Drone Delivery System for Takeaway Business

Design Document

By

Ximo Wang (ximow2)

Yanbing Yang (yanbing7)

Yang Chen (yangc7)

Yuzheng Zhu (yz83)

Design Document for ECE 445, Senior Design, Spring 2024

TA: Yanzhao Gong

17th March 2024

Project No. 7

Contents

1. Introduction

1.1 Problem and Solution Overview

We are going to design and realize an airway delivery system with drone, container, and cloud server. Delivery of light weight, medium range, fast response within a city is a strong demand especially during rush hour. Traditional airway delivery drones with GPS navigation are not precise enough for landing in limited space. Existing delivery drones on the market usually require manual operation during picking and placing the goods, while our design aims to achieve the process automatically. Intelligent drones are still in the early progress of being developed. The market and application of drones are expanding rapidly, which proves that we are participating in a research field with a high demand.

Foodpanda has piloted food deliveries in Singapore using multirotor drones from ST Engineering and in Pakistan using VTOL drones from Woot Tech. Flytrex has delivered over 55K orders by drones in three towns in North Carolina and Texas since 2022, including Starbucks coffee, Walmart, Chick-fil-A, Papa John's pizza and more.[1] Our solution differs from existing solutions since it’s more convenient and cheaper. The drone and the delivery container are integrated, with the container serving both as a storage unit and a landing pad. Both merchants and customers need only to concentrate on the order itself. Placing an order through the program is all that's required, as all decision-making, delivery, and interfacing are fully automated. We use low-cost standard components to reduce expenses.

The system we are going to construct can be divided into 3 subsystems due to the platform: Intelligent drone, ground-based terminal, and cloud server. All these subsystems are connected to each other by communication network and cooperate during the delivery tasks.

The first significant part is the drone subsystem. The drone carries a host computer which is responsible for autopilot of drone and the communication with the ground terminal. At the same time, the drone is equipped with RTK devices and GPS locator to achieve precise landing and path tracking. The structure of the drone is also designed for clamping delivery boxes and assisting to land precisely.

Secondly, the ground terminal subsystem with a landing platform and storage cabinets can provide the RTK signal for the drone and guide the drone to land precisely. The elevator and claws equipped on the terminal can achieve the function of receiving and sending deliveries.

The third component of the system is the server and APP for data processing and task planning. The other subsystems can communicate with the server all the time during the delivery task by wireless network. The cloud server is responsible for managing the information exchange between the clients and the system devices and giving the command to them.

1.2 Visual Aid

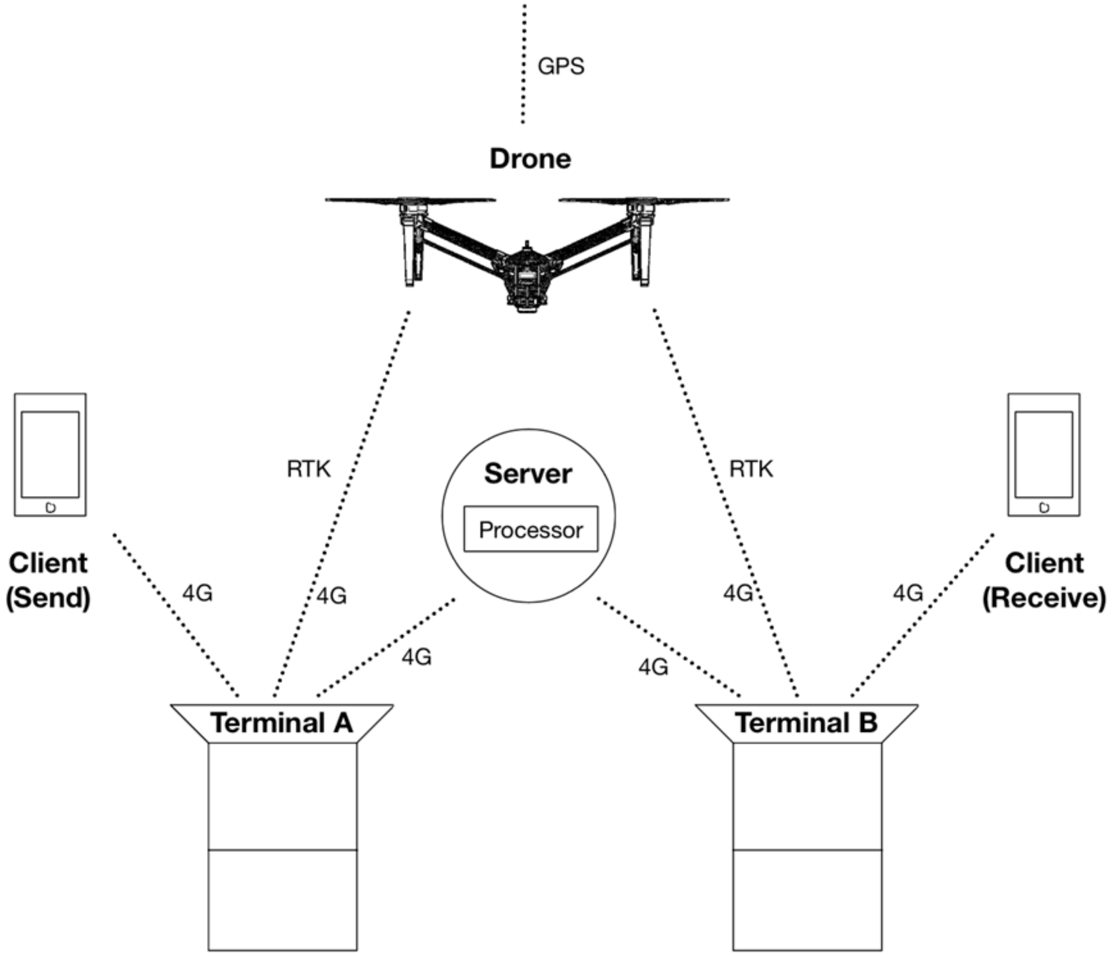

Figure 1: System View Visual Aid

1.3 High-level requirements list

- Flight distance should be longer than 4km within 5min when carrying a maximum load of 3kg in a single flight.
- Flight control system should allow the drone to take off and land automatically, avoiding severe crash on possible obstacles.
- Ground terminal should be able to receive messages from drone and cloud service. What's more, it should be able to determine whether to take out or store the goods based on the messages and physically conduct it.
- Mobile applications should be able to monitor and command the automatic process of the delivery with the assistance of cloud server.

