Escola Universitària
Politècnica de Mataró

Centre adscrit a:

UNIVERSITAT POLITÈCNICA
DE CATALUNYA

Grado de Ingeniería Informática

Control del dron a través de la plataforma Mocap

Memoria

Miguel Angel Callejón Bosque
PONENT: Josep López Xarbau

PRIMAVERA 2017

TecnoCampus
Mataró-Maresme

Dedicatoria

A mis padres por todo el apoyo durante este tiempo

Agradecimientos

A mi tutor Josep López, por su ayuda incondicional.

A mis padres por su apoyo en momentos complicados del proyecto.

Y por ultimo…a mis compañeros durante estos cuatro años.

Resumen

El objetivo del proyecto consiste en desarrollar un software para poder controlar el dron a través de la plataforma mocap instalada en la universidad. Mediante el servidor del mocap se recibirán las coordenadas del actor y/o de los cuerpos rígidos y el software será capaz de procesarlas para enviarle a través del módulo de radio frecuencia las ordenes de movimiento necesarias para que el dron llegue a las coordenadas correctas. También el software dispone de una interfaz gráfica para dirigirlo sin la necesidad de utilizar el mando radio control.

Resum

L'objectiu del projecte consisteix a desenvolupar un software per poder controlar el dron a través de la plataforma mocap instal·lada a la universitat. Mitjançant el servidor del mocap es rebran les coordenades de l'actor i/o dels cossos rígids i el software serà capaç de processar-les per enviar-li a través del mòdul de ràdio freqüència les ordres de moviment necessàries perquè el dron arribi a les coordenades correctes. També el programari disposa d'una interfície gràfica per dirigir-ho sense la necessitat d'utilitzar el comandament radio control.

Abstract

The aim of this project is to develop a software that is able to control a dron through the mocap platform installed in the University. The coordinates of the actor and/or the rigid bodies will be received by the mocap server and the software will be able to process them in order to send the movement instructions needed for the dron to reach the correct coordinates through the radio frequency module. In addition, the software has a graphical interface to direct it without the need of using the remote radio control.

Índice.

Índice de figuras.

Índice de tablas.

Glosario de términos.

Streaming	Sistema de difusión de datos
Dron	Vehículo aéreo no tripulado
Mocap	Captura de movimiento (Motion Capture)
PID	Mecanismo de control por realimentación
RF	Radiofrecuencia
RFID	Identificación por radiofrecuencia
IDE	Entorno de desarrollo integrado (Integrated Development Environment)
TFG	Trabajo final de Grado

1. Objetivos.

1.1. Propósito.

El propósito de este proyecto, de final de grado, es la creación de una herramienta que permita la interacción del movimiento de una persona con un dron a través de una plataforma Mocap.

1.2. Finalidad.

Dirigir el dron mediante la captura de movimiento de la plataforma mocap con los movimientos de un actor. De esta forma se evitaría la utilización del mando radio control para controlar la dirección del dron.

1.3. Objeto.

Crear un software en phyton para controlar la posición y el movimiento del dron. El programa hará de intermediario y recibirá las coordenadas tanto del actor como del dron. Las coordenadas serán recibidas via streaming desde la plataforma MOCAP.

1.4. Alcance.

Con el software creado, el actor mediante sus movimientos conseguirá mover el dron sin la utilización del mando radio control del dron.

2. Introducción.

El siguiente proyecto se realiza como trabajo final de grado de Ingeniería Informática del Tecnocampus Mataró-Maresme.

El motivo de escoger esta temática para el desarrollo del proyecto se debe a la investigación sobre nuevas tecnologías que pueden ser vinculadas con la rama de la informática.

Día a día es más habitual ver como se introducen conceptos nuevos sobre los drones y parece ser que, en un futuro no muy lejano, tendrán un papel importante en muchos hábitos de la rutina diaria del ser humano.

El resultado final del proyecto tiene un objetivo concreto, permitir al usuario final poder controlar el dron mediante a través de la plataforma Mocap o poder controlarlo mediante un software diseño con interfaz gráfica.

2.1. ¿Qué es un Dron?

Un dron es un vehículo volador no tripulado y puede ser controlado por sistemas de control remoto desde la tierra. Existe una gran variedad de tamaños, formas y con una gran diversidad de funciones. Debido a la amplia gama, actualmente no hay un uso especifico diseñado para un dron.

Una de las características mas destacadas es que son fáciles de controlar, prácticamente no requieren de combustibles para sus tareas y no ponen en peligro la vida de quien lo controla.

Es muy frecuente su utilización en los servicios militares ya que pueden volar a grandes alturas y así evitar ser detectado. Otras tareas pueden ser la de rastreo en superficies devastadas por desastres naturales, rescates, análisis del tiempo, etc.

Figura 2.1. Modelos de drones

2.2. Principales usos del Dron

Los usos principales de los drones pueden ser los siguientes:

- Proporcionar soporte inteligente y táctico
- Comprobar las bombas o dispositivos peligrosos en carreteras y áreas de tierra
- Observar el tráfico y el comportamiento público
- Proporcionar soporte aéreo
- Seguir o atacar objetivos sospechosos
- Delivery
- Incendios forestales
- Realizar grabaciones de vídeo

Además, son efectivos para la búsqueda de personas, ya que la posibilidad de volar a poca altura en zonas dificultosas o aisladas junto con una cámara permite el reconocimiento de personas perdidas en bosques o montañas.

2.3. ¿Qué es Mocap?

Mocap, también conocido en inglés como motion capture, es la técnica de grabación del movimiento de actores y/o cuerpos rígidos para transferirlo a un modelo digital. El movimiento es representado por los cambios de traslación y rotación que a su vez hace posible calcular velocidades y aceleraciones para su posterior utilización. La tecnología de captura de movimientos surgió en biomecánica, pero también se utiliza principalmente en el campo de los videojuegos o en la industria del cine.

El sistema se compone del reconocimiento óptico de la posición en el espacio (coordenadas X Y Z) de los marcadores puestos sobre el traje del actor, también cuenta con un numero de cámaras infrarrojas distribuidas por el escenario de acción que son capaces de reconocer un mismo punto y deducir la posición mediante triangulación.

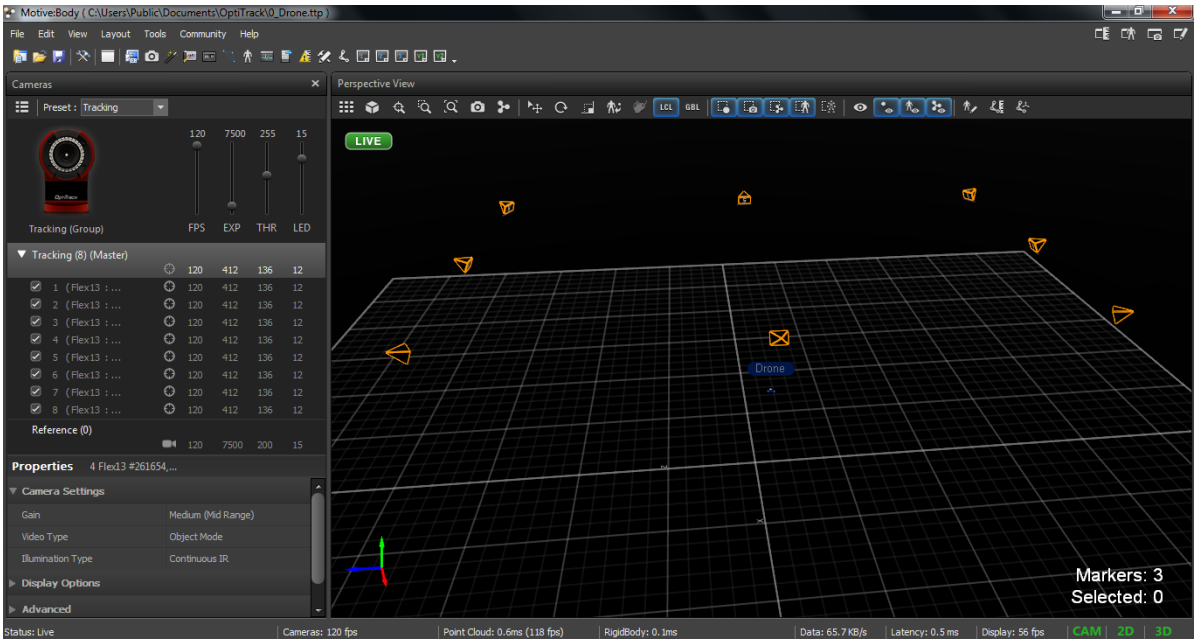

Figura 2.2. Ejemplo software Motive

Existen diferentes tipos de captura de movimientos:

Captura de movimientos óptica pasiva: Esta técnica utiliza marcadores retro reflectantes que son seguidos por las cámaras infrarrojas.

Captura de movimientos óptica activa: Esta técnica utiliza marcadores LEDs conectados por cables al traje de captura de movimientos.

Captura de movimientos en vídeo o Markerless: Esta técnica no requiere de marcadores, en su lugar se basa en el software para rastrear el movimiento de los objetos o actores.

Captura de movimientos inercial: Esta técnica no requiere cámaras excepto como herramienta de localización. El sujeto lleva los sensores y los datos de dichos sensores se transmiten de forma inalámbrica a un ordenador.

La captura de movimientos se puede aplicar para los siguientes campos.

- En el mundo de los **videojuegos**, se utiliza para dotar a los personajes u objetos de animaciones más realistas o también se utiliza para interactuar con el propio juego.

- En la **animación**, se utiliza para crear personajes con movimientos y expresiones corporales más cercanos al mundo real.

- En la **industria militar**, se utiliza para permitir a los militares interaccionar con más partes de su cuerpo, agilizando las maniobras en combate, las cuales requieren velocidad y destreza.

- En **medicina** y **deportes**, se utiliza para analizar la locomoción. Sirven tanto para estudiar problemas físicos como para mejorar el rendimiento deportivo.

A continuación se presentan las principales ventajas e inconvenientes en las cuales el sistemas Mocap puede representar una alternativa a lo tradicional.

Ventajas:

- Reduce los costes de desarrollo de animaciones frente a las técnicas clásicas de creación frame a frame.
- Permite un mayor número de pruebas o variaciones en busca de la escena final.
- Se recrean fácilmente movimientos complejos y realistas.

Desventajas:

- Se necesita software y hardware específico para obtener y procesar los datos.
- El coste del equipamiento y del personal especializado es alto.
- El sistema de captura puede requerir configuraciones específicas para el escenario en el que es utilizado.
- La captura en personajes cuadrúpedos puede llegar a ser muy complicada.
- Los movimientos que no siguen las leyes de la física generalmente no pueden ser capturados.

2.4. ¿Qué es Arduino?

Arduino es una placa basada en un microcontrolador, es decir, se trata de una plataforma de hardware libre. Tiene un circuito integrado en el que se pueden grabar instrucciones programadas utilizando un lenguaje de programación y así permitir al usuario crear programas para interactuar con los circuitos electrónicos. El lenguaje utilizado es un lenguaje propio basado en lenguaje de alto nivel similar a C++.

Es una placa impresa con los componentes para que funcione el microcontrolador y se pueda comunicar con un ordenador a través de la comunicación serial.

El arduino cuenta con una interfaz de entrada y otra de salida.

- **Interfaz de entrada:** Su objetivo principal es trasladar la información al microcontrolador, que es el encargado de procesar los datos.
- **Interfaz de salida:** Su objetivo principal es trasladar la información procesada a los periféricos para utilizar los datos.

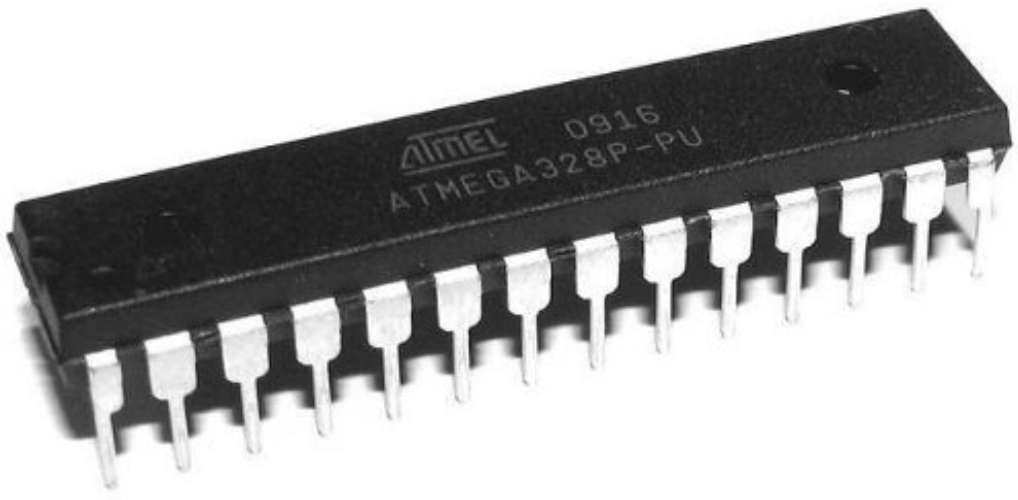

Figura 2.3. Microcontrolador Atmel utilizado por Arduino

2.5. ¿Qué es un módulo de radiofrecuencia?

