



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

**Desarrollo de instrumentación
electrónica de un dron de servicios
de entrega de paquetes**

TESIS

Que para obtener el título de

Ingeniero Eléctrico Electrónico

P R E S E N T A

Luis Manuel Juárez Ruiz

DIRECTOR DE TESIS

Dr. Saúl de la Rosa Nieves



Ciudad Universitaria, Cd. Mx., 2025



**PROTESTA UNIVERSITARIA DE INTEGRIDAD Y
HONESTIDAD ACADÉMICA Y PROFESIONAL
(Titulación con trabajo escrito)**



De conformidad con lo dispuesto en los artículos 87, fracción V, del Estatuto General, 68, primer párrafo, del Reglamento General de Estudios Universitarios y 26, fracción I, y 35 del Reglamento General de Exámenes, me comprometo en todo tiempo a honrar a la institución y a cumplir con los principios establecidos en el Código de Ética de la Universidad Nacional Autónoma de México, especialmente con los de integridad y honestidad académica.

De acuerdo con lo anterior, manifiesto que el trabajo escrito titulado DESARROLLO DE INSTRUMENTACION ELECTRONICA DE UN DRON DE SERVICIOS DE ENTREGA DE PAQUETES que presenté para obtener el título de INGENIERO ELÉCTRICO ELECTRÓNICO es original, de mi autoría y lo realicé con el rigor metodológico exigido por mi Entidad Académica, citando las fuentes de ideas, textos, imágenes, gráficos u otro tipo de obras empleadas para su desarrollo.

En consecuencia, acepto que la falta de cumplimiento de las disposiciones reglamentarias y normativas de la Universidad, en particular las ya referidas en el Código de Ética, llevará a la nulidad de los actos de carácter académico administrativo del proceso de titulación.

LUIS MANUEL JUAREZ RUIZ
Número de cuenta: 314102236

Dedicatoria

Quiero dedicar mi tesis a todas aquellas personas que permitieron
hacer posible esta tesis:

A mi madre y padre por su apoyo incondicional, su amor, su
confianza en mis decisiones y por demostrarme que la fuerza
yace dentro de uno mismo.

A mi hermana por su apoyo y disposición en ayudarme a lo que
necesitase.

A mi tío Gerardo por ayudarme en aspectos técnicos de la carrera.

A mi asesor de tesis por su paciencia y guía en la revisión y
elaboración de este proyecto.

Y a todas aquellas personas que no menciono explícitamente,
pero se encuentran presentes en mis pensamientos y, que de
alguna u otra forma me brindaron su apoyo, amor y cariño,
contribuyendo a hacer posible este proyecto.

De la misma manera, quiero dedicar mi tesis a todas aquellas
personas que se esfuerzan cada día, que creen en sus sueños y
tienen algún problema o situación que los aflige; Los exhorto a
que por más difícil que sea la situación no se rindan, crean en sí
mismos y en lo que son capaces de hacer, pues como dice el
poeta Khalil Gibran:

“Por muy larga que sea la tormenta, el sol siempre vuelve a brillar
entre las nubes”.

Índice

1.	Introducción	1
1.1	Objetivo General	1
1.2	Alcances	1
1.3	Definición del problema	2
2.	Investigación del Estado del Arte	4
2.1	Descripción general	4
2.2	UAV	4
2.3	Aplicaciones	5
2.4	Clasificación de los UAV	5
2.4.1	Clasificación basada en MTOW y riesgo de impacto en el suelo	5
2.4.2	Clasificación de acuerdo con la NOM-107-SCT3-2019	7
2.5	Arquitectura del sistema	7
2.5.1	Componentes básicos de un dron	8
2.5.2	Sensores	9
2.5.2.1	Evitar colisiones	10
2.5.3	Protocolo de comunicación	11
2.5.3.1	Soportadas por el FC	11
2.5.3.2	Protocolos de comunicación serial	12
2.5.3.3	Conexión a la red	12
2.5.4	Sistema de potencia	13
2.5.5	Configuraciones de la estructura	14
2.6	Arquitectura del Software	15
2.7	Limitaciones	16
2.7.1	Tiempo de vuelo	17
2.7.2	Comunicación	17
2.7.3	Seguridad	18
2.7.4	Ambiente	19
2.8	Componentes electrónicos y drones comerciales	20
2.8.1	Controladores de vuelo (FC) y sistemas integrados para drones	21
2.8.2	Sensor LIDAR	24
2.8.3	GNSS	25

2.8.4	Drones repartidores permitidos	26
3.	Marco teórico	28
3.1	Aspectos generales sobre el Dron	28
3.2	Normativas para la implementación de un UAV en México	29
3.2.1	Requerimientos y Limitaciones del RPAS Pequeño (6.2)	29
3.2.2	Autorización para operaciones especiales	30
3.3	Ciclo de Conducción	31
3.3.1	Ecuaciones para el ciclo de conducción	32
3.4	LIDAR	33
3.4.1	Principio de funcionamiento	33
3.4.2	Tipos de LIDAR y tecnología	35
3.4.3	Sensores de movimiento inercial	37
3.5	Sistema de Comunicación	40
3.5.1	Espectro radioeléctrico en México	40
3.6	Sistema de potencia	40
3.6.1	Batería y otras fuentes	41
3.6.1.1	Voltaje de batería y voltaje de corte	41
3.6.1.2	Capacidad (C)	41
3.6.1.3	Descarga nominal y tasa de carga	42
3.6.1.4	Curvas de carga y descarga	42
3.6.2	Convertidores de conmutación	44
3.6.2.1	Topología del convertidor Buck	45
3.6.3	Medición de corriente y voltaje en dispositivos de DC	46
3.6.3.1	Filtro EMI	48
3.6.3.2	Amplificador operacional	49
3.6.3.2.1	Amplificador operacional ideal y real	49
3.6.3.2.2	Amplificadores de detección de corriente	50
3.6.3.2.3	Amplificador diferencial	51
3.6.3.2.4	Parámetros de selección	52
3.6.3.3	Filtros de entrada	53
3.7	Algoritmos de control	54
3.7.1	Algoritmo de puntos de LIDAR (Representación poligonal)	54

2.7.1.1	Convex Hull	57
3.7.2	Algoritmo RANSAC.....	58
3.7.3	Algoritmo DBSCAN	61
3.7.4	Algoritmo Voronoi	61
3.7.4.1	Algoritmo de Fortune	63
3.7.5	Algoritmo Dijkstra.....	64
3.7.6	Aplicación de Algoritmo de navegación con GNSS.....	65
4.	Diseño	67
4.1	Diseño del concepto	67
4.1.1	Requerimientos	67
4.1.2	Diagrama en bloques del sistema	68
4.2	Diseño a nivel sistema	69
4.2.1	Diagrama funcional	70
4.2.2	Estructura	72
4.2.3	Subsistema de comunicación	73
4.2.4	Subsistema de detección de obstáculos.....	74
4.2.5	Subsistema de posicionamiento.....	75
4.2.6	Subsistema de control de vuelo	75
4.2.7	Computadora de abordó	77
4.2.8	Subsistema de potencia	82
4.3	Diseño a detalle	83
4.3.1	Estructura	83
4.3.1.1	Clase y dimensiones	84
4.3.1.2	Mecanismo de entrega.....	84
4.3.2	Controlador de vuelo.....	85
4.3.2.1	Plataforma PX4	86
4.3.2.2	MAVLink.....	86
4.3.3	Motores Brushless, controladores electrónicos de velocidad (ESC) y hélices	86
4.3.4	Subsistema de Comunicación.....	90
4.3.4.1	Comandos AT.....	91
4.3.5	Subsistema de Posicionamiento.....	91

4.3.5.1	Protocolos de comunicación (GNSS)	94
4.3.6	Subsistema de Detección de objetos	94
4.3.6.1	Sensores LIDAR y mecanismo de movimiento	95
4.3.6.2	Sensores de movimiento	96
4.3.6.3	Unidad de procesamiento y placa de evaluación de alto desempeño.....	97
4.3.7	Subsistema de potencia	98
4.3.7.1	Reguladores de conmutación IC.....	100
4.3.7.2	Unidad de procesamiento de bajo consumo y placa de evaluación	102
4.3.7.3	Medición de voltaje y corriente	102
4.3.7.3.1	Medición de voltaje a través del divisor de tensión	103
4.3.7.3.2	Medición de corriente a través de resistencias shunt y amplificadores de detección de corriente	105
4.3.7.4	Consideraciones del acondicionamiento de las señales de medición.....	109
4.3.7.5	Interruptores manuales y controlados	110
4.3.7.6	Bluetooth.....	111
4.3.7.7	Batería	111
4.3.8	Computadora de abordó	113
4.3.8.1	Implementación de Software	115
4.3.8.2	Herramientas de desarrollo de software	117
4.3.9	Software	117
4.3.9.1	Software a bajo nivel de los microcontroladores de núcleo Cortex-M4.....	118
4.3.9.2	Implementación de los algoritmos del subsistema de detección de objetos	121
4.3.9.3	Implementación de los algoritmos para la unidad de control del subsistema de potencia	124
4.3.9.4	Implementación de los algoritmos de control	127
4.3.9.4.1	Algoritmo DBSCAN	128
4.3.9.4.2	Algoritmo Convex Hull.....	130
4.3.9.4.3	Algoritmo RANSAC.....	131
4.3.9.4.4	Algoritmo de Voronoi.....	132
4.3.9.4.5	Algoritmo de Dijkstra.....	135
4.3.10	Diseño de placas para pruebas (PCB)	136

4.3.10.1	Diseño de PCB para la computadora de a bordo	136
4.3.10.2	Diseño de PCB para el subsistema de potencia.....	138
5.	Pruebas y resultados	140
5.1	Resultados obtenidos del subsistema de detección	140
5.2	Resultados obtenidos de la unidad de control (MCU) del subsistema de Potencia	150
5.3	Desempeño de la computadora de abordó	164
5.4	Resultados del subsistema de posicionamiento	176
5.5	Resultados del subsistema de comunicación.....	179
5.6	Resumen de los resultados.....	183
6.	Conclusiones.....	185
7.	Trabajo a Futuro	187
Apéndices		189
APÉNDICE I.	CÁLCULO, SELECCIÓN DE LOS ELEMENTOS Y SIMULACIÓN DEL REGULADOR LTC7821 .	189
APÉNDICE II.	CÁLCULO, SELECCIÓN DE LOS ELEMENTOS Y SIMULACIÓN DEL REGULADOR LT8609	196
APÉNDICE III.	CÁLCULO, SELECCIÓN DE LOS ELEMENTOS Y SIMULACIÓN DEL REGULADOR LT8610	199
APÉNDICE IV.	GENERACIÓN DE LA PCB PARA LA COMPUTADORA DE ABORDO	203
APÉNDICE V.	GENERACIÓN DE LA PCB PARA EL SUBSISTEMA DE POTENCIA	207
APÉNDICE VI.	VERIFICACIÓN DE LA PCB DE POTENCIA.....	212
APÉNDICE VII.	Diagrama de flujo para el subsistema de potencia	216
APÉNDICE VIII.	Diagrama de flujo para el subsistema de detección de objetos	218
APÉNDICE IX.	Diagrama de flujo del algoritmo DBSCAN	220
APÉNDICE X.	Diagrama de flujo del algoritmo Convex Hull.....	221
APÉNDICE XI.	Diagrama de flujo del algoritmo Voronoi-Fortune.....	222
APÉNDICE XII.	Diagrama de flujo del algoritmo Dijkstra.....	223
APÉNDICE XIII.	Configuración de los periféricos.....	224
APÉNDICE XIV.	Promedio de las corrientes del subsistema y cálculo del menor espacio de memoria requerido	228
APÉNDICE XV.	Generación de la aplicación para el sistema	232
Referencias		240

1. Introducción

El presente documento contiene la descripción del análisis y desarrollo necesario para el diseño de la instrumentación de control de a bordo de un dron cuyo propósito es realizar entregas de manera autónoma. El enfoque del diseño está dirigido hacia la instrumentación de una computadora de abordo, la cual se encargará de dar la autonomía necesaria al sistema, con la finalidad de lograr las entregas oportunamente.

Dicho lo anterior, la tesis se dividirá en 5 capítulos centrales. El primer capítulo consiste en una descripción general y alcance esperado del proyecto. El segundo capítulo abordará el Estado del Arte, donde se hará una recopilación del alcance que han obtenido los drones en este tipo de aplicaciones, mencionando aspectos relevantes como la arquitectura, su clasificación y las limitaciones en este tipo de sistemas. En el tercer capítulo se explicará la teoría en la que se sustenta el sistema en diseño. Posteriormente, en el cuarto capítulo se presenta el diseño de la instrumentación de control de a bordo. Finalmente, se incluye un último capítulo mostrando las características y la correcta operación del sistema propuesto.

1.1 Objetivo General

Diseño de la instrumentación de abordo de un dron capaz de realizar servicios de entregas de paquetes dentro de un área específica.

1.2 Alcances

Debido a la complejidad y el propósito de la tesis, se dividirá el sistema principal en diversos subsistemas y se definirán alcances para cada uno de ellos, como se enlistan en la Tabla 1.

Tabla 1. Alcances del proyecto.

Subsistema	Alcance
Computadora de abordo	Como computadora central y gestor de las tareas del sistema del dron, la computadora de abordo debe realizar de manera mínima los siguientes puntos: • Uso de protocolos de comunicación para comunicación con otros dispositivos y subsistemas. • Manejo de GPS y periféricos. • Contar con algoritmos para, 1. Interpretación del ambiente, 2. Planeación de rutas,

Comunicación	El objetivo del subsistema de comunicación es enviar información entre dos puntos, por tanto, se debe seleccionar un dispositivo capaz de enviar datos en la mayoría del tiempo entre el dron repartidor y algún dispositivo diferente como un computador o un dispositivo móvil.
Detección de obstáculos	Como lo indica su nombre, este subsistema debe entregar información acerca del ambiente para que, cuando estos datos sean procesados por la computadora de abordo, se pueda identificar si existe algún obstáculo en la ruta. Los datos que debe enviar este subsistema deben ser, • Datos sobre información del ambiente, como posiciones referentes a la orientación de Lidar y, distancias de objetos.
Posicionamiento	El subsistema de posicionamiento debe dar información sobre la posición en la que se encuentra el sistema, es decir, altura, longitud y latitud.
Potencia	Este subsistema debe cumplir mínimamente con las siguientes funcionalidades. • Realizar el procesamiento y monitoreo de señales que se observarán en el subsistema en tiempo real a través de una unidad de control (MCU), • Envío de información sobre el estado del sistema.

Los anteriores son las funciones mínimas que se pretenden lograr, debido a la complejidad y alcance del proyecto.

1.3 Definición del problema

En los últimos años, gracias a que los drones son cada vez más accesibles, su uso es cada vez más amplio y cada vez más personas pueden disponer de esta tecnología. Además, las empresas desarrolladoras se han dedicado a hacer los drones cada vez más intuitivos y fáciles de usar para todo el público.

Cabe destacar que a la vez que los drones han ido adquiriendo una gran cantidad de aplicaciones, el comercio electrónico ha ido evolucionado y creciendo. Con las mejoras en logística ahora podemos gozar de entregas más rápidas, sin embargo, los repartidores aún enfrentan problemas como el tráfico, limitando así el número de entregas que pueden realizar por día, por lo cual, se presenta a los drones como una forma ventajosa de hacer uso del espacio aéreo y, de esta manera, lograr realizar entregas en menor tiempo y en mayor cantidad.

Por último, se puede mencionar que los drones como repartidores no sólo son útiles en el mercado electrónico, sino que presentan una ventaja a la hora de realizar entregas urgentes como lo es el sector médico y, también para alcanzar zonas donde el acceso es complicado. Asimismo, en el esquema de la pandemia que estamos observando, el uso de drones podría incluso evitar menos contagios humanos ocasionados por el contacto directo con personas enfermas. En ese contexto, diseñar la instrumentación de a bordo de un dron para una aplicación específica permitirá contar con un sistema que cumpla con los estándares de seguridad y eficiencia requeridos.

2. Investigación del Estado del Arte

2.1 Descripción general

En este capítulo se hace una recopilación del alcance tecnológico que han tenido los drones, principalmente en la aplicación de entrega de paquetes. El principal tópico es obtener la arquitectura capaz de sustentar el diseño de un dron capaz de realizar entregas con cierto nivel de autonomía.

Por último, pero no menos importante, se presenta una breve definición de lo que es un UAV, sus aplicaciones, la manera en qué se clasifican y las limitaciones encontradas en la implementación de estos sistemas.

2.2 UAV

Un sistema aéreo no tripulado (UAV, por sus siglas en inglés) puede ser definido como una aeronave que no lleva un operador humano y es capaz de volar bajo control remoto o programación autónoma. Otra forma de definir un UAV es como una aeronave que está diseñada o modificada para no llevar un piloto humano y se opera a través de una entrada electrónica iniciada por el controlador de vuelo o por un sistema capaz de permitir el vuelo autónomo, donde no se requiere la intervención del controlador de vuelo.

Durante los últimos años, se han utilizado los UAV en diferentes aplicaciones, siendo importante así, el esfuerzo en desarrollar tecnología capaz de aumentar el tiempo de vuelo y la carga útil de los UAV, lo cual resulta en varias configuraciones de UAV con diferentes tamaños, niveles de resistencia y capacidades.

Las principales características con las que debe contar un UAV son las siguientes:

1. Operación en ambientes grandes al aire libre.
2. Movimiento en 3 dimensiones.
3. Entorno operativo incierto y dinámico.
4. Detección de la presencia de obstáculos móviles y fuerzas ambientales que afectan el movimiento.
5. Limitaciones en movimiento.

Sin un piloto a bordo de la aeronave, la arquitectura de la computadora de a bordo debe ser capaz de proporcionar el entorno para la reutilización y la reconfiguración de la ruta. El controlador de supervisión de alto nivel recibe los comandos de misión de alguna estación de mando y control y los descompone en submisiones que luego se asignarán a los módulos conectados en el sistema. Tras la recepción de los puntos de inicio y destino del controlador supervisor, el planificador de rutas genera la mejor ruta en forma de puntos de ruta que debe seguir la aeronave.

Además de lo mencionado, un punto importante en el diseño de los UAV es definir los diferentes niveles de autonomía presentes en los UAV. En [1], se definen los siguientes niveles:

1. El ser humano toma todas las decisiones.
2. La computadora calcula un conjunto completo de alternativas.
3. La computadora elige un conjunto de alternativas.
4. La computadora sugiere una alternativa.
5. La computadora ejecuta la sugerencia con aprobación.
6. El ser humano puede vetar la decisión de la computadora dentro del marco de tiempo.
7. La computadora ejecuta y luego informa.
8. La computadora solo informa si se le solicita.
9. La computadora solo informa si quiere.
10. La computadora ignora al humano.

Por último, definir el nivel de autonomía del UAV permitirá conocer el grado de instrumentación necesario con que deberá contar el sistema con la finalidad de desarrollar de manera eficiente su misión.

2.3 Aplicaciones

En esta sección se debe destacar que a parte de la aplicación del UAV o dron como repartidor de paquetes, sus aplicaciones no sólo se limitan a esto, sino que su uso está creciendo rápidamente en varios dominios de aplicación, incluidos los sistemas de transporte inteligente (ITS), la agricultura, la preservación de la vida silvestre, la seguridad y la vigilancia, la detección remota y la geología, la atención médica, la educación y las ciudades inteligentes.

2.4 Clasificación de los UAV

Muchas clasificaciones de UAV se han definido utilizando múltiples consideraciones como masa/peso, velocidad, resistencia, altitud, configuración y uso militar. Debido al propósito del documento, únicamente se considerarán las clasificaciones basadas en la altitud, el peso y el riesgo que presenta para las personas.

2.4.1 Clasificación basada en MTOW y riesgo de impacto en el suelo

El peso medio de despegue (MTOW, por sus siglas en inglés), es una buena forma de clasificar a los drones de acuerdo con su peso y el riesgo que presentan para las personas en caso de impactar en el suelo. Una posible clasificación se muestra en la Tabla 2.

Una segunda forma de clasificar los UAV, se puede realizar por medio considerar el MTOW y la altitud típica de operación. Este tipo de clasificación se muestra en la Tabla 3.

Tabla 2. Clasificación basada en MTOW e impacto de riesgo. T_{GI}^1 es el tiempo mínimo entre accidentes de impacto en el suelo. Tabla recuperada de [2]

Número de categoría	T_{GI}^1 (h)	MTOW	Nombre	Nota
0	10^2	Menos de 1kg	Micro	La mayoría de los países no regulan esta categoría ya que estos vehículos plantean amenaza mínima para la vida humana o propiedad
1	10^3	Hasta 1kg	Mini	Estas dos categorías aproximadamente corresponden a aviones modelo R/C
2	10^4	Hasta 13.5kg	Small	
3	10^5	Hasta 242kg	Light/ultralight	La categoría puede basarse en aviones ultraligeros o incluso normal
4	10^6	Hasta 4,332 kg	Normal	Basado en MTOW estos vehículos corresponden a aviones normales
5	10^7	Arriba de 4,332 kg	Large	Estos vehículos corresponden mejor a la categoría de transporte

Tabla 3. Clasificación de UAV basado en MTOW y su altitud máxima típica. Tabla recuperada de [2]

Número de Categoría	MTOW (kg)	Rango	Altitud máxima típica [m]
0	≤ 25	Cercano	305
1	25-500	Corto	4572
2	501-2,000	Medio	9144
3	>2,000	Largo	Arriba de 9144

¹ En otras palabras, TGI se refiere al máximo tiempo que puede sobrevolar un sistema aéreo no tripulado, sin que ocurra algún accidente.

2.4.2 Clasificación de acuerdo con la NOM-107-SCT3-2019

La NORMA OFICIAL MEXICANA NOM-107-SCT3-2019, establece su propia clasificación para los sistemas de aeronave pilotada a distancia (RPAS, por sus siglas en inglés). Esta clasificación se basa principalmente en MTOW y, se puede observar en la Tabla 4.

Tabla 4. Clasificación de los RPAS de acuerdo con la NOM. Recuperada de [3]

Peso máximo de despegue [kg]	Nombre
Igual o menor a 2 Kg	RPAS Micro
Mayor a 2 kg y hasta 25 Kg	RPAS Pequeño
Mayor a 25 kg	RPAS Grande

2.5 Arquitectura del sistema

La investigación del estado del arte ha demostrado que la arquitectura de los UAV se puede presentar en dos partes: los sistemas de la aeronave y la carga útil que permitirá desempeñar la misión correctamente. Básicamente, la parte del avión se encarga de desempeñar el control de vuelo del UAV, mientras que la carga útil consta de la computadora de abordo y sensores, cuyo propósito es procesar la información de los sensores y otros componentes del sistema, permitiendo tomar decisiones para cumplir eficientemente con las tareas. Esta arquitectura junto con sus subsistemas se muestra de manera general en el siguiente diagrama a bloques.

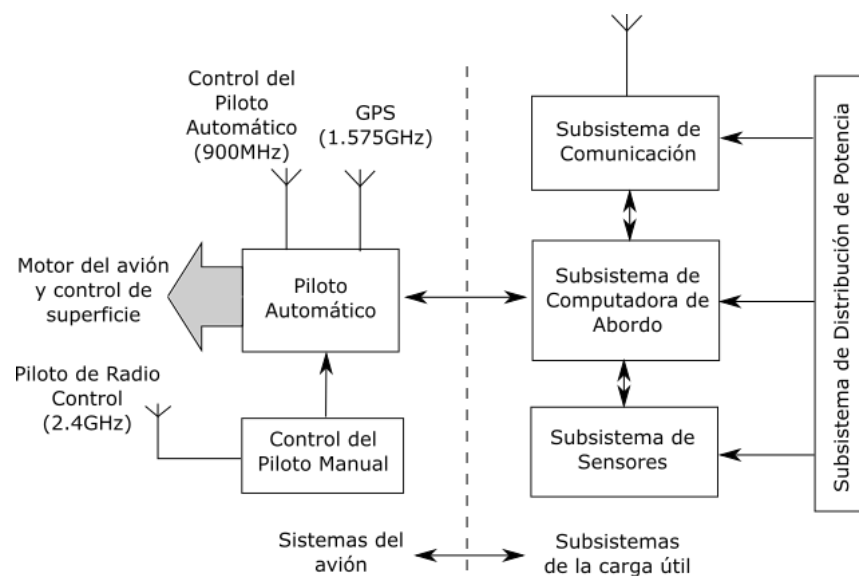


Figura 1. Subsistemas típicos en UAV. Imagen adaptada de [2]

En la Figura 1 planteada en [2], se observa que el sistema del vehículo aéreo se constituye de un piloto automático que proporciona información de posición y estado de la carga útil el cual, a su vez, se encarga de recibir comandos de la computadora de abordo. El autopiloto usa 3 sistemas de comunicaciones: Un sistema de comunicación con la estación base (900MHz), un GPS que recibe señales a 1.975GHz, y una conexión RF al piloto manual (radio control).

En el lado de la carga útil son 4 subsistemas que lo componen, siendo estos:

- Un subsistema de sensores para adquirir información en vuelo (generalmente un sensor electroóptico (EO), siendo las cámaras los principales sensores debido a costos y facilidad de adquisición).
- Un subsistema de comunicación para proporcionar conocimiento de la situación al operador de tierra y recibir comandos de éste.
- Un subsistema de computadora a bordo para analizar los datos de los sensores, ejecutar tareas, gestionar energía y proporcionar información útil sobre el estado de la aeronave al operador de tierra.
- Un subsistema de potencia para los subsistemas de la carga útil y de la aeronave en general.

En este punto, de acuerdo con [2], se debe tomar en cuenta que, para el computador de abordo, los objetivos de diseño dictan el uso de procesadores y componentes fácilmente disponibles que puedan adaptarse a las necesidades de procesamiento futuras, comportamientos autónomos y cargas útiles de los sensores. Además, el procesador debe tener múltiples opciones de entrada/salida y requisitos de energía relativamente bajos, además de ser liviano.

El diseño de una carga útil de UAV implica compensaciones entre los subsistemas del UAV mencionados. Las decisiones tomadas para un subsistema a menudo afectan los diseños de otros subsistemas. Además, mientras se gestiona el equilibrio entre los diseños de subsistemas, se deben mantener las limitaciones generales de peso y volumen del sistema UAV pequeño.

2.5.1 Componentes básicos de un dron

Anteriormente se ha mencionado mucho a los UAV, sin embargo, el tipo de UAV que se abordará en este proyecto son los de hélice rotatoria (rotary-wing), específicamente hablando de drones. Con esto en mente, es más fácil conocer cuáles son los componentes básicos de un dron y, que se encuentran de manera fácil en el mercado. A continuación, se enlistan estos componentes.

1. *Estructura*: Este es el esqueleto y lugar donde van los elementos del dron. Generalmente están hechos de fibra de carbono, por lo que son ligeros, aunque rígidos.
2. *Motor*: El motor es el principal actuador que se encarga generar el torque en las hélices para permitir el vuelo. La selección de un motor está vinculada al peso del dron e influye en la elección de otros componentes como la batería o los controladores electrónicos de velocidad.
3. *Controlador electrónico de velocidad (ESC, por sus siglas en inglés)*: Un ESC es un circuito electrónico con el fin de variar de un motor eléctrico de velocidad, su dirección y, posiblemente, también para actuar como un freno dinámico. La elección del ESC a comprar es fundamental, ya que el comportamiento de cada motor debe ajustarse continuamente. Por esta razón, se deberían preferir los ESC con una alta frecuencia de actualización.
4. *Controlador de vuelo (FC, por sus siglas en inglés)*: El FC es el encargado de dar las señales de control para orientación y estabilización del dron. En la actualidad existe de manera comercial una gran diversidad de controladores de vuelos con código abierto (Open-source), diferentes características y que permiten desarrollar diversas tareas como comunicación básica, regreso a casa, etc. Sin embargo, a pesar de su gran desarrollo, hacer uso de sólo el controlador no permitirá procesar de manera eficiente la información necesaria.
5. *Batería*: Este se encarga de entregar la potencia al sistema. Usualmente se usan de polímero de litio.
6. *Hélices*: Estas deben elegirse teniendo en cuenta la longitud del brazo del dron, y la velocidad de los motores. Su propósito es generar la aerodinámica necesaria que permite al dron elevarse.

Conocer los componentes básicos del dron ayudará a comprender de mejor manera la instrumentación que debe ser agregada al sistema para desempeñar su objetivo. En este punto, observamos que si bien, los controladores de vuelo cumplen con una variedad de tareas, no obstante, se necesita diseñar un sistema de carga útil adecuado para procesar y desarrollar todas las tareas requeridas por el sistema.

2.5.2 Sensores

Los UAV pueden incorporar diferentes tipos de sensores, cada uno sensible a una propiedad física diferente del entorno, cuyos datos se pueden integrar para hacer más robusta la percepción de la aeronave y obtener nueva información.

El principal tipo de sensores utilizados con drones, son los electroópticos (EO). Estos se han convertido en equipamiento estándar a bordo de muchos vehículos aéreos, debido a sus características, es decir, a comparación de otros sensores, una cámara es sencilla de conseguir y existe una variedad con diferentes costos, además de que, una cámara

funciona de manera similar a nuestros ojos. Sin embargo, el desafío se encuentra en el procesamiento e interpretación de la información adquirida por los sensores EO.

En [1] se mencionan la mayoría de los sensores utilizados en los UAV, donde los sensores EO principales son los siguientes:

- Cámaras de día
- Cámaras de visión nocturna
- Cámaras infrarrojas (IR)

Por otra parte, existen sensores para determinar el alcance, la altitud, la dirección o la velocidad de diferentes objetos, permitiendo así su evasión, siendo estos, más precisos y permitiendo una menor capacidad de procesamiento para la computadora de abordo. Entre estos destacan dos sensores para usos exteriores, los cuales son:

- LIDAR
- Radar

Otro tipo de sensor utilizados en los drones es la unidad de medida inercial (IMU, por sus siglas en inglés) o unidad de referencia inercial (IRU, por sus siglas en inglés) que contiene en general tres acelerómetros y tres giroscopios. Estos sensores están montados rígidamente en una base común para mantener la misma orientación relativa. Por lo general estos sensores están contenidos en el controlador de vuelo usados en la orientación y estabilidad, no obstante, también se pueden utilizar para otras aplicaciones, como lo es el funcionamiento del sensor LIDAR.

Por último, pero no menos importante, Los sensores GPS e INS son la pareja de sensores preferida para la mayoría de los sistemas de piloto automático. Estos se pueden complementar con altímetros para mejorar su precisión.

2.5.2.1 Evitar colisiones

En la actualidad existen diferentes métodos para evitar posibles colisiones con objetos. En general, se puede evitar de 3 maneras esto: En la planificación de la ruta, con el uso de un sistema de alerta de tráfico y prevención de colisiones (TCAS) y, a través de la información obtenida por los sensores.

Tomando en cuenta los sensores mencionados anteriormente, este último sistema se puede implementar de dos maneras:

- Uso de cámaras para conocer el ambiente. Puede ser un desafío lograr el procesamiento en tiempo real de las imágenes del sensor, dada la naturaleza computacionalmente intensiva del procesamiento de imágenes algoritmos. Sin embargo, esto se ha ido superando gradualmente con el desarrollo de hardware de procesamiento paralelo especializado.

- Otra manera es a través del uso de radar o LIDAR. En este caso, ambos sistemas nos ofrecen de manera más precisa la localización y distancia del objetivo. Sin embargo, a diferencia de las cámaras, estos sistemas son más costosos y requieren de bastante energía para el caso de LIDAR.

Finalmente, se muestra la Tabla 5 comparando las características de los diferentes sensores mencionados con anterioridad.

Tabla 5. Cuadro comparativo entre diferentes sensores. Tabla adaptada de [4]

Sensor	Precisión	Rango	Limitaciones	Tamaño	Precio	Ventajas
LIDAR	Alta	<200m	Afectada por el clima	Grande (Puede variar)	Alto	Detección y localización de obstáculos
RADAR	Media	<250m	-	Chico	Medio	Detección de largo alcance
Cámara estéreo	Alta	<100m	Afectada por el clima e iluminación	Mediano	Bajo	Simple y de bajo peso
Cámara Infrarrojo	baja	Depende del ambiente	Afectada por el clima	Chico	Bajo	Aplicable en condiciones oscuras

2.5.3 Protocolo de comunicación

Una manera sencilla de definir un protocolo de comunicación es como aquellas reglas utilizadas para la transmisión de información entre dos puntos. Al igual que una simple comunicación entre personas, estos protocolos permiten comunicar entre diferentes dispositivos electrónicos, sistemas, etc.

En el contexto de la arquitectura del UAV, son dos los tipos de comunicación indispensables en el sistema: comunicación serial y la comunicación utilizada para conectarse a una estación base. Cada uno de estos estándares serán descrito brevemente de acuerdo con la documentación encontrada.

2.5.3.1 Soportadas por el FC

Conocer los diferentes estándares de comunicación de los FC resulta de gran utilidad al momento su elección. Actualmente estos soportan una gran cantidad de protocolos siendo estos: SBus, iBus, SumD, SumH, PM, PPM, PWM, CRSF, JetiExBus, DSM, XBUS, XBee, FrSky, SmartPort, HoTT, LTM, MavLink, SRXL, UAVTalk, Multiwii e Iridium SBD, por mencionar algunos.

Si bien, los FC soportan una gran cantidad de protocolos de comunicación, no obstante, la mayoría son empleados en radio control, estando estos limitados en cuanto a su alcance y área de aplicación, principalmente. Asimismo, permiten la conexión por medio de un enlace satelital (Iridium SBD) con una tasa de transmisión baja. Por el momento no se entrará en detalles acerca de las limitaciones debido a que se profundizará más adelante.

Uno de los puntos de interés sobre el proyecto es el protocolo de comunicación capaz de generar la comunicación entre la computadora de abordo y el FC. En este encontramos al estándar MAVLink, que es un protocolo para establecer comunicación con el UAV por medio de mensajes codificados y, este se puede implementar a través de una estación en tierra o de una computadora de abordo.

2.5.3.2 Protocolos de comunicación serial

La comunicación serial es la recepción y transmisión de datos bit a bit. Este tipo de comunicación es la que implementan nuestros dispositivos electrónicos para comunicarse entre ellos, por ejemplo, con sensores, sistemas embebidos, computadores, unidades de almacenamiento, etc.

En los principales protocolos de comunicación serial tenemos UART, SPI, I2C, CAN y USB. De estos, los FC generalmente implementan en mayor cantidad el uso de UART, siendo este el principal medio para realizar la comunicación con la computadora de abordo por medio de MAVLink. Se puede decir que en general soporta los protocolos de comunicación básicos, siendo SPI el que tiene un menor uso entre estos.

Para finalizar esta sección, el comprender los diferentes tipos de comunicación ayuda al momento de tomar una decisión sobre los requerimientos de comunicación serial necesarios en la computadora de abordo.

2.5.3.3 Conexión a la red

La comunicación inalámbrica es una herramienta esencial para transferir la información necesaria entre el dron y la estación base. En la actualidad existen diferentes formas de tener conexión a la red inalámbrica. La Tabla 6 proporciona algunos de los protocolos de comunicación cada uno con sus características.

Tabla 6. Diferentes protocolos de comunicación. Tabla obtenida de [5]

Tipo de conexión	Tecnología	Banda de frecuencia	Tasa de transmisión	Rango
WPAN	Zigbee Bluetooth 4.0	2.4GHz (ambos)	250 kbps 24 Mbps	Hasta 100m 80-100m
WLAN	Wi-fi a/b/g/n	2.4GHz	11Mbps (b) 54 Mbps (a/g) hasta 1Gbps (n/ac)	Hasta 250 m Hasta 1km

LPWAN	SigFox	ISM EU: 868 MHz US: 902 MHz	100 bps subida 600 bps bajada	10 km urbano 50 km rural
	LoraWan	ISM 433/868/780/915 MHz	290bps-50kbps	2–5 km urban 45 km rural
Celular	NB-IoT	Licencia LTE (7-900 MHz)	234.7 kbps subida 204.8 kbps bajada	1.5 km urban 20–40 km rural
	LTE-M	Licencia LTE (7-900 MHz)	200 kbps–1 Mbps	35 km 2G200 km 3 G
	LTE D2D	2-300 GHz	234.7 kbps subida 204.8 kbps bajada	200 km 4G Hasta 300km
	5G	6-24 GHz	10 Gbps	500 m–few km

Comúnmente los drones en aplicaciones civiles suelen operar en frecuencias de 2.4 GHz a 5.8 GHz. Una frecuencia se dedica normalmente al control remoto del dron y la otra dedicada a la adquisición de vídeo. En [5] mencionan como WiFi, Bluetooth y ZigBee y los nuevos estándares en redes de área amplia de baja potencia (LPWAN) como NB-IoT, Sigfox y, LoRaWAN son preferidos en aplicaciones militares y a gran escala de drones, a diferencia de los estándares comunes de RF.

Gran parte de los drones comerciales operan con un sistema de transmisión Wi-Fi, la mayoría de las veces Wi-Fi Direct, siendo esta la comunicación directa entre el dron y la estación base. Por otra parte, en [5] se sugiere a LoRaWAN como un mejor sistema en bajo consumo, confiabilidad y misiones de largo alcance en comparación con Wi-Fi, mientras que LTE proporciona el mejor sistema de comunicación confiable y de baja latencia.

Como se observa, estos sistemas de transmisión de información no sólo nos permitirán obtener la telemetría básica con los drones, sino que, a su vez permitirá realizar aplicaciones con el internet de las cosas (IoT).

2.5.4 Sistema de potencia

El sistema de potencia es una de las partes fundamentales y limitantes de los drones, principalmente porque los motores brushless y todos los instrumentos a bordo del UAV consumen una cantidad de energía significativa, a la vez que impactan en el peso a levantar y, por consiguiente, repercuten en la capacidad y tamaño de la batería, limitando a su vez el radio de entrega.

Con esto en mente y, conociendo que por lo general el principal suministro de energía de un dron se basa en las baterías de litio recargable, debido a su excelente densidad de energía, es indispensable hablar de un sistema híbrido de potencia capaz de extender el tiempo de vuelo y uso de nuestro sistema. En el estado de arte se han encontrado pocos

autores mencionando esta posibilidad, por ejemplo, en [5] hacen referencia a 3 autores considerando las celdas solares en la implementación de UAV, sin embargo, por el momento aún no se encuentran en uso drones con celdas solares.

Lo anterior, no significa que no sea viable la idea de utilizar celdas solares, principalmente, porque es fácil notar que México es un lugar que goza de una buena irradiancia anual. Esta consideración es una razón suficiente para analizar la factibilidad de integrar estos dispositivos al sistema de drones repartidores.

Otra posible solución es el uso de celdas de hidrógeno como principal suministro de energía en los drones. Algunas empresas como Quaternium, ver [6] y [7] para más información, sugieren la implementación de celdas de hidrógeno en drones a comparación de las baterías tradicionales de litio, esto es porque de acuerdo con sus publicaciones, estas celdas proporcionan aproximadamente 3 veces la energía de las baterías de litio por el mismo tamaño, permitiendo de esta manera alargar el tiempo de vuelo sin generar emisiones contaminantes. A pesar de los beneficios anteriores, no se ha logrado encontrar drones repartidores que implementen este tipo de tecnología.

Por último, la distribución de la potencia en el sistema se realiza a través de una unidad de manejo de potencia (PMU, por sus siglas en inglés), cuya regulación de voltaje es implementada a través de fuentes conmutadas como se sugiere en [8]. La ventaja de usar reguladores conmutados en lugar de lineales está en la alta eficiencia que proveen al hacer la conversión DC-DC. Es debido a lo anterior, por lo cual, varios dispositivos comerciales implementan este tipo de reguladores como en [9] y [10], por mencionar algunos ejemplos.

2.5.5 Configuraciones de la estructura

Un aspecto relevante para considerar es la configuración de la estructura del dron. Las configuraciones más comunes encontradas en aplicaciones con UAV repartidores son la de cuadricóptero, hexacóptero y octacóptero, cada una con diferentes ventajas y desventajas, abordadas en la Tabla 7.

Tabla 7. Comparación entre diferentes configuraciones comunes con UAV.

Configuración	Características	Ventajas	Desventajas
Cuadricóptero	Consta de una estructura con distribución de 4 brazos donde se colocan los rotores y hélices	-Son ligeros, rápidos y maniobrables -Son menos costosos de fabricar y requieren menos componentes a comparación de las otras configuraciones	-El peso de carga se ve limitado por la cantidad de rotores -Tiene menor estabilidad -Si alguno de los rotores falla, no hay probabilidad de seguir volando

Hexacóptero	Su estructura permite una distribución de 6 rotores y hélices	-Tiene mayor maniobrabilidad y estabilidad respecto al cuadricóptero -Su redundancia en rotores permite el vuelo aún si uno de estos falla -Permite un mayor peso de despegue	-El precio se incrementa debido a los componentes requeridos por cada rotor -El peso de despegue aumenta por la cantidad de elementos requeridos
Octacóptero	Este tipo se caracteriza por permitir 8 rotores y hélices en su estructura	-Se incrementa la potencia y capacidad de carga -Mayor estabilidad comparada a las configuraciones anteriores	-Dadas las características de distribución de 8 motores, se requiere hélices de menor diámetro y actuadores con gran velocidad de respuesta -El peso de despegue y costo aumentan debido a la cantidad de componentes

Si bien las configuraciones en hexacóptero y octacóptero presentan mayores ventajas en cuanto a redundancia y seguridad en caso de falla de los rotores, sin embargo, una de las principales desventajas es el precio incrementado por la cantidad de componentes necesarios por rotor.

2.6 Arquitectura del Software

El software incorporado en el hardware del UAV debe ser capaz de poder generar múltiples tareas, así como correr un alto nivel de procesamiento, permitiendo de esta manera el intercambio de mensajes tanto con los diferentes dispositivos electrónicos abordo como con la estación base.

Si bien el software al igual que su contraparte el hardware es de vital interés, gracias a que en él se contienen los algoritmos necesarios para llevar a cabo la misión de manera segura y correcta. Entre las tareas que se encargan de desempeñar los procesadores se enlistan algunas como son:

1. Procesamiento de imagen y vídeo en tiempo real.
2. Detección y evasión de posibles obstáculos.
3. Aterrizaje de forma segura.
4. Toma de decisiones.

Estas tareas en el contexto de un sistema operativo en tiempo real (RTOS) deben ejecutarse en tiempo y forma dentro de un período límite conocido como “*deadline*”. Este tipo de definición entraría en la clasificación de tipo “*hard*”, en donde, por ejemplo, si la

computadora no logra dar una respuesta al controlador de vuelo en tiempo, el UAV puede estrellarse y herir a alguien, lo cual está lejos de los estándares de seguridad de los UAV.

Dentro del UAV los principales componentes de software son 3:

- Firmware: Maneja el programa a bajo nivel, siendo ejecutado desde los microprocesadores del dron.
- Middleware: Funciones del UAV, como la navegación y telecomunicaciones, se operan a través de un controlador de comunicación y algoritmos responsables de monitorear el estado del dron.
- Sistema Operativo: Se encarga de gestionar las tareas mencionadas con anterioridad, entre otras.

Estos componentes junto con las interfaces de operador/usuario harán posible desarrollar la aplicación del UAV como repartidor de paquetes. Es por esa razón que tanto el hardware como el software resultan cruciales.

En la actualidad existen diferentes herramientas de software especializado con el fin de analizar los diferentes algoritmos que se planeen implementar en los sistemas de UAV. Una de estas herramientas es MATLAB permitiendo realizar una gran variedad de simulaciones con su software.

De la misma manera existe una gran cantidad de algoritmos de control diseñados específicamente para ser lo más eficiente posible. El interés de diseñar estos algoritmos radica en el hecho de que proveen una autonomía necesaria para la navegación segura, así como permitir a las baterías proporcionar un mayor tiempo de vuelo.

Lo anterior es una visión muy general y resumida en cuanto al software, no obstante, se invita al lector a indagar más sobre los diferentes métodos y programas desarrollados para las aplicaciones con UAV. Dicho esto, en [11] se muestran algoritmos para la guía y evasión de obstáculos implementados por métodos diferentes como lógica difusa. De igual manera en [1] se da un amplio panorama acerca de los algoritmos como el control de los UAV, la planeación de las rutas y el vuelo autónomo seguro. Finalmente, en [6] se hace una recopilación de varias aplicaciones de UAV, explicando los diferentes algoritmos implementados en los UAV, como el caso de un sistema basado en la orientación en un ambiente sin conexión a GPS, entre otros.

2.7 Limitaciones

Anteriormente en otros subtemas se mencionaron algunas limitaciones encontradas en los UAV. El objetivo de este apartado es abordar las limitaciones presentes en los UAV enfocados a su aplicación como repartidores.

2.7.1 Tiempo de vuelo

Una de las principales limitantes en los UAV es la capacidad de energía que puede proporcionar el sistema de potencia a todos los componentes de éste. En ello interfieren aspectos importantes como el peso y la potencia requerida por cada subsistema.

Por lo general los drones como repartidores están diseñados para alcanzar distancias desde 1km hasta un poco más de 20km. Cada uno de estos difiere en la cantidad de sensores y dispositivos usados en sus sistemas. Es por esa razón que en cuanto más dispositivos aumentemos, menor será el tiempo de vuelo, debido a las capacidades de la batería.

En este capítulo es imprescindible mencionar cómo los actuales sistemas en paquetes (SiP) disminuyen considerablemente el tamaño de los procesadores necesarios para realizar diversas tareas. Esto a su vez, permite disminuir considerablemente el peso y la potencia demandada por los motores para levantar el UAV.

Dadas estas consideraciones, el efectuar un correcto análisis de la energía consumida resulta un factor clave en el desempeño y alcance del sistema. Una manera muy sencilla de llevar a cabo este análisis es a través del ciclo de conducción, cuyo modelo matemático está basado en el peso, movimiento y maniobras hechas por el dron repartidor. Este método, presentado en [12] y retomado en el capítulo 3.3, es la manera más común de estimar un aproximado de la energía necesaria en la ruta del dron repartidor y, de manera general, el análisis es similar para cualquier aplicación con drones.

Un análisis distinto sobre un modelo híbrido para la implementación de un sistema de potencia con celdas de hidrógeno y baterías es desarrollado en [13]. En este documento se presenta tanto el análisis para el consumo de energía del dron hasta el modelo híbrido propuesto.

Si bien, estos modelos presentados son una buena aproximación en la estimación del gasto energético de nuestro sistema, no obstante, la instrumentación a bordo es otro factor indispensable que considerar al momento de querer aumentar el tiempo de vuelo, justo como se ejemplifica en [8]. Esto en cierta medida se logra utilizando dispositivos de bajo consumo y con una escala de peso relativamente ligero.

2.7.2 Comunicación

Referente a este capítulo se encuentran diferentes factores que influyen entre la comunicación de la estación y el dron. Uno de esos es el alcance para establecer la conexión necesaria por el subsistema.

En este sentido el piloto por radio control tiende a perder el enlace con el dron a medida que se aleja de éste. De la misma manera, la interferencia se incrementa debido a los

diferentes dispositivos móviles o electrónicos que operan en la misma banda de frecuencias otras, no obstante, actualmente se usa la frecuencia de 2.4GHz, donde se han implementado protocolos para evitar este tipo de interferencia. Dicho esto, se debe mencionar que, por lo general, los drones operados por radio control se usan en espacios abiertos para evitar tener problemas de interferencias por otros dispositivos.

Además, otros aspectos de relevancia son la tasa de transferencia y la latencia en los sistemas de comunicaciones. En donde, la tasa de transferencia se define de acuerdo con las necesidades de datos por enviar y, por su parte, una menor latencia nos proporciona un mejor tiempo de respuesta por parte del sistema.

Una solución en cuanto al alcance y conectividad con diversos dispositivos es a través del enlace a una red celular. Esto en términos generales, da una mayor cobertura de área en ciertas zonas y promueve al desarrollo de aplicaciones con IoT, sin embargo, como se mencionó anteriormente, la latencia comienza a ser una limitante importante en este tipo de sistema al no ser una conexión punto a punto. Aunque, por otro lado, existen nuevas tecnologías en estas redes como la 5G, en donde, una de sus ventajas es proporcionar una baja latencia de hasta 5[ms], permitiendo una respuesta deseable en tiempo real para el sistema.

Si bien las redes celulares presentan grandes ventajas, pero de igual manera, algunos requieren de un consumo mayor a comparación de las Redes de Área Amplia de Bajo Consumo (LPWAN, por sus siglas en inglés). En este aspecto, se debe recordar que este a diferencia de las redes móviles tienen un alcance de distancia menor y, si se desea mejorar su alcance, se recomienda implementar varios puntos de acceso en el sistema general.

Finalmente, en [14] se menciona cómo la tecnología con redes celulares y LPWAN en aplicaciones con IoT se complementan la una con la otra, permitiendo una comunicación eficiente para diferentes aplicaciones con UAV. Asimismo, en este se mencionan otros aspectos generales acerca de la conexión por medio de redes celulares.

2.7.3 Seguridad

Respecto a la seguridad, los drones como repartidores tienen que atender al hecho de efectuar una navegación y entrega del paquete sin el riesgo de ocasionar incidentes para las personas.

En este aspecto, los drones deben ser capaces de detectar objetos capaces de ocasionar una colisión y derribarlos. Sobre esto, la pérdida del dron sería lo de menos, el problema ocurre cuando nuestro sistema pone en peligro la integridad de una persona.

Lo anterior igual ocurre al momento del aterrizaje para la entrega del paquete. En cierta forma las hélices pueden herir a cualquiera que entre en contacto con ellas. Es por ese motivo que el sistema igual debe considerar un aterrizaje seguro o un mecanismo para liberar el paquete de forma segura.

En el caso de México, la Norma Oficial Mexicana (NOM-107-SCT3-2019) es la encargada de regular las acciones y operaciones con drones o sistemas de aeronaves piloteados a distancia (RPAS, por sus siglas en inglés). Por ejemplo, algunas de estas regulaciones en cuestión de seguridad se enumeran a continuación:

1. Los RPAS deben operar hasta una cierta distancia de la base o el piloto.
2. Se limita la distancia de operación cerca de aeródromos, aeropuertos, etc.
3. No se permite la operación de los RPAS sobre personas que no estén participando en el proyecto o en la ejecución de la misión del dron.
4. No se permite su operación de noche.
5. No operar los RPAS de una forma irresponsable y que atente contra la seguridad de otras personas.
6. Todo daño a propiedad privada o a personas ocasionada por los RPAS debe ser reportada en un plazo límite.

Finalmente, si se desea llevar a cabo algunas de estas operaciones, como en el caso del proyecto presente, será necesario contar con una serie de permisos y requerimientos establecidos por esta norma en [3]. Una vez cumplidos estos permisos, haremos constar la seguridad de nuestro sistema repartidor implementado con drones.

2.7.4 Ambiente

El ambiente hace referencia a las condiciones en las cuales el dron se encontrará navegando de manera autónoma. En este cobra gran importancia las condiciones del clima, porque un clima con condiciones de lluvia o fuertes vientos no será favorable para operar nuestro sistema.

Algunas de las principales razones es por los efectos que ocasiona en los sensores o el mismo sistema. Por ejemplo, el agua en la lente de la cámara no permitiría obtener la imagen necesitada por el sistema o, en el caso de LIDAR, se tendrían errores en las mediciones. Con el propósito de ver cómo afecta el ambiente a los sensores, se presenta la Tabla 8.

Tabla 8. Comparación del desempeño de diferentes sensores a condiciones ambientales

Sensor	Afectado por el día	Afectado por noche	Afectado por lluvia	Afectado por niebla	Temperatura de operación aproximada ²
LIDAR	Para usos exteriores, se recomienda un valor de 100kLux ³	No afecta	Las gotas de lluvia pueden detectarse como objetos y absorber el haz	La niebla puede dispersar el haz del sensor	-20 °C a 60° C
RADAR	No afecta	No afecta	No afecta	No afecta	-45 °C a 85 °C
Cámara estéreo	Está afectada por los reflejos de la luz	Poca o nula visibilidad si no se cuenta con la iluminación adecuada	El lente de la cámara se puede ver afectado bloqueando la vista	Nula o poca visibilidad	-30 °C a 70 °C
Cámara Infrarrojo	No se recomienda para este caso	Recomendadas para operar de noche en condiciones muy poca iluminación	El lente de la cámara se puede ver afectado bloqueando la vista	No afecta	-10 a 60° C

En este caso, la Tabla 8 nos muestra cómo el radar es el menos afectado de todos los sensores por el clima. Después tendríamos a LIDAR y la cámara de visión estéreo con mayores afectaciones por la lluvia y la niebla, principalmente. Y, por último, el cámara infrarrojo sería una opción en ambientes donde la iluminación es muy poca.

Para concluir este punto, se debe considerar no sólo estas afectaciones en los sensores, sino que, también se consideran otros parámetros mencionados en el capítulo 2.5.2.1 como la resolución, etc. En el caso de radar, este sensor no sería muy bueno para detectar objetos de tamaño pequeño dentro del entorno del dron.

2.8 Componentes electrónicos y drones comerciales

En el mercado actual se encuentra una gran variedad de componentes y sistemas relacionados con los drones o UAV, siendo estos:

² La temperatura se obtuvo comparando algunos sensores comerciales, por lo que no indica un valor genérico.

³ Un Lux es una unidad de medición de iluminación sobre un área específica.

1. Dispositivos electrónicos utilizados para implementar en los drones o UAV.
2. Empresas encargadas de dar soluciones para el diseño del dron en alguna aplicación en específico.
3. Los drones que se encuentran oficialmente operando como repartidores de paquetes.

Este capítulo tiene el propósito de realizar una recopilación acerca de los elementos y sistemas más relevantes, con la finalidad de dar una solución acertada al sistema en diseño.

2.8.1 Controladores de vuelo (FC) y sistemas integrados para drones

El controlador de vuelo se encarga de dar la dirección y estabilidad requerida por el dron. Existe una gran variedad de FC en el mercado, algunos de ellos con código abierto y con diferentes características y aplicaciones.

Un punto para considerar en la elección del controlador de vuelo es seleccionar alguna que tenga una plataforma de código abierto para configurar el FC a nuestros requerimientos. Además de eso, debe considerar su aplicación para dar cierta autonomía al dron. Entre las principales plataformas que ofrecen estas características encontramos:

- ArduPilot
- PX4
- Paparazzi
- iNAV
- LibrePilot

En los primeros dos puntos se encuentra ArduPilot y PX4 con un mayor tiempo de existencia y con bastante documentación referentes al uso y aspectos de estas plataformas. Asimismo, no se debe perder de vista que todas estas plataformas contienen el estándar MAVLink, el cual posibilitará la conexión con la computadora de abordó.

Con el objetivo de entender mejor las características de los controladores de vuelo, algunos de estos se presentan a continuación en la Tabla 9.

Tabla 9. Características de algunos controladores de vuelo.

Nombre	Interfaces	Sensores/Módulos	Procesador/Otros	Modo automático
DJI A3	CAN, UART, varios pines de E/S para diferentes dispositivos como baterías inteligentes, micro-USB.	Cuenta con 3 unidades de medición inercial (IMU) de alta precisión. Cuenta con un módulo GNSS-brújula de alta precisión.	No especifica el procesador, unidad de manejo de potencia, kit de herramientas de desarrollo de software (SDK)	Sí
Pixhawk 2.4.8	UART, I2C, SPI, CAN, micro-USB	Giroscopio digital de 16 bits de 3 ejes L3GD20, Acelerómetro/ Magnetómetro LSM303D de 3 ejes y 14 bits, etc.	STM32F427 Cortex M4 de 32 bits con memoria Flash de 2MB	Sí
iFlight Beast STD F7 AIO	5 UART, conector micro-USB	Giroscopio MPU6000, Brújula QMC5883 incorporada, Barómetro DPS310.	BGA-STM32F745 con memoria Flash de 125 MB, sistema digital Vista Plug-and-Play	Sí
SKYLINE3 2 mini	UART, micro-USB	Giroscopio, brújula, posición de reserva para acelerómetro y/o barómetro.	Microprocesador ARM de 32 bits que funciona a 3.3V/72MHZ.	Sí

Se debe destacar que estas son características básicas encontradas en un FC, sin embargo, muchos otros controladores como DJI A3 agregan más funciones en su sistema, lo cual, hace incrementar sus capacidades y a la vez su precio.

Aparte de los FC, hay que hacer notar los sistemas integrados enfocados a mejorar el desempeño de las misiones realizadas por los drones. Los controladores de drones SiP (sistema en paquetes) mostrado en la Figura 2 permiten ahorrar un espacio significativo y reducir el peso para prolongar el tiempo de vuelo del dron, logrando efectuar varias tareas a la vez. En [15] se dispone de un panorama general sobre estos sistemas, por lo que solamente se mostrará información relevante de éste.



Figura 2. Sistema completo de drones OSD32MP157C-512M-BAA. Imagen recuperada de [15]

La principal característica del sistema presentado es integrar varios elementos en un espacio muy reducido, como son:

- Dos núcleos ARM Cortex-A7 que funcionan a 800MHz para procesamiento de sensores y control de los motores brushless DC.
- Un procesador ARM Cortex-A4 a 209MHz para procesamiento auxiliar.
- Permite la interfaz con memorias flash, como la uSD, facilitando el almacenamiento de datos relevantes.
- Una unidad de procesamiento gráfico 3D (GPU, por sus siglas en inglés) OpenGL.
- Memoria adicional para los núcleos del procesador, incluyendo 256 Kbytes de RAM del sistema y 384 Kbytes de RAM del microcontrolador, entre otros.
- Cuenta con una amplia variedad de puertos seriales y de control (hasta 148 GPIO). Incluyen seis SPI, seis I2C, cuatro UART, y cuatro interfaces USART que pueden conectarse a sensores MEMS y módulos GPS, dos convertidores analógico-digitales (ADC) independientes, entre otros varios.
- Incluyen varios temporizadores (hasta 29) para el control de diferentes motores.
- Módulos de seguridad para la encriptación de los códigos e información del sistema integrado.

Este tipo de sistema engloba bastantes elementos en un tamaño reducido, posibilitando desempeñar todo el procesamiento requerido por sensores, actuadores y otros dispositivos implementados en diferentes aplicaciones con drones. Asimismo, en la Figura 3 se observa de manera general algunas de las tareas ejecutadas por los SiP.

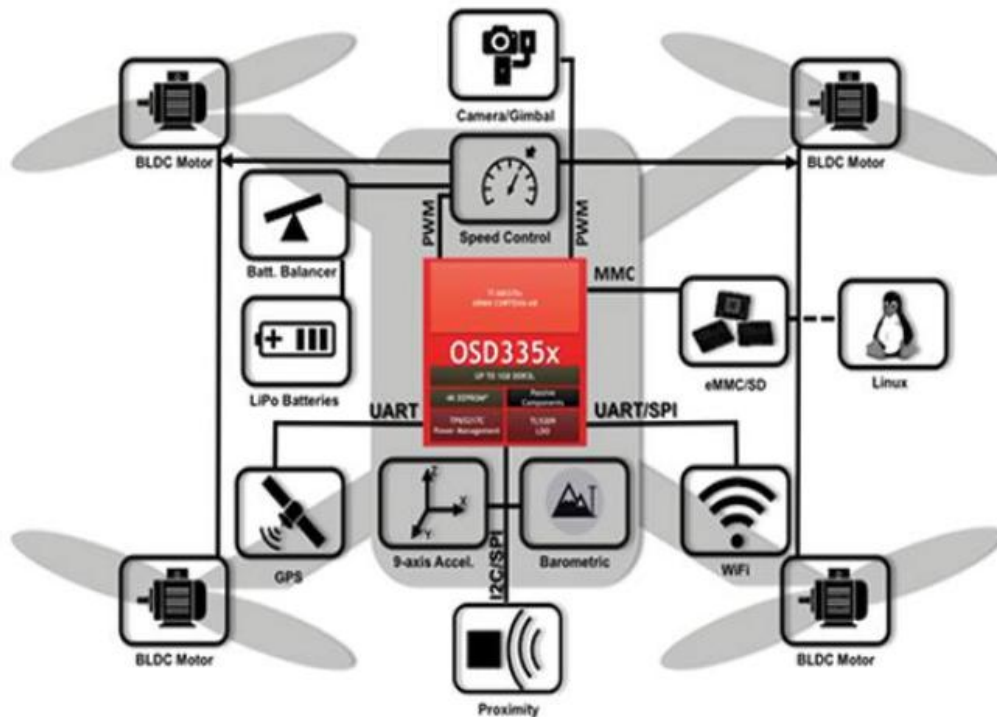


Figura 3. Esquema general de las funciones realizadas por SiP. Imagen recuperada de [15]

Como se logra apreciar, el integrar 3 procesadores de la familiar ARM Cortex en el dispositivo facilita el control y comunicación con diferentes componentes electrónicos como son los sensores de proximidad, sensores IMU, GPS, conexión a módulos WiFi, control de motores brushless, almacenamiento de información, entre otros.

Finalmente, si analizamos la arquitectura del hardware de este sistema y la comparamos con el del capítulo 2.5, su principal ventaja es que éste engloba las capacidades de procesamiento de la computadora de abordo y el controlador de vuelo (sin los módulos IMU) en uno solo.

2.8.2 Sensor LIDAR

Con referencia a los sensores se hizo el estudio principalmente de LIDAR, que nos facilita en cierta medida la detección de objetos, proporcionando su distancia y ubicación. Aunque claro, esto no quiere decir que LIDAR es mejor que una cámara. Algunos autores sugieren combinar tanto cámara con LIDAR para desarrollar un mejor sistema de detección.

Si bien existen muchos tipos de LIDAR en el mercado, pero como se ha mencionado, el costo es relativamente alto a comparación de otros, no obstante, se deben evaluar las posibilidades en el mercado. Estos sensores se enlistan en la Tabla 10.

Tabla 10. Diferentes tipos de LIDAR en el mercado.

Nombre del dispositivo	Rango	Mecanismo de mapeo	FOV	Voltaje y/o corriente	Precio Aprox. MXN
Velodyne LIDAR Puck	100m	Permite un escaneo en 3D del área.	30° Vert. 360° hor.	9-32[V]	\$75,527
YDLIDAR TX8	10m	Permite hacer escaneo en 2D, haciendo un giro de 360°.	No específica	5[V] con 450[mA] de arranque.	\$2,111
TF03	100 o 180 m	No	0.5°	5[V] con corriente mayor a 200[mA]	\$5,173
TF02-Pro	40m	No	3°	5-12[V] con 300[mA] pico	\$2,238
TFmini-S	12m	No	2°	5[V] con 200[mA] pico	\$925

Con base en la Tabla 10, se puede mencionar que los sensores hasta un rango de 40 m son una opción viable y económica para detectar objetos a diferencia de tener uno de mayor distancia. Dicho esto, por su gran rango TF02-Pro es una opción para utilizar como sensor principal en dirección al sentido vuelo, sin embargo, el campo de visión deja mucho que desear en cuanto al área de detección. Por otro lado, el dispositivo YLIDAR TX8 nos proporciona un mapeo haciendo un giro de 360°, el problema con este sensor es que no se tiene completo control sobre el área escaneada.

Una opción viable sería junto con el sensor TF03, realizar un mecanismo que permita un al dispositivo tener un mayor campo de visión (FOV, por sus siglas en inglés). Lo anterior debido a que el campo de visión de interés es hacia donde se dirige el dron.

2.8.3 GNSS

Existen muchas empresas que nos ofrecen soluciones con referencia a la localización y posicionamiento del UAV. Estas empresas son Topcon, Inertial Sense, Trimble, Sino GNSS, NovAtel, y u-blox, por mencionar algunos.

Entre los principales productos encontramos diversos módulos y placas de desarrollo para utilizar el posicionamiento preciso a través de GNSS. Cada uno cumple desde una precisión estándar hasta una alta precisión, es decir, se puede tener errores desde los 2.5[m] hasta 0.1[m], aproximadamente. Es evidente mencionar que, al aumentar la

precisión, el precio del producto sube respectivamente. En este punto es conveniente mostrar algunas de las características de los diferentes GNSS en el mercado como se encuadra a continuación.

Tabla 11. Diferentes características en GNSS

Nombre de módulos	Canales de recepción	Precisión de posición	Frecuencia de Actualización	Precisión de rumbo	Altitud y velocidad máx.	Precio Aprox. MXN ⁴
NEO-M9N-00B	92 canales	2 [m] Horizontal	25Hz	0.3°	80,000[m], 500m/s	\$1312
NEO-M8N-0	72 canales	2.5[m]-4[m] horizontal	5Hz-10Hz	0.3°	50,000[m], 500m/s	\$506
TITAN X1	99 canales	2.5[m]-3[m] horizontal	10Hz máx.	-	80,000[m], 515m/s	\$1014
ZOE-M8Q	72 canales	2.5[m]-4[m] horizontal	18Hz máx	0.3°	50,000[m], 500m/s	\$1014

En la Tabla 11, se analizaron varias opciones de GNSS/GPS comerciales de precisión estándar. Los resultados muestran cómo la mayoría tienen características muy similares, sin embargo, en este caso el módulo NEO-M9N-00B nos ofrece una mayor precisión a comparación de otros dispositivos estudiados. La ventaja de este módulo es que tiene un modo de operación llamado “multi-constelación” capaz de recibir simultáneamente señales de hasta 4 diferentes constelaciones de satélites, con el objetivo de mejorar la precisión de la posición, a diferencia de otros módulos que reciben menos constelaciones al mismo tiempo. De la misma manera, la frecuencia de actualización es mayor a la mayoría de GNSS encontrados con precisión estándar, permitiendo así desempeñar aplicaciones dinámicas.

Finalmente, estas mismas empresas se encargan del diseño de antenas para los GNSS, permitiendo un mejor enlace satelital y disminuir el ruido o la interferencia ocasionada por otros dispositivos.

2.8.4 Drones repartidores permitidos

Actualmente se encuentran en operación drones repartidores de algunas empresas como Google, Amazon y Emqopter, por mencionar algunos. En este caso, nos enfocaremos en

⁴ El precio marcado es de las placas de desarrollo encontradas en Sparkfun y en otros sitios.

algunos de los aspectos de los drones usados por Emqopter (Figura 4), cuya información es más accesible.



Figura 4. Dron repartidor de la empresa EmQopter. Imagen recuperada de [16]

La principal característica de su sistema es que son redundantes en cuanto a los sensores y dispositivos que utilizan, haciendo de esta manera que su sistema opere con mayor seguridad. Asimismo, los elementos que conforman a su dron son los siguientes:

1. Integran de 6 a 8 motores con hélices en caso de falla.
2. Sensores láser para detección de objetos, parecidos a LIDAR.
3. Tienen un controlador de vuelo pixhawk premium.
4. Usan un computador de a bordo para realizar tareas como evasión de obstáculos.
5. Cuentan con 2 GPS, implementados mediante módulos uBlox GNSS de octava generación.
6. Piloto manual en caso de fallas.
7. Incorporan un sistema de telemetría redundante operando a 433MHz.
8. Cámara Full HD con transmisión y visualización de video digital.
9. Batería estándar que permite su operación por al menos 15 minutos.

Este sistema es muy robusto como se puede apreciar y, de la misma manera, debido a la cantidad de dispositivos albergados en él es el motivo por el cual puede recorrer menores distancias (3km) a comparación de otros drones como los de Amazon (7 a 21 km).

3. Marco teórico

3.1 Aspectos generales sobre el Dron

De manera general, resulta importante conocer ciertos términos relacionados con los drones, como el eje de referencia y la forma en cómo estos generan la dinámica necesaria para su vuelo. Dado lo anterior, se describe los ejes de referencia de un dron con base en la Figura 5.

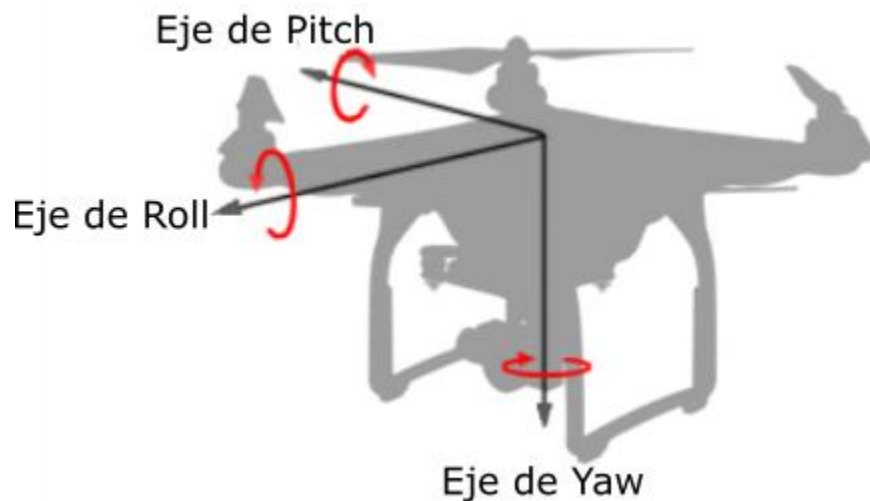


Figura 5. Ejes de referencia de un dron. Adaptado de [17]

En esta Figura 5 se observan los 3 ejes referenciados para un dron o en general para un UAV, siendo denominados *pitch*, *yaw* y *roll*. Cada uno de estos representa el movimiento del dron, justo como se describe a continuación.

- *Pitch*: Este eje mide la rotación sobre el eje Y, indicando de esta manera si el dron va hacia adelante o hacia atrás.
- *Roll*: Este eje mide la rotación sobre el eje X, indicando de esta manera si el dron va hacia la derecha o la izquierda.
- *Yaw*: Este eje mide la rotación sobre el eje Z, su función es hacer rotar al dron sobre su propio eje, cambiando de esta manera el curso de navegación.

Otro aspecto importante es conocer de manera breve cómo estos sistemas generan la dinámica para elevarse y desplazarse en el aire. El principio del vuelo de un dron está relacionado con sus hélices (principio de Bernoulli), las cuales, en conjunto al girar hacen que se genere un momento angular sobre el dron (Figura 6).



Figura 6. Giro de las hélices.

En esta Figura 6, se observa cómo es necesario un par de motores (1 y 3) girando en sentido antihorario, mientras que el otro par gira en sentido horario (2 y 4). Ahora bien, dependiendo del torque que se genere en cada uno de los motores del dron, esto hará que el momento angular cambie, generando diferentes movimientos sobre los ejes del dron y, permitiéndole de esta manera desplazarse en el aire.

3.2 Normativas para la implementación de un UAV en México

La NOM-107-SCT3-2019 es la encargada de regular las operaciones de los sistemas de aeronaves piloteados a distancia (RPAS, por sus siglas en inglés) dentro del espacio aéreo mexicano. En él se establecen los requerimientos mínimos y los permisos necesarios en cuanto a su uso privado y recreativo.

De acuerdo con la Tabla 3 mencionada anteriormente, un RPAS apto para realizar reparto como otros drones encontrados en el estado del arte, sería uno de tipo pequeño. En este caso, la norma menciona que, si el uso es privado no comercial o comercial, debe atender con los puntos 4.1, 4.11 y 6.2. Y, en caso de necesitar permisos especiales sobre estos, se deben checar los requerimientos del punto 8.

En este documento se abordarán de manera muy resumida los requerimientos de los RPAS de tipo pequeño, correspondiente a los puntos 6.2 y 8, por lo que se invita al lector a checar la demás documentación necesaria presentada en [3].

3.2.1 Requerimientos y Limitaciones del RPAS Pequeño (6.2)

Los puntos encontrados importantes en este apartado se resumen de la siguiente manera:

- Contar con una autorización de operación emitida por la Autoridad Aeronáutica.
- Se incluyen limitaciones al operar cerca de aeródromos.
- No exceder la velocidad máxima especificada por el fabricante.

- Operar la RPAS a una altura de 122[m] a excepción de operar cerca de aeródromos.
- No operar la RPAS más allá de una distancia horizontal de 457 metros (1500 ft.) respecto al piloto.
- Mantener una visibilidad mínima de 1.5 Km. (0.8 MN) desde la localización de la estación de control, antes de iniciar la operación de la RPAS.
- No operar sobre personas, a menos que participen directamente en la operación de la RPAS o estén situadas debajo de una estructura que les provea de una protección razonable en caso de desplome de la RPAS

De la misma manera todo RPAS para esta clasificación, deben contar con lo siguiente:

- Un software que automáticamente no le permita a la RPAS volar más allá de una distancia horizontal del piloto.
- Un software que automáticamente no le permita a la RPAS volar más allá de la altura máxima permitida.
- Un número de serie.
- Manual de usuario o instrucciones de uso.

Además de los puntos anteriores, en la normativa se describen otros aspectos, sin embargo, en esta tesis no se consideran porque hacen referencia a los papeles y permisos requeridos para operar el dron.

3.2.2 Autorización para operaciones especiales

En este apartado se mencionan los requerimientos para contar con la autorización emitida por la Autoridad Aeronáutica para realizar operaciones especiales como:

- Desde un vehículo en movimiento.
- Fuera del campo de visión (BVLOS, por sus siglas en inglés).
- Sobre personas.
- En la noche.
- Con múltiples RPAS. (Excepto RPAS Grande)
- Que excedan las siguientes limitaciones operacionales:
 1. Velocidad (la velocidad máxima de operación establecida por el fabricante del RPAS).
 2. Visibilidad mínima (menor a 1.5 Km. (0.8 MN), para RPAS pequeño y menor a 5 km. (2.7 MN) para RPAS grande, desde la localización de la estación de control).

Los requerimientos de visibilidad mínima son de gran interés en este proyecto porque permitirán definir una primera aproximación sobre la altura de operación del dron y la distancia máxima de viaje en línea vertical.

3.3 Ciclo de Conducción

El ciclo de conducción es un método industrial propuesto en [12] con el propósito de ayudar a realizar un correcto análisis del desempeño y alcance de un dron repartidor antes de comenzar su manufactura. Entre los principales aspectos se contempla el consumo energético del sistema y, por consiguiente, el radio máximo de entrega de éste.

El ciclo de conducción abordado es el más simple y se ilustra en la Figura 7. En este se considera un solo dron como repartidor, llevando a cabo su misión, desde su salida de la base hasta su regreso.

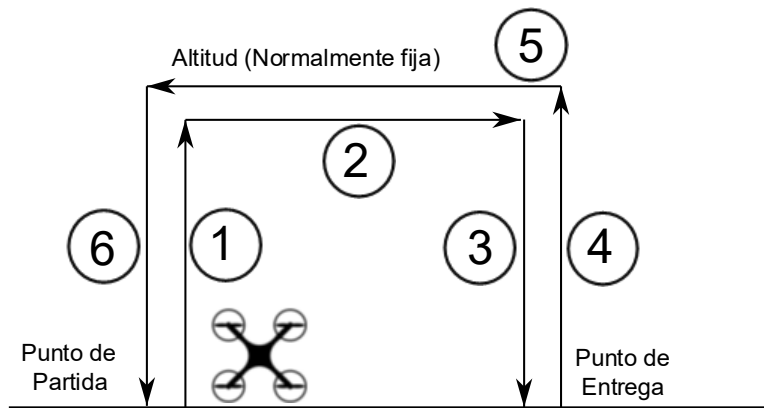


Figura 7. Ciclo de conducción de un dron repartidor. Adaptado de [12]

El ciclo más sencillo consta de 3 movimientos básicos: ascender, descender y el viaje de la ruta. A su vez, al llevar a cabo estas acciones dos veces, el ciclo de conducción es dividido en 6 segmentos, donde el dron se mueve con y sin el paquete de la siguiente manera:

1. El dron asciende con el paquete.
2. El dron se mueve hacia su destino con el paquete.
3. El dron aterriza con el paquete.
4. El dron asciende sin el paquete.
5. El dron se regresa a la base sin el paquete.
6. El dron aterriza sin el paquete.

Considerando esos 6 tramos, el siguiente punto es determinar el máximo radio de entrega del dron bajo las condiciones de funcionamiento deseadas. Lo anterior se hace calculando

cuánta energía se utiliza en cada uno de los movimientos del dron, de acuerdo como se describe en el siguiente subcapítulo.

3.3.1 Ecuaciones para el ciclo de conducción

Las ecuaciones para el cálculo de la energía consumida en cada uno de los segmentos de la ruta del dron se muestran a continuación. Primero se inicia con la energía mecánica gastada al ascender:

$$E_{ascender} = qpt \int_0^{t_f} \frac{rpm * Torque * \pi}{30 * Eficiencia} dt \quad ec. (1)$$

Posteriormente, la energía usada en el aterrizaje del dron se calcula como sigue:

$$E_{descender} = qpt * \frac{rpm * Torque * \pi * h}{30 * Eficiencia * V_{aterrizaje}} \quad ec. (2)$$

Donde los parámetros de las ecuaciones anteriores son:

- qpt se puede visualizar de manera sencilla como el número de “brazos” requeridos en la estructura del dron, conocido propiamente como “*powertrain*”⁵, es decir, los componentes utilizados para transformar la energía almacenada en energía cinética con propósitos de propulsión, siendo sus unidades adimensionales.
- t_f es el tiempo en segundos que tarda en alcanzar la altura de viaje el dron
- h es la altura de viaje del dron
- $V_{aterrizaje}$ es la velocidad del aterrizaje en segundos sobre metros
- $rpm, Torque$ y $Eficiencia$ son cualidades del “*powertrain*” dadas en unidades de rotación por minuto, newton por metro y porcentaje fraccionario, respectivamente.

Por último, la energía usada en la ruta del dron se calcula primero haciendo un ajuste al ángulo “ θ ” de viaje del dron, de tal manera que se calcula el ángulo ideal con menor gasto energético para los casos con y sin paquete. Después, con esos ángulos se obtiene la energía usada por cada metro y, por consiguiente, conociendo la energía restante para el vuelo horizontal se resuelve el rango máximo de entrega del dron, justo como sigue:

$$E_{restante} = \frac{E_{Total}}{SF} - \sum E_{ascender} - \sum E_{descender} \quad ec. (3)$$

⁵ Un “*powertrain*” consta de diferentes elementos como son los ESC, múltiples motores y hélices.

$$EPM = qpt * \frac{rpm * Torque * \pi}{30 * Eficiencia * \sqrt{\frac{2 * M * a_g * \tan(\theta)}{C_{H\text{ Arrastre}} * \rho}}} \quad ec. (4)$$

$$d = \frac{E_{restante}}{\sum EPM} \quad ec. (5)$$

En donde:

- E_{Total} , $E_{restante}$, $E_{ascender}$ y $E_{descender}$ son la energía total disponible, la energía usada en el viaje y, la energía utilizada en los dos ascensos y descensos, respectivamente en joules.
- SF es un factor de seguridad establecido por el usuario.
- EPM es la energía gastada en un metro con la configuración actual del dron, dada en joule sobre metro.
- d es la distancia horizontal máxima de entrega.
- M es la masa del dron en kilogramos.
- a_g es la aceleración de la gravedad.
- $C_{H\text{ Arrastre}}$ es el coeficiente del área de arrastre del dron en la orientación vertical.
- ρ es la densidad del aire en kilogramos sobre metros cúbicos.

