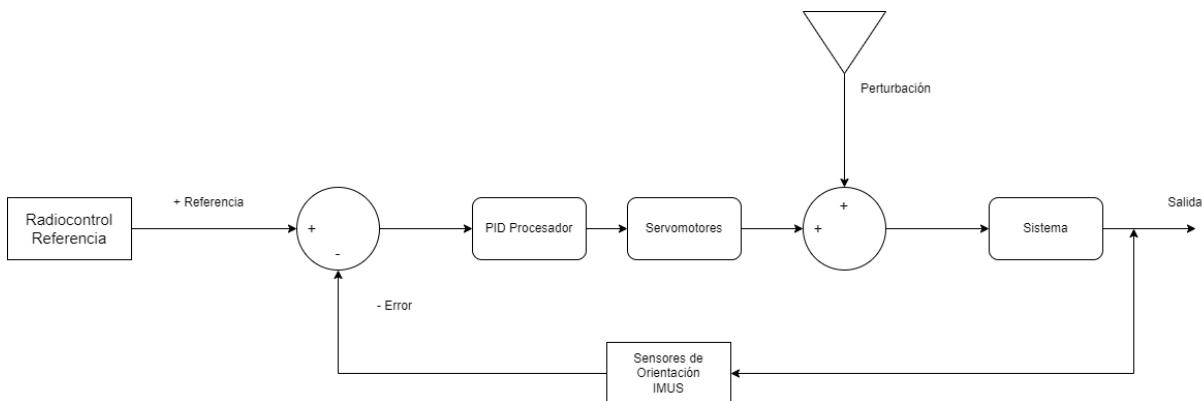


Figura 29: PID básico Aeronave

12.10.1. Modos de Operación

- ◊ **Modo de Vuelo de Autoestabilización:** Este es el modo de operación por defecto. En este modo, si el operador de vuelo no proporciona ninguna señal de referencia, por ejemplo, si el valor de referencia del

roll es 0, el UAV de ala fija moverá los alerones para autoestabilizarse y alcanzar este valor de referencia. Este mismo principio se aplica a los ángulos de yaw y pitch, asegurando que la aeronave mantenga su orientación estable.

- ◊ **Modo de Operación Fly-by-Wire:** En este modo, el operador de vuelo establece un ángulo de referencia específico para la orientación de la aeronave. Este modo de operación se activa al encender el switch del canal 5. En este modo, el comportamiento del transmisor del radiocontrol es diferente: en lugar de enviar una señal para que un servomotor gire a un ángulo deseado, el operador envía un ángulo de referencia al que desea que esté orientada la aeronave. Los valores de los ángulos están limitados a:

Roll: -25° a 25°, puesto que es el ángulo de banqueo máximo permitido para un vehículo aéreo no tripulado.

Pitch: -20° a 20°, puesto que es el ángulo de elevación en el que se evita la perdida de sustentación "stall".

12.11 Indicadores Adicionales

12.11.1. Dispositivo de visualización

El sistema de control del UAV incluye un dispositivo de visualización que muestra en tiempo real indicadores cruciales como brújula, temperatura, pitch, roll, altitud, yaw, latitud y longitud. Esto permite al operador verificar el funcionamiento adecuado del UAV durante las pruebas de vuelo, asegurando que los parámetros críticos se mantengan dentro de los límites operativos. El dispositivo facilita el monitoreo continuo, la detección rápida de problemas y la validación de que todos los sistemas responden correctamente.

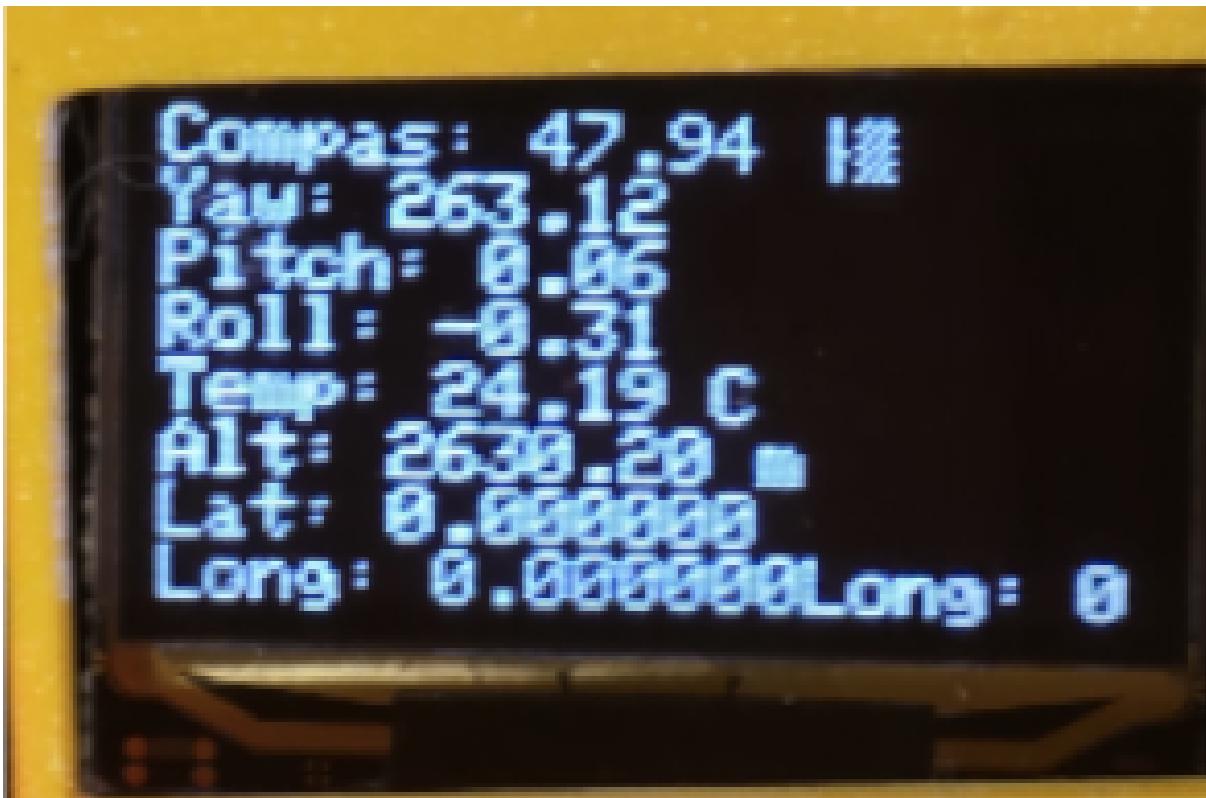


Figura 30: PID básico Aeronave

12.11.2. Buzzer

Un **Buzzer** o **zumbador** es un componente que se utiliza para emitir señales acústicas. En este caso, se utiliza para proporcionar alertas auditivas en diferentes situaciones. Como lo son la emisión de sonido para informar que el dispositivo se ha inicializado o que se ha conectado el GPS u otros módulos, también puede proporcionar alertas en caso tal de que la batería se esté descargando. El **Buzzer** se conecta al **ESP32-S3** a través de Un pin **GPIO**, particularmente al **GPIO-46** configurado como salida digital. Cuando el microcontrolador envía una señal de alto nivel, el **Buzzer** emite un sonido en particular alertando al usuario del estado general del sistema.

13 Interfaz

La interfaz de usuario desarrollada para el UAV de ala fija incluye varios componentes que permiten una visualización integral de los parámetros de vuelo y el estado de la aeronave en tiempo real. Los parámetros considerados más cruciales se han incorporado en diferentes secciones de la interfaz, facilitando el monitoreo y el control efectivo del UAV. A continuación se describen estos componentes:

13.1 Mapa de Posición en Tiempo Real

En la figura ?? se puede observar un apartado en el que se observa la posición del UAV en tiempo real utilizando un mapa interactivo. La posición actual se muestra con coordenadas de latitud y longitud, permitiendo un seguimiento preciso del UAV durante su vuelo.

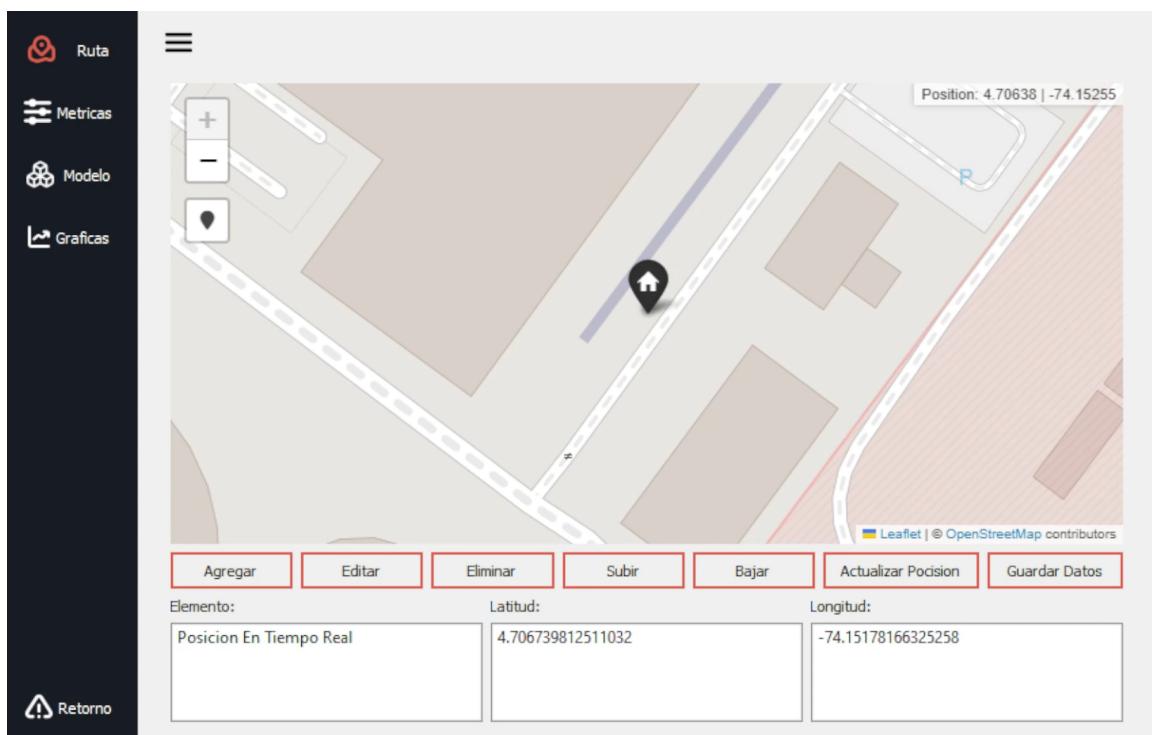


Figura 31: Mapa de Visualización UAV

13.2 Indicadores de Vuelo:

Los indicadores de vuelo se presentan mediante instrumentos de control similares a los que se encuentran en una aeronave real (véase ??). Estos instrumentos incluyen:

- ◊ **Horizonte Artificial:** Indica la actitud del avión en términos de yaw, pitch y roll.
- ◊ **Altímetro:** Mide la altitud del UAV con referencia a la altitud inicial del dispositivo.
- ◊ **Barómetro:** Indica la presión atmosférica.
- ◊ **Airspeed:** Mide la velocidad del aire.
- ◊ **Manifold Pressure:** Muestra la presión del colector, relevante para el rendimiento del motor.