2 Design

2.1 Block Diagram

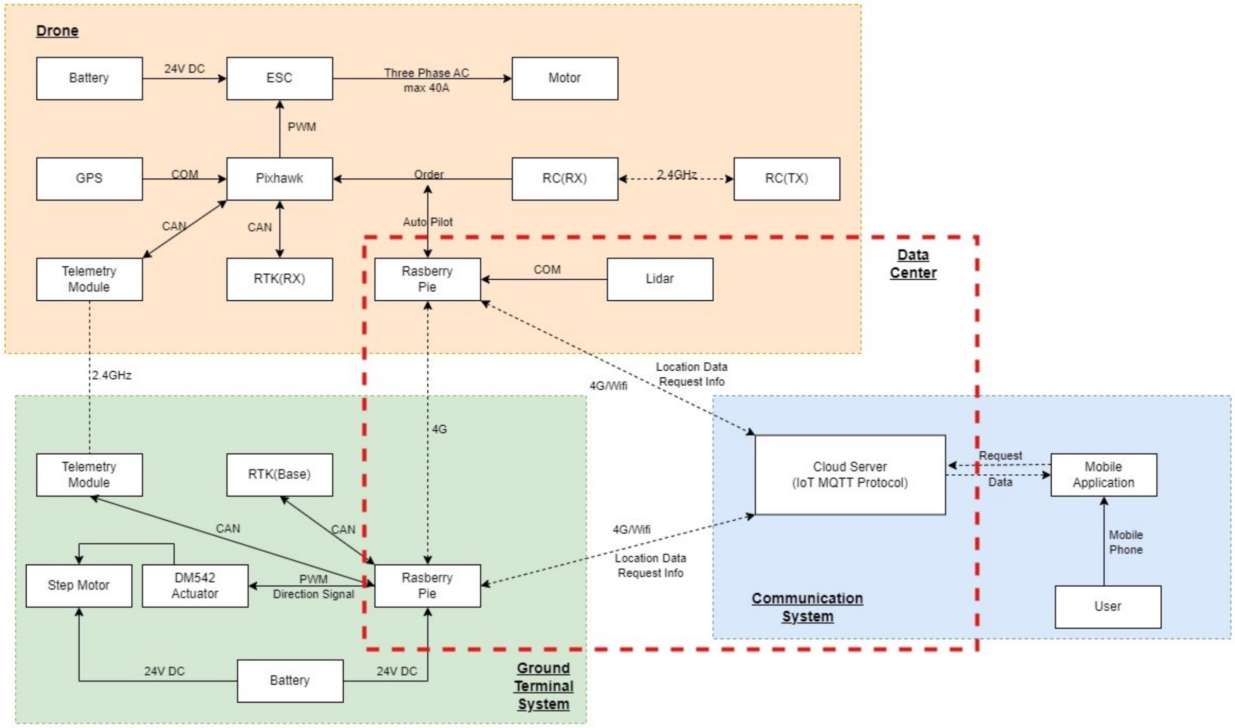

Figure 2: Overview Block Diagram

The drone carries two 4700 mAh batteries in parallel as the power source, which provide 253 Wh energy for the drone’s flight and calculation. The designed max take-off weight is 7.5kg and the unloaded mass of the drone is 4.5kg, so that the drone can carry 3kg load at maximum and conduct flight tasks longer than 4km. Pixhawk is the control center of drone while Raspberry Pi bonding the drone with the whole system. Connected with GPS, telemetry module and Raspberry Pi, Pixhawk will be able to guide the drone, by driving motors through ESC (electronic speed controller). At the same time, Raspberry Pi will fetch orders from server though 4G network.

The drone carries two 4700 mAh batteries in parallel as the power source, which provide 253 Wh energy for the drone’s flight and calculation. The designed max take-off weight is 7.5kg and the unloaded mass of the drone is 4.5kg, so that the drone can carry 3kg load at maximum and conduct flight tasks longer than 4km. Pixhawk is the control center of drone while Raspberry Pi bonding the drone with the whole system. Connected with GPS, telemetry module and Raspberry Pi, Pixhawk will be able to guide the drone, by driving motors through ESC (electronic speed controller). At the same time, Raspberry Pi will fetch orders from server though 4G network.The Raspberry Pi equipped in Ground terminals will be informed by cloud server of the coming goods or the request of getting goods. Then it will control the motor to conduct corresponding actions, including take out and store the goods. Cloud server handles data from all devices including mobile application and Raspberry Pis. Handling data processing and ensure all devices are informed and there’s no data collision or packet loss.

2.2 Physical Design (if applicable)

The physical design appearing in the project mainly focuses on the structure design of the delivery UAV and the ground transfer station. The dynamic layout of the drone is improved based on the H4 quadrotor structure. There are specially designed conversion structures for flight and landing, so that it can meet the requirements of flight obstacle avoidance and stable landing in two situations respectively. In addition to the GPS and RTK modules needed for flight control and positioning, the drone will also carry lidar and Raspberry Pi devices for navigation and obstacle avoidance in autonomous flight. The ground terminal station will provide the landing platform for the UAV and storage space as a cabinet, and the container controlled by the Raspberry Pi will automatically access the cabinet and synchronize the data information to the cloud server.

