CySat: Satellite Mission Design

DESIGN DOCUMENT

Team Number: sdmay21-25
Client: M2I
Adviser: Dr. Phillip Jones
Team Members/Roles:
Alexis Aurandt - OBC Lead and Payload Sub-Lead
Alex Constant - Ground Station Front-End Lead
Chandler Jurenic - Payload Lead and OBC Sub-Lead
Jeffrey Richardson - ADCS Lead/Team Lead
John Lenz - Radio Lead
Scott Dressler - EPS Lead
Team Email: cysat-f2020@iastate.edu
Team Website: http://sdmay21-25.sd.ece.iastate.edu
Revised: October 4th/1.0

Executive Summary

Development Standards & Practices Used

- NASA and CubeSat hardware standards
- Consistent code commenting and documentation for software.
- Code reviews by team before merges
- UART
- I2C
- Python 3+ for Ground Station

Summary of Requirements

- Needs to power up after been deployed from the International Space Station
- Needs to stabilize and point itself towards earth
- Needs to take soil moisture readings from Earth via a microwave radiometer
- Needs to transmit data back to the ground station in Ames, IA
- Needs to be able to collect data for its orbit life (6 months)
- Needs to meet NASA's CubeSat requirements

Applicable Courses from Iowa State University Curriculum

List all Iowa State University courses whose contents were applicable to your project.

New Skills/Knowledge acquired that was not taught in courses

List all new skills/knowledge that your team acquired which was not part of your Iowa State curriculum in order to complete this project.

Table of Contents

List of Figures/Tables/Symbols/Definitions

ADCS – Altitude Determination and Control System

EPS – Electrical Power System

M:2:I – Make to Innovate

OBC – On-board Computer

SDR – Software Defined Radio

UHF Radio- Ultra High Frequency Radio

CubeSat – A standard, miniaturized scientific satellite comprising connected 10 x 10 x 10 cm cubes

1 Introduction

1.1 Acknowledgement

We would like to thank last year's senior design team for providing handoff documentation, providing guidance, and for being available for questions. Additionally, we would like to thank Dae-Young Lee for his expertise and his guidance with the ADCS sub-system. Finally, we want to thank Dr. Jones for meeting with us weekly and for giving us guidance.

1.2 Problem and Project Statement

The CySat is a cube satellite, which is a standardized form of miniaturized scientific satellite. CySat will be deployed from the International Space Station, after which it will orbit the earth for approximately 6 months with the goal of collecting and relaying soil moisture data back to our ground station in Howe Hall on the Iowa State University campus.

The CySat is a student project started and operated by M:2:I. The sole purpose of this project is to get students engaged in a hands-on project, and the driving force behind the project is students wanting to develop a satellite that will be launched into space. Originally, this project was only for Aerospace engineering students, but M:2:I soon realized they needed Computer, Electrical, and Software engineers to take care of the onboard electronics. This is where our senior design team comes in. There are many different subsystems that need our expertise.

The CySat comprises six subsystems. These subsystems control the satellite's orientation with respect to the Earth, collect and process data, and stream the data back to Earth during communications windows. The subsystems are as follows:

- The OBC communicates with all the other subsystems and ensures that the satellite is operating according to specifications.
- The ADCS stabilizes and points the satellite toward Earth.
- The EPS regulates power from the solar cells.
- The radio relays data and commands between the CySat and Earth.

- The ground station receives data, sends commands, and acts as the interface between the M:2:I satellite team and the satellite.
- The payload uses an SDR to gather soil moisture readings from earth.

1.3 Operational Environment

The CySat will be launched into space, and it will orbit the Earth. This necessitates that the internal hardware of the CySat, as well as software running on that hardware, are robust and capable of failure recovery with minimal loss.

1.4 Requirements

1.4.1 Functional Requirements

- Must power up after been deployed from the International Space Station
- Must stabilize and point itself towards earth
- Must take soil moisture readings from Earth via a microwave radiometer
- Must transmit data back to the ground station in Ames, IA
- Must collect data for its orbit life (6 months)
- Must meet NASA's CubeSat standards and regulations
- Must receive and execute commands issued by the Ground Station
- Must successfully deorbit at the end of its lifespan

1.4.2 Nonfunctional Requirements

- Ground Station UI is performant

1.5 Intended Users and Uses

Our senior design team and the M:2:I team are the end users for this CySat. This CySat is a creative project for students to be able to launch a satellite into space and receive data back.

