



Guidelines for the Progress Report And Design Document

Team Name: Herotech

County: Sibiu



Table of Contents

1	INTRODUCTION	4
1.1	Team organisation and roles	4
1.2	Mission objectives	6
2	CANSAT DESCRIPTION	7
2.1	Mission overview	7
2.2	Mechanical/ structural design	8
2.3	Electrical design	13
2.4	Software design	25
2.5	Recovery system	27
2.6	Ground support Equipment	28
3	PROJECT PLANNING	29
3.1	Resource estimation	29
3.1.1	Budget	29
3.1.2	External support	29
3.2	Test plan	29
4	OUTREACH PROGRAMME	30
5	REQUIREMENTS	31

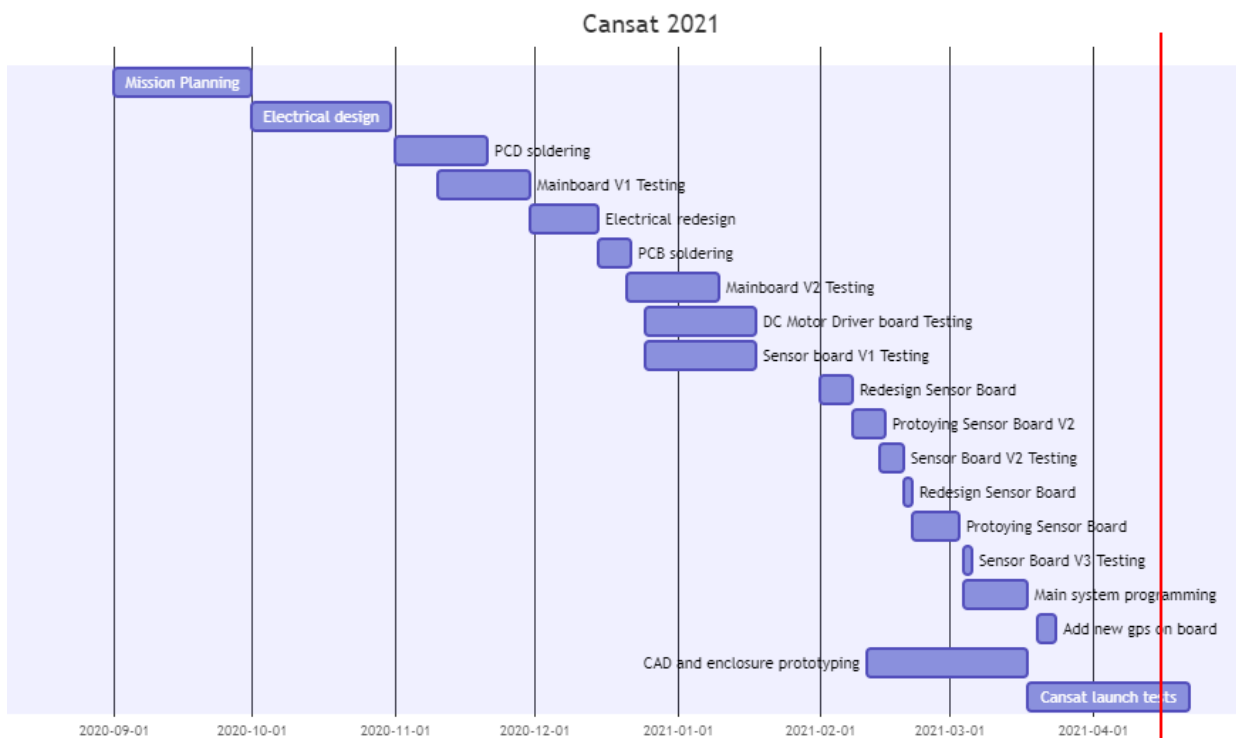


PROGRESS REPORT

i. The project is almost finished.

We are in the stage of testing our software. While the CanSat can carry the primary mission without problems, the secondary mission needs a lot more tuning in order to achieve our goal of landing in a designated area. We had more problems than we expected due to having to redesign the system boards multiple times and because of a short circuit that destroyed our main board. This put us a little behind schedule, but we are determined to finish everything we set out to do with this project.

ii. Task list



iii. Detailed project status

Our main problem revolves around the voltage regulator that is overheating because the voltage drop is high, and the electrical current is quite strong. Momentarily, we analyze the behavior of our voltage regulator and are looking for new solutions to our problems.

The GPS that we currently use takes too long to receive information's from the satellites at boot when performing a cold start.

We also cannot reach the 1km altitude desired to test out our CanSat with a drone.



1 INTRODUCTION

1.1 Team organisation and roles

Our team is composed of one teacher and 4 students as showed in the table below:

Name	Role	Field of work within the team	Expected workload within the team	Hours dedicated at school	Hours dedicated after school
Demco Andreea	Teacher	Supervisor	Supervising the team	-	-
Marpozan Daniel	Team Coordinator	Hardware, Assembly, PCB designing	Designing and making the PCB's and other electronical parts, as well as their functions	2h per week during the physics course	8h
Muntean Codrin	Team Member	Software writing	Writing the software and all the necessary code	2h per week during the informatics course	1h
Radac Alexandru	Team Member	Mechatronics design and components assembly	Designing, printing, and assembling of components and casings	2h per week during the physics course	3h
Bleaja Sebastian	Team Member	Visual design, text writing, and illustrations	Designing the aspect of visuals, writing, and translating the texts	2h per week during the informatics course	2h

The team was formed by Daniel, Alexandru and Codrin in 2018 and is competing in electronics, informatics, and robotics contest ever since. Today, all the four members are in the last year of high school and are pursuing varied universities from inside and outside the country. The background of each member varies greatly as each has a different personality and way of experiencing the world around them.

To start off, Marpozan Daniel, the team's coordinator, is an organized and hard-working person. Under the cheerful attitude that he always has is an exceptional rationality that allows him to objectively lead the team. His background is centered around electronics and informatics



competitions. He is currently pursuing applied electronics at UTCN university in Cluj and his interests revolve around mechanical engineering and electronics.

Next is Muntean Codrin, the sole programmer of Herotech that writes all the necessary software and code. He could be most definitely characterized as not very talkative and often shy, but his ability to write code is proven by the fact that Herotech does not need other programmers to keep up with the amount of code needed. His background revolves gravitates around informatics and contests as he always had a passion for programming. His first code competition was back in the 9th grade when he participated in the informatics Olympiad. From that point on he, along with his other two colleagues formed Herotech and participated together in many competitions. Codrin plans to pursue software development at the UBB university in Cluj.

Radac Alexandru is responsible for designing the parts and components our team needs. As a person, he is a visionary. His imagination and calm temperament allow him to create perfect parts that fit together perfectly. He always had a passion for aeromodelling and physics and that made him take extra-curricular aeromodelling courses before entering high school. Having entered high school and taking part in the foundation of the Herotech team he has completed the team and is designing parts, components and planning how they should work ever since. Alex plans to pursue aerospace in TU DELFT in the Netherlands.

Lastly, Bleaja Sebastian, the latest addition to the team is responsible for the non-technical part of the team like visual design, social media, including translating and writing the teams texts. The word that defines Sebi the best is open-minded, as it is a crucial aspect of the imagination needed for completing his tasks. His background started in the 9th grade with a position in the high school's student's council. In that field followed 4 years of work in the student council in various positions, the most important being the VP of the county council of students. Aside from the student council he participated in many writing competitions, public speaking competitions and has collaborated with many entities such as ONG's through his position. Moreover, he has a background in article writing, having published numerous articles, and having experience in layout design. He entered Herotech in 2019 being needed in a position that required his skills. Ever since then he has worked as a permanent team member. He is planning to pursue medicine at the Medicine University of Vienna in Vienna.

Competitions that Herotech pursued as a team are:

- First Tech Challenge Season 4 and are currently participating in season 5
- Genius Olympiad
- World Robot Summit
- ENEL electronics competition
- + numerous small competitions

As of this year, Herotech has become a Non-Governmental-Organisation. This choice to move from robotics team to NGO was because it makes collaborating with different entities easier, not needing a third-party association to make partnerships and sponsorships. Another reason for founding an NGO was for the sake of the future development and growth of Herotech.



1.2 Mission objectives

The secondary mission we have chosen consists in a guided airdrop system. The CanSat should be able to guide itself during the fall by pulling the strings attached to the parachute with the target of landing in a nearby area relative to the dropping point. This presents a great potential in the development of an air drop package delivering system for faster and more convenient delivery which does not require human interaction. It is also a solution for dropping necessary items to people in need or for military drops. A CanSat of this size cannot carry much since it's light, but the project can be scaled to handle small items such as food packages, a medical kit and other similar-sized objects.

The most important part of our mission is the control system. The exact position of our landing cannot be achieved with a lot of interfering factors, but we strive for stability. In order to call the mission a success we have the following goals:

- Landing of the CanSat without damaging the carcass or the internal systems.
- Correctly reading data provided by sensors for further analyzing.
- Guiding the CanSat for landing with high accuracy in the designated area.

