



UNIVERSIDAD DE JAÉN

Escuela Politécnica Superior de Jaén

Trabajo Fin de Grado

DISEÑO Y CONSTRUCCIÓN DE UN ROBOT MÓVIL COMO UN AGENTE AUTÓNOMO

Alumno: Adrián Luque Luque

Tutor: Alberto Fernández Hilario
Dpto: Departamento de Informática



Universidad de Jaén
Escuela Politécnica Superior de Jaén
Departamento de Informática

Don Alberto Fernández Hilario, tutor del Trabajo Fin de Grado titulado: Diseño y construcción de un robot móvil como agente autónomo, que presenta Adrián Luque Luque, autoriza su presentación para defensa y evaluación en la Escuela Politécnica Superior de Jaén.

Jaén, 15 Junio de 2015

El alumno:

El tutor:

Adrián Luque Luque

Alberto Fernández Hilario

AGRADECIMIENTOS

Este documento ha sido el fruto de un gran esfuerzo desarrollado durante varios meses y del que estoy realmente orgulloso por la solución alcanzada.

Por ello, me gustaría dar las gracias a mis compañeros, profesores y familia que me han apoyado durante estos años en la universidad. En especial a mi tutor Alberto por su ayuda.

Me gustaría también agradecer a mi pareja por soportarme y apoyarme durante los días de trabajo

GRACIAS A TODOS

Índice

Índice de ilustraciones.....	6
Índice de tablas	7
1. INTRODUCCIÓN	8
1.1. Motivación	9
1.2. Objetivos	10
1.3. Estructura y planificación del proyecto.....	12
1.3.1. División de tareas	12
1.3.2. Calendario	14
1.3.3. Diagrama de Gantt	14
1.4. Coste del proyecto	17
1.4.1. Recursos humanos	17
1.4.2. Hardware y software.....	18
1.5. Estructura de la memoria.....	20
2. ESTUDIO DE SOLUCIONES ACTUALES.....	22
2.1. Lily	23
2.2. Hexo+	25
3. ANÁLISIS	28
3.1. Requisitos funcionales.....	28
3.2. Requisitos no funcionales	29
3.3. Posibles escenarios	30
4. DISEÑO	31
4.1. Componentes del drone.....	31
4.1.1. Controladora de vuelo.....	31
4.1.2. Motores	33
4.1.3. Controladores electrónicos de velocidad (ESC)	34
4.1.4. Hélices	34
4.1.5. Frame	35
4.1.6. Universal battery eliminator circuit (UBEC)	36
4.1.7. Emisora de radio y receptor	36
4.1.8. Batería	37
4.1.9. Cables, adaptadores y otros.....	39
4.2. Componentes del sistema de seguimiento	41
4.2.1. Diseño del seguimiento.....	41

4.2.2.	Microprocesadores.....	45
4.2.3.	Cámaras.....	48
4.2.4.	Powerbank	49
4.3.	Corrección del coste del proyecto.....	51
4.4.	Montaje del drone	52
4.5.	Comunicación hardware-software.....	61
4.6.	Plataformas de desarrollo	65
4.6.1.	Lenguajes de programación	65
4.6.2.	Entornos de desarrollo integrado	66
4.7.	Software de seguimiento	68
4.7.1.	Intel Edison.....	68
4.7.2.	Arduino Nano	71
5.	INTEGRACIÓN Y PRUEBAS	73
5.1.	Integración hardware y software	73
5.2.	Pruebas.....	73
6.	CONCLUSIONES Y TRABAJO FUTURO	76
6.1.	Futuras mejoras para el prototipo	76
6.2.	Creación de una empresa.....	76
	Descripción de los anexos	78
	Bibliografía	79

Índice de ilustraciones

<i>Ilustración 1.1 - División de tareas</i>	<i>13</i>
<i>Ilustración 1.2 - Diagrama de Gantt.....</i>	<i>16</i>
<i>Ilustración 2.1 - DJI Naza v2</i>	<i>22</i>
<i>Ilustración 2.2 - Lily.....</i>	<i>23</i>
<i>Ilustración 2.3 - Baliza Lily.....</i>	<i>24</i>
<i>Ilustración 2.4 - Hexo+.....</i>	<i>25</i>
<i>Ilustración 2.5 - Gimbal H3 3D de DJI.....</i>	<i>26</i>
<i>Ilustración 4.1 - HobbyKing kk2.1 HC</i>	<i>32</i>
<i>Ilustración 4.2 - Motor 2212.....</i>	<i>33</i>
<i>Ilustración 4.3 - ESC Opto-15A</i>	<i>34</i>
<i>Ilustración 4.4 - Hélices DJI.....</i>	<i>35</i>
<i>Ilustración 4.5 - Frame F450 DJI.....</i>	<i>35</i>
<i>Ilustración 4.6 - UBEC 5A HobbyKing</i>	<i>36</i>
<i>Ilustración 4.7 - Radio y receptor Turnigy 9X</i>	<i>37</i>
<i>Ilustración 4.8 - Batería 3S Turnigy</i>	<i>38</i>
<i>Ilustración 4.9 - Avisador de batería baja</i>	<i>40</i>
<i>Ilustración 4.10 - Diagrama eléctrico baliza.....</i>	<i>43</i>
<i>Ilustración 4.11 - Baliza (interior).....</i>	<i>44</i>
<i>Ilustración 4.12 - Baliza</i>	<i>44</i>
<i>Ilustración 4.13 - Baliza encendida.....</i>	<i>45</i>
<i>Ilustración 4.14 - Intel Edison</i>	<i>46</i>
<i>Ilustración 4.15 - Arduino Nano</i>	<i>48</i>
<i>Ilustración 4.16 - Webcam C170 Logitech.....</i>	<i>49</i>
<i>Ilustración 4.17 - Powerbank.....</i>	<i>50</i>
<i>Ilustración 4.18 - Esquema eléctrico drone</i>	<i>53</i>
<i>Ilustración 4.19 - Esquema eléctrico sistema de seguimiento</i>	<i>54</i>
<i>Ilustración 4.20 - Montaje chasis inferior.....</i>	<i>55</i>
<i>Ilustración 4.21 - Montaje ESC</i>	<i>55</i>
<i>Ilustración 4.22 - Placa distribución energía</i>	<i>56</i>
<i>Ilustración 4.23 - Detalle conexión motor y ESC.....</i>	<i>57</i>
<i>Ilustración 4.24 - Chasis detalle microcontroladores</i>	<i>58</i>
<i>Ilustración 4.25 - Chasis detalle powerbank.....</i>	<i>58</i>
<i>Ilustración 4.26 - Chasis detalle cámaras.....</i>	<i>59</i>
<i>Ilustración 4.27 - Distribución quadrópteros</i>	<i>59</i>
<i>Ilustración 4.28 - Esquema conexión electrónica</i>	<i>62</i>
<i>Ilustración 4.29 - Detalle de movimientos del drone.....</i>	<i>64</i>
<i>Ilustración 4.30 - IDE Arduino.....</i>	<i>67</i>
<i>Ilustración 4.31 - PyCharm IDE Arduino</i>	<i>68</i>
<i>Ilustración 4.32 - Umbral cámaras.....</i>	<i>70</i>

Índice de tablas

<i>Tabla 1.1 - Calendario</i>	<i>14</i>
<i>Tabla 1.2 - Horas de trabajo.....</i>	<i>18</i>
<i>Tabla 1.3 - Coste recursos humanos.....</i>	<i>18</i>
<i>Tabla 1.4 - Coste estimado drone.....</i>	<i>18</i>
<i>Tabla 1.5 - Coste estimado sistema seguimiento.....</i>	<i>19</i>
<i>Tabla 1.6 - Coste estimado hardware</i>	<i>19</i>
<i>Tabla 1.7 - Coste Software</i>	<i>19</i>
<i>Tabla 1.8 - Coste estimado final.....</i>	<i>20</i>
<i>Tabla 4.1 - Coste drone</i>	<i>51</i>
<i>Tabla 4.2 - Coste baliza</i>	<i>51</i>
<i>Tabla 4.3 - Coste sistema de seguimiento.....</i>	<i>51</i>
<i>Tabla 4.4 - Coste final del proyecto.....</i>	<i>52</i>

1. INTRODUCCIÓN

En los últimos años ha crecido el entusiasmo por los llamados drones [1], vehículos aéreos no tripulados, que aunque se asocian al ámbito militar, es todo aquel vehículo capaz de volar sin necesidad de tripulación dentro de este, desde drones de ataque militares [2] a los que intenta poner en marcha Amazon para sus envíos [3]. En este proyecto nos centraremos en los multicópteros o multirrotores, más específicamente en los quadrópteros [4], helicópteros propulsados por cuatro rotores de empuje vertical.

El abaratamiento de los costes de producción y su popularización, ha permitido, que estos pasen del ámbito militar y de investigación, al ámbito del ocio, y la creación de empresas especializadas en el desarrollo y la venta de drones. También han aumentado los casos de particulares que empiezan proyectos propios, ya sea para la mejora de modelos existentes o para la construcción de uno propio usando componentes comprados por separado.

Existen multitud de aplicaciones en las que el uso de un drone ha significado una gran ventaja con respecto a las soluciones tradicionales, entre otras, destacamos la ayuda en desastres naturales, rescate en montañas, acceso a zonas peligrosas para los humanos, etc. Aunque ya se usan para la realización de grabaciones en zonas de difícil acceso o de gran altura, grabaciones en las que se necesita seguir al objetivo de cerca, como puede ser una exhibición de saltos en moto o similares, tareas de seguridad y vigilancia, y un largo etcétera.

Hasta ahora, si se necesitaba la realización de alguno de los anteriores casos, significaba realizar el despliegue de un vehículo aéreo, helicóptero o avioneta, de un piloto y de uno o varios operarios de cámaras. Eso en el caso de que ese vehículo sea capaz de entrar en la zona.

Dadas las características de estos aparatos, a saber, velocidad, maniobrabilidad, agilidad, tamaño y peso reducido, y capacidad de vuelo en espacios reducidos, se ha empezado a investigar su uso en diversas aplicaciones.

Todos estos hechos resaltan la importancia de los drones en el panorama actual. De ahí el interés y necesidad de potenciar su uso, incluso para temas didácticos.

1.1. Motivación

Este proyecto pretende la construcción e implementación desde cero de un drone para uso particular. El fin último será dotarlo de autonomía de acuerdo al seguimiento de objetos móviles, que por facilidad en la implementación estarán marcados con una baliza. Para ser capaces de llevar a cabo la tarea de construcción un drone totalmente personalizado, será necesario un profundo conocimiento de las partes que los componen, incluyendo sistemas de tipo eléctricos, electrónico y por supuesto mecánico.

Gracias a que no nos ceñimos a un diseño prefabricado, tenemos la ventaja de una libre elección de componentes. Por tanto, de acuerdo a los requisitos que nos marquemos, contaremos con la posibilidad de seleccionar aquéllos que mejor se adecúen a las necesidades y objetivos dispuestos a lo largo del proyecto.

Dada la falta de productos similares en el mercado y el gran potencial que tienen estas soluciones, se estudiará la posibilidad de aprovechar todo el tiempo dedicado en este proyecto para el desarrollo de nuevos drones y su posterior comercialización.

Cabe destacar también el reto que supone un desarrollo de estas características y el interés del alumno por los multirrotores de radio control y el aprendizaje que obtendrá durante la construcción del prototipo.

1.2. Objetivos

A lo largo del desarrollo del presente TFG, los objetivos generales a alcanzar serán los siguientes:

- Estudiar las posibilidades del mercado en función de los requerimientos del robot. Ello dependerá del tipo de entorno en el que finalmente se decida trabajar, a saber, que en este caso será el medio aéreo.
- Realizar todo el trabajo "electrónico" de montaje y ensamblado de las piezas necesarias para construir el agente móvil.
- Analizar los diferentes microcontroladores y sus características para integrar aquél que mejor se adapte a los requerimientos definidos en la fase de diseño.
- Implementar mecanismos de inteligencia artificial sobre el microcontrolador. De este modo, se pretende guiar al robot en un entorno a priori no acotado.

Centrándonos en el proyecto de construcción de un drone volador autónomo, para el desarrollo de un prototipo de estas características definimos objetivos más específicos:

- Análisis de los modelos de quadrópteros existentes y de las características que ofrecen. Estudiar las necesidades que cubren de las planteadas en el proyecto.
- Diseño del esquema de construcción. Piezas necesarias, microcontroladores más apropiados, interconexión entre componentes, cámaras, etc.
- Construcción del quadróptero realizando la unión de las piezas, conexión electrónica y eléctrica de circuitos y rotores, y acoplamiento de los microcontroladores.
- Implementación del software y de la comunicación hardware-software para el vuelo mediante radiocontrol.
- Desarrollo del software y protocolo de comunicación hardware-software para el vuelo autónomo mediante baliza.

- Estudio de la viabilidad para la futura comercialización del esquema de diseño y montaje del dron.

No queremos cerrar esta sección sin reiterar el **gran reto** asociado a la realización del presente trabajo. Debemos tomar en consideración el alto nivel de ambición que se persigue de acuerdo a los objetivos anteriormente planteados, que se plasman en un esfuerzo significativo a diferentes niveles:

- En primer lugar, a un nivel mecánico es necesario diseñar y construir un prototipo de dron mediante piezas básicas. No radica solo en unir los componentes, si no ubicarlos correctamente sobre una base sólida que permita dotar al mismo de estabilidad y capacidad de vuelo.
- En segundo lugar, a un nivel electrónico, en lo relativo a realizar todas las conexiones cableadas entre los diferentes componentes, consiguiendo una correcta comunicación entre los mismos.
- Por último, y no menos importante, la aplicación de conocimientos software tanto a alto como bajo nivel de programación. En concreto, en este trabajo partiremos del código implantada en la controladora de vuelo seleccionada a tal efecto. Sin embargo, a partir de este algoritmo de vuelo base sobre el que apoyarnos, iremos diseñando e implementando un modo de vuelo de acuerdo a las características particulares del dron y de las necesidades de vuelo objetivo. Especialmente significativo será el programa asociado a la autonomía del dron en el seguimiento de la baliza supone un amplio desafío en el sentido que implica la captura de datos de diferentes sensores y la corrección del vuelo a través de una cuidadosa optimización de los rotores.

1.3. Estructura y planificación del proyecto

En este apartado se expone, en primer lugar, la división del proyecto en tareas más simples para facilitar su realización. A continuación, mostraremos la división del tiempo disponible entre estas tareas. Por último, se introducirá un diagrama de Gantt [5] que facilitará la comprensión de estas divisiones y mostrará detalles como la posibilidad de hacer en paralelo ciertas tareas.

1.3.1. División de tareas

Como se puede ver en la ilustración 1.1, se han creado las tareas típicas de un proyecto, como son análisis, diseño, puesta en funcionamiento y pruebas, además del estudio de soluciones similares y la redacción de la memoria.

