MicroCART

DESIGN DOCUMENT

Team Number: 27

Client: Dr. Phillip Jones
Advisor: Dr. Phillip Jones

Team Members:
Alex Bjerke –- Project Manager
Amith Kopparapu Venkata Boja –- Embedded Software Lead
Theodore Davis –- Embedded Hardware Lead
Grayson Goss –- Technical Lead | CAD Design Lead
Hannah Mohamad -– Team Webmaster
Russ Paulsen -– Ground Control Lead
Alfonso Raymundo -– PCB Design Lead
Trent Woodhouse –- High-Level Software Lead

Team Email: sdmay21-27@iastate.edu
Team Website: https://sdmay21-27.sd.ece.iastate.edu

Revised: 10/25/20 - Version 2

Executive Summary

Development Standards & Practices Used

Communication standards being used in this project include: WiFi, RF, PWM, and I2C. Industry standards being used in the project include: IPC, IEEE, UASSC, and ITU Radio Regulations. In order to maintain successful development, the team follows the Agile development style.

Summary of Requirements

The requirements for this project were given to us by Dr. Phillip Jones who wants us to develop a Mini 4.5 x 4.5 in, educational friendly drone. The drone should be simple to code with and can be controlled by radio frequency and wifi. The drone will have a ground control station where the students can perform testing on it, program it, and recharge its battery. The total cost per drone needs to be less than $50. Here is a list of the parts we plan to use

- MicroController (Adafruit Feather M4 Express)
- Accelerometer + Gyro (Adafruit LSM6DSOX + LIS3MDL FeatherWing)
- WiFi + Bluetooth Microcontroller (NodeMCU-32S ESP32)
- Propellers
- Motors
- Coreless DC motors
- Expansion connector pins
- Battery/battery holder
- Easy for students to program
- Ground control station
- Cheap (less than $50)

Applicable Courses from Iowa State University Curriculum

CprE 288 Embedded Systems I:

- Overview of embedded systems and embedded programming.
- Interrupts, I/O, Timers, peripherals, resource allocation and optimization.
- Applications of embedded devices.

- Served as a foundation course for us to understand the nature of the MicroCART project.

ComS 309 Software Development Practices:

- Introduction to managing software development, process models, requirements analysis, object-oriented design, coding, testing, and maintenance.

- Gave us first hand experience with a project development cycle and project management.

ComS 319 Construction of User Interfaces:

- Overview of user interface design.
- Evaluation and testing of user interfaces.
- Review of principles of object orientation, object oriented design adn analysis using UML.
- Design of windows, menus, and commands
- Developing web and windows-based user interfaces.

EE 333 Electronic System Design:

- Introduction to Arduino tutorials, Sensors, Switched-mode power supplies (SMPS), KiCad and Printed circuit boards.

- Gave us first hand experience with a project development cycle and project management. From building the prototype, making the Schematic, Having the PCB made and ordering the parts and, building the Final PCB how to solder the components on to the PCB.

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.

Will have this section completed in the next version, when we will be able to give a complete, comprehensive list.

Table of Contents

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

1 Introduction

1.1 Acknowledgement

We would like to express our greatest gratitude to Dr. Phillip Jones who gave us the opportunity to be a part of the team that can contribute to the Electrical and Computer Engineering Department. He has been a wonderful advisor for the continuous support and his guidance.

This project could not be possible without the team members of the project: Alex Bjerke, Theodore Davis, Alfonso Raymundo, Amith Kopparapu Venkata Boja, Russ Paulsen, Trent Woodhouse, Grayson Goss, Hannah Aisya Mohamad. We will continue to give our best to execute the project successfully.

1.2 Problem and Project Statement

The Electrical and Computer Engineering department wants to create a mini quadcopter drone that is suitable for lab experiments in the CprE 488. The drone must be at most 4.5 by 4.5 inches in size and must be easy for students to use their own C code to control the drone. In addition to designing this drone, there must be a ground control program for students to connect to, and communicate with the drone. There must also be a test station where the drone's movement can be observed about a particular axis.

In order to solve this problem, the team of eight students will break down into sub groups and work on specific components that make sense for their skill sets. There are four main sub groups to consider: test station design, ground control program, drone software, drone hardware. We will use open-source and previous drone projects to guide our understanding and execution of the project.

