# UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

PROGRAMA DE MAESTRÍA Y DOCTORADO EN INGENIERÍA
INGENIERÍA ELÉCTRICA-SISTEMAS ELECTRÓNICOS

## “IMPLEMENTACIÓN DE ALGORITMO DE CONTROL EN UN SISTEMA EMBEBIDO PARA SEGUIMIENTO AUTÓNOMO EN UN CUADRICÓPTERO”

**T E S I S**

QUE PARA OPTAR POR EL GRADO DE**:**

MAESTRO EN INGENIERÍA

PRESENTA**:**
ING. IRVING I. MARTÍNEZ GARCÍA

TUTOR PRINCIPAL:
DR. J. MARIO PEÑA CABRERA, IIMAS-UNAM

**CIUDAD UNIVERSITARIA, CD.MX., 2016**

**JURADO ASIGNADO**

Presidente: Dr. Martínez López José Ismael

Secretario: Dra. Velasco Herrera Graciela

Vocal: Dr. Peña Cabrera Juan Mario

1er Suplente: Dr. Prado Molina Jorge

2do Suplente: M.I. Alvares Castillo Jesús

México Ciudad de México, Diciembre de 2016
Ciudad Universitaria
Facultad de Ingeniería

**Tutor de Tesis:**
Dr. Peña Cabrera Juan Mario

_____

**Firma**

Este trabajo fue realizado en el IIMAS UNAM, bajo la dirección del Dr. Juan Mario Peña Cabrera, con apoyo de Conacyt, con la beca No. 399784.

*Agradezco a Dios y a cada una de las personas que me ayudaron para realizar mis estudios de maestría.*

**Índice de figuras**

**Índice de tablas**

**Nomenclatura**

# Índice

# Capítulo 1- Generalidades.



# Capítulo 2- Estado del arte.



# Capítulo 3- Marco teórico.

# Capítulo 4- Descripción de los elementos del cuadricóptero.



# Capítulo 5- Implementación en hardware y pruebas.

# Capítulo 6- Resultados y Conclusiones.

## Índice de figuras

## Índice de tablas

**Nomenclatura**

A: Ampers
FOSS: Free and open source software
CES: Control electronic speed
BEC: Battery eliminator circuit
UBEC: Ultimate battery eliminator circuit.
UAV: Unmanned Aerial Vehicle
UAS: Unmanned Aerial System
RF: Radio frecuencia
GPS: Global Positioning Systems
App: Application
VANT: Vehicula aéreo no tripulado
iOS: i Operative System
3D: Tres dimensiones
WiFi: Proveniente de la marca Wi-Fi Alliance
MP: Mega pixeles
IMU: Inertial Measurement Unit
CW: Clockwise
CCW: Counterclockwise
LED: Light emitting diode
CD: Corriente directa
CC: Corriente continua
CA: Corriente alterna
V: Volts
W: Watts
RPM: Revoluciones por minuto
mAh: Miliampers hora
Hz: Hertz
PCB: Printed Circuit Board
HDL: Hardware Description Language
AVR: Advanced RISC Virtual
EEPROM: Electrically Erasable Programmable Read-Only Memory
DMP: Digital Motion Processor
I²C: Inter-Integrated Circuit
SDA: System Data Array
SCL: System Clock line
GND: Ground
Mbps: Mega bits por segundo
dBW: Dedicated band width
SSL: Secure socket layer
SO: Sistema operativo
IDE: Integrated development environment

# *CAPÍTULO 1*

## 1.1. Introducción

En esta tesis se aborda el trabajo que lleva por nombre “Implementación de algoritmo de control en un sistema embebido para seguimiento autónomo en un cuadricóptero”, la presente tesis se divide en 6 capítulos: 1) Generalidades, 2) Estado del Arte, 3) Marco teórico, 4) Descripción de los elementos del cuadricóptero, 5) Implementación en Hardware y Pruebas y 6) Resultados y Conclusiones.

El término inglés Drone (zángano), se designa a diversos tipos de vehículos aéreos no tripulados, al principio este término se refería a aparatos de uso militar y con aspecto al de un avión. La denominación Vehículo Aéreo no Tripulado (VANT), proviene del inglés Unmanned Aerial Vehicle (UAV) [1].

Para distinguir los VANT de los misiles, se define como un vehículo sin tripulación reutilizable, capaz de mantener un nivel de vuelo controlado y sostenido, y propulsado por un motor de explosión o de reacción.

Los VANT se pueden clasificar por formas, tamaños, configuraciones y características en el diseño de estos. A lo largo de su historia los vehículos aéreos no tripulados eran aviones pilotados remotamente, pero cada vez más se está empleando el control autónomo de estos [1]. Se han creado dos variantes como es 1) controlados desde una ubicación remota con utilización de control RF, y 2) volar de forma autónoma sobre la base de planes.

El cuadricóptero es un sistema constituido de una estructura mecánica componentes electrónicos y software, utiliza cuatro motores en sus extremidades, cuenta con dos pares de hélices de paso fijo con rotación contraria, teniendo los rotores 1 y 3 con giro anti horario y los rotores 2 y 4 con giro en sentido horario, esto cancela casi en su totalidad el efecto giroscópico presentado en los helicópteros comunes, en el cuadricóptero es utilizado para cambiar su orientación.

De manera general los UAV se usan  comúnmente como dispositivos de uso militar, por otro lado las aeronaves de RC se encuentran en gran parte aplicadas en el uso civil, en el ámbito de la observación los vehículos como los UAV y las aeronaves de RC tienen múltiples aplicaciones y posibilidades en el mercado civil algunas de estas son el monitoreo del estado de la atmósfera, gestión de cultivos, seguridad y monitoreo, captura de imágenes y video, búsqueda de víctimas en zonas de desastres naturales, control de obras y evaluación de su impacto, distribución de señal gratuita de internet, entrega de productos o mensajería, seguimiento de las áreas boscosas, control de incendios, etc.

La idea de volar por el ser humano se remonta a épocas muy primitivas pero en si el origen de la idea que concierne a este tema es la de mantener un objeto en el aire debido a una fuerza de empuje generada por la rotación de unas hélices esta se remonta hasta la antigua cultura china aproximadamente en el año 400 A.C., ese artefacto se basaba en un palillo con un tipo de hélice en el extremo superior, colocada perpendicularmente; este se hacía girar con las manos y era liberado.

En el siglo XV Leonardo Da Vinci diseñó diversos mecanismos planeadores, basándose en las extremidades propulsoras de las aves, realizó bocetos de estos dispositivos sin llegar nunca a construirlos [2].

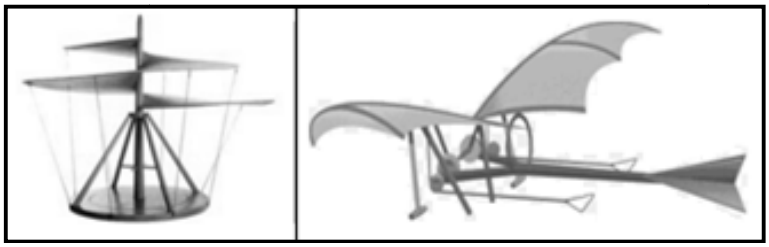


Figura 1.-  Maquinas voladoras de Leonardo Da Vinci.

Los primeros precursores de la aviación no tripulada se remontan al siglo XIX, cuando los norteamericanos Elmer Sperry junto a Lawrence y Peter Cooper Hewitt, establecieron las bases, enfocándose a las aéreas de radio control y guiado inercial [3].

En 1917 Charles Kettering, de General Motors, desarrolla un biplano no tripulado pre-programado conocido como 'torpedo aéreo Kettering'. Este vehículo accionado por un mecanismo de relojería debería plegar las alas en un lugar programado y caer sobre un enemigo como una bomba [4].

La fecha formalmente de el primer vuelo con éxito de una aeronave no tripulada es el 6 de Marza de 1918 esto en Copiague, Long Island con el lanzamiento del torpedo aéreo Curtiss-Sperry, aunque los intentos fracasaron [3].

Investigaciones para construir una aeronave no tripulada se llevaban a cabo en la Unión Soviética, de 1930-1940 el diseñador de aviones Nikitin desarrolló un drone planeador armado con torpedo PSN-1 y 2 de tipo de 'ala volante' en dos modalidades: una como blanco aéreo para entrenamiento de pilotos y otra con automatización completa. En1933 se realizó la primera prueba exitosa en el Reino Unido del primer UAV Queen Bee, desarrollado a partir del biplano Fairey Queen, este se operaba por control remoto desde un barco [4]

Hasta la segunda guerra mundial (1939-1945) que se comenzó a realizar un empleo efectivo  con blancos aéreos para validar la eficacia de los cañones anti aéreos. A raíz de esto y con el paso de los años poco a poco paso de ser un blanco a su aplicación principal [3].

El desarrollo de vehículos aéreos no tripulados se intensificó inmediatamente después de la Segunda Guerra Mundial. Para el año de 1940  se produce el primer drone en serie a gran escala, fue el estadounidense Radioplane OQ-2, que sirvió como blanco volante para la formación de pilotos. Además, en 1944, fue utilizado por primera vez en el mundo el UAV 'clásico' de ataque el Interstate TDR. A principios de los años 70 la Oficina de diseños Tupolev desarrolló varios drones de gran alcance para misiones de reconocimiento, los llamados Tu-123 Yástreb, Tu-141 Strizh y Tu-143 Reis [5].

El desarrollo de vehículos aéreos no tripulados continuó con algunos comandantes militares que cuestionaban su utilidad. La actitud hacia los UAV cambió con la victoria de la Fuerza Aérea israelí sobre la Fuerza Aérea Siria con el uso coordinado de éstos en 1982. Los Drones israelíes fueron utilizados como señuelos electrónicos, bloqueadores electrónicos y vigilancia en tiempo real. Se podría argumentar que esta campaña marcó el comienzo de la era moderna de los UAV. Para 1994  se realizó el primer vuelo de un Predator, se trata del primer UAV operativo que usa el sistema de posicionamiento global GPS en lugar de estar programado o de ser manipulado por control RF [5].

En los últimos 15 años se han desarrollado sistemas aéreos no tripulados que pueden realizar un seguimiento autónomo como lo es Zano, Hexo +, lily camera, etc.

## 1.2. Objetivos

Desarrollar un algoritmo para realizar el seguimiento de manera autónoma a través de un cuadricóptero, implementando dicho algoritmo en un sistema embebido.

- Generar un algoritmo para que el cuadricóptero Phamtom Q450 realice un seguimiento autónomo.
- Implementar en un sistema embebido (Arduino) para que el cuadricóptero elegido realice un seguimiento autónomo.
- Mantener un rango de operación de hasta 20 metros de distancia a la redonda entre el dispositivo móvil y el cuadricóptero.
- Mantener un intercambio de información con el cuadricóptero, a través de comunicación inalámbrica, por medio de la tecnología Bluetooth.

## 1.3. Justificación

Los dispositivos, en específico los cuadricópteros con característica de seguimiento autónomo, han retomado un interés en los últimos quince años, no solo por el uso en el ámbito militar, sino ahora para aplicaciones civiles donde tienen interacción con las personas, la mejora de la facilidad de manejo de este tipo de vehículos es de gran importancia para que dicha interacción hombre–cuadricóptero sea más sencilla.

Se pretende generar y realizar la implementación de un algoritmo en un cuadricóptero (Phamtom Q450) para que posea la característica de seguimiento autónomo haciendo uso de tecnología Open Source.

## 1.4. Metodología

- Estudios exploratorios de sistemas relacionados, recopilación de información de dispositivos y temas afines recientes.
- Estudiar y probar las tarjetas de desarrollo Arduino Uno y Arduino Micro
- Estudiar algoritmo de vuelo de cuadrorotor implementado en Arduino Uno.
- Estudiar y generar algoritmo para generar la comunicación inalámbrica entre el cuadrorotor y el dispositivo móvil.
- Estudiar y generar algoritmo para obtener posicionamiento con GPS.
- Modificar algoritmo de vuelo de cuadrorotor implementado en Arduino Uno.
- Diseño de algoritmo para el seguimiento autónomo en un sistema embebido (Arduino).
- Codificar e implementar los algoritmos propuestos en lenguaje C para la placa Arduino Micro.
- Pruebas de funcionamiento del cuadricóptero.

## 1.5. Alcances

- Dotar al cuadricóptero con capacidad de señalización de eventos y toma de decisiones basado en sus registros de lecturas.
- Generar el algoritmo para que el sistema aéreo pueda seguir un objetivo de manera autónoma inalámbricamente.
- Implementar el algoritmo para la tarjeta de desarrollo Arduino Micro.
- Desarrollar una App para que pueda ser instruido a través de un dispositivo móvil con sistema operativo Android.
- Pruebas de operación del cuadricóptero.

# *CAPÍTULO 2*

## 2.1. Estado del arte

De acuerdo con el trabajo, en el estado del arte se presentan algunas de las aeronaves similares, con capacidad de seguimiento, así como trabajos referentes al estudio del diseño de controladores de cuadrorotores y algunos trabajos con enfoque práctico de control por visión.

En 2010 Ivan Reed Man junto con un equipo de trabajo desarrollo un sistema con seguimiento llamado Zano, es un mini cuadricóptero, enfocado a la fotografía aérea personal como plataforma de captura de vídeo, es muy pequeño cabe en la palma de la mano y es inteligente para poder volar por sí mismo, Zano se conecta a un dispositivo iOS o Android [6].

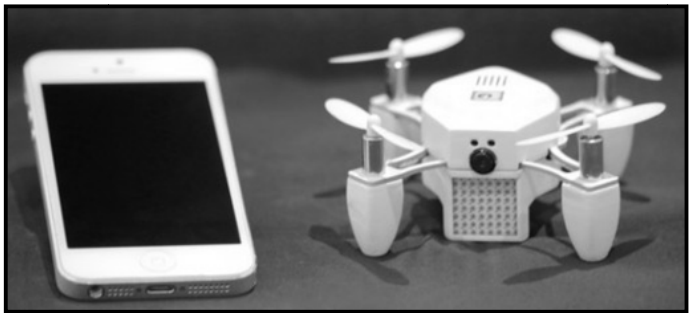


Figura 2.- Sistema Zano.

Antoine Balaresque y Henry desarrollaran Lily camera, esta es una aeronave que está dotada con seguimiento, esta se activa simplemente con encenderlo y aventarlo ligeramente frente al usuario, después de esto se puede dejar solo, el dispositivo hará el seguimiento en automático. Su función principal es simular a una cámara de acción, con una distancia de operación máxima de 15 m, está construida con una carcasa de policarbonato que es capaz de soportar los choques, también es resistente al agua, puede volar por 20 minutos, el dispositivo cuenta con un mando a distancia, que sirve como un faro de ubicación [7].


Figura 3.- Sistema Lily camera.

Otra de las aeronave con característica de seguimiento aunque de mayor tamaño es Hexo +, este es un hexacóptero equipado con una cámara, diseñado para grabar desde las alturas. Es capaz de seguir de forma completamente autónoma el proyecto comenzó enKickstarter. Todo el proceso de configuración del encuadre y la programación del Drone se realiza a través del teléfono móvil, empleando una aplicación específica, una vez hecho esto el propio Drone se encarga de lo demás de forma autónoma, incluidos el despegue y el aterrizaje. Puede mantenerse a una distancia de hasta 50 metros de nosotros y se desplaza a una velocidad máxima de 70 Km/h [8].

Figura 4.- Hexacoptero Hexo +.

Dentro de la información mencionada se ve que en la actualidad ya existen diversas aeronaves que poseen la característica del seguimiento autónomo, dentro de la información que se buscó durante la documentación para la realización de este trabajo no se pudo tener la posibilidad de ver de qué manera era en que se realizaba el seguimiento de dichos dispositivos, si era por GPS, por RF o bien por sensores, ya que por el momento la información es muy limitada por los creadores de estos equipos. Así también mencionar que los trabajos relacionados con cuadricópteros, se enfocan mayormente al control de vuelo, abordando los diferentes tipos o maneras en que se puede realizar dicho control.

En el trabajo de Marcelo Teixeira, Ronaldo, Matheus, y Veronica [9] se aborda el control de un dron a través de teleoperación usando los lentes google. Este artículo propone el uso de un dispositivo que se usa para la visualización y control en asociación con un UAV aplicado a la inspección estructural de edificios. Nos muestra diferentes trabajos relacionados en los cuales se utiliza la teleoperación y la manera en que podría realizarse la conexión de los dispositivos, así también la manera en que se hizo esta, explica la manera en que se efectuaron las pruebas así como los inconvenientes que se tuvieron y propone mejoraras para una siguiente versión.

El trabajo [10] menciona el control a través del uso de visión, en este trabajo se parte de la idea de que un inconveniente de los vehículos aéreos micro es su carga útil y su tamaño partiendo de esto se explica por qué incorporar la visión por computadora en estos. Menciona que un gran problema que tienen estas aeronaves es la de evitar las colisiones para esto se hace el uso de la visión por computadora, también menciona el proceso que lleva a cabo para la realización de la navegación a través de la visión por computadora y la realización de un algoritmo.

En el artículo [11] se aborda el control a través de la técnica denominada Backstepping o de retroceso, en este artículo se presenta un modelo dinámico no lineal para un cuadrorotor, nos menciona que debido a la propiedad de bajo accionamiento del cuadrorotor, el regulador puede fijar la trayectoria del helicóptero en tres direcciones (x, y, z) y el ángulo de guiñada a sus valores deseados y estabilizar los ángulos, se presenta un control por Backstepping para estabilizar todo el sistema y la metodología de diseño que fue basada en la teoría de estabilidad de Lyapunov.

Otra de las maneras de realizar el control es a través de modos deslizantes o Sliding Mode [12-14], en [12] se presenta un nuevo método de diseño para el control de vuelo de un vehículo cuadrorotor, de acuerdo a la propiedad de accionamiento del cuadrorotor, menciona al igual que el anterior que el control puede hacer que el cuadrorotor se mueva tres posiciones (x, y, z) y el ángulo de guiñada a sus valores deseados y estabilice los otros dos ángulos, propone un control de modo deslizante en cascada para estabilizar un sistema de bajo accionamiento y se presenta el análisis de estabilidad global del sistema de lazo cerrado.

En el artículo [15], se describe principalmente el desarrollo de un sistema de control no lineal de vehículos basado en las ecuaciones de Riccati dependientes del estado, el controlador se integra en un concepto de sistema de misión global para UAVs.

En el artículo [16] el autor propone un nuevo controlador no lineal para un vehículo aéreo no tripulado cuadrorotor utilizando redes neuronales (NNs) y la retroalimentación de salida, introduce estas redes para aprender la dinámica completa del UAV en línea, incluyendo términos no lineales inciertos como la fricción aerodinámica y el giro de las hélices, menciona que a pesar de que estos sistemas son de baja respuesta, propone un esquema de entrada de control virtual con redes neuronales que permite controlar los seis grados de libertad del UAV usando sólo cuatro entradas de control. Muestra usando la teoría de Lyapunov que los errores de seguimiento de posición, orientación y velocidad, de estimación de control virtual y de observador, y los de estimación de peso de NN para cada red son globalmente todos uniformes y limitados.

En [17] se desarrolla un controlador a través de adaptación difusa para el mando de vuelo de un cuadrorotor. El controlador PID difuso se implementa mediante la simulación Hardware-In-the-Loop, define reglas así como el rango de funciones de pertenencia para la realización del controlador.

En el artículo [18] el autor propone un método de control, en primer lugar deriva el modelo del cuadrorotor utilizando la ecuación de Euler-Lagrange y realizar experimentos para identificar el parámetro del modelo, después divide el sistema cuadrorotor en dos subsistemas 1) el sistema de orientacion y 2) el sistema de altitud. Para el control de la primera, utiliza el método de control PID, y para el control de la altitud, utiliza el método de control dinámico de superficie (DSC). De la teoría de la estabilidad de Lyapunov, prueba que todas las señales de un sistema del cuadrorotor son uniformemente acotadas (UUB).

En el artículo [19] proponen un sistema de control PID no lineal, se plantean dos comandos uno para seguimiento en orientación y un comando de seguimiento de posición. Los sistemas de control los desarrollan directamente en el grupo Euclideano especial para evitar singularidades de representaciones de orientacion mínimas, además propone una nueva forma de términos de control integral para garantizar una estabilidad asintótica casi global cuando existen incertidumbres en la dinámica del cuadrorotor.

En [20] se presentan dos técnicas lineales completas basadas en la teoría del control óptimo. El primero está basado en la norma L2, es un controlador servo cuadrático Lineal (LQServo), y el segundo, optimiza la norma L∞ y se diseña utilizando el método de control H∞, esto a causa de las perturbaciones externas y particularmente las ráfagas de viento, al final realiza una comparación entre los dos controladores.

# *CAPÍTULO 3*

## 3.1. Revicion de ecuaciones de movimiento de un cuadricóptero

Se usa un sistema de coordenadas fijo al cuerpo con origen en el centro de masa del cuadricóptero. Los ejes $x_B$ y $y_B$ entan en linea los brazos del cuadricóptero y son ortogonales entre si y el eje $z_B$ es perpendicular al plano formado por $x_B$ y $y_B$ los ejes fijos a la tierra están definidos por $x_w$, $y_w$ y $z_w$, las fuerzas de sustentación se representan por los vectores $\mathbf{F_i}$, $\mathbf{F_2}$ $\mathbf{F_3}$ y $\mathbf{F_4}$, cada motor produce igualmente un momento perpendicular $\mathbf{M_1}$, $\mathbf{M_2}$, $\mathbf{M_3}$ y $\mathbf{M_4}$ al plano de rotación de la hélice. Los momentos producidos $\mathbf{M_1}$ y $\mathbf{M_3}$ actúan en la dirección z mientras que $\mathbf{M_2}$ y $\mathbf{M_4}$ rotan en la dirección contraria (-z) [21].

La figura 5 describe el modelo del cuadricóptero que se usa para el análisis de éste.

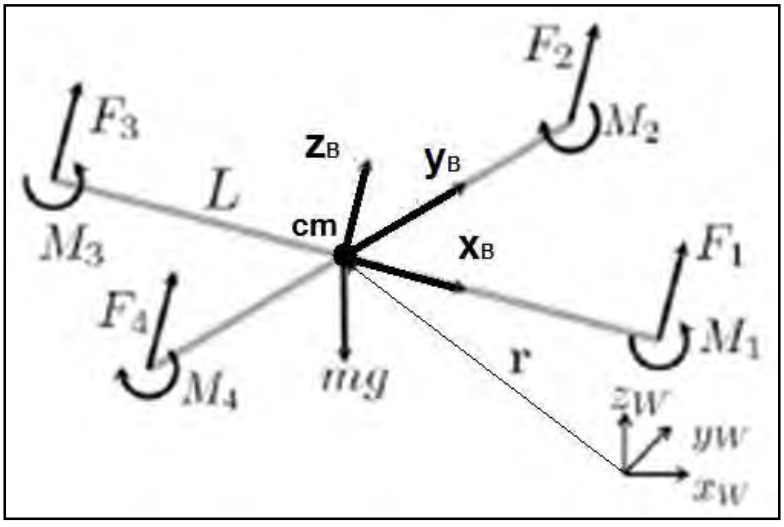



Figura 5.- Sistema de coordenadas fuerzas y momentos en el cuadricóptero [21].

La ecuación que gobierna la aceleración del centro de masa considerando la gravedad mostrada en la figura 5 es la que se muestra a continuación.

$$\vec{F} = m\ddot{a} = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + A \begin{bmatrix} 0 \\ 0 \\ \sum F_i \end{bmatrix} \quad (1)$$

Donde:

$\ddot{a}$ = Aceleración del centro de masa

$F_i$ = Fuezas de sustentación

A = Constante

Las componentes de velocidad angular fijas al cuadricóptero son *p, q* y *r,* estos valores están relacionados a la derivada de los ángulos de rotación (roll), cabeceo (pitch) y guiñada (yaw) de acuerdo a:

$$\bar{\boldsymbol{\omega}} = \begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} cos\theta & 0 & -cos\phi sen\theta \\ 0 & 1 & sen\phi \\ sen\theta & 0 & cos\phi cos\theta \end{bmatrix} \begin{bmatrix} \phi \\ \theta \\ \Psi \end{bmatrix} \quad (2)$$

### 3.2 Modelo del motor

Cada motor tiene una velocidad angular $\boldsymbol{\omega_i}$ y produce una fuerza $\mathbf{F_i}$ de acuerdo a:

$$\overrightarrow{F_i} = k_F \omega_i^2 \quad (3)$$

Donde:

$\overrightarrow{F_i}$ = Fuerza de sustentación [*N*].

ω = Velocidad angular [*rpm*].

$k_F$= Constante de fuerza de motor $\left[\frac{N}{rpm}\right]$.

Los motores también producen un momento de acuerdo a:

$$M_i = k_M \omega_i^2 \qquad (4)$$

Donde:

$M_i$ = Momento de fuerza [$Nm$].

$k_M$ = Constante $\left[\frac{N}{rpm}\right]$.

Considerando que los ángulos de rotación y cabeceo son pequeños (aprox.=0) entonces cos ϕ ≈ 1, cos θ ≈ 1, sen ϕ ≈ ϕ, sen θ ≈ θ. Si tomamos encuenta que la suma de las fuerzas de sustentación en los cuatro motores para un vuelo de suspensión debe ser proporcional a la fuerza de gravedad que actúa en el cuadricóptero y de forma ideal las constantes $k_F$ son iguales se debe satisfacer.

$$mg = \sum_{i=1}^{4} k_F \omega_h^2 \qquad (5)$$

Donde:

$k_F$= Constante del motor (masa).

$\omega_h$= Velocidad del motor.

Y las velocidades de los motores están dadas por

$$\omega_h = \sqrt{\frac{mg}{4k_F}} \qquad (6)$$

En resumen variando la velocidad angular de cada par de rotores, se puede cambiar la fuerza de empuje y se puede generar de la misma manera un movimiento vertical, frontal o lateral. El movimiento de cabeceo (pitch) es generado por una diferencia de velocidades entre los rotores 1 y 3, El movimiento de alabe (roll) se genera por la diferencia de velocidades entre los rotores 2 y 4, tomando en cuenta el efecto giroscópico, el movimiento de guiño (yaw) es el que resulta de la diferencia de velocidades entre el par de rotores que giran en sentido horario y el par que giran en sentido anti horario.

La fuerza de empuje total es igual a la suma de las fuerzas de empuje de cada una de los cuatro rotores

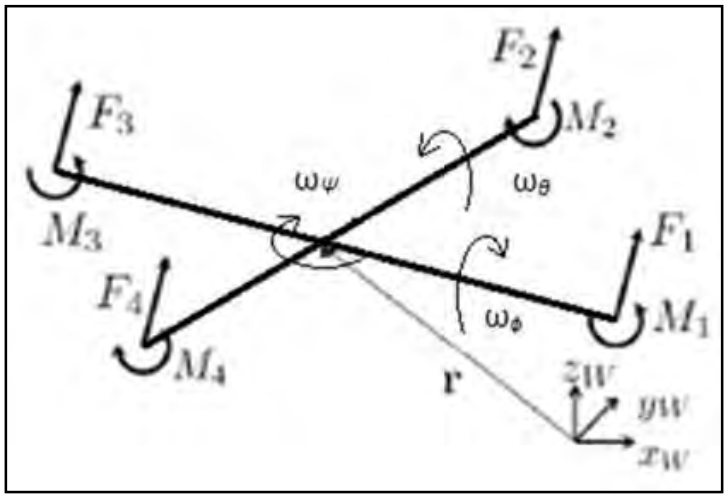



Figura 6.- Sistema de movimientos del cuadricóptero.

# *CAPÍTULO 4*

## 4.1. Descripción de los elementos

En este capítulo se describen los elementos para implementación del cuadricóptero de acuerdo a la figura 7, la cual muestra las partes que lo conforman. Como se muestra en la siguiente figura la aeronave está conformado por sensores como lo es un giróscopo, un GPS y un módulo Bluetooth, los cuales le proporcionaran la capacidad de adquisición e intercambio de información,

De la misma manera cuenta con dos placas de desarrollo la cuales contienen un microcontrolador, ATmega328P, ATmega32UP4, dentro de estos microcontroladores montados en las placas Arduino Uno y Micro, se implemento un código, dentro de la placa Arduino Uno se uso la parte del control de vuelo, la cual cabe mencionar que no se desarrollo, se tomo como base un código ya desarrollado previamente para el vuelo del cuadricóptero, este se adaptó y se hicieron las modificaciones pertinentes para su utilización, de la misma forma en la placa Arduino micro se implementó el algoritmo desarrollado para el seguimiento, este fue realizado en su totalidad, el cuadricóptero contiene actuadores (motores) que serán accionados y gobernados para su funcionamiento adecuado por el trabajo en conjunto de las partes anteriormente mencionadas, por último el cuadricóptero se apoya de un dispositivo móvil para que haga el seguimiento, este dispositivo contiene una aplicación la cual está conectada vía Bluetooth.

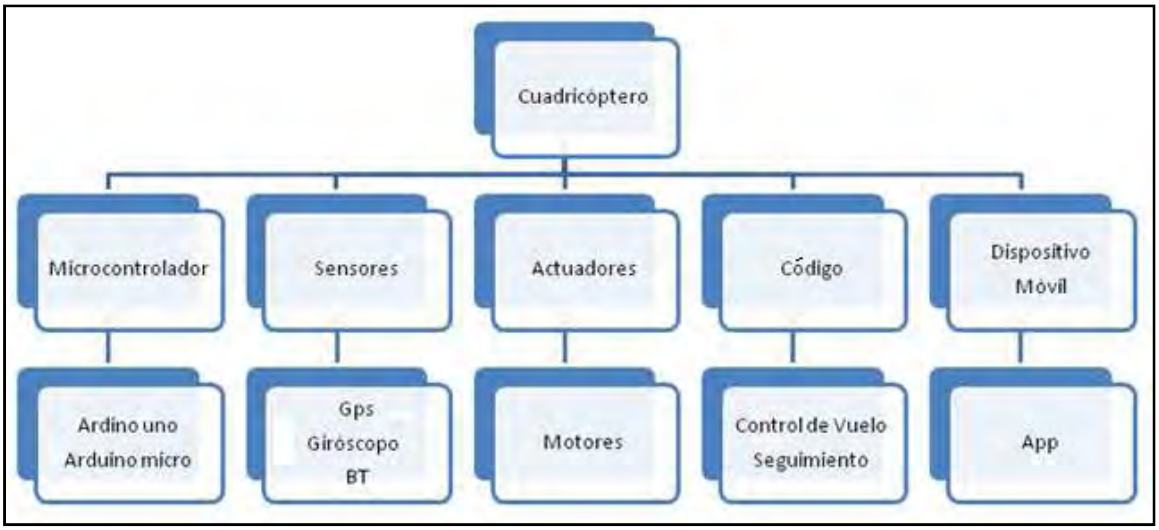



Figura 7.- Elementos que conforman al cuadricóptero.

En primer lugar, se comienzo por 4.2.2., el frame, también llamado armazón o marco, este es el soporte físico donde van todos los componentes, en el caso de un cuadricóptero tiene la peculiaridad de ser una cruz o equis y en cada uno de los extremos va anclado un motor. Posteriormente se determinó qué tipo de 4.2.3., motor, utilizaría y cómo iba a ser controlado. De tal manera que para este trabajo se optó por emplear motores brushless (sin escobillas). Cada uno de ellos es controlado por un 4.2.5., ESC (Electronic Speed Controller), también llamado variador de velocidad. Es imprescindible mencionar que el código está contenido dentro de dos 4.2.7., placas Arduino. La primera de estas se encuentra a cargo de mandar señales sobre los ESCs cuya función será mover los motores, y la segunda se encargará de la recepción y envió de las señales. De igual manera se encuentra una 4.2.8., IMU (Inertial Measurement Unit), situada en el microcontrolador (que consta de un acelerómetro y un giroscopio de tres ejes). Para realizar la conexión entre el microcontrolador y el dispositivo móvil, se cuenta con 4.2.9., un módulo Bluetooth. Las 4.2.4., hélices unidas a los motores crearan la propulsión necesaria para que este despegue y mantenga el vuelo. Mientras tanto el seguimiento está a cargo del 4.2.10., GPS el cual indicara en que poción se encuentra el cuadricóptero y el dispositivo a seguir.

La lista por de los elementos adquiridos  para la implementación del Hardware del cuadricóptero queda de la siguiente manera:

● Chasis. (4.2.2.)

● 4 Motores sin escobillas. (4.2.3.)

● 4 Hélices. (4.2.4.)

● 4 ESCs (Electronic Speed Controller). (4.2.5.)

● Batería Lipo. (4.2.6.)

● 2 Placas Arduino. (4.2.7.)

● IMU (Inertial Measurement Unit). (4.2.8.)

● Modulo Bluetooth. (4.2.9.)

● GPS. (4.2.10.)

● Elementos electrónicos complementarios (reguladores de voltaje, resistencias, leds, conectores)
.

### 4.2.1 Configuración del cuadricóptero

Anteriormente se habló de la configuración de los  sistemas cuadricópteros, pudiendo ser  utilizada en distribución en “x” o en “+”, la configuración en “x” fue la que se optó pues esta tiene una mejor esteticidad y buena funcionalidad en estabilidad, los motores trabajan en pares dando más facilidad para su movilidad.

Para el vuelo se debe considerar el sentido de giro de cada hélice que es de vital importancia se hace girar dos de las hojas en sentido horario (CW) y otras dos de ellas en sentido contrario(CCW) como en la figura 8. Si todas girarán en las agujas del reloj, por ejemplo, el cuadricóptero comenzaría a girar en sentido inverso continuamente.

