TinX

Technical High School no. 6 in Rzeszów

Final Design Review

CHANGELOG

Date	Description	Page
2021-01-28	Add terrain mapping	25
2021-02-07	Add final budget	32

1. INTRODUCTION

1.1. Team organisation and roles

Due to COVID-19 pandemic team started working online. We're planning CanSat development during meetings on discord platform. Each team member dedicates one hour per week during meetings and members are prepared to dedicate needed time in after school time.

- Ms. Anna Pękala (teacher)

  Background:

  Ms. Pękala has a Masters in Economics, and is a certified career counsellor. She has 15 years working experience with students in the fields of basic economics, entrepreneurship and career planning. Ms. Pękala has a number of interests, these include economics, psychology and personal development.

  Field of work within team:

  Ms. Pękala will be responsible for work planning. She will also be responsible for helping to motivate team members so that they can deliver their best work on this project. Ms. Pękala will also help members of the team to develop and grow through this process.

  Ms. Pękala will also assist in management of the budget for this project, and assist in fundraising as necessary.

- Mikołaj Data (Team Leader).

  Background:

  Mikołaj Data is a student of Technical High School no. 6 in Rzeszów. He has extensive knowledge in the fields of programming, data analysis, electronics

and astronautics. He has 6 years of experience in IT. Languages mainly used by him are C and Python.

Field of work with team:

He will be responsible for work planning, assignment of tasks, software and electronic development. His duties will also include website and social media managing.

- Przemysław Lib

  Background:
  Przemysław Lib is a 3D designer with many years of experience. Inventing useful devices and mechanisms is his passion. The programs he uses are Fusion 360 and Simplify 3D. He is a student of Technical High School no. 6 in Reszów.
  Field of work with team:

  His main task is to develop a safe recovery system, design CanSat's body. He is also responsible for finding financial support in external sources.

- Filip Różak

  Background:

  Filip Różak has experience in designing and troubleshooting electronic devices from different fields: RF circuitry, SMPS, industrial, production, automation. Furthermore, he develops software for microcontrollers such as AVR, PIC, ARM mainly in the C language. He has great interest in creating robust and cheap solutions for everyone. Filip Różak is a student of Technical High School no. 6 in Rzeszów.

  Fields of work with team:

His main task is to make design decisions about electronics in CanSat, develop schematics, circuit board(s) and write applications for onboard microcontrollers.

- Mateusz Soszyński

  Background:

  He is a full stack developer - has experience in many different aspects of programming - from embedded devices, to mobile apps, to managing Linux servers. He is also a passionate open-source community member, so he will be the most active in managing and maintaining our GitHub repositories.

  Fields of work with team:

  His main task is to take care of handling the data from satellites on the ground - processing and analyzing it on the back-end, and displaying it on the front-end.

- Aleksy Malawski

  Background:

  He is a graphic and motion designer. He is interested in animation and film editing. He has experience taking up many logos, illustration and motion design projects. Mostly creates his works using Adobe Illustrator and After Effects.

  Fields of work with team:

  He's responsible for creating posts, films and animation for teams' social media platforms.

1.2. Mission objectives

Due to some problems with cost of elements and COVID-19 pandemic, few of our goals were changed (see PDR report to see previous goals).

Primary mission: Measurements of atmospheric pressure and temperature with subsequent data analysis.

Key goals:

- Measurement of pressure and temperature outside the CanSat.
- Radio transmission of measured data in minimal rate one packet per second.
- Calculation of altitude based on atmospheric pressure assisted with geolocalization data.
- Measurement of temperature inside and outside the CanSat. To provide data about electronic parts performance.

Secondary mission: Demonstration of CanSat system for rescue operations during natural disasters.

Key goals:

- Design system that meet following requirements:
  - Low cost of design and produce
  - High accuracy and stability
  - Easy access to information about position, altitude, temperature, acceleration
- Take photos from the visible range camera - and analyze them.

Optional goals:

- Investigate a fire resistant biofiber composite with which body of CanSat will be built, keeping cost as low as possible.
- Create an open access dataset for aerial photography classification.

2. CANSAT DESCRIPTION

2.1. Mission overview

Mankind is more often a bystander of natural disasters caused by global warming. Our project is focused on helping rescue teams during their hard work. To do that we propose a CanSat prototype that is equipped in camera that will be collecting images and strength led diodes that will be indicating position of the device and marking evacuation points for victims of natural disasters such as floods, bushfires, earthquakes etc.

Images are a very good source of information. Thanks to them we are able to create maps, find potential victims locations and estimate the range of disaster. (See Data Analysis for more informations)

We are developing that solution as an alternative for drones. Nowadays UAVs are expensive and their capabilities are limited by battery, so they can't operate in big areas and during disasters there is the risk that UAV can be destroyed. If our solution will be produced on a massive scale costs could be decreased.

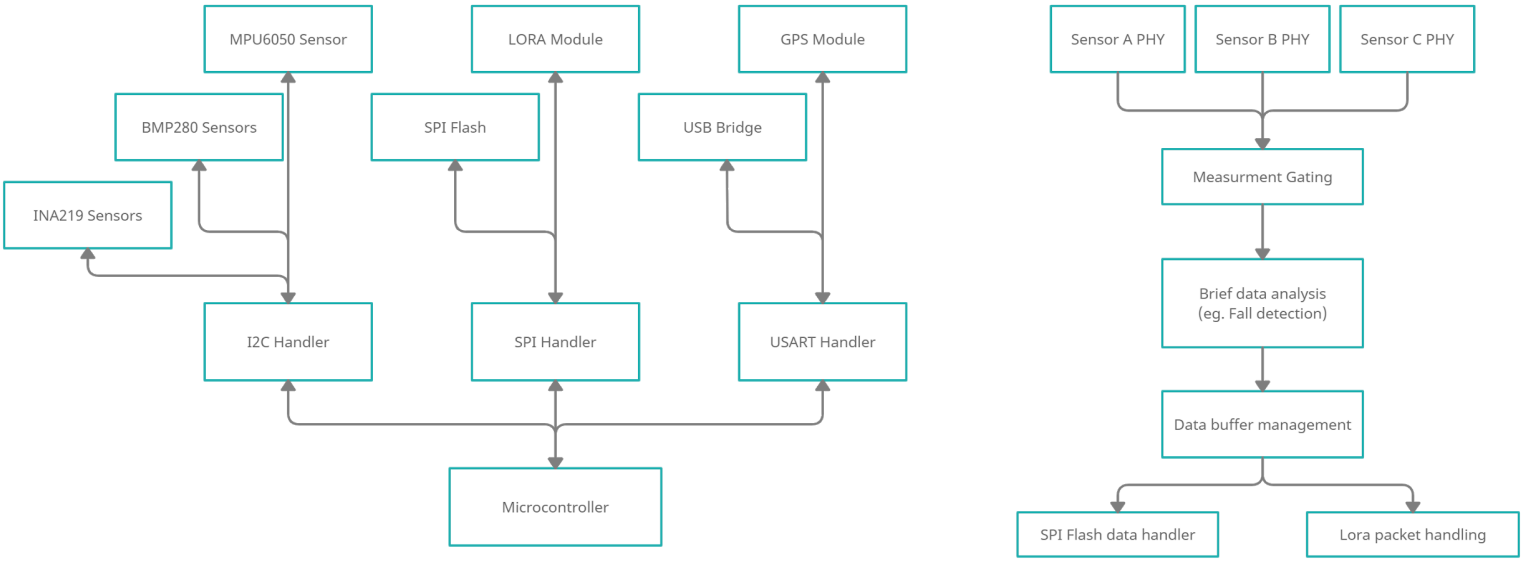
The CanSat competition organisers provide rocket or UAV as a test platform. Although to achieve low-cost and long in-air time we propose to use balloons and airplanes to operate the CanSat.