1.3 Operational Environment

This drone will be used exclusively indoors and deal with few to no environmental obstacles. The main operational area will be in the lab the class takes place in. This means the operational environment is very controlled and unchanging. The biggest obstacle for this operational area will be flight time of the drone.

1.4 Requirements

The requirements for this project were given to us by Dr. Phillip Jones who wants us to develop a Mini 4.5 x 4.5 inch programmable drone. The drone should be easy to code and can be controlled by radio frequency and or wifi. The drone will have a ground control program where the students can send commands to the drone and receive data back from the drone over WiFi. The total cost per drone needs to be less than $50. Additionally, the drone should have about 10 minutes of flight time.

In addition to the drone, there should be a test station that the drone can attach to. This test station should be able to gather rotational data upon at least one axis. This data should be sent to the ground control application for students to analyze their code's performance on the drone.

1.5 Intended Users and Uses

MicroCART is designed specifically for students of CprE 488. It's intended use is to provide students an opportunity to work with embedded systems, data collection, and testing. Using Arduino IDE, students can directly program functionality into the drone. Then, they can control the drone with a radio controller, or alternatively, use the provided desktop application to control the drone via wifi. Students can then collect the data from the drone and the testing station and view it within the provided desktop application, and can make adjustments to the code accordingly.

1.6 Assumptions and Limitations

Assumptions

- Users of the drone will be knowledgeable in C
- All applications and uses for the drone are conducted indoors
- Users are given a primer on aspects of the drone before use
- Adjustments for additional payloads will be on the user to code
- This drone will likely be broken due to erroneous code or improper flight
- System will be rechargeable (Li-Po or Li-Ion batteries)

Limitations

- Cost per drone must not exceed $50 USD (client requirement)
- Payloads must not exceed 1/5th of the drone's total weight (flight limitation)
- Drone's size shall not exceed 4.5"x4.5" (client requirement)
- Drone must be able to communicate on several protocols (client requirement)
- Drone will not be subjected to harsh winds (> 5 mi/hr)

1.7 Expected End Product and Deliverables

The overall goal of this project is to build a drone design that will help students in CprE 488 class use it for learning purposes. Since this project is not commercialized, it does not need to be described from a commercial perspective.

1.7.1 Building a design

The first major section of our project is to build a working design of a drone. It should have the basic functionality of flying with the help of an RF controller. It should also use all the sensors attached to the drone to stabilize the drone or send data out in a readable form. For example, the accelerometer will be used to stabilize the drone, and a temperature sensor could be used to send the current condition's information to the user. Sensors like the magnetometer could be used by the students to read specific data and build algorithms to automate the flight of the drone. The student will be able to charge the battery and program the hardware of the drone using a USB port. This segment as a whole will ensure that students will focus on embedded programming aspects of the drone rather than the functioning of the design.

1.7.2 User-Friendly GUI

This is the next phase of a user- friendly experience. One of the best ways to truly understand a drone's functionality is to read its data. Having a user-friendly graphical user interface will help students constantly get data from the drone and the ground station for future analysis. There will also be a command-line interface option for commanding the drone.

1.7.3 Testing Station

The best alternative way of testing the drone's performance is to see its performance from a field's perspective. This testing station will record data of the drone in a field to determine the accuracy of its calculations. It will also help students read data from the user end to understand the drone better.

1.7.4 Additional Improvements

After the drone was successfully built with an interactive GUI application, the next phase is to build a simple and secure library for the microcontroller programming of the drone, so it is easier for students to use to call certain functions.

2 Project Plan

2.1 Task Decomposition

Initial tasks:

The first major project-based tasks that were necessary were establishing requirements and doing preliminary research. The preliminary research was meant for the team to gain a sufficient understanding about the operations of quadcopters and what goes into designing a mini quadcopter.

Drone:

- PCB/Hardware
    - The PCB/Hardware component of the drone is perhaps the biggest subgroup. The tasks here include determining parts and sensors to use, designing the PCB, ensuring size requirements are met, and prototyping.
- Embedded Software
    - The embedded software component of the drone reflects any software "living" on the quadcopter. This subgroup will need to research libraries supported by the chosen microcontroller, research other open source software used in

programmable drones, and be responsible for the communication between the drone and external sources (ground station and RF controller).

- CAD
    - Drone chassis and wiring harness need to be constructed for this custom drone.

Test Station:

- Sensors
    - There will be a time when the test station's sensors and microcontroller need to be selected.
- CAD
    - There are previous teams' CAD files that will need to be reviewed. Also, once the team determines the sensors and microcontroller to use, someone will need to prototype/design the final product.
- Embedded Software
    - There will need to be software embedded in the test station responsible for sending data to the ground control.

