Critical Design Review

for **CanSat** competition

Team

SobieskiSat

1. Changelog

- PDR
    - Initially we planned to use an outside communication module with a chip supporting LoRa, but since it is already in CanSat kit provided to us, it was deemed unnecessary and we opted to came with some other ideas for the secondary mission.
    - As of the PDR, our main goal in the secondary mission has changed. Now we are going to collect humidity and video data alongside PM density.
    - In addition to the main goal of the secondary mission we also added plans to develop a descension assistance system (DAS).
- CDR
    - We decided to leave out the camera because we could not find any suitable one for our needs and compatible to our board.
    - We've updated our component list with final components we chose according to our financial capabilities and electrical requirements.
    - Mission stages from PDR were modified due to software development.
    - DAS (Descension Assistance System) was delayed due to prototyping restrictions and ultimately its goal was changed to predict the landing site using linear regression
    - We found that LM35 thermometer was outputting unusable disturbed data while sending via radio. Because of that we use BMP280 as our main thermometer.
    - We changed GPS module to Adafruit Ultimate GPS, it includes RTC module so one planned in PDR is not needed anymore.
    - Particulate matter sensor was changed to SPS30

2. Introduction

2.1. Team composition

Our team consists of 5 students and our tutor, all of us are attending High School No. 2 named after King John III Sobieski in Kraków. Aleksandra Musiał - Our tutor, CS teacher and also an administrator at our school's 3d printing laboratory.

- <u>Jakub Czarny</u> - Team leader & Electronic designer. His primary role is managing our progress and mission but he also specializes in electronic design and is responsible for choosing components according to our financial and electrical constraints. He is also our 3D designer.

- Andrzej Szablewski - PR spokesperson, he takes care of all PR related missions like finding sponsors and managing and updating our site. He also works on 3D modelling and testing.
- Maciej Gamrot - High-Level Developer, he is responsible for main system development and communication with ground station. He also works on machine learning, look after our repository and develops our website.
- Damian Legutko - Embedded Developer, his job is to write code for managing data retrieved from sensors, designing backup procedures. On top, Damian designs graphical parts of our project when needed.
- Krzysztof Parocki - Hardware Engineer, he is responsible for electronic design, testing prototypes and overall design of the construction. Krzysztof makes photo-documentation of our progress, and makes a movie based on our creation process to promote our project in social media.

2.2. Available resources and time management

We’ve established cooperation with our school that provided us a classroom, where we hold our meetings and 3D printing lab that is under our tutor’s management. We have access to all the tools needed for the project in a public workshop - Hackerspace Kraków. Outside of the school we work on our own as well as in group through communication apps like Messenger and Discord. We meet once a week in our classroom to summarize the previous week and divide tasks for the next few days.

2.3. Mission overview

2.3.1. Primary mission

We've met almost all criteria for completing the primary mission. All we have left to is completing the parachute test it with our case design.
We completed tests of communication and were able to transfer all necessary data: pressure, temperature and elevation. At our range tests we we were able to transfer location at a range of 2.8km through obstacles (buildings and land elevation) and stopped not because of loss of signal but because of time restrictions. Our highest recorded RSSI was about -129 so we expect only better results in further tests.
We have completed our first design of the case and 3D printed it and following iterative design we will have more prototypes in the future to ensure our final case is up to our needs.
The electrical design is fully tested, functional and ready for secondary mission upgrades.

CanSat mission is divided into few stages. Because of the importance of primary mission, mission stages are dependent on the time from hardware oscillator clock and photoresistor, secondary mission readings are used only as reference.

- **Initialization**
  During *Initialization,* software will diagnose every component and save received data in log files on SD card. Checking if everything is working properly would prevent from errors that might occur during the mission. In this stage Can should be standing still on a flat surface for about 3 seconds because few sensors inside would be calibrating.
- **Await**
  The Can will wait for being placed inside the rocket. Probe will occasionally collect sensor data and *photoresistor* readings will indicate if is put in tray and trigger next stage.
- **Launch**
  Except for light intensity, acceleration values from IMU module would be used as reference to mission stage. Sudden acceleration during lift might cause electrical components to behave in unexpected way, to avoid possible errors most of them will go into sleep mode. If Can could not receive any data after estimated time goes into next stage.
- **Fall**
  As a main part of primary mission, in this stage probe will collect data from all sensors, save them to SD card and send via radio.

- **Recovery**
  Probe goes into recovery stage after estimated time or when it detects landing via accelerometer or gyroscope. It waits to be found by searching team but still is collecting data used in secondary mission.

2.3.2. Secondary mission

Our main goal in the secondary mission is to gain first information about disasters and also volcanic activity. Our idea in case of volcanic activity is based on dropping the satellite inside the volcano, while it’s dormant.
Particulate matter density and humidity can be used for controlling and predicticting volcanic activity. Particulate matter is a very important factor in our secondary mission. Dust intensity determines chances for survival for both animal and human beings. High densities are very dangerous for life and are a direct information that the region needs evacuation. Other factors like temperature and humidity are also important as an information about life conditions. Our system can be used in fires as well as other natural disasters. The whole probe is supposed to output the collected data for the processing to happen on ground. This allows us to quickly adapt to whatever is needed and analyze the data thoroughly.
As for launching the probe in finals we expect to test all systems and check for errors as we don’t predict the army range will output dangerous data signalyzing any disasters. The whole point of the probe is life-sensitive data collection which can be interpreted in many ways.
In addition we will use landing site prediction software to help us understand where we are measuring and assist the primary mission recovery.

3. Probe description

3.1. Introduction

The whole probe will fit in a 3D printed case. Our list of electrical components is now final when it comes to primary mission components and matches our needs and requirements. Every planned sensor and its function is described below.

3.2. Components description

3.2.1. Microcontroller board

We managed to test the CanSat Kit microcontroller board and decided to use it inside the probe due to its big capabilities. It completely fulfills our needs and no board change is needed.