To prove our concept is working during test flight the following conditions must be met:

- CanSat must work minimum 8 hours
- The map made from gathered data must be legible
- CanSat must be visible from ground station

Images and telemetry gathered during the mission will be available for other engineers, scientists to help them improve their solutions. CanSat on-board computer start is planned for +/- 2 hours before take-off. During take-off the

satellite accelerometer will inform the on-board computer about rocket launch, then the primary mission starts. When CanSat will be deployed the secondary mission starts. Landing is predicted 4 minutes after deployment.



CanSat architecture and data flow

2.2. Mechanical/structural design

General objectives for the mechanical structure of the CanSat:

1. We will introduce the construction of CanSat inertial dampers to protect sensitive electronics, such as cameras, against vibrations and high acceleration stresses. - Canceled
2. The internal frame with electronics will be covered in a semi-fireproof shell made of fibers and ceramics.- Canceled

3. The internal frame will be printed with 3D technology, using nylon or materials with similar properties.-Semi included
4. Sensors will be placed in the external shell and will be connected by flexible wires to the main board.- Included

During development, we decided to exclude dampers from the project. Tests showed that small rubber pieces don't work as well as we expected, moreover it complicates the production process too much and weakens the connection between PCB bracket and internal frame of the canstat.

CanSats body still will be based on an internal PCB bracket screwed to the internal frame, but we recognised that our fireproof material is too heavy to use in cansat project. We made several different versions of fireproof material based on natural fibers but all of them were heavier than our assumptions. Instead of this we decided to use the same material like we used to make canstats frames, that material is nylon reinforced with glass fibers.

Before the production process we decided to exclude PCB frame from cansat because we expect that pdb is hard enough to cope with stresses in vertical direction. Frame construction has been applied to batteries , which are covered in special holders screwed to the cansats internal frame.

Sensors are attached to external covers with thermoplastic glue. Flexibility of that kind of glue ensures some protection in case of parachute failure.

General structure description:

Whole case is 3D printed using nylon reinforced with glass fibers. We decided to ask the Friberlogy company to print our cansat's body, because that kind of filament is really hard to print and requires a 3D printer with specific design. External covers and battery holders are combined with body using M3 screws and special threaded brass inserts, which are placed using soldering iron set to 250 degree celsius. PCB is screwed to the body using screws and nuts.

2.3. Electrical design

General architecture

Electronic part of Cansat has been optimized for low-cost, highly efficient and fail-safe production process, assembly and operation. Usage of Off-the-shelf components has a significant impact on overall cost and opens a very cheap way to mass production.

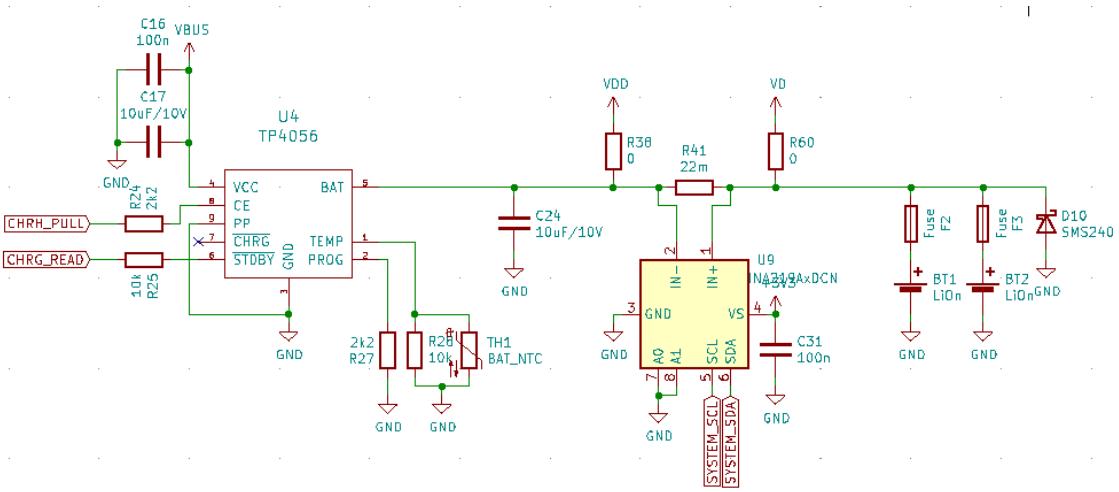
Power supply

When in operation, device will be powered from li-on cells. This chemistry was chosen, because it's comparably safe, operates well enough in a wide temperature range, is widely available, inexpensive and comes in almost every physical shape. To save space inside, flat (eg. phone) style batteries are used.

Li-on cells need to be charged properly - in CC and CV modes. Exceeding any of their normal parameters will decrease battery capacity and possibly damage the cells (or even the whole device due to explosion/leakage). Used batteries have protection circuits integrated inside. It protects individual cells against:

1. Over current discharge protection
2. Over discharge protection

To cover all of the battery requirements, also charging voltage and current has to be regulated. Most of this is done with just one physical chip (TP4056), which is a very common, cheap and trusted li-on cell charger. It does CC/CV modes switching, over charge protection.



This part of the schematic shows the battery charging and monitoring system. F2 and F3 polyfuses protect against overcurrent/short circuit, if battery without protection inside is used. D10 protects circuit against connecting battery backwards. U9 measures battery voltage and current (using R41 shunt). U4 is an integrated charger with control/status pins connected to a microcontroller.

Battery level can be estimated from voltage measured on battery (using U9) and/or calculating charged/discharged energy from battery.

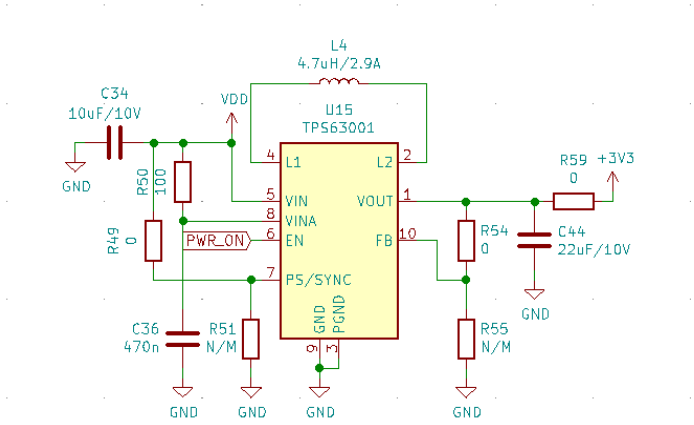
Power management:

To optimize battery live - INA219 integrated bidirectional voltage/current sensors will be used. It makes power monitoring simple, reliable and cheap. INA219 has an integrated A/D converter and PGA, thus no additional analog components are not needed. This leads to very low price and efficient usage of PCB space (<1cm^2 for complete current/voltage measurement). Two of such monitors has been used across the board:

1. Measures battery charged/discharged current and battery voltage.
2. Measures battery voltage and current took by high voltage devices (Led Light and buzzer)

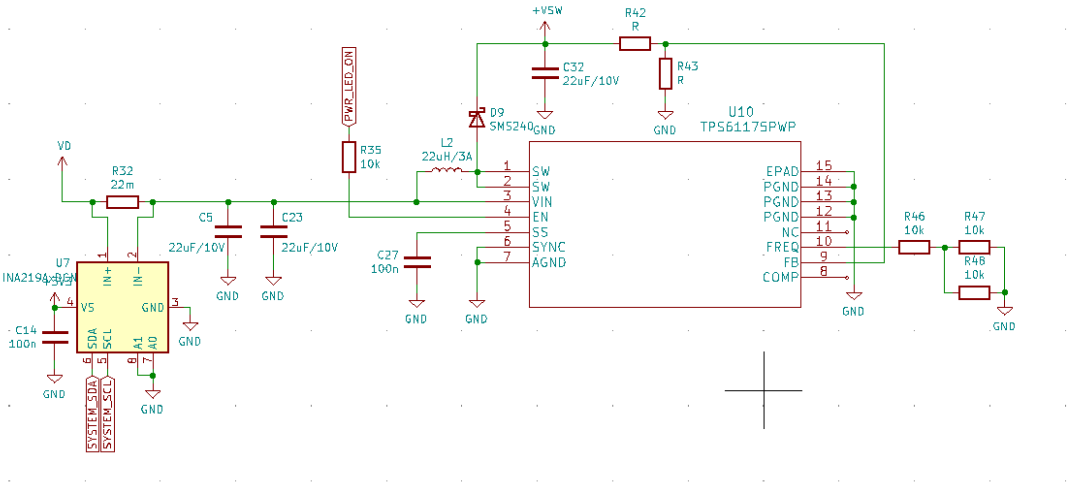
Creating System voltage rails:

Most of the IC's on the board (including main controller) operate on 3.3V - thus there is a need of creating stable, reliable supply rail. To operate on the whole battery capacity - voltage must be stepped up (VBAT < 3.3V) or stepped down (VBAT > 3.3V). This can be achieved by using Buck-Boost converters such as TPS63001. It has integrated switches, automatic mode (Buck/Boost/passthrough) selection, Enable input (used to completely power off device) and operates on single, physically small inductor (due to high switching frequency: 1,2-1,5Mhz). At currents >100mA estimated efficiency of this converter is >80%.

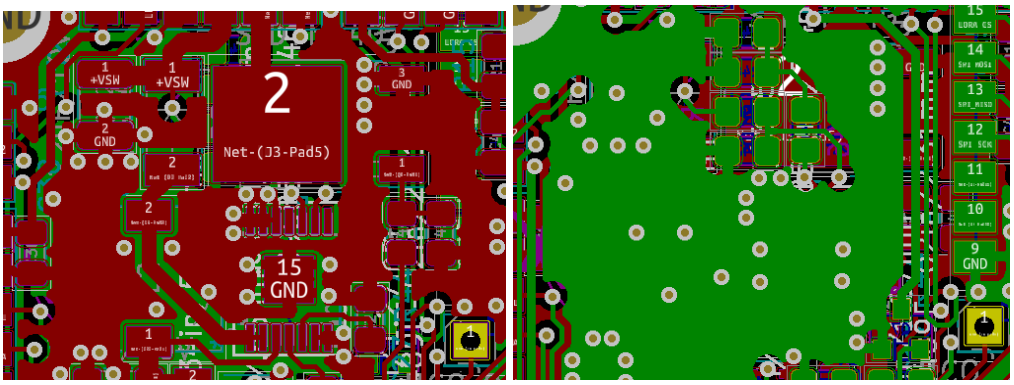


Creating 3.3V rail is very simple, and thus reliable using such an integrated converter. Low external components count minimizes board space taken by power management, to let more room for other circuitry.

To power Led Lightning with approx. 15V, stepping up of the voltage is needed (buzzer is also powered from this rail). TPS61175 integrated Step-up is used for this purpose. It's small and simple to implement, guarantees high efficiency (>85%).



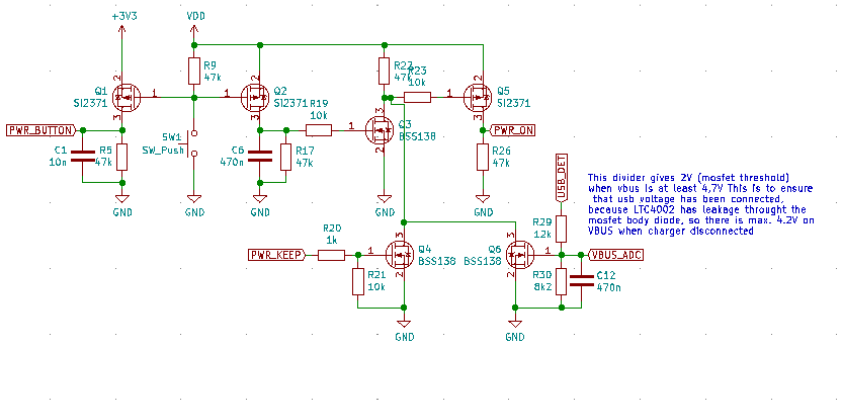
To keep EMI at a low level, and not disturb sensors in the device (or even other devices), proper layout of switching converters must be utilized.



1. Ground plane under most noisy traces
2. Thick ground connection to switching components and bypass capacitors.
3. No analog traces near the converter
4. Concise structure of the whole converter.

<u>Power-up logic:</u>

In order to make the device operate only, when needed (save battery during transport and so on), power-up logic is implemented. To switch the device off, 3.3V converter “Enable” input is used.



Microcontroller (and other circuitry on 3.3V rail) is powered only if:

1. Momentary button is pressed
2. Processor supplies power to itself
3. USB charging port is connected

This approach makes power-on sequence very reliable - cansat can be turned on by a momentary button, then a microcontroller starts up and latches power to itself (PWR_KEEP signal). Then - button can be used to power cansat down, but this function will be enabled only after the mission or corresponding lora command. This makes sure that no false power off will be triggered. Microcontroller also operates, when the USB charging port is connected - to show battery level on status LED, and integrate power injected to battery.

Estimating needed battery:

All values are maximal, worst case (in exact mode). Needed operating time is 3h in operation and 10h in standby (beacon).

Component	**Operation**	**Standby**
BMP280 x2 (Pressure + Temp)	19uA * 200Hz = 3,8mA	2,7uA
MPU6050 (Gyro+Accel)	3,9mA	500uA
A25l032 (Flash)	21mA	15uA
STM32F103RGT6	70mA (impossible - all pheripherials enabled, max speed)	15uA
INA219 x2 (current measurement)	1mA	15uA
ESP32-CAM	200mA	20uA
Lora SX1278	120mA	1.5u
Led lightning	700mA 2/5 duty = 140mA	-
TPS63001	60uA	1uA
TPS61175	3.5mA	1.5uA
Buzzer circuit	5mA	-
CP2102	26mA	100uA
TP4056	6uA (from battery)	6uA (from battery)
Genuine GPS receiver	5mA	200uA

Sum	510mA	1,4mA

We could do further calculations, but this data is not reliable - we are estimating power consumption of ~350mA when in operation and 2mA standby. Two 1,5Wh 3,7V Li-On cells with protection are used. Tests will show real values of consumed power.