Ground Control:

As there are no initial, hard-set requirements given to us for the ground control application, the first major task is determining what is needed, possible, and feasible. After making these decisions, the next step is to begin making a skeleton application. During this process, documentation should be made.

Website:

Tasks for the team website include learning how it works and making updates when necessary.

2.2 Risks And Risk Management/Mitigation

- Drone
    - Risk Factors:
        - Connections may not be soldered correctly. 0.5
        - Erratic flight. 0.6
        - Unable to send data to the ground station. 0.2
        - Drone is too heavy for take off. 0.1
        - Battery may die while the drone is in flight. 0.1
    - Risk Mitigation:
        - Use practice soldering boards or go to a workshop.
        - Demo with open source working drone code.

- Ground Station
    - Risk Factors:
        - Does not connect to the drone properly. 0.2
        - Does not display sensor information correctly. 0.1
        - Ground station dies. 0.02

        - Ground station disconnects. 0.08

- Testing Station
    - Risk Factors:
        - Sensors do not record data. 0.05
        - Is unable to send data to the ground station. 0.2
        - Drone is unable to move with negligible friction. 0.3

2.3 Project Proposed Milestones, Metrics, and Evaluation Criteria

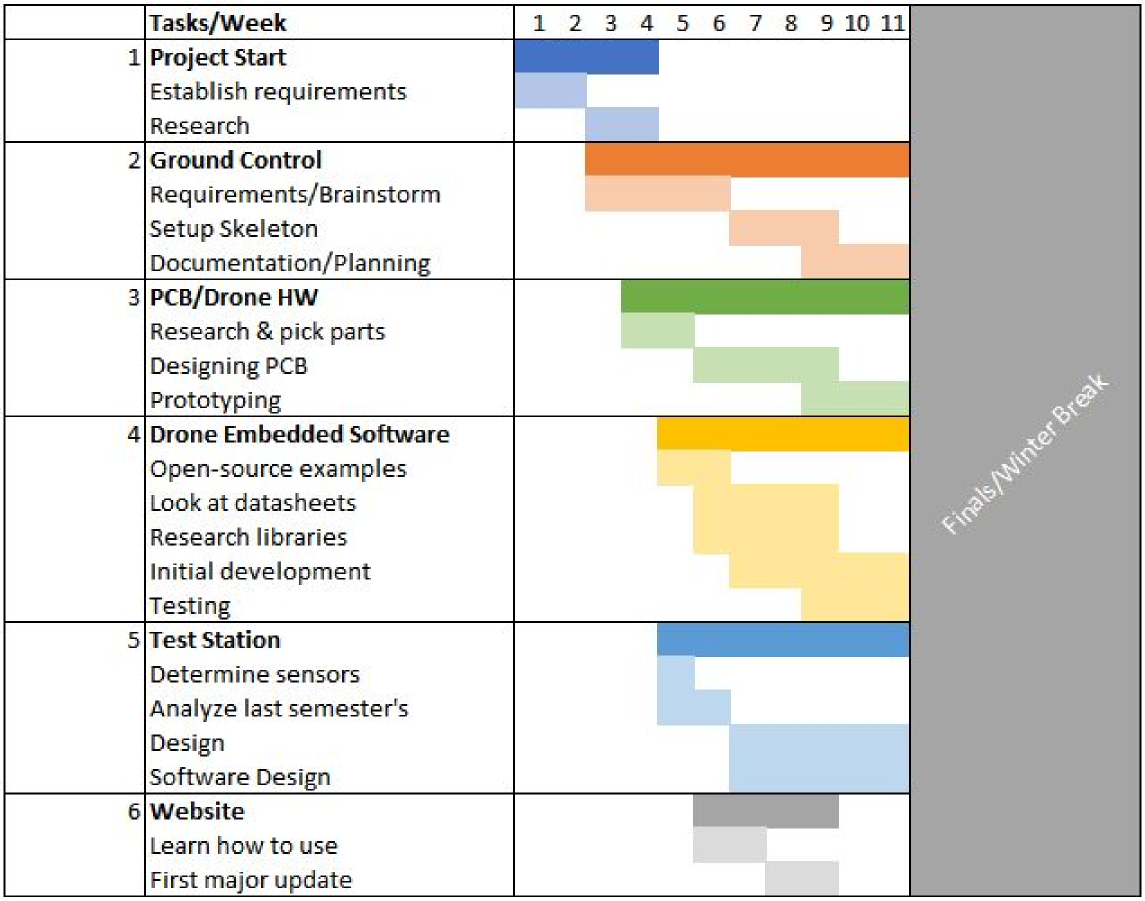
The biggest milestones of our project are :

