Diseño e implementación del sistema de control de vuelo de un UAV

Fabio Nelson Castañeda García

Ramiro Andrés Henao

Fabio Alejandro Valencia

Universidad Católica de Manizales

Facultad de Ingeniería y arquitectura

Manizales, Colombia

2016

Diseño e implementación del sistema de control de vuelo de un UAV

Fabio Nelson Castañeda García
Ramiro Andrés Henao
Fabio Alejandro Valencia

Tesis o trabajo de investigación presentada(o) como requisito parcial para optar al título de:

Ingenieros Telemáticos

Director (a):

Ingeniero Oscar Cardona Morales

Universidad Católica de Manizales
Facultad de Ingeniería y arquitectura
Manizales, Colombia

2016

Lema

La preocupación por el hombre y su destino siempre debe ser el interés primordial de todo esfuerzo técnico. Nunca olvides esto entre tus diagramas y ecuaciones.

Albert Einstein

Agradecimientos

Primeramente gracias a Dios por ser nuestro guía, a Jesús por ser nuestra inspiración, modelo y por ser el ejemplo más grande de amor en este mundo y nuestros padres por darnos el ejemplo de vida a seguir.

CONTENIDO

RESUMEN 10

INTRODUCCIÓN 11

OBJETIVOS 12
1. Objetivo General 12
2. Objetivos Específicos 12

CAPÍTULO 1. VEHÍCULOS AÉREOS NO TRIPULADOS 13

CAPITULO 2. COMPOSICIÓN GENERAL DEL SISTEMA CUADRICÓPTERO 20

CAPITULO 3 ESTRUCTURA DEL CUADRICÓPTERO 34

CONCLUSIONES 85

LISTA DE FIGURAS 86

Bibliografía 87

RESUMEN

El principal objetivo de este proyecto es la implementación y puesta en marcha desde sus fases iniciales de una unidad de vuelo no tripulado (UAV) en el modelo de un cuadricoptero con él se pretende aportar a un proyecto de investigación denominado "Aplicabilidad de índices espectrales para la evaluación del estado de los humedales alto andinos empleando una cámara multiespectral soportada por un vehículo aéreo no tripulado. Caso de estudio complejo de humedales el ocho y paramo de Letras", que permita la captura de imágenes multiespectrales. Con el fin de dar inicio a este trabajo, primero se buscaron los materiales necesarios para construir el cuadricoptero, procedimiento que se realizó mediante páginas especializadas en cada componente, con el fin de estudiar sus funcionalidades y poner en marcha la construcción del mismo.

Posteriormente se inició el diseño del software, el cual fue construido mediante el estudio e investigaciones de otros proyectos, con la reutilización y modificación de códigos para el presente proyecto. Seguido de este paso, se dio inicio al ensamblaje de todos sus componentes y la realización de pruebas, con el fin de corregir errores en el código, hasta lograr su adecuado funcionamiento.

El proyecto concluye con el diseño y puesta en marcha del cuadricóptero.

INTRODUCCIÓN

El presente trabajo da cuenta del proceso realizado para Implementación y puesta en marcha de una unidad de vuelo no tripulado (UAV) para la captura de imágenes multiespectrales que se diseñó en el Marco del proyecto investigación que se lleva a cabo la universidad Católica de Manizales.

Los vehículos no tripulados (del inglés UAV- Unmanned Aerial Vehicle) han tenido un gran avance en los últimos años desde diferentes campos y para diversos usos como el comercial, el entretenimiento y últimamente en el área militar; esto debido a su capacidad de acceder a entornos difíciles y que pueden representar un peligro para el ser humano, es así que su uso también es común en la búsqueda y rescate, exploración de edificaciones entre otras.

Es por lo tanto pertinente poder avanzar en el búsqueda de otros usos posibles de los UAV y poder contribuir con este proyecto de grado al macro proyectó de la universidad Católica en el campo ambiental.

El trabajo está estructurado en tres sesiones; la primera hace referencia a un contexto conceptual de los UAV, como sus características, usos y propiedades.

En el segundo capítulo se hace referencia a la composición general del sistema UAV, haciendo una breve descripción de los sistemas y dinámicas de estos, el modelo propiamente empleado en este proyecto que es el cuadricoperto.

Y el tercer capítulo contiene las especificaciones técnicas del cuadricoptero ensamblado y puesto en marcha.

OBJETIVOS

1. Objetivo General

- Implementación y puesta en marcha de una unidad de vuelo no tripulado (UAV) para la captura de imágenes multiespectrales en el marco del proyecto ambiental que lleva a cabo la universidad Católica de Manizales.

2. Objetivos Específicos

- Adaptación de las piezas del modelo cuadricoptero para la realización de pruebas piloto.

- Diseño del algoritmo de control de vuelo necesarios para completar un vuelo estable en cualquier tipo de condición climatológica.

CAPÍTULO 1. VEHÍCULOS AÉREOS NO TRIPULADOS

SISTEMAS UAV:

- **Definición**

Un Vehículo Aéreo no Tripulado (UAV: Unmanned Aerial Vehicle) es un vehículo controlado autónomamente o desde tierra utilizando planes de vuelo programados. Las aplicaciones de este tipo de vehículos es cada día mayor en tareas que implican algún tipo de dificultad o riesgo para vehículos convencionales tripulados por personas, como son la detección de incendios, la identificación de manchas de petróleo en el mar, el seguimiento del tráfico, la inspección de líneas de tendido eléctrico, etc.(1)

Estos vehículos han sido utilizados en aplicaciones militares tales como reconocimiento de terreno y ataque; así mismo son muy útiles en la detección de incendios, control policial en situaciones de riesgo, reconocimiento de desastres naturales, situaciones que sin visión aéreas serían más difíciles de detectar.

Además de presentar algunas ventajas como son menor coste que las aeronaves tripuladas, no se arriesgan vidas, capacidad de incorporar muchos sensores y la posibilidad de acceder a sitios peligrosos o de difícil acceso.

- **Clasificación de los UAV**

En la actualidad podemos clasificar a los UAV según sus características de vuelo, en la siguiente tabla se hace referencia tanto aplicaciones civiles como militares.

Categoría	***Acrónimo***	***Alcance(km)***	***Altitud de vuelo (m)***	***Autonomía(h)***	***Carga máxima (Kg)***
Micro < 250 gr.	Micro	< 10	250	1	<5
Mini <25 kg.	Mini	< 10	150 y 300	<2	<30
Alcance cercano	CR	10 a 30	3.000	2 a 4	150
Alcance corto	SR	30 a 70	3.000	3 a 6	200
Alcance medio	MR	70 a 200	5.000	6 a 10	1.250
Altitud baja	LADP	>250	50 a 9.000	0,5 a 1	350
Autonomía media	MRE	>500	8.000	10 a 18	1.250
Autonomía alta / Altitud baja	LALE	>500	3.000	>24	>30
Autonomía alta / Altitud media	MALE	>500	14.000	24 a 48	1.500
Autonomía alta / Altitud alta	HALE	>2.000	20.000	24 a 48	12.000

Combate	UCAV	1.500	10.000	2	10.000

Ofensivo	**LETH**	**300**	**4.000**	**3 a 4**	**250**
Señuelo	DEC	500	5.000	<4	250
Estratosférico	STRATO	>2.000	20.000 y 30.000	>48	ND
EXO-estratosférico	EXO	ND	>30.000	ND	ND

Figura 1: ***Clasificación de los UAV (1)***

- **Aplicaciones de los UAV**

En la actualidad hay gran diversidad de aplicaciones que utilizan los UAV debido a su gran versatilidad y uso, a continuación se presentan algunas de los más comunes (2):

- **Misiones militares**, existe una gran diferencia a nivel económico entre el ámbito militar y el civil. En la anterior tabla hemos podido observar el gran número de tipos de UAV que discurren en el ámbito militar. En espionaje los UAV son bastante utilizados, desde los más pequeños (difícil de detectar por los radares) hasta los más grandes. Una de las grandes características a destacar a la hora de utilizar los UAV en misiones militares es que no arriesgamos vidas humanas y lo único que podemos perder es todo el sistema UAV.

- **Supervisión**, una de las capacidades más valoradas es la gran maniobrabilidad que presentan algunos sistemas UAV como puede ser los cuadricopteros. Esto puede ser utilizado para la inspección de obras civiles

como grandes puentes, edificios o estructuras de gran envergadura. Estos vehículos pueden ir dotados de cámaras de video y ayudar a la supervisión de la obra desde diferentes puntos sin poner en riesgo las vidas humanas y reduciendo el tiempo de ejecución ya que no hay que montar ninguna estructura externa para poder acceder al sitio en concreto.

- **Fotografía aérea**, la fotografía aérea o adquisición de datos aéreos es un área importante ya que podemos extraer bastante información del terreno. En este aspecto podemos utilizar tanto los vehículos más pequeños hasta los más grandes, ya que según la altitud en la que queramos hacer las fotografías elegiremos uno u otro.

- **Vigilancia**, los equipos de seguridad tanto privados como los policiales pueden utilizar estos sistemas para la vigilancia de una zona en concreto o de soporte en alguna tarea específica

- **Reconocimiento de desastres**, al ser vehículos no tripulados, puede servir para el reconocimiento inicial de desastres tales como explosiones nucleares o reconocimientos en lo que se pondría en peligro la integridad de los pilotos.

Los UAV son eficaces para aplicaciones donde sea necesario el censado de parámetros y adquisición de datos, fundamentalmente en lugares de difícil acceso y sin poner en peligro ninguna vida humana.

- **Tipos de UAV**

Existen también diferentes tipos de UAV, según su despegue, el siguiente mapa ilustra los diferentes tipos (2)

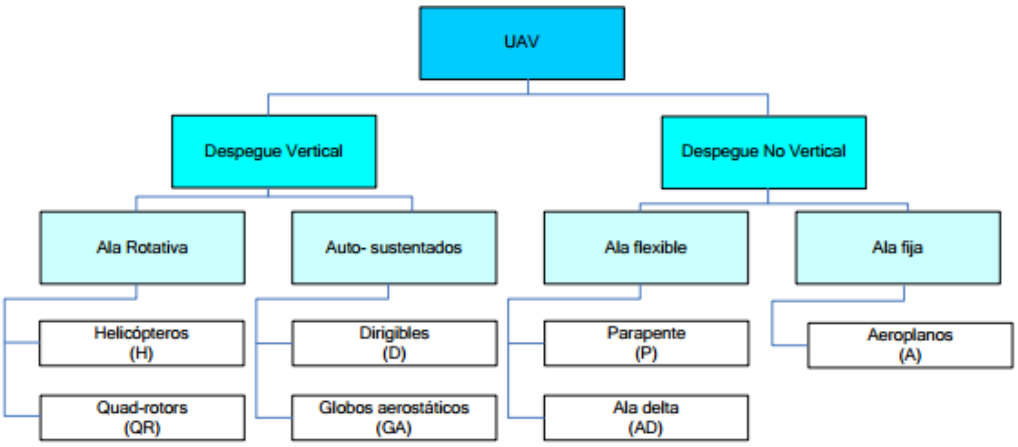


Figura 2: ***Tipos de UAV Según su despegue*** (1)

CUADRICOPTERO (Quad-rotors)

- **Definición**

Un Cuadricoptero se puede definir como una aeronave que se eleva y se desplaza por el movimiento de cuatro motores colocados en los extremos de una estructura en forma de cruz. Normalmente se utiliza el nombre inglés quadcotor aunque también existe la traducción cuadricóptero. Tal como se presenta en la figura, el vehículo dispone de 4 motores con sus palas respectivas, se utiliza la velocidad de los motores para controlar la estabilidad y movimientos del vehículo aéreo. (3)

Figura 3: ***Ejemplo de un cuadricoptero***

- **Principales características**

Los sistemas cuadricopteros se sitúan en la categoría de mini UAV (< 25 kg). Como se observa en la tabla de clasificaciones las características generales de vuelo de esta categoría son las siguientes:

- Alcance: < 10 km.
- Altitud de vuelo: < 300 m
- Autonomía: < 2 h
- Carga máxima: < 5 kg.

De las particularidades a subrayar es la gran versatilidad que pose este vehículo. Al tener cuatro motores es más fácil de controlar, lo que ayuda a utilizarlo en aplicaciones donde la exactitud de vuelo es muy importante. Una aplicación donde se aprecia esta característica es en la navegación de interiores y sitios de espacio reducido.

Como en el helicóptero, estos vehículos disponen de una capacidad de vuelo vertical que los hacen únicos, esta función es ventajosa cuando no queremos tener mucha velocidad horizontal y cuando queremos tener una buena capacidad de vuelo estacionario, lo que ayuda a elegir este tipo de sistemas cuando se quieren adquirir datos desde el vehículo.

