Preliminary Report And Design Document

Team: S4ndHunt3rs

February 26, 2023

Contents

1 Introduction

1.1. Purpose of the mission

The purpose of the mission is to collect data about air, pressure and temperature based on altitude and images of the land.The collected images will be used for mapping agricultural areas.

1.2. Team organisation and roles

S4ndHunt3rs (coming from Sand Hunters is based on our secondary mission, to collect and analyse data, following the desertification rate, though for this launch, we will not looking just for deserted areas in our images) is a project of 5 students, 4 following the bachelor degree in Military Technical Academy "Ferdinand I" and one following the master degree at Politehnica University of Bucharest.

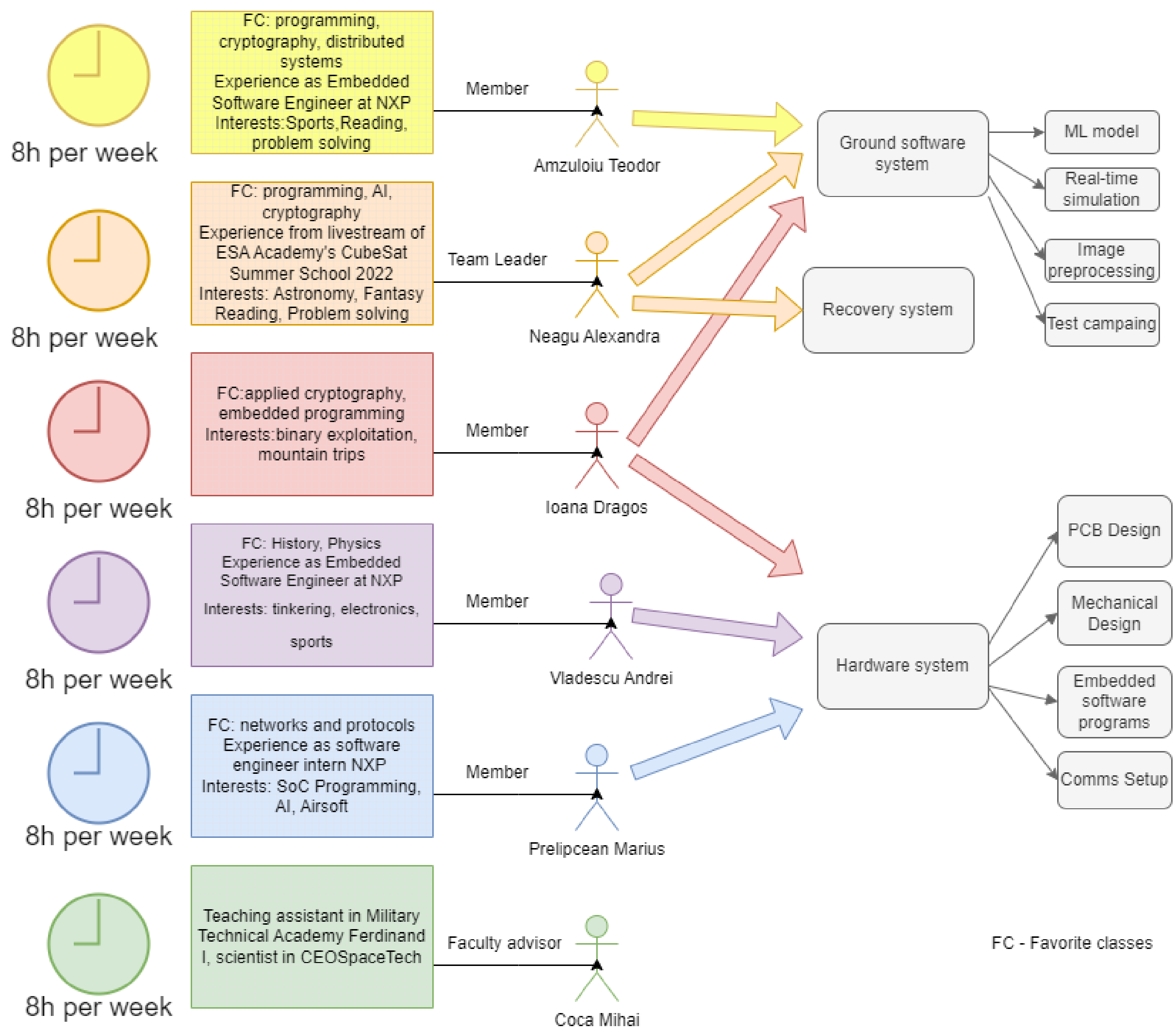

Figure 1: Team members

1.3. Mission objectives

CanSat's primary mission is to measure the pressure, the air and the temperature of the atmosphere based on altitude during the descent and to send those measurements to a radio ground station every few seconds. The data will be used to calculate and create a real-time simulation of the CanSat's drop. The location of CanSat after landing will be provided by the GPS, so we can locate it easier.
The second mission will capture land images from a top down perspective and save them in the satellite memory. This images will be processed and sent to a machine learning model. The model will be pre-trained with similar data in order to compute the outline and the percent of the agricultural areas. The collected images can be used in various ways based on the launch area. For example:

⇒ Buildings areas.
⇒ Desertification areas.
⇒ Agricultural areas.
⇒ Deforestation areas.

The purpose of our secondary mission is to demonstrate the possibility of collecting images and processing in more than one way.Based on the launch area we can choose the image processing mechanism. For now, we will choose the detection of agricultural outline. In the future this can be expanded to detect all the region from one or more images and compute a percent for each. Our satellite is a cheaper alternative to keep track in time of a chosen area.

To ensure the success of the mission, the following goals must be met:

- Recovery system: the parachute must open and ensure a safe CanSat landing.
- Power supply system: must be safe, the energy must be sufficient for the assumed number of working hours.
- Communication system: must ensure one-way communication from satellite to ground satellite during the flight and after landing
- Electronic system and mechanical structure must work: during flight, during descent and survive landing.
- Software: must correctly implement the assumed algorithm, show in a correct way the simulation of the CanSat.
- Images: it must capture a set of images each landing in a good quality that can be processed after that.

2 CanSat description

2.1. Mission overview

The mission will be carried out by designing and building a CanSat that will be launched and deployed from a rocket at an altitude of about 1000 metres. The CanSat must descend in the interval of 8 to 11 metres/second. Once launched, the CanSat will measure the air, the pressure and the temperature in an subsequent interval and send the results to the ground station. The data will be displayed in graphics and a simulation of the landing.

During descent, the CanSat will capture images of the land, save them on the memory and later compute the results of the set using the machine learning algorithms.

CanSat will descend with a parachute that will expand as soon as it is dropped out of the rocket.

The satellite will be equipped with sensors to measure the parameters of the atmosphere and with a camera for taking pictures. The on-board computer will be responsible for acquisition of data form sensors, capturing the images, save the data on the SD card and control the radio communication system.

The radio communication system is designed to send data to the ground station, based on an interval which will be calculated later.

The GPS module is connected to the on-board computer and will provide information about the current CanSat position. This signals will be transmitted during launch and after landing for more hours to make it easier to find the CanSat on the ground.

