BRNO UNIVERSITY OF TECHNOLOGY

Faculty of Electrical Engineering
and Communication

MASTER'S THESIS

Brno, 2024

Bc. Jiří Veverka

BRNO UNIVERSITY OF TECHNOLOGY

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

FAKULTA ELEKTROTECHNIKY
A KOMUNIKAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF RADIO ELECTRONICS

ÚSTAV RADIOELEKTRONIKY

EDUCATIONAL POCKETQUBE SATELLITE

VÝUKOVÝ SATELIT FORMÁTU POCKETQUBE

MASTER'S THESIS
DIPLOMOVÁ PRÁCE

AUTHOR
AUTOR PRÁCE
Bc. Jiří Veverka

SUPERVISOR
VEDOUCÍ PRÁCE
doc. Ing. Aleš Povalač, Ph.D.

BRNO 2024

BRNO UNIVERSITY OF TECHNOLOGY FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

Master's Thesis

Master's study program **Space Applications**

Department of Radio Electronics

Student: Bc. Jiří Veverka ***ID:*** 205866

Year of study: 2 ***Academic year:*** 2023/24

TITLE OF THESIS:

Educational PocketQube Satellite

INSTRUCTION:

Explore the design of PocketQube format nanosatellites suitable for education, with a focus on miniaturization in the 1.5p to 2p format. Specifically, investigate battery placement and develop an energy-efficient communication system. Strive for compatibility with [1] whenever possible. Design a concept for a simplified operational model of the PocketQube, including a block concept of the electronics with basic fail-safe measures.

Create circuit diagrams, relevant circuit boards, and prototypes of the individual electronic components. Integrate them into a unified PocketQube, along with a suitable payload, and handle internal interconnects. Verify the operation of the nanosatellite system, demonstrate its communication capability, and provide a detailed description of the results.

RECOMMENDED LITERATURE:

[1] KOŠÚT, Martin. Návrh a realizace výukového CubeSatu. Brno, 2022. Dostupné také z: https://www.vut.cz/studenti/zav-prace/detail/141541. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav radioelektroniky.

[2] CAPPELLETTI, Chantal, Simone BATTISTINI a Benjamin K. MALPHRUS. CubeSat handbook: from mission design to operations. San Diego, CA, United States: Academic Press is an imprint of Elsevier, [2021]. ISBN 978-0-12-817884-3.

Date of project specification: 16.2.2024 ***Deadline for submission:*** 20.5.2024

Supervisor: doc. Ing. Aleš Povalač, Ph.D.

doc. Ing. Tomáš Götthans, Ph.D.
Chair of study program board

ABSTRACT

This diploma thesis is focused on the design of a 2P (50×50×100mm) PocketQube type educational satellite and Ground station, which will be used in STEM laboratory classes. The proposed system was designed with emphasis on modularity, ease of replacement using commercial off-the-shelf components, and easy understanding of the subsystems one can find in a satellite of this type - On-Board Computer, Communication module, Electric Power System module, and Payload. Among these modules was developed an external structure, made entirely out of PCBs and together form the basis of the satellite. The developed satellite and Ground station utilize RFM98W radio module with LoRa modulation for communication.

KEYWORDS

On-board Computer (OBC), Electrical Power System (EPS), Communication system (COM), PocketQube, satellite, STM32, LoRa, RFM98W, Ground station, Payload

ABSTRAKT

Tato diplomová práce je zaměřena na návrh výukového satelitu typu PocketQube 2P (50×50×100mm) a pozemní stanice, které budou využívány ve výuce STEM laboratoří. Navržený systém byl navržen s důrazem na modularitu, snadnou výměnu za použití komerčních volně dostupných součástek a snadné pochopení subsystémů, které lze v družici tohoto typu nalézt, tedy palubní počítač, komunikační modul, napájecí systém a payload. Mimo vnitřních modulů byla vyvinuta i vnější konstrukce, která je celá vyrobena z desek plošných spojů a dohromady tvoří základ družice. Vyvinutá družice a pozemní stanice využívají ke komunikaci rádiový modul RFM98W s modulací LoRa.

KLÍČOVÁ SLOVA

Palubní počítač, Napájecí systém, Komunikační systém, PocketQube, satelit, STM32, LoRa, RFM98W, Pozemní stanice, Payload

VEVERKA, Jiří. *Educational PocketQube Satellite*. Brno: Brno University of Technology, Faculty of Electrical Engineering and Communication, Department of Radio Electronics, 2024, 114 p. Master's Thesis. Advised by doc. Ing. Aleš Povalač, Ph.D.

Author's Declaration

Author:	Bc. Jiří Veverka
Author's ID:	205866
Paper type:	Master's Thesis
Academic year:	2023/24
Topic:	Educational PocketQube Satellite

I declare that I have written this paper independently, under the guidance of the advisor and using exclusively the technical references and other sources of information cited in the paper and listed in the comprehensive bibliography at the end of the paper.

As the author, I furthermore declare that, with respect to the creation of this paper, I have not infringed any copyright or violated anyone's personal and/or ownership rights. In this context, I am fully aware of the consequences of breaking Regulation § 11 of the Copyright Act No. 121/2000 Coll. of the Czech Republic, as amended, and of any breach of rights related to intellectual property or introduced within amendments to relevant Acts such as the Intellectual Property Act or the Criminal Code, Act No. 40/2009 Coll. of the Czech Republic, Section 2, Head VI, Part 4.

Brno . . . .

author's signature*

*The author signs only in the printed version.

ACKNOWLEDGEMENT

I would like to thank the supervisor of my thesis, doc. Ing. Aleš Povalač, Ph.D. for his valuable comments and for leading this thesis until its successful end. I'd also like to thank to my wife Denise for her unwavering support throughout all of my studies.

Contents

List of Figures

List of Tables

Introduction

This diploma thesis is focused on a development of a PocketQube type educational satellite of size 2p (50×50×100mm). The satellite will be used as a learning platform for students in STEM labs to demonstrate the various subsystems of such a satellite and their purpose and operation. In addition to demonstrating the satellite, students will be encouraged to develop their own extension modules for this satellite format, or to create their own PocketQube satellite from scratch. A set of laboratory projects is included, and can also serve as a basic guide for the creation of other similar assignments.

The thesis begins with a theory chapter describing what the PocketQube is, its known applications, realized missions, a description of the various subsystems of the satellite, a brief description of the purpose of the ground station, and a mention of the CubeSat Space Protocol[6] that the resulting satellite and its system bus support.

The following chapter describes the design of the individual modules of the satellite, the ground station and the external structure and their components. The chapter 3 contains a description of the hardware implementation itself, the tools and programs and used components. At the same time, the first problems encountered during development are described. Chapter 4 describes the software of the satellite and Ground station, including the Python terminal application that interprets the received data. The next chapter is devoted to communication and a description of the module used, its configuration, the commands supported and the method used to test the RF communication. The second to last chapter is a short description of the proposed educational contribution and the last chapter describes the testing of the whole system and its results. The DeepL tool[7] was used during the solution of this work for checking the corectness of used grammar.

1 Theory

The PocketQube is a type of pico-satellite with a base unit of 1p (50×50×50mm) per 250grams of weight, and its structure is standardized by Alba Orbital, Delft University of Technology, and GAUSS Srl [8]. These satellites usually use a standardized COTS (Commercial off-the-shelf) modules, that are usually already flight proven for their application, which brings the costs down. To get the satellite to the orbit, a device called deployer box, which is a standardized device in which the satellite is placed and then as a payload on a rocket launched into the orbit. The final orbit is usually selected by the primary customer, since micro and pico satellites are usually a secondary payload. In the space the deployer opens and releases satellites out. To get as much as possible from the deployment, a ballast is usually placed inside the satellite itself, to give it more weight and thus higher momentum at the moment of the release. This helps the satellite to get to higher orbits and can prolong the life-span of the mission. Alba Orbital offers regular launch opportunities for prices starting at 25.000€ for 1p PocketQube[1][9].

1.1 PocketQube application

Just like CubeSats[10] and other micro-satellites, PocketQubes have similar mission applications. Thanks to the unified standard they are easily applicable in a wide range of scenarios. Their application can be categorized as[9][1]:

- Earth Observation (EO) missions - These missions are dedicated to observing the Earth from the space. This can be done for various reasons, eg. Traffic monitoring, weather prediction, monitoring and optimizing the naval routes, etc. Satellite in this configuration usually contain a high-resolution cameras for data collection.
- Communication missions - These missions are usually done to provide real-time communication connection for Earth-Earth, Earth-Space or even Space-Space purposes and IoT (Internet of Things) style missions[1].
- Technological demonstrators - These missions purpose is to demonstrate a newly developed concepts or systems.
- Scientific experiments - These missions are focused on scientific experiments in space enviroment, eg. biology research, material testing, optical communication research, etc.
- Satellite constellations / swarms - These missions main focus is to provide a constellation of satellites for capturing a data or providing a service that would otherwise not be economically viable[1].

One of the few companies focused on PocketQube development is Alba Orbital. This company has developed a flight proven 3p PocketQube platform called Unicorn-2 (in Figure 1.1) and it was developed to match its performance and offer an alternative to 3U CubeSat (10×10×30cm), lowering the costs for similar missions and usages by over 50%.[1] It is the first PocketQube by the company that has been successfully deployed to the orbit.[11] It was placed on Low-Earth Orbit (LEO) at 403×348km[1], 97 deg., and lasted there for 123 days. This has moved the system to TRL-9[2] and the company now offers that system in a price range from 69.999€ to 249.000€, depending on number of ordered satellites and their configuration.[11]

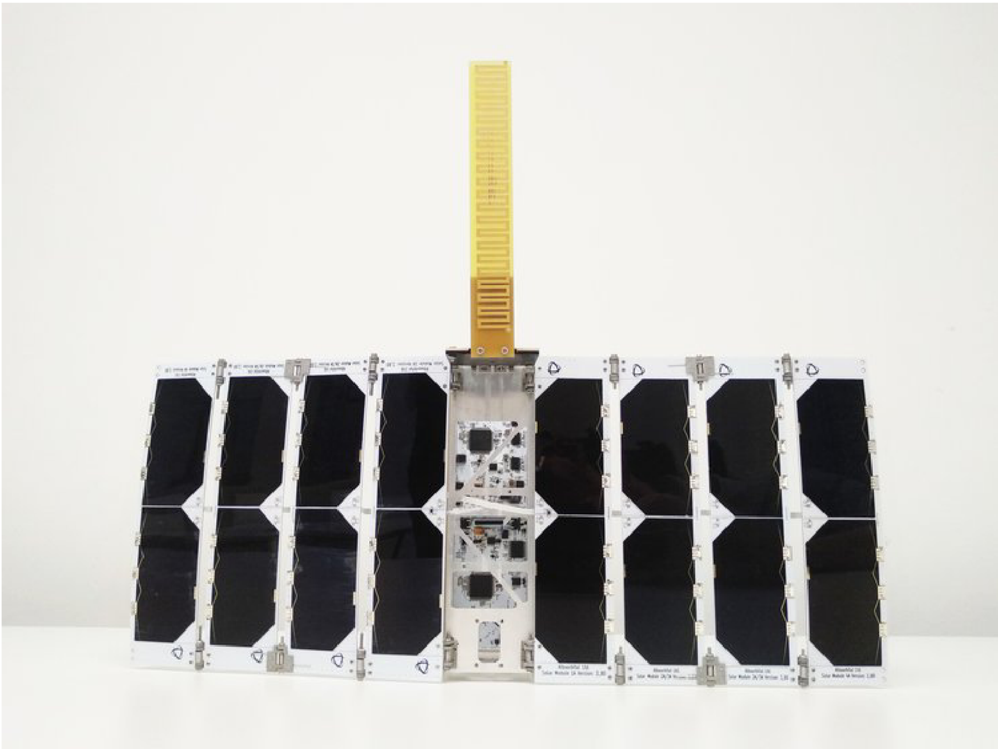


Fig. 1.1: Unicorn-2 PocketQube by Alba Orbital[1].

1.2 Educational application

In the classes, students will be encouraged to contribute by designing their own modules compatible with the proposed system bus pin header and satellite architecture. Through this approach, students will gain invaluable practical experience in

[1] Perigee×Apogee

[2] Technology Readiness Level, 9 is the highest

firmware, electrical and printed circuit board (PCB) design, problem troubleshooting, functional analysis, and system engineering[12].

1.3 Satellite subsystems

Satellites are usually composed of several subsystems that, when integrated into one unit, ensure satellite operations. To ensure proper functionality in the harsh space conditions, some systems are usually redundant. This is done in case the primary one starts malfunctioning, we don't lose the entire mission.

1.3.1 On-board Computer (OBC)

On-board Computer is the satellite's central processing unit. The OBC takes care in executing the received commands, telemetry data handling, fault detection and recovery, time synchronization and operating the mission autonomously. The computer communicates with other modules by utilizing available interfaces (CAN, I2C, RS-485, SPI, etc.). As the main processing unit are usually used microcontrollers like AVR, MSP430 or ARM Cortex-M series. These units are usually programmed in C, eventually in ASM. This is to ensure the atomicity of the code and to have everything under control. The code is mostly designed with a static allocation to prevent run-time errors, meaning that the memory allocation size and addresses are determined and set at compile time, giving the programmer more control over the system.[9]

The satellite control can be centralised or decentralised. Centralised control means, that everything is controlled only by the OBC and it is a more common and easier approach to the problem. Decentralised means, that every subsystem works and communicates with the communication subsystem independently in real-time and in case of OBC failure the mission is not destined to its end. Implementation in this case is more complex and OBC primary task in this configuration is collecting and processing telemetry data.[9]

1.3.2 Electrical Power System (EPS)

The EPS is the system that takes care of the generation, storage, control and distribution of electrical power to the other subsystems of the satellite. The energy is generated by the photovoltaic cells on the external structure of the satellite, which are usually selected by the highest efficiency (usually around 30%). To maximize the power input of the photovoltaic cells, an MPPT controller is used to maximize their efficiency. The energy gained is then stored in a battery cell inside the satellite

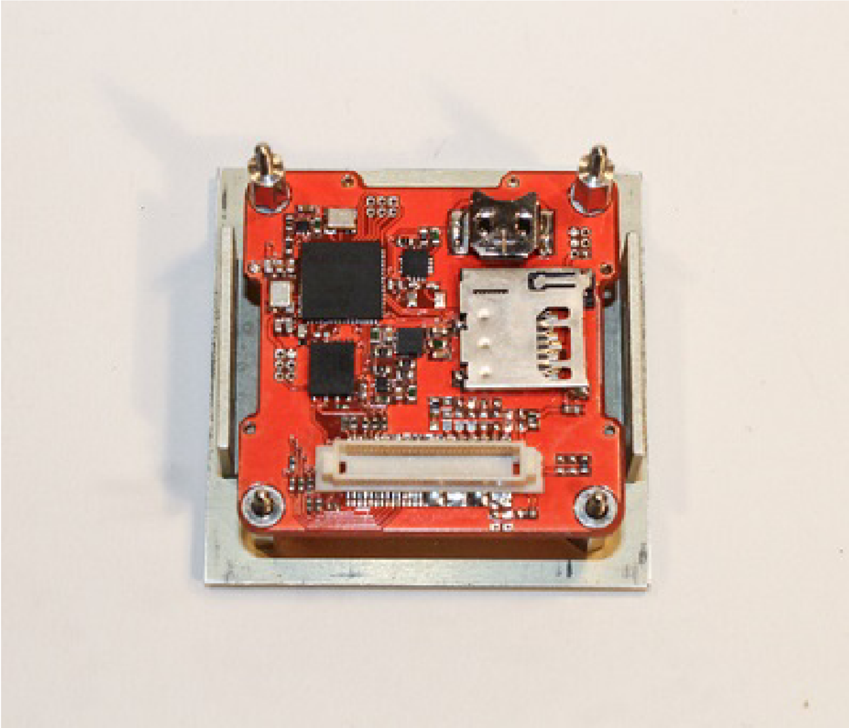


Fig. 1.2: PocketQube OBC by Alba Orbital.[1]

before it needs to be redistributed to the subsystems. During distribution, the energy is regulated to the correct voltage levels required by each subsystem to function properly, usually by using LDO regulators.

The EPS often includes a separate microprocessor that monitors the battery charge, its lifetime, energy input from the photovoltaic cells and the satellite's consumption. This is either handled autonomously or reported to the main computer unit, which then adjusts the satellite's operations to avoid a drop in energy to critical levels. The so-called OAP (orbit average power) is used to calculate the average available power. To determine its value, is used a "rule of thumb", which states, that the OAP is 60% of the power of one solar panel per 4 installed panels, since the power input must always be higher than the power output. It is also important to be able to separately turn off and on each connected subsystem, as most of them do not require to operate 100% of the time and hence do not need to be powered all the time, thus saving energy.[9]

1.3.3 Communication System (COM)

Another subsystem of the satellite is the communication module that provides the connection between the satellite and the Earth. This is an absolutely critical part of the system, as loss of communication means the end of the mission. For communication, a digital link with frequency modulation at the physical layer (e.g. FSK or LoRa for communication below the noise level) is usually used. At the link layer, the AX.25 protocol is often used for packet transmission. Communication often takes place in amateur or commercial radio bands, but this is slowly becoming a problem due to the increasing commercialisation of the sector and their over saturation.

The bands used for communication are:

- VHF - 145MHz (amateur radio band),
- UHF - 435MHz (amateur radio band),
- L-band - 1.3GHz (amateur radio band, large volume of data),
- S-band - 2.2GHz (satellite links).

Due to the absence of a grounding plane, dipole antennas made of flexible material (e.g. nitinol or measuring tape) are often used, which are wrapped around the satellite and secured with a nylon string, before being inserted into the deployer, to secure them against early unwinding. The actual deployment of the antenna is then done by burning/melting the string with a heating element (resistor), which releases the antenna, enabling it to unfold to its original shape. Another option is to use a professional solution with its own launch mechanism. The transmission power of a PocketQube satellite is usually in the hundreds of milliwatts.[9][10] Recently, optical communication systems have also started to be used, especially in higher orbits (due to the reduced angle of aim of the optical beam).

1.3.4 Attitude Determination and Control System (ADCS)

The ADCS is used to detect and adjust the position of the satellite. This system receives data from various sensors such as a sun sensor, magnetometer or star tracker to determine the satellite's position and then adjusts it using three-axis, passive or rotational stabilisation. Passive stabilization is basically just a magnet permanently placed in one position that "locks" the satellite in one position relative to the earth's magnetic field. Another option is a so-called reaction wheel on each axis, which spins in the opposite direction of the satellite's rotation to stabilise it (the satellite loses torque, which then slows its rotation). However, this approach needs to be desaturated by gradually slowing down the rotation of the reaction wheel. This is usually done by using a so-called magnetorquer (implemented as a ferrite-core coil or solenoid), which interacts with the earth's magnetic field when current passes through it. For larger satellites, jets are used for stabilisation. Stabilization of a satellite is a fairly complex matter, and for small satellites only passive stabilization is often used, because of the many design limitations.[9]

1.3.5 Payload

The word “Payload“ originally came from the marine environment and its meaning is “revenue-producing cargo“. Nowadays payload in space terms is the main mission element dedicated to producing mission data which are then transmitted back to Earth.[13] Payloads can be categorized according to their intended use and specific requirements of the mission. Some of these are[14]:

- Scientific payloads,
- Communication payloads,
- Navigation payloads,
- Observation payloads,
- Remote sensing payloads[15],
- Technological demonstrators,
- Development of new space technologies.

Example of a technological demonstrator payload is the NOOR 1A/1B mission (also known as Unicorn 2B/C), which purpose was to validate the structure, thermal, OBC, EPS, encrypted communication and operating system of the Unicorn-2 PocketQube. The aim was also to demonstrate the integration of the satellite with ground station, allowing 3rd party satellites to request data transfer. The spacecraft successfully deployed its solar panels and demonstrated its communication capabilities.[16][11]

1.4 Ground Station

For communication with the satellite we need a ground station. This is a tramsmitter/receiver device located on the Earth, which communicates with the satellite on the same frequency. For good signal reception, antennas with circular polarisation are often used so that communication can take place even if the satellite is not rotated in an ideal position. Another option is the use of directional antennas on a so-called rotator, which mechanically rotates and moves the antenna to point to the satellite by using tracking software. However, in this solution we need to keep the moving parts maintained. The system often includes an LNA (low noise amplifier) module, especially when long cables are used between the antenna and the radio to improve SNR, and a SAW filter to suppress unwanted frequencies.[9]

Another option is to use SatNOGS. This is an open-source network of ground stations communicating in the VHF and UHF bands. The goal of the project is to create and provide a worldwide, interconnected network of ground stations for LEO satellites. For full members of the network with their own station, the ability to schedule records for satellite passing over any station connected to the network is available.[9][17]

1.5 CubeSat Space Protocol

The CubeSat Space Protocol (CSP) is a small communication protocol written in C language following the TCP/IP model and implements transport and routing protocols and several MAC-layer interfaces. It is designed for communication in small networks of distributed embedded systems, such as CubeSats. The goal is to provide the functionality of a TCP/IP stack without the added IP header which is too large. The protocol provides its capabilities to all connected devices without the need for a control master node.[6] The protocol is more complex to implement but has the advantage of allowing limited system functionality even in the event of a main computer failure.[9]

CSP is not commonly used on PocketQube satellites, but there is nothing to prevent its use due to its simplicity if a suitable system architecture is used.

2 Educational PocketQube design

The design of satellite's architecture aligns with PocketQube standard[8] and is focused on development of the On-Board Computer, Communication subsystem, Electrical Power System and the satellite structure. Great emphasis has been placed on modularity and usage of COTS parts for easy interchangeability, as the resulting device will be used mainly in STEM laboratory classes, where students will try to test and develop multiple different combinations. The developed PCB designs contain also blank templates for future internal modules development, eg. payload or ADCS.

2.1 Structure design

To reduce cost of the satellite and to avoid any potential errors with deployment of the satellite, the PocketQube structure dimensions are strictly defined by a standard[8]. The structure is usually made out of aerospace grade aluminium with plated finish[1], which is lightweight and durable enough, to withstand harsh flight conditions. A 2p PocketQube skeleton wall structure with Flight Heritage[1] can be bought for 2.999£(Figure 2.1). Part of the construction is also a pair of "kill-switches" (for double redundancy)[1], which serve a purpose of keeping the satellite electronics turned OFF until the deployment.

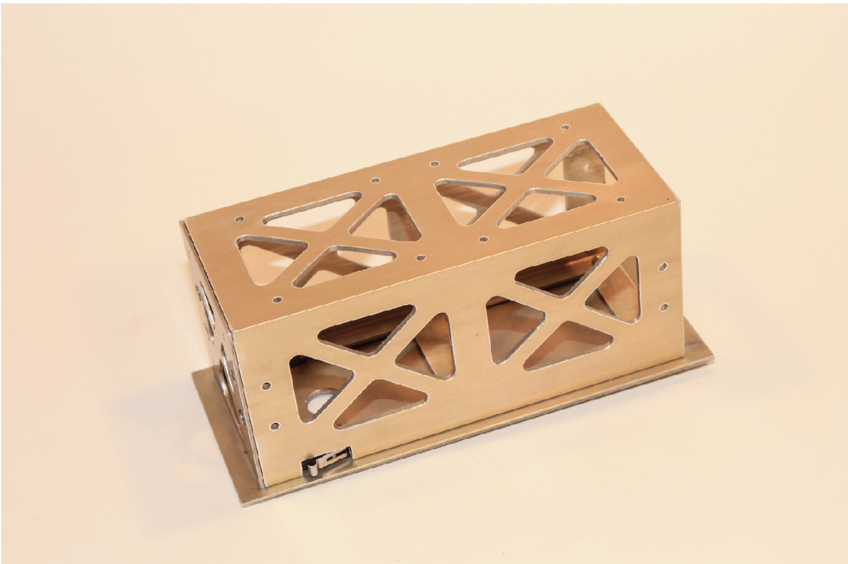


Fig. 2.1: Photo of the Alba Orbital 2p PocketQube aluminium structure.[1]

[1]Flight Heritage means, that the concept has already been tested in a real-life conditions

The design of external structure can be separated in two parts - the upper structure, containing all the internal components inside and a so called “sliding backplate“ which forms the bottom side of the satellite and servers for securing the satellite into the deployer (Figure 2.2).

