CanSat 2023

Critical Design Review (CDR)

Team MakeItReal, Poland

Bednarska Szkoła Realna

Table of Contents

1. CHANGELOG

Electrical Design:

1.1 To correct our onboard computer’s performance we decided to change ATMEGA 328p to RP2040.

1.2 We decided to use the Ra-02 LoRA module for all of our CanSat’s and the ground base.

1.3 We decided to stop using the DS18B20 temperature sensor and focused solely on using BMP280 to measure the pressure and temperature.

1.4 We will not use the FS100A module anymore because we don’t need it.

Mechanical Design:

1.1 After testing the separation system of our cansat, we decided to slightly change the position of the springs and change them to a smaller version.

Communication system:

1.1 We won’t be using the Lora Lostik Device because of its high price and limited capabilities, instead we will use Ra-02 with Raspberry Pico.

1.2 After problems with building our own antenna we decided to buy a pre-build one.

Ground Support Equipment:

1.1 We won’t use an AWR programmer since we redesigned our PCB board and added USB-C.

Recovery System:

1.1 After creating and testing our parachute we decided to change its shape and size.

1.2 Estimated decent rate changed to 6m/s.

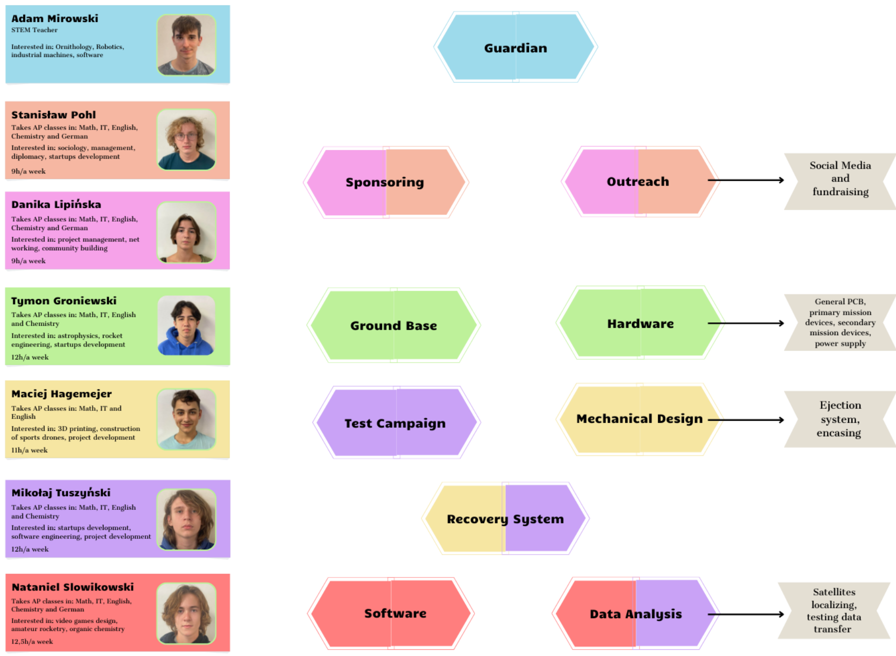
1.3 We have added swivels to the connection system.

1.4 The parachute will be dodecagonal and made of trapezoids.

1.5 We recalculated our drag coefficient value from 0.75 to 1.2.

2. Introduction

2.1 Team organisation and roles



2.2 Mission Objectives

The primary mission of every CanSat is to measure air temperature and atmospheric pressure during its descent and to send those measurements to the ground station every second. The same data will be collected and stored in CanSat's memory on a microSD card, for further analysis. We plan to use the data from the primary mission to tell when our microsats landed, so we can start our secondary mission. Also based on this data we can calculate the height of every micro-sat during their descents. This data put together with data from our secondary mission can also be used to simulate the trajectory of each micro-sat in post-analysis.

The goal of our secondary mission is to build a CanSat that will split into three microsats, which will restart after landing, create a radio network, and then the closest one will connect to the ground base. After establishing two-way communication between the network and the ground base, we will commission it to perform various tasks, e.g., uploading and modifying a file to each micro-sat, and then returning it to the base, or sending temperature and pressure data from each micro-sat using a radio connection only with one of them.

The second mission was inspired by our (humanity's) attempts to colonise other celestial bodies and by our problems with cosmic garbage. Using a communication system on the surface of the planet will assure communication over long distances without sending multiple satellites, requiring only one rocket launch, to place them in the correct places

The mission's purpose is to conduct an experiment that could show the way of building a simple communication web, which could provide communication in a large area, or over big distances without satellites. Such a solution would be cheaper to maintain than a satellite and replacing it after it expires would be less expensive and more ecological. Additionally, this could be an additional way of communicating with earth if one of the hubs had lost its ability to do so.

To ensure the success of the missions, we must achieve what follows:

- The ejection system must work properly and dismantle the microsatellites into self-dependent machines.
- All three of the parachutes must open, and they cannot interrupt each other. All of the microsatellites must survive the descent and the landing and be able to work after landing.
- The microsatellites must be able to implement the web system after landing, connecting to each other and via one of them to the ground base. Executing all of the commands afterward.
- The communication system must be stable, as it is a key part of our experiment, it has to be able to communicate on six channels (3 between the ground base and 3 between the microsatellites) and then on four (1 between the web and the ground base and three inside the web) – firstly to take the measurements from each microsatellite and then to execute the experiment.
- The software must work and execute all assumed algorithms which are: save and transfer the measurements during the flight, disconnect the ground station from the microsatellites, connect the microsatellites and start the web, connect the station to the web via one of the microsatellites, and execute all of the experiments, to prove the connection between them.
- The power supply must be small and lightweight because there must be the same power supply in each of the micro-satellites. Moreover, it has to have a high energy reserve so it will be sufficient for the system for the expected 6 or more hours and it must be proven to be safe to use.

3. CANSAT DESCRIPTION

- 3.1 Mission Overview

Our CanSat will be prepared to fall from an altitude between 500 to 3000 metres from a drone or a rocket. As soon as the CanSat starts descending 3 parachutes will deploy. This will allow our CanSat to split into 3 microsatellites. Each is capable of collecting, sending, and receiving data. The parachute system will ensure that the falling speed doesn't exceed 12m/s.

