CySat

DESIGN DOCUMENT

Team Number: 49

Client: Dr. Jones / M:2:I

Advisor: Dr. Jones

Bryan Friestad - Team Lead / EPS Lead

Kyle Muehlenthaler - UHF Lead

Ryan Hansen - SDR Lead

Talon Stromgren - Circuit Design Lead

Chase Kirchner - Ground Station Lead

Xiangzhu Yan - ADCS Lead

Team Email: sdmay20-49@iastate.edu

Team Website: http://sdmay20-49.sd.ece.iastate.edu/

Executive Summary

Development Standards & Practices Used

Knowledge of digital and analog circuit measurement and analysis. Knowledge of PC104, FPGA, and CPU circuit component communication. Knowledge of I²C and JTAG communication standard. Fluency in C with embedded systems specialty. Python will be used to write software for the ground station and software-defined radio. Knowledge of radio frequency and antenna design. The CubeSat standard was developed in 1999 by Stanford University and California Polytechnic State University. This is the standard that defines the satellite for our project. Specifically, it determines the volume and mass constraints. The CubeSat standard is intended for satellites in low earth orbit.

Summary of Requirements

- The onboard computer must be able to communicate with all other satellite subsystems
- The onboard computer can remotely talk to the ground station
- Software-defined radio can capture and store scientific data
- The onboard computer can perform health checks of all subsystems
- Unit tests for each subsystem
- Onboard computer software to complete the main mission
- Conduct stack tests and final assembly
- Mock launch success

Applicable Courses from Iowa State University Curriculum

- CPRE 185/186, 281, 288, 488
- COMS 309, 326
- EE 230, 330

New skills/knowledge acquired that was not taught in courses

The only thing new that we have come across on this project so far is the process involved with working with a team that is much larger than in any of our previous project classes. The CySat project has many people who have worked on it to some degree and so finding the right person who has knowledge on a specific subject for continued development can be difficult at times since guidance is not immediately available within our senior design group. We will update as progress continues.

Table of Contents

Document Acronyms

M:2:I - Make 2 Innovate
OBC - Onboard computer
EPS - Electrical power system
ADCS - Attitude determination and control system
SDR - Software-defined Radio
LNA - Low noise amplifier
FPGA - Field programmable gate array
ISS - International Space Station
LEO - Low Earth Orbit
CRS - Commercial Resupply Service
PCB - Printed circuit board
UHF - Ultra-High Frequency
COTS - Commercial Off The Shelf

Document Figures

1. Introduction

1.1 ACKNOWLEDGEMENT

Significant funds have been provided for this project by the Iowa Space Grant Consortium, Make 2 Innovate (abbreviated M:2:I) and NASA, as well as other donors. The M:2:I team will also provide expertise and labor on other physical aspects of the project.

1.2 PROBLEM AND PROJECT STATEMENT

M:2:I, a group on the Iowa State University campus has been in the process of developing a CubeSat to be put into orbit around Earth. A CubeSat is a small, uniformly sized satellite which will contain a scientific or research component. M:2:I has enlisted the help of our senior design team to write software for, test, and integrate the various components of the CubeSat (named CySat). In order to get the satellite into orbit, we have to complete these tasks by a deadline in the early spring, so we can hand the satellite off to NASA, who will facilitate the launch.

According to CySat's mission statement, the purpose of the satellite is to demonstrate the technology of the satellite and the software running on it. The satellite will have a radiometer onboard, which will be used to collect moisture data from the surface of the Earth. The data collection and processing via the software-defined radio will provide insight into previous thesis work and pave the way for additional research in future satellites. M:2:I hopes to use the technology and lessons learned from CySat-1 on future satellites, which may have missions such as surveying asteroids or other space objects.

1.3 OPERATIONAL ENVIRONMENT

The end product will have to endure the intense vibrations of taking off from Earth in one of the International Space Station (ISS) resupply trips. From there, the satellite will be released from the airlock into Low Earth Orbit (LEO) where it will communicate the data it collects with a ground station in Ames, Iowa. These constraints impose great physical requirements that our team will not deal with very much.

