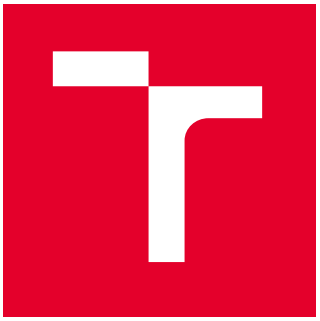

BRNO UNIVERSITY OF TECHNOLOGY

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

FAKULTA ELEKTROTECHNIKY
A KOMUNIKAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF RADIO ELECTRONICS

ÚSTAV RADIOELEKTRONIKY

APPLICABILITY OF LOW-COST PICOSATELLITES

VYUŽITELNOST NÍZKONÁKLADOVÝCH PIKOSATELITŮ

MASTER'S THESIS
DIPLOMOVÁ PRÁCE

AUTHOR — **Bc. Maroš Roman**
AUTOR PRÁCE

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

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. Maroš Roman ***ID:*** 222951

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

TITLE OF THESIS:

Applicability of Low-Cost Picosatellites

INSTRUCTION:

Conduct research on the applicability of picosatellites and femtosatellites for educational purposes, with a focus on cost-saving and weight minimization measures. Evaluate the feasibility of using high-altitude balloons as an alternative launch method for laboratory exercises. Based on the gathered information, design a block concept for prototypes and select appropriate components.

Create circuit diagrams that encompass all necessary systems (EPS, OBC, COM, payload). Fabricate the prototypes, conduct testing and measurements. Prepare a technical report and documentation. If feasible, carry out a test launch of the manufactured prototypes and evaluate the results.

RECOMMENDED LITERATURE:

[1] 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.

[2] BARNHART, David J., Tanya VLADIMIROVA, Adam M. BAKER a Martin N. SWEETING. A low-cost femtosatellite to enable distributed space missions. Acta Astronautica [online]. 2009, 64(11-12), 1123-1143 [cit. 2023-05-26]. ISSN 00945765. Dostupné z: doi:10.1016/j.actaastro.2009.01.025

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 thesis verifies the applicability of picosatellite or femtosatellite platforms in education. It describes the design of a femtosatellite platform, called Femtostrat-1, that can serve as an educational tool for university laboratory exercises. The platform is designed to contain all of the systems expected to be used on a real satellite platform. The options for usage of this platform are discussed, including multiple launch methods or different communication standards. Additionally, the thesis describes the design and implementation of a Femtostrat-1 flight model. Last, the stratospheric test flight of the Femtostrat-1 is described and its results are detailed.

KEYWORDS

femtosatellite, stratospheric launch, education, LoRa, LoRaWAN, system engineering

ABSTRAKT

Tato práce zkoumá využitelnost platforem picosatelitů nebo femtosatelitů ve vzdělávání. Popisuje se zde návrh femtosatelitové platformy jménem Femtostrat-1, která může sloužit jako vzdělávací nástroj pro univerzitní laboratorní cvičení. Platforma je navržena tak, aby obsahovala všechny systémy, které se očekávají na skutečné satelitní platformě. Jsou probírány možnosti využití této platformy, včetně různých metod vypuštění nebo různých komunikačních standardů. Navíc tato práce práce popisuje návrh a realizaci letového modelu Femtostrat-1. Na závěr je popsán stratosférický testovací let Femtostrat-1 a výsledky tohoto letu.

KLÍČOVÁ SLOVA

femtosatelity, stratosférický let, vzdělávání, LoRa, LoRaWAN, systémové inženýrství

Rozšířený Abstrakt

Úkolem této diplomové práce je ověřit použitelnost femtosatelitů a pikosatelitů v oblasti výuky. Na úvod práce jsou popsány platformy třídy picosat a femtosat. Pro třídu femtosat jsou navíc uvedeny různé uskutečněné mise. Podrobně je popsáno možné použití femtosatů v laboratorních cvičeních. Dále jsou poskytnuty různé příklady znalostí, které může tato platforma rozvíjet.

Dále je definována platforma jménem Femtostrat-1, která ověřuje praktickou využitelnost femtosat platformy ve vzdělávání. Cílem platformy navržené v této práci je poskytnout nízkonákladové řešení pro vzdělávání v oblasti inženýrství vesmírných systémů. Jsou popsány různé metody vypuštění. Dále je podrobně prozkoumán návrh platformy i samotný návrhový proces. Jsou popsány jednotlivé primární subsystémy platformy Femtostrat-1, například elektrický napájecí systém, palubní počítač, komunikační subsystém a užitečný náklad.

V práci je poskytnut podrobný přehled výroby, obvodů a firmware této platformy. Je vysvětlen engineering model, protyp letového modelu a jejich využití počas testováni platformy.

Práce je zakončena popisem letového modelu a jeho vypuštení. Výsledky testovacího letu jsou ukázány formou grafů a výpisu z letových záznamů. Popis letu je doplněn fotografiemi ze dne testovacího letu.

ROMAN, Maroš. Využitelnost nízkonákladových pikosatelitů. Brno, 2024. Dostupné také z: https://www.vut.cz/studenti/zav-prace/detail/159866. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav radioelektroniky. Vedoucí práce Aleš Povalač.

Author's Declaration

Author:	Bc. Maroš Roman
Author's ID:	222951
Paper type:	Master's Thesis
Academic year:	2023/24
Topic:	Applicability of Low-Cost Picosatellites

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 advisor of my thesis, doc. Ing. Aleš Povalač, Ph.D. for his valuable suggestions and advice throughout the making of this thesis. Furthermore, I would like to thank the crew of the SHMÚ meteorological station in Gánovce, specifically Ing. Jozef Depta, who provided access to the meteorological balloon and helped organise the launch of my femtosatellite. I would also like to thank Marián Lukačko, who found my prototype after landing and provided camera footage, voltage testing after landing as well as sending the prototype back to Brno. Additionally, I would like to thank my father Ing. Igor Roman for providing contacts for the launch site, transport, and help with the launch. Last, I would like to thank my wife Eliška and family for supporting me throughout making of this work.

Contents

List of Figures

List of Tables

Introduction

In the early 21st century, the introduction of the first cubesat-class satellites significantly increased the availability of space-based research for commercial entities and educational institutions.

As is widely known, the main limit for space exploration is often the prohibitively large costs associated with designing, creating, and launching space vehicles. Although the average cost of a cubesat, according to [1], is 7.5 to 7 million USD for a dedicated launch and as low as 90,000 USD for a rideshare launch of a 1U cubesat, such a price is still unfeasible for smaller or less funded institutions.

This thesis focuses on conducting research into the feasibility of using picosatellites or femtosatellites in education. A brief overview of picosat and femtosat platforms is provided. The applicability of femtosats in laboratory exercises is detailed, with examples of specific knowledge provided by them.

In further chapters, the thesis describes the proposed femtosatellite platform called Femtostrat-1, outlining its objectives and general operation. The purpose of the platform drafted in this thesis is to verify the applicability of femtosatellites in education and provide a low-cost solution to space systems engineering education. The launch method trade-off analysis and final choice is also outlined. Next, the platform design as well as the design process itself is explored in detail. Each of the primary subsystems of the Femtostrat-1 platform is described. This includes the Electric Power System (EPS), the On-Board Computer (OBC), the communication subsystem (COM), and the payload.

In the next chapter, the implementation of Femtostrat-1 is described. The circuit board design and schematics are shown. Next, the engineering model and the proto-flight model are detailed. The testing process that utilised them is also explained. The fabrication process of the proto-flight model and flight model is described together with the dimensions and properties of the final boards. Afterwards, the ground station and concept of operations is described.

The last chapter is focused on the test flight of Femtostrat-1. It describes the events and actions that occurred during the preflight, flight, and postflight phases. Next, the results of the test flight are shown with associated figures and listings. The last part is concerned with issues that occurred before or during the flight.

1 Satellite Classification

This chapter defines the properties of micro- and femto- class satellites in education to demonstrate their significance and possible usage.

Standard mainline satellite missions performed by large organisations, such as NASA or ESA, and commercial entities, such as SpaceX, generally fall into classes of satellites that are 'small' or larger, that is, their mass is above 600 kg.

Such satellites are likely unattainable with a university budget. With the rise of commercial space launches and dedicated deployers, satellites of micro-, pico-, and lower classifications have become popular. These satellites have masses of 200 kg and lower, with most of them not reaching a mass higher than 10 kg. A typical example of a university satellite is a 1U cubesat, which is described in section 1.1.

However, as launch and technology capabilities increased, the democratisation of space allowed for an even greater decrease in the mass requirements. Described in section 1.2, satellites of the femto- class have begun appearing in various university proposals and have even been used in space missions. These satellites, which have a mass under 100 g, offer possibilites for even lower budget missions.

1.1 Cubesats

The term 'Cubesat' was introduced as part of a collaboration between professor of the California polytechnic university and the university of Stanford. The intention was to reduce both the development time and cost of picosatellite platforms. [1] This effort eventually resulted in a great increase of spaceflights, especially in the commercial and educational sectors. The cubesat standard, for example, enabled the creation of rideshare programmes supported by several transport providers, reducing the cost of satellite projects even further.

A cubesat is defined as a "class of satellites that adopt a standard size and form factor, which unit is defined as 'U'". [1]

A single unit is defined as a 10 cm by 10 cm by 10 cm cube with a maximum mass of 2 kg. [1] In practice, this mass is usually enforced through the use of ballasts[1], if necessary. The upper limit of the units a single cubesat can have does not exist. However, it should be noted that most launch providers and rideshare programmes limit the units, with the upper limit being 16U as of writing this thesis (see appendix A.1).[2]

Most cubesats follow the same verified design patterns. A typical cubesat consists of an aluminum frame with outside facing rails, facilitating safe deployment from

[1]Dead weight used for the adjustment of the centre of gravity (CG) of a spacecraft.

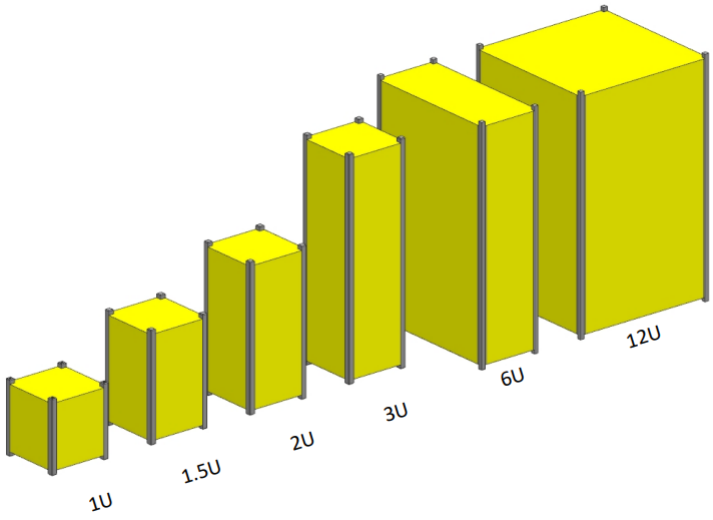

Fig. 1.1: Standard cubesat sizes - Reprinted from CubeSat Design Specification[1]

a cubesat deployment rack. On the inside of this frame, circuit boards are placed, with each board usually representing a single system within the cubesat.

The boards are typically stacked vertically, connected by a PC/104 standard bus. The outer shell of the cubesat is commonly covered with solar panelling, with some sides being reserved for a payload, for example, a camera.

Cubesats of today are almost exclusively launched by deployers, with some exceptions, such as being launched by hand by astronauts. [3] The deployers, similarly to cubesats, follow a standard to facilitate the safe deployment. Cubesats can either utilise a rail design or a tab design with a constraint mechanism. In both cases, a spring pushes the cubesats out of the deployer.

The launch costs of a cubesat are highly variable, with different launch providers offering different costs, as well as rideshare programmes further decreasing the associated costs. For example, a company called Nanoracks offers flights to the International Space Station (ISS) for as low as 90,000 USD per 1U. [4] Another opportunity, especially for educational applications of cubesats, are the several programmes that allow for a free launch of a cubesat if a competion is won, such as the 'Fly your satellite' programme by the ESA. [5]

1.2 Femtosats

In contrast to cubesats, femtosats lack a standardized framework. The definition of a femtosatellite is broad, encompassing any satellite with a mass below 100 g. [6] As such, there are no constraints on the shape of femtosats.

One suggested standard is embodied in the SunCube which advocates for a unit denoted as 1F. This unit comprises a 3 cm by 3 cm by 3 cm satellite with a maximum mass limit of 35 g. [7]

A more prevalent form of femtosats are single board femtosats, where all essential subsystems are placed onto a single silicon board. An example of such a satellite is depicted in figure 1.2.

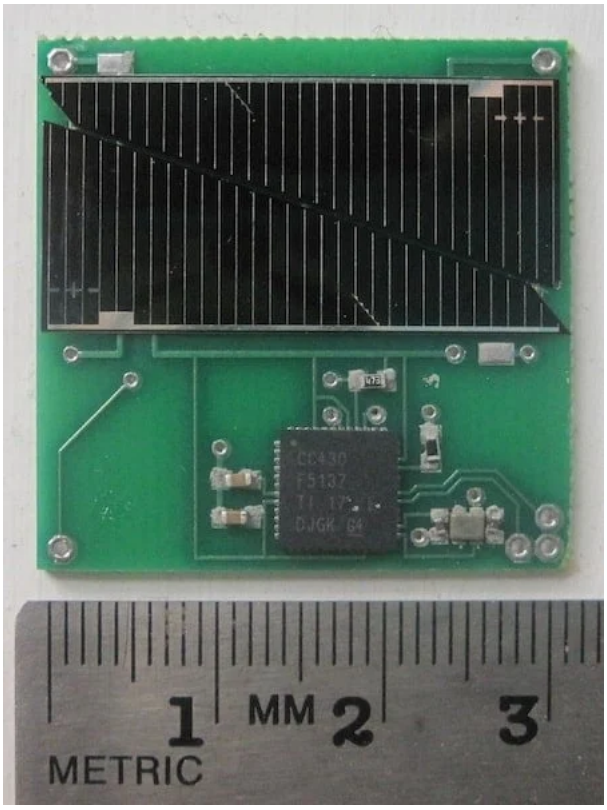

Fig. 1.2: An example of a femtosatellite. - credit: Cornell University

The general concept of femtosats is that because to their low size and mass profile, swarms of these can be launched at once and the main tasks can be distributed between each member of the swarm. Thanks to such behaviour, loss of a single satellite does not affect the mission, because the rest of the swarm takes control of its processes. [8]

According to Mason Peck [9], the very small envelope of femtosats makes it possible to reach distant positions. That is because of their similarity to space dust. Such 'particles' are affected by solar winds and can 'sail' without fuel, similar to solar sail satellites.

Several femtosats have successfully reached space through various missions. The initial venture into femtosat technology involved a collaboration between Cornell Uni-

versity and NASA. Within this mission, three femtosatellites denoted as 'Sprites' were equipped with instrumentation capable of collecting data on radiation, chemistry, and particle impacts. Instead of being launched directly into orbit, these satellites were affixed to the MISSE-8 external platform on the ISS. [9]

Cornell University also conducted a large-scale test of femtosats as part of the Kicksat and Kicksat-2 missions. Although, the former experienced an unsuccessful deployment due to a technical glitch, Kicksat-2 attained success by deploying 105 'Sprite' femtosats. These miniature satellites were essentially circuit boards fitted with sensors and a short-range transmitter capable of relaying telemetry data. The inclusion of solar cells allowed them to generate their own power. [10] [11]

Operating in the 400 MHz range with a power output measured in milliwatts, these femtosats transmitted signals that were received by multiple ground stations. Their functionality was verified successfully during their brief three-day lifespan. After that, they were destroyed due to high heat when they re-entered Earth's atmosphere. This marked a significant development and application of femtosatellite technology. [11]

2 Femtosatellites in Education

With a significantly smaller size compared to traditional satellites, including cubesats, the applications of femtosatellites might appear limited. However, their small envelope makes them substantially more accessible. This means that femtosats might be an effective method of educating space systems engineers. It allows for practical knowledge to be given to university or high school students about various areas of satellite design, such as power systems, long range communication, and others.

Compared to the costs associated with launching cubesats, typically involving costs up to hundreds of thousands of euros and years of preparation, femtosats can give a 'hands-on' experience with tangible results. Combined with a stratospheric launch instead of a space launch, a full mission involving such a satellite can be fulfilled within the span of a single term, meaning such an endeavour can be used as a basis for a single or two term laboratory exercise.

Despite the significant cost reductions, femtosatellites remain a source of teaching all concepts of system engineering, as well as development of various space subsystems. In addition, stratospheric launches provide significant practical experience in the area of ground operations and radio communication.

Although the size of the satellite limits the payloads that can be included, there are several possible meaningful experiments in both the upper atmosphere and low Earth orbits.

As an example, the Arizona State University (ASU) developed 3F sun-cube standard femtosats that cost only up to 1,000 USD for a launch to the ISS. Jekan Thanga of the ASU explains that in the future, such satellites have the potential to be sold in public markets and used for STEM education, as well as hands-on integration and testing experience for students. [12]

In addition to lowering the entrance cost for hands-on spacecraft creation, complete research missions have been proposed for femtosat platforms. For instance, Himangshu et al. proposed a mission utilising up to hundreds of femtosats for location tracking, secure communication, or Earth imaging. Moreover, the proposed mission intends to provide inter-satellite laser communication and swarm configuration. [13]

2.1 Femtosats for Laboratory Exercises

The specific use for the femtosat proposed in this thesis is the implementation of it as part of a laboratory exercise. The femtosat platform has several advantages when used for education.

First, the platform is very small, the proposed Femtostrat-1, described in later chapters, is only 7 cm by 7 cm, with practically no depth. This allows for very easy integration with other systems. This is important because it gives the opportunity of launching the femtosatelite both individually, which might be more expensive, as well as part of a larger payload. Its size allows it to be part of a cubesat platform, a meteorological probe, and, with some modifications to the footprint, it might also be integrated into a model rocket as part of a cansat mission. [14]

Another advantage comes with its lower costs. Platforms which are planned to be used regularly, that is, once a year or once a term, are expected to have reduced costs to remain feasible. The smaller size and a lack of complex systems of femtosats means that the pricing of the system is very low compared to similar educational platforms, such as KitSat, that comes with prices of over 2,000 EUR for its base model with a stratospheric flight kit. [15] The lack of complexity also allows for fabrication and assembly to occur within the time-frame of a single term.

The last advantage comes with its simplicity that allows for extension and full understanding of the whole system within one term. Essentially, the only necessary requirements for using the platform are a small budget, a latex balloon, and an amount of helium or hydrogen.

There are many concepts students can learn by working on such a project. One such concept is EPS and power budgets that come from the need to explore the topic of power consumption and management. The long distance communication would necessitate deep understanding of COM systems and antenna theory. To be able to understand and modify the firmware, it is necesarry to learn advanced software engineering concepts, such as real-time operating systems, low power modes, watchdog timers, and others.

In conclusion, the femtosatellite platform allows for fast and affordable practical examples in system engineering to be performed within the span of a single term.

3 Proposed Femtosatellite

Collated data on femto- and pico- satellites was used to propose a femtosatellite platform that fulfills the objectives of this thesis. This platform had two primary objectives. The first objective requires the satellite to serve as an educational tool for future space engineering students. The second primary objective aims for minimizing the weight and cost of the mission.

The secondary objective is to provide a platform that is feasible to launch every school year.

3.1 Launch Method

Several options were consided to launch the femtosatellite. Although, the launching of the femtosatellite model into outer space was a possible option, there were several factors which made this unfeasible. First, the time schedules required for a space launch are incompatible with the timeline of creating this thesis. A typical design, testing, and integration process for a space-grade component is one year at the lowest. Second, the cost requirements of such a mission are in thousands of euros. Even if such a mission was funded, the primary goal of targeting the resulting femtosatellite at a laboratory exercise would not be fulfilled. Last, the COTS components required for the femtosatellite would either be prohibitively expensive or would not pass acceptance tests.

The second considered option was an air launch, similar to launches provided for the cansat competition. Such a launch is implemented with a general aviation aircraft. The aircraft is capable of lifting the femtosatellite to heights of several thousand meters at most. Although, this method is available and can be listed as a possibility for the laboratory exercise, the maximum height made it less suitable.

The last, and chosen, option is a startospheric launch. A stratospheric launch has several advantages. The maximum height of a stratospheric launch is approximately 40 km. Such heights partially simulate the conditions of outer space. A very low pressure is encountered at this height. Additionally, measurable levels of radiation can be detected at this altitude.

The entry cost of a stratospheric launch is roughly 1,000 CZK, divided between 500 CZK for a latex balloon rated for tens of grams, lifted up to about 10 km of height, [16] and approximately 500 CZK for the necessary helium. The need for helium instead of hydrogen is due to regulations imposed on hydrogen, as it is a highly flammable substance not commonly sold in larger quantities to individuals. However, when bought for an institution, these regulations might be lifted. A large

additional cost, both in time and money, is the need for requesting an airspace corridor for the launch.

On the other hand, a long term partnership with a meteorological institute is a significant possibility for cheap launches. A typical meteorological station might launch up to two meteorological high-altitude sondes per day. A balloon required for such a sonde has a margin of tens of grams. An airspace corridor is already established for this launch. A femtosatellite with a weight in tens of grams is considered negligible for such a launch. As a result, a launch might be organised with the institute each year with no associated costs.