Fig. 2.2: Photo of the Alba Orbital deployer.[2]

The educational PocketQube entire external structure is designed as six PCB plates with a thickness of 1.6mm, which, when assembled, form a 2p PocketQube satellite (58×128×1.6mm[8]). There are 4 (or 2) solar cells placed on each side, providing the satellite with energy supply and on the top panel, there is a hole for USB-C connector (external power supply) and a kill-switch to turn the satellite ON/OFF.
This approach of using only PCBs was chosen based on its relatively easy way of design, durability and because it provides much more space and weight budget for internal components in comparison with 3D printed structure.

The internal modules are in a strict order stacked on top of each other and secured in place by distance posts (as is described in chapter 3.4) and all are electrically interconnected by a system bus. Their dimensions must be precisely designed to fit in the structure without any problem. For these purposes were made the blank PCB templates (Figure 2.4), to ensure that all of the modules comply with the strict requirements for the dimensions. In this way, all of the internal modules have the exact same dimensions and the system bus and mounting holes are placed in the same spot for smooth assembly. The 3D model of the assembled structure is depicted in Figure 2.3.

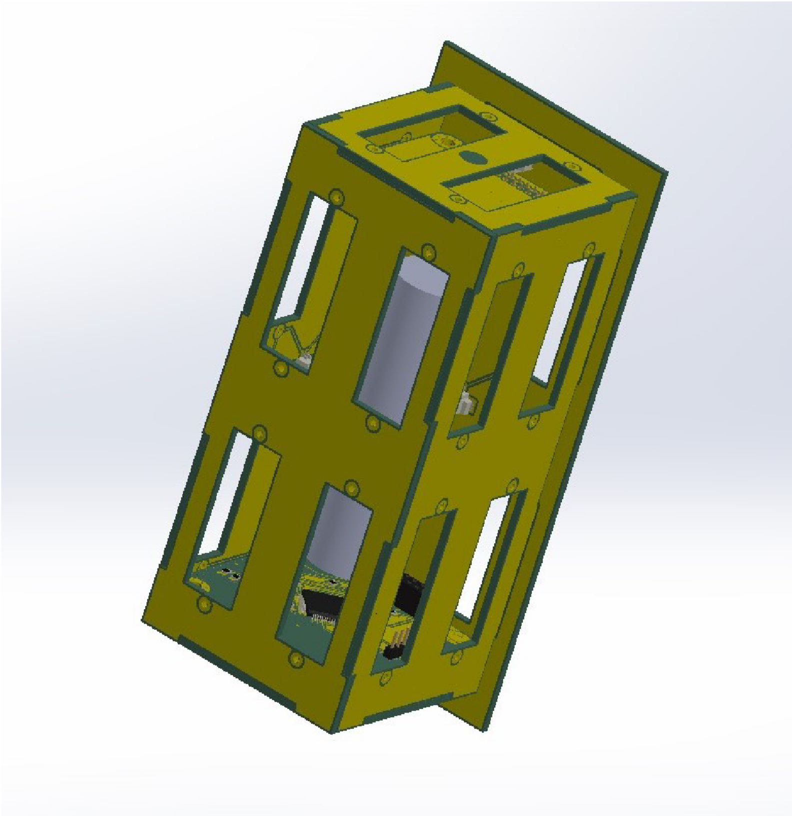


Fig. 2.3: 3D CAD model of the unpopulated structure.

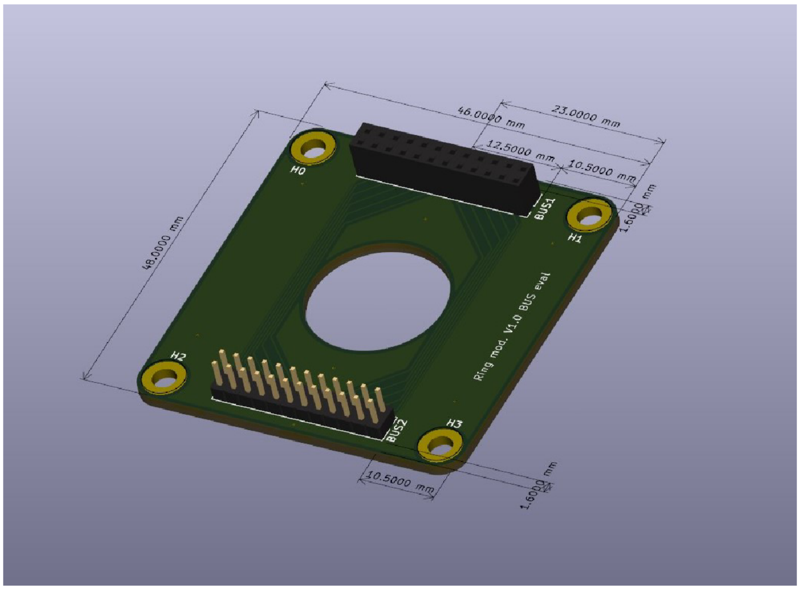


Fig. 2.4: Internal PCB blank template with output for pin testing.

2.2 OBC design

The On-Board Computer (OBC) is designed as a separate module, controlling the functionality of all of the satellites subsystems. It is driven by the STM32F446RET microcontroller (MCU) manufactured by STMicroelectronics, with an external HSE 8MHz crystal and optional LSE 32kHz crystal. The MCU is designed to be directly soldered on the PCB board, because any considered plug-in options (eg. Black-pill development board) are too big in size to fit into the PocketQube standardized unit.

The properties of the used MCU are [18]:

- 32-bit ARM Cortex-M4 core with FPU and MPU
- Power-supply 1.7V to 3.6V
- Core frequency up to 180MHz
- 512 Kbytes of internal Flash memory
- 128 Kbytes of internal SRAM
- 3×12-bit 24 channel ADCs (2.4 MSPS)
- 2×12-bit D/A converters
- General purpose DMA
- 17 timers (2×Watchdog, 1×SysTick, 12×16-bit, 2×32-bit)
- SWD and JTAG debug interfaces
- GPIOs with external interrupts
- 4×SPI (45 MBits/s, three with I2S audio class)
- 2×CAN
- 4×I2C interfaces
- Up to 4×USARTs and 2×UARTs (11.25 Mbit/s)
- CRC calculation unit and RTC
- 96-bit chip unique ID
- 1×USB interface

The module further includes an external 256kB EEPROM memory on I2C bus, which can be easily swapped for a different size. PCB 3D model of the designed OBC is depicted on image 2.5.

The OBC is connected to other subsystems using a pair of system bus connectors which is standardized across all modules. Through the system bus, the OBC is capable of acquiring data from the EPS or processing and transmitting data through the COM module. All other unused pins on the MCU are left unconnected because of the space restrictions.

The OBC module mainly utilizes the SPI and I^2C interfaces, with an extra

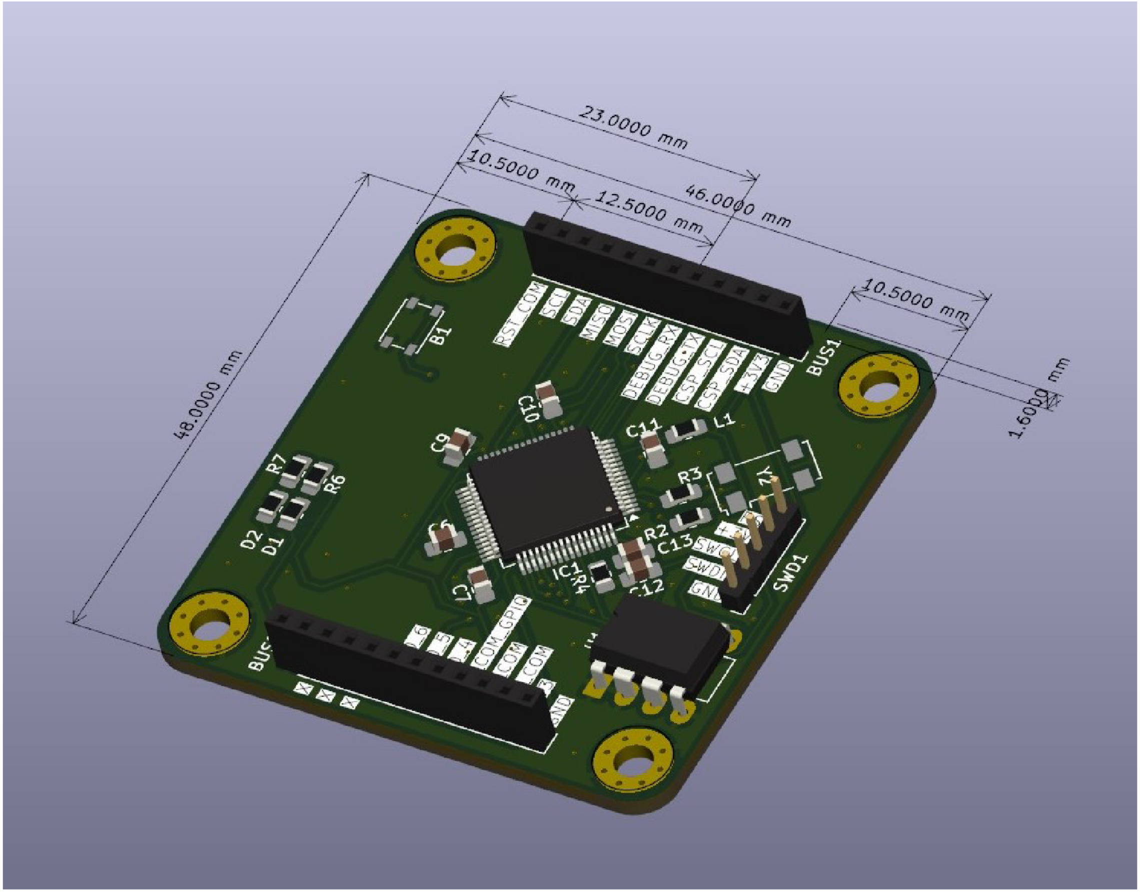


Fig. 2.5: 3D model of OBC design.

I^2C_CSP interface, which is isolated from the first I^2C interface and is destined to be used in the future, specifically designated for communication over the CubeSat Space Protocol (CSP)[6] and one USART interface for debugging. For programming the device with compiled firmware binary, an SWD interface will be used.

To ensure a proper functionality of the satellite, the OBC module also serves as the control unit for the EPS, monitoring the available battery capacity to determine whether to continue operations according to current settings or to suspend the satellite's activities and put itself into a sleep mode. Block diagram of the OBC is depicted in Figure 2.6.

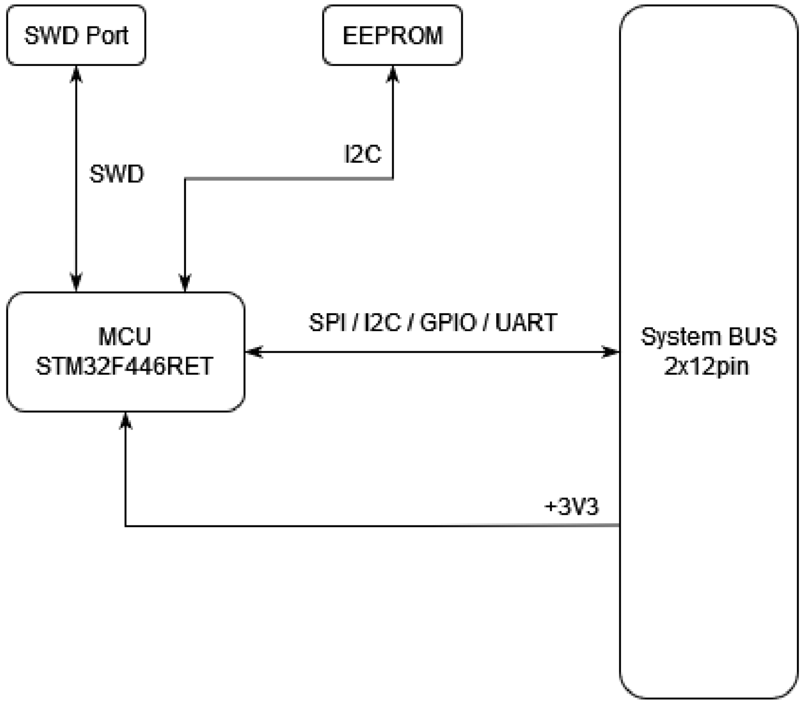


Fig. 2.6: Block diagram of the OBC design.

2.3 EPS design

Another module of the PocketQube design is an Electrical Power System. This module provides the satellite with an energy to operate. EPS module consists of the Li-Ion 18650 battery with a capacity of 3350mAh for energy storage, charging circuit with integrated overcharge protection, solar cells connected from the external structure, kill-switch, 3.3V LDO stabiliser, current measuring circuits and ADC expanders.

The energy from 20 mono-crystal solar cells (45×15×2.1mm) mounted on the external structure, have maximum peak power 123mW per cell and in normal condition a cell efficiency of 25% [19]. These solar cells are connected in 10p2s configuration, to ensure that the energy keeps flowing into the system, even if not all of the solar cells are illuminated. For short circuit protection, on each parallel set of solar panels, Schottky diodes are placed directly on the internal side of the external structure. Solar cells input is monitored from OBC via ADC lines connected to ADC I2C expander.

Their energy input is stored in one Li-Ion 18650 battery with a capacity of 3350mAh, which is connected to the system by two PCBs with connector springs, one on each side of the battery. To keep the battery firmly in place, two another PCBs with a hole in the middle are placed around the battery. Design of these PCBs

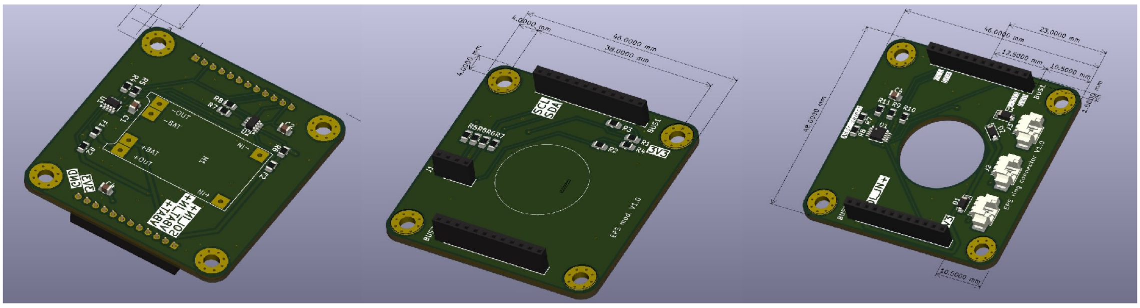


Fig. 2.7: 3D model of EPS PCBs

is depicted in Figure 2.7. The cathode side PCB also contains an I2C connector for payload module.

The battery itself is protected against a short circuit and overcharge with a TP4056 Li-Ion COTS module that has an USB-C connector for external charging and power delivery.

System input and output current is measured by two INA219 current measuring modules, each placed on the battery input and output. These two modules are connected via the I2C interface to the OBC which then monitors and regulates the power consumption itself. To turn the satellite ON and OFF, a kill switch is connected to the main power rail with a 2N7002 N-channel MOSFET in parallel, controlling the state of IRLML6402 P-channel MOSFET, which stays closed if the switch is in the OFF position. In the current setting, main power rail provides only 3.3V of voltage power. The block diagram of the EPS can be seen in Figure 2.8.

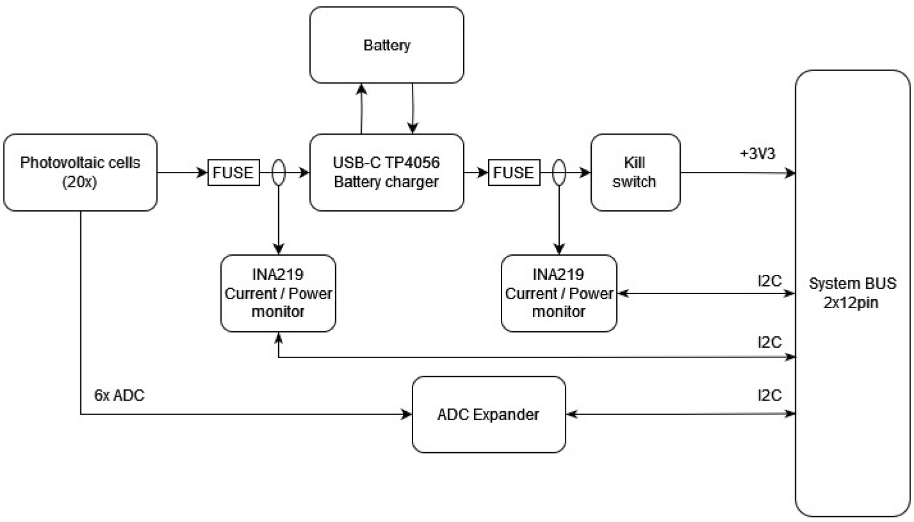


Fig. 2.8: Block diagram of the EPS design.

2.4 COM design

As the heart of the Communication (COM) subsystem was chosen the RFM98W RF Transceiver module from HopeRF[4] (Figure 2.11) which is a cheaper version[20] of the SX1278[21] radio IC from Semtech. These modules offer a wide range of modulations, such as LoRa, FSK and OOK. The maximum output power of the radio is +20dBm (100mW) and operates at a selectable frequencies in range from 410 to 525MHz for used module, covering the lower UHF band. This module shares the same footprint and pinout with other versions from the manufacturer (RFM95, RFM96 and RFM97), which differ mainly in supported modulations and frequencies. The internal functionality between all versions is similar, so there should be only a little change in the firmware registers needed in case of change for a different type. This means, that the radio module can be easily changed for another one supporting different options. This module is connected via a system bus to the rest of the satellite, utilizing the SPI interface for communication with the OBC. The antenna for the RF communication is designed to be connected via an SMA connector. 3D model of the COM subsystem module is depicted on image 2.9.

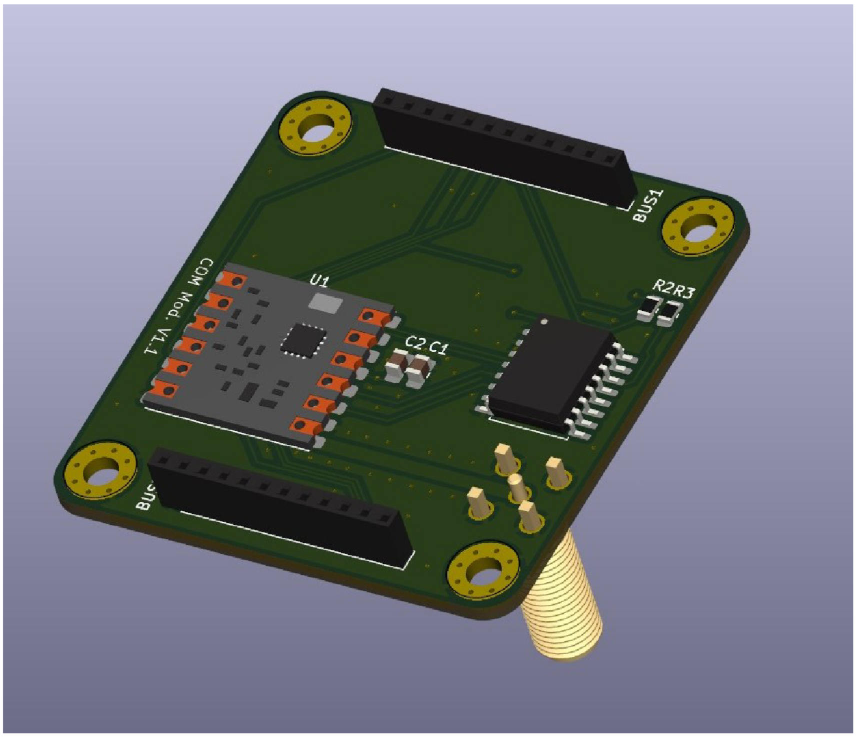


Fig. 2.9: 3D model of COM subsystem PCB.

The key features of RFM98W module:

- Modulations: FSK/(G)FSK/MSK/(G)MSK/OOK/LoRa modulations
- Power supply: 1.8-3.7V
- Interface: SPI
- Bandwidth range: 7.8kHz - 500kHz
- Max. data rate: 300Kbps
- RSSI dynamic range: 127dB
- High sensitivity: -148dBm
- Max. output power: +20dBm (100mW)
- +14 dBm high efficiency power amplifier
- Built-in temperature sensor and low battery indicator
- Module size: 16×16×1.8mm
- Automatic RF sense
- Packets up to 256bytes with CRC

Besides the RFM98W module, the COM subsystem also has a PCF8574 GPIO expander [22] providing easy access from the OBC to unused RFM98W pins via I2C bus. In Figure 2.10 is presented a block diagram of the COM subsystem module.

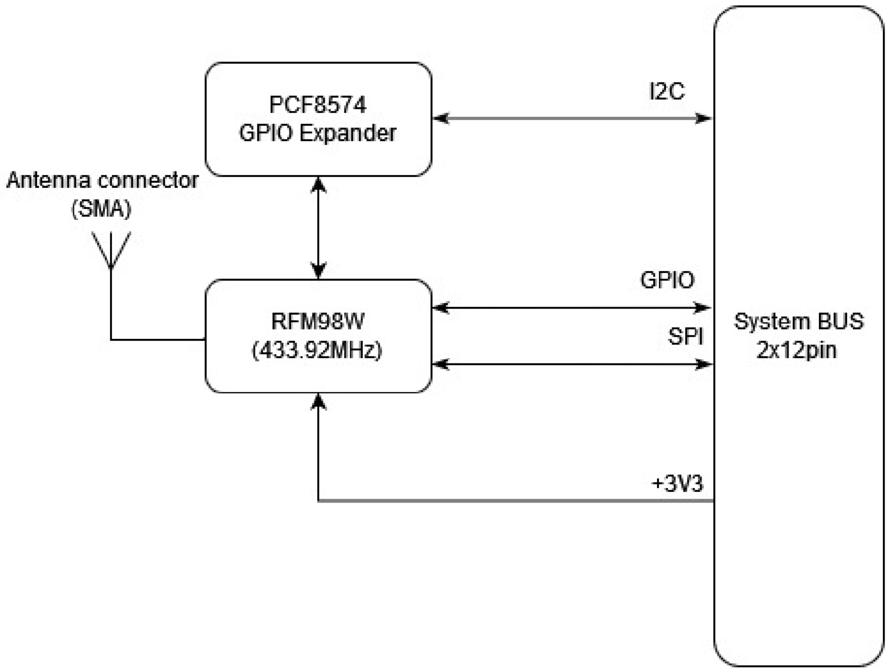


Fig. 2.10: Block diagram of the COM design.

2.5 Ground station design

The Ground station design is designed similarly to the COM subsystem, utilizing the RFM98W module on 433MHz frequency and an SMA connector for an 433MHz tuned COTS antenna. The PCB itself is designed as a shield with Arduino UNO V3 spacing, which is also compatible with the most of the ST Nucleo development boards[23]. Since there are plenty of unused GPIO pins on the Nucleo, there is no need for a PCF8574 GPIO expander as those pins can be connected to the Nucleo development board directly. The image of RFM98W can be seen in Figure 2.11 and the 3D PCB design for the Ground station can be seen in Figure 2.12.

Fig. 2.11: RFM98W module by HopeRF.[3]

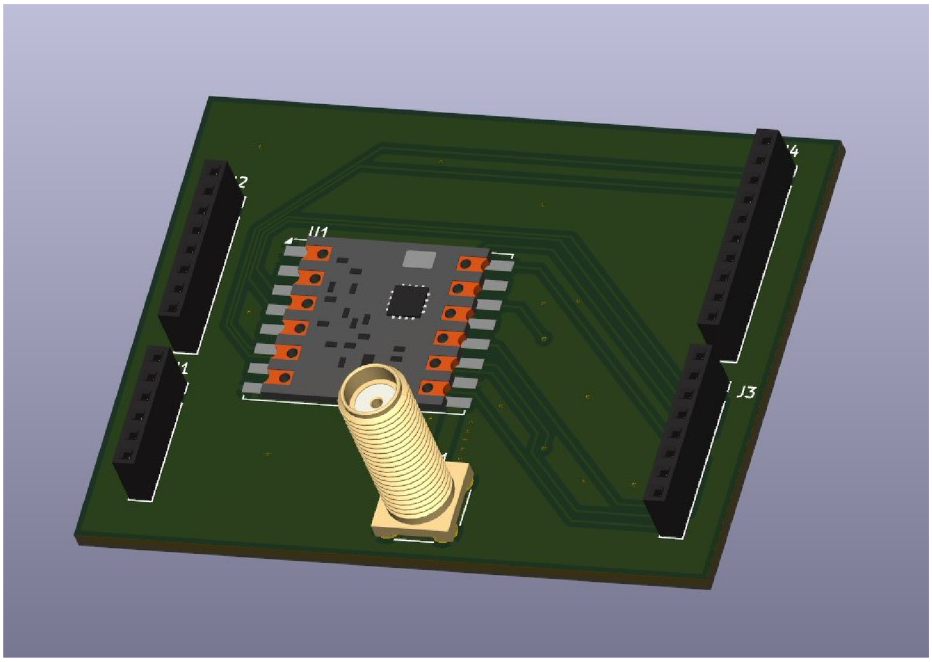


Fig. 2.12: Ground station 3D model.