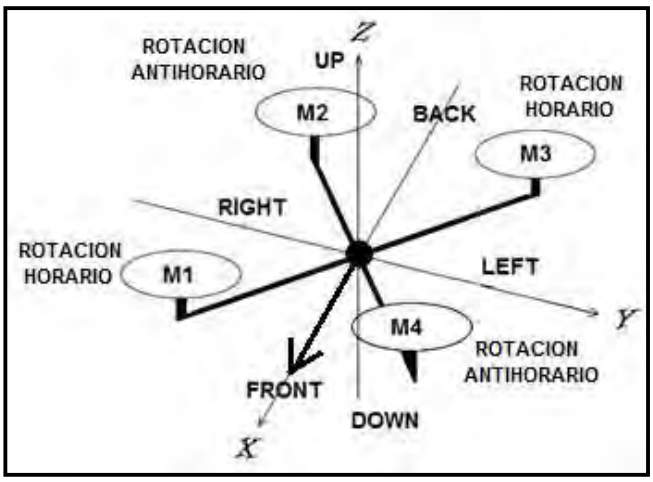



Figura 8.- Cuadricóptero configuración “X”.

### 4.2.2. Marco o chasis

Existen muchos tipos de formas y tamaños de marcos, lo siguiente era tomar en consideración las características que fueran de utilidad en este proyecto como el material, dimensión, y figura del mismo, sin embargo para tener la mejor elección ello existen criterios que deben ser tomados en cuenta, un ejemplo claro es el peso pues es evidente que un material de gran densidad no sería la mejor elección pues se aumentaría la fuerza así como el consumo de energía para lograr el mismo resultado de propulsión, al igual que si se elige un material demasiado ligero se corre el riesgo de que este sea demasiado frágil o débil. Sin embargo existen materiales que aun siendo ligeros son igual de fuertes que los primeros o incluso más. También se debe tener presente que una mayor flexibilidad del marco evitará posibles daños en su estructura en caso de caída. Por lo cual se eligió un marco con este tipo de características conformado de fibra de carbón ya que le proporciona flexibilidad, dureza y peso adecuado como lo es el Q450 Ghost Edition LED Night Quad-Copter Frame de la figura 9 este tiene por especificaciones un largo de 450 mm, una altura de 55 mm, un peso de 280 g. y cuenta con orificios para el montaje del motor de 16/19 mm algunas de las características de este modelo se listan a continuación:

Características de la estructura:

• Brazos con luces led
• Conectores en PCB integrados para soldadura directa
• Conexiones de latón pre roscados
• Pestañas grandes para facilitar el montaje de cámara
• Fácil ensamble

Figura 9.- Chasis Phamtom Q450.

### 4.2.3. Motor sin escobillas

La elección de los motores es una decisión de gran importancia pues son los encargados de una parte esencial para efectuar el vuelo. Existen dos tipo, los primeros son motores con escobillas también conocidos como motores de DC o brushed motors y los segundos motores sin escobillas o motores de DC sin escobillas

También llamados Brushless CD motors, son motores que carecen de colector y escobillas o carbones, en vez de funcionar en CD, funcionan con una señal trifásica que aunque idealmente debería de tener una forma sinusoidal, en la práctica son pulsos [22].

En los motores brushless, las bobinas rodean los imanes. Es el variador electronico el encargado de activar estas bobinas consecutivamente, haciendo que el rotor gire. El motor dispone de sensores que son capaces de detectar la orientación del rotor en cada momento, para así poder activar y desactivar las bobinas en el momento adecuado. Los imanes del centro son atraídos por la polaridad de un campo magnético generado en las bobinas, las cuales reciben pulsos en un patrón específico [23].

Dentro de la familia de los motores brushless, existen dos tipos diferentes, los tipos Inrunner y los  Outrunner. Los motores outrunner, tienen los imanes situados en el exterior de la estructura, y se puede ver como la parte exterior del motor, gira. En los motores inrunner ocurre lo contrario, los imanes están en el interior de la estructura, y por tanto se ve que lo único que gira es el eje. Los motores outrunner giran mucho más despacio y su par es mayor, la mayor ventaja es el hecho de que no es necesaria una caja de cambios, lo que los hace más silenciosos. Son ligeramente menos eficientes que los inrunner, pero es tan pequeña la diferencia que no debiera de ser un factor determinante a la hora de hacer la elección. Los motores inrunner son más eficientes cuanto más rápido gira el motor y en general son mejores que los outrunner, necesitan de un elemento adicional entre el motor y la hélice. La parte negativa de los motores inrunner es que estas partes adicionales pueden y suelen dar problemas.

La tabla 1 muestra las características de los motores sin escobillas, aunque para ello sea necesario la incorporación de variadores de velocidad, las ventajas son muchas comparado con los inconvenientes que tiene, el motor que se eligió fue el modelo de Motor brushless MT2213 multistar mostrado en la figura 10.

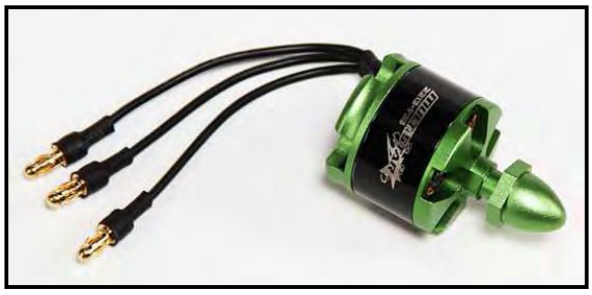


Figura 10.- Motor sin escobillas MT2213.

Los motores sin escobillas son un tipo de motor síncrono, esto es, que tanto el campo magnético generado por el rotor como el del estator giran a la misma frecuencia. Una de las características que define este tipo de motor es que no existe el comúnmente denominado “deslizamiento”. Este tipo de motor existe en diferentes configuraciones aunque la más normal es la configuración dotada de tres fases. Respecto al estator, este se compone de tres devanados en la mayoría de los casos, pudiendo estar estos conectados en estrella o en triángulo aunque la configuración más común es la de estrella. Las tensiones inducidas son de forma trapezoidal como muestra la figura 11 [24].

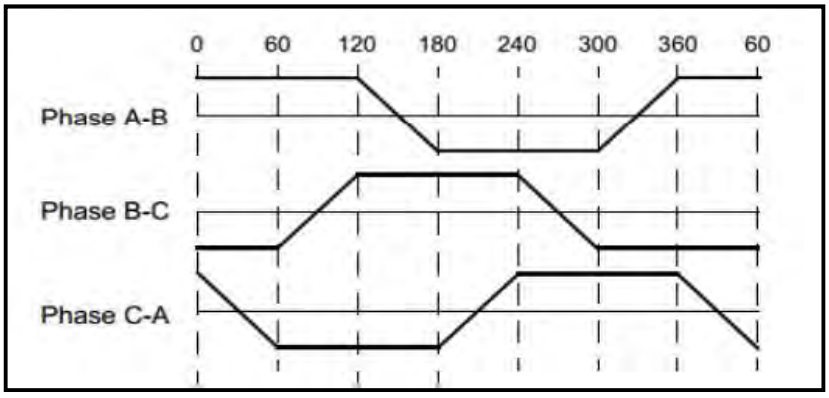



Figura 11.- Fases de terminales inducidas.

El funcionamiento de las secuencias de alimentación de este tipo de motores consiste en que al tener tres devanados en estrella figura 12, se va alimentando siempre uno con una tensión positiva por el que entra la corriente, otro con tensión negativa saliendo la corriente por este y un tercer devanado que no se encuentra alimentado y por lo tanto no circula corriente por el mismo. Para que el motor funcione y por lo ende genere par, se tiene que dar la condición de que exista un ángulo entre el campo magnético generado por los devanados alimentados y el campo magnético propio de los imanes del rotor. Para obtener el máximo par y un funcionamiento perfecto el objetivo es mantener siempre el ángulo lo más cercano a 90º dando lugar al par máximo. De esta manera la secuencia en cada momento ha de ir adecuándose al giro del rotor de forma que se mantengan lo más posible la perpendicularidad entre ambos campos magnéticos.

La característica de este motor en cuanto a par-velocidad es totalmente horizontal, esto se traduce en que es capaz de dar par nominal en todo el rango de velocidades entre cero y la velocidad nominal del motor.

Fuera de esta zona de funcionamiento tenemos que existe una caída del par máximo al superar la velocidad nominal y otra zona que va desde velocidad cero hasta la velocidad nominal en la que el motor de forma temporal (el tiempo dependerá de cada modelo concreto de motor) es capaz de un par mayor al par nominal, esto se traduce en un sobrecalentamiento por eso se puede definir esta capacidad de dar un par “extra” como una característica dinámica.

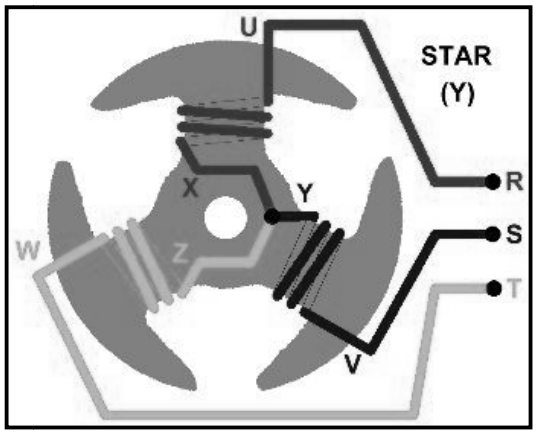



Figura 12.- Devanados de motores en “Y”.

TABLA 1.- Comparativa de motores.

| | **MOTOR SIN ESCOBILLAS** |
|---|---|
| **CONMUTACIÓN** | De tipo electrónica basada en sensores de posición |
| **MANTENIMIENTO** | Poco ya que carece de escobillas |
| **VIDA ÚTIL** | Vida útil larga |
| **RENDIMIENTO** | Alto porque no hay perdidas en el rotor |
| **RANGO DE VELOCIDAD** | Alto por no estar limitado mecánicamente |
| **CORRIENTE DE ARRANQUE** | Corriente nominal |
| **DESLIZAMIENTO** | Nulo entre el rotor y el estator |

Con base en el peso del cuadricóptero, y recordando la configuración de fuerzas cada rotor debería producir un empuje igual a un cuarto del peso total así, si el peso fuera de 1,2 kg cada rotor debe producir 300 gr de empuje vertical.

Dado lo anterior y viendo la capacidad de empuje vertical necesario para cada motor, se decidió que la fuerza fuera generada por los motores brushless outrunner MT2213-935KV MultiStar antes descritos, que tienen por especificaciones técnicas 935 KV (RPM/V), celdas LiPo 2-4s, proporción máxima de 200 W, máxima corriente(10s) de15 Amps., Corriente sin carga de 0.4 Amps, Resistencia interna de 0.180 ohm, numero de polos 14, dimensiones 28 x 26 mm , eje del motor de 3mm , eje de las hélices de 8mm, hélices de 10 x 4.5(3S), 8 x 4.5 (4S), peso de 55g y espacio de los orificios de montaje de 16mm * 19mm.

### 4.2.4. Hélices

Otro de los aspectos importantes son las hélices ya que junto con los motores son los que desarrollan la fuerza de empuje vertical y los movimientos del cuadricóptero, las hélices normalmente suelen ser de dos palas, pero pueden ser de tres o cuatro, aparte del número de palas, las dos características principales son el tamaño y el paso como se ver en a figura 13, ambas medidas suelen estar expresadas en pulgadas (").

El tamaño de la hélice de dos palas es lo que mide de punta a punta (si es de tres o cuatro palas será el diámetro máximo) las palas se puede decir de otra manera que son las aspas. Cabe mencionar que el paso de una hélice es suponiendo que estuviera en un medio sólido, las pulgadas que avanzaría con cada giro completo.

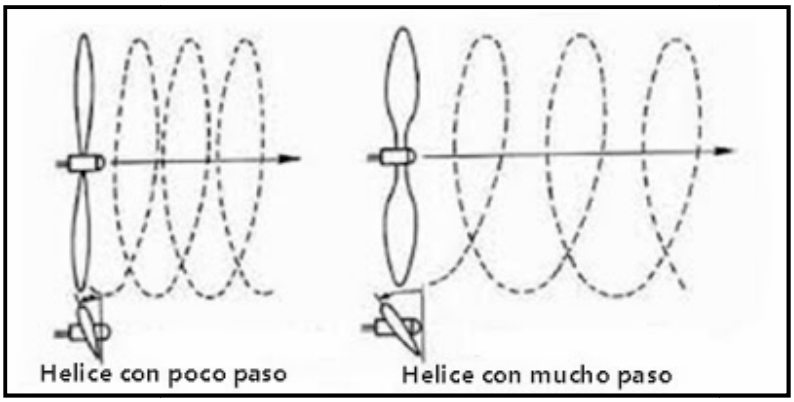



Figura 13.- Tipo de paso de las hélices.

Para un mismo paso, cuanto mayor sea el tamaño de la hélice, más empuje se obtiene y mayor por consecuencia. Cuanto mayor sea el paso de la hélice, mayor será la velocidad máxima y menor el empuje (estático), pero de igual manera aumentara el consumo del motor.

Como norma general emplearemos motores de pocas revoluciones por voltio (Kv), con hélices grandes y de poco paso para aviones o multicopteros de tracción (remolcadores de veleros y multicopteros para filmación aérea) y aviones de vuelo lento (entrenadores de ala alta) pero, en cambio, emplearemos motores con altas revoluciones por voltio y hélices pequeñas con mucho paso para aviones ligeros de vuelo rápido (aviones para carreras de pilón) [25].

Las hélices que se seleccionaron fueron de acuerdo a la configuración del giro de los motores de un cuadricóptero, en el cual dos giran en un sentido es decir horario y dos en sentido contrario o anti horario dos denominadas flyer y dos pusher, el motor seleccionado incluía una recomendación de hélices ya que venía en un combo de un motor y dos hélices que tienen por características 10 x 4,5 pulgadas cada hélice figura 14.

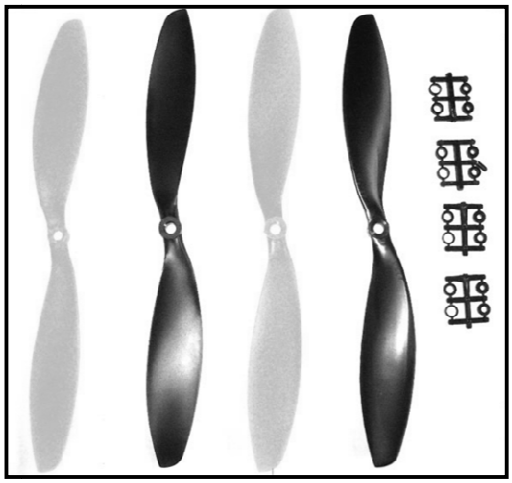


Figura 14.- Hélices y adaptadores.

### 4.2.5. ESC O Variador

Un controlador de velocidad electrónico o ESC (del Inglés Electronic Speed Controller) también conocido como variador, es un circuito electrónico cuyo objetivo es variar la velocidad de un motor eléctrico pudiendo ser de igual manera freno dinámico.

En este caso se utilizaron cuatro ESCs para controlar los motores brushless, cada ESC proporcionará a cada motor una señal eléctrica trifásica que generará la rotación de las hojas. Un ESC es un controlador PWM para motores eléctricos. Un ESC, dispone de dos cables de alimentación que van conectados a la batería, contiene otros tres cables que van al motor, y dispone de otros 3 cables más que van al microcontrolador. Mediante estos tres cables, los cuales son más delgados que el resto es por los que el microcontrolador le indica al ESC a qué velocidad desea que el motor gire. Una de las características de selección de los controladores de velocidad es la corriente de operación máxima ya que los motores seleccionados consumen como máxima corriente 15 A [26].

Los tres cables que van al motor, como se muestra en la figura 15, son la señal trifásica que hacen girarlo, los dos cables del lado contrario en el ESC van hacia la batería y los tres cables que son los más delgados van hacia el receptor, en este caso sólo se utilizaran dos de estos cables, el cable de la señas y el negativo.

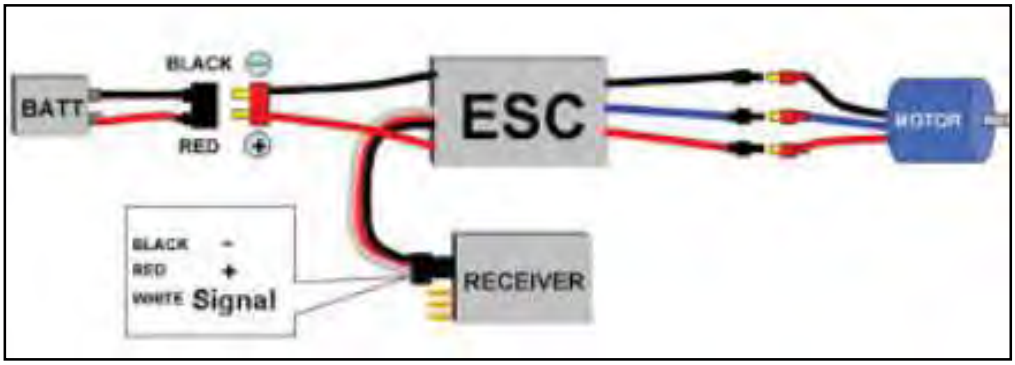



Figura 15.- Conexión del ESC.

El ESC tiene como función principal ajustar las revoluciones del motor a lo que se pida, típicamente a través del canal 3 del receptor (movimiento vertical de la palanca izquierda,) pero también es muy común que sea el encargado de suministrar la corriente a los otros componentes electrónicos, como el receptor, servos, etc. Para ello incorporan una fuente de alimentación que suministra un voltaje estabilizado a 5-6V a través del canal 3 del receptor y se conoce como BEC, sí nuestro variador no tiene el BEC incorporado tendremos que poner una batería de 5V aparte o emplear un BEC dedicado que alimente al receptor y servos [26].

Los variadores, igual que los motores, pueden soportar un máximo de voltaje y amperaje, si excedemos este máximo, se quemarán, dejando de suministrar corriente al motor y al resto de componentes, por lo que perderemos el control total del avión, helicóptero o multicoptero de radiocontrol que estemos manejando. Para saber el máximo voltaje admitido por el variador debemos consultar las especificaciones. El máximo amperaje soportado normalmente se incluye a continuación del nombre, por ejemplo AfroESC 30A, Turnigy plus 20A, etc. [26].

Se aplica, si el consumo que hemos estimado para un motor, por ejemplo, es de 20A tendremos que comprar un variador de por lo menos 24A, es decir, hay que aplicar un coeficiente de seguridad de un 20% en el consumo del variador.

El variador o ESC recibirá la señal PWM de 50 Hz y dependiendo de la longitud del ancho de pulso entregará más o menos potencia al motor. La longitud del pulso PWM varía de 1 ms a 1.5 ms, parado y a máxima potencia respectivamente. Las longitudes de estado lógico alto varían en función del fabricante y el modelo, por lo tanto puede ser que para otro fabricante diferente, Normalmente son: 1-1.5 ms, 1-2 ms, 1-2.5 ms. El pulso mínimo es 1 ms y no 0 ms, para evitar errores en la transmisión, por ejemplo interferencias.

Los ESC dentro, tienen un circuito electrónico controlado por un microcontrolador, normalmente un Atmel de tipo AVG. Podrían tener mayor frecuencia de refresco este microcontrolador, pero se comercializan para ser utilizados en RC por lo que no se fabrican para tener mayor frecuencia de refresco sino para responder correctamente a los receptores RC. Existen ESC comerciales con más frecuencia de refresco, pero normalmente no se comercializan ya que el standard es el PWM 50Hz.

Algunos ESC se comunican con el protocolo I2C alcanzando altas tasas de refresco, pero tienen el gran problema de que son muy caros ya que su aplicación es prácticamente la construcción de multicopteros. Se podrían modificar un ESC con control por PWM para funcionar con I2C, existen muchos modos para diferentes ESC comerciales aunque suelen ser procesos complejos donde se invierte mucho tiempo para el dinero que se ahorra.

Las siguiente imágen muestra el ESC utilizado figura 16, tomando en cuenta lo anteriormente visto, y viendo la capacidad requerida de los ESC para el cuadricóptero, se eligió un UBEC de capacidad de 20A, y tienen por especificaciones técnicas corriente constante de 20A, máxima corriente de 25A, batería 2-4S Lipoly/5-12s NiXX, BEC de 5v/3A, dimensiones de 54 x 26 11mm y peso de 30g [27].


Figura 16.- ESC 3A UBEC.

### 4.2.6 Batería

La capacidad de una batería se mide por el ratio de descarga. El ratio de descarga C de una batería, es la máxima cantidad de corriente que puede proporcionar.

Se trata de una batería de 3 celdas (3S) de 3000mAh que es equivalente a una capacidad de 3000mA (3A.) en una hora. El 20 C es el ratio de descarga o la máxima cantidad de corriente que puede proporcionar. Podemos calcular la corriente total (I), 3000ma x 20C = 3A x 20 = 60 A.

Si dividimos los 60A entre los 4 motores que tenemos, obtenemos un resultado de 15A. Demostrando así que esto coincide perfectamente con la especificación de los motores de máxima corriente.

Se eligió por tal motivo la batería ZIPPY Flightmax 3000mAh que se muestra en la figura 17 las especificaciones son capacidad de 3000mAh, voltaje de 3S1P / 3 Cell / 11.1v, descarga de 20C Constante / 25-30C Burst, peso de 180g, dimensiones 102x37x24mm y Balance Plug JST-XH [28].


Figura 17.- Batería de litio polímero.

### 4.2.7. Placa Arduino

El Open source hardware o hardware libre se refiere al conjunto de dispositivos diseñados con la misma filosofía que el software FOSS (del inglés Free and open source software). El término significa que la información sobre el hardware está accesible fácilmente, tanto el diseño del hardware (planos mecánicos, esquemas, PCB, código HDL y disposición del circuito integrado) como a software que utiliza el hardware. Ambos son llevados a cabo con el enfoque de software libre abierto y gratuito. Por tal motivo y por qué uno de los objetivos externos es que esté basado en tecnología de este tipo, se ha elegido una placa que sigue esta filosofía.

Arduino es una plataforma de prototipo electrónico basada en el principio del hardware y software libre. Arduino es el caballo de batalla en el mundo del Hardware Open Source, el proyecto nace en Italia en 2005.

Arduino puede tomar información del entorno a través de sus pines de entrada de una gran gama de sensores y puede realizar control como lo puede ser de luces, motores y otros actuadores. El microcontrolador en la placa Arduino se programa mediante el lenguaje de programación Arduino (basado en Wiring) y el entorno de desarrollo Arduino (basado en Processing). Los proyectos hechos con Arduino pueden ejecutarse sin necesidad de conectar a un ordenador, si bien tienen la posibilidad de hacerlo y comunicar con diferentes tipos de software [29].

Hay muchos otros microcontroladores y plataformas con microcontroladores disponibles para la computación física, Parallax Basic Stamp, raspberry, BX-24 de Netmedia, Phidgets, Tiva c, y muchos otros ofrecen funcionalidades similares. Todas estas herramientas organizan el complicado trabajo de programar un microcontrolador en paquetes fáciles de usar.

Arduino, además de simplificar el proceso de trabajar con microcontroladores, ofrece algunas ventajas respecto a estos y otros sistemas como lo son ser [29]:

● Asequible - Las placas Arduino son más asequibles comparadas con otras plataformas de microcontroladores.

● Multi-Plataforma - El software de Arduino funciona en los sistemas operativos Windows, Macintosh OSX y Linux. Teniendo una amplia gama de utilización y abarcando la mayoría de computadoras para su utilización.

● Hardware ampliable y de Código abierto - Arduino está basado en los microcontroladores ATMEGA168, ATMEGA328 y ATMEGA1280. Los planos de los módulos están publicados bajo licencia Creative Commons, por lo que diseñadores de circuitos con experiencia pueden hacer su propia versión del módulo, ampliándolo u optimizándolo.

● Software ampliable y de código abierto- El software Arduino se publica bajo una licencia libre y preparada para ser ampliado por programadores experimentados. El lenguaje puede ampliarse a través de librerías de C++, y si se está interesado en profundizar en los detalles técnicos, se puede dar el salto a la programación en el lenguaje AVR C en el que está basado.

● Entorno de programación simple y directo - El entorno de programación de Arduino es fácil de usar para principiantes y lo suficientemente flexible para los usuarios avanzados.

La decisión de elegir usar Arduino para este proyecto son cuatro, la primera ser económico y de fácil adquisición, la segunda que es de Open Source, la tercera por su facilidad de utilización y familiarización y por ultimo pero no menos importante la expansión de su uso. El modelo que se ha utilizado es el Arduino

#### 4.2.7.1. Arduino Uno

El Arduino Uno figura 18 dispone de un procesador ATMEL ATmega328 a 16 MHz, memoria Flash de 32 KB y 1 KB de memoria EEPROM (memoria que no se borra al cortar el suministro eléctrico), los pines operan a 5 V. Tiene 14 digitales de los cuales 6 pueden actuar como salida de tipo PWM, se usan cuatro de ellos para la salida de las señales que van a los ESC para el control de velocidad de los motores, tiene 6 entradas analógicas a las donde de conectaran los sensores que se utilizaran voltaje de operación5V, voltaje de entrada de 7-12V, voltaje de entrada (limites) de 6-20V, largo de 68.6 mm, ancho de 53.4 mm y Peso 25g [30].

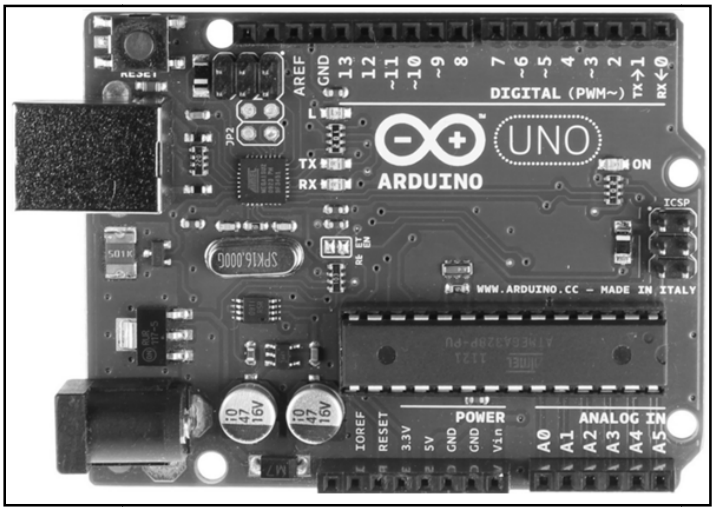



Figura 18.- Arduino Uno.

### 4.2.7.2. Arduino Micro

El Arduino Micro figura 19 es una placa electrónica basada en el ATmega32U4, cuenta con 20 pines digitales de entrada / salida, de los cuales 7 se pueden utilizar como salidas PWM y 12 entradas como analógicas, tiene un factor de forma que le permite ser fácilmente colocado en una placa, voltaje de operación 5V, voltaje de entrada de 7-12V, voltaje de entrada (limites) de 6-20V, largo de 48 mm, ancho de 18 mm y Peso de 13g [31].

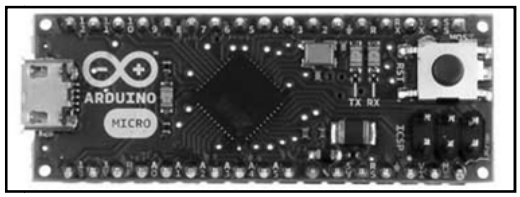



Figura 19.- Arduino Micro.

### 4.2.8. Módulo IMU

IMU (Inertial Measurement Unit) o unidad de medición inercial es un dispositivo electrónico que mide e informa acerca de la velocidad, orientación y fuerzas gravitacionales de un aparato, usando una combinación de acelerómetros y giróscopos. La IMU funciona detectando la actual tasa de aceleración usando uno o más acelerómetros, además detecta los cambios en atributos rotacionales como son el yaw pitch y roll mediante giroscopios.

El módulo Giróscopo en un sensor de rango angular de 3 ejes con datos de temperatura. El giróscopo muestra el cambio de rango en rotación en sus ejes X, Y y Z. Los datos de salida de temperatura y el rango angular medido. Los datos de salida de temperatura y la medición angular se pueden acceder de la interface seleccionable I²C o SPI. El módulo es un diseño pequeño y tiene acceso a una interface SIP con un orificio de montaje para una rápida conexión a los proyectos. Este está diseñado para trabajar con una amplia gama de micro controladores y diferentes requerimientos de voltaje.

El módulo L3G4200D figura 20 es un sensor de rango de 3 Ejes (vertical, lateral y longitudinal) cuenta con tres escalas seleccionables 250/**500**/2000 grados/segundos, contiene sensor de temperatura integrado -40 a + 85 °C un rango de salida de datos de 16-bits y rango de salida de datos de temperatura de 8-bits

Las características técnicas son consumo de 2.7 a 6.5 Volts de CD, interface de comunicación I2C (hasta 400 kHz) o SPI (10 MHz; 4 y 3 cables), temperatura de operación de -40 a +85 °C y dimensiones de 2.16 X 2.03 cm.

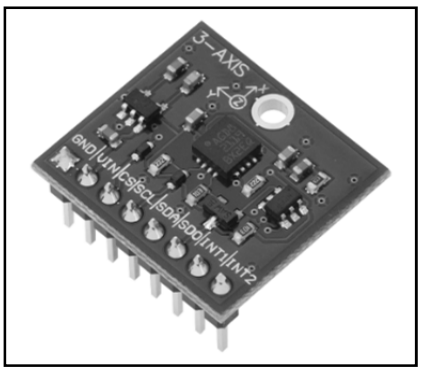



Figura 20.- Módulo L3G4200D.

### 4.2.9. Módulo Bluetooth

La tecnología por la cual se optó ocupar para el proyecto fue el Bluetooth, porque es un enlace que no requiere de recursos extras para su operación, generando accesibilidad total sin mayores gastos para su operación, implementación y funcionamiento, al mismo tiempo de tener características excelentes para la transmisión, acoplándose a las necesidades requeridas del proyecto. El módulo que se utilizó se eligió por su accesibilidad y bajo costo, pero al mismo tiempo se tomó en cuenta el ajuste de la comunicación con el microcontrolador.

Bluetooth es un enlace de radio de corto alcance sin necesidad de visión en línea directa entre los dispositivos implicados en la comunicación. Las especificaciones de Bluetooth son desarrolladas por el Bluetooth SIG (Special Interest Group). El grupo de interés especial fue fundado en febrero de 1998 por IBM, Intel, Ericsson, Nokia y Toshiba. La versión 1.0 del estándar abierto Bluetooth fue publicada en 1999, y la versión 1.1 apareció a finales del año 2001 y es la que implementan los dispositivos actuales [32].

El rango de frecuencias utilizada por Bluetooth es de 2,4 GHz ISM (Industrial Scientific and Medical); en concreto las frecuencias entre 2400 y 2483,5 MHz. Se puede transmitir voz, datos e incluso vídeo a velocidades de hasta 721 kbps. En efecto, Bluetooth no sólo soporta las comunicaciones de datos, sino que, además ofrece hasta tres canales de voz 64 kbps, ampliando así el número de aplicaciones.

Están definidas en el estándar dos potencias de emisión en función de la distancia que se desea cubrir, para 10 metros con 1 mW y para 100 metros con 100 mW. Esta tecnología limita además la potencia de salida de los transmisores de radio exactamente al valor necesario, lo cual permite aumentar el tiempo de vida de las baterías de los dispositivos que lo usan. Además, mediante esta tecnología también se puede desplazar el modo de operación a baja potencia (estado de standby) cuando el dispositivo no esté transmitiendo información [32].

Los datos se pueden intercambiar a velocidades de hasta 1 Mbps (considerando los bits de control de errores, autentificación, etc.). El esquema de transmisión de espectro ensanchado con saltos de frecuencia aleatorios (spread spectrum - frecuency hop) permite a los dispositivos comunicarse inclusive en áreas donde existe una gran interferencia electromagnética. Además, Bluetooth dispone de mecanismos de encriptación y autenticación para controlar la conexión y evitar que cualquier dispositivo no autorizado, pueda acceder a los datos o modificarlos. Así, puesto que el enlace es codificado y protegido contra interferencia y pérdida de enlace, Bluetooth puede considerarse como una red inalámbrica de corto alcance muy robusta y segura [32].

La red se divide en picoceldas (o piconets) de radio de cobertura reducida, como se ve en la figura 21, siendo ocho el número máximo de unidades que pueden participar activamente en una simple piconet un maestro y siete esclavos. Para establecer la piconet, la unidad maestra debe conocer la identidad del resto de unidades que están en modo standby en su radio de cobertura. El maestro o aquella unidad que inicia la piconet transmite el código de acceso continuamente en saltos de portadora hasta que el receptor contesta o se excede el tiempo de respuesta [32].

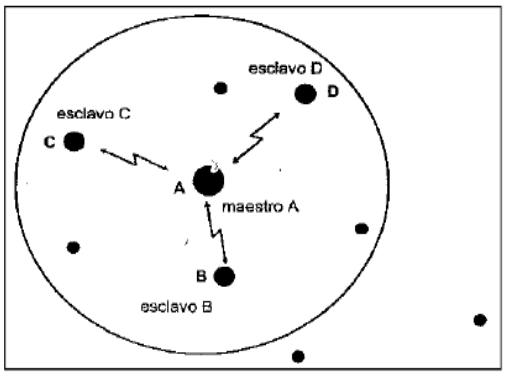



Figura 21.- Picored Bluetooth.