Un módulo de radiofrecuencia (RF) es un microchip y una antena asociada capaz de enviar y recibir una señal RF. El módulo recibe información de una fuente externa y la envía a un dispositivo RFID.

Un dispositivo RFID (Radio Frequency Identification) es una tecnología de identificación remota e inalámbrica en la cual un dispositivo lector vinculado se comunica a través de una antena con un transponder (tag/Etiqueta) mediante ondas de radio.

Para establecer la comunicación es necesario que cada uno de los dispositivos implicados incorporen una antena RF.

Banda de frecuencias	**Descripción**	**Rango**
125 kHz	LF (Baja frecuencia)	Hasta 50 cm
13,56 MHz	HF (Alta frecuencia)	De 8 cm
400 MHz – 1.000 MHz	UHF (Ultra alta frecuencia)	De 3 a 10 m
2,45 GHz – 5,4 GHz	Microondas	Más de 10 m

Tabla 2.1 Bandas de frecuencia de los sistemas RFID

3. Objetivo.

3.1. Objetivos del proyecto

El objetivo de este proyecto es el desarrollo de un software que permita controlar el dron a través de la captura de movimientos de un actor y/o cuerpo rígido mediante la plataforma mocap instalada en la universidad Tecnocampus de Mataró.

Para poder llevar a cabo el proyecto, previamente se tendrá que hacer un análisis de los requisitos que estarán implicados en todas las fases del proyecto. La primera fase consta de hacer una búsqueda para valorar los diferentes modelos del dron y de sistemas de radio frecuencia. La segunda fase constará de implementar todas las partes de la comunicación entre el ordenador, el módulo de radio frecuencia, el sistema mocap y el dron.

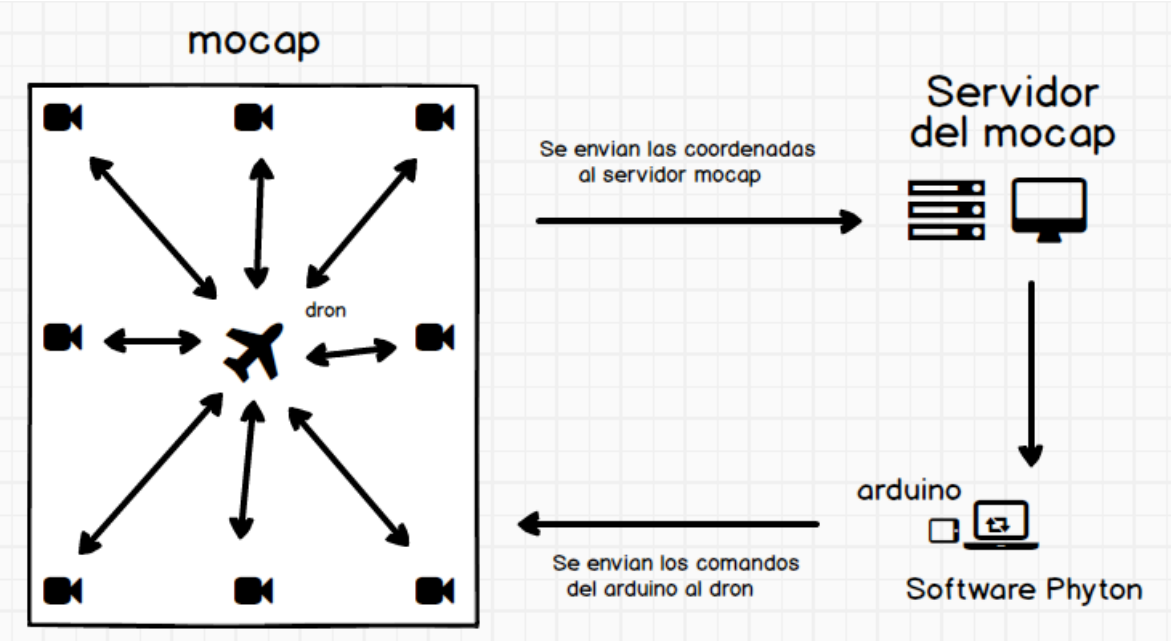

Figura 3.1. Esquema de flujo de información

El software cuenta con una interfaz gráfica en la cual el usuario final deberá escoger el puerto serie para la comunicación con el módulo de radio frecuencia y elegir la ip del servidor del mocap para poder recibir las coordenadas.

También cuenta con otra opción alternativa, controlar el dron a través de la propia interfaz gráfica mediante el uso de botones para dirigir la dirección del dron.

El funcionamiento interno del software se basa en la recepción de las coordenadas del dron y del actor que serán emitidas por los marcadores colocados en ambos objetos y a su vez serán capturadas por las cámaras infrarrojas de la plataforma mocap. Una vez el servidor mocap ha recibido las coordenadas, las envía al ordenador del cual el usuario final se ha conectado al servidor y el software del proyecto procesará las coordenadas tanto del dron como del actor. Mediante una placa arduino conectada a un módulo de radio frecuencia se enviará la coordenada en la que el dron se tiene que dirigir. De esta forma el software simula el comportamiento del mando radiocontrol del dron.

4. Estado del arte

El estado del arte describe la situación de una determinada tecnología que permite definir cómo ha evolucionado el tema, su estado actual y cual es su evolución.

Una primera fase de investigación sirvió para tener el primer acercamiento a otros proyectos con funcionalidades similares. El hecho de conocer otros proyectos y obtener una idea general de las funcionalidades existentes, sirvió para poner en claro las ideas propuestas y a la vez ayudar de forma específica al enfoque propuesto.

La segunda fase permitió leer, analizar y clasificar mas específicamente toda la información obtenida dentro del marco del proyecto. A partir de aquí, se seleccionó la información fundamental que permitió la ampliación del proyecto.

La investigación permitió desarrollar un proyecto con funcionalidades diferentes a las investigadas, como por ejemplo permitir el control del dron desde una misma aplicación a través de una interfaz gráfica mediante botones o con la conexión al servidor Mocap y procesando las coordenadas. También sirvió a la hora de decidir el modelo de dron más adecuado junto al módulo de radio frecuencia.

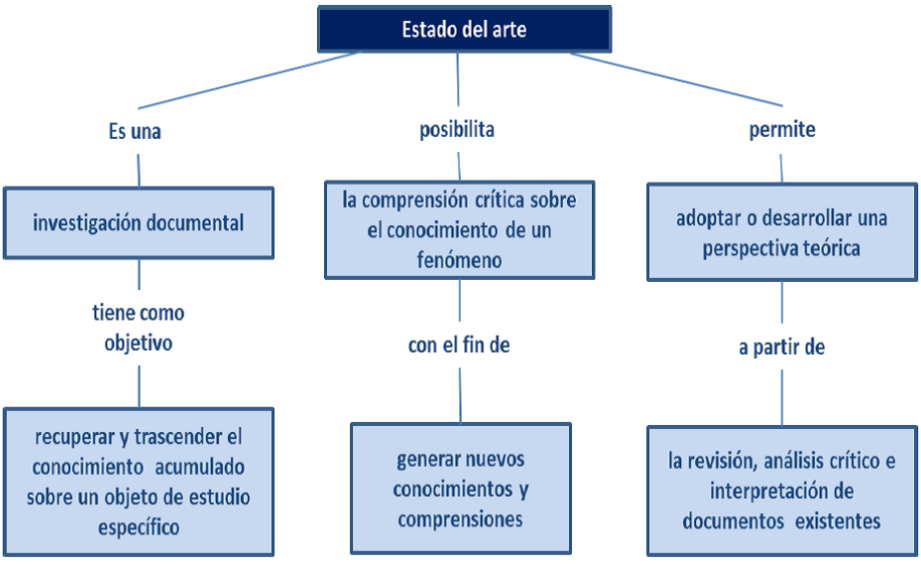

Figura 4.1. Estado del arte

5. Estudio de requisitos

Tras la fase de investigación en el estado del arte, se compararon las diferentes posibilidades para el desarrollo del proyecto tanto a nivel de software como de hardware para el desarrollo del proyecto. Se ha intentado escoger el material mas económico o el software libre para ahorrar los costes del proyecto.

5.1. Evaluación del dron

Después de ver los proyectos similares que ya habían sido realizados, la mayoría había escogido el modelo Hubsan X4 H107C. Tras comparar varios modelos por foros y paginas especializadas en drones sobre sus características y el precio, el mas adecuado para el proyecto era el propio Hubsan X4 H107C. Para escoger el dron, también se pidió información en tiendas físicas especializadas y todos los resultados llevaron a la misma elección. El modelo Hubsan cumplía uno de los requisitos fundamentales y era que tenia que ser compatible con Arduino.

El Hubsan X4 H107C es un dron para personas que se están iniciando, es muy resistente a los golpes y a las caídas. Cuenta con la posibilidad de poder comprar recambios de todas sus piezas. Su autonomía es de unos 7 minutos volando aproximadamente y se pueden intercambiar la batería sin dificultad alguna.

Figura 5.1. Dron Hubsan 4X H107C

Dimensiones	5.8 x 5.8 x 3 centímetros
Alcance	30 metros
Peso	50 gramos
Velocidad	44 km/h
Altura máxima	90 metros
GPS	No
Cámara	640 x 480 píxeles
Compatible con GoPro	No
Controlar con el móvil	No
Batería Extra	Si
Recambios	Si

Tabla 5.1. Especificaciones técnicas del Hubsan X4 H107C

5.2. Evaluación del sistema de radio frecuencia

Para la realización del circuito de conexión entre el dron y la placa arduino, se optó por el módulo de radio frecuencia A7105. Una de las razones principales fue por el hecho de tener disponibles de otros proyectos las librerías del módulo para poder cargarlas en el arduino y comunicarse con el dron. El único inconveniente fue que no estaba a la venta en España y se tuvo que comprar a China, esto llevó una espera de unas semanas.

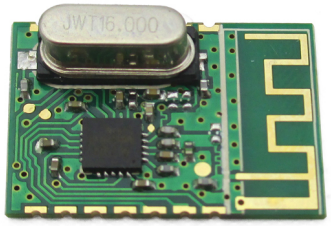

Figura 5.2. Circuito de radio frecuencia A7105

Ítem	**Especificación**
Alimentación	2.5V~3.6V
Frecuencia	400 – 2483 MHz
Sensibilidad Rx	105 dBm @ 10Kbps modo, Dev = 40 KHz 100 dBm @ 250Kbps modo, Dev = 93 KHz 96 dBm@ 500Kbps modo, Dev = 186 KHz
Modulación	FSK or GFSK
Distancia de transmisión	100 metros

Interfaz	8pin 2.54 mm
Dimensiones	23.3mm(L) x 12.5mm(W) x 5mm(H)
Temperatura de funcionamiento	-40 ~ 85ºC

Tabla 5.2. Especificaciones del circuito A7105

5.3. Evaluación del microcontrolador

Tras ver las especificaciones de los diferentes modelos de microcontrolador, se escogió Arduino porque es una plataforma de prototipos de electrónica de código abierto basada en hardware y software fácil de utilizar. Los microcontroladores Arduino son mas baratos a comparación con otras marcas, es multiplataforma, su entorno de programación es simple y claro y es de código abierto para software y hardware.

Dentro de las posibilidades que ofrecía Arduino, se escogió el modelo Arduino Uno ya que es el modelo más económico y cumplía con los requisitos necesarios.

El microcontrolador Arduino Uno cuenta con 14 I/O digitales, 6 entradas analógicas, 6 salidas PWR, 1 UART, 32kb de memoria y su precio ronda los 20€.