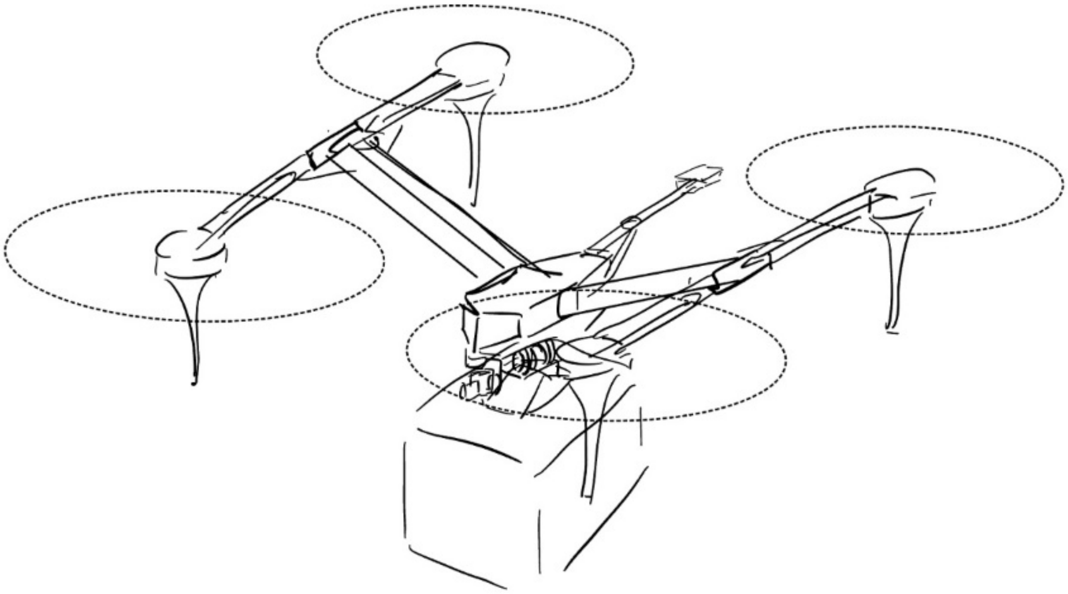

Figure 3: Sketch of the Drone

2.3 Drone Subsystem

2.3.1 Drone Skeleton Design

The multi-rotor UAV skeleton used in this project is specially designed for its working conditions and is individually designed and manufactured under the joint consultation of the members of the group. The structure of the drone is improved on the structure of the traditional H4 quadrotor drone. Two T-shaped arms are connected to the body by hinges (Figure 2.3.1). On the premise of ensuring flight stability, functions such as transferring center of gravity and adjusting lift configuration are added. The two forms of the drone are the auxiliary precise landing scene that needs high flexibility (Figure 2.3.1 left picture) and the high-speed flight scene that needs high efficiency and stability (Figure 2.3.1 right picture). The conversion between the two forms is controlled by a set of four-link links driven by screw rods (Figure 2.3.2).

The landing support of drone is a non-foldable design, but it is equipped with shock absorber to reduce the impact of landing, and its damping and elastic parameters still need to be tested in the later stage to obtain the optimal solution.

The load-bearing structure of the body are mostly carbon fiber resin composite materials, aluminum alloy standard parts and CNC manufactured parts, while the others are stainless steel standard parts and FDM or SLS 3D-printed industrial plastic parts. Most of the structures are enough to withstand the normal working load in FEA simulation.

Table 1: Drone Skeleton Requirements and Verifications

Requirement	Verification
1. Structure is strong enough to be able to bear the load during flight and landing impact. Capable to convert between 2 forms according to the working condition	1. a. CAD the model of the skeleton and apply the theoretical load on the model in FEA. b. Design the linkage structure and simulate it in the CAD model to verify the kinematic characteristic. c. Build the first version prototype to verify the assembling rationality

Figure 4: Improved based on H4 skeleton

Figure 5: Landing form and flying form

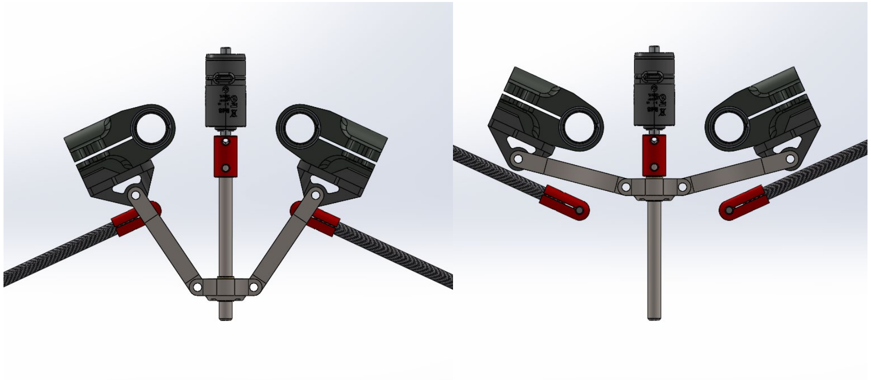

Figure 6: Screw rod driven linkage form conversion

2.3.2 Dynamic Analysis and Power Configuration

The dynamic layout is the traditional H4 quadrotor structure, and the lift force, flight attitude and flight speed are controlled by adjusting the rotation speed of the four rotors located at the four vertices of the approximate square. The diagonal rotors rotate in the same direction, and the adjacent rotors rotate in the opposite direction, so that the Roll, Yaw and Pitch three axes can be controlled independently while maintaining the lift force. The lift center of the UAV is located on the plane where the rotor is located, and the center of mass of the UAV can be adjusted by form transformation: In the landing form, the center of mass and the center of lift are approximately in the same plane, so that the UAV is in a neutral equilibrium, the attitude is easier to control, the response is faster, and the movement is more flexible, so as to ensure that it adjusts its positioning in time and responds to interference (such as crosswind and vortex ring state) during the landing process. In the flying form, the center of mass is located about 25cm below the center of lift, so that the drone is in a stable equilibrium, the attitude is more stable, the anti-disturbance ability is stronger, and the power and efficiency loss due to the controlling of attitude under high-speed flight are reduced. In terms of power configuration, the designed no-load take-off weight of the drone is 4.5kg, the maximum take-off weight is 7kg, the power supply uses 6S Li-Po battery, the theoretical endurance at maximum take-off weight is 10min, and the one-way flight can reach the destination in 5km at the speed of 30km/h, which meets the demand of small-weight and short-distance take-out delivery in the city. The specific power equipment parameters are listed in the following table:

Table 2: Power configuration of the drone

Battery Voltage	**21.0~25.2 V (6S)**
Battery Capacitance	4700 mAh * 2 in parallel
Motor KV	340 KV
Propeller specification	1760 (17’’ diameter)
Motor max thrust	8.4 (2.1 kg * 4)
Take-off weight (TOW)	4.5 kg
Endurance at TOW	22 min
Max TOW	7 kg
Endurance at Max TOW	10 min
Max diagonal size	1330 mm
Height	360 mm

Table 3: Drone dynamic configuration

Requirement	Verification
1. Max take-off weight should be larger than 7 kg. 2. Flight endurance at max TOW should be longer than 10 min. 3. The flight form can be transformed back and force with a stable attitude.	1. Test the motors and rotors respectively and make sure the max thrust is over 2 kg. 2. Obtain the endurance at different load and find the max TOW which satisfies the requirement. 3. Test the transforming structure during flight and record the drone’s attitude.

