Critical Design Report

Team Name:
Jeźdźcy Jarzynowej A.K.A. JJ CanSat Team

Country:
Poland

1 INTRODUCTION

1.1 Team organisation and roles

Patryk Gałczyński

Project manager, supervisor

Patryk is a third year Computer Science student at the AGH University of Science and Technology. He founded and runs RoboTeam faculty in our high school, where we know him from. He made couple autonomous mobile robots called mini-sumo and line-followers. He works as PaaS layer cloud platform developer in Dreamlab, which means lots of DevOps challenges and writing tons of code in Python (which is his favorite programming language these days).

Tasks:

Patryk is acting as our project manager, and supervisor. That means, he provides us with tools to work efficiently (like trello or git repository), organizes our work by planning meetings, and pushing us to work hard every day. He also is our inexhaustible source of knowledge about electronics, unix systems and software engineering.

Karol Oleszek

Team leader, ground station software

Tasks:

As a leader, Karol binds the group together, distributes tasks and makes sure they are completed on time. He also takes part in programming the base station for our CanSat, documentation and communication with our sponsors.

Programming:

- development of GUI applications (Delphi, C#, C++)
- games (Delphi, Unity 5)
- web solutions (PHP, JavaScript, MySql, HTML5, CSS3) in e-commerce (school's book sale) and web games (simulation of aerospace industry);

Physics:

- participated in “International Masterclasses Hands on Particle Physics” workshop organized by CERN;

Economy and social studies:

- laureate in Voivodeship Contest of Economy and Social Sciences Competition for Middle School Students, earned two times;

Jan Kostecki

Electronics, receiver software, construction

Robotics:

- participated in several robotics competitions,
- member of school RoboTeam;

Programming:

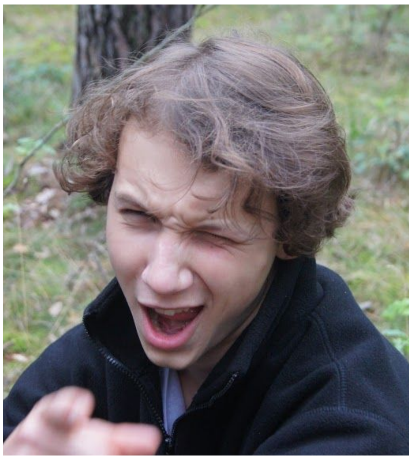
- programming in C++, SQL, Java, Arduino IDE and NXC;
- interested in development of web solutions (PHP, JavaScript, MySql, HTML5, CSS3) in e-commerce (school book sale, together with Karol Oleszek) and web games (simulation of aerospace industry with Karol), attends mobile applications programming course;

Physics:

- laureate in Voivodeship Contest of Physics for Middle School Students,
- participated in "International Masterclasses Hands on Particle Physics" workshop organized by CERN;

Tasks:

- Jan designs electronic and mechanical solutions for our CanSat, such as its wind speed sensor, as well as its radio receiver software.



Bartłomiej Biesiadecki

Construction, electronics

Robotics:

- took part in Nationwide High School Robotics Contest "µBot" (together with Jan Kostecki), member of school RoboTeam;

Programming:

- programs in C++, SQL, attends mobile applications programming course (Swift);

Physics:

- participated in "International Masterclasses Hands on Particle Physics" workshop organized by CERN;

Economy:

- took part in Expo FXCUFFS 2016 Investment Congress;

Tasks:

- Bartłomiej is mostly involved in developing mechanical solutions, such as the parachute, and the overall process of landing. He also helps Jan Kostecki in development of electronic solutions.

Tymoteusz Chmielecki

On-board software

Programming:

- Programming in C, C#, C++, Rust, Python3, HTML5, Java, JavaScript, Arduino IDE;

Aeronautics:

- in his free time builds and pilots quadcopters;
- Participant of the Lesser Poland's Scholarship Programme for Especially Talented Students for his high grades;

Tasks:

- Tymoteusz writes code for onboard electronics and integrates internal solutions with ground station software.

Jan Dziedzic

Physics calculations

Programming:

- programming in C++, Bash, Python, Arduino IDE;
- laureate in Voivodeship Contest of IT for Middle School Students,
- laureate in Theoretical Computer Science Olympics for Middle School Students (individual & team)

Astronomy:

- Polish representative on ESO Astronomy Camp 2015,
- completed total of 4 weeks of internships in Astronomical Observatory of University of Warsaw,
- built his own roll-off roof remote astronomical observatory;

Physics:

- laureate in Voivodeship Contest of Physics for Middle School Students,
- participated in "International Masterclasses Hands on Particle Physics" workshop organized by CERN;
- Participant of the The Polish Children's Fund Programme for his achievements in numerous competitions of IT, Maths and Physics;

Tasks:

- Jan takes care of most of the calculations needed for the project, his role will be most important during the process of post-launch data analysis. He is supervising all calculations done in the group. He also prepares mock data to test live data analysis software and handles team purchases.

Photo 1, The team

Although everybody in team has his own role and specific tasks designated, we focus on teamwork and cooperation. Every bigger issue, problem that someone faces and new feature he wants to implement is discussed by whole team or in smaller groups (i.e. CanSat software is reviewed by other code-writing team members through pull request system in our GitHub repository). We are constantly in touch with each other since we use Facebook group conversation to discuss present actions. We use Trello kanban table to organise bigger tasks into lists of single task.

We organise weekly meetings in order to track progress of our project and exchange ideas for fixing bugs, and share knowledge to each other. Key decisions are made there and time schedule is being prepared. We use Google Calendar and Doodle to arrange our meetings and set milestones.

1.2 Mission objectives

1.2.1 Secondary mission

The secondary mission of our CanSat is to investigate landing area and using the gathered data, decide whether the place is suitable for landing of a heavy lander. The mission goal is to simulate a cheap and accurate way of preparing planetary landing of heavier equipment.

We have chosen this job for our device, because it might be used in numerous situations, such as preparation for manned Mars landing (after adaptation of the probe's parachute) or investigating remote location by dropping the probe from helicopter or plane.

1.2.2 Measurements and tests

Gathering and processing information about conditions and their impact on the landing of a heavy lander in the studied landing area includes:

1. Measuring the speed of wind and analysis of its effect on lander descent,
2. Taking pictures of the surface in order to assess the landing site,
3. GPS tracking of probe's flight,
4. Saving data to onboard storage for further analysis.

1.2.3 Objectives

We will consider CanSat launch successful when the following objectives are completed:

1. Proper transmission of gathered data to the ground station,
2. Online assessment of CanSat's measurements with the priority of correct determination whether analised site is suitable for heavier lander,
3. CanSat's safe landing with no damage or minor damage not compromising its reusability nor data survival.

1.2.4 Secondary mission criteria

People proved that exploring any location is possible, from landing a probe in acidic high-pressure dense and hot atmosphere of Venus, through using a skycrane to lower a rover onto Mars surface, to descending a not propelled lander onto cold 67P/Churyumov–Gerasimenko comet core.

There are still some limiting factors we can’t overcome easily during a descend of a really heavy lander like a base supply mission or sample return module. Due to this we have decided to prepare following criteria, which, meant for Earth, somehow resemble those applicable on Mars in terms of complication.

Parameter	Min. value	Max. value
Temperature	-20°C	50°C
Humidity	0%	90%
Atmospheric pressure	400 hPa	1200 hPa
Wind speed	0 m/s	20 m/s
wind speed fluctuation per 10s	0 m/s	10 m/s

A notable problem of Earth's atmosphere is its ability to store water in its gaseous form, this occurs in only limited capacity at specific temperature. A border temperature for atmosphere at which it can hold as much water vapor as it contains is called a dew point. When object with temperature lesser than abovementioned one enters such environment, water droplets begin to form on its surface. Such event may cause electronic malfunction destroy samples or obscure optics. An obvious solution to this problem is to warm these components to temperatures exceeding dew point or put them in dried hermetic containers. During data acquisition, our software will calculate dew point, allowing future mission preparation to compensate this effect.

1.2.5 Expected results

We expect that the competition area will be suitable for a heavy lander, but we consider an option that the weather may change the result. If wind is too fast or atmospherical conditions are somehow concerning, we expect that ground analysis will result in negative information about area’s suitability for heavy landing.

2 CANSAT DESCRIPTION

2.1 Mission overview

2.1.1 Overview

Design, manufacture, assembly and testing

CanSat flight is preceded by long preparations including design, prototyping and numerous tests. Project concept is described in this document, but with further testing some specification may change.

Before takeoff, CanSat is prepared with packing parachute and charging batteries.

2.1.2 Take-off

Before the flight CanSat will stay in standby mode which decreases energy demand, ensuring it can work for three continuous hours. After rocket launch / deployment from the plane, all systems are going to be turned on and the packed parachute will unfold.

2.1.3 Flight

CanSat is about to descend with velocity about 9 m/s (no faster than 11 m/s). During the flight, numerous measurements are conducted (wind speed, atmospheric pressure, air temperature, acceleration) and sent back to the ground station, where they are analyzed and results are being displayed and possibly published online. GPS position is tracked, altitude above the surface calculated and photos of terrain are taken.

2.1.4 Landing

Although CanSat is descending with a parachute, special mechanical construction is being applied to ensure that it will withstand touch-down with velocity exceeding desired one in case of partial parachute malfunction.

We have consulted an offshore manufacturer of both standard-sized parachutes and parachute canvas. A decision has been made to purchase a standard parachute and then, if necessary, apply minor changes to it, making sure it meets competition requirements.

2.1.5 Key elements

CanSat electronics includes various sensors, camera, GPS with additional antenna module, radio transmitter, it's antenna and mechanical elements - parachute, wind speed sensor and casing.

2.1.6 Block diagram of electronic components

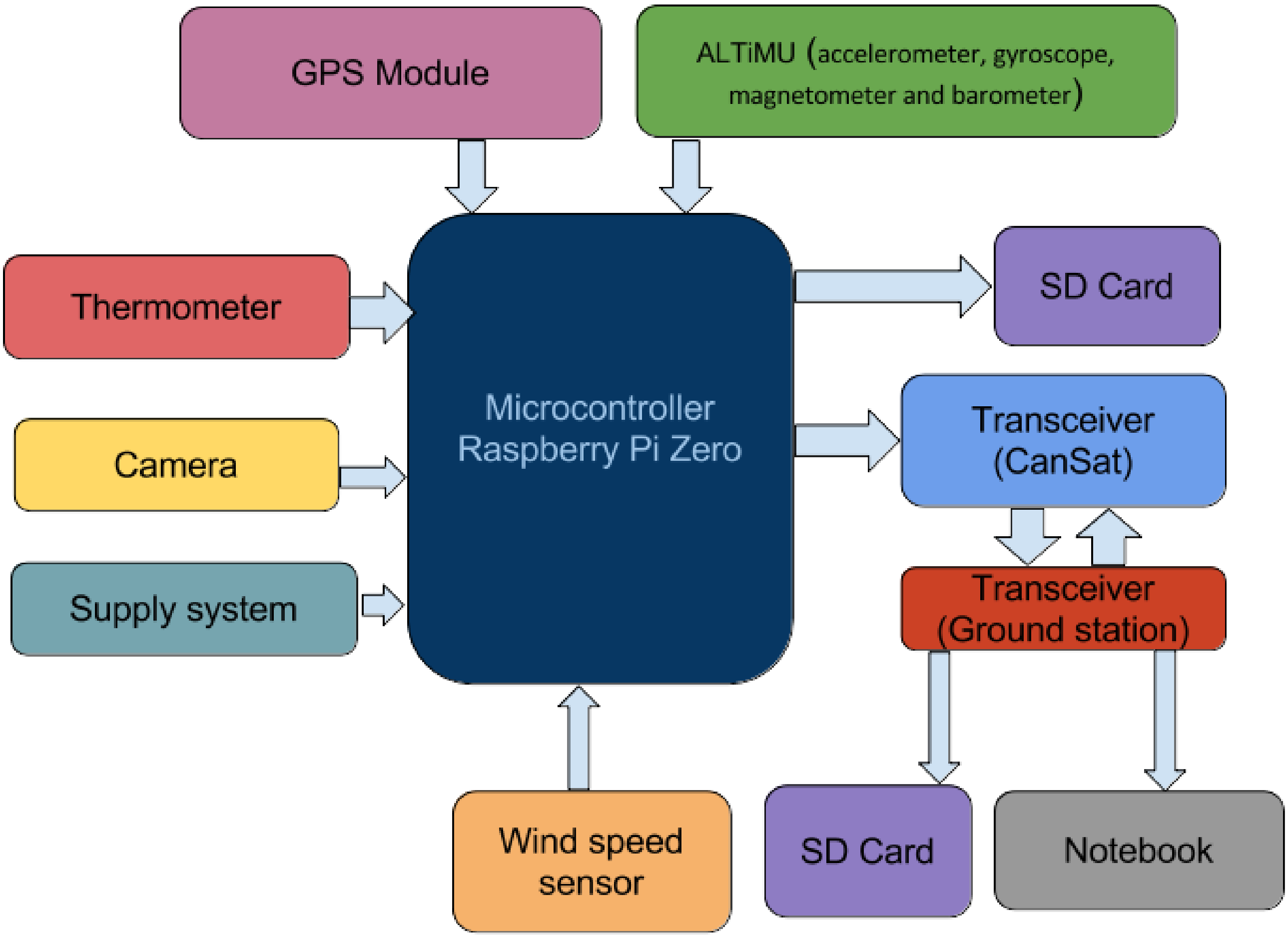
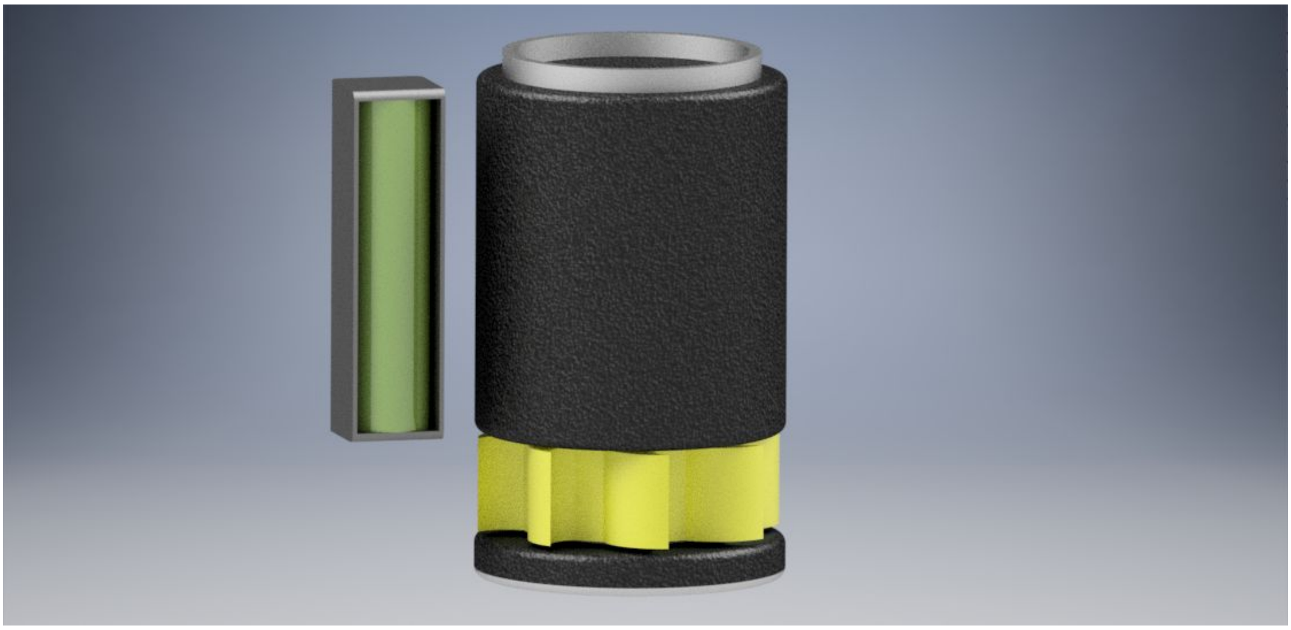


