

Diseño y Construcción de un Quadcopter

Memoria

Nombre:

Sergio Moyano Díaz

Índice

Glosario.....	6
Índice de figuras.....	7
Índice de Tablas.....	9
Resumen.....	10
Resum.....	10
Abstract.....	11
1. Introducción.....	12
1.1. Contexto	12
1.2. Personas implicadas en el proyecto	12
2. Estado del arte	14
2.1. Introducción	14
2.2. Análisis de productos del mercado	14
2.2.1. Parrot Ar Drone	14
2.2.2. PCB QuadCopter	15
2.3. Conclusión	16
3. Alcance del proyecto.....	17
3.1. Objetivo del proyecto.....	17
3.2. Metodología	17
3.3. Riesgos.....	17
4. Planificación del proyecto.....	18
4.1. Planificación general.....	18
4.1.1. Duración estimada.....	18
4.1.2. Consideraciones.....	18
4.2. Planificación del proyecto.....	18
4.2.1. Planificación y viabilidad.....	18
4.2.2. Análisis y Diseño del proyecto.....	18
4.2.3. Etapas Intermedias.....	19

4.2.4. Etapa final.....	20
4.3. Tiempo estimado por etapa.....	20
4.4. Diagrama de Gantt.....	21
4.5. Plan de acción.....	22
4.6. Recursos.....	22
5. Diseño Estructural y Mecánico.....	23
5.1 Funcionamiento y Movimiento de Quadcopte.....	23
5.2 Selección de Materiales.....	26
5.3 Bases del diseño.....	29
5.4 Esbozo de la Estructura.....	30
5.5 Estructura Final.....	31
5.5.1 Chasis.....	31
5.5.2 Cruceta.....	32
5.5.3 Soporte de los motores.....	33
5.5.4 Carcasa.....	33
5.5.5 Plataforma para la placa electrónica.....	37
5.5.6 Hélices.....	37
5.5.7 Peso Total.....	38
6. Diseño del Hardware.....	40
6.1. Bases de Diseño.....	40
6.2. Motores.....	40
6.2.1. Selección de los motores.....	40
6.2.2. Motores seleccionados: EMAX CF2822.....	41
6.2.3. Caracterización de los motores.....	42
6.3. ESC (Electronic Speed Controller).....	43
6.4. Batería.....	45
6.4.1. Selección de la Batería.....	45
6.4.2. Calculo del Tiempo de Vuelo.....	46
6.5. Sensores	47
6.6. Bluetooth	48

6.7. Cámara.....	49
6.8. Microcontrolador.....	49
6.9. Placa electrónica.....	51
6.10. Configuración Final.....	53
7. Diseño del Software.....	55
7.1. Entorno de Programación.....	55
7.2. Software de Arduino.....	56
7.2.1. Módulo de Motores.....	56
7.2.1.1. Calibrado y uso de motores.....	56
7.2.1.2. Funciones.....	57
7.2.2. Módulo de Bluetooth.....	57
7.2.2.1. Protocolo de comunicación.....	58
7.2.2.2. Funciones.....	59
7.2.3. Módulo de Control.....	59
7.2.3.1. Control del movimiento del Quadcopter.....	59
7.2.3.2. Diseño del PID.....	61
7.2.3.3. Controladores de Pitch, Roll y Yaw.....	62
7.2.3.4. Funciones.....	63
7.2.4. Módulo de Sensores.....	63
7.2.4.1. Firmware de Razor 9DOF.....	63
7.2.4.2. Funciones.....	64
7.2.5. Módulo de Cámara.....	64
7.2.5.1. Protocolo de comunicación.....	64
7.2.5.2. Funciones.....	68
7.2.6. Módulo de Leds.....	68
7.2.7. Módulo Principal.....	68
7.3. Aplicación para PC	68
7.3.1 Diagrama de clases.....	69

8. Conclusiones.....	71
8.1. Resumen de Características.....	71
8.2. Conclusión Final.....	72
8.3. Futuro Trabajo.....	73
9. Presupuesto.....	74
9.1. Consideraciones del presupuesto.....	74
9.2. Presupuesto de recursos humanos.....	74
9.3. Presupuesto de hardware.....	74
9.4. Presupuesto de software.....	75
9.5. Presupuesto Total.....	75
10. Contexto del proyecto.....	76
10.1. Impacto social y medioambiental.....	76
10.2. Control y sostenibilidad.....	76
10.3. Viabilidad económica.....	76
11. Referencias.....	77
12. Bibliografía.....	78

Glosario

Android

Es un sistema operativo basado en el Kernel de Linux y se usa en dispositivos móviles, como tablets, teléfonos, etc.

Arduino

Es una plataforma de electrónica abierta para la creación de prototipos basada en software y hardware flexibles y fáciles de usar.

Bluetooth

Es un protocolo de comunicación que posibilita la transmisión de voz y datos entre diferentes dispositivos mediante un enlace por radiofrecuencia en la bande de los 2,4 GHz.

ESC (Electronic Speed Controller)

Es un circuito que se encarga de generar una señal trifásica que alimenta motores Brushless. La velocidad de giro se varía mediante una señal suministrada por PWM.

IOS

Es el sistema operativo diseñado por Apple para dispositivos móviles.

PWM (Pulse Width modulation)

Es una técnica en la que se modifica el ciclo de trabajo de una señal periódica, ya sea para transmitir información a través de un canal de comunicaciones o para controlar la cantidad de energía que se envía a una carga.

SPI (Serial Peripheral Interface)

Es un protocolo de transmisión síncrona que envía la información en serie. Está pensado para funcionar en modo maestro-esclavo y es usado básicamente por sensores y Tarjetas SD.

UART (Universal Asynchronous Receiver Transmitter)

Es el dispositivo encargado de la comunicación asincrónica donde los datos se envían en serie, tiene diversas especificaciones dependiendo del voltaje que se use para comunicarse.

UAV (Unmanned Aerial Vehicle).

Es un sistema autónomo que puede operar sin intervención humana alguna durante su funcionamiento.

WI-FI

Es un protocolo de comunicación que permite intercambiar datos o conectarse a internet usando la banda 2,4 GHz UHF o la banda de 5 GHz.

Índice de Figuras

FIGURA 1 – AR Drone.....	14
FIGURA 2 – PCB Quadcopter.....	15
FIGURA 3 – Disposición del Quadcopter.....	23
FIGURA 4 – Rotaciones del Quadcopter.....	23
FIGURA 5 – Control de Altitud.....	24
FIGURA 6 – Control del Yaw.....	24
FIGURA 7 – Control del Roll.....	25
FIGURA 8 – Control del Pitch.....	25
FIGURA 9 – Barras de Aluminio.....	26
FIGURA 10 – Fibra de Carbono.....	27
FIGURA 11 – Plancha de Poliestireno Extrusionado.....	27
FIGURA 12 – Pieza imprimida en PLA.....	28
FIGURA 13 – Fibra de vidrio.....	28
FIGURA 14 – Esbozo.....	30
FIGURA 15 – Prototipos de plástico.....	31
FIGURA 16 – Chasis.....	32
FIGURA 17 – Barra Soporte de motor.....	32
FIGURA 18 – Soporte de los motores.....	33
FIGURA 19 – Piezas de la Carcasa.....	33
FIGURA 20 – Carcasa de Plastico.....	34
FIGURA 21 – Carcasa de fibra de vidrio.....	35
FIGURA 22 – Plancha superior de la Carcasa de Poliestireno.....	35
FIGURA 23 – Plancha inferior de la Carcasa de Poliestireno.....	36
FIGURA 24 – Carcasa de Poliestireno Extrusionado.....	36
FIGURA 25 – Plataforma de la placa electrónica.....	37
FIGURA 26 – Plataforma de los sensores.....	37
FIGURA 27 – Hélice normal, inversa y portahélices de 3 mm.....	38
FIGURA 28 – Motor Emax CF2822.....	41
FIGURA 29 – Dimensión del motor.....	41
FIGURA 30 – Banco de caracterización de los motores.....	42
FIGURA 31 – Fuente de PC con adaptador T-Dean.....	43
FIGURA 32 – ESC EMAX 25 A.....	44
FIGURA 33 – Batería Lipo 2200mAh.....	45
FIGURA 34 – Cargador de Baterías Lipo.....	46
FIGURA 35 – Sensor Altimu-10.....	47
FIGURA 36 – Sensor Razor 9 DOF.....	48

FIGURA 37 – Bluetooth Stick.....	48
FIGURA 38 – Cámara Jpeg TTL.....	49
FIGURA 39 – Arduino Due.....	50
FIGURA 40 – Raspberry Pi Modelo B.....	51
FIGURA 41 – Protoboard Arduino.....	51
FIGURA 42 – Esquema Electrónico Protoboard.....	52
FIGURA 43 – Placa de los periféricos.....	53
FIGURA 44 – Esquema Electrónico Placa Periféricos.....	53
FIGURA 45 – Diagrama de Bloques del Sistema.....	53
FIGURA 46 – Montaje Final del Hardware.....	54
FIGURA 47 – IDE de Arduino.....	55
FIGURA 48 –Diagrama de Clases Arduino.....	58
FIGURA 49 – Disposición del Quadcopter.....	59
FIGURA 50 – Controlador PID de un eje.....	62
FIGURA 51 – Conexión con la cámara.....	66
FIGURA 52 – Configuración de cámara.....	66
FIGURA 53 – Obtención de la imagen.....	67
FIGURA 54 – Aplicación de control.....	68
FIGURA 55 – Diagrama de Clases aplicación PC.....	70
FIGURA 56 – Montaje Final Parte Superior.....	71
FIGURA 57 – Montaje Final Parte Inferior.....	72

Índice de Tablas

Tabla 1 – Tiempo por Etapa.....	20
Tabla 2 – Diagrama de Gantt.....	21
Tabla 3 – Distribución del Peso del Prototipo.....	39
Tabla 3 – Tabla de características de los motores.....	41
Tabla 4 – Caracterización de los motores.....	43
Tabla 5 – Características de la Batería Lipo.....	45
Tabla 6 – Estructura del Paquete de comandos del Bluetooth.....	58
Tabla 7 – Estructura del Paquete de Datos del Bluetooth.....	58
Tabla 8 – Matriz de comandos.....	60
Tabla 9 – Información enviada por el Razor 9DOF	64
Tabla 10 – Estructura del Paquete de comandos del cámara.....	64
Tabla 11 – Estructura de los paquete de datos de la cámara.....	64
Tabla 12 – Lista de comandos de la cámara.....	65
Tabla 13 – Resumen de Características.....	72
Tabla 14 – Presupuesto de Recursos Humanos.....	74
Tabla 15 – Presupuesto de Hardware.....	74
Tabla 16 – Presupuesto de Software.....	75
Tabla 17 – Presupuesto Total.....	75

RESUMEN

Este proyecto trata el diseño y construcción de un Quadcopter, es decir, un helicóptero de 4 hélices.

Esta memoria está dividida en tres grandes bloques:

El primer bloque se centra en la parte mecánica, donde se detalla el proceso de diseño, los componentes mecánicos y su funcionalidad. También se trata la selección del material teniendo en cuenta la resistencia mecánica, peso y precio de los materiales.

El segundo bloque se centra en la parte electrónica, donde se detalla todos los componentes que se han usado y la razón de su uso. Con esto nos referimos a las baterías, ESC, Arduino, la placa de electrónica, los motores y los sensores, la cámara y el Bluetooth.

El tercer bloque se detalla todo el software realizado. Es decir, cada una de librerías implementadas en Arduino para usar cada componente, y la librería que contiene el algoritmo de control. Además se trata la aplicación que controlará el Quadcopter.

RESUM

Aquest projecte tracta el disseny i construcció d'un Quadcopter, és a dir, un helicòpter de 4 hèlixs.

Aquesta memòria està dividida en tres grans blocs:

El primer bloc se centra en la part mecànica, on es detalla el procés de disseny, els components mecànics i la seva funcionalitat. També es tracta la selecció del material tenint en compte la resistència mecànica, pes i preu dels materials.

El segon bloc se centra en la part electrònica, on es detalla tots els components que s'han usat i la raó del seu ús. Amb això ens referim a les bateries, ESC, Arduino, la placa electrònica, els motors i els sensors, la càmera i el Bluetooth.

El tercer bloc es detalla tot el programari realitzat. És a dir, cadascuna de llibreries implementades en Arduino per utilitzar cada component, i la llibreria que conté l'algorisme de control. A més es tracta l'aplicació que controlarà el Quadcopter.

ABSTRACT

The objective of this project is the design and construction of a Quadcopter, an helicopter with 4 propellers.

This report is divided into three sections:

The first block is centered on the mechanical part, where the process of designing, the mechanical components and its functionality is detailed. A Material selection is detailed considering the strength, weight and price of materials.

The second section focuses on the electronic part. It explain all the components that we have been used and the reason for its use is detailed. The components are: the batteries, ESC, Arduino board electronics, motors and sensors, camera and Bluetooth.

The third block explains the software: each implemented in Arduino library for using each component and the control library which contains the control algorithm. Also, it explains the application that will handle the Quadcopter.

1. Introducción

1.1 Contexto

En estos últimos años, el desarrollo de vehículos no tripulados (del inglés UAV- Unmanned Aerial Vehicle [1]) ha tenido un avance muy significativo, sobre todo para el uso en aplicaciones civiles. Las aplicaciones usuales de este tipo de vehículo se centran en tareas donde tienen entornos de acceso difícil o que representa algún peligro, como por ejemplo: tareas de vigilancia y seguimiento, reconocimiento geográfico, etc.

Dentro de los UAV, se está centrando mucha investigación académica son los conocidos como Mini-UAV, que son sistemas no tripulados de pequeño tamaño, que es el que nos ocupa en este proyecto. El desarrollo de estos sistemas ha sido posible gracias al desarrollo del microcontroladores, que ofrecen la posibilidad de hacer cálculos muy complejos en poco espacio, a la miniaturización de los sensores (MEMS – Micro Electromechanical Systems [2]) así como las mejoras en los sistemas de almacenamiento de energía.

Este proyecto trata de fabricar un Quadcopter pilotado desde un dispositivo mediante Bluetooth. El Quadcopter dispone de una cámara que puede hacer fotografías y enviarlas a un dispositivo. La idea principal de este proyecto es conseguir desarrollar una plataforma desde cero en la cuales en futuros trabajos pueda seguir ampliándose.

La idea surgió en una Asignatura de la especialidad del grado, donde se hizo un estudio de viabilidad y una visión global de las tecnologías disponibles para hacerlo. La motivación principal es poder realizar un proyecto multidisciplinar donde intervienen el diseño de hardware y software. Desde un principio ya se tenía claro que iba a resultar un proyecto muy ambicioso y de gran dificultad técnica.

1.2 Personas implicadas en el proyecto

En esta sección se explicará en detalle qué personas están implicadas en este proyecto:

- **Desarrollador:**

El desarrollador de este proyecto es exclusivamente el autor de este trabajo final de grado. El desarrollador es el encargado de llevar a cabo el proyecto entero.

- **Director del proyecto**

El Director del proyecto es Pere Marés Martí, su papel es supervisar que el proyecto cumple con el calendario establecido y cumple con los objetivos marcados.

- **Usuarios que estén interesado en desarrollar un Quadcopter.**

Los usuarios que están interesados en desarrollar un Quadcopter estarán muy interesados en este proyecto.

- **Usuarios final**

Los usuarios finales son la gente que acabará usando el Quadcopter construido para sus intereses propios, como tareas de seguimiento y vigilancia, tareas de reconocimiento geográfico, etc.

2. Estado del Arte

2.1 Introducción

En esta sección se va a abordar la fase previa al desarrollo, para poder determinar si ya existen productos comerciales que resuelvan este proyecto o si hace falta diseñar uno de nuevo.

Existen una cantidad reseñable de proyectos que afrontan el reto de construir un Quadcopter. Dependiendo del enfoque desde donde los miremos, se pueden dividir de varias maneras:

- **Open-source vs Propietario:**

Podemos separar entre proyecto de código de abierto y código propietario. Existen varias comunidades que desarrollan software de código abierto orientadas a determinadas plataformas. Otras en cambio, no publican su código. Pero no solo eso. Al tratarse de hardware también, los esquemas de algunas partes pueden hacerse públicas o no. Un claro ejemplo de hardware open-source es Arduino [3].

2.2 Análisis de los productos del mercado

2.2.1 Parrot Ar Drone

El Parrot Ar Drone[4] es un proyecto reciente de la empresa Francesa Parrot[5]. En el año 2010 presentaron su Quadcopter AR Drone en una conferencia de Las Vegas:

Un Quadcopter con una carcasa de plástico y espuma de aproximadamente 30 centímetros de largo.



Figura 1. AR Drone

Dispone dos microcámaras incorporados que permiten grabar video. Se controla mediante una aplicación para Smartphone que está disponible para dispositivos iOS y Android. La comunicación es vía Wi-Fi.