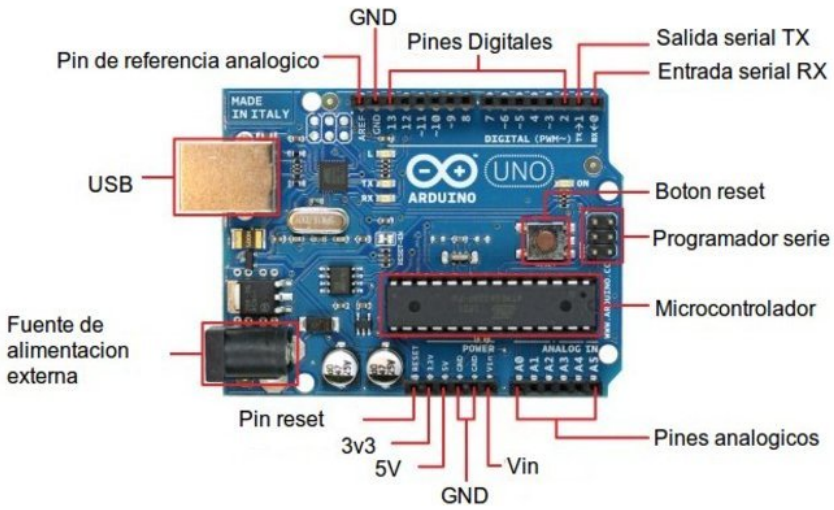

Figura 5.3. Arduino Uno

	Arduino Uno	**Arduino Mega2560**	**Arduino Leonardo**	**Arduino Due**
Microcontrolador	ATmega 328	ATmega 2560	ATmega32u4	AT91SAM3X8E
Puertos digitales	14	54	20	54
Puertos PWM	6	15	7	12
Puertos analógicos	6	16	12	12
Memoria	32k	256k	32k	512k
Clock	16 Mhz	16 Mhz	16 Mhz	84 Mhz
Conexión	USB	USB	Micro USB	Micro USB
Tensión	5v	5v	5v	3.3v

Tabla 5.3. Comparativa microcontroladores Arduino

6. Circuito de conexión

Para lograr la comunicación entre el Arduino Uno y el módulo de radio frecuencia A1705 se ha creado un circuito de conexión.

6.1. Módulo de radio frecuencia

El módulo de radio frecuencia utilizado es el módulo A1705 y a continuación se puede ver la composición de los pins para conectarlo a el Arduino a través del circuito impreso.

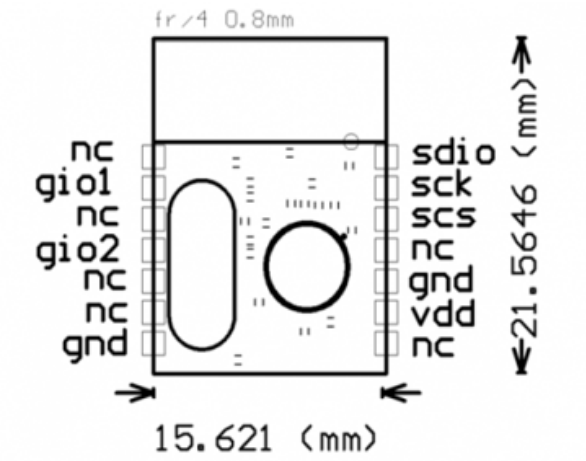

Figura 6.1. Encapsulado del módulo RF A1705

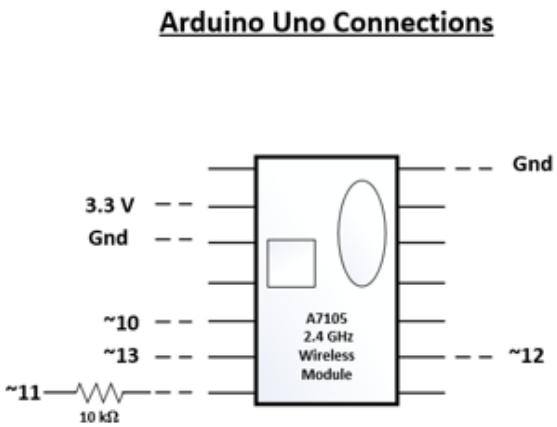

Figura 6.2. Configuración de los pins de conexión del módulo RF A1705

Símbolos	Descripción
GND	Ground
VDD	RF Module Supply Voltage Input
NC	No Connection
SCS	SPI Chip Selection
SCK	SPI Clock
SDIO	SPI DATA I/O
GIO1	General Purpose I/01
GIO2	General Purpose I/02

Tabla 6.1. Descripción de los símbolos del módulo RF A1705

6.2. Esquema del circuito

En la imagen posterior se puede ver el esquema del circuito para conectar el módulo RF A1705 con el Arduino.

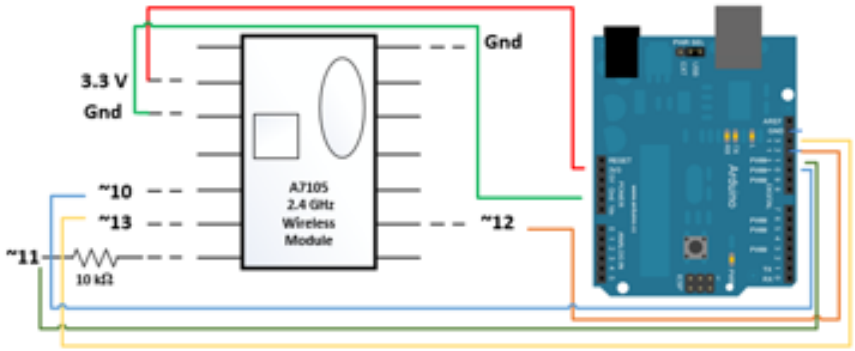

Figura 6.3. Esquema de conexión del módulo RF A1705 con el arduino

Para finalizar una vez identificados los pines en el módulo RF y en el Arduino, conectamos ambas partes mediante cables que hacen de puente tal y como se muestra en la imagen superior del esquema del circuito.

Para la fabricación del circuito se ha seguido el siguiente orden:

1. Identificar los 6 pines en el módulo RF A7105 que se necesitan
2. Asignar a la placa arduino los pines de conexión
3. Siguiendo el esquema de conexionado se prueba la conexión del arduino con lel módulo RF A1705
4. Se substituye los cables por la placa electrónica.

Los componentes necesarios para crear el circuito han sido:

- Módulo de radio frecuencia
- Resistencias
- Placa electrónica
- Equipo de soldar

Creando el circuito impreso, se evita el uso de una Breadboard y de cables para hacer de puente.

De esta forma se reemplaza el mando radio control del Dron por la Arduino. Por lo tanto, durante el proyecto se utilizará la Arduino codificada para enviar las instrucciones de transmisión que luego serán enviadas de forma inalámbrica al Dron.

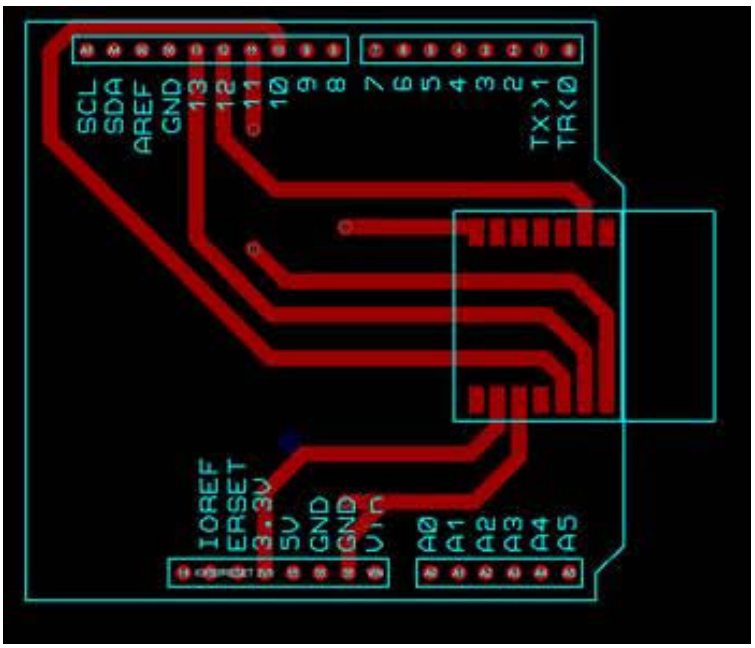

Figura 6.4. Esquema del circuito

7. Entorno de desarrollo

En un primer momento se decidió desarrollar el software en lenguaje Java. A lo largo del grado, Java ha sido el lenguaje mas estudiado y practicado, de ahí el motivo de la elección. Sin embargo, durante el desarrollo del programa se comprobó que Java no era compatible con las librerías del software de la plataforma Mocap y se buscó una alternativa, en este caso Python.

Python utiliza una sintaxis intuitiva y es un lenguaje orientado a objetos. A su vez es compatible para trabajar con librerías de escritura y lectura sobre el puerto Serie, requisito principal para poder desarrollar el proyecto. Otro requisito era la compatibilidad de poder crear una interfaz gráfica para el usuario.

Por otra parte, la herramienta elegida para la programación del software fue Eclipse con la versión Neon . Es una plataforma de software compuesta por un conjunto de herramientas de programación de código abierto y es compatible con diversas plataformas. Eclipse permite tener los proyectos estructurados en forma de árbol. Como se puede ver a continuación, el proyecto es el paquete principal y dentro se puede dividir por paquetes y dada paquete puede tener sus clases. También te permite visualizar las clases importadas al proyecto. Dependiendo del sistema operativo en el que esté instalado Eclipse, pueden variar las librerías utilizadas para desarrollar el programa.

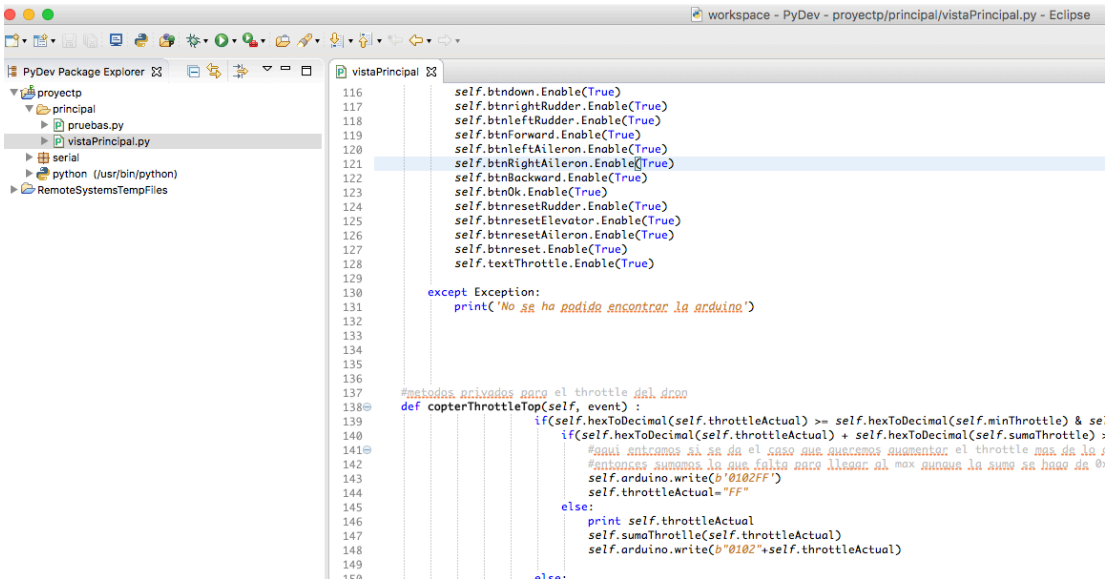

Figura 7.1. Vista general del entorno de desarrollo Eclipse

8. Arduino IDE

Arduino es una plataforma de hardware libre basada en una placa con un micro controlador y un entorno de desarrollo (IDE).Además facilita un entorno de desarrollo que consta de un editor de código, un compilador, un depurador y un constructor de interfaz gráfica.

