CougSat Simulation Software

Solution Approach

Mentor

Aaron Crandall

Team Viking

Courtney Snyder
Angie Park
Solomon Egwuonwu

Course: CptS 421 Software Design Project I
Instructor: Aaron Crandall

TABLE OF CONTENTS

I. Introduction

The purpose of this document is to outline the work we have done this semester and the work to be done to complete our alpha prototype by December.

Our client, the club Cougs in Space of Washington State University in Pullman, wants to know that their cubesat, CougSat I, will successfully launch and orbit. To help them feel more confident and help them further develop their design, we will build a testing suite for them. This testing suite will go through all stages of both the launch and orbit to determine if anything could fail and identify the cause of failure. This will enable the club to create a more robust system and more likely make for a successful mission.

Development of the testing suite will be broken up into two phases, the alpha prototype and the beta prototype. This semester, we have been working on the alpha prototype, which is a state machine that looks at all of the top-level behaviors that CougSat I will have, most of which will be represented by various power modes.

The rest of the document contains sections for system overview, architecture design, user interface design, the state of the project, and future work for the project this semester.

II. System Overview

We were instructed by our mentor to construct a state machine of CougSat I since we are simulating its behavior from launch to orbit. To understand how the cubesat itself works, we designed a general state machine, which is a high level view of how the satellite behaves as a whole. Then, we broke up the satellite into the individual circuit boards: Attitude, Communications, Data Collection/Payload, and Power. Conveniently, Cougs in Space has teams for each of these boards and we were able to communicate with the team leads to get better ideas of what each of these boards does. We were then able to integrate the individual circuit boards into the general state machine design.