El software corre sobre el sistema operativo Linux. Es un producto completamente comercial y han decidido mantener el código como propietario. Sin embargo, dispone de una API con la que se permite programar hasta cierto punto el comportamiento del drone. Tiene la

habilidad de reconocer objetos 3D y es compatible con juegos que introducen realidad aumentada.

En cuanto a las especificaciones técnicas, dispone de un microcontrolador ARM9 468 MHz embebido con 128 Megabytes de RAM. Dispone de comunicación mediante Wi-Fi y USB. Un acelerómetro de 3 ejes, dos giroscopios y un altímetro ultrasónico. La estructura está fabricada en aluminio. Posee 4 motores Brushless de 15 W y una batería recargable de 1 Ah de Litio que proporciona 11.1 voltios nominales. Con un peso de entre 380/420 gramos es capaz de volar durante 12 minutos con una velocidad de 5 metros/segundo o lo que es lo mismo, 18 km/h.

Su precio actual es de 280€.

2.2.2 PCB Quadcopter

El PCB Quadcopter [6] es proyecto open-source tanto software como hardware desarrollado por un par de aficionados a la electrónica. El objetivo de este proyecto crear un Quadcopter donde la propia estructura fuera la placa electrónica. Para ello no podía ser muy grande, ya que la estructura no aguantaría las tensiones.



Figura 2: PCB Quadcopter

En cuanto a las características técnicas, mide de motor a motor 16,5 cm, usa motores Brushless con hélices 4 pulgadas, una batería de 11.1 V de 370 mAh, está controlado por un mando radiocontrol, pesa 138 g y puede volar 8 min autónomamente.

2.3 Conclusión

Actualmente el uso masivo de teléfonos permite reducir costes en el ámbito de los dispositivos controlados por control remoto, haciendo innecesario el uso de otros sistemas de radiocontrol.

En modelos de Quadcopter controlados mediante móvil, hay que destacar que el Parrot AR Drone que ha implementado esta idea recientemente. Sin embargo, no se ofrecen esquemas del hardware y el código es propietario, por lo que se habrá de analizar y ver cuál es la mejor manera de llevar a cabo el proyecto.

Así pues, se ve que en el mercado no hay productos open-source que tenga las mismas características que nuestro proyecto, así que se desarrollará desde 0 basándonos en plataformas abiertas y con un coste lo más reducido posible.

3. Alcance del proyecto

3.1 Objetivos del proyecto

El alcance de proyecto incluye todo el desarrollo necesario para tener el Quadcopter volando y controlado por Bluetooth, que es:

- Diseño y Construcción de la Estructura.
- Diseño de los algoritmos de Control.
- Diseño y Construcción de la placa electrónica.
- Integración del Bluetooth y la cámara.
- Integración y configuración de los sensores.
- Integración de los motores.
- Selección del Microcontrolador.

El desarrollo de sistemas de control para este tipo de vehículos no es trivial, debido principalmente a la dinámica tan compleja inherente a los sistemas aerodinámicos. Estos sistemas son multivariables y presentan características no lineales. Ya que el objetivo del proyecto no es el diseño del algoritmo de Control, se implementará un algoritmo sencillo basado en PID. En un futuro se podrán seguir experimentando diferentes funciones de control y navegabilidad.

3.2 Metodología

Debido al poco tiempo disponible para desarrollar este proyecto, lo mejor es usar una metodología de desarrollo ágil. Este método nos ofrece flexibilidad, rápido desarrollo y nos permite obtener resultados en menos tiempo en comparación con otros métodos, cosa que nos ayudará a evitar los riesgos que conlleva el proyecto.

Primero se hará un intensivo inicial de unas 4 semanas donde se hará la planificación y el presupuesto de proyecto.

Posteriormente se harán intensivos de 1 a 4 semanas donde se harán cada uno de las tareas del proyecto, primero se analizará el volumen de trabajo de la tarea, después se implementará y finalmente se probará que todo funcione correctamente.

3.3 Riesgos

En propia realización del proyecto se prevén ciertos riesgos que pueden afectar al correcto desarrollo del proyecto.

- Mala planificación: Debido a que es difícil aproximar la duración de algunas tareas del proyecto, es posible que el proyecto finalice más tarde o más temprano de lo esperado.
- Uso incorrecto del hardware: Debido a un mal uso no intencionado puede provocar que el hardware deje de funcionar como la propia batería o el propio microcontrolador y se deban de comprar otra vez.

4. Planificación del proyecto

4.1 Planificación general

4.1.1 Duración Estimada

La duración del proyecto es aproximadamente 3 meses. La fecha de inicio del proyecto es el 16 de febrero de 2014 y la fecha de finalización es el 5 de Junio de 2014.

4.1.2 Consideraciones

Es importante tener en cuenta que las fechas calculadas en la planificación inicial pueden ser modificadas y actualizadas durante la realización del proyecto, debido a que pueden surgir complicaciones inesperadas.

Además, durante la semana santa (14/04/2014 al 21/04/2014) el proyecto será parado por vacaciones.

4.2 Planificación del proyecto

4.2.1 Planificación y viabilidad

En esta etapa pertenece al curso de gestión de proyectos incluido en el trabajo final de grado y está compuesto por 5 fases:

- Alcance del proyecto
- Planificación del proyecto
- Presupuesto del proyecto
- Estado del arte
- Pliego de condiciones

4.2.2 Análisis y diseño del proyecto

El objetivo de esta fase es hacer un análisis en profundidad del proyecto, determinar que tecnología se usará y obtener diseño detallado del Quadcopter.

Para empezar se ampliará el estudio del arte, para tener una idea de que tecnologías son las más convenientes para desarrollar el proyecto, y así poder conocer en profundidad las necesidades del sistema a construir, en este caso un Quadcopter.

Posteriormente se hará el diseño, se determinará qué características son necesarias en el microcontrolador a usar, qué motores son los más apropiados para el proyecto, cual es el espacio y el peso que ocupará el Quadcopter.

4.2.3 Etapas intermedias

Las etapas intermedias que conforman el proyecto son las siguientes:

- Diseño y Construcción de la Estructura:

En esta etapa se realiza el propio diseño estructural del Quadcopter y se obtienen los modelos 3D para su construcción.

Se subdivide en: Diseño, Construcción y Validación.

- Diseño y Construcción de la placa electrónica:

Se realiza el esquemático electrónico de la placa que contiene el microcontrolador y se construye la placa y se testea que funcione bien.

Se subdivide en: Diseño, Construcción y Test.

- Integración de los motores:

Se montan los motores sobre la estructura, se elabora una librería para el microcontrolador para poder usarlos y se prueban.

Se subdivide en: Montaje, Diseño y Test.

- Integración y configuración de los sensores:

Se montan los sensores sobre la placa, se elabora una librería para el microcontrolador para poder usarlos y se prueban.

Se subdivide en: Montaje, Diseño y Test.

- Integración del Bluetooth y la cámara:

Se montan el Bluetooth y la cámara sobre la placa, se elabora una librería para el microcontrolador para poder usarlos y se prueban.

Se subdivide en: Montaje, Diseño y Test.

- Diseño de la aplicación para móvil/ordenador.

Se crea una aplicación para móvil u ordenador para poder controlar el Quadcopter.

Se subdivide en: Diseño y Test

- Diseño de los algoritmos de Control:

Se determina el tipo de algoritmo que usará para controlar el Quadcopter, se implementa y se prueba su funcionamiento.

Se subdivide en: Análisis, Diseño y Test.

4.2.4 Etapa final

En esta etapa se ultiman los detalles de proyecto para cerrarlo definitivamente. También se elabora la memoria del proyecto, la presentación final y una pequeña demo del Quadcopter funcionando.

4.3 Tiempo estimado por etapa

Etapas	Dedicación estimada (horas)
Planificación y viabilidad	80
Análisis y diseño del proyecto	70
Diseño y construcción de la estructura	100
Diseño y construcción de la placa electrónica	100
Integración de los motores	30
Integración y configuración de los sensores	30
Integración del Bluetooth y la cámara	40
Diseño de la aplicación móvil/ordenador	40
Diseño del algoritmo de control	60
Etapa Final	50
Total	600 horas

Tabla 1: Tiempo por Etapa

4.4 Diagrama de Gantt

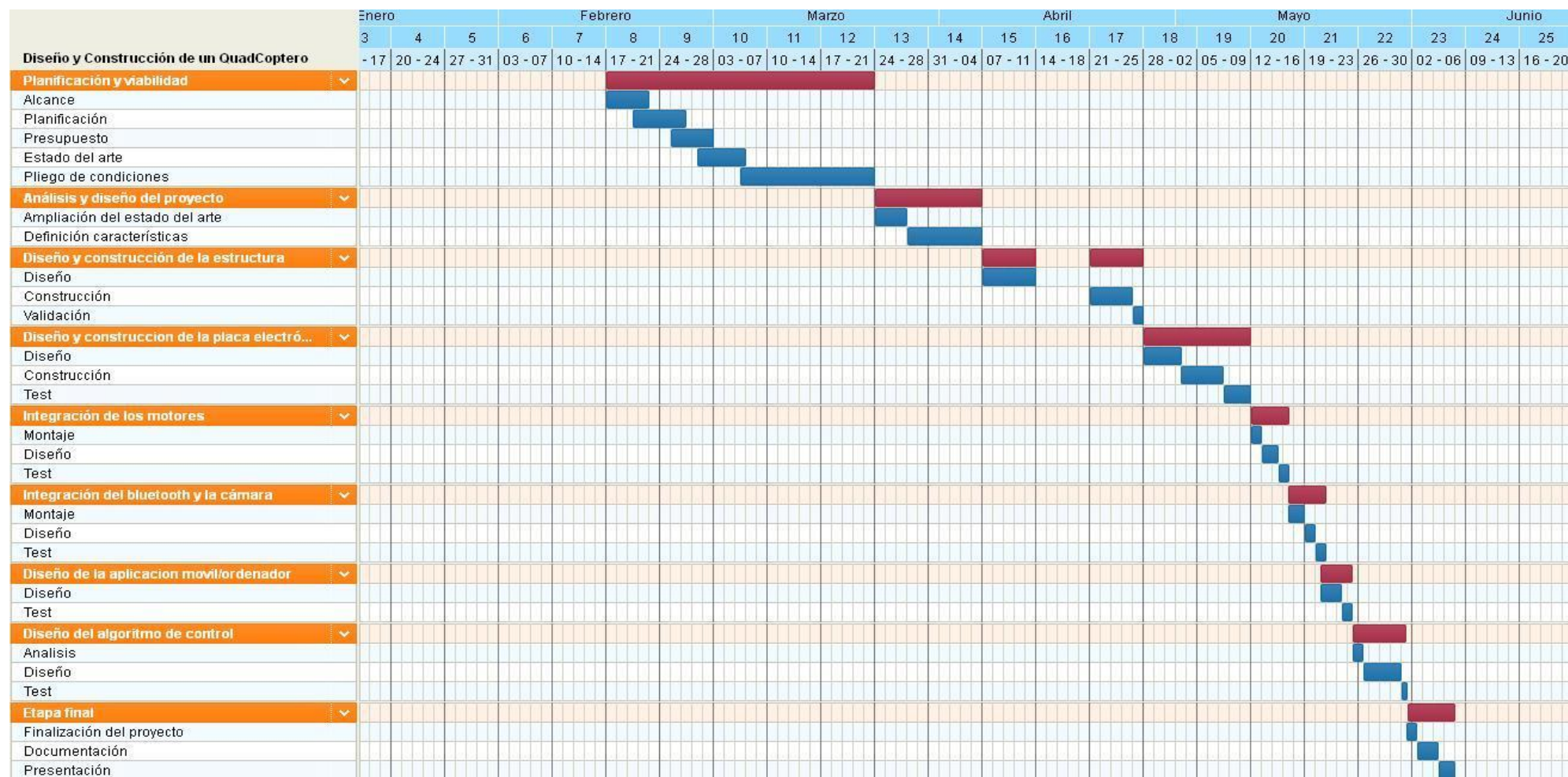


Tabla 2: Diagrama de Gantt

4.5 Plan de acción

Como se ha dicho anteriormente, se usará una metodología de desarrollo ágil. Consecuentemente si las etapas mencionadas en el punto anterior tienen una duración diferente de lo esperado, se modificarán consecuentemente. Por ejemplo, si la etapa tiene una duración inferior de lo esperado, se iniciará de inmediato la siguiente. Aunque, si la tarea dura más de lo esperado, que retrasará las siguientes tareas.

Al final de cada etapa habrá una reunión con el director del proyecto, se llevará a cabo con el fin de supervisar el proyecto y confirmar que está desarrollándose correctamente.

Así en conclusión, será aproximadamente diez reuniones y la estimación de las horas dedicadas por semana es de 40 horas / semana. Por lo tanto, la planificación del proyecto es alcanzable.

4.6 Recursos

Para desarrollar el proyecto, se usarán los siguientes recursos:

Hardware:

- Notebook Toshiba NB500
- HTC Wildfire S
- Impresora 3D Prusa Mendel
- Fuente de alimentación de Pc



Software:

- Microsoft Windows XP
- Microsoft Office 2010
- Microsoft Project
- Eclipse
- Adobe Reader
- IDE Arduino
- Cacao
- SolidWorks



5. Diseño Estructural

5.1 Funcionamiento y Movimiento de Quadcopter

La característica principal del Quadcopter son la disposición simétrica y su rotación asimétrica dos a dos. Esto le permite equilibrarse aplicando rotaciones iguales en los 4 motores, y al mismo tiempo hace calcular los giros.

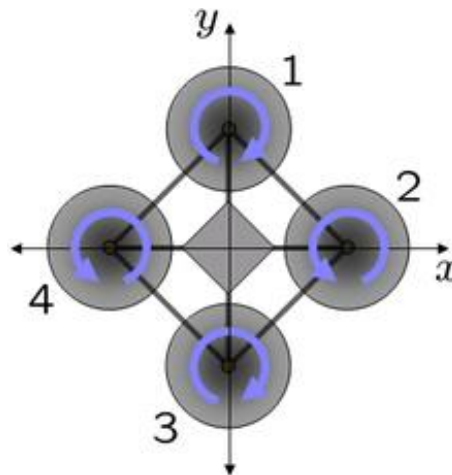


Figura 3: Disposición del Quadcopter

Los pares generados por 1,3 y 2,4 de figura 3 son contrarios y eso auto-estabiliza el Quadcopter siempre y cuando se apliquen empujes de mismo módulo en los 4 motores. De esta manera no se necesita la hélice lateral de un helicóptero, simplificando el cálculo y el diseño.

En aeronáutica, las rotaciones que puede ejecutar un aparato según sus ejes se llaman Yaw, Roll y Pitch. En la siguiente figura se puede ver a qué ejes se refiere cada término.

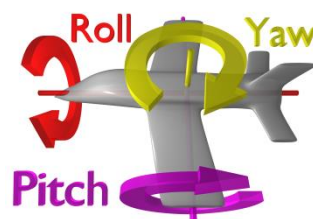


Figura 4: Rotaciones del Quadcopter

Así pues las cuatro configuraciones de motores y sus movimientos asociados son:

- Control del Altitud

La posición base es con los 4 motores aplicando el mismo empuje. Con la cantidad total de empuje podemos llegar a 3 movimientos:

- Estabilizarse en el aire
- Subir altura
- Bajar altura



Figura 5: Control de Altitud

- Control del Yaw

Para ajustar el yaw se sube el empuje de dos motores opuestos mientras los otros dos se mantienen estables.



Figura 6: Control del YAW

- Control del Roll

El movimiento de roll se consigue modificando dos motores opuestos con la misma diferencia, pero en uno se sube y en otro se baja.



Figura 7: Control del Roll

- Control del Pitch

El movimiento de pitch se hace de la misma manera que el roll, pero tocando los otros dos motores.



Figura 8: Control del Pitch

5.2 Selección de Materiales

Antes de empezar el diseño, hay que tener una idea de los materiales que vamos a poder usar y así poder diseñar siendo conscientes de las características de los materiales que disponemos. Así pues se ha hecho una búsqueda intensiva de materiales para poder construir el Quadcopter, centrándose en los que nos proporcionará un alto ratio resistencia/peso. Los principales materiales que se han encontrado con este fin son:

- **Aluminio:**

Es un metal no ferromagnético, y tiene unas propiedades mecánicas que lo hacen muy útil en la ingeniería como una baja densidad ($2,7 \text{ g/cm}^3$) y una alta resistencia mecánica, según que aleación disponga el aluminio se puede llegar hasta los 690 MPa. Además es un material barato y se mecaniza con facilidad. En el mercado se pueden encontrar todo tipos de formas básicas como planchas, barras, varillas, etc.

