



# HACIENDO PROGRAMABLE Y ESTABLE CON FPGA UN DRONE COMERCIAL



Trabajo Fin de Grado

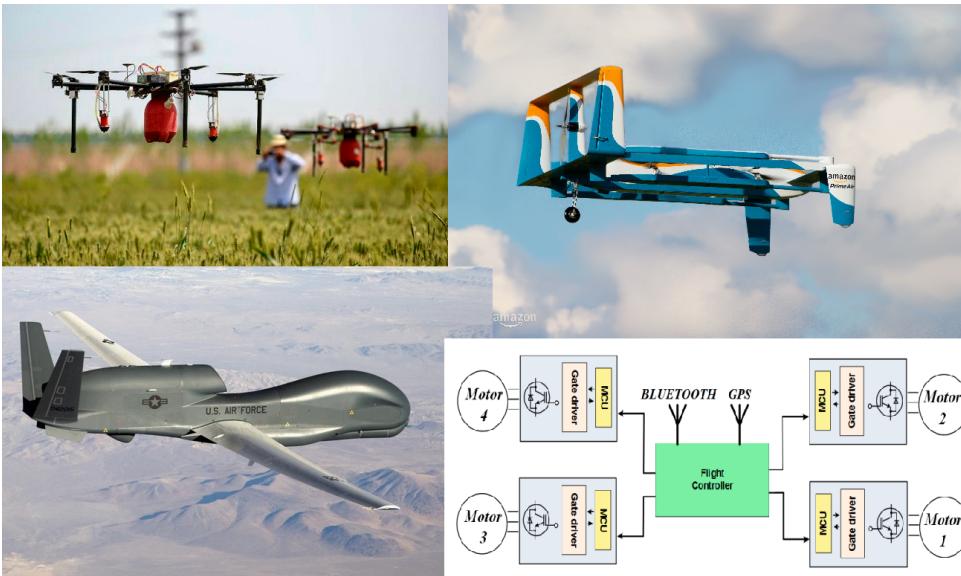
*Eloy Navarro Morales*

# Índice

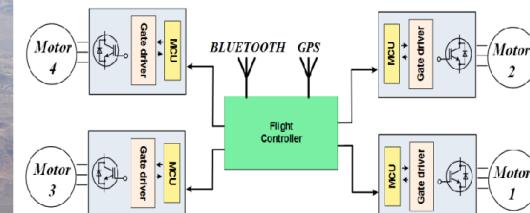
- Introducción
- Objetivos
- Infraestructura
- Dron comercial programable
- Validación experimental
- Conclusiones

# Introducción

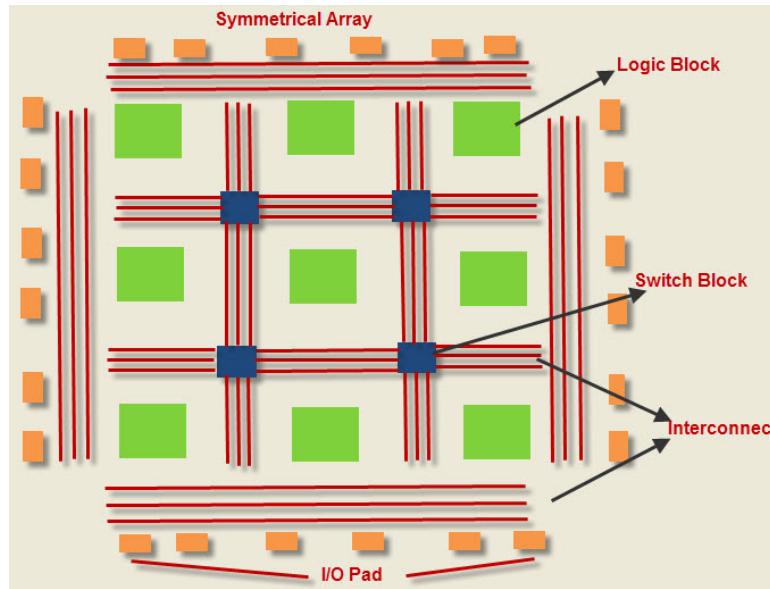
## Drones



- Aplicaciones en entorno civil y militar
- Sistemas de control variados según aplicación



## FPGAs



- Potencia paralela
- Reconfiguración
- Desarrollo abierto



# Objetivos

Hacer **estable y programable** el vuelo de un dron comercial de bajo coste haciendo uso de FPGAs libres

Sub-objetivos:

- Sensorizar vehículo y enlazarlo con tierra
- Diseñar estación de tierra en comunicación con dron y PC
- Diseñar software para PC de mando en Python

