MASTER THESIS

TITLE: Drone-based platform system for reflectometry mapping.
MASTER'S DEGREE: Master's degree in Applications and Technologies for Unmanned Aircraft Systems.
AUTHOR: Francesc Sancho Capdevila
ADVISOR: Mª Eulàlia Parés Calaf
TUTOR: Gemma Hornero Ocaña
DATE: 19.10.2022

ABSTRACT

In recent years there has been an exponential increase in the use of drones for new and different applications. In this project, I wanted to focus on the use of the drone as a new platform for the installation and assembly of a video camera and a GNSS antenna for reflectometry data. With these two examples, we can see the complexity behind the assembly of a new payload that can be extrapolated to any other drone. Starting with the video camera, it was necessary first to install the hardware (Raspberry, wiring, connection to the battery, etc.) and finally the creation of the necessary scripts and the installation of the software. During the process, we faced the problems encountered and the solutions adopted. Due to the equipment available for the thesis, the video transmission was done via Wi-Fi, and the steps to create a hot spot on the drone itself are shown, thus being able to check the quality of the video signal in different configurations after four test flights.

On the other hand, taking advantage of the knowledge acquired from a previous project of the CTTC center, the drone has been used to take GNSS positioning measurements and collaborate on a reflectometry project, demonstrating that measurements that are normally taken by ground personnel can be taken from a drone. A GNSS antenna was installed for that purpose and all the hardware and software steps necessary for the installation were performed, with the particularity that we integrated the antenna as another system of the drone since the same power source was shared, and a separate battery was not used. Finally, three test flights were performed in which the data was analyzed and checked in order to draw conclusions about the test.

Acknowledgement

Thanks to my family for the infinite patience and support, to the UPC university for the opportunity to do this master's degree, and to the CTTC team. Especially thank you to Eulàlia Pares, Miguel Valero, Pedro Espin and the CTTC team for their support and help when carrying out this work.

CONTENTS

ACRONYMS

A

Accelerometer: An instrument that measures acceleration. This is primarily used for flight stabilization purposes.

AGL (Above Ground Level): In aviation, AGL refers to the height or altitude of the drone as compared to the ground surface. If a drone is flying 250 feet above a 100-foot building, the AGL is 350 feet.

Altitude Hold Function: In this flight mode, the drone maintains a consistent altitude through its onboard barometric pressure sensor.

A Mode: Abbreviation for Attitude Mode, wherein the drone will maintain a consistent altitude.

ARM: Advanced RISC Machine, processor family of CPUs

ATC/ATM (Air Traffic Control or Management): A service in which air traffic controllers on the ground direct traffic within controlled airspace.

Autonomous Aircraft: The International Civil Aviation Organization classifies autonomous aircraft as, "unmanned aircraft that does not allow pilot intervention in the management of the flight."

Autonomous Flight: This is flight guided by GPS, without intervention from the pilot.

Autopilot: Drone feature that allows the flight to continue without manual control by the pilot.

Axis: In a drone, this is what the gimbal rotates around.

B

Balanced Battery Charger: Smart technology used to charge and balance LiPo (Lithium Polymer) batteries internally.

Barometric Pressure Sensor: Device used to measure the pressure of the atmosphere. In aircraft, this, combined with barometric readings, measures the drone's altitude.

Bind: The process of enabling the controller to communicate with the selected drone.

BVLOS (Beyond Visual Line of Sight): this is the ability to fly a drone beyond the pilot's line of sight.

C

CG or CoG (Center of Gravity): The drone's centre of balance.

Collision/Obstacle Avoidance: A feature built-in to some drones that allow the aircraft to sense and avoid obstacles, minimizing collisions.

Commercial Drones: A drone that is used for business or profit.

Controlled Airspace: A type of airspace in which air traffic control services are offered depending on the defined dimensions and classification.

Cron command-line utility, it is a job scheduler.

CTTC Centre Tecnològic de Telecomunicacions de Catalunya.

D

Drone: Another term for “unmanned aerial vehicles” (UAVs) or “remotely piloted aircraft” (RPA), covering a wide range of functions.

E

EASA European Aviation Safety Agency.

ECHO Directorate-General Humanitarian Aid and Civil Protection.

EETAC Escola d’Enginyeria de Telecomunicació i Aeroespacial de Catalunya.

Electromagnetic Interference: A disturbance in radio frequency by an external source that disrupts the operation of electronic devices.

Elevation The height of a place above the level of the sea.

ESC (Electronic Speed Control) Regulates the speed and direction of the drone’s motors.

F

Firmware: Located in a drone’s flight controller, this is permanent software updated solely by manufacturers.

Fixed-wing drones: Like airplanes but without a human pilot on board, these are drones that have a non-detachable wing that make the aircraft capable of flight. They are typically larger and more powerful than commercial drones.

Flight Controller: Device used to control how the drone moves, by receiving and processing information from the drone’s sensors.

Flight Envelope: The limits to ranges of motion to ensure that the aircraft remains stable.

Fly Away: Accidental flight outside of the set boundaries of operation.

Fly-Away Protection System: When the communication between pilot and drone is lost, this system will return the drone to the pre-selected area.

FLS: FMS Landing system

FOV (Field of View): The “drone’s-eye” view.

G

GCS (Ground Control Station): A command centre on-the-ground that controls airborne drones.

Geofencing: A predefined set of virtual geographic boundaries that prevents drones from crossing into restricted areas.

Gimbal: A device that allows the drone’s camera to remain stable while moving.

GIS (Geographic Information System): Technology that captures, analyses, and manages spatial and geographic data.

GLONASS (Global Navigation Satellite System): An alternative to GPS technology that defines a drone’s location.

GNSS: Global Navigation Satellite System

GNSS R: GNSS reflectometry.

GPS (Global Positioning System): A satellite-based navigation system owned by the U.S. government.

Gyroscope: A circular device connected to the drone’s flight controller that allows the drone to remain level.

H

Hexacopter or Hexicopter: A drone with 6 rotor arms.

Hobby Drone or Hobby-Grade: Typically, ready-to-fly drones designed for the drone enthusiast.

I

IMU (Inertial Measurement Unit): Measuring device that relies on accelerometers, gyroscopes, and magnetometers to report the drone's orientation.
INS (Inertial Navigation System): Internal method by which a drone determines its own position.
IOPES Indoor-Outdoor Positioning for Emergency Staff
IP Rating or Code: A rating system that codifies the level of protection a drone has against the physical elements.
ISM Industrial, Scientific, and Medical (ISM) radio bands are radio bands (parts of the radio spectrum) internationally reserved for the use of radio frequency (RF) energy for industrial, scientific and medical purposes other than telecommunications.

L

LiPo (Lithium Polymer): The type of battery most commonly found in drones.
LoS (Line of Sight): An FAA requirement for drone operation, this states that the drone must be always visible from the pilot's operating position.

M

Magnetometer: A device inside the flight controller that acts as a compass by measuring the Earth's magnetic field to determine its orientation.
mAh (Milliampere Hours): This is used to measure the power in drone batteries.
MAVLink (Micro Air Vehicle Link): Most found as protocols for communication between drones and ground control systems (GCS) to convey orientation, speed, etc.
METAR (Meteorological Terminal Aviation Routine Weather Report): Format used to report specific weather information.
Micro-air vehicle: A small drone. Many have size restrictions.
MSL (Mean Sea Level): A drone's altitude/elevation in reference to the average height of the sea.
Multicopter: A drone with 2 or more rotor arms.

N
NOTAM (Notice to Airmen): Information given to drone operators that warns them of immediate hazards or restrictions not yet published.

O

OAS (Obstacle Avoidance System): System pre-programmed into a drone that alerts pilots to obstacles and collision dangers.
Octocopter: A drone with 8 motor arms.
OSD (On Screen Display): On a drone, this projects flight data like speed, battery life, and the like on-screen.

P

Payload: This is the capacity at which a drone can handle additional weight, such as cameras.
PDB (Power Distribution Board): The link between a drone's battery and other aircraft components.
Photogrammetry: Drone photography that measures the distance between objects.
Pitch: A term for moving the drone up or down.
P Mode (Positioning Mode): The most popular flight mode, this activates all sensors to ensure stability while hovering.
POI (Point of Interest): Available on most drone models, this flight mode keeps the drone's camera trained on an area or object.
PPK (Post-Processing Kinematic): A satellite positioning technology similar to RTK.
Pre-Flight Planning: A set of activities before taking off, including checking the weather, equipment, and flight path.
Props (Propellers): An essential component of a drone, featuring angled blades that allow it to fly.
PUTTY free and open source-terminal emulator, serial console and network file transfer.

Q

Quadcopter: A drone with 4 rotor arms.

R

Radio: This transmitter controls the drone's movements such as pitch, yaw, and roll.
R/C or RC (Radio Controlled): Operating a drone via radio waves.
Receiver: In FPV, this is what links the drone camera's live stream to the goggles. Also called a "video receiver" (RX), "transmitter" (Tx) or "video transmitter" (VTX).
Relative altitude: is the height above Home Point (i.e., the take-off point) at which the drone flies.
Roll: Moving the drone side to side laterally.
Rotorcraft: A drone that generates lift through rotor blades instead of wings.
Rpanion: Australian company offering hardware and software solutions for autonomous vehicles specialised in integration with MAVLink-based flight controllers.
RPAS (Remotely Piloted Aerial/Aircraft System): A combination of a drone, pilot, and its command systems.
RPM (Revolutions Per Minute): The unit of measurement used to determine the number of times the drone's motors turn in 1 minute.
RSSI (Received Signal Strength Indicator): The unit of measurement used to estimate the radio signal strength between a controller and a drone.
RTF (Ready-to-Fly): Great for beginners, this drone requires no (or minimal) assembly and can be flown right out of the box.
RTH (Return to Home) or **RTL** (Return to Launch): An automated drone feature that grounds the drone at the selected home point.
Rudder: Another term for "yaw." This is rotating the drone clockwise or counterclockwise.

Rx (Receiver): Component built-in to a drone that receives and interprets radio signals.

S

SDR: Software Defined Radio (SDR) is a radiocommunication system where several of the components typically implemented in hardware (mixers, filters, modulators / demodulators, detectors, etc.) are implemented in software, using a personal computer.
Sense And Avoid: If a drone is enabled with this technology, it will automatically steer away from obstacles and other aircraft.
Servo (Servomotor or Servomechanism): Drone component that assists with positioning and acceleration. Sometimes also called an "actuator."
SNR: The signal-to-noise ratio, SNR, S/N, is used to define the sensitivity performance of radio communications equipment. Signal to noise ratio defines the difference in level between the signal and the noise for a given signal level. The lower the noise generated by the receiver, the better the signal-to-noise ratio.
SORA: Specific Operation Risk Assessment.
Spatial Awareness: The capability of a drone to be aware of its boundaries and positioning.
Spotter: When flying in FPV, this person keeps the drone in their visual line of sight (VLOS).

T

Telemetry: The communications stream between a drone and its ground control system (GCS).
Thrust: The amount of force driven by the throttle that generates lift.
Trim: The buttons on the side of a drone's remote control that manage roll, pitch, yaw, and throttle.
Tx (Transmitter): In FPV, this is what links the drone camera's live stream to the goggles

U

UART: Universal Asynchronous Receiver/Transmitter
UAS: (Unmanned Aircraft System): An FAA term for unmanned aircraft, operated by a pilot on the ground. This includes drones. Also called an "unmanned aerial vehicle" (UAV).
UAV :(Unmanned Aerial Vehicle): An FAA term for unmanned aircraft, operated by a pilot on the ground. This includes drones. Also called an "unmanned aircraft system" (UAS).
Ultrasonic sensors: On a drone, these sensors calculate height and balance.
Uncontrolled Airspace: Term used by the FAA to indicate class F and G airspace, unregulated by air traffic control (ATC).
U-Space is a set of new services and specific procedures designed to support safe, efficient and secure access to airspace for large numbers of drones.