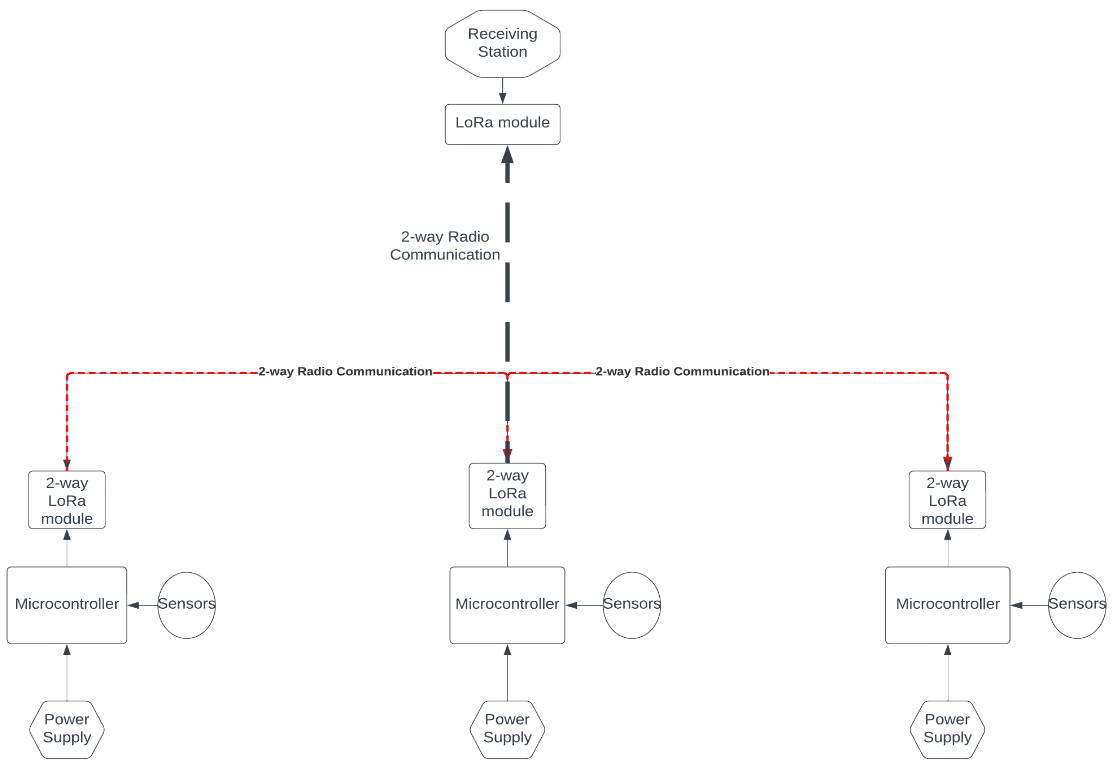
Every Part of the satellite will have an onboard data collection system which consists of a pressure and temperature sensor. Our microcontroller will be responsible for collecting data from the sensors, recording them on an SD card, and sending them by the radio module.

Every CanSat needs to land slowly to assure the safety of the mission. After landing, communication between the microsatellites will start. This will allow us to create a small data network capable of collecting data from many places and then sending them to the one that's connected to the base. The system is optimised to collect and transfer data every second.

The base transmitter will stop the connection with the transmitters and switch to network mode. This will be achieved by sending a ping signal and the first part to receive it (the closest one) will start a 2-way communication with the part that's closest to the base. This will be possible thanks to the directional antenna at the station and onboard LoRa modules. In a situation where it is not possible to choose the closest one it will be decided by the ground base.

Using the collected data we will be able to draw conclusions about the environment we are testing.

Additionally to what was said above, we plan to calculate the signal delay of the satellite that is closest to the base. It will allow us to know the approximate distance from the microsatellite without using GPS. We know that it is physically possible but at this stage we don't have sufficient resources to achieve it.



- 3.2 Mechanical/structural design

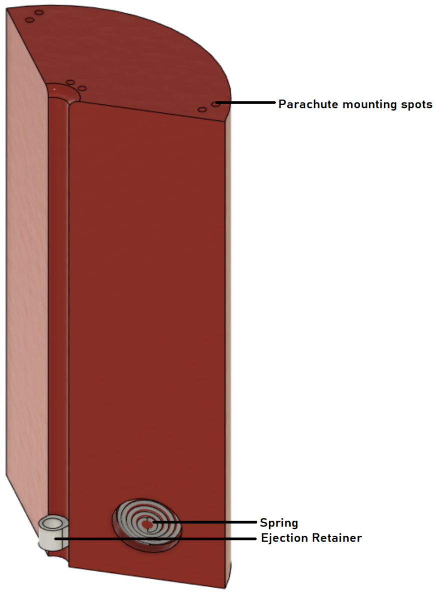


Figure 1 *Figure 2*

Figure 1 presents one of the three microsatellites that our CanSat will split into. The can will separate into three parts fully mechanically without any electronic components. To accomplish that we settled on using three springs which will separate the can and help to get some distance between them.

Figure 2 presents an updated version of CanSat shell. When we printed the first version of our shell we noticed that there were some problems with its design. The holes for mounting the ejection retainer turned out to be too small and the parachute mounting spots were in the wrong places. Therefore we changed the design to match our needs.

- *3.2.1 Electronics Slide*

<- Figure 4 *Figure 3*

The Electronic Slide will slide through an opening in the bottom of the can and will be used to mount all of the components. That way we will be able to quickly access our microcontroller and we will be able to swap batteries.

Since we knew that the PCB and battery should be attached to the slide. We decided to re-project the slide. (*Figure 4*) That concluded in adding a battery mount and microcontroller screw holes. The screw holes can't be seen in this picture because they are too small to stand out enough.

- 3.2.2 Ejection mechanism

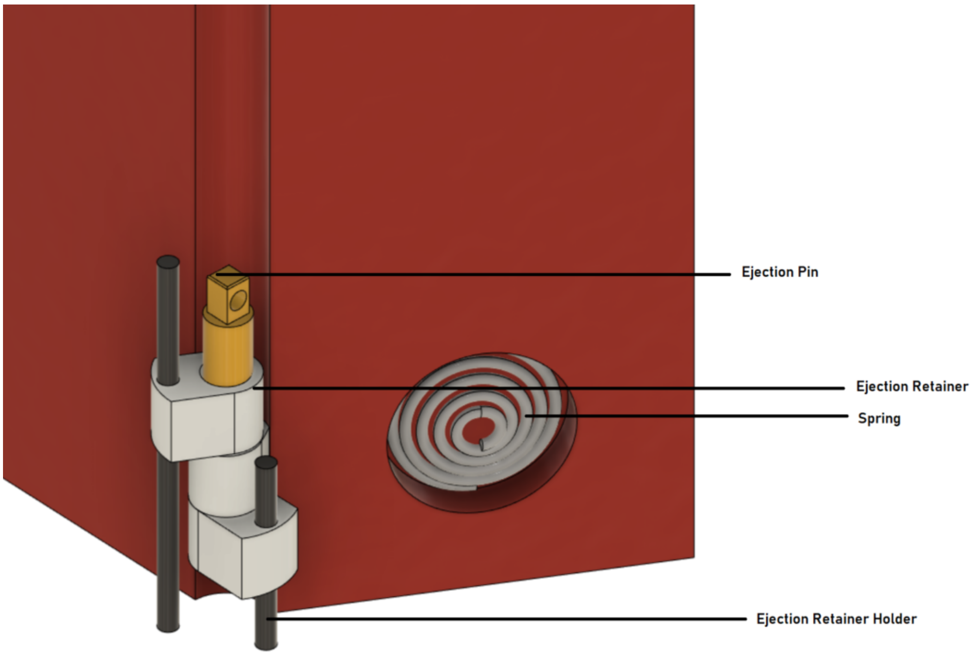
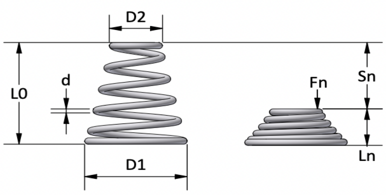
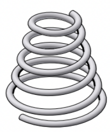


Figure 5

Figure 5 presents our ejection retention mechanism. One of our three parachutes will have one extra shorter string which will get pulled on the moment that the parachute deploys. When that moment occurs, the ejection pin will come out and release the springs. We were worried that the pin might get tangled in the parachute lines, so we decided it would stop on the last Ejection Retainer. In addition, to assure the proper work of our design the Ejection mechanism will use a low-friction material for the Ejection retainer. We settled on a material called iglidur (https://www.igus.pl/product/12404?artNr=I150-PF-0175-0750). It is a material made by IGUS. Igludur is easy to process while being strong and most importantly low friction. The Ejection Retainer Holders' purpose is to hold the Ejection retainer inside of our CanSat. It will most likely be made out of metal. After testing we found out that this design does not fit our previous expectations. Because of that, we have already come up with a different mechanism and we are starting the design process. It may use a servo to release the spring but we have to test it and see if it's achievable. We will explain how it works in the next stage.