Por otro lado, es resistente a la corrosión, es buen conductor de la electricidad y el calor, aunque en este proyecto no se aprovechen esas propiedades.



Figura 9: Barras de Aluminio

- **Fibra de carbono**

Es una fibra constituida por finos filamentos de 5- 10 μm de diámetro y compuesto principalmente por carbono. Cada filamento de carbono es la unión de miles de fibras de carbono. Tiene propiedades similares al acero, su resistencia longitudinal esta sobre los 1100 MPa, pero la transversal esta sobre los 50 MPa, y además tiene una baja densidad ($1,6 \text{ g cm}^3$). Pero su principal inconveniente es su elevado coste.

Por otro lado, es una gran aislante térmico, conductor de la electricidad, resistente a los cambios de temperatura, y a agentes externos.

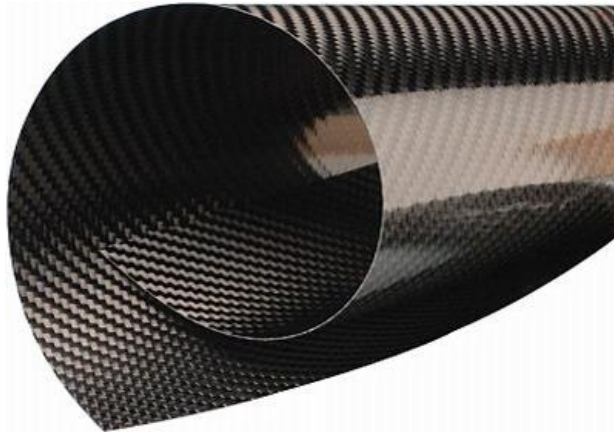


Figura 10: Fibra de carbono

- Poliestireno Extrusionado

Es una espuma rígida resultante de la extrusión del Poliestireno en presencia de un gas espumante. Su principal propiedad es su bajísima densidad de ($0,033 \text{ g/cm}^3$) pero una baja resistencia (250 KPa), también es un gran aislante térmico. Su principal uso, aparte del aislante, es como protector para choques utilizando un grosor elevado. Es capaz de absorber golpes otorgando muy poco peso.



Figura 11: Plancha de Poliestireno Extrusionado

- PLA

Es un plástico biodegradable derivado de recursos renovables como el almidón de maíz o la caña de azúcar. Sus principales propiedades son una densidad baja (1.25g/cm^3) y su resistencia mecánica de 65 MPa. Sus principales uso es como filamento para impresoras 3D. Permite crear piezas altamente complejas usando un software de CAD como Solidworks. Aparte es buen aislante de calor y la electricidad.

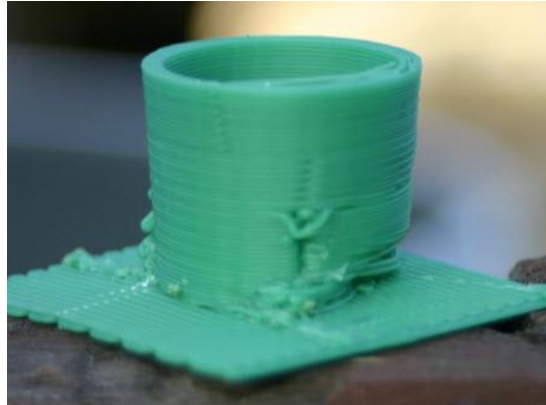


Figura 12: Pieza imprimida en PLA

- Fibra de vidrio

Es una material que consta de fibras numerosas y extremadamente finas de vidrio. Sus propiedades son una densidad alta ($2,58\text{ g/cm}^3$) y en cuanto a resistencia tiene propiedades similares a la fibra de carbono, siendo su resistencia longitudinal 1080 MPa, pero la transversal esta sobre los 50 MPa, pero su ventaja está en el precio que es muy bajo.



Figura 13: Fibra de vidrio

5.3 Bases del diseño

Hoy en día, en el mercado hay muchos tipos de estructuras prefabricadas para la construcción de Quadcopter. La mayoría de ellas están fabricadas de plástico y usan una forma optimizadas para minimizar el peso, pero sin embargo ninguna de ellas se ajusta al proyecto. Además la idea original del proyecto era la propia construcción de la estructura para poder satisfacer a nuestras necesidades perfectamente.

A la hora de hacer el diseño, las principales características a tener en cuenta son la resistencia y el peso. Pero desde luego, el peso es el parámetro crítico en el diseño, ya que a más peso se use más fuerza tendrán que hacer los motores, más capacidad de batería se necesitará y con lo que tendrá más peso aun.

Así pues para empezar a diseñar se debe saber cuál será el empuje máximo de los cuatros motores, para poder situar un punto de diseño a alcanzar. En este caso es un Quadcopter para tomar fotografías, así que no es muy necesario que disponga que mucha aceleración que Quadcopter, ni tampoco es de carga así que no hace falta que sea extremadamente ligero, con situarse en el 50-60 % del empuje máximo será suficiente para tomar fotografías.

Obviamente cuanto menos peso mejor, pero esto puede incrementar enormemente el coste del proyecto, por tener que usar materiales extremadamente caros pero muy ligeros y resistentes como es la fibra de carbono.

Otra característica importante del diseño a tener en cuenta es la modularidad, es decir, que cada uno de los componentes sean reemplazables fácilmente y dependan lo mínimo del resto de las piezas, así se consigue optimizar el tiempo de construcción y el de reparación.

5.4 Esbozo de la Estructura

La primera decisión a tomar en el diseño de la estructura es la dimensión del Quadcopter. En el caso de este proyecto se decidió que el Quadcopter iba a medir 50 x 50 cm. Como podemos ver en la figura siguiente, debido a la simetría de la estructura era muy fácil calcular el tamaño de las hélices. Simplemente es la mitad del ancho, es decir 25 cm. Para dejar un margen para el grosor de la carcasa, se decide usar hélices de 8 pulgadas, 20 cm. El espacio para colocar el hardware es el espacio interior definido por los 4 círculos de 25cm de diámetro.

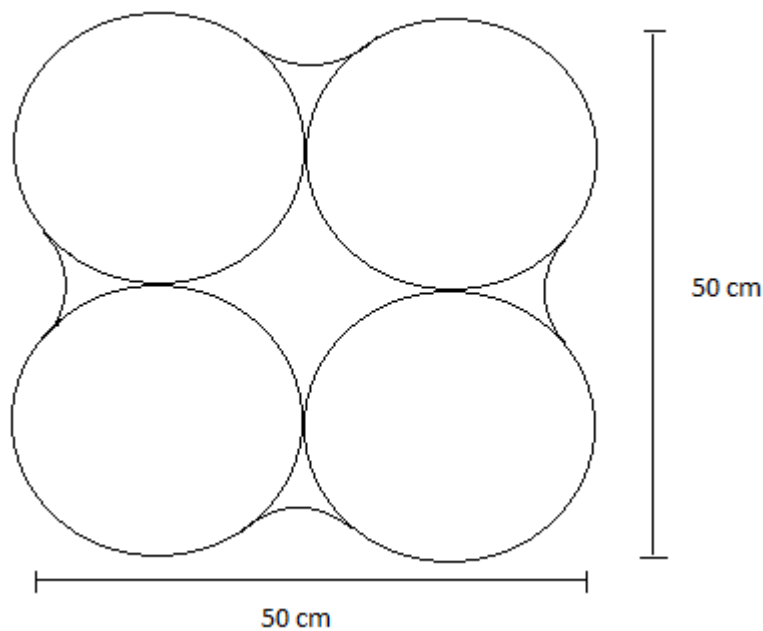


Figura 14: Esbozo

Una vez calculado el tamaño de las hélices, se adquirió los motores más baratos que usaran este tipo de hélice, tras consultar el fabricante, cada motor con esas hélices son capaces de levantar como máximo 550 gramos. Es decir el empuje máximo de Quadcopter sería 2200 gramos.

Para hacerse una idea de del aspecto el Quadcopter, se modelo mediante un Software de CAD (SolidWorks) varias piezas del Quadcopter y se imprimirá en una impresora 3d para ver la consistencia del modelo y el peso resultante. Tras varias pruebas se vio que el peso se era demasiado.

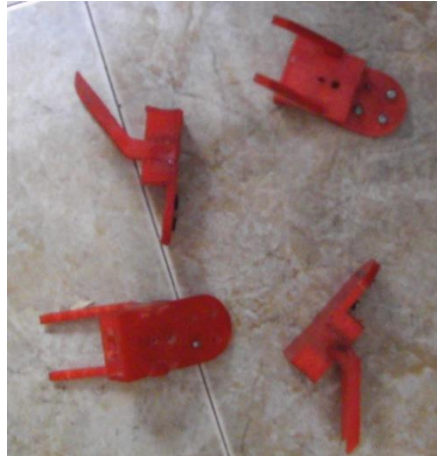


Figura 15: Prototipos de plástico

Finalmente se optó por hacer una estructura de aluminio. La principal ventaja del aluminio es que proporciona un alto ratio resistencia/peso, usando el espesor adecuado se puede conseguir una estructura altamente resistente y muy ligera.

La estructura final está compuesta por 4 barras de sección 11x7cm de espesor de 2 mm de 18 cm de largo. Las 4 barras están unidas a soporte central en forma de cruz. El soporte central está compuesto por unas platinas que conforman varios niveles para poder situar todo los componentes, un nivel para las baterías y otro para la placa electrónica y el Arduino. La platina superior es de aluminio y resto de platinas de fibra de carbono para intentar minimizar el peso.

5.5 Estructura Final

5.5.1 Chasis

El chasis constituye la estructura donde se ubican el resto de componentes. Está compuesto por 3 pletinas, la superior es de aluminio que es quien se encarga de otorgar la resistencia, y el resto está formado por platinas de fibra de carbono para minimizar el peso y están unidas mediante varillas roscadas de métrica 2 mm.

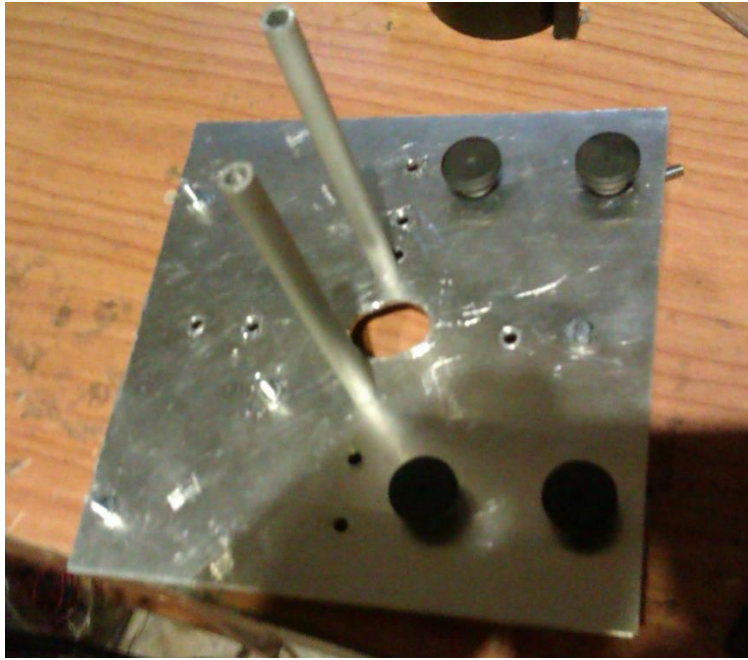


Figura 16: Chasis

5.5.2 Cruceta

Está formado por 4 barras de aluminio rectangulares de perfil 11x7mm y de espesor 2 mm huecas por dentro, que es por donde se pasan los cables desde el extremo de las barras al chasis. Además tiene atornillado 4 varillas de aluminio con 1 tornillo de métrica 4 mm, las cuales actúan como tren de aterrizaje. Cada barra está unida al chasis mediante 2 tornillos de métrica 3.

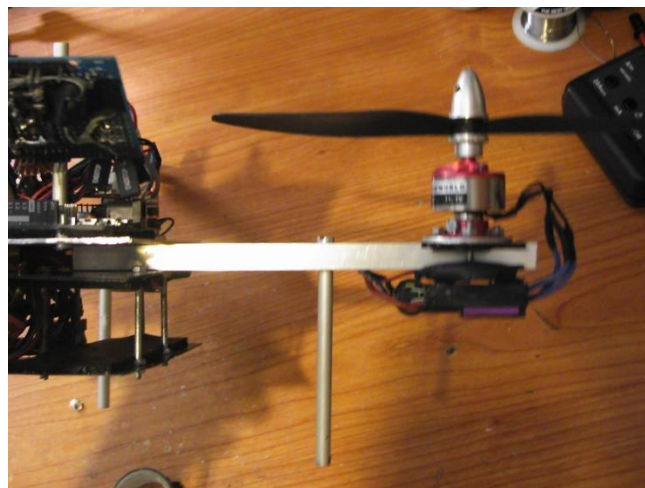


Figura 17: Barra de soporte de motor

5.5.3 Soporte de los motores

Está formado por 2 pletinas de aluminio circulares pasadas entre la cruceta para reforzar que la sujeción del motor al soporte. Además también sirve de sujeción para el circuito que controla la velocidad de los motores, los ESC.



Figura 18: Soporte de los motores

5.5.4 Carcasa

La Carcasa es el elemento fundamental para poder aislar las hélices y protegerlas de posibles impactos, y así evitar que las personas cercanas puedan salir heridas. La elección del material de esta parte es crítico, ya que es una parte muy voluminosa y puede conllevar un peso muy elevado. Se fueron probando con diferentes materiales, debido a que la forma de la carcasa es compleja y es difícil determinar que material ofrecerá el mínimo peso con una resistencia aceptable.

En total se probaron por orden temporal 3 materiales:

- Plástico PLA

Dado que la carcasa es simétrica solo se tuvo que diseñar en SolidWorks 3 piezas: el soporte central y el exterior del círculo y el soporte interior.

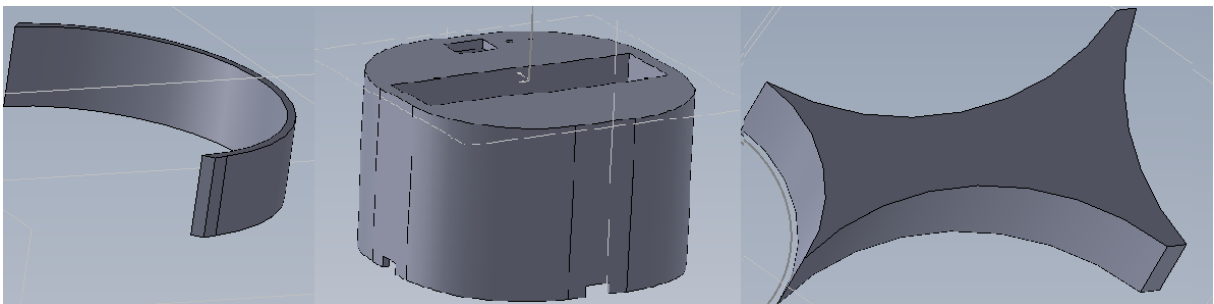


Figura 19: Piezas de la Carcasa

Para conformar la carcasa en total se imprimieron 4 soportes centrales, 4 exteriores y el soporte interior.



Figura 20: Piezas de la Carcasa en SolidWorks

El peso final fue 750 g, un peso que era inaceptable, ya que representaba 34 % del empuje máximo.

- Fibra de vidrio

En un intento de conseguir una carcasa más ligera, se probó a hacer una carcasa de fibra de vidrio usando de molde la carcasa de plástico. Debido a que la fibra de vidrio era muy barata. El proceso de construcción es a base de ir poniendo capas fibra de vidrio y resina, regulando espesor de estas capas se podría conseguir reducir el peso. Pero al poner la primera capa y dejar que se secará, ya pesaba 500 g y era muy débil la estructura. Así que se descartó esta opción.



Figura 21: Carcasa de fibra de vidrio.

- Poliestireno Extrusionado.

Finalmente, se decidió usar este material debido a que es muy ligero y que tiene una resistencia aceptable.

Se partió de una plancha de 50x50x5 cm, y se pintó el esbozo de la carcasa encima de una cara. Y usando una taladradora sujeta a un soporte, se siguió las marcas del esbozo para hacer los agujeros.



Figura 22: Plancha superior de la Carcasa de Poliestireno

Usando otra plancha de 3cm de alto, se marcó el perfil interior de la carcasa y se cortó de la misma forma que la anterior carcasa. Esta plancha tiene la utilidad de levantar la carcasa con el fin de un hueco para situar los componentes electrónicos.



Figura 23: Plancha inferior de la Carcasa de Poliestireno

Más tarde se unieron estas dos planchas mediante un pegamento especial para Poliestireno llamado ADEFIX. Esta carcasa está unida al chasis mediante 2 varillas que pasan por dos tacos especiales para repartir el esfuerzo generado por la varillas.

