

# MicroCART

## DESIGN DOCUMENT

Team Number: 50

Client: Dr. Phillip Jones

Advisers: Matt <Lastname> James <Last name>

Team Members:

- Evan Blough -- Technical Team Lead, Embedded Software Lead
- Kynara Fernandes -- Ground Control Station Lead
- Aaron Szeto -- Controls Lead
- Joe Gamble -- Embedded Hardware Lead
- Shubham Sharma -- Crazy Fly Implementation Lead, Website Manager
- Jacob Brown -- Physical Hardware Lead

Team Email: [sdmay20-50@iastate.edu](mailto:sdmay20-50@iastate.edu)

Team Website: <http://sdmay20-50.sd.ece.iastate.edu/>

Revised: 09/04/20

# Executive Summary

## Development Standards & Practices Used

Code documentation

Follow the Principles of Programming

IEEE Code of Ethics

VHDL Coding Standards

IEEE Floating Point Standard

Wireless Networking – "WiFi"

Coding Standards for High-Confidence Embedded Systems

## Summary of Requirements

List all requirements as bullet points in brief.

- 1) Integration of a **secondary** quadcopter into the high-speed camera system,
- 2) Extend modular design to enable testing advanced controllers and signal processing,
- 3) Design and develop capabilities to support the implementation of the real world application,
- 4) Design and implement a mock real-world demo for visitors.

## Applicable Courses from Iowa State University Curriculum

List of Iowa State University courses which were applicable towards the MircoCART project:

- CPRE 288:
  - This course:
    - Goes over elementary embedded design flow/methodology
    - It gives students a basic understanding into micro-controllers.
    - Applications laboratory exercises with embedded devices.
  - We work with microcontrollers a lot with the project as evident from the project title. Hence, this course is a great foundation to assist us in the project.
- EE 230:
  - This course:
    - Is an overview of circuitry design and analysis
    - Gives students experience working with laboratory instrumentation and measurements
  - This is an applicable course as the project consists of circuit design knowledge from 230.
- CPRE 381:
  - This course:
    - Is an Introduction to computer organization, evaluating the performance of computer systems
    - Datapath and control, scalar pipelines, introduction to memory and I/O systems
  - The project consists of ARM-processors we would need to work with. And since this course is centered heavily around processors it is helpful towards the project.
- EE 330:
  - This course:
    - Goes over semiconductor technology for integrated circuits.
    - Analysis and design of analog building blocks.
    - Laboratory exercises and design projects with CAD tools and standard cells.
  - The project would require an understanding of circuit issues such as unmatched impedance and circuit devices such as current mirrors. Which is why this course is applicable for the project.

- EE 224:
  - This course:
    - Introduction to using Matlab for EE work
    - Analysis of Signals and various signal systems like LTI
  - This class teaches how to use Matlab and quadcopter controls are entirely based in Matlab hence it is helpful for the project.
- EE 321:
  - This course:
    - Continuation of EE 224
    - Focuses more on modulation and data transmission
  - Focuses on data transmission which applies to how the quadcopter communicates with the controls and ground station, which is heavily used in the project.
- CPRE 488:
  - This course is about:
    - Embedded microprocessors and embedded memory
    - Component interfaces communicating to embedded software
    - Platform-based FPGA technology w/ hardware synthesis
    - And Real-time operating system concepts
  - This project requires knowledge with VHDL and C development on Xilinx platforms, multidisciplinary projects, embedded system design, and control systems on drones. Hence, why the course is applicable to the project.

