



ugr | Universidad
de Granada

TRABAJO FIN DE GRADO
INGENIERÍA INFORMÁTICA

Control y coordinación de drones Crazyflie

Autor

Ángel Hurtado Flores

Tutor

Héctor García de Marina Peinado



E.T.S. DE INGENIERÍAS INFORMÁTICA Y DE TELECOMUNICACIÓN

—
Granada, Junio de 2024

Control y coordinación de drones Crazyflie

Ángel Hurtado Flores

Palabras clave: Paparazzi UAV, robótica, firmware, drones, C

Resumen

Los recientes avances en electrónica, robótica e informática han permitido el desarrollo y creación de un sinfín de nuevas tecnologías. Entre ellas, ha destacado recientemente el crecimiento de los drones, formalmente conocidos como UAV (*Unmanned Aerial Vehicle*), dispositivos que permiten una gran variedad de aplicaciones, desde juguetes y hobbies, hasta aplicaciones profesionales como la agricultura o fotografía.

El abanico de posibilidades es uno de los mayores fuertes de esta tecnología, sin embargo, los resultados obtenidos no siempre son los esperados. La mayoría de soluciones aportadas recientemente se apoyan en incrementar el número de UAVs trabajando simultáneamente, lo que implica que se necesita control y coordinación entre los vehículos, generalmente con la ayuda de la informática y las telecomunicaciones.

En este trabajo nos centraremos en el control y coordinación de varios UAVs, principalmente en la subcategoría conocida como cuadricópteros, popularmente llamados drones. Este tipo de UAV destaca por ser capaz de sustentarse mediante la propulsión de 4 hélices permitiendo un buen manejo en tres dimensiones. Se propone como objetivo final conseguir la coordinación entre *Crazyflies*, cuadricópteros ligeros desarrollados por la empresa *Bitcraze*, mediante la implementación de un firmware de código abierto basado en el proyecto *Paparazzi UAV* y la inclusión de un algoritmo para movimiento coordinado entre estos.

Control and coordination of Crazyflie drones

Ángel Hurtado Flores

Keywords: Paparazzi UAV, robotics, firmware, drones, C

Abstract

Recent advancements in electronics, robotics and computing led to the development of countless new technologies. Among them, the growth of drones, formally known as UAV (Unmanned Aerial Vehicles), has recently skyrocketed.

The range of possibilities is one of the greatest strengths of this technology; however, obtained results are not always what is expected from it. Recently proposed solutions rely on increasing the number of UAVs working simultaneously, which implies the need for control and coordination among vehicles, which is typically assisted by computer science and telecommunications.

In this work, we will focus on the control and coordination of multiple UAVs, primarily in the subcategory known as quadcopters, popularly called drones. This type of UAV stands out for its ability to stay aloft thanks to the thrust provided by 4 propellers, allowing for impressive three-dimensional maneuverability. The ultimate goal is to achieve coordination among *Crazyflies*, lightweight quadcopters developed by the company *Bitcraze*, through the development of an open-source firmware based on the *Paparazzi UAV* project and the implementation of an algorithm for coordinated movement between them.

Yo, **Ángel Hurtado Flores**, alumno de la titulación Ingeniería Informática de la **Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación**, con DNI 51184395B, autorizo la ubicación de la siguiente copia de mi Trabajo Fin de Grado en la biblioteca del centro para que pueda ser consultada por las personas que lo deseen.

Firmado: Ángel Hurtado Flores

Granada a 1 de Junio de 2024

D. **Héctor García de Marina Peinado**, Profesor del Departamento Ingeniería de Computadores, Automática y Robótica de la Universidad de Granada.

Informa:

Que el presente trabajo, titulado ***Control y coordinación de drones Crazyflie***, ha sido realizado bajo su supervisión por **Ángel Hurtado Flores**, y autorizo la defensa de dicho trabajo ante el tribunal que corresponda.

Y para que conste, expiden y firman el presente informe en Granada a 1 de Junio de 2024

El director:

Héctor García de Marina Peinado

Agradecimientos

A mis amigos, por las “noches de durum”.

