UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería del Diseño

Diseño y construcción de un UAS mediante Raspberry Pi y código abierto

Trabajo Fin de Grado
Grado en Ingeniería Aeroespacial

Autor: Vicent Barrera Gaspar
Tutor: Ángel Rodas Jordá
Cotutor: Pablo Morcillo Pallarés

Contenido de la presentación

Motivación y objetivos

Conceptos previos

Evolución

Tipos

Funcionamiento UAS

Requerimientos

Propuesta de UAS

Hardware

Software

Pruebas de campo

Conclusiones y trabajos futuros

Presupuesto

Turno de preguntas

Motivación y objetivos

Motivación

Puesta en práctica de conocimientos del grado

Aprendizaje: campo multidisciplinar y en auge

Numerosas aplicaciones futuras

Objetivos

Diseño y construcción de un UAS (Unmanned Air System) de Código abierto

Documentación del proceso: memoria divulgativa

Conceptos previos

Drone o Drone: Traducción de zángano

UAV: Unmanned Aerial Vehicle. Obsoleto

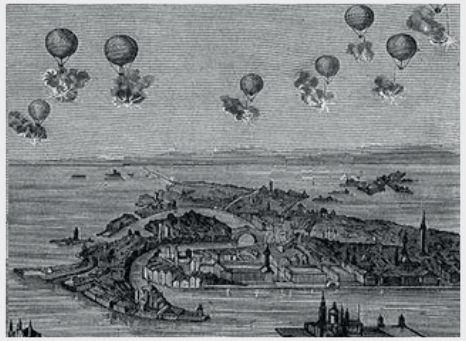
UA: Unmanned Aircraft

UAS: Unmanned Aerial System

RPA: Remotely-Piloted Aircraft

RPAS: Remotely-Piloted Aircraft System

Evolución



1849

1945-1989
(Guerra Fría)

1898

sXXI

Tipos

Según su...

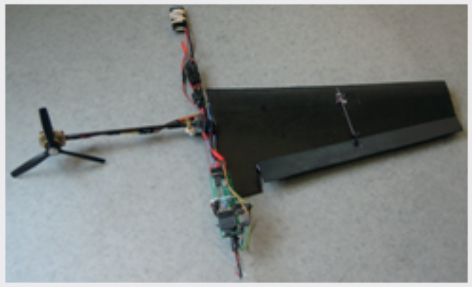
Ala: fija o rotatoria

Número de motores: cuadricóptero, hexacóptero...

Fuente de su energía eléctrica: solar, fósil o batería (LiPo)

Forma de comunicación: WiFi, Telemétrica, Radio,...

Aplicación: civil o militar



Funcionamiento UAS

Órdenes desde Radiocontrol o Estación Terrestre

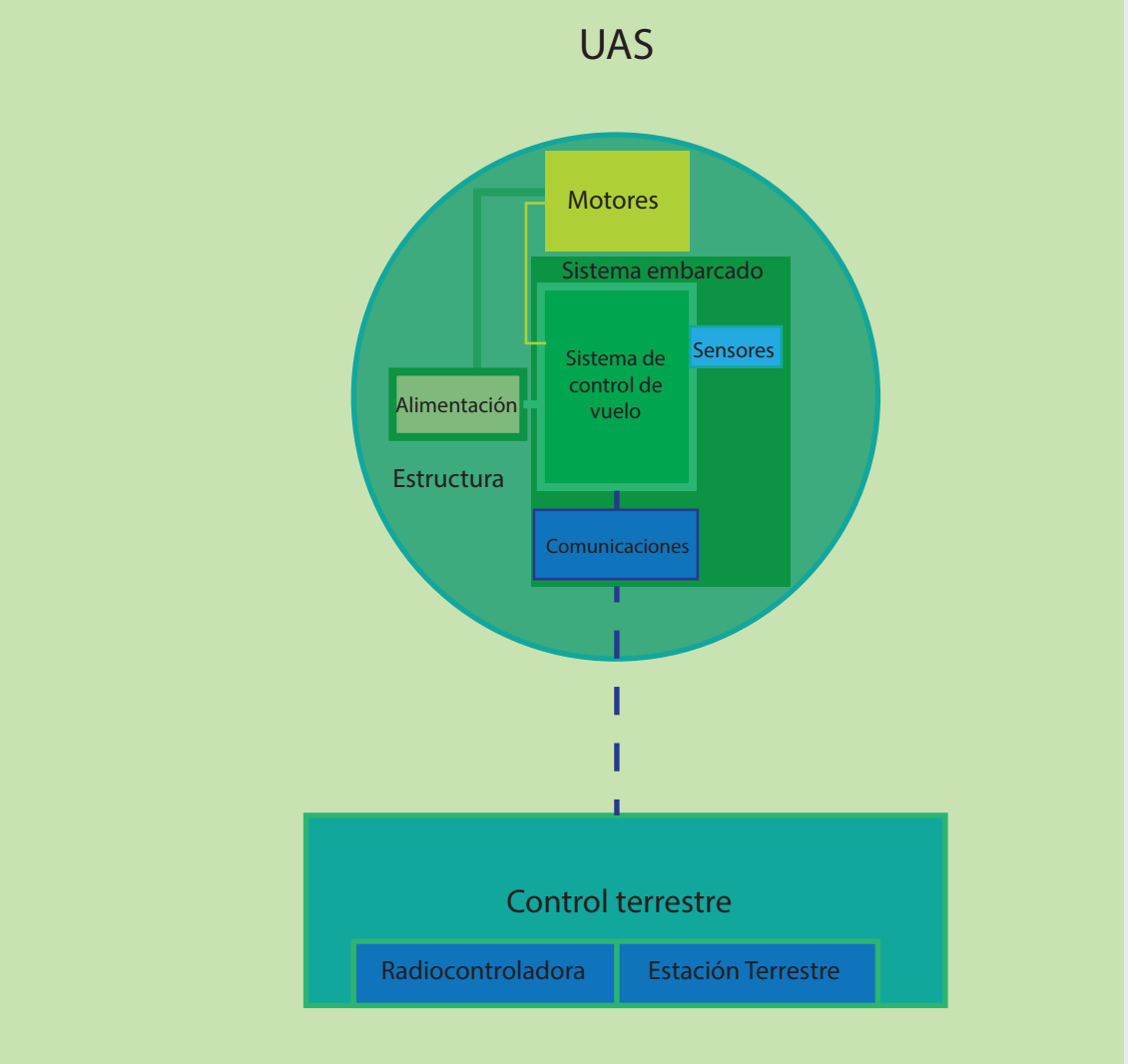
Lectura, ejecución y control de la orden:
Sistema de control de vuelo