1.4 REQUIREMENTS

- The ground station in Ames, Iowa must be functional and able to communicate with the satellite
- The satellite must not begin communicating with the ground station until 30 minutes after it has been released from the ISS
- The EPS must provide power to the other subsystems of the Satellite
- The OBC must delegate processing time to the various subsystems to ensure all necessary tasks are completed in a timely manner
- The ADCS must be able to orient the satellite

1.5 INTENDED USERS AND USES

The end-user of this product is the M:2:I group and the members of it who will collect and use the data transmitted by the satellite. It will be equipped with a radiometer, which will collect data about the water levels on Earth.

1.6 ASSUMPTIONS AND LIMITATIONS

Assumptions

- The satellite will deorbit and burn up after roughly 6 months in LEO
- If the satellite can pass our vibration tests, then it will withstand the launch to the ISS

Limitations

- The physical dimensions of the device are set to 3u (one u is 10cm x 10cm x 11.35cm)
- The satellite must not weigh more than 4kg (1.33 kg per u)
- We have a strict deadline in order to go up on the CRS2 NG14 supply launch

1.7 EXPECTED END PRODUCT AND DELIVERABLES

- SDR carrier board designed and fabricated
- Voltage boost board designed and fabricated
- LNA board designed and fabricated
- OBC communicating with EPS, ADCS, SDR and UHF transceiver
- Ground Station UI
- Ground Station communicating with OBC
- Dry fits 1, 2, and 3 (done by M:2:I team)
    - Dry fit 1 is physical constraint testing
    - Dry fit 2 is wiring
    - Dry fit 3 includes power-up sequence
- Software beta version - due week of February 14th
    - Demonstrates all critical tasks, including health check and ground station to OBC communication. Some systems may still not be integrated.
- Software Version 1.0 - due week of March 13th
    - Can complete the entire mission and mock launch, which will be conducted by M:2:I. Certain aspects must be simulated, such as antenna deployment.
- Assembled Satellite - due mid-April 2020
    - To be handed off to NASA for inspection at this time.

2. Specifications and Analysis

2.1 PROPOSED DESIGN

Each major subsystem of our project with a focus on the proposed design is delineated below.

ADCS - ADCS refers to attitude determination and control system, it's controlled by OBC through I2C serial bus and it includes several important peripheral components such as magnetometer. There are 256 telecommand and telemetry requests can be used to manage those peripheral components, my work is to write C functions for OBC to send those telecommands required by initiate and detumbling process. Those c functions have been built but still not tested yet.

Boost Board - The boost board needs to boost 5 volts up to 8 volts. The circuit uses a chip from Texas Instruments that can handle our specific power requirements. The circuit has been made and the printed circuit board has been ordered. We hope to receive it soon so proper testing can be done.

SDR Carrier Board - The carrier board interfaces between the SDR and the LNA board. It was created before, but we redesigned some of it because we needed a 10 pin connector on it instead of a 5 pin connector so we can have more functionality. It has not been reordered yet, so we will have to wait to test it until after the winter break.

SDR - The functional requirements for the SDR component is comprised of to being able to take signals in, convert it to data that can be saved to a file, and then transfer that data over UART to the OBC for transmission to earth. Communication over UART with the SDR component and a standalone serial USB connected computer has been tested and verified as working. Continued development on the SDR component will make it possible to communicate between the SDR and OBC. Once successful communication between the SDR and OBC is established, development can move onto getting a signal from the radiometer to feed into the SDR and be processed.

UHF Transceiver/Antenna - Tested communication over UART between Laptop and UHF Transceiver. The laptop will be replaced by the OBC running C Code to configure the UHF Transceiver in future tests. We are also in the process of building a testing feature to see the data that comes through the Transceiver and antenna, this will also be able to test the ground station when ready.

EPS - The software running on the OBC which controls the EPS needs to ensure that the subsystem passes all health checks required by the M:2:I team. Additionally, power consumption should be observed to ensure that the satellite is not draining its resources too quickly. The OBC software needs to be able to enable and disable power to certain components by controlling the PC-104 pins driven by the EPS. So far, I have written one function which will request the software version of the EPS microcontroller via the I2C bus and return it to the OBC. I have not yet tested this function with the actual components.

OBC - The OBC runs nearly all of the software for the satellite, except for the scientific calculations performed by the SDR. Nearly all of the requirements for the OBC component of the satellite are functional requirements needed to complete the mission. So far we have started some development on temporary, cheap boards with the same chipset. This is because the actual OBC is very expensive and we would rather damage a cheaper, disposable board.