- Gain sufficient understanding in the functionality of a drone
- Building the smallest PCB possible (i.e. A PCB that can fit on a 4.5" X 4.5" drone) while trying to keep the layer size of the PCB to 2 layers
- Make all the sensors on the drone to communicate with the ground control program and also send data with at least 90% accuracy.
- Build an effective design prototype of the drone that incorporates all the sensors and motors while meeting the requirements. Minimize the size and weight of the drone by using special designed parts through CAD.
- Build an effective testing station that reads all the necessary information from the drone. It also determines the location of the drone in the field with 80% accuracy.

We will evaluate the effectiveness of the contents on the drone and the testing station by comparing the expected results to the actual results of the sensors.

2.4 Project Timeline/Schedule

The figure below represents the first semester's timeline. This is using a start date of September 6, 2020. The first semester will be focused on the setup for a successful second semester. The plans for the second semester are very tentative at the moment. The second semester will begin in a spot where prototyping has begun, and there is a clear path to the end of the project. Integration of different parts will take place, and there will be lots of testing needed.



2.5 Project Tracking Procedures

Our team will be using GitLab's Issues and Issue Board to track progress. This is a way to create tasks, assign them, and track their status. Additionally, we hold weekly meetings to sync with the team, with an additional weekly report of progress for our client, and a bi-weekly report for the class.

2.6 Personnel Effort Requirements

Task	Time and Description
Prototyping	(2-4 weeks) This stage primarily consists of testing individual parts of the project then combining them together to form a complete prototype.
Embedded System Design (HW/SW)	(5-7 weeks) Embedded software and hardware will probably be one of the most intensive portions of the drone's construction since there are several modules that require

	programming for correct function.
Test Station Design and Construction	(1-2 Weeks) This stage is composed of the design behind the testing station for use in drone calibration. While we do already have existing files on hand, we have yet to determine the sensors to be used in this station.
Chassis Design Construction	(2-3 weeks) Due to this being a custom drone, a custom chassis will need to be modelled and simulated to determine weak points before production. Having this chassis be 3D printed will allow for a lower cost construction. This extended time accounts for potential failures in 3D printing
Ground Station Design	(4-6 weeks) Depending on the amount of open source code is available, the Ground Station (which will be used to communicate with the drone and the test station) will take between 4-6 weeks
PCB Design	(< 1 week) With a fully prototyped design established, the design of a PCB should be relatively easy considering this step consists of organizing parts and routing connections

2.7 Other Resource Requirements

We have finalized a few components that we will be using in order to build drones, however, there are a few devices that we have yet to select and discuss more. The list that we will be dealing with are as follows:

- Microcontroller (MCU)
- Sensors for Gyro, Accelerometer
- RF receiver
- Motors
- Batteries
- PCB manufacturer - Oshpark.com
- Propellers

2.8 Financial Requirements

The final drone should be built for less than 50 dollars, and the test station should also be less than 50 dollars. The financial part of this project is being handled by Dr. Phillip Jones. This is done by us finding pieces and parts and giving a list of what we need to Dr. Phillip Jones and he orders the pieces and parts for us if he decides those parts are good enough for what he envisions.

3 Design

3.1 Previous Work And Literature

For this project we have had ample sources to compare from and see what works commercially. Most of the commercially available drones on the market aren't readily reprogrammable One product on the market today that we are taking heavy inspiration from is the Crazyflie 2.1 (fig 3.1.1).

(Fig 3.1.1 Crazyflie 2.1)

The Crazyflie is a reprogrammable quadcopter of the size of which we are aiming for. The reason why our client hasn't chosen to purchase these is the price tag of %195.00. This design is built for indoor/outdoor high altitude flight. Our sole purpose for the MicroCART is a reprogrammable quadcopter capable of indoor low altitude flight.

