



**UNIVERSIDAD TECNOLÓGICA ISRAEL**

**TRABAJO DE TITULACIÓN EN OPCIÓN AL GRADO DE:**

**INGENIERO EN ELECTRÓNICA DIGITAL Y  
TELECOMUNICACIONES**

**TEMA:** SISTEMA DE CONTROL Y OPTIMIZACIÓN DE ENERGÍA PARA DRONES  
CUADRICÓPTEROS DURANTE VUELO ESTACIONARIO

**AUTOR:** RAMIREZ GUAÑA ALEX RODRIGO

**TUTOR TÉCNICO:** Mg. CORTIJO LEYVA RENÉ ERNESTO

**TUTOR ACADÉMICO:** Mg. MORALES ARÉVALO FLAVIO DAVID

**QUITO, ECUADOR**

**2020**

# UNIVERSIDAD TECNOLÓGICA ISRAEL

## DECLARACIÓN DE AUTORÍA

Yo, Alex Rodrigo Ramírez Guaña, declaro bajo juramento que los contenidos de éste Trabajo de Titulación son absolutamente originales, auténticos de exclusiva responsabilidad legal y académica del autor, mismos que no han sido presentados en alguna calificación profesional; y que se han tomado en cuenta todas las referencias bibliográficas que se incluyen en este documento.

Quito D.M, Marzo 2020

.....  
Tlgo. Alex Rodrigo Ramírez Guaña.  
CI: 171583237-2

# UNIVERSIDAD TECNOLÓGICA ISRAEL

## CERTIFICACIÓN DEL TUTOR

En mi calidad de tutor del trabajo de titulación certifico:

Que el trabajo de titulación “SISTEMA DE CONTROL Y OPTIMIZACIÓN DE ENERGÍA PARA DRONES CUADRICÓPTEROS DURANTE VUELO ESTACIONARIO”, presentado por el Tlgo. Alex Rodrigo Ramírez Guaña, estudiante de la carrera de Electrónica Digital y Telecomunicaciones, reúne los requisitos y méritos suficientes para ser sometido a la evaluación del Tribunal de Grado, que se designe, para su correspondiente estudio y calificación.

Quito D.M, Marzo de 2020

TUTOR ACADÉMICO

.....

Ing. Flavio Morales Arévalo, Mg.

# UNIVERSIDAD TECNOLÓGICA ISRAEL

## APROBACIÓN DEL TUTOR

En mi calidad de tutor del componente práctico certifico:

Que el trabajo de titulación “SISTEMA DE CONTROL Y OPTIMIZACIÓN DE ENERGÍA PARA DRONES CUADRICÓPTEROS DURANTE VUELO ESTACIONARIO”, presentado por el Tlgo. Alex Rodrigo Ramírez Guaña, estudiante de la carrera de Electrónica Digital y Telecomunicaciones, reúne los requisitos y méritos suficientes para ser sometido a la evaluación del Tribunal de Grado, que se designe, para su correspondiente estudio y calificación.

Quito D.M, Marzo de 2020

TUTOR TÉCNICO

.....

Ing. René Ernesto Cortijo Leyva, Mg.

## **AGRADECIMIENTO**

Al Padre del Cielo por darme la oportunidad de nacer en este hermoso país, regalarme a mis padres y a toda mi linda familia. Un agradecimiento muy especial a mi querida esposa Raquel, mis hijos Alan y Alejandro, quienes han sido un pilar fundamental en la consecución de esta etapa.

A todos los compañeros de aula, que con sus anécdotas, experiencia y conocimientos hicieron de este paso por la Universidad Israel, uno de los mejores recuerdos de compañerismo, respaldo y amistad.

A nuestros queridos docentes que con su conocimiento, carisma y comprensión lograron hacer suyo nuestro futuro, con compromiso social, trabajo en equipo y rigor académico, han formado profesionales de excelencia, éticos, solidarios y consientes para el beneficio de la sociedad.

Amigos, entre otros, muchas gracias.

**ALEX RODRIGO**

## **DEDICATORIA**

Esta tesis va dedicada a mi hijo Alan Rodrigo, que ha sido participe de alegrías, vivencias, anécdotas, testigo de lazos solidos de amistad y hermandad que uno puede ir formando en la vida si trabajas con sinceridad, honestidad, compromiso, responsabilidad y sobre todo con un inmenso amor por hacer el bien al prójimo.

Decirte querido hijo, que el camino puede parecer difícil, pero cuando crees y tienes fe en ti, los sueños los puedes hacer realidad.

**ALEX RODRIGO**

## TABLA DE CONTENIDO

DECLARACIÓN DE AUTORÍA .....	ii
CERTIFICACIÓN DEL TUTOR.....	iii
APROBACIÓN DEL TUTOR.....	iv
AGRADECIMIENTO .....	v
DEDICATORIA.....	vi
TABLA DE CONTENIDO .....	vii
LISTA DE FIGURAS .....	xii
LISTA DE TABLAS.....	xvi
RESUMEN.....	xvii
ABSTRACT .....	xviii
INTRODUCCIÓN.....	1
Planteamiento del problema.....	2
Justificación del problema.....	3
Objetivos .....	4
Objetivo general .....	4
Objetivos específicos.....	4
Alcance.....	4
Descripción de los capítulos.....	5
CAPITULO 1 .....	7
FUNDAMENTACIÓN TEÓRICA .....	7
1.1 Terminología relacionada .....	7
1.2 Generalidades UAV's .....	8
1.3 Clasificación .....	9
1.3.1 Por el peso del despegue .....	9

1.3.2	Por el tipo de misión o actividad .....	10
1.3.3	Por su alcance.....	10
1.3.4	Por el tipo de despegue.....	11
1.4	Multirrotores .....	12
1.4.1	Ventajas.....	13
1.4.2	Desventajas.....	13
1.5	Cuadricóptero o quadrotor .....	13
1.5.1	Definición.....	13
1.5.2	Antecedentes .....	14
1.5.3	Aplicaciones .....	17
1.5.4	Características principales.....	17
1.5.5	Disposición física del drone .....	18
1.5.6	Movimientos básicos de desplazamiento .....	19
1.5.7	Modos de vuelo .....	20
1.5.8	Vuelo estacionario o estático.....	21
1.5.9	Arquitectura general de un drone cuadricóptero .....	22
1.6	Cuadricópteros como plataforma de investigación.....	24
1.6.1	Hummingbird de Ascending Technologies.....	25
1.6.2	Draganflyer X4.....	26
1.6.3	DIY Drones Ardupilot.....	27
1.7	Sistemas de energía para drone actuales .....	28
1.8	Drone cuadricóptero F450-APM 2.8 de Ardupilot para proyecto .....	29
1.8.1	Características del drone cuadricóptero F450-APM 2.8 de Ardupilot.....	30
1.8.2	Características de la batería.....	31
1.9	Procesos de control y optimización energética.....	31
1.9.1	Sistema de control de energía por demanda.....	32
1.9.2	Sistema de optimización de energía .....	34



CAPITULO 2 .....	35
MARCO METODOLÓGICO .....	35
2.1    Introducción .....	35
2.2    Etapas del proyecto .....	35
2.2.1    Investigación .....	35
2.2.2    Desarrollo .....	35
2.2.3    Diseño.....	35
2.2.4    Implementación.....	36
CAPITULO 3 .....	37
PROPUESTA .....	37
3.1    Propuesta de solución .....	37
3.2    Diseño de solución.....	38
3.3    Definición de estructura para vuelo estacionario del dron.....	39
3.4    Acondicionamiento de señales.....	40
3.4.1    Señal de voltaje de alimentación al Arduino NANO .....	41
3.4.2    Señal de corriente hacia Arduino NANO.....	41
3.5    Arquitectura del sistema de medición de datos de consumo online .....	41
3.5.1    Diagrama circuital para el sistema de medición datos de consumo .....	42
3.5.2    Descripción del sistema de adquisición de datos de consumo .....	43
3.5.3    Calibración de sensores .....	44
3.5.4    Diagrama de flujo del algoritmo de medición a implementar.....	46
3.5.5    Descripción del algoritmo propuesto para etapa de medición .....	48
3.5.6    Descripción del algoritmo propuesto para etapa de medición .....	50
3.6    Arquitectura del sistema de control y optimización de energía.....	51
3.6.1    Diagrama circuital del sistema de control .....	59
3.6.2    Descripción del diagrama circuital del sistema de control.....	60
3.6.3    Dispositivos electrónicos principales de la placa de control.....	63

3.6.4	Diagrama de flujo para algoritmo de control y optimización a implementar ...	67
3.6.5	Descripción del algoritmo propuesto para etapa de control .....	69
3.6.6	Descripción del algoritmo propuesto para etapa de control .....	71
3.6.7	Análisis de estimación de optimización teórica de la propuesta.....	72
3.7	Propuesta de uso de software especializados.....	74
3.7.1	Software Arduino IDE .....	74
3.7.2	LabView (Laboratory Virtual Instrument Engineering Workbench).....	74
3.7.3	Software Mission Planner - Ardupilot .....	75
3.7.4	MatLab® de MathWorks .....	76
3.8	Análisis económico del proyecto .....	76
3.8.1	Costo estimado de la placa de sistema de medición de datos de consumo .....	76
3.8.2	Costo de la placa de sistema de control y optimización.....	77
3.8.3	Costo estimado de recursos materiales, humanos y varios. ....	78
3.9	Ventajas del sistema.....	78
CAPITULO 4 .....		79
IMPLEMENTACIÓN .....		79
4.1	Desarrollo.....	79
4.2	Implementación .....	79
4.2.1	Construcción del sistema de medición de datos de consumo.....	79
4.2.2	Implementación de etapa de hardware .....	79
4.2.3	Implementación de la etapa de software .....	83
4.2.4	Construcción del sistema de control y optimización de energía .....	91
4.2.5	Implementación de etapa de hardware .....	91
4.2.6	Implementación de la etapa de software .....	96
4.3	Pruebas de funcionamiento .....	106
4.3.1	Sistema de medición de datos de consumo .....	106
4.3.2	Sistema de control y optimización de energía.....	108

4.4	Análisis de resultados .....	109
4.4.1	Sistema de control y optimización de energía.....	110
4.4.2	Curvas obtenidas con el sistema de medición de datos de consumo .....	120
4.4.3	Análisis final del sistema implementado.....	125
	CONCLUSIONES.....	126
	RECOMENDACIONES .....	127
	REFERENCIAS BIBLIOGRÁFICAS .....	128
	ANEXOS.....	132

## LISTA DE FIGURAS

Figura 1. 1 Primer UAV de alas planas .....	8
Figura 1. 2 General Atomics MQ-1 Predator .....	9
Figura 1. 3 Drones multirotores .....	13
Figura 1. 4 Primer dron construido- siglo 20 .....	14
Figura 1. 5 Aeronave de ala giratoria Boing V22 Osprey .....	15
Figura 1. 6 Dron cuadricóptero Parrot AR. Dron 2.0 .....	16
Figura 1. 7 Nano cuadricópteros en arreglos lineales .....	16
Figura 1. 8 Formatos básicos de cuadricópteros .....	18
Figura 1. 9 Movimientos básicos del cuadricóptero .....	19
Figura 1. 10 Movimiento estacionario o estático del cuadricóptero .....	22
Figura 1. 11 Arquitectura genérica del cuadricóptero .....	22
Figura 1. 12 Dron cuadricóptero Hummingbird Asc Tec .....	25
Figura 1. 13 Dron cuadricóptero Draganflyer 4X .....	26
Figura 1. 14 Dron cuadricóptero .....	27
Figura 1. 15 Dron cuadricóptero F450-APM 2.8 .....	29
Figura 1. 16 Sistema básico de control.....	31
Figura 1. 17 Proceso de control de lazo cerrado .....	33
Figura 3. 1 Estructura 1 de vuelo estacionario .....	39
Figura 3. 2 Estructura 2 de vuelo estacionario .....	40
Figura 3. 3 Estructura final de vuelo estacionario .....	40
Figura 3. 4 Circuito de acondicionamiento para señales de entrada a Arduino .....	41
Figura 3. 5 Arquitectura del sistema de medición de consumos .....	42
Figura 3. 6 Diagrama circuital para el sistema de medición .....	42
Figura 3. 7 Pruebas y calibración de sensores de corriente .....	44
Figura 3. 8 Diagrama de flujo del sistema de medición en placa Arduino.....	46
Figura 3. 9 Diagrama de flujo de sistema de medición en LabView.....	47
Figura 3. 10 Algoritmo de parada en LabView .....	48
Figura 3. 11 Diagrama de bloques del sistema de medición - placa Arduino. ....	49
Figura 3. 12 Diagrama de bloques de sistema de medición - LabView .....	50
Figura 3. 13 Arquitectura del sistema de control y optimización propuesto.....	51

Figura 3. 14 Conexión inicial del módulo RF al APM 2.8.....	52
Figura 3. 15 Descripción de pines de Receptor FS-IA6 .....	52
Figura 3. 16 Módulo GPS de cuadricóptero en estudio.....	53
Figura 3. 17 Conexión original del módulo GPS al APM 2.8.....	54
Figura 3. 18 Módulo de iluminación del drone .....	55
Figura 3. 19 Conexiones módulo de iluminación.....	55
Figura 3. 20 Conexión original del transmisor del módulo de telemetría al APM 2.8.....	56
Figura 3. 21 Conexión de receptor del módulo de telemetría al PC o Smartphone .....	56
Figura 3. 22 Conexión del transmisor Telemetría a APM 2.8 .....	57
Figura 3. 23 Sistema FPV, cámara y trasmisor .....	58
Figura 3. 24 Sistema FPV montado en drone.....	58
Figura 3. 25 Diagrama circuital del sistema de control y optimización - Proteus.....	60
Figura 3. 26 Circuito de aislamiento entre lógica y control .....	61
Figura 3. 27 Circuito para control de modulo .....	62
Figura 3. 28 Opto acoplador 4N25 .....	64
Figura 3. 29 Transistor: 2SA715 .....	64
Figura 3. 30 Sensor de corriente ACS-712 5 A - 30 A .....	65
Figura 3. 31 Tarjeta Arduino NANO .....	66
Figura 3. 32 Diagrama de flujo del sistema de control y optimización de energía .....	67
Figura 3. 33 Diagrama de bloques – sistema de control y optimización.....	68
Figura 3. 34 Diagrama de bloques - sistema control y optimización .....	69
Figura 3. 35 Diagrama de flujo del sistema de control y optimización de energía .....	70
Figura 3. 36 Diagrama de bloques – sistema de control y optimización LabView.....	71
Figura 3. 37 Software Arduino IDE .....	74
Figura 3. 38 Software LabView .....	75
Figura 3. 39 Software Mission Planner – Ardupilot .....	75
Figura 3. 40 Software MatLab de MathWorks .....	76
Figura 4. 1 Diagrama circuital de medición creado en software Proteus .....	80
Figura 4. 2 Diseño PCB en software proteus - sistema de medición.....	80
Figura 4. 3 Circuito impreso a doble lado - Sistema de medición .....	81
Figura 4. 4 Peso de la tarjeta electrónica de medición .....	81
Figura 4. 5 Tarjeta electrónica creada e instalada en el drone previo a mediciones .....	82
Figura 4. 6 Selección de placa Arduino a utilizar.....	83

Figura 4. 7 Definición de variables y comunicación serial Arduino - PC.....	84
Figura 4. 8 Lectura y acondicionamiento de las señales de corriente y voltaje .....	84
Figura 4. 9 Toma de muestras de señal a la entrada del microcontrolador. ....	85
Figura 4. 10 Creación y codificación de trama enviada al Software LabView. ....	86
Figura 4. 11 Etapa de inicio de sesión y recepción de datos en LabView.....	87
Figura 4. 12 Etapa de decodificación de datos en LabView .....	88
Figura 4. 13 Etapa de verificación del dato obtenido para visualización .....	88
Figura 4. 14 Interfaz gráfica HMI, para visualización de medidas .....	89
Figura 4. 15 Interfaz gráfica HMI, estado de descarga de la batería.....	89
Figura 4. 16 Asignación de atributos a datos para registro .....	90
Figura 4. 17 Etapa de almacenamiento de datos de medida.....	91
Figura 4. 18 Directorio raíz para almacenamiento de hoja de calculo .....	91
Figura 4. 19 Diagrama circuital del sistema de control y optimización de energía .....	92
Figura 4. 20 Diseño PCB de control en software Proteus .....	93
Figura 4. 21 Circuito impreso de sistema de sistema de control .....	94
Figura 4. 22 Placa de sistema de control y optimización ensamblada .....	94
Figura 4. 23 Tarjeta electrónica de control instalada en el Drone F450-APM 2.8 .....	95
Figura 4. 24 Selección de placa Arduino NANO .....	96
Figura 4. 25 Definición de variables de control y comunicación serial Arduino - PC.....	97
Figura 4. 26 Etapa de medición y conformación de trama.....	98
Figura 4. 27 Acondicionamiento de la señal de voltaje de la batería .....	99
Figura 4. 28 Algoritmo de control y optimización de energía .....	100
Figura 4. 29 Curva de descarga de batería sin algoritmo de control y optimización .....	101
Figura 4. 30 Etapa de inicio de sesión y recepción de datos LabView-Arduino NANO ..	102
Figura 4. 31 Etapa de decodificación de datos en LabView .....	103
Figura 4. 32 Etapa de verificación del dato obtenido para visualización HMI.....	103
Figura 4. 33 Interfaz HMI, para visualización de medidas .....	104
Figura 4. 34 Asignación de atributos de dato dinámico para registro .....	104
Figura 4. 35 Generación de hoja de cálculo, con datos medidos.....	105
Figura 4. 36 Directorio y datos recolectados en hoja de cálculo .....	105
Figura 4. 37 Pruebas de funcionamiento de sistema de medición.....	106
Figura 4. 38 Interfaz HMI online de sistema de medición (rotores y batería) .....	107
Figura 4. 39 Directorio y datos recolectados en una hoja de cálculo .....	107
Figura 4. 40 Pruebas de funcionamiento de sistema de medición.....	108

Figura 4. 41 Interfaz HMI online de sistema de control y optimización.....	108
Figura 4. 42 Directorio y datos recolectados en hoja de calculo.....	109
Figura 4. 43 Voltaje batería en vuelo autónomo controlado .....	110
Figura 4. 44 Corriente de batería en vuelo autónomo controlado .....	111
Figura 4. 45 Potencia batería en vuelo autónomo controlado .....	112
Figura 4. 46 Voltaje de batería en vuelo sin control autónomo.....	113
Figura 4. 47 Corriente de batería en vuelo sin control autónomo .....	114
Figura 4. 48 Potencia batería en vuelo sin control automático.....	115
Figura 4. 49 Voltaje de batería en vuelo con y sin control.....	116
Figura 4. 50 Corriente de batería en vuelo con y sin control.....	117
Figura 4. 51 Potencia batería en vuelo con y sin control.....	118
Figura 4. 52 Corriente del motor 1 en vuelo sin control.....	120
Figura 4. 53 Corriente del motor 2 en vuelo sin control.....	120
Figura 4. 54 Corriente del motor 3 en vuelo sin control.....	121
Figura 4. 55 Corriente del motor 4 en vuelo sin control.....	121
Figura 4. 56 Corriente total de rotores en vuelo sin control .....	122
Figura 4. 57 Corriente de batería vs. Corriente total de rotores .....	123
Figura 4. 58 Corriente de todos rotores .....	123

## LISTA DE TABLAS

Tabla 1. 1 Clasificación de los UAV's según la OTAN .....	10
Tabla 1. 2 Clasificación de los UAV's por categorías .....	11
Tabla 1. 3 Comparativa de sistemas de energía en drones Clase I.....	28
Tabla 1. 4 Comparativa de sistemas de energía en drones Clase II y III.....	29
Tabla 1. 5 Características del drone cuadricóptero.....	30
Tabla 1. 6 Módulos componentes del drone cuadricóptero.....	30
Tabla 1. 7 Características de la batería LIPO de Drone F450-APM 2.8 .....	31
Tabla 3. 1 Niveles de voltaje de entrada y salida del Arduino NANO.....	45
Tabla 3. 2 Consumos de corriente estimados por módulo.....	72
Tabla 3. 3 Estimación teórica de optimización.....	73
Tabla 3. 4 Costo estimado de materiales para sistema de medición.....	77
Tabla 3. 5 Costo estimado de materiales de sistema de control y optimización .....	77
Tabla 3. 6 Costo estimado de recursos humanos, materiales y varios.....	78
Tabla 4. 1 Materiales utilizados en la placa de medición.....	82
Tabla 4. 2. Materiales, elementos y dispositivos de la tarjeta de control y optimización ...	95
Tabla 4. 3 Parámetros de consumo de corriente del Drone F450-APM 2.8 Ardupilot .....	118
Tabla 4. 4 Parámetros de consumo de potencia de Drone F450-APM 2.8 Ardupilot.....	119
Tabla 4. 5 Tiempo de vuelo de Drone F450-APM 2.8 Ardupilot .....	119
Tabla 4. 6 Corriente consumida en promedio por cada rotor del drone .....	124
Tabla 4. 7 Corriente consumida promedio de la batería.....	124
Tabla 4. 8 Consumo de corriente real por módulo .....	125
Tabla 4. 9 Análisis final del sistema.....	125



## **RESUMEN**

En el presente proyecto de investigación se desarrolla un sistema de control y optimización de energía para drones cuadricópteros de uso aficionado durante vuelo estacionario, se basa principalmente en determinar a través de mediciones y cálculos teóricos el consumo de corriente de un drone cuadricóptero (por módulo componente), para lo cual se diseñará un sistema de medición y adquisición de datos de consumo online a través de una tarjeta electrónica de desarrollo Arduino NANO (desmontable) para su análisis previo. Posterior a la determinación del consumo total de corriente del cuadricóptero, a través de la teoría clásica de control y placa Arduino NANO, se genera el sistema de control automático de encendido y apagado de los módulos componentes del drone, para lograr la optimización de energía que permita un tiempo de vuelo adicional, que puede orientarse para finalizar una misión con un aterrizaje controlado y precautelando la integridad del drone.

Para el desarrollo de éste proyecto se utilizó un drone cuadricóptero F450-APM 2.8 de Ardupilot, equipo de costo accesible, compatible con múltiples entornos de programación y flexible con plataformas de estaciones terrenas de control de drones, con un firmware de código abierto, robusto e idóneo para desarrollar proyectos de investigación en el área de los UAV's.

Palabras Claves: UAV, drone, cuadricóptero, APM, HMI.

## **ABSTRACT**

This research project develops an energy control and optimization system for amateur quadcopter drones during stationary flight, mainly based on determining through theoretical measurements and calculations the current consumption of a quadcopter drone (per component module), for which a system of measurement and acquisition of consumer data will be designed through an Arduino NANO (datachable) electronic development card for prior analysis. Following the determination of the total current consumption of the quadcopter through the classical control theory and an Arduino NANO board, the automatic on/off control system of the drone component modules is generated, to achieve the energy optimization that allows an additional flight time, which can be oriented to finish a mission with a controlled landing and precautioning the integrity of the drone.

For the development of this project, a quadcopter F450-APM 2.8 drone from Ardupilot, accesible cost equipment, compatible with multiple programming environments and flexible with drone control earth station platforms, with an open source firmware, robust and ideal for developing research projects in the area of UAV's.

Keywords: UAV's, drone, quadcopter, APM, HMI.

## INTRODUCCIÓN

### **Antecedentes de situación objeto de estudio.**

Realizada la investigación respectiva en bibliotecas digitales y consultas a profesionales afines al área de estudio e investigación, se encontró que Carvajal Hoyos, C.A., & Susa Velandia J, S. (2018), Rojas Vásquez, M.S., M.S, & Tafur, J.L (2017), Chamorro Hernández, W.O.,& Medina Mora, J.L. (2013) y Alvear Chimarro, J.O.,& Pavón Mena E-A (2017), realizaron investigaciones relacionadas con ésta área de estudio, brindando aportes importantes que permitieron desarrollar con mayor claridad la presente investigación. Las investigaciones permitieron conocer más a fondo la estructura, funcionalidad, lógica, complejidad, operatividad y aplicabilidad de los UAV's de uso aficionado en actividades y servicios civiles.

Carvajal Hoyos, C.A., & Susa Velandia J, S., investigadores que desarrollaron un módulo energético y de control de temperatura de la Sonda Sabio Caldas II, que es un vehículo estratosférico, que añade baterías de respaldo alimentadas con paneles solares, para su geo-ubicación y rescate una vez finalizada la misión, ventajas que no poseía la sonda antecesora (Carvajal Hoyos & Susa Velandia, 2018). El conocimiento desarrollado en ésta investigación permitió conocer la estructura modular de este tipo de vehículo aéreo no tripulado y determinar que la variable energía en los UAV's, limita su tiempo de funcionamiento, adicional permitió definir la propuesta de solución al proyecto planteado, con un tipo de control y optimización de energía aplicado desde el ámbito de la gestión de la energía entregada por la batería a cada uno de los módulos del cuadricóptero.

Rojas Vásquez, M.S., M.S, & Tafur, J.L, su proyecto se basó en el desarrollo e implementación de una plataforma aérea con software flexible la cual permite realizar pruebas con diferentes algoritmos de vuelo con el fin de lograr la estabilidad del dron. La citada investigación permitió conocer el principio de funcionamiento, así como las distintas plataformas de control de vuelo remotas con software libre y determinar la importancia de

desarrollar una etapa de medición previa a la implementación, a fin de contrastar a través de datos obtenidos antes y después del cumplimiento de la tesis del presente proyecto.

Alvear Chimarro, J.O., & Pavón Mena E-A, el principal objetivo del proyecto es integrar el sistema de control al hexacóptero de monitoreo a través de un controlador de vuelo con software libre, para que sin la necesidad de utilizar un radio control realice misiones de vuelo. La investigación desarrollada brindó aportes relacionados a la integración de los sistemas de medición y sistema de control al dron cuadracóptero en estudio.

Como se puede apreciar cada investigación realizada dentro del campo de los UAV's, son novedosas, cada tema investigado ha servido de referente para el desarrollo de la presente investigación.

Tomando como punto de partida el conocimiento generado y camino trazado por investigadores anteriores, se desarrolló un sistema de control y optimización de energía para drones cuadracópteros durante vuelo estacionario, utilizando el dron cuadracóptero F450-APM 2.8 de Ardupilot de uso aficionado, cuyas características se ajustan para el desarrollo de la presente investigación.

### **Planteamiento del problema**

El que no exista un sistema de control y optimización de energía para drones cuadracópteros durante vuelo estacionario, hace que el tiempo de vuelo de un dron de uso aficionado de categoría mini, sea corto y solamente cumpla el tiempo determinado por su batería, aproximadamente menor a veinte minutos e inclusive en ciertos modelos de dron de categoría micro el tiempo sea aún menor.

Actualmente se presentan las siguientes condiciones generales en el funcionamiento de los cuadracópteros de uso aficionado.

- Descarga rápida de su batería.

- Tiempo de vuelo muy reducido para tareas que requieren un mayor tiempo de funcionamiento.
- Consumos fantasmas de dispositivos que conforman el dron.
- Ausencia de un sistema de control y optimización de energía.

El presente trabajo se centra en buscar una solución al problema de poco tiempo de vuelo de drones en específico de los cuadricópteros, haciendo uso de la teoría básica de control y de las características modulares de los drones cuadricópteros de uso aficionado.

- Los cuadricópteros de uso aficionado están conformados por cuatro servomotores, módulos y sistemas en los cuales el investigador deberá intervenir para alcanzar el objetivo de optimizar energía.
- Independientemente de la aplicación del dron, éste funciona en dos estados, movimiento y estacionario, éste último es el que se aprovechará para el estudio de consumo de energía y determinación de tiempo de vuelo antes como posterior a la implementación.
- El módulo de energía generalmente está diseñado para proporcionar energía al controlador de vuelo, al receptor RC (Radio Control) y sus accesorios (GPS, Radio Telemetría), no a los servomotores para éstos se debe utilizar el propio ESC/BEC (Electronic Speed Controller/Battery Elimination Circuit), para brindar energía a los servos.

### **Justificación del problema.**

El diseño e implementación de una tarjeta electrónica para el control y optimización de energía en drones cuadricópteros de uso aficionado, permitirá a drones o vehículos de exploración comerciales obtener mayores tiempos de funcionamiento.

Con la creación del sistema de control y optimización de energía para drones cuadricópteros de uso aficionado, se tendrán mayores tiempos de vuelo, los cuales traerán beneficios a varios servicios y tareas que se desarrollan con drones en la actualidad, tales como:

- Vigilancia de incendios
- Telemetría
- Inspección de mayores campos de cultivo. (Agricultura)
- Seguridad y vigilancia
- Fotografía, Topografía
- Investigación, exploración
- Aplicaciones de impacto medioambientales
- Inspección termográfica-sector fotovoltaico

## **Objetivos**

### **Objetivo general**

- Desarrollar un sistema de control y optimización de energía para drones cuadricópteros durante vuelo estacionario.

### **Objetivos específicos**

- Establecer un análisis comparativo de los distintos sistemas de energía de drones cuadricópteros de uso aficionado actuales.
- Determinar a través de mediciones y datos teóricos el consumo total de energía del dron en estudio.
- Diseñar una tarjeta electrónica que permita el control y optimización de la energía en el dron cuadricóptero.
- Implementar sistema de control y optimización de energía.
- Establecer resultados del sistema de control y optimización de energía en el equipo en estudio.

## **Alcance**

En el presente proyecto de investigación se implementará un sistema de control y optimización de energía para drones cuadricópteros, durante vuelo estacionario en un

determinado equipo (drone) de uso aficionado, a fin de obtener mayor tiempo de funcionamiento es decir obtener un tiempo de vuelo adicional.

Es importante considerar que muchos equipos o dispositivos de uso aficionado de éste tipo se pierden, se malogran o se estrellan por no disponer de un sistema de control y optimización de energía en su estructura, que permita gestionar y resguardar su integridad con un control de energía adecuado.

Este control permitirá la correspondiente gestión/optimización de la energía en función del porcentaje del nivel de batería del drone, mismo que desencadenará un control de desactivación de módulos internos a fin de obtener un tiempo adicional de vuelo que puede direccionarse a finalizar una misión y resguardar la integridad del dron.

En la actualidad dentro del mundo de los UAV's y vehículos eléctricos, se están desarrollando muchas investigaciones relacionadas con obtener mayores tiempos de autonomía en este tipo de dispositivos de uso aficionado, estudios que van desde dotar de sistemas de combustible, hasta sistemas sofisticados sistemas de transmisión de energía inalámbrica.

### **Descripción de los capítulos**

A continuación se resumen los contenidos de cada capítulo de la presente propuesta de proyecto de investigación.

**Capítulo 1:** Se presenta la fundamentación teórica necesaria para la comprensión del funcionamiento de los drones, arquitectura en general, aplicaciones, clasificación, definiciones, estructura física, módulos componentes, modos de vuelo y características principales del drone a utilizar (cuadricóptero F450-APM 2.8 de Ardupilot). Se describe la fundamentación teórica necesaria en relación a los diferentes sistemas básicos y clásicos de control a partir de los cuales se determinan los esquemas de control y optimización a aplicar en el desarrollo del presente proyecto.

**Capítulo 2:** Sección orientada a desarrollar el marco metodológico propuesto para el presente proyecto.

**Capítulo 3:** Se presenta la propuesta de solución para el desarrollo del sistema de control y optimización de energía para drones cuadricópteros de uso aficionado durante vuelo estacionario dividiéndolo en dos etapas. En una primera etapa se diseña el sistema de medición y adquisición de datos de consumo de energía del drone (voltaje y corriente en puntos críticos), el acondicionamiento de la señal, la calibración de sus componentes (sensores), la programación implementada para la lectura de las señales en los sensores, su almacenamiento, presentación y análisis previo a la implementación.

En una segunda etapa se diseñan las arquitecturas de control a implementar, los diagramas de flujo, diagramas circuitales para el control y optimización de la energía del drone con la capacidad de controlar el apagado de los módulos del drone, durante su vuelo estacionario, se realiza una descripción de los procesos de lectura de las señales, su control, almacenamiento y presentación.

**Capítulo 4:** Se presenta el proceso de construcción, programación e implementación del proyecto, en el que se describen las pruebas de funcionamiento, gráficas de resultados, finalizando con el respectivo análisis de la implementación, conclusiones y recomendaciones que se obtuvieron de la propuesta desarrollada.



# CAPITULO 1

## FUNDAMENTACIÓN TEÓRICA

### 1.1 Terminología relacionada

**Dron/Drone**, la palabra es de origen inglés “drone” que significa abeja macho, sin ser considerado oficialmente dentro del vocablo español, es de uso muy frecuente en el ámbito de los vehículos aéreos no tripulados (Montero, 2017).

**UA (Unmanned Aircraft)**, término que refiere a los vehículos aéreos no tripulados, sin considerar si son autónomos o comandados por algún tipo de radio control (Montero, 2017).

**UAV (Unmanned Aerial Vehicle)**, siglas que corresponden a los vehículos aéreos no tripulados, término poco difundido en el campo civil, es más frecuente en el ámbito militar (Montero, 2017).

**RPA (Remotely Piloted Aircraft)**, siglas que se refieren a vehículos aéreos no tripulados que son comandados desde una estación terrena por un radio control o control remoto (Montero, 2017).

**UAS (Unmanned Aerial System)**, las siglas refieren a un sistema completo de vehículo aéreo no tripulado con mayores prestaciones, generalmente compuesto por la aeronave, un sistema de control, enlace de comunicaciones y estación terrena (González, 2016).

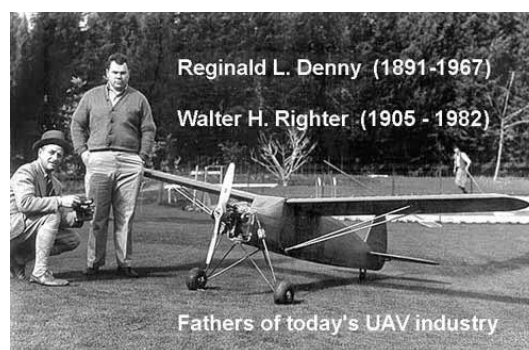
**RPAS (Remotely Piloted Aircraft System)**, siglas que hacen referencia a un sistema UAV controlado de forma remota, que consta al igual que un RPA de un sistema de control, un enlace de comunicación y estación de tierra (González, 2016).

## 1.2 Generalidades UAV's

El estudio y desarrollo de los UAV's, se ubica inmediatamente después de la I Guerra mundial, los cuales fueron ya de utilidad durante la Segunda Guerra mundial con el fin de entrenar a los soldados que operaban cañones antiaéreos. Paralelamente a su vez se empezó a experimentar con proyectiles y bombas aéreas no tripuladas (Mayorga, 2009).

En los siguientes periodos de guerra los aviones biplanos se transformaron en drones, como es el caso de los misiles crucero que se crearon, éstos usaban alas de elevación, un sistema de propulsión por reactor para mantener el vuelo sostenido ("Misil de crucero", 2017).

Durante la segunda guerra mundial (1939 – 1945), EEUU utilizó varios modelos de UAV's radio-controlados, creados por Reginald L. Denny (1891-1967) y Walter H. Reighter (1905-1982). (Bonelli, 2019). Todos los desarrollos tecnológicos militares desarrollados antes, durante y después de estos conflictos bélicos han sido pilares fundamentales de lo que hoy representan los Vehículos Aéreos No tripulados, puesto que ya en esas épocas se vislumbraban más avances como simuladores de vuelo, sistemas submarinos, fotografía aérea, vehículos aeroespaciales, modelos estratosféricos, etc. En la Figura 1.1 se presenta uno de los primeros UAV's de ala fija, utilizados en la primera guerra mundial ("1910s & 1920s - UAV universe", 2019).



**Figura 1. 1 Primer UAV de alas planas (Mundrone, 2020)**

En la Figura 1.2 se observa uno de los UAV Predator de última generación de uso del ejército de los Estados Unidos en la operación Paz duradera en Afganistán y en Irak en el 2003 (EduRed, 2020).



**Figura 1. 2 General Atomics MQ-1 Predator (Parsch, 2020)**

Desde su invención los UAV's, han servido para fines militares y usos especiales, sin embargo su gran desarrollo ha permitido su aplicación en tareas civiles de menor escala como tareas de reconocimiento aéreo, fotografía geo-referenciada, vigilancia y seguridad, monitoreo de incendios o catástrofes naturales, etc., (G, Li, & Chen, 2005).

El Ecuador en el 2008 adquirió seis UAVs de ala fija, dos del modelo *Heron* y cuatro del tipo *Searcher*, de procedencia Israelí, las cuales son utilizadas óptimamente en labores de vigilancia marítima y en operaciones para combatir la piratería, el narcotráfico y contrabando (Chamorro & Medina, 2013).

### **1.3 Clasificación**

Al existir gran variedad de este tipo de dispositivos su clasificación es muy amplia, se consideran varios aspectos de su estructura, de su funcionamiento, o varios de sus detalles como alcance, tiempo de vuelo, resistencia, máxima altitud, peso, despegue, etc.

#### **1.3.1 Por el peso del despegue**

En un intento por estandarizar a estos dispositivos la OTAN (Organización del Tratado del Atlántico Norte) los clasifica de acuerdo al peso máximo de despegue (MTOW – Maximun Take-Off Weight). En base a ésta clasificación en varios países europeos se imparten cursos de licencia de operador de UAV's (Van Blyenburgh, 2014). En la Tabla 1.1, se aprecia la clasificación de los UAV's, según la OTAN.

**Tabla 1. 1 Clasificación de los UAV's según la OTAN**

<b>CLASIFICACIÓN UAV OTAN</b>					
<b>Clase (MTOW)</b>	<b>Categoría</b>	<b>Empleo</b>	<b>Altitud de Operación AGL</b>	<b>Radio de Misión</b>	<b>Ejemplo</b>
<b>CLASE I ≤ 50 Kg</b>	MICRO <2Kg	Táctico (Sección)	Hasta 60 m	5 Km (LOS)	Hummingbird
	MINI 2-20 Kg	Táctico (Compañía)	Hasta 304 m	25 Km (LOS)	Raven
	LIGERO > 20 Kg	Táctico (Batallón)	Hasta 365 m	50 Km (LOS)	Scan Eagle
<b>CLASE II ≤ 600 Kg</b>	TÁCTICO	Táctico (Brigada)	Hasta 3000 m	200 Km (LOS)	Shadow
<b>CLASE III &gt; 600 Kg</b>	MALE (Mediana altitud, larga resistencia)	Operacional	Hasta 13,7 Km	Sin límites (BLOS)	Predator B
	HALE (Gran altitud, larga resistencia)	Estratégico	Hasta 19,8 Km	Sin límites (BLOS)	Predator B
	Combate	Estratégico	Hasta 19,8 Km	Sin límites (BLOS)	Global Hawk

**Autor: (Van Blyenburgh, 2014)**

### 1.3.2 Por el tipo de misión o actividad (Nadales, 2009)

Dentro de este grupo se encuentran seis tipos de UAV's.

1. De simulación de blancos
2. De reconocimiento de terreno
3. De combate
4. De logística
5. De investigación
6. Desarrollo

### 1.3.3 Por su alcance

En este grupo principalmente se hallan los siguientes equipos. (Nadales, 2009)

- a. Handheld, con un alcance menor a 600 metros de altura y con un radio no mayor a 2 km de distancia.
- b. Orbitales (estratosféricos), que transitan en las orbitas LEO a grandes velocidades.

### 1.3.4 Por el tipo de despegue

El despegue es un aspecto muy importante a considerar dentro del funcionamiento del equipo, existen UAV's con el siguiente tipo de despegue:

- a. Vertical
  1. Ala Rotativa - multirrotores (H-Helicópteros, QR-Quadrotors).
  2. Auto-sustentados (dirigibles, globos aerostáticos).
- b. Despegue no vertical
  1. Ala flexible (parapente, ala delta)
  2. Ala fija (aeroplanos)

En la Tabla 1.2, se puede apreciar más clases de UAV's de la rotativa utilizados para aplicaciones civiles, en la actualidad se los puede encontrar en el mercado con mayor presencia en las categorías de mini como micro, con alturas y radios de alcance cortos.

**Tabla 1. 2 Clasificación de los UAV's por categorías**

CATEGORÍA	ACRÓNIMO	ALCANCE (Km)	AUTONOMÍA (horas)	ALTUTUD DE VUELO (m)	CARGA MÁXIMA (Kg)
Micro <250 gr	Micro	<10	<1	250	< 5
Mini < 25 gr	Mini	<10	<2	150 a 300	< 30
Alcance cercano	CR	10 a 30	2 a 4	3000	150
Alcance corto	SR	30 a 70	3 a 6	3000	200
Alcance mediano	MR	70 a 200	6 a 10	5000	1250
Altitud baja	LADP	> 250	0,5 a 1	50 a 9000	350
Altitud media	MRE	> 500	10 a 18	8000	1250
Autonomía alta Altitud baja	LALE	> 500	> 24	3000	<30
Autonomía alta Altitud media	MALE	> 500	24 a 48	14000	1500
Autonomía alta Altitud alta	HALE	> 2000	24 a 48	20000	12000
Combate	UCAV	1500	2	10000	10000
Ofensivo	LETH	300	3 a 4	4000	250
Señuelo	DEC	500	4	5000	250
Estratosférico	STROTO	> 2000	> 48	2000 a 30000	ND
Exo-estratosférico	EXO	ND	ND	> 30000	ND

Fuente: (Nadales, 2009)

El tipo de controlador que se utilice define el alcance del UAV, dentro del grupo de aeroplanos. Los vehículos de alas rotativas (QR-Quadrotors) son muy adecuados para trabajos en interiores, como en estadios y conciertos. Por tal detalle es muy utilizado para la investigación y desarrollo (Barrientos, Del Cerro, San Martín, & Rossi, 2012).

#### 1.4 Multirotores

Los multirotores son naves más ampliamente conocidas y desarrolladas a la que pertenecen los helicópteros y comúnmente es la primera imagen que en nuestro cerebro se forma al hablar de rotores. Un multirotor es una UAV, cuya *sustentación* se produce por la actuación y giro de múltiples rotores ubicados a la misma distancia del centro geométrico del dispositivo. Con distintas combinaciones de velocidad y de sentido de giro, se consigue una alta maniobrabilidad (Ortega, 2015).

La gran diferencia respecto a los drones de ala fija radica en cómo éstos se sustentan en el aire. La misma que es generada a través de sus hélices colocadas paralelas al rotor y sujetadas en el eje vertical del rotor (Hinojosa & Maestre, 2018).

A los multirotores se los puede agrupar en función del número de rotores (motores) que se utilice, partiendo desde los tri-rotores hasta deca-rotores. Cuantos más rotores utilice, mayor es la estabilidad y fuerza de propulsión requerida (Ortega, 2015).

Existen subtipos dependiendo de la cantidad de rotores que lo conformen (Hinojosa & Maestre, 2018).

- a. 3 rotores: tricóptero
- b. 4 rotores: cuadricóptero
- c. 6 rotores: hexacóptero.
- d. 8 rotores: octacóptero.

En la Figura 1.3, se puede observar drones multirotores comerciales, disponibles en el mercado.



Figura 1. 3 Drones multirotores (NovoDrone, 2020)

### 1.4.1 Ventajas

La característica más aprovechada en este tipo de drones multirotores es su estabilidad es decir la capacidad de desarrollar vuelos estáticos, que le permiten desarrollar varios servicios en movimiento y estacionarios.

### 1.4.2 Desventajas

Su tiempo de vuelo está limitado por la gran demanda de energía que requieren los rotores (rotores), es por ello que regularmente se los utiliza para vuelos a poca altura y para carga con pesos ligeros, lo cual reduce aún más su tiempo de vuelo.

El factor meteorológico de la localidad en el que desarrolle su misión es un factor externo a considerar puesto que reduce el desempeño de este tipo de dispositivos. En la presente investigación se ha elegido un drone de ala rotativa de cuatro rotores que se ajusta perfectamente a este proyecto.

## 1.5 Cuadricóptero o quadrotor

### 1.5.1 Definición

El quadrotor o drone cuadricóptero es un UAV, que se encuentra en la categoría de ala rotatoria o multirotor de tipo VTOL (Vertical Take-Off and Landing–Despegue y aterrizaje vertical), compuesto por cuatro rotores con su respectiva hélice sujetados sobre

un bastidor, estructura o chasis en forma +, X o H. Los rotores producen sustentabilidad en el aire permitiendo el ascenso/descenso, vuelo estacionario, vuelo horizontal, desplazamiento mediante la variación de la velocidad de dos los rotores en determinadas combinaciones.

### 1.5.2 Antecedentes

En el siglo 20, los pioneros dentro del campo de los *cuadricópteros* fueron los franceses Charles Richet y Louis Bréguet, quienes construyeron un helicóptero tripulado que poseía cuatro alas giratorias de 8.1 m de diámetro, giraba gracias a la transmisión de movimiento de correas y poleas, que venían de un solo motor a combustión ubicado en el centro del cuadricóptero, con una potencia de 50 HP, un peso total de 578 Kg, como se puede apreciar en la Figura.1.4, el primer diseño de un quadrotor (Mayorga, 2009)

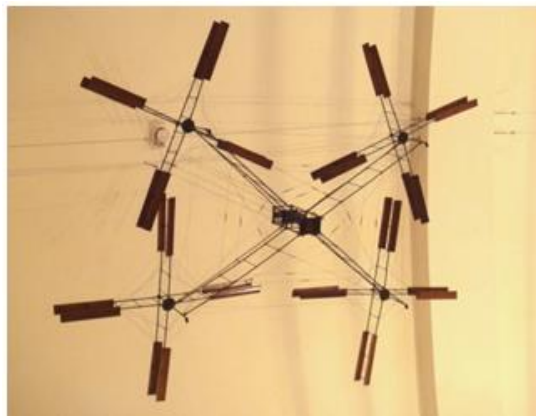


Figura 1. 4 Primer drone construido- siglo 20 (Cardenas & Olarte, 2020)

El modelo de los franceses en el año 1907, logró ascender una altura máxima de 1.5 metros del suelo e inmediatamente aterrizó por varios problemas de estabilidad y control del vuelo.

En lo posterior el estudio y análisis de drones no tripulados se había vuelto una tendencia para el medio académico francés, lo que conllevó a principios de 1922, la presentación de un nuevo prototipo denominado *Corvetawings* modelo A, para uso militar y civil. La principal diferencia con su antecesor es la presencia de dos rotores a combustión, que controlaban la velocidad de los cuatro rotores ubicados en sus extremos, para generar su desplazamiento.



La empresa Bell, en el año 1970 construyó su modelo Boeing V22 Osprey, apostando a las características de los rotores basculares, con los que alcanzó el cielo en 1977 y la confianza de una potencia mundial (USA) para servir a su ejército con la garantía de brindar despegues así como aterrizajes verticales con desplazamientos aéreos como cualquier aeronave de largo alcance. En la Figura 1.5 se puede apreciar la estructura del Boeing V22 Osprey de ala rotatoria utilizado por el ejército de los Estados Unidos.



**Figura 1. 5 Aeronave de ala giratoria Boeing V22 Osprey (Wikiwand, 2020)**

A partir de la segunda década de los 90's, las empresas Bell y Agusta Aerospace Company, desarrollan múltiples aeronaves de similares características como el Agusta Westland AW609, el Moller Skycar el famoso cuadricóptero prototipo a manera de auto volador, quienes se mantuvieron únicamente dentro del campo de la investigación. Paralelamente la NASA desde los años 70's, desarrollaba investigaciones sobre el control de sus aeronaves desde una estación en tierra, lo que buscaba era eliminar el piloto que se requería para atender en los casos de fallas mecánicas durante el vuelo (Velásquez, 2012, pág. 30). Los sistemas aparatosos y complejos por su volumen propiciaron que solo tuvieran aplicación militar en aquella época.

Con el advenimiento de la tecnología, la miniaturización de la electrónica en conjunto con el interés de varias agencias aeroespaciales estatales (USA, Japón, China) y empresas privadas (Parrot, DJI, 3DRobotics, AscTec, QwinOut, KMeI Robotics, etc.), durante el resto de la década de los 90's, se desarrollaron sistemas más ligeros con componentes más versátiles, eficientes y cada vez de menor tamaño tanto en dimensiones como en peso (Dyna, 2016).

En la Figura.1.6, se puede apreciar un drone cuadricóptero Parrot AR.Drone 2.0, de última generación de fabricación norteamericana con un tamaño 77.7 x 38.3 x 12.5 mm, de 380 g de peso total, su plataforma de control compatible con Android y autonomía de vuelo de 36 minutos aproximadamente (Quadcopter Arena, 2018).



**Figura 1. 6 Drone cuadricóptero Parrot AR. Drone 2.0 (Quadcopter Arena, 2018)**

El avance tecnológico resultado de las investigaciones realizadas al comportamiento de animales e insectos voladores permitieron el desarrollo de drones más pequeños con características, estructuras y patrones de vuelo similares a los estudiados, permitiendo la adición de dispositivos electrónicos como: cámaras, sensores, luces, transmisores, receptores de bajo consumo a el fin de dotar de mayores funcionalidades y servicios a los drones cuadricópteros (Lastra, 2017, pág. 32). En la Figura.1.7, se puede apreciar varios mini drones en formación de enjambre, compuesto por varios nano quadrotors volando en formación sincronizada.



**Figura 1. 7 Nano cuadricópteros en arreglos lineales (NewAtlas, 2020)**

### 1.5.3 Aplicaciones

La constante evolución de los drones cuadricópteros (multirotors) ha permitido que se les pueda utilizar en diversas tareas con un costo relativamente bajo dentro de entornos rurales y urbanos. Siendo de gran aplicación en actividades gubernamentales como civiles. Dentro de las múltiples aplicaciones en que pueden ser utilizados están las siguientes:

- a. Transporte aéreo automático de carga
- b. Tareas de búsqueda y rescate
- c. Video vigilancia y seguridad
- d. Monitoreo y reconocimiento aéreo
- e. Polinización agrícola
- f. Exploración de campos
- g. Agricultura de precisión.
- h. Fumigación de campos
- i. Fotografía aérea
- j. Entretenimiento o recreativo
- k. Recolección de datos
- l. Primeros auxilios
- m. Vigilancia de incendios
- n. Telemetría
- o. Inspección de campos de cultivo
- p. Topografía
- q. Investigación y exploración.
- r. Aplicaciones de impacto medioambientales
- s. Inspección termográfica en el sector fotovoltaico
- t. Cinematografía
- u. Geolocalización.

### 1.5.4 Características principales

Este tipo de drones como su palabra lo dice quadrotors o cuadricópteros, disponen de cuatro rotores/motores cada uno con su respectiva hélice, para su desplazamiento se aprovecha la capacidad de variar la velocidad de éstos, así se controla la estabilidad y los distintos movimientos del drone (*Pitch, Roll, Yaw, Throttle*).

Los cuadricópteros comerciales y de uso aficionado que se encuentran en el mercado generalmente pertenecen a la categoría de ala rotativa o multirotor, tipo Handheld a nivel mini y en común tienen las siguientes características:

- Alcance, menor a 10 Km
- Altitud de vuelo, menor a 300m
- Autonomía, menor a 2 horas
- Carga máxima de 5kg
- Capacidad de vuelo, vertical (vuelo estacionario/vuelo horizontal)
- Alta maniobrabilidad

Estas aeronaves (cuadricópteros) al ser de conformación más compleja, son más lentas que los UAV's de ala fija, tienen un tiempo de vuelo más limitado, con una autonomía no mayor a 30 minutos en promedio, con una velocidad aproximada de 60 Km/h ideales para misiones cortas puesto que sus rotores demandan potencia en todo momento por lo que su tiempo de vuelo es corto.

### 1.5.5 Disposición física del dron

Dentro de éstos UAV's, específicamente de ala rotativa o multirotor, tipo cuadricóptero o quadrotor, en la que se centra la presente investigación, existen varios tipos de disposición de la estructura física de éstos drones, en la Figura 1.8, se puede observar dos de las disposiciones más comunes.

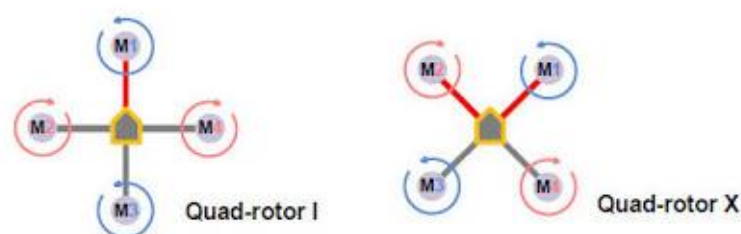


Figura 1. 8 Formatos básicos de cuadricópteros (Aeromodelismofacil, 2020)

Los dos tipos de constelación más comunes son las de Tipo + (plus) y tipo X (cruz), la diferencia entre ambos radica en cómo se ubica el frente del cuadricóptero ubicado en el centro de masa del dispositivo. Para el caso de la izquierda, la parte frontal es el rotor M1, mientras que para la Figura de la derecha la parte frontal del drone está entre los rotores M2 y M1 (Chamorro & Medina, 2013, pág. 8).