El peso final de la carcasa es 283 gramos, solo representa el 12 % del empuje por lo que es perfectamente asumible.



Figura 24: Carcasa en Poliestireno Extrusionado

5.5.5 Plataforma para la placa electrónica

Para tal de aislar la placa electrónica de la vibración producida por los motores, se separó la placa electrónica del chasis mediante unas gomas especiales, llamadas Silent-Block, que absorben parte de la vibración del sistema.

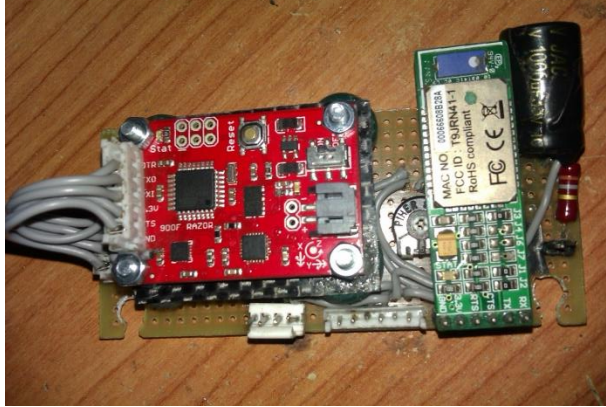


Figura 25: Plataforma de la placa Electrónica

Para conseguir más aislamiento para los sensores se usó la misma estrategia una plataforma unida mediante unas gomas Silent-Block a la placa electrónica.

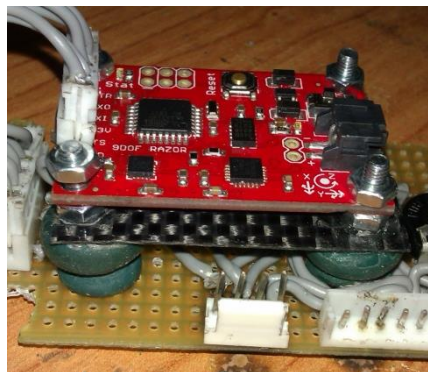


Figura 26: Plataforma de los sensores

5.3.6 Hélices

Se ha escogido hélices bipala especiales para Quadcopter, en concreto el modelo de 8"x4.5" (miden 8 pulgadas y tienen 4.5 pulgadas de paso). Están unidas al motor usando un portahélices de 3 mm. Así se podrá remplazar cualquier hélice de manera sencilla, rápida, y minimizar las vibraciones del sistema.

Debido a la configuración del Quadcopter debe haber un par de hélices que giran en sentido horario (llamadas normales) y otro par que giran en sentido anti horario (llamadas inversas).

Esta configuración es así con el fin de controlar el giro en eje de Yaw, ya que un par de hélices generan un torque en sentido y el otro par para en el sentido opuesto.

En cuanto al material de la hélice, la venden de dos tipos: de plástico y de fibra de carbono. Las fibras de carbono son más resistentes y más ligeras que las de plástico. Se puede usar indistintamente, ya que cambian las propiedades del motor bastante poco (empuje y vibración).

Lo más importante es que las hélices estén equilibradas, es decir, que cada pala sea simétrica a la otra pala y pesen lo mismo, sino pueden producir vibraciones inesperadas y hay que proceder al equilibrado.

En varios casos, tras la compra y el montaje de la hélice, se vio que estaban ligeramente desequilibradas. Para proceder al equilibrado, se debe pasar por el agujero central de la hélice una varilla que encaje con el ancho de agujero, el cual nos sirva de apoyo. Dejando la hélice en reposo, está debería quedar en posición horizontal y sin que nada las perturbe. En el caso en que la hélice se incline hacia un lado es indicativo de que la hélice esta descompensada. Para corregir esta desviación, hay varias formas o se añade peso en una pala o se quitar peso de la pala opuesta. La forma más recomendada es limar ligeramente la superficie de la pala que se inclinó hacia abajo. Se debe repetir este operación hasta conseguir que la hélice se mantenga completamente horizontal.



Figura 27: Hélice normal, inversa y porta hélices de 3 mm

5.3.7 Peso total

El prototipo final pesa en total 1483 gramos, que representa un 67% de empuje total, que se pasa ligeramente del rango calculado inicialmente, debido a imprevistos durante la construcción. Pero para tomar fotografías sigue siendo perfectamente válido.

Para tener una idea de la distribución del peso del prototipo, y así saber que partes son las más pesadas, se ha elaborado una lista de los componentes que potencialmente representan una carga de peso.

Descripción	Peso (g)	Unidades	Peso Total(g)
Hardware			857,1
Arduino Due	45	1	45
Motor Brushless Emax CF 2822	39	4	156
ESC EMAX 25 A	40	4	160
Batería Lipo 2200 mAh	190	2	380
Protoboard para Arduino	36	1	36
Placa Electrónica			
Razor 9 DOF	1,1	1	1,1
Bluetooth Stick	6	1	6
Cámara TLL	8	1	8
Otros componentes	10	1	10
Interruptor unipolar de 20 A	5	1	5
Cable de Sección 2,5 mm ² 1m de largo	15	1	15
Cable de Sección 0,2 mm ² 10m de largo	25	1	25
Estructura mecánica			625,9
Hélices 8"	6	4	24
Portahélices	4	4	16
Carcasa Poliestireno	283	1	283
Chasis	110	1	110
Soporte placa electrónica	15	1	15
Cruceta			
Soporte para los motores	3	8	24
Barra de aluminio	16	4	64
Varillas de aluminio	4	4	16
Tornillos de métrica 3 20 mm de largo	1,5	31	46,5
Tornillos de métrica 4 20 mm de largo	2	6	12
Tuercas de métrica 3	0,4	31	12,4
Tuercas de métrica 4	0,5	6	3
Peso Total(g)			1483

Tabla 3: Distribución del Peso del Prototipo.

Se interesante recalcar que donde está el mayor peso del prototipo es en el Hardware y no la Estructura mecánica. Y que la carcasa representa casi un 50 % de la estructura mecánica, pero es un elemento necesario para proteger el propio Quadcopter.

6. Diseño del Hardware

6.1 Bases de Diseño

A la hora de diseñar el hardware, las principales necesidades que hay que hacer frente para hacer volar el Quadcopter son las siguientes:

- Determinar qué tipo motor y que características son necesarias
- Seleccionar el circuito que controlará los motores.
- Determinar cuanta potencia y energía es necesaria.
- Determinar qué tipo sensores nos hacen falta para determinar la orientación.
- Seleccionar un microcontrolador que sea capaz de comunicarse con los otros y que tenga suficiente capacidad de cálculo para procesar toda la información.

En los siguientes apartados se discute cada uno de los componentes del hardware y porqué han sido seleccionados.

6.2 Motores

6.2.1 Selección de los motores

La primera decisión a tomar es el tipo motor que nos hace falta, básicamente hay de 2 tipos: motores con escobillas o sin escobillas. Los primeros utilizan las escobillas para contactar con el eje de rotación del motor, como consecuencia estas escobillas se desgastan y es ineficiente. En contraposición, los motores sin escobillas (brushless) no tienen conexión física entre las partes eléctricas en movimiento. Así pues, son mucho más eficientes y elimina en gran parte el mantenimiento. Pero tiene un coste mucho más alto.

Dentro de los motores sin escobillas existen dos tipos: los outrunner y los inrunner.

Los outrunner son motores que están diseñados para trabajar a bajas revoluciones en aplicaciones de alto torque. La disposición de los imanes permanentes en este tipo de motores están en la carcasa externa, mientras que el bobinado se encuentra fijo en la bancada. La zona móvil del motor es la propia carcasa.

En cambio los motores inrunner están diseñados para trabajar a altas revoluciones, en aplicaciones de par bajo. En este tipo de motor es similar a un motor DC con escobillas, pero los imanes están fijados al rotor.

En nuestro prototipo se han usado motores brushless outrunner ya que su principal virtud es que tiene mucha potencia en muy poco peso. Estos motores son controlados mediante una placa electrónica, llamada ESC (Electronic Speed Controller) que generan 3 señales de sinusoidales que están conectadas al motor y regula su velocidad.

6.2.2 Motores seleccionados: EMAX CF2822

El motor escogido son los EMAX CF2822, ya que eran los motores más baratos que permitan usar hélices de 8", siendo su empuje máximo 550 g para hélices de 8".



Figura 28: Motor EMAX CF2822

Las características principales de motor Brushless son: el empuje máximo, la constante KV y la intensidad máxima del motor y el tipo de hélice que recomienda el fabricante.

Número de celdas de batería	2-3X Li-Poly
Dimensiones del estator	22x10 mm
Diámetro del eje	3 mm
Peso	39 g
Empuje	De 200 a 600 g
Tamaños de la hélice	8", 9", 10"
Constante KV	1200
Rpm Max	7700
Intensidad máxima	15 A

Tabla 4: Tabla de características de los motores

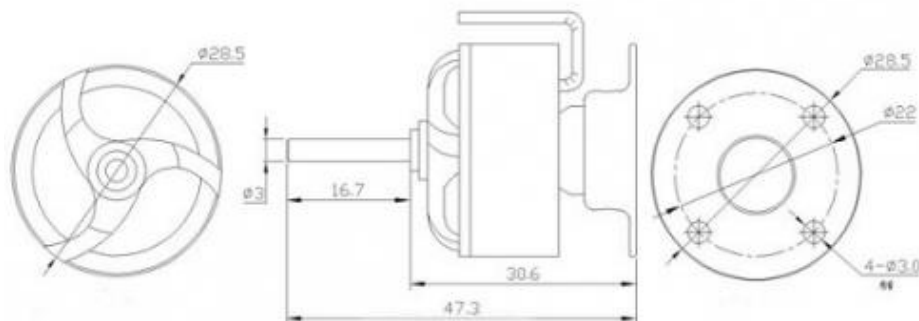


Figura 29: Dimensión del motor (en mm)

6.2.3 Caracterización de los motores

Es importante saber el comportamiento del motor a mediada que son regulados por el ESC, es decir, saber que empuje hay para cada una de las entradas del ESC. Para así comprobar que los 4 motores que se usan tienen el mismo comportamiento y tener la posibilidad de linealizar los motores.

Para realizar este ensayo para cada uno de los motores, se ha montado un banco de ensayos formado por una barra cilíndrica de cobre apoyada sobre dos soportes que permiten que la barra gire. El Quadcopter se acopla a la barra de cobre mediante dos tornillos de métrica 4 mm a las varillas del tren de aterrizaje, así permitiendo girar el Quadcopter. El principio básico es usar la ley de la palanca, es decir, situar a la misma distancia a la que se encuentra separado el motor de la barra cilíndrica, una balanza, tal y como podemos ver la figura siguiente.



Figura 30: Banco de caracterización de los motores

Otra condición importante para el ensayo es que todos los motores reciban el mismo voltaje. Para ello se usa una fuente de alimentación de 12 V que sea capaz de alimentar los motores con la intensidad máxima es decir 15A. La mayoría de los transformadores comerciales que se pueden encontrar solo generan entre 2-3 A, pero una fuente de PC es relativamente barata y es capaz de dar más 15 A (Fuente de 500W), simplemente hay que puentear una conexión para usarlas sin el propio PC. Así que se optó por usar una fuente de PC. Como consecuencia se evita el uso de la batería. Ya que a medida que se usa, su voltaje disminuye y hace difícil garantizar las mismas condiciones para todos los ensayos. Además se le añadió un conector T-Dean a la Fuente PC para alimentar el Quadcopter



Figura 31: Fuente de PC con adaptador T-Dean

Los resultados de los ensayos son los siguientes:

Entrada (%)	Motor 1 (g)	Motor 2 (g)	Motor 3 (g)	Motor 4 (g)
0	0	0	0	0
5	6	6	8	6
10	28	29	28	29
15	54	54	53	55
20	79	85	78	84
25	113	117	112	119
30	156	158	153	160
35	192	196	187	198
40	231	235	225	240
45	266	273	259	275
50	300	304	291	307
55	338	344	327	343
60	374	377	361	378
65	401	410	388	408
70	419	431	408	424
75	436	454	427	448
80	451	470	450	468
85	472	486	464	484
90	490	515	483	509
95	510	515	494	524
100	550	580	532	566

Tabla 5: Caracterización de los motores

Se puede observar que hay motores que hacen más empuje que otros. Una posible forma igualar todos los motores sería mediante software. Hacer que el 100% de empuje para el resto de los motores sea el empuje más pequeño de los cuatro motores.

Pese a este ligero desequilibrio, el algoritmo de control ya se encarga de absorber estas pequeñas diferencias entre los motores, únicamente ajustándolo conseguiremos una respuesta más estable.

6.3 ESC (Electronic Speed Controller)

El ESC es un circuito que se encarga de generar una señal trifásica que alimenta el motor. La velocidad de giro se varía mediante una señal suministrada por el PWM.

El ESC seleccionado es una ESC EMAX de 25 A con fin de sobredimensionar con respecto a los 15 A que consume cada motor.

Este ESC permite ajustar el rango de PWM a un mínimo 1ms a un máximo 2 ms, por defecto detecta el 0% de empuje a 1,2ms y el 100% de empuje a 2ms.



Figura 32: ESC EMAX 25 A

Además incorpora varios tipos de protecciones:

- Protección de baja tensión:
Ya sea para apagar el motor de inmediato o para bajar la potencia cuando la entrada tensión cae por debajo del umbral de protección programado el motor deja de proporcionar par.
- La pérdida de la protección de señal:
Se reduce la Potencia automáticamente al 20% o menos cuando se pierde la señal durante 1 segundo, y se reanuda cuando se detecta la señal de nuevo.
- La protección del sobrecalentamiento: Cuando la temperatura aumenta hasta por encima de 110 grados Celsius, la potencia es reducida hasta un 35%.

- Hardware auto-comprobación:

El sistema comprobará por sí mismo cuando la batería está conectada. Cualquier fallo de hardware, provocará 20 sonidos cortos en forma de "Beep".

6.4 Batería

6.4.1 Selección de la Batería

A la hora de seleccionar una batería hay que tener en cuenta principalmente tres cosas:

- Intensidad de descarga constante.
- La capacidad de la batería.
- El voltaje de la batería

Se escogió usar baterías Lipo porque son capaces de subministrar mucha energía en poco tiempo y siendo a la vez muy ligeras, en contra posición de la típicas baterías de plomo o Niquel-Cadmio.

Dentro de las baterías Lipo, se escogió la marca Tiger Power 2200mAh 3S (11,1V) 25C porque se quería garantizar un tiempo de vuelo largo. Para poder para cumplir con la intensidad que tiene que subministrar a los motores (60 A) se conectan dos baterías del mismo tipo en paralelo.



Figura 33: Batería Lipo 2200 mAh

Número de celdas de batería	3
Dimensiones	26x34x106mm
Capacidad	2200 mAh
Peso	190 g
Intensidad de Descarga Máxima	25 C (55 A)
Intensidad de Carga	2 C (4,4 A)
Intensidad de Descarga Pico	50 C (110 A)

Tabla 6: Característica de la Batería Lipo

Hay que añadir que estas baterías se cargan mediante un cargador que balancea las 3 celdas de la batería Lipo, para así asegurar que todas tienen el mismo voltaje.



Figura 34: Cargador de Baterías Lipo

6.4.2 Calculo del Tiempo de Vuelo

En el prototipo hay 2 baterías Lipo de 2200 mAh puestas en paralelo, con el objetivo de doblar la autonomía. En total hay disponibles 4400 mAh a un voltaje nominal de 11.1 V.

Para realizar el cálculo solo se tiene en cuenta el consumo de los motores, ya que es resto de despreciable en comparación con el consumo de los motores.

Consumo de los motores a máxima potencia: $11.1 \text{ V} \cdot 15 \text{ A} = 166,5 \text{ W}$

Consumo total de los motores: $166,5 \text{ W} \cdot 4 = 666 \text{ W}$

Tiempo de vuelo a máxima potencia: $\frac{E}{P} = \frac{4,4 \text{ Ah} \cdot 11,1 \text{ V}}{666 \text{ W}} = 0,0733 \text{ h} = 4,4 \text{ min}$

Este es el tiempo volando a máxima potencia, pero tras varias pruebas a media potencia la autonomía puede llegar hasta 10 min.

6.5 Sensores

A la hora de seleccionar, se busca un placa que integre como minimo el magnetometro, el acelerómetro y el giroscopio. Principalmente la decisión a tomar estaba entre dos circuitos que incluían ya los 3 sensores:

- **AltImu-10**

El AltImu-10 es una placa de medición inercial que incorpora un giroscopio de 3 ejes L3GD20, un acelerómetro con magnetómetro de 3 ejes LSM303DLHC y un barómetro LPS331AP. Todos los sensores funcionan con el bus I2C que permite lecturas utilizando solo una línea de datos. Funciona con un rango de alimentación de 2.5V hasta 5.5V.