Movimiento de la nave: Motores

Información de vuelo: Sensores

Transferencia de datos y órdenes: Comunicaciones

Protección y sujeción: Estructura



Requerimientos

Estabilidad → Ala rotatoria; multicóptero hexacóptero

Autonomía → Tipo de alimentación: baterías precargadas LiPo

Configuración flexible y adaptable → Software libre

Diversas vías de control → Control mediante radiocontroladora y estación terrestre (WiFi y Telemetría)

Conexión GNSS → Acceso a constelaciónes satelitales GNSS

Toma de imágenes → Instalación de sensores básicos y cámara

Potencial final : desarrollo y prueba de aplicaciones → Posibilidad de conectar un segundo procesador de vuelo y conectar periféricos adicionales (Arduino)

Propuesta de UAS

Ala rotatoria; multicóptero hexacóptero

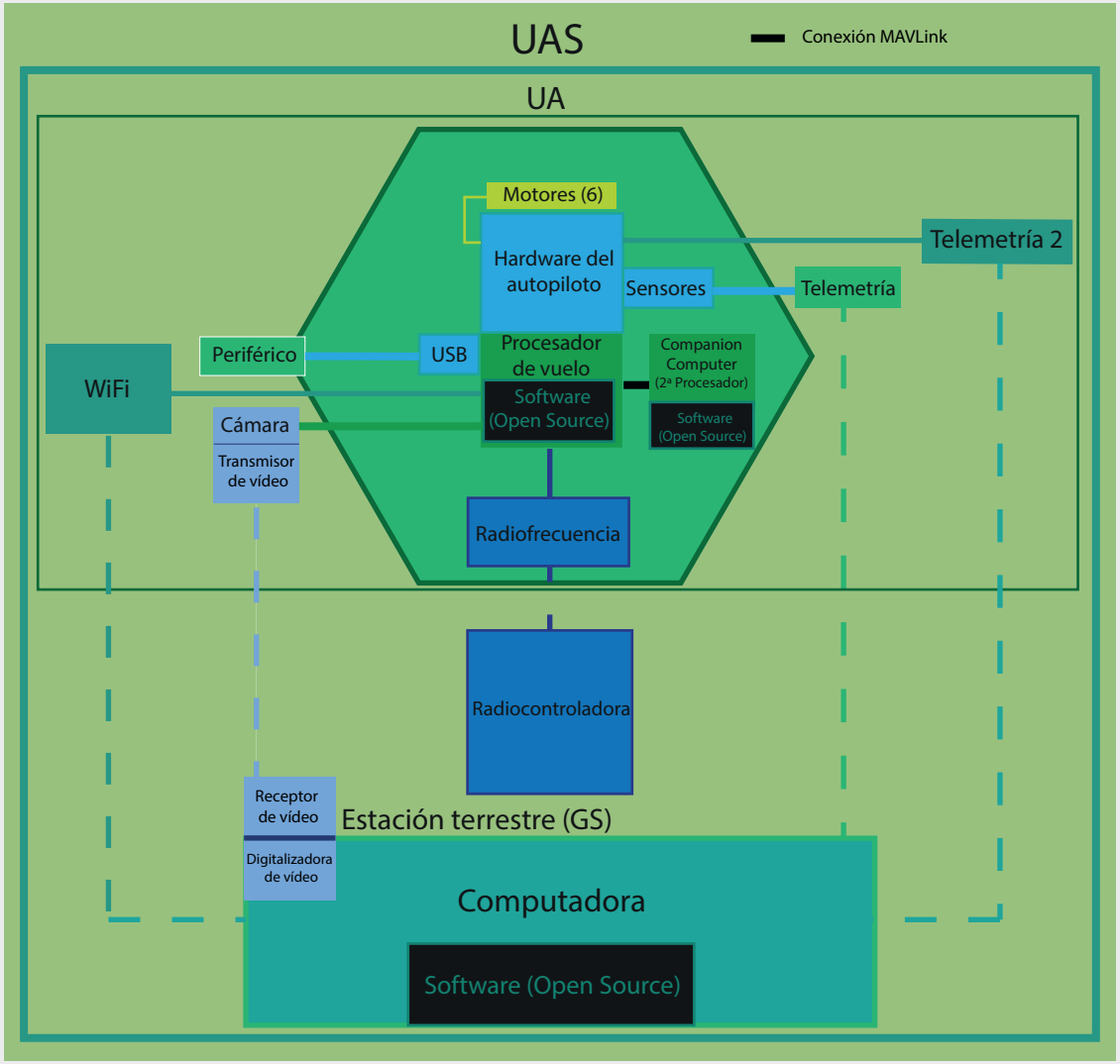
Tipo de alimentación: baterías precargadas LiPo

Software libre

Control mediante radiocontroladora y estación terrestre (WiFi y Telemetría)

Instalación de sensores básicos y cámara

Posibilidad de conectar un segundo procesador de vuelo (Companion Computer) y conectar periféricos adicionales (Arduino)