### 1.5.6 Movimientos básicos de desplazamiento

Un drone cuadricóptero para mantenerse en vuelo necesita producir un empuje hacia arriba, que lo consigue a través de conducir aire hacia abajo por medio de los rotores y de las hélices de las que dispone en su estructura.

Cada par de hélices gira a una misma velocidad en contra rotación como se puede observar Figura 1.8, puesto que es sumamente importante que el par neto generado por la rotación de las hélices en sentido horario y anti horario sea cero para que el drone cuadricóptero pueda mantener un vuelo estable en el aire una vez haya despegado (Chimarro & Pavón, 2017, pág. 8).

Existen cuatro movimientos básicos que han sido determinados por la investigación aeroespacial, que son muy similares a los designados para el control de satélites. Estos movimientos definen la orientación que tomará el drone cuadricóptero una vez en vuelo, según como se indica en la Figura 1.9, los movimientos toman como base la variación de la velocidad de cada rotor que lo compone.

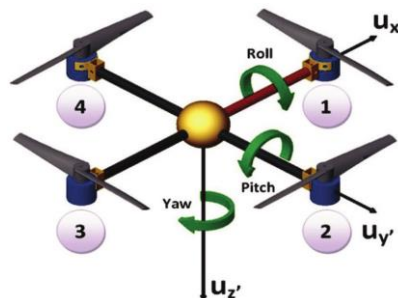


Figura 1. 9 Movimientos básicos del cuadricóptero (Prem, Anand, & Ganesh, 2016)

Los movimientos definidos internacionalmente son (Chimarro & Pavón, 2017, págs. 10,11,12,13) :

- a. *Altitud (throttle)*, movimiento de ascenso o descenso, éste tipo de movimiento está en función de la velocidad angular de todos los rotores en un mismo momento.
- b. *Alabeo (roll)*, movimiento hacia la izquierda o derecha, se consigue variando la velocidad angular de rotores laterales.
- c. *Cabeceo (pitch)*, movimiento hacia adelante o atrás, se consigue variando la velocidad de rotores del frente y posterior.
- d. *Guiñada (yaw)*, es el giro del drone sobre su propio eje, se consigue variando la velocidad angular de determinados rotores.

### 1.5.7 Modos de vuelo (Drones.mx, 2020)

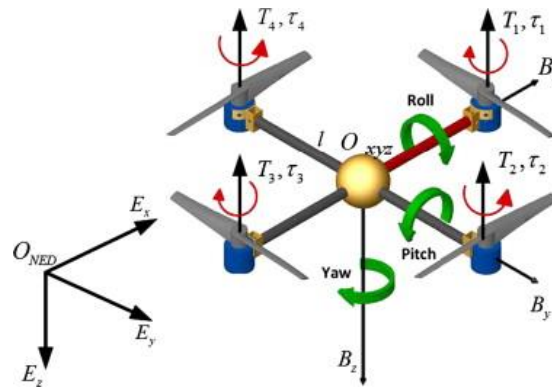
- a. **Modo manual o estándar**, principalmente controlado desde un radio control, éste modo de vuelo permite al piloto del drone, desplazarlo a cualquier dirección, siempre considerando un frente del dispositivo. Muy usado en concursos de carreras de drone a nivel micro.
- b. **Modo manos libres**, en éste modo de vuelo es drone tiene la capacidad de reconocer el norte y sur, dependiendo de la ubicación del piloto. Vuelo que permite auto estabilización y mantener la inercia del drone, lo que permite al drone tratar de nivelarse si se pierde el control.
- c. **Modo vista en primera persona FPV (Fisrt Person Vision)**, el drone q se utilice en este modo de vuelo deberá tener una cámara frontal, con un sistema de transmisión de video vía RF o wifi. Este modo de vuelo da al piloto la sensación de estar sobre el drone y con el control en mano, todo esto es posible si dispone de las gafas de visualización.
- d. **Pilotos automáticos**, generalmente controlados por el sistema de posicionamiento global GPS.

- e. **Modo de mantener altitud**, el drone tiende a mantener una misma altura, en todo su funcionamiento, podrá desplazarse en cualquier dirección pero a una misma altura.
- f. **Modo loiter**, similar desempeño que el modo de mantener altitud, con la diferencia que el operador puede abandonar los controles y el drone se auto estabilizará y mantendrá en la última ubicación registrada.
- g. **Modo estabilizado**, este tipo de vuelo buscará siempre estabilizarse automáticamente posterior a cualquier cambio de dirección, es decir buscará siempre la estabilidad en el punto deseado.
- h. **Modo de regreso a tierra**, característica importante para precautelar el dispositivo, el momento en que el equipo pierda comunicaciones, automáticamente regresa al punto de partida.

Para la presente investigación se utilizará el modo de vuelo estabilizado, en el cual el movimiento básico que desarrolla en drone cuadricóptero es de ascenso/descenso, con el fin de buscar el estado estacionario que se requiere para él análisis. En éste modo de vuelo el cuadricóptero trata de mantener una altura estable, el movimiento básico que permitirá esta condición de vuelo estacionario es el *Throttle* o altitud.

### 1.5.8 Vuelo estacionario o estático

Un movimiento primordial en el despegue y aterrizaje del drone es el *throttle*, o movimiento de altura de ascenso y descenso puesto que se utiliza para controlar el movimiento vertical del cuadricóptero. Un estado estacionario se lograría cuando todos los rotores mantienen la misma velocidad angular y se pueda contra restar los efectos de la gravedad de la tierra a cierta altura. En la Figura 1.10 se puede observar el sentido de giro de los rotores para conseguir el drone despegue. Al aumentar o disminuir la aceleración de los rotores, éste se desplazará hacia arriba o hacia abajo respectivamente (Magnusson, 2014).



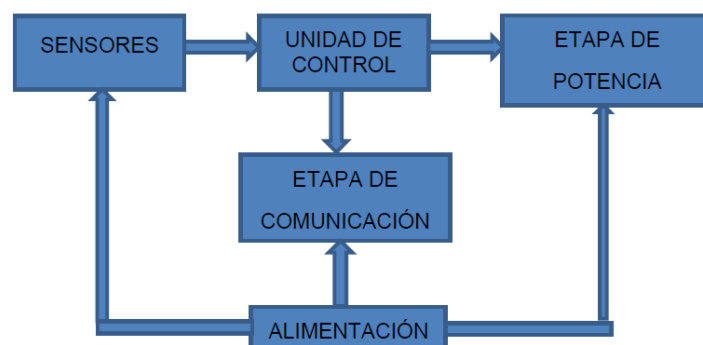
**Figura 1. 10** Movimiento estacionario o estático del cuadricóptero (Weihua & Tiauw, 2012)

Encontrar un estado estacionario o de suspensión perfecto en cualquier ambiente de vuelo estará sujeto a varios factores internos y externos al cuadricóptero, como ejemplo se tiene el desgaste de batería que provoca la pérdida de potencia y por ende un descenso del dron, así como factores meteorológicos viento, lluvia, altura, etc., en exteriores.

Para el presente trabajo se definirá una estructura física para laboratorio que permita simular un vuelo estacionario y que será compensado con el movimiento de ascenso o *Throttle* por medio del piloto desde el radio control.

### 1.5.9 Arquitectura general de un dron cuadricóptero

En la Figura 1.11, se presenta un diagrama de bloques que representa la interrelación entre las unidades fundamentales de las que está conformado un UAV, en este caso un dron cuadricóptero. A continuación se detalla cada uno de ellos.



**Figura 1. 11** Arquitectura genérica del cuadricóptero (Chamorro & Medina, 2013)



### **a. Sensores**

Dispositivos que captan señales externas las convierten en sus equivalentes eléctricos y son recibidas por un microcontrolador para su tratamiento, control y gestión del dispositivo UAV.

1. Acelerómetro, mide la aceleración asociada con la variación de la posición.
2. Giroscopio, mide la aceleración angular.
3. Magnetómetro, es una brújula digital, para la orientación del equipo.
4. Sensores ultrasónicos, son utilizados para el despegue y descenso, así como determina la altura a la que se encuentra de tierra el equipo.
5. Conversores A/D, convierten las señales del medio para ser procesadas por la controladora de vuelo.
6. Microprocesador, es el cerebro del drone, es quien procesa las señales y envía a los rotores del drone.

### **b. Unidad de control**

Microcontrolador dispositivo que es el lugar en donde se aloja el firmware o lógica de funcionamiento del cuadricóptero, generalmente éste dispositivo en la actualidad viene embebido en una sola placa electrónica llamada controladora de vuelo, la que permite configurar diferentes tipos de navegación y control de los recursos como sensores, actuadores y diferentes módulos como de telemetría bidireccional que actualmente se disponen para cuadricópteros de tipo modular y de código abierto.

Los desarrolladores constantemente se encuentran actualizando las versiones de firmware de la tarjeta controladora de vuelo de cada drone comercial, haciéndola cada vez más compatible con otros entornos de programación. Para drones cuadricópteros de tipo modular y de código abierto es común el trabajo con entornos de desarrollo como: Arduino IDE, Programación SDK (Software Development Kit), softwares de estaciones terrenas como APM Planner, HK KCS, soporte multiplataforma en Windows, Mac, Android, Linux.

**c. Etapa de potencia y propulsión**

1. Rotores brushless, de alta eficiencia, con gran capacidad de generar el empuje necesario para elevar al dron.
2. ESC/BEC, son variadores de velocidad con regulador de voltaje, que trabajan en conjunto con los rotores brushless y hélices a fin de permitir el movimiento del dron o mantenerse estabilizado.
3. Hélices, en determinada configuración su giro provoca un empuje hacia abajo para el despegue, ascenso y desplazamiento del dron en cualquier dirección.

**d. Alimentación o sistema de energía**

1. Baterías, en los drones de categoría micro, la fuente de alimentación principalmente son las baterías de polímero de litio, llamadas también LIPO. De gran aplicación en los UAV's de tamaño reducido, alcances cortos y misiones rápidas. Sus principales características son:
  - a. Capacidad de almacenamiento, en el orden de 200 mAh a 20000 mAh.
  - b. Tasa de carga y descarga, que se direccionan con los elementos que demandan altos porcentajes de potencia.
  - c. Peso y dimensiones, está en función del número de celdas dispuestas en su interior.

**e. Etapa de comunicación**

Se refiere a las instrucciones, ordenes o comandos de información que se envían desde el radio control al cuadricóptero, en este tipo de UAV, la comunicación comúnmente se la realiza a través de radio frecuencia RF.

**1.6 Cuadricópteros como plataforma de investigación**

Desde su invención los UAV's, entre ellos los drones cuadricópteros han servido para fines militares y usos especiales, su gran desarrollo ha permitido su aplicación en tareas civiles de menor escala como tareas de reconocimiento aéreo, fotografía, vigilancia, seguridad, monitoreo, etc., (G, Li, & Chen, 2005)

Su amplia aplicación ha hecho que centros de investigación internacional, universidades e instituciones gubernamentales se interesen en conocer su funcionamiento, investigar y crear nuevas técnicas de control para adaptarlo a más servicios civiles en ambientes urbanos y rurales.

Su complejo funcionamiento y conformación completa hace que estos dispositivos no estén al alcance del público en general, de igual forma sus plataformas de control, en mucho de los casos son sistemas de código propietario que los hace menos aprovechables. A continuación se detallan los drones cuadricópteros que han sobresalido y permiten relativa intervención de la mano de los desarrolladores e investigadores.

### 1.6.1 Hummingbird de Ascending Technologies

Drone cuadricóptero de fabricación alemana, su estructura robusta acompañada de accesorios extra ligeros y reemplazables como sus hélices flexibles, permiten al UAV realizar despegues y aterrizajes rápidos con movimientos bruscos en fin de brindar maniobras rápidas durante el vuelo, convirtiéndose en una herramienta perfecta para el desarrollo, el Hummingbird actualmente se valora en \$5000 (Chamorro & Medina, 2013).

En la Figura 1.12, se puede observar al cuadricóptero en mención durante su performance en campo abierto.



Figura 1. 12 Drone cuadricóptero Hummingbird Asc Tec (Niermeyer & Raffler, 2015)

### 1.6.2 Draganflyer X4

Los cuadricópteros Draganflyer X4-P, Draganflyer X4-ES, Draganflyer X6 y Draganflyer Guardián son de uso profesional y direccionados para la recolección de datos en actividades de fotografía aéreas como mapeo, agricultura, topografía, seguridad pública, asistencia en emergencias (Saskatoon, 2020).

Éstos UAV's están contruidos de fibra de carbono con componentes producidos a inyección, posee siete sensores de vuelo que le permiten auto-estabilizarse en el aire, en conjunto con el software de control Draganview, permite al usuario mayor control y estabilidad a fin de concentrarse en obtener las fotografías deseadas.

Draganfly ha redefinido los servicios de los UAV's y está conquistando el mercado de servicios de emergencia con sus nuevos modelos como el que se puede apreciar en la Figura 1.13 (Saskatoon, 2020).



**Figura 1. 13 Drone cuadricóptero Draganflyer 4X (Saskatoon, 2020)**

El cuadricóptero Draganflyer X4-ES es una plataforma aérea de calidad profesional, potente y fácil de volar, diseñada bajo un sistema de código propietario su valor oscila en los \$12000 (Chamorro & Medina, 2013).

### 1.6.3 DIY Drones Ardupilot

Es un proyecto creado por la empresa Ardupilot y representa a una línea completa de drones modulares, que se basa en el desarrollo de su controladora de vuelo Ardupilot Mega 2 (tarjeta Arduino ATmega 128p), que consiste en un sistema de piloto automático completo, confiable, flexible y lo más importante de código abierto que acepta muchos tipos de vehículos aéreos no tripulados como: multirrotores de ala rotatoria, aviones de ala fija, barcos, submarinos, vehículos de exploración (rovers), etc., (ArduPilot, 2019).

En la Figura 1.14, se puede observar un drone cuadricóptero modular de la línea Ardupilot, durante su vuelo en campo.



**Figura 1. 14 Drone cuadricóptero (ArduPilot, 2019)**

El código fuente (firmware) desarrollado para la controladora de vuelo, es constantemente actualizado por una gran comunidad de profesionales, desarrolladores y aficionados, dentro de sus principales características se pueden enumerar las siguientes:

1. Al tratarse de una línea de drones modulares está más al alcance de los investigadores que los dos casos anteriores. Compuesto por una gran variedad de módulos para su conformación y finalidad.
2. La controladora de vuelo APM (Ardupilot Mission), viene actualizada con la última versión de firmware, es compatible con varios entornos de desarrollo IDE, así como con software's de estaciones terrenas para su configuración, control y monitoreo como Mission planner, APM Planner 2, QGroundControl, UgCS entre otras.

3. En éste top de drones, la controladora de vuelo alberga en una sola placa electrónica, a los sistemas de GPS, dispositivos acelerómetros, giroscopios, sensores de nivel, de altura, de presión barométrica, que lo hacen muy ligero y apto para el desarrollo e investigación, puesto que permiten la adición de mayores accesorios dependiendo de su fin.
4. Un drone cuadricóptero completo de ésta línea bordea los \$ 600,00 sin costear gastos de importación y mano de obra.

La presente investigación se desarrollará sobre un drone de tipo modular, con base en la controladora de vuelo APM 2.8 de la empresa Ardupilot.

### 1.7 Sistemas de energía para drone actuales

En la actualidad la principal fuente de energía utilizada a nivel de drones de ala rotatoria, clase I, categoría micro y mini, dentro de la categoría que se encuentra el drone que se utiliza en el presente proyecto, son las baterías de polímero de litio, como se indica en la Tabla 1.3. La principal razón de su uso, es por sus ventajas en tanto a capacidad de corriente y potencia entregada, mientras que su gran desventaja es el peso que añade al dispositivo.

**Tabla 1. 3 Comparativa de sistemas de energía en drones Clase I**

Clase I (<50 Kg) Categoría Micro – Mini – uso civil				
Característica	Hummungbird	DraganFlyer X4	DJ NAZA V2	ArduPilot APM 2.8
Peso	1kg	4kg	8 Kg	1 Kg
Autonomía	20 min	25 min	20 min	10 min
<b>Sistema combustible</b>				
Batería LIPO	X	X	X	X
<b>Celda Fotovoltaica</b>				
<b>Sistema híbrido</b>				

Fuente: Elaborado por el autor

Para el caso de los drones de clase II y III, táctico y combate la tecnología en el sistema de energía tiende hacia lo híbrido, una combinación entre fuentes eléctricas combinadas con sistemas de combustibles fósiles, que se convierten en una fuente de energía alterna para los casos en que se requiera un mayor tiempo de vuelo, generalmente aplicada en el ámbito militar, según la Tabla 1.4.

**Tabla 1. 4 Comparativa de sistemas de energía en drones Clase II y III**

<b>Clase II y III (=&gt;600 Kg) Categoría Táctico – uso militar</b>			
<b>Característica</b>	<b>ScanEagle</b>	<b>Predator MQ</b>	<b>Global Hawk</b>
Peso (cargado)	18 Kg	1020 Kg	10387 Kg
Autonomía	12 horas	35 horas	36 horas
Sistema combustible	X	X	X
Sistema eléctrico	X	X	X
Celda Fotovoltaica			
Sistema híbrido	X	X	X

**Fuente: Elaborado por el autor**

### **1.8 Drone cuadricóptero F450-APM 2.8 de Ardupilot para proyecto**

El Quadrotor o drone cuadricóptero que se presenta en la Figura 1.15, es un drone modular de código abierto de la línea Ardupilot, que se encuentra en la categoría de ala rotativa o multirotor, tipo Handheld a nivel mini, con un alcance menor a 10 Km, una altitud de vuelo menor a 300 m, con una autonomía menor a una hora, carga máxima de 5kg, una capacidad de vuelo vertical y horizontal.



**Figura 1. 15 Drone cuadricóptero F450-APM 2.8 (ArduPilot, 2019)**

El dispositivo F450-APM 2.8, es el dron cuadricóptero que en el presente proyecto se interviene a fin de implementar un sistema de control y optimización de energía para drones cuadricópteros durante vuelo estacionario.

### 1.8.1 Características del dron cuadricóptero F450-APM 2.8 de Ardupilot.

Las principales características del dron cuadricóptero que se va a utilizar para el desarrollo del proyecto se observan en la Tabla 1.5

**Tabla 1. 5 Características del dron cuadricóptero**

<b>CARACTERÍSTICA</b>	<b>VALOR</b>
Peso del equipo	1107 gr.
Alcance	Menor a 10 Km
Autonomía	Aprox. 15 min
Carga máxima	5kg
Capacidad de vuelo	Vuelo vertical y horizontal
Altura mínima de vuelo	5 metros

**Fuente: (Nadales, 2009)**

El cuadricóptero F450-APM 2.8 de Ardupilot, de estructura modular está conformado por los módulos que se describen en la Tabla 1.6

**Tabla 1. 6 Módulos componentes del dron cuadricóptero**

<b>MÓDULOS DRON F450-APM 2.8</b>
Módulo de propulsión (rotores)
Controladora de vuelo APM 2.8
Módulo de Radio Frecuencia RF
Módulo de Sistema de posicionamiento global GPS
Módulo de iluminación
Módulo de sistema FPV o cámara
Módulo de telemetría

**Fuente: Elaborado por el autor**



### 1.8.2 Características de la batería

La batería que en general es la más utilizada por la potencia y corriente que puede brindar es la construida de polímeros de litio, comúnmente llamadas baterías LIPO. Son muy utilizadas en el mundo de los drones de uso civil y aficionado, las características se observan en la Tabla 1.7.

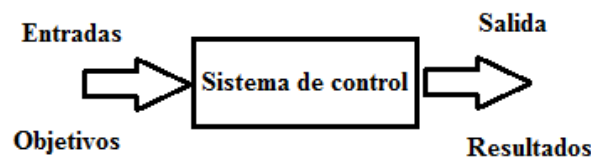
**Tabla 1. 7 Características de la batería LIPO de Drone F450-APM 2.8**

PARAMETRO	ESPECIFICACIÓN
Modelo	Max Force RC Profesional
Capacidad	2200 mAh
Voltaje	11.2 V
Numero de celdas	3 en serie
Taza de descarga max.	30 C
Peso	181 gr

Fuente: Elaborado por el autor

### 1.9 Procesos de control y optimización energética

El constante avance tecnológico en la industria, hace que todo sistema sea controlado con el objetivo de optimizar y brindar el mejor desempeño de los procesos en los sistemas automatizados. Las técnicas y procedimientos han evolucionado que en la actualidad se destacan sistemas que regulan magnitudes de presión, temperatura, flujo, velocidad, seguridad, electrónica, gracias al desarrollo de sofisticados microcontroladores basados en la teoría clásica de control de retroalimentación y análisis de sistemas lineales, como el mostrado en la Figura 1.16, aplicables al campo de la ingeniería aeronáutica militar, en el control de vehículos teledirigidos y robótica a nivel comercial (Carrillo, 2011).



**Figura 1. 16 Sistema básico de control**

Fuente. Elaborado por el autor

### **1.9.1 Sistema de control de energía por demanda**

Un sistema de control de energía por demanda es un dispositivo que actúa sobre una señal o una variable de entrada, que temporalmente apaga cargas eléctricas, manteniendo la demanda de energía (salida del sistema) bajo control. El controlador establece ciclos de operación cuando la demanda alcanza cierto valor o decrece de un valor determinado. Implementar sistemas de control de demanda es una oportunidad de ahorro energético, sin afectar en gran escala el desempeño del sistema a controlar (Juna, 2015).

#### **a. Clasificación de controles por demanda (Juna, 2015).**

1. Controles manuales de encendido y apagado, solamente se obtiene su beneficio si son supervisados, son simples de bajo costo. Ejemplos: Controles de tiempo, fotoceldas, termostatos, sensores.
2. Controles automáticos programables e inteligentes, permiten controles autónomos, comandados por un microcontrolador que se encarga de las operaciones de conexión o desconexión en distintos periodos e instantes de tiempo, adicional integran varios módulos que dan versatilidad y mayor funcionalidad al sistema controlado.

#### **b. Sistemas básicos de control (Escalona & Morillo, 2017)**

1. Sistema de control a lazo abierto, en el que la salida del sistema no es afectada por la señal de entrada, es decir la señal de salida no es realimentada nuevamente a la entrada del sistema, muy similar a la Figura 1.16.
2. Sistema de control a lazo cerrado (control realimentado), el controlador se alimenta con la diferencia de la señal de entrada y una señal retroalimentada proveniente de una etapa de medición o verificación del sistema a fin de reducir errores en el sistema y lograr obtener las salidas deseadas. En la Figura 1.17, se muestra la estructura de un sistema de control de lazo cerrado y se describen sus partes.

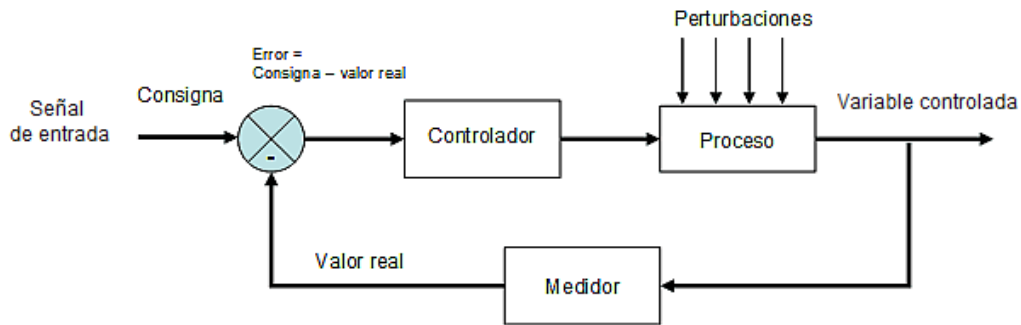


Figura 1. 17 Proceso de control de lazo cerrado (Tecingenieriaclasico, 2020)

1. **Proceso**, etapa de operación progresiva en la cual se realizan acciones o movimientos controlados frente a perturbaciones, que conducen a un resultado o propósito determinado.
2. **Medición**, encargado de captar las señales del proceso final realimentarlas a fin de reducir el error del sistema.
3. **Proceso**, elemento que se encuentra en contacto físico con el proceso y lo comanda el controlador y muestra la porción de acción de control.
4. **Controlador**, permite cumplir al proceso su objetivo, comparando la señal medida y el valor de la señal de consigna, a fin de corregir errores.

**c. Otros sistemas de control** (Chimarro & Pavón, 2017)

Dentro del mundo de la automatización y control industrial, existen gran cantidad de tipos y métodos de control, a manera de información se describen a continuación los más importantes.

1. Control proporcional derivativo (PD), tiene el carácter de previsor, su desventaja es que amplifica la señal de ruido, saturando al controlador.
2. Control proporcional integral (PI), el más utilizado a nivel mundial dentro de la industria en dinámicas de primer orden.
3. Control proporcional integral derivativo (PID), muy sofisticado e incluye los siguientes tipos de controles:

Control Proporcional (P-Kp), proporcional al error

Control integral (I-Ki), proporcional al error acumulado

Control derivativo (D-Kd), predice el futuro

### **1.9.2 Sistema de optimización de energía (Juna, 2015)**

Un sistema de optimización es un conjunto de actividades orientadas a reducir y ahorrar el recurso energético, mejorando su utilización, a través de acciones correctivas en beneficio del mismo sistema. Su objetivo principal es la reducción del gasto energético.

#### **a. Optimización mediante modelos de control**

Estos sistemas buscan realizar un uso racional de la energía así como reducir el consumo de sus partes constitutivas, a fin de prolongar la vida útil de los dispositivos que lo componen y obtener beneficios para el sistema como reducción de averías.

#### **b. Beneficios de utilizar sistemas de control para optimizar la energía**

1. Aumentar el tiempo de funcionamiento
2. Gestión energética de las partes constitutivas del sistema
3. Ahorro energético
4. Reducción de averías
5. Prolongación de la vida útil de los dispositivos
6. Ahorros en mantenimiento

La realización del presente proyecto de investigación, busca desarrollar un sistema de control y optimización de energía para drones cuadricópteros durante vuelo estacionario, para lo cual el presente proyecto se enmarcará dentro de la teoría clásica de control haciendo uso de esquemas de lazo abierto para una etapa previa de medición y otro esquema de control de lazo cerrado o realimentación para la etapa de control y optimización de energía basado en modelos de optimización mediante modelos de control.

## **CAPITULO 2**

### **MARCO METODOLÓGICO**

#### **2.1 Introducción**

La presente investigación tiene un enfoque cuantitativo, puesto que dentro del desarrollo se obtendrán datos de mediciones que serán sometidos a comprobación y análisis por tratarse de un proyecto de campo así como experimental, se hará uso de los siguientes métodos según las siguientes etapas.

#### **2.2 Etapas del proyecto**

##### **2.2.1 Investigación**

Para la etapa de investigación de antecedentes, características de drones se hará uso métodos de investigación asistidos por revisión documental y bibliografía. Se hará uso del método analítico/sintético a fin de establecer el análisis entre los distintos modelos de drones cuadricópteros de uso aficionado de mayor aplicación en la investigación, sus sistemas de energía y determinar el sistema de control a implementar.

##### **2.2.2 Desarrollo**

Dentro de esta etapa se hará uso de método matemático, uso de tablas y gráficas, así como el método analítico/sintético para análisis datos teóricos del consumo de energía del dron cuadricóptero de uso aficionado, así como para las mediciones previas a la implementación y plantear posibles soluciones.

##### **2.2.3 Diseño**

Para la etapa de diseño de la tarjeta electrónica que permita el control y optimización de la energía en drones cuadricópteros, se aplicará un método sistemático

para el desarrollo y obtención de los programas de software para la medición y control que cumplan los objetivos perseguidos.

#### **2.2.4 Implementación.**

Para construcción de la tarjeta electrónica se aplicarán métodos empíricos experimentales, puesto que se realizarán varias pruebas en laboratorio, así como se realizarán consultas a expertos en el campo de las UAV's. Para la etapa final, se hace uso del método matemático de prueba-hipótesis, para determinar los resultados obtenidos posteriores a la implementación.

## **CAPITULO 3**

### **PROPUESTA**

Para la consecución y desarrollo del presente proyecto se ha determinado la siguiente propuesta.

#### **3.1 Propuesta de solución**

El presente proyecto de investigación radica principalmente en desarrollar un sistema de control y optimización de energía, para drones cuadricópteros, durante un vuelo estacionario a fin de obtener un tiempo de vuelo adicional, aprovechando al máximo el recurso energético que provee su batería.

La optimización se la realizará desde el ámbito de la gestión de la energía, tomando como referencias los valores de consumo total del dispositivo y los valores de los consumos parciales de cada uno de los diferentes módulos de los que se compone generalmente un drone cuadricóptero de este tipo.

El sistema de control energético tomará como base la señal de voltaje máximo de carga de la batería en voltios, con su respectivo porcentaje, para que en determinados umbrales o límites, tomar decisiones y empezar la desconexión de módulos prescindibles durante el vuelo estacionario. El tiempo adicional de vuelo que podría obtenerse, puede direccionarse a precautelar la integridad del drone con un aterrizaje controlado o un aterrizaje de emergencia, su geoubicación y rescate en caso de pérdida.

Para el desarrollo de la presente investigación se seleccionará el drone cuadricóptero más adecuado (F450-APM 2.8 de Ardupilot) compuesto por el hardware (módulos) y software más común en los drones cuadricópteros de uso aficionado, teniendo

presente que el desarrollo debe ser adaptativo a la mayoría de los drones modulares que existen en el mercado.

### **3.2 Diseño de solución**

El presente proyecto es experimental y específico, para ello se debe definir varias condiciones para la investigación como: tipo de drone a utilizar, drone de interiores (ambiente de laboratorio), estructura de estudio para simular vuelo estacionario.

Al disponer de un drone cuadricóptero completamente equipado y operativo, se presentará un diseño que sea lo menos invasivo posible, puesto que es necesario intervenir en ciertos puntos y conexiones del drone durante las etapas de medición e implementación.

Para dar soporte a la tesis que rige a este proyecto, en primera instancia se establece una etapa de medición antes de la implementación, para ello se desarrollará un sistema de adquisición de datos online con cable, que sea desmontable, que permita obtener datos de consumo de voltaje, corriente y potencia de los puntos críticos del drone, como la batería y módulo de propulsión (rotores). El propósito de ésta etapa es crear y disponer de una base de datos con valores de corriente y voltaje en función del tiempo, que permitirá evidenciar que porcentaje de la energía de la batería se direcciona al módulo de propulsión, para mantener al drone en el aire, así como determinar el tiempo de vuelo que el drone mantiene con todos sus módulos en funcionamiento (iluminación, sistema FPV o cámara, telemetría, GPS, radio control RF, módulo de propulsión y controladora de vuelo). Datos que permiten generar gráficas y evidenciar el SOD (state of discharge) de la batería durante el tiempo de vuelo del drone, lo que permitirá definir umbrales adecuados de voltaje para iniciar el proceso de control y optimización de energía.

Posterior al desarrollo de la etapa de medición y obtención de datos de consumo de cuadricóptero, se desarrollará el sistema de control y optimización de energía, que radica principalmente en la gestión de la energía que se proporciona a los módulos componentes del drone y que pueden prescindirse sin afectar su desempeño en vuelo estacionario. El algoritmo de control implementado en el microcontrolador de la placa Arduino NANO gestionará la energía entregada a cinco de sus módulos (iluminación, sistema FPV o cámara, telemetría, GPS y radio control) a través de un circuito de control electrónico con



dispositivos semiconductores ideales para proteger y garantizar el apagado de los módulos. El sistema de control utilizará umbrales de voltaje de la batería en descarga para tomar decisiones automáticas a fin de optimizar energía y direccionarla hacia obtener un tiempo de vuelo adicional con un aterrizaje controlado mientras va perdiendo potencia.

### 3.3 Definición de estructura para vuelo estacionario del drone

En vista que el proyecto es experimental y específico se requiere tomar medidas durante su tiempo de vuelo, para ello se definirá una plataforma o estructura para el sostén del drone cuadricóptero en el laboratorio, de tal forma que se pueda simular un tipo de vuelo estacionario definido en el proyecto. El modelo de la estructura se lo definirá tomando las consideraciones necesarias para que no afecte su desempeño.

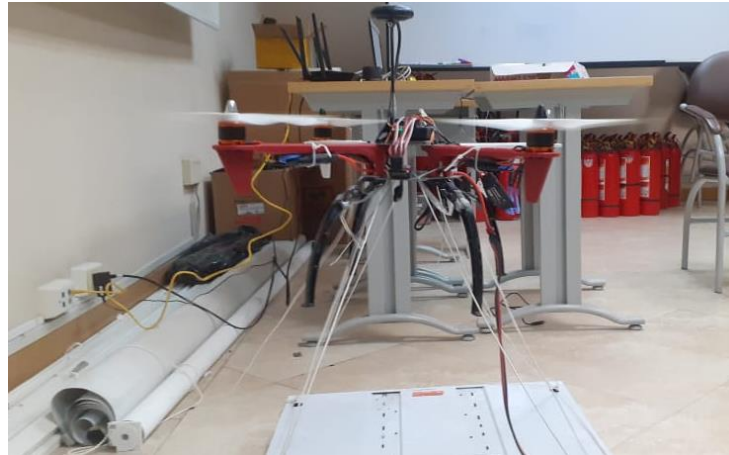
Se han analizado varias estructuras, soportes y esquemas para obtener un vuelo estacionario, así mismo conocer el parámetro de la altura mínima de vuelo sobre el terreno para mantener un vuelo estable en el modelo de drone a utilizar, mismo que varía según el tipo y modelo, conforme fue descrito en el capítulo uno.

Para el caso del drone utilizado en el presente proyecto, según su plataforma de control Mission Planner, la altura mínima para mantener un vuelo estable está sobre los cinco metros, por lo que se determinó la necesidad de probar varias estructuras, para lograr un vuelo estacionario, como se puede evidenciar en las Figuras 3.1, 3.2 y 3.3. La tercera opción de soporte permite una altura estable a través de cordones de sostenimiento.



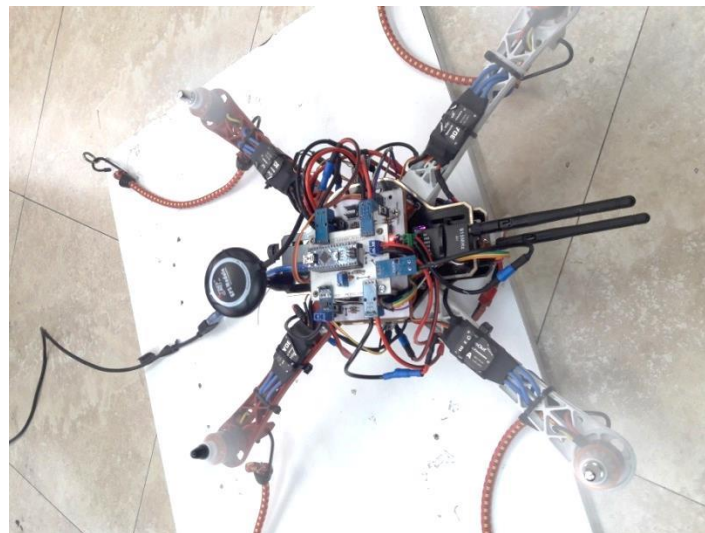
**Figura 3. 1 Estructura 1 de vuelo estacionario**

**Fuente: Elaborado por el autor**



**Figura 3. 2 Estructura 2 de vuelo estacionario**

**Fuente: Elaborado por el autor**



**Figura 3. 3 Estructura final de vuelo estacionario**

**Fuente: Elaborado por el autor**

### **3.4 Acondicionamiento de señales**

Tanto para el sistema de medición de consumo de corriente, voltaje y potencia, así como para el funcionamiento de los dispositivos electrónicos que conformarán la propuesta de solución para éste proyecto, como sensores, transistores, resistencias y demás dispositivos semiconductores, es necesario acondicionar la señal de voltaje que proporciona la batería a la placa Arduino NANO.

### 3.4.1 Señal de voltaje de alimentación al Arduino NANO

El circuito de la Figura 3.4, representa un divisor de voltaje capaz de acondicionar los 12.6 Vdc de la batería hacia los 5 Vdc máximo, admitidos por las entradas analógicas del Arduino NANO, en conjunto con resistencias y capacitores para el filtrado de la señal que reducen el ruido producido. Ésta señal servirá como voltaje de referencia para posteriores análisis.

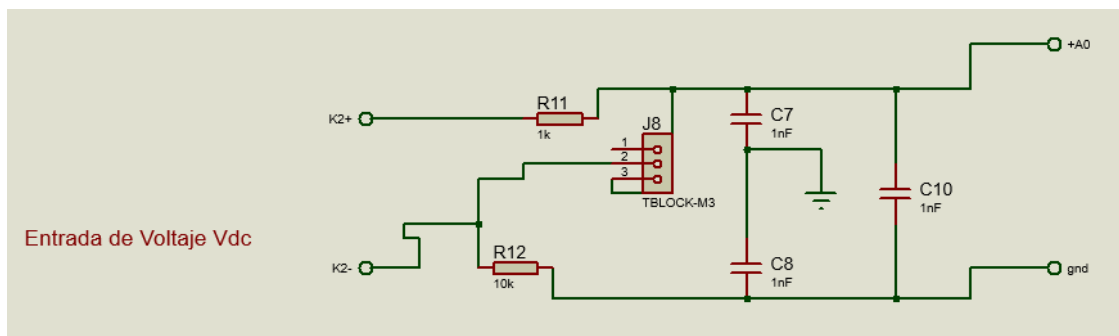


Figura 3. 4 Circuito de acondicionamiento para señales de entrada a Arduino NANO (V – I)

Fuente: Elaborado por el autor

### 3.4.2 Señal de corriente hacia Arduino NANO

La señal de corriente no requiere de un acondicionamiento adicional puesto que el módulo sensor de corriente ACS 712-05A, ACS 712-30A, entrega una señal proporcional de voltaje entre 0 y 5 Vdc.

### 3.5 Arquitectura del sistema de medición de datos de consumo online (desmontable)

El presente sistema tiene como objetivo obtener, procesar y presentar los datos de consumo, es decir mediciones de corriente (I), voltaje (V) y potencia (W). En la Figura 3.5 y Anexo No. 2, se visualiza el diagrama de bloques que refleja la arquitectura general de todo el sistema de medición.

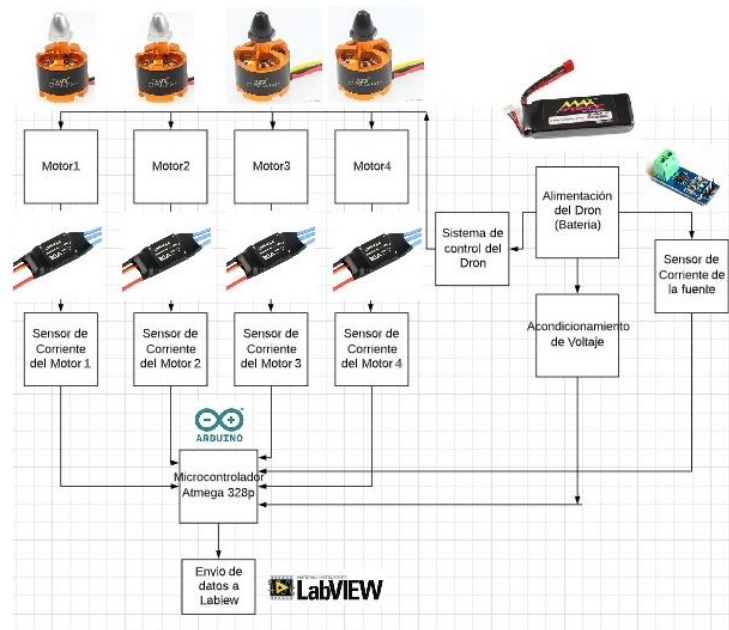


Figura 3. 5 Arquitectura del sistema de medición de consumos

Fuente: Elaborado por el autor

### 3.5.1 Diagrama circuital para el sistema de medición datos de consumo

En la Figura 3.6, se presenta el diagrama circuital que incluye la etapa de acondicionamiento de las señales de entrada (V y I), así como los terminales en donde se conectan los módulos sensores a la placa Arduino NANO, que conforman el sistema de medición de datos de consumo.

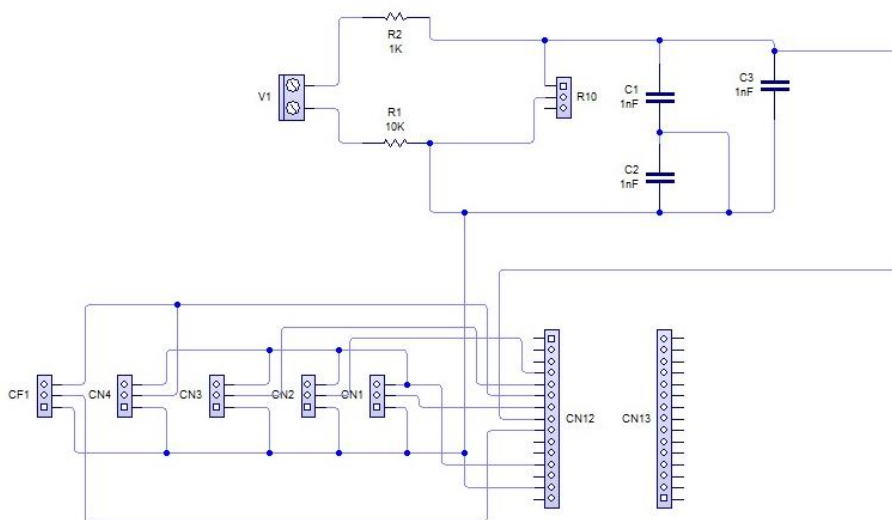


Figura 3. 6 Diagrama circuital para el sistema de medición

Fuente: Elaborado por el autor

Dónde:

<b>CN1:</b>	sensor de corriente motor 1
<b>CN2:</b>	sensor de corriente motor 2
<b>CN3:</b>	sensor de corriente motor 3
<b>CN4:</b>	sensor de corriente motor 4
<b>CF1:</b>	sensor de corriente Fuente
<b>CN12 – CN13:</b>	borneras para Arduino NANO
<b>R1 – R2:</b>	resistencias de acondicionamiento de voltaje
<b>R10:</b>	potenciómetro para calibración de la medición de voltaje
<b>C1 – C2 – C3:</b>	capacitores para acondicionamiento de voltaje
<b>V1:</b>	bornera de entrada de voltaje

### 3.5.2 Descripción del sistema de adquisición de datos de consumo

Se desarrollará una PCB que permita la adquisición de los valores de voltaje y corriente registrados en cada uno de los rotores y batería. Esta PCB se acoplará al cableado existente en el dron y podrá ser desmontable. La placa recibirá cinco señales de entrada correspondientes a voltaje y corriente, mismas que serán enviadas a las entradas del Arduino NANO. De ésta forma se utilizarán cinco entradas analógicas del microcontrolador ATmega328 de la placa Arduino.

El microcontrolador se encargará del procesamiento de la información y envío de los datos a través de comunicación serial hacia la PC, donde se realizará el guardado en forma de matriz dentro de una hoja de cálculo, gracias al software LabView, registrando así toda la información necesaria para su análisis.

Los datos obtenidos permitirán establecer un análisis cuantitativo del consumo de la batería durante el tiempo de vuelo. Se observará el consumo de los cuatro rotores y su valor porcentual respecto al consumo total de corriente. Esto permitirá observar el consumo de energía en función del tiempo de vuelo, lo que permitirá tomar decisiones de control para la gestión de módulos y la optimización de la energía.

### 3.5.3 Calibración de sensores

#### a) Sensores de corriente

El sensor entrega un valor de 2.5 voltios para una corriente de 0A y a partir de allí incrementa proporcionalmente de acuerdo a la sensibilidad, teniendo una relación lineal entre la salida de voltaje del sensor y la corriente.

La ecuación (1), es una línea recta cuando se gráfica la función de voltaje en función de la corriente donde la pendiente es la sensibilidad ( $m$ ) y la intersección en el eje Y es 2.5 voltios. La ecuación de la recta sería la siguiente:

$$V = mI + 2.5 \quad (1)$$

Despejando  $I$ , se tiene la ecuación para hallar la corriente a partir de la lectura del sensor.

$$I = \frac{V - 2.5}{m} \quad (2)$$

Sin embargo, si bien la sensibilidad definida para los sensores de 5 A es de 0.185 V/A, los valores reales entregados por el sensor deben ser correctamente calibrados. Para esto, se han realizado diferentes pruebas de calibración según se puede apreciar en la Figura 3.7, que llevaron a los siguientes valores finales de ecuaciones de cada sensor:



**Figura 3. 7 Pruebas y calibración de sensores de corriente**

**Fuente: Elaborado por el autor**

**Ecuación Motor 1 (Sensibilidad Original = 0.185 V/A):**

$$I_{M1} = \frac{V - 2.52}{0.1498} \quad (3)$$

**Ecuación Motor 2 (Sensibilidad Original = 0.185 V/A):**

$$I_{M2} = \frac{V - 2.51}{0.16949} \quad (4)$$

**Ecuación Motor 3 (Sensibilidad Original = 0.185 V/A):**

$$I_{M3} = \frac{V - 2.49}{0.17033} \quad (5)$$

**Ecuación Motor 4 (Sensibilidad Original = 0.185 V/A):**

$$I_{M4} = \frac{V - 2.45}{0.17857} \quad (6)$$

**Ecuación Corriente Total (Sensibilidad Original = 0.066 V/A):**

$$I_{M5} = \frac{V - 2.496}{0.0627} \quad (7)$$

**b) Sensores de voltaje.**

En la Tabla 3.1, se indica el acondicionamiento realizado en el que se encuentra calibrado los siguientes parámetros:

**Tabla 3. 1 Niveles de voltaje de entrada y salida del Arduino NANO**

	V_IN	V_OUT
<b>Max.</b>	12.6 V	5
<b>Min.</b>	0 V	0 V

**Fuente: Elaborado por el autor**

Por lo tanto, su ecuación correspondiente es:

$$V_{OUT} = 2.52 * V_{IN} \quad (8)$$

### 3.5.4 Diagrama de flujo del algoritmo de medición a implementar

En las Figuras 3.8, se presenta el diagrama de flujo del algoritmo de funcionamiento para la placa Arduino NANO, el cual estará encargado de definir las variables a controlar, establecer la comunicación con el software LabView y controlar el intercambio de datos de medición.

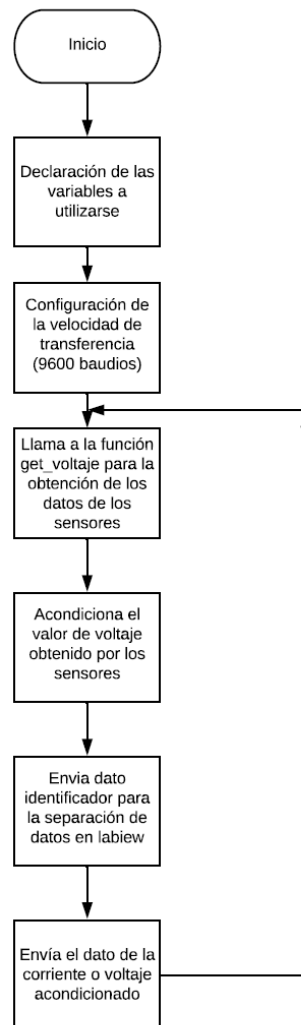


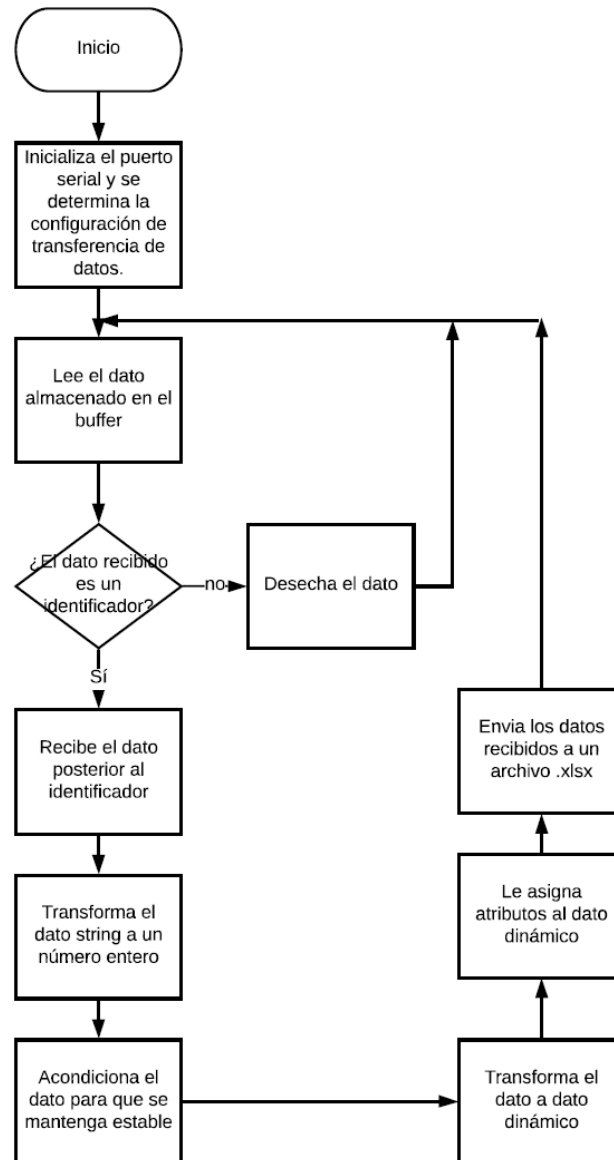
Figura 3. 8 Diagrama de flujo del sistema de medición en placa Arduino

Fuente: Elaborado por el autor



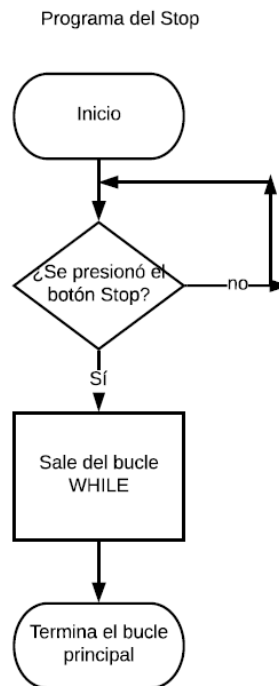
En la Figura 3.9, se presenta el algoritmo a implementar en el software LabView, para la etapa de lectura, acondicionamiento, presentación y almacenamiento de los datos proporcionados por la placa arduino obtenidos desde los sensores.

Programa del lazo Principal

**Figura 3. 9 Diagrama de flujo de sistema de medición en LabView**

Fuente: Elaborado por el autor

Un programa de parada controlado por una función condicional while se muestra en la Figura 3.10, el mismo que interrumpirá la ejecución del algoritmo si se presiona el botón de parada manual (stop).



**Figura 3. 10 Algoritmo de parada en LabView**

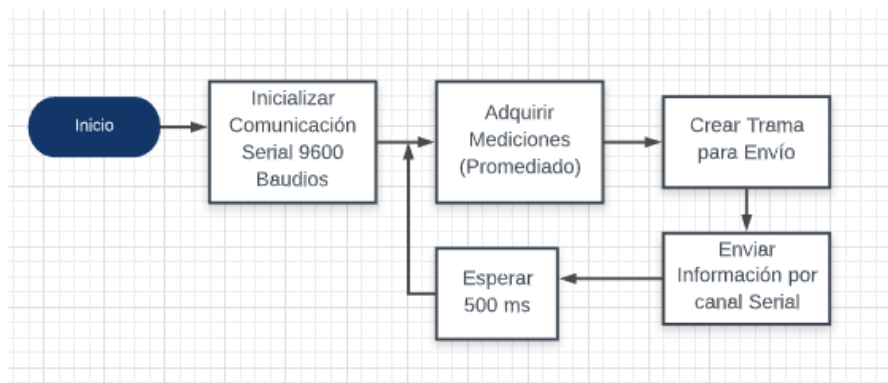
**Fuente: Elaborado por el autor**

El sistema está diseñado para conseguir las mediciones durante el tiempo de vuelo del cuadricóptero. Se medirá la corriente consumida por cada uno de los cuatro rotores, así como el valor de corriente y voltaje suministrado por la batería. Toda esta información es enviada a través de comunicación serial hacia una PC, en donde a través del software LabView se registra dicha información en tiempo real para su posterior análisis.

### **3.5.5 Descripción del algoritmo propuesto para etapa de medición (Arduino NANO)**

En la Figura 3.11, se puede observar el diagrama de bloques resumido del funcionamiento del algoritmo que se grabará en el Arduino NANO para el proceso de

medición de datos de consumo online, las comunicaciones y envío de la información del microcontrolador al software LabView en el computador personal.



