Aerospace Europe Conference 2023 - 10th EUCASS - 9th CEAS

DOI: ADD DOINUMBER HERE

Four Months to Orbit: Fast-Tracking CubeSat Development for Reliability Through In-Orbit-Demonstrations

Michael Linder[⋆†], *Aziz Belkhiria*[⋆], *Robin Bonny*[⋆], *Joaquim Silveira*[⋆] *Raphaël Temperli*[⋆]
Nicolas Bouron[⋆], *Saverio Nasturzio*[⋆], *Taras Pavliv*[⋆], *Santiago Evangelista*[⋆], *Rico G. Fausch*[‡]

[⋆]*EPFL Spacecraft Team*
Lausanne, Switzerland

[‡]*University of Bern*
Bern, Switzerland

michael@dphispace.com · aziz@dphispace.com · robin.bonny@alumni.epfl.ch
joaquim.silveira@alumni.epfl.ch · raphael.temperli@epfl.ch · nicolas.bouron@alumni.epfl.ch
saverio.nasturzio@alumni.epfl.ch · taras.pavliv@alumni.epfl.ch · santiago.evangelista@epfl.ch
rico.fausch@unibe.ch

[†]Corresponding author

Abstract

A consortium led by the EPFL Spacecraft Team is developing the Constellation of High Energy Swiss Satellites (CHESS) mission, which will launch two 3U CubeSats to low Earth orbit in 2025 to analyze the absolute number density profiles of the chemical composition, their dynamics, and the total electron content in situ. To reduce subsystem risks and increase reliability, the team conducted an in-orbit demonstration referred to as the *Bunny* mission, launching an onboard computer demonstrator to low Earth orbit in January 2023. This study outlines the architecture of the payload, the technical challenges faced, and the systems engineering approach. In-orbit demonstrations are shown to be effective in mitigating risks for the mission whilst providing educational value. Agile methods, including in-orbit software updates, are shown to be compatible with such missions, and test results show the hardware's functionality after five months in orbit. The project's success highlights students' ability to contribute to space technology and encourages others to do the same. This de-risking paves the way for the follow-up satellites of CHESS to provide long-awaited data to study the status, origin, and evolution of a habitable atmosphere.

1. Introduction

The EPFL Spacecraft Team (EST) is a student team developing two 3U CubeSats for the Constellation of High Energy Swiss Satellits (CHESS) mission[1] aiming to study Earth's upper atmosphere. The team consists of around 50 bachelor and master students per semester, supported by various laboratories from EPFL and other Swiss institutions, namely the University of Bern, ETHZ, Hochschule Luzern, HES-Arc, and HES-SO. Each CHESS satellite hosts a miniaturized mass spectrometer[2] and a GNSS instrument[3] . Given the scientific payloads on board the CHESS satellites, risk mitigation is the major driver for this mission as opposed to other student-led CubeSat missions. Therefore, critical subsystems like the on board computer (OBC) should be rigorously tested and, ideally, successfully operate in orbit to demonstrate technical readiness level 9, implying flight heritage enabled by an in-orbit demonstration.

In May 2022, the team was offered the opportunity of sending a hosted payload on board of D-Orbit's ION orbit transfer vehicle (OTV). This mission differs from D-Orbit's nominal mission profile as it was on a tight schedule and had a higher risk profile due to a substantial orbit raise never performed by ION before. This is similar to the CHESS requirements, which require the OTV to launch one satellite in an elliptical orbit to measure the altitude profiles of the chemical species. Especially the tight deadline, which meant development, testing, and integration within four months, posed a major challenge to the team. EST took this opportunity and chose to fast develop a new version of their OBC: *Bunny*. Before this mission, the CHESS mission passed preliminary design review (PDR), and first prototypes of an OBC developed in-house were built and tested in a lab. The Bunny OBC, as shown in figure 1, left panel, has