- 3.2.3 Spring





-

D1-12.19 mm
D2-3.96mm
d-0.81mm

L0-15.88mm
Fn-max load in LN- 27.27 N
SN spring travel-14.26mm
Ln-length under max load-1.62mm
The spring will be used to separate our CanSat into three parts. The recoil will help us to get a satisfying distance between the microsatellites.

Source(https://www.sodemann-sprezyny.pl/a480-032-062)

We decided not to use this spring for two main reasons. The spring turned out to be too expensive and It would take more than a month to get delivered. The second reason is that this spring would be too strong for our model. When we saw the delivery time we went to AliExpress and found a similar spring. When it got delivered and we put it in, we could not get the can to separate.

After that, we bought similar but smaller springs.

Spring that we ended up using.

This spring is smaller than the one that we tried to use previously. With that, there was a lot less pressure on the ejection pin. Because of that, the pin could slide out easily when tested.

- 3.2.4 Outer Shell

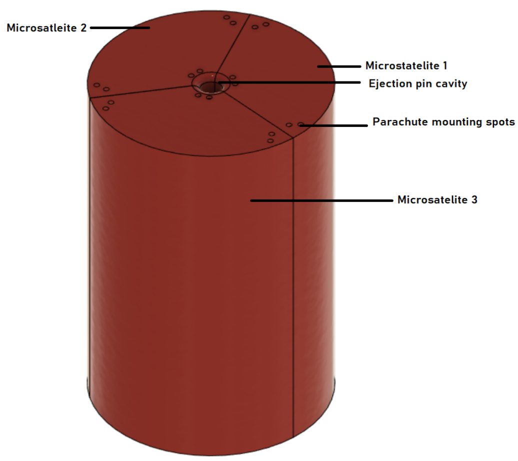


Figure 6 *Figure 7*

Figure 6: presents the outer shell. Its purpose is to contain our ejection system and electronic systems.

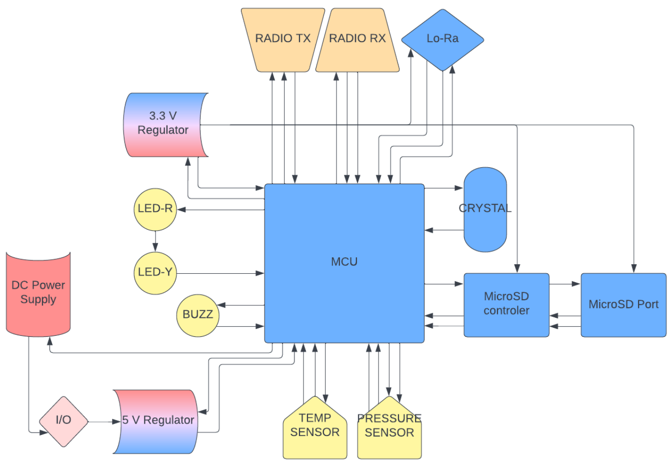
Figure : Here we can see our fully printed CanSat shell. It looks almost identical to our first design. There are some minor changes, making the hole in the middle larger and moving the springs to a different position.

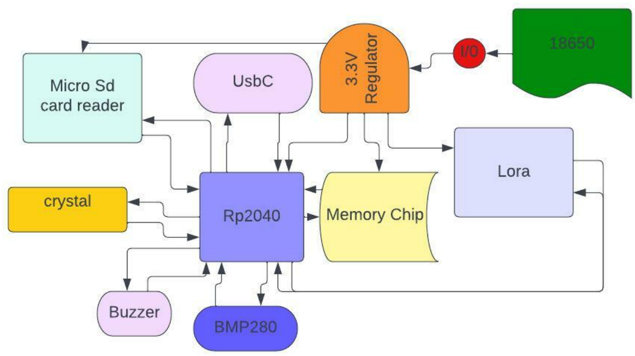
- 3.3 Electrical design

- 3.3.1 General architecture

Since our CanSat will be divided into three smaller ones, we need to develop our PCB which will fit in a microsatellite and will contain all the systems that need to be inside a full-sized CanSat. Therefore the PCB will include our MCU, crystal, memory saving system (microSD), power supply control system, and primary communication system.

Here you can see the simplified diagram of our system. The colours represent their purpose and location: Blue parts - main PCB. Red parts - power circuit. Yellow parts outside of the main PCB due to the sensors and LED (to fulfil their purpose) requirement to contact the atmosphere. The Orange parts - locating the micro-sats - outside of the main PCB.

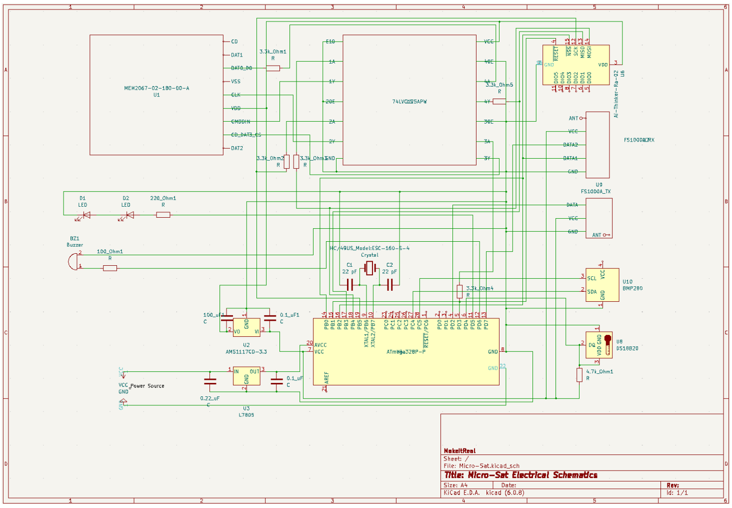




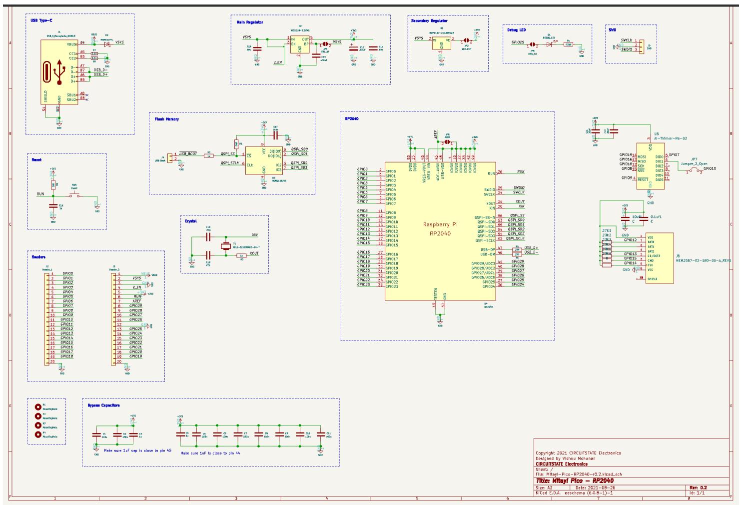
Old Simplified Diagram *New Simplified Diagram*

The electrical design faced a lot of changes. The first and most important one was to replace atmega 328p with a newer Raspberry Pi rp2040 processor. During more specific research we found out that the rp2040 is faster, more reliable, cheaper, and can run on 3.3v. This is necessary since we won't be able to fit a 5v power source inside our CanSat. We decided not to use the fs1000 RX and TX because we can calculate the location of the transmitters using triangulation and our Lora module.