Diagram 1, electronic components

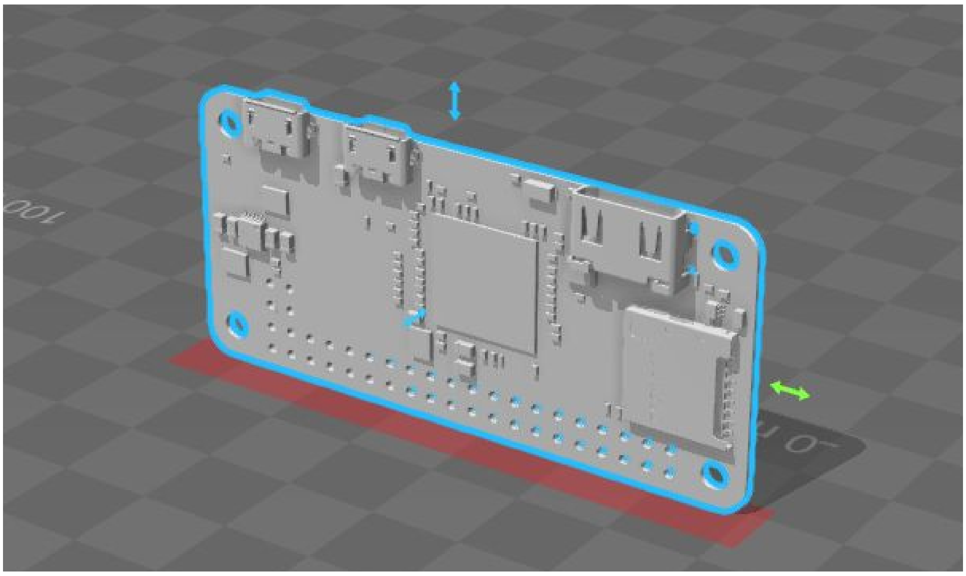
2.2 Mechanical/ structural design

3D model

We have created a 3D model of our CanSat which contains housing, wind sensor and batteries with theirs housings. We also acquired a model of Raspberry Pi, that will be used in model, when set of all instruments will be finally done.



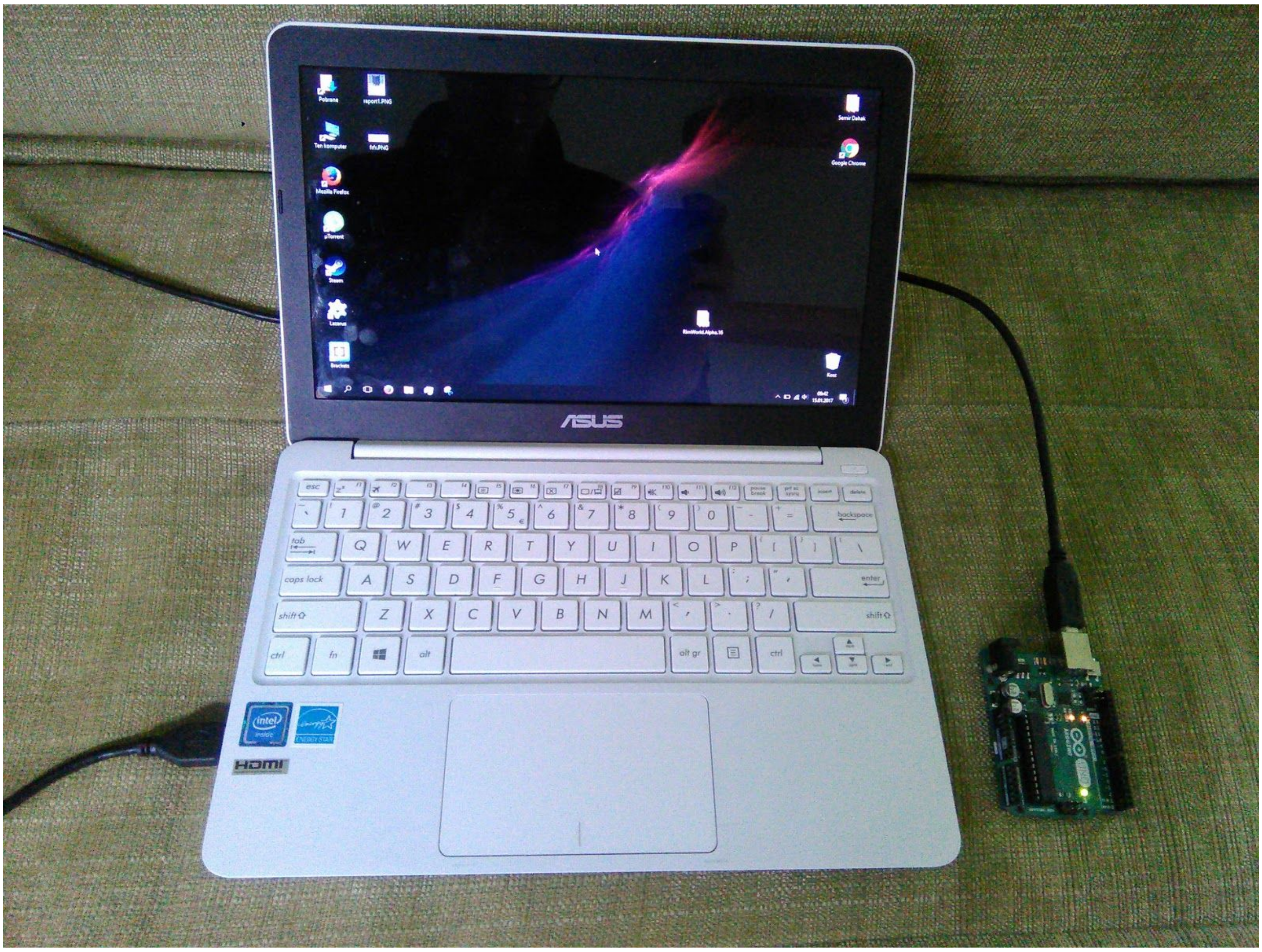
Model 3D of CanSat with wind sensor and battery in housing



Raspberry Pi model

Raspberry Pi model rendered

2.2.1 Ground Station



Minimal base station configuration (without radio receiver connected to Arduino)

Ground Station consists of two main elements: the receiver(described below) and a computer :

Receiver - based on an Arduino module - includes transceiver (to receive data), YAGI antenna (to amplify signal), microSD module (to save received data) and barometer (for reference while computing altitude from atmospheric pressure). Received data is going to be saved on microSD (backup purposes) card and send to the computer via USB or COM port.

2.2.2 CanSat mechanical construction

All electronic elements will be soldered to the PCB (battery holders too), perhaps we will use multiple boards. Empty space inside will be filled with styrofoam balls for additional cussion.

Materials

- **Housing**

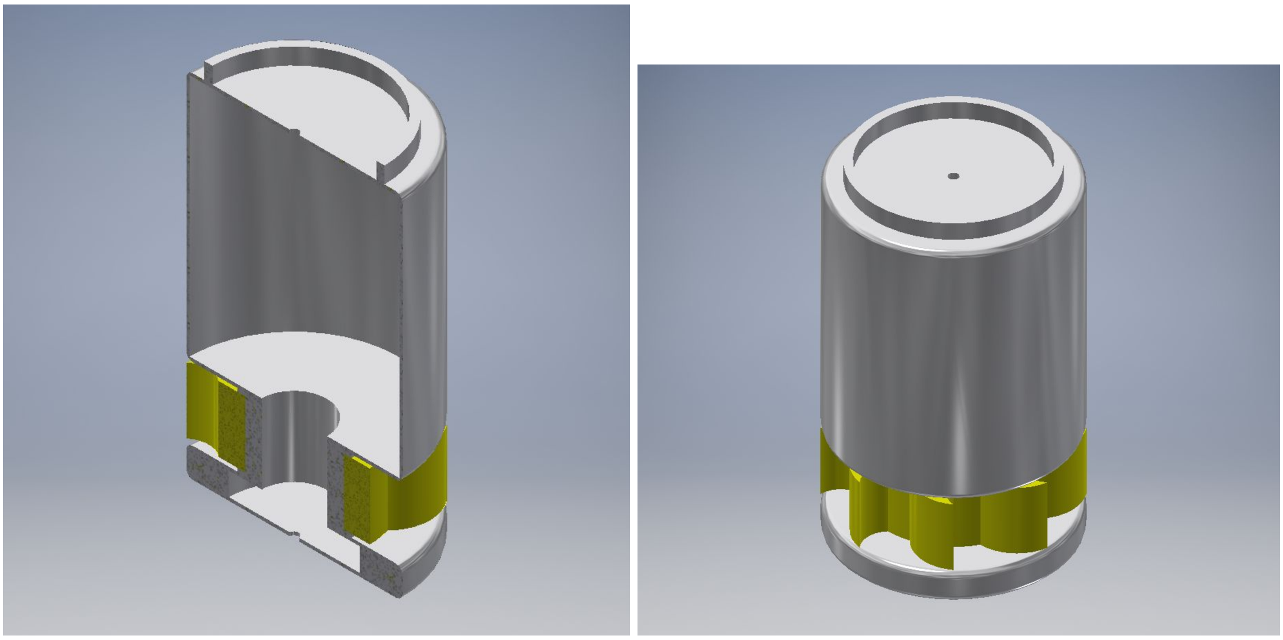
Until recently we reflected on the materials as ABS which is very durable (f.e. most of helmets are made of ABS) and aluminum which is respectively light, but now we are mainly focused on carbon or carbon-aramid fiber (kevlar) that would be formed on the housing. That’s why we are using carbon texture in our 3D models.

At first we thought that it’s not possible to model carbon fiber, but after next research we are going to follow this way.

Material	ABS	Aluminum	Carbon fabric
modeling	easy (3D printing)	milling	time-consuming manual modeling
strenght	very durable	dents undergoing	very durable
weight	+-60g	+-20g	+-30g

- **Cover plates and core**

The top and bottom plates and core of CanSat will be made of aluminum, because of its weight and possibilities to model.



Model 3D without textures

Upon consulting custom carbon fiber casings manufacturer and discussing the price of such solution we have decided to rule it out. Our current conception is to 3D print the outer shell, possibly in pieces requiring further manual assembly or custom aluminium casing.