1.6 Assumptions and Limitations

1.6.1 Assumptions

- We assume correct installation of the sub system by M2I.
- Software on the CySat is of the same version as what we used to implement the subsystem functionality
- Databases for the Ground Station are not yet implemented

1.6.2 Limitations

- The hardware and software of the CubeSat must comply with NASA regulations as well as CubeSat standards, and the hardware must fit within a 10 x 10 x 10 cm cube in the satellite housing.
- Few operation times in lab due to COVID regulations
- Mock launches are the only feasible way of testing the subsystems

1.7 EXPECTED END PRODUCT AND DELIVERABLES

The hardware and software for a CubeSat will be delivered to M:2:I, who will hand the CubeSat off to CubeSat testing for prelaunch testing, after which the CubeSat will be launched to the International Space Station for deployment. The deliverables for this project are each of the subsystems; the expectations for each, as well as their tentative due dates, are laid out below.

The Ground Station (May 03)

The Ground Station software will be installed on a desktop device in Howe Hall. The Ground Station will allow the user to visualize the satellite's location, send commands to the satellite, and receive and store data from the satellite.

The Payload SDR (May 03)

The payload's SDR system will be connected to the satellite via the OBC using UART. This SDR will be capable of collecting data with the radiometer and capable of transmitting the data back to the OBC. Though most of the subsystem will be partially completed throughout the Fall 2020 and Spring 2021 semesters, the finished payload subsystem will be ready by May 3rd.

The ADCS (May 03)

The ADCS must be able to switch between modes of operations such as active and passive detumbling. ADCS will send telemetry data to the OBC to be recorded / sent to the ground station via radio. ADCS will have designated modes of operation for gathering scientific information and transmitting to the ground station based off of orbit.

The UHF Radio (March 29).

UHF Radio subsystem must be able to receive instructions from the ground station. Take the instructions and communicate instructions to OBC. OBC will then send the requested data back to the ground station through the UHF Radio.

The EPS (March 8)

The EPS provides energy to the rest of the satellite and reports its heath to the OBC. It controls these systems differently based on the satellite's current mode of operation.

The OBC (May 03)

The OBC is the heart of the entire CySat. The OBC must be able to communicate to all the different subsystems. It must be able to efficiently read messages and give commands in order to achieve the CySat's purpose of successfully relaying moisture data back to Earth.

Handoff Documentation (May 03)

If this project does not get done on time for whatever reason, the responsibility to finish the project will be on next years' CySat team. If we get into the Spring 2021 semester and realize we don't have enough time, handoff documentation must be created.

2 Project Plan

2.1 Task Decomposition

We will be decomposing tasks by subsystem. Each numbered task is a known requirement of the subsystem, and the lettered subtasks below are tentative subtasks for implementing the required functionalities.

2.1.1 Ground Station

1. Ground Station Communication
    a. Ground Station receives packets
    b. Ground Station interprets packets
    c. Ground Station sends packet
2. Ground Station Data and Command Logging
    a. Create a database to store data
    b. Hook ground station into database to store data
    c. Create a logging schema for Command logging
    d. Create a database to store logs
    e. Hook ground station into database to store logs
3. Ground Station Common Commands
4. Ground Station Custom Commands
    a. Create custom command schema for Ground Station
    b. Implement custom commands
5. Ground Station Visualization
    a. Subsystem Health Check Visualization
    b. Ground Station visualizes satellite's location around the Earth

2.1.2 UHF Radio

1. Radio Communication
    a. Receive and send packets from computer to radio for debugging
    b. Receive and send health check to OBC
    c. Receive and send commands to OBC
    d. Receive and send beacon and packets to Ground Station
    e. Receive and send packets from Ground Station and OBC

2.1.3 EPS

1. EPS Communication
    a. Receive and send commands to OBC
    b. Send health checks to OBC
2. EPS heath checks
    a. Monitor battery temperature
    b. Check voltage and current inputs and outputs
    c. Verify charging status
    d. Track energy levels
3. EPS charge and discharge
    a. EPS can be charged by M:2:I solar cells
    b. EPS can power CySat subsystems
4. EPS battery protection
    a. Battery temperature control
    b. Short-circuit/over-discharge/over-charge protection