The next diagram is a complete schematic of the electrical circuit:



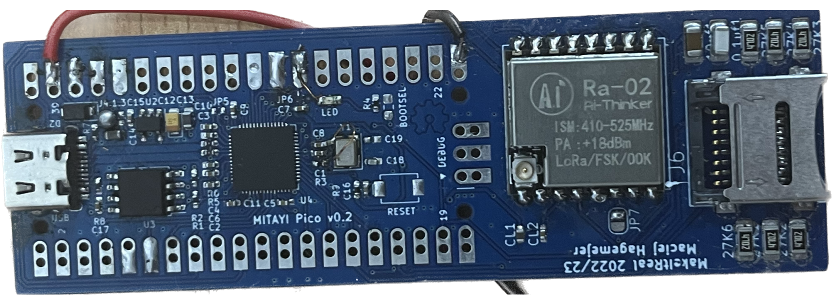
Old Schematic



New Schematic

The PCB was Based On a design that we found on the internet ”MITAYI Pico RP2040 r0.2“. We modified the schematic and design. We added Ai-Thinker-Ra02 and a micro sd card reader to it. When the PCBs arrived it turned out that there was a design flaw, the lines for the crystal were hooked up to the wrong pads. But we cut the trace and directly soldered a short wire to bypass it.

- ***3.3.2 Main system PCB***



Pcb V1

We needed the PCB because we have to work with less space and fit 3 electrical circuits into one Micro-sat. Without the boards it would be challenging to fit all of them. While it is technically possible, it would look rather messy inside the CanSat which would make replacing anything a very difficult task. Additionally lack of space for so many components could cause damage during landing. We placed the components on solder paste that we spread on the PCB using an SMT stencil and we soldered the PCBs using Hot Air. To program the board we are using Arduino IDE with a raspberry pi pico core. This revision of the PCB has a couple mistakes, the 12mhz clock is not wired correctly. So we need to cut one of the traces and solder a short jumper wire to the right pad. But in the next version of the PCB all of these errors will be fixed. The new PCB has already been ordered. We will show the changes in the next stage.

System MCU: Raspberry Pi rp2040

- Dual-core Arm Cortex-M0+ processor, flexible clock running up to 133 MHz
- 264kB on-chip SRAM
- 2 × UART, 2 × SPI controllers, 2 × I2C controllers, 16 × PWM channels
- 1 × USB 1.1 controller and PHY, with host and device support
- 8 × Programmable I/O (PIO) state machines for custom peripheral support
- Supported input power 1.8–5.5V DC
- Operating temperature -40°C to +85°C
- Drag-and-drop programming using mass storage over USB
- Low-power sleep and dormant modes
- Accurate on-chip clock
- Temperature sensor
- Accelerated integer and floating-point libraries on-chip
- Price per piece: 1.6EUR

MicroSD reader chip: MEM2067-02-180-00-A

This particular model is compatible with the chosen logic level shifter chip and is a reliable product.
Type: 8 contact, temperature operating rate: -40 °C ~ +85 °C

Link: MEM2067-02-180-00-A Price: €1.24

Additional peripherals:

Voltage Regulators:

*AMS1117***:** V_{in}: <15 V V_{Out}: 3.3 V, Current: 1 A, Operating: -50°C ~ 125°C Link: AMS1117 Price: €0,52
L7805CD2T-TR: V_{in}: 7-30 V V_{Out}: 5 V, Current: 1.5 A, Operating: 0°C ~ 125°C Link: L7805CD2T-TR Price: €0,36

Other components:

Antennas: Generic Lora Antenas

Various 0402,0608,1206 smd resistors and capacitors.

3.3v converter

- 3.3.3 Primary mission devices

To complete the primary mission we are planning on using:

Fermion: BMP280 Digital Pressure Sensor - as our pressure and temperature sensor

Operating Voltage: 3.3V, Current consumption: 2.7 mA, Pressure Measuring Range: 300~1100hPa, Absolute Accuracy: ±1hPa, Temperature Measuring Range: -40 to 85°C, Size: 18x11.5mm, Communication Interface: I2C

We will use this sensor to measure the pressure and temperature, firstly for further analysis and after landing, it will be part of our experiment of building a communication web.

We chose this sensor because of its high accuracy, long term stability, small size, and low power consumption.

Link: BMP280 Price: €7.12

- 3.3.4 Secondary mission devices

Because the goal of our project is to execute experiments on our peer 2 peer web and locate our Micro-Sats, the devices for the secondary mission are also used for our communication system.

Ai-Thinker Ra-02

SPECIFICATION

Supply voltage and work: 3.3 v, Work temperature: -40 to + 80 C, Range: up to 15km, Transmission speed: up to 300kbps, Transmitter power: +20dBm - 100mW, Working frequency: 433MHz, High sensitivity: -148dBm, Communication method: LoRA, Communication protocol: SPI
The Ra-02 module guarantees that every one of our micro-satellites will be in the range of the signal, which is necessary to complete our secondary mission, in addition, it meets the requirements of frequency for CanSat's and it has a good price. Link: Ra-02 Price: €8,19

- 3.3.5 *Power supply*

We decided on using 18650 Li-on batteries. They are small and reliable which are the most important factors in deciding on our power supply. Mounting is described in mechanical design. We choose PANASONIC NCR18650B 3400mAh Li-ION 3,7V 18,6x65,2mm 4.87A as our model. After testing we figured out the capacity greatly exceeds our needs but we decided not to change it, because additional power may be used as a safety buffer.

- 3.4 Communication system

Radio waves from the base will be only directed to one of three Micro-Sats. The satellite which receives the signal first (It will be the closest one) will be marked as "Main Satellite" in the software. The main satellite will send signals to other parts of the satellite and create a full radio communication net. If the software cannot decide which satellite should be marked as the "Main Satellite", it will be chosen randomly.

- Maximum power will not exceed 20 dBm to fulfil the requirements in the rules.
- We will use a LoRa module for radio communication.
- Communication will be working on the 433MHz public channel/frequency.

All 3 Micro-Sats will be equipped with an elastic antenna which will be directly connected to the parachute. The antenna will straighten up when the parachute opens.

Most of the time our already separated Micro-Sat's will solely reflect the signal and exchange signals with each other. Unlike our base, they will not broadcast new signals.

The first signal which will initiate the radio net process will be directed from the base.

