Diapositiva 1. Portada

**Título:** Desarrollo e integración de una FPGA en un dron con telemetría de altitud  
**Subtítulo:** Proyecto de la asignatura Robots Aéreos y Submarinos  
**Autores**

Diapositiva 2. Introducción

**¿Qué vamos a hacer?**  
En este proyecto, modificaremos un dron de fábrica para integrar una FPGA que nos permitirá procesar y visualizar en tiempo real la altitud durante el vuelo.

Para ello, utilizaremos un entorno de simulación (SITL) para validar previamente el sistema y una Raspberry Pi para gestionar la visualización de los datos.

**¿Por qué es importante?**Este tipo de integración se aplica en diversos campos, como la navegación autónoma, la vigilancia aérea o la inspección industrial. Además, permite validar la compatibilidad entre hardware y software en sistemas de control de drones.

Diapositiva 3. División del Trabajo

1. Configuración y simulación en SITL (Persona 1): Configuración del entorno de simulación, ejecución de pruebas virtuales y validación inicial del sistema.
2. Desmontaje del dron e integración de la FPGA (Persona 2): Modificación del hardware del dron, instalación de la FPGA y pruebas iniciales de conexión.
3. Control y procesamiento de datos (Persona 3): Desarrollo del software para recibir, procesar y mostrar la altitud en tiempo real.

Diapositiva 4: Configuración y Simulación en SITL - NAT

Para validar la integración del sistema, primero realizamos pruebas en un entorno simulado utilizando SITL y QGroundControl.

1. Configuración del simulador SITL y QGroundControl en el ordenador.
2. Realización de pruebas de vuelo: despegue, navegación y aterrizaje.
3. Simulación de diferentes escenarios de vuelo y recopilación de datos de telemetría.
4. Validación de la correcta transmisión de datos MAVLink, como la altitud.

Diapositiva 5: Desmontaje del Dron e Integración de la FPGA - LU

Para implementar el sistema de telemetría en tiempo real, fue necesario modificar el hardware del dron e instalar la FPGA.

* Desmontaje del dron y análisis de su configuración interna.
* Identificación de los puntos de conexión clave para alimentación y comunicación.
* Instalación física de la FPGA y verificación de su posicionamiento.
* Conexión eléctrica de la FPGA con el sistema del dron.
* Pruebas iniciales para confirmar que la FPGA recibe datos correctamente.

Diapositiva 6: Control y Procesamiento de Datos - NURIA

El software desarrollado en la FPGA o Raspberry Pi permite procesar los datos de telemetría del dron y mostrarlos en tiempo real.

* Implementación de código para recibir datos MAVLink del dron.
* Procesamiento y filtrado de los datos de altitud.
* Integración de un sistema de visualización en pantalla o display externo.
* Validación del funcionamiento en pruebas de vuelo.

Diapositiva 7 : Arquitectura

**Descripción General del Sistema**

Nuestro sistema está compuesto por 4 elementos principales:

1. Drone (Pixhawk-based)

* Es el dispositivo principal que vuela y recopila datos de altitud en tiempo real.
* Utiliza sensores como el barómetro y el GPS para medir la altitud y otros parámetros.
* Transmite los datos a través del protocolo MAVLink hacia la FPGA.

1. FPGA

* Actúa como un intermediario entre la Pixhawk y la Raspberry Pi.
* Recibe datos MAVLink de la Pixhawk, los procesa y los adapta para su transmisión.
* Asegura que la información sea filtrada y compatible con la Raspberry Pi.

1. Pixhawk

* Es el controlador de vuelo del dron.
* Se encarga de ejecutar los comandos de vuelo y recopilar datos de sensores.
* Envía la información de telemetría, incluida la altitud, a la FPGA mediante MAVLink.

1. Raspberry Pi

* Funciona como la interfaz de visualización del sistema.
* Recibe los datos procesados desde la FPGA y los muestra en una pantalla.
* Puede permitir la visualización en tiempo real a través de una interfaz gráfica o aplicación web.

Diapositiva 8. Flujos de datos en el Sistema

El sistema opera mediante la comunicación entre el **dron (con Pixhawk), la FPGA y la Raspberry Pi**, permitiendo el procesamiento y visualización en tiempo real de la altitud del dron. A continuación, se describe el flujo de datos y la interconexión de los componentes:

1. **Dron → Pixhawk**

Conexión: La Pixhawk está integrada en el dron como su controlador de vuelo principal, recopilando datos en tiempo real de sensores (barómetro, GPS, IMU).  
Proceso: Procesa los parámetros de vuelo y genera datos de telemetría, incluyendo la altitud, que se formatean en mensajes MAVLink.