2.3.3 Flight Controller and Obstacle Avoidance

The flight controller of our drone is an open-source Pixhawk flight controller equipped with Ardupilot firmware, which supports flight control of H4 quadrotor power layout. The supporting software "Mission Planner" enables parameter setting and mission planning. The attitude control and motion monitoring of the flight will be carried out by the IMU data, GPS positioning and RTK base station monitoring data, which will ensure the accurate and stable flight and landing without remote control. Autopilot will be executed by two modules, Pixhawk and Raspberry Pi. Serving as a mini-PC, Raspberry Pi will transmit the origin and destination information of the goods obtained from the server to the ground station and plan the general task route. When the path is given, Pixhawk will guide the drone to the destination. At the same time, the drone is equipped with RPLIDAR C1 lidar for real-time obstacle avoidance during flight. When detects obstacles on the flight path, it will send the information to Raspberry Pi for path avoidance calculation and sends the new path to flight control.

Table 4: Navigation and obstacle avoidance system

Requirement	Verification
1. Pixhawk should be able to get command from Raspberry Pi and execute. Raspberry Pi should be able to fetch order information from cloud server and plan the path. Using ToF lidar, Raspberry Pi should allow the drone from severe crash on obstacles.	1. a. Test basic actions commanded by Raspberry Pi to verify the reliability of Pixhawk b. The path planning program could be tested offline, using pure software. c. Plan a path with a building on its way, check whether the drone will adjust the altitude in advance to avoid crashing.

2.3.4 Data Communication and Remote Control

The UAV and server need to receive and report mission information and execution status through 4G wireless signals, and its mini-PC will be equipped with LTE communication module to synchronize information with the server in real time. In addition, the HERE3 RTK module carried by the UAV will relate to the ground base station, calculate the flight attitude and motion data in real time, and feed back to the flight control through the serial port for compensation, so that the UAV can accurately locate and land at a fixed point in the automatic flight. Although receiving mission information and performing flight tasks are all independently performed by mini-PC and flight control, remote control is still needed to take over out-of-control situations during the test phase of the UAV. Pixhawk flight controller supports 2.4G s.bus communication and can interrupt the automatic flight mode at any time through remote control to ensure safety in case of accident.

Table 5: Communication and remote-control subsystem

Requirement	Verification
1. HERE3 module should enable accurate take-off and landing.	1. Check if the drone can take off from the terminal and land on it automatically.
2. Remote control needs to be a qualified backup.	2. Try to take over control during the flight.

2.4 Ground Terminal Subsystem

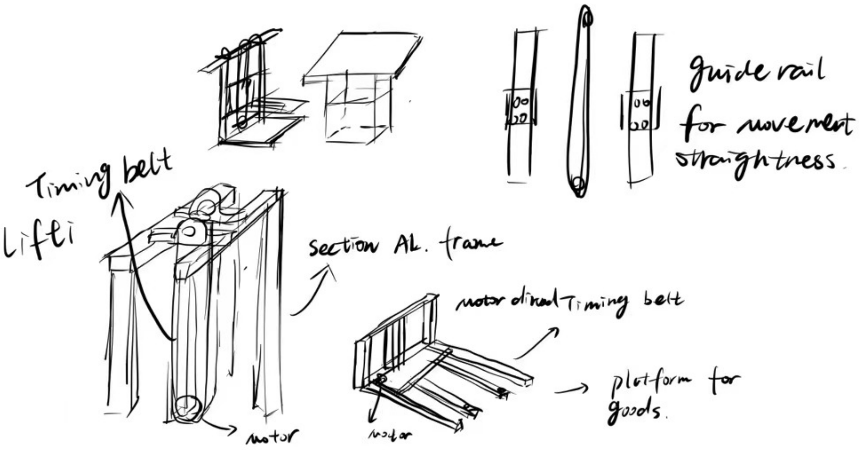

Figure 7: Sketch of Ground Terminal Subsystem

2.4.1 Ground Terminal Frame

The Ground Terminal has selected the 2020A section aluminum as the material for the framework of both the containers and the automatic lifting-picking machine. Such material is easy to cut, drill, and connect, which facilitates the design of containers of any size. The use of corner brackets and T-nuts, accompanied with side panel, ensures the strength of the frame and makes it capable of bearing a vertical load and torque from platform.

Table 6: Ground Terminal Frame Requirements and Verifications

Requirement	Verification
1. Capable of withstanding a maximum platform load of 10 kilograms, with a minimum of 2 safety factor.	1. Build the frame in simulation software for load testing; construct a 1:1 prototype. Gradually add load until failure to test the load capacity.

2.4.2 BF5M-15 Timing Pulley & 5M Timing belt

This belt-pulley combination is responsible for achieving the vertical movement of the lift platform. The pulley will be paired with a 57-step motor and according to the selection manual, a pulley model of 5M-15 teeth is sufficient to bear a vertical load of 100N. The belt will engage with the platform through a clamping plate to connect plate, and screws, which will transfer the load to the 57-step motor.

Table 7: BF5M-15 Timing Pulley & 5M Timing belt Requirements and Verifications

Requirement	Verification
1. Bear a vertical load of 100N with a minimum of 2 safety factor.	1. a. Take a section of the belt and secure both ends to the tensile testing bench. b. Gradually increase the pressure until the belt breaks. c. Observe if the breaking stress is more than twice the required load.

2.4.3 57 Step Motor.

This motor is mounted at the bottle of lifter framework. It’s powered and driven by a DM542 2H Microstep Driver, and works in conjunction with a 5M-15 belt and pulley transmission system to move the platform up and down.

Table 8: 57 Step Motor Requirements and Verifications

Requirement	Verification
1. In conjunction with the timing belt, the minimum movement increment does not exceed 1cm.	1. a. Correctly connect the motor to the power supply and Microstep Driver. b. Correctly connect the motor to the timing pulley-timing belt system. c. Mark a point on the timing belt and set a fixed reference object beside it. d. Input 100 pulse signals on the Driver and record the distance L by which the marked point on the timing belt advances. The minimum advance distance unit is thus L/100
2. Can provide 3.6 Nm of torque.	2. a. Correctly connect the motor to the power supply and Microstep Driver. b. Correctly connect the motor to the timing pulley-timing belt system. c. Fix one end of the timing belt to a force sensor, then output the motor's maximum torque until the rotor stops. The output force multiplied by the radius of the timing pulley equals the maximum output torque.