**Figura 3. 11 Diagrama de bloques del sistema de medición - placa Arduino.**

**Fuente: Elaborado por el autor**

- 1. Inicializar comunicación:** Esta etapa del programa se realiza una sola vez y habilita la comunicación serial entre placa Arduino y cualquier otro elemento conectado.
- 2. Adquirir mediciones:** Esta etapa del programa consiste en la lectura secuencial de los pines analógicos descritos anteriormente en el esquema general de conexiones. Para este proceso se toman exactamente 2000 muestras, de donde se obtiene su valor medio que es registrado como el valor instantáneo de corriente o voltaje según corresponda. Este valor es luego multiplicado por 100 para su envío como variable entera en el siguiente subproceso.
- 3. Crear trama para envío:** En esta etapa del programa se han tomado las mediciones de todas las entradas analógicas configuradas y se genera una “trama” que se envía hacia la PC. Esta trama tiene la siguiente forma:

**“a C1 b C2 c C3 d C4 e CF f VB”**

Dónde:

**a – b – c – d – e – f**, son letras que separan los datos a enviarse,

**C1 – C2 – C3 – C4 – CF – VB**, son las variables de tipo entero resultado del subproceso anterior.

4. **Enviar información:** Esta etapa del programa consiste en el envío por el canal serial de la trama creada anteriormente.
5. **Esperar 500 ms:** Este puede considerarse el tiempo de muestreo definido para el proceso. Dado que apenas se están monitoreando variables y no se está aplicando ningún esquema de control sobre el funcionamiento de los rotores, no es necesario trabajar a valores menores de tiempo.

### 3.5.6 Descripción del algoritmo propuesto para etapa de medición (LabView)

En la Figura 3.12, se presenta un diagrama de bloques resumido del funcionamiento de la interfaz entre la placa de Arduino y software LabView, para la recepción e interpretación de los datos que se recibirán por la comunicación serial entre los dos dispositivos.

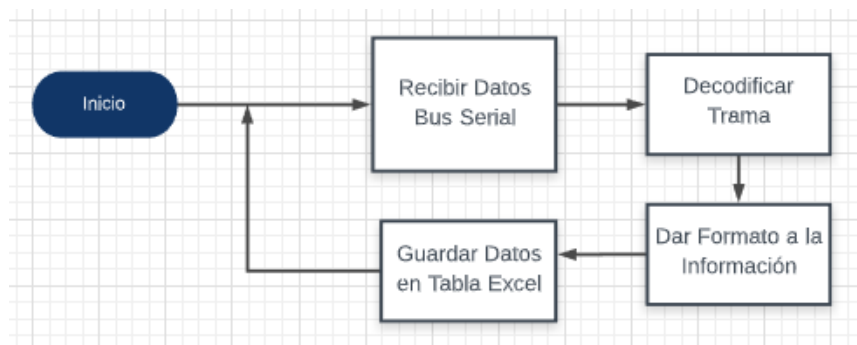


Figura 3. 12 Diagrama de bloques de sistema de medición - LabView

Fuente: Elaborado por el autor

1. **Recibir datos bus serial:** El programa durante su ejecución se encuentra siempre leyendo el bus serial a la velocidad establecida, de modo que al recibir una trama se da inicio al algoritmo diseñado.
2. **Decodificar trama:** En este punto se realiza el proceso inverso al indicado para la placa Arduino. Una vez recibida la trama se observa la letra precedente en la trama, lo cual permite conocer a qué parámetro pertenece la variable entera que llega inmediatamente después.

3. **Dar formato a la información:** Antes de proceder a la publicación de la información en un archivo, es necesario que la variable se convierta nuevamente a flotante y se almacene correctamente en forma de bus.
4. **Guardar datos en tabla:** Finalmente se procede a la modificación de la hoja de cálculo que se creó, incluyendo los nuevos parámetros de la medición y finalizando el ciclo.

### 3.6 Arquitectura del sistema de control y optimización de energía

El sistema presentado a continuación radica principalmente en la gestión de la energía que se proporciona a los módulos componentes del drone cuadricóptero y el apagado de los módulos que pueden prescindirse durante su vuelo estacionario. La arquitectura propuesta como solución al presente proyecto se presenta en la Figura 3.13, así como en el Anexo No. 3.

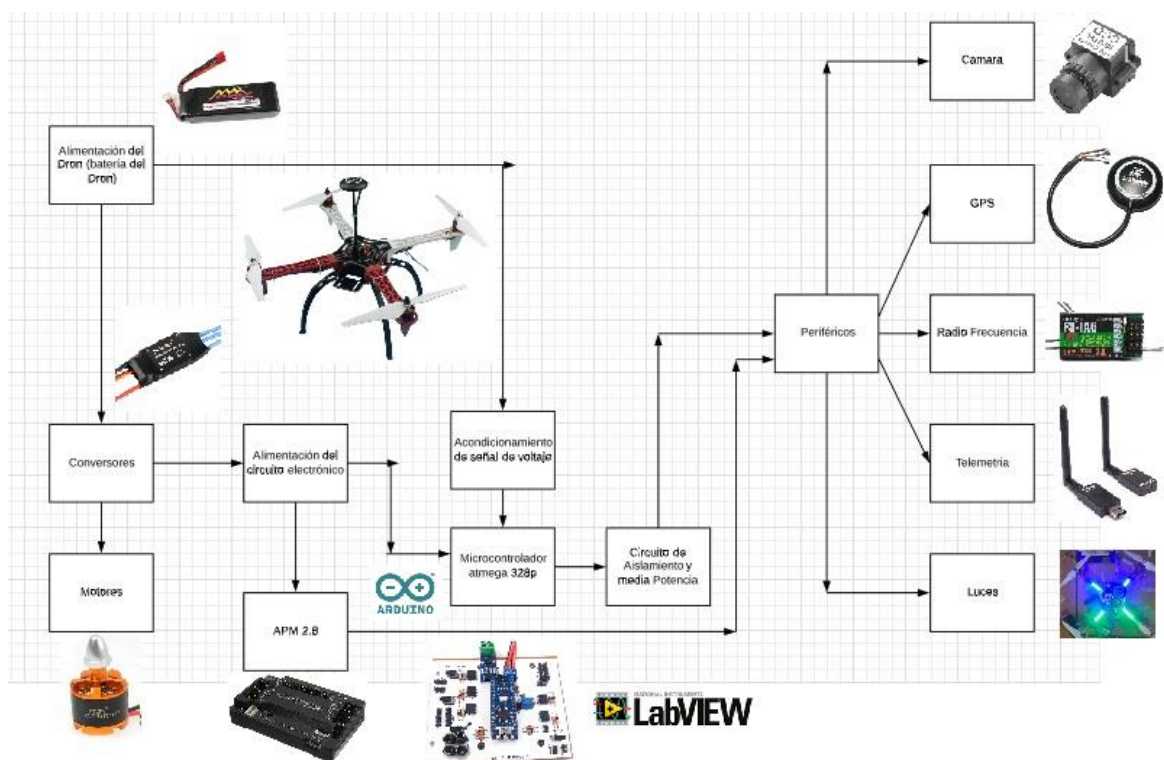


Figura 3. 13 Arquitectura del sistema de control y optimización propuesto

Fuente: Elaborado por el autor

A continuación se hace una breve descripción de cada módulo del drone cuadricóptero F450-APM 2.8, a intervenir y controlar. La propuesta de solución debe ser adaptativa y lo menos invasiva posible en el drone en estudio.

#### a. Módulo de radio control RF (FlySky FS-i6)

El módulo receptor de RF, tiene tres pines conectados (señal, vcc, gnd) a la controladora de vuelo APM 2.8 de Ardupilot, como se puede apreciar en las Figura 3.14 y Figura 3.15. A través de un transistor PNP en corte y saturación (conmutación), con la programación creada desde el microcontrolador ATmega 328P de la placa Arduino NANO y se realizará la desconexión de la señal de alimentación (Vcc) para el control.

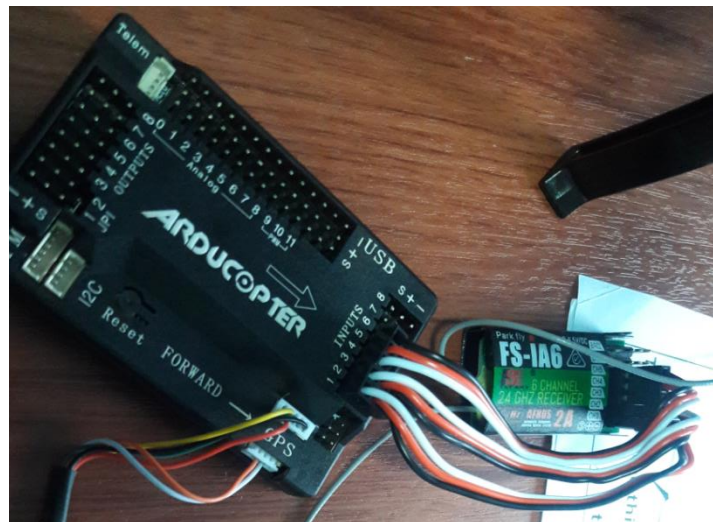


Figura 3. 14 Conexión inicial del módulo RF al APM 2.8

Fuente: Elaborado por el autor

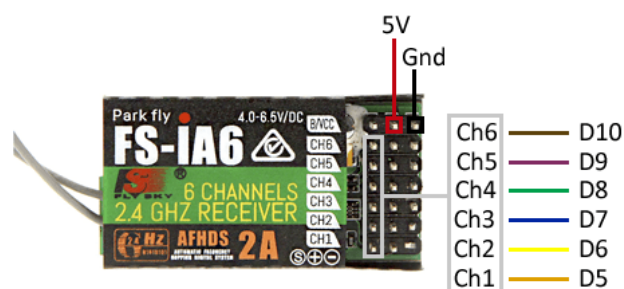


Figura 3. 15 Descripción de pines de Receptor FS-IA6 (Ardupilot, 2019)

**Especificaciones (FlySky, 2014)**

- Frecuencia de trabajo: 2.4GHz.
- Posee 6 canales de 10 bits.
- Potencia de Rf: 20dBm.
- Corriente de operación: 0,06 A.

**b. Módulo del sistema de posicionamiento global GPS**

En la Figura 3.16, se puede apreciar la ubicación inicial del módulo GPS en el drone a una altura de 25 cm aproximadamente, para evitar interferencias que puedan generarse durante el funcionamiento de los rotores y la batería del sistema.



**Figura 3. 16 Módulo GPS de cuadricóptero en estudio**

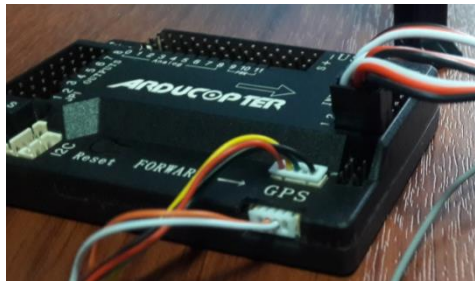
**Fuente: Elaborado por el autor**

**Especificaciones (Waveshare, 2014)**

- Recibe 50 canales.
- Frecuencia de actualización: 5 Hz máximo.
- Sensibilidad de navegación: -161 dBm.
- Temperatura de operación: -40 °C a 85°C.
- Voltaje de operación: 2.7 a 5.0 Vdc.
- Impedancia TXD/RXD: 510  $\Omega$ .
- Corriente de operación: 0,01 A

En la Figura 3.17, se puede apreciar la conexión original del módulo GPS, los pines tienen la siguiente distribución.

- Negro: GND
- Verde: TX
- Amarillo: RX
- Rojo:+5V
- Blanco: SDA
- Naranja: SCL



**Figura 3. 17 Conexión original del módulo GPS al APM 2.8**

**Fuente: Elaborado por el autor**

Todos los pines son terminaciones tipo hembra por lo que se hará uso de conectores macho en la placa de control para su correcta conexión. Para la desconexión de este módulo se intervendrá la señal de alimentación vcc o cable rojo (5 voltios), la que será controlada desde el microcontrolador (Arduino NANO).

### **c. Módulo de iluminación**

En la Figura 3.18, se puede observar el módulo de iluminación compuesto por un juego de cuatro luces tipo led azul y verde, ubicados cada uno en los brazos del drone, que sirven para diferenciar la parte frontal y trasera del drone cuadricóptero. Este módulo es muy importante para asistir visualmente al piloto durante un vuelo. En la actualidad la iluminación que pueda tener o incluirse en un drone se aprovecha en la presentación de espectáculos nocturnos con la conformación de arreglos de drones en forma matricial.





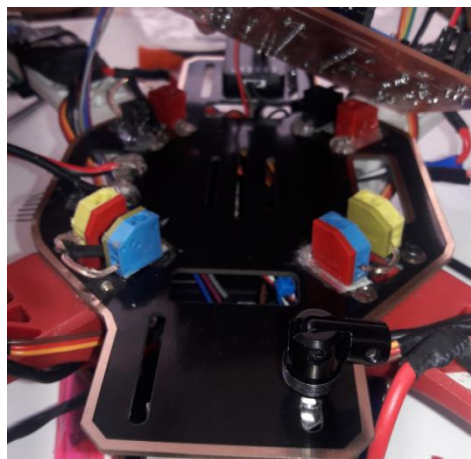
**Figura 3. 18 Módulo de iluminación del drone**

**Fuente: Elaborado por el autor**

### **Especificaciones**

- Luces tipo led bicolor (azul-verde)
- Tamaño: 6 cm x 1,5 cm.
- Alimentación 12 Vdc.
- Consumo: 0,48 A

El módulo de iluminación presentado en la Figura 3.19, está conformado de un circuito en paralelo, al que se conectan cuatro luces tipo led (azul-verde), se dispone de las señales de entrada comunes vcc-gnd para su alimentación y control.



**Figura 3. 19 Conexiones módulo de iluminación**

**Fuente: Elaborado por el autor**

#### d. Módulo de telemetría RF

En forma general el presente módulo permite el intercambio de información de varios parámetros de vuelo entre su transmisor (Aire) y una estación en tierra (receptor conectado a PC).

En las Figuras 3.20 y 3.21, se puede observar la conexión del transmisor del módulo a la controladora de vuelo APM 2.8, (drone), como en la estación terrena en una PC o dispositivo móvil (ArduPilot, 2019).

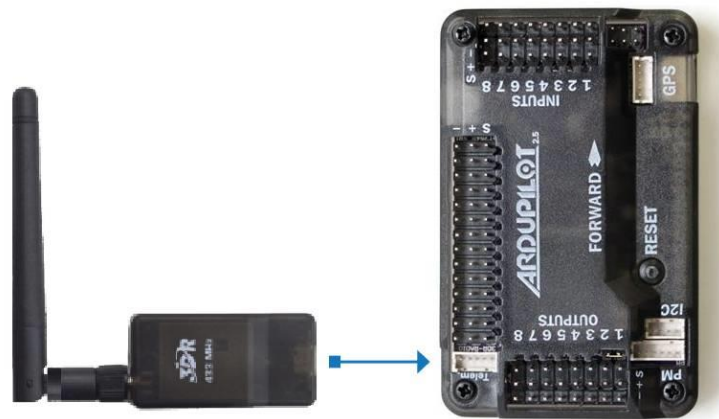


Figura 3. 20 Conexión original del transmisor del módulo de telemetría al APM 2.8 (ArduPilot, 2019)



Figura 3. 21 Conexión de receptor del módulo de telemetría al PC o Smartphone (ArduPilot, 2019)

### Especificaciones (ArduPilot, 2019)

- Canal de frecuencia: 915 MHz.
- Sensibilidad de recepción: -121dBm
- Potencia de transmisión: hasta 20 dBm (100 mW)
- Conector de antena: RP-SMA. 5.8 G
- Rango de alcance: 500 m
- Potencia 500mW.
- Corriente consumida: 0.24 A

A la placa de control se enviarán las cuatro señales que se muestran en la Figura 3.22, la que permitirá la continuidad en los terminales GND, RX, TX, mientras que la señal VCC se conectará a un interruptor electrónico (transistor) que permitirá el control del encendido y apagado de los cinco módulos del drone. Dicho control se lo realizará mediante un algoritmo implementado en el microcontrolador de la placa Arduino NANO, de esta forma saldrán las cuatro señales hacia la controladora de vuelo APM 2.8, para completar el circuito.

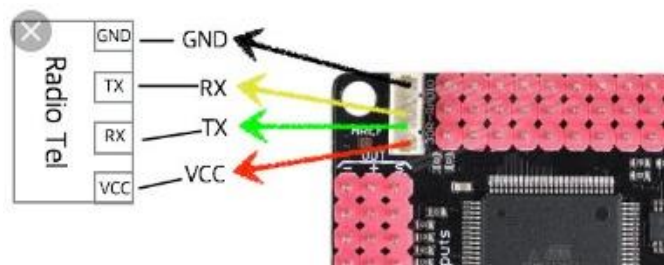


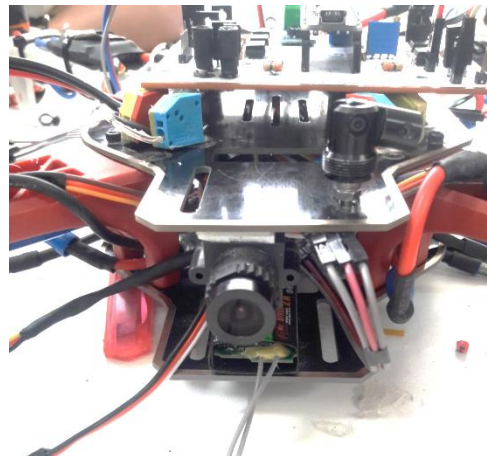
Figura 3. 22 Conexión del transmisor Telemetría a APM 2.8 (ArduPilot, 2019)

#### e. Módulo de cámara o sistema FPV (First Person View)

En forma general el presente módulo está constituido por una mini cámara de 1000TVL, que envía señal de video a su transmisor, los que se alimentan desde la controladora de vuelo APM 2.8. El transmisor envía inalámbricamente (vía RF) señales de video que recibe la cámara del medio hacia un receptor de video en tierra. En las Figuras 3.23 y 3.24 se pueden observar los dispositivos del sistema FPV: cámara, transmisor, conectores y su ubicación en el drone.



**Figura 3. 23 Sistema FPV, cámara y transmisor (EACHINE, 2019)**



**Figura 3. 24 Sistema FPV montado en drone**

**Fuente: Elaborado por el autor**

## **Especificaciones**

### Cámara (EACHINE, 2019)

- Marca: Eachine 1000TVL 1/3 CCD 110 Degree 2.8mm Lens Mini FPV Camera.
- Voltaje de entrada 5Vdc.
- Corriente consumida: 0,055 A - 5V
- Temperatura de operación: 0 a 40°C.
- Peso neto: 10,4 gr.

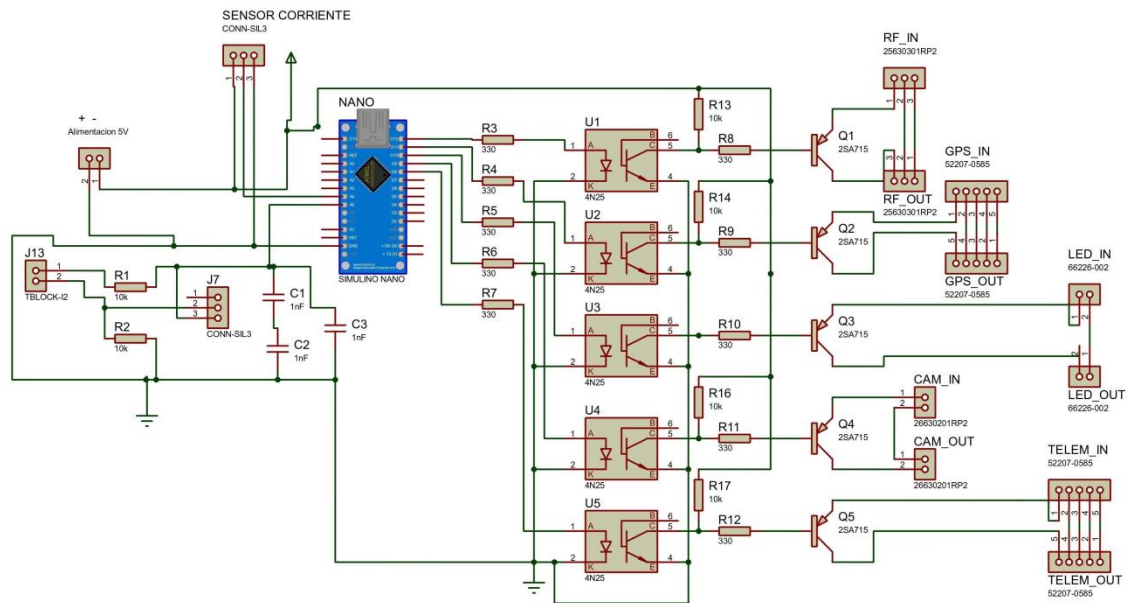
#### Transmisor (NexWareRF, 2014)

- Marca: NextWareRF
- Frecuencia de trabajo: 5.8 GHz.
- Voltaje de alimentación: 6 a 25 Vdc.
- Consumo de energía: 3 Watts
- Potencia de salida: 300mA máx.
- Peso: 18 gr.
- Consumo de corriente 0,28 A

Como se mencionó anteriormente, la cámara se conecta al transmisor de video y éstos a su vez a la controladora APM 2.8 de la cual obtienen su alimentación. El control se lo realizará desde sus cables de alimentación, dónde el cable negro es GND y el cable rojo es el terminal positivo VCC de 5 Vdc. Los cables se dirigen a la placa de control donde la referencia GND es común para todo el sistema del drone, mientras que el terminal positivo se controlará mediante el algoritmo de programación a través de la conmutación del transistor para su control.

### **3.6.1 Diagrama circuital del sistema de control**

La lógica del algoritmo de control y optimización, en conjunto con la parte de hardware específicamente creada para la desconexión de los módulos, se la realizará mediante circuitos de control electrónicos por intermedio de dispositivos semiconductores que garantizan la protección de los módulos. En la Figura 3.25, se puede apreciar el diagrama circuital de la parte de hardware que realizará las funciones operativas de control y optimización de energía en el presente proyecto.



**Figura 3. 25 Diagrama circuital del sistema de control y optimización - Proteus**

**Fuente: Elaborado por el autor**

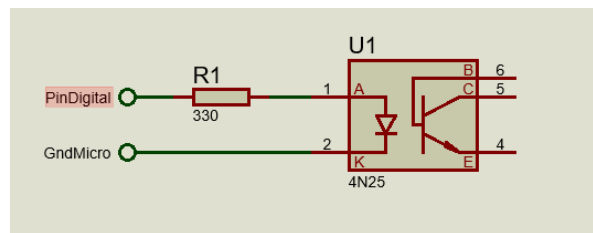
### 3.6.2 Descripción del diagrama circuital del sistema de control

Puede observarse en el diagrama circuital de la Figura 3.25, la presencia de una serie de elementos y dispositivos electrónicos semiconductores, tales como resistencias, capacitores, transistores mosfet, opto acopladores, circuitos integrados de aislamiento, etc.

Dispositivos que configuran el circuito de control para el apagado de los módulos. A continuación en la Figura 3.26, se presenta la conexión y el cálculo realizado para definir el valor de la resistencia que se ubicará en cada uno de los pines digitales de salida del microcontrolador que se han definido para su uso y con la que se da inicio a la etapa de control propuesto.

El siguiente análisis es aplicable y común para los cinco módulos del drone a controlar entre ellos módulo de radio control, GPS, teletría, iluminación y sistema FPV o cámara.

### a. Cálculo de la resistencia de entrada al opto acoplador CI 4N25



**Figura 3. 26** Circuito de aislamiento entre lógica y control

**Fuente:** Elaborado por el autor

Como se puede observar en la Figura 3.26, internamente en la entrada del opto acoplador se encuentra un diodo LED infrarrojo, por el que circulará una corriente de control, para ello se requiere definir una resistencia que permita que la corriente circule por el opto acoplador (CI 4N25) y ésta sea la correcta. Según el datasheet del circuito integrado (Anexos No. 13), el voltaje que llega a caer en el diodo led cuando el fototransistor NPN se encuentre en la región de corte es de 1.15 voltios y circulará una corriente de 10 mA. Por lo que se determina la siguiente ecuación:

$$R = \frac{5 - 1.15}{10mA} = 385\Omega$$

(9)

Para encontrar en el mercado la resistencia aproximada se debe estandarizar a un valor comercial, por lo que se hará uso de la resistencia de valor 330Ω. Este valor de resistencia, tienen tres objetivos fundamentales, proteger al led, permitir el paso de la corriente adecuada que permita las transistor pase del estado de corte a saturación y adicional también limitar la corriente que suministra el pin digital del Arduino NANO, puesto que éste pin puede llegar a suministrar una corriente de 40mA como máximo.

Para determinar la corriente que circularía usando la resistencia determinada en el inciso anterior, se usa simplemente la ley de ohm:

$$I_{pin} = \frac{V_{pin}}{R_{out}};$$

(10)

$$I_{pin} = \frac{5}{330} = 15.15mA$$

Es recomendable que la corriente que circule sea menor al valor máximo dado por el fabricante (40mA).

### b. Cálculo de la resistencia de base del transistor para etapa el control

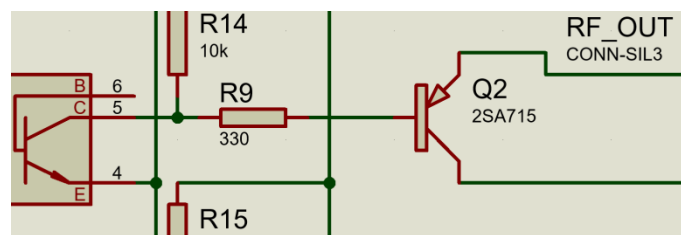


Figura 3. 27 Circuito para control de modulo

Fuente: Elaborado por el autor

Al tratarse de un transistor se conoce que éste debe activarse por corriente. Según el datasheet del transistor PNP-2SA715 de la Figura 3.27 (Anexo No. 13) la corriente que debe circular por la base debe ser superior o igual a 30 mA, para lo que se tiene que determinar la resistencia adecuada que permita nuestro requerimiento. Para ello se realiza un análisis de voltajes en la configuración del transistor y rama respectiva.

$$R = \frac{V_{cc} - V_{BE}}{I_B}$$

(11)

$$R = \frac{12 - 1}{30mA}$$

$$R = 366.6 \Omega$$



Con el fin de adquirir el dispositivo para la implementación se debe normalizar el valor de la resistencia al valor comercial y poder adquirirla en el mercado, el valor normalizado es de  $330 \Omega$ .

La placa electrónica de control a implementar recibirá todas las señales necesarias de los módulos del dron, es decir se convierte en un puente de gestión entre la controladora de vuelo APM y sus módulos, precisamente para que través de ella se configuren las señales de control dentro de un algoritmo de programación (microcontrolador) para el control y optimización de la energía en base a varios umbrales de voltaje de la batería en descarga y que permitirá tomar acciones de apagado de módulos.

### 3.6.3 Dispositivos electrónicos principales de la placa de control

A fin de dotar la protección respectiva a las salidas digitales de la placa Arduino, se incluyen elementos de aislamiento tales como el circuito integrado 4N25 (opto acoplador).

#### a. Opto acoplador 4N25

Compuesto internamente por un diodo LED infrarrojo y un fototransistor NPN, como se muestra en la derecha de la Figura 3.28 y Anexo No. 13. El dispositivo pasa al estado de conducción según la polarización del diodo interno, como se describió en el inciso anterior.

Usado comúnmente para aislamiento de elementos lógicos entre dos etapas de un circuito, para eliminación de ruidos. Sus principales características son las siguientes:

#### **Especificaciones** (Fairchild, 2015)

- Entrada tipo: DC
- Salida: Tipo transistor
- Canales: 1
- Voltaje aislamiento hasta 5000 VRMS
- Compatible con familias lógicas comunes
- Capacitancia de acoplamiento  $<0.5\text{pF}$
- Empaque: DIP-6

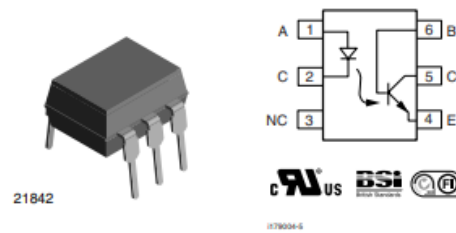


Figura 3. 28 Opto acoplador 4N25 (Fairchild, 2015)

#### b. Transistor: 2SA715

Se trata de un transistor tipo PNP de media potencia que se muestra en la Figura 3.29 y Anexo No. 13, que permite manejar señales de corrientes significativas y que trabajará en la región de corte/saturación para la conducción e interrupción del fluido de corriente, es decir permitirá el control de los módulos del drone.

#### Especificaciones (SavantIC, 2020)

- Voltaje Colector Base 35 V
- Voltaje Colector Emisor -35 V
- Voltaje Base Emisor 5 V
- Disipación hasta 10 W (Fairchild, 2015).

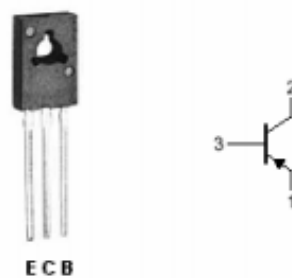


Figura 3. 29 Transistor: 2SA715 (Fairchild, 2015)

### c. Sensor de Corriente ACS-712 5A - 30A

El sensor que se muestra en la Figura 3.30, ACS712 ideal para medir corriente, internamente trabaja con un sensor de efecto Hall que detecta el campo magnético que se produce por inducción de la corriente que circula por la línea que se está midiendo. El sensor entrega una salida de voltaje proporcional a la corriente, dependiendo de la aplicación se puede usar el ACS712-05A, ACS712-20A o el ACS712-30A, para rangos de 5, 20 o 30 amperios respectivamente, ver Anexo No. 13 (Allegro, 2020).

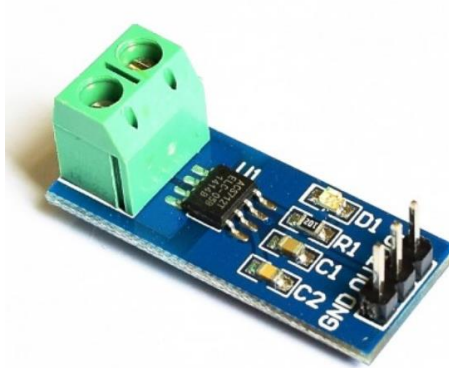


Figura 3. 30 Sensor de corriente ACS-712 5 A - 30 A (Allegro, 2020)

#### Especificaciones (Allegro, 2020)

- Ancho de banda 80KHz
- Resistencia interna 1.2 mΩ
- Voltaje de operación: 5.0 V
- Sensibilidad de salida: 66 to 185 mV/A

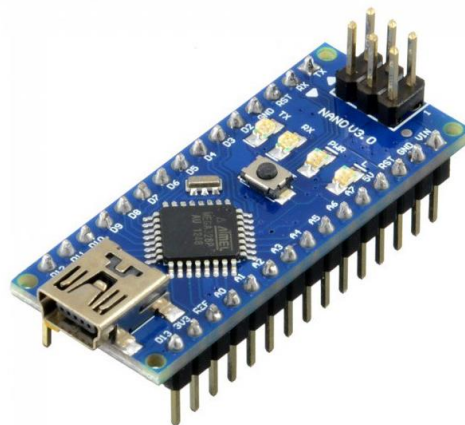
### d. Arduino NANO

Considerando que nuestro proyecto se desarrolla sobre un drone en el que las dimensiones y el peso se debe considerar, las opciones de placas Arduino que se adapta perfectamente al proyecto en desarrollo es el modelo NANO por sus capacidades comparables a las de un Arduino UNO (mayor tamaño), tanto en potencia del microcontrolador, prestaciones, conectividad y tamaño reducido.

Arduino Nano es una placa de desarrollo de tamaño reducido, completo y compatible a nivel de software con múltiples entornos de desarrollo IDE. En la Figura 3.31, así como en el Anexo No. 13, se muestra la placa, basada en el microcontrolador ATmega328P y ATmega168 a 16MHz, sus dimensiones reducidas hacen que sea el preferido para proyectos de menor tamaño y grandes especificaciones.

### **Especificaciones Arduino NANO (Arduino, 2020)**

- Microcontrolador: ATmega328.
- Voltaje de operación: 5V.
- Voltaje de alimentación (Recomendado): 7-12V.
- I/O Digitales: 14 (6 son PWM)
- Corrientes por pines: 40mA
- Entradas Analógicas: 8
- Memoria Flash: 32KB.
- EEPROM: 1KB.
- Frecuencia de trabajo: 16MHz.
- Dimensiones: 18 x 45 mm.
- Conexión mini USB
- Cristal: 16 MHz.
- Consumo de energía: 19mA
- Peso: 7 gr



**Figura 3. 31 Tarjeta Arduino NANO (Arduino, 2020)**

### 3.6.4 Diagrama de flujo para algoritmo de control y optimización a implementar

En la Figura 3.32 se presenta el diagrama de flujo encargado del control de la energía entregada a los módulos del dron, en base a determinados umbrales de voltaje de la batería, la programación respectiva se implementará en el microcontrolador de la placa Arduino.

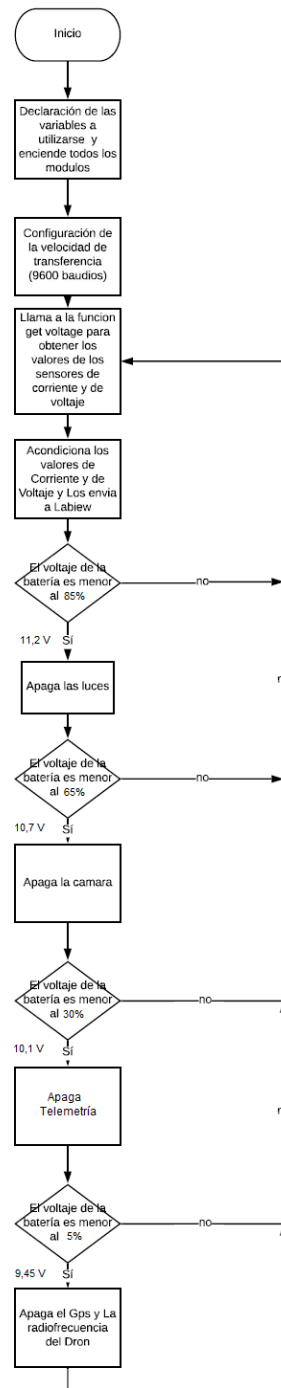
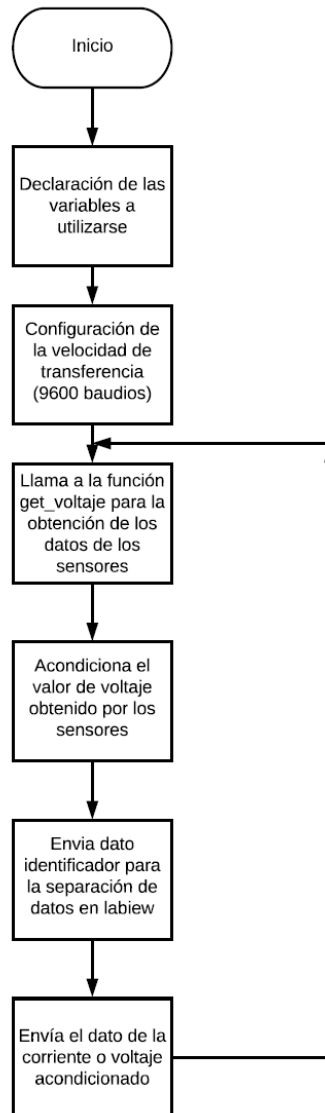


Figura 3. 32 Diagrama de flujo del sistema de control y optimización de energía

Fuente: Elaborado por el autor

En la Figura 3.33, se presenta el algoritmo a implementar tanto en Arduino IDE, así como en el software LabView, para la etapa correspondiente a la lectura, acondicionamiento de los datos proporcionados por la placa arduino para la etapa de control y optimización.



**Figura 3. 33 Diagrama de bloques – sistema de control y optimización**

**Fuente: Elaborado por el autor**

En la Figura 3.34, se presenta el algoritmo a implementar en el software LabView, para la etapa de decodificación, presentación y almacenamiento de los datos tratados por el software descrito, acompañado del programa de parada controlado por una función condicional, el mismo que suspenderá la ejecución del programa de cumplir la condición de parada.

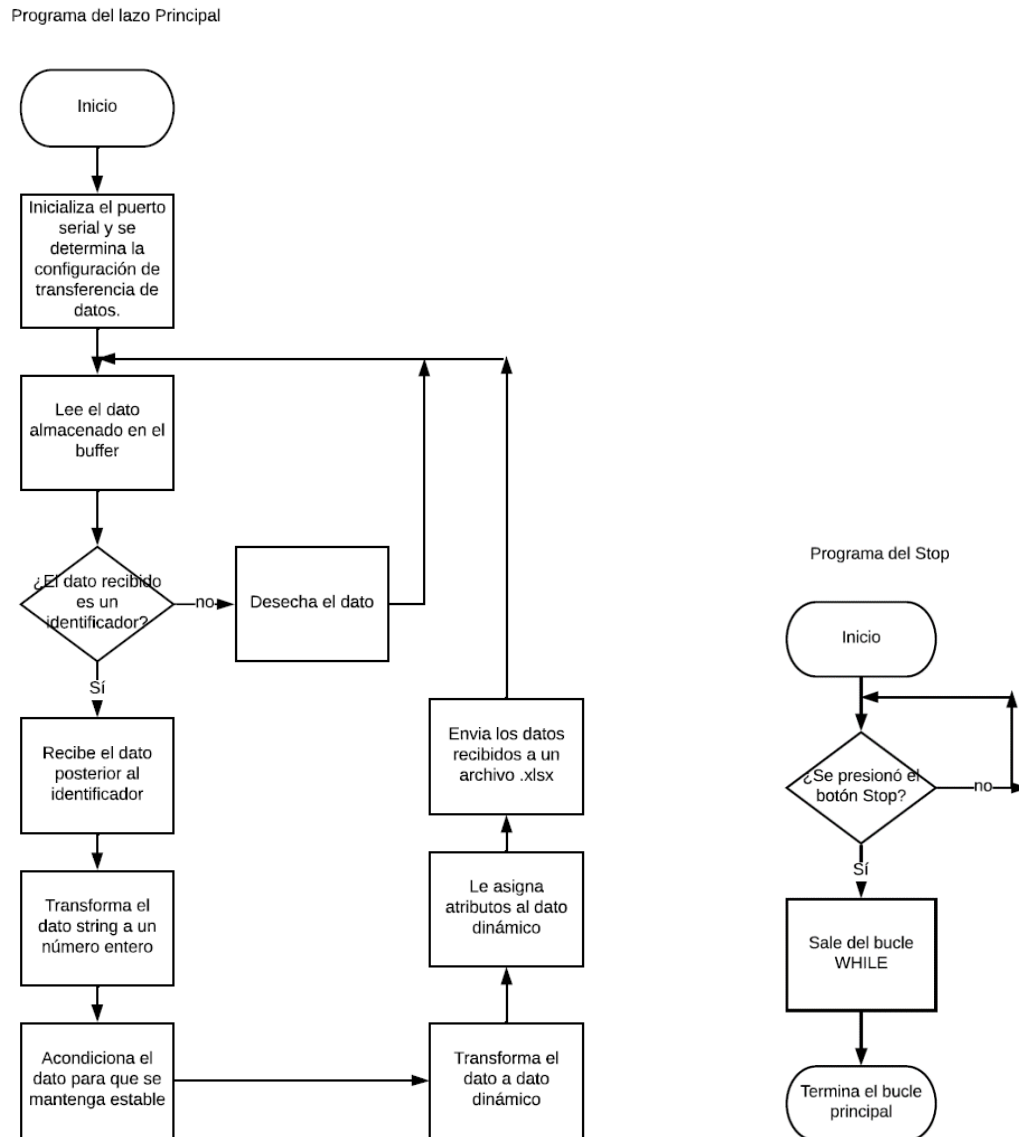


Figura 3. 34 Diagrama de bloques - sistema control y optimización

Fuente: Elaborado por el autor

### 3.6.5 Descripción del algoritmo propuesto para etapa de control (Arduino NANO)

Para el desarrollo del esquema de control se han definido los siguientes datos para la batería:

- **V. Max = 12.7 V** (Batería totalmente cargada)
- **V. Min = 9.45 V** (Batería sin capacidad, nivel crítico)
- **V. Trabajo = 11.5 V** (Voltaje a partir del cual la batería del dron desciende linealmente hasta 9.45 V)

Partiendo de los datos antes definidos, el proceso de funcionamiento del algoritmo de control y optimización de energía se enumera después de la Figura 3.35.

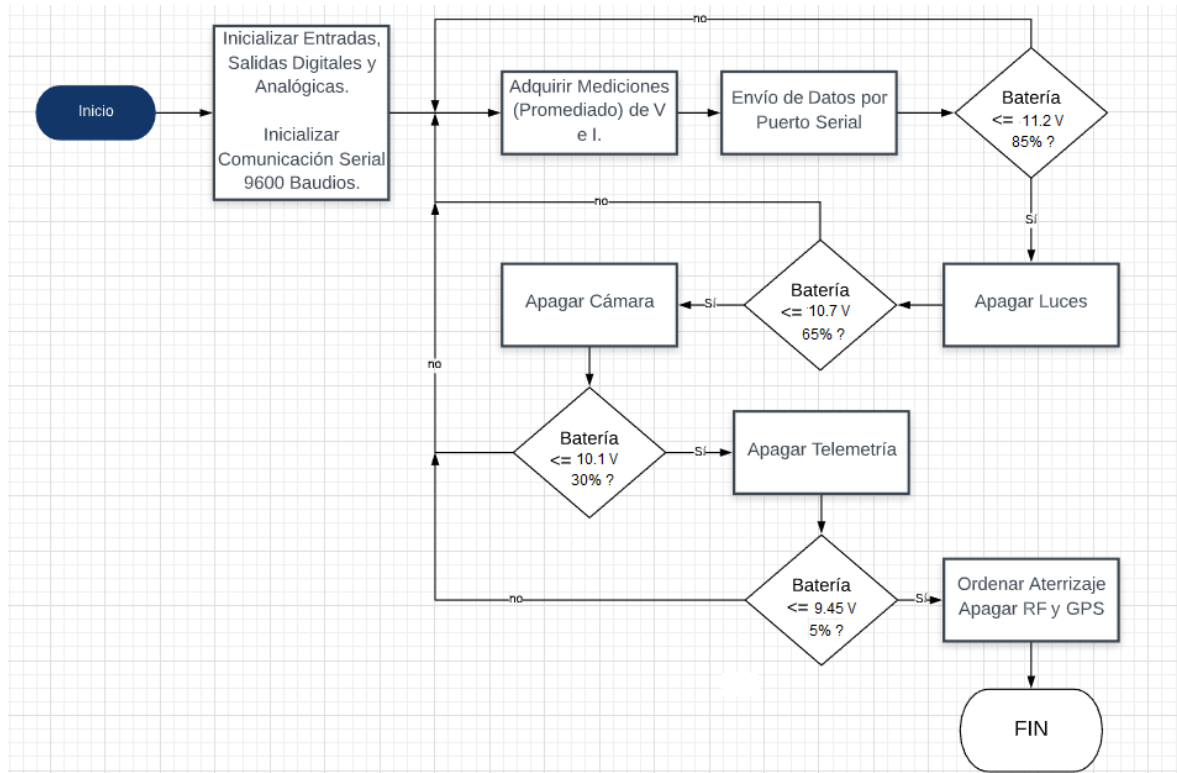


Figura 3. 35 Diagrama de flujo del sistema de control y optimización de energía

Fuente: Elaborado por el autor

1. **Inicializar entradas / comunicación:** Esta etapa del programa se realiza una sola vez y habilita la comunicación serial entre Arduino y cualquier otro elemento conectado, así como la habilitación de los pines configurados como entradas analógicas y salidas digitales.
2. **Adquirir mediciones:** Esta etapa del programa consiste en la lectura secuencial del pin analógico utilizado para la medición. Para este proceso se toman exactamente 2000 muestras, de donde se obtiene su valor medio que es registrado como el valor instantáneo de corriente o voltaje según corresponda. Este valor es luego multiplicado por 100 para su envío como variable entera en el siguiente subproceso.



- 3. Envío datos:** En esta etapa del programa se toma la medida de la entrada analógica configurada y se genera la “trama” que se envía hacia la PC. Esta trama tiene la siguiente forma:

“e CF f VB”

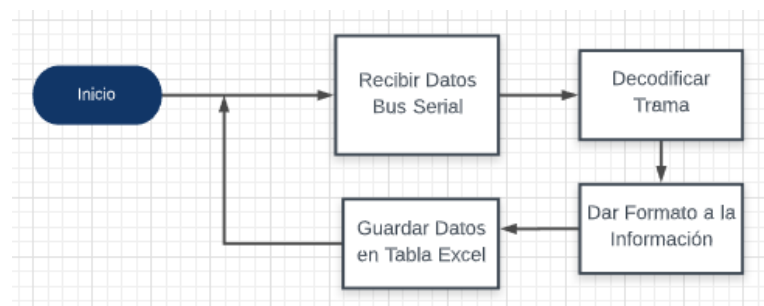
Dónde:            **e – f:**            Letras discriminantes que separan a los datos a enviarse

**CF – VB:**        Variables de tipo entero dato resultado del subproceso anterior

- 4. Control y optimización:** Este apartado indica los niveles, umbrales u etapas bajo las cuales se comanda la gestión de la energía con la desconexión cada uno de los cinco módulos, de acuerdo a los módulos prescindibles de mayor a menor consumo que tiene el drone. El código del algoritmo desarrollado para el programa de control y optimización de energía dentro del entorno de desarrollo Arduino IDE, se presenta en el Anexo No. 6.

### 3.6.6 Descripción del algoritmo propuesto para etapa de control (LabView)

El proceso de funcionamiento del algoritmo, puede explicarse con ayuda de la Figura 3.36, que corresponde a un diagrama de bloques de la etapa de control y optimización, programada en el software LabView.



**Figura 3. 36 Diagrama de bloques – sistema de control y optimización LabView**

**Fuente:** Elaborado por el autor

1. **Recibir datos bus serial:** El programa durante su ejecución se encuentra siempre leyendo el bus serial a la velocidad establecida 9600 baudios, de modo que al recibir una trama se da inicio al algoritmo diseñado.
2. **Decodificar trama:** En este punto se realiza el proceso inverso al indicado para la placa Arduino. Una vez recibida la trama se observa la letra precedente en la trama, lo cual permite conocer a qué parámetro pertenece la variable entera que llega inmediatamente después.
3. **Dar formato a la información:** Antes de proceder a la publicación de la información en un archivo, es necesario que la variable se convierta nuevamente a flotante y se almacene correctamente en forma de bus.
4. **Guardar datos en tabla:** Finalmente se procede a la modificación de la hoja de cálculo creada, incluyendo los nuevos parámetros de la medición y finalizando el ciclo.

### 3.6.7 Análisis de estimación de optimización teórica de la propuesta

A continuación se presentan los consumos de corriente estimados por módulo a intervenir. Los datos presentados en la Tabla 3.2, se obtuvieron de datasheet de los dispositivos y valores recabados del fabricante.

**Tabla 3. 2 Consumos de corriente estimados por módulo**

<b>Módulo</b>	<b>Consumo (A)</b>
	<b>Medido</b>
Propulsión (4 rotores, 2.6 A c/u)	10,4 A
Controladora de vuelo APM 2.8	0,24 A
Radio Frecuencia (RF)	0,06 A
Sistema GPS	0,1 A
Módulo de Iluminación	0,48 A
Módulo de Cámara FPV	0,1 A
Módulo de Telemetría	0,24 A
<b>TOTAL</b>	<b>11,62 A</b>

**Fuente:** Elaborado por el autor

En la Tabla 3.3, se presenta un análisis de estimación teórica de la optimización de energía que se lograría con el sistema de control propuesto.

**Tabla 3. 3 Estimación teórica de optimización**

<b>CANTIDAD-DESCRIPCIÓN</b>	<b>CORRIENTE [%]</b>
7 módulos (funcionando)	11,62 A [100 %]
Módulo de propulsión (4 rotores)	10,4 A [89,5 %]
2 módulos (imprescindibles)	10,4 A + 0.24 A = 10,64 A [91,54 %]
<b>Corriente estimada a optimizar</b>	<b>0,98 A [8,43 %]</b>

**Fuente: Elaborado por el autor**

Obsérvese que del total de la corriente suministrada por la batería el 91.57% no puede ahorrarse, ésta siendo suministrada a los rotores y controladora de vuelo APM. Lo cual permite a priori deducir la optimización de la energía representaría un valor poco significativo, pero que si representaría un tiempo de vuelo adicional que estaría en el orden del minuto, que en el ámbito de los UAV's, drones cuadricópteros de uso aficionado, en el modelo aplicado al presente proyecto, la energía optimizada puede redirigirse a un aterrizaje controlado conforme el equipo va perdiendo potencia, de esta forma salvaguardar la integridad del mismo, con un aterrizaje controlado.

Por lo tanto, con todos los módulos funcionando se estima que el dron agotará su batería en:

$$T_{Vuelo} = \frac{2200 \text{ mAh} \times 1 \text{ A} \times 60 \text{ min}}{1000 \text{ mA} \times 11.62 \text{ A} \times 1 \text{ h}} = 11.36 \text{ min} \quad (11)$$

Luego, funcionando únicamente con los módulos necesarios se estima que el dron agotará su batería en:

$$T_{Vuelo} = \frac{2200 \text{ mAh} \times 1 \text{ A} \times 60 \text{ min}}{1000 \text{ mA} \times 10.64 \text{ A} \times 1 \text{ h}} = 12.41 \text{ min} \quad (12)$$

Bajo las circunstancias descritas se prevé optimizar, ahorrar 1.05 minuto, que en términos de corriente optimizada representa un 8,43 %, del consumo total de corriente del dron, bajo las consideraciones y esquema de control que pretende implementarse. Esto se ve justificado precisamente en el hecho de que la mayor parte del consumo de la batería es destinado precisamente a sostener al dron en vuelo.

### **3.7 Propuesta de uso de software especializados**

#### **3.7.1 Software Arduino IDE**

Arduino IDE es un entorno de desarrollo para la creación de programación de código abierto, la cual está basada en hardware y software libre, flexible y fácil de utilizar para los desarrolladores. En la Figura 3.37, se muestra su logo, que es la imagen de su icono cuando se requiera ubicar programas de éste tipo. Esta plataforma permite crear diferentes tipos de microordenadores en una sola placa y que la comunidad de creadores puede darles diferentes tipos de uso. Muy ampliamente utilizado en la ingeniería de control (Arduino, 2020).

Para el presente proyecto el software se adapta perfectamente para realizar la programación tanto de medición de datos de consumo (corriente, potencia, voltaje), así como para desarrollar, configurar, manipular las variables y parámetros de control en el microcontrolador (ATMega 128p) de la placa Arduino que se ha elegido.



**Figura 3. 37 Software Arduino IDE (Arduino, 2020)**

#### **3.7.2 LabView (Laboratory Virtual Instrument Engineering Workbench)**

Es una plataforma y entorno de desarrollo (software) para ingeniería de sistemas de control, con un lenguaje de programación gráfica, que le permite pruebas, mediciones y

control con acceso rápido a hardware e información de datos. En la parte izquierda de la Figura 3.38 se muestra el ícono con el que se puede identificar a los archivos de este programa. Existen versiones disponibles para las plataformas Windows, UNIX, MAC, GNU y Linux actualmente (National Instrument, 2020).

Para el presente proyecto se utilizará este software para la adquisición de toda la información de medición durante el tiempo de vuelo del drone, se creará interfaces de usuario que permitirán visualizar comportamiento de señales, todo esto se realizará con una comunicación serial entre el software (PC) y el drone en estudio en el laboratorio.

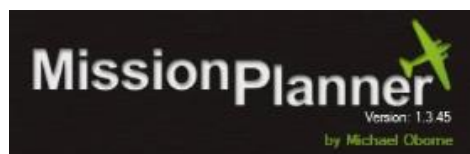


**Figura 3. 38 Software LabView (National Instrument, 2020)**

### **3.7.3 Software Mission Planner - Ardupilot**

En la Figura 3.39, se muestra el ícono de la plataforma de control Mission Planner, que es un software gratuito de código abierto disponible únicamente para Windows, es una interfaz para el controlador o piloto de UAVs, que permite a los usuarios configurar, probar y ajustar al vehículo no tripulado, así como actualizar el firmware de la controladora de vuelo APM. Es llamado también GCS (Ground Control Station) o estación de control de tierra (ArduPilot, 2019).

Para el presente proyecto, éste software permite ver y configurar parámetros de vuelo estacionario, así como observar varios datos de vuelo que son importantes para todo piloto de UAVs.



**Figura 3. 39 Software Mission Planner – Ardupilot (ArduPilot, 2019)**

### 3.7.4 MatLab® de MathWorks

El software MatLab es una plataforma visual con un lenguaje de programación que desarrolla las matemáticas de matrices y arrays directamente, a través de una sintaxis adecuada y flexible para desarrolladores, científicos e investigadores. MatLab® simplifica el análisis de datos a través de etapas previas de pre procesamiento de datos, permite crear modelos de machine learning, su logotipo se presenta en la Figura 3.40, (MathWorks, 2020)

Es prácticamente el software adecuado para procesar varios tipos de datos conectados a sistemas de medición o adquisición de datos, en distintos formatos como archivos de planos, datos de historiadores, texto, imagen, video, etc., de dispositivos, sistema, sensores. En presente trabajo hará uso de una herramienta muy potente de MatLab, para la generación de gráficas, en la etapa de presentación de resultados.