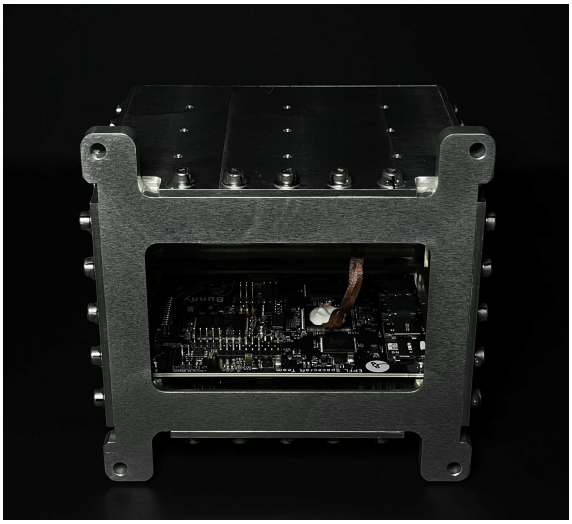

Figure 1: Left: The Bunny payload inside its casing. Right: A render of the Bunny payload hosted on the ION satellite from D-Orbit. The outer dimensions of the box are 140x140x123 mm^3

a similar architecture as the previous prototypes with a microcontroller (MCU) and a Field Programmable Gate Array (FPGA).

2. The Mission

Considering the stringent time constraint associated with the development and integration of Bunny, the management team of EST decided to overcome the conventional waterfall project life cycle usually used for major space missions, as detailed in the NASA Systems Engineering Handbook[4] . Instead, an agile method[5] was adopted to accelerate the development and cope with the schedule. The method consists of an iterative approach that implements a feedback loop between each iteration to propagate learnings and improvements until reaching the flight model, as shown in figure 3.

2.1 Development

After an initial set of mission requirements were defined, the team decided to start directly with developing the first hardware version (v), referred to as Bunny v1. This development was crucial as the printed circuit board (PCB) design, manufacturing, and assembly takes time and was predicted to be the bottleneck of the mission. The goal of Bunny v1 was to identify hardware design problems early and allow software development directly on relevant hardware. Figure 3 shows the design flow. Due to this fast start of hardware development, there were changes between the first and second versions of Bunny, referred to as Bunny v2 (figure 2). A secondary experiment testing four SD cards was added, and the main microcontroller was changed due to supply chain issues. The final design relied on an *STM32L476ZGT3* microcontroller, an *A3P125* FPGA, and *IS64WV1024* and *MB85R1001ANC* memories. These components all have an operating temperature range of at least −40 °C to +85 °C (everything except the FRAM to +125 °C)

After Bunny v1 was electrically tested, software development began. With the stringent time constraints, the software team set to focus on the following core elements:

- Communication with the host spacecraft through the D-Orbit micro-python interface.
- Command interpretation through the Bunny microcontroller.
- Components health-checks.

As the second iteration of Bunny was being developed, the foundation of the software was outlined with an established communication between the MCU of the first iteration and the payload controller. The software team did not consider software updates in the beginning and focused exclusively on the core components of the software. However, it soon became clear that given the minimal testing of the software, there had to be a method that enabled software updates in orbit to allow for continuous improvements of the payload. This would allow the team to start developing stress tests

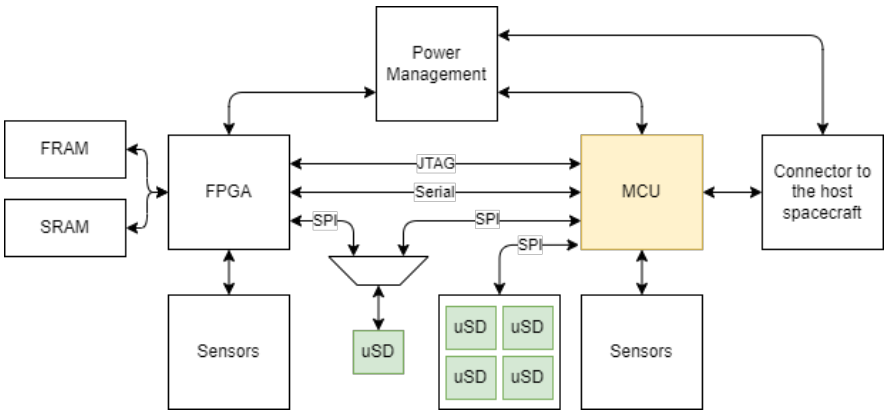

Figure 2: Some components were present on Bunny v1 and v2 (white), some only on v2 (green), and some were changed (yellow).

for the components, which provided deeper insights beyond the health checks. As a result, over-the-air (OTA) update capabilities were implemented, allowing the team to correct bugs and implement new features in orbit.