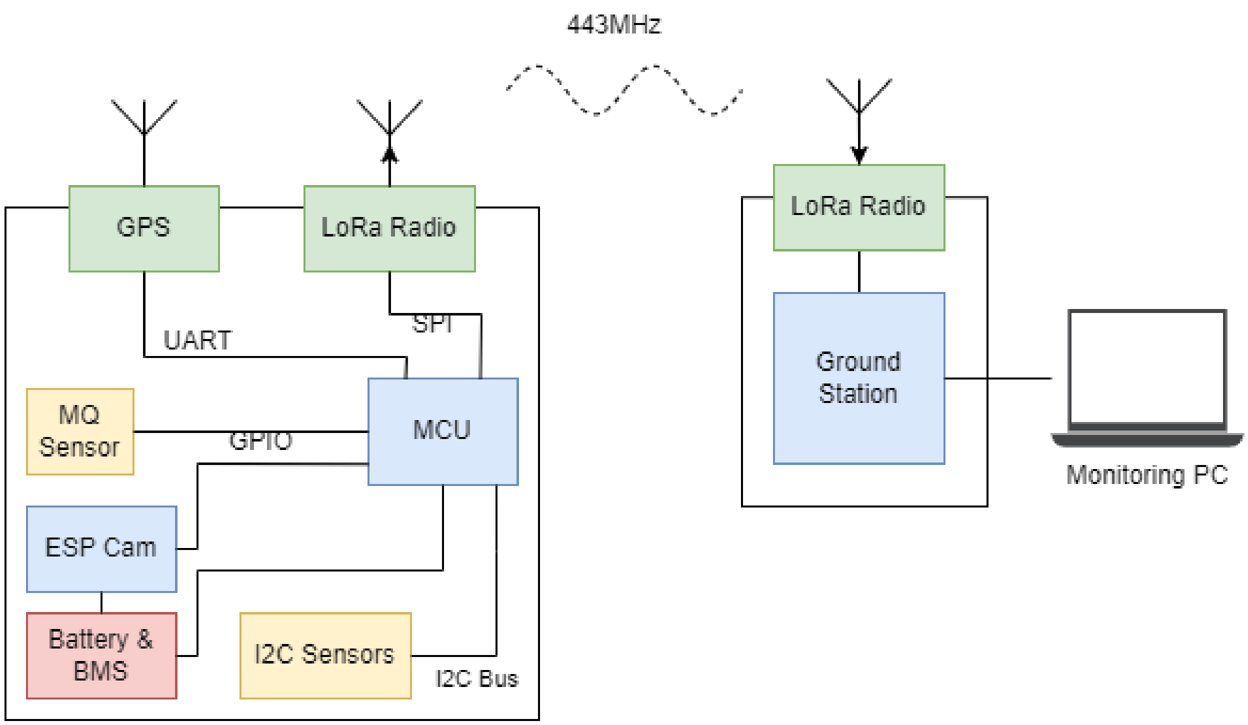

Figure 2: Mission design

2.2. Mechanical / structural design

The mechanical structure has been designed in Fusion 360, and is mainly comprised of 3D printed material, PETG, so that it can withstand greater mechanical shocks without breaking or deforming. The supporting system is achieved with the help of long screws, bolts and nuts. The PCB alignment differs between 2 design ideas, horizontal stacking and vertical - like an "I" beam. The PCBs also serve as a resistance factor, as they are made out of glass-fiber.

The 2 models have in common a design feature - the ESP CAM must be on the bottom, so that the camera can be oriented in a way that it can capture the ground beneath.

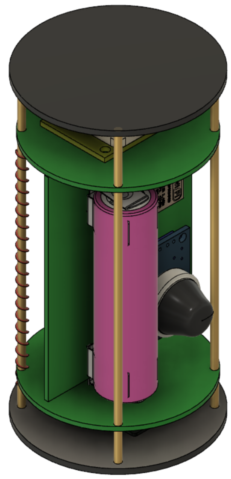

Figure 3: Vertical Alignment

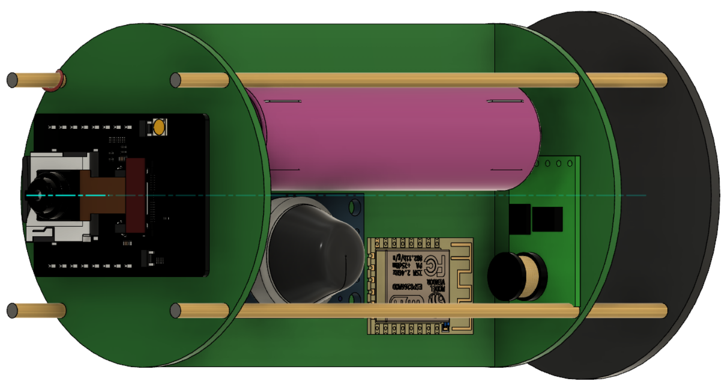

Figure 4: Vertical Alignment Close View

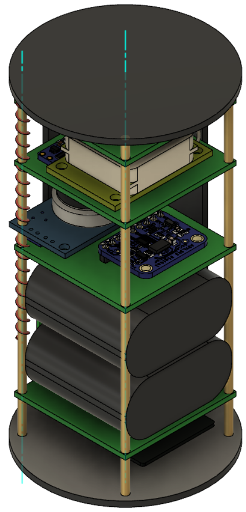

Figure 5: Horizontal Alignment

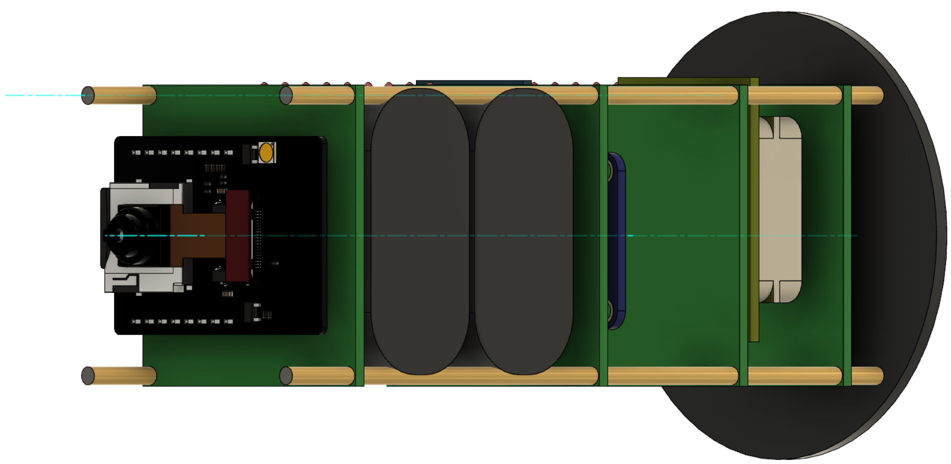

Figure 6: Horizontal Alignment Close View

2.3. Electrical design

2.3.1. General architecture

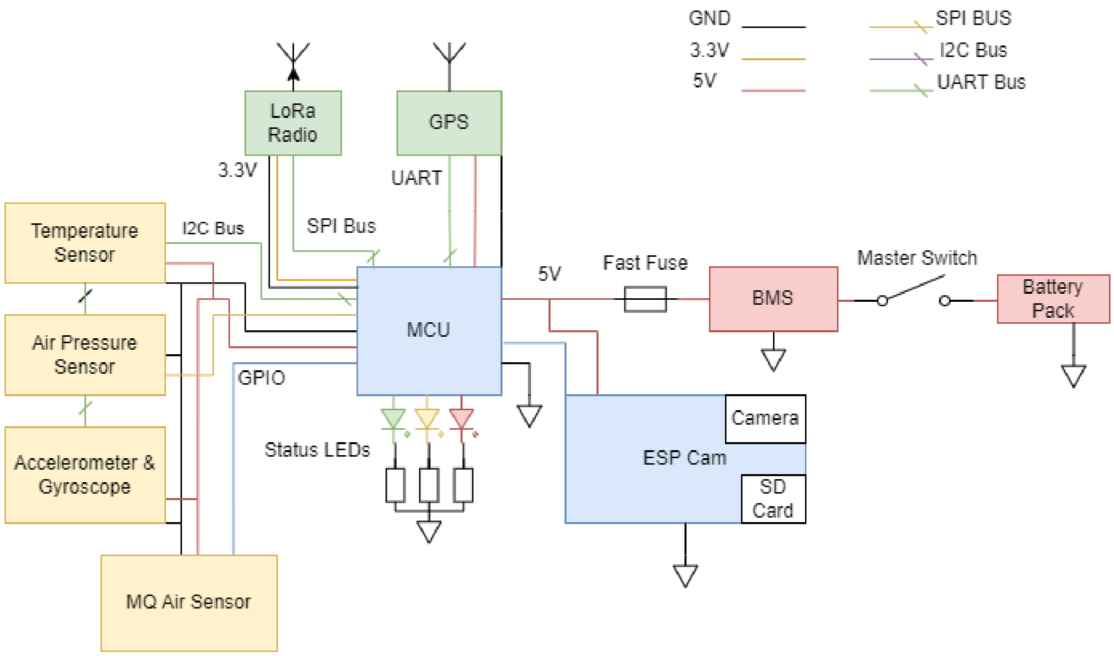

Figure 7: General electrical scheme

2.3.2. Primary mission devices

- Temperature sensors: AHT10
  - Measures temperature and transmits it into a digital format with I2C;
  - Temperature range is between $-40°$ and $85°$, with a resolution of $0.01°$ and $\pm$ 0.3 accuracy;
  - It can also measure humidity, as a bonus;
  - We are using this sensor since it is cheap, has good accuracy, low drift over time and has an I2C interface;
  - Price of 1.19 €.
- Atmospheric pressure sensor: BMP280
  - Measures the pressure of the atmosphere and transmits it into a digital format with I2C;
  - It can measure between 300 and 1100 hPa, with a resolution of 0.01 hPa and $\pm$ hPa accuracy;
  - It can also measure temperature, and be a backup for the dedicated temperature sensor;
  - Price of 0.35 €.
- Air quality sensor: MQ135
  - They are used in air quality control equipment and are suitable for detecting NH3,NOx, alcohol, Benzene, smoke and CO2;
  - Has a digital and analog output;
  - Price of 1.03 €.

- LoRa radio module: RFM95W
  - +5 to +20 dBm up to 100 mW Power Output Capability;
  - Can be used with 433MHz or 868MHz band;
  - Has an SPI bus;
  - Price of 1.68 €.