Current status: Tested and ready for final assembly

3.2.2. Batteries

We chose to go with 2600mAh modules (ICR18650-26FM) because we found a particular seller which would pre-weld them to a tin strip resulting in a 3P configuration of 7800mAh. This is still a massive battery pack that after a few hours of testing communication its voltage went down just 0.04V and will surely last long enough for the whole flight.

Current status: Tested and ready for final assembly

3.2.3. Atmospheric pressure sensor

This sensor fulfills all primary mission data gathering requirements, from received data we are able to calculate altitude of falling Can. The BMP280 provided with the CanSat Kit works just fine through the I^2C interface and we haven't encountered any problems so far. It's also working as our main thermometer, because it's much more stable and calibrated than the LM35.

Current status: Tested and ready for final assembly

3.2.4. Particulate matter sensor

The particulate matter sensor is our main focus right now in the secondary mission. Basing on data collected from this sensor it is possible to examine life conditions. We opted for the SPS30 which provides us with not only higher accuracy compared to the PMS7003 chosen earlier but also a high sampling rate of 1Hz that erases the need for slower descension.

Current status: In shipment

3.2.5. Humidity sensor

Air humidity values would be used in secondary mission to predict possible fires. Very low humidity is often main cause of those disasters.
The DHT22 which we use to measure it has proven to be difficult to operate. It is causing board to behave in unstable way yet it update rate is very low. So far we could not find any better component to replace it.

Current status: Tested, possible upgrade planned

3.2.6. GPS module

We decided to buy the Adafruit Ultimate GPS module based on MTK3339 chip. It was more expensive than our initial choice but its capabilities and big community support made it a best option. It also provided us with a RTC powered by a backup battery.

Current status: Tested and ready for final assembly

3.2.7. IMU module

Accelerometer readings from this module could be used to predict earthquakes and avalanches.
Additionally IMU module will give us information about Can momentum additionally it is very useful during tests. We plan to use it when determining landing spot.
With use of special library we managed to get it working as intended. Kalman filters and Magdwick algorithm used in the library give us very accurate data but we need to convert it into a form that satisfies our needs.

Current status: Tested, needs software upgrade

3.2.8. Buzzer

Buzzer will help the search team in locating the Can on the ground after finishing the mission. Sounds made by it can exceed 85 dB, and definitely would be heard by search team.

Current status: Planned

3.2.9. Photoresistor

We can get information about CanSat mission stage in a very tricky way, thats by implementing photoresistor. When probe is outside the rocket it detects large amounts of light, if is inside a tray and is covered, no light shines onto photoresistor. So far we don't need it and that is why we have not included it inside out prototypes.

Current status: Planned

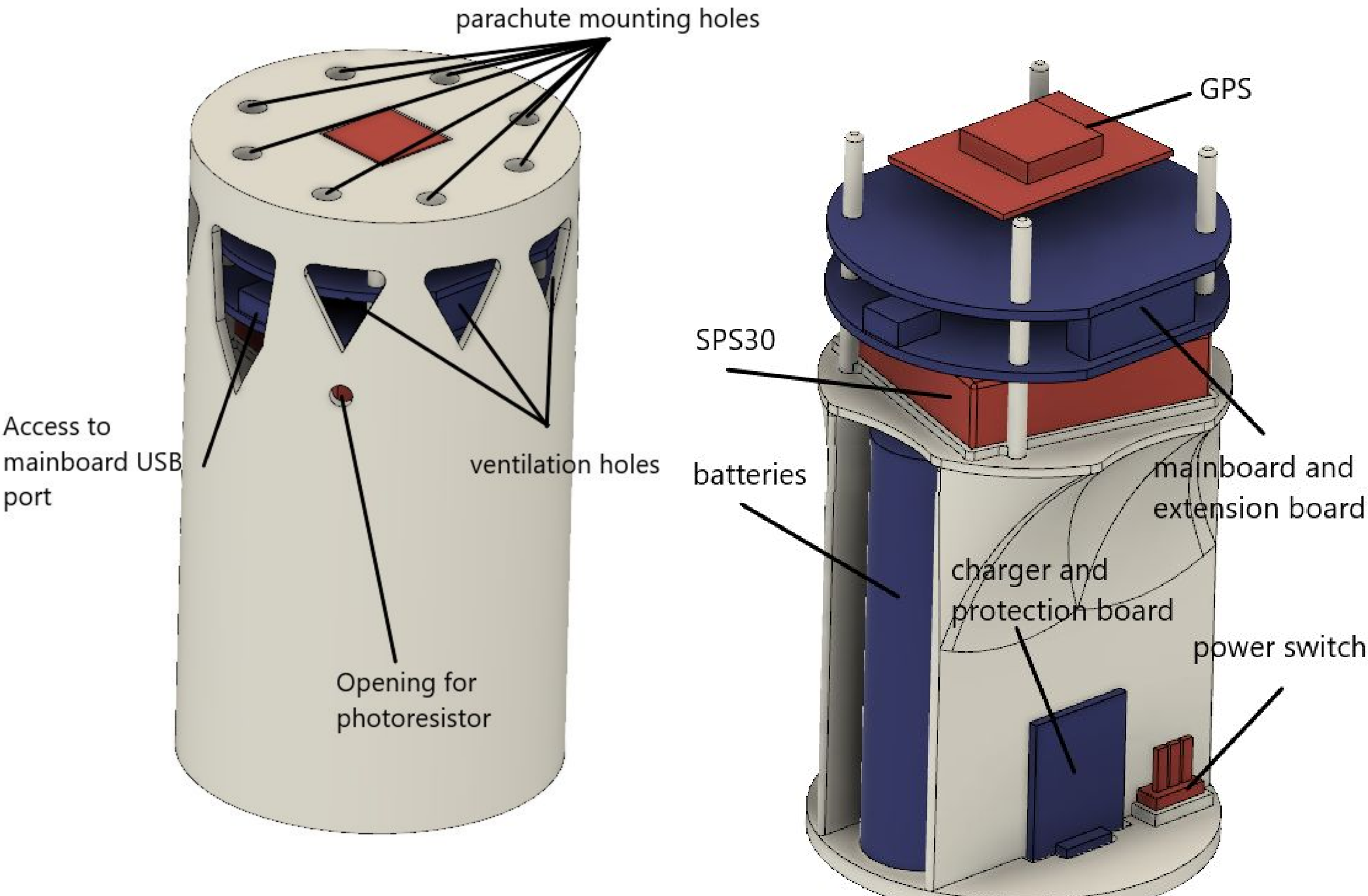
3.3. Measurements

Table below presents data that will be gathered during the mission. **Minimum** and **maximum** columns represent predicted values to be received. Readings precision is presented in **Significant figures** column with format: *[before ; after decimal point]* in base 10.