## New Skills/Knowledge acquired that was not taught in courses

Qt software libraries

RC communication

PID Controls

# Table of Contents

1 Introduction	4
1.1 Acknowledgement	4
1.2 Problem and Project Statement	4
1.3 Operational Environment	4
1.4 Requirements	4
1.5 Intended Users and Uses	4
1.6 Assumptions and Limitations	5
1.7 Expected End Product and Deliverables	5
2. Specifications and Analysis	5
2.1 Proposed Design	5
2.2 Design Analysis	6
2.3 Development Process	6
2.4 Design Plan	6
3. Statement of Work	6
3.1 Previous Work And Literature	6
3.2 Technology Considerations	7
3.3 Task Decomposition	7
3.4 Possible Risks And Risk Management	7
3.5 Project Proposed Milestones and Evaluation Criteria	7
3.6 Project Tracking Procedures	7
3.7 Expected Results and Validation	7
4. Project Timeline, Estimated Resources, and Challenges	8
4.1 Project Timeline	8
4.2 Feasibility Assessment	8
4.3 Personnel Effort Requirements	8
4.4 Other Resource Requirements	8
4.5 Financial Requirements	9
5. Testing and Implementation	9
5.1 Interface Specifications	9
5.2 Hardware and software	9

5.3	Functional Testing	9
5.4	Non-Functional Testing	9
5.5	Process	10
5.6	Results	10
6.	Closing Material	10
6.1	Conclusion	10
6.2	References	10
6.3	Appendices	10

List of figures/tables/symbols/definitions (This should be similar to the project plan)

Term	Definition
CLI	Command-line interface
Continuous Integration	An automated process of running tests on every commit to the repository

Demo	Short for demonstration; this is one of the deliverables of the project: a demonstration of the quad's capabilities, for example, doing a backflip with the quad, finding an object and following it, communicating with a second quad to perform flight patterns
FPGA	Field Programmable Gate Array; this is a board that can be programmed to simulate electrical hardware components. Used often in reconfigurable computing
GCS	Ground Control Station, the application that runs on a host computer that communicates with the quad via a Wi-Fi connection and sends its coordinates to the quad
GUI	Graphical user interface
IR	Infrared wavelengths of light longer than visible light; used in the VRPN system to determine the position of the quad
LIDAR	Light Detection and Ranging; this is a system for determining the altitude (z) of the quad using the onboard sensor
Optical Flow	A system using pattern of motion of objects, surfaces, and edges caused by the relative motion between the and the scene to determine position; used by the quad to calculate the position (x, y) when not in the lab using the VRPN system
PID	Proportional-integral-derivative control system; standard control algorithm used on the quad
Quad	Short for quadcopter; this is the hardware platform we use in this project
Setpoint	In a control system, the target value for an essential variable
VRPN	Virtual-Reality Peripheral Network; this is the system used to determine the position (x, y, z) and orientation ( $\phi, \theta, \psi$ ) of the quad in the lab using a set of 12 stationary cameras and an IR transmitter on the quad

# 1 Introduction

## 1.1 ACKNOWLEDGEMENT

The development of this project has been aided by ECPE Faculty and graduate students at Iowa State. They offer valuable technical advice and insight. We would like to acknowledge their contributions to this project below.

- Matthew Cauwels
- Dr. Phillip Jones
- James Talbert
- May 2020 MicroCart Team

## 1.2 PROBLEM AND PROJECT STATEMENT

### 1.2.1 Problem Statement

MicroCART is a drone platform that will be used by graduate students to perform research on embedded systems and controls topics. The platform will also need to demonstrate its operation to professionals or future students to showcase the utility of skills obtained through the course of a EE/CPRE/SE degree. This platform will also need to be maintainable and modifiable by future senior design teams. Currently, the MicroCART platform is functional, and it will be extended to a copy of the current platform to implement swarm flight. In order to achieve these objectives, we will need to copy, refine, and modify the current platform to have more intuitive interface and more accurate flight performance.