A mi familia, aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa bbbbbbbbbbcccccc

A Héctor y Jesús,

ÍNDICE GENERAL

1. Introducción	19
1.1. Descripción general del sistema	19
1.1.1. Hardware: Crazyflies y posicionamiento	20
1.1.2. Software: Paparazzi UAV	21
1.2. Objetivos	22
2. Montaje y pruebas iniciales	23
2.1. Montaje de los drones	23
2.2. Pruebas con software oficial	24
2.2.1. Preparación	25
2.2.2. Control básico con Crazyflie Client	25
2.2.3. Variables de control en Crazyflie Client	26
2.3. Pruebas con Paparazzi UAV	27
2.3.1. Paparazzi Center	27
2.3.2. Simulaciones en Paparazzi Center	28
2.3.3. Firmware de Paparazzi para Crazyflie 2.1	29
3. Posicionamiento en Paparazzi	31
3.1. Posicionamiento de UAVs en Paparazzi	31
3.2. Implementación del driver	31
3.2.1. Driver para Deck	31
3.2.2. Driver para Node	31
4. Coordinación entre Crazyflies	33
4.1. Algoritmo de coordinación	33
4.2. Implementación del algoritmo	33
4.2.1. Función importante 1	33
4.2.2. Función importante 2	33

5. OtroCapítulo	35
5.1. Apartado1	35
5.2. Apartado2	35
6. Conclusiones	37
A. Instalación de Paparazzi en Debian	39
A.1. Instalación de Paparazzi Center	39
A.2. Instalación de Paparazzi GCS	40
Bibliografia	44

ÍNDICE DE FIGURAS

1.1.	Imagen de un Crazyflie 2.1 montado	20
1.2.	Loco Positioning Deck (izquierda). Loco Positioning Node (derecha)	21
1.3.	Captura de la interfaz de Paparazzi GCS	21
2.1.	Componentes de un Crazyflie 2.1	24
2.2.	Crazyradio 2.0	25
2.3.	Interfaz de Crazyflie Client	26
2.4.	Ejes de rotación de un vehículo aéreo	27
2.5.	Captura de Paparazzi Center bajo Debian 12 con KDE	28
2.6.	Paparazzi GCS ejecutando la simulación por defecto	28

ÍNDICE DE TABLAS

2.1. Componentes de un Crazyflie	2.1	23
2.2. Variables de control de un Crazyflie en Crazyflie Client	2.2	26

Capítulo 1

INTRODUCCIÓN

En este trabajo nos centraremos en el control y coordinación de varios UAVs, principalmente en la subcategoría conocida como cuadricópteros, popularmente llamados drones. Este tipo de UAV destaca por ser capaz de sustentarse mediante la propulsión de 4 hélices permitiendo un buen manejo en tres dimensiones.

Se propone como objetivo final conseguir la coordinación entre *Crazyflies*, cuadricópteros ligeros desarrollados por la empresa Bitcraze, mediante la implementación de un firmware de código abierto basado en el proyecto *Paparazzi UAV* [1] y la inclusión de un algoritmo para la coordinación entre estos.

1.1. Descripción general del sistema

Para la coordinación entre dispositivos necesitaremos como mínimo dos o más Crazyflies, el sistema de localización mediante *anchors* de Bitcraze para los Crazyflies (detallado en subsección 1.1.1.) y un dispositivo de control central, generalmente un portátil con un software para el control de los UAVs.

A este sistema nos referiremos de aquí en adelante como **UAS** (*Unmanned Aerial System*, en inglés), mientras que los términos **UAV**, **rotorcraft** y **dron** se utilizarán indistintamente para referirnos, por lo general, a los Crazyflies u otro dispositivo similar, pero nunca para referirnos al sistema completo.

En las siguientes subsecciones detallaremos cada componente del sistema individualmente, divididos por hardware y software.

1.1.1. Hardware: Crazyflies y posicionamiento

Se trabajará concretamente con los **Crazyflie 2.1**, drones de tan solo 27 gramos y dimensiones 92x92x29mm [2]. Su bajo peso y tamaño implican que tan solo es deseable usarlos en espacios interiores. Entre sus características más destacables, se encuentra la capacidad de sustituir el firmware oficial, consecuencia de ser un dispositivo *open source* y lo que nos permitirá que este trabajo pueda llevarse a cabo.