La placa también es compatible con software AHRS para calcular orientación en el espacio. Se puede usar por ejemplo en un Quadcopter, en guantes de posición o cualquier otra aplicación donde se necesite orientación.

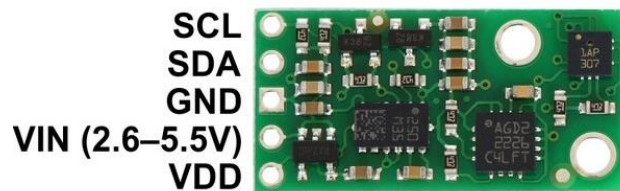


Figura 35: Sensor AltImu-10

- **Razor 9 DOF**

El Razor 9DOF es un sistema de medición inercial que dispone de 3 sensores de alta calidad, un giroscopio de tres ejes ITG3200, un acelerómetro ADXL345 de 3 ejes, y un compás/magnetómetro HMC5883L de 3 ejes. El conjunto proporciona 9 grados de libertad para medición inercial. Todas las salidas de los sensores son procesadas por un ATmega328 que envía a su vez la información por el puerto serie UART para poder recuperarlos de forma limpia.

La placa viene programada para ser utilizada con el IDE de Arduino, además por defecto tiene un firmware que envía los datos de los sensores por el puerto UART y permite ver su funcionamiento.

Puede conectarse un conversor Serie/USB de 3,3V directamente a su salida.



Figura 36: Razor 9 DOF

Al final se ha seleccionado el Razor 9 DOF, ya que incorpora un propio microcontrolador para procesar la información de los sensores, y así hacer un preprocesado de la información y devolvernos de manera sencilla la información por el puerto UART. Además para esta placa existen librerías que permiten usar esta placa como un AHRS, es decir, como sistema para calcular la orientación en el espacio. Más a más, permite ser programado como un Arduino, dejando la posibilidad de modificar el firmware fácilmente.

6.6 Bluetooth

La idea original del proyecto era usar un Stick Bluetooth para comunicarse con la aplicación de control, ya que el protocolo Bluetooth es muy sencillo. Se podría haber usado WIFI para la comunicación que en comparación con el Bluetooth tiene una velocidad de transmisión muy elevada, pero en cambio el protocolo Wifi es más complejo de implementar. Por tanto debido al tiempo limitado de este proyecto se optó por usar Bluetooth.

El modulo seleccionado es un Bluetooth Stick RN-41.

Éste módulo funciona con UART (TX/RX) y permite establecer una comunicación inalámbrica con una velocidad de transferencia de entre 9600 y 115200bps. El alcance aproximado son unos 100 metros.

Tiene que ser alimentado con una tensión de 3,3V.



Figura 37: Bluetooth Stick

6.7 Cámara

En cuanto a la cámara, se decido usar una cámara Jpeg que se comunica mediante UART.

La UART es protocolo relativamente sencillo de usar pero tiene el inconveniente que puede llegar a ser muy lenta la transmisión de la fotografías, ya que el baudrate más alto compatible es de 115200 bps y una foto de 640x480 píxeles puede ocupar varios Kilobytes.

La Cámara de video JPEG es una cámara integrada que puede ser conectada prácticamente a cualquier sistema encastado que necesite tomar imágenes.

La cámara utiliza un módulo CMOS VGA de alta calidad con soporte de compresión JPEG. La resoluciones admitidas son 640x480, 320x240 y 160x120 píxeles en color. La lente está incorporada.

Lo interesante de esta cámara son sus extras, por ejemplo dispone de foco ajustable, auto balance de blancos, ajuste de brillo automático y contraste automático. También incluye sensor de movimiento integrado por lo que se puede hacer que capture una fotografía cuando algo se mueve.



Figura 38: Cámara Jpeg TTL

6.8 Microcontrolador

A la hora de seleccionar el microcontrolador, hay que tener en cuenta que es el encargado de gestionar y procesar toda la información de los otros dispositivos: Cámara, Bluetooth, Sensores y Motores.

Así pues el microcontrolador que nos hace falta, debe de tener todas conexiones necesarias para comunicarse con los otros dispositivos, que son:

- 3 UART para los Sensores, Cámara y Bluetooth.
- 4 PWM independientes para controlar los Motores.

Principalmente la decisión a tomar estuvo entre estos dos micros controladores:

- Arduino Due

El Arduino Due es una placa electrónica basada en la CPU Atmel SAM3X8E ARM Cortex-M3. Es la primera placa Arduino basado en un microcontrolador núcleo ARM de 32 bits. Lleva 54 entradas / salidas digitales (de las cuales 12 se pueden utilizar como salidas PWM), 12 entradas analógicas, 4 UARTs (puertas seriales), un reloj de 84 MHz, una conexión capaz USB OTG, 2 DAC (de digital a analógico), 2 TWI, un conector de alimentación, una cabecera de SPI, un encabezado JTAG, un botón de reinicio y un botón de borrado.

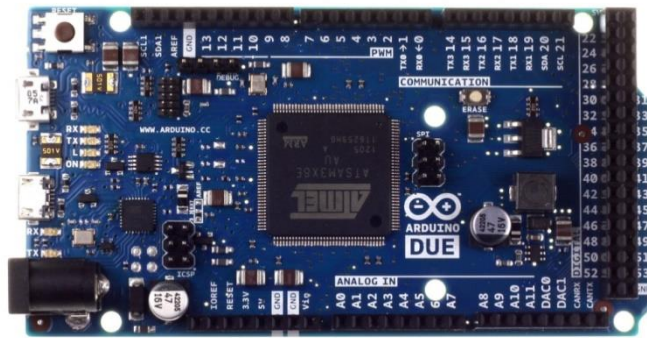


Figura 39: Arduino Due

- Raspberry Pi

Raspberry Pi es un ordenador de placa reducida de bajo coste, desarrollado en Reino Unido por la Fundación Raspberry Pi.

El diseño incluye un System-on-a-chip Broadcom BCM2835, que contiene un procesador central (CPU) ARM1176JZF-S a 700 MHz (el firmware incluye unos modos "Turbo" para que el usuario pueda hacerle overclock hasta 1 GHz sin perder la garantía), un procesador gráfico (GPU) VideoCore IV, y 512 MiB de memoria RAM). El diseño no incluye un disco duro ni unidad de estado sólido, ya que usa una tarjeta SD para el almacenamiento permanente.

Aparte tiene salida de video HDMI, salida de Jack de audio, conexión Ethernet y dos USB.

Dispone un chip GPIO que otorga las siguientes características:

- 8 Entradas/salidas digitales
- 1 UART
- 1 SPI
- 1 I2C

Además permite instalar una versión Linux que lo convierte básicamente en un ordenador portátil. Hay que destacar que es muy barato para tener todas estas características, salió al mercado al precio de 50 euros.



Figura 40: Raspberry Pi Modelo B

Se decantó por usar el Arduino Due, ya que tenía todas conexiones que nos hacía falta número de UARTS y PWM que la Raspberry Pi no disponía y la potencia de cálculo del Arduino ya era suficiente para implementar una algoritmo de control basado en PID.

Por otro lado, desde un inicio se descartó el uso de PICS y/o DSP ya que Arduino y Raspberry Pi ofrecen una gran facilidad a la hora de programar y está muy bien documentado. Así podremos céntranos en hacer aplicaciones más complejas en mucho menos tiempo. En único inconveniente es el precio, el coste de un PIC o DSP puede estar entre 1 a 10 euros, en cambio Arduino cuesta 50 euros, pero para hacer un solo prototipo es asumible.

6.9 Placa de electrónica

En total se tuvo que diseñar dos placas electrónicas por una cuestión de espacio y modularidad.

La primera es la encargada de integrar a todos los periféricos (Cámara, Bluetooth y Sensores) y alimentar mediante una fuente propia estabilizada, para así eliminar cualquier interferencia producida por los motores.

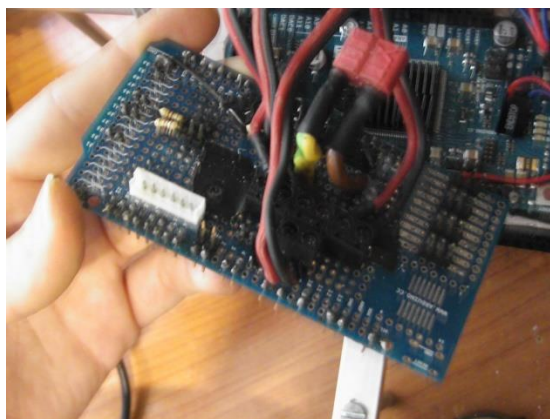


Figura 41: Protoboard Arduino

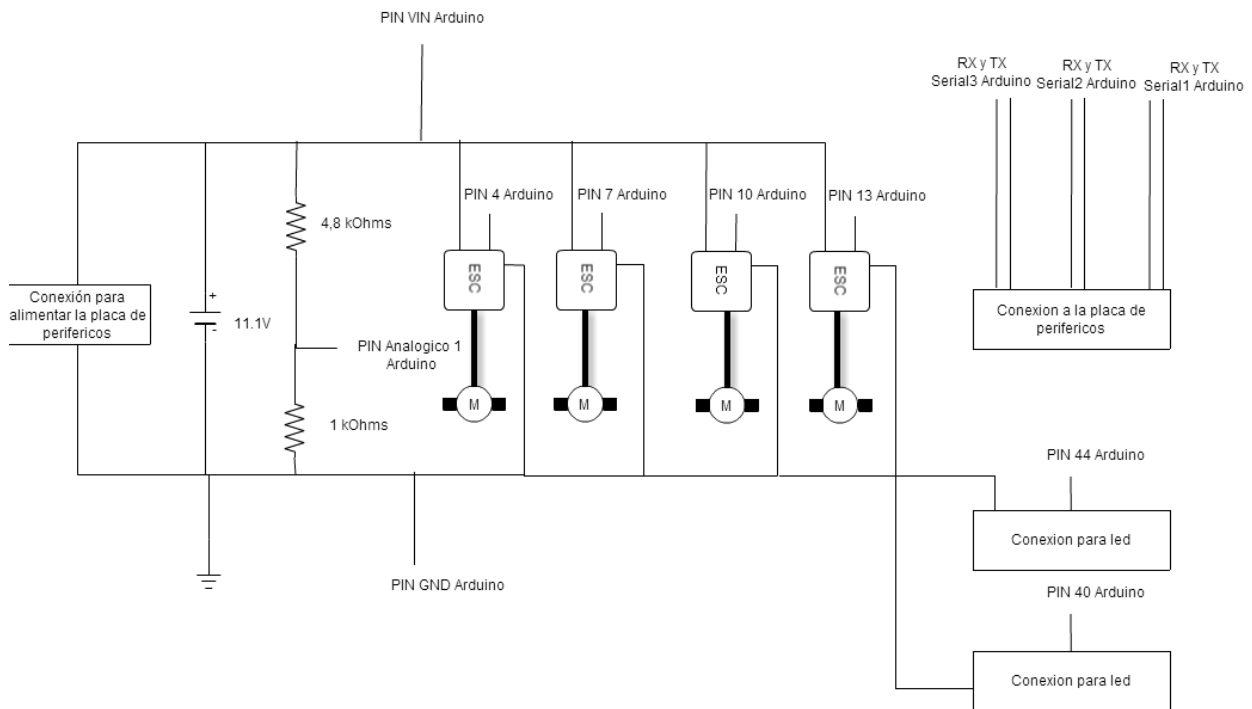


Figura 42: Esquema Electrónico Protoboard

La segunda es la Protoboard que se conecta directamente al Arduino, y es quien se encarga de las siguientes funciones:

- Conectar los motores y alimentar los motores
- Monitorizar el voltaje de la batería mediante un divisor de tensión.
- Conectar la otra placa electrónica al Arduino.
- Dispone 2 conexiones para conectar 2 Leds.

Además tiene un filtro RC calculado para cortar a toda frecuencia superior a 80 Hz aproximadamente.



Figura 43: Placa de los periféricos

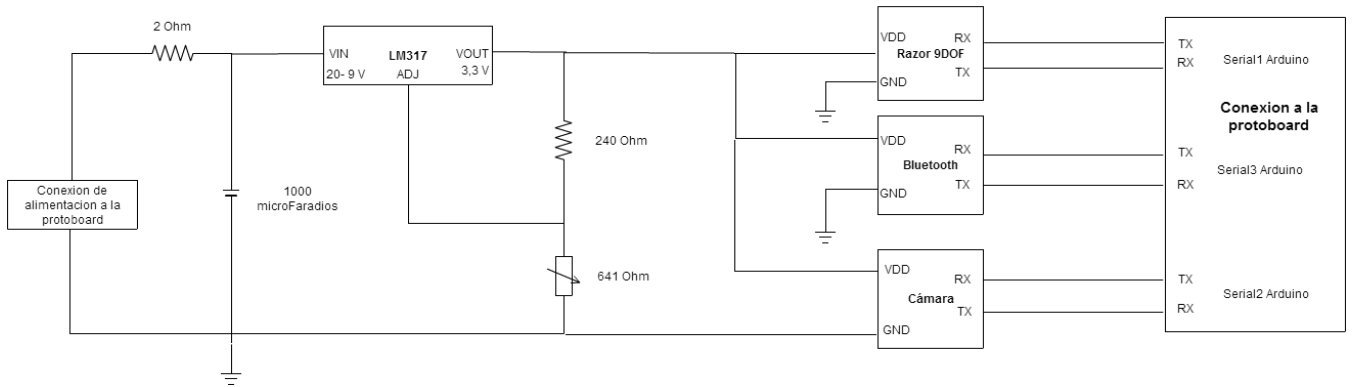


Figura 44: Esquema Electrónico Placa Periféricos

6.10 Configuración Final

Para tener una visión global del sistema se ha hecho un diagrama de bloques donde se puede ver a grandes rasgos como es el conexionado del hardware.

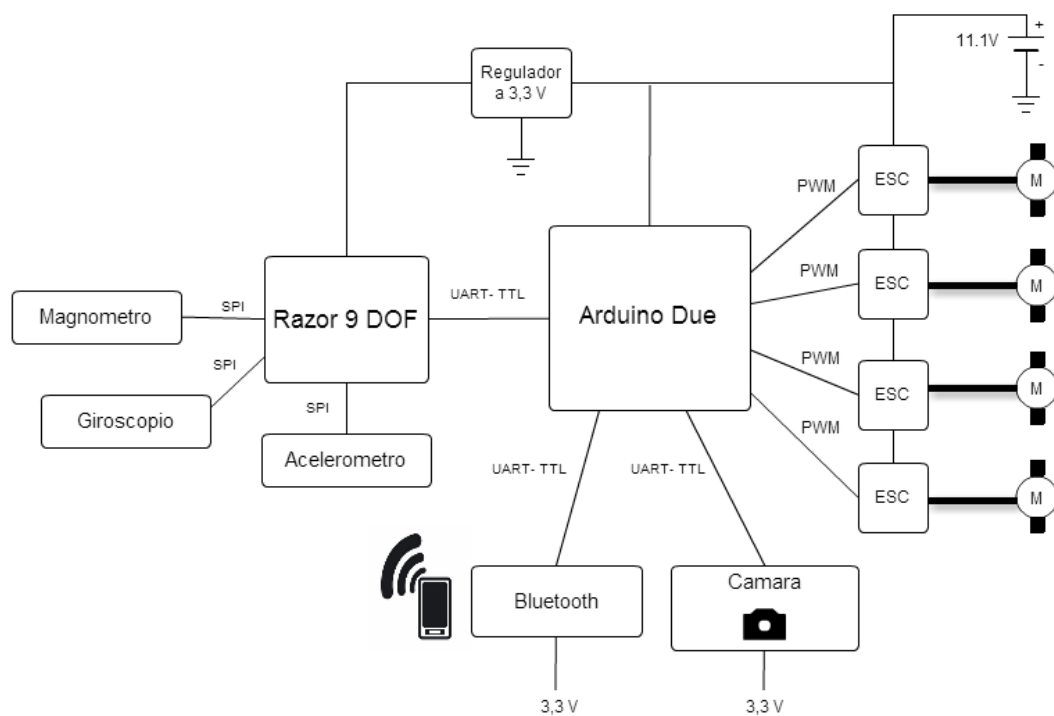


Figura 45: Diagrama de Bloques del Sistema

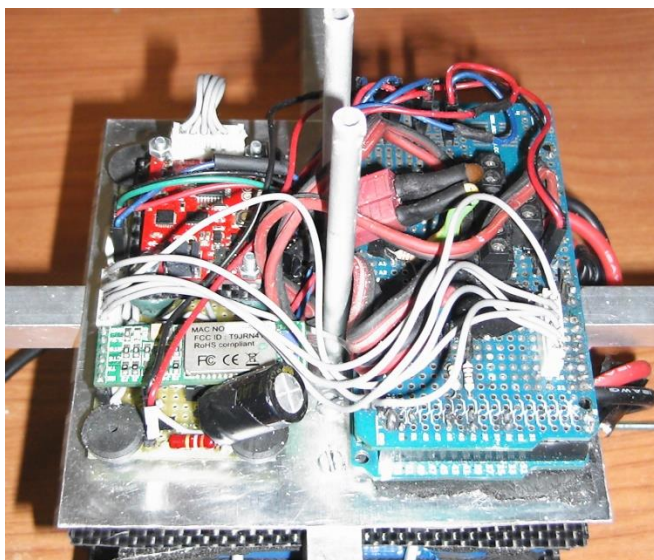


Figura 46: Montaje Final del Hardware

7. Diseño Software

7.1 Entorno de Programación

La elección de Arduino como microprocesador implica el uso de IDE para el Desarrollo. Es un entorno de desarrollo basado en Processing[4] y Wiring[5]. La interfaz es muy similar a la del Processing y incorpora todas las librerías que tiene Wiring. Además se programa en C++.