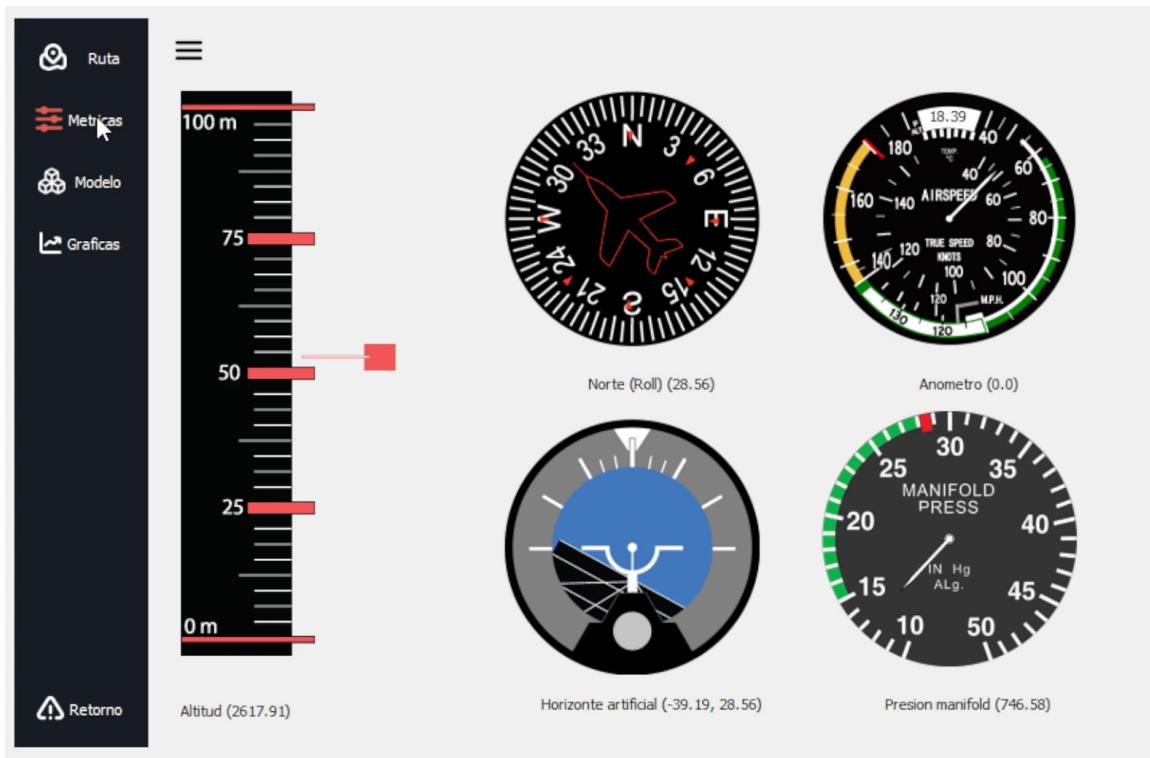


Figura 32: indicadores de vuelo para un UAV

13.3 Modelo 3D del UAV:

Se incluye un modelo tridimensional del UAV que representa su orientación en el espacio ???. Proporcionando una visualización clara de la dirección hacia donde está apuntando el UAV, facilitando la comprensión de su comportamiento dinámico.

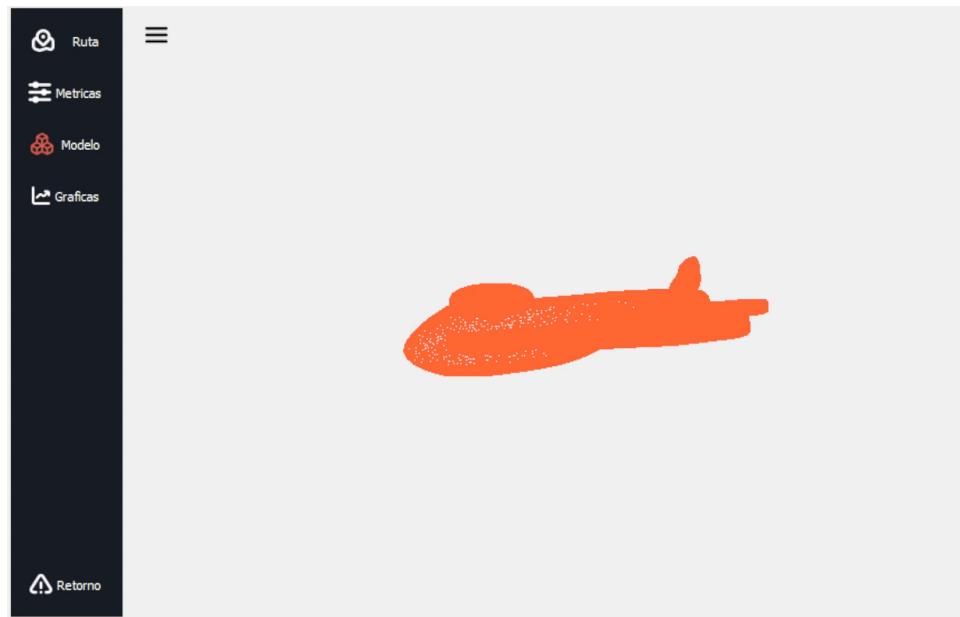


Figura 33: Modelo 3d UAV en la Interfaz

13.3.1. Gráficas en Tiempo Real:

Esta sección permite visualizar diferentes comportamientos históricos de la aeronave. Las gráficas muestran datos como movimientos inerciales (roll, pitch, yaw), altitud, temperatura, presión, y velocidad en función del tiempo. Estas gráficas son cruciales para analizar y evaluar el rendimiento del UAV durante vuelo para observar y analizar el comportamiento histórico de la aeronave.

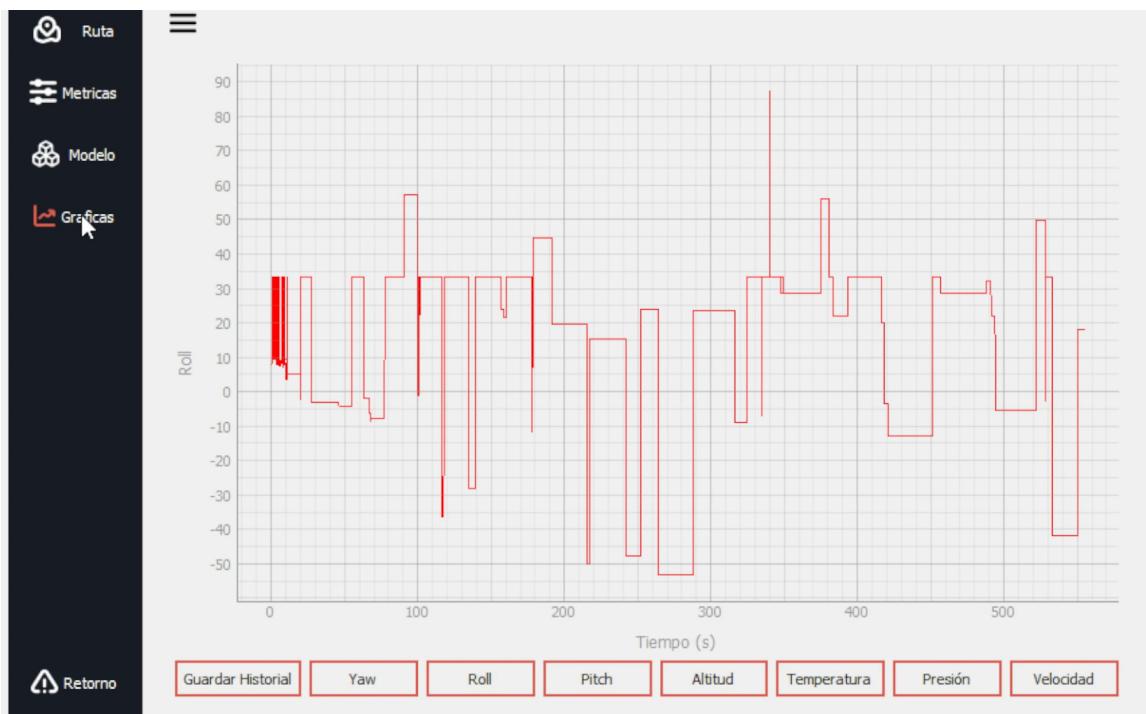


Figura 34: Gráficas de funcionamiento UAV

14 Costos del Sistema

Finalmente se hace necesario resaltar los costos asociados al desarrollo del controlador estos se encuentran en el apartado de Anexos (véase ??). Se resalta que el costo de los componentes electrónicos y mecánicos fue de alrededor de los **\$2'000.000**. Cabe resaltar que el desarrollo del controlador incluyendo la etapa de diseño electrónico, diseño mecánico, software, firmware y pruebas llevo alrededor de 350 horas ocasionando que el costo de desarrollo del controlador de vuelo fuera de cerca de los **\$6'4000.0000** pesos teniendo en cuenta el salario de un ingeniero electrónico promedio en Colombia que es de **\$12.664** por hora .

Item	Imagen/Descripción	Cantidad	Pesos
Radio control Flysky FS-i6 actualizado a 10 canales con receptor		1	\$ 289,995.00
Adafruit 9-DOF orientación absoluta IMU fuga de mezcla - BNO055.		1	\$ 143,635.00
MATEK M10Q-5883 GNSS y brújula soporta GNSS (GPS, GLONASS, Galileo y BeiDou), ofrece una gran sensibilidad de RF para aviones drones		1	\$ 168,458.60
ESP32-S3-DevKit CN16R8		4	\$ 205,672.00
BMP280		2	\$ 15,000.00
Nrf24l01		3	\$ 600,000.00
Módulo Mpu6050		2	\$ 27,000.00
Filamento de Impresión 3d		1	\$ 80,000.00
Tornillos, Conectores,Inserts		1	\$ 20,000.00
Bateria		1	\$ 400,000.00
Horas Trabajadas	El salario ingeniero electronico promedio en Colombia es de \$27.657.404 al año o \$12.664 por hora.	350	\$ 4,432,400.00
Total Sin Impuestos	\$ 6,382,160.60	Total Sin Horas Trabajadas	\$ 1,949,760.60

Figura 35: Costos desarrollo Controlador

15 Pruebas de Funcionamiento

15.1 Introducción

En esta sección se describen las pruebas realizadas para validar el funcionamiento del controlador de vuelo en conjunto. Primeramente se van a describir ciertas pruebas que se realizaron a cada sección de componentes, verificando su funcionamiento individual y la estabilidad del sistema bajo condiciones operativas.

15.2 Prueba de Almacenamiento de Datos

Para este apartado de prueba de datos se utilizo el modulo de Almacenamiento de Datos que esta situado en la parte inferior del dispositivo tal y como se observa en la imagen , seguidamente se procedió a configurar la **ESP32-S3** para poder escribir y guardar datos en una unidad **microSD**. Para realizar esto, debemos recordar que la comunicación con el modulo de almacenamiento de datos se realiza mediante el protocolo **SPI** por lo que se debe instanciar los pines de **MISO, MOSI, SCK, CS** de este protocolo con los siguientes **GPIOs** :

```

1 #define MOSI_PIN 35
2 #define SCK_PIN 36
3 #define MISO_PIN 37
4 #define CS_PIN 38

```

Posteriormente, se procedió a realizar un script que fuera capaz de leer la cantidad de archivos en la unidad de almacenamiento y ponerle una cabecera de nombre que fuera **dataSaved_n.csv** donde **n** es la cantidad de archivos que tiene la **microSD**, de tal forma que el nombre fuese único cada vez que se reinicia el dispositivo. Esto se realizo de esta forma para evitar estar sobrescribiendo información en un mismo archivo. Adicionalmente, la información que se va a guardar en la microSD, es la información crucial de todos los datos de vuelo que el UAV ha generado desde su puesta de operación hasta el momento en el que se recupere esta información.

Para ver la capacidad de recolección de datos del dispositivos y el tamaño del archivo generado .Se procedió

a dejar simulando el dispositivo durante 3 horas y 20 minutos, generando un archivo CSV con un tamaño de 1,18 MB (1.245.184 bytes). El archivo contiene un total de 17.031 líneas de datos.

Para estimar la capacidad de almacenamiento a largo plazo, realizamos una proyección utilizando una memoria de 64 GB. A continuación, se muestra el cálculo detallado:

- ◊ Tamaño del archivo generado: 1,18 MB (1.245.184 bytes)
- ◊ Duración de la prueba: 3 horas y 20 minutos (3,33 horas)
- ◊ Número de líneas en el archivo: 17.031

Para determinar cuántas horas de operación se pueden almacenar en una memoria de 64 GB (68.719.476.736 bytes), utilizamos la siguiente regla de tres:

$$\frac{1.245.184 \text{ bytes}}{3,33 \text{ horas}} = \frac{68.719.476.736 \text{ bytes}}{x \text{ horas}} \quad (3)$$

Despejando x :

$$x = \frac{68.719.476.736 \text{ bytes} \times 3,33 \text{ horas}}{1.245.184 \text{ bytes}} = 183.816,34 \text{ horas} \quad (4)$$

Para convertir las horas de operación a años, usamos la relación de 1 año = 8.760 horas:

$$\frac{183.816,34 \text{ horas}}{8.760 \text{ horas/año}} = 20,99 \text{ años} \quad (5)$$

Por lo tanto, se puede almacenar información en una memoria de 64 GB durante aproximadamente 183.816,34 horas o 20,99 años, considerando el ritmo de generación de datos y el tamaño de los archivos registrados en esta prueba.

15.3 Verificación de la lectura y escritura del valor de los servos con osciloscopio

Una vez configurado el apartado de lectura de los canales del transmisor **FLYSKY-I6**, se procedió a verificar que la lectura de los canales y la escritura de los valores **PWM** en la **PCA9685** se realizaran correctamente. Para ello, se utilizó un osciloscopio para visualizar tanto la señal de entrada como la de salida, como se muestra en la figura (??).