3.2 Proposed Femtosatellite

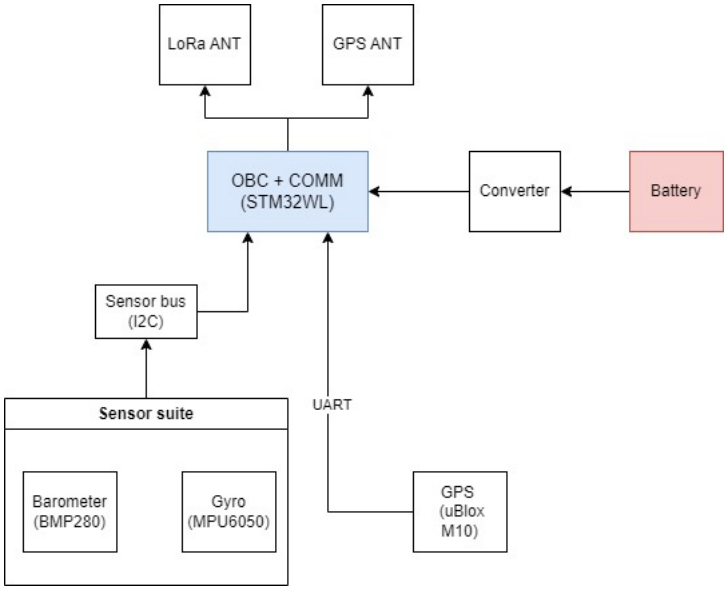

Fig. 3.1: Proposed femtosat block diagram.

To fulfill key mission objectives and perform within the constraints of a stratospheric launch, a specific platform was proposed. The platform, and subsequently the mission name, was chosen to be Stratospheric Femtosatellite 1, or Femtostrat-1.

The proposed femtosatellite is described by a block diagram in figure 3.1. The chosen architecture corresponds to the set objectives. To provide educational value to space engineering students, the Femtostrat-1 should contain all primary components of a typical satellite. For that purpose, the central part of Femtostrat-1 is formed by the On-Board Computer (OBC) module, together with the Communication module (COM). This forms the central pillar of data output, command input, and processing. To provide power, an Electric Power System (EPS) is included. For the payload,

three, simple to implement, modules were chosen. It includes a gyrometer for general movement analysis, a barometric module for temperature verification and altitude calculation, and a GNSS module for position tracking. All these sensors were considered capable of providing meaningful data for education on data handling and processing.

One of the considered parts of the architecture was the inclusion of a modular payload slot for custom sensor or device development during a laboratory exercise. The inclusion of this slot might be considered for future work.

In addition to its usage as an educational tool, the proposed satellite is also expected to be as low-cost as possible. Therefore, another requirement is that the satellite shall demonstrate the use of low-cost COTS components. In practice, this means that, during the selection process of individual components, a large portion of the selection was based on weight and dimensions. When a larger selection was available, the component with the lowest cost was chosen.

The basic mission profile of the platform is to reach a height of several thousand meters and be capable of four vital processes: power generation with solar cells, sensor data generation, command reception, and data transmission.

4 Femtosatellite Design

In this chapter, the design of the femtosatellite is described in detail. Additionally, the process of designing is explained for further reference.

4.1 Design Procedure

For the preliminary design phase, a draft design was formed based on initial block diagrams and mission requirements. The main drivers of this phase were component availability, size, price, and mission requirements. After the draft design was formed, an initial component list was created. In addition to components on the list, THT or Breakout versions of the required components were procured. These were integrated into the engineering model (EM).

The engineering model was used to test component function and to create and test firmware for the OBC.

After detailed schematics were created, additional components were incorporated into the design, such as power control MOSFETs for switching payloads, latchups, and debugging headers.

Moreover, actual measurements of the components revealed issues with the preliminary design. In such cases, the components were replaced with appropriate alternatives. Such processes are explained in further sections.

The design also changed based on the parameters of the launch. Because of a possibility of both individual and 'rideshare' launches, several alternative designs needed to be prepared. One such case was the possibility of the satellite not receiving sunlight when integrated into a larger structure.

Time constraints were also a major factor in the design process. Long order times caused the design-testing loop to be longer, requiring less trials and revisions before arriving at the final design.

4.2 Power System

For the power system, three alternatives were considered, one of which was chosen and fabricated.

The initial form of the mission incorporated solar panels as a simulation of a typical satellite EPS, where power is charged and discharged based on usage. Two alternatives were considered for this design. The 1-B model would use solar cells as a power source with only a high capacitance capacitor used as a kind of battery. This model would maximise weight reductions, because the battery is the heaviest

component on the satellite. The 1-C model would utilise solar cells and a regular battery to maximise power storage and output at the cost of higher weight.

As explained in section 4.1, because of the possibility of not receiving sunlight, a design using only battery power as the main power source was created. This design was marked as Femtostrat1-A and was chosen as the flight configuration.

The Femtostrat1-A model power system consists of three major components. The first component is the battery. This is the main power source for all of the systems of the satellite. The second is the voltage converter circuit to convert the battery voltage. The last is the power control circuit that provides the capability of controlling the power input of the payload.

4.2.1 Battery

Initially, the main battery to be used was a CR2032 coin battery. However, as can be seen in figure 4.1, the internal resistance of the battery at the maximum power input of the LoRa module (110 mA), would cause the voltage of the battery to drop significantly, causing a brown-out.

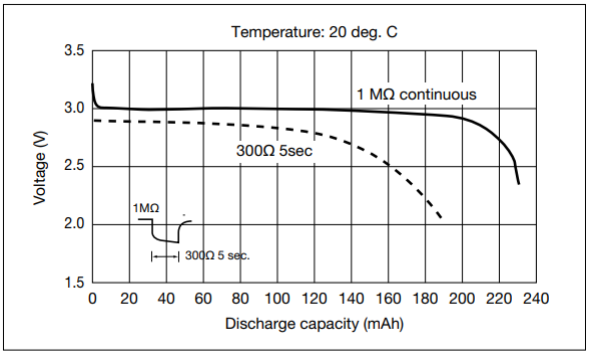

Fig. 4.1: CR2032 Pulse discharge characteristics - reprinted from cr2032 datasheet

As a result, the design was revised to use an AAA format battery with a step-up voltage booster.

The chosen battery was the L92 Energizer AAA battery. The nominal voltage of the model is 1.5 Volts. Compared to the CR2032 battery model, the L92 has a significantly better internal resistance profile (see figure 4.2.). [17]

Additionally, the L92 battery has several other desirable traits. First, the model has a better pulse response (see appendix B.3.) and an operating temperature range of -40 to +60 °C, with a low temperature impact on capacity at the current drain requirements of Femtostrat-1 (see appendix B.4.). [17]

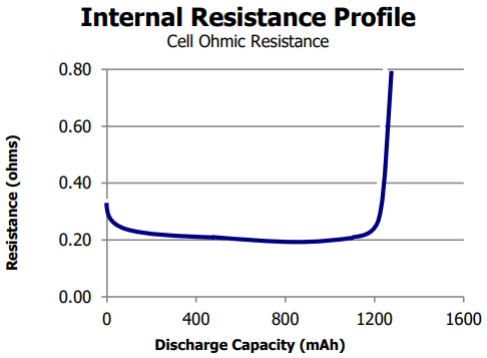

Fig. 4.2: L92 Internal resistance profile - reprinted from L92 datasheet

4.2.2 Voltage Converter Circuit

The OBC of the satellite requires a steady 3.3 V input to function nominally. However, the chosen battery is only capable of outputting 1.5 V in nominal operation.

Because of weight constraints, it was not acceptable to use two of such batteries in series as a solution. Therefore, a voltage booster was necessary to convert the voltage to an acceptable level.

For this purpose, the MCP1624 step-up DC-DC converter was used. It is a regulator created to specifically act as a power supply solution for microcontrollers. Among the main advantages of the regulator is a low inrush current, allowing it to start at voltages as low as 0.65 V. It has a nominal efficiency of 96%. However, based on figure 4.3, at the expected input voltage of 1.5 V and maximum current requirement of approximately 111 mA, the efficiency is expected to drop to approximately 80%. [18]

In stand-by mode, its consumption is only 19 µA [18], limiting its impact on the power budget of the satellite.

It is available as a SOT-23 package for easy integration. It is suitable for stratospheric applications, with an operating temperature range of -40 °C to +85 °C.

The circuit is integrated to Femtostrat-1 with additional components into a power supply module.

4.2.3 Power Control Circuit

The last subsystem of the power module of Femtostrat-1 is the payload power control circuit. This simple circuit was designed to utilise two MOSFETs as switches for the GPS module, the gyroscope, and barometer payloads.

The MOSFET chosen for switching the power input for the payloads was the IRLML6402.

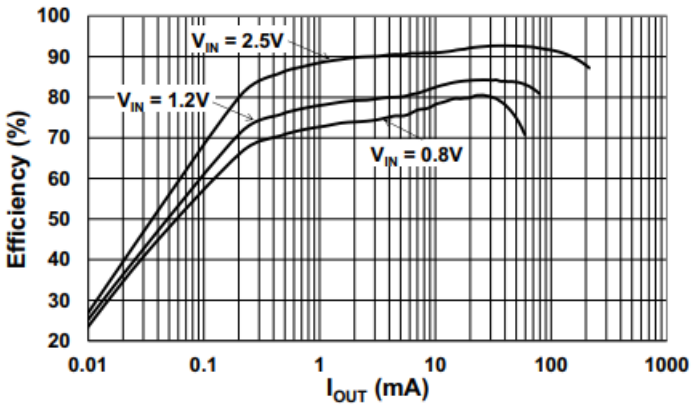

Fig. 4.3: MCP1624 efficiency versus I_{out} at $V_{out} = 3.3V$ - Reprinted from MCP1624 Datasheet

It has a very small footprint and low profile, available as a SOT-23 package. It has a junction temperature range of -55 °C to +150 °C, making it viable for stratospheric use without any additional protection.

It is rated for a maximum of 12 V gate-to-source voltage and -20 V of drain-source voltage.

There is a separate MOSFET for the sensor payload and the GPS payload for optimal use of the power budget.

4.2.4 1-B and 1-C Power Systems

As mentioned at the introduction of this chapter, additional two power system architectures were considered.

The Femtostrat-1B was a variant of the Femtostrat-1 platform that incorporated the maximisation of weight reduction measures. A theory was formed that, should a capacitor of sufficiently high capacitance be used, the entire platform could be powered through solar cells alone, with the gaps in power generation caused by rotation being regulated by the capacitor. It should be noted that this design method would be highly susceptible to harsh weather and time of day. Any operation in permanent shadow or at night would be unfeasible.

The Femtostrat-1C was a variant considered as an optimal platform, with less weight reductions to facilitate full EPS functionality, similar to that of a real satellite. The model would utilise several solar cells together with a battery. To support power generation, storage, and safe distribution, a battery management circuit was added to the design.

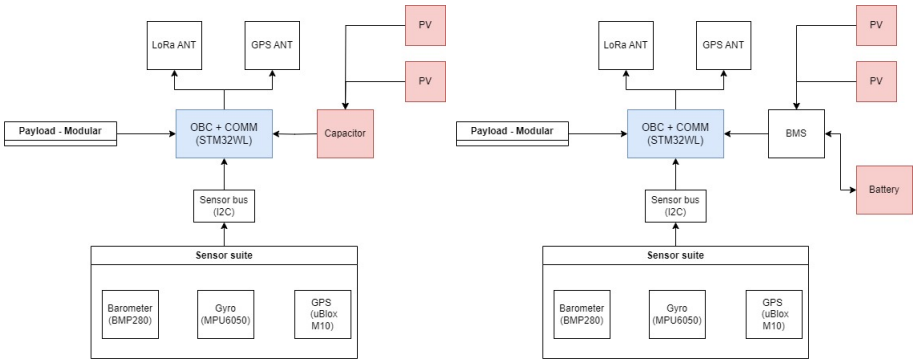

Fig. 4.4: Proposed Femtostrat-1B and Femtostrat-1C block diagram.

4.3 OBC and COM

The central component of the Femtostrat-1 prototype is the combined OBC and COM module. The chosen module is the Wio E5 Module by Seeed Studio. The module is a combination of the STM32WLE5JC chip and the SX126X embedded high-performance LoRa chip. These chips are described in the following sections.

4.3.1 Selection of OBC

As part of the trade-off analysis for the selection of the OBC and COM module, several components were considered.

Among the alternatives was the NRF24L01 single chip transceiver. This chip offered combined OBC and COM functionality and very low power requirements. Its RX and TX power requirement is as low as 13.5 mA and 11.3 mA respectively. The main disadvantage of this package is that it is a 2.4 GHz ISM band device with very short range which would make it unsuitable for use beyond the premises of a laboratory. [19]

Second, a separate transceiver and OBC setup was considered. A STM32-family microcontroller would be used, together with the SX1278 LoRa transceiver. The transceiver itself has a similar cost to an STM32WL family of chips. It offers low power consumption of 100 mW at +20 dBm. However, it is not capable of fulfilling OBC functions. Because of this, an additional component would be required. Moreover, multiple discrete components, such as capacitors and resistors, might have been necessary. [20]

Because of the shortcomings of components mentioned previously, the Wio-E5, which combines the functionality of a separate LoRa transceiver and a microcontroller in simplified and compact packaging, was chosen as the preferred variant for the prototype.

4.3.2 Selection of COM

To maximise the output of the prototype throughout the flight, a trade-off analysis was performed. For communication, a wide range of methods and protocols are available. Among these, the most suitable methods were 433 MHz or 868 MHz APRS radios, custom LoRa communication, and the use of the public LoRaWAN network.

When utilising a custom radio implementation, typically the only point of contact with the satellite is a ground station. However, there are notable exceptions. For radio on 433 MHz and 868 MHz frequency range, the TinyGS network of ground stations is available to supplement the custom ground station.

Similarly, the LoRaWAN network offers a large number of public gateways, automatically forwarding received data to a database. However, a typical LoRa application outside the network would only offer the use of self-made ground stations.

The result of the trade-off analysis was to use the LoRaWAN network, together with a LoRaWAN gateway as the main ground station. The chosen band for the communication system was the 868 MHz license-free European band. The choice was primarily based on the band in which the chosen gateway hardware for the Femtostrat-1 mission operated. However, the 433 MHz band was also suitable.

This was done for several reasons. First, the LoRaWAN network has much higher density of ground stations in the Czech and Slovak Republic where the launch was expected to occur. This would allow for using public LoRaWAN gateways for communication after the contact with the main gateway is lost. Second, the implementation of a complete system utilising this communication method is simple. Pre-made solutions for both the gateway and the communication module are available and generally low-cost. Last, the chosen OBC module was already made with LoRaWAN protocol support, making the integration even simpler.

4.3.3 Overview of OBC Module

As described in the previous chapter, the Wio-E5 LoRaWAN module is an 'ultra-low-power', low-cost, and low-size device. In Wake-up On Receive (WOR) mode, its current consumption can be as low as 2.1 µA. It has Embedded LoRaWAN protocol support as well as an AT command suite for fast prototyping. The package size of the module is 12 mm by 12 mm by 2.5 mm. It is a 28 pin SMT package for easier soldering. [21]

In addition to the SX126X transceiver, two crystal oscillators are included. One High-Speed External (HSE) 32 MHz and the second Low-Speed External (LSE) 32.768 kHz. [21]

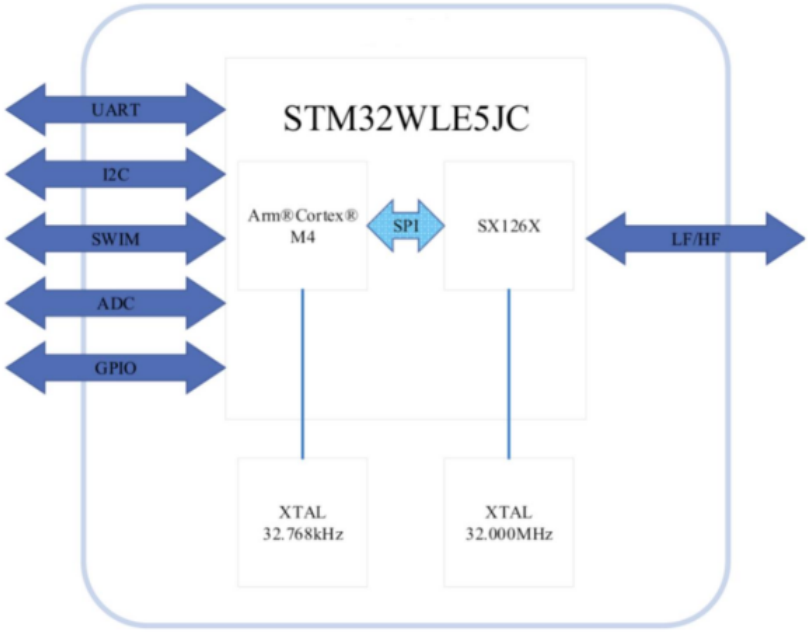

Fig. 4.5: Wio-E5 block diagram - Reprinted from Wio-E5 Datasheet[21]

The STM32WLE5JC[1] is described as a long-range wireless ultra-low-power device. Integrated within is a low-power compliant Low Power Wide Area Networking (LPWAN) standard radio compatible with LoRa, FSK, MSK, and BPSK keying methods. The central processing unit of the module is the ARM Cortex M4 32-bit RISC core. [22] The module has a supported frequency range of 150 MHz to 960 MHz. For the chosen LoRa standard, it has a RX sensitivity of up to -148 dBm. Transmitter high output power might be programmable up to +22 dBm. It is compliant with a wide range of frequency regulations across the world. [23]

Similarly to other components on Femtostrat-1, the operating temperature of the module is -40 °C to +85 °C. By itself, that is, without the rest of Wio-E5, its active-mode RX current drain is 4.82 mA. At 20 dBm active-mode LoRa TX, its current drain is 82 mA. For ultra-low-power applications, it might be set into stand-by mode which reaches currents as low as 360 nA at V_{DD} input of 3 V. [23]

The module specifically used in Wio-E5 has a 256 kB Flash memory and 64 kB RAM memory, leaving significant amounts of space for complex programs. [23]

[1]Later written as 'the module'.

4.3.4 Overview of the COM Module

The SX1268[2] is a sub-GHz radio transceiver made specifically for long range wireless communication. It has good power characteristics, with only 4.2 mA active receive mode current and 107 mA transmit mode current at +22 dBm[3]. It has an operating temperature range of -40 °C to +85 °C and operating voltage range of 1.8 V to 3.7 V. [24]

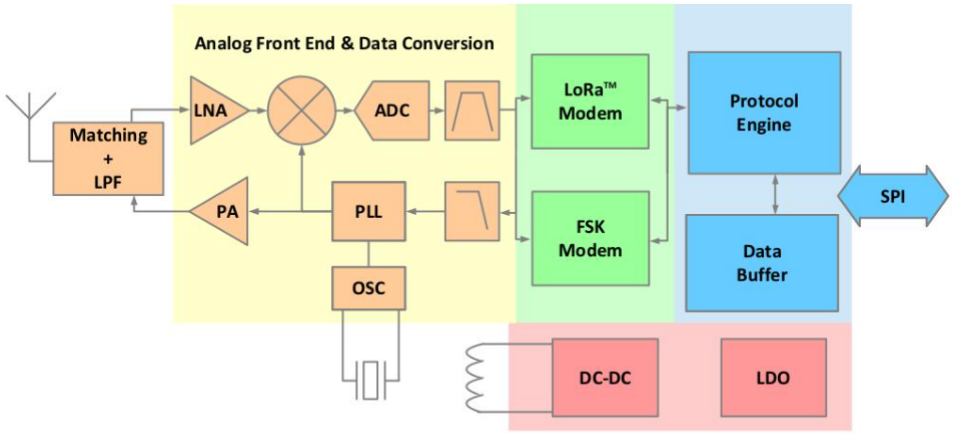

Fig. 4.6: SX1268 block diagram - Reprinted from SX1268 Datasheet[24]

The SX1268 transceiver is specifically designed for LoRa modulation in LPWAN use-cases. It is compatible with the LoRaWAN standard and complies with its physical layer requirements.

4.3.5 LoRaWAN Network

The Long Range Wide Area Network (LoRaWAN) is specified as a Low Power, Wide Area networking protocol which is designed to wirelessly connect devices to the internet. It is designed to fulfill IoT requirements, such as bi-directional communication, end-to-end security, mobility, and localisation services. [25]

The LoRaWAN network has a star-of-stars topology, where gateways (ground stations) are used as relays between the central network server and end devices, such as Femtostrat-1. The gateways are connected to the central network through a typical internet connection. [25]

End-point devices in the LoRaWAN network can have three different classes. Class A is the default class which must be supported by any end-point device within

[2]The Wio-E5 Manufacturer does not specify which version of the SX126x family of chips it uses. Therefore the SX1268 Datasheet is used as the baseline for information.

[3]Tested in the 490 MHz band.