Ground Station - The ground station will allow Earth (M:2:I) to communicate with the satellite. It will be able to request scientific data, health checks, battery level, telemetry data, and receive the beacon. It communicates through serial input and will be developed using the discovery board. C code will be written to handle the serial commands in order to send and receive data from the satellite. Extensive testing will be required to make sure the satellite is able to properly handle different commands.

2.2 DESIGN ANALYSIS

Each major subsystem of our project with a focus on the design analysis is delineated below.

ADCS - C functions for all Telecommand and Telemetry requests needed for ADCS initiate and detumbling process. Those c functions can perform basic telecommand and telemetry requests such as setting Unix time or measuring wheel speed. These functions cannot be tested yet, but we have used the IMU chip to test the I2C communication between OBC and slaves and have been successful. Both are based on the same principle, so these c functions will work with high probability, except for a few possible minor problems.

Boost Board - A circuit that met the power requirements was created using a Texas Instruments chip. The circuit was then made in a computer program and transferred to a printed circuit board layout where everything was connected together. This board has just been ordered, so there has been no physical testing on it yet.

SDR Carrier Board - This was created before we joined the team, but some revisions had to be made. We decided to use different connectors so the board had to be slightly redesigned and all the components had to be rerouted.

SDR - The previous semesters Computer Engineering 488 class had done some work with this component before our team was handed this project. Navigating and making sense of the work that had been done ahead of time as well as doing our own testing to ensure understanding of each level of complexity took a while. Runtime scripts had to be verified, UART communication scripts had to be completed and tested, and the development of scripts on separate components for UART communication had to be developed. Initial testing of UART communication exposed some issues with data mishandling. Continued development for control flow for SDR component operation and communication between the OBC and the SDR will be the next major undertaking.

UHF Transceiver/Antenna - Configured Transceiver to receive data at the specified frequency. Set the beacon value to “Hello World”. Connected a dummy antenna to allow testing without fear of frying the board. Commands are written to be sent straight to the UHF Transceiver for configuring the values while in space, as well as sending direct to the Antenna. These are to be sent to the antenna so the data can be received by the ground station.

EPS - So far, little has been written for the EPS software because most of the effort has been directed towards getting the OBC able to communicate with the other components of the satellite. However, some software has been written to initiate data collection from the EPS via I2C. It has not been tested yet.

OBC - For the OBC, some software has been written to experiment with I2C and UART communications. We have been mostly working a cheap, disposable development board to reduce the risk of damaging the satellite's final parts. After testing some of these communication protocols, we would like to begin communicating with the subsystems of the satellite, such as the ADCS, EPS and UHF transceiver. In this situation, to reduce risk, we want to use the PC-104 bus to physically connect the components of the satellite, since this is a recognized standard and is integrated on the boards. So, we have ported the software from the development board back to the satellite's OBC for further testing. This will require careful programming to ensure that we don't accidentally fry any of the boards by having bus contention or other hardware issues. Luckily, I2C does not allow for bus contention.

One of the strengths of developing in this manner, where we write code on one device, test it and then port it to another device is that it leads to far less opportunity for mistakes with expensive satellite hardware components. One weakness however, is that the two devices are not exactly the same, no matter how similar the chipsets are. Porting embedded systems software from one device to another is sure to produce some unforeseen issues. We have not yet run into this issue, but I am certain we will once we begin to test ported software.

Ground Station - Began implementation of the python interface to communicate with the satellite from Earth. Leveraging the Python tkinter package for creating an easy-to-use interface. Writing the C functions to handle serial input from the python application. Ramping up took some time due to having to meet with different members of M:2:I and sorting through documentation to determine the best course of action.

2.3 DEVELOPMENT PROCESS

The development process we are following is a Waterfall model, which is a breakdown of project activities into linear sequential phases, where each phase depends on the deliverables of the previous one and corresponds to a specialization of tasks. The project is divided into several subsystems such as OBC, SDR, EPS, ADCS Ground Station, and Boost Board, more details can be found in Figure 2.2. Each team member is responsible for at least one of these. Every team member needs to work on their subsystems, those works include building, debugging and testing, etc. After this series of groundwork has been completed, the next step is to integrate the subsystems, which is followed by a series of interface/integration testing. Finally, with the completion of system-level testing/Acceptance testing, the project will come to an end.