En conclusión, con este sencillo método se logra dar una estimación de la distancia horizontal máxima recorrida por el dron repartidor, a través de conocer la energía disponible, o bien, si se analiza de otra manera el problema, se puede llegar a otra solución a través de conocer la energía necesaria en el subsistema de potencia para viajar una distancia en concreto. Asimismo, en [12] proporcionan la interfaz para conocer de manera sencilla la solución al radio máximo de entrega.

3.4 LIDAR

EL sistema de medición y detección de objetos mediante láser (LIDAR, por sus siglas en inglés) es una técnica muy utilizada para tomar mediciones del ambiente. Entre las aplicaciones principales de LIDAR encontramos el mapeo para el modelado 3D del ambiente, como detector de objetos, como altímetro, entre otras varias.

3.4.1 Principio de funcionamiento

El principio de funcionamiento LIDAR es similar al de muchos sensores de distancia. Este consta de manera muy básica en un foco emisor de haces de rayos láser

infrarrojos (generalmente 900nm-1550nm), y de una lente receptora infrarroja capaz de detectar estos (Figura 8). Esta medición se realiza de la siguiente forma:

1. Se emite un pulso de luz y se inicializa un conteo del tiempo.
2. Cuando el receptor detecta el pulso reflejado, se registra el tiempo total de regreso.
3. Usando la velocidad constante de la luz, el retraso puede convertirse en una distancia de separación entre el sensor y el objeto.
4. Si se conoce la posición y orientación del sensor, la coordenada XYZ de la superficie reflectante se puede calcular.

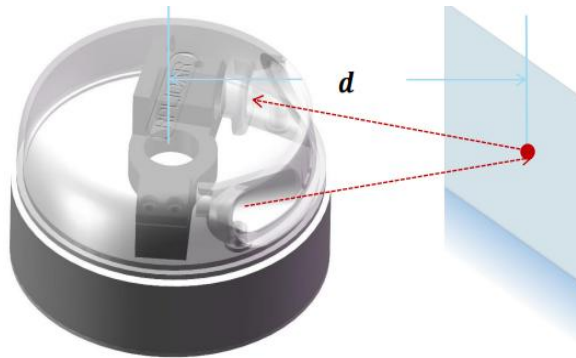


Figura 8. Representación esquemática de LIDAR, Obtenida de [¿]

El cálculo de la distancia entre el objeto y el sensor se realiza de la siguiente manera:

$$t = \frac{2R}{C} \quad \text{ec. (6)}$$

Donde:

- R: es la distancia entre el objeto y el sensor
- C: es la velocidad de la luz
- t: el tiempo

De la misma, manera. El cálculo de la posición del objeto con referencia al sensor se puede obtener por medio de trigonometría, considerando la Figura 9.

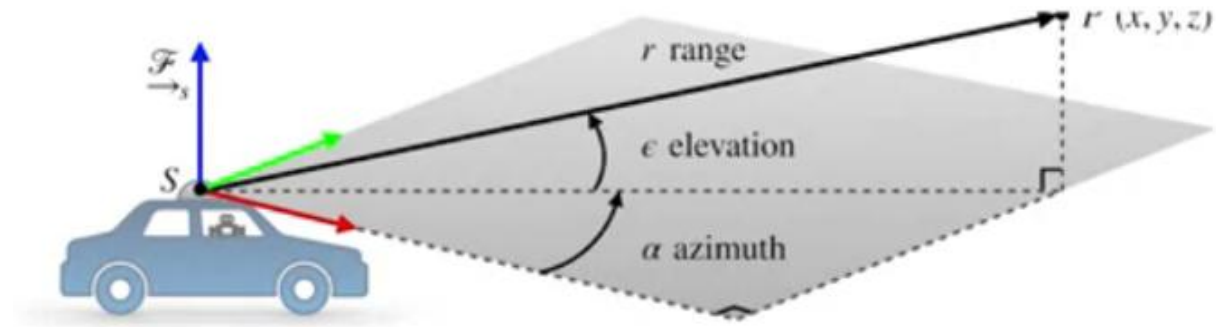


Figura 9. Determinación de la posición del objeto. Obtenido de [i]

En el plano tridimensional, las ecuaciones que satisfacen la posición del objeto son las siguientes:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} r \cos \alpha \cos \epsilon \\ r \sin \alpha \cos \epsilon \\ r \sin \epsilon \end{bmatrix} \quad \text{ec. (7)}$$

Con esta información se puede saber si el objeto se debe considerar como un obstáculo para el sistema. En caso de sí, se deberán tomar decisiones en cuanto a esto.

Finalmente, el sensor LIDAR puede registrar la onda reflejada de dos formas diferentes:

- Un retorno discreto que registra puntos (discretos) para los picos en la forma de la curva.
- Un sistema LIDAR de forma de onda completa registra una distribución de la energía luminosa devuelta.

3.4.2 Tipos de LIDAR y tecnología

Son 3 los tipos de LIDAR existente (Figura 10). Su clasificación está basada en el tipo de medición implementada por el sensor LIDAR, siendo estos:

- 1D LIDAR: Este tipo de sensor se caracteriza porque sólo mide en una dirección, es decir, distancias.
- 2D LIDAR: Un LIDAR 2D se puede ajustar por rango a una determinada zona de interés. La línea láser reflejada proporciona una sección transversal completa del objetivo de inspección.
- 3D LIDAR: Este sensor se caracteriza por permitir la medición de un área determinada, usando un campo de visión vertical como en el caso de 2D y, un

campo de visión horizontal de 360°. A diferencia de 2D, este permite un mapeo a lo largo y alto de diferentes objetos en el ambiente.



Figura 10. Diferentes tipos de LIDAR: 1D (izquierda superior), 2D (izquierda inferior) y 3D (derecha).

En términos generales, los sensores nos dan diferentes dimensiones en el plano tridimensional, para ello implementan diferentes tipos de tecnología y métodos en su sistema. Los componentes básicos que encontraremos en estos tipos de LIDAR, principalmente en 2D y 3D son los siguientes:

1. Unidad de Medición Inercial (IMU, por sus siglas en inglés): Este dispositivo implementado en el FC del dron se puede usar para calcular la orientación del sensor y, por tanto, proporciona información importante para calcular la posición del objeto.
2. Arreglo de lentes: Algunos sensores usan lentes para dispersar el tamaño del haz de luz y así, permitir un mayor rango de medición vertical u horizontal.
3. Mecanismo de rotación: Este mecanismo se implementa en diferentes sistemas con LIDAR con la finalidad de generar mapas en 2D o en 3D. Esto permite obtener varios puntos del ambiente en donde se mueve el UAV.
4. Sensores CCD o CMOS: Este tipo de sensores tienen como principio de funcionamiento el efecto fotoeléctrico, para esto se usan matrices de células fotoeléctricas, las cuales, al recibir el haz de luz infrarroja permiten el flujo de electrones, ayudando así a calcular la distancia o tamaño del objetivo.
5. Unidad de procesamiento: Algunos sensores ya integran en su sistema un mecanismo para enviar la información procesada del sensor, es decir, ya proporcionan datos como la posición y la distancia.

Otro de los métodos más usados en LIDAR es el conocido como triangulación mostrada en la Figura 11. Este se caracteriza por realizar la medición de la distancia por medio de

sensores CCD o CMOS, en lugar de calcular el tiempo de vuelo (ToF, por sus siglas en inglés).

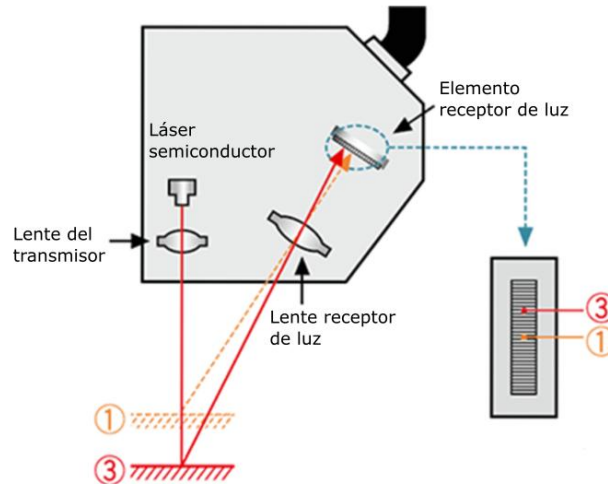


Figura 11. Método de triangulación. Esquema adaptado de [18]

Este método se caracteriza por ofrecer una mejor precisión en cuanto a la medición de la distancia del objeto. Dependiendo del arreglo del emisor y receptor, este sistema detectará objetos altamente reflejantes. En este caso, en la Figura 11 se muestra el funcionamiento de un sensor LIDAR implementado por el método de triangulación: En un principio el transmisor emite el haz del láser, el cual viajará hasta encontrar un obstáculo en su camino (1), el cual, dependiendo de sus características reflejará o difractará el haz, permitiendo al láser incidir en alguna de las celdas fotoeléctricas del receptor (formado por sensores CMOS, CCD o PSD) y, por tanto, obtener la distancia al obstáculo. Asimismo, si la distancia del objeto cambia (3), el haz incidirá en otro punto diferente del sensor, cambiando así la distancia medida.

Finalmente, la configuración mostrada en la Figura 11 no funcionaría en objetos altamente reflejantes, sino que, para este caso, el receptor y el emisor deben estar alineados de tal manera que se contemple una trayectoria justo como se ilustra en la Figura 8.

3.4.3 Sensores de movimiento inercial

Anteriormente se habló de los sensores de movimiento inercial (IMU, por sus siglas en inglés) como una manera para obtener la orientación de algún objeto, sin embargo, esto no explica del todo cómo se calcula la orientación a través de estos sensores, lo cual resulta importante para el interés de esta tesis.

El acelerómetro es un dispositivo que mide la fuerza de aceleración resultante en cada uno de sus ejes, siendo estos de 3 ejes comúnmente. La unidad de salida en los acelerómetros se mide en “g” y su equivalencia es $1g \approx 9.81 \frac{m}{s^2}$.

Ahora, una manera para entender sencillamente cómo usar el acelerómetro para medir la orientación, se puede observar a través del análisis de fuerzas de un acelerómetro en un solo eje, justo como en la Figura 12.

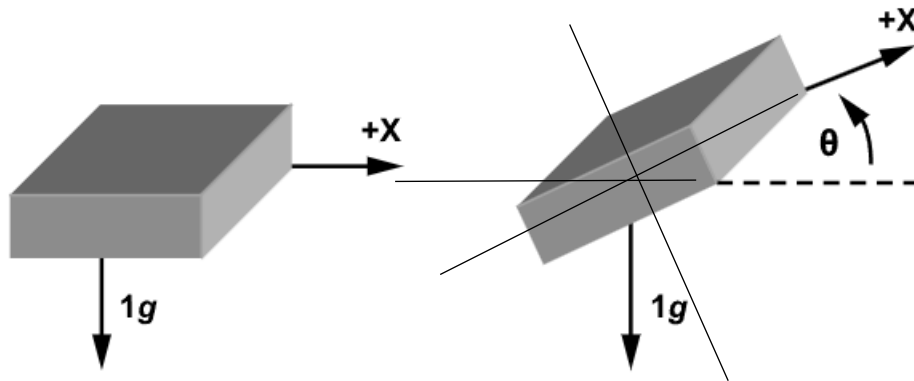


Figura 12. Representación de un acelerómetro cambiando su orientación. Tomado de [19].

Si hacemos el análisis de las fuerzas vectoriales de la Figura anterior para el primer diagrama, obtenemos lo siguiente,

$$A_x = 0$$

$$A_y = 1g$$

Sabemos, por tanto, que, si el eje “y” del sensor está alineado con el eje de la tierra, la aceleración resultante sobre ese eje es la fuerza de gravedad. Entonces, si inclinamos un poco el sensor como en el segundo diagrama, las fuerzas de aceleración no son las mismas, por lo cual, haciendo la descomposición vectorial y usando trigonometría básica para proyectar los componentes en los ejes “x” e “y” del sensor, llegaremos a que la aceleración en cada eje es,

$$A_y = g \times \cos(\theta)$$

$$A_x = g \times \cos(90 - \theta) = g \times \sin(\theta)$$

Por consiguiente, si conocemos la aceleración del acelerómetro, se puede conocer el valor del ángulo que produjo un cambio en la salida del sensor, de la siguiente manera,

$$\theta = \sin^{-1}\left(\frac{A_x}{g}\right) = \cos^{-1}\left(\frac{A_y}{g}\right) \quad ec. (8)$$

El pasado análisis es una forma sencilla de entender cómo calcular el ángulo de inclinación cuando cambia únicamente sobre un eje, pero ahora si es un acelerómetro de 3 ejes, se llegará a las siguientes ecuaciones mencionadas en [19]

$$\theta = \tan^{-1}\left(\frac{A_x}{\sqrt{A_y^2 + A_z^2}}\right) \quad ec. (9)$$

$$\Psi = \tan^{-1}\left(\frac{A_y}{\sqrt{A_x^2 + A_z^2}}\right) \quad ec. (10)$$

$$\Phi = \tan^{-1}\left(\frac{\sqrt{A_x^2 + A_y^2}}{A_z}\right) \quad ec. (11)$$

En el caso del giroscopio este se encarga de medir la velocidad angular de un objeto respecto a un eje de referencia inercial, el valor de salida del sensor se mide en [°/s]. Respecto a este sensor, la manera en que se puede medir el movimiento angular respecto a su posición inicial es por medio de integrar la velocidad angular ω . Si conocemos que la velocidad angular está definida como,

$$\omega = \frac{d\theta}{dt} \quad ec. (12)$$

Entonces,

$$\theta = \int \omega dt \quad ec. (13)$$

Esta expresión se puede calcular de forma discreta de la siguiente forma,

$$\theta = \omega(t_2 - t_1) \quad ec. (14)$$

Finalmente, la información del acelerómetro y el giroscopio se complementa entre ambos, porque mientras que con uno se puede calcular la inclinación del eje de referencia respecto al eje de la tierra, con otro podemos medir cuántos grados se mueve respecto a un eje.

Asimismo, existen muchos algoritmos que se utilizan en conjunto con estos sensores para diferentes aplicaciones, sin embargo, no se mencionarán por alcances de esta tesis.

3.5 Sistema de Comunicación

3.5.1 Espectro radioeléctrico en México

No todos los continentes y regiones en el mundo operan con las mismas bandas de frecuencias, por esa razón se debe conocer cuáles bandas de frecuencia se manejan en la región donde se planea tener servicios de comunicación por medio de redes celulares. Dicho esto, el Instituto Federal de Comunicaciones (IFT), hace una recopilación sobre el espectro radioeléctrico en México dispuesto en [20]. Específicamente este subcapítulo se enfocará a las redes de mayor capacidad y menor latencia como la tecnología LTE, aunque también se retomarán la conectividad por GSM/GPRS/EDGE y, de acuerdo con [20]⁶, las frecuencias en México para las tecnologías anteriores se enlistan a continuación.

- LTE-FDD: B2, B4, B5, B7, B8, B9, B10, B18, B21, B25, B28, B66.
- LTE-TDD: B33, B37, B39, B40, B43.
- GSM/GPRS/EDGE: 850MHz, 900MHz, 1800MHz, 1900MHz.

Tener presentes estas bandas nos asegura que nuestro dispositivo de comunicación podrá tener gran cobertura y funcionará en la región donde nos encontremos operando. En general, de la conectividad en México, las bandas de frecuencia más usadas para LTE son B2, B4, B5, B7 y B28.

3.6 Sistema de potencia

El sistema de potencia o unidad de manejo de potencia (PMU, por sus siglas en inglés) es el encargado principalmente de realizar la conversión de voltaje a través de reguladores de voltaje. Asimismo, no solamente se limita a esto, sino que también puede distribuir, controlar y monitorear el sistema de manera eficiente, con lo cual, se obtienen datos para conocer y mejorar el uso de energía disponible. Entre los principales elementos encontrados en este sistema se encuentra:

1. Una unidad de control o un SOC (system on chip).
2. Reguladores de tensión (lineales o de conmutación).
3. Interruptores.
4. Sensores de corriente y voltaje.
5. Diferentes tipos de fuentes de alimentación.
6. Otros dispositivos dependiendo de la aplicación.

⁶ El documento muestra varias bandas con perfil LTE, pero no deja claramente si está cubierto por ese mismo perfil, con lo cual no se consideraron.

3.6.1 Batería y otras fuentes

La batería es el principal elemento importante en el sistema en general, porque sin energía, el sistema no funciona. Además de ello, el definir la batería y sus características, permite definir algunos elementos dentro del sistema de potencia, por ejemplo, si se tendrá una única entrada de fuente o si es posible tener más de una fuente de alimentación como se ha mencionado en el subcapítulo 2.5.4 y, por lo general, la fuente principal son baterías de polímero de litio.

Con lo anterior, en este apartado se darán a conocer algunos conceptos importantes sobre la batería y cómo de acuerdo con esos parámetros elegir o aproximar una batería al diseño del dron.

3.6.1.1 Voltaje de batería y voltaje de corte

El voltaje de la batería de una celda de LiPo por lo general tiene una tensión de 3.7[V] y esta única celda es considerada en aplicaciones con drones como “1S”. Por tanto, si se seleccionará una batería de 6S, se estará disponiendo de una tensión de $6 \times 3.7[V] = 22.2[V]$.

Conocer la tensión no sólo resulta crucial en el diseño del sistema de potencia, sino que, a su vez determina la capacidad en el torque de los motores, entre mayor sea el voltaje, mayor serán las revoluciones y, por tanto, el torque ofrecido por estos mismos ayudará a mover más rápido al dron, siempre y cuando se consideren las especificaciones del motor.

Como último punto, el voltaje de corte es otra característica de diseño del sistema, donde este denota la tensión a la cual la batería se considera que está descargada completamente y, donde, una descarga adicional podría dañarla. Este valor usualmente lo especifica el fabricante de las baterías indicando el umbral de carga y descarga.

3.6.1.2 Capacidad (C)

La capacidad se define como la cantidad de amperes que puede ofrecer la celda durante una hora a temperatura ambiente antes de alcanzar el voltaje de corte. Sus unidades son los Amper-hora [Ah] y, se calcula de la siguiente manera.

$$C = A * t[Ah] \quad ec. (15)$$

Donde:

- C es la capacidad
- A es la corriente requerida para descargar la batería en 1 hora

- t es el tiempo en horas

3.6.1.3 Descarga nominal y tasa de carga

Además de la capacidad de la batería, existen parámetros relevantes en la selección de la batería como “C nominal”. Este parámetro básicamente consta de dos factores: la capacidad de la batería y un factor numérico. Su única función es la de indicar la capacidad máxima a la que se puede descargar la batería de manera continua y segura, por ejemplo, si se tuviese una batería de 12000mAh con un factor 15C, está podría ofrecer $15(12000mAh) = 180Ah$ ó desde otro punto de vista 180A por un par de minutos.

Existen aplicaciones donde se requiere una mayor capacidad a la que indican las baterías, por lo cual, este tipo de baterías ofrecen la posibilidad de descargar a una mayor corriente sus baterías con la excepción de disminuir su tiempo de uso. La forma de calcular el tiempo de uso de sus baterías es similar a la fórmula anterior, y los fabricantes la definen como:

$$t_{Batería}(h) = \frac{C(Ah)}{Corriente_{requerida}(A)} \quad ec.(16)$$

Donde:

- C es “C nominal”
- A es la corriente requerida
- t es el tiempo en horas

La única discrepancia con la anterior al aplicar esta fórmula es que se debe verificar el factor “C nominal” para verificar efectivamente que la batería puede proveer de la potencia necesaria en el tiempo estimado.

Finalmente, la tasa de carga indica de la misma manera la máxima corriente con la que se carga la batería y, esta se puede considerar para el sistema de potencia si se desea cargar directamente al dron, pero de la misma manera se puede retirar disminuyendo el hardware y, facilitando al uso de otra batería.

3.6.1.4 Curvas de carga y descarga

A diferencia de los cálculos ideales indicados anteriormente, las curvas de descarga muestran información precisa sobre el comportamiento de la batería, más específicamente el tiempo de uso. Un ejemplo de lo anterior se muestra más detalladamente en [21] con la Figura 13.

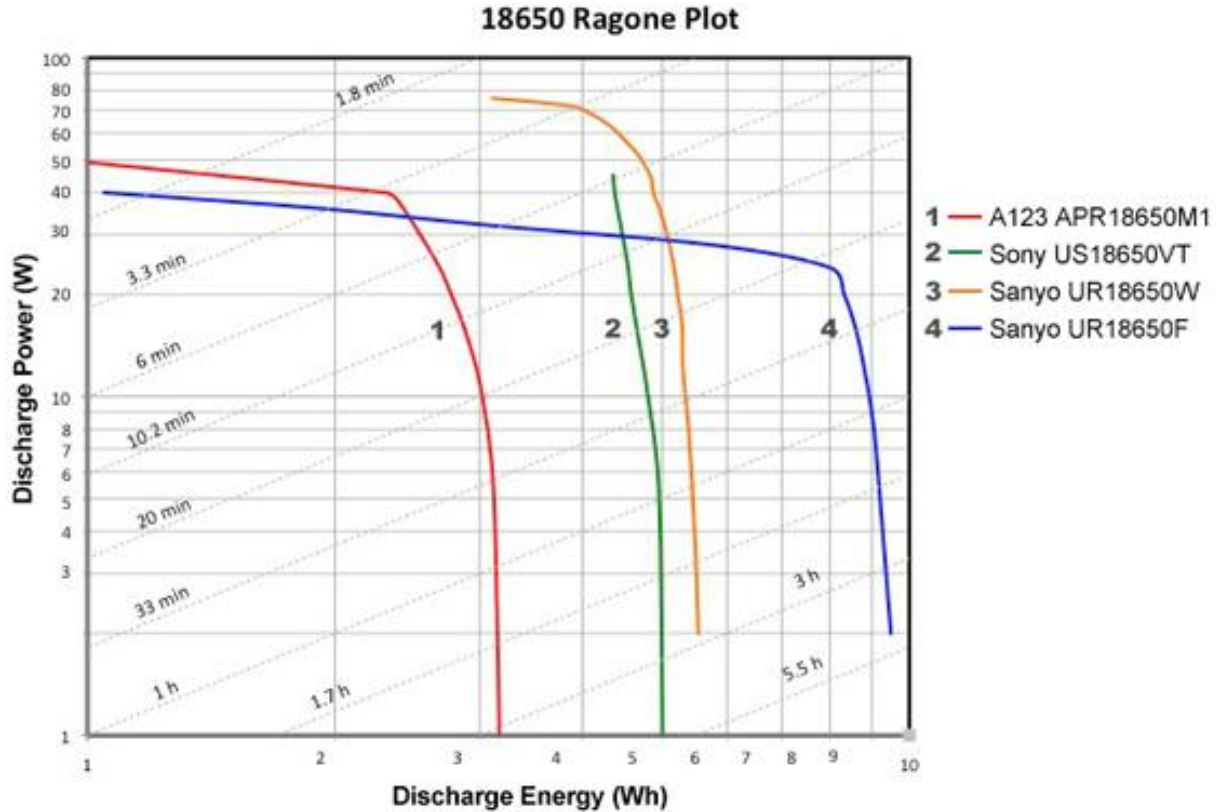


Figura 13. Curvas de descarga celdas de Li-ion. Tomada de [21].

En esta gráfica se divulga algunas de las curvas de descarga de 4 celdas diferentes de Li-ion y, con ello se puede conocer el tiempo de uso por medio del consumo de potencia o de energía estimada. Lo relevante de este estudio es cómo la batería puede tener diferentes tiempos de operación a diferentes cargas, por ejemplo, en el caso de la celda A123 APR18650M1 cuya capacidad es de 1.1Ah, se observa cómo para una carga que requiera 10[W], el tiempo de operación es de 20 minutos, donde se enfatiza claramente que la celda puede suministrar más potencia en un menor tiempo.

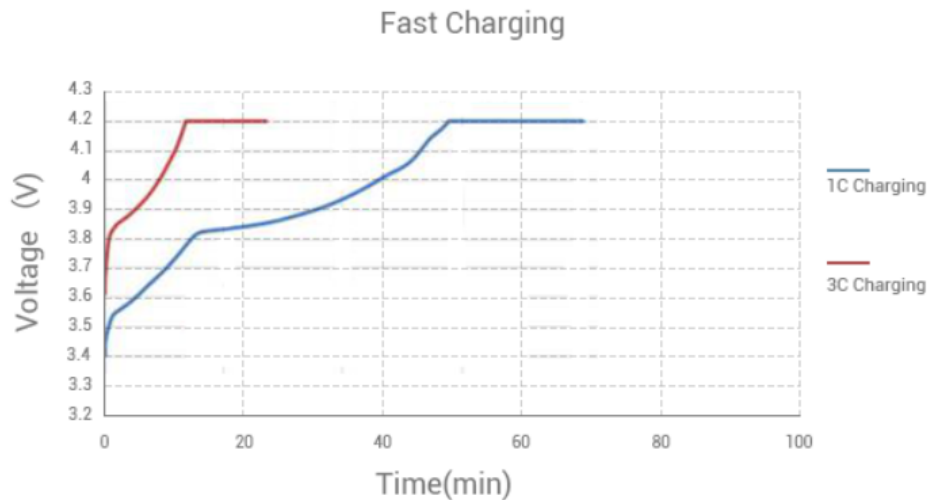


Figura 14. Curvas de carga de celdas LiPO. Tomado de [22]

Por otra parte, las curvas de carga indican el tiempo en el que aproximadamente las celdas se encuentran cargadas completamente, ésta al igual que en las curvas de descarga se puede cargar a diferentes capacidades, para una carga más rápida, de la misma manera, el proveedor especifica la máxima capacidad a la que debe ser cargada la batería.

Se debe destacar que, algunos de los fabricantes proveen por lo general las curvas de carga y descarga para una de sus celdas, sin embargo, no en todos los casos muestran este tipo de curvas. La ventaja de ésta es elegir una batería cuyas características se adecuen más a las necesidades del sistema. A pesar de que existen factores que afectan el desempeño de la batería como lo es el clima, obtener la curva de descarga es una manera más práctica de conocer realmente cuánto durará en operación a diferencia del cálculo ideal.

3.6.2 Convertidores de conmutación

Un convertidor de conmutación o regulador de voltaje conmutado es un dispositivo encargado de convertir un valor de DC dado a otro diferente o igual. A diferencia de los reguladores lineales, estos tienen una mayor eficiencia de conversión, porque utilizan elementos cuya potencia de disipación es cercana a cero, favoreciendo de esta manera que toda la potencia fluya hacia la carga.

Existen diferentes topologías sobre los convertidores conmutados como: Buck, Buck-Boost, Boost, cuck, etc., sin embargo, en el trabajo de investigación concierne retomar la topología Buck o de bajada, principalmente, por las cuestiones tratadas en el subcapítulo 3.6.1.1. Al requerir baterías con tensiones de voltaje superiores para los elementos montados sobre el dron, también hace precisar la presencia de un dispositivo capaz de

disminuir esos valores de tensión por uno adecuado para estos últimos, en donde, entran exactamente los reguladores de bajada.

3.6.2.1 Topología del convertidor Buck

El diagrama más sencillo de un convertidor se muestra en la Figura 15 donde los componentes más importantes son el interruptor controlado, el inductor y el capacitor en conjunto generan un nivel de tensión menor al de la fuente de entrada. El principio de funcionamiento del convertidor se basa en la carga y descarga del inductor cada vez que se cierra y abre el circuito por medio del interruptor, generando en la salida un voltaje promedio, lo anterior sería cierto del todo si se considerase el capacitor con un valor infinito, sin embargo, el capacitor tiene un valor finito, por lo que el voltaje de salida presenta fluctuaciones a la salida, denominado como rizo de voltaje alrededor del voltaje promedio de la salida.

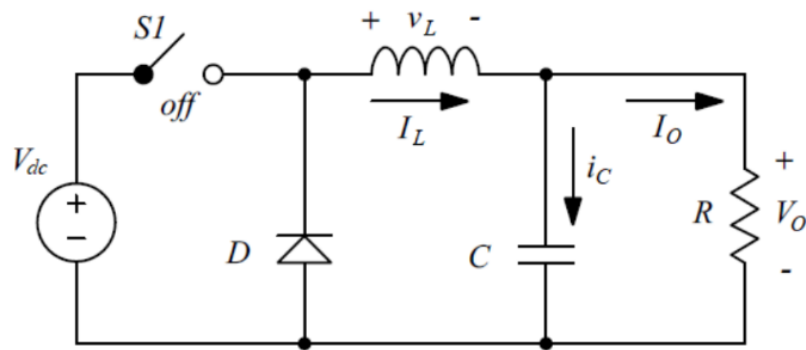


Figura 15. Diagrama de Convertidor Buck

El análisis, para este circuito se puede realizar por medio de observar la corriente y el voltaje en el capacitor e inductor con el interruptor abiertos y cerrado, no obstante, para no largar este trabajo, se dará en concreto algunas ecuaciones útiles para su diseño.

Voltaje de salida en el convertidor Buck

$$V_o = DV_{cd} \quad ec. (17)$$

Corriente promedio en el inductor y capacitor

$$I_c = 0 \quad ec. (18)$$

$$I_L = I_o = \frac{V_o}{R} \quad ec. (19)$$

Rizo de corriente

$$\Delta i_L^+ = \frac{(V_{dc} - V_o)}{L} DT \quad ec. (20)$$

Cálculo de la inductancia

$$L = \frac{(1 - D)V_o}{\Delta i_L} T \quad ec. (21)$$

Rizo de voltaje

$$\Delta V_o = \frac{\Delta i_L T}{8C} \quad ec. (22)$$

Cálculo de la capacitancia

$$C = \frac{\Delta i_L T}{8\Delta V_o} \quad ec. (23)$$

3.6.3 Medición de corriente y voltaje en dispositivos de DC

Realizar la medición de energía es algo muy común en la mayoría de los dispositivos electrónicos como computadoras, celulares, tabletas, etc. Uno de sus principales propósitos es proteger a la batería principal de alguna sobredescarga o sobrecarga de energía, impidiendo afectar el estado de la batería; y por otra parte la medición de energía permite conocer el presupuesto energético disponible para planear y tomar decisiones de tiempo de operación del sistema que se está alimentando. Con lo anterior en mente, los principales parámetros de medición en sistemas de DC resultan ser dos:

- Voltaje
- Corriente

La medición del voltaje en la batería comúnmente se realiza a través de un divisor de potencia (Figura 16), esto debido a que los ADC de los microcontroladores funcionan con voltajes de referencia menores a 5[V] o 3.3[V]. Este puede estar conformado por dos o más resistores dependiendo de los requerimientos del sistema y, un determinante en la selección de divisor se puede obtener por medio de la Ley de Ohm, especificando la corriente de operación del circuito, el voltaje de la batería y el voltaje de salida de la resistencia donde se tomará la medición.

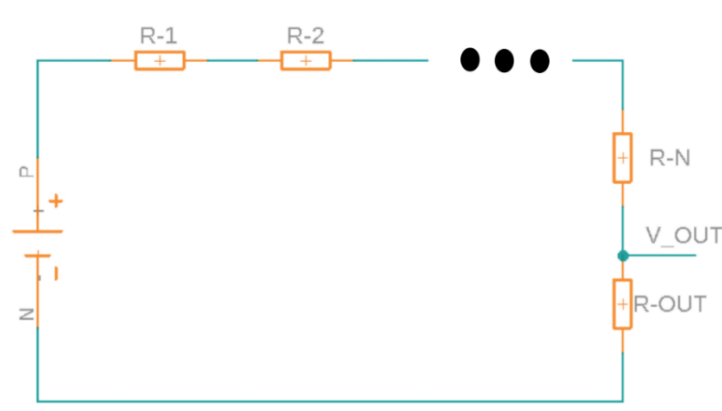


Figura 16. Divisor de tensión

De manera, sencilla, el divisor se modela con la siguiente ecuación.

$$V_o = \frac{R_{out}}{R_1 + R_2 + \dots + R_{n-1} + R_n + R_{out}} * V_i \quad ec. (24)$$

Donde:

- R_n es el enésimo resistor del divisor de potencia
- R_{out} es el resistor donde se toma la salida de voltaje adaptada para el ADC

Posteriormente, a este dispositivo se puede implementar un filtro paso-bajas, evitando la fluctuación debido al ruido generado por otros dispositivos o por el mismo circuito.

En el caso de los sensores de corriente, su medición se lleva a cabo de dos maneras:

- Efecto Hall:
- Resistencias Shunt:

Estos sensores de corriente al igual que lo de voltaje, requieren de acondicionamiento para generar la señal deseada en el ADC, en donde, se implementa un filtro paso-baja y un amplificador de señal, debido a la diferencia de potencial de la señal obtenida en estos dispositivos.

Asimismo, las resistencias shunt se pueden configurar de dos maneras en el lado alto del circuito (conectado a VCC) o del lado bajo del circuito (conectado a tierra), siendo la configuración de lado alto la más utilizada, principalmente por reducir las perturbaciones de tierra o como permitir la detección de condiciones de cortocircuito de la carga.

Finalmente, la configuración más adecuada y generalizada para realizar mediciones de energía en el sistema se muestra en la Figura 17. Esta se compone del divisor de potencia

con un filtro paso-bajas y, de un resistor shunt colocado en el lado alto del sistema, junto con su filtro y amplificador diferencial requerido.

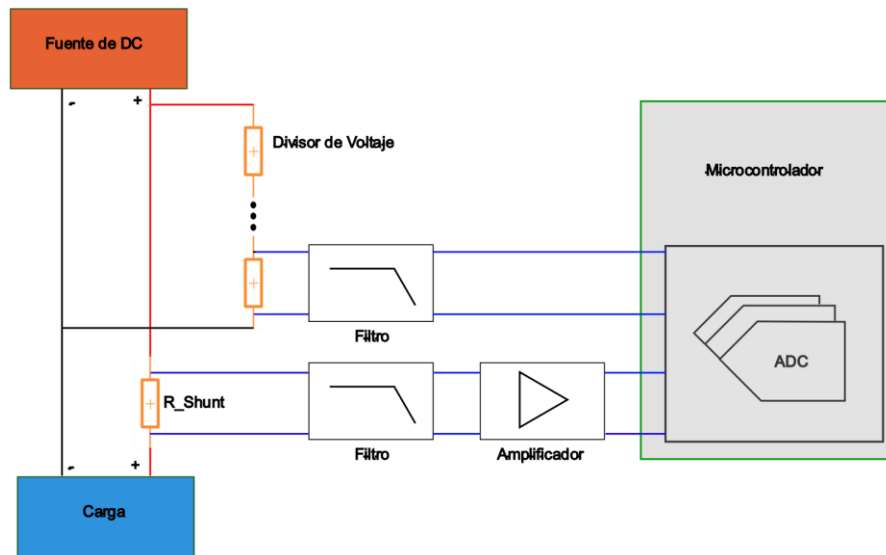


Figura 17. Medición de energía de DC

3.6.3.1 Filtro EMI

La interferencia electromagnética (EMI, por sus siglas en inglés), se refiere al ruido electromagnético que puede afectar en el correcto funcionamiento de los dispositivos electrónicos y, a su vez, puede ser generado por una gran variedad de diferentes motivos por estos mismos. Por ese motivo, es que se diseñan circuitos capaces de reducir la señal de interferencia dentro de la señal de interés, siendo estos conocidos como filtros.

Un filtro eléctrico en términos sencillos y de manera ideal, es un dispositivo que se encarga de rechazar o atenuar señales con un rango de frecuencias de acuerdo con una frecuencia de corte. En general existen 3 tipos básicos de filtros, siendo estos paso-bajas, paso-banda y paso-altas, estos a su vez pueden ser implementados por medio de componentes pasivos y activos, sin embargo, en este capítulo nos centraremos específicamente al filtro EMI, debido a su aplicación en la medición de voltaje y corriente en una fuente de DC.

Básicamente, un filtro EMI es un filtro paso-bajas cuya función es bloquear el flujo de ruido mientras pasa la entrada deseada que puede ser de DC. De manera general, este tipo de filtros se construye con elementos pasivos que son el capacitor e inductor. Una forma sencilla para seleccionar las características del filtro se muestra en [23], donde se rescata de la misma manera la Figura 18. para seleccionar la configuración de los elementos con respecto a las impedancias de entrada y salida.

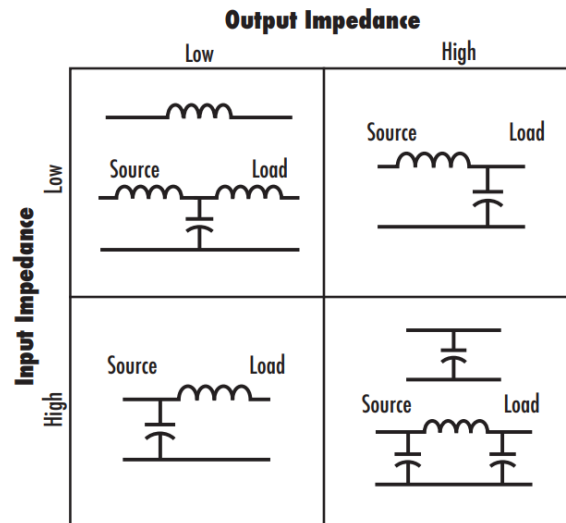


Figura 18. Configuración del filtro paso-bajas. Tomado de [23]

Para los filtros EMI, su frecuencia de corte se puede calcular para cualquier tipo de topología con la frecuencia de corte del filtro paso bajas, siendo esta:

$$f_{cutoff} = \frac{1}{2\pi\sqrt{LC}} \quad ec. (25)$$

Esta frecuencia a su vez puede cambiar dependiendo de otros elementos utilizados para mejorar el diseño del filtro.

3.6.3.2 Amplificador operacional

Un amplificador operacional (*OP AMP*, por sus siglas en inglés) es un dispositivo analógico que es diseñado para amplificar señales analógicas, entre otras varias aplicaciones, como filtros, buffer, comparadores, etc.

Estos amplificadores operacionales se pueden dividir en 3 categorías: bipolar op amps, JFET-input op amps y CMOS op amps. Sin embargo, en este apartado no se pretende profundizar en los op amp, sino que se desea dar herramientas en la selección del op amp a implementar en el circuito en diseño.

3.6.3.2.1 Amplificador operacional ideal y real

Una manera muy práctica de realizar el análisis y diseño de un amplificador operacional en sus diversas configuraciones existentes es por medio de considerar su modelo ideal mostrado en la Figura 19.

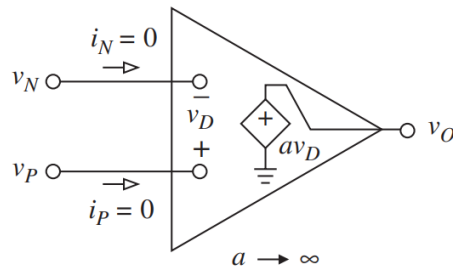


Figura 19. Modelo ideal de un OP AMP. Tomado de [24]

En este modelo se toma en consideración las siguientes asunciones en su rol:

- $a \rightarrow \infty, V_+ = V_-$, la ganancia de voltaje en lazo abierto es infinita
- $r_d = \infty$, la resistencia diferencial es infinita
- $r_o = 0$, la resistencia de salida es cero
- $I_P = I_N = 0$, la corriente en las terminales positiva y negativa es igual a cero.

Además del modelo ideal, se presenta un modelo más aproximado a la realidad del amplificador operacional, mostrado en la Figura 20.

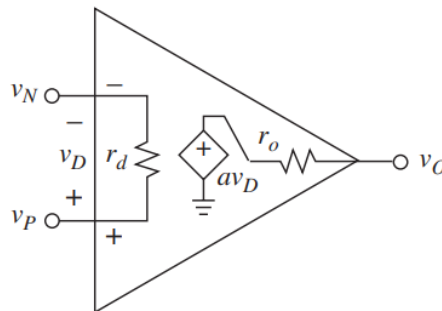


Figura 20. Modelo más realista del OP AMP. Tomado de [24]

Este modelo se utiliza al igual que el modelo ideal para realizar el análisis y diseño del amplificador operacional en diferentes configuraciones, sin embargo, se llega como resultado a que el modelo ideal y real dan un resultado muy similar, haciendo ver el modelo ideal como una buena opción de análisis.

3.6.3.2.2 Amplificadores de detección de corriente

Ahora bien, el principio de amplificador diferencial se utiliza para generar circuitos integrados para aplicaciones específicas como lo es en este caso, la medición de corriente. Este tipo de circuitos se conocen como amplificadores de medición de corriente. Y, de acuerdo con [25], se definen como “un amplificador diferencial de circuito integrado de

propósito especial que está diseñado para detectar el voltaje desarrollado a través de una derivación de corriente y generar un voltaje proporcional a la corriente medida”.

Las ventajas de este amplificador de propósito específico es que está diseñado para tener un CMRR alto para eliminar el voltaje generado por ruido, además de que muchos ofrecen el error de ganancia o el ruido de entrada, mientras que su única desventaja es que en algunos casos la ganancia del amplificador ya se encuentra definida y, de manera típica son ganancias de 20V/V, 50V/V, 100V/V, etc. Por lo que, si se deseara otro tipo de ganancia, esta no estaría disponible para implementar en estos integrados.

3.6.3.2.3 Amplificador diferencial

Como se mencionó, los amplificadores de detección de corriente están basados en el diseño de los amplificadores diferenciales, por ello, en este capítulo se presenta de manera breve las características de este tipo amplificador (Figura 21).

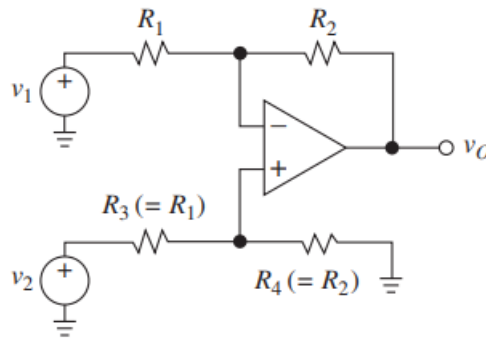


Figura 21. Amplificador diferencial. Tomado de [24]

La salida de voltaje para el circuito de la Figura 21 se define como:

$$V_o = \frac{R_2}{R_1} (V_2 - V_1) \quad ec. (26)$$

Asimismo, la resistencia de entrada y salida en la configuración de amplificador diferencial quedan de la siguiente forma:

$$R_{id} = 2R_1 \quad ec. (27)$$

$$R_o = 0 \quad ec. (28)$$

La ventaja de este circuito se encuentra al momento de agregar las entradas en modo diferencial y modo común en el amplificador operacional (Figura 22). Con ello se obtiene que para una entrada en modo común ($V_{CM} \neq 0$), independientemente de la señal diferencial, la amplificación en la salida será $V_o = 0$. Lo anterior en el caso del amplificador

ideal, sin embargo, en muchos amplificadores este parámetro es lo más cercano a cero dependiendo de sus especificaciones técnicas.

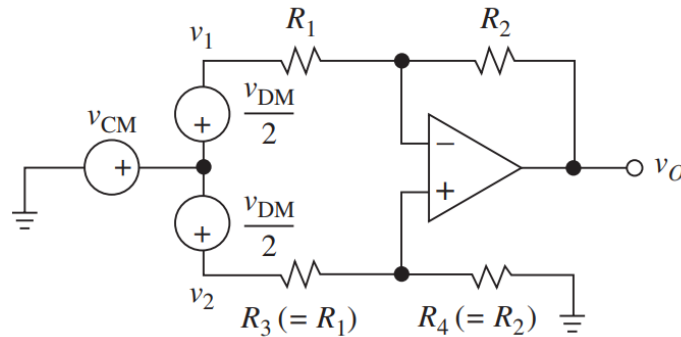


Figura 22. Expresando las entradas en términos de las componentes de modo común y modo diferencial. Tomado de [24]

Por otro lado, además de este amplificador diferencial en modo “single-ended⁷”, también existen los amplificadores completamente diferenciales, cuya salida está en modo diferencial. La única diferencia entre ambos es que los amplificadores completamente diferenciales tienen una resistencia inherente a las fuentes de ruidos externas, permitiendo una mayor precisión en la lectura de la señal deseada.

3.6.3.2.4 Parámetros de selección

Idealmente en el amplificador operacional existen muchos parámetros que no se consideran, no obstante, de manera práctica estos afectan mucho al diseño de nuestro circuito, por esa razón de manera rápida y sencilla se enlistan varios parámetros de selección en el diseño de este amplificador.

- Slew rate: Es la tasa máxima de cambio del voltaje de salida respecto al tiempo, medida en V/us.
- Ancho de banda: Es el margen de frecuencias donde opera el amplificador operacional, este factor depende mucho de la señal que se va a amplificar, por lo que ésta debe encajar en ese rango de frecuencias.
- Input Bias current: Este parámetro corresponde a la corriente de polarización circulando en las terminales de entrada del amplificador operacional, de manera ideal se consideraba que la corriente en las terminales era cero, sin embargo, en un caso práctico este valor es diferente de cero.
- Offset current: Este es la diferencia entre las corrientes Bias en las terminales positivas y negativas del amplificador operacional.

⁷ “Single-ended” hace referencia a la configuración de la señal, en este caso, la medición se realiza de la salida V_{out} a tierra común.

- CMRR: El ratio a rechazo del modo común es una capacidad del op amp para rechazar señales que son comunes para ambas entradas del dispositivo, de esta manera se evita amplificar señales de ruido no deseadas en la configuración deseada.
- Ruido: Este ruido es generado internamente por el op amp como voltaje y corriente, y se refleja en los pines de entrada como fuentes de voltaje y corriente conectados en paralelo. Además, del ruido generado por EMI, existen otras fuentes de ruido provenientes del mismo circuito, un caso muy común es el ruido térmico que está presente en todos los elementos resistivos pasivos, cuya definición se muestra a continuación.

$$V_{TH} = \sqrt{4kTBR} \quad ec.(29)$$

Donde:

- V_{TH} es el voltaje de ruido térmico
- k es la constante de Boltzman 1.38×10^{-23}
- T es la temperatura (°K)
- R es la resistencia
- B es el ancho de banda en Hz

Adicionalmente, existen soluciones embebidas, en este caso hablando de amplificadores de detección de corriente, las cuales incluyen el filtro para ruido EMI (filtro resistivo-capacitivo o filtro RC) junto con el amplificador diferencial. Este filtro se conecta a la entrada del amplificador de detección de corriente y sirve para disminuir el ruido EMI ocasionado por diferentes dispositivos, siendo este el caso de los reguladores de conmutación.

3.6.3.3 Filtros de entrada

A diferencia de los filtros EMI, los amplificadores operacionales usan como filtros de entrada filtros resistivos, que básicamente es un filtro paso-bajas, con una salida amortiguada. Su aplicación principal es disminuir el ruido generado en la señal diferencial, debido a los convertidores de conmutación.

Para el propósito de este trabajo se abordará únicamente la configuración del filtro en modo diferencial, el cual, es el más usado en este tipo de aplicaciones (Figura 23), cuya frecuencia de corte se muestra respectivamente.

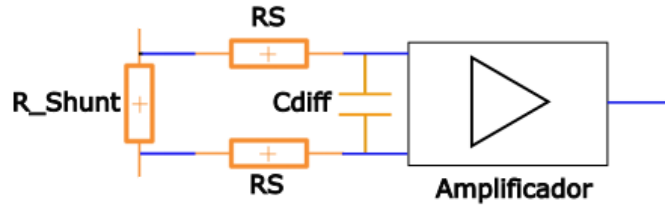


Figura 23. Filtro resistivo-capacitivo (RC) en modo diferencial

$$f = \frac{1}{2\pi(2R_S C_{diff})} \quad ec.(30)$$

La selección del filtro se debe realizar de manera correcta una vez caracterizada la señal deseada. Esto con la finalidad de introducir lo menos posible de ruido. En este trabajo se considera el filtro más básico mostrado, sin embargo, para más consideraciones del diseño, revisar [26] y [27].

3.7 Algoritmos de control

Los algoritmos de control establecen requerimientos importantes para el diseño de la computadora de abordó, en donde destaca principalmente la necesidad de recursos como tiempo de ejecución, memoria y consumo de energía. En cuanto mayor sea la eficiencia del algoritmo, menor serán los requerimientos de la computadora y se podrán ahorrar recursos como consumo energético y memoria requerida.

En este capítulo únicamente se abordarán los algoritmos comúnmente más utilizados en aplicaciones de robots móviles, dando un breve resumen y abordando los aspectos más importantes a implementar en cuestión de software. Gran parte de estos algoritmos fueron tomados de [28] y en este mismo se explican otros tipos de aspectos. Otros algoritmos más complejos y desarrollados para aplicaciones específicas se engloban en algunas otras referencias como [1], [2], [5], [6], [8], [11], [14], [29], [30] y [31].

3.7.1 Algoritmo de puntos de LIDAR (Representación poligonal)

Uno de los modelos tradicionales para la representación del ambiente es por medio de la representación de polígonos, usando los vértices de un polígono (Figura 24)

$$Polígono = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\} \quad ec.(31)$$

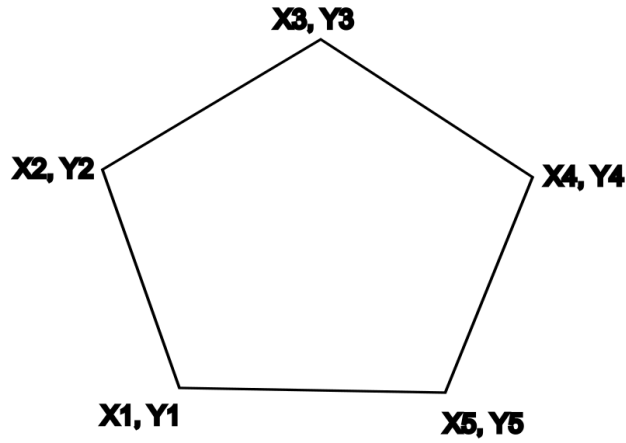


Figura 24. Ejemplo de representación poligonal

Con los vértices del polígono se encuentran las líneas rectas que lo unen (Figura 24). La ecuación general de estas líneas en forma paramétrica es:

$$y = y_0 + nt \quad ec. (32)$$

$$x = x_0 + mt \quad ec. (33)$$

Para los vértices (x_k, y_k) , (x_L, y_L) , se define su ecuación paramétrica, considerando a “t” como un parámetro que toma los valores de 0 y 1. Y, además, se adopta la convención que “t” toma el valor de $t = 0$ empezando en la orilla $K(x_k, y_k)$ y con $t = 1$ termina en la otra orilla $L(x_L, y_L)$.

Para que sea mayor entendible la obtención de la ecuación, se tienen que utilizar las ecuaciones anteriores. Entonces, se evalúa con $t = 0$ y 1 , para la ecuación paramétrica de y ,

$$y = y_k \text{ con } t = 0$$

$$y_k = y_0 + n \quad ec. (34)$$

$$y = y_L \text{ con } t = 1$$

$$y_L = y_k \quad ec. (35)$$

$$n = y_L - y_k \quad ec. (36)$$

Lo cual resulta en la siguiente ecuación,

$$y = y_k + (y_L - y_k)t \quad ec. (37)$$

Entonces las ecuaciones paramétricas que definen lo anterior se muestran a continuación:

$$y = y_k + (y_L - y_k)t \quad ec. (38)$$

$$x = x_k + (x_L - x_k)t \quad ec. (39)$$

Esta ecuación puede ser convertida en forma implícita, por medio de despejar la variable "t",

$$t = \frac{y - y_k}{(y_L - y_k)} \quad ec. (40)$$

$$t = \frac{x - x_k}{(x_L - x_k)} \quad ec. (41)$$

Igualando las ecuaciones,

$$\begin{aligned} \frac{y - y_k}{(y_L - y_k)} &= \frac{x - x_k}{(x_L - x_k)} \\ (y - y_k)(x_L - x_k) &= (x - x_k)(y_L - y_k) \\ (y_k - y_L)x + (x_L - x_k)y - y_k(x_L - x_k) + x_k(y_L - y_k) &= 0 \\ (y_k - y_L)x + (x_L - x_k)y - y_k(x_L) + x_k(y_L) &= 0 \end{aligned} \quad ec. (42)$$

Por tanto, la función implícita, en donde los puntos (x, y) estarán sobre la línea, siempre y cuando se cumpla $f(x, y) = 0$, se muestra respectivamente a continuación.

$$f(x, y) = (y_k - y_L)x + (x_L - x_k)y + x_k y_L - x_L y_k \quad ec. (43)$$

Por lo tanto, para saber si un punto está dentro del espacio ocupado que define el polígono, con sus vértices ordenados con las manecillas del reloj, se debe cumplir que para todas las líneas formadas por los pares

$$\left(((x_1, y_1), (x_2, y_2)), ((x_2, y_2), (x_3, y_3)) \dots, ((x_n, y_n), (x_1, y_1)) \right)$$

$$f(x, y) < 0$$

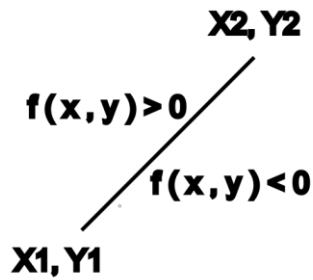


Figura 25. Línea recta formada por los vértices del polígono.

Esto se puede analizar de manera sencilla, con la función implícita que define una de las líneas del polígono (Figura 25), y dada la convención tomada para formar el polígono (Figura 24). Entonces, si sabemos que con $f(x,y) = 0$ estamos en el contorno del polígono, por ende, un valor $f(x,y) < 0$ indicaría que el punto seleccionado está dentro del polígono y, un valor $f(x,y) > 0$ implicaría que el punto está fuera del polígono analizado.

2.7.1.1 Convex Hull

El término Convex Hull hace referencia a la envolvente convexa que se genera por la unión de diferentes puntos que limitan una figura poligonal, justo como se muestra en la siguiente Figura 26.

La manera más sencilla de identificar los puntos que conforman la envolvente es haciendo uso de los ángulos formado por las dos rectas generadas con 3 puntos. De esta manera, se compara si el punto actual hace girar la recta a la izquierda o a la derecha y, con esta información, se puede determinar si el punto forma parte de la envolvente o no.

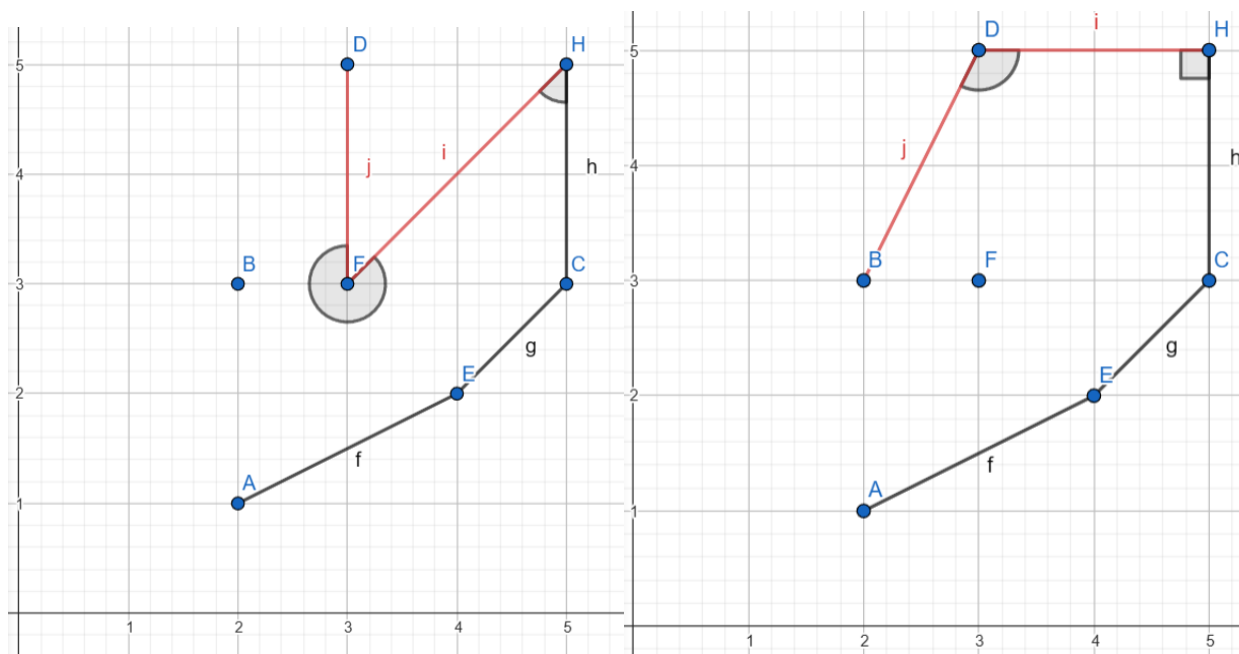


Figura 26. (a) Giro a la derecha, (b) Giro a la izquierda

En la Figura 26, los segmentos de recta $\overline{AE}$, $\overline{EC}$ y $\overline{CH}$, se pueden considerar como si fuesen una sola recta rotada cierto ángulo para coincidir con los puntos de la envolvente. Entonces, al considerar las rectas $\overline{HF}$ y $\overline{FD}$ en (a), se genera un cambio en la recta haciéndola dar un giro hacia a la derecha. Por tanto, este punto no pertenece a la envolvente y se analizan otros puntos. Finalmente, en la Figura (b) se observa que con las rectas $\overline{HD}$ y $\overline{DB}$, se mantiene un giro a la izquierda del segmento de recta correspondiente.

3.7.2 Algoritmo RANSAC

RANSAC, por sus siglas en inglés, es un algoritmo muy popular en la segmentación de datos con LIDAR, de acuerdo con [32]. El objetivo de este algoritmo es determinar valores pertenecientes a un grupo en común “inliers”, con lo cual, se puede obtener un modelo matemático capaz de describir la forma del grupo de datos.

La manera más sencilla para abordar este tipo de problema es considerando que queremos modelar los grupos de puntos obtenidos con una línea recta, justo como en la Figura 27.

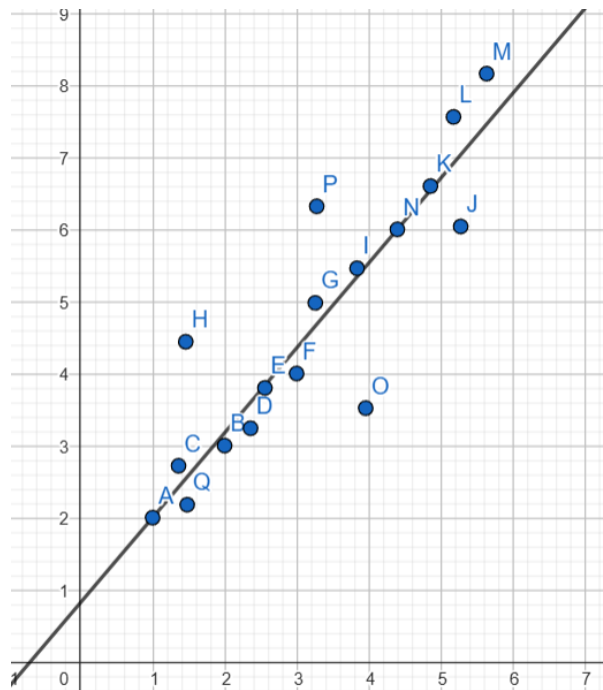


Figura 27. Puntos de datos modelados en una recta

En esta imagen, se observa que el modelo matemático en forma de línea recta que mejor se adapta a los puntos, es el de la recta formada por el punto A y N. Por tanto, con el algoritmo de RANSAC, se obtendría un modelo matemático que describe la forma de los puntos obtenidos por LIDAR. Esto mismo se puede aplicar para otros tipos de modelos matemáticos, como, por ejemplo, polígonos, círculos, triángulos, etc.

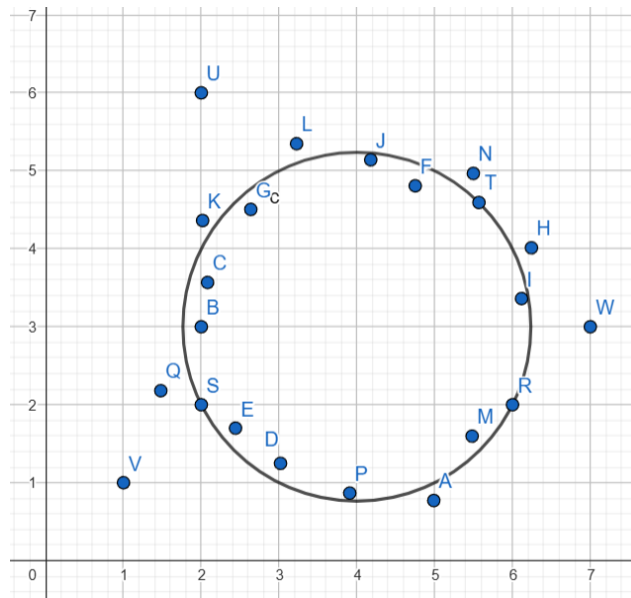


Figura 28. Puntos de datos modelados en un círculo

El algoritmo de RANSAC, resulta sencillo de implementar, lo único que puede diferir es la manera en que se generan los diferentes modelos matemáticos por medio de los puntos de datos obtenidos por LIDAR. Este algoritmo se describe a continuación y se puede observar más información en [33].

1. Definir los parámetros de RANSAC, descritos debajo:
 - p : Es la probabilidad de que al menos uno de los conjuntos de las muestras aleatorias no incluya un valor atípico (Se define entre los valores 0 a 1, entre más cercano a 1, es mejor la aproximación).
 - N : Es la cantidad de iteraciones máximo que debe ejecutar el algoritmo (al inicio se coloca un valor muy grande $N \rightarrow \infty$).
 - Threshold: Se define un límite de distancia a la cual debe encontrarse un punto del modelo escogido.
 - n : El mínimo número de puntos que se necesitan para considerar válido el modelo. Este valor es por lo general, menor a la cantidad de números de datos totales.
2. Se define un modelo para describir al conjunto de puntos (círculo, recta, cuadrilátero, triángulo, etc.).
3. Del conjunto de puntos total se selecciona la cantidad mínima de puntos necesarios para formar el modelo matemático, por ejemplo, para generar un círculo se requieren mínimo 3 puntos diferentes (estos puntos deben ser escogidos al azar).
4. Obtener todos los puntos que coinciden con el modelo matemático y dentro del límite establecido.
5. Si, la cantidad de puntos del nuevo modelo es mayor a un modelo anterior,
 - Guardar el nuevo modelo y actualizar variables,
 - Calcular el nuevo valor de iteración de acuerdo con la siguiente ecuación, donde,
 - ❖ ω^n : es la probabilidad de que los n puntos son puntos pertenecientes al modelo "inliers"

$$N = \frac{\log(1 - p)}{\log(1 - \omega^n)} \quad \text{ec. (44)}$$

$$\omega = \frac{\text{Número de "inliers" en el modelo actual}}{\text{Número de puntos totales de los datos}} \quad \text{ec. (45)}$$

6. Incrementar el número de iteraciones actual.
7. Repetir del paso 3 a 5, hasta que se alcance el número máximo de iteraciones N permitido.

3.7.3 Algoritmo DBSCAN

DBSCAN, por sus siglas en inglés, es un algoritmo utilizado en la agrupación de puntos, permitiendo diferenciar un objeto de otro. Este algoritmo se tratará de manera básica y para más información se puede revisar [34].

Este algoritmo funciona de manera sencilla, por medio de delimitar una distancia mínima entre un punto central y los demás, y la cantidad mínima que debe contener un punto central para ser considerado un “núcleo”. Con lo anterior se puede diferenciar puntos entre “núcleos” y “bordes”, permitiendo identificar uno o varios puntos formando un objeto.

Por ejemplo, en la Figura 29, se muestran varios puntos. Donde los puntos A, C y K, conforman los núcleos mencionados anteriormente, y donde, los núcleos A y C se sobreponen. Para ello, se definió un radio de 1[unidad] y un mínimo de 4 puntos. Entonces, conociendo los núcleos y los bordes definidos por los demás puntos, se puede saber que tenemos 3 agrupaciones de puntos distintas.

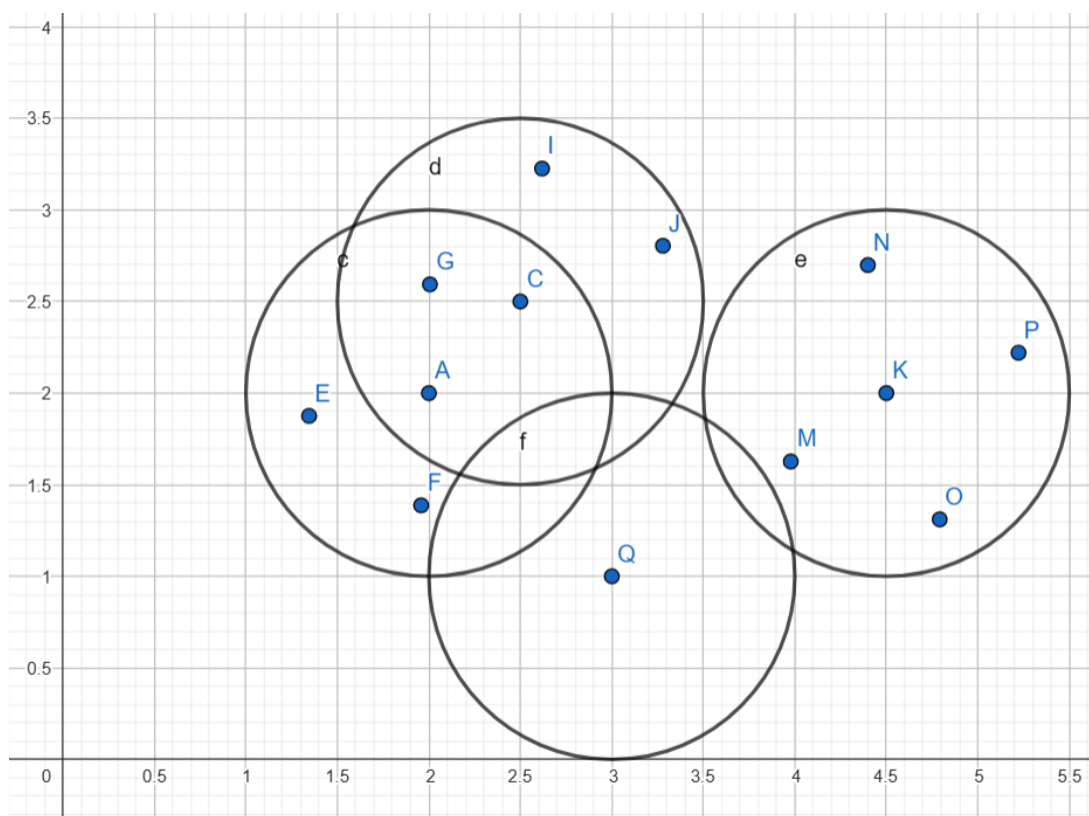


Figura 29. Ejemplo de agrupación, utilizando DBSCAN.

3.7.4 Algoritmo Voronoi

Considérese un punto $\langle x, y \rangle \in C$ en el espacio libre. Los puntos base de $\langle x, y \rangle$ son los puntos $\langle x_0', y_0' \rangle$ más cercanos en el espacio ocupado $\bar{C}$.

Los diagramas de Voronoi encuentran una separación del espacio tomando como referencia los centroides de los objetos, encontrando líneas a la mitad de distancia entre estos. Además, estas líneas dejan un margen para que el robot pueda circular entre ellos sin chocar.

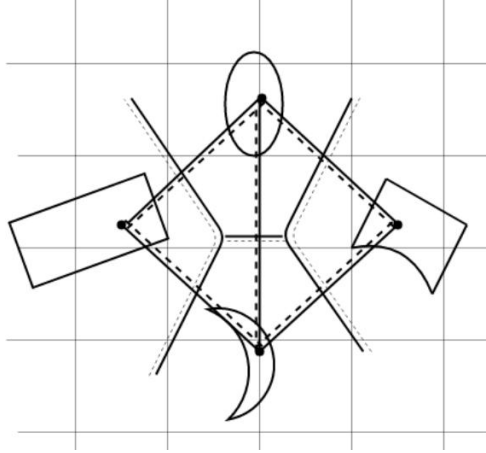


Figura 30. Ejemplo de Diagrama de Voronoi. Tomado de [28]

El diagrama de Voronoi es el conjunto de puntos en el espacio libre que tiene cuando menos dos puntos base diferentes a la misma distancia y que se encuentran separados por una δ .

$$|(x'_1, y'_1) - (x, y)| = |(x'_2, y'_2) - (x, y)| = d \quad \text{ec. (46)}$$

$$|(x'_1, y'_1) - (x'_2, y'_2)| < \delta \quad \text{ec. (47)}$$

El conjunto de puntos (x_i, y_i) que tienen dos puntos base diferentes forman el diagrama de Voronoi como se muestra en la Figura 31:

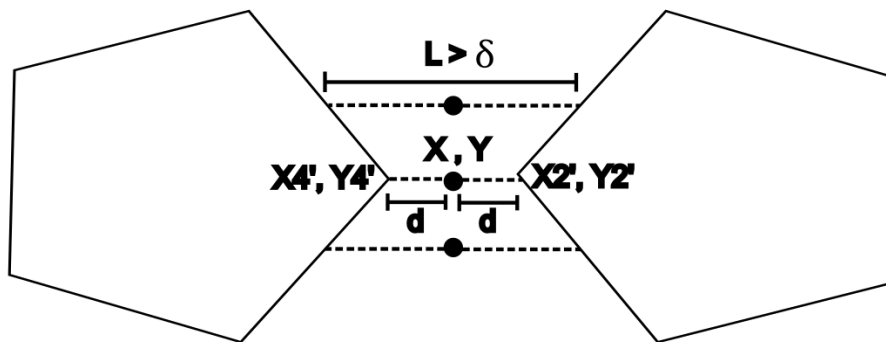


Figura 31. Ejemplo de Diagrama Voronoi con dos objetos.

En la Figura 31, se observa cómo el punto (x, y) se forma por los puntos base (x'_4, y'_4) y (x'_2, y'_2) . Estos, además de estar a la misma distancia “d”, deben cumplir con un valor de δ capaz de permitir el paso del robot por ese punto.

Los puntos críticos $\langle x, y \rangle$ son puntos en el diagrama de Voronoi que minimiza el margen localmente: Existe una $\epsilon > 0$ para la cual el margen de todos los puntos en el vecindario de $\langle x, y \rangle$ no es menor. Este postulado se utiliza únicamente para el mapeo de zonas, separando zonas de diferente tamaño.

3.7.4.1 Algoritmo de Fortune

Una manera de implementar un algoritmo eficiente de Voronoi, es por medio del algoritmo Fortune, explicado a mayor detalle en [35]. En términos sencillos, este algoritmo utiliza ciertos valores de intersecciones generadas entre parábolas, cuyos parámetros se definen por medio de los puntos de centroides de los objetos encontrados y, por medio de una directriz común para todas las parábolas.

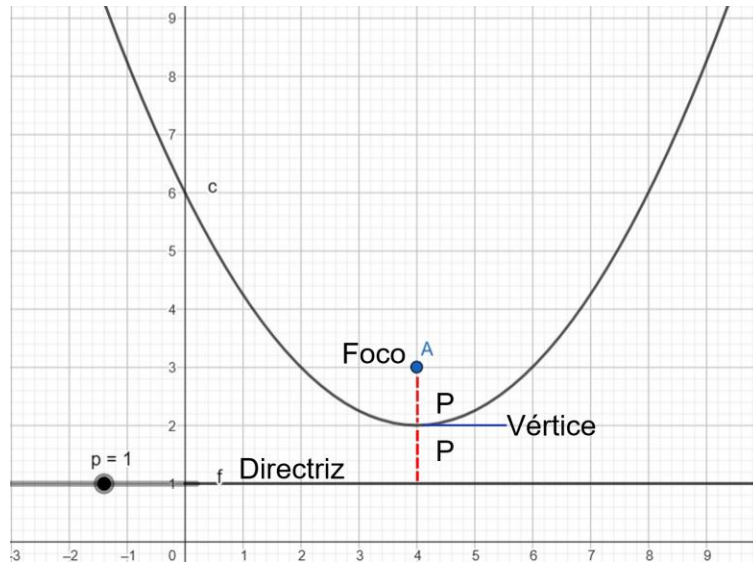


Figura 32. Parámetros de una parábola

Entonces, conociendo la ecuación canónica de una parábola, la directriz y el foco de ésta, se pueden obtener los parámetros que describen cada una de las parábolas.

$$(x - x_1)^2 = 4p(y - y_1) \quad \text{ec. (48)}$$

$$y = \frac{1}{4p}x^2 - \frac{2x_1}{4p}x + \frac{x_1^2}{4p} + y_1 \quad \text{ec. (49)}$$

Donde, estas ecuaciones son de la forma

$$y = Ax^2 + Bx + C \quad \text{ec. (50)}$$

Finalmente, al generar las parábolas, se van verificando dos tipos de eventos conocidos como eventos de sitio y circular. El evento de sitio se encarga de ir añadiendo las parábolas conforme se van encontrando por medio de la directriz; mientras que el evento circular ocurre con tres puntos, los cuales forman un círculo que es tangente a la directriz común. Es con estos eventos con los que se van definiendo los puntos de Voronoi que generan rutas entre los objetos.

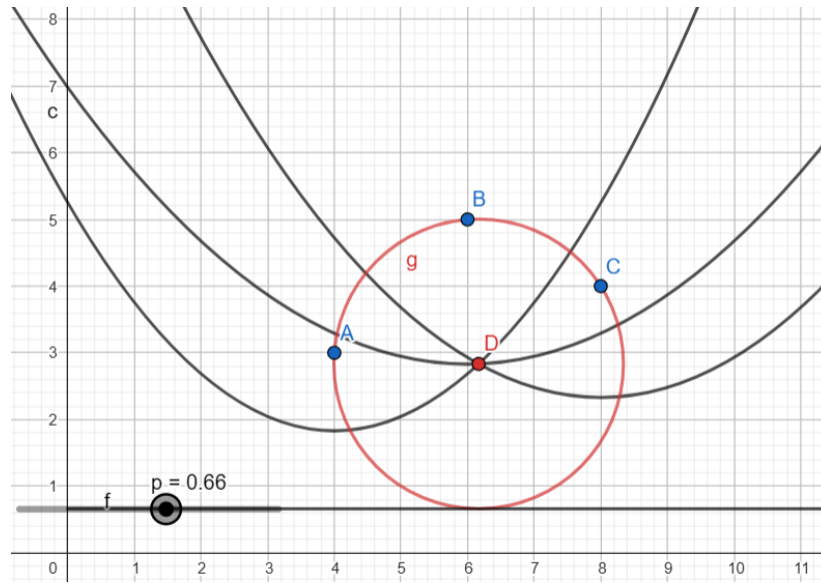


Figura 33. Ejemplo de un evento circular

3.7.5 Algoritmo Dijkstra

El algoritmo de Dijkstra funciona en etapas. En cada etapa selecciona el menor de los vértices etiquetados como desconocidos, y declara el camino más corto del nodo s al nodo v desconocido. La parte final de cada etapa es actualizar los valores de la distancia acumulada d_j , de los vértices adyacentes de v , si $d_j = \infty$ entonces $d_w = d_v + C_v$, para el nodo adyacente w .

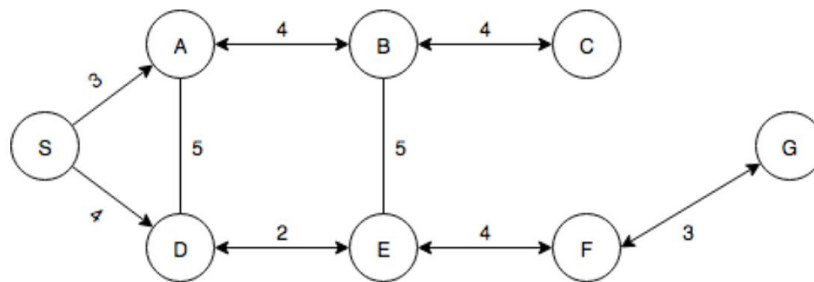


Figura 34. Ejemplo de planeación de ruta. Obtenido de [28]

Por ejemplo, en la Figura 34. Se desea ir del punto S al punto G, para ello se considera que en un inicio el valor acumulado de costos para cada punto que no se encuentre directamente relacionado con el inicio S es ∞ . Cuando se hace la primera iteración, se consideran los costos de S-A, S-D y S-A-D, donde el valor de menor costo es el de S-A. Posteriormente, se sigue iterando, y se comparan los costos de las rutas posibles, llegando a la solución de la ruta con menor costo y la ruta a tomar, que en el ejemplo corresponde a la ruta S-D-E-F-G.

Finalmente, otra forma de ver de entender más fácil el ejemplo es a través de considerar que el algoritmo se utiliza para la obtención de la ruta más corta, donde los costos corresponden de forma sencilla a la distancia entre cada punto.

3.7.6 Aplicación de Algoritmo de navegación con GNSS

Además de los algoritmos anteriores, se requiere un algoritmo de navegación a través de dos puntos de referencia. En un principio el dron recibe las coordenadas de referencia de inicio vistas desde el plano como (x_0, y_0, z_0) y, a su vez, recibe las coordenadas de destino (x_d, y_d) . De manera simplificada la coordenada z_0 se mantiene constante y las únicas coordenadas que variarán en el tiempo serán (x_t, y_t) .

Obviamente, la ruta más corta al destino es una línea recta entre los puntos (x_0, y_0) y (x_d, y_d) , no obstante, al hacer el muestreo del ambiente, irán apareciendo obstáculos que cambien el rumbo de dirección, este rumbo se escogerá de acuerdo con el criterio del algoritmo de Dijkstra, donde cada cierto tiempo t , se comparará la posición del GNSS definida como (x'_t, y'_t) y la posición teórica (x_t, y_t) (obtenida del diagrama de Voronoi) para evitar que el dron se salga de su curso y se mantenga siguiendo una ruta consistente, a través de definir un porcentaje de error estipulado. Esto, se ejemplifica mejor con la Figura 35.

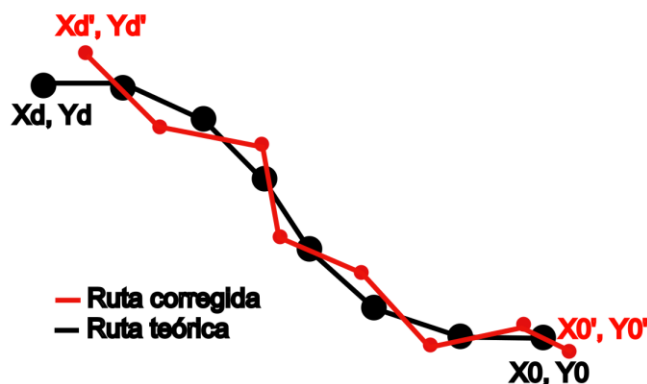


Figura 35. Representación de la ruta teórica y la ruta corregida con ayuda del GNSS

Para mejorar el desempeño y rendimiento del dron repartidor, es indispensable que el sistema conozca de manera previa a su operación el terreno donde volará, esto ayudará a poder generar las mejores rutas óptimas y con el sensor de distancia LIDAR, se podrá asegurar y prever mejor la detección y evasión de los obstáculos.