Component	**Weight [grams]**
Power supply system	~100 (47.5g each battery)
Housing	~30
Raspberry Pi Zero	9
GPS	3
ALTiMU	1

Camera	3
DC-DC converter	1
Transceiver	5
GPS Antenna	20

Table 1, components weight

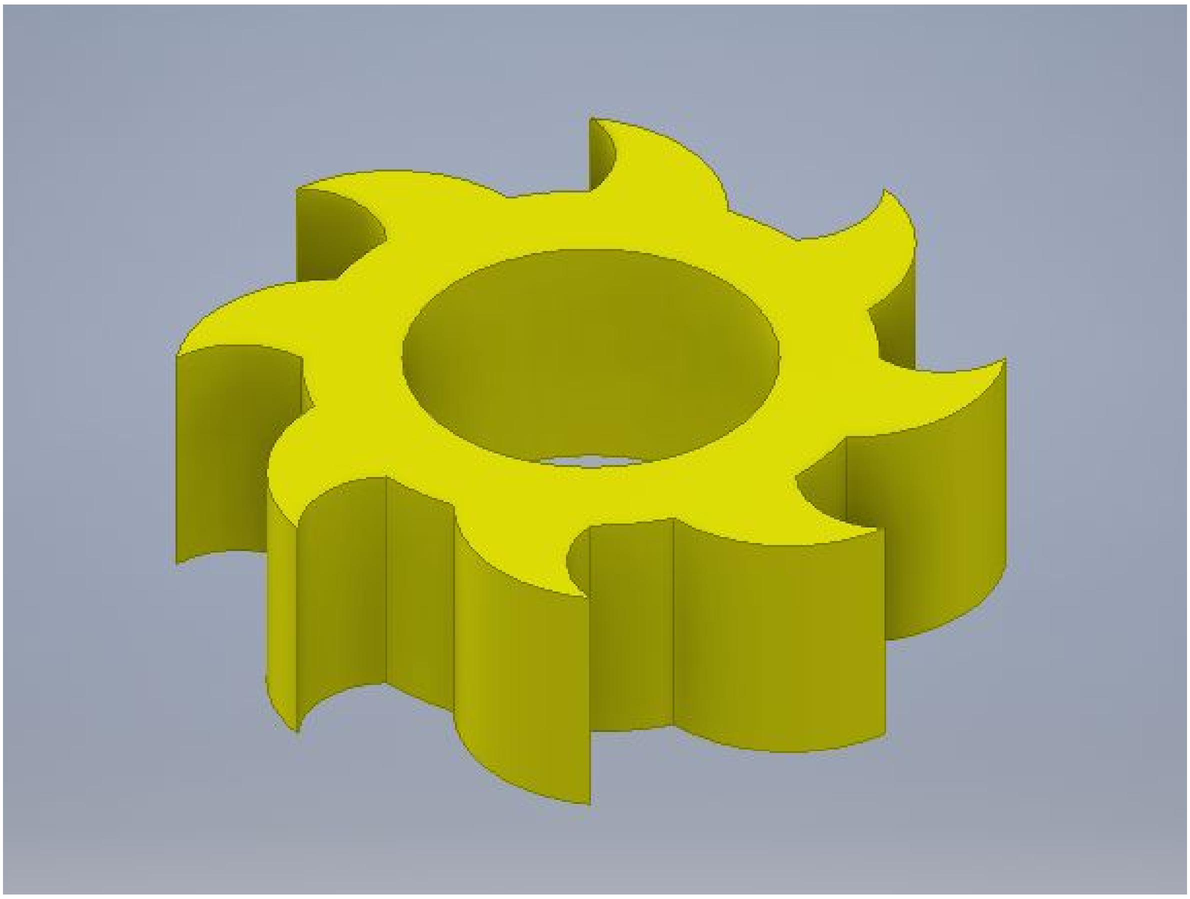
Weight of our CanSat certainly won't exceed the limit of 350g, additional weights might be added to final construction to ensure proper descend velocity.

2.2.3 The wind speed sensor

During the landing, information about speed of the wind might be useful. In the Internet, there are many wind sensors to buy, but their mechanical structure does not fit our project - they are too big and do not withstand excessive acceleration of rocket launch. We decided to develop our own sensor.

Our sensor is going to be a ”ring” driven by the wind. Inside of the CanSat, optical encoder will collect data about its rotation speed. The central computer (Raspberry Pi Zero), will store and transmit abovementioned data and base station will compute accurate wind speed including CanSat's drift with the wind.

First prototype have been developed. We have taken into account size of the CanSat, as well as we have made its blades more aerodynamic, each one has bigger surface and is more rounded.



The wind speed sensor projected in Autodesk Inventor

The wind speed sensor and can of cola for comparison

As the method of measuring "ring" angular velocity we have chosen an optical encoder. We consider it the best solution because it greatly decreases need to use other mechanical elements. Inside, the "ring" will be painted in black and white stripes consecutively (change of the reflection factor means rotation by known angle).

2.3 Electrical design

2.3.1 Electronic elements

The following list includes electronics elements which are going to be inside of our CanSat.

(Detailed parts names/numbers with links can be found in Attachment 1 at the end of this report)

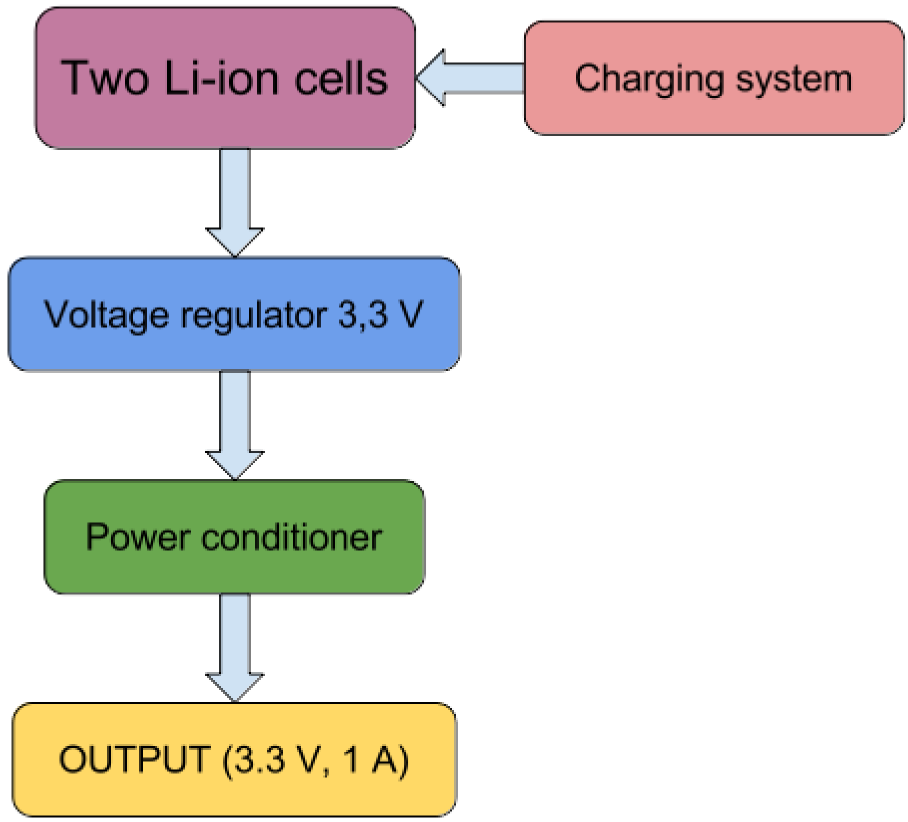
- Raspberry Pi Zero - main computer, collects data from sensors, passes them to the transceiver and save on the microSD card,
- Thermometer Adafruit SHT31 - measuring the temperature,
- Optical encoder Pololu - measuring the angular velocity of the wind sensor,
- Pololu AltIMU-10 v5 - includes accelerometer, gyroscope, magnetometer and barometer. Acceleration measurement will be used in the algorithm of noise reduction from the wind speed sensor,
- Sparkfun GPS Venus with additional antenna - to locate CanSat after landing and pinpoint place of the heavy landing, as well as measure horizontal speed of the probe,
- Raspberry Pi Zero Camera - we are planning to take pictures of the terrain from big height, camera is going to be fixed on the bottom of the CanSat,
- Li-Ion batteries Panasonic NCR-18650B - power supply system,
- DC-DC converter 3.3V S7V8F3 - part of the supply system, Raspberry Pi Zero and others electronic elements require 3.3V to operate,
- Power conditioning system - system including resistors, capacitors and diodes to make sure the current is stable,
- Transceiver HC-12 or SX1276 - to send data to the ground station.

We plan to include charging modules inside our CanSat, allowing quick battery recharge without a need of removing them from their holders. Both size and weight of them does not greatly attribute to parameters of the whole CanSat and greatly increases its reusability.

We are currently in the process of making the final PCB layout in Eagle application.

2.3.2 Battery & power supply

Supply system



After long search for the best power supply, we decided to use Li-Ion cells.

We noticed that the same type of batteries is popular in powerbanks. First we wanted to use just the whole system of powerbank (cells and electronics), but now we have figured that better solution will be to use separate Li-Ion cells.

Batteries used in our design are four type 18650 Panasonic NCR-18650B Li-Ion cells with nominal capacity of no less than 3250mAh each and nominal voltage of 3.6V. One cell should be sufficient, but extreme conditions may affect battery capacity, therefore we are taking two onboard.

Our estimated maximum flow current has been rated at no more than 1A and we've decided to use components powered with 3.3V DC. Therefore our choice was to use two step-up/step-down voltage regulators each rated at no less than 750mA of output current in a pessimistic case of batteries hitting their minimum voltage which is rated at 2.5V. Following table represents estimated power usage in a condition of maximum performance.

Component	**Amperage rating [mA]**
Raspberry Pi Zero v1.3	250
GPS module	90
Raspberry Pi Camera Module v2	250
Radio communication module	120
Wind speed sensor (optical encoder)	12
ALTiMU	6
Thermometer	<1
TOTAL	~730

Table 2, power consumption estimation

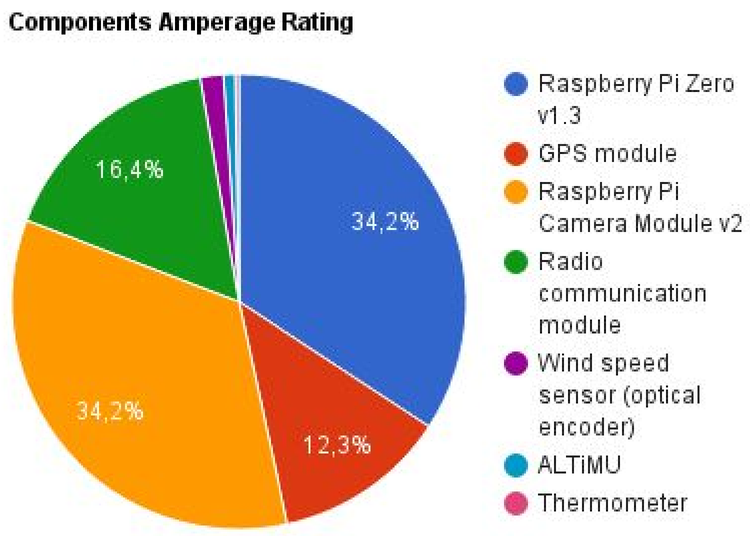


Chart 1, components power consumption percentage share

All estimates are based on official component manufacturer specification with exclusion of Raspberry Pi Zero computer module which power rating we have measured ourselves as data provided by the Raspberry Pi Foundation were based on measurements with both USB accessories and HDMI port powered. During the test we've figured that Raspberry Pi had a greatest load while decoding compressed video with passing the image over X enabled SSH, this resulted in the abovementioned (table) value.

2.3.3 Radio communication system

Due to high equipment availability, suitable wavelength legal limitations and contest rules we have decided to use 433MHz band for communication with our CanSat. It's designated as LPD433 which stands for Low Power Device 433MHz and is meant to be used for amateur radio applications requiring relatively long range. This frequency corresponds with 70 cm wavelength therefore allowing usage of 70 cm, 35 cm and 17 cm long antennae which are optimal considering CanSat dimensions. Our experimental idea is to embed a 70 cm long antenna in a parachute with a corresponding diameter. As we gather CanSat parts we will begin testing of different antennae shapes and wires used to produce it in order to estimate an optimal solution both in terms of range and construction, considering flexibility and least interference with parachute itself. A quarter wavelength antenna will also be a part of our communication set.

Unfortunately we encountered a serious issue with our offshore supplier of radio modules. What appears to be a package lost in transit resulted in almost a two month long delay in testing.

Not discouraged by this problem we decided to buy a pair of modules from local supplier regardless their rather poor specification (in comparison with delayed ones) we managed to perform testing in suburban location Brzezie north west of Cracow. These tests didn't go well as we only managed to receive signal from about 220m. Further data analysis revealed that the issue originated from wrong transceiver settings and previously not spotted metal obstruction as well as hardware design issue resulting in radio module not being fed enough power.

Together with said modules our onboard antennas got stuck or lost in transit. The abovementioned test had to be performed with poor quality spring antennas 1/16th of the wavelength. We presume this was also a key factor in our failure.

This taught us that proper supply chain is essential to ensure project success. Also such issues lead us to perform some abstract thinking. Fruit of which was a handmade antenna made one night. By slowly bending wire we managed to fit a total of above 2m of it in a 8.5 cm spring antenna. further testing….

It seems important to note that our testing capabilities in Cracow are obstructed by low temperatures and dangerous air pollution. We had to take advantage of -20C weather and exposed our equipment to these conditions. This test was a full success as no problems occurred.

On the receiving end we plan to use a commercial grade CDMA-10/TNC/15 antenna with the same radio module as in our CanSat. Modules on transmitting and receiving end will be controlled with on-board Raspberry Pi and ground based Arduino.

Despite hardware capability for now we don't plan to maintain nor even attempt a 2-way communication. Further testing will indicate whether CanSat antennas will allow receiving of signal with reasonable quality and at reasonable range.