3 Hardware Implementation

The PCB design, circuit diagrams and 3D models of both internal and external modules was created in KiCad version 8.0[24]. 3D model of the entire assembly was created in SolidWorks 3D CAD[25], which license is provided by the university.

Each part is designed as a 2-layered PCB, even though the design could've been made by using 4-layers. This decision has been made based on the final cost considerations and design simplicity. Every internal module has two 1×12 2mm pinheaders (system bus), that align with each other and make completing the final assembly and future expandability easier. Each module have a mounting hole in every corner for distance pillars, to hold everything tightly together. To isolate possibly exposed parts that might cause short circuits, common electrical tape was used to cover these spots, just to be sure.

Another used parts that are in common on all modules, are resistors, diodes and capacitors, that were used in SMD with 0805 footprint. All parts were soldered manually.

Some of the parts needed to be modified after the production because of the faults in the circuit design. These faults, their consequences and how they were fixed are described in sections below. Final design was manufacured by a company JLCPCB[26]. Due to delays in manufacturing process, only one prototype has been made and so the error fixes will be incorporated in the future designs.

3.1 OBC

The main component on the OBC is a STM32F446RET MCU[18], controlling all of the satellite operations. This MCU has been directly soldered to the top layer of the PCB alongside a DIP-8 socket for easy interchangeability of the currently used AT26C256 256Kb EEPROM[27]. For ability to manually reset the device, a button has been added to the top layer, which, when pressed, closes the electrical circuit and restarts the MCU. On the bottom layer of the PCB an 8MHz HSE crystal[28] has been placed. The 32KHz LSE crystal has been left unsoldered since it has no real usage in current configuration. For visual debugging and status signalization, two SMD LEDs (red and green) have been soldered to the top side. For debugging and programming, 5-pin header has been populated on the board from the top side. The used populating plan is in Appendix T and PCB design in Appendix S.

On Figure 3.1 can be seen the finished unit from both sides.

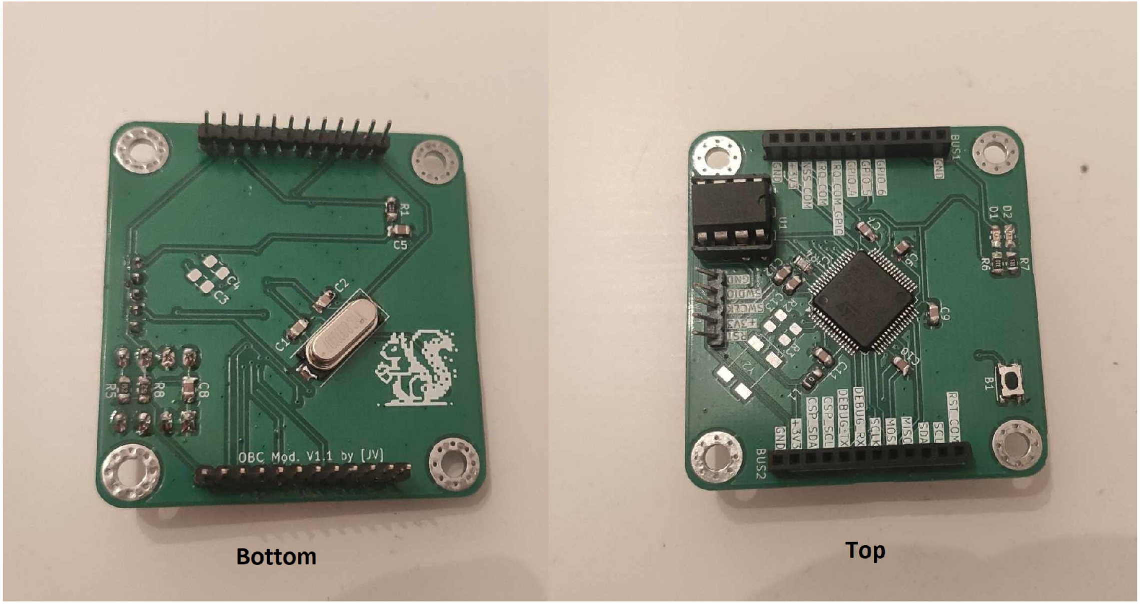


Fig. 3.1: Both sides of the completed OBC module.

3.2 EPS

EPS module has been separated into 3 parts - Cathode side, Anode side and a “Ring Connector“. The first mistake was found when completing the Cathode side - a footprint for battery spring has been incorrectly placed on the bottom layer of the PCB instead of top. To fix this mistake, a hole has been drilled through the PCB by which was then threaded a wire soldered on the bottom to the footprint and on top to the spring. Copper around the drilled hole has been isolated from the rest of the PCB to avoid any short circuits. This fix works as intended and unexpectedly enabled an access to the battery cathode for measuring purposes - through a hole designed in a structure, originally intended as an access port for the antenna.
To the Cathode side has also been soldered and glued (for easy interchangeability) a MPU9250/6500 - Gyroscope, Magnetometer and Accelerometer COTS module[29] acting as a satellite payload. On the Figure 3.2 can be seen the finished product from both sides.

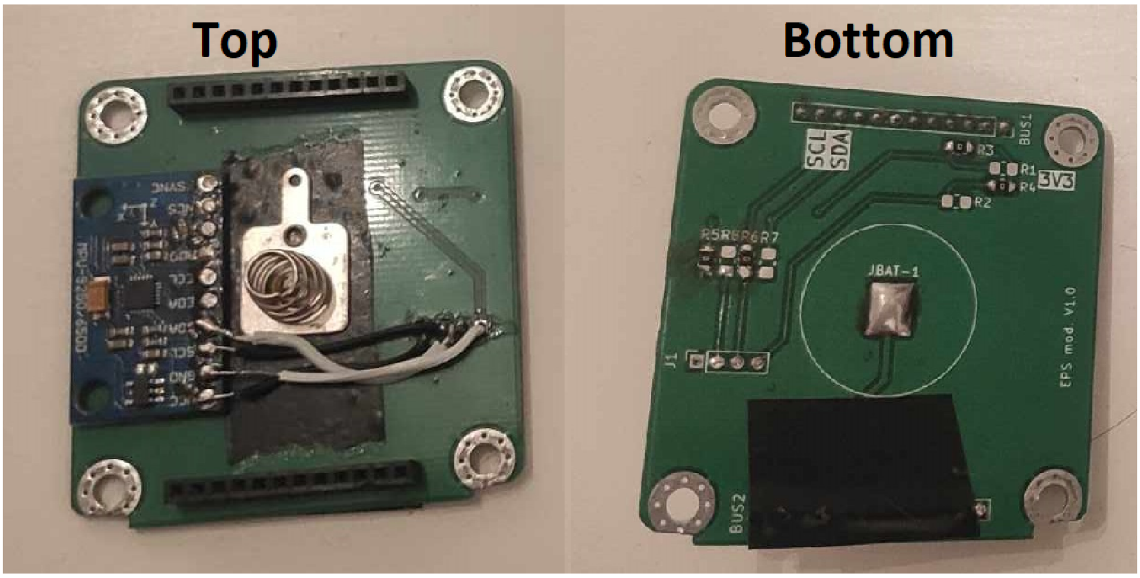


Fig. 3.2: EPS Cathode part from both sides.

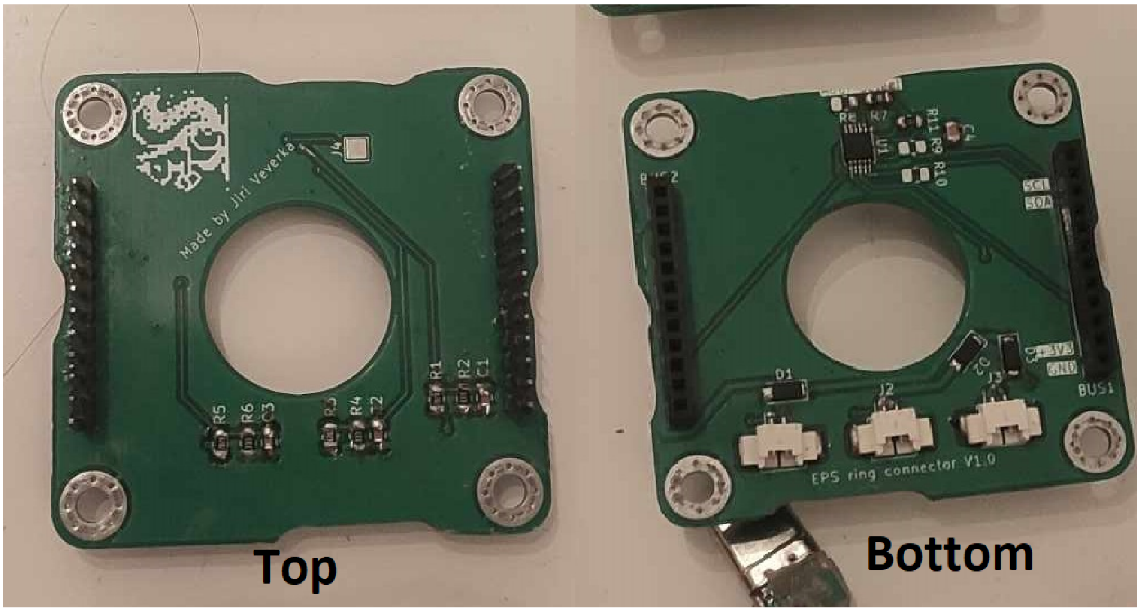


Fig. 3.3: EPS Ring connector from both sides.

For connecting the solar cells to the circuit, 4 molex 2-pin female connectors have been placed on the top layer of the "Ring Connector" with Schottky diodes and ADS1115 ADC expander module[30] for measuring the solar cells input. Thanks to the hole in the middle for the battery, this PCB also acts as a stabilizer, keeping battery tightly connected in place. Total number of 2 "Ring Connectors" have been finished and used in the final assembly. In Figure 3.3 are the final results from both sides.

The last part of the EPS is the Anode side. This PCB contains all of the EPS power logic components. For battery charge protection, TP4056 USB-C COTS module[31] on pins has been soldered to the bottom, together with two INA219 IC[32] for battery input and output measurements. On the top side is placed a battery spring, molex 2-pin female connector for kill-switch, P-Channel MOSFET IRLML6402[33], N-Channel MOSFET 2N7002[34] and a 3.3V LDO MCP1703A-33[35]. During the testing phase was found out that the design is missing a MPPT module to get as much energy as possible from the solar cells and this shall be addressed in a future development. For energy storage is used a INR18650 Li-Ion battery with capacity of 3350mAh.[36]. Finished Anode side can be seen in Figure 3.4. The EPS PCB design and populating plans are in Appendices K and L respectively.

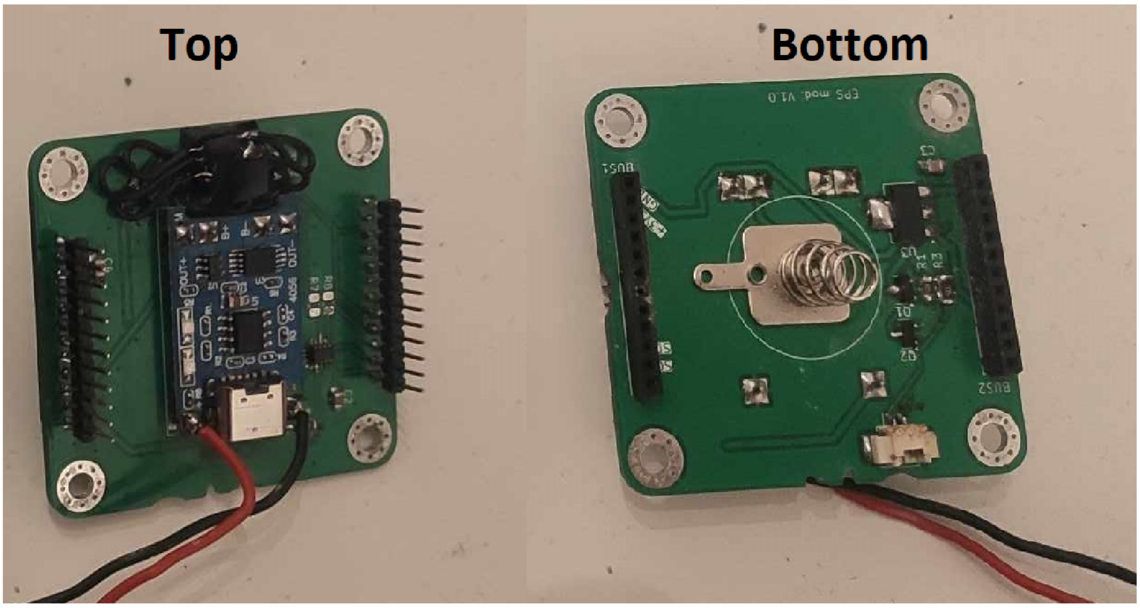


Fig. 3.4: EPS Anode part both sides.

3.3 COM and Ground Station

The Ground Station and COM PCBs have the most parts in common. Both modules have a RFM98W radio module on top layer and an SMA antenna connector. This connector is placed on top layer of the Ground station PCB and on bottom layer of the COM PCB. Both are connected to their respective radio modules by a 50Ω feeding line, isolated from both sides by vias. COM module has also an extra PCF8574 GPIO expander soldered on its top layer. Both modules posses a COTS 433MHz tuned antenna[37][38], attached to their respective SMA connectors, with the addition of 50Ω wire connected between the COM SMA connector and the structure antenna port, which were added based on results of the communication testing phase. An error has been made in the design phase of the Ground station, where the 2mm pin headers were used instead of the 2.54mm size. This problem has been resolved by soldering wires between the 2mm and 2.54 pin headers and in future version of the board it will be fixed. Figure 3.5 shows COM module implementation from both sides and final Ground station assembly is shown in Figure 3.6. Ground station PCB design is in Appendix O and the COM one in Appendix G. Placement schematic for Ground station is in Appendix P and for COM in Appendix H.

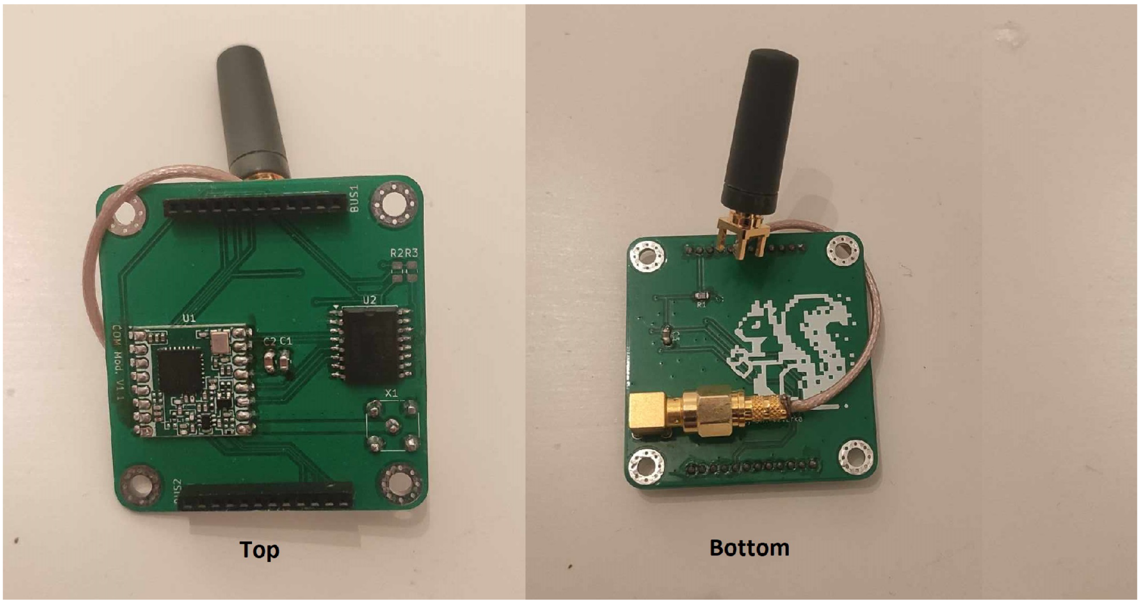


Fig. 3.5: Both sides of the completed COM module.

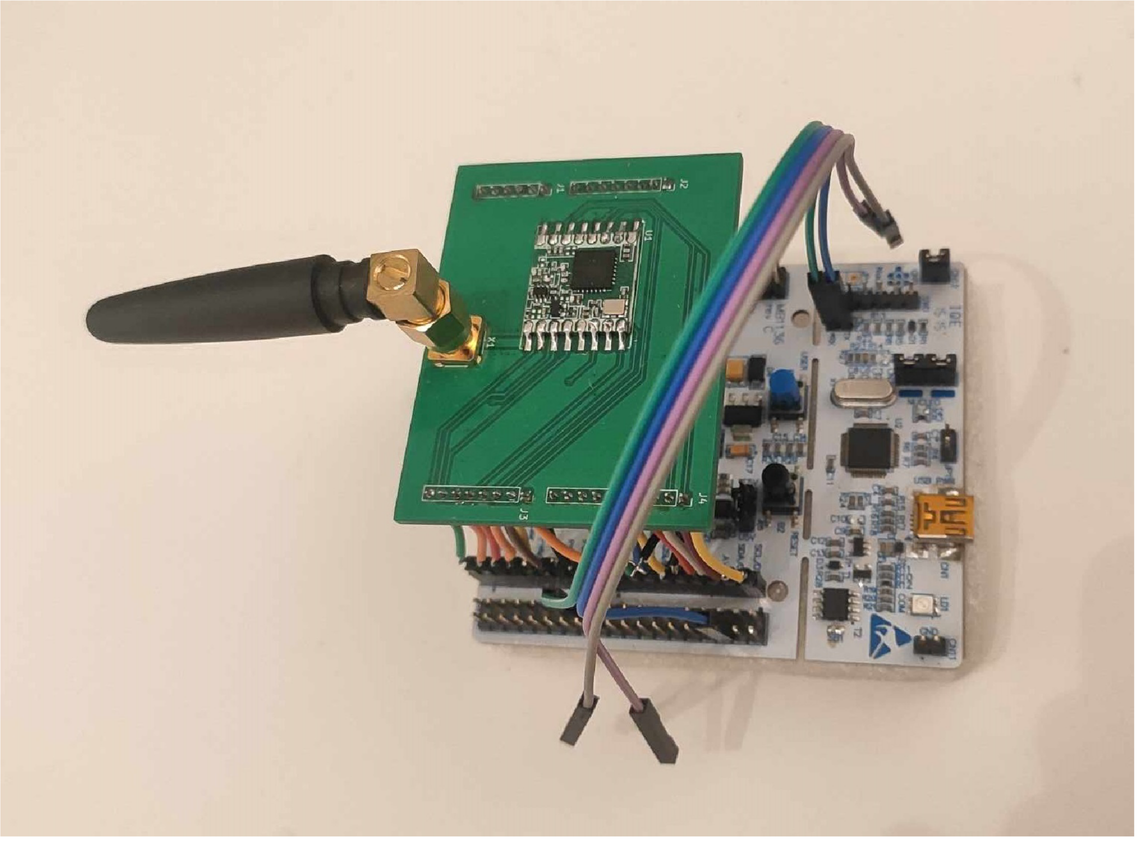


Fig. 3.6: Ground Station assembly.

3.4 Structure

The satellite's structure is entirely made out of PCBs and can be categorized in two parts - an internal structure made out of the modules described above and external structure. The external structure is assembled by soldering the slider and side panels together, forming a "box" for the internal components. The bottom panels are placed on top and down of the assembled internal structure, which is then inserted altogether in the already pre-prepared "box". The top panel is then placed on the opening, effectively closing everything inside and is being held in place by adjusting a tension from screws on the bottom panels. On every panel (except bottoms) have been soldered 4 solar cells SM141K0LV[19] (total 16) with SMD Schottky diodes, providing the system with energy (as described in section). The bottom panels have a circular opening in the middle for an antenna, which were modified a little bit to fit the SMA connector in. On the top panel is soldered a kill-switch to turn the system ON/OFF and an an USB-C female connector soldered inside a charging port. This USB-C connector will be also connected to the SWD debugging interface in the future, so the structure does not need to be disassembled with every firmware upgrade. The opened external structure can be seen in Figure 3.8 and every (unpopulated) panel in Figure 3.7.

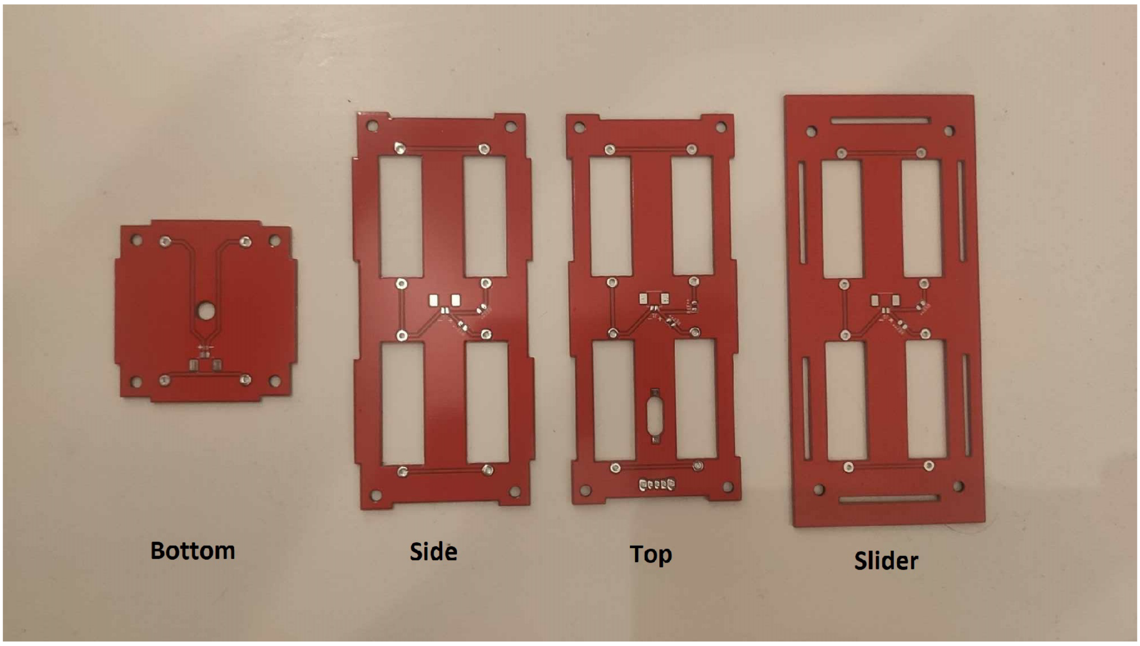


Fig. 3.7: Unpopulated external structure parts with descriptions.