4. Diseño

En capítulos previos se dio un panorama muy general del alcance, las limitaciones y la tecnología implementada en los UAV. Como se ha analizado, los drones como repartidores contienen una gran variedad de sensores y dispositivos abordo, necesarios para su operación autónoma, atendiendo cada uno de ellos a los estándares requeridos por el UAV.

En este marco, realizar el diseño conceptual del sistema proporcionará una solución al problema planteado. Para ello se comienza con el planteamiento de los requerimientos y especificaciones indispensables en la instrumentación a bordo del dron. Una vez desarrollados esos puntos se dará una propuesta del diagrama funcional del sistema.

Finalmente, con esta propuesta el siguiente paso es el diseño de detalle de cada uno de los subsistemas integrados en el dron repartidor. El diseño de cada subsistema permitirá conocer la capacidad de procesamiento con la que deberá contar nuestro computador de abordo para realizar las diferentes tareas.

4.1 Diseño del concepto

4.1.1 Requerimientos

Los requerimientos son las cualidades y características esenciales con las que contará el dron. Estas están escritas en un lenguaje no técnico y son las bases en la solución del problema planteado.

Primero se comienza delimitando los requerimientos generales del producto, en este caso, el sistema completo de reparto:

- El peso máximo del paquete a soportar es de hasta 2.0kg.
- El área y la altura de vuelo se delimitarán de acuerdo con la NOM-107-SCT3-2019. Para ello se define una velocidad de hasta 20[m/s] y un tiempo de operación de 10 [min], logrando un radio de cobertura máximo de 6[km]⁸. Y, la altura de operación será hasta de 50[m].
- La estación será la encargada de recibir las solicitudes de pedido y, a partir de esto, se alistará a un dron para realizarla. De la misma manera, se podrá recibir información y enviar comandos del dron a la estación.
- Las entregas por cuestiones de seguridad se llevarán a cabo asignando un paquete por dron. Asimismo, se deben conocer las condiciones ambientales para el despliegue del dron.

⁸ El radio de cobertura máximo se calcula considerando que el dron repartidor realizará el viaje de ida y regreso antes de que la batería llegue a su capacidad mínima.

Por su parte, los requerimientos de la instrumentación de abordó del dron se enumeran de acuerdo con su importancia, iniciando por el más importante, de la siguiente manera:

1. La computadora de abordó tiene que ser capaz de proporcionar el procesamiento y manejo de información necesario por parte del sistema. Entre eso, la computadora principal se encargará de realizar tareas como generar la guía del dron, evitar obstáculos, entre otras.
2. Los sensores por utilizar proporcionarán la información necesaria del ambiente para detectar objetos.
3. El enlace de conexión puede ser a través de wifi, haciendo uso de internet de las cosas (IoT, por sus siglas en inglés). Este enlace debe contar con baja latencia para poder realizar control por piloto manual.
4. El GPS debe proporcionar información precisa para la localización y guía del dron repartidor.
5. El controlador de vuelo se encargará de dar la dirección para el movimiento del dron. Esto lo deberá hacer de acuerdo con las instrucciones proporcionadas por la computadora de abordó.
6. El sistema de potencia proporcionará la energía suficiente al dron, con la finalidad de mantenerse en operación hasta cumplir con la entrega.

4.1.2 Diagrama en bloques del sistema

Una vez analizados los requerimientos del sistema, la propuesta de la arquitectura del dron se muestra por medio de un diagrama de bloques. El diagrama de bloques mostrado en la Figura 36 es una primera aproximación y, posibilita la visualización de los subsistemas en el dron.

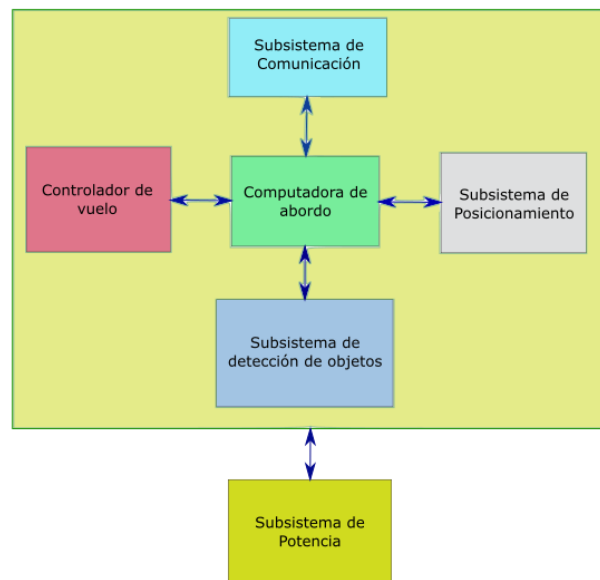


Figura 36. Diagrama a bloques del concepto.

Por el momento este diagrama únicamente presenta los subsistemas requeridos por el dron repartidor. En este se distingue a la computadora de abordo como el subsistema esencial del diseño, lo cual encaja conforme a lo comentado en capítulos anteriores.

4.2 Diseño a nivel sistema

En el diseño a nivel sistema se definen las especificaciones funcionales de cada uno de los subsistemas establecidos en el primer diagrama a bloques. Esta información es de gran relevancia porque posibilita dividir el diseño del sistema en varias tareas más concretas.

A manera de facilitar el análisis de las especificaciones funcionales de cada subsistema, se presenta la Tabla 12, donde se describen las entradas/salidas, la función y/o especificación de estos.

Tabla 12. Especificaciones funcionales de cada subsistema.

Subsistema	Entradas	Salidas	Funciones y/o especificaciones
Comunicación	-Comandos de la estación	-Telemetría y/o vídeo	-Ancho de banda suficiente para datos (Mensaje de texto, etc) -Conexión a red celular, para soluciones con IoT -Latencia menor a 1[s]
Detección de objetos	-Instrucciones a fin de medir el entorno	-Información del entorno	-Detectar objetos a una distancia mínima de 8[m] -Campo de visión vertical mínimo de 30° -Campo de visión horizontal dependerá del tamaño del UAV
Posicionamiento	-Instrucciones para mediciones -Señales satelitales	-Información de posición	-El GPS proporcionará una precisión de hasta 2.5[m], por medio de una conexión estable -El altímetro permitirá realizar mediciones de hasta 150[m]
Computadora de abordo	-Dirección de movimiento del FC -Comandos del subsistema de comunicación -Posición del GPS y altímetro -Posición de posibles obstáculos -Estado de la batería	-Datos procesados al subsistema de comunicación -Instrucciones para mediciones -Comandos de guía hacia al FC	-Procesa la información de los subsistemas e interpreta -Algoritmo para guía del UAV -Algoritmo de Dijkstra -Interpretación del espacio medido por sensores -Permite almacenaje de datos con una uSD -Debe contar con los puertos E/S necesarios

Control de vuelo	-Comandos del piloto automático o manual	-Señales para el control de los motores	-Permite controlar la dirección del dron -Debe permitir la conexión con una computadora de abordo
Potencia	-	-Suministro de energía para cada subsistema y actuadores -Información del estado de la batería	-Regula la tensión a 3.3 y 5[V] con un error menor o igual al 1%. -Alimenta a los motores brushless -Para mejorar el consumo de energía se recomienda usar reguladores con alta eficiencia.

En ese sentido, la computadora de abordo es el subsistema principal del dron, debido a su comunicación con todos los dispositivos dentro del dron. Su diseño debe atender a las necesidades de procesamiento abordo y tiempo de operación.

En un segundo punto se encuentran los sensores junto con el GPS, capaces de otorgar información precisa a la computadora de abordo. Todos estos elementos en conjunto, además de localizar objetos y ayudar en la navegación del curso del dron, también contribuyen en la toma de decisiones como, por ejemplo, saber si alguno de estos objetos se considera un obstáculo potencial en el camino del dron e incluso indicar el lugar donde se entregará la mercancía de manera segura y correcta.

Por otra parte, el sistema de comunicación se define de acuerdo con las necesidades de transmisión por parte del dron. Por lo general, la transmisión de imágenes o vídeo, ocupan la mayoría del ancho de banda requerido por el sistema. Asimismo, en caso de esperar una respuesta rápida por el sistema, se deberá considerar la latencia del subsistema.

Finalmente, la capacidad del sistema de potencia debe ser diseñado tomando en cuenta el consumo de cada dispositivo. Anteriormente se mencionó que las baterías de polímeros de litio son las principales usadas en este sistema, sin embargo, para tratar de expandir el tiempo de vuelo, es posible considerar otras alternativas.

4.2.1 Diagrama funcional

Considerando las características expuestas en la Tabla 12, a continuación, se presenta en la Figura 37 el diagrama funcional del sistema, marcando de forma concreta cada una de las actividades desempeñadas dentro del dron.

En este diagrama se muestra cómo la solución al problema es por medio del uso del concepto de IoT, permitiendo de manera general el control a través de dispositivos móviles. Además, se hace distinción por medio del color azul sobre los dispositivos

añadidos a los componentes básicos de un dron, siendo estos el controlador de vuelo, los controladores de velocidad y los motores brushless.

Por otra parte, en este se hace notar cómo toda la información de los demás subsistemas es recibida, procesada y después enviada (en su caso) a otros subsistemas, todo esto a través de la computadora de a bordo. Asimismo, ésta se encarga de permitir el control del dron de manera manual por medio del celular, para esto es necesario un sistema de comunicación con una latencia menor a 1[s], logrando proveer de una buena respuesta en el control manual del dron.

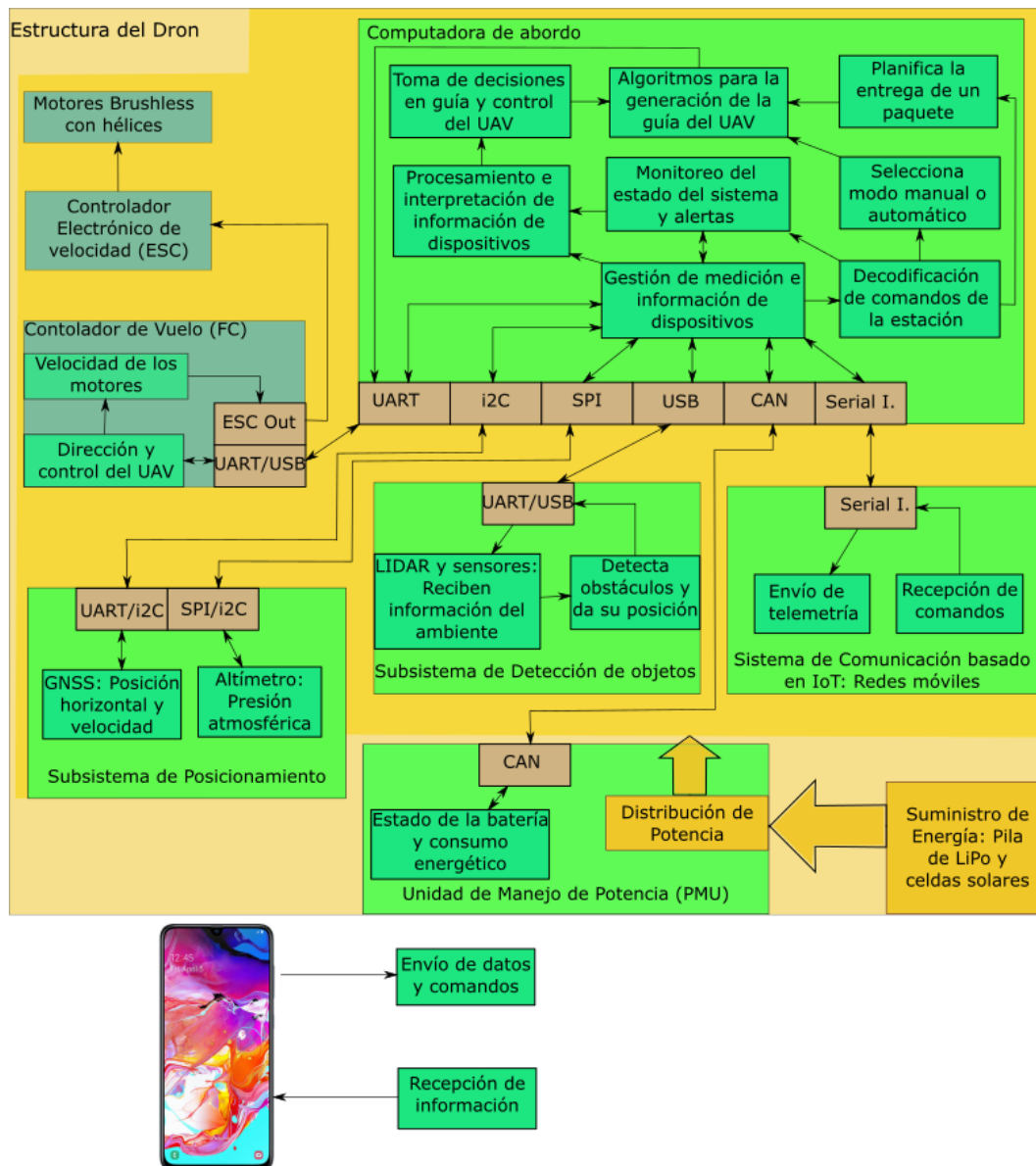


Figura 37. Diagrama funcional a bloques del sistema.

Por otro lado, LIDAR junto con otros sensores se encarga de dar la información necesaria del ambiente para evadir obstáculos y tomar decisiones importantes como el despliegue de la mercancía. En este punto, el altímetro se encarga de dar información oportuna para no sobrepasar el límite de establecido por la NOM. Además, GPS que no es un sensor tal cual, se encarga de proporcionar la localización del dron para su guía y, a su vez, otorga de una buena precisión necesaria en la entrega del paquete.

Como último punto en el diagrama, el subsistema de potencia conformado por la batería de litio y las celdas solares suministran la energía al sistema de manera general. En este también se integra una unidad de manejo de potencia capaz de regular el voltaje necesario para dispositivos con un menor valor de voltaje. Un punto para considerar sobre esto es que las celdas solares por su baja generación de corriente no alimentarán directamente a los motores brushless, sino que proveerán energía a otros subsistemas de menor consumo por medio de aprovechar la irradiancia en México⁹.

Consecuentemente, se presenta de manera más detallada la descripción de cada uno de los subsistemas en los subsecuentes subcapítulos. Estos contienen información específica referente a cómo operar y desempeñar puntualmente cada una de las tareas asignadas en cada uno de ellos.

4.2.2 Estructura

En la Figura 37, se muestra cómo la estructura es la encargada de soportar todos los dispositivos electrónicos utilizados en la misión del dron. En cuanto a requerimientos, esta estructura debe ser de un material rígido y contar con el espacio suficiente para colocar todos los dispositivos requeridos.

Debido a las características del peso de la mercancía a transportar (2kg), se puede hablar de estructuras con dimensiones mayores a muchos drones comerciales recreativos como por ejemplo los drones DJI phantom. Con esto en mente, la estructura debe contar con una placa central de dimensiones mayores a 20cmx20cm, o si cuenta con doble compartimiento, las medidas pueden ser de hasta 15cmx15cm, proporcionando espacio suficiente para el FC, la computadora de abord, el dispositivo de comunicación y otros sensores.

Por otro lado, la distancia de separación entre motores requerida debe contemplar el espacio suficiente para que las hélices no choquen entre sí. Estas hélices en el caso de levantar pesos mayores a 4 kg o más, tienden a aumentar su diámetro considerablemente, por lo cual, se requiere una estructura con una distancia entre ejes diagonal mayor a

⁹ Las celdas se pueden implementar o no, dependiendo de la corriente demandada por los motores y el peso del dron.

450mm, entre mayor sea ésta, mayor será la separación entre cada motor y, por ende, se podrán colocar hélices con diámetros superiores.

Finalmente, se debe considerar un mecanismo para liberar o entregar el paquete cuando se le indique al dron repartidor, éste principalmente debe atender cuestiones de seguridad como evitar estar tan cerca del usuario final.

4.2.3 Subsistema de comunicación

El sistema de comunicación por medio de redes móviles propicia el intercambio de información entre el dron repartidor y la estación de control. Este sistema operará directamente con la estación base a través de comandos capaces de permitir ejecutar las siguientes acciones:

- Permitir la planificación y modificación de una ruta
- Realizar cambio entre piloto automático y manual, y viceversa, para el control del dron por medio de IoT.
- Ajustar parámetros sobre el dron como velocidad, altitud y gestión de energía.
- Solicitar y enviar telemetría básica sobre el estado del dron.
- Envío de comandos para el control en piloto manual del dron.
- Transmisión de datos de LIDAR para visualización del entorno.

De manera básica, nuestro subsistema de comunicación envía y recibe datos. Estos datos pueden ser desde comandos de corta longitud como mensajes de texto, hasta un ancho de banda suficiente para transmisión de la información de LIDAR o de vídeo. Además, como se mencionó, debido a la latencia necesaria tanto en la operación manual del dron como en la transmisión de vídeo, es necesario que el subsistema pueda transmitir el vídeo en tiempo real al mismo tiempo que recibe los comandos para moverse en diferentes direcciones.

Para comunicar al subsistema con el sistema en general, se necesitará de una conexión, comúnmente serial, a través de la computadora de abordó. Esta computadora se encargará de manejar y procesar toda la información enviada y recibida por el subsistema de comunicación basado en IoT.

Por ende, se requiere de un módulo de comunicación por medio de redes celulares con latencia menor a 1[s], donde en este caso puede entrar el estándar GSM con 0.5-1[s] de latencia para un número reducido de datos, u optar por el estándar 4G con una menor latencia de hasta 26[ms] con un ancho de banda superior al GSM.

Finalmente, se presenta en la Tabla 13 el contenido para enviar mensajes entre el sistema y la estación de control.

Tabla 13. Estructura de Mensaje

Número de bytes	Nombre y/o función
1	Cabecera
1	Tipo de comando
3	longitud
1-93,750	Carga útil ¹⁰
4	ACK

4.2.4 Subsistema de detección de obstáculos

El subsistema de detección de obstáculos basado en LIDAR se encarga de proporcionar una serie de paquete de datos, correspondientes a puntos obtenidos respecto al marco de referencia (dron). Estos puntos en conjunto permiten saber si el sistema se encuentra en colisión con algún posible objeto, lo cual, permite a la computadora de abordó elegir un camino libre para seguir un camino sin colisiones. De la misma forma, este tipo de tecnología se puede utilizar para buscar un área libre en donde se pueda liberar la mercancía llevada por el dron.

En este caso, el sistema de detección se encuentra escaneando constantemente la zona de interés de navegación y, toda la información se envía de manera serial (UART o USB) a la computadora de abordó, encargada de procesar y analizar los datos.

De manera general, el sistema detección de obstáculos basado en LIDAR permiten un rango de vista de 30° de forma vertical y, con una visión de 360° de forma horizontal, con una gran cantidad de muestras tomadas del ambiente. Un punto clave en el sistema es la cantidad de muestras capaz de tomar del ambiente y, la velocidad de navegación del dron. Lo anterior está ligado en cuestión de velocidad de procesamiento, debido a que, si el sistema no es capaz de detectar objetos y tomar una decisión a la velocidad de deseada, el sistema simplemente fallará.

El subsistema de detección por consiguiente debe contar con un alcance mínimo de 8[m], y proveer más o al menos 1000[muestras/s], entre mayor el número de muestras, mejor será la resolución del ambiente. De la misma manera, se puede considerar un giro de 360°, capaz de ver por detrás del dron. Los datos enviados por este subsistema deben

¹⁰ La estimación de los datos útiles se hizo calculando el máximo volumen de bytes enviados en una revolución por diferentes LIDAR comerciales con mayor capacidad al diseñado en cuestión.

contener indispensablemente la distancia, la posición y el instante de tiempo desde la cual se tomó la medición.

4.2.5 Subsistema de posicionamiento

Uno de los puntos indispensables del proyecto es conocer la ubicación del dron en todo momento. Estos datos son indispensables, principalmente porque ayudan en la generación de la guía del UAV y, porque también indica las coordenadas de donde se deberá entregar el paquete.

Con la finalidad de tener conocimiento sobre los 3 ejes coordenados de nuestro dron, se debe incluir un GNSS y un altímetro que provean suficiente precisión en la posición del sistema. En cuanto mejor sea la precisión de los dispositivos, se tendrán excelentes resultados en la entrega final.

Al considerar estos elementos en este subsistema, de la misma manera se debe considerar con una conexión serial con la computadora de abordo para usar directamente estos datos. Estos dispositivos se encargan de cumplir con un funcionamiento específico, así, el altímetro se encarga de mantener una altura fija y que no se debe superar de acuerdo con la NOM, mientras que el GNSS, cubre el plano de movimiento horizontal, permitiendo desplazarse hacia la dirección destino.

En el caso del GNSS, al utilizarlo como guía, se requiere además de la precisión, una velocidad de actualización rápida de los datos, con la finalidad de asegurar que el dron está siguiendo el curso adecuado y no se saldrá de éste, además esto ayuda a mantener al dron en una posición de vuelo fijo sin el problema de chocar con objetos cercanos.

En este subsistema el principal requerimiento referente al altímetro es que debe ser capaz de realizar mediciones a partir de la altitud en el Estado de México (2683 m.s.n.m), permitiendo mediciones de hasta 150[m] de altura y con un error de precisión de hasta 10[m], en el caso de ser sensores de presión atmosférica. Para el GNSS, la precisión de la ubicación debe permitir un rango de hasta 2.5[m] y, al ser parte de la orientación del curso, la frecuencia de actualización de los datos debe ser por lo menos mayor a 10Hz, para evitar que el dron se desvíe mucho de su rumbo.

4.2.6 Subsistema de control de vuelo

El control de vuelo específicamente referido al FC, los ESC y los motores brushless, se encargan de generar el control sobre la dinámica del dron. De manera más puntual, el FC recibe comandos de la computadora de abordo, los cuales, indican hacia dónde debe moverse el dron y, esta dirección se procesa por el FC para que, de acuerdo con sus sensores, se den las señales de control necesarias a los ESC, indicando a cada motor el par necesario para trasladar al dron de un punto a otro.

En este subsistema se considera como elemento principal al FC, debido a su comunicación directa con la computadora de abordo. Esta comunicación se puede realizar por medio de UART, o a través de USB, donde los mensajes compartidos entre dispositivos se realizan siguiendo un protocolo conocido como MAVLink.

Es por medio de la comunicación anterior que la computadora tiene un total control sobre el FC y, este último, se encarga de efectuar las instrucciones y controlar a los demás dispositivos necesarios en el vuelo.

Uno de los requerimientos esenciales del controlador de vuelo es posibilitar el control a través de una computadora de abordo, en estos requerimientos se recomienda usar un *baudrate* de 921600 o más para enviar los mensajes con longitud de 11-279 bytes. Después, tiene que contar mínimo con 4 salidas para los ESC y, soportar plataformas de código abierto, con la finalidad de sacar ventaja de acciones autónomas con el dron.

Los ESC por su parte, deben soportar la corriente de operación requerida por los motores brushless y, otro aspecto a considerar importante es la frecuencia de operación indicada por el protocolo soportado por estos. Una forma sencilla de visualizar cómo afecta la frecuencia de operación en la maniobrabilidad del dron es, por ejemplo, a través de calcular la distancia recorrida por el dron antes de recibir un nuevo valor de torque en cada motor, o dicho de otra manera, antes de cambiar su dirección de movimiento. Para ello se consideran algunas velocidades de operación de los drones repartidores que abarcan desde los 70km/h a 160km/h aproximadamente. Asimismo, se considera el análisis para el protocolo Dshot1200 que permite el envío de 1200kbits/s y, se compara con otros protocolos con una menor velocidad como Oneshot42, con una frecuencia de actualización máxima de 12kHz.

Dicho lo anterior, analizamos el primer caso con Dshot1200, que requiere de una trama de 16 bits para cambiar el torque del motor, dando como resultado una frecuencia de actualización de 75kHz. Con esto, si sabemos que nuestro dron viaja a una velocidad de 160km/h, o bien, 44.5m/s, podemos hacer un aproximado de cuánta distancia recorrerá el dron antes de que se tenga un nuevo valor de torque y, por ende, el dron comience a modificar su curso. Estos valores de velocidad y frecuencia dan como resultado una distancia de 0.59mm, mientras que en el caso del protocolo con actualización de 12kHz da un valor de 3.7mm, lo cual resulta en un valor de distancia chico para la velocidad de operación del dron.

Es cierto que la distancia avanzada es bastante pequeña, sin embargo, a eso se deben agregar otros factores como los retardos en las señales, la latencia en la comunicación, el procesamiento de las señales del control (por lo general es rápida comparada a los protocolos ESC) y, además que el cambio del movimiento del dron no se da de forma

instantánea. En ese sentido, se hablaría de unos centímetros recorridos, donde convendría bastante usar un protocolo más rápido como Dshot1200, aunque, por un lado, se consideró una de las velocidades máximas encontradas, donde estas son operadas de manera general en zonas con áreas mayormente despejadas. En otros casos, como en el del dron EmQopter, cuya velocidad es de 70 km/h, se reduce considerablemente el requerimiento de velocidad en los ESC, con lo cual, para propósitos de esta tesis se puede seleccionar una frecuencia de operación mayor o igual a 8kHz.

Finalmente, los motores siendo uno de los componentes con un alto consumo de energía, deberán proveer de una eficiencia cercana al valor 7 [g/W], además de proveer un empuje total de por lo menos el doble del peso aproximado del dron.

4.2.7 Computadora de abordo

La computadora de abordo cumple con la mayoría de las tareas requeridas por el sistema para la autonomía del dron. Es debido a la gran cantidad de tareas por la cual se explica puntualmente la manera en que operará este subsistema, atendiendo a cada uno de los subsistemas comunicados a éste.

Para empezar con el subsistema es necesario considerar desde la planificación y preparación de la entrega de un paquete, hasta el último momento en donde se libera. En este caso lo primero es la manera en que se programa al dron para salir, es decir, al servidor llegan los datos del usuario y, estos datos que contienen la ubicación del destino son cargados por medio del subsistema de comunicación a la computadora de abordo, cuyo objetivo es trazar una ruta óptima y, una vez generada esta, el dron repartidor debe ser llevado a un espacio de despegue donde, por medio de la comunicación se le dará órdenes para iniciar la navegación.

Una vez iniciada la navegación, la computadora de abordo indicará al FC, que se mueva de manera vertical hasta alcanzar la altura adecuada, siendo esta información obtenida del subsistema de posicionamiento. Después de esta acción, la computadora activará el subsistema de detección de objetos para conocer su entorno y poder evitar colisiones, para ello, se requiere de algoritmos que interpreten la información del ambiente y, permitan escoger una desviación en la ruta, de acuerdo con los datos recolectados.

Inmediatamente de alcanzar la altura y activar el sistema de detección de objetos, la computadora indica al FC que puede comenzar a desplazarse de manera vertical con ayuda del subsistema de posicionamiento. En la navegación, la información tanto del subsistema de detección de objetos como del de posicionamiento en conjunto son esenciales en la computadora de abordo para que el dron no se salga del curso de navegación y, para generar rutas alternas hasta llegar al lugar de destino.

Al llegar al destino, son varias acciones que se deben desempeñar. Estas son indicadas por la computadora de abordo; primero se inicia un vuelo estacionario; luego se debe verificar la zona donde se despachará la mercancía hasta encontrar una adecuada; posteriormente, se verifica que el usuario desea obtener su mercancía por medio de códigos OTP¹¹, esto, para evitar cualquier tipo de problemas por pérdida o robo; y, por último, la computadora de abordo indica al dron que debe soltar el paquete. Con la entrega del paquete listo, únicamente se regresa al dron a la estación, de acuerdo con lo descrito en los párrafos anteriores.

Además de lo anterior, faltan formalizar otras tareas, como la telemetría enviada, los comandos recibidos y el modo de operación manual de nuestro dron. En este caso, algunos aspectos relevantes de la telemetría son el estado de la batería, de conexión y de la posición el sistema, estas son las que el dron necesita enviar de manera constante cada cierto tiempo. Asimismo, las alertas únicamente son necesarias justamente cuando ocurra alguna falla de uno de los subsistemas u otras razones ajenas al sistema.

En el caso de los comandos que ayudan a realizar ciertas acciones con el dron de manera remota, estos deberán ser monitoreados y, a su vez, podrán ocurrir en cualquier momento de la misión. La computadora encargada de monitorear el subsistema de comunicación revisa la cola en busca de comandos y, decodifica la solicitud para realizar una acción, además, el subsistema deberá verificar este tipo de comandos e indicar al operador la correcta ejecución del comando. Por ejemplo, indicar cuando el dron está listo para pasar de modo manual a automático o viceversa, o simplemente con la recepción de la telemetría solicitada.

Al considerar un modo manual, la computadora de abordo debe ser capaz de administrar y dejar de ejecutar algunas de las tareas que realiza de modo automático, como, por ejemplo, seguir generando rutas, disminuir el procesamiento en la computadora de abordo, entre otros. En lugar de esto, pasa a codificar todos los comandos el subsistema de comunicación que servirán en el manejo del dron y, le enviará estas instrucciones de nuevo al FC. Además, se dispone de activar la transmisión de los datos proporcionados por LIDAR para uso del operador del dron.

Como se logra analizar, la computadora de abordo realiza una gran cantidad de tareas, por lo cual, se necesita de un sistema operativo en tiempo real capaz de manejar a todas ellas. Por otra parte, hablando del hardware, al encontrarse conectado con diferentes sistemas y realizar el procesamiento de estos, se dispone a diseñar una placa que cuente con las interfaces y memorias suficientes de acuerdo con los requerimientos descritos.

¹¹ OTP por sus siglas en inglés, significa contraseña de un solo uso y es un código que permite asignar una contraseña a un usuario, para ser usada una única vez en su autenticación de identidad.

Una manera para facilitar los requerimientos de la computadora de abordó es asignando un porcentaje de los recursos de la computadora a diferentes tareas, las cuales se muestran en la Tabla 14.

Tabla 14. Asignación de prioridad de cada tarea.

Prioridad	Nombre de la tarea
1	Captura de LIDAR
2	Planeación, guía y control del UAV
3	Procesamiento de LIDAR (reconstrucción y formación de rutas libres)
4	Gestión de medición e integración de información de subsistemas
5	Estado del sistema y alertas
	Reserva

En la Tabla 14, se debe mencionar que la información obtenida por LIDAR resulta vital en el funcionamiento del sistema, sin embargo, la tarea del procesamiento de los datos de LIDAR consumirán un valor superior de recursos, conformado por algoritmos de reconstrucción y decisión, los cuales, requieren mínimamente las siguientes especificaciones¹²:

- Reconstrucción con nubes de puntos de LIDAR (Representación poligonal): En este se recrea el espacio a través de polígonos y se reconocen por sus vértices. De manera general, este tipo de algoritmos tienen un tiempo de ejecución de $O(n \log n)$, donde n es el número de puntos.
- Algoritmo de Voronoi: El tiempo de ejecución crece de la forma $O(n^2)$ en el peor de los casos, donde n es el número de puntos ocupados. De manera general tiene una complejidad $O(n)$.

Por su parte, los algoritmos de control y planeación para la guía del dron en modo manual tienen las siguientes especificaciones:

¹² $O(n)$ hace referencia a la cota superior de una función e indica que el algoritmo no rebasará este valor.

- Algoritmo de planeación (Algoritmo de Dijkstra): Este algoritmo de manera general el tiempo de ejecución crece de la forma $O(n^2)$, siendo n el número de vértices.
- Algoritmo de control (Navegación basada en GNSS): Por medio de la comunicación de la computadora de abordo con el FC, se pueden enviar hasta 412 mensajes/s para controlar al UAV, asimismo, como la navegación se basa en GNSS, y de esta se definió un mínimo de actualización de 10Hz, nos deja un algoritmo con tiempo máximo de ejecución de 100[ms] y un mínimo de 2.4[ms] debido a la cantidad de mensajes enviados de la computadora de abordo al FC.

De manera general también se investigó otros algoritmos basados en LIDAR con mayor capacidad de puntos para detección de obstáculos, donde principalmente en [36] el algoritmo presenta una complejidad de $O(n)$, este algoritmo puede funcionar como referencia para escalar la capacidad de la computadora de abordo.

Con estos requerimientos de algoritmos y la cantidad de información se puede hacer una estimación de los requerimientos mínimos para la computadora de abordo, principalmente hablando de velocidad de procesamiento, memoria RAM e interfaces seriales.

En el caso de la velocidad de procesamiento, esta se puede determinar de acuerdo con el número de operaciones requeridas por los algoritmos. Por ejemplo, para el algoritmo de detección de objetos desarrollado en [36], este tiene una tendencia a realizar 164 instrucciones¹³ por cada punto, de acuerdo con el procesador de 4 núcleos desempeñado por el autor en cuestión. Ahora bien, si se define el procesamiento para 30,000 puntos muestras de una revolución de 10Hz y considerando lo anterior, se puede hacer un estimado de la velocidad del procesador requerido, donde, si se conoce el rango máximo del sensor y se deja un margen mínimo para que avance el dron (asegurando que no habrá colisión) se puede calcular un tiempo de procesamiento adecuado para cada nube de puntos nuevos, siendo la cantidad de instrucciones y el tiempo de procesamiento los que definirán la frecuencia del reloj. En este caso, se considera un margen de error de 1[m], el alcance de LIDAR como 8[m] y una velocidad del dron de 20[m/s], se puede adecuar la velocidad de procesamiento para un procesador de 1 núcleo, justo como sigue¹⁴:

¹³ La estimación de las instrucciones necesarias por cada punto de LIDAR, se obtuvo del algoritmo desarrollado en [36], donde el tiempo de ejecución es de .156s para 2.384 millones de puntos con un procesador de 2.53GHz y cuatro núcleos.

¹⁴ Para este cálculo se consideró que cada instrucción del procesador se ejecuta en un ciclo de reloj y, al obtener el tiempo de un procesador de 4 núcleos, el número de instrucciones puede considerarse un múltiplo para un solo procesador. De la misma manera, la cantidad de puntos se considera adecuada de acuerdo con la frecuencia de muestreo típica de algunos LIDAR.

$$f_{reloj} = \frac{\#Instrucciones}{t_{Ejecución}} = \frac{\#Instrucciones}{\frac{d_{Margen}}{v_{Dron}}} \quad ec. (51)$$

$$f_{reloj} = \frac{\#Instrucciones(v_{Dron})}{d_{Margen}} = \frac{(164 * 4)(30,000)(20m/s)}{(1m)} = 393.6[MHz]$$

Si bien, la estimación del reloj se realizó para un procesador de 1 núcleo, sin embargo, esto es sólo con el propósito de facilitar la comparación de los procesadores con diferentes cantidades de núcleos capaces de proveer esta velocidad de procesamiento.

En este caso, se debe destacar que el reloj requerido para el procesador de 1 núcleo es de aproximadamente 400MHz, considerando que el algoritmo de detección de objetos es superior en cuanto a la cantidad de instrucciones requeridas, a diferencia de los demás algoritmos. Sin embargo, si, por ejemplo, se desease realizar todas las tareas en 0.1[s] y en total el tiempo de ejecución de todos los algoritmos exceden este valor, entonces, se podría considerar aumentar la frecuencia del reloj, para tener un tiempo de respuesta adecuado. En cuanto mayor sea este, con mayor rapidez se ejecutarán los algoritmos requeridos por la computadora de abordo.

Con el enunciado previo, el tiempo de ejecución total de la computadora se considera de aproximadamente el doble del tiempo de ejecución del algoritmo de detección de objetos para LIDAR 3D, es decir, la computadora de a bordo debe ejecutar todas sus tareas en un tiempo de 100[ms], con lo cual, se puede hacer la asignación del uso de CPU a este tiempo límite.

Después un requerimiento mínimo para la memoria RAM se puede obtener por medio de estimar los datos máximos recopilados por la computadora de abordo para cada subsistema, descritos en la Tabla 15.

Tabla 15. Estimación máxima de datos a procesar.

Subsistema	Datos máximos
Comunicación¹⁵	256 bytes
Detección de objetos	97,500 bytes
Posicionamiento¹⁵	270 bytes

¹⁵ Se define un mínimo de memoria para almacenar información, debido a que el protocolo no establece un límite, o para simplificar la comunicación.

Control de vuelo	280 bytes
Potencia¹⁵	256 bytes
Datos extra	5000 bytes (+10%)

Estos datos se consideraron atendiendo que la información requerida para operar por el subsistema se utiliza una sola vez y, después de ello se sobrescriben los datos en la memoria, debido a que no se necesita almacenar información, para ello se podría optar por incluir una uSD a la computadora de abordo. En conjunto los datos dan un mínimo de 110kB en memoria RAM requerido para el sistema, en caso de aumentar la cantidad de datos por LIDAR, los requerimientos aumentarán considerablemente.

Finalmente, nuestra computadora de abordo debe contar de manera mínima con 6 interfaces seriales, en las que destacan 1 SPI, 1 I2C, 2 UART, 1 CAN, 1 USB, donde la alimentación de la computadora se puede desempeñar por medio de CAN o USB.

4.2.8 Subsistema de potencia

El subsistema de potencia encargado de proveer de la energía requerida a cada uno de los componentes electrónicos dentro del sistema se compone básicamente de una unidad de manejo de potencia y de los dispositivos de almacenamiento de energía.

La unidad de manejo de potencia se encarga justo como su nombre lo dice, obtener la potencia de la batería y de las celdas solares para distribuirla a los diferentes componentes, de acuerdo con las necesidades de operación de voltaje de cada uno. Además de distribuirla, se encarga de proporcionar información sobre el estado de la potencia restante y, también, permite conocer en qué momento es conveniente alimentar ciertos dispositivos con las celdas solares, en lugar de la batería principal.

De la misma manera, estos datos del sistema de potencia son enviados a la computadora de abordo a través del bus CAN, el cual es utilizado para mejorar la precisión debido a la interferencia que se pueda presentar. Estos datos, se usan para telemetría o en la ayuda de aterrizajes forzados para el sistema. Si bien, la capacidad del subsistema de potencia se puede calcular como se ha mencionado de acuerdo con la distancia máxima de entrega esperada o, por medio de los recursos con los que ya se dispongan como la batería.

Como alimentación principal en el subsistema se utilizará una batería de LiPo cuya capacidad se determinará con respecto al tiempo de vuelo, después de manera secundaria se analizará la implementación de paneles solares con eficiencia mayor al 25%. El voltaje obtenido de ambos dispositivos deberá ser convertido (DC-DC, normalmente con convertidor Buck) a valores de 3.3[V], 5[V] y 12[V], con un error del 1%. Asimismo, este

subsistema contará con una unidad microcontroladora (MCU, por sus siglas en inglés) que le permitirá efectuar el control sobre el manejo de potencia y envío de información.

El MCU debe contar con mínimo 4 PWM, 5 ADC, 1 CAN y varios GPIO para señales de control. A diferencia de la computadora de abordo, la capacidad de procesamiento se reduce considerablemente, debido a que las tareas principales de este subsistema será muestrear voltaje de sensores, generar señales PWM para el control de los convertidores y, enviar información sobre el estado y consumo de la batería. En general, se puede optar por un microcontrolador de bajo consumo y precio para disminuir costos.

4.3 Diseño a detalle

En subcapítulos previos se propuso el concepto del dron repartidor y, eso conllevó a su desarrollo al nivel sistema, propiciando así, un mejor entendimiento sobre los requerimientos y funciones desempeñadas por cada uno de los subsistemas. Dicho esto, el siguiente paso por abordar y llevar a cabo es el diseño a detalle.

En el diseño de detalle se establecen las características definitivas de cada uno de los subsistemas contemplados en el proyecto. En otras palabras, este apartado trata sobre la selección y diseño de los dispositivos electrónicos capaces de cumplir con las limitaciones expuestas para el correcto funcionamiento del sistema en general.

Anticipadamente en el Estado del Arte además de exponer las características de diferentes UAV y, en general, del dron repartidor, se presentaron y compararon diferentes dispositivos electrónicos con el objetivo de facilitar la selección de cada uno de estos. Esto no sólo facilita el escoger los elementos de acuerdo con su funcionalidad, sino que, también posibilita la elección con base al precio del producto.

4.3.1 Estructura

La estructura es uno de los elementos básicos de un dron, en donde se colocan todos los dispositivos necesarios en la misión. Una de las principales características de la estructura es que provee de una buena rigidez, a la vez que es ligera, es por esa razón que uno de los materiales más utilizados, dentro de aplicaciones con UAV, es la fibra de carbono.

Por otra parte, un segundo punto a tratar en la estructura es la configuración, o dicho de otra manera, del número de motores a incluir en el dron. En este caso, hablando de un prototipo y, debido a que la capacidad del paquete de entrega de los drones repartidores es menor a 3kg, resulta factible seleccionar como estructura la configuración del cuadricóptero, siendo éste el más barato de desarrollar, en caso de querer algo más seguro y redundante, se puede cambiar la estructura por un hexacóptero, cuando todos los elementos del sistema funcionen en conjunto de manera correcta.

4.3.1.1 Clase y dimensiones

La clase hace referencia a la distancia máxima existente entre rotores, cuya trayectoria pasa a través de la diagonal principal del dron. Habitualmente, este parámetro es el primero que proporcionan la mayoría de los proveedores de estructuras de drones y, es un parámetro importante al momento de elegir el tamaño de las hélices, evitando que estos choquen entre sí.

Bajo esa perspectiva, la base principal del dron debe contar con el suficiente espacio para montar todos los componentes descritos en el sistema. Una opción viable y con el propósito de disminuir el tiempo de desarrollo del proyecto en cuestión, es a través de adquirir una de las estructuras disponibles en el mercado con las características anteriormente discutidas y, hacer modificaciones pertinentes, si es que así se necesitase.

Dadas las directrices anteriores, la estructura escogida fue de un cuadricóptero de fibra de carbono, denominado “ZD550”, cuyas características se muestran a continuación:

- La distancia entre ejes diagonal de la base del motor es de aproximadamente 550mm y, la distancia de separación entre ejes es de aproximadamente 395mm.
- El espacio del tablero central es de 150x150mm, con una altura media del tablero central de aproximadamente 36mm.
- Peso aproximado de 630g.

4.3.1.2 Mecanismo de entrega

El mecanismo para implementar en la liberación del paquete estará basado en los drones de entrega de Google, los cuales, emplean una especie de grúa mecanizada [37] y, cuando se debe entregar la mercancía, este simplemente se desenrolla y desengancha del mecanismo, para finalmente volver a su posición inicial.

Por esa razón el mecanismo se realizará usando el diseño de un malacate, combinado con un motor brushless con un torque mínimo de 2kg y, además necesita un interruptor o un sensor para indicar cuando el paquete o el gacho esté completamente pegado al dron.

Una manera sencilla de realizar esto es por medio de un simple motor con un par mínimo de 2kgcm, junto con una cuerda y un gancho metálico como se muestra en la Figura ¿?



Figura 38. Figura representativa del mecanismo

De igual forma el mecanismo necesita conocer cuando el paquete está pegado correctamente al dron, para ello se puede utilizar interruptor de carrera que se presione al subir completamente la plataforma.

4.3.2 Controlador de vuelo

Como se ha visto, el controlador de vuelo cumple con la función de proporcionar las señales de control al ESC para la estabilización y movimiento del dron. La principal característica del controlador de vuelo a seleccionar es la de permitir la comunicación con una computadora de abordo, para que en conjunto generen la guía y autonomía del dron repartidor, propiciando de esta manera la disminución del tiempo de desarrollo e implementación del control de vuelo. Esto a su vez, ayuda a distribuir la capacidad de procesamiento, liberando a la computadora de abordo del procesamiento requerido por sensores y algoritmos empleados en el movimiento del dron, por lo que únicamente se centrará en aspectos más relevantes como proporcionar la autonomía necesaria en la misión del dron repartidor.

Una opción, debido a su compatibilidad con computadoras de abordo, su documentación y su implementación en este tipo de aplicaciones, es la familia de FC de Pixhawk, siendo estos conocidos como sistemas de piloto automático de código abierto orientado a aviones autónomos. Dicho esto, el FC seleccionado para fungir el papel del control del dron es el modelo Pixhawk 2.4.8, mostrado en la Figura 39.

Las características de este dispositivo se enlistan a continuación:

- Microprocesador STM32F427 de 32 bits Cortex M4 con unidad de punto flotante (FPU, por sus siglas en inglés). Este cuenta con un oscilador de 168MHz, 246kB de memoria RAM y, 2MB de memoria Flash.
- Los sensores implementados por el FC: giroscopio digital de 16 bits de 3 ejes L3GD20, acelerómetro/magnetómetro de 14 bits de 3 ejes LSM303D y, giroscopio/acelerómetro de 3 ejes MPU6000.
- 14 salidas PWM/servo, permitiendo la operación de un octacóptero.
- Abundantes opciones de conectividad para periféricos adicionales: 5 UART, 2 CAN, 1 I2C, 1 SPI y 1 puerto externo micro USB.
- Soporta plataformas de firmware como ARDUPILOT y PX4, estas permiten configurar en modo manual y automático, facilitando la implementación de acciones como ascender, descender y volar fijamente.
- Tiene compatibilidad con memoria microSD para guardar datos.



Figura 39. Controlador de vuelo Pixhawk 2.4.8

Dadas las características de este dispositivo, se eligió controlador de vuelo del dron, donde la guía e indicaciones se llevarán a cabo por medio de la computadora de abordo diseñada, disminuyendo de esta manera el tiempo de desarrollo y enfocándolo en la computadora de abordo.

4.3.2.1 Plataforma PX4

4.3.2.2 MAVLink

Como se ha visto, MAVLink es un protocolo para enviar mensajes ya sea a través desde la cabina de control o bien, por medio de una computadora de abordo. La manera en que se estructuran o se deben llenar los campos del mensaje a enviar se encuentra contenidos en [38], siendo estos enviados a través de UART por medio de la computadora de abordo hacia el controlador de vuelo. Esta computadora de abordo en general podrá cambiar en cualquier momento el curso del dron y, además, podrá acceder a cierta información del controlador de vuelo, como la de sus sensores.

4.3.3 Motores Brushless, controladores electrónicos de velocidad (ESC) y hélices

Los principales actuadores implementados en drones son los motores brushless cuya característica principal es la de disminuir el desgaste mecánico y pérdidas por fricción, aumentando de esta manera la eficiencia de los motores.

Uno de los aspectos relevantes a estimar en la selección de los motores es que estos ofrezcan el suficiente torque para elevar el peso del dron junto con el peso del paquete estimado.

Para este caso, el peso del dron considerando el paquete se estima en aproximadamente 5kg, aunque esto no indica que el empuje máximo requerido por los 4 motores sea éste. En este caso, es recomendable seleccionar los motores en configuración de cuadricóptero, tal que, generen como mínimo un empuje del doble del peso del dron al 100% de aceleración, es decir, los motores deberán proporcionar el torque necesario para un empuje máximo de 10kg al 100% de aceleración, o, dicho de otra manera, cada motor debe proporcionar un empuje mínimo de 2.5kg. Además, si se desea aumentar la velocidad, los motores deberán propiciar un empuje mayor al descrito.

Por otra parte, la eficiencia en los motores es otro aspecto importante en su elección, siendo éste un factor relevante en el consumo energético del sistema. En el caso de estos, la eficiencia se mide por la cantidad de gramos capaz de levantar por cada watt consumido [g/W], en otras palabras, en cuanto mayor sea la eficiencia, aumentará la capacidad del sistema para levantar peso con un menor gasto energético. Por lo general, se recomienda elegir motores con una eficiencia cercana a 7 o mayor a este valor.

Existen dos formas de obtener estos parámetros: de manera práctica o teórica. La primera suele ser proporcionada por el fabricante, mientras que, en el otro caso, se debe hacer el cálculo de acuerdo con los parámetros de cada componente.

En este caso, una aproximación de los motores que se pueden implementar en el sistema y que cumplen con las características comentadas se llaman “SunnySky X4112S” (Figura 40), cuyos aspectos relevantes son:

- Máximo empuje 4100g.
- Peso de 197g y diámetro 47.5mm.
- Peso de despegue recomendado en configuración cuadricóptero 4.0-6.8kg.
- Buena ventilación para disipación de calor.
- Embobinado con baja resistencia interna, con una alta eficiencia.



Figura 40. Motores SunnySky X4112S

Si bien, se debe constar que los motores seleccionados pueden ofrecer un empuje mucho mayor al necesario, no obstante, en la práctica tener un mayor empuje permite de cierta manera al dron aumentar su velocidad, tener mayor maniobrabilidad o llevar paquetes un poco más pesados. Incluso el mismo proveedor recomienda el peso de despegue que pueden soportar los motores para diferentes configuraciones del dron, especificando cómo en configuración de cuadricóptero el peso del dron debe estar en el rango de 4-6.8kg, el cual es el caso de este prototipo.

Una vez definidos los motores, el siguiente paso resulta más sencillo, en la selección de ESC y hélices. En este punto, el proveedor de estos motores recomienda la capacidad del ESC y el tamaño de las hélices que se pueden implementar en ellos.

En el caso de los ESC, una de sus principales cualidades a atender es la corriente nominal de operación, donde se sugiere usar ESC con un valor de 40-60[A]. La selección del valor nominal utilizado se realiza de acuerdo con las condiciones de operación de los motores, siendo los de 40[A] una opción viable y barata. Sin embargo, el escoger ESC con mayor cantidad de corriente nominal tiene ventajas como evitar el sobrecalentamiento y una mayor eficiencia, donde sus desventajas son el peso y precio de estos. Asimismo, es relevante mencionar que la eficiencia de los ESC es un factor clave en su selección, puesto que, estos son los que consumen la gran mayoría de energía de potencia, por ende, en cuanto más eficientes, menos energía desperdiciarán, aunque este no es un factor que proporcionen los fabricantes.

Otro aspecto relevante de los ESC son los protocolos utilizados para comunicarse, de los cuales la principal diferencia es la velocidad en el envío de información y si es analógico o digital. A pesar de la diferencia de ejecución entre cada protocolo, en algunos casos resultan en la redundancia de la información enviada y, por tanto, el desempeño para el control de los motores no se ve mejorado por la velocidad de transferencia.

Además, otra ventaja a destacar de los ESC actuales es la implementación de 4 ESC en una sola placa. Estas permiten realizar las mismas funciones que 4 ESC por separados, pero con la diferencia de tener un menor tamaño y peso. El único problema con este tipo de ESC es que, si uno de estos se descompone, se recomienda reemplazar la placa completa por otra.

En ese contexto, los ESC seleccionados se denominan “Emax BLHeli series ESC”, cuya descripción se muestra debidamente a continuación:

- Basado en el firmware BLHeli, optimizado aún más para el rendimiento perfecto de la unidad.
- Protección de bajo voltaje, protección contra sobrecalentamiento y protección contra la pérdida de señal del acelerador.
- Soporta una corriente continua de 60[A], con un pico de corriente de hasta 80[A].
- Permite la programación del dispositivo, por lo que sus parámetros se pueden configurar, permitiendo trabajar con frecuencias de 8khz y 22khz.



Figura 41. Emax BLHeli series ESC

Algunos de los motivos para escoger estos, además de los mencionados con anterioridad, fue a su gran popularidad y al descuento en cuanto a su precio, con lo cual, se puede utilizar para aplicaciones con mayor exigencia en los ESC. Por otra parte, a diferencia de los ESC de 40[A], este ofrece un mayor soporte de corriente y la diferencia en tamaño varía únicamente en el peso por unos 23[g].

Con estos dos elementos definidos, sólo resta seleccionar las hélices que proporcionarán la aerodinámica necesaria para el empuje y movimiento del dron. Sobre esto, es indispensable mencionar cómo el uso de diferentes configuraciones y tipos de hélices contribuyen significativamente en la eficiencia y, por consiguiente, en el radio de entrega del sistema.

En la elección de las hélices se optó por las hélices recomendadas de “15x5.5”, hechos con material de fibra de carbono. Lo anterior, debido al desempeño observado en los datos de prueba, logrando el empuje requerido por el sistema sin la generación de una temperatura excesiva en los motores.

Dos de los parámetros importantes en la selección de las hélices son proporcionadas por todos los fabricantes, siendo estos el diámetro de la hélice y el ángulo de ataque (pitch). En la hélice a usar, el 15 en [in] representa el diámetro o tamaño total, mientras que los 5.5[in] indican el pitch, cuya importancia es el desplazamiento producido por una hélice en una revolución completa.



Figura 42. Hélices de fibra de carbono 15x5.5

Finalmente, otra consideración es el tipo de hélice implementadas en la configuración del dron. En general, se clasifican como hélices de giro horario y antihorario y, de la misma manera, suelen ser vendidas por pares. La única cuestión acerca de esto es colocar en la posición adecuada cada una de ellas, como en el caso del cuadricóptero que usa dos de tipo horario y dos de tipo antihorario como se ha explicado en el capítulo 3.1.

4.3.4 Subsistema de Comunicación

Para facilitar el tiempo desarrollo se utilizará las placas de desarrollo del módulo de comunicación basado en redes celulares. En general, se decidió utilizar los módulos de la empresa SIMCOM porque se dedican a elaborar específicamente este tipo de tecnología para tener conexión a la red por medio de redes celulares, específicamente en la red 4G, otra opción sería utilizar la red 5G que presenta mayores ventajas frente a la 4G, no obstante, en México aún existen bastantes zonas donde no se cuenta con la disponibilidad de este servicio.

Otro punto importante es que debe contar con la menor cantidades de características para disminuir el costo de este dispositivo y desempeñar únicamente la tarea de comunicar al dron con la estación base. Con lo anterior en mente, el módulo escogido es A7670SA (Figura 43) y se muestra a continuación con su descripción correspondiente.

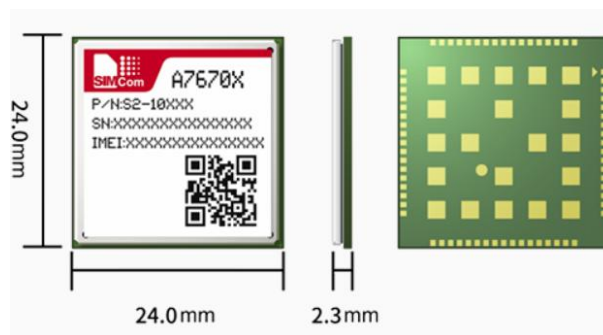


Figura 43. Módulo 4G A7670SA

1. Soporta una gran variedad de conectividad: LTE-TDD/LTE-FDD/GSM/GPRS/EDGE
2. Bandas de frecuencias soportadas:
 - LTE-FDD¹⁶: B1/B2/B3/B4/B5/B7/B8/B28/B66
 - GSM: 850/900/1800/1900MHz
 - Transferencia de datos¹⁷:
 - LTE (Mbps): 10 (DL)/5 (UL)
 - GPRS/EDGE (Kbps): 236.8 (UL/DL)

¹⁶ LTE FDD es ...

¹⁷ UL y DL, por sus siglas en inglés, significan subida y descarga, respectivamente, haciendo referencia a la velocidad de transferencia de la conductividad utilizada.

3. Soporta una gran variedad de software enlistados en [39].
4. Interfaces: SIM card, USB, UART, I2C, PCM, ADC, GPIO.
5. Voltaje de alimentación de 5V (placa de desarrollo).
6. Control por medio de comandos AT.

Una de las principales razones de la selección de este módulo es justamente porque usa 4G para la transferencia de datos, independientemente de que incorpora GSM; la segunda razón es las bandas de frecuencia de operación correspondientes a las empleadas en México; asimismo, la misma empresa da recomendación sobre la región en donde deben operar sus dispositivos, por ejemplo, el que se seleccionó en cuestión se encuentra de manera general como A7670XX, donde estas XX, se cambian por notaciones como SA, NA, E, JC, especificados en [40], siendo SA una referencia a Sudamérica y otras regiones, donde específicamente México entraría en la notación NA (Norte América), sin embargo, las bandas del dispositivo seleccionado coinciden más con las utilizadas en nuestro país.

Finalmente, la empresa misma ofrece una gran variedad de documentación en la implementación de sus dispositivos, facilitando la búsqueda y el tiempo de desarrollo de su sistema.

4.3.4.1 Comandos AT

Los comandos AT son instrucciones codificadas que conforman un lenguaje de comunicación entre el hombre y una terminal modem, se pueden hacer cosas como:

- Lectura, escritura y eliminación de mensajes SMS
- Envío de mensajes SMS (Short Message Service).
- Seguimiento de la intensidad de la señal.
- Seguimiento del estado de carga y nivel de carga de la batería.
- Lectura, escritura y búsqueda de datos del directorio de contactos.

La empresa SIMCOM ofrece documentación sobre los comandos AT soportados por sus diversos módulos para el desarrollo del sistema. Toda la información relevante sobre su funcionamiento se encuentra documentado en [39].

4.3.5 Subsistema de Posicionamiento

En el estado del arte se halló que los GNSS más utilizados en aplicaciones con drones son los de la empresa conocida como u-blox, en cuya página se presentan diferentes soluciones de acuerdo con nuestras necesidades como, por ejemplo, comunicación de largo y corto alcance, posicionamiento y aplicaciones automotrices.

En este caso, concierne ver los dispositivos de posicionamiento, en donde se encuentra una gran variedad de soluciones, donde en nuestro interés podemos dividir básicamente

en dos: los de alta precisión y los de precisión estándar, los cuales, además de la posición pueden proveer de diferentes servicios. Anteriormente (véase capítulo 2.8.3), se hizo un análisis sobre diferentes GNSS y, se llegó a que el GNSS a utilizar dadas sus características, servicios y precios es el módulo GNSS NEO-M9N (Figura 44)



Figura 44. Módulo NEO-M9N

Una vez seleccionado el GNSS NEO-M9N como parte del subsistema de posicionamiento, es necesario enlistar sus características más relevantes como se muestra a continuación:

1. Los receptores u-blox M9 atienden aplicaciones de seguimiento industrial y automotriz, como navegación, telemática y UAV.
2. Este tipo de productos son capaces de detectar interferencias y suplantación de identidad. Asimismo, cuentan con algoritmos de filtrado avanzados para mitigar este tipo de problemas.
3. Permite la integración con diferentes interfaces como UART, USB, I2C y SPI.
4. Cuenta con diferentes modos de asistencia para entregar opcionalmente por medio de protocolos HTTP o HTTPS efemérides satelitales inmediatas, información de salud y datos de ayuda, para sistemas receptores con acceso directo a Internet.
5. Los receptores u-blox M9 admiten la recepción simultánea de cuatro GNSS, permitiendo una mejor precisión de hasta 2.0[m] CEP¹⁸.
6. Cuenta con una velocidad de actualización de navegación de hasta 25Hz.
7. Sus límites de operación son de hasta 80,000[m] de altura y una velocidad de 500m/s. En este punto también se asegura una precisión de la velocidad de 0.05m/s para una velocidad de navegación de 30m/s o 108km/h dentro del 50% de las veces.
8. Tiene niveles de operación eléctricos de 3.3[V] con un máximo de 50[mA] de consumo.

¹⁸ CEP por sus siglas en inglés, significa error de probabilidad circular, lo cual indica que el GNSS se encontrará dentro de los dos metros de la medición real en el 50% del tiempo.

Una de las ventajas de este módulo es que de manera comercial se encuentra la placa de desarrollo del GNSS seleccionado, cuyas características principales a considerar son:

1. Cuenta con una interfaz de comunicación por medio de UART para el GNSS.
2. Integra una brújula magnética para comunicar por medio de I2C.
3. Integra una antena cerámica Taoglas con una ganancia máxima de 3.5dBi¹⁹ (Figura 45).



Figura 45. Antena cerámica Taoglas. Tomada de []

Dicho lo anterior, los principales parámetros de operación del GNSS serán dos: El primero es operar en el modo de multi-constelación para mejorar la precisión de la posición; y la segunda es operar con una frecuencia de actualización de 25Hz. Adicionalmente, sería útil utilizar los algoritmos de mitigación de interferencias, sin embargo, el principal punto aquí es de proveer al dron de los suficientes datos para orientarse y mantenerse dentro del curso de navegación.

En este apartado también corresponde ver el altímetro a utilizar para la medición vertical de la posición el dron. Para este punto se utilizará el altímetro incorporado en el FC, cuya información se puede obtener a través de UART. El altímetro que incorpora el FC se conoce como barómetro MEAS MS5611, cuyas especificaciones principales son las siguientes:

- Rango de medición de 10mbar a 1200mbar.
- Precisión de ± 1.5 mbar

En el caso el barómetro sólo es de interés que el altímetro nos pueda proporcionar el rango de medición establecido por la NOM, esto es así, porque en el caso de la altura, esta se puede mantener fija en la mayoría del tiempo, a menos que se necesiten cambios imprevistos.

¹⁹ Los dBi hacen referencia a la ganancia isotrópica que tienen las antenas, es decir, la ganancia en todas las direcciones.

4.3.5.1 Protocolos de comunicación (GNSS)

Debido a que la información del altímetro se puede obtener por medio del protocolo MavLink descrito anteriormente (4.3.2.2), en esta sección únicamente se abordará los protocolos utilizados para la comunicación con el GNSS.

Entre estos protocolos destacan UBX y NMEA, los cuales, se decidió implementar en el proyecto. El protocolo UBX es de manera general una forma para enviar mensajes desarrollado por la misma empresa, cuya característica principal es recuperar información más detallada acerca del dispositivo en sí, es decir, este protocolo funciona de mejor manera para enviar mensaje de la computadora al GNSS, mientras que el protocolo NMEA, es un estándar utilizado para enviar los datos del satélite al GNSS.

Dado que el GNSS únicamente entrega mensajes de salida con el protocolo NMEA, es por esa razón que se implementan ambos protocolos, para obtener la información de los satélites y, por otro lado, para obtener información del mismo dispositivo. La documentación utilizada para implementar esta forma de comunicación se encuentra en [41] y [42].

4.3.6 Subsistema de Detección de objetos

Del estado de arte se sabe que la mayoría de LIDAR 3D integran varios sensores LIDAR en un tamaño reducido con un mecanismo para recolectar puntos del medio ambiente. En este caso, debido al precio de LIDAR, se implementará uno con su propio mecanismo de movimiento para obtener más información del ambiente. Una forma de visualizar y facilitar el diseño de este subsistema es a través de su diagrama funcional en la Figura 46.

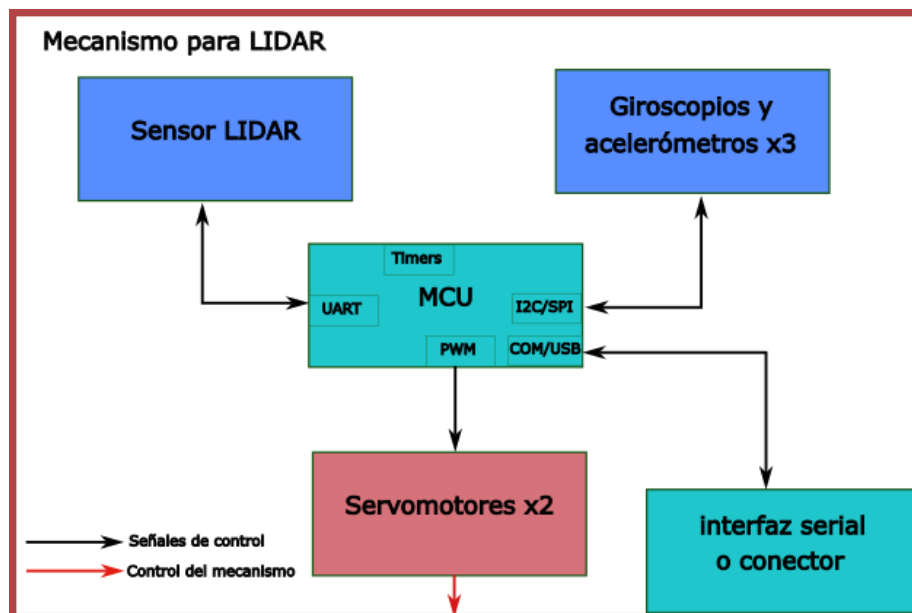


Figura 46. Diagrama funcional para el subsistema de detección con base en LIDAR.

En general, al sensor LIDAR se implementa un microcontrolador para el control del subsistema. En él se controla LIDAR para tomar puntos del ambiente y, estos datos se combinan con la orientación del sensor, indicado por los acelerómetros y giroscopios. Sobre el mecanismo se utilizará uno de los muchos que existen en el mercado para cámaras, cuyo control por lo general es por medio de servomotores, esto para facilitar el diseño. Finalmente, la información recabada se puede enviar a través de comunicación serial como UART o USB.

4.3.6.1 Sensores LIDAR y mecanismo de movimiento

Como sensor principal se escogió el sensor LIDAR TF02-PRO (Figura 47), cuyas características se muestran respectivamente.



Figura 47. Sensor LIDAR TF02-PRO

- Rango de medición de hasta 40[m]
- Resistencia a la luz ambiental (100klux)
- Alta actualización de trama (1000Hz)
- FOV 3°
- Interfaces de comunicación: I2C/UART
- 50[g]

Por su parte, el mecanismo LIDAR, se puede adaptar de las plataformas para cámaras disponibles en el mercado (Figura 48), las cuales, por lo general ya incluyen los servos para el control y, además, permiten el rango de movimiento de 30° de manera horizontal, pero con un rango de 180° de manera horizontal.



Figura 48. Mecanismo para LIDAR

Los servos integrados en el sistema comprado corresponden a MG90 y sus principales cualidades se enumeran para definir el alcance del sistema:

- Peso de 13.4[g]
- Tamaño: 22.5x12x35.5 mm
- Par: 1.8kgcm (4.8V)
- Voltaje de funcionamiento: 4.8-6[V]
- Velocidad angular: 0.1s/60° (4.8V)
- Temperatura de operación: -30 a 60 °C
- Banda muerta: 5μs

Con la finalidad de obtener más puntos de muestra, se planea implementar un sensor LIDAR de 360° que únicamente hace un escaneo en 2D del plano horizontal (Figura 49). Este nos permite tener un poco más de visión atrás del dron con el propósito de disminuir las maniobras del dron.



Figura 49. LIDAR LDS01RR

4.3.6.2 Sensores de movimiento

Los acelerómetros y giroscopios se implementarán por medio de 4 módulos debido a su disposición, además de que están diseñados para este tipo de aplicaciones, donde, algunos de ellos tienen embebidos en un solo sensor el giroscopio y el acelerómetro. Las

especificaciones de los sensores seleccionados se enlistan en la Tabla 16 para facilitar su visualización.

Tabla 16. Parámetros de los sensores de movimiento seleccionados

Dispositivo	Sensor	Rango	Sensibilidad	No linealidad	Densidad de ruido	Velocidad de datos
Módulo L3GD20H	Giroscopio L3GD20H	±245, ±500, ±2000 dps (°/s)	8.75, 17.50, 70.00 mdps/digit	0.2%	0.011 dps/VHz	11.9Hz, 23.7Hz, 47.3Hz, 94.7Hz, 189.4Hz, 378.8Hz, 757.6Hz
Módulo GY-87	Giroscopio MPU-6050	±250, ±500, ±1000, y ±2000 dps (°/s)	131, 62.5, 32.8, 16.4 LSB/dps	0.2%	0.005 dps/VHz	4Hz-8,000Hz
	Acelerómetro MPU-6050	±2g, ±4g, ±8g y ±16g	16,384, 8192, 4,096, 2048 LSB/g	0.5%	400 µg/VHz	4Hz-1,000Hz
Módulo GY-91 Módulo MPU-9250/6500	Giroscopio MPU-9250	±250, ±500, ±1000, y ±2000 dps (°/s)	131, 62.5, 32.8, 16.4 LSB/dps	0.1%	0.01 dps/VHz	4Hz-8,000Hz
	Acelerómetro MPU-9250	±2g, ±4g, ±8g y ±16g	16,384, 8192, 4,096, 2048 LSB/g	0.5%	300 µg/VHz	4Hz-4,000Hz

4.3.6.3 Unidad de procesamiento y placa de evaluación de alto desempeño

De acuerdo con lo investigado del estado del arte, los procesadores de STMicroelectronics tienen mucha aplicación en los dispositivos usados en el dron. Con esto en consideración, se escogerá un microcontrolador de alto desempeño en esta aplicación con LIDAR, donde la notación para este tipo de microcontroladores corresponde generalmente a STM32FXXXXX, por tanto, el dispositivo seleccionado es el microcontrolador STM32F446RE con encapsulado LQFP, cuyas características son:

- Núcleo: Arm® 32-bit Cortex®-M4 CPU con FPU.
- Hasta 114 I/Os con capacidad de interrupción.

- 17 temporizadores, con dos dedicados al control de motores.
- Hasta 512 KB de memoria Flash.
- 128 KB de SRAM.
- Varias interfaces: 4x I2C (SMBus/PMBus), 4x USARTs y 2x UARTs, 4x SPIs, 2x CAN (2.0B Active), 1x SDMMC y USB 2.0
- Otras varias características.

Con la finalidad de facilitar las pruebas del microcontrolador se usará la placa de evaluación Nucleo-F446RE, asimismo, se utilizarán los “Shield” para cada uno de los sensores de movimiento seleccionados en el proyecto.

Posteriormente, con la selección y prueba de los componentes del subsistema se realizarán la fabricación de la PCB y se integrarán todos estos. Por otra parte, se dejará que el subsistema sea programable para que se adecuen los parámetros de acuerdo con diferentes necesidades.

Finalmente, la organización de la trama de datos se realizó con formato LSB y, de acuerdo con como se muestra en la Tabla 17.

Tabla 17. Trama de datos.

Número de bytes	Nombre y/o función
1	Byte de inicio
4	Movimiento en el eje x [°]
4	Movimiento en el eje y [°]
4	Movimiento en el eje z [°]
4	Posición del eje x [°]
4	Posición del eje y [°]
4	Posición del eje z [°]
2	Distancia de Lidar
2	Temperatura del sensor
4	Tiempo
1	Byte de fin

4.3.7 Subsistema de potencia

Contar con una unidad de manejo de potencia adecuada para el dron es una tarea que favorece al desempeño y control de la energía disponible a bordo. Como se ha mencionado, comúnmente, los drones usan baterías de Li-Po, cuyas tensiones van variando de acuerdo con la necesidad del motor y los ESC, proporcionando voltajes diferentes a los requeridos por otros dispositivos.

En general, los voltajes de conversión serán de 12V, 5V y 3.3V para diferentes dispositivos, para ello se usarán los convertidores conmutados (IC) disponibles en el mercado. Existen una gran variedad de categorías para estos reguladores de tensión, pero los de interés del proyecto en general son los convertidores reductores o *step-down*.

Sobre el subsistema se debe aclarar que el regulador de 12V tiene como función principal suministrar la alimentación a dispositivos LIDAR 3D u otros dispositivos compatibles con este nivel de tensión; por su parte los reguladores de 5V y 3.3V suministrarán la energía a los demás dispositivos y subsistemas a bordo, que es lo más común para diferentes sensores, procesadores y dispositivos electrónicos.

De la misma manera, un esquema general de la solución de este subsistema se presenta en la Figura 50 para facilitar la integración de los convertidos integrados y diversos dispositivos requeridos por éste.

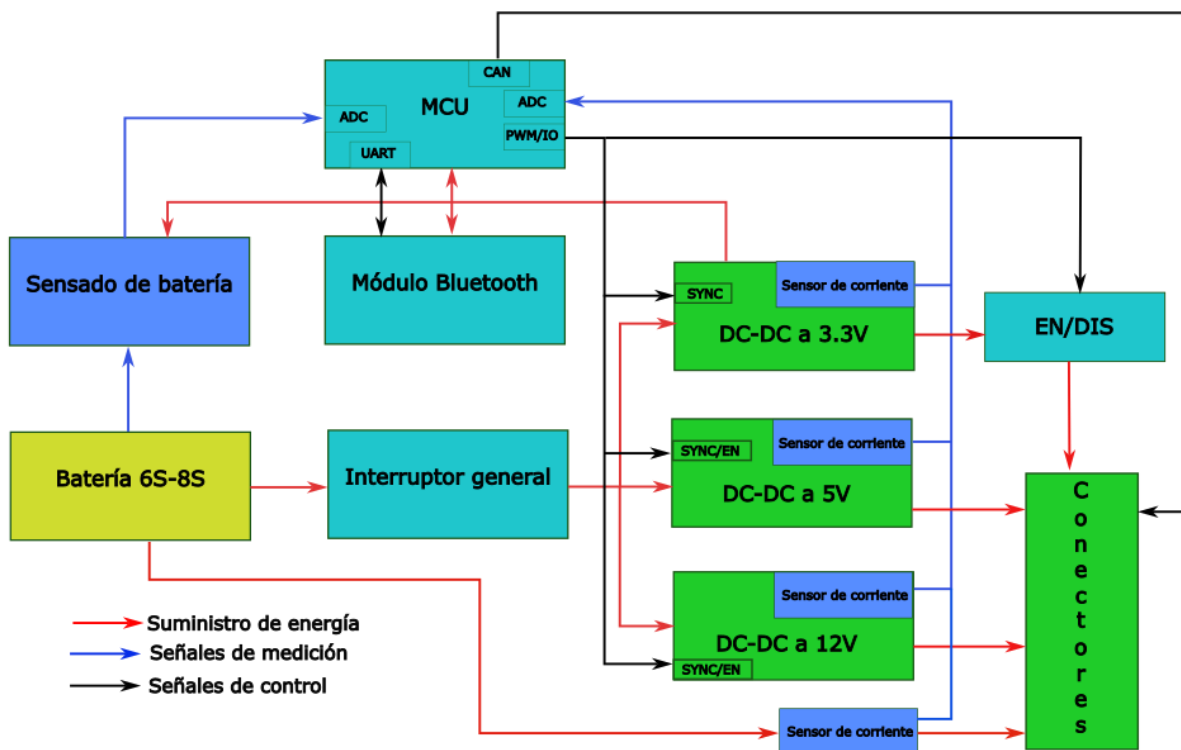


Figura 50. Diagrama funcional de la unidad de manejo de potencia (PMU).

Como se muestra en el diagrama el subsistema, como en la mayoría de los dispositivos, cuenta con un interruptor general para habilitar el funcionamiento de todos los elementos conectados a éste. Asimismo, el subsistema de potencia puede ser controlado por medio

de Bluetooth (de preferencia) o a través de CAN por la computadora de abordo, permitiendo un control remoto sobre el suministro de todos los elementos en el dron repartidor. De igual forma, se incorporan los sensores correspondientes para medir los niveles de voltaje y corriente utilizados en el sistema.

Otro aspecto relevante es que algunos convertidores de conmutación cuentan con sus propias líneas de habilitación y de sincronización²⁰, siendo el convertidor de 3.3V el único al que no se considera la línea de habilitación, porque este energiza directamente al procesador encargado de controlar el subsistema, sin embargo, para otros casos se consideran interruptores externos como los relevadores para generar ese control extra a partir del microcontrolador.

4.3.7.1 Reguladores de conmutación IC

Para seleccionar los convertidores de voltaje más adecuado es importante hacer un estimado sobre la carga requerida en este mismo. En este estudio se consideran todos los elementos incorporados en el sistema y se les agrega hasta un porcentaje extra de uso en la carga, con lo cual en la Tabla 18, se presenta a manera de resumen las cargas estimadas

Tabla 18. Carga estimada para cada convertidor

Convertidor de voltaje	Carga estimada ²¹
DC-DC a 3.3V	375[mA]
DC-DC a 5V	5[A]
DC-DC a 12V	1.5[A]

La estimación anterior, permite escoger correctamente a los convertidores que desempeñan una mejor eficiencia en la conversión del voltaje requerido, debido a que esta eficiencia varía con la diferencia de conversión (ej. 22.2V a 3.3V) y con la carga o corriente requerida por los elementos conectados. Algunas opciones para mejorar la eficiencia es usar fuentes conmutadas conectadas en cascada, por ejemplo, hacer la conversión desde la fuente de 22.2V a 5V y, después convertir de la de 5V a 3.3V, lo cual aumentaría la eficiencia a hacerla de manera directa, sin embargo, en este proyecto no se consideró

²⁰ La señal de sincronización se utiliza para enlazar con la frecuencia de conmutación.

²¹ En la estimación de la carga se hizo un análisis utilizando los parámetros de potencia de los dispositivos utilizados en el proyecto, además a estos se les agregó un porcentaje consumo extra.

para permitir la habilitación del convertidor de 5V directamente desde su terminal de habilitación y evitar hardware extra.

Con los datos de la Tabla 18, se presenta la siguiente Tabla 19, con el propósito de mostrar de manera simple y ordenada las características de los circuitos integrados seleccionados para regular las diferentes salidas de voltaje necesarias.

Tabla 19. Característica de los convertidores conmutados seleccionados

Salida regulada	Dispositivo	Voltaje de entrada	Frecuencia de operación	Corriente de operación	Eficiencia	Sincronización
De 3[V]	LT8609	3[V]-42[V]	200[kHz]-2.2[MHz]	3[A]	93% para conversión de 12[V] a 5[V]	Sí
De 5[V]	LTC7821	10[V]-72[V]	200[kHz]-1.5[MHz]	Depende de sus elementos	97% para conversión de 42[V] a 12[V]	Sí
De 12[V]	LT8610	3.4[V]-42[V]	200[kHz]-2.2[MHz]	2.5[A]	96% para conversión de 12[V] a 5[V]	Sí

Cada uno de estos circuitos integrados se seleccionó por su desempeño a diferentes cargas, considerando los aspectos descritos anteriormente. Por consiguiente, se enlistan rápidamente los criterios de selección.

- El integrado LTC7821 se escogió en el convertidor de 5[V] por su alta eficiencia para diferencias de conversiones muy altas y con un consumo de corriente mayor a 2[A]. Además, su óptima eficiencia se obtiene cuando $V_{out} \cong 4V_{in}$, un valor cercano la conversión de 22.2[V] a 5[V].
- En el caso del convertidor de 12[V] se empleará el convertidor LT8610, por su alta eficiencia y estabilidad a diferentes valores de carga utilizados en su rango de posibilidad.
- Y, por último, para el regulador de 3.3[V] se seleccionó el integrado LT8609, principalmente por desempeñar altas eficiencias en corrientes menores a 1[A].

Los cálculos realizados para la selección de los componentes en cada uno de los reguladores de voltajes se obtuvieron de acuerdo con el datasheet de cada dispositivo y, estos se muestran en los apéndices: APÉNDICE I, APÉNDICE II y APÉNDICE III.

4.3.7.2 Unidad de procesamiento de bajo consumo y placa de evaluación

La unidad de procesamiento para este subsistema también será STM32, atendiendo a lo mencionado en el capítulo 4.2.8. Entonces, los microcontroladores de bajo consumo se numeran como STM32LXXXXX y, además, como se necesita conectividad por medio de CAN, el dispositivo debe ser de arquitectura ARM Cortex-M4 o superior, debido a que se ha encontrado en la página de STMicroelectronics que las arquitecturas inferiores a la descrita no cuentan con CAN, por tanto, la notación queda como STM32L4XXXX y, el dispositivo seleccionado es el microcontrolador STM32L432KC con encapsulado QFJ., con las siguientes características.