El principal problema de los cuadricopteros es su control. El sistema debe de incorporar mecanismos de estabilización para ayudar a la navegación. La capacidad de carga es bastante alta con relación al peso de todo el sistema (5 kg / 25 kg), podemos encontrar vehículos que soporten una carga superior al peso que tienen. Esta característica hace posible el incorporar un gran número de sensores. Una de las características más

importantes a tener en cuenta en los sistemas de vuelo es la autonomía. La autonomía de vuelo no suele ser muy buena (< 2 h), esta fue una de las limitaciones por la que los UAV tardaron un cierto tiempo en avanzar. (2)

CAPITULO 2. COMPOSICIÓN GENERAL DEL SISTEMA CUADRICÓPTERO

En este capítulo se muestra la elaboración de un sistema cuadricoptero; se presentara en dos unidades, la primera es una introducción a la dinámica del cuadricoptero desde el punto de vista práctico y la segunda unidad hará referencia a las partes mecánicas, desde los motores hasta los sensores pasando por el sistema de procesado de información.

- **Dinámica del Cuadricóptero (3)**

La dinámica de un cuadricoptero se puede explicar a partir de la inercia de cada rotor. Cada rotor consta de un motor, un mecanismo de engranaje y un rotor de palas con ángulo fijo. Una de las características más importantes del cuadricoptero es que el motor delantero y el motor trasero rotan en sentido anti-horario (2,4), los otros dos motores rotan en sentido horario (1,3). Con ello, los efectos giroscópicos y los momentos aerodinámicos tienden a cancelarse en vuelo estacionario.

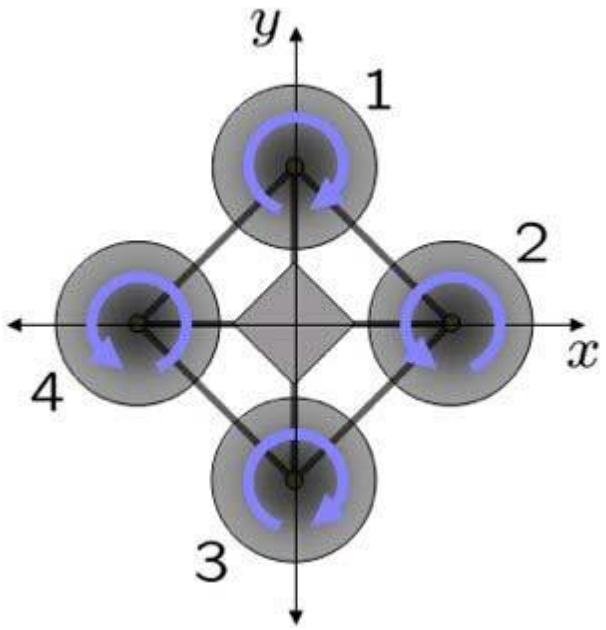


Figura 4*: Rotores Cuadricoptero* (2)

Para lograr movimiento hacia adelante la velocidad del rotor trasero (4) debe ser aumentada y, simultáneamente, la velocidad del rotor delantero (2) debe ser disminuida. El desplazamiento lateral se ejecuta con el mismo procedimiento, pero usando los rotores de la derecha y de la izquierda. El movimiento de rotación sobre el mismo se obtienen con cada par de rotores, es decir, se acelera los dos rotores con sentido horario mientras se desacelera los rotores con sentido anti-horario, y viceversa.

- **Etapa de Potencia – Driver**

Esta etapa del sistema es importante, ya que es la etapa que nos proporciona el control de los motores, nos va a convertir la información de nuestro centro de control / emisor, y va a proporcionar la señal adecuada para el control de los motores. Un aspecto en esta etapa es la corriente que vamos a tener circulando por el sistema, ya que, según el motor que se utilice deberemos elegir un diseño u otro. Podemos encontrar muchos esquemas de control, pero todos se basan en la misma idea pero ofreciendo características diferentes como por ejemplo la

potencia máxima que pueden soportar. Podemos encontrar circuitos integrados que nos proporcionen todo lo necesario o podemos optar por circuitos con componentes discretos. Con los motores brushless existen controladores que proporcionan todo lo necesario para utilizar estos motores. Para conseguir variaciones de velocidad, ya sea desde nuestra emisora radiocontrol o desde una unidad de control se utiliza habitualmente una modulación por anchura de pulso (PWM). Dicha señal la podemos generar desde un circuito integrado o lo más usual desde un microcontrolador.

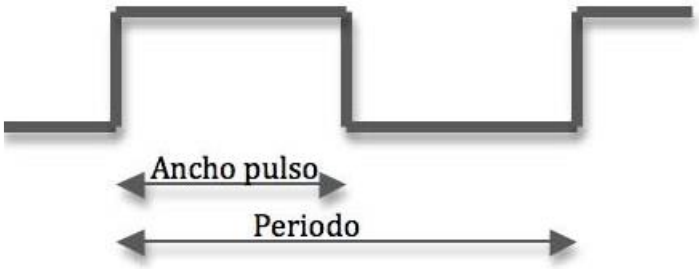


Figura 5*: PWM*

La señal PWM se utiliza como técnica para controlar circuitos analógicos. El periodo y la frecuencia del tren de pulsos pueden determinar la potencia entregada a dicho circuito. Si, por ejemplo, tenemos un voltaje de 9 V y lo modulamos con un ciclo de trabajo del 10%, obtenemos 0.9 V de media. Para entenderlo mejor, realmente estamos alimentando el motor y apagándolo en períodos de tiempo del orden de milisegundos. Gracias a la inercia del propio motor estos cambios provocan que se mueva a una velocidad constante. Así pues, cuanto mayor sea el ancho del pulso, mayor velocidad obtendremos.

- **MOTORES** (3)

Los motores más utilizados en aeromodelismo son los motores DC y los motores Brushless.

- **Motores DC**

Los motores de corriente continua son muy utilizados en mini robots. Podemos encontrar motores de este tipo en prácticamente todos los coches teledirigidos y/o coches de juguete y los ventiladores de mano. El rotor es el dispositivo que gira en el centro del motor DC y está compuesto de arrollados de cable conductores de corriente. Esta corriente es suministrada al rotor por medio de las "escobillas" generalmente fabricadas de carbón. La fuerza con la que el motor gira es proporcional a la corriente que hay por los conductores. A mayor tensión, mayor corriente y mayor par motor.

- **Motor Brushless**

Actualmente se utilizan mucho los motores brushless, o trifásicos. Estos motores son muy superiores a los motores DC en dos aspectos fundamentales: relación potencia-peso (también menor tamaño para la misma potencia) y eficiencia. Esto implica que la cantidad de energía eléctrica que se transforma en energía mecánica es mucho mayor. La palabra brushless se puede traducir como "sin escobillas", estos motores carecen de colector y escobillas o carbones. En vez de funcionar en DC funcionan en AC, la mayoría se alimenta con una señal trifásica, esta señal idealmente debería ser sinusoidal, pero en la práctica son pulsos.

Figura 6*: Motor Brushless*

Las ventajas de los motores Brushless son las siguientes:

- Mayor eficiencia (menos perdida por calor)
- Mayor rendimiento (mayor duración de las baterías para la misma potencia)
- Menor peso para la misma potencia
- Requieren menos mantenimiento al no tener escobillas
- Relación velocidad/par motor es casi una constante
- Mayor potencia para el mismo tamaño
- Rango de velocidad elevado al no tener limitación mecánica.

Las desventajas:

- Mayor coste de construcción
- El control es caro y complejo
- Siempre hace falta un control electrónico para que funcione, que a veces duplica el coste.

La tendencia en los sistemas cuadricóptero es utilizar los motores brushless, ya que ofrecen mayor rendimiento ante las baterías y mayor potencia.

- **ALIMENTACIÓN Y CONTROLADOR**

Uno de los principales problemas de los sistemas portables es la alimentación, la solución a este problema es la utilización de almacenadores.

Para proporcionar una rápida aceleración y duraciones considerables, las baterías de cualquier sistema autónomo tienen que proporcionar una gran potencia y energía. Además, han de ser de bajo precio, fáciles de mantener, seguras bajo condiciones de funcionamiento externo y tolerantes a los abusos a los que los vehículos aéreos están expuestos (aire libre, caídas, etc.). La capacidad de una batería es la cantidad total de carga producida en la reacción electroquímica y se define en unidades de coulombs (C) o amperios hora (Ah), que es la más usada como especificación de las baterías. Los parámetros a considerar de una batería son los siguientes:

- La tensión de salida, medida en voltios.
- La capacidad eléctrica, se mide por referencia a los tiempos de carga o de descarga en amperios hora (Ah) En las baterías se utiliza un múltiplo, el miliamperio hora (mAh). Este parámetro es muy importante y debemos fijarnos en él cuando compremos la batería pues cuanto mayor sea, más tiempo tardará en descargarse por el uso. El precio está en relación directa con este parámetro.

- **Li-Po (baterías de polímero de litio)**
  Son una variación de las Baterías Litio-Ion (Li-ion). Tienen una densidad de energía entre 5 y 12 veces las de Ni-Cd o las de Ni- MH. Son baterías mucho más ligeras y que pueden adoptar cualquier forma. No poseen efecto memoria. El voltaje de cada elemento es de 3.7 V. La desventaja es

que necesitan una carga mucho más lenta que las de Ni-Mh. Este tipo de baterías son ideales para alimentar motores muy potentes. Batería Li-Po Se debe cuidar de que el consumo máximo del motor sea menor que la descarga máxima de la batería para evitar que la vida de ésta se acorte demasiado. Su reducido tamaño y ligereza también las hace ideales para aeromodelos y helicópteros

En el diseño del cuadricoptero se han utilizado unas baterías tipo Li-Po igual que utiliza el UAV comercial Draganflyer, se ha elegido este tipo de batería principalmente por la gran densidad de carga que tienen en poco espacio y peso. Las mejores baterías desde el punto de vista capacidad / peso, muy importante en vehículos aéreos, son las de zinc-aire, pero aún no están comercializadas, aunque el ejército Israelí ya las utiliza en sus UAV. Actualmente se pueden encontrar baterías primarias, para aplicaciones sobretodo en audífonos, pero no secundarías.

Figura 7: ***Turnigy 2200mAh 3S 30C Lipo Pack***

- **Controlador de velocidad** (4)

ESC o variadores de velocidad electrónicos actúan como mensajero entre el controlador de vuelo y motores. Ellos proporcionan una señal pulsante a los motores para accionar un cambio en RPM y por lo tanto un movimiento

multicopter resultante. Un buen ESC es de suma importancia para el vuelo estable. Como motores sin escobillas son multi-etapas, normalmente 3 fases, que no sólo puede aplicar alimentación de CC para encenderlos. Un ESC es básicamente una tarjeta del controlador de motor que tiene un / entrada de señal de la batería y una salida trifásica para el motor. Cada ESC se controla de forma independiente por una señal PPM que oscila entre 1 ms (velocidad min = desactivar) de 2ms (velocidad máxima) de ancho de pulso. La frecuencia de las señales varía entre los fabricantes, pero para un multirotor se recomienda utilizar un ESC que soporta una alta tasa de refresco. Un ESC rápida permitirá a los motores para ajustar rápidamente la mejora de la respuesta del acelerador, aumentando drásticamente la estabilidad y proporcionar un más encerrado en sentimiento.

En aviones pequeños (indoor o muy sencillos) con motores de escobillas (brushed) son simples interruptores accionados por un canal del receptor o los más avanzados hacen gestión PWM (enviando impulso de anchura controlada) y regulan la velocidad del motor.

Estos sistemas son muy poco eficientes, hoy en día el boom de los eléctricos se debe a un tipo más sofisticado y que actualmente son los más habituales. Se obtienen potencias mucho mayores gracias a los motores Brushless que cuentan con un bobinado especial semejante al de los motores industriales trifásicos y aplicando mucha más frecuencia. Estos motores no se pueden conectar directamente a la batería sino que requieren de un circuito electrónico que los maneje: **el variador ESC**.

Los variadores pueden llegar a ser muy sofisticados y manejar capacidades de corriente muy elevadas (llegan a alcanzase los 200A en modelos extremos) obteniendo unas potencias que compiten con los modelos de gasolina.

En función del motor y tipo de vuelo que necesites puedes elegir entre una amplia gama de fabricantes. Suele ser una recomendación generalizada sobredimensionar el ESC, esto es, si del motor se espera un consumo máximo de 35A elegir un ESC que soporte al menos un 10-25% más. Lo cual nos garantiza que el variador responderá correctamente en condiciones de máxima carga.

Estos circuitos hacen una gestión avanzada del motor y requieren ser "programados" antes de usarlos. Para ello suelen ofrecer varios métodos, siendo la más simple a través de una secuencia de movimientos de la palanca de gas en la emisora al conectar, pasando por accesorios en forma de tarjeta con botones para las diferentes opciones, hasta los más sofisticados con conexión al ordenador.