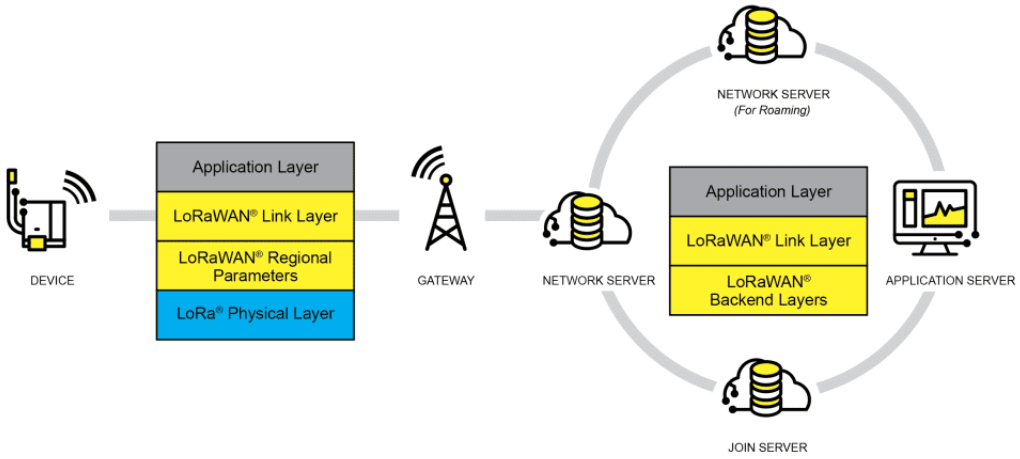

Fig. 4.7: LoRaWAN network architecture - Reprinted from [25]

the LoRaWAN network. It is meant as the lowest power bi-directional class of devices. The device in this class can initiate a transmission at any time. The transmission is then followed by two short downlink windows. Class B devices are, in addition to properties of a Class A device, synchronised to the network by using periodic beacons. Furthermore, it has scheduled downlink slots. Class C devices are always receiving, allowing the network to initiate a downlink (sending a command) at any time at the cost of more power consumption. [25]

LoRaWAN is capable of baud rates between 0.3 kbps up to 50 kbps. The data rate and radio output power for each end-point device is managed by the network. [25]

4.4 Payload

The main payload of Femtostrat-1 is a sensor suite which includes a barometer, a gyroscope, and a GPS module with a ceramic antenna. Additional space on the satellite board might be used to mount additional payloads.

4.4.1 Gyrometer

The gyrometer sensor of the satellite is an MPU6050 sensor placed on a GY-521 breakout board (see figure 4.8). This package is a very cheap COTS component, commonly used for prototyping applications with microcontrollers, such as Arduino or Raspberry Pi. A MEMS gyrometer measures the acceleration and gyroscope values based on motion imparted on a mechanical system. This system in turn changes the voltage that is read by analog-to-digital (ADC) converters.

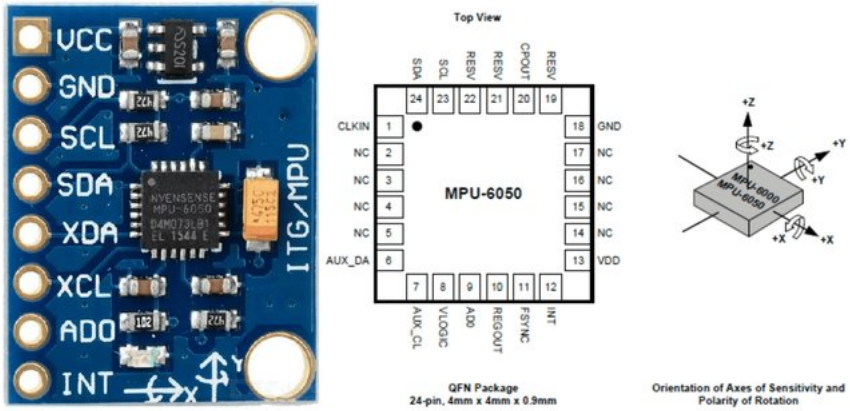

Fig. 4.8: MPU6050 Breakout board and QFN Package - Reprinted from[26]

The MPU6050 is a MEMS gyrometer, capable of outputting 3-axis acceleration data and 3-axis angular velocity data. Both the accelerometer and gyroscope are connected to three 16-bit ADCs each. It also features a built-in temperature sensor. It can be interfaced through I2C which allows for an easy integration into the wider system. The sensor package contains a Digital Motion Processor (DMP) capable of complex motion computations. The DMP offloads processing power from the microcontroller and lowers the amount of polls required. It has an operating current of 3.9 mA at its highest usage, with gyroscope, accelerometer, and DMP running at the same time. Its operating range is from -40 °C to +85 °C making it capable of handling temperatures in the stratosphere with almost no protection. [27]

The MPU6050 package was deemed too small for manual soldering onto the board. Therefore, it was opted for a breakout board. The GY-521 breakout board consists of a voltage regulator to the 3.3 V input, I2C Bus pull-up and interrupt pull-down. [28]

The package size of the module, including margins, is 20.261 mm by 15.56 mm. The module can additionally be screwed onto the board for further stability with 3 mm screw holes. For reference, the schematic of the breakout board might be found in appendix B.1. The GY-521 pinout consists of eight pins as shown in table 4.1.

The GY-521 module can only be interfaced through I2C serial bus. The auxiliary I2C pins are intended for connecting other I2C modules, such as magnetometers, to enhance the gyrometer. These auxiliary pins are optional. Similarly, the interrupt and address select are optional. These optional pins were not connected to the rest of the system in the final design.

The rotation and accelerometer data can be processed to extrapolate (citation

Pin Number	Pin Name
1	VCC
2	Ground
3	Serial Clock (SCL)
4	Serial Data (SDA)
5	Auxiliary Serial Data (XDA)
6	Auxiliary Serial Clock (XCL)
7	I2C Adress Select (AD0)
8	Interrupt (INT)

Table 4.1: GY-521 pinout

needed) the current position of satellite based on its last location and the accelerometer data. It can also be used for attitude estimation. Finally, the included temperature sensor provides an additional source of temperature readings for the satellite.

4.4.2 Barometer

The barometer sensor on board the Femtostrat-1 is a BME280 barometric sensor from Bosch. The sensor itself is in a small 2.5 mm by 2.5 mm package. It can be interfaced through I2C, or SPI serial interfaces. It has a wide supply voltage, from 1.71 V to 3.6 V. Additionally, it has very low current consumption. At its highest usage, when humidity, pressure, and temperature is processed, it has a current consumption of 3.6 µA at 1 Hz. It also features a wide operating range from -40 °C to + 85 °C, making it suitable for stratospheric applications. However, it should be noted that the barometer was not expected to provide pressure readings beyond an altitude of approximately 9,000 m as its pressure reading range is 300 hPa to 1,100 hPa. Additionally, the external temperature sensor is not rated for temperatures lower than 0 °C. [29]

Because of its thermal sensitivity and small size, the integration of the chip by hand was deemed unpractical. Therefore, a small breakout board was chosen to be mounted on the satellite instead, providing easy mounting as well as reducing the amount of necessary discrete components.

The breakout board for the BME280, in addition to the BME280, consists of the necessary regulation circuit for easy interfacing with a microcontroller. It contains a 3.3 V LDO regulator and a logic level shifter. Thanks to these components, the module can be interfaced with both 5 V and 3.3 V microcontroller pins. Last, the module has four interface pins (see table 4.2) and three I2C address selection pads. For further reference on the breakout board, see appendix. B.2

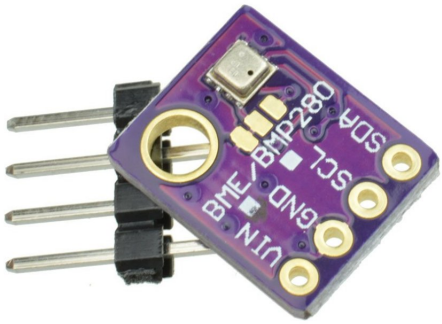

Fig. 4.9: BME280 breakout board - Reprinted from dratek.cz[29]

Pin Number	Pin Name
1	VIN
2	Ground
3	Serial Clock (SCL)
4	Serial Data (SDA)

Table 4.2: BME-280 module pinout

The barometric pressure from the sensor can be used to approximate the height at which the satellite is located. The external temperature sensor might be cross-checked with the temperature sensor included in the OBC to provide further data about the status of the satellite and the environment around it.

4.4.3 GNSS Module

The GNSS module payload chosen for Femtostrat-1 is the uBlox MAX-M10S. The MAX-M10S is a small 10 mm by 9.6 mm GNSS module which made it suitable for use in Femtostrat-1. It has a very low active power consumption of 25 mW. Its operational altitude is as high as 80 km, ensuring no issues during the stratospheric flight. Its operating temperature range is -40 °C to +85 °C. Similarly to other payloads, this makes it resistant to the ambient temperature at the expected operating altitudes of up to 40 km [30].

The module is a package capable of acquiring all L1 GNSS signals from GPS, GLONASS, Galileo, and BeiDou networks. The low power consumption enables

long duty cycles for acquiring location data.

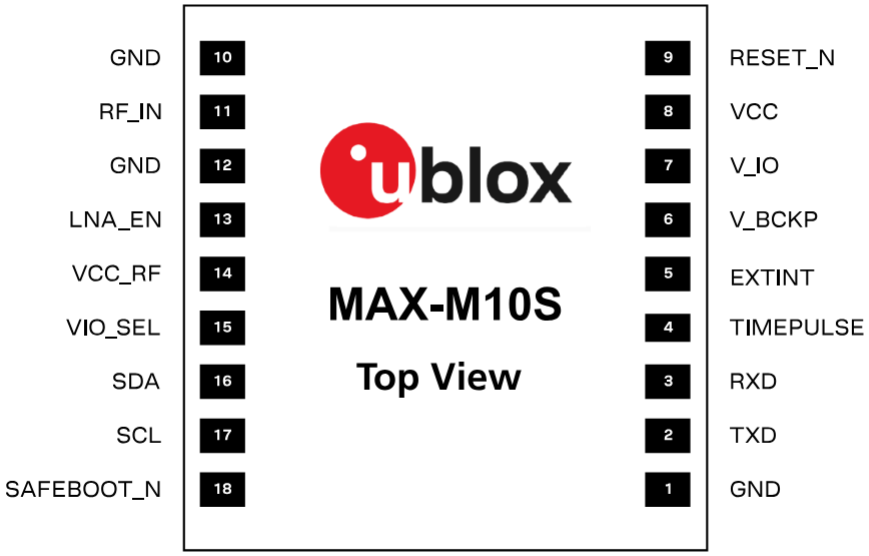

Fig. 4.10: MAX-M10S GNSS module pinout - Reprinted from MAX-M10S Datasheet[30]

The module is capable of communicating through both I2C and UART serial lines. To simplify integration into Femtostrat-1, the UART interface was used for this module.

Additionally, a GNSS antenna was connected to the module. Two antennae were considered and acquired. The Kyocera AVX M830120 SMD antenna was considered for its small dimension and very low mass of only approximately 200 mg. Because of complex soldering and board layout requirements, it was not chosen as the final GNSS antenna for the prototype.

The Maxtena MPA-134-GPS ceramic microstrip GNSS antenna was chosen as the prefered antenna for the prototype. The MPA-134-GPS has larger size and mass of approximately several grams compared to the Kyocera AVX antenna. However, it had significantly simpler integration and higher sensitivity which were considered more important for the initial prototype.

5 Femtostrat-1 Implementation

5.1 Circuit Board Design

The circuit board design of Femtostrat-1 was built in revisions. The initial revision was intended as a starting point, allowing for a rough estimate of size and weight. It included a CR2032 battery as its power source which was deemed insufficient early in development. In addition, the board lacked sufficient grounding and the antennae lacked impedance matching,

The second revision incorporated an AAA type battery which also required a DC-DC Converter circuit to generate a stable 3.3 V power source. The power circuit was initially intended to be created with THT components, with the intention being that it would be simpler to prototype. The lack of ground was also resolved. This revision had tracing issues and the THT components were unsuitable for developing the board further.

The third revision incorporated SMD components as well as GNSS and Sensor module switches in the form of MOSFETs. The tracing issues were resolved and the ST-Link Header was extended to include the RST pin.

The fourth revision was the first board to be fabricated. This revision was considered the Prototype Flight Model (PFM). The barometer and GNSS modules were moved away from antennae to avoid any interference. Furthermore, following the antennae datasheets, the ground was removed at the edge of the PCB where antennae were placed to improve its performance. The DC-DC converter circuit was reworked according to the integration manual to ensure optimal performance.

The fifth, and the last, revision served as the final flight model of Femtostrat-1. An issue with the wiring of the DC-DC converter was found and fixed. A manual reset switch was added to simplify programming the OBC. A status LED, together with its resistor, was added to provide visual checks during the pre-flight phase. A Debug UART header was also added to allow for debugging with serial output. Last, the I2C pull-up resistors were connected to the sensor 3V3 input to prevent current drain through the bus if the sensors are off.

5.1.1 Circuit Schematics

The circuit schematic was based on the initial design. However, reviews and the associated revisions also affected the overall design. Ultimately, the schematic was divided into sections defined in the final design of Femtostrat-1.

The gyroscope module contains only the gyroscope header. The auxilliary clock and data pins, interrupt, and chip select pins are unused. The main I2C clock

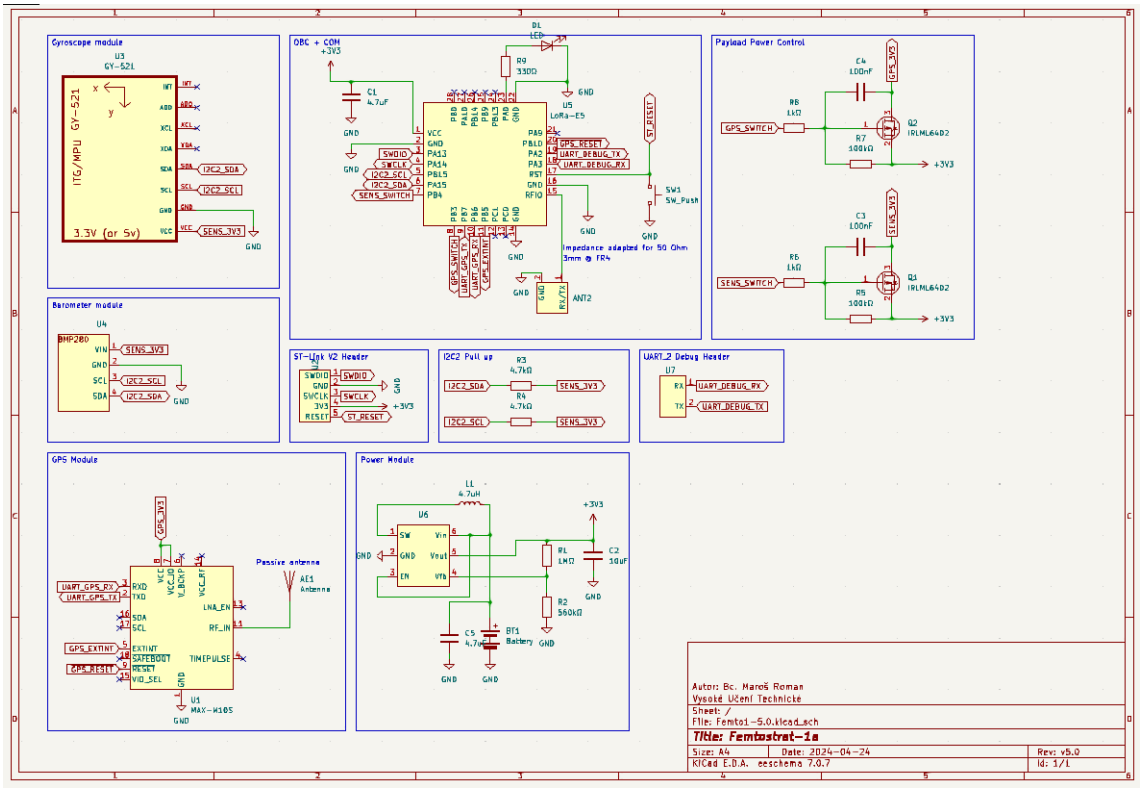

Fig. 5.1: FS-1 circuit schematic

and data pins are connected to the I2C bus. The 3.3 V input pin is connected to the sensor 3.3 V bus.

Similarly to the gyroscope module, the barometer module only contains the barometer header. Its I2C pins are connected to the I2C bus, and its 3.3 V input is connected to the sensor 3.3 V bus.

The GNSS module consists of the module itself and its antenna. The VCC and VCC_IO pins are connected to its own 3.3 V switched power source. The UART RXD and TXD pins are connected to the USART1 TX and RX pins on the microcontroller respectivelly. Last, the external interrupt and reset pins were connected to the associated pins on the microcontroller.

The power module consists of the MCP1624 DC-DC converter, the battery, and the associated regulating circuit. At the regulator input is a non-rechargeable battery, with the output decoupled to ground with a 4.7 µF capacitor. The battery voltage is also fed into the enable pin of the regulator. Additionally, the battery voltage is also fed into the switch node pin which carries the 4.7 nH inductor current from the input pin. The enable pin powered on at all times, as it is undesirable to power off the regulator.

The feedback and output voltage pins are connected to a resistor divider. This

provides the voltage regulation to 3.3 V. Last, the 3.3 V output is decoupled to ground with a 10 µF capacitor and fed to the satellite and the power control circuit.

The payload power control circuit encompasses the MOSFET switches and their regulating circuits. The gate of each MOSFET is switched by a separate GPIO port of the OBC. To prevent inrush current causing a reset of the system, a soft-start circuit is formed for each switch with a 1 kΩ resistor and a 100 nF capacitor. The second 100 kΩ resistor is added to facilitate a standby mode during OBC restart.

The OBC and COMM module mainly consists of the WIO-E5 module, the Lynx 868 MHz antenna, and the 3.3 V input. The OBC is the central point of the PCB. All communication buses originate in this module and both power switches are controlled from here.

The last two small sections of the schematic are the ST-Link header which connects the ST-Link to the associated pins on the Wio-E5 module. The I2C pull-up contains the pull-up resistors connected between the I2C bus and the 3.3 V sensor voltage source. Last, the DEBUG_UART header connects the RX and TX pins of a FTDI converter to the RX and TX pins of UART2 of the WIO-E5 module.

5.2 Engineering Model

Before fabricating a prototype model of the Femtostrat-1 satellite, all functions were tested on an engineering model. The entire engineering model was placed on a breadboard to allow for fast prototyping (see appendix C.1).

All components and functions that were expected on the circuit board of the first prototype were acquired in a package that could be easily integrated into the engineering model. Each such component was then tested. This included temperature sensor testing, gyroscope testing, main computer, and transmitter testing. Because of the easy access and fast component exchange, the firmware and component drivers were also developed and tested on the engineering model. The engineering model was also used to test communication with the ground station.

Additionally, because of the difficulty of testing SMD components on a breadboard-based engineering model, a proto-flight model was created.

The proto-flight model was fully functional, with theoretical flight capabilities. However, there were slight changes to accommodate testing. Instead of being directly soldered, the sensor boards were placed in headers that were soldered onto the board. This allowed for fast exchange of sensors between the engineering model and the proto-flight model. Additionally, instead of a battery case, jumper cables were used for selecting either a laboratory DC source or a battery power source.

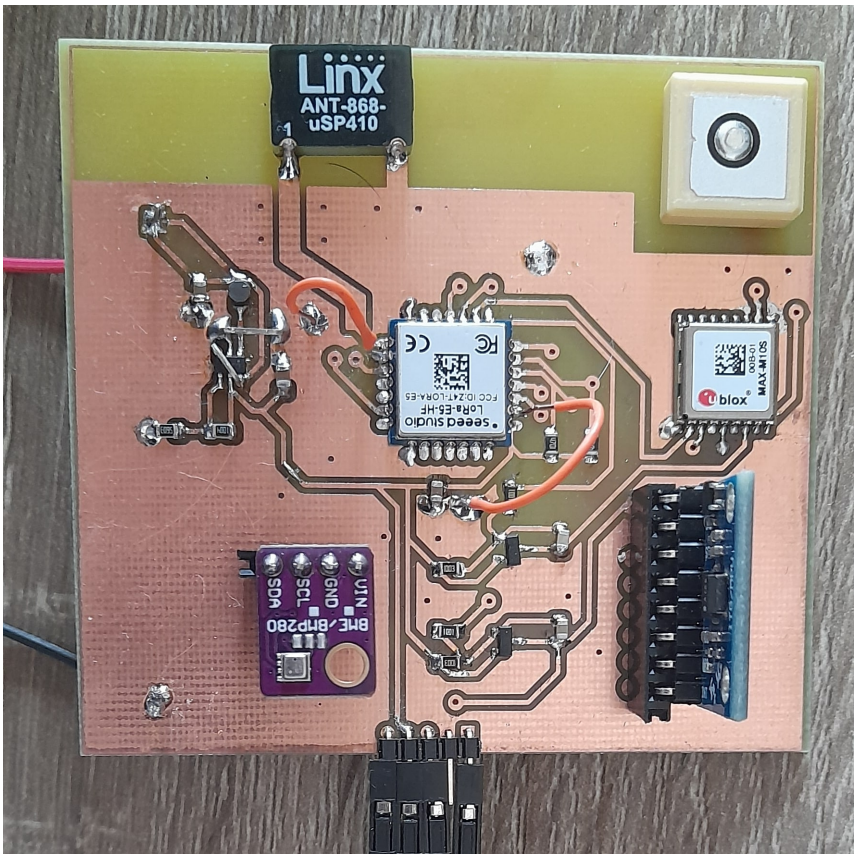

Fig. 5.2: Proto-flight model

5.3 Testing and Verification

The availability of an engineering model, and later a prototype flight model, allowed for extensive testing.