Primary mission devices:

Primary mission of the device is to acquire temperature, pressure values, store them with altitude information and let the user read the data using a memory card. Radio communication will occur at least 1 time per second.

To measure Altitude and atmospheric pressure BMP280 sensors are used due to their numerous advantages:

1. Low cost
2. High reliability (tested in millions of devices)
3. Fast Data acquisition
4. Low power consumption
5. Simple to implement: I2C communication
6. Accurate enough, factory calibrated

To improve temperature data collection in speed domain (and relative accuracy) additional external sensors (LM35) are used - they have analog output and are physically small, this allows measurement of very fast temperature changes outside the device.

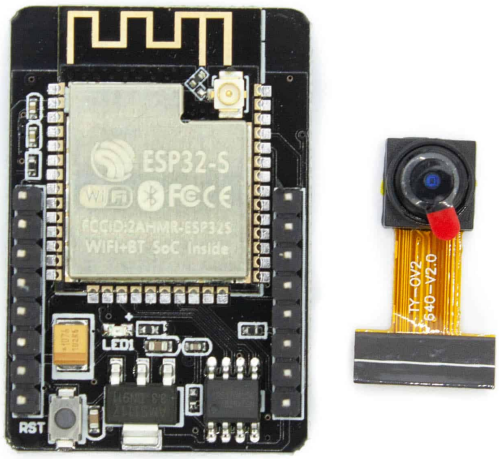
Main controller has been chosen, based on:

1. Low price
2. Good availability
3. Ease of programming/development. Evaluation board for this family exists.

4. Long production live (+10 years guaranteed)
5. Enough resources for possible expanding functionality

To meet all of the above, STM32F103RGT6 microcontroller has been selected. It has Cortex-M3 core, and a handful of useful pheripherials. It's maximum frequency is 72Mhz on the core, and 36Mhz on fast periphs. All the inputs/supply pins operate at 3V3 logic levels, as well as every other chip on the board, so no level shifters are needed. This microcontroller also has automotive temperature range, clock save system and reliable BOR circuitry.

Secondary mission devices



Main goal of the secondary mission is to make photos from the air, for further analysis. Usage of off-the-shelf cameras is not cheap enough for this device. However cheap ESP32-cam modules are available, they have reasonable capabilities compared to their very low price. Such modules integrate ESP32 processor together with external flash memory for firmware and camera connection. The OV2640 camera can be used with the module directly - it's small, has 2Mpx resolution and is very cheap. Usage of a secondary processor just for image handling is very efficient, because it has to be done as fast as possible and not interfere with other tasks. Images will be directly saved on memory card, due to their enormous size.

To obtain device rotation and acceleration in three-dimensional space (this is needed to determine mission phase and camera pointing direction), a gyroscope

and accelerometer is used. MPU 5060 has been chosen due to it's small physical dimensions, speed (measurement up to 1Khz) and low price. To achieve this performance, a separate I2C bus has been occupied as well as dedicated interrupt pin as well as frame synchronisation.

To obtain device geolocation position, a generic Telit GPS module is used. It uses a supercapacitor to keep satellite data in RAM, this allows for hot-start of the Receiver and makes overall fix time faster. Communication with the module is one-way.

All data, excluding photos, are saved to on-board flash memory - A25l032. It's very fast (SPI speeds up to 100Mhz) and has internal ram for "per page" writing, which leads to even faster transactions on the bus, leaving processor time for other tasks than storing data.

Purpose of the Led light is to make the device visible and recognizable, when falling. Selected Light Emitting diodes are 3W 10000k white Bridge lux (4 of them). Every LED has a voltage drop of 3.5-3.7V. Connecting them in parallel to the battery is not an option, due to the fact that battery voltage can fall to 3,1 volts at normal operation (near discharge). Step up switching converter has been utilized to power the LEDs, connected all in series, which gives voltage of approx. 15V.

Communication system

Communication going to be performed in two ways:

1. From CanSat to ground station - acquired data, actual state, diagnostics

2. From ground station to CanSat - control commands

To communicate over a recommended distance of 3kms and meet power requirements, a highly efficient system has to be used. One of approaches to achieve that is to use a modulation technique optimized for low-power long-range

transmission. Adequate tests have to be done, but from research can be seen, that LORA standart meets above requirements.

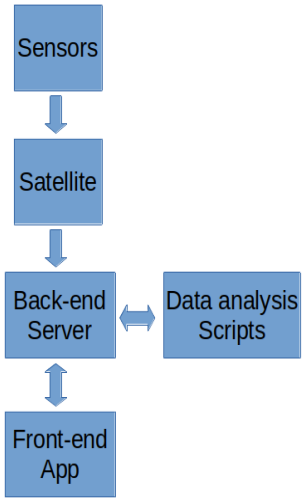
Selected LORA module is SX1278, in the RA-02 case. It communicates over SPI bus at speeds up to 10Mhz. During on-the-air transactions, it needs a maximum 600mA of current, so an appropriate capacitor bank has to be placed physically near the module and power traces should be thick enough.

All long range communication systems require tuned antennas, to increase connection budget (tuned antennas have connection budget profit). Possible antenna tuning will be possible with Vector Network Analyzer. As for the ground station antenna - GP type will be used, Cansat antenna is still under development.

2.4. Software design

Primary requirements for software in project:

1. Managing peripherals onboard CanSat.
2. Store and transmit data.
3. Analyse data.



To achieve all listed requirements software will be divided into three parts: satellite, ground station and algorithms for image analysis.

Software will be developed under GPLv3 license, and shared on our Github (https://github.com/TinXsat).

Satellite software design

Satellite will be operating in 4 modes:

1. Before flight - calibration of sensors, low-level self test.
2. Lifting - data collection from accelerometer and possibly environment inside the rising system.
3. Falling - data collection and processing for on-board use. Execution of primary and secondary mission tasks. Maintenance of communication with the base station.
4. On the ground - enter low power mode, transmit recognition packets, possibly also with GSM. Fetch data to removable storage (sd card) for further analysis.

To automatically distinguish between mission phases, state machine can be used. Such a system would take into account several variables: accelerometer reading, pressure (altitude). However, if some parameters would not be as assumed, the base station can send command to change mode.

Onboard data analysis will provide only basic information about the environment, due to limited processing power and speed needed to acquire data. Lack of operating system force usage of interrupts and timing peripherals of the microprocessor to implement multitasking, but also improves real-time operation synchronization.

Acquisition of such an amount of data from environmental sensors requires extensive averaging and filtering. To provide enough data to work with, high speed sampling has to be used.