V

VLOS (Visual Line of Sight): An FAA requirement that the drone in operation remains visible by the pilot's naked eye, unaided by devices.

VO (Visual Observer): An optional teammate that aids the pilot by scanning the airspace for hazards while the drone is in flight. Can also be called a “spotter.”
VTOL (Vertical Take-off): This is the ability of the drone to take off upright, as opposed to fixed wing aircraft. Also called “upright launch.”

W

Waypoints: Helpful in creating flight paths, these are GPS coordinates used to define a point in space.
WiFi FPV: This type of FPV is common with more inexpensive drones. Typically, the pilot connects to a mobile app to receive the drone’s live stream.

X

XWing: Refers to the x design on a drone’s framework.

Y

Yaw: Rotating the drone clockwise or counterclockwise. Another term is “rudder.”
YWing: Refers to the Y design on a drone’s framework.

LIST OF FIGURES

CHAPTER 1. INTRODUCTION

1.1 Motivation

This master's has been developed at the Technological Centre of Telecommunications of Catalonia (CTTC) found in Castelldefels (Barcelona). The main activities are research in technologies related to communication systems and Geomatics. The idea of the project came from how drone technology could be used to collect and study the position error of the signals obtained by satellites without having to use terrestrial means and save human and economic resources. On the other hand, we wanted to take the opportunity of having a drone to install new sensors and perform the process from scratch, thus learning all the steps and problems that are encountered and completing the learning started during the master's degree.

1.2 Goals

The key goals of this master thesis are described below:

1. Choose according to our interest the range of frequencies to detect.
2. Install complementary hardware and software like Raspberry and video camera.
3. Adapt the software and hardware for the chosen tests (GNSS signal)
4. Do the necessary tests on ground and in the air before the check flight.
5. Once checked, install the hardware and software in the drone (antenna and software into SD)
6. Make flights with the drone in an area approved for the flight and carry out different tests.
7. Collection of evidence and subsequent analysis
8. Results presentation
9. Conclusions

1.3 Technologies involved

1.3.1 Hardware

- Antenna and receiver U-Blox to detect GNSS signal from ground reflect.
- Drone Hexsoon 450
- Raspberry Pi 4 Model B 8GB
- Intel Real Sense D435 camera

1.3.2 Software

- Virtual Machine 6.1 with Linux OS Ubuntu 3.8.1.0 64bit
- Python 3
- Raspberry Pi Image
- GNSS Parser
- Mission Planner, ground control station for flight assistance.
- Google Earth Pro
- PUTTY free and open source-terminal emulator, serial console and network file transfer.
- Rpanion (software solutions for autonomous vehicles).

Hardware

Antenna to detect GNSS signal from ground reflect.

This antenna was built for the reflectometry project by PhD Pedro Espín and Fermín Mira of CTTC. The function of this antenna is to capture the GNSS signal bouncing off the surface.

Fig. 1.1 GNSS antenna made by Pedro Espin and Fermín Mira at CTTC for the project. Francesc Capdevila CTTC (03-2022). The left and center photos are the reflectometry antenna, the right photo is the receiver.

Drone Hexsoon 450

The drone used has been provided by the UPC within the Transversal Project and was developed during the first months of the master's in applications and Technologies for Unmanned Aerial Systems.
The drone is a Hexsoon 450 model, a mainly educational model but equipped with GPS, camera, altimeter, optical sensor, etc. The development of the project, the elements and sensors of the drone can be consulted in this blog.
https://med.upc.edu/team4-2021/

Fig.1.2 Drone Hexsoon 450 provided by EEATC, classroom at master UAV, Francesc Capdevila, Castelldefels (10-2021)

Raspberry Pi 4 Model B 8GB

Specifications: https://www.raspberrypi.com/products/raspberry-pi-4-model-b/specifications/

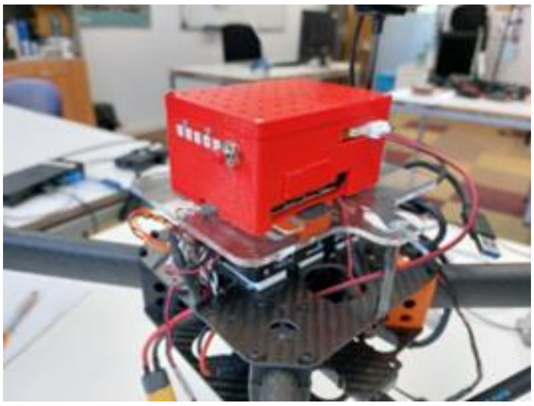

Fig.1.3 Raspberry Pi 4 Model B 8GB mounted on top of the drone.

Intel Real Sense D435 camera

Fig.1.4 Intel Real Sense Camera D435

In this project only the RGB camera module is used, although this model offers many more possibilities (object detection) especially in the field of robotics. https://www.intelrealsense.com/depth-camera-d435/

1.4 Planning

The project started in October 2021. As a research topic, it was proposed to carry out a study in the field of the existing radio frequency spectrum around an airport facility, coming from fixed or mobile stations such as airplanes. This first part was finished the first week of January, and at the end of February the work started on the final thesis, after the signature of the master's thesis by the university, the CTTC, and the student.

In April the necessary software and hardware were available and installed and the research began in the study of the frequencies that seemed most interesting to work, due to the limitation of time and resources, only two frequencies were chosen, the most used by drones, 2.4Ghz for Wi-Fi video transmission and GPS L1C/A, GLONASS L10F (Table 2.1 and 2.5) for the GNSS signal. At the end of May, testing began with the GNSS antenna and the laptop, although the drone was not yet ready. In May, the video camera (Intel Real sense 435), Raspberry 4 and GNSS antenna were installed on the drone, this process included software installation, wiring and system checks. During the transversal project at the university, there were two activities that were not required but were very interesting to perform, it was decided to include these two activities in the project. The first activity was to install the software that detects if the Raspberry is powered by the battery or not, the installation was completed at the end of July. The second activity was to perform several simulated flights to obtain the best Wi-Fi transmission configuration, and once accepted, perform a real flight in the Dronelab with that configuration, the tests were completed in August. Finally, reflectometry tests, data comparison, results, and conclusions were completed in September.

CHAPTER 2. RADIO FREQUENCIES USED BY UAV's AND MANNED AIRCRAFT.

2.1 Introduction to unmanned and manned aircraft communication and systems frequencies

This thesis work is mainly focused on the use of W-Fi as an alternative or complementary data system from the drone to the GCS and the use of GNSS signals from the drone platform for reflectometry projects, it is important to know some characteristics of these frequency bands because as we will see later the streaming video transmission, for example, is affected by these characteristics. As an example, DJI mini 3 Pro drone uses a dual band 2.4 GHz (telemetry and radio control) and 5.8 Ghz (video) frequency. However, we will see later that data transmission is not affected by short distances (less than a mile) but video streaming is, because our drone the Hexsoon 450 has only one channel available at 2.4 GHz.

Radio frequencies used by unmanned aircraft.

The most useful bands for C2 (telemetry) and useful data links are the ISM (Industrial, Scientific, and Medical) without license bands, the amateur radio license bands and the short supplies device bands (SRD).

Table. 2.1 The most frequent frequency bands used are the following:

• 2.4GHz (telemetry and radio control) and 5.1GHz to 5.8 GHz (most for video transmission) for commercial small drones and more sophisticated

and expensive drones, according to the standard Wi-Fi. Utilizing the 2.4 GHz ISM band for radio control (a part of the C2 link[1]).
- 433 MHz (ISM to Europe) or 915 MHz (ISM to the USA) for telemetry (long distance).
- 1.3 GHz (Amateur Radio) or 5Ghz.
- 8 GHz (ISM) bands for video transmission, use standard Wi-Fi rules.

There are other possibilities to use these or even other bands, that's the case of the 868 MHz band (SRD), used in some drones in Europe.

Table. 2.2 Transmission characteristics

- 433 MHz low data rate, very limited band, large antennas, little attenuation.
- 2.4 GHz short range, 1 Nm and band for video, saturated band Wi-Fi and Bluetooth.
- 5 GHz has more attenuation than 2.4 GHz, 5 GHz requires LoS, it has more bands available, no ISM, reduces link budget and interference, more suitable for video transmission.

2.2 Allocation of frequencies, unmanned aircraft

Initially, most of drone communications take place in this green-shaded frequency band.

UNMANNED AIRCRAFT

In this table we can see where most of the telemetry, video and Wi-Fi communications are concentrated.

TLF	ELF	ULF	VLF	LF	MF	HF	VHF	UHF	SHF	EHF
<3Hz	3-30Hz	30-300 Hz	3 kHz-30 kHz	30 Khz-300 kHz	300 kHz-3 MHz	3 Mhz-30 MHz	30 MHz - 300 MHz	300 Mhz-3 GHz	3 Ghz-30 GHz	30 Ghz-300 GHz

Table. 2.3 Allocation of frequencies. Summary table of frequencies used by unmanned aircraft

[1]*New FREQUENCY AWARD: 2380-2390Mhz (2021) National Table of Attribution of Frequencies (CNAF)*
Order ETD / 666/2020, of July 13, which modifies Order ETU / 1033/2017, of October 25, approving the National Table of Frequency Attribution.
"The frequency range 2380-2390 MHz is reserved for communications for the exclusive use of the aeronautical mobile service in the downstream direction from unmanned aircraft systems (UAS). The exploitation of this spectrum will be authorized in a shared way by different users, so that the authorized use of the spectrum must be previously coordinated by them."
"The C2 link is fundamentally used for the telemetry of flight (in addition to a possible radio link for traffic and traffic between Air Traffic Control and other pilots), the ascendant and descending links."

Europe
- 433.05 – 434.79 MHz “433 MHz band”
- 863 – 870 MHz (not an ISM band)
- 2.4 – 2.4835 GHz “2.4 GHz band”
  Most popular for Radio control
- 5.725 - 5.875 GHz, 5 GHz band.

USA/Canada
- 260 – 470 MHz (not an ISM band)
- 902 – 928 MHz “915 MHz band”
- 2.4 – 2.4835 GHz, 2.4 GHz band”
- 5.725 - 5.875 GHz, 5 GHz band.

2.3 Manned aircraft: Systems and communication frequencies

Manned aircraft communications take place in this yellow-shaded frequency band.

MANNED AIRCRAFT

TLF	ELF	ULF	VLF	LF	MF	HF	VHF	UHF	SHF	EHF
<3Hz	3-30Hz	30-300Hz	3kHz-30kHz	30Khz-300kHz	300kHz-3Mhz	3Mhz-30Mhz	30-300Mhz	300Mhz-3Ghz	3Ghz-30Ghz	30 GHz-300 GHz

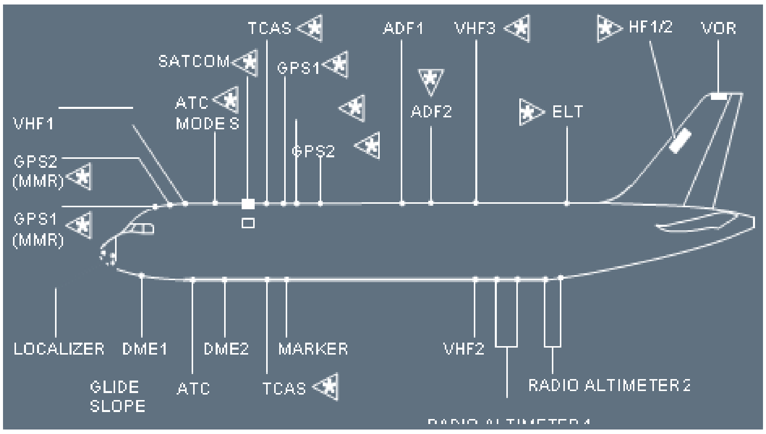

Fig. 2.1 Location of transmitting and receiving antennas on a commercial aircraft. Airbus manuals A320 aircraft, Tech Aviator (01-2022)

Table. 2.4 Summary table of frequencies used by manned aircraft.