- Núcleo: Arm® 32-bit Cortex®-M4 CPU con FPU.
- Hasta 26 I/Os rápidos.
- 8 temporizadores.
- Hasta 256 KB de memoria Flash.
- 64 KB de SRAM.
- 1x 12-bit ADC 5 Msps, hasta 16-bit con sobremuestreo y 10 canales, VDDA: 1.6[V] a 3.6[V].
- Varias interfaces: 1x SAI (serial audio interface), 2x I2C FM+(1 Mbit/s), SMBus/PMBus, 2x USARTs (ISO 7816, LIN, IrDA, modem), 1x LPUART (Stop 2 wake-up), 2x SPIs (and 1x Quad SPI), CAN (2.0B Active), SWPMI single wire protocol master I/F, IRTIM (Infrared interface).
- Otras varias características.

Si se escoge un valor de $VDDA = 3.3[V]$ con referencia a $0[V]$, la resolución del ADC será la siguiente:

$$resolución = \frac{3.3[V] - 0[V]}{2^{12}} = 800\mu V$$

Como último punto, para facilitar el tiempo de desarrollo, al igual que con el subsistema de detección de objetos, se usará la placa de evaluación Núcleo-L432KC del microprocesador seleccionado, sin embargo, para el caso de los reguladores y sensores se fabricará una PCB de prueba, debido a que no se cuentan con los “shield” de estos dispositivos debido a su poca accesibilidad y precio.

4.3.7.3 Medición de voltaje y corriente

Los dispositivos en la medición deben ser escogidos adecuadamente para que soporten las condiciones de operación establecidas con anterioridad. En este punto, lo principal es definir la manera de medir el nivel de tensión y corriente en la batería, aunque específicamente para la corriente se medirá individualmente en cada uno de los reguladores y cargas como se muestra en la Figura 50.

4.3.7.3.1 Medición de voltaje a través del divisor de tensión

Dicho lo anterior, de acuerdo con el estado del arte, la manera más común en la medición del voltaje de una fuente de DC es por medio de un divisor de voltaje, el cual, puede ser acondicionado por un filtro paso-bajas para mejorar su desempeño.

Aquí se considera cómo el valor resistivo afecta en cierta medida al consumo energético de la batería para este divisor, es decir, en cuanto mayor sea el valor, menor será la corriente circulante por el circuito, por ello, en este caso, se diseñará un divisor con un consumo de aproximadamente $1[\mu A]$, permitiendo conectar diferentes tamaños de batería de 6S a 8S.

$$V_i = R_{Eq} \times I_T \quad \text{ec. (52)}$$

$$R_{EqMin} = \frac{22.2[V]}{1 \times 10^{-6}[A]} = 22.2[M\Omega]$$

$$R_{EqMax} = \frac{29.6[V]}{1 \times 10^{-6}[A]} = 29.6[M\Omega]$$

Entonces, si se quiere tomar la medición de un resistor de $3.3[M\Omega]$, para fijar un voltaje de salida de $3.3[V]$, el resto de las resistencias tendrán un valor de $26.3[M\Omega]$. Además, se debe considerar la diferencia entre las resistencias máxima y mínima para poder ajustarlo a la batería, la cual resulta de $7.4[M\Omega]$, entonces como mínimo se debe tener un potenciómetro de ese valor, para ello se escogieron dos potenciómetros de $5[M\Omega]$ por su disponibilidad. Posteriormente, esto deja a los resistores fijos con un valor de $16.3[M\Omega]$ y, dado que en el mercado existen resistencias de $8.2[M\Omega]$, se usarán dos de estas.

Ahora lo importante es observar la potencia disipada por cada resistor, donde, en general el circuito equivalente disipará una potencia máxima de $29.6[\mu W]$, con lo cual, resistencias con valores menores a $1/10[W]$ funcionarán correctamente en el divisor y, este último se muestra en la Figura 51 con el valor de las resistencias calculadas.

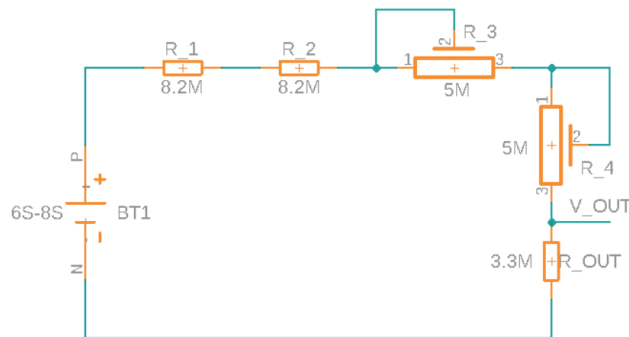


Figura 51. Diagrama del divisor de voltaje.

Este circuito permite si bien la posibilidad de colocar una batería con mayor tensión, pero a la vez se debe recordar que el cambio de peso de batería afecta en el tiempo del vuelo al sistema en general, además de que los motores deberán ser capaces de operar a ese nivel de voltaje.

Un punto relevante por considerar en el divisor de voltaje es el efecto de carga que ocurre cuando se conecta algún dispositivo para monitorear el voltaje de salida. Lo anterior puede ocasionar que el voltaje monitoreado sea distinto si no se considera este efecto.

Para determinar cómo influye lo anterior, se ocupará el teorema de Thevenin con la finalidad de obtener un circuito equivalente para la Figura 51, el cual se representa por una fuente de voltaje fija conectada a una resistencia en serie.

Como primer punto se considera el análisis para el circuito abierto de la Figura 51. Y, para obtener la resistencia de Thevenin la fuente independiente de voltaje se considera como cortocircuito. Con lo anterior, se obtiene el circuito mostrado en la Figura 52.

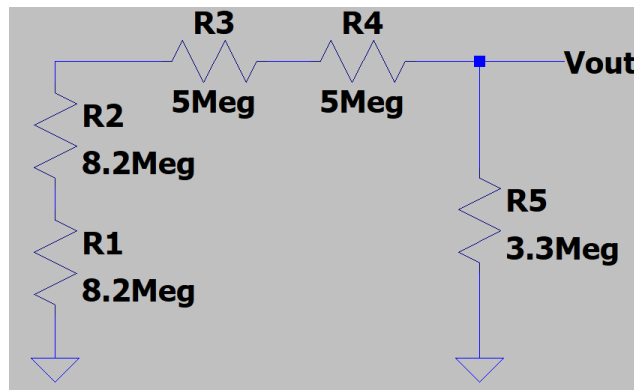


Figura 52. Circuito eléctrico usado para determinar la resistencia equivalente de Thevenin

En esta Figura 52 se observan varias resistencias conectadas en serie, cuya resistencia equivalente se encuentra en paralelo con la resistencia de $3.3[M\Omega]$. Entonces, la resistencia de Thevenin se calcula como,

$$R_{TH} = (R_1 + R_2 + R_3 + R_4) || (R_5)$$

$$R_{TH} = (26.4[M\Omega]) || (3.3[M\Omega]) = 2.93[M\Omega]$$

Y, el voltaje de Thevenin se obtiene calculando el voltaje V_{out} , el cual, como se definió al inicio corresponde a $3.3[V]$. De igual forma, este se puede calcular con un divisor de voltaje.

$$V_{TH} = \frac{R_5}{R_1 + R_2 + R_3 + R_4 + R_5} (V_{in}) \quad ec. (53)$$

$$V_{TH} = \frac{3.3[M\Omega]}{29.6[M\Omega]} (29.6[V]) = 3.3[V]$$

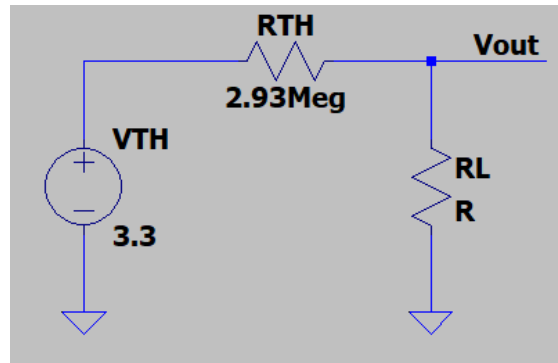


Figura 53. Circuito equivalente de Thevenin con la carga conectada.

El equivalente de Thevenin se muestra en la Figura 53 y a este se le añade la carga para ver el efecto sobre el voltaje que se debe medir. Entonces, se analiza el voltaje de salida usando un divisor de voltaje y se obtiene lo siguiente:

$$V_{OUT} = \frac{R_L}{R_{TH} + R_L} (V_{TH}) \quad ec. (54)$$

$$V_{TH} = \frac{R_L}{2.93[M\Omega] + R_L} (3.3[V])$$

Aquí se puede analizar que si el valor de R_L es significativamente mayor a $R_{TH} = 2.93[M\Omega]$, es decir, si $R_L \gg R_{TH}$, hace que el cociente se aproxime a,

$$\frac{R_L}{2.93[M\Omega] + R_L} \approx 1$$

Y, por tanto,

$$V_{TH} \approx 3.3[V]$$

Lo anterior es lo deseado y nos indica que cualquier dispositivo que se conecte al nodo V_{out} no modificará este valor, siempre y cuando $R_L \gg R_{TH}$.

4.3.7.3.2 Medición de corriente a través de resistencias shunt y amplificadores de detección de corriente

Por otra parte, los sensores de corriente utilizados se implementarán por medio de resistencias shunt junto con los amplificadores de detección de corriente (estos son de

propósito especial y se consiguen en el mercado). Estos resistores se configurarán en el lado alto del sistema en cada una de las salidas: motores brushless y reguladores de voltaje de 3.3[V], 5[V] y 12[V]. Estos se enlistan en la Tabla 20 con la finalidad de mostrarlos de una manera más compacta, así mismo, en la Figura 50 se muestra a nivel sistema cómo se interconecta cada una de las resistencias shunt usadas en los sensores de corriente. En este caso las resistencias shunt se colocan a la salida de los convertidores de voltaje de 3.3[V], 5[V] y 12[V] para determinar la corriente consumida por la carga conectada a cada convertidor. Además, se conecta una resistencia shunt entre la fuente y los conectores de los ESC, la cual se usará para determinar la corriente suministrada por los ESC que proporcionan las señales de control requeridas por los motores brushless.

En la selección de las resistencias se consideró el rango de medición máximo deseado y la sensibilidad del dispositivo para medir décimas, centésimas y milésimas de amperios, de acuerdo con cada salida de voltaje. Por ejemplo, para el caso de la salida a los ESC y el regulador de 3.3[V] se escogió una corriente máxima de 125[A]²² y 0.75[A], respectivamente, para una tensión de salida de 3[V] en el amplificador. Después, a partir de la Ley de Ohm se obtuvo una resistencia equivalente (R_{Esc}) y dependiendo de la ganancia deseada se calculó la resistencia correspondiente a cada caso, considerando que en cuanto menor sea ésta, menor será la potencia consumida, sin embargo, la ganancia tiende a aumentar para desempeñar el rango de medición, esto se puede ver con la siguiente ecuación.

$$V_{Amp} = G * V_{Shunt} = G * R_{shunt} * I_{Carga} = R_{Esc} * I_{Carga} \quad ec. (55)$$

Donde:

- V_{Amp} es el voltaje amplificado
- V_{Shunt} es el voltaje en la resistencia shunt
- G es la ganancia del amplificador
- R_{Esc} es la resistencia escalada por la ganancia

²² La selección de la corriente máxima a medir se fundamenta en la tabla de desempeño de los motores brushless proporcionada por el fabricante en [73]. La selección de la corriente en los motores se obtuvo atendiendo el diseño del capítulo 4.3.3, donde los principales parámetros considerados son: las hélices de tamaño “15x5.5”, un peso de empuje de aproximadamente 10kg y, una batería de 22.2V. Por ende, cada uno de los motores operando bajo las condiciones anteriores consumen una corriente de 20.2[A] y proporcionan un empuje de 2650[g], lo cual, para 4 motores corresponde a una corriente de 80.8[A] y un empuje total de 10.6[kg]. Se debe mencionar que, se escogió el valor de 125[A] porque esto permite escalar el sistema a un hexacóptero (6 motores) que es más seguro de acuerdo con los estándares descritos anteriormente.

Otro aspecto considerado fue la resolución del dispositivo. Esta se puede establecer de dos maneras, si bien, por medio de utilizar la resistencia calculada R_{shunt} , o definiendo la resolución y, calculando con ello el rango y la resistencia que puede ofrecer. Este enunciado se puede observar con las siguientes ecuaciones:

$$R_{Shunt} = \frac{V_{OutAmpMin}}{I_{LoadMin} * G} \quad ec. (56)$$

$$R_{Shunt} = \frac{V_{OutAmpMax}}{I_{LoadMax} * G} \quad ec. (57)$$

$$\frac{V_{OutAmpMax}}{I_{LoadMax}} = \frac{V_{OutAmpMin}}{I_{LoadMin}}$$

$$I_{LoadMax} = \frac{V_{OutAmpMax}}{V_{OutAmpMin}} I_{LoadMin} \quad ec. (58)$$

Con la ecuación anterior se pueden definir ciertos valores mínimos para la medición, donde $V_{OutAmpMax} = 3[V]$, debido a que es casi el límite de lo que puede medir el convertidor ADC, por consiguiente, sólo quedaría escoger $V_{OutAmpMin}$ e $I_{LoadMin}$, una vez definidos, se puede calcular el rango capaz de desempeñar y, a su vez, la resistencia a utilizar.

Tabla 20. Sensores de corriente.

Salida	Dispositivo	Características	Parámetros de diseño ²³
De 3[V]	Resistencia shunt: CFN1206AFZ-R040ELF	Resistencia 40mΩ, Potencia nominal 1[W]	Rango máx. a 3V: 0.75A Resolución: 4mV → 1mA Ganancia: 100V/V
De 5[V]	Resistencia shunt: WSLP12066L000FEA	Resistencia 6mΩ, Potencia nominal 1[W]	Rango máx. a 3V: 10A Resolución: 3mV → 10mA Ganancia: 50V/V
De 12[V]	Resistencia shunt: ERJ-6CWDRO30V	Resistencia 30mΩ, Potencia nominal 0.5[W]	Rango máx. a 3V: 2A Resolución: 15mV → 10mA Ganancia: 50V/V
22.2[V]	Resistencia shunt: LRMAP5930B-R0003FT	Resistencia 300μΩ, Potencia nominal 7[W]	Rango máx. a 3V: 100A Resolución: 1.5mV → 50mA Ganancia: 100V/V

²³ Algunos rangos de medición se escogieron de esta manera para aprovechar las capacidades de operación de los reguladores de conmutación.

Además, el sensor de corriente requiere un amplificador de alta ganancia para adecuar el valor de lectura en el ADC. Como se mencionó en el capítulo 3.6.3 se utilizarán los amplificadores de propósito especial “current sense amplifier” y, el error en la ganancia se determinará de acuerdo con la ganancia y la resolución de salida de medición. De acuerdo con la siguiente ecuación.

$$V_{out_{max,min}} = (V_{shunt} \pm V_{os}) * G \left(1 \pm \frac{E_G}{100} \right) \quad ec. (59)$$

Donde:

- $V_{out_{max,min}}$ es el voltaje de salida máxima o mínima que puede proporcionar el amplificador.
- V_{shunt} es la caída de tensión en la resistencia shunt
- V_{os} es un offset de voltaje, cuyo valor se proporciona en las hojas de datos
- G es la ganancia del amplificador
- E_G es el error de ganancia del amplificador

Entonces, considerando la resolución de medición establecida con anterioridad de cada amplificador y la salida ideal, se puede establecer un voltaje de salida máxima y mínimo, mientras que las demás variables igualmente se conocen. Siendo por consiguiente el error de ganancia el que se debe buscar. Tomando esto en consideración y procurando mantener una caída de voltaje cercana al 1% o menor, se encontró que los amplificadores cuyos requerimientos se aproximan a los deseados se enlistan en la Tabla 21.

Tabla 21. Amplificadores de detección seleccionados

Salida monitoreada	Amplificador de detección de corriente	E_G típico y máximo	V_{os} típ. y máx.	CMRR	Swing voltage	VCM	VCC
3.3[V]	INA199B2	$\pm 0.03\%$ - $\pm 1.5\%$	$\pm 5\mu V$ - $\pm 150\mu V$	100-120	(V+)-0.05 (VGND)+ 0.005	-0.3 - 23[V]	2.7-26V
5[V]	INA185A2	$\pm 0.05\%$ - $\pm 0.2\%$	$\pm 25\mu V$ - $\pm 130\mu V$	96-100	(V+)-0.02 (VGND)+ 0.0005	-0.2-26[V]	2.5-5.5V
12[V] y 22.2[V]	INA186A2/A3	-0.04 - $\pm 1\%$	-3 μV - $\pm 50\mu V$	120-150	(V+)-0.02 (VGND)- 0.00005	(GND-0.2) - 40[V]	1.7-5.5V

En general, se decidió seleccionar estos dispositivos, porque su ganancia de error corresponde con los valores deseados, de acuerdo con los parámetros requeridos mostrados en la Tabla 21. Sin embargo, existen factores que disminuirán los parámetros de este amplificador, debido a los filtros requeridos para los convertidores de conmutación, ya que estos disminuyen la ganancia del amplificador.

4.3.7.4 Consideraciones del acondicionamiento de las señales de medición

Un punto importante al realizar la medición del voltaje y corriente, la cual se analizó en la metodología en el capítulo 4.3.7.3, es acondicionar la señal que se medirá con los convertidores analógicos, esto se realiza generalmente utilizando un filtro paso-bajas para disminuir el ruido ocasionado por diferentes fuentes. En el caso del divisor de voltaje mostrado en la Figura 51, se hizo el análisis para obtener su circuito equivalente de Thevenin, donde la impedancia de salida corresponde a la resistencia de Thevenin y este valor es de $2.93M\Omega$, esto en principio es de gran relevancia, porque si se conecta algún dispositivo de medición o algún filtro a este nodo, se debe asegurar un correcto acoplamiento de las impedancias, porque de otro modo, al conectar el instrumento de medición, este cambiará el voltaje de la salida que se desea medir. Para resolver este problema, se puede considerar diseñar un filtro que tenga una impedancia de entrada varias veces mayor a la impedancia de salida del divisor de voltaje, es decir, $Z_{IN_OPAMP} \gg 2.9M\Omega$, esta desigualdad dependiendo del criterio se puede interpretar de la siguiente forma $Z_{IN_OPAMP} > (10)2.9M\Omega$, es decir, se considera que diez veces el valor es comparablemente mayor aunque este criterio puede variar, sin embargo, esto tiene varias implicaciones en el diseño del filtro que puede hacer que no funcione correctamente, para ello, es preferible conectar en la salida del divisor del voltaje un seguidor de voltaje para acoplar correctamente la carga con el filtro (debido a su alta impedancia de entrada). Para este tipo de filtro se considerarán los principios de diseño del capítulo 3.6.3.1, donde se requiere la configuración de alta impedancia de entrada con una baja impedancia de salida para el ADC.

A diferencia del divisor de voltaje, la señal de medición en las resistencias de derivación requiere filtros debido al ruido ocasionado por los mismos reguladores de voltaje. Una de las características de este filtro es que deben ser amortiguados, el filtro más común y sencillo que se recomienda en este tipo de aplicaciones es un filtro RC a la entrada del amplificador de detección de corriente y es el que muestra en la Figura 23. Este filtro en el caso de los amplificadores de detección de corriente disminuye la ganancia y además añaden ruido al sistema. Antes de definir estos filtros, se debe realizar la caracterización de las señales a medir y, con ello establecer una frecuencia de corte para el filtro RC.

4.3.7.5 Interruptores manuales y controlados

Sobre los interruptores se usarán únicamente dos: uno de control manual y otro por medio de señales de control, estos se muestran en la Tabla 22, para distinguirlos de manera más sencilla. El interruptor controlado a diferencia del manual requiere una breve consideración respecto a la caída de voltaje, si bien, el interruptor de carga o *load switch* es un dispositivo que utiliza transistores para el control de encendido y apagado, sin embargo, éste genera una diferencia de potencial y, a su vez, un consumo de potencia. Este valor está definido como “resistencia de encendido” u “*on-Resistance*” y se calcula de la siguiente manera, con la Ley de Ohm.

$$R_{ON} = \frac{\Delta V_{max}}{I_{Load}} \quad ec. (60)$$

Donde:

- ΔV_{max} es la máxima caída de tensión de V_{in} a V_{out}
- I_{Load} es la corriente de la carga
- R_{ON} es resistencia de encendido del dispositivo para un voltaje de entrada dado

En este caso, se considerará una caída de tensión del 1% para el regulador de 3.3V, siendo este 33mV, con ello se calculará la resistencia de encendido máxima a implementar.

$$R_{ON} = \frac{33mV}{400mA} = 82.5m\Omega$$

Tabla 22. Interruptores seleccionados.

Dispositivo	Tipo	Voltajes de operación	Corriente de operación
LRA511-RS-B/012V	Manual	6[V] a 30[V]	10[A] a 16[A]
SLG5NT1594V (28.5 mΩ)	Señal de control	1.5[V] A 5.5[V]	1[A] con un pico de 1.5[A].

4.3.7.6 Bluetooth

El dispositivo de comunicación inalámbrica Bluetooth se implementará por medio del módulo HC-05, la principal razón de escoger éste fue porque ya se contaba con uno de ellos. Consecuentemente, las características de este dispositivo se enlistan continuación.

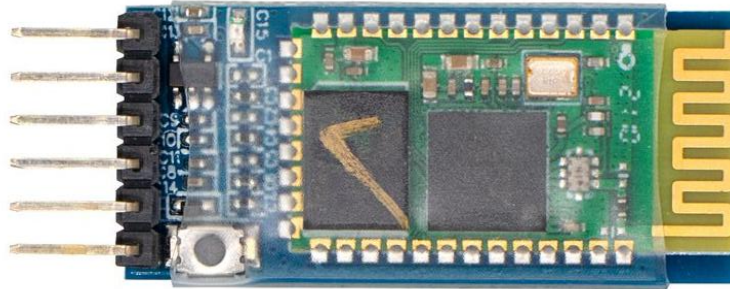


Figura 54. Módulo Bluetooth HC-05

- Bajo consumo Low Power 1.8V Operation.
- Interfaz UART con baudrate programable.
- Antena integrada.
- Chip: BC417143.
- Potencia de transmisión: 4dBm, Clase 2 (hasta 10[m]).
- Sensibilidad: -84dBm a 0.1% BER.
- Consumo Corriente: 30 mA a 50mA.
- Configuración: Comandos AT.
- Protocolo Bluetooth: Bluetooth Especificación v2.0+EDR.
- Frecuencia de banda: ISM 2.4GHz.
- Modulación: GFSK.

4.3.7.7 Batería

Otro aspecto básico para considerar de acuerdo con el tiempo de operación establecido de 10 minutos es la batería, un análisis rápido para dimensionar la capacidad de la batería es utilizando la estimación hecha con anterioridad para los convertidores y el consumo de los motores seleccionados en el apartado 4.3.3, dando como resultado un consumo máximo del sistema de 87[A] en estado estacionario, en donde básicamente los motores consumen la mayor parte de la energía.

Para este caso se pretender utilizar una batería de la marca Tattu de 6S, debido a su disponibilidad en el mercado y a su implementación en drones. Entonces, si se considera el tiempo de operación de 10 minutos, se puede verificar que la capacidad ("C nominal") de la batería seleccionada proporcionará el tiempo adecuado de vuelo, justo como se

especifica en [21], [43], [44], [45] y [46]. Usando la ec. 16 y, considerando la corriente requerida de 87[A], se llega al siguiente valor

$$t_{Batería}(h) = \frac{16Ah}{87[A]} = .184[h]$$

$$t_{Batería} \approx 11[min]$$

En este caso, se hace constar que de manera comercial existen baterías de 14000mAh, 16000mAh, 22000mAh, etc., sin embargo, la batería que más se ajustó al tiempo requerido es la de 16Ah, aunque de manera práctica esta batería proporcionará un tiempo muy cercano o menor a los 10 minutos propuesto. Aun con lo anterior, por motivos de presupuesto y al precio de estas baterías, se usará la batería de 16Ah a pesar de que el tiempo de vuelo esté muy justo con el requerido. Por consiguiente, la batería aproximada para el proyecto se muestra en la Figura 55 y sus capacidades se enlistan de igual manera.



Figura 55. Batería LiPo TATTU

- Capacidad: 16000mAh
- Voltaje: 6S (22.2V)
- Máxima descarga continua 240A

Como últimos elementos a seleccionar se retoman los conectores, que permitan una mayor facilidad al momento de conectar los subsistemas, para ellos se usarán los conectores molex, donde únicamente se verificará que el cable soporte el amperaje requerido por cada subsistema.

Finalmente, con todas las características anteriores, sigue el diseño de la PCB de todos los dispositivos que conformarán al subsistema de potencia, en este caso, se recomienda desarrollar una tabla de comandos para indicar al usuario final la forma de comunicarse y operar con el subsistema de potencia a través del bus CAN.

4.3.8 Computadora de abordó

La computadora de abordó como se ha mencionado es uno de los elementos más importantes, por esa razón, conociendo todos los subsistemas incorporados y la función de cada uno de ellos, se puede diseñar una computadora a la medida de acuerdo con la aplicación desarrollada, por ese motivo, se decidió dejar como último subcapítulo el diseño de la computadora de abordó.

Como se ha expuesto frecuentemente, la detección de objetos implementada por medio de LIDAR a comparación de usar cámaras, se distingue en requerir una menor capacidad de procesamiento. Por esa razón, se enfatiza la importancia de considerar los algoritmos requeridos de acuerdo con los sensores seleccionados, con el propósito de obtener diseños más a la medida. Si bien, en este caso la computadora de abordó se escaló para la visión con LIDAR con lo cual, se espera un funcionamiento adecuado para la mayoría de estos dispositivos, sin embargo, si se deseara implementar la visión artificial por cámara, se necesitaría conocer los algoritmos para saber si la computadora de abordó tiene la suficiente capacidad de procesamiento en la detección de objetos.

Además de lo anterior y los requerimientos del capítulo 4.2.7, se debe enfatizar que los fabricantes también dan recomendaciones sobre el rango de aplicaciones de los microprocesadores. Siendo el grado de aplicación automotriz el requerido para llevar este proyecto a los estándares de seguridad y fiabilidad requeridos.

De manera, sencilla en el grado de aplicación automotriz se realizan ciertas pruebas bajo estándares para asegurar que los componentes pueden soportar las duras condiciones creadas en un ambiente automotriz, estas pruebas, por lo general consisten en cómo se desempeñan los componentes a ciertos rangos de temperatura, designando al grado automotriz el rango de -50°C a 150°C , lo cual, a su vez, establece el tipo de aplicaciones a desarrollar por el dispositivo. Para más información sobre las especificaciones del grado automotriz y aplicaciones, se pueden revisar las referencias [47] y [48].

Dicho lo anterior, se seleccionó la unidad de procesamiento de la empresa STMicroelectronics STM32MPC151C con encapsulado BGA, cuyas características se enlistan a continuación.

- 32-bit Arm® Cortex®-A7 hasta 800MHz
- 32-bit Arm® Cortex®-M4 con FPU/MPU hasta 209MHz
- 708 Kbytes de SRAM
- Hasta 176 I/O puertos con capacidad de interrupción
- Hasta 25 temporizadores
- Varias interfaces: 6x I2C FM+(1 Mbit/s, SMBus/PMBus), 4x UART + 4x USART (ISO 7816, LIN, IrDA, SPI), 6x SPI (and 1x Quad SPI), 3x SDMMC, 2x USB 2.0.

- Otras características

Su selección en parte se debió, además del bajo costo (entre algunas opciones), al conocimiento y empleo de los microcontroladores de esta empresa, con lo cual, en lugar de iniciar desde cero en el entendimiento y manejo de las herramientas de desarrollo de otra empresa, se optó por elegir la unidad de procesamiento de esta empresa.

Además, para facilitar el desarrollo y pruebas, así como disminuir costos, se decidió por implementar los periféricos en un “System-On-Module (SOM²⁴, por sus siglas en inglés)” MYC-YA151C correspondiente a la unidad de procesamiento seleccionado. De la misma manera, sus especificaciones se muestran a continuación.

- ST STM32MP1 MPU
- 256MB/512MB DDR3L, 256MB Nand Flash/4GB eMMC Flash, 32KB EEPROM
- Power Management IC (PMIC)
- Soporta Linux

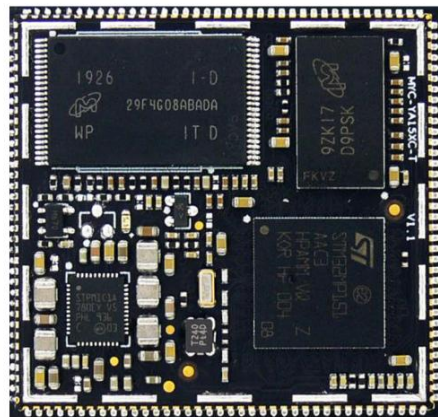


Figura 56. MYC-YA151C

El único problema de este dispositivo es no contar con la interfaz de CAN en el dispositivo, sin embargo, existen circuitos integrados que permiten implementar CAN a través de la conexión por medio de otro protocolo como SPI, por tanto, para esto se escogió el integrado MCP2515 para expandir las capacidades de la computadora de abordó respecto a CAN 2.0 a través de SPI.

²⁴ “System on Module” o SOM se refiere a una placa de evaluación que proporciona los componentes centrales de un sistema de procesamiento embebido, incluyendo el núcleo del procesador, interfaces de comunicación y bloques de memoria, entre otros.

Además, para facilitar aún más el trabajo, se usará una placa de expansión de desarrollo denominada “Serial CAN Bus Module” y, con esto los transceptores requeridos por CAN BUS quedan definidos por la misma placa. Las características de este producto se enlistan y en la Figura 57 se muestra la placa de evaluación usada.



Figura 57. Módulo Serial CAN BUS

- Interfaz CAN a serial
- Basado en los dispositivos MCP2521 y MCP2515
- Velocidad de transferencia de hasta 1Mb/s
- Voltaje de operación 3.3[V]-5[V]

Si bien, el microprocesador no apunta a aplicaciones del tipo “automotriz”, no obstante, el dispositivo encaja con todas las características mínimas requeridas tanto en procesamiento, interfaces y costo (39-48 dólares) indispensable en la computadora de abordo y el sistema desarrollado en cuestión, además de que se trata de un prototipo. Si se deseara que cumpliera con estos estándares, se podría optar por otros dispositivos de precio muy superior con características similares a lo tratado en la tesis como, por ejemplo, el módulo OSD32MP1-BRK, donde por un precio de casi 91 dólares, aun le hace falta la integración de las memorias. Asimismo, otra posible opción es el módulo SAM9X60D1G-I/LZB que por un precio de casi 82 dólares permite alcanzar una velocidad máxima de operación con un núcleo de 600MHz, ajustándose más a los requerimientos. Aquí la problemática surge con las limitantes de la cantidad de SOM disponibles para desempeñar las funciones de prototipo, lo cual ayuda a reducir el tiempo de desarrollo y ajustar los requerimientos del sistema, con lo cual, hacer una selección adecuada del procesador y adaptarla a una PCB, podría ser una opción más viable, siendo los microprocesadores de la empresa NXP los más adecuados para las aplicaciones mencionadas en este proyecto.

4.3.8.1 Implementación de Software

En la implementación del software y los algoritmos de control para el funcionamiento del sistema, se deben considerar otras características inherentes del dispositivo seleccionado.

Los principales puntos para mencionar es la memoria ROM y las posibilidades de implementar el *bootloader* en la computadora de abordó.

Como se sabe, en los microcontroladores o microprocesadores el primer código en ejecutarse cuando se alimenta con energía el dispositivo es el código contenido en la memoria ROM, la cual, no se puede modificar y es la que provee el fabricante. Este código almacenado en la ROM permite seleccionar la interfaz de arranque desde donde será cargada la primera etapa del bootloader en la memoria RAM del sistema. Para definir la interfaz, se utilizan señales de selección a través de interruptores justo como se enlista debajo

Tabla 23. Selección de bootloader

BOOT2	BOOT1	BOOT0	Modo inicial de arranque
0	0	0	USART2/3/6, UART4/5/7/8 y USB
0	0	1	Flash Serial NOR
0	1	0	eMCC
0	1	1	Flash NAND
1	0	0	Acceso a debug
1	0	1	Tarjeta SD
1	1	0	USART2/3/6, UART4/5/7/8 y USB
1	1	1	Flash serial NAND

En la Tabla 23 se presentan los diferentes modos para el arranque de la computadora, los cuales se seleccionan por medio de resistencias pull up en los pines conocidos como BOOT0, BOOT1 y BOOT2, donde la combinación de estas 3 entradas indicará cómo arrancará la computadora.

En el desarrollo y diseño de los archivos a implementar, se deben considerar las siguientes especificaciones proporcionadas por el fabricante:

- Para que el código ROM lo reconozca correctamente, el archivo binario cargado desde el dispositivo de arranque debe cumplir con un formato propietario, comenzando con un encabezado STM32.
- Las líneas STM32MP15x cumplen con el encabezado STM32 v1.

4.3.8.2 Herramientas de desarrollo de software

Existen varias herramientas de desarrollo que facilitan el uso y manejo de los microprocesadores y microcontroladores de esta empresa, las cuales se enlistan y se explican de manera breve a continuación:

- STM32CubeMX o STM32CubeIDE: Ambos son entornos de desarrollo para aplicaciones con los dispositivos de la empresa STM, la diferencia entre ambos es que uno contiene más herramientas para el desarrollo, pero ambos tienen el mismo propósito. La ventaja de estas es que el IDE genera todos los drivers y configuraciones necesarias, así como contar con capas de aplicaciones que facilitan las pruebas e implementación de los sistemas embebidos.
- STM32CubeProgrammer: Esta es una herramienta que nos permite leer, escribir y verificar la memoria del dispositivo de manera sencilla a través de las interfaces de depuración y de arranque. Asimismo, cuenta con una gran cantidad de características que incluso permite generar imágenes de arranque para cualesquiera dispositivos.

En sí, esos son los ambientes de desarrollo de software dedicados a la programación y depuración de estos dispositivos. No obstante, existen paquetes de software que contienen diversos archivos con algunas aplicaciones específicas que se pueden descargar y son proporcionados por la misma empresa para disminuir los tiempos de desarrollo.

Si bien una de las ventajas de estos dispositivos es que facilitan la implementación y el tiempo de desarrollo para pruebas. Actualmente, muchos microprocesadores utilizan el término de LINUX embebido que es en términos sencillos la implementación del kernel de LINUX cuyo propósito es utilizar todos los archivos desarrollados en LINUX bajo este nuevo marco de desarrollo para los sistemas embebidos. La única desventaja es que, si se desea utilizar otro sistema operativo en estos dispositivos, no será posible al menos con los paquetes proporcionados por la empresa STMicroelectronics, por lo cual, resultará un poco más complicado de implementar un sistema operativo diferente al proporcionado.

4.3.9 Software

En el trabajo con microcontroladores, se necesitan herramientas de programación, compilación y depuración (*toolchain*) para programar y ejecutar la aplicación principal que se requiere desempeñar en los diferentes dispositivos seleccionados. Lo anterior se puede llevar a cabo de dos formas; la primera es utilizando las herramientas del fabricante que ya contienen todo el entorno de desarrollo necesario para programar, compilar y depurar nuestras aplicaciones; la segunda es definir por nuestra cuenta las herramientas que se requiere utilizar, por ejemplo, si usaremos interfaz gráfica o una línea de comandos, el

toolchain para el dispositivo objetivo, las características del depurador, entre otras herramientas necesarias para esto.

Dado que se han seleccionado microcontroladores de la familia ST es recomendable utilizar la interfaz de desarrollo proporcionada por el fabricante, la cual se denomina “STM32CubeIDE”, este ya incluye todas las herramientas para programar y depurar a estos microcontroladores. Además, el lenguaje de programación que se usará para todos los algoritmos y proyectos será C por ser un estándar en su aplicación con sistemas embebidos.

4.3.9.1 Software a bajo nivel de los microcontroladores de núcleo Cortex-M4

En este proyecto, con la finalidad de tener mayor control y entendimiento sobre el software del sistema embebido, se procedió a crear carpetas que contienen los códigos de programación del sistema embebido, a estas carpetas se les denomina “proyecto”. En el “proyecto” se incluyen carpetas organizadas que contienen módulos, estos módulos son la unidad de código que agrupa funciones, variables y estructuras relacionadas, los cuales, se forman por un archivo “.c” y un archivo “.h”, cuyo propósito es programar al microcontrolador y configurar a sus diferentes periféricos.

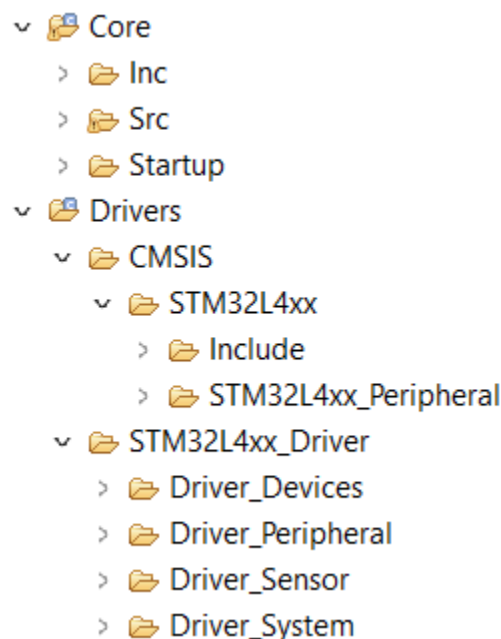


Figura 58. Organización general de las carpetas de los proyectos.

En la Figura 58 se muestra cómo está organizado el “proyecto” del microcontrolador para el subsistema de potencia y, esta misma forma de organización se utiliza para el “proyecto”

del subsistema de detección de objetos. Se debe mencionar que la organización de estos archivos suele variar entre empresas, pero por lo general es una buena práctica organizarlos como se muestra anteriormente.

A continuación, se explica brevemente que contiene cada una de estas carpetas:

- Core: Contiene el código principal de la aplicación, es decir, se incluye el archivo denominado normalmente como “main.c”.
- Drivers: Esta carpeta contiene todos los archivos y carpetas necesarias para controlar y configurar los periféricos y dispositivos conectados al microcontrolador, esta se subdivide en dos carpetas:
 1. CMSIS (Cortex Microcontroller Software Interface Standard): Esta carpeta es un estándar en muchas empresas y se usa para incluir el software base de los microcontroladores, en este se define principalmente apuntadores a los registros de los periféricos del microcontrolador.
 2. STM32L4xx_Driver: Esta carpeta contiene la parte de la aplicación que sirve para manejar a los periféricos y diversos dispositivos que se conectarán a esta unidad de control.
- Src: De forma general, este tipo de subcarpeta se visualiza en la mayoría de las carpetas principales y se usa para contener a los archivos fuentes (.c) los cuales contienen la definición de las funciones a utilizar en el programa principal, por ejemplo, una función para enviar datos o más simple, una función para hacer una simple operación como una suma.
- Inc: Los archivos en estas carpetas corresponden a los archivos de cabecera (.h), los cuales contienen la declaración de las funciones, así como otro tipo de definiciones para datos, por ejemplo, aquí se definen los apuntadores a direcciones de memoria del microcontrolador.

Una vez explicada la forma de organizar los proyectos realizados, se debe seguir una buena práctica de programación para generar los diferentes módulos que cumplirán con aplicaciones específicas. En este caso, al tratarse de un diseño de programación a bajo nivel, se debe atender a las especificaciones y manuales proporcionados por el fabricante, con la finalidad de poder configurar y utilizar cada periférico correctamente. Estos documentos son manuales de referencia y se pueden encontrar en su documentación simplemente buscando el modelo en [49]. Además, se debe tener cierto grado de conocimiento sobre cómo funcionan los diferentes protocolos de comunicación serial y cómo utilizar cada uno de estos.

Una vez encontrada la documentación, lo primero será definir las direcciones de los registros base para los diferentes periféricos o definir las direcciones bases de los buses que conectan a cada periférico, justo como se muestra en la Figura 59. En este ejemplo, primero se definen las direcciones bases de cada bus del microcontrolador y, posteriormente se definen las bases para cada registro del ADC. Lo anterior se debe realizar para cada periférico y, se recuerda que estas direcciones son únicamente las direcciones bases de los periféricos, por lo que para obtener acceso a otros registros específicos se les debe agregar un offset a las direcciones bases. Asimismo, como una recomendación extra, se sugiere el uso de estructuras en C para programar y hacer más eficiente la definición de punteros.

```
//PERIPHERAL BASE REGISTERS
#define PERIPH_BASE          0x40000000UL // Peripheral base address

#define APB1PERIPH_BASE      PERIPH_BASE
#define APB2PERIPH_BASE      (PERIPH_BASE + 0x00010000UL)
#define AHB1PERIPH_BASE      (PERIPH_BASE + 0x00020000UL)
#define AHB2PERIPH_BASE      (PERIPH_BASE + 0x08000000UL)
#define AHB3PERIPH_BASE      (PERIPH_BASE + 0x60000000UL)

#endif /* CMSIS_STM32L4XX_INCLUDE_STM32L4XX_BASEREG_H_ */

#define ADC1_BASE             (AHB2PERIPH_BASE + 0x08040000UL)
#define ADC2_BASE             (AHB2PERIPH_BASE + 0x08040100UL)
#define ADC3_BASE             (AHB2PERIPH_BASE + 0x08040200UL)
#define ADC123_COMMON_BASE    (AHB2PERIPH_BASE + 0x08040300UL)
```

Figura 59. Definición de registros al microcontrolador.

Con las direcciones de los registros definidos se generan apuntadores con acceso a memoria para los registros de los periféricos. Posteriormente, se deben generar los módulos para configurar y controlar los periféricos de acuerdo con el manual de referencia del fabricante, aquí sólo se recuerda que por conveniencia los archivos “.c” van a las carpetas Src y los “.h” en la carpeta Inc. Finalmente, una vez definidos cada módulo, podremos ver las declaraciones de todas nuestras funciones como se muestra en la Figura 60.

```
void ADCx_GPIOInit(GPIO_TypeDef *Port_, uint8_t Pin_);
void ADCx_Init(ADC_HandlerDef *ADC, ADC_JGroupCFG *jgroup, ADC_RGroupsCFG *rgroup, ADC_OVSCFG *ovs, uint32_t SMPR1, uint32_t SMPR2,
void ADCx_CMCFG(ADC_HandlerDef *ADC, uint8_t R_ConvMode, uint8_t J_ConvMode);
void ADC1_IRQHandler();
void ADC3_IRQHandler();
void ADCx_Calibration(ADC_HandlerDef *ADC);
void ADCx_Enable(ADC_HandlerDef *ADC);
void ADCx_Disable(ADC_HandlerDef *ADC);
void ADCx_EnEOCInt(ADC_HandlerDef *ADC);
void ADCx_EnEOSInt(ADC_HandlerDef *ADC);
void ADCx_DisEOCInt(ADC_HandlerDef *ADC);
void ADCx_DisEOSInt(ADC_HandlerDef *ADC);
void ADCx_SetADSTART(ADC_HandlerDef *ADC);
void ADCx_SetJADSTART(ADC_HandlerDef *ADC);
void ADCx_ResetDIFSEL(ADC_HandlerDef *ADC, uint32_t difsel);
void ADCx_CfgOVERSAMPLER(ADC_HandlerDef *ADC, ADC_OVSCFG *ovs);
void ADCx_SetEOSInt(ADC_HandlerDef *ADC);
void ADCx_SetJEOSInt(ADC_HandlerDef *ADC);
void ADCx_SetEOCInt(ADC_HandlerDef *ADC);
void ADCx_SetJEOSInt(ADC_HandlerDef *ADC);
void ADCx_WaitSetFlag(volatile uint32_t *SR, uint32_t BitReg);
void ADCx_WaitResetFlag(volatile uint32_t *SR, uint32_t BitReg);
void Delay(uint32_t time);
```

Figura 60. Declaración de funciones generadas para el periférico ADC.

Sobre este punto se debe recalcar que en este trabajo escrito no se presenta una lista organizada con todas las funciones realizadas por cada “proyecto” y para cada periférico y dispositivo, debido a que son demasiadas funciones, lo cual incrementaría significativamente el tamaño de hojas del escrito y, además, ya no coincide con el propósito de este trabajo.

Cabe destacar que la programación de todos los periféricos y dispositivos se hizo desde cero sin el uso de ninguna biblioteca, a excepción de la biblioteca USB, cuyo protocolo es complicado de implementar.

4.3.9.2 Implementación de los algoritmos del subsistema de detección de objetos

La implementación de los algoritmos para el subsistema de detección de objetos atiende a los requerimientos del diseño planteados en los capítulos 4.2.4 y 4.3.6 y se utilizan conceptos del capítulo 3.4.

De acuerdo con la Figura 46 se observa que principalmente se necesitan varios periféricos para la comunicación serial y otros para temporizadores.

Primero iremos con el análisis de los periféricos para las interfaces de comunicación serial, donde se requieren los siguientes:

- 1x UART para comunicación con el sensor TF02.
- 1x SPI por cada sensor de movimiento.
- 1x USB para transmisión serial de datos.

En el caso de los temporizadores, se requieren varios de estos para desempeñar diversas tareas:

- 1x Timer para generar una señal PWM de 50 Hz para el control de los servomotores.
- 1x Timer para sensores, este se usa para verificar que el dato ha llegado en el tiempo estimado. Debido a que LIDAR es el sensor principal, la frecuencia máxima de este sensor es de 1000Hz, por lo que este temporizador tiene una frecuencia de este valor.
- 1x Timer operando a 10Hz para cambiar de forma horizontal el mecanismo de movimiento, esto se debe a que si queremos un ángulo de visión de 60° y el fabricante indica que el motor tiene una velocidad de rotación de 0.1s/60°, lo anterior implica que por cada 0.1[s] podemos girar 60°
- 1x Timer operando a 10.5[Hz] para cambiar de forma vertical el mecanismo. Este valor se establece considerando que de forma vertical la visión del sensor se eleva o baja 3°, entonces, al colocar esta frecuencia aseguramos el mecanismo comience

con su movimiento de forma horizontal, justo después de que el motor haya completado 3° de movimiento de forma vertical.

- 1x Timer para sincronización de todos los demás temporizadores y para indicar envío de los datos recolectados durante 1s.

Es pertinente también mencionar las configuraciones de los sensores y sus razones:

- TF-02: Este es el sensor Lidar y se configura a 1000Hz para aprovechar su máxima capacidad y, en caso de que algún dato sea incorrecto o no sea fiable, el sensor lo indica y este se puede descartar. Aquí lo recomendado es utilizar una interfaz que sea capaz de transmitir toda la trama (8 bytes) del sensor en menos de 0.1s (frecuencia de muestreo), por ejemplo, con un baudrate de 115200bps, se garantiza que la trama se envía en aproximadamente 0.5ms.
- Sensores de movimiento inercial: Los sensores de movimiento pueden operar a frecuencias de muestreo de 8kHz, 4kHz y 1kHz, sin embargo, usar una mayor frecuencia de muestreo no siempre es mejor, para este caso, con la frecuencia de 1kHz los sensores implementan el uso de filtros digitales para disminuir el ruido y mejorar la precisión de los datos, asimismo, el sensor de LIDAR se limita a una frecuencia de operación de 1kHz, entonces no tiene sentido implementar una mayor cantidad de mediciones con ruido si el sensor principal no permite operar a una mayor frecuencia de muestreo. Además de lo anterior, los sensores permiten añadir un bias que es un valor diferente de cero que muestra el sensor cuando está en reposo, este valor se calcula obteniendo el promedio de varias mediciones mientras el dispositivo se encuentra en reposo.

Otras consideraciones sobre el diseño del algoritmo para que funcione correctamente se muestran a continuación:

- Los servomotores funcionan con una señal PWM de 50Hz, la cual, requiere un pulso con una duración que varía de 1ms a 2ms y este pulso se encarga de indicar si el movimiento es en sentido horario o antihorario. Lo anterior se puede calcular y configurar en los registros de los comparadores del timer como,

$$CCR = ValorCarga \left(\frac{50Hz}{f_{Pulso}} \right) \quad ec. (61)$$

- El sensor TF-02 envía sus datos de forma periódica, así que se debe verificar si el dato llega en tiempo, en caso contrario se descarta. Además, se envían 8 bytes, de acuerdo como se describe en [50], donde sólo 2 bytes corresponden a la parte alta y baja de la distancia y los demás se usan para la integridad de datos y del sensor.

- Los sensores de movimiento inercial entregan sus datos en un formato de complemento a 2, por tanto, se debe hacer la correcta representación de los datos, esto se puede realizar fácilmente aplicando un casteo de los datos obtenidos y, además de esto, se debe dividir este valor por el factor de escala del sensor indicado en el manual.
- La inclinación de los ángulos con el acelerómetro se calcula de acuerdo con las ecuaciones ec. 9, ec. 10 y ec. 11.
- El desplazamiento se calcula con los giroscopios capturando el tiempo del temporizador de 1[s] y multiplicando este valor por la velocidad angular del sensor, de igual forma, se puede asumir que este tiempo es aproximado a 1kHz como se mencionó. La ecuación utilizada es la ec. 14.
- Todos los datos de los sensores de medición inercial se encuentran en formato de tipo flotante. Para evitar hacer conversiones, se envían los datos en este formato y, este valor se puede obtener fácilmente por medio de un apuntador del tipo entero sin signo apuntando al formato flotante, esto permite obtener directamente la codificación del dato en su formato flotante.

Todos estos datos son almacenados en arreglos de manera ordenada como se muestra en la Tabla 17 y, cuando se finaliza un segundo de captura se comienzan a reescribir todos los datos, a su vez esto implica que, para almacenar todos los datos durante un segundo, se necesita definir un arreglo de un tamaño bastante grande, siendo este valor de 34000 datos para un tipo de datos de 8 bits sin signo.

Una vez definido todo lo anterior, se configura el sistema, los periféricos y las interfaces de comunicación serial, todo atendiendo al datasheet de cada dispositivo y usando los drivers que se realizaron, cuya descripción se muestra en el capítulo anterior. La configuración para cada periférico se muestra en la Tabla 41. Seguido de la configuración, se habilitan las interrupciones para las interfaces seriales y para los temporizadores, todo el flujo de la aplicación se muestra en el APÉNDICE VIII, donde esta aplicación en el programa principal se encarga de procesar los datos obtenidos por los sensores y de almacenarlos en arreglos de forma estructurada. Por otra parte, las interrupciones en las lecturas de los sensores se usan para evitar consumir tiempo esperando por poleo los datos, así como, generar las señales correctas en el tiempo establecido.

Por último, se muestra la carpeta mostrando algunos de los módulos generados para la aplicación del subsistema de detección de objetos.

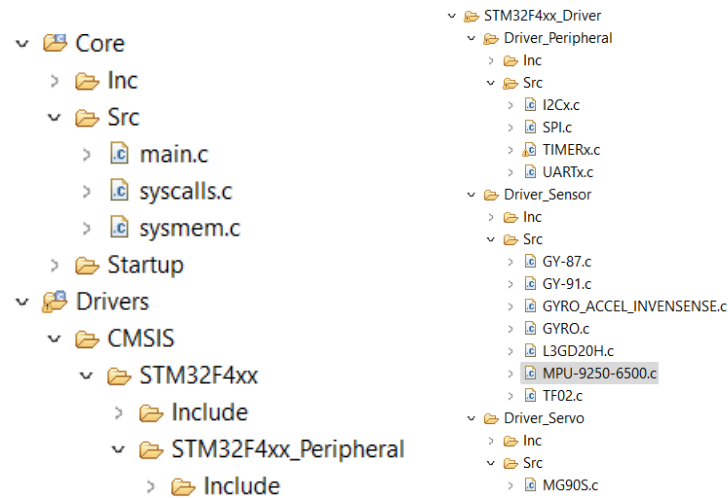


Figura 61. Carpetas del proyecto de detección de objetos.

4.3.9.3 Implementación de los algoritmos para la unidad de control del subsistema de potencia

La implementación de los algoritmos para la unidad de control del subsistema de potencia atiende al diseño referido en el capítulo 4.3.7

Al igual que en el capítulo anterior iremos con el análisis de los periféricos a configurar de acuerdo con la Figura 50:

1. Se necesitan 5 canales del ADC:
 - 1x Sensor de voltaje.
 - 4x Sensores de corriente para reguladores y medición de corriente en los motores.
2. 1x LPUART para comunicación con el dispositivo bluetooth.
3. 1x CAN para comunicación con la computadora de a bordo.
4. 3x Timers para generar señales PWM de sincronización que ayuden a los reguladores de conmutación.
5. 3x GPIOs configurados como salida para deshabilitación de los reguladores.
6. 1x RTC para capturar eventos sobre cambios acerca del estado del microcontrolador y para monitorear el estado del subsistema de potencia.

Algunas consideraciones en la configuración de los periféricos se describen a continuación:

- Se recomienda configurar el ADC en el modo de operación “Injected groups”, esto debido a que esta configuración permite realizar las mediciones de forma secuencial de acuerdo con cómo se defina el orden para cada canal, es decir, se definió que los 5 canales se muestreen de forma secuencial, iniciando primero por el sensor de voltaje, después un sensor de corriente y así, de esta manera. Además, se recomienda configurar un temporizador para que dispare la secuencia de

muestreo, con esto se puede definir una frecuencia de muestreo para los sensores, aquí se debe destacar que en cuanto mayor sea la frecuencia de muestreo se podrá tener una mejor estimación de la carga consumida por el sistema. Para casos prácticos se estableció una frecuencia de 1kHz.

- El LPUART únicamente puede operar a 9600 baudios debido a que en su modo de operación de bajo consumo esta es su máxima velocidad de transferencia y, por otra parte, el módulo bluetooth no puede ser reconfigurado una vez que se encuentra intercambiando datos de forma inalámbrica.
- La frecuencia de los temporizadores para la señal de sincronización atiende el diseño mostrado en los apéndices APÉNDICE I, APÉNDICE II y APÉNDICE III.
- El RTC se debe configurar con la hora y fecha correcta en que se comience a usar.

Otro punto importante es interpretar correctamente los valores del ADC como valores de voltaje y corriente y, además cómo calcular la carga consumida por el sistema.

Como primer punto se hablará de la interpretación del valor del ADC, para ello se hace referencia a la resolución que se calculó con anterioridad cuyo valor es de $800\mu V$. Entonces, para conocer el voltaje que mide el ADC, simplemente se realiza lo siguiente:

$$V_{ADC} = (ValorADC)(resolución) = (ValorADC)(800\mu V) \quad ec.(62)$$

En el caso del voltaje, este corresponde a un valor escalado de la batería de 22.2[V] a un valor de 3.3[V]. Entonces a 3.3[V] sabemos que la batería está cargada completamente al 100% y, de acuerdo con las recomendaciones del fabricante, el voltaje de corte para cada celda es de 3.5[V], esto para 6 celdas da un voltaje de corte de 21[V], al cual estaría descargada la batería y, este valor de corte de la batería en el divisor de voltaje corresponde a un valor que se puede calcular de la siguiente manera.

$$V_{outCutoff} = \frac{R_{out}}{R_{Tot}}(V_{cutoff}) \quad ec.(63)$$

$$V_{outCutoff} = \frac{3.3[M\Omega]}{22.2[M\Omega]}(21[V]) = 3.1216[V]$$

Con el paso anterior, podemos obtener la diferencia entre 3.3[V] y 3.1216[V], y este valor se considerará como la batería cargada al 100%. Después, por una simple regla de 3, se puede definir el nivel de porcentaje de carga de la batería, considerando ahora la diferencia entre el valor medido por el sensor y el voltaje de corte de 3.1216[V].

En el caso de los sensores de corriente, si se conocen los parámetros de diseño, fácilmente se puede calcular la corriente para cada sensor de la siguiente forma,

$$V_{shunt} = \frac{V_{ADC}}{Ganancia} \quad ec. (64)$$

$$I_L = I_{shunt} = \frac{V}{R_{shunt}} \quad ec. (65)$$

La anterior es una de varias formas de calcular la corriente en cada resistencia, otra manera es utilizando la resolución de los sensores de corriente.

Conociendo la corriente en cada sensor, se puede calcular cuánta carga se está consumiendo por fracción de segundo, esto se calcula como se muestra a continuación,

$$Q = I_L \times t_{muestreo} \quad ec. (66)$$

Con esto definido se puede determinar cuándo se ha consumido completamente la carga de la batería. Consecuentemente si se conoce que la batería tiene una capacidad de 16000mAh, esto a su vez es equivalente a decir que la batería puede proporcionar una carga de $16Ah \times \frac{3600[s]}{1[h]} = 57000[C]$. Con esto se puede comparar cuánta carga se ha consumido de la batería y, por ende, si aún se cuenta con energía para seguir operando.

Para usar el modo de operación de bajo consumo de energía en el microcontrolador, solo se debe considerar guardar en memoria la configuración de registros del sistema del microcontrolador, porque al entrar en este modo, se cambia la configuración del sistema y se recomienda deshabilitar varios periféricos. En ese sentido, el RTC puede monitorear de forma periódica si el sistema se encuentra en estado de operación o no y, en caso de no estar operando, puede entrar en un modo de bajo consumo para ahorrar energía por su propia cuenta.

Con todo lo anterior se configura el sistema, los periféricos y las interfaces de comunicación serial, todos atendiendo al datasheet de cada dispositivo y usando los drivers que se realizaron. La configuración para cada periférico se muestra en la Tabla 40.

Después de realizar las configuraciones, se habilitan las interrupciones de todos los periféricos a excepción de los temporizadores y, todo el flujo de la aplicación se muestra en el APÉNDICE VII, donde la aplicación en el programa principal se encarga de procesar las muestras del ADC de los sensores, el cálculo de la carga consumida y, procesar los datos enviados por bluetooth o por parte de la computadora de abordaje y con esto llevar a cabo acciones. Por otra parte, las interrupciones en las lecturas de los sensores se usan para evitar consumir tiempo esperando por poleo los datos y, lo mismo con las interrupciones en los protocolos de comunicación que se usan para recibir mensajes por parte de otros dispositivos.

Finalmente, se implementó el código de la aplicación y se generaron diversos módulos para todos los elementos mencionados en esta parte, estos se muestran en la Figura 62.

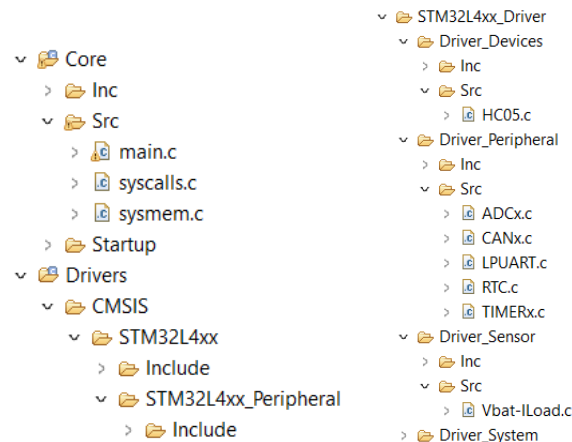


Figura 62. Carpetas del proyecto de la unidad de control del subsistema de potencia.

4.3.9.4 Implementación de los algoritmos de control

A diferencia de los algoritmos implementados en los microcontroladores de núcleo Cortex M4, los algoritmos de control se pueden probar en cualquier entorno de desarrollo con un compilador y depurador que implementen el lenguaje de programación C. Lo anterior, debido a que únicamente se necesita generar un ejecutable que utilice los módulos de estos algoritmos (estos no acceden a periféricos del sistema). Por otra parte, el ejecutable debe ser generado para el dispositivo objetivo mostrado en el capítulo 4.3.8.

Entonces, siguiendo buenas prácticas de organización del proyecto, se genera una carpeta que contiene los módulos para cada uno de los algoritmos requeridos, los cuales se muestran en la Figura 63.

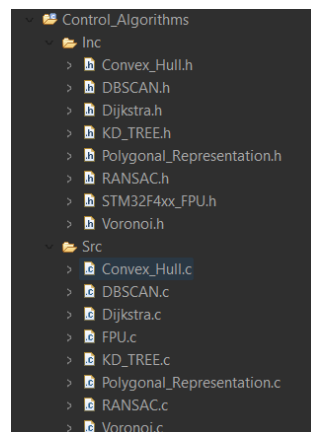


Figura 63. Organización de los algoritmos de control.

Al igual que los casos anteriores, se recomienda hacer uso de estructuras para definir parámetros relacionados a los algoritmos y con ello poder manejarlos bajo un mismo nombre. En este caso, como se habla de datos del ambiente, estos se pueden representar a través de coordenadas espaciales, sin embargo, para facilitar el desarrollo de esta tesis, se hablará de las coordenadas en el plano cartesiano de dos dimensiones, por ejemplo, en la Figura 64 se define un tipo de estructura que nos permite manejar las coordenadas espaciales de cualquier punto.

```
typedef struct{  
    float    x, y, z;  
}PointCoordinates_Def;
```

Figura 64. Tipo de dato definido para manejar puntos de coordenadas cartesianas.

Así como en el ejemplo anterior se recomienda analizar qué tipo de estructura se requiere definir antes de comenzar a generar las funciones, porque esto permite disminuir código y trabajar de forma más eficiente con datos que están relacionados bajo un mismo alias. Además, cabe destacar que los algoritmos se fueron desarrollando en cierto orden de jerarquía, principalmente, porque las salidas de algunos algoritmos son las entradas de otros algoritmos. Por tanto, se comienza con los algoritmos de agrupamiento por clústeres para identificar qué puntos pertenecen a un objeto, después se sigue con la representación poligonal donde se obtienen los vértices de los objetos, para posteriormente obtener el centroide de cada objeto, los cuales se utilizan para generar los puntos de Voronoi, siendo estos último requeridos por el algoritmo de Dijkstra en la determinación de la ruta con menor costo. Se debe mencionar que para algunos de estos algoritmos se hizo uso de la biblioteca “math.h” para cálculos relacionados con funciones trigonométricas, entre otras funciones.

Es relevante destacar que cada uno de los algoritmos instrumentados fueron adecuados a partir de los originales y se instrumentó cada uno de acuerdo a como se muestra en los diagramas de los apéndices APÉNDICE IX, APÉNDICE X, APÉNDICE XI y APÉNDICE XII.

4.3.9.4.1 Algoritmo DBSCAN

El algoritmo de DBSCAN funciona a través de definir ciertos parámetros en el algoritmo, siendo los siguientes:

- Un radio mínimo cuyo valor indica si un punto se interconecta con otro y, por ende, si forma parte de este.
- Un número mínimo de puntos (típicamente denominado “e”) que indica si un punto dentro de las muestras es un núcleo o un posible borde.

Con estos parámetros definidos, el algoritmo analiza cada punto comparando su distancia respecto al resto de puntos. Aquí se debe detallar que no es necesario calcular todas las distancias de cada punto analizado con el resto de los puntos porque algunas de estas distancias se repiten, es decir, se calcula la distancia de P1 a P2, pero se omite volver a calcular la distancia P2 a P1, debido a que es lo mismo. Para saber si un punto se interconecta con otro si tienen dos formas posibles: la primera es por la distancia euclidiana entre dos puntos y la segunda comparando geométricamente si el punto se encuentra contenido en el radio del punto analizado. Esta última forma es computacionalmente menos costosa para los procesadores porque evita calcular la raíz cuadrada, lo anterior se verifica de la siguiente manera:

$$(x - h)^2 + (y - k)^2 \leq r^2 \quad \text{ec. (67)}$$

Para cada punto analizado, se almacena su información que posteriormente se analiza para determinar qué puntos son núcleos o posibles bordes. Estos se almacenan en tipos de estructuras como se muestra en la Figura 65.

```
struct DBSCAN_PointDef{
    struct DBSCAN_PointDef *ptrPrev;
    struct DBSCAN_PointDef *ptrNext;
    PointCoordinates_Def **matchedPoints;
    PointCoordinates_Def *point;
    uint32_t noMatchedP;
    bool Core;
};
```

Figura 65. Estructura utilizada para el algoritmo DBSCAN.

A continuación, se dará una breve explicación para que el lector este mayor familiarizado de la razón de este tipo de estructura:

- ptrPrev y ptrNext: Estos son apuntadores que permiten crear listas enlazadas, cada elemento dentro de la lista contiene información sobre los puntos analizados. Definir estas estructuras como lista es útil porque cada vez que se termine de usar el algoritmo DBSCAN se puede liberar esta memoria.
- matchedPoints: Esta variable es un apuntador a apuntadores, esto permite almacenar la dirección de los puntos que están interconectados a cada punto base, con lo cual se evita crear nuevas copias para los datos disminuyendo el almacenamiento necesario.
- point: Es el punto base sobre el cual se está analizando el algoritmo.

- noMatchedP: Indica cuántos puntos se interconectan con el punto base, lo cual determina si es un borde o un núcleo.
- Core: Indica si es un núcleo o un posible borde.

Con todos los datos analizados y definidos como núcleos y posibles bordes, se compara en el algoritmo cuáles de estos núcleos comparten bordes (se comparan apuntadores a memoria), en caso de que compartan bordes, estos pertenecen a un mismo objeto, caso contrario, pertenecen a otro objeto. Todo este proceso se repite hasta definir los diferentes objetos conformados por núcleos. Asimismo, si ninguno de los puntos definidos como posible borde se encuentra contenido dentro de algún núcleo este se considera simplemente como ruido.

El diagrama de flujo que muestra brevemente cómo se implementó este algoritmo atendiendo a los puntos anteriores se muestra en el APÉNDICE IX.

4.3.9.4.2 Algoritmo Convex Hull

Con el algoritmo del capítulo anterior se obtienen todos los puntos que conforman a un solo objeto, estos puntos son los que se utilizan para los algoritmos de Convex Hull y el de RANSAC (explicado en el siguiente capítulo).

El primer paso de este algoritmo es definir un punto base, este punto corresponde al de menor valor en X y Y. Una vez hecho lo anterior, se calcula el ángulo entre el eje X y las rectas formadas por el punto base con el resto de los puntos. Este cálculo se puede simplificar (debido a que una recta es el eje X) como se muestra a continuación.

$$m = \frac{y_2 - y_1}{x_2 - x_1} \quad ec. (68)$$

$$\theta = \arctan(m) \quad ec. (69)$$

Donde m es la pendiente de la recta formada por el punto base y otro punto del objeto. Aquí se hacen algunas consideraciones, primero cuando $x_2 = x_1$ la pendiente no está definida, pero se sabe que tiene un valor de 90° , y en otro caso, cuando el ángulo de diferentes rectas es el mismo se debe escoger la recta de mayor longitud.

Considerando lo anterior, se ordenan los puntos restantes de acuerdo con el ángulo formado entre las rectas descritas, empezando con los ángulos de menor valor hacia los de mayor valor.

Con los puntos ordenados, se generan vectores para obtener el producto cruz entre dos vectores. Estos vectores se definen de la siguiente forma al inicio del algoritmo:

- El punto base y el segundo punto ordenado forman un vector.

- El segundo punto ordenado y el tercer punto forman otro vector.

El vector de los puntos que va de A hacia B se calcula de la siguiente forma,

$$\overrightarrow{AB} = B - A \quad ec. (70)$$

El producto cruz entre los vectores para cada caso indica si el ángulo da un giro a la izquierda o la derecha y, esto conlleva varias consideraciones:

- Si el giro es a la izquierda, este punto se considera parte de los vértices del polígono y se almacena, por tanto, ahora el segundo vector pasa a considerarse el primer vector, mientras que el segundo vector se conforma por el último punto del primer vector con el siguiente punto ordenado sin analizar, y así sucesivamente.
- En caso de que sea un giro a la derecha, el punto no pertenece a los vértices y se descarta. Ahora los vectores se conforman de la siguiente manera, el primer vector sigue considerándose como el primer vector y, el segundo vector se conforma por el segundo punto del primer vector con el siguiente punto ordenado sin analizar.

El producto cruz que se ocupó se calcula de la siguiente manera siendo los vectores $\overrightarrow{V_1} = (x_1, y_1)$ y $\overrightarrow{V_2} = (x_2, y_2)$,

$$\overrightarrow{V_1} \times \overrightarrow{V_2} = x_1 y_2 - x_2 y_1 \quad ec. (71)$$

Con estos cálculos y comparaciones es como se define cuáles son los puntos que pertenecen a la envolvente o los vértices del polígono. Sólo se debe considerar que pueden existir puntos que no producen un cambio en la dirección del vector, por lo cual, se debe analizar si el siguiente punto se debe considerar como vértice o no.

El diagrama de flujo correspondiente a este algoritmo y que muestra brevemente cómo se implementó este atendiendo a los puntos anteriores se encuentra en el APÉNDICE X.

4.3.9.4.3 Algoritmo RANSAC

Los cálculos y el algoritmo utilizados para RANSAC se definen directamente en el capítulo 3.7.2. En este capítulo lo importante es ver cómo se implementa el modelo que se ajusta a los datos en el algoritmo, el cual, por facilidad y para pruebas se consideró para el modelo de un cuadrilátero. De acuerdo, con el algoritmo, los modelos deben generarse con puntos aleatorios, dicho esto, sólo se debe considerar lo siguiente:

- Un cuadrilátero se conforma por 4 puntos,
- Un cuadrilátero no puede tener más de dos puntos con el mismo valor en X y Y,
- En caso de que los puntos no satisfagan la composición del cuadrilátero, se deben descartar y volver a obtener otros puntos para generar el modelo.

Una vez definidos los puntos del modelo se ordenan sus puntos como se explicó en el capítulo 3.7.1. Esto sirve para definir las funciones implícitas (descrito en ese mismo capítulo) de las rectas formadas por los pares de puntos del cuadrilátero y, con estas funciones se calcula la distancia entre un punto (x_1, y_1) en los dos datos y las rectas del modelo. Este cálculo se realiza como se describe debajo,

$$d = \frac{|Ax_1 + By_1 + C|}{\sqrt{A^2 + B^2}} \quad ec. (72)$$

Siendo la función implícita,

$$Ax + By + C = 0 \quad ec. (73)$$

Con lo anterior se verifica cuáles puntos se consideran como parte del modelo y cuáles no y, con esto se obtiene la cantidad de puntos contenidos en el modelo, esto se repite con cada iteración comparando y buscando un modelo que se adapte mejor a los puntos de muestra. Asimismo, se debe mencionar que es importante acotar las rectas que definen al cuadrilátero, porque estas pueden tomar cualquier rango de valores fuera del modelo. Una forma de hacer esto es generar vectores ortogonales en los vértices y después utilizar el análisis del capítulo 3.7.1 para observar si este punto se encuentra contenido entre dos rectas perpendiculares formadas por los vértices.

4.3.9.4.4 Algoritmo de Voronoi

Después de obtener los objetos representados por sus vértices, se obtiene el centroide para cada objeto y este se representa como un único punto. En este caso, por facilidad se considera que se cuenta con los centroides de los obstáculos.

El algoritmo implementado fue el de Fortune debido a su eficiencia. Para este algoritmo un parámetro importante a considerar es el decremento de una recta (la cual se denominará “recta de barrido”) paralela al eje X que se encarga de escanear los puntos (centroide de los objetos) sobre el eje Y.

Dicho lo anterior, se recomienda ordenar los puntos de mayor a menor con respecto a su coordenada en el eje Y, mientras que los valores en la coordenada del eje X no se ordenan. Una vez ordenados, se sitúa a la recta que hará el barrido en el mayor valor encontrado entre los puntos que representan a los objetos, e inmediatamente después de situarse en este valor, se procede a hacer el decremento. Como se mencionó, cada vez que la “recta de barrido” tiene un valor menor respecto a la coordenada en Y de los objetos se genera una nueva parábola y esta recta pasa a convertirse en la directriz para cada caso. Los

parámetros requeridos para las parábolas se pueden guardar en una estructura como la que se describe en la Figura 66.

```
typedef struct{
    float          A;           //A*x^2
    float          B;           //B*x
    float          C;           //C
    PointCoordinates_Def centroide;
    PointCoordinates_Def vertex; //V(h,k)
    float          p;           //(x-h)^2=4*p|(y-k)|
}Parabola_ParametersDef;
```

Figura 66. Estructura que describe los parámetros de una parábola.

Estos parámetros cambian de valor cada vez que la “recta de barrido” se decrementa (esta recta se define con una variable llamada “P” que almacena su valor), lo que a su vez provoca que la parábola se abra más. La forma en que se calculan los parámetros se muestra a continuación:

$$p = \frac{y_{centroide} - P_{valor}}{2} \quad ec. (74)$$

$$x_{vértice} = x_{centroide} \quad ec. (75)$$

$$y_{vértice} = \frac{y_{centroide} + P_{valor}}{2} \quad ec. (76)$$

$$A = \frac{1}{4p} \quad ec. (77)$$

$$B = -\frac{2x_{vértice}}{4p} \quad ec. (78)$$

$$C = \frac{x_{vértice}^2}{4p} + y_{vértice} \quad ec. (79)$$

Donde los únicos valores que no cambian son las coordenadas del vértice de la parábola. Con estos parámetros se define cada parábola y, cuando se encuentra más de una parábola se procede a calcular los puntos donde se intersecan las parábolas. En general, esto se puede resolver usando el método de igualación, es decir, se sabe que los parámetros anteriores representan la ecuación ec. 50.

Entonces, si se iguala la ecuación de ambas parábolas $y_1 = y_2$ se obtiene una nueva ecuación, la cual al resolver por medio de la ecuación general da como resultado los valores de x para los cuales se intersecan las parábolas y por consiguiente los puntos donde se intersecan. Lo anterior es de forma general, pero se puede evitar usar la ecuación general si alguno de los parámetros A , B ó C son cero.

Al inicio del código, cuando se calcula la primera o primeras intersecciones, por lo general estos puntos se definen como los primeros puntos de Voronoi. Estos puntos a su vez conforman un nodo justo como que se definen en la Figura 67. Estos nodos contienen las coordenadas del punto base, el número de nodos a los que se puede viajar y, los nodos hacia donde se puede mover desde cada punto base.

```
typedef struct VoronoiNode_Def{
    uint32_t                noConections;
    PointCoordinates_Def    pointVoronoi;
    struct VoronoiNode_Def  **ptrPointsConection;
}Voronoi_Node;
```

Figura 67. Definición de puntos de Voronoi

Sin embargo, no todas las intersecciones entre las diferentes posibilidades de parábolas corresponden a puntos de Voronoi. Algunos de estas intersecciones dejan de tomar relevancia y se deben descartar, lo mismo sucede con las parábolas, estas dejan de proveer intersecciones útiles para los puntos de Voronoi y por tanto se deben descartar.

Lo anterior se puede resolver tomando como referencia los arcos que se forman con las intersecciones y, además, estos deben ser los arcos que sobresalen. Una forma de manejarlos es como una lista enlazada como se observa en la Figura 68 y, esta lista debe estar ordenada por rangos, por ejemplo, primero el arco $Arc_1 \in [-\infty, x_1)$, después sigue el arco que va del rango $Arc_2 \in [x_1, x_2)$, y así sucesivamente.

```
typedef struct ARCNode{
    struct ARCNode          *ptrNext;
    struct ARCNode          *ptrPrev;
    Parabola_ParametersDef   *parabola;
    Range_Def               range;
};
```

Figura 68.. Definición de arcos utilizados para generar intersecciones

Definir estas estructuras facilita la manera en que se generarán las intersecciones entre parábolas. Se debe mencionar que para obtener un punto de intersección como mínimo

se requieren dos parábolas, entonces, el algoritmo genera las intersecciones entre parábolas haciendo uso de la lista enlazada de arcos, es decir, en el primer caso el algoritmo calcula las intersecciones para el primer elemento y el segundo elemento de la lista de arcos, posteriormente, entre el segundo elemento y el tercer elemento y así sucesivamente hasta recorrer toda la lista. Ahora bien, una parábola puede intersectar con otra en dos puntos, ¿cómo saber cuál es el correcto? Y la respuesta es bastante sencilla, para eso se definen los límites de cada arco y se ordenan, esto nos permite saber si uno de los puntos que se obtuvo en la intersección sale de este rango y, por tanto, se puede descartar.

Otro aspecto por tratar aquí es cómo se van agregando las parábolas, cada vez que se encuentra un nuevo objeto (punto) al hacer el barrido se compara con los arcos formados cuál es el arco que genera el menor costo con el objeto respecto a X . El arco que corresponda es con el que se calculan las intersecciones, así mismo, esta lista se debe actualizar para agregar este nuevo arco.

Además, aún faltan mencionar cómo se obtienen los demás puntos de Voronoi. Estos se van añadiendo y conectando conforme ocurre un evento denominado “evento circular”, este evento ocurre cuando se tiene más de una intersección en el mismo punto, es decir, 3 parábolas coinciden en un único punto, esta coincidencia no es exactamente igual, sino que debe coincidir dentro de un rango de valores, esto debido al decremento definido para la “recta de barrido”. En este evento ocurre la mayoría de las actualizaciones de puntos de Voronoi, siendo estos los siguientes:

- 3 parábolas coincidentes en un punto indican que una parábola ya no es necesaria para generar los puntos de Voronoi, por tanto, la lista de arcos se actualiza.
- Se añade un nuevo punto de Voronoi que se interconecta con otro punto de Voronoi establecido anteriormente por la parábola recién descartada.

Finalmente, cuando la recta de barrido llega hasta un cierto valor deseado, se hace una última captura de las intersecciones y los puntos resultantes se agregan e interconectan como los últimos nodos de Voronoi.

El diagrama de flujo correspondiente a este algoritmo y que muestra brevemente cómo se implementó este atendiendo a los puntos anteriores se encuentra en el APÉNDICE XI.

4.3.9.4.5 Algoritmo de Dijkstra

En este algoritmo se procesan los nodos de Voronoi obtenidos en el capítulo anterior. Como mínimo se debe ingresar al algoritmo dos nodos, el nodo inicial y el nodo destino.

Entonces, como ya se tienen los nodos interconectados, el algoritmo comienza desde el nodo de inicio, para ello se considera que no se conocen las distancias hacia ningún otro nodo marcándose como infinito. Posteriormente, el nodo inicial comienza analizando los

nodos hacia los que puede moverse y, una vez conociendo las posiciones de ambos puntos se puede calcular la distancia euclidiana como sigue,

$$d = \sqrt{(x - x_1)^2 + (y - y_1)^2} \quad ec. (80)$$

Conociendo la distancia a los nodos (los que no alcanza el nodo se siguen considerando infinito) se escoge el de menor costo, al avanzar a este nodo se debe considerar que ya no se podrá regresar al nodo desde el cual se viajó. Dicho lo anterior, en cada iteración se analizan los nuevos nodos a donde se puede avanzar y en cada etapa se actualizan los costos escogiendo el menor valor hacia cada nodo, asimismo se debe indicar desde que nodo se viajó. Esto se realiza iterativamente hasta llegar al nodo destino.