2.4 DESIGN PLAN

Our design plan is shown in figure 2.1. We divided this project into several subsystems shown in figure 2.2 and assign them to each team member. After the work and component testing are done for those subsystems, we plan to work on integration and System level testing to finish this project.

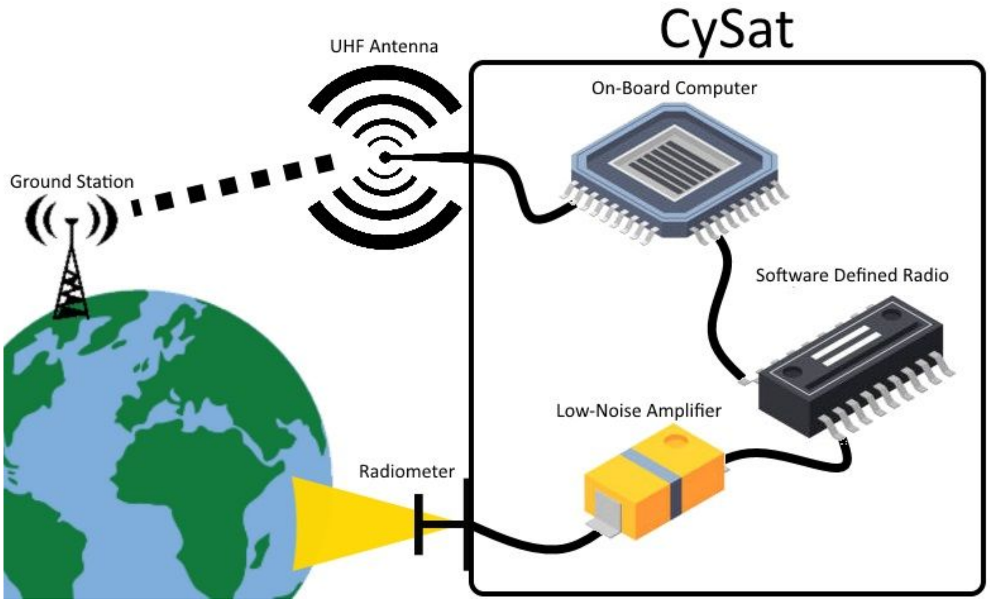

Figure 2.1 - Simplified design overview of components related to CySat's core mission operation.