In this table we can find all the frequencies used by the equipment installed in a commercial aircraft, type Airbus 320 and whose location is in the previous figure (Fig 2.1).

VOR	108.0-117.95 MHz
HF	2.8 Mhz-24 MHz
ADF/NDB	190-1750 Khz
RADIOBALIZA	75 MHz
VHF (Voice)	118-136.975 MHz
GPS/Galileo/Glonass/MBeidu	1164-1610 MHz
TCAS	1090 MHz AIR
TRANSPONDER SSR	1030 MHz GROUND
DME AIR	1025-1150 MHz
DME GROUND	963-1213 MHz
RADAR PRIMARY (Ground station)	2700-2900 MHz
MLS	5000-5250 MHz
UNMANNED AIRCRAFT CONTROL FAA (USA)	5030-5091 MHz
TLWU Terminal LAN Wireless. Air to Ground, antenna is beneath. Air to Satellite, antenna on top of the aircraft.	For ground-to-ground, 800 MHz The two main frequencies allocated for satellite internet are Ku-band (12–18 GHz) and Ka-band (26–40 GHz)
LOCALIZER	108.1-111.95 MHz
GS	329.15-335.0 MHz
RADIO ALTIMETER	4.2-4.4 GHz
ELT	406 MHz 243 MHz 121,5 MHz
ADS-B	1090 MHz and 978 MHz, 978 MHz in USA below18000’.

2.4 Satellite position for unmanned aircraft

Within the professional world of drones, one of the most important parts to be able to carry out our work is the positioning system. Drones can obtain this information using several types of sensors, e.g., inertial sensors, cameras, radars, and of course satellite positioning signals. In this work, we will focus on the satellite signal since it is the one that can be interfered with and at the same time affects other manned aircraft in the area and how the signal is received, and their error can be also used for reflectometry purposes.

SATELLITES

GNSS Frequencies and Signals

Table. 2.5 GNSS frequency range of the drone used in our project.

System	Signal	Frequency (MHz)
GPS (Global Positioning System)	L1 C/A	1575.42
	L1C	1575.42
	L2 C	1227.6
	L2 P	1227.6
	L5	1176.45
GLONASS	L1 C/A	1598.0625-1609.3125
	L2 C	1242.9375-1251.6875
	L2 P	1242.9375-1251.6875
	L3 OC	1202.025

The drone used in this project uses the Here 3 GPS GNSS module Pixhawk 2, it is a new version based on the Here 2 GNSS GPS unit. Compared to the Here GNSS GPS unit, the Here 3 GPS module has a more accurate positioning, faster response and LED lights, it receives signals from GPS and GLONASS systems, in the table above we can see the frequencies of these two systems. A relevant data within the characteristics of this antenna is the accuracy, 2.5 meters, so it is not a very accurate antenna.

Features:
Accuracy: 3D: 2.5 meters / RTK 0.025 meters.
Processor: STM32F302

IMU sensor: ICM20948
Concurrent reception of up to 3 GNSS (GPS, GLONASS)
Advanced jamming and signal spoofing detection
CAN port
RGB status LED
Accelerometer/ Gyro/ Barometer / Compass
8Hz refresh rate

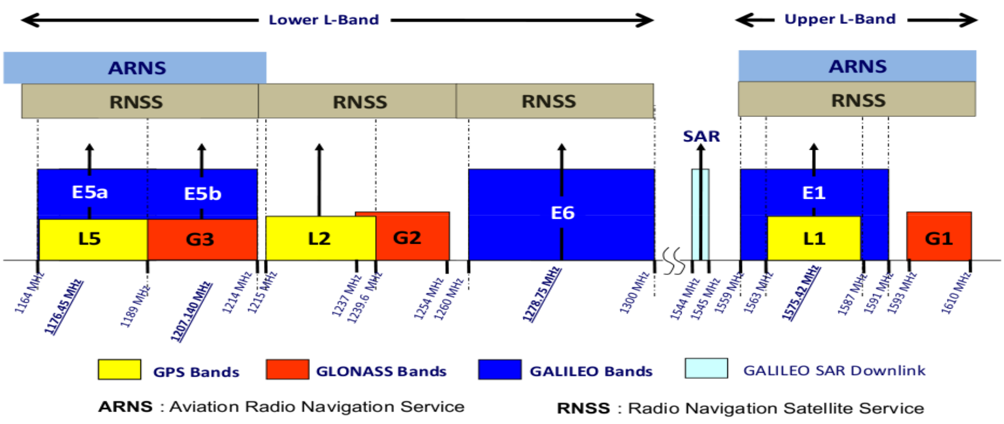

Fig. 2.2 Satellite allocation frequencies, Everything RF (Aug 2021)

This figure shows the frequency distribution of the GPS, Galileo (Europe) and GLONASS (Russia) systems.

2.5 Introduction to GNSS reflectometry (GNSS-R)

GNSS reflectometry involves making measurements from the Earth's reflections of navigation signals from global navigation satellite systems. GNSS reflectometry (GNSS-R) is a remote sensing technique that uses satellite signals bounced off the earth's surface to find out information about natural phenomena.

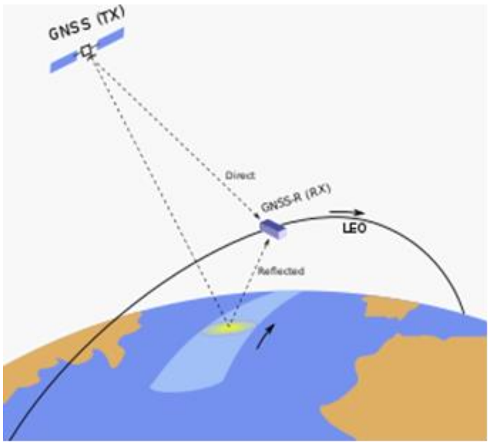

Fig. 2.3 Basic example of how reflectometry works, GNSS system diagram, Carlos Molina (10-2020) Wikipedia.

In Fig.2.3 we can see of reflectometry is performed using satellites in different orbits, from the highest orbit the signal is sent to the earth's surface and another satellite from a lower orbit would collect the bounced signal and the direct signal from the satellite above. GNSS-R research applications are found in altimetry, oceanography (wave height and wind speed), cryosphere monitoring and soil moisture monitoring. GNSS-R can also be applied using a receptor fixed in the ground or mounted in a UAV, applications such as lake altimetry or snow height estimation are widely known.
The idea of the test is to see the results obtained by measuring the GNSS signal from the drone antenna and the external antenna mounted as a payload system in the same drone. The results will not be applied to any surface, we simply want to demonstrate that it is possible to measure using a drone as a platform.

CHAPTER 3. Hardware; Raspberry's wiring installation

Initially, the project was started with two Raspberry, one that was already assembled with the antenna and GNSS receiver, and another provided during the master course. Due to compatibility issues, it was decided to transfer all the software on the memory card from the CTTC Rpi to our Rpi and use this one only as it came with the necessary connections for the drone's flight controller. To communicate with our laptop or with the flight controller there are two phases, the software and the hardware part. In this chapter, we show the easiest, hardware but also the ones that caused more problems due to the fragility of the pins and the little space between them. In addition, the wrong connection of any of the cables causes the Rpi not to communicate with the flight controller.
We have a port that communicates the Rpi with our laptop (base station) this UART port is the middle one (Fig 3.2) and the right UART port (Fig 3.2) that communicates with the power distribution board (Fig 3.4) and gives power to the Rpi from the battery. Pins, number 2 (5V power), number 8 (GPIO 14) and number 10 (GPIO 15) of the right columns (Fig 3.1) are the ones used for the

middle port, and pins number 34 (ground) and number 4 (5V) (Fig 3.1) are the ones used for the right port (Fig 3.2).

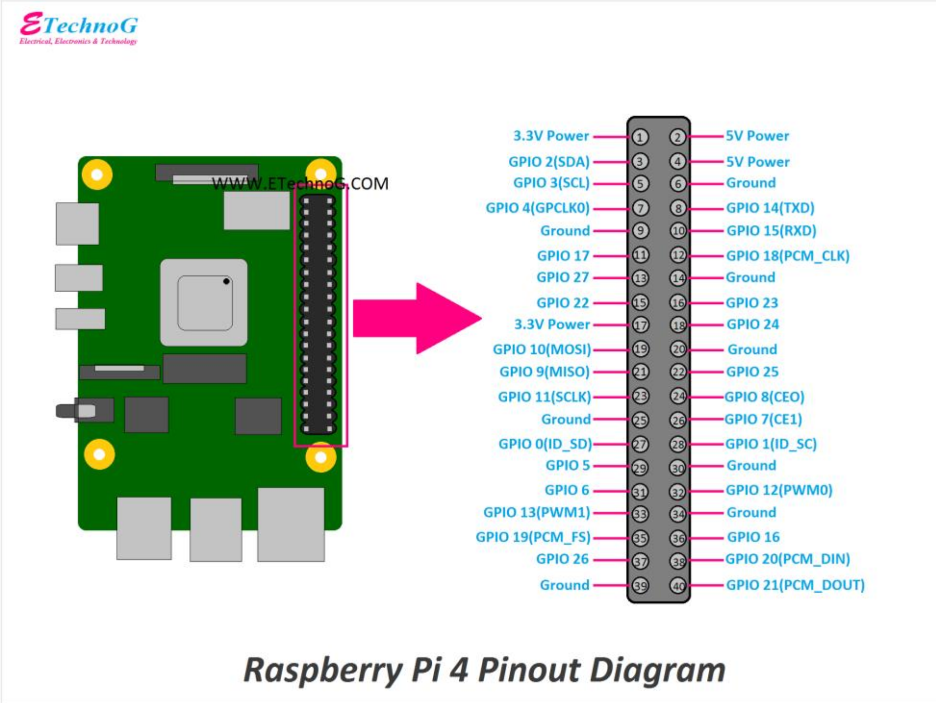

Fig. 3.1 Source: EtechnoG, Pinout configuration RPi4

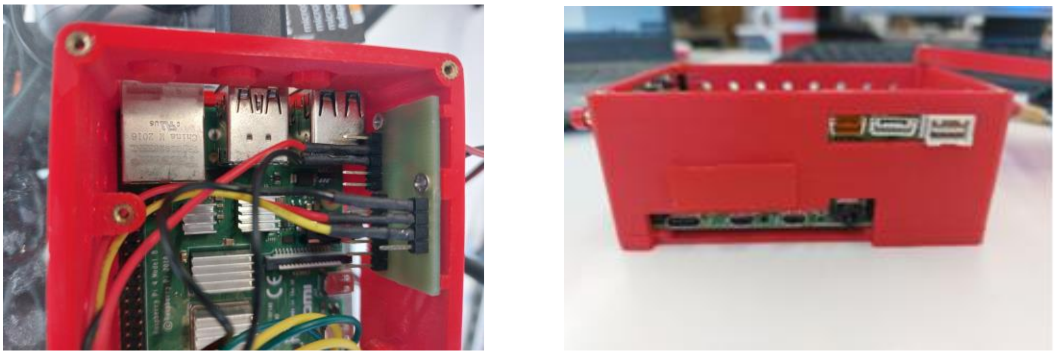

Fig. 3.2 Yellow wire goes to number 8 (GPIO 14) TXD Red goes to number 10 (GPIO 15) RXD and black goes to number 6 (Ground)