2.1.4 Payload

1. Payload Communication
    a. Send and Receive data to/from OBC
    b. Send Radiometer data to OBC
    c. GNU radio to SDR
    d. UART functionality
    e. I2C to LNA board
2. Payload Data Collection
    a. Packet Protocol functionality
    b. OBC transfers data through UHF antenna
    c. Capture mode needs to be completed and tested
    d. Radiometer program to run at startup of Linux subsystem
3. Payload Functionality
    a. Must be powered by EPS
    b. SDR will require OBC to use data collected
    c. Update time
    d. Operated by OBC

2.1.5 ADCS

1. Storing Telemetry data when out of ground station range
2. Mode activation control
    a. Passive Detumble
    b. Active Detumble
    c. Active Comms Control
3. Set magnetometer configuration (compute magnetometer offset and sensitivity matrix)
4. Y-Wheel Ramp-up Test
5. Y-momentum mode commissioning 1 & 2
6. Sun/Nadir sensor test
7. Active comms control
8. 8-bit Health check
    a. Communication with OBC
    b. X,Y,Z rates
    c. Current Altitude
    d. Current Sun sensor
    e. Nadir sensor
    f. Magnetometer deployment
9. Re-entry

2.1.6 OBC

1. OBC Communication
    a. Receive and send info to computer for debugging
    b. Receive and send packets to Ground Station
    c. Receive and send commands to EPS
    d. Receive and send commands to UHF Radio
    e. Receive and send commands to ADCS
    f. Receive and send commands to SDR Payload
    g. Store payload information on file system
2. OBC Optimization
    a. Interrupt based I2C and UART
    b. FreeRTOS investigation
3. OBC Live Updates
    a. Create bootloader that allows live patches
    b. Receive live patches from Ground Station
4. OBC Land Tests
    a. Implement Mock Mock Launch
    b. Implement Mock Launch
    c. Implement Mock Mission

2.1 Risks And Risk Management/Migration

Many of the risks associated with this project are difficult to mitigate, as they are related to the satellite's performance in space. The mitigation for these types of risks is to test extensively, consistently, and well. Below are a few of the risks our team has chosen to identify based on current knowledge of the CySat project.

Risk	Explanation	Estimated Probability
Loss of communication with Ground Control	Risk of losing communication with the ground control station based off of the tumbling of the CySat	10% Mitigated by having passive detumbling modes for when communication cannot be established
Difficulties setting two-way communication with a single radio module	Radio can be set as a transceiver on the CySat, but previous teams have had difficulty with this configuration	60% Mitigated by installing a second radio system on the CySat.
Connecting OBC to other subsystems	When combining separate subsystems there is a possibility of the connection messing up already working functionalities	50% Since the 6 subsystems are communicated through Git, we can just rollback the error the connection caused.
Loss of power	Battery operated CySat could lose power mid operation.	20% CySat has on board solar panels to recharge its battery EPS sends battery status to OBC
Task exceeds expected time	We expect sometime in the project to have a task or tasks which will need to be re-evaluated based off of the difficulty they are currently presenting.	90% Mitigated by using the scrum workflow and by communication blockers throughout the team / Dr. Jones to solve pending issues.

2.2 Project Proposed Milestones, Metrics, and Evaluation Criteria

The milestones and metrics associated with the CySat satellite are based on preliminary estimations of performance. Where numbers are unknown, an 'x' is used to delineate the unknown number – these values are to be determined through discussion with M:2:I and iteration on the project deliverables.

Milestones are roughly grouped by their associated subsystem, and follow below:

Milestone: Ground Station Receives OBC Communication

> **Metrics:** Packets sent/received per second
>
> **Evaluation Criteria**: Ground Station will successfully handle x packets per second with no loss due to concurrency

Milestone: Ground Station Sends OBC Communications

> **Metrics:** Packets sent/second, confirmation of receipt of packets
>
> **Evaluation Criteria**: Ground station may send up to x packets/second, confirms receipt of packet for x% of packets

Milestone: UHF Radio Communicates with OBC

> **Metrics:** Messages will be transmitted to the OBC send back to the UHF Radio, and printed correctly through putty.
>
> **Evaluation Criteria**: UHF will successfully transmit and receive data from the OBC with a packet loss percentage below some percentage

Milestone: UHF Radio Communicates with Ground Station

> **Metrics:** Messages will be transmitted to the Ground Station send through the UHF Radio, and back to the Ground Station, being printed correctly back through the ground station UI.
>
> **Evaluation Criteria**: Messages will be transmitted to the Ground Station send through the UHF Radio, and back to the Ground Station

Milestone: EPS Health Check

Metrics: When prompted, the EPS sends its own health data to the OBC. This data will contain the temperature of the batteries in Celsius, the input and output voltages in Volts, the input and output currents in mili-Amperes, and the battery capacities.

