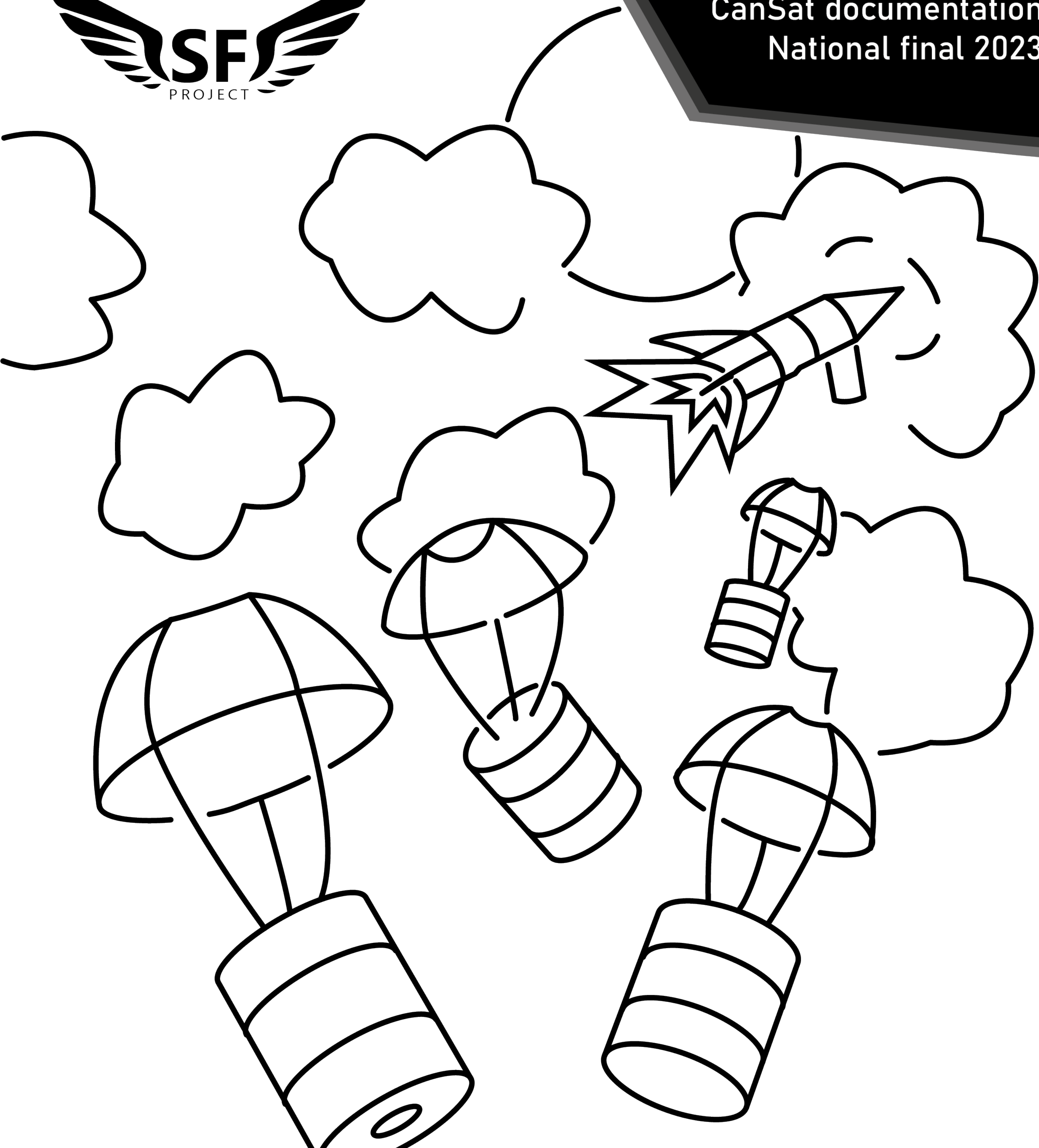
CanSat documentation
National final 2023



School: SPŠE a VOŠ Pardubice/Gymnázium, Pardubice, Dašická 1083

Mentor: Ing. Vladimír Kašpar

Table of contents

1. Introduction

Hello, we are the team Project Skyfall, one of the competing teams in the annual CanSat competition held by ESA. We've already participated last year and came in third in the national finals. But for those that don't know us yet, we would like to introduce ourselves once again.

1.1 Our team

Our team is still comprised of 5 students from SPŠE a VOŠ Pardubice and one student from Gymnázium, Pardubice, Dašická 1083. Here is a listing of the current team members and what each individual member does.

1.1.1 David Haisman

David handles electronics and programming on our CanSat. He designed PCB's, made some libraries and wrote the whole CanSat code. During the development of the CanSat he gained a lot of experience regarding PCB design and programming in C++.

1.1.2 Radim Leština

Radim is in charge of designing the outer construction of the CanSat itself. He has a lot of experience in designing and 3D printing mechanical contraptions.

1.1.3 Lukáš Nastoupil

Lukáš designed and made the ground station, which is an essential part for our secondary mission. He designed the PCB's, modeled and programmed the ground station. He also worked on the real-time transmission of video from CanSat down to the ground station.

1.1.4 Václav Horáček

Václav is in charge of multiple departments, mainly the ones for documentation and recovery. In documentation he made the styles, graphics and oversees the documentation making process. His contribution in the recovery department is designing and sewing the CanSat's parachute.

1.1.5 Adam Keřka

Adam leads the team itself and also the department of publicity. This mainly includes maintaining social networks accounts and communicating with sponsors. He also designed a lot of the graphics used in advertising and posts on social networks.

1.1.6 Vít Brázda

Vít is responsible for data visualization running on ground station, as well as making index of habitability of planet, where CanSat is landing. He has a lot of experience regarding Python.

1.1.7 Ing. Vladimír Kašpar

Ing. Vladimír Kašpar is our mentor in the CanSat competition. We chose Vladimír Kašpar as our mentor based on his experience. In the CanSat 2019 he was a mentor of team Charles the fourth that made it to the European finale. He is also an electronics teacher at SPŠE a VOŠ Pardubice.

1.2 External support

Due to the limits of team composition, we have some special people that helped us in one way or another that we cannot include in our team. Here is a listing of them:

1.2.1 Jan Novotný

Jan Novotný is a freelance developer who makes web and mobile applications. He programmed the Project SkyFall webpage and supports our team on various other competitions and outgoings.

1.2.2 Ing. Zdeněk Nosek

Ing. Zdeněk Nosek is an electronics teacher at SPŠE a VOŠ Pardubice that also has a little bit of experience with CanSat competition.

1.2.3 Ing. Petr Bubák

Ing. Petr Bubák is an electronics teacher at SPŠE a VOŠ Pardubice who gave us many useful pieces of advice about electronics.

1.2.4 Tereza Truncová

Bc. Tereza Truncová is a graphics teacher at SPŠE a VOŠ Pardubice who helped us with making identity of this team and several graphics.

1.3 Sponsors

Special members of our team are our sponsors, who help us finance our ideas, but not just that. We always have the opportunity to communicate with them any bug or just things we aren’t sure about. Last but not least is propagation. Some of our sponsors offered to share our project on social media or write an article for example. Thanks to them our publicity has an even bigger impact.

1.3.1 LaskaKit

LaskaKit belongs to the biggest e-shops with electronics and microcontrollers in the Czech Republic. Most of the electronics we have used come from their racks.

- Source: https://www.laskakit.cz/

1.3.2 T-CZ

T-CZ is a successor of the well-known Tesla Pardubice in radiocommunication. The company helps us with telemetry and provided us with an aluminum part of CanSat that we designed. We often visit them to communicate any troubleshooting or just thing things we want to help with.

- Source: https://www.tcz.cz/

1.3.3 FYFT

FYFT is a store focusing on free-time activities including building drones. Fyft gave us a tiny camera with 4K resolution.

- Source: https://www.fyft.cz/

1.3.4 Voltio

Voltio is a store installing smart home devices and offering 3D printing as a service. Voltio purchased a filament we chose and printed for us most of construction in CanSat.

- Source: https://www.voltio.cz/cs/

1.3.5 Espressif

Espressif is one of the biggest world-wide companies when it comes to microcontrollers and chips. They provided us with chips and microcontrollers.

- Source: https://www.espressif.com/

1.3.6 Foxconn

Foxconn is one of the biggest world-wide companies when it comes to electronics. Foxconn financed our project.

- Source: https://www.foxconn.cz/

1.3.7 RPi shop

RPi shop is the only official shop providing Raspberry in the Czech Republic. They offered us electronics.

- Source: https://www.rpishop.cz

1.3.8 Technecium

Technecium is a free-time centrum that provided a workplace for us to hold our developing sessions.

1.3.9 Schools

Special thanks go to our schools GYPCE and especially SPSE that also finances our project.

- Source: https://www.spse.cz/
- Source: https://www.gypce.cz/

1.4 Our mission

Our mission is to make a planetary exploration device, we mean a probe that could measure the oxygen levels, temperature, light spectrum and much more in the atmosphere of the Galilean moons for example.

We are going to achieve it with double-sided communication between the ground station and CanSat, but not just that. We developed a graphical user interface (GUI) that suits our needs. GUI visualizes data during the fall but also enables us to take control over the CanSat.

Our team went all out on adding more sensors for measuring the surrounding conditions. That means that we have added sensors for measuring the oxygen in the atmosphere, measuring the amount of each wavelength in the light spectrum and other features to determine if the probed planet is safe for life or why it might not be. This is where GUI comes in.

2. CanSat

The CanSat is the main part of the competition. Each CanSat is supposed to be unique and to work in different ways. It is what makes a team stand out.

2.1 Construction

Construction is divided into two main parts. The internal infrastructure and the CanSat cover.

The internal infrastructure is made out of the Oxygen holder, the Battery household and the Camera holder.

2.1.1 Main structure

The main structure is made out of four shredded rods, two aluminium plates and the CanSat cover.

Four shredded rods make main rails for everything in our CanSat (Power Unit, Control unit and internal 3d printed infrastructure).

CanSat cover is mounted between two aluminium plates. The aluminium plates are holding CanSat cover in his place.

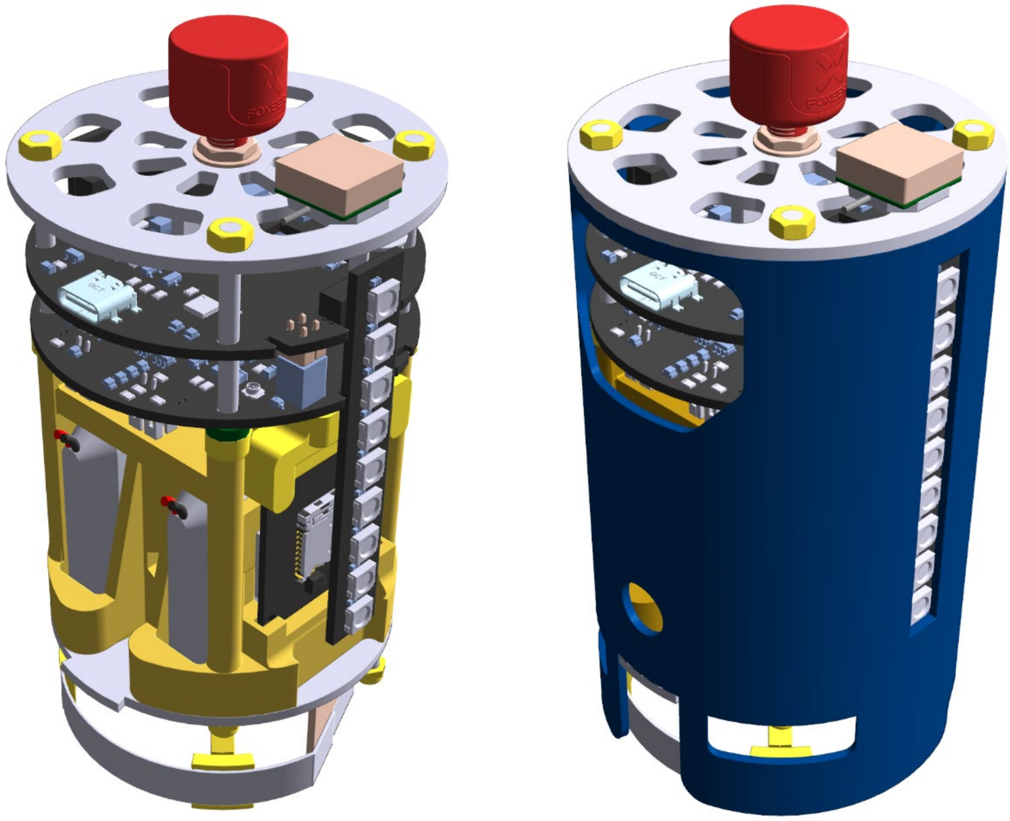


Figure 1- The CanSat without and with it's shield

2.1.2 CanSat cover

The CanSat cover is 3d printed from the PETG material.

The biggest hole in the front of CanSat cover is there for easier way to manually control our CanSat with CanSat cover included. With it we can manipulate with DIP switches and reset button. USB C cable and micro-SD card can be plugged or unplugged through this hole.

Three vertical holes with 90° offset are holding Side Units. Side Units could be all time inserted in these holes, they aren't affected by CanSat cover movement, if we take it off.

With holes placed at the bottom of CanSat cover nuts M3 can be easily grabbed with wrench.

The hole under the biggest hole is there for easier way to manipulate with screw M3, which is holding Battery household, bottom aluminium and Camera holder together.