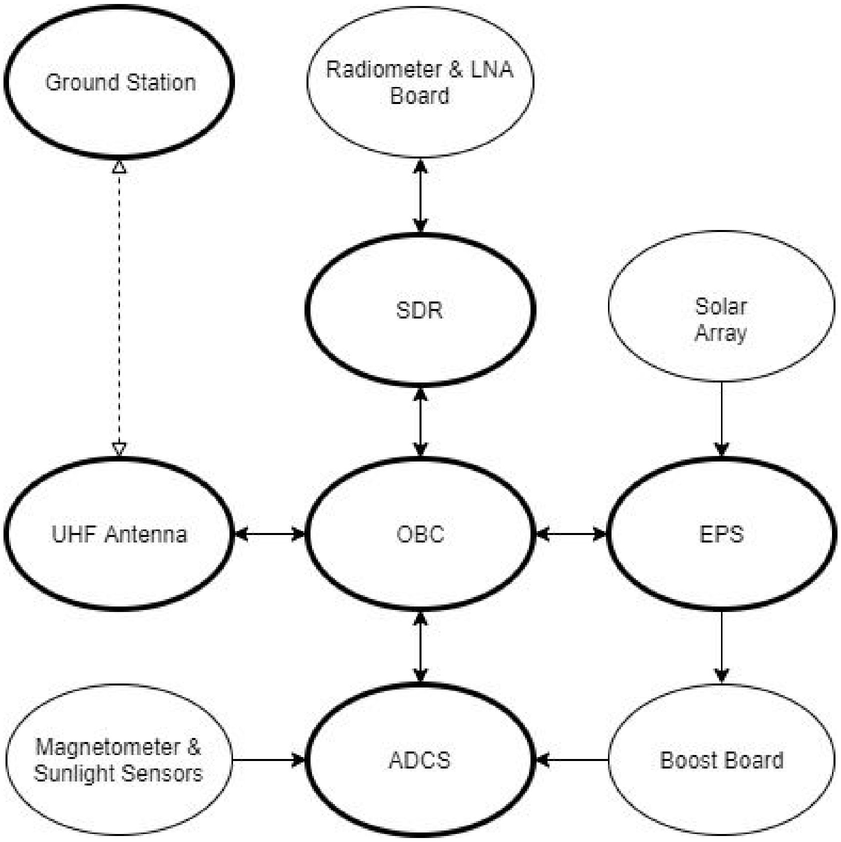

Figure 2.2 - Simplified CySat component subsystem diagram. Major subsystems with software relevance bolded.

3. Statement of Work

3.1 PREVIOUS WORK AND LITERATURE

Iowa State University's M:2:I team has been working on developing a CubeSat for many years now. In 2017 the payload was approved by NASA and supporting hardware was researched. During each semester since 2017, students have optimized the design and chosen the hardware that would meet their mission requirements. Our task as the software team is to integrate all of the various electronic systems that have been acquired and develop software for the satellite that will allow it to complete its mission.

Much of our design implementation will be based on the documentation in the CySat CyBox archive that was already created by the M:2:I team prior to our team's involvement with CySat 1.0. In 2017, NASA published *CubeSat 101: Basic Concepts and Processes for First-Time CubeSat Developers*. This article details a lot of basic information about CubeSat development. Utilizing parts of this literature and the M:2: I documentation allowed our team to get up to speed with the overall status of CySat's development.

3.2 TECHNOLOGY CONSIDERATIONS

Many CubeSats, as will CySat, use a PC-104 stack for compact bussing of data signals between the various satellite components. Because this standard is widely used, finding off the shelf components that interface with the PC-104 stack is not difficult. Many components like the OBC and ADCS have PC-104 built into them for ease of integration.

Once CySat is launched into space it can no longer be serviced by our team. Thus, we have to be particularly cautious in selecting quality components, which ultimately drives up the cost of CySat, and be sure to integrate them properly prior to launch.

Some of the hardware needed for CySat is not available as a complete off the shelf drop-in solution. This particularly is the radiometer payload and SDR component combination. Our team has worked to take several off the shelf components which may or may not have a PC104 connection and develop our own PCBs to integrate the various systems. Several custom PCB s with suitable dimensions and integrated hardware are required to fulfill CySat's requirements.

3.3 TASK DECOMPOSITION

CySat has various components that will all need to be worked on separately before the entire system is integrated. Communication established with the OBC will be the first task that will need to be completed. Next, we will need to establish communication from the OBC to each of the various subsystems (UHF Radio Antena, SDR, ADCS, and EPS). After communication with each major component is established individual subcomponent software development will take place. Each subsystem will go through rigorous testing of all major functions and then be integrated for full system testing. Once successful, the development will continue for items deemed as "quality of life" tasks.

3.4 Possible Risks And Risk Management

Our team has a very basic knowledge of embedded systems and software design. Specialized systems, like that used on CySat, requires learned knowledge and understanding before jumping straight to development. The components used on CySat are quite expensive and unfamiliarity with them can lead to mishandling or possibly destroyed hardware. Our team will be working closely with the individuals who selected the various components to gain familiarity and confidence before working with the hardware.

Since CySat is a longstanding project, there is a wealth of information to digest about the structures already set in place. Our team will meet weekly with the M:2:I group and our project advisor to keep our team in the loop with other project developments as well as provide us time for seeking knowledge on the various subsystems from those with experience.

CySat requires custom hardware to be built in order for it to complete its mission requirements. Our team will be tasked with creating the necessary PCBs CySat will require. Since our team has no experience working with custom PCB creation, it is likely that mistakes will be made along the way. We will be focusing on getting the custom hardware designed, built, and tested earlier on in our project timeline so that any mistakes made can be corrected without becoming a time-critical task.

3.5 Project Proposed Milestones and Evaluation Criteria

Since CySat has so many systems, project milestones will be very important for tracking the overall development progress of various components. Startup and functioning “Hello World” programs for each major component is our first milestone. Following successful “Hello World” implementations, development for communication between the OBC and each subcomponent will be another major milestone focused on.