3.2 Design Thinking

Coming into this class, students will have an understanding of working with and programming microcontrollers. In CprE 488, students are expected to expand on what they already know and apply their knowledge to drones. As a result, the focus of our product is to provide students with an opportunity to learn and grow in their studies, particularly within computer engineering. We plan to focus our design thinking around the students' wants and needs when working with the drone, all while meeting the classes' course objectives. This will be done by focusing on making our product simple to use.

When programming the drone and controlling it, the applications used are designed to be easily understood by students. All of the main functionality will still be programmed by the students, but the internal code that doesn't need to be managed by students will be handled and hidden from the students, giving students a simple programming experience.

3.3 Proposed Design

This section covers the design aspects of the drone, ground station, and test station.

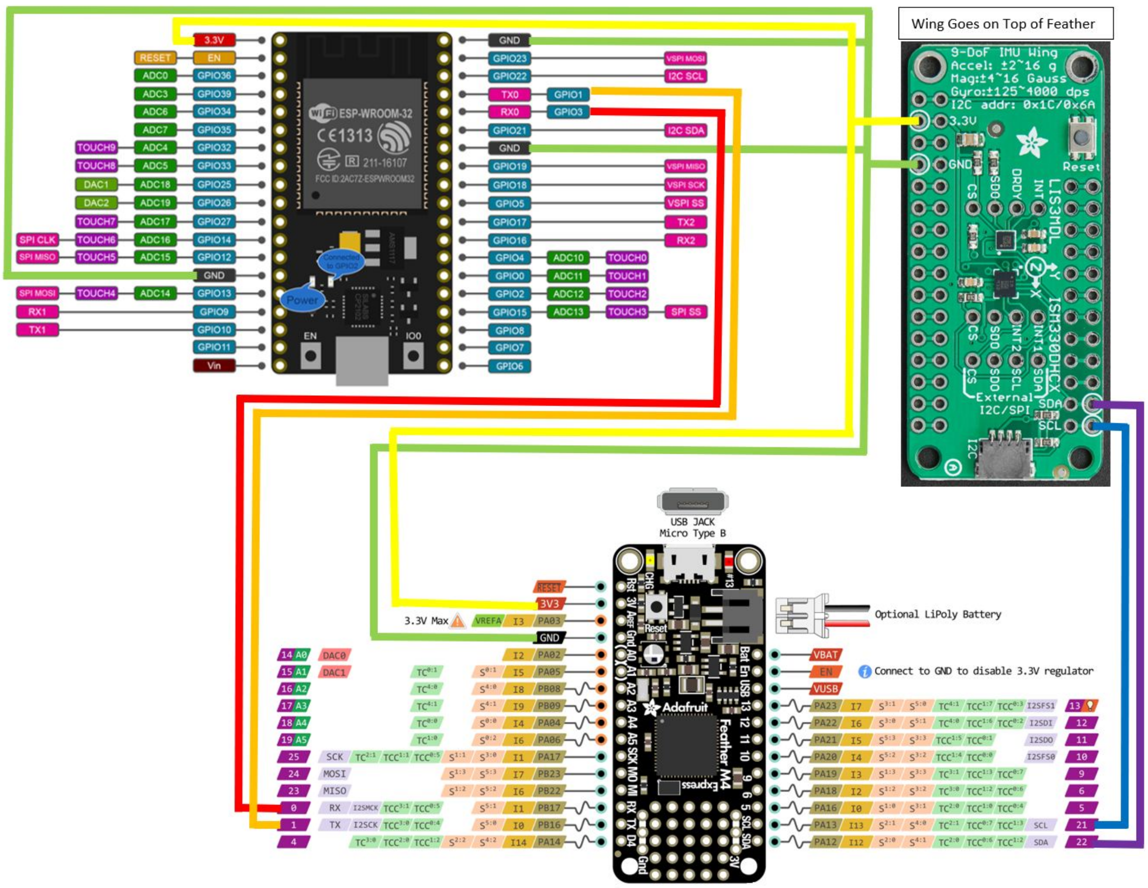
3.3.1 Drone

- Drone PCB Design:
    - The following components will be included in the pcb design:
        - Adafruit Feather M4 Express (ATSAMD51 Cortex M4)
        - Adafruit LSM6DSOX + LIS3MDL FeatherWing
        - NodeMCU-32S ESP32
    - The Drone will follow the following pin assignment and wiring diagram. As well as provide pinouts for pins along with ground: Pin 5, 6, 10, 12. To provide pwm signals or the electronic speed controller.