2.4.4 DM542 2H Microstep Driver

This driver is used in conjunction with the 57-step motor. It receives high- and low-level direction signals and PWM pulse signals from the Raspberry Pi's GPIOs, and controls the 57-step motor to operate according to the script on the Raspberry Pi. The driver is powered by a 4V-6000mAh li-ion battery, and then transfers energy to the motor and drive it.

Table 9: DM542 2H Microstep Driver Requirements and Verifications

Requirement	Verification
1. Can drive the 57-step motor	1. a. Connect it correctly to the power supply, Raspberry Pi, and 57 step motor. b. Generate a PWM signal with a function generator that has a high level of 5V, a low level of 0V, and a frequency of 10Hz, then connect the function generator to the driver c. Turn on the power and observe if the motor can be driven.

2.4.5 42HS28 Step Motor.

This motor is mounted on the platform. It’s powered by a 4V-6000mAh li-ion battery, driven by a TB6600 2H Microstep Driver, and works in conjunction with a 2GT-6 belt and pulley transmission system to move the bar on the platform outward to get goods.

Table 10: 42HS28 Step Motor Requirements and Verifications

Requirement	Verification
1. In conjunction with the timing belt, the minimum movement increment does not exceed 1cm.	1. a. Correctly connect the motor to the power supply and Microstep Driver. b. Correctly connect the motor to the timing pulley-timing belt system. c. Mark a point on the timing belt and set a fixed reference object beside it. d. Input 100 pulse signals on the Driver and record the distance L by which the marked point on the timing belt advances. The minimum advance distance unit is thus L/100
2. Can provide 1Nm of torque.	2. a. Correctly connect the motor to the power supply and Microstep Driver.

	b. Correctly connect the motor to the timing pulley-timing belt system. c. Fix one end of the timing belt to a force sensor, then output the motor's maximum torque until the rotor stops. The output force multiplied by the radius of the timing pulley equals the maximum output torque.

2.4.6 TB6600 2H Microstep Driver

This driver is used in conjunction with the 42-step motor on the platform. It receives high- and low-level direction signals and PWM pulse signals from the Raspberry Pi's GPIOs, and controls the 42-step motor to operate according to the script on the Raspberry Pi. The driver is powered by a 4V-6000mAh li-ion battery, and then transfers energy to the motor and drive it.

Table 11: TB6600 2H Microstep Driver Requirements and Verifications

Requirement	Verification
1. Can drive the 42-step motor	1. a. Connect it correctly to the power supply, Raspberry Pi, and 57 step motor. b. Generate a PWM signal with a function generator that has a high level of 5V, a low level of 0V, and a frequency of 10Hz, then connect the function generator to the driver c. Turn on the power and observe if the motor can be driven.

2.4.7 Voltage regulator.

This regulator supplies the required 5V to two motor drivers from the 3.3V signal supplied by Raspberry Pi GPIOs. This chip must be able to handle both the peak input from the Raspberry Pi, and then output stable 5V signal.

Table 12: Voltage regulator Requirements and Verifications

Requirement	Verification
1. Provides 5V from a 3.3V source. 2. Can operate at current within 300mA	1 &2. a. Correctly connect the voltage regulator with power supply and voltmeter. b. Use power supply to provide a 3.3V input with 300mA current.

	c. Measure the voltage at output with an oscilloscope to check whether the output voltage is 5V.

2.4.8 Raspberry Pi

The Raspberry Pi is responsible for running scripts, utilizing the RPi.GPIO library to output PWM signals and high/low level direction signals through the GPIO, to control the motor's forward and backward movement, as well as the precise distance of these movements.

Table 13: Raspberry Pi Requirements and Verifications

Requirement	Verification
1. Can successfully output PWM signal and 3.3v high voltage signal.	1. a. Correctly power the Raspberry Pi. b. Write a test code to generate PWM and high voltage signal in two of the GPIOs. c. Measure the voltage at output with an oscilloscope.

2.4.9 2GT-6 Timing Pulley & 2GT Timing belt

This belt-pulley combination is responsible for achieving the horizontal movement of the Reaching bar on the platform. The pulley will be paired with a 42-step motor and according to the selection manual, a pulley model of 2GT-6 teeth is sufficient to bear a horizontal load of 20N. The belt will engage with the platform through a clamping plate to connect plate, and screws, which will transfer the load to the 42-step motor.

Table 14: 2GT-6 Timing Pulley & 2GT Timing belt Requirements and Verifications

Requirement	Verification
1. Bear a vertical load of 20N with a minimum of 2 safety factor.	1. a. Take a section of the belt and secure both ends to the tensile testing bench. b. Gradually increase the pressure until the belt breaks. c. Observe if the breaking stress is more than twice the required load.

2.4.10 24V-6000mAh li-ion battery

The 24V-6000mAh li-ion battery will be connected to two motor drivers and provide power for our motor. Since we designed to use one fully-charged battery for 12hrs, nearly 5000mAh capacity is required. Considering the capacity loss after long-term use, we set a standard of 6000mAh.

Table 15: 24V-6000mAh li-ion battery Requirements and Verifications

Requirement	Verification
1. Can support our motor to use 12 hrs.	1. a. Charge the battery to full capacity. b. Correctly connect the battery to two of our ground terminals. c. Write a control script to continuously drive our motor with full load. d. Record the operating time after the battery is out of charge. Make sure it's greater than 12hrs.

2.4.11 Matched battery charger.

The charger is required to safely charge our li-ion battery within a temperature range of 100°C. the charging voltage is set to 25.2V, and the current does not exceed 2A. As it is a complementary product to the batteries, the parameters are well-adapted and compatible.

Table 16: Battery charger Requirements and Verifications

Requirement	Verification
1. Output 25.2V voltage with a current that not exceed 2A.	1. a. Correctly connect the charge to our battery and connect an oscilloscope in parallel. b. Read out the output voltage and current. Make sure the voltage is 25.2V and the current is within 2A.
2. The temperature is lower than 100°C.	2. a. Charge an empty 24V-6000mAh li-ion battery. b. Continuously recorded the temperature with an electronic thermocouple in the whole charging process. c. Make sure the maximum temperature is within 100°C.

2.4.11 Reach bar.

The reach bar, made of flat steel strips, is designed to support goods. The goods-supporting segment is equipped with vertical limit screws to prevent the goods from tipping over. The two steel strips must ensure that when a force of 5kg is applied at one end, the vertical displacement of the strips does not exceed 3cm, preventing the cargo from tipping over from the limit screw position.