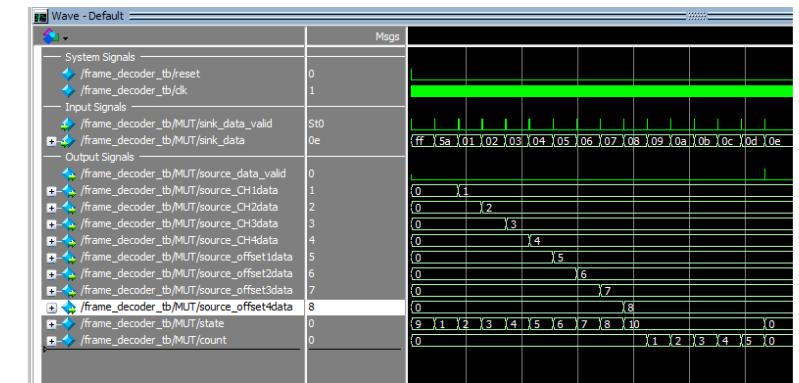
## Requisitos

- FPGAs **libres**
- Control en tres grados de libertad independientes
- Manejo en tiempo real
- Reprogramación de parámetros de control
- Instrucciones de manejo de **alto nivel**
- Conjunto de **bajo coste**

# Infraestructura

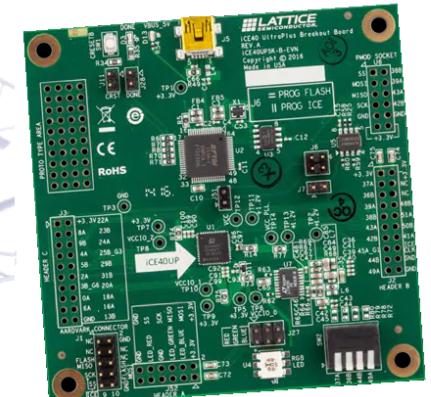
## Software

- IDEs: Quartus, IceCube2, ArduinoIDE
- Simulación: **ModelSim**
- Programación y depuración: Diamond, FT\_Prog, **Logic**



## Hardware

- Plataformas: Arduino, **ICE40**, Eachine E010, SYMA X5C
- Comunicaciones: NRF24L01
- Sensores: Flow breakout board



# Dron comercial programable

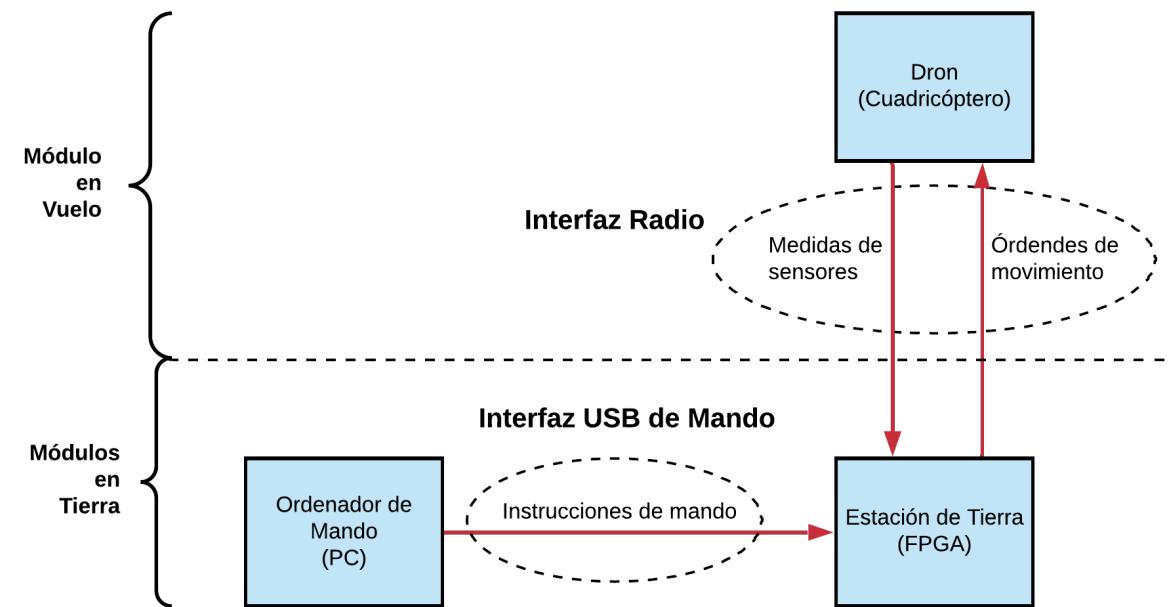
Sistema basado en sensorización adicional a bordo del dron e infraestructura de control en tierra

