Collaborative Control of Ground and Aero Vehicles

ECE 445
Design Document

Team 19
Alvin Sun (yixiaos3)
Jialiang Zhang (jz23)
Mingda Ma (mingdam2)

TA: Andrew Chen
Spring 2021
March 1, 2021

Contents

1 Introduction

1.1 Problem and Solution Overview

Autonomous delivery over drone networks has become one of the new trends which can save a tremendous amount of labor. However, it is very difficult to scale up development due to the increasing likelihood of collision between multi-rotor drones and ground vehicles, especially when carrying payload. To actually have such a system deployed in big cities, we could take advantage of the large ground vehicle network which already exists with rideshare companies like Uber and Lyft as well as public transportation networks such as buses and mailing / delivery services. The roof of an automobile has plenty of space to hold packages and a drone network can optimize for flight time and efficiency while having minimal interference with the automobile's route. While this can dramatically increase delivery coverage and efficiency, the problem of safely docking a drone onto ground vehicles in motion remains a big challenge.

We aim at implementing the idea in a lab environment by developing a decentralized multi-agent control system that automatically synchronizes a drone with a moving ground vehicle when in close proximity. As a proof of concepts, the project takes the assumptions that vehicle states (such as its position and orientation) can be accurately estimated. The infrastructure of the lab, the drone, and the ground vehicle will be provided by the support of our generous sponsor, Professor Naira Hovakimyan. We will achieve the synchronized motion through a collaborative peer-to-peer control scheme. More specifically, the ground vehicle will estimate its own trajectory several seconds into the future, and will periodically send the trajectory to the drone. Since the drone cannot acquire absolute position read from the motion capture system, the ground vehicle is also in charge of estimating the drone's poses (through motion capture). The drone will then optimize its current control to track this future trajectory.

1.2 Visual Aid

As shown in Figure 1, we will design a collaborative control system so that the drone can accurately track the predicted trajectory of the ground vehicle in real-time. After taking off from random initial conditions, the drone will fly towards the ground vehicle and stabilize into a certain proximity range with respect to the ground vehicle. The problem of dynamic estimation is achieved through Vicon, an indoor motion capture system. In addition, we will also design a standalone alignment indicator that runs on board of the ground vehicle. This is built with a customized LED matrix which can indicate the quality of the spatial alignment between the two vehicles. This hardware will display an overall color of green to indicate a good alignment, and red for a bad alignment. It also estimates the relative poses between the ground vehicle and the drone and displays such relative displacement through LED patterns. As shown in Figure 2, the circle represents the position of the drone relative to the ground vehicle's orientation. The overlap between the circle and the center square indicates a good alignment situation.