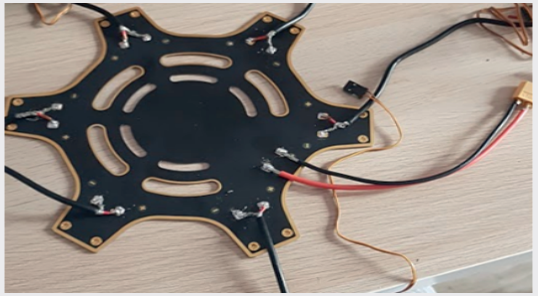


Hardware: Motores y Estructura

Motores: DJI 2312E - 960RPM

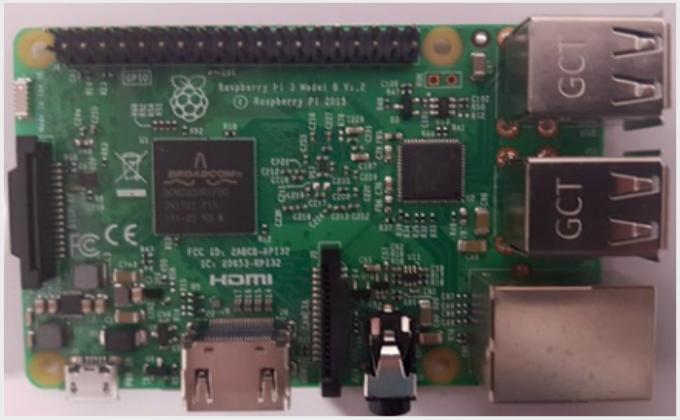
Estructura: DJI Flamewheel F550

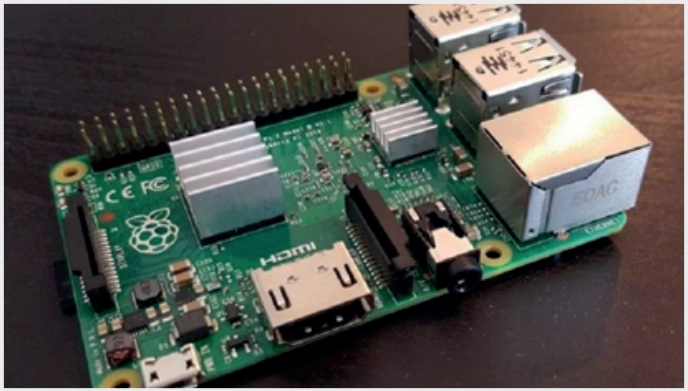
Montaje

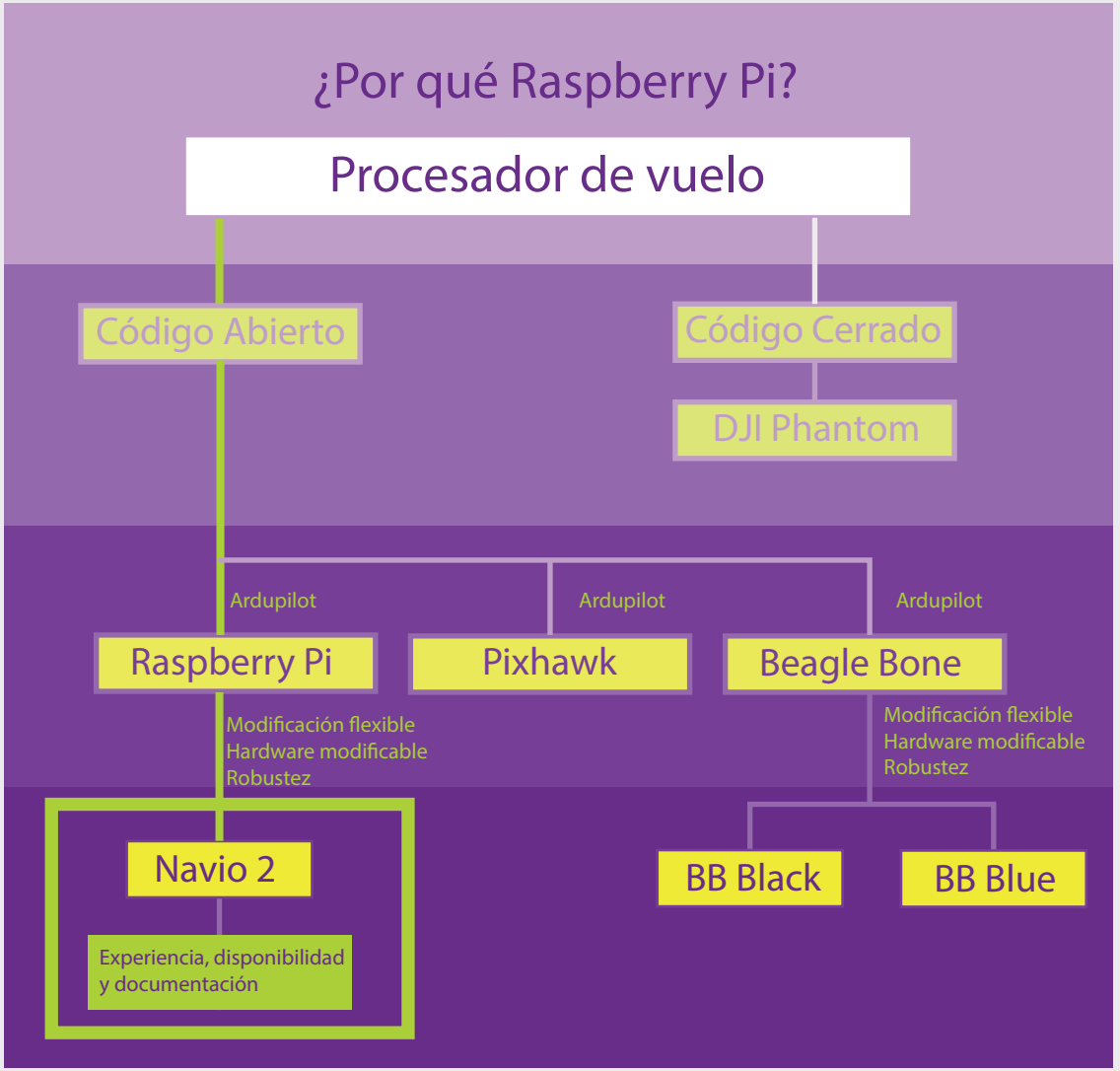


Hardware: Sistema embarcado (Procesador de vuelo)