Figura 8: ***Imagen del ESC***

- **Unidad de control**

La unidad de control es un dispositivo de los sistemas cuadricopteros bastante peculiar. Como todo sistema aéreo debe de disponer de unos sensores para medir parámetros como la aceleración de los ejes o la velocidad angular, esto ayuda al sistema a proporcionar la estabilización del mismo. La función más compleja de la unidad de control es la estabilidad del sistema, aunque debe de realizar otras funciones como el almacenamiento de datos entre otras. Las funciones principales de una unidad de control son las siguientes:

- Estabilizar el sistema mediante los sensores.

- Proporcionar comunicación estación base.
- Interpretar comandos de control enviados desde la estación base. o Guardar / enviar datos recogidos de los sensores.

- **Sensores**

Un sensor es un dispositivo que, a partir de la energía del medio donde se mide (altitud, temperatura, inclinación), crea una señal de salida que podemos interpretar mediante algún parámetro que se modifica en función a la variable medida.

En un sistema cuadricoptero podemos encontrar un gran número de sensores orientados a la estabilización del sistema y a la navegación. Algunos de los sensores más utilizados en estos sistemas son los acelerómetros y giroscopios, aunque en algunos sistemas más avanzados, también se pueden encontrar sensores de presión. Por otra parte, tenemos los sensores o equipos utilizados para la navegación, que van desde sensores de proximidad hasta receptores GPS.

- **Acelerómetro**

Los sensores utilizados para medir las aceleraciones son los denominados acelerómetros. Son sensores inerciales que miden la segunda derivada de la posición. Por tanto un acelerómetro mide la fuerza de inercia generada cuando una masa u objeto es afectado por un cambio de velocidad. Existen diferentes tipos de acelerómetros basados en varios tipos de tecnologías como son:

- **Tecnología capacitiva**: se basan en variar la capacidad entre dos o más conductores entre los cuales existe siempre un material dieléctrico.

- **Tecnología piezoeléctrica**: Al sufrir una deformación física del material se produce un cambio en la estructura cristalina y por consecuencia cambian sus características eléctricas.

- **Tecnología piezoresistiva**: Estos dispositivos basan su funcionamiento en la propiedad que tienen las resistencias eléctricas de cambiar su valor cuando el material se deforma mecánicamente.

- **Tecnología mecánica**: Estos acelerómetros utilizan bobinas, imanes para medir aceleración. Los componentes utilizados son una masa y resortes.

En sistemas de cuadricopteros podemos encontrar sensores de presión, estos sensores se emplean para limitar la altura del UAV, ya sea para corregir un posible fallo o para limitar alguna acción intencional. Otro sensor muy utilizado son los sensores de temperatura, este sensor se utiliza para corregir los pequeños errores que puedan producir los cambios de temperatura en los giroscopios, la desviación puede ser de mayor o menor importancia según el modelo utilizado de giroscopio.

En cualquier sistema en que se utilicen motores, tenemos que mantener monitorizada la temperatura de dichos motores. El sensor de temperatura nos puede ayudar a detectar posibles excesos de temperatura y así poder avisar al usuario o en casos extremos realizar alguna maniobra de aterrizaje y evitar un daño del sistema. Al tratarse de un sistema aéreo en la mayoría de los casos, agregan sensores más o menos complejos que ayudan a la navegación. Encontrarnos sensores de proximidad hasta receptores GPS, muchos de estos sistemas aéreos pueden programarse para que funcionen de forma autónoma gracias a estos sensores

.

Una de las aplicaciones más utilizadas en los sistemas cuadricopteros es la captura de imágenes aéreas, esta línea de aplicación se está investigando gracias al gran avance de las cámaras de video o fotografía. Podemos encontrar usuarios que incorporan a sus sistemas aéreos cámaras más sencillas para obtener una visión en directo de la imagen o usuarios que incorporan cámaras más sofisticadas que pueden proporcionar un video en alta definición HD.

- **Giroscopio**

Un giróscopo es un sólido rígido en rotación alrededor de un eje principal de inercia. Normalmente se monta sobre anillas en suspensión. De esta manera, ningún movimiento que realizase el conjunto causaría momento externo. Por tanto, el momento angular o momento cinético se conservaría y el eje de rotación mantendría una dirección fija en el espacio.

Existen giróscopos cuyo funcionamiento se basa en un elemento mecánico, la realidad es que la mayoría de los sensores actuales de pequeño tamaño, como los que se utilizan en modelos de helicópteros y robots, están basados en integrados con pequeñas lengüetas vibratorias, construidas directamente sobre el chip de silicio. Su detección se basa en que las piezas cerámicas en vibración están sujetas a una distorsión que se produce por el efecto Coriolis (cambios de velocidad angular). Como resultado de esta deformación el giroscopio genera un voltaje de salida que es proporcional a esta velocidad angular de rotación.

- **Magnetómetro**

Se llaman magnetómetros a los dispositivos que sirven para cuantificar en fuerza o dirección la señal magnética de una muestra. El magnetismo varía de lugar en lugar y a las diferencias en el campo magnético terrestre (la magnetósfera) y puede ser causada por las diferentes naturalezas de las rocas y la interacción entre las partículas cargadas del Sol y la magnetósfera de un planeta. Los

magnetómetros son un frecuente componente instrumental de naves espaciales que exploran planetas. (5)

- **GPS**

GPS, o Sistema de Posicionamiento Global (Global Positioning System) es un sistema de navegación basado en satélites y está integrado por 24 satélites puestos en órbita por el Departamento de defensa de los Estados Unidos. Originalmente, fue pensado para aplicaciones militares, aunque a partir de los años 80's el gobierno de USA puso el sistema de navegación disponible a la población civil. El GPS funciona en cualquier condición climatológica, en cualquier parte del mundo las 24 horas del día. No hay ningún costo de suscripción o cargos iniciales de preparación para usar el GPS.

- **Microcontrolador**

Los microcontroladores son los dispositivos programables más importantes de los sistemas digitales, los cuales son la base para el control de procesos en la industria, adquisición y procesado de datos. Todo desarrollo tecnológico moderno apoyado en la microelectrónica, que posee una unidad central de control, comprende a un microprocesador como el corazón mismo de dicha unidad; o bien a un dispositivo programable simplificado llamado microcontrolador.

Un microcontrolador es un tipo de computadora en miniatura que podemos encontrar en pequeños dispositivos electrónicos (radios, teléfonos, etc.). No debemos de confundir los microcontroladores con los microprocesadores, ya que estos últimos sólo es un unidad central de procesos, que de manera externa, hay que conectarle la memoria, puertos y unidades de almacenamiento, tiene la ventaja de ser más escalable que un microcontrolador, lo que nos da la capacidad de poder agregarle más componentes según sea nuestra necesidad.

Para el control del sistema cuadricoptero es suficiente utilizar un microcontrolador, la utilización de los conversores A/D, los contadores y la pequeña memoria (RAM y ROM) que incorporan hacen que estos dispositivos sean los más utilizados en este campo.

- **Sistema de Comunicación**

El control más común en estos sistemas UAV son las emisoras de radio control. La gran ventaja de utilizar emisoras de radio es la expansión que ha tenido durante muchos años en el mundo de aeromodelismo.

La principal limitación y causa de problemas es el uso de la misma frecuencia por más de un usuario. La segunda causa de problemas es la falta de estabilidad de emisoras y receptores, a los que pueden influir canales adyacentes. Utilizando buenos receptores este problema se minimiza mucho, pero si una emisora adyacente emite con mucha potencia y con bastante desviación del canal, entonces terminará influyendo y cuando el vehículo se aleje de nuestra emisora puede tener serios problemas.

Esto es una tecnología muy simple y antigua, comercializada hace varias décadas buscando la simplicidad para poder limitar el precio, tamaño y peso de los receptores. El avance en las comunicaciones inalámbricas abre un gran abanico en lo que se refiere a sistemas de comunicación. Podemos encontrar codificaciones muy eficientes y que consigan velocidades de transmisión importantes. Al tratarse de vehículos aéreos tendremos que tener en cuenta la distancia máxima que pueden alcanzar con las diferentes codificaciones. Podemos encontrar protocolos como Zigbee que proporcionen una distancia de comunicación en campo abierto de 1 km.

CAPITULO 3 ESTRUCTURA DEL CUADRICÓPTERO

En este capítulo se expone el diseño propio del sistema y las especificaciones técnicas de cada parte que compone el cuadricoptero.

- **Cuerpo del UAV**

Figura 9 ***Vista inferior y superior del cuerpo del UAV***

La estructura se compone de dos ejes perpendiculares formando una cruz simétrica. En sus extremos incorpora unos soportes para los motores. Toda la electrónica y batería está colocado en el centro de la estructura para ayudar a su ensamblaje.

- **Especificaciones de la estructura.**

Diámetro del cuerpo	450mm
Distancia entre hoyos de montaje del motor	16mm x 2/ 19mm x 2
Diámetro del hoyo principal	4mm
Peso necesario para volar	Inferior a 2Kg
Capacidad de carga	600~800g

- **Motores y Controlador de Velocidad**

Los motores escogidos son motores NTM Prop Drive Series 28-26A 1200kv sin escobillas. Está construido sin eje externo que le permite montar fácilmente este motor a su marco multi- rotor sin la necesidad de ninguna modificación.

Figura 10: ***Imagen del motor***

Rendimiento del motor: las siguientes imágenes muestran el rendimiento del motor de acuerdo al tipo de batería.

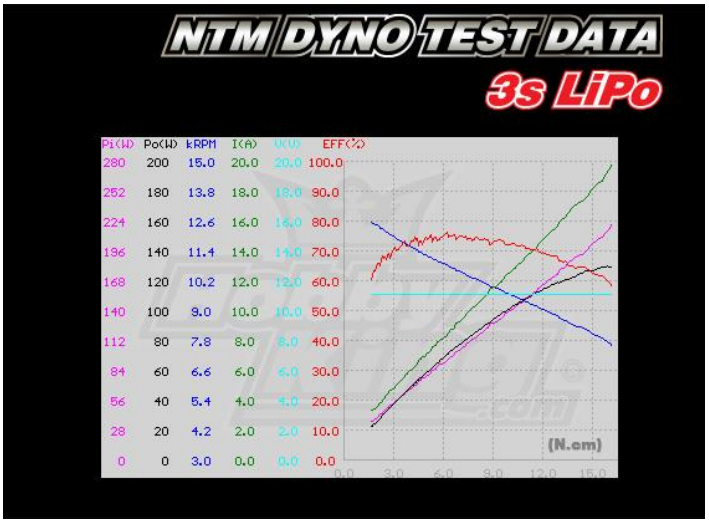


Figura 11: ***Rendimiento del Motor LiPo 3s***

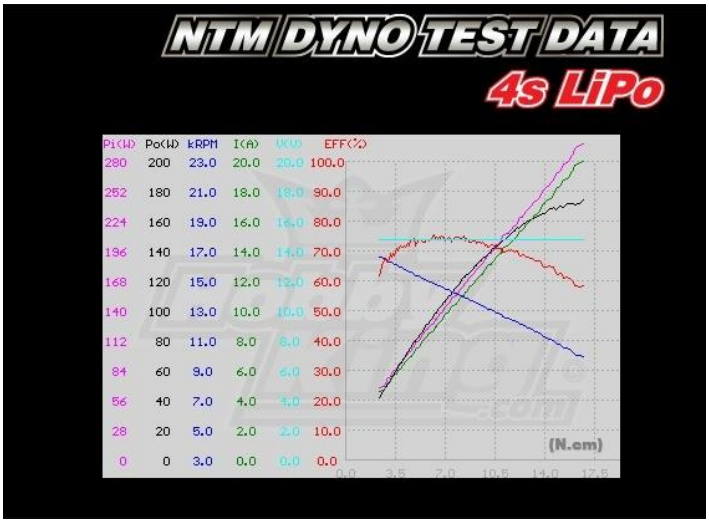
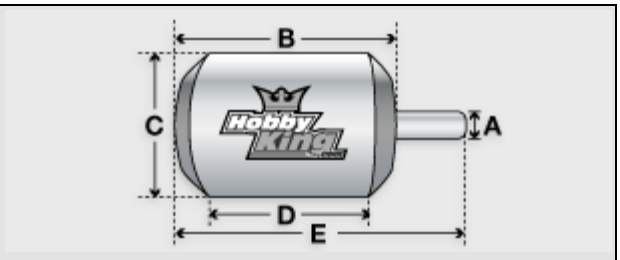


Figura 12: ***Rendimiento del motor LiPo 4s***

- **Especificaciones técnicas del motor:**

Modelo	NTM Prop Drive Series 28-26A 1200kv
Kv	1200 rpm/v
Corriente máxima	17A
Potencia máxima	215W @ 12v (3S) / 286W @ 15v (4S)
Eje	3mm
Peso	57.9g
ESC	20A
Cuenta de celdas	3s – 4s lipo
Hoyos para pernos	16mm & 19mm
Rosca de pernos	M3
Conexión	3.5mm conector bala

- **Dimensiones Del motor**

Eje A (mm)	-	
Largo B (mm)	26	
DIametro C (mm)	28	

Longitud (mm)	15	
Longitud Total E (mm)	42	

Figura 13: ***Dimensiones Del Motor***

- **Controladores de Velocidad**

Al elegir un ESC este debe de coincidir con el amperaje y voltaje de sus motores y la batería (LiPo). La calificación actual por lo general da como 2 cifras; un valor que representa la carga de corriente continua máxima y una figura de ráfaga más alta que corresponde a una corriente de corto plazo (permite normalmente durante no más de 10-30 segundos) por ejemplo, la producida en el despegue. Se recomienda que seleccione un controlador de velocidad que permita un margen de seguridad del 20%. La otra calificación principal ESC es la tensión máxima. Esto se expresa normalmente como el número de células, por ejemplo, un 6S LiPo (22.2V).