Figure 8: AHT10 Sensor

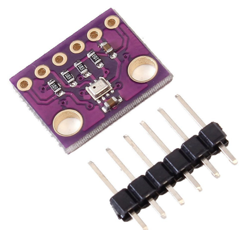

Figure 9: BMP20 Sensor

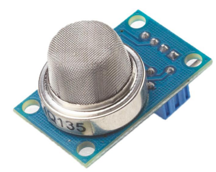

Figure 10: MQ135 sensor

Figure 11: RFM95W module

2.3.3. Secondary mission devices

- 3-axis Accelerometer and Gyroscope: MPU6050
  - The accelerometer and gyroscope data is transmitted through I2C;
  - It also doubles down as a low accuracy temperature sensor;
  - Price of 1.68 €.
- GPS module: GY-NEO6MV2
  - Transmits NMEA data through UART;
  - Can be used to determine the position and speed of the satellite;
  - 2.5 meters precision;
  - Price of 2.69 €.
- Camera module: ESP-32CAM
  - Records the camera (OV2640) feed to the SD card;
  - Built-in SD card holder;
  - 1600x1200 resolution 15FPS maximum;
  - Low power consumption of 125mW;
  - Price of 6.35 €.

Figure 12: MPU6050 accelerometer and gyroscope

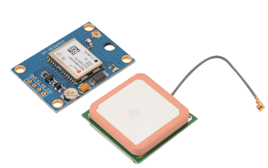

Figure 13: GY-NEO6MV2 module

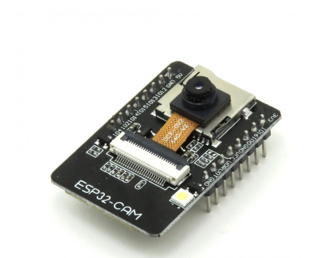

Figure 14: ESP-32CAM module

2.3.4. Electrical scheme

The electrical scheme is presented in the next figure.

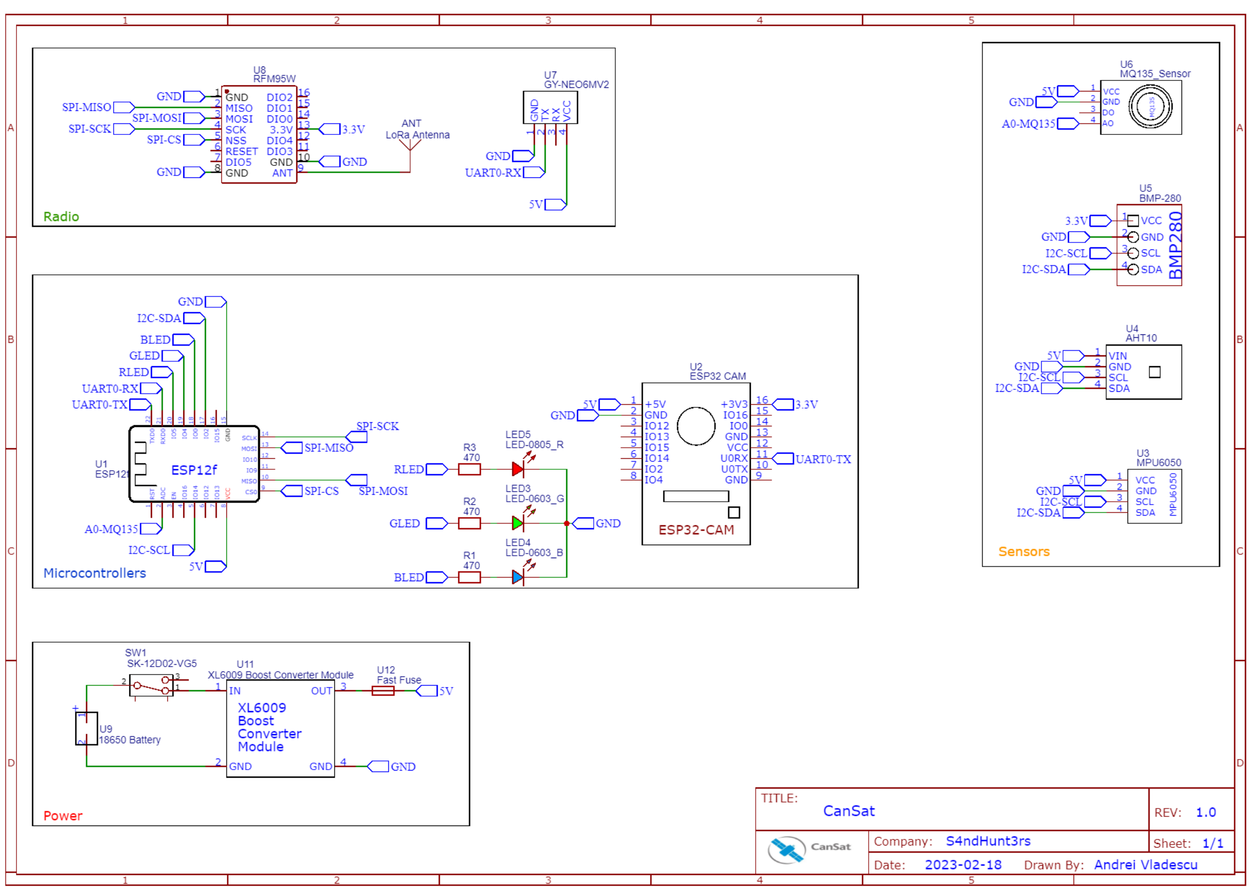

Figure 15: Schematic design

There are 4 subsystems:

- Microcontrollers
- Battery Management System
- Sensors
- Radio

The main microcontroller is an ESP from Espressif, but it is not decided wether to use the ESP-12F or the more powerfull, dual-core ESP-32. Performance can be gained by using the ESP-32 with multitasking on the 2 tensilica cores. Also, we leave the possibility of Over-The-Air updates for these microcontrollers, without needing any debugging cable attached to the CanSat.Inside the microcontroller block there is an RGB LED, for debugging purposes.

The Battery Management System is comprised of the battery pack, control circuitry and the boost converter, to boost the voltage to 5V TTL. A glass fast fuse has been introduced for current limiting. 3.3V are achieved by using the LDO that the microcontroller's boards' contain.

2.3.5. Communication system

We are using the LoRa protocol, with the help of the RFM95W module. Our architecture is a simple point-to-point network between the ground and satellite. The communication will take place over the 868 MHz band.

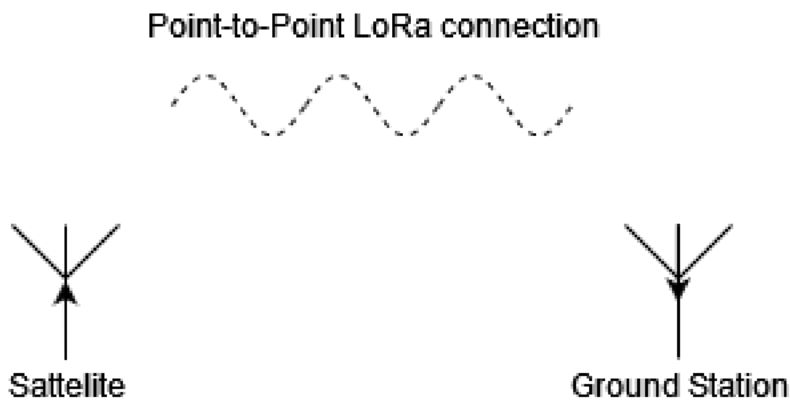

Figure 16: LoRa Communication

The communication between our CanSat and the ground station will be aided by a hand-built Yagi-Uda antenna since it can improve the communication process by increasing the signal strength and range, reducing interference, and improving the reliability of our communication system.

In other words, the set-up for communication will mainly consist of:

- 2x RFM95W modules (one for the CanSat and one for the ground station)
- 1x Yagi-Uda antenna (used by the ground station)
- 1x Wire antenna (used by the CanSat)

Building the Yagi-Uda antenna:

- We have agreed upon a 3-element Yagi-Uda antenna design which should fulfill our need for a reliable communication between our satellite and ground station.