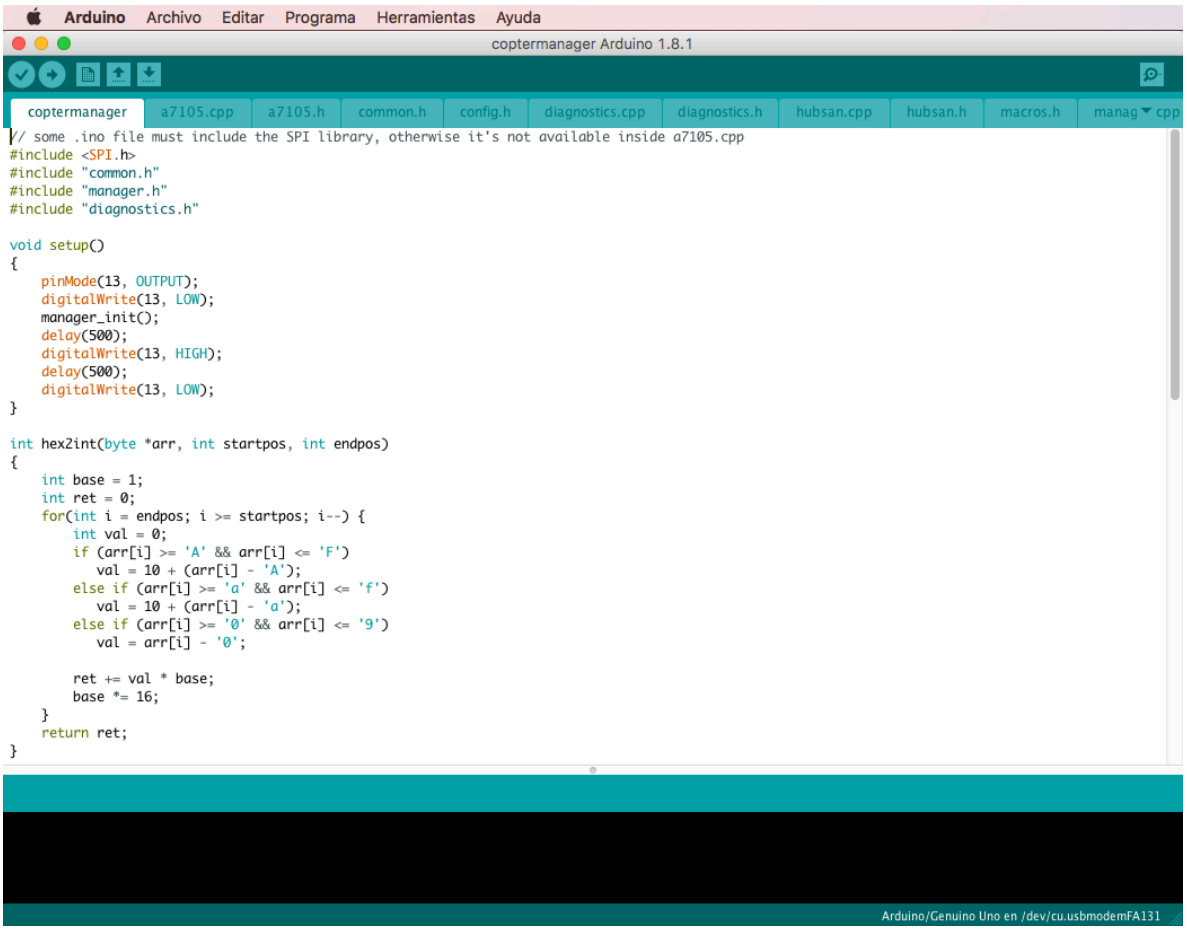

Figura 8.1. Vista general del entorno de desarrollo de Arduino

Los entornos en los que se puede aplicar Arduino son:

- Robótica
- IoT
- Impresoras 3D
- Otros

Para el desarrollo del proyecto, se han aprovechado las librerías de Arduino modificando el código de otros proyectos ya implementados referente al módulo de radio frecuencia A7105. La adaptación ha sido necesaria para probar la comunicación desde el entorno de desarrollo de Arduino y el Dron.

Para configurar el entorno de programación se han de realizar los siguientes pasos:

- Se selecciona Herramientas → Placa: "Arduino/Genuino Uno" → Arduino/Genuino Uno, en este punto, dependiendo la placa utilizada se seleccionara un modelo u otro.

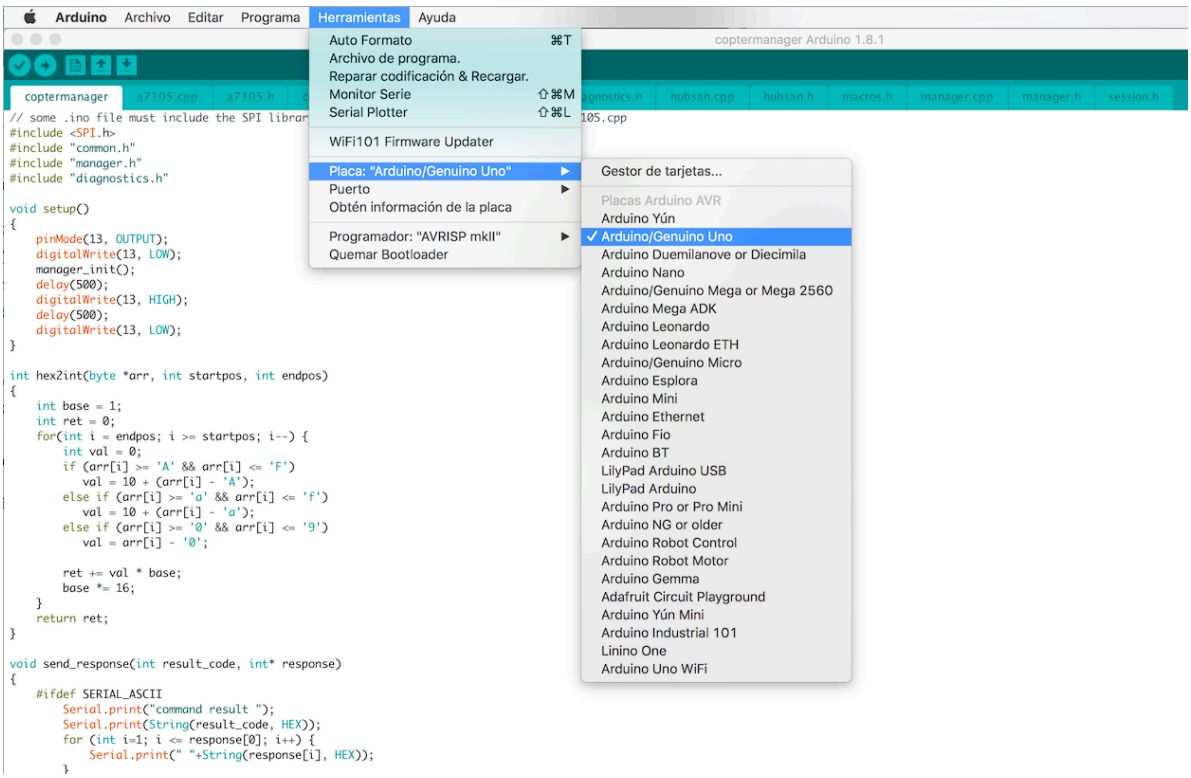

Figura 8.2. Selección de la placa en el IDE Arduino

- A continuación se selecciona Herramientas → Puerto: "/Dev/cu.usbmodel141111(Arduino/Genuino Uno)" , en este punto, dependiendo el puerto utilizado se seleccionara un puerto u otro.

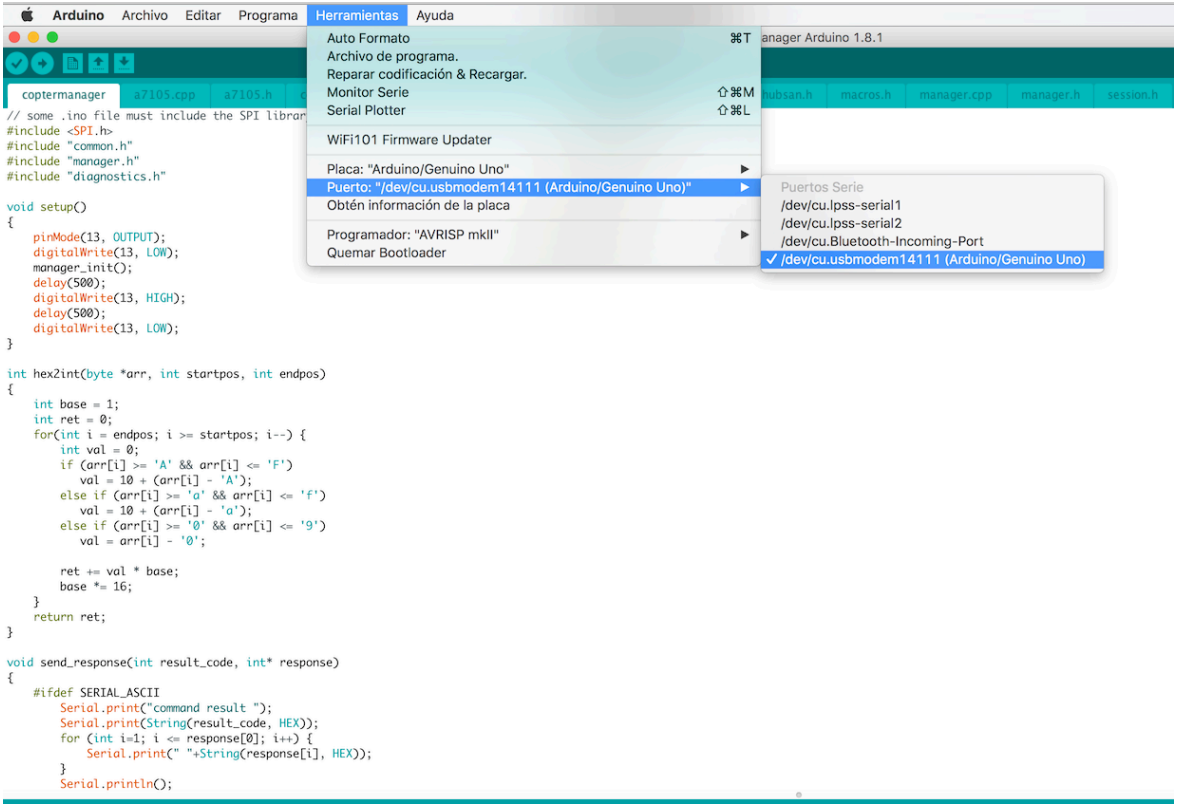

Figura 8.3. Selección del puerto en el IDE Arduino

Una vez se han cargado las librerías en la Arduino, mediante los siguientes comandos se puede establecer la conexión con el Dron y enviarle por el puerto serie conectado a la placa Arduino una serie de instrucciones para que el Dron las ejecute.

Nombre del comando	Código	Valor del comando	Descripción
COPTER_BIND	0x01	Tipo de helicóptero	Inicia el proceso de emparejamiento con el dron.
COPTER_THROTTLE	0x02	Valor del throttle (Rango 0x00 – 0xFF)	Ajusta el valor del acelerador (arriba / abajo)
COPTER_RUDDER	0x03	Valor del rudder (Rango 0x34 – 0xCC)	Ajusta el valor del timón (rotación izquierda / derecha)
COPTER_AILERON	0x04	Valor del aileron (Rango 0x45 – 0xC3)	Ajusta el valor del alerón (desviación izquierda / derecha)
COPTER_ELEVATOR	0x05	Valor del elevator (Rango 0x3E – 0xBC)	Ajusta el valor del elevador (adelante / atrás)
COPTER_LED	0x06	Activado o desactivado (1 o 0)	Leds activados/desactivados
COPTER_FLIP	0x07	Activado o desactivado (1 o 0)	Flip activado/desactivado
COPTER_VIDEO	0x08	Activado o desactivado (1 o 0)	Video activado/desactivado
COPTER_GETSTATE	0x09	-	Obtiene el estado del emparejamiento

COPTER_EMERGENCY	0x0A	-	Activar el modo de emergencia, todos los valores se ajustan a los valores predeterminados.
COPTER_DISCONNECT	0x0B	-	Desconectar el dron
COPTER_LISTCOPTERS	0x0C	-	Obtiene la lista de drones conectados

Tabla 8.1. Lista de comandos del módulo de radio frecuencia A7105

Nombre del comando	Código	Descripción
PROTOCOL_OK	0x00	Ok
PROTOCOL_UNBOUND	0xE0	El Dron no está unido
PROTOCOL_BOUND	0xE1	El dron está unido
PROTOCOL_INVALID_COPTER_TYPE	0xF0	Tipo de Dron no válido
PROTOCOL_ALL_SLOTS_FULL	0xF1	El rango disponible de Drones está

		completo.
PROTOCOL_INVALID_SLOT	0xF2	Dron in no válido
PROTOCOL_VALUE_OUT_OF_RANGE	0xF3	Valor fuera de rango
PROTOCOL_EMERGENCY_MODE_ON	0xF4	Si el modo de emergencia está activado, sólo se admite el comando de desconexión
PROTOCOL_UNKNOWN_COMMAND	0xF5	Comando desconocido

Tabla 8.2. Lista de códigos de respuesta del módulo de radio frecuencia

En el siguiente ejemplo se sincroniza el dron a través del IDE de Arduino con la placa fabricada.

- Una vez se han cargado las librerías del módulo A1705 en el arduino, abrimos una consola del IDE Arduino (Monitor serie). Se puede comprobar que la inicialización ha sido satisfactoria, esto quiere decir que las librerías han sido cargadas correctamente. El siguiente paso será mandarle el código "0101" que significa que al dron id = 01 se le envía el código 01. El código 01 como se puede ver en la tabla anterior, significa el inicio del proceso de emparejamiento con el dron.