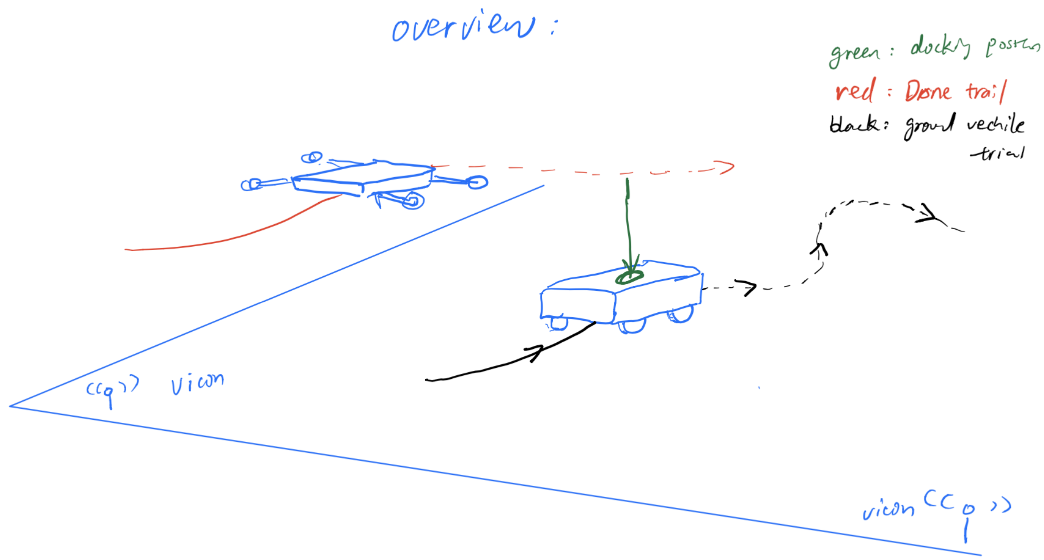
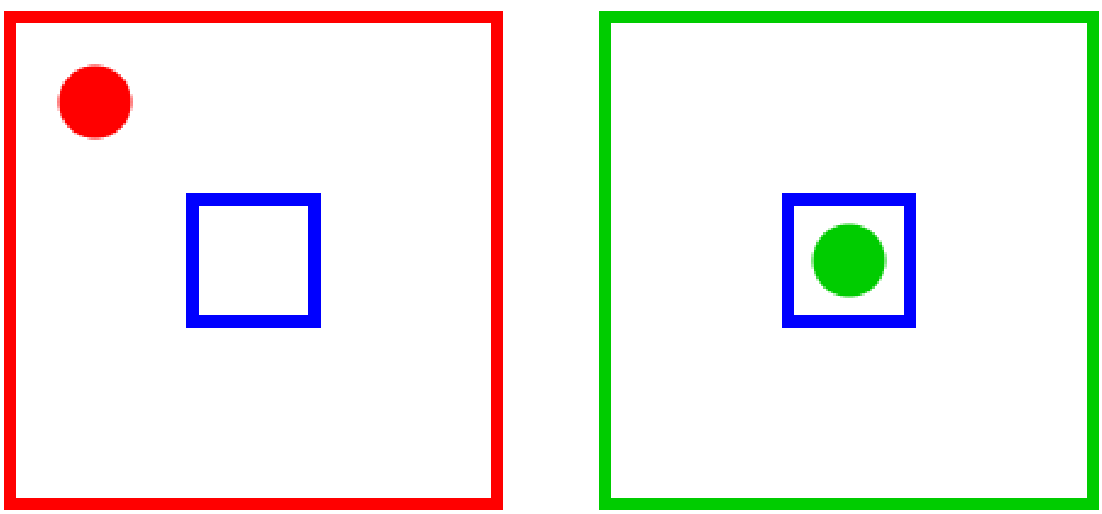


Figure 1: Overall Design Illustration



(a) Bad Alignment

(b) Good Alignment

Figure 2: Alignment Indicator Illustration

1.3 High-Level Requirements

The following requirements are all meant to be met under lab environment:

- The elapsed time between our aerial vehicle taking off and reaching the synchronous state (The drone could moving with the vehicle at the same speed. More details stated below) with our ground vehicle is no more than **20 seconds**.

- The spacial proximity error, at the synchronous state, between two vehicles is within a circle with **a radius of 30 centimeters**, centered at the spacial center of our ground vehicle.
- The LED indicator on our ground vehicle continuously reflects the spacial error between two vehicles with a delay of no more than **one second**.

2 Design

Block Diagram See Figure 3.

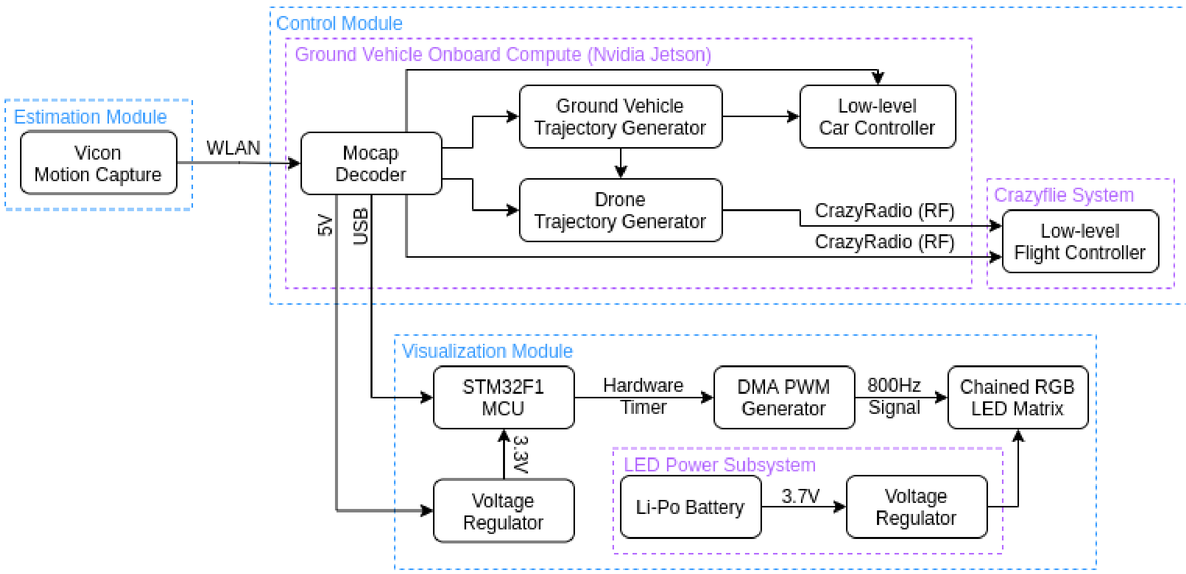


Figure 3: Block Diagram

Physical Design See Figure 4.