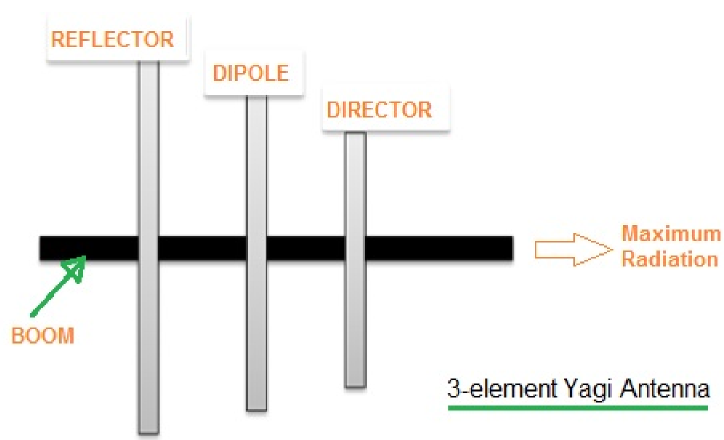

Figure 17: 3 element Yagi-Uda antenna

- First, we started by calculating the manufacturing parameters of the antenna using this 3-element Yagi-Uda Antenna Calculator and this antenna modeler and optimizer. The final measurements are these:

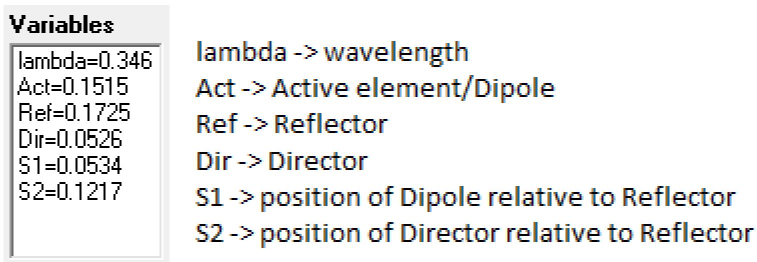

Figure 18: Measurements (in meters)

- Then, we started simulating the behavior of our antenna:

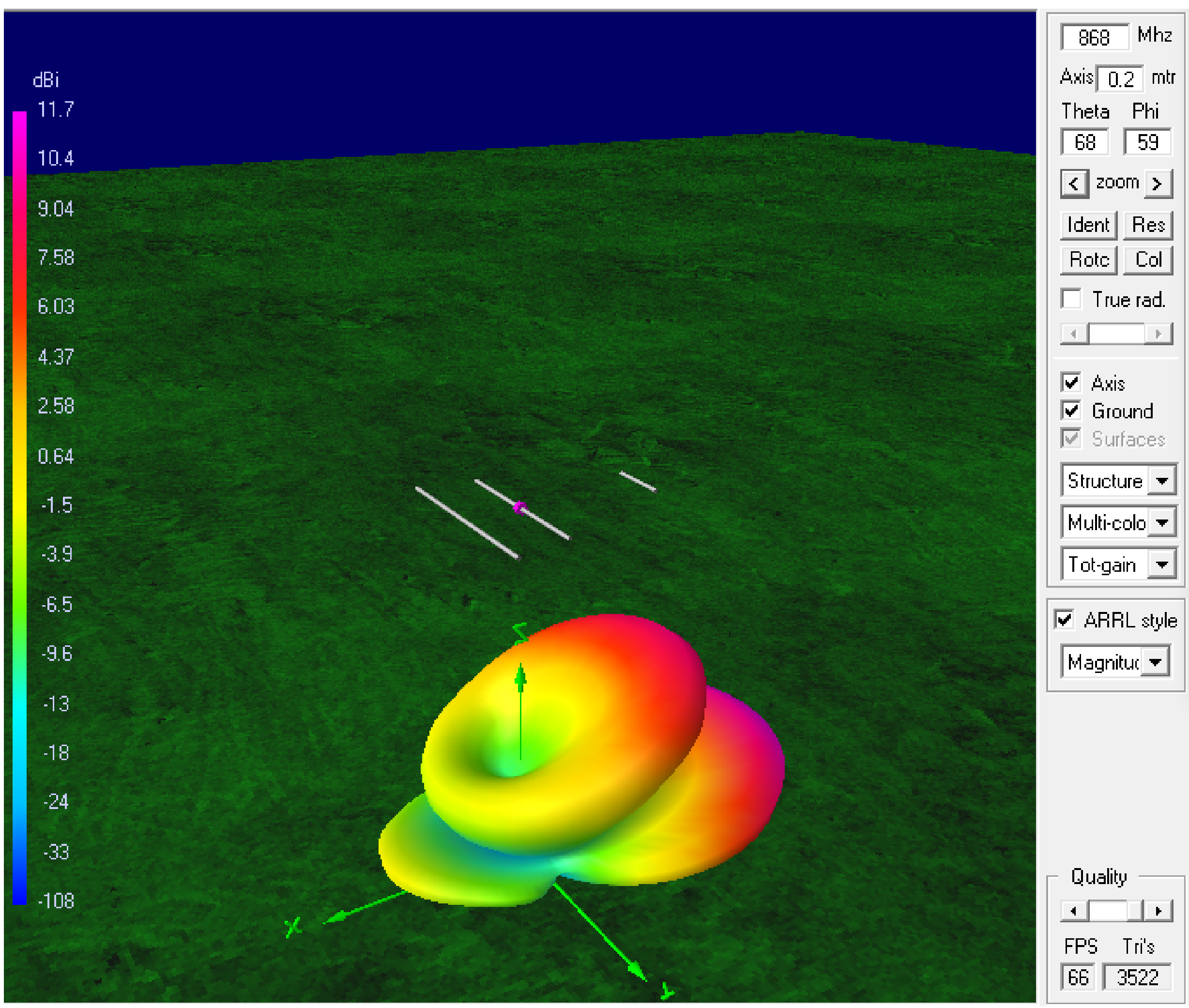

Figure 19: Simulation 1

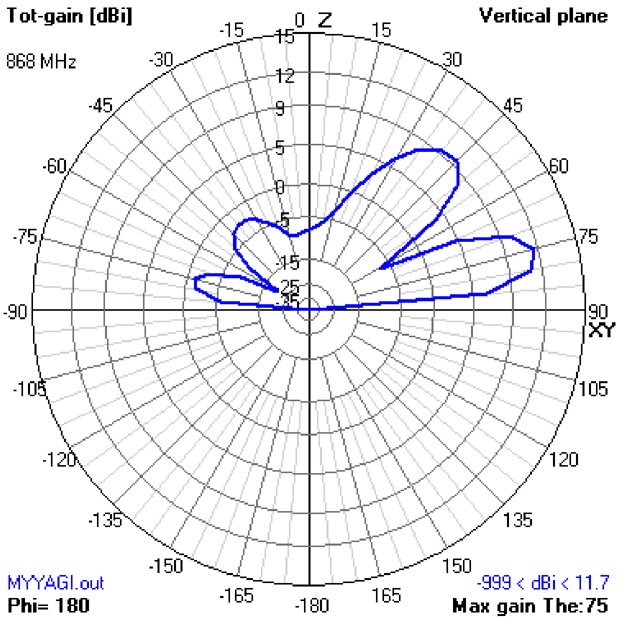

Figure 20: Simulation 2

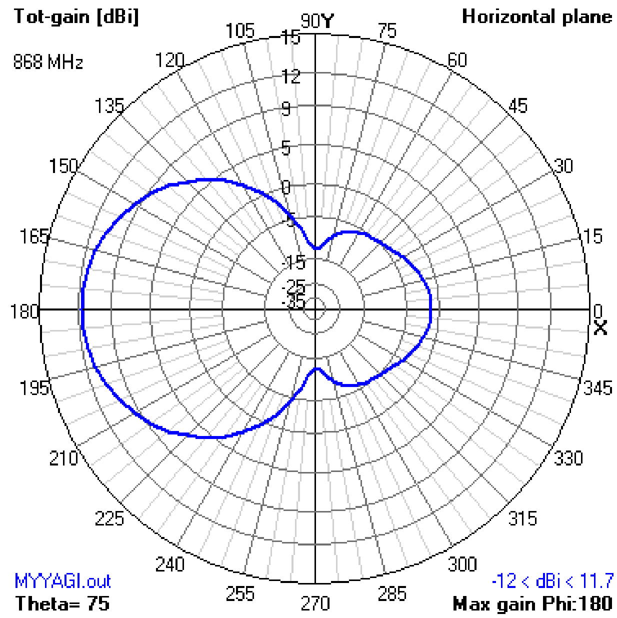

Figure 21: Simulation 3

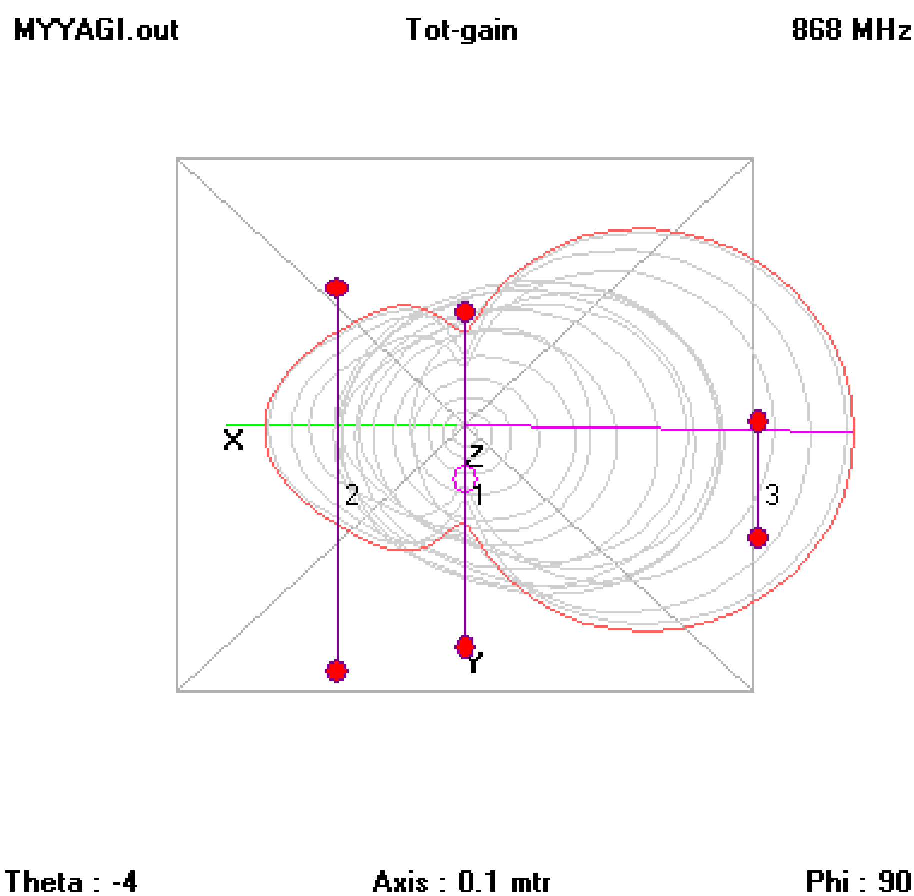

Figure 22: Simulation 4

- We are aiming for an SWR (standing wave ratio) value in the range of 1-2, which we managed to get:

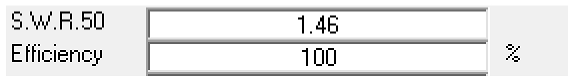

Figure 23: SWR value

- At this point we can start building our Yagi-Uda antenna. The following materials are required for the build:
  - 1x PVC tube;
  - 3x Aluminum/Copper rods for the 3 elements of the antenna (Reflector, Dipole, Director);
  - 1x RG-58/U or RG-8X coaxial cable;
  - 1x 1:1 balun to match the impedance of the antenna to the one of the coaxial cable;
  - Nuts, bolts, and screws;
  - Mounting brackets.

2.4. Software design

The software will be connected to the hardware based on the next diagrams. The basic flux is shown in the simple one, the mission being divided on Pre-Parachute with Release State, Launch State and Landed State.

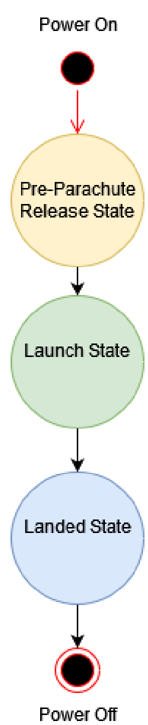

Figure 24: General State FSM

Each state is described in the completed diagram system :

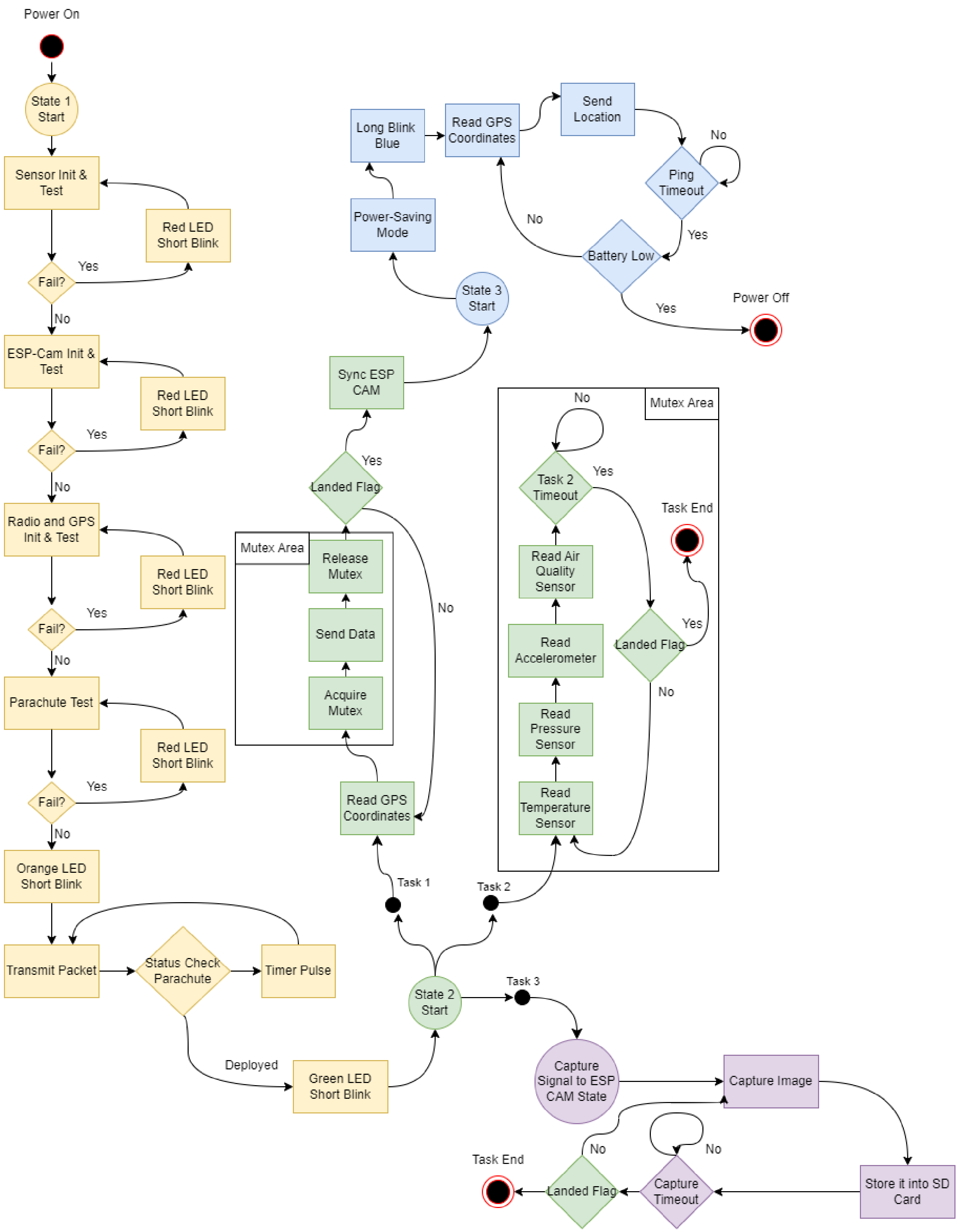

Figure 25: Detailed FSM

- State 1: Module initialisation and testing
- State 2: Capturing relevant data and transmission
  - Task 1: Relay GPS information and sensor data;
  - Task 2: Read sensors data;
  - Task 3: Capture images and store them on the SD Card.
- State 3: Transmit ping probes to localise the satellite

The 3 tasks are ideally run within a multi-threaded manner, to have the fastest response time.

2.4.1. Machine Learning Model

The machine learning model is used after landing and recovering the SD Card from the CanSat. It is based on processing the captured images of the secondary mission. Each saved image from the SD Card will be processed as follow:

- For increase the quality of the final result we will pass the image through deblurring and/or sharpening algorithms.
- Image patching: Because the ML model needs a specific resolution of the inputs, we resize the image to get to a multiple of model needed resolution. After the resizing, we divide the image into 28x28px resolution inputs for ML model.
- The pre-trained ML model takes as inputs the patches from the previous phase. Each patch will be classified accordingly.
- We are interested only on the crops fields results. The final image will be constructed with the resulted patches from the previous phase.