Requirements:

- Directional antenna and well-working software in the ground station. We will also need a receiver to receive signals returning from the "Main Satellite".
- 3 LoRa modules and 3 antennas inside of satellites.

The products shown here are only examples and may change depending on availability. Especially the LoRa device gateway which has many substitutes may change depending on the needs and availability.

These are some examples of our possible choices for Lo–Ra communication devices:

Ai-Thinker Ra02

Ai thinker is an easy-to-use, small LoRa device. It is highly compatible with raspberry pico which will be used as the main computer in all of our CanSat and ground base. Very versatile and simple to code.

Connector SMA male/ SMA female

Length – 10.0m

Price - 6.16€

USB delock cable 5.0m

Flexible USB extension allowing to connect our LoRa device to a computer

USB 2.0 on both endings, socket type (A)

Length – 5.0m

Price = 11.57€

- *3.4.1 Ground station antenna*

For a strong receiving circuit to cover large distances we need to use antennas with good gain.

Antennas that meet our requirements are directional antennas for example Yagi antennas which

are commonly known for their good performance.

- *3.4.2 Satellite radio system*

To calculate the correct length of the transmitting antenna we need to calculate wavelength with the equation -

λ = c*T = c/f. Where λ – states for wavelength.

With that, we have the following variations

(λ) – full-wave antenna, antenna length = 70[cm]

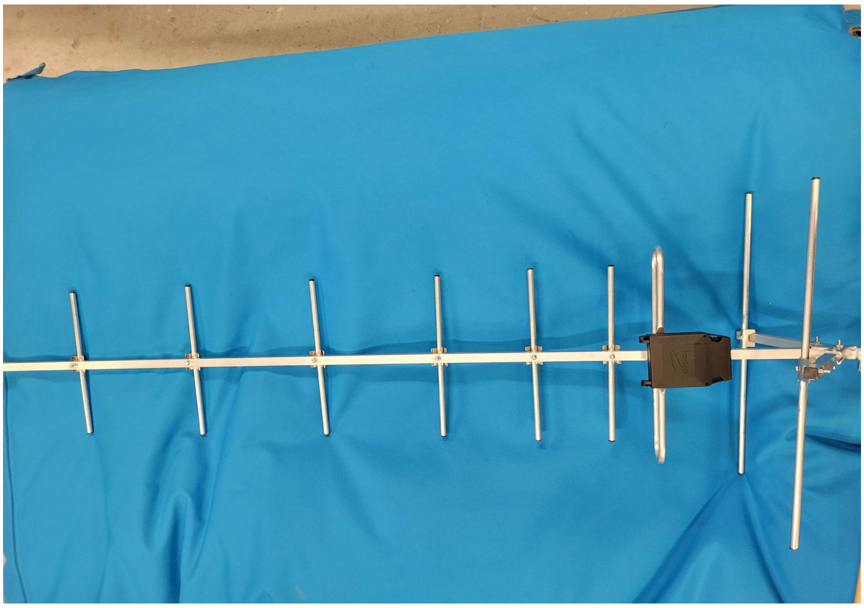
(λ) – half-wave antenna, antenna length = 35[cm]

(λ) – quarter-wave, antenna length = 17,5[cm]

c = speed of light, T = period, f = frequency

Because of the difficulties we faced trying to create the antenna we decided to buy one. For that reason, we assumed that the most reasonable choice would be to buy a full wave antenna for its great range.

The antenna we bought:



3.4.3 Radio triangulation

Radio triangulation is a way of locating the source of the signal by calculating distance and angle from your position to the RF transceiver. To calculate distance and angle we will use two antennas and one RF receiver. By connecting the antenna to the RF receiver and using received signal strength indication (RSSI) generated by the RF receiver, we will be able to measure the angle from which the signal is coming from. By moving and measuring our travel distance and then repeating that action with a second antenna we will be able to calculate the approximate location and angle of the signal transceiver. We will use a magnetic loop antenna and a YAGI antenna to measure the minimum and the maximum amplitude of the received signal. We use two antennas to increase precision of the angle measurement - a loop antenna is highly directional and has a zero in its directional (angular) characteristic in the direction perpendicular to the loop surface. A YAGI antenna directional characteristic has a maximum in the direction of its boom.

The distance to the transmitter can be computed from the formula:

$$d = \frac{b}{\sin \Theta}$$

Where *d* is the distance to the receiver, *b* is the travel distance, and Θ is the difference of the angles measured in first and second positions.

- 3.5 Software design

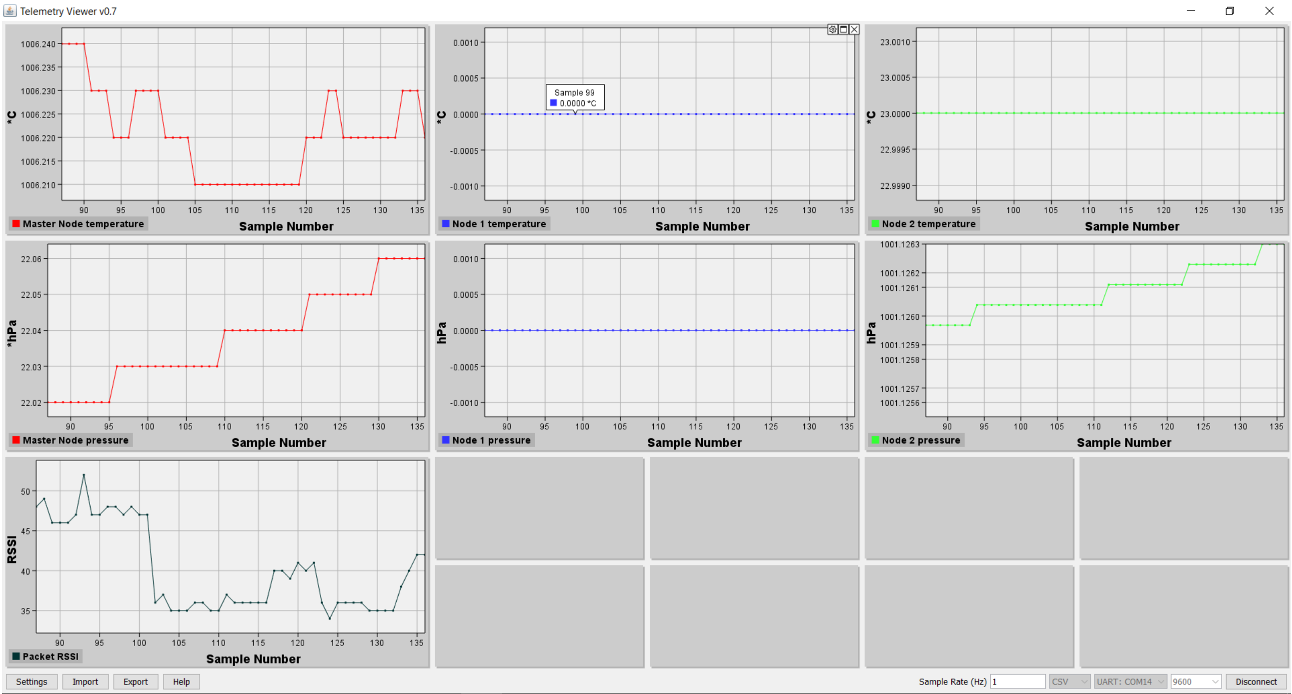
The software designed for the onboard computer and base station will be designed in the Arduino IDE environment, using C++. A program written in Java will be used to visualise the data.