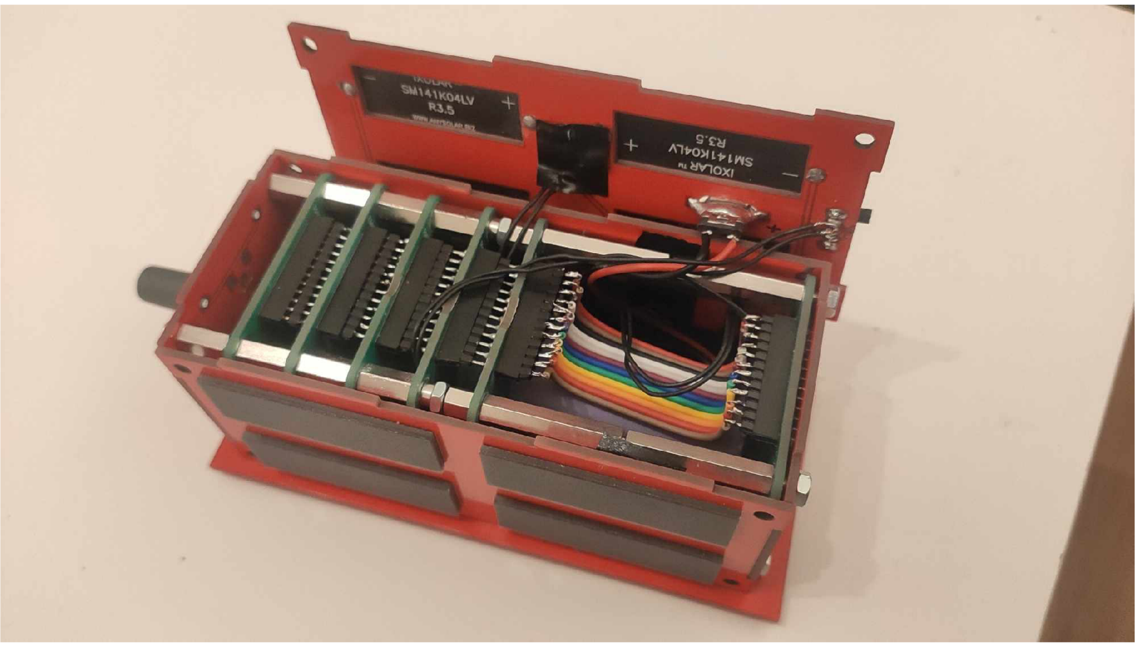


Fig. 3.8: Photo of the opened partially assembled structure.

The internal structure is being held together by distance posts and needs to be assembled in a strict order to work properly and fit inside the external structure without any problems.

The assembly order is:

1. Bottom panel
2. COM module
3. OBC module
4. EPS Anode module
5. EPS Ring connector
6. EPS Ring connector
7. EPS Cathode module
8. Bottom panel

and is being shown in Figure 3.9.

The finished assembly can be seen in Figure 3.10 and the satellite weighs 334g in total, which fits inside the PocketQube standards limits[8], even though it was not a requirement of the design.

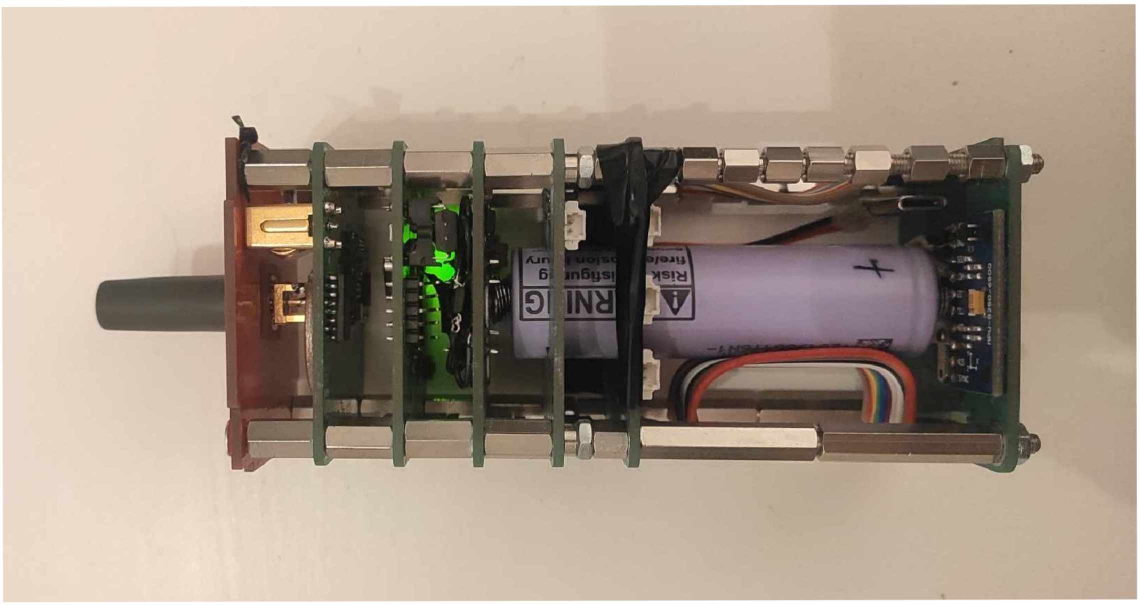


Fig. 3.9: Photo of the internal structure.

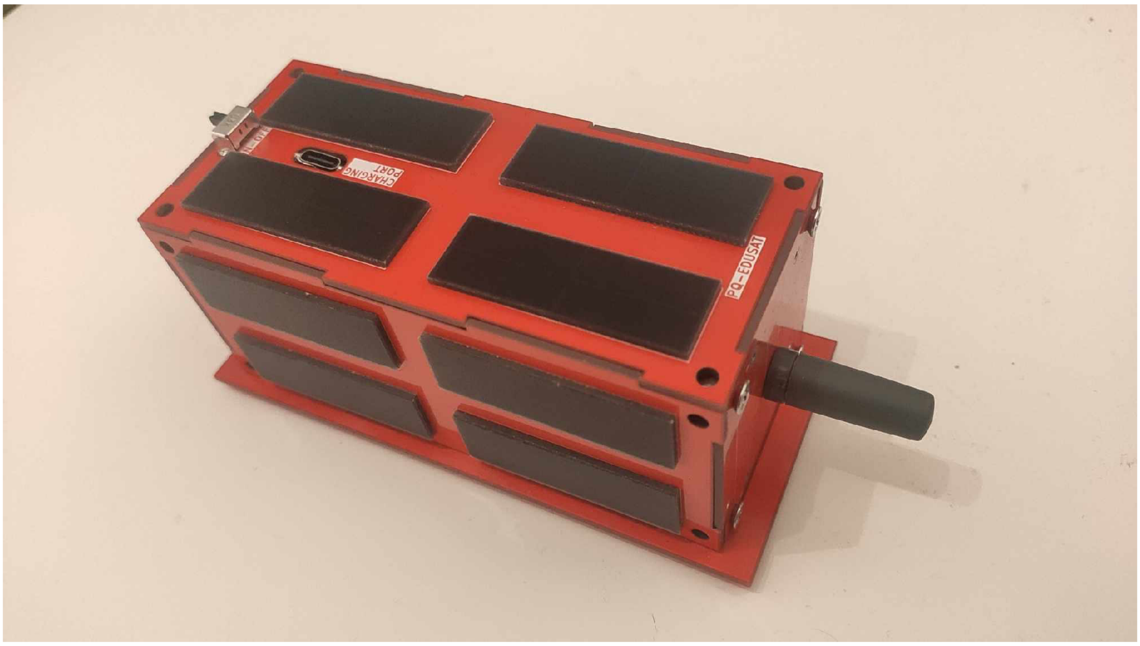


Fig. 3.10: Photo of the complete assembly.

4 Software Implementation

The work also included the development of the satellite and ground station firmware and a terminal application. These codes are used to demonstrate the functionality of the resulting satellite, its modules, data transmission and their human readable representation. The codes are easily modifiable and the tools used make them suitable for teaching in STEM laboratories.

4.1 Firmware

The firmware of the Satellite and Ground station was written in C programming language in the STM32CubeIDE development environment. The HAL (Hardware Abstraction Layer) library, which is already part of the IDE, was used for the implementation. Since the satellite and ground station operate on the same type of MCU (STM32F446RET), they share the same core code and it was not necessary to make too many changes in their codes, so most of the functionality could be verified on the Ground station even before the satellite itself was finished.
Both devices communicate using the RFM98W module, which is connected to the MCU using an SPI bus, set in full-duplex mode with 8-bit data size. The control of the NSS pin of the module is controlled from the program.

For the implementation, a modification of the RadioHead[5] library was used, which was customized by adding the RFM98W support by the GitHub user x893 in the LMiC[39] project. On top of this the whole firmware code logic was then built. The processor clock was left at the default setting of 16MHz to maintain reduced power consumption, but can be set up to a maximum clock of 180MHz (or 45MHz for APB1).

Ideally, the code would include over-the-air upgrade functionality[40] to flash the code remotely to a second memory bank at a runtime, however the microcontroller used does not have dual-bank memory and therefore this implementation will be done in a future STEM lab assignment (Appendix E).
In the dual-bank scenario, the system would behave in such a way that while it is running an application from bank 1, a new application is being written to bank 2. After a reboot, the implemented bootloader would then decide which bank to run the application from. This decision is usually made based on an integrity check of the program in the memory and a timestamp that determines the version of the application.

The STM32 also provides the functionality of two software watchdogs that reset the entire system if not updated periodically - IWDG (Independent Watchdog) and WWDG (Window Watchdog). IWDG is connected to an independent 32kHz LSI oscillator and its advantage is that it remains active even if the main clock fails. The IWDG waits for a refresh, which must occur within the maximum length of the set window, and if the refresh does not occur, it restarts the entire system.[41]
Unlike IWDG, the WWDG is directly connected to the APB1 system clock bus and is not able to detect an error if the main clock fails. However, its use is suitable for applications where a task has to react in a well-defined window in which it must be refreshed.[42]
To keep the implementation and debugging simple, no watchdog has been implemented and its use is therefore left to the STEM labs.

The ST-Link programmer was used for uploading the code to the devices. In the case of the Ground station, this is already a built-in part of the Nucleo development board, but a separate USB programmer had to be used for the satellite.

4.1.1 Firmware variants

The firmware for the satellite and Ground station was written in two variants - one using the freeRTOS[43] to control the behaviour of the program in a pseudo-parallel preemptive way and the second one is a typical application where everything is handled from a super loop (or main loop), i.e. an infinite while loop containing all the code. The result is therefore a total of 4 firmware applications (freeRTOS for satellite and ground station, superloop for satellite and ground station) in which the basic functionality of the device (communication) was tested. Further development, such as implementation of commands, module control and communication debugging, took place only in the superloop version while the freeRTOS version is being reserved for educational purposes.

4.1.2 OBC Firmware

Device configuration

For communication with individual COTS modules, the I2C bus (I2C1) is used, which has been configured to alternative pins PB7 for SDA and PB8 for SCL in standard mode with a clock frequency of 100KHz.
The addresses of the used slave devices have a 7-bit address format with the last, 8th bit setting the read or write operations.
In addition, the I2C2 bus was configured in the same mode as I2C1 for future use

of the CSP protocol.
To control the connected green and red LEDs, pins PA10 and PA9 were set to output mode in the open-drain setting.
For debugging purposes, the SWD interface was configured, being traced from the MCU to a separate connector on the board and the USART bus (USART2) was configured in default state with baud-rate of 115200bit/s for debug data output.
For communication purposes with the RFM98W radio module, the SPI bus (SPI1) was configured and left in the default state. In addition, output pin PB5 was set in open-drain mode to control the NSS pin of the radio and pin PB4 in interrupt mode (EXTI4) with an internal pull-down resistor connected to the DIO0 pin of the radio to detect received data.
For future use of the GPIO expander on the COM module, pin PB3 was set up in interrupt mode (EXTI3) with an internal pull-down resistor to detect any interrupts on the pins of the IC. Unused pins PC11, PC10 and PA15, which are reserved for future use, were set as open-drain output.

The pinout of the MCU can be seen in the Figure 4.1.

Implementation description

After switching the satellite on, it initializes its peripherals and then the RFM98W into client mode. After an successfull initialization of the RF module the satellite initializes its other modules like INA219 and payload and then starts to perform the main function of the satellite in a super loop. There it waits to receive broadcast data from Ground station on a set address. When it receives such data, it responds to the broadcast address with a packet containing its UID and a DHCP request flag, requesting the Ground station to assign an address. The address is sent back to it again in the form of a broadcast data packet with a DHCP response flag. At this point, the device is registered with the server and all further ongoing communication is point-to-point. When the satellite has an address assigned, it automatically starts sending a status message every 15 seconds containing its uptime and voltage and current output. When the satellite picks up a packet containing data (flag data), a function is executed that parses the incoming data, performs the action, and then sends back a response (unless it is a RESET command, which restarts the satellite immediately without a response - this is in case something blocks the RF TX output, which would make the reset hanging). The switching between RF RX/TX modes is done by defaulting the module to TX mode, and when incoming data is detected, an interrupt is triggered and it switches to RX mode where it starts receiving data and storing it in the buffer. After storing the incoming data, it switches back to TX mode and the received data are processed in a super loop. The resulting compiled

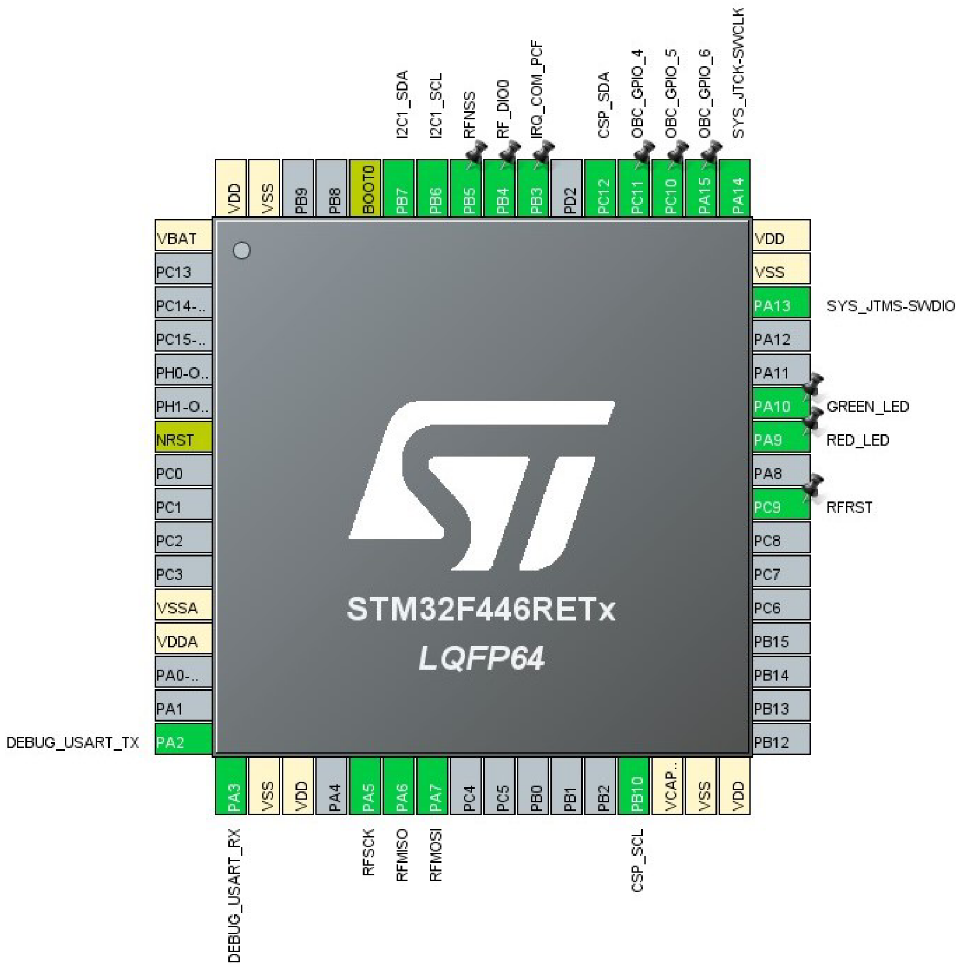


Fig. 4.1: Configuration of the satellite MCU in STM32CubeIDE.

application takes around 41.47KB (depending on the connected modules), which takes only 2 memory sectors of the 512KB flash size.
Diagram of the application operations is in Figure 4.2.

4.1.3 Ground station Firmware

Device configuration

For communication purposes with the RFM98W radio module, the SPI bus (SPI1) was configured in the alternate function mode and left in the default state. In addition, output pins PB8 and PB9 were set in open-drain mode to control the NSS pin and reset pin of the radio respectively. Pin PA9 was set to interrupt mode (EXTI9) with an internal pull-down resistor connected to the DIO0 pin of the radio to detect received data. For debugging purposes, the SWD interface with SWO (for

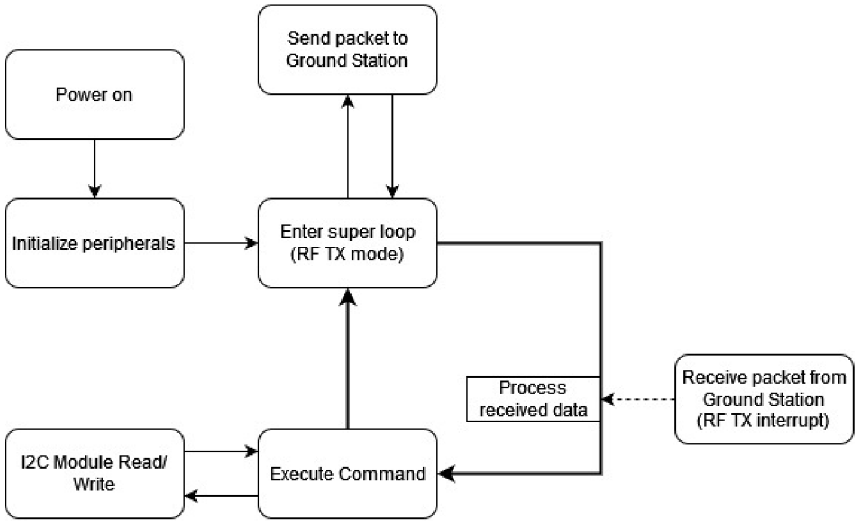


Fig. 4.2: Satellite operations diagram.

Serial Wire Viewer output) was configured, and the USART bus (USART3) was set in DMA mode with interrupts enabled and baud-rate of 115200bit/s. USART interface is used for sending and receiving the data from computer by connecting the USART3 RX/TX pins to the TX/RX pins on the built-in ST-Link which is then connected via an USB cable to the computer as a virtual COM port.
All other unused connected pins were left in default reset state. The pinout of the MCU can be seen in the Figure 4.3.

Implementation description

When the Ground station application starts, it initializes all its peripherals and then the RFM98W module into server mode and then starts to perform the main function in a super loop. There it sends broadcast data (ping) to all devices connected to the same network address every 15 seconds. When the Ground station receives a DHCP request from a satellite, it performs a check of the already registered clients by UID and if device is not found among them, it performs the registration and assigns an ID to the said device. If the device is already registered. it simply returns its client ID. The Ground station then replies to the satellite with the assigned ID via broadcast packet with DHCP response flag. Every communication from that moment is point-to-point except regular broadcast pings.

When the satellite picks up an RF packet containing data (flag data), it forwards it using the USART-USB interface to the terminal application where the packet is further processed. If it intercepts incoming communication from the terminal into the DMA buffer, it sets the flag for later processing in the subsequent interrupt

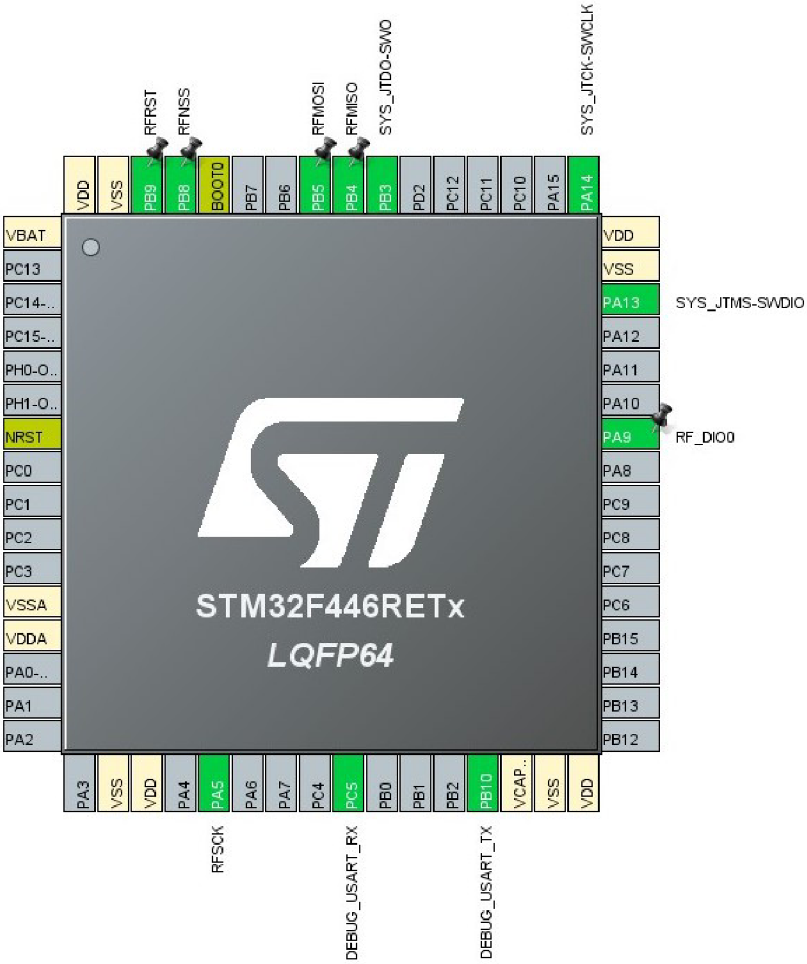


Fig. 4.3: Configuration of the Ground station MCU in STM32CubeIDE.

handler and copies the incoming data into the buffer using memcpy. The later processing checks if the incoming command is valid and if so, it is send via RF to the satellite.

The switching between RF RX/TX modes is done by defaulting the module to TX mode, and when incoming data is detected, an interrupt is triggered and it switches to RX mode where it starts receiving data and storing it in the buffer. After storing the incoming data, it switches back to TX mode and the received data are processed in a super loop. The resulting compiled application is 25.84KB, which takes only 1 memory sector of the 512KB flash size. Diagram of the Ground station operations can be seen in Figure 4.4.

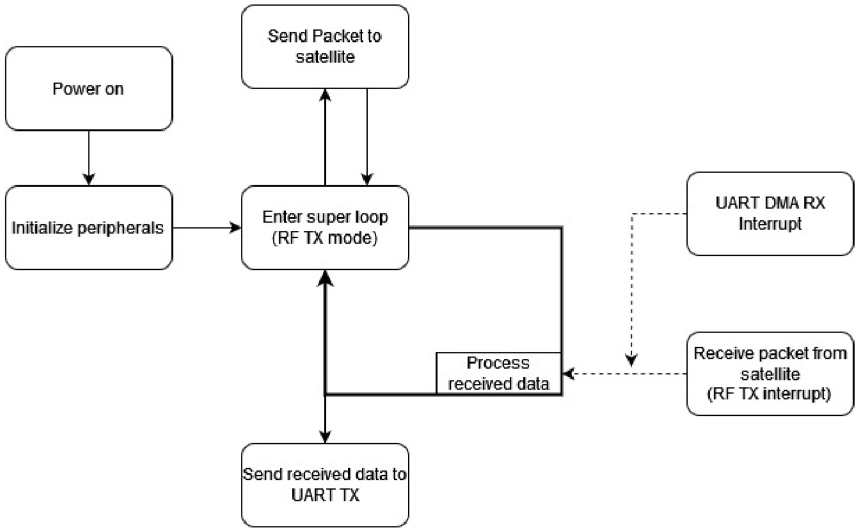


Fig. 4.4: Ground station operations diagram.

4.2 Terminal application

A terminal application in python was written to operate the ground station. After connecting to the correct serial port in non-blocking mode, the application allows to read received data and send commands to the satellite. The received data are interpreted into human readable format, including checking the correct structure of the received packet frame and data. The application waits in an infinite loop for a flag indicating the presence of data waiting to be read and processes the data upon receipt. User input is implemented in the form of a system interrupt, where commands entered by the user are processed in a pseudo-parallel way and are only sent after the enter key is pressed, thus avoiding blocking the main loop. The application has also implemented a simple help output, which can be displayed using the "help" command, or it will display itself on invalid user input. The help contains a short description of the application and a list of supported commands with their purpose.