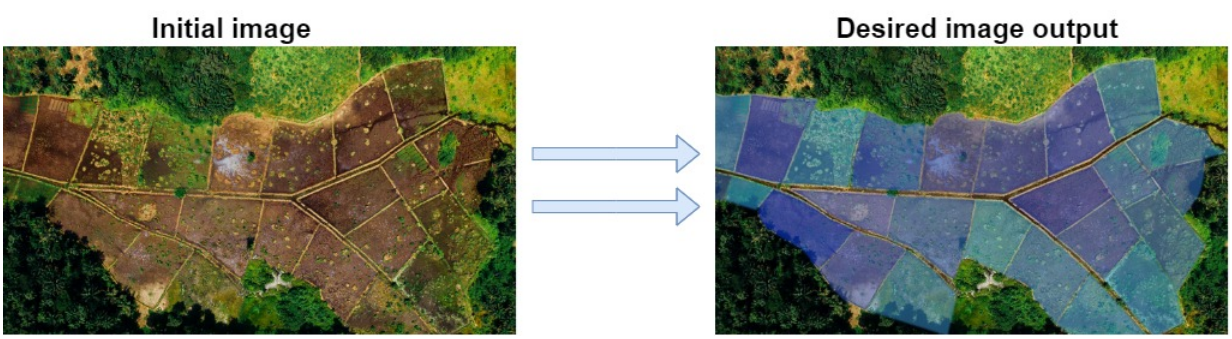

Figure 26: Principle of machine learning outputs

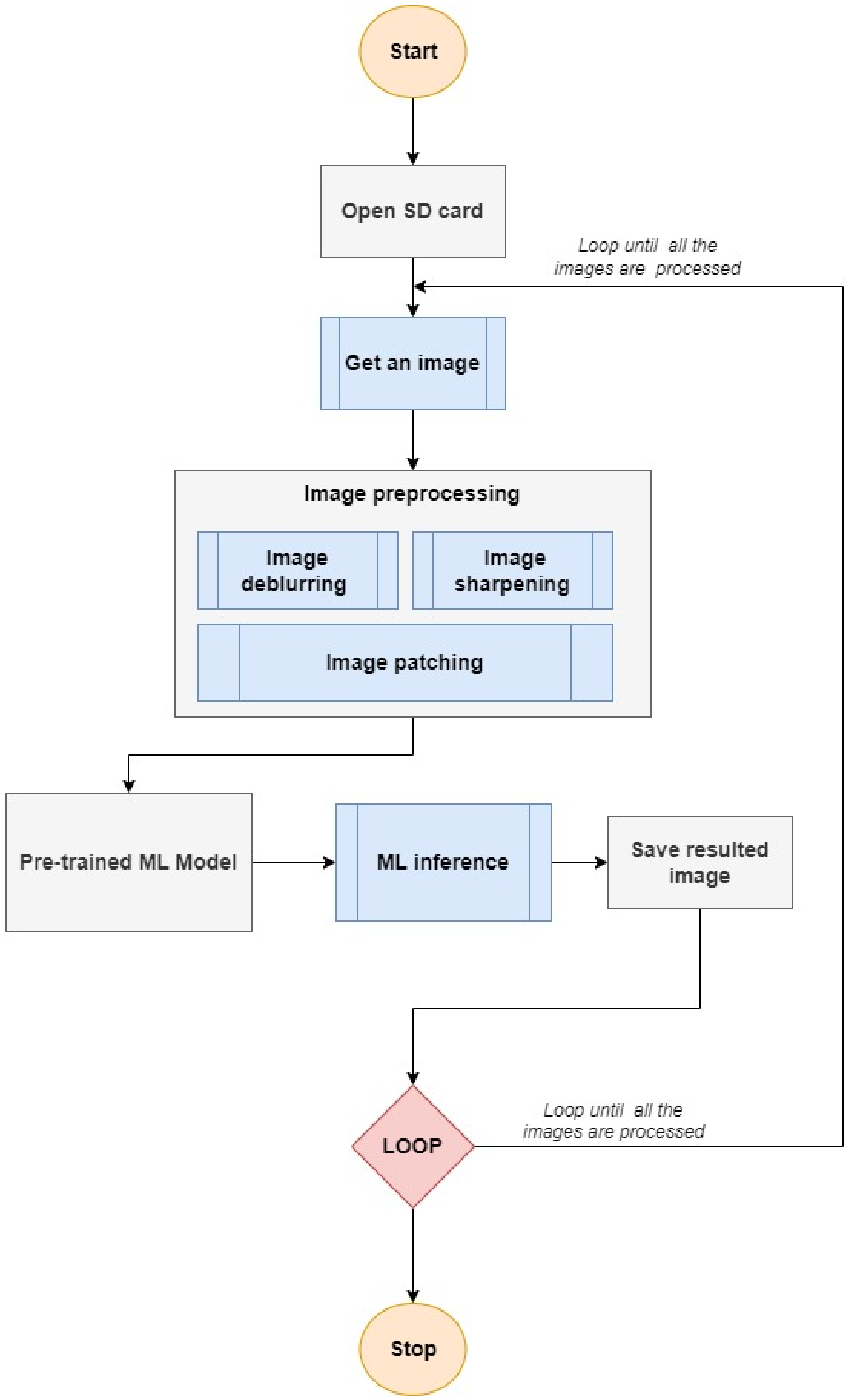

Figure 27: Machine learning model

2.4.2. CanSat fall simulation

This simulation will consist of a Python program that receive and show the information from sensors in real-time and an easier way. Interface will be consisted of:

- Graphs of sensor data according to altitude:

Will consist of displaying the values received from the sensor on the graph, based on the altitude.

- A preview of the CanSat motion:
  Will consist of moving CanSat in different position and observing only if the motion simulation is displayed accordingly yo the movement of the CanSat.
- A preview of the CanSat road map:
  GPS CanSat movement on the map test will consist in moving in different directions and for various distance with the CanSat. The map must accurately show the previous movement path of the CanSat.
- Last received value of each sensor.

The program will be write in Python and will be executed for all the duration of the flight. In the same time we show the data on the display, we will store them in different files. The altitude will be computed with the help of GPS sensor and the acceleration.

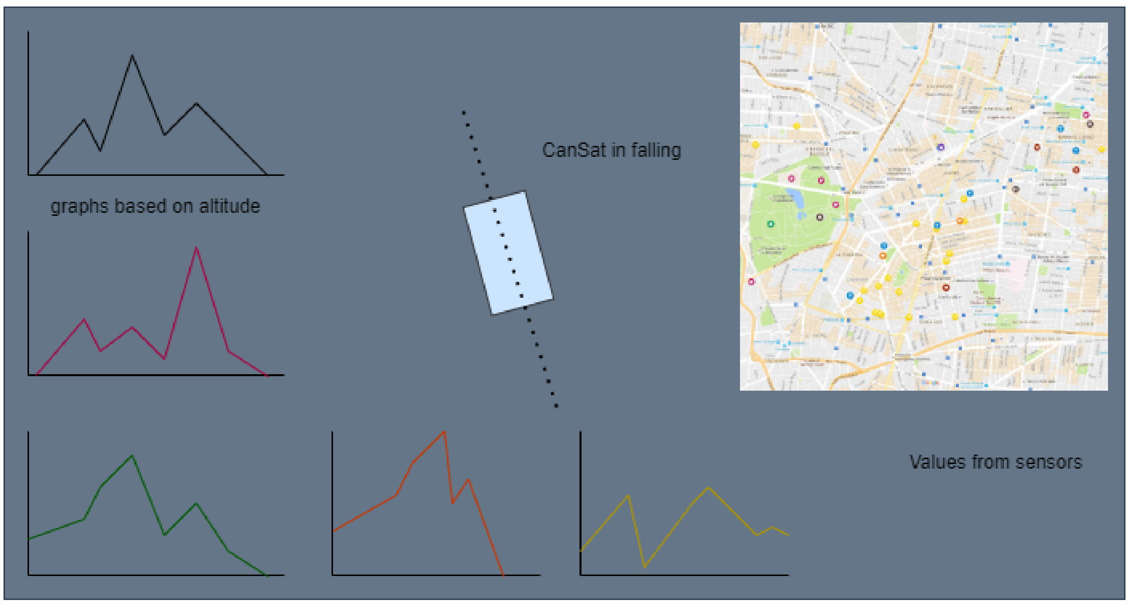

Figure 28: Simulation design

2.5. Recovery system

As a recovery system we have decided on using a parachute. The drag force, which can be easily calculated will allow the CanSat to fall with targeted speed of 8 - 11 m/s. Based on our wish to do as many images as we can we decided on terminal velocity around 8 m/s. Using the terminal speed equation we can figure out the area of the parachute. We choose as the shape of our parachute the round canopy, as it has the highest drag coefficient among the square, cross and round. We will consider to do a spill hole that would let the air flow in a controlled way. The usually diameter of the hole is around 20% ($\frac{1}{5}$) of the diameter of the canopy, which reduces the total area of the parachute by only 4%. The canopy will be attached on a swivel which will in turn be attached to the CanSat.