The responsibilities and development within the team were split as follows:

- **Sensor and SD Cards Drivers:** The main components producing telemetry in the mission. The temperature sensors, magnetometers, and accelerometer drivers were developed from scratch. The SD card library was based on Elm Chan's SPI library for SD cards[6] .
- **Host Interface Communication:** As Bunny is able to generate substantial telemetry data, it is necessary to develop the downlink interfacing through D-Orbit's interface. This process required the careful development of a communication protocol between high-level and low-level software, ensuring smooth data transmission and system performance.
- **FPGA Reprogramming Library:** One of the main focus areas of the software development was to allow for reprogramming the FPGA from the microcontroller on the fly. This required the adaptation and implementation of the *DirectC* library from MicroChip, Chandler, United States.
- **Bootloader Development:** As mentioned before, the short software development time implied minimal time for intensive testing of the software. Hence, in parallel, we developed and implemented a bootloader on the microcontroller, which would allow us to patch any software bug on the main flight software we would encounter in the future. Since the only software module that would not be remotely patchable if a bug was to be identified once in orbit is the bootloader, extensive testing had to be performed on it to minimize the risk of a single point of failure.
- **Core Flight Software:** The development of this module was time-consuming due to the necessity of establishing a secure method for thread communication within the software. Additionally, it was crucial to safely transfer data to and from the host interface. Any corruption in the System-on-Chip file for the FPGA could, in the worst-case scenario, result in short circuits.

The development was concluded with functional tests verifying the successful integration of the software with the hardware. The software was first thoughtfully tested on Bunny v1, followed by tests on the Bunny v2 engineering and flight models. The tests on Bunny v1 were done to reduce the risk of payload malfunction due to coding errors, as only one flight and one engineering model were available at that time.
After all functional tests succeeded, Bunny was ready for environmental acceptance testing.

2.2 Integration and Testing

Even though the team already had extensive experience in technical development from previous projects, the integration and testing procedures designed for this flight mission represented a new challenge to overcome. Given the tight schedule imposed by integration, it was decided to limit environmental testing to the minimum and only perform tests required by the launch provider, in this case, SpaceX, USA. Bunny was considered to be a containerized CubeSat unit,

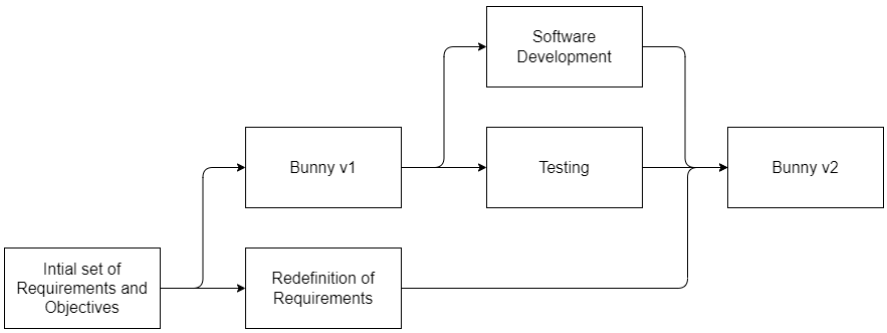

Figure 3: The simplified development flow from an initial set of requirements to the Bunny flight model.