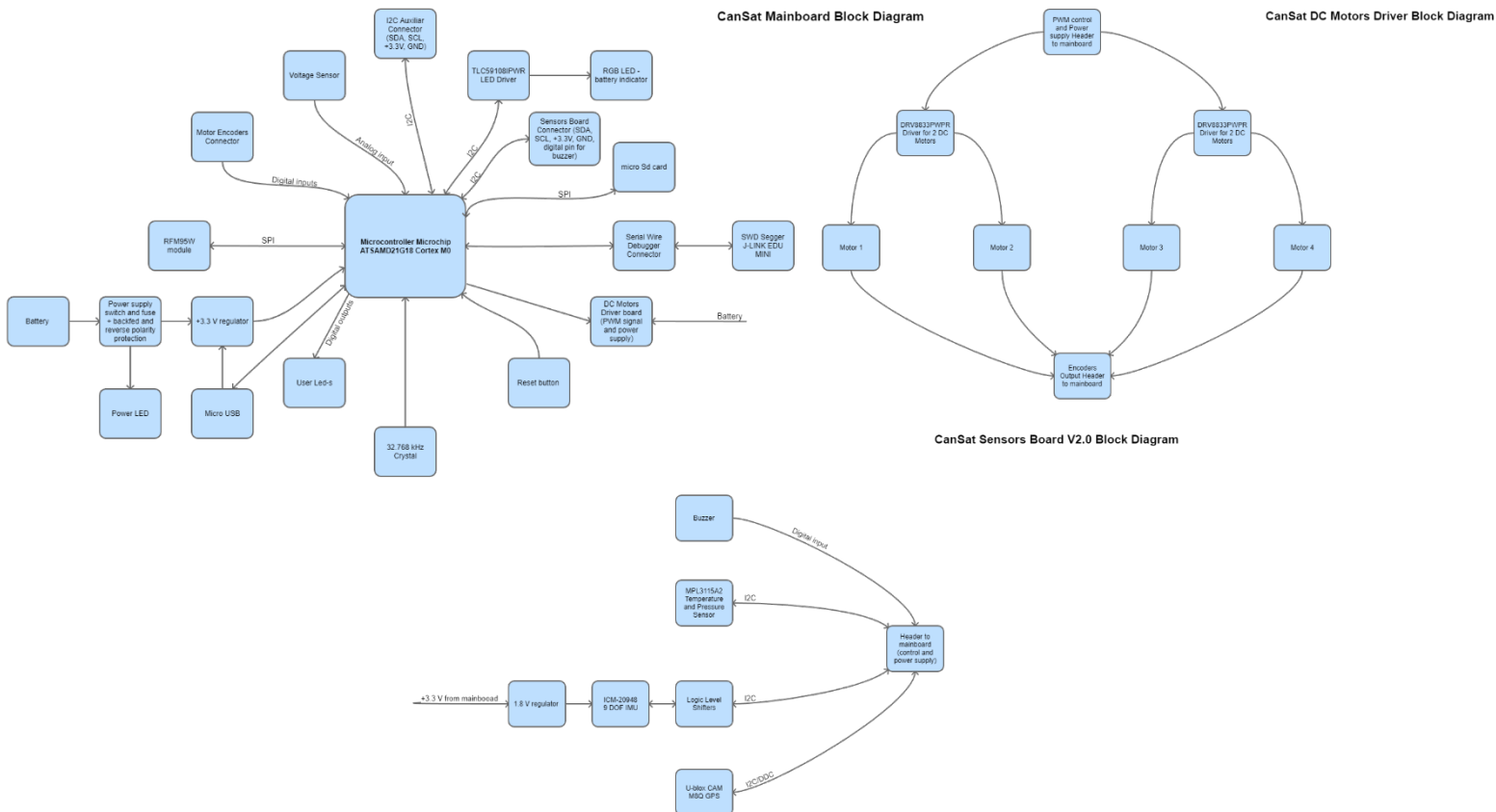


2 CANSAT DESCRIPTION

2.1 Mission overview

The CanSats mission objective is to collect data while descending with the help of a parachute and land in a designated area having a radius of no more than 25 meters. The mission begins with the CanSat being in a state of low battery usage. After being thrown from an altitude of 1km, its parachute immediately opens. Data from the accelerometer is continuously being sent to the main board, which will detect the beginning of the fall, switching all other functions on. While falling, the mainboard saves data on a secure digital card regarding altitude, temperature, pressure, geographic coordinates which are being read from a GPS and a pressure sensor. For the CanSats navigation, the global position at a point in time is determined by the GPS. The CanSats forward vector is determined by a magnetometer device whose tilt is compensated by the accelerometer and gyroscope. Comparing the current position to the target landing destination the CanSat will determine which direction it should face, and 4 motors will determine the parachute to go to that direction by extending or retracting the ropes connected to the parachute. After landing the CanSat will switch to a low power mode, stopping data collection and turning on a buzzer to facilitate its recovery.

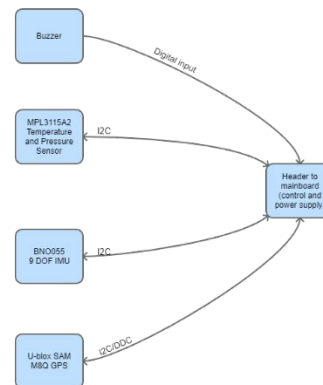
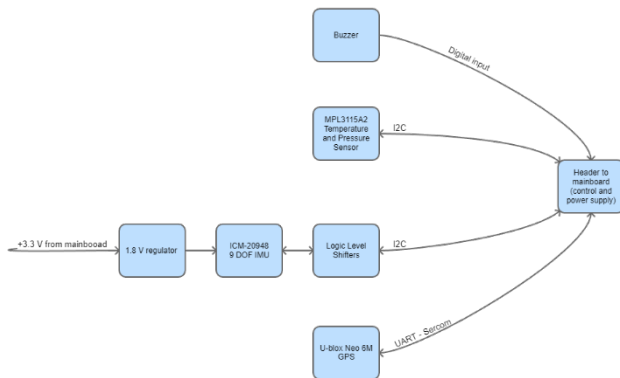
Block diagram:





CanSat Sensors Board V3.0 Block Diagram

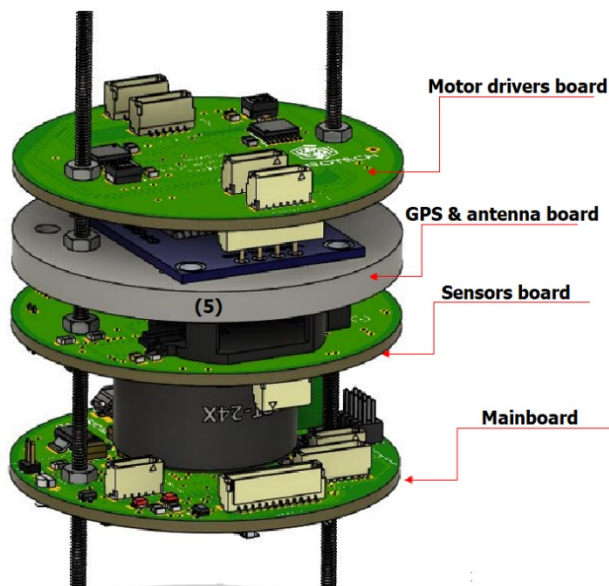
CanSat Sensors Board V1.0 Block Diagram



2.2 Mechanical/ structural design

The CanSat was firstly designed using Autodesk Fusion 360, where all the 3D printed parts were modelled. We started by creating a maximum volume to make sure that everything we design was in parameters.

The first step was to place all the boards on top of each other, leaving as little space as possible between them.



The second step was to find and buy a suitable battery to fit in the CanSat. We decided to use one LiPo 800 mAh 7.4 V accumulator.



Next, we started modelling the parts necessary to close and protect the circuitry, keeping in mind the fact that they had to hold up at 20G of acceleration.

After all the required parts were modelled, we started printing the using PLA as we decided that it's properties would best suit the purpose of the CanSat and assure proper strength.