Evaluation Criteria: All included measurements are accurate (including charging and discharging) and can be received by the OBC.

Milestone: Payload's SDR Communicates with OBC

Metrics: The OBC will use UART to utilize the SDR to receive data that the SDR collected using the radiometer application.

Evaluation Criteria: The OBC successfully can display the information that the SDR got from the radiometer.

Milestone: ADCS Communicates with OBC

Metrics: The OBC will use I2C to send commands to the ADCS for transitioning between operational modes. The ADCS will send telemetry data to the OBC to be recorded.

Evaluation Criteria: The OBC successfully can display and store telemetry data from ADCS. OBC will be able to switch modes of operation on the ADCS through I2C.

Milestone: ADCS has multiple operational modes defined

Metrics: Code for operational modes are implemented, such as active and passive detumble.

Evaluation Criteria: Operational modes are defined based off of the operation mode control flow for ADCS. Each operation modes takes in telemetry data as well as OBC command data.

Milestone: OBC Optimization

Metrics: Multithreading through FreeRTOS and/or creating interrupts for UART and I2C will increase the speed and response time of the OBC to outside commuication from other subsystems.

Evaluation Criteria: The response time of the OBC will increase by ~50%

Milestone: OBC Live Updates

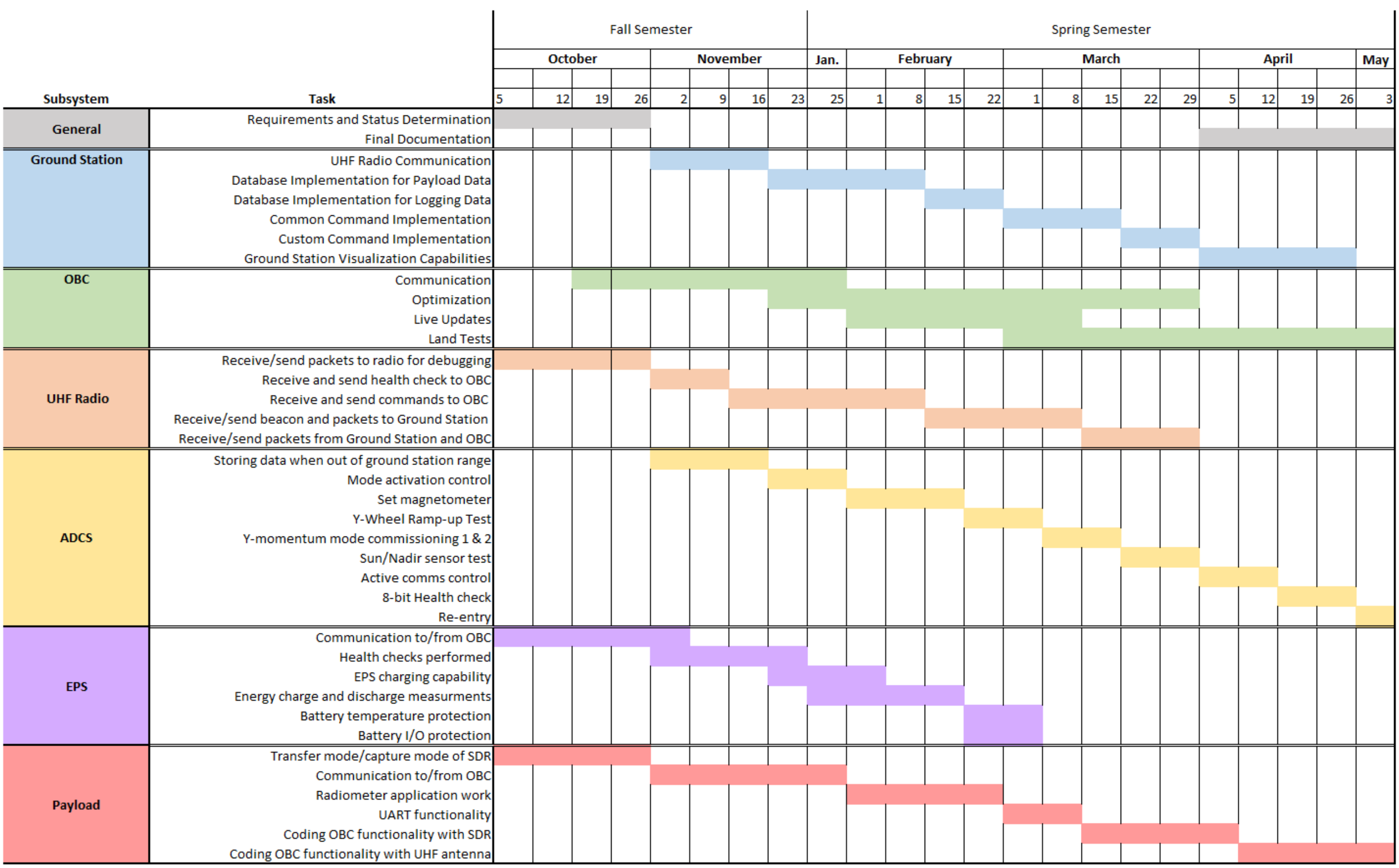
> **Metrics:** The Ground Station will give live patches to OBC through the UHF radio. The Ground Station will use a bootloader to successfully update.
>
> **Evaluation Criteria:** The OBC will successfully accept a live patch from the Ground Station.