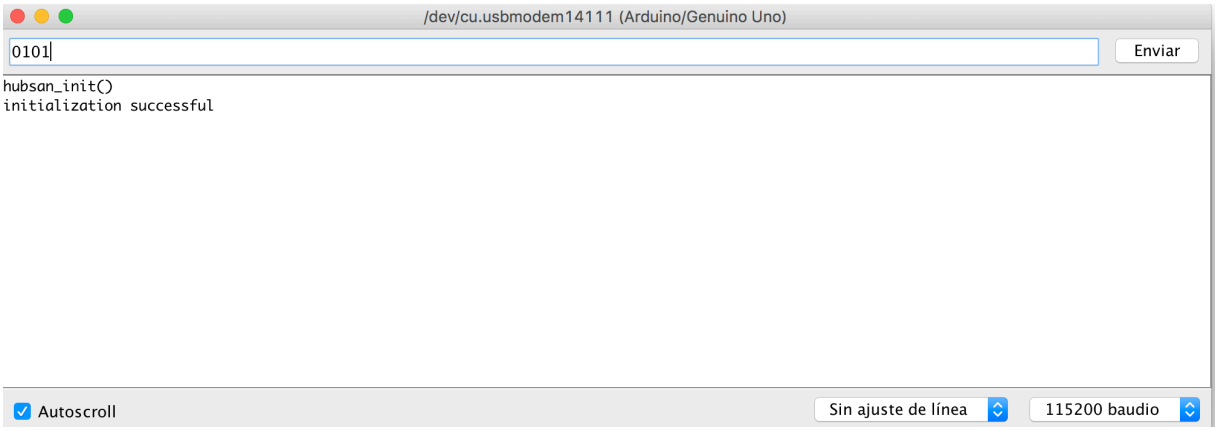

Figura 8.4. Emparejamiento del dron desde el IDE Arduino

- Una vez se ha emparejado el dron nos devuelve como resultado "copter bound", quiere decir que ya hemos emparejado el dron correctamente.

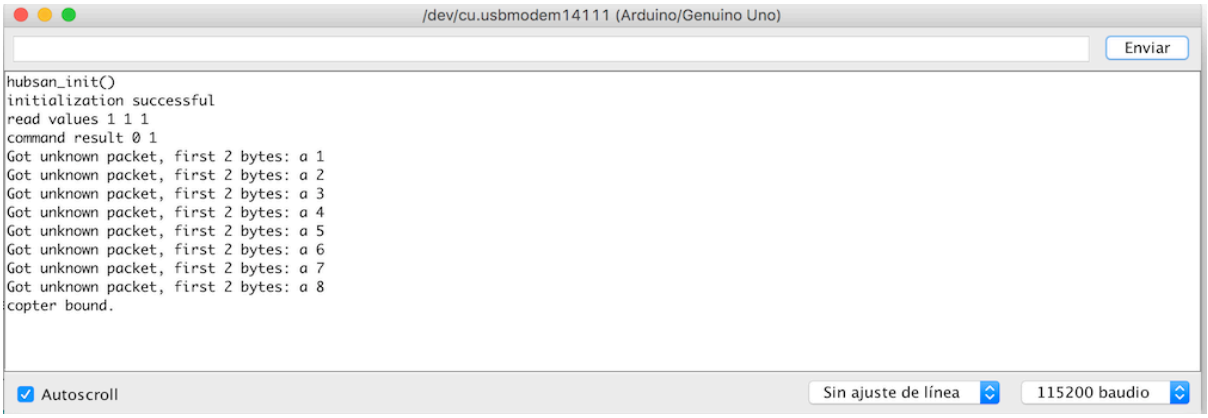

Figura 8.5. Confirmación del emparejamiento del dron desde el IDE Arduino

9. Entorno del software Motive

El software Motive se trata del programa que utiliza la plataforma mocap. Una vez ejecutamos el programa, podemos ver un aspecto general de la vista.

- Como se observa, en el panel izquierdo se muestran las configuraciones de las cámaras infrarrojas del sistema. En el panel central se puede ver la superficie del mocap, en este caso no se ve nada por que no se han seleccionado los marcadores del objeto. Por ultimo, en el panel derecho se puede configurar el calibrado del sistema.

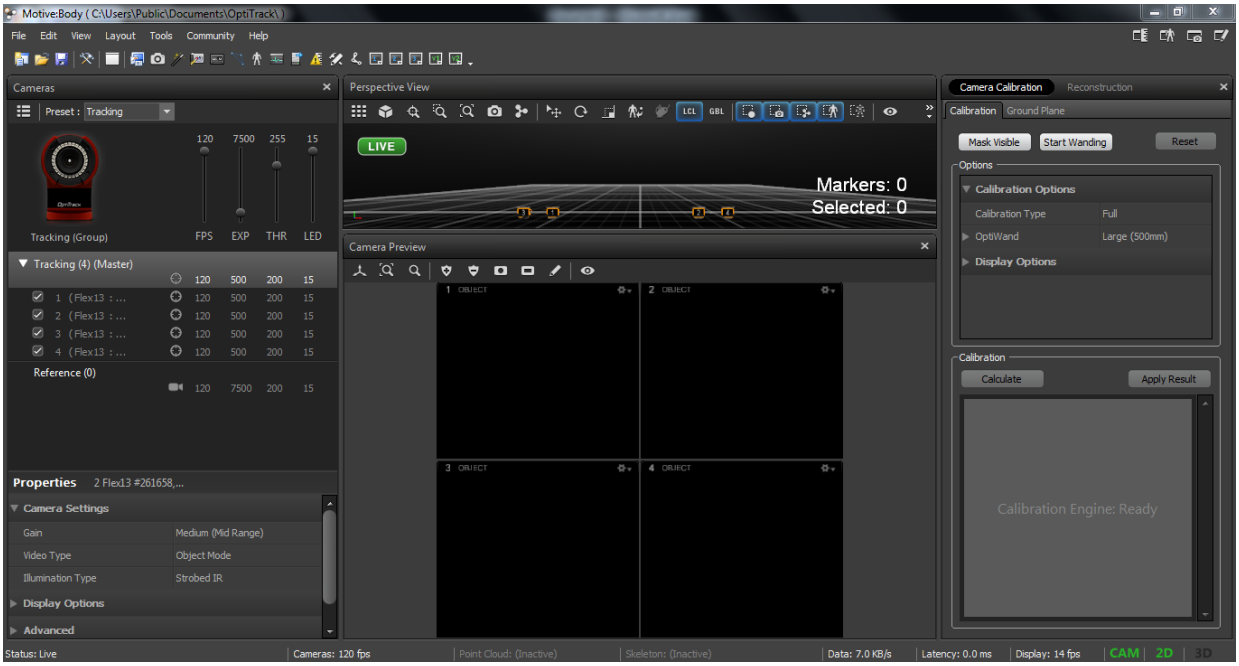

Figura 9.1. Ventana principal del Motive

- A continuación se puede ver como aparecen las 8 cámaras infrarrojas conectadas y el dron en medio de la superficie con los 3 marcadores emitiendo las coordenadas.

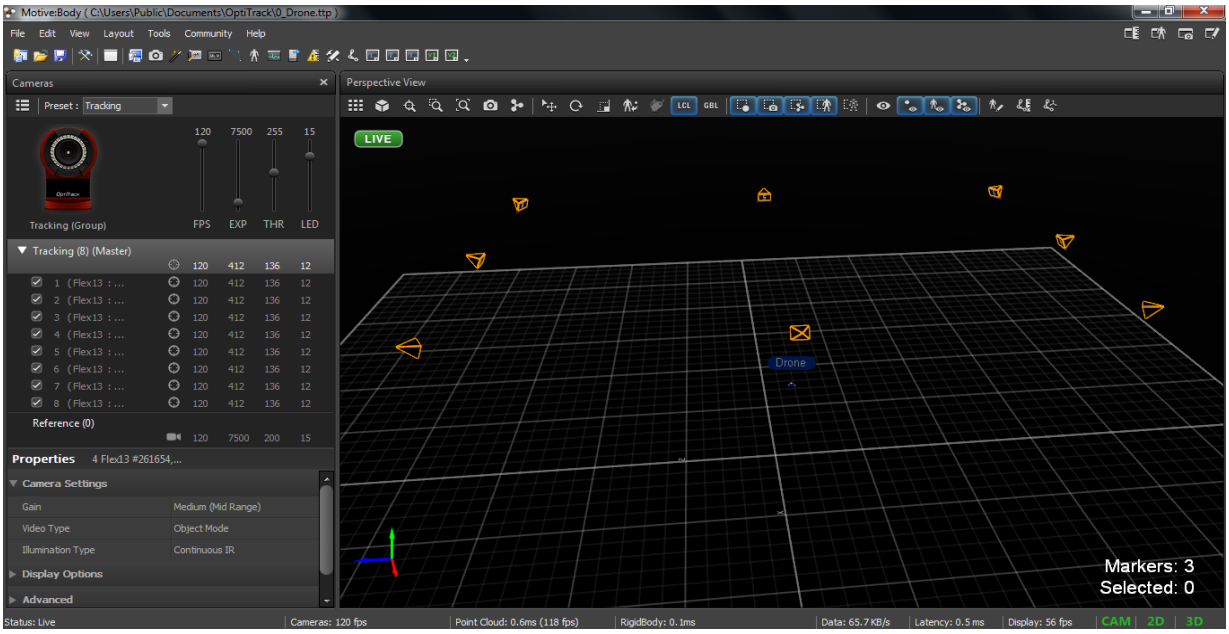

Figura 9.2. Vista de la superficie del mocap

- Para configurar el streaming de datos seleccionamos en la barra del menú 'view' → 'Data Streaming'

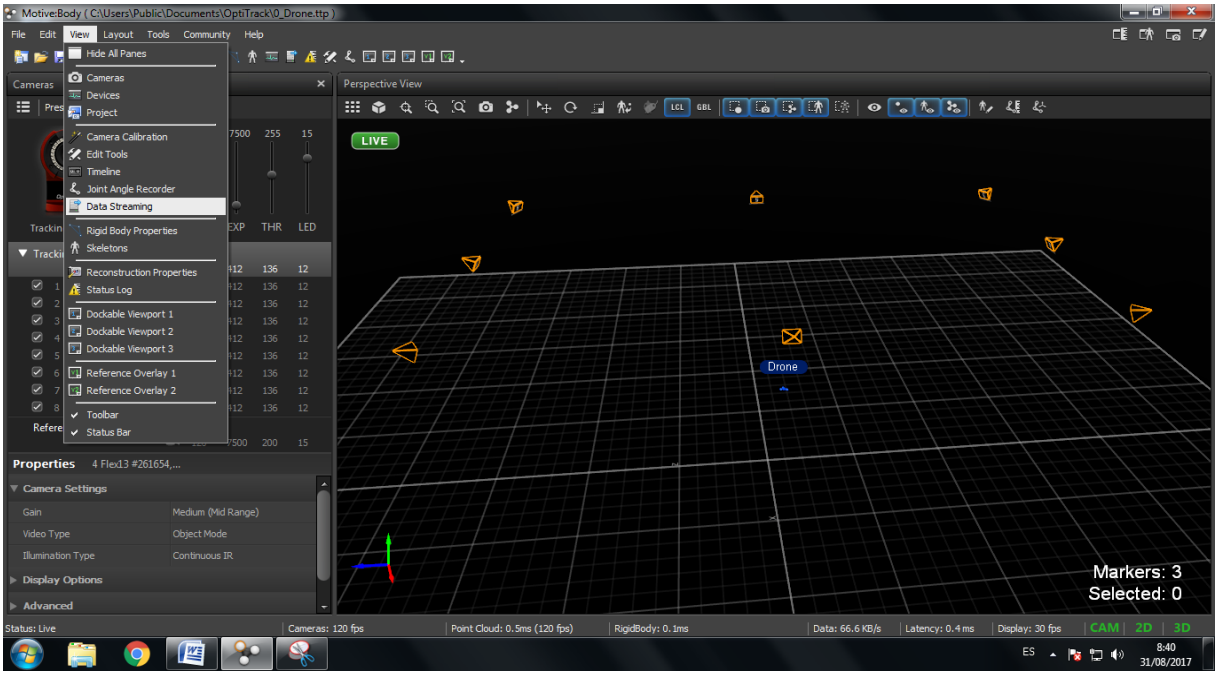

Figura 9.3. Configuración del streaming en Motive

- A continuación, en el panel derecho aparecen los parámetros para configurar el streaming de datos. Pudiendo elegir el tipo de emisión Unicast/Multicast, el puerto de envío de datos, la IP del servidor, etc. Una vez hemos configurado los parámetros, seleccionamos la opción 'Broadcast Frame Data' y el sistema mocap comenzará el envió de las coordenadas.