Sensor	**Reading variables**	**Minimum**	**Maximum**	**Significant figures**	**Sampling time [ms]**
BMP280 (I^2C)	Pressure	600 *hPa*	1100 *hPa*	4 ; 1	40
	Temperature	-10*°C*	30*°C*	2 ; 1	40
Adafruit Ultimate GPS (UART)	Latitude	-24,09°	-14,07°	2 ; 4	1000
	Longitude	49,00°	54,50°	2 ; 4	1000
	Altitude	0 *m*	3000 *m*	4 ; 1	1000
	Clock	mission start time	mission end time	-	1000
DHT22 (Digital communication)	Humidity	0 %	100 %	2 ; 0	2100
	Temperature	-10*°C*	30*°C*	2 ; 1	2100
SPS30 (UART)	Mass concentration	0.3 $\mu g/m^3$	10 $\mu g/m^3$	1 ; 1	1000
	Number Concentration	0.3 $1/cm^3$	10 $1/cm^3$	1 ; 1	1000
MPU 9250 (I^2C)	Tilt	0°	360°	3 ; 2	10
	Acceleration	0 *g*	4 *g*	1 ; 2	10
	Magnetic	0°	360°	3 ; 0	30

	orientation				
Photoresistor (Analog)	Light intensity	0 *lux*	10 *lux*	1 ; 2	10

3.4. Mechanical design



3.4.1. Weight

We are almost sure that
the whole weight of the can and the parachute will be in a required range (300-350g). On this day our can weigh exactly 282g. We estimate use of about $0.62m^2$ parachute material with an area density of $38g/m^2$ — the parachute will weigh about 24g. Beside that we need few more components — about 30g.
Totally - 336g.

3.4.2. Case

We have decided to use ABS based filament called Z-UltraT for 3D printing the case, because it's very strong and light. The parachute will be made from a professional grade material - Skytex 27 Classic. 3D design of the case is being done completely in Autodesk Fusion 360. Since PDR we went through 4 different design concepts of which 2 were 3D printed for development purposes and final concept testing. We plan to perform strength tests really soon and update the final design accordingly. Since our only available material is based on ABS it's unlikely the case will fail.

The current status of our 3D designs can be found in attachments either in .f3d or .step files.

Because we are following the iterative design process these models are only concepts which were 3D printed for prototyping but will be improved with further testing and printing.

3.4.3. Parachute

We performed several tests of our first prototype parachute made from a precisely cut plastic foil at a diameter of 50cm, which resulted in maximum speed of 7.5 m/s after falling from 15m.

$$h_{max} = \frac{at^2}{2} \quad v_{max} = at$$

$$t \approx 4s \quad h_{max} \approx 15m \quad v_{max} \approx 7,5\frac{m}{s}$$

This was done after overestimating the necessary diameter of the parachute and allowed us to move on to the next version made from the material provided by one of our sponsors.

$$r = \sqrt{\frac{2 \cdot 9,81\frac{m}{s^2} \cdot 0,35kg}{\pi \cdot C_d \cdot 1,28\frac{kg}{m^3} \cdot \left(8\frac{m}{s}\right)^2}}$$

According to our calculations our parachute should be about 36 cm in diameter but after adjusting for the center hole version 2 of the parachute will be 40 cm. After tests of version 2 we will be able to adjust accordingly.