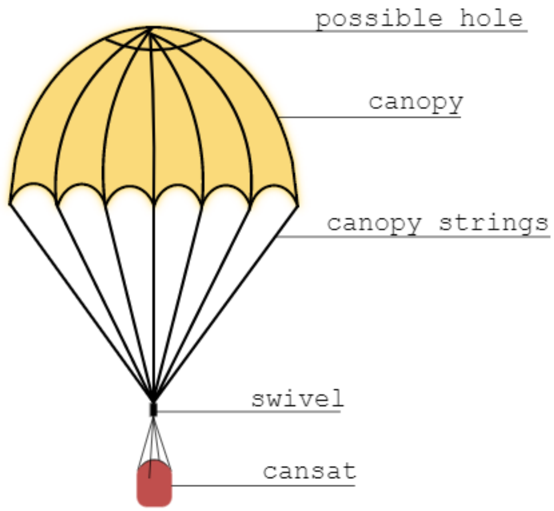

Figure 29: Parachute design

- Air density $\rho = 1.125 \dfrac{kg}{m^3}$
- Gravity acceleration approximation $g = 9.81 \dfrac{m}{s^2}$
- CanSat mass approximation $m = 0.3kg$
- Drag coefficient for canopy $Cd = [0.5; 1.5] - \textit{to be calculated}$
- Terminal velocity of CanSat $v = 8 \dfrac{m}{s}$
- Pi constant approximation $\pi = 3.1415$
- A - *area of canopy*
- R_1 - *radius of the canopy*
- R_2 - *radius of the spill hole*

$$\begin{cases} M = m * g \\ D_f = \frac{Cd * \rho * V^2 * A}{2} \end{cases} \implies A = \frac{2 * m * g}{Cd * \rho * v^2} \quad (1)$$

$$5\% * A = \pi * R_2^2$$

$$A = \begin{cases} \pi * R_1^2 - \pi * R_2^2 = \pi * R_1^2 * 95\% & \textit{we have a hole} \\ \pi * R_1^2 & \textit{we do not have a hole} \end{cases}$$

$\implies$ From here we can find the radius of the canopy based on the data we have

Products that will be used:

- Canopy fabric: Nylon ripstop
  - price: 14 €;
  - dimension: 1.5 x 2 m.
- Canopy lines : Polypropylene string
  - price: 1 €;
  - dimension: 30m x 0.8mm;

- Swivel: SAXA BRASS ROLLING SWIVEL - SIZE 6 - Whirlpool
    - price: 3.86 €.

Figure 30: Nylon Ripstop

Figure 31: Polypropylene string

Figure 32: Swivel

2.6. Ground support equipment

The ground support equipment will consist of:

- Laptop(s)
- Ground receiving station:
    - Microcontroller
    - RFM95W
    - Yagi-Uda directional antenna
    - USB connection to the laptop
- Software application

2.6.1. Laptop

We will have one or two notebooks that will have the software part installed and will assure the communication.

2.6.2. Software application

It consists of the machine learning model that will be installed on the laptop and the communication application. These are described in the software section.

2.6.3. Ground receiving station

The ground station is just a point-to-point connection to the CanSat, which transmits the data feed to the PC connected to it by serial USB. It has a Yagi directional antenna to compensate for the bigger spreading factor of the LoRa modules. The software on the PC will log the data, which then it will be processed by another module to fulfill the primary and secondary objectives.

3 Project planning

3.1. Time schedule of the CanSat preparation

We use Notion for time and tasks schedule.
The basic tasks are divided according to the time at which they have to be completed (as seen in the next figure) :

- Documentation (Formal deadlines and periods)→ IN PROGRESS
- Design → IN PROGRESS
- Build software → TO DO
- Build hardware → TO DO
- Test and integration → TO DO
- Outreach → TO DO

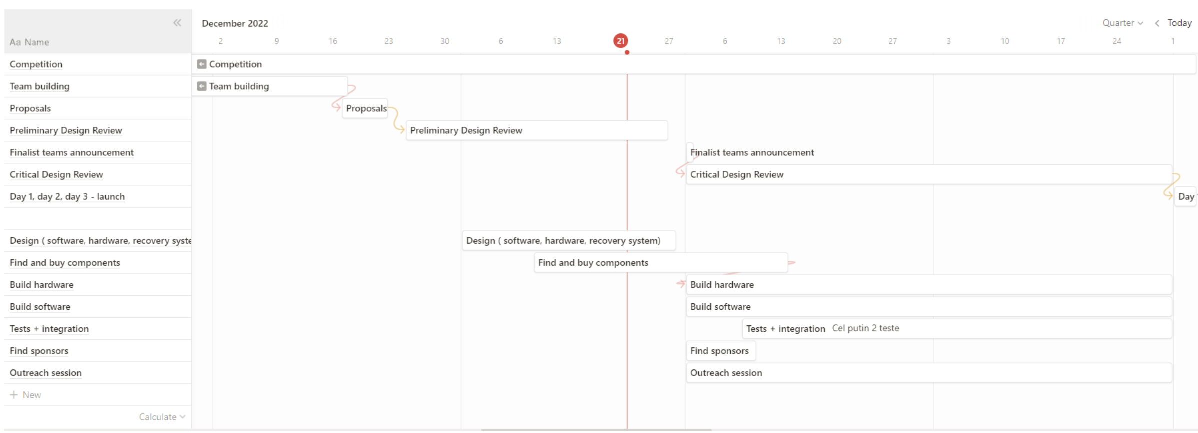

Figure 33: Timeline schedule

Each task is divided in more activities that must be completed based on task's time schedule. Each activities has his own members and some details about it, as seen in the picture. This is not the final list, as we could add more tasks/activities in the coming weeks.

Not started 10

Activities	Member(s)	Task	Details
Build recovery system	Ada	Building hardware	*build and test canopy *buy strings and swivel
Build communication ground system	Marius, Andrei	Building hardware	install software on laptops, unit testing
Build ground antenna	Andrei, Marius	Building hardware	buy components
Machine learning model	Ada, Teo, Dragos	Building software	find datasets, build model, test
CanSat simulation	Ada, Teo	Building software	python library
Build cansat hardware	Andrei, Marius, Dragos	Building hardware	
Prepare test plan	FULL	Testing	unit testing and integration testing
Image preprocessing	Ada, Teo	Building software	segmentation, blur/sharpen, resize
Program microcontrollers	Andrei, Marius, Dragos	Building hardware	
Instagram + facebook page	FULL	Outreach	after PDR

+ New

COUNT 10

In progress 5

Activities	Member(s)	Task	Details
PDR	FULL	Documentation	get text from the team
Design software applications	Teo, Ada	Design	
Buy recovery system components	Ada	Buying	
Buy hardware components	Andrei, Marius	Buying	
Design ground antenna	Andrei, Marius	Design	

+ New

COUNT 5

Done 4

Activities	Member(s)	Task	Details
time schedule	Ada, Teo	Documentation	
Design circuit's design	Andrei	Design	
Design Recovery System	Ada	Design	
Design mechanical structure	Andrei	Design	

+ New

COUNT 4

Figure 34: Activities list

3.2. Resource estimation

3.2.1. Budget

Component	**Cost (€)**
2 x Microcontrollers	10
Temperature sensor AHT10	1.19
Atmospheric pressure sensor BMP280	0.35
Air quality sensor MQ135	1.03
3-axis Accelerometer and Gyroscope MPU6050	1.68
2 x LoRa radio module RFM95W	9.9
GPS module GY-NEO6MV2	2.69
Camera module ESP-32CAM	6.35
Boost converter	2.05
Battery Pack	3.1
Nylon ripstop fabric	14.25
Polypropylene string	1.0
Saxa Brass Rolling Swivel	3.86
Ground Station Antenna	10
PCBs	5
Misc	13.5
Total budget	**85.95**

3.2.2. External support

The mission is supported from personal funds for now. We are looking forward to get in touch for with some companies that are working with hardware components, so they would sponsor us. For testing the prototype we are looking to use a drone from our University.

3.3. Test plan

3.3.1. Primary mission tests

1. Measurement tests

   - pressure sensor test:
     Will consist in exposing the sensor to a pressure change and comparing the results with a reference device. The test will be considered success when both sensors show the same values.
   - temperature sensor test:
     Will consist in comparing the results of the sensor with a reference device. The test will be considered a success if it will show the correct temperature.
   - air quality sensor test:
     Will consist of comparing the result of the sensor with a reference device or value.

3.3.2. Secondary mission tests

1. Measurement tests

- GPS sensor tests:
  Will consist in changing the position of the sensor in a bigger place and verify with another device the data transmitted.
- Camera module:
  Will consist of taking test images and verify the quality and resolution. The images will be compared with a reference set made with a phone or camera and calibrated to send the most suitable.