Stage 1: Starts after power on. CanSat is waiting for deployment and a basic test starts.
Stage 2: Starts after deployment. The satellite collects primary mission data and sends them to the base.
Stage 3: Starts after landing. The most important part of our mission. As soon as every part has landed they start searching for each other to create a network. After receiving confirmation that the microsatellites are connected. We send a ping signal from the base to locate the closest part and the first one to receive it begins a 2-way communication with the base. A more detailed description of the ground support transmitter code can be found in the *Ground Support Equipment* section.

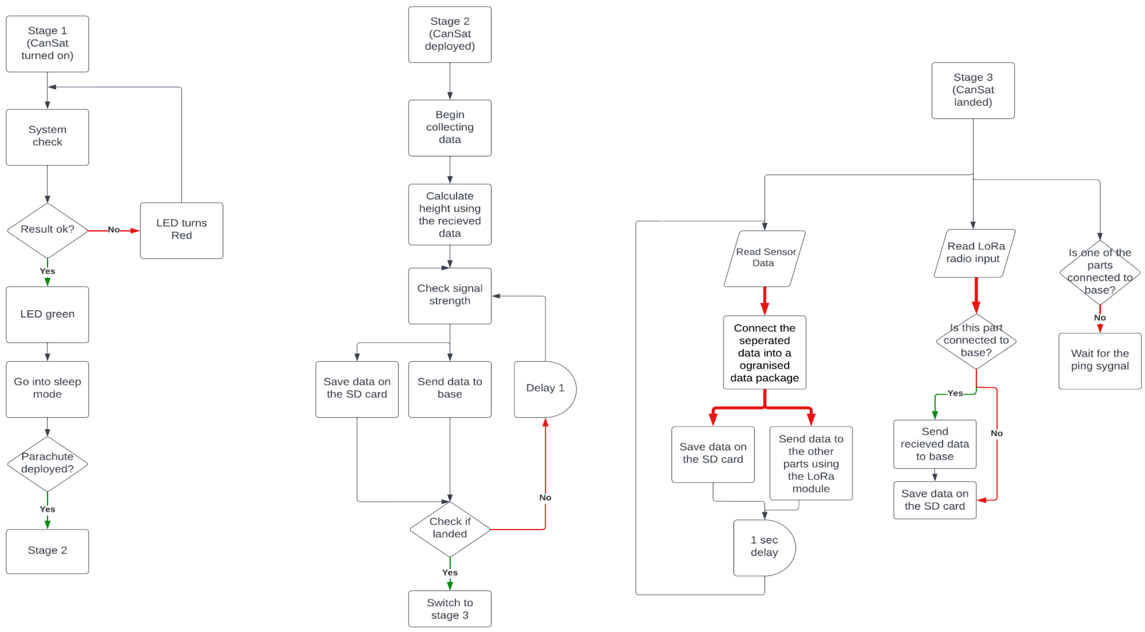
After landing we are planning to calculate the distance using signal delay. We know it is possible but as of today, we have not developed a solution yet.

The amount of data gathered is too small to have an impact on the mission so we decided not to calculate it before the test campaign.

We created a simplified version of the software allowing us to test every design aspect. The biggest difference from the final version is that we have a predefined main node that communicates with the base. But other than that our software works as intended, two slave nodes send telemetric data to the main node and save them to the SD card. At the same time the main node collects data from the two slave ones and itself, compiles them into a CSV data packet, and sends them to the base. We managed to collect the data and analyse it using an Open Source Telemetry Viewer program. Because we decided to keep all of our work open source the code is available on our Git Hub.



The diagram presents a simplified way of how the program works.



- 3.6 Recovery system

To maintain the maximum speed of 12 m/s for our recovery system we decided to use a round-hole parachute. We chose a round-hole parachute because it’s the easiest type of parachute to calculate drag force, which means that we can easily control the maximum falling speed. We still consider ram-air parachutes for drifting away purposes. We are also strongly thinking about using a swivel to deal with possible entanglements. We decided on a terminal velocity equivalent to 6m/s in order not to exceed maximum speed safely. With known terminal velocity we can easily calculate the area and the radius of the parachute. We will also implement the spill hole on the top of the parachute to ensure that the parachute will not tilt sideways. The spill hole will cover 5% of the parachute surface as usual in a round hole parachute. As our satellite will split into three parts we will create three same-looking parachutes to handle maximum speed for all of them. We were inspired by NASA's (Orion parachute system) research and we will try to create a similar system.

The parachute will be made out of Nylon

Here are the calculations and formulas for our round-hole parachute:

(Equation for the area of non-round parachutes) - $S = \frac{n}{4} \times D^2 \times tan(\frac{180°}{n})$

$$S = \frac{2 \times g \times m}{\rho \times Cd \times V^2}$$

$$V = \sqrt{\frac{2 \times g \times m}{\rho \times Cd \times S}}$$

S = 0.024 area of the parachute

$$Rh = \sqrt{\frac{0.05 \times S}{\pi}}$$

Rh = 0.019m Radius of the spill hole

$$Rp = \sqrt{\frac{S}{0.95 \times \pi}}$$

Rp = 0.089m Radius of the parachute (without spill hole)

V = 6m/s Terminal velocity of Cansat (assumption)

m = 0.116 - Cansat mass (Only for one part, not 3), (approximation)

0.75 - Drag coefficient (assumption)

π = 3.1415 – Pi constant (approximation)

g = 9.81m/- Gravity acceleration (approximation)

ρ = 1.225kg/- Air density (approximation)

We've decided to change our shape to a dodecagon with a dodecagonal hole. This allows us to attach ropes at regular distances from one another and assure bigger internal pressure under the parachute. The Idea of possible use of a ram parachute was abandoned due to its needs for resources and difficulties with using it correctly.

To construct a well-working dodecagonal parachute we sew 12 trapezes together and strengthen the edges and places connecting with lines with additional fabric to ensure the safety and reliability of the parachute.

The first approximation of our Drag Coefficient was a commonly used approximation applied for example in a Hollow semi-sphere opposite stream parachute which doesn't apply to our new shape. Therefore we changed it to 0.75 which is correct for our new shape.

We also changed its size to lower our terminal velocity to 6 m/s. This decision was made to assure the safe landing of all of our three micro-sats (we need to stay within our maximum weight while using more components and filament which affects the durability of the micro-sats). The Diameter was changed to approximately 30 cm with an apex hole with a diameter of 6 cm (~4% of the area). Therefore our new dimensions are included in this table:

Area of Parachute:	Diameter of parachute:	Apex hole area:	Diameter of apex hole:	Trapezium height:	Trapezium 1 base length:	Trapezium 2 base length:
715.4 cm2	30.2 cm	27.5 cm2	6 cm	12 cm	2 cm	8 cm

We calculated those values based on an online calculator (link) and the first equation for the area of the dodecagon from its Wikipedia page. Based on that information we were able to determine the appropriate values for our trapeze heights and bases. All of those values are approximations to their first decimal places.