TABLA 2.- Características de Bluetooth.

| | |
|---|---|
| **Banda de frecuencia** | 3.4 GHz (banda ISM) |
| **Potencia del transmisor** | 1mW para alcance de 10m<br>100mW para alcance de 100m |
| **Tecnología** | Espectro ensanchado<br>Saltos en frecuencias |
| **Canales máximos** | Voz:3 por piconet<br>Datos:7 por piconet |
| **Velocidad de datos** | Hasta 721Kbps por piconet |
| **Números de dispositivos** | 8 por piconet y hasta 10 piconets |
| **Consumo de potencia** | 30uA-30mA transmitiendo |
| **Interferencia** | Mínima al emplear saltos rápidos en frecuencia de 1600 veces por seg. |

Al inicio del proyecto se analizó de qué manera se iba a realizar la comunicación mediante WiFi o Bluetooth, ambas opciones eran viables, sin embargo se eligió la opción de Bluetooth por las características antes mencionadas, ahora bien, con el módulo HC-05 de la figura 22, se conecto el teléfono con las placas Arduino, para realizar la comunicación y ejecución de instrucciones o señalización.

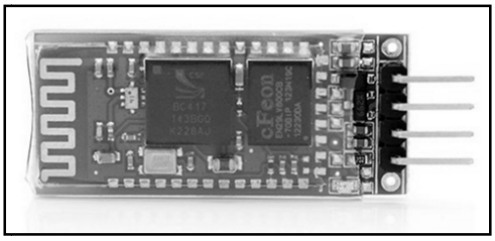


Figura 22.- Módulo Bluetooth.

### 4.2.10. GPS

Un receptor para un Sistema de Posicionamiento Global debe captar las señales de cuatro de los satélites GPS para dar una completa posición tridimensional. Para este proyecto se pretende usar el GPS para obtener las coordenadas de ambos elementos, tanto del dispositivo con sistema operativo Android como del cuadricóptero. Diferentes tipos de receptores hacen uso de diferentes partes de la estructura de la señal GPS [33].

1.- Código C/A (Coarse Acquisition)
2.- Fase portadora L1
3.- Fase portadora L2
4.- Código P (militar)

El uso de estos aparatos va desde la gama de receptores de mano y dispositivos de navegación personal, hasta herramientas de precisión topográfica.

1.- Código C/A (Coarse Acquisition)
2.- Código C/A + fase portadora L1
3.- Código C/A + fase portadora L1 + fase portadora L2
4.- Código C/A + fases portadoras L1, L2 + código P (militar)

Los principales errores del sistema GPS son una combinación de ruido y desajustes, estas se deben al cronometraje impreciso del reloj del receptor. El sistema incorporado en el receptor, utiliza señales de radio de microondas que viajan a la velocidad de la luz desde al menos tres satélites, para calcular su posición, altitud y velocidad, diferencias minimas provocan que las distancias (datos) puedan variar.

También existe una fuente de error intrínseca en el GPS asociada al modo de funcionamiento del sistema. Los receptores GPS analizan las señales de tres satélites y calculan el tiempo que han tardado en captar cada señal. Esto les permite realizar un cálculo de la trilateración para identificar la ubicación exacta del receptor.

Desgraciadamente, el detector de los dispositivos GPS estándar sólo es preciso en el 1% del tiempo. Esto representa aproximadamente diez mil millonésimas de segundo (10 nanosegundos). Dado que las señales de microondas GPS viajan a la velocidad de la luz, esto es equiparable a un error de 3 metros. Por lo tanto, los GPS estándar no pueden determinar la posición con un error de precisión inferior a 3 metros [33].

Hay otros errores derivados de alteraciones atmosféricas que distorsionan las señales antes de llegar al receptor. Los reflejos de edificios u otros objetos grandes y sólidos también pueden causar problemas de precisión del GPS, asimismo puede haber problemas con la precisión de cronometraje de un satélite. Estos problemas de precisión son evitados por los receptores GPS, que se comunican con más de tres satélites para obtener datos.

a) Otras fuentes de error

Los errores de los relojes de los satélites no corregidos por las estaciones de control de tierra pueden originar errores de 1 metro.
Los errores de acontecimientos imprevistos pueden producir 1 metro de error, retardos troposféricos, 1 metro. La tropósfera es la capa más baja de la atmósfera (desde la superficie hasta entre 8 y 13 Km), esta capa está afectada por cambios de temperatura, presión y humedad asociados a cambios meteorológicos [33].

Retardos por la ionosfera, 10 metros. La ionosfera es la capa de la atmósfera que va desde los 50 hasta 500 Km de altura y consiste en aire ionizado. El modelo de transmisión para esta capa, que es enviado en la trama de datos, puede eliminar la mitad de los estos dejando un residuo que puede dar errores de 10 metros. Los receptores que usan las portadoras L1 y L2 pueden corregir todo el error [33].

Multicamino: ½ metro. Es causa por la reflexión de las señales en superficies próximas al receptor y puede interferir o producir errores en las señales que llegan directamente desde los satélites al receptor. El error por multicamino es muy difícil de detectar y en ocasiones es imposible de evitar.

b) Características técnicas y prestaciones

El Sistema Global de Navegación por Satélite (GPS) está compuesto por:

Satélites en la constelación: 24 (4 × 6 órbitas)
Altitud: 20km – 20.2 km
Período: 11 h 58 min (12 horas sidéreas)
Tiempo sidéreo o sideral es el tiempo medido por el movimiento diurno aparente del equinoccio vernal, que se aproxima aunque sin ser idéntico, al movimiento de las estrellas, el tiempo sidéreo se define como el ángulo horario del equinoccio vernal.
Cuando el equinoccio vernal culmina en el meridiano local, el tiempo sidéreo local es 00.00.
Inclinación: 55 grados (respecto al ecuador terrestre).

Segmento de control (estaciones terrestres)

Estación principal: La estación central o maestra (Consolidated Satellite Operation Center-CSOC o Master Control Center) que se encuentra en Colorado Springs, exactamente en la base Falcon de la U.S. Air Force, hay otra estación central de reserva en Sunnivale (California), concretamente en la Base Ozinuka de la U.S. Air Force.

Antena de tierra: 4
Estaciones monitoras (de seguimiento):Colorado Springs, Hawái, Kwajalein, Isla de Ascensión e Isla de Diego García

Señal RF (Frecuencia portadora)
Civil – 1575,42 MHz (L1). Utiliza el Código de Adquisición Aproximativa (C/A).
Militar – 1227,60 MHz (L2). Utiliza el Código de Precisión (P), cifrado.
-Nivel de potencia de la señal: –160 dBW (en la superficie).
-Polarización: circular dextrógira (movimiento circular a la derecha).

Precisión: oficialmente aproximadamente 15 m (en el 95 % del tiempo). En la realidad un GPS portátil monofrecuencia de 12 canales paralelos ofrece una precisión de entre 2,5 y 3 metros en más del 95 % del tiempo. Con el WAAS / EGNOS / MSAS activado, la precisión asciende de 1 a 2 metros..

Hora (tiempo): 1 ns.
Cobertura: mundial
Capacidad de usuarios: ilimitada
Integridad: tiempo de notificación de 15 minutos o mayor.
Disponibilidad: 24 satélites y 21 satélites.
Sistema de coordenadas: Sistema Geodésico Mundial 1984 (WGS84).

Hay varias distancias de los puntos del perimeto de la tierra hasta su centro, estos están en un rango que va desde el radio polar de 6357 km, al radio ecuatorial de

6378 km. La precisión teórica que se tomo para la longitud 1° de la circunferencia de la tierra equivale a 111.3171 km y para la latitud 1° equivale a 110.95058 km, partiendo de estas equivalencias se realizó una escala tanto para 1° de la longitud así también como para 1° se la latitud, esto se muestra en las tablas 3 y 4 respectivamente. Se puede observar que con el uso de de la equivalencia de cuatro dígitos antes del punto decimal en grados se tiene una precisión de 11.13171 m para longitud y de 11.095058 m para latitud, y con cinco dígitos se tiene una precisión de 1.113171 m y 1.1095058 m cm respectivamente.

2*π*r= π*d
Radio polar de 6357 km * 2 = 12, 714 km
Radio ecuatorial de 6378 km * 2 = 12, 756 km
12, 714 km * π = 39, 942209 km → 39, 942.209 km / 360 = 110.95058 km
12, 756 km * π = 40, 074156 km → 40, 074.156 km / 360 = 111.3171 km
**110.95058 km** por cada grado para latitud
**111.3171 km** por cada grado para longitud

TABLA 3.- Escala longitud 1°.

| GRADOS | DISTANCIA |
|---|---|
| 0.1° | 11.13171 km |
| 0.01° | 1.113171 km |
| 0.001° | 111.3171 m |
| 0.0001° | 11.13171 m |
| 0.00001° | 1.113171 m |

TABLA 4.- Escala latitud 1°.

| GRADOS | DISTANCIA |
|---|---|
| 0.1° | 11.095058 km |
| 0.01° | 1.1095058 km |
| 0.001° | 110.95058 m |
| 0.0001° | 11.095058 m |
| 0.00001° | 1.1095058 m |

Teniendo en cuenta lo antes mencionado y observamos que la tecnología GPS cuenta con las capacidades necesarias, para ser aplicada en el proyecto ya que se adecua a las necesidades.

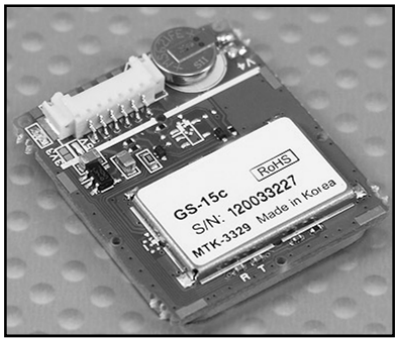



Figura 23.- GPS.

# *CAPÍTULO 5*

## 5.1. Desarrollo del sistema, diagramas y circuitos

En la figura 24 se muestra el diagrama a bloques que representa la relación, de las conexiones simplificadas y operatividad de cada elemento de control, comunicación, interfaces, y actuadores que conforman al cuadricóptero.

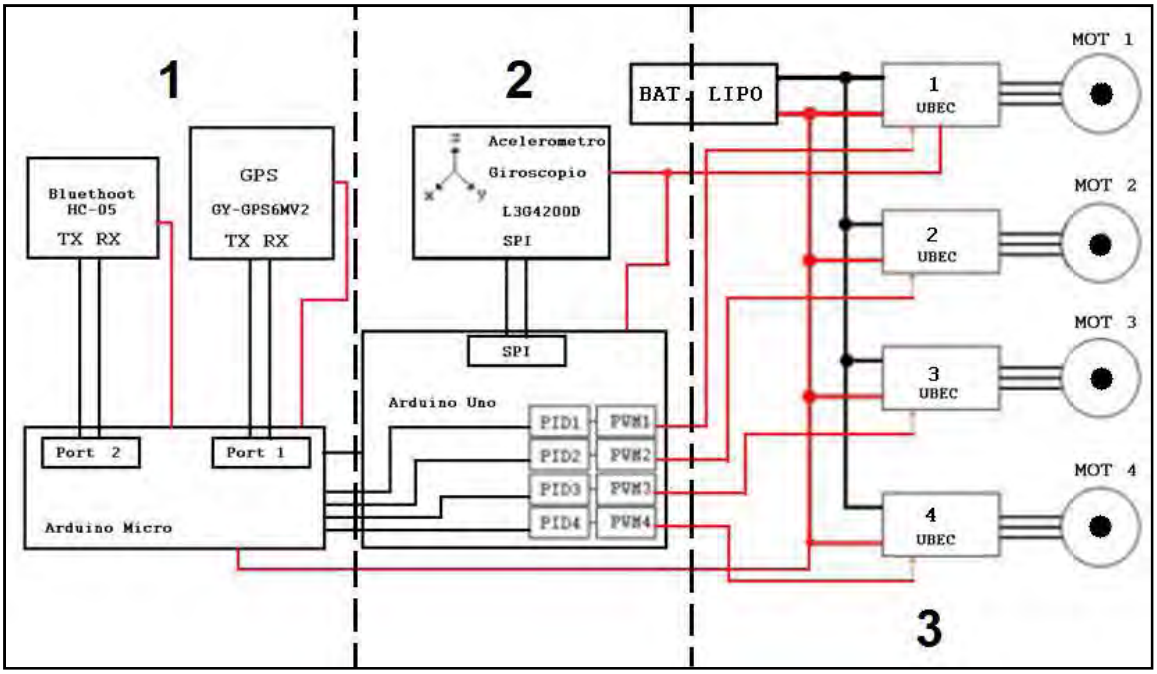



Figura 24.- Diagrama a bloques de la implementación del cuadricóptero.

En la figura 25 se muestra el diagrama dividido en tres secciones, la 1 está conformada por un módulo BT HC-05, el cual se comunica con el módulo BT del dispositivo móvil que trae incorporado para recibir datos e información, un módulo GPS GY-GPS6MV2, el cual proporciona la información de la posición del cuadricóptero y un Arduino Micro con la cual están comunicados a través de dos puertos seriales, en la placa se implementa el algoritmo realizado para el seguimiento, el cual al terminar los procesos de adquisición, adecuación y realización de las diversas operaciones este envía la información resultante a la placa Arduino sección 2 de la figura 24.

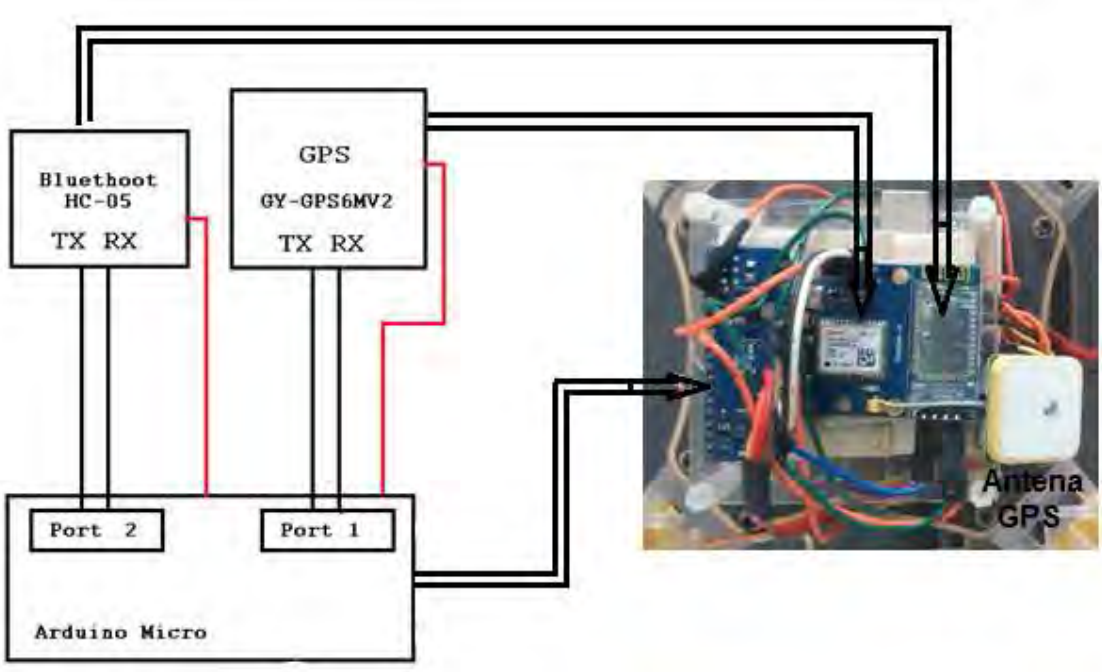



Figura 25.- Implementación física y su relación Arduino Micro.

En la sección 2 se muestra el modulo giróscopo L3G4200D, es un sensor angular de 3 ejes (vertical, lateral y longitudinal) con datos de temperatura, el giroscopio muestra el cambio de rango en rotación en sus ejes X, Y y Z, por tal motivo se coloca en la parte central del cuadricóptero, se accede a los datos de salida mediante la interface SPI que va a la placa Arduino uno el cual contiene el algóritmo de control de vuelo del cuadricóptero.

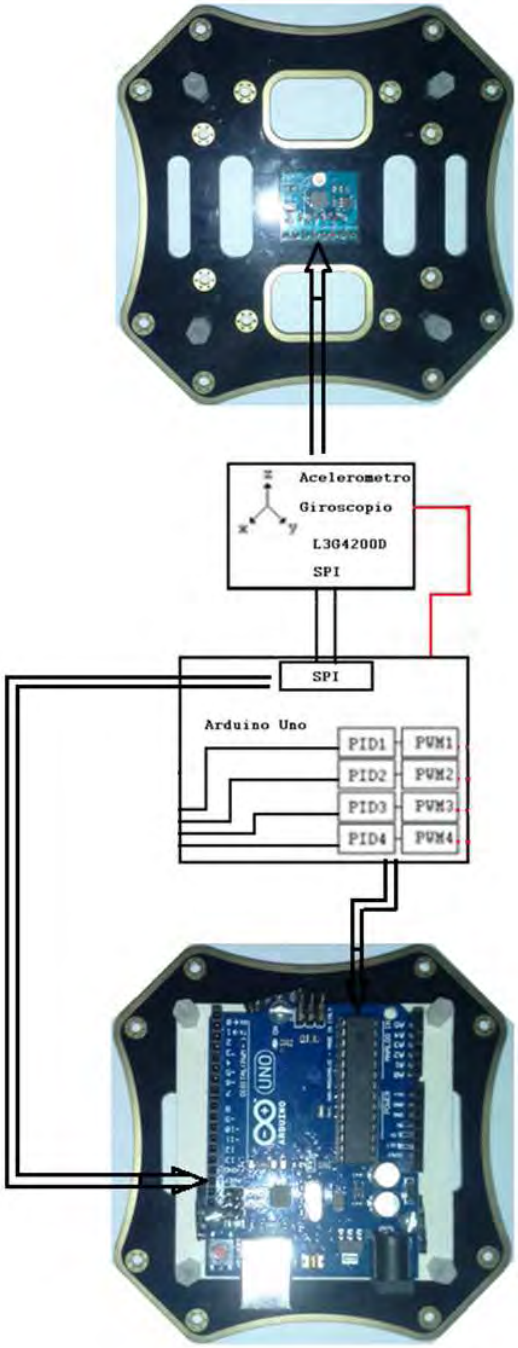



Figura 26.- Implementación física y su relación Arduino Uno.

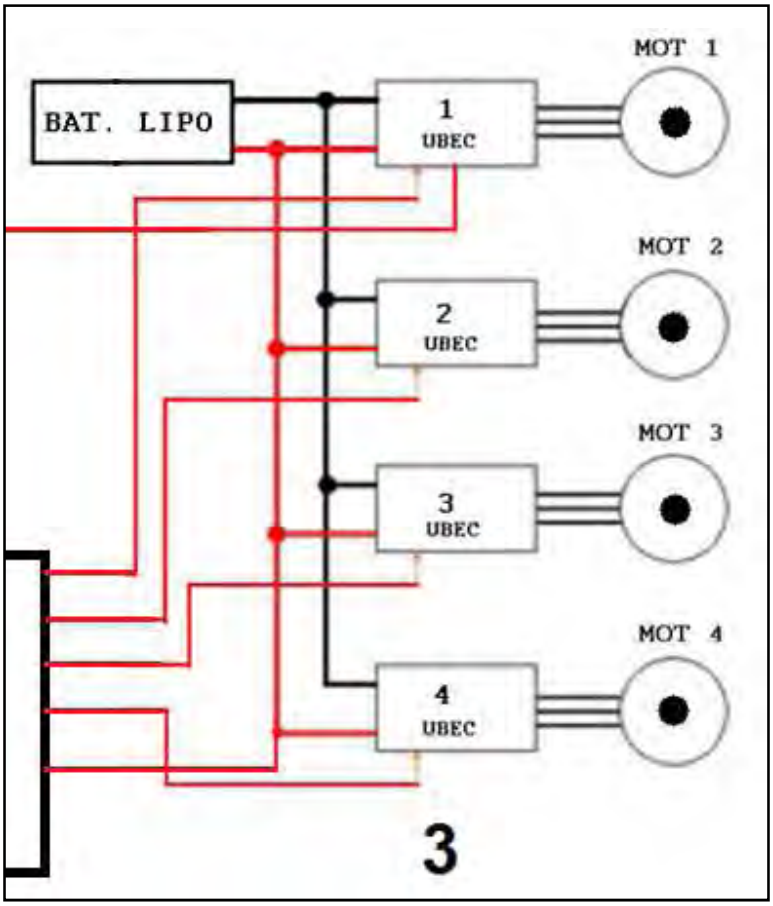



Figura 27.- Sección 3 del diagrama de bloques.

En la sección 3 se tienen los ESC los cuales le indican a los motores que tan rápido o lento deben de ir de acuerdo a las instrucciones mediante PWM provenientes de la placa Arduino Uno, los motores, las hélices y la batería LIPO la cual proporciona la energía a cada componente del cuadricóptero.

## 5.2. Desarrollo del seguimiento

Para el seguimiento, se hizo el uso de sistemas móviles con S.O. Android, como lo es un Sony Xperia M2, Sony Xperia Z y la tableta electrónica Lenovo de la serie A, a los cuales se les descargó la aplicación que se desarrolló.

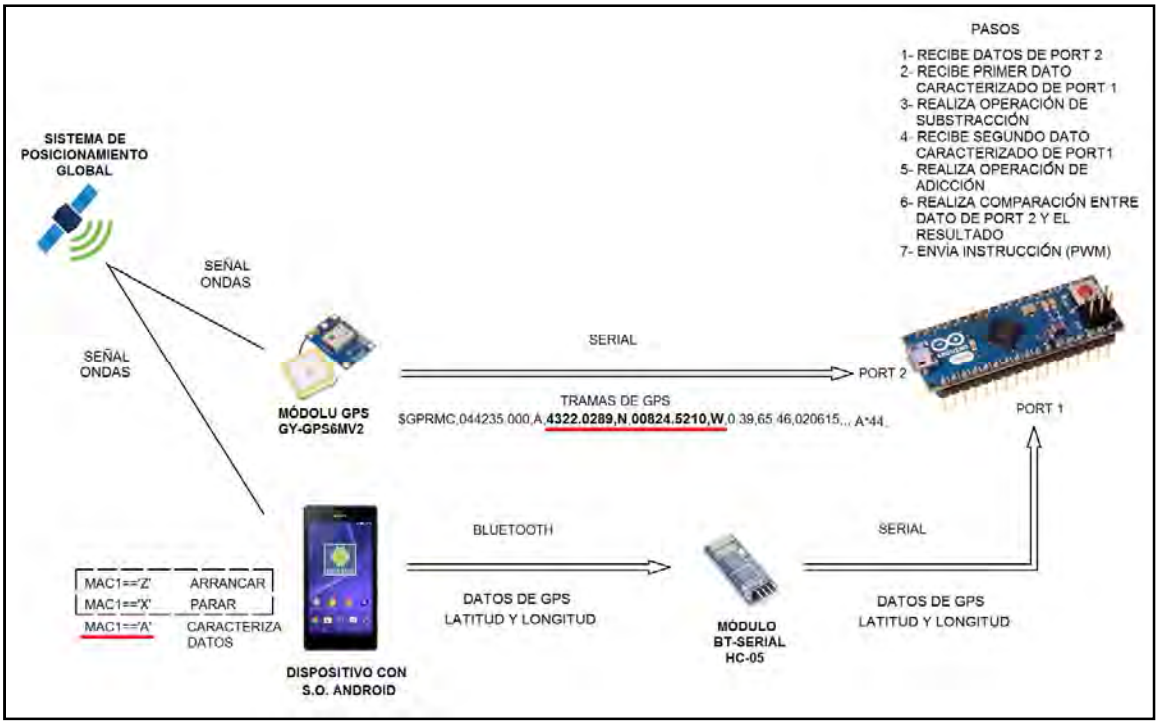



Figura 28.- Diagrama de transmisión de la información.

Teniendo en cuenta que se realiza una comunicación inalámbrica por Bluetooth para la transferencia de los datos generados de la aplicación en el dispositivo móvil, la transferencia de esta se envía a través de la terminal inherente en el dispositivo móvil y llega a un Módulo Bluetooth serial (figura 29), estos dos dispositivos son conectados a través de una dirección específica que contiene el módulo serial, por medio del SPP (puerto paralelo estándar) define cómo configurar los puertos serie virtuales y conectar dos dispositivos habilitados para Bluetooth del dispositivo móvil.

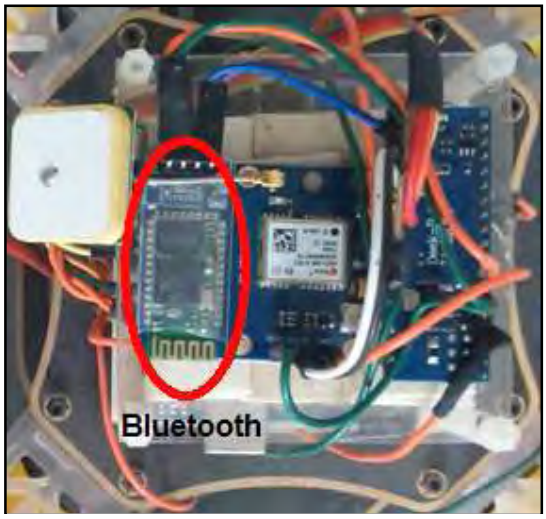



Figura 29.- BT montado en el cuadricóptero.

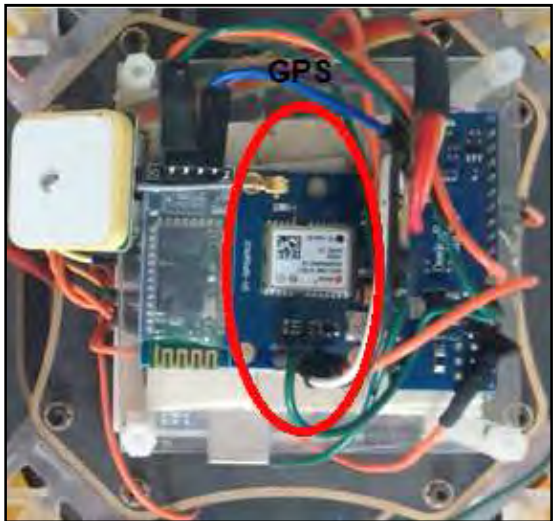



Figura 30.- GPS montado en el cuadricóptero.

### 5.2.1. Datos del GPS

Dicho lo anterior se describe la manera en que se desarrolló el algoritmo así como los criterios que se tomaron para la realización del mismo. Los datos que recibimos en nuestro módulo GPS (figura 30) siguen el protocolo NMEA (siglas de National Marine Electronics Asociation), las cuales son sentencias estándares para la recepción de datos GPS. Una de ellas y la más usada es la sentencia $GPRMC, con la cual se trabajo en este trabajo.

Como bien se mencionó anteriormente la sentencia $GPRMC es utilizada en este trabajo la cual tien como trama los siguientes caracteres $GPRMC,044235.000,A,4322.0289,N,00824.5210,W,0.39,65.46,020615,,,A*44, analizando la secuencia de este ejemplo y basándose en el protocolo NMEA, se determinan las siguientes variables:

- 044235.000 representa la hora GMT (04:42:35)
- “A” es la indicación de que el dato de posición está fijado y es correcto. “V” sería no válido
- 4322.0289 representa la longitud (43º 22.0289´)
- N representa el Norte
- 00824.5210 representa la latitud (8º 24.5210´)
- W representa el Oeste
- 0.39 representa la velocidad en nudos
- 65.46 representa la orientación en grados
- 020615 representa la fecha (2 de Junio del 2015)

Dentro del código implementado dentro de la placa Arduino Micro se hace la adquisición de los datos provenientes de GPS, dentro de las tramas se encuentra la $GPRMC, de la cual se extraen los datos que necesitamos, la siguiente figura muestra las secuencias provenientes de GPS, la que está indicada con una flecha en la figura 31 es la cual se utiliza únicamente.

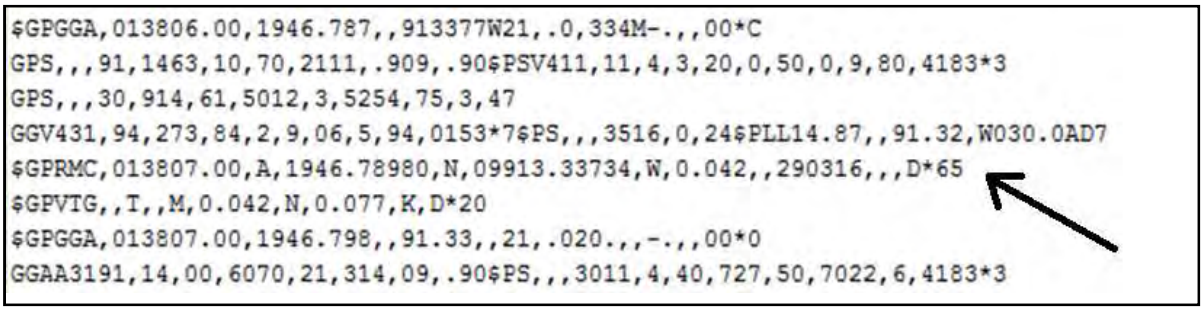



Figura 31.- Trama de datos de GPS.

Después de ver la trama de la cual se extraerán los datos para el posicionamiento, se debe saber que estos son tanto la longitud como la latitud.

La longitud mide el ángulo a lo largo del Ecuador desde cualquier punto de la Tierra. Se acepta que Greenwich en Londres es la longitud 0 en la mayoría de las sociedades modernas. Las líneas de longitud son círculos máximos que pasan por los polos y se llaman meridianos.

La latitud mide el ángulo entre cualquier punto y el ecuador .Las líneas de latitud se denominan paralelos. La latitudes el ángulo que existe entre un punto cualquiera y el Ecuador, medida sobre el meridiano que pasa por dicho punto. La distancia en km a la que equivale un grado de dichos meridianos depende de la latitud, a medida que la latitud aumenta disminuyen los kilómetros por grado.

### 5.2.2. Criterios para elaboración del código

Se revisó la longitud y la latitud de México, para poder determinar los criterios que se iban a seguir para el desarrollo del código de seguimiento, la figura 32 muestra las coordenadas extremas de la república mexicana.

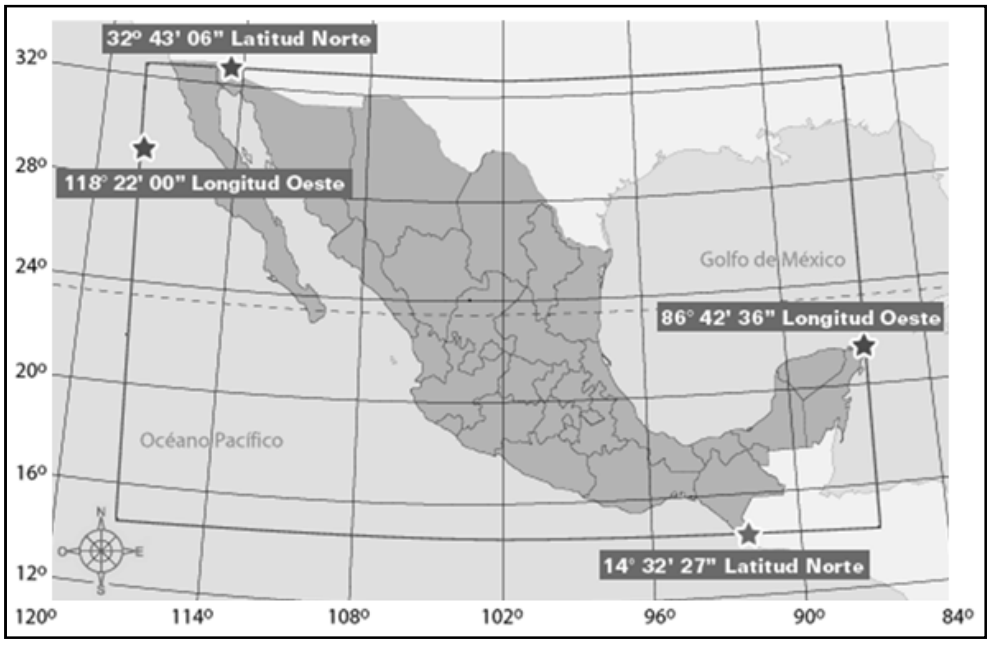



Figura 32.- Latitud y longitud extremas México.

Se observaron las coordenadas en las cuales se encuentra la república mexicana, en base a eso se traslado al plano como se ve en la figura 33 con medidas extremas en longitud oeste de 85° a 120° y latitud norte de 10° a 35°. Si se divide todo el plano de la tierra en cuatro cuadrantes la parte oeste la consideramos como parte negativa y la parte norte como positiva.

