Onboard Computing Platform for Cube Satellite

by

Joyce Lee

Steven Terryberry

Chris Bechard

(Advisor: Dr. Jayshri Sabarinathan)

ECE 4415 Computer Engineering Project

Technical Proposal

Department of Electrical and Computer Engineering

Western University

London, Ontario, Canada

October 26, 2018

Proposal Summary

Western University and Nunavut Arctic College, partnered with the Canadian Space Agency, Canadian Cubesat Program, and Canandesys aims to launch a 2U VR imaging cube satellite within at most 4 years of May 2018. The finished product must remain functional and performant for the entire operational lifetime of 6-12 months in low earth orbit.

The aim of this capstone design project is to design an onboard computer for the satellite such that the final products meet the standards for "Phase B: Preliminary Design and Technology Completion." This includes preliminary designs that would meet primary mission requirements for communications, data handling, attitude control and determination, and electrical power subsystems.

For hardware, the proposed design method is to identify a platform of off-the-shelf components to meet the requirements of both an S-Band and UHF design, and to demonstrate key aspects of the system on prototype hardware. For software, the design process is to evaluate and integrate an operating system, communication user interface, and essential operational functionality from each subsystem.

Problem Statement

Commercial computer systems are not completely suitable for space exploration because the conditions and requirements they were developed for on earth are far different from the that of outer space.

Background Information

CubeSats enable independent researchers to conduct small-scale launches allowing for more frequent, creative, and low-risk space exploration projects. Environmental factors and limited scale, require many considerations to be made for the on-board computer subsystem to prepare it for use in orbit.

Due to the ionizing radiation satellites are subjected to, the OBC hardware needs to be protected against degradation through use of radiation hardening. However, the short timeframe of operation and larger budget impact renders complete radiation hardening for a 2U CubeSat infeasible. Instead, protection can be provided through error-checking or shielding [1]. A CubeSat must be self-sustaining, which can be achieve by using solar power. Due to the minimal power generated compared to other methods, the OBC must maintain a low power requirement. Temperatures while in orbit can change drastically between hot and cold extremes, thus the hardware must be able to withstand such temperature shifts. To fit the frame of a 2U, the PCB must have an area no larger than 10 x 10 cm. This has been achieved in the past by using stackable PC/104 modules [2].

The logistics of operating software in space must also be considered. CubeSat projects in the past have used a variety of OS such as freeRTOS, Linux, and Windows, as all are viable choices. A hard RTOS is not necessary since there is no risk of harm, but certain aspects will have soft real-time needs. The OBC will need to manage the scheduling of tasks from the other subsystems that require sending and receiving data within the CubeSat. Multimedia being stored and transmitted, in contrast do not have real-time needs as they are only sent when the satellite is within the transmission range of a ground station which will be for very limited durations. Since the primary goal of the OBC is manipulation of data, it is essential to understand the rate at which data can be transferred. UHF and S bands have maximum data rates of 9600 bps and 256 kbps respectively [3]. This low bit rate can cause the memory to fill up if more data is being stored than can be transmitted. Thus, data usage must be budgeted into different modes of operation such as imaging mode, power saver mode, and communications mode [1] to ensure functionality.

Over the past couple decades, there have been many CubeSat launches that have utilized a variety of different specifications and operating systems. These launches have experienced mixed results with the successful completion of mission tasks. The most common cause of failure was a lack of adequate power [1].

Project Objectives

The objective of this project is to design a computer system which will be responsible for processing image data captured by the CubeSat camera, communicate with the ground station, and interface with the components designed by the electrical power, communications and structural teams. Two methods of communication are currently being investigated by the other multidisciplinary teams: S-Band (spirit) and UHF (opportunity). The onboard computing system will need to be compatible with both systems. This will require selecting a piece of hardware and a compatible operating system that can withstand space conditions and have the computing power necessary to buffer and transmit compressed images.

This project is the first of three computer capstone projects and it will define the preliminary design of the computer system onboard the CubeSat. The goal for this phase of the computer system CubeSat project is to produce two deliverables: functional design documents that will be used for future projects and a working prototype which will demonstrate the computer system's essential functionality.

Methodology

Thermal, power, mechanical, and communication channel design methodologies are essential to the overall design. However, in the interest of narrowing scope, this section explores methodologies only as applied to the hardware and software interfaces of the COMMS, C&DH, ADCS, and EPS. ADCS subsystems. The structure will be similar to a waterfall or V-model development life cycle [5], with an emphasis on top down requirements, system specification, development, and subsystem testing with a plan for integration and deployment. This is where the objectives of a “Phase B: Preliminary Design and Technology Evaluation” [6] conclude.