Each update of firmware on the EM and PFM was followed by functionality tests. For updates on the EM, the desired results were verified on a debug serial link. Debug printouts and sensor outputs were verified. For example, the gyrometer was rotated or moved at a certain velocity and the results were verified in real time through debug printouts of its outputs.

Furthermore, during each firmware update, correct transmissions were succefully confirmed against both the Femtostrat-1 ground station and local ground stations in Brno.

After the prototype flight model was fabricated and became available, the tests were transferred onto it. Because of it being battery powered, field tests were possible. Its transmissions were first verified against the Femtostrat-1 ground station. Next, the PFM was taken outside to a nearby hill to check whether it could receive and transmit signals from third party ground stations (see figure 5.3).

Fig. 5.3: PFM testing process

During the testing itself, all functions of the COM and OBC module were functional. Therefore, the flight model revision could be created and fabricated with a reasonable degree of confidence in its functionality.

5.4 Fabrication

The Femtostrat-1 was fabricated in two revisions. The first was a proto-flight model. This model was using the revision 4 of the Femtostrat-1 circuit board.

For the first iteration of a fabricated board, it was decided that the cheapest option with the most room for error would be to fabricate the board by using the available student workshop. The PFM was fabricated onto a FR4 substrate. The circuit board had to be further modified prior to providing the deign files to the workshop staff. The changes included enlarging all via drill sizes to minimum of 0.6 mm with an additional space of the copper area of the via. Moreover, the spacing between traces had to be enlarged significantly. This caused the loss of some ground fill.

The fabrication took approximately two days. Because of the last minute adjustments, several issues were found with the PFM board. A trace had connected both sides of a resistor in the sensor switch circuit, causing a short. Additionally, because of the loss of some ground fill, two out of four grounds on the OBC were unconnected. These had to be connected manually with wires soldered onto the outside ground fill. These wires can be seen in figure 5.2 as additional orange wiring.

Furthermore, there was no reserve for the header pins of the sensors and the ST-Link header. After the holes had copper added to create a via, about 0.05 mm to 0.1 mm of hole thickness was lost, causing the pins to not fit. Because of this, the holes had to be additionally drilled. However, the design did not assume the holes to use a via. Therefore, all the lines, with an exception of the gyroscope VIN pin, were traced from the bottom layer. This made the soldering from a single side sufficient. An issue did arise with the gyroscope VIN pin later, because it was discovered that additional soldering from the top was necessary to provide a connection.

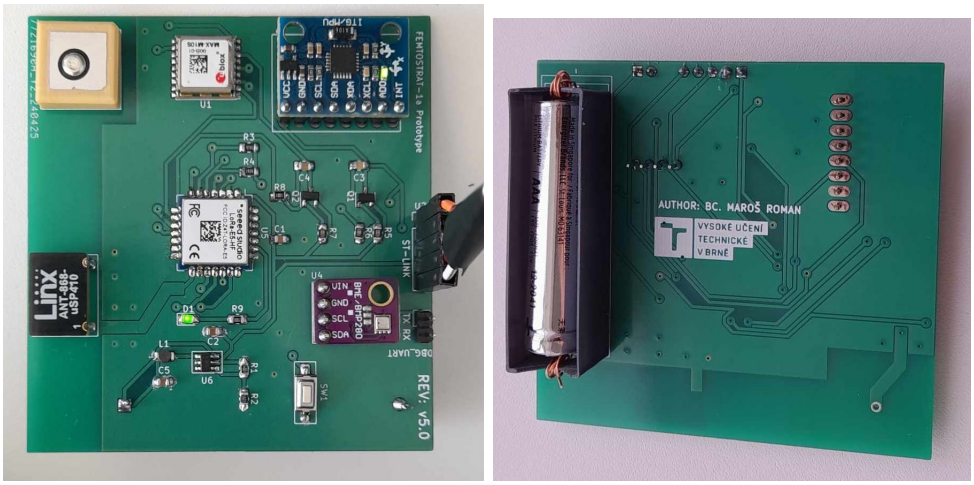

Fig. 5.4: Femtostrat-1 flight model

As mentioned in section 5.1, an issue with the design of the traces for the voltage regulation module was causing a short circuit. As a workaround, spare header pins were bent and used to rewire the DC-DC converter to prevent the short circuit.

Soldering of the proto-flight model was done inside the premises of the space laboratory of Brno University of Technology. The main drawbacks were a low marging of error because the tools of the laboratory were insufficient for unskilled resoldering of components.

The flight model was fabricated based on the revision 5 of the PCB design (see section 5.1). In this case, the PCB was fabricated professionally by JLCPCB. The circuit was fabricated on a 1.2 mm thick board. The final dimensions were

71.1 mm by 66 mm. The surface finish chosen for the pads and vias was HASL. The choice of surface finish primarily depended on price and the finish also allowed easier soldering.

The soldering of the flight model was done in the laboratories of VZLÚ. For future implementations of the board, it is recommended that at least a pincer-type solder head is available, because it was necessary for certain component types. Particularly, the pincer-type solder head is necessary for error correction or resoldering components.

5.5 Firmware

This chapter describes the basis of the Femtostrat-1 firmware package, the philosophy behind its design, and issues encountered during its creation.

5.5.1 Overview

Based on its design, the Femtostrat-1 firmware package had several objectives.

The first objective was to connect the prototype to the LoRaWAN network. The second objective was for firmware to facilitate receiving and trasmitting data on the board serial lines to gather sensor and GNSS data. Third, the prototype firmware had to be capable of receiving and processing commands from a LoRaWAN gateway. Last, the firmware had to be robust, capable of self-correcting, and avoiding being blocked.

5.5.2 Firmware Basis

The basis of the firmware for Femtostrat-1 was formed according to the documentation listed on the Seeed Studio wiki page. [31]

The initial codebase used for Femtostrat-1 was a public repository provided by Seeed. [32] The repository was run as-is on the Femtostrat-1 engineering based on instructions in the README.md file of the repository and documentation provided by the Seeed wiki.

The values necessary for identifying the device and joining the LoRaWAN network were entered into the template code. After setting up the Femtostrat-1 gateway as mentioned earlier in the thesis, the code was verified to succesfully transmit packets to TTN.

Afterwards, a define statement was overridden to use Cayenne LPP coding for the communication and gyroscope telemetry was added to the message trasmitted. However, during an attempt to modify UART settings of the board through

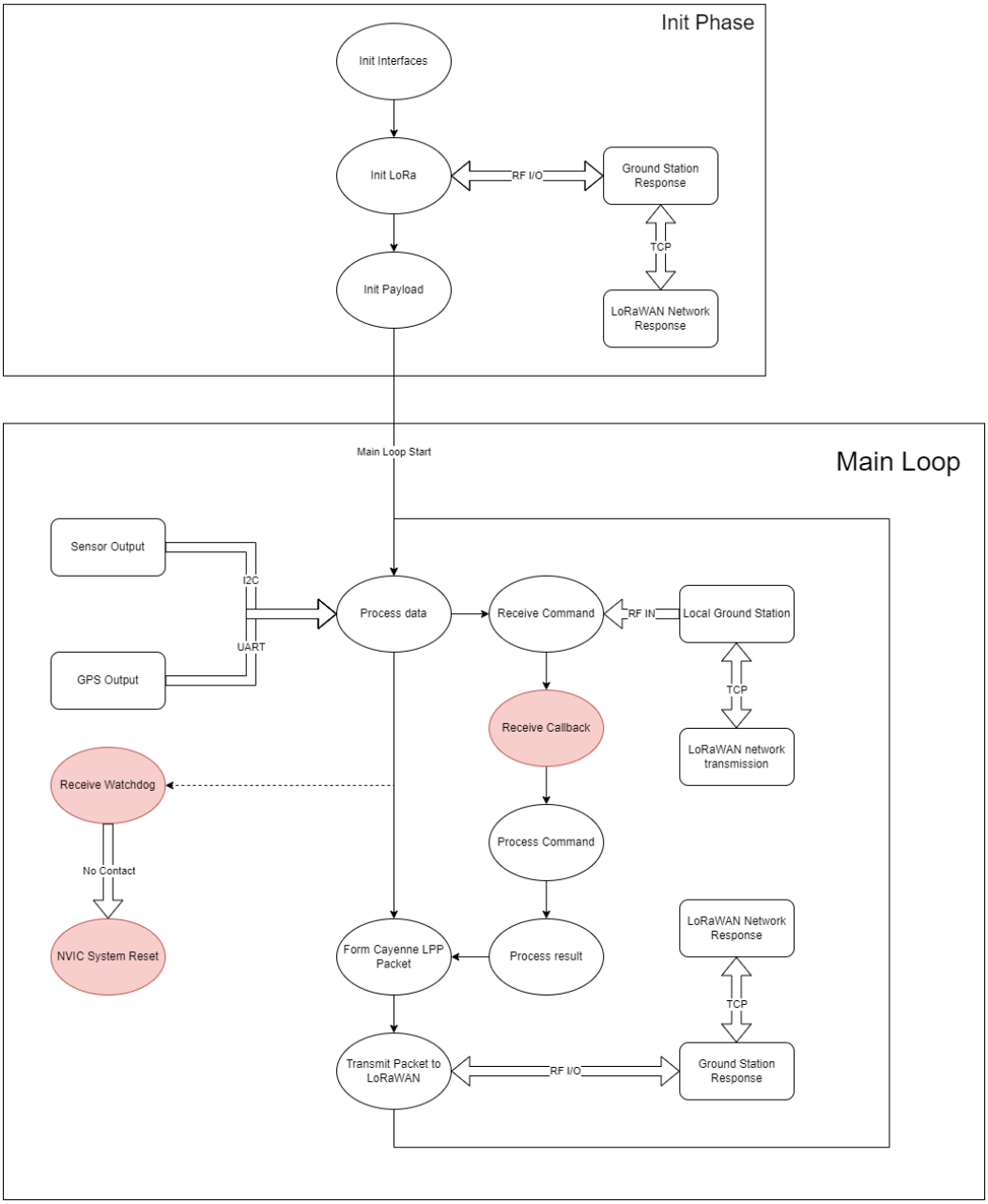

Fig. 5.5: Firmware operation diagram

the STM32CubeIDE visual interface, an issue was found with the version of the Seeed GitHub example code. This issue caused large scale overwriting of the code, rendering the code unusable.

To circumvent this issue, a fork of the original example was found which partially resolved the issue with STM32CubeIDE code overwriting. [33] Because of this, changes could be made to the board interfaces without losing the ability to build the resulting code. Any additional issues with overwriting were resolved with manual

diff checks against a checked out repository.

Because of the fixed example being only a basic central loop instead of an RTOS implementation, it was decided against a rewrite and the main loop method was chosen instead.

5.5.3 Operation Overview

The firmware operates in two modes: initialisation mode and transmitting mode. The transmitting mode is then further split into several submodes.

The initialisation mode begins with initialising all peripherals, the flash interface, and systick. This includes the sub GHz radio module, the I2C2 bus, both UART buses, and the LoRaWAN middleware.

The sensor power switches are enabled by default. However, there is code included for disabling them on start-up. Additionally, the GNSS reset pin is set to HIGH.

Next, the main LoRaWAN middleware UART and I2C handles are set. The gyrometer is initialised on I2C2 using its driver.

Last, the I2C bus is checked and a general status of the board is transmitted to the debugging UART interface using a custom printing function. The initialisation mode then triggers the main loop and ends.

The main loop consists only of one function which is the LoRaWAN processing function. Its operation is based on timers and callbacks. The main timer used is the TX timer which was set to 10 seconds for the flight model. This timer determines the duration of time between individual transmissions. At the end of each time period, the transmission callback function is run.

The transmission function consists of several steps. First, it reads all available data from sensors, the GNSS module, and ADC registers. For each, it outputs debug status messages and visually confirms its activity by using the onboard LED. Second, based on the submode being run at this point, either a Cayenne LPP message is formed or a plain hexadecimal message is created. The Cayenne LPP message is created from the retrieved data. Each data point receives its own channel for differentiation. In case of a repeater, or hexadecimal transmit submode, the hexadecimal message size is copied into the transmission buffer. In both cases, the transmission buffer is transmitted to the ground station without confirmation from the station itself.

Upon receiving a message, the RX callback function is started. The RX function consists of several operations. The RX timeout watchdog, described later in this chapter, is reset. A description of the received message is output on the debugging UART interface. Last, the message contents are decoded and processed. In case

the target port of the message is the LoRaWAN switch class port (port number 3), the associated LoRaWAN middleware function is run to process the request from the network. An example of this is a rejoin interaction between Femtostrat-1 and the LoRaWAN network.

In case the port is the user port (port number 2), it is assumed the message contents are a command. This message is then processed according to the command table of Femtostrat-1.

Command	Description
0x01	Visually confirm transmission has been received - Blink debugging LED 20 times.
0x02	Switch to plain hexadecimal broadcast submode - Transmits: "This is a trasmission from Femtostrat-1, a prototype stratospheric satellite from Brno! Sensors <ON/OFF>! GNSS <ON/OFF>!".
0x03	Switch to Cayenne LPP telemetry broadcast submode.
0x04 <DATA>	Repeat the content of data received as argument of this command. Note: Must be in plain hexadecimal submode.
0x05	Toggle sensor power.
0x06	Toggle GNSS power.
0x07	Restart the MCU.

Table 5.1: Femtostrat-1 command table.

To ensure the stability of the system and manage edge cases where received messages might not be processed for any reason after a set amount of time, a watchdog timer was included. This timer is started at the point of initialisation of the LoRaWAN middleware and is set to 10 minutes. In case the RX callback function is started, the watchdog is reset. In the case that no message is received for more than 10 minutes (meaning the RX function did not start for this time), the watchdog timer runs out and the MCU is restarted to rejoin the LoRaWAN network.

5.6 Ground Station

For the ground station hardware, several alternatives were considered. Initially, as described in previous chapters, the TinyGS network was considered as a viable method of receiving telemetry. Although the TinyGS network would be compatible

with the chosen band in which the prototype would communicate, a custom implementation of communication algorithms would be required. Additionally, if no ground stations would be available in the launch region, custom ground station hardware would be required as well. The second significant disadvantage of the TinyGS network is the inability to transmit commands. This would mean that a custom ground station is required if two-way communication is needed.

Therefore, a waveshare SX1302 LoRaWAN Gateway HAT was used for the local ground station. This is a header that is connected directly to a Raspberry Pi. For the realised ground station, a Raspberry Pi model 1 B+ was used as the central computer.The header was then connected to the Raspberry Pi. The header has two coaxial connectors. The first is connected to a 868 MHz antenna for receiving data from transmitting LoRa devices. The second connector is connected to a GNSS antenna, capable of receiving transmissions from GPS, Gallileo, and Glonass satellite networks. It should be noted that providing the precise location of the ground station is completely optional but considered a good practice.

The Raspberry Pi is equipped with an 8 GB memory in the form of an SD card, which contains the operating system.

To prepare the ground station for operation, an internet connection is necessary. For this implementation, a direct connection to the internet was created for the RPi with a LAN cable connected to a router. For interfacing with the RPi, a monitor and a keyboard was connected. The necessary software was provided by waveshare directly at URL together with instructions on installing the software.

Because of the age of the Raspberry Pi 1 B+, it was necessary to complete additional steps. First, it was necessary to open the SPI and I2C ports through the raspi-config utility of the RPi. Next, the ttyS0 port had to be configured following the instructions at the embeddedpi.com page. [34]

After completing these steps, the connection to the TTN was verified. When the gateway was connected and forwarding packets, the Femtostrat-1 engineering model was powered on and transmissions were successfully forwarded by using the ground station.

The usage of a ground station significantly lowered the response times of the satellite and significantly less packets were lost during the transmission.

The final layout of the completed ground station can be found in figure 5.6.

In addition to the hardware, a software for processing the received data was created. The TTN has an option of providing the forwarded data as a published message on a MQTT broker. The ground station software was designed as a subscriber to this MQTT server. For each packet received, a message is published to the MQTT server. This message is then received and interpreted by the ground station software. The data is integrated into the user interface as part of the associated

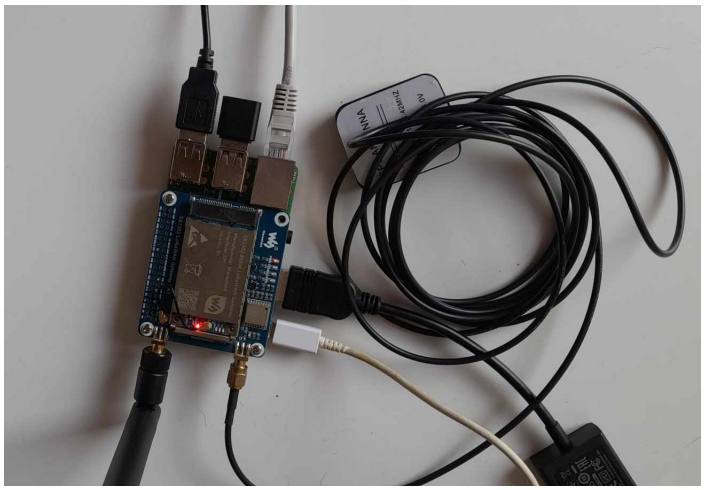

Fig. 5.6: Femtostrat-1 ground station

figure. Afterwards, the data is appended to a .csv file together with a timestamp.

The following data was saved: SNR, RSSI, angular velocity, acceleration, gyrometer temperature, barometer temperature, humidity, pressure, MCU temperature, MCU voltage, and gateway information. The Femtostrat-1 ground station UI can be seen in figure 6.3.

5.7 Femtostrat-1 Concept of Operations

The Femtostrat-1 launch consists of two segments, similar to a space mission: the stratospheric segment and the ground segment.

The ground segment, described in the previous chapter, consists of the ground station, the software suite, and the LoRaWAN network.

The ground station receives data directly from the stratospheric segment. It consists of a 868 MHz antenna, a GPS antenna, and a processing computer. Although the GPS antenna is optional, it is considered good practice to provide the location of available ground stations in the LoRaWAN network. The processing computer itself is a Raspberry Pi B+. It is connected to a local network through an RJ45 port. Furthermore, a monitor and a keyboard is available for monitoring the ground station status.

The software suite consists of two parts. The first is a packet forwarder software designed for the ground station hardware. It forwards packets of data received by the ground station directly to the LoRaWAN network. This application should run by default during the whole mission.

The second part of the software suite is the monitoring application. This application is expected to run on a second separate computer. It contains a detailed GUI that shows the latest status of the spacecraft and saves data gained from the LoRaWAN network through a MQTT subscription for later use.

The stratospheric segment consists of the Femtostrat-1 itself.

Before the launch sequence, the Femtostrat-1 board is expected to go through several preparations.

First, the Femtostrat-1 is powered on through the ST-Link connection. A debug UART connection is made as well. All functions of the platform are verified. Among the checked items are the transmission and reception of LoRa messages, tested against the local ground station, sensor check, and GPS data check.

Next, the debug UART and ST-Link is disconnected. A battery is placed into the platform to power it up. The powered-up platform is checked one more time against the ground station. If packets are received, the platform is encased in the protective polysterene shell.

Last, the platform is connected to the stratospheric baloon and ready to launch.

During the flight, the platform is expected to transmit telemetry every 10 seconds. Furthermore, it should receive and process commands from the ground station.

6 Femtostrat-1 Launch

The Femtostrat-1A prototype was succesfully launched on 7^{th} May 2024 from the Gánovce meteorological complex near the city of Poprad in Slovakia.

This chapter describes the events during the pre-flight, flight and end-of-mission phases.

6.1 Pre-flight

In preparation for the flight, several actions needed to be done. First, the polystyrene casing needed to be created for the prototype. This was done on the day of the flight, because no other hardware modifications were expected to change the envelope of the prototype.

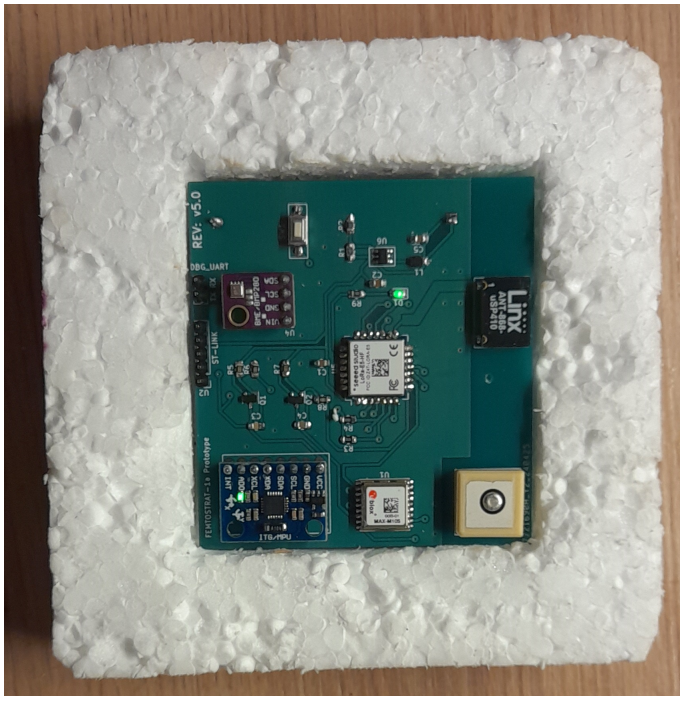

Fig. 6.1: Femtostrat-1A in polystyrene casing

On the day of the launch it was raining heavily directly above the launch location (see appendix G.1). The flight date was preplanned based on the time constraints of this thesis. As such, it was not feasible to change the launch date. Because of this, it was decided to proceed with the launch despite the unfavourable weather.