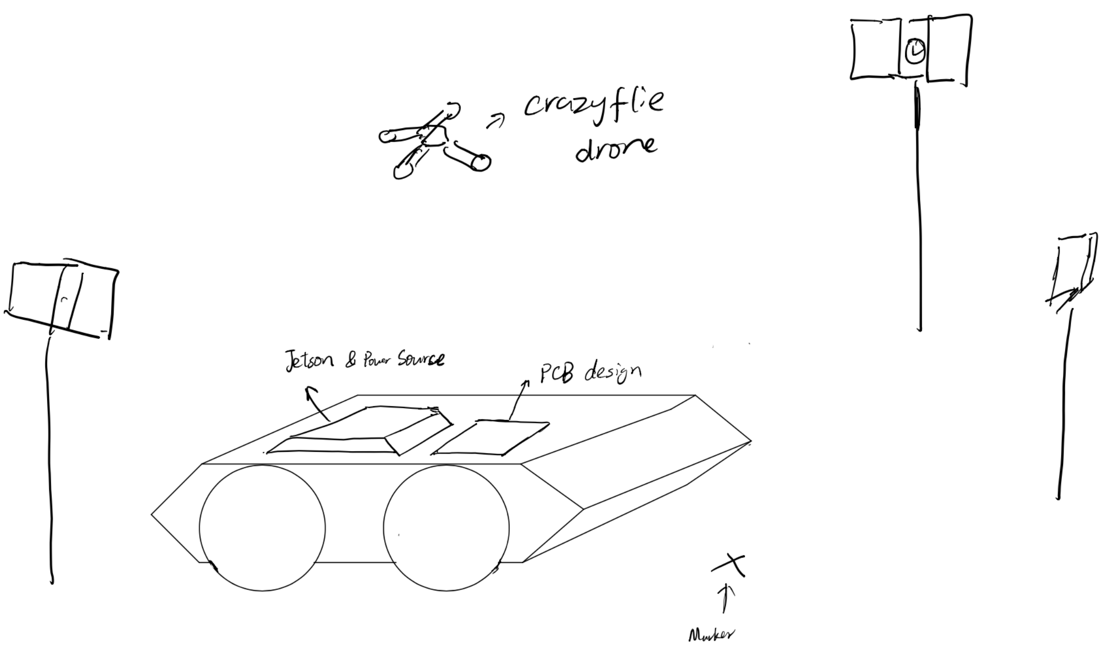


Figure 4: Physical Design

2.1 Estimation Module

The Vicon motion capture system is a commercialized indoor localization system that takes advantage of multi-view high-speed imaging technologies. To configure it properly, we will need to install reflective markers to both ground and aero vehicles and calibrate their extrinsic poses before using the localization data coming from the system. The requirement for this module is to successfully capture the calibrated poses of the two robots with the Vicon system.

Vicon Motion Capture System Vicon system uses high-speed cameras deployed around the lab to capture the motion of objects and collect the raw data. The Mocap decoder will be used to decode the collected raw data and broadcast it to local network for downstream tasks. Please refer to Table 1 for requirements and verification.

2.2 Control Module

This module is in charge of stabilizing both the ground vehicle and the drone. The overall requirement for this module is to achieve synchronous motion between the drone and the vehicle within certain proximity range.

2.2.1 Ground Vehicle Onboard Compute System

This system is in charge for handling multiple software tasks including algorithms computation and message dispatching. It must accomplish the following tasks:

- Read in and decode the location and orientation of both robots coming from the motion capture system.
- Pre-plan a feasible and safe trajectory for the ground vehicle.
- Control the ground vehicle to follow the pre-generated trajectory.
- Generate feasible and safe trajectory for the drone based on the local ground vehicle trajectory.
- Send serialized messages including trajectory and mocap data to the drone.