First, the communications system will include both an S-band and UHF compatible platform. The system modelling activities will include: one, UML use case and activity diagrams for the telecommand interface, two, the frequency and volume of housekeeping and science data, three, network layers above the physical layer, four, in collaboration with the communications teams, the physical network layer, and five, the input/output hardware interfaces. The hardware platform primarily has transceiver trade-offs to consider with options for an ASIC vs FPGA, encoding schemes, and modulation schemes. By writing portable software and keeping as much hardware constant as possible, effort for the S-band and UHF variants is minimized. The hardware platform would include but is not limited to: data encapsulation and networking, encoding and modulation configuration, the operating system and middleware platform, and a reliable user authenticated telecommand and telemetry service. An integrated subsystem should include secure bootloaders and drivers for the platform. A software verification plan might involve unit tests for telecommand and telemetry operations involving only the communications subsystem, construction of basic breadboard prototype, and/or integration testing with other available prototypes or mocked interfaces, with the aim of testing operation under all proposed system modes. A prototype does not necessarily have to include every hardware component, simulations may be conducted with the chosen microcontroller in-loop.

Second, the command and data handling subsystem had the responsibility of managing system state, logging housekeeping data, and transmitting data to or receiving data from the ground station. The design activities should include modelling: one, the support for the payload camera interfaces, two, UML messaging and sequence diagrams for each subsystem operation, three, a UML state machine model for different operational modes, and four, the required memory, data interfaces, and processor capabilities. For hardware development, this subsystem has little to no unique peripheral requirements, and should have the widest availability of low cost hardware options. Additionally, the payload cameras will each have their own onboard memory, reducing the requirements of the C&DH hardware. For software development, the C&DH system will contain the bulk of the earth observation application development. The test strategies should verify, through similar methods as the communications subsystem, that the application uses operational information from other subsystems to correctly to avoid and recover from system failure states.

Third, the attitude control and determination subsystem with corresponding attitude management application logic will be modelled by the ADCS team. The hardware platform for this subsystem must include the desired peripherals for their model. Aside from this, the attitude state will have implications for the behaviour of the rest of the system, necessitating high availability and reliability testing for the C&DH and COMMS subsystems.

Last, the electrical power subsystem with corresponding power management profile will be modelled by the EPS team. The design stage methods for this subsystem in the context of the OBC are to model: one, the response of the EPS to hardware failure events such as, and two, model the impact of different operational modes on power consumption. The hardware and software challenges of this system are outside of the domain knowledge of this team and available resources, therefore the primary design goal should be to interact reliably with a third party system.

Overall, each system will have strategies at each stage if the SDLC to identify successful models, development strategies, testing strategies, prototypes, and future recommendations.

Project Tasks and Responsibilities of the Team Members

Each team member is responsible for attending weekly meetings with the project manager and other multidisciplinary teams, providing updates, collaborating and communicating with other subsystem teams to ensure compatibility, and working on assigned individual tasks that are decided on by the team.

The following is a list of anticipated tasks required to complete the project with approximate time estimates. This timeline was created with an approximate 30 week (~8 month) timeline in mind. The tasks specified within each phase will be planned in sprints.

Task	**Duration (Weeks)**
Research Phase - Acquire data from previous research teams (transition) - Research strengths and weaknesses of existing CubeSat computer systems - Understand challenges faced by previous research teams - Define the scope of project	3

Exploration and Design Phase
- Define objectives and constraints
- Identify margins for power budgeting
- Identify considerations/connections required to interface with other subsystems (mounting, power, mass etc)
    - Specify a communication interface for transmitting digital application data from satellite to ground station
    - Specify application data interfaces for each subsystem
    - Specify power interfaces for each subsystem
    - Specify thermal interfaces for each subsystem
	6

- Telecommand interface development - Data budgeting - Mid-term progress report	
Build Phase - Identify criteria for success and create tests for each subsystem component which will be used during the testing phase - Identify methods to demonstrate the system (prototype hardware, testing simulations) - Prioritize subsystem tasks that rank critical functions of the computer in relation to the overall mission of the CubeSat (ie. memory preservation, transmitting real-time data, attitude determination and control) - Request and purchase parts (after Project Manager Matt Cross submits an application for satellite license in January) - Application modelling for essential functionality (some components might be “black boxes” such as power management and the actual ADCS functionality) - Confirm mass budget for structural team (with basic mounting details) - Thermal budgeting	10
Testing and Iteration Phase - Test each individual subsystem component and identify successes and failures - Make adjustments to fix failures - Simulate whole mission (day in the life testing) [4] - Demonstrate essential functionality with a prototype	8
Project Deliverables and Presentations - Design demonstration - Design presentation	3

- Western CPSX Space Day presentation - Final report and reflection	

Preliminary Budget and Parts List/Software Tools

The grant acquired for the project according to the proposal is $250,000 with $50,000 going to educational outreach activities, leaving $200,000 for launch and some unknown amount for CubeSat development.