Figura 36: Pruebas de lectura y escritura valores PWM

La prueba del osciloscopio se realizó midiendo directamente en el pin de salida del receptor del canal 1 y en el pin de salida de la **PCA9685**. Los resultados mostraron que el cambio en la modulación por ancho de pulso (PWM) es proporcional al ángulo deseado, lo cual se puede evidenciar tanto en la lectura como en la escritura de los valores PWM.

En la figura (??) se presentan tres capturas de pantalla del osciloscopio, que muestran las señales PWM correspondientes a ángulos de 0°, 90° y 180°.

- ◊ En la primera imagen, se observa una señal PWM con un ángulo de 0°. La duración del pulso es de 940 μ s, lo que indica la posición mínima posible.
- ◊ En la segunda imagen, la señal PWM corresponde a un ángulo de 90°. La duración del pulso es de 1.48 ms, indicando la posición por defecto o media del control.
- ◊ En la tercera imagen, la señal PWM corresponde a un ángulo de 180°. La duración del pulso es de 1.92 ms, indicando la posición máxima posible.

Adicionalmente, es importante señalar que en las imágenes la línea verde representa la señal del receptor, mientras que la línea amarilla representa la señal de la **PCA9685**. Estos resultados confirman que la modulación por ancho de pulso (PWM) cambia proporcionalmente con el ángulo deseado, garantizando un control preciso y predecible de los servomecanismos o actuadores conectados al sistema.

15.4 Comparación de Controladores

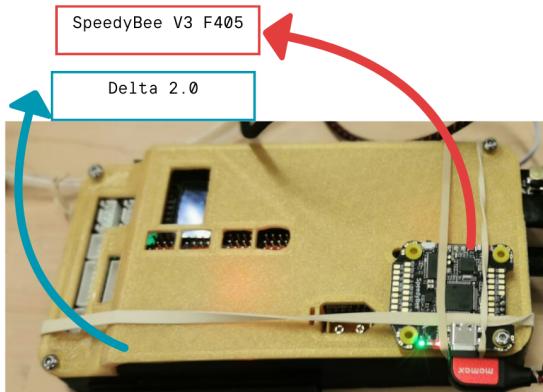
Para esta prueba, se compararon los valores de los ángulos de Roll y Pitch entre el controlador diseñado y un controlador comercial, el SpeedyBee V3 F405. Los dos controladores se situaron como se observa en la imagen y se configuraron los offsets iniciales para asegurar una medición precisa desde el mismo punto de referencia.

Para obtener los valores del controlador SpeedyBee, se utilizó la aplicación de Betaflight, la cual permite obtener los valores de Yaw, Pitch y Roll. Se contrastaron las mediciones cada diez grados de los ángulos y se puede observar en las imágenes que el comportamiento de ambos controladores es bastante similar.

15.4.1. Descripción del SpeedyBee V3 F405

El SpeedyBee V3 F405 es un controlador de vuelo comercial diseñado para drones. Este controlador es conocido por su estabilidad y precisión en la medición de ángulos, y es ampliamente utilizado en la comunidad de drones por su integración con la aplicación Betaflight, que facilita la configuración y obtención de datos de vuelo.

Montaje de Pruebas de validación Angulos de Giro



Montaje Controladores



Interfaz Betaflight

Figura 37: Pruebas de validación y montaje controladores

15.4.2. Resultados de la Prueba

Cuadro III: Toma de datos 1 y 2 para Roll

X	Toma de datos 1			Toma de datos 2		
	Roll SpeedyBee v3 f405	Roll Delta	% Error	Roll SpeedyBee v3 f405	Roll Delta 2	% Error
1	0	0,44	1 %	0	0	0
2	10,3	10,44	1 %	10,1	10,38	3 %
3	20,6	20,2	2 %	19,5	20,12	3 %
4	31,2	30,62	2 %	30,7	30,69	0 %
5	41,7	40,19	4 %	40,8	40,38	1 %
6	52,1	50,63	3 %	52	50,25	3 %
7	63,2	60,56	4 %	61,3	60,63	1 %
8	72,8	70,5	3 %	73,2	70,12	4 %
9	83,1	80,56	3 %	84,2	80,44	4 %
10	86,5	87,1	1 %	92,9	88,44	5 %
11	-8,7	-10,38	19 %	-9,6	-10,56	10 %
12	-19,8	-20,31	3 %	-19,1	-20,25	6 %
13	-30	-30,19	1 %	-29,4	-30,25	3 %
14	-40,7	-39,1	4 %	-39,1	-40,13	3 %
15	-50,9	-50,63	1 %	-49,2	-50,88	3 %
16	-59,8	-60,25	1 %	-59,5	-60,75	2 %
17	-69,7	-70,37	1 %	-67,8	-70,19	4 %
18	-80,1	-80,69	1 %	-77,9	-80,81	4 %
19	-89,2	-90,06	1 %	-87,3	-90,06	3 %

Cuadro IV: Toma de datos 1 y 2 para Pitch

X	Toma de datos 1			Toma de datos 2		
	Pitch SpeedyBee v3 f405	Pitch Delta	% Error	Pitch SpeedyBee v3 f405	Pitch Delta 2	% Error
1	0	-0,44	1 %	0,1	0,94	0 %
2	10,8	10,6	2 %	10,8	10,44	3 %
3	21,6	20,12	7 %	19,8	20	1 %
4	31,9	30,44	5 %	30,1	30,31	1 %
5	41,9	40,13	4 %	40,8	40,63	0 %
6	53,2	50,63	5 %	50,1	50,19	0 %
7	63,5	60,96	4 %	60,9	60,19	1 %
8	73,3	70,75	3 %	71,5	70,44	1 %
9	83,3	80,12	4 %	82,5	80,37	3 %
10	86,3	87,06	1 %	87,5	88	1 %
11	-9,6	-10,38	8 %	-11,8	-10,13	14 %
12	-20,5	-20,31	1 %	-21,9	-20,81	5 %
13	-29,6	-30,5	3 %	-32,1	-30,44	5 %
14	-40	-40,31	1 %	-43,6	-40,75	7 %
15	-49,2	-50,19	2 %	-53,5	-50,44	6 %
16	-58,3	-60,13	3 %	-62,6	-60,75	3 %
17	-69	-70,25	2 %	-72,1	-70,5	2 %
18	-78,9	-80,44	2 %	-82,1	-80,62	2 %
19	-87,5	-89,44	2 %	-86,6	-87,19	1 %

15.4.3. Análisis de las Tablas

En las tablas de datos de Roll y Pitch, se observa que ambos controladores presentan comportamientos bastante similares. El error relativo entre las mediciones de los controladores Delta y SpeedyBee V3 F405 varía generalmente entre el 1 % y el 4 %. Sin embargo, hay excepciones notables, como cuando se miden ángulos cercanos a los -10°, tanto en Roll como en Pitch, donde el error puede llegar hasta el 19 %.

Es importante destacar que existe cierta incertidumbre asociada a la forma en que se recopilaron los datos. Fijar los ángulos en diferentes grados puede introducir variaciones en las lecturas de ambos controladores debido a pequeñas discrepancias en la posición exacta y la respuesta de los sensores. Esto puede afectar la precisión de las mediciones comparadas.

15.4.4. Comparación de Roll

En los gráficos de la Figura ??, se puede observar que los ángulos de Roll medidos por ambos controladores son muy similares. En la primera toma de datos, los valores siguen una tendencia casi idéntica, con variaciones mínimas en algunos puntos. En la segunda toma de datos, se observa un comportamiento similar, reafirmando la consistencia entre las mediciones de ambos dispositivos.

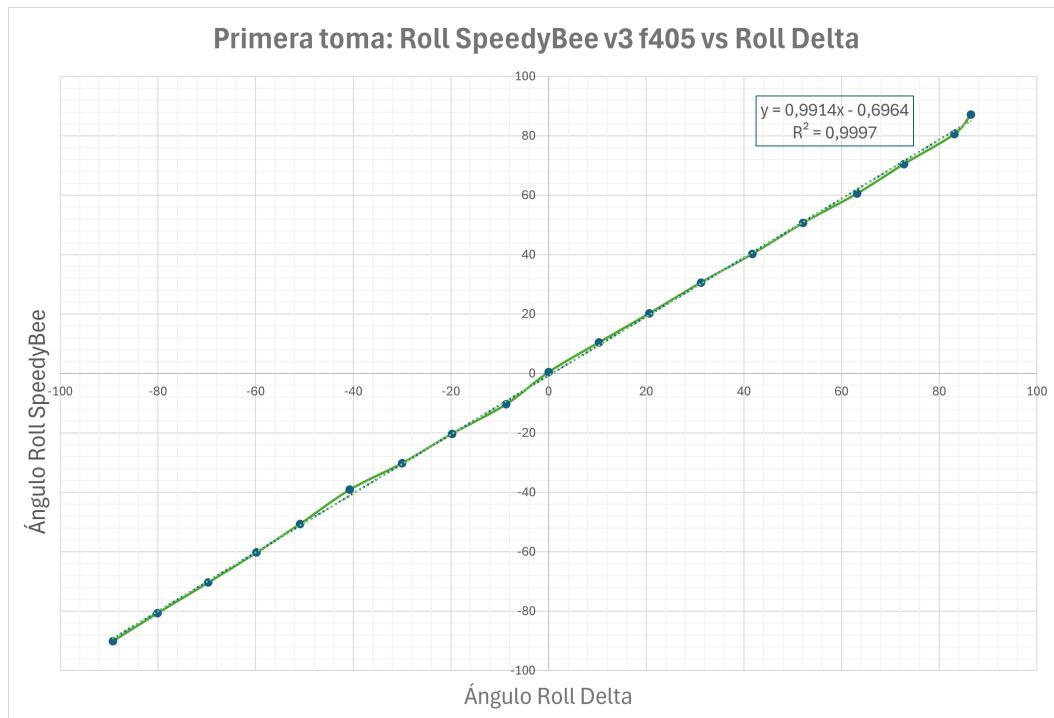


Figura 38: Comparación de ángulo Roll entre controladores Delta y SpeedyBee V3 F405 en la primera toma de datos

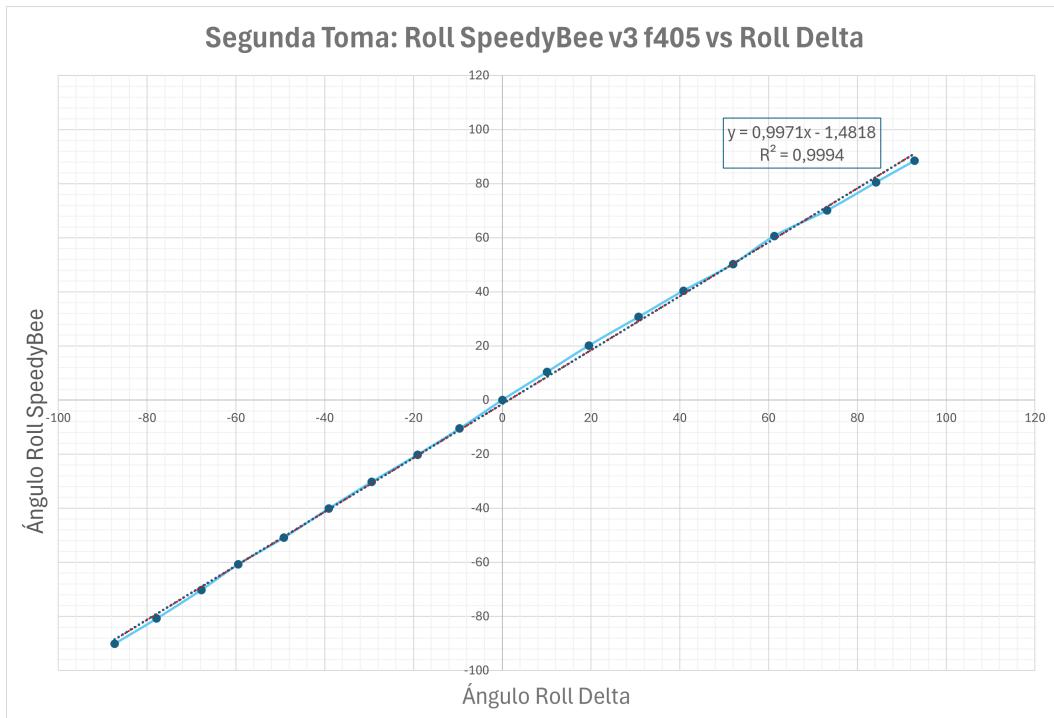


Figura 39: Comparación de Roll entre controladores Delta y SpeedyBee V3 F405 en la segunda toma de datos

15.4.5. Comparación de Pitch

De manera similar, los gráficos de la Figura ?? presentan la comparación de los ángulos de Pitch. Aquí también se evidencia que ambos controladores tienen un comportamiento muy parecido. Las diferencias entre los valores son mínimas y generalmente se mantienen dentro de un margen de error aceptable.