One of two project critical milestones will be the completion of hardware and software that can collect, process, and save scientific data from the radiometer on the SDR component. The next critical milestone will be to transfer the scientific data to the OBC and relay that information over the UHF antenna to the ground station.

Final operational tests of all CySat’s components integrated together and successfully performing a mock launch scenario will be the last major milestone for our team. Verification of each component’s successful operation will be crucial before sending CubeSat to space. Subsequent “quality of life” development will be schedule dependent but is also an intended milestone.

3.6 Project Tracking Procedures

Our team will organize tasks by utilizing the issue ticket system on GitLab. We will meet as a team and establish tasks that are to be worked on in the coming weeks. Our team has received an abbreviated list of tasks from M:2:I that we will enter into GitLab as our baseline tasks. We will establish other tasks based on more specific items relating to hardware or software requirements surrounding those abbreviated tasks. These tasks will be assigned to the team member they seem most appropriate for and progress will be tracked by noting the completion of each issue on GitLab. Subsequent tasks will be suggested in our weekly meetings by M:2:I or our project advisor and added to our GitLab for tracking and completion.

3.7 Expected Results and Validation

Since CySat will be headed to LEO, an environment that cannot be simulated easily or accurately on Earth, there will be a point in which our tests will no longer be able to accurately replicate conditions. We will only be able to expose the system to potential conditions in which we can replicate. Final testing of CySat's software will have to be rigorous and thorough to minimize as many of the unknown conditions as possible.

In conclusion of our development, CySat should be able to autonomously control its subsystems, take measurements using its radiometer, and relay that information over radio waves to a communication station. The successful operation of these tasks will conclude our portion of the development of CySat. Conclusively, we should have a fully operational and tested CubeSat that will be ready to launch into space to begin its mission. Initial communication and data retrieval with CySat after launch will be the testament of a successful design.

4. Project Timeline, Estimated Resources, and Challenges

4.1 Project Timeline

August

- Team formed
- Assign Tasks and roles

September

- Achieve an understanding of our roles
- Understand documentation
- Established meeting times

October

- Trial of equipment and troubleshooting
- Update of documentation
- Stack tests as needed

November

- Hello world functionality among our subsystems
- ConOps discussion (incremental mission tasks)
- Establish definitions of mission-critical components.
- Written timeline of critical software

December

- Boost board order
- After Boost Board arrives decide final revisions.
- Update documentation

January

- All boards on the final revision
- 25-50% unit functionality

February

- Beta version of the software. This should be able to complete critical components but not necessarily the whole mission.

March

- ConOps discussion (incremental mission tasks)
- 50-75% unit functionality
- Software version 1.0
- Stack tests as needed
- Update functionality

April

- 100% unit functionality. Final software version.
- Stack tests as needed, more in-depth
- Begin mock launch procedure
- Update documentation

May

- Mock launch procedure
- Mission-life procedure

This timeline is to make sure that our development schedule aligns with the June handoff.