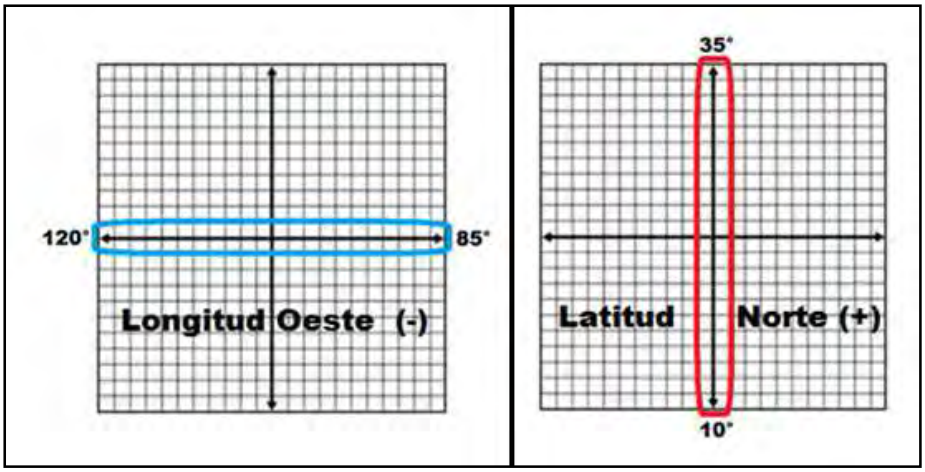



Figura 33.- Signo de latitud y longitud de México.

Igualmente se revisó la precisión que se iba a manejar, ver tablas 3 y 4. Posteriormente se realizaron mediciones de la exactitud del GPS de los dispositivos móviles como lo es el Xperia Z y el Xperia M2, para así despues revisar las medidas y observar la precisión que se debía utilizar, estas medidas se muestran a continuación de manera gráfica, observándose que se pude usar la precisión de cinco dígitos ya que las medidas son menores a los 11 metros de esta manera que está dentro del rango de distancia para tener una lectura nueva.

Las figuras 34 y 35 muestran las mediciones que se realizaron con los dispositivos móviles que se utilizaron, es estas se puede ver gráficamente que en ninguna de las medidas se encuentra una diferencia mayor a los 11 metros de distancia de la medida anterior.

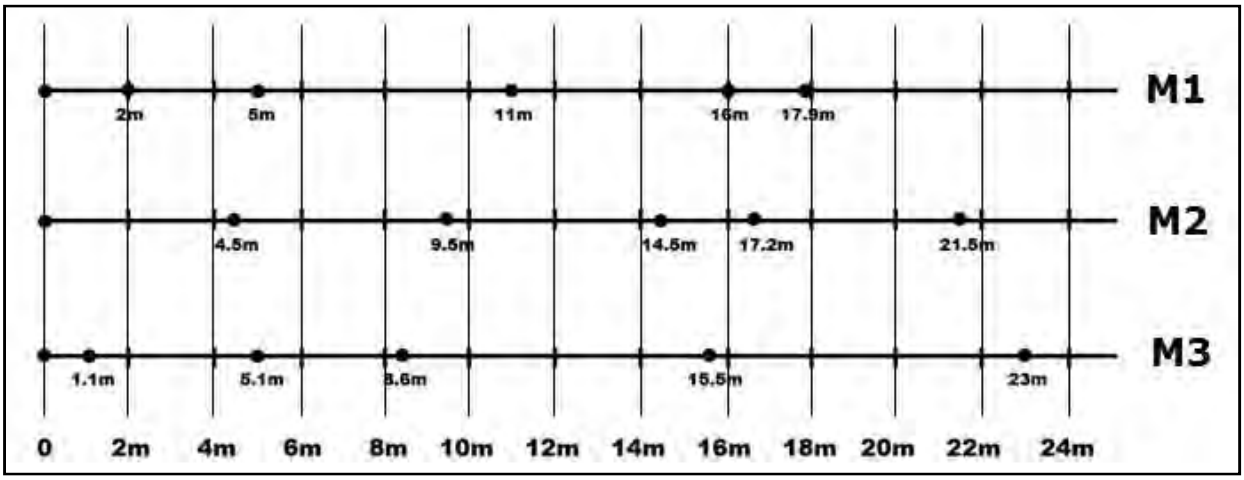



Figura 34.- Mediciones GPS Xperia Z.

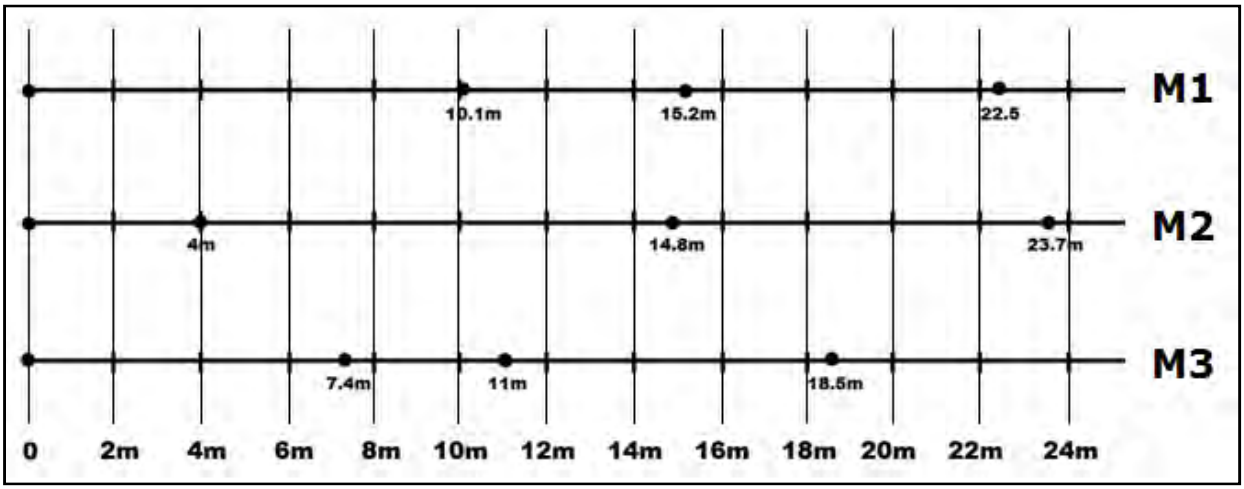



Figura 35.- Mediciones GPS Xperia M2.

Las figura 36, muestran que si solo se utilizara una de las dos medidas, ya sea longitud o latitud se tendría un problema el cual es que un mismo objeto el línea en longitud o en latitud tendría las mismas coordenadas aun cuando este se moviera dentro de esa línea, por tal motivo es que se utiliza tanto la longitud como la latitud para un posicionamiento adecuado.

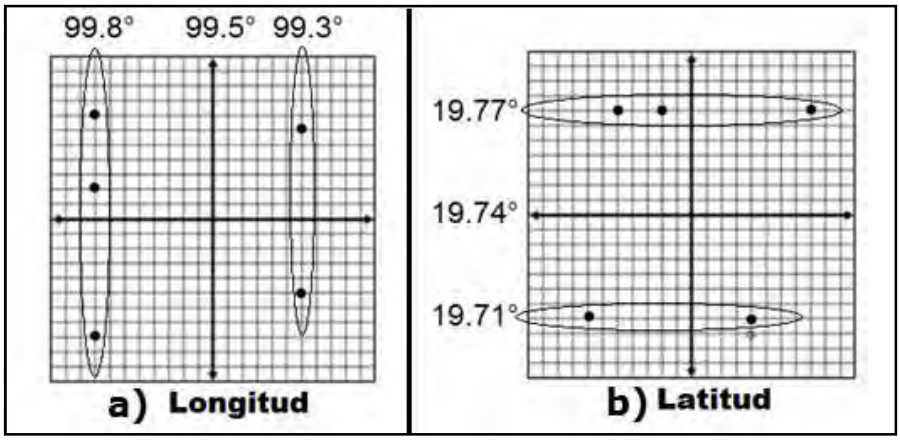



Figura 36.- Posición en latitud y longitud.

Habiendo visto todo lo anterior y teniendo en cuenta los criterios de los signos respecto a la poción de la república mexicana (ver figura 32), se tomó en consideración que como tanto la latitud como la longitud total está dentro de un mismo signo, para longitud (-) y para latitud (+), se tomó que para los cálculos relacionados con la diferencia de posicionamiento se modifican los datos proporcionados como negativos, para la longitud, y se pasan a positivos para su manejo.

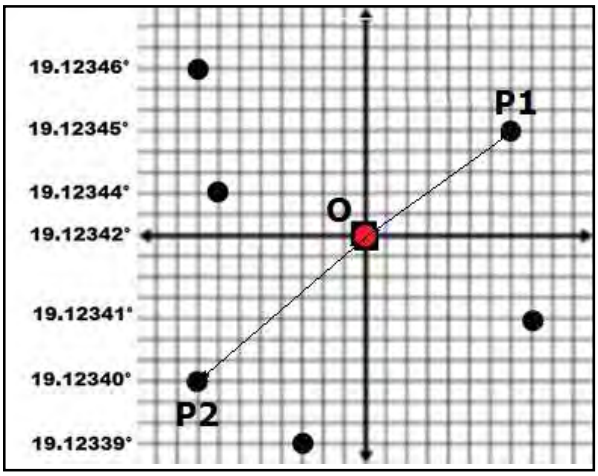



Figura 37.- Diferencia de posición.

Asi de esta manera al hacer las operaciones de diferencia de posicionamiento y teniendo como positivas todas las cantidades medidas el resultado de la substracción resulta positivo, y al momento de realizar la segunda operación se suma de la nueva posición y estas no se ven afectadas ya que ambas cantidades son positivas.


Al final de estas dos operaciones se realiza la comparación de la medida inicial de posición, con el resultado final de la substracción de la primera lectura y suma de la segunda, este resultado como se puede observar en las operaciones de arriba nos indican si el movimiento que se debe hacer es hacia enfrente o hacia atrás, izquierda o derecha según sea la operación con la latitud o con la longitud. Si el resultado final de las dos operaciones es menor a la medida inicial el movimiento será hacia atrás en latitud y hacia la izquierda en longitud y si el resultado final de las dos operaciones es mayor a la medida inicial es movimiento será hacia enfrente en latitud y derecha en longitud.

La expresión desarrollada para el seguimiento que realiza en cuadricóptero queda como:

$$D_n - C = R_1 \quad \rightarrow \quad R_1 + D_{n+1} = R_2 \quad \rightarrow \quad \boldsymbol{D_n - C + D_{n+1} = R_2}$$

Donde las siguientes comparaciones nos indican la acción a realizar en los motores:

$$\boldsymbol{D_n < R_2} \qquad \boldsymbol{D_n = R_2} \qquad \boldsymbol{D_n > R_2}$$

Donde:

$D_n$ = Posición inicial del dispositivo móvil

$C$ = Posición del cuadricóptero

$R_1$ =Resultado de la substracción de la posición inicial del dispositivo móvil y el cuadricóptero

$D_{n+1}$ = Posición siguiente del dispositivo móvil

$R_2$ = Resultado de la suma de R1 y D n+1

Este desarrollo del algoritmo se muestra en el diagrama de flujo de la figura 38.

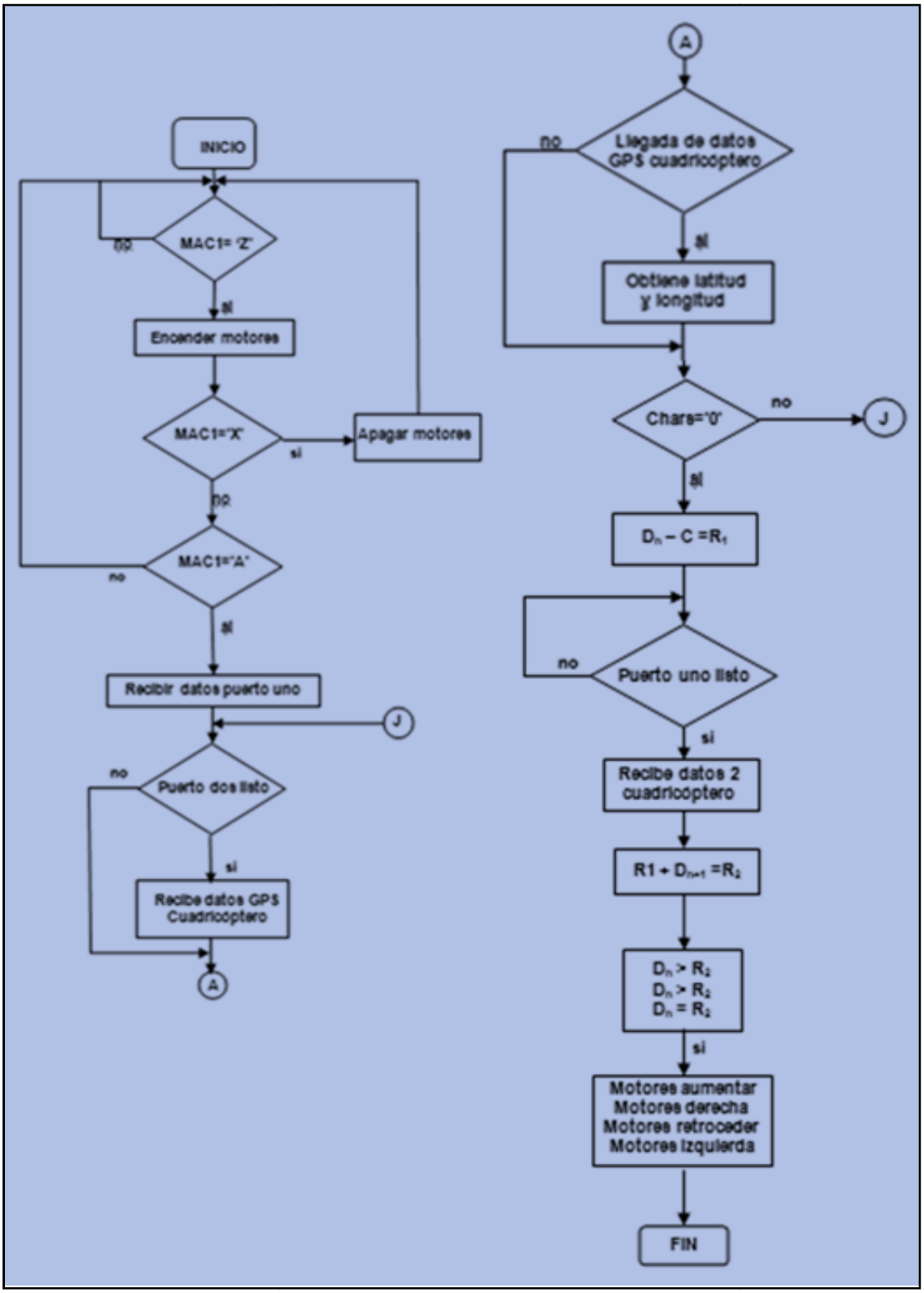



Figura 38.- Algoritmo de programa principal de seguimiento

La figura 39 muestra el diagrama con la secuencia que lleva el algoritmo del programa encargado del control de vuelo. Este código fue generado por Joop Brokking y se adquirió en [35].

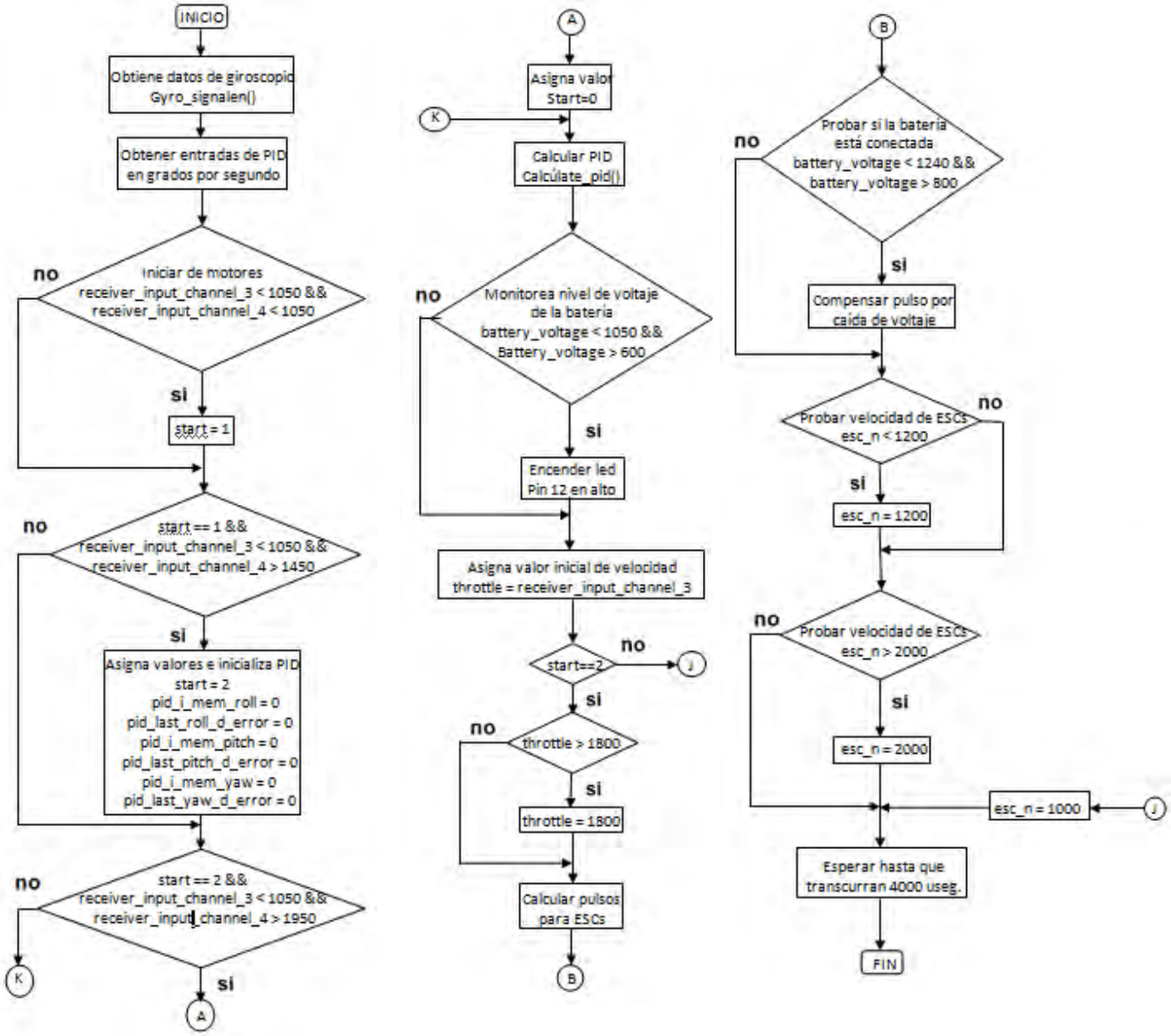



Figura 39.- Algoritmo de control de vuelo.

### 5.3. Secuencia de código Arduino Uno

El código comienza con una inicialización posteriormente se pasa a la parte del código conocida como *void loop()* en el cual se encuentra en primera instancia la instrucción *gyro_signalen()*, esta es la orden que envía a la subrutina para la obtención de datos del giróscopo. Regresando de la subrutina se presentan a la conversión de dato del giróscopo a grados por segundo para ser utilizados como entradas en el PID así como su ajuste del mismo.

Posteriormente se prueban las señales de entrada para verificar si estas corresponden a una instrucción de paro o arranque.

En seguida se localiza el PID este pone los puntos en grados por segundos esto es determinado por la entrada del receptor roll si se divide por 3 la velocidad máxima de roll es aproximadamente 164 grados por segundo (500-8)/3 = 164d/s). La instrucción *calculate_pid()*, envía a una subrutina para la obtención de datos del cálculo de las salidas del PID.

Dentro de la secuencia del código se encuentra una expresión que indica el monitoreo de carga de la batería, ya que el voltaje de la batería se necesita para la compensación de la velocidad de los motores y finalmente un filtro complementario es usado para reducir el ruido. Se tiene un led para señalización cuando este se prende indica que el voltaje de la batería es bajo.

En seguida se realiza el cálculo del pulso para cada ESC, cabe mencionar que se tiene la señal de trottle como señal base, para que nunca comience con cero. Luego se compensa el pulso de cada ESC por caída de voltaje

Se establece el periodo de trabajo para cada ciclo de operación, así también la frecuencia de actualización para cada uno de los pulsos del ESC, esta es de 4 ms.

Por último se encuentra la salida del pulso PWM del Arduino UNO de 1000 a 2000 micro segundos a los ESC, este genera una señal PWM (Modulación de ancho del pulso), que alimenta al motor brushles. Este a su vez, gira y genera otra fuerza llamada contra electromotriz o Back-emf. Esta señal (PWM) se alimenta a cada una de las fases del motor con un defasamiento de 120 grados. Para los motores, la señal se repite varias de veces por segundo.

La figura 40 se muestra el diagrama  con la secuencia que lleva el algoritmo del programa encargado del control de vuelo modificado para adaptarse a las necesidades requeridas.

.

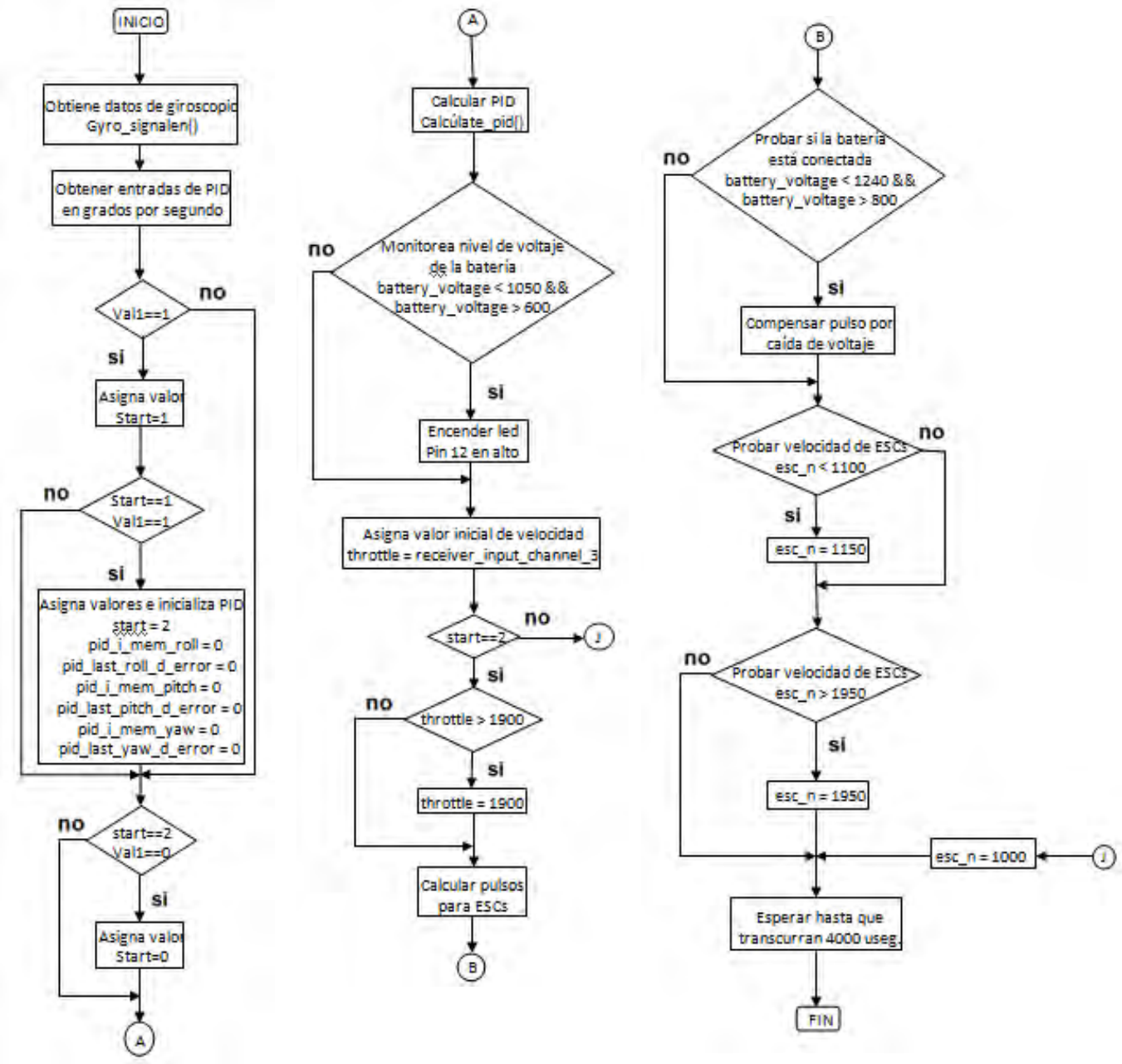



Figura 40.- Algoritmo de control de vuelo modificado.

Las figuras 41 y 42 muestran el diagrama con la secuencia que lleva el algoritmo de las interrupciones que se realizan cada vez que se modifica la instrucción de aumento o disminución de la velocidad de los motores.

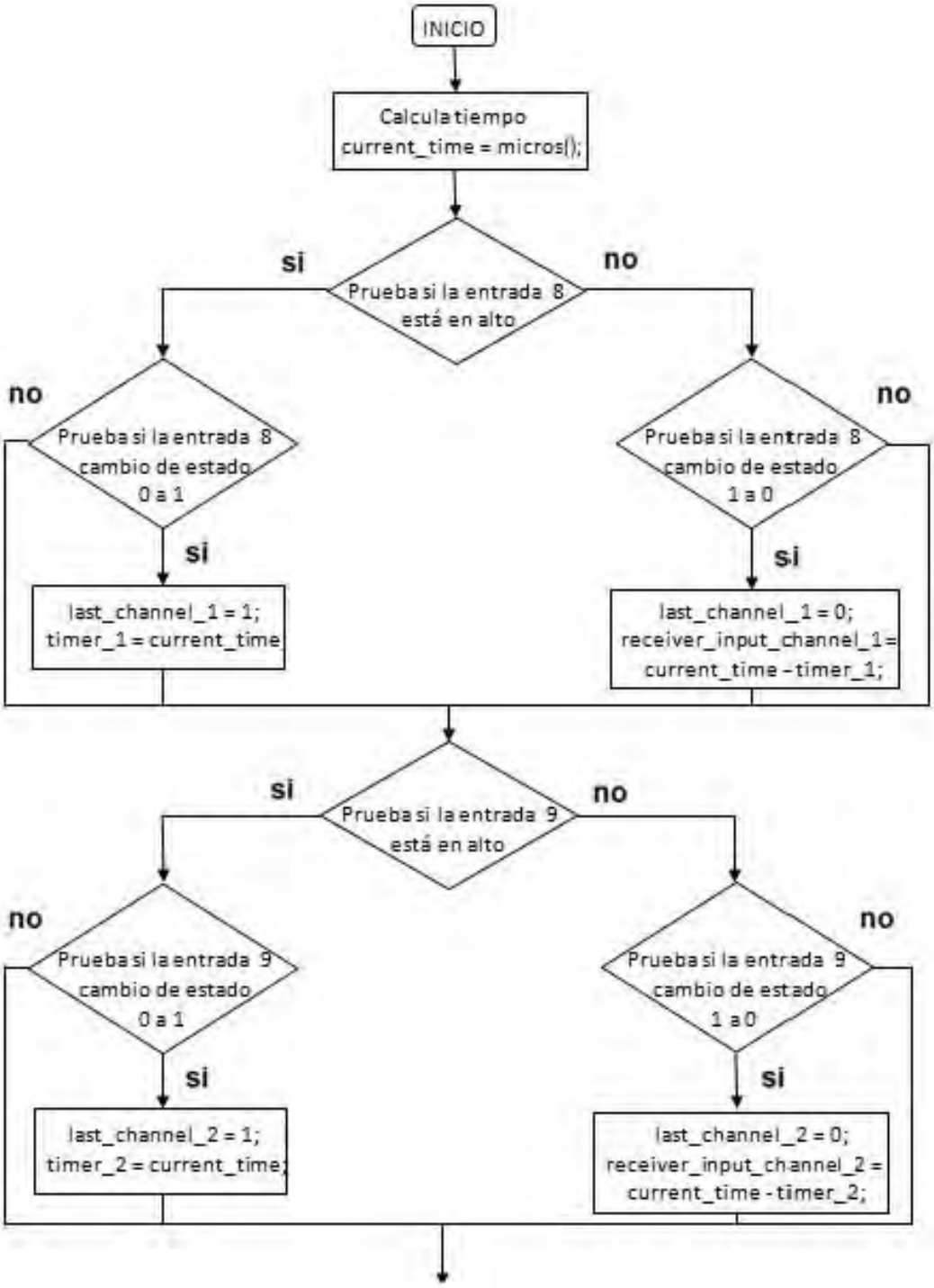



Figura 41.- Algóritmo de interrupción parte 1.

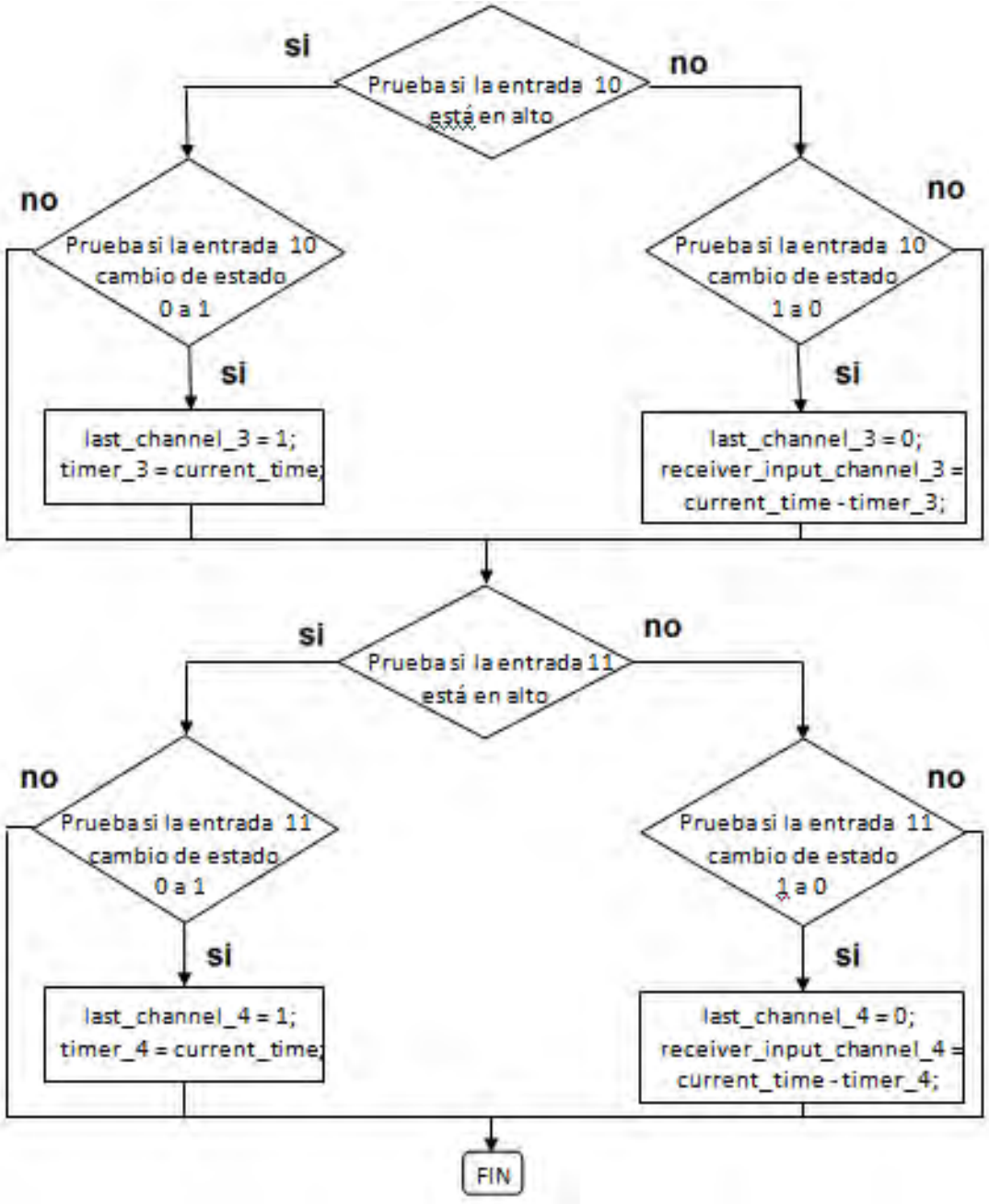



Figura 42.- Algóritmo de interrupción parte 2.

La figura 43 muestra el diagrama con la secuencia del algoritmo de la rutina *Calculate_PID ( )*.

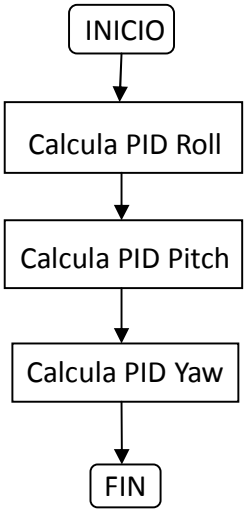



Figura 43.- Rutina *Calculate_PID ()*.

La figura 44 muestra el diagrama con la secuencia del algoritmo de la rutina *Gyro_signalen ( ).*

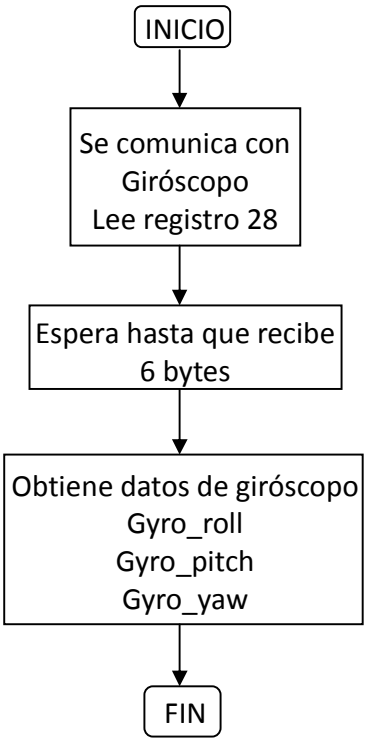



Figura 44.- Rutina *Gyro_signalen ()*.