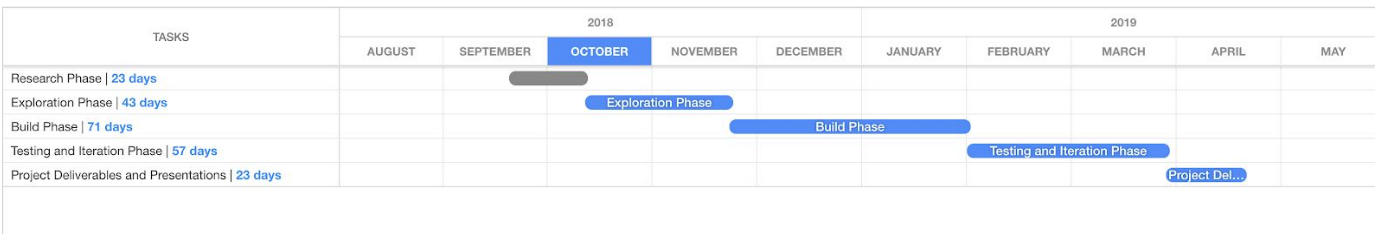
The project will prioritize using proven open source or off-the-shelf software/hardware components. As a general estimate, the primary cost for this project will be integrated subsystems. Even if they are not purchased in this design cycle, the recommended costs should be accounted for. More detailed part selection will be evaluated in the design phase of the project. The following is a preliminary list of parts and software tools which give a rough estimate for the cost of the project:

Item	**Approximate Cost**
Integrated EPS	~ $6700 [7]
S-band integrated subsystem	~ $6700 [7]
UHF integrated subsystem	~ $6700 [7]
ADCS	~ $6700 [7]
C&DH	< $300 (without radiation hardening) [8]

Software development tools	**Cost**	**License**
Git with Github	$0	Git uses GNU GPL v2 and

		GNU LGPL v2.1 Github is a proprietary service
UML with Microsoft Visio	$0 (Western license)	Proprietary

Gantt Chart



References

[1] Kulendran, S. (2018). *OBC Brief*. [Online] Slack-files. Available: https://slack-files.com/files-pri-safe/TAPF7KG3D-FDGU1LC5C/obcbrief.pdf?c=1540509948-c67eb7ed93c8e20f89d9eac41e00d04b413ba873 [Accessed 26 Oct. 2018].

[2] Bannatyne, R. (2018). *Designing a rad-hard CubeSat onboard computer*. [online] PC/104 and Small Form Factors. Available: http://smallformfactors.mil-embedded.com/articles/designing-rad-hard-cubesat-onboard-computer/ [Accessed 26 Oct. 2018].

[3] D. Selva and D. Krejci, *A survey and assessment of the capabilities of Cubesats for Earth observation*. Cambridge, USA: Acta Astronautica, 2018, p. 5 [Online]. Available: http://systemarchitect.mit.edu/docs/selva12b.pdf [Accessed: 26 Oct. 2018]

[4] C. Jamie et al, *CubeSat 101: Basic Concepts and Processes for First-Time CubeSat Developers.* California Polytechnic State University, San Luis Obispo, USA. 2017. [Online]. Available: https://www.nasa.gov/sites/default/files/atoms/files/nasa_csli_cubesat_101_508.pdf [Accessed: 26 Oct. 2018]

[5] N. B. Ruparelia, "Software development lifecycle models," *ACM SIGSOFT Software Engineering Notes*, vol. 35, no. 3, p. 8, May 2010. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1764810.1764814 [Accessed: 26 Oct. 2018]

[6] S. R. V. Hirshorn, "NASA Systems Engineering Handbook," Feb. 2017.[Online]. Available: https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20170001761.pdf [Accessed: 26 Oct. 2018]

[7] Cube Computer, CubeSatShop., Delft, Netherlands. Available: https://www.cubesatshop.com/product/cube-computer/ [Accessed: 26 Oct. 2018]

[8] Y. Kovo, "State of the Art Report," *NASA*, 20-Feb-2015. [Online]. Available: https://sst-soa.arc.nasa.gov/08-command-and-data-handling. [Accessed: 26-Oct-2018].