2.4 Software design

2.4.1 CanSat software (based on RaspberryPi Zero)

CanSat software (based on RaspberryPi Zero)

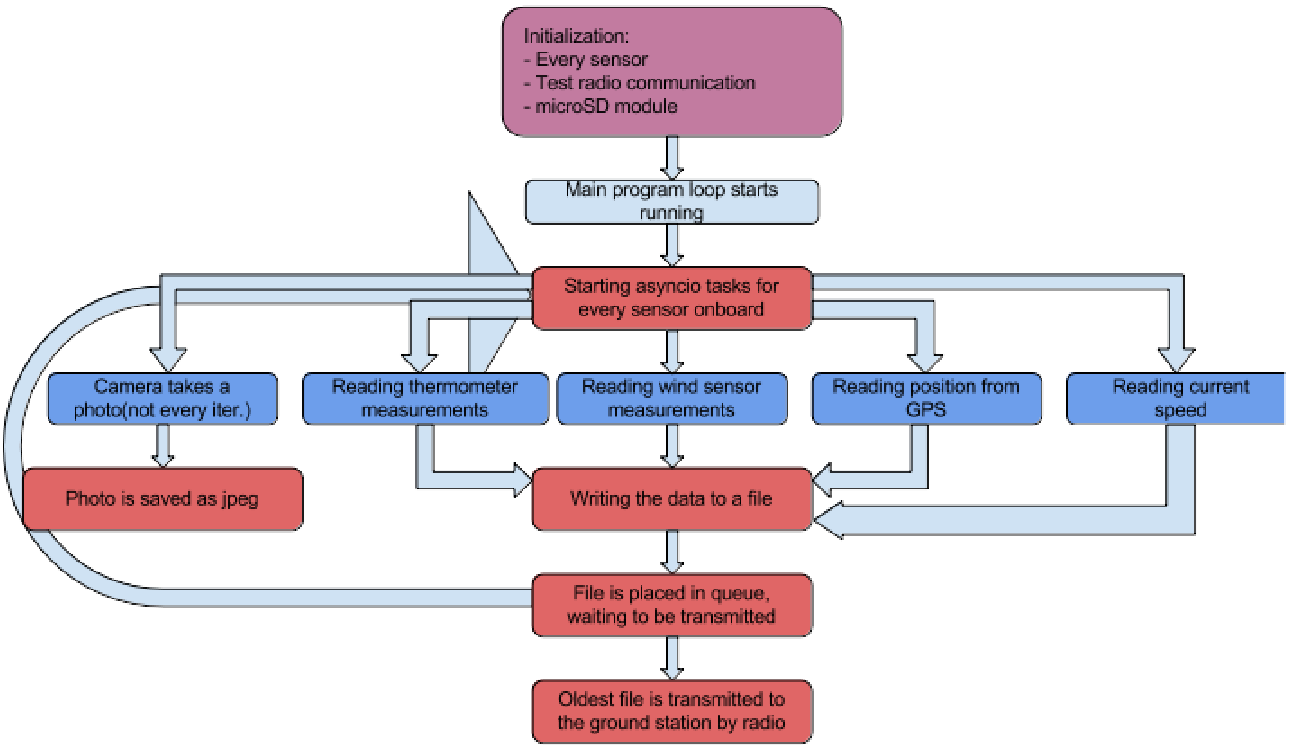


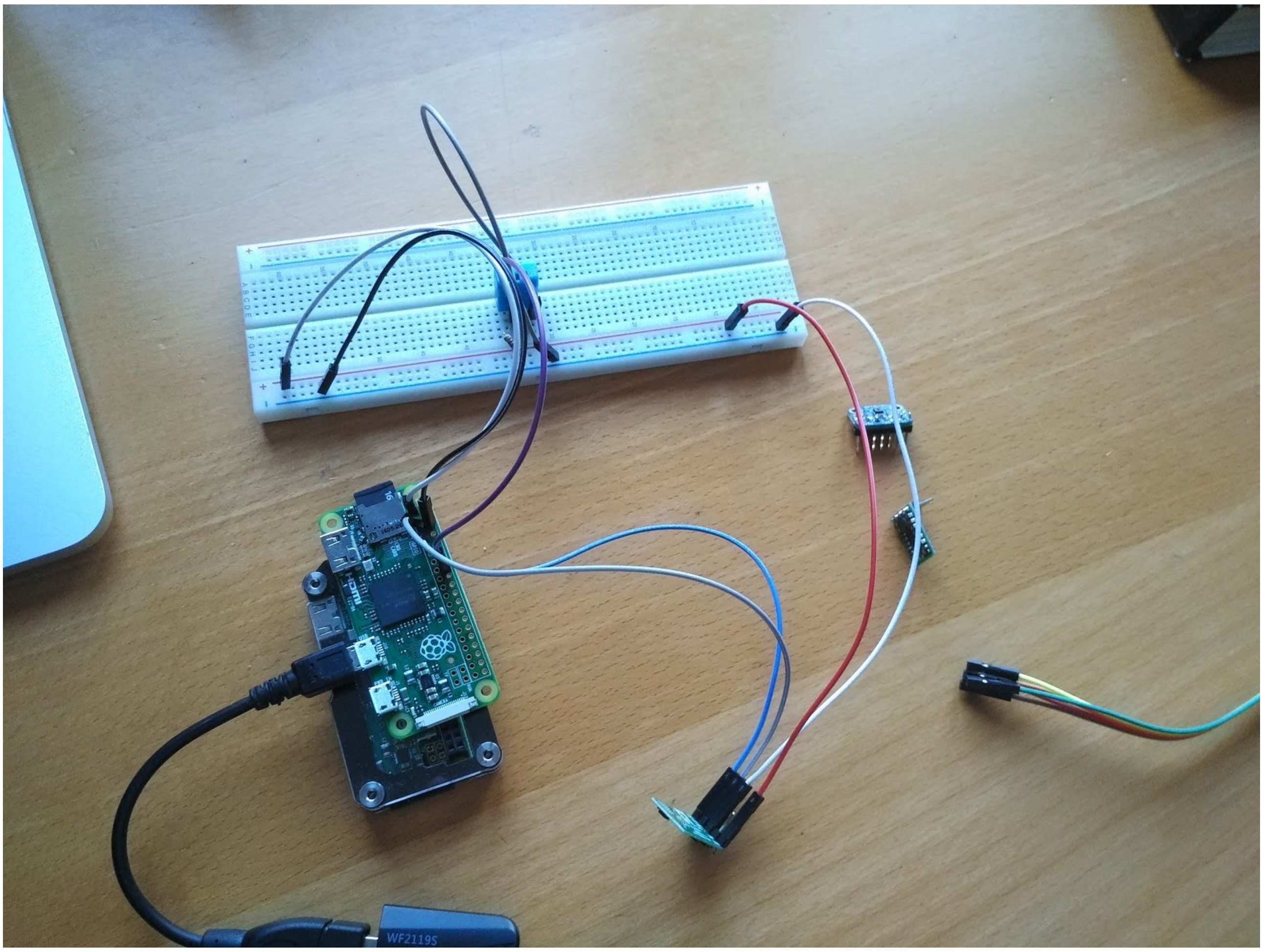
Diagram 3, CanSat software flow diagram

Our concept for on board software is to use benefits of Raspberry Pi OS - Raspbian. Since, this it is a full blown linux distribution, we would like to take advantage of systemd services. The idea is to write a small separate service for every sensor. This mechanism will prevent us from losing data due to unexpected application crashes (possible segfaults caused by cosmic radiation etc). When systemd service notices that certain sensor app is down, it will automagically starts it up again. We also consider making some sort of observer, which will ping (or perform another healthchecking test) every sensor application once a while.

Our Raspbian image is a specially lightened version of the light one. This version comes without a graphical environment as it's supposed to used as a headless (without a display) computer. Additionally, the HDMI port and the LED is disabled. For now the usb port is enabled because, we're using a WIFI antenna to download all the needed software and conduct development. After the development phase is done, these things will be disabled as they're just using battery power and CPU cycles for nothing.

The small programs will be written in Python3 and are going to be focused around main loops which log data from the sensors. They're meant to be as simple as possible, which reduces the risk of bugs and crashes. Inside the loop, we read the input from a sensor and write it to a bash fifo queue. One program will occasionally take a photo of the land beneath. All data gathered will be stored on an onboard SD Card. After the parachute deployment the Cansat will begin transmitting already accumulated data and begin live parameters transmission. We aim to achieve a data resolution of 1 second. The images will be taken in 10 second intervals. Following landing, our CanSat will begin a loop transmission of all parameters gathered throughout the flight to ensure missing packets will be retransmitted. Further collection of onboard storage may optimistically turn out to be required only to acquire photos taken during flight. A micro SD card proves to be a reliable method of storage of both onboard code and gathered data as contemporary devices of this type are resistant to fluctuating temperature, water and even mild radiation. Their small size practically does not attribute to CanSat mass nor limit space available inside. Also a maximum capacity of many hundreds of Gigabytes is certainly above our requirements.

We decided to opt for a simple radio connection, and not implementing a TCP/IP stack, because the amount of benefits from such a solution, doesn't outweigh the disadvantages.



Raspberry Pi Zero with the thermometer and accelerometer during testing.

All avionics and environmental data gathered have a simple form of an integer which takes 64 bits per number, together with a timestamp we expect about 0.5KB of data acquisition per second (excluding photos). Such excess may even allow continuous retransmission of legacy data. Connection stability testing will certainly answer whether such solution will prove useful.

Camera used in our design is a dedicated Raspberry Pi Camera module version 2 which uses an 8Mpx sensor. Assuming 8 bits per each of three channels an expected weight of each photo is 2MB, thanks to JPEG compression. Such a number is not a challenge for our onboard storage (provided a decent interval of about 10 seconds) but certainly exceeds our prognosed radio capabilities. Happily we've decided to put a

Linux-driven computer onboard and an attempt to highly compress and transmit an image is certainly within our technological capabilities. Mainly data rate tests will decide whether we will attempt to do so.

2.4.2 Used IDE's and languages

Ground station:

Language: FreePascal

IDE: Lazarus

Receiver:

Language: C++

IDE: Arduino IDE

CanSat (Raspberry Pi Zero):

Language: Python3, Bash

2.5 Recovery system

At the beginning of planning the manners in which CanSat will be recovered, two methods were taken into account:

- parachute (necessary)
- airbag (optional)
- BLE beacon (experimental)

Parachute as an indispensable and main element of recovery system was analyzed in several versions and combinations. It is able to reduce the speed of the CanSat efficiently and stabilize the descend simultaneously. The most important test was one, which was meant to determine the optimal shape of the parachute canopy (pic. 2.5.1). More informations about tests and their results are in section 3.3 "Describe tests" .

Airbag was an option that we had to reject because of many factors, these are the most important ones:

- There are no airbags widely available that we can afford to use (due to our budget, maximum weight and space inside the CanSat);
- It's not possible to build adequate airbag without an adequately supplied laboratory;

- In our CanSat we are going to use a camera module, which is located underneath the Can, therefore it poses a serious limitation in terms of placing such an airbag.

The way in which we want to fix the parachute to the CanSat is to connect top loop of the core with loop of a paracord by a climbing carabiner (strength 23 kN).

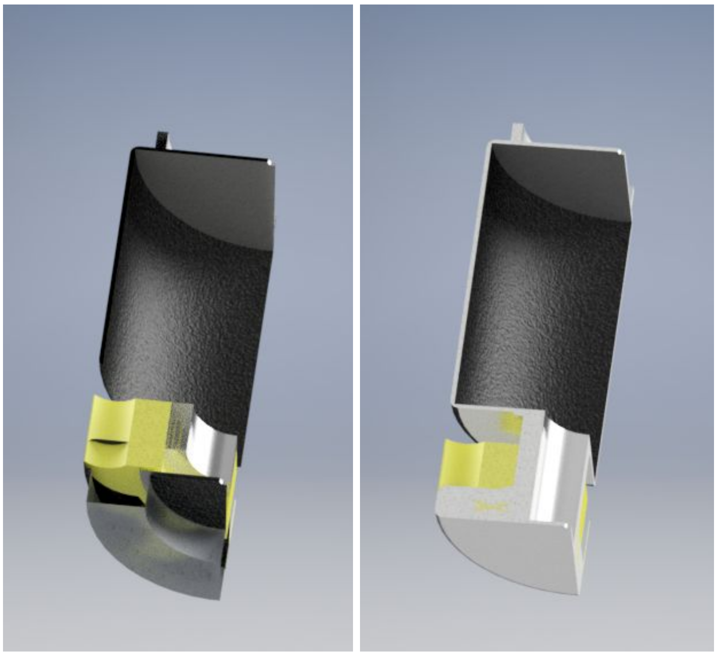
Assuming a height of 3000m and speed of 7 m/s to 11 m/s , maximum fall time is about 7 minutes and minimum time is about 4 minutes 30 seconds.

Photo 3, handmade parachutes

BLE beacon is in our consideration as additional help to fine locating our CanSat. Together with GPS providing rough estimate of the location, this new technology will allow instant fine locating of our CanSat by the ground crew allowing easy, fast and successful recovery.

Airbag

We also have been thinking about a kind of an airbag which could be placed underneath the can in order to reduce the impact force when in contact with ground after flight. In case of using airbag, we would have to reduce space in the main part of the CanSat. Finally, there is little likelihood of developing this idea.



Section of the can with the wind sensor and an airbag underneath and section without airbag

Choosing a parachute

We have established contact with the German experts from **Raketenmodellbau Klima GmbH-** a company dedicated to models of rocket. They told us, that for our project 70 cm (diameter) parachute is most suitable, but we have to add hole in a canopy by ourselves. As a result, we have the support of professionals in the aeronautics industry.

Dear Bartlomiej,

Thank you for your inquiry.
It is always nice for us to hear from young people getting involved in to rocketry.

For the weight of your payload I would normally recommend our 70 cm parachute,
but that does not have a hole in the canopy, so you would have to add that yourself.

The 125 cm parachute is the smallest size with a hole in the canopy.

Tension strength is no problem in my opinion, as our chutes are made of high quality materials.

Hope this info is of some help.
Let me know if you have further questions.