## Módulo en vuelo

- Dron enriquecido

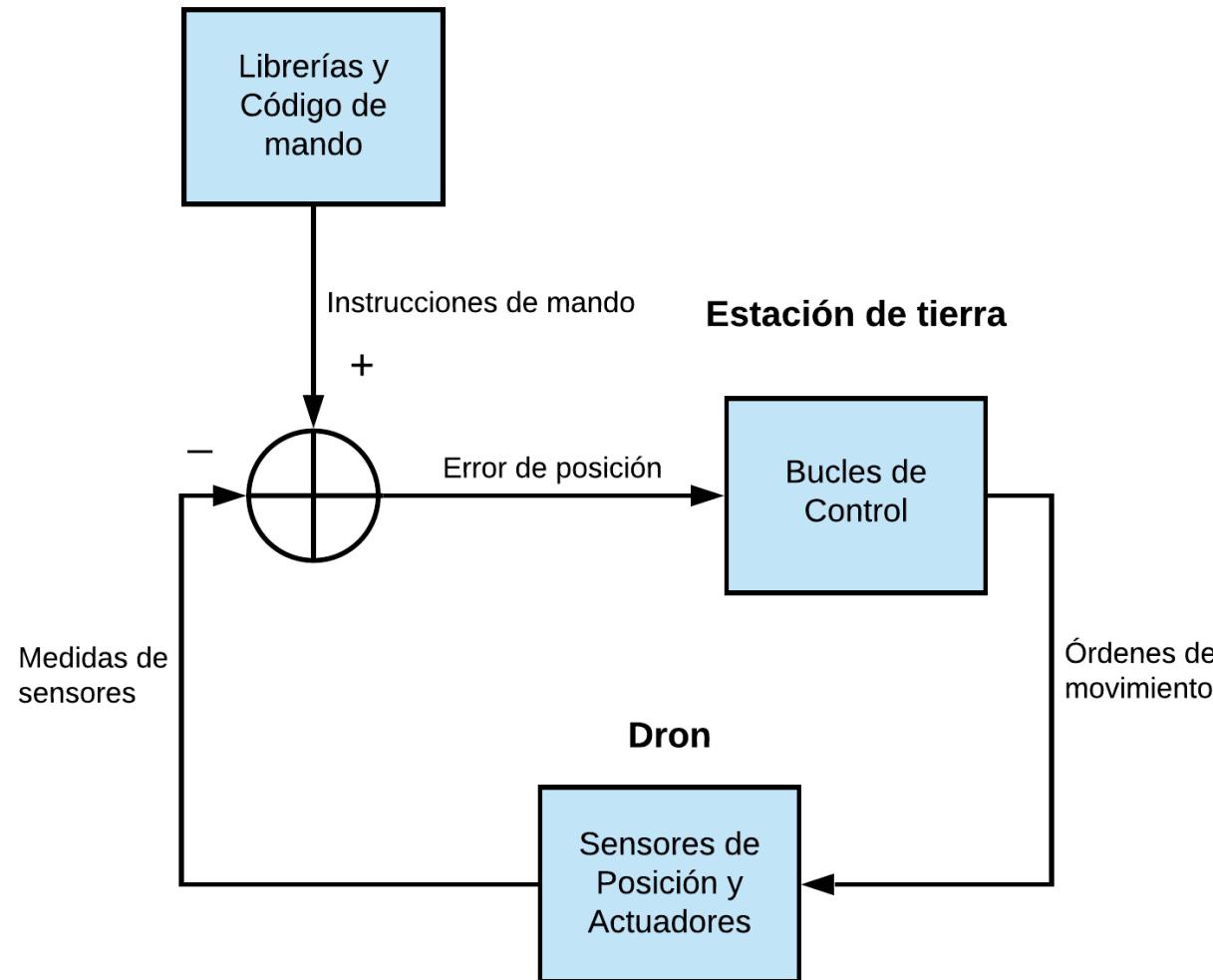
## Módulos en Tierra

- PC de mando: Python para el manejo del vehículo
- Estación de tierra con FPGA para control (Verilog)