## 5.4. Secuencia de código en Arduino Micro

Dentro del cuadricóptero se hizo el uso de dos placas Arduino para su funcionamiento, una de ellas es una placa Arduino Micro en la cual se realiza tanto la tarea de conexión con el dispositivo móvil, así como la trasferencia de la información de las instrucciones para el movimiento con la placa Arduino Uno, haciendo el trabajo de interfaz entre las instrucciones provenientes del dispositivo móvil y la ejecución de las instrucciones,

Dentro de esta placa se desarrolló y se implementó un código el cual verifica de manera continua la llegada de datos al puerto serial, posteriormente almacena los datos de llegada en la variable llamada *Mac1,2,3* siendo identificados como Z-armar, X-desarmar, T- trottle, R-roll, P-pitch, Y-yaw, inmediatamente realiza una comparación progresiva de los 3 datos provenientes del dispositivo móvil, teniendo caracterizado el dato hace el envió a una subrutina para que a continuación saque el dato resultante por el puerto que le corresponde, siendo M1-Trottle, M2-Roll, M3-Pitch, M4-yaw y finalmente vuelve a verificar si llegó otro dato o bien, limpia y regresa al inicio de la rutina en espera de datos.

Posteriormente se mantiene en un bucle hasta la indicación de entrar al programa para el seguimiento, si llega la instrucción y pasa al bucle siguiente, revisa si se ha dado la instrucción de parar y salir, dentro del mismo bucle y espera por los datos de entrada de la primer posición del dispositivo móvil.

Luego caracteriza los datos entrantes del GPS del dispositivo móvil, en seguida realiza la lectura del GPS En el cuadricóptero y decodifica la información, en seguida ve si hay datos nuevos y las guarda.

No obstante, de no haber realizado el paso anterior prueba y avisa si no llegaron datos nuevos del GPS y del cuadricóptero.

Habiendo obtenido la información necesaria se realiza la primera operación, esta consiste en restar lo asignado de los datos que llegaron y se caracterizaron del GPS del dispositivo móvil, al dato que se recopilo del GPS del cuadricóptero, tanto como para latitud como para longitud. Las siguientes imágenes muestran los datos enviados de la App y como llegan los datos a la placa Arduino Micro.

Posteriormente entra a otro bucle y espera por los datos de entrada de la segunda posición proveniente del dispositivo móvil, de esta manera nuevamente se realiza la caracterización de los datos provenientes de la segunda lectura del dispositivo móvil. En la siguiente figura se ve la espera del segundo dato y el resultado de la comparación final para la toma de decisión del movimiento.

Una vez obtenido y caracterizado los datos entrantes del GPS del dispositivo móvil por segunda vez, se realiza la segunda operación, esta es sumar los datos que llegaron y se caracterizaron del GPS del dispositivo móvil, al dato proveniente del GPS del cuadricóptero.

Ahora bien, con el resultado de la segunda operación se hace la comparación con el dato del GPS del cuadricóptero guardado en una variable y según el resultado de la comparación realizada se envía la instrucción aumentar, izquierda, derecha o retroceder. De tal modo, se realiza el cambio de velocidades en los motores para el seguimiento según la decisión tomada.

La figura 45 muestra el diagrama del algoritmo como programa para el seguimiento, el cual integra cada elemento, módulos, operaciones y procesos, el cual se programó en el entorno Arduino 1.6.0

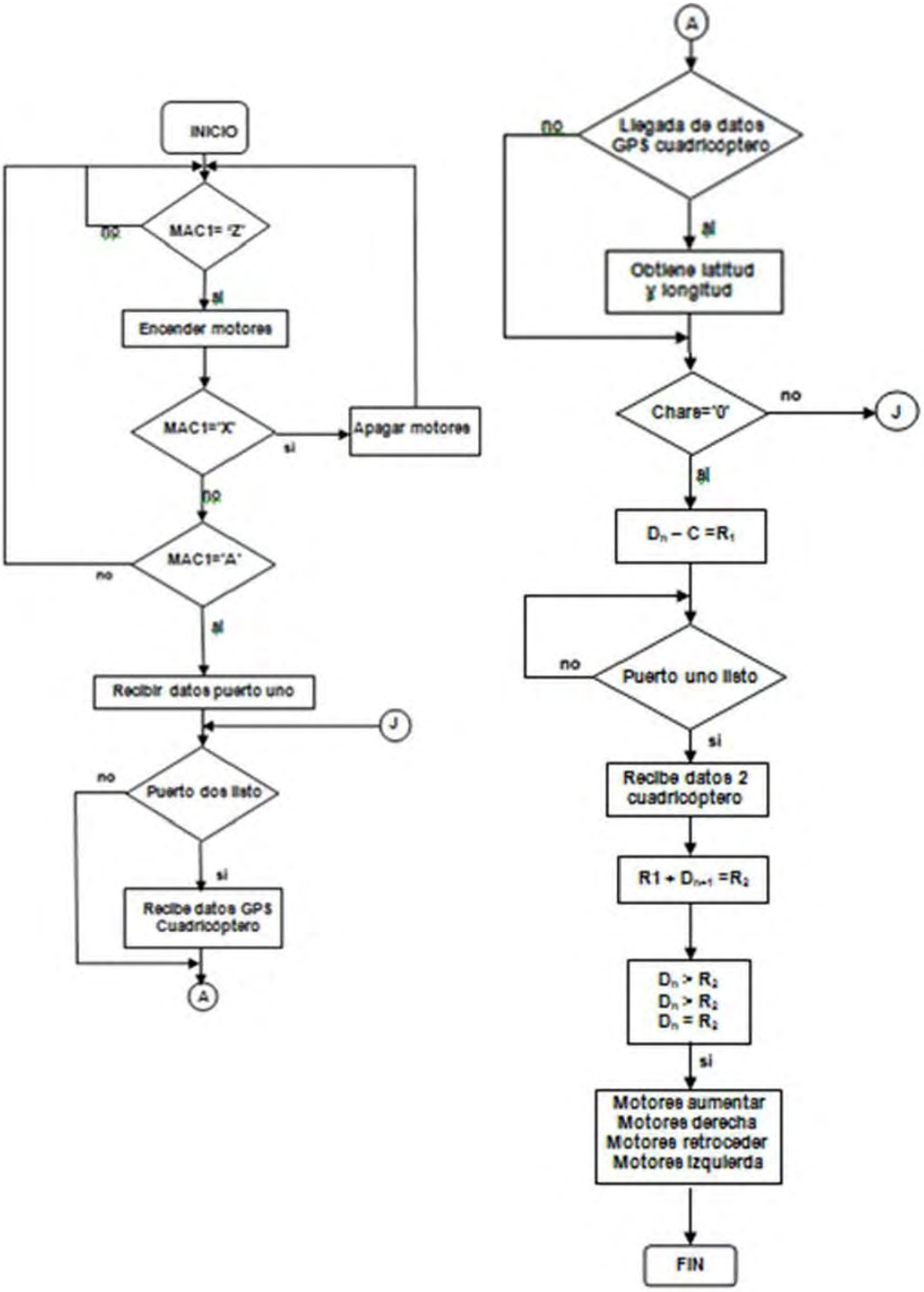



Figura 45.- Algoritmo del programa de seguimiento.

En las tablas se muestran los datos recopilados de las pruebas en campo abierto de la conexión de BT, entre el dispositivo móvil y el módulo en el cuadricóptero.

TABLA 5.- Tiempos para conexión Bluetooth dirección norte

| | 5m | 10m | 15m | 20m | 25m |
|---|---|---|---|---|---|
| Medida1 | 08:3 | 06:7 | 05:2 | 03:9 | 19:1** |
| Medida2 | 06:6 | 06:0 | 04:4 | 03:6 | 09:7** |
| Medida3 | 07:5 | 05:0 | 03:9 | 10:5* | 03:6* |
| Medida4 | 08:6 | 07:8 | 12:1* | 06:0 | 14:5** |
| Medida5 | 05:7 | 04:5 | 05:9 | 06:0 | 29:0** |

*dos intentos para conexión
** tres intentos para conexión

TABLA 6.- Tiempos para conexión Bluetooth dirección sur

| | 5m | 10m | 15m | 20m | 25m |
|---|---|---|---|---|---|
| Medida1 | 06:0 | 07:1 | 04:7 | 04:9 | 21:0** |
| Medida2 | 05:6 | 07:0 | 05:2 | 05:6 | 11:7** |
| Medida3 | 06:7 | 06:0 | 04:9 | 09:5* | 12:5** |
| Medida4 | 07:8 | 05:8 | 07:1 | 05:5 | 05:6* |
| Medida5 | 06:4 | 03:9 | 08:9 | 05:5 | 19:3** |

*dos intentos para conexión
** tres intentos para conexión

TABLA 7.- Tiempos para conexión Bluetooth dirección este

| | 5m | 10 | 15m | 20m | 25m |
|---|---|---|---|---|---|
| Medida1 | 05:4 | 05:9 | 06:9 | 04:9 | 16:0** |
| Medida2 | 05:8 | 05:8 | 07:4 | 05:7 | 14:7** |
| Medida3 | 07:2 | 07:0 | 05:6 | 07:3 | 14:5** |
| Medida4 | 05:1 | 07:6 | 05:2 | 06:5 | 15:6** |
| Medida5 | 07:2 | 05:3 | 06:2 | 05:5 | 12:8* |

*dos intentos para conexión
** tres intentos para conexión

TABLA 8.- Tiempos para conexión Bluetooth dirección oeste

| | 5m | 10m | 15m | 20m | 25m |
|---|---|---|---|---|---|
| Medida1 | 07:3 | 07:6 | 05:8 | 04:7 | 17:1** |
| Medida2 | 06:3 | 05:9 | 05:2 | 04:9 | 11:8** |
| Medida3 | 06:5 | 06:0 | 06:9 | 07:8 | 06:6* |
| Medida4 | 07:4 | 07:7 | 07:3 | 11:1* | 12:5** |
| Medida5 | 06:0 | 05:2 | 06:9 | 06:3 | 18:7** |

*dos intentos para conexión
** tres intentos para conexión

# *CAPÍTULO 6*

## 6.1. Resultados

Se integraron los elementos se realizó la programación para las pruebas con el cuadricóptero, con el código modificado (Anexo A), se diseñó una aplicación (Adonis 1.0) para dichas pruebas con la cual se verificó que reconociera las instrucciones que se le mandaban, con lo cual se observó en cada prueba que su reacción era de acuerdo a la instrucción que le indicaba.

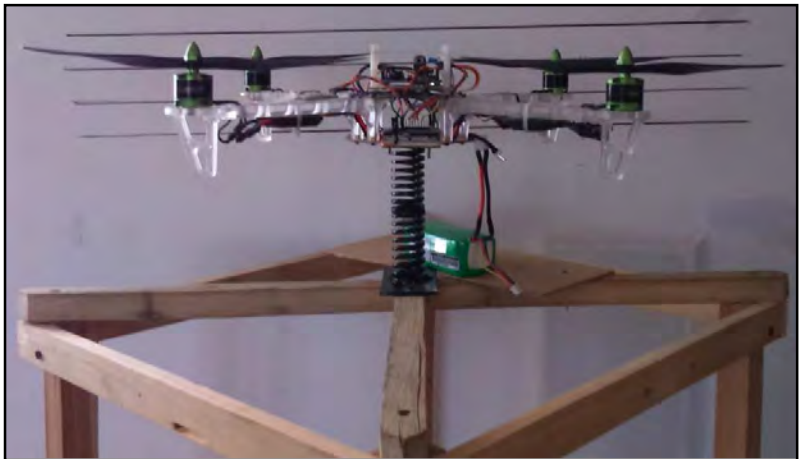


Figura 46.- Cuadricóptero sobre base de pruebas.

Posteriormente se realizaron las pruebas de conexión a distancia, en ástas se hicieron diversas medidas hacia los cuatro puntos cardinales, las pruebas se realizaron en campo abierto a diferentes distancias 5, 10, 15, 20, 25 m.


Figura 47.- Prueba conexión Bluetooth 5 metros.


Figura 48.- Prueba conexión Bluetooth 25 metros.

Estas pruebas dieron como resultado las gráficas mostradas en las figuras 49, 50, 51 y 52. En las cuales se puede observar el comportamiento de la conexión del BT en relación a la distancia entre los dos dispositivos a conectarse.

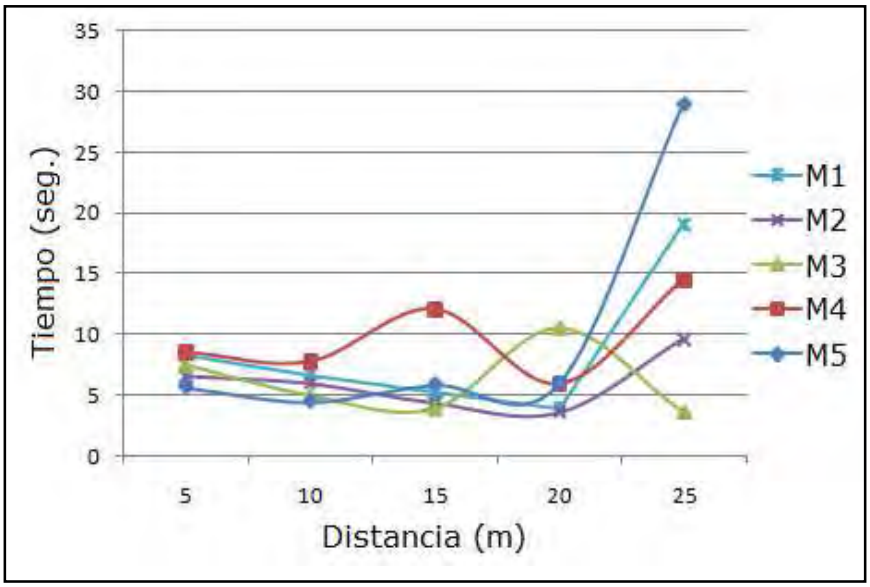



Figura 49.- Gráfica para conexión Bluetooth dirección norte.

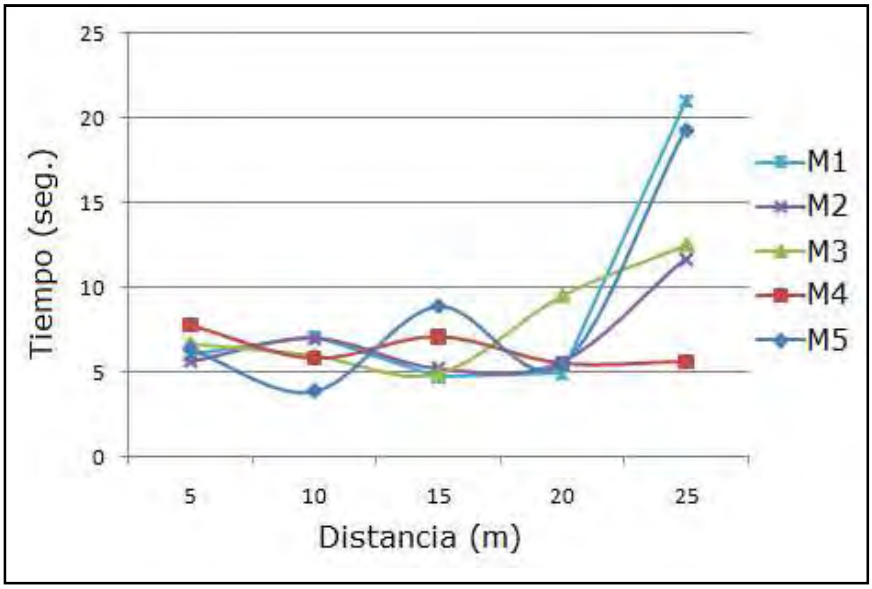



Figura 50.- Gráfica para conexión Bluetooth dirección sur.

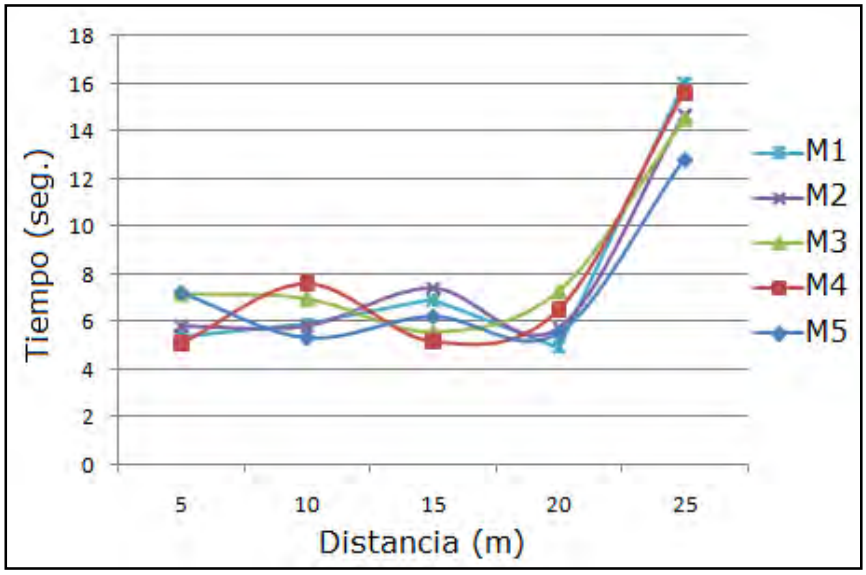



Figura 51.- Gráfica para conexión Bluetooth dirección este.

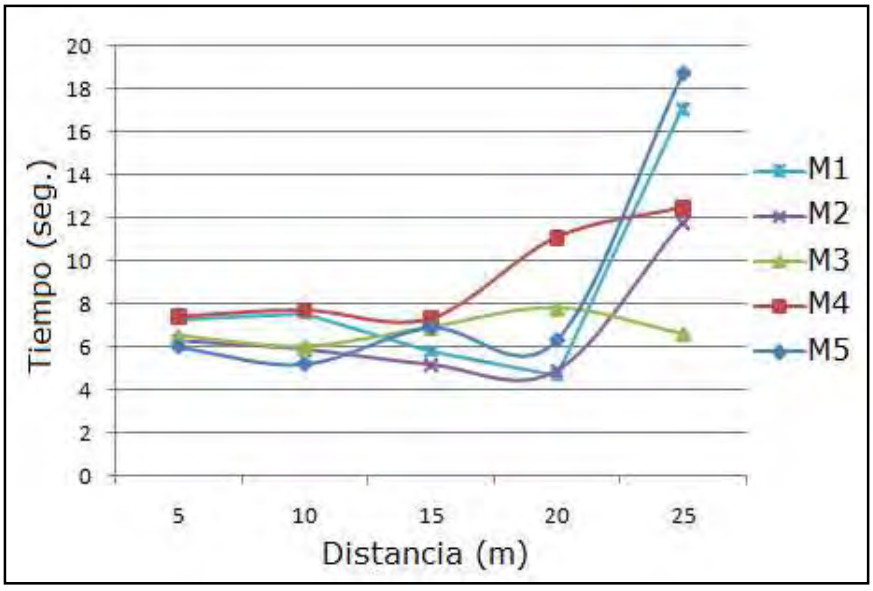



Figura 52.- Gráfica para conexión Bluetooth dirección oeste.

Se obtuvieron datos del GPS a bordo del cuadricóptero de diferentes puntos para comparar la exactitud de dichas lectura, con las lecturas observadas en la pantalla del dispositivo móvil.

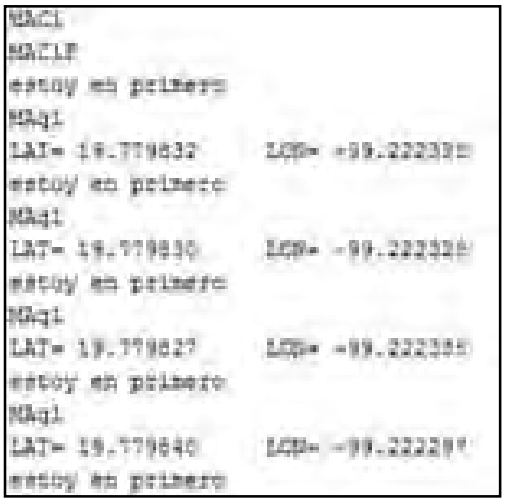


Figura 53.- Pantalla de datos de Latitud y Longitud

Se generó el código de seguimiento (Anexo B), se diseñó una y se generó una nueva aplicación (Adonis 2.0) para realizar las pruebas de seguimiento con la cual se verificó que realizara este, se observó en las diferentes pruebas que se ejecutaban las instrucciones para seguir al dispositivo móvil.

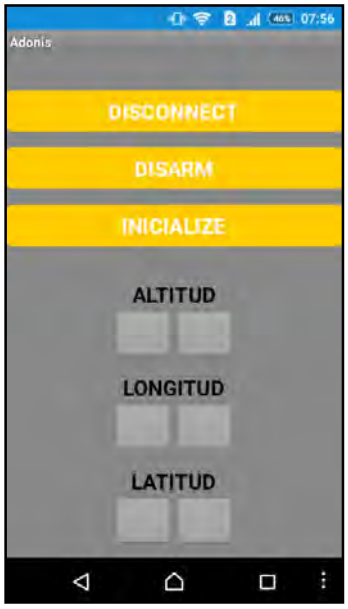



Figura 54.- Pantalla de la App.

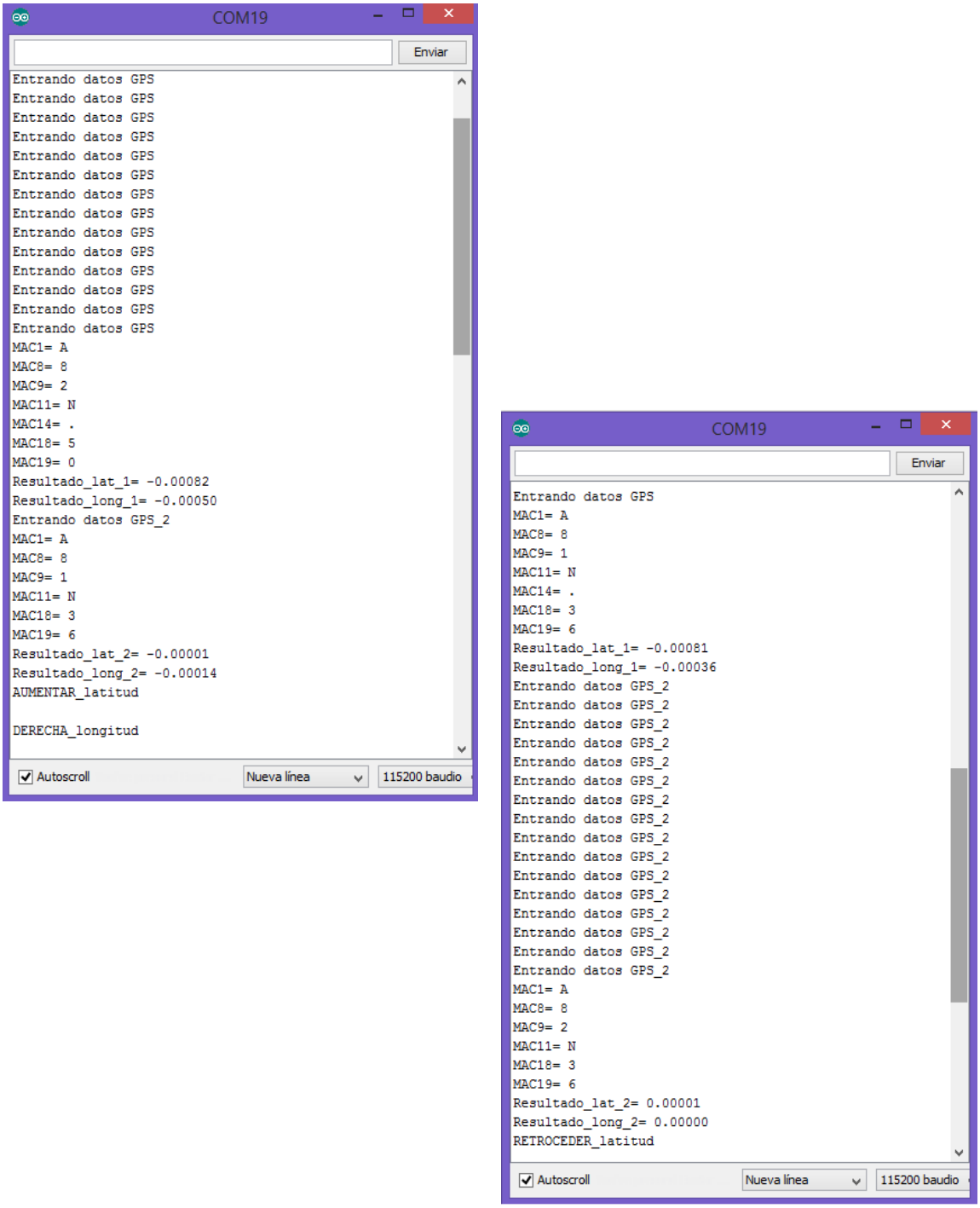



Figura 55.- Pantallas obtención de datos.

Para la realización de las pruebas de seguimiento se construyó una base con rodamientos sobre de ésta se fijó un resorte con dos placas a los extremos figura 56, la placa de abajo se utilizó para unir a la base rodante y la placa de arriba se usó para colocar el cuadricóptero. Las pruebas se realizaron en un área abierta con poca presencia de nubes, sobre una superficie plana.

Figura 56.- Cuadricóptero sobre base móvil.

## 6.2. Conclusiones

Se logró desarrollar un algoritmo de control para el seguimiento autónomo, el cual es complementado con la aplicación desarrollada para el sistema operativo de Android.

Se abordo la idea de que la tecnología era sinónimo de altos costos y complicadas operaciones de manejo (referente a “cuadricópteros con seguimiento autónomo”), además de que sólo personas con alto nivel económico podían acceder a ellos, sin embargo al paso de los años este cliché ha sido desechado ya que se percato de que es más que eso pues ha encontrado un nuevo significado y beneficios a la misma.

Fue por ello que surgió este trabajo pues aporta otra opción para el desarrollo de tecnología de sistemas similares. A diferencia de los drones que podemos encontrar en tiendas especializadas o sistemas con los que se comparo y cuyos costos son elevados, este cuadricóptero está conformado con plataformas libres (Android y Arduino) que permiten el acceso de cualquier persona, así también como una implementación sencilla.

De inicio se planteó generar el algoritmo para un cuadricóptero el cual tuviera seguimiento autónomo de manera inalámbrica, el resultado fue el esperado, se logró realizar el código para el seguimiento. Se generó el algoritmo de control con el cual el cuadricóptero realiza el seguimiento autónomo, de igual manera las pruebas de conexión inalámbrica dieron como resultados que el alcance de esta entre el cuadricóptero y el dispositivo móvil a seguir estuvo dentro del rango de 20 metros de distancia a la redonda de su objetivo.

Además la implementación del control del cuadricóptero por medio de una interfaz de transmisión de datos con un dispositivo móvil con sistema operativo Android, a través de comunicación inalámbrica se realizó con éxito y se mostró que tanto la conexión como la transmisión de los datos enviados desde el dispositivo móvil con S.O. Android se efectuaban satisfactoriamente.

Se dotó al cuadricóptero de señalizaciones indicadas a través del parpadeo un led así como de un algoritmo con el cual realiza toma de decisiones basado en sus registros de lecturas las cuales provienen de las lecturas generadas tanto del sistema de navegación (IMU) como del de posicionamiento (GPS).

Se desarrolló una App con la cual el cuadricóptero puede ser controlado a través de un dispositivo móvil con S.O. Android. Cabe mencionar que en un principio se generó una App la cual se utilizó para pruebas de funcionamiento y comportamiento del cuadricóptero desarrollándose en segunda instancia la aplicación con la cual se realiza el seguimiento.

## REFERENCIAS


[1] Luis Antonio Juárez García de León. (2012). Construcción y control de estabilidad de un cuadricóptero. México, CD. MX. UNAM, Facultad de Ingeniería.

[2] F. B. Hector Alberto, T. L. Isai Jhonatan, R. C. Ulises. (2016). Diseño, construcción y control de una aeronave tipo dron, CD.MX. UNAM, Facultad de Ingeniería.

[3] Rangel Vargas R. (2015). Implementación de un piloto automático para un UAV (Vehículo aéreo no tripulado), monitoreado y controlado por módulos de tecnología zigbee mediante una estación terrena. México, CD. MX. UNAM, Facultad de Ingeniería.

[4] RT (2012). Drone, historia de un arma de altos vuelos https://actualidad.rt.com/actualidad/view/80396-vehiculos-aereos-tripulados-hitos-historicos

[5] IRTIC (2015). Origen y desarrollo de los drones http://drones.uv.es/origen-y-desarrollo-de-los-drones/

[6] Zano, Septiembre 2015 http://www.flyzano.com/

[7] Lily Robotics, Inc., (2015). Lily - The Camera That Follows You, Septiembre 2015 https://www.lily.camera/

[8] Squadrone System, (2016) Hexo+, Febrero 2016 https://hexoplus.com/

[9] Teixeira J. M., Ferreira R., Santos M., Teichrieb V., (2014), Teleoperation Using Google Glass and AR.Drone for Structural Inspection. Brazil. IEEE., pp. 28-36.

[10] Kong L., Sheng J., Teredesai A., (2014). Basic Micro-Aerial Vehicles (MAVs) Obstacles Avoidance Using Monocular Computer Vision. USA. IEEE., pp.1051-1056.

[11] Madani T., Benallegue A., (2006).Backstepping Control for a Quadrotor Helicopter. Francia. IEEE., pp. 3255-3260.

[12] Xu R., Ozguner U., (2006). Sliding Mode Control of a Quadrotor Helicopter. USA. IEEE., pp. 4957-4962.

[13] Besnard L., Shtessel Y., Landrum B., (2007). Control of a Quadrotor Vehicle Using Sliding Mode Disturbance Observer. USA. IEEE., pp. 5230-5235.

[14] Luque-Vega L., Castillo-Toledo B., Loukianov A. G., (2012). Block Linearization Control of a Quadrotor Via Sliding Mode. Canada, IEEE., pp. 149-154.

[15] Voos H., (2006). Nonlinear State-Dependent Riccati Equation Control of a Quadrotor UAV. Alemania. IEEE.

[16] Dierks T., Jagannathan S.*, (2010).* Output Feedback Control of a Quadrotor UAV Using Neural Networks, EE.UU. IEEE., pp. 50-66.

[17] Mehranpour M., Mohammad A., Emamgholi O., Farrokhi M., (2013). A New Fuzzy Adaptive Control for a Quadrotor Flying Robot. Iran. IEEE.

[18] Lee K., Sol Kim H., Bae Park J., Ho Choi Y., (2012). Hovering Control of a Quadrotor. Corea. IEEE. Pp. 162-167.

[19] Goodarzi F., Lee D., Lee T, (2013). Geometric Nonlinear PID Control of a Quadrotor UAV on SE(3). Suiza. IEEE., pp. 3845-3850.

[20] Araar O. Aouf N., (2014). Full Linear Control of a Quadrotor UAV, LQ vs H∞. UK. IEEE., pp. 133-138.

[21] D. Mellinger, N. Michael, M. Kumar. (2012). Trajectory Generation and Control for Precise Aggressive Maneuvers with Quadrorotors. The international journal of Robotics, Vol. 31.

[22] Academia, (2015). Motor de corriente directa, Enero 2015. http://www.academia.edu/14266427/MOTOR_DE_CORRIENTE_DIRECTA_CD_Qu%C3%A9_es_un_motor_de_corriente_directa

[23] CMSimple , (2015). Motores Brushless, Febrero 2015 http://www.e-radiocontrol.com.ar/?Motores_Brushless

[24] Introducción al control del motor CC sin escobillas, Junio 2015. http://www.digikey.com.mx/es/articles/techzone/2013/mar/an-introduction-to-brushless-dc-motor-control

[25] Cuadricóptero, (2013). Hélices para cuadricóptero, Marzo 2015. http://cuadricoptero.net/helices-para-cuadricoptero/

[26] Multicoptero FPV, (2013). El variador (ESC), el motor brusless y la hélice a escoger. Marzo 2015. http://blog.multicopterofpv.com/2013/12/el-variador-esc-el-motor-brusless-y-la.html

[27] Hobbyking. Speed Controllers (ESC). Abril 2015. https://hobbyking.com/hobbyking/store/__453__182__Speed_Controllers_ESC_-20_30A.html

[28] Hobbyking. Baterías y accesorios. Marzo 2015.
http://www.hobbyking.com/hobbyking/store/__545__85__Batteries_Accessories-ZIPPY_Compact.html

[29] Xabier Legasa Martín-Gil (2012). Cuadricóptero Arduino por control Remoto. España. Facultad de Informática de Barcelona (FIB)

[30] Arduino. Arduino uno. Noviembre 2015.
https://www.arduino.cc/en/Main/ArduinoBoardUno

[31] Arduino. Arduino micro. Noviembre 2015.
https://www.arduino.cc/en/Main/ArduinoBoardMicro

[32] Irving, M.Garcìa. (2014). Activación y control vía inalámbrica de equipos eléctricos a través de un dispositivo móvil con sistema operativo android, utilizando un microcontrolador de tipo pic. Cuautitlan, Mexico. UNAM, FES-C.