Best regards
Floyd Wurster
Product Manager

KLIMA

Raketenmodellbau Klima GmbH,
An der Laugna 1,
D-86494 Emersacker
Germany

Tel +49 8293 1734
Mobil(e) +49 151 64198412
Fax +49 8293 7815

E-mail from company's Product Manager

2.6 Ground support Equipment

2.6.1 Ground station

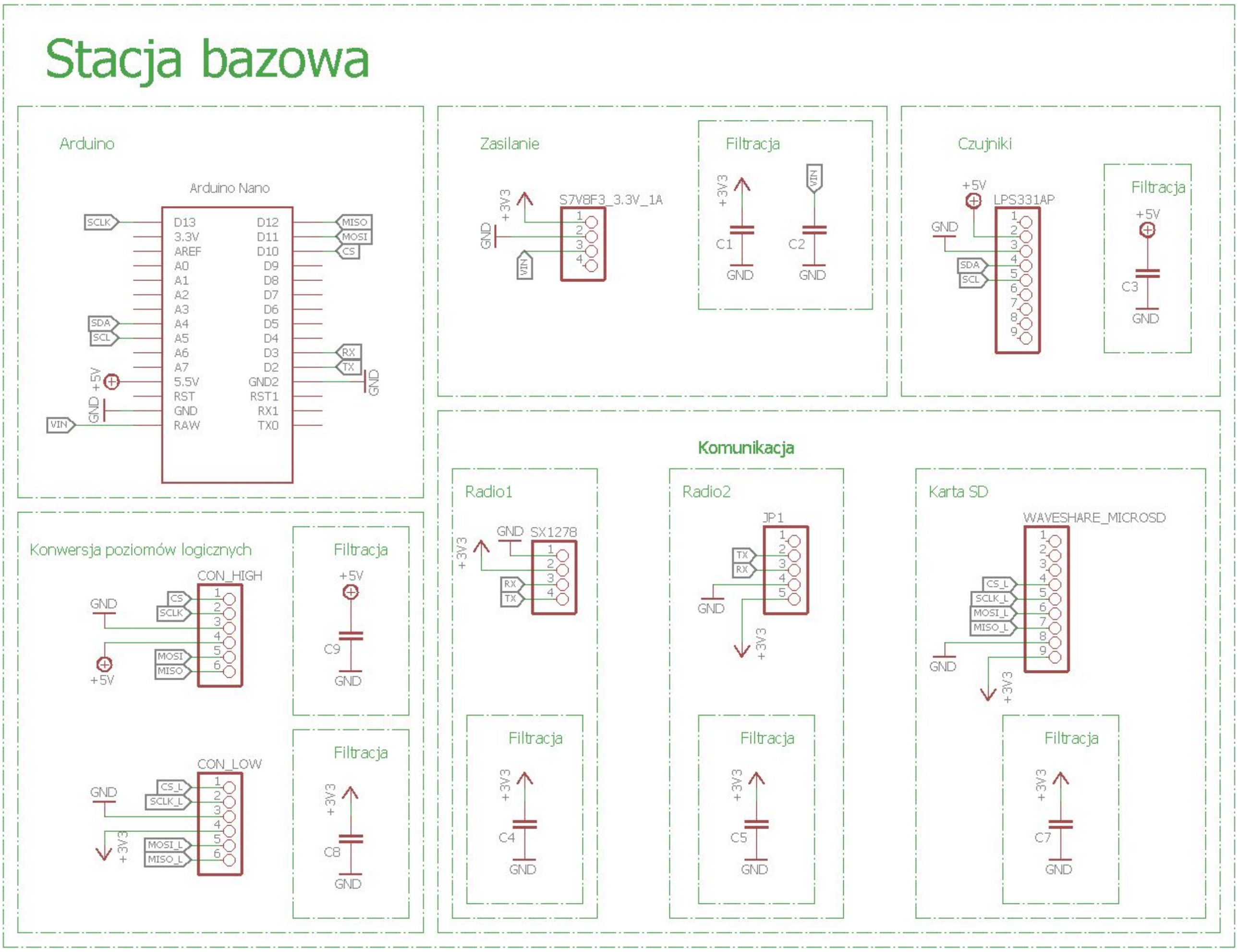
By this term we consider a notebook equipped with receiving module and software for analyzing data gathered and transmitted by our CanSat.

Radio receiver

Software

Ground station

Radio data receiver is based on the Arduino Nano (we choose this one instead of Arduino Uno just because of its size). Radio receiver module (HC-12 or SX1278), with soldered female SMA connector, as an antenna input. Received data are being read by Arduino (UART communication), then (if possible) Arduino converts it to JSON object and send to computer. Whole system is secured in case of computer's error - data are saved on the SD card, we can get them later and manually analyze. For better measurements of CanSat's altitude, we also measure pressure in ground station, using Pololu LPS331AP, which gives us measures of temperature and altitude, communicates over I^2C. To hold the SD card we use Waveshare module, which communicates over SPI interface. MicroSD cards are working only in 3.3V, that's why we are using also logic level converter. First we were thinking about switching Arduino Nano for 3.3V (whole logic), but this solution looks easier. We have already designed schematic of a PCB in Eagle program.



Ground station software is written in Arduino IDE. The code is available on our git repository (Github), a pull request has been submitted but it is still waiting for proper code review by the team. It contains an algorithm to calculate current time, based on the time received only once from the computer but it won't be used in the next version of the program. This might be useful in case of technical problems with computer, but unfortunately, Arduino's function millis(), which should return time in milliseconds from the beginning of the running program, is not as precise as we thought - translation in the time would be too big.

Ground station software consists of two main parts:

- Software in the radio receiver device,
- Desktop application for analysis and display of gathered data.

Diagram 4, radio receiver algorithm

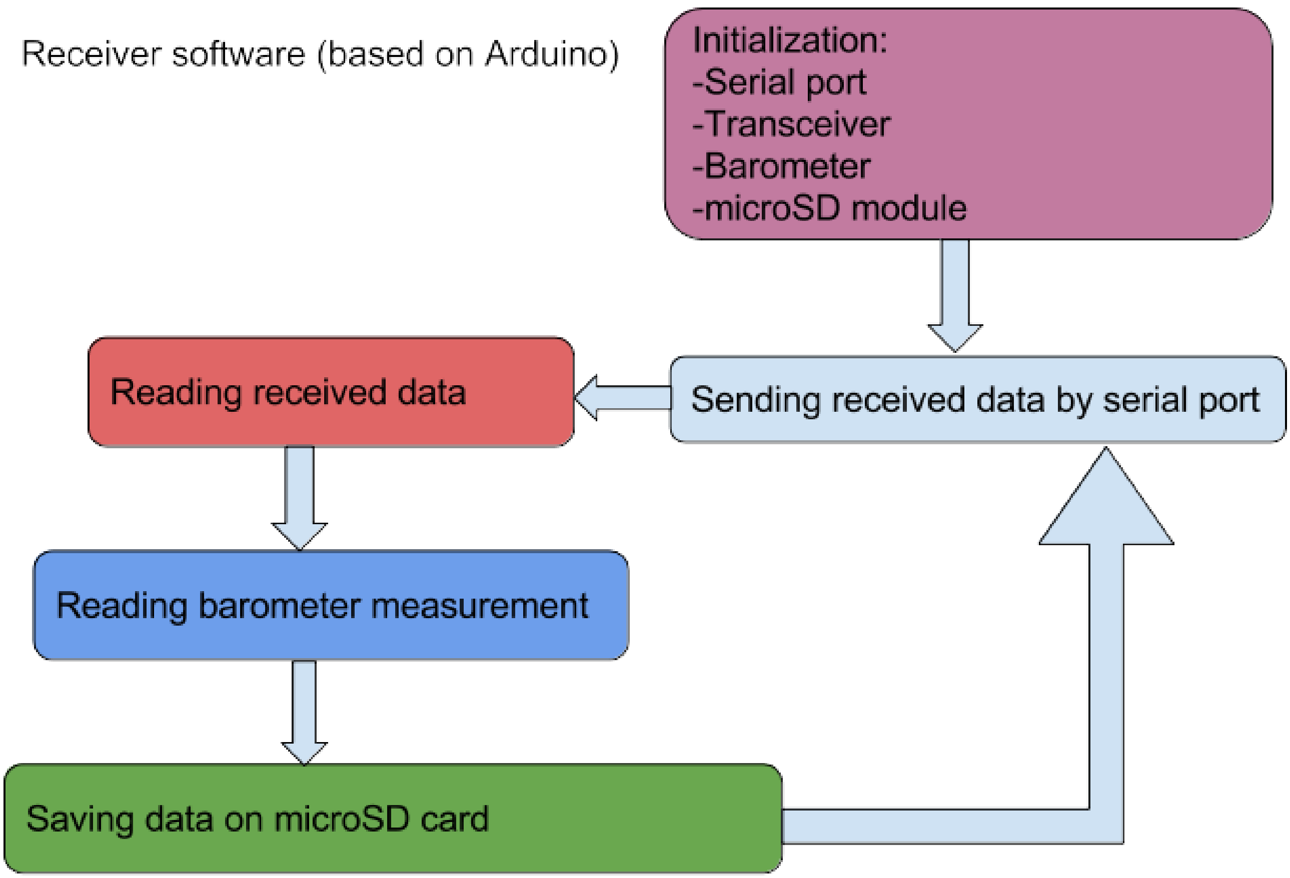
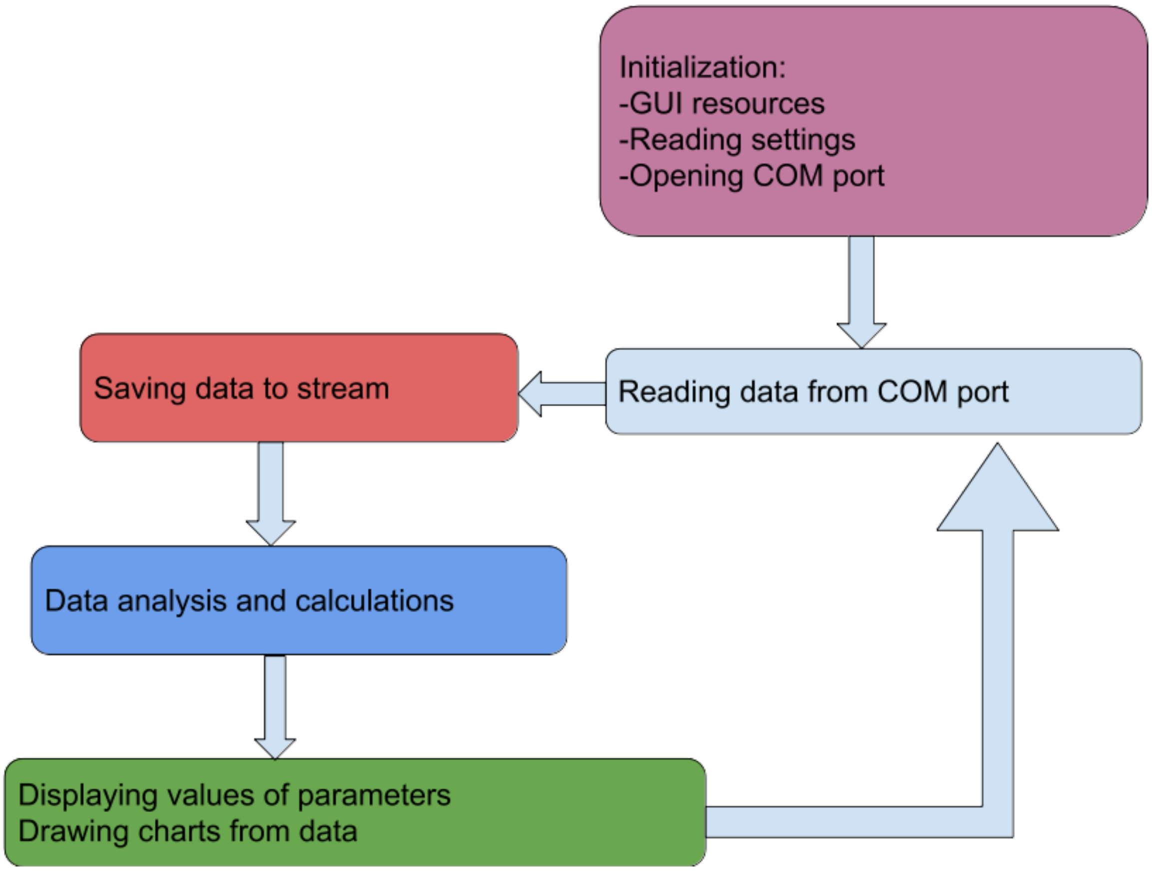