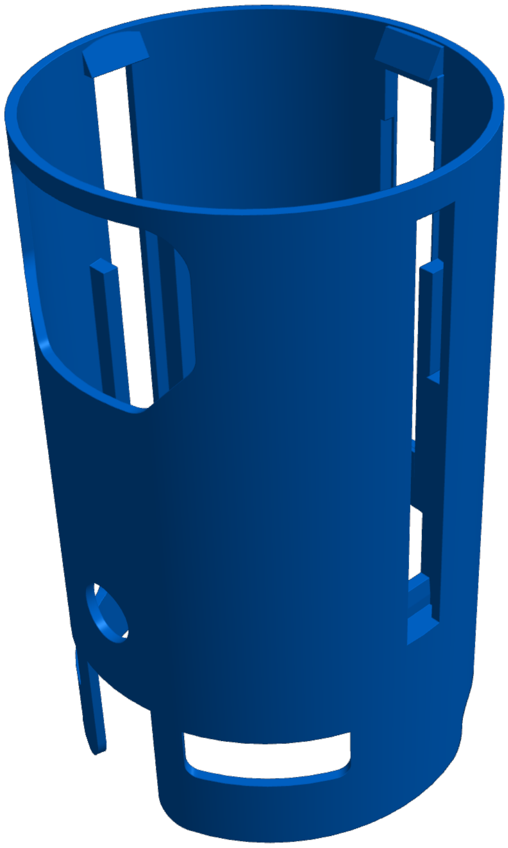


Figure 2- The CanSat cover shields the main construction inside

2.1.3 Oxygen holder

Oxygen holder is a 3d printed part, that supports the Oxygen sensor under the Power Unit. It is also holding the camera module.

It is attached to the holding rods with its three supports. One is missing since it made accessing the camera module significantly harder.

It has three arms with 3mm holes in them to fit between shredded rods. One arm is missing due to camera connector accessibility.

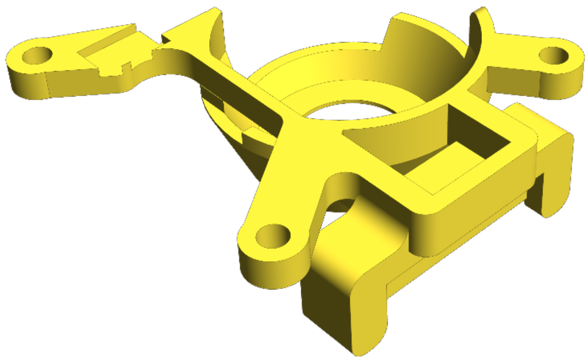


Figure 3 - The Oxygen holder supports the oxygen sensor and the camera module

2.1.4 Battery household

The battery household is placed under the Oxygen holder.

It's supporting the batteries, so that they will stay in their positions and makes It easier to put CanSat together.

On the right side there is a camera PCB socket. With similar socket as in the Oxygen holder, together these sockets are holding this camera.

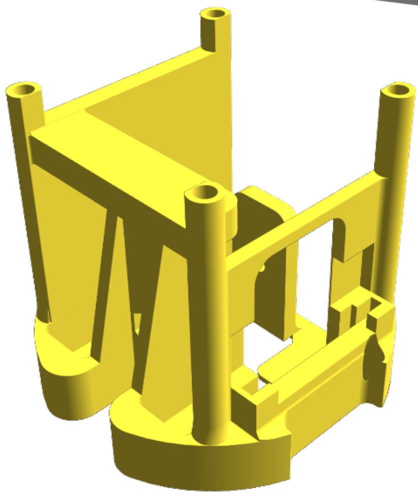


Figure 4 - The battery holder prevent the batteries from moving on impact

2.1.5 Camera holder

It's holding the camera on the bottom of CanSat. Thanks to the Camera holder, the CanSat cover can't be seen in the field of view of the camera.

It can be connected to the Battery household via one M3 screw with a M3 nut.

These two higher towers help to hold batteries to prevent side moves.

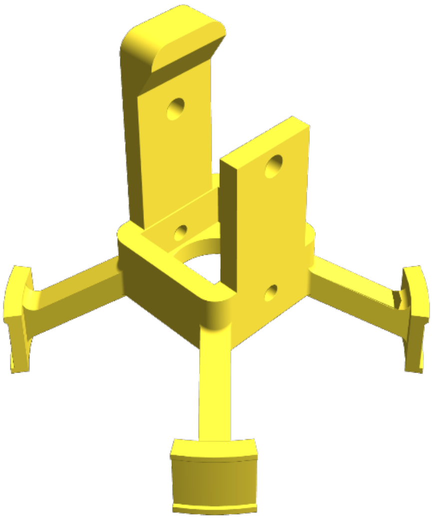


Figure 5 - The camera holder is a part, which holds the camera still

2.2 Electronics

Without the electronics, the CanSat would just be a weight falling on a parachute.

With our electronics construction it is possible to power and gather data from multiple sensors to fulfill the requirements of our primary and secondary mission.

2.2.1 ESP32-MINI-1

The ESP32-MINI-1 is CanSat's one and only microcontroller.

Everything from saving data to removable storage, sending & receiving data through air, communicating with all sensors and other peripheries to hosting internal webserver used to runtime configuration is done thanks to this microcontroller.

The IC offers 28 GPIOs, 4 MB flash, powerful Xtensa® dualcore 32bit LX6 microprocessor, built-in PCB antenna for Wi-Fi and Bluetooth standard.

Figure 6 - ESP32-MINI-1 controls whole CanSat

2.2.2 CP2102N

The CP2102 is USB 2.0 to UART converter.

The IC handles communication between the host computer and CanSat's microcontroller. New versions of CanSat's program are also uploaded through this IC.

Some of the chip pins are used for automatic switching to download mode or signalizing active communication between devices.

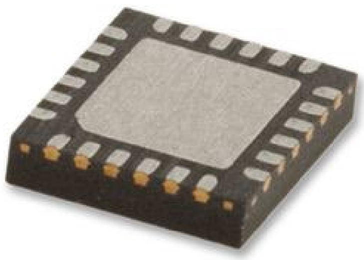


Figure 7 -The CP2102N handles the communication between the CanSat and a computer

2.2.3 BME280

The BME280 is an IC that can measure temperature, humidity and pressure.

This sensor is necessary for obtaining temperature and pressure data required for primary mission.

- Configuration:
  - Operating voltage: 3.3 V
  - Bus: I^2C_0
- Measured data:
  - Temperature:
    - Range: -40 to 85 °C
    - Accuracy: ±0.5 °C (0 to 65 °C)
  - Pressure:
    - Range: 300 to 1100 hPa
    - Accuracy: ±1 hPa (0 to 65 °C, 300 to 1100 hPa)
  - Humidity:
    - Range: 0 to 100 %RH
    - ±3 %RH

Figure 8-The BME280 sensor measures the surrounding temperature, humidity and pressure

2.2.4 DS18B20

The DS18B20 is a temperature sensor. It is used together with BME280 to acquire data required for the primary mission.

- Configuration:
  - Operating voltage: 3.3 V
  - Bus: OneWire
- Measured data:
  - Temperature:
    - Range: -50 to 125°C
    - Accuracy: ±0.5 °C (-10 to 55 °C)

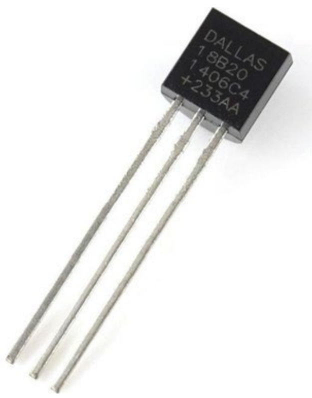


Figure 9 – The DS18B20 is a temperature sensor used for measured temperature compensation

2.2.5 MAX1626

The MAX1626 is a step-down DC-DC controller with fixed output voltage.

Whenever the output voltage is 3.3 or 5 V depends on voltage level of control pin.

IC works with 300 kHz PWM frequency, that can be relatively easily filtered out. The downside is that this frequency requires a significantly bigger inductor.

Figure 10 - The MAX1626 is a step-down converter used for powering the board

- Configuration:
  - Input voltage: 2.6 to 16 V
  - Output voltage: 3.3 or 5 V
  - Frequency: 300 kHz
  - Efficiency: >90 %

2.2.6 PT7C4339

The PT7C4339 is a low-power real-time clock and calendar.

The date is automatically adjusted at the end of months with fewer than 31 days, same thing applies for leap years up to year 2100. IC automatically detects power failure and switches to backup power maintaining actual date and time.

External 32.768kHz crystal is mandatory for this IC to work.

Figure 11 – The PTC7C4 is an integrated circuit keeping time in our CanSat

- Configuration:
  - Operating voltage: 3.3 V
  - Backup voltage: 1.3 to 3.7 V
  - Bus: I^2C_0

2.2.7 RFM96W

The RFM96W is a half-duplex transceiver featuring LoRa™ and FSK/OOK modulation.

Output power can be as high as 100 mW. The SyncWord function is used to filter out other nearby RFM9xW transceivers. The module has a built-in CRC control, which can repair broken messages.

The IC offers 5 GPIO pins with changeable predefined meanings. CanSat handles interrupts to check whenever message was successfully sent and to read an incoming message.

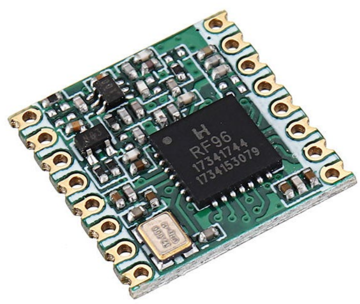


Figure 12 - The RFM96 is a module used for double-sided communication

- Configuration:
  - Operating voltage: 3.3 V
  - Bus: SPI
  - Transmitting power: 10 mW (or 50 mW when operated by an authorized person)

2.2.8 NEO-M9N

The NEO-M9N is a GNSS module from the company U-BLOX.

It is used together with active GNSS antenna to acquire position, altitude, date & time with second precision and number of satellites providing data every half a second.

Backup power source enables hot-start and maintain RTC, that could also be used to keep date and time when power is lost. CanSat itself has independent RTC on Control Unit, so this RTC is used as a backup system only.

External low-power LED blinks every second to indicate whenever GNSS module acquired FIX. When no FIX is established, LED remains turned off.

An average cold start takes approximately 26 s for the first FIX. When hot or aided start is enabled, FIX time can be as low as 2 s.

- Configuration:
  - Operating voltage: 3.3 V
  - Bus: I^2C_1
- Measured data:
  - Longitude & latitude
    - Accuracy: ±2 m
  - Altitude
  - Date & time
  - SIV (satellites used in acquiring FIX)

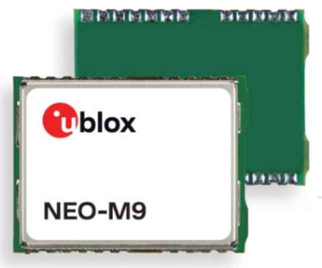


Figure 13

- The NEO-M9N is a GNSS module for position acquirement.

2.2.9 SCD41

The SCD41 is a miniature NDIR sensor measuring PPM concentration of $C0_2$ gas in the atmosphere. It's a very small sensor, especially when compared with its predecessor SCD30.

Due to working principle, minimum measurement cycle is 5 seconds and can be only slowed.

- Configuration:
  - Operating voltage: 3.3 V
  - Bus: I^2C_0
- Measured data:
  - CO_2 concentration:
    - Working range: 0-40000 ppm
    - Accuracy: ±40 ppm

Figure 14 – The SCD41 is a sensor which measures CO2 levels in it's surroundings.

2.2.10 INA219

The INA219 is a current shunt and power monitor.

IC works based on proven principle of measuring voltage drop-out over resistor with small and well-known resistance.

In our CanSat it is used to measure input current and power in hand with actual accumulator voltage.

- Configuration:
  - Operating voltage: 3.3 V
  - Bus: I^2C_0
- Measured data:
  - Current
  - Voltage
  - Power

Figure 15 – The INA219 is a sensor which measures the input current and power

2.2.11 PCA8574

The PCA8574 is a GPIO extender with eight quasi-bidirectional I/O ports. This means, that IC requires only one register to control all I/O ports, that can be input or output at the same time (for more detailed working principle please refer to product datasheet).

Almost every IC on Power Unit requiring some I/O for operation is in fact realized through I/O ports of PCA8574. Thanks to this, the total number of connectors between CanSat BCBs could be lowered significantly.

- Configuration:
  - Operating voltage: 3.3 V
  - Bus: I^2C_0

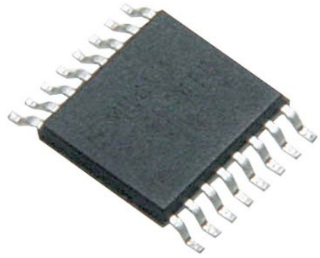


Figure 16 – The PCA8574 is a GPIO extender used to connect more sensors

2.2.12 BNO055

The BNO055 is a System in Package integrating 14-bit accelerometer, 16-bit gyroscope and geomagnetic sensor together with 32-bit microcontroller running own sensor fusion software.

- Configuration:
  - Operating voltage: 3.3 V
  - Bus: I^2C_0
- Measured data:

Figure 17 - The BNO055 is a gyroscope which also measures acceleration

- Fused rotation angles (roll, pitch, yaw)
- Acceleration

2.2.13 ME2-02

The ME2-O2 is an electrochemical sensor working based on chemical reaction between electrolyte and oxygen in atmosphere. Sensor then acts like current source with output proportional to oxygen concentration.

Excessive vibrations and high accelerations should be avoided. Obviously, this is not possible in CanSat, we expect highly inaccurate data during the flight, that will come back to normal after landing.