Table 17: Reach bae Requirements and Verifications

Requirement	Verification
1. The vertical displacement of the strips does not exceed 3cm when applying 50N vertical force at one side.	1. a. Fix a bar at one side. b. Applying 25N force at the other side vertically. c. Record the vertical displacement. Make sure it does not exceed 3cm.

2.5 Communication Subsystem

As described in the high-level requirement, the communication subsystem is responsible to let all devices within the delivery know the status of each other. It contains the following subparts: mobile application, cloud server and Raspberry Pi ground terminals and delivery drone.

2.5.1 Mobile Application

Due to the complexity of iOS (developed by Apple™), our mobile application will be first developed and tested on Android™ using Android Studio™.

The application should be able to perform the following operations:

- Select the starting ground terminal and the destination terminal.
- Monitoring the status (mainly the location) of the delivery drone.
- Avoiding duplicate good in the same container of the terminal.

Table 18: Mobile Application Requirements and Verifications

Requirement	Verification
1. Can send order message to ground terminal within 100ms (ideally).	1. a. Send a message using mobile application or using MQTT simulator. b. Using terminal or Raspberry Pi to check whether the message is arrived. c. Using "Device Log" module on Aliyun Cloud Platform to check the latency (explained in 2.5.4).
2. Can display drone location with error less or equal than 5m.	2. a. Let the drone take off to perform a simple delivery. b. Check the mobile phone to see whether the location data is correctly displayed.

3. Can avoid duplicate orders to the same starting or destination container.	3. a. Let the drone take off to perform a simple delivery. b. Send a new delivery request, see whether an error message pop out.

2.5.2 Raspberry Pi in Ground Terminal

Raspberry Pi in our ground terminal would play an important role during our delivery. It should be able to perform the following operations:

- Correctly interpret the operation code from the cloud server and invoking the motor script to open or close the cabinet.
- (Receiver Terminal) Be alerted when the delivery drone is approaching.

Table 19: Terminal Raspberry Pi Requirements and Verifications

Requirement	Verification
1. Can open or close the correct cabinet.	1. a. Turn on the Raspberry Pi in the terminal and connect it to the motors. b. Send a test message using mobile application. c. Using "Device Log" module on Aliyun Cloud Platform to check the message and see whether the correct cabinet is open or closed.
2. Can be alerted destination ground terminal when approaching in 50m.	2. a. Change drone into delivery mode. (not necessarily to take off) b. Put the drone within 50m range of the terminal. c. Using "Device Log" module on Aliyun Cloud Platform to check whether the message is delivered.

2.5.3 Raspberry Pi in Delivery Drone

Raspberry Pi on the drone would be responsible to guide the drone and communicate with the cloud server. It should be able to perform the following operations:

- Determine the path to destination and avoid obstacles using ToF and Pixhawk flight controller. (see 2.3.3 not include in the R&V table in this section)
- Fetch order and report current location (GPS) with cloud server.

- Automatically land on the ground terminal using RTK. (see 2.3.3, not include in the R&V table in this section)
- Load/unload goods by controlling the central motor and cooperating with the container. (see 2.4.8, not include in the R&V table in this section)
- Can alert destination ground terminal when approaching.

Table 20: Drone Raspberry Pi Requirements and Verifications

Requirement	Verification
1. Can send location data to cloud server every 30s.	1. a. Let the drone take off to perform a simple delivery. b. Automatically send location data. (which should be already written in the program) c. Using "Device Log" module on Aliyun Cloud Platform to check the data delivery rate.
2. Can alert destination ground terminal when approaching in 50m.	2. a. Change drone into delivery mode. (not necessarily to take off) b. Put the drone within 50m range of the terminal. c. Using "Device Log" module on Aliyun Cloud Platform to check whether the message is delivered.
3. Can interpret the operation code form cloud server correctly.	3. a. Let the drone be ready to take off. b. Send an order on mobile application. c. See whether the drone go to the right terminal.

2.5.4 Cloud Server

Our cloud server is based on Aliyun Cloud Platform™, it should be a data transferring center to let every device in the system know the status of each other to avoid data collision. It should be able to perform the following operations:

- Handle messages transfer bidirectionally by applying Data Forwarding [2].
- Avoid duplicate message delivery or packet lost (covered in former R&V tables already).
- Ensure high speed of data transfer. Message delivery should be under 100ms giving the condition that the signal is well (covered in former R&V tables already).
- Monitor the status of all devices, check whether messages received or delivered [3] (covered in former R&V tables already).

Table 21: Cloud Server Requirements and Verifications

Requirement	Verification
1. Can handle messages transfer bidirectionally	1. a. Let all devices be online using MQTT protocol. b. Send test message "hello world".

	c. Using “Device Log” module on Aliyun Cloud Platform to check the status of the message.

2.6 Tolerance Analysis

Focusing on research in robotics and UAVs, we understand that precise landing of UAVs is a challenging task being pursued by many teams. In our project, we need to land a UAV with 17-inch propellers on a landing pad with a surface area 1.6 times its projected area. Such precision landing remains challenging in related research projects, and existing teams are adopting various approaches, such as multi-camera visual recognition with graphical computation, optical flow-assisted stable flight, and the ground RTK base station motion computation that we are employing.

The precise landing of UAVs requires its flight controller to obtain accurate kinematic information and adjust the power to keep the aircraft near the target coordinates and attitude. The kinematic computation precision of the HERE3 real-time kinematic (RTK) solution we are using can ensure accuracy within approximately 0.025 meters, sufficient to support high-precision UAV positioning. The flight control system can read high-frequency, high-precision motion information from the module and fuse and correct the IMU data to compute the UAV's motion state. Ideally, the UAV can adjust its current flight attitude and motion state accurately through high-precision kinematic data to remain precisely at the target position and respond to the target status set by the miniPC, achieving precise point landing. However, in actual flight, the UAV's attitude and motion are subject to various external disturbances, such as crosswinds in the environment and vortices generated by the propellers themselves, necessitating the control to resist disturbances along multiple axes. Under the control parameters set during project initiation, the UAV can resist disturbances within a controllable range and stabilize its attitude and position within a certain range. However, in real-world testing, due to environmental variability and structural deformation of the aircraft itself, positioning offsets in the face of interference during landing become noticeably exacerbated.