Shortly after arriving at the launch location, a ground station and a testing setup was created on site. The testing setup consisted of a notebook with a UART converter and a ST-Link. A pre-flight checklist was prepared and followed before the

flight. This included regulated voltage checks, battery checks, transmission checks, and commanding checks. The ground segment was also tested during this phase. The Things Network UI and the custom ground station software was verified. An issue was found in the ground station software during this phase, but it was not fixed in time for the flight.

Because of the adverse weather conditions and the necessity of having an external antenna (the building was blocking RF signals), the entire ground station was placed in protective packaging (see appendix E.1) and placed outside. It was connected through an RJ45 cable to a ground station computer.

On a separate computer, The Things Network UI was started to view and save unprocessed .json output.

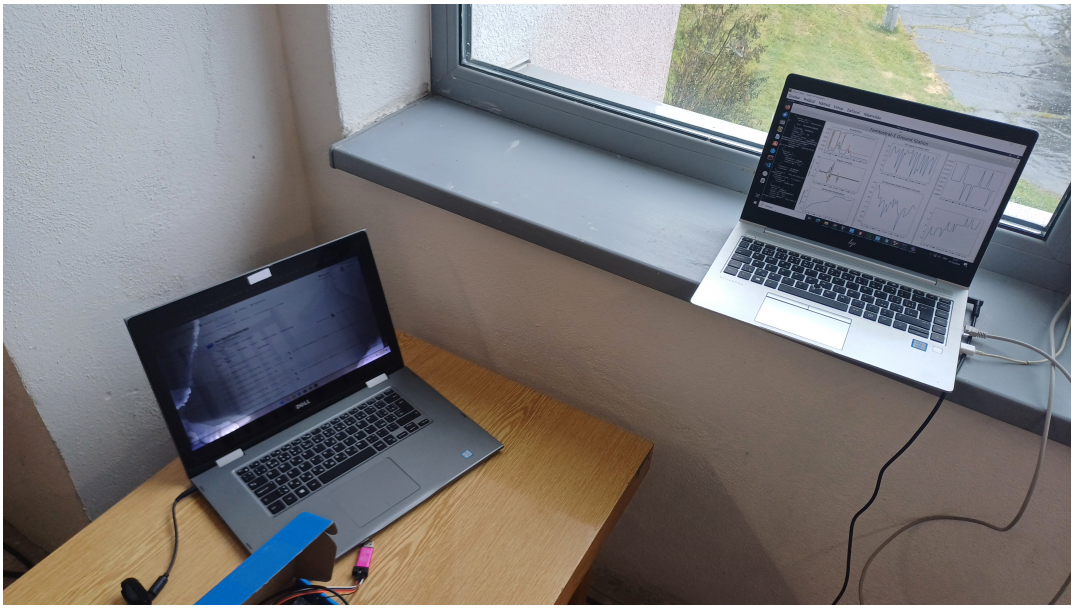

Fig. 6.2: Femtostrat-1 ground segment

After the ground segment and the stratospheric segment were checked out, the prototype was placed into its protective polystyrene shell. At T-10m from launch, a battery was placed into the prototype and a fast health check was performed. Afterwards, the polystyrene shell was sealed with duct tape. Four nylon lines were added to the shell to keep the prototype in an 'upright' position, that is, the prototype antenna was facing upwards. This orientation helped use the antenna directivity fully. The shell was then connected to the high-altitude balloon with the nylon lines.

6.2 Flight

During the flight phase, the prototype experienced three types of radio contact.

Telemetry contact was a type of radio contact where full telemetry packets were decoded and forwarded to TTN. During this time, the ground station software was able to process and save the decoded telemetry.

Non-telemetry contact was radio contact without telemetry. During this type of contact, the LoRaWAN gateways received packets from the prototype but were unable to decode their contents, as such only the SNR, datarate, and RSSI values were extracted from the communication. In some cases, instead of signal information, rejoin communication occured as the prototype attempted to re-enter the LoRaWAN network.

No contact occured when no gateway reported a packet from the prototype and was considered as complete radio silence.

The prototype was launched together with a meteorological sonde at 11:19:11 UTC. Its vertical velocity was approximately 5 m/s. Within the first five minutes, the balloon crossed the cloud cover and visual contact was lost. The ground station was successfully receiving packets from the prototype. However, the aforementioned crashing issue of the ground station telemetry software appeared and 10 minutes of data were not processed after four initial packets. This data was however stored through TTN UI as .json files. Therefore, the data was not lost.

Constant telemetry contact was kept with the prototype for further 20 minutes. At T+28m, telemetry was lost (but packets were partially received by the ground station). This was later discovered to be an issue of the balloon being flown behind the building in relation to the ground station. After the balloon returned into visual range of the ground station, telemetry was received again until T+48m. The last telemetry packets received by the ground station in Gánovce were at T+58m4s.

At T+1h4m15s, telemetry contact was re-established from a LoRaWAN gateway in the city of Levoča. The station received approximately one packet per minute until T+1h8m25s. During this time, the prototype was at an altitude of 20 km. The external temperature was -56 °C. The prototype was 32 km from the Gánovce ground station at this point. From the telemetry packet, it was shown that the internal temperature of the polystyrene casing was -1.8 °C and rising at approximately 0.3 °C per minute. The angular velocity of the prototype was relatively stable at X: -19 °/s, Y: -3 °/s and Z: 166 °/s after compensation. This showed that the packaging was effective at mitigating X and Y axis rotation. Regulated MCU voltage readings were at 3.28 V, showing that the regulator had no temperature or input issues.

The relative signal strength indicator of the prototype was -114 dBm and the sig-

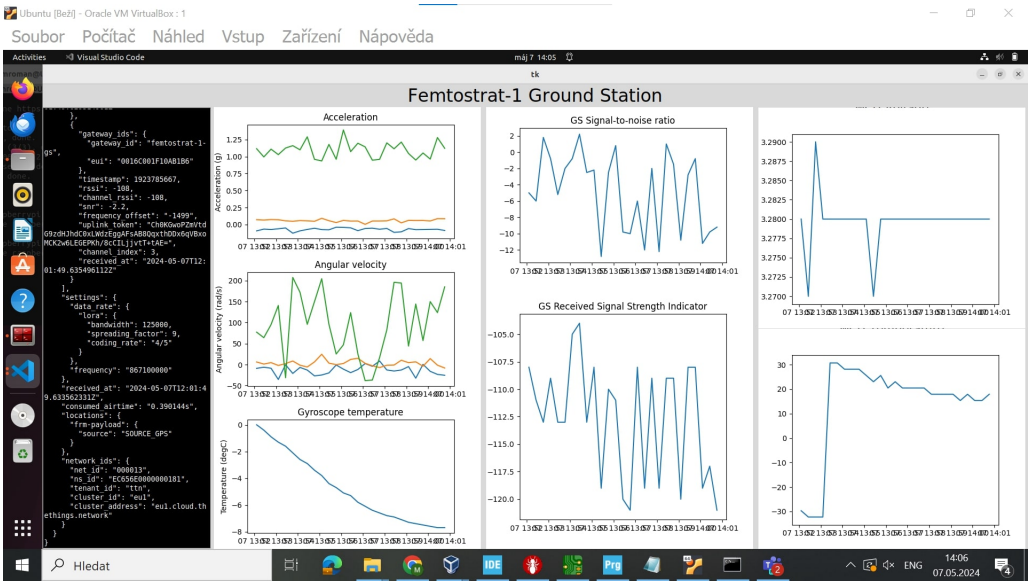

Fig. 6.3: Example Femtostrat-1 ground station output during flight

nal to noise ratio reached -8.2 dB. For comparison, the RSSI and SNR values at launch were -70 dBm and 14 dB respectively.

At T+1h8m25s, telemetry data was lost. Constant contact attempts by the prototype showed good health. Radio silence did not occur for more than 10 minutes at any point of the flight phase.

At T+1h55m19s, the high-altitude balloon burst and stopped transmitting meteorological data. Its average vertical velocity was estimated to be more than -10 m/s from GNSS data.

At T+2h16m4s, telemetry contact was re-established. The gyrometer angular velocity telemetry reading of X: 100.12 °/s, Y: -65.41 °/s, and Z: -246.13 °/s after compensation confirmed that the meteorological balloon had burst and the prototype was in free-fall. The internal temperature of the casing was at -0.3 °C and falling at approximately 1.5 °C per minute. At T+2h23m14s, the minimum internal temperature was reached at -10.1 °C. The external temperature at this point was extrapolated from previous data to be about -45 °C. It is assumed that the low external temperature and the cloud cover contributed to this minimum.

Constant telemetry contact was kept until T+2h34m44s, at which point the high-altitude balloon hit the roof of the Prešov Faculty Hospital.

The prototype was recovered and powered off at approximately 17:36 UTC (T+6h24m) by a volunteer. The readings made by the volunteer had shown that the battery voltage at the point of shut-down was 1.48 V. From the low drain performance graph of the L92, it could be discerned that the battery was above 50%.

For further information about the recovery and health checks of Femtostrat-1, see appendix G.

6.3 Results

The ground station software and TTN logs have provided sufficient data to create several analyses.

Although original concept of operations counted with barometric data and GNSS data, in-flight issues caused a lack of these. However, the gyrometer, the MCU ADC registers, and the ground stations provided enough data to show meaningful results. Additional data for comparison were gathered from the radiosonde on which the Femtostrat-1 flew.

It should be noted that because of crashes of the ground station and telemetry loss, large gaps in data occurred. These gaps can be filled with TTN .json logs. However creating scripts for extracting this data was not feasible within the timeline of the thesis.

6.3.1 Gyrometer

As described in section 6.2, when in flight, the prototype was very stable. As can be seen in Figure 6.4, except the initial large value caused by the launch itself, for the first few minutes, the angular velocity was low. This was most likely due to low wind speeds of around 5 m/s at the launch site. Additionally, at this time, the nylon thread holding Femtostrat-1 was still unwinding, dissipating some of the momentum.

At T+10m, it can be seen that significant movement on the Z axis had started occurring. This was most likely due to wind. It can be seen that the X and Y axes of the gyrometer were not experiencing angular velocities beyond 30 °/s. This state has stayed until T+1h8m after which telemetry contact was lost. When contact was reestablished at T+2h23m, Femtostrat-1 was already in free-fall. This state can clearly be seen in very high angular velocities in all axes.

The acceleration of Femtostrat-1 was relatively constant throughout the flight. At the start of the flight, there was a significant acceleration on the Z axis (Zenith of the prototype), showing acceleration when the prototype got pulled upwards (see Figure 6.5). Afterwards, no significant change occured until loss of contact. It can additionally be seen that the direction of the prototype was constant from the fact that the gravitational acceleration was seen on the Z axis with low changes. After contact was regained, the acceleration of the prototype was once again peaking at 2 g. These peaks changed as it rotated, with a minimum of -1.8 g on the Z axis. The X and Y axes did not experience as much acceleration, suggesting that

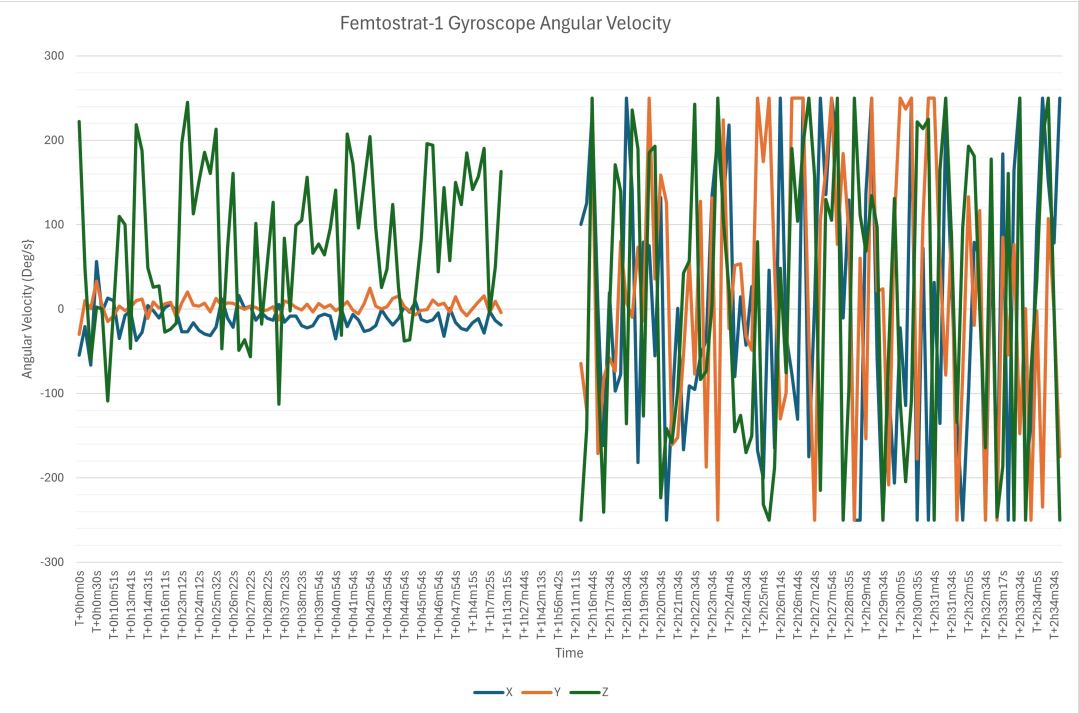

Fig. 6.4: Gyroscope angular velocity

the prototype had been 'gliding' or possibly acting as a parachute together with the remnants of the balloon.

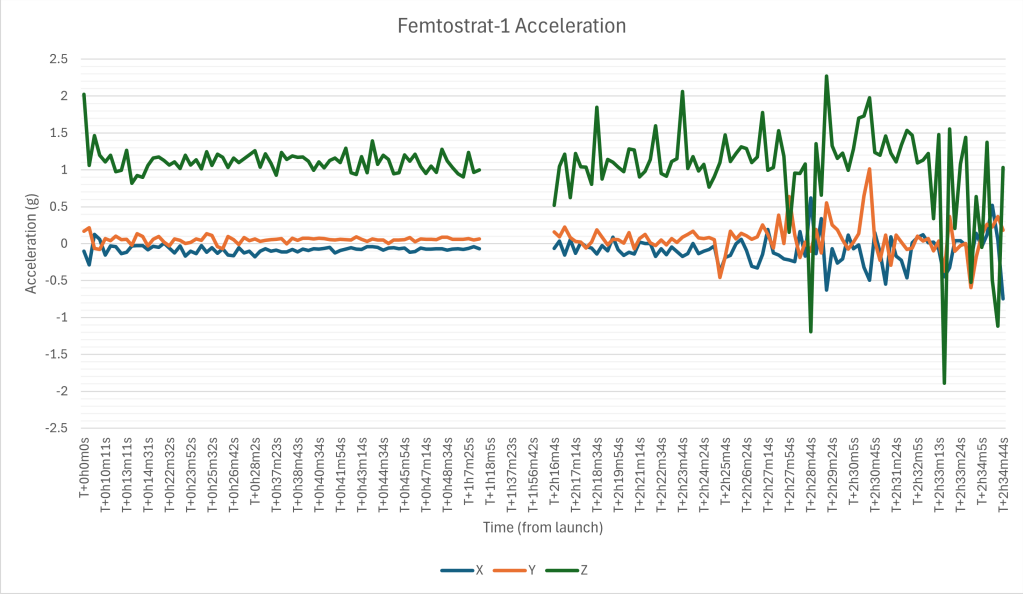

Fig. 6.5: Gyrometer acceleration

6.3.2 Temperature

The internal temperature of Femtostrat-1 was primarily expected to be gained from the barometric module. However, due to a failure of the module, the gyrometer temperature readings were used instead.

Multiple results can be seen in Figure 6.6. First, it can be seen that the internal temperature of the casing had 'inertia' when compared to the outside temperature. External temperature affected the internal temperature with large delays. Additionally, when examining the values at approximately the 8,000 m mark, it can be seen that the temperature change briefly stopped with even a small increase of +0.1 °C. Likely, the cause is that the moment at which Femtostrat-1 left the cloud cover it began receiving heat from the sun. At 11 km, it can be seen that the solar heat was insufficient at this point, with external temperatures reaching -60 °C.

At a 20 km height, contact was re-established and temperatures were shown to be rising again. The external temperature was -55.7 °C at this point in the flight which marked an increase of the temperature, but this would be insufficient for an internal increase. The prototype could have been covered in a protective layer of ice, or the low pressure allowed for a faster gain of heat than dissipation.

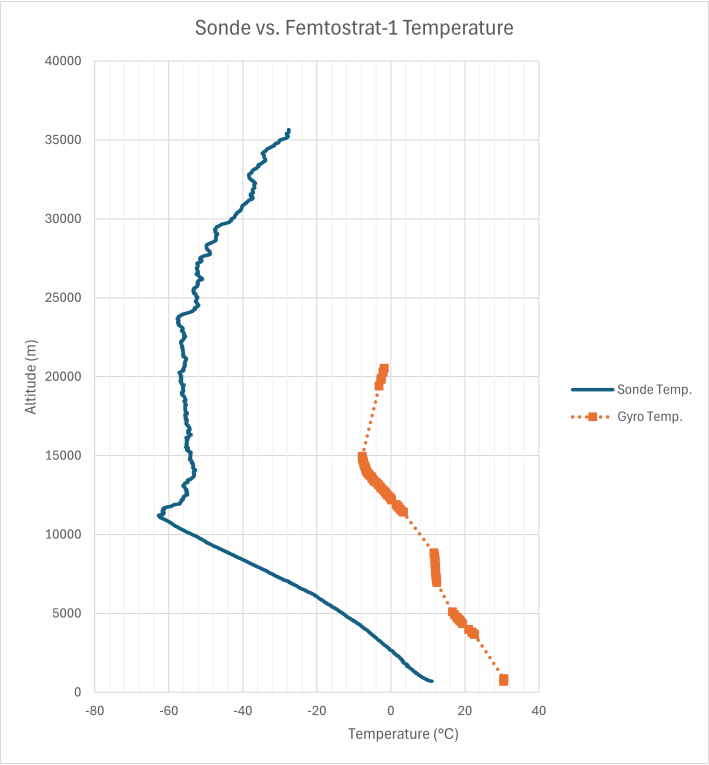

Fig. 6.6: Comparison of external and internal temperature

Additionally, the ADC register for the temperature reading of the OBC was out-

putting datapoints throughout the flight. However, these had large fluctuations and could not be decoded in a verifiable manner. The values seem to suggest a floating point value of around 30 °C and 10 °C at the least (see Figure 6.7).

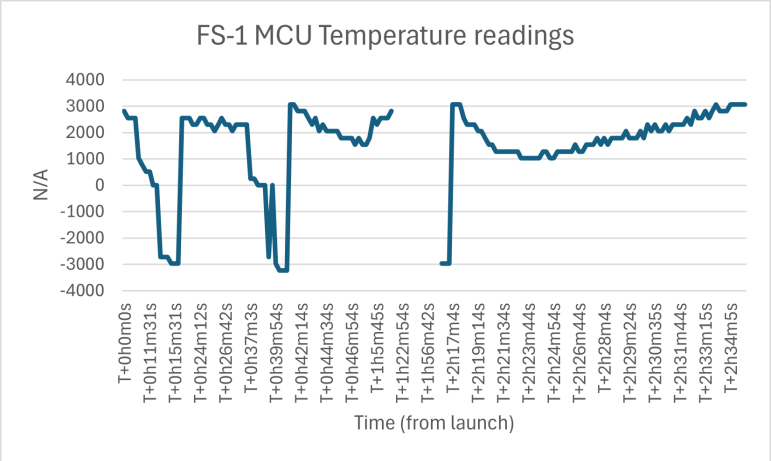

Fig. 6.7: OBC temperature readings

6.4 Gateway Information

Because of constant contact, a large amount of information was extracted from the signal information from the receiving gateways. This information was available even if no telemetry was provided. In figures 6.9 and 6.8 it can be seen that, as the altitude and distance of the prototype increased, the signal to noise ratio and signal strength decreased. It might seem that the SNR and RSSI stayed mostly constant between 0 dB and -10 dB, and -100 and -120 dB respectively. However, the graphs provided were extracted from the ground station software which could only provide signal data in case a telemetry packet was gained. This means that the figures 6.9 and 6.8 actually show the minimum required SNR and RSSI to receive telemetry.

In addition to information extracted from the ground station software, more information was gained from the .json logs extracted from TTN.

The last partial packet before the balloon burst was received at 13:05:45 UTC. The message contained settings information of Femtostrat-1 (see listing 1).

This packet was received by nine different gateways. Most did not provide location data, however some information was extracted. The one gateway which did provide location data was located in Novy Sacz in Poland (see listing 2).