Una gran cantidad de los ESC incorporan un BEC (batería Eliminación Circuito). Esta es una característica muy útil, ya que significa que (en ciertas condiciones) no necesita una batería de receptor separado. Ellos trabajan por dimitir el voltaje de la batería principal a lo general 5V. Sin embargo, este proceso es ineficiente y genera calor; cuanto mayor sea el diferencial entre la tensión de la batería y 5V más será calor generado.

En consecuencia BEC (a menos que el tipo de cambio) sólo funcionan con un número determinado de células, a menudo sólo hasta un 3S LiPo y también dentro de una carga de corriente especificado. Si no se utiliza, la

ESC'sBEC debe ser desactivado mediante la eliminación de hilo medio (positivo) para el receptor.

Los ESC tienen tres conjuntos de cableado. Por un lado, dos cables, uno negro y uno rojo que van a la placa de la batería o la distribución de energía. También en el mismo lado, que normalmente tienen un cable de servo / receptor de 3 pines que normalmente se conecta a una salida del motor en el controlador de vuelo. Finalmente el otro lado comúnmente tiene 3 salidas, que se conectan a los cables del motor. Si el motor está girando en la dirección equivocada, simplemente intercambiar las posiciones de dos de estos cables.

También debe considerar cuidadosamente la posición de sus ESC dentro de su oficio. Ellos son fuentes concentradas de interferencias magnéticas, por lo que sitúan lejos de la electrónica de vuelo como la brújula.

Asegúrese de que estén ventilados, ya que son propensos a fallar si se sobrecalientan. Algunas personas recomiendan la colocación de ellos hacia fuera en los brazos de bastidor por debajo de los motores de manera que se enfrían por el lavado propio; otros advierten en contra de este enfoque, ya que son más vulnerables a los daños y cables más largos aumentan el riesgo de picos de tensión perjudiciales.

Por último, algunos ESC vienen pre-calibrados, sin embargo es recomendable calibrar todos los ESC antes de su uso, siguiendo las instrucciones del fabricante.

Figura 14: ***Imagen del ESC***

- **Especificaciones del controlador de velocidad**

Corriente	20A continuos
Rango de voltaje	2 – 4S Lipo
BEC	0.5ª lineal
Frecuencia de entrada	1KHz
Firmware	afro_nfet.hex
Cables descarga	16AWG/macho 3.5mm
Cable motor	16AWG/hembra 3.5mm
Peso	22.8g (incluye: cable, terminal, envoltura)
Tamaño	45 x 25.7 x 10.5 mm

- **Sensores**

  - **Barómetro y altímetro (BMP085)**

El sensor BMP085 permite lecturas de presión atmosférica y temperatura a través del bus I2C. Como la presión atmosférica cambia con la altura, también se puede

utilizar este sensor como un altímetro. El sensor se entrega soldado a un PCB listo para conectarse en un protoboard o tarjeta perforada.

Este sensor es fácil de usar con arduino debido a que ya se han escrito librerías que permiten leer la temperatura, presión atmosférica y altitud.

- Voltaje de operación: 3 a 5V
- Rango de operación: 300-1100 hPa (9000m a -500m sobre el nivel del mar)
- Hasta 0.03hPa / 0.25m de resolución
- Temperatura de operación: -40 to +85°C
- Precisión en medida de temperatura: +-2°C

Figura 15: ***Imagen del BMP085***

- **Especificaciones técnicas**

BMP085 Pin	Pin Function
VCC	Power (1.8V-3.6V)
GND	Ground
EOC	End of conversion output
XCLR	Master Clear (low-

	active)
SCL	Serial Clock I/O
SDA	Serial Data I/O

El BMP085 tiene una interfaz digital, I2C para ser específico. Esto significa que no puede estar un poco más arriba para conseguir comunicarse con su microcontrolador, pero a cambio obtiene datos que son mucho menos susceptibles al ruido y otros factores que pueden obstaculizar una señal analógica. I2C es una interfaz de dos hilos síncrono, el primer alambre, SDA, transmite los datos, mientras que un segundo alambre, SCL, transmite un reloj que se utiliza para realizar un seguimiento de los datos. Si se está utilizando un Arduino para hablar con el BMP085, la biblioteca Wire es la indicada para hacerse cargo de la mayor parte del trabajo en la comunicación con el sensor.

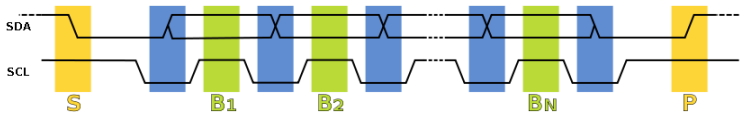


Figura 16: ***Transferencia de Datos I2C***

Hay una gran cantidad de proyectos que pueden hacer uso de los datos de presión barométrica. Usted probablemente ha escuchado sobre los sistemas de baja presión y sistemas de alta presión, que se debe a la presión atmosférica y puede estar directamente relacionado con los cambios en el clima. Baja presión normalmente significa más nublado, lluvioso, más nieve, y un tiempo en general más feo. Mientras alta presión generalmente significa más claro, el clima más soleado. Esto significa que el BMP085 sería un componente perfecto para su proyecto.

Un segundo uso, ampliamente aplicado por los sensores de presión está en altimetría. La presión barométrica tiene una relación mensurable con la altitud, lo que significa que puede utilizar el BMP085. La presión atmosférica a nivel del mar es, en promedio, 1.013 hPa, la presión del aire promedio es de 831.4 hPa. Los límites de medición del BMP085 deben permitirá medir la presión en las elevaciones en cualquier lugar entre -1640 a unos 29.000 pies sobre el nivel del mar.

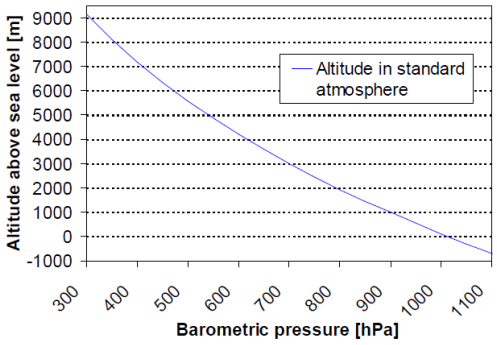


Figura 17: ***Presión Barométrica***

Un vistazo a la parte inferior de la BMP085 Breakout nos muestra las etiquetas de todos los seis pines: "SDA", "SCL", "XCLR", "EOC", "GND, y 'VCC'. VCC y GND son, evidentemente, los pines de alimentación. SDA y SCL son las líneas de comunicación I2C. SDA siendo donde se transmiten los datos, y SCL es el reloj que realiza un seguimiento de los datos. Los últimos dos alfileres, XCLR y EOC, hay un par de funciones adicionales del BMP085. XCLR actúa como un reinicio maestro. Es baja activa, por lo que si se tiró a GND se restablecerá la BMP085 y establecer sus registros a su estado predeterminado. EOC, que significa "final de la conversión", es una señal generada por el BMP085 que se activa cada vez que una conversión de la presión o la temperatura han terminado. Estas últimas dos pasadores son opcionales, y si no es necesario usarlos se pueden dejar sin conectar.

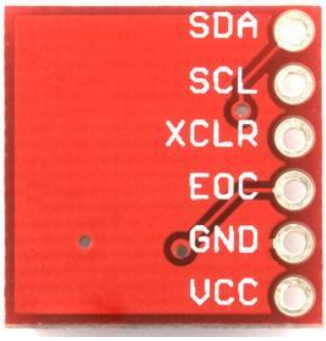


Figura 18*: Etiqueta de los pines*

- **Conexión y lectura de datos del Barómetro y altímetro (BMP085):**

Esta es la conexión del sensor al arduino:

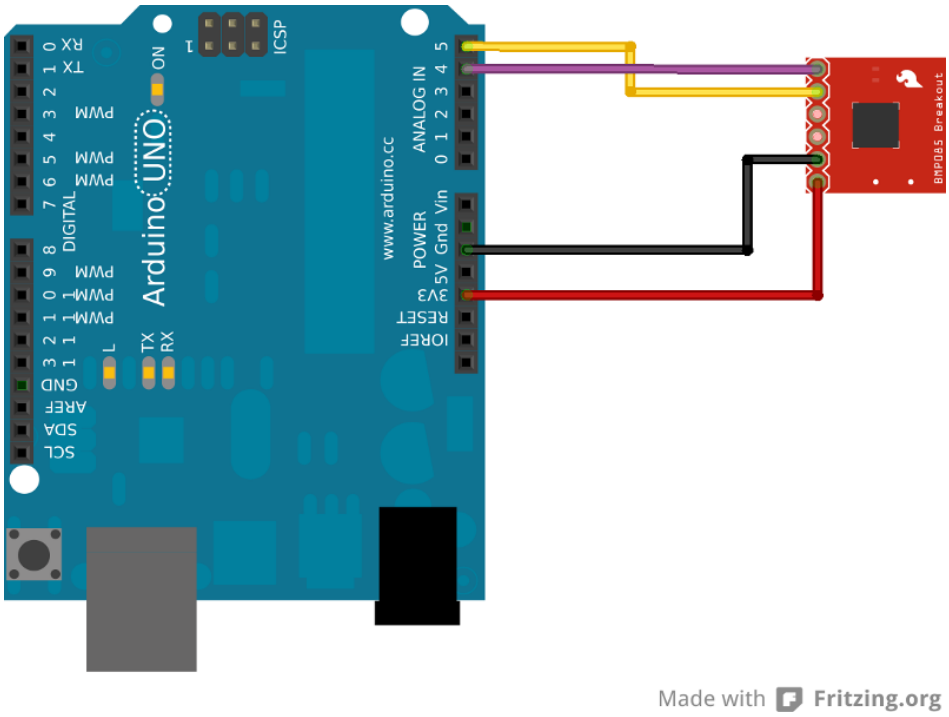


Figura 19: ***Conexión Sensor - Arduino***

Pin del barómetro	Pin de Arduino
SDA	A4
SCL	A5
Vcc	3.3V
GND	GND

Es muy importante utilizar estos pines, ya que la comunicación se realiza por *i2C* y estos pines son los que cuentan con este tipo de conexión en la mayor parte de los Arduinos (todos los basados en los chips 168 o 328).

Otro punto fundamental es la alimentación. Algunos sensores BMP085 vienen integrados en un módulo que cuenta con un regulador de tensión, lo que nos permite conectarlo directamente a 5V, pero no en todos los casos es así.

El sensor funciona a 3.3V, por lo que conectarlo a 5V sin protección lo destruirá de forma definitiva.

Una vez lo tenemos conectado, es el momento de empezar a pedirle lecturas.

- **Acelerómetro y giroscopio (MPU-6050)**

El MPU-6050 combina un giroscopio de 3-ejes y un acelerómetro de 3-ejes en el mismo chip. Así mimo incorpora un procesador de movimiento capaz de procesar una compleja fusión de movimiento en 9 ejes. Se comunica a través de la interfaz I2C y posee una librería muy difundida para su utilización inmediata.

Este sensor puede entregar 6 grados de libertad (DOF) e incorpora un regulador de tensión a 3.3V y resistencias pull-up para su uso directo por I2C.