In similar experimental projects, small quadcopter UAVs ranging from 5 to 7 inches in size experience positioning offsets within a 10 cm square under minimal indoor disturbances during point landing. Experiments indicate that for UAVs of the same configuration, the size of the propellers correlates with wind resistance. Our 17-inch propeller UAV exhibits slightly better resistance to interference than smaller models. However, the weight and moment of inertia of the frame increase with size, and the response rate of control correspondingly slows down. The flexibility and response speed of a 17-inch propeller UAV are slightly inferior to smaller models. For potential landing positioning offsets, we may employ physical guiding structures to automatically move the UAV to the precise position. Depending on the size of the UAV take-off and landing platform, acceptable position offsets are within a range of 20 cm square.

In summary, our UAV structure and flight control are based on conventional design principles and open-source technologies, with most improvements incorporating relatively novel designs. While the final demo confirms the stability and controllability of flight, and while RTK can offer centimeter-level kinematic calculation accuracy, we cannot guarantee that the autonomously designed frame and the control parameters tuned during experimentation will effectively adapt to RTK-assisted positioning for achieving high-precision landing. The final assembly process and debugging optimization can maintain UAV landing positioning offsets within this range. Thus, the functionality of automatic precision landing can be deemed successfully achieved. However, if the final experimental results reveal offsets beyond operable limits, manual intervention may be necessary to position the UAV within operable bounds.

3. Cost and Schedule

3.1 Cost Analysis

We use the average of UIUC EE graduate salary as reference [4], which can be roughly calculated as: 87,769$/250days/8hrs = 44$/hour.

TOTAL Labor Cost = 10weeks * 5days * 3hours/day * 44$/hour *2.5 * 4 person = 66000$

Table 22: Components Cost

Part#	Name	Description	Manufacturer	Quantity	Cost
1	Aluminum Extrusions	The aluminum framework of the container and auto-lifter.	Hang Zhou Jin En Aluminum Industry Co	26 parts (10m length in total)	60$
2	Standard Fasteners	To connect parts into a whole machine.	Hang Zhou Jin En Aluminum Industry Co	N/A	10$
3	57-step motor & DM542 motor driver	To achieve the vertical movement of the auto-lifter	Shen Zheng electromechanical shop	2	50$
4	42-step motor & TB6600 motor driver	To achieve the horizontal movement of reach bar on the platform of the auto-lifter	Shen Zheng electromechanical shop	2	24$
5	BF5M-15 Timing Pulley & 5M Timing belt	The transmission mechanism to achieve the vertical movement of the lift platform	Far East Belt Co., Ltd	2	5$
6	2GT-6 Timing Pulley & 2GT Timing belt	The transmission mechanism to achieve the horizontal movement of the reach bar on the platform.	Far East Belt Co., Ltd	2	4$
7	24V-6000mAh li-ion battery	Power supply for motor and raspberry Pi.	Dong Guan Qi Suo Electronics Co., Ltd	2	20$
8	Power Charger	Charger to charge the power supply.	Dong Guan Qi Suo Electronics Co., Ltd	1	3$
9	Raspberry Pi	Upper computer to receive message from cloud service and conduct python script to move the auto-driver.	Raspberry Pi Foundation	3	160$
10	Slider rail& Slider table	To ensure linearity, limits are set for both vertical and horizontal synchronous belt movement by slider rail.	Su Zhou Hao Cheng Industry Co	4	40$

11	KP08 Bearing	To fix the timing pulley	Japan KIF Company	4	4$
12	Voltage Regulator	input 3.3V from Raspberry Pi GPIOs and output 5V control voltage for motor driver	Raspberry Pi Foundation	**3**	4$
13	Non-standard custom Standard parts	For the T-shaped optical axis that allows the timing pulley to rotate, and the CNC connectors that connect the platform and the timing belt.	Hang Zhou En Da CNC company	2	30$
14	Pixhawk	Flight controller used for the drone	Holybro	1	160$
15	Slamtec Rplidar C1	ToF lidar used on the drone for obstacle detection	SLAMTEC	1	62$
16	Carbon fiber sheet	Skeleton of the drone.	Fusheng Carbon Fiber Industry	N/A	35$
17	Carbon fiber tube		Kuaijie-Jingmi CNC industry	N/A	40$
18	DJI M2006	Motor used to adjust the arm of the drone.	DJI	1	28$
19	DJI C610	ESC cooperates with M2006.		1	15$
20	DJI TB 47 Battery	6S Li-Po Battery with 4700mAh.		2	400$
21	17’’ Propeller	Carbon fiber resin composite propellers for the drone.	Yibang Carbon Fiber Industry	4	18$
22	Motor	Core of the drone power system	SunnySky Motor	4	114$
23	ME909s-821	Module used to provide network connection for the drone	HUAWEI	1	30$
24	HobbyWing ESC 40A	ESC for the drone power motors	HobbyWing	4	45$
25	HDMI Video Capture Card	Display Raspberry OS.	HaGiBis	1	10$
SUM	**66000$ (Labor) + 1371$ (Material) = 67371$**				

3.2 Schedule

Table 23: Detailed Schedule Table

Week	Task	Responsibility
Before	Skeleton of hardware and software connection. Basic algorithm choice of navigation and obstacle avoidance.	Ximo
	Finish CAD of skeleton of UAV and construct the first prototype to verify the assembly.	Yanbing
	Initialization of IoT Cloud Platform. Built code skeleton for both MQTT and Mobile Application.	Yang
	Finish half CAD design for Auto-lifter and goods container	Yuzheng
3/24	**Design Document Due**	**All**
	Simulation setup and raw algorithm verification.	Ximo
	Construct the first version of the UAV and CAD the delivery grabbing structure.	Yanbing
	1-gen for mobile application ***Talon Eat***.	Yang
	Finish CAD design for Auto-lifter and goods container (1-gen).	Yuzheng
3/31	Control of center motor with CAN.	Ximo
	Test flight the drone and start to adjust the controlling parameters.	Yanbing
	Intergrade MQTT code into Raspberry Pis, test on them.	Yang
	Reach out online shops for appropriate materials.	Yuzheng
4/7	GPS-driven auto flight verification.	Ximo
	Build the last version of the drone with full functions required in the tasks. Finish the basic parameter optimization.	Yanbing
	Add more functions into ***Talon Eat***, debugging simultaneously with other subsystems.	Yang
	Build the 1-gen prototype and test its ability.	Yuzheng
4/14	Path planning and obstacle avoidance algorithm programming.	Ximo
	Improve the flight controlling parameters to adapting to flights with different load.	Yanbing
	Synchronize program threads on Raspberry Pi, handle control flow in order. (e.g. Receive data message then call the operation of motor in the terminal or autopilot in drone)	Yang
	Improve the model design based on test results. Build 2-gen prototype.	Yuzheng
4/21	Integration with communication subsystem.	Ximo
	Verify the grabbing structure in the testing tasks and improve the skeleton of drone to have better dynamic characteristics.	Yanbing
	Mock test by combining all subsystems together, debugging if needed.	Yang