Area of a dodecagon: $3cot\left(\frac{\pi}{12}\right)a^2 \;=\; 3\left(2+\sqrt{3}\right)a^2 \;\simeq\; 11.1961a^2$

Area of a Trapezium: $\frac{(a+b)\times h}{2}$

On this stage of the project we also chose the lengths of the parachute lines (35 cm) and to make stitched loops on the ends of the lines (it will take 5 more cm) because that will prevent the lines from unbinding during the freefall. The diameter of our parachute lines is (1.5 mm).

Additionally, we have decided to add swivels to our parachute. They will be made out of steel rings and assure the connectivity between the parachute and the micro-sats.

The Micro-Sats will be connected to the swivels using smaller diameter lines (1 mm) with stitched loops on one end and glued with cyanoacrylate glue to the Micro-Sats shell. In the next stage of the project we plan to make steel connectors in the shell to abandon the necessity of using glue which is unreliable. Their length will be 16 cm - 5 cm for loops, 1 cm for glueing area, and 10 cm for the middle part.

- 3.7 Ground Support Equipment

Our ground base will consist of these parts:

- Laptop
- Software to analyse data
- Spare parts and batteries
- Receiving and transmitting station:
    - YaGi directional antenna
    - LoRa module
    - Software to capture and send data using the radio module

1. Laptop

Any laptop compatible with our software and the LoRa module.

2. Software

- Data analysis:

This software collects all the telemetry data and then sends them to a visualisation program. This will allow us to analyse the data easily.

- Data receiving and transmitting:

During the microsatellites descent, there will be a one-way communication used just to transfer telemetric data. After landing the connection will switch into a network mode where the base station will have a 2-way communication to the closest micro-sat (that is already connected to the other two). Then using the LoRa module we will also have the ability to send data into the network and not only collect it, which will allow us to execute experiments to prove the proper work of our web.

3. Receiving station
    - Described in Communication System

- 4 TEST CAMPAIGN

- 4.1 Primary mission tests

- *4.1.1 Measurement and data download tests*

Pressure and temperature sensor test: We will use a reference device to measure the pressure in different environments and then compare the result with our sensors. This will let us check the work of sensors and SD card modules to find every possible problem in both the sensor system and data saving system.

- We conducted a test using our BMP280 sensor and compared the results to a home thermometer. The sensor exceeded our expectations but we weren't sure about the thermometer's accuracy. That's why we decided to test it using a better electric temperature sensor. The results were acceptable, we observed that the error margin was around 0.2*C. It is good enough for our needs. The pressure was easier to check and the result was an error margin of around 0.15 hPa

- 4.1.2 Sending/receiving tests

The test will contain multiple trials to check if the connection is stable and reliable enough to send data every second. It will also allow us to check the delay and the speed of data compliance.

- We tested the software used to control the LoRa module, there hasn't been any data corruption so
- it was considered successful, but the delay was negligible.

- 4.1.3 Communication system range test

1. **In an environment similar to the one in which the mission will be completed we will carry out a test to check the maximum range of our transmitter at the height that the mission will be performed**
   - A test was run using our directional antenna and a prototype board was passed. We got a satisfying result of about 2 kilometres. It was a radio-polluted environment so we are planning to repeat the test in a rural territory where we expect the range to be a lot bigger.

2. **A range test while freefall will also be performed**
   - Postponed

- 4.1.4 Energy budget tests

We are planning to use the battery which will be able to provide power for each of our MicroSat's for 4 to 6 hours. To test the battery capacity we will perform a variety of tests including

1. Calculating battery usage in every mode
   - We conducted a test in which we connected the board to an ammeter and measured energy usage in different modes. These are the results:

     30mA - while every component is in standby

     40mA - while LoRa waits for the packet and every component is in standby.

     60-70mA - while sending a packet and saving on an SD card. The jump lasts for a small amount of time.

     So if the Can will wait for launch for about 4 hours in standby mode and the mission will last 2 hours and send a packet will be sent once each second: we will need around: 4 hours in standby = 120mAh, 2 hours receiving = 80mAh, and a maximum of twenty minutes in the most consuming mode = 25 mAh. The calculations show that if everything goes right the mission requires around 225 mAh/per microsatellite.

   - The results surprised us, we thought we needed way more energy. That's why we decided to make another test later where we will just leave the circuit running for a few hours and see how much energy was consumed.
2. Verifying battery capacity under natural environmental conditions
   - Postponed to the next stage
3. Verifying battery capacity in low temperature
   - Postponed

- 4.2 Secondary mission tests

- *4.2.1 Satellite splitting test*

1. We plan to drop the Cansat from a special drone module to test if CanSat is splitting into 3 parts properly. For that, we will have multiple tests on the splitting mechanism. This test is going to help us predict the spring recoil and distance between each part after landing.
   - We performed a test by dropping the Cansat from a drone. It wasn't exactly perfect but we managed to make some observations. The splitting test didn't work as expected. The second test was performed with the parts already disconnected so we can find the distance between them. Unfortunately, we lost the parts due to a flight error.
2. In the same test we will also check the connection between parachutes and the splitting mechanism. To test this we are going to drop down the CanSat multiple times to find the best and worst variants of the system to avoid the possible consequences of entanglement of the parachute ropes.
   - During the cansat splitting test we did not detect any possible entanglements.

- *4.2.2 Data network test*

We will connect all 3 transmitters and a computer to simulate the network. Then we will input information and analyse the way data is managed. If everything turns out to be working right, transmitters should send the signal right away, without any errors. A range test between each part excluding the ground base will be performed.

- The test was successful. There were a few errors at first but after some slight changes, we managed to have a flawless run.
- As for the range test, during the tests performed there weren't any electronics inside the cans so we couldn't get exact results, but the release system as well as the temporary base station had our PCB board to control a lora module with the small antennas we plan to use. During deployment, the drone was pretty high and we still had a good signal. It showed us that most likely the small antennas we bought for each of the three parts will give us enough range.

- 4.2.3 TOF Sensor test

Our research shows that counting distance based on signal delay is possible. But we are not sure if we will be able to make this a feature of our Cansat due to time contraints. To check if it will be possible the testing will consist of using the system to measure distance and then checking it using GPS. It will also help us to define an error margin if one exists.

"t1 + t2 – delay = t0"   "t1 = time of flight from base to the Cansat" "t2= time of flight from the Cansat to the base" "(t1 = t2)" "Delay = time between receiving and transmitting the pulse in the satellite"

*"d = c*0.5(t0 – delay)" "d = distance" " c= speed of light" "t0 =time between transmitting and receiving the pulse in the base unit"*

After checking our schedule we decided to postpone TOF tests because of its huge time requirements and tough research. Additionally coding and choosing good components for the advanced circuit will be extremely hard with our limited space in Cansat. For this time we will stop this project and focus on the rest of the Cansat, maybe soon after completing the rest of the components and requirements, with additional time we will attempt to create this but for this time we will postpone. An additional way is buying a pre-prepared TOF module which can be much more precise than anything that we can make in the limited time. One example of such a module can be found in LoRA Tx1280 with the TOF system already built in. but unfortunately this system works only on 2.4 GHz passage so for the time being we need to search for another alternative.

- 4.2.4 Satellite direction test

In this test, we will try to find the direction of the satellite. For this test, we will use a directional YAGI antenna with significant directional gain. When we will detect maximum radio wave power we will know approximately where the satellite is.