[33] Presicion y Margen de Error - gps america. Agosto 2015.
http://soluciones.gpsamerica.com.mx/gps/index.php?option=com_content&view=article&id=59&Itemid=390

[34] INEGI. Sistema de Posicionamiento Global. Agosto 2015.
http://www.inegi.org.mx/geo/contenidos/geodesia/gps.aspx?dv=c1

[35] Joop Brokking (2015). YMFC-3D part 5 – Quadcopter PID controller and PID tuning.
https://www.youtube.com/watch?v=JBvnB0279-Q

[36] DDCstaff-Fer (2015). PID para Cuadricópteros ¿Que es?, Marzo 2015
http://dronesdecarreras.com/pid-para-cuadricopteros-que-es/

# APÉNDICE A

## CÓDIGO ARDUINO UNO

```
#include <Wire.h> //Include the Wire.h library so we can communicate with the gyro.

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//PID gain and limit settings (ajustes de limites)
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
float pid_p_gain_roll = 1.3;               //Gain setting for the roll P-controller (1.3)
float pid_i_gain_roll = 0.05 ;             //Gain setting for the roll I-controller (0.3)
float pid_d_gain_roll = 15;                //Gain setting for the roll D-controller (15)
int pid_max_roll = 400;                    //Maximum output of the PID-controller (+/-)

float pid_p_gain_pitch = pid_p_gain_roll;  //Gain setting for the pitch P-controller.
float pid_i_gain_pitch = pid_i_gain_roll;  //Gain setting for the pitch I-controller.
float pid_d_gain_pitch = pid_d_gain_roll;  //Gain setting for the pitch D-controller.
int pid_max_pitch = pid_max_roll;          //Maximum output of the PID-controller (+/-)

float pid_p_gain_yaw = 4.0;                //Gain setting for the pitch P-controller. //4.0
float pid_i_gain_yaw = 0.02;               //Gain setting for the pitch I-controller. //0.02
float pid_d_gain_yaw = 0.0;                //Gain setting for the pitch D-controller.
int pid_max_yaw = 400;                     //Maximum output of the PID-controller (+/-)

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//Declaring Variables
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
byte last_channel_1, last_channel_2, last_channel_3, last_channel_4;
int receiver_input_channel_1, receiver_input_channel_2, receiver_input_channel_3, receiver_input_channel_4;
int counter_channel_1, counter_channel_2, counter_channel_3, counter_channel_4, loop_counter;
int esc_1, esc_2, esc_3, esc_4;
int throttle, battery_voltage;
unsigned long timer_channel_1, timer_channel_2, timer_channel_3, timer_channel_4, esc_timer,
esc_loop_timer;
unsigned long zero_timer, timer_1, timer_2, timer_3, timer_4, current_time;
int cal_int;
unsigned long loop_timer,loop_timer2;
double gyro_pitch, gyro_roll, gyro_yaw;
double gyro_roll_cal, gyro_pitch_cal, gyro_yaw_cal;
byte highByte, lowByte;

char MAC2;
int val1=0;          // variable to store the read value
int start=0;

float pid_error_temp;
float pid_i_mem_roll, pid_roll_setpoint, gyro_roll_input, pid_output_roll, pid_last_roll_d_error;
float pid_i_mem_pitch, pid_pitch_setpoint, gyro_pitch_input, pid_output_pitch, pid_last_pitch_d_error;
float pid_i_mem_yaw, pid_yaw_setpoint, gyro_yaw_input, pid_output_yaw, pid_last_yaw_d_error;

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//Setup routine (instalacion)
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
void setup(){
  Wire.begin();

  pinMode(3, INPUT);
  pinMode(4, OUTPUT);          //Configure digital poort 4, 5, 6 and 7 as output.
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(7, OUTPUT);
```

```
  pinMode(8, INPUT);
  pinMode(9, INPUT);
  pinMode(10, INPUT);
  pinMode(11, INPUT);
  pinMode(12, OUTPUT);
  pinMode(13, OUTPUT);  //Configure digital poort 12 and 13 as output.

  //Use the led on the Arduino for startup indication
  digitalWrite(12,HIGH);
  digitalWrite(13,HIGH);
  delay(1000);
  digitalWrite(13,LOW);
  digitalWrite(12,LOW);

  Wire.beginTransmission(105);
  Wire.write(0x20);
  Wire.write(0x0F);
  Wire.endTransmission();
  Wire.beginTransmission(105);
  Wire.write(0x23);
  Wire.endTransmission();

   delay(250);

   //Let's take multiple gyro data samples so we can determine the average gyro offset (calibration).
   for (cal_int = 0; cal_int < 2000 ; cal_int ++){              //Take 2000 readings for calibration.
    if(cal_int % 15 == 0)digitalWrite(12, !digitalRead(12));
    gyro_signalen();          //Read the gyro output.
    gyro_roll_cal += gyro_roll;
    gyro_pitch_cal += gyro_pitch;
    gyro_yaw_cal += gyro_yaw;

   digitalWrite(4, HIGH);       //Set digital poort 4, 5, 6 and 7 high.
   digitalWrite(5, HIGH);
   digitalWrite(6, HIGH);
   digitalWrite(7, HIGH);

   delayMicroseconds(1000);           //Wait 1000us.

   digitalWrite(4, LOW);        //Set digital poort 4, 5, 6 and 7 low.
   digitalWrite(5, LOW);
   digitalWrite(6, LOW);
   digitalWrite(7, LOW);

   delay(3);
  }

  digitalWrite(12,LOW);

 //Now that we have 2000 measures, we need to devide by 2000 to get the average gyro offset.
    gyro_roll_cal /= 2000;                      //Divide the roll total by 2000.
    gyro_pitch_cal /= 2000;                     //Divide the pitch total by 2000.
    gyro_yaw_cal /= 2000;                       //Divide the yaw total by 2000.

  PCICR |= (1 << PCIE0);              //Set PCIE0 to enable PCMSK0 scan.
  PCMSK0 |= (1 << PCINT0);        //Set PCINT0 (digital input 8) to trigger an interrupt on state change.
  PCMSK0 |= (1 << PCINT1);
  PCMSK0 |= (1 << PCINT2);
  PCMSK0 |= (1 << PCINT3);
 ////////////////////////////////////////////////////////////////////////////////////
 ////////////////////////////////////////////////////////////////////////////////////
```

```
  //Load the battery voltage to the battery_voltage variable.
  //65 is the voltage compensation for the diode.
  //12.6V equals ~5V @ Analog 0.
  //12.6V equals 1023 analogRead(0).
  //1260 / 1023 = 1.2317.
  //The variable battery_voltage holds 1050 if the battery voltage is 10.5V.
  battery_voltage = (analogRead(0) + 66) * 1.2317;

 }

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
                                          //Main program loop
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

void loop(){

  digitalWrite(12,HIGH);
  digitalWrite(13,HIGH);
  //Let's get the current gyro data and scale it to degrees per second for the pid calculations.
  gyro_signalen();
  gyro_roll_input = (gyro_roll_input * 0.8) + ((gyro_roll / 57.14286) * 0.2);          //Gyro pid input is deg/sec.
  gyro_pitch_input = (gyro_pitch_input * 0.8) + ((gyro_pitch / 57.14286) * 0.2);
  gyro_yaw_input = (gyro_yaw_input * 0.8) + ((gyro_yaw / 57.14286) * 0.2);

////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////////////////
 INICIAR_MOTORES:

//For starting the motors: throttle low and yaw left (step 1).
   val1 = digitalRead(3);

   if(val1==0){start = 1;}

   if(start == 1 && val1==1){
    start = 2;

 //Reset the pid controllers for a bumpless start.
    pid_i_mem_roll = 0;
    pid_last_roll_d_error = 0;
    pid_i_mem_pitch = 0;
    pid_last_pitch_d_error = 0;
    pid_i_mem_yaw = 0;
    pid_last_yaw_d_error = 0;
   }

////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////////////////
  PARAR_MOTORES:

  //Stopping the motors: throttle low and yaw right.
  if(start == 2 && val1==0){start = 0;}

  //The PID set point in degrees per second is determined by the ROLL receiver input.

  //In the case of deviding by 3 the max roll rate is aprox 164 degrees per second ( (500-8)/3 = 164d/s ).
  pid_roll_setpoint = 0;

  //We need a little dead band of 16us for better results.
  if(receiver_input_channel_1 > 1500)pid_roll_setpoint = (receiver_input_channel_1 - 1500)/3.0;
  else if(receiver_input_channel_1 < 1500)pid_roll_setpoint = (receiver_input_channel_1 - 1500)/3.0;
```

```
//The PID set point in degrees per second is determined by the PITCH receiver input.

//In the case of deviding by 3 the max pitch rate is aprox 164 degrees per second ( (500-8)/3 = 164d/s ).
pid_pitch_setpoint = 0;

//We need a little dead band of 16us for better results.
if(receiver_input_channel_2 > 1500)pid_pitch_setpoint = (receiver_input_channel_2 - 1500)/3.0;
else if(receiver_input_channel_2 < 1500)pid_pitch_setpoint = (receiver_input_channel_2 - 1500)/3.0;

//The PID set point in degrees per second is determined by the YAW receiver input.

//In the case of deviding by 3 the max yaw rate is aprox 164 degrees per second ( (500-8)/3 = 164d/s ).
pid_yaw_setpoint = 0;

//We need a little dead band of 16us for better results.
if(receiver_input_channel_3 > 1050){ //Do not yaw when turning off the motors.
  if(receiver_input_channel_4 > 1500)pid_yaw_setpoint = (receiver_input_channel_4 - 1500)/3.0;
  else if(receiver_input_channel_4 < 1500)pid_yaw_setpoint = (receiver_input_channel_4 - 1500)/3.0;
}
////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////////////////

 //PID inputs are known. So we can calculate the pid output.
calculate_pid();

////////////////////////////////////////////////////////////////////////////////////////////////////
//The battery voltage is needed for compensation.
//A complementary filter is used to reduce noise.
//0.09853 = 0.08 * 1.2317.
battery_voltage = battery_voltage * 0.92 + (analogRead(0) + 66) * 0.09853;

//Turn on the led if battery voltage is to low.
if(battery_voltage < 1050 && battery_voltage > 600)digitalWrite(12, HIGH);

////////////////////////////////////////////////////////////////////////////////////////////////////
throttle = receiver_input_channel_3;                                    //We need the throttle signal as a base signal.

if (start == 2){                                                        //The motors are started.
  if (throttle > 1800) throttle = 1800;                                 //We need some room to keep full control at full throttle.
  //Calculate the pulse for esc 1 (front-right - CCW)
  esc_1 = throttle - pid_output_pitch + pid_output_roll - pid_output_yaw;
  //Calculate the pulse for esc 2 (rear-right - CW)
  esc_2 = throttle + pid_output_pitch + pid_output_roll + pid_output_yaw;
  //Calculate the pulse for esc 3 (rear-left - CCW)
  esc_3 = throttle + pid_output_pitch - pid_output_roll - pid_output_yaw;
  //Calculate the pulse for esc 4 (front-left - CW)
  esc_4 = throttle - pid_output_pitch - pid_output_roll + pid_output_yaw;

  if (battery_voltage < 1240 && battery_voltage > 800){                 //Is the battery connected?
    esc_1 += esc_1 * ((1240 - battery_voltage)/(float)3500);    //Compensate the esc-1 pulse for voltage drop.
    esc_2 += esc_2 * ((1240 - battery_voltage)/(float)3500);    //Compensate the esc-2 pulse for voltage drop.
    esc_3 += esc_3 * ((1240 - battery_voltage)/(float)3500);    //Compensate the esc-3 pulse for voltage drop.
    esc_4 += esc_4 * ((1240 - battery_voltage)/(float)3500);    //Compensate the esc-4 pulse for voltage drop.
  }

  if (esc_1 < 1100) esc_1 = 1150;                                       //Keep the motors running.
  if (esc_2 < 1100) esc_2 = 1150;                                       //Keep the motors running.
  if (esc_3 < 1100) esc_3 = 1150;                                       //Keep the motors running.
  if (esc_4 < 1100) esc_4 = 1150;                                       //Keep the motors running.

  if(esc_1 > 1950)esc_1 = 1950;                                         //Limit the esc-1 pulse to 2000us.
```

```
    if(esc_2 > 1950)esc_2 = 1950;                                 //Limit the esc-2 pulse to 2000us.
    if(esc_3 > 1950)esc_3 = 1950;                                 //Limit the esc-3 pulse to 2000us.
    if(esc_4 > 1950)esc_4 = 1950;                                 //Limit the esc-4 pulse to 2000us.
  }

  else{
    esc_1 = 1000;                                           //If start is not 2 keep a 1000us pulse for ess-1.
    esc_2 = 1000;                                           //If start is not 2 keep a 1000us pulse for ess-2.
    esc_3 = 1000;                                           //If start is not 2 keep a 1000us pulse for ess-3.
    esc_4 = 1000;                                           //If start is not 2 keep a 1000us pulse for ess-4.
  }

  //The refresh rate is 250Hz. That means the esc's need there pulse every 4ms.
  while(micros() - loop_timer < 4000);          //We wait until 4000us are passed.
  loop_timer = micros();                        //Set the timer for the next loop.
  //loop_timer2 = micros() - loop_timer; = 4004 diferencia del loop de arriba

  digitalWrite(4, HIGH);              //Set digital outputs 4,5,6 and 7 high.   PORTD
  digitalWrite(5, HIGH);
  digitalWrite(6, HIGH);
  digitalWrite(7, HIGH);

 //Calculate the time of the faling edge of the esc-1 pulse.
  timer_channel_1 = esc_1 + loop_timer; // = 1000 + 4000
 //Calculate the time of the faling edge of the esc-2 pulse.
  timer_channel_2 = esc_2 + loop_timer;
 //Calculate the time of the faling edge of the esc-3 pulse.
   timer_channel_3 = esc_3 + loop_timer;
 //Calculate the time of the faling edge of the esc-4 pulse.
   timer_channel_4 = esc_4 + loop_timer;

   while(PORTD >= 16){                      //Stay in this loop until output 4,5,6 and 7 are low.
     esc_loop_timer = micros();             //Read the current time.
     //Set digital output 4 to low if the time is expired.
     if(timer_channel_1 <= esc_loop_timer)PORTD &= B11101111;
     //Set digital output 5 to low if the time is expired.
     if(timer_channel_2 <= esc_loop_timer)PORTD &= B11011111;
     //Set digital output 6 to low if the time is expired.
     if(timer_channel_3 <= esc_loop_timer)PORTD &= B10111111;
     //Set digital output 7 to low if the time is expired.
     if(timer_channel_4 <= esc_loop_timer)PORTD &= B01111111;
   }
 }

 ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
 //This routine is called every time input 8, 9, 10 or 11 changed state
 ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
 ISR(PCINT0_vect){
   current_time = micros();
   //Channel 1=========================================
   if(PINB & B00000001){               //Is input 8 high? PINB- lee todo el puerto es solo lectura su funcion
     if(last_channel_1 == 0){          //Input 8 changed from 0 to 1
       last_channel_1 = 1;             //Remember current input state
       timer_1 = current_time;         //Set timer_1 to current_time
     }
   }
   else if(last_channel_1 == 1){                    //Input 8 is not high and changed from 1 to 0
     last_channel_1 = 0;                            //Remember current input state
     receiver_input_channel_1 = current_time - timer_1;        //Channel 1 is current_time - timer_1
   }
   //Channel 2=========================================
   if(PINB & B00000010 ){                           //Is input 9 high?
```

```
    if(last_channel_2 == 0){                                //Input 9 changed from 0 to 1
      last_channel_2 = 1;                                   //Remember current input state
      timer_2 = current_time;                               //Set timer_2 to current_time
    }
  }
  else if(last_channel_2 == 1){                             //Input 9 is not high and changed from 1 to 0
    last_channel_2 = 0;                                     //Remember current input state
    receiver_input_channel_2 = current_time - timer_2;      //Channel 2 is current_time - timer_2
  }
  //Channel 3=============================================
  if(PINB & B00000100 ){                                    //Is input 10 high?
    if(last_channel_3 == 0){                                //Input 10 changed from 0 to 1
      last_channel_3 = 1;                                   //Remember current input state
      timer_3 = current_time;                               //Set timer_3 to current_time
    }
  }
  else if(last_channel_3 == 1){                             //Input 10 is not high and changed from 1 to 0
    last_channel_3 = 0;                                     //Remember current input state
    receiver_input_channel_3 = current_time - timer_3;      //Channel 3 is current_time - timer_3

  }
  //Channel 4=============================================
  if(PINB & B00001000 ){                                    //Is input 11 high?
    if(last_channel_4 == 0){                                //Input 11 changed from 0 to 1
      last_channel_4 = 1;                                   //Remember current input state
      timer_4 = current_time;                               //Set timer_4 to current_time
    }
  }
  else if(last_channel_4 == 1){                             //Input 11 is not high and changed from 1 to 0
    last_channel_4 = 0;                                     //Remember current input state
    receiver_input_channel_4 = current_time - timer_4;      //Channel 4 is current_time - timer_4
  }
}
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//Subroutine for reading the gyro
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
void gyro_signalen(){
  Wire.beginTransmission(105);          //Start communication with the gyro (adress 1101001)
  Wire.write(168);              //Start reading @ register 28h and auto increment with every read
  Wire.endTransmission();               //End the transmission
  Wire.requestFrom(105, 6);             //Request 6 bytes from the gyro
  while(Wire.available() < 6);          /Wait until the 6 bytes are received
  lowByte = Wire.read();                //First received byte is the low part of the angular data
  highByte = Wire.read();               //Second received byte is the high part of the angular data
  gyro_roll = ((highByte<<8)|lowByte);      //Multiply highByte by 256 (shift left by 8) and ad lowByte
  if(cal_int == 2000)gyro_roll -= gyro_roll_cal;    //Only compensate after the calibration
  lowByte = Wire.read();                            //First received byte is the low part of the angular data
  highByte = Wire.read();                           //Second received byte is the high part of the angular data
  gyro_pitch = ((highByte<<8)|lowByte);             //Multiply highByte by 256 (shift left by 8) and ad lowByte
  gyro_pitch *= -1;                                 //Invert axis
  if(cal_int == 2000)gyro_pitch -= gyro_pitch_cal;  //Only compensate after the calibration
  lowByte = Wire.read();                                //First received byte is the low part of the angular data
  highByte = Wire.read();                               //Second received byte is the high part of the angular data
  gyro_yaw = ((highByte<<8)|lowByte);                   //Multiply highByte by 256 (shift left by 8) and ad lowByte
  gyro_yaw *= -1;                                       //Invert axis
  if(cal_int == 2000)gyro_yaw -= gyro_yaw_cal;   //Only compensate after the calibration
}
```

```
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//Subroutine for calculating pid outputs
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//The PID controllers are explained in part 5 of the YMFC-3D video:
//www.youtube.com/watch?v=JBvnB0279-Q

void calculate_pid(){

  //Roll calculations
  pid_error_temp = gyro_roll_input - pid_roll_setpoint;
  pid_i_mem_roll += pid_i_gain_roll * pid_error_temp;
  if(pid_i_mem_roll > pid_max_roll)pid_i_mem_roll = pid_max_roll;
  else if(pid_i_mem_roll < pid_max_roll * -1)pid_i_mem_roll = pid_max_roll * -1;

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

  pid_last_roll_d_error = pid_error_temp;

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

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

  pid_last_pitch_d_error = pid_error_temp;

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

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

  pid_last_yaw_d_error = pid_error_temp;
}
```