**Figura 3. 40 Software MatLab de MathWorks (MathWorks, 2020)**

## 3.8 Análisis económico del proyecto

En el presente apartado se contemplan costos estimados en relación a los sistemas de medición y control que se van implementar como propuesta de solución al presente proyecto, así como costos de recursos humanos y varios.

### 3.8.1 Costo estimado de la placa de sistema de medición de datos de consumo

En la Tabla 3.4, se presenta un detalle de los componentes electrónicos considerados para la implementación del sistema de medición planteado como solución del presente proyecto.

**Tabla 3. 4 Costo estimado de materiales para sistema de medición**

ELEMENTO	ESPECIFICACIÓN	UNIDAD	SUBTOTAL
Sensor de corriente	ACS-712 5A	4	\$ 4,50
Sensor de corriente	ACS-712 20A	1	\$ 4,50
Capacitores	1nF	3	\$ 0,08
Potenciómetro de precisión	1k $\Omega$	1	\$ 0,50
Borneras de CI	dobles	1	\$ 1,00
PCB doble lado	10x10 cm	1	\$ 2,78
Funda de cloruro férrico	100 gr	1	\$ 1,00
Placa Arduino	NANO	1	\$ 6,40
Resistencias	1k $\Omega$	1	\$ 0,08
	10k	1	\$ 0,08
<b>Total</b>			<b>\$ 20,92</b>

Fuente: Elaborado por el autor

### 3.8.2 Costo de la placa de sistema de control y optimización

En la Tabla 3.5, se presentan los costos de los componentes parte del sistema de control y optimización de energía.

**Tabla 3. 5 Costo estimado de materiales de sistema de control y optimización**

ELEMENTO	ESPECIFICACIÓN	UNID	SUBTOTAL
Sensor de corriente	ACS 712 (20A)	1	\$ 4,50
Opto acoplador	4N25	5	\$ 0,50
Borneras de CI	Dobles	2	\$ 2,00
	1k $\Omega$	1	\$ 0,08
Resistencias	330 $\Omega$	10	\$ 0,08
	10k $\Omega$	6	\$ 0,08
Transistores PNP	2SA715	5	\$ 0,50
Potenciómetro de precisión	1k $\Omega$	1	\$ 0,50
Capacitores cerámicos	1nF	3	\$ 0,08
PCB doble lado	10x10 cm	1	\$ 2,78
Cloruro Férrico	100 gr	1	\$ 1,00
Placa Arduino	NANO	1	\$ 6,40
<b>Total</b>			<b>\$ 18,50</b>

Fuente: Elaborado por el autor

### 3.8.3 Costo estimado de recursos materiales, humanos y varios.

En la Tabla 3.6, se hace mención a los costos de los recursos materiales, humanos, documento escrito y varios.

**Tabla 3. 6 Costo estimado de recursos humanos, materiales y varios**

ELEMENTO	DESCRIPCIÓN	UNID	SUBTOTAL
Diseño	Sistema de medición	1	\$ 150,00
	Sistema de control	1	\$ 150,00
Implementación	Sistema de medición	1	\$ 100,00
	Sistema de control	1	\$ 100,00
Documento escrito (tiempo, materiales de oficina, etc.)	Elaboración	1	\$ 150,00
	Impresión	4	\$ 50,00
Varios	Trasporte, comida	1	\$ 100,00
<b>Total</b>			<b>\$ 800,00</b>

**Fuente: Elaborado por el autor**

En el Anexo No. 16, a modo informativo se presenta una tabla comparativa respecto del drone cuadricóptero utilizado para el presente proyecto, en función de los costos en el que se lo puede encontrar en el mercado de acuerdo a tres proveedores.

### 3.9 Ventajas del sistema

- El producto que se creará será ligero y adaptativo modularmente, agregará un peso de aproximadamente 70 gramos, prácticamente mucho menor en comparación al peso que agrega su batería.
- El consumo de corriente que requiere el sistema de control y optimización es mínimo, está en el orden de los 19 mA, que representa principalmente al funcionamiento de la placa Arduino.
- Puede ser aplicado a drones cuadricópteros de tipo modular de otros fabricantes y de código abierto realizando un análisis y algunas adecuaciones físicas, puede ser aplicable a drones con ésta característica.
- El sistema de control así como el de medición son modulares por lo que pueden ser montables y desmontables, para su uso en el laboratorio.



## **CAPITULO 4**

### **IMPLEMENTACIÓN**

#### **4.1 Desarrollo**

En este capítulo se describe los procesos de implementación del sistema de control y optimización de energía para drones cuadricópteros durante vuelo estacionario y demás componentes del proyecto planteado, en base en los diseños descritos en el capítulo anterior.

En sintonía en el capítulo anterior a continuación se describe el desarrollo del presente proyecto en dos etapas: medición e implementación del sistema de control y optimización de energía para drones cuadricópteros durante vuelo estacionario.

#### **4.2 Implementación**

##### **4.2.1 Construcción del sistema de medición de datos de consumo**

##### **4.2.2 Implementación de etapa de hardware**

En el Anexo No. 14, se presenta el diagrama circuital implementado en el software Proteus, para la generación de las placas PCB (Printed Circuit Board).

Como se puede apreciar en la Figura 4.1, existen cinco módulos sensores de corriente ACS712, cuatro de 5 A y un sensor de 30 A, así como los contactos o puntos terminales para ubicar un socket para la inserción de la placa Arduino NANO, las resistencias y capacitores van ensamblados directamente a la tarjeta electrónica creada.

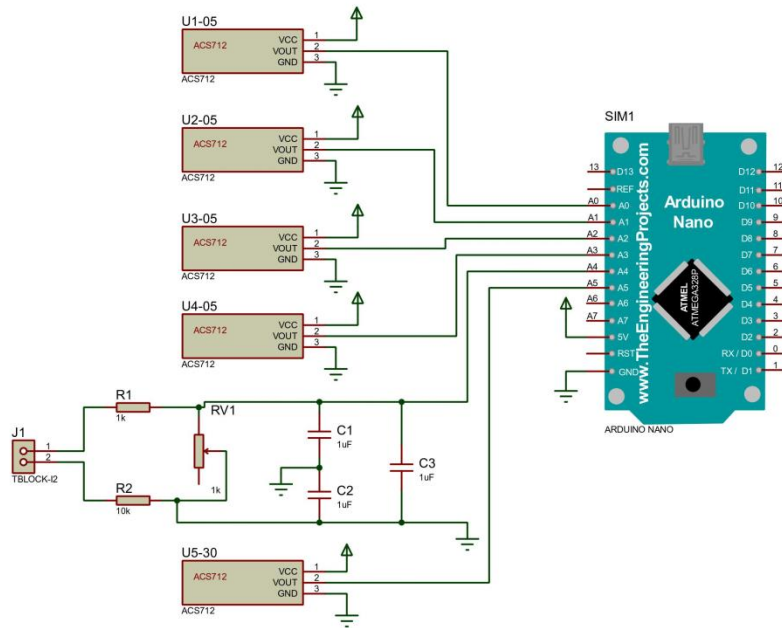


Figura 4. 1 Diagrama circuital de medición creado en software Proteus

Fuente: Elaborado por el autor

La PCB generada en el software proteus, se presentan en la Figura 4.2, mientras que en la Figura 4.3, se presenta el circuito impreso en la baquelita de doble lado (6.62 cm x 4.98 cm) con la configuración más óptima de tamaño y espacio.

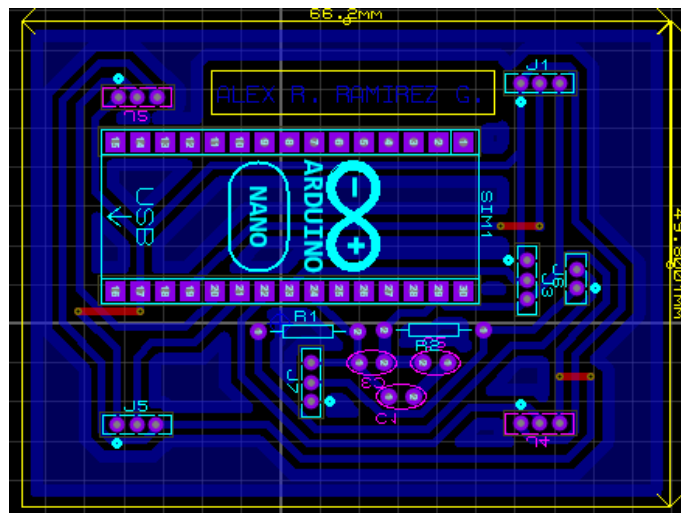
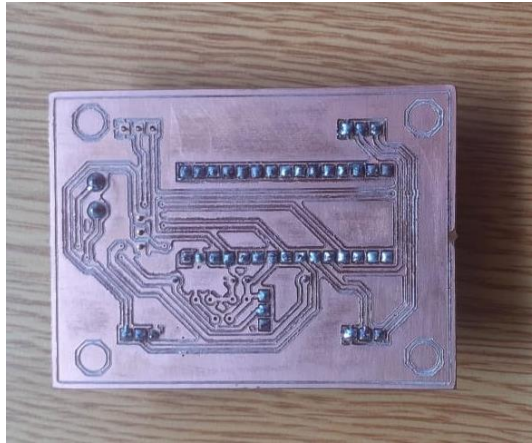


Figura 4. 2 Diseño PCB en software proteus - sistema de medición

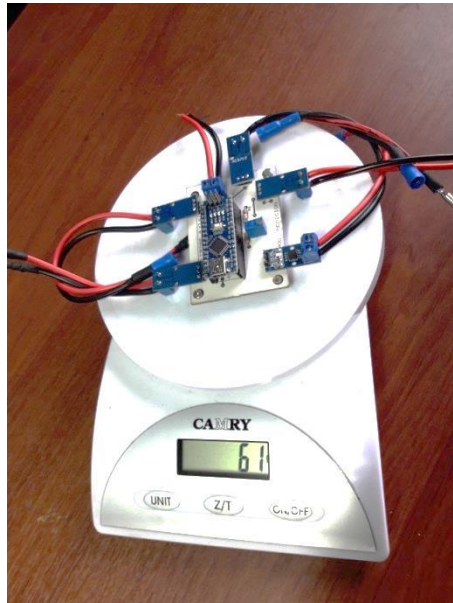
Fuente: Elaborado por el autor



**Figura 4. 3 Circuito impreso a doble lado - Sistema de medición**

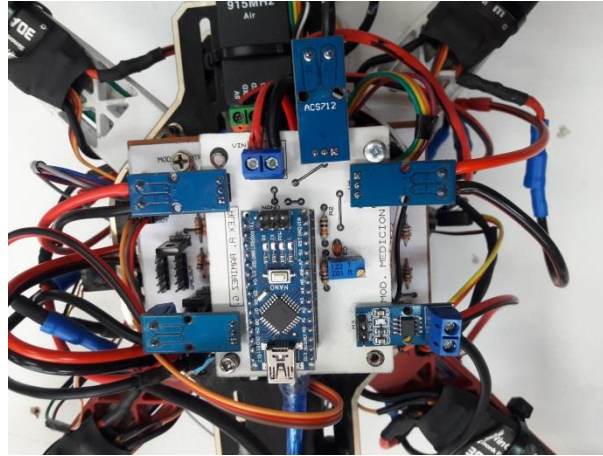
**Fuente: Elaborado por el autor**

En la Figura 4.4, se presenta el sistema de medición de datos de consumo, ensamblado en la baquelita, con todos los dispositivos y cables necesarios. El sistema creado agregará al dron un peso de 61 gr, mientras se obtienen las mediciones antes de la implementación, como se puede apreciar en la Figura 4.5.



**Figura 4. 4 Peso de la tarjeta electrónica de medición**

**Fuente: Elaborado por el autor**



**Figura 4. 5 Tarjeta electrónica (desmontable) creada e instalada en el drone previo a mediciones**

**Fuente: Elaborado por el autor**

En la Tabla 4.1, se detallan todos los materiales y dispositivos que fueron necesarios para la implementación de la placa electrónica de medición de datos de consumo.

**Tabla 4. 1 Materiales utilizados en la placa de medición**

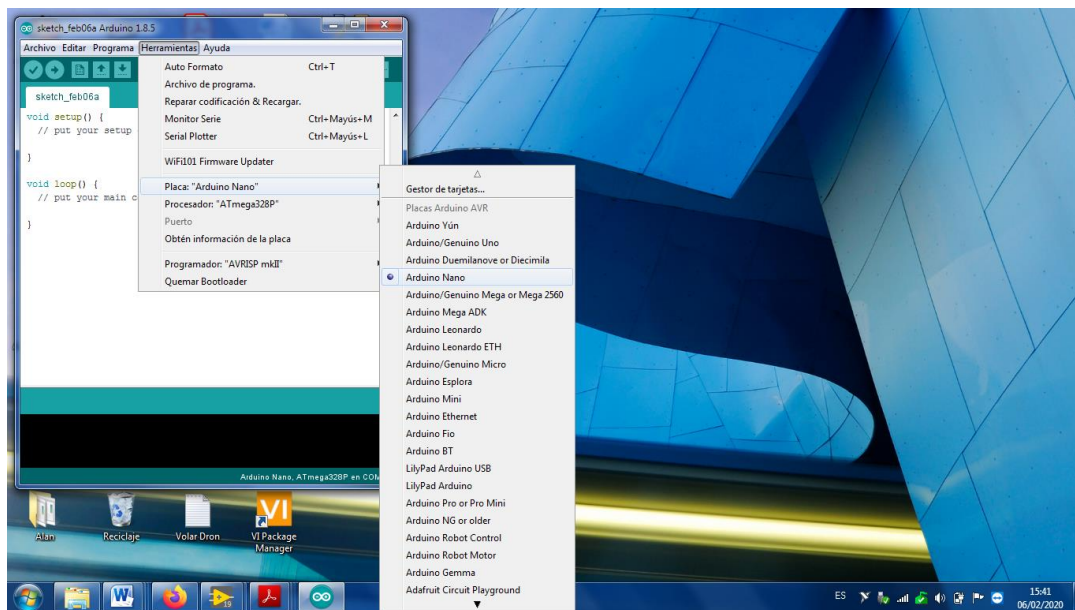
ELEMENTO	ESPECIFICACIÓN	CANTIDAD
Sensor De Corriente ACS-712	5 A	4
	20 A	1
Capacitores	1Nf	3
Potenciometro de precisión	1k $\Omega$	1
Borneras de CI	Dobles	4
PCB doble lado	10x10 cm	1
Funda de Cloruro Férrico	100 gr	1
Placa Arduino	Nano	1
Socket Arduino nano	30 pines	1
Sockets Tipo espadín	30 pines	1
Resistencias de ½ W	1k $\Omega$ +/- 5%	1
	10k +/- 5%	1
Cable comunicación serial	Mini USB a USB	1
Cable gemelo de alimentación	12 AWG	2
Tornillos	06-32	4
Cautín	40W	1
Tubo de suelda 60/40	10 gr	.1
Lavacara	10 cm (radio)	1

**Fuente: Elaborado por el autor**

### 4.2.3 Implementación de la etapa de software

#### a. Programación de la placa Arduino NANO en Arduino IDE

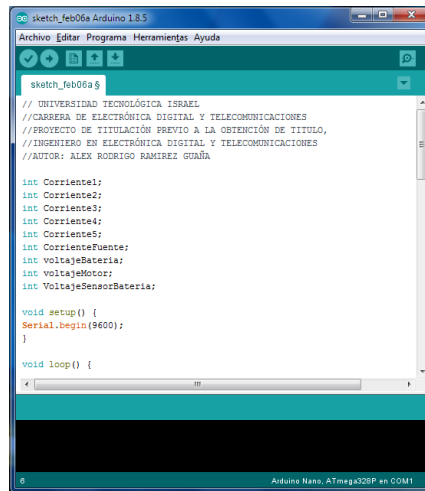
Dentro del entorno de programación Arduino IDE, se escogió el tipo de placa que se utilizará en nuestro caso la placa Arduino NANO, que se ajusta a nuestra necesidad, por sus ventajas, versatilidad, dimensiones, peso y potencia, como se muestra en la Figura 4.6.



**Figura 4. 6 Selección de placa Arduino a utilizar**

**Fuente: Elaborado por el autor**

La programación para la etapa de medición, se desarrolló en función del diagrama de bloques presentado en el Anexo No. 11, el mismo que da inicio con la definición de las variables para las señales de corriente y voltaje que se van a medir. Las señales de entrada son acondicionadas adecuadamente, para que el microcontrolador obtenga los datos de los sensores conectados a las entradas análoga A0-A5 y a continuación por una sola ocasión, establece la comunicación serial a 9600 baudios con la PC, como se muestra en la Figura 4.7.



```

sketch_feb06a $
// UNIVERSIDAD TECNOLÓGICA ISRAEL
//CARRERA DE ELECTRÓNICA DIGITAL Y TELECOMUNICACIONES
//PROYECTO DE TITULACIÓN PREVIO A LA OBTENCIÓN DE TÍTULO,
//INGENIERO EN ELECTRÓNICA DIGITAL Y TELECOMUNICACIONES
//AUTOR: ALEX RODRIGO RAMIREZ GUARÍA

int Corriente1;
int Corriente2;
int Corriente3;
int Corriente4;
int Corriente5;
int CorrienteFuente;
int voltajeBateria;
int voltajeMotor;
int VoltajeSensorBateria;

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

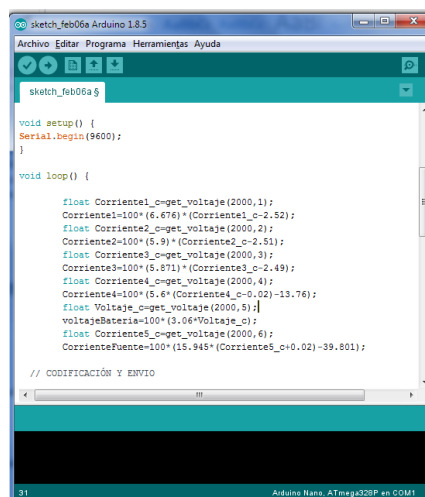
void loop() {

```

Figura 4. 7 Definición de variables y comunicación serial Arduino - PC

Fuente: Elaborado por el autor

En el cuerpo del programa se llama a las funciones *get* para la adquisición de datos de corriente o voltaje, que consiste en la lectura secuencial de los seis pines analógicos (A0-A5) a través de un bucle *for* y un condicional *if* anidado, como se puede ver en la Figura 4.8, en la que se toman 2000 muestras de cada uno de ellos. A manera de un ejemplo se presenta la siguiente sintaxis: *float corriente1\_c=get\_voltaje(2000,1)*; que corresponde a la obtención de un valor medio del total de muestras de la señal de corriente que circula por el rotor 1, valor que posteriormente se lo acondiciona tomando en consideración la ecuación (3) del capítulo anterior, para su envío como variable entera al siguiente subproceso, como se puede apreciar en la Figura 4.8.



```

sketch_feb06a $

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

void loop() {

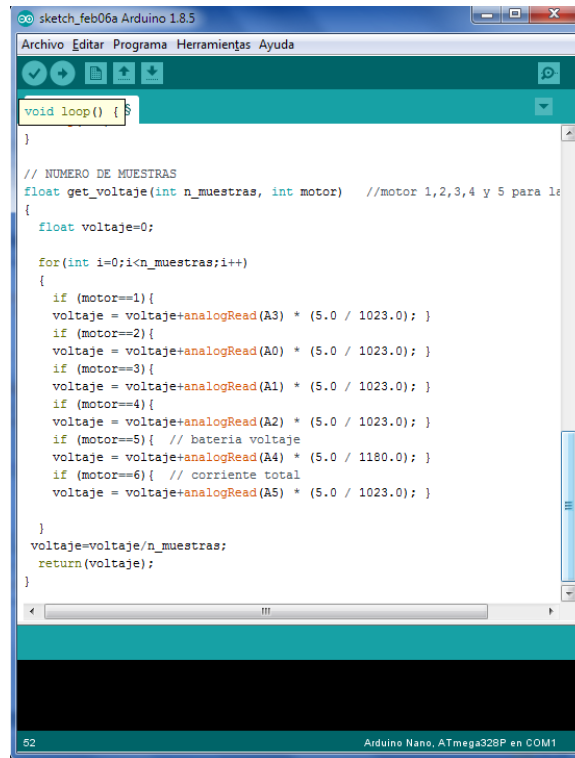
  float Corriente1_c=get_voltaje(2000,1);
  Corriente1=100*(6.676)*(Corriente1_c-2.52);
  float Corriente2_c=get_voltaje(2000,2);
  Corriente2=100*(5.9)*(Corriente2_c-2.51);
  float Corriente3_c=get_voltaje(2000,3);
  Corriente3=100*(5.871)*(Corriente3_c-2.49);
  float Corriente4_c=get_voltaje(2000,4);
  Corriente4=100*(5.6*(Corriente4_c-0.02)-13.76);
  float Voltaje_c=get_voltaje(2000,5);
  voltajeBateria=100*(3.06*Voltaje_c);
  float Corriente5_c=get_voltaje(2000,6);
  CorrienteFuente=100*(15.945*(Corriente5_c+0.02)-39.801);

  // CODIFICACIÓN Y ENVÍO

```

Figura 4. 8 Lectura y acondicionamiento de las señales de corriente y voltaje detectadas

Fuente: Elaborado por el autor



```
void loop() {  
}  
  
// NUMERO DE MUESTRAS  
float get_voltaje(int n_muestras, int motor) //motor 1,2,3,4 y 5 para la  
{  
    float voltaje=0;  
  
    for(int i=0;i<n_muestras;i++)  
    {  
        if (motor==1){  
            voltaje = voltaje+analogRead(A3) * (5.0 / 1023.0); }  
        if (motor==2){  
            voltaje = voltaje+analogRead(A0) * (5.0 / 1023.0); }  
        if (motor==3){  
            voltaje = voltaje+analogRead(A1) * (5.0 / 1023.0); }  
        if (motor==4){  
            voltaje = voltaje+analogRead(A2) * (5.0 / 1023.0); }  
        if (motor==5){ // bateria voltaje  
            voltaje = voltaje+analogRead(A4) * (5.0 / 1180.0); }  
        if (motor==6){ // corriente total  
            voltaje = voltaje+analogRead(A5) * (5.0 / 1023.0); }  
    }  
    voltaje=voltaje/n_muestras;  
    return(voltaje);  
}
```

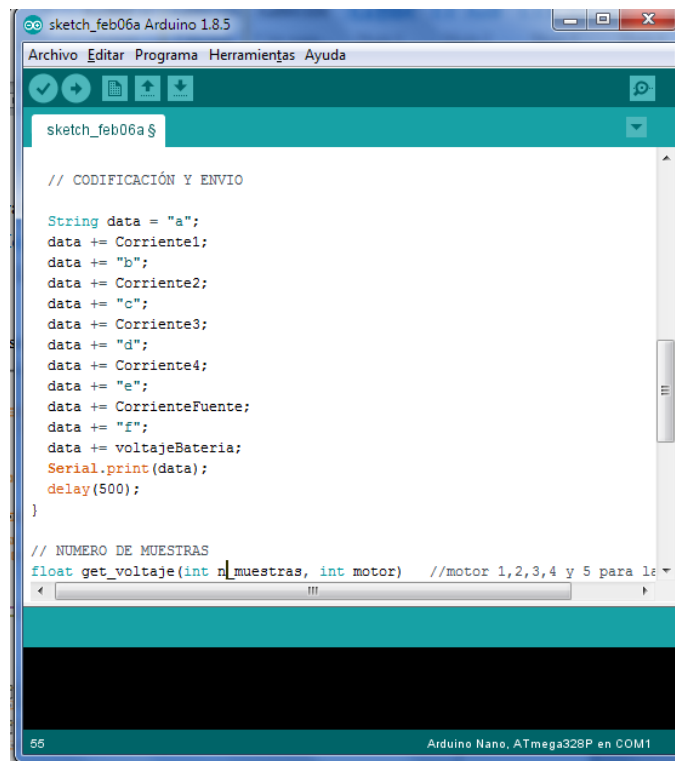
**Figura 4. 9** Toma de muestras de señal a la entrada del microcontrolador.

**Fuente:** Elaborado por el autor

Una vez que se dispone de los valores enteros presentes en las entradas análogas del microcontrolador ATmega328P de la placa de Arduino, éstos son codificados para su impresión en forma de trama en el bus serial. El microcontrolador gestiona el intercambio de información con el software LabView residente en un computador personal a través de una comunicación serial, una vez que se haya iniciado la sesión y es en donde se almacena los datos de consumo del dron.

En la Figura 4.10 se muestra el código creado para que el microcontrolador, forme e imprima la trama codificada en su puerto serial, misma que a través del software LabView, en una computadora es recibida, decodificada, acondicionada, almacenada y presentada a través de una interfaz HMI, para su visualización y seguimiento.

Todo el código de programación creado en el entorno de desarrollo Arduino para ésta etapa de medición se puede observar en el Anexo No. 4.



```
sketch_feb06a Arduino 1.8.5
Archivo Editar Programa Herramientas Ayuda
sketch_feb06a $
// CODIFICACIÓN Y ENVIO
String data = "a";
data += Corriente1;
data += "b";
data += Corriente2;
data += "c";
data += Corriente3;
data += "d";
data += Corriente4;
data += "e";
data += CorrienteFuente;
data += "f";
data += voltajeBateria;
Serial.print(data);
delay(500);
}
// NUMERO DE MUESTRAS
float get_voltaje(int n_muestras, int motor) //motor 1,2,3,4 y 5 para le
65 Arduino Nano, ATmega328P en COM1
```

Figura 4. 10 Creación y codificación de trama enviada al Software LabView.

Fuente: Elaborado por el autor

## b. Programación en LabView (PC)

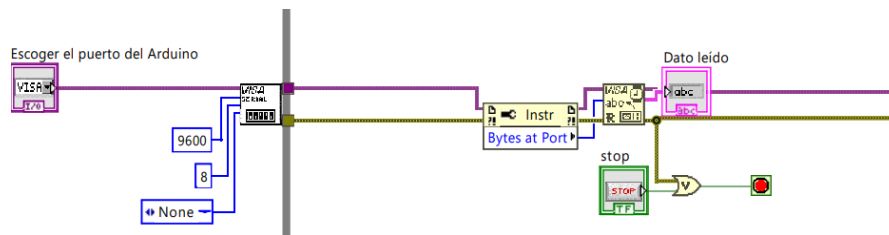
En el Anexo No. 5, se presenta una vista completa de la programación realizada en el software LabView, que corresponde a las consiguientes etapas de recepción, decodificación, formato, almacenamiento y presentación de los datos censados de voltaje y corriente en los rotores 1 - 4 y batería del drone. A continuación se describirán cada uno de los siguientes subprocesos descritos:

### 1. Etapa de inicio de sesión y recepción de datos

A través del bloque *visa*, que se encuentra en la parte central de la Figura 4.11, inicia la etapa con la apertura del puerto del Arduino con el que se va a iniciar la sesión de comunicación, al establecerla previamente como serial, se definen varios parámetros en el bloque *visa serial* como: frecuencia de muestreo 9600 baudios, número de bits a transmitir



8 bits, el tipo de paridad (sin paridad), parámetros suficientes para nuestro intercambio de información.



**Figura 4. 11** Etapa de inicio de sesión y recepción de datos en LabView

**Fuente:** Elaborado por el autor

En la parte derecha de la Figura 4.11, se encuentra un nodo de propiedades, el que trasmite los parámetros importantes definidos en el inicio de la sesión de comunicación, como el número específico de bytes transmitidos en la interface. El bloque *Visa Read*, identifica el número de bytes y presenta el dato en su buffer de salida, para el siguiente subproceso de decodificación, de no existir comunicación en el puerto, a través de un *bucle while* se continúa ejecutando el algoritmo hasta que se establezca la comunicación o se detenga a través del control de parada.

Se ha ubicado etiquetas estratégicamente que corresponden a un mismo punto y dato obtenido, para mejor visualización y comprensión del algoritmo desarrollado.

## 2. Etapa de Decodificación

Los datos recibidos en cadena ingresan al bloque *match pattern*, uno tras de otro, con el siguiente formato (letra-dato1, letra-dato2, letra-dato3, letra-dato4, letra-dato5 letra-dato6), conformada de los caracteres respectivos (a, b, c, d, e, f) que corresponden respectivamente al dato de corriente que circula por cada rotor del drone (rotor 1, 2, 3, 4), los últimos dos identificadores e, f, corresponden a datos de corriente y voltaje de la batería, como se puede apreciar en la Figura 4.12 y en el Anexo No. 5

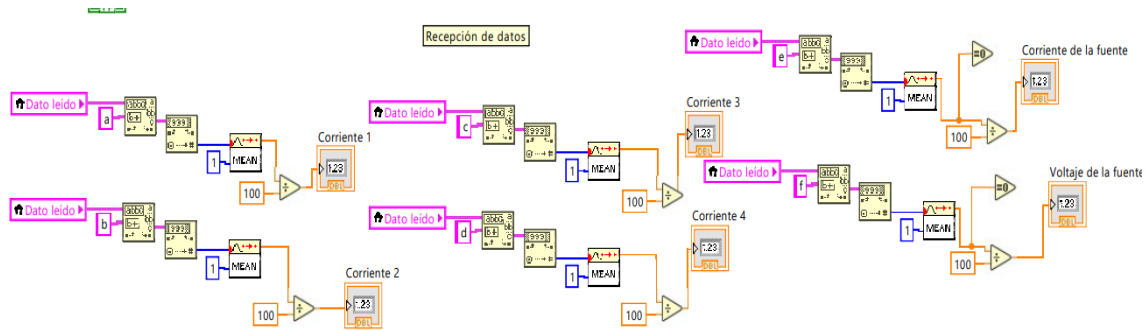


Figura 4. 12 Etapa de decodificación de datos en LabView

Fuente: Elaborado por el autor

La cadena de datos que sale del bloque *match pattern*, ingresa al siguiente bloque que devuelve únicamente un dato entero, a través de una integración decimal. El dato tomado por el sensor es promediado con una muestra definida, para presentarse como un dato de coma flotante. Este proceso se realiza para cada dato obtenido, conformando la trama enviada por el Arduino.

### 3. Etapa de presentación o visualización

Al disponer del dato que representa el valor de la corriente que circula en ese instante por el punto de medición, se realiza una verificación y selección del dato obtenido en el subproceso anterior. El bucle de realimentación que se presenta en la Figura 4.13, permite almacenar y presentar el dato guardado en la interfaz grafica HMI creada para poder visualizar las mediciones en tiempo real, según Figura 4.14.

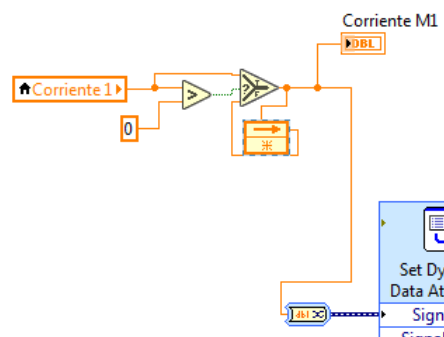
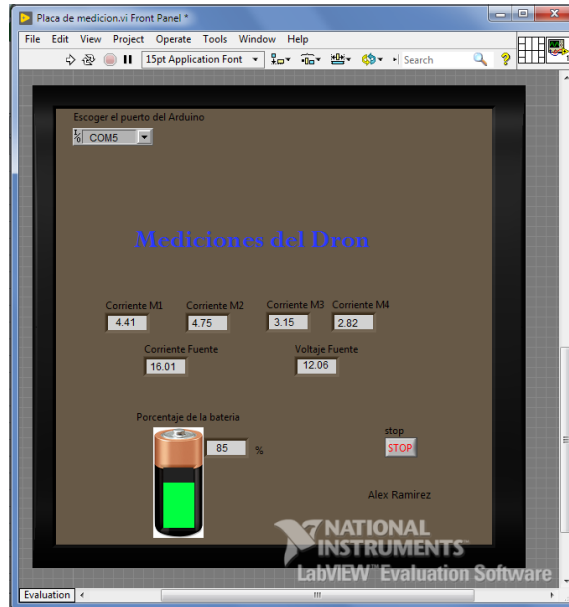


Figura 4. 13 Etapa de verificación del dato obtenido para visualización

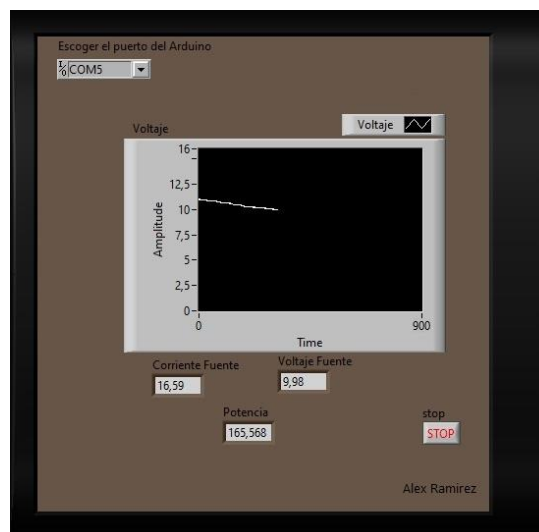
Fuente: Elaborado por el autor



**Figura 4. 14** Interfaz gráfica HMI, para visualización de medidas

**Fuente:** Elaborado por el autor

Para el caso especial de la batería del drone se ha creado una interfaz HMI que se muestra en la Figura 4.15, dentro del software LabView, a travez del bloque *waveform chart*, se puede observar en tiempo real el estado de descarga de la batería SOD, en función del tiempo de vuelo del drone.



**Figura 4. 15** Interfaz gráfica HMI, estado de descarga de la batería

**Fuente:** Elaborado por el autor

## 4. Formato

Para poder almacenar los datos de consumo obtenidos de los rotores y batería, así como mantener un registro de las mediciones, se cambió al dato obtenido del proceso anterior (dato flotante) a un dato dinámico través de un bloque convertidor y continuación asignar los atributos y características respectivos para su registro en una hoja de calculo, conforme se muestra en la Figura 4.16 y Anexo No. 5.

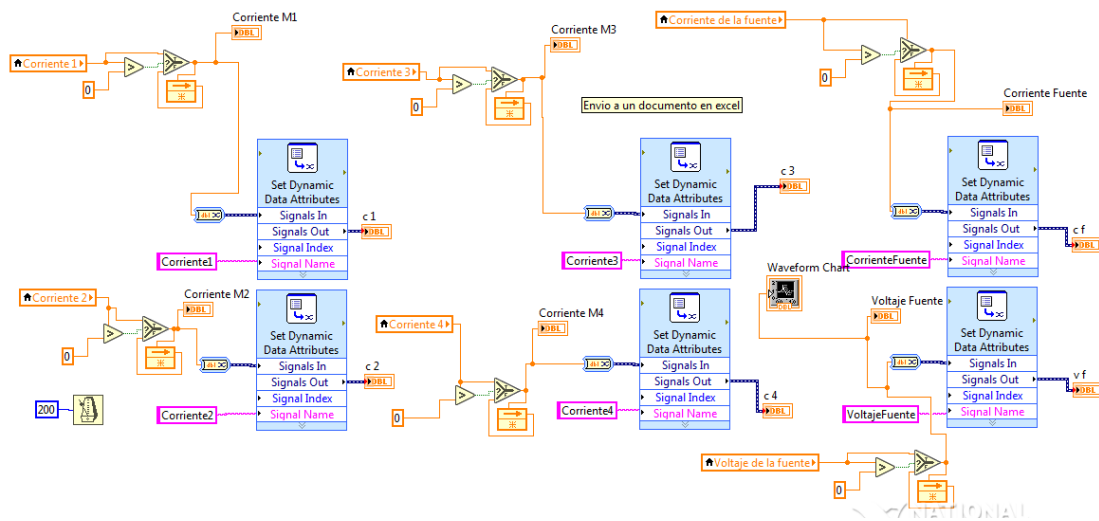
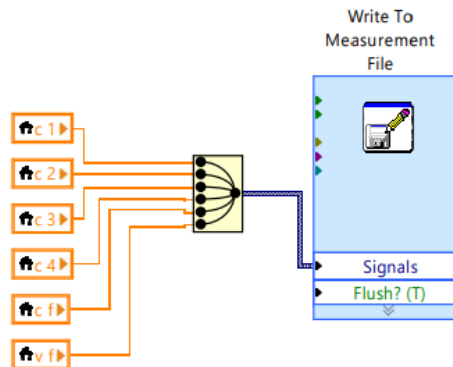


Figura 4. 16 Asignación de atributos a datos para registro

Fuente: Elaborado por el autor

## 5. Etapa de almacenamiento de datos

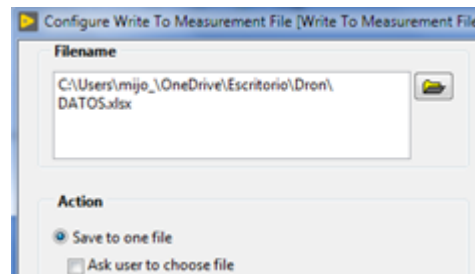
Los seis datos dinámicos obtenidos del subproceso anterior convergen en un bloque fusionador de señales como se muestra en la Figura 4.17, el cual entrega los datos dinámicos al bloque de generación de archivo de registros, éste archivo se puede almacenar en un archivo de tipo texto, binario u hoja de cálculo, para su posterior análisis, en nuestro caso se almacenarán registros del consumo por cada motor y batería.



**Figura 4. 17** Etapa de almacenamiento de datos de medida

**Fuente:** Elaborado por el autor

En la Figura 4.18 se puede observar a manera de ejemplo un directorio raíz en el directorio de la computadora personal en la que se esté trabajando, donde se almacena el archivo digital en forma de hoja de cálculo, resultado de las mediciones y ejecución del programa.



**Figura 4. 18** Directorio raíz para almacenamiento de hoja de calculo

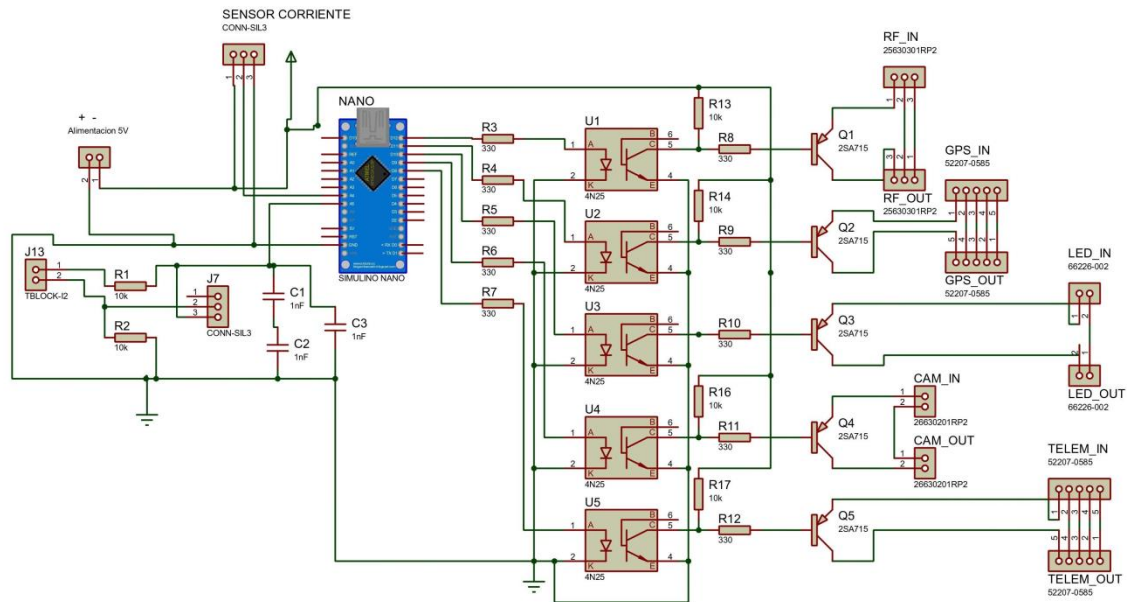
**Fuente:** Elaborado por el autor

#### 4.2.4 Construcción del sistema de control y optimización de energía

#### 4.2.5 Implementación de etapa de hardware

En la Figura 4.19 y Anexo No. 15, se presenta el diagrama circuital implementado en el software Proteus, para la generación de las placas PCB (Printed Circuit Board). Como se puede apreciar el diagrama circuital dispone de un circuito de acondicionamiento de

voltaje, para las señales que ingresan hacia la placa Arduino NANO desde la batería del dron cuadricóptero.



**Figura 4. 19 Diagrama circuital del sistema de control y optimización de energía**

**Fuente: Elaborado por el autor**

El algoritmo implementado en el microcontrolador de la placa Arduino, de la Figura 4.19, tomará decisiones de control automáticas bajo la lógica de la desconexión de sus cinco módulos componentes, para ello se consideró resultados de varias pruebas de vuelo, de las cuales se definió varios parámetros y variables de control, como el tiempo de vuelo, el voltaje y la corriente de la batería, el estado de descarga de la batería, para posteriormente desarrollar un algoritmo que permita ir generando la desconexión según cuatro umbrales de voltaje de la batería, es decir se tomará en cuenta la curva de estado de descarga de la batería.

Es decir cuando el voltaje baja del umbral de 11.2 Vdc, el sistema de control procede a interrumpir el suministro de corriente módulo de iluminación. A continuación la siguiente desconexión es a un voltaje menor o igual a 10.7 Vdc, con el sistema FPV o de cámara, a un nivel de descarga menor o igual a 10.1 Vdc, inmediatamente se interrumpen las comunicaciones vía RF de telemetría con la estación terrena. Restando por controlar dos módulos que son importantes para el desempeño del dron, pero que con ellos el

sistema de control y optimización de energía busca precautelarán la integridad del drone, en vista que su apagado se realiza a pocos centímetros de tierra a un voltaje de 9.45 Vdc, ya que la batería conforme avanza su tiempo de uso, va perdiendo potencia, a su vez dicho voltaje representa el valor mínimo de voltaje que una batería de polímeros de litio (LIPO), puede mantener para no degradar su estructura y no afectar su tiempo de vida útil.

El código de programación desarrollado para la etapa de control se halla disponible en el Anexo No. 6, lo correspondiente a la programación de la placa Arduino NANO.

El tiempo adicional que se obtiene, se ha direccionado a continuar con tiempo mayor de vuelo que el inicial, que en nuestro caso es de aprox. 1.06 minutos, el mismo que puede direccionarse un mayor tiempo de vuelo o para precautelar la integridad del drone con un aterrizaje controlado, un aterrizaje de emergencia o su rescate.

La PCB de doble lado generada en el software Proteus, se presenta en la Figura 4.20 y 4.21, la cual tiene unas dimensiones de 9.67 cm x 7.38 cm y representa un peso de 70 gr, con la configuración más óptima de tamaño.

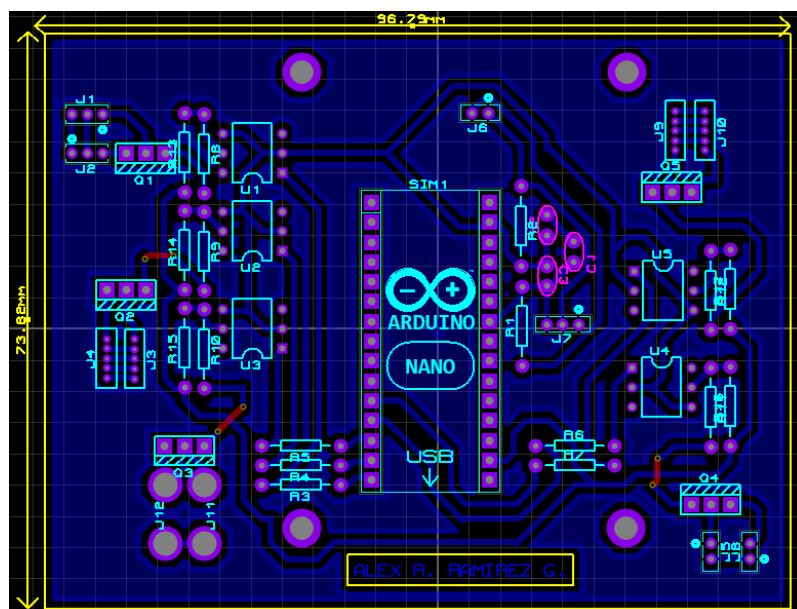
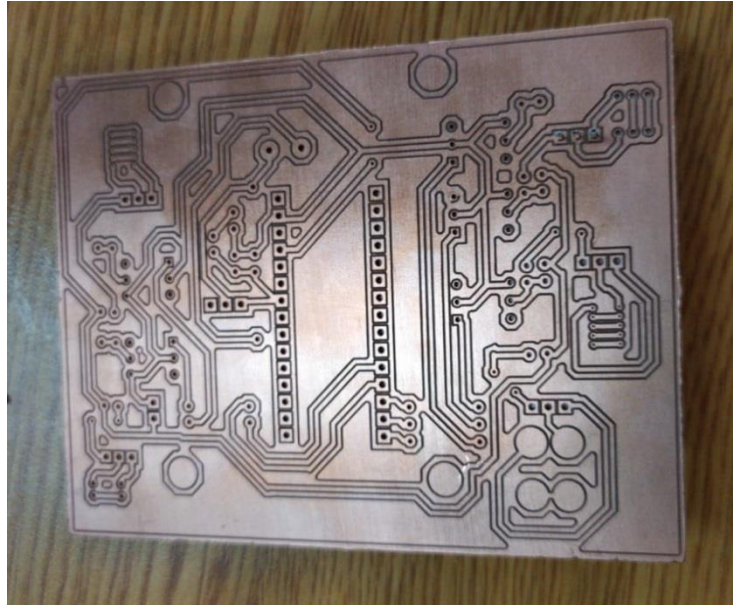


Figura 4. 20 Diseño PCB de control en software Proteus

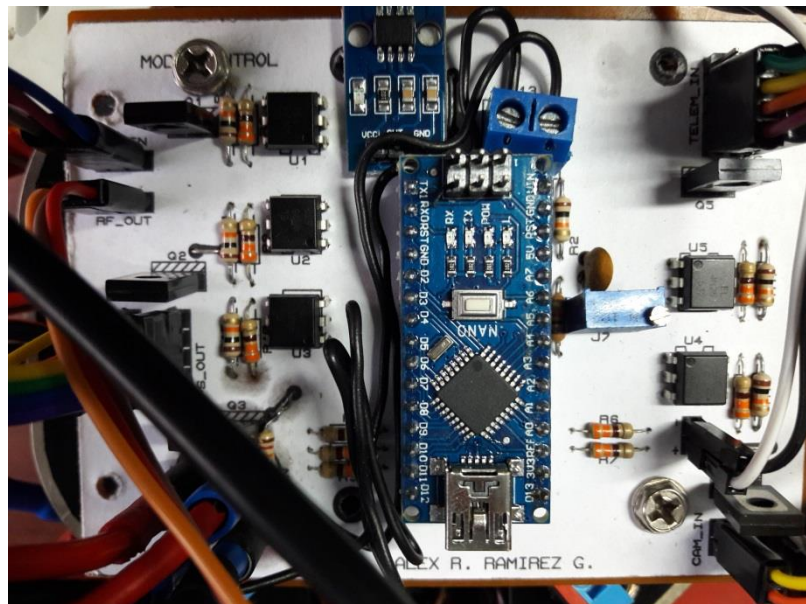
Fuente: Elaborado por el autor



**Figura 4. 21 Circuito impreso de sistema de sistema de control**

**Fuente: Elaborado por el autor**

En las Figura 4.22, se muestra la placa de control y optimización creada con todos sus componentes ya ensamblados, mientras que en la Figura 4.23, se presenta su adaptación a la estructura del dron utilizado.



**Figura 4. 22 Placa de sistema de control y optimización ensamblada**

**Fuente: Elaborado por el autor**





**Figura 4. 23 Tarjeta electrónica de control instalada en el Drone F450-APM 2.8 de prueba**

**Fuente: Elaborado por el autor**

En la Tabla 4.2, se detallan los materiales, elementos y dispositivos electrónicos que fueron necesarios para la conformación de la tarjeta electrónica de control y optimización.

**Tabla 4. 2. Materiales, elementos y dispositivos electrónicos de la tarjeta de control y optimización**

ELEMENTO	ESPECIFICACIÓN	UNID
Sensor de corriente	ACS 712 (20A)	1
Opto acoplador	CI 4N25	5
Borneras de CI	Dobles	2
Resistencias ½ W	1kΩ +/- 5%	1
	330Ω +/- 5%	10
	10kΩ +/- 5%	6
Transistores PNP	BD136	5
Potenciómetro de precisión	1kΩ	1
Capacitores cerámicos	1nF	3
PCB doble lado	10x10 cm	1
Cloruro Férrico	100 gr	1
Placa de Arduino	NANO	1
Socket Arduino nano	30 pines	1
Sockets Tipo espadín	30 pines	1
Cable comunicación serial	Mini USB a USB	1
Cable gemelo de alimentación	12 AWG	2
Tornillos	06_32	4
Cautín	40W	1
Tubo de suelda 60/40	10 gr	.1
Lavacara	10 cm diámetro	1

**Fuente: Elaborado por el autor**

## 4.2.6 Implementación de la etapa de software

### a. Programación de módulo Arduino NANO

Dentro del entorno de programación Arduino IDE, como se muestra en la Figura 4,24, se elige la placa de desarrollo Arduino NANO, que se determinó por las ventajas, versatilidad, dimensiones, peso y potencia que posee.

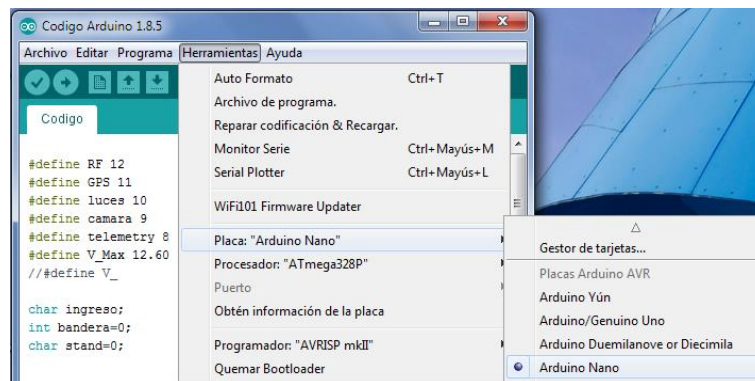


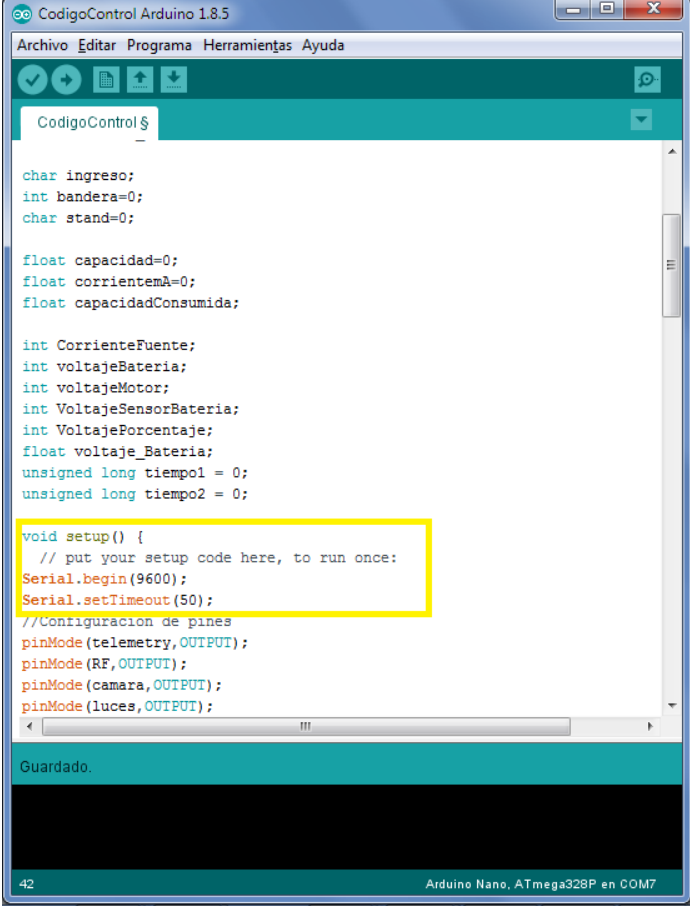
Figura 4. 24 Selección de placa Arduino NANO

Fuente: Elaborado por el autor

La implementación del código de programación se definió en cuatro etapas, la que se desarrolló en función del diagrama de bloques presentado en el Anexo No. 12 y que se describe a continuación.