Figura 20: ***Imagen del MPU-6050***

- **Especificaciones MPU-6050**

- Tensión de alimentación: 3~5v
- Interfaz digital: I2C Digital
- Sensor angular de 3 ejes (giroscopio) con una sensibilidad de hasta 131 LSBs/dps y a full-escala de ±250, ±500, ±1000, and ±2000dps
- Acelerómetro de 3 ejes con una precisión programable en rangos de ±2g, ±4g, ±8g and ±16g
- Algoritmos embebidos para mejorar la calibración sin intervención del usuario.
- Dimensiones: 14 x 21mm

- **Conexión y lectura de datos del Acelerómetro y giroscopio (MPU-6050)**

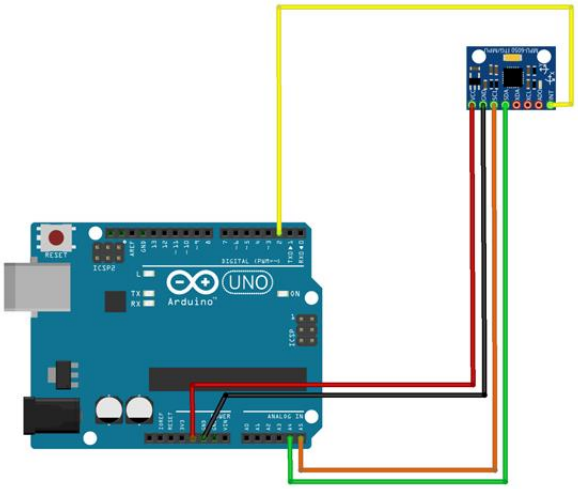


Figura 21: ***Conexión Arduino – MPU (6050)***

- MPU Vcc ->Arduino 3.3v (o 3v3 en algunos modelos)
- MPU Gnd ->ArduinoGnd
- MPU SCL ->Arduino A5
- MPU SDA ->Arduino A4

La web oficial de Arduino dispone de un ejemplo para leer datos de la MPU-6050. Partiré de ella, aunque habrá que modificarla considerablemente ya que sólo da los valores sin refinar del Giroscopio y el Acelerómetro.

#define MPU 0x68 es la dirección I2C de la IMU que se especifica en la documentación oficial.

Los ratios de conversión son los especificados en la documentación. Deberemos dividir los valores que nos den el Giroscopio y el Acelerómetro entre estas

constantes para obtener un valor coherente. RAD_A_DEG es la conversión de radianes a grados.

La IMU da los valores en enteros de 16 bits. Como Arduino los guarda en menos bits, hay que declarar las variables que almacenarán los enteros provenientes de la IMU como un tipo de enteros especiales. int16_t AcX, AcY, AcZ, GyX, GyY son, pues, los raw_values de la IMU.

- **Magnetómetro GY – 271**

  El módulo de brújula está diseñado para la detección magnética de bajo campo con una interfaz digital para dar información sobre el rumbo preciso.

  Este sensor compacto encaja en pequeños proyectos tales como vehículos aéreos no tripulados y sistemas de navegación, este sensor convierte cualquier campo magnético a una salida de tensión diferencial en 3 ejes.

  Este cambio de voltaje es el valor de salida digital, que luego se puede utilizar para calcular las partidas o sentido magnético de los campos que vienen de diferentes direcciones.

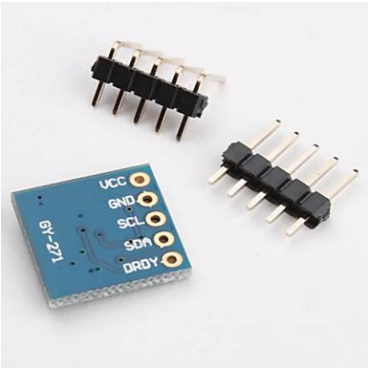


Figura 22: ***GY-271*** *(6)*

Especificaciones:

- Voltaje DC 3V-5V
- Chipset HMC5883L
- Protocolo Comunicación vía I2C
- Rango de Medición: ± 1.3-8 Gauss
- Dimensiones 14.8 x 13.5 x 3.5mm

Configuración de los pines:

- VCC: 3V-5V DC
- GND: tierra
- SCL: Entrada Análoga (A5)
- SDA: Entrada Análoga (A4)
- DRDY: No Conectado

- **GPS ublox neo-6m** (7)

La serie NEO- 6 es una familia de stand-alone GPS receptores que ofrecen un alto rendimiento con un motor de posicionamiento u-blox 6. Estos receptores flexibles y rentables ofrecen numerosas opciones de conectividad en una miniatura 16 x 12,2 x 2,4 mm. Su arquitectura compacta y opciones de energía y de memoria hacen NEO - 6 el módulo ideal para la batería operada por dispositivos móviles con muy estrictas limitaciones de coste y espacio.

El u- blox 6 de 50 canales cuenta con un Time-To -Fisto -Fix (TTFF) de menos de 1 segundo. El motor dedicado de adquisición, con 2 millones de

correlacionadores, es capaz de búsquedas paralelas masivas de espacio de tiempo / frecuencia, lo que le permite encontrar satélites al instante.

Diseño y tecnología innovadora suprime las fuentes de interferencia y mitiga los efectos de trayectoria múltiple, dando a los receptores GPS NEO - 6 excelente performance de navegación, incluso en los más difíciles entornos.

Figura 23: ***Imagen GPS ublox neo-6m***

A continuación se describen algunas de las características del módulo:

- **Características**

- Modulo GPS Ublox NEO-6M
- Comunicación serial
- Voltaje de alimentación: (3.5 – 5)VDC
- Antena cerámica activa incluida
- LED indicador de señal
- Tamaño de antena 22x22mm
- Tamaño de módulo 23x30mm
- Batería incluida
- BAUDRATE: 9600
- EEPROM para guardar configuración de parámetros
- Sistema de coordenadas: WGS-84

- Sensibilidad de captura -148dBm
- Sensibilidad de rastreo: -161 dBm
- Máxima altura medible: 18000
- Máxima velocidad 515 m/s
- Exactitud: 1micro segundo
- Frecuencia receptora: L1 (1575.42 Mhz)
- Código C/A 1.023 Mhz
- Tiempo de inicio primera vez: 38s en promedio
- Tiempo de inicio : 35s en promedio

- **Cerebro del Sistema**

El Arduino Mega 2560 es una placa electrónica basada en el Atmega2560 Cuenta con 54 pines digitales de entrada / salida (de los cuales 15 se pueden utilizar como salidas (PWM), 16 entradas analógicas, 4 UARTs (hardware puertos serie), un oscilador de cristal de 16 MHz, una conexión USB, un conector de alimentación, un header ICSP, y un botón de reinicio. Contiene todo lo necesario para apoyar el microcontrolador; simplemente conectarlo a un ordenador con un cable USB o el poder con un adaptador de CA o la batería a CC para empezar. El Mega es compatible con la mayoría de los escudos diseñados para el ArduinoDuemilanove o Diecimila.

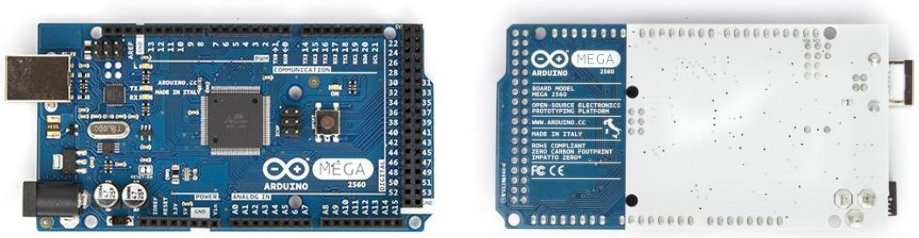


Figura 24: ***Vista Frontal y Posterior del Arduino Mega 2560***

- **Resumen de especificaciones:**

Microcontrolador	ATmega2560
Voltaje	5V
Voltaje de Entrada(Recomendado)	7-12V
Voltaje de Entrada(Limite)	6-20V
Pin Digital I/O	54 (de los cuales 15 proporcionan salida PWM)
Pin Entrada Digital	16
Corriente DC por Pin I/O	40 mA
Corriente DC por Pin 3.3V	50 mA
Memoria Flash	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
ClockSpeed	16 MHz

El Mega Arduino puede ser alimentado a través de la conexión USB o con una fuente de alimentación externa. La fuente de alimentación se selecciona automáticamente.

El tablero puede funcionar con un suministro externo de 6 a 20 voltios. Si se suministra con menos de 7V, sin embargo, el pin de 5V puede suministrar menos de cinco voltios y puede ser inestable. Si se utiliza más de 12V, el regulador de voltaje se puede sobrecalentar y dañar la placa. El rango recomendado es de 7 a 12 voltios.

Los pines de alimentación son como sigue:

- VIN. El voltaje de entrada a la placa Arduino cuando se utiliza una fuente de alimentación externa (por oposición a 5 voltios de la conexión USB u otra fuente de alimentación regulada). Usted puede suministrar tensión a través de este pin, o, si el suministro de tensión a través de la toma de poder, acceder a él a través de este pin.

- 5V. Este pin como salida una 5V regulada del regulador en el tablero. El tablero puede ser alimentado ya sea desde la toma de alimentación de CC (7 - 12 V), el conector USB (5V), o el pin VIN del tablero (7-12V). El suministro de tensión a través de los pines de 5V o 3.3V no pasa por el regulador, y puede dañar su tablero. No aconsejamos ella.

- 3V3. Un suministro de 3,3 voltios generada por el regulador de a bordo. Empate Corriente máxima es de 50 mA.

- GND. Patillas de tierra.

- Instrucción IOREF. Este pin de la placa Arduino proporciona la referencia de tensión con la que opera el microcontrolador. Un escudo configurado puede leer el voltaje pin instrucción IOREF y seleccione la fuente de alimentación adecuada o habilitar traductores de voltaje en las salidas para trabajar con el 5V o 3.3V.

- **Memoria**

El Atmega2560 tiene 256 KB de memoria flash para almacenar código (de los cuales 8 KB se utiliza para el cargador de arranque), 8 KB de SRAM y 4 KB de EEPROM (que puede ser leído y escrito con la biblioteca EEPROM).

- **Entrada y Salida**

Cada uno de los 54 pines digitales en el Mega se puede utilizar como una entrada o salida, utilizando pinMode (), digitalWrite (), y las funciones digitalRead ().

Operan en 5 voltios. Cada pin puede proporcionar o recibir un máximo de 40 mA y tiene una resistencia de pull-up (desconectado por defecto) de 20 a 50 kOhm. Además, algunos pines tienen funciones especializadas:

- Serial: 0 (RX) y 1 (TX); Serial 1: 19 (RX) y 18 (TX); Serial 2: 17 (RX) y 16 (TX); Serial 3: 15 (RX) y 14 (TX). Se utiliza para recibir (RX) y transmitir datos en serie (TX) TTL. Pernos 0 y 1 también están conectados a los pines correspondientes de los USB-to-TTL chips Serial ATmega16U2.

- Interrupciones externas: 2 (interrumpir 0), 3 (interrumpir 1), 18 (interrumpir 5), 19 (interrumpir 4), 20 (interrumpir 3), y 21 (interrumpir 2). Estos pines pueden configurarse para activar una interrupción en un valor bajo, un flanco ascendente o descendente, o un cambio en el valor. Ver la función attachInterrupt () para más detalles.

- PWM: 2 a 13 y 44 a 46. Proporcionar una salida PWM de 8 bits con la función analogWrite ().

- SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS). Estos pines soportan la comunicación SPI utilizando la librería SPI. Los pines SPI también se desglosan en la cabecera ICSP, que es físicamente compatible con el Uno, Duemilanove y Diecimila.

- LED: 13. Hay un LED incorporado conectado al pin digital 13. Cuando el pasador es de alto valor, el LED está encendido, cuando el pasador es bajo, es apagado.

- TWI: 20 (SDA) y 21 (SCL). Comunicación TWI Apoyo mediante la biblioteca de alambre. Tenga en cuenta que estos pines no están en la misma ubicación que los pines TWI en el Duemilanove o Diecimila.

El Mega2560 tiene 16 entradas analógicas, cada una de las cuales proporcionan 10 bits de resolución (es decir, 1.024 valores diferentes). Por defecto se miden

desde el suelo a 5 voltios, aunque es posible cambiar el extremo superior de su rango usando el pin AREF y función analogReference ().

Hay un par de patas de la placa:

- AREF. Tensión de referencia para las entradas analógicas. Se utiliza con analogReference ().

- Restablecer. Traiga esta línea BAJO para reajustar el microcontrolador. Normalmente se utiliza para añadir un botón de reinicio para escudos que bloquean el uno en el tablero.

- **Comunicación Arduino Mega2560**

El Arduino Mega2560 tiene una serie de instalaciones para la comunicación con un ordenador, otro Arduino, u otros microcontroladores. El Atmega2560 ofrece cuatro UART hardware para TTL (5V) de comunicación serie. Un ATmega16U2 (ATmega 8U2 sobre la revisión y revisión 1 2 tablas) en los canales de mesa uno de ellos a través de USB y proporciona un puerto com virtual para software en el equipo (máquinas Windows necesitarán un archivo .inf, pero las máquinas OSX y Linux reconocer el tablero como un puerto COM de forma automática. El software de Arduino incluye un monitor de serie que permite a los datos textuales sencillos para ser enviados hacia y desde el tablero. Los LEDs RX y TX de la placa parpadean cuando se están transmitiendo datos a través de la ATmega8U2 / ATmega16U2 chip y conexión USB al ordenador (pero no para la comunicación en serie en los pines 0 y 1).

