ONBOARD DATA HANDLING FOR CUBE SATELLITE

by

Joyce Lee

Steven Terryberry

Christopher Bechard

(advisor: Dr. Jayshri Sabarinathan)

ECE 4415 Computer Engineering Design Project

Analysis and Test Report

Department of Electrical and Computer Engineering

Western University

London, Ontario, Canada

March 12, 2019

Glossary

ADCS	Attitude Determination and Control System
C&DH	Command and Data Handling
IDE	Integrated Development Environment
OBC	Onboard Computer

1. Final Analysis of Design Concept

There are two major deliverables for this capstone project: research and initial groundwork which will be used in the final cube satellite project, and a prototype which will demonstrate the functionality of the OBC and how it will interact with other subsystems.

The major requirement for the first deliverable is to select an onboard computer which has the necessary data interfaces and power requirements to be compatible with the communications, electric power, and attitude determination and control subsystems and the Canadensys VR camera. The major factors affecting this decision were addressed in the Progress Report. The team is confident that the Endurosat CD&H board is the best possible option available for the long term success of the cube satellite project, given factors such as cost, compatibility with other subsystems, and customer support. A decision will be made by the Project Manager for which communication method will be used in the final design; the board we have chosen accommodates both those options.

The goal of the prototype is to demonstrate how signals from different subsystems can interface with each other through the onboard computer. In the progress report, the team noted that one of the major difficulties they were facing was the lack of mock data to simulate interactions between subsystems. In order to circumvent this, the team chose to purchase three development boards; one would serve as the main onboard computer and two would serve as external subsystems (camera and communications) which would send simple, hard-coded signals over the serial interfaces. This would also allow each team member to learn to program on the same development board. An overview of the prototype design concept is shown in Figure 1.

2. Prototype Fabrication

The engineering methodology we employed for this project was a waterfall or V-model development life cycle, with emphasis on top down requirements, system specification, development and subsystem testing with a plan for integration and deployment. The V-model has four major phases: requirement analysis, system design, architecture design and module design, followed by the coding phase. Requirement analysis, system design, and architecture design for the overall project were completed in the Progress Report. This report will focus on module design, which is implemented in Embedded C code on the STM32 Nucleo-144 development board (Nucleo-F429ZI).

As discussed in the progress report, three possible boards were considered for the prototype, as shown in Table 1. The Nucleo-F429ZI was chosen for the prototype hardware because it was a cost-effective option with a similar processor, data interfaces, and operating system as the Endurosat OBC. The board itself however, was not an appropriate recommendation for the final design because its components would not survive the harsh space environment. The cost of the prototype and the software tools used is shown in Table 2 and Table 3.

Table 1. *Hardware prototype options*

Option	Item	Cost
1	Endurosat OBC with pre-installed FreeRTOS	$3625 USD
2	180 MHz Cortex M4	Low: $30: Nucleo-F429ZI, High: $80: EA-QSB-016 (incorrect clock rate, more peripherals)
3	216 MHz Cortex M7	Low: $20: NUCLEO-F767Z, High: $70: MIMXRT1020-EVK (incorrect clock rate, more peripherals)

Table 2. *Prototype hardware cost breakdown*

Part	Unit Cost	Quantity	Total Cost
Nucleo-F429ZI	~$30	3	$90
Miscellaneous cables	~$3	3	$9

Table 3. *Software tools cost breakdown*

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
Eclipse IDE	$0	Open Source
GNU ARM Embedded Toolchain	$0	Open Source
GNU MCU Eclipse OpenOCD	$0	Open Source

Module Design Overview

In order to demonstrate how the onboard computer will interact with other subsystems, two Nucleo-F429ZI boards will be used to send and receive mock data to and from the onboard computer. The prototype will consist of three modules: the camera, onboard computer, and communications, as shown in Figure 1. The data interface for the Canadensys camera is RS-422 over UART and the data interface for the communications subsystem is the CAN buses.

The other subsystems have been omitted from the prototype for simplicity. The prototype will demonstrate the board's ability to receive serial data from the camera and reformat it into packets for the transceiver. Other tasks will be structured in a similar way, where information from one subsystem is parsed from a serial input, then reformatted and output to another subsystem.