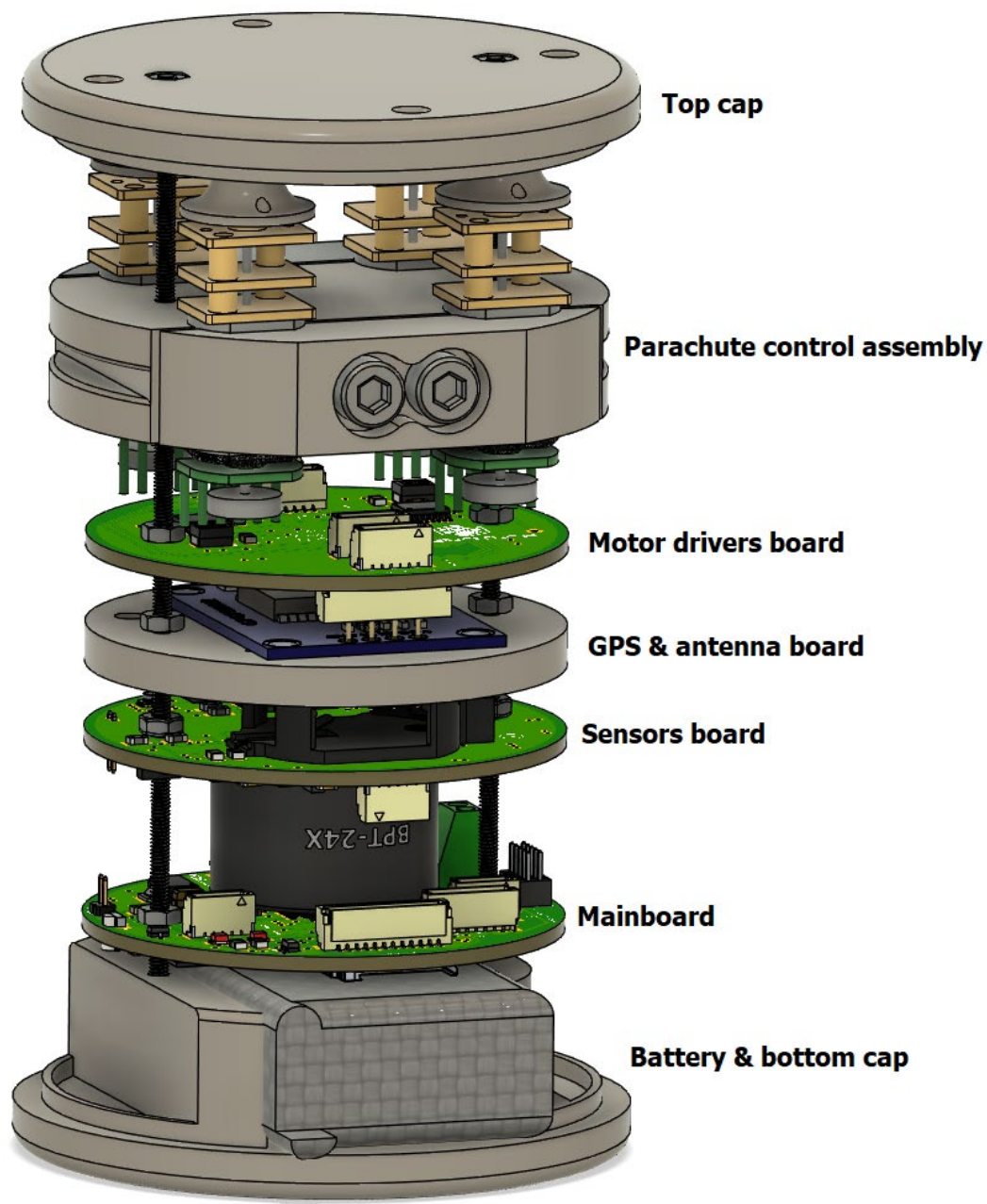
We were aware that the GPS would run at quite a high temperature, so we measured it (70 ° C) and it was too high for a PLA board to not deform at that temperature, especially at a high G force during launch. Due to this fact, we decided to use ABS to print it which is a better thermoplastic than PLA when it comes to temperature resistance.

COMPONENT	MATERIAL	FILAMENT (GRAMS)	MASS
TOP CAP (1)	Polylactic acid	11.86	
RIGHT WALL (2)	Polylactic acid	27.07	
LEFT WALL (3)	Polylactic acid	26.87	
BOTTOM CAP (4)	Polylactic acid	15.17	
GPS & ANTENNA BOARD (5)	Acrylonitrile Butadiene Styrene	5	
MOTOR BRACKET (A) (6)	Polylactic acid	3.01	
MOTOR BRACKET (B) (7)	Polylactic acid	3.11	
MIDDLE MOTOR BRACKET (8)	Polylactic acid	10.09	
STRING PULLEYS (9)	Polylactic acid	3.6	
		TOTAL:	105.78

When all the pieces were put together, we used the remaining space to incorporate the Parachute control assembly.

The motors have a gear reduction of 210 to 1, giving us a high torque to pull the strings of the parachute.

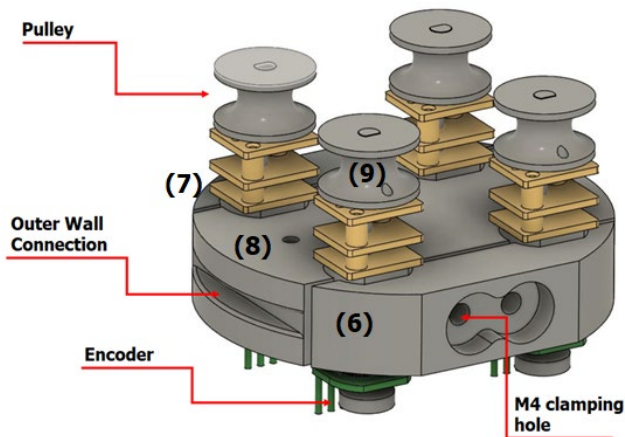
After printing all the parts, we assembled the CanSat using two M2 threaded rods that go through the whole CanSat and keep all pieces in place.



For extra protection, the CanSat is covered in a two-piece wall that also connects to the Parachute control assembly to provide more stability.

We are using plastic M2 nuts with 3D printed spacers between the boards to be sure we are not tightening the nuts too hard, damaging the PCB.

1. The parachute control assembly:



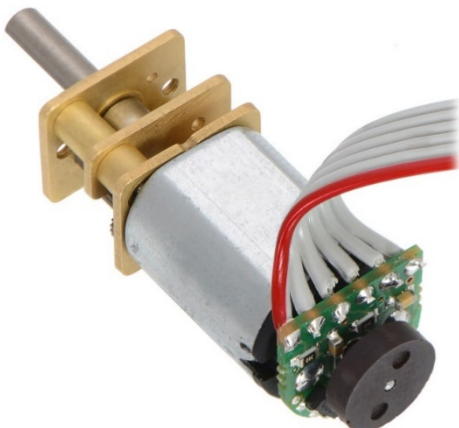
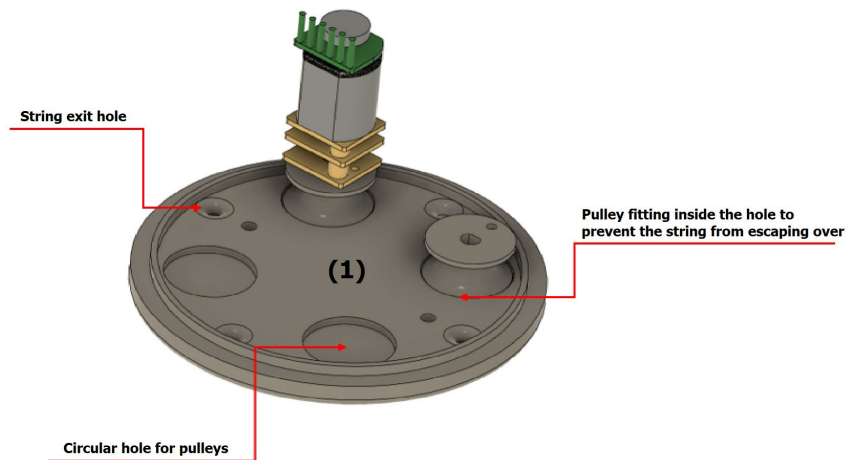
This assembly forms the upper part of the CanSat, and its purpose is to steer the parachute, allowing us to attempt a guided landing.

We are using 4 DC micromotors, each one of them connected to the body of the CanSat with 3 3D printed parts that clamp them together with the use of two long M4 Screws. The printed parts are secured to the rest of the CanSat in 4 points: -2 by the M2 rod and 2 by pressing the sides in the outer walls, providing a tight fit and ease of removal.

To each one of the motors, 3D printed pulleys are attached that allow the length of the strings of the parachute to be completely controlled.

With this in mind, we can completely steer the parachute, which is a paraglider, by pulling on the side strings, or even slowing the descent rate by pulling all the strings simultaneously.

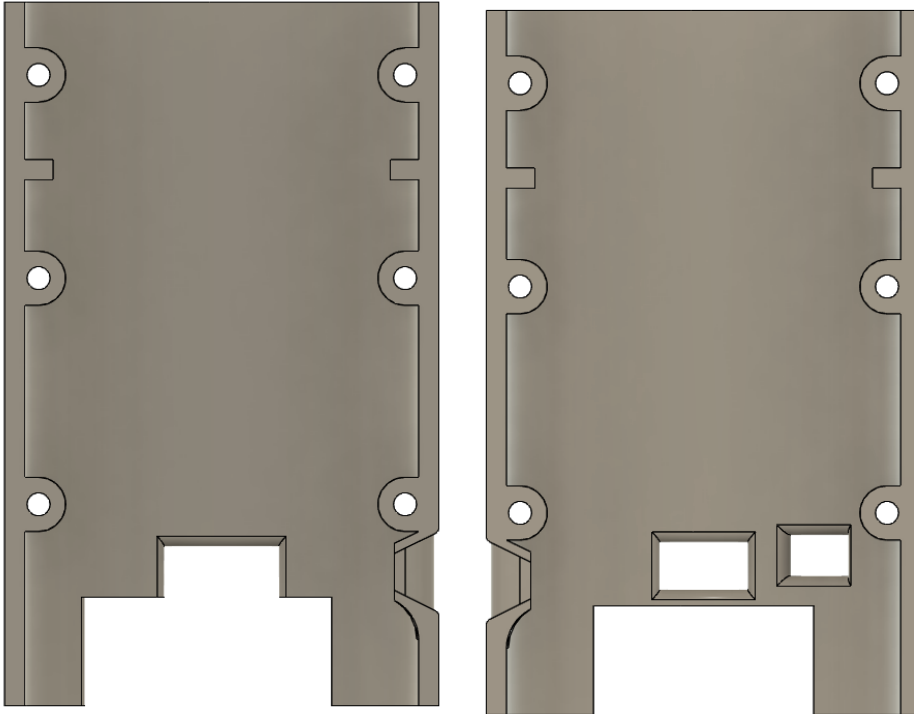
The motors are equipped with encoders that we use to make sure the parachute is correctly controlled.