therefore, being bound by the test level and durations presented in table 3-16 of the SpaceX Rideshare User's Guide[7] . Hence, the tests performed include shock, random vibration, combined thermal vacuum and thermal cycle, and static load testing. Acoustic and sinusoidal vibration testing was initially considered not to be required, given the payload's category. Furthermore, testing for pressure systems was not applicable for Bunny, as no such systems were included within the payload, nor was electromagnetic compatibility to be tested, as the payload remained purely passive and shut off during launch and early orbit phases (LEOP). Table 1 summarises the aforementioned tests to be conducted. This specifically applies to proto-qualification for single-unit testing, as fleet qualification was not applicable for Bunny, given the one-off nature of the payload. In October 2022, SpaceX revised the Rideshare User's Guide to further alleviate the testing requirements for containerized CubeSat units, removing the requirement for static load testing. For the testing requirements indicated in table 1, thermal vacuum cycling, shock, random vibration, and static load tests were to be performed on the unit level before integration on D-Orbit's ION spacecraft, where only random vibration was strictly required, with the remaining tests being strongly advised. Combined thermal vacuum and thermal cycling could be performed on-site at EPFL using the test facilities operated by Space Innovation, Lausanne, Switzerland. Mechanical environmental testing was performed at the testing facilities of the Space Research & Planetary Sciences division of the Physics Institute of the University of Bern, Bern, Switzerland.

Prior to launch, SpaceX raised questions regarding the test reports as part of reviewing the provided test documentation. Consequently, the team was asked to rework the contamination compliance and provide further insights on the vibration testing. Therefore, it remains important to keep raw measurement and testing data at hand throughout all development phases to provide further insights that may go beyond a specific test at a later point in time.

Table 1: Tests required for a containerized CubeSat single unit, including test levels and durations, as derived from reference 3 with Mission maximum predicted environments (MPE) as determined by SpaceX

Test	**Characteristic load**
Shock	3 dB above MPE, 2 times in each of 3 orthogonal axes
Random vibration	MPE spectrum for 1 minute in each of 3 orthogonal axes
Combined thermal vacuum and thermal cycle	± 5 °C beyond acceptance for 20 cycles total
Static load	1.25 times the limit load

In relation to the assembly, verification, and testing (AVT) procedures, the team learned about the importance of proper quality assurance (QA). During the integration of the Bunny payload on D-Orbit's host spacecraft, it became clear that one resistor was missing on the PCB that was assigned to become the flight model. Most of the software development was performed on a replica of this PCB that included the critical resistor; therefore, this issue was not identified during functional testing. Furthermore, the functional testing suite of the flight model was not extensive enough to trigger the issue during software development and functional testing, as the specific failure mode seldom occurred. Nevertheless, this issue could still be fixed during integration at D-Orbit's facilities. The landing pads that were supposed to hold the missing resistor could be electrically shorted, ensuring that the flight model became fully functional.

2.3 Operations

In contrast to other missions, for example, CHESS, EST could not operate the payload on its own for the Bunny mission. In fact, D-Orbit served as a relay for the communication between EST and the satellite. Despite this being an unusual approach for CubeSats, this is a nominal procedure for major space missions, extending the educational

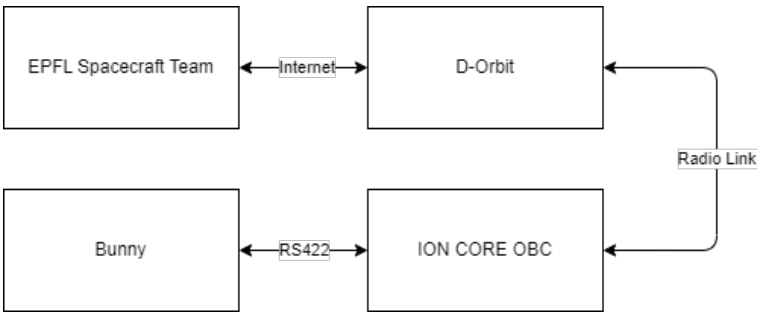

Figure 4: All communication with Bunny had to go through D-Orbit. This resulted in a minimization of command packets sent.

component of this project further. After a command is sent to D-Orbit, it is uplinked to the ION OTV, where the received data is processed by D-Orbit's core OBC. The commands are then forwarded to the Bunny OBC in the next operation window. The data generated by Bunny is sent back through the same path. Figure 4 illustrates this data flow from EPFL to Bunny and back.