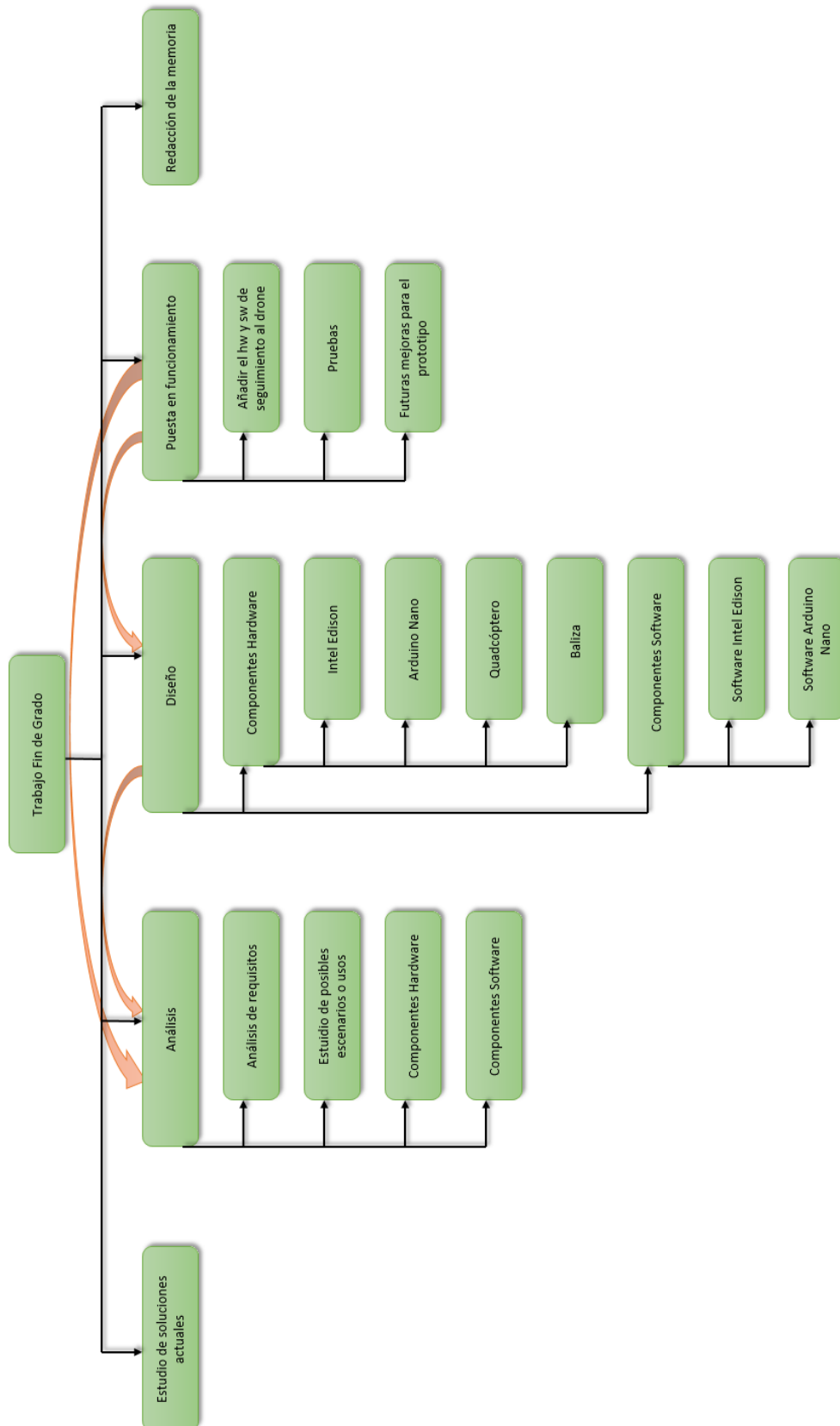


Ilustración 1.1 - División de tareas

1.3.2. Calendario

El tiempo especificado para la realización del trabajo de fin de grado es de 295 horas, según la guía docente, a razón de 2.5 horas diarias y 5 días laborables por semana.

El calendario propuesto, tabla 1.1, ha sido elaborado teniendo en cuenta conocimientos previos del alumno, en áreas que se desarrollan en este proyecto, obtenidos en asignaturas como procesamiento de la información visual o programación hardware y en proyectos propios.

Nombre de tarea	Duración
Estudio de soluciones actuales	10 días
Análisis	20 días
Análisis de requisitos	4 días
Estudio de posibles escenarios o usos	2 día
Componentes Hardware	6 días
Componentes Software	8 días
Diseño	40 días
Componentes Hardware	22 días
Intel Edison	6 días
Arduino Nano	4 días
Quadróptero	10 días
Baliza	2 día
Componentes Software	18 días
Software Intel Edison	14 días
Software Arduino Nano	4 días
Puesta en funcionamiento	16 días
Añadir el hw y sw de seguimiento al drone	4 días
Pruebas	8 días
Futuras mejoras para el prototipo	4 días
Redacción de la memoria	32 días

Tabla 1.1 - Calendario

1.3.3. Diagrama de Gantt

Por último, se ha creado un diagrama de Gantt [5], una gráfica de barras con el tiempo en el eje horizontal y en el eje vertical las tareas a realizar, a pesar de su sencillez ayuda en la visualización de la planificación temporal.

Las tareas “Componentes Hardware” y “Componentes Software”, dentro de la tarea “Diseño”, han sido divididas en tareas que pueden realizarse en paralelo. La duración de estas tareas padre no ha sido redimensionada a la mayor de las tareas en las que se descompone, porque aunque no es necesaria la realización de una tarea hija antes de empezar la siguiente, el hecho de que solo una persona esté realizando el proyecto impide que se puedan realizar a la vez.

La memoria, como se puede ver en el diagrama de Gantt del proyecto,

Ilustración 1.2, se ha redactado conforme avanzaba el proyecto.

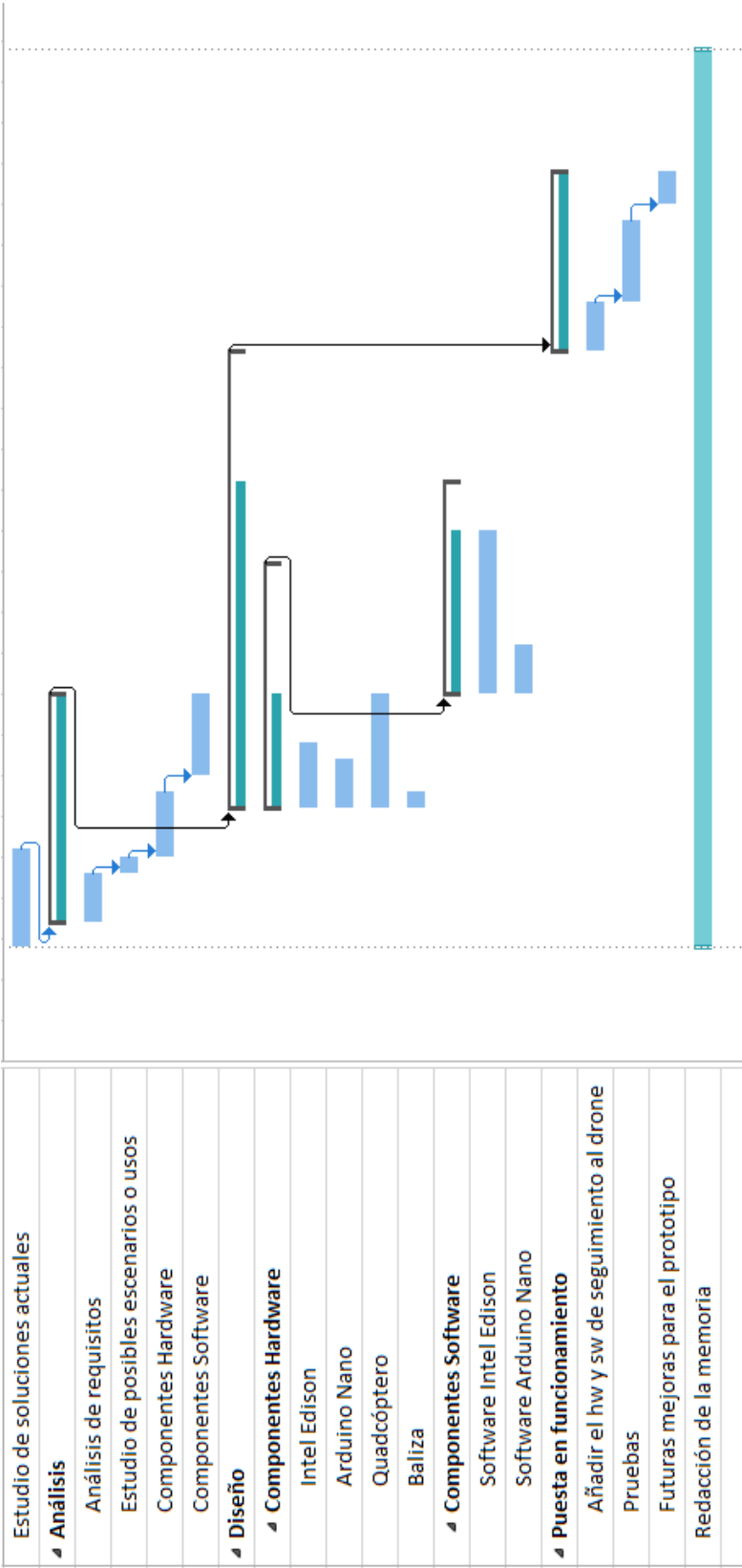


Ilustración 1.2 - Diagrama de Gantt

1.4. Coste del proyecto

En este apartado descompondremos todo el coste de proyecto, separando costes en contratación de personal y en hardware y software necesarios para el desarrollo del prototipo.

1.4.1. Recursos humanos

Para el cálculo del coste de los trabajadores, hemos separado el trabajo del alumno, como si tuviera roles distintos para calcular los costes reales que supondría para una empresa. Como es normal en el desarrollo de proyectos informáticos varios roles trabajarán juntos en las distintas fases.

- Jefe de proyecto, se encargará de la investigación previa al proyecto, del estudio para la creación de una empresa derivada de este proyecto y de la redacción de la memoria, además de participar en la fase de análisis.
- Analista Programador, trabajará en la fase de análisis, en la fase de diseño como programador, ayudando a unir el software de reconocimiento con el drone y a la búsqueda de futuras mejoras.
- Técnico, tendrá la responsabilidad del montaje y puesta a punto del drone, y colaborará con el analista en la puesta en funcionamiento del drone.
- Tester, será una persona lo más similar posible a un futuro usuario del sistema, al que se le impartirá una pequeña lección del funcionamiento y uso del prototipo, para que pueda llevar a cabo las pruebas finales.

La división de horas por tarea y rol se puede ver en la tabla 1.1, como se ha explicado antes, hay ciertos momentos en que dos trabajadores deben cooperar, por lo que la suma de tiempo de cada rol en una tarea concreta no tiene por qué ser igual al tiempo estimado de la tarea.

Tareas Principales	Estudios	Análisis	Diseño	Puesta en marcha	Memoria	Total
Jefe de proyecto	25	35	0	0	80	140
Analista Programador	0	35	75	15	0	125
Técnico	0	0	35	15	0	50
Tester	0	0	0	20	0	20

Tabla 1.2 - Horas de trabajo

Para poder asignar un sueldo todo lo realista posible se ha hecho una búsqueda de ofertas de trabajo para puestos similares a los descritos anteriormente, los resultados, en sueldo neto por hora y sueldo total, son los mostrados en la tabla 1.2.

Rol	Jefe de proyecto	Analista Programador	Técnico	Tester	Totales
Días de trabajo	56	50	20	8	134
Horas de trabajo	140	125	50	20	335
Sueldo por hora	30	22	15	10	77
Sueldo total	4.200,00 €	2.750,00 €	750,00 €	200,00 €	7.900,00 €

Tabla 1.3 - Coste recursos humanos

1.4.2. Hardware y software

Aquí desglosaremos todo el coste estimado del hardware y del software. Para facilitar la lectura de los datos, se han creado cinco tablas:

- Coste estimado del drone, coste de cada pieza que compone el drone, el coste individual no incluye gastos asociados al envío, estos están especificados al final, tabla 1.4.

Partes del Drone	Coste
Controladora de vuelo	20 - 50 €
Control electrónico de velocidad + Motores + Palas	120 - 130 €
Frame	50 - 80 €
Mando radio control (Transmisor + Receptor)	70 - 80 €
Batería y cargador	55 - 75 €
Cables, adaptadores y otros	15 €
Envío	20 - 25 €
Total	350 - 460 €

Tabla 1.4 - Coste estimado drone

- Coste estimado del sistema de seguimiento, solo se detallan los componentes hardware, el software se muestra más adelante, tabla 1.5.

Sistema de seguimiento	Coste
Microcontroladores	30 - 100 €
Cámaras	35 €
Baliza	3 – 15 €
Batería 5Voltios	7 - 10 €
Envío	10 €
Total	85 - 170 €

Tabla 1.5 - Coste estimado sistema seguimiento

- Coste estimado del hardware, suma de los totales anteriores y del ordenador usado para el desarrollo del proyecto, tabla 1.6. Para el coste del pc se ha tenido en cuenta el tiempo normal que dura un ordenador portátil, unos cuatro años, la duración del proyecto y el coste inicial, 765€.

Hardware	Coste
PC	42,19 €
Drone	350 - 460 €
Sistema de seguimiento	85 - 170 €
Total	477,19 - 672,19 €

Tabla 1.6 - Coste estimado hardware

- Coste total del software, tabla 1.7.

Software	Coste
Windows 8.1	0,00 €
IDE Python (PyCharm)	0,00 €
IDE Arduino	0,00 €
Open Office	0,00 €
Total	0,00 €

Tabla 1.7 - Coste Software

Durante el cálculo del coste del PC usado para el desarrollo del proyecto se ha tenido en cuenta el tiempo normal que dura un ordenador portátil que son unos cuatro años, la duración del proyecto tres meses y el coste inicial 765€.

Como podemos ver el coste de los envíos de las partes drones es alto, esto se debe a que pocas piezas se pueden comprar en España, en concreto solo el frame, cargador y batería se han comprado en tiendas españolas, el resto ha sido en tiendas internacionales.

También se puede apreciar que todo el software ha sido gratuito, gracias a las licencias de Windows que la Universidad de Jaén proporciona a los alumnos y al uso de programas, que aunque son muy completos, son gratuitos.

Ahora que hemos terminado de presentar cada gasto del proyecto, podemos juntar todos los costes en una sola tabla, tabla 1.8.

Coste final	Coste
Recursos Humanos	7.900,00 €
Hardware	477,19 - 672,19 €
Software	0,00 €
Total	8377,19 - 8572,19 €

Tabla 1.8 - Coste estimado final

A este presupuesto hay que añadir los diferentes gastos de contratar a los trabajadores como la seguridad social, y el alquiler de los espacios necesarios para el proyecto.

Aunque en este apartado hemos enumerado diferentes partes del drone o la baliza, en las secciones correspondientes se explicará en qué consisten y su funcionamiento dentro el sistema.

1.5. Estructura de la memoria

A continuación se muestra un resumen de los siguientes capítulos de la memoria:

En el capítulo 2, realizaremos un estudio de soluciones actuales, analizaremos dos proyectos similares, drones para uso personal capaces de movimiento autónomo.

A continuación, el capítulo 3 se centra en el análisis de la propuesta, definiremos cuales son los requisitos necesarios que tiene que cumplir el

prototipo, buscaremos posibles escenarios de uso, y por último diferenciaremos que requisitos necesitamos en la parte hardware y software del proyecto.

Después del análisis viene el diseño, en el capítulo 4, aquí buscamos, seleccionamos y montamos el hardware, y creamos el software que controlará todo este hardware, siempre teniendo en cuenta las limitaciones impuestas por el análisis.

En el capítulo 5, probaremos en funcionamiento el drone, su puesta en marcha comprende la unión del drone con el dispositivo de seguimiento, hasta ahora solo se habían probado por separado, pruebas de funcionamiento del prototipo y una búsqueda de mejoras para el prototipo.

En el capítulo 6, aparecen las conclusiones, objetivos alcanzados, consideraciones personales, etc.

Por último se incluye la bibliografía y referencias de todos los recursos usados, y una descripción de los anexos, aquí se enumerarán todos los anexos, videos, fotos, esquemas, etc., que se adjuntarán junto a la memoria.

2. ESTUDIO DE SOLUCIONES ACTUALES

Según lo expuesto en los objetivos de este proyecto, el fin último es de obtener un drone quadróptero que permita el seguimiento de objetos móviles marcados con una baliza. Realizando un estudio de mercado, podemos encontrar diversos productos que permite un vuelo autónomo, pero no existe actualmente ninguno que sea lo suficientemente completo o versátil, que logre alcanzar la meta que nos hemos propuesto.

Entre las diferentes opciones comerciales, podemos encontrar controladoras de vuelo con gps y alguno de los siguientes modos, Return-To-Home o vuelo por Waypoints, un ejemplo de estas controladoras son las ofrecidas por DJI [6], Ilustración 2.1.



Ilustración 2.1 - DJI Naza v2

Return-To-Home, es un modo que permite a un drone que está fuera de la línea de visión del piloto, o que haya perdido la señal del transmisor de radio, volver al punto desde el que empezó el vuelo, funciona gracias a un receptor gps, aunque es un buen método para prevenir accidentes cuando se pierde el control por distancia, no es capaz de evitar posibles obstáculos en su vuelta a casa.