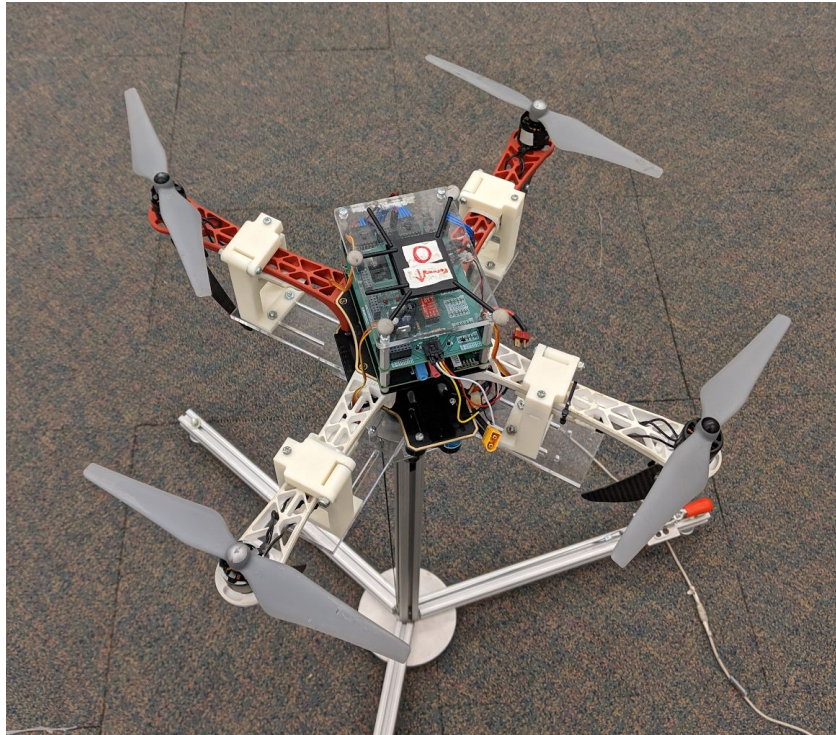


Figure 1.2.1: Current drone implementation

### 1.2.2 Solution Approach

We plan to further our knowledge of the previous development done on the project. We will need to read current documentation of the drone's subsystems and infer the functionality of portions that aren't documented. We also elected focus areas for each team member to specialize in areas of drone development. Once each of us are familiar with our respective subsystems, we can start building the second platform.

There is a smaller drone platform that we can use to familiarize ourselves with navigation. We hope to implement swarm flight with these smaller drones, and once we have experience with the smaller drones we will be able to move on to navigating the larger drone. By the end of the first semester, we hope to have both large drones operating with swarm flight.

The second semester would then consist of us adding new features to the drone to fulfill the purposes outlined above. The controls and embedded software leads will work on making more intuitive source for swapping controls algorithms on the drone. The GUI leads will be working on creating an easy to use interface for the drone. The embedded hardware will generate fast feedback loops to make the control systems more efficient and regulate memory use for each control algorithm. The hardware lead will be... As a team we might work to add Linux to the second core to promote development on our platform. This second Linux core could have high level C++ libraries like OpenCV that could be used for soft real time object tracking.

### 1.3 OPERATIONAL ENVIRONMENT

The operational environment for the quad is the inside of Coover 3050, and this room has a VRPN camera system that can be used for drone navigation. For indoors operation there will not be any hazardous environmental factors like extreme heat or cold. The environment in the lab will have people in it, and the drone could collide with expensive equipment in the lab, so failsafes for flight faults are important considerations for our operating environment. Due to the fact that the operation environment is identical to its use environment, we expect little design difficulty caused by the operational environment

### 1.4 REQUIREMENTS

The functional requirements for this project are building our quadcopter, get it flying at the same time as another quadcopter, and modifying the controls system to be compatible with other control systems. Some economic/market requirements include last years quadcopter, quadcopter replacement parts, and spare Lithium Ion batteries. The environmental requirement is that the quadcopter should be able to fly in an outdoor environment. UI requirements include making the control systems more user friendly.

### 1.5 INTENDED USERS AND USES

Platform will be used to test different control systems and their varying efficiencies for students. Student will require quick, simple interfaces to maximize learning and reduce time use for lab. The system will also be utilized in future senior design projects, requiring our team to use good documentation and written communication techniques. Finally, the project will be utilized in college demonstrations as a means of showing off to our employers what our college can do as well as an attempt to recruit future students.