The following subsections discuss the tests run on Bunny within the first five months. Two of those tests (Test 1 and 2) were implemented before launch, while the rest was added during flight (Test 3 to 5).

2.4 Pre-Launch Developed Tests

Test 1: I am alive This test is the equivalent of a ping in a network. The test is considered as passed if Bunny replies with a predefined message to the command sent from the core OBC over the RS-422 interface.

Test 2: Bootloader A file was created and deleted. In combination with a reboot, this procedure is sufficient to demonstrate updating capabilities. After this test is passed, the software can be updated with minimal risk.

2.5 Post-Launch Developed Tests

Test 3: Sensor Read The core OBC requests sensor data, which is then collected and returned. The test is passed if the data is received and consistent. There are no sensors right next to Bunny to verify the measurements taken by the sensors. However, D-Orbit provides data from temperature sensors close enough to Bunny to detect major problems in the measurements. This test is considered as passed when the data between the different internal sensors agree with each other and the sensor from D-Orbit.

Test 4: FPGA Reprogramming A new *.dat* file is uploaded to the MCU and then used to reprogram the FPGA. The test is passed if communication with the reprogrammed FPGA is established and the operation of the new design is demonstrated.

Test 5: Microcontroller Reprogramming The updated version of the main flight software is uploaded to the microcontroller and loaded. The test is considered successful if the new software runs on the MCU and the system answers a ping.

2.6 In-Flight Code Development

Even after the integration of Bunny's flight model, software development activities continued to be carried out as the software team kept on refining the software using the engineering model. This presented an important opportunity to iterate and refine the software based on direct in-flight feedback of the tests performed. Unexpected issues and failures led to changing the software verification strategies, in which we run multiple tests, with some expected to succeed and others in which we tried to recreate the failures encountered. All of this led to a more comprehensive understanding of the system's behavior and better identification of potential issues. By performing in-orbit testing, the software development process was able to benefit from testing in the target environment and iterative refinement to ensure the robustness and reliability of the system. One major difference between testing code on the ground and in orbit is the needed effort of up and downlinking the data. This is associated with financial costs. Therefore, multiple tests were

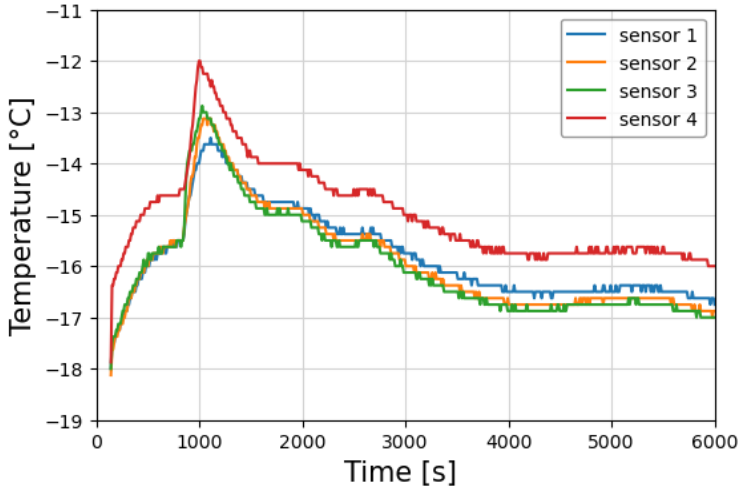

Figure 5: The temperature measured by four temperature sensors placed on the Bunny PCB.

uploaded in a single batch. As an issue with a test has effects on the following tests, the test execution was ordered by our perceived risk of such errors (in ascending order).

3. Results

All tests described in section 2.3 were run within the first five months after launch. The results are shown in table 2. At the time of writing (July 2023), Bunny is still operational and further results are expected.

Table 2: The results of the tests run on the Bunny OBC. We differentiate between *Passed* (success criteria met on the first run), *Finally Passed* (success criteria met after at least one failed run), and *Failed* (the success criteria were not met).