Cuando se han expandido todas las rutas posibles hacia el destino, únicamente se compara la que tenga el menor costo y, se recuperan los nodos desde los cuáles se viajó con la finalidad de encontrar la ruta de menor costo que se conecta con el inicio.

El diagrama de flujo correspondiente a este algoritmo y que muestra brevemente cómo se implementó este atendiendo a los puntos anteriores se encuentra en el APÉNDICE XII.

4.3.10 Diseño de placas para pruebas (PCB)

En el diseño de los prototipos de las placas de PCB para pruebas se atendió principalmente a las consideraciones proporcionadas por diferentes empresas que se encargan del diseño de estas mismas. Algunas consideraciones generales se muestran en [51] y [52].

Se debe mencionar que existen herramientas como “SATURN PCB Design” que permite definir parámetros requeridos para el diseño de la PCB, como lo son el espaciado mínimo entre conductores, así como el ancho de pista necesario para soportar la corriente requerida.

Finalmente, se puede utilizar diferente tipo de software para el diseño de PCB, donde para este proyecto se destaca el uso de Autodesk Fusion 360 y Altium Designer.

4.3.10.1 Diseño de PCB para la computadora de a bordo

En el diseño de la PCB para la computadora de a bordo se usó el software de Autodesk Fusion 360 debido a que ofrece las herramientas mínimas para diseñar este tipo de placas. Como se explicó en el capítulo 4.3.8, el dispositivo MYC-YA151C seleccionado para desempeñar el trabajo de la computadora de a bordo ya cuenta con los siguientes elementos:

- La unidad de procesamiento,
- Memoria DRAM,
- Memoria Flash,

- Una unidad de manejo de potencia para la regulación de voltaje dentro de este sistema.

Por tanto, lo único que se necesita implementar son las interfaces físicas para conectar diversos dispositivos a la computadora de a bordo, estas son necesarias porque no resulta muy cómodo y conveniente soldar directamente los pines de la computadora con los cables de los dispositivos externos. Además, también se necesita una interfaz para seleccionar la interfaz de arranque, la cual se implementa con los pines de selección.

Como primer punto se solicitó el manual denominado “Product Manual” en [53] para la asignación de pines. En este manual el fabricante indica cuáles son las funciones de los pines de este dispositivo. En esta parte se debe hacer mención que, si únicamente se planea implementar la imagen de Linux proporcionada por el fabricante, entonces no se recomienda hacer ningún cambio sobre esta asignación. Dicho esto, para futuras pruebas, se decide hacer una reasignación de las funciones descritas en este manual para cada pin. Esta reasignación se hizo principalmente para puertos cuya configuración es la de GPIO y, se realiza con ayuda del datasheet del microprocesador encontrado en [54]. Los únicos pines reasignados se muestran en la Tabla 24

Tabla 24. Reasignación de definición de pines.

No. Pin	Función Alterna
83	PWM1
92	ETH_TD1 ó SPI2_MISO
115	UART7_Tx
116	UART7_Rx
118	SPI2_NSS
119	SPI2_CLK
120	SPI2_MOSI
130	PWM3

Con las funciones definidas para cada pin y, atendiendo a los conectores de cada uno de los dispositivos externos (estos dependen del fabricante y son conectores específicos) es que se comienza con el diagrama de conexiones que se muestra en el APÉNDICE IV. Asimismo, para realizar este diagrama de conexiones se siguieron las recomendaciones

en el diseño de las interfaces proporcionadas por el fabricante (contenidas en [55]), estas recomendaciones se usaron principalmente para la interfaz de las memorias micro SD, la interfaz de USB y, la interfaz de selección del bootloader (en esta no se menciona, pero de forma predeterminada, los pines de selección tienen una resistencia pull-down interna).

Una vez realizado el diagrama de conexiones ateniendo lo anterior, se sigue con el enrutamiento de señales, aquí lo más importante es definir las reglas de enrutamiento con las que debe cumplir el diseño de PCB, estas reglas son importantes porque el mismo software indica si se están cumpliendo o no. Algunas de estas se muestran a continuación:

- Para un conductor externo en una PCB operando a un voltaje de 0-15[V] se recomienda un espaciado entre conductores de 0.1[mm].
- Para una corriente de 1[A] se recomienda un ancho de pista correspondiente a .25[mm], donde de acuerdo con el fabricante del microprocesador, este puede consumir hasta 1[A]. La referencia del ancho de pista se encuentra en [56].
- Las señales para USB son señales diferenciales, así que se deben enrutar como tal, manteniendo la longitud de cables lo más parecido posible.

Para otras interfaces como la de Ethernet, el controlador de LCD y los GPIO, únicamente se enrutaron las señales directamente hacia headers, lo anterior debido a que por el momento no se utilizan para este proyecto.

Por último, con todo lo considerado anteriormente es como se obtiene el modelo en 3D de la placa diseñada, mostrada en el APÉNDICE IV

4.3.10.2 Diseño de PCB para el subsistema de potencia

En el diseño de la PCB para el subsistema de potencia se usó el software Altium Designer, principalmente porque este software ofrece una herramienta para el análisis de la integridad de potencia en la PCB, lo cual indica si el diseño de nuestra PCB soportará la potencia de operación establecida.

Primero se explicará la razón del diagrama de conexiones. Este diagrama de conexiones atiende al diseño del subsistema de potencia mostrado en la Figura 46. Aquí tal vez la pregunta es ¿físicamente cómo se definieron las conexiones? Se debe recordar que en el capítulo 4.3.7 se seleccionan cada uno de los reguladores de conmutación y sus componentes se calculan y simulan justo como se indica en los apéndices APÉNDICE I, APÉNDICE II y APÉNDICE III.

La conexión de los reguladores de voltaje en el diagrama de conexiones es independiente y, lo único en lo que coinciden estos es que todos son alimentados por la fuente de energía principal que es la batería de 22.2[V].

Una vez definido el diagrama de conexiones con los reguladores se analiza de nuevo qué más se necesita conectar de acuerdo con la Figura 46. Entonces, para cada regulador a la salida se requiere medir la corriente que proporciona a cada carga y, esto se describe los capítulos 3.6 y 4.3.7, por consiguiente, para realizar lo anterior se necesita lo siguiente:

- Una resistencia shunt por cada regulador,
- Un filtro RC,
- Un amplificador de detección de corriente (comercial).

La conexión de estos tres elementos es muy simple: la resistencia shunt se conecta a la salida de cada regulador; el filtro RC encontrado en el estado del arte se compone de dos resistores y un capacitor y se conecta como se muestra en la Figura 23; y el amplificador de detección de corriente se conecta de acuerdo como lo indica el fabricante para cada uno de los dispositivos enlistados en la Tabla 21. En resumen, las conexiones físicas son las que se muestran directamente en la Figura 23. Adicionalmente se añade otra resistencia shunt entre la batería y los conectores de los ESC para determinar el consumo de corriente por los motores y el ESC y, a esta resistencia se añade lo descrito anteriormente.

Asimismo, para la salida del regulador de 3.3[V] se agrega un interruptor controlado por voltaje que únicamente quita la alimentación a todos los dispositivos que no son la unidad de control del subsistema de potencia.

Con lo anterior definido, se conectan en el diagrama de conexiones las señales de control a headers, los cuales, serán usados para conectar la placa de potencia con la placa de desarrollo de la unidad de control.

Al igual que el caso anterior, con el diagrama de conexiones definido se comienza con el enrutamiento de señales (siguiendo las recomendaciones de enrutamiento) y se definen las reglas de enrutamiento con las que debe cumplir nuestro diseño de PCB. En esta parte se atiende a las recomendaciones del fabricante, como mantener los más cerca posible los componentes de cada regulador y, además, lo más importante es atender al análisis de la integridad de la potencia en la PCB. Este análisis de la integridad de potencia de la PCB se lleva a cabo en el APÉNDICE VI.

5. Pruebas y resultados

En este capítulo se abordará de manera detallada los resultados alcanzados con cada uno de los subsistemas del dron, donde se mencionarán sus características por separado.

5.1 Resultados obtenidos del subsistema de detección

El fundamento del diseño del subsistema de detección se documenta en el capítulo 4.3.6, cuyas características finales se enlistan en la Tabla 25.

Tabla 25. Características del subsistema de detección

Parámetro	Valor
Tasa de actualización de datos	993Hz
FOV (desempeño en 1s)	30° vertical 60° Horizontal
Rango de operación de LIDAR	0.1[m]-40[m] @90%reflectividad
Resolución	1 cm
Precisión	±1% (>5[m])

En el desarrollo del subsistema de detección de objetos se lograron los siguientes resultados con la tarjeta de desarrollo NUCLEO-F446RE operando a 80[MHz]:

1. Se generaron drivers para controlar por medio de servomotores el mecanismo de movimiento del sensor LIDAR TF02-Pro. En este caso se utilizaron 4 temporizadores sincronizados para generar las secuencias de movimientos definidos.
2. Se realizaron drivers para configurar y obtener mediciones por SPI, de los sensores conformados por giroscopios y acelerómetros con la finalidad de poder determinar la orientación del mecanismo de LIDAR.
3. Se realizaron drivers para obtener mediciones y configurar el sensor LIDAR TF02-Pro por medio de UART.
4. Finalmente, se conjuntaron los drivers en una sola aplicación, en la cual, mientras se controlaba el movimiento de LIDAR por temporizadores, también se obtiene cada 1[ms] los datos de sensores para ser almacenados dentro de un arreglo de datos con un formato de tipo flotante.

Para más información de la configuración de los periféricos de este subsistema, se recomienda revisar el APÉNDICE XIII. Algunas pruebas del sistema en operación se

muestran y describen a continuación. Asimismo, en la Figura 69 se observa un diagrama que muestra la manera en que se realizaron las pruebas y mediciones.

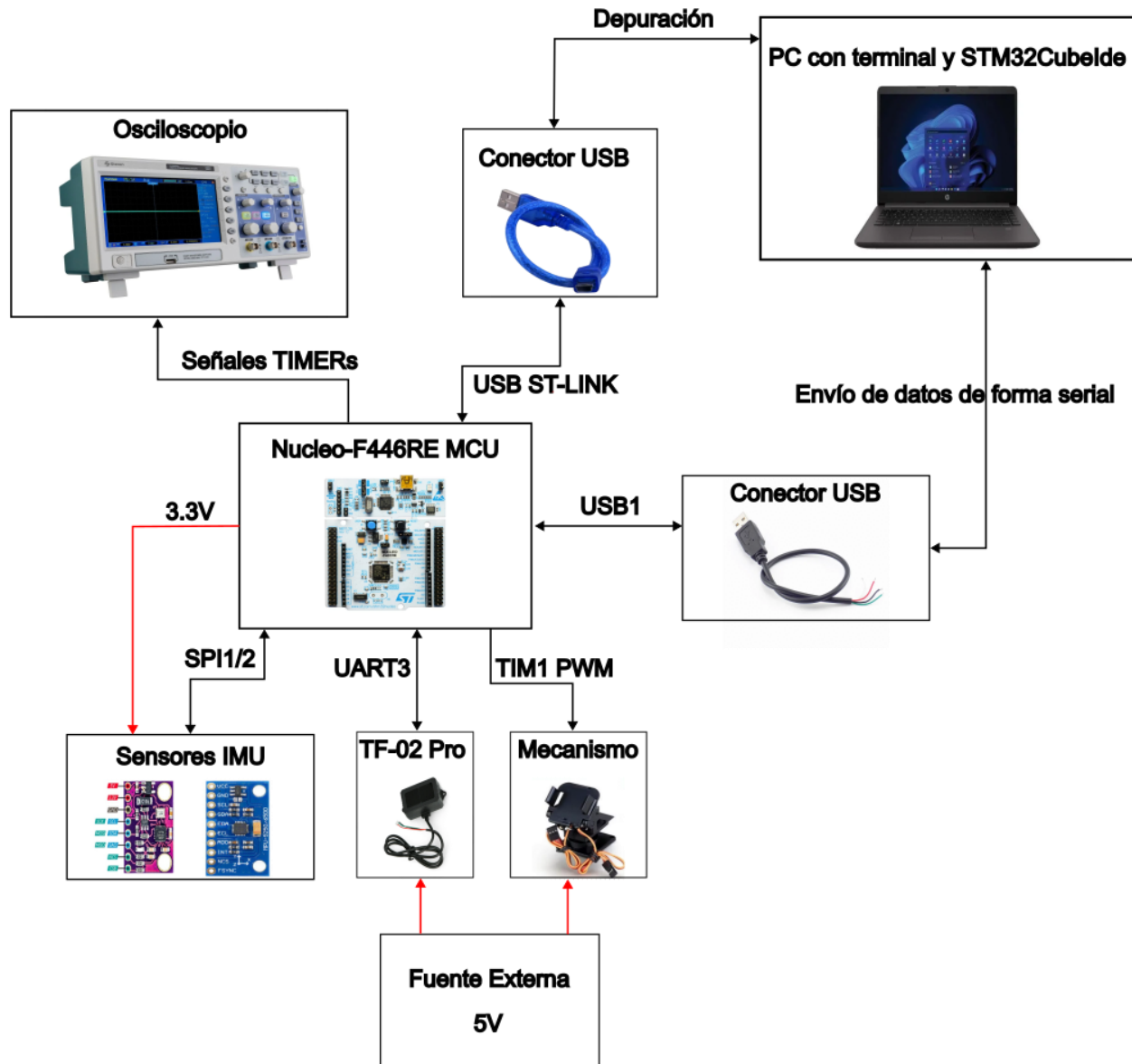


Figura 69. Diagrama del banco de pruebas para el subsistema de detección de objetos.

Una forma de probar en conjunto el subsistema de detección de objetos, fue desempeñar su operación de manera estática, alineando los sensores de LIDAR junto con los sensores IMU y realizar la lectura de los sensores, verificando obstáculos a diversas distancias en la frecuencia de muestreo solicitada.

En el manejo de los servomotores para observar la secuencia del sensor LIDAR, se hizo uso del osciloscopio, donde las señales observadas corresponden a: tiempo de duración de muestreo de 1[s]; tiempo para el cambio de dirección del mecanismo 0.1[s]; movimiento del servomotor horizontal; movimiento del servomotor vertical.

En la Figura 70 se observan las señales de control generada por PWM para el control de los servomotores, de aquí, se obtiene con el osciloscopio que la frecuencia de operación es aproximadamente 50Hz.

Por otra parte, en la Figura 71 se observan dos pulsos de diferente duración, de acuerdo con el osciloscopio, estos intervalos son de 1.16666[ms] y 1.83333[ms], cuyo valor corresponde a una posición de 30° y -30° respecto al eje de referencia del servomotor, respectivamente, lo cual, al intercalar estos valores, hace que el mecanismo haga un barrido de izquierda a derecha. Asimismo, en la Figura 72 se obtuvieron dos pulsos, una con duración máxima y otra con duración mínima, los cuales, se ajustaron para generar un valor de -15° a 15° de movimiento, respecto al eje de movimiento del mecanismo en sí, estos valores son de 1.5833[ms] y 1.9166[ms], considerando que el mecanismo en su posición vertical se encuentra centrado para una señal PWM con un ancho de pulso de 1.75[ms].



Figura 70. Señales de control de los servomotores: a) Amarillo-Horizontal; b) Azul-Vertical

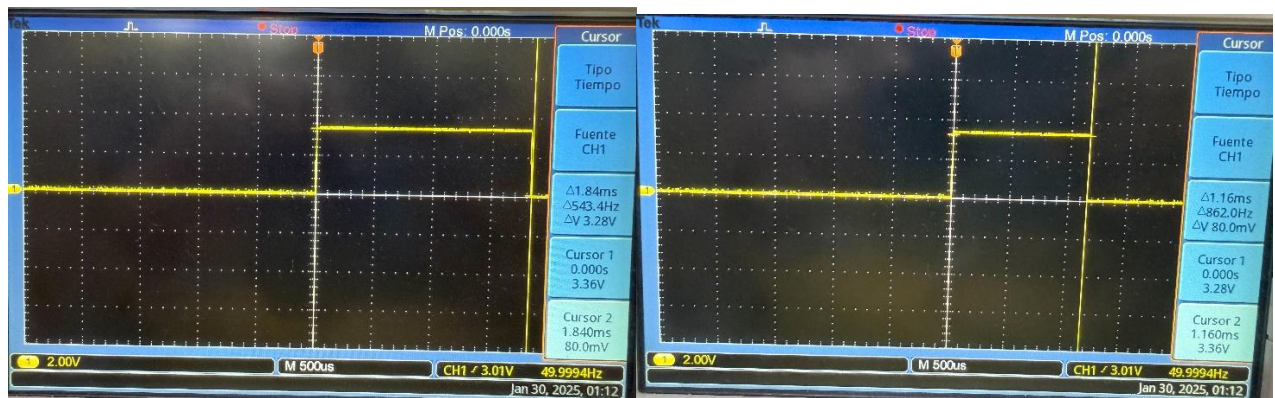


Figura 71. Medición del ancho de pulso: (izq.) Giro a la derecha; (dcha.) Giro a la izquierda.

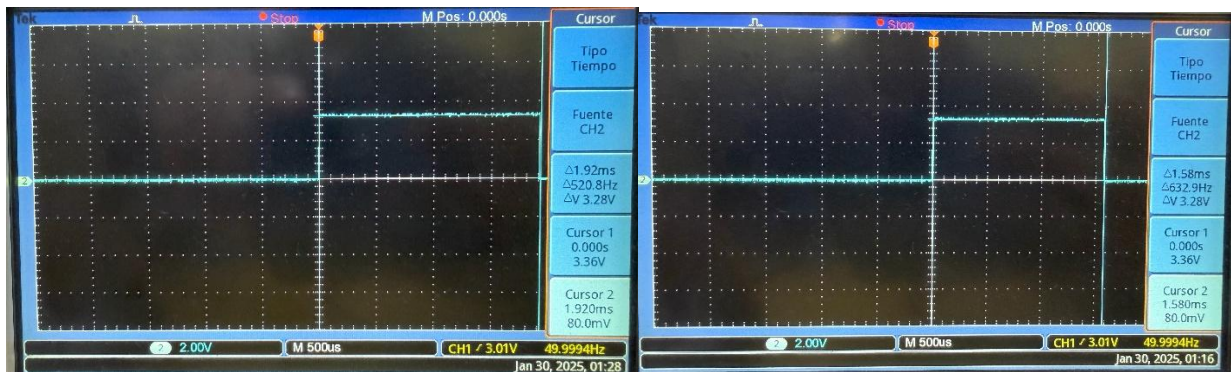


Figura 72. Medición del ancho de pulso: (izq.) Mov. hacia arriba; (dcha.) Mov. hacia abajo.

Las señales de temporización utilizadas para cambiar el sentido de movimiento del mecanismo en conjunto se muestran en el temporizador de 1[s] y el de 0.1[s], donde, el temporizador de menor frecuencia funciona como un maestro que sincroniza las señales de los temporizadores.

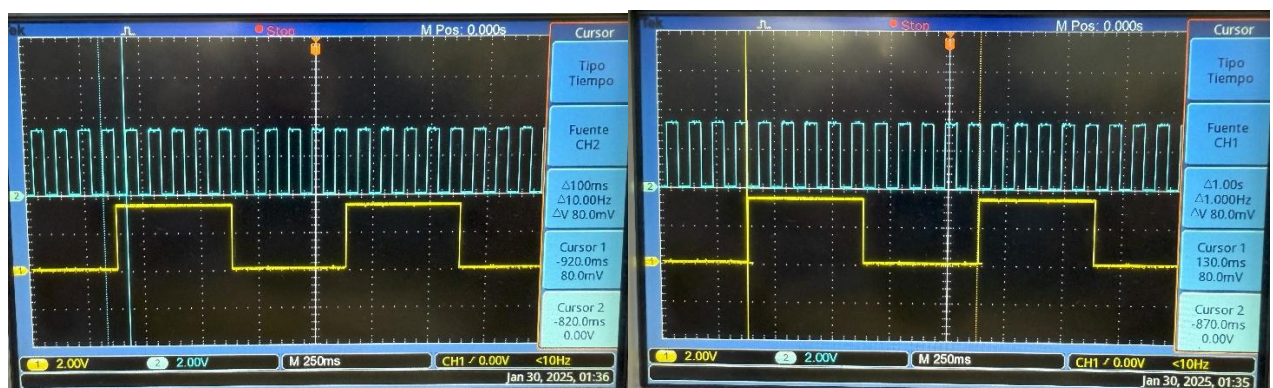
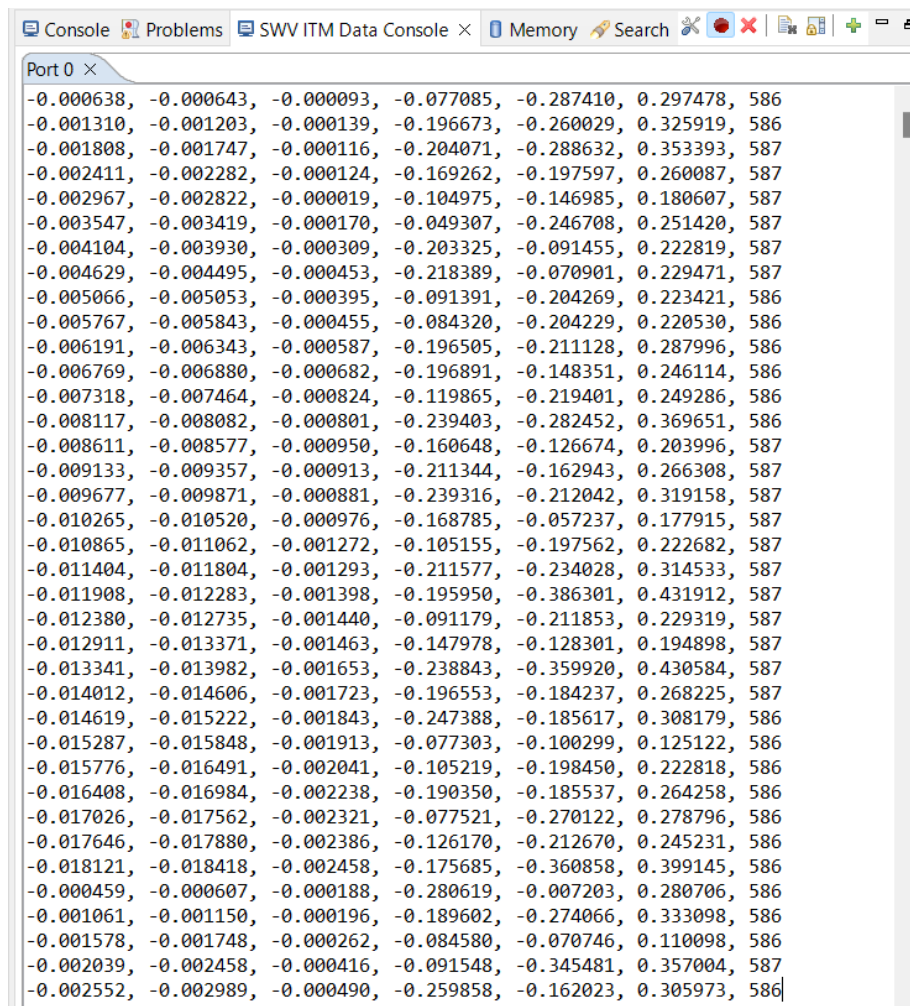


Figura 73. Medición de los temporizadores que se encargan de cambiar la posición de los servomotores.

Como prueba de desempeño, se realizó la medición con un flexómetro desde una pared hasta el sistema de detección de objetos, cuyo valor se midió de 5.86[m]. En esta parte, además de medir de manera continua la distancia, se obtuvieron valores de posicionamiento y de movimiento, con los acelerómetros y giroscopios, respectivamente, obteniendo un total de 2900 muestras. Los resultados capturados se muestran en la Figura 74, donde, los primeros 3 datos corresponden al desplazamiento, los siguientes 3 datos a la inclinación respecto a un eje de referencia, todo en cuestión de X, Y, Z y, el último valor corresponde a la distancia medida en centímetros.

Asimismo, estas 2900 muestras se utilizaron para observar la raíz del error cuadrático medio (RMSE), donde, si el dispositivo se encuentra operando de manera estática, el valor de desplazamiento debería corresponder a 0 en cualquier dirección y, el ángulo de inclinación respecto al eje de referencia debe ser 0, y, el valor de la distancia medida debe ser de 5.86[m]. Los valores obtenidos para este caso se muestran en Tabla 26.



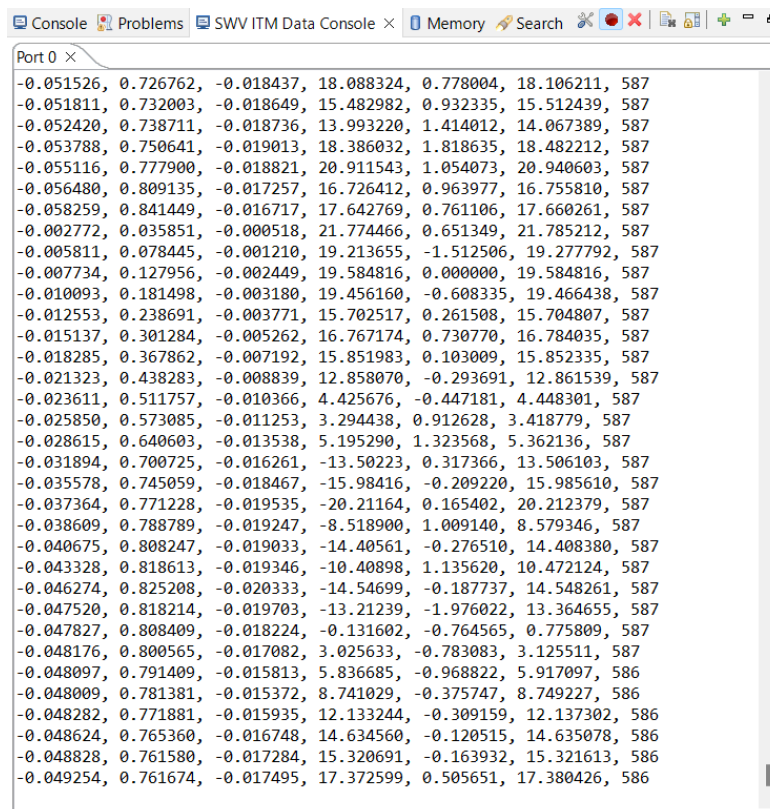
Displacement X	Displacement Y	Displacement Z	Inclination X	Inclination Y	Inclination Z	Distance (cm)
-0.000638	-0.000643	-0.000093	-0.077085	-0.287410	0.297478	586
-0.001310	-0.001203	-0.000139	-0.196673	-0.260029	0.325919	586
-0.001808	-0.001747	-0.000116	-0.204071	-0.288632	0.353393	587
-0.002411	-0.002282	-0.000124	-0.169262	-0.197597	0.260087	587
-0.002967	-0.002822	-0.000019	-0.104975	-0.146985	0.180607	587
-0.003547	-0.003419	-0.000170	-0.049307	-0.246708	0.251420	587
-0.004104	-0.003930	-0.000309	-0.203325	-0.091455	0.222819	587
-0.004629	-0.004495	-0.000453	-0.218389	-0.070901	0.229471	587
-0.005066	-0.005053	-0.000395	-0.091391	-0.204269	0.223421	586
-0.005767	-0.005843	-0.000455	-0.084320	-0.204229	0.220530	586
-0.006191	-0.006343	-0.000587	-0.196505	-0.211128	0.287996	586
-0.006769	-0.006880	-0.000682	-0.196891	-0.148351	0.246114	586
-0.007318	-0.007464	-0.000824	-0.119865	-0.219401	0.249286	586
-0.008117	-0.008082	-0.000801	-0.239403	-0.282452	0.369651	586
-0.008611	-0.008577	-0.000950	-0.160648	-0.126674	0.203996	587
-0.009133	-0.009357	-0.000913	-0.211344	-0.162943	0.266308	587
-0.009677	-0.009871	-0.000881	-0.239316	-0.212042	0.319158	587
-0.010265	-0.010520	-0.000976	-0.168785	-0.057237	0.177915	587
-0.010865	-0.011062	-0.001272	-0.105155	-0.197562	0.222682	587
-0.011404	-0.011804	-0.001293	-0.211577	-0.234028	0.314533	587
-0.011908	-0.012283	-0.001398	-0.195950	-0.386301	0.431912	587
-0.012380	-0.012735	-0.001440	-0.091179	-0.211853	0.229319	587
-0.012911	-0.013371	-0.001463	-0.147978	-0.128301	0.194898	587
-0.013341	-0.013982	-0.001653	-0.238843	-0.359920	0.430584	587
-0.014012	-0.014606	-0.001723	-0.196553	-0.184237	0.268225	587
-0.014619	-0.015222	-0.001843	-0.247388	-0.185617	0.308179	586
-0.015287	-0.015848	-0.001913	-0.077303	-0.100299	0.125122	586
-0.015776	-0.016491	-0.002041	-0.105219	-0.198450	0.222818	586
-0.016408	-0.016984	-0.002238	-0.190350	-0.185537	0.264258	586
-0.017026	-0.017562	-0.002321	-0.077521	-0.270122	0.278796	586
-0.017646	-0.017880	-0.002386	-0.126170	-0.212670	0.245231	586
-0.018121	-0.018418	-0.002458	-0.175685	-0.360858	0.399145	586
-0.000459	-0.000607	-0.000188	-0.280619	-0.007203	0.280706	586
-0.001061	-0.001150	-0.000196	-0.189602	-0.274066	0.333098	586
-0.001578	-0.001748	-0.000262	-0.084580	-0.070746	0.110098	586
-0.002039	-0.002458	-0.000416	-0.091548	-0.345481	0.357004	587
-0.002552	-0.002989	-0.000490	-0.259858	-0.162023	0.305973	586

Figura 74. Datos obtenidos del subsistema de detección de objetos, operando en condiciones estáticas.

Tabla 26. RMSE obtenido en un régimen estático.

Descripción	RMSE	Unidades
Movimiento en X	0.01090766	[°]
Movimiento en Y	0.01138425	[°]
Movimiento en Z	0.001577413	[°]
Inclinación en X	0.23777108	[°]
Inclinación en Y	0.22028717	[°]
Inclinación en Z	0.32332235	[°]
Distancia	0.33062115	[cm]

Como siguiente prueba, se puso configuró al subsistema de detección para que el mecanismo escaneara de forma vertical de arriba hacia abajo y de manera inversa. Los datos obtenidos se muestran en Figura 75 y Figura 76.



Port 0
-0.051526, 0.726762, -0.018437, 18.088324, 0.778004, 18.106211, 587
-0.051811, 0.732003, -0.018649, 15.482982, 0.932335, 15.512439, 587
-0.052420, 0.738711, -0.018736, 13.993220, 1.414012, 14.067389, 587
-0.053788, 0.750641, -0.019013, 18.386032, 1.818635, 18.482212, 587
-0.055116, 0.777900, -0.018821, 20.911543, 1.054073, 20.940603, 587
-0.056480, 0.809135, -0.017257, 16.726412, 0.963977, 16.755810, 587
-0.058259, 0.841449, -0.016717, 17.642769, 0.761106, 17.660261, 587
-0.060772, 0.835851, -0.000518, 21.774466, 0.651349, 21.785212, 587
-0.060811, 0.078445, -0.001210, 19.213655, -1.512506, 19.277792, 587
-0.067734, 0.127956, -0.002449, 19.584816, 0.000000, 19.584816, 587
-0.010093, 0.181498, -0.003180, 19.456160, -0.608335, 19.466438, 587
-0.012553, 0.238691, -0.003771, 15.702517, 0.261508, 15.704807, 587
-0.015137, 0.301284, -0.005262, 16.767174, 0.730770, 16.784035, 587
-0.018285, 0.367862, -0.007192, 15.851983, 0.103009, 15.852335, 587
-0.021323, 0.438283, -0.008839, 12.858070, -0.293691, 12.861539, 587
-0.023611, 0.511757, -0.010366, 4.425676, -0.447181, 4.448301, 587
-0.025850, 0.573085, -0.011253, 3.294438, 0.912628, 3.418779, 587
-0.028615, 0.640603, -0.013538, 5.195290, 1.323568, 5.362136, 587
-0.031894, 0.700725, -0.016261, -13.50223, 0.317366, 13.506103, 587
-0.035578, 0.745059, -0.018467, -15.98416, -0.209220, 15.985610, 587
-0.037364, 0.771228, -0.019535, -20.21164, 0.165402, 20.212379, 587
-0.038609, 0.788789, -0.019247, -8.518900, 1.009140, 8.579346, 587
-0.040675, 0.808247, -0.019033, -14.40561, -0.276510, 14.408380, 587
-0.043328, 0.818613, -0.019346, -10.40898, 1.135620, 10.472124, 587
-0.046274, 0.825208, -0.020333, -14.54699, -0.187737, 14.548261, 587
-0.047520, 0.818214, -0.019703, -13.21239, -1.976022, 13.364655, 587
-0.047827, 0.808409, -0.018224, -0.131602, -0.764565, 0.775809, 587
-0.048176, 0.800565, -0.017082, 3.025633, -0.783083, 3.125511, 587
-0.048097, 0.791409, -0.015813, 5.836685, -0.968822, 5.917097, 586
-0.048009, 0.781381, -0.015372, 8.741029, -0.375747, 8.749227, 586
-0.048282, 0.771881, -0.015935, 12.133244, -0.309159, 12.137302, 586
-0.048624, 0.765360, -0.016748, 14.634560, -0.120515, 14.635078, 586
-0.048828, 0.761580, -0.017284, 15.320691, -0.163932, 15.321613, 586
-0.049254, 0.761674, -0.017495, 17.372599, 0.505651, 17.380426, 586

Figura 75. Datos obtenidos por LIDAR moviendo el mecanismo de forma vertical

En la primera y tercera columna de Figura 75 y Figura 76, se observan que los valores son menores a 0.1, esto se debe a que, al realizar el movimiento de manera vertical, los giroscopios calculan la velocidad de movimiento sobre el eje Y, que es donde se notan valores cercanos a 1[°]. Posteriormente, los ángulos de inclinación (cuarta y sexta columna) que cambian debido al movimiento vertical son sobre el eje de referencia X y Z, en donde se debe observar un aumento y decremento de los ángulos debido al movimiento. Y, de la misma manera, se logra observar como la distancia crece y decrece abruptamente, esto se debe principalmente a que cuando el mecanismo orienta al sensor hacia abajo, este se topa con la mesa, dando como resultado una medida de menor valor.

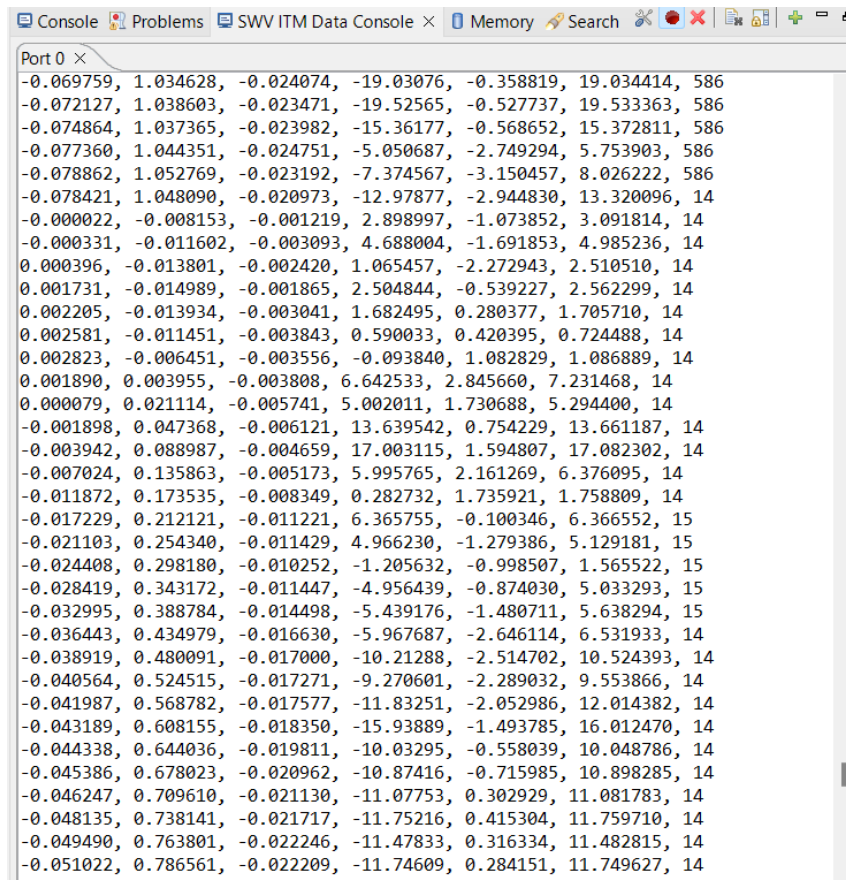
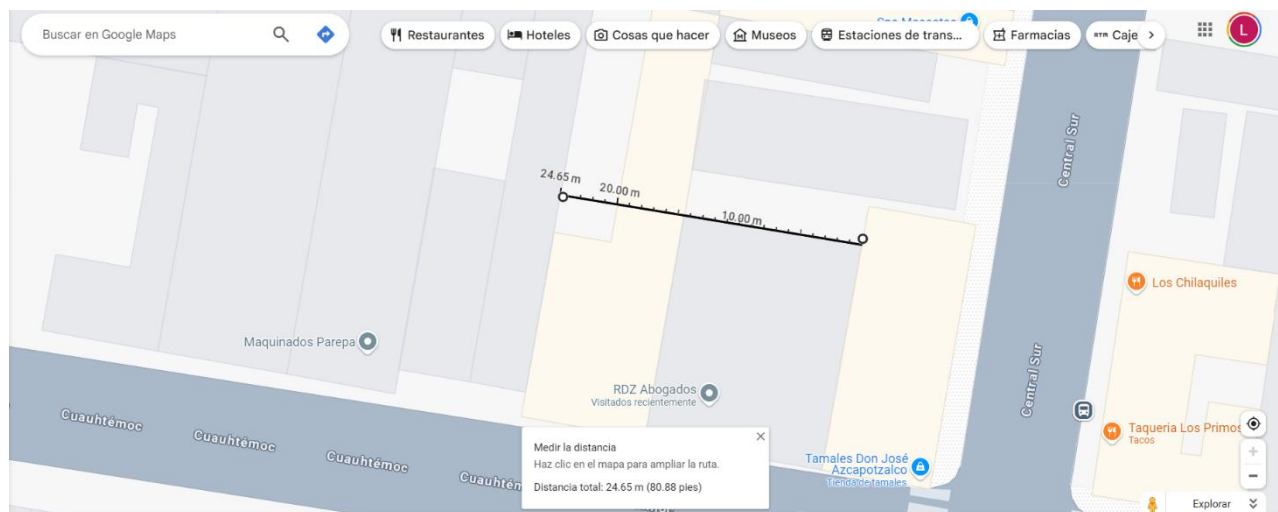
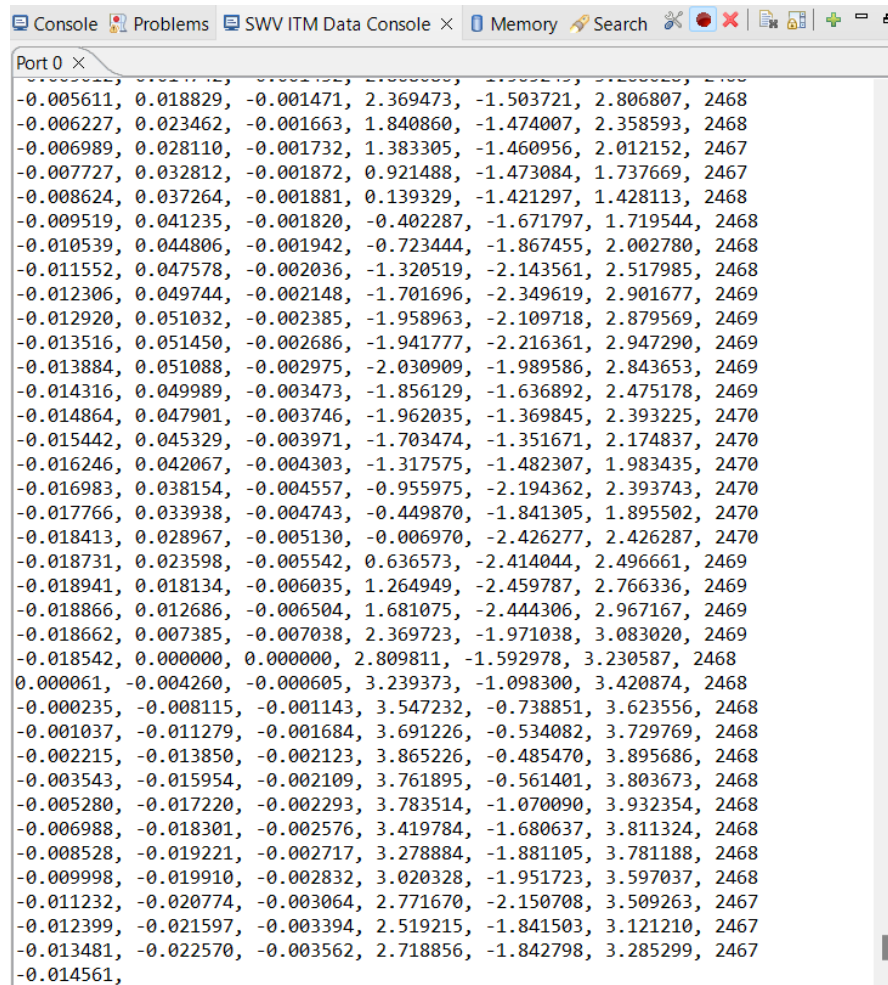


Figura 76. Datos obtenidos por LIDAR moviendo el mecanismo de forma vertical

Otro experimento realizado fue probar el alcance del sensor LIDAR haciendo comparación con la herramienta de medición de Google Maps, para esto se orientó hacia un edificio el sensor y se obtuvieron diferentes muestras de la distancia. En Google se midió una aproximación de la distancia que se midió respecto a dos puntos, la posición del sensor y un edificio (Figura 78). Los resultados obtenidos por el sensor se muestran en Figura 77, dando un valor alrededor de los 24.67[m] y 24.7[m], donde de acuerdo con Google Maps, el valor aproximado es de 24.65[m].



Un experimento distinto que se realizó fue enviar las tramas de datos obtenidos por medio de USB, desde el microcontrolador hacia una computadora por medio de una terminal serial. En el experimento se corrobora que el dispositivo por medio del cual se enviaran los datos a través de USB es reconocido por el administrador de dispositivos (Figura 79) y, los datos enviados, a través de USB se reciben correctamente en la terminal (Figura 80).

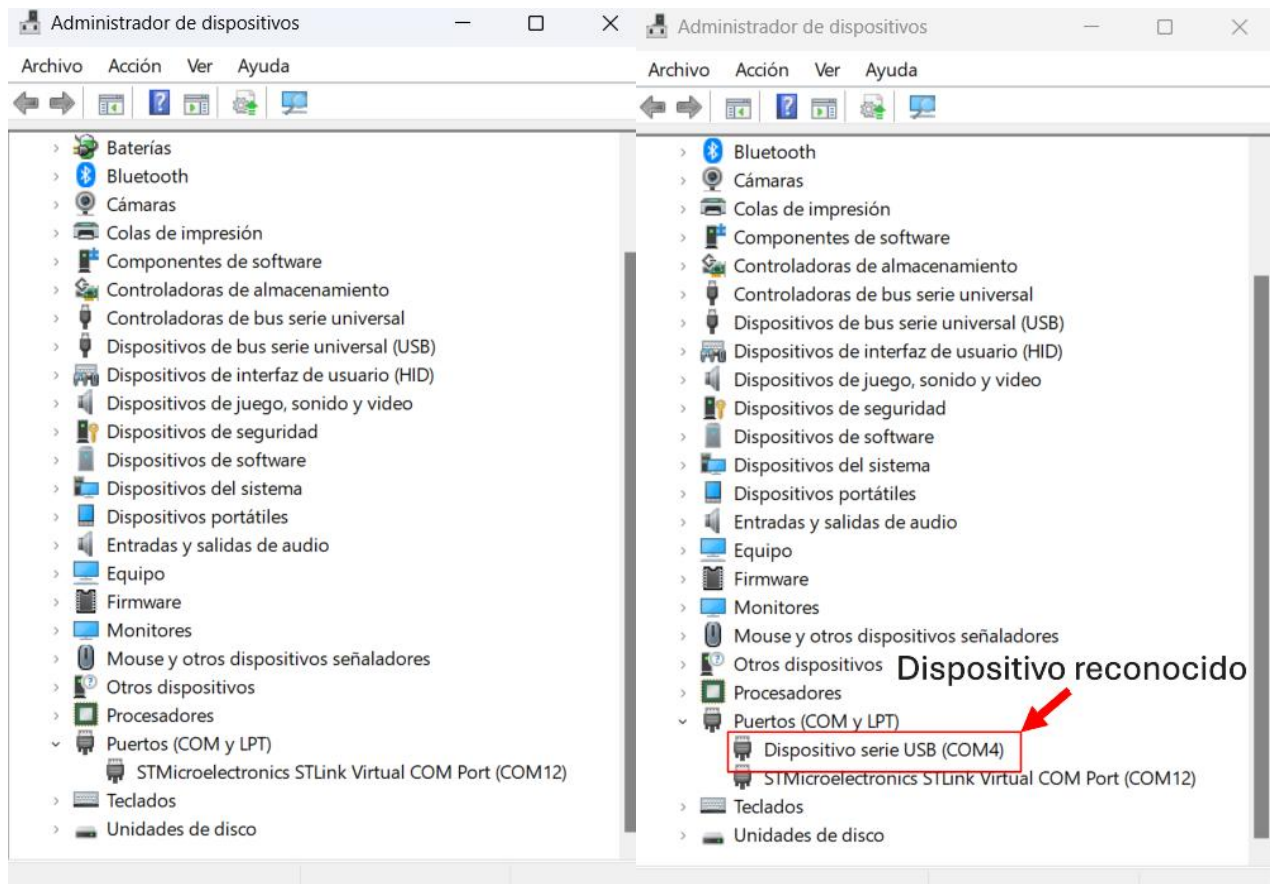


Figura 79. Administrador de dispositivos antes y después de ejecutar el programa con USB.

De la Figura 80 se debe destacar que el puerto al que está conectado la terminal corresponde con el puerto COM4 que nos indica el administrador de dispositivos, además, en la parte inferior de la terminal se muestran la cantidad de caracteres o datos que se recibieron del microcontrolador. Aquí, se debe destacar que la ventaja de usar USB sobre otros protocolos seriales es debido a su velocidad de transferencia que es 12Mb/s para la velocidad “FS” de antiguas versiones y, alcanzando velocidades de transmisión más altas a esta (hasta 480Mb/s para el modo “HS”).

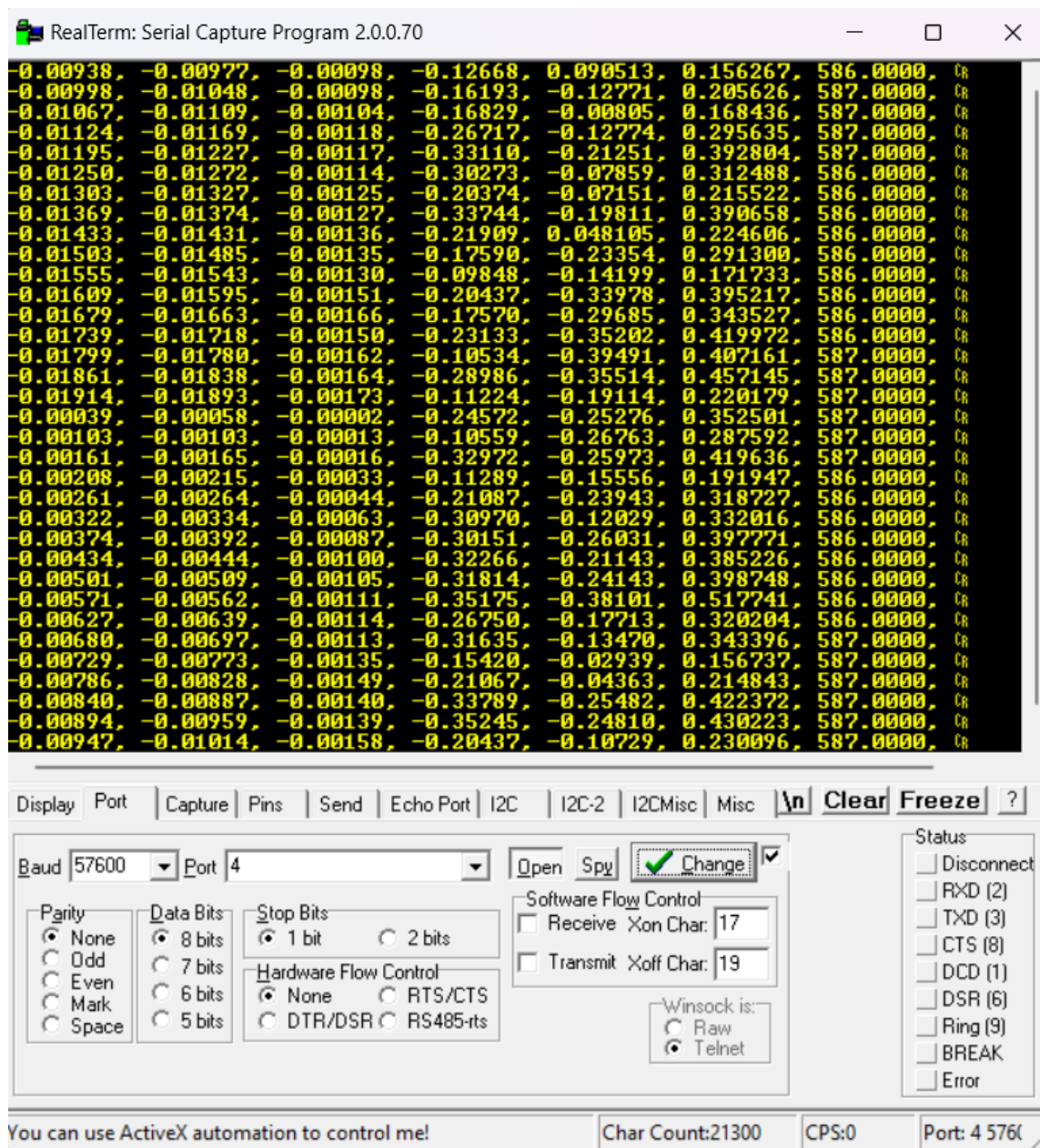


Figura 80. Datos recibidos con la terminal, enviado a través del puerto COM4.

Finalmente, se muestran imágenes del edificio que se enfocó para medir la distancia y el sensor de distancia junto con los sensores de movimiento colocados en el mecanismo de movimiento.



Figura 81. (izq.) Sensores colocados en el mecanismo de movimiento; (dcha.) Edificio donde se midió una distancia de 24.7[m].

5.2 Resultados obtenidos de la unidad de control (MCU) del subsistema de Potencia

El subsistema de potencia está fundamentado en el capítulo 4.3.7. La unidad de control del subsistema de potencia se implementó con la tarjeta de desarrollo NUCLEO-L452RE-P debido a que se contaba con la disponibilidad de ésta.

En este capítulo se mostrará principalmente los resultados de los algoritmos implementados en la unidad de control para manejar al subsistema de potencia, no obstante, como también se trabajó el diseño a nivel sistema del subsistema de potencia, se describirá de forma general los alcances obtenidos, los cuales se muestran a continuación:

1. Se generaron drivers para permitir la conectividad de la unidad de control del subsistema de potencia por medio de Bluetooth y, a través de LPUART²⁵.
2. Se realizaron drivers para permitir comunicar a la computadora de abordaje con la unidad de control del subsistema de potencia por medio del protocolo bus CAN 2.0.
3. Se implementaron los drivers que se encargan de tomar mediciones en los sensores de corriente a través de diferentes canales del ADC con una resolución de hasta 16 bits. La lectura de los sensores se complementa con funciones para interpretar el valor del voltaje de las resistencias shunt en corriente, así como el voltaje de la batería en porcentaje de energía restante.

²⁵ LPUART, por sus siglas en inglés, hace referencia al módulo de comunicación UART utilizado para bajo consumo de energía.

4. Se diseñaron drivers para entrar en el modo de bajo consumo con el microcontrolador de la placa de desarrollo NUCLEO-L452RE-P, con la finalidad de disminuir el consumo de energía cuando el sistema no se encuentra en operación, es decir, cuando no se está realizando ninguna entrega.
5. Con los drivers anteriores se generó una aplicación donde el microcontrolador de la placa de desarrollo NUCLEO-L452RE-P simula la medición y análisis del consumo energético total del sistema cuando se encuentra en operación y, además, se simula la gestión del consumo de los dispositivos conectados al subsistema de potencia.
6. Asimismo, se generó una propuesta final del diseño de una PCB para implementar todos los dispositivos requeridos en el subsistema (APÉNDICE V). Para corroborar el diseño de la PCB, primero se hicieron los análisis mencionados con anterioridad y se simuló los reguladores de voltaje utilizados, los cuales, se muestran en APÉNDICE I, APÉNDICE II y APÉNDICE III, así como también se simuló el comportamiento de la placa en el APÉNDICE VI.

Para más información de la configuración de los periféricos de este subsistema, se recomienda revisar el APÉNDICE XIII. Algunas pruebas del sistema en operación se muestran y describen a continuación. Adicionalmente, en la Figura 82 se observa un diagrama que muestra la manera en que se realizaron las pruebas y mediciones.

Como principal prueba, se probó el software implementado de acuerdo con el APÉNDICE VII. Para ello se simuló los sensores de corriente y voltaje por medio de simples potenciómetros variando con el tiempo a un valor constante. Toda esta información fue capturada por otro dispositivo (o el microprocesador en su caso), a través del CAN BUS, asimismo, se enviaron diferentes mensajes al subsistema de potencia para “desactivar” y obtener el consumo de potencia en cada uno de los reguladores. De la misma manera, se observaron las señales de sincronización utilizados por el mismo, y se hicieron pruebas del microcontrolador operando en condiciones de bajo consumo.

Entonces, como primer punto de este capítulo se observaron las señales de sincronización que recibirán los conmutadores de regulación en su pin de entrada Sync (se debe mencionar que estos no necesariamente se utilizan, pero se usan para mejorar su desempeño, lo cual, se puede observar en las simulaciones) que se generaron de acuerdo con los apéndices APÉNDICE I, APÉNDICE II y APÉNDICE III, los cuales se muestran en las imágenes Figura 83, Figura 84 y Figura 85.

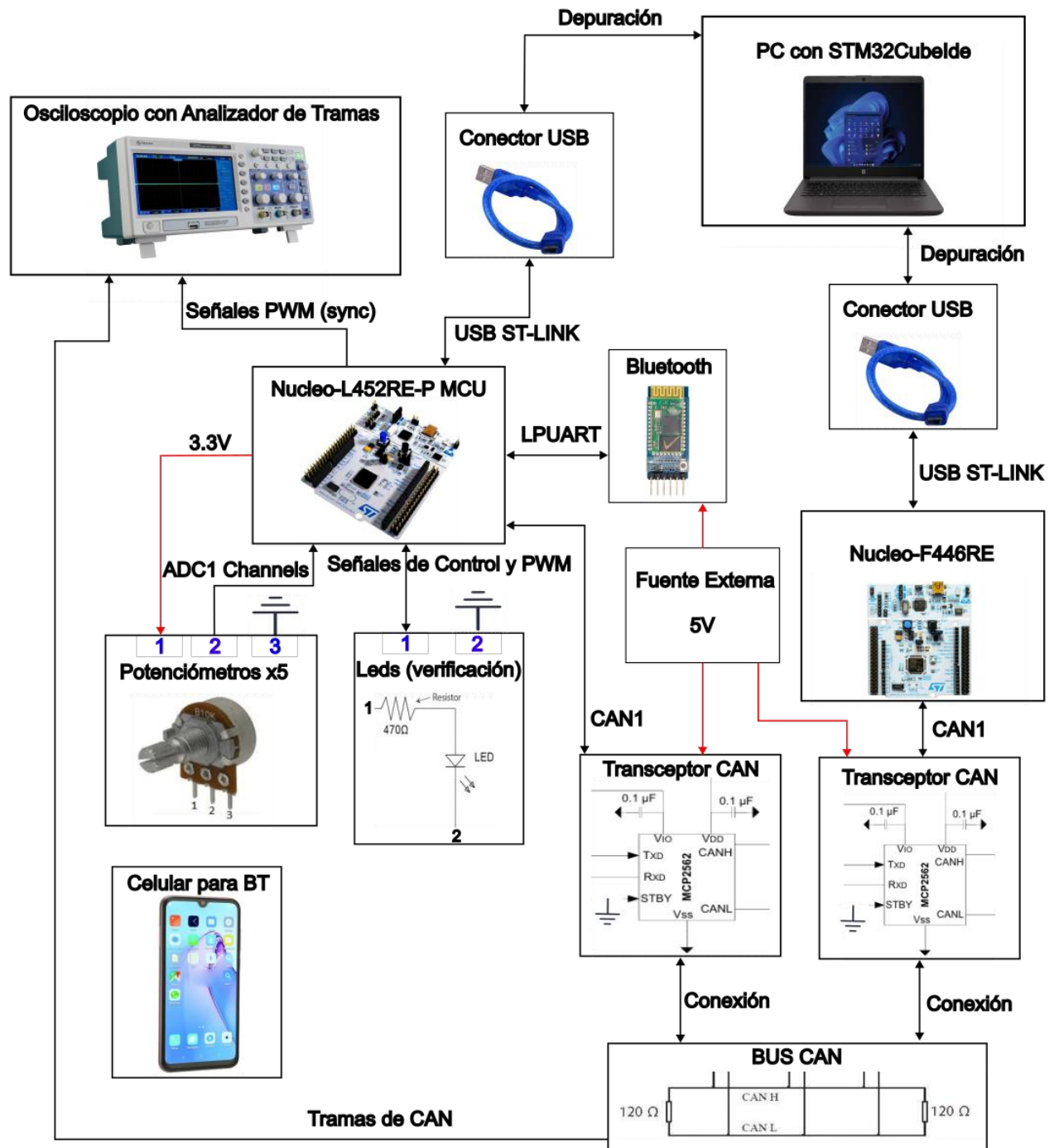


Figura 82. Diagrama del banco de pruebas para la unidad de control del subsistema de potencia.



Figura 83. Señal de sincronización para el regulador de 5[V]



Figura 84. Señal de sincronización para el regulador de 3.3[V].



Figura 85. Señal de sincronización para el regulador de 12[V]

En la Figura 83, Figura 84 y Figura 85 se puede analizar que las señales de sincronización obtenidas están subamortiguadas lo cual no es deseado. Esta señal se puede filtrar y queda pendiente para ser corregido antes de ser implementado con los reguladores de conmutación.

Ahora bien, como prueba principal, se enviaron tramas remotas y de datos a través del bus CAN, las tramas remotas solicitaban información sobre el consumo de corriente de cada sensor y sobre el porcentaje de carga de la batería. Todos estos datos se observaron en ambos dispositivos por medio de un puerto serial, tanto del que solicitaba información, como el que la recibía y respondía (unidad de control del subsistema de potencia). Y, por otro lado, la trama de datos se utilizó para enviar información que solicitara al microcontrolador desactivar o activar algún regulador de voltaje, así como, indicar al dispositivo entrar en el modo LPM. Los resultados se muestran en la Figura 86 y se puede verificar que lo que se envía y recibe coincide en ambos casos. Además, se agrega como prueba la captura de una trama de CAN con el analizador digital (Figura 87).

Port 0 ×

Solicitando informacion del regulador de 3.3V
Identificador a enviar: 10476
Datos recibidos:
252, ON **Tramas remotas e información recibida**
Solicitando informacion del regulador de 5V
Identificador a enviar: 20476
Datos recibidos:
552, ON
Solicitando informacion del regulador de 12V
Identificador a enviar: 40476
Datos recibidos:
30, ON
Solicitando informacion del sensor de alta corriente
Identificador a enviar: 80476
Datos recibidos:
3590
Solicitando informacion de la bateria
Identificador a enviar: 100476
Datos recibidos:
78, 658.372
Solicitando informacion de todo el subsistema
Identificador a enviar: 1F0476
Datos recibidos:
251, 30, 552, 3490; ON, ON, ON; 79; 728

Solicitando conmutacion del regulador de 3.3V
Identificador a enviar: 10476
Solicitando conmutacion del regulador de 12V
Identificador a enviar: 40476
Solicitando conmutacion del regulador de 3.3V
Identificador a enviar: 10476
Solicitando conmutacion del regulador de 12V
Identificador a enviar: 40476 **Tramas de datos**
Solicitando entrar en modo LPM
Identificador a enviar: 10476

Solicitando informacion del regulador de 3.3V
Identificador a enviar: 10476
Solicitando informacion del regulador de 5V
Identificador a enviar: 20476
Solicitando informacion del regulador de 12V
Identificador a enviar: 40476
Tramas remotas

Port 0 ×

Identificador recibido: 10476
Se solicitaron datos
252, ON, **Tramas remotas recibidas y datos enviados**
Identificador recibido: 20476
Se solicitaron datos
552, ON,
Identificador recibido: 40476
Se solicitaron datos
30, ON,
Identificador recibido: 80476
Se solicitaron datos
3590
Identificador recibido: 100476
Se solicitaron datos
78, 658.372
Identificador recibido: 1F0476
Se solicitaron datos de todos los sensores
Enviando datos:
Consumo de corriente: 251, 30, 552, 3490,
ON, ON, ON; 79, 728.901

Identificador recibido: 10476
Recibiendo una trama de datos
Desactivando el regulador de 3.3V
Identificador recibido: 40476 **Tramas de datos recibidas**
Recibiendo una trama de datos **y acciones tomadas**
Desactivando el regulador de 12V
Identificador recibido: 10476
Recibiendo una trama de datos
Activando el regulador de 3.3V
Identificador recibido: 40476
Recibiendo una trama de datos
Activando el regulador de 12V
Identificador recibido: 10476
Recibiendo una trama de datos
Entrando en el modo LP

Figura 86. Resultados obtenidos de la comunicación por medio del bus CAN: (Izq.) Dispositivo con CAN; (dcha.) Subsistema de Potencia

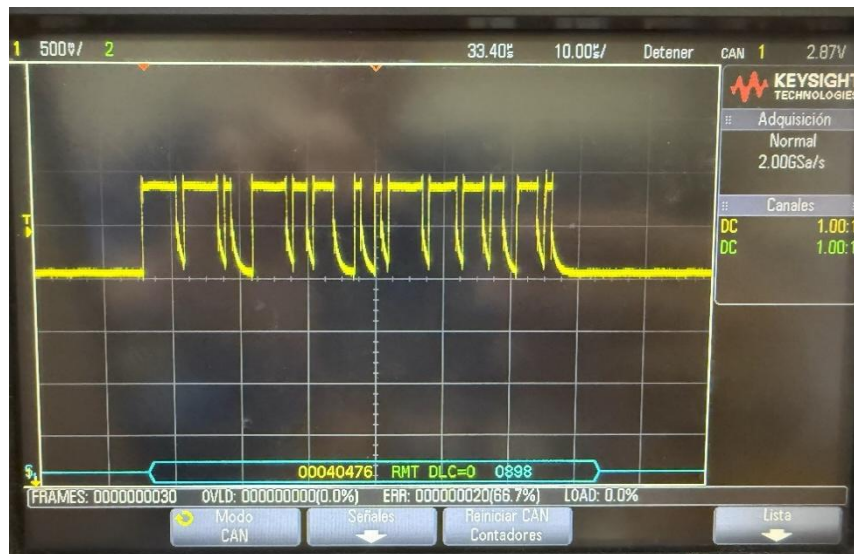


Figura 87. Trama de CAN capturada con el analizador digital.

De acuerdo con la Figura 86, en la trama remota se solicitan datos y el subsistema de potencia responde con una trama de datos, donde estos se reciben correctamente (cuadro rojo); primero se obtienen los datos para cada sensor y, por último, para todos los sensores en conjunto. Posteriormente, se envía una trama de datos hacia el subsistema el cual conmuta o activa o desactiva a los reguladores de 3.3V y 12V. Y, esta secuencia se muestra en las siguientes imágenes tomadas al momento de hacer la comunicación.

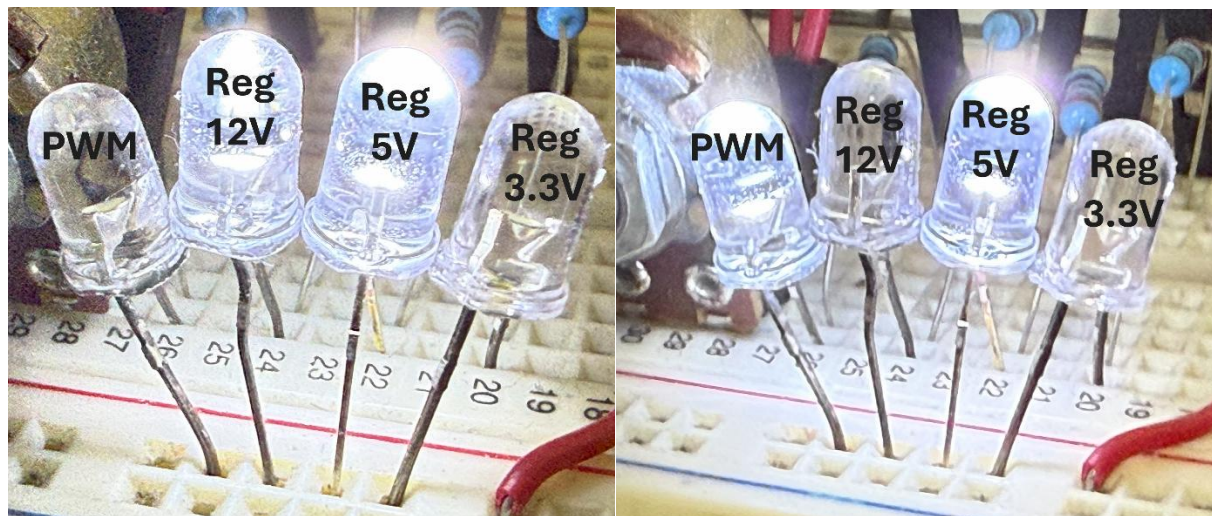


Figura 88. Conmutación de la habilitación de los reguladores de 3.3[V] y 12[V]

De la Figura 88 se muestra como la unidad de control del subsistema de potencia recibe la solicitud para conmutar la habilitación de los reguladores a un estado de inactividad, donde primero se inactiva el regulador de 3.3[V] y, posteriormente el de 12[V], lo cual,

corresponde con los mensajes de la Figura 86 (cuadro azul). Posteriormente, se solicita la activación de estos reguladores, donde primero se habilita el regulador de 3.3[V] (Figura 89) y, después el de 12[V].

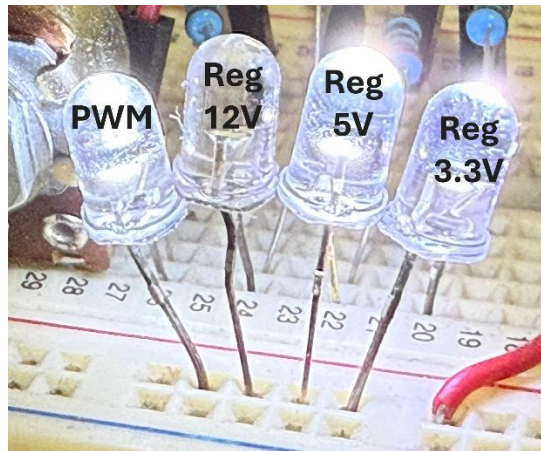


Figura 89. Habilitación del regulador de 3.3[V].

Como último comando, con CAN se envía información para indicar al subsistema de potencia que debe entrar en un modo de operación de bajo consumo (LPM). Cuyo resultado se muestra en la Figura 90. En este modo de operación se desactivan los reguladores y varios periféricos y fuentes de reloj del sistema, y la única forma de salir de este estado es por medio de LPUART a través del módulo bluetooth, por tanto, si se observa la Figura 86, se aprecia que después de entrar en el modo de bajo consumo se le envían comandos para solicitar información del subsistema de potencia, en donde, no se obtiene respuesta por parte de éste y, a su vez, el subsistema de potencia no recibe ningún mensaje porque CAN deja de funcionar en este modo de operación.

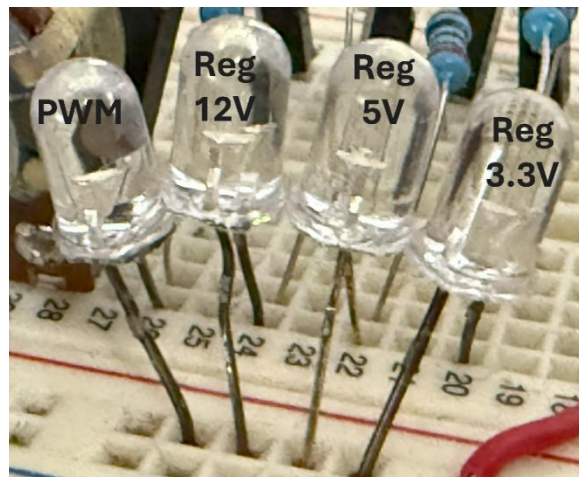


Figura 90. Sistema entrando en modo de bajo consumo

Para observar el funcionamiento del dispositivo bluetooth, primero se configuró este dispositivo por medio de UART para operar a un baudrate de 9600 bps y, a su vez, se cambiaron ciertos parámetros de este. Posteriormente, con ayuda de una aplicación para teléfono celular inteligente que se diseñó especialmente para este proyecto, se realizó el envío de datos del subsistema hacia un celular (para más información de la aplicación véase APÉNDICE XV), cuyos resultados se describen a continuación.

Al iniciar la aplicación se muestra una pantalla de inicio, donde, si se presiona el botón “start” sin tener conectado algún dispositivo, este enviará un mensaje que solicitará conectar un dispositivo antes para poder continuar.

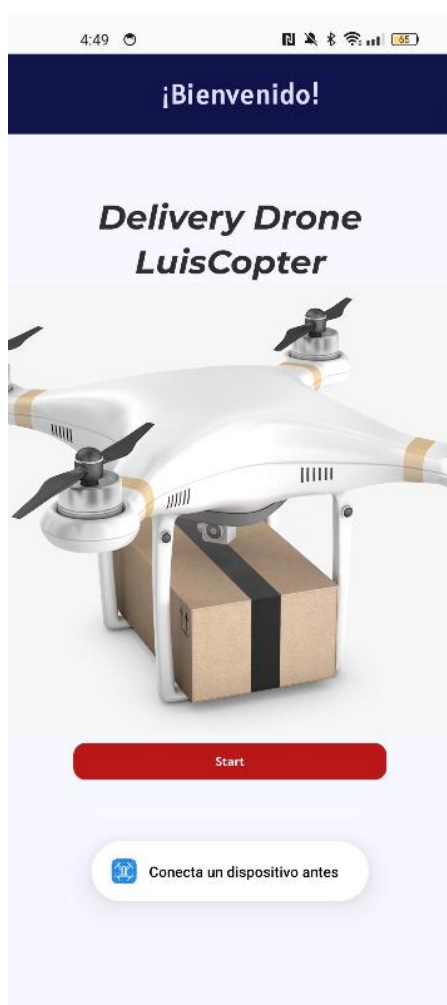


Figura 91. Pantalla de inicio de la aplicación del teléfono celular con aviso de recomendación

Para conectar el dispositivo, se presiona el botón alusivo a esto y, se abre un cuadro de diálogo, en donde se indica la conexión por bluetooth. Una vez seleccionado esta opción,

se abre otro cuadro de diálogo, el cual contiene dos listas: dispositivos emparejados y dispositivos listos para emparejar. En este caso se selecciona el dispositivo denominado como “LuiOSDrone” (utilizado para este proyecto) y se presiona el botón de seleccionar, lo cual, solicitará una contraseña para emparejar el dispositivo. Al agregar esta contraseña empezará el emparejamiento y posteriormente se establecerá la conexión con el dispositivo bluetooth.

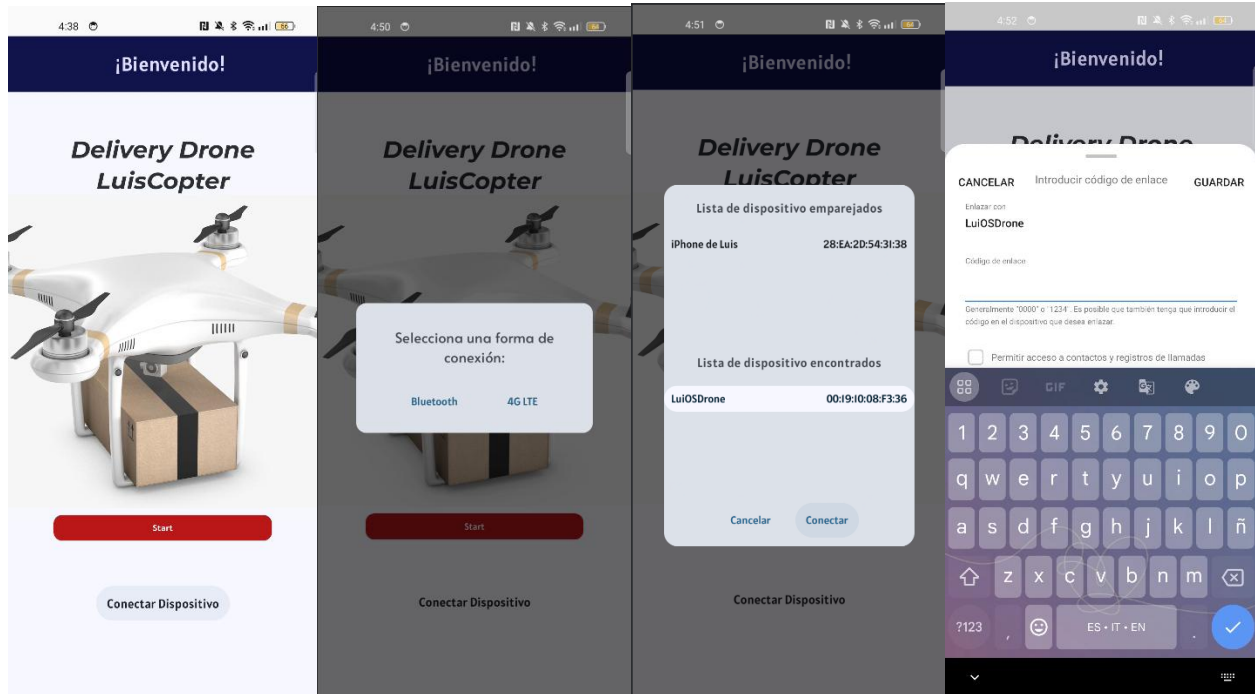


Figura 92. Pasos para emparejas un dispositivo bluetooth

Una vez establecida la comunicación entre ambos dispositivos, la aplicación solicita información sobre el sistema, el cuál será procesada y mostrada en la pantalla denominada “Acerca del sistema”, además, de la misma manera podremos comenzar a navegar en la aplicación.

Sobre esta parte, son 4 pantallas que muestran información o configuran la unidad de control del subsistema por medio de bluetooth, siendo estas:

1. Menú: Se solicita un envío de datos contantes sobre el porcentaje de batería y, este se muestra en una animación de batería.
2. Estado del dron: Se recibe información sobre el estado y consumo de los reguladores, así como el modo de operación del sistema (ej. bajo consumo).
3. Configuración: En esta pantalla, por medio de interruptores se habilita o deshabilita el funcionamiento de los reguladores, al igual que con el modo de bajo consumo.

4. Acerca del sistema: Esta pantalla muestra la información del dron repartidor, solicitada únicamente después de establecer conexión con el dispositivo bluetooth.

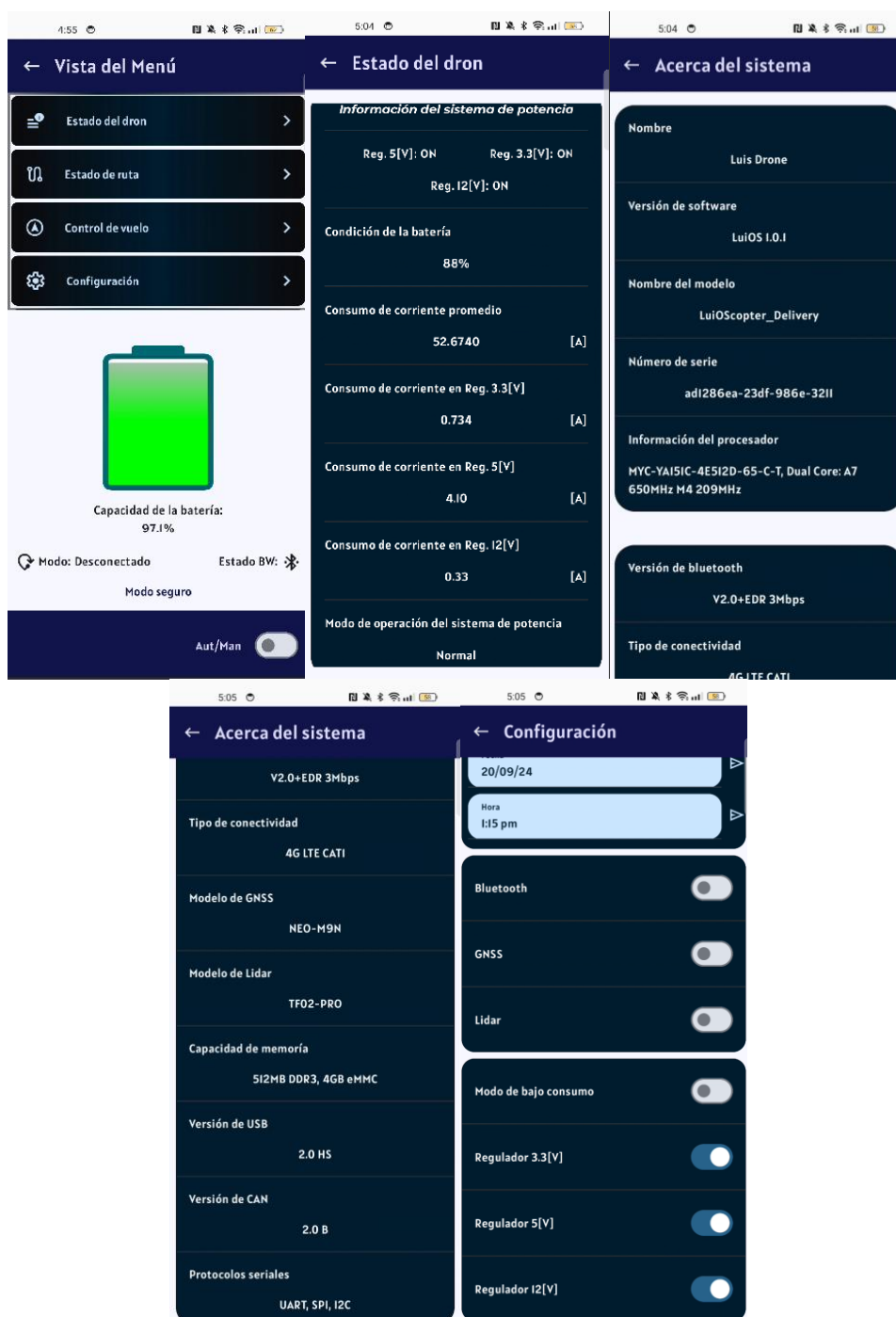


Figura 93. Diferentes pantallas mostrando información obtenida por bluetooth

Como se mencionó, la pantalla “Configuración” permite cambiar ciertos parámetros de operación del sistema. Como ejemplo de su uso, se muestra el funcionamiento con tres

leds que simulan la habilitación y deshabilitación de los reguladores implementados. Cuando se presiona algunos de los interruptores asociados a estos, el led cambia su estado (es decir, se prende o se apaga). Se debe mencionar que, al inicio cuando se solicita información del sistema, también se obtiene información sobre el estado de los reguladores (Encendido de manera predeterminada, Figura 93). Entonces, en este ejemplo se desactivan los reguladores de 3.3[V] y 12[V] y se muestran los resultados.