The gateway with highest RSSI was an unknown packetbroker gateway. It had an RSSI of -109 dB and an SNR of -20.5 dB. The gateway with the highest SNR

```
"settings": {
        "data_rate": {
          "lora": {
            "bandwidth": 125000,
            "spreading_factor": 12,
            "coding_rate": "4/5"
          }
        },
        "frequency": "868100000",
        "timestamp": 4282091236,
        "time": "2024-05-07T13:05:43.733643Z"
      },
```


Listing 1: Settings information of a packet at 13:05


```
{
  "gateway_ids": {
    "gateway_id": "eui-2cf7f1106030004c",
    "eui": "2CF7F1106030004C"
  },
  "time": "2024-05-07T13:05:43.742155075Z",
  "timestamp": 1447031263,
  "rssi": -136,
  "channel_rssi": -136,
  "snr": -21.25,
  "location": {
    "latitude": 49.6288511807637,
    "longitude": 20.716530576563,
    "altitude": 294,
    "source": "SOURCE_REGISTRY"
  },
  "uplink_token": "CiIKIAoUZXVpLTJjZjdmMTEwNjAzMDAwNGMSCC...,
  "received_at": "2024-05-07T13:05:43.726969457Z"
},
```


Listing 2: Polish gateway information at highest point in flight

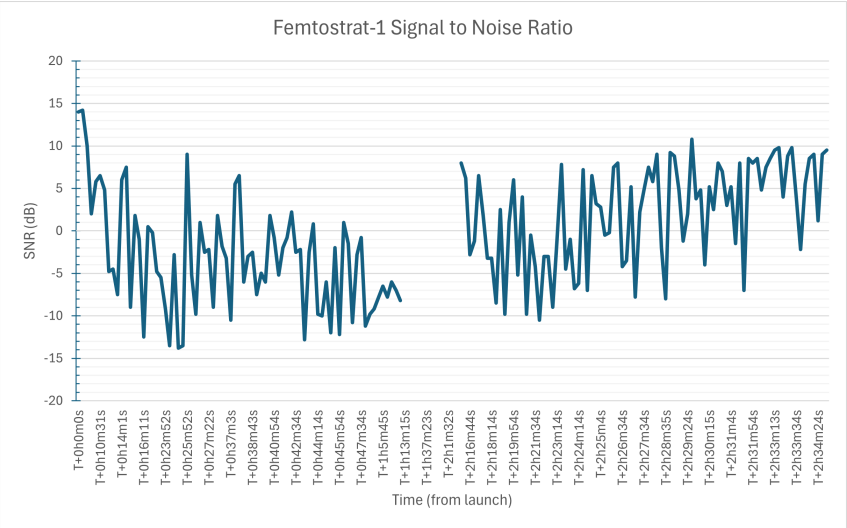

Fig. 6.8: Femtostrat-1 Signal to Noise ratio

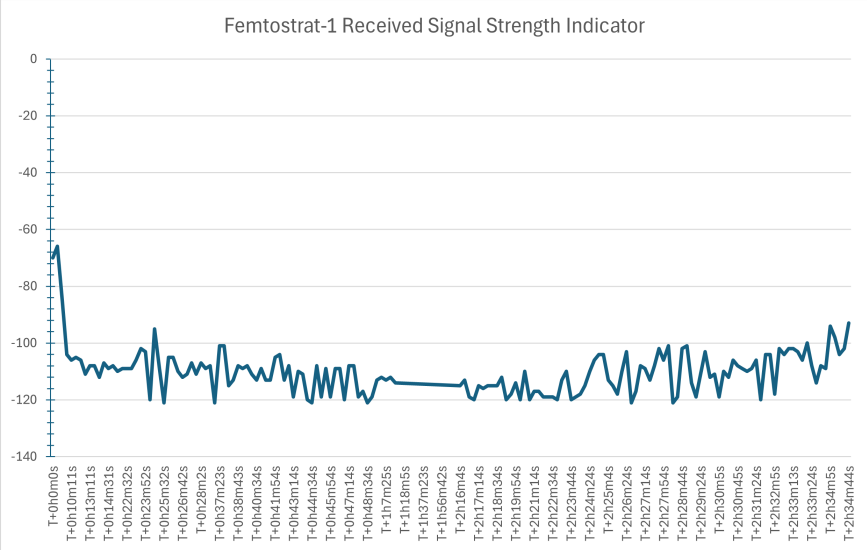

Fig. 6.9: Femtostrat-1 Received Signal Strength Indicator

was another unknown packetbroker, with an SNR of -3.8 dB but an RSSI of -119 dB.

In figure 6.10, a summary of receiving gateways can be seen as a map of points, with Femtostrat-1 in the centre. These were the gateways receiving at one of the highest points in flight, approximately three minutes after the balloon burst. Notable locations include Smilovice near Třinec at 192 km distance and Busko-Zdrój at 163 km. Data provided by the meteorological sonde before the balloon burst was also collected to form a flight-path which can be seen in figure 6.11.

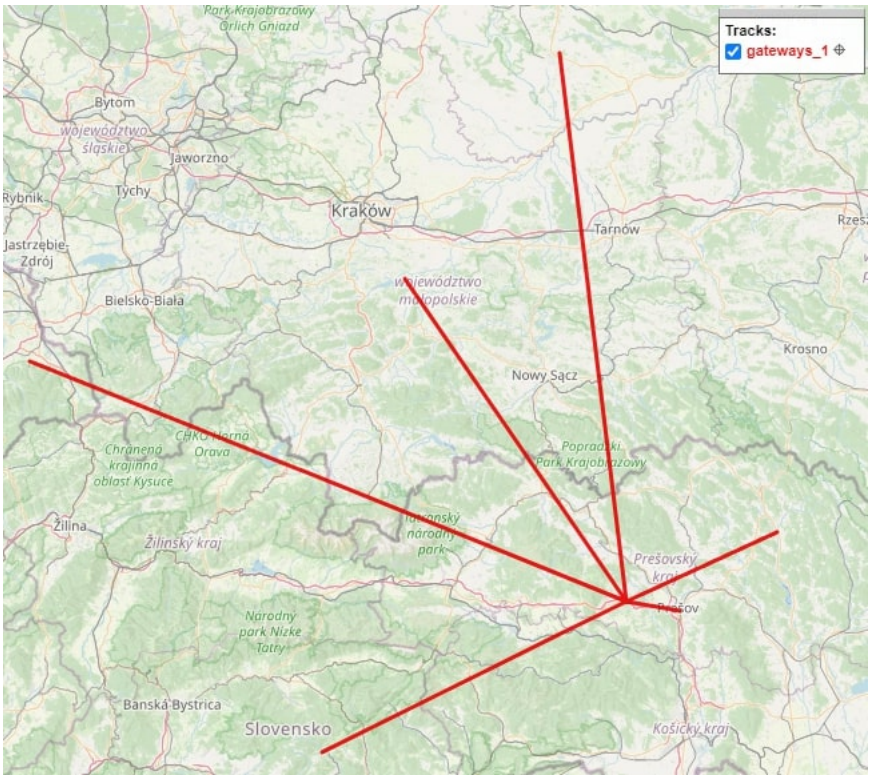

Fig. 6.10: Receiving gateways at the highest point in flight

Fig. 6.11: Flight path formed from meteorological sonde tracker data

6.4.1 Known issues

Following the testing flight of the FM, several issues were identified with Femtostrat-1. Most of these issues were found in the firmware of the satellite.

The two main issues of the flight model were the non-functioning barometer and GNSS receiver. The BME280 board was expected to respond on either the I2C adress of 0x76 or 0x77. Instead, the adress detected was 0xEC. It was later detected

that the incoming bit were shifted incorrectly on either the MCU or barometer side. This also caused the Bosch driver for the STM32 to behave unexpectedly. After the reading and writing function was fixed, correct responses were incoming from the barometer, including chip ID and telemetry. However, the BME280 chip was stuck in sleep mode. Repeated attempts at changing the mode register of the barometer were unsuccessful, despite confirmation of communication. A replacement barometer could not be procured in time for the flight.

The GNSS receiver was successfully communicating full NMEA format sentences to the UART serial port of the MCU. This was verified by passing this data onto the debug UART. The received data can be seen in listing 3. However, the sentences did not contain any location data or satellite information. The expected cause was a low signal environment. However, even after the launch, no location data was received at high altitudes and during good satellite visibility. The cause of this is unknown. However, a possible cause could be an incorrect soldering of the GNSS antenna. The pin of the antenna was too short for the board and the solder might have not reached through to it.

Last, the aforementioned ground station software crashing issue caused the loss of several telemetry packets. The issue was caused by an incorrect handling of received .json objects that might not have been received in their full form which the software did not account for.

```
$GNVTG,,,,,,,,N*2E
$GNGGA,,,,,,0,00,99.99,,,,,,*56
$GNGSA,A,1,,,,,,,,,,,,,99.99,99.99,99.99,1*33
$GNGSA,A,1,,,,,,,,,,,,,99.99,99.99,99.99,3*31
$GNGSA,A,1,,,,,,,,,,,,,99.99,99.99,99.99,4*36
$GNGSA,A,1,,,,,,,,,,,,,99.99,99.99,99.99,5*37
$GNGLL,,,,,,V,N*7A
$GNRMC,,V,,,,,,,,,N,V*37
$GNVTG,,,,,,,,N*2E
$GNGGA,,,,,,0,00,99.99,,,,,,*56
$GNGSA,A,1,,,,,,,,,,,,,99.99,99.99,99.99,1*33
$GNGSA,A,1,,,,,,,,,,,,,99.99,99.99,99.99,3*31
$GNGSA,A,1,,,,,,,,,,,,,99.99,99.99,99.99,4*36
$GNGSA,A,1,,,,,,,,,,,,,99.99,99.99,99.99,5*37
$GNGLL,,,,,,V,N*7A
$GNRMC,,V,,,,,,,,,N,V*37
$GNVTG,,,,,,,,N*2E
$GNGGA,,,,,,0,00,99.99,,,,,,*56
$GNGSA,A,1,,,,,,,,,,,,,99.99,99.99,99.99,1*33
$GNGSA,A,1,,,,,,,,,,,,,99.99,99.99,99.99,3*31
$GNGSA,A,1,,,,,,,,,,,,,99.99,99.99,99.99,4*36
$GNGSA,A,1,,,,,,,,,,,,,99.99,99.99,99.99,5*37
$GPGSV,4,1,16,03,,,08,06,,,08,08,,,08,09,,,09,1*63
```

Listing 3: Example GNSS debug output

Conclusion

This thesis has researched the applicability of femto- and pico- satellites in education. It also provided a description of previous missions as well as concepts of femtosatellite usage, such as swarms or long-distance missions. Additionally, the specific usage of femtosatellites in laboratory exercises was described, with particular focus on the practical knowledge that can be provided through their use.

The next chapters provided a thorough description of the proposed mission called Femtostrat-1 to verify the theoretical applicability of a femtosatellite as a tool for university education. The available launch methods were outlined and their trade-off analysis was provided. The block diagram of the drafted femtosatellite design was shown as well as a detailed look at the design itself. In chapter 4, each component of the femtosatellite was explained with particular emphasis on the power system, on-board computer, and payload modules. For each module, the electrical, software and physical characteristics were described. Additionally, the procedure of designing the prototype was outlined. Last, the ground station hardware and software was described.

After the descriptory chapters, the implementation steps for the engineering model, proto-flight model, and the final flight model were shown. For each model, the design, fabrication, their usage, and results were detailed. The testing process was explained with a description of results found during the testing process. The final state of the firmware was shown including an overview of the operation of Femtostrat-1. In addition, the issues found during development were outlined. At the end of the chapter, the general concept of operations was shown.

In the last chapter, the launch of Femtostrat-1 flight model was detailed. The pre-flight, flight and post-flight phases were explained. The outline of events that occurred during each phase, especially the flight itself, was provided. Detailed results in form of graphs, listings, and figures were shown.

Based on the contents of this thesis, it can be presumed that its objective has been fulfilled. The thesis provides a functional and flight-tested hardware capable of transmitting and receiving data. The overall cost of the hardware, including launch costs, was only approximately 1,000 CZK, making it very competitive compared to contemporary educational models. The detailed schematics and firmware provide sufficient data for educating university students on the concepts of space engineering in a low-cost manner.

This thesis might be further expanded. The known issues can be fixed to provide full functionality of the femtosatellite, including full telemetry and positional data. The Femtostrat-1 platform has been designed with extensibility in mind. Therefore, with relatively minor changes to the PCB, additional payloads, such as a camera

module, might be added. Although the LoRaWAN network is an accessible method of communication, issues described in this thesis might be resolved through the use of a custom LoRa communication algorithm or the usage of TinyGS ground stations.

With the extensions, fixes, and a possible long-term cooperation with a high-altitude balloon provider, the Femtostrat-1 platform can be a useful asset for providing practical experience for space engineering students.

Bibliography

[1] California Polytechnic State University. *CubeSat Design Specification Rev. 14.1*. Cal Poly, 2022.

[2] Erik Kulu. Nanosats database. URL: https://www.nanosats.eu/.

[3] NASA. Cubesat 101: Basic concepts and processes for first-time cubesat developers. URL: https://www.nasa.gov/wp-content/uploads/2017/03/nasa_csli_cubesat_101_508.pdf.

[4] SatCatalog. Cubesat launch costs, Apr 2024. URL: https://www.satcatalog.com/insights/cubesat-launch-costs/.

[5] ESA. Cubesats - fly your satellite! URL: https://www.esa.int/Education/CubeSats_-_Fly_Your_Satellite.

[6] Luis Izquierdo and Joshua Tristancho. Next generation of sensors for femtosatellites based on commercial-of-the-shelf. In *2011 IEEE/AIAA 30th Digital Avionics Systems Conference*, pages 8A4–1–8A4–7, 2011. doi:10.1109/DASC.2011.6096139.

[7] Mercedes Martinez, Andrew Warren, Aman Chandra, and Jekan Thanga. *SunCube FemtoSat Design Specifications (SFDS)*. Arizona State University, 2017.

[8] Álvaro Díaz-Flores, José Fernández, Leonard Vance, Himangshu Kalita, and Jekan Thangavelautham. Femtosats for exploring permanently shadowed regions on the moon. In *2021 IEEE Aerospace Conference (50100)*, pages 1–9. IEEE, 2021.

[9] Elizabeth Simpson. Cornell chronicle: Cu satellites depart on endeavour's last run. URL: https://web.archive.org/web/20121209021155/http://www.news.cornell.edu/stories/April11/endeavoursatellite.html.

[10] Herbert J. Kramer. Kicksat nanosatellite mission. URL: https://web.archive.org/web/20140516065644/https://directory.eoportal.org/web/eoportal/satellite-missions/content/-/article/kicksat.

[11] Cornell University. Swarm of 105 tiny sprite chipsats successfully deployed, Jun 2019. URL: https://newatlas.com/sprite-chipsat-swarm-deployed/59994/.

[12] Scott Seckel. The next big thing in space is really, really small. URL: https://news.asu.edu/20160406-creativity-asu-suncube-femtosat-space-exploration-for-everyone.

[13] Himangshu Kalita, Leonard Dean Vance, Vishnu Reddy, and Jekan Thangavelautham. Use of laser beams to configure and command spacecraft swarms.

[14] ESERO. Cansat - postav si vlastní satelit!, 2023. URL: https://esero.spaceacademy.cz/projekty/cansat/.

[15] Spacelab next door. Spacelab next door store page. URL: https://spacelabnextdoor.com/.

[16] Kaymont. Hab-200. URL: https://www.kaymont.com/product-page/hab-200.

[17] Energizer. Energizer l92 datasheet, 12 2022. URL: https://data.energizer.com/pdfs/l92.pdf.

[18] Microchip. Microchip mcp1624 datasheet, 2016. URL: https://ww1.microchip.com/downloads/aemDocuments/documents/OTH/ProductDocuments/DataSheets/40001420D.pdf.

[19] Nordic Semiconductor. nrf24l01 datasheet, 03 2008. URL: https://www.sparkfun.com/datasheets/Components/SMD/nRF24L01Pluss_Preliminary_Product_Specification_v1_0.pdf.

[20] Semtech. Sx127x datasheet, 05 2020. URL: https://semtech.my.salesforce.com/sfc/p/#E0000000JelG/a/2R0000001Rc1/QnUuV9TviODKUgt_rpBlPz.EZA_PNK7Rpi8HA5..Sbo.

[21] Seeed Studio. Wio-e5 datasheet, 2023. URL: https://files.seeedstudio.com/products/317990687/res/LoRa-E5%20module%20datasheet_V1.1.pdf.

[22] Arm. Arm cortex m4 datasheet, 2020. URL: https://developer.arm.com/documentation/102832/latest/.

[23] ST Microelectronics. Stm32wle5j8/jb/jc datasheet, 02 2020. URL: https://files.seeedstudio.com/products/317990687/res/STM32WLE5JC%20Datasheet.pdf.

[24] Semtech. Sx1268 datasheet, 06 2019. URL: https://semtech.my.salesforce.com/sfc/p/#E0000000JelG/a/3n000000qOUu/tpcW.qidS7dE9fxMVN_v5okCY64Xiq5OUeNNFal1a8c.

[25] LoRa Alliance. What is lorawan® specification, Dec 2023. URL: https://lora-alliance.org/about-lorawan/.

[26] Rishabh Mitra and Raghavendra Ganiga. A novel approach to sensor implementation for healthcare systems using internet of things. *International Journal of Electrical and Computer Engineering (IJECE)*, 9:5031, 12 2019. doi:10.11591/ijece.v9i6.pp5031-5045.

[27] InvenSense. Mpu-6000 and mpu-6050 product specification, 08 2013. URL: https://invensense.tdk.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf.

[28] Components101. Mpu6050 accelerometer and gyroscope module, Mar 2021. URL: https://components101.com/sensors/mpu6050-module.

[29] Bosch. Bme280 combined humidity and pressure sensor, 09 2018. URL: https://www.mouser.com/datasheet/2/783/BST-BME280-DS002-1509607.pdf.

[30] uBlox. Max-m10s datasheet, 03 2024. URL: https://content.u-blox.com/sites/default/files/MAX-M10S_DataSheet_UBX-20035208.pdf.

[31] Seeed. Wio-e5 mini: Seeed studio wiki. URL: https://wiki.seeedstudio.com/LoRa_E5_mini/.

[32] Seeed-Studio. Seeed-studio/lorawan-e5-node. URL: https://github.com/Seeed-Studio/LoRaWan-E5-Node/tree/qian.

[33] Androbi-Com. Androbi-com/lora-e5-mini-endnode. URL: https://github.com/androbi-com/LoRa-E5-Mini-EndNode.

[34] Embedded Micro Technology Ltd. Mypi industrial raspberry pi uart1 (ttys0) configuration. URL: https://www.embeddedpi.com/documentation/com-ports/mypi-industrial-raspberry-pi-uart1-ttys0-configuration.

[35] Tom Nardi. The future of space is tiny, Jun 2019. URL: https://hackaday.com/2019/06/25/the-future-of-space-is-tiny/#more-362858.

[36] Josh Berk, Jeremy Straub, and David Whalen. The open prototype for educational nanosats: Fixing the other side of the small satellite cost equation. In *2013 IEEE Aerospace Conference*, pages 1–16. IEEE, 2013.

[37] LoRa Alliance Technical Commitee. Lorawan 1.1 specification, 2017. URL: https://resources.lora-alliance.org/technical-specifications/lorawan-specification-v1-1.

[38] Luboš M. Senzor bme280 - měření teploty, relativní vlhkosti a barometrického tlaku, Jan 2017. URL: https://navody.dratek.cz/navody-k-produktum/

senzor-bme280-mereni-teploty-relativni-vlhkosti-a-barometrickeho-tlaku.html.

[39] Gy-521 mpu6050 3-axis acceleration gyroscope. URL: https://www.hotmcu.com/gy521-mpu6050-3axis-acceleration-gyroscope-6dof-module-p-83.html.