Raspberry Pi 3

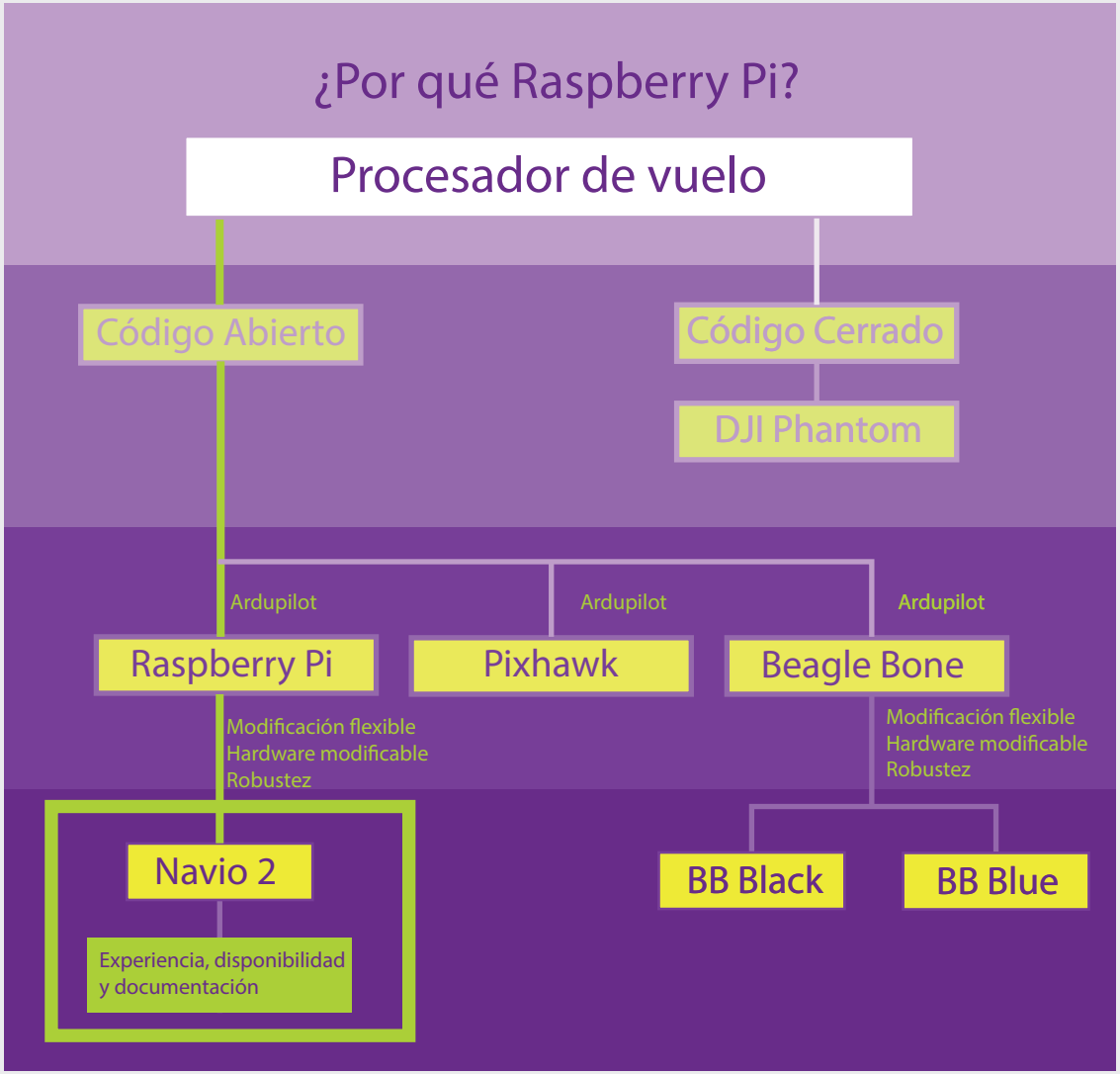






Hardware: Sistema embarcado (Procesador de vuelo)

Raspberry Pi 3 + RaspiCam



Hardware: Sistema embarcado (Hardware del autopiloto)

NAVIO2

IMU Dual

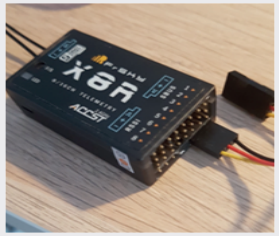
Receptor GNSS: GLONASS, GPS, Galileo, Beidou y SBAS

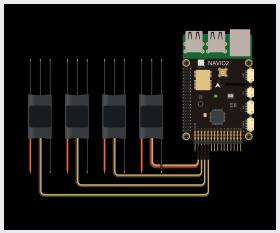
Coprocesador RC I/O

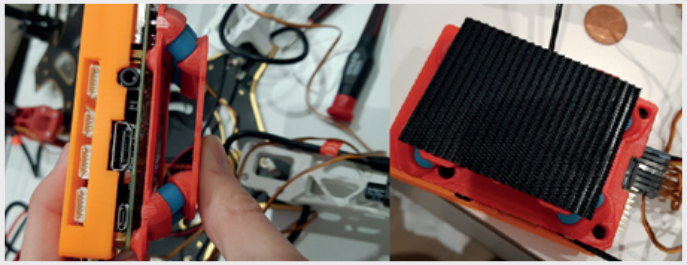
Suministro de energía de triple redundancia