Milestone: OBC Mock Test Launch/Mission

> **Metrics:** All subsystems will successfully communicate with the OBC. The OBC will receive commands from the Ground Station via the UHF radio. The OBC is in full control of what each of the subsystems are supposed to do.
>
> **Evaluation Criteria:** All subsystems will perform as desired in space.

2.4 Project Timeline/Schedule

Our project timeline is grouped according to subsystem. This timeline delineates tentative due dates for numbered tasks from section 2.1. As we are using a SCRUM environment with week-long sprints, we afford ourselves the ability to change course and add/subtract both tasks and subtasks while maintaining a structured approach to project completion. Below is the initial, tentative breakdown of tasks:



2.5 Project Tracking Procedures

Our team will use Gitlab for version control. We will use Gitlab's issue boards to keep track of tasks and their relevant commits and branches, and to build and maintain a backlog of tasks. We will also be using slack for scheduling meetings, communicating about tasks, and asking questions.

2.6 Personnel Effort Requirements

Our tentative personnel effort requirements are laid out below, based on our initial understanding of project requirements and knowledge of the previous CySat teams' committed hours:

Substem	Task	Description	Hours
Ground Station	UHF Radio Communication	Sending, receiving, interpreting, and responding to CySat data	20
	Database Implementation for Payload Data	Permanent storage of Payload data, stored chronologically	25
	Database Implementation for Logging Data	Permanent storage of Commands Sent/Received and other Logs	15
	Common Command Implementation	Implementation of common commands to be sent to CySat	20
	Custom Command Implementation	Implementation of custom user command creation and use	12
	Ground Station Visualization Capabilities	Visualization of Satellite/Subsystems	20
	--	Total hours for Ground Station	112
UHF Radio	Receive and send packets from computer into radio for debugging	Hello world, Packet structure, addtional functional	30
	Receive and send health check to OBC	Make sure UHF is running properly	5
	Receive and send commands to OBC	Prompts OBC to access data or communicate with other subsystems	25
	Receive and send beacon and packets to Ground Station	Line of communctation between the satellite and its users	25
	Receive and send packets from Ground Station and OBC	Integration between OBC and Ground Station communcation	25
	--	Total hours for UHF Radio	110
Payload	Transfer mode/capture mode of SDR	Sending/collecting data via the SDR	20
	Communication to/from OBC	Being able to send or receive commands, data, etc between SDR & OBC	25
	Radiometer application work	Getting application to run on embedded Linux start up	20
	UART functionality	UART communication testing and completion	10
	Coding OBC functionality with SDR	Programming the OBC to be able to command the SDR using UART	20
	Coding OBC functionality with UHF antenna	Programming the OBC to transfer data using the UHF antenna	20
		Total hours from Payload	115
ADCS	Storing Telemetry data when out of ground station range	Recording telemetry dat for when the CySat is within range	12
	Mode activation control	Major component of programing the ADCS, flow control for op modes	40
	Set magnetometer configuration	Comput magnetometer offset and sensitivity matrix	12
	Y-Wheel Ramp-up Test	Testing for Y-Wheel Ramp-up Test	12
	Y-momentum mode commissioning 1 & 2	2 stages for Y-momentum mode commissioning	12
	Sun/Nadir sensor test	Testing of sun sensor for determining position	12
	Active comms control	Ground station control operational modes	12
	8-bit Health check	Additional time for health check of the system	15
		Total hours for ADCS	127
EPS	Communication to/from OBC	Update I2C to new version	40
	Health check	New I2C protocols add more parameters to be checked	25
	Charging and discharging	Measure and calculate the energy v. time of the batteries	30
	Battery protection	Change operation based on the data from the health check	15
	--	Total hours for EPS	110
OBC	Communication to all Subsystems	Sending, receiving, interpreting, and responding to all other subsystems	30
	Optimization	Look into FreeRTOS and use interrupts for UART and I2C	30
	Live Updates	Create bootloader that allows for live patches	30
	Land Tests	Implement mock mock launch, mock launch, and mock mission	50
	--	Total hours for OBC	140