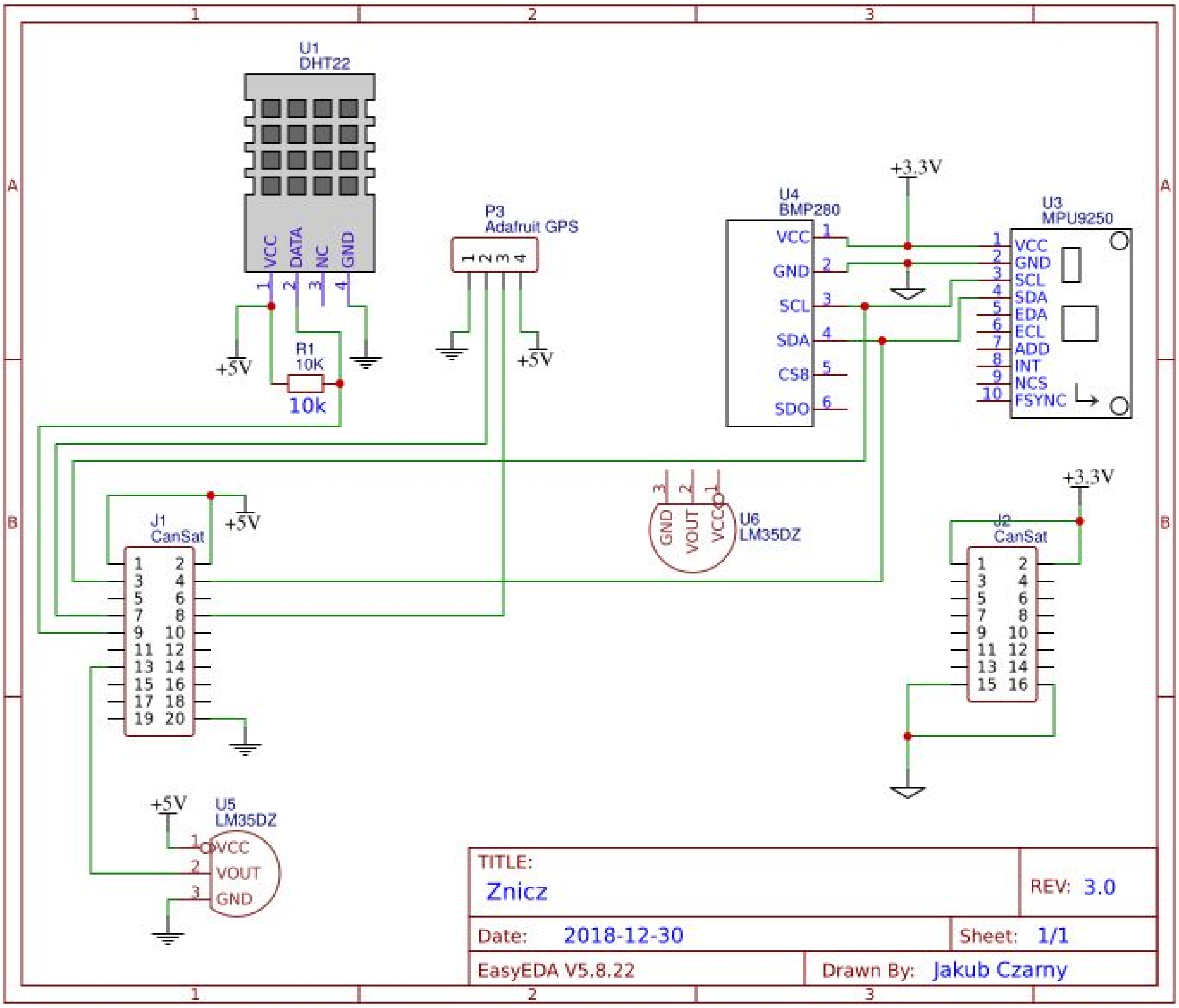
3.4.4. Transmitting antenna

For the transceiving antenna on the can we used a 17 cm long solid core wire soldered directly into the microcontroller board. So far it suited its purpose but in case of it being not enough we can replace it with a manufactured one.

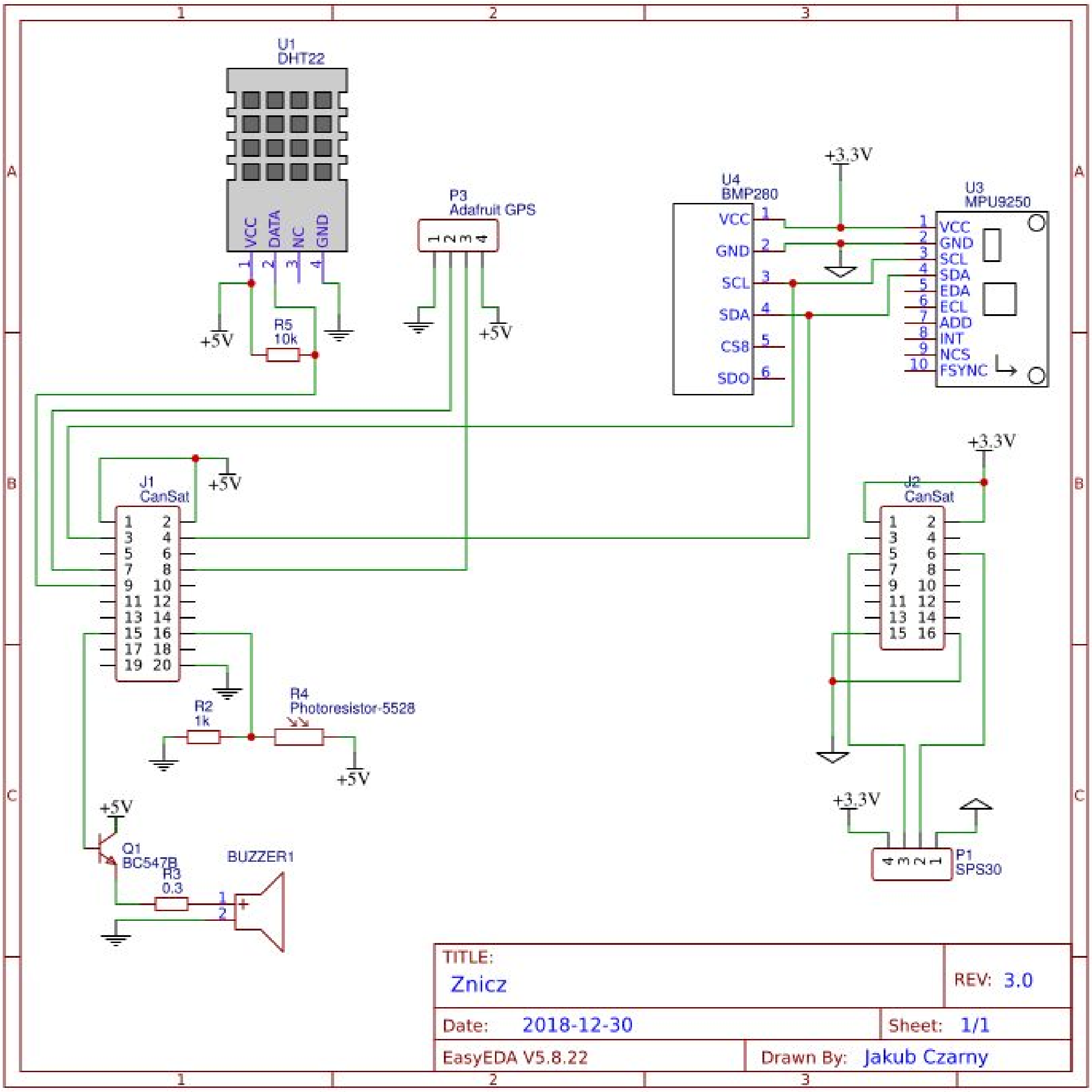
3.5. Electrical design

3.5.1. Prototyping board design

Here is the schematic of a board we used for development:



And here is our final design that features only components we settled on:



3.5.2. Power consumption

Here is the theoretical maximum power consumption e.g. worst case scenario.