Tiene un gran abanico de librerías que simplifican enormemente cualquier tarea a realizar como lectura y escritura de los puertos digitales y analógicos, implementación de los protocolos de comunicación como SPI, I2C, UART y como además también permite el control de timers y de los PWM. Estas son de gran utilidad ya permite desarrollar de forma rápida y sencilla.

Como se puede ver en la siguiente figura, es un entorno sencillo y muy amigable. Tiene muchas opciones preconfiguradas que permiten simplificar procesos como la propia compilación y carga del programa en el Arduino, la comunicación con el PC para debugar y la inclusión de librerías.

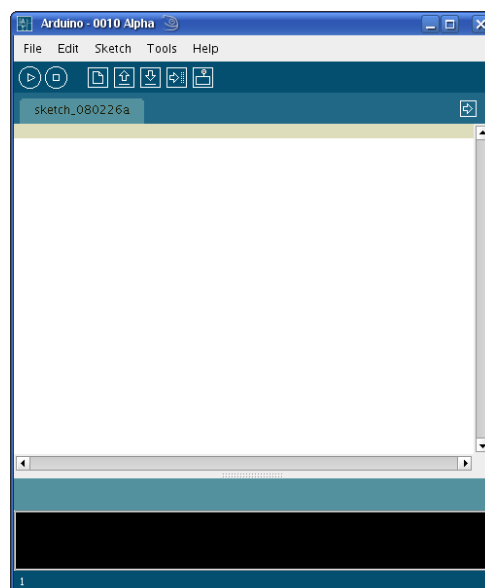


Figura 47: IDE de Arduino

Por otro lado, Arduino viene un conjunto de programas donde se puede estudiar cada una de las librerías que viene con el programa y aprender como usarlas. Acelerando así el aprendizaje sobre este entorno.

7.2 Software de Arduino

En cuanto al diseño de software, se ha decidido usar un diseño modular, es decir todas la funcionalidades que están relacionadas entre sí van a misma clase, así aumentar la independencia de las clases y que sean remplazables fácilmente. En total hay 7 módulos:

- Módulo AHRS
- Modulo Bluetooth
- Módulo Cámara
- Modulo Control
- Modulo Leds
- Modulo Motores
- Modulo Principal

En siguiente diagrama de clases de podemos qué tipo de relación hay entre los módulos:

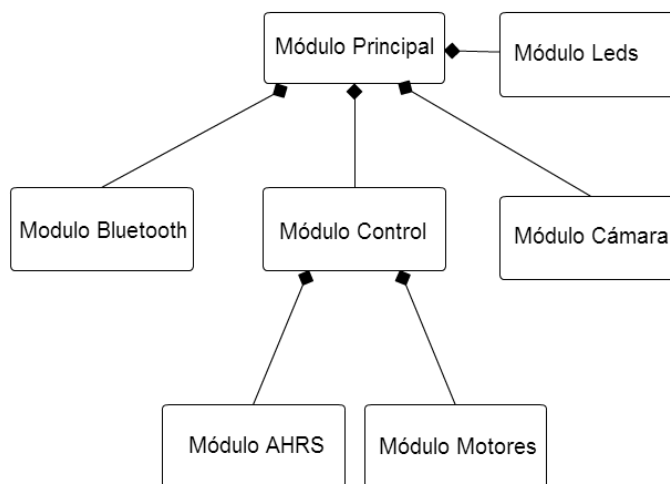


Figura 48: Diagrama de clases Arduino

7.2.1 Modulo de Motores

Este módulo se encarga de gestionar todas las funciones con los motores. Se comunica el Arduino con los ESCs mediante PWM.

7.2.1.2 Calibrado y uso de motores

Para poder usar los motores, hay que tener claro el protocolo que usan los ESC para el inicio y calibrado.

- Inicio normal

Se conecta el ESC a la alimentación del sistema (normalmente mediante la conexión de las baterías). El ESC detecta el valor mínimo del PWM, y hace un largo sonido en forma de "beep". Entonces el sistema detecta voltaje de la batería y hace varios sonidos cortos en

forma de "beep", lo que denota el número de células de la batería y finalmente el ESC lleva a cabo la auto-comprobación. Si es normal, se oye una melodía "♪ 1 2 3".

- Procedimiento de calibrado de rango del ESC

Se conecta el ESC a la alimentación del sistema, el ESC detecta el PWM máximo y hace dos sonidos en forma de "beep", lo que denota que Max PWM ha sido confirmada y se guarda. (Si mantiene pulsado durante más de 6 segundos después de este paso, se le llevó al modo de programación). Después de 6 segundos el sistema detecta la mínima señal del PWM y hace un largo sonido en forma de "beep". A continuación el ESC detecta voltaje de la batería y hace varios sonidos cortos en forma de "beep", lo que denota el número de células de la batería. Y finalmente el ESC lleva a cabo la auto-comprobación. Si es normal, se oye una melodía "♪ 1 2 3". Si el sistema no detecta la señal de PWM, hace constantemente sonidos en forma de " beep". Cualquier fallo en la auto-comprobación, hará que suene 20 sonidos cortos en forma de "beep".

7.2.1.2 Funciones

El modulo contiene básicamente 3 funciones principales:

- Armado de motores:

Esta función se usa para armar los motores para poder usarlos, consiste en poner el PWM mínimo a todos los motores.

- Calibración de los motores

Esta función se usa calibrar los motores, es decir, ajustar al valor de PWM máximo y mínimo definido en firmware de los ESC.

- Modificar la velocidad de los motores

Esta función se usa para poner el motor seleccionado a un porcentaje de su empuje máximo.

7.2.2 Modulo de Bluetooth

Este módulo se encarga de gestionar todas funciones relacionadas con el Bluetooth. Se comunica mediante una interfaz UART (Serial3 de Arduino).

7.2.2.1 Protocolo de comunicación

Para hacer la comunicación segura y fiable se tuvo que implementar un protocolo nuevo. Se basa en agrupar la información y tener un pequeño checksum para así estar seguros que la información se ha enviado correctamente.

Así pues, la información se agrupa en paquete que tiene la siguiente forma, según la función del paquete podemos distinguir dos tipos:

- Paquete de comandos
- Paquete de Datos

Paquete de Comandos

Este paquete se usa para enviar los comandos con parámetros:

0xFF	Comando	Longitud del Paquete	Byte 0 del Parámetro	byte 1 del Parámetro	CHECKSUM
------	---------	----------------------	----------------------	----------------------	----------

Tabla 6: Estructura del Paquete de Comandos (cada caja representa un byte)

El checksum se calcula sumando todos los bytes de los paquetes (a excepción del checksum) se aplicar al número resultante la operación lógica NOT y posteriormente se aplica la operación lógica AND con 0xFF.

Aunque el paquete de comandos tiene siempre la misma longitud, se envía para tener la puerta abierta a futuras ampliaciones del protocolo.

Paquete de Datos

Hay que añadir que para enviar la información de la cámara, la forma del protocolo se ha modificado ligeramente. Ya que las imágenes pueden ocupar hasta varios kilobytes, se dividen en partes de 512 bytes, para ser eficientes y no dejar el Arduino colgado solo enviado. Aunque el último paquete que se envía, tiene un tamaño es variable por eso hace falta enviar la longitud del paquete.

0xFF	Byte 0 longitud del paquete	Byte 1 longitud del paquete	Byte 0 longitud de la imagen	Byte 0 longitud de la imagen	DATA	CHECKSUM
------	-----------------------------	-----------------------------	------------------------------	------------------------------	------	----------

Tabla 7: Estructura del Paquete de Datos del Bluetooth

7.2.2.2 Funciones

El modulo contiene básicamente 3 funciones principales:

-Recibir el comando desde la aplicación de control.

Se encarga de estar consultando a todo momento si recibe un comando nuevo, en el tal caso del devuelve el comando recibido, y si no recibe ningún comando, devuelve que no hay un nuevo comando.

-Enviar la información a la aplicación de control.

Se encarga de encapsular la información usando el protocolo definido. Y calcular el checksum del paquete.

7.2.3 Modulo de Control

7.2.3.1 Control del movimiento del Quadcopter

Para poder controlar el Quadcopter, es necesario saber cómo modificar los motores para conseguir mover el Quadcopter en los 4 grados de libertad que dispone. Para ello se ha definido una matriz de comandos:

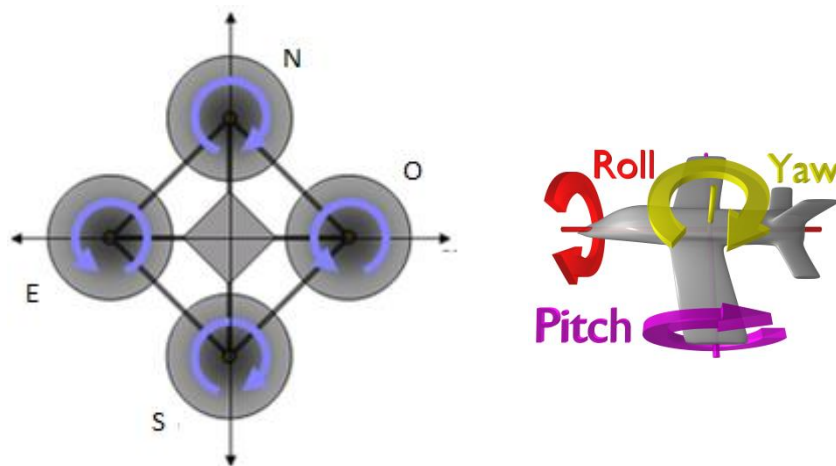


Figura 49: Disposición del Quadcopter

Matriz de comandos

	Norte	Sud	Este	Oeste
Throttle	+	+	+	+
Pitch	-	0	+	0
Roll	0	-	0	+
Yaw	-	+	-	+

Tabla 8: Matriz de comandos

El Quadcopter tiene cuatro motores que pueden girar a diferentes velocidades, por lo que parece razonable que tiene cuatro "grados de libertad" que se pueden controlar.

Los cuatro grados de libertad controlables son:

1. Throttle
2. Pitch
3. Roll
4. Yaw

Este paso se centra en cómo llegar de esos cuatro comandos (los inputs) a las cuatro velocidades del motor (las salidas).

Viendo el esquema anterior somos capaces de manera intuitiva averiguar cuál debería ser la asignación de entrada / salida para un Quadcopter. Es más fácil de representar como una matriz de 4x4, que es sólo una tabla donde las filas son entradas y las columnas son salidas (o viceversa).

Un + indica que el motor debe ser acelerado en respuesta a un comando de entrada positiva.

Un - indica que el motor debe ser frenado en respuesta a un comando de entrada positiva.

Un 0 indica que la velocidad del motor no se ve afectada por un comando de entrada positiva.

7.2.3.2 Diseño del PID

En un mundo ideal, se podría tener como entradas los 4 grados de libertad queremos, throttle, pitch, roll y yaw y usando la matriz de comandos obtener la consigna a aplicar a cada motor.

Pero hay algunas cosas que hacen que sea imposible hacer esto en la vida real:

- Las perturbaciones tales como el viento y no ideales, como las diferencias en los motores y hélices, provocan que el comportamiento del Quadcopter de la vida real sea ruidoso y variable. La Matriz de comandos no toma esto en cuenta, y nuestra mente, los ojos y las manos podría no ser lo suficientemente rápido para reaccionar a estos cambios en tiempo real.
- Queremos que el Quadcopter tener un cierto grado de autonomía. En particular, sería bueno si el Quadcopter podría autonivelarse, volviendo a estar horizontal cuando mandamos pitch cero.

Para solucionar estos problemas se utiliza técnicas de control. En concreto, una estructura común para el control de realimentación se llama control PID (Proporcional - Integral - Derivativo).

Lo que hay que controlar en este caso es el ángulo (pitch, roll, yaw) del Quadcopter. Sin órdenes de entrada, tratamos de controlar el ángulo que es cero. Sin embargo, también podemos mandar un ángulo distinto de cero. Los comandos que enviamos a los motores se basan en el error entre el ángulo que queremos y el ángulo de hecho tenemos, según lo medido por la IMU.

Las componentes 3 que forma el control el PID son:

- Proporcional (P) : El comando es proporcional a la cantidad de error angular que tenemos. Ayuda a devolver el Quadcopter al ángulo deseado.
- Integral (I) : El comando es proporcional al error acumulado con el tiempo . Puede ayudar a combatir trastornos tales como el viento o el rendimiento del motor asimétrica
- Derivada (D) : El comando es proporcional a la tasa de cambio del error . Se resiste el movimiento y mantiene el ángulo de rebasamiento del objetivo.

Al no tener el modelo de la planta no se puede determinar empíricamente las constantes K_p y K_d , la mejor forma es calcularlas experimentalmente teniendo en cuenta que:

- El aumento de K_p empuja la Quadcopter hacia el ángulo deseado más rápido, pero también puede tener más sobreimpulso y oscilación.
- El aumento de K_d ralentiza la velocidad de rotación del Quadcopter, pero también puede amortiguar las oscilaciones.

Hay que tener en cuenta que para ciertos valores de las ganancias puede provocar el modelo sea vuelva inestable. Si K_p , K_d , o ambos son demasiado altos, el controlador inicia la amplificación de ruido, dando lugar a oscilaciones y la inestabilidad. Estas oscilaciones tienden a ser a una frecuencia más rápida que las oscilaciones que se verían desde un máximo K_p .

Se debe de tener en cuenta que la sintonización de ganancias requiere práctica y experiencia, y depende las preferencias del vuelo.

7.2.3.3 Controladores de pitch, roll y Yaw :

El Quadcopter tiene tres controladores de retroalimentación independientes, uno para cada giro: Roll, Pitch y Yaw. El Throttle se asigna directamente a los cuatro motores, aunque con un sensor de altitud, se podría agregar un cuarto controlador de retroalimentación con el throttle.

Los controladores de Roll y Pitch y Yaw son controladores PD que tiene la forma de la figura siguiente. Las salidas del control proporcional y del control derivativo se suman y se envían a los motores a través de la matriz de comandos.

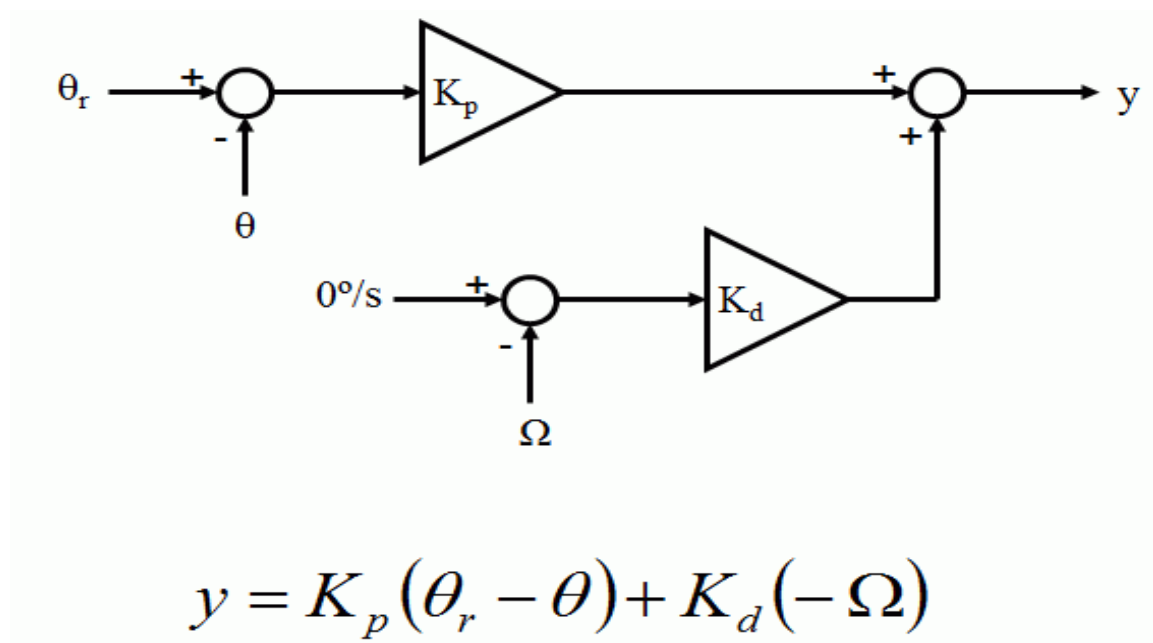


Figura 50: Controlador PID de un eje

Kp - Ganancia Proporcional
Kd - Ganancia derivativa
 θ_r - Referencia / Angulo Comando
 θ - Medida de ángulo de la IMU
 Ω - Medido de Rotación de la IMU .