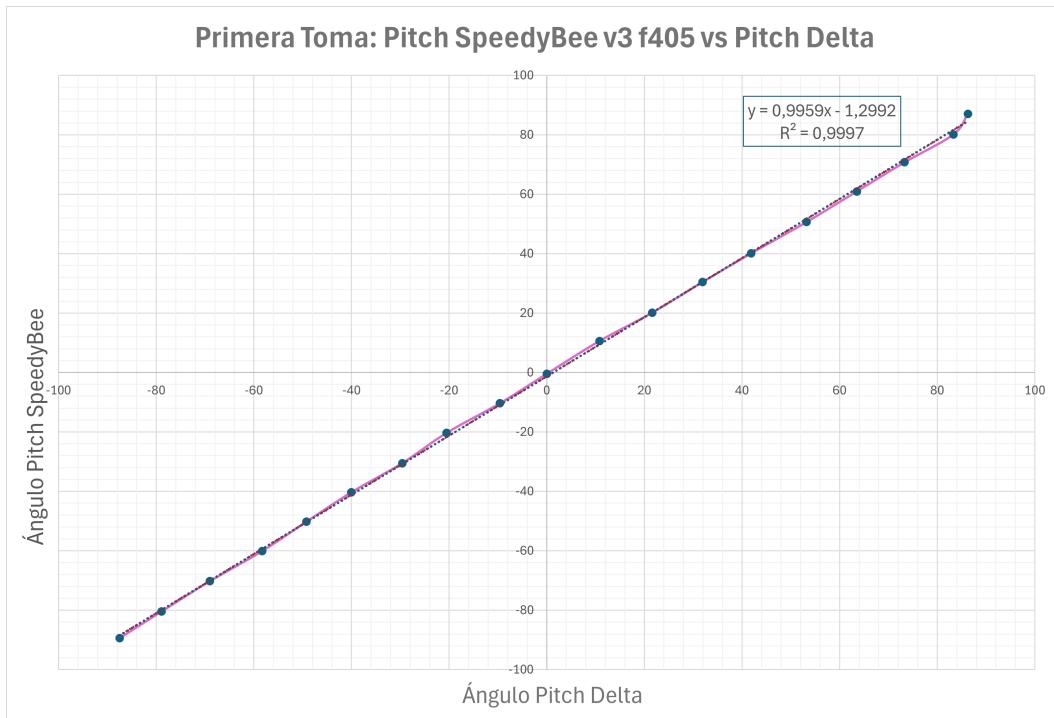


Figura 40: Comparación de Pitch entre controladores Delta y SpeedyBee V3 F405 en la primera toma de datos

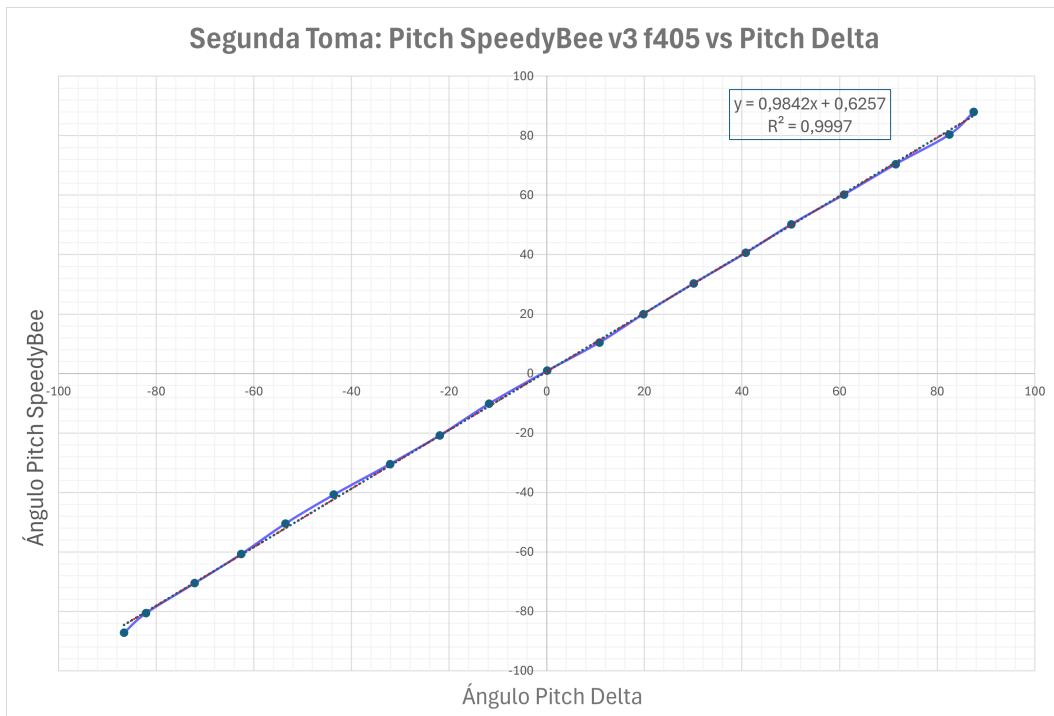


Figura 41: Comparación de Roll entre controladores Delta y SpeedyBee V3 F405 en la segunda toma de datos

15.5 Pruebas de Variación de Altura con el Barómetro BMP280

Para evaluar el desempeño del sensor barométrico BMP280 en la medición de la altura, se realizó una prueba en el Edificio Mario Laserna. Durante esta prueba, se utilizó un ascensor para cambiar de piso en diferentes niveles del edificio. La prueba se inició en el nivel S2 y se ascendió hasta el piso 8. Finalmente, se descendió al nivel inicial y se capturaron los datos correspondientes a la variación de altura.

15.5.1. Procedimiento de la Prueba

1. **Inicio en el Nivel S2:** La prueba comenzó en el nivel subterráneo 2 del edificio.
2. **Ascenso al Piso 8:** Se utilizó el ascensor para ascender secuencialmente desde el nivel S2 hasta el piso 8, realizando paradas en cada piso para registrar los datos de altura.
3. **Descenso al Nivel Inicial:** Posteriormente, se descendió de nuevo hasta el nivel S2, registrando los datos durante todo el recorrido.

15.5.2. Resultados

La figura adjunta muestra los resultados obtenidos de la prueba de variación de altura. Se puede observar cómo la altura medida por el BMP280 cambia a medida que el sensor se mueve a diferentes niveles del edificio. Los datos capturados reflejan un aumento progresivo de la altura durante el ascenso y una disminución correspondiente durante el descenso.

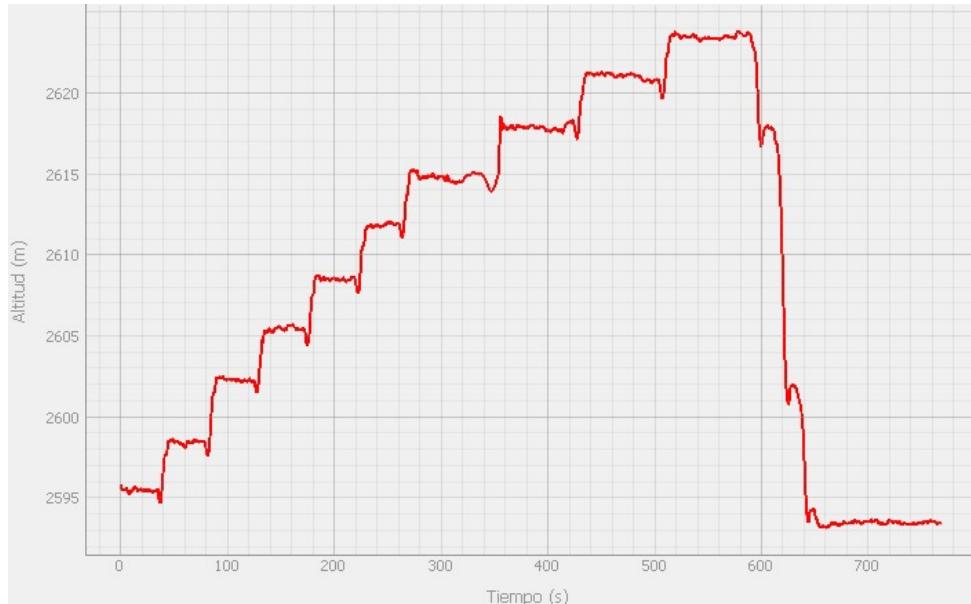


Figura 42: Variación de altura medida por el BMP280 durante el ascenso y descenso en el Edificio Mario Laserna.

15.5.3. Análisis de las Alturas del Edificio

La prueba de variación de altura realizada en el Edificio Mario Laserna nos permitió obtener datos precisos sobre la altitud a diferentes niveles del edificio. Según los datos registrados, la altura en el nivel S2 era de aproximadamente 2595 metros sobre el nivel del mar (msnm), mientras que en el piso 8 la altura era de aproximadamente 2625 metros msnm.

Para calcular la diferencia de altura entre estos dos niveles, se realiza la siguiente operación:

$$\text{Diferencia de altura} = 2625 \text{ msnm} - 2595 \text{ msnm} = 30 \text{ metros}$$

Por lo tanto, la altura total del edificio, desde el nivel S2 hasta el piso 8, es de 30 metros.

Estos resultados demuestran la capacidad del sensor BMP280 para medir con precisión las variaciones de altura en un entorno controlado como el de un edificio de varios pisos.

15.6 Pruebas de GPS

Se realizó una prueba de GPS en el espacio abierto conocido como **El Bobo** en la Universidad de los Andes. Durante esta prueba, los datos de ubicación fueron almacenados en la tarjeta SD del dispositivo para su posterior análisis y ploteo en un mapa. Tal y como se muestra en la imagen adjunta, se realizó un desplazamiento en forma de bucle para verificar la precisión del GPS.

El recorrido se realizó alrededor de la zona verde, registrando puntos de ubicación en intervalos regulares. Los datos capturados fueron luego representados en un mapa, mostrando el trayecto seguido. Esta prueba permitió evaluar la precisión del GPS en un entorno controlado y abierto, confirmando su capacidad para rastrear movimientos en tiempo real y proporcionar datos de ubicación precisos.



Figura 43: Recorrido de prueba de GPS en **El Bobo**, Universidad de los Andes.

16 Prueba de vuelo

Se realizó una prueba de vuelo para comprobar el funcionamiento del controlador Delta 2.0 en un UAV de ala fija. Esta prueba se realizó en la Pista de Aeromodelismo Vuelo Libre, en Tenjo, Cundinamarca, Colombia. El controlador de vuelo fue instalado en el fuselaje de un avión tipo Protégé de Carl Goldberg, justo debajo de las alas. Esta ubicación estratégica permite un fácil acceso y brinda la protección adecuada para los componentes electrónicos durante el vuelo. (veasé ??)

- ◊ **Imagen Izquierda:** La imagen muestra la instalación del controlador de vuelo dentro del fuselaje del avión. Se puede ver que el controlador está firmemente sujetado justo debajo de las alas. Esta posición facilita la conexión con otros componentes del sistema de vuelo y asegura que el controlador permanezca estable durante el vuelo.
- ◊ **Imagen Derecha:** La imagen muestra el UAV-RAS Uniandes tipo Protégé completamente ensamblado y listo para volar. Se puede apreciar cómo el controlador de vuelo está integrado de manera discreta en el fuselaje, sin interferir con la aerodinámica del avión.

Sujeción Controlador de Vuelo al Protégé



Figura 44: Instalación del Controlador de Vuelo Protégé

16.1 Metodología de la Prueba

En esta prueba se tiene una comunicación bidireccional entre la tarjeta en base y la tarjeta en vuelo. La tarjeta en base es conectada mediante un cable tipo C de transmisión de datos a un ordenador que despliega la interfaz, tal como se muestra en la figura ??.

Las pruebas que se llevaron a cabo se dividen en cinco categorías principales: Sensórica, Actuadores, GPS, Comunicación, Almacenamiento de Datos e Interfaz. Las pruebas se llevaron a cabo en tres fases: en tierra, durante el montaje en el UAV y en vuelo.

En la categoría de sensórica, las lecturas del giroscopio (Yaw, Pitch y Roll) fueron precisas en todas las fases, obteniendo una calificación del 100 %. Sin embargo, en las pruebas del barómetro se presentaron picos abruptos

en las lecturas de presión atmosférica y altura durante el vuelo, lo que no fue consistente con la información de vuelo. Estos cambios abruptos en la recepción de datos resultaron en una calificación del 50 %.