Component	**Maximum current draw [mA]**	**Voltage [V]**	**Efficiency[2] [%]**	**Effective power [mW]**
Microcontroller	20[1]	3.3	50	132
SX1278	120	3.3	50	792
BMP280	1	3.3	80	7

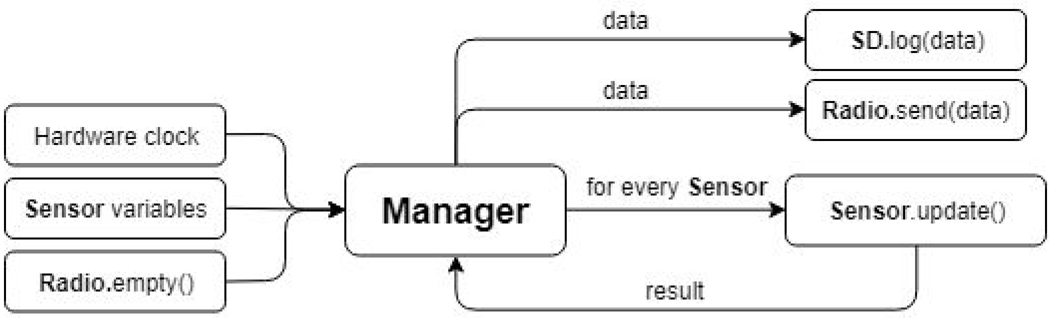
SPS30	80	5	80	500
DHT22	1.5	5	80	9,4
Adafruit Ultimate GPS	25	5	80	156,3
MPU9250	3.7	3.3	50	24,4
Total	**251,2**	-	-	**1621,1**
Buzzer is only going to operate at the end of the mission. Photoresistor is drawing an unnoticeable amount of current so we decided to skip them in the calculations				

[1] - Here we used the average values provided in the CanSat Kit documentation, because we think they are more suitable and we couldn't find the maximum power draw for the microcontroller.
[2] - Efficiency is based on which power source is used by the component and it's either 80%(Efficiency of the step-up converter) for +5V or ~50%(80%*66% - combined efficiency of the step-up converter and line regulator) for +3.3V

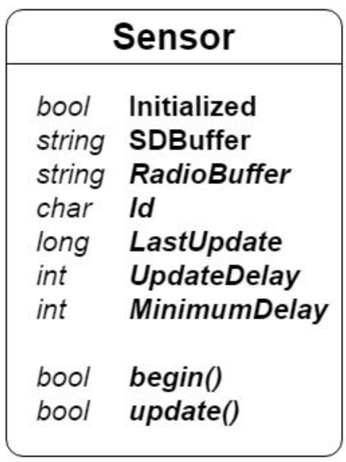
Battery capacity = 2600mAh*3*3,7V = 28860 mWh
Total power consumption = 1621,1 mW
Theoretical maximum operational time = ~17,8h

3.6. Software design

3.6.1. System structure



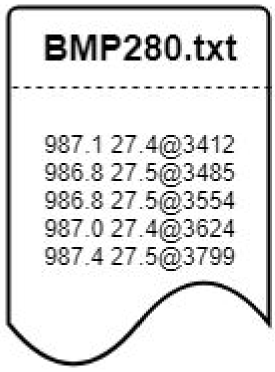
Main body of CanSat software is a **Manager** object, it defines mission stage and distributes program time between all classes inside probe. It decides for example when to save sensors buffers to SD card or send them via radio. While developing software we found out that board crashes if some of instructions are send one after another, **Manager** cares to not invoke such incidents by leaving proper time



intervals between those instructions and is responsible for error handling if they occur.
For every sensor inside probe, there is a class with defined data reading methods that inherits from abstract **Sensor** class.
Sensor class consists of initialization and update methods, radio and SD buffers, diagnosis variables and update time. Software structure like this allows to use every sensor in unified way without worrying about it's other methods required for proper working.

3.6.2. Data gathering

Every sensor has different specifications and by that every has its own update time, to not lose data while waiting for those with low sampling rate, sensors are invoked dynamically by **Manager** when their update time expires. With every measurement sensors input new reading into their *SD buffer* and every reading is marked with hardware clock time. For example *BMP280* saves pressure and temperature floats and *GPS* saves last received *NMEA* sentence. Methods responsible for data gathering are different for every one of them but their outputs are later unified to fit **Sensor** base.
SD card contains separate *.txt* files for each sensor. Every measurement occupies a single line with a **time marker** at its end. *Example file is presented on image on the left.*



3.6.3. Event logging and timers

Our CanSat software implements functions responsible for logging too. Every important action inside probe will be saved with time marker on *.txt* log file.
By **time marker** in CanSat we understand *millis()* function that returns amount of milliseconds elapsed since board start returned by hardware clock. Additionally software gets current time from *RTC* module inside *GPS* board, at the beginning of log file there is an information about real time and time marker.
Example heading of a log file: *#110119161503412@1302*
This string lets us know exact day, month, etc.. and *time marker* during this moment (number after @ char). Every action after this line will be marked only with time marker because we can simply add difference of elapsed milliseconds to get event time.

3.6.4. Radio transmitting

So far during our tests we have transmitted data in the standard way, that is by sending values in decimal form in strings.