Test	**Result**
Test 1: I am alive	Passed
Test 2: Bootloader	Passed
Test 3: Sensor Read	Finally Passed
Test 4: FPGA Reprogramming	Failed
Test 5: Microcontroller Reprogramming	Passed

3.1 Pre-Launch Developed Tests

Test 1 and Test 2, which were both developed and tested pre-launch, both succeeded in the first run.

Test 1 was run within the first week after launch to verify the payload's most basic functionality. After a waiting period of one month imposed by the host spacecraft, the functionality of the bootloader could be verified. This test was successfully rerun again four months after launch as part of Test 5.

3.2 Post-Launch Developed Tests

Test 3 failed four times before data was successfully returned. The four failed tests were caused by problems with the FPGA programming that was done in the same operation window, as identified later. Figure 5 shows 6 000 seconds of temperature measurements on Bunny after a cold-start. The Pearson correlation between the different time series varies between 0.976 (sensors 1 and 3) and 0.993 (sensors 2 and 4). We, therefore, conclude that the data between the sensors is consistent and that the test is passed. In addition, non-published reference measurements on the host spacecraft agree with the average measured temperature within a margin of ±3 °C.

Test 4 failed five times due to different reasons. The first three failures can be attributed to incorrect verification on the ground. The first test failure was caused by a file naming conflict, while the next two were due to timeouts. The root

cause of the timeouts was a difference in time measurement on the flight and engineering model (minutes and seconds) and too short margins that were allocated for file transfer and write time. As a consequence of the three failing tests, software was now run ten times on the engineering model prior to uplink. The next two attempts of reprogramming the FPGA failed with unknown root causes. The log files show that the fourth run failed because the OBC was not properly responding to a ping, while the fifth run failed while transmitting data from the ION core OBC to Bunny. However, the last test returned sensor data, as mentioned previously.

Test 5 succeeded in the first run. Re-uploading an updated version of the flight software worked multiple times without any issues.

4. Discussion

The Bunny mission enabled the in-orbit demonstration of a student-built onboard computer in low earth orbit, raising its technology readiness level (TRL) to 9. This was accomplished by a team of students with a development time of 4 months as part of a broader initiative to mitigate risks associated with CubeSat missions.

This mission provided a platform for the student team to demonstrate their ability to manage tight payload delivery timelines whilst maintaining the reliability aspects that ensured a functional system capable of transmitting telemetry to the ground. Consequently, the team accumulated expertise related to the requirements of space systems, including material selection, environmental testing procedures, and dynamic design decisions that allowed for the payload to continually evolve post-launch. This acceleration of in-orbit demonstrations proved useful in coping with the mission schedule. However, accelerating the development of the payload presented unexpected challenges, particularly hardware issues during integration and in-orbit software testing failures. These complications can be attributed primarily to the team's novice experience in defining comprehensive testing strategies prior to launch. To mitigate such hardware-related issues in the future, it is advisable to adhere to systematic checklists during assembly, assuming that these issues are not directly linked to design flaws. Regarding software testing, a more comprehensive end-to-end testing approach should be considered, where the entire test process, inclusive of results interpretation and sanity checks on the data generated, is undertaken. In contrast to other longer-lasting projects like CHESS, this mission allowed a single student team to experience the full project life cycle of a space mission without major changes within the team. This led to students directly experiencing the impact of their decision in future mission phases. Furthermore, passing through all mission phases allowed the management team to identify previously unanticipated hurdles for the CHESS mission. As a consequence of the Bunny mission, such hurdles were addressed in the CHESS mission reducing the risk of delays. We highly encourage other student teams to adopt this approach of in-orbit demonstrations in preparation for a full CubeSat mission.

Due to the time constraints of this first mission undertaken by EST, the learning experience was inclined more towards understanding the complexities of launching components into space rather than effectively evaluating the hardware for potential reuse in the CHESS mission. This can be further shown by the fact that the OBC intended to fly the CHESS mission does not reuse the architecture and components of Bunny. Therefore, the technology readyness level (TRL) of the CHESS OBC was not increased.