Compatible con radio y más sensores

Puede utilizar ArduPilot







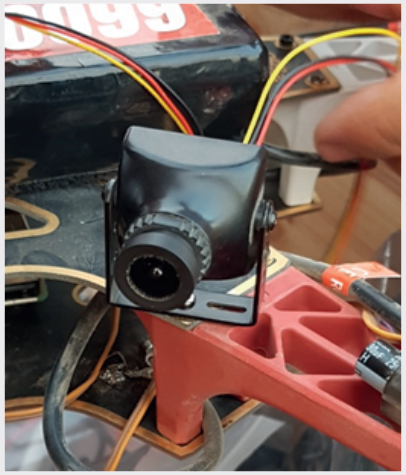
Hardware: Control terrestre

Radiocontroladora

Telemetría

Hardware: Sensores y transmisiones

Transmisor de vídeo (5.8GHz) y cámara (Foxeer)



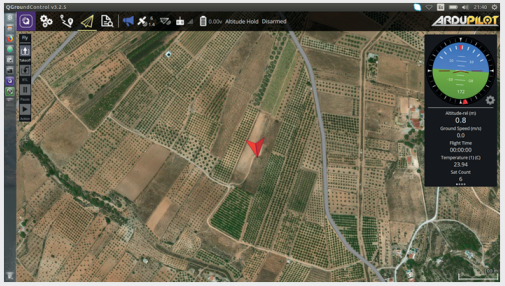
GNSS (GlobalSat)

Software: Estación terrestre

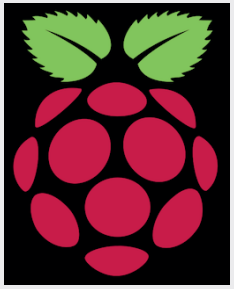
Requisito: Código Abierto

Kubuntu

QGroundControl



Procesador de vuelo: Emlid-Raspbian



Python 2.7

DroneKit

Autopiloto: ArduPilot

OpenCV

Software: Configuración

1. Instalación de sistemas operativos

2. Configuración inicial y activación de ArduPilot (apéndice 2)

3. Instalación de QGroundControl, Python y sus librerías (DroneKit y OpenCV)

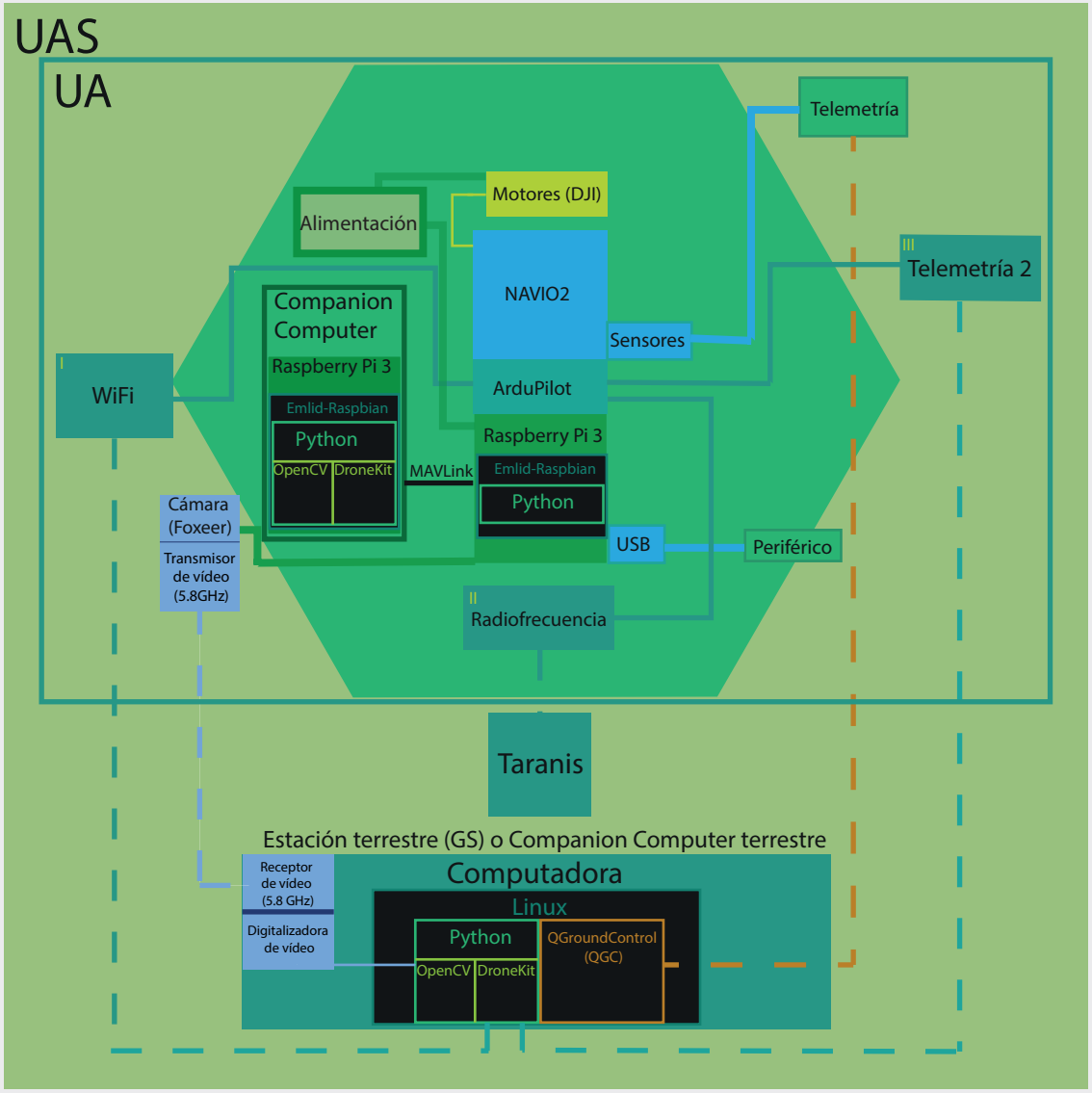
4. Preparar y configurar el control remoto:

   - Control mediante emisora radio: configuración de QGroundControl (apartado 3.4.2) y calibrado de la radiocontroladora (Apéndice 5)

   - Control mediante WiFi: apartado 3.3.3.2

   - Control mediante Telemetría: apartado 3.3.3.3 y apéndice 2

Esquema Final



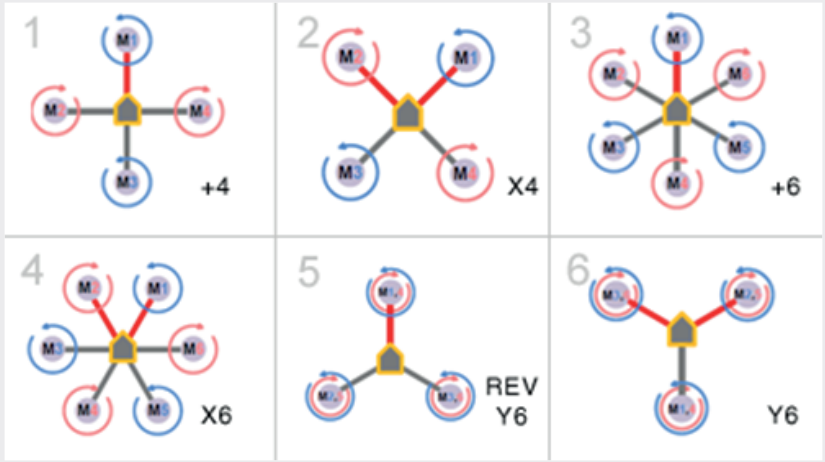
Pruebas de campo: Comprobaciones previas

Sistema eléctrico

Atornillado y fijaciones

Alimentación de motores

Giro de los motores



Pruebas de campo: vuelo con emisora

Encendido de la radioemisora

Asegurar el correcto calibrado de QGrounControl (apartado 3.4.2) y del radioenlace

Armado de motores

Prueba de dos modos:

- AltHold
- PosHold

Pruebas de campo: vuelo con DroneKit

Vehicle_state

Comando a lanzar el programa:

```
python vehicle_state.py --connect /dev/ttyUSB0
```

Como respuesta, el UA envía los datos de vuelo en tiempo real.

Nos envía: localización global y localización global relativa (cada una con su respectiva latitud, longitud y altura) y localización local NED (North East, Down), así como velocidad en los tres ejes (x,y,z)

Modificaciones del código original:

Línea 42: `vehicle = connect(connection_string, wait_ready=True, baud=57600)`

Línea 278 : `heartbeat_timeout=180`

o bien:

Línea 2431: `def set(self, name, value, retries=3, wait_ready=False)`

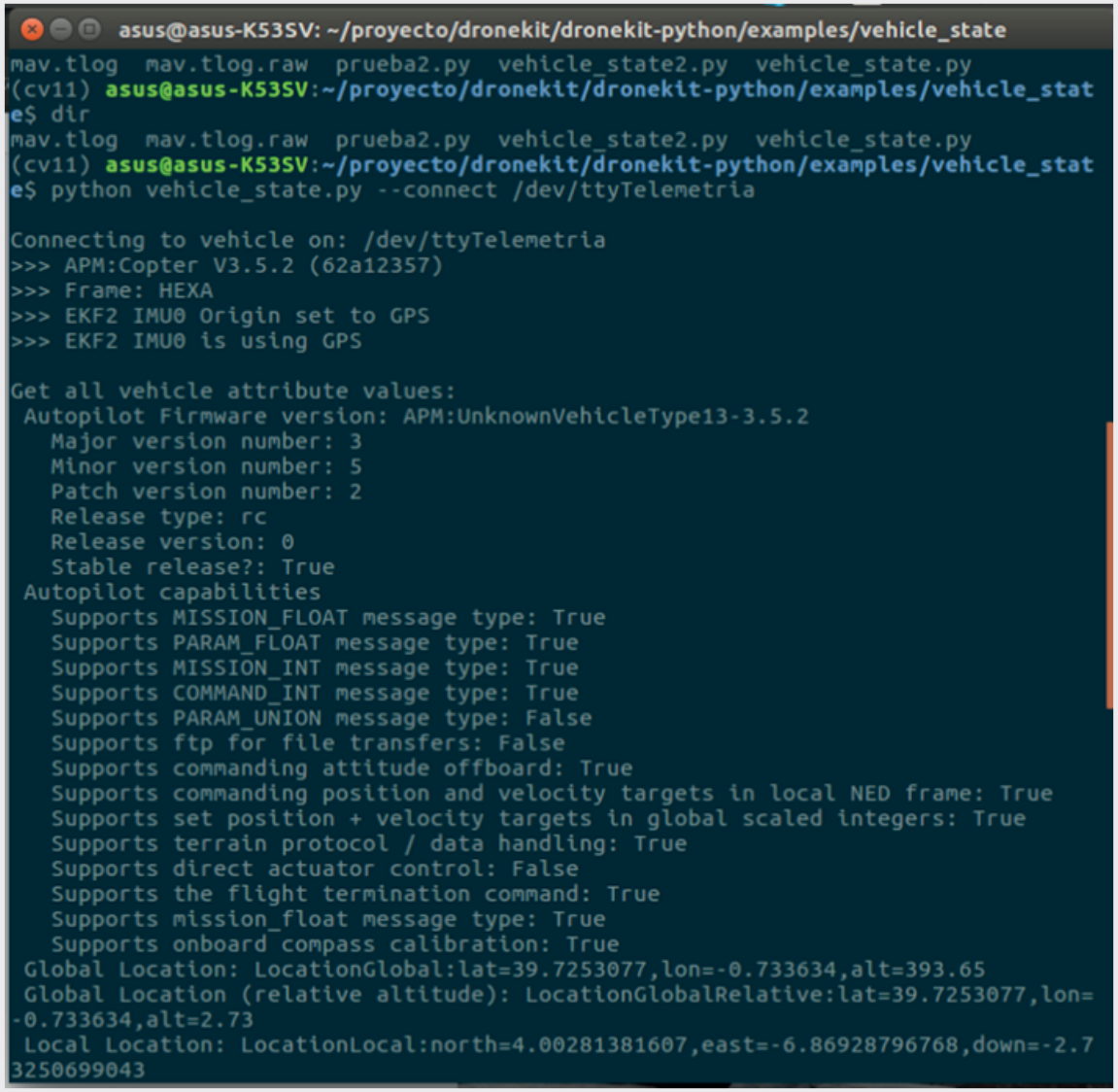
Datos emergentes:

```
print "Local Location: %s" % vehicle.location.local_frame    #NED
print "Velocity: %s" % vehicle.velocity
...
```