En el modo “Vuelo por Waypoints”, antes del vuelo el piloto debe asignar *waypoints* o puntos de control, que una vez en funcionamiento el dron seguirá. El piloto debe asegurarse de que los puntos de control están bien situados y a suficiente altura para evitar tocar árboles, casas, y distintos obstáculos que se eleven del terreno.

De acuerdo a la descripción de estos productos y sus modos de funcionamiento, se observa que ninguno llega a permitir un vuelo que se pueda considerar inteligente.

Sin embargo, se están realizando diferentes desarrollos comerciales para permitir el seguimiento con balizas. En concreto, nos vamos a centrar en las dos opciones más conocidas como son Lily (Sección 2.1) y Hexo+ (Sección 2.2).

2.1. Lily

Se trata de un quadróptero en desarrollo [7], Ilustración 2.2, capaz de seguir una baliza, ilustración 2.3, sin necesidad de contacto visual con esta.



Ilustración 2.2 - Lily



Ilustración 2.3 - Baliza Lily

Para conseguir que el drone siga a la baliza se ha instalado un receptor gps en el drone y otro en la baliza y gracias a la comunicación por bluetooth, la baliza es capaz de enviar sus coordenadas al drone, estos datos junto al tipo de seguimiento, desde una lado, desde un lateral, y los que se añadirán al producto final, permiten al drone calcular la ruta a seguir.

Principales de las características de Lily:

- 26.1 cm de lado y 8.18 cm de alto
- 1.3 Kg de peso
- 20 minutos de vuelo
- 4h de funcionamiento de la baliza
- Altura sobre la baliza, mínima 1.75m y máxima 15m
- Distancia de seguimiento, mínima 1.75m y máxima 30m
- 40 Km/h de velocidad máxima
- Resistente al agua, es capaz de aterrizar y flotar sobre el agua
- Grabación de video en FullHD

Al tratarse de un producto no finalizado, la empresa advierte de que puede sufrir cambios en las especificaciones antes de salir a la venta, que se prevé que sea en Febrero de 2016, con un coste 999\$.

2.2. Hexo+

Hexo+ [8], Ilustración 2.4, es un producto muy similar al anterior, usa gps y distintos modos de grabación para poder mantener una ruta relativa a la baliza.



Ilustración 2.4 - Hexo+

A diferencia de Lily, Hexo+ es un hexacóptero, tiene seis rotores en vez de cuatro, lo que permite mayor capacidad de carga, gracias a esto es posible incluir un gimbal [9], ilustración 2.5, lo que permite a los usuarios instalar cualquier cámara del tipo compacta. Incluye un gimbal para la cámara deportiva GoPro [10].



Ilustración 2.5 - Gimbal H3 3D de DJI

Un gimbal es un dispositivo capaz de mantenerse estable en una posición concreta, contrarrestando cualquier influencia externa. En el contexto en el que nos encontramos, son utilizados para la grabación de video, gracias a este accesorio es posible una grabación sin los movimientos bruscos o vibraciones que se produzcan durante el vuelo.

Para su funcionamiento no es necesaria una baliza específica, basta un smartphone con gps y bluetooth, lo que permite un uso más versátil, como manejar dos con un solo móvil, aunque no simultáneamente.

Principales características de Hexo+:

- Dimensiones: 62 * 52 * 12 cm
- 980 gr de peso en vacío
- 15 minutos de autonomía, con GoPro
- 45 Km/h de velocidad máxima

El proyecto estuvo en kickstarter [11], en el que consiguió una cifra de 1.306.920\$, de un total de 2336 patrocinadores. Durante la campaña, el precio del kit más básico de Hexo+ fue de 499\$. Actualmente el precio es de 1299\$, en pre compra, las primeras unidades se entregarán en Septiembre de 2015.

Analizando los desarrollos comerciales anteriores, podemos ver que su coste es muy alto, en parte porque buscan producir un producto completo, en vez de añadir autonomía a soluciones actuales.

Aunque nuestro proyecto se ha realizado para un modelo personalizado de multirrotor, es fácil exportar el desarrollo a cualquier otro drone sin apenas modificaciones. Esto permitiría dos formatos de venta, drone mas módulo de seguimiento y módulo de seguimiento por separado, para aquellos que ya dispongan de un drone.

3. ANÁLISIS

En este capítulo vamos a definir los requisitos que deberá cumplir el prototipo. También separaremos los que estén asociados a los componentes software y hardware.

3.1. Requisitos funcionales

En esta sección vamos a detallar las tareas prototipo debe de ser capaz de realizar [12]:

- Son necesarios dos modos de vuelo, uno manual y otro autónomo. Esto es útil en varios escenarios, por ejemplo, volar hasta donde se encuentra la baliza y activar el seguimiento o recuperar el control del drone en situaciones imprevistas.
- El cambio entre modos debe ser inmediato y sencillo, un interruptor en la emisora de radio puede ser útil y está siempre accesible.
- El prototipo tiene que ser capaz de volar de una forma estable. En modo autónomo, muchas vibraciones o movimientos bruscos pueden introducir errores en la detección de la baliza y en modo manual ayudara en su manejo.
- El microprocesador del sistema de seguimiento que se conecte al sistema del drone debe de ser capaz de leer y enviar señales Modulación por Ancho de Pulso (en inglés Pulse Width Modulation (PWM)) [13]. Esta es la forma habitual de comunicación en los dispositivos a radio control.
- Tiene que ser capaz de seguir objetos en movimiento marcados con una baliza, al ser un prototipo la velocidad de seguimiento estará auto limitada y nos restringiremos a espacios controlados.
- La baliza tiene que poder cambiarse de posición u objeto de forma fácil, el hecho de que no sea fija, hará más versátil el uso del quadróptero.

- El drone tiene que soportar el peso adicional que suponga el hardware de seguimiento, un peso por encima del recomendado reducirá el tiempo de vuelo, la vida útil de motores y baterías y puede producir accidentes.
- El quadróptero tiene que comportarse de forma segura ante los errores del sistema de seguimiento, como puede ser la pérdida de la baliza.
- El gasto de energía del sistema de seguimiento tiene que ser contenido. Un gasto excesivo de energía conlleva la necesidad de utilizar baterías de mayor peso, reduciendo la autonomía de vuelo.
- La velocidad de actualización de la posición de la baliza debe de ser rápida, al menos diez veces por segundo. Esto permite una mayor rapidez de reacción al prototipo.
- Los componentes del sistema de seguimiento deben de poder montarse en cualquier quadróptero con componentes estándar y capacidad de carga suficiente.

3.2. Requisitos no funcionales

Los requisitos no funcionales son aquellos criterios sobre la calidad del producto, usabilidad, rendimientos, estética, etc. En nuestro caso, hemos seleccionado los siguientes requisitos para el sistema:

- La autonomía debe de ser mayor de diez minutos una vez que el sistema esté completamente montado. Menos autonomía limita en gran medida los usos del drone.
- Un usuario con algo de experiencia en el área de los vehículos aéreos a radio control debe de poder manejar el prototipo, solo explicándole el funcionamiento y los controles. Si el sistema necesita de un periodo de aprendizaje muy largo limitamos los usuarios potenciales a los que podemos llegar.
- En el microcontrolador, la localización de la baliza que es la tarea que más tiempo va a consumir no debe ralentizar u obstaculizar la comunicación con el resto del sistema. Hay varias soluciones,

como usar interrupciones para la comunicación o un microcontrolador que permita la ejecución en paralelo de varios hilos.

3.3. Posibles escenarios

Vamos a enumerar casos que se pueden dar mientras se está usando el prototipo, más adelante en la sección de diseño se explicará cómo reaccionará ante estas situaciones:

- Cambio de modo automático a manual.
- Estando en modo manual, tiene localizada a la baliza y está listo para pasar a modo automático.
- Batería baja.
- Perdida de la baliza.
- Perdida de la comunicación con el sistema de seguimiento. El software o hardware no responde o alguno de los cables se ha desconectado.

4. DISEÑO

En este capítulo, se compone de las siguientes partes, elección de componentes hardware y software, diseño del protocolo de comunicaciones entre los diferentes sistemas, montaje del drone y desarrollo del software de seguimiento.

4.1. Componentes del drone

Los componentes seleccionados de acuerdo a los requisitos marcados en el inicio serán los que se detallan a continuación, acompañados de una descripción del papel que juegan dentro del prototipo. Entre estos componentes, destacamos la controladora de vuelo, el conjunto de motores, controladores de velocidad y hélices, el frame o chasis, el convertidor de voltaje, la emisora de radio y la batería.

4.1.1. Controladora de vuelo

Se puede considerar a la controladora de vuelo como el cerebro del drone, recibe la información del usuario a través del receptor de radio y el estado del drone gracias a distintos sensores y actúa sobre los rotores, realizando las correcciones oportunas para seguir las órdenes del piloto y a la vez mantener estable al drone. Es el equivalente a una centralita electrónica de un coche moderno, al igual que esta, en determinadas ocasiones es capaz de ignorar las acciones del piloto si lo cree necesario con el fin de evitar situaciones de riesgo.

La elección de la controladora ha sido relativamente sencilla, descartando aquellas que contenían características que no necesitábamos como gps, o soporte y que encarecían mucho el coste, y gracias a los muchos comentarios positivos de usuarios de distintos foros de aficionados al aeromodelismo y a los vehículos de radio control [14] [15] [16], hemos escogido la controladora de vuelo kk2.1 HC de HobbyKing (firmware versión 1.6) [17], Ilustración 4.1.



Ilustración 4.1 - HobbyKing kk2.1 HC

Además de cumplir los requisitos citados en el capítulo anterior, soporta comunicación por PWM y es sencilla de configurar y usar, la característica que más nos ha llevado a elegir este modelo, es la de auto-nivelado o self-level, se trata de un modo de vuelo seleccionable desde la radio, que mediante el sensor de aceleración [18] y el giroscopio [19] permite mantener una posición estática y oponerse a fuerzas externas como el viento.

La función de self-level tiene dos parámetros P-Gain y P-Limit. P-Gain permite ajustar que es más importante, los valores de los sensores o el control del piloto, un valor alto resta importancia a las decisiones del piloto, y el drone se volverá más “perezoso”, por el contrario un valor bajo lo hará más reactivo. P-Limit acota el control que tiene el usuario sobre el drone, si el límite está en 15 y la entrada del receptor de radio es 25, se reduce a 15.

La característica más diferenciadora de este modelo es la inclusión un accesorio compuesto de una pantalla y cuatro botones, que permite configurarlo sin la necesidad de un ordenador. Puedes cambiar los valores de la controladora y corregir fallos en cualquier sitio.

4.1.2. Motores

Los motores que se usan en quadrópteros son sin escobillas [20], a diferencia de los motores eléctricos tradicionales. Otros quadrópteros de tamaño muy reducido, menos de 15 centímetros de diámetro y 50 gramos de peso, también llamados microcópteros, sí que utilizan motores de corriente continua comunes.

A diferencia de los motores con escobillas, en los motores sin escobillas o brushless, no hay un cambio de polaridad mediante piezas mecánicas en constante rozamiento, lo que hace que aumente la eficiencia y disminuya el calor, además de alargar la vida útil del motor y necesitar un mantenimiento menor.

El problema de los motores brushless, es que no funcionan con corriente continua directamente, funcionan con corriente trifásica, por lo que es necesario un controlador electrónico de velocidad o ESC, sección 4.1.3, lo que hace que aumente su coste, pero compensa por la gran cantidad de potencia que es capaz de desarrollar con un peso menor.

EL modelo que hemos escogido de motor es el 2212 de DJI, la elección ha sido principalmente porque DJI comercializa un kit de propulsión, el E300 [21], que incluye motores, ESC y hélices. El precio del paquete es menor que soluciones similares de la competencia e incluye dos juegos de hélices completos.



Ilustración 4.2 - Motor 2212

4.1.3. Controladores electrónicos de velocidad (ESC)

Como se ha explicado en la sección anterior, para hacer funciona un motor brushless, es necesario cambiar de corriente continua a corriente alterna trifásica, de esto se encarga el ESC [22].

El funcionamiento es simple, dependiendo de la posición del rotor en el motor manda corriente por una de las tres conexiones que tiene con el motor para crear el campo electromagnético que hará que gire el rotor, los otras dos cables hacen el papel de conexión a tierra, y gracias a la corriente que pasa por cada uno de esos dos cables el ESC es capaz de saber el estado interno y cambiar si es necesario la conexión que alimenta al motor.

El controlador de velocidad incluido en el kit e300, es el OPTO-15A, Ilustración 4.3.



Ilustración 4.3 - ESC Opto-15A

4.1.4. Hélices

Las hélices incluidas en el kit e300 tiene la ventaja de que son autorroscables y no necesitan de equilibrado. Lo que las hace muy cómodas de usar y guardar. Las hélices tienen unas dimensiones 24 x 11 centímetros, Ilustración 4.4.



Ilustración 4.4 - Hélices DJI

4.1.5. Frame

El frame es el cuerpo del multirrotor, debe de ser resistente, ligero y con suficiente espacio para montar todos los componentes, el modelo elegido es el F450 [23] de DJI, Ilustración 4.5, es un chasis muy resistente y con espacio de sobra para nuestras necesidades.



Ilustración 4.5 - Frame F450 DJI

Además del chasis hemos tenido que adquirir un tren de aterrizaje [24], para permitir la colocación de las cámaras por debajo de las hélices y que estas no estorben.

4.1.6. Universal battery eliminator circuit (UBEC)

El UBEC tiene la función de reducir el voltaje de salida de las baterías, al voltaje de trabajo del resto de componentes del sistema, 5V. Normalmente este componente está integrado en los ESC, pero nuestro modelo no lo incluía.

El UBEC elegido es el distribuido por HobbyKing de 5 Amperios [25], Ilustración 4.6, este componente no es ni caro, ni complejo, la elección de este modelo en concreto ha sido por el prestigio del fabricante, más que por sus características.



Ilustración 4.6 - UBEC 5A HobbyKing

4.1.7. Emisora de radio y receptor

Al igual que en el caso de la controladora de vuelo, la elección del modelo ha estado condicionada por las opiniones de los aficionados a los multirrotores.

Hay una gran cantidad de modelos desde los profesionales de más de 2000€, a los enfocados a los aficionados.

El modelo elegido es el 9X [26] de Turnigy, ilustración 4.7, incluye, el receptor de radio ya emparejado, cinco canales auxiliares y un alcance de más de cinco kilómetros.



Ilustración 4.7 - Radio y receptor Turnigy 9X

4.1.8. Batería

La batería es de los últimos componentes que se escogen, dependen del resto del sistema, es necesario una batería que soporte la descarga de energía hacia el sistema, sobre todo hacia los motores, cuando esté a pleno rendimiento. Lo normal es sobredimensionar las baterías, para elegir una batería hemos tenido en cuenta que el conjunto de componentes escogidos puede alcanzar picos de 80 - 85 Amperios.

El fabricante de los ESC y motores que hemos escogido recomiendan baterías 3S o 4S. Las baterías se identifican por las celdas que tiene y su configuración, una batería 3S1P o 3S está compuesta por 3 celdas en serie,

3.7 voltios por celda hacen un total de 11.1V. La configuración en serie aumenta el voltaje y en paralelo aumenta la capacidad, aunque esta última es más rara.

Para este proyecto hemos elegido una batería 3S con una capacidad de 5000mAh y un peso de 412 gramos, de la marca Turnigy [27], Ilustración 4.8, lo que se traduce en unos 20 minutos de vuelo.



Ilustración 4.8 - Batería 3S Turnigy

Las baterías que se necesitan para los quadrópteros están hechas con unos componentes distintos al resto de baterías comunes como pueden ser las de los teléfonos móviles. Están compuestas de polímeros de litio, lo que les permite soportar grandes descargas de corriente, nuestro modelo soporta 100 Amperios de descarga de forma continua y 150 Amperios durante 10 segundos sin dañarse.

Tienen la ventaja de una densidad energética mayor que las de iones de litio, las comunes en móviles y pequeños aparatos electrónicos, pero las desventajas de que no pueden bajar de los 3V o se dañan de forma irreparable y de que existe un alto riesgo de incendio al usar o recargar baterías dañadas.