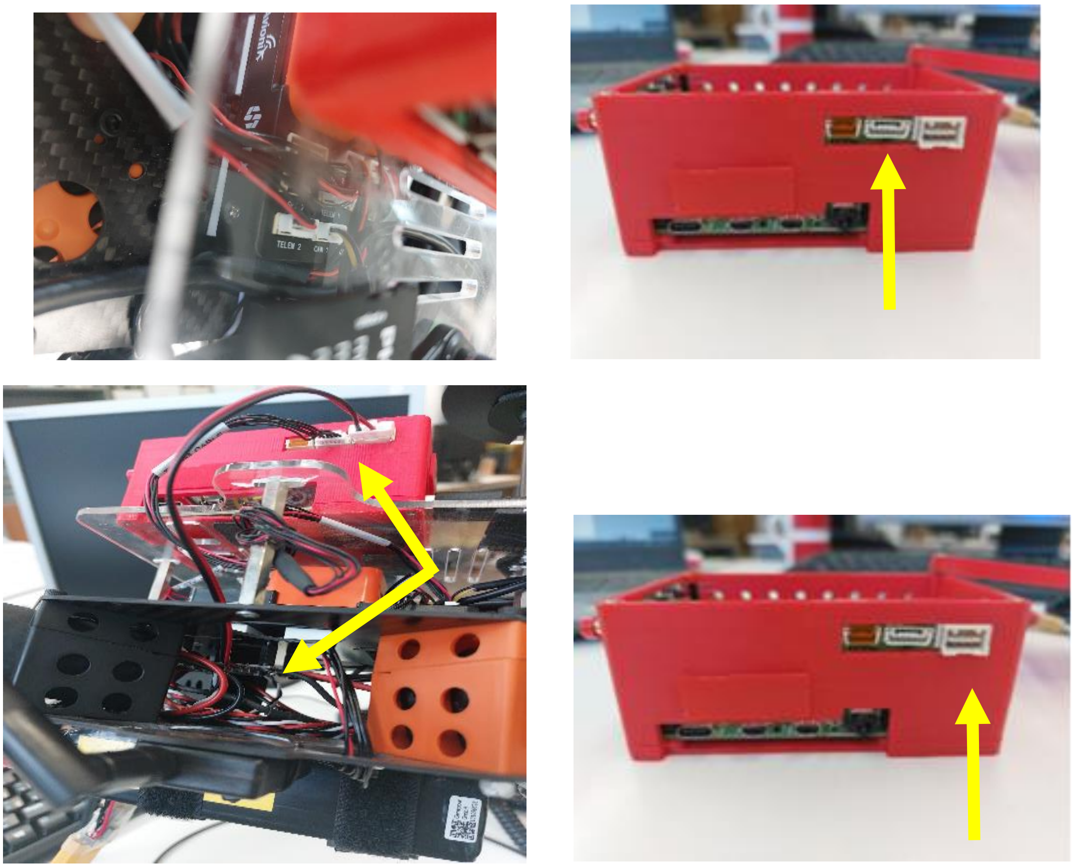

Fig. 3.3 Telemetry2 port from Flight controller goes to middle Rpi UART port. We can see the Raspberry before the installation and after the installation with the two cables connected. The cable on the right is the power cable from the distribution board and the cable on the left is the one that connects to the telemetry port.

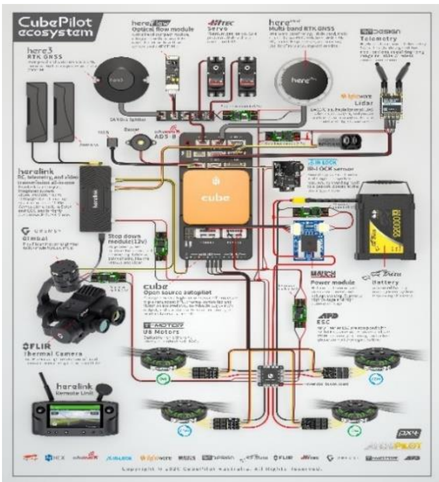

Fig. 3.4 Cube ecosystem. Ardupilot (2021). This photo shows all existing connections and their functions from the flight controller located in the center of the picture, to the power distribution at the bottom right, the GNSS antenna at top center, the ADS (orange cube) or telemetry.

CHAPTER 4. SOFTWARE INSTALLATION

4.1 Raspberry basic configuration

The configuration consists of installing the necessary software from our laptop on the SD inserted in the Rpi and then installing the software so that the Rpi and the autopilot can communicate. As the Rpi was new, everything had to be configured from scratch.

4.1.1 Installation of the operating system and the libraries

An operating system (Raspberry Pi imager) on the SD must be installed initially, this step can be done from a laptop that already has the Linux operating system. Therefore, first VirtualBox was downloaded on the laptop and then Ubuntu 64-bit as the operating system (VirtualBox is an application used to make virtual machines with operating system installations). Once the SD is ready it can be inserted into the RPI and from there works directly. To perform this operation the RPi must be connected to a monitor, a keyboard, and an internet connection. The next step is to download Python3 and OpenCV to obtain the images, object recognition and detection, capture the images and process them. The *libgl1-mesa-glx* library is a 3D graphics library and the *vidgear* library can read, write, process, send and receive video frames from various devices in real time. To send images during the flight from the camera to our laptop (ground station) in streaming (Wi-Fi) two scripts had to be created, one for the Rpi and one for the laptop. The scripts are available in Annex I and Annex II.

4.1.2 Installing the software to run the UART connectors in the RPi and autopilot

As previously mentioned, when developing the assembly from scratch, the RPI ports must also be configured, first with the wiring and after with the necessary software. In the transversal project of the master, there is a chapter that explains in detail the process. We need to install a program called minicom, and once installed, we must set the port name and connection bauds (9600). Once configured, must download another software on your laptop called PUTTY that serves as a link between the Rpi and the laptop to send the necessary information and messages.

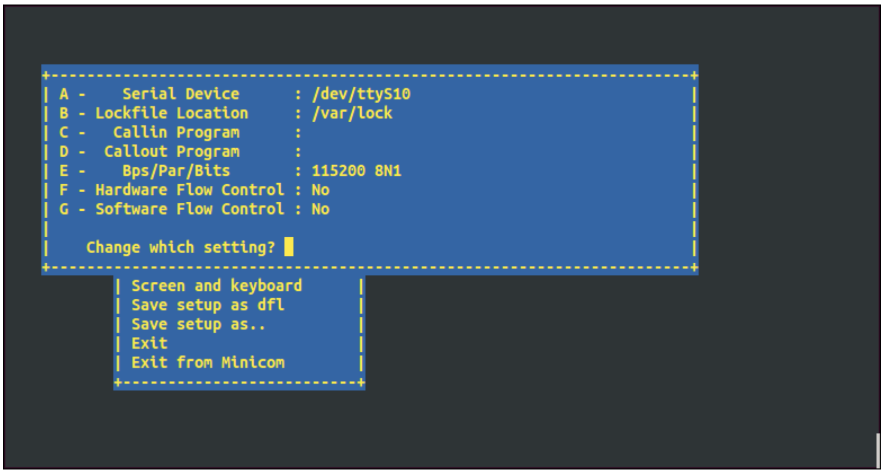

Fig. 4.1 UART minicom configuration screen

4.2 Installation of supplementary software (green LED's)

Within the transversal project, there was an exercise that was decided to perform in the master's thesis for its interest. It was decided to install on the Raspberry 4 the peripherals and the software needed to activate the power status, and for this, the four LEDs already installed in our RPi were used. In addition, the installation of the wiring was necessary to power the camera from the drone battery and not from another independent battery. By doing this installation and not using a separate battery, the weight of the drone was reduced, and we also knew if the power distribution board was powering the RPi.

Fig. 4.2 Wiring necessary to light up the LED's.CTTC Castelledefels, Francesc Capdevila (03-2022) The wires on the left are the ones that connect the LED lights to the Raspberry pins and the UART port.

To run the peripherals of the RPi the library called *gpiozero* must be downloaded, as python was already previously downloaded now, we can go to the terminal and download this library (*sudo apt install python3-gpiozero*).
To get the lights to turn on, the python's script must be installed inside a file that we called exp1.py into the SD card, this file simply turns on the 5 green Leds (255

is the maximum bright value). This script is activated each time the Rpi is powered, the way to run this script each time is by doing a *cron job*. Cron job periodically runs all the commands inside the crontab file. The command to do this is: *reboot sudo python3/home/pi/Desktop/exp1.py*. With the crontab file every time there is a reboot the script is turned on. The script is available in the annex III.
After completing the wiring installation (manual provided by the Transversal project documentation) and connecting the drone with the battery that was the result:

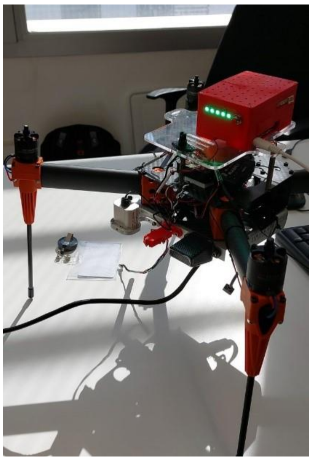

Fig. 4.3 Video LED Francesc Sancho Capdevila. In the following video we can see how after connecting the drone to the battery it energizes the Raspberry, and the software downloaded to the SD card illuminates the LED lights. Both the installation of the UPRT ports and the script to turn on the lights work. https://www.youtube.com/watch?v=QebfCaTs6QM

4.3 Installation of the Intel Real D435 and Wi-Fi configuration

Camera installation consists of two steps. Step One is the preparation of the Rpi and the base station (laptop) through the Mission Planner, and step two is to find the best location for the camera and the connection with the Rpi.

4.3.1 Step one, software installation

The Rpi was brand new, so it was necessary to perform a series of steps. The ports were already prepared and the internal wiring of the Rpi to be able to give power from the drone battery to the Rpi and the camera was done in the previous steps 3.1 and 3.2. To connect the camera to the Rpi the USB 3.0 can be used and connected in the Rpi’s port, so this step was already done, we also knew that it is powered because the LED light is on, and the SD is well connected (Fig 4.4).

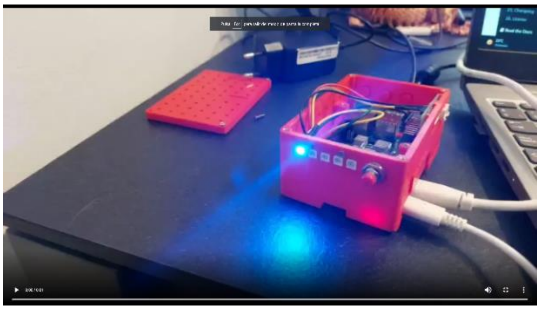

Fig. 4.4 Rpi LED

During the installation of the camera software a new problem appeared, the motherboard of our Rpi is an ARM while our camera is Intel, they are incompatible, for that reason another way to connect the camera with the Rpi and with the Mission Planner was necessary to install. An extra software package was downloaded from the company *rpanion,* with this software a HOTSPOT was created to be able to emit a video signal from the Intel RealSense D435 camera to the installed Mission Planner on our laptop. The version downloaded was 0.8 from *rpanion* website into the SD card of the Rpi, once the SD card was inserted in the Rpi we could start the second phase.

We turn on the Raspberry and connect our laptop to the Wi-Fi network "rpanion".Once we are connected, we open a page in our browser and paste the following address: https://10.0.2.100:3000 (This IP address can be found on the rpanion web page (Fig 4.5)), we go to Video Streaming and choose as device: Intel Real Sense Depth Camera, the first of the two, the first one is the RGB camera, if we choose the second one it would be the black and white camera. The resolution and the bits can be changed to increase the transmission quality of the camera, otherwise, the image is transmitted with much latency. Clicking on Start Streaming some new tabs will appear, must take the last option (Mission Planner Connection Strings).