### 1.6 ASSUMPTIONS AND LIMITATIONS

#### 1.6.1 Assumptions

- We won't have more than two large quadcopters in our operating environment
  - Further maybe hazardous/ too complicated
- The operating environment indoors doesn't contribute any significant hazards
  - Humans acting in a hazardous manner might complicate this
- We are assuming the shield board added to the drone works properly
  - Not tested by the previous team

#### 1.6.2 Limitations

- Drone flying environment limited to the area of the camera sensors within Coover 3050
  - The flight is dependent upon cameras tracking drone
- The computational power of the drone

- Limited in processing time
- Limited in memory access speed
- Feedback delays in control systems by the camera systems
  - Delay in ms expected
  - From Wi-fi communication
- Battery to payload ratio imposes limitations on overall runtime
  - Battery technology and weight characteristics of drone limit flight to appr. 5 minutes
  - Frequent replacement of battery systems necessary

## 1.7 EXPECTED END PRODUCT AND DELIVERABLES

Our product is academic in nature so we won't have commercialized descriptions. We will need to provide detailed documentation through gitlab documentation and documentation generators like doxygen.

### 1.7.1 Implementation of Second Drone

We are going to make an exact copy of the current drone platform from the previous year. Most of the work done on this portion will be in the form of research on the existing system. In order to make another platform we will need to research the existing documentation. We are making this deliverable, because we need to have one stable platform to demo our solution to people and one for a senior design team to do development on. This deliverable should be completed by October 25th.

### 1.7.2 Swarm Flight of Crazy Flies

After we have a second drone, we will hopefully have access to several crazy flies. We can use these platforms to test out extending our GCS software to multiple platforms. This is an intermediate deliverable. We will use this deliverable to get experience with programming swarm flight with inexpensive platforms and move on to our permanent solution for swarm flight. We should have this portion completed by November 1st.

### 1.7.3 Swarm Flight of large drones

After we have experience with swarm flight on the crazy flies and have the second drone built, we will work on having two drones operate simultaneously. This deliverable relates to our client need of demoing to students and faculty. Showing multiple vehicles operating simultaneously is more impressive than just a single vehicle. This showcases the usefulness of skills obtained with an ECPE degree. Controls students also will want to test control algorithms with multiple platforms interacting. This should be done by December 2nd.

### 1.7.4 Addition of Linux to Second Core

After we have both drones fully implemented and functioning properly, we will need to add new features to our board to showcase skills that we have learned from our degree. One of the features we plan on adding is Linux to the second core. The Zybo board has two ARM cores. We will install Linux on the second core and this will give researchers and future project teams greater flexibility to control our platform. We can also use it to view the value of a degree. We can install

an image detection library like OpenCV and a camera on Linux and use image detection to give control commands to the drone. This will be done at some point during the second semester of this course.