7.2.3.4 Funciones

El modulo contiene una función que encarga de leer de los sensores y aplicar el control PID para cada uno de los ejes.

7.2.4 Modulo de Sensores

Este módulo se encarga de leer toda la información procesada por el Razor, se comunica con el mediante UART, usando el Serie 1 de Arduino.

7.2.4.1 Firmware de Razor 9 DOF

- Algoritmo de fusión de sensores

Internamente, la fusión de acelerómetro, magnetómetro y giroscopio de datos se realiza usando una matriz de coseno (DCM) algoritmo de Dirección. El algoritmo también se encarga del manejo de ruido del sensor y los errores numéricos. Por suerte el fabricante da un firmware que implementa este algoritmo y nos devuelve los 3 grados de Euler para orientar el Quadcopter. Aunque ha habido que modificarlo para nuestro propósito, para que devolviera además las velocidades angulares necesarias para el PID.

- Calibración de los sensores

Hay que calibrar el sensor, una vez instalado el firmware, para asegurar que la medidas que toma son las correctas. Los principales errores a compensar en los 3 sensores son:

-Error de escala: Por ejemplo, si el eje x acelerómetro mide 200 unidades, mientras que el eje y el acelerómetro mide 230 unidades con la misma fuerza aplicada. Los tres ejes por sensor deben ser consistentes.

-Traslados de origen: por ejemplo, si alguno de los ejes giroscopio informa algo diferente a cero cuando la tarjeta no está en movimiento.

El magnetómetro tiene algunas particularidades en lo que respecta a la calibración, ya que hay no sólo inexactitud sensor interno y el ruido, sino también las distorsiones de campo magnético externo. El buen desempeño del magnetómetro es más crucial para dar un rumbo correcto en todas las direcciones, por lo que si se calibra en un entorno distorsionado, siempre tendrá errores.

7.2.4.2 Funciones

El modulo contiene una función que encarga de procesar la información enviada por la UART, que tiene la siguiente forma:

Pitch	Roll	Yaw	Giroscopio X	Giroscopio Y	Giroscopio Z
-------	------	-----	--------------	--------------	--------------

Tabla 9: Información enviada por el Razor 9 DOF

7.2.5 Modulo de Cámara

Este módulo se encarga de procesar la información enviada por la cámara, se comunica mediante UART usando el Serial2 de Arduino.

7.2.5.1 Protocolo de comunicación

Para poder comunicarnos con la cámara hay saber la forma y secuencia a usar para recibir e enviar datos. Para ello hay que tener en mente los siguientes puntos:

- Formato de los paquetes

Para la comunicación con la cámara el fabricante especifica dos tipos de paquetes en función de su objetivo:

- Paquete de comandos

0XAA	ID Comando	Parámetro 1	Parámetro 2	Parámetro 3	Parámetro 4
------	------------	-------------	-------------	-------------	-------------

Tabla 10: Estructura de los paquetes de comandos cámara

- Paquete de datos



Tabla 11: Estructura de los paquetes de datos de la cámara

El ID es identificador de paquete y empieza en 0

El Data size es el tamaño de la imagen

El verify code es un checksum para asegurar que el paquete ha sido enviado correctamente.

Se calcula mediante la siguiente forma, donde N es el tamaño de paquete:

verify code = LOWBYTE(sum(byte[0] to byte[N-2]))

El tamaño por defectos de los paquetes es de 64 Bytes pudiendo llegar a ser hasta 512 bytes.

- Matriz de comandos

Esta es la matriz que define todos los posibles comandos a enviar a la cámara:

Command	ID Number	Parameter1	Parameter2	Parameter3	Parameter4
INITIAL	AA01h	00h	Colour Type	RAW Resolution (Still Image only)	JPEG Resolution
GET PICTURE	AA04h	Picture Type	00h	00h	00h
SNAPSHOT	AA05h	Snapshot Type	Skip Frame (Low Byte)	Skip Frame (High Byte)	00h
SET PACKAGE SIZE	AA06h	08h	Package Size (Low Byte)	Package Size (High Byte)	00h
Set Baud Rate	AA07h	1 st Divider	2 nd Divider	00h	00h
RESET	AA08h	Reset Type	00h	00h	XXh*
DATA	AA0Ah	Data Type	Length Byte 0	Length Byte 1	Length Byte 2
SYNC	AA0Dh	00h	00h	00h	00h
ACK	AA0Eh	Command ID	ACK Counter	00h / Package ID Byte 0	00h / Package ID Byte 1
NAK	AA0Fh	00h	NAK Counter	Error Number	00h
LIGHT	AA13h	Frequency Type	00h	00h	00h

Tabla 12: Lista de comandos de la cámara

- Conectar con la cámara

Para conectar con la cámara hay que enviar un comando especial llamado SYNC y dejar un esperar un tiempo fijo del orden de 100 ms, consultar si nos ha devuelto la cámara el ACK y la confirmación de envío. En tal caso estaremos conectados con la cámara.

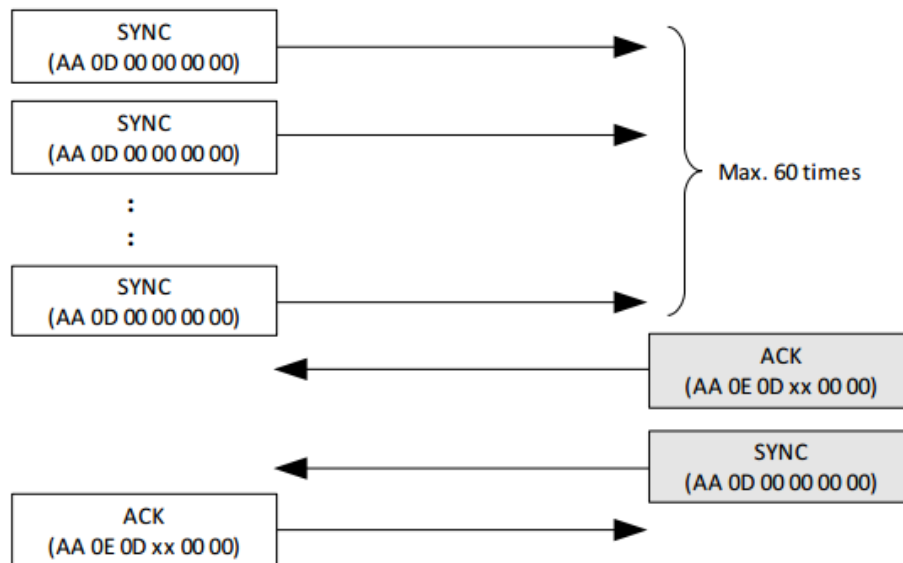


Figura 51: Conexión con la cámara

- Configuración de la cámara

Para la configuración de la cámara, se envía un paquete de comandos y si la cámara responde con ACK es que el cambio ha sido realizado.

Se puede configurar muchos parámetros, el tamaño de paquete, la resolución de la cámara, etc

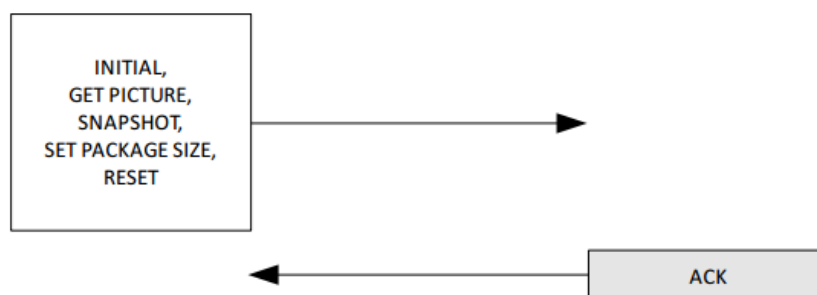


Figura 52: Configuración de la cámara

- Obtención de la imagen

Para la obtención de la imagen es un proceso algo más complicado. Primero se tiene que configurar la resolución a usar mediante un comando inicial mas tarde, se configura el tamaño de los paquetes con los que nos enviará la imagen, y usando el comando get picture vamos obteniendo cada uno de los paquetes de la imagen.

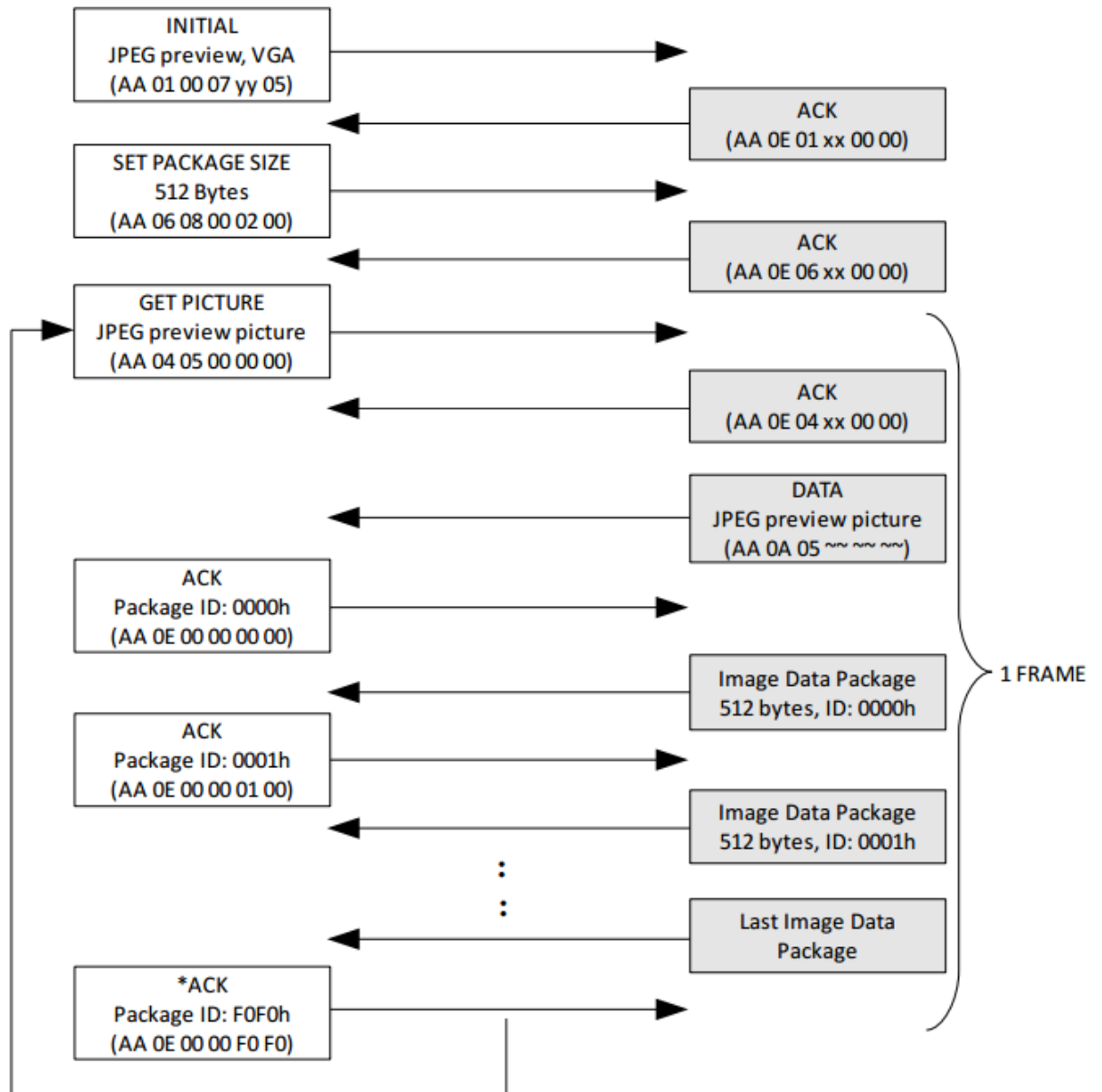


Figura 53: Obtención de la imagen

7.2.5.2 Funciones

Las principales funciones de modulo realizado son:

- La conexión con la cámara.
- Configuración de la cámara.
- La obtención de la imagen

En cuanto a la obtención de la imagen no se almacena en el Arduino sino que se usa un buffer del tamaño del paquete y se va enviando a la aplicación de forma secuencial a medida que se van obteniendo los paquetes.

7.2.6 Modulo de Leds

Este módulo se encarga de la monitorización de la batería, y la gestión de los Leds.

7.2.7 Modulo Principal

Este módulo se encarga de instanciar y usar el resto de los módulos.

Principalmente se encarga de la gestión de los comandos obtenidos por el Bluetooth y de llamar a las clases correspondiente para cada comando.

7.3 Aplicación para PC

Para el control del Quadcopter se desarrolló una pequeña aplicación en Java para PC. La principal ventaja de Java es que la aplicación puede ser utilizada en cualquier sistema operativo.

La idea del diseño de la interface era que fuera muy amigable y fácilmente usable.

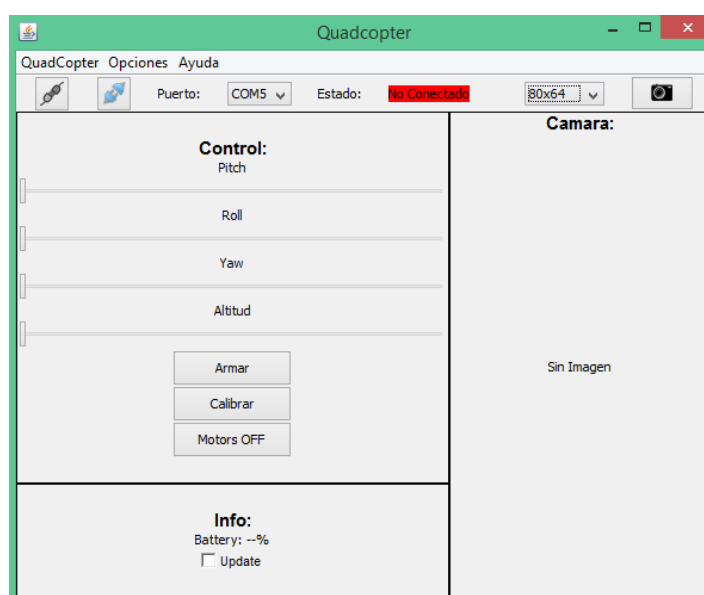


Figura 54: Aplicación de control

La aplicación permite saber el estado de la batería, capturar fotografías y los cuatro grados de libertad del Quadcopter, Pitch, roll y altitud.

7.3.1 Diagrama de Clases

El Programa está compuesto por 8 clases que se encargan de las distintas tareas de la aplicación:

- Main

Esta clase es el programa principal. Su única función es instanciar la clase Quadcopter.

- Quadcopter

Esta clase se extiende de un JFrame. Se encarga de instanciar a los demás clases y comunicarlás entre sí

- Vista

Es una clase que extiende de un JPanel y por defecto con un configurado un GridLayout. Es una una clase preparada para visualizar componentes, como Label, Buttons, etc.

- Vista Variables

Es una clase que extiende de Vista, se encarga de visualizar todas las variables adicionales como el nivel de batería del Quadcopter.

- Vista Cámara

Es una clase que extiende de Vista, se encarga de mostrar la imagen.

- Vista Mando

Es una clase que extiende de Vista, se encarga de capturar todos los comandos, desde la información de los grados de libertad, Yaw Pitch, Roll y Altitud, armar, calibrar y apagar los motores.

En la siguiente figura se puede ver la relación entre las clases.