- We tried finding satellite direction by connecting the YAGI antenna to the LoRa module. After uploading RSSI software we were able to detect the direction of the source of the signal. The test was successful and we can approximately establish in which way we should go for finding the transmitter.

4.3 Mechanical system test

- 4.3.1 *Endurance tests*

During landing, any electronic part of the Cansat can't be damaged as it can disrupt the whole mission. We are planning to do crash tests by using a dummy can to measure the impact and check if the materials and electronics won't get damaged if something during the landing goes wrong.

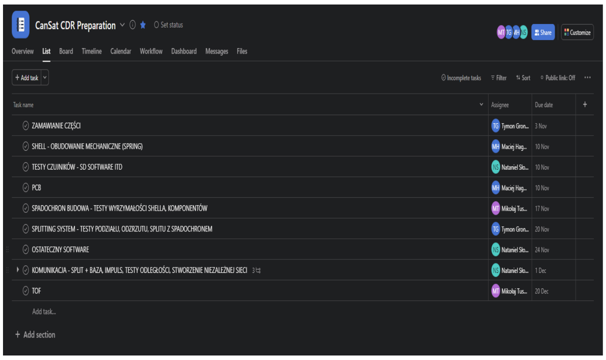
1. Drop-down test: Using a drone or by going to the top of a high building we will drop our satellite from various heights to ensure:
    a. Can a Cansat shell withstand dropping down on the ground with a terminal velocity of 6m/s?
        - Shell survived the test
    b. Can the Cansat components withstand dropping down on the ground with a terminal velocity of 6m/s?
        - Postponed

- 4.3.2 *Recovery system test*

1. Parachute deployment test: using a drone or by going on top of a high building we will drop our satellite from a big height in various weather conditions, to check if these important elements of our mission will be completed:
    - One test was conducted using a drone and our prototype. It wasn't successful, we lost 2 of the 3 parts, but we managed to make some observations for improvement.
    a. Is the descending speed lower or equal to 12 m/s?
        - We couldn't calculate the exact speed because we haven't registered the exact landing moment. But according to our observations, the speed was definitely in the allowed range, also the can was not damaged so we considered the parachute was working.
    b. Is there a possibility of the parachutes tangling in hostile weather conditions?
        - postponed
    c. How will the CanSat behave in bad weather?
        - postponed
    d. What is the spread of microsatellites?
        - It was quite alarming that from what we could see the spread was low. That brought us to think about redesigning the splitting system so it can release the parts faster and maybe spread them for a longer distance.

- 5 Project Planning

- 5.1 Task list and Time Schedule



In a program called Asana, we managed to create a task list that helped us to organise our work while working on this report. (First screenshot)

We also created another schedule that contains dates to accomplish major steps of the next stage of the project. (Second screenshot)

- 5.2 Resource estimation

- *5.2.1 Budget*

Based on our calculations one full prototype will cost about 100 euros. The list is still not perfect, it doesn't include batteries which we have not chosen yet. We tried to include as many PCB parts as possible but we had no way of checking the exact price. Figure 3.1 presents necessary cansat parts and their costs (Described in detail here). This is a minimal amount of money needed for a successful prototype.

Table presents estimated costs of one prototype

Part name:	Price (euro):
Parachute material	2
Parachute lines	1
Custom PCB board	4
PCB components	20
Batteries	7.41 * 3 = 22,23 euro
Onboard Antenna	1
BMP280 Sensor	0,34 * 3 = 1,02 euro
Buzzer	0,40 * 3 = 1,20 euro

Shell (3d printed)	0,5
Release system	1,5
Sum:	54.45

Table presents estimated ground support base cost

Part name:	**Price (euro):**
Directional Antenna	12.74 euro
SMA cable	7.85 euro
Raspberry Pico	6.37 euro
Ra-02 module	7.74 euro
Raspberry Pico	6.37 euro
Sum:	41 euros

- ### *5.2.2 External support*

We started our outreach program by making a Facebook and Instagram page. It will help us gather financial funds and achieve our goal of collecting 1500.00€ and build interest in our project.

We are planning to get funds by finding sponsors. We are looking into mBank's company program for talented young people called "*M jak matematyka*" (M like maths). It is aimed at helping students who are mathematically gifted and want to make maths interesting for everyone. We are also open to more options and finding some additional sponsors who can make achieving our goal easier. We have already prepared draft emails that are ready to be sent to potential sponsors after we qualify for the next stage.

In addition to that, we are planning to make a fundraising campaign that will be promoted in our school community and among our friends and family members. It is going to help us to collect money easily and to start working early.

For the last 2 months, we have been sending emails to potential sponsors, unfortunately, we haven't got the "M jak matematyka'' grant but we got a couple responses from different companies and foundations. In the end, we got a grant from "Mam Pomysł" foundation. That helped us to finish our work with sponsors and secure our budget. We are still collecting emergency funds via crowdfunding.

Additionally we managed to collect 3298 PLN (697 euros) via crowdfunding. It was enough to start our work and build our first prototype.

- 5.3 Outreach Programme

We are planning on posting on our social media once a week while working on the project. This will include making at least once a month a report on our progress and milestones in our work.

In our school, we organised a competition to choose the best logo design for our project. It helped to spread awareness about our participation in the competition.

We are also planning to promote our project on the school's forum to get more people interested in our project.

During the second stage of our project, our outreach program is one of the things we are proudest of, it helped us to develop new skills like public speaking and promote the competition in front of an audience so big we never even thought it was possible.

Our Instagram page is working great, we already reached around 700 accounts, gained 90 followers and approximately 400 likes. We are frequently posting our work updates there and spreading awareness of our project.

Additionally, we made a music video about our team and sent it to a movie festival for young people called "Kameralnie ". We did it to raise awareness about our project and got more people interested in it: Our music video got an award for an interesting idea for the festival.

We spoke about Cansat twice on our School open day to a large audience of candidates and their families (about 350*2 people) It helped to promote our crowdfunding and raise awareness about the competition. The second speech went great. We collected about 950 PLN (about 200 €) in a day.

We were invited for two live radio interviews (one in “Radio dla Ciebie”, and the second one in “Czwórka”). During the interviews, we were presenting our project to a large audience (we were on the air during rush hours: 8 in the morning in “RDC” and 18 in “Czwórka”) and further promoting it. Both Radio stations featured us on the front page of their web pages. It was an educational and exciting experience for us. It helped us to get more comfortable with public speaking

A Part of our team also went to an event organised by ESA where they met the only Polish reserve Astronaut for the European Space Agency and an operator of the Atlas Experiment in CERN - Sławosz Uznański. While talking to him, he gave us some great tips and posted about us on his Instagram page. It was a really interesting event for our team members especially since we are all interested in space and physics.

- 6 CanSat characteristics

Characteristics	Figure
Height of the CanSat	115 mm
Diameter of the CanSat	64 mm
Mass of the CanSat	-
Estimated descent rate	6m/s
Radio transmitter model and frequency band	LoRa RA-02 SX1278 433MHz
Estimated time on battery (primary mission)	4-6 hours
Cost of the CanSat	54.45 euros

This photo is portraying a prototype. It was assembled during the testing phase. The parts are working together well enough for the stage of the project. In the photo there is a parachute, a shell, a slide, the electronics and the cover. We had put it all together and managed to receive data in another LoRa.