During tests we found packet of length 47 bits as most optimal transmission configuration. It gives us about 2 transmissions per second with 97B/s speed.
In the final version we plan to send compressed values based on measurement estimated value range and its binary representation. Data values collected from sensors would be mapped to some binary number with length specified by reading precision. For example knowing that temperature is in range of -10 to 30°*C* and reading precision is: two digits before and one after decimal point, we can map value from 0 as -10.0°*C* up to 400 as 30.0°*C*. Numbers in that range can be represented as binary number with length of 9 bits. While sending string like: ‘14.2’, 4 chars are send and that is equal to 32 bits. In this case we have compressed our data by factor of about 3.5!

4. Ground station

4.1. Radio receiving

Antenna:
Our base station antenna (LPRS YAGI-434A) was ordered from UK and works without any problems. We had to design a handle and print it on a 3D printer. After installation it performs exactly as intended. It already was equipped with an SMA connector so no modifications were necessary. Its aluminium construction makes it very light and easy to handle which helps make it handheld.

Data received by the receiver will be outputted to a serial port in raw form. Software ran on the computer connected to the receiver via USB will listen for input. Read data will by parsed, displayed and subjected to further analysis. Each input will be also saved by EventManager both in parsed and raw form. Example file by EventManager:

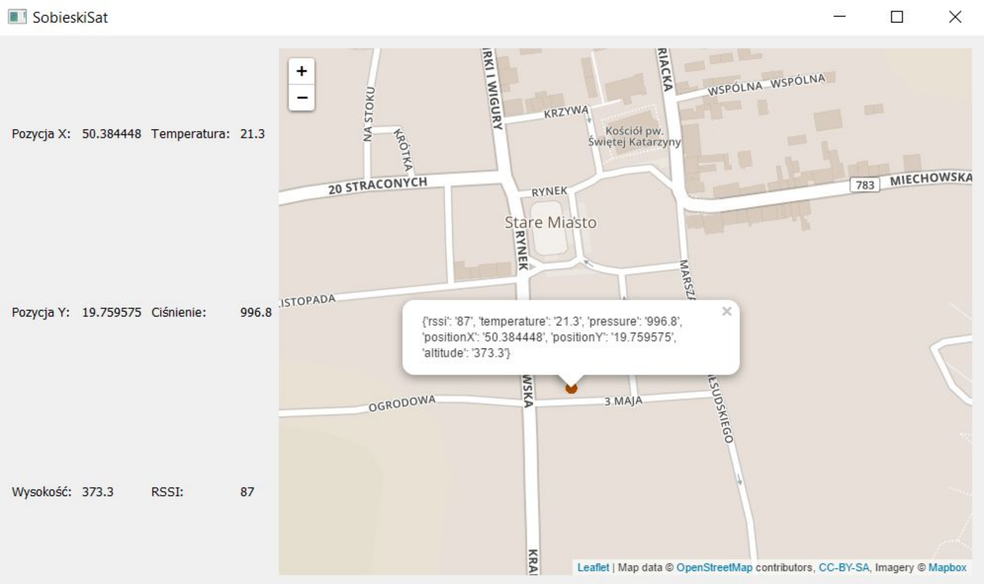
GroundStationLog.txt

```
Tue Jan  8 12:34:40 2019 Radio __init__ {'kwargs': {'kwargs': {'baudrate': '115200', 'port': 'COM9', 'timeout': '1',
'kwargs': {'event_manager': <Modules.EventManager.Manager object at 0x0369CA90>}}}}
Tue Jan  8 12:34:40 2019 Radio open_serial {'kwargs': {'kwargs': {'baudrate': '115200', 'port': 'COM9', 'timeout': '1'}}}
Tue Jan  8 12:34:40 2019 Radio readline:call
Tue Jan  8 12:34:40 2019 Radio readline:received {'kwargs': {'kwargs': {'data': b'15_23.13_980.62_50.0684_19.9318_231.6\r\n'}}}
Tue Jan  8 12:34:41 2019 Radio readline:call
Tue Jan  8 12:34:41 2019 Radio readline:received {'kwargs': {'kwargs': {'data': b'14_23.13_980.59_50.0684_19.9318_231.6\r\n'}}}
Tue Jan  8 12:34:42 2019 Radio readline:call
```

4.2. Ground Station Interface

We divided main window of ground station graphical interface into a few groups:

- Information Board display live information from satellite (temperature, humidity, etc)
- Interactive Gyroscope Widget informs about the inclination of satellite relative to the ground by processing data from GPS
- Interactive Map Widget plots current and past location of satellite creating the flight path of our satellite. Each point is colored depending on RSSI and can be clicked to display further information about satellite at that moment. We also plan to add landing zone prediction feature using machine learning based on location, speed, acceleration and wind strength.
- Group of buttons used to control the software (work in progress)
- Interactive graph plotting live data from satellite (work in progress)



4.3. Landing prediction feature

We plan to implement landing prediction feature into our ground station software. So far we found linear regression methods as the best candidate. From received *GPS* data we are able to represent the Can's route as a line in three dimensional space. Having known ground altitude we can solve the linear equation and get the predicted landing position. The results definitely would not be very accurate, in the future we plan to also use IMU data providing we find proper methods to use them.

4.4. Noise Cancelling

During tests, we discovered that some data packets were corrupted during long-distance data transmission, so we decided to develop specialized software that could recognize these corrupted packets based on previous one. We also consider using machine learning for this since we have access to a large amount of data.

4.5. Data Analyzing Software

After flight probe can be plugged to the computer for analyzing information saved on SD Card. Software will recognize connection and enter Debug Mode in which every measurement will be plotted in a window similar to the one with live data, but with feature to reenact flight history in a real time and readout by readout. All information will also be compared and analyzed with regard to irregularities and interferences.

5. Project summary

5.1. Resource management

5.1.1. Budget and costs

Note: All prices in this section are converted from PLN to EUR with exchange rate 1 PLN = 0.232996887 EUR (05.01.2018 - Google)

We have started with 267,95 EUR. We hope we will get more money and partners. Now we are on a good way to get a sponsorship from Kraków Airport. On this day (05.01.2019) our spending is presented in the table below.