Nunca se deben de dejar baterías que han sufrido algún golpe cerca de objetos inflamables, ni dejar sin supervisión baterías cargándose. Para reducir los riesgos es normal guardarlas en bolsas ignífugas como esta [28], estamos acostumbrados a tratar con baterías en nuestro día a día y pueden resultar exageradas tantas precauciones, pero en una búsqueda rápida por internet podemos encontrar baterías arden al cargarlas sin tener ningún signo evidente de daño.

Por último decir, que la batería, al tener varias celdas en serie, impide que se carguen por igual, por lo que es necesario un cargador especial, el cargador que hemos comprado es el cargador DJI Phantom [29], para baterías de 2S a 4S. Este tipo de cargadores equilibran el voltaje de todas las celdas para evitar que ninguna corra el riesgo de sobrecarga.

4.1.9. Cables, adaptadores y otros

Aparte de los componentes principales ha sido necesario adquirir distintos cables o adaptadores, como este [30], porque el cargador y la batería no tenían el mismo tipo de conector, esto es normal en el mundo del radio control, cada marca comercializa sus productos con un tipo distinto de conector.

Otro de los componentes que hemos comprado es este avisador de batería baja [31], revisa el voltaje de cada celda individual y del conjunto, y cuenta con dos altavoces piezoeléctricos de gran potencia, se pueden escuchar hasta una distancia de 50 metros.



Ilustración 4.9 - Avisador de batería baja

Gracias a este accesorio hemos resuelto parcialmente el requerimiento de cómo reaccionará el drone en situaciones de batería baja, la solución más eficiente sería un aterrizaje autónomo, pero en este caso no es posible, porque como veremos a continuación ninguno de los microcontroladores es capaz de medir una corriente eléctrica superior a los 5 Voltios, por lo que es una solución de compromiso.

Una vez explicados todos los componentes escogido es necesario dar el siguiente detalle, el peso recomendado por el kit de motorización e300 es de 300 gramos por eje, como estamos construyendo un quadcóptero, el peso recomendado es 1.2 kilogramos, y el empuje máximo que entregan los motores es 2.4 kilogramos.

El peso total del drone es de 1.15 kilogramos incluyendo batería, lo que deja algo de margen para colocar una cámara de video o algún accesorio extra.

4.2. Componentes del sistema de seguimiento

A lo largo de esta sección detallaremos todos los componentes del sistema de seguimiento. Empezaremos definiendo el tipo de seguimiento que se llevara a cabo (Sección 4.2.1). A continuación, introduciremos las diferentes alternativas de microcontroladores que están disponibles en el mercado y la selección que mejor se ajusta a nuestros objetivos (Sección 4.2.2). Una pequeña descripción de las cámaras que usaremos para el seguimiento (Sección 4.2.3) y la fuente de alimentación usada para dar energía al sistema de seguimiento (Sección 4.2.4). Por último, daremos un resumen algorítmico del programa implementado para realizar el control del seguimiento (Sección 4.2.5).

4.2.1. Diseño del seguimiento

Existen distintas formas de seguimiento, basado en video, por triangularización o con sensores de distancia. En particular, hay dos modelos que se utilizan más ampliamente y que definimos a continuación:

- Una de las formas de seguimiento investigadas es la triangularización, que como su nombre indica consiste en usar tres receptores en el drone y un emisor en la baliza, pudiendo estar basados en comunicación por bluetooth [32], radiofrecuencia o ultrasonidos. Gracias a las diferencias en la fuerza de la señal recibida se puede calcular la posición de la baliza.

Sin embargo, esta opción no es factible por varias razones. En primer lugar por ser muy cara y difícil de llevar a cabo. La posición de los receptores está limitada al tamaño del drone, y en tan poco espacio no es posible conseguir precisión suficiente. Por último, las señales producidas por el mando de radio control y el ruido y los campos electromagnéticos generados por los motores, pueden empeorar la precisión.

- El siguiente método es el seguimiento usando video, dentro del seguimiento por video hay varias ramas que hemos tenido en

cuenta, la más interesante es la búsqueda del objeto a seguir mediante el análisis de características del objeto en la imagen, además el algoritmo es capaz de variar las características que se buscan, es decir, aunque el objeto gire o se oculte parcialmente el seguimiento no se detiene.

Este método sigue sin ser factible por su elevado consumo de recursos, hay que recordar que el seguimiento se realizará con un microcontrolador. Necesitamos un algoritmo más simple, por eso elegimos la búsqueda de patrones simples en la imagen, un ejemplo es la lectura de códigos de barras o códigos QR [33], códigos de barras bidimensionales. Estos algoritmos son extremadamente simples, pero para que sea posible la lectura de un patrón, como los QR, a una distancia de seguimiento, entre 1 y 3 metros es necesario que sea grande, demasiado para ser útil.

- Finalmente, un formato fácil de localizar en video y que tenga un tamaño reducido será el uso de una baliza que disponga de con diodos emisores de luz o leds [34] .

Las dos ventajas principales son, no depende de la luz exterior y simplifica la localización de la baliza. Al emitir luz propia, es inmune a los cambios en la iluminación y solo es necesario buscar en la imagen fuentes de luz.

En la solución escogida, los leds se sitúan formando un rombo, uno en cada vértice. La elección de esta disposición ha estado condicionada por las propiedades geométricas de los paralelogramos [35]:

- Cualquier transformación afín no degenerada transforma un paralelogramo en otro paralelogramo. Aunque se gire la baliza en cualquier eje del espacio, en la imagen que capture la cámara siempre se verá un paralelogramo.
- El centro del paralelogramo se encuentra en el punto en que se bisecan sus dos diagonales. Si recorremos los puntos de luz en la

imagen por parejas, calculando el centro del segmento que las une, y encontramos que dos de esos centros están cerca, podemos determinar que esos dos pares de puntos forman un paralelogramo.

Una vez decidido qué tipo de baliza vamos a usar, buscamos los componentes en una tienda de electrónica, son comunes y bastante baratos. Leds, resistencias, una pila de 9V, un porta pilas, una placa perforada para simplificar el montaje y una cartulina de goma eva para el cuerpo de la baliza. Los leds utilizados son de alta luminosidad, por lo que necesitan más voltaje y obliga a usar una pila de 9V.

El diagrama eléctrico de la baliza, ilustración 4.10, es muy simple e incluye un interruptor para simplificar su uso y una apertura con velcro para cambiar la pila cuando sea necesario.

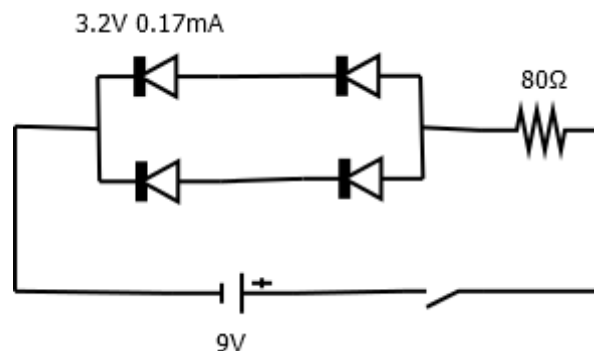


Ilustración 4.10 - Diagrama eléctrico baliza

En las siguientes imágenes se muestra el interior de la baliza, ilustración 4.11, la baliza finalizada, ilustración 4.12, y la baliza encendida, ilustración 4.13.

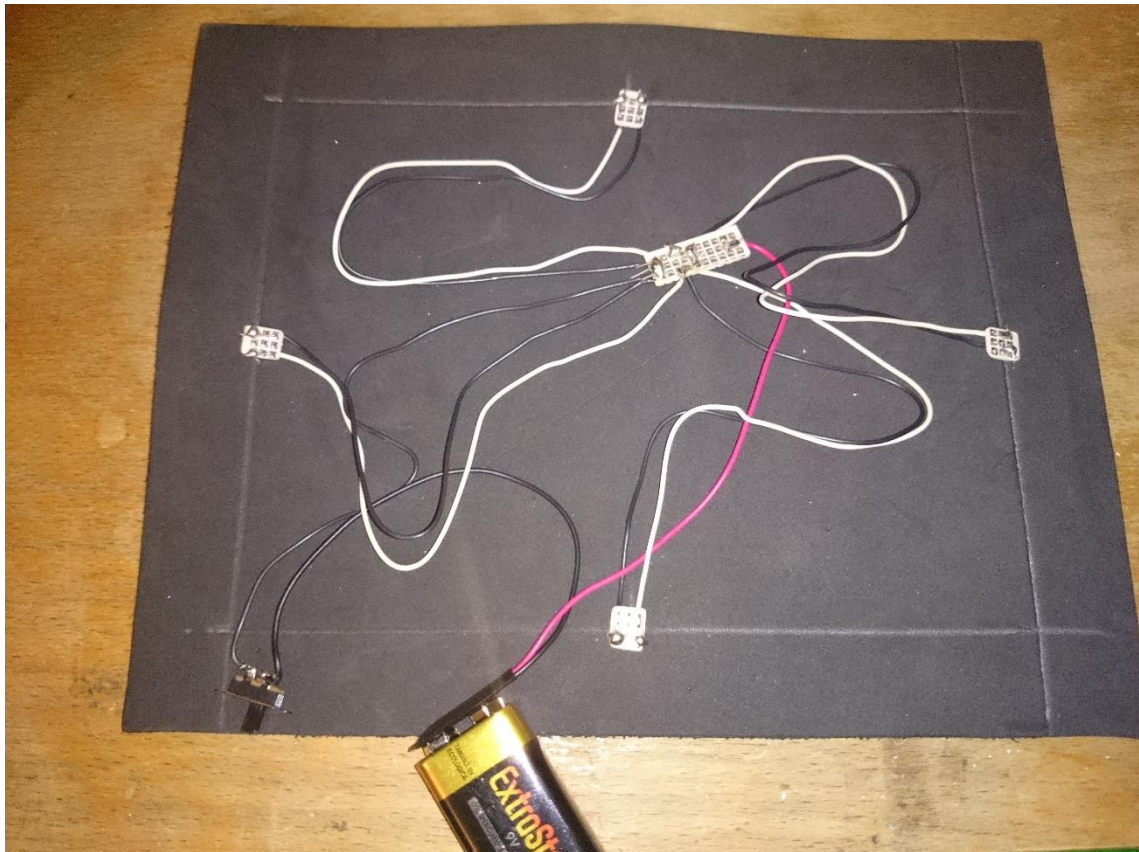


Ilustración 4.11 - Baliza (interior)



Ilustración 4.12 - Baliza



Ilustración 4.13 - Baliza encendida

Con esta baliza podemos conseguir una localización rápida y que se puede realizar con una webcam normal, no hay necesidad de una alta calidad o resolución. La posición se puede determinar en dos ejes altura y anchura, nos faltan los datos de un tercer eje, la distancia al drone, que podemos adquirir añadiendo una segunda cámara.

Al usar estereovisión [36], tratamiento de imágenes usando dos cámaras, podemos simular la percepción humana de la profundidad, calculando la distancia de la baliza en las dos imágenes determinamos la profundidad. A más distancia entre imágenes, más cerca se encuentra el objeto, aunque implica doblar el tiempo de proceso.

4.2.2. Microprocesadores.

Como se ha explicado anteriormente en la sección 4.2.1, es necesario un microprocesador relativamente potente para procesar dos fuentes de video, capaz de comunicarse por PWM, compatible con webcams estándares, de pequeño tamaño y poco consumo.

4.2.2.1. Intel Edison

Los microprocesadores que hemos encontrado que cumplen las condiciones anteriores son Raspberry PI [37] e Intel Edison [38]. Ambos se encuentran en un rango de precios similar, el Edison es algo más caro, pero es el doble de potente en ejecuciones mono núcleo y hasta cuatro veces más potente en ejecuciones que aprovechan los dos núcleos que integra, más eficiente y mucho más pequeño. En la ilustración 4.14 se puede ver el tamaño.

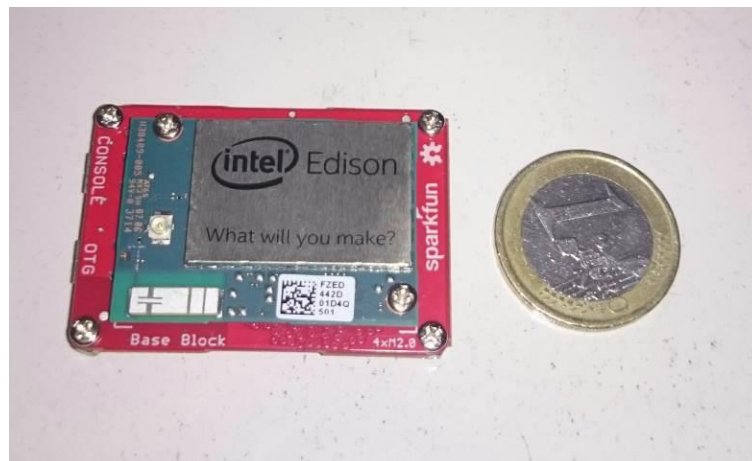


Ilustración 4.14 - Intel Edison

La elección entre la Raspberry Pi y el Intel Edison está clara, gana Intel Edison prácticamente en todos los aspectos, excepto en el precio, a lo que hay que añadir el precio de una placa de base [39], ya que carece de interfaces. La placa base añade dos puertos Micro USB, uno para conectar periféricos y otro para alimentación. A pesar del incremento de peso merece la pena la inversión.

Intel Edison es una plataforma desarrollada por Intel para facilitar el desarrollo de dispositivos wearables [40], accesorios de pequeño tamaño, que se colocan encima o incluido en la ropa. Dado su enfoque, el producto final debía de ser muy pequeño, y con un consumo muy reducido, pero a la vez potente. Salió a la venta en noviembre de 2014 y en los pocos meses que lleva en el mercado ha conseguido que una gran comunidad de desarrolladores lo apoye. Sus principales características son:

- Procesador Intel Atom, doble núcleo a 500MHz.
- 1 GB de memoria RAM.

- 4 GB de memoria interna.
- Wifi n y Bluetooth 4.0.
- Dimensiones: 3.5 x 2.5 x 0.39 centímetros.
- Consumo en reposo: 13mW y 35mW con Wifi activo.
- Consumo máximo: picos de 500mW usando Wifi.

El único requisito que no cumple ninguna de las dos placas es la comunicación por PWM. Ambas la soportan pero al funcionar sobre un sistema operativo multitarea, Raspbian [41] en la Raspberry Pi y Yoctolinux [42] en el Intel Edison, comparten tiempo de CPU con otros procesos lo que provoca una pérdida de precisión en la señal. La precisión mínima necesaria es de ± 4 microsegundos.

Este problema se soluciona usando otro microcontrolador de bajo coste, el Arduino Nano [43]. En la sección 4.2.2 detallaremos sus características y el trabajo que desempeña en el sistema.

4.2.2.2. Arduino Nano

En la anterior sección se explicaba la necesidad de un microcontrolador extra capaz generar señales PWM con más precisión que el Intel Edison, el Arduino Nano, ilustración 4.15.

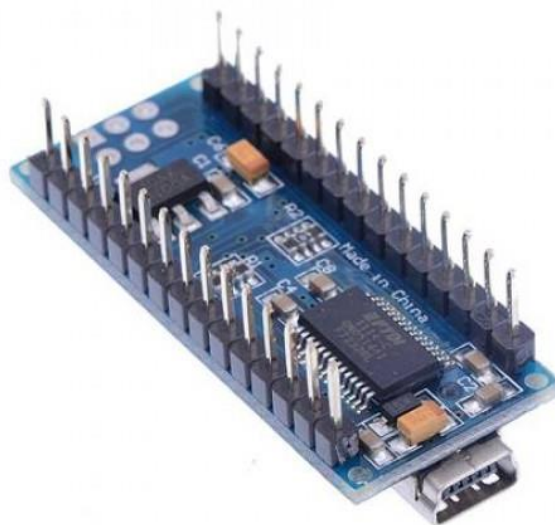


Ilustración 4.15 - Arduino Nano

Arduino es una plataforma hardware Open-Source [44] para el desarrollo de proyectos electrónicos. La empresa Arduino distribuye placas oficiales, pero al ser hardware libre, muchos otros fabricantes crean sus propios productos, a veces añadiendo modificaciones sobre los esquemas originales, y que normalmente son más económicos.

Otra de las ventajas de añadir este microcontrolador al prototipo, es que Arduino nos va ayudar a intercomunicar el Edison con el resto del sistema. El hecho de que se diseñara enfocado a proyectos de electrónica facilita mucho la integración en el sistema.