The experience gained through the Bunny mission serves as a foundation to pursue similar launches aiming at minimizing the risks on the CHESS CubeSat subsystems. The development of a more complex version of the OBC, in addition to an in-house developed X-band transmitter, was initiated at the end of 2022, with the goal of performing in-orbit demonstration/validation mission of these components no earlier than the fourth quarter of 2024. These subsystems are undergoing more thorough testing with a higher number of iterations, given their increased respective complexity, further applying lessons learned through Bunny.

5. Conclusion

The Bunny mission resulted in the successful development, testing, integration and operation of a student-developed onboard computer in orbit. The project provided the team with valuable insights into project management and testing, thus paving the way for the upcoming launches.

This de-risking of core subsystems of the CHESS satellites considerably contributes to the risk mitigation strategy of CHESS. The work presented here represents the first steps towards in-situ monitoring of the chemical composition, the total number density, and the total electron content of Earth's upper atmosphere. These valuable measurements will provide insights into the dynamics and evolution of Earth's atmosphere over timescales ranging from seconds to

the age of the Solar System. The scientific data resulting from this mission will serve as a baseline for comparative planetology to study the different evolution of atmospheres of the three sibling planets, Earth, Venus, and Mars.

Acknowledgments

We would like to thank all the involved institutions and sponsors for their support. We especially thank the EPFL Space Center (Lausanne, Switzerland) and Space Innovation (Lausanne, Switzerland) for their academic support in making this project possible and the University of Bern (Bern, Switzerland) for providing us with testing facilities. We also thank our industrial partners Eurocircuits (Leuven, Belgium), Swissbit (Bronschhofen, Switzerland), and The Countdown Company (Orbe, Switzerland), whose financial support enabled this mission. Finally, we are grateful for our partner D-Orbit (Fino Mornasco, Italy), who enabled us to go to space.

All authors declare that they have no conflicts of interest.

Acronyms

AVT	Assembly. Verification, and Testing
CHESS	Constellation of High Energy Swiss Satellites
EST	EPFL Spacecraft Team
FPGA	Field Programmable Gate Array
MCU	Microcontroller
OBC	On Board Computer
OTA	Over-the-air
OTV	Orbit Transfer Vehicle
PCB	Printed Circuit Board
PDR	Preliminary Design Review
QA	Quality Assurance
TRL	Technology Readiness Level

References

[1] Rico G. Fausch, Gregor Moeller, Markus Rothacher, Nicolas Martinod, Tristan Trébaol, Alfonso Villegas, Jean–Paul Kneib, François Corthay, Marcel Joss, François Tièche, Marek Tulej, and Peter Wurz. CHESS: Measuring the Dynamics of Composition and Density of Earth's Upper Atmosphere with CubeSats. In *2022 IEEE Aerospace Conference (AERO)*, pages 01–13, March 2022. (https://www.doi.org/10.1109/AERO53065.2022.9843791).

[2] Rico G. Fausch, Claudio Zimmermann, Thomas Gerber, Janis Schertenleib, Martina Föhn, Audrey E. Aebi, and Peter Wurz. Monitoring space weather with a sensitive 1 u cubesat mass spectrometer. In *2023 IEEE Aerospace Conference*, pages 1–11, 2023.

[3] Gregor Moeller, Flavio Sonnenberg, Alexander Wolf, and Markus Rothacher. A high-precision commercial off-the-shelf GNSS payload board for nanosatellite orbit determination and timing. In *44th COSPAR Scientific Assembly. Held 16-24 July*, volume 44, page 3389, July 2022. https://www.cospar-assembly.org/user/download.php?id=31195&type=abstract&section=congressbrowser.

[4] S.J. Kapurch. *NASA Systems Engineering Handbook*. DIANE Publishing Company, 2010. (https://books.google.ch/books?id=2CDrawe5AvEC).

[5] Cliff Berg. Spacex's use of agile methods, December 2019. (https://cliffberg.medium.com/spacexs-use-of-agile-methods-c63042178a33).

[6] Elm Chan. How to use mmc/sdc. http://elm-chan.org/docs/mmc/mmc_e.html. Accessed: 2023-07-03.

[7] Space Exploration Technologies Corp. Rideshare Payload User's Guide, March 2022.