In our CanSat, a parallel 100 Ω resistor is used to convert current output to stable voltage, that is then amplified with non-inverting operating amplifier MCP6486UT. The actual oxygen concentration is then estimated from analog input value and fact, that normal oxygen concentration in atmosphere is 21 %.

Figure 18 – The ME2-O2 is a sensor which measures the oxygen level in

- Configuration:
  - Operating voltage (amplifier): 3.3 V (non-symmetrical)
  - Bus: analog input
- Measured data:
  - O_2 concentration:
    - Range: 0-25 % (30 % max)
    - Response time: <15 s

2.2.14 Caddx Loris 4k

The Caddx Loris 4k is a drone analog camera, capable of recording up to 4k 60 fps video to microSD card.

Analog output can be then inserted to RCA input of television of into FPV transmitter.


Figure 19 - Caddx Loris 4K analog FPV camera

- Configuration:
  - Operating voltage: 5 to 16 V

2.2.15 TBS-UNIFY PRO32 NANO 5G8

The TBS-UNIFI PRO32 NANO 5G8 is a miniature analog FPV transmitter.

The transmitter itself has a built-in U.FL connector for antenna connection.


Figure 20 - 2.2.15 TBS-UNIFY PRO32 NANO 5G8 FPV analog transmitter

2.2.16 104050

The 104050 is a Lithium Polymer accumulator with 2500 mAh capacity.

Lithium polymer accumulators generally offer great capacity to weight ratio, high discharge current and come out in easily integrable shapes.

LiPos used in our CanSat come out with integrated safety circuit, that checks overcharge, overdepletion and overcurrent.

Figure 21 - 104050 Lithium Polymer accumulators

2.2.17 AS7265x

The AS7265x is a unique type of sensor for measuring intensity of light on 18 different channels and due to that used as a spectroscope.

The sensor in fact consists of 3 separate ICs. AS72651, AS72652 and AS72653. Each IC measure in different part of light spectrum.

The triad is and I^2C slave device. AS72651 is used to communicate with I^2C master, but it also works as an I^2C master for its own I^2C bus connecting the rest of the triad.

2.2.18 BH1730

The BH1730 is light intensity sensor of the primary visible light spectrum, but also of the infrared spectrum.

- Configuration:
  - Operating voltage: 3.3 V
  - Bus: I^2C_0
- Measured data:
  - Light intensity:

Figure 22 - The BH1730 is a sensor which measure light intensity

2.2.19 WS2812E

The WS2812E are addressable RGB LEDs.

Each LED require 3 bytes of information about color intensity of each color. LEDs can be mounted in series, when each LED trims out 3 bytes and sends rest of message to next LED.

- Configuration:
  - Operating voltage: 5 V
- Bus: NeoPixel

Figure 23 - WS2812E addressable RGB LED

2.3 PCB

To avoid having a whole lot of cables in your CanSat for inconvenience's sake, a PCB (Printed Circuit Board) is ideal. A PCB can host a lot of sensors without taking much space.

This year we are using multiple four-layered, double-sided PCB's to conserve as much space as possible. To double down on this, we also fully integrate our sensors into the PCB's. Our PCB's now don't include any commercially available sensor's breakout boards.

The specific PCB's are the Power Unit, the Control Unit and the Side Unit.

2.3.1 Control Unit

The Control Unit, as name suggests, is a PCB dedicated to control the whole CanSat. The PCB groups peripheries that only makes sense when used together (for example: microcontroller and RTC).

The Control Unit is capable of stand-alone completion of primary mission when suitable power source is provided.

USB-C has huge advantages over other USB type connectors. Nowadays it is the most used connector offering highest transfer speed (that is not used due to speed limitation of USB to UART converter), highest current and it is over time the most mechanically stable USB type connector.

Two buttons are mounted in case that the auto uploader does not work.

The CP2102N converts USB data to UART. 2 LEDs are used to indicate active communication between the microcontroller and the host computer.

A Square hole is placed in the middle of the board because of a SMA connector placed on top plate and its connector extends through top unit. U.FL must go through the Control Unit, so hole size and shape is adjusted to this.

Three battery connectors are used to created contact between the Control Unit and each Side Unit.

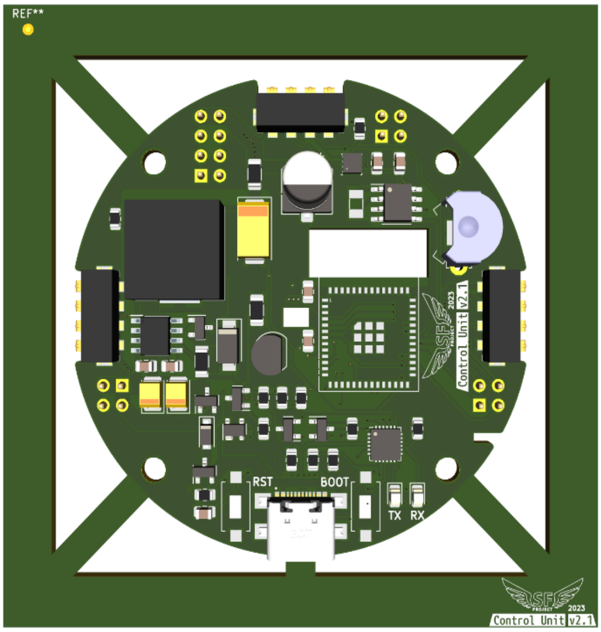


Figure 25: Control Unit-front side

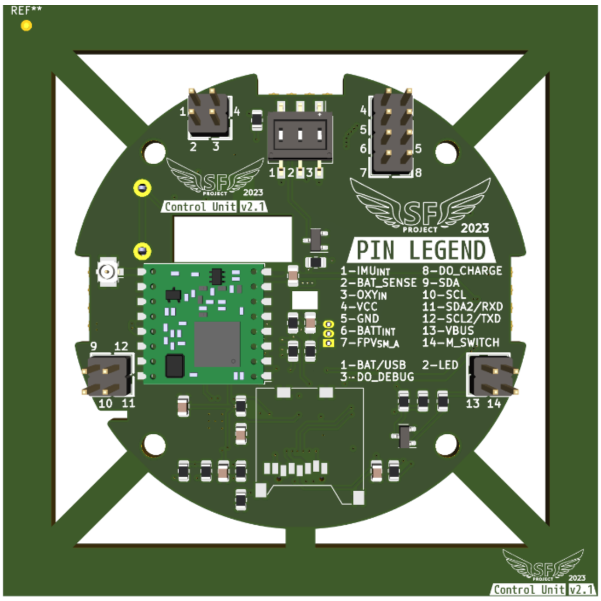


Figure 24: Control Unit-back side

A separated ground plane is created for DC-DC converter. This ground plane is than connected to global ground through 0 Ω resistor.

A push-push microSD card slot is placed at the bottom side of Control Unit, right under the USB-C connector. Thanks to this, the microSD card can be removed without the need of disassembling whole CanSat construction. The push-push slot has an advantage over push-pull slot, that being that the inserting and removing card is easier and the slot can be placed further from board' edge.

A U.FL connector is used to connect RFM96W to the unique CanSat antenna at the bottom of construction. The track connecting module to this connector is narrow as possible to limit RF leakage.

DIP switches are used control some of CanSat’s features.

In the integration manual of ESP32 microcontrollers is a mention, that at least 15×10 cm spare area is needed for proper Wi-Fi functions. This is not possible in CanSat so right under PCB antenna of ESP32-MINI-1 square hole is created to try balance out this design violation.

The PT7C4339 requires some kind of backup power source. It is nearly impossible to find miniature a 3 V power source. One option was to use a supercapacitor and keep it charged with built-in trickle charge function. In the end, two different 1.55 V batteries are placed into battery holder originally designed to hold only one battery. To avoid a short circuit, little bits of electric tape are used to prevent batteries from an unwanted contact.

In a bit of spare place, a pin and a DIP switch legend is printed out to make working with the CanSat easier.

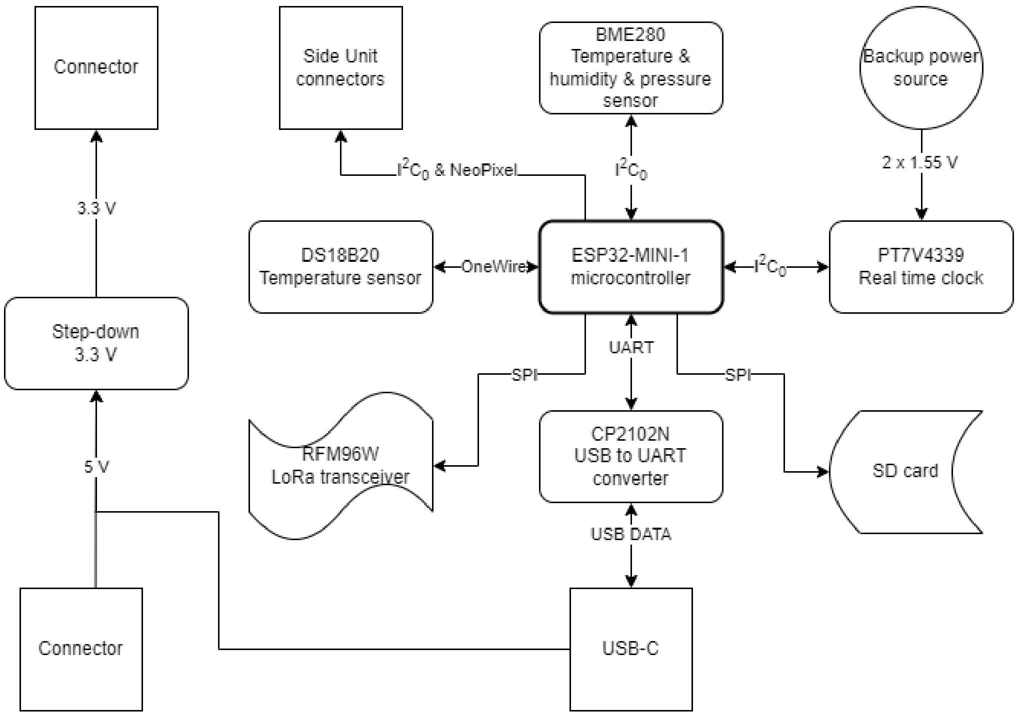


Figure 26: Control Unit-Block diagram

2.3.2 Power Unit

Power Unit v2 is a succeeder of Power Unit v1 from CanSat 2022, whose purpose was solely to regulate and distribute power across other PCBs. Power Unit v2 still use this name although beside power distribution it hosts most of the CanSat's sensors and peripheries.

The accumulators are connected through JST-PH connectors. Charging and balancing of those accumulators is done via external charger and balancer connected through 3-pin JST-PH. Charging is disabled when CanSat is on.

The input power is monitored with INA219A. Kelvin connection ensures that the current is measured with high accuracy.

An 8-bit GPIO extender controls some of sensors and peripheries. Spared pins are connected as inputs to some sensors.

Whenever the GNSS module is controlled with UART or I^2C_1 depends on the jumper position. By default, NEO-M9N communicates via I^2C_1. To indicate GNSS's fix LED is mounted. When power is removed, IC automatically switches to backup power. Impedance isolation of bus pins is required.

Same as on Control Unit, the MAX1626 regulates accumulator voltage to stable 5 V. This voltage then goes to Control Unit, where it is used to power up LEDs on Panel 2 & 3 and as input for another MAX1626 that creates stable 3.3 V. To filter out nasty high frequency noise on power line, set of tantalum capacitor is used.

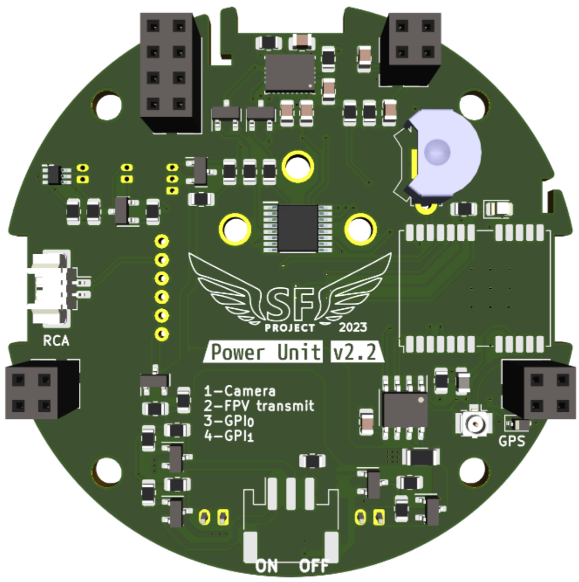


Figure 27: Power Unit-front side

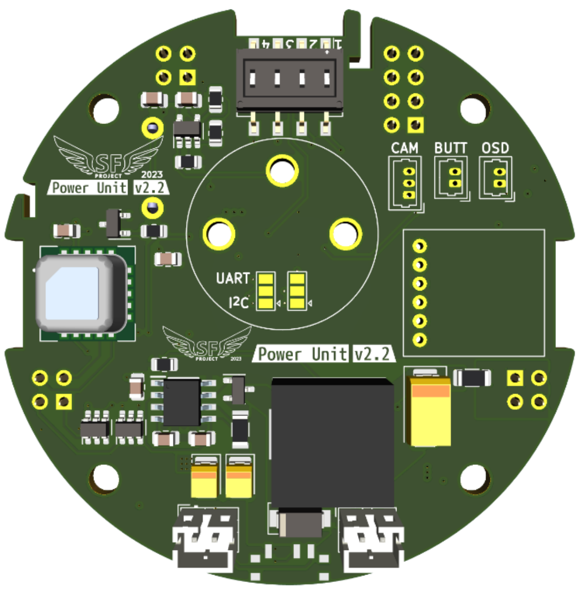


Figure 28: Power Unit-back side

The camera is connected to the Power Unit with set of Molex PicoBlade connectors. The camera's control board is modified to enable electronical controlling of the camera. The RX and TX pins are mounted on a control board, but manufacturer does not offer any documentation over this topic. The board doesn't react on any given baudrate. Camera is now controlled with transistor that "pushes" the built-in button.