### 1. Inicializar entradas y comunicación

Como se puede observar en la Figura 4.25, el programa da inicio con la definición de varias variables entre ellas las correspondientes a los cinco módulos del drone que se van a controlar, junto con la señal de voltaje/corriente de la batería que se va a evaluar en cada instante para desarrollar la lógica de control de apagado de los módulos y optimización de energía que está sujeta a varios umbrales de voltaje considerando los módulos del drone que puede prescindir y de mayor consumo. Adicional se inicializan varios puertos digitales tanto como entrada o salida.



```
CodigoControl Arduino 1.8.5
Archivo Editar Programa Herramientas Ayuda

CodigoControl $

char ingreso;
int bandera=0;
char stand=0;

float capacidad=0;
float corrienteA=0;
float capacidadConsumida;

int CorrienteFuente;
int voltajeBateria;
int voltajeMotor;
int VoltajeSensorBateria;
int VoltajePorcentaje;
float voltaje_Bateria;
unsigned long tiempo1 = 0;
unsigned long tiempo2 = 0;

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  Serial.setTimeout(50);
  //Configuracion de pines
  pinMode(telemetry, OUTPUT);
  pinMode(RF, OUTPUT);
  pinMode(camara, OUTPUT);
  pinMode(luces, OUTPUT);
}
```

Guardado.

42 Arduino Nano, ATmega328P en COM7

**Figura 4. 25 Definición de variables de control y comunicación serial Arduino - PC**

**Fuente:** Elaborado por el autor

El programa principal inicia definiendo una velocidad de comunicación de 9600 baudios definida para la conexión con la PC a través del puerto serial de la placa Arduino NANO, para la programación de control que se alojada en el microcontrolador del sistema o para realizar un monitoreo de consumo online de voltaje, corriente, potencia, durante el tiempo de vuelo del drone F450 APM 2.8, con cable en el laboratorio, sobre el software LabView. La programación para éste apartado se muestra en la Figura 4.25.

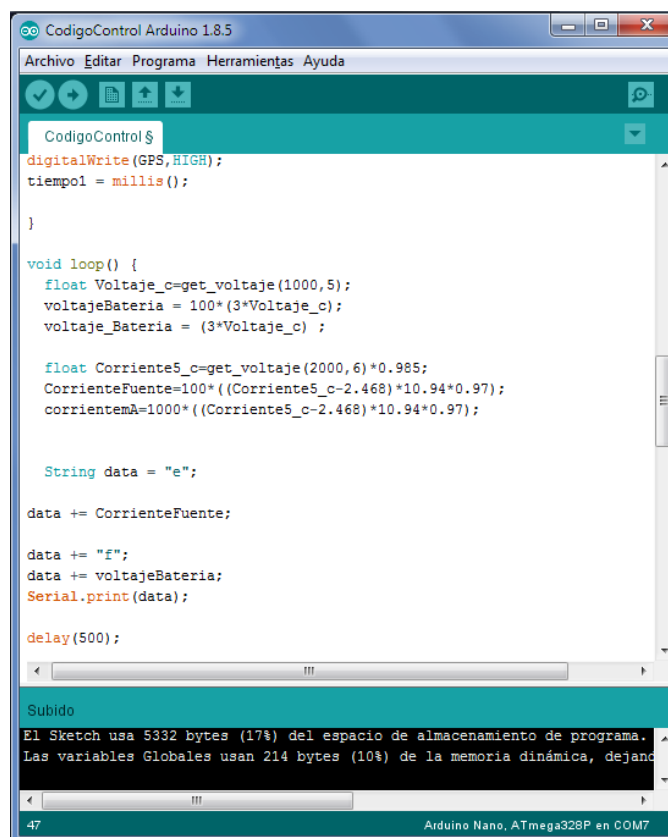
## 2. Adquirir mediciones

Dentro del *void loop* del programa se llama a dos funciones *get* para la adquisición de los datos de voltaje y corriente de la batería, la que toma lectura de los pines analógicos (A5, A6) a razón de 1000 y 2000 muestras respectivamente. A continuación se presentan dos líneas de sintaxis, relacionada con lo descrito.

$$\text{float Voltaje\_c}=\text{get\_voltaje}(1000,5);$$

$$\text{float Corriente5\_c}=\text{get\_voltaje}(2000,6)*0.985;$$

Líneas de programación que corresponden a la obtención de un valor medio del total de muestras de la señal de voltaje y corriente que entrega la batería, valor que posteriormente se lo acondiciona tomando en consideración la ecuación (7) para el caso de la corriente y la ecuación (8) para caso del voltaje, descritas en el capítulo anterior. La programación implementada en este apartado se la puede apreciar en la Figura 4.26.



```

CodigoControl Arduino 1.8.5
Archivo Editar Programa Herramientas Ayuda
CodigoControl $
digitalWrite(GPS,HIGH);
tiempo1 = millis();
}

void loop() {
  float Voltaje_c=get_voltaje(1000,5);
  voltajeBateria = 100*(3*Voltaje_c);
  voltaje_Bateria = (3*Voltaje_c) ;

  float Corriente5_c=get_voltaje(2000,6)*0.985;
  CorrienteFuente=100*((Corriente5_c-2.468)*10.94*0.97);
  corrientemA=1000*((Corriente5_c-2.468)*10.94*0.97);

  String data = "e";

  data += CorrienteFuente;

  data += "f";
  data += voltajeBateria;
  Serial.print(data);

  delay(500);
}

Subido
El Sketch usa 5332 bytes (17%) del espacio de almacenamiento de programa.
Las variables Globales usan 214 bytes (10%) de la memoria dinámica, dejand
47 Arduino Nano, ATmega328P en COM7

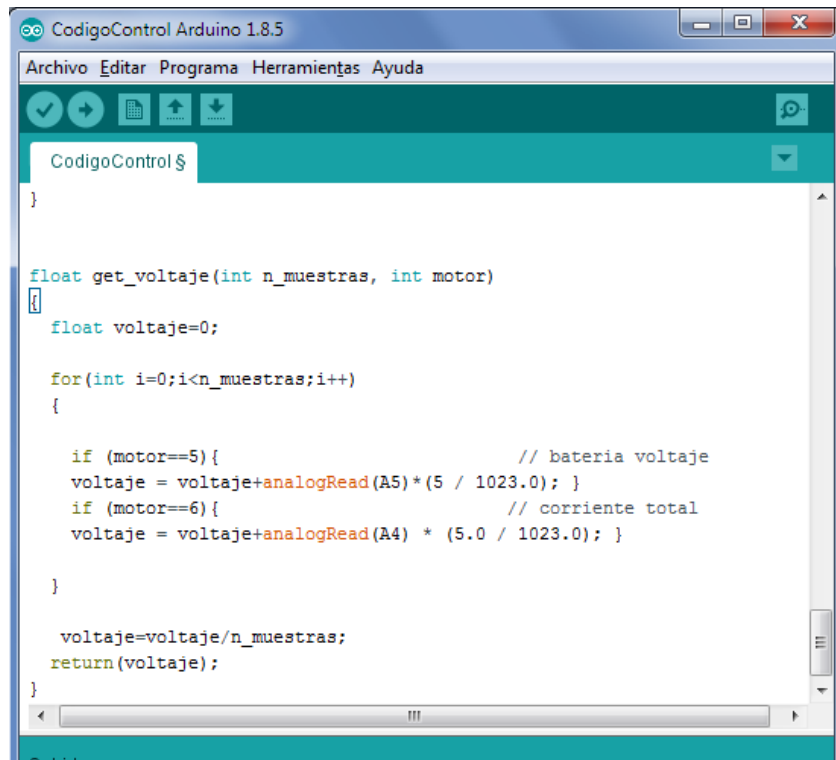
```

Figura 4. 26 Etapa de medición y conformación de trama

Fuente: Elaborado por el autor

El dato de corriente y voltaje obtenido posterior al debido acondicionamiento, se imprime a modo de *String* en el puerto serial del Arduino, para su lectura por el software LabView.

A través de un bucle *for* y un condicional *if* anidado, se realiza el proceso de acondicionamiento de los datos obtenidos desde la batería, como se puede ver en la Figura 4.27.

The image shows a screenshot of the Arduino IDE interface. The window title is "CodigoControl Arduino 1.8.5". The menu bar includes "Archivo", "Editar", "Programa", "Herramientas", and "Ayuda". The toolbar contains icons for file operations and execution. The main editor area shows the following C++ code:

```
CodigoControl $
}

float get_voltaje(int n_muestras, int motor)
{
    float voltaje=0;

    for(int i=0;i<n_muestras;i++)
    {

        if (motor==5){ // bateria voltaje
            voltaje = voltaje+analogRead(A5)*(5 / 1023.0); }
        if (motor==6){ // corriente total
            voltaje = voltaje+analogRead(A4) * (5.0 / 1023.0); }

    }

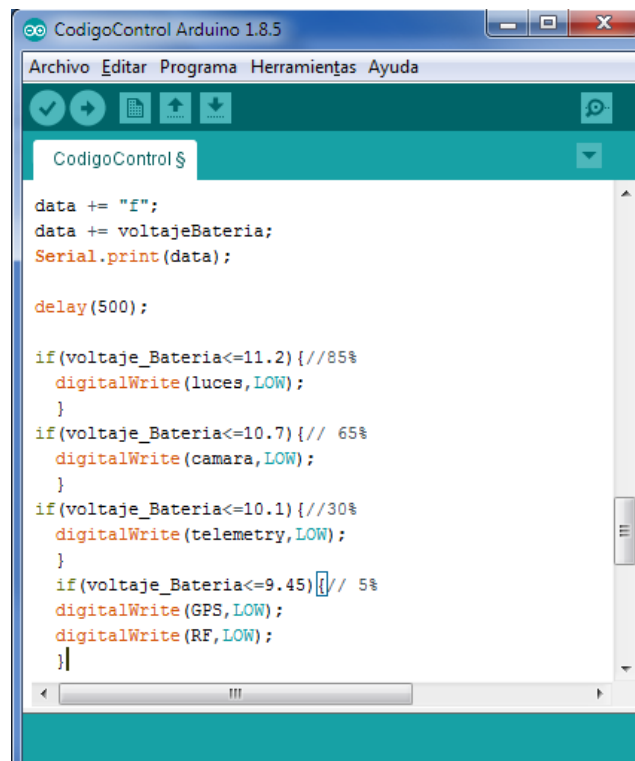
    voltaje=voltaje/n_muestras;
    return(voltaje);
}
```

**Figura 4. 27 Acondicionamiento de la señal de voltaje de la batería**

**Fuente: Elaborado por el autor**

### 3. Control y optimización

La programación elaborada para el proceso de toma de decisiones y gestión de la energía para su optimización, con el apagado paulatino de los cinco módulos prescindibles, se lo realiza según cuatro umbrales de voltaje según el módulo de mayor consumo, todo esto se empieza a ejecutar cuando la curva de descarga de la batería empieza a ser lineal en un valor de voltaje promedio de 11.5 Vdc, como se puede apreciar en la Figura 4.29, dato definido de las mediciones obtenidas del desempeño de la batería del drone en vuelo autónomo sin control.



```
CodigoControl Arduino 1.8.5
Archivo Editar Programa Herramientas Ayuda

CodigoControl $

data += "f";
data += voltajeBateria;
Serial.print(data);

delay(500);

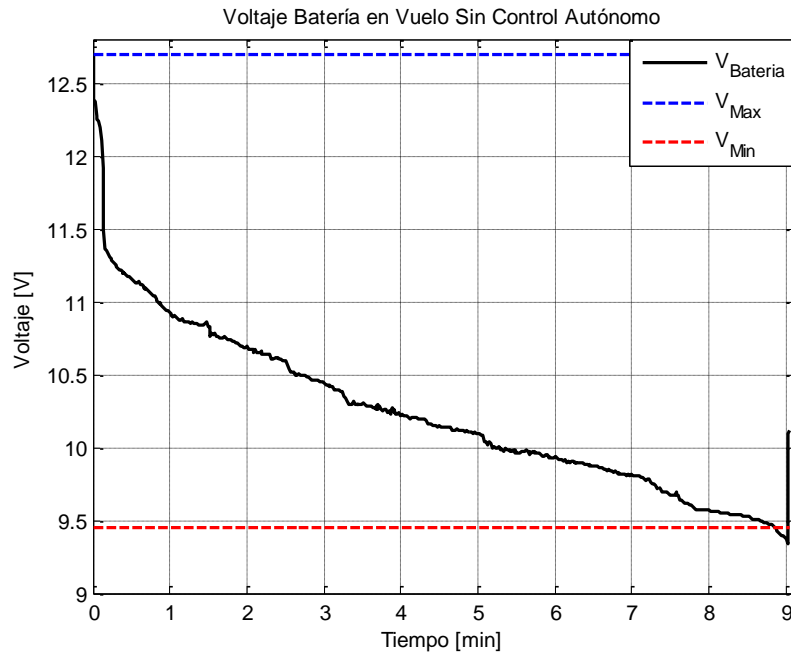
if(voltaje_Bateria<=11.2){//85%
  digitalWrite(luces,LOW);
}
if(voltaje_Bateria<=10.7){// 65%
  digitalWrite(camara,LOW);
}
if(voltaje_Bateria<=10.1){//30%
  digitalWrite(telemetry,LOW);
}
if(voltaje_Bateria<=9.45){// 5%
  digitalWrite(GPS,LOW);
  digitalWrite(RF,LOW);
}
```

Figura 4. 28 Algoritmo de control y optimización de energía

Fuente: Elaborado por el autor

Para el desarrollo del algoritmo de control y optimización de energía se tomó en consideración los siguientes parámetros definidos de las especificaciones técnicas y funcionamiento de la batería LIPO que se utiliza en el dron cuadricóptero de este proyecto, todo esto contrastado en la Figura 4.29, gráfica resultado de las mediciones previas a la implementación del sistema de control y optimización.

- **V. Max** = 12.7 V (Totalmente cargada)
- **V. Min** = 9.45 V (Batería sin capacidad, nivel crítico)
- **V. Trabajo** = 11.5 V (Voltaje a partir del cual la batería del dron desciende linealmente hasta 9.45 V)



**Figura 4. 29** Curva de descarga de batería sin algoritmo de control y optimización

**Fuente:** Elaborado por el autor

El microcontrolador gestiona el intercambio de información con un software residente en un computador personal a través de un archivo ejecutable con extensión `.vi` de LabView a través de una comunicación serial una vez se haya iniciado la sesión. A través de LabView se pueden almacenar registros de datos de consumo voltaje, corriente y potencia durante el funcionamiento del dron en una hoja de cálculo, en el directorio que el usuario defina. Toda la programación desarrollada en el entorno de desarrollo Arduino IDE, se muestra en el Anexo N. 6

### **b. Programación en LabView (PC)**

En el Anexo No. 7, se presenta una vista completa que incluye todos los siguientes subprocesos implementados en el software LabView, que corresponden a las etapas consiguientes de recepción, decodificación, formato, almacenamiento y presentación de los datos obtenidos (voltaje, corriente y potencia) de la batería con el sistema de control y optimización de energía implementado en el dron cuadricóptero.

A continuación se enumeran y describen cada uno de los siguientes subprocesos mencionados:

## 1. Etapa de inicio de sesión y recepción de datos

A través del bloque *visa*, se da inicio a la etapa, con la apertura del puerto del Arduino NANO con el que se va a iniciar la sesión de comunicación serial, se definen varios parámetros de comunicación en el bloque *visa serial* como: frecuencia de muestreo 9600 baudios, número de bits a transmitir 8 bits, el tipo de paridad (sin paridad).

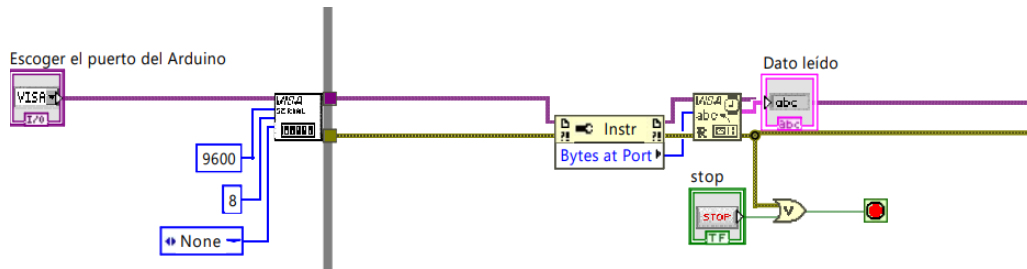


Figura 4. 30 Etapa de inicio de sesión y recepción de datos LabView-Arduino NANO

Fuente: Elaborado por el autor

En la Figura 4.30, se encuentra un nodo de propiedades, el que mantiene parámetros del inicio de la sesión de comunicación. El bloque *Visa Read*, identifica el número de bytes presentes en su buffer de salida para continuar con la decodificación.

## 2. Etapa de Decodificación

Para ésta etapa de decodificación, los datos recibidos en cadena ingresan al bloque *match pattern*, uno tras de otro, con el siguiente formato (letra-dato1, letra-dato2), estructurado con los caracteres:

e CF f VB.

Dónde:        **e y f**            letras que separan a los datos obtenidos a enviarse.  
               **CF – VB:**        Corriente de fuente y voltaje de la batería, como se puede apreciar en la Figura 4.31.



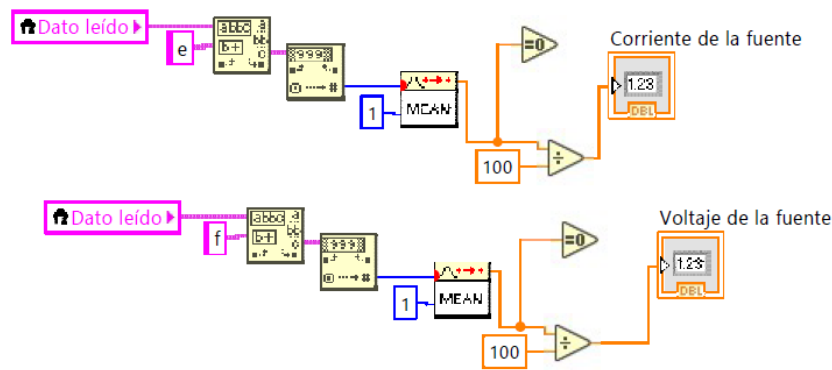


Figura 4. 31 Etapa de decodificación de datos en LabView

Fuente: Elaborado por el autor

### 3. Etapa de presentación o visualización

En esta etapa se representa el valor del voltaje y corriente que circula en ese momento en el punto de medición de la batería. El bucle de realimentación permite almacenar y presentar el dato medido en la interfaz gráfica HMI, para visualizar el comportamiento de la batería en tiempo real, según Figura 4.32.

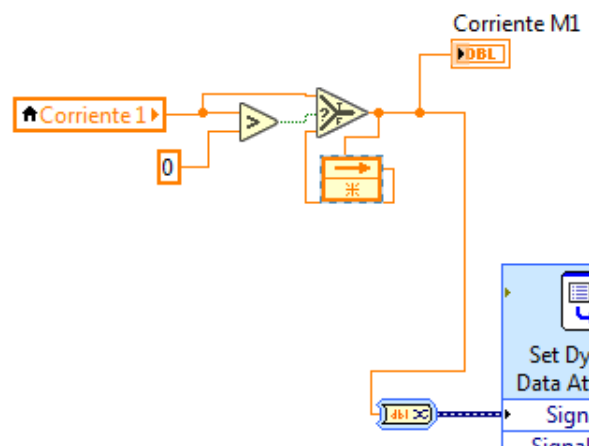


Figura 4. 32 Etapa de verificación del dato obtenido para visualización HMI

Fuente: Elaborado por el autor

A través de la interfaz hombre máquina HMI, se puede observar el comportamiento de la batería del dron de acuerdo a una curva que reproduce el software LabView, a través del bloque *waveform chart*, como se presenta en la Figura 4.33.



Figura 4. 33 Interfaz HMI, para visualización de medidas

Fuente: Elaborado por el autor

#### 4. Formato

El bloque de formato permite dar atributos dinámicos a los datos obtenidos para almacenarlos como una hoja de calculo y de ésta forma disponer de un registro o historial de mediciones para posterior análisis. En la Figura 4.34, se muestra el proceso antes descrito.

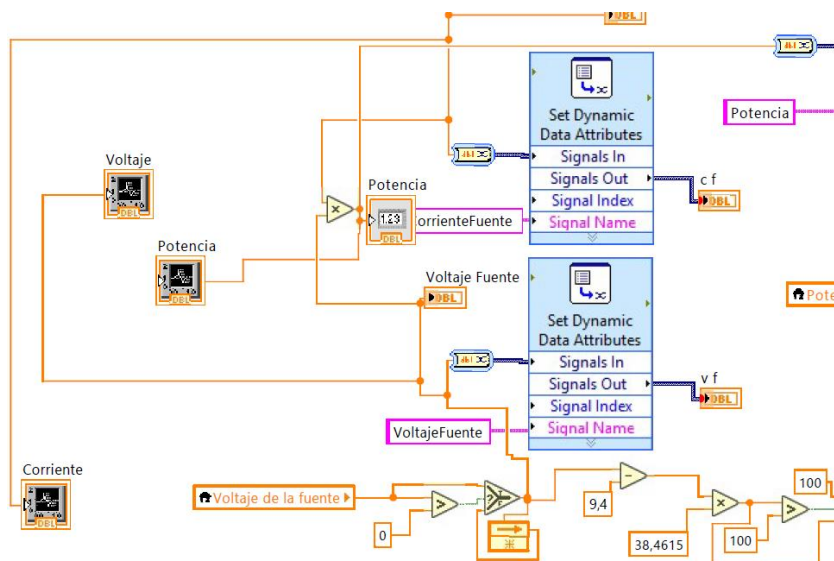


Figura 4. 34 Asignación de atributos de dato dinámico para registro

Fuente: Elaborado por el autor

## 5. Etapa de almacenamiento de datos

Los tres datos dinámicos de potencia, corriente y voltaje de la batería obtenidos del subproceso anterior ingresan al bloque fusionador de señales previo a la generación de la hoja de cálculo, para su tratamiento y análisis, como se indica en la Figura 4.35.

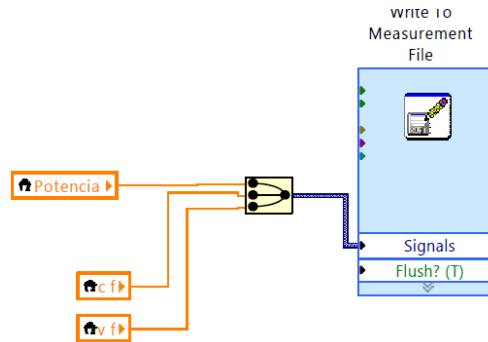


Figura 4. 35 Generación de hoja de cálculo, con datos medidos

Fuente: Elaborado por el autor

En las Figura 4.36 se presenta una captura de pantalla de la hoja de cálculo que se genera y se almacena en un directorio definido por el usuario en una computadora personal.

Time	Potencia	Corriente	Voltaje
29/2/2020 14:20:02,664	4,04	1,01	4
29/2/2020 14:20:03,051	4,0198	1,01	3,98
4 29/2/2020 14:20:03,355	4,0198	1,01	3,98
5 29/2/2020 14:20:03,862	4,0198	1,01	3,98
6 29/2/2020 14:20:04,165	4,0198	1,01	3,98
7 29/2/2020 14:20:04,471	4,0198	1,01	3,98
8 29/2/2020 14:20:04,798	4,0198	1,01	3,98
9 29/2/2020 14:20:05,169	4,0198	1,01	3,98
10 29/2/2020 14:20:05,460	4,0198	1,01	3,98
11 29/2/2020 14:20:05,765	16,419	1,3	12,63

Figura 4. 36 Directorio y datos recolectados en hoja de cálculo

Fuente: Elaborado por el autor

### 4.3 Pruebas de funcionamiento

#### 4.3.1 Sistema de medición de datos de consumo

El circuito electrónico desarrollado final instalado y adaptado al drone, para mediciones antes y después de la implementación se presenta en las Figuras 4.37, 4.38 y 4.39, cabe recordar que este sistema no consume recursos de batería del drone, su alimentación se da a través del puerto USB-serial por el que se realizan las comunicaciones con la placa de Arduino y la PC en el laboratorio.



Figura 4. 37 Pruebas de funcionamiento de sistema de medición

Fuente: Elaborado por el autor



Figura 4. 38 Interfaz HMI online de sistema de medición (rotores y batería)

Fuente: Elaborado por el autor

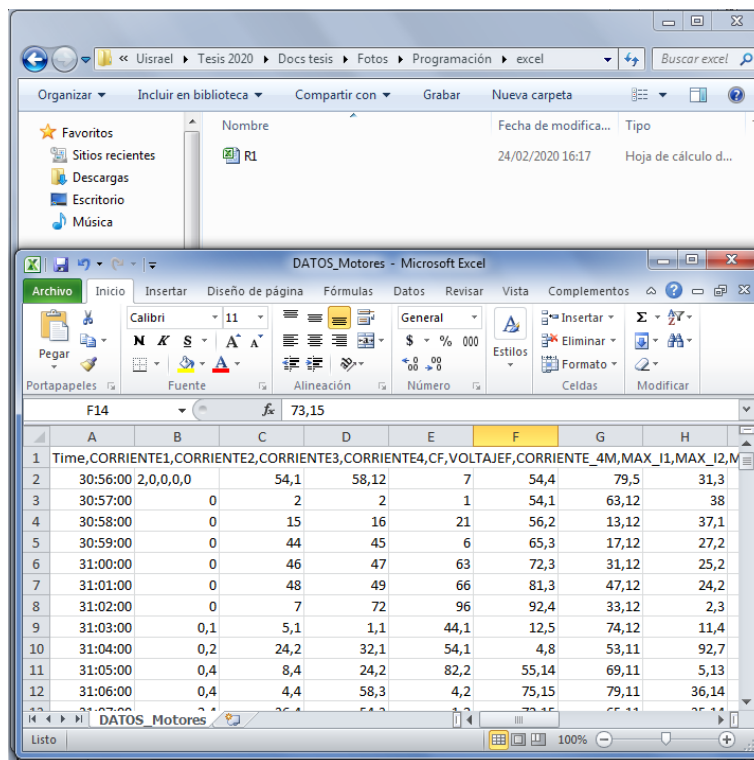


Figura 4. 39 Directorio y datos recolectados en una hoja de cálculo

Fuente: Elaborado por el autor

### 4.3.2 Sistema de control y optimización de energía

El circuito electrónico desarrollado final instalado y adaptado al drone, para el sistema de control y optimización de energía se presenta en las Figuras 4.40, 4.41 y 4.42, hay que puntualizar que el sistema es autónomo y consume recursos de batería, el control lo realiza de manera autónoma. A través del puerto USB-serial en conjunto con el software LabView se pueden observar las mediciones y almacenar en tiempo real registros de datos de voltaje, corriente y potencia durante el tiempo de vuelo del drone en el laboratorio.



Figura 4. 40 Pruebas de funcionamiento de sistema de medición

Fuente: Elaborado por el autor

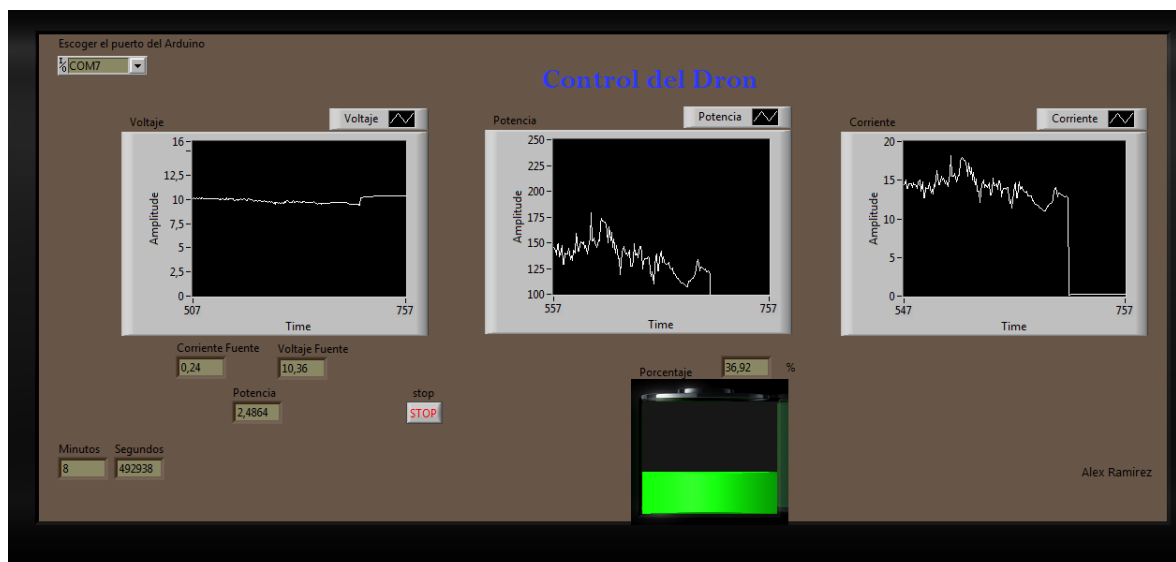
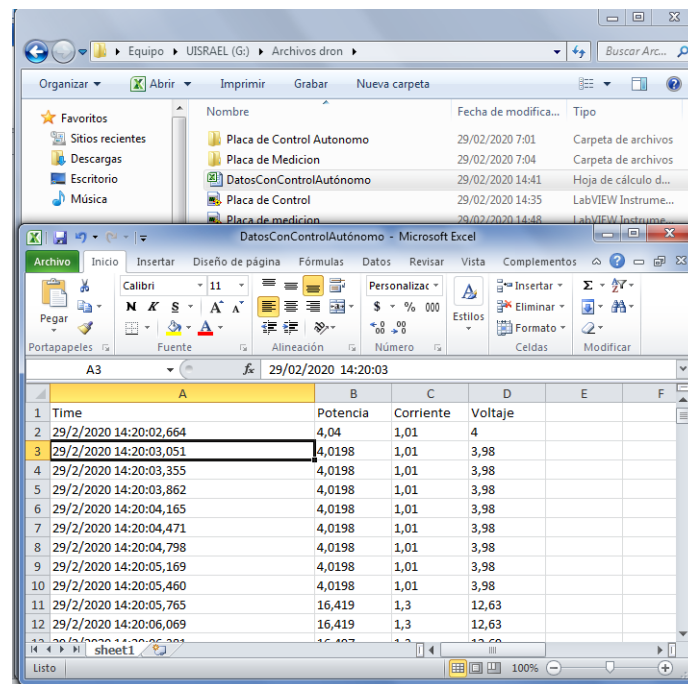


Figura 4. 41 Interfaz HMI online de sistema de control y optimización

Fuente: Elaborado por el autor



**Figura 4. 42 Directorio y datos recolectados en hoja de calculo**

**Fuente: Elaborado por el autor**

#### 4.4 Análisis de resultados

El análisis de resultados se lo realizará en base a los registros almacenados en la hoja de cálculo, obtenida de las mediciones antes y después de la implementación, utilizando la funcionalidad del software MatLab para generar gráficas a partir de los datos e información obtenida con el software LabView y los sistemas implementados.

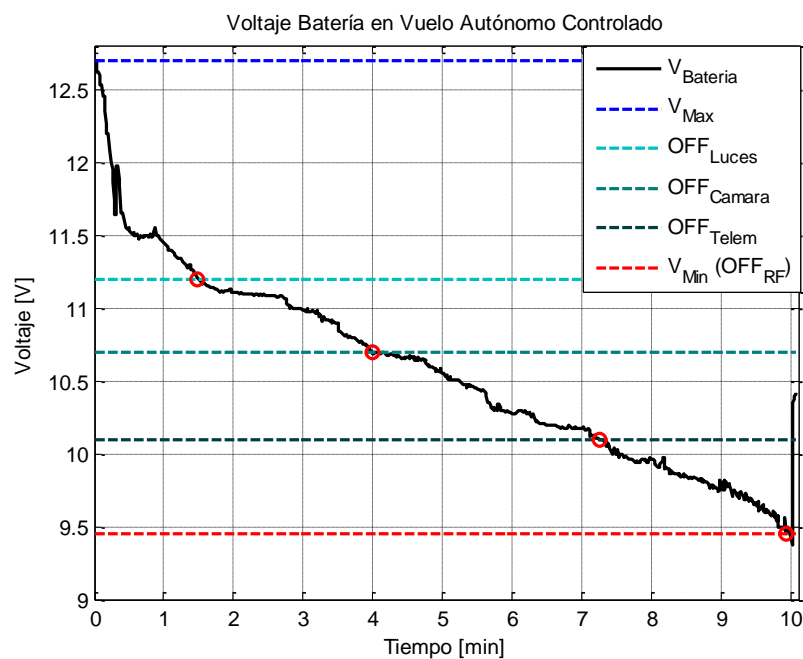
Una vez desarrollado todo el sistema, se han procesado las tablas generadas desde la interfaz en el software Labview, de esta forma las gráficas obtenidas y procesadas se describen de acuerdo al siguiente orden de interés.

- DATOS\_Control.csv
- DATOS\_SinControl.csv
- DATOS\_Medicion.csv

Las tablas antes descritas fueron importadas hacia el software MatLab, herramienta muy adecuada para la generación de gráficas de datos cuyos resultados permiten evidenciar las conclusiones correspondientes a la presente investigación. El código importado y desarrollado en Matlab se puede observar en el Anexo No. 8, del presente documento.

#### 4.4.1 Sistema de control y optimización de energía

Así, las gráficas obtenidas se presentan a continuación:



**Figura 4. 43 Voltaje batería en vuelo autónomo controlado**

**Fuente: Elaborado por el autor**

En la curva de la Figura 4.43, se muestra el proceso de descarga de la batería con la implementación del sistema de control propuesto. Puede observarse también cada uno de los puntos donde se realiza el corte de los módulos que componen el dron. (11.2 V- módulo de luces, 10.7 V- sistema FPV o cámara, 10.1 V- telemetría y 9.45 V- GPS y RF, aterrizaje)

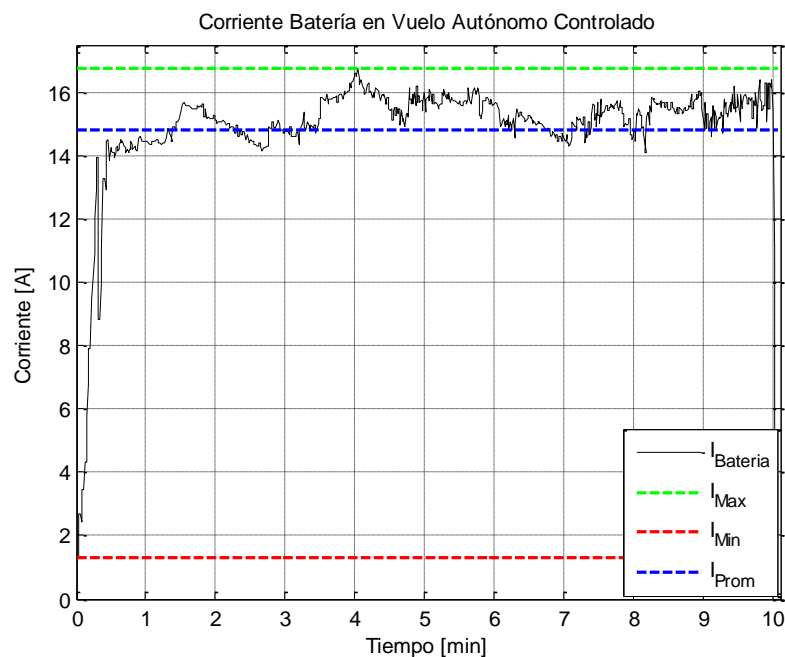
El descenso brusco de voltaje durante el primer minuto se explica considerando la etapa de arranque propia del dron, donde la corriente consumida se dispara hasta conseguir



el vuelo estacionario. A partir del primer minuto, puede observarse un proceso de descarga con características lineales propio del comportamiento de las baterías de polímero de litio LIPO.

Los puntos de color rojo indican el tiempo y el valor de voltaje con los que se realizaron cada uno de los cortes. Es importante observar que, tras la ejecución del corte, la pendiente de descarga de la batería tiende a disminuir, es decir, se vuelve más horizontal sin perder evidentemente la tendencia al descenso.

Finalmente, en el último tramo puede apreciarse que, tras alcanzar el punto crítico de voltaje se comanda el apagado del dron, ante lo cual inmediatamente la curva de voltaje tiende a subir dado que ya no se demanda la entrega de energía. Esto no significa que la batería nuevamente se encuentra en condiciones de trabajo, lo cual puede verificarse intentando iniciar nuevamente el dron y observando una caída significativa del voltaje hacia el valor crítico en cuestión de segundos.



**Figura 4. 44 Corriente de batería en vuelo autónomo controlado**

**Fuente: Elaborado por el autor**

La gráfica de la Figura 4.44 demuestra la evolución de corriente, ésta variable tiende a mantenerse constante, lo cual resulta evidente considerando la necesidad de trabajar en vuelo estacionario y bajo condiciones de perturbación prácticamente nulas. Así, las variaciones de corriente observadas responden a la necesidad de conservar la altitud a pesar del proceso de descarga propio de la batería, lo que implica aumentar el valor de la señal PWM asignada a los rotores a fin de exigir mayor potencia de parte de la batería.

El valor mínimo observado evidentemente no es 0 puesto que antes del despegue el dron se encuentra alimentando a la totalidad del sistema, lo que compete al controlador principal APM, GPS, telemetría, RF, Sistema FPV o cámara y luces, consumiendo alrededor de 1.4 A.

Finalmente puede observarse que la evolución inicial de la corriente tiene una pendiente significativamente elevada, lo que significa que se exige una descarga de energía importante a la batería, lo cual explica la necesidad de trabajar con baterías diseñadas con características de alta entrega de energía en tiempos reducidos (baterías LIPO).

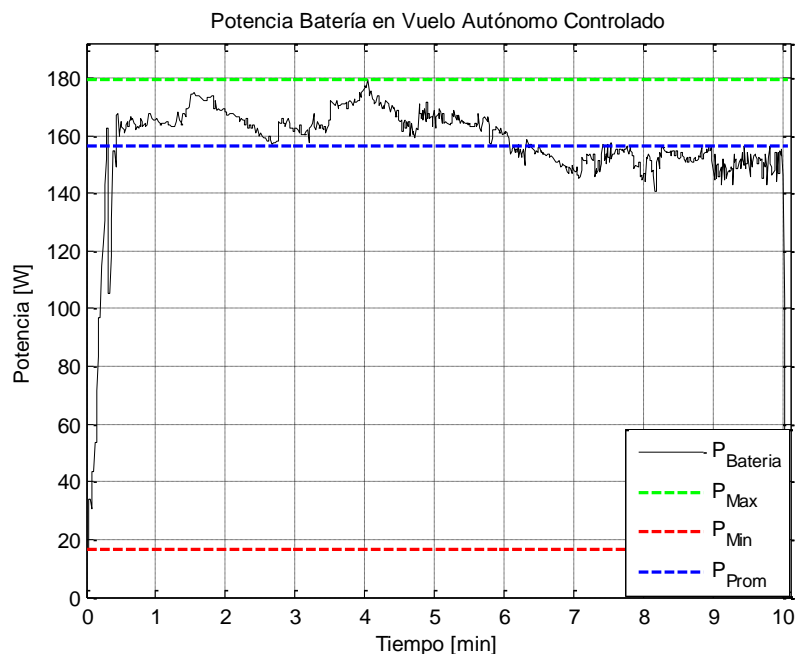
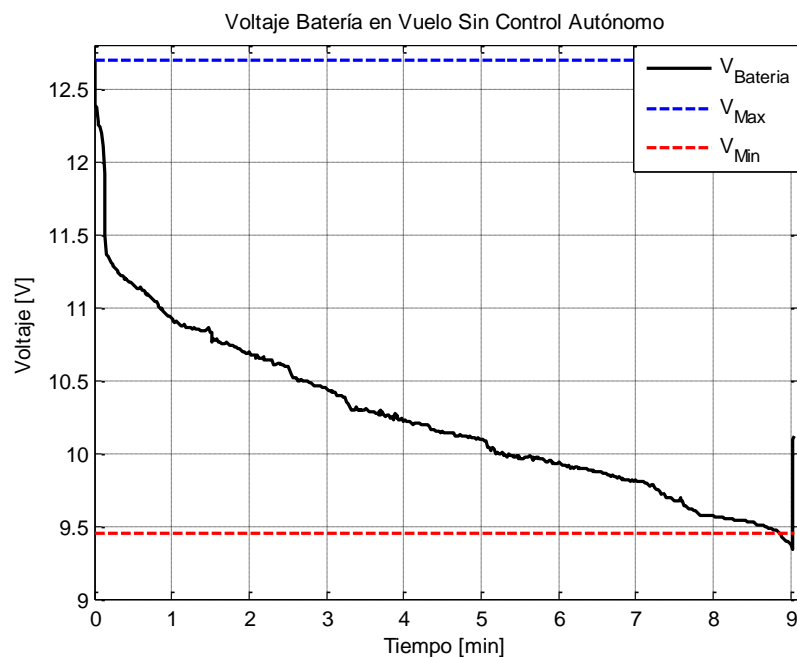


Figura 4. 45 Potencia batería en vuelo autónomo controlado

Fuente: Elaborado por el autor

La curva de potencia indicada en la Figura 4.45, para el esquema de vuelo autónomo controlado demuestra precisamente una tendencia al descenso. Esto se explica considerando la naturaleza de la corriente a permanecer constante, pero el voltaje tendiendo al descenso. Esto permite reconocer que a pesar de que la corriente tiende a mantenerse constante, la entrega de potencia de la batería tiende siempre a disminuir, tal y como lo demuestra la Figura 4.45. Esto a su vez explica que cuando la batería no es capaz de suministrar energía suficiente al dron, este tiende a descender, o si se encuentra a nivel del suelo sea incapaz de arrancar.

Por estas razones precisamente se ha fijado un valor de estado crítico de la batería, ante la cual se ordena el aterrizaje para precautelar la integridad tanto del dron como de la batería.



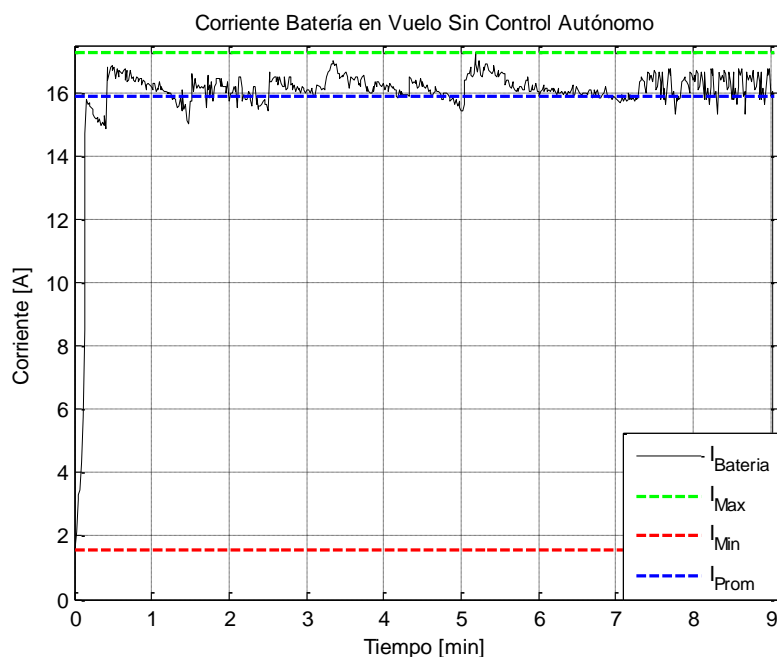
**Figura 4. 46 Voltaje de batería en vuelo sin control autónomo**

**Fuente: Elaborado por el autor**

La Figura 4.46 presenta la curva de descarga de voltaje sin el modelo de control autónomo, demuestra una tendencia mucho mayor a la pérdida de voltaje, lo cual se puede observar desde el arranque del dron. De este modo, es evidente que el tiempo de autonomía se ve

reducido a 1 minuto menos que el tiempo conseguido a través del modelo de control y optimización propuesto.

A pesar de la tendencia a mayor descarga de la batería, puede observarse que el comportamiento de la curva de descarga es idéntico a aquél del modelo de vuelo controlado. Así, se observa en la etapa inicial una descarga abrupta debida al proceso de arranque, una descarga con característica lineal durante todo el resto de la etapa y finalmente una pequeña recuperación de voltaje tras el comando de aterrizaje del dron.



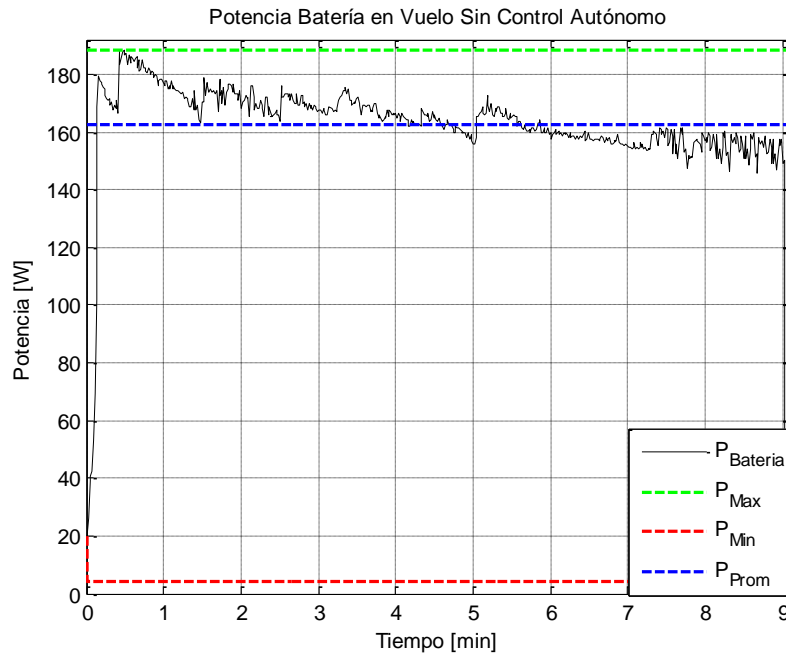
**Figura 4. 47 Corriente de batería en vuelo sin control autónomo**

**Fuente: Elaborado por el autor**

En la Figura 4.47, gráfica de corriente de la batería sin control se puede notar con mayor claridad la tendencia de la corriente a mantenerse constante, con inevitables fluctuaciones debido a la necesidad de compensar la altura del dron ante la descarga progresiva de la batería.

Puede notarse que del mismo modo que con voltaje, el comportamiento de la corriente es idéntico que con la prueba de control Autónomo, sin embargo las magnitudes

de dichas variables son considerablemente diferentes, lo cual explica la menor autonomía del presente caso.

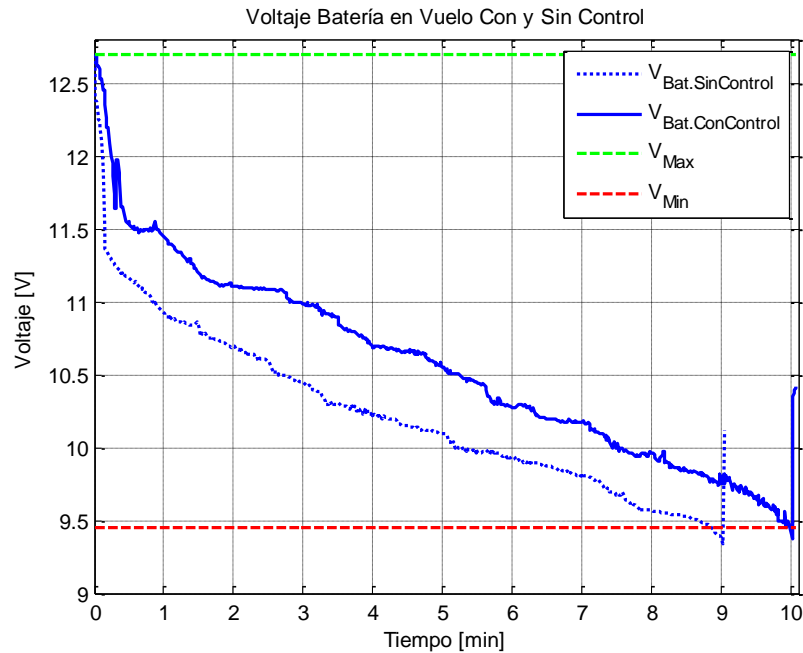


**Figura 4. 48 Potencia batería en vuelo sin control automático**

**Fuente: Elaborado por el autor**

El comportamiento de la potencia para el caso presente de la Figura 4.48, se demuestra precisamente la descarga progresiva de la batería, hasta alcanzar un punto en el que el suministro de potencia de la batería hacia el dron resulta insuficiente para mantenerlo en el aire.

Es importante anotar también que si después de realizada la prueba de vuelo se intenta arrancar nuevamente al dron, se observaría que la batería entrega toda la potencia a su alcance, pero el proceso de descarga sería inmediato. Este no es un procedimiento recomendable debido a que aumenta el desgaste aplicado sobre la batería y su degradación, ante lo cual el control de aterrizaje propuesto resulta ser apropiado.



**Figura 4. 49 Voltaje de batería en vuelo con y sin control**

**Fuente: Elaborado por el autor**

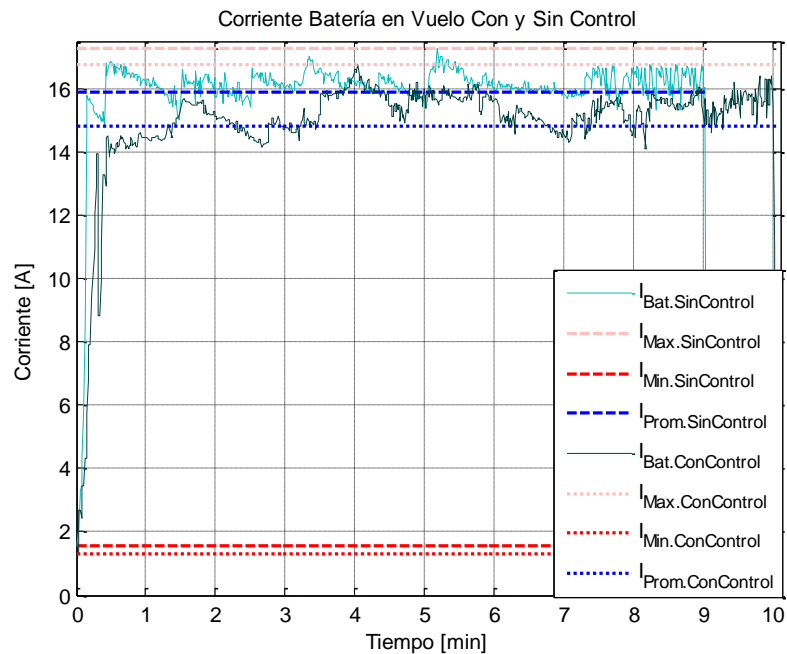
La Figura 4.49, demuestra con claridad que se ha optimizado energía en términos de alcanzar un mayor tiempo de autonomía al gestionar de manera adecuada los tiempos de encendido y apagado de cada uno de los módulos del dron. Tal y como se había estimado, con lo que se ha verificado el ahorro en tiempo de autonomía en un 1 minuto.

Al considerar este resultado en el marco del tiempo de autonomía del dron, puede decirse que conseguir 1 minuto adicional como tiempo de vuelo en estado estacionario es un resultado importante para proveer un aterrizaje controlado, mantener la integridad y extender la vida útil del sistema.

Por otro lado, innegablemente el tiempo de vuelo “ganado” todavía no es suficiente para ser considerado significativo, lo cual afirma la importancia de la problemática de extender el tiempo de vuelo de un dron atacando otros puntos del sistema tales como la batería, o la búsqueda de nuevos métodos de energización.

La naturaleza de las curvas que se han obtenido no variará si se reproducen las pruebas con una batería diferente. La variación que podrá notarse radica en el tiempo de

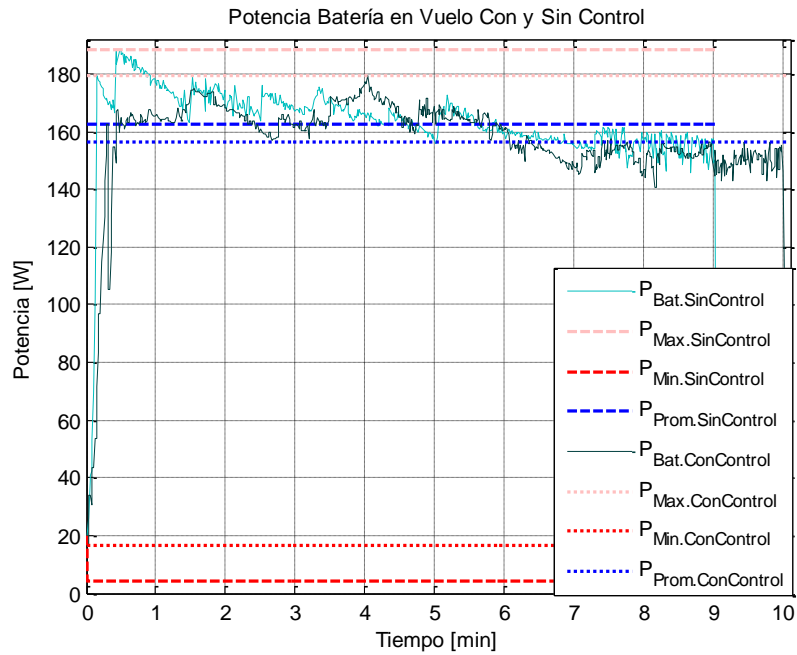
autonomía conseguido, garantizando evidentemente la ganancia de tiempo al aplicar el esquema de control propuesto. Una batería proporcionará al dron mayor o menor tiempo de vuelo según su estado de desgaste.