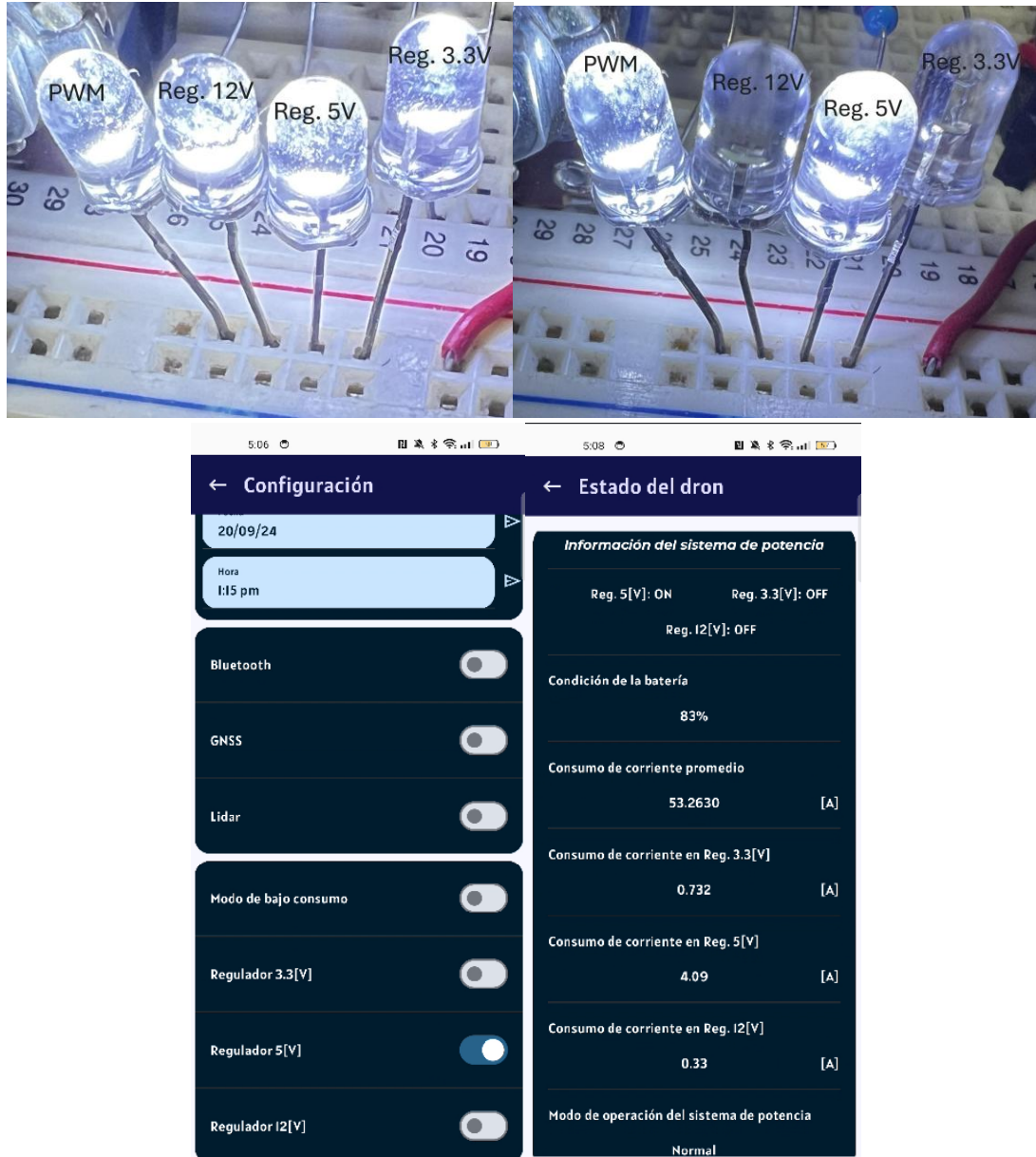


Figura 94. Pruebas de envío de datos por bluetooth por conmutación de leds.

Otro punto importante es configurar el modo de operación del microcontrolador para funcionar en modo de bajo consumo de energía. En este estado, se detiene incluso el

procesador de este mismo. Una forma en que se visualiza esto es con ayuda de la señal PWM, la cual, se detiene porque no se configura para operar en modo de bajo consumo.



Figura 95. Configuración en modo de bajo consumo

Posteriormente, si regresamos al inicio se podrá observar como la carga de la batería disminuyó debido a la operación continua de los potenciómetros simulando la carga en los sensores de corriente.

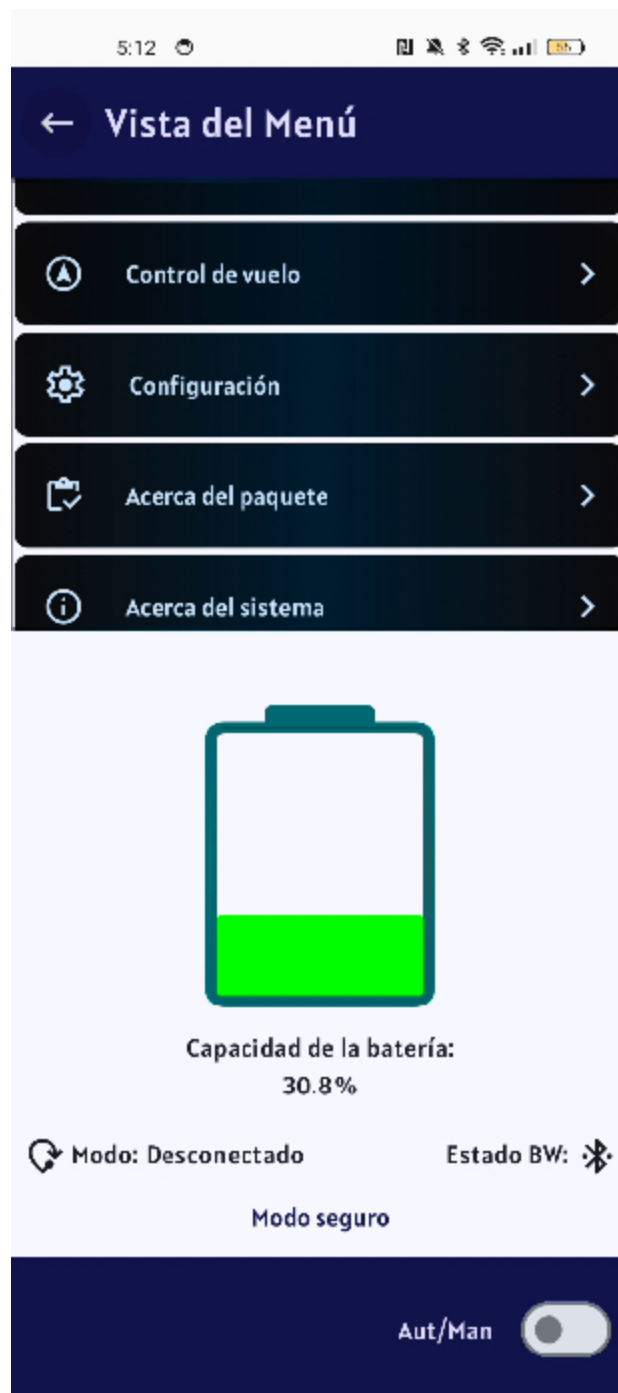


Figura 96. Estado de la batería después de operar durante un tiempo.

Finalmente, se debe destacar que durante estas pruebas estuvieron operando un temporizador a 1KHz que disparaba la medición de la secuencia de muestreo en los canales del ADC los cuales muestreaban los potenciómetros que simulaban el comportamiento de los sensores de corriente y voltaje. Asimismo, se encontraban operando los periféricos CAN, LPUART (bluetooth), y el reloj de tiempo real en conjunto. Aquí se debe destacar únicamente que el reloj en tiempo real se utiliza periódicamente para observar el estado del subsistema (Figura 97) y, si encuentra inactividad, activa el modo de bajo consumo. Para la prueba del RTC se comparó las horas de captura con las del reloj mundial del computador, en donde se tomaron la captura al inicio y al final después de entrar en el modo de bajo consumo y, en este se comprueba que hizo la captura en cada intervalo de un minuto.

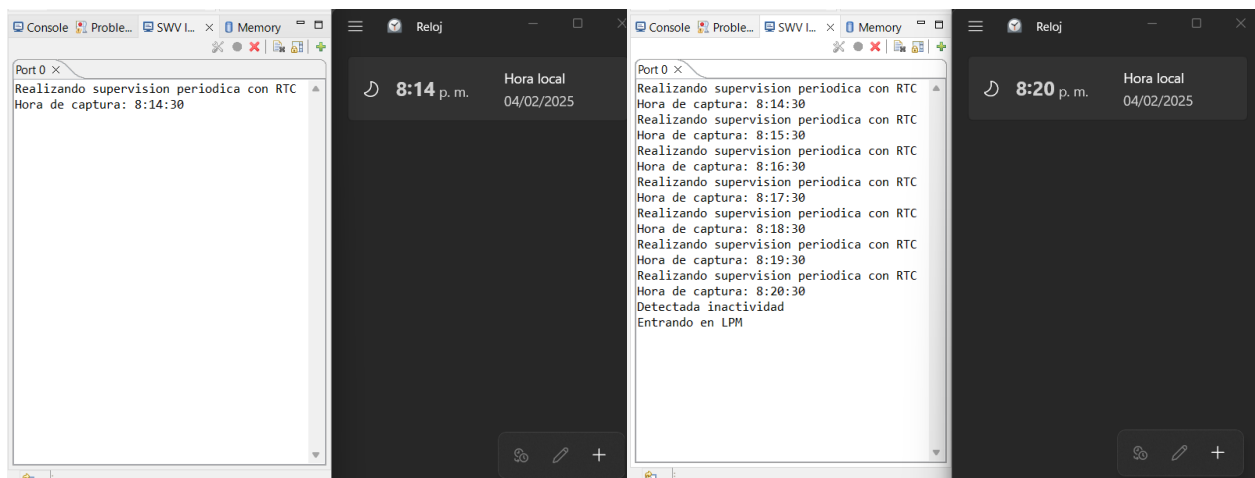


Figura 97. Capturas del tiempo en que supervisó el RTC el subsistema de potencia.

5.3 Desempeño de la computadora de abordó

La computadora de abordó como computadora central capaz de dar la autonomía del dron, requirió de diversas pruebas e implementación de software. Los resultados obtenidos que permiten la integración del hardware y software de la computadora de abordó son los siguientes:

1. Generación de una PCB para pruebas y facilitación de conexión con los dispositivos de los otros subsistemas (APÉNDICE IV). Para esto se pueden utilizar muchas guías disponibles en la red, donde las principales utilizadas se encuentra en [53] para la asignación de pines y en [55] se dan recomendaciones para el diseño de las interfaces de los diferentes periféricos.
2. Generación de software y drivers para la computadora de abordó.
3. Generación de algoritmos de control para la autonomía del sistema siendo estos:
 - Algoritmo Convex Hull para obtener la envolvente de la figura,

- RANSAC en la segmentación para generar un modelo adecuado para los puntos.
- DBSCAN en el agrupamiento de diferentes objetos, permitiendo diferencias un objeto de otro,
- Voronoi para la generación de rutas con ayuda de LIDAR.
- Dijkstra en la selección de la ruta más corta hacia el destino.

Con los puntos anteriores se hicieron diferentes simulaciones para verificar la funcionalidad del código y, asimismo, se verificó su desempeño y tiempo de ejecución para cada caso.

Entonces, una vez que se generó el prototipo de la computadora de abordo (Figura 98), se procedió a realizar una prueba básica del sistema para verificar la funcionalidad del hardware.

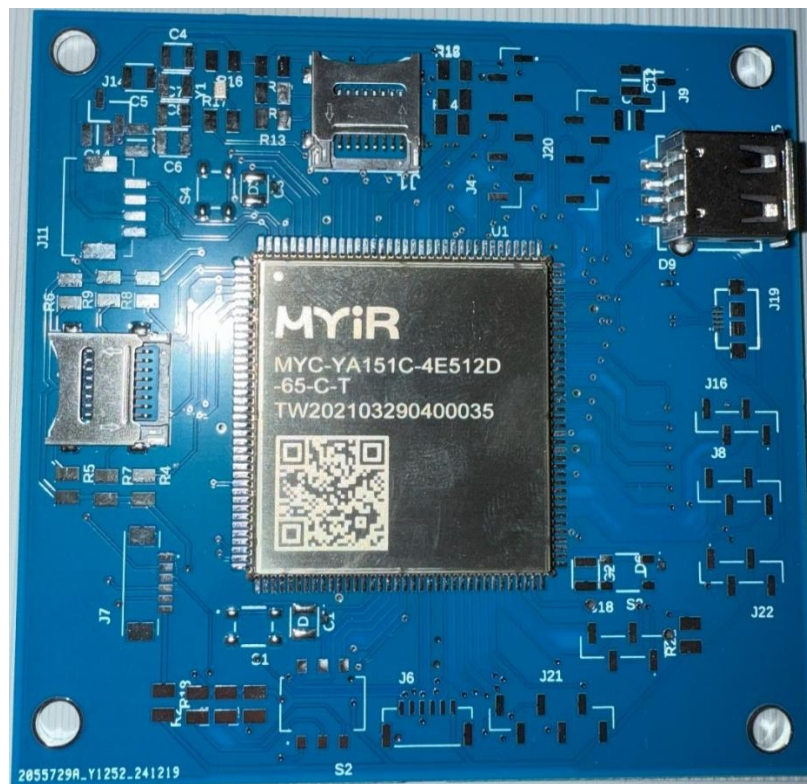


Figura 98. Prototipo de la computadora de abordo²⁶.

²⁶ En la imagen se muestran algunos dispositivos conectados, sin embargo, para realizar las pruebas que se describen posteriormente, es necesario soldar otros dispositivos tales como: los conectores J14, J18, J19 (USB) y J22 (UART4); las resistencias y capacitores para los pines de la uSD (MMC1); y el dip switch utilizado para controlar el arranque del dispositivo.

La prueba de hardware más sencilla es utilizando el software STM32CubeProgrammer mediante la conexión a través de USB o de un convertidor de USB a UART, haciendo uso de UART7 (conector J4). Entonces, al iniciar el software programador, se escoge una de estas dos opciones y, se identifica el puerto donde se encuentra el dispositivo. Posteriormente, se seleccionará la opción “Connect” y si todo está funcionando correctamente, veremos en la terminal el ID (0x500) del dispositivo y las características de la conexión, justo como se muestra en la Figura 99.



Figura 99. Prueba de funcionamiento mínimo de la computadora.

Ahora bien, lo anterior solo es una prueba básica de la integridad del microprocesador dentro del SOM, pero el siguiente paso es programar el dispositivo con esta herramienta de desarrollo, para ello se debe utilizar una imagen del sistema operativo que acepte y cargue el bootROM. Esta imagen se puede obtener de diferentes formas, una de ella es

obtenerla por medio de Yocto Project (código abierto) o utilizar una de las tantas imágenes que proporcionan los fabricantes de estos dispositivos. En este caso, para disminuir tiempo se hará uso de una de las imágenes adaptadas y proporcionadas por medio del soporte en [53], que consiste en un sistema operativo basado en Linux.

Una vez con la carpeta que contiene las imágenes del sistema operativo, se recomienda seguir el video tutorial presentado por STMicroelectronics en [57]. Donde se debe destacar que para esta parte será necesario utilizar el conector de USB (J19) en la programación del dispositivo.

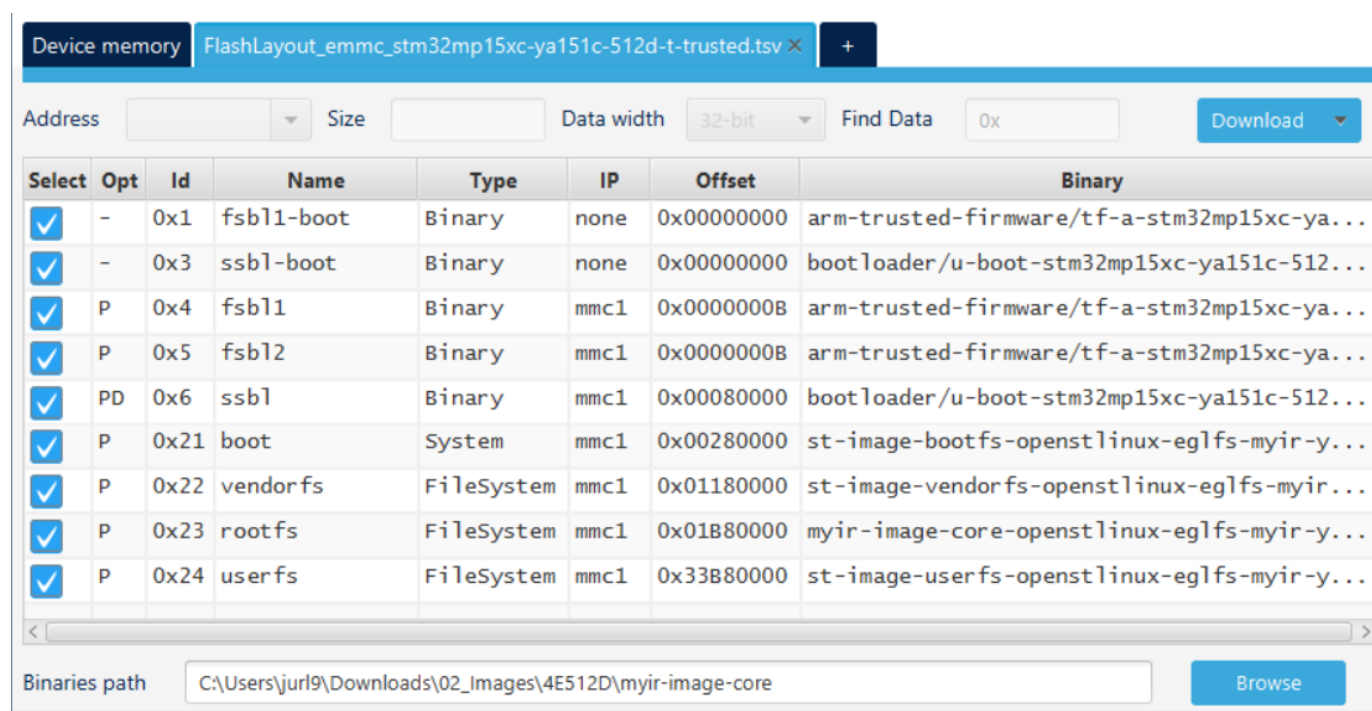


Figura 100. Ejemplo de imagen cargada a la computadora.

En la Figura 100 se observa en la parte superior una pestaña en azul, con el nombre del archivo seleccionado que indicará los archivos a cargar en memoria emmc (de la misma manera se puede programar en una uSD) y, en la parte inferior se encuentra la ruta que contiene todos los archivos necesarios para cargar la imagen. Una vez teniendo seleccionados estas dos partes, se procede a dar clic en el botón de “Download” y esperar a que el dispositivo se programe.

Después de terminar de programar la imagen en el dispositivo (dependiendo si sea en la memoria eMMC o la uSD), se debe de quitar la alimentación y por medio del seleccionador de arranque (dip switch) seleccionar la forma en la que el dispositivo arrancará. En este caso se selecciona la memoria uSD como se muestra en Figura 101 y, se debe verificar que la memoria uSD esté conectada al programar y después de iniciar el arranque.

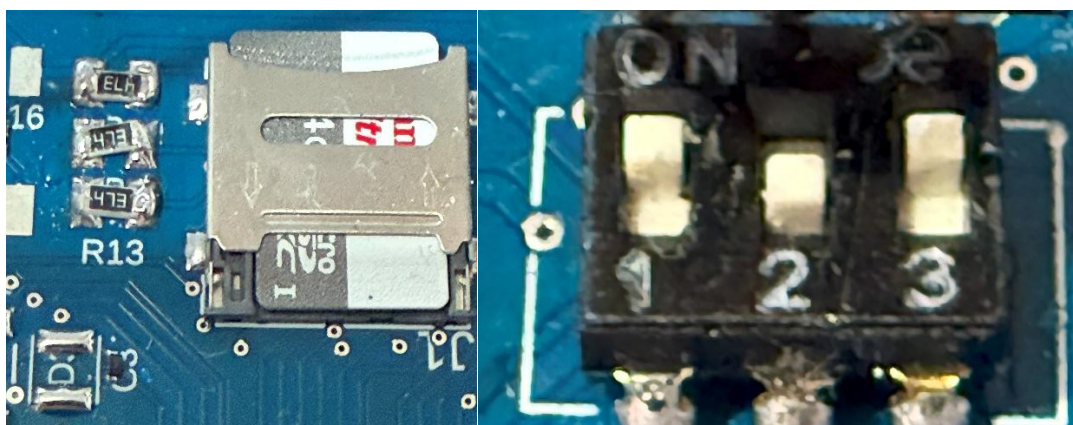


Figura 101. Selección de la memoria uSD como arranque.

Ahora bien, para comunicarse con el dispositivo desde una pc o laptop, se debe usar estrictamente la comunicación serial por UART4 a través de un convertidor a USB, esto se debe principalmente a las características de la imagen que se cargó en el dispositivo. Y, para poder observar la terminal de la computadora, podemos usar varios métodos, pero en este caso se usará una máquina virtual cargada con Ubuntu 16 con el cual estableceremos conexión por medio de minicom justo como se muestra en la Figura 102.

```

luisju@UBUNTU16: ~
luisju@UBUNTU16:~$ ls /dev/ttyUSB* /dev/ttyACM*
ls: cannot access '/dev/ttyACM*': No such file or directory
/dev/ttyUSB0
luisju@UBUNTU16:~$ sudo minicom -D /dev/ttyUSB0
[sudo] password for luisju: 
Welcome to minicom 2.7
OPTIONS: I18n
Compiled on Nov 15 2018, 20:18:47.
Port /dev/ttyUSB0, 13:30:07
Press CTRL-A Z for help on special keys
root@mylr-ya151c-t-4e512d:~#

```

Figura 102. Comunicación con la computadora a través de minicom.

Al establecer la comunicación podremos observar el arranque del sistema en la computadora o en su caso, si ya terminó, no se observará nada más, por lo cual, presionando alguna tecla, podremos observar que aparece el nombre y ruta del dispositivo.

Con lo anterior ya se pueden enviar programas ejecutables a la computadora de a bordo y, probar en esta su tiempo de ejecución. Pero, antes de iniciar con estas pruebas se recomiendan seguir el manual denominado “MYD-YA15XC-T Linux System Development Guide” el cual se puede solicitar en [53], como se mencionó anteriormente. Para ello se recomiendan leer o seguir los capítulos 2 y 7, esto porque necesitamos utilizar un compilador cruzado para generar los ejecutables de la computadora de abord y, estos son los que se podrán enviar y ejecutar directamente en la computadora de abord.

Además, si los archivos se encuentran en diferentes carpetas, se necesitará de la generación de un archivo Makefile.

Dadas las consideraciones anteriores enviaremos los archivos ejecutables a la computadora y se obtendrán los tiempos de ejecución.

Algoritmo Convex Hull

El algoritmo Convex Hull nos da como resultado la envolvente de una muestra de puntos, este algoritmo se verificó con dos pruebas, la primera fue usar 15 puntos como entrada y, la segunda fue usar 22 puntos de entrada, para ambos se aplicó el algoritmo y se obtuvieron los puntos que conforman a la envolvente (Figura 103), junto con el tiempo de ejecución (Tabla 27).

Tabla 27. Tiempo de ejecución del algoritmo Convex Hull

#Puntos de entrada	de Tiempo de ejecución [μs]
15	550.29
22	559.20

```

luisju@UBUNTU16: ~
root@myir-yai51c-t-4e512d:~# nice -n -20 ./Convex1
Imprimiendo valores del polígono:
{4.000000, 0.000000},
{11.000000, 2.000000},
{9.000000, 9.000000},
{4.000000, 12.000000},
{1.000000, 8.000000},
{1.000000, 1.000000},
Tiempo de ejecución: 0.000550291 segundos
root@myir-yai51c-t-4e512d:~# nice -n -20 ./Convex2
Imprimiendo valores del polígono:
{4.000000, 0.000000},
{11.000000, 2.000000},
{14.000000, 6.000000},
{12.000000, 10.000000},
{8.000000, 13.000000},
{5.000000, 15.000000},
{2.000000, 14.000000},
{0.000000, 10.000000},
{1.000000, 1.000000},
Tiempo de ejecución: 0.000559209 segundos

```

Figura 103. Resultados obtenidos al aplicar el algoritmo Convex Hull

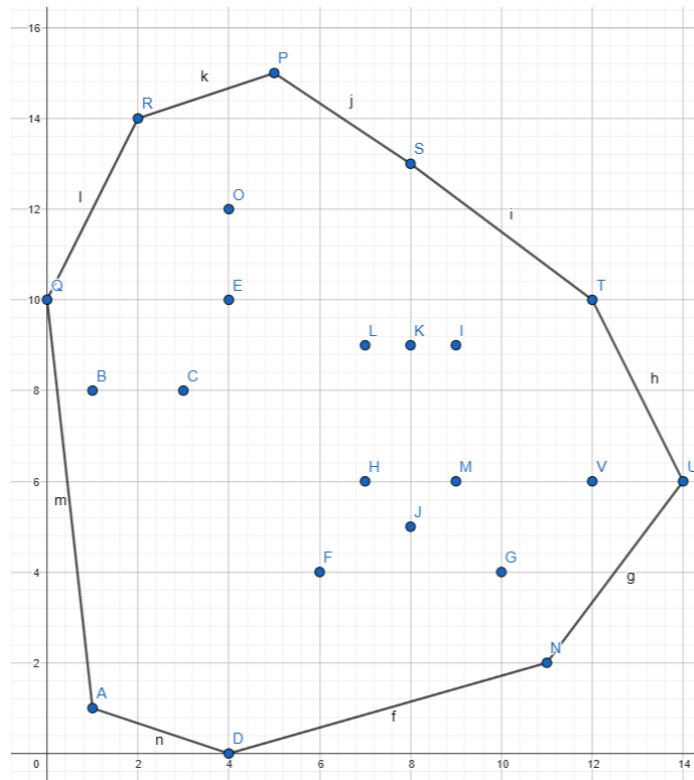


Figura 104. Resultado de la envolvente visualizado de manera gráfica.

En la Figura 104 se observa de forma gráfica la envolvente que calculó el programa con los 22 puntos de entrada proporcionados, en este caso, se puede apreciar que todos los puntos al interior fueron descartados de la envolvente.

Algoritmo RANSAC

Con el algoritmo de RANSAC se hicieron dos pruebas con los mismos parámetros para dos tipos de datos de puntos, los cuales se ajustaron para un modelo de cuadrilátero. En la Tabla 28 se observan los parámetros y tiempos de ejecución.

Tabla 28. Pruebas realizadas con RANSAC

Parámetros: $n = 20$, $p = 0.99$. Donde n son los puntos necesarios a contener por el modelo y p es la probabilidad de que el modelo sea la mejor aproximación.	
#Puntos de entrada	Tiempo de ejecución [s]
28	0.2123
30	1.3795

En la Figura 105 se observan los resultados que arroja el algoritmo para diferentes puntos muestra. Estos valores corresponden a los vértices del modelo de un polígono que mejor se adapta a todos los datos; el resultado de manera gráfica se puede visualizar en la Figura 106.

```

luisju@UBUNTU16: ~
root@myir-ya151c-t-4e512d:~# nice -n -20 ./RANSAC1
Imprimiendo valores del modelo:
{1.000000, 1.000000},
{1.000000, 8.000000},
{8.000000, 8.000000},
{8.000000, 2.000000},
Tiempo de ejecucion: 0.212385875 segundos
root@myir-ya151c-t-4e512d:~# nice -n -20 ./RANSAC2
Imprimiendo valores del modelo:
{3.000000, 0.000000},
{0.000000, 4.000000},
{5.000000, 9.000000},
{9.000000, 1.000000},
Tiempo de ejecucion: 1.379591417 segundos
root@myir-ya151c-t-4e512d:~#

```

Figura 105. Resultados obtenidos con RANSAC.

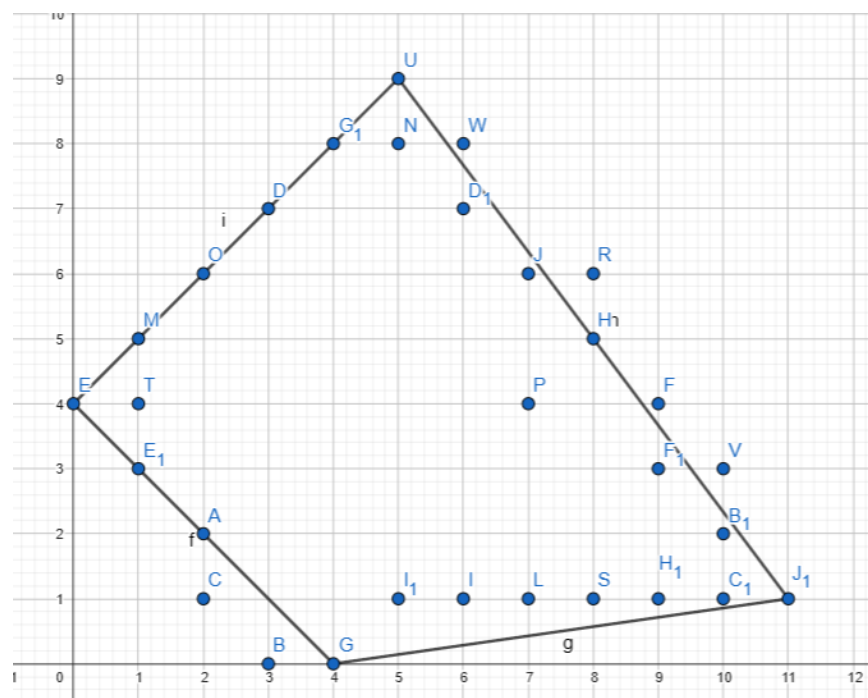


Figura 106. Representación gráfica de los resultados

Se debe destacar que a pesar de que ambas pruebas tienen la misma cantidad de datos de entrada, el tiempo de ejecución varía bastante debido a la forma de los datos, y a la manera en que se ejecuta el algoritmo. Además, en la primera prueba el modelo de puntos no fue el óptimo, pero sí el más aproximado, esto debido a que como datos de entradas

se dieron los puntos de un cuadrado y, como resultado, se obtuvieron los puntos para un polígono irregular.

Algoritmo DBSCAN

Con DBSCAN se realizaron dos pruebas para diferentes entradas de puntos con los mismos parámetros de análisis, con las cuales, se lograron distinguir dos objetos en el espacio. En la Tabla 29 se observan los parámetros y tiempos de ejecución.

Tabla 29. Tiempos de ejecución de DBSCAN.

Parámetros: -Mínimo número de puntos: 5 -Distancia máxima: $\varepsilon = 2[u]$	
#Puntos de entrada	Tiempo de ejecución [μs]
14	553.87
22	699.37

En la Figura 107 se muestran como resultados dos nodos que hacen referencia a los objetos, los cuales, se conforman de núcleos y bordes. Si se revisan estos núcleos y bordes, se obtendrá que algunos puntos se repiten, los cuales, son los que nos indican si un núcleo forma parte o no de un objeto.

```
luisju@UBUNTU16: ~/Projects/Simulation_F446RE
root@mylr-ya151c-t-4e512d:~# nice -n -20 ./DBSCAN1
-----Object 1:-----
Core 1: {6.000000, 4.000000},
Borders: {5.000000, 3.000000}, {4.500000, 5.000000}, {7.000000, 3.000000}, {7.000000, 5.500000},
Core 2: {8.000000, 7.000000},
Borders: {7.000000, 5.500000}, {7.000000, 8.000000}, {9.000000, 8.000000}, {9.000000, 6.000000},
-----Object 2:-----
Core 1: {14.000000, 4.000000},
Borders: {12.500000, 5.000000}, {15.000000, 5.000000}, {12.500000, 3.000000}, {15.000000, 2.500000},
Tiempo de ejecucion: 0.000553875 segundos
root@mylr-ya151c-t-4e512d:~# nice -n -20 ./DBSCAN2
-----Object 1:-----
Core 1: {6.000000, 4.000000},
Borders: {5.000000, 3.000000}, {4.500000, 5.000000}, {7.000000, 3.000000}, {7.000000, 5.500000},
Core 2: {8.000000, 7.000000},
Borders: {7.000000, 5.500000}, {7.000000, 8.000000}, {9.000000, 8.000000}, {9.000000, 6.000000},
Core 3: {6.500000, 9.500000},
Borders: {5.500000, 11.000000}, {7.500000, 11.000000}, {5.000000, 8.500000}, {7.000000, 8.000000},
-----Object 2:-----
Core 1: {14.000000, 4.000000},
Borders: {12.500000, 5.000000}, {15.000000, 5.000000}, {12.500000, 3.000000}, {15.000000, 2.500000},
Core 2: {16.000000, 6.500000},
Borders: {15.000000, 8.000000}, {17.000000, 8.000000}, {17.000000, 5.000000}, {15.000000, 5.000000},
Tiempo de ejecucion: 0.000699375 segundos
root@mylr-ya151c-t-4e512d:~#
```

Figura 107. Resultados obtenidos con RANSAC.

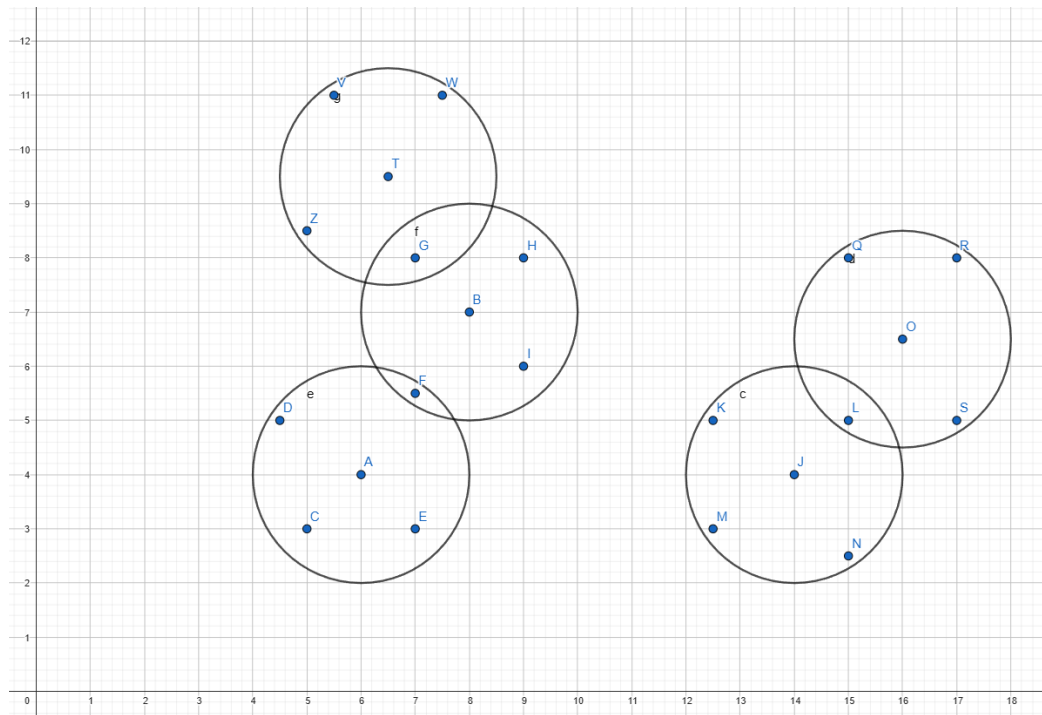


Figura 108. Representación gráfica de los resultados

Finalmente, en la Figura 108, se muestran los núcleos, encerrados por una circunferencia, la cual, contiene otros puntos que coinciden con los bordes de cada núcleo obtenido. Entonces, se logra apreciar que efectivamente son dos objetos formados por diferentes cantidades de núcleos y bordes.

Algoritmo de Voronoi

Con el algoritmo de Voronoi se realizaron dos pruebas considerando un incremento de 0.01[u] y los puntos de entrada como obstáculos, donde, el tiempo de ejecución para cada prueba se muestra en la Tabla 30. Y, los puntos de Voronoi obtenidos para cada caso se muestran en la Figura 109, destacando que estos puntos están interconectados a niveles de estructuras, es decir cada uno de estos puntos está enlazado para poder desplazarnos hacia otro punto contenido en este mismo.

Tabla 30. Tiempos de ejecución de algoritmo Voronoi.

#Puntos de entrada	de Tiempo de ejecución [ms]
7	8.1
10	9.83

```

luisju@UBUNTU16: ~/Projects/Simulation_F446RE
root@myir-ya151c-t-4e512d:~# nice -n -20 ./Voronoi1
Imprimiendo puntos de Voronoi:
{2.500000, 15.495041}, {0.591370, 4.567260}, {1.248632, 3.252735}, {-10.081258, 25.912514},
{1.667846, 2.832154}, {1.250000, 2.293963}, {0.977409, 2.235482}, {2.358676, 1.403059},
{0.942288, 2.233084}, {0.625712, 2.359715}, {0.950000, 1.010655}, {-2.393477, 3.567391},
{2.500000, 2.756251}, {3.489897, 2.750001}, {2.250000, 2.304950}, {2.868781, 1.631219},
{5.728567, 2.749999},
Tiempo de ejecución: 0.008097959 segundos
root@myir-ya151c-t-4e512d:~# nice -n -20 ./Voronoi2
Imprimiendo puntos de Voronoi:
{2.500000, 15.495041}, {3.499999, 15.495056}, {0.591370, 4.567260}, {1.248632, 3.252735},
{-10.081258, 25.912514}, {1.667846, 2.832154}, {1.250000, 2.293963}, {0.977409, 2.235482},
{2.231343, 1.504926}, {0.942288, 2.233084}, {0.625712, 2.359715}, {0.950000, 1.010655},
{-2.393477, 3.567391}, {2.240908, 2.259092}, {3.500000, 2.756251}, {2.500000, 2.756251},
{3.608873, 2.553548}, {5.223125, 3.199247}, {3.618744, 1.811245}, {6.689255, 3.653553},
{3.430754, 1.292260}, {7.289599, 4.025843}, {2.284813, 1.357711},
Tiempo de ejecución: 0.009838000 segundos
root@myir-ya151c-t-4e512d:~#

```

Figura 109. Resultados obtenidos con el algoritmo de Voronoi

Se debe hacer constar que estos puntos si bien son un obstáculo cuyas dimensiones varían, sin embargo, se puede hacer la representación por medio del centroide de cada obstáculo, en este caso, si se trabaja con polígonos irregulares bastaría con obtener el centroide de cada uno de estos polígonos y sustituirlos en el algoritmo de Voronoi para obtener las rutas que permitan circular entre estos obstáculos.

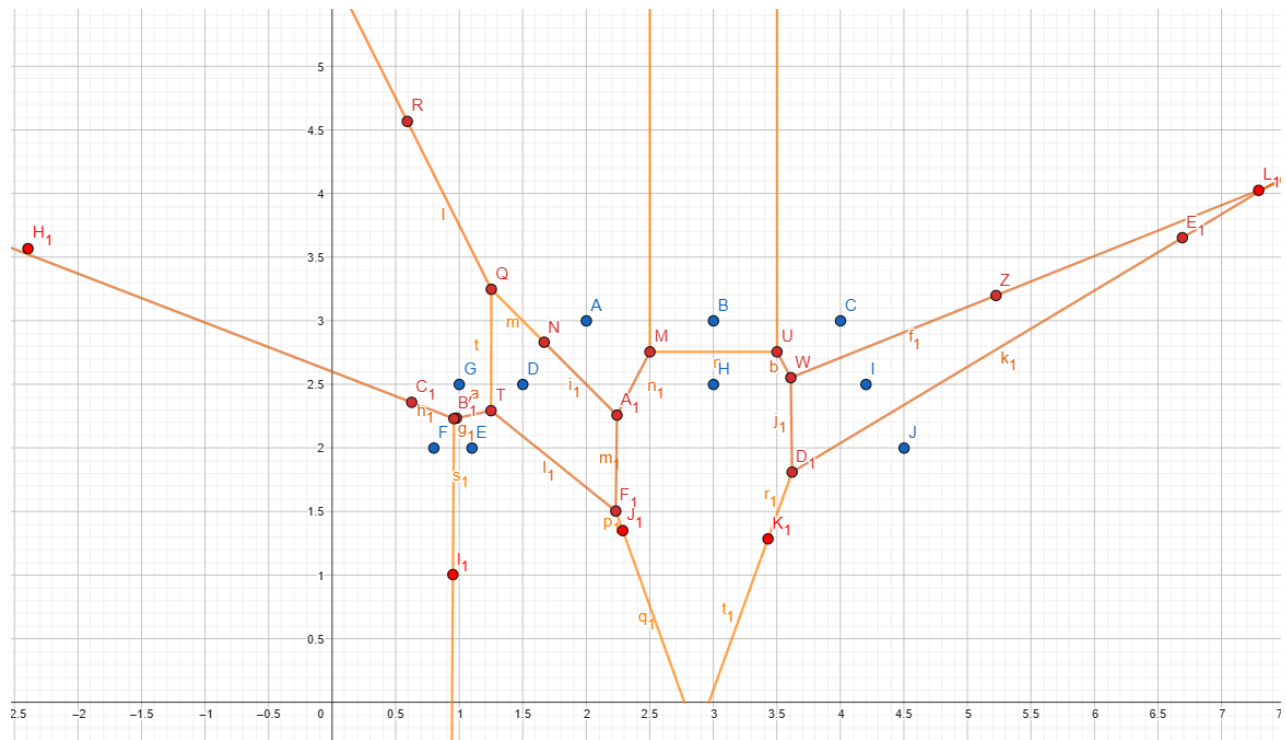


Figura 110. Representación gráfica de los puntos obtenidos con Voronoi

En la Figura 110 se observa de manera gráfica cómo se pueden representar los puntos de Voronoi obtenidos anteriormente. En una primera instancia tenemos de color azul los obstáculos que corresponde a los centroides de estos, posteriormente, al ejecutar el algoritmo de Voronoi obtenemos una serie de puntos enmarcados con color rojo, los cuales son nodos que se encuentran interconectados a diferentes puntos, al ir de un nodo a otro generamos las trayectorias definidas con líneas anaranjadas y, estas líneas son las trayectorias por las cuales circulará el dron sin chocar con los obstáculos.

Algoritmo de Dijkstra

Finalmente, el algoritmo de Dijkstra se probó con referencia a los puntos de Voronoi obtenidos en la Figura 110. Estos puntos se utilizaron de tal manera que se seleccionó un punto de partida y un punto de destino y, una vez definidos ambos, se ejecutó el algoritmo para calcular la ruta más corta, donde el tiempo de ejecución del algoritmo para estas pruebas se muestran en la Tabla 31.

Tabla 31. Tiempo de ejecución del algoritmo Dijkstra

#Puntos de entrada	de Tiempo de ejecución [μ s]
16 (S1 a O)	250.16
16 (L1 a C1)	242.83

En la Figura 111 se observan los resultados de ejecutar el algoritmo de Dijkstra indicando los nodos interconectados, así como los nodos de inicio y destino. Los puntos que se obtienen son las coordenadas de los nodos por los cuales debe moverse el dispositivo para llegar a su destino recorriendo la menor distancia posible.

```

luisju@UBUNTU16: ~
root@mylr-ya151c-t-4e512d:~# nice -n -20 ./Dijkstra1
Imprimiendo ruta:
{2.800000, 0.000000}->{2.230000, 1.500000}->
->{2.240000, 2.260000}->{2.500000, 2.750000}->
->{2.500000, 15.500000}
Tiempo de ejecucion: 0.000250167 segundos
root@mylr-ya151c-t-4e512d:~# nice -n -20 ./Dijkstra2
Imprimiendo ruta:
{7.290000, 4.020000}->{3.610000, 2.550000}->
->{3.500000, 2.750000}->{2.500000, 2.750000}->
->{2.240000, 2.260000}->{2.230000, 1.500000}->
->{1.250000, 2.290000}->{0.980000, 2.230000}->
->{0.620000, 2.360000}
Tiempo de ejecucion: 0.000242833 segundos
root@mylr-ya151c-t-4e512d:~#

```

Figura 111. Resultados obtenidos de la ejecución del algoritmo de Dijkstra

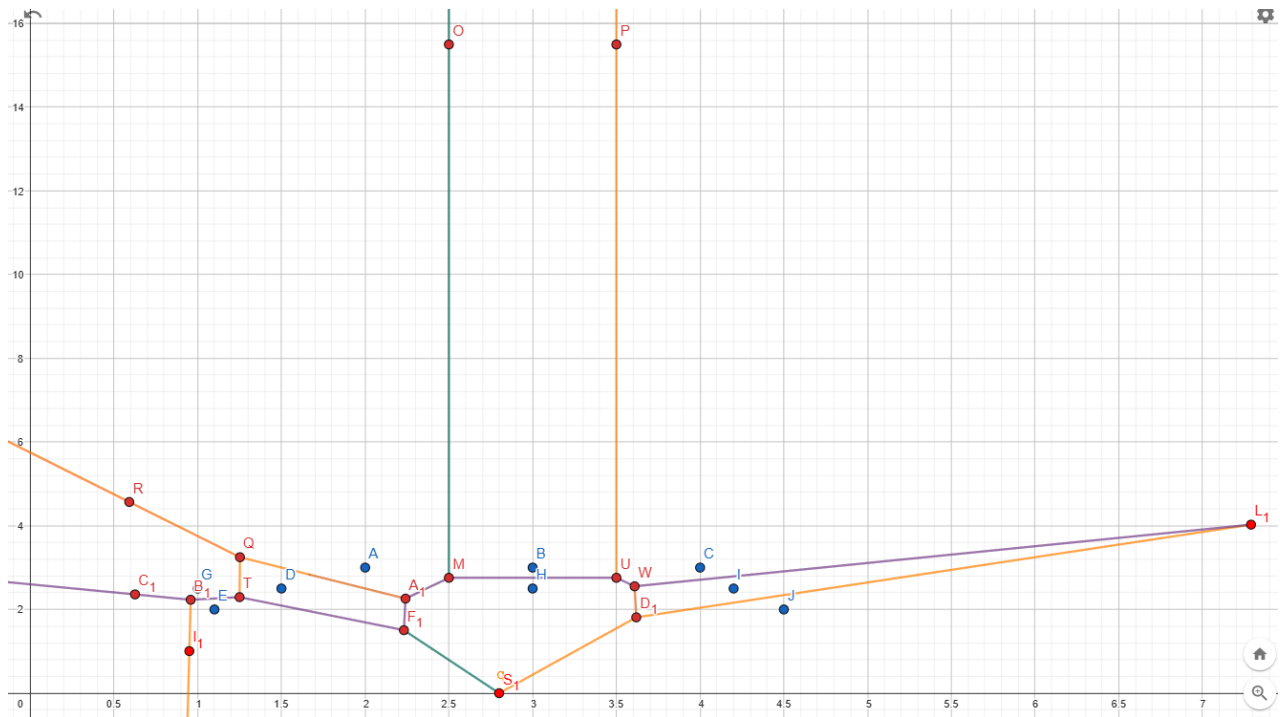


Figura 112. Representación gráfica de los resultados obtenidos con el algoritmo.

La Figura 112 muestra todos los nodos y cómo se interconectan para generar rutas, estas trayectorias se ven de diferentes colores. En un inicio todas las conexiones se marcan en anaranjado, mientras que las rutas generadas con el algoritmo se muestran en morado y verde. La ruta de menor costo para el primer caso, que se describe en la ruta verde, va del punto S1 hacia el punto O y este contiene a los nodos F1, A1 y M; la ruta de menor costo para el segundo caso se marca en color morado y va del punto L1 hacia C1, conteniendo a los nodos W, U, M, A1, F1, T y B1.

5.4 Resultados del subsistema de posicionamiento

Las características del subsistema de posicionamiento se fundamentan en el capítulo 4.3.5. En este caso, se estableció comunicación entre el módulo GNSS y el barómetro y, con estos datos, se determinó la posición del objeto. Los resultados de este subsistema son los siguientes:

1. Se generaron drivers para configurar y obtener datos del subsistema de posicionamiento por medio de UART.
2. Se realizó la interpretación de la posición del objeto.

Para las pruebas únicamente se colocó el GNSS en diferentes sitios no muy lejanos, donde se observó el cambio e interpretación de la localización.

gnss_msj	MSJ_HandlerDef	{...}
msj	MSJ_Parameters...	{...}
> class_or_TT	uint8_t *	0x200002f0 <class> "\001\001"
> ID_or_SSS	uint8_t *	0x200002f1 <id> "\001"
protocol	uint8_t	2 '\002'
typePacket	uint8_t	5 '\005'
> payload	uint8_t *	0x20000160 <payloadData> "àC è..."
ACK_Decript	uint8_t	0 '\0'
CK	_Bool	false
length	uint16_t	92
frameCounter	uint16_t	100
payloadCounter	uint16_t	94
packetValidation	uint8_t	2 '\002'
ID_n_Class_Validation	uint8_t	2 '\002'
> auxPacket	uint8_t *	0x20000228 <auxPacket> "\001\001\001"
ignorePayload	_Bool	false

Figura 113. Información de mensaje recibido por el GNSS.

En la Figura 113 se observa una estructura que contiene información referente al mensaje sobre la navegación del dispositivo. En este se verifica con dos variables denominadas “packetValidation” y “ID_n_Class_Validation” que el mensaje que se envió por el GNSS es correcto, es decir, la clase e ID que se solicitaron fueron respondidas en el mensaje y, además, se verificó con el “checksum” la integridad de los datos.

longitudF	float [25]	0x2001ff88	latitudF	float [25]	0x2001ff24
longitudF[0]	float	-99.1514282	latitudF[0]	float	19.4702702
longitudF[1]	float	-99.1514282	latitudF[1]	float	19.4702663
longitudF[2]	float	-99.1514282	latitudF[2]	float	19.4702663
longitudF[3]	float	-99.1514282	latitudF[3]	float	19.4702663
longitudF[4]	float	-99.1514282	latitudF[4]	float	19.4702663
longitudF[5]	float	-99.1514282	latitudF[5]	float	19.4702663
longitudF[6]	float	-99.1514282	latitudF[6]	float	19.4702663
longitudF[7]	float	-99.1514282	latitudF[7]	float	19.4702644
longitudF[8]	float	-99.1514282	latitudF[8]	float	19.4702606
longitudF[9]	float	-99.1514282	latitudF[9]	float	19.4702606
longitudF[10]	float	-99.1514359	latitudF[10]	float	19.4702606
longitudF[11]	float	-99.1514359	latitudF[11]	float	19.4702606
longitudF[12]	float	-99.1514511	latitudF[12]	float	19.4702606
longitudF[13]	float	-99.1514511	latitudF[13]	float	19.4702568
longitudF[14]	float	-99.1514664	latitudF[14]	float	19.4702568
longitudF[15]	float	-99.1514664	latitudF[15]	float	19.4702568
longitudF[16]	float	-99.1514664	latitudF[16]	float	19.4702549
longitudF[17]	float	-99.1514664	latitudF[17]	float	19.4702549
longitudF[18]	float	-99.1514816	latitudF[18]	float	19.4702549
longitudF[19]	float	-99.1514816	latitudF[19]	float	19.4702549
longitudF[20]	float	-99.1514816	latitudF[20]	float	19.4702549

Figura 114. Mediciones de posición realizadas con el GNSS.

Una vez validados los mensajes, se decodifican los bytes enviados obtenidos a través de UART y, estos se almacenan en arreglos con la finalidad de observar los resultados en cada medición realizada, las cuales se muestran en la Figura 114.

Con los datos anteriores, se puede verificar por medio de la aplicación de Google maps, si la posesión obtenida es la indicada. Esto se puede verificar a su vez con permitir la ubicación en el dispositivo móvil utilizado.

En este caso, en la Figura 115 se observan dos puntos, uno azul (posición del ordenador) y un punto rojo (posición obtenida con el módulo NEO-M9N), lo cual corrobora que la posición obtenida por el módulo es la correcta.

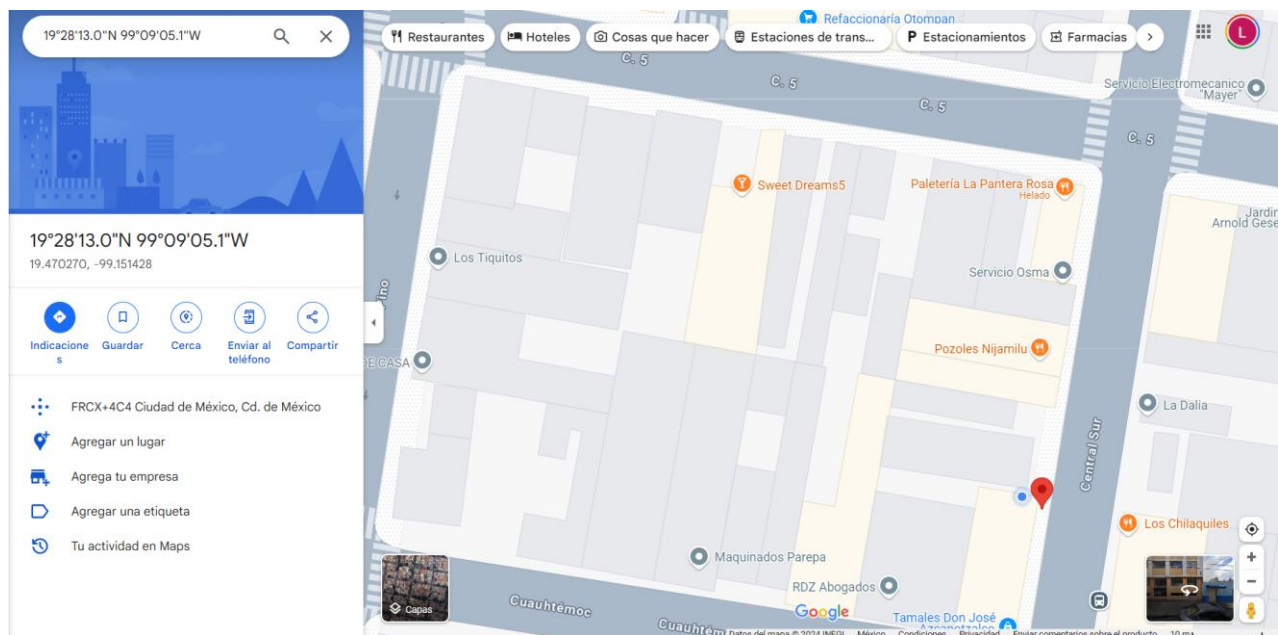


Figura 115. Verificación de la posición con uso de Google Maps

Se debe mencionar que, para obtener correctamente la posición con este dispositivo, se deben considerar los siguientes puntos:

1. Configurar los valores geodésicos de usuario, de acuerdo con el marco de referencia geodésico usado en el país.
2. Utilizar el GNSS SBAS y desactivar los satélites que no pertenezcan al sistema de navegación WAAS, utilizado comúnmente en Norte América.
3. Cuando el dispositivo obtiene una buena conexión, este comenzará a emitir una luz intermitente de color verde.

5.5 Resultados del subsistema de comunicación

La selección del subsistema de comunicación se explica en el capítulo 4.3.4 y, los resultados obtenidos se enlistan a continuación.

1. Generación de drivers para comunicarse y configurar el sistema de comunicación con la computadora de abordo.
2. Envío de telemetría de la computadora de abordo por medio del subsistema de comunicación, hacia un dispositivo móvil o computadora conectados a internet.

En la comunicación entre dos dispositivos remotos se implementó el uso del protocolo MQTT, el cual es muy común en el uso de la mensajería de telemetría para aplicaciones de IoT o internet de las cosas.

Como primer punto se verificó el módulo SIMCOM A7670SA, observando el nivel de señal de la antena y la conexión a la red móvil; en caso de que el dispositivo no se encuentre conectado a la red, se deberá conectar por medio de comandos AT. Después de asegurar la conexión, se deben activar los servicios para el protocolo MQTT y, conectarse a un servidor donde se enviará información. Entonces, las respuestas obtenidas del dispositivo se muestran más adelante en la Figura 119.

Para poder comunicar dos dispositivos de manera remota, se debe proceder con la generación o utilización de algún servidor donde se estará, publicando los mensajes por medio de este protocolo. En este caso para facilitar la creación del servidor, se utilizó la página denominada “MQTT ONE” (Figura 116) el cual, al registrarnos nos genera un servidor por defecto donde podremos conectarnos como cliente.

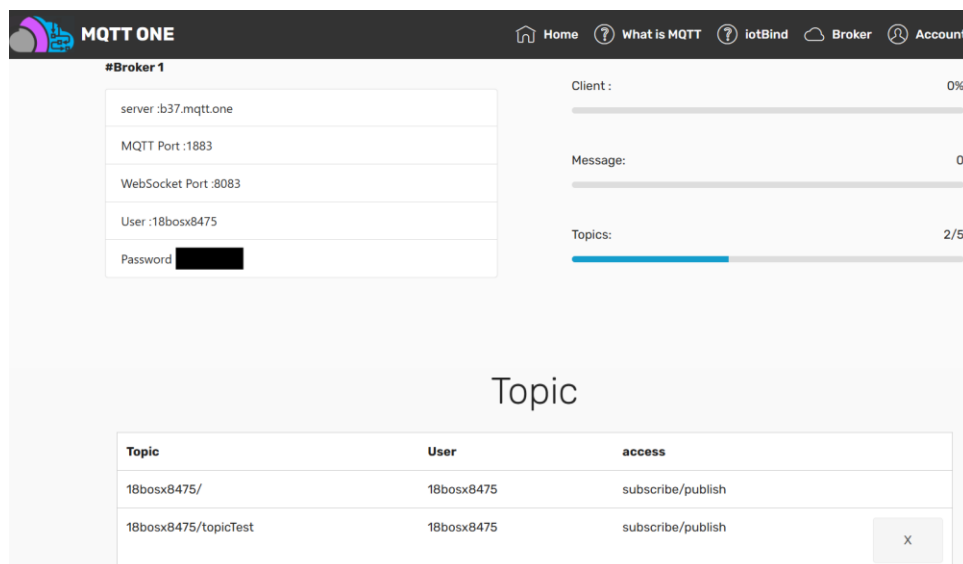
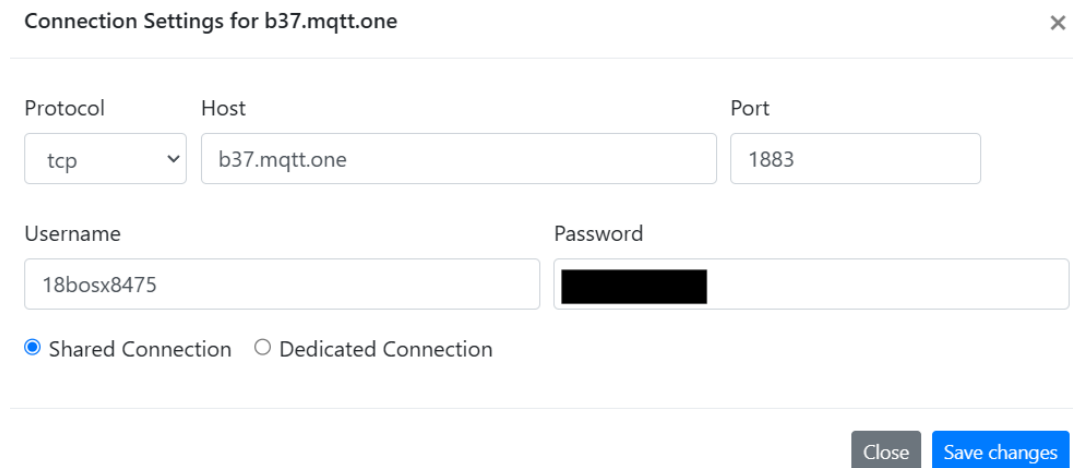


Figura 116. Servidor MQTT generado con MQTT ONE

Con el servidor funcionando, se puede conectar cualquier cliente con los datos solicitados por este mismo. Por tanto, ahora se utilizará una de las muchas páginas que operan como cliente en el protocolo MQTT, el cual se conectará al servidor anterior (Figura 117).



Connection Settings for b37.mqtt.one

Protocol: tcp Host: b37.mqtt.one Port: 1883

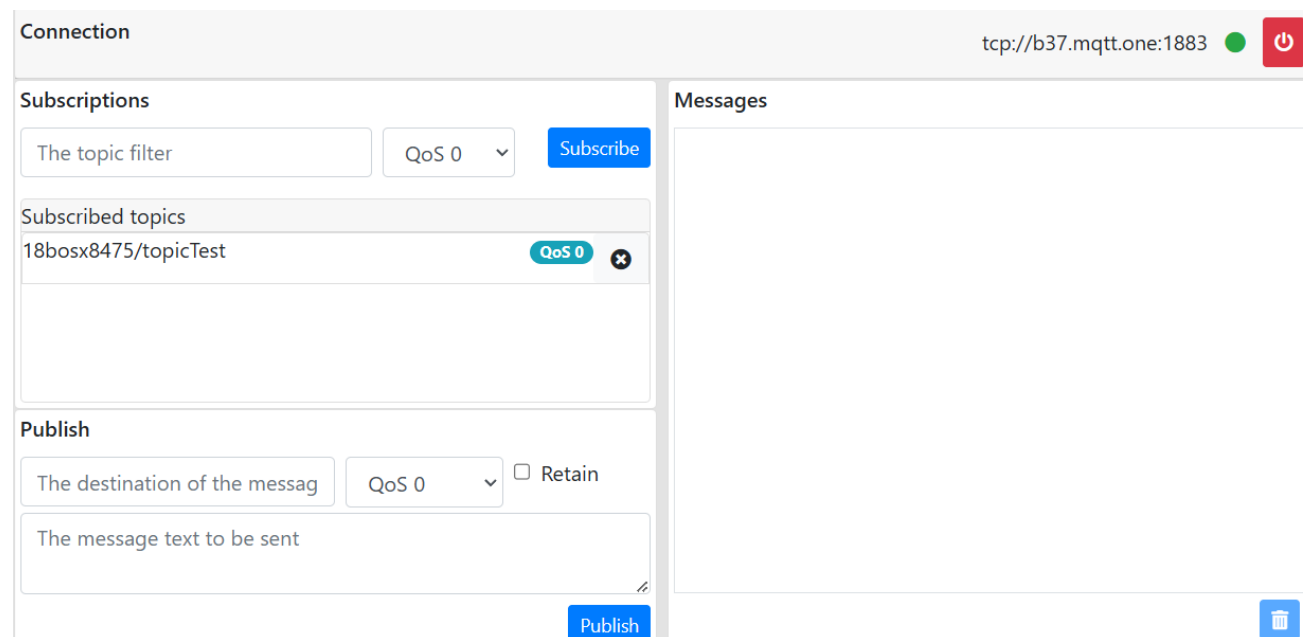
Username: 18bosx8475 Password: [REDACTED]

☒ Shared Connection ☐ Dedicated Connection

Close Save changes

Figura 117. Ejemplo de conexión al servidor MQTT

Y, una vez conectados, nos suscribiremos al tópico donde realizaremos las pruebas (Figura 118).



Connection: tcp://b37.mqtt.one:1883

Subscriptions

The topic filter QoS 0 Subscribe

Subscribed topics

18bosx8475/topicTest QoS 0

Publish

The destination of the message QoS 0 Retain

The message text to be sent Publish

Messages

Figura 118. Ventana mostrando la conexión al servidor MQTT y la suscripción al tópico.


```
Console Problems Executables Memory Search SWV ITM Data Console ×
Port 0 ×
OK
+CSQ: 27,99 Nivel de señal
OK
+CREG: 0,1
OK
+CGREG: 0,1 Registro en la red celular
OK
+CPSI: LTE,Online,334-20,0x20A,██████████,381,EUTRAN-BAND66,67036,5,0,25,54,52,18
OK
+CGDCONT: 1,"IP","APN","",0,0,,, Estado del sistema y red
+CGDCONT: 8,"IPV4V6","IMS","",0,0,0,2,1,1 celular
OK
+CGACT: 1,1
+CGACT: 8,0
OK
INCORPORATED
OK
OK
+CMQTTSTART: 0
+CGEV: ME PDN DEACT 8 Configuración MQTT y
OK conexión a un servidor
OK MQTT
+CMQTTCONNECT: 0,0
+CMQTTCONNECT: 0,"tcp://b37.mqtt.one:1883",60,1,"18bosx8475",██████████
+CMQTTCONNECT: 1
OK
OK
+CMQTTSUB: 0,0 Suscripción a un tópico
```

Figura 119. Respuestas obtenidas por el módulo SIMCOM.

Retomando la configuración de nuestro módulo SIMCOM, después de verificar la conexión del dispositivo y haber creado un servidor MQTT, podemos conectar nuestro dispositivo a este servidor, donde las respuestas de que se estableció una correcta conexión con este se muestran en la Figura 119.

Con todo lo anterior, se facilitará la observación de la información publicada por el dispositivo cuando este lo realice. Por tanto, para esta prueba, se hará envío de telemetría de la simulación del subsistema de potencia utilizada para probar el dispositivo bluetooth justo como en el capítulo 5.2.

En la Figura 120 se observan los mensajes que se publican del subsistema de potencia por medio del módulo SIMCOM; en este caso, los datos que se enviaron fueron el estado y la corriente de cada uno de los reguladores simulados, así como el consumo total de corriente y el porcentaje de carga en la batería. Un punto por observar es que en uno de los mensajes se encierra en un cuadro rojo información referente a la fecha y hora en que se suscribió el mensaje, en donde si comparamos los demás mensajes, notaremos que estos tienen diferencia de un segundo aproximadamente.

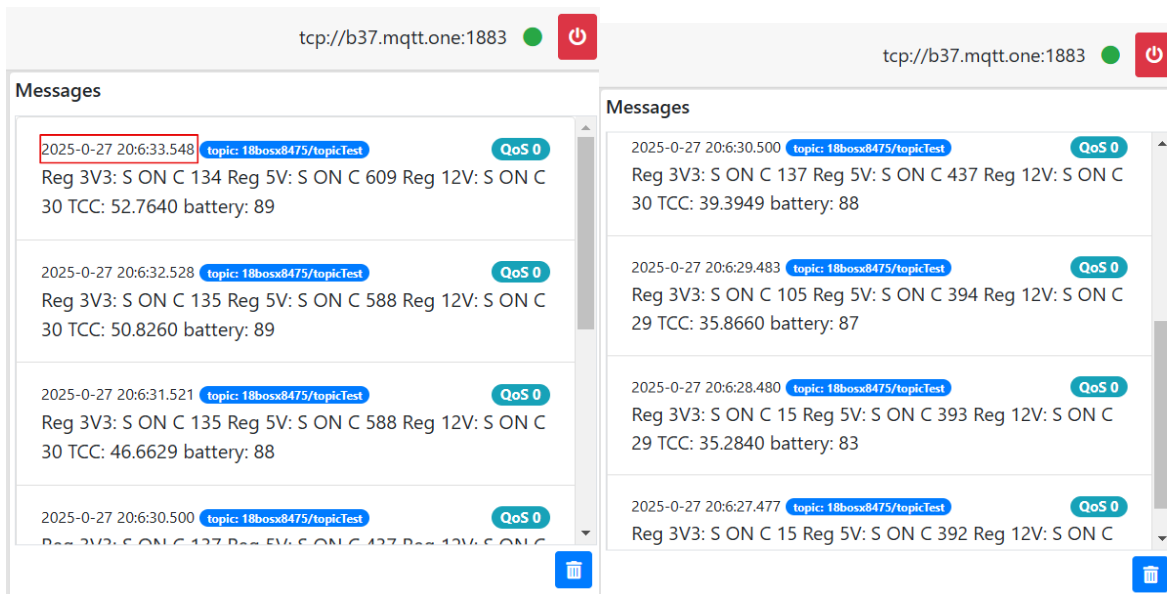


Figura 120. Mensajes publicados y visualizados con un cliente MQTT de una página web

Finalmente, en la Figura 121 se muestran dos imágenes que contienen las respuestas mandadas por el módulo al momento de enviar la telemetría y, cuando se reciben los mensajes al tópico que nos suscribimos. Es decir, primero realizamos la parte de la publicación de la telemetría en el tópico que deseamos y, el módulo responde con varios “OK” y, cuando el mensaje se publica en este tópico, el módulo lo detecta y envía los datos del mensaje recién publicado al tópico, lo cual se observa en las cadenas de datos de texto, o, dicho de otra forma, escribimos el mensaje y leemos ese mismo mensaje en el servidor.

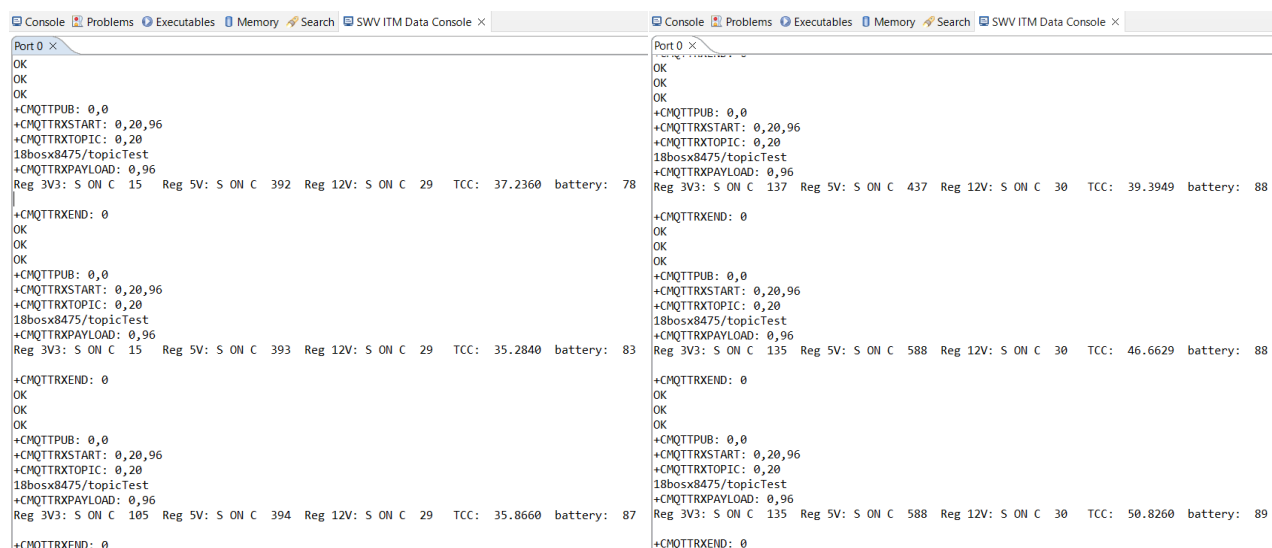


Figura 121. Mensajes publicados obtenidos con el módulo SIMCOM

5.6 Resumen de los resultados

Los resultados alcanzados por cada subsistema se enlistan a continuación en la siguiente Tabla 32. Tabla de resultados finales

Tabla 32. Tabla de resultados finales

Subsistema	Resultados
Subsistema de Detección	Se obtuvo un subsistema de detección de objetos capaz de proporcionar información útil para localizar objetos en el ambiente. Los sensores de movimiento inercial a distancias de 40m tienen un error máximo de .22m aproximadamente, lo cual, es poco con respecto al tamaño del haz de Lidar, que a esta distancia corresponde a 2.08m.
Unidad de control del subsistema de Potencia	Se logró proporcionar un diseño a nivel sistema del subsistema de potencia, donde, se probaron y verificaron los algoritmos requeridos para el monitoreo y control del subsistema. Estos algoritmos incluyen las siguientes funcionalidades: • Monitoreo de señales por medio de canales ADC para el cálculo de la carga consumida. • Control y señales sincronización para reguladores. • Envío de telemetría por CAN o Bluetooth. • Configuración del sistema para operar en modo de bajo consumo de energía Asimismo, se simularon los reguladores de voltaje a implementar en el subsistema de potencia, donde los resultados de las simulaciones se muestran en los apéndices correspondientes. Finalmente, se generó un prototipo para las pruebas de potencia, el cual, se verificó con herramientas de verificación de la integridad de potencia.
Subsistema de Comunicación	Se seleccionó un dispositivo de comunicación compatible con la red celular de México. Para este dispositivo, se generaron drivers que lo gestionan y envían telemetría a través de la red haciendo uso del protocolo MQTT.

Subsistema de Posicionamiento	Se seleccionó un módulo GNSS con una precisión de hasta 2m, del cual se generaron drivers que implementan el protocolo UBX para su configuración y para la obtención de datos de posición en tiempo real.
Computadora de abordo	Se generó una PCB que provee diferentes interfaces físicas a la computadora de a bordo para conectarse y comunicarse con diferentes dispositivos. Se validaron y obtuvieron tiempos de ejecución para los algoritmos de control del dispositivo implementados a nivel código. En la investigación se encontró algunos drones operando a 20m/s, por lo que, si se deja un margen de 1m de distancia para procesar los datos, dado que el sensor permite medir hasta 40m de distancia, esto nos deja un tiempo de ejecución total de los algoritmos de 50ms. Entonces, los tiempos máximos de ejecución para las simulaciones obtenidas son: • DBSCAN: 700us para 22 puntos • Convex Hull: 559us para 22 puntos • Voronoi: 9.83ms 10 obstáculos • Dijkstra: 250us para 16 puntos de voronoi Estos algoritmos básicos en general tienen un tiempo de ejecución de aproximadamente 11.34ms. Esto es un tiempo que entra en el intervalo definido, pero aún faltan considerar más pruebas. Por otra parte, el algoritmo de RANSAC no puede desempeñarse por esta computadora, porque el máximo tiempo para un solo objetos es alrededor de 1.38s y, este tiempo para el dron significaría avanzar 27.6m sin haber procesado su información.

6. Conclusiones

En este trabajo se logró el diseño a nivel sistema y la instrumentación parcial de un dron de servicios de entregas de paquetes, donde, se abordó su resolución por medio de dividir el sistema completo en diferentes subsistemas (proveniente de la muy conocida expresión “Divide y vencerás”). Este sistema se completó parcialmente debido a varias limitaciones como presupuestos, tiempo y alcances del proyecto, sin embargo, se dejan las pautas para seguir con el desarrollo de la instrumentación de los subsistemas que no se pudieron probar de forma física para caracterizar y ver el desempeño completo del sistema.

Se debe mencionar que en este proyecto se hizo el análisis y diseño de un sistema a la medida, es decir, cada uno de los subsistemas diseñados operan atendiendo los requerimientos de diseño definido, esto se explica a continuación.

- La computadora proporciona un tiempo de procesamiento en tiempo real adecuado para algoritmos básicos como los descritos en este trabajo (las simulaciones se ejecutan en un tiempo menor a 50ms), sin embargo, para algoritmos más complejos como RANSAC, la computadora ya no es adecuada para usar este algoritmo en la detección de objetos en tiempo real.
- El subsistema de detección de objetos proporciona datos y una visión del ambiente de acuerdo con lo establecido, sin embargo, a largas distancias como 40m, el subsistema se limita por el sensor principal y solo puede detectar objetos con una longitud de 2.08m, esto puede ser un problema para detectar objetos pequeños desde lejos, pero en el caso de grandes edificios es suficiente.
- El subsistema de posicionamiento y de comunicación permiten correctamente la recepción de datos en la región de México y, en el caso del subsistema de comunicaciones permite correctamente el envío de datos a través de la conexión a la red. La única limitante que se encontró fue que el subsistema de posicionamiento opera de mejor forma en lugares abiertos y no se recomienda usar en lugares cerrados.
- Del subsistema de potencia la unidad de control efectúa de forma correcta en la parte de control con los algoritmos diseñados, sin embargo, presenta algunas limitaciones como la resolución del ADC que es de 12 bits y las señales de sincronización subamortiguadas que produce al generar señales PWM con frecuencias mayores a 500KHz.

Parte del propósito del trabajo es tratar de reducir costos e identificar cuáles son los dispositivos que mejor se adaptan al proyecto para operar de manera eficiente. La única desventaja de realizar esto es en cuestión de tiempo e investigación, esto porque por un lado se buscan las mejores ofertas y los dispositivos más eficientes, capaces de

desempeñar los objetivos deseados; y, por otro lado, se deben validar y caracterizar el desempeño de cada uno de estos.

Finalmente, este proyecto deja las bases y lineamientos para el diseño de un sistema que no sólo se puede aplicar para un dron repartidor, sino que también puede servir para ayudar en el análisis para la selección de dispositivos electrónicos y en la generación de computadoras de abordo que se requieran diseñar con diferentes aplicaciones a las descritas.

7. Trabajo a Futuro

En la tesis presente se expuso el diseño del sistema de un dron repartidor, el cual, se abordó dividiéndolo en diferentes subsistemas, en donde se cumplió el alcancé definido al principio del trabajo.

El trabajo a futuro que se requiere como una etapa de conclusión del proyecto se presenta en la Tabla 33.

Tabla 33. Trabajo a futuro

Subsistema	Trabajo a futuro
Subsistema de Detección	Quedaron pendientes dos tareas fundamentales: 1. Integración del software para el manejo y envío de datos por medio de USB. 2. Determinación de la orientación con los sensores inerciales cuando el dron se encuentre en movimiento.
Subsistema de Potencia	Queda pendiente mandar a elaborar la PCB y probar el funcionamiento y desempeño del hardware propuesto para desempeñar la parte de distribución de energía, regulación de voltaje, medición de energía y control en la distribución de la energía. Asimismo, falta la caracterización de señales para implementar correctamente los filtros y ver el desempeño de otros dispositivos con este mismo.
Subsistema de Comunicación	Queda pendiente analizar el uso de otros alternativos protocolos cliente-servidor para posibles aplicaciones con el sistema completo.
Subsistema de Posicionamiento	Queda pendiente probar el GNSS operando en movimiento.
Computadora de abordó	Con la computadora de abordó, queda pendiente el desarrollo de una aplicación que englobe y controle a cada uno de los subsistemas y algoritmos presentados en este proyecto. Además,

	falta probar su desempeño junto al subsistema de potencia. Por último, falta incorporar el manejo y comunicación con el controlador de vuelo seleccionado.
Operación del sistema completo	Falta probar en conjunto todo el sistema para verificar su desempeño y hacer análisis que permitan mejorar la eficiencia del sistema.