No.	Thing we have bought	Price (EUR)	Category
1.	website hosting	4,66	Promotion
2.	Facebook advertisement	5,82	Promotion
3.	2x Charger TP5056, 2x Buzzer, 1x DHT22 and shipment	7,47	Modules
4.	1x MPU9250 sensor, 3x batteries, welding service and shipment	21,64	Modules
5.	1x Adafruit Ultimate GPS and shipment	38,09	Modules

6.	Antenna and shipment	53,42	Modules
7.	Electrical tape, sandpaper	1,05	Modules
8.	Food, snacks, drinks for meetings	31,43	Meetings
	SUM	**163,58**	

Now we have 267,95 - 163,58 = **104,37 EUR**. We have already bought everything we need except few parts. Our estimated expenses are: few more modules and eventually cost of the participation in the final for the fifth member of a team.

5.1.2. CanSat Value

Note: All prices in this section are converted from PLN to EUR with exchange rate 1 PLN = 0.232996887 EUR (05.01.2018 - Google)

The whole value of our CanSat is presented in the table below.

No.	CanSat part	Value (EUR)
1.	Adafruit Ultimate GPS	34,92
2.	Batteries and welding service	11,87
3.	MPU9250	7,69
4.	DHT22	3,69
5.	Buzzer	0,47
6.	Li-Ion Charger	1,21
7.	SPS30	40,93
8.	Case - 80g Z-ULTRAT material	4,66
9.	Parachute and cords	5,00
	SUM	**110,44**

5.1.3. Sponsors

Note: All prices in this section are converted from PLN to EUR with exchange rate 1 PLN = 0.232996887 EUR (05.01.2018 - Google)

We wanted to amp up our chances for getting a sponsor so we have written to the following companies:
Comarch, Asseco, Cd projekt, Orange, Play, Coca cola, Pepsi, Rmf fm, Ailleron, Shell, Orlen, Lotos, Arge, Tvn, Radio zet, Tok fm, Kraków Airport, Codewise, Flytech UAV, Kraksky, botland.pl, School parental council, Company which want to be anonymous.

We have several sources of money as well as partnerships. They are presented in the tables below.

No.	Source of money	Amount (EUR)
1.	School parental council	186,40
2.	Team contribution	58,25
3.	Team advisor contribution	23,30
	SUM	**267,95**

No.	Partnership	Our benefits
1.	Our school	We have unlimited access to our school's 3D printing lab
2.	Botland.pl	We got a 7% discount on all their items
3.	Company which wants to be anonymous	We got parachute material and parachute cords - The material is called DOKDO-N20DMF-WR-HD38
4.	Kraków Airport	We are on a good way to get a sponsorship from Kraków Airport. All we need now is an official confirmation from CanSat organizers about safety and other regulations of the satellite launch.

5.2. Schedule

No.	Date	Planned action
1.	17.01	Finishing Parachute
2.	19-20.01	Testing Particulate Matter Sensor
3.	25.01-5.02	Final development and designing final case
4.	10.02	Printing final version of the case
5.	10.02-20.02	Adjusting all components and writing final code
6.	20.02	Fitting all components in the can, building it ultimately and checking weight
7.	25-28.02	Ultimate tests of communication, transmitting data, parachute and checking if all components work together without any problems
8.	02.03-9.03	Ultimate trials of everything and writing FDR
9.	10.03	FDR

5.3. Documentation

5.3.1. Social media & promotion

We have over 300 likes on our Facebook account (linked below). Posting regularly 20 posts, we have on average 400 views under every single one. We have also several posts with about 1000 views, and also one, promoted, with nearly 8000. This post is a movie-type which is a vlog from our first radio transmission tests. By doing this, we wanted to promote our project as well as the whole competition among the local and school community. Due to the project promotion we hope we will get more partners. Beside that we promoted ourselves on our schools facebook group which has over 3000 members.

Facebook site: https://www.facebook.com/SobieskiSat/

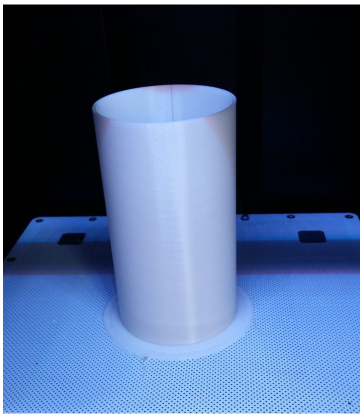
5.3.2. GitHub Repository

We decided to create a public organization in github, so that everyone interested in our project could follow our progress. You can find there the entire code of the satellite and ground station. We also include all 3D models, diagrams and documents of our project. This helps to organize and archive all of our work, enabling easy tracking.

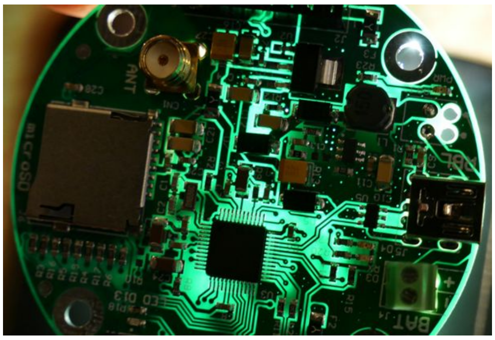
Our GitHub: https://github.com/SobieskiSat

5.3.3. Timeline

26 September – The date of one of our first actions taken to develop our micro-satellite. This day we finally got access to the school 3D lab, where we printed the first case to see exactly how much space we do have to fit in all components. We have also mentioned this action in PDR, as one of the Project Summary.