**Figura 4. 50 Corriente de batería en vuelo con y sin control**

**Fuente: Elaborado por el autor**

En la Figura 4.50 se presenta la superposición de las curvas de corriente obtenidas a través de pruebas con el esquema de control y sin el control del dron, permiten verificar no solamente que el consumo de corriente es mayor sin el esquema de control, sino también que la diferencia en el consumo de corriente entre ambos esquemas tampoco es demasiado significativo. En otras palabras, esta gráfica permite evidenciar que, si bien se ha conseguido una optimización de la corriente suministrada por la batería, aún se requieren altos valores de corriente para sostener al dron en vuelo, lo cual explica el hecho de haber ganado 1 minuto en vuelo autónomo.



**Figura 4. 51 Potencia batería en vuelo con y sin control**

**Fuente: Elaborado por el autor**

En la Figura 4.51, puede observarse que las curvas de potencia de ambos esquemas de trabajo tienden a desplegar un mismo comportamiento. Esto puede comprenderse notando que la corriente sin control es mayor, pero el voltaje tiende a un descenso más pronunciado, lo cual se compensa con el comportamiento inverso del esquema de control, donde la corriente es menor pero el voltaje tiende a descender más lentamente. Esta gráfica indica también que la batería entrega toda la potencia que la carga le requiera y tiende a disminuir dicha entrega de potencia progresivamente hasta la descarga completa de la batería. En la Tabla 4.3, se presentan el margen de corriente desarrollada antes y después de la implementación del sistema de control.

**Tabla 4. 3 Parámetros de consumo de corriente del Drone F450-APM 2.8 Ardupilot**

	CORRIENTE CON CONTROL [A]	CORRIENTE SIN CONTROL [A]
<b>MÁXIMO</b>	16.78	17.26
<b>MÍNIMO</b>	1.29	1.58
<b>PROMEDIO</b>	14.83	15.91

**Fuente: Elaborado por el autor**



Los parámetros mostrados en la Tabla 4.4, corresponden a los valores de potencia, requeridos por el dron durante las pruebas de vuelo antes y después de la implementación.

**Tabla 4. 4 Parámetros de consumo de potencia de Drone F450-APM 2.8 Ardupilot**

	POTENCIA CON CONTROL [W]	POTENCIA SIN CONTROL [W]
<b>MAXIMO</b>	179.38	188.38
<b>MINIMO</b>	16.37	20.07
<b>PROMEDIO</b>	156.13	162.48

**Fuente: Elaborado por el autor**

La autonomía o tiempo de vuelo que se obtuvo de las mediciones realizadas antes y después de la implementación se presentan en la Tabla 4.5.

**Tabla 4. 5 Tiempo de vuelo de Drone F450-APM 2.8 Ardupilot**

	VUELO CON CONTROL	VUELO SIN CONTROL
<b>TIEMPO DE VUELO [min]</b>	10.07	9.05

**Fuente: Elaborado por el autor**

Si bien las estimaciones del tiempo de vuelo no fueron exactas, se cumplió la estimación de tiempo de ahorro planteada al finalizar el capítulo anterior, al evidenciarse 1 minuto adicional de vuelo.

#### 4.4.2 Curvas obtenidas con el sistema de medición de datos de consumo

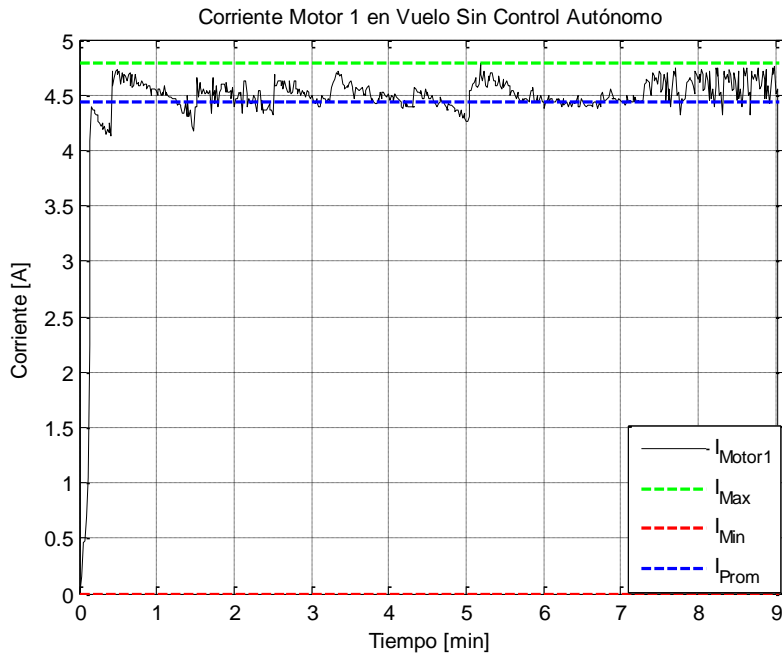


Figura 4. 52 Corriente del motor 1 en vuelo sin control

Fuente: Elaborado por el autor

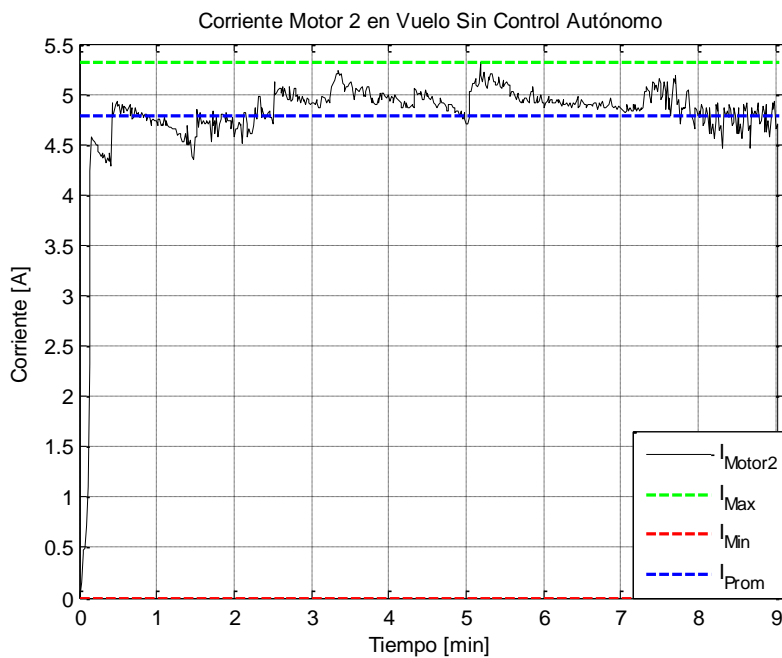
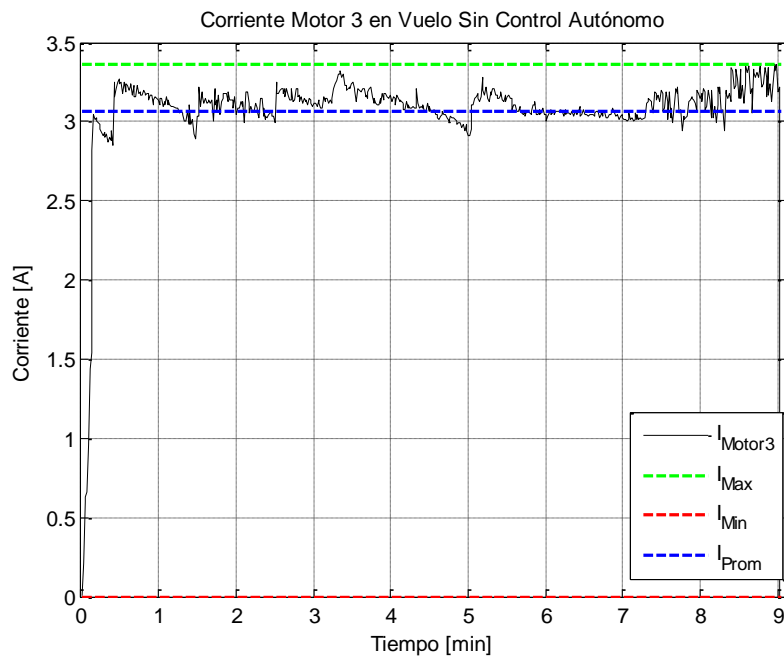


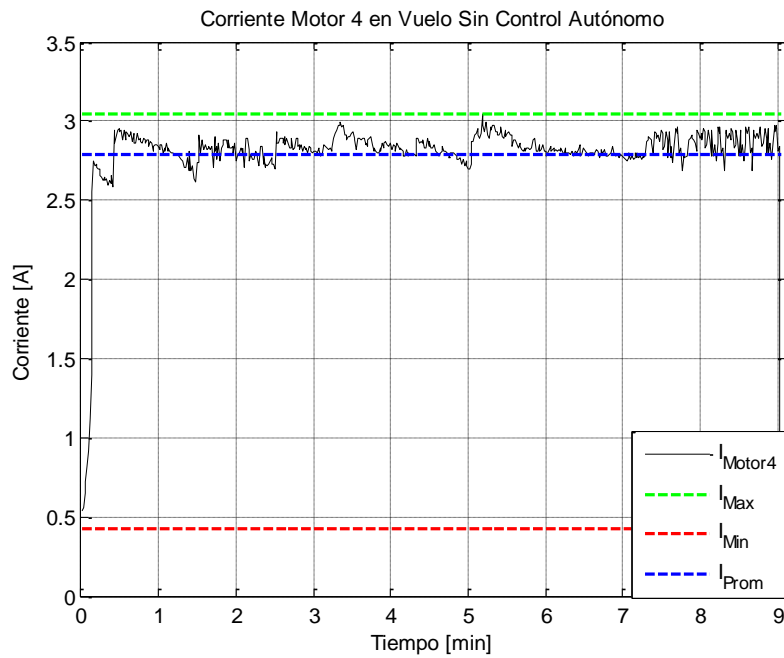
Figura 4. 53 Corriente del motor 2 en vuelo sin control

Fuente: Elaborado por el autor



**Figura 4. 54 Corriente del motor 3 en vuelo sin control**

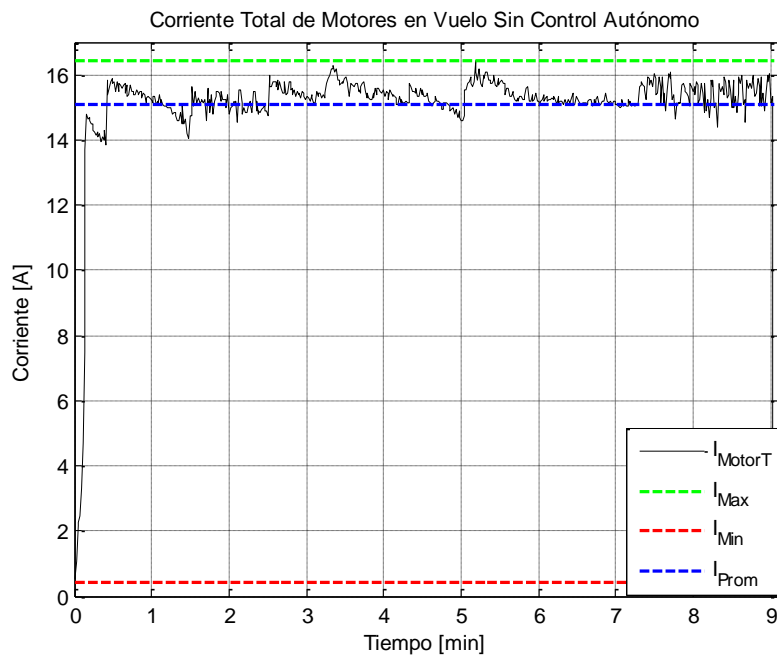
**Fuente: Elaborado por el autor**



**Figura 4. 55 Corriente del motor 4 en vuelo sin control**

**Fuente: Elaborado por el autor**

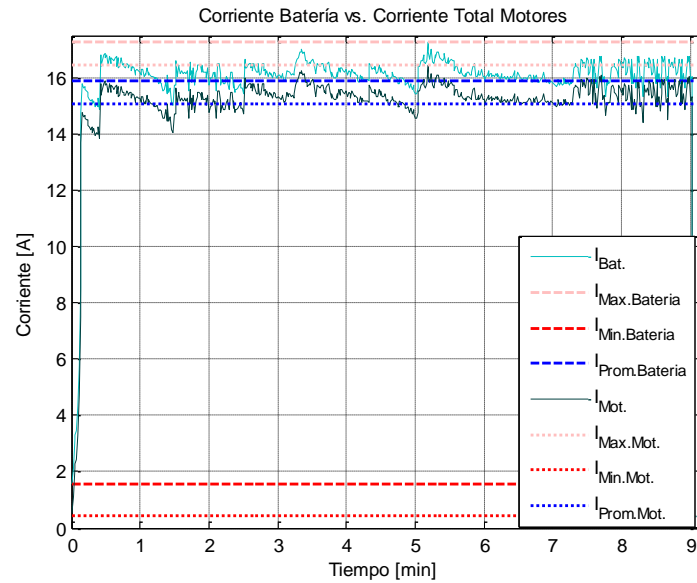
Las gráficas de las Figuras 4.52, 4.53, 4.54 y 4.55, presentadas en anteriormente detallan el comportamiento de la corriente que circula a través de cada uno de los rotores del dron. Puede notarse que si bien las magnitudes de la corriente consumida por cada motor son diferentes, dichas corrientes tienden a seguir un mismo comportamiento. Así mismo, tal y como se evidenció en gráficas anteriores de corriente, se puede notar que esta variable tiende a permanecer constante.



**Figura 4. 56 Corriente total de rotores en vuelo sin control**

**Fuente: Elaborado por el autor**

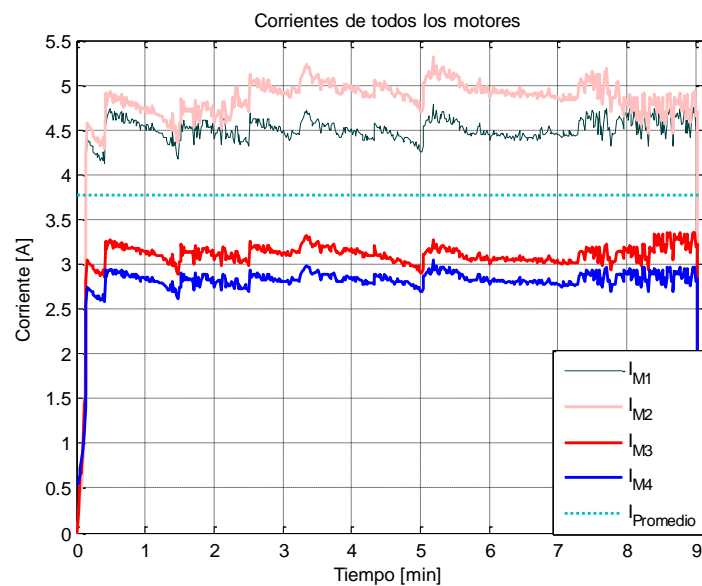
La gráfica de la Figura 4.56, es la totalidad de la corriente consumida por los rotores. Esta corriente no es igual a la corriente total de la fuente puesto que no se contabiliza la corriente suministrada hacia cada uno de los módulos que conforman el sistema, puesto que se realiza la medición de consumo en cada rotor del dron.



**Figura 4. 57 Corriente de batería vs. Corriente total de rotores**

**Fuente: Elaborado por el autor**

La gráfica mostrada en la Figura 4.57, es fundamental para demostrar que las estimaciones presentadas inicialmente fueron correctas. Puede notarse que la corriente consumida por los rotores se encuentra muy cercana a la corriente total consumida por todo el sistema, lo cual fue predicho correctamente. Esto ha implicado que la corriente que puede ahorrarse no resulte ser significativamente mayor, lo cual se traduce en una pequeña ganancia de tiempo de autonomía.



**Figura 4. 58 Corriente de todos rotores**

**Fuente: Elaborado por el autor**

Tal como se ha descrito anteriormente, con la gráfica de la Figura 4.58, puede demostrarse que el comportamiento de las corrientes de cada uno de los rotores sigue una misma tendencia, lo cual es lógico en el marco de vuelo estacionario y más aun considerando un ambiente prácticamente libre de perturbaciones externas. Puede notarse que las corrientes consumidas varían en magnitud según la Tabla 4.6, puesto que cada motor independientemente del control aumenta o disminuye sus revoluciones para alcanzar la estabilización y un vuelo estacionario, adicional puede tomarse en cuenta que el estado de vida útil de cada motor resulta ser diferente, siendo posible que se requieran mayores o menores corrientes para alcanzar determinados valores de revoluciones por minuto [rpm].

De los datos obtenidos durante el desarrollo del proyecto se puede determinar con exactitud la optimización y ahorro de energía conseguido al aplicar nuestra propuesta a un dron cuadricóptero F450-AMP 2.8 de Ardupilot de uso aficionado, como se lo puede evidenciar en el siguiente análisis con datos reales resultado de las mediciones obtenidas.

**Tabla 4. 6 Corriente consumida en promedio por cada rotor del dron**

	<b>CORRIENTE ROTOR 1 [A]</b>	<b>CORRIENTE M2 ROTOR 2 [A]</b>	<b>CORRIENTE M3 ROTOR 3 [A]</b>	<b>CORRIENTE M4 ROTOR 4 [A]</b>
<b>MÁXIMO</b>	4.79	5.31	3.36	3.04
<b>MÍNIMO</b>	0.00	0.00	0.00	0.42
<b>PROMEDIO</b>	4.43	4.7	3.06	2.79

**Fuente: Elaborado por el autor**

En la Tabla 4.7, se puede observar la cantidad corriente que se entrega al sistema propulsor (rotores) por parte de la batería.

**Tabla 4. 7 Corriente consumida promedio de la batería**

	<b>CORRIENTE FUENTE [A]</b>	<b>CORRIENTE TOTAL ROTORES [A]</b>
<b>MÁXIMO</b>	17.26	16.43
<b>MÍNIMO</b>	1.58	0.42
<b>PROMEDIO</b>	15.91	15.07

**Fuente: Elaborado por el autor**

### 4.4.3 Análisis final del sistema implementado

A continuación se presentan los consumos de corriente reales por módulo. Los datos presentados en la Tabla 4.8, se obtuvieron de mediciones realizadas en laboratorio.

Los valores de consumo son considerando con la batería totalmente cargada:

Capacidad de la Batería: 2200 mAh

**Tabla 4. 8 Consumo de corriente real por módulo**

<b>Módulo</b>	<b>Consumo (A) promedio</b>
Propulsión (4 rotores 3,77 A C/U)	15,08 A
Controladora de vuelo APM 2.8	0,2 A
Radio Frecuencia (RF)	0,06 A
Sistema GPS	0,04 A
Módulo de Iluminación	1,03 A
Módulo de Cámara FPV	0,08 A
Módulo de Telemetría	0,18 A
<b>TOTAL</b>	<b>16,67 A</b>

**Fuente: Elaborado por el autor**

En la Tabla 4.9, se presenta el análisis real de la optimización de energía que se logró con el sistema de control propuesto.

**Tabla 4. 9 Análisis final del sistema**

<b>CANTIDAD-DESCRIPCIÓN</b>	<b>CORRIENTE [%]</b>
7 módulos (funcionando)	16,67 A [100 %]
Módulo de propulsión (4 rotores)	15,08 A [90,46 %]
2 módulos (imprescindibles)	15,08 A + 0,2 A + 0,04 = 15,32 A [91,90 %]
<b>Corriente final ahorrada y optimizada</b>	<b>1,35 A [8,10 %]</b>

**Fuente: Elaborado por el autor**

## CONCLUSIONES

- La investigación actualmente se enfoca en desarrollar otras fuentes de alternativas para poder dotar de mayores tiempos de autonomía, como la energía solar, energía química (hidrógeno), emisión de rayos láser desde una estación terrena, mismos que aún se mantienen a nivel investigativo.
- En la actualidad en el caso de los drones multirotores del tipo mini, de características similares del drone F450-AMP2.8 de Ardupilot de uso aficionado, la fuente de energía más común es la batería de polímero de litio, su desventaja radica en que entre más capacidad se requiera más peso representa al sistema, lo que afecta a la autonomía, por lo que este tipo de drones son usados en misiones corto alcance.
- Se ha determinado a través de las mediciones realizadas que dentro de un drone Cuadricóptero del tipo F450-APM 2.8 (Ardupilot) durante vuelo estacionario de uso aficionado, el sistema direcciona más del 90 % al sistema propulsor (rotores), por lo que se evidencia que la investigación busca optimizar energía en alrededor del restante 10%, es por eso que los módulos componentes mantienen un bajo consumo de la batería.
- La corriente total que se ha optimizado o ahorrado es de 1.35 A, resultado del último análisis, que equivalente al 8.10%, que en términos de vuelo de drone representa aproximadamente +/- 1 minuto, que se puede direccionar a salvaguardar el equipo con un descenso controlado, o asignar a un tiempo adicional de vuelo. Es innegablemente el tiempo de vuelo “ganado” todavía no es suficiente para ser considerado significativo, lo cual afirma la importancia de la problemática de extender el tiempo de vuelo de los drones de uso aficionado, atacando otros puntos del sistema tales como la batería, estructura o la búsqueda de nuevos métodos de energización.
- El presente sistema de control y optimización es compatible principalmente con drones cuadricópteros de tipo modular de uso aficionado, puede ser aplicable a otros dispositivos de exploración terrena, acuática, juguetes de estructura modular, después de una revisión de la estructura del vehículo no tripulado.



## RECOMENDACIONES

- En el presente proyecto se desarrolló también un sistema de medición de consumos de voltaje, corriente y potencia online en el drone cuadricóptero F450-APM 2.8 de Ardupilot para su aplicación en laboratorio a través de comunicación serial con el software LabView, partiendo de los resultados obtenidos sería importante continuar con la investigación en temas relacionados como:
  - Un sistema de monitoreo de datos de medición offline que registre los consumos de los parámetros antes mencionados durante el vuelo en exteriores.
  - Implementar un sistema de energía alterna que pudiera coexistir con el actual sistema de alimentación (baterías) que normalmente utilizan los UAV's.
- El sistema de control y optimización de energía desarrollado en el presente proyecto para los casos de drones de sistemas embebidos y de código propietario de uso aficionado, es poco recomendable su aplicación debe someterse a revisión de personal capacitado en el área de UAV's.
- Para las pruebas en laboratorio y obtener mediciones, el drone en estudio debe estar conectado al software LabView a través de comunicación serial cableada y sostenido para simular un vuelo estacionario seguro.
- Al desarrollar proyectos que demanden de sistemas de baterías de polímero de litio, con altas capacidades de amperaje se recomienda utilizar conectores con sistemas extintores de chispa, puesto que un sobre pico de corriente puede averiar algún dispositivo.
- Es importante antes de iniciar una prueba de vuelo, disponer de las baterías totalmente cargadas, dentro del rango de su funcionamiento descrito en el presente trabajo, de igual forma es recomendable no consumir la batería de polímeros de litio LIPO por debajo de su límite (9.4 Vdc) a fin de no degradar su batería y afectar su tiempo de vida útil

## REFERENCIAS BIBLIOGRÁFICAS

- "1910s & 1920s - UAV universe". (13 de 01 de 2019). Obtenido de Sites.google.com:  
<https://sites.google.com/site/uavuni/1910-s>
- "Misil de crucero". (2017). Recuperado el 12 de 01 de 2020, de Es.wikipedia.org:  
[https://es.wikipedia.org/wiki/Misil\\_de\\_crucero](https://es.wikipedia.org/wiki/Misil_de_crucero).
- Aeromodelismofacil. (12 de 12 de 2020). *Aeromodelismofacil*. Obtenido de Aeromodelismofacil:  
[http://www.aeromodelismofacil.com/cuadricoptero\\_plegable\\_NAZA.htm](http://www.aeromodelismofacil.com/cuadricoptero_plegable_NAZA.htm)
- Allegro, M. (21 de 02 de 2020). *Allegro MicroSystems*. Obtenido de Allegro MicroSystems: [www.allegromicro.com](http://www.allegromicro.com)
- Arduino, I. (21 de 02 de 2020). *Arduino*. Obtenido de Arduino:  
<https://store.arduino.cc/usa/arduino-nano>
- ArduPilot, D. T. (17 de 02 de 2019). *ArduPilot Versatile, Trusted, open*. Obtenido de ArduPilot Versatile, Trusted, open: <https://ardupilot.org/copter/docs/common-sik-telemetry-radio.html>
- Barrientos, A., Del Cerro, J., San Martín, R., & Rossi, C. (2012). "Vehículos aéreos no tripulados para uso civil". *Proyecto de Titulación*. Madrid, España: Universidad Politécnica de Madrid.
- Bonelli, M. (10 de 12 de 2019). "*Historia de los drones*". Obtenido de Mundrone.blogspot.com: <https://mundrone.blogspot.com/p/historia/-de-los-drones.html>.
- Cardenas, A., & Olarte, V. (06 de 03 de 2020). *Dronesvoac*. Obtenido de Dronesvoac:  
<https://dronesvoac.blogspot.com/2019/05/drones-integrantes-alexandra-cardenas.html>
- Carrillo, A. J. (2011). *Sistemas automáticos de control*. Santa Rita, Edo. Zulia Venezuela: UMERMB.
- Carvajal Hoyos, C., & Susa Velandia, J. (01 de 03 de 2018). Módulo Energético y de control de temperatura de la Sonda Sabio Caldas II. *Módulo Energético y de control de temperatura de la Sonda Sabio Caldas II*. Bogotá, Colombia: D.R.A.
- Chamorro, W., & Medina, J. (01 de Abril de 2013). Ensamblaje y control de un cuadrcóptero. *Proyecto previo a la obtención de título de Ingeniero en Electrónica y Control*. Quito, Pichincha, Ecuador: Escuela Politécnica Nacional EPN.

- Chimarro, J., & Pavón, E. (01 de 12 de 2017). Integración de un sistema de control para el seguimiento de rutas de vuelo mediante GPS en un hexacóptero de monitoreo. *Trabajo de titulación previo a la obtención del Título de Ingeniero en Electrónica y Control*. Quito, Pichincha, Ecuador: Escuela Politécnica Nacional DRA.
- Drones.mx. (05 de 03 de 2020). *DRONES MX*. Obtenido de DRONES MX: <https://drones.mx/blog/modos-vuelo-drones/>
- Dyna. (junio de 2016). *Magazine*. Obtenido de Evolución histórica de los vehículos aéreos no tripulados hasta la actualidad: <http://dx.doi.org/10.6036/7781>
- EACHINE. (17 de 02 de 2019). *eachine*. Obtenido de eachine: [https://www.eachine.com/Eachine-1000TVL-1-or-3-CCD-110-Degree-2\\_8mm-Lens-Mini-FPV-Camera-NTSC-PAL-Switchable-p-357.html](https://www.eachine.com/Eachine-1000TVL-1-or-3-CCD-110-Degree-2_8mm-Lens-Mini-FPV-Camera-NTSC-PAL-Switchable-p-357.html)
- EduRed. (13 de 01 de 2020). *Ecured.cu*. Obtenido de General Atomics MQ-1 Predator: [https://www.ecured.cu/General\\_Atomics\\_MQ-1\\_Predator](https://www.ecured.cu/General_Atomics_MQ-1_Predator)
- Escalona, M., & Morillo, J. (2017). *TEORÍA CLÁSICA DE CONTROL AUTOMÁTICA Y APLICACIONES EN INGENIERÍA* (Vol. TOMO IV). QUITO, PICHINCHA, ECUADOR: JURÍDICA DEL ECUADOR.
- Fairchild. (4 de 2015). *ON Semiconductor*. Obtenido de I+D Electrónica: <https://www.onsemi.com/pub/Collateral/4N37M-D.pdf>
- FlySky. (21 de 07 de 2014). *FCCID*. Obtenido de FCCID: <https://fccid.io/N4ZFLYSKYIA6/User-Manual/User-Manual-2330552>
- G, Z., Li, C., & Chen, P. (2005). Unmanned aerial vehicle (UAV) real-time video registration for forest fire monitoring. *Geoscience and Remote Sensing Symposium, 2005*, 1803-1806.
- González, D. (18 de Abril de 2016). *"DRON,DRONE, UAV, UAS, RPA, RPAS.¿QUE SON? ¿CÓMO LLAMARLOS CORRECTAMENTE?"* Obtenido de Masscience: <https://masscience.com/2015/08/11/2051/>
- Hinojosa, M., & Maestre, J. (2018). Trabajo de fin de carrera Grado en Ingeniería en Tecnologías Industriales. *Diseño y construcción de un drone autónomo para medición de radiación solar*. Sevilla, España: DRA Universidad de Sevilla.
- Juna, E. (2015). Tesis previa a la obtención de título de Ingeniero Eléctrico. En E. J. Juca, *Eficiencia energética mediante sistemas SCADA para el control de la demanda de una residencia* (págs. 7,9). Quito: DRA, Universidad Politécnica Salesiana.
- Lastra, J. (2017). Diseño de un drone programable de bajo coste. *Proyecto de fin de Master*. Cantabria: Universidad de Cantabria.

- Magnusson, T. (12 de 6 de 2014). Attitude Control of a Hexarotor. *Proyecto de grado*. Linköping, Suecia: Universidad de Linköping.
- MathWorks. (04 de 03 de 2020). *Mathworks.com*. Obtenido de Mathworks.com: <https://la.mathworks.com/solutions/data-science.html>
- Mayorga, R. (2009). *Sistema de navegación para vehículos aéreos cuadricópteros*. Cataluña: Universidad Politécnica Superior de Cataluña.
- Montero, J. (31 de 12 de 2017). *Que diferencias hay entre RPA--UAV-RPAS-AUS-dron*. Obtenido de Todrone.com: <http://www.todrone.com/diferencias-hay-entre-rpa-uav-rpas-uas-dron/>.
- Mundrone. (06 de 03 de 2020). *Mundo Drone*. Obtenido de Mundo Drone: <http://mundrone.blogspot.com/p/historia-de-los-drones.html>
- Nadales, C. (2009). "Control de un Quadrotor mediante la plataforma arduino". *Proyecto de Titulación de Ingeniería Técnica en Telecomunicaciones*. Cataluña, España: Universidad Politécnica Superior de Cataluña.
- National Instrument, C. (21 de 02 de 2020). *National Instruments TM*. Obtenido de National Instruments TM: <https://www.ni.com/es-cr/shop/labview.html>
- NewAtlas. (31 de 01 de 2020). *NewAtlas.com*. Obtenido de NewAtlas.com: <https://newatlas.com/grasp-nano-quadrotor-robots-swarm/21302/>
- NexWareRF. (07 de 07 de 2014). *ImmersionRC Real Virtuality*. Obtenido de ImmersionRC Real Virtuality: <https://docs.gestionaweb.cat/0644/5-8ghz-600mw-tx-manual-es.pdf>
- Niermeyer, P., & Raffler, T. (01 de 2015). *researchgate.net*. Obtenido de researchgate.net: [https://www.researchgate.net/figure/Ascending-Technologies-Hummingbird-extended-by-a-custom-ight-control-computer\\_fig3\\_275887697](https://www.researchgate.net/figure/Ascending-Technologies-Hummingbird-extended-by-a-custom-ight-control-computer_fig3_275887697)
- NovoDrone. (06 de 03 de 2020). *Novodrone*. Obtenido de Novodrone: <https://novodrone.com/tipos-de-drones/?v=31f50d618f89>
- Ortega, D. (2015). *Piloto de Drones RPAS*. Madrid-España: S.A Ediciones Paraninfo.
- Parsch, A. (06 de 03 de 2020). *Designation-systems.net*. Obtenido de Designation-systems.net: <http://www.designation-systems.net/dusrm/app2/q-1.html>
- Prem, K., Anand, S., & Ganesh, E. (01 de 06 de 2016). *researchgate.net*. Obtenido de researchgate.net: [https://www.riverpublishers.com/journal\\_read\\_html\\_article.php?j=JGE/5/2/2](https://www.riverpublishers.com/journal_read_html_article.php?j=JGE/5/2/2)
- Quadcopter Arena. (2018). *quadcopterarena.com*. Obtenido de <https://quadcopterarena.com/parrot-ar-drone-2-0-power-edition-review/>

- Saskatoon, S. (28 de 02 de 2020). *CISION PR Web*. Obtenido de prweb.com:  
<https://www.prweb.com/releases/2015/04/prweb12644560.htm>
- SavantIC, E. (20 de 02 de 2020). *SVNTC*. Obtenido de SVNTC: <http://www.svntc.com/>
- Tecingenieriaclasico. (16 de 03 de 2020). *tecingenieriaclasico.blogspot*. Obtenido de tecingenieriaclasico.blogspot: <http://tecingenieriaclasico.blogspot.com/p/1.html>
- Van Blyenburgh, P. (2014). *Aperiron UAV*. Obtenido de proyectoapeiron.wordpress.com:  
<https://proyectoapeiron.wordpress.com/2014/02/11/clasificacion-uav-segun-la-otan/>
- Velásquez, G. (2012). Vehículos aéreos no tripulados UAV. Santiago: Universidad Santiago de Chile.
- Waveshare. (01 de 07 de 2014). *Terraelectronica*. Obtenido de Terraelectronica:  
[https://www.terraelectronica.ru/pdf/show?pdf\\_file=%2Fz%2Fdatasheet%2FU%2FUART+GPS+NEO-6M+User+Manual.pdf](https://www.terraelectronica.ru/pdf/show?pdf_file=%2Fz%2Fdatasheet%2FU%2FUART+GPS+NEO-6M+User+Manual.pdf)
- Weihua, Z., & Tiau, H. (26 de 05 de 2012). *ScienceDirect*. Obtenido de ScienceDirect:  
<https://www.sciencedirect.com/science/article/abs/pii/S001600321300389X?via%3Dihub>
- Wikiwand. (14 de 02 de 2020). *Wikiwand.com*. Obtenido de Wikiwand.com:  
[https://www.wikiwand.com/es/Bell-Boeing\\_V-22\\_Osprey](https://www.wikiwand.com/es/Bell-Boeing_V-22_Osprey)

## **ANEXOS**

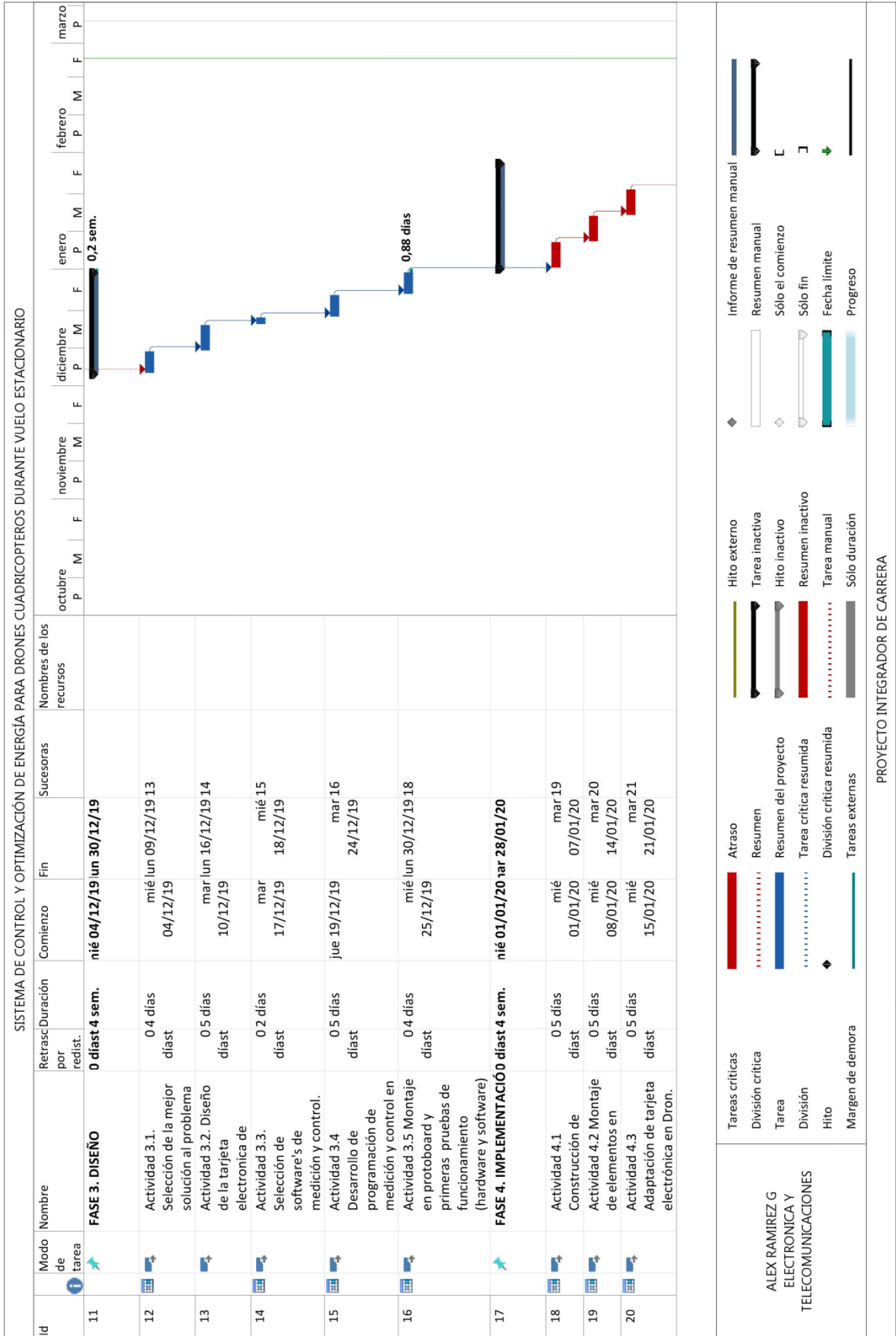
**ANEXO No. 1**  
**CRONOGRAMA DEL PROYECTO (PROJECT)**

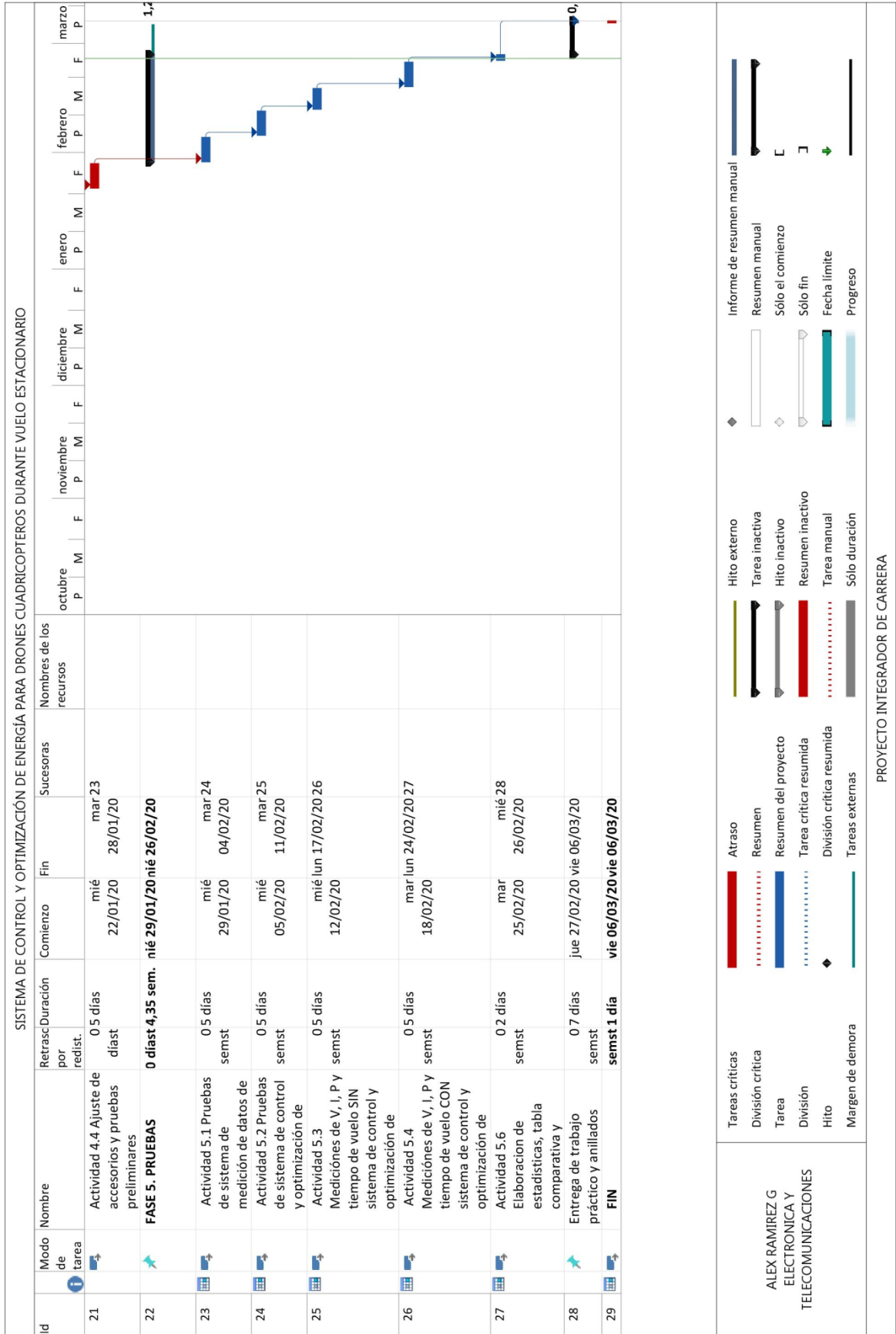
SISTEMA DE CONTROL Y OPTIMIZACIÓN DE ENERGÍA PARA DRONES CUADRICOPTEROS DURANTE VUELO ESTACIONARIO													
Id	Modo de tarea	Nombre	Retraso/Duración por redist.	Comienzo	Fin	Sucesoras	Nombres de los recursos	octubre	noviembre	diciembre	enero	febrero	marzo
								P	M	F	P	M	F
1		Sistema de control y optimización de energía para drones	0 21,98 días sem.	mié 09/10/19	vie 06/03/20								
2		INICIO	0 días	mié 09/10/19	mié 09/10/19								104,88
3		FASE 1 .INVESTIGACIÓN	0 días 4 sem.	mié 09/10/19	mar 05/11/19								
4		Actividad 1.1 Recabar información de antecedentes históricos y actuales de Drones	0 10 días	mié 09/10/19	mar 22/10/19								
5		Actividad 1.2. Investigación sobre diferentes tipos de sistemas de energía para drones	0 10 días	mié 23/10/19	mar 05/11/19								
6		FASE 2 .METODOLOGÍA	0 días 4 sem.	mié 06/11/19	mar 03/12/19								
7		Actividad 2.1 .Inicio desarrollo marco	0 5 días	mié 06/11/19	mar 12/11/19								
8		Actividad 2.2 Elección de Métodos y técnicas de análisis de datos	0 5 días	mié 13/11/19	mar 19/11/19								
9		Actividad 2.3 Elección del equipo (drone) para desarrollo de	0 5 días	mié 20/11/19	mar 26/11/19								
10		Actividad 2.4 . Metodología de trabajo y Técnicas de recolección de datos	0 5 días	mié 27/11/19	mar 03/12/19								

PROYECTO INTEGRADOR DE CARRERA	
ALEX RAMIREZ G ELECTRONICA Y TELECOMUNICACIONES	<p><b>Tareas críticas</b> (Red solid bar)</p> <p><b>División crítica</b> (Red dotted bar)</p> <p><b>Tarea</b> (Blue solid bar)</p> <p><b>División</b> (Blue dotted bar)</p> <p><b>Hito</b> (Black diamond)</p> <p><b>Margen de demora</b> (Green solid bar)</p> <p><b>Atraso</b> (Red solid bar)</p> <p><b>Resumen</b> (Red dotted bar)</p> <p><b>Resumen del proyecto</b> (Blue solid bar)</p> <p><b>Tarea crítica resumida</b> (Blue dotted bar)</p> <p><b>División crítica resumida</b> (Black diamond)</p> <p><b>Tareas externas</b> (Green solid bar)</p> <p><b>Hito externo</b> (Black diamond)</p> <p><b>Tarea inactiva</b> (Grey solid bar)</p> <p><b>Hito inactivo</b> (Grey dotted bar)</p> <p><b>Resumen inactivo</b> (Red solid bar)</p> <p><b>Tarea manual</b> (Red dotted bar)</p> <p><b>Sólo duración</b> (Grey solid bar)</p> <p><b>Informe de resumen manual</b> (Black diamond)</p> <p><b>Resumen manual</b> (Black dotted bar)</p> <p><b>Sólo el comienzo</b> (Black solid bar)</p> <p><b>Sólo fin</b> (Black dotted bar)</p> <p><b>Fecha límite</b> (Green solid bar)</p> <p><b>Progreso</b> (Blue solid bar)</p>

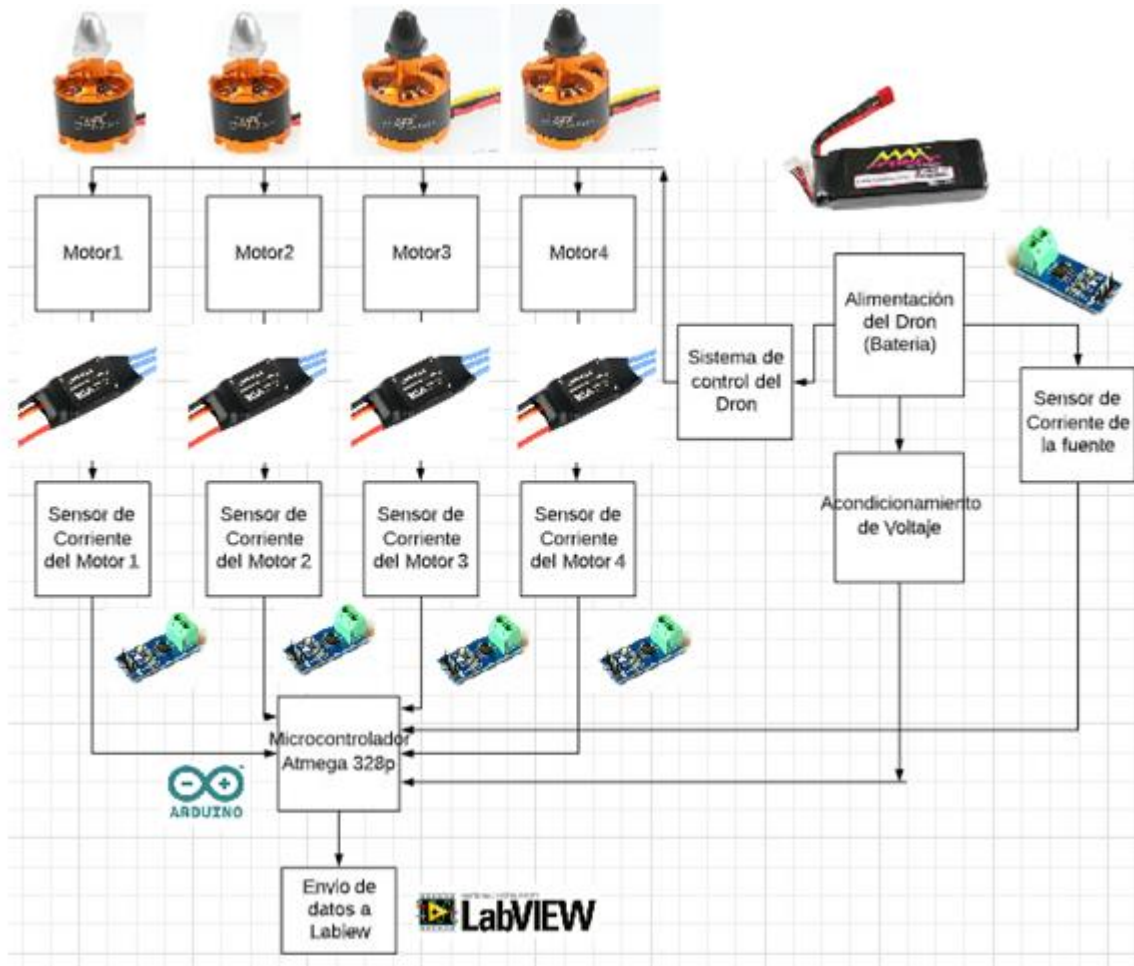




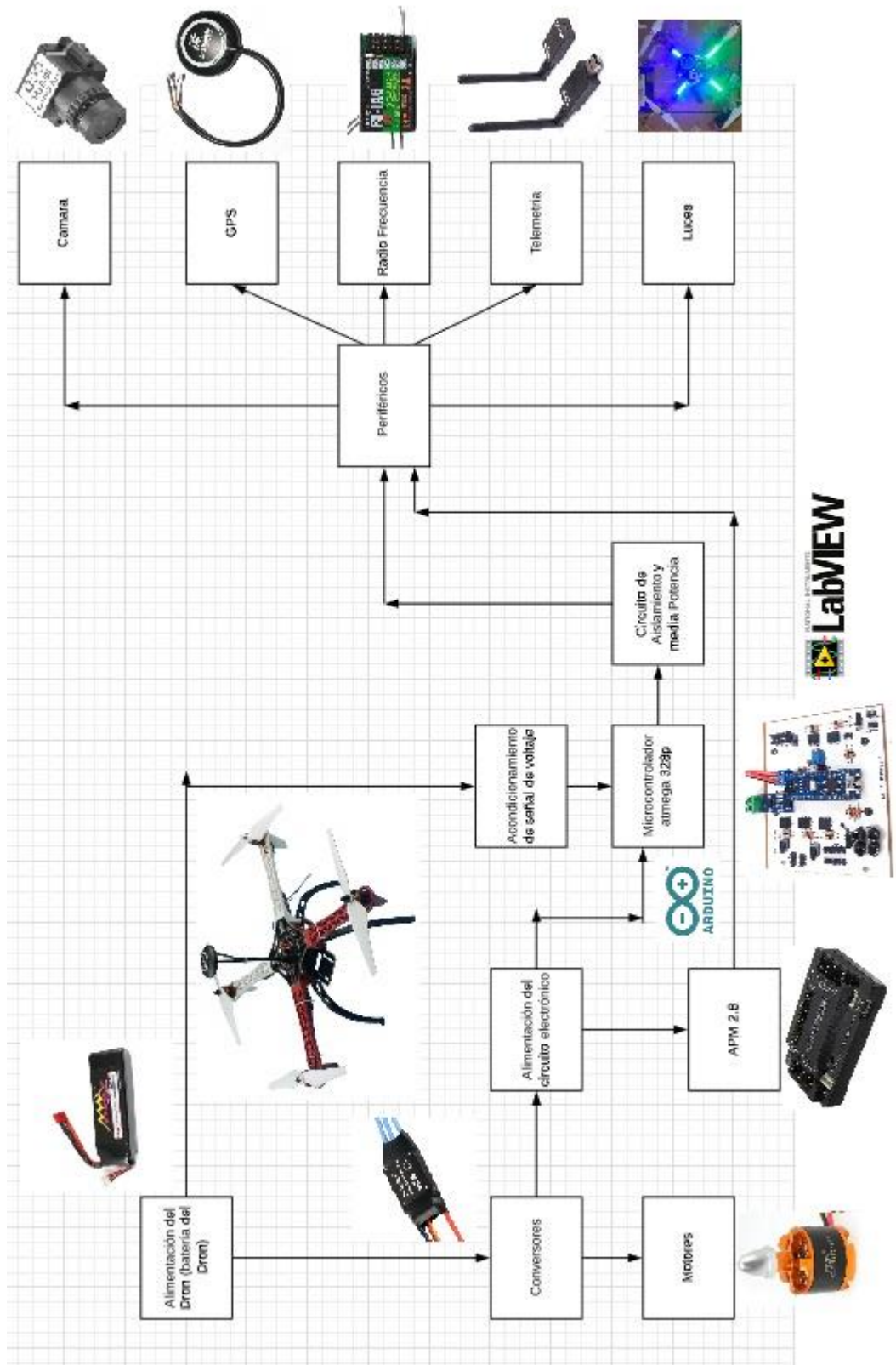


## ANEXO No. 2

### ARQUITECTURA DEL SISTEMA DE MEDICIÓN



**ANEXO No. 3**  
**ARQUITECTURA DEL SISTEMA DE CONTROL Y OPTIMIZACIÓN**



## ANEXO No. 4

### PROGRAMACIÓN DEL SISTEMA DE MEDICIÓN (ARDUINO IDE)

```
// UNIVERSIDAD TECNOLÓGICA ISRAEL
//CARRERA DE ELECTRÓNICA DIGITAL Y TELECOMUNICACIONES
//PROYECTO DE TITULACIÓN PREVIO A LA OBTENCIÓN DE TÍTULO,
//INGENIERO EN ELECTRÓNICA DIGITAL Y TELECOMUNICACIONES
//AUTOR: ALEX RODRIGO RAMIREZ GUAÑA

int Corriente1;
int Corriente2;
int Corriente3;
int Corriente4;
int Corriente5;
int CorrienteFuente;
int voltajeBateria;
int voltajeMotor;
int VoltajeSensorBateria;

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

void loop() {

  float Corriente1_c=get_voltaje(2000,1);
  Corriente1=100*(6.676)*(Corriente1_c-2.52);
  float Corriente2_c=get_voltaje(2000,2);
  Corriente2=100*(5.9)*(Corriente2_c-2.51);
  float Corriente3_c=get_voltaje(2000,3);
  Corriente3=100*(5.871)*(Corriente3_c-2.49);
  float Corriente4_c=get_voltaje(2000,4);
  Corriente4=100*(5.6*(Corriente4_c-0.02)-13.76);
  float Voltaje_c=get_voltaje(2000,5);
  voltajeBateria=100*(3.06*Voltaje_c);
  float Corriente5_c=get_voltaje(2000,6);
  CorrienteFuente=100*(15.945*(Corriente5_c+0.02)-39.801);

  // ENVIO
  String data = "a";
  data += Corriente1;
  data += "b";
  data += Corriente2;
  data += "c";
  data += Corriente3;
  data += "d";
  data += Corriente4;
  data += "e";
  data += CorrienteFuente;
  data += "f";
  data += voltajeBateria;
  Serial.print(data);
  delay(500);
}
```

```
float get_voltaje(int n_muestras, int motor) //motor 1,2,3,4 y 5 para
la batería
{
    float voltaje=0;

    for(int i=0;i<n_muestras;i++)
    {
        if (motor==1){
            voltaje = voltaje+analogRead(A3) * (5.0 / 1023.0); }
        if (motor==2){
            voltaje = voltaje+analogRead(A0) * (5.0 / 1023.0); }
        if (motor==3){
            voltaje = voltaje+analogRead(A1) * (5.0 / 1023.0); }
        if (motor==4){
            voltaje = voltaje+analogRead(A2) * (5.0 / 1023.0); }
        if (motor==5){ // batería voltaje
            voltaje = voltaje+analogRead(A4) * (5.0 / 1180.0); }
        if (motor==6){ // corriente total
            voltaje = voltaje+analogRead(A5) * (5.0 / 1023.0); }

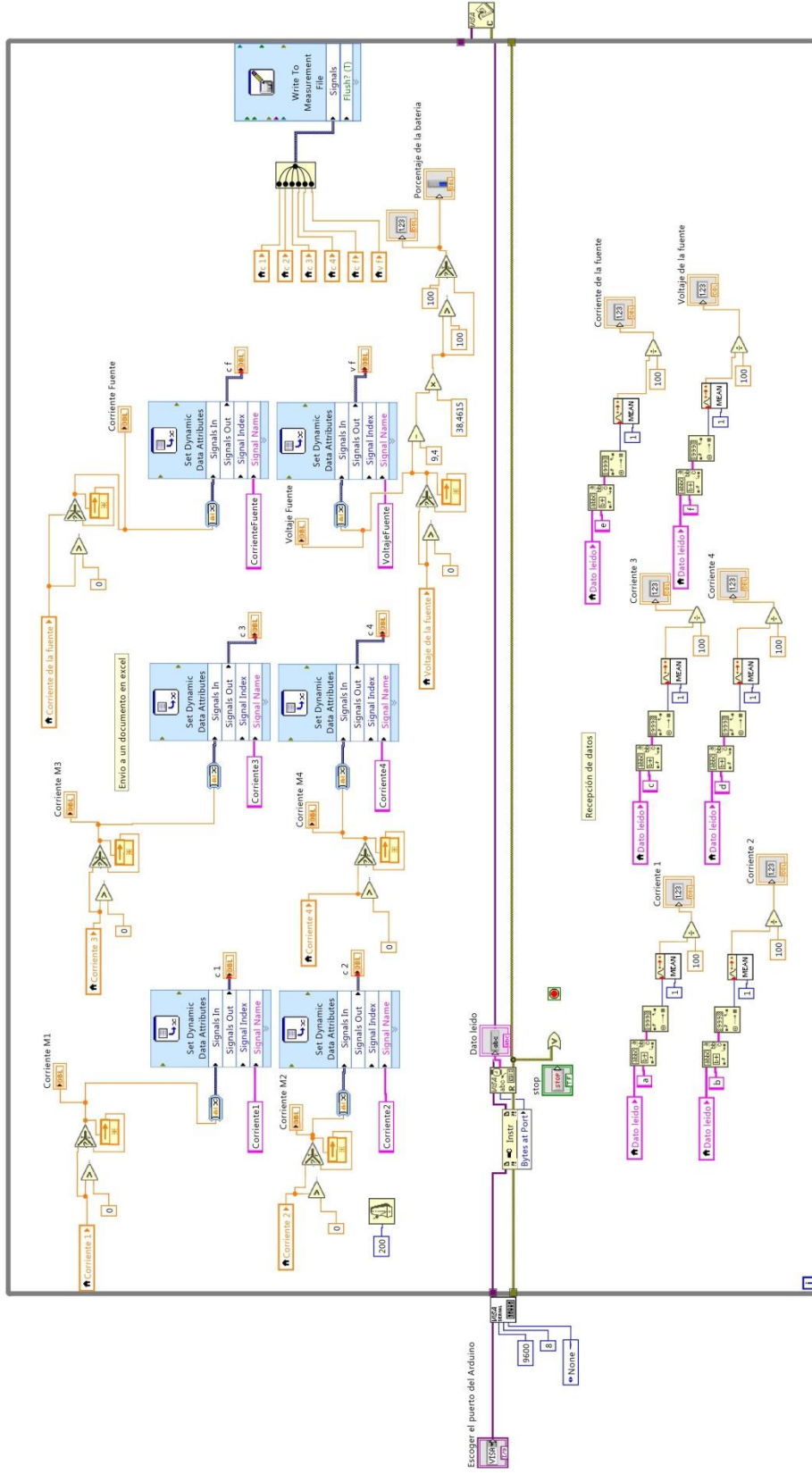
    }
    voltaje=voltaje/n_muestras;
    return(voltaje);
}
```