Las características más destacadas de este microcontrolador son las siguientes:

- Microcontrolador: Atmel ATmega328.
- 14 pines de entrada/salida digitales, de los cuales 6 soportan salida PWM, cualquier pin digital es capaz de leer PWM.
- Velocidad de reloj: 16MHz.
- Dimensiones: 4.3 x 1.9 centímetros.

4.2.3. Cámaras

Para el proyecto hemos escogido unas cámaras estándar pero con una calidad decente para su precio, el modelo es el C170 de Logitech [45], ilustración 4.16. Ofrecen una resolución máxima de 1024 x 768, 30 imágenes por segundo y un bajo peso. El algoritmo desarrollado trabaja a esa resolución, pero a solo 15 cuadros por segundo, también disminuiríamos su peso y tamaño quitando la carcasa.



Ilustración 4.16 - Webcam C170 Logitech

4.2.4. Powerbank

Una powerbank es una batería de 5 Voltios que se suele usar para la recarga de móviles, cámaras compactas o mp3 en momentos en los que no se tiene acceso a un cargador de pared. Nuestro modelo cuenta con una capacidad de 2200mAh y solo unos pocos gramos de peso.

El Intel Edison, el Arduino Nano y las cámaras se alimentan de la powerbank a través de una conexión USB, ilustración 4.17,



Ilustración 4.17 - Powerbank

4.3. Corrección del coste del proyecto

Una vez elegidos todos los componentes hardware podemos actualizar el presupuesto del proyecto con los costes reales.

En la tabla 4.1 podemos ver el coste total del drone.

Partes del Drone	Coste
Controladora de vuelo	21,57 €
Controladores de velocidad + Motores + Palas	109,00 €
Frame	49,00 €
Tren de aterrizaje	17,00 €
Universal battery eliminator circuit	4,56 €
Radio (transmisor y receptor)	49,79 €
Batería + cargador	48,35 €
Cables, adaptadores y otros	15,00 €
Envío y Aduanas	50,00 €
Total	364,27 €

Tabla 4.1 - Coste drone

En la tabla 4.2 se incluye el coste de la baliza y en la tabla 4.3 el coste del sistema de seguimiento al completo.

Baliza	Coste
4 Leds	1,20 €
Goma Eva A3	0,60 €
Porta pilas	0,35 €
Placa perforada	1,50 €
Resistencia	0,05 €
Pila 9V	1,15 €
Total	4,85 €

Tabla 4.2 - Coste baliza

Sistema de seguimiento	Coste
Intel Edison	59,00 €
Placa Base Edison	35,60 €
Arduino Nano	8,50 €
Cámaras	39,20 €
PowerBank	7,50 €
Envío	9,50 €
Total	159,30 €

Tabla 4.3 - Coste sistema de seguimiento

Y por último en la tabla 4.4, tenemos el coste total del proyecto, podemos ver que se ha mantenido el presupuesto estimado en un principio.

Coste final	Coste
Recursos Humanos	7.900,00 €
Hardware	565,76 €
Software	0,00 €
Total	8.465,76 €

Tabla 4.4 - Coste final del proyecto

El coste estimado era de 8377,19 a 8572,19 euros, y el real de 8.465,76 euros. Hemos cumplido con el presupuesto gracias a que se han elegido componentes que cumplían con nuestros requisitos y no los hemos sobredimensionado, eligiendo artículos más caros de gamas superiores.

4.4. Montaje del drone

Antes de comenzar el montaje del drone es necesario establecer donde se va a colocar cada componente. El chasis elegido dispone de seis zonas para colocar componentes:

- Zona central superior, ahí colocaremos la controladora de vuelo.
- Zona central, al ser una estar más protegida, es una buena zona donde colocar la placa que distribuirá la energía y el UBEC.
- Zona central inferior, como está cerca del centro de masas es ideal para colocar el elemento más pesado. La batería, pesa unos 400 gramos, más del 30% del total.
- Brazos del chasis, en los extremos van los motores, en el centro los ESC y en la base de dos brazos pondremos el Arduino y la receptora de radio.
- Laterales, en un lateral se colocará el Edison y en el otro la powerbank. El peso de la powerbank ayudara a compensar el de los dos microcontroladores y la receptora de radio al lado opuesto.

Para ayudar en el montaje del drone hemos creado dos pequeños esquemas eléctricos del drone, uno para el drone en sí, ilustración 4.18, y otro para el sistema de seguimiento, ilustración 4.19, puesto que usan dos fuentes de alimentación distintas.

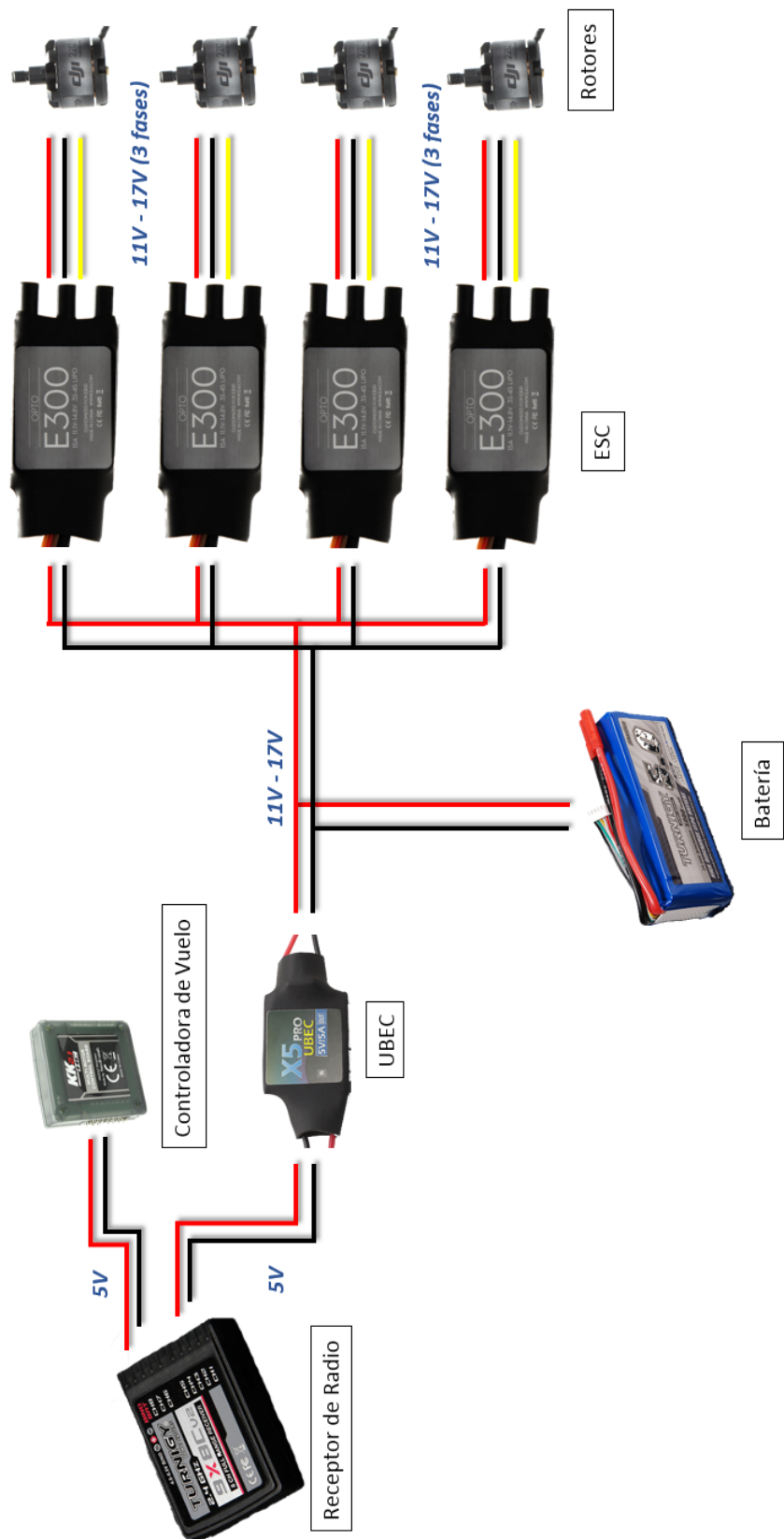


Ilustración 4.18 - Esquema eléctrico dron

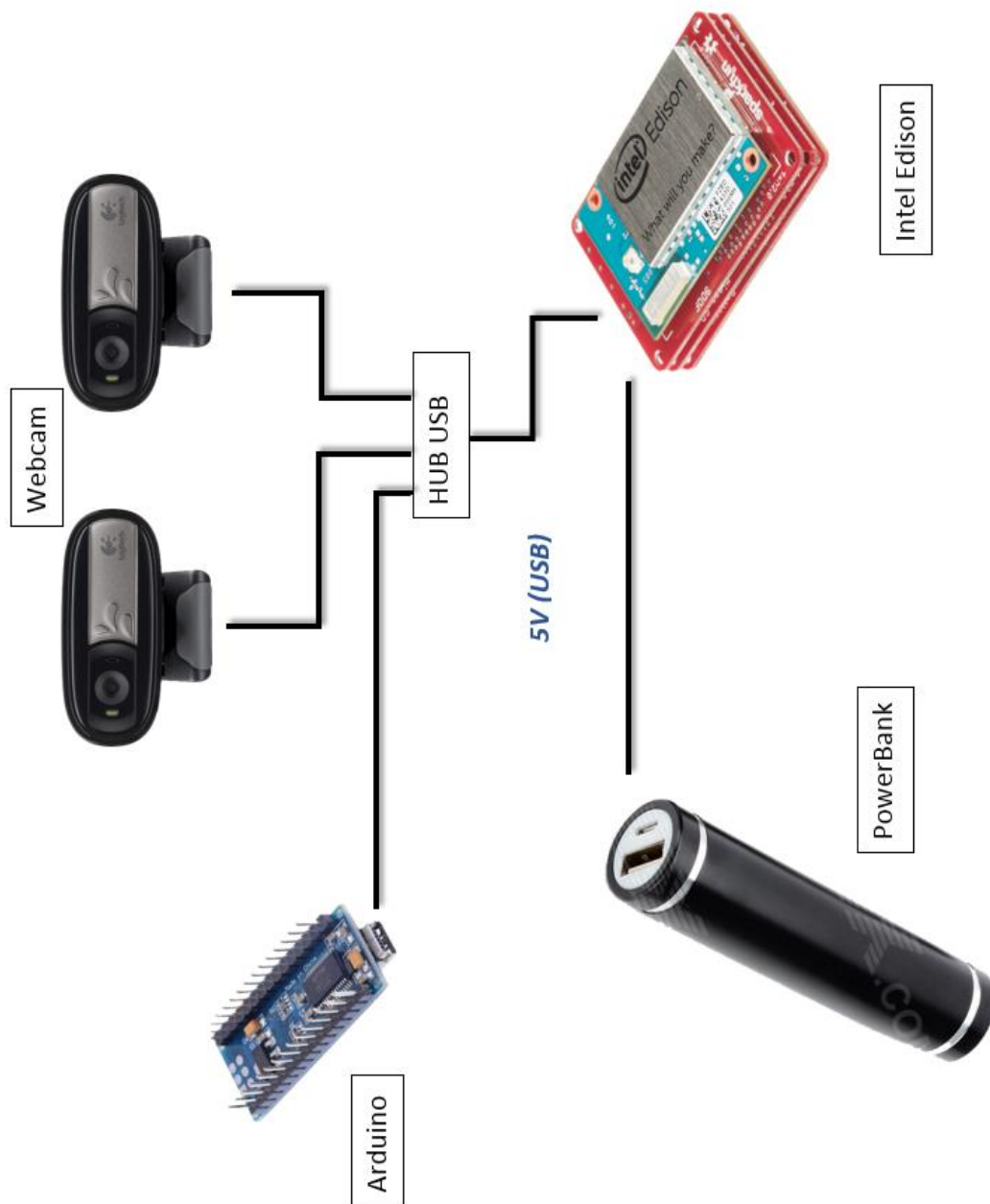


Ilustración 4.19 - Esquema eléctrico sistema de seguimiento

Ahora que tenemos todo lo necesario, vamos a empezar a la construcción de quadróptero. Lo primero será ensamblar los brazos y el tren de aterrizaje a la placa central inferior, en este paso aprovechamos para colocar el velcro que soportará la batería, ilustración 4.20.



Ilustración 4.20 - Montaje chasis inferior

El siguiente paso es el montaje de los ESC, se montan en la parte inferior de los brazos con la ayuda de bridas, ilustración 4.21. Los ESC se calientan con el uso, y esa posición ayuda a reducir la temperatura.



Ilustración 4.21 - Montaje ESC

El kit de motorización incluye una pequeña placa para ayudar a hacer el montaje eléctrico, tiene dos zonas para soldar el conector de la batería y siete salidas para los ESC y otros accesorios que necesiten alimentación como el UBEC. Ha sido difícil soldar los cables por su grosor, en especial el que conecta la batería. Para proteger las soldaduras y evitar cortocircuitos con otras partes de metal, se ha cubierto la parte superior con una pequeña capa de gomaespuma.

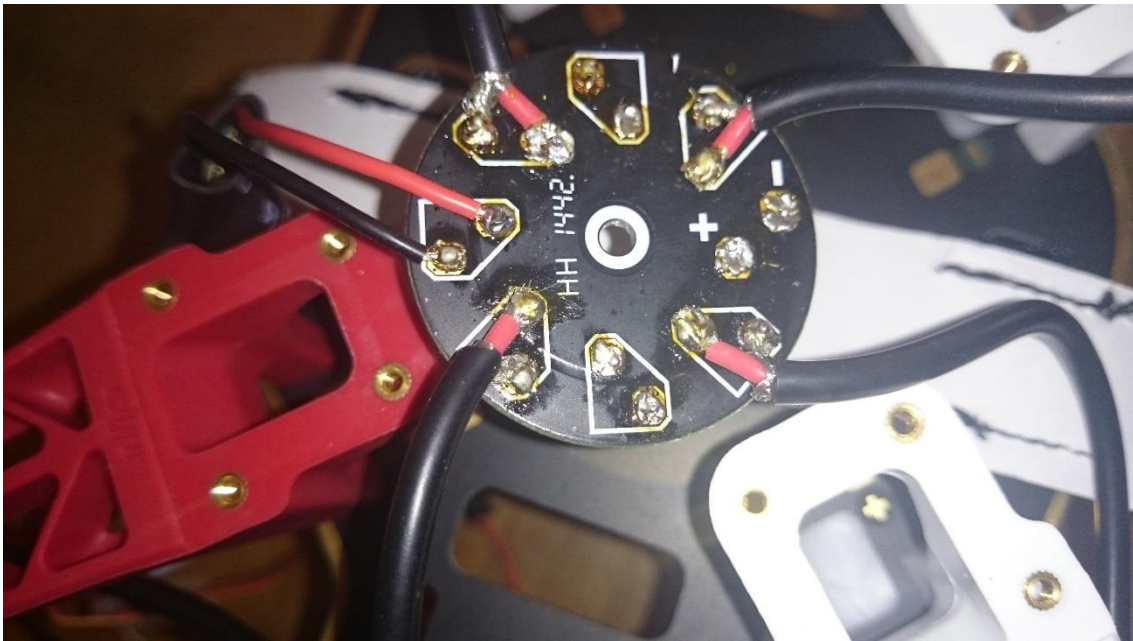


Ilustración 4.22 - Placa distribución energía

Lo siguiente es atornillar los motores a los brazos y conectarlos a los ESC. Los tres cables que conectan cada motor a su controladora de velocidad son iguales y no vienen marcados, ilustración 4.23. Al ser corriente trifásica no existe polaridad, aunque si no están conectados en el orden correcto, el motor girará en sentido contrario. Es importante comprobar esto una vez montado el drone, si algún motor no gira en el sentido que debe, basta con cambiar dos conectores entre sí para invertir el giro.