1. **Pixhawk → FPGA**

Conexión: Envía los datos de telemetría a la FPGA a través de una interfaz serial (UART) utilizando el protocolo MAVLink.

Proceso: La FPGA recibe los datos MAVLink, los filtra para eliminar inconsistencias y los convierte en un formato estructurado para su posterior procesamiento.

1. **FPGA → Raspberry Pi**

Conexión: Se comunica mediante un protocolo serial (UART, SPI o I2C) o un bus de datos de alta velocidad.

Proceso: La FPGA transmite los datos de altitud y telemetría procesados a la Raspberry Pi, garantizando su compatibilidad para la visualización.

1. **Raspberry Pi → Pantalla/Interfaz**

Conexión: Conectada a una pantalla LCD, monitor HDMI o una interfaz web para visualización.

Proceso: La Raspberry Pi recibe e interpreta los datos procesados, mostrando en tiempo real la altitud y telemetría para el monitoreo del usuario.

Diapositiva 9. Módulos

Para organizar mejor nuestro proyecto, lo hemos dividido en módulos interconectados, donde cada uno desempeña una función clave en el flujo de datos y operación del dron :



1. Simulación y Validación
2. Hardware y Conectividad
3. Procesamiento de datos
4. Control y Visualización de datos
5. Módulo Simulación y Validación (Nat)

**Responsable :** Natalia

**Rol dentro del sistema :**

* Actuar como un entorno de pruebas antes de la implementación real.
* Generar datos simulados para validar la transmisión de telemetría mediante MAVLink.
* Asegurar que los módulos de procesamiento y visualización pueden recibir y mostrar la altitud correctamente.

**Interacción con otros módulos :**

* Proporcionar datos de telemetría simulados al módulo de procesamiento (lo explicará mi compañera Nuria).
* Verificar que el software de visualización en el Módulo de control y visualización muestra correctamente la altitud simulada.

1. Módulo Hardware y Conectividad (Lu)

**Responsables :** Lucia

**Rol dentro del sistema :**

* Implementar la infraestructura física para conectar la FPGA al dron
* Asegurar que la comunicación entre el dron y la FPGA sea estable.
* Verificar el funcionamiento de sensores y otros componentes que interactúan con la FPGA.

**Interacción con otros módulos :**

* Recibe datos del dron físico y los envía al módulo de procesamiento.
* Coordina la conexión de hardware con el módulo de control y visualización para garantizar que los datos puedan mostrarse en tiempo real.

1. Módulo Procesamiento de datos (Nuria)

**Responsables :** Nuria

**Rol dentro del sistema :**

* Programar la FPGA para interpretar los datos MAVLink enviados desde el dron-
* Filtrar y convertir la información de telemetría en un formato que pueda visualizarse
* Garantizar una comunicación eficiente con la Raspberry Pi.

**Interacción con otros módulos :**

* Recibe datos del dron real o del simulador y los procesa
* Envía información procesada al **Módulo de Control y Visualización**.
* Verifica con el módulo de simulación que los datos simulados y reales sean coherentes.

1. Módulo Control y Visualización de datos (Nuria)

**Responsables :** Nuria

**Rol dentro del sistema :**

* Mostrar los datos de altitud de manera clara y en tiempo real.
* Recibir y gestionar la información proveniente de la FPGA.
* Posibilidad de enviar datos a otro sistema para análisis posterior.

**Interacción con otros módulos :**

* Recibe información procesada del **Módulo de Procesamiento**.
* Muestra los datos en una interfaz de usuario accesible desde una pantalla o aplicación.

Diapositiva 15 : Hasta el momento-Resumen de Responsabilidades

| Módulo | Responsable | Rol Principal | Interacción |
| --- | --- | --- | --- |
| Simulación y Validación | Natalia | Simula datos MavLink y verifica la transmisión de telemetría. | Proporciona datos al módulo de procesamiento y verifica la visualización. |
| Hardware y Conectividad | Lucia | Conecta la FPGA al dron, y gestiona la comunicación física | Recibe datos del dron y los envía al procesamiento. |
| Procesamiento de Datos | Nuria | Interpreta datos de MAVLink y los convierte en información útil | Recibe datos del hardware/simualción y los envía a la visualización. |
| Control y Visualización | Nuria | Muestra la altitud en una pantalla en tiempo real. | Recibe datos procesados y los presenta en un formato accesible. |

Esta estructura modular nos permite organizar el proyecto de manera clara, asegurando que cada parte del sistema funcione de forma coordinada desde la simulación hasta la visualización en tiempo real.