III. Architecture Design

III.1. Overview

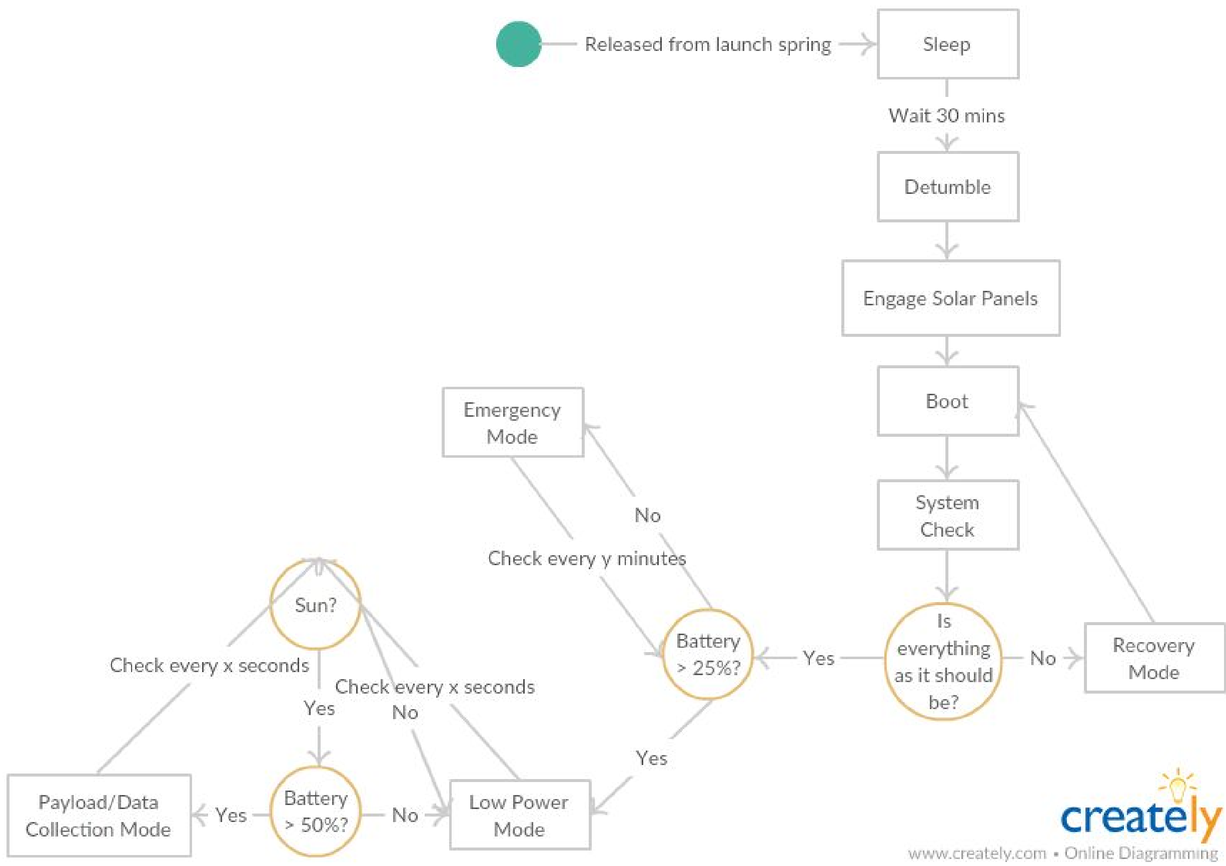


Figure 1. General state machine for CougSat I

For our design, we implemented a state pattern. Since we are more concerned with CougSat I's behavior and our alpha prototype is a state machine, this pattern made the most sense.

As seen in Figure 1, we have determined three power modes: Emergency, Low Power, and Payload/Data Collection. Please note that Recovery mode is not a power mode because it is not dependent on how much battery power is remaining; rather it is dependent on whether or not the satellite successfully passes the system check.

Each power mode is determined by how much battery power is remaining; Emergency is for less than 20% battery, Low Power is for between 50% and 20% battery, and Payload/Data Collection mode is for greater than 50% battery. Within each of those power modes, different subsystems will be turned on, and will exhibit different behaviors.

When CougSat I is in Low Power mode, the Communications, Power, and Attitude subsystems will be on. This is the normal operating behavior of the satellite, as it allows for communication with the ground station, on-board power control, and satellite stabilization. However, if the battery level is too low, having all of these subsystems on will quickly drain the battery.

Thus, when CougSat I is in Emergency mode, only the Power subsystem will be on. None of the other subsystems will be able to use valuable power at this point because our client would rather have the satellite efficiently budget and collect power than die.

Tentatively, when CougSat I is in Payload/Data mode, the Communications, Data Collection/Payload, Power, and Attitude subsystems subsystems will be on. When the satellite is in Payload/Data mode, it will behave in the same way as it does in Low Power mode, but the payload subsystem will also be on and collecting data to send to the ground station in some capacity.

III.2. Subsystem Decomposition

We broke our project into three main subsystems because there are currently four circuit boards on CougSat I and we wanted to illustrate both how they work individually and how they work together. The general state machine subsystem is the cubesat's behavior as a whole and includes all of the circuit boards. The Attitude, Communication, and Power subsystems illustrate individual circuit board behaviors. Note that the Communications subsystem contains both inner-satellite communications and ground communications. The inner satellite communications are all through the main circuit board; that is, the main board is connected to each of the other circuit boards and the other boards are not connected to each other.
Using this design, we achieved low coupling and high cohesion since the state machines are not dependent on each other until we put them all together in the general state machine.