(fig 3.3.1.1 Pin Assignments for ATSAMD51 and ESP32)

Pin Number	Port	Periphals	Use
Pin 0	PB17	UART RX	Communicate with ESP32
Pin 1	PB16	UART TX	Communicate with ESP32
Pin 5	PA16	TC2 / PWM	Signal for ESC
Pin 6	PA18	TC3 / PWM	Signal for ESC
Pin 8	PB03	GPIO / LED	On board Neo Pixel - status?
Pin 10	PA20	TC7 / PWM	Signal for ESC
Pin 12	PA22	TC4 / PWM	Signal for ESC
Pin 13	PA23	GPIO / LED	On board LED
Pin 21	PA13	I2C SDA	Communicate with Sensors
Pin 22	PA12	I2C SCL	Communicate with Sensors
ESP 32			
GPIO1	TX0	UART0 TX	Communicate with MCU
GPIO3	RX0	UART0 RX	Communicate with MCU

(fig 3.3.1.2 Wiring Diagram)



- Drone Software:
  - ESP32:
    - The main program will wait for either condition to happen and loop:
      - Data is received through UART then is forwarded over WiFi.
      - Data is received through WiFi and is forwarded over UART.
  - ATSAMD51:
    - Receiving instructions from ground control:

      Reading the instructions sent by ground control regarding the sensor data needed to be read. These instructions are read from UART coming from the ESP32.

    - Reading Sensor Data:

      This segment of the code is responsible for reading raw data from the Berry IMU, which contains the gyroscope, accelerometer and magnetometer, through the I2C module. This segment also converts the raw data into a readable format.

        - Transferring of the data:

          The formatted data is sent in bytes through UART to the ESP32, which is forwarded over WiFi to ground control.

        - Reading and executing motion instructions:

          This segment is responsible for constantly reading and decoding the PWM signals from the RF controller. It is also responsible for using this using this information to run the propellers of the drone.

3.3.2 Ground Control

Below is the current description of the ground control:

- Main program written in C with the following behaviors:
    - Two threads responsible for communicating with the drone. One thread will be responsible for packing and sending data to the drone, then putting this data into a shared queue, called Q_TX_DRONE. The second thread will be responsible for receiving and unpacking data from the drone, then putting this data into a shared queue, called Q_RX_DRONE.
    - The main portion of the program will have two different modes, GUI mode and terminal mode
        - GUI Mode - Spawning a local JavaScript GUI that will then connect to the main C program via a local socket. There will be two more threads here for the JavaScript communication. One of these threads will be for receiving and unpacking data sent from the GUI. This data will then be put into a queue and treated as input to the main C program. The second thread will be responsible for taking data from Q_RX_DRONE, then packing and sending the response back to the GUI.
        - Terminal Mode - Taking input from stdin, from the terminal, then output to stdout. This will be a basic, command-line interface for sending commands to the drone, then seeing the drone's response (if any).
    - Test Station Communication - TBD
- JavaScript GUI
    - When this is spawned from the C program, a websocket connection is made between this and the C program and allows data to pass back and forth.
    - The Javascript application is split into three pages:
        - Home - Students can get a basic overview of the application and/or the class, and other helpful information.
        - Connections - Students can connect to drones and testing stations, and can control the drones from this page.
        - Data - All data provided by the drone and testing station are sent to this application via websockets, and is presented in the form of graphs. Recording data and exporting it is also an option.

3.3.3 Test Station

- Design of the test station is ongoing. We are drafting a current design that allows for analyzing different cardinal movements of the drone through the use of different drone orientations and fasteners for the drone
- The goal of the test station is to allow for analysis and calibration of the drone's function by providing data feedback to the users of the drone's performance at runtime.
- Current design consists of a baseplate (to secure the overall apparatus to a table, floor, etc.). a sensor rig along the axis (to measure the various positional and velocity related datas), and testing plate (to which the drone is attached in various orientations such that we can measure the lift from roll, pitch or yaw.

3.4 Technology Considerations

Strengths:

- Ability to refactor and change easily: New components can be added and existing components can be easily removed from the drone because of the control we have on every aspect of the drone.
- Simplicity in finding helpful resources: Every software and hardware component used in this project is commonly used for engineering projects, so if in the future, there were any unexpected issues, there are a lot of resources that can help solve the problem.

  The availability of resources will also give us an opportunity to get new ideas for improvising our product output.