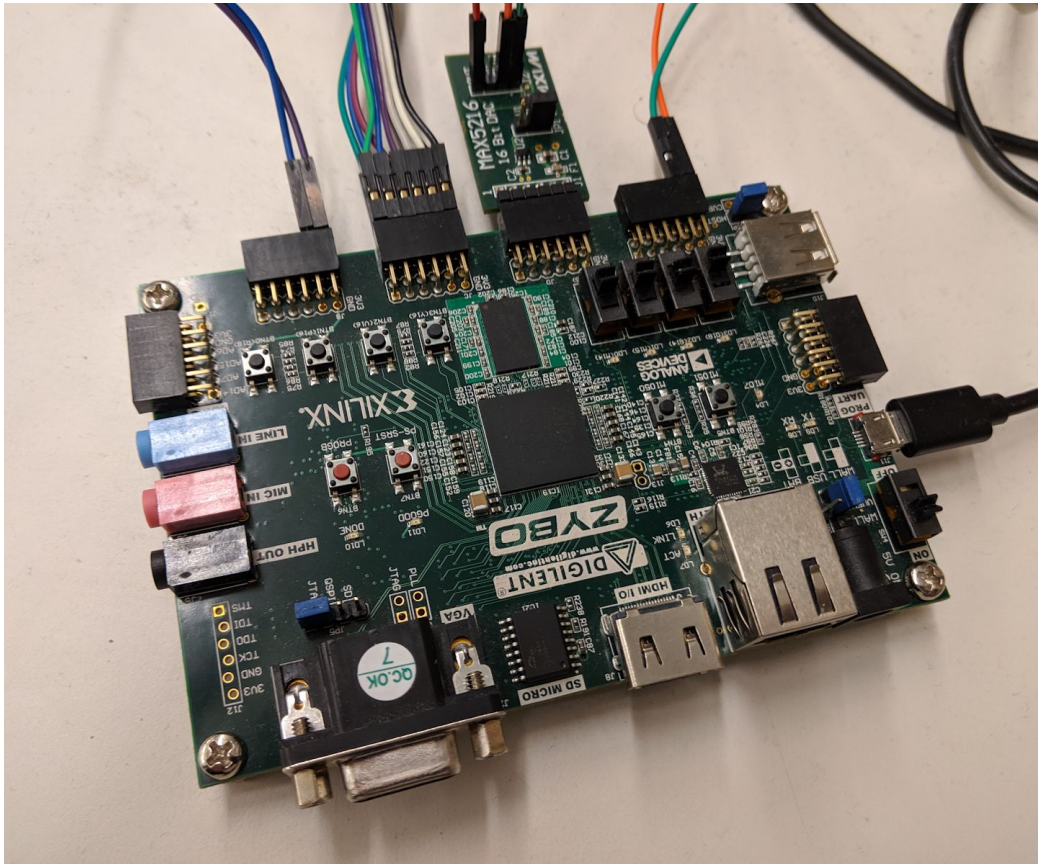


Figure 1.7.4: Zybo platform for drone development

### 1.7.5 Modular Control Algorithm Swapping on the Drone.

Another feature we plan on implementing is the modularity on control algorithms. Currently, the drone control program runs a PID based position correction algorithm. It measures the differences between the set points and the actual position of the drone that is captured.

### 1.7.6 More Interactive and Comprehensive UI

When talking with the current users of our project, we found that it would be helpful to create a more interactive element for the data from the quadcopter, camera system and ground station. We will do this by adding new CTEs, sliders and graphs to help visualize the input, output and calculated data.

## 2. Specifications and Analysis

### 2.1 PROPOSED DESIGN

So far we've started looking at the documentation and getting familiar with previous years work. We've also talked with our client about what's expected of us and to help us plan out what we want to do with the quadcopter. We're planning to get the soldering done this week and start working on the crazyflies as a prototype for our work on the quadcopter. To approach solving these problems we are using the Agile method to coordinate our coding work and the discord app to coordinate in-person meetings to work on the hardware aspects of the quadcopter. When working on our project we will follow IEEE safety standards.

### 2.2 DESIGN ANALYSIS

During the beginning design portion of the project we have worked to understand the current implemented system. Each component of the system has been built over many years and passed through many teams. It takes time to understand each portion of the system to an extent at which it can be utilized; therefore, the majority of the time has been spent reviewing documents.

Our ability to process the current documents has progressed well. We have begun to run the drone independently and some initial work. Each person on the team has demoed the drone and has read documentation. We have begun to complete initial work including soldering, control system analysis, gui interfaces, and initial design document creation.

#### **Strengths**

- Team comradery and communication
- Good mix of disciplines w/ large knowledge base
- Contacts with previous MicroCART users to answer questions
- Large group to distribute work

#### **Weakness**

- Large number of systems with intricacies
- No control theory knowledge base
- Poor documentation
- Time to complete is only a year

### 2.3 DEVELOPMENT PROCESS

After being assigned our project, we met with our customer and advisor to discuss the requirements for our project. Since we're working on an embedded systems project, our system has multiple components that would have different requirements that would change as we move through the school year.