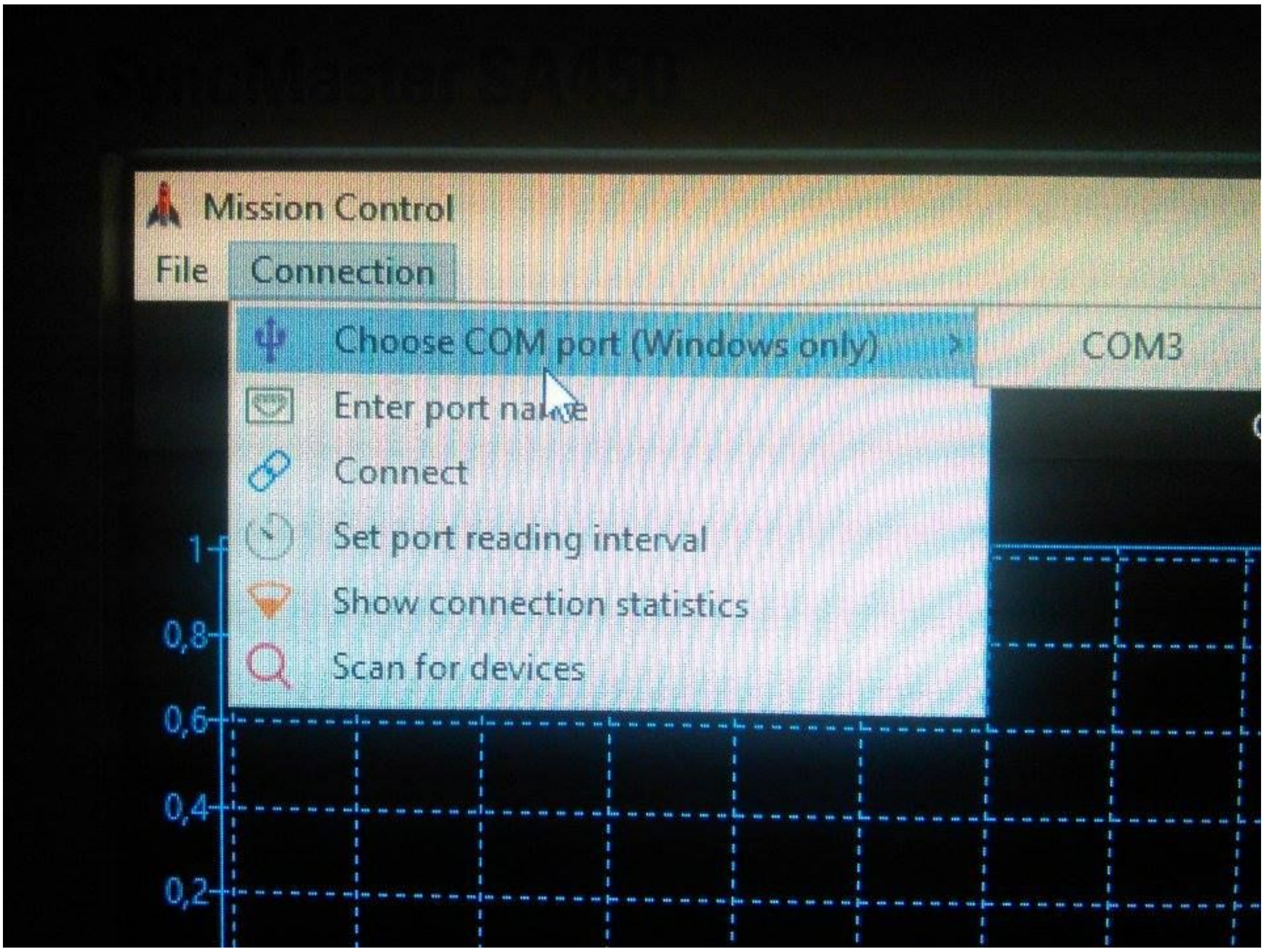
Diagram 5, Desktop application algorithm



Ground station desktop application

Initialization

After launching, the application prepares all structures for handling received data, load last configuration, prepare charts library and it waits for user to choose COM port, to which base station is plugged.



Connection menu

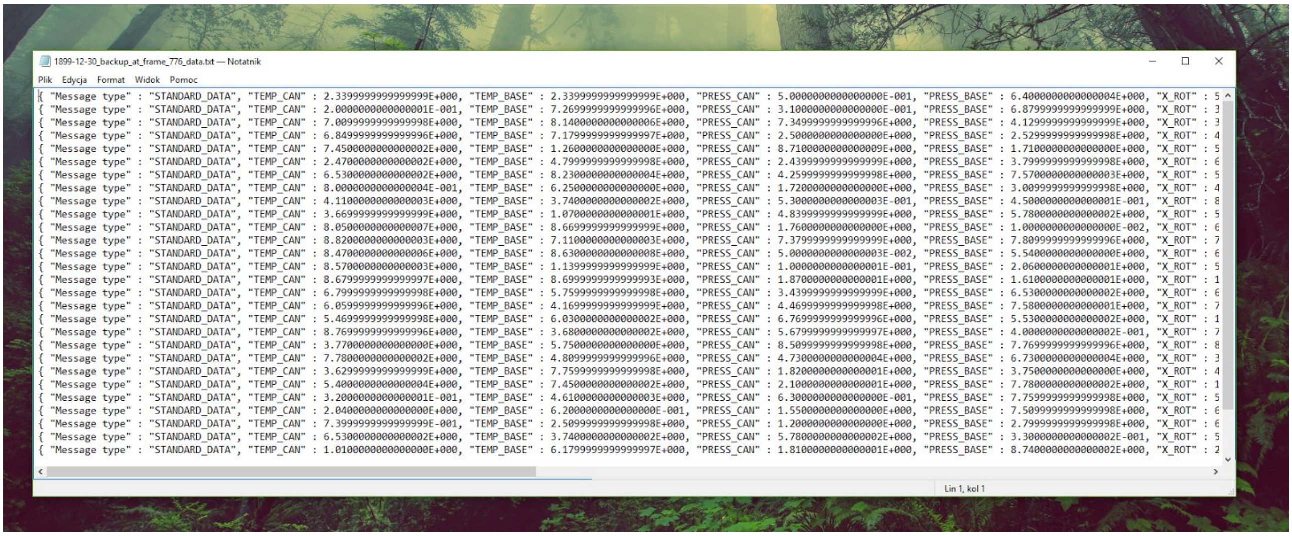
Reading data from COM port

Data sent by the radio receiver to the COM port via USB cable is being read by the application. Once user chooses which COM port to use (i.e. COM3, dev/tty3, etc.), the connection is established.

Data is transmitted in JSON frames, which structure is described in additional document attached to this report.

Saving data to stream

Received data is being saved to stream and proper data structures. After CanSat landing the file with all received data will is created, while every 5 frames a backup file is made. If application crashes accidently we can restore received data (also other backup solutions have been implemented: data can be restored from CanSat SD Card and Base Station radio receiver SD Card (from Arduino)).



1899-12-30_backup_at_frame_776_data.txt — Notatnik

Plik Edycja Format Widok Pomoc


```
{ "Message type" : "STANDARD_DATA", "TEMP_CAN" : 2.3399999999999999E+000, "TEMP_BASE" : 2.3399999999999999E+000, "PRESS_CAN" : 5.0000000000000000E-001, "PRESS_BASE" : 6.4000000000000004E+000, "X_ROT" : 5
{ "Message type" : "STANDARD_DATA", "TEMP_CAN" : 2.0000000000000001E-001, "TEMP_BASE" : 7.2699999999999996E+000, "PRESS_CAN" : 3.1000000000000000E-001, "PRESS_BASE" : 6.8799999999999999E+000, "X_ROT" : 3
{ "Message type" : "STANDARD_DATA", "TEMP_CAN" : 7.0099999999999998E+000, "TEMP_BASE" : 8.1400000000000006E+000, "PRESS_CAN" : 7.3499999999999996E+000, "PRESS_BASE" : 4.1299999999999999E+000, "X_ROT" : 3
{ "Message type" : "STANDARD_DATA", "TEMP_CAN" : 6.8499999999999996E+000, "TEMP_BASE" : 7.1799999999999997E+000, "PRESS_CAN" : 2.5000000000000000E+000, "PRESS_BASE" : 2.5299999999999998E+000, "X_ROT" : 4
{ "Message type" : "STANDARD_DATA", "TEMP_CAN" : 7.4500000000000002E+000, "TEMP_BASE" : 1.2600000000000000E+000, "PRESS_CAN" : 8.7100000000000009E+000, "PRESS_BASE" : 1.7100000000000000E+000, "X_ROT" : 5
{ "Message type" : "STANDARD_DATA", "TEMP_CAN" : 2.4700000000000002E+000, "TEMP_BASE" : 4.7999999999999998E+000, "PRESS_CAN" : 2.4399999999999999E+000, "PRESS_BASE" : 3.7999999999999998E+000, "X_ROT" : 6
{ "Message type" : "STANDARD_DATA", "TEMP_CAN" : 6.5300000000000002E+000, "TEMP_BASE" : 8.2300000000000004E+000, "PRESS_CAN" : 4.2599999999999998E+000, "PRESS_BASE" : 7.5700000000000003E+000, "X_ROT" : 5
{ "Message type" : "STANDARD_DATA", "TEMP_CAN" : 8.0000000000000004E-001, "TEMP_BASE" : 6.2500000000000000E+000, "PRESS_CAN" : 1.7200000000000000E+000, "PRESS_BASE" : 3.0099999999999998E+000, "X_ROT" : 4
{ "Message type" : "STANDARD_DATA", "TEMP_CAN" : 4.1100000000000003E+000, "TEMP_BASE" : 3.7400000000000002E+000, "PRESS_CAN" : 5.3000000000000003E-001, "PRESS_BASE" : 4.5000000000000001E-001, "X_ROT" : 8
{ "Message type" : "STANDARD_DATA", "TEMP_CAN" : 3.6699999999999999E+000, "TEMP_BASE" : 1.0700000000000001E+000, "PRESS_CAN" : 4.8399999999999999E+000, "PRESS_BASE" : 5.7800000000000002E+000, "X_ROT" : 5
{ "Message type" : "STANDARD_DATA", "TEMP_CAN" : 8.0500000000000007E+000, "TEMP_BASE" : 8.6699999999999999E+000, "PRESS_CAN" : 1.7600000000000000E+000, "PRESS_BASE" : 1.0000000000000000E-002, "X_ROT" : 6
{ "Message type" : "STANDARD_DATA", "TEMP_CAN" : 8.8200000000000003E+000, "TEMP_BASE" : 7.1100000000000003E+000, "PRESS_CAN" : 7.3799999999999999E+000, "PRESS_BASE" : 7.8099999999999996E+000, "X_ROT" : 7
{ "Message type" : "STANDARD_DATA", "TEMP_CAN" : 8.4700000000000006E+000, "TEMP_BASE" : 8.6300000000000008E+000, "PRESS_CAN" : 5.0000000000000003E-002, "PRESS_BASE" : 5.5400000000000000E+000, "X_ROT" : 6
{ "Message type" : "STANDARD_DATA", "TEMP_CAN" : 8.5700000000000003E+000, "TEMP_BASE" : 1.1399999999999999E+000, "PRESS_CAN" : 1.0000000000000001E-001, "PRESS_BASE" : 2.0600000000000001E+000, "X_ROT" : 5
{ "Message type" : "STANDARD_DATA", "TEMP_CAN" : 8.6799999999999997E+000, "TEMP_BASE" : 8.6999999999999993E+000, "PRESS_CAN" : 1.8700000000000001E+000, "PRESS_BASE" : 1.6100000000000001E+000, "X_ROT" : 1
{ "Message type" : "STANDARD_DATA", "TEMP_CAN" : 6.7999999999999998E+000, "TEMP_BASE" : 5.7599999999999998E+000, "PRESS_CAN" : 3.4399999999999999E+000, "PRESS_BASE" : 6.5300000000000002E+000, "X_ROT" : 6
{ "Message type" : "STANDARD_DATA", "TEMP_CAN" : 6.0599999999999996E+000, "TEMP_BASE" : 4.1699999999999999E+000, "PRESS_CAN" : 4.4699999999999998E+000, "PRESS_BASE" : 7.5800000000000001E+000, "X_ROT" : 7
{ "Message type" : "STANDARD_DATA", "TEMP_CAN" : 5.4699999999999998E+000, "TEMP_BASE" : 6.0300000000000002E+000, "PRESS_CAN" : 6.7699999999999996E+000, "PRESS_BASE" : 5.5300000000000002E+000, "X_ROT" : 1
{ "Message type" : "STANDARD_DATA", "TEMP_CAN" : 8.7699999999999996E+000, "TEMP_BASE" : 3.6800000000000002E+000, "PRESS_CAN" : 5.6799999999999997E+000, "PRESS_BASE" : 4.0000000000000002E-001, "X_ROT" : 7
{ "Message type" : "STANDARD_DATA", "TEMP_CAN" : 3.7700000000000000E+000, "TEMP_BASE" : 5.7500000000000000E+000, "PRESS_CAN" : 8.5099999999999998E+000, "PRESS_BASE" : 7.7699999999999996E+000, "X_ROT" : 8
{ "Message type" : "STANDARD_DATA", "TEMP_CAN" : 7.7800000000000002E+000, "TEMP_BASE" : 4.8099999999999996E+000, "PRESS_CAN" : 4.7300000000000004E+000, "PRESS_BASE" : 6.7300000000000004E+000, "X_ROT" : 3
{ "Message type" : "STANDARD_DATA", "TEMP_CAN" : 3.6299999999999999E+000, "TEMP_BASE" : 7.7599999999999998E+000, "PRESS_CAN" : 1.8200000000000001E+000, "PRESS_BASE" : 3.7500000000000000E+000, "X_ROT" : 4
{ "Message type" : "STANDARD_DATA", "TEMP_CAN" : 5.4000000000000004E+000, "TEMP_BASE" : 7.4500000000000002E+000, "PRESS_CAN" : 2.1000000000000001E+000, "PRESS_BASE" : 7.7800000000000002E+000, "X_ROT" : 1
{ "Message type" : "STANDARD_DATA", "TEMP_CAN" : 3.2000000000000001E-001, "TEMP_BASE" : 4.6100000000000003E+000, "PRESS_CAN" : 6.3000000000000000E-001, "PRESS_BASE" : 7.7599999999999998E+000, "X_ROT" : 5
{ "Message type" : "STANDARD_DATA", "TEMP_CAN" : 2.0400000000000000E+000, "TEMP_BASE" : 6.2000000000000000E-001, "PRESS_CAN" : 1.5500000000000000E+000, "PRESS_BASE" : 7.5099999999999998E+000, "X_ROT" : 6
{ "Message type" : "STANDARD_DATA", "TEMP_CAN" : 7.3999999999999999E-001, "TEMP_BASE" : 2.5099999999999998E+000, "PRESS_CAN" : 1.2000000000000000E+000, "PRESS_BASE" : 2.7999999999999998E+000, "X_ROT" : 6
{ "Message type" : "STANDARD_DATA", "TEMP_CAN" : 6.5300000000000002E+000, "TEMP_BASE" : 3.7400000000000002E+000, "PRESS_CAN" : 5.7800000000000002E+000, "PRESS_BASE" : 3.3000000000000002E-001, "X_ROT" : 5
{ "Message type" : "STANDARD_DATA", "TEMP_CAN" : 1.0100000000000000E+000, "TEMP_BASE" : 6.1799999999999997E+000, "PRESS_CAN" : 1.8100000000000001E+000, "PRESS_BASE" : 8.7400000000000002E+000, "X_ROT" : 2
```


