Chair of Spacecraft Systems
TUM School of Engineering and Design
Technical University of Munich

Engineering Project / Bachelor Thesis / Semester Thesis

Start date: As soon as possible

Topic: *On-Board Computer for EventSat mission*

We are seeking students to help in the development of the onboard computer for EventSat mission. This is an incredible opportunity to get hands-on programming experience for a real mission going to space.

The EventSat mission is a 6U CubeSat technology demonstration mission sponsored by the Chair of Spacecraft Systems to apply event cameras to detect stars and other space objects. Event cameras are a specialized type of camera that use changes in brightness on a pixel-by-pixel level to detect motion in view. Building upon existing work, which has demonstrated the effectiveness of these cameras on ground in conjunction with telescopes, our mission seeks to demonstrate the effectiveness of this class of camera for space-based observation.

As a university CubeSat project, we recognize the fact that most students have never had the chance to work on flight software, and thus the tasks on this project are intended to be a learning process where students develop the necessary skills alongside the satellite itself. The most important skill we are looking for in students is the passion for the work and the willingness to devote the time to build the necessary programming skills and associated programs.

There are a multitude of opportunities available, whether it is writing embedded software, creating software architecture design documents, or performing unit testing and validation of existing code. We are happy to tailor tasks to your interests, existing strengths, and skills you would like to build professionally long-term.

Tasks

We are in the initial phases of the development of the onboard computer, and thus there are lots of opportunities for work. Below, we highlight several main themes of the work we are pursuing:

Software Architecture Definition

A software architecture document is used to provide a comprehensive architectural overview of the system, using a number of different architectural views to depict different aspects of the system. The architecture definition defines the function of each subsystem, as well as associated coding functions to support this purpose, the inputs to these functions, the outputs, and how they are initiated. Students in this theme will work to define the architecture and then write flight software code to fulfil the systems' functions. Possible tasks include:

- Define functions in each sub-system of the coding architecture (naming convention, inputs, outputs)
- Develop embedded code to implement these functional requirements of different satellite subsystems, using identified functional requirements and supplementary subsystem requirements
- Collaborate with the unit testing team members to ensure verification and validation of written software

The goal of a software architecture document is to provide the reader with a complete review from a software perspective of how the system functions.

Software Interface Control Document

Within the satellite, each subsystem has sensors available that measure the subsystem's health. The onboard computer is responsible for interacting with this subsystem sensing data and operating the spacecraft accordingly. Students working on this theme will work on defining the data format passed from different spacecraft subsystems, functions to interact with this data, outputs, and intersection with spacecraft operational modes. Possible tasks include:

- Define sub-system boundaries, which are used to separate software elements with each system from one another. Sub-system boundaries are used to determine what data will be exchanged, and which subsystem has the associated sensing capability to record the information
- Define commands sent between sub-systems and events that components publish or subscribe to
- Perform a system theoretic process analysis (STPA) to identify failure modes and interaction-related error sources in the current interface/architecture definition, and then produce a systems theoretic accident model and process (STAMP) to resolve identified failure modes

Work from this theme will define the interface control document, which will create a formalized record of interface requirements.

Unit Testing Software Definition

Unit testing allows developers to detect and fix issues early in the development process, improve code quality, and reduce the time and cost of testing later. *Unit testing is one of the single most important tasks a student could work on in flight software.* Students in this grouping will check specific functions within the OBC computing system to verify it is working correctly. Possible tasks include:

- Establish unit testing protocol documentation and best practices so that future coding produced by OBC will fulfil functional unit tests
- Writing unit tests to verify the intended functions of existing functions built into the OBC for spacecraft subsystems

Requirements

The following skills are nice for students to have, but not required, as students can learn on the job with sufficient time investment.

- Basic understanding of CubeSats and their operations in low-Earth orbit
- Proficiency in C/C++ programming, with an emphasis on object-oriented programming
- Note: this project is subject to export control regulations. An NDA must be signed, and some nationalities might not be eligible to work on the OBC.

Expected results:

The ultimate goal of all this work is to produce a fully functional onboard computer capable of executing basic flight modes and ensuring effective communication and control of other subsystems.

- A comprehensive semester thesis in double-column paper format.
- Well-documented code maintained on a GitLab repository.
- A brief presentation outlining the project results.

Depending on the nature of the project, the quality of work produced, and an internal evaluation, outstanding candidates may receive additional funding to present their work at a world-class international conference.

References

[1] L. Schuberth *et al.,* "Leveraging Event-Based Cameras for Enhanced Space Situational Awareness: A Nanosatellite Mission Architecture Study," *75th International Astronautical Congress (IAC)*, 2024. [Online]. Available: https://mediatum.ub.tum.de/node?id=1759594

Supervisor

Clemente J. Juan Oliver, M.Sc.
Phone: +49 89 289 – 55752
E-mail: clemente.juan@tum.de