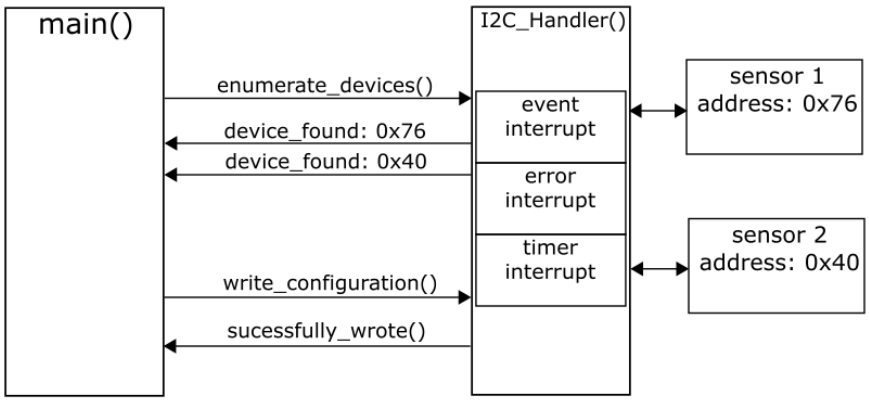
Data communication with different onboard devices has to be solid and fail-save. In order to achieve that, extensive self testing of every funcional block has to be implemented. Software should report failure and (if possible) change/disable faulty block(s) and continue to work - this reduces downtime of the system in the air.

Calibration procedure before flight will improve absolute accuracy of onboard sensors and their adaptation to different environmental conditions (atmospheric pressure changes rapidly even within a day).

To handle all of the needed auxiliary sensors, devices - several buses have to be used in the system:

1. SPI bus for SX1278 Lora module and Flash memory
2. I2C bus for MP6050 Gyroscope/accelerometer.
3. I2C bus for BMP280 sensors and INA219 meters.
4. RS232 (in 3V3 logic) for camera configuration/management

Each of those need to be handled separately, allowing pseudo multitasking for the system. This functionality is achieved by using various interrupts offered by the peripherals.



This picture shows the working principle of I2C_Handler - it creates an independent state machine, so the main program might be doing one thing, while busses are busy.

Ground station software design

Ground station will be responsible for collecting, storing, processing and presenting data received from CanSat. To handle all those tasks we are going to create a back-end server connected with a front-end app.

All changes in code are versioned with the “Git Flow” technique - every stable version has it’s semantic version code (for example 1.5.3), and is available to download in GitHub releases tab on it’s repository.

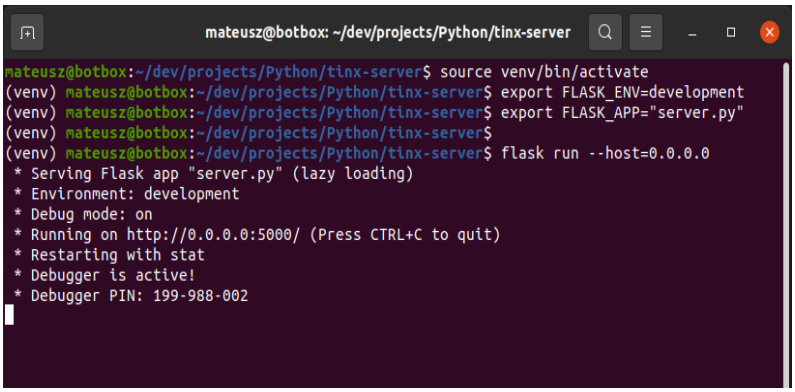
This way, we can easily debug which versions have certain bugs, and roll back to old ones if new changes cause problems.

Back-end server is the program that collects and processes the data received from the satellite.

Data from the radio receiver is read by USB UART serial, and saved in the database/csv files.

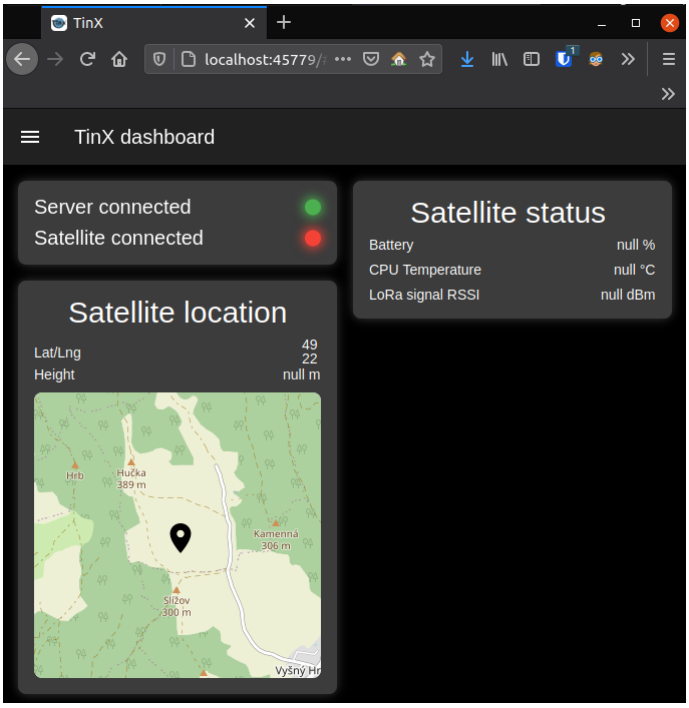
So essentially, it acts as a simple proxy between LoRa and front-end, and also records the data.

It is written in Python, and uses Flask to serve REST API for the front-end.



Repository:
https://github.com/TinXsat/server

Front-end app, that is, everything that end-users interact with when our satellite launches. It is made with Flutter, because it’s cross-platform, fast to develop, and very easy to build good-looking UI. It will be served from the same computer where the back-end will live - no need to install any extra software for the end user.



Repository: https://github.com/TinXsat/frontend

Data analysis

To improve the development process of algorithms we migrate to Google Colab platform. All notebooks are stored in README.md in a repository for data analysis on GitHub.

Repository: https://github.com/TinXsat/data-processing-algorithms

Flight telemetry analysis. Thanks to sensors aboard CanSat we've got access to velocity, height, temperature inside and outside of CanSat. It's very important to analyze all of it and draw conclusions. That data will help us to improve our device for next launches and future missions of the device.

Photo analysis. We are using a dataset created from videos publicly available on YouTube platform (https://www.kaggle.com/miki164/tinxsat). Advantage of this solution is access to photos that will be very similar to photos gathered during CanSat launch. Disadvantage of this solution is the small size of the dataset (826 photos), so we have to use many augmentation techniques to make the dataset more useful.

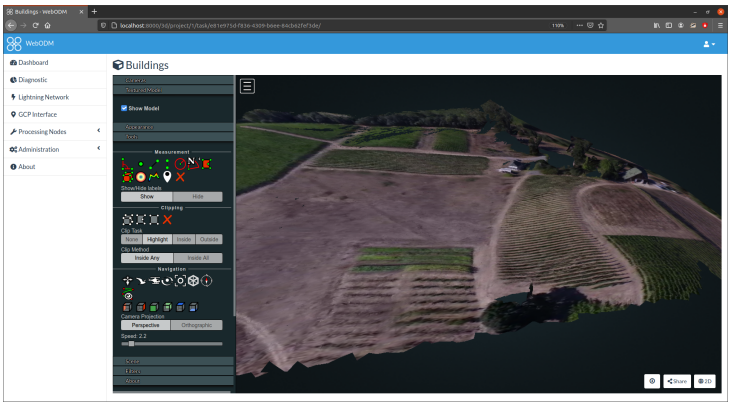
Team develops algorithms for:

- Image segmentation to simplify images and make them easier to analyze (see more in *Secondary Mission Tests*)
- Generative adversarial network (GAN) to generate artificial images based on photos provided from cansat to augment dataset. (see more in *Secondary Mission Tests*)
- Road detection algorithms (works are now suspended due to small amount of images in dataset)

Next algorithms will be provided in the future.