Figura 1.1: Imagen de un Crazyflie 2.1 montado

Aunque los Crazyflies se pueden utilizar independientemente con un mando radiocontrol o un portátil con el conjunto del software y firmware oficial, el objetivo es la coordinación de movimientos entre los drones, por tanto necesitamos conocer la posición relativa o absoluta de todos los drones en tiempo real. Para ello, Bitcraze dispone de unos módulos basados en UWB (*Ultra WideBand*, banda ultraancha en español) que permiten la localización de los Crazyflies en el espacio con gran precisión. Este sistema, llamado ***Loco Positioning System*** se divide en dos componentes [3]:

- **Loco Positioning Node:** conocidos como *anchors* al estar estáticos en una sala como puntos de referencia. Se sitúan al menos 6 para permitir la triangulación en tres dimensiones de la posición de un Crazyflie. Utilizan los módulos de UWB Decawave DWM1000, ampliamente usados y conocidos módulos de UWB *open source*.
- **Loco Positioning Deck:** módulo de expansión para un Crazyflie, que le permite comunicarse con las *anchors*. También utiliza los Decawave DWM1000.



Figura 1.2: Loco Positioning Deck (izquierda). Loco Positioning Node (derecha)

1.1.2. Software: Paparazzi UAV

Las pruebas iniciales con el sistema serán con el conjunto del firmware oficial para los Crazyflies y el software oficial de ordenador. Una vez se verifique el correcto funcionamiento de cada dron, se pasará al firmware de Paparazzi para los Crazyflie. Junto al software de control, **Paparazzi Center** y **Paparazzi Ground Control Station** [4] se completa la configuración e instalación del sistema para el control y coordinación de los UAV.



Figura 1.3: Captura de la interfaz de Paparazzi GCS

Para conseguir un funcionamiento completo del sistema, se necesita la implementación del *Loco Positioning System* de Bitcraze en Paparazzi. Actualmente solo se dispone del firmware para los Crazyflie [5] el cual únicamente permite un funcionamiento básico.

1.2. Objetivos

Dada la descripción general del sistema, ya es posible entender los objetivos de este trabajo de fin de grado. Los objetivos, ordenados por capítulos y en orden cronológico de desarrollo, son los siguientes:

- **Capítulo 2:** montaje y pruebas iniciales del hardware. Se trata de verificar que el hardware funciona de forma correcta siguiendo los pasos del fabricante, así como una breve familiarización con el control básico de un rotorcraft. Tras verificación con el firmware y software oficial, se realizarán las mismas pruebas básicas con Paparazzi.
- **Capítulo 3:** posicionamiento en Paparazzi. Verificado el funcionamiento mínimo con Paparazzi, se entenderá el funcionamiento de la posición de UAVs en Paparazzi. Esto permitirá el desarrollo de drivers para los módulos de posicionamiento UWB de Bitcraze, consiguiendo un control completo de un solo Crazyflie.
- **Capítulo 4:** coordinación entre Crazyflies. Gracias a la previamente implementada capacidad de localización en el espacio, se usará un algoritmo de trayectorias 3D que será implementado en el firmware de Paparazzi. Los Crazyflies podrán moverse de forma coordinada gracias a este algoritmo.
- **Capítulo 5:** capítulo que aún no se si metere, dependiendo del progreso de este TFG y añadidos opcionales.
- **Capítulo 6:** terminamos este TFG con las conclusiones extraídas (quizá meter aquí ampliaciones futuras?).

Resumidamente, el objetivo final trata de conseguir que los Crazyflies puedan seguir trayectorias 3D de forma coordinada utilizando el conjunto de firmware y software *open source* de Paparazzi UAV.

Capítulo 2

MONTAJE Y PRUEBAS INICIALES

Tras una breve introducción a los dispositivos que se usarán para el desarrollo de esta obra, procederemos a las primeros pasos necesarios para la configuración y control de los drones.

En adelante, se detallan los procedimientos para lo que es conocido como *Maiden Flight*, el primer vuelo de cualquier tipo de vehículo aeronáutico o aeroespacial. Posteriormente, tras las pruebas con el firmware y software oficiales, se realizarán las mismas pruebas con el firmware del proyecto Paparazzi UAV.