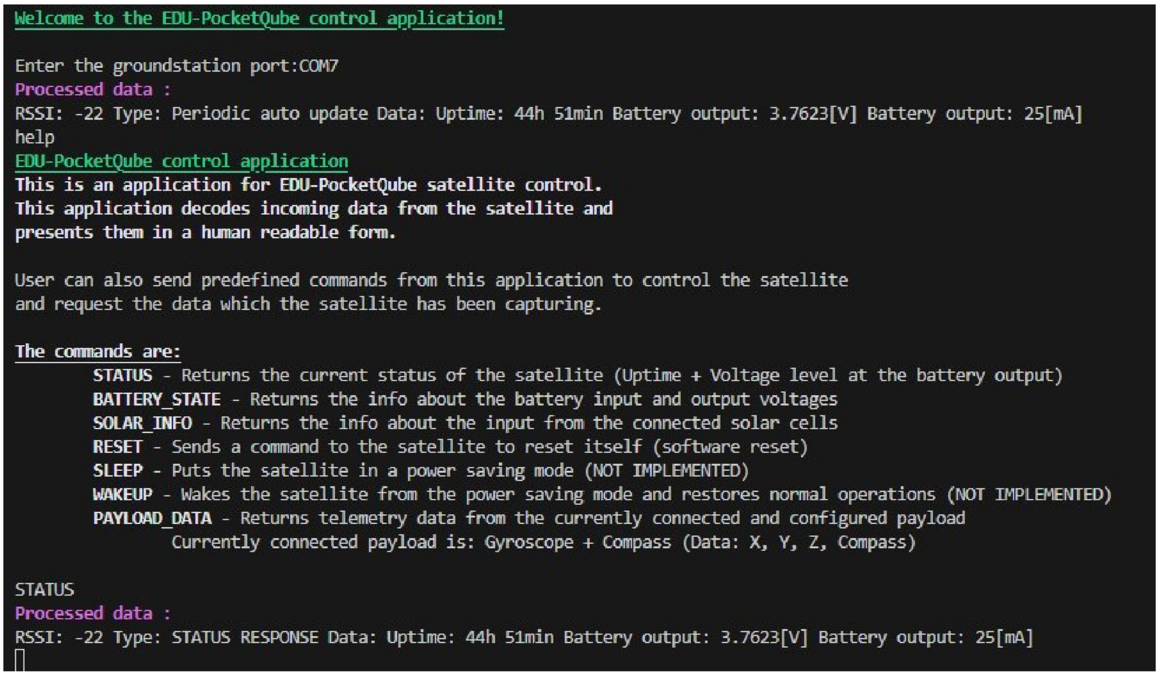


Fig. 4.5: Terminal application example output.

5 Communication

This chapter describes the communication module used, its available modulation types, modes, configurations and default packet types. It also describes the supported user commands. At the end of the chapter, the methods and testing process used to make the modifications are described.

5.1 RF communication description

For communication between the satellite and Ground station RFM98W modules, the modem was set to LoRa mode with bandwidth 125kHz, code rating 4/8 and spreading factor 4096chips/symbol. The measured RSSI was calculated according to the formula in the datasheet[4], where the constant -137 is subtracted from the value in the register. Also, CRC was enabled in the header of the sent/received packets, making it possible to detect if the data were not corrupted.

5.1.1 Modulations

The module supports both standard FSK and long range spread spectrum (LoRa) modems. This is configured by setting the variable “LongRangeMode“ in the “RegOpMode“ register, only in the sleep mode. If FSK modem is selected, user can choose if they wish to use the (G)FSK, OOK or (G)MSK (de)modulation by setting the “ModulationType“ variable in the “RegOpMode“ register. In the LoRa mode, the address of the “ModulationType“ variable called “AccessSharedReg“ instead and its purpose is to enable or disable the access to shared FSK registers. Configuration registers and their options are described in Table 5.1.

RFM98W Register Map				
Name (Address)	Bits	Variable name	Value	Description
RegOpMode (0x01)	7	LongRangeMode	0	FSK/OOK Mode
			1	LoRa Mode
(FSK/OOK mode only)	6	ModulationType	00	FSK
			01	OOK
(LoRa mode only)	6	AccessSharedReg	00	LoRa (0x0D - 0x3F)
			01	FSK (0x0D - 0x3F)

Tab. 5.1: Modem and modulation configurations.[4]

5.1.2 Modes

To set the current device mode, variable "Mode" in the RegOpMode register has to be configured.

Available device modes are listed in Table 5.2.

RFM98W Register Map				
Name (Address)	Bits	Variable name	Value	Description
RegOpMode(0x01)	2-0	Mode	000	SLEEP
			001	STDBY
			010	FSTX
			011	TX
			100	FSRX
			101	RXCONTINUOUS
			110	RXSINGLE
			111	CAD

Tab. 5.2: Modes configurations.[4]

Each mode has its own purpose, which is described in the Table 5.3 below.

Operating Mode	Description
SLEEP	Low-power mode. In this mode SPI registers except LoRa FIFO are accessible.
STDBY	RF and PLLs disabled, Crystal oscillator and LoRa are turned on.
FSTX	TX Frequency synthesis mode.
TX	Transmit mode.
FSRX	RX Frequency synthesis mode.
RXCONTINUOUS	Receives data continuously until change to OpMode is made.
RXSINGLE	Waits for reception of a valid packet and then returns to STDBY.
CAD	Device will check a given channel to detect LoRa preamble signal.

Tab. 5.3: Description of modes.[4]

5.1.3 Configurations

To define the device bandwidth, coding rate, spreading factor and CRC, user must configure the "RegModemConfig" register. The register, its variables and values are described in Table 5.4.

RFM98W Register Map				
Name (Address)	Bits	Variable name	Value	Description
RegModemConfig 1 (0x1D)	7-4	Bw	0000	7.8kHz
			0001	10.4kHz
			0010	15.6kHz
			0011	20.8kHz
			0100	31.25kHz
			0101	41.7kHz
			0110	62.5kHz
			0111	125kHz
			1000	250kHz
			1001	500kHz
	3-1	CodingRate	001	4/5
			010	4/6
			011	4/7
			100	4/8
RegModemConfig 2 (0x1E	7-4	SpreadingFactor	6	64chips/symbol
			7	128chips/symbol
			8	256chips/symbol
			9	512chips/symbol
			10	1024chips/symbol
			11	2048chips/symbol
			12	4096chips/symbol
	2	RxPayloadCrcOn	0	CRC off
			1	CRC on

Tab. 5.4: Bandwidth, coding rate, spreading factor and CRC configurations.[4]

5.1.4 Packet types

To be able to filter and sort the types incoming communications, flags from the RadioHead[5] library described in Table 5.5 were used.

Flag	Value	Description
RH_FLAGS_NONE	0x00	Empty packet
RH_FLAGS_DHCP_RQ	0x01	DHCP request packet
RH_FLAGS_DHCP_RSP	0x02	DHCP response packet
RH_FLAGS_BCAST	0x03	Broadcast data
RH_FLAGS_DATA	0x04	Data

Tab. 5.5: Header flags definitions.[5]

5.1.5 Supported RF communication commands

To control the satellite and obtain data from individual modules, the satellite provides the option of implementing custom commands. The satellite currently supports the following 7 commands, described in table 5.6

Command	Value	Description
STATUS	0x01	Request for satellite status (Uptime + batery output)
BATTERY_STATE	0x02	Request for the state of the battery (input/output energy)
SOLAR_INFO	0x03	Request for solarpanels input
RESET	0x04	Reset satellite
SLEEP	0x05	Put satellite in sleep mode (not implemented)
WAKEUP	0x06	Wake the satellite up (not implemented)
PAYLOAD_DATA	0x07	Request for payload data

Tab. 5.6: Supported commands.

The final form of the outgoing RF packet with the command is described in table 5.7, incoming RF packets are described in table 5.8, where N denotes the packet length in bits.

Bit	6	5	4	3	2	1	0
Meaning	To	From	Id	RH_FLAGS_DATA	RSSI	0x1	Command

Tab. 5.7: Structure of the Ground station outgoing command packet.

The incoming packet is then deconstructed and sent to the terminal application for interpretation. An important feature is that the received data is surrounded on both sides by the command number to which it belongs, allowing subsequent detection of the correct format of the received data.

Bit	N	N-1	N-2	N-3	N-4	N-5	N-6	...	0
Meaning	To	From	Id	RH_FLAGS_DATA	RSSI	Length	Command	Data	Command

Tab. 5.8: Structure of the Ground station incoming command packet.

5.2 RF Testing realization

Communication testing was conducted in three phases.

1. Initial testing of the link, without antenna and data,
2. Stability testing of the communication in a highly populated environment,
3. Applying fixes based on the results from Phase 1 and 2. and their testing.

In phase 1, it was confirmed that the satellite is able to establish a stable link with the ground station and an average signal quality of -65 RSSI. In Phase 2, the equipment was taken outdoors into a highly populated area (approx. >1000 people/km2) to evaluate the effects of the environment on signal quality. It was found that even with a very short distance between the satellite and the ground station (about 15cm), the signal quality was only -90 RSSI and the link range was at most 15m. Based on this data, it was decided to purchase and install COTS antennas tuned to 433MHz frequency, the application of which brought a dramatic improvement in signal quality, whose average value is now between -20 and -40 RSSI, depending on the distance of the devices from each other.

6 Educational Outcomes

This thesis also includes a set of tasks for education in STEM laboratory for working with the resulting satellite. The aim of these laboratory tasks is to provide insight into a simulated real development environment, where students will practice a teams cooperation, project leadership, system engineering, problem solving and will get experiences in fields of firmware, hardware, structure and communication development.

The tasks described in the following pages are:

- Power management system (Smart EPS) (Appendix A),
- Structure and solar cells design (Appendix B),
- Communication and antenna design (Appendix C),
- Scientific payload development (Appendix D),
- Satellite software development (Appendix E).

These tasks are designed and can be realised separately, however to ensure compatibility among all resulting solutions, teams are encouraged to cooperate between each other. In the ideal scenario students will create their own, entirely new PocketQube satellite which will be incorporated in the educational process for the next class of students.

7 Final testing results

Most of the bugs described in the previous chapters were found during the testing processes described in this chapter.

The satellite was tested in several configurations:

- OBC + COM only, powered by the connected ST-Link debugger,
- Full assembly, battery power only,
- Entire assembly, external system power and battery charging.

Each of these combinations confirmed that the satellite is capable of autonomous long-term operation without any major problems. The most interesting result was the discovery that the satellite was able to operate for nearly 40 hours straight on a single charge, during which time it was autonomously sending periodical data about its status.

The two available payloads (both GMC5883 and MPU9250/6500) were tested only on the Nucleo, which serves as the Ground station controller, but worked flawlessly and returned all the necessary data. An unresolved problem remained with the INA219 battery input and output current / voltage monitors, which were tested in the final satellite assembly and were found to return incorrect values. The independent COTS module did not have this problem, so the problem is most likely a faulty wiring or sensor calibration.

The testing was done in the following way:

1. Start the Ground station,
2. Start the Terminal application on the correct port,
3. Start the satellite,
4. Wait for the broadcast packet to start registering the client (satellite) on the server (Ground station),
5. After successful registration, test each supported command and its response,
6. Restart the client device (satellite),
7. Repeat several times from step 4.,
8. Leave the satellite running autonomously, checking the status occasionally,
9. Analyse the results.

This process was used to evaluate the stability of both the RF and USART communication and the used firmware.

To capture the debug data, a logic analyzer connected to the I2C and SPI interfaces was used.Figure 7.2 and Figure 7.1 contains an example of captured packets on the buses. The progress of testing (receiving packets from the satellite) can be seen in Figure 7.4. It is worth noting the RSSI value in Figure 7.3, which is very

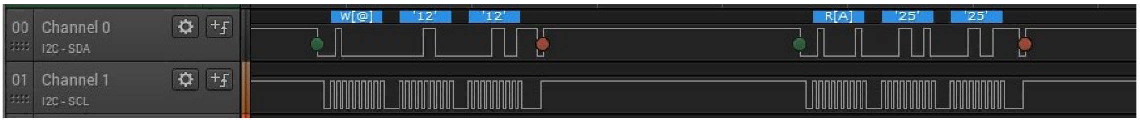


Fig. 7.1: Data captured by logic analyzer on I2C bus.

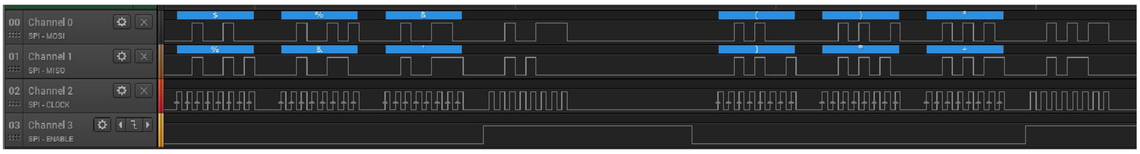


Fig. 7.2: Data captured by logic analyzer on SPI bus.

low compared to Figure 7.4. These are data from testing that took place before the addition of the tuned 433MHz antennas, which dramatically improved the signal quality.

The ADCs connected to the solar panels were not tested, as the primary goal was to verify proper bus functionality and RF communication.

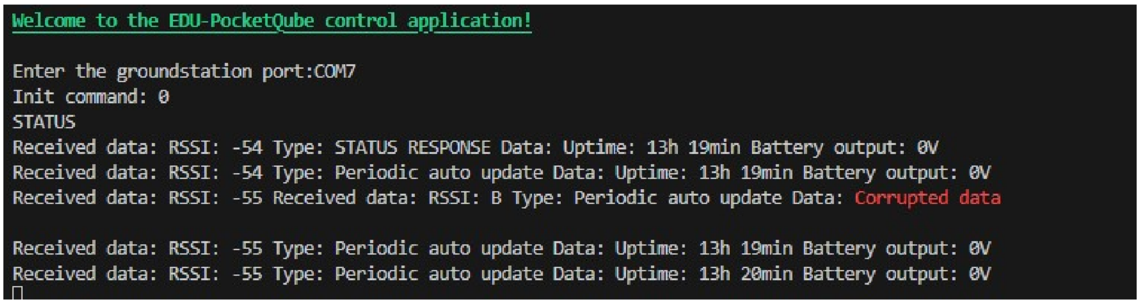


Fig. 7.3: Output from Terminal application - communication without antennas.

During the testing phase, it was also found, that the proposed structure is very impractical for educational purposes, as it is complicated to assemble and takes a lot of time.

Since much of the testing was done with the satellite closed, it was not possible to monitor the status of the internal OBC and EPS (TP4056) LEDs, so the next version will include LEDs routed to the external structure as well, so that the status of the satellite can be monitored without having to disassemble it.

```
Processed data :
RSSI: -21 Type: Periodic auto update Data: Uptime: 66h 47min Battery output: 3.7623[V] Battery output: 25[mA]
Processed data :
RSSI: -22 Type: Periodic auto update Data: Uptime: 66h 47min Battery output: 3.7623[V] Battery output: 25[mA]
Processed data :
RSSI: -22 Type: Periodic auto update Data: Uptime: 66h 48min Battery output: 3.7623[V] Battery output: 25[mA]
Processed data :
RSSI: -22 Type: Periodic auto update Data: Uptime: 66h 48min Battery output: 3.7623[V] Battery output: 25[mA]
Received no data
RSSI: -22 Type: Periodic auto update Data: Uptime: 66h 48min Battery output: 3.7623[V] Battery output: 25[mA]
Processed data :
RSSI: -22 Type: Periodic auto update Data: Uptime: 66h 48min Battery output: 3.7623[V] Battery output: 25[mA]
Processed data :
RSSI: -22 Type: Periodic auto update Data: Uptime: 66h 49min Battery output: 3.7623[V] Battery output: 25[mA]
```

Fig. 7.4: Output from Terminal application - communication with antennas.

Conclusion

The scope of this work was to investigate the design of a PocketQube format satellite in 1.5p or 2p format that would be suitable for educational purposes. Based on this investigation, a design for a 2p satellite was created. The location of the battery was chosen based on the type used (Li-Ion) so that it would fit comfortably inside the structure, and the communication was designed based on the used RFM98W module with LoRa modulation capability. This type of modulation was then utilized for the communication between the satellite and the ground station, as it is a relatively energy-efficient method of communication. Block diagrams of individual modules, their circuit designs and wiring diagrams were created as a part of the satellite design. The design was then implemented into a single PocketQube satellite, with several errors found during the implementation process, that will need to be corrected in the future. The gyroscope, accelerometer, and magnetometer modules were used as payloads in the testing, and were used to test the bus configurations before the actual satellite was built. All internal connections within the satellite were implemented through a unified interface, the so-called system bus, which allows the individual modules to communicate with each other. The resulting functionality of the satellite was tested afterwards, and it was confirmed that all configurations work as they should and the satellite and ground station are able to communicate with each other without any problems.

Based on the results of the test phase, the communication module was slightly modified by adding an external antenna tuned to 433MHz. A set of lab projects have been designed for use in the STEM laboratories, that can be implemented with the satellite, but students are encouraged to collaborate and create their own new version of the PocketQube satellite.

As a result, it is a fully functional device that meets all the points of the assignment and will be ready for use in the classroom once all the issues have been addressed.

In writing the text of this thesis, the DeepL tool was used for checking the grammar of the English language.

Bibliography

[1] Alba Orbital. Alba Orbital - Unicorn-2 Satellite Platform, 2020. URL: http://www.albaorbital.com/unicorn-2.

[2] eoportal. Alba Orbital Unicorn PocketQubes - eoPortal. URL: https://www.eoportal.org/satellite-missions/alba-orbital#launches.

[3] TME. RFM98W-433S2 HOPE MICROELECTRONICS - Modul: transceiver | RF; FSK,GFSK,GMSK,LoRa,MSK,OOK; 433,92MHz | TME Czech Republic s.r.o. - Elektronické součástky. URL: https://www.tme.eu/cz/details/rfm98w-433s2/moduly-rf/hope-microelectronics/.

[4] HopeRF. Low Power Long Range Transceiver Module RFM95/96/98(W) V2.0., 2018.

[5] RadioHead. RadioHead: RadioHead Packet Radio library for embedded microprocessors. URL: https://www.airspayce.com/mikem/arduino/RadioHead/.

[6] libcsp/libcsp, May 2024. URL: https://github.com/libcsp/libcsp.

[7] DeepL. DeepL Translate: Nejpřesnější překladač na světě. URL: https://www.deepl.com/translator.

[8] S Radu, M S Uludag, S Speretta, J Bouwmeester, A Menicucci, A Cervone, A Dunn, and T Walkinshaw. PocketQube Standard, 2018. URL: http://www.albaorbital.com/pocketqube-standard.

[9] A. Povalač. *Nanosatellite Design and Electronics*. Brno: Brno University of Technology, Faculty of electrical engineering and communications, 2021.

[10] CAPPELLETTI, Chantal, Simone BATTISTINI, Benjamin K. MALPHRUS. *CubeSat Handbook: From Mission Design to Operations*. Elsevier Science, 2020.

[11] Alba Orbital. Unicorn-2C (NOOR 1B) Success on orbit, April 2020. URL: http://www.albaorbital.com/new-blog/2020/4/10/unicorn-2c-noor-1b-success-on-orbit.

[12] VEVERKA, J; POVALAČ, A.; JAŠŠO, K. Educational PocketQube Satellite Demonstrator. *In Proceedings of the 30th Conference STUDENT EEICT 2024 Selected Papers, University of Technology: Faculty of Electrical Engineering and Communication*, 2024.

[13] Europian Space Association. About Payload Systems, 2024. URL: https://www.esa.int/Enabling_Support/Space_Engineering_Technology/About_Payload_Systems.

[14] EVONA. What Is... A Payload? | EVONA, 2024. URL: https://www.evona.com/blog/what-is-a-payload/.

[15] KOVARIK, V;. *Using satellites in Remote Sensing*. Brno: Brno University of Technology, Faculty of Mechanical Engineering; Space Structures Design (OK3-A), 2022.

[16] Gunter's Space Page. NOOR 1A, 1B (Unicorn 2B, 2C), 2024. URL: https://space.skyrocket.de/doc_sdat/noor-1.htm.

[17] DIY Satellite Ground Station and Network, January 2015. URL: https://satnogs.org/diy-satellites/.

[18] STMicroelectronics. STM32F446RE - High-performance foundation line, Arm Cortex-M4 core with DSP and FPU, 512 Kbytes of Flash memory, 180 MHz CPU, ART Accelerator, Dual QSPI - STMicroelectronics. URL: https://www.st.com/en/microcontrollers-microprocessors/stm32f446re.html.

[19] JaeWoo Choi. IXOLARTM High Efficiency SolarMD. 2023.

[20] interfaces:semtech - wiki.kewl.org. URL: https://wiki.kewl.org/dokuwiki/interfaces:semtech.

[21] SX1278. URL: https://semtech.my.salesforce.com/sfc/p/#E0000000JelG/a/2R0000001Rc1/QnUuV9TviODKUgt_rpBlPz.EZA_PNK7Rpi8HA5..Sbo.

[22] NXP Semiconductors. PCF8574; PCF8574A Remote 8-bit I/O expander for I2C-bus with interrupt. Datasheet PCF8574_PCF8574A, May 2013. URL: https://www.nxp.com/docs/en/data-sheet/PCF8574_PCF8574A.pdf.

[23] STMicroelectronics. UM 1724 User Manual - STM32 Nucleo-64 boards (MB1136).

[24] KiCad. KiCad EDA. URL: https://www.kicad.org/.

[25] SOLIDWORKS. SOLIDWORKS 3D CAD, March 2024. URL: https://www.solidworks.com/product/solidworks-3d-cad.

[26] JLCPCB. PCB Prototype & PCB Fabrication Manufacturer - JLCPCB. URL: https://jlcpcb.com/.

[27] ATMEL. 24C256 - 256Kb Two-wire Serial EEPROM. URL: https://www.laskakit.cz/user/related_files/24c256.pdf.

[28] WE: Würth Elektronik. 8MHz Quartz Crystal Specification.

[29] Inven Sense. MPU-6500 Product Specification. URL: https://www.laskakit.cz/user/related_files/mpu-6500-datasheet2.pdf.

[30] Texas Instruments. ADS111x-Q1 Automotive, Low-Power, I2C-Compatible, 860-SPS, 16-Bit ADCs. URL: https://www.ti.com/lit/ds/symlink/ads1113-q1.pdf?ts=1716161686344&ref_url=https%253A%252F%252Fcz.mouser.com%252F.

[31] NanJing Top Power ASIC Corp. TP4056 1A Standalone Linear Li-Ion Battery Charger with Thermal Regulation in SOP-8. URL: https://dratek.cz/docs/produkty/1/1113/tp4056.pdf.

[32] Texas Instruments. INA219 Zero-Drift, Bidirectional Current/Power Monitor with I2C Interface. Datasheet SBOS448G, December 2015. URL: https://www.ti.com/lit/ds/symlink/ina219.pdf.

[33] IRF - International Rectifier. IRLML6402PbF Power MOSFET. URL: https://www.infineon.com/dgdl/irlml6402pbf.pdf?fileId=5546d462533600a401535668d5c2263c.

[34] Nexperia. 2N7002 60 V, 300 mA N-channel Trench MOSFET. *mA N*, 2017, 2011.

[35] Microchip. MCP1703A - 250 mA, 16V, Low Quiescent Current LDO Regulator.

[36] Yoido-Dong Youngdungpo-Gu. Rechargeable Lithium Ion Battery Model : INR18650 F1L.

[37] G-NiceRF. 433MHz Straight Rod Antenna SW433-ZT28. URL: https://www.laskakit.cz/user/related_files/433mhz_straight_rod_antenna_sw433-zt28_v2-2.pdf.

[38] G-NiceRF. 433MHz Elbow Rod Antenna SW433-WT36. URL: https://dratek.cz/docs/produkty/0/403/1511245948_nicerf_433mhz_ant_na_prutov_hlov_sw433_wt36_45_mm.pdf.

[39] x893. x893/LMiC GitHub Repository, 2024. URL: https://github.com/x893/LMiC/tree/master.

[40] STMicroelectronics. On-the-fly firmware update for dual bank STM32 microcontrollers, 2019.

[41] STMicroelectronics. STM32WB WDG TIMERS - Independent-Watchdog IWDG.

[42] STMicroelectronics. STM32WB WDG TIMERS - System Window Watchdog WWDG.

[43] freeRTOS. FreeRTOS - Market leading RTOS (Real Time Operating System) for embedded systems with Internet of Things extensions. URL: https://www.freertos.org/index.html.