Mocap Decoder Vicon decoder takes the raw data collected by motion capture and calculates the corresponding positions and orientations of the ground vehicle and the drone. It then broadcasts the calculated results to different subsystems through Wireless Local Area Network (WLAN). Please refer to Table 1 for requirements and verification.

Ground Vehicle Trajectory Generator (GVTG) GVTG generates a pre-programmed trajectory that will be passed on to the low-level car controller. GVTG will be deployed on the Nvidia Jetson TX2 embedded platform. Please refer to table 2 for requirements and verification.

Requirements	Verifications
The Vicon system should be able to locate the given object accurately and broadcast its corresponding coordinates to all the devices within the predetermined communication channel based on the local Wi-Fi system.	Specify a certain channel and use a PC to access it. Print the value and check if it matches with the object's coordinates in Vicon.

Table 1: RV Table for Vicon and Mocap Decoder

Requirements	Verifications
This module should be able to generate a sequence of pre-planned way points that are inside the bound of the Vicon motion capture system.	Numerically test all generated way points against the allowed rectangular boundary defined by the Vicon motion capture system and make sure non of them are outside the boundary. Also plot the generated trajectories before executing the low-level trajectory tracker to ensure the way points resembles simple geometries such as circles and polygons.

Table 2: RV Table for Ground Vehicle Trajectory Generator

Low-Level Car Controller Controls the movement of the ground vehicle and ensure that it is accurately tracking the trajectory. Please refer to Table 3 for requirements and verification.

Requirements	Verifications
The controller should be able to connect to and pair the ground vehicle with a PS4 controller and allow the user to move the car with it.	By taking the measurements from Vicon, we can get the actual trajectory that the controller executed. We can then compare the execution with the generated trajectory and verify that the controller does track achieve those way points within 20cm radius accuracy.

Table 3: RV Table for Low-level Car Controller

Drone Trajectory Generator (DTG) Similar to GVTG, DTG generates a dynamically planned trajectory based on the local trajectory of the ground vehicle. This trajectory will then be passed on to the low-level flight controller on the drone for it to follow. However, due to the possible delays along the communication pipeline, DTG also needs to predict the ground vehicle's possible position and orientation within the next

2 seconds. DTG will also be deployed on Nvidia Jetson TX2 embedded platform. Please refer to Table 4 for requirements and verification.

Requirements	**Verifications**
1. The trajectory generator should allow the drone to follow the car within a proximity of 30cm of the vehicle.	1. Start to time the process when the drone and the car starts to move. 2. After the drone successfully synchronize with the car, move the car in random directions and check if the drone could still follow it autonomously. 3. Check the visualization module on the car to make sure it is always green.
2. The synchronization process mentioned above should be done within 20 seconds.	4. If step 2 is finished, check the timer to make sure the process takes less than 20 seconds.

Table 4: RV Table for Drone Trajectory Generator

2.2.2 Crazyflie System

Crazyflie is a commercialized product for nano-drone development. The requirement for this subsystem is to develop add-on firmware that can successfully decode the external measurement and trajectories data sent from the ground vehicle, and integrate those information into their already implemented low-level controllers.

Low-Level Flight Controller Controls the movement of the drone and ensures it is accurately tracking the trajectory. Please refer to Table 5 for requirements and verification.

Requirements	**Verifications**
The controller should be able to connect to and pair the drone with a controller and allow the user to move the drone with it.	1. Pair the controller. 2. Try to move the drone with the controller in all directions.

Table 5: RV Table for Low-level Flight Controller

2.3 Visualization Module

2.3.1 LED Power Subsystem

Power subsystem must be able to support a stable 3.6V voltage and 2A current.

Power Regulator Use correct resistors and capacitors to maintain the voltage of the power source to our designed value. Exact values of the resistors and capacitors remained to be decided. Please refer to Table 6 for requirements and verification.

Requirements	Verifications
The voltage regulator should be able to handle voltage of 4.5V-5.5V and convert it to a DC voltage at around 3.3V.	1. Connect the voltage regulator with a 5V power source. 2. Measure the output voltage and make sure it is within the range listed in requirements.

Table 6: RV Table for Power Regulator

Li-Po Battery One cell Li-Po battery used for power source. Please refer to table 7 for requirements and verification.

Requirements	Verifications
1. The battery should be able to provide a voltage of 3.2-4.2V to fulfill the requirements of the voltage regulator.	1. Connect the battery with an multimeter. 2. Make sure the voltage across the battery is within the required voltage range. 3. Connect the battery with a resistor of 2.5 Ohms.
2. The battery should be able to provide a current of no less than 1.92A. Since each battery is used to power 32 LED lights, each with an operation current of 60mA.	4. Measure the current using the multimeter to make sure that the current is higher than 1.92A.