Ilustración 4.23 - Detalle conexión motor y ESC

Una vez llegados a este punto y habiendo comprobado que las soldaduras están bien, podemos colocar la placa superior del chasis y unir los microcontroladores, la controladora de vuelo, el receptor de radio, el concentrador USB (ilustración 4.24), y la powerbank (ilustración 4.25) al chasis. Todos estos componentes se unen al frame mediante tiras de gomaespuma con pegamento en las dos caras. La gomaespuma reduce las vibraciones y permite pegar superficies irregulares como la superficie del Arduino nano. Los componentes pequeños con poca superficie para la tira de pegamento, se han reforzado con bridas por seguridad.

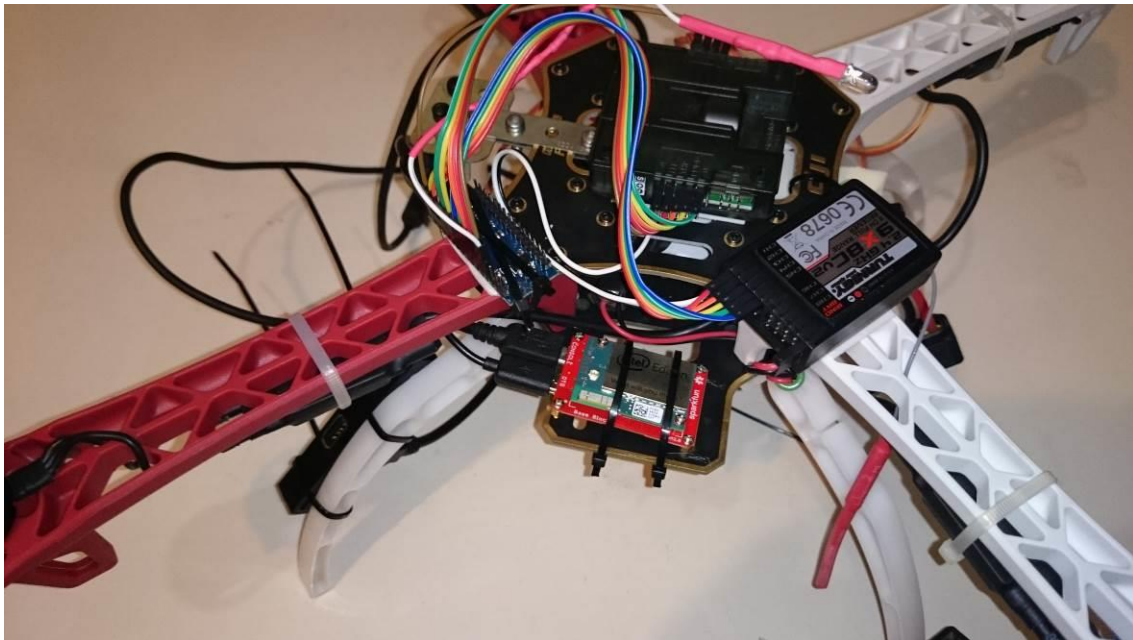


Ilustración 4.24 - Chasis detalle microcontroladores



Ilustración 4.25 - Chasis detalle powerbank

Como se puede ver en la parte superior de la ilustración 4.24, hay un led que más tarde se fijará a la parte posterior, para poder mostrar posibles errores al usuario. Las cámaras se han montado sin su carcasa, ilustración 4.26, para disminuir peso y tamaño. Para unir las al chasis en una posición en la que no molesten las hélices se han usado piezas del juego de construcción Meccano [46], ya que son versátiles, ligeras y resistentes.

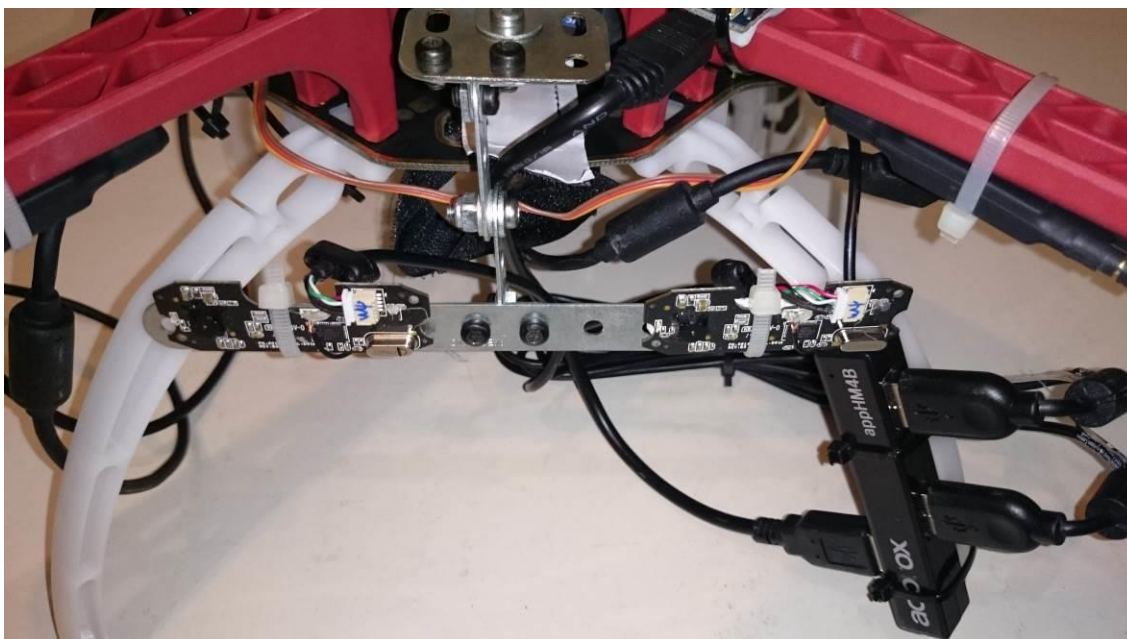


Ilustración 4.26 - Chasis detalle cámaras

Con esto terminamos el montaje del drone. Lo siguiente es la configuración de la controladora de vuelo, gracias a su pantalla es un proceso bastante sencillo. Lo primero de todo es seleccionar el tipo de multirrotor. Con cuatro motores podemos elegir dos distribuciones, ilustración 4.27, por las características del chasis, no es simétrico si usamos la configuración de la izquierda, y por la colocación de las cámaras, tenemos que usar la distribución de la derecha. La línea que se aprecia en la imagen, muestra el frontal del drone.

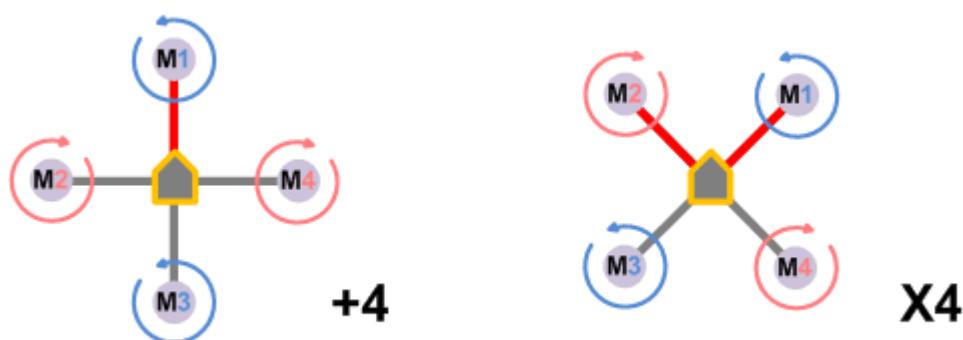


Ilustración 4.27 - Distribución quadcópteros

Una vez seleccionado la distribución de motores tenemos que comprobar el sentido de giro de los mismos. Por seguridad todos estos cambios es

necesario hacerlo con las hélices desmontadas, como se ha indicado antes. Si algún motor no gira en el sentido correcto es posible invertirlos intercambiando dos de los cables que conectan el motor con el ESC.

Según el manual, lo siguiente que tenemos que hacer es calibrar los ESC. esto permite que los controladores de velocidad sepan cual es el recorrido de los controles del mando. La secuencia es sencilla, con la radio encendida y el stick de potencia en la posición superior, conectar la batería al drone pulsando los botones 1 y 4 de la controladora de vuelo, cuando se escuchen pitidos de confirmación bajar el stick a la posición más baja, los ESC emitirán un secuencia “alegre” de tonos como confirmación, o una secuencia repetida de tres tonos iguales en caso de fallo, esto último puede variar dependiendo del modelo de ESC.

Siguiendo el manual, ahora se deben calibrar los sensores, para lo cual es necesario colocar el drone en una superficie horizontal y activar la calibración. Lo siguiente es la configuración de los valores P-gain, P-limit, I-gain e I-limit, de cada uno de los tres ejes del espacio, estos dependen de la forma y de la distribución de pesos del drone. Esta configuración es difícil de realizar incluso para gente con experiencia, pero ayudan al algoritmo de estabilización a reducir vibraciones o movimientos erráticos no deseados. Los valores por defecto son para multirrotores con una distancia entre motores de 40 a 50 centímetros, nuestro prototipo tiene 45 centímetros, por lo que decidimos mantenerlos, en las primeras pruebas comprobamos que vuela de forma estable, aunque con suficiente tiempo y algo de ayuda, podríamos mejorar de forma notable la experiencia de vuelo.

El último paso es la configuración del modo self-level, detallado en la sección 4.1.1. Como curiosidad decir que en el primer vuelo de prueba el drone tendía a “botar”, por lo que no conseguíamos estabilizarlo en altura, aunque sí en el resto de direcciones. Buscando el error en foros de aeromodelismo encontramos el siguiente video [47], en el que detallan que comportamiento tiene el drone según la configuración del self-level. En los anexos se incluye un video mostrando el vuelo del quadróptero en modo manual con y sin el efecto “bote”.

4.5. Comunicación hardware-software

A continuación detallaremos como se comunican los diferentes componentes del sistema, lo primero es mostrar un esquema de las conexiones realizadas, ilustración 4.28.

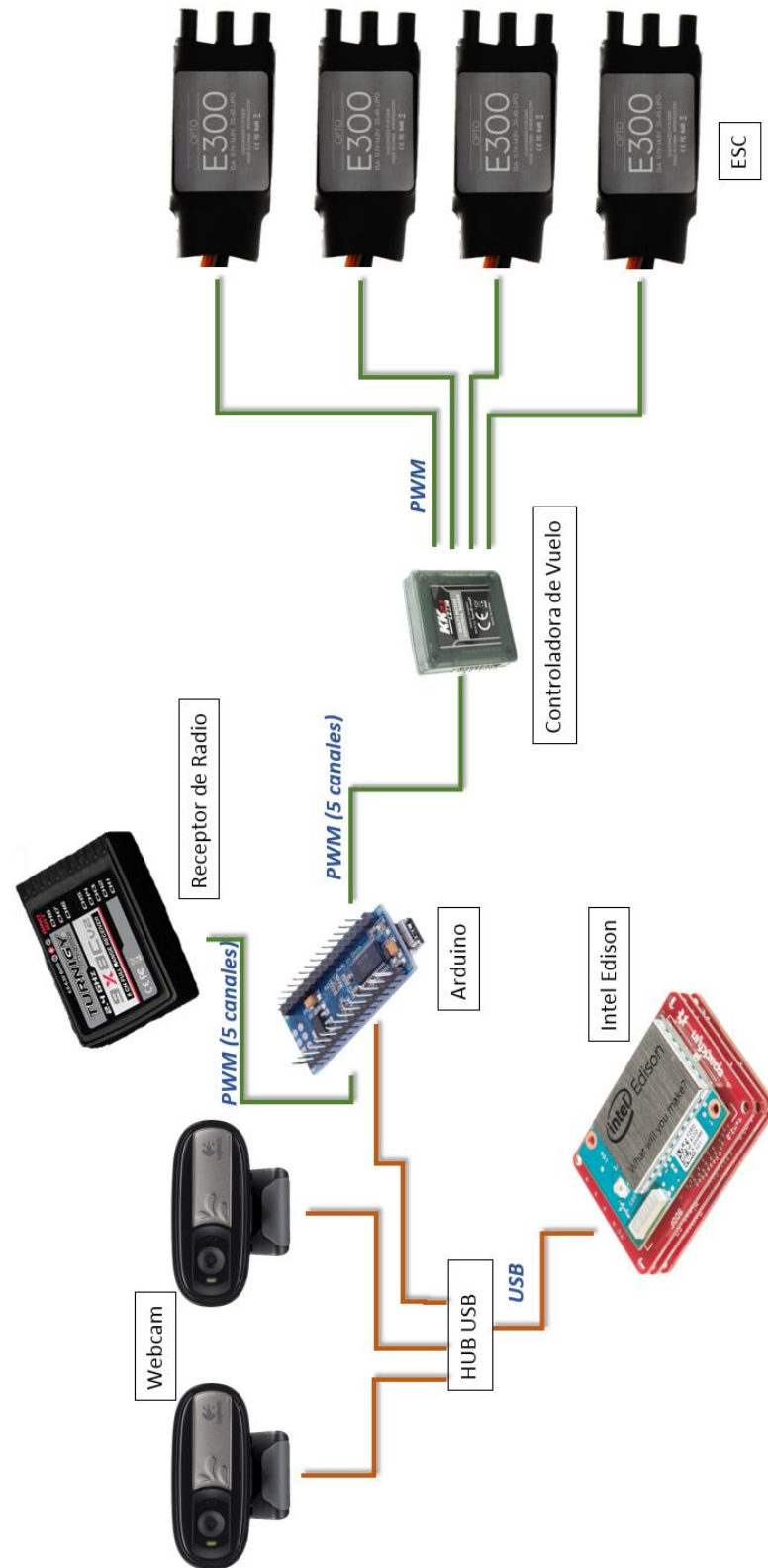


Ilustración 4.28 - Esquema conexión electrónica

Como se puede ver en el diagrama el punto central es el Arduino Nano, este se conecta al Edison por una conexión serie a través de USB, con el receptor de radio por cinco canales PWM y con la controladora de vuelo con otros cinco canales PWM.

El Arduino recibe tanto la información del usuario como la del sistema de seguimiento, elige una u otra dependiendo del modo de vuelo, manual o autónomo, y la envía a la controladora de vuelo. El quinto canal de la radio está asociado a un interruptor en el mando para elegir entre los dos tipos de vuelo, normalmente este canal activa y desactiva el modo self-level, esta función la asume ahora el Arduino que mantiene siempre activo este modo, para que la controladora ayude a un vuelo estable.

El resto de canales tienen las siguientes funciones:

- *Aileron*, controla el movimiento a derecha e izquierda, esto se consigue aumentando la potencia de los motores en el lado contrario del movimiento, y reduciendo la potencia de los otros dos.
- *Elevator*, mueve el drone hacia delante y hacia atrás, se consigue cambiando la potencia de los dos motores anteriores y posteriores de forma conjunta, al igual que el anterior.
- *Throttle*, modifica directamente la potencia de los motores, el resto de movimientos se consiguen modifican esta señal. Aunque es el que controla la altura del drone, no lo hace directamente, diferentes condiciones externas, como viento, menos densidad del aire, o el hecho de estar cerca del suelo, provocan que a la misma señal el empuje vertical sea distinto.
- *Rudder*, el cuarto y último canal, hace que el dron gire sobre sí mismo. Este efecto se consigue aumentando la potencia de dos motores opuestos y disminuyendo la de los otros dos.

En la página web [48], se explica de forma clara y gráficamente (ilustración 4.29) como se consiguen los diferentes movimientos en un quadróptero y de otros vehículos aéreos.