- Simplified IMU: The major sensing system of the drone comes down to a simple board, helping us simplify the model of the drone and reduce weight and clumsiness of wires on the drone.
- User-friendly application: Ability to read important sensor data with the click of a button or a simple command, which would help people using the drone draw accurate and precise assumptions about the drone.

Weaknesses:

- Cost: Some components in the drone have components that are left unused, so money can be used for components that might not be necessary.
- Size simplicity of the parts: The parts used have potential to be reduced to meet just our requirements, but require special tools, and so there is more space on the drone needs.

Trade Offs:

- Even though the cost increases, it would give an opportunity to add additional components to the drone in the future. For example, a camera module can be added to the drone by connecting to the open microcontroller module.
- Even though size increases, we are only overestimating the drones requirements and not underestimating, which leads to malfunctioning drones. It might take more cost, energy and time to fix some of the new malfunctions.

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?

The project's current state does not yet reflect whether this design has worked or not.

3.6 Development Process

Our team is using the Agile development process. Given the nature of the project, we are easily able to split the project into multiple sub-parts, then work in an agile style from there. The group is split into four main sub-groups, listed below. While the people listed for each group are the main contributors, some people may have some less major contributions to different groups.

- Ground Control (Trent, Alex)
- Test Station (Russ, Grayson)
- Drone Software (Amith, Hannah)
- Drone Hardware/Design (Theodore, Alfonso, Grayson)

Since these subgroups are easily able to work in parallel, it is natural to follow the Agile process. Additionally, we are using Git Issues to track/assign tasks and enhance the Agile experience.

3.7 Design Plan

Due to the novel nature of this drone design, we will require an iterative design plan to deal with setbacks in any part of the project. We will start out with just a simple prototype. No chassis, just the electronic components wired together to see if they all work together as planned. Then, after this is shown, we will construct a full drone with chassis and wiring harness. This will be the basis of our work with both the ground station and the testing station. We will start with testing how the drone flies and how all the components (sensors, motors, computation units, communication units) work together. After initial errors are ironed out, we will start on the ground station, which is the core controller of the drone. The ground station will connect over several wireless protocols to the drone and allow for us to test remote flight of the drone as well as begin crafting a UI for the intended users. After the ground station is completed, we will work on the test station. This hardware integrates with the ground station as a means of communicating data from the test station to the ground station, where data can be read and used to tweak the drone to a proper configuration. After all the modules are constructed, a full validation test will be conducted on the entire system to ensure that there are no bugs or faults. After a clean bill of health, the project is deemed complete.

4 Testing

We are leaving the template's text in this position for the time being. The team has not reached a point of discussing in-depth test plan details.

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

Individual components that will need to be tested independently are:

- The ground control program's communication via WiFi
- Test station sensors
- The drone sensors (accelerometer, gyroscope, magnetometer)
- The drone's electronic speed controller

4.2 Interface Testing

– After a software/hardware module successfully passes the 1st round of unit tests, we assume that the modules work.

– Next, we perform the 2nd round of tests on the drone modules to verify that the Circuit board is working with the ESP 32, Feather M4 & Wing.

– After they were verified, we ran a program on the Feather M4 board to see if they were interfacing with one another. These tests confirmed functionality for the Feather M4 and modules within the prototype circuit board of the quadcopter.

– Here is a list of what hardware modules communicate with each other. The Feather M4 is of course the main module. The Feather M4 will communicate with ESP 32, Wing, Motors, radio-frequency module, ground station, and test station sensor data.

1) The Feather M4 will control/tell the ESP 32 to look for wifi and other info it can give to the Feather M4 for later use.

2) The Feather M4 will control/tell the sensor what to do and save the info to give to the ground station for later use.

3) The Feather M4 will control/tell the wings to share the data/info it gets from its Accelerometer+Gyroscope sensors with the Feather M4 for its own use.

4) The Feather M4 will control the speed that the motors turn using the supply voltage from the battery.

5) The Feather M4 will work together with the radio-frequency module so it will take the input from the students & change the output of the motor so it can fly around the way the students want.

6) The Feather M4 will communicate with the ground control. Sharing data from the test station sensors with the ground control. This info will/can be used by students as they wish.

4.3 Acceptance Testing

We will be conducting the acceptance test as the last procedure of evaluation to see whether our development on our drones system will meet our client's requirements and expectations. We will be evaluating our drones with our client and advisor, Dr. Phillip Jones.

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

Our team has not reached the point of being able to provide testing results.

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 reiterate your goals. Then, reiterate 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.