The analog video output from the camera can be easily transmitted through a RCA cable. A Molex PicoBlade connector is mounted specially for this cable.

The FPV transmitter is mounted on pin headers to be easily removable and reusable. To secure it in place, a special platform is put underneath it.

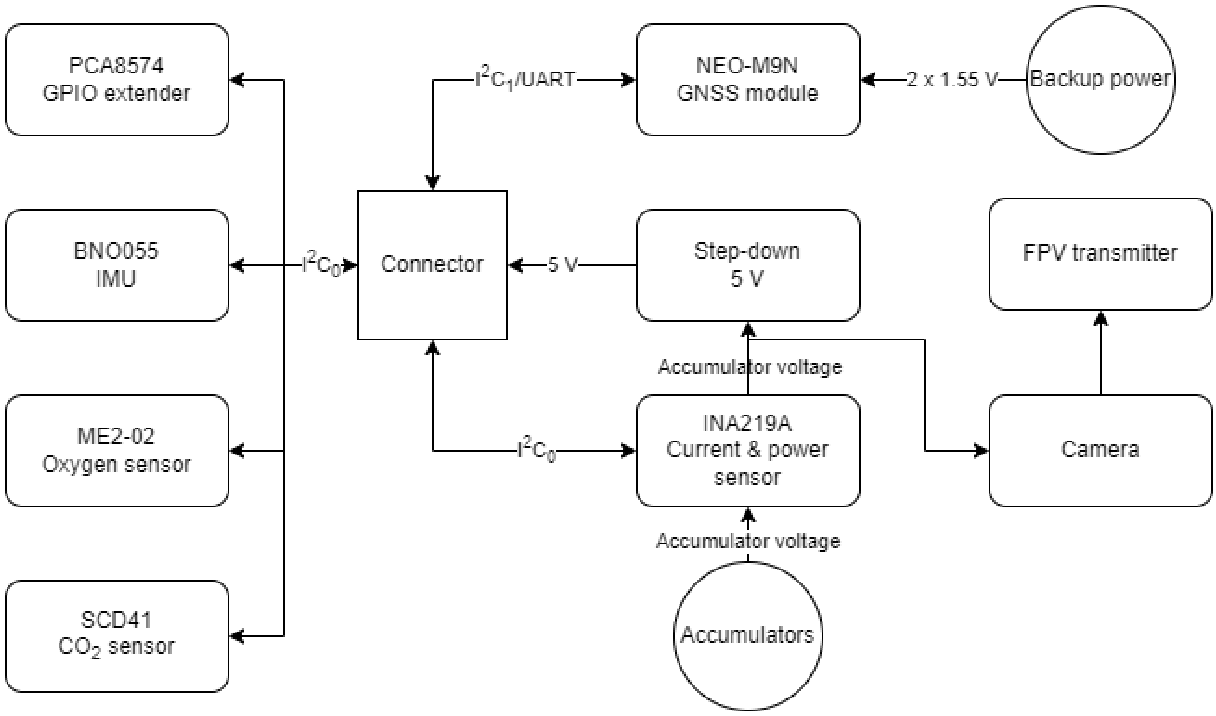


Figure 29 – A block diagram visualizing the inner workings of the Power Unit

2.3.3 Side Unit

Side Unit in fact consist of three separated board or as we call them, Panels. These Panels are placed perpendicular to other CanSat's PCBs and connected to the Control Unit through pads on back side. These pads create contact with battery connectors.

All light dependent sensors are placed on Panel 1. This includes a triade spectroscope AS7265x and a light intensity sensor BH1730.

Remaining two Panels are each filled up with eight RGB addressable LEDs. Every LED serves as a status indicator of each sensor or periphery. LEDs are normally turned off or lit up on low power. LEDs. WS2812E can be boosted up to ease finding CanSat in conditions with lowered visibility.

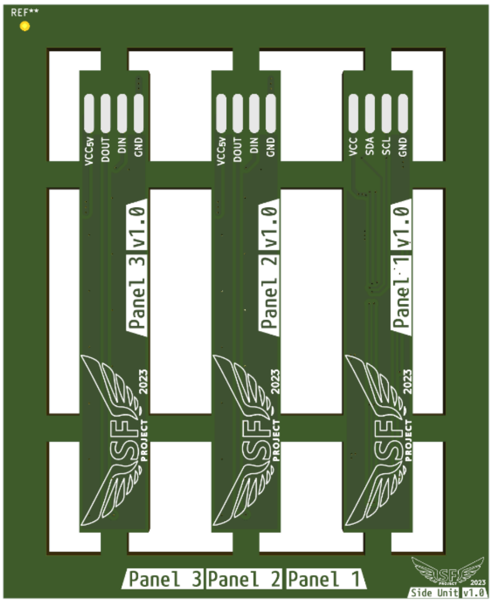


Figure 30 - The back side of the Control Unit with the visible battery contacts

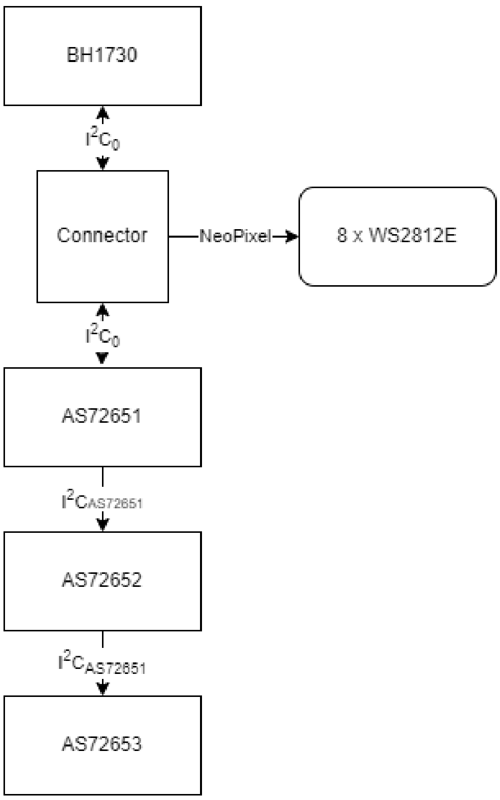


Figure 31 – A block diagram visualizing the inner workings of the Side Unit

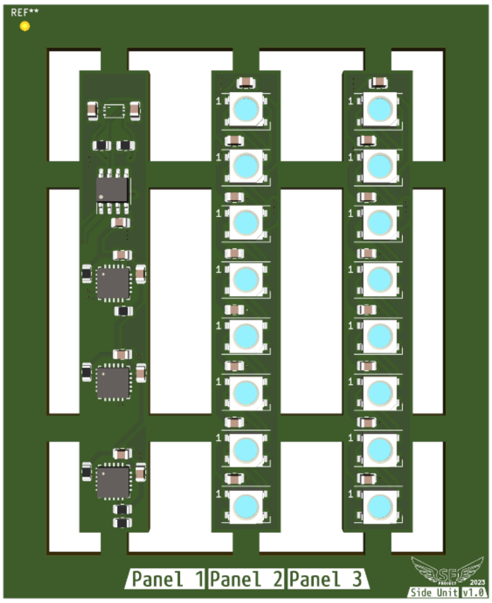


Figure 32 -The front side of the Control Unit including the light sensors and the LED's

2.4 Software

CanSat's software is written in Platformio, Visual Studio Code's extension. This extension is well known for offering Visual Studio Code environment and functions, runtime debug, Git source control, full Arduino's Wiring library implementation and much more, while keeping things easy to use like integrated compiler, uploader, terminal and Serial monitor.

Software itself runs dualcore with help of FreeRTOS operating system. Several security features like watchdog and wireless controlled interrupts are used to ensure smooth operation.

Each sensor has own class that makes software clear to read and more Arduino friendly.

Most of the sensors consists of three individual class methods: *setup*, *getData* and *printData*.

2.4.1 Setup

At the very beginning, all buses are started.

After that, each in dividual sensor and periphery runs through its own setup function.

Then things related to FreeRTOS are created like semaphores and tasks itself.

2.4.2 Get Data Task

This is the main CanSat task that handles getting data out from all sensors and peripheries, saving data to removable storage, and sending data to Ground Station.

Status of each sensor is set to *Status_NACK*.

Tasks responsible for handling slow peripheries are resumed for async read-out on second core.

Data are then read out from rest of sensors.

Software waits until task handling slow peripheries are done or until max wait time passes.

Data are then sent to Ground Station and saved to already opened allocation space on microSD card.

Task once again suspends itself until saving to microSD card is done or until max wait time passes.

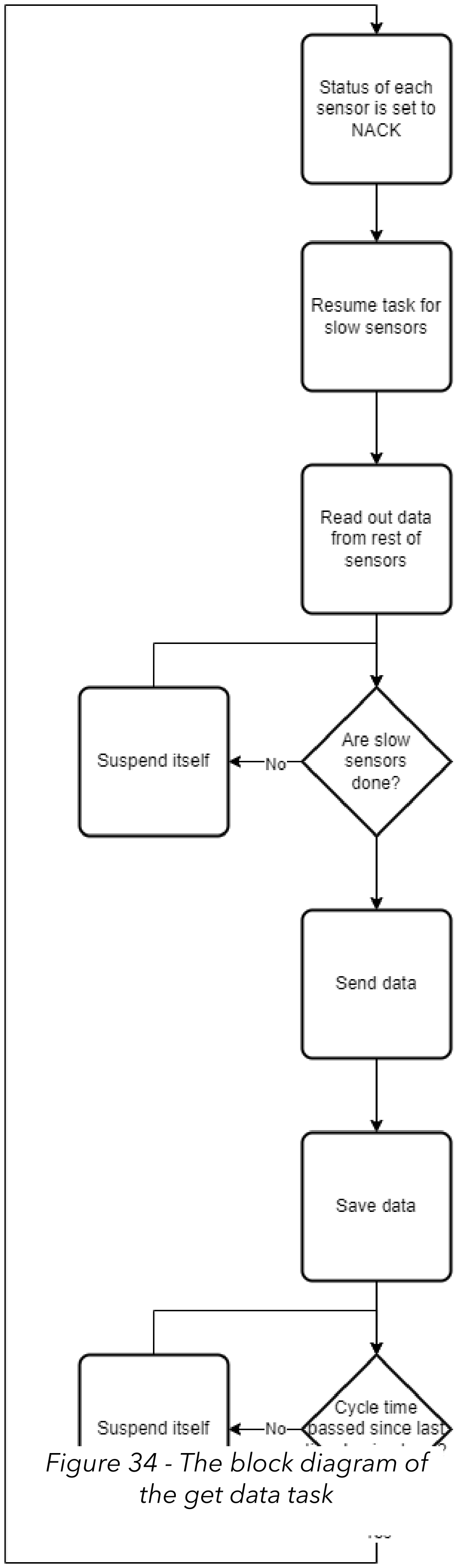


Figure 34 - The block diagram of the get data task

Figure 33 - getData function prototype

Status and measured data are then printed out from now resumed *printData* task.

Task finally suspends itself for duration remaining to start of next cycle.

3. Ground station

The ground station is a key component for our secondary mission.

The device is capable of double-sided communication and able to run the visualization script to run real-time data analysis. The ground station is mostly the receiving end to the double-sided communication between is and the CanSat.

3.1 Construction

The mechanical construction of the ground station is divided into two parts. The first part being the main circuit board cover with an openable top and the second part being the display and FPV receiver cover. Both parts are connected together with nuts and bolts.

3.1.1 Circuit board cover

The circuit board cover is made out of two parts, the top and the bottom part. Both parts are connected together by a threaded rod, which also works as a joint. On the bottom part there are holes for mounting the main circuit board. To keep the cover closed, M3 nuts are glues in bottom part in their printed holders. On the right side, there are four M3 holes for mounting the display cover.

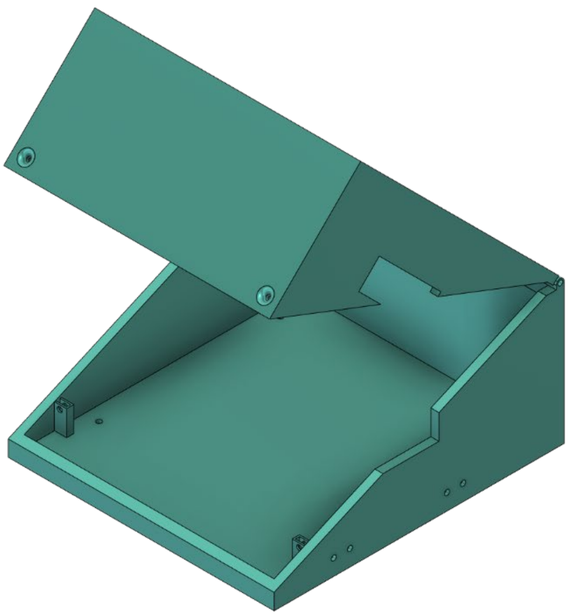


Figure 35 – The circuit board cover is used to house the internal electronics of the ground station.

3.1.2 Display and FPV cover

The display covers' main task is to hold the display and the FPV receiver. The display cover is made of two parts, that being the display mounted on the top part and the second task, which houses the inner electronics and holds the FPV receiver for video transmission. On the back of this cover there are antenna connectors and a power switch. This part is mounted directly to the first part.