# APÉNDICE B

## CÓDIGO ARDUINO MICRO

```
#include <SoftwareSerial.h>
#include <Servo.h>
#include <TinyGPS.h>

/* Realizado y compilado en Arduina 1.6.0 */

TinyGPS gps;
SoftwareSerial mySerial(10, 11); // RX 11, TX 10 bluetooth
SoftwareSerial portdos(8, 9);  // RX 9, TX 8

Servo M1;
Servo M2;
Servo M3;
Servo M4;

float coordenada, coordenada2;

char MAC1, MAC2, MAC3, MAC4, MAC5, MAC6, MAC7, MAC8, MAC9, MAC11, MAC12, MAC13, MAC14,
MAC15, MAC16, MAC17, MAC18, MAC19, MAC20;

float GPS_latitud, GPS_longitud;
float Resultado_lat_1, Resultado_long_1;
float Resultado_lat_2, Resultado_long_2;


void  setup()
{
  Serial.begin(115200);
  mySerial.begin(9600); //BT
  portdos.begin(9600);

  pinMode(2, OUTPUT);
  pinMode(3, OUTPUT);

  M1.attach(4);  //trottle
  M2.attach(5);  //roll
  M3.attach(6);  //pitch
  M4.attach(7);  //yaw

  digitalWrite(3, HIGH); //inicializacion

   MAC1=' ';
   MAC2=' ';
   MAC3=' ';
   MAC4=' ';
   MAC5=' ';
   MAC6=' ';
   MAC7=' ';
   MAC8=' ';
   MAC9=' ';
   MAC11=' ';
   MAC12=' ';
   MAC13=' ';
   MAC14=' ';
   MAC15=' ';
   MAC16=' ';
   MAC17=' ';
```

```
   MAC18=' ';
   MAC19=' ';
   MAC20=' ';

  pinMode(12, OUTPUT);
  pinMode(13, OUTPUT);  //Configure digital port 12 and 13 as output.
}


void loop()
{
  delay (5);
  bool newData = false; // variable visible solo dentro del lazo
  unsigned long chars;
  unsigned short sentences, failed;
////////////////////////////////////////////////////////////////////////////
 PRIMERO:

  mySerial.listen();

  if (mySerial.isListening())
   {
     delay(5);
     if (mySerial.available() > 0)
        {MAC1= mySerial.read();}

     if(MAC1=='Z')   //
       {digitalWrite(3, LOW);
        M1.writeMicroseconds(1150); //trottle
        delay (2000);
        M1.writeMicroseconds(1300);
        delay (1000);
        M1.writeMicroseconds(1500);
        MAC1=' ';
        goto PRINCIPIO;}

        goto PRIMERO;
   }

////////////////////////////////////////////////////////////////////////////
//************************************************
////////////////////////////////////////////////////////////////////////////
 PRINCIPIO:

 mySerial.listen();

  if (mySerial.isListening())
   {
     delay(3000);
     if (mySerial.available() > 0)
        {MAC1= mySerial.read();}

      if(MAC1=='X')   //
        {digitalWrite(3, HIGH);
        M1.writeMicroseconds(1450);
        delay (3000);
        M1.writeMicroseconds(1300);
        delay (1500);
        M1.writeMicroseconds(1100);
        MAC1=' ';
        goto PRIMERO;}
```

```
    Serial.println("Entrando datos GPS");

 PRINCIPAL:
    if(MAC1=='A')   //
      {
         Serial.print("MAC1= ");
         Serial.println(MAC1);
         delay(10);
         if (mySerial.available() > 0)
             {MAC2= mySerial.read();}
         delay(5);
         if (mySerial.available() > 0)
             {MAC3= mySerial.read();}
         delay(5);
         if (mySerial.available() > 0)
             {MAC4= mySerial.read();}
         delay(5);
         if (mySerial.available() > 0)
             {MAC5= mySerial.read();}
         delay(5);
         if (mySerial.available() > 0)
             {MAC6= mySerial.read();}
         delay(5);
         if (mySerial.available() > 0)
             {MAC7= mySerial.read();}
         delay(5);
         if (mySerial.available() > 0)
             {MAC8= mySerial.read();
              Serial.print("MAC8= ");
              Serial.println(MAC8);}
         delay(5);
         if (mySerial.available() > 0)
             {MAC9= mySerial.read();
              if (MAC9 == 'N')
                 {MAC11 = MAC9;
                  MAC9 = '0';
                  Serial.print("MAC9= ");
                  Serial.println(MAC9);
                  goto Otra;}
              Serial.print("MAC9= ");
              Serial.println(MAC9);}
//*****************************************************
         delay(5);
         if (mySerial.available() > 0)
             {MAC11= mySerial.read();
              Serial.print("MAC11= ");
              Serial.println(MAC11);}
         delay(5);
         if (mySerial.available() > 0)
             {MAC12= mySerial.read();}
         delay(5);
         if (mySerial.available() > 0)
             {MAC13= mySerial.read();}
         delay(5);
         if (mySerial.available() > 0)
             {MAC14= mySerial.read();
             if (MAC14 == 'F')
                 {Serial.print("FALSO");
                  goto PRINCIPIO;}
              Serial.print("MAC14= ");
              Serial.println(MAC14);}
         delay(5);
```

```
          if (mySerial.available() > 0)
              {MAC15= mySerial.read();}
          delay(5);
          if (mySerial.available() > 0)
              {MAC16= mySerial.read();}
          delay(5);
          if (mySerial.available() > 0)
              {MAC17= mySerial.read();
               if (MAC17 == 'F')
                 {goto PRINCIPIO;}}
          delay(5);
          if (mySerial.available() > 0)
              {MAC18= mySerial.read();
               if (MAC18 == 'F')
                 {goto PRINCIPIO;}
               Serial.print("MAC18= ");
               Serial.println(MAC18);}
          delay (5);
          if (mySerial.available() > 0)
              {MAC19= mySerial.read();
               if (MAC19 == 'F')
                 {MAC19 = '0';}
               Serial.print("MAC19= ");
               Serial.println(MAC19);
               goto finn;}
//***********************************************
 Otra:     delay (5);
          if (mySerial.available() > 0)
              {MAC13= mySerial.read();
               MAC12=MAC13;}
          delay(5);
          if (mySerial.available() > 0)
              {MAC14= mySerial.read();
               MAC13=MAC14;}
          delay(5);
          if (mySerial.available() > 0)
              {MAC15= mySerial.read();
               MAC14=MAC15;}
          delay(5);
          if (mySerial.available() > 0)
              {MAC16= mySerial.read();
               MAC15=MAC16;}
          delay(5);
          if (mySerial.available() > 0)
              {MAC17= mySerial.read();
               MAC16=MAC17;}
          delay(5);
          if (mySerial.available() > 0)
              {MAC18= mySerial.read();
               MAC17=MAC18;
               if (MAC17 == 'F')
                 {goto PRINCIPIO;}}
          delay(5);
          if (mySerial.available() > 0)
              {MAC19= mySerial.read();
               MAC18=MAC19;
               if (MAC17 == 'F')
                 {goto PRINCIPIO;}
               Serial.print("MAC18= ");
               Serial.println(MAC18);}
          delay(5);
          if (mySerial.available() > 0)
```

```
                {MAC20= mySerial.read();
                 MAC19=MAC20;
                if (MAC19 == 'F')
                   {MAC19 = '0';}
                 Serial.print("MAC19= ");
                 Serial.println(MAC19);}
 finn:
        delay(1);
    }
      if (MAC1 != 'A')
         {goto PRINCIPIO;}

          MAC1=' ';
          MAC11=' ';
          goto numera;
 }
////////////////////////////////////////////////////////////////////////////
//*************************************************
////////////////////////////////////////////////////////////////////////////
 CICLO:

  portdos.listen();
  delay(10);
  if (portdos.isListening())
  {
  // For one second we parse GPS data and report some key values
  for (unsigned long start = millis(); millis() - start < 1000;)
   {
    while (portdos.available())
     {
      char x = portdos.read();
    //Serial.write(x); //  para ver el flujo de datos de GPS
      if (gps.encode(x)) // Did a new valid sentence come in?
        newData = true;
     }
    }
  }
//////////////////////////////////////////////////////////////////////////////
//***********************************************************
//////////////////////////////////////////////////////////////////////////////
  if (newData)
  {
    float flat, flon;
    unsigned long age;
    gps.f_get_position(&flat, &flon, &age);

 LATITUDE:
    Serial.print("LAT= ");
    Serial.print(flat == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 : flat, 5); //obteniendo data GPS Dron
    GPS_latitud = flat;
    Serial.print("          GPS_latitud= ");
    Serial.println(GPS_latitud, 5);

 LONGITUDE:
    Serial.print("LON= ");
    Serial.print(flon == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 : flon, 5); //obteniendo data GPS Dron
    GPS_longitud = (-1)*(flon);
    Serial.print("         GPS_longitud= ");
    Serial.println(GPS_longitud,5);
  }
//////////////////////////////////////////////////////////////////////////////
//*********************************************************
```

```
////////////////////////////////////////////////////////////////////////////
  gps.stats(&chars, &sentences, &failed);
  if (chars == 0)
  {
    Serial.println("** No se reciven datos de GPS **");
    digitalWrite(13, HIGH);
    delay (500);
    digitalWrite(13, LOW);

    goto CICLO;
  }
////////////////////////////////////////////////////////////////////////////
//**********************************************************
////////////////////////////////////////////////////////////////////////////

 //MOV_LATITUDE:
    Resultado_lat_1 = GPS_latitud - coordenada;
    Serial.print("Resultado_lat_1= ");
    Serial.println(Resultado_lat_1,5);
 //MOV_LONGITUDE:
    Resultado_long_1 = GPS_longitud - coordenada2;
    Serial.print("Resultado_long_1= ");
    Serial.println(Resultado_long_1,5);

    goto PRINCIPIO_2;

 CONTINUA:
 //MOV_LATITUDE_3:
    Resultado_lat_2 = Resultado_lat_1 + coordenada;
    Serial.print("Resultado_lat_2= ");
    Serial.println(Resultado_lat_2,5);
    delay(5);
 //MOV_LONGITUDE_3:
    Resultado_long_2 = Resultado_long_1 + coordenada2;
    Serial.print("Resultado_long_2= ");
    Serial.println(Resultado_long_2,5);
    delay(5);
////////////////////////////////////////////////////////////////////////////
//**************************************************************************
////////////////////////////////////////////////////////////////////////////
 AUMENTAR:
    if (GPS_latitud > Resultado_lat_2)
       {Serial.println("AUMENTAR_latitud");
        Serial.println();
        goto LAT_MAS;}
 IZQUIERDA:
    if (GPS_longitud < Resultado_long_2)
       {Serial.println("IZQUIERDA_longitud");
        Serial.println();
        goto LON_MENOS;}
 DERECHA:
    if (GPS_longitud > Resultado_long_2)
       {Serial.println("DERECHA_longitud");
        Serial.println();
        goto LON_MAS;}
  RETROCEDER:
    if (GPS_latitud < Resultado_lat_2)
       {Serial.println("RETROCEDER_latitud");
        Serial.println();
        goto LAT_MENOS;}

    if (GPS_latitud == Resultado_lat_2)
```

```
      {//TROTTLEE:
         M1.writeMicroseconds(1500);
         delay (10);
       //PITCHH:
         M3.writeMicroseconds(1500);
         delay (10);
        Serial.println("Latitud_igual");}

    if (GPS_longitud == Resultado_long_2)
      {//TROTTLEE:
         M1.writeMicroseconds(1500);
         delay (10);
        //PITCHH:
         M3.writeMicroseconds(1500);
         delay (10);
       Serial.println("Longitud_igual");
       Serial.println();
       goto PRINCIPIO;}

       goto PRINCIPIO;

////////////////////////////////////////////////////////////////////////////
//**************************************************
////////////////////////////////////////////////////////////////////////////
 LAT_MAS:
  //PITCHH:
         M3.writeMicroseconds(1600);  // mov hacia enfrente
  //TROTTLEE:
         M1.writeMicroseconds(1600);
         delay (500);
   //TROTTLEE:
         M1.writeMicroseconds(1500);
         delay (500);
   //PITCHH:
         M3.writeMicroseconds(1500);
         goto IZQUIERDA;
  LON_MENOS:
   //ROLL4:
         M2.writeMicroseconds(1400);   // mov hacia izquierda
   //TROTTLE4:
         M1.writeMicroseconds(1600);
         delay(500);
   //TROTTLE4:
         M1.writeMicroseconds(1500);
   //ROLL4:
         M2.writeMicroseconds(1500);
         goto DERECHA;
  LON_MAS:
   //ROLL3:
         M2.writeMicroseconds(1600);  // mov hacia derecha
   //TROTTLE3:
         M1.writeMicroseconds(1600);
         delay(500);
   //TROTTLE3:
         M1.writeMicroseconds(1500);
   //ROLL3:
         M2.writeMicroseconds(1500);
         goto RETROCEDER;
  LAT_MENOS:
   //PITCH2:
         M3.writeMicroseconds(1400);  // mov hacia atraz
   //TROTTLE2:
```

```
        M1.writeMicroseconds(1600);
        delay(500);
   //TROTTLE2:
        M1.writeMicroseconds(1500);
   //PITCH2:
        M3.writeMicroseconds(1500);
        goto PRINCIPIO;
////////////////////////////////////////////////////////////////////////
//*************************************************
////////////////////////////////////////////////////////////////////////
   numera:
        {
          if(MAC8=='0')
           {
              if(MAC9=='0')
                {coordenada = 00.00000;
                 goto CONTINUA_0;}
              if(MAC9=='1')
                {coordenada = 00.00001;
                 goto CONTINUA_0;}
              if(MAC9=='2')
                {coordenada = 00.00002;
                 goto CONTINUA_0;}
              if(MAC9=='3')
                {coordenada = 00.00003;
                 goto CONTINUA_0;}
              if(MAC9=='4')
                {coordenada = 00.00004;
                 goto CONTINUA_0;}
              if(MAC9=='5')
                {coordenada = 00.00005;
                 goto CONTINUA_0;}
              if(MAC9=='6')
                {coordenada = 00.00006;
                 goto CONTINUA_0;}
              if(MAC9=='7')
                {coordenada = 00.00007;
                 goto CONTINUA_0;}
              if(MAC9=='8')
                {coordenada = 00.00008;
                 goto CONTINUA_0;}
              if(MAC9=='9')
                {coordenada = 00.00009;
                 goto CONTINUA_0;}
           }
////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////
           if(MAC8=='1')
             {
              if(MAC9=='0')
                {coordenada = 00.00010;
                 goto CONTINUA_0;}
              if(MAC9=='1')
                {coordenada = 00.00011;
                 goto CONTINUA_0;}
              if(MAC9=='2')
                {coordenada = 00.00012;
                 goto CONTINUA_0;}
              if(MAC9=='3')
                {coordenada = 00.00013;
                 goto CONTINUA_0;}
              if(MAC9=='4')
```

```
                {coordenada = 00.00014;
                 goto CONTINUA_0;}
               if(MAC9=='5')
                {coordenada = 00.00015;
                 goto CONTINUA_0;}
               if(MAC9=='6')
                {coordenada = 00.00016;
                 goto CONTINUA_0;}
               if(MAC9=='7')
                {coordenada = 00.00017;
                 goto CONTINUA_0;}
               if(MAC9=='8')
                {coordenada = 00.00018;
                 goto CONTINUA_0;}
               if(MAC9=='9')
                {coordenada = 00.00019;
                 goto CONTINUA_0;}
              }
////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////
          if(MAC8=='2')
           {
              if(MAC9=='0')
                {coordenada = 00.00020;
                 goto CONTINUA_0;}
               if(MAC9=='1')
                {coordenada = 00.00021;
                 goto CONTINUA_0;}
               if(MAC9=='2')
                {coordenada = 00.00022;
                 goto CONTINUA_0;}
               if(MAC9=='3')
                {coordenada = 00.00023;
                 goto CONTINUA_0;}
               if(MAC9=='4')
                {coordenada = 00.00024;
                 goto CONTINUA_0;}
               if(MAC9=='5')
                {coordenada = 00.00025;
                 goto CONTINUA_0;}
               if(MAC9=='6')
                {coordenada = 00.00026;
                 goto CONTINUA_0;}
               if(MAC9=='7')
                {coordenada = 00.00027;
                 goto CONTINUA_0;}
               if(MAC9=='8')
                {coordenada = 00.00028;
                 goto CONTINUA_0;}
               if(MAC9=='9')
                {coordenada = 00.00029;
                 goto CONTINUA_0;}
              }
 ////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////
          if(MAC8=='3')
            {
              if(MAC9=='0')
                {coordenada = 00.00030;
                 goto CONTINUA_0;}
               if(MAC9=='1')
                {coordenada = 00.00031;
```

```
                goto CONTINUA_0;}
              if(MAC9=='2')
                {coordenada = 00.00032;
                 goto CONTINUA_0;}
              if(MAC9=='3')
                {coordenada = 00.00033;
                 goto CONTINUA_0;}
              if(MAC9=='4')
                {coordenada = 00.00034;
                 goto CONTINUA_0;}
              if(MAC9=='5')
                {coordenada = 00.00035;
                 goto CONTINUA_0;}
              if(MAC9=='6')
                {coordenada = 00.00036;
                 goto CONTINUA_0;}
              if(MAC9=='7')
                {coordenada = 00.00037;
                 goto CONTINUA_0;}
              if(MAC9=='8')
                {coordenada = 00.00038;
                 goto CONTINUA_0;}
              if(MAC9=='9')
                {coordenada = 00.00039;
                 goto CONTINUA_0;}
             }
////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////
        if(MAC8=='4')
          {
              if(MAC9=='0')
                {coordenada = 00.00040;
                 goto CONTINUA_0;}
              if(MAC9=='1')
                {coordenada = 00.00041;
                 goto CONTINUA_0;}
              if(MAC9=='2')
                {coordenada = 00.00042;
                 goto CONTINUA_0;}
              if(MAC9=='3')
                {coordenada = 00.00043;
                 goto CONTINUA_0;}
              if(MAC9=='4')
                {coordenada = 00.00044;
                 goto CONTINUA_0;}
              if(MAC9=='5')
                {coordenada = 00.00045;
                 goto CONTINUA_0;}
              if(MAC9=='6')
                {coordenada = 00.00046;
                 goto CONTINUA_0;}
              if(MAC9=='7')
                {coordenada = 00.00047;
                 goto CONTINUA_0;}
              if(MAC9=='8')
                {coordenada = 00.00048;
                 goto CONTINUA_0;}
              if(MAC9=='9')
                {coordenada = 00.00049;
                 goto CONTINUA_0;}
             }
 ////////////////////////////////////////////////////////////////////////////////
```

```
////////////////////////////////////////////////////////////////////////////////
          if(MAC8=='5')
            {
              if(MAC9=='0')
                {coordenada = 00.00050;
                 goto CONTINUA_0;}
              if(MAC9=='1')
                {coordenada = 00.00051;
                 goto CONTINUA_0;}
              if(MAC9=='2')
                {coordenada = 00.00052;
                 goto CONTINUA_0;}
              if(MAC9=='3')
                {coordenada = 00.00053;
                 goto CONTINUA_0;}
              if(MAC9=='4')
                {coordenada = 00.00054;
                 goto CONTINUA_0;}
              if(MAC9=='5')
                {coordenada = 00.00055;
                 goto CONTINUA_0;}
              if(MAC9=='6')
                {coordenada = 00.00056;
                 goto CONTINUA_0;}
              if(MAC9=='7')
                {coordenada = 00.00057;
                 goto CONTINUA_0;}
              if(MAC9=='8')
                {coordenada = 00.00058;
                 goto CONTINUA_0;}
              if(MAC9=='9')
                {coordenada = 00.00059;
                 goto CONTINUA_0;}
             }
 ////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////
          if(MAC8=='6')
            {
              if(MAC9=='0')
                {coordenada = 00.00060;
                 goto CONTINUA_0;}
              if(MAC9=='1')
                {coordenada = 00.00061;
                 goto CONTINUA_0;}
              if(MAC9=='2')
                {coordenada = 00.00062;
                 goto CONTINUA_0;}
              if(MAC9=='3')
                {coordenada = 00.00063;
                 goto CONTINUA_0;}
              if(MAC9=='4')
                {coordenada = 00.00064;
                 goto CONTINUA_0;}
              if(MAC9=='5')
                {coordenada = 00.00065;
                 goto CONTINUA_0;}
              if(MAC9=='6')
                {coordenada = 00.00066;
                 goto CONTINUA_0;}
              if(MAC9=='7')
                {coordenada = 00.00067;
                 goto CONTINUA_0;}
```

```
                if(MAC9=='8')
                  {coordenada = 00.00068;
                   goto CONTINUA_0;}
                if(MAC9=='9')
                  {coordenada = 00.00069;
                   goto CONTINUA_0;}
                }
 ////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////
              if(MAC8=='7')
                {
                if(MAC9=='0')
                  {coordenada = 00.00070;
                   goto CONTINUA_0;}
                if(MAC9=='1')
                  {coordenada = 00.00071;
                   goto CONTINUA_0;}
                if(MAC9=='2')
                  {coordenada = 00.00072;
                   goto CONTINUA_0;}
                if(MAC9=='3')
                  {coordenada = 00.00073;
                   goto CONTINUA_0;}
                if(MAC9=='4')
                  {coordenada = 00.00074;
                   goto CONTINUA_0;}
                if(MAC9=='5')
                  {coordenada = 00.00075;
                   goto CONTINUA_0;}
                if(MAC9=='6')
                  {coordenada = 00.00076;
                   goto CONTINUA_0;}
                if(MAC9=='7')
                  {coordenada = 00.00077;
                   goto CONTINUA_0;}
                if(MAC9=='8')
                  {coordenada = 00.00078;
                   goto CONTINUA_0;}
                if(MAC9=='9')
                  {coordenada = 00.00079;
                   goto CONTINUA_0;}
                }
////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////
              if(MAC8=='8')
                {
                if(MAC9=='0')
                  {coordenada = 00.00080;
                   goto CONTINUA_0;}
                if(MAC9=='1')
                  {coordenada = 00.00081;
                   goto CONTINUA_0;}
                if(MAC9=='2')
                  {coordenada = 00.00082;
                   goto CONTINUA_0;}
                if(MAC9=='3')
                  {coordenada = 00.00083;
                   goto CONTINUA_0;}
                if(MAC9=='4')
                  {coordenada = 00.00084;
                   goto CONTINUA_0;}
                if(MAC9=='5')
```

```
                {coordenada = 00.00085;
                 goto CONTINUA_0;}
              if(MAC9=='6')
                {coordenada = 00.00086;
                 goto CONTINUA_0;}
              if(MAC9=='7')
                {coordenada = 00.00087;
                 goto CONTINUA_0;}
              if(MAC9=='8')
                {coordenada = 00.00088;
                 goto CONTINUA_0;}
              if(MAC9=='9')
                {coordenada = 00.00089;
                 goto CONTINUA_0;}
             }
////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////
        if(MAC8=='9')
          {
             if(MAC9=='0')
               {coordenada = 90e-5; //00.00090;
                goto CONTINUA_0;}
             if(MAC9=='1')
               {coordenada = 91e-5; //00.00091;
                goto CONTINUA_0;}
             if(MAC9=='2')
               {coordenada = 92e-5; //00.00092;
                goto CONTINUA_0;}
             if(MAC9=='3')
               {coordenada = 93e-5; //00.00093;
                goto CONTINUA_0;}
             if(MAC9=='4')
               {coordenada = 94e-5; //00.00094;
                goto CONTINUA_0;}
             if(MAC9=='5')
               {coordenada = 95e-5; //00.00095;
                goto CONTINUA_0;}
             if(MAC9=='6')
               {coordenada = 96e-5; //00.00096;
                goto CONTINUA_0;}
             if(MAC9=='7')
               {coordenada = 97e-5; //00.00097;
                goto CONTINUA_0;}
             if(MAC9=='8')
               {coordenada = 98e-5; //00.00098;
                goto CONTINUA_0;}
             if(MAC9=='9')
               {coordenada = 99e-5; //00.00099;
                goto CONTINUA_0;}
            }
//============================================================
 CONTINUA_0:
          if(MAC18=='0')
           {
             if(MAC19=='0')
               {coordenada2 = 00.00000;
                goto CICLO;}
             if(MAC19=='1')
               {coordenada2 = 00.00001;
                goto CICLO;}
             if(MAC19=='2')
               {coordenada2 = 00.00002;
```

```
                goto CICLO;}
              if(MAC19=='3')
                {coordenada2 = 00.00003;
                 goto CICLO;}
              if(MAC19=='4')
                {coordenada2 = 00.00004;
                 goto CICLO;}
              if(MAC19=='5')
                {coordenada2 = 00.00005;
                 goto CICLO;}
              if(MAC19=='6')
                {coordenada2 = 00.00006;
                 goto CICLO;}
              if(MAC19=='7')
                {coordenada2 = 00.00007;
                 goto CICLO;}
              if(MAC19=='8')
                {coordenada2 = 00.00008;
                 goto CICLO;}
              if(MAC19=='9')
                {coordenada2 = 00.00009;
                 goto CICLO;}

          }
////////////////////////////////////////////
////////////////////////////////////////////
          if(MAC18=='1')
            {
              if(MAC19=='0')
                {coordenada2 = 00.00010;
                 goto CICLO;}
              if(MAC19=='1')
                {coordenada2 = 00.00011;
                 goto CICLO;}
              if(MAC19=='2')
                {coordenada2 = 00.00012;
                 goto CICLO;}
              if(MAC19=='3')
                {coordenada2 = 00.00013;
                 goto CICLO;}
              if(MAC19=='4')
                {coordenada2 = 00.00014;
                 goto CICLO;}
              if(MAC19=='5')
                {coordenada2 = 00.00015;
                 goto CICLO;}
              if(MAC19=='6')
                {coordenada2 = 00.00016;
                 goto CICLO;}
              if(MAC19=='7')
                {coordenada2 = 00.00017;
                 goto CICLO;}
              if(MAC19=='8')
                {coordenada2 = 00.00018;
                 goto CICLO;}
              if(MAC19=='9')
                {coordenada2 = 00.00019;
                 goto CICLO;}
            }
////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////
          if(MAC18=='2')
```

```
            {
               if(MAC19=='0')
                 {coordenada2 = 00.00020;
                  goto CICLO;}
               if(MAC19=='1')
                 {coordenada2 = 00.00021;
                  goto CICLO;}
               if(MAC19=='2')
                 {coordenada2 = 00.00022;
                  goto CICLO;}
               if(MAC19=='3')
                 {coordenada2 = 00.00023;
                  goto CICLO;}
               if(MAC19=='4')
                 {coordenada2 = 00.00024;
                  goto CICLO;}
               if(MAC19=='5')
                 {coordenada2 = 00.00025;
                  goto CICLO;}
               if(MAC19=='6')
                 {coordenada2 = 00.00026;
                  goto CICLO;}
               if(MAC19=='7')
                 {coordenada2 = 00.00027;
                  goto CICLO;}
               if(MAC19=='8')
                 {coordenada2 = 00.00028;
                  goto CICLO;}
               if(MAC19=='9')
                 {coordenada2 = 00.00029;
                  goto CICLO;}
              }
 ////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////
            if(MAC18=='3')
              {
               if(MAC19=='0')
                 {coordenada2 = 00.00030;
                  goto CICLO;}
               if(MAC19=='1')
                 {coordenada2 = 00.00031;
                  goto CICLO;}
               if(MAC19=='2')
                 {coordenada2 = 00.00032;
                  goto CICLO;}
               if(MAC19=='3')
                 {coordenada2 = 00.00033;
                  goto CICLO;}
               if(MAC19=='4')
                 {coordenada2 = 00.00034;
                  goto CICLO;}
               if(MAC19=='5')
                 {coordenada2 = 00.00035;
                  goto CICLO;}
               if(MAC19=='6')
                 {coordenada2 = 00.00036;
                  goto CICLO;}
               if(MAC19=='7')
                 {coordenada2 = 00.00037;
                  goto CICLO;}
               if(MAC19=='8')
                 {coordenada2 = 00.00038;
```

```
                goto CICLO;}
              if(MAC19=='9')
                {coordenada2 = 00.00039;
                 goto CICLO;}
             }
////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////
          if(MAC18=='4')
           {
              if(MAC19=='0')
                {coordenada2 = 00.00040;
                 goto CICLO;}
              if(MAC19=='1')
                {coordenada2 = 00.00041;
                 goto CICLO;}
              if(MAC19=='2')
                {coordenada2 = 00.00042;
                 goto CICLO;}
              if(MAC19=='3')
                {coordenada2 = 00.00043;
                 goto CICLO;}
              if(MAC19=='4')
                {coordenada2 = 00.00044;
                 goto CICLO;}
              if(MAC19=='5')
                {coordenada2 = 00.00045;
                 goto CICLO;}
              if(MAC19=='6')
                {coordenada2 = 00.00046;
                 goto CICLO;}
              if(MAC19=='7')
                {coordenada2 = 00.00047;
                 goto CICLO;}
              if(MAC19=='8')
                {coordenada2 = 00.00048;
                 goto CICLO;}
              if(MAC19=='9')
                {coordenada2 = 00.00049;
                 goto CICLO;}
             }
////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////

          if(MAC18=='5')
            {
              if(MAC19=='0')
                {coordenada2 = 00.00050;
                 goto CICLO;}
              if(MAC19=='1')
                {coordenada2 = 00.00051;
                 goto CICLO;}
              if(MAC19=='2')
                {coordenada2 = 00.00052;
                 goto CICLO;}
              if(MAC19=='3')
                {coordenada2 = 00.00053;
                 goto CICLO;}
              if(MAC19=='4')
                {coordenada2 = 00.00054;
                 goto CICLO;}
              if(MAC19=='5')
                {coordenada2 = 00.00055;
```

```
                goto CICLO;}
              if(MAC19=='6')
                {coordenada2 = 00.00056;
                 goto CICLO;}
              if(MAC19=='7')
                {coordenada2 = 00.00057;
                 goto CICLO;}
              if(MAC19=='8')
                {coordenada2 = 00.00058;
                 goto CICLO;}
              if(MAC19=='9')
                {coordenada2 = 00.00059;
                 goto CICLO;}
             }
////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////
          if(MAC18=='6')
            {
              if(MAC19=='0')
                {coordenada2 = 00.00060;
                 goto CICLO;}
              if(MAC19=='1')
                {coordenada2 = 00.00061;
                 goto CICLO;}
              if(MAC19=='2')
                {coordenada2 = 00.00062;
                 goto CICLO;}
              if(MAC19=='3')
                {coordenada2 = 00.00063;
                 goto CICLO;}
              if(MAC19=='4')
                {coordenada2 = 00.00064;
                 goto CICLO;}
              if(MAC19=='5')
                {coordenada2 = 00.00065;
                 goto CICLO;}
              if(MAC19=='6')
                {coordenada2 = 00.00066;
                 goto CICLO;}
              if(MAC19=='7')
                {coordenada2 = 00.00067;
                 goto CICLO;}
              if(MAC19=='8')
                {coordenada2 = 00.00068;
                 goto CICLO;}
              if(MAC19=='9')
                {coordenada2 = 00.00069;
                 goto CICLO;}
             }
////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////
           if(MAC18=='7')
             {
              if(MAC19=='0')
                {coordenada2 = 00.00070;
                 goto CICLO;}
              if(MAC19=='1')
                {coordenada2 = 00.00071;
                 goto CICLO;}
              if(MAC19=='2')
                {coordenada2 = 00.00072;
                 goto CICLO;}
```

```
                if(MAC19=='3')
                  {coordenada2 = 00.00073;
                   goto CICLO;}
                if(MAC19=='4')
                  {coordenada2 = 00.00074;
                   goto CICLO;}
                if(MAC19=='5')
                  {coordenada2 = 00.00075;
                   goto CICLO;}
                if(MAC19=='6')
                  {coordenada2 = 00.00076;
                   goto CICLO;}
                if(MAC19=='7')
                  {coordenada2 = 00.00077;
                   goto CICLO;}
                if(MAC19=='8')
                  {coordenada2 = 00.00078;
                   goto CICLO;}
                if(MAC19=='9')
                  {coordenada2 = 00.00079;
                   goto CICLO;}
                }
////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////
          if(MAC18=='8')
            {
              if(MAC19=='0')
                {coordenada2 = 00.00080;
                 goto CICLO;}
              if(MAC19=='1')
                {coordenada2 = 00.00081;
                 goto CICLO;}
              if(MAC19=='2')
                {coordenada2 = 00.00082;
                 goto CICLO;}
              if(MAC19=='3')
                {coordenada2 = 00.00083;
                 goto CICLO;}
              if(MAC19=='4')
                {coordenada2 = 00.00084;
                 goto CICLO;}
              if(MAC19=='5')
                {coordenada2 = 00.00085;
                 goto CICLO;}
              if(MAC19=='6')
                {coordenada2 = 00.00086;
                 goto CICLO;}
              if(MAC19=='7')
                {coordenada2 = 00.00087;
                 goto CICLO;}
              if(MAC19=='8')
                {coordenada2 = 00.00088;
                 goto CICLO;}
              if(MAC19=='9')
                {coordenada2 = 00.00089;
                 goto CICLO;}
              }
////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////
          if(MAC18=='9')
            {
              if(MAC19=='0')
```

```
                {coordenada2 = 00.00090;
                 goto CICLO;}
               if(MAC19=='1')
                {coordenada2 = 00.00091;
                 goto CICLO;}
               if(MAC19=='2')
                {coordenada2 = 00.00092;
                 goto CICLO;}
               if(MAC19=='3')
                {coordenada2 = 00.00093;
                 goto CICLO;}
               if(MAC19=='4')
                {coordenada2 = 00.00094;
                 goto CICLO;}
               if(MAC19=='5')
                {coordenada2 = 00.00095;
                 goto CICLO;}
               if(MAC19=='6')
                {coordenada2 = 00.00096;
                 goto CICLO;}
               if(MAC19=='7')
                {coordenada2 = 00.00097;
                 goto CICLO;}
               if(MAC19=='8')
                {coordenada2 = 00.00098;
                 goto CICLO;}
               if(MAC19=='9')
                {coordenada2 = 00.00099;
                 goto CICLO;}
              }
        }
//============================================
//============================================
//============================================
 PRINCIPIO_2:
  mySerial.listen();
  delay(5);
  if (mySerial.isListening())
   {
     delay(3000);
     if (mySerial.available() > 0)
       {MAC1= mySerial.read();}

     Serial.println("Entrando datos GPS_2");

     if(MAC1=='A')   //
       {
          Serial.print("MAC1= ");
          Serial.println(MAC1);
          delay(10);
          if (mySerial.available() > 0)
              {MAC2= mySerial.read();}
          delay(5);
          if (mySerial.available() > 0)
              {MAC3= mySerial.read();}
          delay(5);
          if (mySerial.available() > 0)
              {MAC4= mySerial.read();}
          delay(5);
          if (mySerial.available() > 0)
              {MAC5= mySerial.read();}
          delay(5);
```

```
          if (mySerial.available() > 0)
              {MAC6= mySerial.read();}
          delay(5);
          if (mySerial.available() > 0)
              {MAC7= mySerial.read();}
          delay(5);
          if (mySerial.available() > 0)
              {MAC8= mySerial.read();
               Serial.print("MAC8= ");
               Serial.println(MAC8);}
          delay(5);
          if (mySerial.available() > 0)
              {MAC9= mySerial.read();
               if (MAC9 == 'N')
                  {MAC11 = MAC9;
                   MAC9 = '0';
                   Serial.print("MAC9= ");
                   Serial.println(MAC9);
                   goto Otra_2;}
               Serial.print("MAC9= ");
               Serial.println(MAC9);}
//**************************************************
          delay(5);
          if (mySerial.available() > 0)
              {MAC11= mySerial.read();
               Serial.print("MAC11= ");
               Serial.println(MAC11);}
          delay(5);
          if (mySerial.available() > 0)
              {MAC12= mySerial.read();}
          delay(5);
          if (mySerial.available() > 0)
              {MAC13= mySerial.read();}
          delay(5);
          if (mySerial.available() > 0)
              {MAC14= mySerial.read();}
          delay(5);
          if (mySerial.available() > 0)
              {MAC15= mySerial.read();}
          delay(5);
          if (mySerial.available() > 0)
              {MAC16= mySerial.read();}
          delay(5);
          if (mySerial.available() > 0)
              {MAC17= mySerial.read();}
          delay(5);
          if (mySerial.available() > 0)
              {MAC18= mySerial.read();
               if (MAC18 == 'F')
                  {goto finn_2;}
               Serial.print("MAC18= ");
               Serial.println(MAC18);}
          delay (5);
          if (mySerial.available() > 0)
              {MAC19= mySerial.read();
               if (MAC19 == 'F')
                  {MAC19 = '0';}
               Serial.print("MAC19= ");
               Serial.println(MAC19);
               goto finn_2;}
//**************************************************
 Otra_2:    delay (5);
```

```
          if (mySerial.available() > 0)
              {MAC13= mySerial.read();
               MAC12=MAC13;}
          delay(5);
          if (mySerial.available() > 0)
              {MAC14= mySerial.read();
               MAC13=MAC14;}
          delay(5);
          if (mySerial.available() > 0)
              {MAC15= mySerial.read();
               MAC14=MAC15;}
           delay(5);
           if (mySerial.available() > 0)
              {MAC16= mySerial.read();
               MAC15=MAC16;}
            delay(5);
           if (mySerial.available() > 0)
              {MAC17= mySerial.read();
               MAC16=MAC17;}
           delay(5);
           if (mySerial.available() > 0)
              {MAC18= mySerial.read();
               MAC17=MAC18;}
           delay(5);
           if (mySerial.available() > 0)
              {MAC19= mySerial.read();
               MAC18=MAC19;
               if (MAC17 == 'F')
                  {goto finn_2;}
               Serial.print("MAC18= ");
               Serial.println(MAC18);}
           delay(5);
           if (mySerial.available() > 0)
              {MAC20= mySerial.read();
               MAC19=MAC20;
              if (MAC19 == 'F')
                 {MAC19 = '0';}
               Serial.print("MAC19= ");
               Serial.println(MAC19);}
 finn_2:
        delay(1);
       }
      if (MAC1 != 'A')
      {goto PRINCIPIO_2;}
       MAC1=' ';
       MAC11=' ';
   }
////////////////////////////////////////////////////////////////////////////
//**********************************************
////////////////////////////////////////////////////////////////////////////
 numera_Dos:
           if(MAC8=='0')
            {
              if(MAC9=='0')
                {coordenada = 00.00000;
                 goto ULTIMA;}
              if(MAC9=='1')
                {coordenada = 00.00001;
                 goto ULTIMA;}
              if(MAC9=='2')
                {coordenada = 00.00002;
                 goto ULTIMA;}
```

```
                if(MAC9=='3')
                  {coordenada = 00.00003;
                   goto ULTIMA;}
                if(MAC9=='4')
                  {coordenada = 00.00004;
                   goto ULTIMA;}
                if(MAC9=='5')
                  {coordenada = 00.00005;
                   goto ULTIMA;}
                if(MAC9=='6')
                  {coordenada = 00.00006;
                   goto ULTIMA;}
                if(MAC9=='7')
                  {coordenada = 00.00007;
                   goto ULTIMA;}
                if(MAC9=='8')
                  {coordenada = 00.00008;
                   goto ULTIMA;}
                if(MAC9=='9')
                  {coordenada = 00.00009;
                   goto ULTIMA;}
              }
////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////
             if(MAC8=='1')
               {
                if(MAC9=='0')
                  {coordenada = 00.00010;
                   goto ULTIMA;}
                if(MAC9=='1')
                  {coordenada = 00.00011;
                   goto ULTIMA;}
                if(MAC9=='2')
                  {coordenada = 00.00012;
                   goto ULTIMA;}
                if(MAC9=='3')
                  {coordenada = 00.00013;
                   goto ULTIMA;}
                if(MAC9=='4')
                  {coordenada = 00.00014;
                   goto ULTIMA;}
                if(MAC9=='5')
                  {coordenada = 00.00015;
                   goto ULTIMA;}
                if(MAC9=='6')
                  {coordenada = 00.00016;
                   goto ULTIMA;}
                if(MAC9=='7')
                  {coordenada = 00.00017;
                   goto ULTIMA;}
                if(MAC9=='8')
                  {coordenada = 00.00018;
                   goto ULTIMA;}
                if(MAC9=='9')
                  {coordenada = 00.00019;
                   goto ULTIMA;}
                }
////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////
             if(MAC8=='2')
               {
                if(MAC9=='0')
```

```
                {coordenada = 00.00020;
                 goto ULTIMA;}
               if(MAC9=='1')
                {coordenada = 00.00021;
                 goto ULTIMA;}
               if(MAC9=='2')
                {coordenada = 00.00022;
                 goto ULTIMA;}
               if(MAC9=='3')
                {coordenada = 00.00023;
                 goto ULTIMA;}
               if(MAC9=='4')
                {coordenada = 00.00024;
                 goto ULTIMA;}
               if(MAC9=='5')
                {coordenada = 00.00025;
                 goto ULTIMA;}
               if(MAC9=='6')
                {coordenada = 00.00026;
                 goto ULTIMA;}
               if(MAC9=='7')
                {coordenada = 00.00027;
                 goto ULTIMA;}
               if(MAC9=='8')
                {coordenada = 00.00028;
                 goto ULTIMA;}
               if(MAC9=='9')
                {coordenada = 00.00029;
                 goto ULTIMA;}
              }
////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////
          if(MAC8=='3')
           {
            if(MAC9=='0')
              {coordenada = 00.00030;
               goto ULTIMA;}
            if(MAC9=='1')
              {coordenada = 00.00031;
               goto ULTIMA;}
            if(MAC9=='2')
              {coordenada = 00.00032;
               goto ULTIMA;}
            if(MAC9=='3')
              {coordenada = 00.00033;
               goto ULTIMA;}
            if(MAC9=='4')
              {coordenada = 00.00034;
               goto ULTIMA;}
            if(MAC9=='5')
              {coordenada = 00.00035;
               goto ULTIMA;}
            if(MAC9=='6')
              {coordenada = 00.00036;
               goto ULTIMA;}
            if(MAC9=='7')
              {coordenada = 00.00037;
               goto ULTIMA;}
            if(MAC9=='8')
              {coordenada = 00.00038;
               goto ULTIMA;}
            if(MAC9=='9')
```

```
                {coordenada = 00.00039;
                 goto ULTIMA;}
              }
////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////
          if(MAC8=='4')
           {
              if(MAC9=='0')
                {coordenada = 00.00040;
                 goto ULTIMA;}
              if(MAC9=='1')
                {coordenada = 00.00041;
                 goto ULTIMA;}
              if(MAC9=='2')
                {coordenada = 00.00042;
                 goto ULTIMA;}
              if(MAC9=='3')
                {coordenada = 00.00043;
                 goto ULTIMA;}
              if(MAC9=='4')
                {coordenada = 00.00044;
                 goto ULTIMA;}
              if(MAC9=='5')
                {coordenada = 00.00045;
                 goto ULTIMA;}
              if(MAC9=='6')
                {coordenada = 00.00046;
                 goto ULTIMA;}
              if(MAC9=='7')
                {coordenada = 00.00047;
                 goto ULTIMA;}
              if(MAC9=='8')
                {coordenada = 00.00048;
                 goto ULTIMA;}
              if(MAC9=='9')
                {coordenada = 00.00049;
                 goto ULTIMA;}
              }
////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////
           if(MAC8=='5')
             {
              if(MAC9=='0')
                {coordenada = 00.00050;
                 goto ULTIMA;}
              if(MAC9=='1')
                {coordenada = 00.00051;
                 goto ULTIMA;}
              if(MAC9=='2')
                {coordenada = 00.00052;
                 goto ULTIMA;}
              if(MAC9=='3')
                {coordenada = 00.00053;
                 goto ULTIMA;}
              if(MAC9=='4')
                {coordenada = 00.00054;
                 goto ULTIMA;}
              if(MAC9=='5')
                {coordenada = 00.00055;
                 goto ULTIMA;}
              if(MAC9=='6')
                {coordenada = 00.00056;
```

```
                goto ULTIMA;}
              if(MAC9=='7')
                {coordenada = 00.00057;
                 goto ULTIMA;}
              if(MAC9=='8')
                {coordenada = 00.00058;
                 goto ULTIMA;}
              if(MAC9=='9')
                {coordenada = 00.00059;
                 goto ULTIMA;}
             }
////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////
          if(MAC8=='6')
            {
              if(MAC9=='0')
                {coordenada = 00.00060;
                 goto ULTIMA;}
              if(MAC9=='1')
                {coordenada = 00.00061;
                 goto ULTIMA;}
              if(MAC9=='2')
                {coordenada = 00.00062;
                 goto ULTIMA;}
              if(MAC9=='3')
                {coordenada = 00.00063;
                 goto ULTIMA;}
              if(MAC9=='4')
                {coordenada = 00.00064;
                 goto ULTIMA;}
              if(MAC9=='5')
                {coordenada = 00.00065;
                 goto ULTIMA;}
              if(MAC9=='6')
                {coordenada = 00.00066;
                 goto ULTIMA;}
              if(MAC9=='7')
                {coordenada = 00.00067;
                 goto ULTIMA;}
              if(MAC9=='8')
                {coordenada = 00.00068;
                 goto ULTIMA;}
              if(MAC9=='9')
                {coordenada = 00.00069;
                 goto ULTIMA;}
             }
////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////
          if(MAC8=='7')
            {
              if(MAC9=='0')
                {coordenada = 00.00070;
                 goto ULTIMA;}
              if(MAC9=='1')
                {coordenada = 00.00071;
                 goto ULTIMA;}
              if(MAC9=='2')
                {coordenada = 00.00072;
                 goto ULTIMA;}
              if(MAC9=='3')
                {coordenada = 00.00073;
                 goto ULTIMA;}
```

```
                if(MAC9=='4')
                  {coordenada = 00.00074;
                   goto ULTIMA;}
                if(MAC9=='5')
                  {coordenada = 00.00075;
                   goto ULTIMA;}
                if(MAC9=='6')
                  {coordenada = 00.00076;
                   goto ULTIMA;}
                if(MAC9=='7')
                  {coordenada = 00.00077;
                   goto ULTIMA;}
                if(MAC9=='8')
                  {coordenada = 00.00078;
                   goto ULTIMA;}
                if(MAC9=='9')
                  {coordenada = 00.00079;
                   goto ULTIMA;}
               }
////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////
          if(MAC8=='8')
            {
               if(MAC9=='0')
                 {coordenada = 00.00080;
                  goto ULTIMA;}
               if(MAC9=='1')
                 {coordenada = 00.00081;
                  goto ULTIMA;}
               if(MAC9=='2')
                 {coordenada = 00.00082;
                  goto ULTIMA;}
               if(MAC9=='3')
                 {coordenada = 00.00083;
                  goto ULTIMA;}
               if(MAC9=='4')
                 {coordenada = 00.00084;
                  goto ULTIMA;}
               if(MAC9=='5')
                 {coordenada = 00.00085;
                  goto ULTIMA;}
               if(MAC9=='6')
                 {coordenada = 00.00086;
                  goto ULTIMA;}
               if(MAC9=='7')
                 {coordenada = 00.00087;
                  goto ULTIMA;}
               if(MAC9=='8')
                 {coordenada = 00.00088;
                  goto ULTIMA;}
               if(MAC9=='9')
                 {coordenada = 00.00089;
                  goto ULTIMA;}
              }
////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////
          if(MAC8=='9')
            {
               if(MAC9=='0')
                 {coordenada = 00.00090;
                  goto ULTIMA;}
               if(MAC9=='1')
```

```
                {coordenada = 00.00091;
                 goto ULTIMA;}
               if(MAC9=='2')
                {coordenada = 00.00092;
                 goto ULTIMA;}
               if(MAC9=='3')
                {coordenada = 00.00093;
                 goto ULTIMA;}
               if(MAC9=='4')
                {coordenada = 00.00094;
                 goto ULTIMA;}
               if(MAC9=='5')
                {coordenada = 00.00095;
                 goto ULTIMA;}
               if(MAC9=='6')
                {coordenada = 00.00096;
                 goto ULTIMA;}
               if(MAC9=='7')
                {coordenada = 00.00097;
                 goto ULTIMA;}
               if(MAC9=='8')
                {coordenada = 00.00098;
                 goto ULTIMA;}
               if(MAC9=='9')
                {coordenada = 00.00099;
                 goto ULTIMA;}
              }
//=============================================================
//=============================================================
//=============================================================
 ULTIMA:
          if(MAC18=='0')
           {
              if(MAC19=='0')
                {coordenada2 = 00.00000;
                 goto CONTINUA;}
               if(MAC19=='1')
                {coordenada2 = 00.00001;
                 goto CONTINUA;}
               if(MAC19=='2')
                {coordenada2 = 00.00002;
                 goto CONTINUA;}
               if(MAC19=='3')
                {coordenada2 = 00.00003;
                 goto CONTINUA;}
               if(MAC19=='4')
                {coordenada2 = 00.00004;
                 goto CONTINUA;}
               if(MAC19=='5')
                {coordenada2 = 00.00005;
                 goto CONTINUA;}
               if(MAC19=='6')
                {coordenada2 = 00.00006;
                 goto CONTINUA;}
               if(MAC19=='7')
                {coordenada2 = 00.00007;
                 goto CONTINUA;}
               if(MAC19=='8')
                {coordenada2 = 00.00008;
                 goto CONTINUA;}
               if(MAC19=='9')
                {coordenada2 = 00.00009;
```

```
                goto CONTINUA;}
            }
////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////
           if(MAC18=='1')
             {
              if(MAC19=='0')
                {coordenada2 = 00.00010;
                 goto CONTINUA;}
              if(MAC19=='1')
                {coordenada2 = 00.00011;
                 goto CONTINUA;}
              if(MAC19=='2')
                {coordenada2 = 00.00012;
                 goto CONTINUA;}
              if(MAC19=='3')
                {coordenada2 = 00.00013;
                 goto CONTINUA;}
              if(MAC19=='4')
                {coordenada2 = 00.00014;
                 goto CONTINUA;}
              if(MAC19=='5')
                {coordenada2 = 00.00015;
                 goto CONTINUA;}
              if(MAC19=='6')
                {coordenada2 = 00.00016;
                 goto CONTINUA;}
              if(MAC19=='7')
                {coordenada2 = 00.00017;
                 goto CONTINUA;}
              if(MAC19=='8')
                {coordenada2 = 00.00018;
                 goto CONTINUA;}
              if(MAC19=='9')
                {coordenada2 = 00.00019;
                 goto CONTINUA;}
             }
//////////////////////////////////////////////////
//////////////////////////////////////////////////
           if(MAC18=='2')
            {
              if(MAC19=='0')
                {coordenada2 = 00.00020;
                 goto CONTINUA;}
              if(MAC19=='1')
                {coordenada2 = 00.00021;
                 goto CONTINUA;}
              if(MAC19=='2')
                {coordenada2 = 00.00022;
                 goto CONTINUA;}
              if(MAC19=='3')
                {coordenada2 = 00.00023;
                 goto CONTINUA;}
              if(MAC19=='4')
                {coordenada2 = 00.00024;
                 goto CONTINUA;}
              if(MAC19=='5')
                {coordenada2 = 00.00025;
                 goto CONTINUA;}
              if(MAC19=='6')
                {coordenada2 = 00.00026;
                 goto CONTINUA;}
```

```
                if(MAC19=='7')
                  {coordenada2 = 00.00027;
                   goto CONTINUA;}
                if(MAC19=='8')
                  {coordenada2 = 00.00028;
                   goto CONTINUA;}
                if(MAC19=='9')
                  {coordenada2 = 00.00029;
                   goto CONTINUA;}
               }
 ////////////////////////////////////////////
 ////////////////////////////////////////////
            if(MAC18=='3')
             {
                if(MAC19=='0')
                  {coordenada2 = 00.00030;
                   goto CONTINUA;}
                if(MAC19=='1')
                  {coordenada2 = 00.00031;
                   goto CONTINUA;}
                if(MAC19=='2')
                  {coordenada2 = 00.00032;
                   goto CONTINUA;}
                if(MAC19=='3')
                  {coordenada2 = 00.00033;
                   goto CONTINUA;}
                if(MAC19=='4')
                  {coordenada2 = 00.00034;
                   goto CONTINUA;}
                if(MAC19=='5')
                  {coordenada2 = 00.00035;
                   goto CONTINUA;}
                if(MAC19=='6')
                  {coordenada2 = 00.00036;
                   goto CONTINUA;}
                if(MAC19=='7')
                  {coordenada2 = 00.00037;
                   goto CONTINUA;}
                if(MAC19=='8')
                  {coordenada2 = 00.00038;
                   goto CONTINUA;}
                if(MAC19=='9')
                  {coordenada2 = 00.00039;
                   goto CONTINUA;}
               }
 ////////////////////////////////////////////////////
////////////////////////////////////////////////////////////
           if(MAC18=='4')
            {
               if(MAC19=='0')
                 {coordenada2 = 00.00040;
                  goto CONTINUA;}
               if(MAC19=='1')
                 {coordenada2 = 00.00041;
                  goto CONTINUA;}
               if(MAC19=='2')
                 {coordenada2 = 00.00042;
                  goto CONTINUA;}
               if(MAC19=='3')
                 {coordenada2 = 00.00043;
                  goto CONTINUA;}
               if(MAC19=='4')
```

```
                {coordenada2 = 00.00044;
                 goto CONTINUA;}
               if(MAC19=='5')
                {coordenada2 = 00.00045;
                 goto CONTINUA;}
               if(MAC19=='6')
                {coordenada2 = 00.00046;
                 goto CONTINUA;}
               if(MAC19=='7')
                {coordenada2 = 00.00047;
                 goto CONTINUA;}
               if(MAC19=='8')
                {coordenada2 = 00.00048;
                 goto CONTINUA;}
               if(MAC19=='9')
                {coordenada2 = 00.00049;
                 goto CONTINUA;}
              }
 ////////////////////////////////////////////////////
//////////////////////////////////////////////////////////////
          if(MAC18=='5')
            {
              if(MAC19=='0')
                {coordenada2 = 00.00050;
                 goto CONTINUA;}
               if(MAC19=='1')
                {coordenada2 = 00.00051;
                 goto CONTINUA;}
               if(MAC19=='2')
                {coordenada2 = 00.00052;
                 goto CONTINUA;}
               if(MAC19=='3')
                {coordenada2 = 00.00053;
                 goto CONTINUA;}
               if(MAC19=='4')
                {coordenada2 = 00.00054;
                 goto CONTINUA;}
               if(MAC19=='5')
                {coordenada2 = 00.00055;
                 goto CONTINUA;}
               if(MAC19=='6')
                {coordenada2 = 00.00056;
                 goto CONTINUA;}
               if(MAC19=='7')
                {coordenada2 = 00.00057;
                 goto CONTINUA;}
               if(MAC19=='8')
                {coordenada2 = 00.00058;
                 goto CONTINUA;}
               if(MAC19=='9')
                {coordenada2 = 00.00059;
                 goto CONTINUA;}
              }
 ////////////////////////////////////////////////////
//////////////////////////////////////////////////////////////////
          if(MAC18=='6')
            {
              if(MAC19=='0')
                {coordenada2 = 00.00060;
                 goto CONTINUA;}
               if(MAC19=='1')
                {coordenada2 = 00.00061;
```

```
                goto CONTINUA;}
              if(MAC19=='2')
                {coordenada2 = 00.00062;
                 goto CONTINUA;}
              if(MAC19=='3')
                {coordenada2 = 00.00063;
                 goto CONTINUA;}
              if(MAC19=='4')
                {coordenada2 = 00.00064;
                 goto CONTINUA;}
              if(MAC19=='5')
                {coordenada2 = 00.00065;
                 goto CONTINUA;}
              if(MAC19=='6')
                {coordenada2 = 00.00066;
                 goto CONTINUA;}
              if(MAC19=='7')
                {coordenada2 = 00.00067;
                 goto CONTINUA;}
              if(MAC19=='8')
                {coordenada2 = 00.00068;
                 goto CONTINUA;}
              if(MAC19=='9')
                {coordenada2 = 00.00069;
                 goto CONTINUA;}
             }
 ////////////////////////////////////////////////////
//////////////////////////////////////////////////////////
          if(MAC18=='7')
            {
              if(MAC19=='0')
                {coordenada2 = 00.00070;
                 goto CONTINUA;}
              if(MAC19=='1')
                {coordenada2 = 00.00071;
                 goto CONTINUA;}
              if(MAC19=='2')
                {coordenada2 = 00.00072;
                 goto CONTINUA;}
              if(MAC19=='3')
                {coordenada2 = 00.00073;
                 goto CONTINUA;}
              if(MAC19=='4')
                {coordenada2 = 00.00074;
                 goto CONTINUA;}
              if(MAC19=='5')
                {coordenada2 = 00.00075;
                 goto CONTINUA;}
              if(MAC19=='6')
                {coordenada2 = 00.00076;
                 goto CONTINUA;}
              if(MAC19=='7')
                {coordenada2 = 00.00077;
                 goto CONTINUA;}
              if(MAC19=='8')
                {coordenada2 = 00.00078;
                 goto CONTINUA;}
              if(MAC19=='9')
                {coordenada2 = 00.00079;
                 goto CONTINUA;}
             }
////////////////////////////////////////////////////
```

```
//////////////////////////////////////////////////////////
          if(MAC18=='8')
            {
              if(MAC19=='0')
                {coordenada2 = 00.00080;
                 goto CONTINUA;}
              if(MAC19=='1')
                {coordenada2 = 00.00081;
                 goto CONTINUA;}
              if(MAC19=='2')
                {coordenada2 = 00.00082;
                 goto CONTINUA;}
              if(MAC19=='3')
                {coordenada2 = 00.00083;
                 goto CONTINUA;}
              if(MAC19=='4')
                {coordenada2 = 00.00084;
                 goto CONTINUA;}
              if(MAC19=='5')
                {coordenada2 = 00.00085;
                 goto CONTINUA;}
              if(MAC19=='6')
                {coordenada2 = 00.00086;
                 goto CONTINUA;}
              if(MAC19=='7')
                {coordenada2 = 00.00087;
                 goto CONTINUA;}
              if(MAC19=='8')
                {coordenada2 = 00.00088;
                 goto CONTINUA;}
              if(MAC19=='9')
                {coordenada2 = 00.00089;
                 goto CONTINUA;}
             }
//////////////////////////////////////////////////////////
//////////////////////////////////////////////////////////
          if(MAC18=='9')
           {
              if(MAC19=='0')
                {coordenada2 = 90e-5; //00.00090;
                 goto CONTINUA;}
              if(MAC19=='1')
                {coordenada2 = 91e-5; //00.00091;
                 goto CONTINUA;}
              if(MAC19=='2')
                {coordenada2 = 92e-5; //00.00092;
                 goto CONTINUA;}
              if(MAC19=='3')
                {coordenada2 = 93e-5; //00.00093;
                 goto CONTINUA;}
              if(MAC19=='4')
                {coordenada2 = 94e-5; //00.00094;
                 goto CONTINUA;}
              if(MAC19=='5')
                {coordenada2 = 95e-5; //00.00095;
                 goto CONTINUA;}
              if(MAC19=='6')
                {coordenada2 = 96e-5; //00.00096;
                 goto CONTINUA;}
              if(MAC19=='7')
                {coordenada2 = 97e-5; //00.00097;
                 goto CONTINUA;}
```

```
            if(MAC19=='8')
              {coordenada2 = 98e-5; //00.00098;
               goto CONTINUA;}
            if(MAC19=='9')
              {coordenada2 = 99e-5; //00.00099;
               goto CONTINUA;}
          }
          Serial.print("NO encontro dato");
         // goto CONTINUA;
}
```