Symbols and abbreviations

ADC	Analog-to-Digital converter
ADCS	Attitude Determination and Control System
ARM	Advanced RISC Machine
ASM	Assembly
CAD	Computer-aided design
CAN	Controller Area Network
COM	Communication System
COTS	Commercial off-the-shelf
CRC	Cyclic Redundancy Check
CSP	CubeSat Space Protocol
D/A	Digital-to-Analog converter
DHCP	Dynamic Host Configuration Protocol
DMA	Direct Memory Access
EEPROM	Electrically Erasable Programmable Read-Only Memory
EO	Earth Observation
EPS	Electrical Power System
FPU	Floating Point Unit
freeRTOS	free Real-Time Operating System
FSK	Frequency-shift Keying
FW	Firmware
GPIO	General-purpose input/output
GS	Ground station
HAL	Hardware Abstraction Layer
HSE	High Speed External

IC	Integrated Circuit
IDE	Integrated Development Enviroment
IoT	Internet of Things
IWDG	Independent Watchdog
JTAG	Joint Test Action Group
KB	Kilo-Byte
Kb	Kilo-Bits
LDO	Low-dropout regulator
LED	Light Emitting Diode
LEO	Low-Earth Orbit
LNA	Low-noise amplifier
LoRa	Long Range
LSE	Low Speed External
MCU	Microcontroller Unit
MPPT	Maximum Power Point Tracking
MPU	Memory Protection Unit
OAP	Orbit Average Power
OBC	On-board Computer
OOK	On-Off keying
PCB	Printed Circuit Board
PLL	Phase-locked loop
RF	Radio Frequency
RSSI	Received Signal Strength Indication
RTC	Real-time clock
RX	Receiver

SAW	Surface Acoustic Wave
SMD	Surface Mount Device
SNR	Signal-To-Noise Ratio
SPI	Serial Peripheral Interface
SRAM	Static Random Access Memory
STEM	Science, Technology, Engineering, Mathematics
SW	Software
SWD	Serial Wire Debug
SWO	Serial Wire One
SWV	Serial Wire Viewer
TRL	Technology readiness level
TX	Transmitter
U(S)ART	Universal Synchronous/Asynchronous Receiver and Transmitter
UHF	Ultra High Frequency
USB	Universal Serial Bus
VHF	Very High Frequency
WWDG	Window Watchdog

List of appendices

A Laboratory task 1: Power management system (Smart EPS)

Team size: 1-2 persons
Task objectives:

- Design and implement a circuitry which controls the EPS delivery with a separate MCU (eg. STM32F103C8), providing the system with "smart" control over the power rails.
- Expand the already existing design by 5V power supply.
- Investigate and if possible, use different source of power than in the initial design (eg. LiPol battery).
- Implement the SLEEP, WAKEUP and low power consumption modes in the original code provided by the EDU-PocketQube project.
- Design and implement according to the space industry standards (eg. ECSS) whenever possible.

Task outcomes:

- Design and implementation of DPS for Smart EPS.
- Code for Smart EPS and update to the EDU-PocketQube firmware.
- Task report (around 10-20 A4 pages) containing the description and analysis of the problem, description of the solution, graphs, used parts, firmware implementation description and diagnostic of the resulting system. Don't forget to mention any violations of the space industry standards.

Bonus task:

- Cooperate with other teams to fit the EPS (and all the necessary components) inside of a 1.5P sized PocketQube structure.
- Analyse the effectivity of the battery charging solar cells on the final 1.5P design. Give an answer to the question whether the used solar cells and they configuration are sufficient to keep the battery charged enough for the system to work without any extra external power input.

B Laboratory task 2: Structure and solar cells design

Team size: 1 person project
Task objectives:

- Analyse and find problems with the current solution.
- Design an external structure for a 2P sized PocketQube satellite.
- Investigate available solar cells on the market and select suitable ones for the best coverage / power generation ratio.
- Perform a durability test of the resulting structure (eg. vibrations, drop test, etc.).
- Abide by the space industry standards (eg. PocketQube design standard, ECSS,..) whenever possible.

Task outcomes:

- Design and implementation of the structure for 2P sized PocketQube satellite.
- Task report (around 10 A4 pages) containing the description and analysis of the problem, description of the solution, used materials, manufacturing process and testing process. Don't forget to mention any violations of the space industry standards.

Bonus task:

- Cooperate with other teams and design a 1.5P sized structure which fits all the necessary components.
- Analyse the effectivity of the solar cells on the final 1.5P design in comparison to the current 2P design.

C Laboratory task 3: Communication and antenna design

Team size: 1-2 persons
Task objectives:

- Analyse the properties of radio communication in space (eg. frequencies, bands, modulations, encryption). If possible, focus on LoRa modulation and LoRaWAN protocol.
- Analyse the COTS radio modules widely available on the market and choose a suitable one for your application.
- Design and manufacture a suitable antenna for your radio, that fits in a 2P PocketQube satellite.
- Test the functionality of your radio and antenna and assess its effectivity.
- Implement code for encryption / decryption of the communication.
- Design and implement according to the space industry standards (eg. ECSS) whenever possible.

Task outcomes:

- Design and implementation of antenna.
- Code for the radio module and encrypting algorythms.
- Task report (around 10-20 A4 pages) containing the description and analysis of the problem, description of the solution, graphs, used parts, code implementation and an example of captured encrypted + decrypted data. Don't forget to mention any violations of the space industry standards.

Bonus task:

- Cooperate with other teams and design an antenna, that fits inside of a 1.5P sized PocketQube structure.
- Cooperate with the EPS team to determine the appropriate power consumption.

D Laboratory task 4: Scientific payload development

Team size 1-2 persons
Task objectives:

- Come with an idea for a scientific payload, that ideally fits inside a 2P (or 1.5P) sized PocketQube structure (eventually 1U sized CubeSat) or can be used in a cluster of multiple satellites.
- Analyse the chosen topic and describe its contribution to science.
- Design a prototype of the scientific payload, use COTS parts whenever possible.
- Demonstrate real-life scenario usage of the payload prototype.
- If possible, get data from real testing enviroment (eg. heat chambers, radiation chambers, vibration testing, etc.).
- Abide by the space industry standards (eg. ECSS) whenever possible.

Task outcomes:

- Design and implementation of the scientific payload prototype.
- The code implementing the prototype functionality (if there is any needed).
- Task report (around 10-20 A4 pages) containing the description, analysis and usage of the chosen scientific experiment and prototype. Describe benefits of your solution over other, already existing (if there are any) and suggest improvements to the device based on the testing data results. Don't forget to mention any violations of the space industry standards.

Bonus task:

- Cooperate with other teams and incorporate the resulting device into the developed 1.5P satellite if possible (if not, use the already existing 2P educational satellite).
- Attempt to get your payload in a real space mission.

E Laboratory task 5: Satellite software development

Team size: 1-2 persons

Task objectives:

- Design and implement bootloader and application firmware for the educational PocketQube satellite in freeRTOS. Develop for a microprocessor with an STM ARM architecture (eg. STM32F446RE), compatible with the educational satellite. Pay increased attention to the effective data handling and atomicity of the instructions.
- Incorporate and if needed, implement the supplied educational PocketQube module drivers.
- Design diagrams describing the bootloader-application operations.
- Abide by the (aero)space industry standards (eg. ECSS, DO-178, etc.) whenever possible.

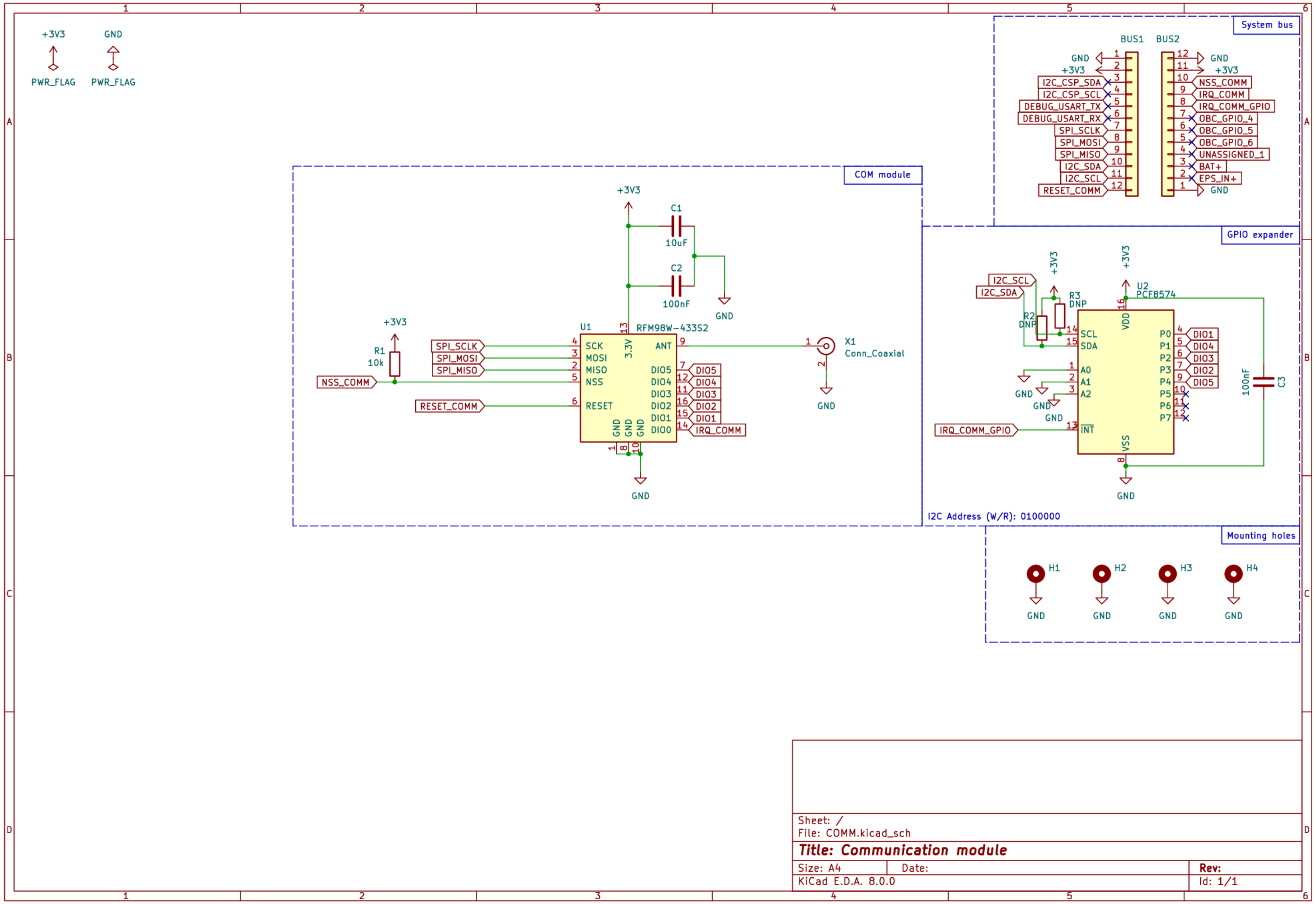
Task outcomes:

- Bootloader and application firmware code compatible with the educational PocketQube satellite.
- Task report (around 10-20 A4 pages) containing the description and analysis of the software design, diagrams, examples of captured peripheral data and suggestions for improvement. Don't forget to mention any violations of the space industry standards.

Bonus task:

- Cooperate with other teams and include their devices into your code. Ensure that everything works correctly as it should.
- Write the code in a strictly no dynamic allocation mode, avoid any goto instructions, use RTEMS distribution of freeRTOS.
- Write a multiplatform application with GUI to control the satellite and to present the received data in an easy understandable way.
- If dual-bank FLASH memory is available, implement on-the-fly update.

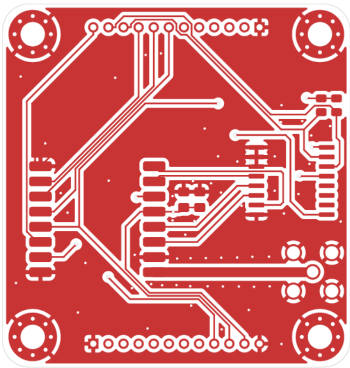
F COM Schematics


+3V3
GND
PWR_FLAG
PWR_FLAG
System bus
BUS1
BUS2
GND
+3V3
I2C_CSP_SDA
I2C_CSP_SCL
DEBUG_USART_TX
DEBUG_USART_RX
SPI_SCLK
SPI_MOSI
SPI_MISO
I2C_SDA
I2C_SCL
RESET_COMM
NSS_COMM
IRQ_COMM
IRQ_COMM_GPIO
OBC_GPIO_4
OBC_GPIO_5
OBC_GPIO_6
UNASSIGNED_1
BAT+
EPS_IN+
COM module
C1
10uF
C2
100nF
R1
10k
U1
RFM98W-433S2
SCK
MOSI
MISO
NSS
RESET
3.3V
ANT
DIO5
DIO4
DIO3
DIO2
DIO1
DIO0
IRQ_COMM
X1
Conn_Coaxial
GPIO expander
I2C_SCL
I2C_SDA
R2
DNP
R3
DNP
U2
PCF8574
VDD
SCL
SDA
A0
A1
A2
INT
VSS
P0
P1
P2
P3
P4
P5
P6
P7
IRQ_COMM_GPIO
C3
100nF
I2C Address (W/R): 0100000
Mounting holes
H1
H2
H3
H4
Sheet: /
File: COMM.kicad_sch
Title: Communication module
Size: A4
Date:
Rev:
KiCad E.D.A. 8.0.0
Id: 1/1

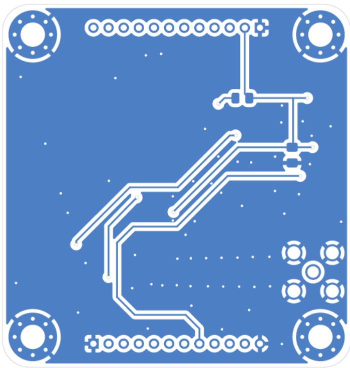
G COM PCB design

PCB dimensions are 48×46mm in a scale of 1:1

G.1 TOP



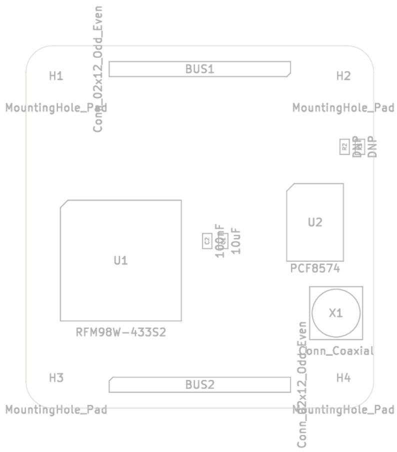
G.2 BOTTOM



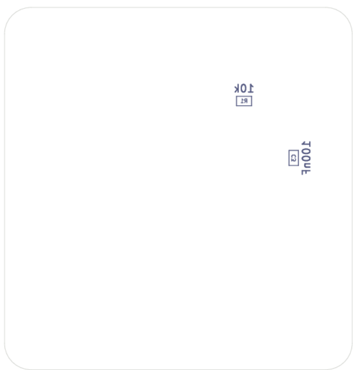
H Component placement - COM

PCB dimensions are 48×46mm in a scale of 1:1

H.1 TOP



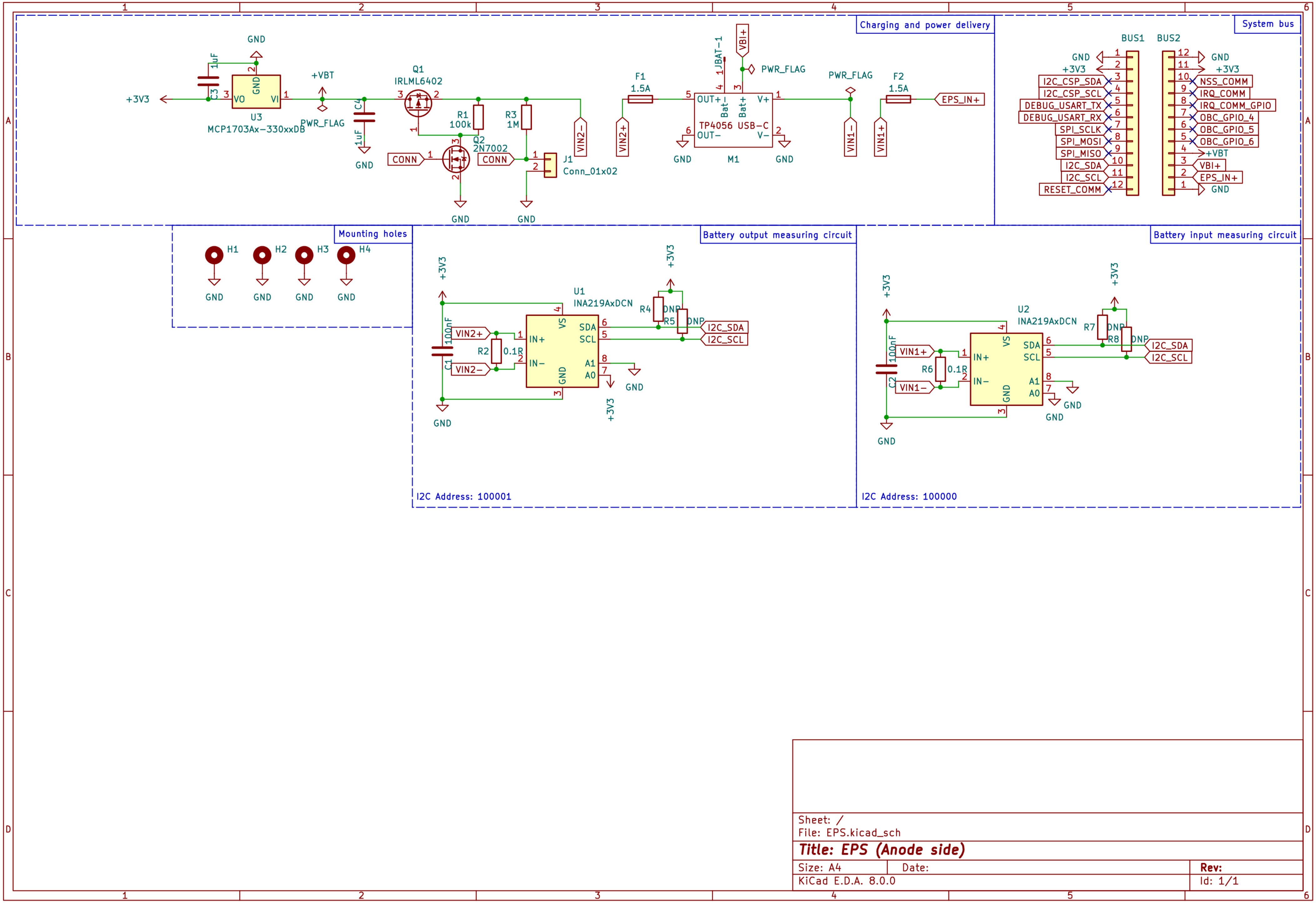
H.2 BOTTOM



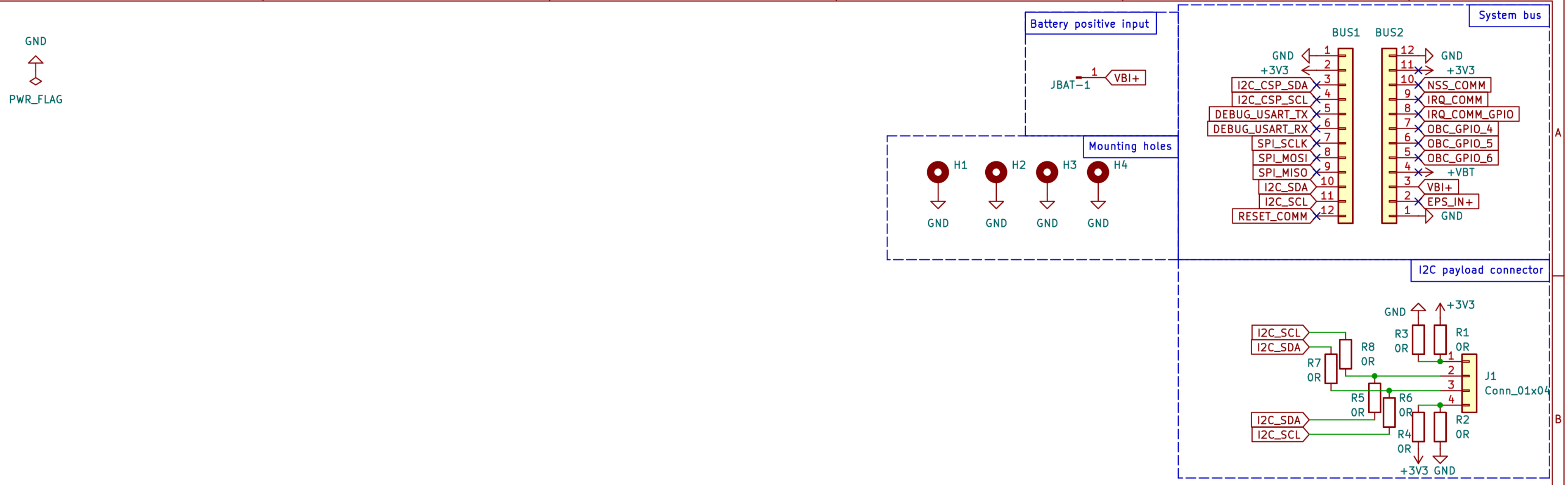
I Bill of Materials - COM

Reference	Value	Footprint	Qty
BUS1, BUS2	Conn_01x12	Connector PinHeader 2.00mm	2
C1	10uF	SMD 0805	1
C2, C3	100nF	SMD 0805	2
R1	10k	SMD 0805	1
R2, R3	DNP	SMD 0805	DNP
U1	RFM98W	RFM98W	1
U2	PCF8574	SOIC-16	1
X1	Conn_Coaxial	Coaxial SMA Vertical	1