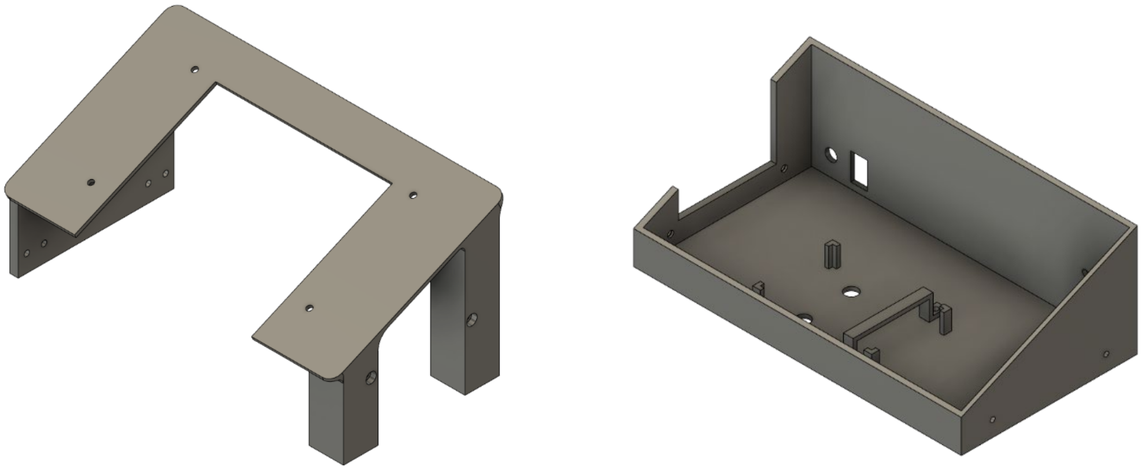


Figure 36 - The cover houses the inner electronics, while also holding the display

3.2 Electronics

The electronic part of the ground station fulfills the same purpose as the CanSat's, that being a sending, acquiring and also visualizing data.

The ground station is composed out of a computer, a screen and also sensors to measure data to calculate differences between the CanSat and the ground station. All of the electronics is mounted on a PCB.

3.2.1 Raspberry Pi 4B

Raspberry pi is the brain of the ground station. It is used to control all the internal and external modules. Raspberry pi is running on Raspbian which is a version of Linux.

Raspberry Pi contains 1.5GHz quad-core ARM Cortex-A72. This processor offers high performance and a lot of GPIO pins with support of various protocols. Graphic card VideoCore VI provides 4Kp60 video and HEVC/H.265 video decoding. Operating memory of type LPDDR4 and size of 8GB is shared with the graphic card.

On Raspberry Pi circuit board is generous amount of various type of connectors such as 2 micro-HDMI and USB 3.0.

- Configuration:
  - Processor:
    - Broadcom BCM2711
    - 1.5GHz quad-core ARM Cortex-A72
    - 64-bit SoC
  - Memory:
    - LPDDR4
    - 8GB
  - Connectivity:
    - 2.4GHz and 5GHz IEEE
    - Bluetooth 5.0
    - Gigabit Ethernet
    - 2 USB 2.0
    - 2 USB 3.0
  - Video and Sound:
    - 2 microHDMI
    - MIPI DSI
    - MIPI CSI
    - 3.5 mm jack
  - Multimedia:
    - H.265
    - H.264
    - OpenGL ES
  - Operating system:
    - Raspbian
  - Power:
    - 5V/3A DC USB-C
    - 5V/3A GPIO header
    - PoE

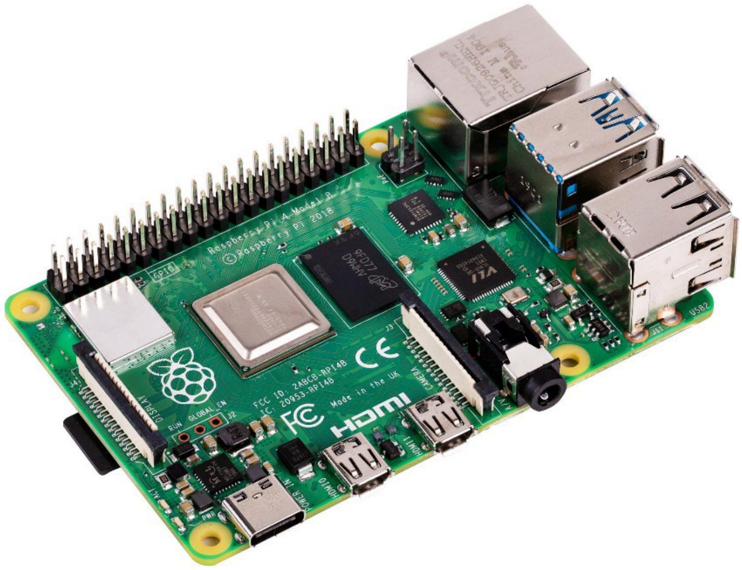


Figure 37 - The Raspberry Pi 4B is the brain of our ground station

3.2.2 RFM96W

The second most important part of the ground station is a module which provides real-time communication.

It communicates on the 433MHz frequency and the module uses a special LoRa modulation to make long distance data transmission possible. The module is connected to an external 6 element antenna.

This module also provides a security function called SyncWord. However, the lack of a python library using this function doesn't exist. This Is handled by hardcoding the function into the script.

- Configuration:
  - Operating voltage: 3.3 V
  - Bus: SPI
  - Transmitting power: 10 mW

Figure 38 – The RFM96W is a long-range communication module

3.2.3 EWRF OTG 5G8 receiver

A second data receiver integrated in the ground station is an EWRF module. It is used for receiving analog video transmission directly from the CanSat. It has its own 5.8 GHz antenna.

- Configuration:
  - Operating voltage: 5V
  - Connector: USB
  - Frequency: 5.8GHz
  - Number of channels: 48

Figure 39 - The EWRF OTG 5G8 receiver receives and transmits data on the 5.8GHz frequency

3.2.4 Raspberry Pi Touch Display

All acquired data are visualized on this display. It is an official Raspberry Pi branded touch screen. It supports 10 finger touch and an on-screen keyboard.

- Configuration:
  - Operating voltage: 5V
  - Connectivity: DSI
  - Display:
    - Size: 7 inch
    - Format: 800 x 480 px

Figure 40 - The Raspberry Pi Touch display is used to visualize received data

3.2.5 GY-NEO6MV2

NEO6MV2 is a GNS module from company U-BLOX. Ground station collects data such as a latitude, longitude, altitude, time and much more. It used to get an exact position of ground station and gain a relative position against CanSat.

- Configuration

- Operating voltage: 3-5 V
- Bus: UART

- Measured data:
  - Longitude
  - Latitude
  - Altitude
  - Date
  - Time

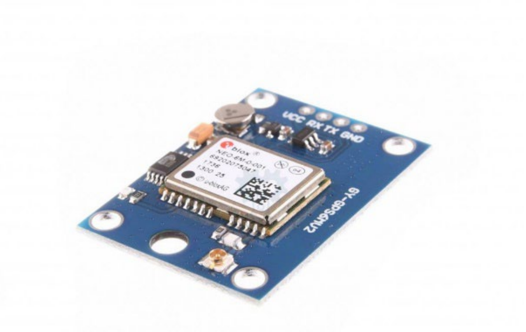


Figure 41 - NEO6MV2 is a GNSS module to acquire the ground station's position

3.2.6 BME280

BME280 is a sensor of humidity, temperature, and pressure from company BOSCH.

The measured pressure is used to calculate the ground station's altitude very precisely. From this information ground station can count a height of CanSat.

Measured temperature can be used for controlling the cooling system integrated in ground station.

- Configuration:
  - Operating voltage: 3.3V
  - Bus: I²C
- Measured data:
  - Pressure
  - Temperature
  - Humidity

Figure 42 -The BME280 is a sensor that measures the surrounding temperature, humidity and pressure

3.2.7 ADS1115

ADS1115 is an analog-digital converter which is used to measure voltage of batteries. It provides 16-bit accuracy at 860 samples per second.

- Configuration:
  - Operating voltage: 2-5V
  - Bus: I²C
- Measured data:
  - Battery voltage

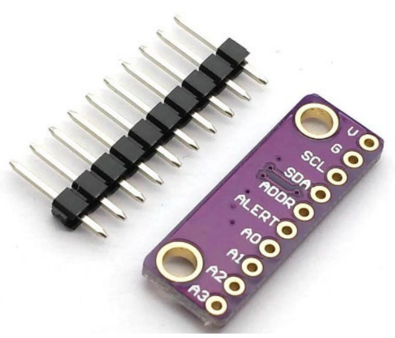


Figure 43 - The ADS1115 is an ADC which measures the voltage level on the batteries

3.2.8 Power supply

The power supply is divided into three parts. The first one is the batteries themselves. The ground station is using two batteries from Panasonic and they are connected in series. The second part is composed of a stepdown converter, which converts 7.4V on input to 5V on output. The last part is a switch and a USB-C connector.

- Configuration:
  - Batteries: Panasonic NCR18650B
    - Nominal voltage: 3.6 V
    - Capacity: 3350 mAh
    - Max. output current: 6.7 A
    - Type: Li-Ion
  - Stepdown converter:
    - Input voltage: 5.3 – 26V
    - Output voltage: 5V
    - Output current 3A
    - Efficiency: 96%

Figure 44 - The ground station is powered by two Li-Ion batteries in series

3.3 PCB

There are three printed circuit boards in the ground station. Two of them are simple breakup boards, which convert the 2mm spacing to 2.54mm. The last and most important one is the main board. The main board contains all electrical components in the ground station.

3.4 LoRa breakup board

This board is used to convert 2mm spacing of LoRa module to standardized 2.54 mm spacing. Eight and six pin headers are used as a connector into the main board.

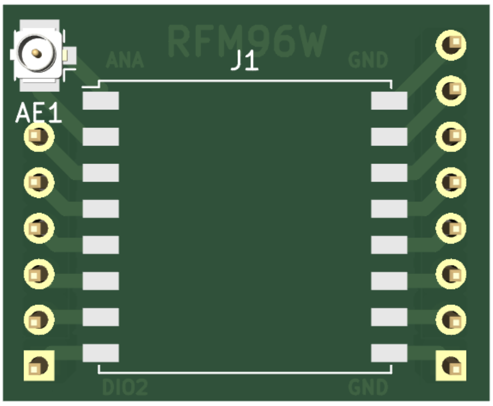


Figure 45 – The Lora breakup board serves as the adapter for the spacing

3.4.1 Stepdown breakup board

This board is used to convert SMD Stepdown converter to standardized 2.54 pitch. Two pin headers are used as a connector into the main board.

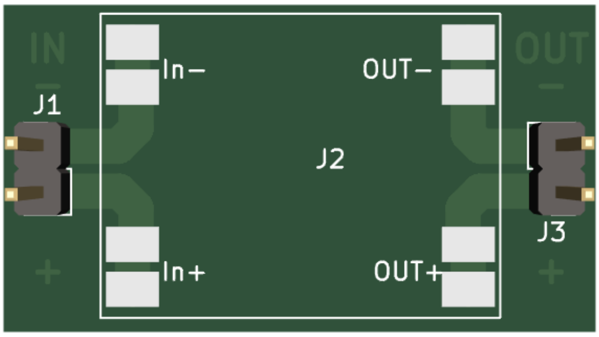


Figure 46 - The step-down breakup board acts as a SMD step down converter

3.4.2 Main board

This board connects every module together and contains control elements. There is two parts on this board.

First part is power supply. On the board there is a battery holder mounted for two batteries. These batteries are connected in series and led to the stepdown breakup board input. The negative path is brought up to the power switch. From the stepdown breakup board output is the voltage led to the power connector.

Second part is the control and sensor section. Raspberry pi is located there, and it is connected to the other sensors. Power for the sensors is also distributed in this part.

Connectors used for the power switch and for the power connector are JST-XH. From these the power is distributed into desired places.

All modules and breakup boards are attached to the boar by M2 and M3 distance pillars. This makes the ground station quite stable.

Display and FPV receiver are connected directly to the Raspberry Pi. Display image is connected via DSI flat cable. Power of the display is secure by USB power cable. FPV receiver is wired up using data USB 3.0 cable.

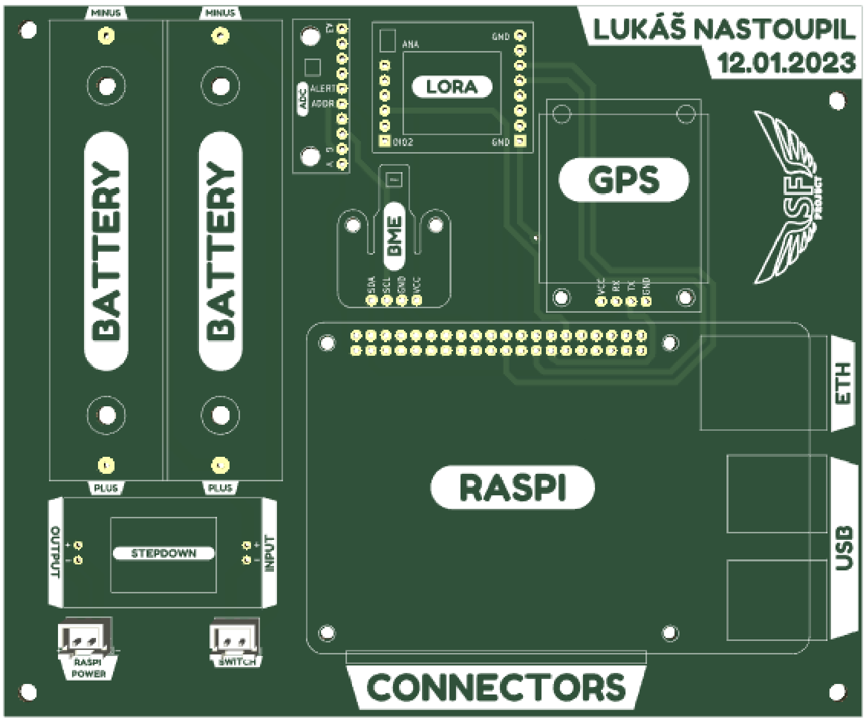


Figure 47 - The main PCB of the ground station

3.5 Software

The Ground station is programed in Python. Python is by far the most used language on Raspberry Pi. There also is a giant community around this language. The program is divided into four separated scripts. Each one can work independently of the others.