En cuanto a los actuadores, estos funcionaron perfectamente al usar el receptor FlySky. Estos mostraron un funcionamiento óptimo tanto en tierra como en vuelo, obteniendo una calificación del 100 %. Durante las pruebas en el montaje del UAV se presentaron delays en las señales del controlador y los servos, lo que resultaría en una afectación en las dinámicas del UAV. Por otro lado, debido a condiciones climáticas no se pudo realizar una prueba de vuelo con el controlador Delta 2.0 por lo que durante el vuelo con el FlySky se posicionó el controlador de vuelo en el interior del UAV pero sin conectar los servos del UAV a este.

En la categoría de GPS, la calibración y recepción de datos fueron correctas en tierra y durante el montaje en el UAV. Sin embargo, durante el vuelo, el tiempo de actualización del GPS no fue lo suficientemente rápido, lo que llevó a pérdidas de conexión con el satélite y resultó en una calificación del 70 %. Este problema causó saltos abruptos en la localización del UAV.

La Comunicación bidireccional entre la estación en tierra y el UAV fue efectiva en tierra, pero durante el vuelo se presentaron corrupciones de paquetes que afectaron la transmisión y recepción de datos. Esto resultó en calificaciones del 85 % y 30 % para la transmisión de datos del giroscopio y el barómetro, respectivamente. Además, hubo una gran pérdida de datos respecto a los datos transmitidos por la tarjeta en vuelo.

En la categoría de Almacenamiento de Datos, tanto la estación en tierra como el controlador de vuelo fueron capaces de almacenar datos correctamente en tierra. Sin embargo, durante el vuelo, hubo pérdidas ocasionales de datos, resultando en una calificación del 70 % para el controlador de vuelo. Este problema fue causado por la frecuencia de datos transmitidos y la capacidad de almacenamiento del sistema.

Finalmente, en la categoría de Interfaz, el despliegue y la visualización de datos fueron correctos en tierra. Sin embargo, durante el vuelo, se observaron delays y corrupción de datos en la tarjeta de vuelo, resultando en una calificación del 50 %. La tarjeta de la estación en tierra funcionó adecuadamente, pero la tarjeta de vuelo mostró problemas significativos.



Figura 45: Diagrama de la prueba de vuelo

16.2 Matriz de Rendimiento de la Prueba de vuelo

A continuación se muestra la matriz de rendimiento de la prueba de vuelo??:

Pruebas de funcionamiento local						
Etapa		Parámetro	Tierra (%)	Montaje en (%)	Vuelo (%)	Observaciones
Sensórica	Giroscopio	Yaw	100	100	100	
		Pitch	100	100	100	
		Roll	100	100	100	
	Barómetro	Presión Atm	100	100	90	Se presentaron un pico abrupto en los datos obtenidos no consistente con la información de vuelo
		Altura	100	100	100	Al inicio de la prueba hubo un pico abrupto con respecto al origen, sin embargo en el resto de la prueba el valor es adecuado
		Temperatura	100	100	100	Cambios abruptos en la recepción de datos resultando en una corrupción en la información brindada por el barómetro
	FlySky		100	100	100	
Actuadores	Delta 2.0		90	80	0	Delay en las señales del controlador a los servos afectando posiblemente las dinámicas de vuelo. Debido a condiciones climáticas no fue posible realizar una prueba de vuelo controlada por el Delta 2.0
	Calibración GPS		100	100	100	
GPS	Recepción Datos	Latitud	100	100	70	Tiempo de actualización del GPS no fue lo suficientemente rápido por lo que hubieron perdidas de conexión con el satélite. Lo que resulta en saltos abruptos en la localización del UAV
		Longitud	100	100	70	
	Bidireccional		90	90	85	Hubo una corrupción de paquetes lo que afectó los datos de transmisión vs recepción
Comunicación	Bidireccional	Giroscopio	70	60	30	Corrupción de paquetes agravada por la limitación de almacenamiento de la información en bits
		Barometro	80	80	30	Gran perdida de datos respecto a los datos obtenidos por la tarjeta en vuelo
		GPS	100	100	60	
Almacenamiento	Estación Tierra		100	100	60	Redundancia en datos presentada por perdidas en conexión
	Controlador Vuelo		100	100	80	Perdida de datos ocasionada por la frecuencia de almacenamiento
Interfaz	Despliegue		100	100	70	Debido a la gran cantidad de datos recibidos se presenta una sobrecarga en el buffer de datos provocando un lag en la interfaz
	Visualización		90	90	60	
	Almacenamiento de datos		70	70	40	Propagación de error debido a los datos corruptos de recepción
	Tarjeta en Tierra		100	100	50	Delays y problemas registrados durante la
	Tarjeta en Vuelo		100	100	NA	No se puede conectar la tarjeta en vuelo a interfaz durante la prueba

Figura 46: Matriz de evaluación para prueba de vuelo

El desempeño general de las pruebas evidencia un comportamiento sólido en tierra y durante el montaje en el UAV. Por otro lado, se evidencian algunas dificultades notables durante las pruebas de vuelo, especialmente en

las áreas de comunicación bidireccional, almacenamiento de datos y la interfaz de usuario. Estas áreas requieren una revisión y optimización para asegurar un rendimiento óptimo en condiciones de vuelo. A continuación se muestra el rendimiento general de la prueba de vuelo??:

Prueba Global	Resumen de prueba		
	Calificación (%)		
	Tierra (%)	Montaje en UAV (%)	Vuelo (%)
Sensórica	100.00	100.00	98.33
Actuadores	96.67	90.00	50.00
GPS	100.00	100.00	80.00
Comunicación	80.00	76.67	48.33
Almacenamiento de Datos	100.00	100.00	70.00
Interfaz	92.00	92.00	55.00
PROMEDIO GENERAL	94.78	93.11	66.94

Figura 47: Matriz de evaluación para prueba de vuelo

17 Resultados Prueba de Vuelo

17.1 Análisis de Resultados GPS Controlador de Vuelo

A continuación, se presenta los resultados obtenidos de la trayectoria GPS del UAV de ala fija. La figura ?? muestra el recorrido efectuado por el UAV de ala fija durante la prueba de vuelo, la cual fue realizada por un piloto especializado especializado en el manejo de este tipo de aeronaves. La trayectoria registrada refleja la complejidad del vuelo, donde el piloto ejecutó diversas maniobras y movimientos a lo largo de un periodo prolongado. La capacidad del piloto para controlar el UAV se evidencia en las variaciones y cambios de dirección observados en la imagen. Se resalta el hecho de que en esta imagen no se puede observar adecuadamente como fue el recorrido de la aeronave, ya que hay gran variedad de virajes realizadas por la aeronave.



Figura 48: Recorrido del UAV prueba de Vuelo

Seguidamente se realizo una imagen que proporciona una visualización en 3D del movimiento del UAV en relación con su altitud (véase ??). En esta gráfica, se distinguen claramente las fases de ascenso y descenso del UAV, con diferentes colores para cada fase del vuelo:

Ascenso (Azul): La sección marcada en azul muestra el trayecto de ascenso del UAV. Se observa una subida gradual y controlada en la altitud, lo que indica que el piloto pudo manejar eficientemente el aumento de altitud del UAV.

Descenso (Verde): La sección marcada en verde indica el trayecto de descenso. Al igual que en el ascenso, el descenso muestra una disminución gradual y controlada en la altitud, lo que sugiere un buen desempeño del piloto en la fase de descenso.

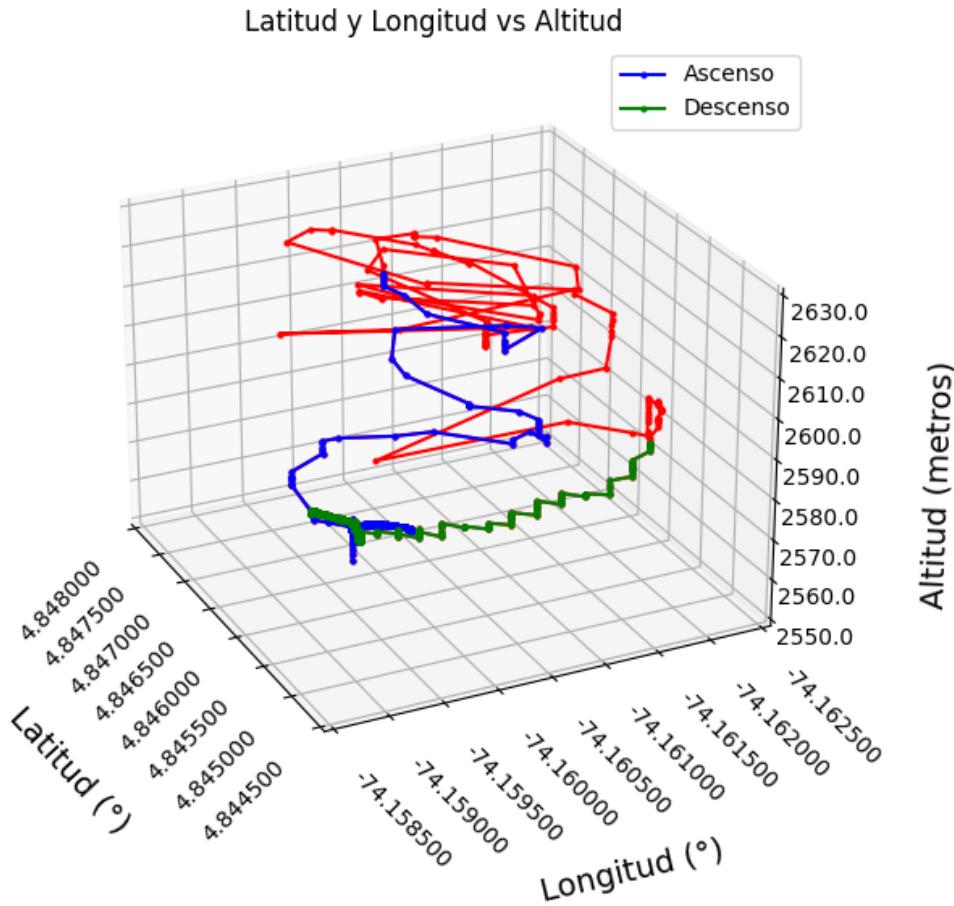


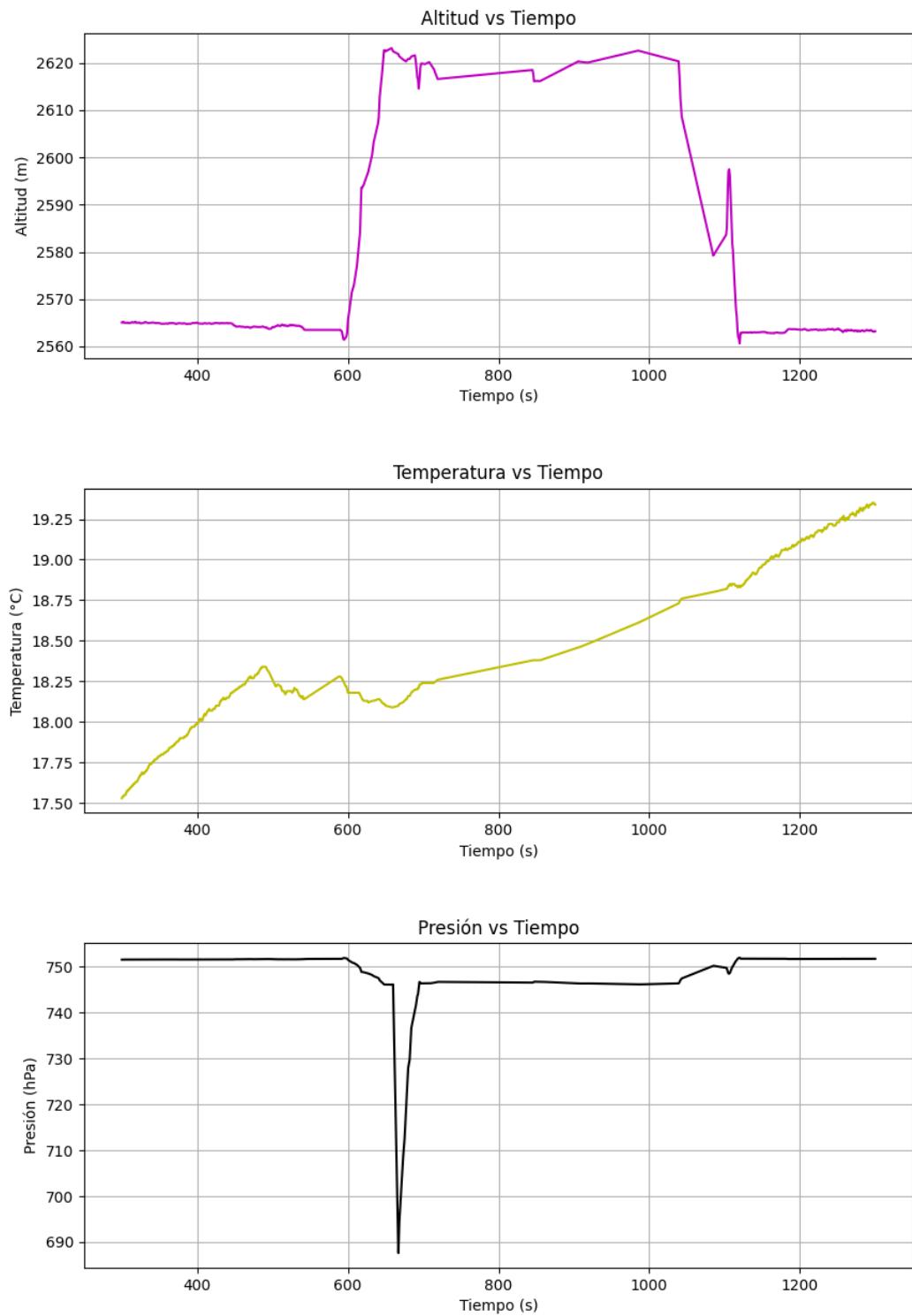
Figura 49: Latitud y Longitud vs Altitud

17.2 Datos Recibidos del Barómetro BMP280 Controlador de Vuelo

Los datos obtenidos del barómetro BMP280 durante la prueba de vuelo del UAV se pueden observar a continuación (véase ??). En donde podemos contrastar altitud, temperatura y presión vs tiempo.