HEROTECH

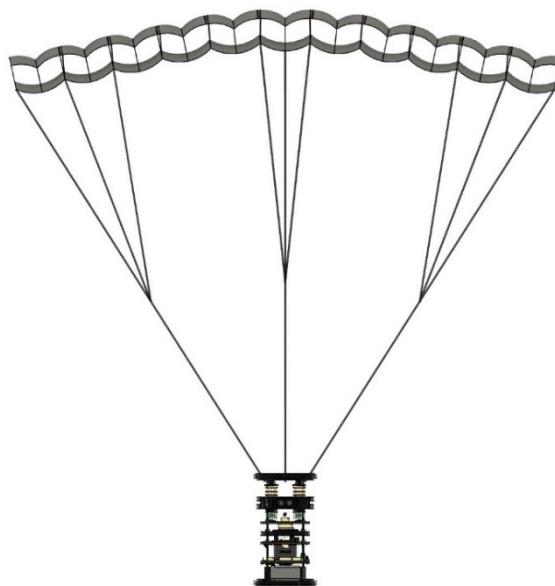
2.The outer walls :



These two walls have the sole purpose to stabilize the CanSat and keep the circuits safe.

They have a thickness of 3mm and are made from PLA, making them quite sturdy.

In the upper part, both have a little extrusion which connects to the Parachute control assembly, keeping it place.





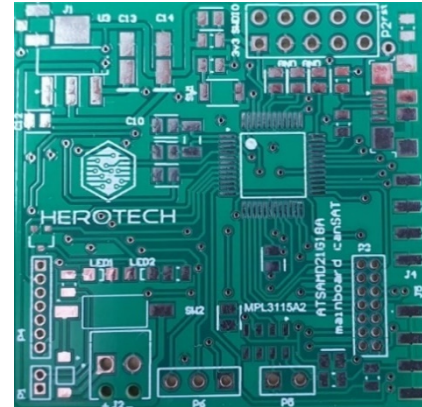
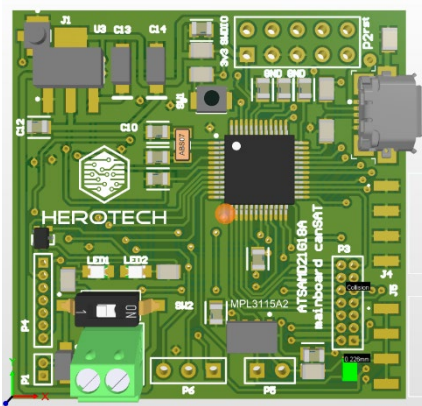
2.3 Electrical design

1. The mainboard

We started with the idea of developing our own printed circuit boards in order to make a customized application with the appropriate requirements. After a good documentation on the available microcontrollers, we decided to use an Arm Cortex M0 processor, being one of the smallest 32-bit processors available on the market. The decision was based on its performance, power consumption, programming accessibility and price. We use the ATSAMD21G18 microcontroller which is ideal for a wide range of automation and industrial applications. It has 256 KB of flash and 32 KB of SRAM with an operating frequency up to 48 Mhz. One of the main advantages of this family of microcontrollers is the serial communication module (SERCOM), configurable as UART, SPI and I2C. Another characteristic that was crucial for our application was the PWM generator capability, having 20 PWM configurable channels. The power supply range is between 1.62 V and 3.63 with a low power consumption. The idle and sleep modes are available for this chip along with many others useful features for a complex embedded system. It comes in a variety of 48 pin packages, but we chose the TQFP one for soldering purposes.

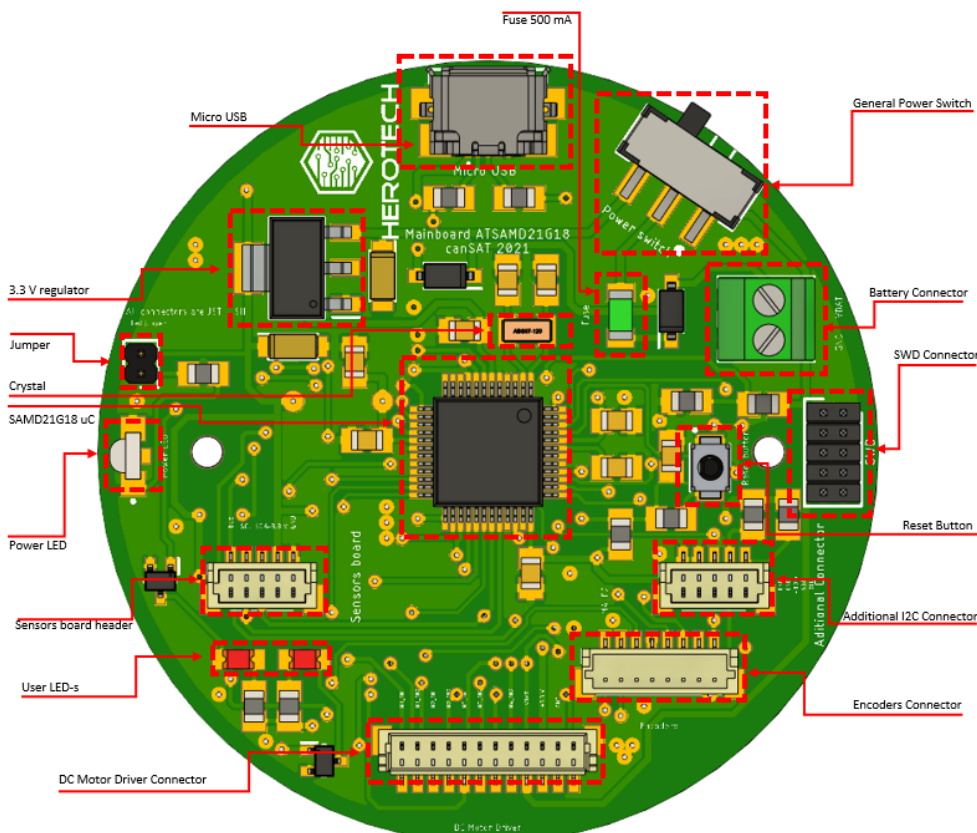
Initially we planned to use the microcontroller on a custom board, but when we realized that the programming part was quite difficult (the available libraries were not compatible with the hardware), we decided to use the same design as an Adafruit Feather M0. In this way, the programming was more accessible because any available library could be easily implemented in our project. We use the same pins as the Feather M0 with small modifications.

After choosing the uC, we started designing the schematic of our project. Being just a test board, just the uC, a radio communication module, and a temperature and pressure sensor were integrated as well as the required auxiliaries (the power supply, the micro-USB, SWD). We left all the pin headers available for further applications on the test board. We designed it in Altium because, at that time, we had a free student license. It was a challenge for our team to work with such a complicated design software, but in the end all the experience was a success. The schematic was designed accordingly to the information specified by the supplier in the datasheet. We followed the recommended typical application schematic and layout. After we designed the schematic and the layout of the PCB-s, we manufactured them with a specialized company. In all iterations of the board, we used 0805 SMD packages for the ceramic capacitors and resistors.

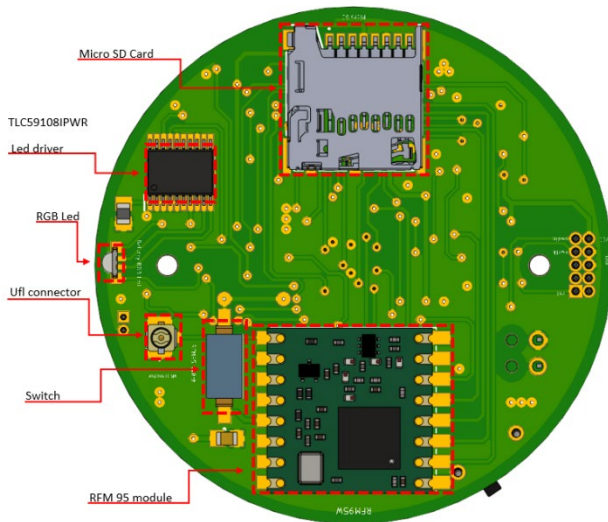


First design of the mainboard

After populating the PCB, we burned the bootloader and did some tests on the board. The results were promising, so we started to redesign it considering all the CanSat requirements. The following descriptions of the mainboard are applicable for the second version.

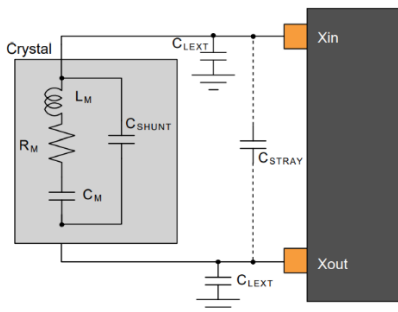


Second design of the mainboard – top



Second design of the mainboard – bottom

The uC requires a few components to work properly: an oscillator (crystal), some ceramic capacitors and a serial wire interface for programming and debugging. As for the **external real time oscillator**, we used ABS07-120-32.768KHZ-T from Abracon.