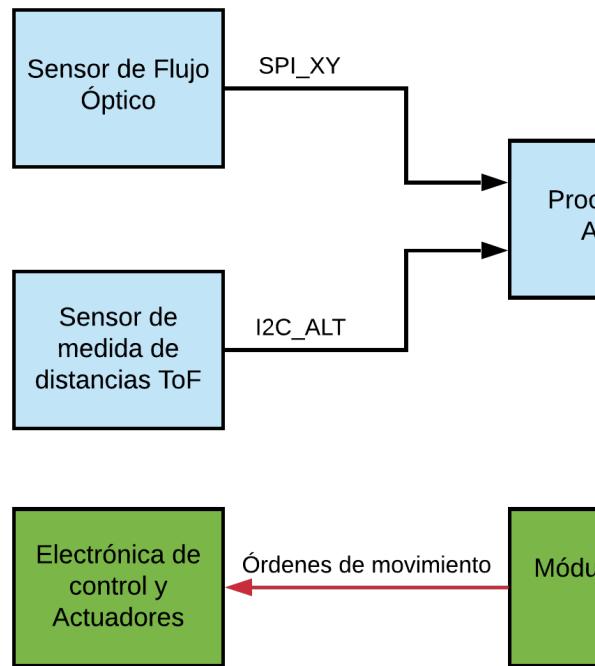


## Flujo de control

### Ordenador de mando

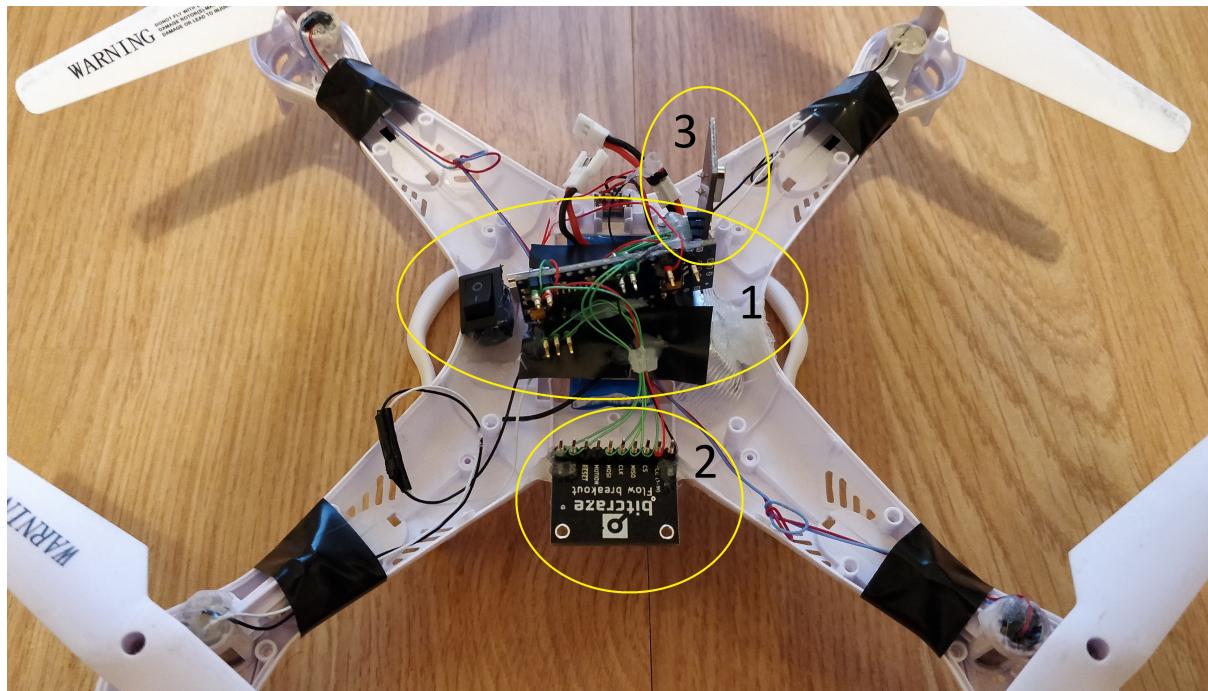


## Dron - Arquitectura



- Sensoriza:
  - Distancia al suelo
  - Posición horizontal
- Transmite medidas
- Obedece órdenes de movimiento