**ANEXO No. 5**  
**PROGRAMACIÓN DEL SISTEMA DE MEDICIÓN (LABVIEW)**





Placa de medicion.vi  
G:\Archivos dron\Placa de medicion.vi  
Last modified on 29/02/2020 at 14:48  
Printed on 18/03/2020 at 14:23



## ANEXO No. 6

### PROGRAMACIÓN DEL SISTEMA DE CONTROL Y OPTIMIZACIÓN (ARDUINO IDE)

```
// UNIVERSIDAD TECNOLOGICA ISRAEL
// CARRERA DE ELECTRONICA DIGITAL Y TELECOMUNICACIONES
// PROYECTO DE TITULACIÓN PREVIO A LA OBTENCIÓN DE TITULO DE INGENIERO
// AUTOR: ALEX RODRIGO RAMIREZ GUAÑA

#define RF 12
#define GPS 11
#define luces 10
#define camara 9
#define telemetry 8
#define V_Max 12.60
#define DEBUG(a)

char ingreso;
int bandera=0;
char stand=0;

float capacidad=0;
float corrientemA=0;
float capacidadConsumida;

int CorrienteFuente;
int voltajeBateria;
int voltajeMotor;
int VoltajeSensorBateria;
int VoltajePorcentaje;
float voltaje_Bateria;
unsigned long tiempo1 = 0;
unsigned long tiempo2 = 0;

void setup() {

  Serial.begin(9600);
  Serial.setTimeout(50);

  //Configuración de pines
  pinMode(telemetry,OUTPUT);
  pinMode(RF,OUTPUT);
  pinMode(camara,OUTPUT);
  pinMode(luces,OUTPUT);
  pinMode(GPS,OUTPUT);
  //Enciende todo
  digitalWrite(telemetry,HIGH);
  digitalWrite(RF,HIGH);
  digitalWrite(camara,HIGH);
  digitalWrite(luces,HIGH);
  digitalWrite(GPS,HIGH);
  tiempo1 = millis();
}

void loop() {

  float Voltaje_c=get_voltaje(1000,5);
  voltajeBateria = 100*(3*Voltaje_c);
```

```

    voltaje_Bateria = (3*Voltaje_c) ;
    float Corriente5_c=get_voltaje(2000,6)*0.985;
    CorrienteFuente=100*((Corriente5_c-2.468)*10.94*0.97); //-27.25
//*11.08
    corrientemA=1000*((Corriente5_c-2.468)*10.94*0.97); //-27.25 //*11.08

    String data = "e";
    data += CorrienteFuente;

    data += "f";
    data += voltajeBateria;
    Serial.print(data);

    delay(500);
    if(voltaje_Bateria<=11.2){//85%
        digitalWrite(luces,LOW);
    }
    if(voltaje_Bateria<=10.7){// 65%
        digitalWrite(camara,LOW);
    }
    if(voltaje_Bateria<=10.1){//30%
        digitalWrite(telemetry,LOW);
    }
    if(voltaje_Bateria<=9.45){// 5%
        digitalWrite(GPS,LOW);
        digitalWrite(RF,LOW);
    }

}

}

float get_voltaje(int n_muestras, int motor) //motor 1,2,3,4 y 5 para
la batería
{
    float voltaje=0;

    for(int i=0;i<n_muestras;i++)
    {

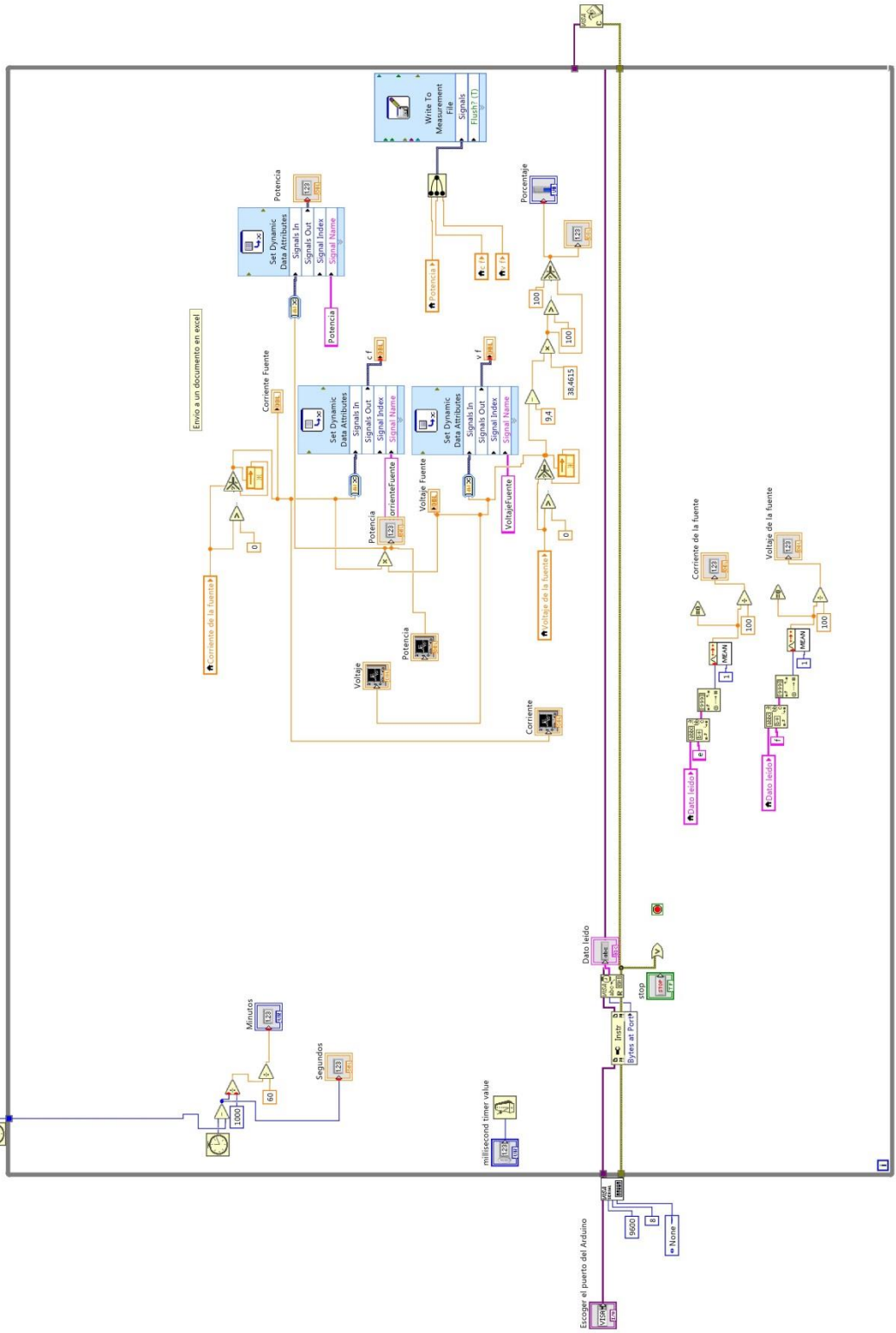
        if (motor==5){ //
bateria voltaje
            voltaje = voltaje+analogRead(A5)*(5 / 1023.0); }
        if (motor==6){ //
corriente total
            voltaje = voltaje+analogRead(A4) * (5.0 / 1023.0); }
    }
    voltaje=voltaje/n_muestras;
    return(voltaje);
}

```

---

**ANEXO No. 7**  
**PROGRAMACIÓN DEL SISTEMA DE CONTROL Y**  
**OPTIMIZACIÓN (LABVIEW)**

Placa de Control.vi  
G:\Archivos diron\Placa de Control.vi  
Last modified on 29/02/2020 at 14:35  
Printed on 18/03/2020 at 14:27



## ANEXO No. 8

### CODIGO MATLAB – GENERACIÓN DE GRÁFICAS

El código MatLab desarrollado para la generación de gráficas.

```
% Script de Procesamiento de Información de Mediciones obtenidas para
DRON

clc

clear all

close all

% Procesamiento de Tabla de Control Autónomo de Vuelo

% Adquisición de Información

A = readtable('DATOS_Control.csv');

T_Control = A.MINUTOS;

V_Control = A.VOLTAJE;

I_Control = A.CORRIENTE;

P_Control = A.POTENCIA;

V_Max_Control = A.V_MAX;

V_Min_Control = A.V_MIN;

V_E1 = A.V_E1;

V_E2 = A.V_E2;

V_E3 = A.V_E3;

[T_E1,V_Eoff01] = polyxpoly(T_Control,V_Control,T_Control,V_E1);
%Definición Tiempos de corte
T_E1 = T_E1(1);
V_Eoff01 = V_Eoff01(1);

[T_E2,V_Eoff02] = polyxpoly(T_Control,V_Control,T_Control,V_E2);
T_E2 = T_E2(1);
V_Eoff02 = V_Eoff02(1);

[T_E3,V_Eoff03] = polyxpoly(T_Control,V_Control,T_Control,V_E3);
T_E3 = T_E3(1);
V_Eoff03 = V_Eoff03(1);

[T_E4,V_Eoff04] = polyxpoly(T_Control,V_Control,T_Control,V_Min_Control);
T_E4 = T_E4(1);
V_Eoff04 = V_Eoff04(1);
```

```

I_Max_Control = A.I_MAX;

I_Min_Control = A.I_MIN;

I_Prom_Control = A.I_PROM;

P_Max_Control = A.P_MAX;

P_Min_Control = A.P_MIN;

P_Prom_Control = A.P_PROM;

% Graficación Voltaje Bateria

figure
plot(T_Control,V_Control,'k','LineWidth',2)
hold on
plot(T_Control,V_Max_Control,'--b','LineWidth',2)
hold on
plot(T_Control,V_E1,'--','Color',[0 0.75 0.75],'LineWidth',2)
hold on
plot(T_Control,V_E2,'--','Color',[0 0.5 0.5],'LineWidth',2)
hold on
plot(T_Control,V_E3,'--','Color',[0 0.25 0.25],'LineWidth',2)
hold on
plot(T_Control,V_Min_Control,'--r','LineWidth',2)
hold on
mapshow(T_E1,V_Eoff01,'DisplayType','point','Marker','o','LineWidth',2)
hold on
mapshow(T_E2,V_Eoff02,'DisplayType','point','Marker','o','LineWidth',2)
hold on
mapshow(T_E3,V_Eoff03,'DisplayType','point','Marker','o','LineWidth',2)
hold on
mapshow(T_E4,V_Eoff04,'DisplayType','point','Marker','o','LineWidth',2)
title('Voltaje Bateria en Vuelo Autónomo Controlado')
xlabel('Tiempo [min]')
ylabel('Voltaje [V]')
grid on
legend('V_B_a_t_e_r_i_a','V_M_a_x','OFF_L_u_c_e_s','OFF_C_a_m_a_r_a','OFF
_T_e_l_e_m','V_M_i_n (OFF_R_F)')
xlim([0 10.1])
ylim([9 12.8])

% Graficación Corriente Bateria

figure
plot(T_Control,I_Control,'k','LineWidth',1)
hold on
plot(T_Control,I_Max_Control,'--g','LineWidth',2)
hold on
plot(T_Control,I_Min_Control,'--r','LineWidth',2)
hold on
plot(T_Control,I_Prom_Control,'--b','LineWidth',2)
title('Corriente Bateria en Vuelo Autónomo Controlado')
xlabel('Tiempo [min]')
ylabel('Corriente [A]')
grid on
legend('I_B_a_t_e_r_i_a','I_M_a_x','I_M_i_n','I_P_r_o_m')

```

```
xlim([0 10.1])
ylim([0 17.5])

% Graficación Potencia Batería

figure
plot(T_Control,P_Control,'k','LineWidth',1)
hold on
plot(T_Control,P_Max_Control,'--g','LineWidth',2)
hold on
plot(T_Control,P_Min_Control,'--r','LineWidth',2)
hold on
plot(T_Control,P_Prom_Control,'--b','LineWidth',2)
title('Potencia Batería en Vuelo Autónomo Controlado')
xlabel('Tiempo [min]')
ylabel('Potencia [W]')
grid on
legend('P_B_a_t_e_r_i_a','P_M_a_x','P_M_i_n','P_P_r_o_m')
xlim([0 10.1])
ylim([0 192])

%% Procesamiento de Tabla de Vuelo Sin Control

% Adquisición de Información

B = readtable('DATOS_SinControl.csv');

T_SinControl = B.MINUTOS;

V_SinControl = B.VOLTAJE;

I_SinControl = B.CORRIENTE;

P_SinControl = B.POTENCIA;

V_Max_SinControl = B.V_MAX;

V_Min_SinControl = B.V_MIN;

I_Max_SinControl = B.I_MAX;

I_Min_SinControl = B.I_MIN;

I_Prom_SinControl = B.I_PROM;

P_Max_SinControl = B.P_MAX;

P_Min_SinControl = B.P_MIN;

P_Prom_SinControl = B.P_PROM;

% Graficación Voltaje Batería

figure
plot(T_SinControl,V_SinControl,'k','LineWidth',2)
hold on
plot(T_SinControl,V_Max_SinControl,'--b','LineWidth',2)
hold on
```



```

plot(T_SinControl,V_Min_SinControl,'--r','LineWidth',2)
title('Voltaje Batería en Vuelo Sin Control Autónomo')
xlabel('Tiempo [min]')
ylabel('Voltaje [V]')
grid on
legend('V_B_a_t_e_r_i_a','V_M_a_x','V_M_i_n')
xlim([0 9.1])
ylim([9 12.8])

% Graficación Corriente Batería

figure
plot(T_SinControl,I_SinControl,'k','LineWidth',1)
hold on
plot(T_SinControl,I_Max_SinControl,'--g','LineWidth',2)
hold on
plot(T_SinControl,I_Min_SinControl,'--r','LineWidth',2)
hold on
plot(T_SinControl,I_Prom_SinControl,'--b','LineWidth',2)
title('Corriente Batería en Vuelo Sin Control Autónomo')
xlabel('Tiempo [min]')
ylabel('Corriente [A]')
grid on
legend('I_B_a_t_e_r_i_a','I_M_a_x','I_M_i_n','I_P_r_o_m')
xlim([0 9.1])
ylim([0 17.5])

% Graficación Potencia Batería

figure
plot(T_SinControl,P_SinControl,'k','LineWidth',1)
hold on
plot(T_SinControl,P_Max_SinControl,'--g','LineWidth',2)
hold on
plot(T_SinControl,P_Min_SinControl,'--r','LineWidth',2)
hold on
plot(T_SinControl,P_Prom_SinControl,'--b','LineWidth',2)
title('Potencia Batería en Vuelo Sin Control Autónomo')
xlabel('Tiempo [min]')
ylabel('Potencia [W]')
grid on
legend('P_B_a_t_e_r_i_a','P_M_a_x','P_M_i_n','P_P_r_o_m')
xlim([0 9.1])
ylim([0 192])

%% Graficación Comparativa entre Vuelo Controlado y Sin Control

% Voltaje

figure
plot(T_SinControl,V_SinControl,'b:', 'LineWidth',2)
hold on
plot(T_Control,V_Control,'b','LineWidth',2)
plot(T_Control,V_Max_Control,'--g','LineWidth',2)
hold on
plot(T_Control,V_Min_Control,'--r','LineWidth',2)
title('Voltaje Batería en Vuelo Con y Sin Control')
xlabel('Tiempo [min]')
ylabel('Voltaje [V]')

```

```

grid on
legend('V_B_a_t_.S_i_n_C_o_n_t_r_o_l','V_B_a_t_.C_o_n_C_o_n_t_r_o_l','V
_M_a_x','V_M_i_n')
xlim([0 10.1])
ylim([9 12.8])

% Corriente

figure
plot(T_SinControl,I_SinControl,'Color',[0 0.75 0.75],'LineWidth',1)
hold on
plot(T_SinControl,I_Max_SinControl,'--','Color',[1 0.75
0.75],'LineWidth',2)
hold on
plot(T_SinControl,I_Min_SinControl,'--r','LineWidth',2)
hold on
plot(T_SinControl,I_Prom_SinControl,'--b','LineWidth',2)
hold on
plot(T_Control,I_Control,'Color',[0 0.25 0.25],'LineWidth',1)
hold on
plot(T_Control,I_Max_Control,':','Color',[1 0.75 0.75],'LineWidth',2)
hold on
plot(T_Control,I_Min_Control,':r','LineWidth',2)
hold on
plot(T_Control,I_Prom_Control,':b','LineWidth',2)
title('Corriente Batería en Vuelo Con y Sin Control')
xlabel('Tiempo [min]')
ylabel('Corriente [A]')
grid on
legend('I_B_a_t_.S_i_n_C_o_n_t_r_o_l','I_M_a_x_.S_i_n_C_o_n_t_r_o_l','I
_M_i_n_.S_i_n_C_o_n_t_r_o_l',...
'I_P_r_o_m_.S_i_n_C_o_n_t_r_o_l',...

'I_B_a_t_.C_o_n_C_o_n_t_r_o_l','I_M_a_x_.C_o_n_C_o_n_t_r_o_l','I_M_i_n
.C_o_n_C_o_n_t_r_o_l',...
'I_P_r_o_m_.C_o_n_C_o_n_t_r_o_l')
xlim([0 10.1])
ylim([0 17.5])

% Potencia

figure
plot(T_SinControl,P_SinControl,'Color',[0 0.75 0.75],'LineWidth',1)
hold on
plot(T_SinControl,P_Max_SinControl,'--','Color',[1 0.75
0.75],'LineWidth',2)
hold on
plot(T_SinControl,P_Min_SinControl,'--r','LineWidth',2)
hold on
plot(T_SinControl,P_Prom_SinControl,'--b','LineWidth',2)
hold on
plot(T_Control,P_Control,'Color',[0 0.25 0.25],'LineWidth',1)
hold on
plot(T_Control,P_Max_Control,':','Color',[1 0.75 0.75],'LineWidth',2)
hold on
plot(T_Control,P_Min_Control,':r','LineWidth',2)
hold on
plot(T_Control,P_Prom_Control,':b','LineWidth',2)
title('Potencia Batería en Vuelo Con y Sin Control')
xlabel('Tiempo [min]')
ylabel('Potencia [W]')

```

```
grid on
legend('P_Bat_.SinControl','P_Max_.SinControl','P_Min_.SinControl',...
       'P_Prom_.SinControl',...

       'P_Bat_.ConControl','P_Max_.ConControl','P_Min_.ConControl',...
       'P_Prom_.ConControl')
xlim([0 10.1])
ylim([0 192])

%% Procesamiento de Tabla de Datos Placa de Medicion

% Adquisición de Información

C = readtable('DATOS_Motores.csv');

T_Medicion = C.MINUTOS;

V_Medicion = C.VOLTAJEF;

I_Medicion01 = C.CORRIENTE1;

I_Medicion02 = C.CORRIENTE2;

I_Medicion03 = C.CORRIENTE3;

I_Medicion04 = C.CORRIENTE4;

I_Medicion4M = C.CORRIENTE_4M;

I_Max_Medicion01 = C.MAX_I1;

I_Max_Medicion02 = C.MAX_I2;

I_Max_Medicion03 = C.MAX_I3;

I_Max_Medicion04 = C.MAX_I4;

I_Max_Medicion4M = C.MAX_I4M;

I_Min_Medicion01 = C.MIN_I1;

I_Min_Medicion02 = C.MIN_I2;

I_Min_Medicion03 = C.MIN_I3;

I_Min_Medicion04 = C.MIN_I4;

I_Min_Medicion4M = C.MIN_I4M;

I_Prom_Medicion01 = C.PROM_I1;

I_Prom_Medicion02 = C.PROM_I2;

I_Prom_Medicion03 = C.PROM_I3;
```

```
I_Prom_Medicion04 = C.PROM_I4;

I_Prom_Medicion4M = C.PROM_I4M;

I_Prom_MedicionCM = C.PROM_ICM;

% Graficación Individual de Corriente

% Graficación Corriente Motor 1

figure
plot(T_Medicion,I_Medicion01,'k','LineWidth',1)
hold on
plot(T_Medicion,I_Max_Medicion01,'--g','LineWidth',2)
hold on
plot(T_Medicion,I_Min_Medicion01,'--r','LineWidth',2)
hold on
plot(T_Medicion,I_Prom_Medicion01,'--b','LineWidth',2)
title('Corriente Motor 1 en Vuelo Sin Control Autónomo')
xlabel('Tiempo [min]')
ylabel('Corriente [A]')
grid on
legend('I_M_o_t_o_r_1','I_M_a_x','I_M_i_n','I_P_r_o_m')
xlim([0 9.1])
ylim([0 5])

% Graficación Corriente Motor 2

figure
plot(T_Medicion,I_Medicion02,'k','LineWidth',1)
hold on
plot(T_Medicion,I_Max_Medicion02,'--g','LineWidth',2)
hold on
plot(T_Medicion,I_Min_Medicion02,'--r','LineWidth',2)
hold on
plot(T_Medicion,I_Prom_Medicion02,'--b','LineWidth',2)
title('Corriente Motor 2 en Vuelo Sin Control Autónomo')
xlabel('Tiempo [min]')
ylabel('Corriente [A]')
grid on
legend('I_M_o_t_o_r_2','I_M_a_x','I_M_i_n','I_P_r_o_m')
xlim([0 9.1])
ylim([0 5.5])

% Graficación Corriente Motor 3

figure
plot(T_Medicion,I_Medicion03,'k','LineWidth',1)
hold on
plot(T_Medicion,I_Max_Medicion03,'--g','LineWidth',2)
hold on
plot(T_Medicion,I_Min_Medicion03,'--r','LineWidth',2)
hold on
plot(T_Medicion,I_Prom_Medicion03,'--b','LineWidth',2)
title('Corriente Motor 3 en Vuelo Sin Control Autónomo')
xlabel('Tiempo [min]')
ylabel('Corriente [A]')
```

```

grid on
legend('I_M_o_t_o_r_3','I_M_a_x','I_M_i_n','I_P_r_o_m')
xlim([0 9.1])
ylim([0 3.5])

```

```
% Graficación Corriente Motor 4
```

```

figure
plot(T_Medicion,I_Medicion04,'k','LineWidth',1)
hold on
plot(T_Medicion,I_Max_Medicion04,'--g','LineWidth',2)
hold on
plot(T_Medicion,I_Min_Medicion04,'--r','LineWidth',2)
hold on
plot(T_Medicion,I_Prom_Medicion04,'--b','LineWidth',2)
title('Corriente Motor 4 en Vuelo Sin Control Autónomo')
xlabel('Tiempo [min]')
ylabel('Corriente [A]')
grid on
legend('I_M_o_t_o_r_4','I_M_a_x','I_M_i_n','I_P_r_o_m')
xlim([0 9.1])
ylim([0 3.5])

```

```
% Graficación Corriente Total Motores
```

```

figure
plot(T_Medicion,I_Medicion4M,'k','LineWidth',1)
hold on
plot(T_Medicion,I_Max_Medicion4M,'--g','LineWidth',2)
hold on
plot(T_Medicion,I_Min_Medicion4M,'--r','LineWidth',2)
hold on
plot(T_Medicion,I_Prom_Medicion4M,'--b','LineWidth',2)
title('Corriente Total de Motores en Vuelo Sin Control Autónomo')
xlabel('Tiempo [min]')
ylabel('Corriente [A]')
grid on
legend('I_M_o_t_o_r_T','I_M_a_x','I_M_i_n','I_P_r_o_m')
xlim([0 9.1])
ylim([0 17])

```

```
%% Graficacion de Corriente total de la fuente vs. Corriente total de
motores
```

```

figure
plot(T_SinControl,I_SinControl,'Color',[0 0.75 0.75],'LineWidth',1)
hold on
plot(T_SinControl,I_Max_SinControl,'--','Color',[1 0.75
0.75],'LineWidth',2)
hold on
plot(T_SinControl,I_Min_SinControl,'--r','LineWidth',2)
hold on
plot(T_SinControl,I_Prom_SinControl,'--b','LineWidth',2)
hold on
plot(T_Medicion,I_Medicion4M,'Color',[0 0.25 0.25],'LineWidth',1)
hold on
plot(T_Medicion,I_Max_Medicion4M,':','Color',[1 0.75 0.75],'LineWidth',2)
hold on

```

```

plot(T_Medicion,I_Min_Medicion4M,':r','LineWidth',2)
hold on
plot(T_Medicion,I_Prom_Medicion4M,':b','LineWidth',2)
title('Corriente Bateria vs. Corriente Total Motores')
xlabel('Tiempo [min]')
ylabel('Corriente [A]')
grid on
legend('I_Bat_', 'I_Max_Bateria', 'I_Min_Bateria', ...
.
.
.
'I_Prom_Bateria', ...
'I_Mot_', 'I_Max_Mot_', 'I_Min_Mot_', ...
'I_Prom_Mot_')
xlim([0 9.1])
ylim([0 17.5])

% Graficacion de Corrientes de cada motor

figure
plot(T_Medicion,I_Medicion01,'Color',[0 0.25 0.25],'LineWidth',1)
hold on
plot(T_Medicion,I_Medicion02,'Color',[1 0.75 0.75],'LineWidth',2)
hold on
plot(T_Medicion,I_Medicion03,'r','LineWidth',2)
hold on
plot(T_Medicion,I_Medicion04,'b','LineWidth',2)
hold on
plot(T_Medicion,I_Prom_MedicionCM,':','Color',[0 0.75
0.75],'LineWidth',2)
title('Corrientes de todos los motores')
xlabel('Tiempo [min]')
ylabel('Corriente [A]')
grid on
legend('I_M_1', 'I_M_2', 'I_M_3', 'I_M_4', 'I_Promedio')
xlim([0 9.1])
ylim([0 5.5])

```



**ANEXO No. 9**

**MANUAL DE USUARIO**

**SISTEMA DE CONTROL Y OPTIMIZACIÓN  
DE ENERGÍA PARA DRONES  
CUADRICOPTEROS  
DURANTE VUELO ESTACIONARIO**

**Autor:**

**Alex Rodrigo Ramírez Guaña  
Quito – Ecuador  
2020**

## CONTENIDO

### INDICE

1. PRECAUSIONES DE SEGURIDAD
  - a. Encendido del dispositivo
  - b. Interferencias
  - c. No moje el dispositivo
  - d. Úselo correctamente
  - e. Acerca del dispositivo
  - f. Conexión y desconexión del dispositivo
  
2. INTRODUCCIÓN
  
3. ARQUITECTURA DEL SISTEMA DE CONTROL Y OPTIMIZACIÓN DE ENERGÍA PARA DRONES CUADRICÓPTEROS DURANTE VUELO ESTACIONARIO
  
4. IDENTIFICACIÓN DE LOS MÓDULOS DEL CUADRICÓPTERO
  - Módulos prescindibles
  - Módulos imprescindibles
  
5. CONSIDERACIONES
  
6. E-MAIL Y SOPORTE TÉCNICO



## **1. PRECAUSIONES DE SEGURIDAD**

El presente sistema de control y optimización de energía para drones (UAV's) cuadricópteros adquirido está direccionado para personas mayores de 15 años de edad, que tengan bases sólidas de electrónica y conocimiento del mundo de drones o UAV's, con estructura modular, puesto que en su ensamblaje y puesta en marcha estará expuesto a señales considerables de corriente DC que provee su batería.

El presente sistema es compatible principalmente con drones cuadricópteros de tipo modular, puede ser aplicable a otros dispositivos de explotación terrena, acuática, juguetes de estructura modular y de código abierto después de una revisión y modificación en la estructura del vehículo no tripulado.

En los casos de sistemas embebidos y de código propietario no es recomendable su aplicación. Evitar realizar ésta adaptación si su drone no es modular. No conectar la batería de su UAV antes de finalizar con la adaptación e implementación del sistema en su equipo.

### **a. Encendido del dispositivo**

Una vez montado el sistema de control y optimización de energía en su UAV, debe tomar en consideración que la energía para el funcionamiento de la tarjeta se brindará a través de la batería principal del equipo.

### **b. Interferencias**

El usuario deberá elegir una ubicación estratégica en su equipo para garantizar conexiones seguras y contactos de terminales adecuados.

### **c. No moje el dispositivo**

Una vez instalado la placa electrónica en su UAV, tener precaución de no salpicar líquidos o exponer al equipo a condiciones climáticas adversas, se deberá proteger la electrónica adicional implementada.

### **d. Úselo correctamente**

No alterar las conexiones del sistema y tampoco adicionar un mayor número de módulos, puesto que el dispositivo está dimensionado únicamente para el control de cinco módulos.

#### **e. Acerca del dispositivo**

La placa del sistema de control y optimización de energía tiene como parte principal una placa de desarrollo Arduino NANO con un microcontrolador ATmega128P de dimensiones y características ligeras.

#### **f. Conexión y desconexión del dispositivo**

Tener precaución en la conexión correcta de los terminales de cada módulo a intervenir y controlar, se debe conectar correctamente cada terminal. Tener siempre presente que es importante que se mantenga la batería del dispositivo totalmente cargada.

## **2. INTRODUCCIÓN**

La tarjeta de control y optimización de energía para drones cuadricópteros durante vuelo estacionario es la encargada de gestionar la energía suministrada a cada uno de los cinco módulos del UAV considerados. Será capaz de interrumpir el suministro de corriente en función del estado de descarga la batería. El sistema en determinado instante de funcionamiento considerará la necesidad de iniciar el proceso de ahorro de energía, la que puede ser direccionada para varios fines como precautelar la integridad del equipo con un aterrizaje controlado o aterrizaje de emergencia.

## **3. ARQUITECTURA DEL SISTEMA DE CONTROL Y OPTIMIZACIÓN DE ENERGÍA PARA DRONES CUADRICÓPTEROS DURANTE VUELO ESTACIONARIO**

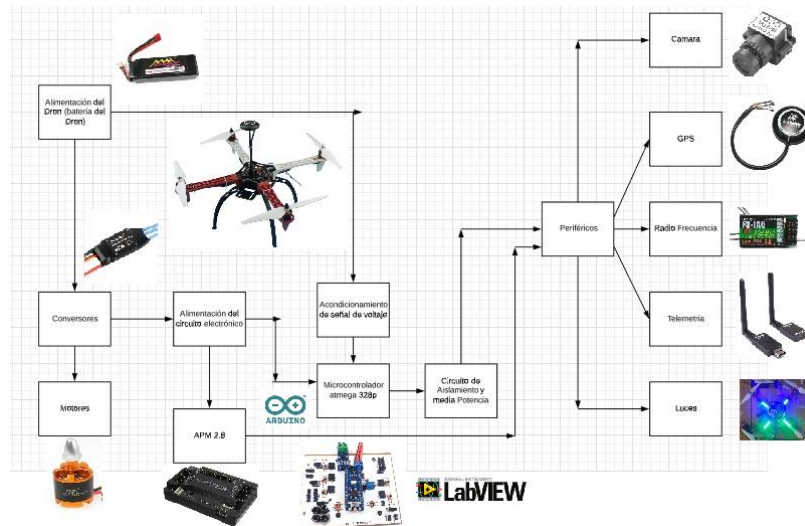


Figura 1. Arquitectura del sistema de control y optimización de energía

La placa de control y optimización de energía es desmontable y dispone de terminales de conexión rápida, tipo espadín con los cables suficientes para realizar las conexiones con la controladora de vuelo y con cada uno de los cinco módulos a controlar. El sistema de control y optimización de energía trabaja sobre las señales de alimentación ( $V_{cc}$ ) de los diferentes módulos de los que disponga su UAV y a través de ellas realiza la su desconexión y gestión optimización de energía dependiendo de la capacidad de corriente que mantenga la batería.

#### 4. IDENTIFICACIÓN DE LOS MÓDULOS DEL CUADRICÓPTERO

Como se puede observar en la Figura 1, del presente manual, dentro de la arquitectura del sistema desarrollado el usuario deberá identificar correctamente cada uno de los módulos de su equipo a intervenir e identificar correctamente sus terminales de alimentación ( $V_{cc}$ , Gnd, Señal), para su conexión en la tarjeta de control. Para el caso de los UAV's drones cuadricópteros de estructura modular como de la Figura 2, éstos comúnmente disponen de módulos prescindibles e imprescindibles como los siguientes:

##### Módulos prescindibles

## Módulo de iluminación



Figura 2. Módulo de iluminación

## Telemetría por Radio frecuencia RF



Figura 3. Módulo RF de telemetría

## Sistema FPV o cámara



Figura 4. Sistema FPV

## Sistema de Posicionamiento Global (GPS)



Figura 5. Módulo GPS

## Radio Frecuencia RF



Figura 6. Módulo de iluminación

## Módulos imprescindibles (no modificar)

Dentro de los drones cuadricópteros y UAV's de estructura modular existen módulos que no se pueden intervenir porque dan la funcionalidad principal del equipo, generalmente constituyen los siguientes:

## Controladora de vuelo

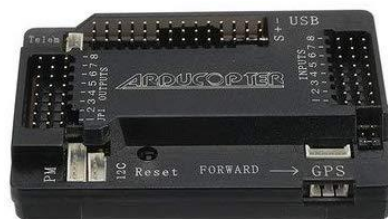


Figura 7. Controladora de vuelo APM 2.8

## Módulo de propulsión



Figura 8. Módulo de propulsión

## Batería



Figura 9. Batería lipo

El usuario deberá adaptar los terminales de cada uno de los módulos a intervenir, con el fin de conectarlos en la placa de control, así también su batería según se indica en la Figura 10.

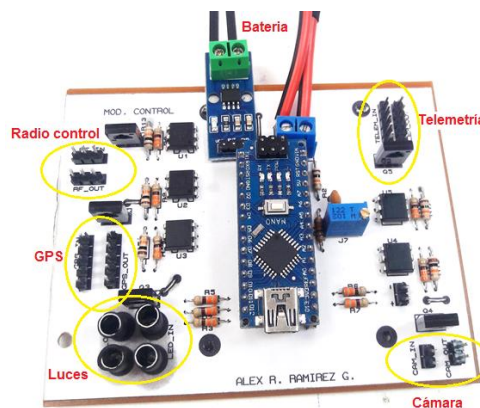


Figura 10. Conexión de módulos a controlar

## 5. CONSIDERACIONES

A continuación, se detallan las consideraciones necesarias para poner en marcha el sistema de medición y control diseñado. Para ello, se asume que el sistema de control ha sido correctamente instalado en el dron:

### **Programar el microcontrolador:**

Para realizar este proceso se requiere disponer de un computador con el software ARDUINO IDE instalado, además del código fuente desarrollado para la tarjeta de control.

Se procede a conectar, mediante el cable correspondiente, el microcontrolador hacia el computador. Si la conexión se ha realizado correctamente, entonces el computador asigna un puerto serial de tipo COM para la comunicación con el módulo.

En el ARDUINO IDE se procede a definir el puerto anteriormente creado para la comunicación. Esto se realiza en la pestaña “Herramientas – PUERTO” en la Barra de Tareas ubicada en la parte superior del programa.

Finalmente, se procede a transferir el programa mediante el botón “Subir” ubicado igualmente en la parte superior del programa. Esta acción lleva a ejecutar la compilación del programa, la construcción del archivo “.hex” y la transferencia del mismo hacia el microcontrolador.

Si la transferencia del archivo se ha realizado correctamente, el programa procede a ejecutar el reinicio del microcontrolador, después de lo cual se ejecutará el nuevo programa cargado.

### **Visualizar información a través de la interfaz creada en LabView:**

El proceso de visualización se lleva a cabo bajo las mismas condiciones que el proceso de programación del microcontrolador. Esto es, a través de un cable que permite la comunicación serial entre el computador y la tarjeta. Otro requerimiento importante es que el microcontrolador ya debe encontrarse energizado.

Una vez se haya conectado ambos elementos y tras haberse creado el puerto COM correspondiente, primeramente, es necesario verificar que ningún programa se encuentre haciendo uso de dicho puerto. Para ello, se recomienda terminar cualquier proceso que involucre el uso de dicho puerto, como por ejemplo los monitores seriales disponibles en ARDUINO IDE.

Inmediatamente, se procede a asignar dicho puerto para comunicación en la interfaz desarrollada en LabView. Esto se realiza preferiblemente sin todavía ejecutar el programa. Antes de ejecutar el proceso de adquisición de datos, es importante verificar que el directorio donde se ha de crear el archivo se encuentre vacío. El proceso de adquisición sobre escribe cualquier archivo que anteriormente haya sido creado, lo que implicaría la pérdida de información.

Una vez asignado el puerto y garantizando que la conexión física es suficientemente larga como para no interferir con el funcionamiento de las hélices del dron, se procede a poner en marcha la interfaz mediante el botón “Run” disponible en la parte superior del programa en LabView.

### **Gestión de la información obtenida desde LabView**

Una vez se ha efectuado la totalidad del vuelo autónomo y tras verificarse el aterrizaje del dron, se procede a parar la adquisición de información por parte de la interfaz de LabView mediante el botón “Stop” en la parte superior del programa.

La finalización del esquema de adquisición de datos involucra la creación de una hoja de cálculo donde se registran todas las mediciones de Voltaje, Corriente, Potencia y Tiempo transcurrido para el vuelo autónomo. Es importante observar que el archivo se encuentra únicamente disponible después de detener el proceso de adquisición. No resulta posible acceder a dicho archivo mientras la interfaz se encuentra funcionando.

Para visualizar la información obtenida se procede a abrir el archivo que se encontrará alojado en el directorio que el usuario haya definido previamente. En este archivo se encontrarán todos los datos en forma de tabla, listos para ser procesados según el usuario considere conveniente.



**Observaciones sobre el Sistema de Control:**

El sistema de control ha sido diseñado para conseguir una gestión autónoma de optimización de la energía disponible en el dron. Esto implica que el usuario no tiene la posibilidad de comandar el apagado de módulos, pero sí de observar las etapas de apagado de cada uno de los módulos en función de la descarga de la batería.

El Sistema de Control ha sido diseñado tomando en cuenta una batería LIPO de 3 celdas, 11.1 V. Esto es importante debido a que, dado que el esquema de control se realiza por voltaje, entonces la alimentación del dron con una batería de mayor o menor voltaje implicará la generación de cortes en puntos de trabajo donde la batería seguramente no está diseñada para trabajar.

**6. CONTACTOS Y SOPORTE TÉCNICO**

Asesor técnico

Ing. Alex Ramírez G

Correo electrónico

[alexrodrigorg@gmail.com](mailto:alexrodrigorg@gmail.com)

Teléfonos

0999353898 / 023034103



**ANEXO No. 10**

**MANUAL TÉCNICO**

**SISTEMA DE CONTROL Y OPTIMIZACIÓN  
DE ENERGÍA PARA DRONES  
CUADRICOPTEROS DURANTE VUELO  
ESTACIONARIO**

**Autor:**

**Alex Rodrigo Ramírez Guaña  
Quito – Ecuador  
2020**

## **CONTENIDO**

### **1. OBJETIVO**

Arquitectura de funcionamiento del sistema

### **2. REQUERIMIENTOS DEL SISTEMA**

Requerimientos de hardware

Módulos de drone

Características técnicas

PCB del sistema de control y optimización

Descripción del sistema

Elementos a utilizar en la etapa de control

### **3. RECOMENDACIONES**

### **4. CONSIDERACIONES A FALLOS Y MANTENIMIENTO**

### **5. E-MAIL Y SOPORTE TÉCNICO**

## 1. OBJETIVO

El presente manual tiene como objetivo brindar al personal técnico una breve descripción del sistema implementado, con el fin de proporcionar conocimiento acerca del funcionamiento, implementación, comprobación y puesta en marcha del sistema, así como brindar las recomendaciones necesarias para dar solución a posibles fallas que pudieran presentarse en su funcionamiento.

### a. Arquitectura de funcionamiento del sistema implementado

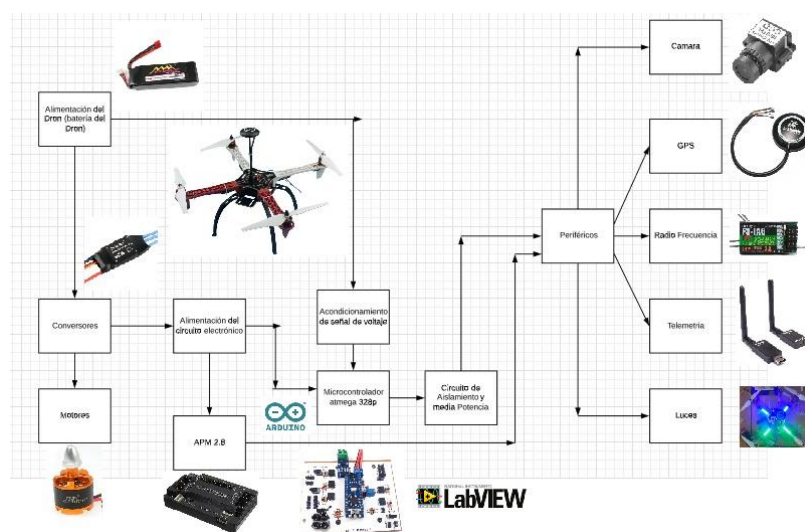


Figura 1. Arquitectura de aplicación de sistema

La lógica de desconexión de los módulos que forman parte del dron o UAV, es desconectar las señales de alimentación ( $V_{cc}$ ) tomando como referencia la variable estado de descarga de la batería, es decir el valor de voltaje en la que se encuentre la batería durante un instante de tiempo durante su vuelo estacionario.

## 2. REQUERIMIENTOS DEL SISTEMA

### Requerimientos de Hardware

Para la aplicación del presente sistema de control y optimización de energía, el dron que disponga el usuario debe ser de tipo modular, es decir, que varios de los

módulos componentes del drone puedan ser claramente identificados y estén conectados directamente a la controladora de vuelo y en los que se pueda identificar claramente las señales de voltaje, gnd y señal, como en los siguientes módulos.

### **Módulo de Radiocontrol (RF)**

#### Características:

Frecuencia de trabajo: 2.4GHz

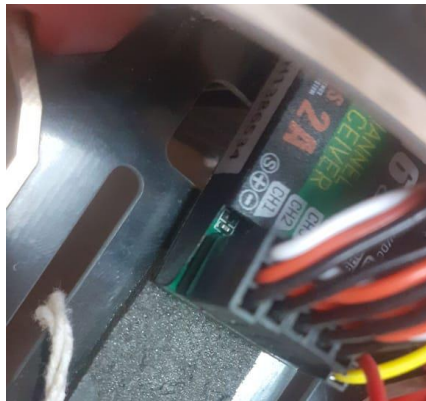


Figura 2. Conectores de módulo de radio control RF montado en el APM 2.8

Este módulo posee de 3 pines de los cuales se realizará la conexión y desconexión del cable de alimentación.

### **Módulo de GPS (Sistema de Posicionamiento Global)**

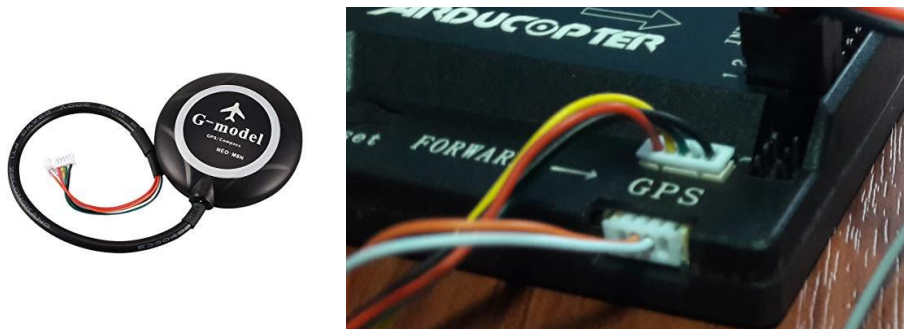


Figura 3. Conectores de Sistema GPS conectados en APM 2.8

### Características

Brújula GPS de 19.7 pies especial para controlador de piloto automático PIX PX4.

Antena de cerámica de alta ganancia integrada de 0.984 x 0.984 x 0.157 in

Con la brújula 5983 – EPROM integrado, para asegurarse que no se pierdan datos

Protección de polaridad inversa incorporada

LED de dos colores integrado, indicación clara del estado del GPS

Con carcasa redonda GPS y soporte para los modelos RC

Especialmente diseñado para Ardupilot Ardupilot APM 2.6-2.8 controlador de vuelo

### Conectores:

Negro: GND.

Verde: TX.

Amarillo: RX.

Rojo:+5V

Blanco: SDA.

Naranja: SCL

Todos los pines son terminaciones tipo hembra por lo que se hará uso de conectores macho en la placa para la correcta conexión entre ellos. Para la desconexión de este módulo se desconectará y volverá a conectar la alimentación de 5 voltios (cable rojo).

### **Módulo de Iluminación**

Se utilizará 4 luces que serán conectadas a la placa de control mediante borneras ubicadas en la placa de control.

### Características:

Alimentación 12 voltios.

Las luces disponen de dos cables de alimentación y se realizará la conexión y desconexión del terminal positivo para controlar el encendido y apagado de las luces.



Figura 4. Módulo de iluminación

## Sistema FPV o Cámara.



Figura 5. Módulo de iluminación

### Cámara y cables de alimentación

Se ha colocado dos cables para controlar la alimentación de la cámara, en donde el cable negro es Gnd y el cable rojo será el terminal positivo de la fuente de 5 voltios. Estos dos cables irán a borneras ubicadas en la placa de control y se conectarán directamente el cable Gnd a la referencia de la batería, mientras que el terminal positivo se controlará mediante un semiconductor el estado de encendido y apagado de la cámara.

### Telemetría

Una vez se haya instalado y configurado correctamente este módulo permite la monitorización del dron a tiempo real para conocer las variables más importantes durante el vuelo.

#### Características

Módulo de transmisión de datos de aire y tierra FPV de 915 Mhz para controlador de vuelo APM 2.8 Pixhawk PIX4

#### Especificaciones:

Canal de frecuencia: 915MHz.

Sensibilidad de recepción: -117dBm

Potencia de transmisión: hasta 20 dBm (100 mW)

Conector de antena: RP-SMA

Potencia de salida del conector: 100 mW (20 dBm)

La sensibilidad se puede ajustar entre 1 y 20 dBm

Interfaz de sensibilidad: estándar TTL UART

Soporte OTG de teléfonos móviles Android y OTG de ordenador. La antena es de 5.8G.

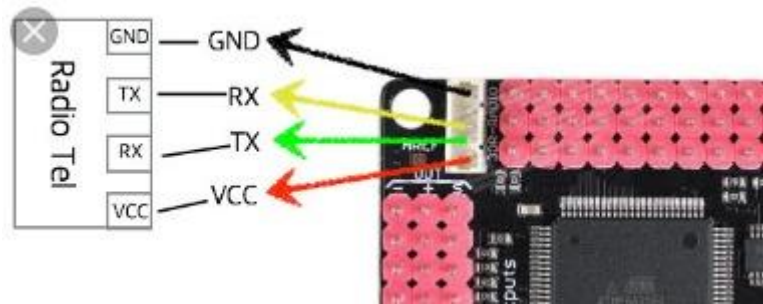


Figura 6. Módulo de telemetría RF

### Ubicación del módulo de telemetría.

En la placa de control se enviará las cuatro señales expuestas en la Figura 5, Gnd, Rx, Tx, Vcc a la placa de control y en la placa de control se permitirá el paso de corriente en los terminales Gnd, RX, Tx. Mientras que la señal Vcc irá a un interruptor que permitirá el control del encendido y apagado del módulo de telemetría. Dicho control se lo realizará mediante un microcontrolador y un semiconductor. Y las 4 señales saldrán hacía el APM 2.8.