In order for the uC to work properly with this Quartz Crystal oscillator, a pair of ceramic capacitors have to be added. The value of those can be determined using the following formula:

Load Capacitance Equation

$$C_{LOAD} = ((C_{XIN} + C_{LEXT}) * [C_{XOUT} + C_{LEXT}]) / ((C_{XIN} + C_{LEXT} + C_{LEXT} + C_{XOUT}) + C_{STRAY})$$

Where:

C_{LOAD} = Crystal Mfg. C_{LOAD} specification

C_{XIN} = XOSC XIN pin data sheet specification

C_{XOUT} = XOSC XOUT pin data sheet specification

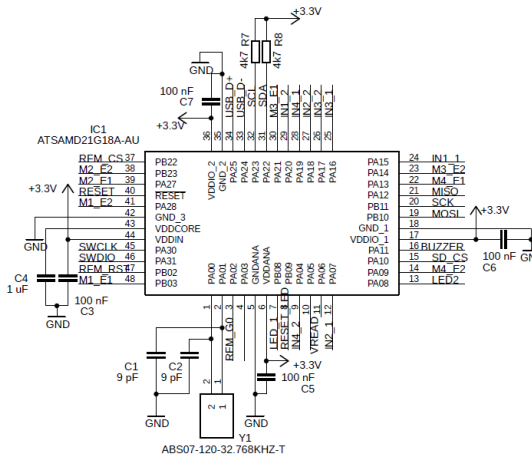
C_{LEXT} = Required external crystal load capacitor

C_{STRAY} (Osc PCB capacitance) = 1.5 pf per 12.5 mm (0.5 inches) (TRACE W = 0.175 mm, H = 36 μ m, T = 113 μ m)

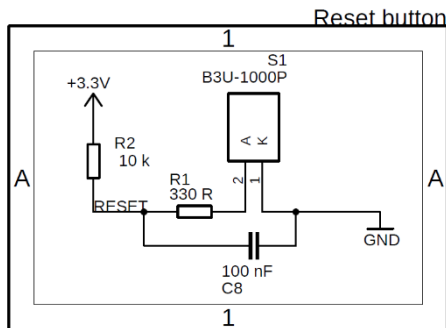
Table 37-48. Crystal Oscillator Characteristics

The purpose of these capacitors is creating a phase shift in order to achieve a 360-degree oscillation. They also have an anti-parasitic purpose.

For a proper functioning of the uC, decoupling capacitors are required between V_{DDIN} and GND and between V_{DDCORE} and GND.

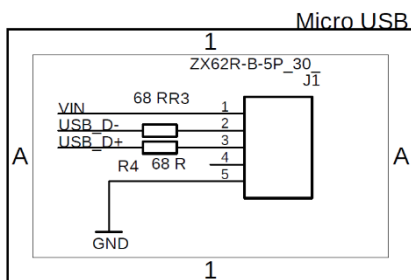


For the **external reset**, a small SMD Omron Push Switch is used. The rest pin is pulled up using a 10 kOhm resistor.

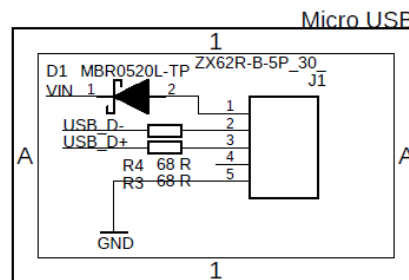


Without the debouncing capacitor (C8), when the switch is pressed, the digital input reset pin gets multiple counts rather than the expected one single count. To prevent this phenomenon, we integrated a small ceramic capacitor that provides a clean edge on the reset input pin

The **USB interface** requires the same trace length on the PCB, with minimum number of VIA-s and angles. On the data lines, two resistors are placed in order to match the traces impedences with the required impedences suggested by the manufacturer.



First design

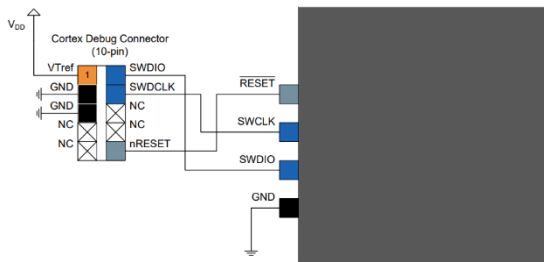


Second design

In our first design, the V_{in} line was directly connected with the V_{in} signal from the battery and when the USB was being used for serial reading or programming simultaneously with the battery as a power supply, it provided back-feeding into the USB port of our computer. To solve this problem, more options were available:

- Mosfet Transistor
- Diode Or-Ing
- Jumper for voltage selector

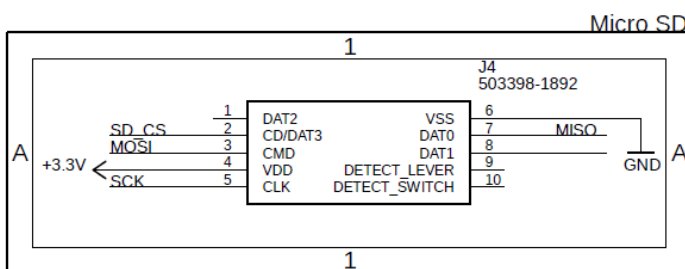
We implemented the diode solutions and more specifically we are using a Schottky Diode for its low voltage drop.



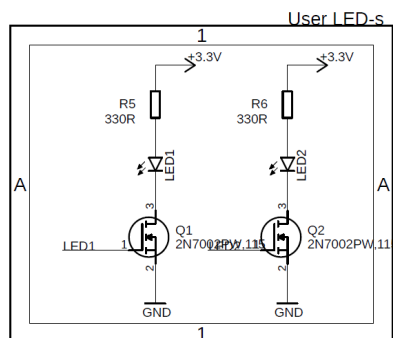
The **Serial Wire Debugger Interface** is being used to upload the bootloader on the chip and for special debugging.

The interface is simple, requiring just 4 pins: GND, V_{Tref} , nRESET, SWDIO and SWDCLK. As

an onboard connector we are using a 5 x 2 1mm pin header, compatible with the programmer cable. The Segger J-link EDU Mini is being used as the programmer.

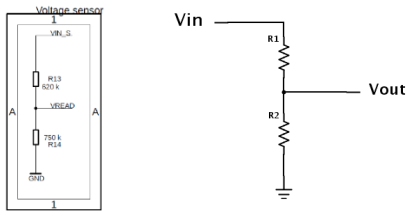


For local storage, we are using a **microSD card** and the communication between it and the uC is quite simple. It requires an SPI interface and power supply.



On the mainboard, two **user LED-s** are mounted, and they proved to be useful for the programming part. We used them to visually indicate the state of some processes our code. The LED-s are controlled using 2 FET transistors in a tiny package – SOT323.

In order to manage the status of the battery, a voltage divider is being used as a **voltage sensor** whose output is read by an ADC capable pin on the uC.

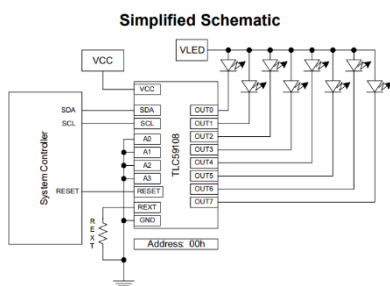


The circuit is widely used in many applications and the working principle is quite simple: the output voltage is divided by a number that follows the formula:

$$V_{out} = V_{in} \times \frac{R_2}{R_1 + R_2}$$

The battery status can be easily checked by the user with an **RGB Led** that indicates a color, according to the voltage of the battery. As we did not have 3 additional available pins which could be used to control the RGB Led, an I2C BUS constant current driver has been used. It has eight constant current output channels, each output being programmable at off, on, programmable LED brightness, programmable group dimming/blinking mixed with individual LED brightness.

In order to work properly it requires an I2C interface, power supply and a reset input pin. We are using a low power RGB Led with SMD mounting pads on the edge of the board.

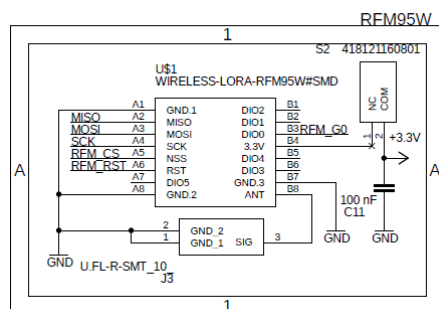
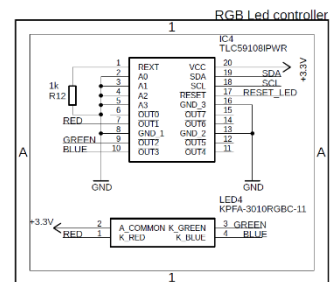


When the board is powered, a LED is turned on for a visual confirmation. For a lower current consumption, we can disable this LED using the jumper connector on the board.

Although in this season the **radio communication** is not mandatory, we integrated a module capable of transmitting and receiving data at 868 Mhz. The module communicates with the uC

via SPI and has an power switch used in case we do not want this type of communication. An uFL connector is being used for an additional antenna meant to increase the range of the module.

The mainboard is connected with other boards, so we are using just a single type of connectors in this project. We chose the **JST SH 1 mm** connectors for the small packages and its quality.