Una biblioteca Software Serial permite la comunicación en serie en cualquiera de los pines digitales del Mega2560.

El Atmega2560 también soporta la comunicación TWI y SPI. El software Arduino incluye una biblioteca de alambre para simplificar el uso del bus TWI; consulte la

documentación para obtener más información. Para la comunicación SPI, utilizar la biblioteca SPI.

- **SISTEMA DE COMUNICACIÓN**

- **TRANSCEPTORRF NRF24L01 CON ANTENA**

Figura 25: ***Modulo transceptor NRF24L01***

Este módulo usa un nuevo transceptor de la compañía Nordic Semiconductor, el nRF24L01+. Este transceptor opera en la banda de 2.4GHz y tiene muchas características nuevas. Incluye circuitería tanto de PA (preamplificador) como de LNA (Low-noiseamplifier), con la antena externa puede alcanzar distancias más grandes que las versiones normales. Es similar al nRF2401+ e incorpora adicionalmente extra pipelines, buffers, y un modo de auto retransmisión.

Viene con la antena para 2.4GHz (2dB) con una tasa de trasmisión de hasta 250kbps al aire libre y puede alcanzar una distancia de hasta 1 kilómetro.

El módulo utiliza 8 pines para su interfaz externa y funciona con el estándar de comunicación SPI.

- **Características NRF24L01**

Voltaje de alimentación: 3-3.6V (3.3V recomendado)
Potencia máxima de salida: +20 dBm
Corriente (máxima) en modo de emisión: 115 mA
Corriente (máxima) en modo de recepción: 45 mA
Modo de apagado-espera: 4.2uA
Sensibilidad a 2 Mbps en modo de recepción: -92dBm
Sensibilidad a 1 Mbps en modo de recepción: -95 dBm
Sensibilidad a 250 kbps en modo de recepción: -104dBm
Ganancia PA : 20dB
Ganancia LNA: 10dB
Figura de ruido LNA: 2.6 dB
Ganancia de la Antena (pico): 2 dBi
Distancia a Tasa de 2MB (zona abierta): 520m
Distancia a Tasa de 1MB (zona abierta): 750m
Distancia a Tasa de 250Kb (zona abierta): 1000m

- **RF-NRF24L01-2-4-GHZ-CON-ANTENA**

Línea	**Pin del Arduino**	**Posición de la Cuadricula**	**Pin de radio**	**Posición de la cuadricula**
GND	GND	U17	1	N17
3V3	3V3	U14	2	N14
CE	9	D15	3	M17
CSN	10	D14	4	M14
SCK	13	D11	5	L17
MOSI	11	D13	6	L14
MISO	12	D12	7	K16

- **ALIMENTACIÓN (Turnigy 2200mAh 3S 30C Lipo Pack)**

Las Baterías Turnigy son conocidas en todo el mundo por su rendimiento, fiabilidad y precio. Las Baterías Turnigy ofrecen toda la capacidad nominal a un precio accesible a todo el mundo.

Estas Baterías están equipadas con descarga de alta resistencia que conduce a minimizar la resistencia y mantener altas cargas de corriente.

Cada paquete de batería está equipado con conectores chapados en oro y conectores de equilibrio estilo JST - XH. Todas las baterías TurnigyLipoly pack se montan utilizando células IR emparejado.

Figura 26: ***Batería Turnigy 2200mAh 3S 30C Lipo Pack***

- **Especificaciones Turnigy 2200mAh**:

  Capacidad Mínima: 2200mAh

  Configuración: 3S1P / 11.1v / 3Cell

  Descargue Constante: 30C

  Descargue Pico (10sec): 40C

  Peso 197g

  Tamaño: 108 x 34 x 27mm

  Enchufe de carga: JST-XH

  Enchufe descarga: XT60

- **GIMBAL ALEXMOS 2 EJES**

Este Gimbal está especialmente diseñado para quadcopters como el DJI Phantom y el Walkera QR X350, pero funciona en cualquier Multirotor e incluso como Steadycam en sus manos.

Figura 27*: Imagen Gimbal Alexmos*

- **Especificaciones del Gimbal**

Estructura simple y liviana, Peso: 205 gramos
2 ejes con motores Brushless
Posee cauchos anti vibratorios
Controlador Alexmos versión actualizada + sensor
Colores: GRIS y NEGRO
Diseñado para cámaras GOPRO o similares
Viene armado y listo para poner en su Drone

- **Joystick**

El módulo Joystick tiene 5 pines: Vcc, GND, VRX, VRYY Y SW. Hay que tener en cuenta que las etiquetas de otros modelos pueden ser ligeramente diferente, dependiendo de dónde se compró el módulo.

El joystick es analógico y debe proporcionar lecturas más precisas que las simples palancas direccionales. Además, se puede pulsar el joystick hacia abajo para activar un pulsador.

Se tienen que usar pines analógicos de Arduino para leer los datos de los pines X / Y, y un pin digital para leer el botón SW. El pasador SW está conectado a tierra, cuando se pulsa el joystick hacia abajo. Para obtener lecturas estables necesita estar conectado a Vcc a través de una resistencia pull-up.

Figura 28: ***Imagen del joystick***

Especificaciones:

- 3 ejes: X, Y (potenciómetros) y Z (botón)
- Potenciómetros (X y Y): 2 x 5Kohm
- Botón: abre normal (Z)
- Conexión de gaza: 2.54mm
- Dimensiones: 37 x 32 x 25 mm
- Peso: 15g

Pin modo de Conexión:
1.GND
2.+5Vcc
3.vRX
4.vRY
5.SW

- **Asignación de pines Arduino – UAV**

Pin Arduino	Pin equipo (UAV)	Equipo conectado
GND	GND	BATERIAL LIPO 3S
A3	CELDA 1	
A2	CELDA 2	
A1	CELDA 3	
SDA 20	SDA	BMP085
SCL 21	SCL	
GND	GND	
3.3V	VCC	
5V	VCC	MPU-6050
GND	GND	
SDA 20	SDA	
SCL 21	SCL	
PWM 2	INT	
PWM 4	SEÑAL (Amarillo)	ESC 1
GND	(cafe)	
PWM 5	SEÑAL (Amarillo)	ESC 2
GND	(cafe)	
PWM 6	SEÑAL (Amarillo)	ESC 3
GND	(cafe)	
PWM 7	SEÑAL (Amarillo)	ESC 4
GND	(cafe)	
GND	GND	NRF24L01 (Transceptor 2.4 GHz)
3.3V	VCC	
46 (PWM)	CE	
53	CSN	
51	MOSI	
52	SCK	
50	MISO	
-	INT	
GND	GND	GPS UBLOX NEO 6M
5V	VCC	
TX1 18	TX	
RX1 19	RX	

- **Asignación pines Arduino – control**

Pin Arduino	Pin equipo	Equipo conectado
GND	GND	NRF24L01 (Transceptor 2.4 GHz)
3.3V	VCC	
46 (PWM)	CE	
53	CSN	
51	MOSI	
52	SCK	
50	MISO	
-	INT	
GND	GND	JOYSTICK IZQUIERDO (Potencia – velocidad - altura)
5V	5V	
A2	VRX	
A1	VRY	
38	SW	
GND	GND	JOYSTICK DERECHO (Dirección - orientación)
5V	5V	
A9	VRX	
A8	VRY	
42	SW	

- **INTERFAZ GRÁFICA**

Se diseñó e implementó una interfaz gráfica con el fin de observar la posición del Drone, es decir poder determinar los grados de inclinación tanto vertical como horizontalmente, que este presenta durante un vuelo, además de esto también se puede observar el sentido de giro y hacia qué dirección se está desplazando. El

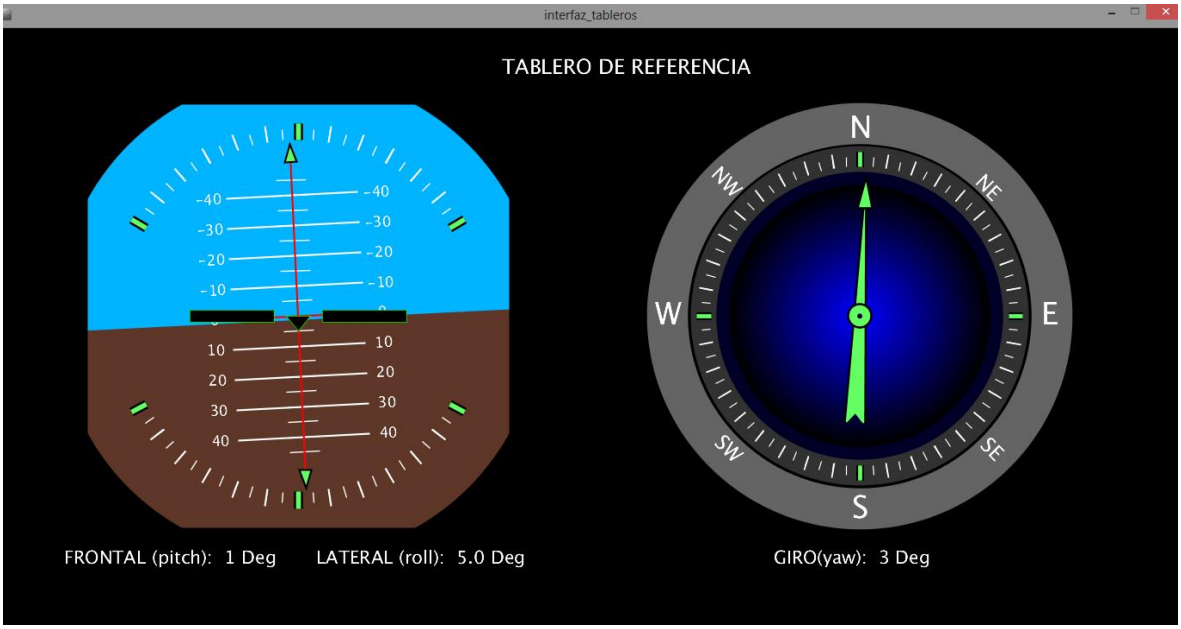
integrado MPU 60-50 que como decíamos anteriormente cuenta con un giroscopio y un acelerómetro ambos de tres ejes, es el encargado de darnos esta información por medio de las lecturas que arrojan estos sensores.

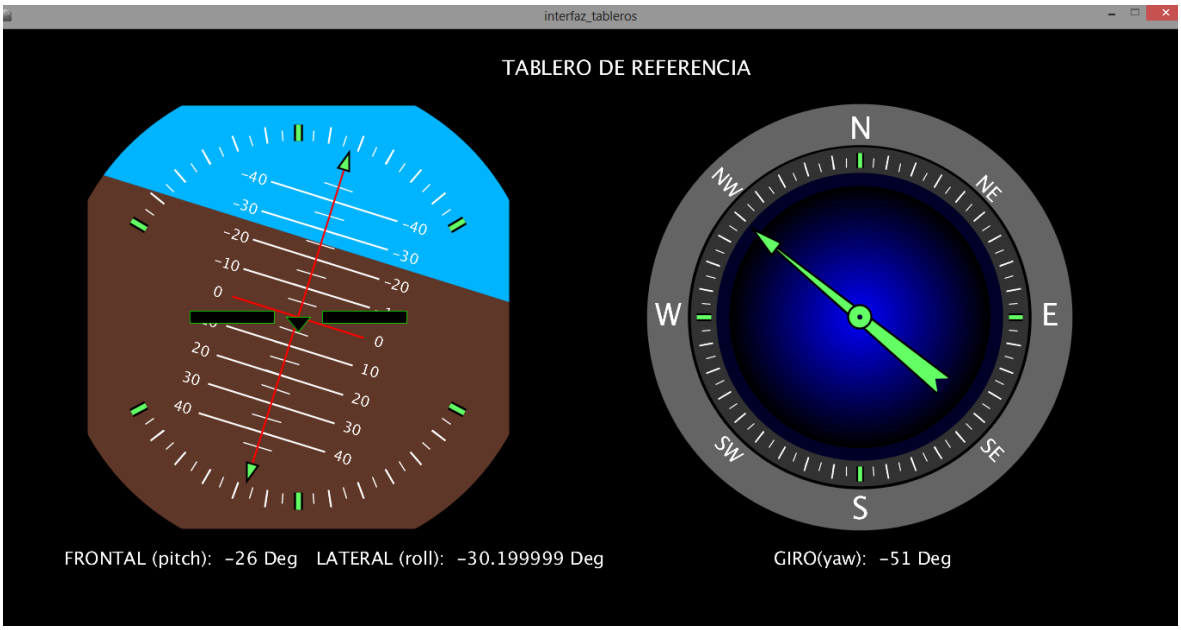
La interfaz fue elaborada por medio del programa Processing que “es un dialecto de Java específicamente diseñado para el desarrollo de arte gráfico, animaciones y aplicaciones gráficas de todo tipo, desarrollado por artistas y para artistas, Processing es software libre, disponible en las plataformas donde Java está disponible (Mac OS, Linux, Windows etc.).