3.5.1 I^{2}C communication script

This script handles communication via I^{2}C bus with BME280 and ADS1115 and reads data out of them. The script reads data out of the sensors and then edits them the into desired form and then they are saved into a file for the next use.

3.5.2 UART communication script

This script handles an UART communication between the Raspberry Pi and the GNSS module. The communication is established on port ttyAMA1 which is the second hardware UART port on Raspberry Pi. The baudrate is set to 9600.

Each message from the module is recognized by the code at the beginning of the message. Desired messages are decoded and received, data is saved into a file for further use.

3.5.3 SPI communication script

This script handles communication between Raspberry pi and the LoRa module connected on SPI bus. It starts with hardcoded a sync word to one of the registers of the module. After the LoRa module receives the packet, it sends the response immediately. Then the received packet is decoded and checked by implemented check sum. The last step is to save data into a file. This script also controls the communication in the opposite way. So once there is a command typed into a GUI script, it will load the command into the buffer. In the next cycle, the command in the buffer is sent with the response message.

3.5.4 Data editing script

The final script's task is to load all the data from the files and reorganize it in the desired order. The remaining data are also calculated, such as the height between the ground station and CanSat. Lastly, organizing the data into into a final file, which is used by the GUI. This script also saves the flight data over time for the flight analysis.

4. Telemetry

An important part of CanSat's mission is telemetry, that is responsible for bidirectional communication with Ground Station.

Telemetry runs on two frequency bands.

4.1 433 MHz

All measured data are sent with LoRa™ modulation on 433 MHz frequency.

The maximal allowed transmit power is established by Czech laws to 10 mW. When operated by person with necessary certificates, transmit power can be boosted up to CanSat's maximal 50 mW.

Transmitting and receiving is done through same antenna that has been specially designed for our CanSat. Type of this antenna is called "Inverted F". It is a small antenna with high gain and good shape parameters.

Figure 48 - CanSat's inbuild Inverted F antenna

The Antenna is built-in to CanSat, and it is not exceeding CanSat's height. In fact, antenna is barely visible.

The Ground station receives and sends data through 5 elements Yagi antenna build from PVC pipe. Antenna is then connected to Ground Station through 3 m long SMA cable.

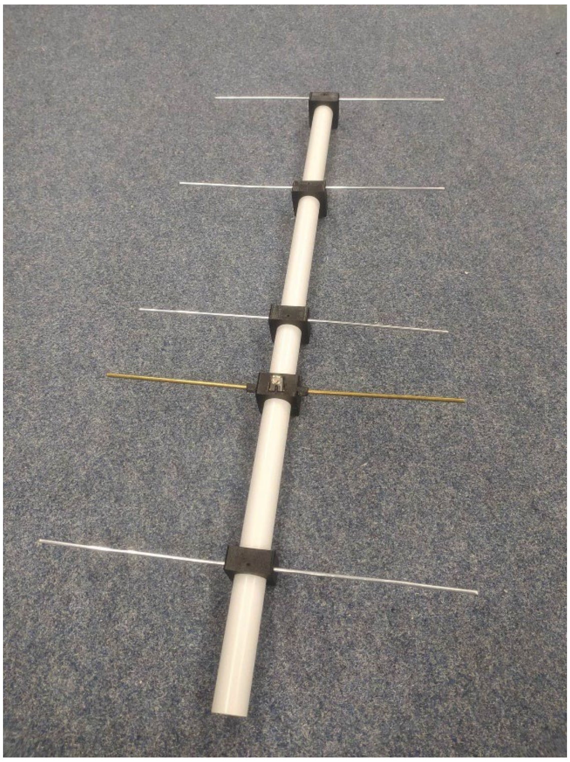


Figure 49 - 5 element Yagi antenna for Ground Station

4.2 5.8 GHz

This band is used for CanSat's FPV video transmitter.

The maximal transmit power established by Czech laws is 25 mW, but same facts in 433 MHz applies here. Maximal transmit power could be as high as 300 mW

A small omnidirectional SMA antenna is mounted on top of CanSat. This small antenna than handles FPV transmit.

Ground Station is equipped with directional 5.8 GHz antenna, that will significantly boost transmit range.

Figure 50 - CanSat's 5.8 antenna

Figure 52 - Ground Station's 5.8 GHz directional antenna-top view

Figure 51 - Ground Station's 5.8 GHz directional antenna-rear view

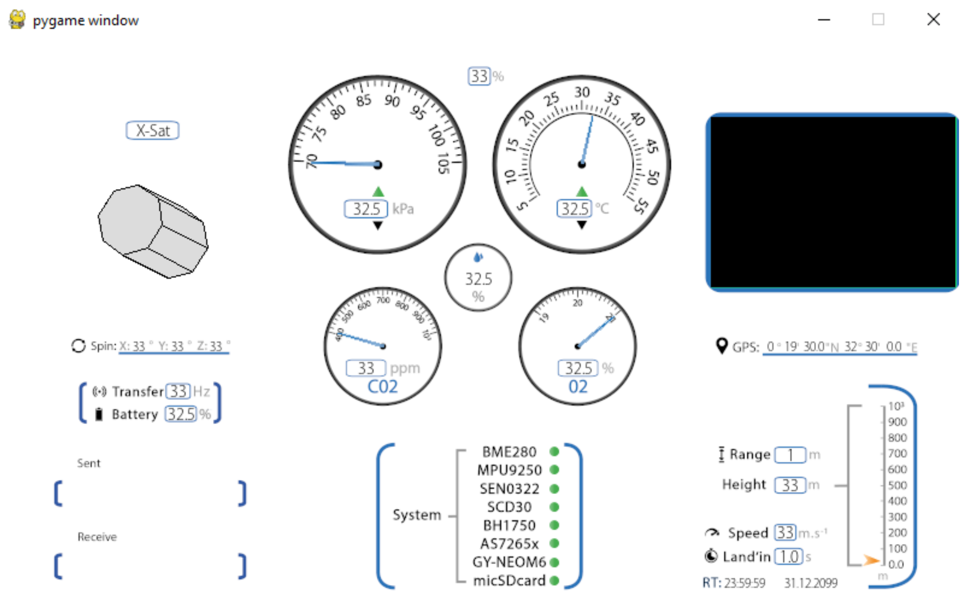
5. GUI

GUI is a shortcut for Graphical user interface. It is a type of interface that allows users to interact with the script through graphical icons. In our project, we use GUI as a tool for visualizing acquired data. The visualization script is running on the ground station. It is coded in Python using the library Pygame. Charts are created using the library Matplotlib.

Our program is divided into 3 main phases.

5.1 First phase

The first phase is for visualizing data in real time measured by the satellite during the fall. GUI will also be showing the data after the satellite lands.

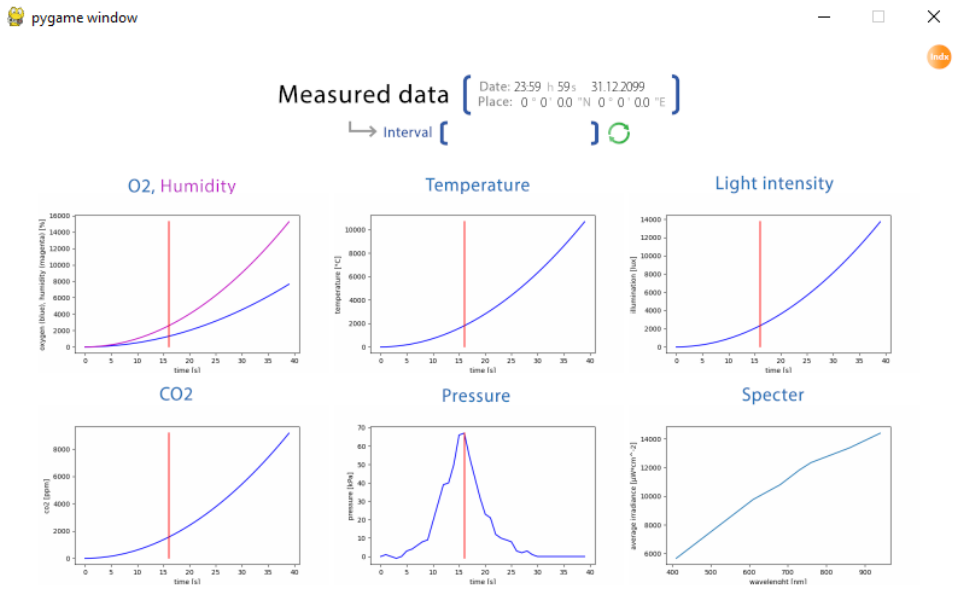


We are visualizing many variables on dials and with text. In the top right corner, we have camera video from CanSat and in top left corner model of CanSat, on which we show rotation of our satelite. This model is computed as eight-gonal prism. We use multiple symmetries to reduce the number of computed points to three.

By the time we finish demonstrating the control over the CanSat on long range, the program will move to the second phase.

5.2 Second phase

Second phase reports stored data that the ground station received during the first phase. This phase is designed for making a preview of received data through charts and their following modification.



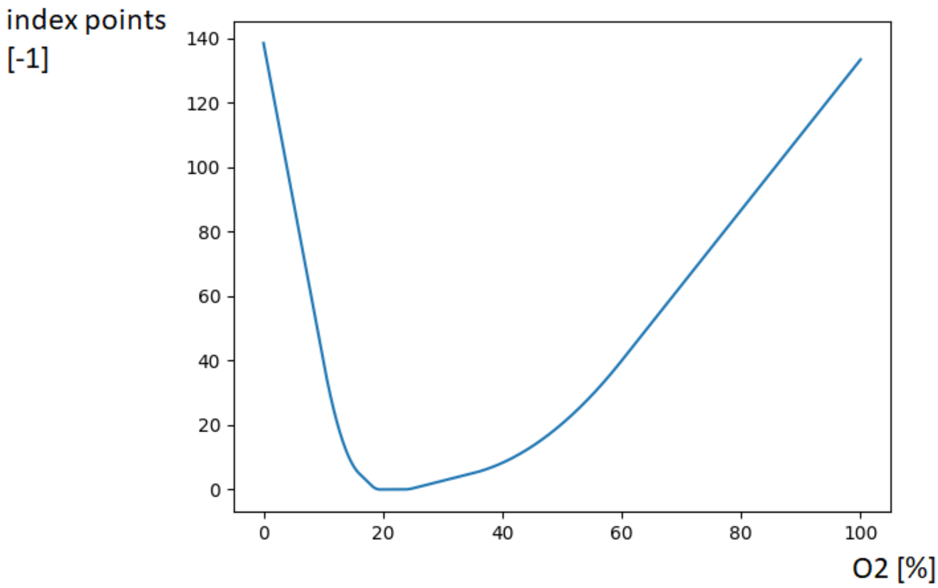
When starting this phase, the program will offer us how to cut the data. This is computed by development of pressure chart to get the most relevant for last phase. If we are not satisfied with the offered cut or we see any damaged data, we can adjust the cut using time interval. If we delete relevant data by mistake, we can always reload all data.

5.2.1 Index of habitability

From the second phase of the GUI we can run computing of index of habitability of planet CanSat is landing on. For most precise computing, we must carefully choose how to cut data in second phase, as any damaged or unwanted data could result in wrong value of this index.

For computing we decided to use data measured after CanSat landed on the ground, because it is where people live, so we get the most relevant data.

The index is computed from values of oxygen and carbon dioxide in the atmosphere, atmospheric pressure, temperature, humidity and light intensity. Using relevant information, we created dependances to compute index.



For example, here is dependance for oxygen values in the air. On x axes we have measured percentage of O2 in the air, on y axes there are points we subtract from the final index. We found on the internet many articles about ideal values, for example here. For oxygen values of 19.5 to 23.5 percent, which is normal value on Earth, so we subtract no point from the final score. In lower values for people to breathe, for higher values it's much easier for dire to start and for much bigger values air can be explosive, so farer the oxygen value is from its normal values, more points we subtract from the final index.

As you can see, these dependencies are made from linear functions connected together with quadratic functions. These are precisely computed to match slope of the function using derivations. Similar dependencies are calculated for all of the variables. The number of points that can be subtracted from the index because of one variable is limited to 50 or 80 (depending on importance of the variable). For example, temperature and light intensity have lower importance as they can be pretty easily compensated thanks to modern technology, while amount of O2 is hard to change.

The final index then has values from 0 to 100, where 100 is the best score, planet is similar to Earth and people can live there without any problems, 0 is planet, that would be hard to colonize as more variables are not properly satisfied.

5.3 Graphic part

The graphical part of the GUI was designed in Illustrator version 2021. Illustrator is a vector graphics editor and design program held by Adobe. The proportion of the GUI is 1080x1920 px, it's the same format as a Web layout for example. Colors are in the RGB regime.