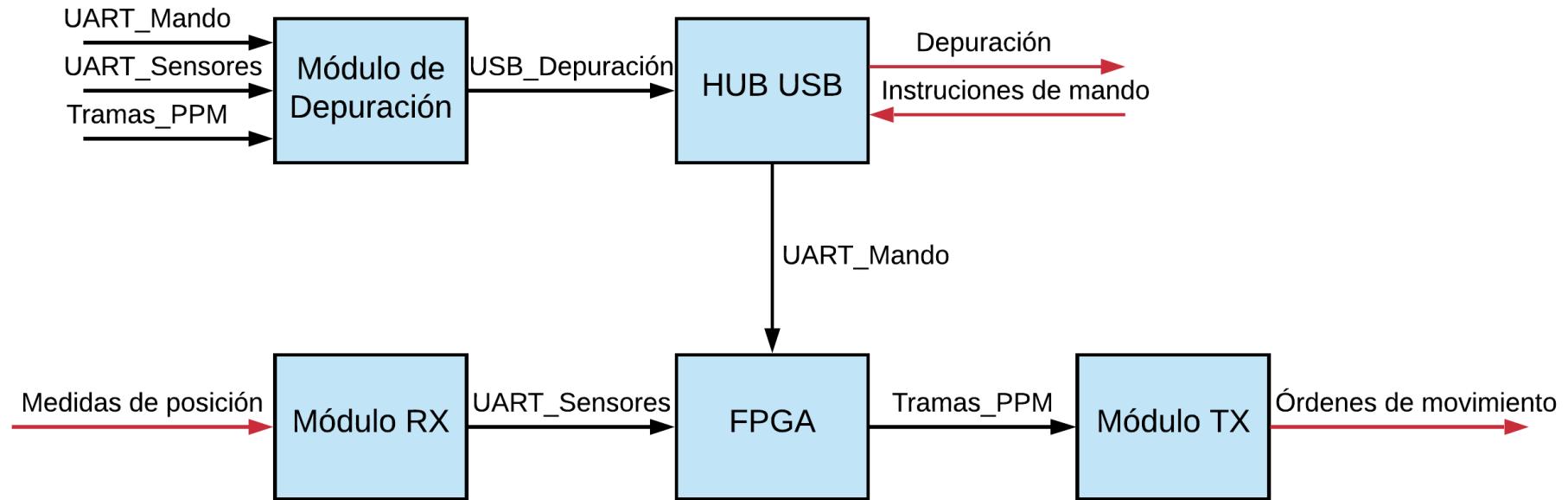
## Dron - Instalación



### Módulos:

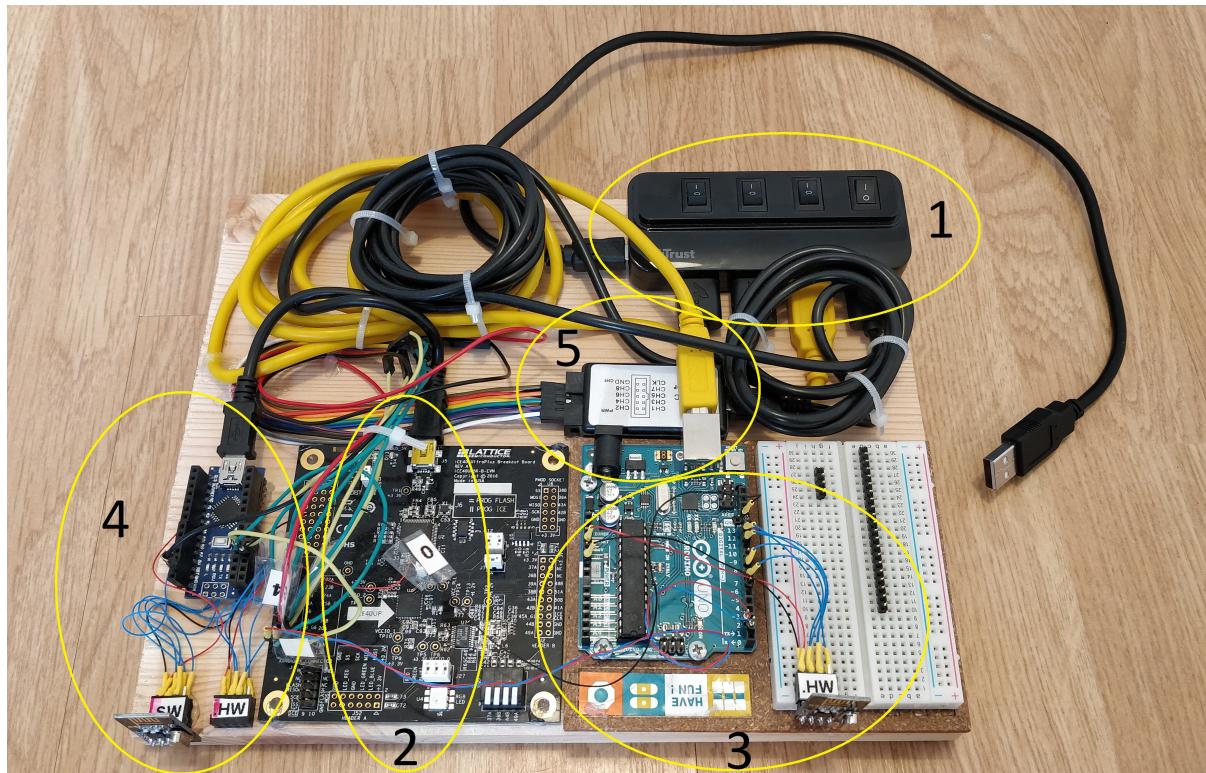
- 1: Procesador
- 2: Sensores
- 3: Transmisor

## Estación de tierra basada en FPGA - Arquitectura



- Atiende tramas de mando
- Atiende tramas desde dron
- Ejecuta algoritmos de estabilización y control
- Transmite órdenes de movimiento
- Encapsula depuración por USB