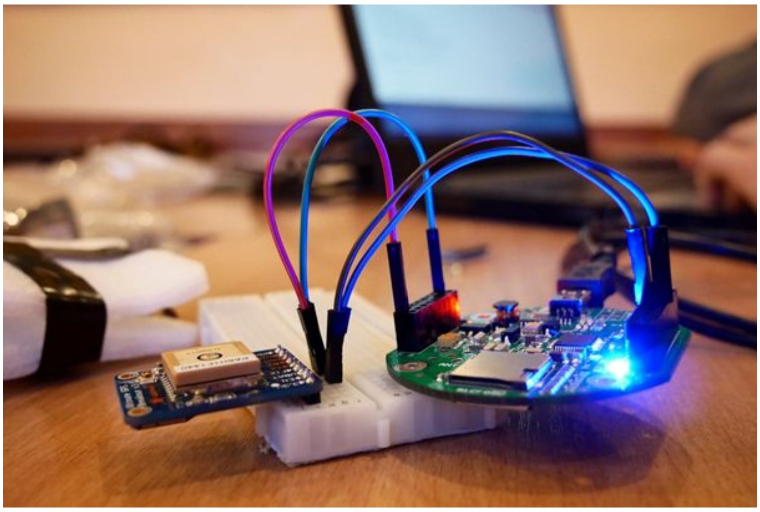


30 September – that day we met with the airplane pilot, who helped us to design and plan a lot of thangs attached to aerodynamics and electronic circuits. We elaborated a project of components composition inside the case, spoke a lot about secondary mission idea and chose the way which our marketing and advertising actions will follow in the future.



20 November – our first meeting during which we were testing modules we had received a couple days before. We have tested components such as accelerometer, gyroscope and magnetometer, simultaneously with other essential modules needed to complete the primary mission. All gages of these we have done on a breadboard, but after that we still needed to wait for missing parts.

27 November – that was our second meeting during which we worked with modules, provided to us a few days after the first meeting. We finished the attunement of the GPS module, and we plugged in the same time every other module to see if it will work correctly – the only one missing was particulate matter module, which we planned to add and trial at one of our future meetings.

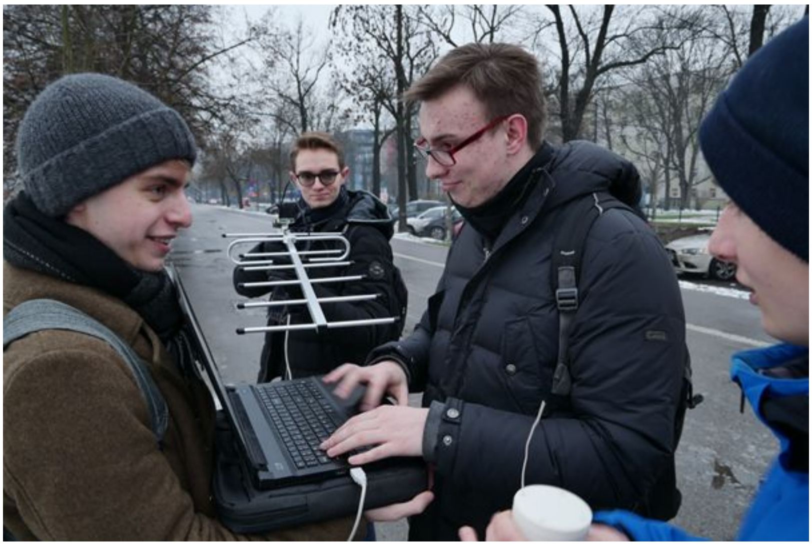


11 December – Our measures to design a first prototype were over. We printed a new case for our modules, this time it was more durable and contained a top, so that we could close components up inside. We fitted in all modules needed to complete primary mission and some from secondary and wrote a code to be able to send data from the can and receive it in our ground station, so the first prototype was finally ready to try.

13 December – we got an high-range aerial ordered a week before, that we needed to receive data from our can and relay it further to the ground station. We did a few adjustments in our code to meet the aerial requirements and make everything work correctly, and decided that we need to probe the real aerial range to check if the signal will work during the final competition.

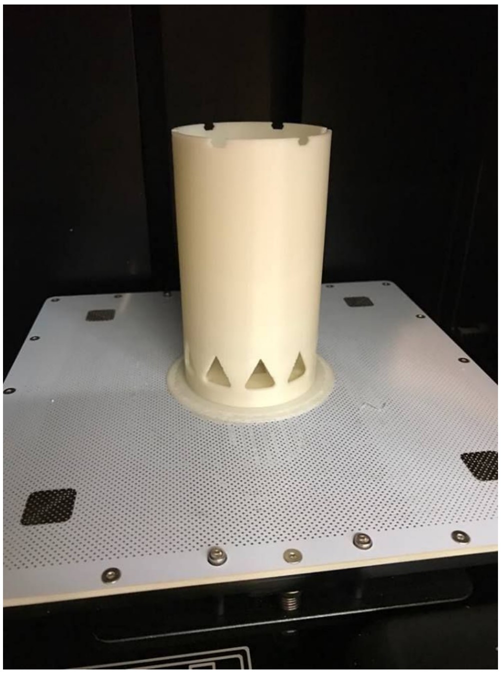
18 December – The day of trials. We were testing our aerial range for about 4 hours, by leaving the ground station in the northernmost part of Błonia, and walking away as far as we could with our sender.

We have reached 2,8 km distance, without a big loss of signal, hence we were certain that out satellite will be transmitting data all the time during the plummeting at the finals. That day we have also completed our stab at cansat falling with parachute. Full report of our efforts was posted on our facebook page as a vlog and animation of our repositioning



23 December – The first usable prototype of the parachute was done. All measurements and calculations had been completed, and afterwards the plastic foil was cut and formed.
Work on the code was also going forward that day

8 January – We have 3D printed a new version of the case with some improvements, for example holes to provide better air-circulation and new latch system to test new ways of joining the two halves. We have also thrown it with a parachute out of a balcony to check if it will fall properly without destroying itself.



11 January – Parachute material was finally with us, provided by the company which wants to stay anonymous – we got over six times more material than we wanted, which enabled us to make multiple versions and test each one separately. We've also set up our Github page