The first phase of the GUI is divided into 3 notional parts: left, center and right, separated by spaces for better organization. Each part of it visualizes a different type of data.

Left part of the first phase gives us essential information about status of the CanSat by:

- Rotation
- Transfer speed
- Capacity of a battery

3D model of the CanSat serves as a digital twin. In this case digital twin is a virtual representation of CanSat that rotates the same way as actual CanSat does in real-time. This is another important factor, when it comes to

Besides the essential data there is additionally a command line, where we can write our command. Right under the command line there is a similar line where we see how the CanSat reacts.

Center part visualizes measured data of:

- Grand station battery
- Altitude
- Temperature
- Humidity
- Carbon dioxide
- Oxygen

We have decided to visualize some of the data in an analog form for a better overview of value changes (decreasing or increasing). However, this option isn't accurately indicating data, therefore we show these data either in a digital form with one decimal number.

Small triangles upon the altitude and temperature indicators signalize increasing and decreasing of values.

Bottom of the center part also indicates functionality of all modules used in the CanSat:

- BME280
- MPU9250
- SEN0322
- SCD30
- BH1750AS7265x
- GY-NEOM6

The green point next to the sensor indicates that Ground station successfully receives data from the particular sensor. In the other case point is red so the Ground station isn't receiving any data from the this sensor. See more about sensors in the chapter CanSat electronics.

Right part of the GUI shows another more information about falling CanSat:

- Live video
- GPS location
- Distance from the Ground station
- Height
- Speed of the fall
- Time to land

By pressing an arrow on keyboard, connected to the Ground station via Bluetooth, the program moves to the second phase.

The second phase shows charts with following data:

- O2
- CO2
- Humidity
- Temperature
- Pressure
- Light intensity
- Specter

In case of there is a problem with receiving data from particular sensor, there appears an error message instead of a chart.

There is additional information above the charts. Whenever you are measuring some data it's important to know the place and date. In the gaps next to the "interval" you can type time interval of data you want to see.

6. Recovery

Every CanSat in the competition should be able to survive more then one trip in the rocket. A large contribution to this probability is to have some sort of a recovery mission.

While there are many ways of safely getting your satellite from the sky back on the ground safely, by far the easiest is to make a parachute.

6.1 Parachute design

There are many important things to take into consideration while making a parachute, most notably what shape you want your parachute to be, how fast you want the CanSat to fall and how you want to attach it.

6.1.1 Parachute shape

The parachute's shape is mostly determined by the way you want to attach it to the CanSat. For our purposes we chose an octagonal shape since we only have four structural rods and a square parachute is rather impractical.

The shape of the parachute also changes the drag coefficient C_d. Octagonal parachutes have a drag coefficient of 0.8.

6.1.2 Physical properties

The rules of the competition subject us to make a satellite in the mass range of 300-350g. After some thinking we estimated that our CanSat will have a mass *m* of 330g, so we went with that.

We are aiming to achieve a falling speed *v* of 8m/s, which is the lowest that the competition rules allows us to. The reason behind this exact speed is to have the longest flight possible for gathering as much data as we can.

However, these parameters aren't the only ones regarding the calculations, but are the main ones nonetheless. Other parameters are the air density *ρ*, since the air get less dense the higher you go. The same happens with the gravitational acceleration *g* which also decreases the higher you go

The area of your parachute is determined by this equation:

$$S_p = \frac{2mg}{v^2 \rho C_d}$$

Figure 53 – The formula used for calculating the area of the parachute

6.2 Parachute manufacturing process

Now that the parachute's area is calculated, the process of making a parachute can begin.

The parachute is made out of parachute nylon, which is ideal because it is light and durable, this year we chose a pink parachute, since it is much easier to lose int the sky.

The shape of the parachute is drawn onto the sheet of nylon and then cut out via a soldering iron, this is because the heat prevents the edges from tearing. A spill hole is also cut out to balance out the parachute.

This year, eight strings are used to attach the CanSat to the parachute. The strings are tied into the parachute via clews. Clew are place on the edges of the parachute.

Figure 54 - A parachute prototype that was tested

7. Testing

Testing is an important part to every project.

In our case testing was done in many places over the time .

7.1 Technecium's stratoballoon

Our CanSat's electronics was sent into the stratosphere via a stratospheric balloon.

Thanks to our Lo-Ra module the balloon was later later after the landing.

Figure 55 - Our CanSat's electronics were present in Technecium's stratobaloon

7.2 The national finale of 2022

In the national finale of 2022 our CanSat worked fine the whole way. We got data from the beginning to the end of the launch and the fall. Our CanSat collected the data and the visualization then made graphs made out of them.

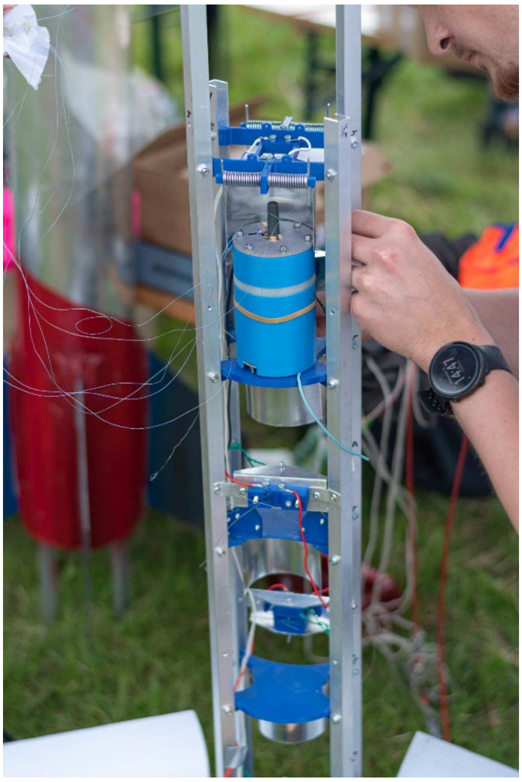


Figure 56 -Our CanSat being inserted into a rocket in the National finale of 2022

7.3 2023 Semifinals

In the CanSat semifinals of 2023, we tested our communication and our ground station.

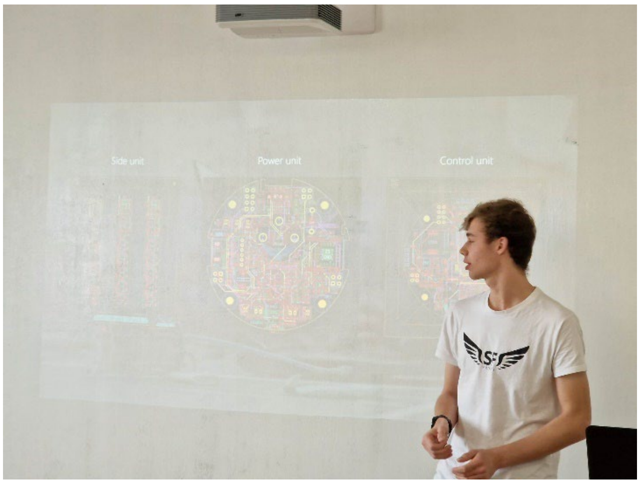


Figure 57 - David presenting on the 2023 semifinals

7.4 Other competetions

We also have taken place in more competitions where we had to upgrade our double-sided communication to show it off to other people and potential sponsors. The transmission was being tested there. Data was sent and visualized.

Figure 58 - Our Cansat on Maker Faire 2022

7.5 Technecium sessions 2023

Most of our work on the CanSat was made in the 2023 Technecium sessions. We once again tested our double sided communication and sent packages back and forth. We also tested our new CanSat boards, which work as intended.

7.6 2023 Parachute testing

Our parachute attached to a bottle filled up with water was thrown off the Technecium building to find out if it will serve it's purpose. It feel down for 2 seconds from a 16 meter tall building. That means that the targeted falling speed of 8m/s was achieved.

Figure 59 - Our parachute once again on the roof of the Technecium building

8. Publicity

The goal of our publicity department is to inform as many people as we can about the CanSat competition through our project. To accomplish such a goal we use multiple social media accounts, our website, articles, posters, and also through other events, such as technical exhibitions. We are trying to make all of our posts and media in English for better readability for non-Czech speaking people.

We tend to advertise mainly to technical groups and companies.

8.1 Identity

Every group of people, from a large company to a small team, should have theit own identity before trying for any form of publicity. Identity should express culture and mainly make society unique. It may be for example the name of the organization, logo, or design that's specific to it.

Figure 60 SkyFall logo in black and white variation

8.1.1 Name

Our team's name is "Project SkyFall". This is the only right version of how it should be written. SkyFall is a conclusion of Sky and Fall, therefore there are two capital letters in it. Word "Project" is also part of the name. Based on our experience this is the most common mistake.

8.1.2 Logo

Our logo was designed in Illustrator version 2021.Logo mainly express fall of the CanSat and signifies team's name by capital letters S and F and small "project" bellow. There are two color versions, black and white. These colors represent our team the most.

8.2 Social media

Social media is one of our most significant, when it comes to propagation. We have active accounts on Instagram and Facebook.

8.2.1 Instagram

Our most attended social media platform is Instagram. There we created a public account: the_project_skyfall, by the beginning of the year 2022. On the profile we share news about our project and propagate the CanSat competition.

- Source: https://www.instagram.com/the_project_skyfall/

Instagram analytic tool insights doesn't visualize profile data that is older than 3 months. Therefore, we are going to share with you most of the profile numbers in interval of 3 months.

There were 357 accounts that we reached in this interval. Reach is a number of unique accounts that have seen your content, at least once.

49 of accounts that we reached, somehow interacted with our content. Interaction may be for example likes, comments, shares, or saves.

Total followers that we get since we've created the project account is 120. That is about 39 followers more than last year.

We have connected Instagram account with our Facebook website. That means these accounts are synchronized. So, when you want to release a post, you can synchronize this action and take it at the same time.

8.2.2 Facebook

Our Facebook website serves mostly for people who doesn't have an Instagram profile. Otherwise, we don't use it because there isn't our target group. Therefore, we don't have a significant impact here.

- Source: https://www.facebook.com/theProjectSkyFall

8.3 Website

Our website is written in a library which allows you to create your own server called Express. You can use more utilities such as graphs from values measured by CanSat and 3D model of our satellite.

We use Azure for hosting our website, although its highly probable we will change hosting service in the future, because it would be nice to have a matching domain for our website (e.g. www.skyfall.cz)

- Source: https://cansat-skyfall.azurewebsites.net/cs

Currently our website consists of five pages.

The first page you can see after entering our website is a home page. Home page holds general information about our project and mission. You can switch between particular pages through navigation:

- CanSat
- Team
- Gallery
- Sponsors

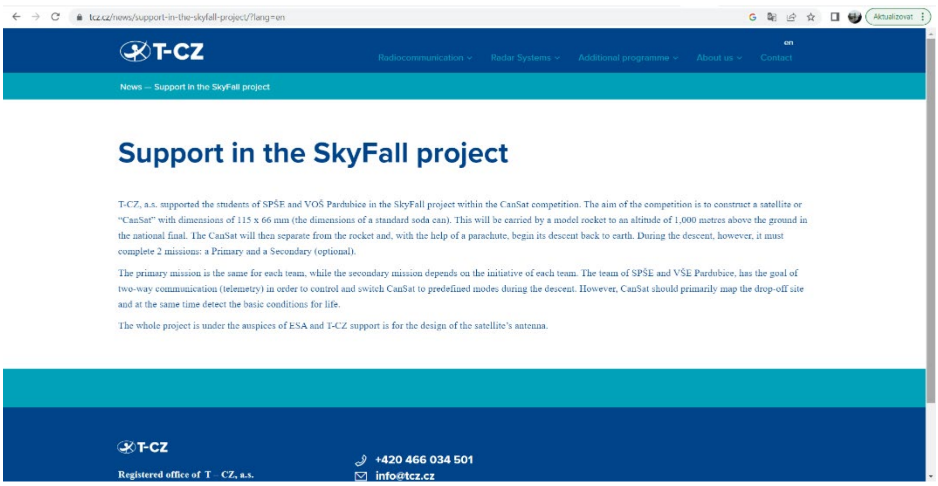
8.4 Articles

Since we have participated in CanSat for our first time in 2021 we have published several articles mostly on the websites of our sponsors. However, we have also published an article in a school magazine Elektronka.

8.4.1 School

Both schools provide us opportunities to propagate our project via websites. We also wrote an article in a school magazine Elektronka.

8.4.2 T-CZ


tcz.cz/news/support-in-the-skyfall-project/?lang=en

T-CZ

Radiocommunication | Radar Systems | Additional programme | About us | Contact

en

News — Support in the SkyFall project

Support in the SkyFall project

T-CZ, a.s. supported the students of SPŠE and VOŠ Pardubice in the SkyFall project within the CanSat competition. The aim of the competition is to construct a satellite or "CanSat" with dimensions of 115 x 66 mm (the dimensions of a standard soda can). This will be carried by a model rocket to an altitude of 1,000 metres above the ground in the national final. The CanSat will then separate from the rocket and, with the help of a parachute, begin its descent back to earth. During the descent, however, it must complete 2 missions: a Primary and a Secondary (optional).

The primary mission is the same for each team, while the secondary mission depends on the initiative of each team. The team of SPŠE and VŠE Pardubice, has the goal of two-way communication (telemetry) in order to control and switch CanSat to predefined modes during the descent. However, CanSat should primarily map the drop-off site and at the same time detect the basic conditions for life.

The whole project is under the auspices of ESA and T-CZ support is for the design of the satellite's antenna.

T-CZ

Registered office of T – CZ, a.s.