To create the algorithms we are using Python and its data science libraries e.g: numpy, pandas, tensorflow, matplotlib. We are planning to develop neural networks and develop computer vision programs to make data gathered from CanSat more useful.

Terrain mapping



Thanks to images gathered during flight the ground team will be able to create 3D maps. To do that we will use Open Drone Map software (https://github.com/OpenDroneMap), which allow to create:

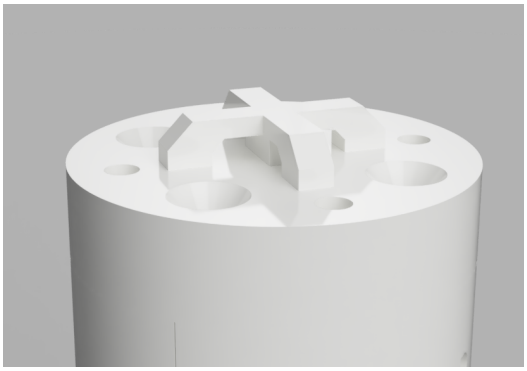
- Classified Point Clouds
- 3D Textured Models
- Georeferenced Orthorectified Imagery
- Georeferenced Digital Elevation Models

Open drone map is free software that gives many options to customize the process of mapping and increase speed of processing by adding more nodes.

Data sharing

Data gathered during CanSat launch will be shared under MIT (or similar) license on Kaggle platform. It will give capability for other people to use data in their works.

2.5. Recovery system



Our recovery system is based on a parachute. That solution is cheap, safe and reliable. Parachute is a hexagonal shaped bowl made with light nylon fabric. Its dimensions are 30x30 cm and were calculated by using a website: descentratecalculator.onlinetesting.net We expect 9.54 m/s free fall velocity. Parachute is attached to a special mounting on the top of the cansat.

2.6. Ground support equipment

Our ground equipment will contain:

- Communication device
- Antenna
- Server (optional)
- Ground control computer

Communication device will be based on the radio receiver module, its task is to receive data from CanSat and pass it to the computer. It will cooperate with the antenna to extend transmission range.

Ground control computer is a device which shows telemetry and all data related to the mission. It can also analyze data and create 3D maps.

To speed up the process of creating maps we recommend usage of a server with docker image of NodeODM (https://github.com/OpenDroneMap/NodeODM). In our case we will use an old Dell laptop (Intel I7, 8GB RAM) with ubuntu server.

3. TEST CAMPAIGN

3.1. Primary mission tests

Temperature and pressure measurement

To measure temperature and pressure - BMP280 sensors are used. For the test - a simple program has been written. It outputs data from sensors at a 100Hz rate over debug UART. RealTerm terminal emulator running on windows machine is used to receive this data.

Due to the fact that we don't have any temperature standards, nor calibrated thermometer, accuracy stated by the datasheet has to be taken as an axiom. However, relative accuracy of pressure has been tested using the fact, that it corresponds to the altitude directly:

altitude = 44330 * (1.0 - pow(pressure / 1013hPa, 0.1903))

(simplified formula - optimized for speed on embedded systems)

One series of measurements was taken on the ground, and another on the third floor.

Results: Average pressure measurements were:

1. On the ground: 990.6hPa (190.27m)
2. On the 8.5m over the ground: 989.6hPa (198.75m)

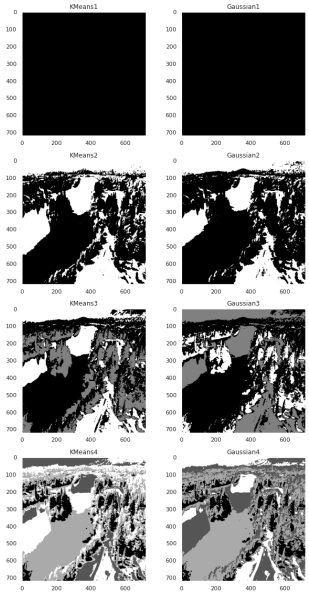
It gives a difference of 8.48m, which is good.

3.2. Secondary mission tests

Notebooks used in tests

1. https://colab.research.google.com/drive/1wgNq5ZRyhjGuFB5EDJgYtSbQyw5s14Gt?usp=sharing
2. https://colab.research.google.com/drive/1PM4GragHtV4pXwYbwWQrfmQu8L9xn6lM?usp=sharing

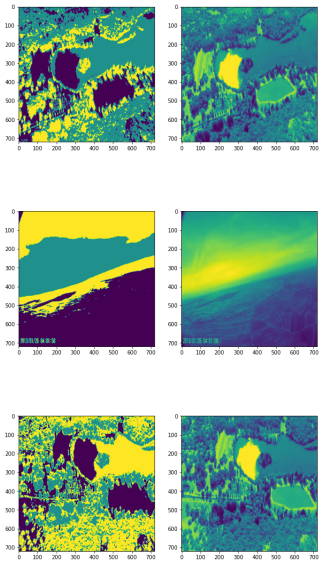
Choosing segmentation algorithm [1]



First step in developing image processing was choosing an image segmentation algorithm. We compared two most popular algorithms KMeans (left) and GaussianMixture (right).

Results: We choose that we try to use both algorithms, because segmented images are very similar. Next tests are needed to determine which algorithm is better.

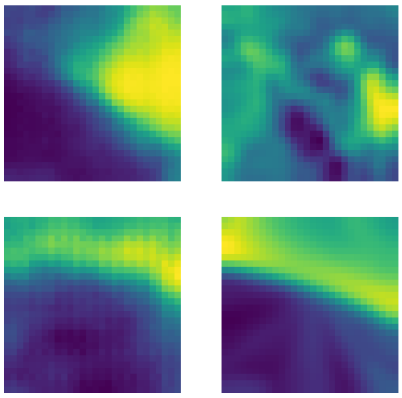
Plotting segmented images [2]



Team created notebooks with algorithms to segment images. We plotted segmented images (left) and grayscale images (right) to present results of segmentation.

Results: Comparison of images shows that segmentation works properly and it can be used for training the neural network and future data processing.

Deep Convolutional Generative adversarial network (DCGAN) for artificial image generation



The original dataset is too small (826 photos), so the team created the GAN to augment the dataset. It will help in future analysis of CanSat flight and in expanding the capabilities of our solution.

Results: Network made by team works well on photos with 28x28 (pictures on right are generated by GAN, pictures on left are from dataset) size, but on higher resolutions model returns random values. After some research we are planning to test HDCGAN (https://arxiv.org/pdf/1711.06491v12.pdf).