Pruebas de campo: vuelo con DroneKit

Vehicle_state

Captura de la ejecución del programa y los datos recibidos en la prueba de vuelo:



Pruebas de campo: Vuelo con DroneKit

follow_me

Comando a lanzar el programa:

```
python follow_me.py --connect /dev/ttyUSB0
```

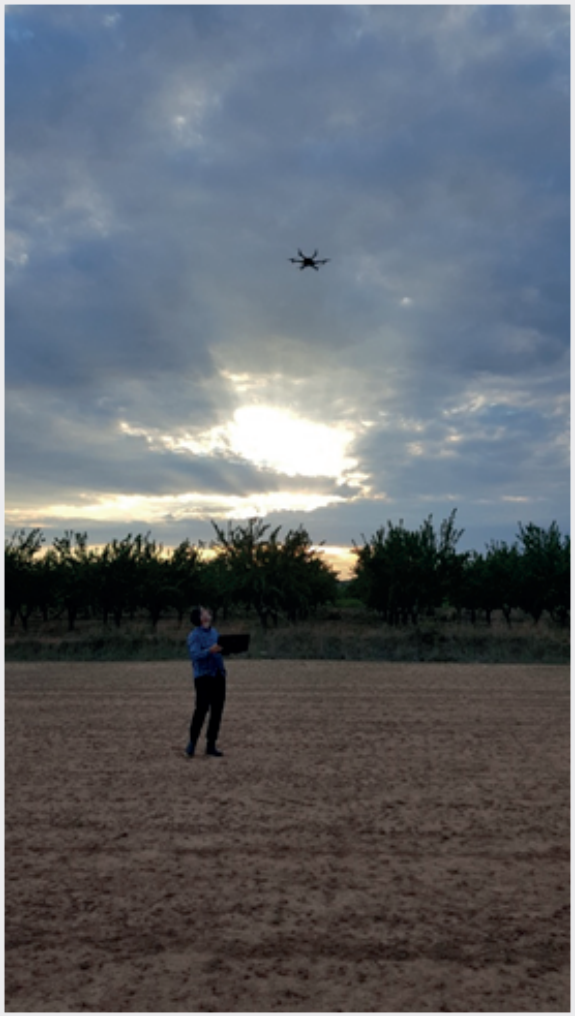
Al ejecutar el código, la aeronave despegará y seguirá la localización del receptor GNSS conectado al ordenador, mediante la lectura de la posición del propio UA y su constante corrección.

Modificaciones del código original:

Línea 108: `time.sleep(0.02)`

Pruebas de campo: Vuelo con DroneKit

follow_me
fotografías de la prueba de vuelo:



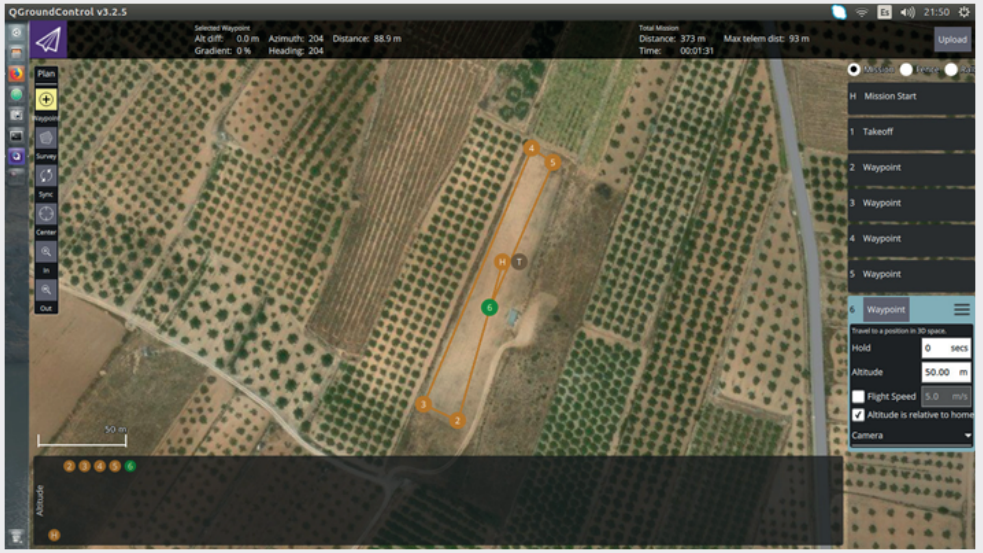
Pruebas de campo: vuelo con QGroundControl

Creación y ejecución de un plan de vuelo

El UA seguirá automáticamente una ruta generada desde QGroundControl.