The circuits is powered by a 7.4 V battery. Most of the sensors and the mainboard use 3.3 V as power supply, so we are using a **voltage regulator** that is capable of providing a current up to 800 mAh. The battery is connected to the board via a dedicated connector, then on the V_{in} line a Schottky diode and a fuse are attached. The general power switch is placed before the input



voltage reaches the 3.3 regulator. On both sides of the regulator, tantalum and ceramic capacitors are placed for filtering and bypassing. These bypass capacitors prevent ripple from being amplified as the output voltage is increased. Tantalum capacitors are being used in this specific application because they have low impedance even at high frequencies. The output of the regulator provides power supply to all the components which require 3.3 V.

We are currently facing some heating problems, because the Voltage drop is quite high for our application. We are analyzing the options to solve this problem. We analyzed the behavior of the regulator on different input voltages and the overheating problem is diminished as the voltage drop decreases. At a 5V input voltage the regulator works properly, without overheating. From our tests and the documentation we read, one option is to add another voltage regulator whose purpose is to decrease the input voltage and its output being the input of the 3.3 V regulator. Another option could be a better thermal management on the board. It is quite difficult to add a heatsink on the board, because the regulator comes in a small SMD package, but an accessible solution could be an increased copper area.

According to the datasheet of the regulator, the power the power dissipated by the regulator, P_D can be calculated:

$$P_D = ((V_{IN} - V_{OUT}) \times I_L) + (V_{IN} \times I_G)$$

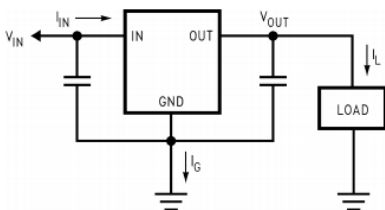
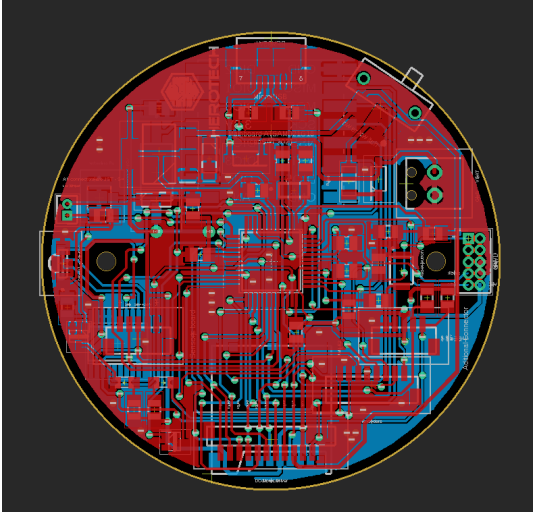


Figure 25. Power Dissipation Diagram

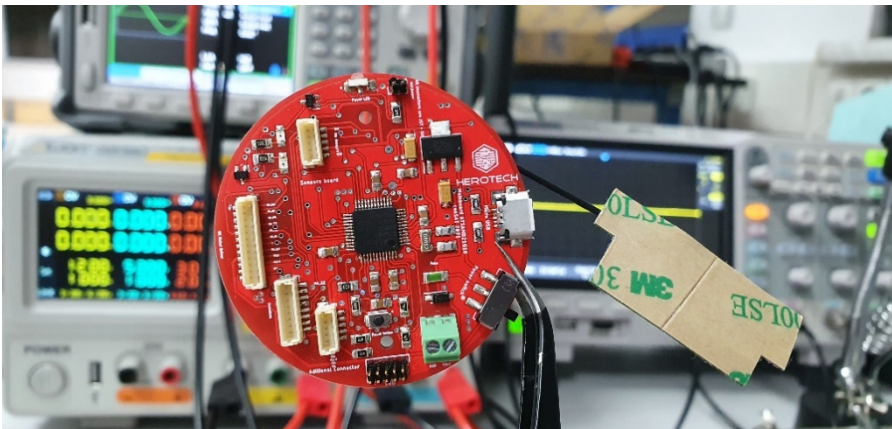
The I_G current is negligible so, from our calculations, the $P_D=1.23$ W. We are currently working on calculations in order to determine the performance characteristics regarding temperature.



Layout of the mainboard:

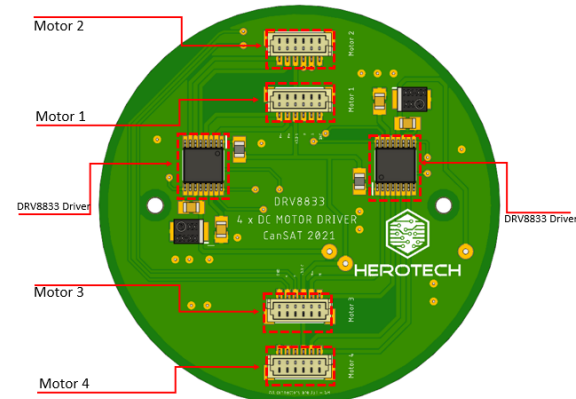
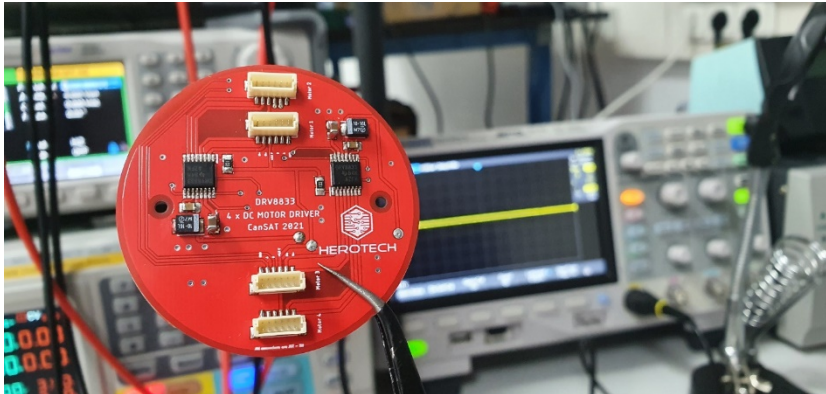


All the PCB-s used in this projects are double sided with the shape of a circle with 50 mm diameter. For mounting, we use 2 2mm diameter holes. All of the parts that have to be accessible by the user have been placed on the edge of the board. All the traces have been calculated for the appropriate current. The calculations have been made using specialized software. On both sides of the PCB, ground planes are integrated. The layout was designed in Autodesk Eagle, taking in consideration the recommendations available in the datasheets of the components. One crucial aspect in designing the mainboard was placing the bypass capacitors as close as possible to the uC



Populated mainboard

2. The DC motors Driver Board



We are using two DRV8833 Dual H-Bridge Motor Drivers in TSSOP package. This driver is very simple to use, and our team already has experience with it. It has a wide power supply voltage range (2.7 V-10.8 V), PWM winding current regulation and current limiting. Also, it comes in a thermally enhanced SMD package that does not require external heatsink.

It provides a dual bridge motor driver solution, being able to drive two DC brush motors or a stepper motor. In our application, it is used to drive 2 DC micro motors from Pololu 210:1. These motors are reliable because they have a low current consumption (max 650 mA) and a 6V operating voltage. The motors are dual shaft, so on one of them we mounted quadratic encoders in order to control them precisely.



On the board, we are using 2 drivers in order to control four motors which are directly connected to the board via JST pin headers. The encoders are mounted on the motor shafts and the connection to the microcontroller is done using the same connector used for controlling the motors.



3. The sensors board

Initially we started from the idea that every sensor should be easily removed off the board in case of damage and replaced with other one.

First iteration

For the temperature and pressure sensor we use the MPL3115A1 from NXP, BNO055 from BOSCH as an IMU and an u-Blox SAM M8Q as a GPS.



The IMU and the temperature and pressure are designed in a small PCB-s with castellated holes on the edges. These are mounted on the board that contains the GPS module. The schematic for these two small boards are really simple, they require just a few capacitors for filtering and some pull-up resistors. The temperature sensor was easily soldered on the board using a hot air station, but the BNO055 was quite difficult to solder because it comes in a small package and after the soldering, we noticed few short circuits between the pads under the chip.

Although we managed to solder it properly, we did not have the proper equipment for the soldering process, so we decided to use another IMU in the next iteration of the sensors board.

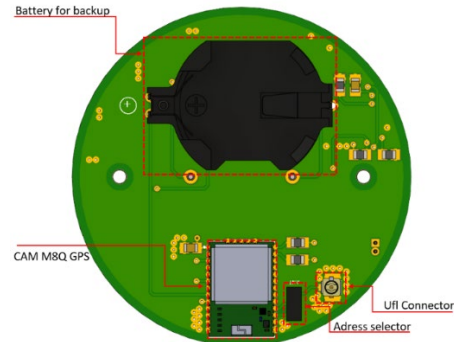
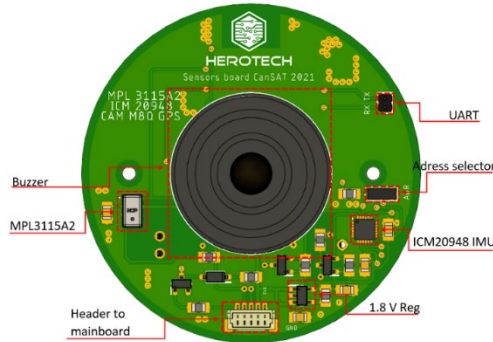
Unfortunately, we did not manage to solder the GPS module properly because it has pins on a big area and the internal components were damaged during the process.