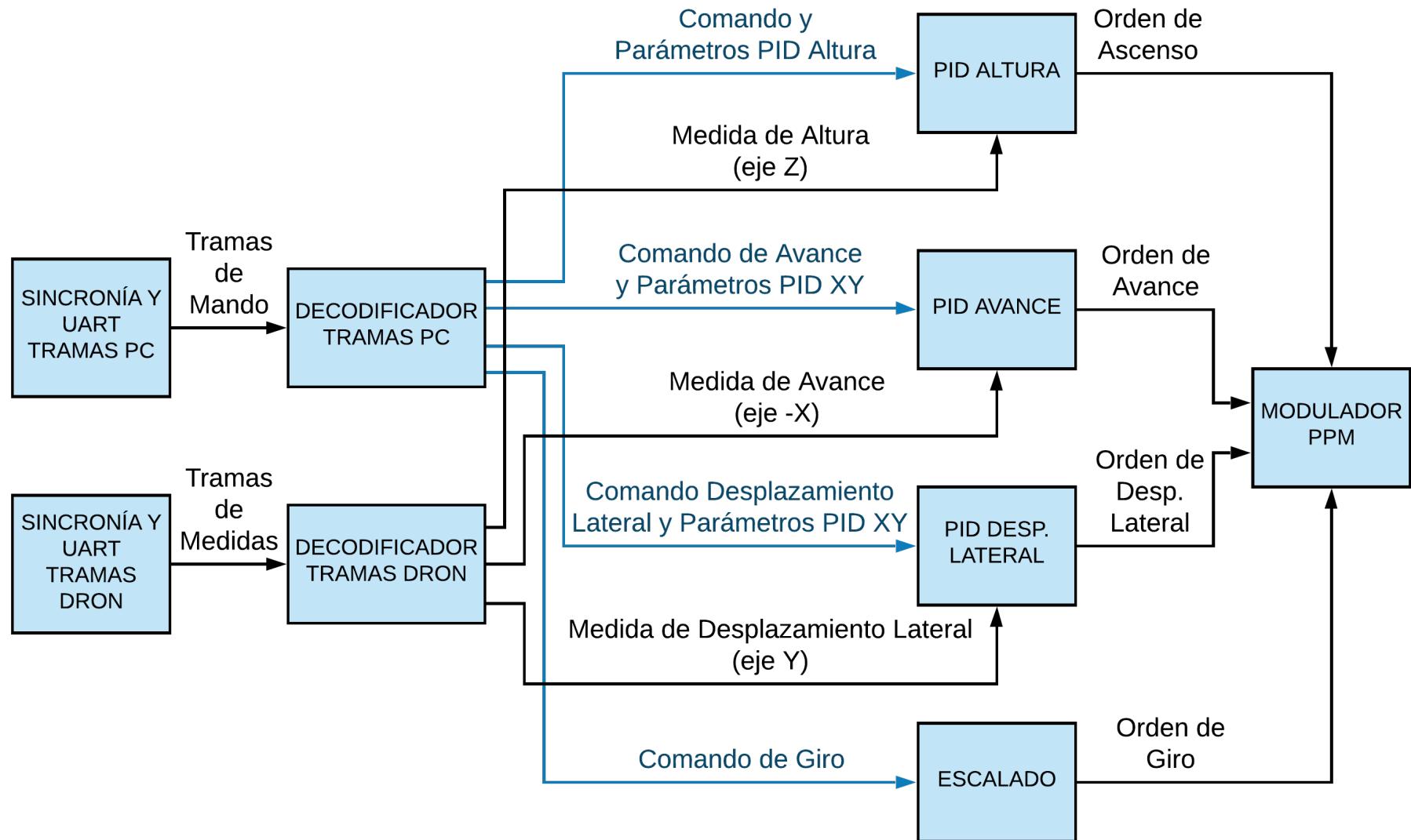
## Estación de tierra basada en FPGA - Montaje



### Módulos:

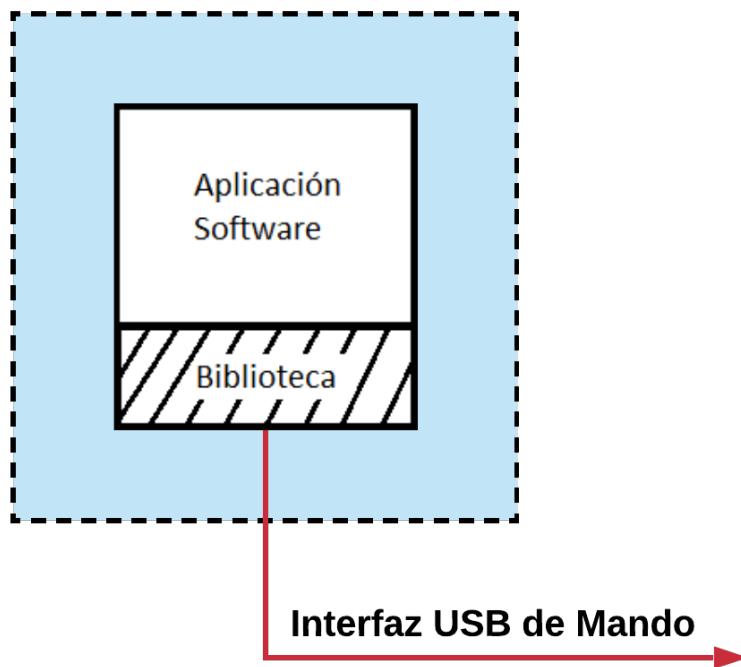
- 1: Hub USB
- 2: FPGA
- 3: Receptor
- 4: Transmisor
- 5: Depuración

## Estación de tierra basada en FPGA - Algoritmo de control



## Ordenador de Mando

Ordenador de Mando  
(PC)



- Biblioteca en Python
- **Facilita el control** del vehículo desde programas ejecutados en el ordenador
- Se comunica con la estación de tierra mediante USB