III.2.1 Attitude

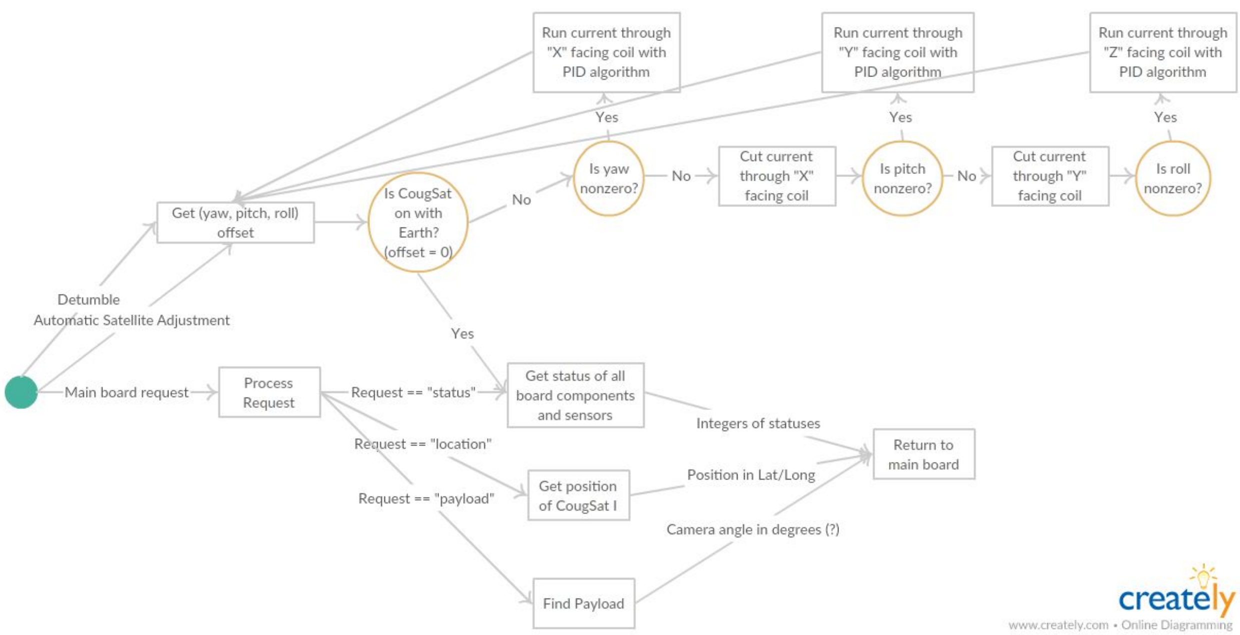


Figure 2. State machine for the Attitude subsystem

a) Description

The attitude state machine represents the attitude circuit board on CougSat I. This subsystem is meant to keep the satellite aligned in the appropriate coordinate system. As of right now, there is one coordinate system such that the face with the antenna is pointing at Earth. Once the payload board is integrated, there may be a second coordinate system such that the camera is pointing at the payload.

b) Concepts and Algorithms Generated

CougSat I will be detumbled using a proportional-integral-derivative (PID) algorithm; this will efficiently slow the angular velocity of the satellite and use minimal power to do so. Once the angular velocity is zero, the attitude board can turn on the coils in the x, y, and z directions to adjust the yaw, pitch, and roll of the satellite. As the satellite orbits the Earth, small adjustments will be needed, and the attitude board will make these adjustments using the same algorithm; correct the pitch offset, then the yaw offset, then the roll offset. The ground station can request the location of the satellite, which the attitude board will provide in latitude and longitude. The ground station can also request the status of the satellite as a whole, which the attitude board will contribute to.

c) Interface Description

The attitude subsystem will only be used if it has been 30 minutes since CougSat I has been released from the spring launcher (Detumbling), if the board reads an offset (Automatic Satellite Adjustment), or if the main board receives a command for information from the attitude board (Main Board Request).

<u>Services Provided:</u>

1. Detumbling

Service provided to: Communications, Power
Description: The attitude board will be connected to an isolated timer that begins counting as soon as the satellite is launched; once the timer reaches 30 minutes, the attitude board will turn on and begin to detumble the satellite. Typically, the attitude board will measure yaw, pitch, and roll offsets using the gyro; right after the launch, the cubesat will be moving so fast that the offsets will all be zero. Instead, we will measure the angular velocity in each direction. The attitude subsystem will use a proportional-integral-derivative (PID) algorithm to efficiently bring the angular velocities to zero, then for each direction we will turn on the coil in the perpendicular face to align it to the initial coordinate system that points at Earth. Once the satellite is no longer tumbling and is pointing at Earth appropriately, it is considered detumbled. Detumbling helps Communications by pointing the antenna at the Earth and helps Power by getting maximal sunlight to the solar panels.

2. Automatic Satellite Adjustment

Service provided to: Communications, Power
Description: After the satellite is detumbled, the attitude board will use the gyro to get the yaw, pitch, and roll offsets, and will turn on the coil in the perpendicular face of the satellite to correct the offset as needed. Automatic Satellite Adjustment helps Communications by pointing the antenna at the Earth and helps Power by getting maximal sunlight to the solar panels.