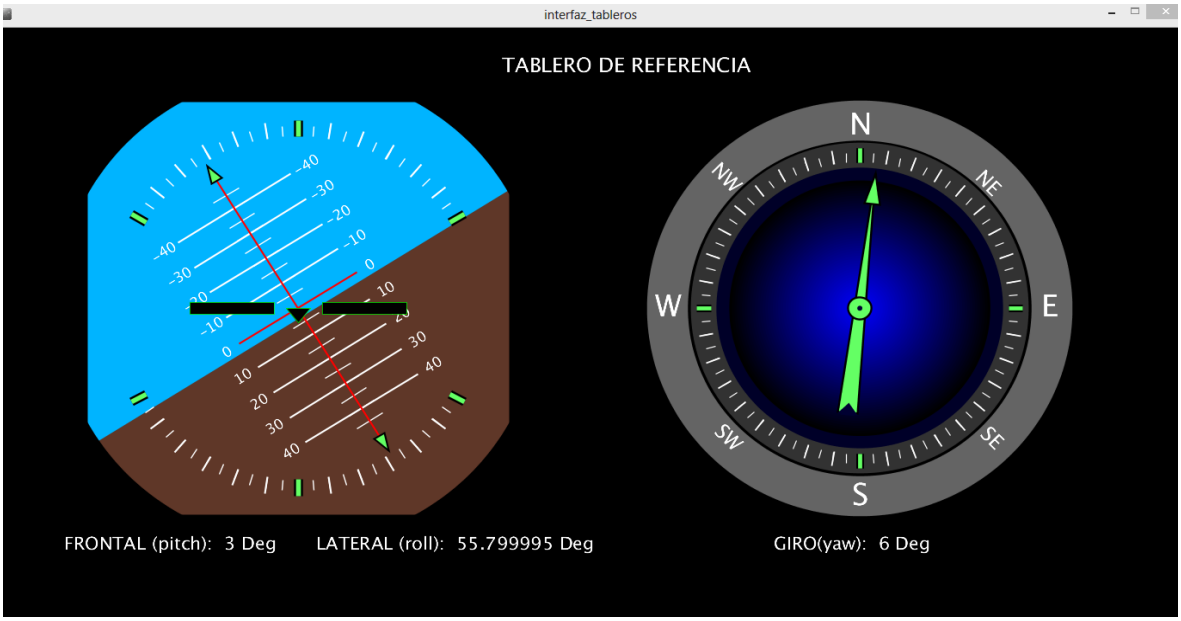
Desarrollado a partir del 2001 en el MIT por Casey Reas y Ben Fry. Inspirado en DBN (Design By Numbers) de John Maeda. (8)

En el código de arduino, la función “LeerMPU()“ nos permite la lectura de los sensores por medio de la liibreria Wire.h que se puede encontrar en la página oficial de arduino. Posteriormente una vez teniendo estos datos con la función “ConvertirAngulos () “se convierten los valores leídos de los sensores a grados para ser enviados a processing.

A continuación se muestran algunas imágenes de la interfaz en funcionamiento







El código utilizado en Arduino fue el siguiente:

```
//función para la lectura de los sensores del mpu
```

```
void Leer_MPU(){

  // Lectura MPU

  Wire.beginTransmission(MPU);

  Wire.write(0x3B);  // starting with register 0x3B (ACCEL_XOUT_H)

  Wire.endTransmission(false);

  Wire.requestFrom(MPU,14,true);  // request a total of 14 registers

  AcX=Wire.read()<<8|Wire.read();  //  0x3B  (ACCEL_XOUT_H)  &  0x3C
(ACCEL_XOUT_L)

  AcY=Wire.read()<<8|Wire.read();  //  0x3D  (ACCEL_YOUT_H)  &  0x3E
(ACCEL_YOUT_L)

  AcZ=Wire.read()<<8|Wire.read();  //  0x3F  (ACCEL_ZOUT_H)  &  0x40
(ACCEL_ZOUT_L)

  Tmp=Wire.read()<<8|Wire.read();  //  0x41  (TEMP_OUT_H)  &  0x42
(TEMP_OUT_L)

  GyX=Wire.read()<<8|Wire.read();  //  0x43  (GYRO_XOUT_H)  &  0x44
(GYRO_XOUT_L)

  GyY=Wire.read()<<8|Wire.read();  //  0x45  (GYRO_YOUT_H)  &  0x46
(GYRO_YOUT_L)

  GyZ=Wire.read()<<8|Wire.read();  //  0x47  (GYRO_ZOUT_H)  &  0x48
(GYRO_ZOUT_L)

}

//Convertir a ángulos lecturas
```

```
Void ConvertirAngulos(){

    // display Euler angles in degrees

        Serial.print(AcZ * 180/M_PI);

        Serial.print("\t");

        Serial.print(AcY * 180/M_PI);

        Serial.print("\t");

        Serial.println(AcZ * 180/M_PI);



        // display Euler angles in degrees

        Serial.print(  GyX * 18/M_PI);

        Serial.print(" ");

        Serial.print(  GyY * 180/M_PI);

        Serial.print(" ");

        Serial.println(  GyZ* 180/M_PI);

}
```

El código utilizado en processing fue el siguiente:

```
//importar librerias
```

```
import processing.serial.*;
import cc.arduino.*;
//resolucion de pantalla del pc
int W=1400;
int H=700;

float Pitch;
float Roll;
float Azimuth;
float ArtificialHoizonMagnificationFactor=0.7;
float CompassMagnificationFactor=0.85;
float SpanAngle=120;
int NumberOfScaleMajorDivisions;
int NumberOfScaleMinorDivisions;
PVector v1, v2;



Serial port;
float Phi;    //Dimensional axis
float Theta;
float Psi;
final String serialPort = "COM4"; // declaracion del puerto del arduino
void setup()
{
  size(W, H);
  rectMode(CENTER);
  smooth();
  strokeCap(SQUARE);//Optional

  println(Serial.list()); //Shows your connected serial ports
```

```
  port = new Serial(this,serialPort , 9600);
  //Up there you should select port which arduino connected and same baud rate.
  port.bufferUntil('\n');
}
void draw()
{
  //interfaz grafica
  background(0);
  translate(W/4, H/2.1);
  MakeAnglesDependentOnMPU6050();
  Horizon();
  rotate(-Roll);
  PitchScale();
  Axis();
  rotate(Roll);
  Borders();
  Plane();
  ShowAngles();
  Compass();
  ShowAzimuth();
}
void serialEvent(Serial port) //Leer datos de entrada
{
   String input = port.readStringUntil('\n');
  if(input != null){
   input = trim(input);
  String[] values = split(input, " ");
 if(values.length == 3){
  float phi = float(values[0]);
  float theta = float(values[1]);
```

```
  float psi = float(values[2]);
  print(phi);
  print(theta);
  println(psi);
  Phi = phi;
  Theta = theta;
  Psi = psi;
   }
  }
}
//Lectura y conversion de los angulos
void MakeAnglesDependentOnMPU6050()
{
  Roll =-Phi/5;
  Pitch=Theta*10;
  Azimuth=Psi;
}

//posicionamiento del angulo Roll
void Horizon()
{
  scale(ArtificialHoizonMagnificationFactor);
  noStroke();
  fill(0, 180, 255);
  rect(0, -100, 900, 1000);
  fill(95, 55, 40);
  rotate(-Roll);
  rect(0, 400+Pitch, 900, 800);
  rotate(Roll);
  rotate(-PI-PI/6);
```

```
  SpanAngle=120;
  NumberOfScaleMajorDivisions=12;
  NumberOfScaleMinorDivisions=24;
  CircularScale();
  rotate(PI+PI/6);
  rotate(-PI/6);
  CircularScale();
  rotate(PI/6);
}
//posicionamiento del angulo Yaw
void ShowAzimuth()
{
  fill(50);
  noStroke();
  //rect(20, 470, 440, 50);
  int Azimuth1=round(Azimuth);
  textAlign(CORNER);
  textSize(35);
  fill(255);
  text("GIRO(yaw):  "+Azimuth1+" Deg", 80, 477, 500, 60);
  textSize(40);
  fill(255,255,255);
 text("TABLERO DE REFERENCIA", -450, -480, 500, 60);
}

//posicionamiento del angulo Pitch
void Compass()
{
  translate(2*W/3, 0);
  scale(CompassMagnificationFactor);
```

```
noFill();
stroke(100);
strokeWeight(80);
ellipse(0, 0, 750, 750);
strokeWeight(50);
stroke(50);
fill(0, 0, 40);
ellipse(0, 0, 610, 610);
for (int k=255;k>0;k=k-5)
{
  noStroke();
  fill(0, 0, 255-k);
  ellipse(0, 0, 2*k, 2*k);
}
strokeWeight(20);
NumberOfScaleMajorDivisions=18;
NumberOfScaleMinorDivisions=36;
SpanAngle=180;
CircularScale();
rotate(PI);
SpanAngle=180;
CircularScale();
rotate(-PI);
fill(255);
textSize(60);
textAlign(CENTER);
text("W", -375, 0, 100, 80);
text("E", 370, 0, 100, 80);
text("N", 0, -365, 100, 80);
text("S", 0, 375, 100, 80);
```

```
  textSize(30);
  //text("COMPASS", 0, -130, 500, 80);
  rotate(PI/4);
  textSize(40);
  text("NW", -370, 0, 100, 50);
  text("SE", 365, 0, 100, 50);
  text("NE", 0, -355, 100, 50);
  text("SW", 0, 365, 100, 50);
  rotate(-PI/4);
  CompassPointer();
}
void CompassPointer()
{
  rotate(PI+radians(Azimuth));
  stroke(0);
  strokeWeight(4);
  fill(100, 255, 100);
  triangle(-20, -210, 20, -210, 0, 270);
  triangle(-15, 210, 15, 210, 0, 270);
  ellipse(0, 0, 45, 45);
  fill(0, 0, 50);
  noStroke();
  ellipse(0, 0, 10, 10);
  triangle(-20, -213, 20, -213, 0, -190);
  triangle(-15, -215, 15, -215, 0, -200);
  rotate(-PI-radians(Azimuth));
}
void Plane()
{
  fill(0);
```

```
  strokeWeight(1);
  stroke(0, 255, 0);
  triangle(-20, 0, 20, 0, 0, 25);
  rect(110, 0, 140, 20);
  rect(-110, 0, 140, 20);
}
void CircularScale()
{
  float GaugeWidth=800;
  textSize(GaugeWidth/30);
  float StrokeWidth=1;
  float an;
  float DivxPhasorCloser;
  float DivxPhasorDistal;
  float DivyPhasorCloser;
  float DivyPhasorDistal;
  strokeWeight(2*StrokeWidth);
  stroke(255);
  float DivCloserPhasorLenght=GaugeWidth/2-GaugeWidth/9-StrokeWidth;
  float DivDistalPhasorLenght=GaugeWidth/2-GaugeWidth/7.5-StrokeWidth;
  for (int Division=0;Division<NumberOfScaleMinorDivisions+1;Division++)
  {
    an=SpanAngle/2+Division*SpanAngle/NumberOfScaleMinorDivisions;
    DivxPhasorCloser=DivCloserPhasorLenght*cos(radians(an));
    DivxPhasorDistal=DivDistalPhasorLenght*cos(radians(an));
    DivyPhasorCloser=DivCloserPhasorLenght*sin(radians(an));
    DivyPhasorDistal=DivDistalPhasorLenght*sin(radians(an));
    line(DivxPhasorCloser,          DivyPhasorCloser,          DivxPhasorDistal,
DivyPhasorDistal);
  }
```

```
  DivCloserPhasorLenght=GaugeWidth/2-GaugeWidth/10-StrokeWidth;
  DivDistalPhasorLenght=GaugeWidth/2-GaugeWidth/7.4-StrokeWidth;
  for (int Division=0;Division<NumberOfScaleMajorDivisions+1;Division++)
  {
    an=SpanAngle/2+Division*SpanAngle/NumberOfScaleMajorDivisions;
    DivxPhasorCloser=DivCloserPhasorLenght*cos(radians(an));
    DivxPhasorDistal=DivDistalPhasorLenght*cos(radians(an));
    DivyPhasorCloser=DivCloserPhasorLenght*sin(radians(an));
    DivyPhasorDistal=DivDistalPhasorLenght*sin(radians(an));
    if
(Division==NumberOfScaleMajorDivisions/2|Division==0|Division==NumberOfScal
eMajorDivisions)
    {
      strokeWeight(15);
      stroke(0);
      line(DivxPhasorCloser,          DivyPhasorCloser,          DivxPhasorDistal,
DivyPhasorDistal);
      strokeWeight(8);
      stroke(100, 255, 100);
      line(DivxPhasorCloser,          DivyPhasorCloser,          DivxPhasorDistal,
DivyPhasorDistal);
    }
    else
    {
      strokeWeight(3);
      stroke(255);
      line(DivxPhasorCloser,          DivyPhasorCloser,          DivxPhasorDistal,
DivyPhasorDistal);
    }
  }
```

```
}
void Axis()
{
  stroke(255, 0, 0);
  strokeWeight(3);
  line(-115, 0, 115, 0);
  line(0, 280, 0, -280);
  fill(100, 255, 100);
  stroke(0);
  triangle(0, -285, -10, -255, 10, -255);
  triangle(0, 285, -10, 255, 10, 255);
}
void ShowAngles()
{
  textSize(30);
  fill(50);
  noStroke();
  //rect(-150, 400, 280, 40);
  //rect(150, 400, 280, 40);
  fill(255);
  Pitch=Pitch/5;
  int Pitch1=round(Pitch);
  text("FRONTAL (pitch):  "+Pitch1+" Deg", -140, 411, 500, 60);
  text("LATERAL (roll):  "+Roll*100+" Deg", 280, 411, 500, 60);
}
void Borders()
{
  noFill();
  stroke(0);
  strokeWeight(400);
```

```
  rect(0, 0, 1100, 1100);
  strokeWeight(200);
  ellipse(0, 0, 1000, 1000);
  fill(0);
  noStroke();
  rect(4*W/5, 0, W, 2*H);
  rect(-4*W/5, 0, W, 2*H);
}
void PitchScale()
{
  stroke(255);
  fill(255);
  strokeWeight(3);
  textSize(24);
  textAlign(CENTER);
  for (int i=-4;i<5;i++)
  {
    if ((i==0)==false)
    {
      line(110, 50*i, -110, 50*i);
    }
    text(""+i*10, 140, 50*i, 100, 30);
    text(""+i*10, -140, 50*i, 100, 30);
  }
  textAlign(CORNER);
  strokeWeight(2);
  for (int i=-9;i<10;i++)
  {
    if ((i==0)==false)
    {
```

```
      line(25, 25*i, -25, 25*i);
    }
  }
}
```