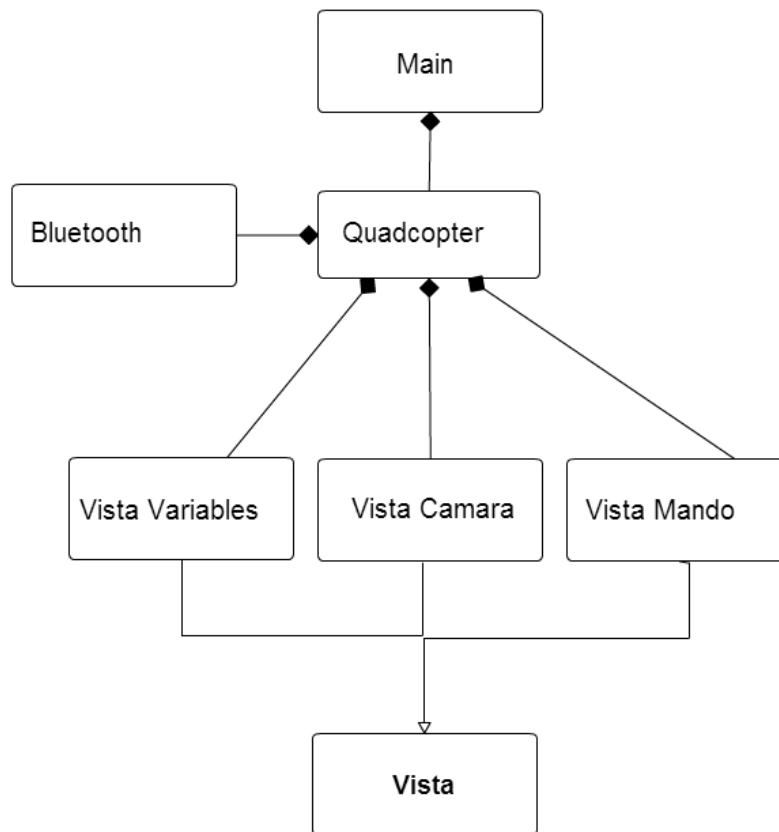


Figura 55: Diagrama de clases aplicación PC

8. Conclusiones

8.1 Resumen de Características

Las características generales del Quadcopter construido son las siguientes:

Medidas	50 cm x 50cm
Batería	3S (11.1V), 2200mAh, 25-50C Lipo
Motores	EMAX CF 2822 Brushless Outrunner 1200kv
ESC	EMAX 25A
Hélices	8'' x2 normales, 8'' x2 inversas
Microcontrolador	Arduino Due
IMU	Razor 9 DOF
Peso Total	1483g
Duración Máxima de vuelo	10 min

Tabla 13: Características generales del Quadcopter

El montaje definitivo tiene la siguiente forma:



Figura 56: Montaje Final Parte Superior

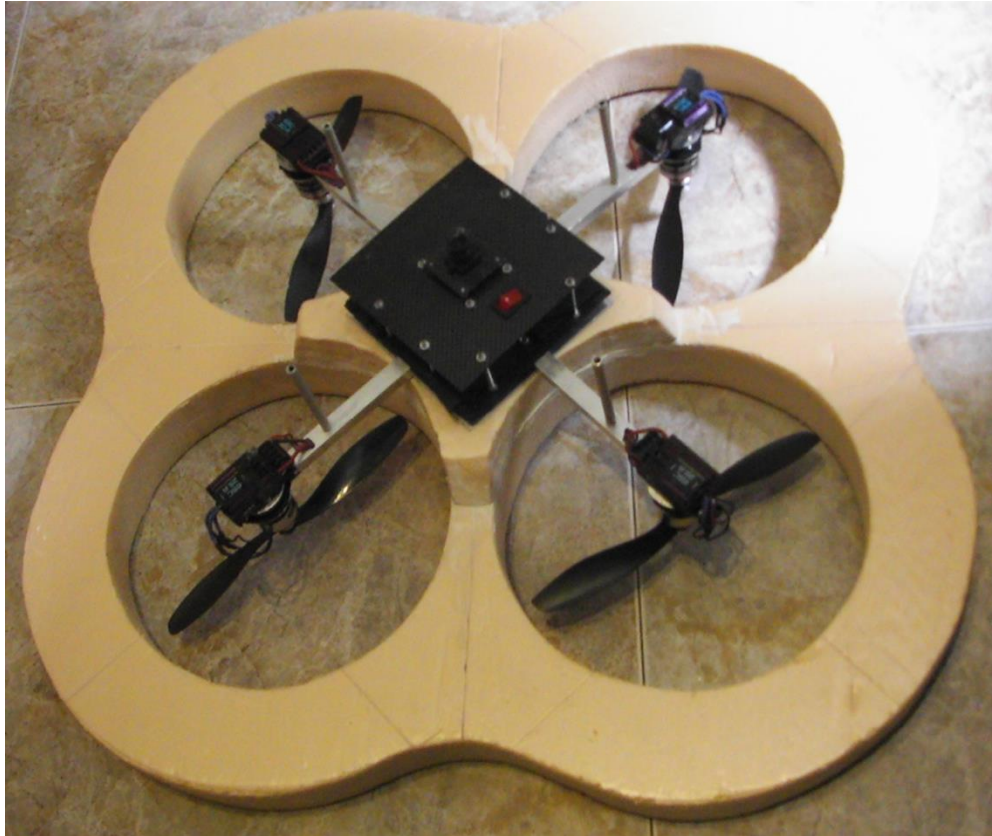


Figura 57: Montaje Final Parte inferior

8.2 Conclusión Final

Como conclusión final hay que destacar que el proyecto ha sido muy ambicioso y de una gran dificultad técnica, el objetivo principal era poder desarrollar un proyecto multidisciplinar donde intervinieran el diseño de hardware y software, como es el diseño y la construcción de un Quadcopter.

El valor de lo aprendido en este proyecto es incalculable, tener la experiencia de crear un sistema desde cero. Ya que nos ha servido para poner en práctica la capacidad para resolver casi cualquier tipo de problema. Siempre pensando la mejor solución en cuanto a calidad/precio y tiempo.

En cuanto a la planificación, gracias al método basado en intensivos, se ha conseguido superar los imprevistos que ha habido a la hora de construir la carcasa y por tanto respetar los plazos que inicialmente se habían calculado para cada una de las partes de proyecto.

El diseño de la estructura ha sido un éxito, se consiguió diseñar una estructura muy ligera y resistente, usando aluminio, fibra de carbono y Poliestireno extruido. La estructura posee casi la misma resistencia que el acero y es muy ligera. Además incorpora una carcasa para proteger el propio Quadcopter del entorno.

En cuanto al diseño de hardware ha sido un éxito, se conseguido integrar todos los componentes: microcontrolador, Cámara, Bluetooth, sensores y motores. El diseño está bastante bien conseguido para que en caso de avería de algunos de los componentes sea fácilmente reemplazable, además del uso de gomas Slient Block que ha permitido aislar completamente los sensores de la vibración producida por los motores.

En cuanto al diseño de software ha sido un éxito, se conseguido diseñar una librería para cada dispositivo, otorgando alta independencia entre clases. En cuanto al control hay que destacar los parámetros encontrados son muy experimentales y varían en función de la tipo de vuelo que se le quiera otorgar, y requieren de ciertos ajustes dependiendo del sitio donde se quiera volar. Lo ideal sería usar control adaptivo donde estos parámetros se ajustaran automáticamente, pero es un tema muy complejo y desde un principio se descartó por una cuestión de tiempo. La comunicación al Pc mediante el Bluetooth funciona perfectamente, y permite saber el estado de la batería, capturar fotografías y los cuatros grados de libertad del Quadcopter, Pitch, roll y altitud.

Como resumen, hemos conseguido realizar todos los objetivos marcados por el proyecto, desde la integración de la estructura mecanica con el hardware y diseño de software de control de Quadcopter permitiéndole volar prácticamente estable.

8.3 Futuro Trabajo

La construcción y el diseño ha sido satisfactorio, pero se han encontrado ciertos puntos a mejorar en futuro:

- Bluetooth vs Wi-Fi
Aunque el Bluetooth funciona perfectamente, tiene la limitación de su baudrate máximo que es 115200 bps provocando que el envío de archivos grandes pueda tardar varios minutos, si se cambiara el protocolo Bluetooth por el Wifi sería mucho más rápido los envíos de archivos, en este proyecto no se contempló por ser más complejo, pero en un futuro con más tiempo se podría implementar.
- Control Adaptativo
El control PID es un control muy básico que no tiene en cuenta la gran mayoría de no linealidades de la dinámica del Quadcoptero, teniendo que ajustar los parámetros para cada entorno. Con un Control Adaptativo, estos parámetros se cambiarían automáticamente evitando la constante reprogramación del Arduino.
- Arduino y Raspberry Pi
Aunque el Arduino Due tiene suficiente capacidad para gestionar las tareas asignadas en este proyecto, si se sigue trabajando en él y se implementa el control adaptativo, quizás nos encontraremos cortos de capacidad de cálculo. Una posible ampliación sería usar el Arduino actual para gestión de los motores y de la comunicación, y conectar directamente el Arduino a una Raspberry Pi, y que la Raspberry Pi haga todos los cálculos pesados. Además la Raspberry Pi dispone una cámara Full HD permitiendo grabar videos a 1980p a 30 fps. Con lo que ganaríamos mucha calidad en cuanto a la cámara.

9. Análisis del presupuesto

9.1 Consideraciones del presupuesto

Para poder ver específicamente el coste de las partes del proyecto, se desglosará el presupuesto en 3 partes:

- Recursos Humanos
- Hardware
- Software

9.2 Presupuesto de Recursos Humanos

Este proyecto será desarrollado por una persona. Como consecuencia esta persona tendrá que ejercer diferentes roles, los cuales son: Project Manager, Ingeniero industrial e Ingeniero informático.

En total se estimó que proyecto costaría de unas 600 horas entre los diferentes roles.

En la tabla siguiente se puede ver un desglose por roles del coste en euros que representa los recursos humanos.

Roles	Horas Estimadas	Coste por hora	Coste estimado
Project Manager	80 horas	50€/h	4.000 €
Ingeniero Industrial	200 horas	60€/h	12.000 €
Ingeniero Informático	320 horas	40€/h	12.800 €
Total	600 horas		28.800 €

Tabla 14: Presupuesto de recursos humano

9.3 Presupuesto de hardware

Para la propia construcción del Quadcopter se hará una lista de componentes necesarios. Además se tendrá en cuenta el hardware necesario para desarrollar el software.

La siguiente tabla se puede ver el coste en euros de todo el hardware necesario

Producto	Precio	Unidades	Vida útil	Total
Batería Lipo 11.1V 2200 mAh	14,9 €	2	4 meses	39,7 €
Arduino Due	47,12 €	1	2 año	7,85 €
Electronic Speed Controler Emax 25 A	18,19 €	4	2 año	3,03 €
Motor Brushless Outrunner CF2822	16,9 €	4	1 año	5,63 €
Modulo Bluetooth RN-BT-DATA-UG	53,16 €	1	2 años	8,86 €
Serial JPEG Cámara uCam	43,44 €	1	2 años	7,24 €
Razor 9-DOF	120,9 €	1	2 años	8,53 €
Protoboard Arduino	11 €	1	N/A	11 €

1 Kg de Plástico Pla	26 €	1	N/A	26 €
Impresora 3d Prusa Mendel	600 €	1	5 años	40 €
Varilla de aluminio	2 €	4	N/A	2 €
1 m ² de fibra de vidrio	5 €	1	N/A	5 €
1 m ² de poliéster expandido	55 €	1	N/A	55 €
Caja de tornillos	13 €	1	N/A	13 €
1 Kg Resina Epoxi + catalizador	12,15 €	1	N/A	12,15 €
Toshiba NB500	400 €	1	5 años	26,66 €
HTC Wildfire S	200 €	1	4 años	33,33 €
Hélice de 8" + Portahelices	8 €	4	1 año	2,6 €
Total	1646,76€			307,58 €

Tabla 15: Presupuesto Hardware

9.4 Presupuesto del software

Adicionalmente nos hace falta el software para desarrollar el proyecto. Algunos de ellos son gratuitos y otros no. Como en el presupuesto del hardware se tendrá en cuenta la amortización.

La tabla siguiente muestra el desglose del coste del software en euros.

Producto	Precio	Unidades	Vida útil	Total de la amortización
Arduino IDE	0 €	1	N/A	0 €
Java SDK	0 €	1	N/A	0 €
Eclipse	0 €	1	N/A	0 €
Microsoft Office 2010	9,99 €/mes	1	4 meses	9,99 €
Windows XP Professional	70 €	1	3 años	7,24 €
Adobe PDF	0 €	1	N/A	0 €
Subversion	0 €	1	N/A	0 €
Cacoo	0 €	1	N/A	0 €
Total	109,96 €			47,2 €

Tabla 16: Presupuesto Hardware

9.5 Presupuesto Total

Juntando todos los presupuestos anteriores, obtenemos el coste económico que representa el proyecto como podemos ver en la tabla siguiente:

Concepto	Coste estimado
Presupuesto de recursos humanos	28.800
Presupuesto de hardware	307,58
Presupuesto de software	47,2
Total	29.155,78 €

Tabla 17: Presupuesto Total

10. Contexto del proyecto

10.1 Impacto social y medio ambiental

La batería Lipo que se usa, si se daña o sobrecarga puede provocar un incendio, pero para evitar esto está ubicado en una parte protegida y controlada del Quadcopter y se recarga mediante un cargador especial que evita las sobrecargas.

10.2 Control y Sostenibilidad

Una de las formas de poder ver el impacto del proyecto si se empezará a fabricar en serie Quadcopter sería analizar a los principales proveedores de las materias primas como Baterías Lipo, etc.

En cuanto a la sostenibilidad de la producción mientras la demanda de materias primas como baterías Lipo, etc , puede ser cubierta por los principales proveedores, sostenibilidad de la producción está garantizada.

En cuanto sostenibilidad de la vida de Quadcopter, tiene incorporado un medidor de tensión de la batería que nos indica cuando esta deja de ser útil para ser cambia y aterriza, cosa que nos asegura poder conservar el Quadcopter a lo largo del tiempo.

10.3 Viabilidad económica

El proyecto supone un coste de 29.155,78 euros es un coste bastante bajo para el tipo de proyecto que es, dado que a la hora de la verdad los costes de recursos humanos no se pagan porque el autor de este TFG quiere presentarlo como proyecto final de grado, solo se tenía en cuenta los costes de hardware y software, en total unos 307,58 euros que es una cantidad que autor está dispuesto a pagar, por lo que el proyecto es viable económicamente tanto como TFG como proyecto real.

11. Referencias

[1] UAV.

<http://www.theuav.com/> - 14/03/2014

[2] MEMS

<https://www.mems-exchange.org/MEMS/what-is.html> - 14/06/2014

[3] Arduino

<http://www.arduino.cc/> - 14/06/2014

[4] Ar Drone Parrot

<http://ardrone2.parrot.com/> - 14/06/2014

[5] Parrot. Company

<http://www.parrot.com/> - 14/06/2014

[6] PCB Quadcopter

<http://www.instructables.com/id/PCB-Quadrotor-Brushless/> - 14/06/2014

[7] Wiring

<http://wiring.org.co/> - 14/06/2014

[8] Processing

<http://www.processing.org> - 14/06/2014

12. Bibliografía

1. 4D Systems, *Datasheet Cámara Jpeg TTL UCam*

Disponible: <http://www.robot-electronics.co.uk/datasheets/uCAM-DS-rev4.pdf>

2. Mikroe, *Datasheet Bluetooth Stick RN-41*

Disponible: http://www.mikroe.com/downloads/get/1617/bluetooth_stick_manual.pdf

3. Peter Bartz, *Tutorial Razor 9 DOF*

Disponible: <https://github.com/ptrbrtz/razor-9dof-ahrs>

4. InveSense, *Datasheet Giroscopio ITG-3200*

Disponible: <https://www.sparkfun.com/datasheets/Sensors/Gyro/PS-ITG-3200-00-01.4.pdf>

5. Analog Sensors, *Datasheet Acelerómetro ADXL 345*

Disponible: <https://www.sparkfun.com/datasheets/Sensors/Accelerometer/ADXL345.pdf>

6. HoneyWell, *DataSheet Magnometro HMC 5883L*

Disponible: <http://dlnmh9ip6v2uc.cloudfront.net/datasheets/Sensors/Magneto/HMC5883L-FDS.pdf>

7. Arduteka, *Tutorial como usar Bluetooth en Arduino.*

Disponible: <http://www.arduteka.com/arduino/>

8. B. Beauregard, *Arduino PID-Guis de uso librería PID.*

Disponible: <http://brettbeauregard.com/blog/wpcontent/uploads/2012/07/Gu%C3%ADa-de-uso-PID-para-Arduino.pdf>

9. M. Margolis, *Arduino Cookbook*, United States of America: O'Reilly, 2011.

10. K. Shirriff y P. Badger, *Secrets of Arduino PWM*

Disponible: <http://arduino.cc/en/Tutorial/SecretsOfArduinoPWM>