3. Main Board Request

Service provided to: Communications

Description: The ground station can ask the satellite different things about itself using a set of commands. The main board will receive the commands, decrypt them, and request information from the satellite. The main board can get status and location from the attitude board. If the command is status, the attitude board will get the status of all of its components and sensors and tell the main board if everything is working as it should be. If the command is location, the attitude board will tell the main board its location in latitude and longitude.

Services Required:
Battery level and power mode are required from the Power subsystem. Communication-level commands and inner transmission are required from the Communication subsystem.

III.2.2 Communication

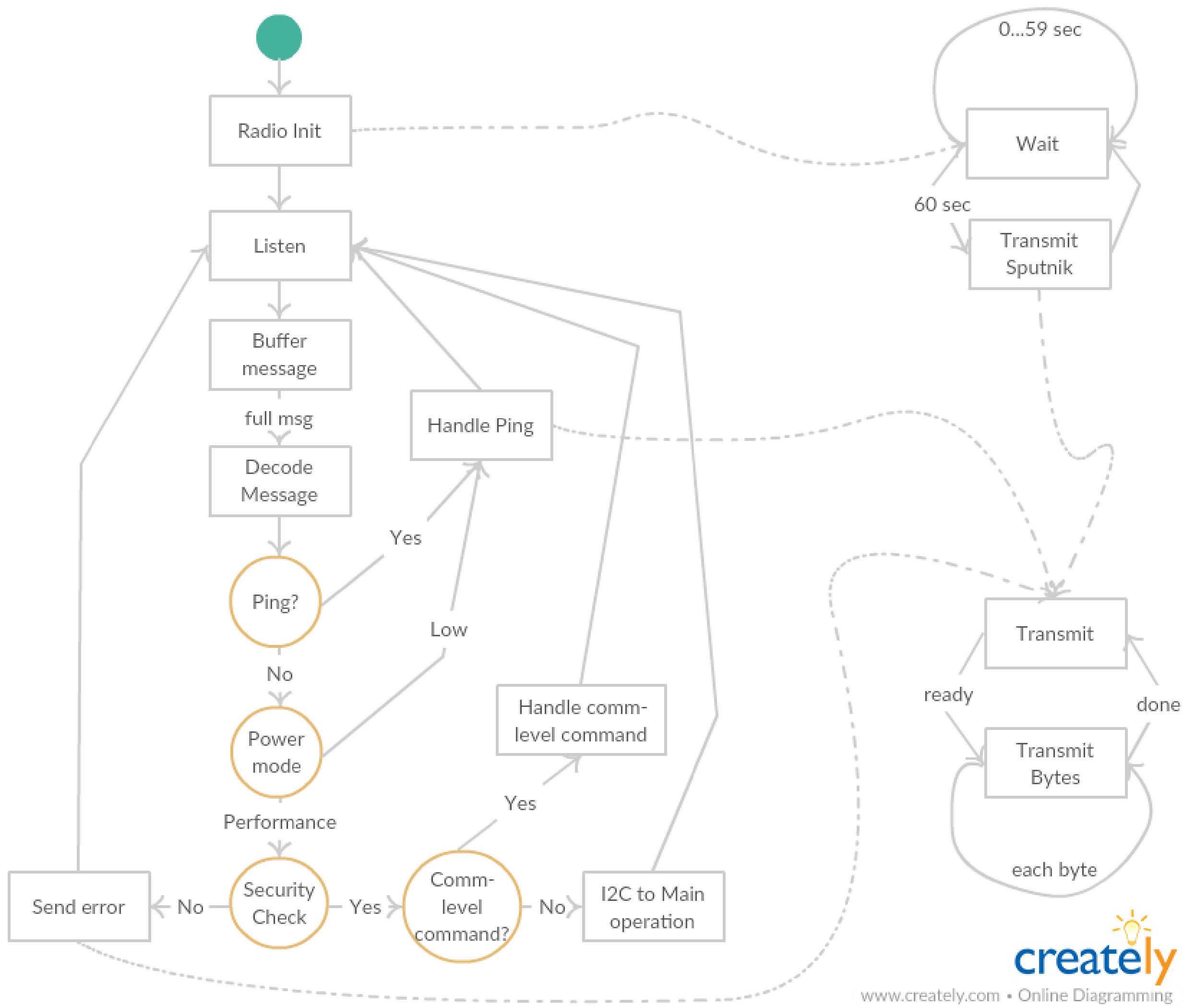


Figure 3. State machine for the Communication subsystem

a) Description

The communication state machine represents the communication circuit board on CougSat I. The board is not available when the satellite is in Emergency mode. It is in charge of both outer and inner communication of CougSat I. The outer communication is mostly about transmissions of commands from the ground control. The inner communication indicates transmissions of information between a couple of boards inside of the satellite in order to perform the requested commands.

b) Concepts and Algorithms Generated

The communication board is not available on the critical battery mode (emergency mode). Therefore, there will be multiple booting ups upon the battery level. Once the board is turned on, it will consistently ping to the ground control periodically as a concept of the first satellite, Sputnik I. It will be on listening mode all the time to get the messages from the ground control. For this process, CougSat I will carry AX.25 protocol which is suitable communication protocol for amateur radio operation. Then, it will decode the entire message to go on the next step. If the message context is about to ping the status, it will handle the command. Otherwise, the communication board needs to check the power level to make sure weather it has enough battery to perform more complex commands. The low power mode will not be able to do it, so it will go back to the listening mode. As long as it is confirmed