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: 10/4/20 - Version 1

Executive Summary

Development Standards & Practices Used

List all standard circuit, hardware, software practices used in this project. List all the Engineering standards that apply to this project that were considered.

Summary of Requirements

The requirements for this project were given to us by Dr. Phillip Jones who wants us to develop a Mini 4 x 4 in, educational friendly drone. Who is easy to code with and can be controlled by radio frequency and or wifi. The drone will have a ground control station where the students can perform testing on it, program & 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 (Teensy 4.1)
- Radio & Power Mgt (STM32F030F4P6 Pwd Sup)
- Propellers
- Motors
- Coreless DC motors
- Expansion connector pins
- Battery/battery holder
- Gyro (MPU 6050)
- 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.

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

MicroCART is a programmable 4.5 by 4.5-inch drone used by CprE 488 students to learn about embedded programming for a niche task. MicroCART will allow students to demonstrate their ability to analyze data sent from the drone and problem-solve to provide stable flight. The end result is a drone with the ability to show its functionality, run complex algorithms, and provide sensory feedback to its operators. To develop the drone, we'll be designing it from the ground up and will incorporate designs from open source drone projects and develop our own designs.

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 x 4 inch programmable drone. The drone should be is easy to code with and can be controlled by radio frequency and or wifi. The drone will have a ground control station where the students can perform testing on it, program & recharge its battery. The total cost per drone needs to be less than $50.

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 interface will help students constantly get data from the drone and the ground station for future analysis.

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 ground station. 0.2
    - Drone is too heavy for take off. 0.1
    - Battery may die while 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 :

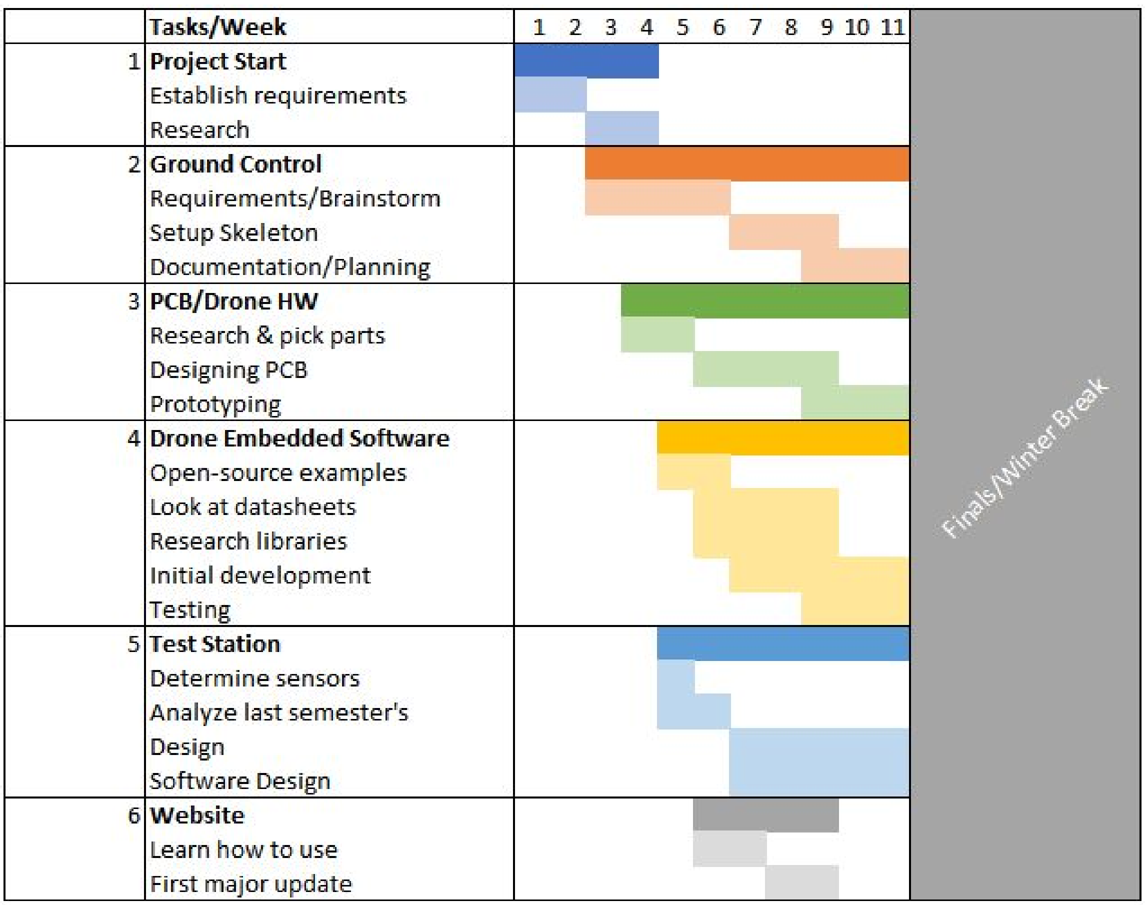
- Gain sufficient understanding in the functionality of a drone
- Building the smallest PCB possible (i.e. A PCB that can fit on a 4.5” X 4.5” drone) while trying to keep the layer size of the PCB to 2 layers
- Make all the sensors on the drone to communicate with ground stations and also send data with 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. 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 end product should be built for 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

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.