Table 7: RV Table for Li-Po Battery

2.3.2 Visualization Module Overall

The visualization module must be able to communicate with the Mocap system and correctly display the correctness and quality of the synchronization by displaying the position of the drone and its relative position to the target platform. Please refer to Table 8 for overall requirements and verification.

STM32F1 MCU Used as the interface with Mocap Decoder using USB port. It is also in charge of controlling the behaviors of the LED matrix. Please refer to Table 9 for requirements and verification.

Chained RGB LED Matrix Used to indicate the quality of the synchronization between the ground vehicle and the drone. The matrix will mainly display the position of

Requirements	Verifications
1. The module can operate under a voltage of 3.7V, with a current of around 1.92A.	1. Connect the module with the required power source. 2. Send bit streams through USB to the module.
2. The module is able to display the patterns shown in the previous descriptions (figure 2).	3. Check the pattern.

Table 8: RV Table for Visualization Module Overall

Requirements	Verifications
1. The MCU should be able to take an input voltage of 3.3V± 0.1V. 2. With the given voltage range, the MCU should output a voltage between -0.5V to -1V as logic "0" to the LED and output a voltage above 3.3V as logic "1" to the LED.	1. Connect the MCU with a voltage source of the given required voltage. 2. Measure the output pins that are going to be used for LEDs individually and confirm that each of them fulfills the required voltage.

Table 9: RV Table for STM controller

the drone and landing position/range of the drone. The LED light box within the matrix will remain red when the synchronization has not been achieved. Once the synchronization has been achieved, the light box will turn to green. Please refer to Table 10 for requirements and verification.

Requirements	Verifications
1. The LED can operate under a voltage of 3.7±0.1V. 2. The data input voltage of the LED should be able to interpret a voltage above 3.7V as a logic "1" in its output, and interpret a voltage below -0.5V as a logic "0" in its output.	1. Connect the LED to a power source with a voltage that fulfills the given requirements. 2. Feed the Din pin with a voltage around 3.7V and check if the light is on. 3. Feed the Din pin with a voltage around -1 to -0.5V and check if the light is off.

Table 10: Chained RGB LED Matrix

2.4 Schematics

We break the schematics down into smaller submodules for visualization purposes.

1. Figure 5 shows the schematics around the STM32 controller. The circuit includes the required crystal oscillator and the power de-noising circuit.

2. Figure 6 shows the schematics for user I/O interfaces, where a reset button and a indicator LED is integrated with the micro controller.

3. Figure 7 shows the schematics of the connection headers, including USB connection, battery connection, and debug pin header.

4. Figure 8 shows the circuit for the addressable LED Matrix built with NeoPixel 5050 LEDs. The schematic also includes capacitor per LED pixel for the purpose of burst current absorption.

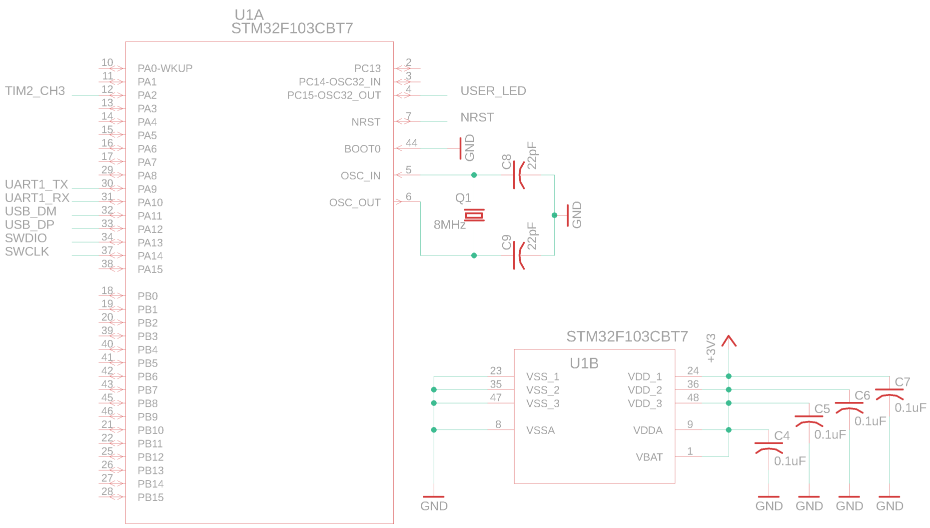


Figure 5: Schematics - STM32F103 Micro Controllers and Peripherals

2.5 Board Layout

The PCB routing layout is shown in Figure 9.