We copy the second address http://10.0.2.100:8554 and we paste it into the Mission planner PFD, in our laptop, clicking the right button mouse and going to video, a tab appears with several options, we choose video and go to SetGSstreamerSource and we paste the address wrote down above. Doing that we will see the image video as the background in our mission (figure 4.6). The video will start streaming, even with the drone not connected to the Mission planner because it is transmitted via Wi-Fi and not via telemetry.

Fig. 4.5. Video streaming page, rpanion. This is the website of the company rpanion in which we can configure the video transmission, changing the image from black and white to colour, number of pixels or number of screens per second. http://10.0.2.100:3000/video

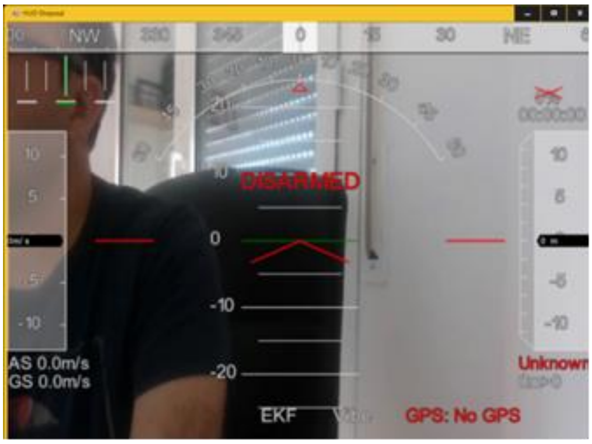

Fig. 4.6 Intel streaming camera image copied from Mission Planner. Image captured from the laptop, shows the image superimposed on the drone's PFD.

Finally, if the VLC application is downloaded in the laptop and the streaming video address is copied, the image and video can be recorded and saved during the flight, do not forget to save it as an Html file, otherwise, it will not be possible to view the file.

4.3.2 Step Two, location of the camera

The best place to install the camera is at the bottom, the main problem is to keep the USB cable as far as possible from the propellers. On the other hand, if the camera is placed on the top of the drone the rotation of the propellers hinders the images, that’s why at end the best location for the camera was underneath the drone's frame.

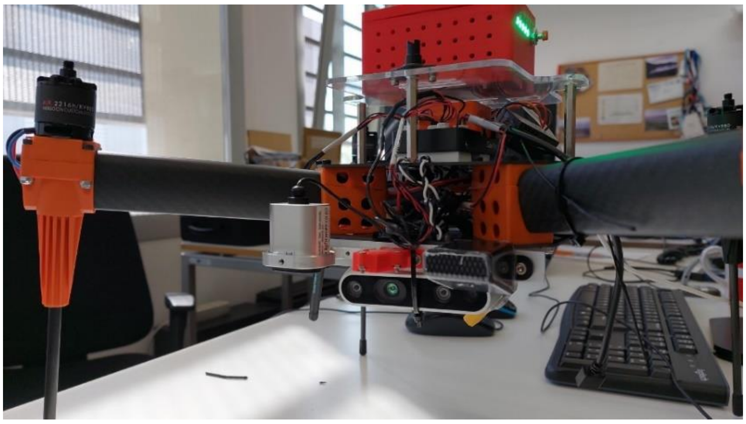

Fig. 4.7 Final camera setup.

4.4 Wi-Fi data transmission architecture

As a summary, attached is the schematic of the connection structure to obtain a video image through a created Wi-Fi hotspot and how to link it to our Mission planner.

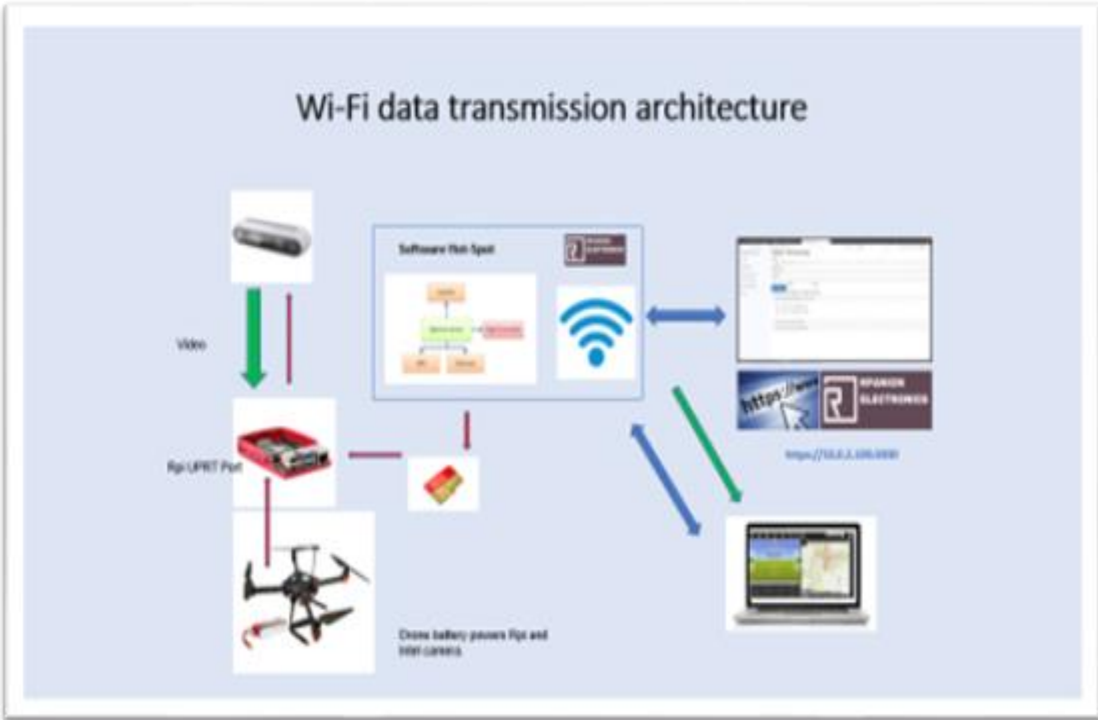

Fig. 4.8 Wi-Fi architecture @ Francesc Capdevila. In this image we can see how the different systems interact, on the right side of the image we have the base station (laptop) which via Wi-Fi is connected to the hot spot of the drone and in turn connected to the rpanion website , where we can change the parameters. In the center of the image we have the SD card, on it we find the rpanion software

that creates the Wi-fi point and the scripts created for that purpose. The SD is located inside the Raspberry (left image) and connected to the camera and the drone's battery (left bottom).

4.5 Wi-Fi Test quality of Wi-Fi video versus distance and quality of image

Once the installation is finished, a series of flights were made to compare the quality of the image by changing the different parameters of the video streaming.

The changed parameters were:

- Resolution (pixels). In pixel resolution, the term resolution refers to the total number of count of pixels in a digital image. For example, if an image has M rows and N columns, then its resolution can be defined as M x N.

If we define resolution as the total number of pixels, then pixel resolution can be defined with set of two numbers. The first number the width of the picture, or the pixels across columns, and the second number is height of the picture, or the pixels across its width.

- Bitrate (Kbps). A unit of measurement used on the Internet or other devices to measure the speed of information transfer over a network or telecommunication line.
- Framerate (fps). It is the measurement unit that shows the frequency at which a player or image logger generates the different frames, in this case, images per second.

Configuration chosen for the test:

1920x1080p - 30 fps FHD full high definition	5000 Kbps
848x480p - 30 fps SD standard definition	2000 Kbps
320x180p-15 fps	1000 Kbps
Distance	30 meters
High	1.5 meters

Fig. 4.9 Video test parameters. Three configurations were chosen up to a maximum distance of 30 meters, this distance was chosen because in the initial tests it was found that from 30 meters the signal was interrupted if, for example, an object was placed between the drone and the base station.

4.6 Flight test results

First test.
The first test was carried out outside the University with a direct LoS (Line of Sight) but with some buildings around. Typical parameters of an FHD video transmission were used. The result was quite significant; the video image was frozen from the beginning and video transmission was not possible. The first image obtained was frozen. Parameters were changed reducing the Kbps to 3000 and the fps from 60 to 15, but still, the image did not move.
In a second test with the same parameters and changing the location of my laptop to higher position we can see how the flight parameters of the drone change, but the image remains frozen, the latency is very high and there are hardly any image changes (Figure 4.10)

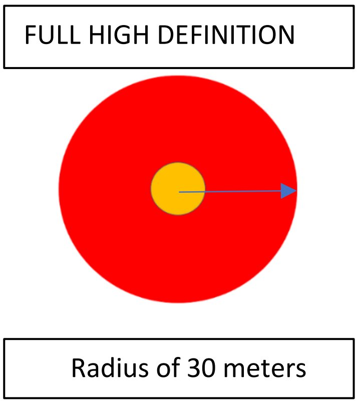

Fig. 4.10 In this video, we can see the result of the first test where the signal is still very weak with quite significant latency but at least the PFD of the drone is providing information and time to time we had some image of the flight.https://youtu.be/GMVKkwDr0Z0

Second test
The second test was more productive, and two videos were done. First, we reduced the resolution and although with some latency the image was kept until reaching about 30m, where the signal was lost not from the video, but from the Hot Spot. You can view the first video of the second test (fig 4.11). Doing a second video with the same parameters but keeping a higher position had better results, at least the image was kept all the time but still with some latency especially when reaching the 30m boundary. (Fig 4.12).

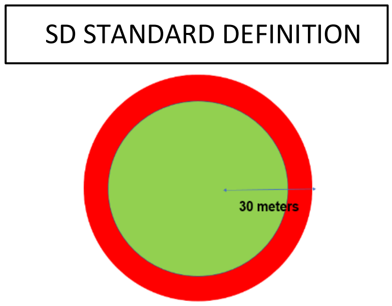

Fig. 4.11 In the first video test we can see that the image is quite good from the beginning (green dot) and when we reach about 30 meters we lose the signal (red line) https://youtu.be/dej59Mi_Jzo

Fig. 4.12 In the second video test we can see that the quality of the image is much better if we raise the GCS to 1.5 m above ground. https://youtu.be/14-Wj6yye68

Third test.
Finally, the third and the last video test was done with the lowest resolution, only 15fps, and 1000Kbps. The result is that the video has little latency but the image cuts intermittently and although it does not freeze, the image appears grey (Fig 4.13).

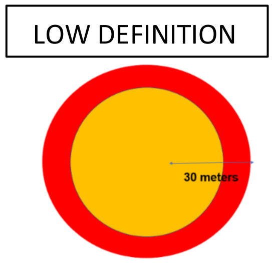

Fig. 4.13 The video transmission in this last test is quite irregular during the first 25 meters (orange dot), when reaching the established limit of 30 meters the signal is lost. https://youtu.be/z0oyGfMUYeg

The best image is obtained with the standard configuration.

CHAPTER 5. REFLECTOMETRY TEST FLIGHT

5.1 Introduction

The software for this test has been developed by CTTC for an earlier project called IOPES[2], and now it has been used to perform this test. This software takes the position of the drone and sends it to a file on the microSD, called *GNSS Parser/results*. To carry out the test, the system must be checked with the drone on the ground. After analyzing the SD files detect that the *results* file is empty, which means that is not recording any data therefore it does not detect any signal. After investigating the source of the problem, whether it was the antenna, the software or the receiver, we discovered that there were several filters previously installed that were not allowing the data to be recorded. CTTC Ph.D. Pedro Espín cleaned these filters and finally, the receiver worked and sent data to the SD card.

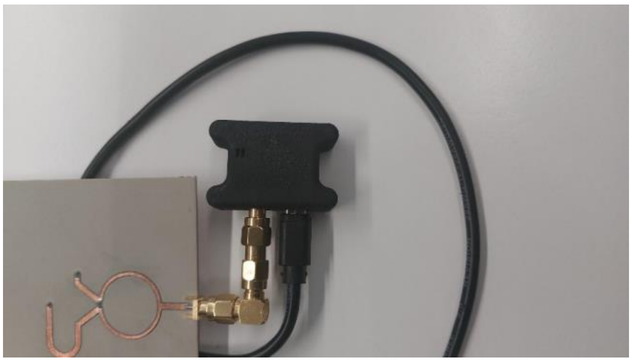

Fig. 5.1 GNSS receiver U-blox

5.2 Test development

The test was carried out in the drone lab located at the EETAC University of Castelldefels and the same Hexsoon 450 drone was used. A series of 3 flights were carried out that produced a series of data that were stored in the same SD.