Pasos:

- Abrimos QGroundControl
- Cambiar a modo Plan View
- Crear los waypoints o comandos deseados
- Enviar o actualizar la misión al vehículo
- Cambiar a modo Flight View y ejecutar la misión o plan de vuelo



Pruebas de campo: toma de imágenes y detección de colores

Opencv_stream (detección de color)

Comando a lanzar el programa (con Python 2.7) :

```
python opencv_stream.py
```

Si no se usa Python 2.7, debemos usar antes:

```
workon cv11
```

El comando permite detectar un color, y en la muestra
de imágenes aparecerá resaltada con un punto y un círculo

Modificaciones del código original:

Línea 17: `vs = VideoStream(src=1, resolution=resolution, framerate=fps).start()`

Pruebas de campo: toma de imágenes y detección de colores

Opencv_stream (detección de color)

Comando a lanzar el programa:

```
python opencv_stream.py
```

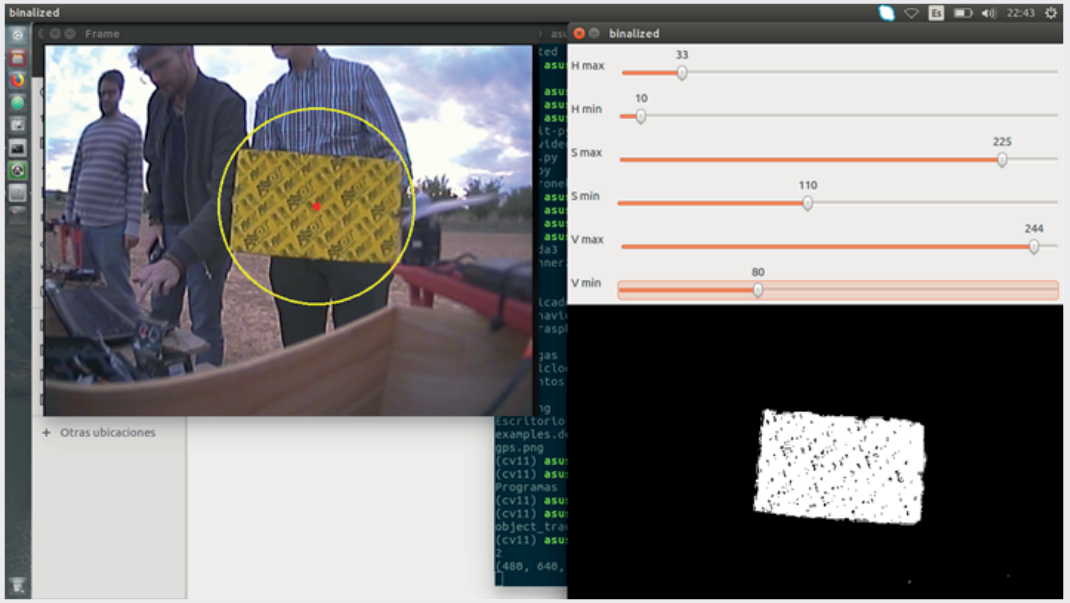
El comando permite detectar un color, y en la muestra de imágenes aparecerá resaltada con un punto y un círculo

Proceso:
Ejecución del comando: apertura de GUI

Aislamiento del color deseado (binarizado) mediante la modificación de valores en la GUI: H(Hue o Matiz), S (Saturation o Saturación) y V (Valor).

Obtención del contorno del objeto mayor (círculo amarillo) y cálculo del centro de masas (punto rojo).

La pantalla muestra la imagen en tiempo real y el color detectado (amarillo en nuestro caso).



Conclusiones y trabajos futuros

Conclusiones

Aprendizaje multidisciplinar

Detección y solución de errores

Se han cumplido los objetivos planteados

Trabajos futuros

Desarrollo de procesado de imágenes y control del UA a través de este

Instalación de un Companion Computar

Mejorar la detección de entorno: Arduino

Presupuesto

Componente	Precio	IVA	TOTAL
Navio2 (Hardware de autopiloto+ Antena GNSS +Cableado + Power Cell)	216,53	45,47	262
BATERÍAS	111,57	23,43	135
Raspberry Pi 3	35,54	7,46	43
Alimentación Raspberry Pi 3 y Cargador baterías	45,45	9,55	55
Tarjetas memoria: Intenso 3413470 Micro SD clase 10 16GB	9,92	2,08	12
HDMI	7,44	1,56	9
L-link Teclado + Ratón LL-KB-816-COMBO USB Negro	7,44	1,56	9
Kit DJI F550 ARF	219,01	45,99	265
FlySky XR8 8CH antena PCB 16CH con Receptor	148,76	31,24	180
Kit Tx/Rx video 5.8Ghz 400mW	40,50	8,50	49
SkyRC Imax B6	34,71	7,29	42
Modulo telemetría 433Mhz 500mW MWC APM-PIX	37,19	7,81	45
Cámara Raspberry Pi 3 + Módulo RF	70,25	14,75	85
Otro material: cables, tornillería, conectores, etc.	24,79	5,21	30
TOTAL			1222 €

Si planteamos el ejercicio de la venta por encargo...

	Precio	Horas invertidas	Total
Honorarios del ingeniero	40 €/h	16	640 €
Materiales	1.222 €	N/A	1.222 €
Precio Total			**1.862 €**

Turno de preguntas