2.6 Tolerance Analysis

The most important parameter within our design is the proximity range between the ground vehicle and the drone. This proximity number is critical, because we need a safe range for the drone to land safely onto the vehicle without missing. We decided to set the proximity range to 20cm when the speed of the drone is 2m/s. The detailed explanation is listed below:

Since the Vicon system within the lab works with a frequency of 100 HZ (meaning it takes 100 pictures per second), we can conclude that the time period between two pictures is 0.01s. During this period of time, the drone relies completely on its own on-board feedback

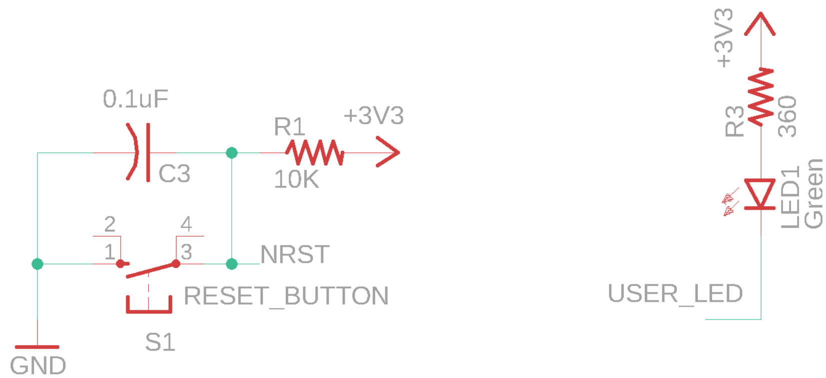


Figure 6: Schematics - User I/O

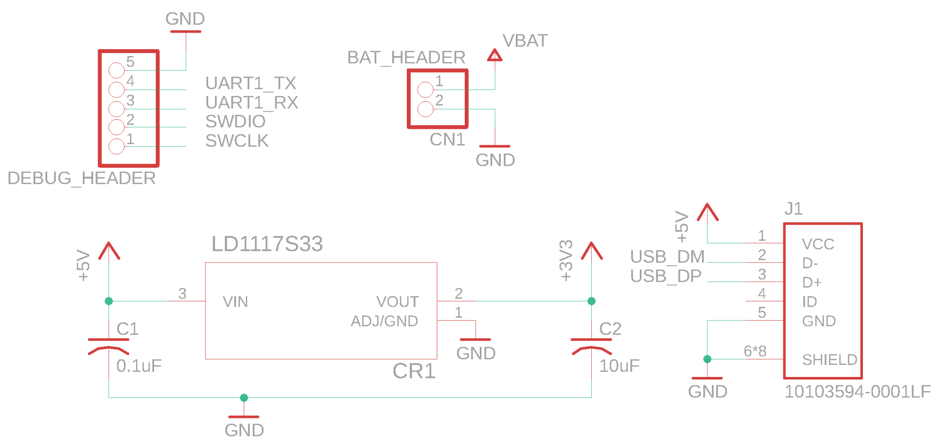


Figure 7: Schematics - Connection Headers

Figure 8: Schematics - LED Matrix

controller to keep it on the right track. The controller is equipped with a Kalman filter to defend against any potential disturbance. For the worst case scenario, we assume the filter fails completely, and the drift between 0.01s is

$$0.01 * 2 = 0.02m = 2cm$$

However, since in the real experiments, there will be delays within communication and feedback controller, we set the proximity range to a larger value (20cm). This number is also allowed since the length of our vehicle is around 50 cm and the length of the drone is about 7 cm. The 20cm difference will still make enough space for the drone to land safely.