Lin 1, kol 1

Example of backup file

Data analysis and calculations

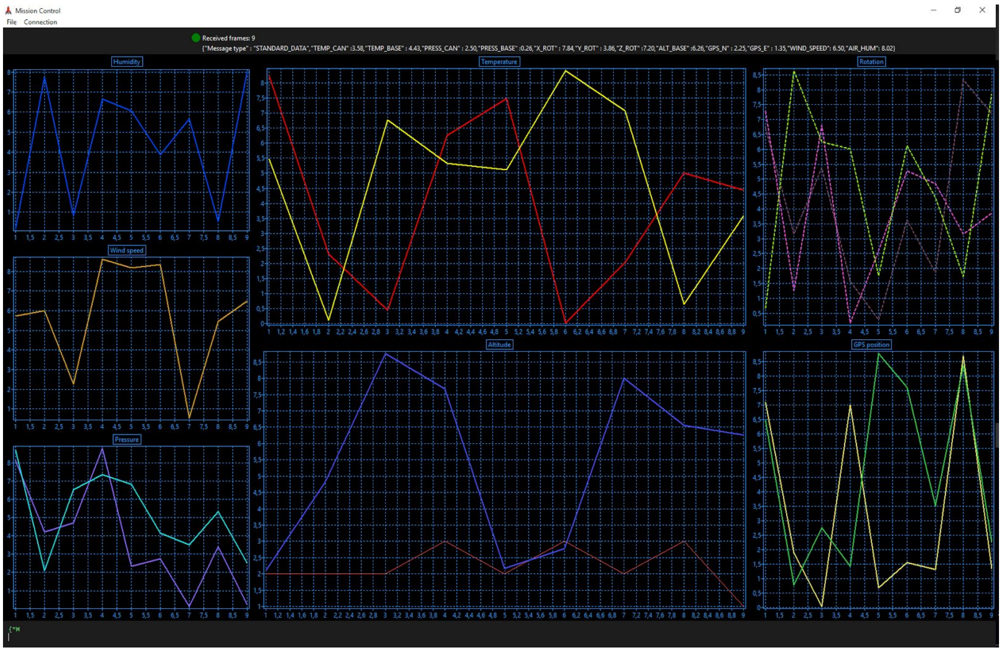
Primary objective of online radio transmission is to acquire measured temperature and pressure and compute CanSat altitude above the ground. It is done with dedicated procedure and displayed on chart.

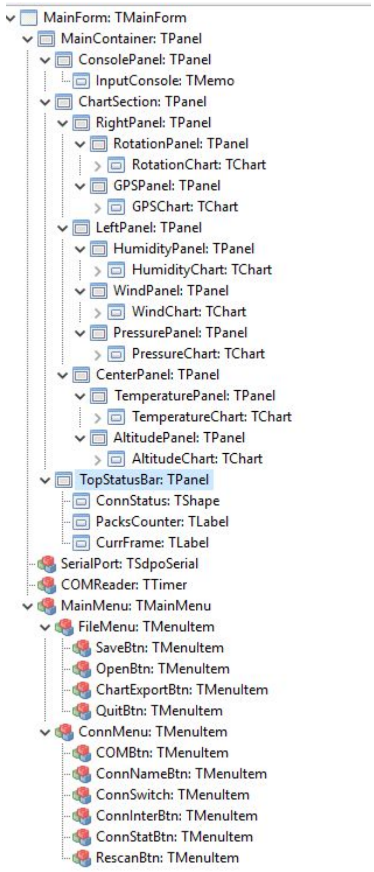
Our secondary objective requirement is to analyze in real-time measurements from additional sensors. They are displayed on the charts alongside with basic parameters. Application is processing and showing conditions for heavy lander's landing at additional panel in application. The measurements are compared with our secondary mission criteria and application indicates whether landing would be safe or not.

Displaying values of parameters and drawing data charts

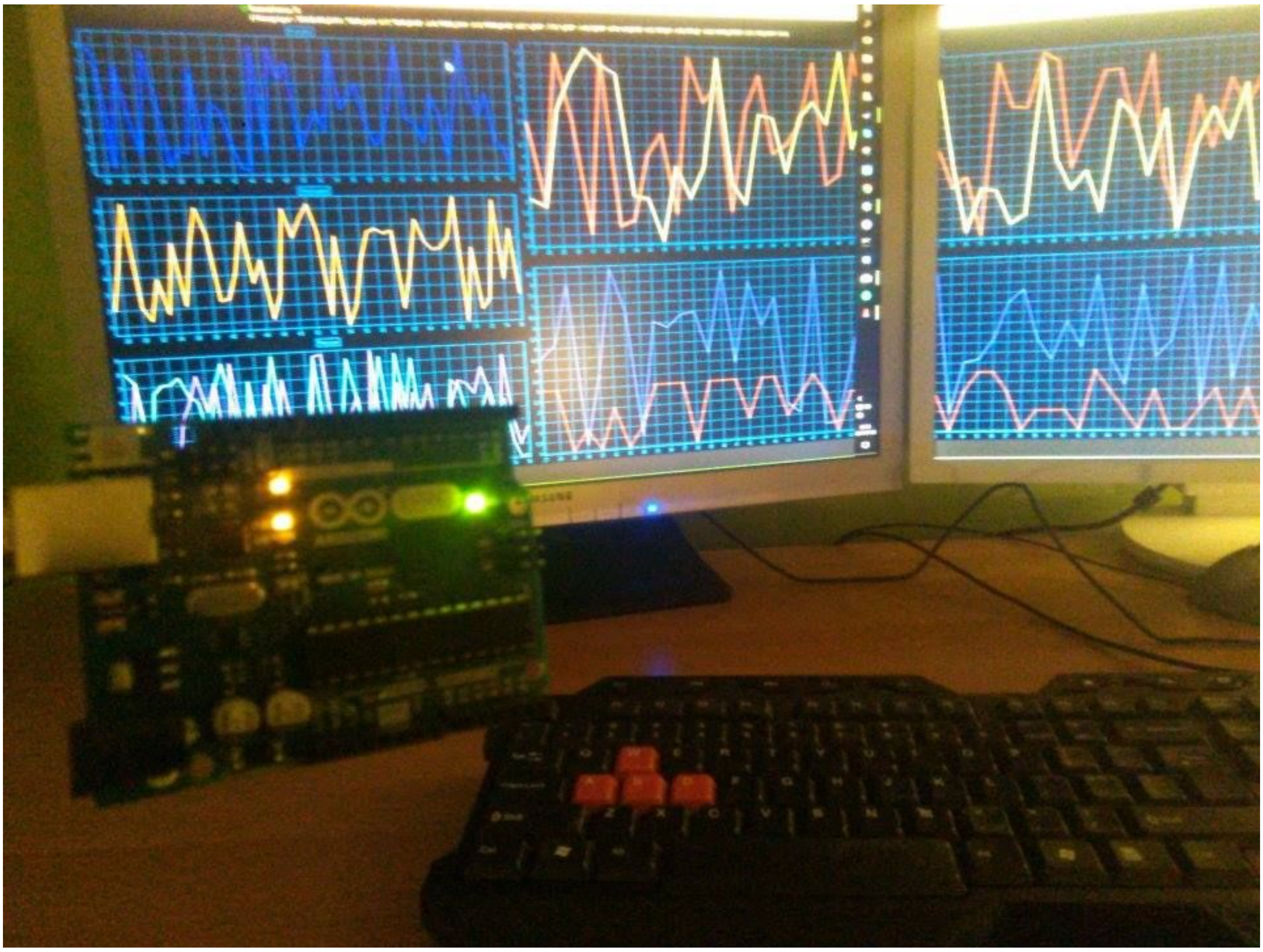
Although standard controls will be used to display parameters values, drawing charts will be optimized for bigger amounts of data.

UI design





GUI controls are placed on hierarchically built panel grid. Main menu located in the top of the window provides control over all useful app features.



Base station application connected to arduino

3 PROJECT PLANNING

Task list of our project can be accessed using the following link.

https://trello.com/b/DM7W659L

3.1 Time schedule of the CanSat preparation

Our schedule is built around 4 key dates:

1. Preliminary Design Review (PDR) 6 November 2016
2. Critical Design Review (CDR) 15 January 2017
3. Final Design Review (FDR) 12 March 2017
4. Competition 13-16 April 2017

One week before all of them we have got planned critical team meetings which are meant to assess work progress.

Every part of our project has it own time schedule, described below.

3.1.1 Software

Missing features will be filled and bugs will be fixed during FDR preparations. Time directly before competition will be spent on polishing prepared solutions.

CanSat

- 21.01.2017 working script for reading data from thermometer
- 28.01.2017 final draft of all the needed data reading scripts
- 30.01.2017 data gathering scripts working properly
- 30.01.2017 project of the board (Eagle)
- 05.02.2017 working prototype of the board
- 12.02.2017 final version of the CanSat board

Ground Station (Arduino)

- 30.01.2017 final project of the board (Eagle)
- 14.02.2017 new algorithm to count received frames
- 28.02.2017 code in libraries and code formatting
- 14.02.2017 other corrections and tests

Ground Station (Desktop application)

- 26.01.2017 operable, highly efficient chart data display
- 29.01.2017 landing conditions tracking panel and analysis
- 02.02.2017 operable GPS tracking and landing site coordinates extrapolation
- 18.02.2017 advanced CanSat rotation tracking
- 26.02.2017 additional backup features
- 05.03.2017 advanced post mission data export

3.1.2 Hardware

Recovery system

- 28.01.2017 parachute mounted to core
- 29.01.2017 final decision on airbag idea

- 04.02.2017 test of appropriate parachute
- 18.02.2017 drone parachute drop test

Mechanical design

- 28.01.2017 final decision on housing material
- 11.02.2017 housing modelling
- 18.02.2017 full-weight CanSat housing drop test
- 25.02.2017 done CanSat 3D model (including electronic boards and core system)

Radio

- 13.02.2017 radio modules arrival
- 17.02.2017 working transmission code
- 19.02.2017 range testing and software tuning
- 24.02.2017 fine tuning and extra testing, final range estimate

3.1.3 Tests

We are going to test every part of project before take-off, in the closest possible date after finishing its design. Considering that separate tests will not provide us assurance on the reliability of project solutions, we plan to execute numerous system integration tests.edw

November:

1. Research meant to determine impact of the shape of the parachute on the descend parameters.
2.

December:

1. Study of the CanSat behavior during descent at a specific rate.
2. Reliability tests of power supply system aimed to determine parts factual power consumption and battery capacity.
3. Radio connection tests.

January:

1. Base station mock data analysis.
2. Reliability of the cansat data gathering scripts during fast movements.

3. Dropping a similar weight to the one of a CanSat (with a parachute) from a tall structure - meant to observe miscellaneous factors we have not yet taken into consideration.

4. Comparison of structural materials.

February:

1. Tests with the finished CanSat, checking if everything is working properly with low load (standing in place, being held in hands and moved a little).
2. Connection testing, refining of the transmission protocol.
3. Drop test of the CanSat simulating the impact it will suffer during the competition.

3.2 Resource estimation

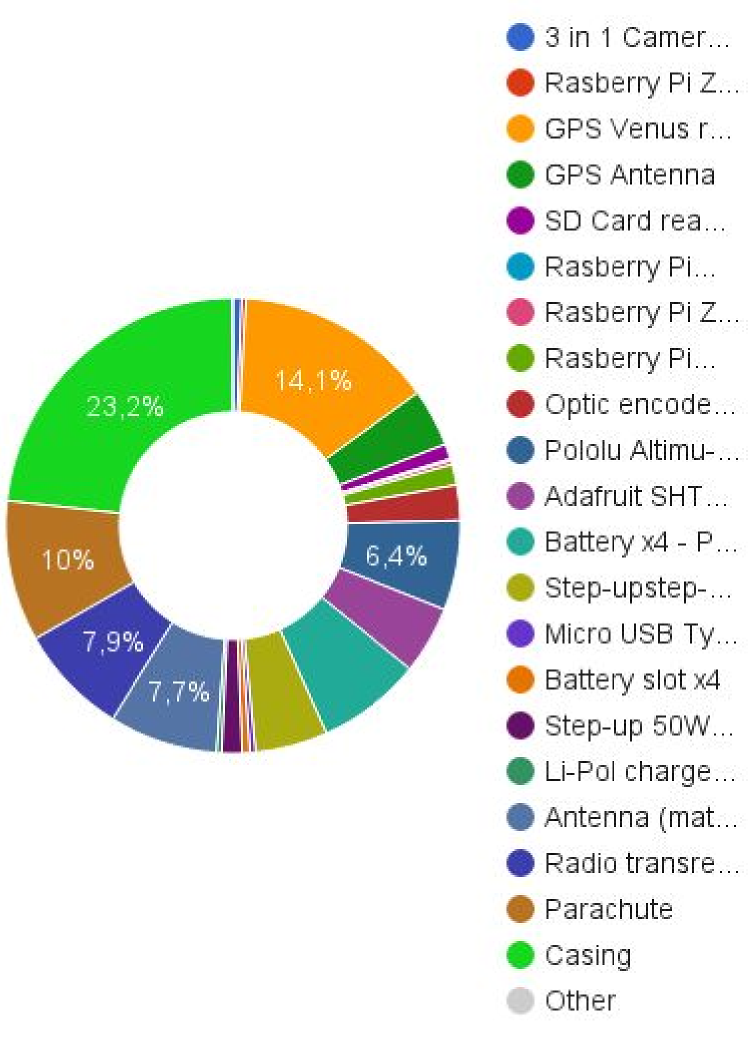
3.2.1 Budget

Position	Price	Status
3 in 1 Camera Lens Set	£8,00	Bought
Raspberry Pi Zero (pimoroni)	£4,00	Bought
GPS Venus receiver	183,21 zł	Bought
GPS Antenna	53,01 zł	Bought
SD Card reader module - waveshare	13,86 zł	Bought
Raspberry Pi Heatsink x2	£1,80	Bought
Raspberry Pi Zero Camera Adapter	£3,60	Bought
Raspberry Pi Camera Module V2	£18,90	Bought
Optic encoder Polo23	32,46 zł	Bought
Pololu Altimu-10 v5 - gyroscope and accelerometer	82,77 zł	Bought
Adafruit SHT31 - humidity sensor	62,31 zł	Bought
Battery x4 - Panasonic Li-Ion NCR-18650B	96,36 zł	Bought