	Test & write the motor control script. Test the stability of picking up and dropping off goods.	Yuzheng
4/28	Obstacle avoidance test.	Ximo
	Final optimization of the whole UAV physical design containing the cowling and the shell.	Yanbing
	Finalize communication subsystem.	Yang
	Test in conjunction with other subsystems.	Yuzheng
5/5	**Mock Demo**	**All**
	Algorithm optimization aiming at comprehensive safety.	Ximo
	Test on stability during several flights and check the safety of the structure. Yields the frequency of necessary maintenance.	Yanbing
	Start the draft of final report and debugging.	Yang
	Continues to improve function.	Yuzheng
5/12	**Final Demo & Final Report**	**All**

4. Discussion of Ethics and Safety

Considering about ethic issues, we mainly referred to ACM Code of Ethics and Professional Conduct, “respect privacy” and “honor confidentiality” should be considered. On the one hand, we will focus on the data security of the app and communication to avoid data breach; on the other hand, when collecting data to navigate during the flight, we will ensure that the data only be used to determine the path. Also, due to the characteristics of UAV, we will apply for permission to operate the drones and make sure that it is allowed to fly in certain locations. This will satisfy the term “Maintain high standards of professional competence, conduct, and ethical practice” and “Know and respect existing rules pertaining to professional work.” In the code, “Recognize and take special care of systems that become integrated into the infrastructure of society” should be considered as well. Since our design aims to be used in delivery and may occupy public space, we may have to make rules to ensure that our system won’t be used improperly. For example, we may have to check what is to be delivered before we take the order.[5]

As we are implementing our design, the safety issues need to be concerned. First, it’s essential to implement precise navigation and collision avoidance systems, to avoid collision and falling risks. Even though our design may be tested in campus, obstacles like trees and strong winds may still cause severe accident. Considering that pedestrians are unavoidable, reliable control system that can prevent crashing is required. Also, emergency dealing operation like motor brake will be design to protect passers-by if our drone does crash in populated areas. Second, drones equipped with cameras could capture images of individuals without their consent, which has potential privacy concerns. There are also Federal Regulations for drone operation, which are necessary to consider and follow. “Small Unmanned Aircraft Systems (UAS) Regulations (FAA Part 107) [6]” regulated the detailed rules for the operation of drones weighing less than 55 pounds including guidelines on operator certification, operational limitations, and airspace restrictions. Each drone should be equipped with Remote ID technology. In fact, the rules may differ in different areas. Therefore, we have made basic safety manual based on DJI Flight Safety Guidelines [7] which has considered about the rules above. The first version is as follows which needs to be further developed.

Appendix

Flight Safety Manual

We are always aimed to ensure the safety of anyone involved with our products. So, to help you avoid any mishaps, here are some pre-flight and after-takeoff drone safety tips that can help minimize a lot of potential problems for the experienced pilots and help the beginner's confidence soar.

Flight Environment Selection

1. Fly your drone in an open field far away from densely populated areas, buildings, and facilities with electromagnetic interference.

2. During flight, keep away from GEO Zones or Restricted Zones to avoid forced landing and signal loss due to mistakenly entering these zones.

3. It is not recommended that you fly your drone in scenic spots, historical sites, parks, and the vicinity of government and military institutions. These places are highly likely to be restricted despite not officially being delimited as GEO Zones. Your drone may be forced to land, and flight may even be illegal in serious cases. Before a flight, it is recommended that you check whether GEO Zones or Restricted Zones exist.

4. Do not fly drones near facilities with electromagnetic interference. Common sources of electromagnetic interference include high-power radars, Wi-Fi hotspots, routers, Bluetooth devices, high-voltage power lines, high-voltage transmission stations, mobile phone base stations, and TV broadcast towers. When flying a drone in these areas, the wireless transmission of the drone will be interfered. If the interference is very strong, the drone signal may be stuck or even disconnected.

5. Do not operate the drone in high-temperature or low-temperature environments for a long time. Flying a drone in high-temperature or low-temperature environments may lead to flight accidents.

Pre-flight Preparation

1. Check the remote controller and battery power before the flight. In the case of sufficient power, you can get a longer battery life and better shooting experience.

2. Install the propellers properly and check whether the propellers are damaged or bent and whether propellers can rotate in all directions.

3. Connect your mobile device to the remote controller. Power on the remote controller and then the drone, and fully extend the antennas of the remote controller. Confirm the linking status and check whether any warnings are reported in the app after self-check. Observe and check whether the signal at the take-off location is adequate for image transmission.

References

[1] Wikipedia contributors, "Delivery drone," Wikipedia, Mar. 07, 2024. Available at https://en.wikipedia.org/wiki/Delivery_drone

[2] Aliyun, "Data Forwarding", Aliyun, Mar.24 2024. Available at https://help.aliyun.com/zh/iot/user-guide/data-forwarding-v2/

[3] Aliyun, "Device Log", Aliyun, Mar.24 2024. Available at https://help.aliyun.com/zh/iot/user-guide/log-service/

[4] ECE department of UIUC, "Salary Averages", UIUC. Available at https://ece.illinois.edu/admissions/why-ece/salary-averages

[5] "ACM Code of Ethics and Professional Conduct," Association of Computing Machinery, June 22nd, 2018. Available at https://www.acm.org/code-of-ethics

[6] "Small Unmanned Aircraft Systems (UAS) Regulations (Part 107)" Federal Aviation Administration, October 6, 2020. Available at: https://www.faa.gov/newsroom/small-unmanned-aircraft-systems-uas-regulations-part-107

[7] "Flight Safety Guidelines" DJI. Available at:

https://support.dji.com/help/content?customId=enus03400006768&spaceId=34&re=US&lang=en