2.1. Montaje de los drones

Cada Crazyflie 2.1 necesita de montaje manual. Para ello, se ha seguido la guía oficial del fabricante [6]. Los componentes de un Crazyflie necesarios para el montaje son los siguientes:

Componente	Cantidad
Crazyflie 2.1 Board	1
Hélices tipo CW	2
Hélices tipo CCW	2
Anclajes para motores	4
Motores DC	4
Batería LiPo	1
Anclaje para batería	1
Pines de expansión	1

Tabla 2.1: Componentes de un Crazyflie 2.1

Además, podemos ver en esta figura los componentes de un Crazyflie previo a su montaje:

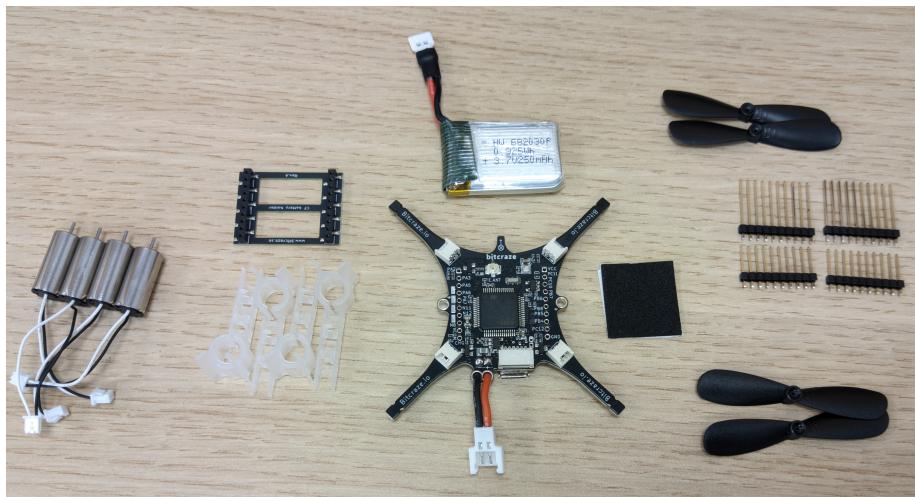


Figura 2.1: Componentes de un Crazyflie 2.1

El procedimiento de montaje, a efectos simplificados, es el siguiente. Se sigue, como bien se ha indicado previamente, la guía oficial [6]:

- Montaje de los motores en cada anclaje para estos.
- Insertar los anclajes en el Crazyflie 2.1 y conectar los motores eléctricamente a este.
- Añadir las hélices a los motores, diferenciando las CW (*ClockWise*) de las CCW (*CounterClockWise*).
- Montar los pines de expansión y la batería.
- Anclar la batería utilizando el anclaje de la batería, que se conecta a los pines de expansión.

2.2. Pruebas con software oficial

Con un Crazyflie montado, podemos proceder a las pruebas iniciales con el conjunto de software y firmware oficial. Desde la guía oficial de montaje se recomienda utilizar la máquina virtual preconfigurada por Bitcraze para conseguir volar el rotorcraft lo antes posible [7]. En esta máquina virtual se incluye los programas para el control, código fuente y otros recursos software de utilidad para el desarrollo con Crazyflies.

2.2.1. Preparación

Previo a la conexión con un Crazyflie, debemos de cargar el firmware al Crazyradio 2.0, el módulo USB que nos permite controlar remotamente un Crazyflie y recibir telemetría de este. Para ello, se sigue la guía oficial del fabricante [8].



Figura 2.2: Crazyradio 2.0

Con ello, ya podemos conectarnos a un Crazyflie desde el software de control Crazyflie Client, no obstante, se recomienda actualizar el firmware del Crazyflie 2.1 previamente utilizando Crazyflie Client. Se sigue la guía oficial de actualización de firmware [9].

2.2.2. Control básico con Crazyflie Client

El último paso previo a cambiar al proyecto Paparazzi consiste en familiarizarse con el control básico de un solo Crazyflie mediante el software oficial. De esta forma se confirma también el correcto funcionamiento de cada dron, asegurando que no es fallo de firmware/software no oficiales (Paparazzi, en este caso). Aquí podemos ver la interfaz de Crazyflie Client:

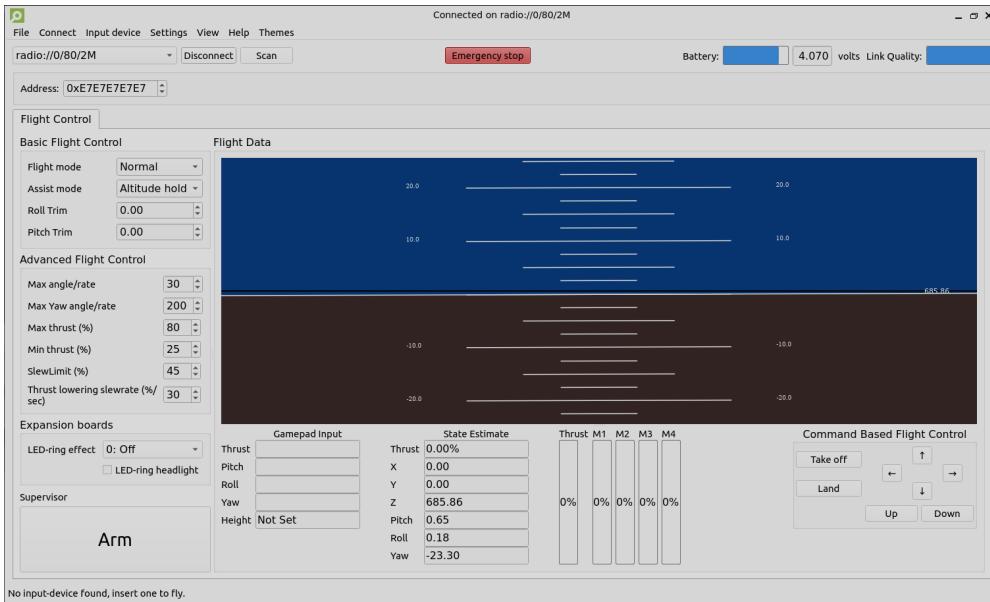


Figura 2.3: Interfaz de Crazyflie Client

2.2.3. Variables de control en Crazyflie Client

Previo a continuar con el desarrollo de este trabajo, es importante entender las variables básicas que definen como se desarrolla el vuelo de un rotorcraft. Aunque existen más, introduciremos aquí todas las variables que Crazyflie Client nos permite saber de un Crazyflie.

En la siguiente tabla se detallan las variables, así como la unidad del Sistema Internacional que se utiliza. Se indican unidades alternativas entre paréntesis, muy utilizadas en otras situaciones o programas.

Variable	Descripción	Unidad
Roll	Rotación sobre el eje X	Grados (Radianes)
Pitch	Rotación sobre el eje Y	Grados (Radianes)
Yaw	Rotación sobre el eje Z	Grados (Radianes)
X	Posición en el eje X	m
Y	Posición en el eje Y	m
Z	Posición en el eje Z	m
Height	Altura respecto al suelo	m
Thrust	Empuje de los motores	%
Battery	Nivel restante de batería	V (%)

Tabla 2.2: Variables de control de un Crazyflie en Crazyflie Client

Cabe destacar las variables **roll**, **pitch** y **yaw** que indican la rotación de un objeto respecto a alguno de sus ejes. Son variables muy usadas en el ámbito de control de vehículos aéreos y aeroespaciales. La siguiente imagen describe de forma gráfica cada una de estas variables.