- ◊ **Altitud vs Tiempo:** Entre aproximadamente 600 y 1150 segundos se realizó la prueba de vuelo, observándose un aumento de altitud de alrededor de 60 metros. A los 1100 segundos se aprecia una bajada progresiva de nivel, que corresponde a las maniobras de descenso del UAV.
- ◊ **Temperatura vs Tiempo:** La temperatura medida dentro del dispositivo muestra un aumento progresivo durante el periodo de vuelo, reflejando el calentamiento de los componentes internos del UAV.
- ◊ **Presión vs Tiempo:** La presión atmosférica se mantuvo estable la mayor parte del tiempo, excepto durante el ascenso del UAV, donde se registró una caída significativa, lo que es coherente con el aumento de altitud. Posteriormente, la presión se estabiliza una vez que el UAV alcanza su altitud máxima y durante el descenso.

BMP280 Datos del Sensor



En resumen, los datos indican que el vuelo realizado por el piloto fue exitoso. El incremento en altitud, el comportamiento térmico y las variaciones en la presión atmosférica observadas son coherentes con las expectativas de un vuelo controlado.

17.3 Análisis de Resultados de los Ángulos de Movimiento del UAV Controlador de Vuelo

Datos Recibidos del Sensor BNO055 Los datos de orientación obtenidos del sensor BNO055 durante la prueba de vuelo del UAV. En las figuras ?? y ?? se pueden observar los registros de los ángulos de roll, pitch, yaw y compass a lo largo del tiempo. Estos gráficos muestran cómo el UAV se comportó en términos de orientación durante las diferentes fases del vuelo.

- ◊ **Roll vs Tiempo:** El ángulo de roll varía entre -40 y 80 grados. Estas variaciones son típicas en aeronaves, ya que valores cercanos a 90 grados podrían comprometer la estabilidad del UAV. Los valores observados son comunes en los ángulos de banqueo de la aeronave.
- ◊ **Pitch vs Tiempo:** El ángulo de pitch varía entre -40 y 40 grados. Estos valores son usuales para evitar que la aeronave entre en situaciones de "pique", lo que podría ser peligroso. Se mantuvo cuidado en esta parte para asegurar la estabilidad del UAV.
- ◊ **Yaw vs Tiempo:** El ángulo de yaw muestra oscilaciones características durante los virajes de la aeronave. Estos cambios abruptos en la orientación están relacionados con los giros efectuados por el piloto.
- ◊ **Compass vs Tiempo:** Similar al yaw, el compás muestra oscilaciones durante los virajes, reflejando los cambios de dirección del UAV. La coincidencia de estos cambios con el roll indica la correlación entre los virajes y el banqueo de la aeronave.

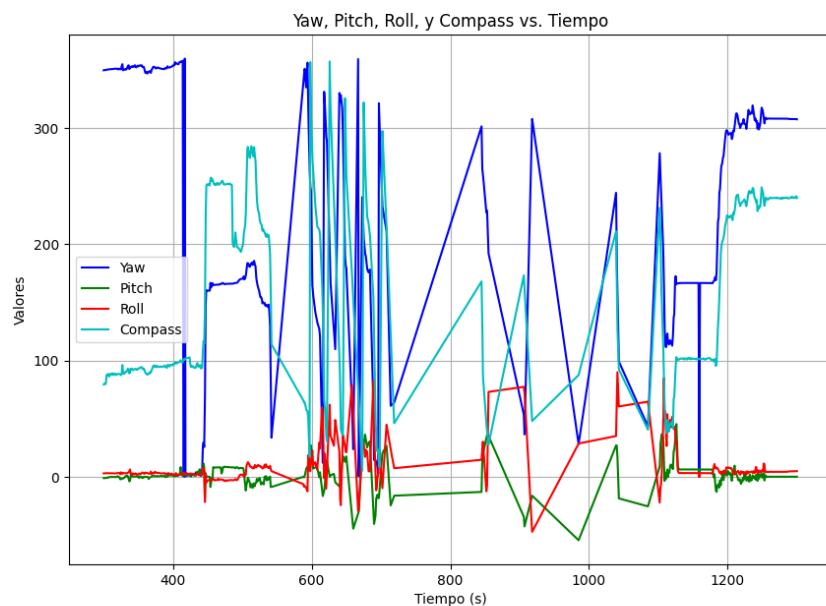


Figura 51: Yaw, Pitch, Roll y Compas vs Tiempo

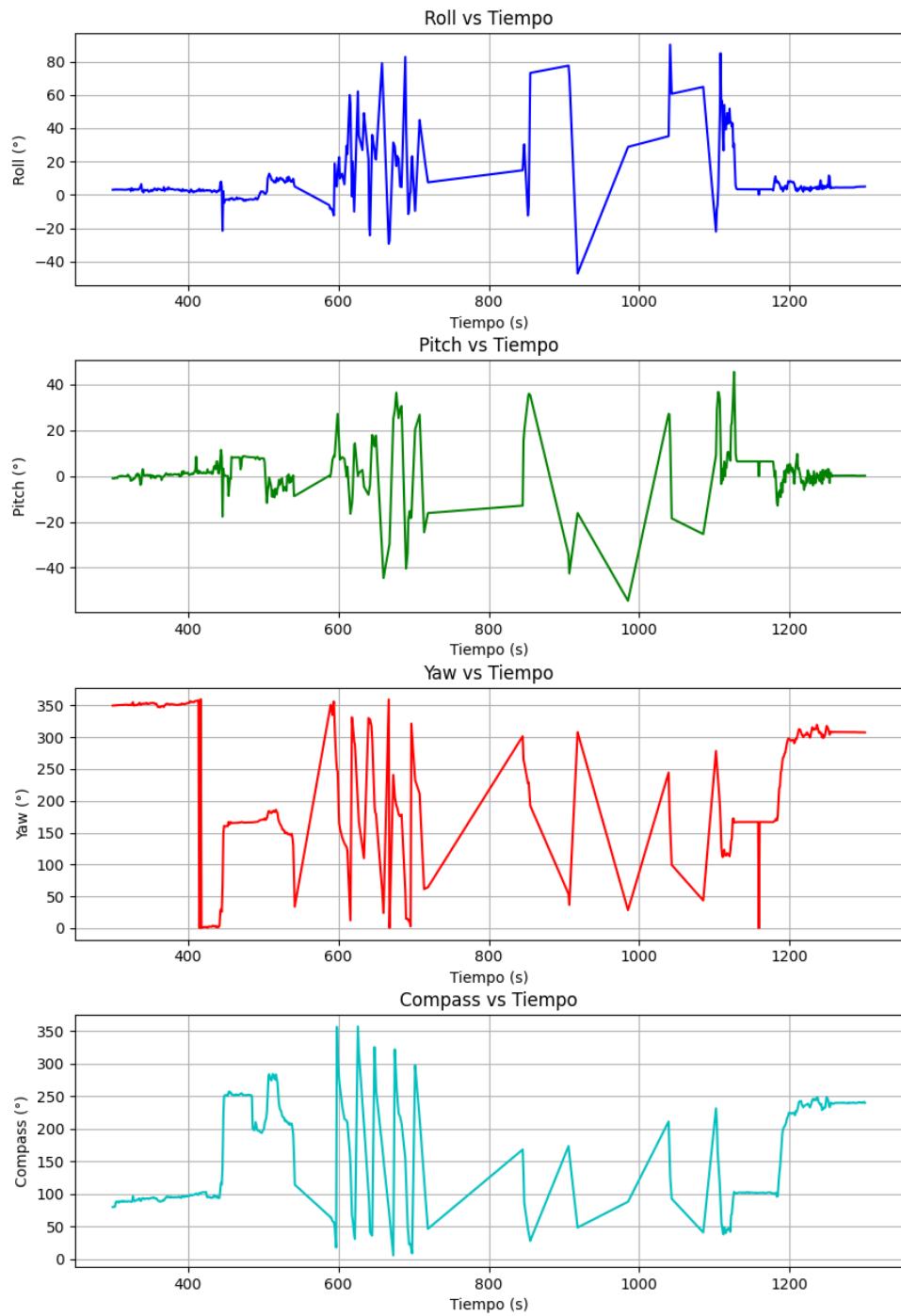


Figura 52: Datos Obtenidos BNO055

17.4 Transmisión y Recepción de Datos

La figura ?? muestra los datos transmitidos y recibidos durante la experimentación del vuelo del UAV. Aunque en algunos momentos los datos parecen coincidir, hay varios puntos en los que se observa una considerable pérdida de paquetes y corrupción de datos.

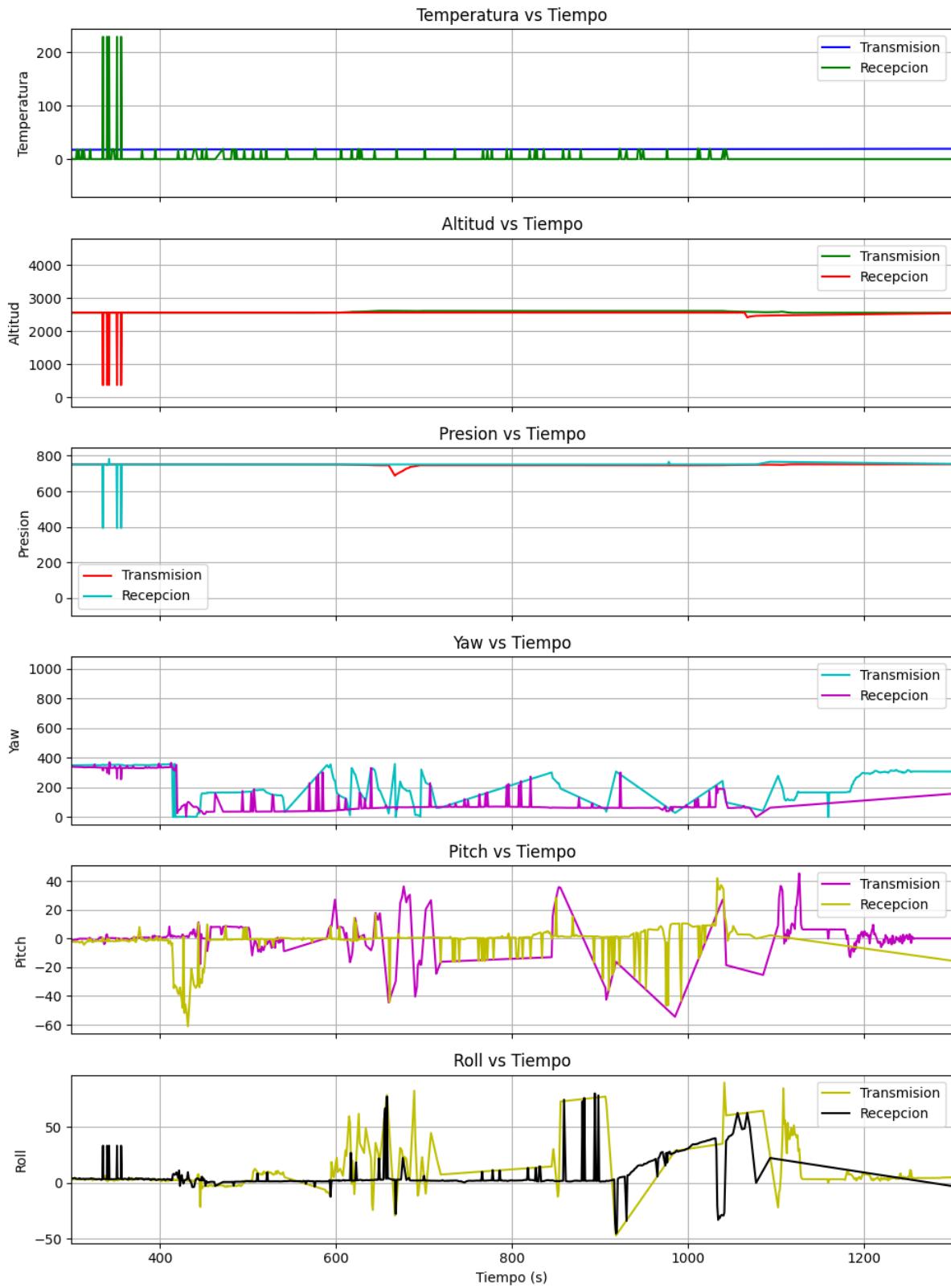
En el caso de la altitud, hay un gran desequilibrio entre los datos enviados y recibidos, con diferencias notables que indican transmisiones fallidas. Esto sugiere que hubo errores en la codificación de los datos antes de su transmisión y en la decodificación después de su recepción. Si estos procesos no se realizan correctamente, los datos pueden llegar dañados, justificando la discrepancia observada.