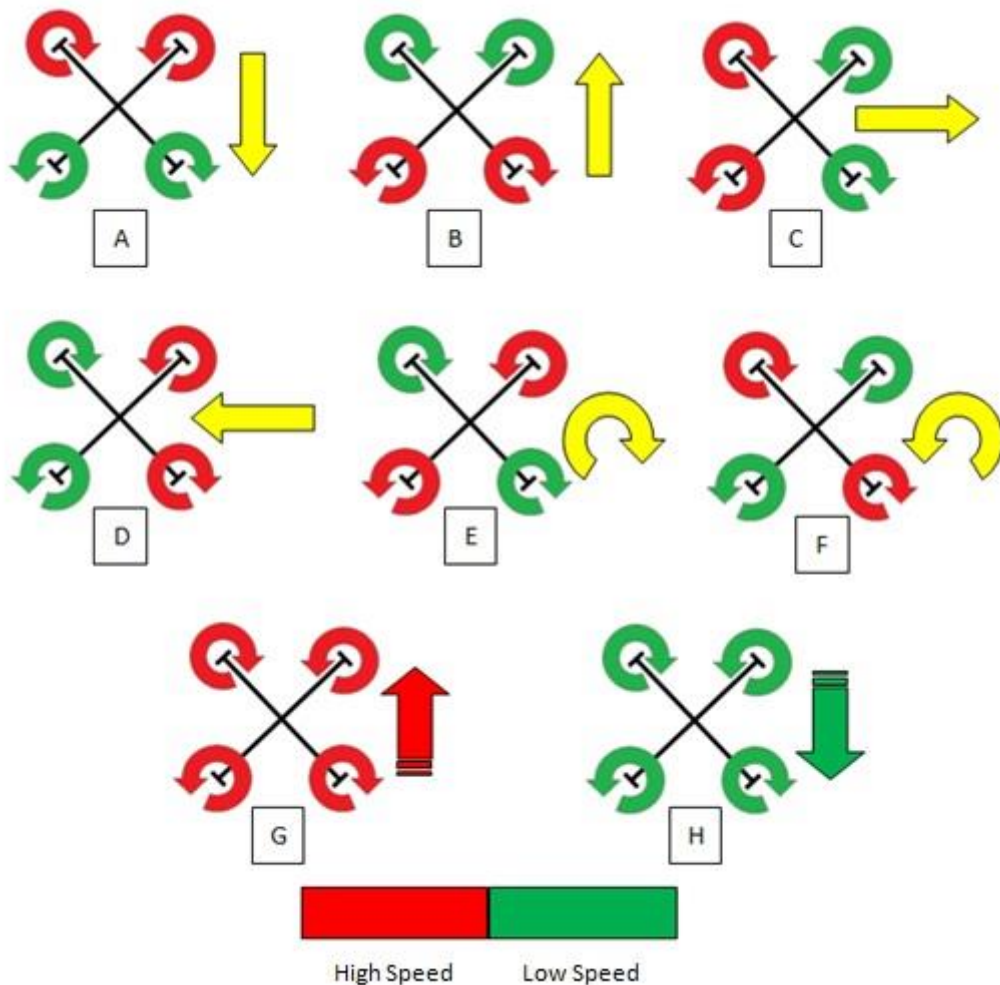


Ilustración 4.29 - Detalle de movimientos del drone

En los canales PWM no hemos tenido que definir la forma de comunicación ya que se trata de un estándar, sin embargo el Arduino y el Edison se comunican por medio de un puerto serie, y para que se puedan entender es necesario definir un protocolo.

Según [49], un protocolo define las reglas para que dos interlocutores o sistemas se entiendan, entre las reglas se incluye, el medio de transmisión, detección de errores, tipos de mensaje, sintaxis, etc. La mayoría de estas reglas están ya definidas en la comunicación por el puerto serie, nosotros definiremos la sintaxis, como se forman y se interpretan los mensajes.

Para simplificar solo habrá comunicación en un sentido, del Intel Edison al Arduino Nano, y como tamaño del mensaje usaremos la mínima unidad de información que se envía a través de este medio, el byte. En los dos primeros

bits del mensaje esta la cabecera, donde se define el tipo de mensaje, el resto son los datos en sí.

Vamos a usar tres cabeceras distintas, una para estados de error, y otras dos para la enviar la posición de la baliza respecto del drone, en la tabla 4.5, podemos ver los distintos tipos de mensajes.

Mensaje	Información
0bXXXX0011	Todo Ok.
0bXXXX0111	La baliza no está localizada.
0bXXXX1011	El hilo que controla las cámaras ha fallado y se ha cerrado.
0bYYYYXX01	YYY es la posición horizontal de la baliza y XXX la distancia.
0bYYYYXX10	XXX es la posición vertical de la baliza.

Tabla 4.5 - Mensajes del protocolo

4.6. Plataformas de desarrollo

En este apartado comentaremos los lenguajes de programación (Sección 4.6.1) e IDEs, entornos de desarrollo integrado (Sección 4.6.2), utilizados.

4.6.1. Lenguajes de programación

La elección de los lenguajes de programación ha sido sencilla. Arduino tiene su propio lenguaje de programación, muy similar a C. Para el microcontrolador Intel Edison, hay más posibilidades, la única limitación que tenemos es la existencia de bibliotecas para comunicarnos con las cámaras, y el puerto serie.

La biblioteca para comunicarnos con las cámaras que vamos a usar es OpenCV Versión 2.4.7 [50], es de código abierto y muy reconocida por usuarios del área visión por ordenador. No solo permite la obtención de imágenes de cámaras web, sino que además contiene una colección muy completa de algoritmos para tratamiento de imágenes y videos. En concreto nosotros usaremos el algoritmo de detección de manchas o blobs en imágenes, SimpleBlobDetector.

OpenCV está disponible para C++ [51], Python [52] y Java [53]. La ventaja de Java y Python con respecto a C++ es la independencia del sistema operativo. Esto permite desarrollar y depurar el código en el ordenador y una vez que está testado, portarlo al Edison con pocos o ningún cambio en el código fuente. La falta de una biblioteca oficial para la comunicación por el puerto serie para Java, nos lleva a decantarnos por Python, en concreto la versión 2.7.

Python es un lenguaje desarrollado para simplificar y reducir el código necesario para crear cualquier algoritmo. Bastan dos líneas de código para establecer comunicación con otros dispositivos serie, una para importar la librería y otra para inicializar el puerto.

El lenguaje que usa Arduino para programar sus microcontroladores está basado en C. Son muy similares, cualquier programador que conozca C puede desarrollar software para Arduino.

4.6.2. Entornos de desarrollo integrado

Al igual que sucede con el lenguaje de programación, Arduino dispone de un IDE propio (ilustración 4.30). Incluye un editor de texto, un compilador y un programador de microcontroladores Arduino. Tiene un diseño simple, limpio y sencillo de usar.



Ilustración 4.30 - IDE Arduino

El IDE para Python que hemos usado ha sido PyCharm [54], ilustración 4.31, lo hemos elegido por su simplicidad y facilidad de uso. Solo hemos utilizado su editor de textos y la consola de Python que incluye, para ejecutar y

depurar el código fuente. Aunque incorpora más utilidades para proyectos de más envergadura, no hemos tenido la necesidad de recurrir a ellas.

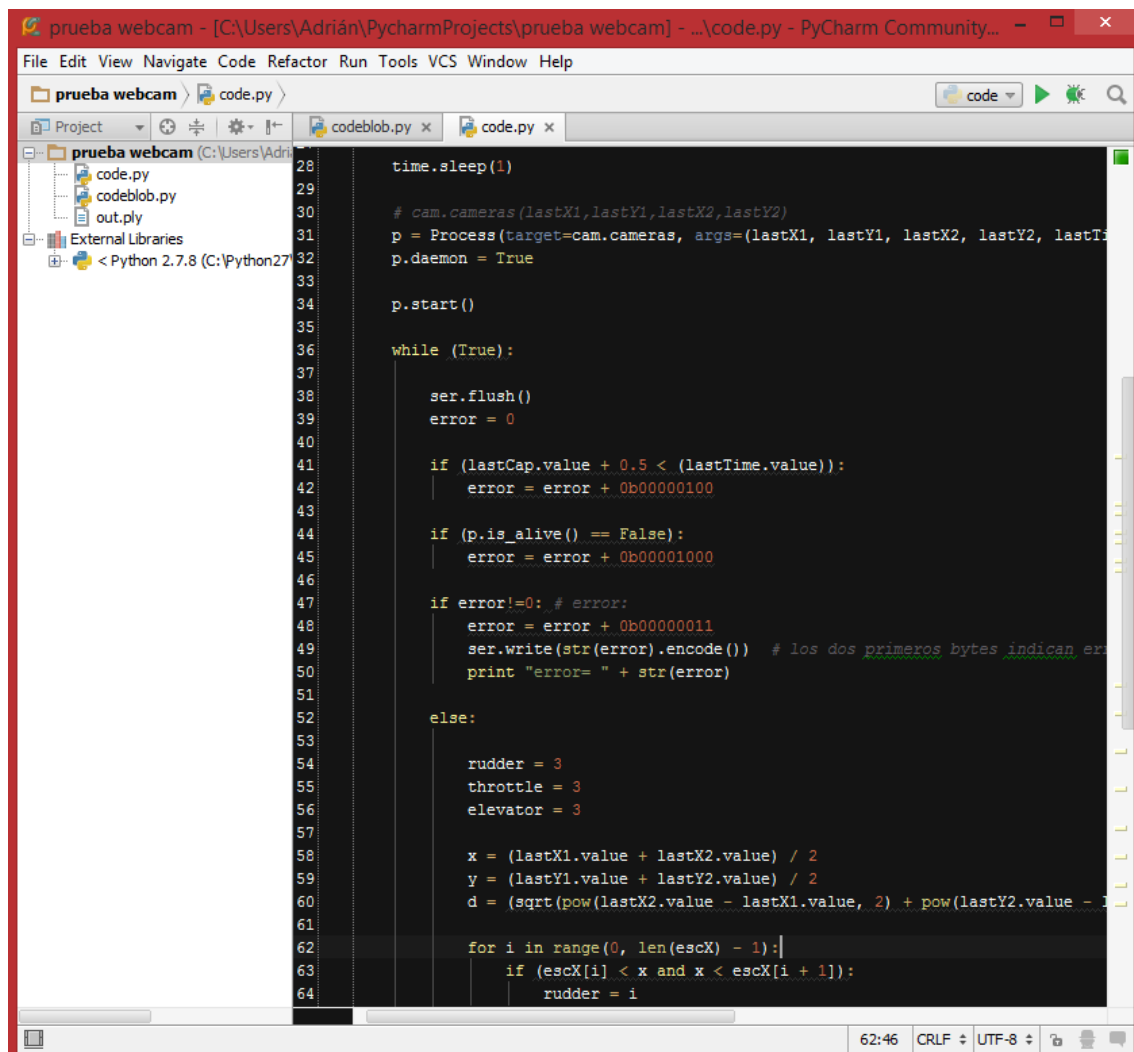


Ilustración 4.31 - PyCharm IDE Arduino

4.7. Software de seguimiento

Empezaremos comentando el algoritmo desarrollado en Python para el Edison (Sección 4.7.1), y a continuación describiremos el algoritmo programado en el Arduino (Sección 4.7.2).

4.7.1. Intel Edison

El algoritmo se divide en dos partes, la detección de la baliza y la comunicación con el Arduino.

El código que controla las cámaras se ejecuta en un hilo individual, lanzado por hilo que se comunica con el Arduino. Este es su pseudocódigo.

```
Inicialización de las cámaras
Inicialización de los parámetros del SimpleBlobDetector

WHILE (True)

    Captura de imagen de las dos cámaras
    Localización de blobs con SimpleBlobDetector
    Búsqueda de paralelogramos en las dos imágenes

    IF En caso de encontrar más de uno por imagen
        Seleccionar el más cercano al anterior

    IF Si se encuentra la baliza en las dos cámaras
        Guardar la posición en las dos cámaras
        Guardar el tiempo de la detección

    Guardar el tiempo de la captura
```

El algoritmo SimpleBlobDetector, realiza el siguiente procesado de las imágenes:

- Convierte la imagen en blanco y negro.
- Busca grupos de píxeles del mismo color y los agrupa en blobs. Un blob es un conjunto de píxeles que comparten alguna característica, en este caso el color.
- Filtra los blobs dependiendo de los parámetros de entrada, forma, color y tamaño.
- Calcula el centro de cada blob y lo devuelve en un vector de coordenadas.

Para simplificar y acelerar el procesado de las imágenes, se ha bajado al mínimo la sensibilidad a la luz de las cámaras. Esto crea imágenes muy oscuras y reduce el tamaño de las zonas brillantes.

El siguiente pseudocódigo es del hilo que se comunica con el Arduino. Para que este hilo tenga acceso a las coordenadas de la baliza, el hilo que controla las cámaras se inicializa con acceso a ciertas variables compartidas.

```

Inicialización de la comunicación serie
Lanzamiento hilo 'cámaras'

WHILE (True)

    Lectura de variables del hilo 'cámaras'

    IF Si se hilo 'cámaras' ha muerto
        Enviar código de error 2

    ELSE IF Si se ha perdido la baliza
        Enviar código de error 1

    ELSE
        Calcular umbral donde está la baliza
        Enviar código de error 0 (Todo OK)
        Enviar posición de la baliza

    Parar hilo 100 milisegundos

```

Para evitar el 'ruido' de pequeños movimientos de la baliza o vibraciones del drone, hemos usado siete umbrales para localizar a la baliza en anchura, altura y profundidad. Movimientos dentro de la zona que establecen estos umbrales no modifican la salida del algoritmo. En la ilustración 4.32, se pueden ver las zonas en las que se divide la imagen capturada por las cámaras, cuanto más se acerque la baliza a los extremos, más rápida será la reacción del drone para seguirla.

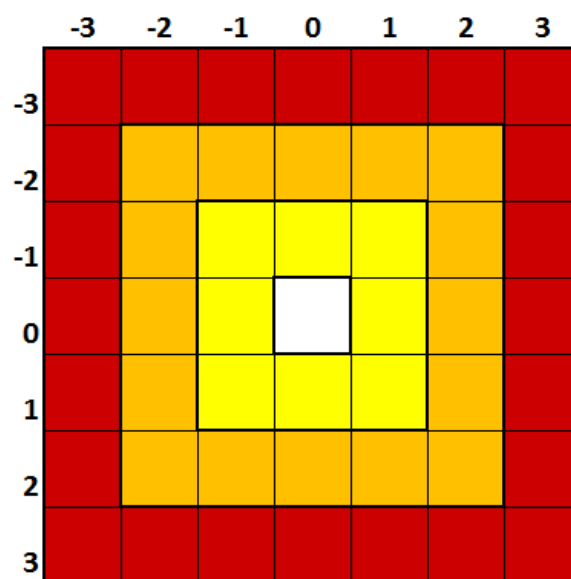


Ilustración 4.32 - Umbral cámaras

4.7.2. Arduino Nano

El código del Arduino también es necesario separarlo en dos. La primera parte es la lectura de la señal PWM que envía la receptora de radio.

La información de una señal PWM se mide por el tiempo que permanece la entrada a un nivel de uno lógico. Para la controladora de vuelo, un señal de PWM de 1500 microsegundos es el valor central en cada canal, 1000 microsegundos es el valor más bajo y 2000 microsegundos el más alto.

Para poder leer un tiempo tan pequeño necesitamos hacerlo mediante interrupciones. Al iniciar el Arduino establecemos en cada canal de entrada una interrupción que se activará cuando ese canal pase de un cero a un uno lógico, rising. Cuando se lance esa interrupción se guarda el tiempo y se activa la interrupción contraria, de uno a cero, falling. Cuando se active la interrupción se resta el tiempo actual al tiempo anterior y se guarda. Este proceso se puede ver más claramente en el siguiente fragmento de pseudocódigo.

```
Interrupción rising del canal X

    Guardar tiempo actual
    Activar interrupción falling del canal X

Interrupción falling del canal X

    Guardar tiempoActual - tiempoRising

    IF Si estamos en modo manual
        Mandar a la controladora la señal

    Activar interrupción rising del canal X
```

Como se puede ver en el código, cuando estamos controlando el dron en modo manual, la señal que llega de la radio se guarda y se reenvía a la controladora de vuelo.

La siguiente parte del software del Arduino es la recepción de los mensajes provenientes del Arduino Edison y control de errores. Para que el usuario pueda ver los errores que se puedan presentar se ha incluido un led en la parte posterior del drone, ilustración 4.24.

```
WHILE (True)

    IF Si hay mensajes pendientes en el puerto serie
        Leer mensaje

        IF Código de estado
            Actualizar variables de estado

            IF Si el código de estado es un error
                Estabilizar el drone

        ELSE IF Código de movimiento
            Calcular señal PWM
            Mandar señal PWM a la controladora

    IF Si se pierde la comunicación con el Edison
        Actualizar variables de estado

    IF Si se está en modo autónomo y existe un error
        Iluminar el led

    IF Si se está en modo manual y baliza detectada
        Iluminar el led
```

Para distinguir entre distintos errores y la detección de la baliza, hacemos que el led se ilumine con las siguientes frecuencias.