5.2.1 Preparation of the drone

The test was performed at the dronelab of the UPC, the steps to follow were:

First: Preparation of the drone to record what we need, we had to connect the drone to the laptop, go to configuration, and then to Standard Params, from there choose the data that interests us, GPS, IMU, etc.

Second: Compass calibration of the drone.

Third: Locate the drone inside the dronelab to record the home position point, this point will be the home location in case to use RTL mode.

Fourth: We arm and disarm it, leaving a few minutes so that it receives enough satellites at the start point so that the drone takes it as a GPS point, position and altitude (Home), we can also press the "Home location" tab in mission planner, and it will also record the position as an exit point.

[2] https://iopes-project.eu/

5.2.2 Preparation of the flight

We create the points of the flight, choosing to point 1 as *take off* and the last point as *Land*, the flight has a total of 7 points at a maximum relative altitude of 10 meters. The points need to be written (Write WPS tab) and save it for a later analysis.

To analyze the data, we can download the data by connecting the drone with the USB cable to our laptop by pressing the tab from the mission planner "*download data flashlog via Mavlink"*. If we want to analyze them from the drone without downloading them, we can go to "*Autoanalysis".*

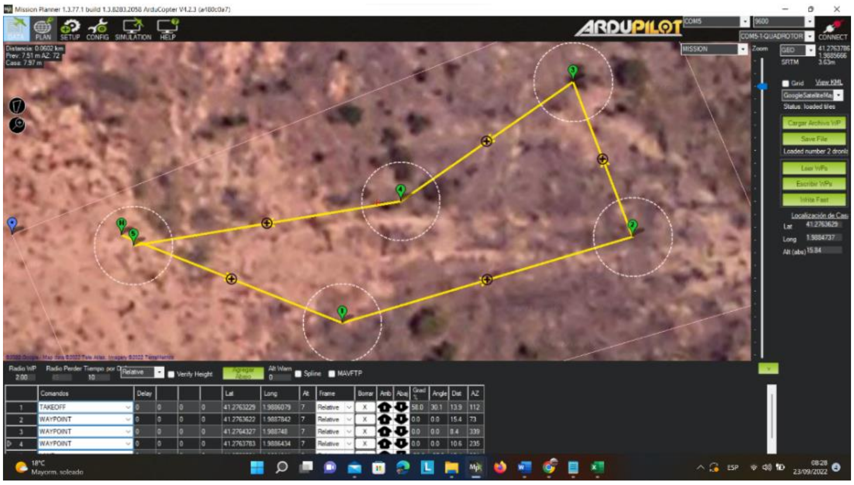

Fig. 5.2 First reflectometry test flight, six points at 7 m of altitude.

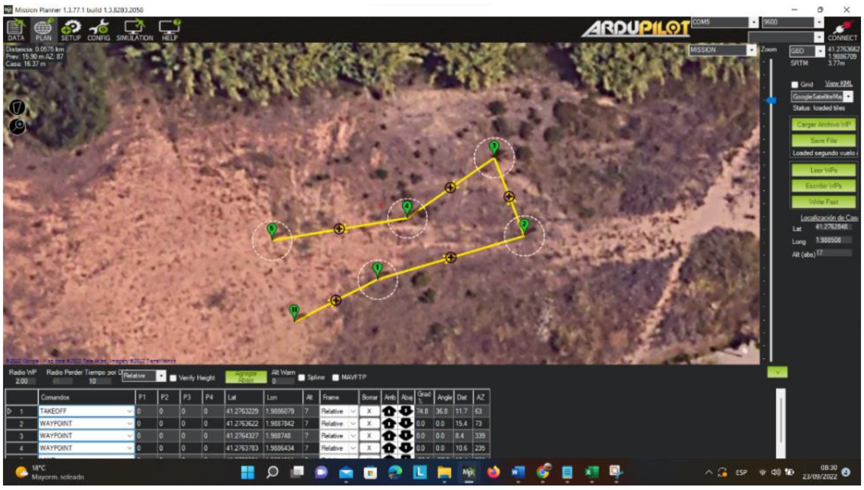

Fig. 5.3 Second reflectometry test flight, 6 points at 7 m of altitude, watch the attached the video of the flight
https://www.youtube.com/watch?v=qa7BQ6EwBwU

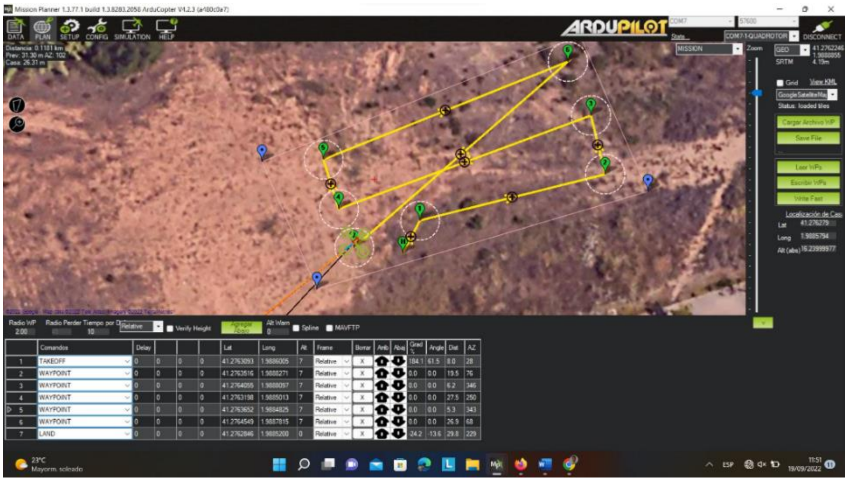

Fig. 5.4 Third reflectometry test flight. 8 points at 7m of altitude, watch the attached the video of the flight, https://www.youtube.com/watch?v=jBpEO3s31xk

5.3 Test Results

After the three flights a total of 3743 points were obtained until landing and disconnection, these data were obtained from the GPS antenna of the drone, of these points, we were interested in the position of Lat and Long and especially the altitude, the other data were disregarded. On the other hand, from the GPS antenna for the reflectometry, a total of 49781 points were obtained, this difference was obtained because this antenna was not disconnected during the 3 flights since it was an independent system to the drone, the first point would be the initial point of the first flight and the last (49781) the final point of the third flight. This forced us to make a manual selection of the reflectometry data later in order to make the comparison. The initial point of the flight started at 11.32 local time however until 11:38 there were not enough satellites connected, therefore the minutes between 11:32 and 11:38 were disregarded. We took 11:38 local time as the first sample minute, and 12:16 as the last one, a total of 44 minutes for the 3-flight preparation and execution. From the GNSS antenna of the drone, we obtained an average of three to four data per second, while from the GNSS reflectometry antenna we obtained about 20 data per second. [3]In order to make a more reliable comparison, it was decided to take the last five minutes of sample from the two antennas, from 12:11:26 local time and reduce it to 400 points, leaving only one sample point per second.

[3] *(Indeed, the amount of GNSS received signals can go from about 15 if measuring from the Earth's surface to even more when working with an air-based system (the higher the receiver's altitude, the higher the number of signals). GNSS-R systems for soil parameters determination. UPN Rodrigo Moreno Cordón Luis Javier Serrano Arriezu Patrizia Savi Pamplona, Julio 2018*

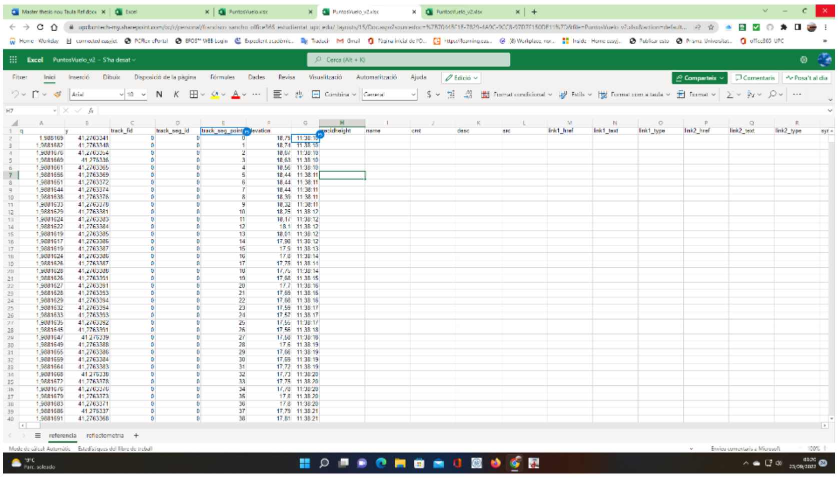

Fig. 5.5 Drone GNSS antenna data, example of data obtained from file log.

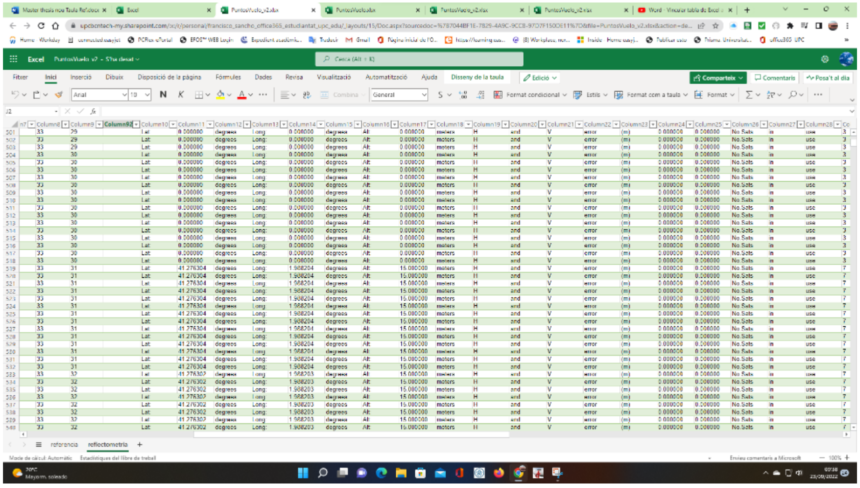

Fig. 5.6 Reflectometry GNSS antenna data, example of data obtained from file log.

Another way to check the quality of the data is by comparing the graph with the number of connected satellites (Fig. 5.7). In this graph we can see that the first minutes of the sample are not very reliable because the number of connected satellites increases gradually from 0 to the maximum number of 12. As can be seen, the altitude data of the first 4000 seconds are out of the graphic dynamics, from that point on, the receiver has 12 satellites captured at all times.