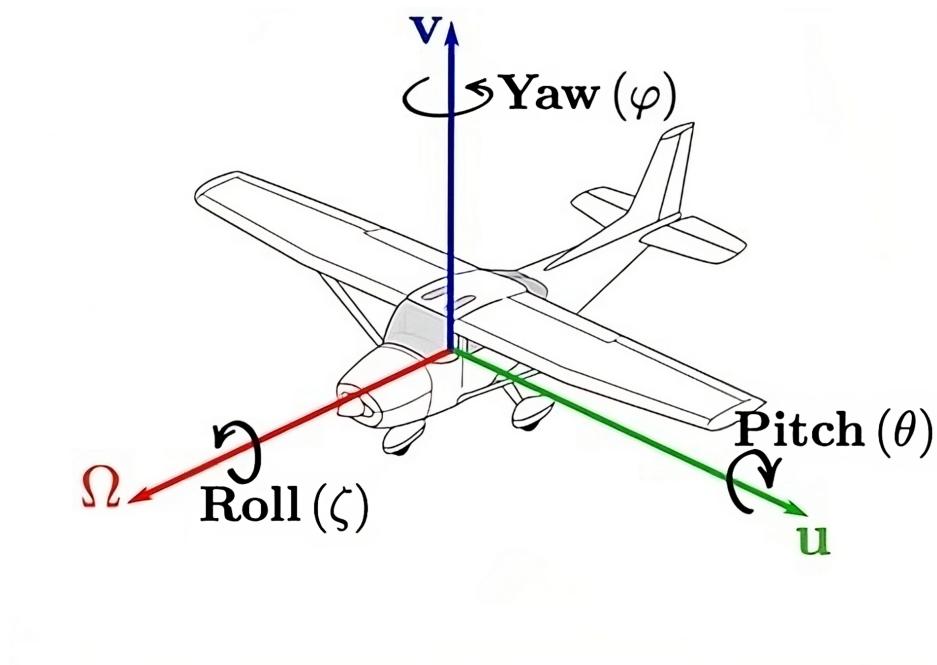


Figura 2.4: Ejes de rotación de un vehículo aéreo

2.3. Pruebas con Paparazzi UAV

Realizadas el montaje y pruebas iniciales, el siguiente paso consiste en la familiarización con el ecosistema Paparazzi. Se instalará Paparazzi Center y acto seguido se realizará una breve simulación de vuelo en Paparazzi GCS. Tras ello, se cargará el firmware de Paparazzi para un Crazyflie y se realizará la primera prueba real.

2.3.1. Paparazzi Center

La instalación de Paparazzi Center se ha realizado siguiendo la guía oficial [10], con leves modificaciones para adaptarlo a Debian 12 (véase Apéndice A). En la siguiente imagen podemos ver la interfaz de Paparazzi Center en Debian 12 con KDE Plasma 5:

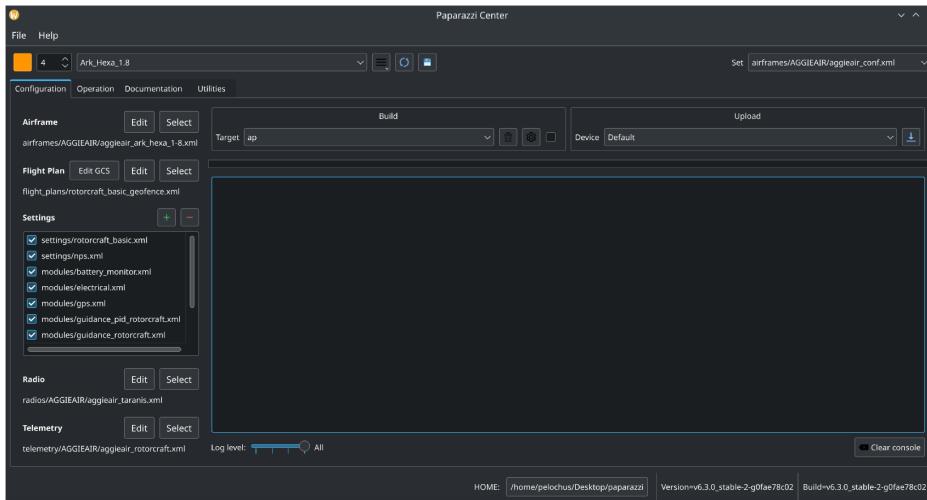


Figura 2.5: Captura de Paparazzi Center bajo Debian 12 con KDE

Desde Paparazzi Center se puede realizar simulaciones, planes de vuelo, observar mensajes de la consola, seleccionar el UAV deseado y un largo etcétera de posibilidades. Se utilizará principalmente los paneles de operación de vuelo, simulación de vuelo, ajustes y consola para depuración.

2.3.2. Simulaciones en Paparazzi Center

Siguiendo la documentación de Paparazzi podemos realizar la primera simulación [12]. Seguir los pasos resulta en la apertura de Paparazzi GCS.



Figura 2.6: Paparazzi GCS ejecutando la simulación por defecto

Desde Paparazzi GCS podemos...

2.3.3. Firmware de Paparazzi para Crazyflie 2.1

Incluir aqui de nuevo [5] que es donde viene la guia de flasheo.

Capítulo 3

POSICIONAMIENTO EN PAPARAZZI

Verificado el correcto funcionamiento del Crazyflie, tanto con el firmware oficial como con el firmware *open source* de Paparazzi, se verá como funciona el posicionamiento de un UAV en Paparazzi y posteriormente se implementará el driver para que los módulos de posicionamiento *Loco Positioning* de Bitcraze funcionen en Paparazzi.