Hourly time commitments are based on our understanding of the complexities of each subsystem, as well as their current states, and our estimations of the amount of work still needed for their completion. These hourly estimates are subject to change as the project moves forward.

2.7 Other Resource Requirements

This project is in direct collaboration with M:2:I. The M:2:I team has/will assist us with access to appropriate hardware and lab time. They will also give us feedback on what needs to be done, and how it affects the overall satellite.

3 Design

3.1 Previous Work And Literature

Include relevant background/literature review for the project

– If similar products exist in the market, describe what has already been done

– If you are following previous work, cite that and discuss the **advantages/shortcomings**

– Note that while you are not expected to "compete" with other existing products / research groups, you should be able to differentiate your project from what is available

Detail any similar products or research done on this topic previously. Please cite your sources and include them in your references. All figures must be captioned and referenced in your text.

3.2 Design Thinking

Detail any design thinking driven design "define" aspects that shape your design. Enumerate some of the other design choices that came up in your design thinking "ideate" phase.

3.3 Proposed Design

Include any/all possible methods of approach to solving the problem:

- Discuss what you have done so far – what have you tried/implemented/tested?
- Some discussion of how this design satisfies the **functional and non-functional requirements** of the project.
- If any **standards** are relevant to your project (e.g. IEEE standards, NIST standards) discuss the applicability of those standards here
- This design description should be in **sufficient detail** that another team of engineers can look through it and implement it.

3.4 Technology Considerations

Highlight the strengths, weakness, and trade-offs made in technology available.

Discuss possible solutions and design alternatives

3.5 Design Analysis

– Did your proposed design from 3.3 work? Why or why not?

– What are your observations, thoughts, and ideas to modify or iterate over the design?

3.6 Development Process

Discuss what development process you are following with a rationale for it – Waterfall, TDD, Agile. Note that this is not necessarily only for software projects. Development processes are applicable for all design projects.

3.7 Design Plan

Describe a design plan with respect to use-cases within the context of requirements, modules in your design (dependency/concurrency of modules through a module diagram, interfaces, architectural overview), module constraints tied to requirements.

4 Testing

Testing is an **extremely** important component of most projects, whether it involves a circuit, a process, or software.

1. Define the needed types of tests (unit testing for modules, integrity testing for interfaces, user-study or acceptance testing for functional and non-functional requirements).
2. Define/identify the individual items/units and interfaces 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

Include Functional and Non-Functional Testing, Modeling and Simulations, challenges you have determined.

4.1 Unit Testing

– Discuss any hardware/software units being tested in isolation

4.2 Interface Testing

– Discuss how the composition of two or more units (interfaces) are to be tested. Enumerate all the relevant interfaces in your design.

4.3 Acceptance Testing

How will you demonstrate that the design requirements, both functional and non-functional are being met? How would you involve your client in the acceptance testing?

4.4 Results

– List and explain any and all results obtained so far during the testing phase

- Include failures and successes
- Explain what you learned and how you are planning to change the design iteratively as you progress with your project
- If you are including figures, please include captions and cite it in the text

5 Implementation

Describe any (preliminary) implementation plan for the next semester for your proposed design in 3.3.

6 Closing Material

6.1 Conclusion

Summarize the work you have done so far. Briefly re-iterate your goals. Then, re-iterate the best plan of action (or solution) to achieving your goals and indicate why this surpasses all other possible solutions tested.

6.2 References

List technical references and related work / market survey references. Do professional citation style (ex. IEEE).

6.3 Appendices

Any additional information that would be helpful to the evaluation of your design document.

If you have any large graphs, tables, or similar data that does not directly pertain to the problem but helps support it, include it here. This would also be a good area to include hardware/software manuals used. May include CAD files, circuit schematics, layout etc,. PCB testing issues etc., Software bugs etc.