Symbols and abbreviations

I2C	Inter-Integrated Circuit
OBC	On-Board Computer
EPS	Electric Power System
COM	Communication System
ISS	International Space Station
ESA	European Space Agency
NASA	National Aeronautics and Space Administration
LoRa	Long Range - Proprietary modulation technique
LoRaWAN	Long Range Wide Area Network
LPWAN	Low Power Wide Area Network
GNSS	Global Navigation Satellite Systems
GPS	Global Positioning System
ASU	Arizona State University
CalPoly	California Polytechnic State University
COTS	Commercial off-the-shelf
BMS	Battery Management System
THT	Through-hole Technology
SMD	Surface Mount Device
MOSFET	Metal Oxide Semiconductor Field Effect Transistor
DC	Direct Current
RX	Receiving
TX	Transmission
ISM	Industrial, Scientific, and Medical band
WOR	Wake-up On Receive
SMT	Surface-mount Technology
HSE	High Speed External (oscillator)
LSE	Low Speed External (oscillator)
RISC	Reduced Instruction Set Computer
AT	ATtention (commands)
RAM	Random Access Memory
IoT	Internet of Things
ADC	Analog-to-Digital Converter
MEMS	Micro Electro Mechanical System
DMP	Digital Motion Processor
SPI	Serial Peripheral Interface
LDO	Low-Dropout Regulator
UART	Universal Asynchronous Receiver-Transmitter
USART	Universal Synchronous/Asynchronous Receiver-Transmitter
PFM	Prototype Flight Model
FM	Flight Model
EM	Engineering Model
PCB	Printed Circuit Board
LED	Light Emitting Diode
GPIO	General Purpose Input Output
FTDI	Future Technology Devices International ltd.
HASL	Hot Air Solder Leveling

VZLÚ	Czech Aerospace Research Centre
TTN	The Things Network
LPP	Low Power Payload
RTOS	Real Time Operating System
MCU	Microcontroller Unit
SD	Secure Digital
LAN	Local Area Network
RPi	Raspberry Pi
MQTT	MQ Telemetry Transport
SNR	Signal-to-Noise Ratio
RSSI	Received Signal Strength Indicator
UI	User Interface
GUI	Graphical User Interface
UTC	Universal Time Coordinated
NMEA	National Marine Electronics Association

List of appendices

A Satellite classes

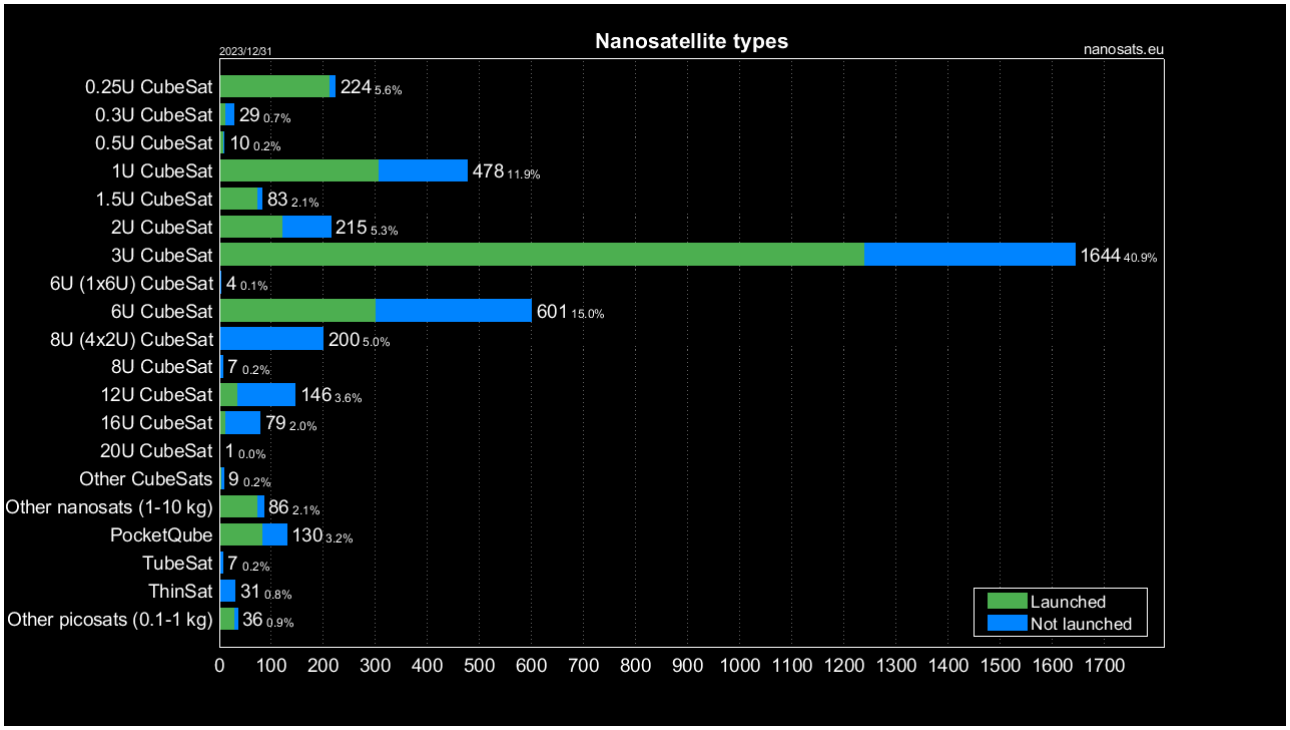

Fig. A.1: Nanosatellite launches by size. - Reprinted from: nanosats.eu[2]

B Femtostrat-1 Components

B.1 Payload

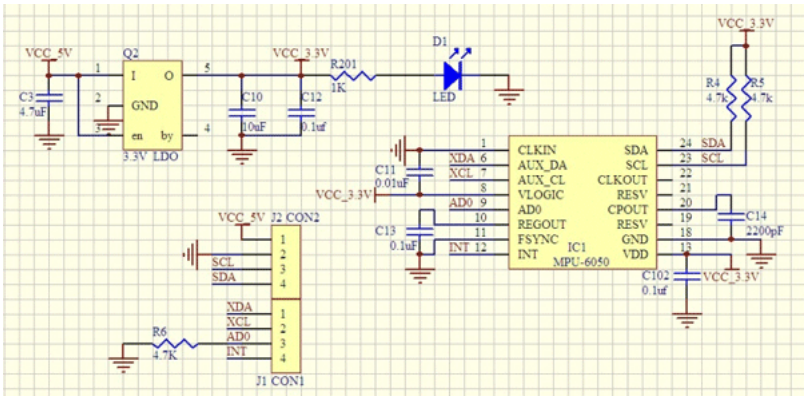

Fig. B.1: GY-521 module schematic - reprinted from hotmcu.com[39]

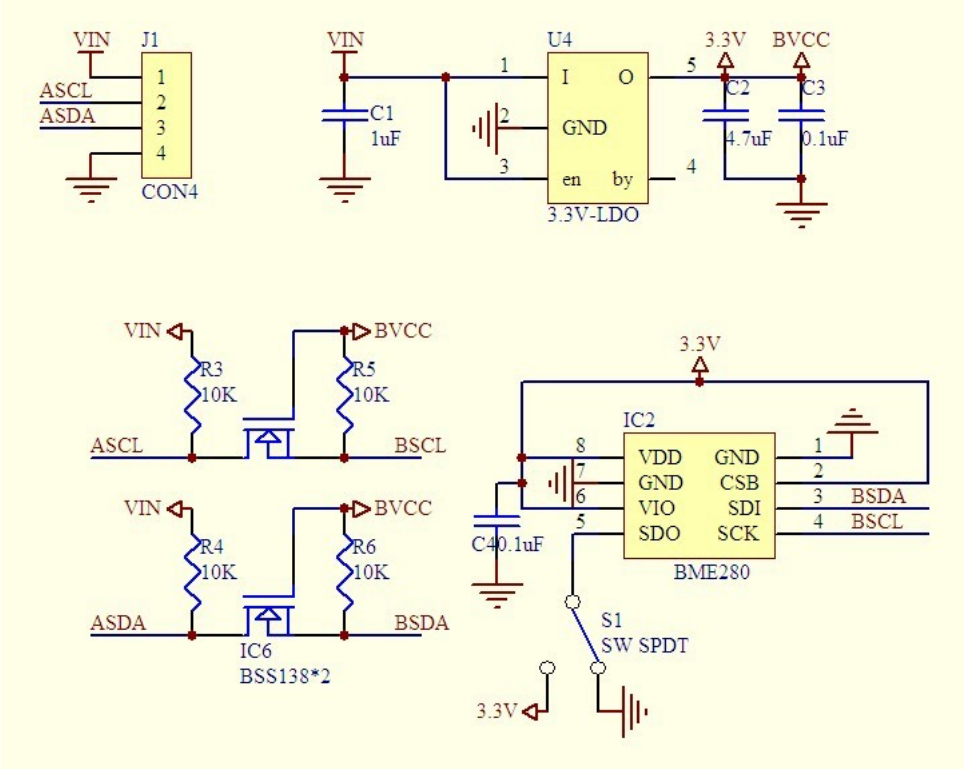

Fig. B.2: BME-280 module breakout board schematic - reprinted from dratek.cz[38]

B.2 Component Properties

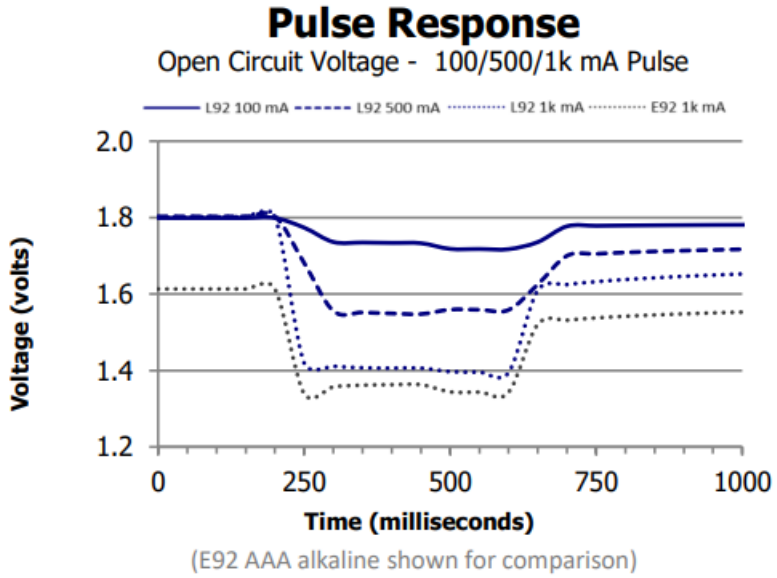

Fig. B.3: L92 pulse response - Reprinted from L92 datasheet[17]

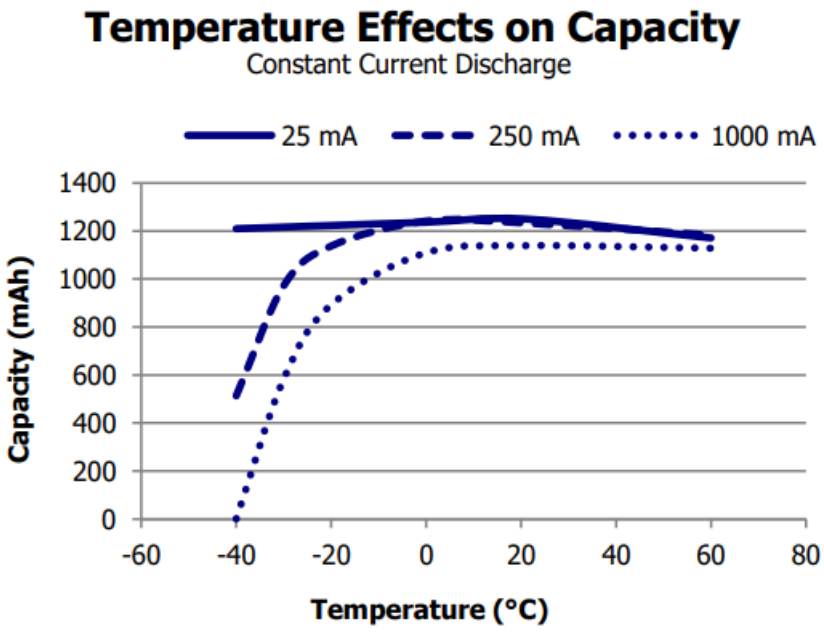

Fig. B.4: L92 Temperature effects on capacity - Reprinted from L92 datasheet[17]