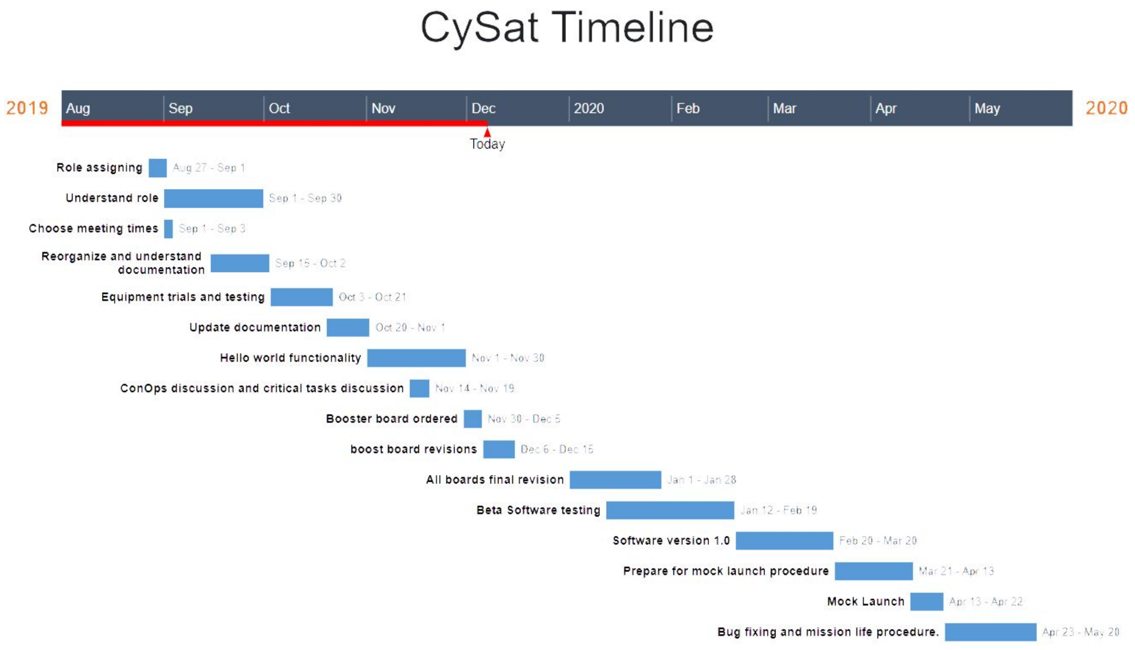

Figure 4.1- CySat Timeline Gantt Chart

4.2 FEASIBILITY ASSESSMENT

After debate with the various components involved with this project, each system has been given its own feasible timeline to work forward from and with this, we are able to more accurately move forward. Since we are no longer looking at the early handoff we are more able to achieve the set milestones and schedules are feasible for each subsystem. We are also working with the M:2:I schedule which has milestones that we built our timelines around, these being the Mock launch and handoff date. These will be working by the time of the M:2:I requirements and we will be able to split to allow extra work in any component that falls behind.

4.3 PERSONNEL EFFORT REQUIREMENTS

Earth-Satellite communication: Kyle and Chase **December-January**

- This task is to be completed between December and January, this will take a combined effort from both the Ground Station and the UHF Satellite. For this to be considered complete the OBC should be able to receive data through the UHF Transceiver from the

Ground Station and the Ground Station should be able to receive information from the Satellite.

Booster Board: Talon **December**

- This task needs to boost the 5V source to 8V. This will be tested after it comes in to ensure that it completes this task, as well as test functionality for the enable pin.

Subsystem Health: Bryan, Xianzhu, Ryan, Chase, Talon, Kyle **January-February**

- After assembly, the OBC will be able to send health requests to each system and each system will be able to respond back to the OBC.

Startup Sequence: Kyle, Xiangzhu, Bryan, Chase **January-April**

- This sequence needs to wait 30 minutes (EPS) and then start detumbling sequences(OBC->ADCS), after detumbling is finished the beacon starts from the UHF system(UHF Transceiver -> UHF Antenna).

Main Operating Mode: Bryan, Ryan, Talon, Kyle **January-March**

- The main operating mode takes place in two parts, the payload mode, and the ground station mode. Communication between the payload, SDR, OBC, and UHF Antenna must work as described for mission requirements.

Diagnostic Mode: Whole Team **April**

- After startup sequence finishes CySat can enter diagnostic mode. in diagnostic mode each individual component needs to be able to send a health check, then components should be able to perform required tasks. This point should also include being able to run the final mission.

4.4 FINANCIAL REQUIREMENTS

All funding has been provided by M:2:I and its partners for this project. Hardware and structural components have either been bought, machined in house, or donated. The CySat project has an overall estimated value of over $120,000.

5. Testing and Implementation

NASA requires extensive testing in order to deploy a satellite into space. Testing requirements include functional, non-functional, vibe, and launch testing to name a few.

Testing Plan:

1. Define the needed types of tests (unit testing for modules, integrity testing for interfaces, user-study for functional and non-functional requirements)

2. Define the individual items to be tested

3. Define, design, and develop the actual test cases 4. Determine the anticipated test results for each test case

5. Perform the actual tests

6. Evaluate the actual test results

7. Make the necessary changes to the product being tested

8. Perform any necessary retesting

9. Document the entire testing process and its results

5.1 INTERFACE SPECIFICATIONS

Subsystems will interface with the OBC throughout the satellite path. The OBC will also be interfaced with Ground Control for data collection and satellite status checks.

5.2 HARDWARE AND SOFTWARE

Vibration Testing: Requires a specialized chamber to verify that the physical components are structurally sound.