Por otra parte, si bien es cierto la visión artificial es una de las cosas indispensables en los robots autónomos hoy en día, esta parte en cierta manera se puede cubrir con LIDAR, sin embargo, hay otros aspectos en el uso de la cámara que proporcionan una mayor información del entorno como la identificación del tipo de objeto, lo cual, en nuestra actualidad y contexto se vuelve un tanto trivial, pero no se descarta, esto debido a que a diferencia de trabajar con autos autónomos, los drones operan en el cielo y los objetos a identificar son menores como los cables, edificios, arboles, aves, por mencionar algunos. Es cierto que las cámaras otorgan mucha información del entorno, no obstante, hacer únicamente el uso de estos dispositivos conlleva a requerir un mayor procesamiento de datos y, por tanto, una unidad capaz de proveer este procesamiento en un tiempo requerido, a diferencia de Lidar que detecta lo indispensable de un objeto, es decir, su ubicación y forma, requiriendo por consiguiente un menor nivel de procesamiento.

Una posible aproximación para mejorar el desempeño del dron repartidor sería integrar Lidar con una cámara, tratando de mejorar la visión de este sistema. El sentido de hacer esto es poder identificar ciertos objetivos (además de localizar obstáculos) al momento de llegar al destino, por ejemplo, mientras que Lidar se basa en la localización y evasión de obstáculos, la cámara se puede centrar en la identificación de personas o lugares hacia donde se destina la entrega, lo cual, quitará la carga de operaciones computacionales de las imágenes obtenidas y, a su vez, propiciará una mayor seguridad en la entrega del paquete.

Finalmente, otro aspecto interesante a tratar es la implementación del uso de inteligencia artificial para operar en el sistema del dron repartidor. Lo cual, podría ayudar a resolver problemas críticos en tiempo de vuelo, sin embargo, en cuestiones de consumo y de eficiencia de la energía es donde se podrá ver reducido el tiempo de vuelo.

Apéndices

APÉNDICE I. CÁLCULO, SELECCIÓN DE LOS ELEMENTOS Y SIMULACIÓN DEL REGULADOR LTC7821

En este apéndice se describe el procedimiento llevado a cabo en el diseño de un regulador de voltaje de 22.2[V] a 5[V] basado en circuito integrado LTC7821, de acuerdo con su hoja de datos referenciada en [58]. Entonces, primero se definieron parámetros como la corriente de operación (I_{out}), el porcentaje de rizo de corriente ($\% \Delta I_{out}$) y la frecuencia de conmutación del dispositivo (f_{sw}).

$$I_{out} = 10[A], \% \Delta I_{out} = 40\% \text{ y } f_{sw} = 500[kHz]$$

Además, el ciclo de trabajo implementado por medio de los transistores como interruptores de conmutación, se calcula de la siguiente forma

$$DutyCycle_{M_1, M_2} = \frac{2V_{out}}{V_{in}} = 0.45 = D$$

El rizo de corriente de acuerdo con lo definido

$$\Delta I_L = \frac{\% \Delta I_{out}}{100} I_{out} = 4$$

Por tanto, la inductancia

$$L = \frac{(1 - D)V_{out}}{f_{sw} \Delta I_L} = 1.38[\mu H]$$

En este caso se usará un inductor con inductancia de 1.5[μH], dando un rizo de corriente de $\Delta I_L \cong 3.65$ (revisar pie de página 32).

La corriente RMS en el inductor

$$I_{RMS} \sqrt{I_{out}^2 + \frac{\Delta I_L^2}{12}} = 10.06[A]$$

Para el inductor además se deben considerar el parámetro DCR²⁷, en este caso, el DCR típico del inductor escogido es de 2.9mΩ y, asumiendo una temperatura de 50° con un coeficiente de temperatura del DCR a 0.4%/°C, nos da un máximo DCR.

$$DCR_{\Delta} = 2.9\text{m}\Omega \left(1 + \frac{\left(\frac{.4}{100}\right) 50^{\circ}\text{C}}{100} \right) = 3.48\text{m}\Omega$$

Y, de acuerdo con la hoja de datos, para un valor $V_{sense} = 50\text{mV}$, el DCR establecerá la corriente pico del inductor a un valor de $I_{Lmax} \cong 15.5[\text{A}]$.

Para la red de detección²⁸ se escoge un capacitor de 0.22μF, para tener un valor de resistencia cercano a los 2 kΩ de acuerdo con sus recomendaciones.

$$R_8 = \frac{L}{(DCR)(C_1)} = 2.35\text{k}\Omega$$

Por otra parte, están los capacitores C_{FLY} y C_{MID} que se cargan a $V_{MID} = V_{IN}/2$, en este caso se recomienda que tengan el mismo valor, para disminuir las pérdidas en los MOSFET y, se seleccionan para un rizo menor al 1% de $V_{IN}/2$.

$$C_{FLY} = C_{MID} = \frac{I_{out}D}{2f_{sw}\Delta V_{MID}} = 40.6\mu\text{F}$$

Y, debido a ciertas consideraciones al seleccionar capacitores como la respuesta de la impedancia en los capacitores y el coeficiente de voltaje, que hacen disminuir el valor de la capacitancia en el capacitor, se selecciona un valor mayor de capacitancia, siendo este,

$$C_{FLY} = C_{MID} = 10 \times 10\mu\text{F}$$

Esta selección influirá en el cambio del rizo de voltaje, por lo cual, se deberá recalcular este parámetro con los capacitores nuevos, dando un rizo de voltaje con valor de,

$$\Delta V_{out} \cong 45\text{mV}$$

²⁷ DCR, por sus siglas en inglés, se refiere a la resistencia en serie de DC del inductor.

²⁸ La red de detección consta de un método implementado en el dispositivo y se usa para medir específicamente la corriente de salida generada en el inductor y así, mantener una corriente promedio a la salida.

Otro aspecto importante del regulador es escoger los MOSFET adecuados, debido a que su selección influye en la eficiencia del regulador. Estos puntos se enlistan para facilitar la redacción:

- En el regulador cargan su puerta VGS desde 0[V] hasta 6[V], de acuerdo con el datasheet LTC7821.
- El MOSFET M1 necesita tener un valor BVDSS mayor a VIN, mientras que los demás MOSFET pueden tener un valor BVDSS mayor a VIN/2.
- M1 y M2 deben tener el menor producto de (Qoss • RdsON) y, M3 y M4 deben ser seleccionados con el menor producto de (Qgd • RdsON) y (Qg • RdsON), respectivamente. Donde estos datos son obtenidos directamente del datasheet de cada MOSFET.

Con lo anterior en mente, los MOSFET seleccionados se muestran en la Tabla 34 con sus parámetros respectivos.

Tabla 34. Parámetros de los MOSFET seleccionados.²⁹

MOSFET	Dispositivo	VDSS	RDSON	ID continua	ID pulso	Qg	Qgd
M1	BSC0906NS	30[V]	5.1[mΩ]	63[A]	Apróx. 110[A]	6.7[nC]	2.2[nC]
M2 y M3	BSZ031NE2LS5ATMA1	25[V]	3.2[mΩ]	80[A]	Apróx. 120[A]	6.3[nC]	1.4[nC]
M4	BSC015NE2LS5IATMA1	25[V]	1.7[mΩ]	147[A]	Apróx. 150[A]	10.5[nC]	2.3[nC]

El MOSFET M1 con un valor de $Q_G = 6.7[nC]$. Por tanto, la capacitancia equivalente en la puerta se define como

$$C_G = \frac{Q_G}{V_{GS}} = 1.12[nF]$$

Con este valor se realiza el cálculo para los capacitores CBST1 a CBST3, considerando que $CBST3 \geq 2CBST2 \geq 2CBST1$

Por tanto,

$$C_{BST1} = 99C_G = 0.116\mu F$$

²⁹ Los parámetros mostrados se obtuvieron de las hojas de datos para valores VGS = 4.5[V] a 5[V]. En el caso de pulso de corriente en el “drenaje”, es usó el diagrama 3, de acuerdo con los datasheet de los MOSFET, para una duración de 10us.

Se escogen los capacitores,

$$C_{BST1} = 0.12\mu F, C_{BST2} = 0.22\mu F, C_{BST3} = 0.47\mu F$$

Para los diodos se recomiendan escoger diodos con una corriente directa mayor a los 0.75[A]³⁰ y capaz de soportar el voltaje de inversa máximo necesario ($V_{IN}/2$). Para ellos se escoge los diodos RB168LAM-60.

Posteriormente, se calcula la caída de voltaje obtenida en V_{MID} , la cual, idealmente debe ser aproximadamente $V_{IN}/2$.

$$Z_{MID} = \frac{\coth\left(\frac{D_1 T_s}{2\tau_1}\right) + \coth\left(\frac{D_2 T_s}{2\tau_2}\right)}{8C_{FLY}f_{sw}} = 11.7m\Omega$$

Donde lo parámetros no mencionados se definen en la página 26 del datasheet. Por consiguiente, la caída de tensión en V_{MID} es aproximadamente,

$$V_{MIDmin} = \frac{V_{in}}{2} - (I_{Lmax} * Z_{MID}) - \Delta V_{MID} = 226mV$$

Este valor nos permitirá seleccionar un valor adecuado en el pin $HYST_PRGM$ para corregir la caída de tensión respecto a los 11.1[V] deseados. Para ello se coloca una resistencia a tierra en esta terminal, por la cual fluyen 10[μA].

$$R_{HYST_PRGM} = \frac{226mV}{10\mu A} = 22.6k\Omega$$

Otro punto, es escoger un divisor para generar un voltaje de referencia y lograr así regular el voltaje a la salida. La configuración se puede realizar de dos maneras, pero las más común es conectar los pines $INTVCC$ con el pin EXT_REF , permitiendo utilizar el voltaje de referencia interno de 0.8[V]. Entonces, el divisor requerido en el pin de retroalimentación se define de la siguiente manera,

³⁰ Como estos diodos operarán con una gran polarización inversa para aplicaciones de alto V_{IN} , las fugas de corriente son mayores, especialmente a temperaturas de funcionamiento más altas. Por esa razón, se sugieren un valor de 0.75[mA] para empezar.

$$0.8 = \frac{R_2}{R_1 + R_2} V_{out}$$

$$R_2 = \frac{0.8R_1}{V_{out} - 0.8} = 0.1904R_1$$

Para un valor de $R_1 = 100k\Omega$

$$R_2 = .1904(100k\Omega) = 19.04k\Omega$$

Asimismo, para fijar la frecuencia se utiliza un resistor de $68k\Omega$, operando a una frecuencia nominal de 500kHz. Y, como punto final, se presenta la Tabla 35 con los valores seleccionados para los elementos del regulador.

Tabla 35. Elementos seleccionados para el regulador LTC7821³¹

Elemento	Producto	Características
L³²	DR125-1R5-R	$1.5\mu H \pm 20\%$, $I_{max} = 13.8[A]$, $I_{sat} = 18.3[A]$, $2.9m\Omega$
R1	RC1206FR-13100KL	$100k\Omega \pm 1\%$, $1/4W$
R2	RK73H2ATTD1912F	$19.1k\Omega \pm 1\%$, $1/4W$
R_{FREQ}	RC1206FR-0768KL	$68k\Omega \pm 1\%$, $1/4W$
M1	BSC0906NS	Ver Tabla ¿?
M2 y M3	BSZ031NE2LS5ATMA1	Ver Tabla ¿?
M4	BSC015NE2LS5IATMA1	Ver Tabla ¿?
CBST1	08055C124KAT2A	$0.12\mu F \pm 10\%$, 50V
CBST2	GRM188R71E224KA88D	$0.22\mu F \pm 10\%$, 25V
CBST3	GRM188R71E474KA12D	$0.47\mu F \pm 10\%$, 25V
R_{HYST_PRGM}	RC1206FR-13100KL	$100k\Omega \pm 1\%$, $1/4W$
R8	RK73H2ATTD2321F	$2.32k\Omega \pm 1\%$, $1/4W$
C1	GRM188R71E224KA88D	$0.22\mu F \pm 10\%$, 25V
CFLY y CMID	CL21A106KAYNNNE	$10 \times 10\mu F \pm 10\%$, 25V
D1, D2 y D3	PMEG40T10ERX	$I_f = 1A$, $V_{rrm} = 40V$, $t_{rr} = 11.5ns$

Simulación

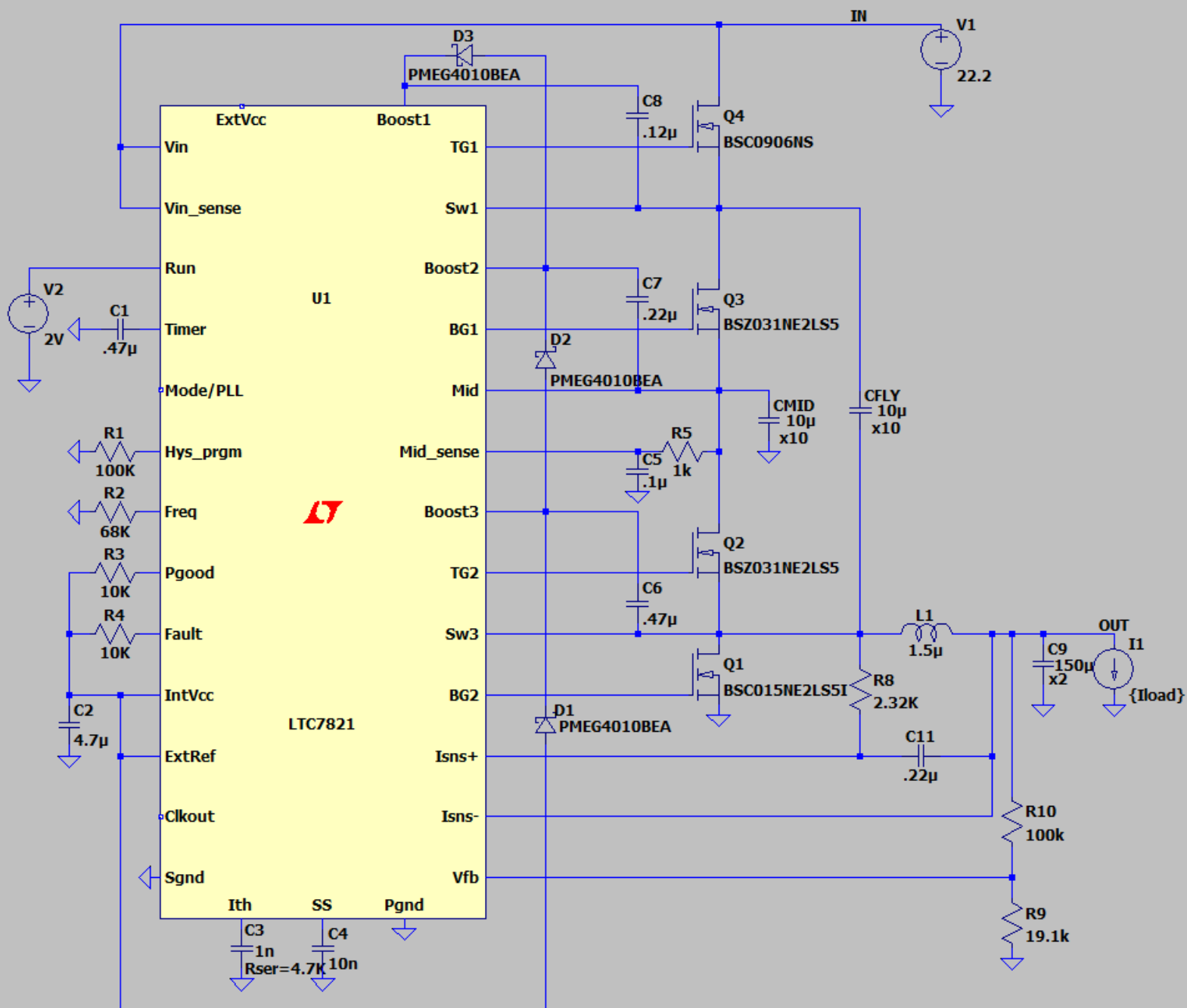


Figura 122. Esquema del circuito simulado

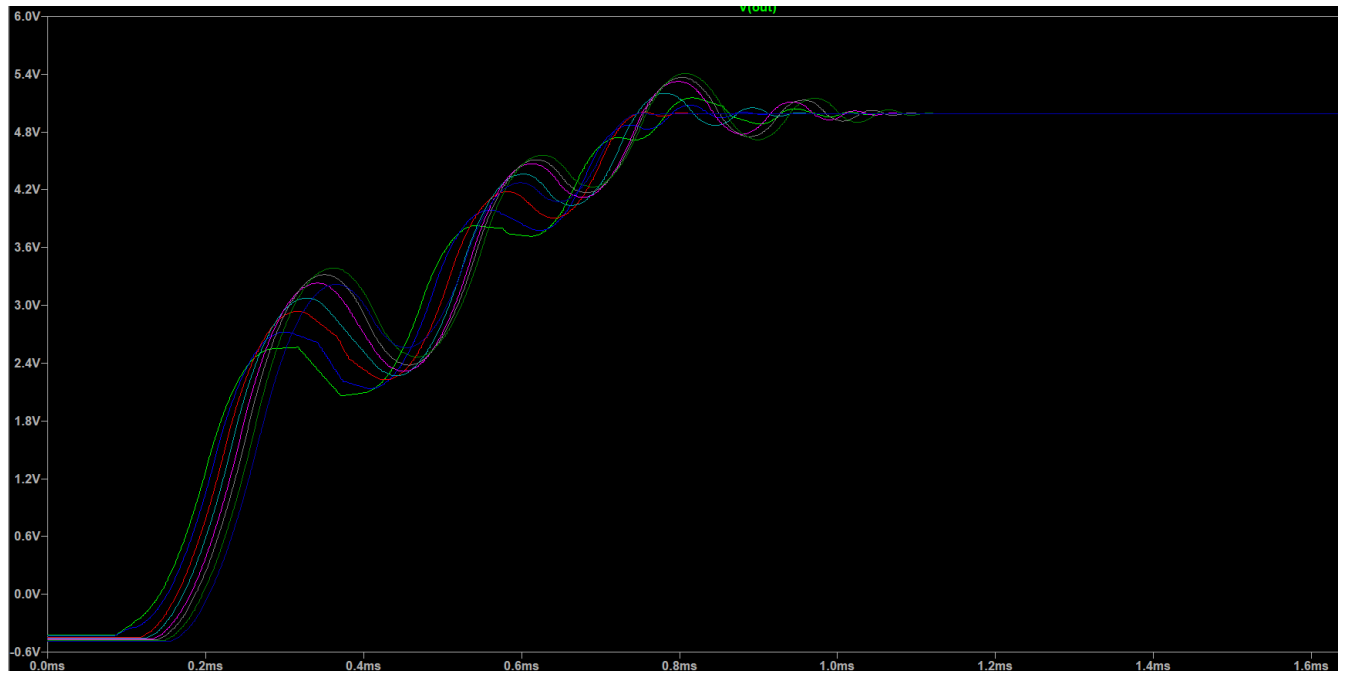


Figura 123. Voltaje de salida para diferentes cargas

SPICE Error Log: C:\Users\jurl9\OneDrive\Docum... X	
Measurement: eff	
step	pout/pin
1	0.954795
2	0.96998
3	0.969135
4	0.968194
5	0.970164
6	0.969751
7	0.96993
8	0.970815
Date: Tue Jul 23 19:16:39 2024	
Total elapsed time: 460.100 seconds.	

Figura 124. Eficiencia obtenida del circuito simulado

APÉNDICE II. CÁLCULO, SELECCIÓN DE LOS ELEMENTOS Y SIMULACIÓN DEL REGULADOR LT8609

En este apéndice se describe el procedimiento llevado a cabo en el diseño de un regulador de voltaje de 22.2[V] a 3.3[V] basado en circuito integrado LT809, de acuerdo con su hoja de datos referenciada en [59]. Primero, se definen parámetros como la corriente de operación (I_{out}) y la frecuencia de conmutación del dispositivo (f_{sw}).

$$I_{out} = 0.75[A] \text{ y } f_{sw} = 800[kHz]$$

El ciclo de trabajo para este dispositivo es,

$$D = \frac{3.3[V]}{22.2[V]} = 0.149$$

Una primera forma de aproximar el inductor utilizado en el regulador es a partir de la siguiente expresión. Y, con este valor se puede calcular el rizo para éste mismo.

$$L = \frac{V_{out} + 0.25[V]}{f_{sw}} = 4.44\mu H$$

$$\Delta I_L = \frac{V_{out}}{L f_{sw}} \left(1 - \frac{V_{out}}{V_{inmax}} \right) = 0.83[A]$$

Como se observa, para esta aplicación, el rizo resulta ser mayor al 100% de la corriente de salida. Por tanto, se adecua un valor porcentaje de rizo del 50% (ver pie de página 32) y, la inductancia será,

$$L = \frac{V_{out}}{\Delta I_L f_{sw}} \left(1 - \frac{V_{out}}{V_{inmax}} \right) = 9.8\mu H$$

Por tanto, la corriente máxima en el inductor será aproximadamente,

$$I_{outmax} = I_{LOADmax} + \frac{\Delta I_L}{2} = 1.16[A]$$

En el caso de los resistores para la retroalimentación y regulación del dispositivo, la hoja de especificaciones proporciona directamente el cálculo. Y, para un resistor $R_2 = 410k\Omega$ se obtiene,

$$R_1 = R_2 \left(\frac{V_{out}}{0.782V} - 1 \right)$$

$$R_1 = 1.32M\Omega$$

Como un punto a considerar, el fabricante recomienda utilizar valores grandes de resistencias para disminuir la corriente consumida en el dispositivo.

En el caso del capacitor de salida, el cálculo de su capacitancia se define como,

$$C_{out} = \frac{100}{V_{out}f_{sw}} = 37.9\mu F$$

Además, se puede calcular el valor mínimo de entrada para el regulador de voltaje como sigue,

$$V_{inmin} = \frac{V_{out} + 0.25}{1 - f_{sw}t_{offmin}} - 0.25 + 0.4 = 3.98[V]$$

Donde t_{offmin} se especifica en el datasheet y es de 90[ns].

Por otra parte, para la frecuencia se usará una resistencia fija con un valor de 52.3kΩ, dando los 800kHz requeridos. Asimismo, se presenta la Tabla 36 con los valores seleccionados para los elementos del regulador.

Tabla 36 Elementos del regulador LT8609³¹.

Elemento	Producto	Características
L³²	SRR1280-100M	$10\mu H \pm 20\%$, $I_{max} = 6[A]$, $I_{sat} = 6.3[A]$, $19.5m\Omega$
Cout	CL21A476MQYNNNG	$47\mu F \pm 20\%$, 6.3V
R1	ERJ-6ENF1334V	$1.33M\Omega \pm 1\%$, 1/8 W
R2	AC1206FR-07412KL	$412k\Omega \pm 1\%$, 1/4 W
RT	ERJ-6ENF5232V	$52.3k\Omega \pm 1\%$, 1/8 W

Simulación

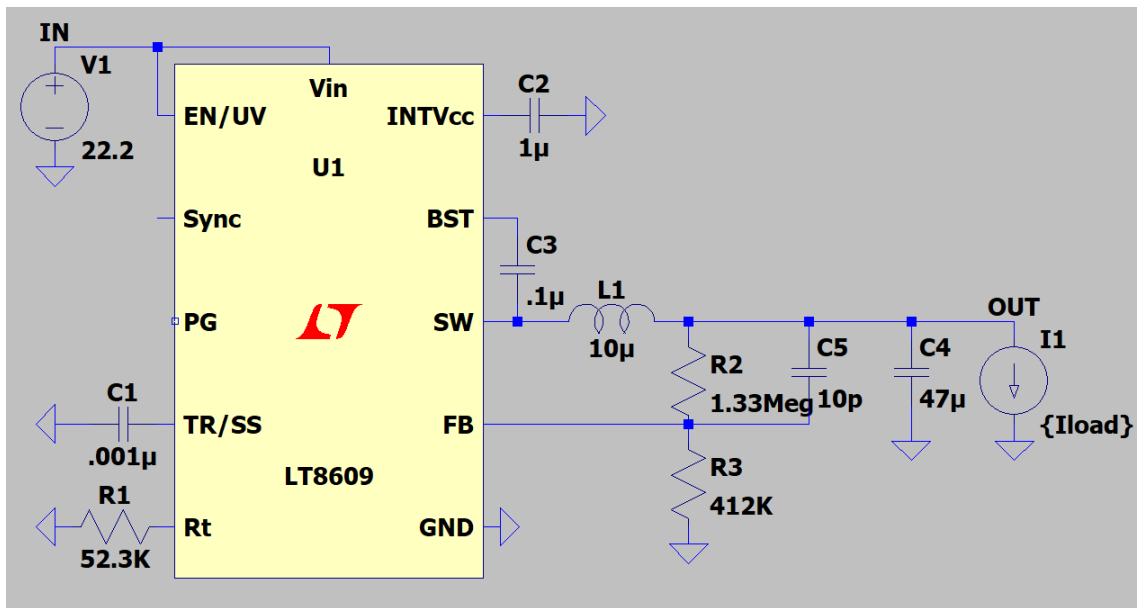


Figura 125. Esquema del circuito simulado

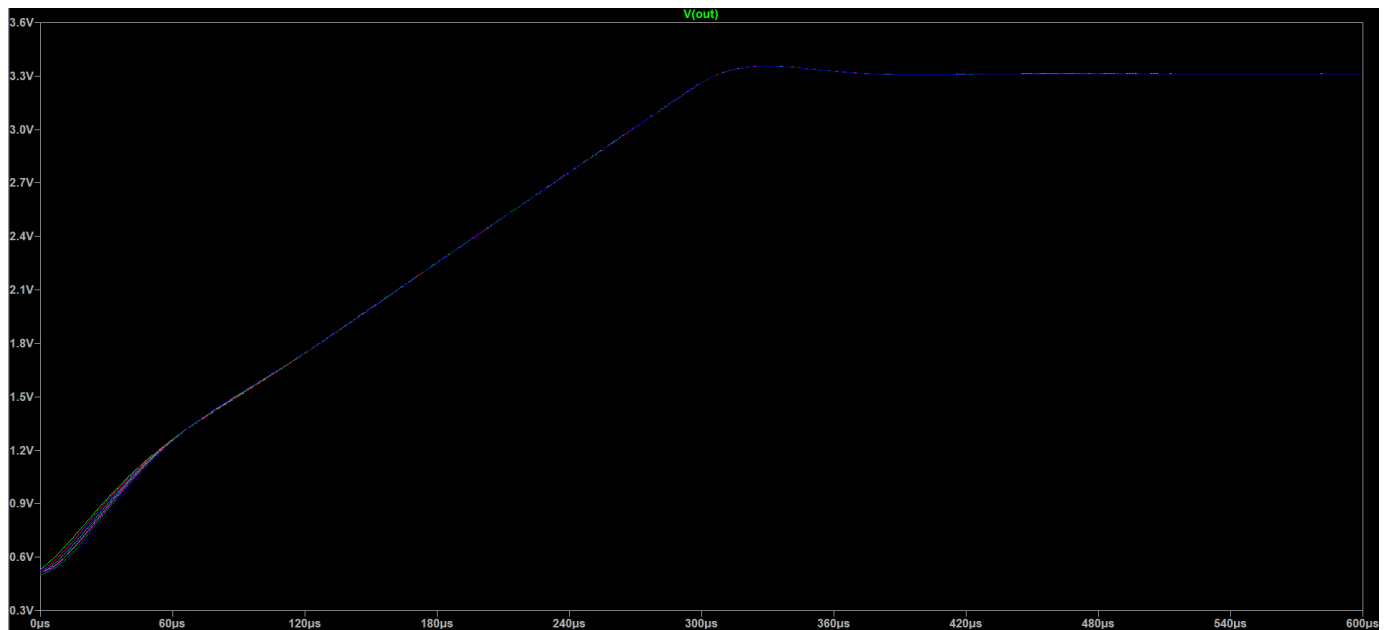


Figura 126. Voltaje de salida para diferentes cargas

APÉNDICE III. CÁLCULO, SELECCIÓN DE LOS ELEMENTOS Y SIMULACIÓN DEL REGULADOR LT8610

En este apéndice se describe el procedimiento llevado a cabo en el diseño de un regulador de voltaje de 22.2[V] a 12[V] basado en circuito integrado LT8610, de acuerdo con su hoja de datos referenciada en [60]. El cálculo para la selección de los elementos de este convertidor, se realizan como en el apéndice anterior. Entonces, se definen parámetros como la corriente de operación (I_{out}), el porcentaje de rizo de corriente ($\% \Delta I_L$) y la frecuencia de conmutación del dispositivo (f_{sw}).

$$I_{out} = 1.5[A] \text{ y } f_{sw} = 600[kHz]$$

El ciclo de trabajo para este dispositivo es

$$D = \frac{12[V]}{22.2[V]} = 0.54$$

Una primera forma de aproximar el inductor utilizado en el regulador es a partir de la siguiente expresión. Y, con este valor se puede calcular el rizo para éste mismo.

$$L = \frac{V_{out} + 0.15[V]}{f_{sw}} = 20.25\mu H$$

$$\Delta I_L = \frac{V_{out}}{L f_{sw}} \left(1 - \frac{V_{out}}{V_{inmax}} \right) = 0.6[A]$$

Como se observa, para esta aplicación, el rizo resulta ser menor al 40% de la corriente de salida. Por tanto, la corriente máxima en el inductor será aproximadamente,

$$I_{outmax} = I_{LOADmax} + \frac{\Delta I_L}{2} = 1.8[A]$$

En el caso de los resistores para la retroalimentación y regulación del dispositivo, la hoja de especificaciones proporciona directamente el cálculo. Y, para un resistor $R_2 = 402k\Omega$ se obtiene,

$$R_1 = R_2 \left(\frac{V_{out}}{0.97V} - 1 \right)$$

$$R_1 = 4.57M\Omega$$

Asimismo, el fabricante recomienda utilizar valores grandes de resistencias para disminuir la corriente consumida en el dispositivo.

En el caso del capacitor de salida se recomienda ver las aplicaciones típicas para seleccionar un valor de capacitancia adecuado.

Igualmente, se calcula el valor mínimo de entrada para el regulador de voltaje como sigue,

$$V_{inmin} = \frac{V_{out} + 0.15}{1 - f_{sw}t_{offmin}} - 0.15 + 0.3 = 12.7[V]$$

Donde t_{offmin} se especifica en el datasheet y es de 50[ns].

Por otra parte, en esta conversión es necesario establecer una inductancia mínima para evitar los sub-armónicos, debido a que el ciclo de trabajo es mayor a 0.5.

$$L_{min} = \frac{V_{in}(2D - 1)}{S_x(1 - D)} = 10.4\mu H$$

Donde S_x es una rampa de tensión alimentada dentro del amplificador de corriente cuyo valor es $2(10^5 A/s)$.

Ahora bien, para la frecuencia se usará una resistencia fija con un valor de $71.5k\Omega$, dando los 600kHz. Asimismo, se presenta la Tabla 37 con los valores seleccionados para los elementos del regulador.

Tabla 37. Elementos del regulador LT8610³¹

Elemento	Producto	Características
L^{32}	SRR1280-220M	$22\mu H \pm 20\%$, $I_{max} = 4.3[A]$, $I_{sat} = 4.1[A]$, $38.6m\Omega$
Cout	CL32A476MOJNNNE	$47\mu F \pm 20\%$, 16V
R1	CRCW08052M26FKEA	$2.26M\Omega \pm 1\%$, 1/8 W
R2	CRCW0805200KFKEAC	$200k\Omega \pm 1\%$, 1/8 W
RT	RC0805FR-0771K5L	$71.5k\Omega \pm 1\%$, 1/8 W

Simulación

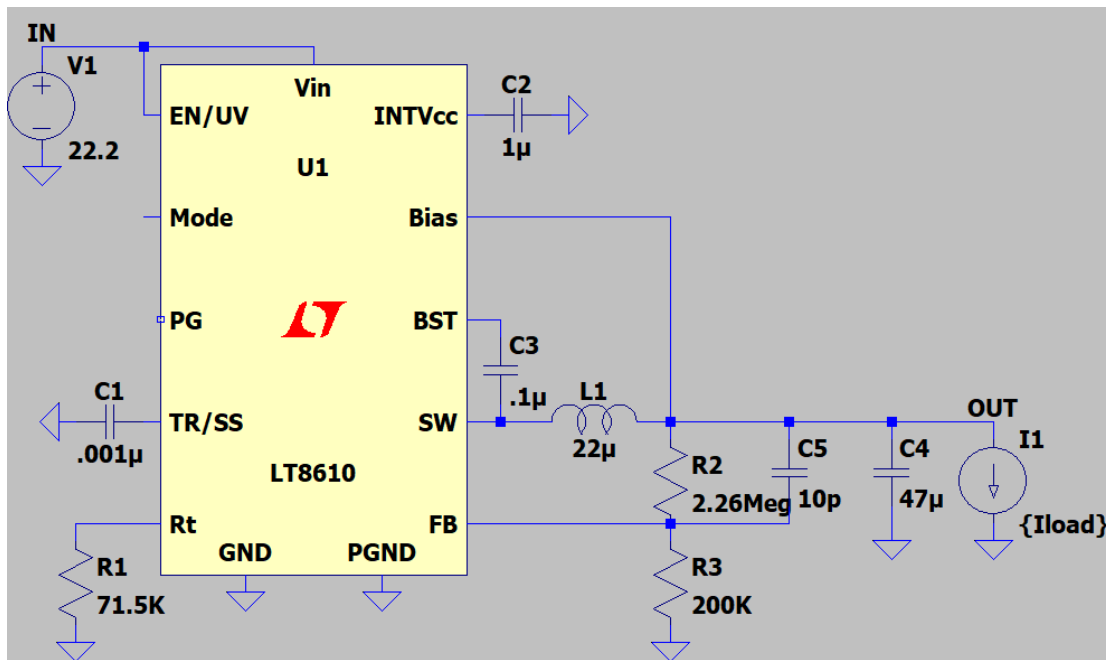


Figura 127. Esquema del circuito simulado

³¹ Debido a las frecuencias de operación de hasta 2.2MHz que permiten los reguladores IC, se recomienda checar que cada uno de los componentes utilizados en conjunto sean capaces de funcionar correctamente a esas frecuencias de conmutación, así como, tener un comportamiento en frecuencia adecuado a las necesidades del regulador.

³² La selección del valor del rizo de corriente en el inductor permite obtener valores de inductancia menores en cuanto mayor sea éste. En la teoría de los reguladores de conmutación, el valor del rizo de corriente no debe exceder el doble de la corriente de salida para no salir del modo de operación de corriente continua, sin embargo, los dispositivos implementados en cuestión permiten un modo de operación discontinua, lo cual, permite trabajar al regulador con rizos de corriente mayores a la corriente de salida, el único problema con seleccionar rizos demasiado grande es que si la carga consume muy poca corriente, la eficiencia del dispositivo se verá disminuido debido al gran rizo de corriente a la salida, por eso mismo, al momento de seleccionar los inductores, se puede considerar elegir un valor mayor al seleccionado.

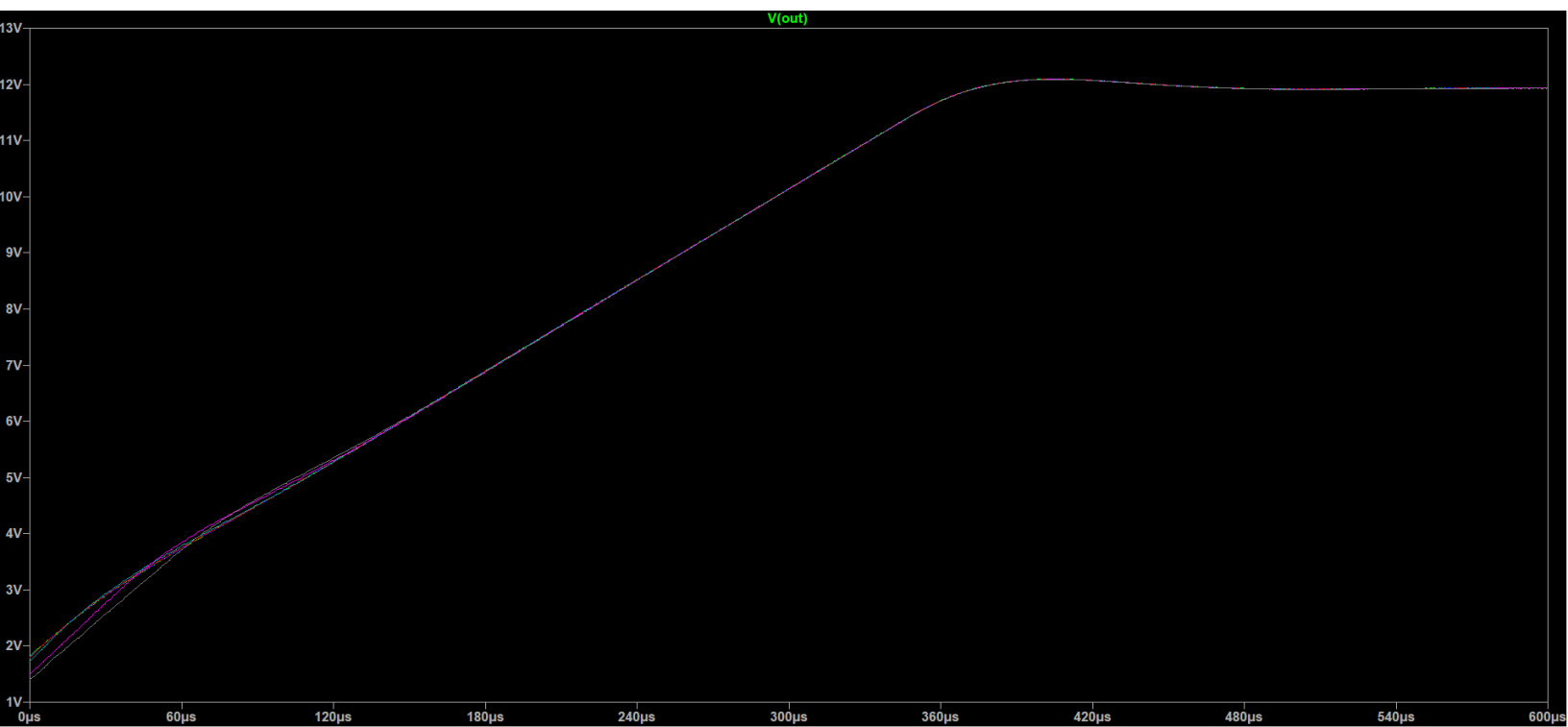


Figura 128. Voltaje de salida para diferentes cargas

APÉNDICE IV. GENERACIÓN DE LA PCB PARA LA COMPUTADORA DE ABORDO

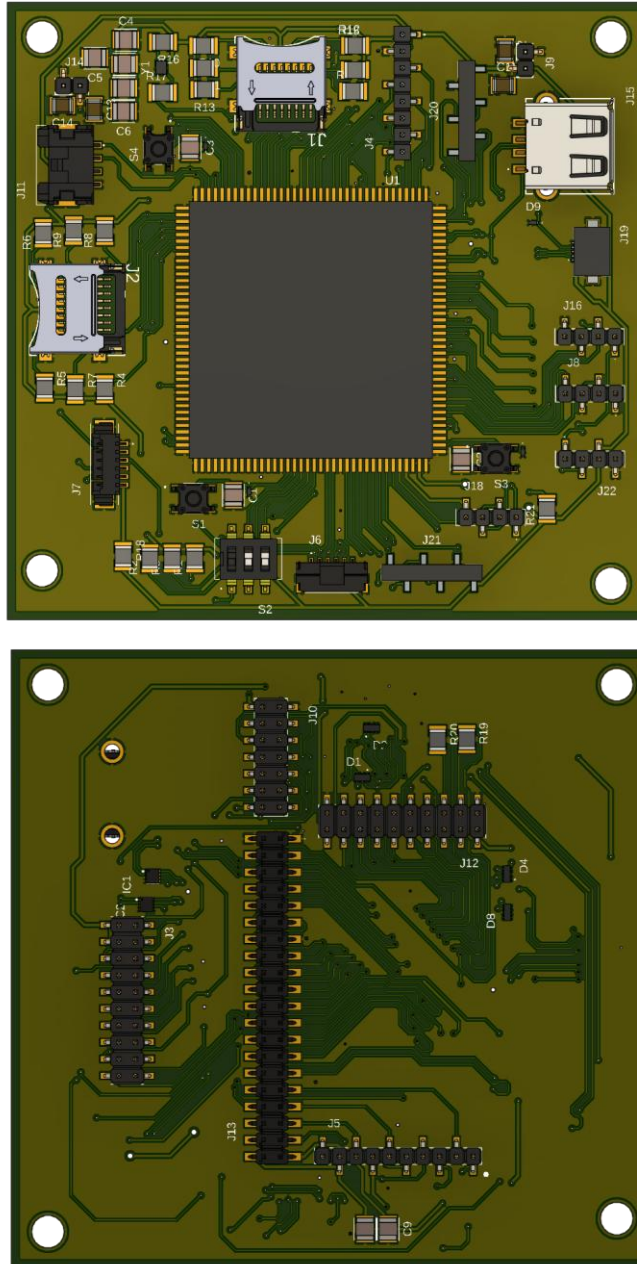
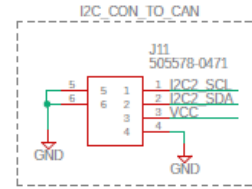
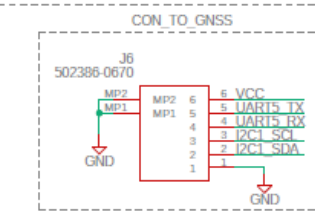
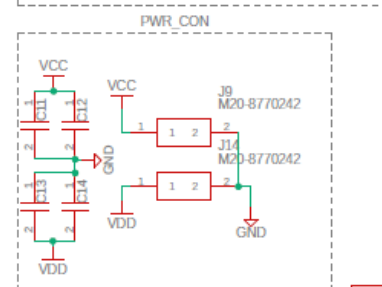
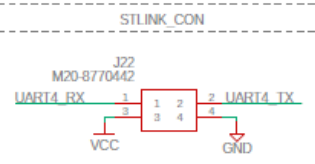
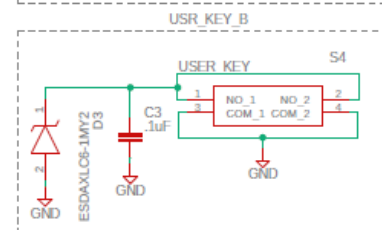
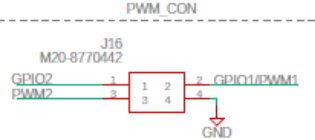
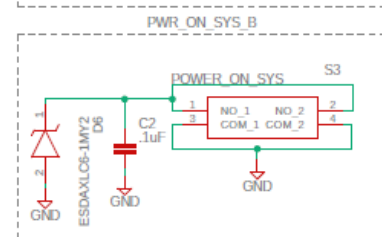
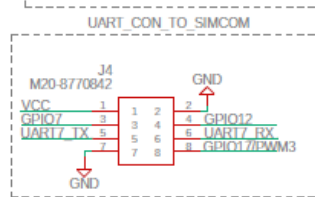
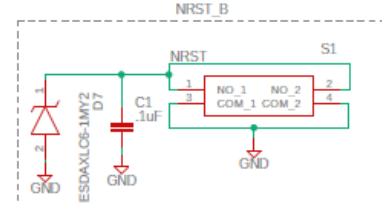
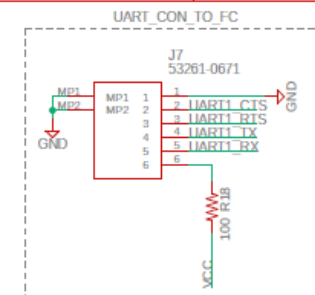
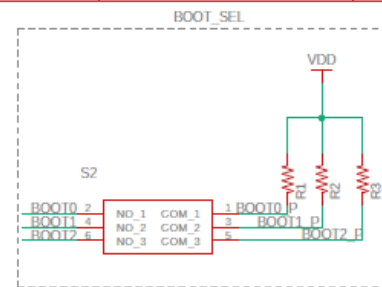
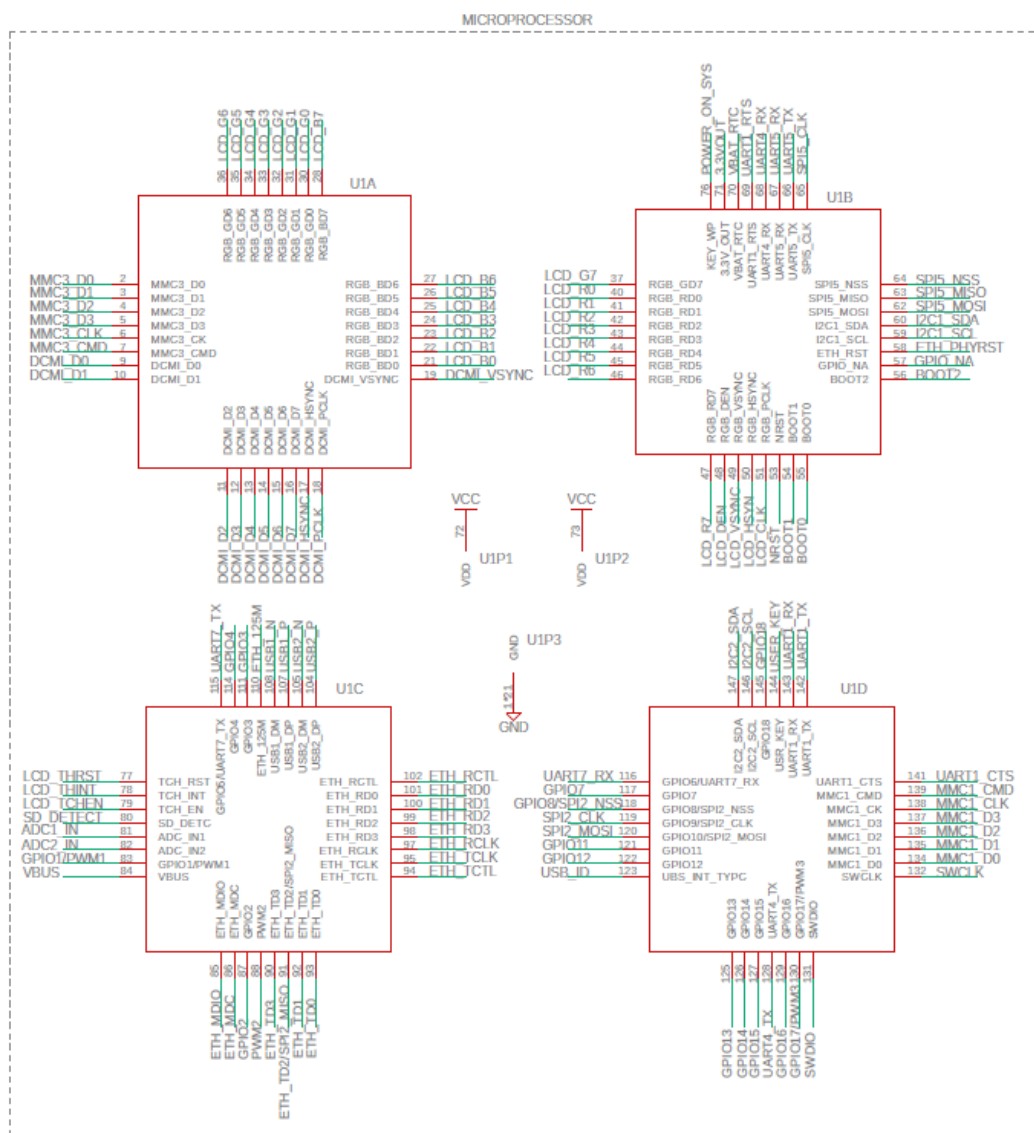
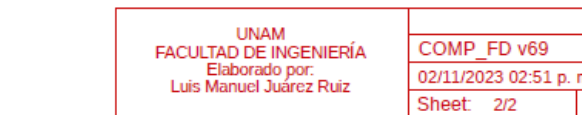
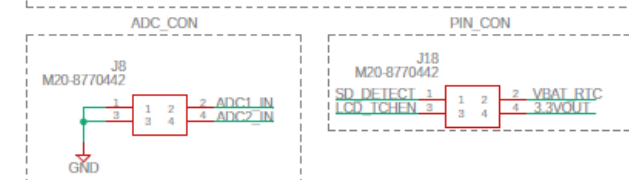
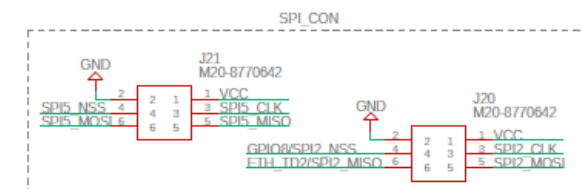
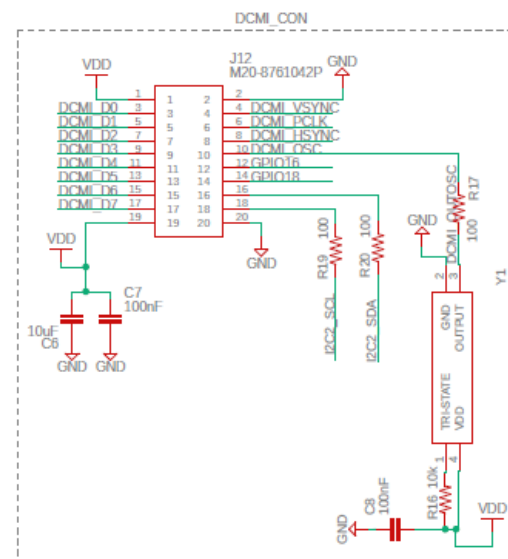
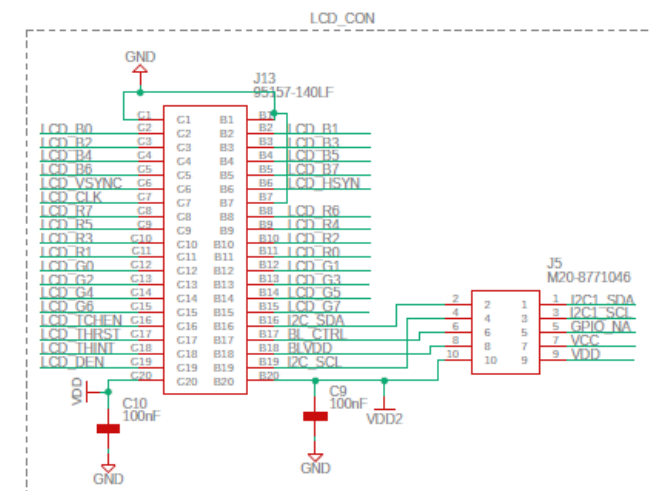
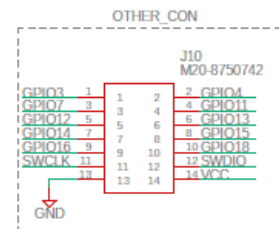
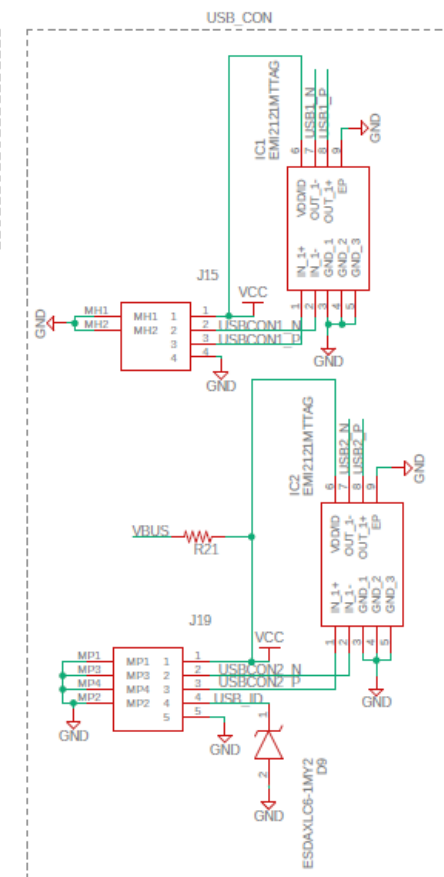
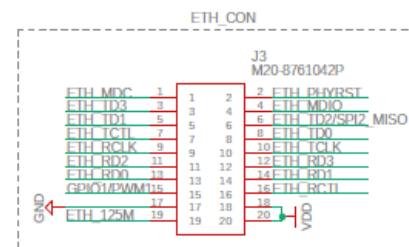
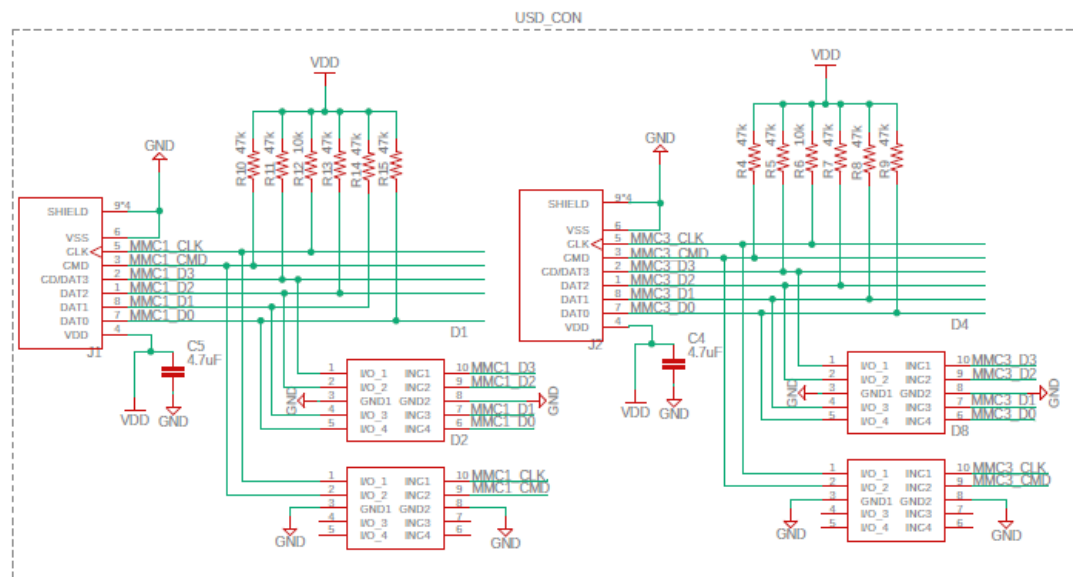


Figura 129. Modelo 3D de la placa generada para la computadora de abordo





UNAM
FACULTAD DE INGENIERÍA
Elaborado por:
Luis Manuel Juárez Ruiz

COMP_FD v69
02/11/2023 02:51 p. m.
Sheet: 2/2

Tabla 38. Lista de materiales para la PCB de la computadora de abordo.

Dispositivo	Descripción	Designación	Cantidad
502386-0670	Cabecal receptáculo (Hembra), 6 posiciones con paso de 1.25mm	J6	1
505578-0471	Cabecal receptáculo (Macho), 4 posiciones con paso de 2mm	J11	1
53261-0671	Cabecal receptáculo (Macho), 6 posiciones con paso de 1.25mm	J7	1
95157-140LF	Headers pin strip, 40 pines con paso de 2.54mm, recto, 2 columnas	J13	1
ASAAIG-24.000MHZ-K-C-S-T	Oscilador 24 MHz (estándar) 3.3V	Y1	1
DS04-254-2-03BK-SMT	DIP Switch, paso de 2.54mm, 3 posiciones	S2	1
EMI2121MTTAG	EMI Filtro Modo Común con protección ESD	IC1, IC2	2
ESDAXLC6-1MY2	Diodo TVS	D3, D6, D7, D9	4
HSP051-4N10	Protección ESD	D1, D2, D4, D8	4
M20-8760742R	Headers pin strip, 14 pines con paso de 2.54mm, recto, 2 columnas	J10	1
M20-8761042P	Headers pin strip, 20 pines con paso de 2.54mm, recto, 2 columnas	J3, J12	2
M20-8770242	Headers pin strip, 2 pines con paso de 2.54mm, recto, 1 columna	J9, J14	2
M20-8770442	Headers pin strip, 4 pines con paso de 2.54mm, recto, 1 columna	J8, J16, J18, J22	4
M20-8770642	Headers pin strip, 6 pines con paso de 2.54mm, recto, 1 columna	J20, J21	2
M20-8770842	Headers pin strip, 8 pines con paso de 2.54mm, recto, 1 columna	J4	1
M20-8771042	Headers pin strip, 10 pines con paso de 2.54mm, recto, 1 columna	J5	1
MEM2067-02-180-00-A_REVB	Conector de memoria microSD	J1, J2	2
MYC-YA151C	Sistema en Módulo (SOM)	U1	1
PTS647SK50SMTR2LFS	Push button, Normalmente abierto	S1, S3, S4	3
USB1046-GF-0190-L-B-A	Conector USB tipo A (Receptáculo)	J15	1
USB3085-30-A	Conector USB tipo micro A/B	J19	1
Capacitor Genérico	CHIP-1210 (3225-MÉTRICO)	C1, C2, C3, C7, C8, C9, C10	7

Capacitor Genérico	10uF, CHIP-1210 (3225-MÉTRICO)	C6	1
Capacitor Genérico	4.7uF, CHIP-1210 (3225-MÉTRICO)	C4, C5	2
CL32A476MOJNNWE	Capacitor Cerámico 47uF, 16V	C11, C12, C13, C14	4
Resistor genérico	CHIP-1210 (3225-MÉTRICO)	R1, R2, R3,, R18, R19, R20, R21	4
Resistor genérico	100 OHM, CHIP-1210 (3225-MÉTRICO)	R17	4
Resistor genérico	10k OHM, CHIP-1210 (3225-MÉTRICO)	R6, R12, R16	3
Resistor genérico	47k OHM, CHIP-1210 (3225-MÉTRICO)	R4, R5, R7, R8, R9, R10, R11, R13, R14, R15	10

APÉNDICE V. GENERACIÓN DE LA PCB PARA EL SUBSISTEMA DE POTENCIA

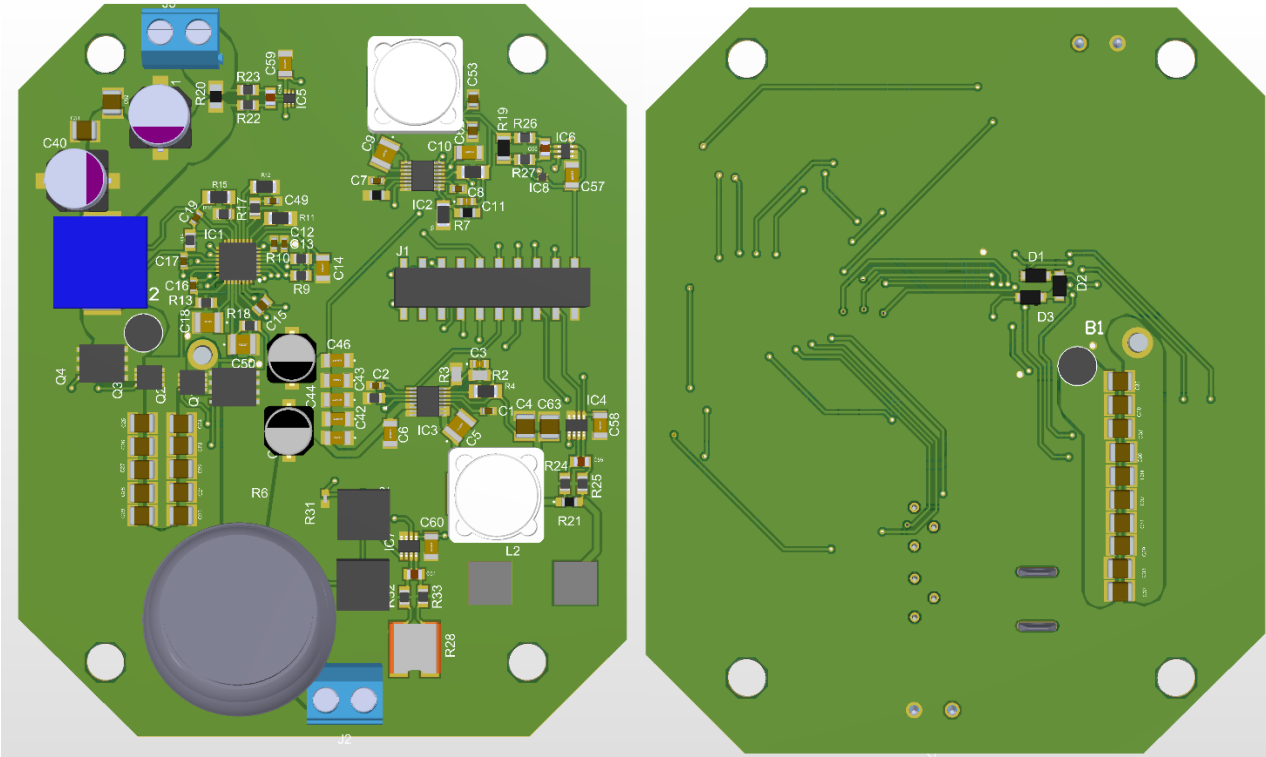
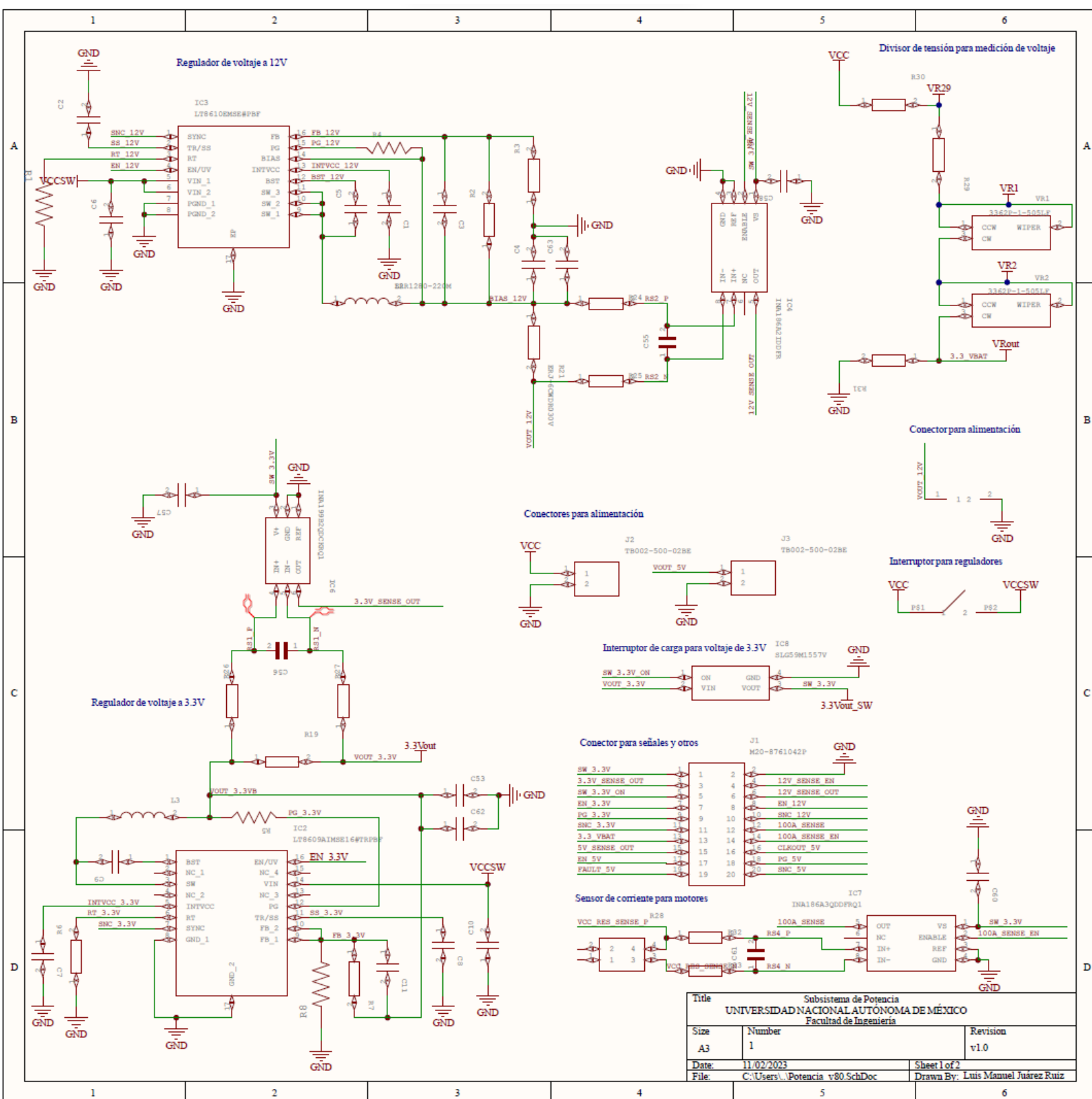


Figura 130. Modelo 3D de la placa generada para el sistema de potencia



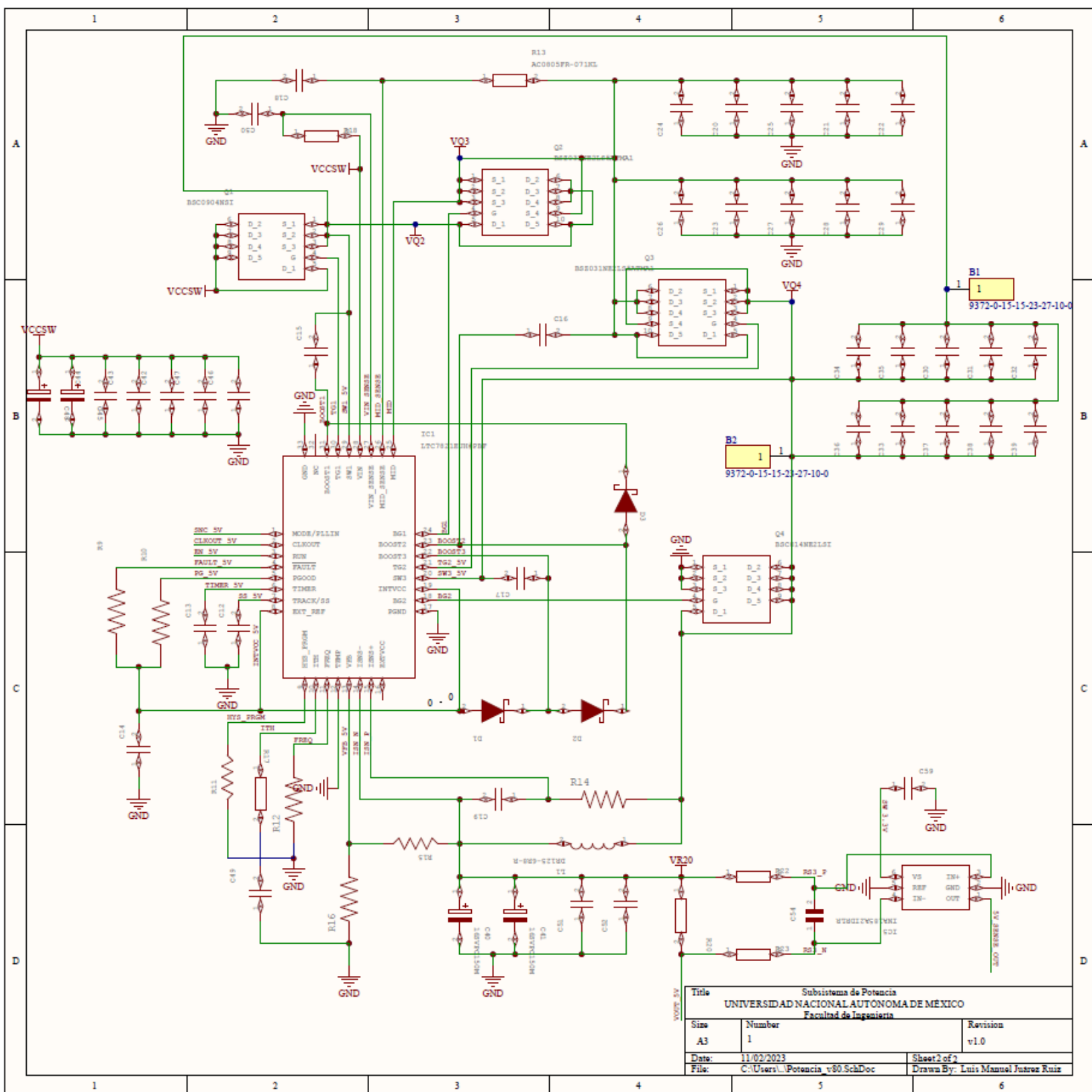


Tabla 39. Lista de Materiales para la PCB del sistema de potencia

Dispositivo	Descripción	Fabricante	Designación	Cantidad
16SVPC150M	Capacitor de polímero 150uF, 16V dc, 20%	Panasonic	C40, C41	2
3362P-1-505LF	Potenciómetro 5M, 0.5W, 10%	Bourns	VR1, VR2	2
08055C124KAT2A	Capacitor cerámico de capas múltiples 120nF, 50 V dc, +/-10%	AVX	C15	1
9372-0-15-15-23-27-10-0	Pin receptáculo, 11.2A	Mill-Max	B1, B2	2
AC0805FR-071KL	Resistor 1 kohm, +/- 1%, 125 mW	YAGEO	R13	1
AC0805FR-0710KL	RES 10K OHM, 1%, 1/8W	YAGEO	R9, R10	2
BSC015NE2LS5IATMA1	MOSFET N-Ch 25V, 100A, TDSO-8 OptiMOS	Infineon	Q4	1
BSC0906NS	MOSFET N-Ch 30V, 63A, TDSO-8 OptiMOS	Infineon	Q1	1
BSZ031NE2LS5ATMA1	MOSFET N-Ch 25V, 40A (Tc), PG-TSDSO-8-FL	Infineon	Q2, Q3	2
C0603C100K3RACAUTO	Capacitor cerámico de capas múltiples, 10 pF, 10%, 25 VDC	KEMET	C3, C11	2
C1206C104K3HACAUTO	Capacitor cerámico de capas múltiples, 25V, 0.1uF, 10%	KEMET	C57, C58, C59, C60	4
C1206C225K5RACAUTO7210	Capacitor cerámico de capas múltiples, 2.2 uF, 10%, 50 VDC	KEMET	C42, C43, C44, C46, C47	5
C1210C104K5RAC7210	Capacitor cerámico de capas múltiples, 0.1 uF, 10%, 50 VDC	KEMET	C5, C9, C18, C50	4
CL10B105KA8NNNL	Capacitor cerámico de capas múltiples, 1uF, +/-10%, 25V	SAMSUNG	C1, C7	2
CL21A106KAYNNNE	Capacitor cerámico de capas múltiples, 10uF, 25V, 10%	SAMSUNG	C54, C55, C56, C61	4
CL21A476MQYNNNG	Capacitor cerámico de capas múltiples, 47uF, 6.3Vdc, +/-20%	SAMSUNG	C53, C62	2
CL32A476MOJNNNE	Capacitor cerámico de capas múltiples, 47uF, 16V, 20%	SAMSUNG	C4, C63	2
CRCW06033M30FKEA	Resistor 3.3M, 0.10W, 1%	Vishay	R29	1
CRCW06038M20FKEA	RESISTOR 8.2M, 1%, 0.1W	Vishay	R30, R31	2
CRCW08052M26FKEA	RESISTOR 2.26M, 1%, 0.125W	Vishay	R2	1
CRCW0805200KFKEAC	Resistor 200K Ohm, 1%, 0.125W(1/8W)	Vishay	R3	1
CRF1206-FZ-R040ELF	Resistor Shunt 40 mOhms, 1%, 1W	Bourns	R19	1
DR125-1R5-R	Inductor de potencia blindado (Ferrita), 1.5uH 20%, 13.8A, 2.9mOhm DCR	Eaton	L1	1
EEH-ZA1H330XP	Capacitor Electrolítico Híbrido, 50V, 33uF, 20%	Panasonic	C45, C48	2
ERJ-6CWDRO30V	Resistor Shunt, 0.03ohm, 0.5W, 0.5%	Panasonic	R21	1

ERJ-6ENF1334V	Resistor 1.33Mohms, .125W, 1%	Panasonic	R7	1
ERJ-6ENF5232V	Resistor 52.3Kohms, .125W, 1%	Panasonic	R6	1
GCD188R71H103KA01D	Capacitor cerámico de capas múltiples, 0.01 uF, 50 VDC, 10%	Murata Electronics	C12	1
GCI188R71E102KA01D	Capacitor cerámico de capas múltiples, 1000 pF, 25 VDC, 10%	Murata Electronics	C2, C8, C49	3
GJ832ER7YA106KA12L	Capacitor cerámico de capas múltiples, 10 uF, 35 VDC, 10%	Murata Electronics	C20, C21, C22, C23, C24, C25, C26, C27, C28, C29, C30, C31, C32, C33, C34, C35, C36, C37, C38, C39, C51, C52	22
GRM31CR71H475KA12K	Capacitor cerámico de capas múltiples, 4.7 uF, 50 VDC, 10%	Murata Electronics	C6, C10, C14	3
GRM188R71E224KA88D	Capacitor cerámico de capas múltiples, 0.22 uF, 25 VDC, 10%	Murata Electronics	C16, C19	2
GRM188R71E474KA12D	Capacitor cerámico de capas múltiples, 0.47 uF, 25 VDC, 10%	Murata Electronics	C13, C17	2
INA185A2IDRLR	Amplificador de detección de corriente	Texas Instruments	IC5	1
INA186A2IDDFR	Amplificador de detección de corriente	Texas Instruments	IC4	1
INA186A3QDDFRQ1	Amplificador de detección de corriente	Texas Instruments	IC7	1
INA199B2QDCKRQ1	Amplificador de detección de corriente	Texas Instruments	IC6	1
KCA2ANA1BBB	Interruptor		U\$1	1
LT8609AIMSE16#TRPBF	Regulador conmutado (Buck) 2A/3A	Analog Devices	IC2	1
LT8610EMSE#PBF	Regulador conmutado (Buck) 2.5A	Analog Devices	IC3	1
LTC7821EUH#PBF	Controladores de conmutación híbridos (Buck)	Analog Devices	IC1	1
M20-8761042P	2.54mm Pin Header, 10+10 contacts	Harwin	J1	1
PMEG40T10ERX	Diodo Schottky 40 V 1A	Nexperia	D1, D2, D3	3
RC0805FR-074K7L	Resistor 4.7k, 0805 (2012M), +/-1%, 0.125W	YAGEO	R17	1
RC0805FR-0771K5L	Resistor 71.5K OHM, 1%, 1/8W	YAGEO	R1	1
RC1206FR-0768KL	RESISTOR 68K OHM, 0.25W, ± 1%	YAGEO	R12	1
AC1206FR-07412KL	Resistor 412K Ohm, 1%, 1/4W	YAGEO	R8	1
RC1206FR-13100KL	Resistor 100kOhms, 1/4W, 1%	YAGEO	R4, R5, R11, R15	4
RK73H2ATTD1912F	Resistor 19.1K Ohm, 1%, 0.25W(1/4W)	KOA	R16	1
RK73H2ATTD2321F	Resistor 2.32K Ohm, 1%, 0.25W(1/4W)	KOA	R14	1

SLG59M1557V	Interruptor de carga 28.5mOhm, 1.0A	Renesas Electronics	IC8	1
SRR1280-100M	Inductor de potencia de Ferrita (blindado), 10 uH, +/-20%	Bourns	L3	1
SRR1280-220M	Inductor de potencia de Ferrita (blindado), 22 uH +/-20%	Bourns	L2	1
TB002-500-02BE	Bloques de terminales	CUI Devices	J2, J3	2
WSLP12066L000FEA	RESISTOR SHUNT 0.006 OHM, 1%, 1W	Vishay	R20	1
WSLP2726L3000FEA	RESISTOR SHUNT 300 UOHM, 1%, 7W	Vishay	R28	1
Resistor Genérico	CHIP_RESISTOR 1206		R18, R22, R23, R24, R25, R26, R27, R32, R33	9

APÉNDICE VI. VERIFICACIÓN DE LA PCB DE POTENCIA

En la generación del diseño de la PCB de potencia, se verificó cuáles son las características de la placa capaz de soportar la corriente propuesta en el diseño, principalmente la de 10[A]. Lo anterior se realizó por medio de una herramienta de análisis de potencia proporcionada por Altium Designer.

Una de las primeras consideraciones para usar esta herramienta es definir la máxima densidad corriente de nuestra placa. Este cálculo se puede aproximar con la ayuda de alguna calculadora de ancho de traza, por ejemplo, la que proporciona Digikey en su página web [61].

Para poder generar el ancho de traza deseado, se deben conocer las características intrínsecas del material utilizado, en este caso, el cálculo se hará para una placa con material FR-4 que es la más común comercialmente hablando y, el cual puede soportar temperaturas hasta 110°C aproximadamente.

Por tanto, en la calculadora se agregan los siguientes parámetros:

- Corriente: 10[A]
- Espesor (cobre): 35[um] ó $1\frac{oz}{ft^2}$ (más común de manera comercial)
- Temperatura ambiente: 25°C
- Aumento de temperatura: 10°C

Con los valores anteriores, la calculadora indica un ancho de traza requerido de 7.2[mm], con esto el ancho de traza diseñado no sería el suficiente, además, este valor de ancho de traza resulta un valor bastante grande. Ahora algunas recomendaciones sería entonces

cambiar el espesor de la placa o el aumento de temperatura, lo cual, permitiría un ancho de traza menor.

La desventaja de aumentar la temperatura en la placa de evaluación no resulta tan conveniente debido a que este factor tiene impactos sobre el mismo material, por ejemplo, cuando un material de cobre se expone a altas temperaturas este tiende a expandirse y alterar la forma de este, entre otros, esto mismo sucede con la placa utilizada. Además, un aumento en la temperatura puede significar una disminución en la vida de los dispositivos electrónicos y, se debe tener en mente que el cambio de temperatura no es algo que podamos controlar.

Por tanto, si cambiamos el espesor de cobre:

- Corriente: 10[A]
- Espesor (cobre): 70[um] ó $2\frac{oz}{ft^2}$
- Temperatura ambiente: 25°C
- Aumento de temperatura: 10°C

La calculadora nos proporciona un valor de 3.6[mm]. Como en el diseño se tiene una traza de 5mm, la calculadora nos indica que esta traza soportará una corriente de 12.7[A]. Ahora la densidad corriente está definida de la siguiente manera:

$$J = \frac{I}{A} \left[\frac{A}{mm^2} \right]$$

Donde:

- J es la densidad de corriente
- I es la corriente que fluye a través del conductor
- A es el área transversal del conductor

El área transversal del conductor lo sacamos con los datos que se mencionaron anteriormente:

$$A = width \times thickness = (70um)(5mm)$$

$$A = 0.35mm^2$$

$$J = \frac{12.7[A]}{0.35[mm^2]} = 36.28 \left[\frac{A}{mm^2} \right]$$

Ahora con este valor, se puede realizar el análisis de potencia con la herramienta de Altium, para ello primero se definen las conexiones de los dispositivos por medio de las redes generadas en el esquemático.