### Diagrama circuital del sistema de control y optimización

De acuerdo a todo lo especificado anteriormente, a continuación, se presenta el diagrama circuital de la tarjeta electrónica de control diseñada en Proteus 8.6 que será capaz de controlar el estado de encendido o apagado de los módulos que componen el Dron.

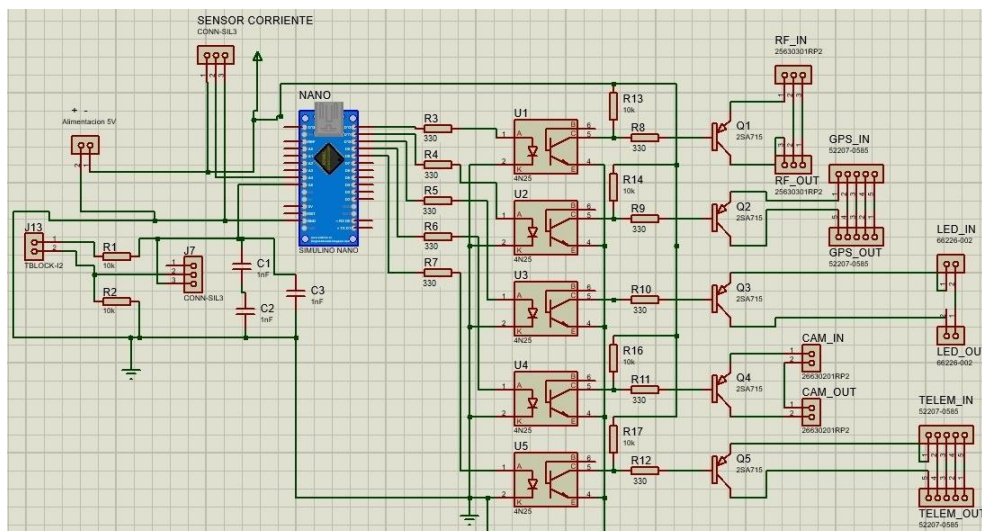


Figura 7. Diagrama circuital de Sistema de control



## Descripción del sistema de gestión de energía ON/OFF para los módulos

Se recibirán todos los elementos tal y como se conectarían al DRONE (conexiones originales). Una vez en la placa, no se interviene sobre ningún cable que involucre tratamiento de señales, sino que, a través de MOSFETs se permite el paso o no de corriente hacia dicho módulo.

Por otro lado, a fin de proteger la salida digital de la placa Arduino, se han incluido elementos de aislamiento tales como el circuito 4N25, el cual es un opto acoplador que dispara la base de su transistor según se haya polarizado el diodo.

### Opto acoplador 4N25

Opto acoplador basado en un fototransistor. Cada opto acoplador consta de un LED infrarrojo y un fototransistor NPN. Usado para aislamiento de elementos lógicos, rechazo de ruidos. Aplicaciones AC, etc.

#### Características

Voltaje aislamiento hasta 5000 VRMS

Fácil inclusión en familias lógicas comunes

Capacitancia de acoplamiento <0.5pF

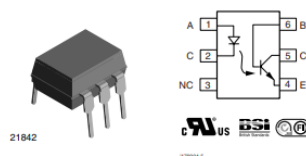


Figura 8. Opto acoplador 4N25

### Transistor 2SA715

Transistor PNP de media potencia.

#### Características

Voltaje Colector Base 35 V

Voltaje Colector Emisor -35 V

Voltaje Base Emisor 5 V

Disipación hasta 10 W

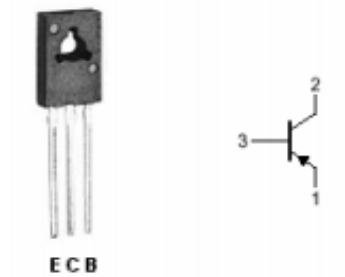


Figura 9. Transistor PNP 2SA715

### Placa de Control desarrollada en Proteus 8.6

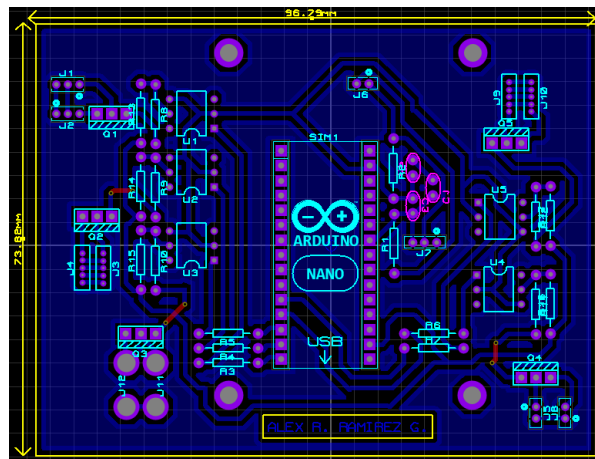


Figura 10. PCB de Sistema de control y optimización de energía

De manera general, puede decirse que la tarjeta cumplirá el objetivo de gestionar o comandar el encendido/apagado de los módulos en función del consumo de corriente de la batería.

Así, se definirá un valor (setpoint) de capacidad de corriente a partir del cual se considerará la necesidad de iniciar el proceso de ahorro de energía. En este sentido es importante observar que el mayor ahorro de energía que podría obtenerse vendrá dado por la desconexión de absolutamente todos los módulos capaces de ser desconectados y que aseguren un aterrizaje seguro.

## ELEMENTOS A UTILIZAR EN LA ETAPA DE CONTROL

### Sensor de Corriente ACS-712 5A y ACS-712 30<sup>a</sup>

El sensor de corriente ACS712 es una solución económica para medir corriente, internamente trabaja con un sensor de efecto Hall que detecta el campo magnético que se produce por inducción de la corriente que circula por la línea que se está midiendo.

El sensor entrega una salida de voltaje proporcional a la corriente, dependiendo la aplicación se puede usar el ACS712-05A, ACS712-20A o el ACS712-30A, para rangos de 5, 20 o 30 amperios respectivamente.

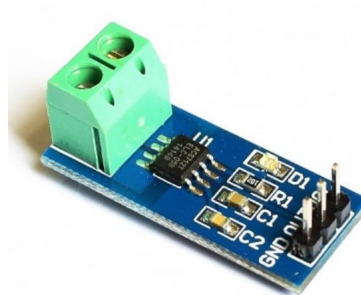


Figura 11. Módulo sensor de corriente ACS 712

### Arduino NANO

Arduino Nano es una placa de desarrollo de tamaño compacto, completa y compatible con protoboards, basada en el microcontrolador ATmega328P. Tiene 14 pines de entrada/salida digital (de los cuales 6 pueden ser usando con PWM), 6 entradas analógicas, un cristal de 16Mhz, conexión Mini-USB, terminales para conexión ICSP y un botón de reseteo.

Posee las mismas capacidades que un Arduino UNO, tanto en potencia del microcontrolador como en conectividad, solo se ve recortado en su conector USB, conector jack de alimentación y los pines cambia un formato de pines header.

Esta placa posee las siguientes especificaciones:

- Microcontrolador: ATmega328.
- Voltaje de operación: 5V.
- Voltaje de alimentación (Recomendado): 7-12V.

- I/O Digitales: 14 (6 son PWM)
- Entradas Analógicas: 8
- Memoria Flash: 32KB.
- EEPROM: 1KB.
- Frecuencia de trabajo: 16MHz.
- Dimensiones: 0.73" x 1.70"

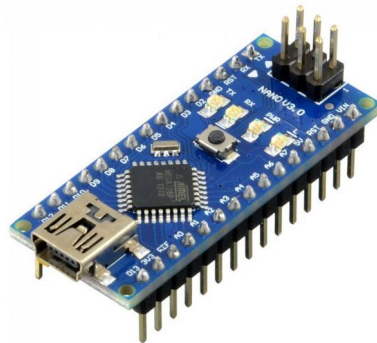


Figura 12. Arduino NANO

### 3. RECOMENDACIONES

- El producto creado es ligero y adaptativo agrega un peso de aproximadamente 70 gr, prácticamente despreciable al dron.
- El sistema de control y optimización de energía viene programado.
- El consumo de corriente que requiere el sistema de control y optimización es mínimo, está en el orden de los 19 mA
- Puede ser aplicado a drones cuadricópteros de tipo modular y de código abierto realizando algunas adecuaciones físicas en el hardware es aplicable a drones cuadricópteros con ésta característica.
- El sistema de control es desmontable, poco invasivo.
- En sistemas embebidos y de código propietario, es poco probable que el sistema sea aplicable, sujeto a revisión y análisis.
- De requerir medición en el laboratorio el dron debe estar conectado al software LabView a través de comunicación serial cableada y sostenido para simular un vuelo estacionario.

#### 4. CONSIDERACIONES A FALLOS Y MANTENIMIENTO

A continuación se detallan diferentes consideraciones para tener en cuenta en términos de fallos y mantenimiento de las placas desarrolladas:

##### Posibles fallos:

- **La tarjeta no efectúa el apagado de algún módulo.**
  - a. Verificar que el proceso de cableado de cada módulo se haya realizado correctamente.
  - b. Verificar que el sensor de voltaje está recibiendo la señal debidamente polarizada.
  - c. Con ayuda del diagrama circuital y la nomenclatura de elementos, realizar un proceso de seguimiento de la señal de control desde la placa Arduino hasta su llegada al transistor.
  
- **Las mediciones de voltaje o corriente se encuentran descalibradas al observar en LabView.**
  - a. Verificar que los puntos de adquisición de las señales en la tarjeta se han conectado correctamente.
  - b. Con ayuda de la información correspondiente a la etapa de calibración, verificar que los voltajes de entrada hacia la tarjeta Arduino correspondan con los valores acondicionados.
  - c. En caso de observar la señal de voltaje descalibrada, puede manipularse levemente el potenciómetro hasta acondicionar nuevamente dicha señal.
  - d. En caso de observar la señal de corriente descalibrada, verificar el voltaje de OFFSET entregado por el sensor sin estar sometido a ninguna carga, en caso de que dicho voltaje no sea 2.5 V o algún valor cercano, puede diagnosticarse que dicho sensor se ha dañado.
  
- **El módulo de luces no se apaga completamente.**
  - a. Si se ha comprobado que el módulo de luces conmuta de un estado de encendido hacia otro estado intermedio donde las luces no se han apagado completamente, puede diagnosticarse directamente como problema del transistor que comanda dicho módulo.

- b. Verificar el voltaje Colector – Emisor del transistor para comprobar el origen del error. Si se verifican errores en la conmutación deberá reemplazarse el transistor.
- **El dron no puede completar el proceso de “armado” previo al vuelo.**
  - a. Verificar que durante la etapa de encendido los módulos esenciales para el arranque se encuentren encendidos y debidamente conectados: GPS y Radio Frecuencia.
  - b. Particularmente verificar que los cables de conexión de datos pertenecientes al GPS se encuentren debidamente conectados, pues la mayoría de las fallas en el armado del dron tienen su causa en dicho módulo.
- **Después de realizar las conexiones, el dron no se energiza.**
  - a. Dado que el proceso de adquisición de la información de consumo de corriente viene dado por una conexión en serie con el sensor, verificar que los cables que conducen hacia dicho sensor se encuentren correctamente conectados y en perfecto estado.
- **No puede establecerse comunicación con el módulo LabView.**
  - a. Verificar que la tarjeta ha sido debidamente reconocida por el computador, en cuyo caso le es asignado un puerto de comunicación serial.
  - b. Verificar que el puerto no está siendo ocupado por ninguna otra aplicación del computador, como por ejemplo el IDE de Arduino.
  - c. Verificar que la interfaz de LabView se esté ejecutando sin errores.

## **MANTENIMIENTO**

- a. Verificar periódicamente el estado de las borneras, particularmente las borneras pertenecientes al sensor de corriente, que es por donde circulan alrededor de 15 A durante el vuelo autónomo.
- b. Verificar que la temperatura de funcionamiento de la tarjeta y de todos sus componentes, no alcance valores excesivos. En caso de encontrar temperaturas de funcionamiento muy altas, considerar el reemplazo del elemento.
- c. Verificar el estado de todos los cables necesarios para poner en funcionamiento el sistema, garantizar la continuidad en todos los puntos.

- d. Verificar que la instalación física del cableado no interfiera con el funcionamiento de las hélices del dron, lo cual podría perjudicar severamente al sistema en vuelo, como la pérdida de continuidad de los cables involucrados.
- e. Garantizar la ausencia de elementos metálicos capaces de conducir electricidad en la cercanía de la tarjeta de control o de medición. Para ello se recomienda trabajar con la cubierta diseñada específicamente para proteger dichos módulos.

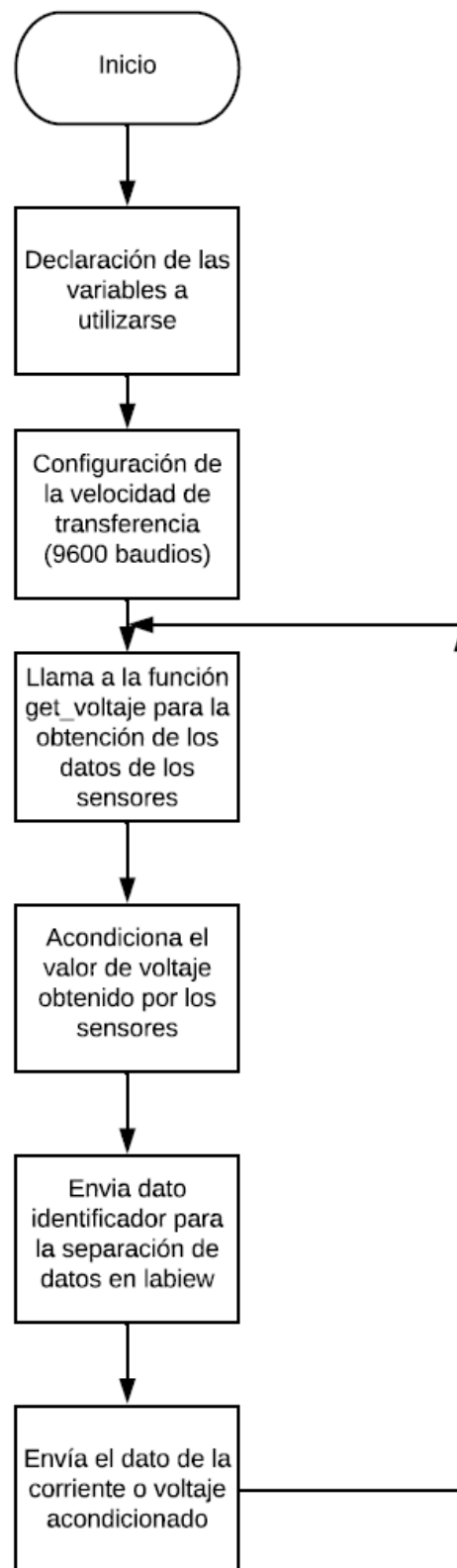
## **5. CONTACTOS Y SOPORTE TÉCNICO**

- Asesor técnico  
Ing. Alex Ramírez G
- Correo electrónico  
[alexrodrigorg@gmail.com](mailto:alexrodrigorg@gmail.com)
- Teléfonos  
0999353898 / 023034103

**ANEXO No. 11**  
**DIAGRAMA DE FLUJO SISTEMA DE MEDICIÓN (ARDUINO-  
LABVIEW)**

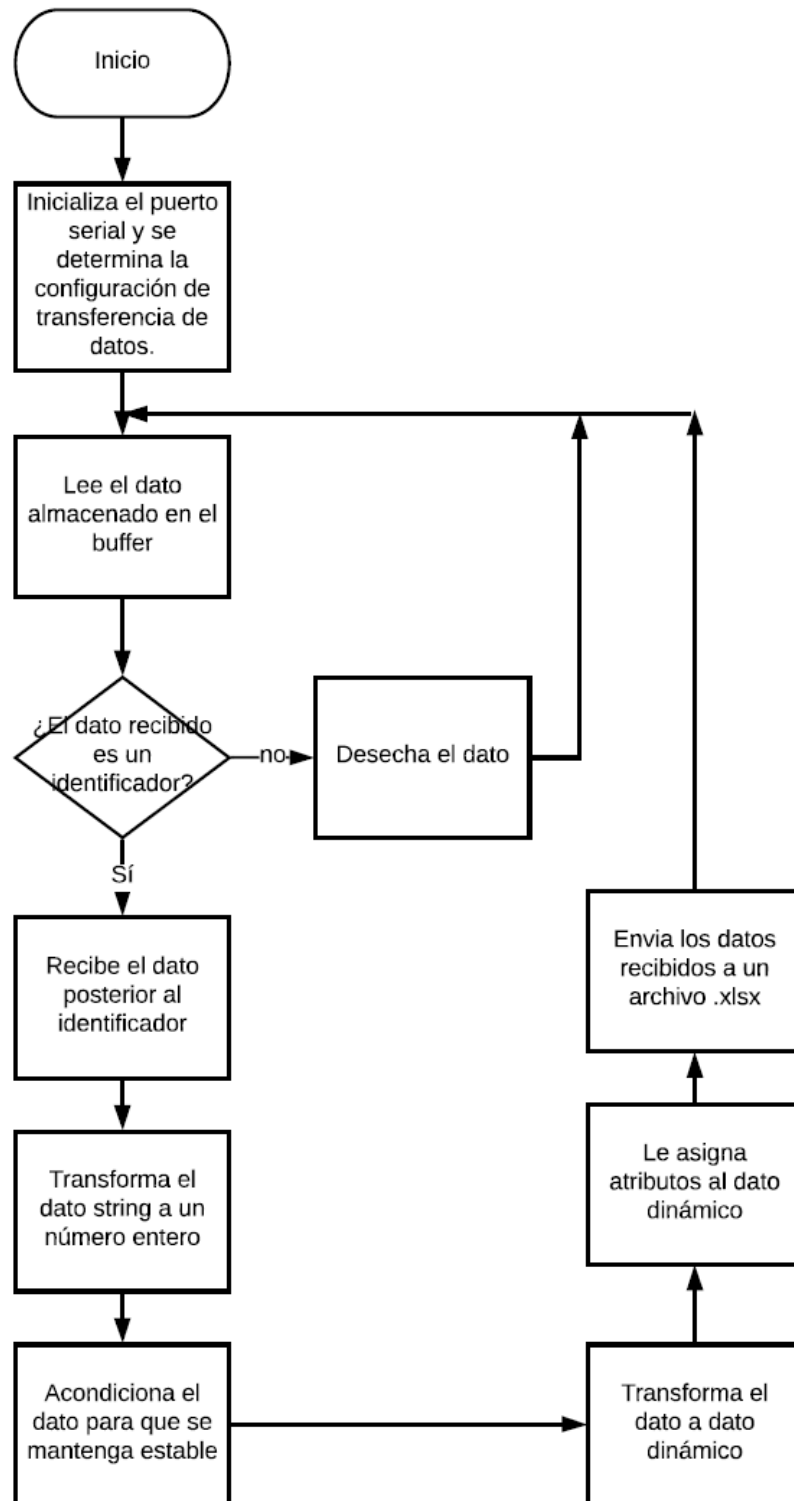


Algoritmo a implementar en placa Arduino.



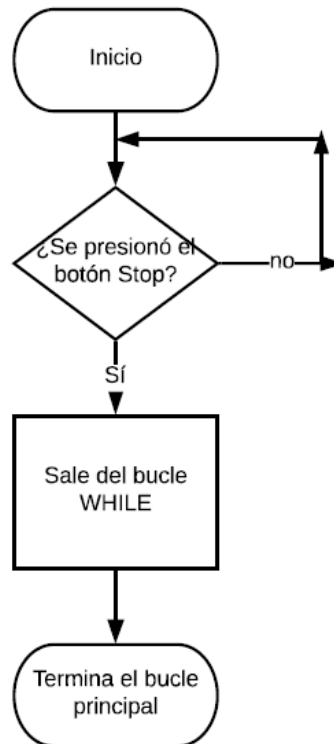
Algoritmo a implementar en LabView IDE.

Programa del lazo Principal



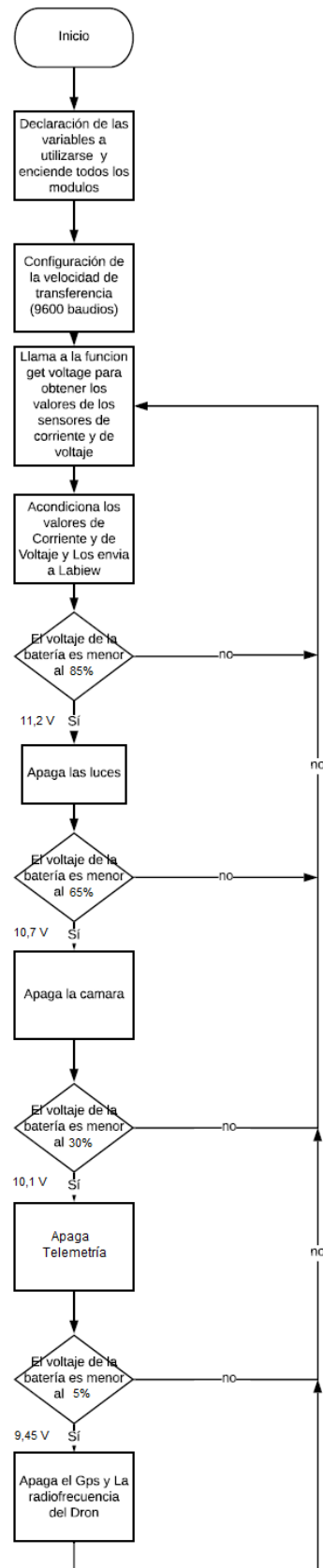
Algoritmo a implementar en LabView IDE.

Programa del Stop

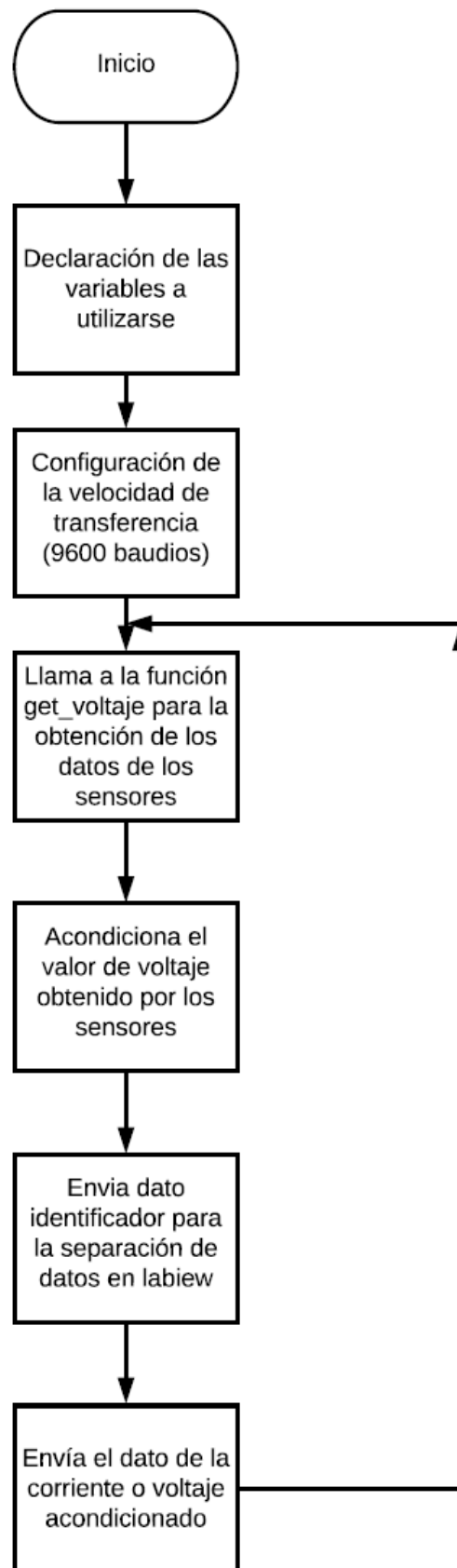


**ANEXO No. 12**  
**DIAGRAMA DE FLUJO DE SISTEMA DE CONTROL Y**  
**OPTIMIZACIÓN (ARDUINO – LABVIEW)**

Algoritmo a implementar en placa Arduino.



Algoritmo a implementar en LabView IDE.



Algoritmo a implementar en LabView IDE.

Programa del lazo Principal

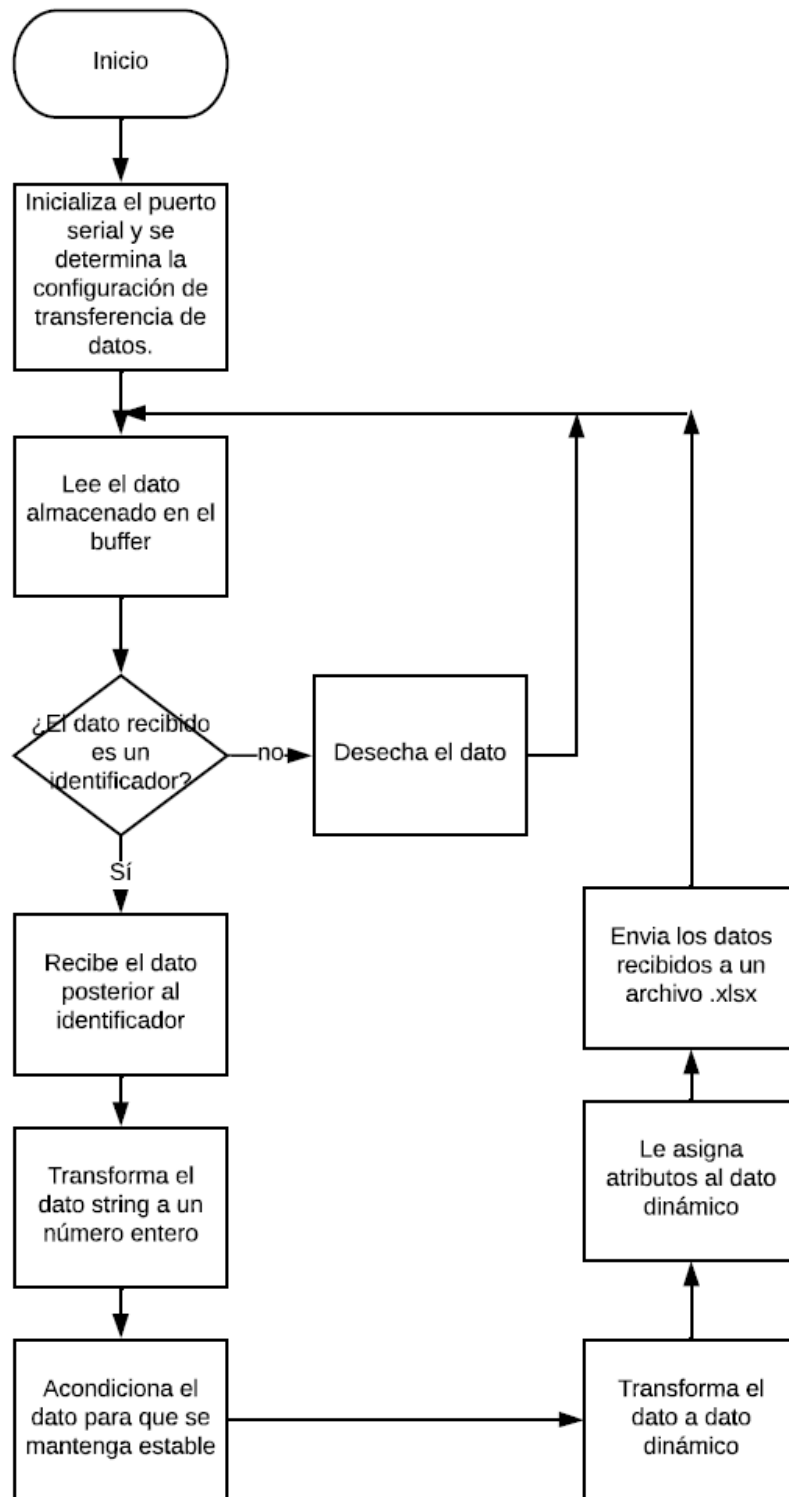
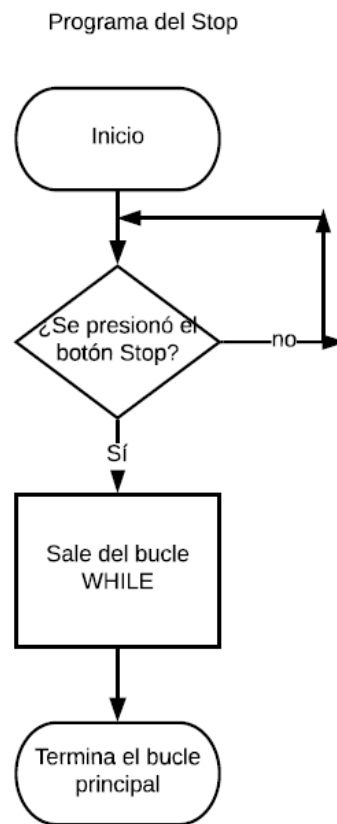


Diagrama de flujo del algoritmo a implementar en entorno de desarrollo LabView IDE.





**ANEXO No. 13**

**DATASHEET**



April 2015

## 4N25M, 4N26M, 4N27M, 4N28M, 4N35M, 4N36M, 4N37M 6-Pin General Purpose Phototransistor Optocouplers

### Features

- Minimum Current Transfer Ratio at  $I_F = 10 \text{ mA}$ ,  $V_{CE} = 10 \text{ V}$ :
  - 10% for 4N27M and 4N28M
  - 20% for 4N25M and 4N26M
  - 100% for 4N35M, 4N36M and 4N37M
- Safety and Regulatory Approvals:
  - UL1577, 4,170  $V_{AC_{RMS}}$  for 1 Minute
  - DIN-EN/IEC60747-5-5, 850 V Peak Working Insulation Voltage

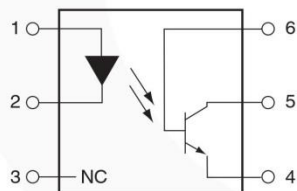
### Applications

- Power Supply Regulators
- Digital Logic Inputs
- Microprocessor Inputs

### Description

The general purpose optocouplers consist of a gallium arsenide infrared emitting diode driving a silicon phototransistor in a standard plastic six-pin dual-in-line package.

### Schematic



PIN 1. ANODE  
 2. CATHODE  
 3. NO CONNECTION  
 4. EMITTER  
 5. COLLECTOR  
 6. BASE

Figure 1. Schematic

### Package Outlines

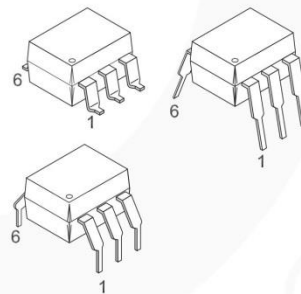


Figure 2. Package Outlines

4N25M, 4N26M, 4N27M, 4N28M, 4N35M, 4N36M, 4N37M — 6-Pin General Purpose Phototransistor Optocouplers

### Safety and Insulation Ratings

As per DIN EN/IEC 60747-5-5, this optocoupler is suitable for "safe electrical insulation" only within the safety limit data. Compliance with the safety ratings shall be ensured by means of protective circuits.

Parameter	Characteristics
Installation Classifications per DIN VDE 0110/1.89 Table 1, For Rated Mains Voltage	< 150 V <sub>RMS</sub>
	< 300 V <sub>RMS</sub>
Climatic Classification	55/100/21
Pollution Degree (DIN VDE 0110/1.89)	2
Comparative Tracking Index	175

Symbol	Parameter	Value	Unit
V <sub>PR</sub>	Input-to-Output Test Voltage, Method A, V <sub>IORM</sub> × 1.6 = V <sub>PR</sub> , Type and Sample Test with t <sub>m</sub> = 10 s, Partial Discharge < 5 pC	1360	V <sub>peak</sub>
	Input-to-Output Test Voltage, Method B, V <sub>IORM</sub> × 1.875 = V <sub>PR</sub> , 100% Production Test with t <sub>m</sub> = 1 s, Partial Discharge < 5 pC	1594	V <sub>peak</sub>
V <sub>IORM</sub>	Maximum Working Insulation Voltage	850	V <sub>peak</sub>
V <sub>IOTM</sub>	Highest Allowable Over-Voltage	6000	V <sub>peak</sub>
	External Creepage	≥ 7	mm
	External Clearance	≥ 7	mm
	External Clearance (for Option TV, 0.4" Lead Spacing)	≥ 10	mm
DTI	Distance Through Insulation (Insulation Thickness)	≥ 0.5	mm
T <sub>S</sub>	Case Temperature <sup>(1)</sup>	175	°C
I <sub>S,INPUT</sub>	Input Current <sup>(1)</sup>	350	mA
P <sub>S,OUTPUT</sub>	Output Power <sup>(1)</sup>	800	mW
R <sub>IO</sub>	Insulation Resistance at T <sub>S</sub> , V <sub>IO</sub> = 500 V <sup>(1)</sup>	> 10 <sup>9</sup>	Ω

**Note:**

1. Safety limit values – maximum values allowed in the event of a failure.



## ACS712

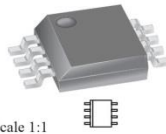
### Fully Integrated, Hall Effect-Based Linear Current Sensor IC with 2.1 kVRMS Isolation and a Low-Resistance Current Conductor

#### Features and Benefits

- Low-noise analog signal path
- Device bandwidth is set via the new FILTER pin
- 5  $\mu$ s output rise time in response to step input current
- 80 kHz bandwidth
- Total output error 1.5% at  $T_A = 25^\circ\text{C}$
- Small footprint, low-profile SOIC8 package
- 1.2 m $\Omega$  internal conductor resistance
- 2.1 kVRMS minimum isolation voltage from pins 1-4 to pins 5-8
- 5.0 V, single supply operation
- 66 to 185 mV/A output sensitivity
- Output voltage proportional to AC or DC currents
- Factory-trimmed for accuracy
- Extremely stable output offset voltage
- Nearly zero magnetic hysteresis
- Ratiometric output from supply voltage



#### Package: 8 Lead SOIC (suffix LC)



Approximate Scale 1:1

#### Description

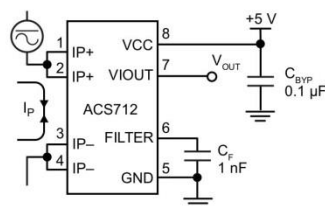
The Allegro™ ACS712 provides economical and precise solutions for AC or DC current sensing in industrial, commercial, and communications systems. The device package allows for easy implementation by the customer. Typical applications include motor control, load detection and management, switch-mode power supplies, and overcurrent fault protection. The device is not intended for automotive applications.

The device consists of a precise, low-offset, linear Hall circuit with a copper conduction path located near the surface of the die. Applied current flowing through this copper conduction path generates a magnetic field which the Hall IC converts into a proportional voltage. Device accuracy is optimized through the close proximity of the magnetic signal to the Hall transducer. A precise, proportional voltage is provided by the low-offset, chopper-stabilized BiCMOS Hall IC, which is programmed for accuracy after packaging.

The output of the device has a positive slope ( $>V_{IOUT(Q)}$ ) when an increasing current flows through the primary copper conduction path (from pins 1 and 2, to pins 3 and 4), which is the path used for current sampling. The internal resistance of this conductive path is 1.2 m $\Omega$  typical, providing low power loss. The thickness of the copper conductor allows survival of

*Continued on the next page...*

#### Typical Application

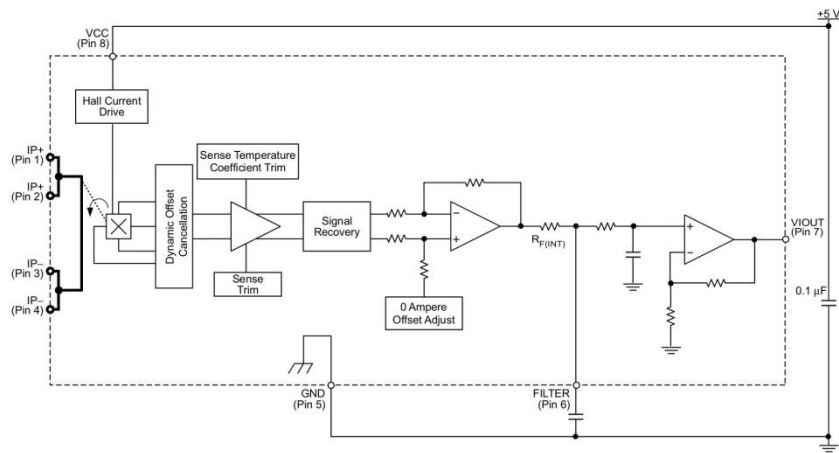


Application 1. The ACS712 outputs an analog signal,  $V_{OUT}$ , that varies linearly with the uni- or bi-directional AC or DC primary sampled current,  $I_P$ , within the range specified.  $C_F$  is recommended for noise management, with values that depend on the application.

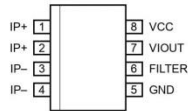
# ACS712

*Fully Integrated, Hall Effect-Based Linear Current Sensor IC with 2.1 kVRMS Isolation and a Low-Resistance Current Conductor*

Functional Block Diagram



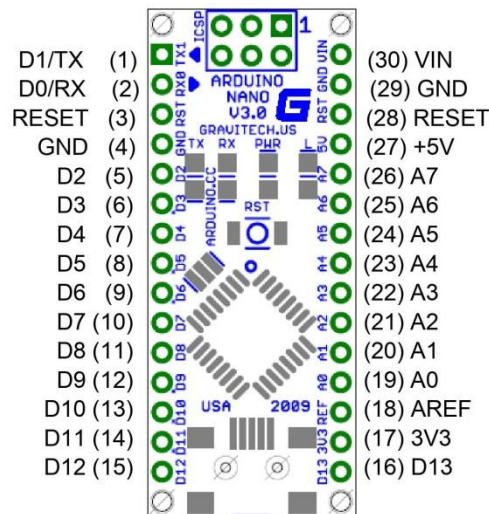
Pin-out Diagram



Terminal List Table

Number	Name	Description
1 and 2	IP+	Terminals for current being sampled; fused internally
3 and 4	IP-	Terminals for current being sampled; fused internally
5	GND	Signal ground terminal
6	FILTER	Terminal for external capacitor that sets bandwidth
7	VIOUT	Analog output signal
8	VCC	Device power supply terminal

### Arduino Nano Pin Layout



Pin No.	Name	Type	Description
1-2, 5-16	D0-D13	I/O	Digital input/output port 0 to 13
3, 28	RESET	Input	Reset (active low)
4, 29	GND	PWR	Supply ground
17	3V3	Output	+3.3V output (from FTDI)
18	AREF	Input	ADC reference
19-26	A0-A7	Input	Analog input channel 0 to 7
27	+5V	Output or Input	+5V output (from on-board regulator) or +5V (input from external power supply)
30	VIN	PWR	Supply voltage

SavantIC Semiconductor

Product Specification

**Silicon PNP Power Transistors****2SA715****DESCRIPTION**

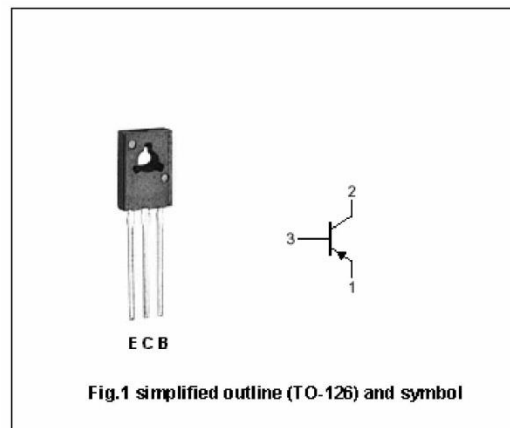
- With TO-126 package
- Complement to type 2SC1162

**APPLICATIONS**

- Low frequency power amplifier applications

**PINNING**

PIN	DESCRIPTION
1	Emitter
2	Collector;connected to mounting base
3	Base

**Absolute Maximun Ratings (Ta=25℃)**

SYMBOL	PARAMETER	CONDITIONS	VALUE	UNIT
$V_{CBO}$	Collector-base voltage	Open emitter	-35	V
$V_{CEO}$	Collector-emitter voltage	Open base	-35	V
$V_{EBO}$	Emitter-base voltage	Open collector	-5	V
$I_C$	Collector current (DC)		-2.5	A
$I_{CM}$	Collector current-Peak		-3	A
$P_C$	Collector power dissipation	$T_a=25^\circ\text{C}$	0.75	W
		$T_C=25^\circ\text{C}$	10	
$T_j$	Junction temperature		150	℃
$T_{stg}$	Storage temperature		-55~150	℃

SavantIC Semiconductor

Product Specification

**Silicon PNP Power Transistors****2SA715****CHARACTERISTICS**T<sub>j</sub>=25°C unless otherwise specified

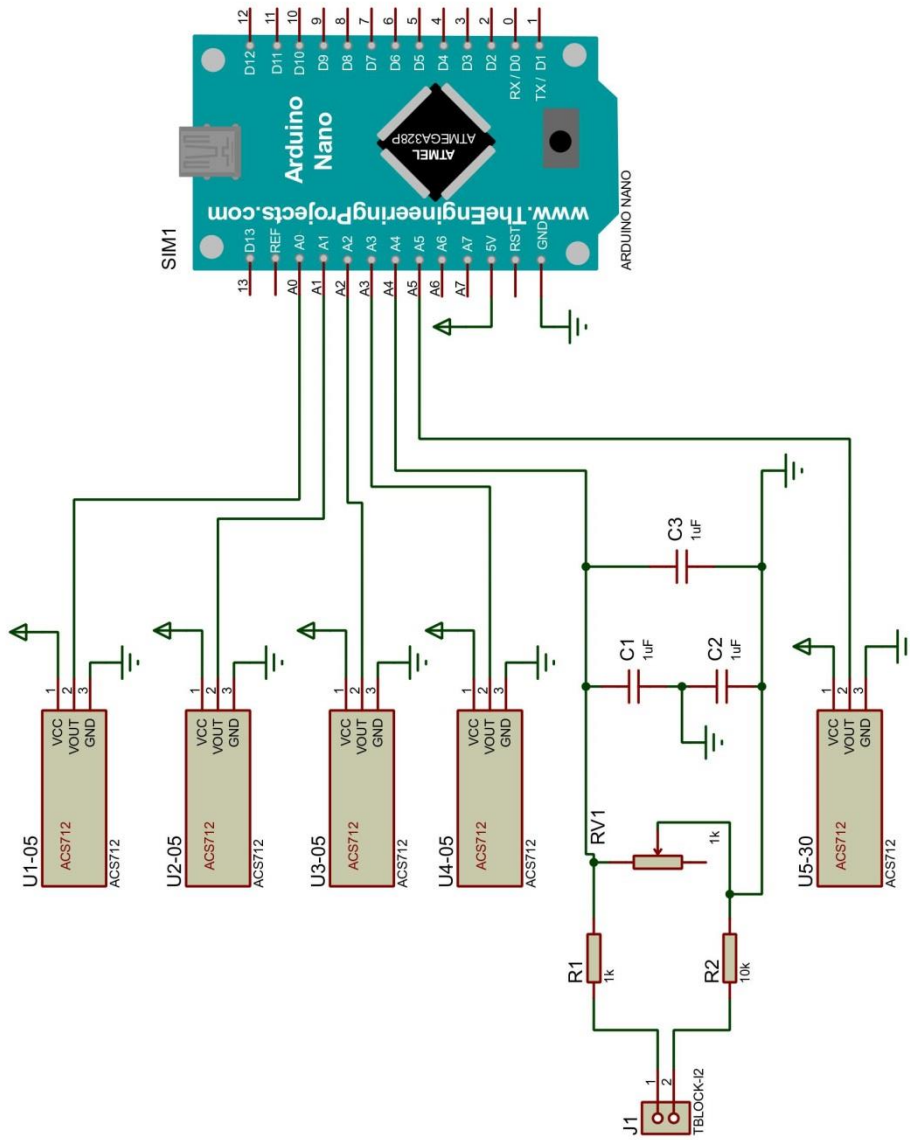
SYMBOL	PARAMETER	CONDITIONS	MIN	TYP.	MAX	UNIT
V <sub>(BR)CEO</sub>	Collector-emitter breakdown voltage	I <sub>C</sub> =-10mA; R <sub>BE</sub> =∞	-35			V
V <sub>(BR)CBO</sub>	Collector-base breakdown voltage	I <sub>C</sub> =-1mA; I <sub>E</sub> =0	-35			V
V <sub>(BR)EBO</sub>	Emitter-base breakdown voltage	I <sub>E</sub> =-1mA; I <sub>C</sub> =0	-5			V
V <sub>CEsat</sub>	Collector-emitter saturation voltage	I <sub>C</sub> =-2.0A; I <sub>B</sub> =-0.2A(Pulse test)		-0.5	-1.0	V
V <sub>BE</sub>	Base-emitter voltage	I <sub>C</sub> =-1.5A; V <sub>CE</sub> =-2V(Pulse test)		-1.0	-1.5	V
I <sub>CBO</sub>	Collector cut-off current	V <sub>CB</sub> =-35V; I <sub>E</sub> =0			-20	μA
h <sub>FE-1</sub>	DC current gain	I <sub>C</sub> =-0.5A; V <sub>CE</sub> =-2V	60		320	
h <sub>FE-2</sub>	DC current gain	I <sub>C</sub> =-1.5A; V <sub>CE</sub> =-2V(Pulse test)	20			
f <sub>T</sub>	Transition frequency	I <sub>C</sub> =-0.2A; V <sub>CE</sub> =-2V(Pulse test)		160		MHz

◆ h<sub>FE-1</sub> Classifications

B	C	D
60-120	100-200	160-320

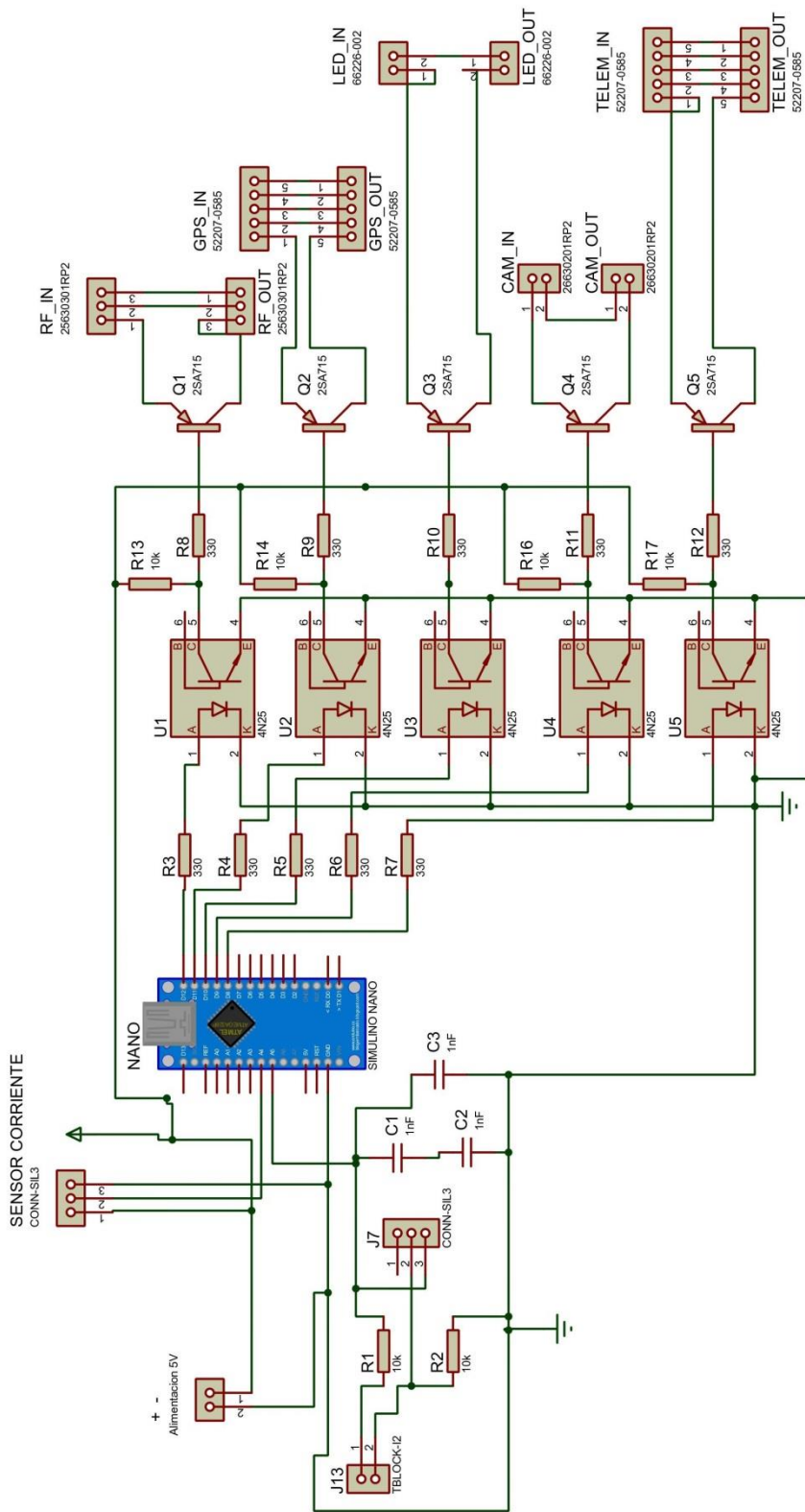


**ANEXO No. 14****DIAGRAMA CIRCUITAL – MEDICIÓN (PROTEUS)**



## **ANEXO No. 15**

### **DIAGRAMA CIRCUITAL – CONTROL Y OPTIMIZACIÓN (PROTEUS)**



**ANEXO No. 16**  
**TABLA COMPARATIVA DE COSTO DE DRONE**  
**CUADRICOPTERO DEL PROYECTO**

PROYECTO DE TESIS		FECHA DE INICIO			
SISTEMA DE CONTROL Y OPTIMIZACIÓN DE ENERGÍA		20 de octubre 2019			
PARA DRONES CUADRICOPTEROS DURANTE VUELO		06 de agosto 2019			
ESTACIONARIO		Alex Ramirez G			
		Proyecto de titulación			
CUADRO COMPARATIVO DE COSTO		Adquisición de kit drone cuadricóptero			
PTJE	PROVEEDOR 1	PROVEEDOR 2	PROVEEDOR 3		
REF	AMAZON	ALIEXPRESS	MERC. LIBRE	PTJE	PTJE
<b>CARACTERÍSTICAS TÉCNICAS</b>					
1 x HJ 450 F de fibra de nylon FlameWheel marco Airframe Kit HJ450					
2 x d2212 920 KV clockwise girelo CW motor sin escobillas					
2 x d2212 920 KV clockwise girelo CCW Motor sin escobillas					
4 x 30 A Brushless ESC					
2 x CW CCW Propeller 9443					
1 x APM2.8 APM 2.8 Multicopter vuelo driver 2.5 2.6 Upgraded integrado Brújula					
1 x 6 m GPS con brújula I5883 25 cm cable					
1 x 140 mm alta Landing Gear Skid ruedas trípode Negro					
5 x 10 cm Servo Extensión plomo alambre Cable macho a macho KK MK MWC Junta de Control de Vuelo para RC Quadcopter	40	CUMPLE (nuevo)	CUMPLE (usado)	40	40
1 x DJI GPS Antena plegable soporte metálico					
12 x 3.5 mm Bullet conector (Conector plátano)					
1 x Hook & Loop Cierre Cinta					
1 x T Plug Conector macho alambre de silicona con 11,5 cm 14 AWG					
1 x IMAX RC B3 Pro Cargador de Balance compacto					
1 x 11.1 V 2200 mAh 30 C 3S1P batería					
1 x Flysky FS-16 6 CH 2.4 G AFHDS 2 A LCD Transmisor Receptor iaf6 Modo 2					
1 x Fibra de Carbono cardán FPV PTZ de amortiguación de montaje para DJI Phantom Quadcopter					
PRECIO SIN IVA	40	256,95	40,00	38,55	37,3
FORMA DE PAGO	10	100%	10	100%	10
TIEMPO DE ENTREGA	10	23 días	8	60 días	10
<b>TOTAL</b>	<b>100</b>		98	<b>93,55</b>	<b>97,3</b>
<b>OBSERVACIONES:</b> Se elige la primera opción de compra en AMAZON, puesto que obtiene el mayor puntaje 98% en los parámetros considerados, el equipo es nuevo y proveedor proporciona garantía de compra					