Regression Testing: Confirms that a recent code push doesn't negatively affect existing features.

-It will be completed through automated and manual procedures.

-Automate testing processes completed within Gitlab after deployments are made.

5.3 FUNCTIONAL TESTING

Examples include unit, integration, system, and acceptance testing

- Unit testing for each subsystem - Making sure each subsystem is fully functioning and communicating properly.
- Vibration testing - Making sure all physical components will stay pieced together on the space shuttle

5.4 NON-FUNCTIONAL TESTING

- Testing on the actual subsystem components to check for any compatibility issues.
- Usability testing is done for ease-of-use for the ground station interface.
- Testing the enable pins on all boards for making sure there won't be any unnecessary power consumption.

5.5 PROCESS

All subsystems are just starting to go through basic testing. Instead of using the actual expensive components of the satellite, we are using development boards with the same type of processor as their respective subsystems. This is so that we do not risk making any mistakes early on that could potentially destroy some expensive components. After there has been thorough testing done with the subsystems on their development boards, testing will proceed on the real satellite components. This will help us know of any changes we need to make to the code developed using the development boards since there could be errors such as compatibility issues.

5.6 RESULTS

Below are some initial testing results from limited testing we have been able to do so far.

- UHF System-
    - Failures-
        - Initial command usage attempts. Needed to configure the send commands to correct the Carriage Return value.
        - Unable to test the UHF Antenna based on its design. We cannot deploy the antenna for testing because we can't put it back after deployment.
    - Successes-
        - Able to send commands to configure and change the UHF Transceiver configurations
        - Have plans to set up an antenna testing system for the UHF system as well as a ground station
    - Learned-
        - How to apply various commands to the UHF system for configuration
        - How the transceiver takes input commands
        - This transceiver will take in data from the antenna and push it out to our OBC. And any value that is not a specific command identified in the ESTCC document it will push it to the antenna.
- SDR
    - Failures
        - The initial creation of runtime files did not call properly and failed.
        - Initial testing of runtime script using UART communication failed to sense data in the input buffer.
    - Successes
        - PuTTY to SDR communication and printed “hello world” as expected.
        - Python script for SDR and OBC equivalent communicating over UART.
    - Learned
        - Correctly setting up runtime files for Unix systems.
        - Serial buffers need to be cared for properly upon setup.
        - Clearing buffer before use gets rid of any erroneous data.
    - Challenges
        - Custom PCBs used for rerouting signals needed for SDR communication with OBC still being developed.

        - Restricted to testing UART communication using a laptop with a serial USB simulating the OBC or development test board acting as temporary OBC.
- OBC
    - Failures
        - Setting up I2C communications initially led to many to many failures due to hardware issues.
        - Malformed data parsing on the receiving side of the I2C bus.
    - Successes
        - After ensuring a correct hardware connection, we were able to properly receive data on the OBC development board.
        - Able to write functions to send data over UART to a virtual com port on a computer.
        - We were able to port the UART and I2C code from the development board to the real, runtime OBC.
    - Learned
        - We learned more about reading from datasheets.
        - How serial communications work from a hardware and software perspective with first-hand experience.

6. Closing Material

6.1 CONCLUSION

As of December 8, 2019, we have gone through a rigorous process of trying to understand and organize all of the documentation for the project and its subsystems. Our goals include getting communication with the On-Board Computer, getting all subsystems to communicate with each other, and achieving thorough unit testing functionality. We have installed the basic software needed on a few different laptops so we can try different ways of communicating with the OBC. Recently, we have been focusing on continuously developing each respective subsystem. Work on creating the Ground Station application has gone underway to communicate with the satellite from Earth. We have a work block time each week to collaborate with each other to discuss any progress or issues that may come up throughout the week. This is the best way to achieve our goals because everyone will be able to bounce ideas off of each other and we will be spending a lot of time in the lab. The written software will be reviewed by other members of the team to maintain coding standards.

6.2 REFERENCES

Our references include all documentation that we have regarding the specific components of the satellite. These are manuals that are provided by the companies where we purchased the components.

6.3 APPENDICES

One large problem we encountered was that the specific files we need to work with the OBC are not version controlled well by Eclipse and it completely messes up the project if you try to import it. There also may be an issue with our JTAG to USB converter, but we have yet to test that.