Una vez que tengamos definido las conexiones del dispositivo podremos verificar en el “árbol” si las conexiones son congruentes con lo que se requiere analizar justo como se muestra en la Figura 131.

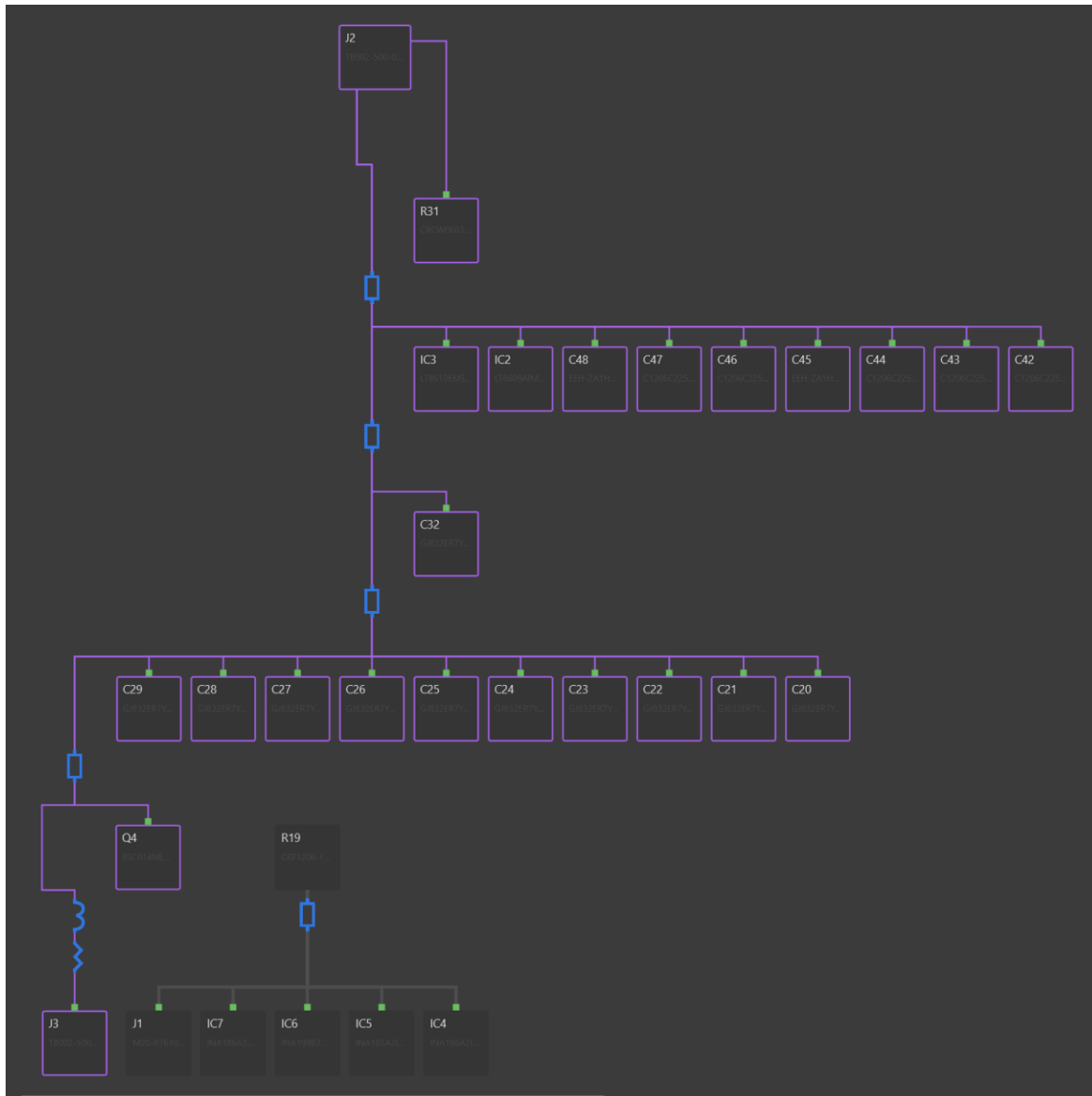


Figura 131. "árbol" del sistema de potencia

Cuando las conexiones estén generadas correctamente, se usará la herramienta de análisis para corroborar el diseño de la PCB (Figura 132).

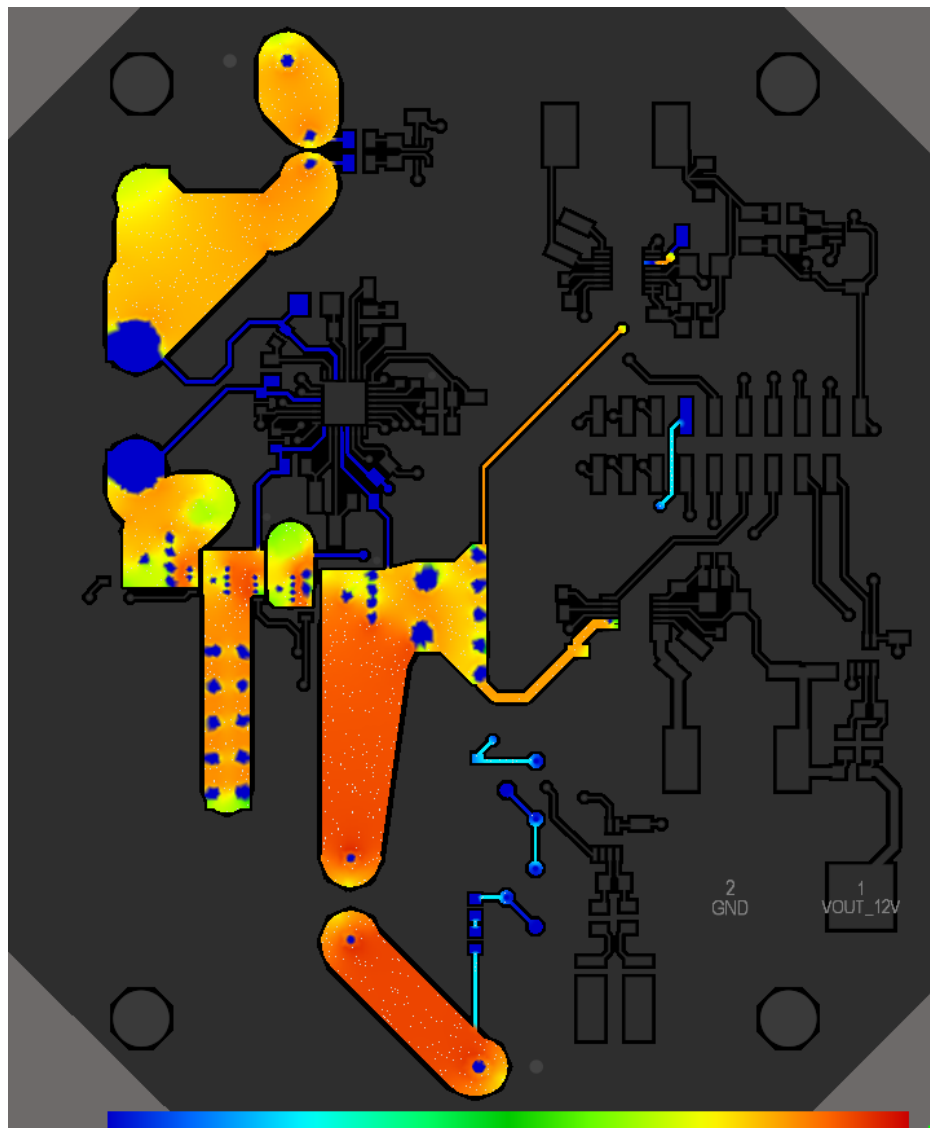


Figura 132. Mapa de calor de la PCB de potencia

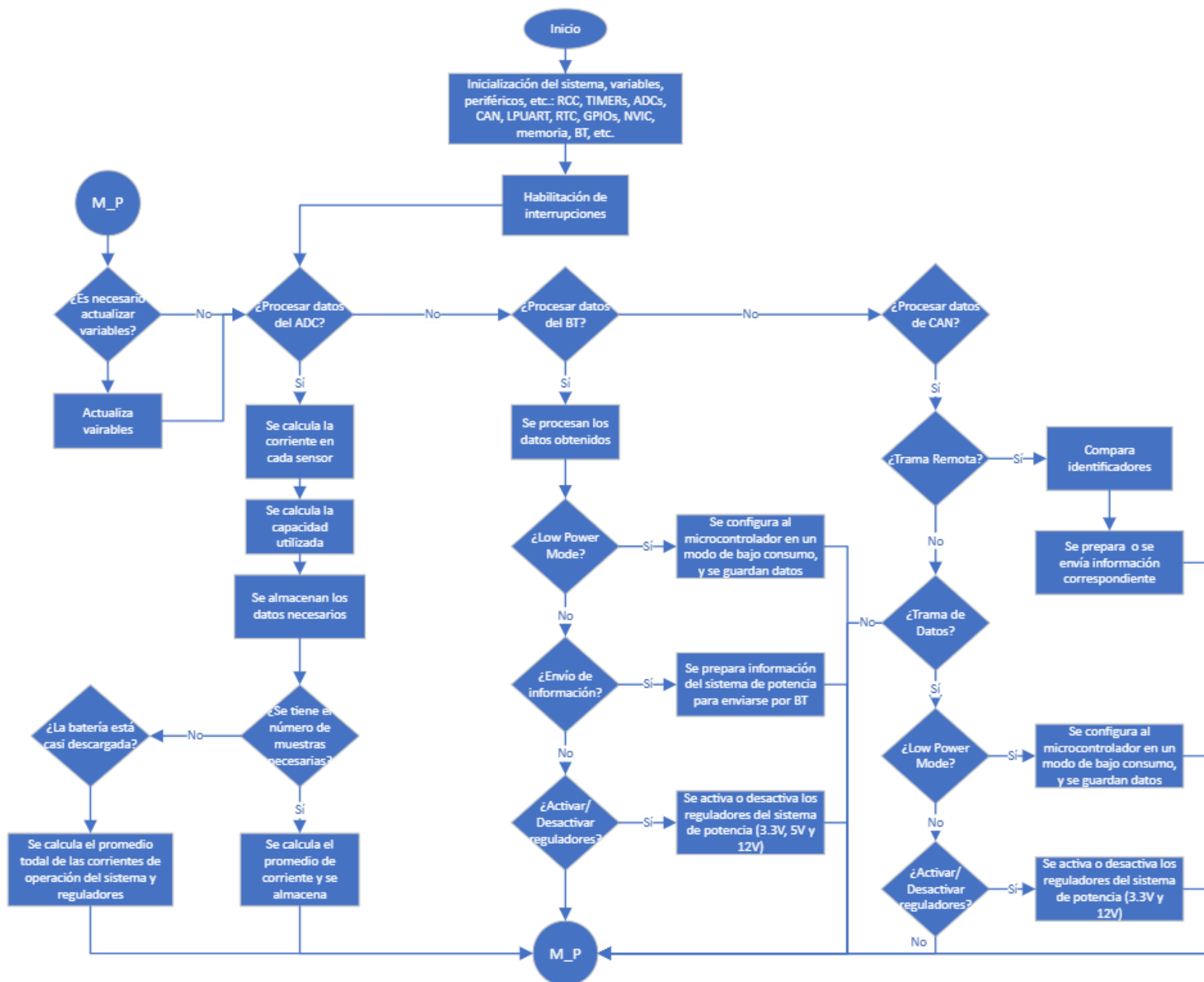
En este caso, la simulación indicó que hay ciertas zonas que sobrepasan la densidad de corriente máxima calculada, sin embargo, los puntos donde ocurren estos valores son donde se ubican las terminales de los circuitos integrados u otros dispositivos electrónicos que son capaces de manejar estas densidades de corriente.³³ Con esto se puede verificar que una PCB con cobre de 0.7um de espesor resulta una buena opción para el prototipo de PCB. Hay que destacar que existen páginas de fabricantes de PCB que dan

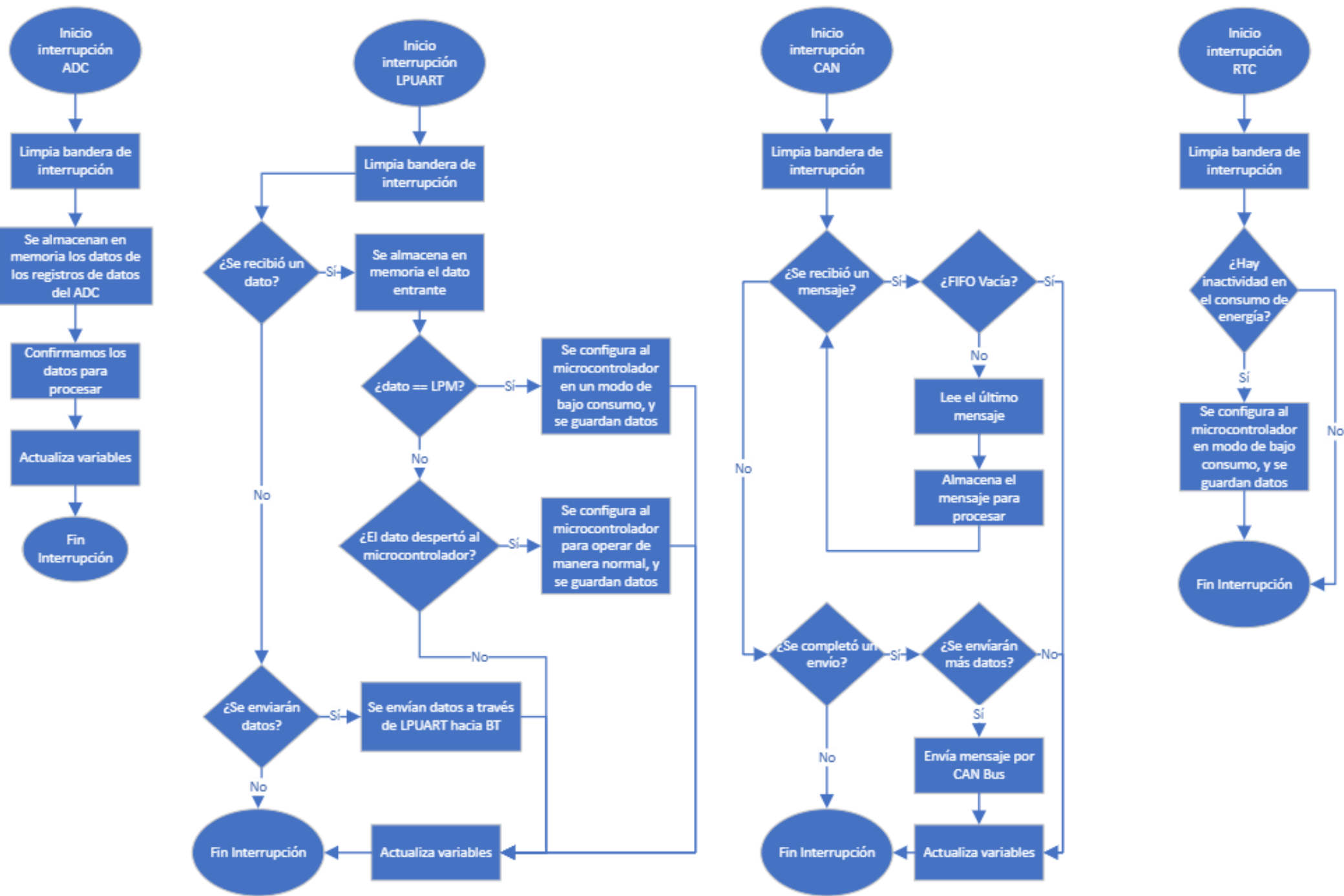
³³ Para el análisis se obtuvieron de la simulación mostrada en el APÉNDICE I los valores de corriente RMS y, con estos se hizo la simulación del análisis de potencia.

recomendaciones sobre el ancho de la pista y la corriente que puede soportar, mencionando el uso de la calculadora anterior.

APÉNDICE VII. Diagrama de flujo para el subsistema de potencia

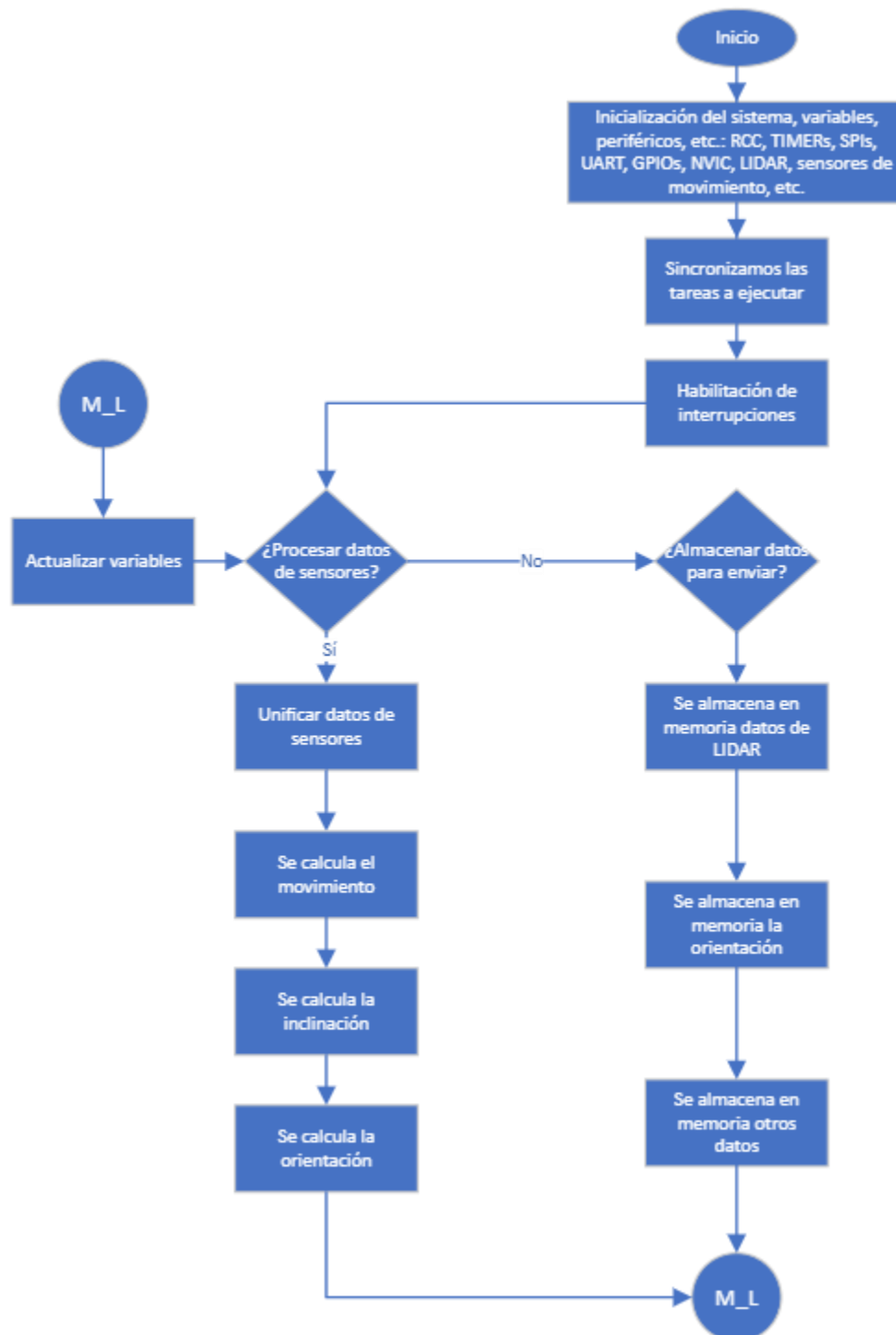
En este apartado se presenta el diagrama de flujo de la aplicación desarrollada para el subsistema de potencia. En estos diagramas se incluye el diagrama de flujo de la función principal y de las interrupciones implementadas para su funcionamiento.

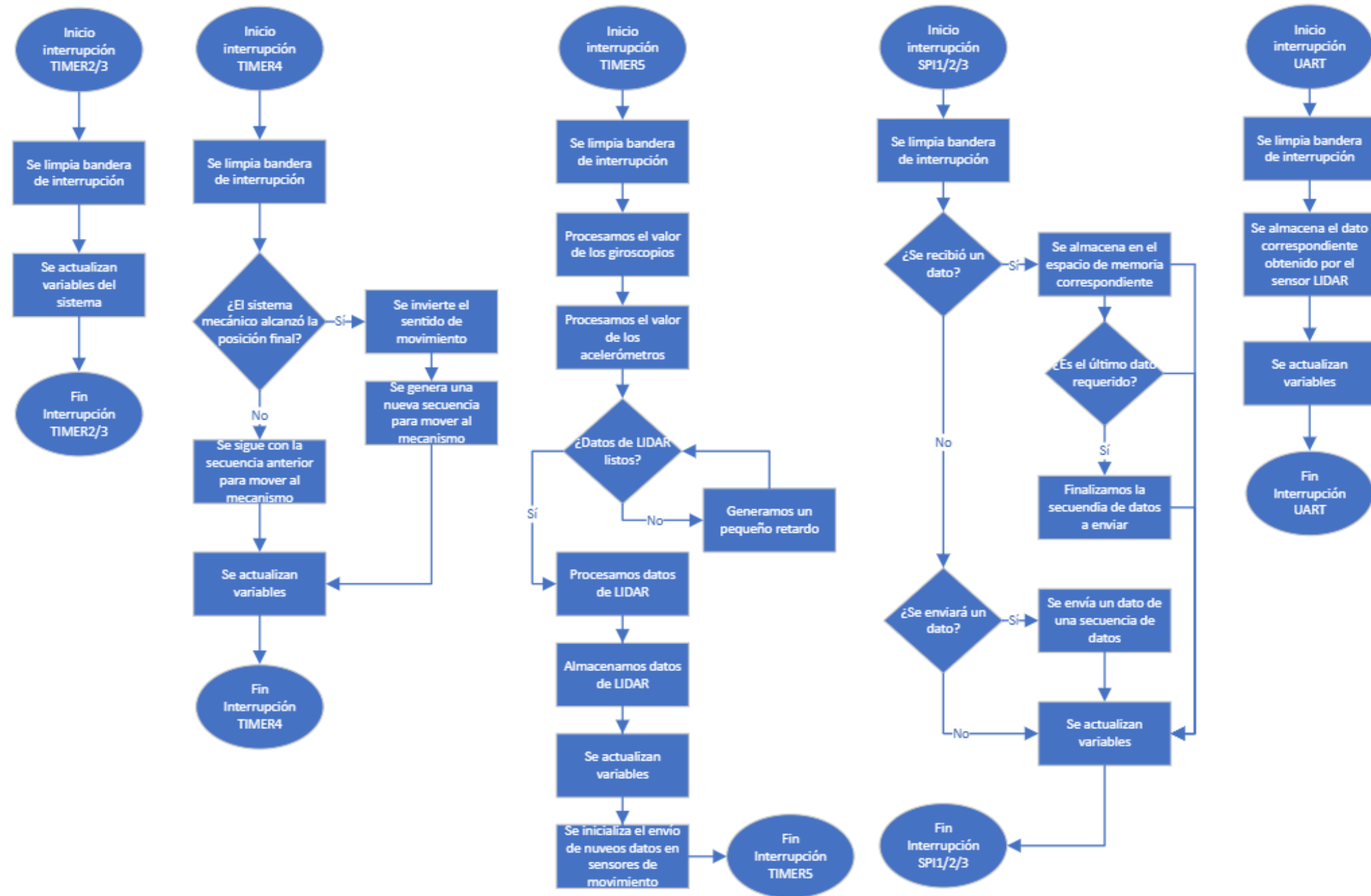




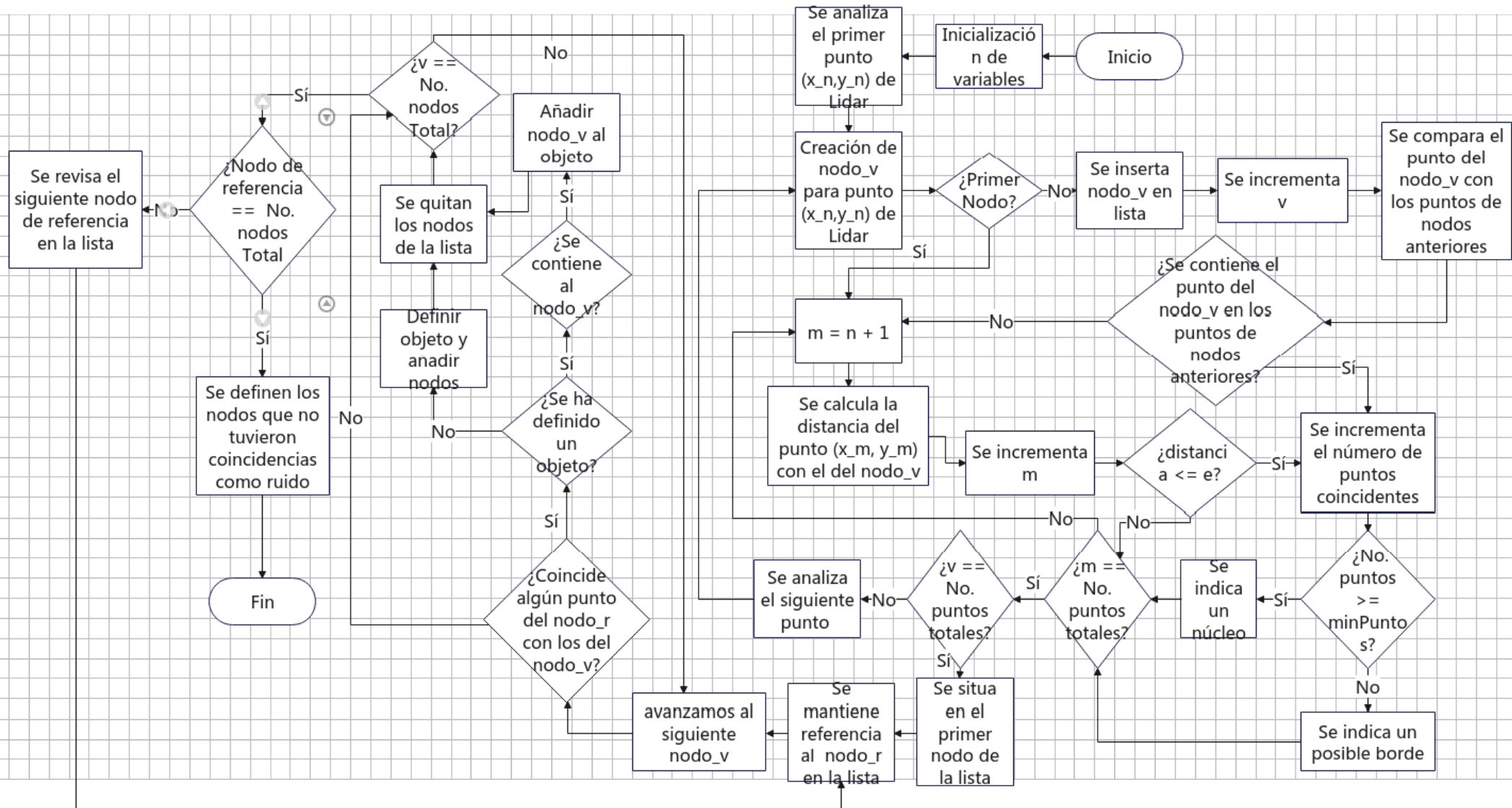
APÉNDICE VIII. Diagrama de flujo para el subsistema de detección de objetos

En este apartado se presenta el diagrama de flujo de la aplicación desarrollada para el subsistema de detección de objetos. En estos diagramas se incluye el diagrama de flujo de la función principal y de las interrupciones implementadas para su funcionamiento.

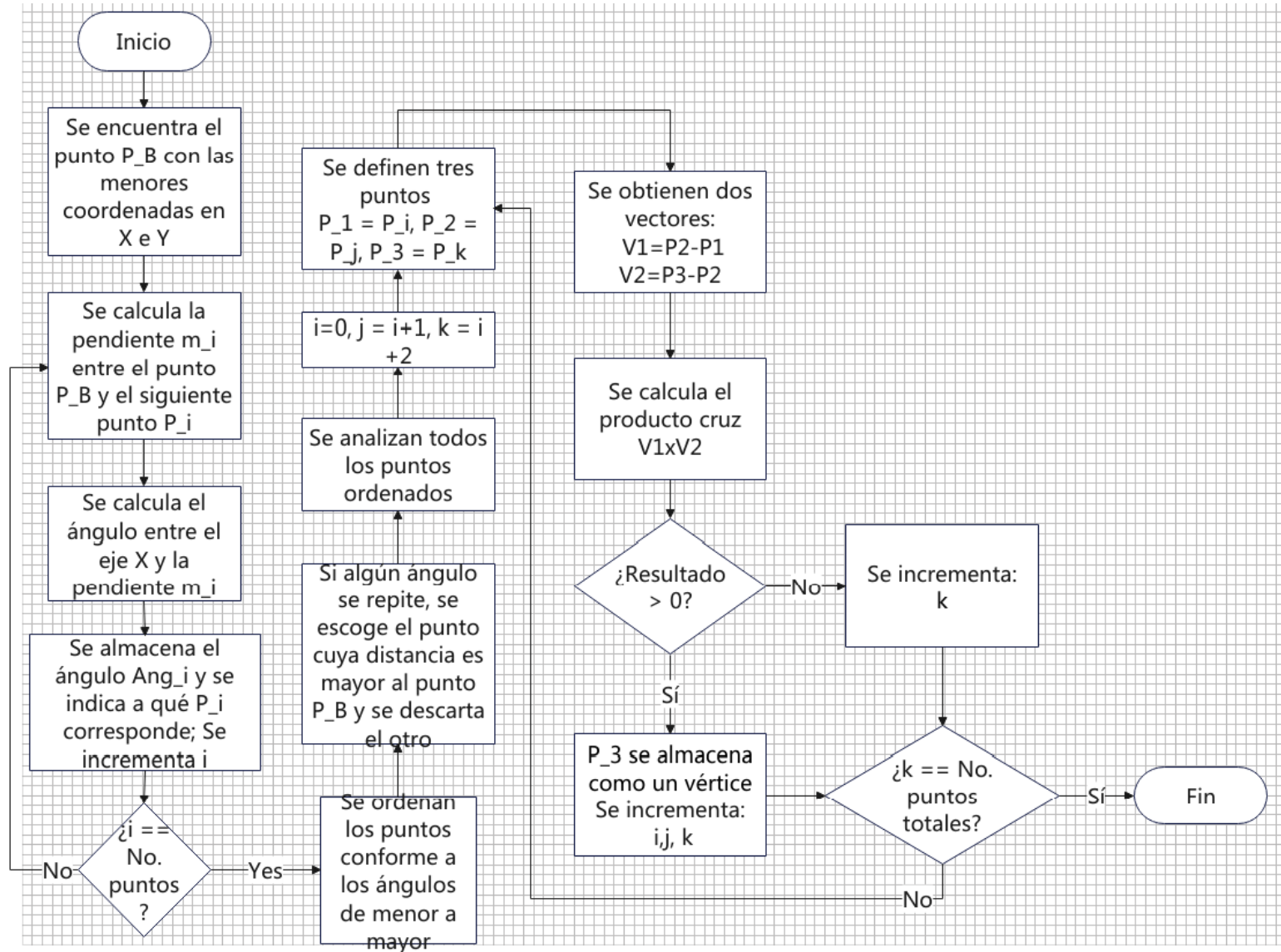




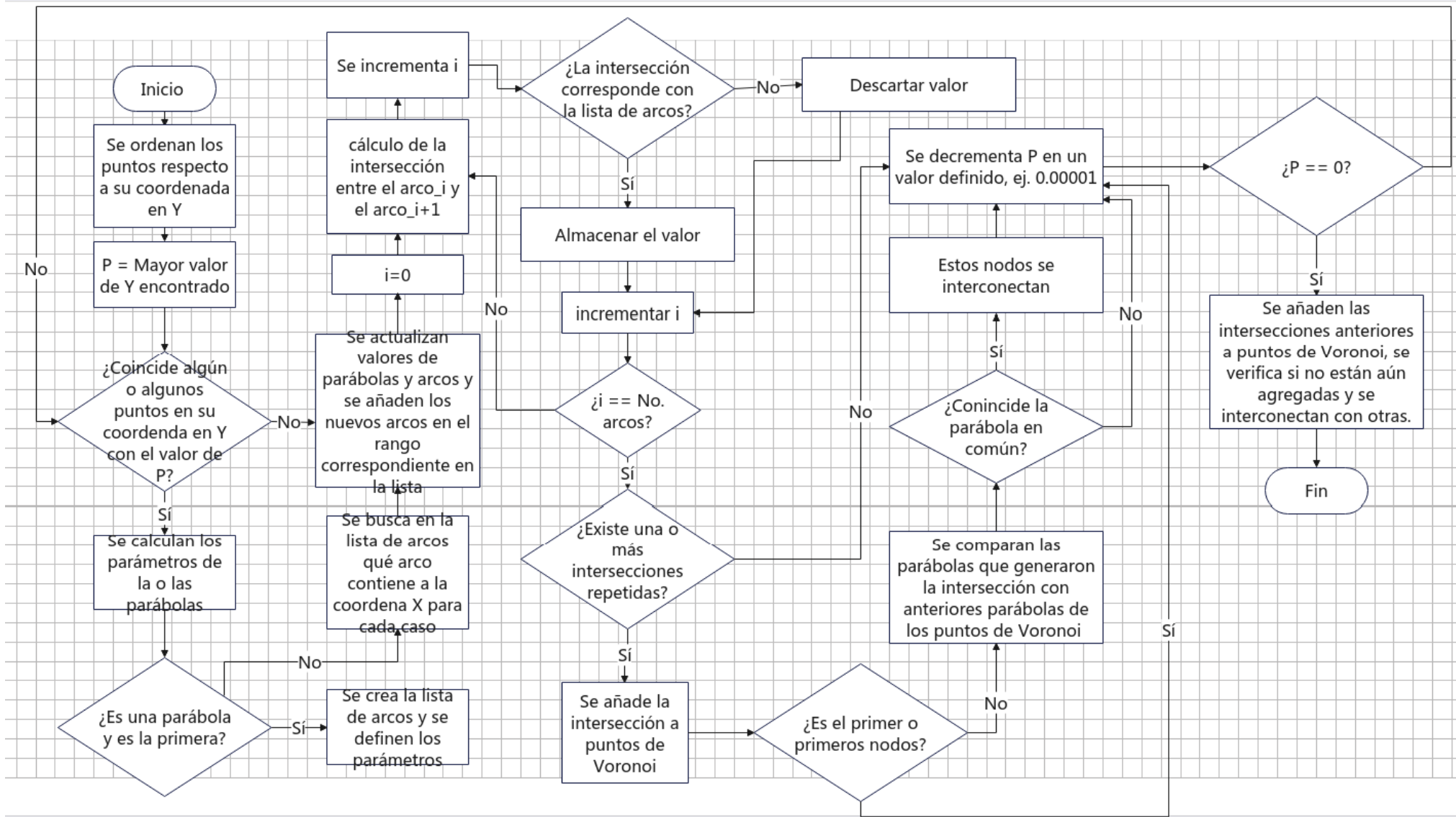
APÉNDICE IX. Diagrama de flujo del algoritmo DBSCAN



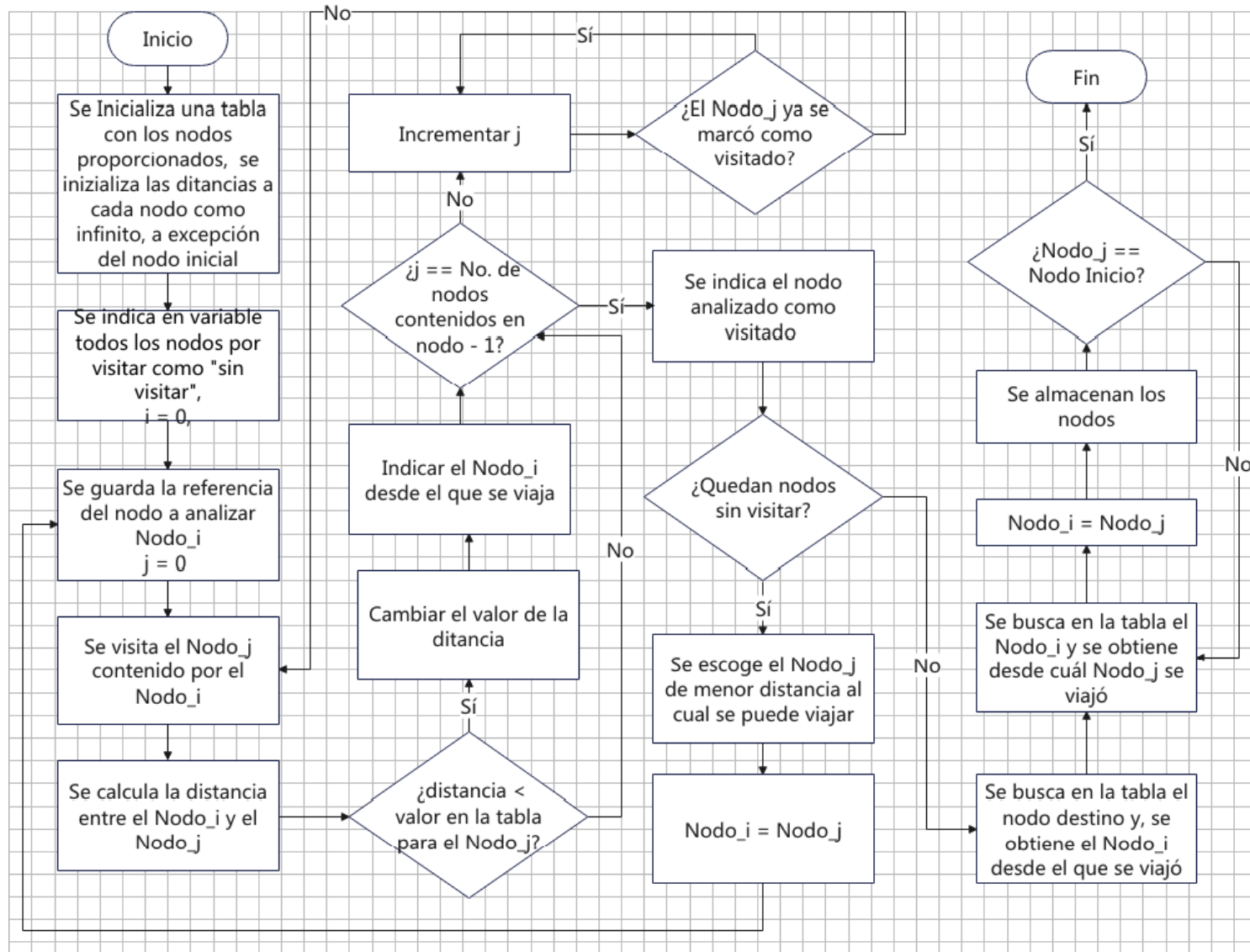
APÉNDICE X. Diagrama de flujo del algoritmo Convex Hull



APÉNDICE XI. Diagrama de flujo del algoritmo Voronoi-Fortune



APÉNDICE XII. Diagrama de flujo del algoritmo Dijkstra



APÉNDICE XIII. Configuración de los periféricos

A manera de complementar el entendimiento del funcionamiento del sistema, se agregan en la Tabla 40 y la Tabla 41 las configuraciones de los periféricos implementados en cuestión.

Tabla 40. Configuración del microcontrolador del subsistema de potencia.

Periférico/otro	Configuración/Descripción	Puertos utilizados
RCC	Frecuencia principal del sistema de 80MHz, implementada por medio de un oscilador interno de 16MHz y el PLL. Los periféricos están limitados a una frecuencia de 40MHz.	NA
TIMER2	-Frecuencia de operación: 1KHz. -Modo de operación: • PWM, • Salidas desactivadas, • Ciclo del trabajo del 100%. -Uso: Muestreo de sensores.	NA
TIMER3/4/5	-Frecuencias de operación: 800KHz, 500KHz, 600KHz, respectivamente. -Modo de operación: • PWM, • Salida del canal 1 activada, • Ciclos del trabajo del 14.9%, 45% y 54%, respectivamente. -Uso: Sincronización para los reguladores.	TIM3_CH1→GPIOA6 TIM4_CH1→GPIOB6 TIM5_CH1→GPIOA0
ADC1	-Modo de operación: • Se utilizan los dos tipos de grupos para el ADC (grupo inyectado y grupo regular), • Sobremuestreo activado para implementar convertidor de 16 bits, • Interrupción por final de secuencia habilitado. -Grupo inyectado: Modo auto-inyección con 4 canales en conversión. -Grupo regular: Modo discontinuo con 1 canal en conversión y, conversión generada por disparo debido al temporizador 2. -Uso: Medición de los sensores para conocer el uso de la batería.	ADC1_CH1→GPIOC0 ADC1_CH2→GPIOC1 ADC1_CH3→GPIOC2 ADC1_CH4→GPIOC3 ADC1_CH6→GPIOA1
CAN	-Modo de operación: • Tasa de transmisión de 1Mbps,	CAN1_Rx→GPIOB8 CAN1_Tx→GPIOB9

	• Modo de operación normal, • Prioridad definida por identificador, • Uso de 6 identificadores para sensores y otros, • Uso de transmisión y recepción por interrupción. -Uso: Comunicación entre la computadora de abordó y el subsistema de potencia.	
LPUART	-Modo de operación: • Tasa de transmisión de 115200 bps, • Modo Full-dúplex, • Trama compuesta por 8 bits de datos, sin bit de paridad, 1 bit de inicio y 1 bit de stop, • Uso de transmisión y recepción por interrupción. • En modo de bajo consumo el baudrate se disminuye a 9600 bps y el reloj es un oscilador externo de 32,768Hz. -Uso: Comunicación con el dispositivo bluetooth	LPUART_Rx→GPIOB10 LPUART_Tx→GPIOB11
RTC	-Modo de operación: • Se usa como fuente del RTC el oscilador externo de 32,768Hz para generar una frecuencia de 1Hz, • Se configura la Alarma A para generar una interrupción cada minuto. -Uso: Capturar valores de tiempo y verificar el estado del subsistema.	NA
GPIOs	-Modo de operación: Se configuraron 3 puertos en el modo de salida. -Uso: Habilitar y deshabilitar los reguladores conmutados.	EN_3.3V→GPIOC10 EN_5V→GPIOC11 EN_12V→GPIOC12
HC05	-Modo de operación: Se configura igual que LPUART -Uso: Comunicación por medio de bluetooth	NA
Low Power Mode (LPM)	-Modo de operación: • En el modo de bajo consumo, el dispositivo se configura en el modo STOP 2, • Se habilita LPUART y RTC para operar en este modo de bajo consumo, por medio del oscilador externo de 32,768Hz. • Se deshabilitan y colocan en un estado bajo los demás periféricos (deshabilitación de reguladores),	NA

	• LPUART se limita a un baudrate máximo de 9600 bps y se configura para poder sacar al microcontrolador de este estado de bajo consumo. • Al salir del modo de bajo consumo, el microcontrolador vuelve a su configuración normal.	
--	---	--

Tabla 41. Configuración del microcontrolador para el subsistema de detección de obstáculos.

Periférico/ Dispositivo	Configuración/Descripción	Puertos utilizados
RCC	Frecuencia principal del sistema de 80MHz, implementada por medio de un oscilador externo de 8MHz y el PLL. Los periféricos están limitados a una frecuencia de 40MHz.	NA
TIMER1	-Frecuencia de operación: 50Hz. -Modo de operación: • PWM, • Salidas del canal 1 y 2 activadas, -Uso: Control de los servomotores.	TIM1_CH1→GPIOA8 TIM1_CH2→GPIOA9
TIMER2	-Frecuencia de operación: 1Hz. -Modo de operación: • PWM, • Salidas desactivadas, • Ciclo del trabajo del 100%, • En modo Maestro envía una señal a los esclavos (TIMER 3 a 5) al actualizarse, • Habilitación de interrupción. -Uso: Control de temporizadores.	NA
TIMER3	-Frecuencia de operación: 10Hz. -Modo de operación: • PWM, • Salidas desactivadas, • Ciclo del trabajo del 100%, • En modo Maestro envía una señal al esclavo (TIMER 4) al actualizarse, • En modo Esclavo se reinicializa el temporizador, • Habilitación de interrupción. -Uso: Control del temporizador 4.	NA
TIMER4/5	-Frecuencia de operación: 10.5Hz y 993Hz, respectivamente.	NA

	-Modo de operación: • PWM, • Salidas desactivadas, • Ciclo del trabajo del 100%, • En modo Esclavo se reinicializa el temporizador, • Habilitación de interrupción. -Uso: Cambio de dirección de servomotores y muestreo de los sensores, respectivamente.	
SPI1/2/3	-Modo de operación: • Configuración del reloj: 625KHz, CPOL = 1, CPHA = 1, • Frecuencia del reloj de 10MHz para lectura, • Modo maestro, • Modo Full-dúplex, • Trama de datos de 16 bits, • NSS por control de software, • Uso de transmisión y recepción por interrupción. -Uso: Comunicación con los sensores de movimiento.	SPI1_SCLK→GPIOA5 SPI1_MISO→GPIOA6 SPI1_MOSI→GPIOA7 SPI1_CS1→ GPIOC1 SPI2_SCLK→GPIOB13 SPI2_MISO→GPIOB14 SPI2_MOSI→GPIOB15 SPI2_CS1→ GPIOC2 SPI3_SCLK→GPIOB3 SPI3_MISO→GPIOB4 SPI3_MOSI→GPIOB5 SPI3_CS1→ GPIOC3
UART3	-Modo de operación: • Tasa de transmisión de 921600 bps, • Modo Full-dúplex, • Trama compuesta por 8 bits de datos, sin bit de paridad, 1 bit de inicio y 1 bit de stop, • Uso de transmisión y recepción por interrupción. -Uso: Comunicación con el sensor TF02.	USART3_Tx→GPIOC10 USART3_Rx→GPIOC11
USB1	-Modo de operación: • El puerto de USB operará con una velocidad 12Mbps (FS), • Se configurará como dispositivo únicamente, • La clase del dispositivo será configurada como CDC para envío de información, • Funcionará por medio de interrupciones. -Uso: Envío de datos del sistema de detección.	USB1_DM→ GPIOA11 USB1_DP→ GPIOA12
MPU9250	-Modo de operación: Los sensores estarán operando por medio de SPI y se deshabilita la interfaz I2C. -Giroscopio: • Escala $\pm 250^\circ/\text{s}$,	NA

	• Muestreo de 1KHz. -Acelerómetro: • Escala $\pm 2g$, • Muestreo de 1KHz. Uso: Estimación de la orientación.	
TF02	-Modo de operación: • Se selecciona UART como protocolo de comunicación, • Baudrate de 921600 bps, • Tasa de muestreo de 1KHz, • Formato de salida en centímetros, • Envío de datos por tiempo. Uso: Detección de objetos.	NA

Como comentario adicional, los drivers utilizados para el manejo de los periféricos y dispositivos mencionados fueron desarrollados desde cero, implementando la mayor parte de las funcionalidades de las interfaces, a excepción de USB cuya biblioteca se obtuvo de los desarrolladores de STMicroelectronics.

APÉNDICE XIV. Promedio de las corrientes del subsistema y cálculo del menor espacio de memoria requerido

El promediar los valores de corriente en el sistema nos permite conocer el consumo de corriente aproximado y con ello buscar formas para mejorar el rendimiento del sistema.

El promedio de un grupo de “N” muestras, siendo x_1 la primera muestra, se puede representar de la siguiente manera:

$$M = \frac{1}{N} \sum_{i=1}^N x_i$$

La manera anterior es la forma más común de promediar diferentes valores, sin embargo, al aumentar el tiempo de operación y disminuir el tiempo de muestreo, el valor de las muestras incrementa considerablemente, por ejemplo, supongamos que queremos obtener el consumo promedio del sistema para 10 minutos y, considerando que la frecuencia de muestreo es de 1KHz, la cantidad de muestras queda definido como,

$$N = \frac{(10[\text{min}]) \left(\frac{60}{1} \left[\frac{s}{\text{min}} \right] \right)}{0.001s} = 600,000 \text{ muestras}$$

Entonces, para disminuir la cantidad de espacio requerido, se planea realizar el promedio de los promedios obtenidos a lo largo del funcionamiento del sistema, es decir, promediamos una cantidad de “n” muestras durante un intervalo de tiempo y, al final se promedian esos promedios para obtener un estimado final. Esto se puede representar de la siguiente forma,

$$M = \frac{1}{n} \sum_{i=1}^n x_i$$

$$I_{Avg} = \frac{1}{m} \sum_{i=1}^m M_i$$

Para demostrar que la proposición anterior nos dará el mismo resultado que realizar el promedio directo de las “N” muestras medidas durante el tiempo de operación del sistema, desarrollamos las expresiones anteriores

$$M = \frac{1}{n} (x_1 + x_2 + \dots + x_{n-1} + x_n)$$

Como las muestras cambian entre cada grupo diferente, podemos designarlo con un valor diferente de x_i , por ejemplo y_i , entonces, tenemos,

$$I_{Avg} = \frac{1}{m} \left(\frac{1}{n} (x_1 + x_2 + \dots + x_{n-1} + x_n) + \dots + \frac{1}{n} (y_1 + y_2 + \dots + y_{n-1} + y_n) \right)$$

$$I_{Avg} = \frac{1}{m \times n} (x_1 + x_2 + \dots + x_{n-1} + x_n + \dots + y_1 + y_2 + \dots + y_{n-1} + y_n)$$

Ahora, si sabemos que “n” es la cantidad de muestras en un intervalo de tiempo y “m” es la cantidad de promedios, al realizar la multiplicación entre ambos, estaremos calculando la cantidad total de “N” muestras obtenidas durante el tiempo de operación del sistema, además, si el grupo de muestras y_i se obtuvieron en el promedio “m”, se puede reescribir lo siguiente

$$y_1 + y_2 + \dots + y_{n-1} + y_n = x_{(m-1)n} + x_{(m-1)n+1} + \dots + x_{m \times n-1} + x_{m \times n}$$

$$y_1 + y_2 + \dots + y_{n-1} + y_n = x_{N-n} + x_{N-n+1} + \dots + x_{N-1} + x_N$$

Por tanto, podemos reescribir la corriente promedio como,

$$I_{Avg} = \frac{1}{N} (x_1 + x_2 + \dots + x_{n-1} + x_n + \dots + x_{N-n} + x_{N-n+1} + \dots + x_{N-1} + x_N)$$

$$I_{Avg} = \frac{1}{N} \sum_{i=1}^N x_i$$

Con lo anterior, queda comprobado que estamos obteniendo un valor promedio correcto. Ahora bien, ¿cómo se puede definir un valor de “m” y “n” de acuerdo con las características de nuestro sistema? La respuesta a esta pregunta es a través del uso de cálculo, para ello plantearemos algunas ecuaciones.

En primer caso, sabemos que necesitamos “n” muestras en un intervalo de tiempo, esto lo podemos obtener de la siguiente manera

$$n = \frac{T_n}{T_{Sensor}}$$

Donde

- n: Es el número de muestras obtenidas en un intervalo de tiempo menor al de operación.
- T_n : Es el periodo requerido para obtener las “n” muestras.
- T_{Sensor} : Es el periodo de muestreo del sensor.

Ahora bien, en segundo lugar, necesitamos calcular “m” promedios durante el tiempo de operación de nuestro sistema,

$$m = \frac{T_{Tot}}{T_n}$$

Donde

- m: Es el número de promedios obtenidos en el tiempo de operación.
- T_n : Es el periodo requerido para obtener las “n” muestras.
- T_{Tot} : Es el periodo de operación del sistema.

Con las ecuaciones anteriores lo que se quiere es obtener el menor espacio de memoria a ocupar, por lo tanto,

$$y = m + n$$

$$y = \frac{T_n}{T_{Sensor}} + \frac{T_{Tot}}{T_n}$$

$$y = \frac{T_n^2 + (T_{Tot} \times T_{Sensor})}{T_n \times T_{Sensor}}$$

Derivando la expresión anterior para poder calcular el mínimo valor,

$$y' = \frac{T_n^2 - (T_{Tot} \times T_{Sensor})}{T_n^2 \times T_{Sensor}}$$

Obteniendo el mínimo valor,

$$T_n^2 - (T_{Tot} \times T_{Sensor}) = 0$$

$$T_n = \sqrt{T_{Tot} \times T_{Sensor}}$$

Finalmente, los valores “m” y “n” se pueden obtener por medio de definir la frecuencia de muestreo y el tiempo de operación del sistema, por ejemplo, regresemos al caso anterior, donde definíamos un tiempo de operación del sistema por 10 minutos y, una frecuencia de muestreo de 1KHz,

$$T_n = \sqrt{600s \times 0.001s} \approx 0.775[s]$$

Por consecuente el valor, de “m” y “n”

$$m = \frac{600}{0.775} \approx 774$$

$$n = \frac{0.775}{0.001} \approx 775$$

Donde claramente,

$$774 + 775 \ll 600,000$$

APÉNDICE XV. Generación de la aplicación para el sistema

En la elaboración de este proyecto, se generó el prototipo de una aplicación (con ayuda del software Android Studio) capaz de dar soporte en el uso y manejo del sistema del dron repartidor. Es por esto por lo que en este apéndice se muestran las diferentes partes conformantes de la aplicación, justo como se muestra a continuación.

En la Figura 133 se muestra una pantalla de inicio, con un botón de inicio, cuyo propósito es establecer conexión con el sistema del dron repartidor, donde, una vez establecida la conexión, enviará a la pantalla del menú (Figura 134)

En la Figura 134 se muestra el menú, donde se ven diferentes elementos, descritos seguidamente:



Figura 133. Pantalla de inicio

1. Una barra de selección del menú, en la cual se pueden escoger 6 opciones: Estado del dron (Figura 135), Estado de ruta (Figura 136), Control de vuelo (Figura 137), Configuración (Figura 138), Estado del paquete (Figura 139) y Estado del sistema (Figura 140).
2. Algunos elementos que muestran el estado del sistema principalmente la batería, su modo de operación y si hay conectividad por bluetooth.
3. Un botón para operar en modo de seguridad, en caso de algún problema con el dron.
4. Un interruptor para cambiar el control de automático a manual.



Figura 134. Pantalla del menú

En la Figura 135 se muestra el estado del dron que despliega diferentes datos, como los que se muestran en esta imagen y, además, también se muestra información sobre el subsistema de potencia si se desliza la pantalla hacia arriba.



Figura 135. Pantalla del estado del dron

En la Figura 136 se muestra una pantalla con un mapa, haciendo referencia a que en esta parte se debe mostrar el camino que recorrerá el dron para hacer la entrega, mostrando en todo momento su posición. Para generar la vista de este mapa, se utilizó una herramienta de código abierto para generar mapas denominada Open Street Map, aunque existen mejores alternativas como Google Maps, solo que está requiere de ciertas llaves de activación para ser usada.

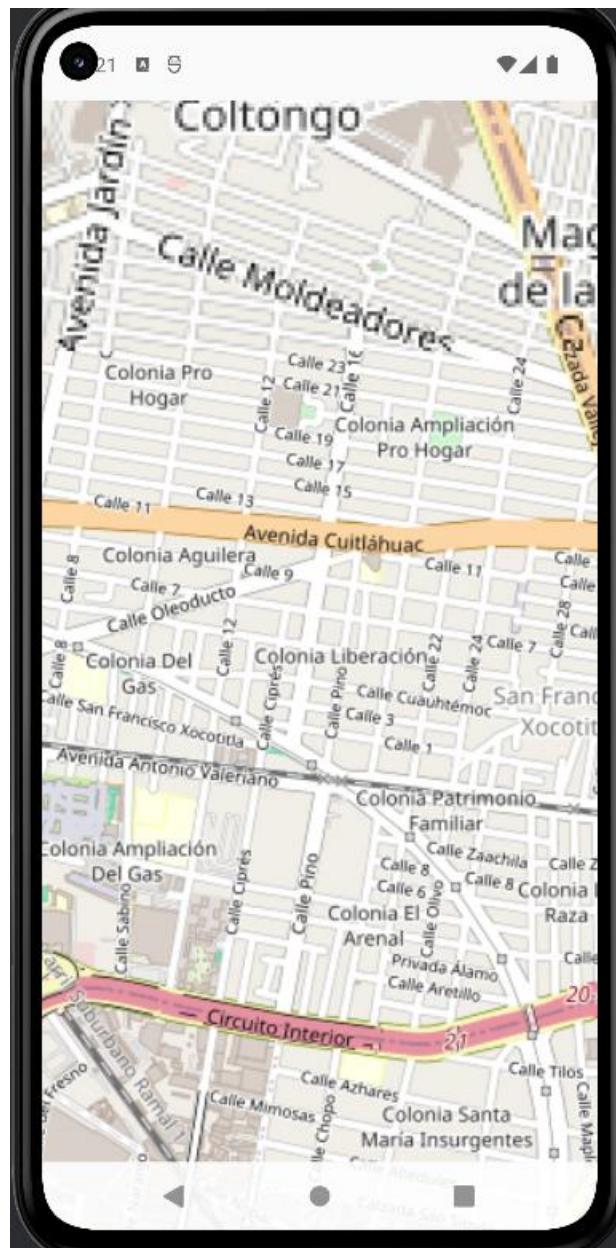


Figura 136. Pantalla del estado de la ruta

En la Figura 137 se muestra una pantalla con dos Joysticks, cuyo propósito es poder controlar de manera manual el dron cuando se active esta opción, además, esta pantalla puede servir para mostrar la vista del dron cuando se necesite.

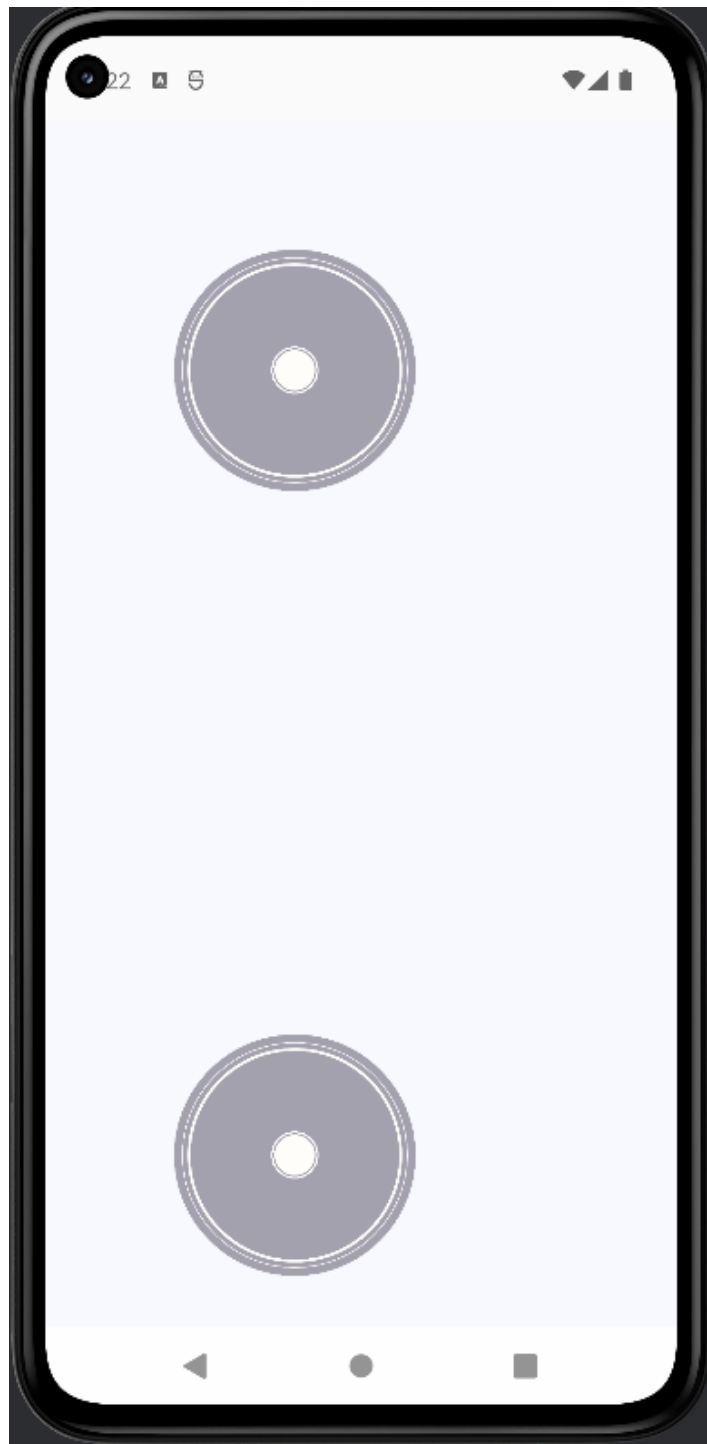


Figura 137. Pantalla de control de vuelo

En la Figura 138 se muestran las diferentes opciones para configurar todos los aspectos del dron a través de diversos interruptores y botones, donde en su mayoría se usan para activar y desactivar elementos en este. Además, se pueden definir algunos parámetros por medio de cadenas de texto, como la altura, la fecha y la hora.

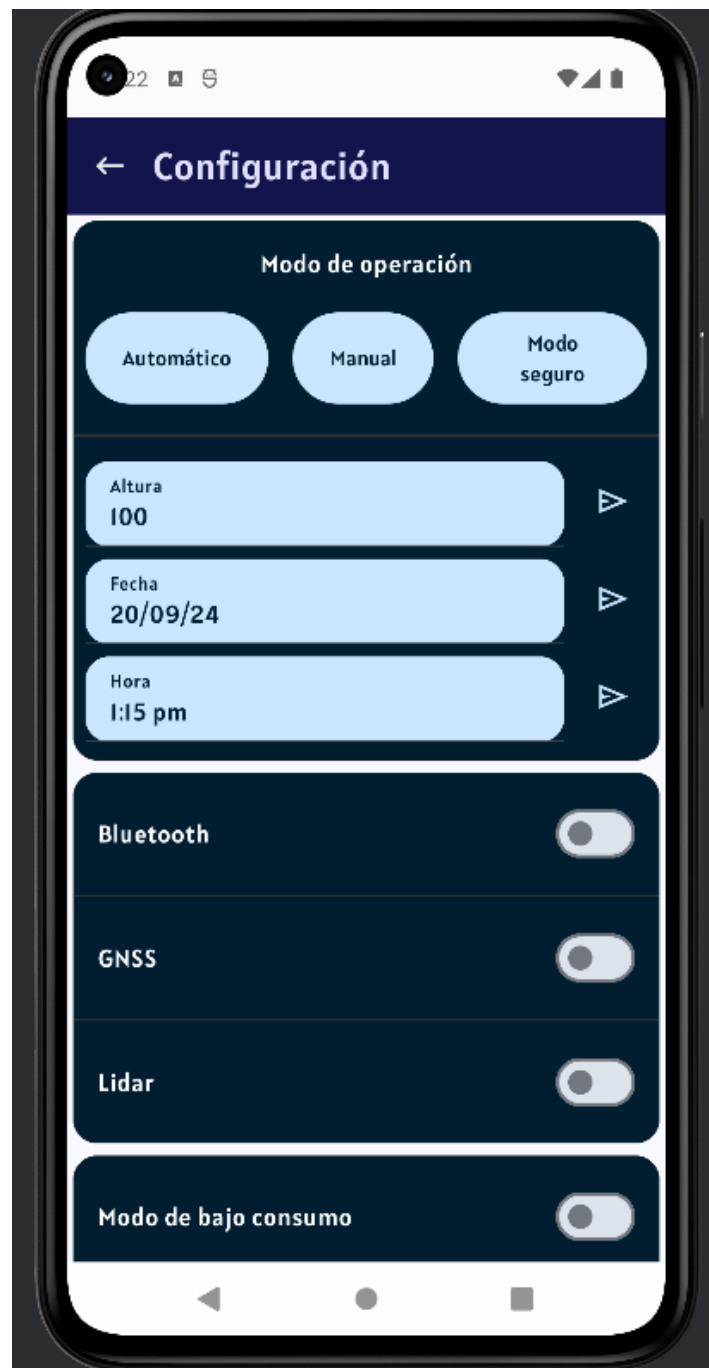


Figura 138. Pantalla de configuración del dron

En la Figura 139 se muestra una pantalla con información relevante del paquete que contendrá el dron repartidor.



Figura 139. Pantalla del estado del paquete

En la Figura 140 se muestra una pantalla que muestra información relevante del sistema del dron repartidor, siendo principalmente datos sobre los dispositivos y elementos que lo conforman.



Figura 140. Pantalla de información del sistema

Referencias

- [1] Y. B. Sebbane, Smart Autonomous Aircrafts, CRC Press, 2016.
- [2] e. a. Kimon P. Valavanis, Handbook of Unmanned Aerial Vehicles, Springer, 2015.
- [3] SECRETARIA DE COMUNICACIONES Y TRANSPORTES, «NORMA Oficial Mexicana NOM-107-SCT3-2019, Que establece los requerimientos para operar un sistema de aeronave pilotada a distancia (RPAS) en el espacio aéreo mexicano.,» Ciudad de México, 2019.
- [4] e. a. Eulalia Balestrieri, «Sensors and Measurements for Unmanned Systems: An Overview,» 2021.
- [5] e. Fadi Al-Turjman, UAVs assessment in software-defined IoT networks: An overview, Elsevier, 2020.
- [6] Z. Gacovski, Unmanned Aerial Vehicles (UAV) and Drones, Arcler Press, 2021.
- [7] «Quaternion,» Quaternion, [En línea]. Available: <https://www.quaternion.com/hybrx-multirotor-breaks-the-barrier-in-flight-time/>.
- [8] e. a. Kimon P. Valavanis, Handbook of Unmanned Aerial Vehicles, Springer, 2019.
- [9] «ECU MASTER,» ECUMaster, [En línea]. Available: <https://ecumasterusa.com/products/ecumaster-pmu16-power-management-unit?variant=35809051115684>.
- [10] «banggood,» banggood, [En línea]. Available: https://www.banggood.com/Lantian-PDB-Power-Distribution-Board-with-5V-12V-3A-BEC-Output-for-CC3C-Naze32-Upgraded-version-for-RC-Drone-FPV-Racing-p-1076479.html?gmcCountry=US¤cy=USD&cur_warehouse=CN&createTmp=1.
- [11] Y. B. Sebbane, Multi-UAV Planning and Task Allocation, CRC Press, 2020.
- [12] Maxime Perreault y Kamran Behdinan, «Delivery Drone Driving Cycle,» 2021.
- [13] e. a. Dimitrios Pavlou, «Suitability Analysis of Implementing a Fuel Cell on a Multirotor Drone,» 2020.
- [14] e. a. Vandana Mohindru, Unmanned Aerial Vehicles for Internet of Things (IoT): Concepts, Techniques, and Applications, Scrivener Publishing LLC, 2021.
- [15] B. Giovino, «Digikey,» 29 10 2020. [En línea]. Available: <https://www.digikey.com.mx/es/articles/how-to-quickly-improve-drone-performance-and-extend-flight-time>.

- [16] «EmQopter,» EmQopter, [En línea]. Available: https://www.emqopter.de/en/deliverydrone.php#link_standard. [Último acceso: 2022].
- [17] «DJI Developer,» DJI, 13 12 2016. [En línea]. Available: https://developer.dji.com/mobile-sdk/documentation/introduction/flightController_concepts.html.
- [18] «KEYENCE,» KEYENCE, [En línea]. Available: https://www.keyence.eu/ss/products/measure/measurement_library/type/laser_1d/.
- [19] C. J. Fisher, «Analog,» Analog Devices, 2010. [En línea]. Available: <https://www.analog.com/en/resources/app-notes/an-1057.html>.
- [20] Instituto Federal de Telecomunicaciones, «El espectro radioeléctrico en México. Estudio y Acciones.,» Ciudad de México.
- [21] «BATTERY UNIVERSITY,» 27 10 2021. [En línea]. Available: <https://batteryuniversity.com/article/bu-503-determining-power-deliver-by-the-ragone-plot>.
- [22] «Grepow,» Shenzhen Grepow Battery Co., Ltd, [En línea]. Available: <https://www.grepow.com/high-discharge-rate-battery.html>.
- [23] schaffnerusa, «Mouser,» [En línea]. Available: <https://www.mouser.com/pdfDocs/schaffner-engineers-guide-designing-emi-filters1-13-17.pdf>.
- [24] S. Franco, DESIGN WITH OPERATIONAL AMPLIFIERS AND ANALOG INTEGRATED CIRCUITS, McGraw-Hill Education, 2015.
- [25] A. Pini, «Digikey,» Digikey, 21 08 2018. [En línea]. Available: <https://www.digikey.com.mx/en/articles/select-apply-current-sense-amplifiers-effectively-manage-power>.
- [26] «ST,» STMicroelectronic, 26 08 2020. [En línea]. Available: https://www.st.com/resource/en/application_note/an4304-how-to-filter-the-input-of-a-highside-current-sensing-stmicroelectronics.pdf.
- [27] «Analog,» Analog Devices, 07 01 2007. [En línea]. Available: <https://www.analog.com/en/resources/technical-articles/performance-of-currentsense-amplifiers-with-input-series-resistors.html>.
- [28] J. Savage Carmona, «Bio-Robotics,» [En línea]. Available: <https://biorobotics.fi-p.unam.mx/robots-moviles/>.

- [29] F. Cazaurang, Multi-Rotor Platform-based UAV Systems, Elsevier Ltd, 2020.
- [30] F. Al-Turjman, Unmanned Aerial Vehicles in Smart Cities, Springer Nature, 2020.
- [31] D. M. H. Sadraey., «Design of Unmanned Aerial Systems,» 2020.
- [32] J. Cohen, «Think Autonomous,» 07 05 2020. [En línea]. Available: <https://www.thinkautonomous.ai/blog/how-lidar-detection-works/amp/>.
- [33] T. Opsahl, «uio,» [En línea]. Available: https://www.uio.no/studier/emner/matnat/its/nedlagte-emner/UNIK4690/v16/forelesninger/lecture_3_3-robust-estimation-with-ransac.pdf.
- [34] N. S. Chauhan, «KDnuggets,» Guiding Tech Media, 04 04 2022. [En línea]. Available: <https://www.kdnuggets.com/2020/04/dbscan-clustering-algorithm-machine-learning.html>.
- [35] P. Kindermann, «Youtube,» 18 06 2020. [En línea]. Available: <https://www.youtube.com/playlist?list=PLubYOWSI9mIuUsLBKM4NlzPrmBA5jKL-r>.
- [36] e. a. Sujit Kuthirummal, «A graph traversal based algorithm for obstacle detection using lidar or stereo,» 2011.
- [37] Wing, «YouTube,» 17 09 2021. [En línea]. Available: <https://www.youtube.com/watch?v=VOFxsFhEXoU>.
- [38] «MAVLink,» Dronecode Project, [En línea]. Available: <https://mavlink.io/en/>.
- [39] «SIMCom,» SIMCom Wireless Solutions Limited, [En línea]. Available: <https://www.simcom.com/product/A7670X.html>.
- [40] «SIMCom,» SIMCom Wireless Solutions Limited, [En línea]. Available: https://www.simcom.com/SIMCom_product_catalogues.html.
- [41] «u-blox,» u-blox, [En línea]. Available: <https://www.u-blox.com/en/docs/UBX-21022436>.
- [42] «github,» SparkFun, [En línea]. Available: https://github.com/sparkfun/SparkFun_u-blox_GNSS_Arduino_Library.
- [43] «DNK Power,» DNK, [En línea]. Available: [https://www.dnkpowers.com/how-to-calculate-battery-run-time/#:~:text=ln%20the%20ideal%2Ftheoretical%20case,Ah\)%2FCurrent\(A\)..](https://www.dnkpowers.com/how-to-calculate-battery-run-time/#:~:text=ln%20the%20ideal%2Ftheoretical%20case,Ah)%2FCurrent(A)..)
- [44] M. Koebler, «Lithium Hub,» Lithium Hub, 28 02 2018. [En línea]. Available: <https://lithiumhub.com/discharge-rate/>.

- [45] «Power-Sonic,» Power-Sonic, [En línea]. Available: <https://www.power-sonic.com/blog/what-is-a-battery-c-rating/>.
- [46] «genstattu,» Gens ace and Tattu, [En línea]. Available: <https://www.genstattu.com/bw/>.
- [47] «Mouser,» Mouser Electronics, [En línea]. Available: <https://www.mouser.com/pdfdocs/te-microchip-automotive-grade-components.pdf>.
- [48] «arm Keil,» ARM, [En línea]. Available: <https://armkeil.blob.core.windows.net/developer/Files/pdf/guide/starters-guide-to-arm-processing-power-in-automotive.pdf>.
- [49] STMicroelectronics, «STMicroelectronics,» [En línea]. Available: https://www.st.com/content/st_com/en.html. [Último acceso: 22 05 2025].
- [50] Benewake, «Benewake,» [En línea]. Available: <https://en.benewake.com/uploadfiles/2024/04/20240426135442695.pdf>. [Último acceso: 22 05 2025].
- [51] Z. Peterson, «Altium,» 04 06 2021. [En línea]. Available: <https://resources.altium.com/es/p/pcb-routing>. [Último acceso: 24 05 2025].
- [52] Proto-Electronics.com, «Proto-Electronics.com,» [En línea]. Available: <https://www.proto-electronics.com/blog/top-10-pcb-routing-tips>. [Último acceso: 24 05 2025].
- [53] «MYIR Tech,» MYIR Tech, [En línea]. Available: <https://www.myirtech.com/list.asp?id=658>.
- [54] STMicroelectronics, «ST,» [En línea]. Available: <https://www.st.com/en/microcontrollers-microprocessors/stm32mp151c.html>. [Último acceso: 25 05 2025].
- [55] «ST,» STMicroelectronis, 09 2022. [En línea]. Available: https://www.st.com/resource/en/application_note/an5031-getting-started-with-stm32mp151-stm32mp153-and-stm32mp157-line-hardware-development-stmicroelectronics.pdf.
- [56] M. C. Limited, «MCL,» [En línea]. Available: <https://www.mclpcb.com/blog/pcb-trace-width-vs-current-table/>. [Último acceso: 25 05 2025].
- [57] STMicroelectronis, «Youtube,» [En línea]. Available: <https://www.youtube.com/watch?v=Ot8B30tc-3E>.

- [58] «Analog Devices,» Analog Devices, [En línea]. Available:
<https://www.analog.com/media/en/technical-documentation/data-sheets/ltc7821.pdf>.
- [59] «ANALOG DEVICES,» ANALOG DEVICES, [En línea]. Available:
<https://www.analog.com/media/en/technical-documentation/data-sheets/lt8609-8609a-8609b.pdf>.
- [60] «ANALOG DEVICES,» ANALOG DEVICES, [En línea]. Available:
<https://www.analog.com/media/en/technical-documentation/data-sheets/lt8610.pdf>.
- [61] samtec, «Digikey,» Digikey, [En línea]. Available:
<https://www.digikey.com.mx/es/resources/conversion-calculators/conversion-calculator-pcb-trace-width>.
- [62] I. E. Ltd, «UNMANNED SYSTEMS TECHNOLOGY,» [En línea]. Available:
<https://www.unmannedsystemstechnology.com/company/intelligent-energy/>.
- [63] J. Flores, «National Geographic España,» National Geographic, 15 12 2022. [En línea]. Available:
https://www.nationalgeographic.com.es/ciencia/que-es-5g-y-como-nos-cambiara-vida_14449.
- [64] «ARDUPILOT,» [En línea]. Available: <https://ardupilot.org/copter/docs/common-pixhawk-overview.html>.
- [65] «PX4,» [En línea]. Available: https://docs.px4.io/main/en/companion_computer/pixhawk_rpi.html.
- [66] «ROBU.IN,» 16 12 2020. [En línea]. Available: <https://robu.in/selecting-quadcopter-motor-a-detailed-guide-2021/>.
- [67] C. Rees, «UNMANNED SYSTEMS TECHNOLOGY,» 12 12 2017. [En línea]. Available:
<https://www.unmannedsystemstechnology.com/2017/12/selecting-drone-propeller-blades-considerations/>.
- [68] «u-blox,» ublox, [En línea]. Available: <https://www.u-blox.com/en/product/neo-m9n-module>.
- [69] «GREPOW,» GREPOW, [En línea]. Available: <https://www.grepow.com/uav/delivery-drone.html>.
- [70] «Bmasdigital,» B+ Digital S. de R. L., [En línea]. Available:
<https://www.bmasdigital.com/amplificadores/bandas-telefonía-celular-en-mexico>.
- [71] e. a. Cecilia Bohler, «On the complexity of higher order abstract Voronoi diagrams,» 2015.
- [72] e. a. Piotr Jurkiewicz, «Empirical Time Complexity of Generic Dijkstra Algorithm,» 2021.

- [73] «Sensible 4,» Sensible 4 Oy, 21 07 2021. [En línea]. Available:
<https://sensible4.fi/technology/articles/imu-and-autonomous-vehicles/>.
- [74] «MICROCHIP,» 09 11 2023. [En línea]. Available:
<https://developerhelp.microchip.com/xwiki/bin/view/products/amplifiers-linear/operational-amplifier-ics/introduction/>.
- [75] «Eaton,» Eaton, 04 2018. [En línea]. Available:
<https://www.eaton.com/content/dam/eaton/products/electronic-components/resources/product-data/eaton-equiv-spicesimple-dr125.pdf>.
- [76] «TY-COMPAS,» Taiyo Yuden, [En línea]. Available:
<https://ds.yuden.co.jp/TYCOMPAS/ut/detail?pn=LSRNJ10165GL100MNY&u=M>.
- [77] «TY-COMPAS,» Taiyo Yuden, [En línea]. Available:
<https://ds.yuden.co.jp/TYCOMPAS/ut/detail?pn=LSRNJ12575GL220MMY&u=M>.
- [78] L. WELLPCB TECH CO., «Linkedin,» 22 11 2022. [En línea]. Available:
https://www.linkedin.com/pulse/pcb-temperature-guide-how-manage-high-temperatures-wellpcbptyltd?trk=organization_guest_main-feed-card_feed-article-content.
- [79] SUNNYSKY, «SUNNYSKYUSA,» SUNNYSKY, [En línea]. Available:
<https://sunnyskyusa.com/products/x4212s?srsltid=AfmBOoodUgRYJSAIMeOFuME-aRe2GXaDqPOGafmRJmbbhUzjJc3TSYOC>. [Último acceso: 2022].