Los datos de presión también muestran una gran diferencia entre los valores transmitidos y recibidos. Este problema podría deberse a la pérdida parcial de paquetes durante la transmisión. El módulo nRF24L01, limitado a 32 bits por paquete, puede experimentar pérdidas fraccionarias que afectan severamente la integridad de los datos. La pérdida o corrupción de partes de un paquete puede hacer que los datos recibidos sean muy diferentes de los datos transmitidos.

Al observar los puntos obtenidos del GPS, se detecta una pérdida considerable de información transmitida durante el vuelo. Se observan incongruencias en los valores obtenidos, como 21.478790,529.000000, que no es una posición geográfica válida. Además, muchos valores del vuelo no se actualizaron correctamente, lo que sugiere una corrupción y pérdida sustancial de la información.

Estas observaciones destacan la necesidad de revisar y mejorar los procesos de codificación y decodificación para asegurar que los datos se transmitan y reciban de manera precisa y coherente, minimizando las pérdidas y corrupciones en futuras pruebas de vuelo del UAV.

Comparison of Transmission and Reception Data



17.5 Recepción datos de la Interfaz

La figura ?? muestra los datos visualizados por la interfaz de comunicación durante el vuelo del UAV, mostrando al igual que como se hizo anteriormente las variables previamente expuestas. En este grafico, a diferencia de los anteriores se muestran picos y fluctuaciones de las distintas señales. Se hace notable que las señales son representadas con pulsos cuadrados, lo que indica problemas en la recepción de información.

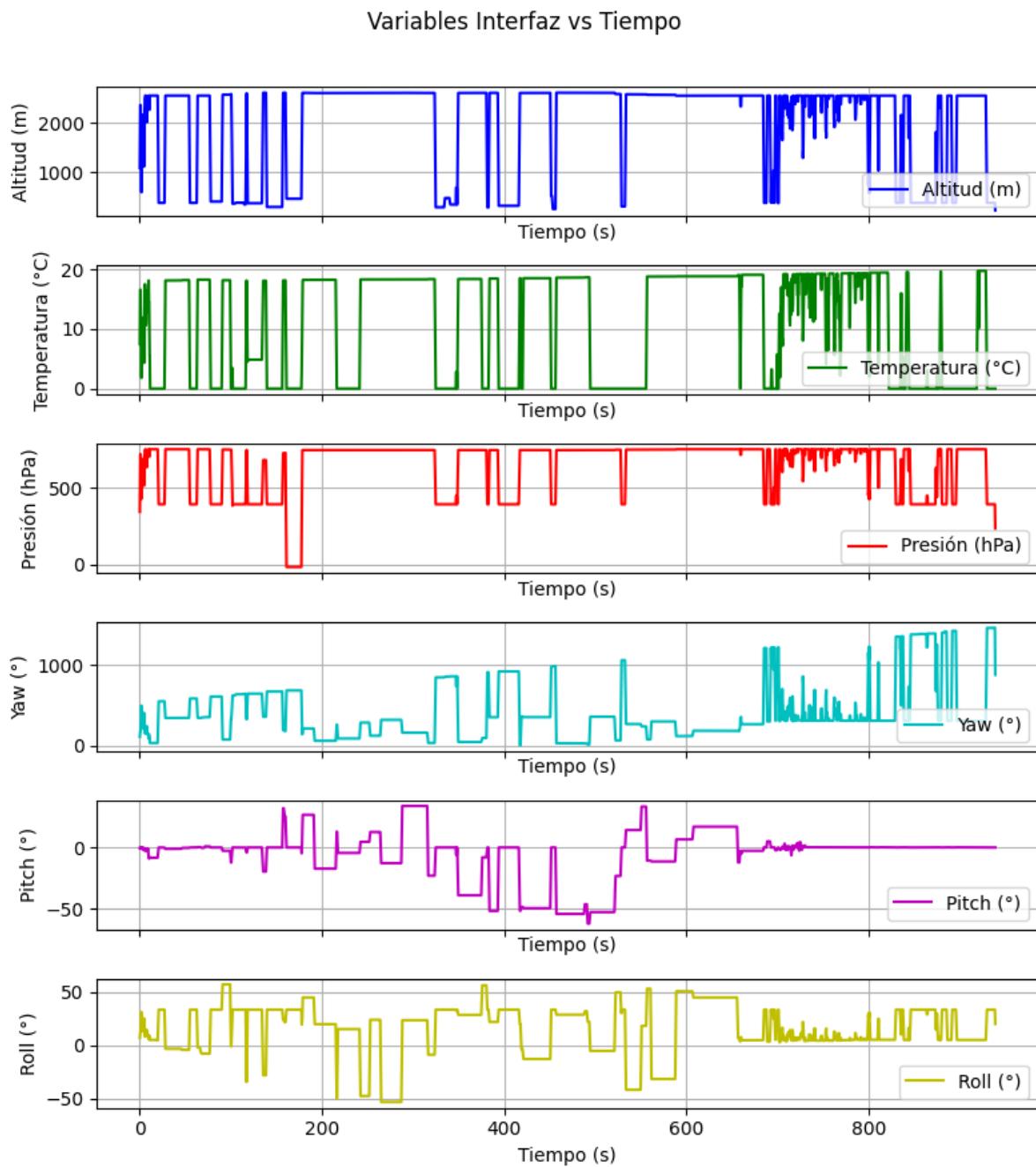
Los datos de altitud presentan numerosos picos, sugiriendo interrupciones y pérdidas de paquetes. Esto refleja problemas significativos en la transmisión de esta información. La temperatura muestra inconsistencias similares, con picos abruptos que indican que no se recibió adecuadamente la información. Los datos de presión también tienen grandes fluctuaciones, lo que sugiere problemas en la transmisión y recepción de esta variable. Además, las variables de orientación (yaw, pitch, roll) muestran variaciones bruscas y picos, reflejando inconsistencias y posibles pérdidas de datos durante la transmisión.

17.5.1. Problemas Identificados

La transmisión de datos está generando un cuello de botella que afecta la integridad de la información recibida. Esto se debe a la limitación del módulo nRF24L01, que puede experimentar pérdidas de paquetes y corrupción de datos.

17.5.2. Solución Propuesta

Para mitigar estos problemas, es esencial implementar un buffer en la interfaz de la comunicación serial. Este buffer almacenará temporalmente los datos antes de su transmisión, permitiendo una mejor gestión del flujo de información y asegurando una transmisión más eficiente. Además, es crucial optimizar la interfaz de usuario para evitar lags y asegurar una fluidez correcta en la visualización de los datos. Esto mejorará la capacidad de monitoreo en tiempo real y la fiabilidad de la información presentada.

**Figura 54:** Datos Guardados Interfaz

18 Conclusiones

En este proyecto se logró hacer el diseño de un controlador de vuelo el cual cumple con todos los requisitos para ser implementado en la operación de un UAV de ala fija. Este diseño se logró implementar y operar en una aeronave real. Esto mediante la conexión exitosa de sensores y actuadores con el controlador de vuelo.

Además, se hizo un diseño del firmware e interfaces de vuelo el cual incluye algoritmos de control y sistemas de comunicación con los sensores. Se lograron transmitir los datos de orientación, altura, presión atmosférica y temperatura a una estación en tierra donde se pudo visualizar y recuperar dicha información.

Adicionalmente, se verificó el funcionamiento de los distintos sensores integrados en la aeronave contribuyendo esto al diseño de un sistema expansible y modular, lo cual facilita futuras mejoras y la integración de nuevos componentes. Fue posible diseñar e implementar un controlador de vuelo que opera efectivamente en condiciones reales de vuelo. Transmitiendo datos de GPS y telemetría en tiempo real se logró correctamente, permitiendo una supervisión constante y la capacidad de analizar estos parámetros en tiempo real mediante el uso de una interfaz.

19 Trabajos Futuros

Durante el desarrollo del proyecto se evidenciaron los siguientes problemas:

- ◊ **Pérdida y Corrupción de Datos:** Durante las pruebas de vuelo, se observó una considerable pérdida y corrupción de datos, especialmente en las variables de altitud y presión. Esta situación se debe a errores en la codificación y decodificación de los datos, así como a limitaciones en la capacidad de transmisión del módulo nRF24L01, que puede manejar solo hasta 32 bits por paquete.
- ◊ **Cuello de Botella en la Transmisión de Datos:** La transmisión de datos generó un cuello de botella, afectando la integridad de la información recibida. Para mitigar este problema, se propone implementar un buffer en la interfaz de la comunicación serial. Este buffer ayudará a gestionar mejor el flujo de datos, asegurando una transmisión más eficiente.
- ◊ **Optimización de la Interfaz de Usuario:** Es esencial optimizar la interfaz de usuario para evitar retrasos y asegurar una visualización más fluida de los datos. Esto mejorará la capacidad de monitoreo en tiempo real y la fiabilidad de la información presentada.

Los tres objetivos principales de mejora que se pueden seguir para hacer del sistema de control de vuelo de mejor calidad, más confiable y fuerte incluyen:

Primero, mejorar el hardware del proyecto al fusionar todos los módulos en una misma placa de circuito impreso, mejorando la eficiencia del dispositivo, reduciendo el ruido y mejorando la conectividad de un módulo a otro. La estabilidad del sistema eléctrico también mejorará con un segundo inversor y un regulador de potencia continuo. Además, un módulo adicional de protección contra el flujo de corriente inversa y un sistema de gestión de baterías (BMS) deben ser parte del sistema para proteger la batería del UAV, lo cual hará que el sistema en su conjunto sea aún más confiable y duradero.

En segundo lugar, se debe seleccionar un módulo de comunicación que tenga más capacidad de transmisión de bits, con una mayor frecuencia de transmisión. Esto mejorará el flujo de datos con menos posibilidades de perdidas. También, con el uso de un sistema de comunicación, como MAVLink, se permitiría detectar fallas en la transmisión de paquetes y tener una fiabilidad mejorada en términos de transmisión de datos.

Finalmente, la interfaz de usuario debería ser mejorada y optimizada. La configuración de un buffer en la interfaz de comunicación serial hará que los datos entrantes se presenten correctamente de manera eficaz, mejorando la interfaz del usuario.

Estas innovaciones permitirán al controlador de vuelo ser más robusto y estable, mejorando la eficiencia, estabilidad y seguridad del sistema, así como la calidad de la interacción con el usuario.