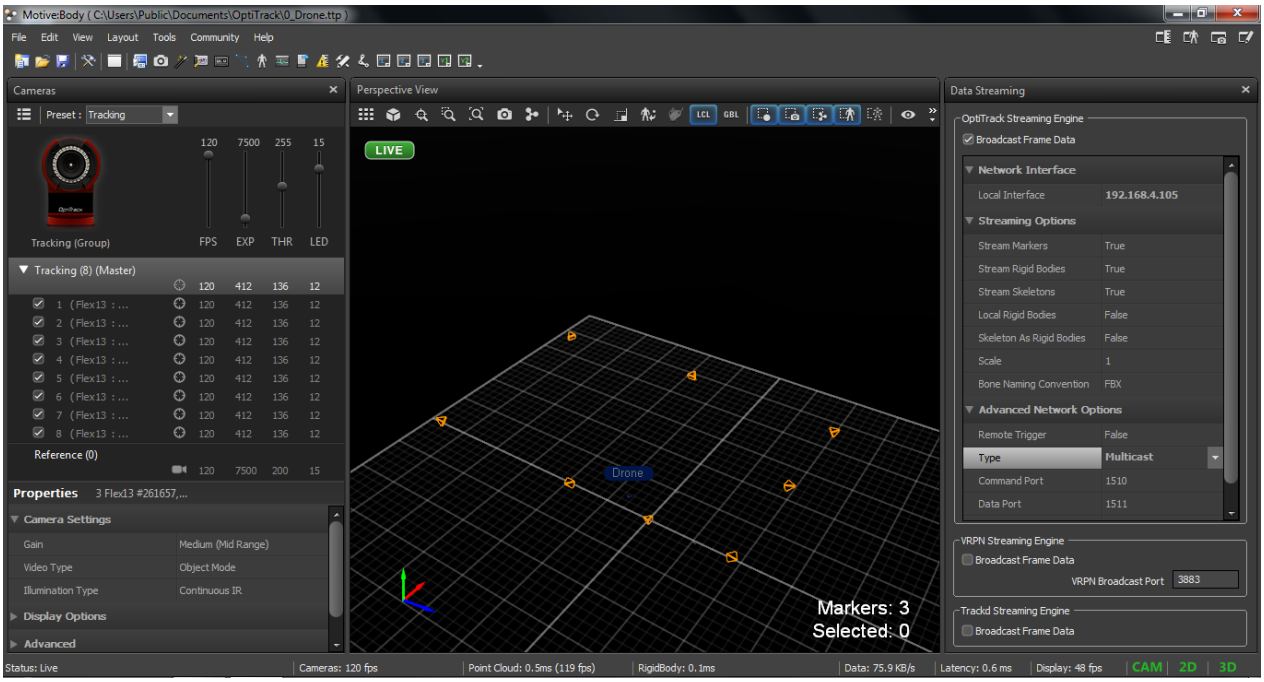

Figura 9.4. Configuración de los parámetros del streaming

- Para finalizar, una vez el servidor está emitiendo datos, podemos añadir un nuevo panel donde nos sale la posición y la orientación del dron recibida por las cámaras.

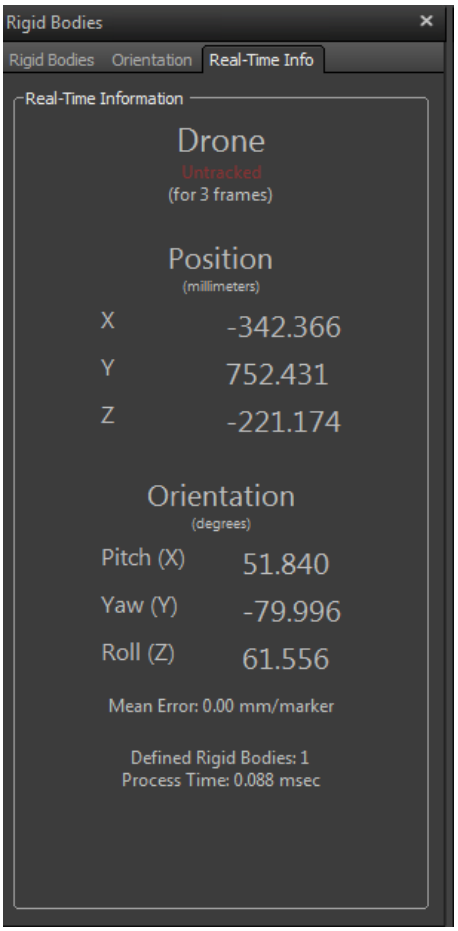

Figura 9.5. Panel de datos del streaming

10. Diseño del software implementado

El software tiene implementado dos entornos de funcionalidad diferentes. Por un lado podemos controlar el dron mediante los botones de la interfaz gráfica, y por el otro lado se puede introducir la IP del servidor del Mocap para procesar los datos y enviarle al dron las ordenes necesarias para dirigirlo a la posición elegida.

A continuación se explica el funcionamiento:

- Al iniciar el programa, se debe seleccionar del desplegable el puerto serie que queremos utilizar para el envío de datos.

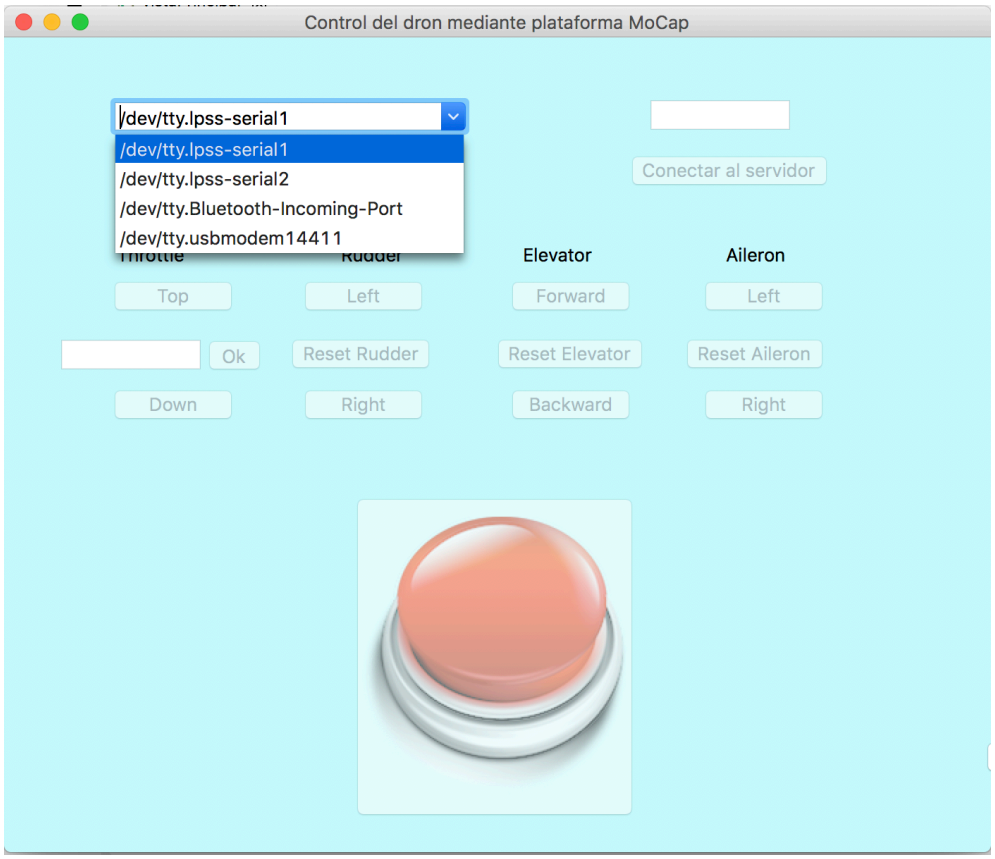

Figura 10.1. Vista principal del software implementado

- Una vez se ha abierto el puerto serie sin problemas, se habilitan los botones del panel. En la parte superior derecha se puede introducir la dirección IP del servidor mocap al que nos queremos conectar, si el valor de la ip es null devuelve error. En el panel central hay una serie de botones en la que permite controlar el dron simulando los botones del mando radio control. Los botones permiten subir y bajar el acelerador, girar hacia los lados y dirigirse hacia delante o hacia atrás.

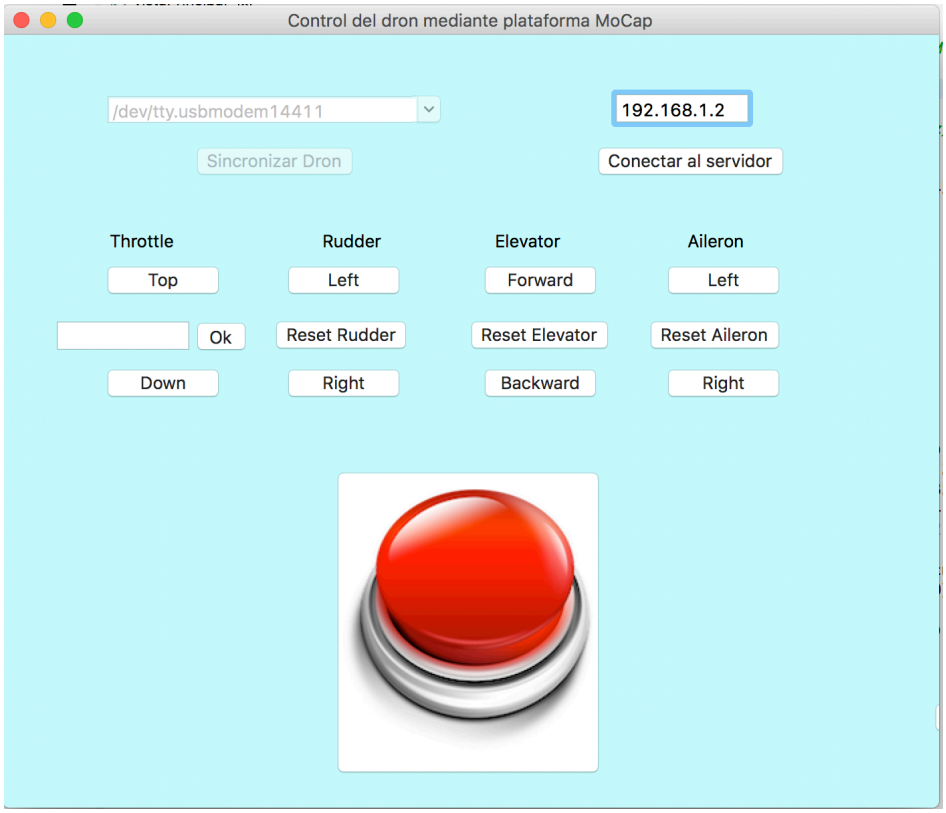

Figura 10.2. Vista principal del software implementado

- El botón situado en la parte inferior, es el encargado del paro de emergencia. Si el usuario necesita reiniciar los valores del dron, al pulsarlo automáticamente el dron volverá a su estado inicial con los valores reseteados.

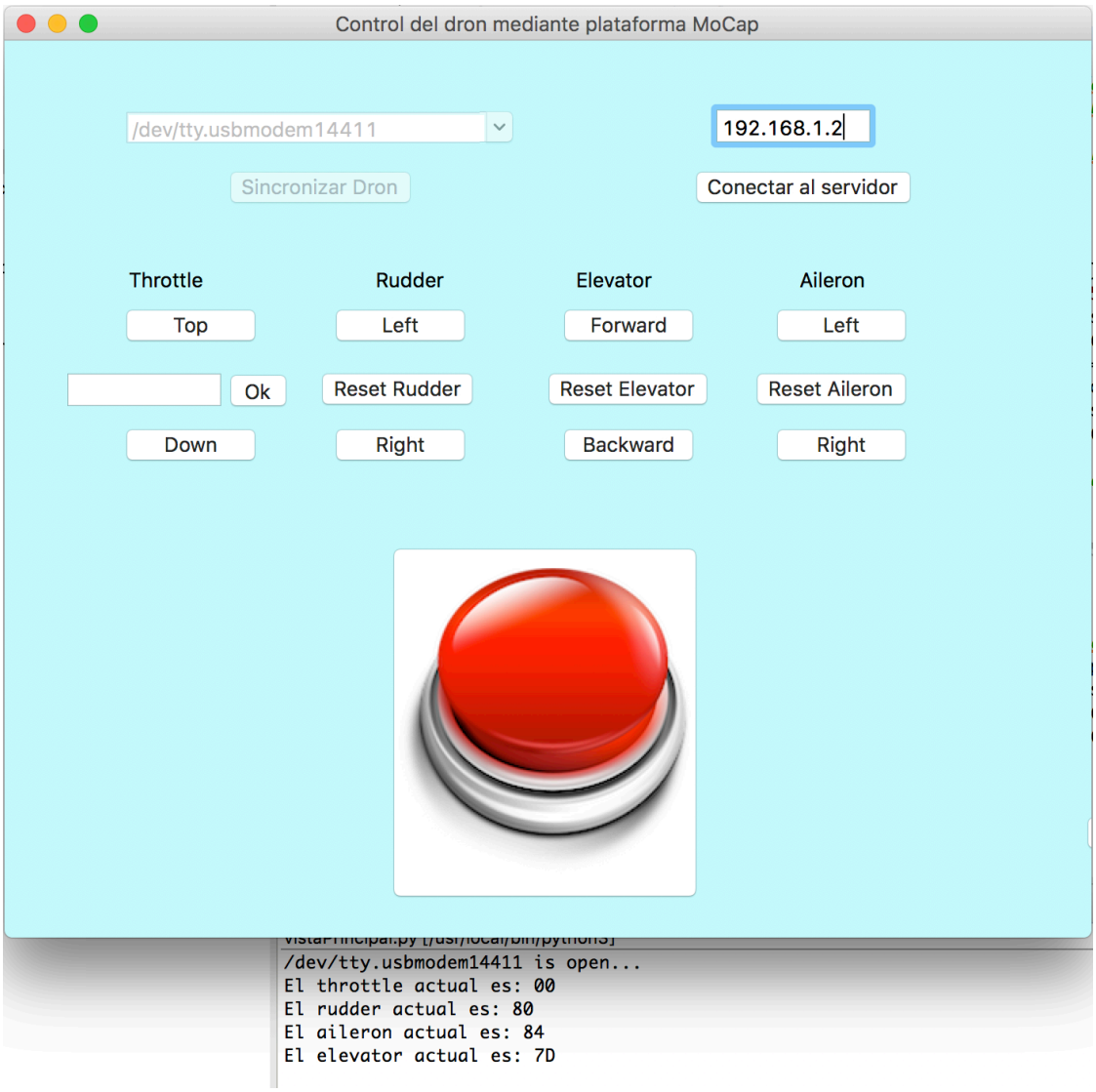

Figura 10.3. Vista principal del software implementado

11. Planificación de tareas

Para poder realizar el proyecto se ha creado un diagrama de Gantt, es una herramienta para planificar y programar las tareas a lo largo de un período determinado.