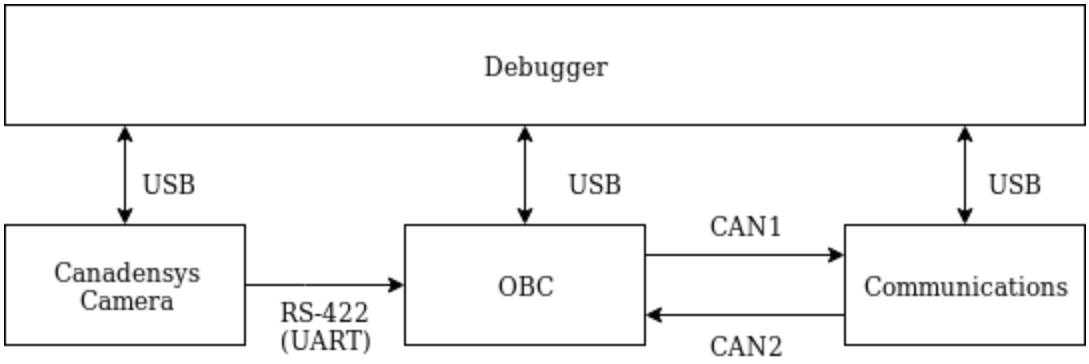

Figure 1. *Module Design Overview Diagram*

Development Environment Setup

The team chose to use the open source Eclipse IDE with the GNU ARM embedded toolchain plugin and OpenOCD for the development of the prototype. This development environment is the same across MacOS and Windows which would allow us to collaborate seamlessly when developing on the Nucleo-F429ZI. The GNU ARM embedded toolchain is an Eclipse plugin containing a set of open source tools that make it easy to program microcontrollers with the Arm Cortex-M and Cortex-R processors. OpenOCD is a debug environment that works on both MacOS and Windows.

3. Validation/Testing Strategy or Protocols

The goal of the prototype is to validate the choice of processor which is used in the Endurosat OBC and Nucleo-F429ZI. In order to validate this, the cube satellite's main research function, which is to send image data to the ground station, will be implemented. The successful implementation of this functionality will show that the cube satellite's main research function can be fulfilled using the chosen hardware with its available power, memory, and processing power. This validation phase is known as Integration Testing in the V-model development life cycle. The following testing strategy will be used:

- Test 1: 13.5MB image can be loaded (in one stream) to the CD&H board. If possible, we will obtain a VR image from Canadensys. Otherwise, any 13.5MB file should suffice. This test will validate the board's compatibility with the Canadensys camera subsystem.
- Test 2: The image is framed as CSP or IP to be compatible with both Endurosat and Satlab packet assemblers. This test will validate that the supporting libraries for this processor can adequately support the data reformatting needs of the project.
- Test 3: Packets can be loaded to the "transceiver" module and the full image can be reconstructed. This test will validate the board's compatibility with the communications subsystem.
- Test 4: The OBC can resend packets if a packet is failed, and the image can still be reconstructed. This test will validate the OBC subsystem's ability to verify the successful transmission of a packet, and take action if necessary.
- Test 5 (time permitting): A partial image can be reconstructed given some failed packets. This test validates that the information sent by the OBC subsystem contains enough contextual information for useful data to be extracted from the cube satellite in case of multiple failed and non-recoverable transmissions.

4. Preliminary Validation and Results Analysis

At this stage of the project, Test 1 was successfully completed and the packet structure for Test 2 was determined. The remaining tests will be implemented in the final prototype.

Test 1: 13.5MB image can be loaded (in one stream) to the CD&H board.
The team was able to successfully transfer a stream of data from one board to another over UART, which validates the processor's compatibility with the Canadensys VR Camera. The data which was transmitted was a hardcoded buffer string because the only information available about the data coming from the Canadensys VR camera is that the images will be processed, compressed, and output over the RS-422 data connection. It is unclear what the exact data format from the Canadensys camera will be, so for this test the team assumed that the output serial data would be a buffer string. In terms of integration testing, we can assume that this successful transfer of data validates the board's compatibility with the Canadensys VR Camera.

Test 2: The board can parse the image and repackage it into packets for the transceiver.
The structure of the packet has been determined, but not yet implemented. The maximum packet size is 100 bytes, according to the communications team. The packet overhead might, at minimum, be: 32 bits for CSP (less overhead than lwIP), 16 bits for CRC-16, 2x8 bit flags + 8 bits control + 8 bits protocol id + 128 bit address for AX.25, for a total of 26/100 bytes. A 13.5 MB image with a compression ratio of 15x is 0.9MB, would be transmitted over 12163 packets, assuming no retransmissions. This is a fairly reasonable number of packets given the contents of the image. Upon researching the available libraries that would assist in the construction of these packets, the team has found that the MBED library which comes with the GNU ARM tools will be able to adequately support this function. Although this test is yet to be completed, the team is confident that it will be possible to implement and that this will validate the data parsing needs of the project.

5. Conclusion

The team plans to present its first version of the prototype to the Project Manager and the other subsystem teams at the weekly meeting on Thursday, March 14. In following with our chosen waterfall/V-model methodology, the feedback from this meeting will be used to implement the next iteration which will be shown at the capstone demonstration presentation. The research conducted by the group will be summarized and presented on a poster for Space Day on April 12.