J EPS Schematics



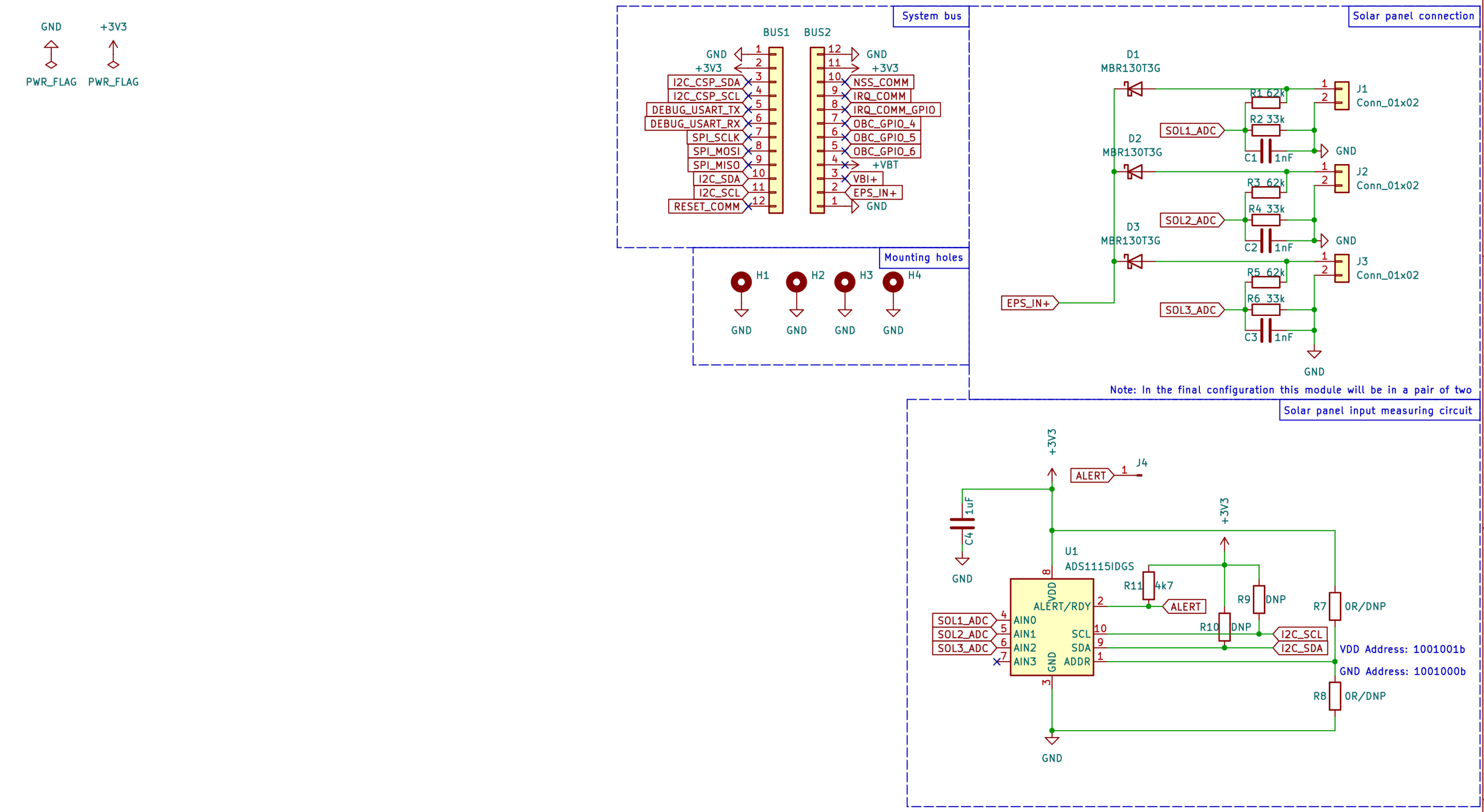
Charging and power delivery
System bus
Mounting holes
Battery output measuring circuit
Battery input measuring circuit
+3V3
GND
+VBT
PWR_FLAG
U3
MCP1703Ax-330xxDB
C3
1uF
C4
1uF
Q1
IRLML6402
Q2
2N7002
R1
100k
R3
1M
CONN
J1
Conn_01x02
VIN2-
VIN2+
F1
1.5A
M1
TP4056 USB-C
JBAT-1
VBI+
VIN1-
VIN1+
F2
1.5A
EPS_IN+
BUS1
BUS2
I2C_CSP_SDA
I2C_CSP_SCL
DEBUG_USART_TX
DEBUG_USART_RX
SPI_SCLK
SPI_MOSI
SPI_MISO
I2C_SDA
I2C_SCL
RESET_COMM
NSS_COMM
IRQ_COMM
IRQ_COMM_GPIO
OBC_GPIO_4
OBC_GPIO_5
OBC_GPIO_6
H1
H2
H3
H4
U1
INA219AxDCN
C1
100nF
R2
0.1R
R4
R5
DNP
I2C Address: 100001
U2
INA219AxDCN
C2
100nF
R6
0.1R
R7
R8
I2C Address: 100000
Sheet: /
File: EPS.kicad_sch
Title: EPS (Anode side)
Size: A4
Date:
Rev:
KiCad E.D.A. 8.0.0
Id: 1/1



Sheet: /
File: EPS_positive.kicad_sch

Title: EPS (Cathode side)

Size: A4	Date:	Rev:
KiCad E.D.A. 8.0.0		Id: 1/1



Sheet: /
File: EPS_ring_conn.kicad_sch

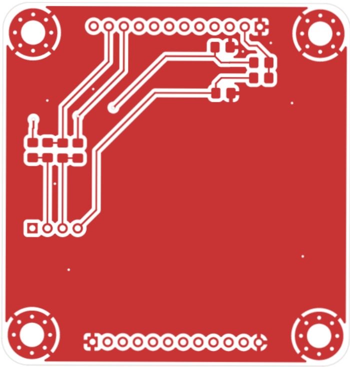
Title: EPS (Ring connector)

Size: A4	Date:	Rev:
KiCad E.D.A. 8.0.0		Id: 1/1

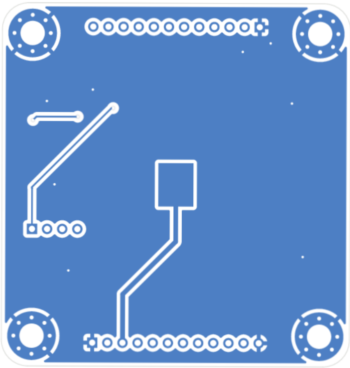
K EPS PCB design

PCB dimensions are 48×46mm in a scale of 1:1

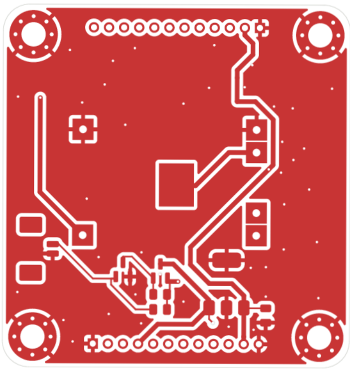
K.1 EPS (Cathode side) TOP



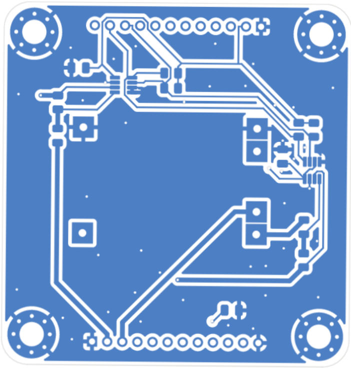
K.2 EPS (Cathode side) BOTTOM



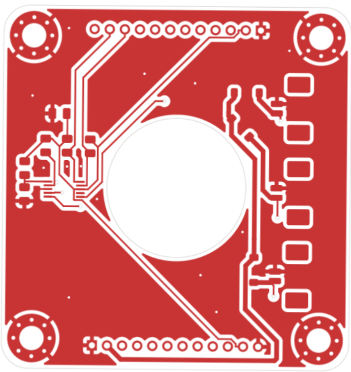
K.3 EPS (Anode side) TOP



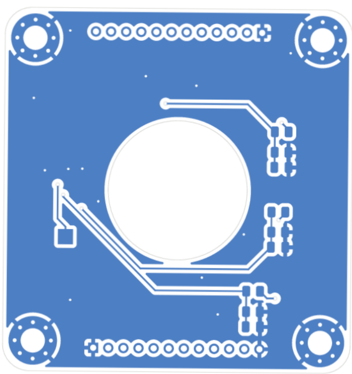
K.4 EPS (Anode side) BOTTOM



K.5 EPS (Ring connector) TOP



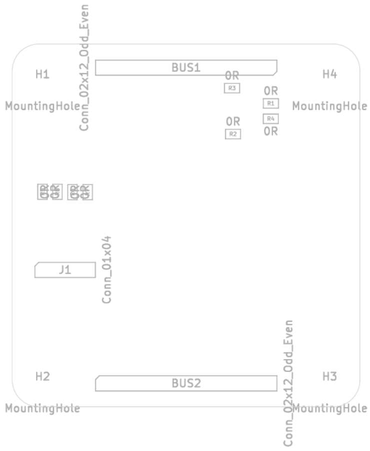
K.6 EPS (Ring connector) BOTTOM



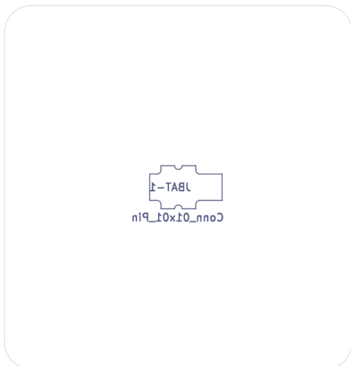
L Component placement - EPS

PCB dimensions are 48×46mm in a scale of 1:1

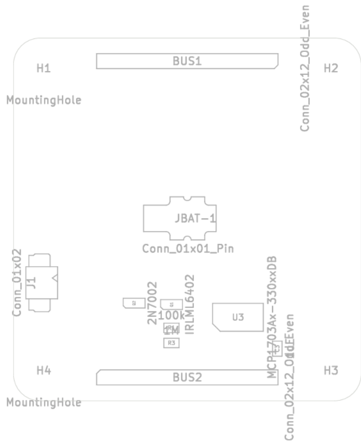
L.1 EPS (Cathode side) TOP



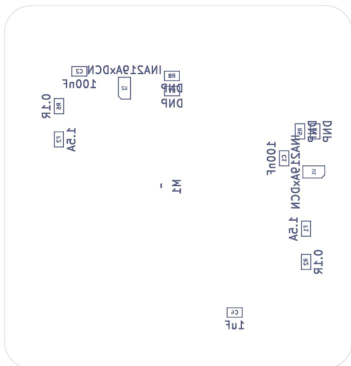
L.2 EPS (Cathode side) BOTTOM



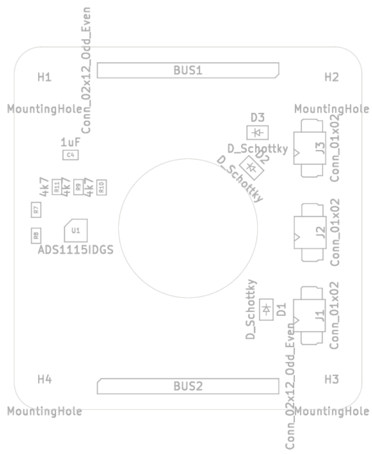
L.3 EPS (Anode side) TOP



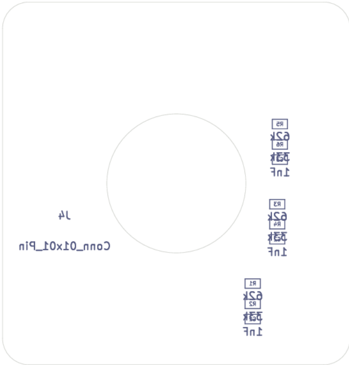
L.4 EPS (Anode side) BOTTOM



L.5 EPS (Ring connector) TOP



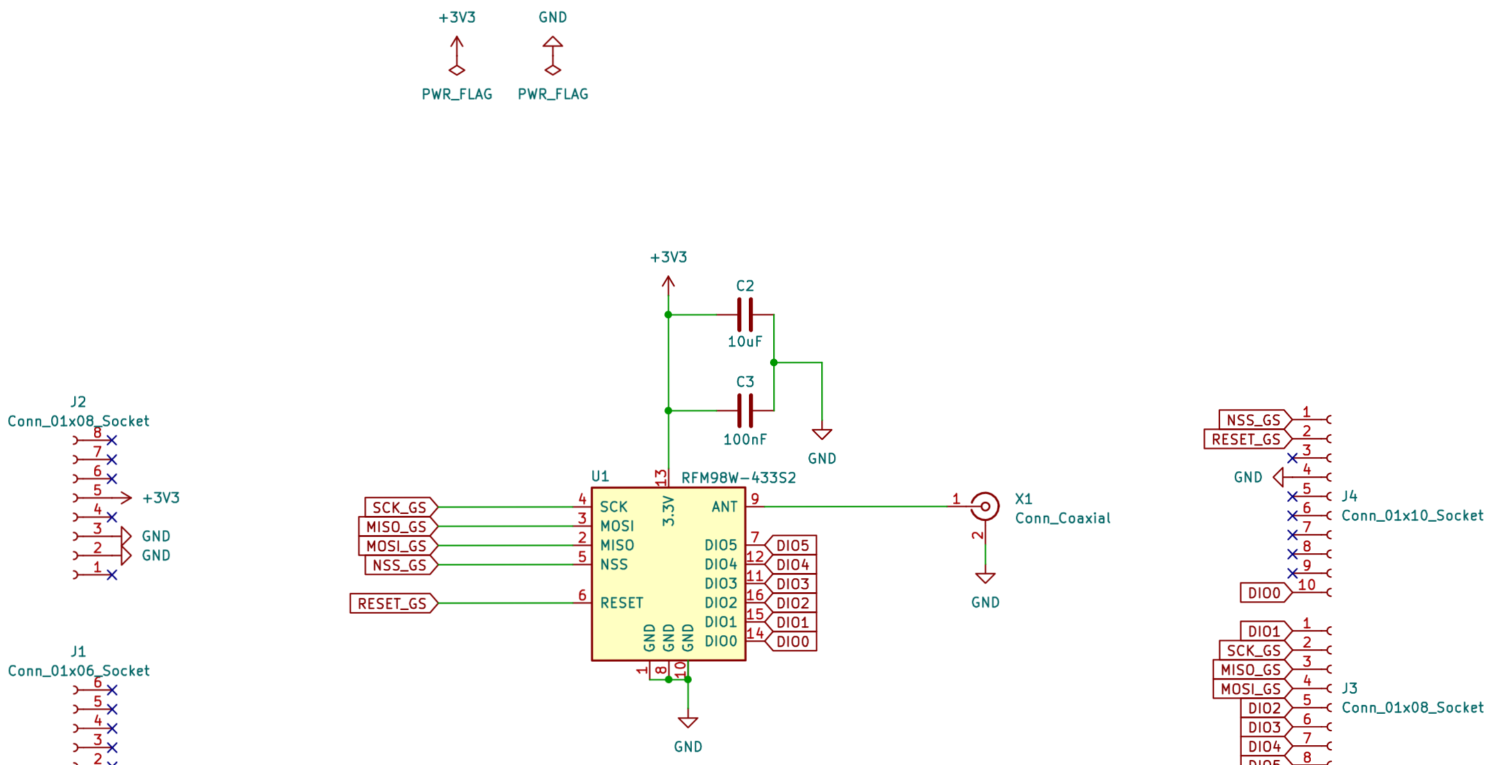
L.6 EPS (Ring connector) BOTTOM



M Bill of Materials - EPS

Reference	Value	Footprint	Qty
EPS Cathode side			
BUS1, BUS2	Conn_01x12	Connector PinHeader 2.00mm	2
J1	Conn_01x04	Connector PinHeader 2.00mm	1
JBAT-1	Battery spring	Battery spring	1
R1, R2, R3, R4, R5, R6, R7, R8	0R	SMD 0805	4
EPS Ring Connector			
BUS1, BUS2	Conn_01x12	Connector PinHeader 2.00mm	2
C1, C2, C3	1nF	SMD 0805	3
C4	1uF	SMD 0805	1
D1, D2, D3	MBR130T3G	D_SOD-123	3
J1, J2, J3	Conn_01x02	Molex PicoBlade	3
J4	Conn_01x01	Connector Pin	1
R1, R3, R5	62k	SMD 0805	3
R2, R4, R6	33k	SMD 0805	3
R7, R8	DNP	SMD 0805	DNP
R9, R10	DNP	SMD 0805	DNP
R11	4k7	SMD 0805	1
U1	ADS1115IDGS	TSSOP-10	1
EPS Anode side			
BUS1, BUS2	Conn_01x12		2
C1, C2	100nF	SMD 0805	2
C3, C4	1uF	SMD 0805	2
F1, F2	1.5A	SMD 0805	2
J1	Conn_01x02	Molex PicoBlade	1
JBAT-1	Battery spring	Battery spring	1
M1	TP4056 Module	TP4056 Module	1
Q1	IRLML6402	SOT-23	1
Q2	2N7002	SOT-23	1
R1	100k	SMD 0805	1
R2, R6	0.1R	SMD 0805	1
R3	1M	SMD 0805	1
R4, R5, R7, R8	DNP	SMD 0805	DNP
U1, U2	INA219AxDCN	SOT-23-8	2
U3	MCP1703Ax-330x	SOT-223-3	1

N Ground station Schematics



Sheet: /
File: Ground Station.kicad_sch

Title: Ground station

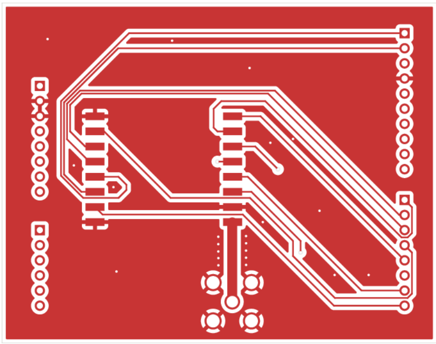
Size: A4 Date:
KiCad E.D.A. 8.0.0

Rev:
Id: 1/1

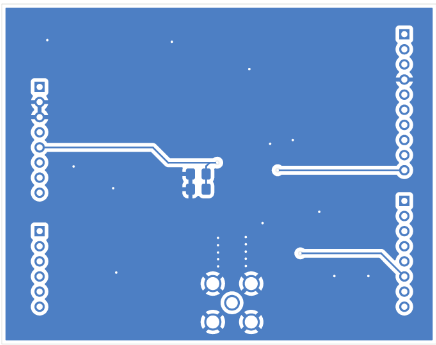
O Ground station PCB design

PCB dimensions are 48×46mm in a scale of 1:1

O.1 TOP



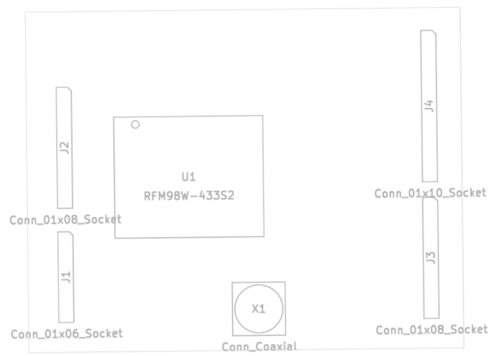
O.2 BOTTOM



P Component placement - Ground station

PCB dimensions are 48×46mm in a scale of 1:1 Components are placed on top side only.

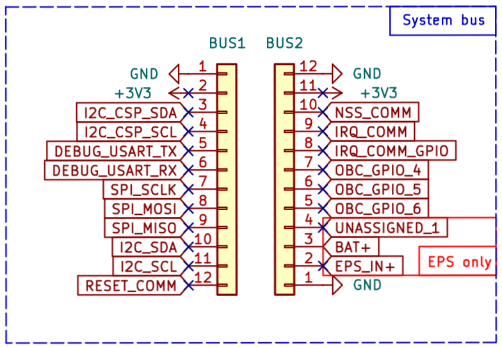
P.1 TOP



Q Bill of Materials - Ground station

Reference	Value	Footprint	Qty
C2	10uF	SMD 0805	1
C3	100nF	SMD 0805	1
J1	Conn_01x06	Connector PinSocket 2.00mm	1
J2, J3	Conn_01x08	Connector PinSocket 2.00mm	2
J4	Conn_01x10	Connector PinSocket 2.00mm	1
U1	RFM98W	RFM98W	1
X1	Conn_Coaxial	Coaxial SMA Vertical	1

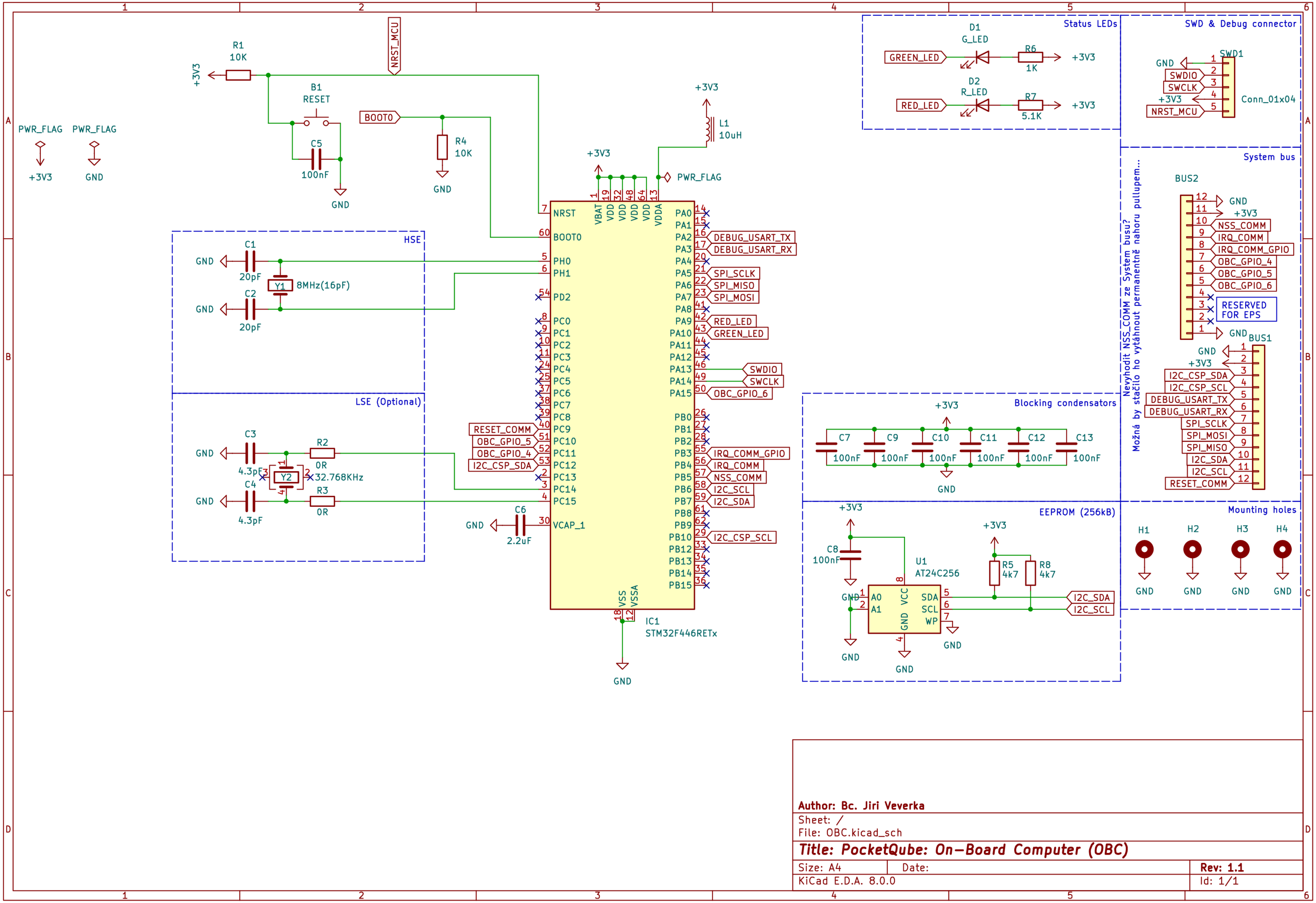
R OBC Schematics



Sheet: /
File: BUS - Conn.kicad_sch

Title: System bus

Size: A4	Date:	Rev:
KiCad E.D.A. 8.0.0		Id: 1/1

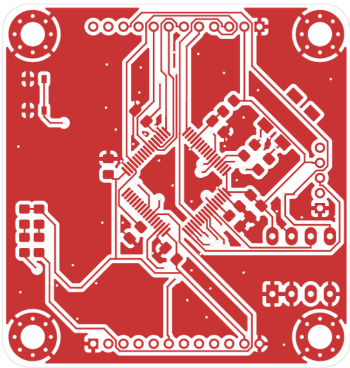


Status LEDs
SWD & Debug connector
System bus
HSE
LSE (Optional)
Blocking condensators
EEPROM (256kB)
Mounting holes
IC1 STM32F446RETx
U1 AT24C256
Nevyhodit NSS_COMM ze System busu?
Možná by stačilo ho vytáhnout permanentně nahoru pullupem...
RESERVED FOR EPS
Author: Bc. Jiri Veverka
Sheet: /
File: OBC.kicad_sch
Title: PocketQube: On-Board Computer (OBC)
Size: A4
Date:
Rev: 1.1
KiCad E.D.A. 8.0.0
Id: 1/1