Second iteration

We developed a new sensors board with another GPS module. The temperature & pressure and the IMU remained the same as in the previous concept. As for the GPS module, we used the CAM M8Q from u-Blox. Because we wanted to accelerate the process of making our new version, we prototyped the PCB in house using photosensitive PCB and UV light for curing. After the UV exposure, the PCB was introduced in a special pcb developer sodium hydroxide solution and then in Ferric Chloride (FeCl_3) for etching. Unfortunately, we did not take in considerations the antenna requirements, so the module was not able to view any satellites.



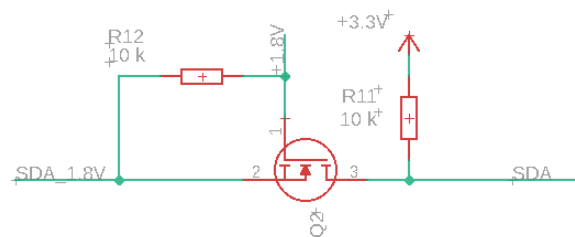
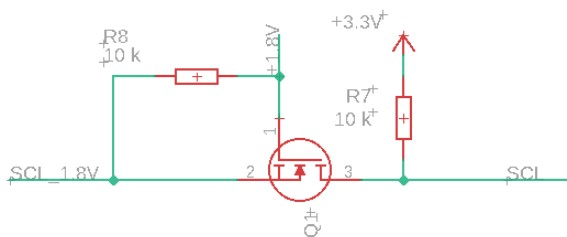
Third iteration



This version came with a proper layout for the GPS module and another IMU. We chose to replace the BNO055 IMU with ICM 20948 for soldering purposes. The only issue was the operating voltage of this IMU: it works with 1.8 V, fact that forced us to use a 1.8V regulator and a logic level converter for the I2C interface.

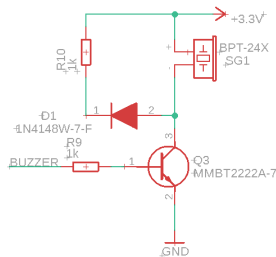
MCP1824 1.8 Vreg is used as a regulator and it provides high current (300 mA), short circuit current limiting and overtemperature protection.

The logic level shifting is achieved by 2 N-Channel MOSFET Transistors (BSS138LT7G) placed on SCL and SDA lines. They come in a small package and the threshold voltage is ideal for our application (0.85 V – 1.5V).





Also, in this version we integrated a battery for GPS backup. In each iteration the buzzer circuit remains unchanged. This is showcased in the picture below:



Basically, it is just a transistor that acts like a switch, opening the circuit when on base is applied a logic high.

Fourth Iteration

A fourth iteration of the sensors board was necessary since we were not happy with the result of the CAM-M8Q's output of data. Inside, the GPS would not be able to receive data from satellites, and when tested outside it would only work sometimes. We decided to test another GPS that we had in the laboratory: the u-Blox Neo-6M. This GPS offered much better results since it had a much bigger and specialized antenna. We ended using this GPS since it satisfied our needs in term of accuracy and signal strength, but we faced difficulties while connecting it to our mainboard since we didn't have enough free pins. To have the GPS communicate through UART, the board needed to have 2 free pins on the same SERCOM. This was made possible by giving up a led and a motor encoder.

Soldering and testing

All the soldering and testing process have been done in an ESD safe environment, in order to prevent damage of sensible components. After we soldered every PCB, we connected them together using proper connectors for the JST SH 1mm pin headers. Although it required special attention, we crimped our own wires using an Engineer PAD 11 crimper tool.

After we made sure each sensor works and sends proper data, we tested the system by connecting it to a lab power supply with current analyzer and having the board request data from each sensor at a small interval of time. The results were close to those calculated on paper before, about 375 mA. After studying its behavior and confirming that the system is stable, we tested powering the CanSat with a LiPo battery with a capacity of 800mA and an output of 7.4V. With the battery fully charged and every sensor and motor in use the system stayed online for about 2.1 hours. On power saver mode, when the system only handled data from the IMU, a battery lifetime of about 10 hours was registered.

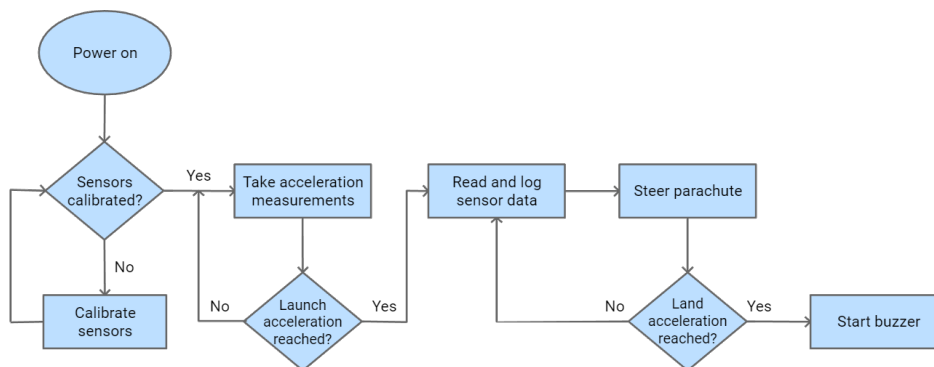


Power Consumption

Component	Max Current Consumption when in use (mA)
ATSAMD21G18 + voltage regulator + peripherals	30
MPL3115A2	10
ICM 20948	5
Buzzer	5
User LED-s	20
RGB Led + driver	25
4 x DC Motors + drivers	250
GPS Neo 6 m	30
Total	375

2.4 Software design

Our CanSats main board controller uses an ATSAMD21G18 chip. We burned a bootloader from Adafruit to it so that it could act as an Adafruit feather m0 board. This was possible because feather m0 uses the same ATSAMD21G18 as a processor. This helped us by opening the possibility to use an Arduino as an IDE to help write, manage and upload C++ code to the board. Another advantage is the already defined configuration for the pins of the board, meaning that we could still use software interfaces such as Serial without having to change the registers or pins configuration.



The CanSat will have 3 phases corresponding to the state of the mission: before launching, launch phase and after landing. To automatically switch between the phases, we are using a state machine system.



Phase 1 – Before Launch. The CanSat will boot up the program at starting. It will calibrate the sensors and make the RGB led green if everything is working properly and the CanSat is ready to start its mission. A blue light is used to indicate a partial initialization, for example if a sensor fails due to different reasons. This is not ideal but the CanSat will be able to complete its mission partly. It is recommended to reboot from the switch if a blue light is showing. A red led is used when a fatal error occurs. This means that the CanSat will not be able to control its descent, or it will not be able to store data collected during it. This phase is working in a low power mode, reading data only from the accelerometer to determine when to switch to the next phase. We are averaging from multiple readings in order to validate the data and not switch to the next phase if the sensor outputs an erroneous value.

Phase 2 – Launch phase. This phase starts when the CanSat detects getting dropped out of the plane/helicopter. At this point the main board will read data from all sensors and will save them to the SD card at a rate of 200 MHz. The read data consists of relative time since start, acceleration in the Z axis, pressure, altitude, temperature, longitude, latitude, battery voltage. At the same time, the yaw angle of our CanSat is calculated from the readings of the ICM20948 sensor. The CanSat will always move in the direction of the parachute's forward vector, so by knowing the yaw angle we also know in which direction the CanSat is heading. We calculate the direction in which the CanSat should be heading by subtracting the landing target coordinates from the current position coordinates and evaluating the arctangent of their division. For turning the CanSat to the desired heading we send commands to the motors in order to retract the ropes. The side where we retract the ropes will act as a pivot for the spinning motion, and the heading will be adjusted, moving the CanSat toward the targeted landing spot.

Phase 3 – After Landing. The CanSat will detect the landing with the help of the accelerometer, using the same concept as the for the launch. After that it will record the GPS position in order to compare it to the targets landing spot and will start the buzzer. It will disable all sensor readings, since the mission has ended and data collection is no longer needed, and since power should be conserved for the battery to last until the CanSat is recovered.

MPL31152A2

This sensor is used to read the air pressure with the help of Adafruit MPL31152A2 library by getting the averages of more sensor's readings. The output in Pa is used to calculate the altitude in meters with the following formula:

$$h = \left(1 - \frac{p}{p_0}\right)^{0.1902632} * 44330.77$$

h = altitude, p = current pressure, p₀ = pressure at ground level

TLC59108

We did not find a library for this RGB led that is compatible with Arduino, so we wrote our own by consulting the datasheet. In our program we define an object of type TLC59108 to which we pass the reset pin in the constructor. We defined different methods for this object such as *setColor(r, g, b)* to which we pass 3 values from 0 to 255, each representing the intensity of the red, green and blue channels.



ICM20948

To read data from this IMU we are using the Sparkfun 9DoF IMU Breakout Arduino Library. By enabling the Digital Motion Processor, the library is able to output the orientation calculated from the fusion of the sensor data from the 3-axis gyroscope, 3-axis accelerometer and 3-axis compass in either Euler or quaternion format. For calculating the compass heading we could technically only use the magnetometer, but the output value would change drastically with a change in roll and pitch of the CanSat. We have to make up for that by plugging those angles in our equations:

$$x_{horizontal} = x * \cos(p) + y * \sin(r) * \sin(p) - z * \cos(r) * \sin(p)$$

$$y_{horizontal} = y * \cos(r) + z * \sin(r)$$

$$\theta = \text{atan}\left(\frac{y_{horizontal}}{x_{horizontal}}\right)$$

x, y, z= magnetometer readings in μT

p, r = pitch and roll angles

θ = compass heading

u-Blox Neo-6M

This GPS module provides the global coordinates, UTC time and temperature with the help of the TinyGPS++ library. On a cold start the GPS might take a few minutes to acquire the data broadcasted by the satellites in order to calculate a fix.