11.1. Planificación temporal

A continuación se detalla la planificación de las tareas del proyecto dividida en fases junto con su duración y fechas de comienzo/fin.

Nombre de tarea	Duración	Comienzo	Fin
Fase 1: Investigación / Estudio de requisitos	**19 días**	**lun 17/10/16**	**vie 04/11/16**
Evaluación del Dron	5 días	lun 17/10/16	vie 21/10/16
Evaluación del sistema de radio frecuencia	5 días	lun 24/10/16	vie 28/10/16
Evaluación del software / hardware	5 días	lun 31/10/16	vie 04/11/16
Fase 2: Desarrollo del Anteproyecto	**47 días**	**lun 07/11/16**	**vie 23/12/16**
Compra del material para el desarrollo del proyecto	47 días	lun 07/11/16	vie 23/12/16
Fase 3: Implementación	**178 días**	**lun 09/01/17**	**mié 05/07/17**
Comunicación PC - Dron	**36 días**	**lun 09/01/17**	**lun 13/02/17**
Creación del circuito electrónico	5 días	lun 09/01/17	vie 13/01/17
Adaptación de las librerias de Arduino	8 días	lun 16/01/17	lun 23/01/17
Testing de la comunicación entre la interfaz Arduino y el Dron	4 días	mar 24/01/17	vie 27/01/17
Programación del software en Java	10 días	lun 30/01/17	mié 08/02/17
Programación del software en Python	5 días	jue 09/02/17	lun 13/02/17
Documentación comunicación PC - Dron	36 días	lun 09/01/17	lun 13/02/17
Comunicación Mocap - PC	**8 días**	**mar 14/02/17**	**mar 21/02/17**
Testing de la comunicación con el software del Mocap	4 días	mar 14/02/17	vie 17/02/17
Documentación comunicación Mocap - PC	4 días	sáb 18/02/17	mar 21/02/17
Comunicación Mocap - Dron	**134 días**	**mié 22/02/17**	**mié 05/07/17**
Programación y adaptación del software del Mocap en Python	24 días	mié 22/02/17	vie 17/03/17
Testing y modificaciones del software Python	108 días	lun 20/03/17	mié 05/07/17
Documentación Mocap - Dron	134 días	mié 22/02/17	mié 05/07/17
Fase 4: Redacción de documentos	**15 días**	**lun 10/07/17**	**lun 24/07/17**
Fase 5: Entrega de documentación	**3 días**	**mié 27/09/17**	**vie 29/09/17**
Fase 6: Preparación del tribunal	**5 días**	**lun 02/10/17**	**vie 06/10/17**

Figura 11.1. Planificación de las tareas

11.2. Diagrama de Gantt

En la siguiente figura se detallan las tareas en su desarrollo en el tiempo de la dedicación de todas las fases y las dependencia de las tareas.

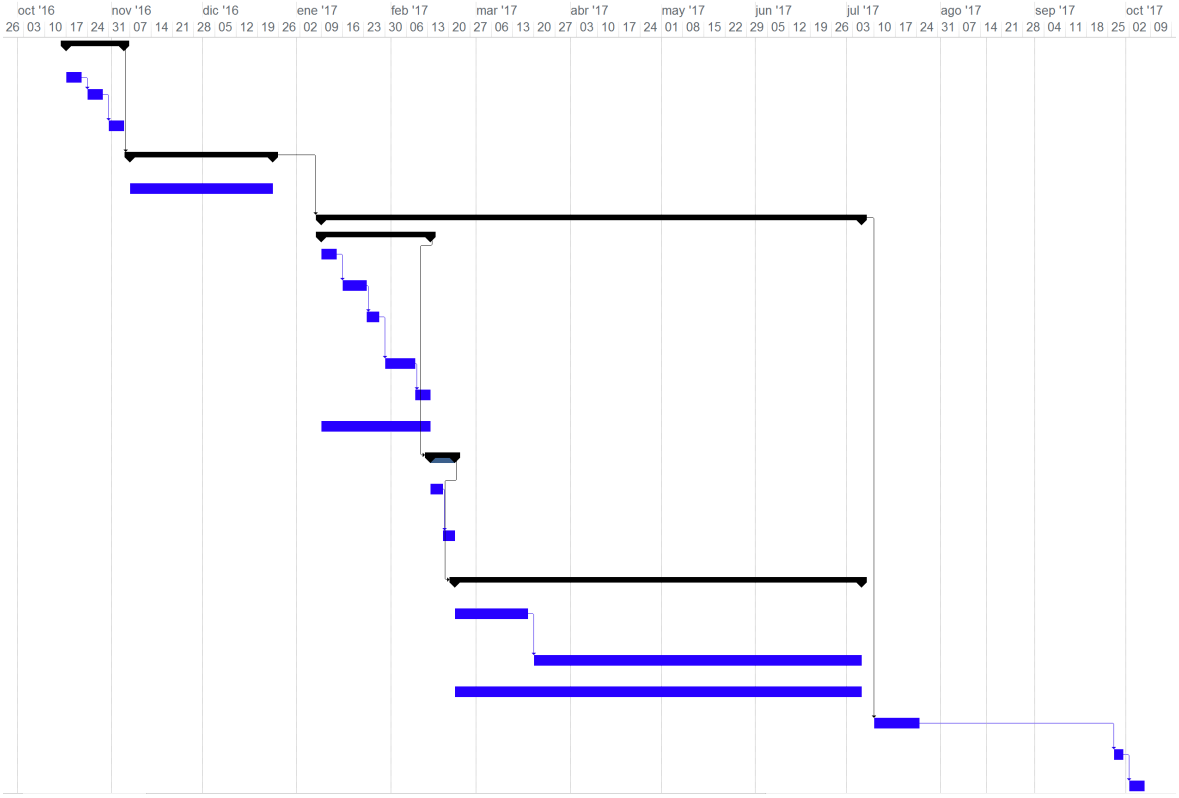

Figura 11.2. Diagrama de Gantt

12. Problemas encontrados

A lo largo del proyecto han salido problemas que en un principio no eran esperados. Esto ha hecho que la planificación del proyecto se retrasase respecto la primera planificación del diagrama de Gantt.

Problemas con la electrónica

Debido al poco conocimiento del campo de la electrónica, se ha tenido que hacer un amplio trabajo de investigación en éste para poder hacer la placa del circuito que conecta el módulo de radio frecuencia con el arduino.

Problemas con el lenguaje de programación

En un principio la idea era hacer todo el software en Java debido al conocimiento de dicho lenguaje. Sin embargo, durante la fase de comunicación con el servidor del Mocap se comprobó que el lenguaje Java no era compatible con las librerías y esto hizo cambiar al lenguaje Python. El cambio de lenguaje provocó retrasos en la planificación debido que es un lenguaje diferente a Java.

Problemas con la versión de Python

Para implementar diversas funcionalidades del software, aprovechando las funciones facilitadas por una serie de librerías, se ha tenido que cambiar todo el código a versión 3.X de Python debido a que estaba implementado en versión 2.X.

Problemas en la sala del Mocap

Estos problemas son debidos a la luz exterior que entra por las ventanas del aula donde está instalado el Mocap, este problema causa la perdida de señal infrarroja de las cámaras e impide la llegada de las coordenadas de los puntos instalados en el Dron y en el actor. Por otra parte se ha visto que desde una altura inferior al metro aproximadamente desde el suelo las coordenadas no son reconocidas.

13. Librerías utilizadas

En el software del proyecto se han utilizado diferentes librerías para desarrollar el código y utilizar diversas funciones para facilitar la implementación.

13.1. Librería Nibabel

Este módulo implementa las rotaciones angulares de Euler y sus conversiones.

El teorema de la rotación de Euler dice que cualquier rotación puede ser descrita por tres ángulos. Los tres ángulos que dan las matrices de rotación son llamados ángulos de Euler, dichos ángulos constituyen un conjunto de tres coordenadas angulares que sirven para especificar la orientación de un sistema de referencia de ejes ortogonales, respecto a otro sistema de referencia de ejes ortogonales normalmente fijos.

La librería ha sido necesaria para hacer la conversión de las coordenadas recibidas del servidor Mocap debido a que el servidor emite las coordenadas en Cuaternión. Los cuaterniones consisten en 4 valores w, x, y, z, donde w es la parte real (escalar), y x, y, z son la parte compleja (vector).

La función **mat2euler(M[,cy_thresh]**) ha sido utilizada para hallar el vector angular de Euler de la matriz 3x3.

Parámetros:

- M: forma de array(3,3)
- Cy_thresh: Ninguno o escalar, opcional

Resultado:

- Z: Escalar
- Y: Escalar
- X: Escalar

La función **quat2mat(q)** ha sido utilizada para calcular la matriz de rotación correspondiente al cuaternión q.

Parámetros:

- Q: array de 4 elementos

Resultado:

- M: matriz (3,3)

  Matriz de rotación correspondiente al cuaternión de entrada q.

La función **quat2euler(q)** ha sido utilizada para obtener los ángulos en Euler correspondientes al cuaternión 'q'.

Parámetros:

- Q: secuencia de 4 elementos

  W,X,Y,Z de cuaternión

Resultado:

- Z: Escalar
  Ángulo de rotación en radianes alrededor del eje z
- Y: Escalar
  Ángulo de rotación en radianes alrededor del eje y
- X: Escalar
  Ángulo de rotación en radianes alrededor del eje x

13.2. Librería Serial

Este módulo encapsula el acceso para el puerto serie y permite el acceso a la configuración del puerto a través de las propiedades de Python. Soporta diferentes tamaños de bytes, bits de parada, paridad y control de flujo con RTS/CTS y/o Xon/Xoff.

Funciones utilizadas:

- Open(): Abre el puerto
- Close(): Cierra el puerto
- Read(size=1): Size = Número de bytes a leer
- Write(data): Data = Datos a enviar
- IsOpen(): Devuelve el estado del puerto serie, True si está abierto, False si está cerrado

Ejemplo para abrir el puerto serie utilizando la librería serial:

p

```
1 import serial
2
3 arduino = serial.Serial("/dev/tty.usbmodem14111", baudrate=9600)
4
5 if arduino.isOpen():
6     print(arduino.name + ' está abierto...')
7
8
```

Figura 13.1.Ejemplo librería serial

Console

```
<terminated> p.py [/usr/local/bin/python3]
/dev/tty.usbmodem14111 está abierto...
```

Figura 13.2. Resultado de la ejecución del ejemplo anterior

13.3. Librería WxPython

La librería WxPython ha sido utilizada para crear la interfaz gráfica del software. Es una librería Open Source y compatible con las plataformas de Microsoft Windows, Mac OS X y Linux.