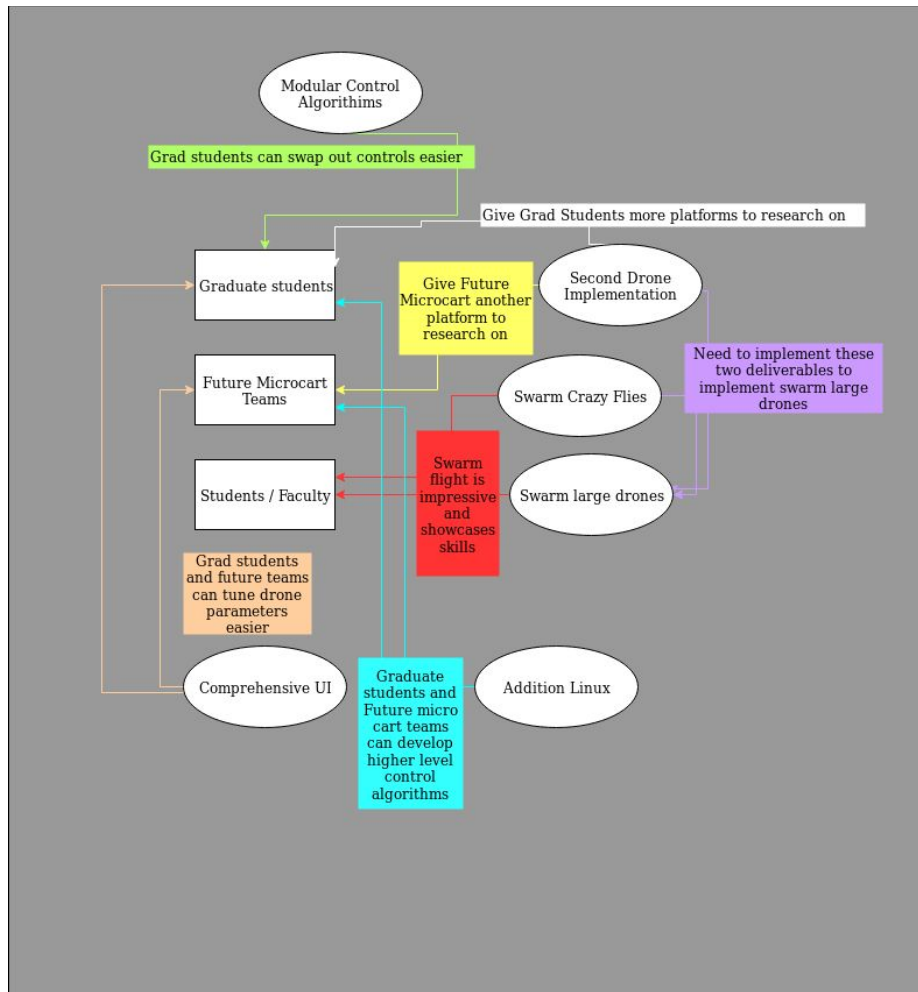
We are currently in the design phase of our project. We are researching materials like control theory etc. to help us understand our customer/users better and in turn, produce a better tailored product.

With these two phases, we are currently in the waterfall process. But since our requirements will change during the phase of the year, we are transitioning into an agile development process. The plan is as such:

- **Tickets:** We are currently using Trello to handle our tasks and keep track of the tasks that need to be done, the tasks that are in progress and our accomplished tasks.
- **Meeting with stakeholders:** We meet with our customers, users, and advisor once a week on Mondays to discuss our accomplished tasks and the concerns or goals they have for the coming sprint or sprints.
- **Sprints:** We have 1-week sprints in which we will go through one iteration of standup, grooming, retrospectives and stakeholder meetings.
- **Grooming:** After our stakeholder meeting, we regroup, and create and prioritize tickets on Trello for the new sprint.
- **Stand-up and Retrospective:** We start our weekly team meetings with scrum, where we talk about what we did, what we plan on doing and if we had any blockers for that sprint. Since we are full-time students, we compromised on doing this weekly instead of daily.