Este código fue elaborado por Adrián Fernández, pero se le realizaron algunos cambios para el correcto funcionamiento y adaptarlo al proyecto; el código se puede encontrar en el siguiente link http://mfurkanbahat.blogspot.com.co/2014/11/artificial-horizon-and-compass-using.html .

- **Comandos de control de los motores**

Para el control de los motores del UAV, una vez conectado los SCA y los controladores correctamente, y cada uno en el sentido correcto, se elaboró el siguiente código en arduino.

Primero en el código del control se leen los valores enviados del joystick que es leída en una señal PWM con valores que están entre 0 y 255.

```
void leer_joystick(){

  JLSW_val  = digitalRead(JoystickLeft_SWR);   // read the input pin

  JLX_val = analogRead(JoystickLeft_X);

  JLY_val = analogRead(JoystickLeft_Y);
```

```
  JRSW_val  = digitalRead(JoystickRight_SWR);   // read the input pin

  JRX_val = analogRead(JoystickRight_X);

  JRY_val = analogRead(JoystickRight_Y);

}
```

Posteriormente se configuró cada uno de los motores, asignándoles los valores máximos y mínimos que pueden recibir.

```
void Consgf_ESC(){

  // Funcion para configurar los motores.

  MOTOR1.attach(MOTOR_ESC1);

  MOTOR2.attach(MOTOR_ESC2);

  MOTOR3.attach(MOTOR_ESC3);

  MOTOR4.attach(MOTOR_ESC4);

  MOTOR1.writeMicroseconds(MAX_SIGNAL);

  MOTOR2.writeMicroseconds(MAX_SIGNAL);

  MOTOR3.writeMicroseconds(MAX_SIGNAL);

  MOTOR4.writeMicroseconds(MAX_SIGNAL);

  MOTOR1.writeMicroseconds(MIN_SIGNAL);

  MOTOR2.writeMicroseconds(MIN_SIGNAL);
```

```
  MOTOR3.writeMicroseconds(MIN_SIGNAL);

  MOTOR4.writeMicroseconds(MIN_SIGNAL);

}
```

Después de la previa configuración de los motores, se procedió a leer los datos enviados al UAV desde control y separar cada uno de estos datos en sus respectivas variables.

```
void Actualizar_Datos(){

// Funcion actualiza los valores de los joystick en el receptor acorde con los valores transmitidos.

  JLSW_val  = Datos_rec[0];

  JLX_val = Datos_rec[1];

  JLY_val = Datos_rec[2];

  JRSW_val  = Datos_rec[3];

  JRX_val = Datos_rec[4];

  JRY_val = Datos_rec[5];

  encendido= Datos_rec[6];

  power= Datos_rec[7];

}
```

Una vez que conocemos los datos enviados por el control del UAV, y separados debidamente el joystick que controla la altura y el joystick que controla la dirección procedemos a configurar los comando para que el UAV se desplace a la posición indicada.

```
void Control_Altura(){

/* Función para controlar la potencia de los motores y define los comandos de los joystick derecho.

Del centro hacia arriba aumenta la potencia y de los centros hacia abajo disminuye.

*/

  //hacia abajo

 if(JLY_val >= 530 ){

     if(MESC_Sen <= 1650){

       MESC_Sen = MESC_Sen -50;

     }

 }



 //hacia arriba

 if(JLY_val <= 490 ){
```

```
    if(MESC_Sen >= 1300){

      MESC_Sen = MESC_Sen +50;

    }

  }




  MOTOR1.writeMicroseconds(MESC_Sen);

  MOTOR2.writeMicroseconds(MESC_Sen);

  MOTOR3.writeMicroseconds(MESC_Sen);

  MOTOR4.writeMicroseconds(MESC_Sen);


}

void Control_Direccion(){

/* función controla la dirección acorde con los comandos del joystick izquierdo

 Según la dirección incrementa la potencia en los motores opuestos a la dirección de movimiento.

 */

    // Hacia adelante

    if(JRY_val > 530){

      MOTOR1.writeMicroseconds(MESC_Sen + 50);
```

```
  MOTOR2.writeMicroseconds(MESC_Sen + 50);

  MOTOR3.writeMicroseconds(MESC_Sen);

  MOTOR4.writeMicroseconds(MESC_Sen);

}

// Hacia atras

if(JRY_val < 490){

  MOTOR1.writeMicroseconds(MESC_Sen);

  MOTOR2.writeMicroseconds(MESC_Sen);

  MOTOR3.writeMicroseconds(MESC_Sen + 50);

  MOTOR4.writeMicroseconds(MESC_Sen + 50);

}


// Hacia la Izquierda

if(JRX_val > 530){

  MOTOR1.writeMicroseconds(MESC_Sen);

  MOTOR2.writeMicroseconds(MESC_Sen + 50);

  MOTOR3.writeMicroseconds(MESC_Sen + 50);

  MOTOR4.writeMicroseconds(MESC_Sen);

}

// Hacia la derecha
```

```
    if(JRX_val < 490){

      MOTOR1.writeMicroseconds(MESC_Sen + 50);

      MOTOR2.writeMicroseconds(MESC_Sen);

      MOTOR3.writeMicroseconds(MESC_Sen);

      MOTOR4.writeMicroseconds(MESC_Sen + 50);

    }

}
```

CONCLUSIONES

- El presente trabajo desarrollado, giro entorno a la realización de un cuadricóptero cumpliendo todas sus fases de desarrollo, sirviendo como apoyo y herramienta al proyecto ambiental realizado por la universidad católica. Dentro de los puntos considerados como significativos del proyecto, es el aporte considerable para monitorear y brindar apoyo en campos u operaciones donde hay restricción de acceso y visibilidad.
- Otro aspecto que consideramos importante para llevar este proyecto, consistió en el aprendizaje obtenido, trabajando en cada una de las etapas que permitían evaluar el avance que se obtenía al estructurar e implementar cada elemento que formaran con exactitud el cuadricoptero, para finalmente observar su funcionamiento y utilidad.
- Se ha logrado integrar todos los componentes: microcontrolador, Cámara, sensores, batería, controladores y motores, diseño que en caso de avería de algunos de los componentes sea fácilmente remplazable.
- Como objetivo cumplido, se consiguió realizar todos los objetivos marcados por el proyecto, desde la integración de la estructura mecánica con el hardware y diseño de software de control de cuadricóptero permitiéndole volar prácticamente estable.

LISTA DE FIGURAS

Figura 1: ***Clasificación de los UAV (1)*** 15
Figura 2: ***Tipos de UAV Según su despegue*** (1) 17
Figura 3: ***Ejemplo de un cuadricoptero*** 17
Figura 4*: Rotores Cuadricoptero* (2) 21
Figura 5*: PWM* 22
Figura 6*: Motor Brushless* 24
Figura 7: ***Turnigy 2200mAh 3S 30C Lipo Pack*** 26
Figura 8: ***Imagen del ESC*** 28
Figura 9 ***Vista inferior y superior del cuerpo del UAV*** 34
Figura 10: ***Imagen del motor*** 35
Figura 11: ***Rendimiento del Motor LiPo 3s*** 36
Figura 12: ***Rendimiento del motor LiPo 4s*** 36
Figura 13: ***Dimensiones Del Motor*** 38
Figura 14: ***Imagen del ESC*** 40
Figura 15: ***Imagen del BMP085*** 41
Figura 16: ***Transferencia de Datos I2C*** 42
Figura 17: ***Presión Barométrica*** 43
Figura 18*: Etiqueta de los pines* 44
Figura 19: ***Conexión Sensor - Arduino*** 44
Figura 20: ***Imagen del MPU-6050*** 46
Figura 21: ***Conexión Arduino – MPU (6050)*** 47
Figura 22: ***GY-271*** *(6)* 48
Figura 23: ***Imagen GPS ublox neo-6m*** 50
Figura 24: ***Vista Frontal y Posterior del Arduino Mega 2560*** 51
Figura 25: ***Modulo transceptor NRF24L01*** 56
Figura 26: ***Batería Turnigy 2200mAh 3S 30C Lipo Pack*** 58
Figura 27*: Imagen Gimbal Alexmos* 59
Figura 28: ***Imagen del joystick*** 60

Bibliografía

1. **Novillo, Pablo Guamán.** Monografias.com. [En línea] SD de SD. http://www.monografias.com/trabajos-pdf5/vehiculos-aereos-no-tripulados/vehiculos-aereos-no-tripulados.shtml.

2. *Modelo y control de un cuadricoptero.* **Fernandez, Luis Sevilla y Galán, Jose Porras.** SD : ICAI – Universidad Pontificia Comillas., SD.

3. **Ingenieros., Master.** *MOTOR BRUSHLESS (SIN ESCOBILLAS).* SD : s.n., SD.

4. **NSK ELECTRONICS. [En línea] Noviembre de 2015. http://www.nskelectronics.in/index.php?route=product/product&product_id=578.**

5. Magneticos, Sensores. 5 Hertz Electronica. [En línea] http://5hertz.com/.

6. Recommendations, Best Electronic Gadgets. randyelectronic. [En línea] 2013. http://randyelectronic.blogspot.com.co/2013/08/gy-271-hmc5883l-module-electronic.html.

7. Torres, Hector. Hetpro, Herramientas Tecnologicas Profesionales. [En línea] Noviembre de 2015. http://hetpro-store.com/TUTORIALES/gps-ublox-neo-6m-modulo-con-matlab/.

8. Portal de descargas y centro neuralgico del mundo proccesing. [En línea] SD, Enero de 2016. http://users.dsic.upv.es/~jlinares/grafics/processing_spa_1.pdf .

9. Verdejo, Daniel Garijo, Peréz, Jesus Ismael López y Estrada, Isaac Pérez. *control de un vehiculo aéreo no tripulado.* Madrid : universidad Complutense de Madrid, SD.

10. *vehiculos aereos no tripulados para uso civil, Tecnología y aplicaciones.* Cerro., A Barrientos: J del. Madrid : Universidad politecnica de Madrid., S.D.

11. Novillo, Pablo Guamán. Monografias. [En línea] 2015. http://www.monografias.com/trabajos-pdf5/vehiculos-aereos-no-tripulados/vehiculos-aereos-no-tripulados.shtml.