VERSIÓN INGLÉS

Diapositiva 1: Portada

**Title:** Development and integration of an FPGA on a drone with altitude telemetry

**Subtitle:** Project of the subject Aerial and Underwater Robots

**Authors:**

**Date:**

📌 Imagen sugerida: Un dron con un esquema de sus componentes o en vuelo.

Diapositiva 2: Introducción

**What are we going to do?**  
In this project, we will modify a factory drone to integrate an FPGA that will allow us to process and display in real time the altitude during flight.

To do this, we will use a simulation environment (SITL) to pre-validate the system and a Raspberry Pi to manage the data display.

**Why is it important?**This type of integration is applied in various fields, such as autonomous navigation, aerial surveillance or industrial inspection. It also allows validation of the compatibility between hardware and software in drone control systems.

📌 Imagen sugerida: Diagrama del dron con FPGA y Raspberry Pi.

Diapositiva 3: Objetivos del Proyecto

* Integrate an FPGA into a factory drone to improve its processing capability.
* Obtain and process real-time altitude data using the MAVLink protocol.
* Validate the system first in simulation with SITL before physical implementation.
* Show the altitude on a display or transmit it to an external application.

📌 Imagen sugerida: Esquema con los componentes clave (Dron, FPGA, Raspberry Pi, Simulador).

Diapositiva 4: División del Trabajo

In order to develop the project efficiently, we have divided it into three main areas:

1. Configuration and simulation in SITL (Person 1): Configuration of the simulation environment, execution of virtual tests and initial validation of the system.
2. Drone disassembly and FPGA integration (Person 2): Modification of the drone hardware, installation of the FPGA and initial connection tests.
3. Control and data processing (Person 3): Development of software to receive, process and display altitude in real time.

📌 Imagen sugerida: Diagrama con los roles de cada persona y sus tareas.

Diapositiva 5: Configuración y Simulación en SITL (Persona 1)

To validate the system integration, we first performed tests in a simulated environment using SITL and QGroundControl.

Steps to be taken:

* Configuration of the SITL simulator and QGroundControl on the computer.
* Flight tests: take-off, navigation and landing.
* Simulation of different flight scenarios and telemetry data collection.
* Validation of the correct transmission of MAVLink data, such as altitude.

📌 Imagen sugerida: Captura de pantalla de QGroundControl mostrando datos del dron en el simulador.

Diapositiva 6: Desmontaje del Dron e Integración de la FPGA (Persona 2)

To implement the real-time telemetry system, it was necessary to modify the drone hardware and install the FPGA.

Steps to be taken:

* Disassembly of the drone and analysis of its internal configuration.
* Identification of the key connection points for power and communication.
* Physical installation of the FPGA and verification of its positioning.
* Electrical connection of the FPGA to the drone system.
* Initial testing to confirm that the FPGA is receiving data correctly.

📌 Imagen sugerida: Esquema del dron con la FPGA instalada.

Diapositiva 7: Control y Procesamiento de Datos (Persona 3)

The software developed on the FPGA or Raspberry Pi allows the telemetry data from the drone to be processed and displayed in real time.

Steps to be taken:

* Implementation of code to receive MAVLink data from the drone.
* Processing and filtering of altitude data.
* Integration of a visualisation system on screen or external display.
* Validation of the operation in flight tests.

📌 Imagen sugerida: Código en Python con PyMavlink o diagrama de flujo de datos (Dron → FPGA → Display).

Diapositiva 8: Tecnologías Utilizadas

For the implementation of the project, we will use various hardware and software tools:

🔹 Software:

* Ardupilot SITL (Simulation)
* QGroundControl (Control interface)
* PyMavlink (MAVLink data reception)

🔹 Hardware:

* FPGA (Data processing)
* Raspberry Pi (Interface and visualisation)
* Factory drone with altitude sensors

📌 Imagen sugerida: Logos o capturas de las herramientas utilizadas.

Diapositiva 9: Conclusiones y Futuro

* Conclusions:
* The project allowed to integrate hardware and software in a real drone in an effective way.
* The methodology based on previous simulations facilitated the physical implementation.
* Communication with MAVLink was key for real-time data transmission.
* Future:
* Possible improvement with additional sensors or flight prediction algorithms.
* Implementation of data transmission to a mobile application or web server.

📌 Imagen sugerida: Foto o render del dron con su sistema final implementado.