that there is enough battery, the board will have security verification on the message it receives. It would not handle the command from unverified source and send a message about the event to the ground control. If the sender is authorized, the board will separate the work as the high-level command and the low-level one, which can be handled within the communication board, and send the high-level commands to the main board to perform what they ask.

c) Interface Description

Services Provided:

1. Ping test

Service provided to: Ground control

Description: This is the most basic interface of the communication system. Ping test service will be running all the time when the board is on. It is a diagnostic checking if the satellite is connected to the ground control and their hosts are reachable to one another. It assures of the connection if it successfully receives a data packet back from the ground server when it sends one to there.

2. Inner transmission

Service provided to: Main board

Description: The communication board handles the inner transmission under specific conditions. After getting the commands that requires internal performance, CougSat I should check if the commands is not ping and the power mode is not Low mode. Then, it should check for the security to make sure the commands are from the ground control or other reliable server. The inner transmission will be performed. All commands will go to the main board to be distributed to the proper sub-board for requested works.

3. Outer transmission

Service provided to: Ground control

Description: The outer transmission is the transmission of buffer from the satellite to the ground control. It works as transmitting each bytes of the buffer per one time. This service is provided when it handles ping and it sends error message about the command from unreliable sources.

4. Communication-level commands

Service provided to: Ground control, Attitude

Description: The board has its own system to digest low-level commands by itself. This can simply deal with the commands asking the current status and location of CougSat I. This service is correlated to the main board request of Attitude board.

<u>Services Required:</u>
Knowing the power level is required to figure out if the high-level commands can be doable. The subsystem that provides it is the power system.

