Diapositiva 7 : System Architecture - Overview - NUR

Our system consists of four main components:

* **Drone (Pixhawk-based)**
  + Flies and collects real-time altitude data
  + Transmits data via MAVLink to the FPGA.
* **FPGA**
  + Acts as a bridge between Pixhawk and Raspberry Pi.
  + Ensures data is filtered and compatible with the Raspberry Pi.
* **Pixhawk**
  + Controls the drone’s flight and collects sensor data.
  + Sends telemetry to the FPGA via MAVLink.
* **Raspberry Pi**
  + Serves as the system’s display interface.
  + Receives processed data from the FPGA and displays it.
  + Supports real-time visualization via a graphical interface or web app

Diapositiva 8: System Data Flow

To understand how our system works, it's essential to look at how data flows between the different components.

The communication between the drone (with Pixhawk), the FPGA, and the Raspberry Pi allows us to process and visualize the drone’s altitude in real time.

Now, let’s break it down step by step

1. **Drone → Pixhawk - NAT**

First, we have the Pixhawk, which is integrated into the drone as its main flight controller.

It continuously collects real-time sensor data, including altitude, from components like the barometer, GPS, and IMU.

This data is then processed by the Pixhawk to generate telemetry information and is formatted into MAVLink messages for communication."

1. **Pixhawk → FPGA - NAT**

Next, the Pixhawk transmits this telemetry data to the FPGA via a serial (UART) connection using the MAVLink protocol.

The FPGA plays a crucial role here by receiving, filtering, and structuring the data. It removes inconsistencies to ensure the information is reliable before sending it to the next stage."

1. **FPGA → Raspberry Pi - LU**

The connection between the devices is established using a serial protocol, such as UART, SPI, or I2C, or through a high-speed data bus, enabling efficient and rapid communication. Regarding the process, the FPGA is responsible for processing the data generated by the drone and transmitting it to the Raspberry Pi. This transmission ensures the compatibility of the data with the system, allowing the Raspberry Pi to display it correctly in real-time for analysis and monitoring.

The connéction bíthín the deváices is estableisht yúzing a sérial prótócol, súch as UART, SPI, or I2C, or thru a jái-speed déita bós, énábling ifíshient and rápid comiunicéishon. Rigárding the próces, the FPGA is ríspónsible for prócesing the déita yéneréited báy the drón and transmíting it to the Rásberry Pái. This transmíshon enshúres the cómpatibility of the déita wíth the systém, aláwing the Rásberry Pái to dispélei it coréctly in ríaltime for análisis and mónitoring.

Devices connect via serial protocols or through a high-speed data bus for efficient communication. The FPGA processes drone data and transmits it to the Raspberry Pi, ensuring compatibility for real-time display and monitoring.

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

The Raspberry Pi, in turn, connects to an LCD screen, an HDMI monitor, or a web interface, enabling clear and accessible visualization of the processed data. In the process, the Raspberry Pi receives and interprets the information transmitted from the FPGA, processing the altitude and telemetry data. These data are then presented in real-time, providing the user with precise and continuous visualization for monitoring and analysis.

The Rásberry Pái, in térn, conécts to an LCD scrín, an HDMI móniotr, or a wéb intérféis, enéibling clír and aksésibl vízualizéishon of the prócesed déita. In the próses, the Rásberry Pái risívs and intérpríts the infórméishon transmítid from the FPGA, prósesing the álti tud and telemétrí data. Díz déita ar thén preséntid in ríal-táim, prováidíng the yúser wíth prísais and contíniuus vízualizéishon for mónitoring and análysis.

The Raspberry Pi connects to an LCD, HDMI monitor, or web interface for clear data visualization.

Diapositiva 9 : Outcomes and Benefits

To structure our project, we have divided it into interconnected modules.

In the following, I will explain the first module, the Simulation and Validation module.

1. **Simulación y Validación - NAT**

First, I’ll talk about the objectives we aim to achieve with this module:

* This module serves as a test environment before real-world implementation.
* We generate simulated data and transmit it using MAVLink to validate that telemetry is correctly sent.
* We also ensure that altitude data is properly received and displayed in the system.

Now, this module interacts with other modules in the following way :

* We provide simulated telemetry data to the Processing Module, so that it can interpret and use the information.
* We verify with the Control and Display Module that the altitude is displayed correctly on the interface.

1. Hardware y Conectividad -LU

The physical infrastructure is implemented to enable the integration of the various system components, including the previously mentioned FPGA, drone, Pixhawk, and Raspberry Pi. Within the system, it is essential to ensure stable and efficient communication between these devices. Additionally, the interaction with other modules is managed, ensuring the proper reception of data from the physical drone and its transmission to the processing module. Furthermore, the hardware connection with the control and visualization module is coordinated, guaranteeing that the information is displayed in real-time.

The fízical infrastructur is implémented to enéibl the intégration of the várious systém cómponents, inclúding the príviously méntioned FPGA, drón, Pixhawk, and Ráspberry Pi. Wíthin the systém, it is essencial to enshúr stéibl and eficient comiunicación bitwín theez deváices. Adicionalí, the interacción wíth óther móduls is mánged, enshúríng the próper reciption of dáta from the fízical drón and its transmisíon to the prósessing módul. Fúrthermóre, the hárdware conéction wíth the control and vízualizéishon módul is coórdinated, gárranteeing that the informéishon is disp léid in réaltime.

The infrastructure integrates the FPGA, drone, Pixhawk, and Raspberry Pi, ensuring stable communication. It manages data reception from the drone and transmission to the processing module while coordinating hardware connections for real-time visualization.

1. Procesamiento de datos - NUR

Processes MAVLink data from the drone, filtering and converting it for display. Ensures efficient communication with the Raspberry Pi. Sends processed data to the Control & Visualization Module. Verifies consistency between real and simulated data.

1. Control y Visualización de datos - NUR

Displays real-time altitude data and manages input from the FPGA. Can send processed data for further analysis. Ensures clear and accessible data visualization.