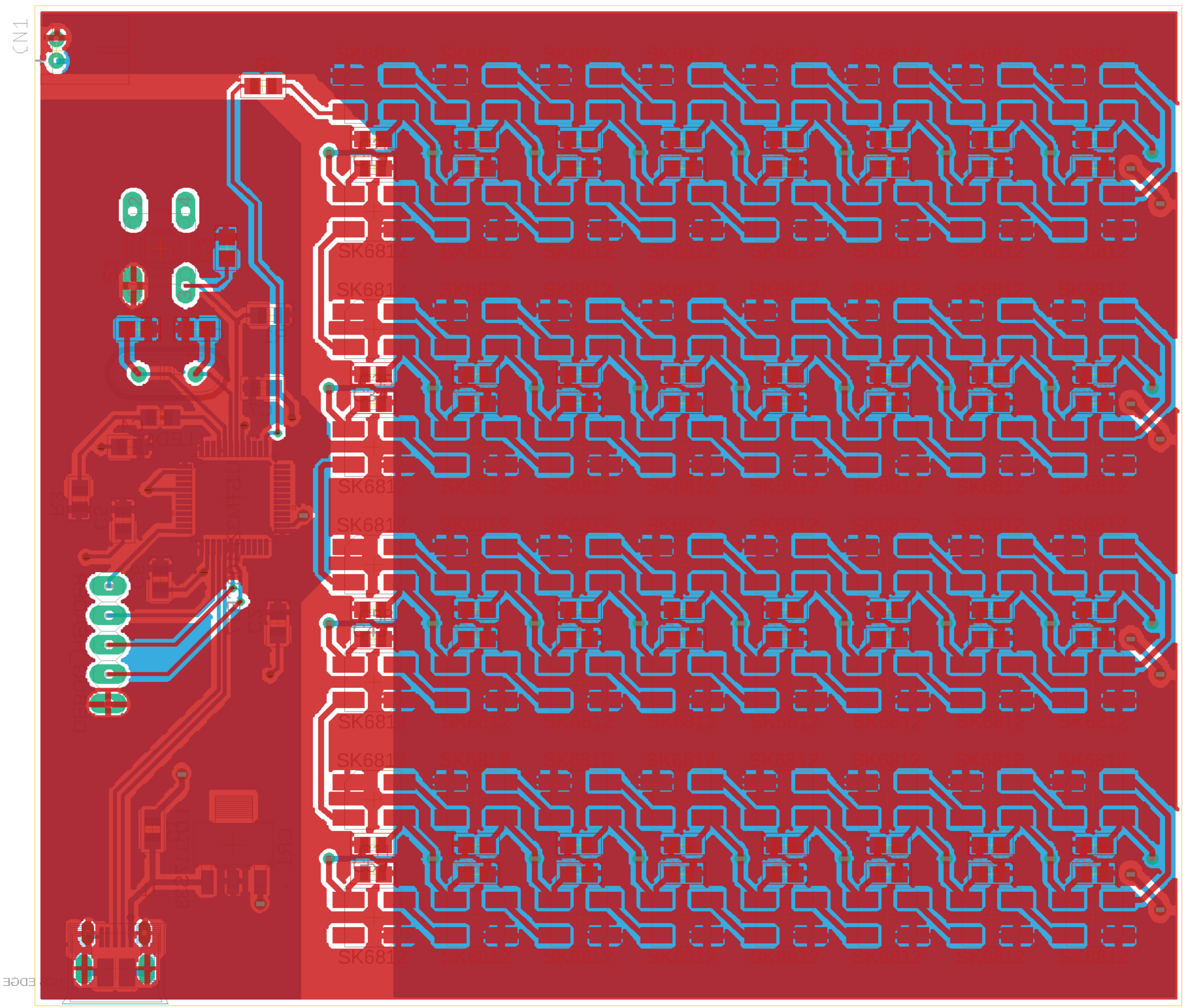


Figure 9: PCB Layout

3 Cost and Schedule

3.1 Cost Analysis

3.1.1 Material Bill

Aside from basic lab infrastructures (the ground vehicle, the drone and the Vicon system), the material costs of our projects are listed in Table 11.

Component	Attributes	Quantity	Unit Price ($)	Total Price ($)
Capacitor	22pF	10	0.07	0.72
	0.1μF	20	0.08	1.52
	1μF	100	0.02	2.2
	10μF	10	0.14	1.44
Resistor	150Ω	10	0.05	0.54
	360Ω	20	0.07	1.44
	10KΩ	10	0.05	0.54
Voltage Regulator	3.3V@800mA	3	0.42	1.26
	5V@3A	2	1.26	2.52
Micro Controller	STM32F103	4	6.57	26.28
LED	NeoPixel 100-Pack	1	39.95	39.95
	10mA 0805	10	0.14	1.44
Level Shifter	8-Channel	3	1.1	3.3
Tactile Switch		10	0.31	3.12
Micro USB Header	R/A	4	0.68	2.72
JST Header	Vertical	3	0.17	0.51
	R/A	3	0.17	0.51
Crystal Oscillator	8MHz	5	0.44	2.2
Total				92.21

Table 11: Material Costs

3.1.2 Labor Cost

Our estimated development cost is $40/hour, 10 hours/week for three people. This semester is 16 weeks long, therefore our total cost for development is:

$$3 * 40 * 10 * 16 = 19200$$

3.2 Schedule

See Table 12.