Referencias

- [1] Ebeid, Emad Samuel Malki & Skriver, Martin & Laursen, Kristian & Jensen, Kjeld v Schultz, Ulrik. (2018). A Survey of Open-Source UAV Flight Controllers and Flight Simulators. *Microprocessors and Microsystems*. 61.10.1016/j.micpro.2018.05.002.
- [2] Rico, Ramon & Rico-Azagra, Javier & Gil-Martinez, Montserrat. (2022). Hardware and RTOS Design of a Flight Controller for Professional Applications. *IEEE Access*. PP. 1-1. 10.1109/ACCESS.2022.3232749.
- [3] Justin et al., "Flight Controller explained: Understanding FPV Drone Control Systems," Oscar Liang, <https://oscarliang.com/flight-controller-explained/> (accessed Dec. 5, 2023).
- [4] AAAES, "Guía UASCUAS,"Fuerza Aérea Colombiana, 2022. [Online]. Available: https://aaaes.fac.mil.co/sites/aaaes/files/AAAES/documentos/normatividad2022/Guias/guia_uascuas.pdf. [Accessed:Jun.19,2024]
- [5] G. X. Beltrán Barreto, "Antecedentes, uso y aplicación de drones en Colombia como herramienta estratégica de análisis en la agricultura de precisión,"modalidad Investigación, Facultad de Ciencias Agropecuarias, Programa Ingeniería Agronómica, Univ. de Ciencias Ambientales y Aplicadas, Bogotá, Colombia, Jul. 2023.
- [6] J. M. Zaraza Espinosa and E. J. F. Buitrago Galvan, "Diseño e Implementación de un Controlador de Vuelo para un Vehículo Aéreo Autónomo No Tripulado (UAV) Parrot Mambo, Universidad Industrial de Santander, Facultad de Ingenierías Físicomatemáticas, Escuela de Ingeniería Mecánica, Bucaramanga, 2023.
- [7] J. A. Gomez Chavarro, "Diseño y Construcción de Vehículo Aéreo No Tripulado Tipo RPV, Universidad Autónoma de Occidente, Facultad de Ingeniería, Departamento de Automática y Electrónica, Programa Ingeniería Mecatrónica, Santiago de Cali, 2008.
- [8] E. nbsp; Iaerocol and Iaerocol, "¿Qué son los drones de Ala Fija? Aprende Sus Características," Instituto Educativo Aeronáutico de Colombia, <https://iaerocol.co/blog/que-son-los-drones-de-ala-fija/> (accessed Dec. 5, 2023).
- [9] [6] Flight Controllers explained for everyone," Fusion Engineering, <https://fusion.engineering/flight-controllers-explained-for-everyone/> (accessed Dec. 5, 2023).
- [10] E. Oland, R. Schlanbusch y R. Kristiansen, Underactuated Waypoint Tracking of a Fixed-Wing UAV, IFAC Proceedings Volumes, vol. 46, no. 30, pp. 126-133, 2013. DOI: 10.3182/20131120-3-FR-4045.00007.
- [11] "Technology readiness levels," NASA, <https://www.nasa.gov/directories/somd/space-communications-navigation-program/technology-readiness-levels/> (accessed Dec. 6, 2023).
- [12] Pérez Rocha, A. (2011). Desarrollo de un controlador óptimo para un aeromodelo tipo avioneta. Uniandes.
- [13] Álvarez Gómez, S. (2018). Construcción de un vehículo aéreo no tripulado con un sistema de aterrizaje autónomo. [online] Universidad de los Andes. [Acceso : 6 de diciembre 2023]
- [14] [11]Defense Update. (2015, September 16). Reaper ER Extends RPA Missions to +33 Hours. Defense Update. Recuperado el [10/03/2024], de https://defense-update.com/20150916_reaper_er.html
- [15] O. Liang, "Flight Controller Explained: Understanding FPV Drone Control Systems," oscar Liang, 13-Sep-2023. [En línea]. Disponible en: <https://oscarliang.com/flight-controller-explained/>. [Accedido: fecha de acceso].
- [16] Equipo de redacción de Drew, "MCU: Main control unit, ¿qué es?,"blog.wearedrew.co, 6 junio 2021. [En línea]. Disponible en: <https://blog.wearedrew.co/mcu-main-control-unit-que-es>. [Accedido: 10 marzo 2024].
- [17] UAV Navigation, ÚAV Navigation en Profundidad: Qué es una IMU y para qué se utiliza,ÚAV Navigation, 10-Feb-2022. [En línea]. Disponible en: <https://www.uavnavigation.com/es/empresa/blog/que-es-un-imu>. [Accedido: 10-Mar-2024].
- [18] Umiles Group, "GPS en Drones: Para qué sirve y cuándo se utiliza,Umiles Group, 23-Ago-2023. [En línea]. Disponible en: <https://umilesgroup.com/gps-drones/>. [Accedido: 10-Mar-2024].
- [19] BricoGeek, "Sensor inercial Absoluto 9 DOF BNO055 Adafruit 4646,"BricoGeek.com. Disponible en: <https://tienda.bricogeek.com/medicion-inercial-imu/802-sensor-inercial-absoluto-9-dof-bno055.html>. [Accedido: 10-Mar-2024].
- [20] Vistronica, "Módulo MPU6050 para Arduino,"Vistronica SAS, [Online]. Disponible en: <https://www.vistronica.com/sensores/imu/modulo-mpu6050-para-arduino-detail.html>. [Accedido: 10-Mar-2024].
- [21] MATEKSYS, "M10Q-5883", 2023. [Online]. Available: <https://www.mateksys.com/?portfolio=m10q-5883>. [Accessed:10-Mar-2024].
- [22] .ESP32-S3 Series Datasheet., Espressif Systems, v1.8, 2023. [En línea]. Disponible: www.espressif.com. [Accedido: 10-Mar-2024].
- [23] FlySky RC Model Technology Co., Ltd, FS-i6 Digital Proportional Radio Control System Instruction Manual, 2015-2016. [Online]. Available: <https://www.flysky-cn.com/fsi6>
- [24] Dejan Nedelkovski, .Arduino Wireless Communication – NRF24L01 Tutorial,"HowToMechtronics, [Online]. Available: <https://howtomechatronics.com/tutorials/arduino/arduino-wireless-communication-nrf24l01-tutorial/>. [Accessed:10-Mar-2024].
- [25] Aldeltatec, "Norma IPC-2221B para Diseño de PCB - Actualización,"[En línea]. Disponible en: <https://www.aldeltatec.com/blog-diseno-con-normas-y-certificaciones/norma-ipc-2221b-para-diseno-de-pcb-actualizacion/>. [Accedido: 10-jun-2024].
- [26] International Civil Aviation Organization, Unmanned Aircraft Systems (UAS), ICAO Circular 328-AN/190, 2011.

20 Anexos

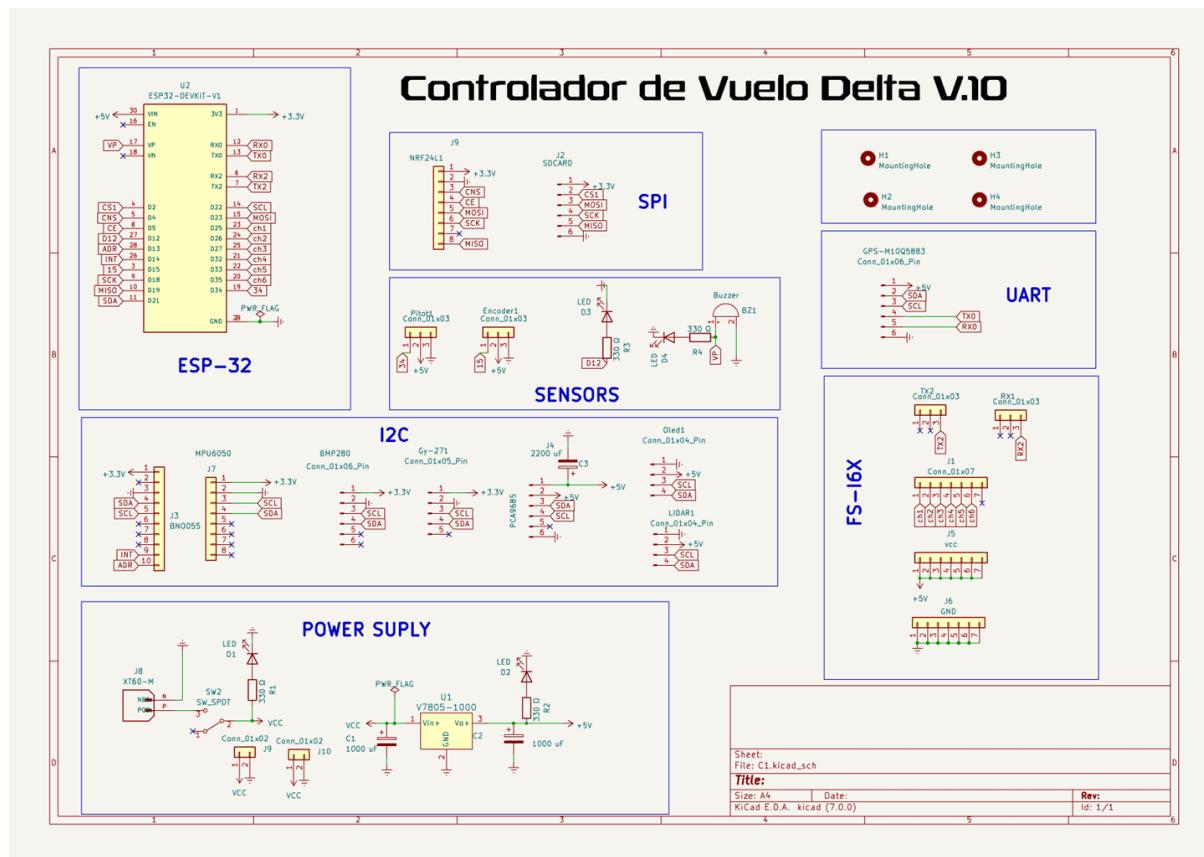


Figura 55: Esquemático Circuito Impreso 1.0

PCB- Design (1.0)

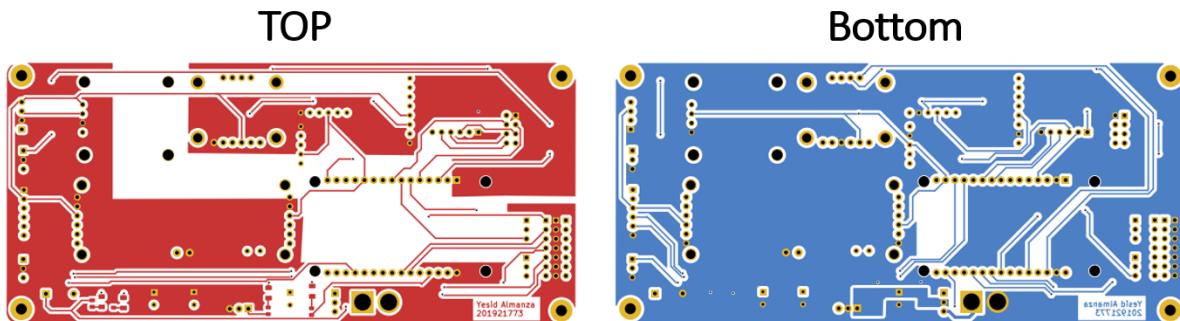


Figura 56: Layout Circuito Impreso 1.0

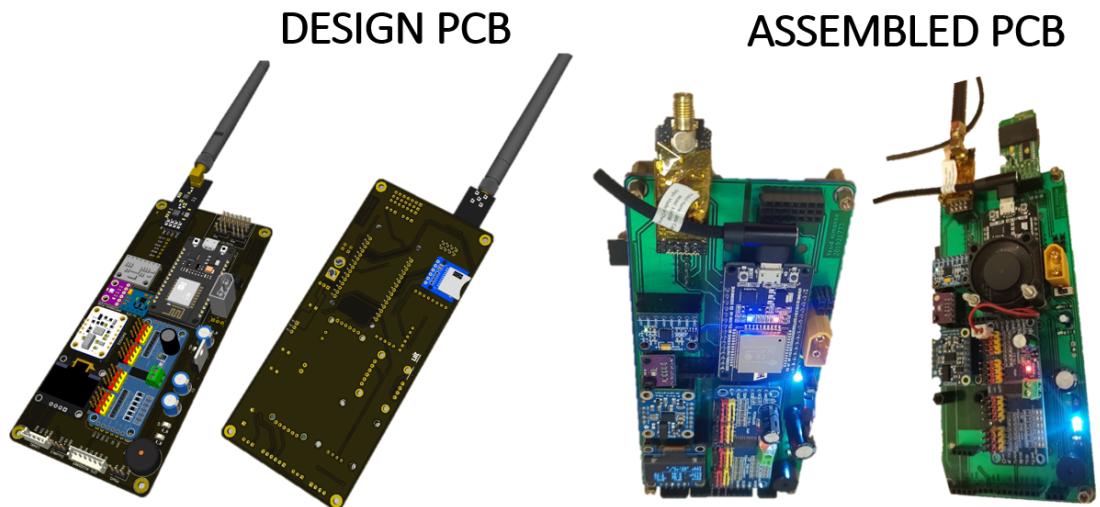


Figura 57: Circuito Impreso Ensamblado 1.0

Carcasa PCB 1.0



Carcasa PCB 2.0



Figura 58: Carcasas Versiones