C Engineering Model

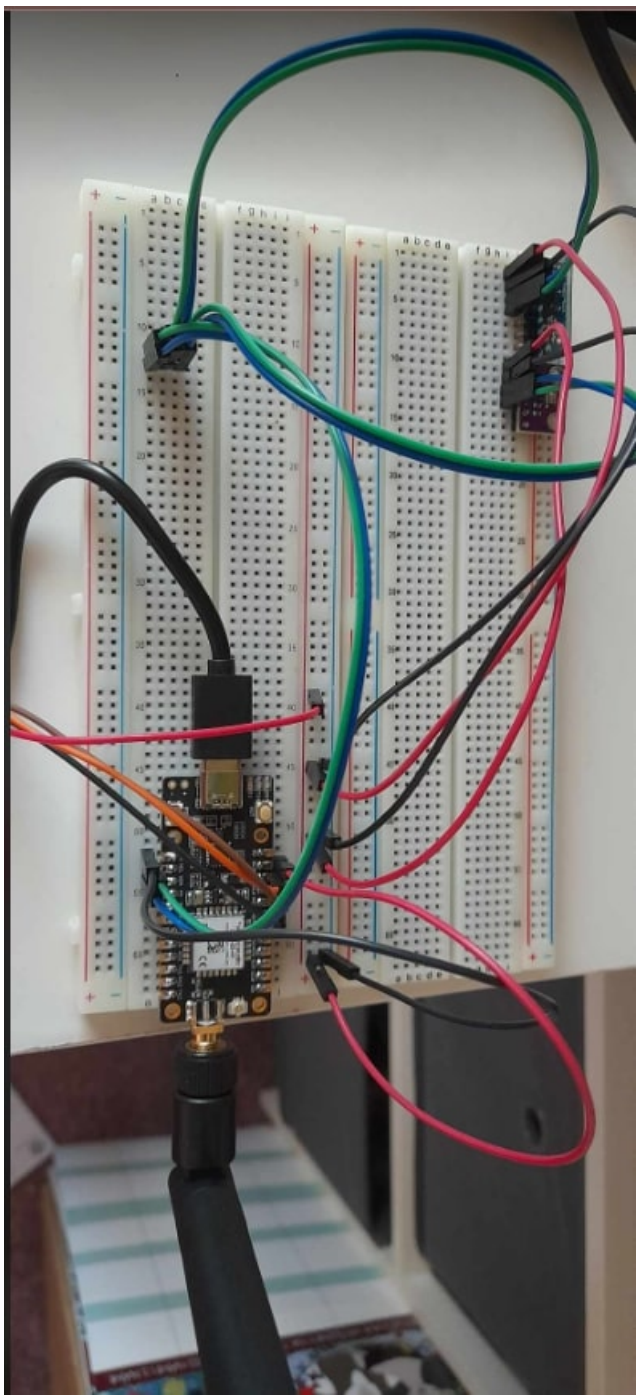

Fig. C.1: Femtostrat-1 engineering model

D Flight Model

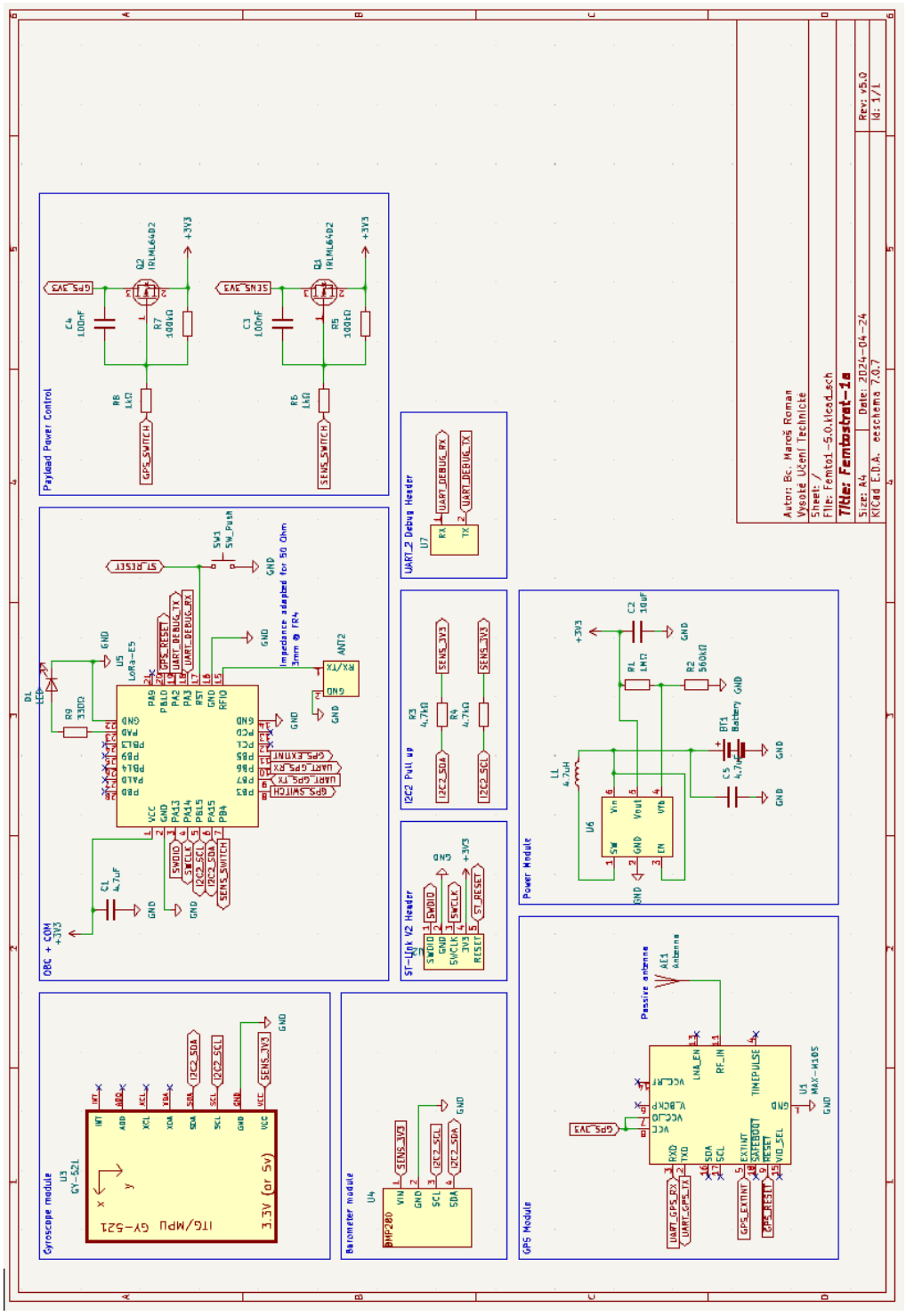

Fig. D.1: Flight model full circuit schematic

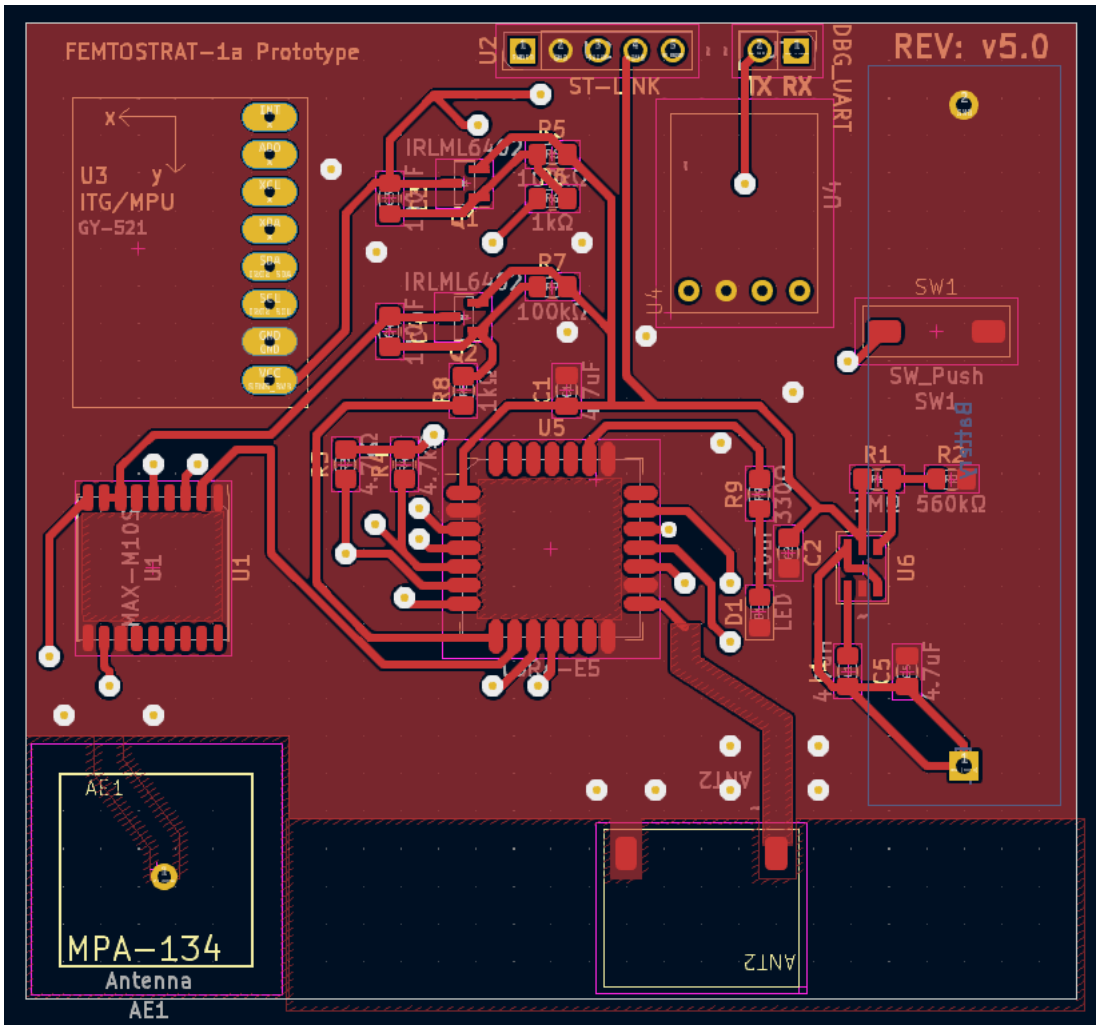

Fig. D.2: Flight model PCB drawing front

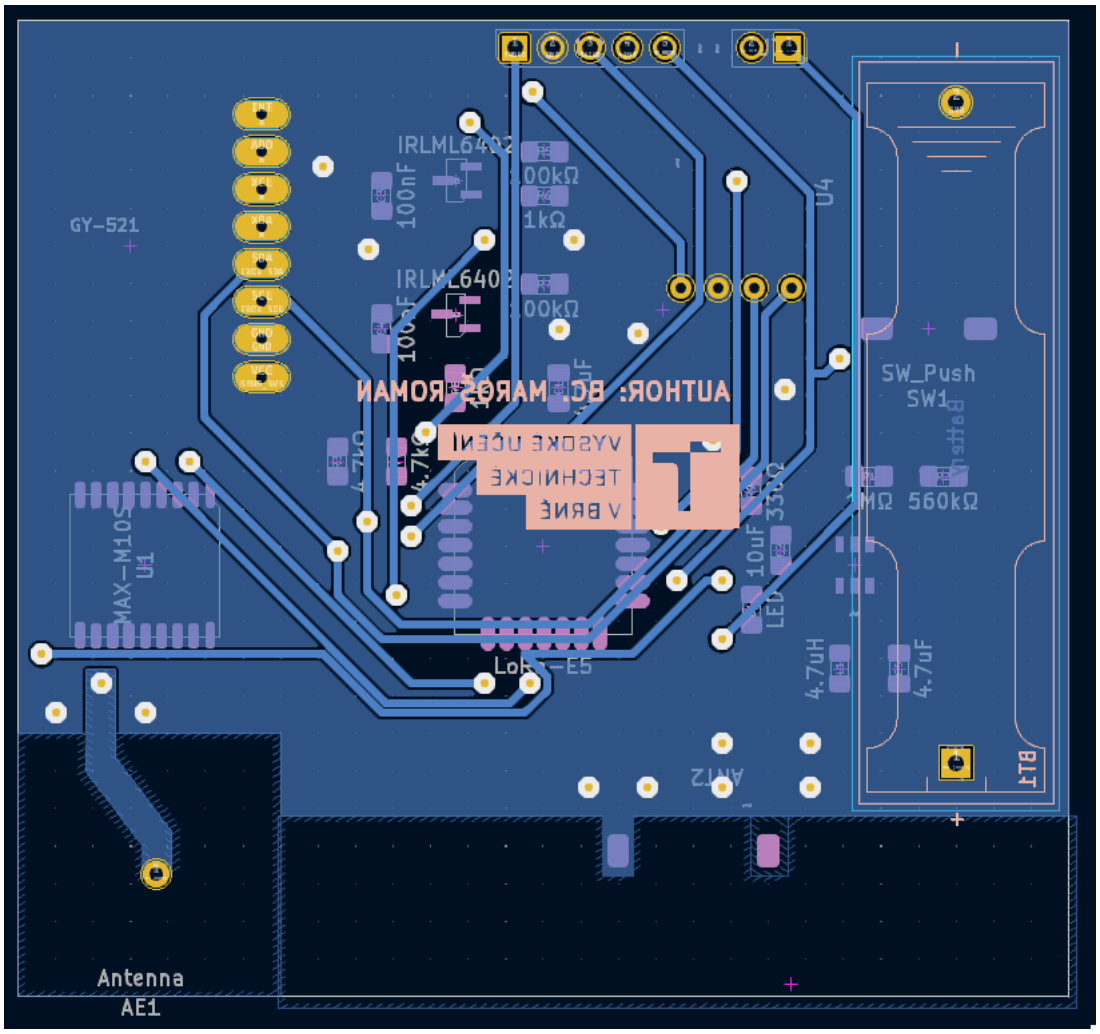

Fig. D.3: Flight model PCB drawing back

E Ground Station

Fig. E.1: Ground station gateway in protective cover.

F Firmware

```c
/*
 * note: The sensor and GNSS switches begin as switched on, so it is no necessary to toggle
 * However commented out below are functions to switch them on in case this is not true.
 */
// note: do not use the APP_LOG macro as the UART it is writing to is used for the GNSS module
debug_print("\r\n STARTING SENSORS...\r\n");
//HAL_GPIO_TogglePin(DBG4_GPIO_Port, DBG4_Pin);
debug_print("\r\n SENSORS STARTED!\r\n");
debug_print("\r\n STARTING GNSS...\r\n");
// Toggle GNSS reset pin
HAL_GPIO_TogglePin(DBG2_GPIO_Port, DBG2_Pin);
//HAL_GPIO_TogglePin(DBG3_GPIO_Port, DBG3_Pin);
debug_print("\r\n GNSS STARTED!\r\n");

loraSetI2CHandle(&hi2c2);
loraSetUARTHandle(&huart1);
debug_print("\r\n INITIALISING GYROSCOPE...\r\n");
while (MPU6050_Init(&hi2c2) == 1);
debug_print("\r\n GYROSCOPE INITIALISED!\r\n");

uint8_t devices = 0u;

debug_print("Searching for I2C devices on the bus...\n");
/* Values outside 0x03 and 0x77 are invalid. */
for (uint8_t i = 0x03u; i < 0x78u; i++)
{
        uint8_t address = i << 1u ;
        /* In case there is a positive feedback, print it out. */
        if (HAL_OK == HAL_I2C_IsDeviceReady(&hi2c2, address, 3u, 10u))
        {
                debug_print("Device found: 0x%02X\n", address);
                devices++;
        }
}
/* Feedback of the total number of devices. */
if (0u == devices)
{
        debug_print("No device found.\n");
}
else
{
        debug_print(TS_OFF, VLEVEL_M,"Total found devices: %d\n", devices);
}
HAL_GPIO_TogglePin(DBG1_GPIO_Port, DBG1_Pin);
/* USER CODE END 2 */

/* Infinite loop */
/* USER CODE BEGIN WHILE */
while (1)
{
        /* USER CODE END WHILE */
        MX_LoRaWAN_Process();

        /* USER CODE BEGIN 3 */
}
```

Listing 4: Femtostrat-1 initialisation - main.c

```c
/*
 * This function takes a string and variadic arguments to print out
 * the message to UART2
 */
void debug_print(const char *str, ...)
{
        va_list argptr;
        va_start(argptr, str);

        char form_str[256];
        vsnprintf(form_str, 256, str, argptr);
        va_end(argptr);
        HAL_GPIO_TogglePin(DBG1_GPIO_Port, DBG1_Pin);
        HAL_UART_Transmit(&huart2, (uint8_t *)str, strlen(str), 1000);
        HAL_Delay(100);
        HAL_GPIO_TogglePin(DBG1_GPIO_Port, DBG1_Pin);
}

/*
 * This function uses the system function for gathering the
 * voltage from the associated ADC register of the MCU
 */
uint16_t get_voltage()
{
        return (uint16_t) SYS_GetBatteryLevel();
}

/*
 * This function uses the system function for gathering the
 * temperature from the associated ADC register of the MCU
 */
float get_temp()
{
        uint16_t temp = SYS_GetTemperatureLevel();
        float celsius = temp;
        debug_print("Temp: %d", temp);
        return temp & 0xFFF;
}

```

Listing 5: Femtostrat-1 utility functions - femto_utils.c

```
1
2                   case LORAWAN_USER_APP_PORT:
3                           if (appData->BufferSize > 0)
4                           {
5                                   // Confirm transmission visually
6                                   if (appData->Buffer[0] == 0x01) {
7                                           for (int i = 0; i < 20; i++) {
8                                                   HAL_GPIO_TogglePin(DBG1_GPIO_Port, DBG1_Pin);
9                                                   HAL_Delay(500);
10                                          }
11                                  }
12                                  // Switch to plaintext broadcast mode
13                                  if (appData->Buffer[0] == 0x02) {
14                                          cayenneMode = 0;
15                                  }
16                                  // Switch to Cayenne LPP mode
17                                  if (appData->Buffer[0] == 0x03) {
18                                          cayenneMode = 1;
19                                  }
20                                  // Repeat buffer content
21                                  if (appData->Buffer[0] == 0x04) {
22                                          setTransmit = 1;
23                                          memcpy(transmitBuff, (appData->Buffer)+1, appData->BufferSize-1);
24                                  }
25                                  // Toggle Sensors
26                                  if (appData->Buffer[0] == 0x05) {
27                                          HAL_GPIO_TogglePin(DBG4_GPIO_Port, DBG4_Pin);
28                                          if (sensorsOn) {
29                                                  sensorsOn = 0;
30                                          } else {
31                                                  sensorsOn = 1;
32                                          }
33                                  }
34                                  // Toggle GNSS
35                                  if (appData->Buffer[0] == 0x06) {
36                                          HAL_GPIO_TogglePin(DBG3_GPIO_Port, DBG3_Pin);
37                                          if (GNSSOn) {
38                                                  GNSSOn = 0;
39                                          } else {
40                                                  GNSSOn = 1;
41                                          }
42                                  }
43                                  if (appData->Buffer[0] == 0x07) {
44                                          HAL_NVIC_SystemReset();
45                                  }
46                          }
47                          break;
48
```

Listing 6: Femtostrat-1 command processing - lora_app.c

```
1
2           // Init GNSS
3           debug_print("\r\n INITIALISING GNSS...\r\n");
4           GNSS_StateHandle GNSS_Handle;
5           GNSS_Init(&GNSS_Handle, &uartHandle);
6           HAL_Delay(1000);
7           GNSS_LoadConfig(&GNSS_Handle);
8           GNSS_GetUniqID(&GNSS_Handle);
9           GNSS_ParseBuffer(&GNSS_Handle);
10          HAL_Delay(250);
11          GNSS_GetPVTData(&GNSS_Handle);
12          GNSS_ParseBuffer(&GNSS_Handle);
13          HAL_Delay(250);
14          GNSS_SetMode(&GNSS_Handle, Portable);
15          HAL_Delay(250);
16          debug_print("\r\n GNSS Initialized!\r\n");
17
18
19
20          int32_t bmp280_temp = 0;
21          uint32_t bmp280_pres = 0;
22          uint32_t bmp280_hum = 0;
23
24          // Read Gyro and Baro
25          debug_print("\r\n Reading Gyro...\r\n");
26          MPU6050_Read_All(&i2CHandle, &MPU6050);
27          debug_print("\r\n Reading Baro...\r\n");
28          uint8_t power_mode = 0x03;
29          bme280_data_readout_template(&bmp280_temp, &bmp280_pres, &bmp280_hum);
30
31          AppData.Port = LORAWAN_USER_APP_PORT;
32          debug_print("BMP280 OUTPUT: TEMP: %d PRES: %d HUM: %d\r\n", bmp280_temp, bmp280_pres, bmp280_hum);
33
34          if (cayenneMode) {
35
36                  // form Cayenne message
37                  CayenneLppReset();
38                  CayenneLppAddTemperatureFloat(channel++, get_temp());
39                  CayenneLppAddTemperatureFloat(channel++, bmp280_temp);
40          ...
41
42          ...
43                  CayenneLppCopy(AppData.Buffer);
44                  AppData.BufferSize = CayenneLppGetSize();
45          } else if (setTransmit) {
46                  memcpy(AppData.Buffer, transmitBuff, 255);
47                  AppData.BufferSize = (uint8_t)255;
48                  setTransmit = 0;
49          } else {
50                  /* not CAYENNE_LPP */
51                  // Form non-cayenne message
52                  char *msg = "This is a transmission from Femtostrat-1, a prototype stratospheric satellite from Brno! ";
53                  size_t length = strlen(msg);
54                  memcpy(AppData.Buffer, msg, length);
55                  AppData.BufferSize = length;
56                  if (sensorsOn) {
57                          char *state = "Sensors ON! ";
58                          memcpy(AppData.Buffer+AppData.BufferSize, state, strlen(state));
59                          AppData.BufferSize += strlen(state);
60                  } else {
61                          char *state = "Sensors OFF! ";
62                          memcpy(AppData.Buffer+AppData.BufferSize, state, strlen(state));
63                          AppData.BufferSize += strlen(state);
64                  }
65                  if (GNSSOn) {
66                          char *state = "GNSS ON! ";
67                          memcpy(AppData.Buffer+AppData.BufferSize, state, strlen(state));
68                          AppData.BufferSize += strlen(state);
69                  } else {
70                          char *state = "GNSS OFF! ";
71                          memcpy(AppData.Buffer+AppData.BufferSize, state, strlen(state));
72                          AppData.BufferSize += strlen(state);
73                  }
74          }
75
```

Listing 7: Femtostrat-1 transmission preparation - lora_app.c

G Launch

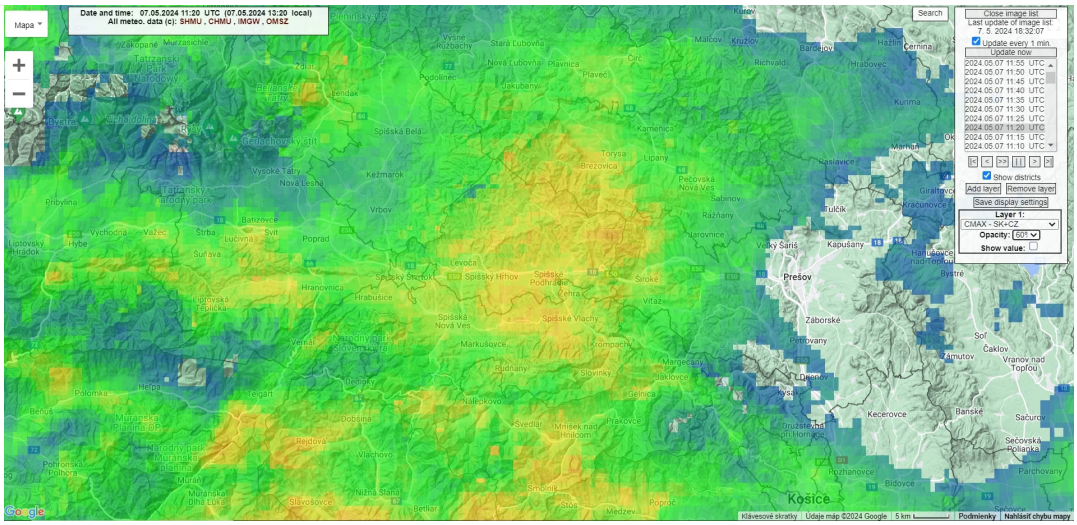

Fig. G.1: Radar report at launch - Credit: SHMÚ

Fig. G.2: Femtostrat-1 integration

Fig. G.3: Femtostrat-1 launch (approximate altitude >100m)

Fig. G.4: Femtostrat-1 Landing site - Credit: Marián Lukačko

Fig. G.5: Femtostrat-1 as landed with meteosonde - Credit: Marián Lukačko

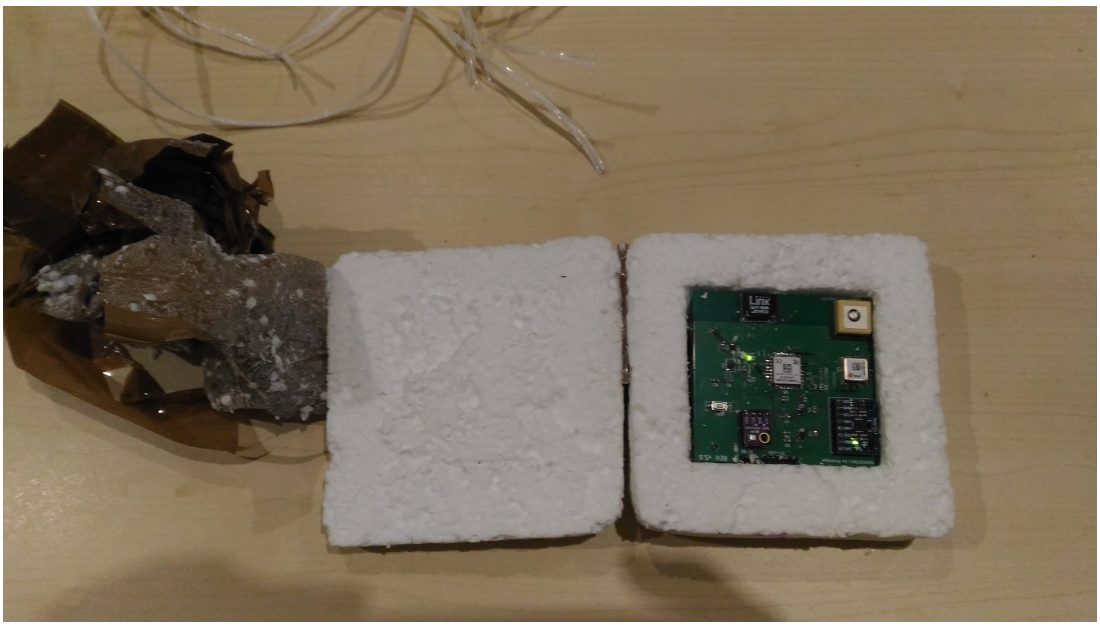

Fig. G.6: Femtostrat-1 as extracted after landing - Credit: Marián Lukačko

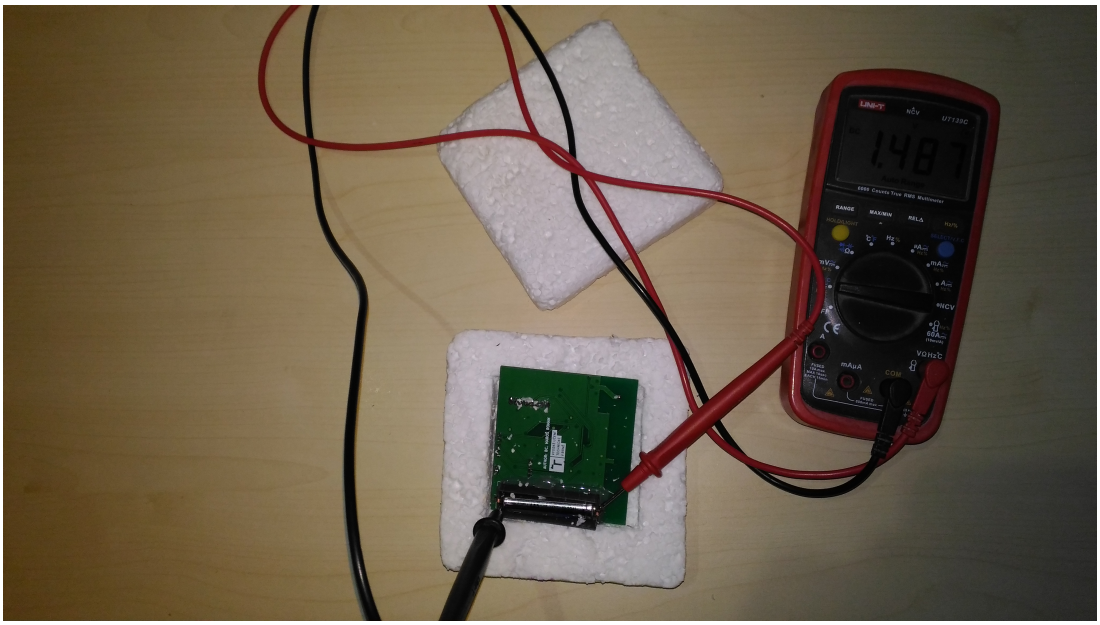

Fig. G.7: Femtostrat-1 post-flight voltage check - Credit: Marián Lukačko

H Results

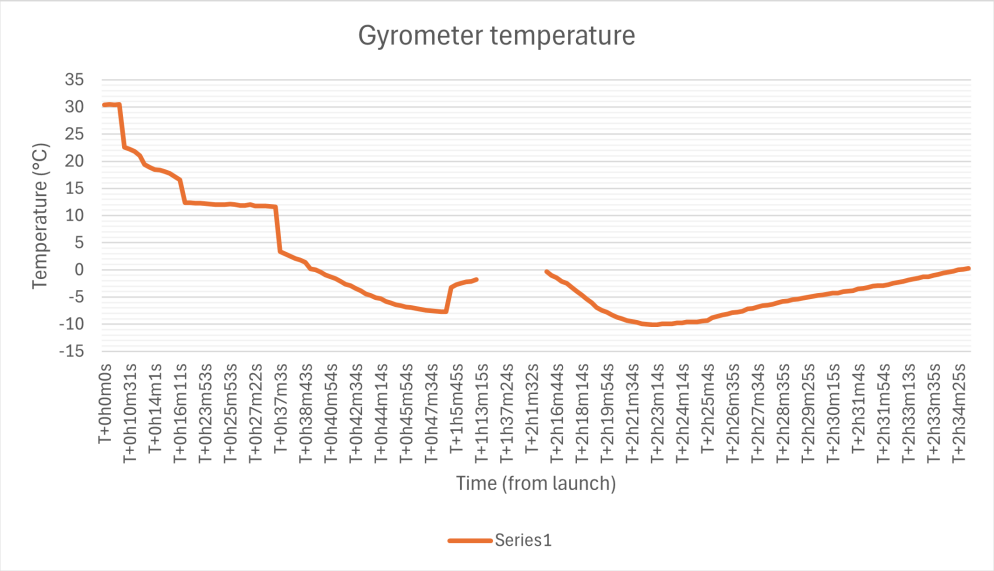

Fig. H.1: Gyroscope temperature graph