Week	Task	Responsibility
3/1	Design document check Finalize design document Prepare DDC presentation	All All All
3/8	Design review sign up Getting access to the lab Set up environment for Vicon, establish an object for the drone Set up environment for Jetson on the vehicle and enable basic remote control for it	All All Alvin Alvin, Jialiang
3/15	Teamwork evaluation First round PCB order Finalize PCB design and schematics Upload and order PCB Soldering assignment Simulation assignment	All All Alvin, Jialiang Mingda All All
3/22	Second round PCB Order Control algorithm for the vehicle Control algorithm for the drone Soldering PCB components and integrate it onto the vehicle	ALL Alvin, Jialiang Alvin, Mingda Jialiang
3/29	Optimize control algorithms for docking Implement and verify different LED patterns	Alvin, Mingda Jialiang, Mingda
4/5	Individual progress report Debug	ALL ALL
4/12	Debug Prepare slides for demo	ALL Jialiang, Mingda
4/19	Mock demo Finalize demo details	ALL ALL
4/26	Demonstration and mock presentation	ALL
5/3	Presentation Final Papers	ALL ALL

Table 12: Planned Schedule

4 Ethics and Safety

Although our project by itself casts little to no ethics or safety concerns because it is in a lab environment with comprehensive safety measures, as a proof of idea, it may raise the following issues:

- **Conflicts of Interests:** The successful deployment of such networks may significantly reduce the needs for labors in relevant industries, taking jobs from workers, and causing conflicts between companies and workers / unions. Such consequences could go against #3 of the IEEE Code of Ethics "to avoid real or perceived conflicts of interest whenever possible, and to disclose them to affected parties when they do exist." [1] We currently do not have a solution for this and consider it far beyond our control.

- **Possible Unlawful Misuse:** Such a autonomous delivery system might offer more vacant for smuggling, taking advantage of the unsupervised time before the packages reaching the destinations, whereas increasing the difficulty for tracking such crimes. Such consequences, together with the next two in the list, would go against #1 of the IEEE Code of Ethics "to hold paramount the safety, health, and welfare of the public, to strive to comply with ethical design and sustainable development practices, to protect the privacy of others, and to disclose promptly factors that might endanger the public or the environment." [1] To avoid such unlawful activities and minimize their damage, recording every delivery specifications and detecting for contraband before the package is sent into the autonomous system is recommend.

- **Potential Hazard to Public Safety:** Aerial vehicles might cause serious secondary injuries under potential misbehavior of the ground vehicles since the drone can cause heavy impact and consequent explosion under high speed. UAV-related incidents are not unusual in today's society as shown by [2]. These experiments [3] suggest the serious aftermaths. In response, we should advocate that drivers to drive safely or use reliable auto vehicle systems to minimize the possibility of accidents as well as to build a emergent evasion response for the drones.

- **Privacy Concern:** In industries, the cyber-security measurement at ending terminals such as the drones could be overlooked. A breach can cause serious violation to public privacy. Potential misuse includes stalking and leaking private information. To protect the civic privacy, the whole system should be protected by reliable hardware / software security such that it is maintained and examined periodically.

With aforementioned concerns, some positive aspects are listed below:

- **Productivity:** Without doubt, autonomous delivery systems could tremendously increase the productivity. This benefit and the next point, help us to develop #1 of the IEEE Code of Ethics [1].

- **Service and User Experience:** Without human intervention, the delivery systems would avoid much mistakes of express and significantly improve the user experience.

- **Social Progress:** The wide use of such a system could push the progress of our society in many aspects, such as productivity, economy, legislation, cyber-security, and so on. This complies with #2 of the IEEE Code of Ethics "to improve the understanding by individuals and society of the capabilities and societal implications of conventional and emerging technologies, including intelligent systems." [1]

References

[1] IEEE, "Ieee code of ethics." https://www.ieee.org/about/corporate/governance/p7-8.html. Accessed: 2021-02-18.

[2] "List of uav-related incidents." https://en.wikipedia.org/wiki/List_of_UAV-related_incidents, Dec 2020. Accessed: 2021-02-18.

[3] E. Tegler, "What happens when a drone crashes into your face?." https://www.popularmechanics.com/flight/drones/a28774546/drone-head-collision/, Aug 2019. Accessed: 2021-02-18.