2. Machine learning model tests
   - Image deblurring test:
     This will consist of applying the deblurring algorithm over a set of blurred images. The test result will be recognized as successful only if the algorithm outputs satisfying quality improvement (quality evaluation visually or using standard image quality metrics such as PSNR and SSIM).
   - Image sharpening test:
     This will consist of applying the sharpening algorithm over a set of low-contrast images. The test result will be recognized as successful only if the algorithm outputs satisfying quality improvement (quality evaluation visually or using standard image quality metrics such as PSNR and SSIM).
   - Testing performance of deblurring/sharpening using real images from CanSat's landing tests.
   - Image patching test:
     This will consist of applying the patching algorithm over a set of images. The test result will be recognized as successful only if the algorithm proper patches the image, in the right order and each patch has the corresponding dimension.
   - Machine learning model test:
     The machine learning will be trained with 80% of the land classification dataset we use. The test will consist of analysing the accuracy and precision of the prediction for the remaining 20% of the dataset. The test result will be recognized as successful only if the algorithm outputs satisfying accuracy and F1 score.
   - Image reconstruction test:
     This will consist of patching a set of images and assembling the patches resulted. The test result will be recognized as successful only if the algorithm outputs the an image identical to the original.
   - Entire image preprocessing module test:
     This will consist of preprocessing a set of land images (normal, blurred, low-contrast) with various dimensions. The test will be recognized as successful only if the module outputs an image that satisfy the quality image metrics and the overall crops fields prediction seems closer the ideal case.

3. Simulation tests
   - Preview the CanSat road map in the correct mode:
     The trajectory need to be visible and clear. We need to be able to identify the place our cansat landed based on the place of launch and this map.
   - Showing the last value received:
     We show the last value received in a corner of the program interface. This values need to be shown in real-time and correct.
   - Graphs based on the sensor's data:

The data from sensors need to be correctly mapped in the graphs. The altitude need to be correctly calculated.

- Simulation of the CanSat:
  The CanSat need to shown in real-time. The rotation and position need to be correctly calculated and used to show the CanSat.

3.3.3. Communication system range tests

Our main objective is to be able to transmit and receive data over a longer distance than the one on the launch day. This is because we want to build a predictable and controlled behavior for our communication process. Therefore, the recipe for this is being able to transfer and fully receive the largest possible packet of data over a distance of 2000m (double the one our CanSat will reach on launch day) in the shortest period of time.
We say the largest because we want to provide our ground station with all the information required. We say the shortest period of time in order to ensure the real-time aspect of our project. Hence the formula:

- $Data\ rate = \frac{Spreading\ factor}{2^{Spreading\ factor}} * \ Bandwidth\ * \ Coding\ rate$
  - Data rate -> the amount of data that can be transmitted over a communication channel in a given amount of time (expressed in bits per second)
  - Spreading factor -> parameter that determines the duration of each transmitted symbol and the level of signal spreading. The lower the spreading factor the higher the data rate, but at the cost of reduced sensitivity and range.
  - Bandwidth -> determines the maximum data rate that can be transmitted over a communication channel, as well as the overall capacity of the channel to support multiple users or services
  - Coding rate -> the ratio of the number of bits in the transmitted signal to the number of bits in the original data. Parameter that affects the overall reliability and robustness of a communication system. A higher coding rate means that more redundant bits are added to the original data, which increases the error correction capability of the system and improves its reliability in the presence of noise or interference. However, a higher coding rate also reduces the effective data rate, since more bits need to be transmitted for each unit of information

We also have to take into account the fact that we are going to use a Yagi-Uda antenna which will support the reliability of transferring data. Therefore, these are the parameters we will start testing first:

- Spreading factor: 9 (higher values provide better range, but lower data rate)
- Bandwidth: 125 kHz (narrower bandwidth provides better sensitivity and selectivity, but lower data rate)
- Coding rate: 4/5 (higher coding rate provides better error correction, but lower data rate)

This will give us a value of 9.77 kbps for our Data rate. As mentioned above, we can alter these parameters in order to match our needs. The tests will take place in both open space areas and urban sites in order to cover a wider range of situations and calibrate our communication parameters to ensure a reliable transmission.

For the Yagi-Uda antenna we can calculate its SWR with both software simulation tools like this antenna modeler and optimizer for our model antenna and SWR meter for the real life antenna.

3.3.4. Energy consumption tests

The power consumption has been calculated for each component at peak, normal operation and special cases.

- AHT10: 1μA peak at 3.3V
- BMP280: 1.1mA peak at 3.3V
- MQ-135: 150mA normal operation at 5V
- MPU6050: 4mA peak at 5V
- ESP-32CAM: 180mA with flash turned off
  or
  20mA with modem sleep at5V
- RFM95W: 120mA RFOP +20dBm at 3.3V
- GY-NEO6MV2: 65mA peak/40mA normal operation at 5V
- Microcontroller: 20mA with modem sleep at 5V

The battery pack used for this project varries between structural alignments, but it will be either an 18650, a pack of 16340s or a LiPo battery of at least 3000mAh capacity. This gives us plenty of time to work with the CanSat, since the total consumption of the system at peak load is around 2.4 Watts. The boost converter has a η of 94%, and the 3.3V supply is achieved using the integrated LDO (Low-Dropout Regulator) of either the microcontroller or the ESP-CAM, giving an estimate of around 9 usable Watts, using a 2700mAh 18650 at 3.7 Volts. This can power the system for at least 4 hours before running down of energy.

3.3.5. Recovery system tests

- Parachute deployment test: We will mount the parachute on the CanSat to load equal the weight of the satellite. We will drop it from a window, a building and eventually from greater heights with a drone that has a specially designed system. The parachute must be fully open when the CanSat land on the ground.
- Parachute test for various weather conditions: We will drop the CanSat on bad weather too, and recognize the test as a success if the parachute is still reusable.

3.3.6. Endurance tests

- Drop resistance test: The team S4ndHunt3rs will be attempting a crash test. The test consists in dropping the CanSat and when the parachute open, no element would be damaged. This is important both when the parachute have ended to open and when the CanSat have landed on the ground. This test is realised after the parachute tests, from a sufficient height to achieve the terminal velocity.
- Shock resistance test: We will do a shaking test using a mechanism that will be decided. If the equipment is not damaged in any way, the test would be passed.

3.3.7. Test equipment

All the sensors and equipment need to be testing separately to see if they work and if they are compatible one with another.
The sensors are:

- Primary sensors: pressure, temperature, air quality;
- GPS sensor;
- Accelerometer and gyroscope;
- Camera.

3.4. Time management

The CanSat mission has a tight schedule we need to follow to be sure the mission will be completed in time. The tasks are similar to the ones from time schedule, as it seems:

- Design phase *Jan 25th - Feb 25th*
  - Define mission requirements and objectives
  - Design electrical, mechanical structure
  - Design software applications
- Building phase *Feb 26th - Apr 15th*
  - Build CanSat
  - Build Recovery system
  - Build Ground System
  - Develop software applications
  - Unit Testing
  - Modify Designs as you need to
- Test phase *Apr 16th - Apr 25th*
  - Integration Testing
  - Launch Testing
  - Data analysis Testing
- Launch phase *TBD*
  - Prepare for launch
  - Lauch CanSat

Even with a strict time schedule, the team identified potential risks we need to consider in the all phases. Theese are:

- Risk 1: Delays in obtaining materials, receiving wrong materials:
  - Mitigation: Place orders in advance and read specification really well.
- Risk 2: Failed tests:
  - Mitigation: Allow extra time for testing part and for calibration after that.
- Risk 3: Bad weather:
  - Mitigation: Build communication system and recovery system that allow to run on non-perfect weather.
- Risk 4: Software:

  - Mitigation: Intensive testing
- Risk 5: Recovery System is not working:
  - Mitigation: Verification at calculations and more tests

4 Data analysis and outreach

4.1. Data Analysis Plan

The software applications is described in the software design section.

For result analysis we will visualize the resulted graphics from the simulation and look for anomalies in the transmitted and stored data. We are verifying the route of the CanSat computed in time of the fall time.
As the image processing, in the first instance we will visually compare the original image against the resulted image, and final percent too. We can manual modify the code when we want to search for another land type/types in our images. For this we are dependent on the launch area.
In the future, we could automate this process by showing all the result in a report and ask the user what we are looking for in the begging.

4.2. Outreach Program

For our outreach program we want to be seen on social media platforms (Instagram and Facebook) and post about our status two-three times per month. This pages will be created after the PDR confirmation. This posts will consists of updates, videos, and new plans of the project and team.

We plan to do presentation about this competition and our project to our High Schools, in order to reach a new generation of students and help them to know about all the possibilities they have.

In our community of students, recent graduates and teacher, we plan to explain in more complex terms about this project, what we have done and how we could continue this project or similar ones.
This presentation would take place after the launch, so we could show the whole process.

5 Conclusion

5.1. Summary of the PDR

This document is the Preliminary Design Review our CanSat will be based in the following steps. All equipment is bought and ready to be put together in the CanSat. Any of the failed tests or changes will be resolved and documented in the future.

5.2. Recommendations for next steps

As recommendations for the next steps, we must be careful at the limitation of the CanSat, at correctness of the tests and calculations. We should have schedule meetings in order to progress smoothly in the build and test phase. As we said in the test part, we will ensure the CanSat is fully functional both in hardware and software, the communication covers the wished range and the data are full transmitted/received. As another recommendation is to schedule the full-test time to prevent anything that can go wrong.