## 2.4 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.



## 3. Statement of Work

### 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 TECHNOLOGY CONSIDERATIONS

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

Discuss possible solutions and design alternatives

### 3.3 TASK DECOMPOSITION

In order to solve the problem at hand, it helps to decompose it into multiple tasks and to understand interdependence among tasks.

### 3.4 POSSIBLE RISKS AND RISK MANAGEMENT

Include any concerns or details that may slow or hinder your plan as it is now. These may include anything to do with costs, materials, equipment, knowledge of area, accuracy issues, etc.

### 3.5 PROJECT PROPOSED MILESTONES AND EVALUATION CRITERIA

What are some key milestones in your proposed project? Consider developing task-wise milestones. What tests will your group perform to confirm it works?

### 3.6 PROJECT TRACKING PROCEDURES

What will your group use to track progress throughout the course of this and next semester?

### 3.7 EXPECTED RESULTS AND VALIDATION

What is the desired outcome?

How will you confirm that your solutions work at a **High level**?

## 4. Project Timeline, Estimated Resources, and Challenges

### 4.1 PROJECT TIMELINE

- A realistic, well-planned schedule is an essential component of every well-planned project
- Most scheduling errors occur as the result of either not properly identifying all of the necessary activities (tasks and/or subtasks) or not properly estimating the amount of effort required to correctly complete the activity
- A detailed schedule is needed as a part of the plan:
  - Start with a Gantt chart showing the tasks (that you developed in 3.3) and associated subtasks versus the proposed project calendar. The Gantt chart shall be referenced and summarized in the text.
  - Annotate the Gantt chart with when each project deliverable will be delivered
- Completely compatible with an Agile development cycle if that's your thing

How would you plan for the project to be completed in two semesters? Represent with appropriate charts and tables or other means.

Make sure to include at least a couple paragraphs discussing the timeline and why it is being proposed. Include details that distinguish between design details for present project version and later stages of project.

### 4.2 FEASIBILITY ASSESSMENT

Realistic projection of what the project will be. State foreseen challenges of the project.



### 4.3 PERSONNEL EFFORT REQUIREMENTS

Include a detailed estimate in the form of a table accompanied by a textual reference and explanation. This estimate shall be done on a task-by-task basis and should be based on the projected effort required to perform the task correctly and not just “X” hours per week for the number of weeks that the task is active

### 4.4 OTHER RESOURCE REQUIREMENTS

Identify the other resources aside from financial, such as parts and materials that are required to conduct the project.

### 4.5 FINANCIAL REQUIREMENTS

If relevant, include the total financial resources required to conduct the project.

## 5. Testing and Implementation

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

Although the tooling is usually significantly different, the testing process is typically quite similar regardless of CprE, EE, or SE themed project:

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

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

### 5.1 INTERFACE SPECIFICATIONS

– Discuss any hardware/software interfacing that you are working on for testing your project

## 5.2 HARDWARE AND SOFTWARE

- Indicate any hardware and/or software used in the testing phase
- Provide brief, simple introductions for each to explain the usefulness of each

## 5.3 FUNCTIONAL TESTING

Examples include unit, integration, system, acceptance testing

## 5.4 NON-FUNCTIONAL TESTING

Testing for performance, security, usability, compatibility

## 5.5 PROCESS

- Explain how each method indicated in Section 2 was tested
- Flow diagram of the process if applicable (should be for most projects)

## 5.6 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 it as you progress with your project
    - - If you are including figures, please include captions and cite it in the text
  - This part will likely need to be refined in your 492 semester where the majority of the implementation and testing work will take place
- Modeling and Simulation:** This could be logic analyzation, waveform outputs, block testing. 3D model renders, modeling graphs.
- List the **implementation Issues and Challenges.**

## 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

This will likely be different than in project plan, since these will be technical references versus 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 that does not directly pertain to the problem but helps support it, include that 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.