3.3. Communication system

Lora range test

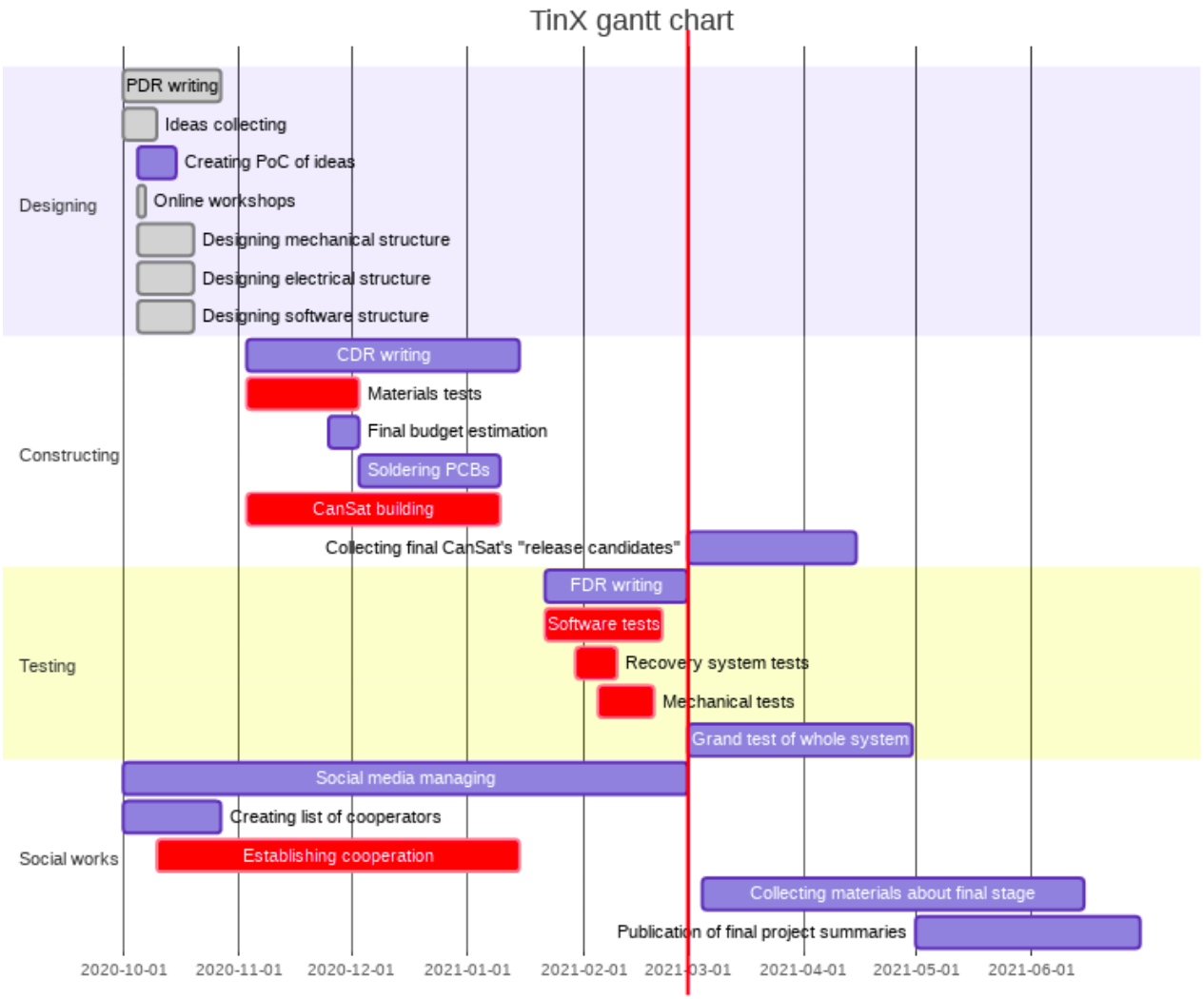
For this test - antennas had not arrived yet, but we tried the lora communication with standard ¼ wavelength antennas (piece of wire with length of 165mm). Polarization was vertical, but unfortunately - antennas were on the same height. This was due to the lack of high enough buildings around.

Results: RSSI which lora can safely work with reaches around -100dB, and sometimes caches even -120dB. Distance, at which RSSI got to -100dB was around 600m away, with lots of buildings and no direct line of sight (needed for high frequency - 433Mhz communication). With such simple antennas - this is more than expected in this environment, in the air it would be at least x2.5 times longer distance. However, on the specially tuned antennas (GP on the ground and Dipole on the air) we expect around 3km range.

4. PROJECT PLANNING

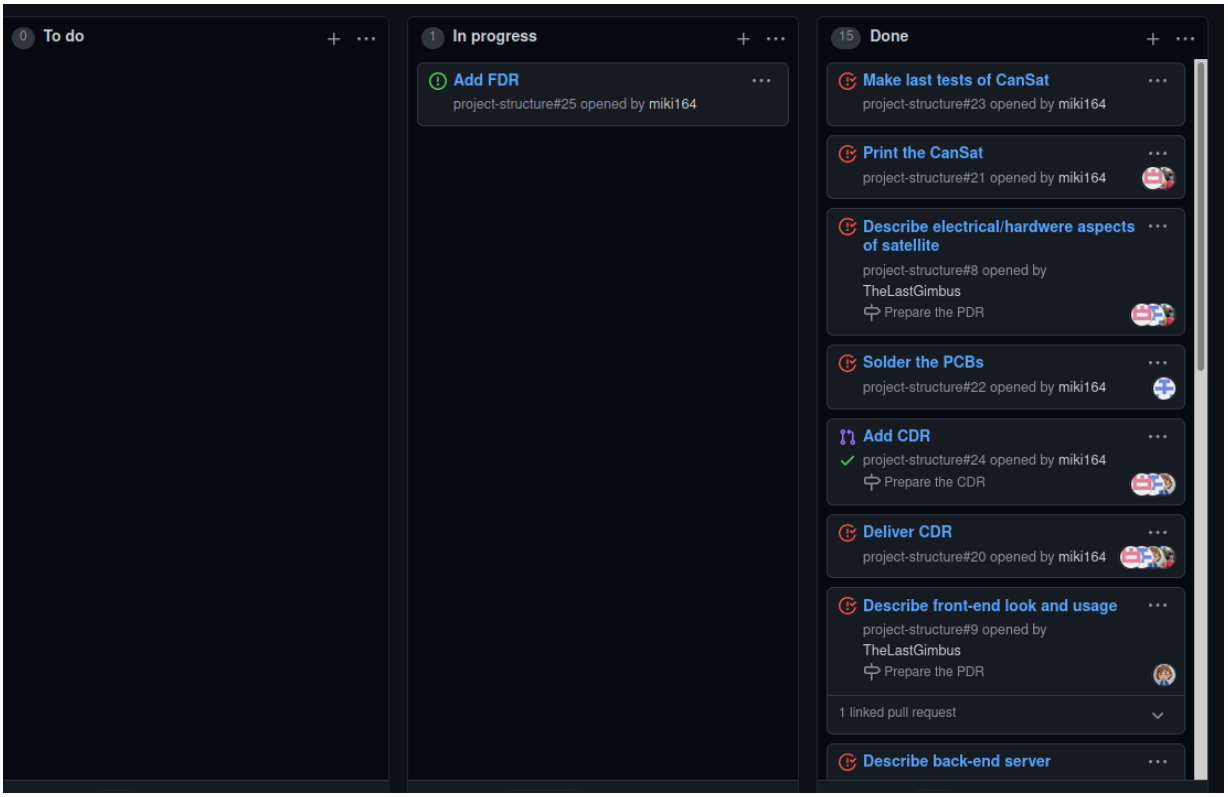
4.1. Time schedule

Time schedule is created using https://github.com/mermaid-js/mermaid



4.2.Task list

Team task list is updated in real time and is shared on Github platform. (https://github.com/orgs/TinXsat/projects/1)