NOTA: APARTIR DE AQUI MENCIONAR EL REPO?? Y CITARLO [13]

3.1. Posicionamiento de UAVs en Paparazzi

Explicar el funcionamiento básico de como se ubica un UAV en Paparazzi

3.2. Implementación del driver

Implementación general del driver

3.2.1. Driver para Deck

Driver para el Deck del Crazyflie

3.2.2. Driver para Node

Driver para los anchors

Capítulo 4

COORDINACIÓN ENTRE CRAZYFLIES

Tras conseguir posicionamiento entre el *Loco Positioning System* y los Crazyflies en Paparazzi, se procede a implementar el algoritmo de coordinación entre dos o más Crazyflies.

4.1. Algoritmo de coordinación

Como funciona, que hace

4.2. Implementación del algoritmo

Implementación en el firmware de Paparazzi

4.2.1. Función importante 1

e

4.2.2. Función importante 2

e

Capítulo 5

OTROCAPITULO

Por llenar

5.1. Apartado1

a

5.2. Apartado2

b

Capítulo 6

CONCLUSIONES

Dados los resultados obtenidos a lo largo de este trabajo de fin de grado...

Capítulo A

INSTALACIÓN DE PAPARAZZI EN DEBIAN

Paparazzi UAV solo se puede instalar oficialmente en Ubuntu [10] sin realizar pasos complejos como compilación desde cero [11].

Desde el equipo de Swarm Systems Lab preferimos el uso de Debian a Ubuntu y, debido a su similitud, hemos descubierto un método para la instalación de Paparazzi en Debian 12, más detallado en el repositorio [13].

La instalación completa se basa en ejecutar el siguiente comando en Debian. Se recomienda ejecutar el script bajo el directorio donde se desee instalar Paparazzi:

```
1 $ wget https://raw.githubusercontent.com/Pelochus/bt-crazyflies/main/scripts  
2      /paparazzi-debian-install.sh && bash paparazzi-debian-install.sh
```

En las dos siguientes secciones se explica como se instaló en detalle Paparazzi en Debian.

A.1. Instalación de Paparazzi Center

Se ha creado el siguiente script de instalación en Bash para automatizar el proceso:

```
1 #!/bin/bash  
2  
3 # Needed to force a PPA repo in Debian  
4 sudo apt install -y build-essential devscripts  
5  
6 # Important note: [trusted=yes] is not recommended  
7 echo "deb [trusted=yes] https://ppa.launchpadcontent.net/paparazzi-uav/ppa/  
8     ubuntu jammy main" | sudo tee -a /etc/apt/sources.list  
9 echo "deb-src [trusted=yes] https://ppa.launchpadcontent.net/paparazzi-uav/  
10    ppa/ubuntu jammy main" | sudo tee -a /etc/apt/sources.list
```

```

9
10 # Rest of the official Paparazzi guide
11 sudo apt update
12 sudo apt install -y paparazzi-dev gcc-arm-none-eabi gdb-multiarch python-is-
    python3 paparazzi-jssim dfu-util
13
14 # Clone Paparazzi
15 git clone --origin upstream https://github.com/paparazzi/paparazzi.git
16 cd paparazzi
17 git checkout v6.3 # Currently last stable version
18
19 # Compile latest stable
20 make clean && make

```

Para entender este script, hay que hacer hincapié en las siguientes líneas:

- **Líneas 11 y 12:** Añadimos los repositorios de descarga para Paparazzi. Para que funcionen en Debian, se añade [`trusted=yes`] de forma que se fuerce la confianza en el repositorio, ya que no es posible añadirlos de la forma natural según la guía. Se indica además, que se utilice los paquetes para la última versión de Ubuntu que disponen desde el equipo de Paparazzi, en este caso, `jammy`.
- **Línea 16:** Se instalan todos los paquetes según la guía oficial para Ubuntu, exceptuando el paquete para Paparazzi GCS, que no es posible utilizarlo en Debian

El resto de líneas son extraídas de la guía oficial de instalación.

Este script debe funcionar en Debian 12 recién instalado si se posee conexión a la red. Es posible, aunque no ha sido comprobado, que funcione en distribuciones basadas en Debian u otras versiones de Debian como Debian 11. No se recomienda utilizar en distribuciones basadas en Ubuntu, se puede seguir la guía oficial en estos casos.