III.2.3 Power

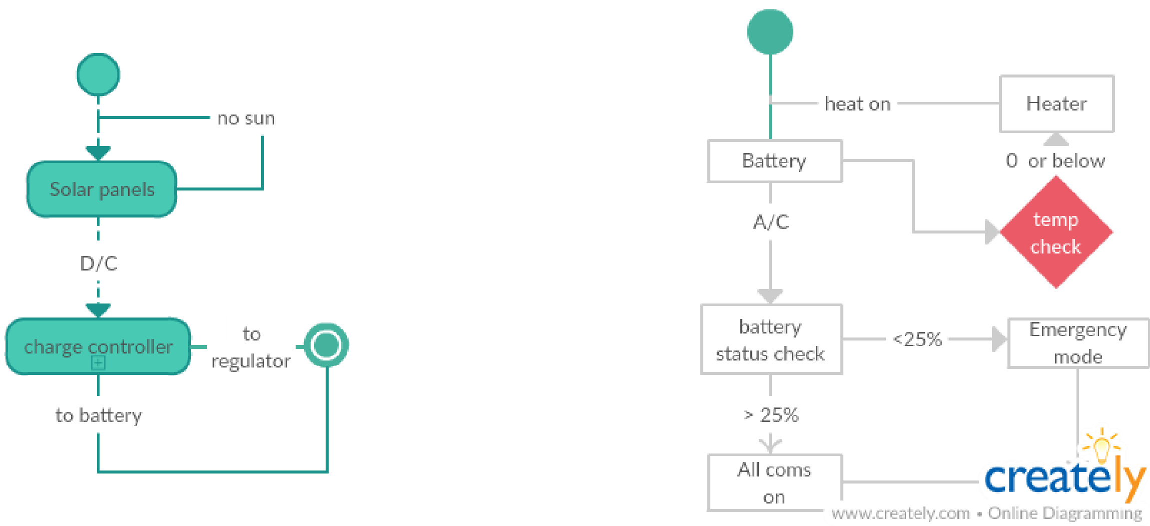


Figure 4. State machine for the Power subsystem

a) Description
The power state machine represents the power circuit board of the CougSat I. This subsystem supplies power to the electronic components on board the CougSat I. The power subsystem includes solar cells, charge controller, batteries, battery charger as well as components required for battery telemetry.

b) Concepts and Algorithms Generated
The power subsystem was based on several cubesat projects that have successfully utilized these components, mitigating project risks. The subsystem has a battery charge regulator that is able to moderate parallel inputs from the solar panels. Once power has been supplied to the power circuit board, it will be converted to the correct voltage and stored the batteries. It also has regulated output buses at 3.3V and 5V.

c) Interface Description
The solar panel collects energy can turns it into power that can either be used in that moment or can be stored to be used later. The battery

<u>Services Provided:</u>
1. Solar panel
Service provided to: Attitude, Communication

Description: The solar panel is a vital importance in the design of the CougSat I as they are the sole source of electrical power to the power subsystem. Without sufficient power supply to the on board power system, power would not be distributed to the remaining subsystems onboard the CougSat I.

2. Battery

Service provided to: Attitude, Communication

Description: The battery is the auxiliary power supply. It provides power to the other subsystems in the CougSat I when the satellite is in eclipse, as the lack of sunlight makes it impossible for the solar panel to provide power.

Services Required:

The solar panel requires power from the sun to be functional as to provide power for recharging the batteries and provide power to other subsystems.

IV. Data design

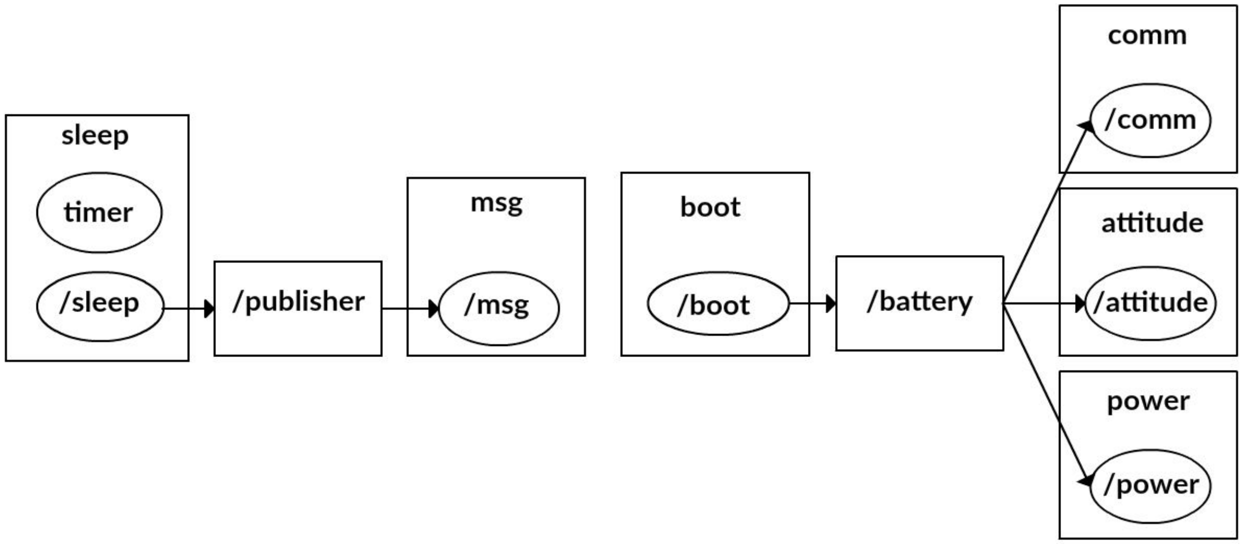
As of right now, we do not see a need for a database to store information sent to and received from CougSat I. This is because the only confirmed command for the satellite to receive and respond to is "ping", and the satellite will just respond with a message confirming that it has not died yet. When the Payload board is incorporated, a database for pictures may be something to consider.