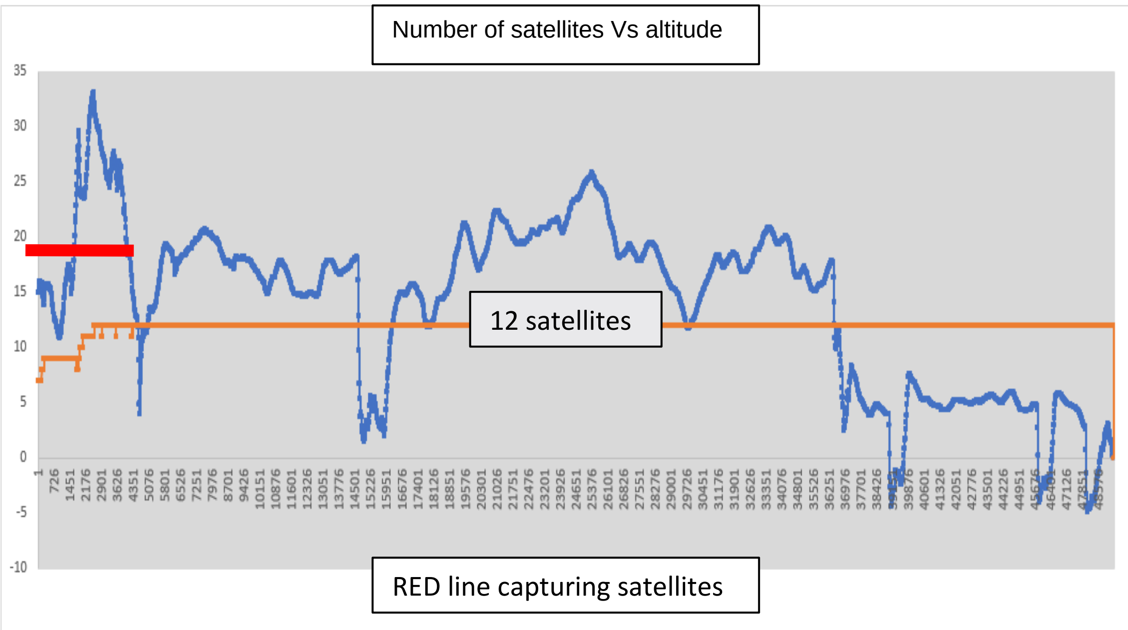

Fig. 5.7 Number of satellites captured per second during the entire sample. In this graph we indicate in red the first minutes of the sample, the orange line is the number of satellites connected.

The Fig 5. 8 shows the altitude points obtained by the GNSS antenna of the drone. The highlight is the number of sample points compared to those of the reflectometry antenna (3500 Vs 47000), we can also observe the three flights and the time when the drone was on the ground (marked in red). We can observe in the last flight (from 2500 to 3500 points) an orange line that separates the last five minutes of data obtained, It was decided to choose the last five minutes of the flight because the endpoint of the data is common in both databases, since the drone stopped suddenly due to an electrical problem. The antennas were transmitting all the time.

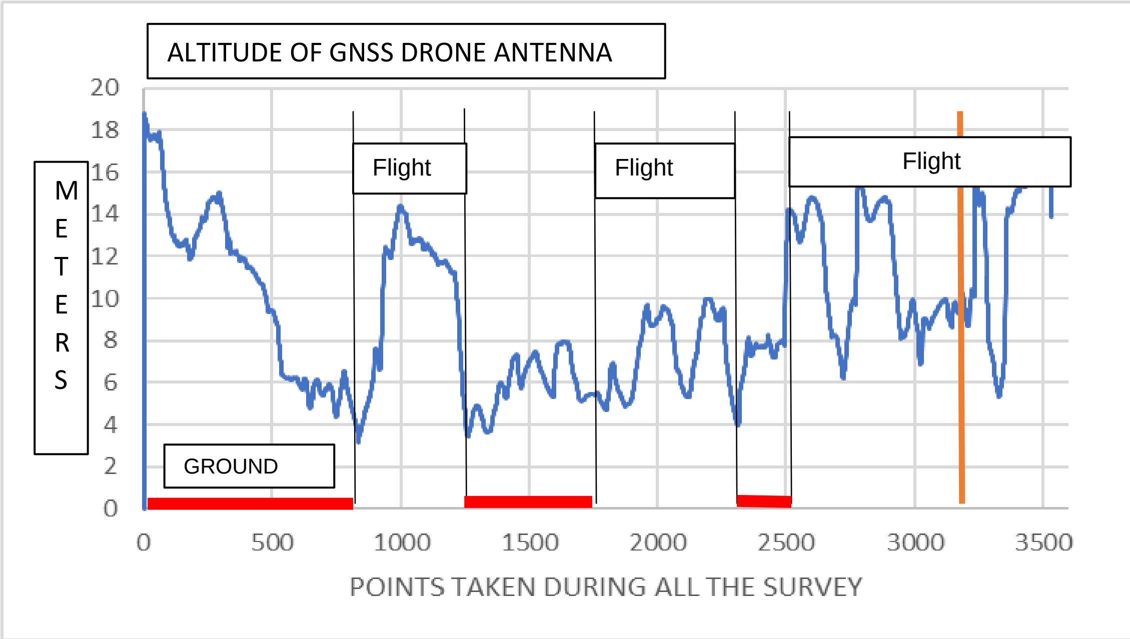

Fig. 5.8 Altitude of GNSS drone antenna. Red lines show when the drone was on the ground. The orange line shows the start of the last 5 minutes flight.

Fig 5.9 shows that the drone antenna was transmitting within the dronelab area, the opposite of the reflectometry antenna where it was transmitting in and out of the dronelab. Due to the characteristics of the drone, the reflectometry antenna could not be disconnected during the three flights, we can observe how the reflectometry antenna was transmitting from the beginning without interruption, even from the preparation table of the dronelab, this situation was an important factor in the production of a large amount of data.

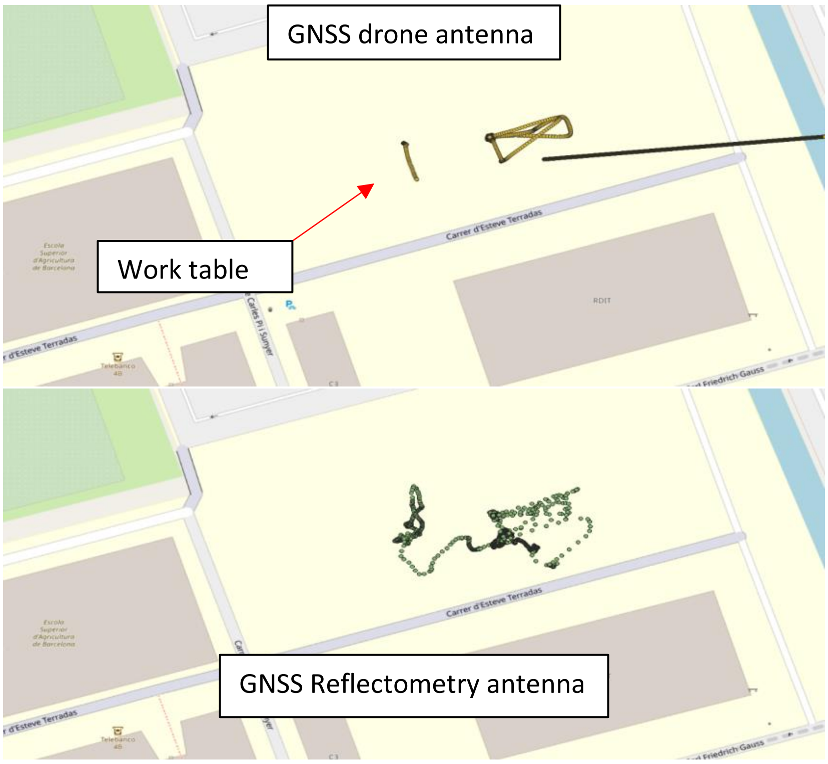

Fig 5.9. Drone and reflectometry antenna position during the last flight. The upper image reflects the points obtained by the drone's antenna in the third and also the last flight, in the lower image the points obtained by the reflectometry antenna during all flights.

In the Fig 5.10, we can see all the points obtained (more than 47000 points) and their relative altitude for the whole test. In the red zone, the drone was not flying, however, it downloaded a large amount of data, during that time the number of satellites was less than 8 and it was outside the flight zone. In Fig 5.9 it can be seen that it was on the dronelab's work table. The orange line shows the start of the last five minutes flight used for a more detailed analysis.

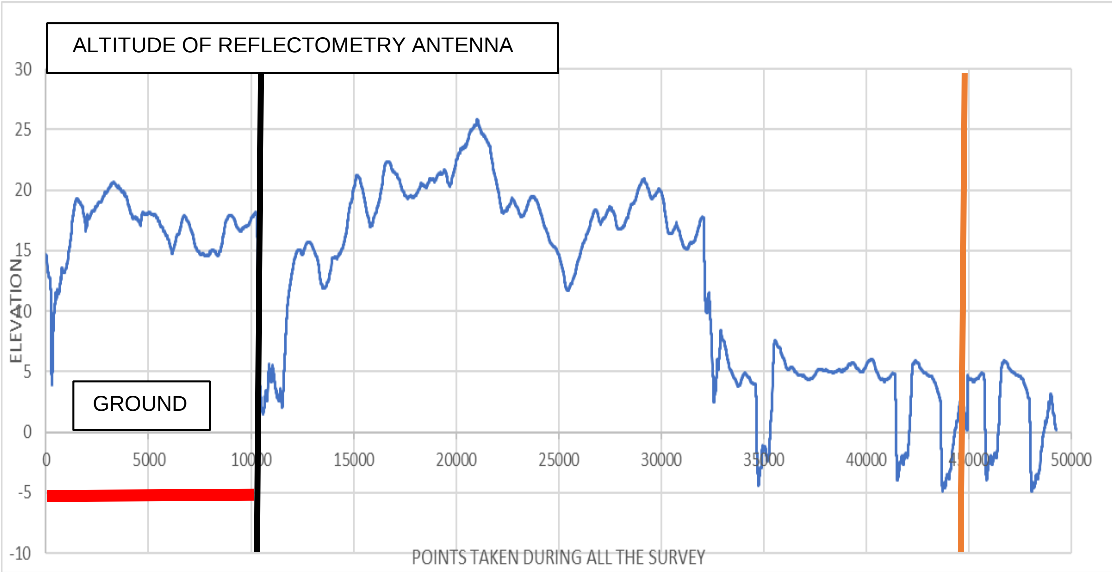

Fig. 5.10 Full flight height data from the reflectometry GNSS antenna. In red the data obtained that was discarded for being irrelevant, the orange line marks where the last 5 minutes of the last flight begun.

If we compare the graphic figures 5.8 and 5.10 we can see that the points do not coincide precisely because the amount of data obtained by the reflectometry antenna is much greater, 3700 vs 47000. Unfortunately due to the difference in tag time with the drone antenna and a large number of points it was not possible to determine the beginning and the final of each flight, only it was able to determine the beginning of the test where the antenna was active but out of the dronelab area and the final of the test where the power supply was cut off. The drone was connected to the battery during the entire series of flights and therefore the reflectometry antenna (which is powered by the battery) was downloading data all the time, this detail meant that in the end, we had about 47,000 position data from one antenna compared to 3500 from the drone's GNSS antenna, this gap in the amount of data caused a gap in the tag time as well, in some cases reaching a difference of more than one minute between the two.

Figure 5.11 is the most interesting and relevant graph, we can verify that the data obtained from both the drone antenna and the GNSS reflectometry receiver antenna are practically the same but inverted with respect to ground altitude, this result shows that the data is valid, since one of the effects that occur when making an observation with reflectometry is that the vector that reflects the signal appears in the database as the inverse point (Fig. 5.12 point R') to the one obtained by the drone's GNSS antenna. Specular point and specular image The equipment is not able to recognize if below the drone there is another surface such as the ground, therefore the data it captures is the intersection of the prolongation of the height of the drone with the satellite signal, and appears on the graph as an inverted signal.

The basic principle in altimetric measurements is based on the difference in reception time between the direct and the reflected signal. The weakness of the reflected signal is affected mainly by the reflection surface, which degrades the intensity, but at the same time translates its characteristics.