+420 466 034 501

info@tcz.cz

8.5 Events

We have participated in many events since we registered in CanSat for our first time. These events help us to earn new sponsors and propagate our mission. Previous year we took part at Maker Faire Prague where we get a new contact and get to know people with the concept of CanSat competition. This year we participated in Amavet competition, where we had also chance to show wide audience what is a CanSat.

9. Final words

We've created a satellite that is capable of measuring several important data, based on which we can decide the habitability of the planet. We are going to achieve it with double-sided communication between the ground station and CanSat and our developed program GUI.

This is our last participation in this competition therefore we will do our best so everything we wanted to will work and everything will go as planned.

We would like to thank all the external help, our sponsors, our schools and specifically Mr. Kašpar. Without this support, this project wouldn't have been possible.

10. Sources

What is the minimum atmospheric pressure that a human can live comfortably in (and not just survive)? - Quora. Quora - A place to share knowledge and better understand the world [online]. Dostupné z: https://www.quora.com/What-is-the-minimum-atmospheric-pressure-that-a-human-can-live-comfortably-in-and-not-just-survive

What air pressure can the human body survive? | Physics Forums. *Physics Forums | Science Articles, Homework Help, Discussion* [online]. Copyright © Physics Forums, All Rights Reserved [cit. 23.04.2023]. Dostupné z: https://www.physicsforums.com/threads/what-air-pressure-can-the-human-body-survive.333248/

Clarification of OSHA's requirement for breathing air to have at least 19.5 percent oxygen content. | Occupational Safety and Health Administration. Home | Occupational Safety and Health Administration [online]. Dostupné z: https://www.osha.gov/laws-regs/standardinterpretations/2007-04-02-0

Everything you need to know about humidity in your home. Airthings Homepage | Airthings [online]. Copyright © Airthings 2023 [cit. 23.04.2023]. Dostupné z: https://www.airthings.com/resources/everything-you-need-to-know-about-humidity

What's the hottest temperature the human body can endure? | Live Science. *Live Science: The Most Interesting Articles, Mysteries & Discoveries* [online]. Copyright © [cit. 23.04.2023]. Dostupné z: https://www.livescience.com/hottest-temperature-people-can-tolerate.html

11. Appendix

```
void getData(void *pvParameters){
    getData_lastTime = xTaskGetTickCount();
    while(true){
        rtc.status = Status::status_NACK;
        bme.status = Status::status_NACK;
        oxygen.status = Status::status_NACK;
        ina.status = Status::status_NACK;
        scd.status = Status::status_NACK;
        bh.status = Status::status_NACK;

        lora.status = Status::status_FAIL;
        sd.status = Status::status_NACK;
        gps.status = Status::status_NACK;
        ds18.status = Status::status_NACK;
        neo.status = Status::status_NACK;

        vTaskResume(gpsGetData_handle);
        vTaskResume(ds18getData_handle);
        rtc.getData();
        bme.getData();
        oxygen.getData();
        ina.getData();
        scd.getData();
        bno.getData();
        bh.getData();
        asx.getData();

        // Wait until other getData from diferent tasks are done or skiped
        if(!xSemaphoreTake(gpsGetDataDone_semaphore, pollingDelay)) //Serial.println("GPS timeout");
        if(!xSemaphoreTake(ds18GetDataDone_semaphore, pollingDelay)) //Serial.println("DS18 timeout");
        // Task are done or skiped when pollingDelay passes

        lora.sendData();
        sd.save();

        //vTaskResume(saveData_handle);

        if(!xSemaphoreTake(saveData_semaphore, dataPrintDelay)) Serial.println("SD save timeout");

        neo.updateStatuses();
        vTaskResume(printData_hadle);

        vTaskResume(openFile_handle);
        if(!xTaskDelayUntil(&getData_lastTime, refreshRate/portTICK_PERIOD_MS)) //Serial.println("Loop to slow!");
        cycle++;
    }
}
```

Figure 61 - getData task showcase

```
void setup(void) {
  Serial.begin(115200);
  Wire.begin(21, 22, 400000ul);
  Wire1.begin(19, 18, 100000ul);
  oneWire.begin(4);
  SPI.begin(33, 27, 14);
  SPI.setFrequency(20000000);

  pinMode(lora.cs, OUTPUT); digitalWrite(lora.cs, HIGH);
  pinMode(sd.cs, OUTPUT); digitalWrite(sd.cs, HIGH);

  pinMode(RUN_SEVER_PIN, INPUT);
  pinMode(37, INPUT);

  Serial.println("Core 0 restart reason: " + verbose_print_reset_reason(rtc_
  Serial.println("Core 1 restart reason: " + verbose_print_reset_reason(rtc_

  spiSemaphore_hadle = xSemaphoreCreateBinary();
  gpsGetDataDone_semaphore = xSemaphoreCreateBinary();
  ds18GetDataDone_semaphore = xSemaphoreCreateBinary();
  saveData_semaphore = xSemaphoreCreateCounting(3, 0);

  xSemaphoreGive(spiSemaphore_hadle);

  Serial.println('\n');
  printResult(rtc.setup(true));
  printResult(sd.setup(true));
  printResult(gps.setup(true));
  printResult(bme.setup(true));
  printResult(ds18.setup(true));
  printResult(lora.setup(true));
  printResult(ina.setup(true));
  printResult(oxygen.setup(true));
  printResult(neo.setup(true));
  printResult(scd.setup(true));
  printResult(bno.setup(true));
  printResult(bh.setup(true));
  printResult(asx.setup(true));
  printResult(pot.setup(true));
  printResult(pca.setup(true));

  digitalWrite(neo.enablePin, HIGH);

  xTaskCreate(runServer, "Run Server", 4096, NULL, 3, &runServer_handle);
  xTaskCreate(runNeo, "Run Neo Pixels", 2048, NULL, 3, &runNeo_handle);

  xTaskCreate(printData, "Print Data", 4096, NULL, 5, &printData_hadle);
  xTaskCreate(openFile, "Open SD file", 4096, NULL, 6, &openFile_handle);
```

Figure 62 - CanSat's setup showcase

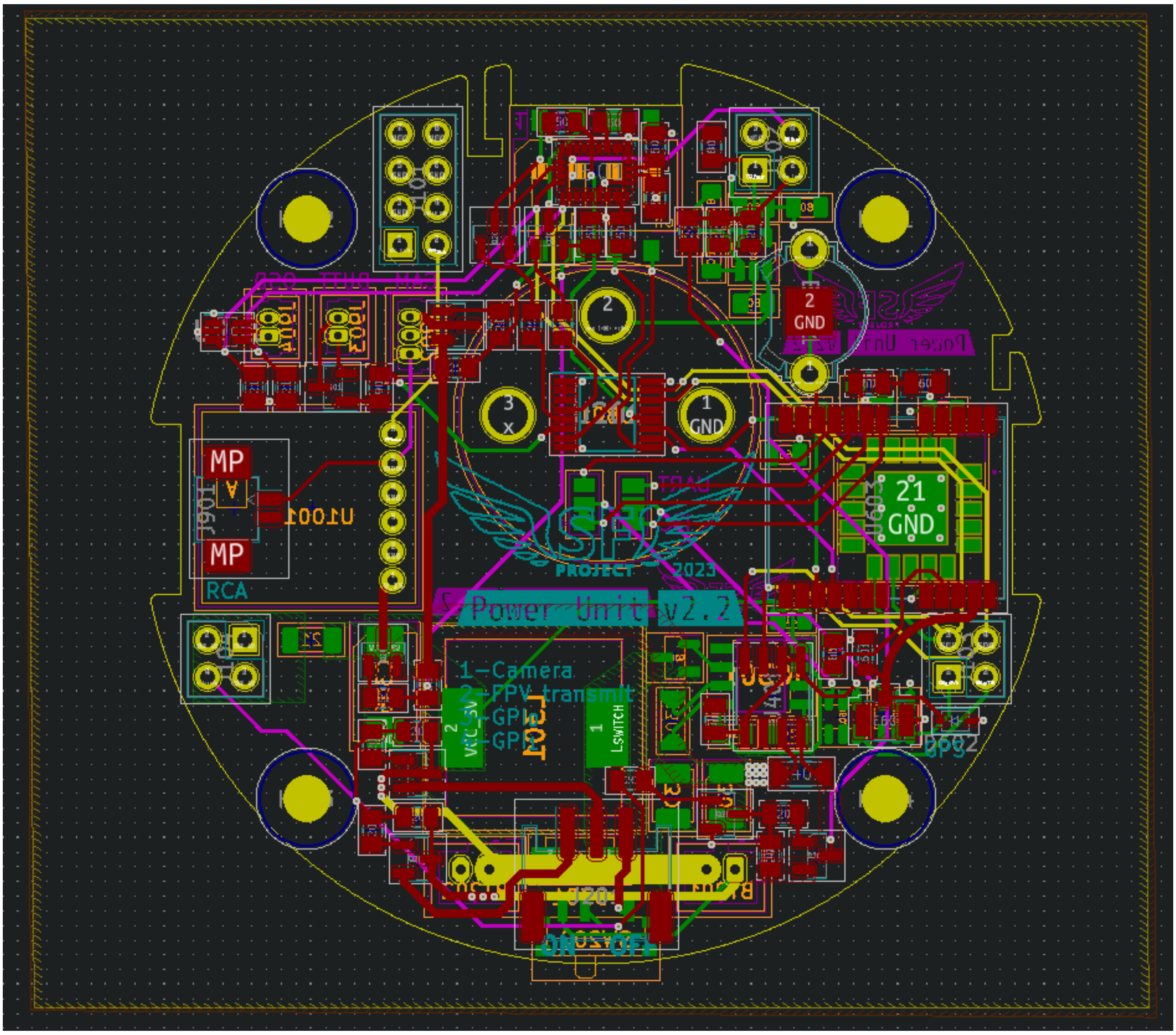


Figure 63 - Power Unit - KiCAD preview

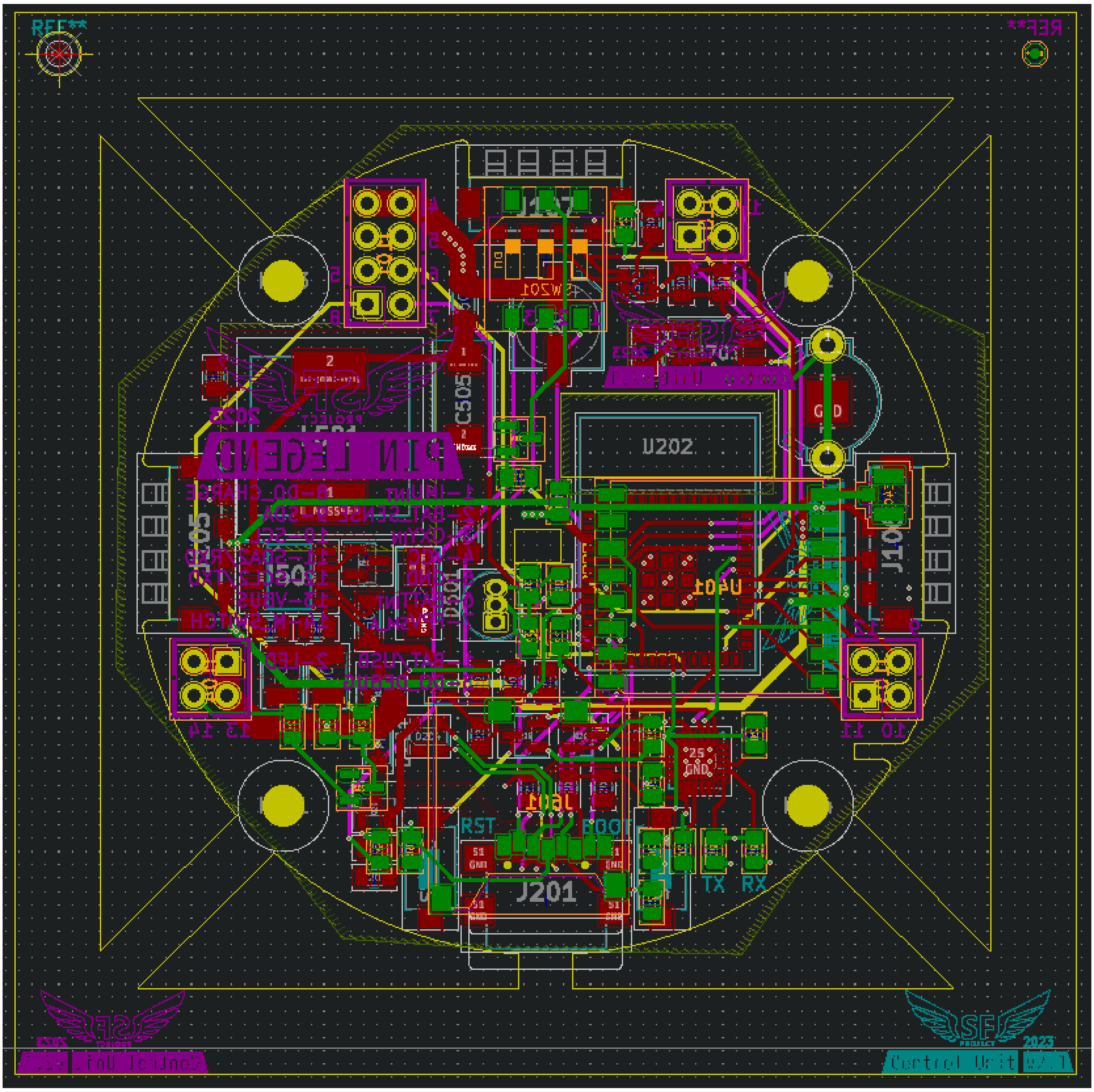


Figure 64 - Control Unit - KiCAD preview