V. User Interface Design

The most challenging part of this project has been setting up the ROS Kinetic filesystem with catkin and navigating the user interface, so we will have documentation on how to set up the testing suite on Linux Mint 18 and Ubuntu Xenial with ROS Kinetic.

Our use cases are successful launch use case, report status use case, and reboot use case. All of these can be examined using an ROS RQT graph because the rqt_graph command creates a dynamic graph of the nodes, functions within nodes, and topics being published in the system of nodes. This will confirm that data is being read and written appropriately.

Using the sleep and boot nodes implemented ROS Kinetic, this is what the RQT graph should display:



For the successful launch use case, the RQT graph will theoretically run until the user exits out of it. While the testing suite is running, the user can get the status of CougSat I and reboot it if necessary. For the report status use case, the user can both see what messages are being published and the satellite will also return some kind of file or data structure with the statuses of each board; that is, if each component is operating correctly. For the reboot use case, the user can see the satellite turn everything off and go through the boot stage again while in orbit; that is, the testing suite itself will not restart.

We hope to launch the simulator using a launch file in ROS; this will allow us to easily run multiple nodes at the same time with collected data to emulate how the boards will behave with these inputs. Then our RQT graph will display multiple nodes with multiple topics being published to and subscribed from, which will allow for a more accurate depiction of the satellite.

VI. Summary of the State of This Project

As of the beginning of November, we have established a general state machine for the satellite as a whole as well as subsystem state machines for the attitude, communications, and power circuit boards.

Over the course of the semester, we have accomplished:

- Understanding what the client needs
- Communication with the client about their expectations
- Agreeing on state machine design concepts with client and mentor
- Identification of the appropriate tools to use for implementation
- Learning how to use the tools
- Continuous collaboration with the client and mentor as changes are made
- Configuration of development environment and tools

We are currently in the early stages of implementing the general state machine in ROS Kinetic using C++. Once we approach more stable versions of the Power, Attitude, and

Communications state machines, we will be able to implement those and incorporate them into the general state machine as appropriate.

VII. Future Work for This Semester

We have created a Gantt chart to organize the work to do for the rest of the semester. The Gantt chart is a bit difficult to read and interpret by itself, so we included the timeline that the chart is based off of.

1	Finalize Communication State Machine Design	16 days	Thu 10/19/17	Thu 11/9/17	Angie
2	Finalize Attitude State Machine Design	16 days	Thu 10/19/17	Thu 11/9/17	Courtney
3	**State Machine Implementation**	**15 days**	**Tue 11/7/17**	**Sat 11/25/17**	
4	Sleep	3 days	Tue 11/7/17	Thu 11/9/17	Courtney
5	Detumble	3 days	Tue 11/7/17	Thu 11/9/17	Courtney
6	Engage Solar Panels	3 days	Tue 11/7/17	Thu 11/9/17	Solomon
7	Boot	3 days	Tue 11/7/17	Thu 11/9/17	Angie
8	Power Board	6 days	Thu 11/9/17	Thu 11/16/17	Solomon
9	Attitude Board	6 days	Thu 11/9/17	Thu 11/16/17	Courtney
10	Communications Board	6 days	Thu 11/9/17	Thu 11/16/17	Angie
11	Recovery Mode	3 days	Thu 11/16/17	Mon 11/20/17	Solomon
12	Emergency Mode	2 days	Fri 11/17/17	Mon 11/20/17	Courtney
13	Low Power Mode	3 days	Thu 11/16/17	Mon 11/20/17	Angie
14	System Check	6 days	Mon 11/20/17	Sat 11/25/17	Courtney
15	Testing	10 days	Mon 11/27/17	Fri 12/8/17	Everyone