Step-upstep-down converter x3	66,69 zł	Bought
Micro USB Type B 5 pin plug	5,11 zł	Bought
Battery slot x4	7,04 zł	Bought
Switch	0,84 zł	Bought
Step-up 50W converter	18,51 zł	Bought
Li-Pol charger TP4056 x2	4,66 zł	Bought
Antenna (materials)	100,00 zł	Bought
Radio transceiver	101,78 zł	Ordered
Parachute	130,00 zł	Ordered
Casing	300,00 zł	Planned

Sum (PLN)	1 258,61 zł
Sum (GBP)	£36,30
Sum (USD)	0
Sum (PLN+GBP+USD)(PLN)	1 438,54 zł
Exchange course (30.09.2016)	
Euro / PLN	4,312
GBP / PLN	4,95662
USD / PLN	3,8558
Sum (Euro)	€333,61

CanSat parts share in budget



3.2.2 External support

- We have started cooperation with Delphi Poland S.A. which should provide us with financial support in exchange for promoting Delphi with our marketing activities. Due to an excessive delay of unknown origin in providing us with financial aid we had to begin investment in the project with our own money. If found necessary, due to continuous Delphi's lack of interest, we might need to seek such aid from other companies. Obviously all our costs resulting from project development have been well documented and we expect them to be refunded by our future sponsor.
- DreamLab provides us with access to their conference room, where some of our meetings take place.
- We consider contacting Fideltronik and Motorola Solutions as well as School Board in search for additional support both financial and technical.

3.3 Test plan

Test schedule is described in the Section 3.1 of this document.

3.3.1 Parachute tests

Parachute test took place at Skałki Twardowskiego - limestone cliffs that form a group of hills located in the southern part of Kraków. Parachutes were made of an airtight material and paracords.
During tests we examined few issues:

- stability of flight trajectory,
- dependence the canopy shape on fall time,
- speed of falling object with parachute attached.

Parachute tests were conducted with three different shapes of canopy that were released from 30 meters above ground:

Data	Dome-shaped canopy	Dome-shaped canopy with a central hole	Square-shaped canopy
Fall time	16.9s	6.7s	15.1s
Stability	medium	perfect	medium
Flight trajectory	spiral	perfectly vertical	linear
Landing precision	medium	perfect	none
Other comments	-	high falling speed	soft but unpre-dictable flightpath

Table 6, parachutes comparison

Given the above data, we decided to focus on a dome-shaped canopy with a hole because of its flight stability, which is key for our CanSat mission. Further tests will be conducted in order to determine both optimal size of the canopy and the hole, as one used during testing appears to be of excessive.

3.3.2 Transmission tests

Unfortunately due to unforeseen circumstances in form of supplier unreliability, our communication tests have been postponed. Further design errors and equipment failure contributed to not having successful results at this time.

As two test attempts were conducted, we managed to prepare software solutions for them. These consist of simple receiving program displaying incoming message and a transmission program sending a simple string and a message number every fixed time. This setup proved to work well in rather low range tests. Providing a message number allows us to track whether there are packets missing or any new received at all.

Receiving antenna direction appears to be an important factor but following moving objects with it proved to be easy. Also we found that accidental covering the transmitting antenna as well as its improper placement may cause transmission disruption.

3.3.3 *Sensors tests*

Sensor reading tests were successfully conducted. External libraries designed specially for handling these devices were easily accessible in the Internet and well documented. Implementation of them was hard at first but as our skills in Python programming progressed, further work didn't cause problems. We find I²C protocol a really great solution for sensors of this advancement.

Tests were conducted on our temperature and humidity sensor as well as pressure sensor. Sensors returned valid data, checked with consumer-grade equipment, performed well in various environments, temperatures and were tested at high altitude of 2000 meters above sea level. Measurements were fast and accurate which assured us that these will meet our expectations regarding this project.

On-board computer's power demand didn't increase notably when sensors were connected (increase was not detected at all with our measurement tools) assuring that we don't have to be concerned with them while calculating power usage.

4 OUTREACH PROGRAMME

CanSat Team DevBlog

https://jjcansat.wordpress.com/

JJ CanSat Facebook profile

https://www.facebook.com/JJCanSatTeam/

GitHub repository

https://github.com/evemorgen/PuszkoSatelita/

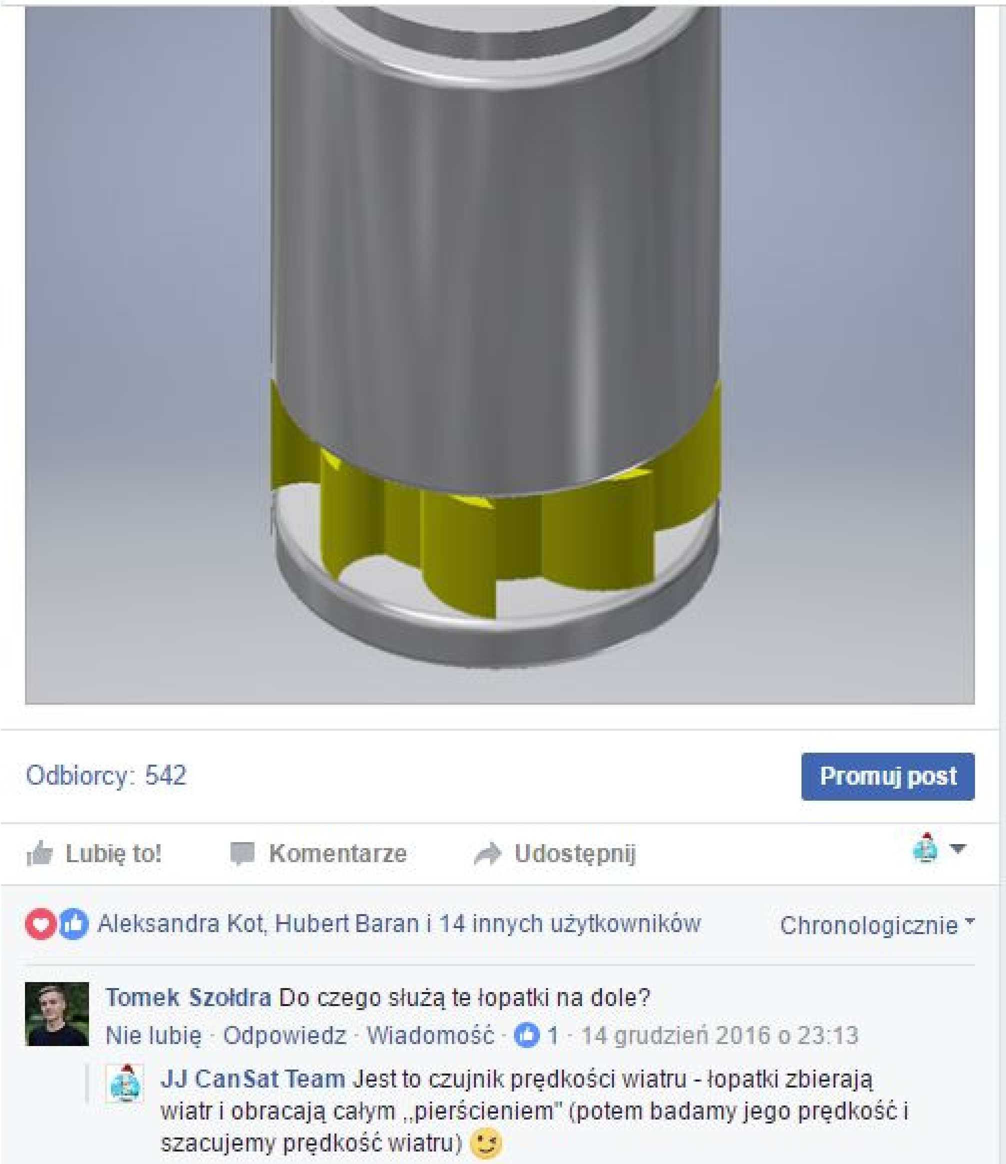
In this field we will be cooperating with our sponsors. We will surely discuss the possibility of presentations in our school with the headmaster.

We created two profiles, one on Facebook.com, wrote in Polish and one on WordPress.com in English. Our Facebook profile is gradually gaining popularity. At the moment of writing this report, we have more than 260 followers on our Facebook page and our last post reached more than 2000 Facebook users by itself.

Our goal is to publish frequently and keep our followers updated on our progress. We aim to produce high quality content for our Facebook page at a constant rate of one post per week.

We are discussing more detailed technical issues in posts in our DevBlog. Many of our followers are interested in technology and science, and our posts on DevBlog are addressed to them. We have received many questions about details of our work in comments at social media and directly. We found it satisfactory to answer them and share our work to other technology lovers. We believe in the power of Open Source and our code and technical documentation can be easily accessed at GitHub.com.

Our followers are amusingly involved and they want to keep up with the state of our project. This is one of many questions we have received since start-up of our work:



Follower: What are these blades for?

JJ CanSat Team: It is a wind speed sensor - blades are moving the whole ring with the power of wind (later we measure its speed and extrapolate the speed of wind) ;)

Competition for our CanSat team name

We have chosen the name for our CanSat by making a competition on social media. A 3D printed statue of the winner was the prize and it has attracted our followers to take part in it. We received many responses and it was hard to choose the one that would satisfy all of us. The decision has been made after voting through an online survey for our team. Finally ***Magnitudo*** became our choice. We have contacted the winner and discussed details of the prize. With nearly 2000 reached users we consider that competition to be a success.

KONKURS - pl. COMPETITION

Our outreach activities range is expanding rapidly - 4 digits growth ratio is not illusory, Facebook built in statistics are based on our average activity.

Zasięg - pl. Range

Aktywność dotycząca postów - pl. Activity connected with posts

Liczba wyświetleń strony - pl. Page views

We are planning to make a promotional video for our project, which would serve as an advertisement of our actions if we succeed in contacting and cooperating with any large media.

We consider contacting with a local radio station (Radio Kraków) in order to arrange a radio interview about our work. The local newspaper, Gazeta Krakowska, is also under consideration of our outreach team.

Done outreach actions
Making Logo
Creating Facebook profile
Creating DevBlog
Making background image for our online profiles
Activity on Facebook
Competition for our CanSat team name

Table 7, done outreach actions

Planned outreach actions
Promotional video
Radio interview
Newspaper interview
Further activity on our profiles
Presentations in our school

Table 8, planned outreach actions

Picture 2, JJ CanSat logo

Picture 3, JJ CanSat background image

5 REQUIREMENTS

Characteristics	**Figure**
Height of the CanSat	113mm
Mass of the CanSat	345g (extra weight included)
Diameter of the CanSat	65mm
Length of the recovery system	780mm
Flight time scheduled	120s
Calculated descent rate	10 m/s
Radio frequency used	433.8MHz
Power consumption	0.75A (3.3V)/2.5W
Total cost	€487,60

Table 9, requirements

On behalf of the team I confirm that our CanSat complies with all the requirements established for the European CanSat competition in the official Guidelines, APPENDIX 1 of this document.

Signature, place and date

Kraków 15.01.2016

6 ATTACHMENTS

Attachment 1 - Parts #1

Attachment 2 - JSON protocol

Attachment 3 - Ground Station Eagle project

Attachment 4 - CanSat model, Autodesk Inventor

Attachment 5 - Ground Station code, Arduino IDE

APPENDIX 1 THE CANSAT REQUIREMENTS

1. All the components of the CanSat must fit inside a standard soda can (115 mm height and 66 mm diameter), with the exception of the parachute, radio antennas and GPS antennas, which can be mounted externally (on the top or bottom of the can, not on the sides). NOTE: the rocket payload area has 4.5cm of space available per CanSat, along the can's axial dimension (i.e. height), which must accommodate all external elements including: parachute, parachute attachment hardware, and any antennas.

2. The antennas, transducers and other elements of the CanSat cannot extend beyond the can's diameter until it has left the launch vehicle.

3. The maximum mass of the CanSat is limited to 350 g.

4. Explosives, detonators, pyrotechnics, and flammable or dangerous materials are strictly forbidden. All materials used must be safe for the personnel, the equipment and the environment. Material Safety Data Sheets (MSDS) will be requested in case of doubt.

5. The CanSat must be powered by a battery and/or solar panels. It must be possible for the systems to be switched on for three continuous hours.

6. The battery must be easily accessible, in case it has to be replaced or recharged in the field.

7. The CanSat should have a recovery system, such as a parachute, which is able to be reused after launch. It is recommended to use bright coloured fabric, which will facilitate recovery of the CanSat after landing.

8. The parachute connection must be able to withstand up to 1000N of force. The strength of the parachute must be tested, to give confidence that the system will operate nominally.
9. The flight time is limited to 120 sec.
10. The descent rate must be between 8 m/s and 11m/s.
11. The CanSat must be able to withstand an acceleration of up to 20g.
12. The total budget of the CanSat should not exceed €500.

If your team has any problems fulfilling some of these terms, please contact the competition organisers for advice. Failure to meet the requirements will be taken into consideration by the jury but will not necessarily prevent the CanSat from being launched.