- La baliza esta visible en modo manual, 2 veces por segundo.
- La baliza no está visible en modo autónomo, 10 veces por segundo.
- El hilo que controla las cámaras no responde o se ha perdido la comunicación con el Edison, 20 veces por segundo.

5. INTEGRACIÓN Y PRUEBAS

En esta sección vamos a realizar el acoplado entre la implementación hardware y el programa software (Sección 5.1). A continuación, vamos a realizar las pruebas de uso del sistema para comprobar que el diseño es correcto y se han alcanzado los objetivos propuestos (Sección 5.2).

5.1. Integración hardware y software

Para poder realizar las pruebas, primero tenemos que programar los microcontroladores con los algoritmos desarrollados.

La programación del Arduino Nano, es muy sencilla basta con pulsar el botón de programar del IDE. El IDE verifica el código, lo compila y programa el microcontrolador de forma transparente para el usuario.

El caso del Intel Edison es distinto, aunque igualmente sencillo. Intel proporciona un ejecutable que instala los drivers necesarios en el ordenador, actualiza el sistema operativo del Edison y da a elegir distintos entornos de programación específicos para su microprocesador.

Una vez configurado todo desde el lanzador de Intel, podemos conectarnos por Wifi al punto de acceso que crea el Edison. Python viene preinstalado, solo tenemos que instalar OpenCV a través del administrador de paquetes de Yoctolinux.

Para eliminar la necesidad de conectarse al Edison para lanzar el programa de seguimiento, hemos creado una tarea que se lanza después del arranque del microcontrolador.

5.2. Pruebas

En la primera prueba que se realiza nos encontramos con el inconveniente que solo una de las cámaras enciende. En el anexo se incluye un video mostrando el resultado del algoritmo de seguimiento de la baliza.

Durante la depuración del error, nos encontramos con el error “*VIDIOC_STREAMON: No space left on device*”. Buscando la fuente del error, encontramos este hilo [55], indican que el error proviene de los drivers de la webcam. Dada la falta de drivers específicos para la mayoría de webcams del mercado, muchas distribuciones de Linux incluyen drivers genéricos, las cámaras funcionan, pero muchas de las opciones de configuración no están disponibles. En nuestro caso, falta la opción de usar video comprimido, al enviarse el video sin compresión se satura el ancho de banda del USB con solo una webcam.

Las dos únicas soluciones posibles para poder hacer las pruebas del prototipo son: (1) la búsqueda de un modelo de webcam con drivers que permitan el uso de dos webcams simultáneas; o (2) usar el ordenador para suplantar el papel del Intel Edison.

Durante la búsqueda de cámaras, no encontramos ninguna a la venta que no dependa de los drivers genéricos de Linux.

Por el motivo anterior, la única salida es conectar el ordenador al dron con un alargador USB. Por supuesto, el resto de componentes no se modifican. Esta solución inicial de compromiso nos permitirá comprobar si el prototipo es capaz de realizar el seguimiento.

Una vez solucionado el problema anterior, nos enfrentamos a otra casuística: cuando la baliza se acerca o aleja del dron, éste se mueve para mantener una distancia aproximada de dos metros. Al realizar este movimiento, el dron se inclina y el algoritmo de seguimiento interpreta que la baliza se está moviendo. Al corregir el movimiento, la baliza sale del campo de visión de las cámaras, avisando de esta situación.

Esta casuística si es más fácil de salvar. Basta modificar el algoritmo de seguimiento para contemplar este escenario, o modificar la estructura para que las cámaras se mantengan en posición horizontal por su propio peso, como un péndulo.

Otra modificación que consideramos es la de aumentar la velocidad de movimiento del dron cuando la baliza está cerca de los bordes del campo de visión de las cámaras. La reacción actual es lineal, la velocidad de giro aumenta al mismo ritmo que se aleja la baliza del centro, cuando debería ser exponencial.

A pesar de todos estos obstáculos encontrados, el último día dedicado a las pruebas se alcanzó con éxito el objetivo de hacer volar el dron, así como conseguir que siga a la baliza. La tensión del cable que une el dron al ordenador hace perder estabilidad a este último, y sin embargo aun así conseguimos realizar un pequeño video, donde se ve al dron siguiendo la baliza mientras el alumno tiene las manos fuera del mando.

6. CONCLUSIONES Y TRABAJO FUTURO

Este ha sido un proyecto muy ambicioso y que he disfrutado realizando. Por una parte está la pasión por el mundo de los multirrotores, y por otra la adquisición de un gran conocimiento en áreas que resultan interesantes como son visión por ordenador, electrónica o inteligencia artificial.

Para concluir este documento, haremos mención a futuras mejoras para actualizar el prototipo (Sección 6.1), y la viabilidad de creación de una empresa de diseño y montaje de drones de acuerdo a los conocimientos y experiencia adquiridos durante el desarrollo del presente trabajo (Sección 6.2).

6.1. Futuras mejoras para el prototipo

A continuación vamos a enumerar posibles mejoras para el prototipo:

- Usar leds infrarrojos y cámaras sin filtro de infrarrojos, para mejorar la detección en exteriores con mucha iluminación.
- Uso de cables personalizados para reducir peso y mejora visual.
- Diseño de una carcasa para proteger, embellecer y organizar todo el interior.
- Añadir un sensor de distancia en la parte inferior para ayudar a mantener la altura de forma más estable.

6.2. Creación de una empresa

Dada la innovación de este proyecto y la falta de productos en el mercado con fines similares. La idea de montar una empresa a partir del prototipo no es una idea descabellada.

La única competencia actual proviene de los drones mencionados en el capítulo 2. Lily y Hexo+, que recordemos que tiene un precio muy alto, 999\$ y 1299\$ respectivamente.

Entre los tres drones no hay ventajas claras en la forma de seguimiento, aunque nuestro prototipo funcionara mejor en interiores y zonas con poca

cobertura GPS. Lily y Hexo+ tiene la ventaja ante que inserten ruido al video, luces muy fuertes o al amanecer o anochecer. Por último, la resistencia al agua de Lily es una gran gaza a su favor.

Para que el prototipo se convierta en una solución real y competitiva es necesario realizar las mejoras definidas en la sección 6.1, además de diseñar un microcontrolador que realice el trabajo del Arduino Nano e Intel Edison y unas cámaras personalizadas. Una placa FPGA [56] puede ser una buena opción.

El diseño de microcontroladores y cámaras propias incrementaría el tiempo de desarrollo del producto. El precio dependerá de la solución final deseada por el usuario, si bien es necesario recordar que sólo los microcontroladores y cámaras usadas en el prototipo tienen un coste superior a los 140€

Descripción de los anexos

Junto a esta memoria se incluyen los siguientes anexos:

- Videos, los videos realizados durante las pruebas. Debido a que algunas de las pruebas se realizaron dentro de las instalaciones de la Universidad de Jaén, no podemos distribuir estos últimos públicamente. En la defensa se mostrará un resumen de estos videos.
- Imágenes, las ilustraciones incluidas en la memoria con la resolución original y sin recortes.
- Código fuente, el código fuente de los algoritmos desarrollados. El mismo que se ha usado durante las pruebas del prototipo.

Bibliografía

- [1] «Drone Wiki,» [En línea]. Available: http://es.wikipedia.org/wiki/Veh%C3%ADculo_a%C3%A9reo_no_tripulado. [Último acceso: Mayo 2015].
- [2] «BBC Military Drones,» [En línea]. Available: <http://www.bbc.com/news/world-south-asia-10713898>. [Último acceso: Mayo 2015].
- [3] «Amazon Prime Air,» [En línea]. Available: <http://www.amazon.com/b?node=8037720011>. [Último acceso: Mayo 2015].
- [4] «Quadcopter,» [En línea]. Available: <http://en.wikipedia.org/wiki/Quadcopter>. [Último acceso: Mayo 2015].
- [5] S. P. y. C. M. Robins, Administración. Octava edición, Pearson Educación, 2005.
- [6] «Dji Naza,» [En línea]. Available: <http://www.dji.com/product/naza-m-v2>. [Último acceso: Mayo 2015].
- [7] «Lily,» [En línea]. Available: <https://www.lily.camera/>. [Último acceso: Mayo 2015].
- [8] «Hexo+,» [En línea]. Available: <https://hexoplus.com/>. [Último acceso: Mayo 2015].
- [9] «Gimbal,» [En línea]. Available: <http://en.wikipedia.org/wiki/Gimbal>. [Último acceso: Mayo 2015].
- [10] «GoPro,» [En línea]. Available: <http://es.gopro.com/>. [Último acceso: Mayo 2015].
- [11] «Kickstarter Hexo+,» [En línea]. Available: <https://www.kickstarter.com/projects/sqdr/hexo-your-autonomous-aerial-camera/description>. [Último acceso: Mayo 2015].
- [12] R. Stephens, Beginning Software Engineering, Wrox, 2015.
- [13] J. M. Angulo Usategui, S. Romero Yesa y I. Angulo Martínez, Microcontroladores PIC: diseño práctico de aplicaciones. Segunda parte: PIC16F87X, PIC18FXXXX. (2a. ed.), McGraw-Hill España, 2006.
- [14] «Foro rcgroups,» [En línea]. Available: <http://www.rcgroups.com>. [Último acceso: Mayo 2015].
- [15] «Foro multicopters,» [En línea]. Available: <http://www.multicopters.es>. [Último acceso: Mayo 2015].

- [16] «Foro aeromodelismo,» [En línea]. Available: <http://www.foro-aeromodelismo.com/>. [Último acceso: Mayo 2015].
- [17] «Hobbyking KK2.1HC,» [En línea]. Available: http://www.hobbyking.com/hobbyking/store/__58031__Hobbyking_KK2_1HC_Multi_Rotor_Hard_Case_Flight_Control_Board_With_Remote_Programmer_EU_Warehouse_.html. [Último acceso: Mayo 2015].
- [18] «Acelerometro wiki,» [En línea]. Available: <http://en.wikipedia.org/wiki/Accelerometer>. [Último acceso: Mayo 2015].
- [19] «Giróscopo wiki,» [En línea]. Available: <http://es.wikipedia.org/wiki/Gir%C3%B3scopo>. [Último acceso: Mayo 2015].
- [20] «Motores sin escobillas,» [En línea]. Available: <http://www.quadruino.com/guia-2/materiales-necesarios-1/motores-brushless>. [Último acceso: Mayo 2015].
- [21] «DJI E300,» [En línea]. Available: <http://www.dji.com/product/e300>. [Último acceso: Mayo 2015].
- [22] «ESC,» [En línea]. Available: http://en.wikipedia.org/wiki/Electronic_speed_control.
- [23] «Frame F450 DJI,» [En línea]. Available: http://electronicarc.com/catalogo/product_info.php?cPath=107_129&products_id=670&osCsid=af597a6815211dd10733d681b14e93b3. [Último acceso: Mayo 2015].
- [24] «Tren de aterrizaje DJI,» [En línea]. Available: http://electronicarc.com/catalogo/product_info.php?cPath=107_125&products_id=1090&osCsid=af597a6815211dd10733d681b14e93b3. [Último acceso: Mayo 2015].
- [25] «UBEC 5A HobbyKing,» [En línea]. Available: http://www.hobbyking.com/hobbyking/store/uh_viewItem.asp?idProduct=36305. [Último acceso: Mayo 2015].
- [26] «Rx Tx Turnigy 9X,» [En línea]. Available: https://www.hobbyking.com/hobbyking/store/__8991__Turnigy_9X_9Ch_Transmitter_w_Module_8ch_Receiver_Mode_1_v2_Firmware_.html.
- [27] «Batería 3S Turnigy,» [En línea]. Available: https://www.hobbyking.com/hobbyking/store/__35834__Turnigy_5000mAh_3S_20C_Lipo_Pack_EU_Warehouse_.html. [Último acceso: Mayo 2015].
- [28] «Bolsa ignífuga HobbyKing,» [En línea]. Available: https://www.hobbyking.com/hobbyking/store/__51450__Heat_Reflective_LiPoly_Storage_Bag_120x70x210mm_.html. [Último acceso: Mayo 2015].

- [29] «Cargador DJI Phantom,» [En línea]. Available: http://electronicarc.com/catalogo/product_info.php?products_id=975&osCsid=d1e9f18386e05d49ee5cab4786ff1354. [Último acceso: Mayo 2015].
- [30] «Adaptador HobbyKing,» [En línea]. Available: https://www.hobbyking.com/hobbyking/store/__16373__HXT_4mm_to_XT_60_Battery_Adapter_2pcs_bag_.html. [Último acceso: Mayo 2015].
- [31] «Avisador batería baja,» [En línea]. Available: https://www.hobbyking.com/hobbyking/store/__18588__HobbyKing_8482_Cell_Checker_with_Low_Voltage_Alarm_2S_8S_.html. [Último acceso: Mayo 2015].
- [32] «Bluetooth,» [En línea]. Available: <http://www.bluetooth.com/Pages/Bluetooth-Home.aspx>. [Último acceso: Mayo 2015].
- [33] «QR code,» [En línea]. Available: http://en.wikipedia.org/wiki/QR_code. [Último acceso: Mayo 2015].
- [34] «Led wiki,» [En línea]. Available: <http://es.wikipedia.org/wiki/Led>. [Último acceso: Mayo 2015].
- [35] «Paralelogramo wiki,» [En línea]. Available: <http://es.wikipedia.org/wiki/Paralelogramo>. [Último acceso: Mayo 2015].
- [36] A. G. I. B. J. L. L. M. M. D. R. Jorge García, «People Detection and Tracking Based on Stereovision and Kalman Filter,» *Revista Iberoamericana de Automática e Informática Industrial RIAI*, vol. 9, nº 4, 2012.
- [37] «Raspberry Pi,» [En línea]. Available: <https://www.raspberrypi.org/>. [Último acceso: Mayo 2015].
- [38] «Intel Edison,» [En línea]. Available: <http://www.intel.es/content/www/es/es/do-it-yourself/edison.html>. [Último acceso: Mayo 2015].
- [39] «Placa Base SparkFun,» [En línea]. Available: <http://www.arduino.cc/en/pmwiki.php?n=Main/ArduinoBoardNano>. [Último acceso: Mayo 2015].
- [40] «Wearables,» [En línea]. Available: <http://www.quees.info/que-es-wearable.html>. [Último acceso: Mayo 2015].
- [41] «Raspbian,» [En línea]. Available: <https://www.raspbian.org/>. [Último acceso: Mayo 2015].
- [42] «Yoctolinux,» [En línea]. Available: <https://www.yoctoproject.org/>. [Último acceso: Mayo 2015].

- [43] «Arduino Nano,» [En línea]. Available: <http://www.arduino.cc/en/pmwiki.php?n=Main/ArduinoBoardNano>. [Último acceso: Mayo 2015].
- [44] «Open Source wiki,» [En línea]. Available: http://es.wikipedia.org/wiki/C%C3%B3digo_abierto. [Último acceso: Mayo 2015].
- [45] «Webcam C170 Logitech,» [En línea]. Available: <http://www.logitech.com/es-es/product/webcam-c170>. [Último acceso: Mayo 2015].
- [46] «Meccano,» [En línea]. Available: <http://www.meccano.com/es/>. [Último acceso: Mayo 2015].
- [47] «Explicación self-level (vídeo),» [En línea]. Available: <https://www.youtube.com/watch?v=f0Bh3fMGHbM>. [Último acceso: Mayo 2015].
- [48] «Explicacion canales,» [En línea]. Available: <http://vueloartificial.com/introduccion/toma-de-contacto/principios-basicos-de-vuelo/>.
- [49] G. C. Hillar, Redes: diseño, actualización y reparación, Editorial Hispano Americana HASA, 2009.
- [50] «OpenCV,» [En línea]. Available: <http://opencv.org/>. [Último acceso: Mayo 2015].
- [51] «C++,» [En línea]. Available: <http://www.cplusplus.com/>. [Último acceso: Mayo 2015].
- [52] «Python,» [En línea]. Available: <https://www.python.org/>. [Último acceso: Mayo 2015].
- [53] «Java,» [En línea]. Available: <https://java.com/es/>. [Último acceso: Mayo 2015].
- [54] «PyCharm,» [En línea]. Available: <https://www.jetbrains.com/pycharm/>. [Último acceso: Mayo 2015].