Figure 5. Timeline for the rest of the semester

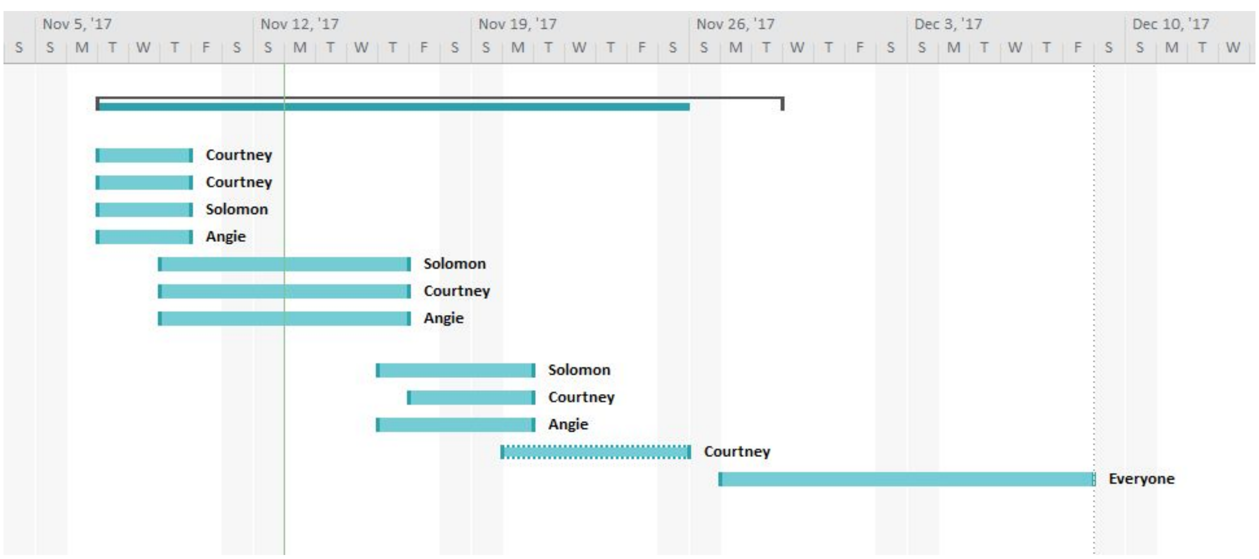


Figure 6. Gantt chart for the timeline for the rest of the semester

Once we complete the state machine implementations in ROS, we can begin implementing the power modes that use the various circuit boards.

Currently, our client has not started designing a Data Collection/Payload circuit board, but in late October it was decided that there will eventually be a separate Payload board. Since we do not have a board design and it is not our place to design the board for our client, we did not specify functions for the Payload subsystem, but we created a separate power mode for it so that it may be integrated later.

We are in the process of implementing the subsystems state machines each team member individually designed. We will merge those together when implementing the different power modes, recovery mode, and the system check.

VIII. Glossary

AX.25: Amateur X.25; a data link layer that is designed for use on amateur radio packet network which has the nearly global coverage.

Detumble: Stop uncontrollable tumbling of a spacecraft.

GUI: Graphical User Interface; an intuitive way for users to interact with the software.

ROS: Robot Operating System; an operating system that is useful for creating state machines and simulations.

RQT: A GUI that displays the network of ROS processes that are processing data together. For our purposes, this includes nodes, messages, and topics.

Sputnik I: The first satellite launched by the Soviet Union in 1957. It transmitted electronic signals to the Earth.

IX. References

Petrescu, Marius, and Victor Toth. “AX.25 Amateur Packet Radio as a Possible Emergency Network.” *Studies in Health Technology and Informatics*, vol. 77, Feb. 2000.

Smith, K.N. “Sputnik 1 Launched The Space Race 60 Years Ago Today.” *Forbes*, Forbes Magazine, 4 Oct. 2017, www.forbes.com/sites/kionasmith/2017/10/04/sputnik-1-launched-the-space-race-60-years-ago-today/.