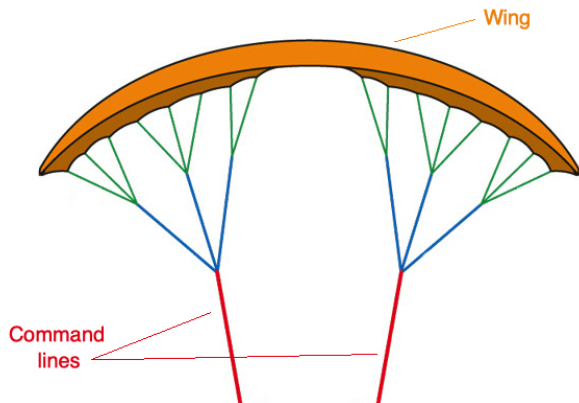
Dc Motors

Dc motor control is done via 2 digital pins per each motor to be able to rotate the motor clockwise or anticlockwise. Each of the 4 quadrature encoders help us determine the absolute rotation of a motor since program start-up by being able to identify rotation counts in both directions. So instead of relying on a feed forward solution in order to actuate the motors, we have a closed loop solution which should be able to generate better results for our descent control system.

2.5 Recovery system

To recover our CanSat after deployment we decided to use a paraglider, which plays a big part of our secondary mission, where we are attempting a directed landing.

Using the command lines of the paraglider which are connected to the Parachute control system, we are going to steer the CanSat. It will be attached to the CanSat through the 3D printed top in multiple points (in dedicated holes, around the motor axes) to make sure they are secure. The paraglider will be made from a light, impermeable material.



The steering is done as follows:

- By pulling on the left line, the parachute steers to the left
- By pulling on the right line, the parachute steers to the right
- By pulling on both lines at the same time we can decrease the drag coefficient, increasing the velocity of descend.

Using the Drag Coefficient formula found in *“Introduction to Flight (John Anderson, 8)”* and considering:

- $q = \text{dynamic pressure} = \frac{\rho V^2}{2}$
- $\rho = 1.225 \text{ kg/m}^3$ from the standard atmosphere definition
- $V = \text{the target velocity} = 8 \text{ m/s}$
- $C_d = \text{the drag coefficient of the paraglider taken from the T10_Parachute_Design from E.S.A. as } 0.75\text{-}1.10, \text{ averaged at } 0.9$
- $D = \text{drag equaling the weight of the CanSat at } 0 \text{ acceleration} = 3 \text{ newtons}$

$$C_d = \frac{D}{qA} \Rightarrow A = \frac{2D}{C_d V^2 \rho}$$

We calculated the necessary area of the parachute, getting the value of approximately 0.085 square meters to reach the target velocity of 8 meters per second, with an approximate flight time of 150 seconds.

With this in mind we started designing the parachute, which proved to be an incredibly difficult task due to the complex airfoil shape of the paraglider, and the necessity to be made of multiple cells. At the moment of writing, we decided that the first parachute we designed in Fusion 360 is a bad idea and we are waiting for a toy kite shaped like a paraglider to test.

2.6 Ground support Equipment

The ground equipment is composed of a raspberry pi 3 with a RFM95W connected to it for receiving the data from the CanSat on the 868 MHz frequency. An antenna is used to increase the reception signal strength. The raspberry transmits data to an ASUS ZenBook laptop through USB connection. On the laptop a flutter web app is running who will draw graphs in real time as the data is being sent. We use the radio transmission for testing purposes on multiple succeeding launches in order to easily debug problems (ex. wrong sensor readings, inaccuracy of the descent control system etc.)



3 PROJECT PLANNING

3.1 Resource estimation

3.1.1 Budget

Product	Qty	Price (EUR)
ATSAMD21G18	1	4
ABS07-120-32.768KHZ-T Cystal	1	1.5
JST Headers and Connectors	NA	30
Segger J-Link Programmer	1	25
Buttons & Switches	3	6
Micro USB Connector	1	2
Micro SD Connector	1	3
RFM 95W	2	50
uFL connectors	2	2
TLC59108IPWR	1	2
User LED-s	2	1
RGB LED	1	1
LM117 3.3 V regulator	1	1
Capacitors and Resistors	NA	5
Diodes	3	1
Transistors	5	2
Neo 6 M GPS	1	15
ICM 20948	1	8
MPL3115A2	1	6
Buzzer	1	4
1.8 V regulator	1	1
DRV8833 driver	2	5
Sensors Board Prototyping	1	1
Mainboard Prototyping	1	1
DC Motor Driver Board Prototyping	1	1
Shipping and Taxes - PCB Prototyping	1	20
Shipping - Electronics Components	3	50
Battery	1	10
LiPo Charger	1	20
Filament	NA	10
Wires and terminals	NA	5
Parachute	1	15
Mechanical Parts	NA	10
micro-SD Card	1	5
soldering (flux, paste, solder wire)	NA	3
TOTAL		326.5

3.1.2 External support

The companies that helped us by sponsoring us are: BSW Metals SRL, Easy Industry SRL, Dacoda Soft SRL and Wenglor.

3.2 Test plan

To rigorously test the CanSat we decided to test in 3 stages:

- I. We will drop our CanSat using a drone with a from incremental altitudes up to about 150m to see if the parachute is opening properly and if it is safe to deploy the CanSat from a higher altitude.
- II. The next stage is deploying the CanSat from a Cessna 172 Plane to test the accuracy of the guided landing system from an altitude of about 1 km.
- III. The final stage consists in building a small sorbitol + potassium nitrate powered rocket to test the CanSat in actual conditions, and see if it holds at high accelerations, higher than the ones that can be provided by a plane or drone. We got our information on how to build the rocket from <http://www.nakka-rocketry.net>.



4 OUTREACH PROGRAMME

4.1 Introduction

Our team beneficiaries of a good media coverage locally. The reason for that is because we have operated in our hometown, Sibiu, for a few years now. On a larger scale, Herotech has had the pleasure of working with important mass media companies in Romania such as Antena 1, a Romanian free-to-air television network owned by the Antena TV Group, part of the Intact Media Group. Our approach to publicizing and communicating about our projects are based on the amount of work we do and the events that are ahead of us. In the case in which we have an event or a competitions phase coming up we usually reach out to journalists and mass media companies to increase our visibility. There are also some cases in which journalists reach out to us and propose to make an article or some news about our activities. Beside media coverage our team possesses social media accounts on different platforms such as Facebook (link + nr. imagine), Instagram (link + nr. imagine) or YouTube (link + nr. imagine). We are active on all of them and try to reach as many interested people as possible, of all ages and social backgrounds. The following of Herotech is constantly growing at a steady rate on all media platforms. We try to post content that is not solely focused on technology, even though tech posts make up to 70% of all our posts. Our other posts include illustrations for different holydays, humanitarian acts, sponsorships, and partnerships.

Another method of publicizing is the team's website herotech.ro. using this website, we increase the visibility of our team, publishing information's about our team and members, about our projects and many more.

4.2 Detailed Information

Wenglor post: For the past few years we have been working with a large international company that has a branch in Sibiu called Wenglor. Wenglor develops smart sensors, image processing and safety technology systems for the efficient automation of industrial facilities worldwide. They help us by letting us use a space in their facility to carry out our activity. They recently posted a Facebook post on their official account about us and our work on the CanSat project. Cc: "The guys from Herotech are working hard on the project for the CanSat 2021 contest held by the European Space Agency and the Romanian Space Agency (ROSA). They are preparing a mini satellite integrated in the volume of a 330 ml soda can. 🚀 "We will build a satellite that is launched from an altitude of about 1 km. During the descent, the satellite has to send and process certain specific data such as temperature, pressure or altitude and perform a scientific mission," explains Daniel Marpozan, member of the team." We have integrated temperature, pressure, altitude sensors, a gyroscope and GPS for continuous satellite location," says Daniel. 🙌 Herotech is composed of 6 students from the National College " Octavian Goga " from Sibiu, gold, and silver medalists in 2019, at the Olympics of Geniuses in the USA, coordinated by their Informatics teacher Andreea Demco. We are glad that the space provided by Wenglor in 2019 helped them to continue their projects and also that it helps us to find out first their plans for the future ...Good job, guys! Daniel Marpozan, Codrin Muntean, Alexandru Radac, Bleaja Seby, Visa Daniel, Radu Flesar #innovation #cansat #ESA #wenglor #wenglorro"






5 REQUIREMENTS

Characteristics	Figure
Height of the CanSat[mm]	113
Mass of the CanSat[g]	305
Diameter of the CanSat[mm]	64
Length of the recovery system [mm]	700(adaptable)
Flight time scheduled [s]	150
Calculated descent rate	8m/s
Radio frequency used [MHz]	868 MHz
Power consumption [W]	2.7W
Total cost [€]	326.5

On behalf of the team, I confirm that our CanSat complies with all the requirements established for the 2021 European CanSat competition in the official Guidelines.

Signature, time and date:.....

15.04.2021