Ejemplo de la creación de un frame utilizando la librería wxPython:

p

```
1  import wx
2
3  app = wx.App()
4
5  frm = wx.Frame(None, title="Ejemplo de la libreria")
6  frm.Show()
7
8  app.MainLoop()
9
10
```

Figura 13.3.Ejemplo librería wx

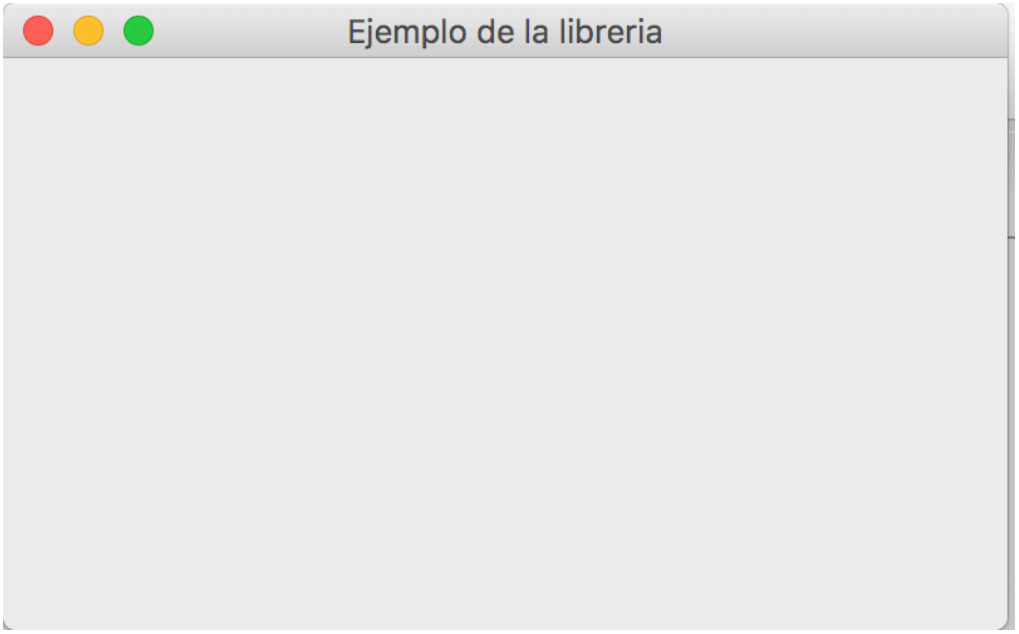

Figura 13.4.Resultado librería wx

13.4. Librería Time

La librería time permite interrumpir el programa. La función time.sleep () utiliza la función sleep () del sistema operativo subyacente.

Función:

- Time.sleep(Segundos): Segundos = Número de segundos que el programa debe detener la ejecución.

Ejemplo de la utilización de la librería Time:

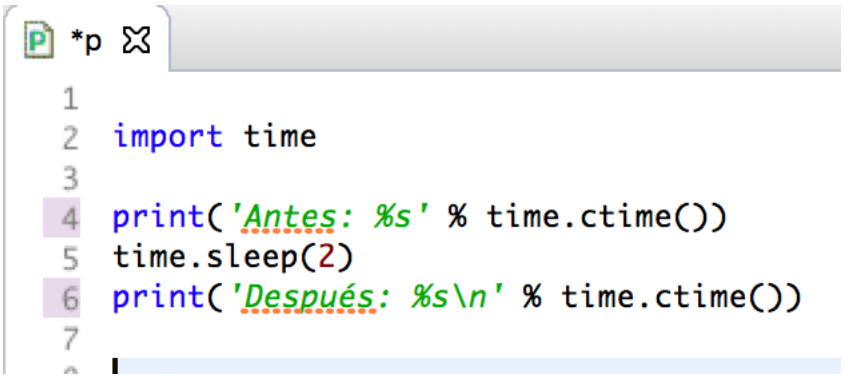

Figura 13.5.Ejemplo librería time

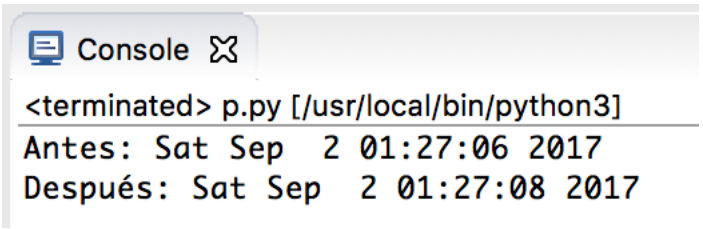

Figura 13.6.Resultado librería time

13.5. Librería Glob

El módulo glob permite buscar todos los pathnames que coinciden con un patrón especificado. Dicho patrón se debe formar por rangos de caracteres expresados entre '[]' y con los signos '*' o '?'.

Función:

- glob.glob('*')

Ejemplos:

Con el siguiente patrón nos devuelve todos los puertos que comienzan por /dev/tty. y a continuación por una letra del alfabeto.

glob.glob('/dev/tty.[A-Za-z]')*

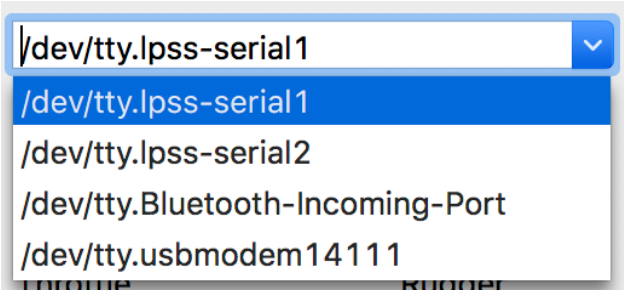

Figura 13.7.Resultado librería glob

13.6. Librería Math

El módulo Math proporciona acceso a las funciones matemáticas definidas por el estándar C para operaciones matemáticas básicas. La función soporta números reales.

Función:

- math.nombrefuncion(X): X = Número real

Algunas funciones disponibles son:

- math.fabs(x): x = Devuelve el valor absoluto de ‘x’
- Math.log10(x): x = Devuelve el logaritmo base10 de ‘x’
- Math.pow(x,y): Devuelve ‘x’ elevado a la potencia ‘y’
- Math.acos(x): Devuelve el arco coseno de ‘x’ en radianes
- Math.radians(x): Convierte el ángulo ‘x’ de grados a radianes
- Math.pi : La constante matemática pi

14. Costes

A continuación se definen los costes totales del proyecto.

Desarrollo del proyecto	26.017,99€
Material	105,10€
Total	**26.123,09€**

Tabla 14.1.Costes total del proyecto

14.1. Desarrollo del proyecto

En el siguiente punto se detallan los costes del desarrollo del software y del hardware/software necesario para su elaboración.

Descripción	**Horas**	**Precio**	**Total**
Jefe de proyecto	160 horas	20€/h	3.200€
Programador	120 horas	15€/h	1.800€
Testing	40 horas	14€/h	560€
		Total	**5.560€**

Tabla 14.2.Costes del desarrollo del software

Descripción	Cantidad	Precio	Total
Macbook Pro	1	2,450€	2,450€
Eclipse	1	0€	0€
Arduino	1	0€	0€
Microsoft Office 2011	1	149€	149€
Parallels Desktop	1	79,99€ anuales	79,99€
Windows 10 Pro	1	279€	279€
OptiTrack - NatNet SDK	1	0€	0€
OptiTrack – Motive Body	1	2.500€	2.500€
OptiTrack - Hardware	1	15.000€	15.000€
		Total	**20.457,99€**

Tabla 14.3.Costes del hardware/software

14.2. Material

En el siguiente punto se detallan los costes del material.

Descripción del producto	Cantidad	Precio	Total
Dron Hubsan H107L x4 v2	1	49,95€	49,95€
Baterías del Dron	3	7,95€	23,85€
Módulo de Radio frecuencia A7105	1	8,95€	8,95€
Kit Arduino Uno	1	21,50€	21,50€
Resistencias eléctricas	1	0,50€	0,50€
Cables de conexión	1	0,40€	0,40€
		Total	**105,10€**

Tabla 14.4.Costes del material

15.Conclusiones

15.1. Valoración del proyecto

El propósito del proyecto de final de grado era la creación de un software que permitiera la interacción entre un actor y un dron mediante una plataforma Mocap.

El objetivo principal era dirigir el dron mediante la captura de movimiento de la plataforma mocap con los movimientos de un actor o unas coordenadas. Dicho objetivo ha sido efectuado en su totalidad por lo que el grado de satisfacción es máximo.

La parte mas decisiva del proyecto ha sido la fase en la que se comunicaba el dron con la placa Arduino, debido principalmente por el desconocimiento en el campo de la electrónica. Cabe recordar que para poder realizar la comunicación ha sido necesario la creación de un circuito electrónico.

A nivel personal la valoración final del trabajo a pocos días de su entrega ha sido muy positiva, tanto por el resultado conseguido como por el aprendizaje sobre otros campos que eran completamente desconocidos. La labor de realizar un proyecto de estas características desde el inicio hasta el final, las dificultades que conlleva realizarlo individualmente y los obstáculos que salen durante el desarrollo hace que aprendas y mejores en aspectos que a lo largo de la carrera no han surgido.

Al ser el primer proyecto que se realiza en la sala del Mocap de la universidad, potenciara en un futuro a que mas alumnos se animen a interactuar con esta tecnología y salgan nuevos proyectos.

15.2. Mejoras

Aunque el objetivo principal del proyecto ha sido satisfactorio, todo proyecto tiene margen de mejora.

Una de las mejoras podría ser implementar un mecanismo de control para permitir un control mas estable sobre el dron, como por ejemplo un mecanismo PID.

El controlador PID es un mecanismo de control sobre una realimentación de bucle cerrado, frecuentemente utilizado en la industria para el control de sistemas. El PID es un sistema al que le entra un error calculado, dicho error se calcula a partir de la salida deseada menos la salida obtenida. Su salida es utilizada como entrada en el sistema que queremos controlar.

El controlador PID viene determinado por tres parámetros: el proporcional, el integral y el derivativo.

A continuación se muestra el esquema del mecanismo de control por realimentación PID.

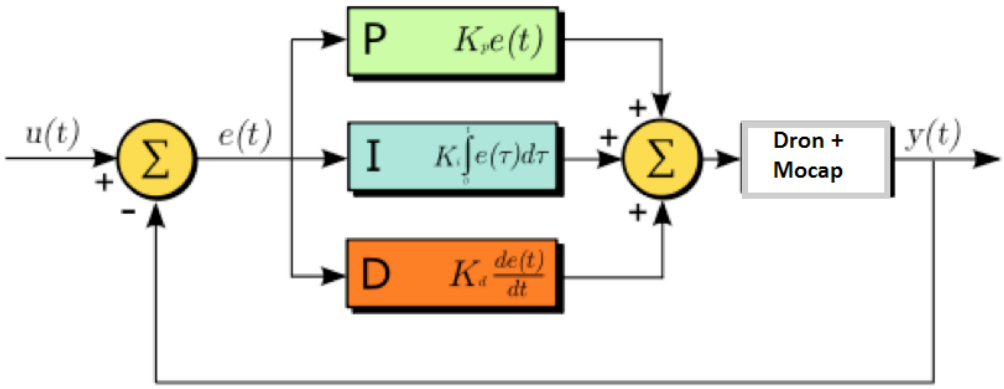

Figura 15.1.Circuito PID

En la salida del sistema (Dron + Mocap) nos da la coordenada del PID implementado. La entrada del sistema es el valor donde tiene que ir el dron. El error es la resta de la entrada menos el error de la salida.

Para cada control del dron que se quisiera controlar, habría que implementar un PID. Por ejemplo para la altura seria un PID, para los giros seria un PID diferente y así para cada control.

16. Bibliografía

[1]http://sabia.tic.udc.es/gc/Contenidos%20adicionales/trabajos/Peliculas/Mocap/introd.htm

[2] https://coatzayork1.wordpress.com/2015/03/30/como-funciona-motion-capture/

[3] https://es.wikipedia.org/wiki/Captura_de_movimiento

[4] http://comofuncionaque.com/que-es-un-drone/

[5] https://definicion.mx/dron/

[6] https://davinci.edu.ar/blog/motion-capture-en-da-vinci/

[7] https://www.bejob.com/que-es-la-programacion-con-arduino-y-para-que-sirve/

[8] https://www.by.com.es/blog/que-es-rfid/

[9] https://msdn.microsoft.com/es-es/library/dd298721.aspx

[10] https://aprendiendoarduino.wordpress.com/2016/09/25/que-es-arduino/

[11] https://github.com/andihit/coptermanager-arduino

[12] http://www.dia.uned.es/~fmorilla/MaterialDidactico/El%20controlador%20PID.pdf

[13] http://nipy.org/nibabel/reference/nibabel.eulerangles.html

[14] http://pyserial.readthedocs.io/en/latest/

[15] https://wxpython.org/

[16] http://pythoncentral.io/pythons-time-sleep-pause-wait-sleep-stop-your-code/

[17] http://docs.python.org/3.1/library/glob.html

[18] https://docs.python.org/3.0/library/math.html

[19] https://www.dronesbaratosya.com/hubsan-x4-h107c/

[20] http://arduino.cl/que-es-arduino/