## APÉNDICE C

### Controlador PID

PID son las siglas para (Proporcional Integral Derivativo). Un controlador PID es un mecanismo de control que calcula la desviación o error entre un valor medido y el valor que se quiere obtener para aplicar una acción correctora [36].

El algoritmo de control calcula tres parámetros diferentes: el proporcional, el integral y el derivativo. El primero, es directamente proporcional al error actual, el Integral hace una corrección del error acumulado en el tiempo (integral del error) y el Derivativo establece la reacción del tiempo en el que el error se produce. Dependiendo como esté escrito el código del algoritmo del controlador PID, afectará al vuelo de forma diferente, de acuerdo a lo antes mencionado de los parámetros [36].

La ecuación que utilizo Joop Brokking para la realización del control PID es la que se muestra a continuación. En la figura 57 se muestra el diagrama del control PID implementado donde se puede observar que se ocupan los datos principales provenientes del giroscopio y del receptor. Este control nos da como resultado la salida PID-output la cual es la suma de las tres salidas la proporcional, la integral y la derivativa.

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{d}{dt} e(t) \qquad (7)$$

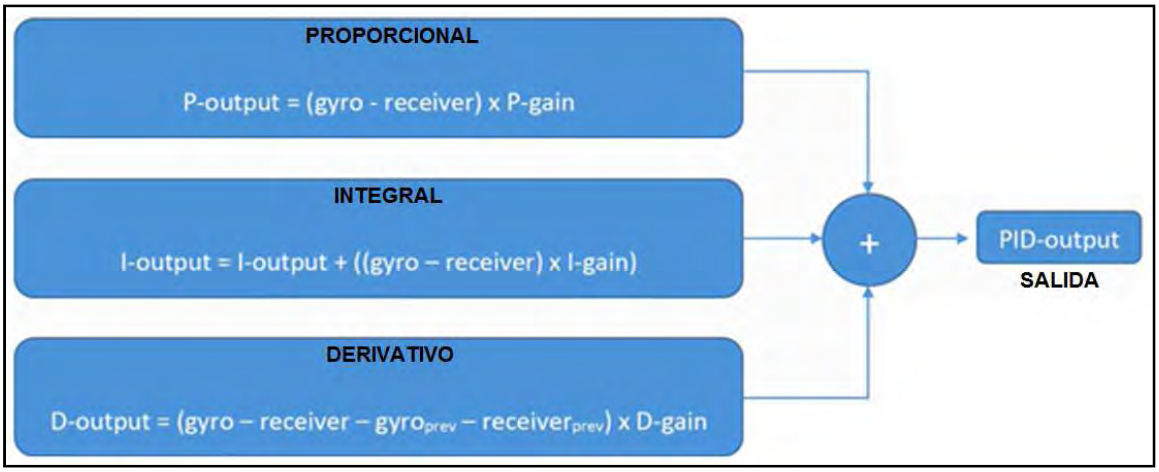



Figura 57.- Diagrama del PID de salida [37].

## ANEXO 1

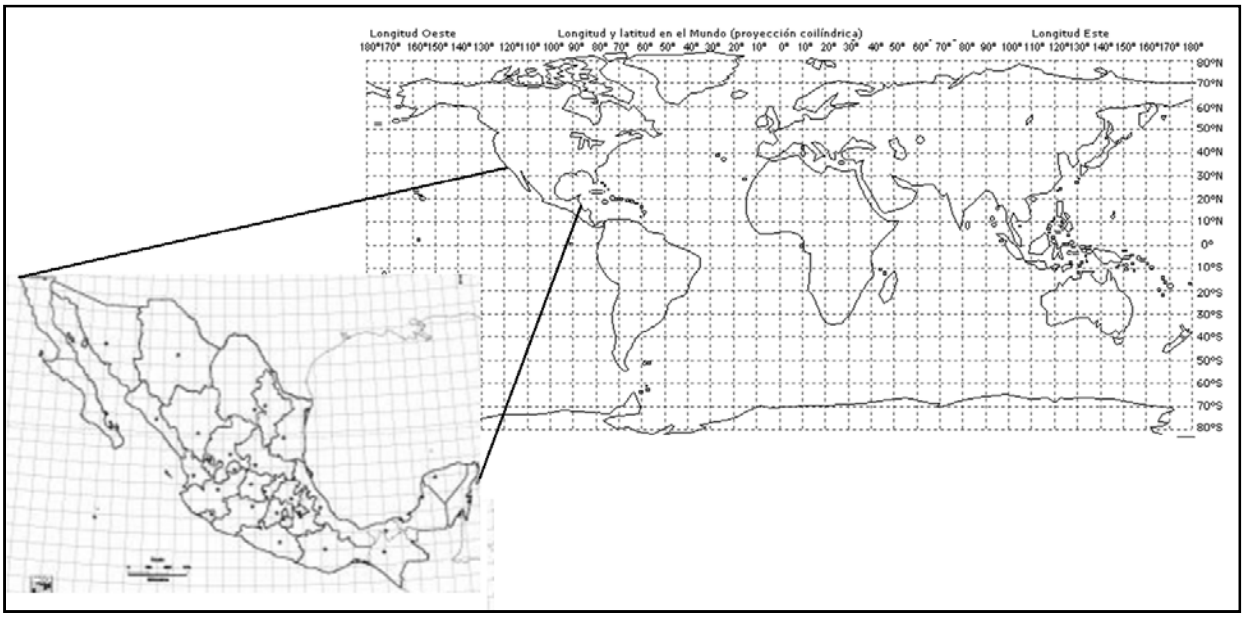



Figura 58.- Posicionamiento de coordenadas de México en el mundo.

## ANEXO 2

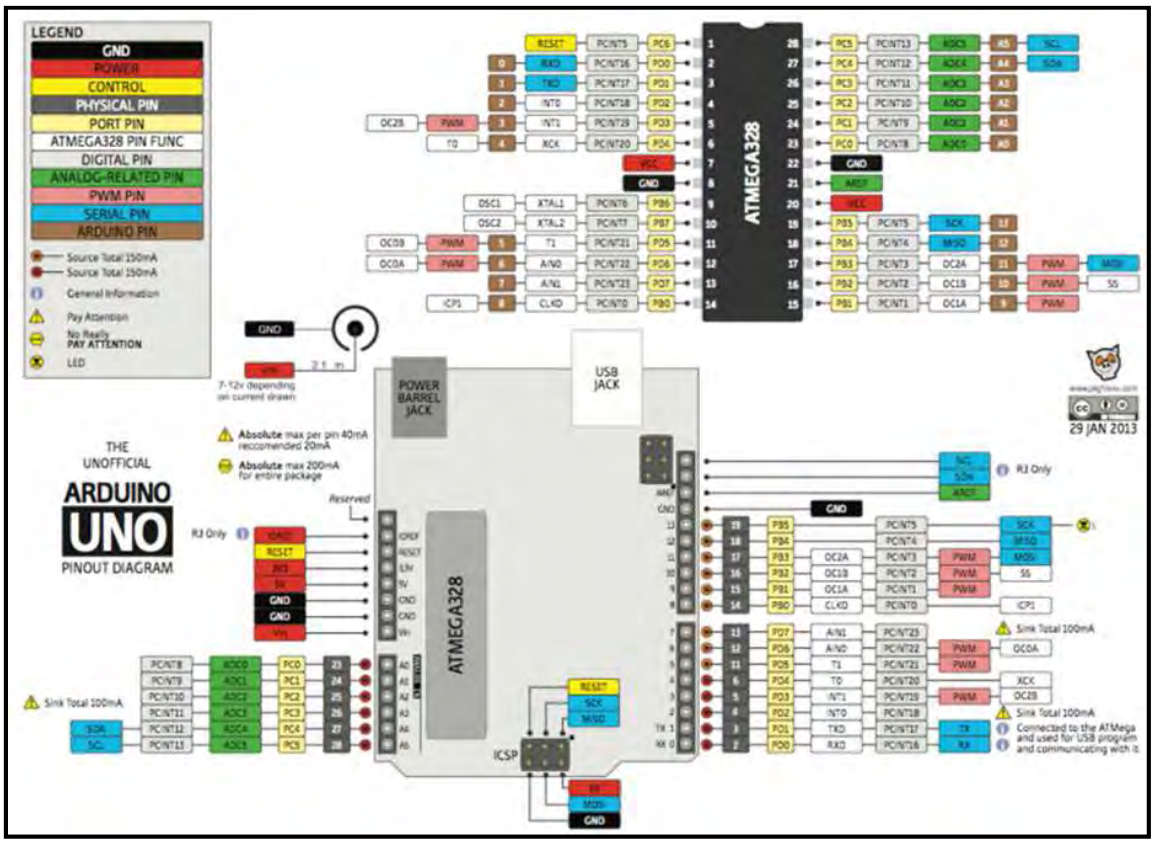



Figura 59.- Esquema de Arduino Uno

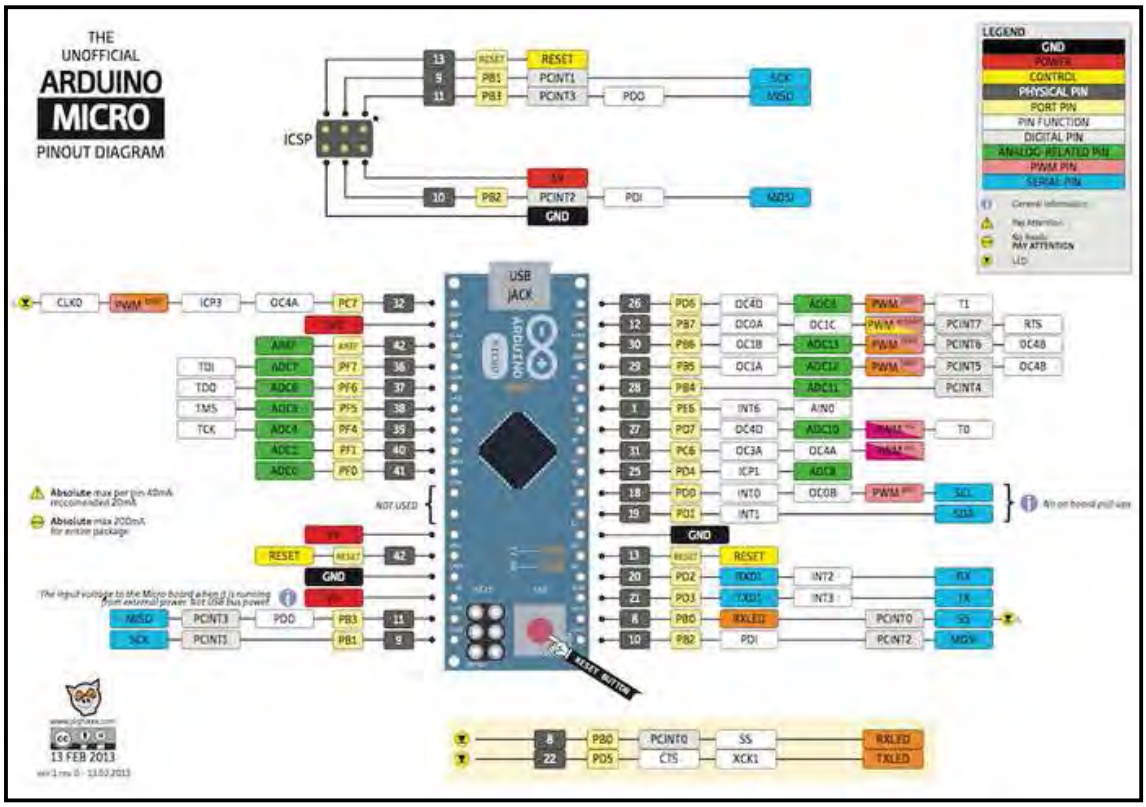



Figura 60.- Esquema de Arduino Micro.

## ANEXO 3

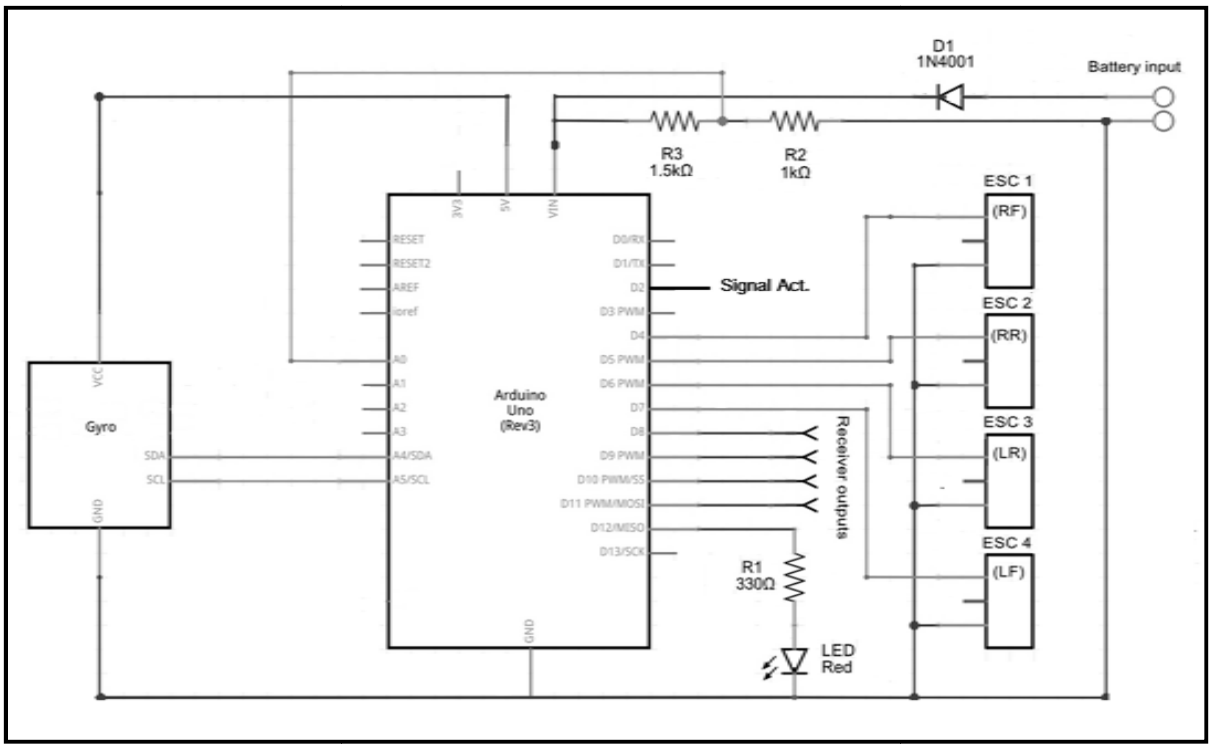



Figura 61.- Diagrama de conexiones de la placa Arduino Uno.

La comunicación de datos por $I^2C$ :

**SCL-** (System Clock) es la línea de los pulsos de reloj.
**SDA-** (System Data) es la línea de datos.
**GND-** (Masa) común.

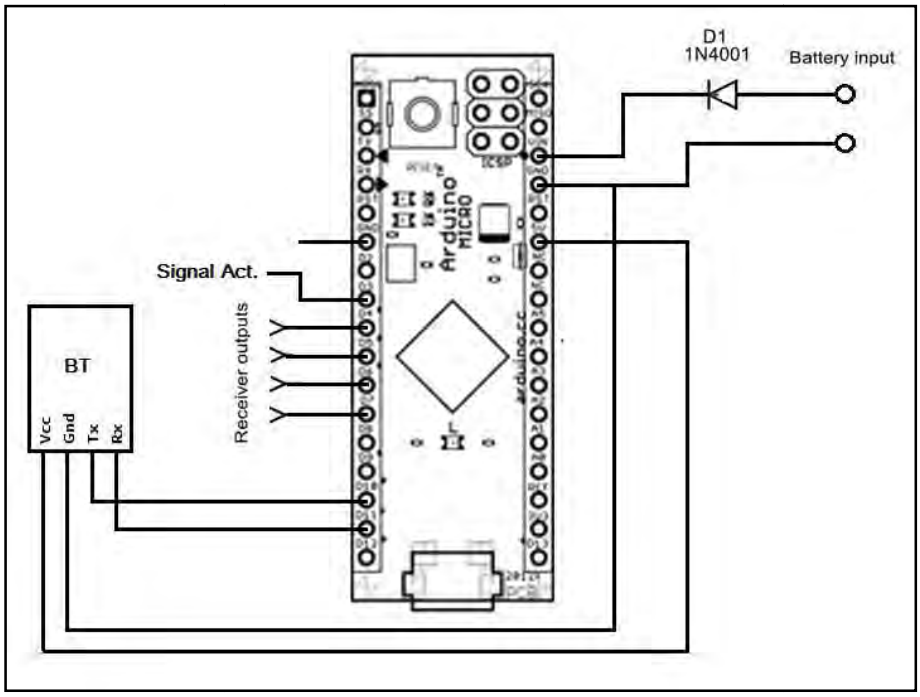



Figura 62.- Diagrama de conexiones de placa Arduino micro y BT.

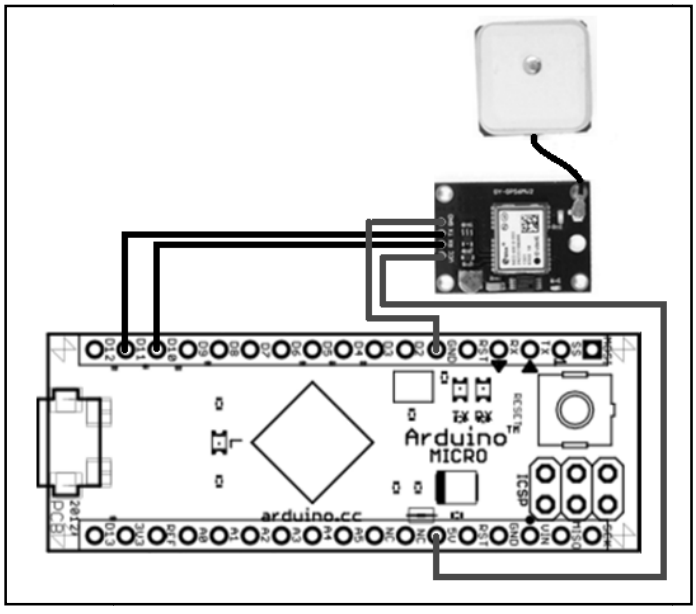
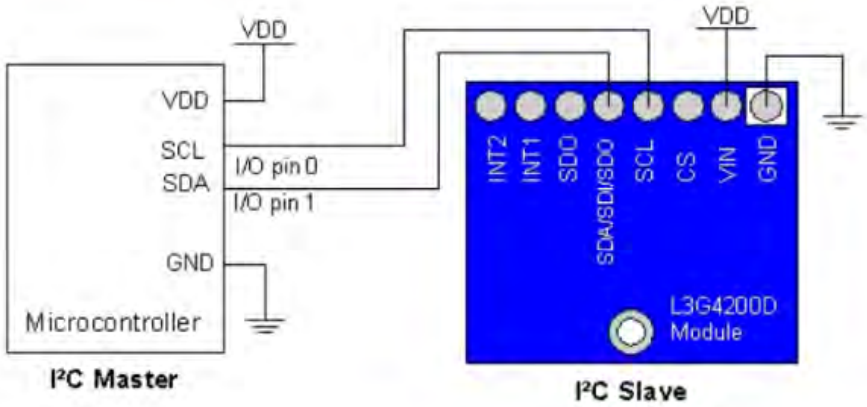



Figura 63.- Diagrama placa Arduino micro y GPS.

## ANEXO 4

# L3G4200D (#27911) GIROSCOPIO 3 EJES




### L3G4200D Características Mecánicas

| Símbolo | Parámetro | Condición de Prueba | Típico | Unidades |
|---|---|---|---|---|
| FS | Rango de medición | Seleccionable por usuario | +/- 250<br>+/- 500<br>+/- 2000 | dps |
| So | Sensibilidad | FS = 250 dps<br>FS = 500 dps<br>FS = 2000 dps | 8.75<br>17.50<br>70 | Mdps/digit |
| SoDr | Cambio de sensibilidad vs. Temperatura | De -40º C a + 85 ºC | +/- 2 % | |
| DVoff | Rango de Nivel-Cero Digital | FS = 250 dps<br>FS = 500 dps<br>FS = 2000 dps | +/- 10<br>+/- 15<br>+/- 75 | dps |
| OffDr | Rango de Nivel-Cero Digital vs. Cambio de Temperatura | FS = 250 dps<br>FS = 500 dps | +/- 0.03<br>+/- 0.04 | dps/ºC |
| NL | No linealidad | Mejor acomodo línea directa | 0.2 % FS | |
| DST | Auto prueba cambio de salida | FS = 250 dps<br>FS = 500 dps<br>FS = 2000 dps | 130<br>200<br>530 | dps |
| Rn | Densidad de rango de ruido | BW = 50 Hz | 0.03 | dps/sqrt(Hz) |
| ODR | Rango de datos salida digital | 100/200/400/800 | Hz | |

Extracto tomado de la hoja de datos del L3G4200D