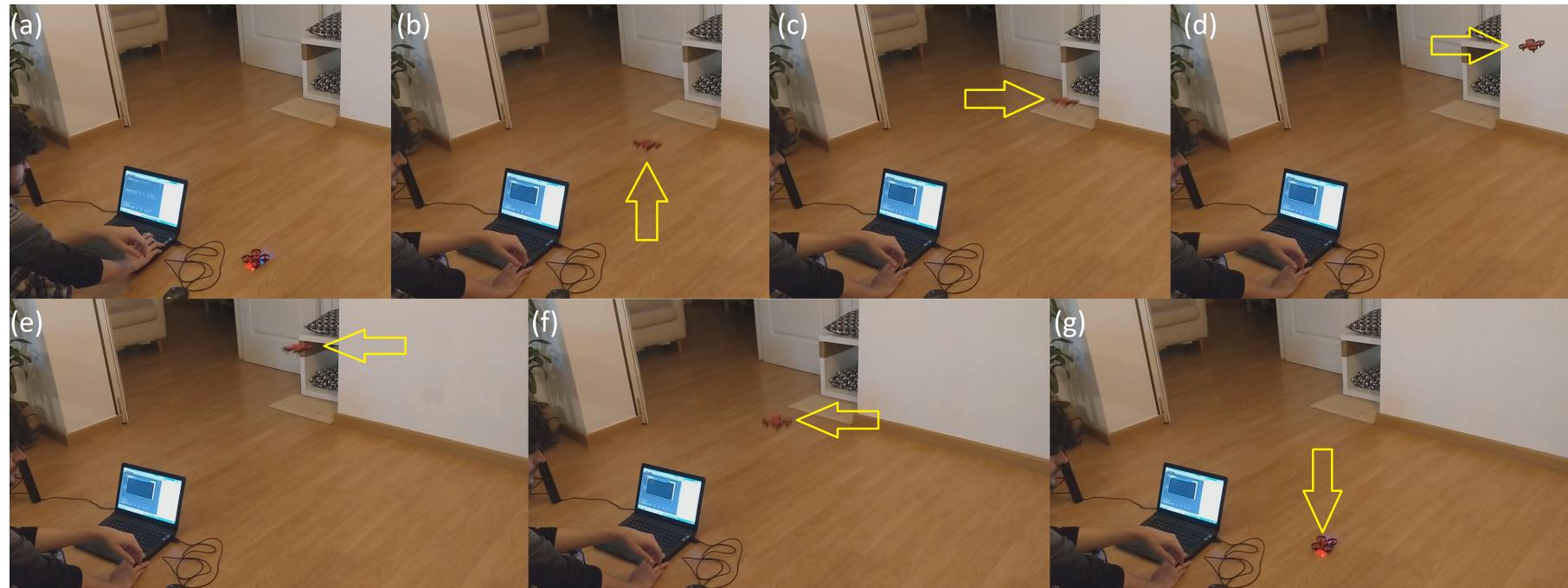
## Ordenador de Mando - Funciones

setPIDValues	Inicialización de valores de parámetros PID y ajuste de giro.
setcontrols	Permite el control directo de los 4 grados de libertad disponibles en el vehículo.
settrace	Dirige una trayectoria progresiva entre la posición actual y el punto indicado.
setcircle	Dirige una trayectoria progresiva en forma de arco entre la posición actual y el ángulo indicado.
takeoff	Arranca los motores y realiza el despegue hasta la altura indicada.
landing	Disminuye la altura progresivamente hasta el contacto con el suelo y apagado de motores.