A.2. Instalación de Paparazzi GCS

Como se ha comentado en la sección previa, el paquete correspondiente a Paparazzi GCS no se puede instalar bajo Debian. Para ello, se puede utilizar el **AppImage** oficial, que funciona similar a un ejecutable en Windows o un contenedor de Docker.

Este AppImage necesita ser integrado con el ejecutable de Paparazzi, ya que por defecto este asume que se ha instalado el GCS desde el gestor de paquetes `apt` y, por tanto, hay acceso universal.

En este caso, se utiliza este script para la configuración e integración de Paparazzi GCS en Debian:

```
1 #!/bin/bash
2
3 cd paparazzi
4
5 # Make Paparazzi GCS AppImage Work
6 echo "# Needed for Paparazzi AppImage to work" >> /home/$USER/.bashrc
7 echo "export PAPARAZZI_HOME=$(pwd)" >> /home/$USER/.bashrc
8 echo "export PAPARAZZI_SRC=$(pwd)" >> /home/$USER/.bashrc
9
10 # Get AppImage and move to /usr/bin/pprzgcs so it can be launched by
11 # Paparazzi Center
12 sudo apt install -y wget # Just in case, you should have it anyway
13 sudo wget https://github.com/paparazzi/PprzGCS/releases/download/v1.0.11/
14     pprzgcs-v1.0.11-x86_64.AppImage -O /usr/bin/pprzgcs
15 sudo chmod 755 /usr/bin/pprzgcs
```

En este caso, es mucho más sencillo entender este script:

- Primeramente se establecen las variables de entorno `PAPARAZZI_HOME` y `PAPARAZZI_SRC`, que son necesarias para el funcionamiento del AppImage.
- Utilizando el ejecutable `wget`, descargamos automáticamente el AppImage y lo movemos a `/usr/bin/pprzgcs`, para que simule el comportamiento de descargar el paquete oficial de la repo.

En general, la instalación de Paparazzi en Debian se condensa en ejecutar el primer script de este apéndice, que se ocupa de llamar a los otros dos scripts, de forma que se pueda modularizar la instalación de ambos componentes.

BIBLIOGRAFÍA

- [1] Paparazzi Team, “Paparazzi UAV – Source Code.” <https://github.com/paparazzi/paparazzi>.
- [2] Bitcraze, “Crazyflie 2.1 Product Page.” <https://www.bitcraze.io/products/crazyflie-2-1/>, 2023.
- [3] Bitcraze, “Loco Positioning System Overview.” <https://www.bitcraze.io/documentation/system/positioning/loco-positioning-system/>, 2023.
- [4] Paparazzi Team, “Paparazzi GCS.” <https://github.com/paparazzi/PprzGCS>.
- [5] Paparazzi Team, “Paparazzi Wiki – Crazyflie 2.0/2.1.” https://wiki.paparazziuav.org/wiki/Crazyflie_2.0.
- [6] Bitcraze, “Getting Started with the Crazyflie 2.X.” <https://www.bitcraze.io/documentation/tutorials/getting-started-with-crazyflie-2-x/>.
- [7] Bitcraze, “Bitcraze Virtual Machine.” <https://github.com/bitcraze/bitcraze-vm>.
- [8] Bitcraze, “Getting started with the Crazyradio 2.0.” <https://www.bitcraze.io/documentation/tutorials/getting-started-with-crazyradio-2-0/>.
- [9] Bitcraze, “Crazyflie Client Guide – Firmware Upgrade.” https://www.bitcraze.io/documentation/repository/crazyflie-clients-python/master/userguides/userguide_client/#firmware-upgrade.

- [10] Paparazzi Team, “Quick Install – Paparazzi UAV ReadTheDocs.”
<https://paparazzi-uav.readthedocs.io/en/latest/quickstart/install.html>.
- [11] Paparazzi Team, “Paparazzi Wiki – Installation in Linux from scratch.”
<https://wiki.paparazziuav.org/wiki/Installation/Linux>.
- [12] Paparazzi Team, “First Simulation – Paparazzi UAV ReadTheDocs.”
https://paparazzi-uav.readthedocs.io/en/latest/quickstart/first_simulation.html.
- [13] Ángel H. F., “Control and coordination of Crazyflie drones — Bachelor’s Thesis.” <https://github.com/Pelochus/bt-crazyflies>.