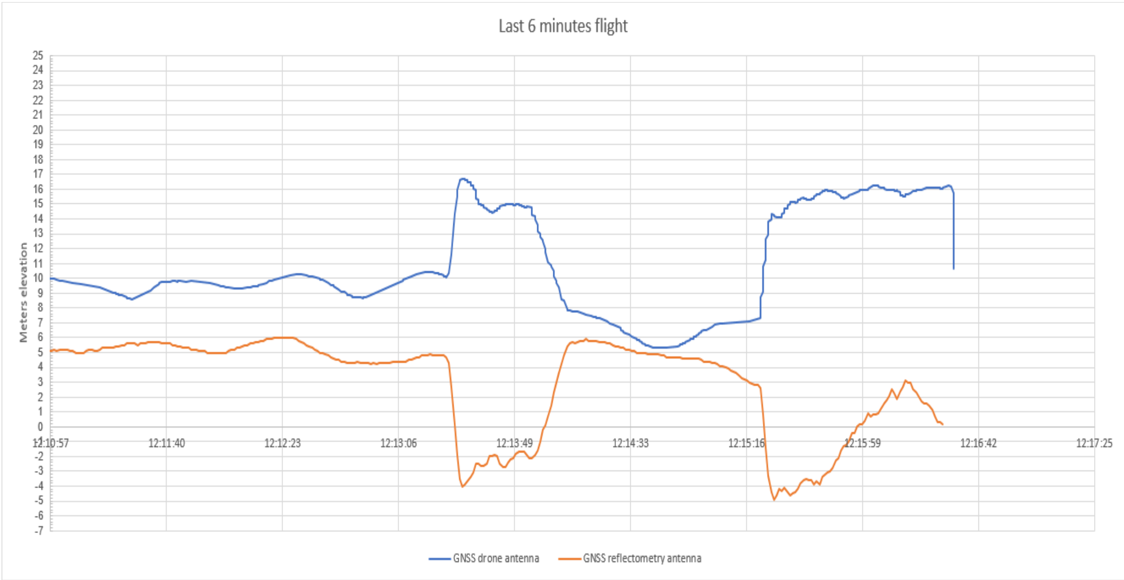

Fig. 5.11 Comparison between the two antennas during the last 5 minutes of the flight. The blue line shows the altitude points from the drone antenna, the orange line shows the altitude points obtained by the reflectometry antenna. The vertical axis is the altitude in meters and the horizontal axis the last five minutes of flight in seconds.

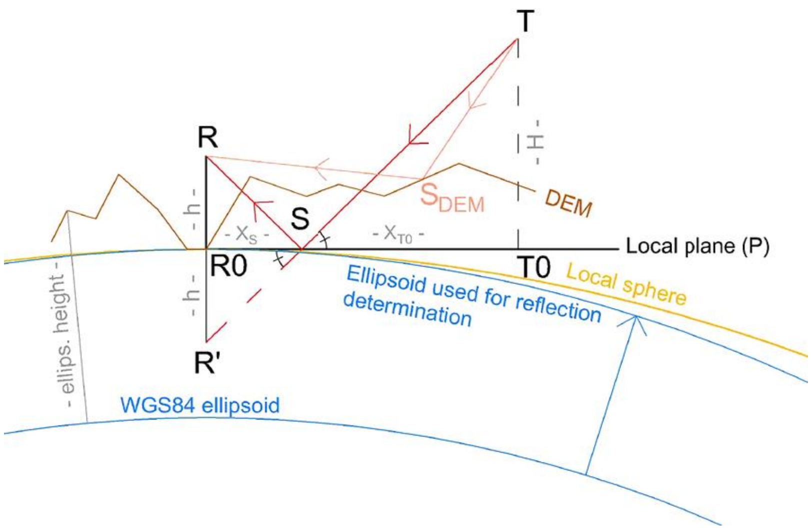

Fig. 5.12. Reflected signal. SpringerLink Theoreticcal fundamentals of GNSS reflectometry (2020). Determination of the specular reflection point in a local plane approximation. S: specular reflection point position. R: receiver position.

T: transmitter/satellite position. h: height of the receiver above the ground surface.

During the stationary flight, the data is quite reliable, except when the drone starts to fly, where there are abrupt variations in the signal. It must be considered that the drone flight was quite abrupt during take-offs due to the characteristics of the drone and that the GNSS receiver was not of high precision (0.5 m error), the original receiver of the drone is a low precision (2 or 3 m error). However, when the drone is stationary or loitered, the results are quite optimal.

Fig. 5.13 True altitude after comparing both data (green line). The orange line indicates the altitude from the drone antenna, the blue line indicates the altitude from the reflectometry antenna, and the green line shows the true altitude of the drone in the dronelab. The yellow dotted line shows the drone in stationary flight. The vertical axis is the altitude in meters and the horizontal axis is the last five minutes of flight in seconds.

6.CONCLUSIONS AND FUTURE WORK

6.1 CONCLUSIONS

The first part of the work carried out in this thesis is based on the assembly of some complementary systems, mainly the installation of an RPi and a video recording camera, the main problem found has been the latency when transmitting video over Wi-fi, this latency mainly depended on whether there was any object between the transmission and reception line, for example, just a person standing in front of the drone would already interrupt the signal, and the other factor observed was the height of the control station (laptop). For example, if the GCS was a meter and a half, half a meter or at ground level, the latency

and quality of the transmission changed a lot. It is clear that in this first test the distance and the presence of objects are the main reason for the latency, in addition, the Wi-Fi network was only used by the drone and anybody else. Therefore, latency caused by the use of the network by different users can be ruled out and the best configuration for a good video transmission was the SD (standard definition). The other problem found was the drone itself, when the thesis started the drone had not been used for a few months, and many parameters had to be configured again since when updating the firmware part of the previous parameters disappeared and this delayed the work quite a bit.
It was possible to connect the Rpi with the drone battery, and an external battery for the Rpi was not necessary. During Rpi installation the main problem was the fragility of the pins (the RPI had to be repaired when two pins were broken) and finally, when the pins were so close to each other at some point the positions were confused and the installation had to be reviewed again. In the end, everything worked perfectly and both the LED lights and the RPi ran on the drone's battery until the end.

Reflectometry; the first conclusions tell us that the data obtained is not very precise at the beginning if we compare the data obtained by the two antennas with each other, the main reason may be the low number of satellites captured, the three flights started from the same point and without disconnecting the drone from the battery, although the mission planner was disconnected to load the flight plans. To simplify the process, the last five minutes of the flight were taken and repeated values were manually removed taking almost two days to manually clean the last 5 minutes of the flight.
The reflectometry antenna acquired 20 or 19 data per second and 90% of the twenty data had the same value, after analyzing the data, it was observed that the reflectometry antenna had acquired negative altitude and also the initial altitude in the three flights varied almost 3 meters without changing position. On the first flight, the altitude was 3,47 meters which I think is the most accurate (elevation of Castelldefels is 3m according to google earth) considering that the antenna is more or less 40 cm above ground. On the second flight it changed to 4, and on the third flight to 5.32m. Another problem has been the synchronization, by using the same power source for all the equipment, the reflectometry antenna could not be disconnected if the battery was not unplugged, comparing the graphs, it is observed that at some points the two data obtained have a good correlation, especially in the last 5 minutes, which was where the data was filtered. We can also observe that when the drone is in stabilized flight or stopped the true elevation data also corresponds to the actual elevation (Fig 5.13)

6.2 FUTURE WORK

I sincerely believe that the use of the drone as a platform to collect reflectometry data works, the points to improve would be to obtain a more stable drone, perhaps a hexacopter would be better than the quadcopter to have more stability especially during the take-off, I think is very important to obtain a software that could work with both data sources at the same time and that could filter non-relevant data, a good correlation of data and time is very important, also an antenna with a better precision could give better results with higher precision and accuracy.CTTC has developed software that improves some of the problems, this software was applied to the above-mentioned IOPES project, and I believe that if this software were used the results would improve. Also, better results could be obtained if a more advanced drone could carry a higher payload to be able to the two GNSS systems with independent batteries, disconnection of data collection during inactivity periods would help to analyze later on all the data obtained.

REFERENCES

1 International Communication Union, ITU-R managing the radio frequency spectrum for world.https://www.itu.int/en/mediacentre/backgrounders/Pages/itu-r-managing-the-radio-frequency-spectrum-for-the-world.aspx (2022)

2 Intel Real Sense Center, Intel realsense camera and Rpi Tutorial to install librealsense, pyrealsense2 library and the other libs (2020)

https://support.intelrealsense.com/hc/en-us/community/posts/360048495493--Intel-Realsense-D435-with-RaspberryPi-Best-practice-installation-guide

3 MDPI, acronym of Multidisciplinary Digital Publishing Institute, is a publisher of open access scientific journals. Dron communications article (2022) https://www.mdpi.com/journal/drones/sections/drone_communications

4 Transversal Project EEATC, (2021-2022) Miguel Valero, teaching and research staff of the Castelldefels School of Telecommunication and Aerospace Engineering https://atenea.upc.edu/course/view.php?id=72009#section-9

5 IOPES. The IOPES is a two-year project co-founded by EC and ECHO. (2019-2021) https://iopes-project.eu/

6 Python (2022) The Python Tutorial. https://docs.python.org/2/tutorial/index.html

7 Hexsoon 450 (2021) https://ardupilot.org/copter/docs/reference-frames-hexsoon-edu450.html

8 Reflectometry (2022) https://en.wikipedia.org/wiki/GNSS_reflectometry

9 Mission planner (2001-2022) Mission Planner is a ground control station for ArduPilot. It provides configuration and flight assistance, as well as review of recorded flights. https://ardupilot.org/planner/

10 Book,Theoretical Fundamentals of GNSS Reflectometry (Springer). Author Kegen Yu (April 2021) https://link.springer.com/chapter/10.1007/978-981-16-0411-9_4

11 GNSS-R systems for soil parameters determination. Trabajo Fin de Grado, Rodrigo Moreno Cordón Luis Javier Serrano Arriezu Patrizia Savi Pamplona, Julio 2018. https://academica-e.unavarra.es/bitstream/handle/2454/31519/TFG%20GNSS-R%20for%20soil%20parameters%20determination.pdf?sequence=1&isAllowed=y

ANNEXES

ANNEX I CAMERA SCRIPT (SD) RPi

```
Import cv2

From vidgear.gears import NetGear

#tried with 0, 1 and finally number 2 worked well
Stream = cv2.VideoCapture(2)

" " "
ret, grame = cap.read ()
if ret :
        cv2, imwrite("image.jpg",  frame)
" " "
#cv2 . startWindowThread()
#cv2 . namedWindow("dron camera")

Options = {"flag" : 0,
                "copy" : False,
                "track" : False}
Server = NetGear (
            address=" 192.168.1.77" ,
            port = "5454" ,
            protocol = "tcp" ,
            pattern = 0 ,
            logging=True ,
            **options)
While True:
            Grabbed, frame = stream.read()
            If not grabbed :
                break
            #cv2.imshow("drone camera" , frame)
            server. send(frame)
stream.release()
server.close()
```

ANNEX II GROUND STATION SCRIPT (LAPTOP)

```
#import required libraries
```

```
from vidgear.gears import NetGear
import cv2

#define tweak flags
options = {"flag" : 0,
         "copy" : False,
         "track" : False}

#Define Netgear Client at fiven IP adrress and define parameters, .155 home IP
address.
client = NetGear(
   address = "192.168.1.77",
   port = "5454",
   protocol = "tcp",
   pattern = 0,
   receive_mode = True,
   logging = True,
   **options)

#loop over
while True:
   Frame  =  client.recv()
   if frame is None:
      break
   gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
   cv2.imshow("Output Frame" , frame)

  key = cv2.waitkey(1) & 0xFF
  If key == ord("q") :
      Break

Cv2.destroyAllWindows ()
Client.close()
```

ANNEX III LED's SCRIPT

```
from gpiozero import Button
from signal import pause
import board
import neopixel
import time
from gpiozero import LED
def lights() :
for i in range (0, 5) :
#Remember RED, GREEN; BLUE, and 255 max values, we can choose a
lower one. (0, 255, 0)
pixels [i] = (0, 255, 0)
ipixels = neopixel.NeoPixel (board. D18, 5)
lights ()
```