4.3. Budget

No.	Part name	Quantity	Price per piece [EUR]
1.	STM32F103	1	3
2.	MPU6050	2	1
3.	BMP280	2	1
4.	INA219	2	2
5.	ESP32CAM	2	5
6.	TP4056	1	1
7.	Led lighting	1	2
8.	Battery	2	3
9.	sx1278	1	8
10.	Filament cost	-	7
TOTAL:			45

4.4. External support

In order to obtain support, the team established cooperation with:

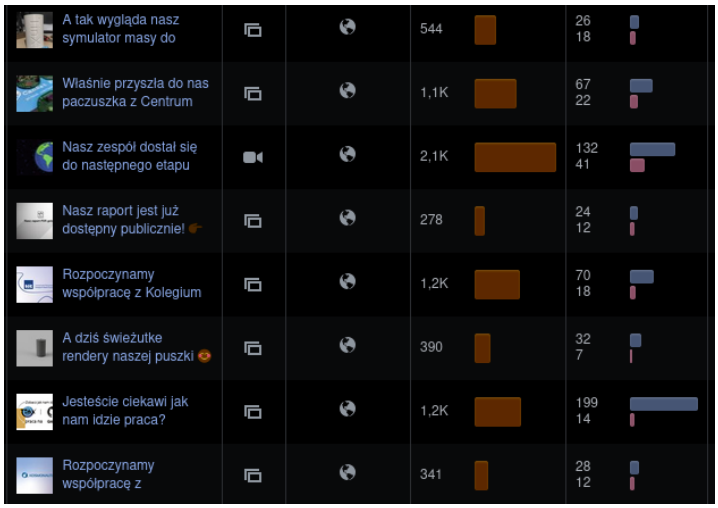
- Nettigo.pl - sponsor and electronic parts provider
- Kolegium Nauk Przyrodniczych Uniwersytetu Rzeszowskiego - substantive assistance and rental of research equipment
- PcbGoGo - PCB manufacturer
- qlrs.pl - the antennas manufacturer

- Blue Dot Solutions - substantive assistance and team advertising
- Kosmonauta.net - substantive assistance and team advertising
- Gabriel Gałęza - website design and development
- Szymon Mytych - website design and development

4.5. Outreach programme

During the competition team started:

- Website - tinxsat.github.io - addressed to our cooperators or future cooperators
- Fanpage on facebook - fb.com/tinxsat - created to encourage people to programming and creating space projects.



  Fanpage got 90 likes. We were creating posts that could reach 2000 people. Posts were created regularly.

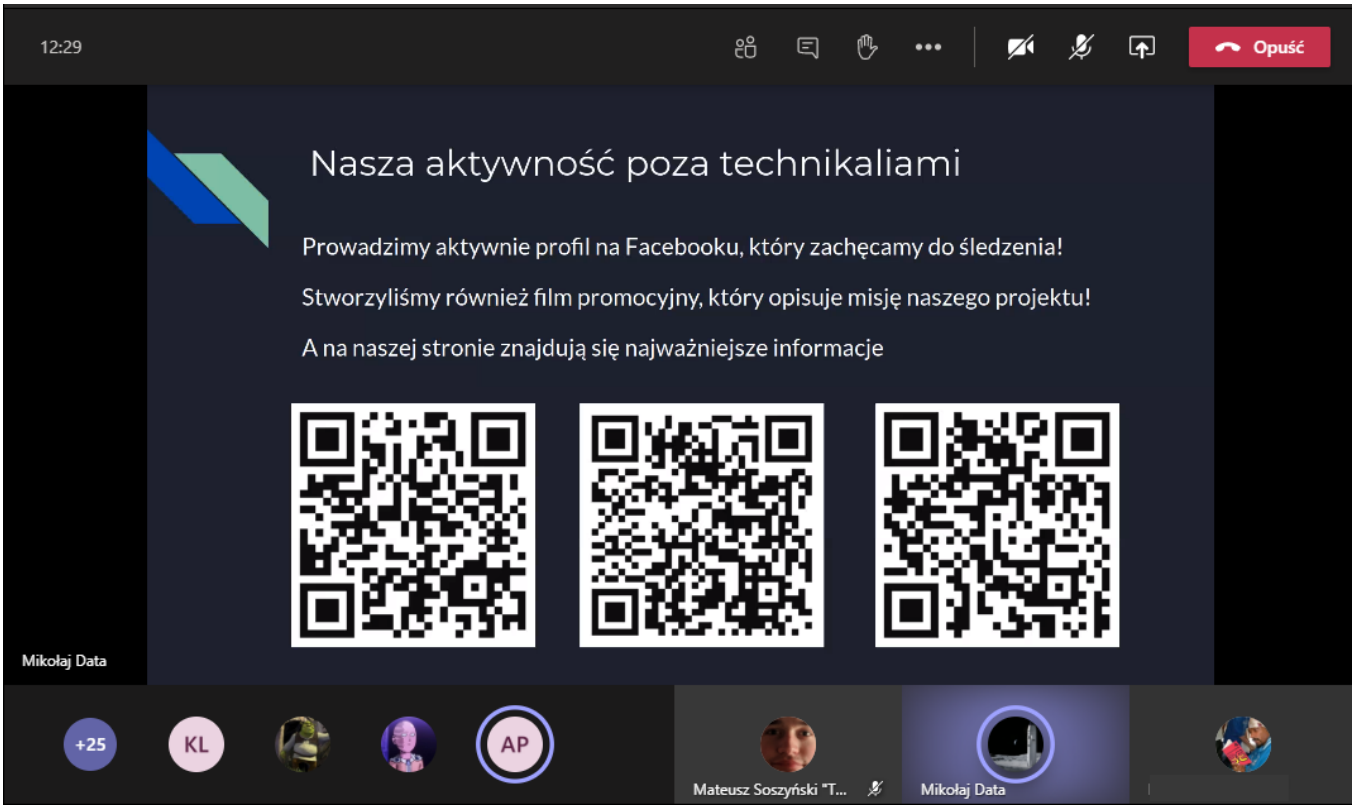
- YouTube channel - created to encourage people to programming and creating space projects.
- Github organisation - github.com/tinxsat/ - addressed for programmers, data scientists. It creates opportunity to get more help from open-source developers community
- An email - tinxasat@gmail.com - created to communicate with companies, cooperators and government agencies.

  We were focused on creating posts regularly to maximize posts reach. Team also created a promotional video (https://youtu.be/_mBKociq5aw) to promote the project and encourage potential sponsors to cooperate. The outreach

programme was one of the most important parts of the project thanks that we obtained many support from companies and other people.

An article was created about the team on the kosmonauta.net portal (https://kosmonauta.net/2021/01/rzeszowski-zespol-kontynuuje-prace-w-konkursie-cansat-jaka-jest-ich-misja/).

Team presented the cansat project during the online lessons in school.



5. CANSAT CHARACTERISTICS

Characteristics	Figure
Height of the CanSat	113.8mm
Diameter of the CanSat	65mm
Estimated mass of the CanSat	348g
Estimated descent rate	9.54m/s
Radio transmitter model	SX1278
Estimated time on battery (primary mission)	8 hours
Cost of the CanSat	45 EUR