# Validación experimental

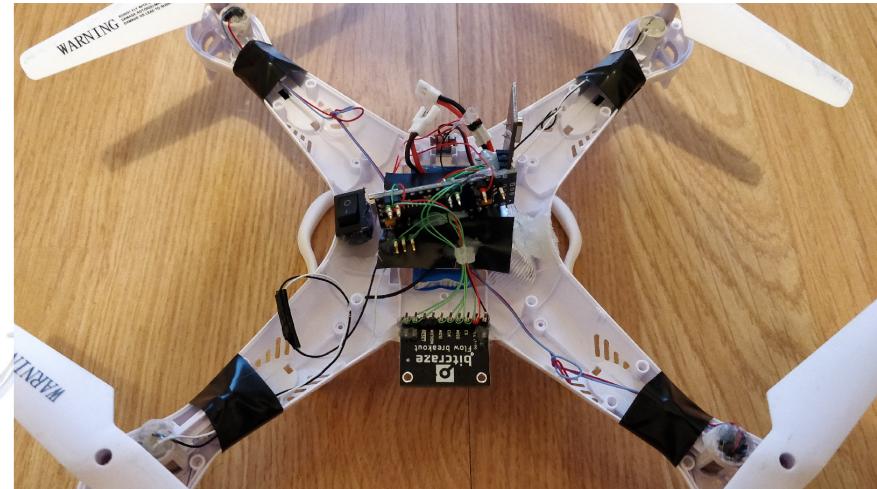
## Dron Eachine-E010

- Ensayos de enlace radio
- Pruebas en bucle abierto

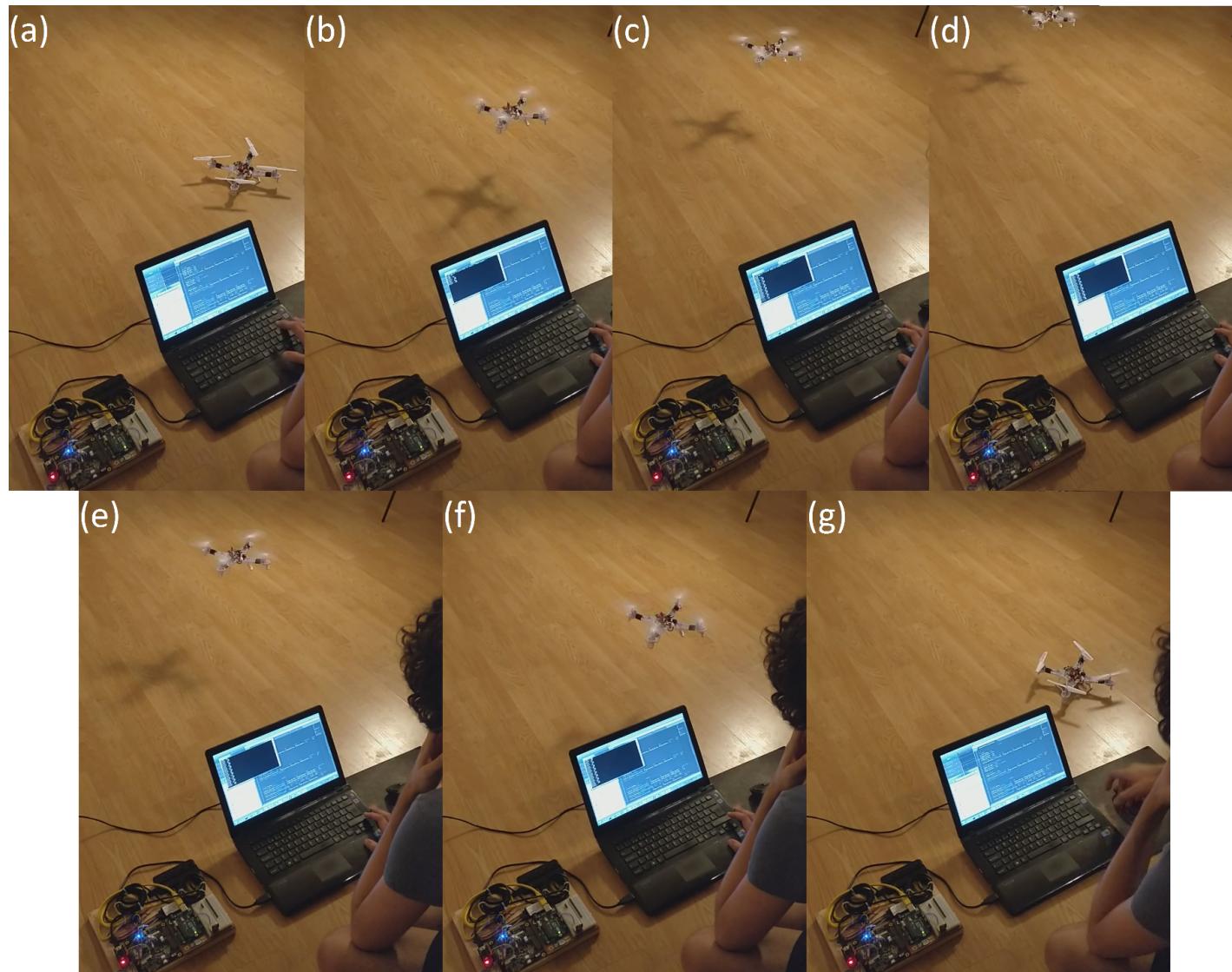


## Dron Syma-X5C

- Modificaciones mayores
  - Aligerado
  - Sensorización
  - Transmisión
- Pruebas en bucle abierto y cerrado

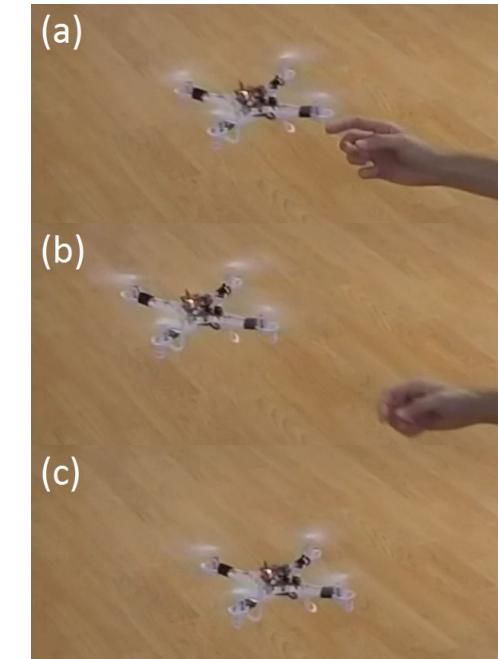


## Dron Syma-X5C Vuelo en Bucle Cerrado



## Conclusiones y líneas futuras

- Cumplido el objetivo general de este TFG y sus tres subobjetivos
  - Enriqueciendo un dron de **bajo coste**
  - Construyendo una estación de tierra basada en una **FPGA libre**
  - Programando el vuelo desde un PC
- Manejo básico con bucles abiertos
- Resultado satisfactorio con **bucles cerrados**



## Líneas futuras

- Añadir sensores: Visión artificial, GPS...
- Cambios en la arquitectura: FPGA a bordo