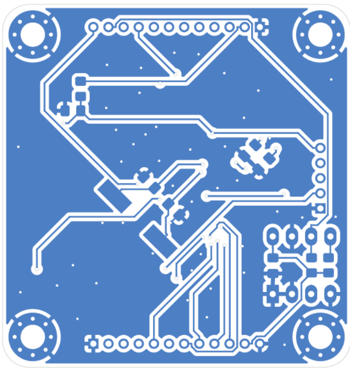
S OBC PCB design

PCB dimensions are 48×46mm in a scale of 1:1

S.1 TOP



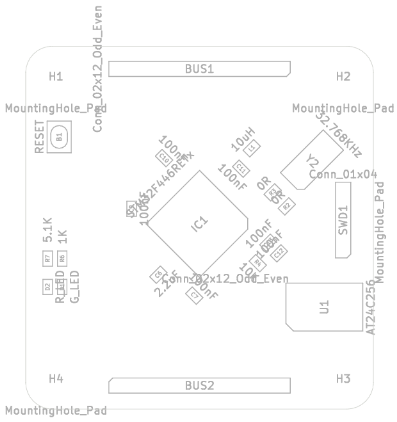
S.2 BOTTOM



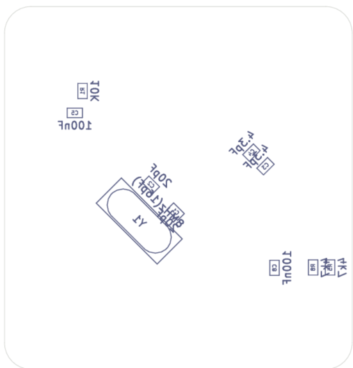
T Component placement - OBC

PCB dimensions are 48×46mm in a scale of 1:1

T.1 TOP



T.2 BOTTOM



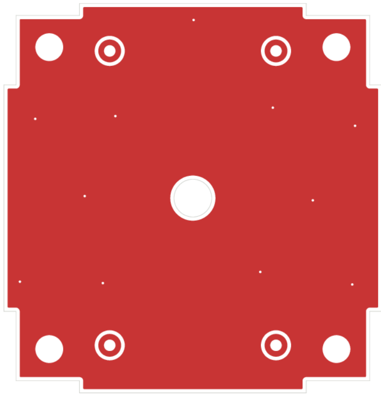
U Bill of Materials - OBC

Reference	Value	Footprint	Qty
B1	RESET	Button switch SMD	1
BUS1, BUS2	Conn_01x12	Connector PinSocket 2.00mm	2
C1, C2	20pF	SMD 0805	2
C3, C4	4.3pF	SMD 0805	2
C5, C7, C8, C9, C10, C11, C12, C13	100nF	SMD 0805	8
C6	2.2uF	SMD 0805	1
D1	G_LED	SMD 0805	1
D2	R_LED	SMD 0805	1
L1	10uH	SMD 0805	DNP
R1, R4	10k	SMD 0805	2
R2, R3	0R	SMD 0805	2
R5, R8	4k7	SMD 0805	2
R6	1k	SMD 0805	1
R7	5.1k	SMD 0805	1
SWD1	Conn_01x04	Connector PinHeader 2.00mm	1
U1	AT24C256	DIP-8	1
Y1	8MHz(16pF)	Crystal SMD HC49	1
Y2	32.768KHz	Crystal SMD ABS25	DNP

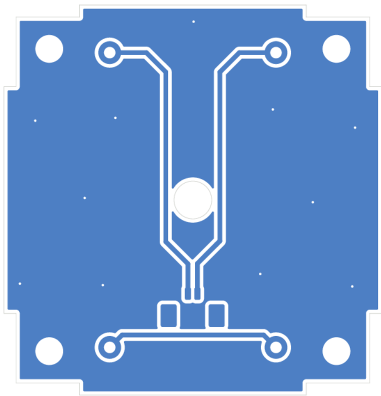
V Structure PCB design

PCB dimensions are 50×51.6mm in a scale of 1:1

V.1 Side (Bottom) TOP

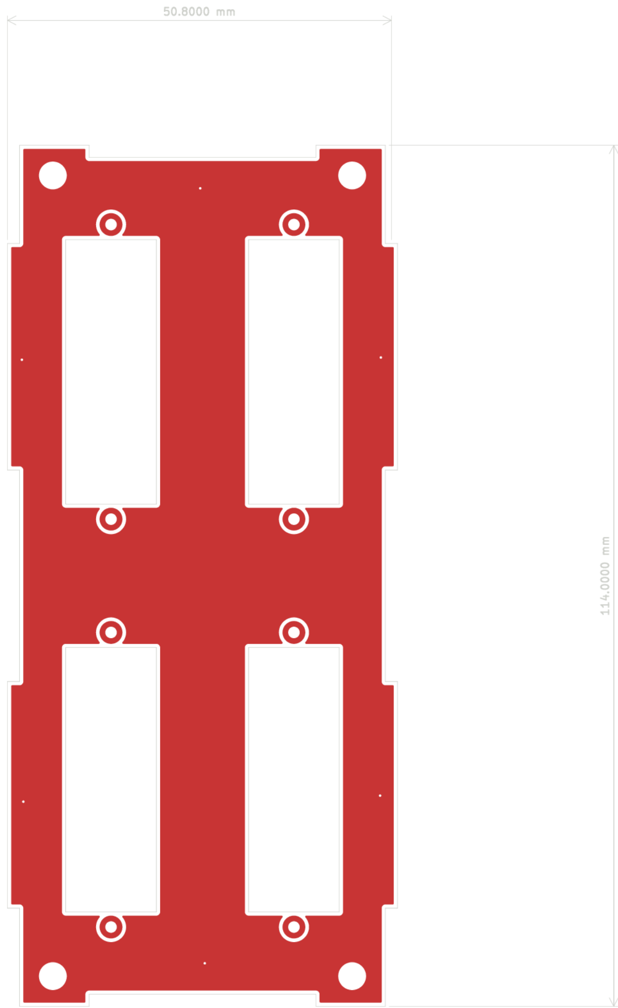


V.2 Side (Bottom) BOTTOM

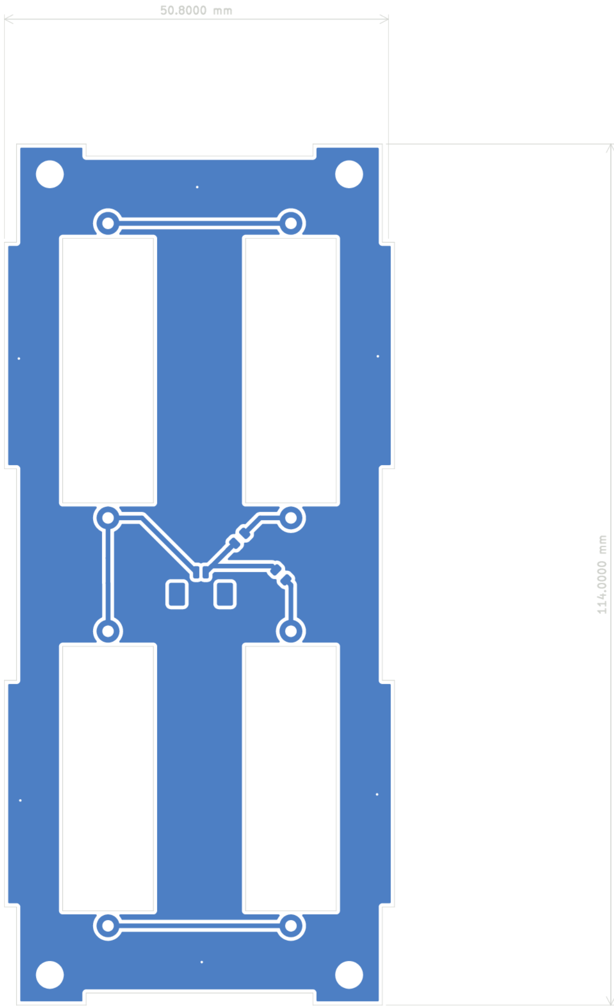


PCB dimensions are 50.8×114mm in a scale of 1:1

V.3 Side (Sides) TOP

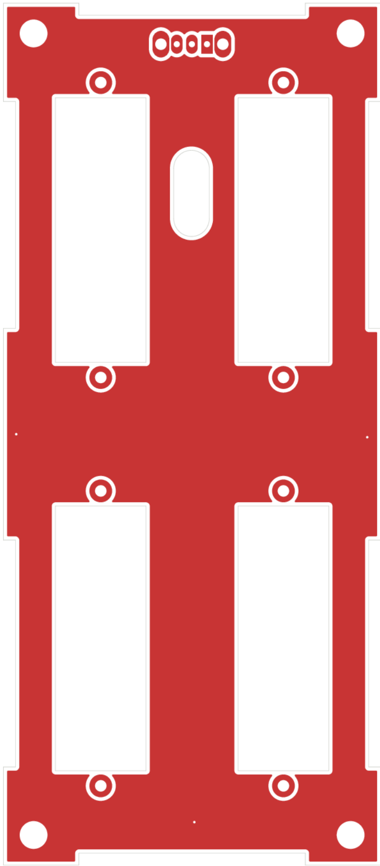


V.4 Side (Sides) BOTTOM

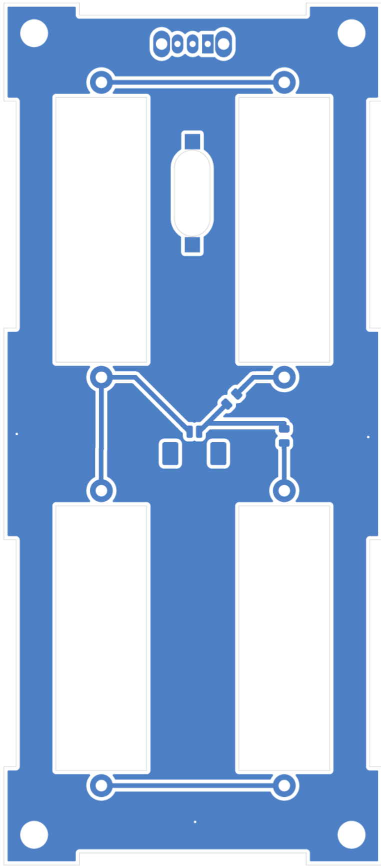


PCB dimensions are 50×114mm in a scale of 1:1

V.5 Side (Top) TOP

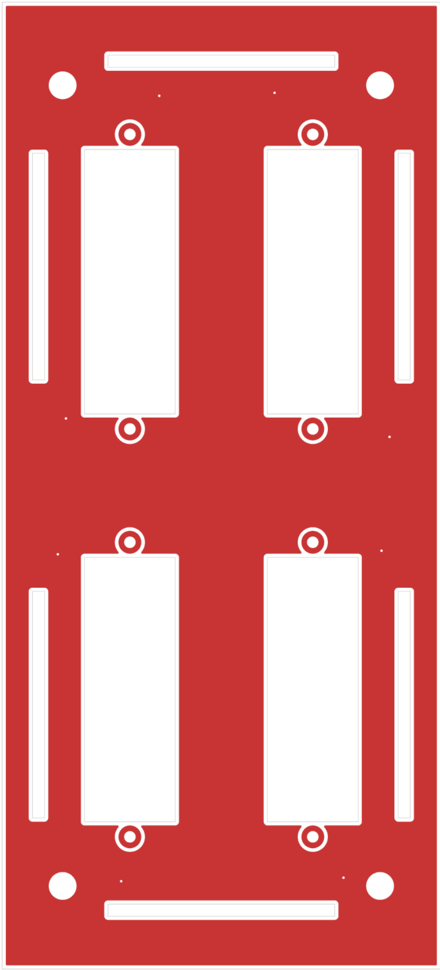


V.6 Side (Top) BOTTOM

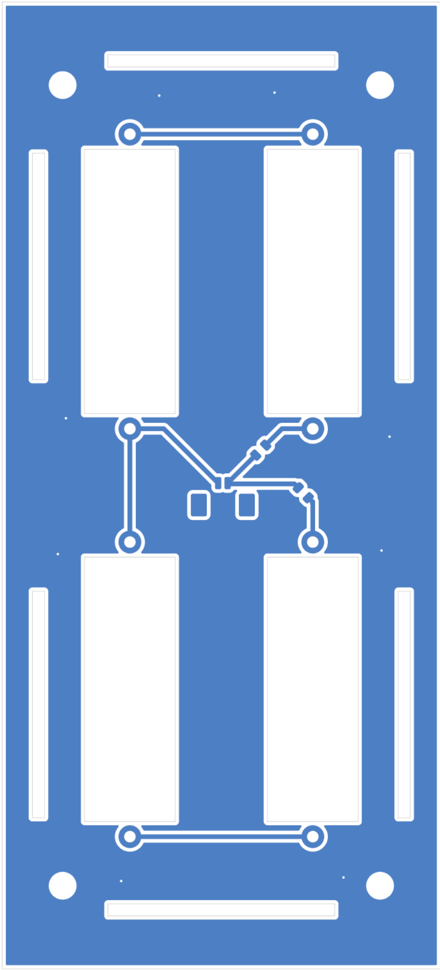


PCB dimensions are 58×128mm in a scale of 1:1

V.7 Side (Slider) TOP



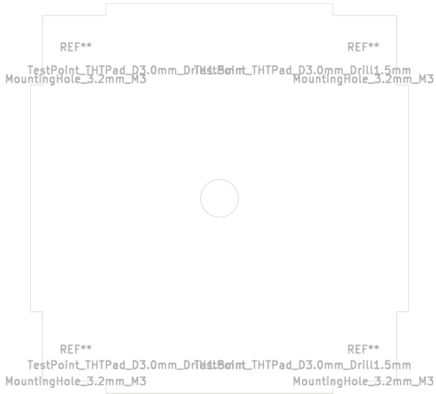
V.8 Side (Slider) BOTTOM



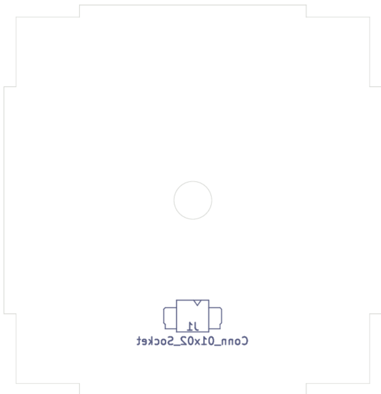
W Component placement - Structure

PCB dimensions are 50×51.6mm in a scale of 1:1

W.1 Side (Bottom) TOP

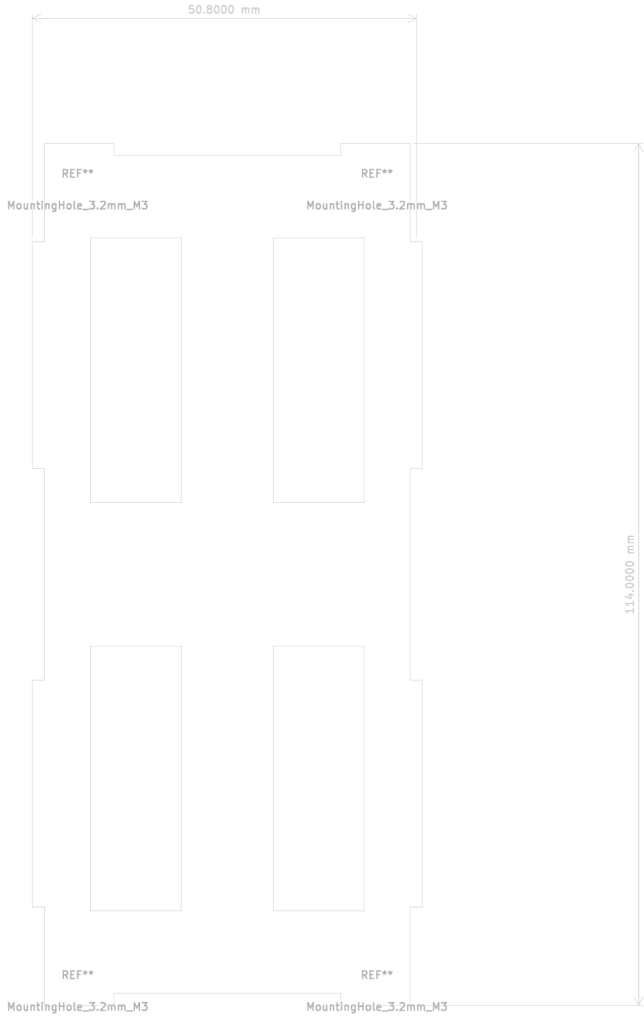


W.2 Side (Bottom) BOTTOM

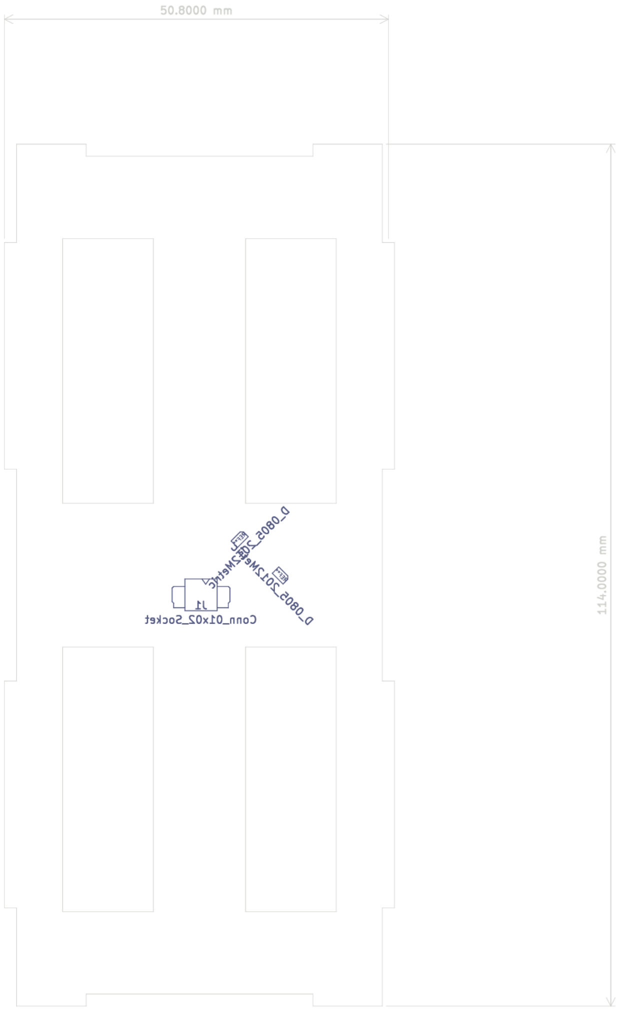


PCB dimensions are 50.8×114mm in a scale of 1:1

W.3 Side (Sides) TOP

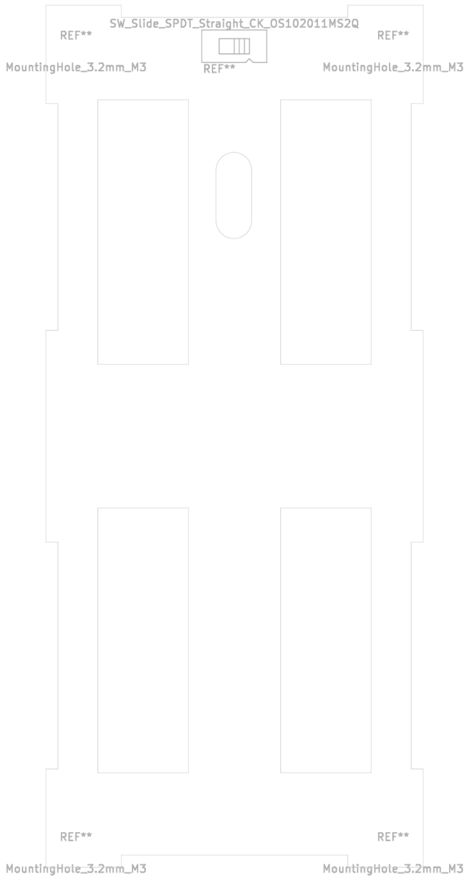


W.4 Side (Sides) BOTTOM

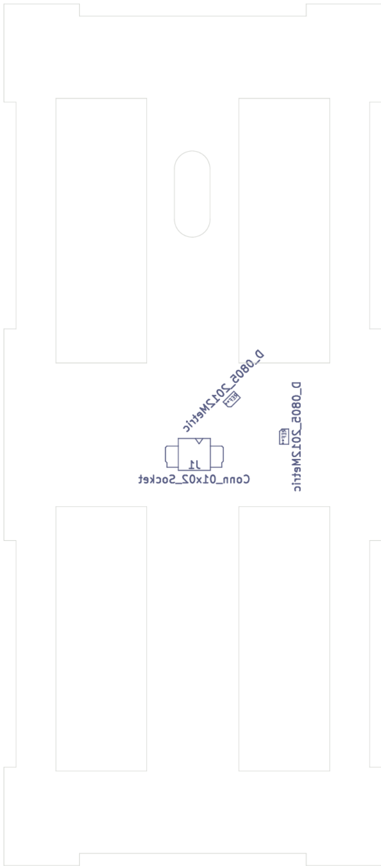


PCB dimensions are 50×114mm in a scale of 1:1

W.5 Side (Top) TOP

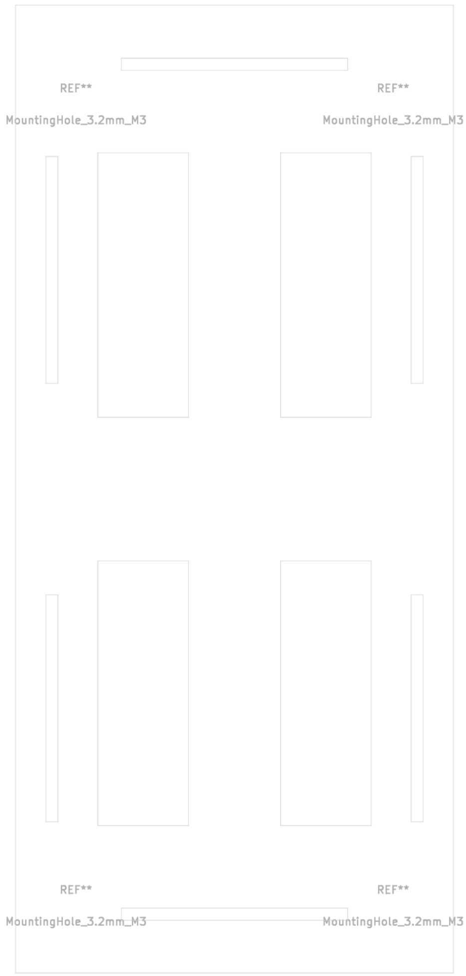


W.6 Side (Top) BOTTOM

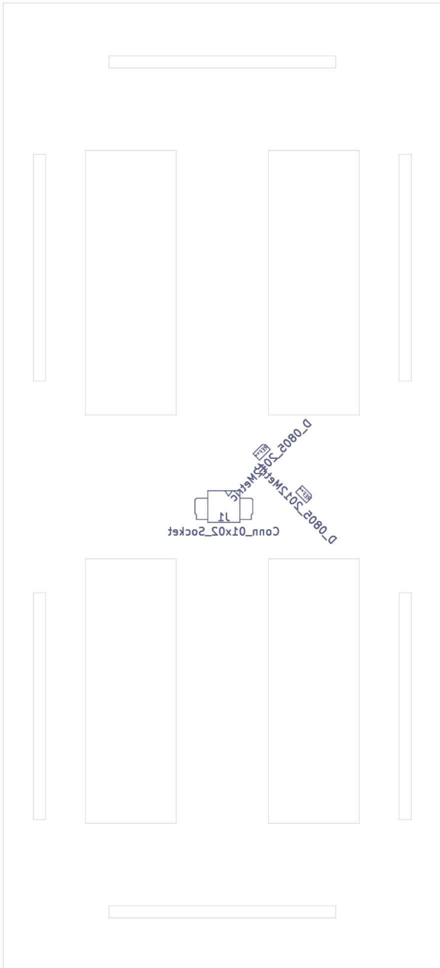


PCB dimensions are 58×128mm in a scale of 1:1

W.7 Side (Slider) TOP



W.8 Side (Slider) BOTTOM



X Content of the electronic attachment

Structure of the directory with electronic attachment

- `/` .... root of the attached archive
  - `PCB and structural design` .... PCB design of the satellite
    - `Electronics` .... KiCad PCB design of the modules
      - `BUS_Connection` .... System bus
      - `COM` .... COM module design
      - `EPS` .... EPS module design
      - `Ground station` .... Ground station design
      - `OBC` .... OBC module design
    - `Footprints` .... Custom footprints
      - `TP4056.pretty`
        - `TP4056_Charger.kicad_mod` .... TP4056 Li-Ion charging module
    - `PCB Templates` .... Templates for board design
      - `Ring_mod_template` .... Template for ring PCB design
      - `Template_inside` .... Template for classic PCB design
    - `Signature image` .... Folder with bitmap signature image
    - `Structural design` .... Structural PCB design
      - `bottom` .... Bottom part of the satellite
      - `sides` .... Sides of the satellite
      - `slider` .... Deployer slider
      - `top` .... Top side of the satellite
  - `Software` .... Folder with software and firmware
    - `App` .... Folder with control software
    - `Firmware` .... Satellite and Ground station firmware
      - `FW Ground Station` .... Ground station firmware (superloop)
      - `FW Ground Station - freeRTOS` .... Ground station firmware (freeRTOS)
      - `FW Satellite` .... Satellite firmware (superloop)
      - `FW Satellite` .... Satellite firmware (freeRTOS)