Drone Delivery System for Takeaway Business

Final Report

By

Ximo Wang (ximow2)

Yanbing Yang (yanbing7)

Yang Chen (yangc7)

Yuzheng Zhu (yz83)

Final Report for ECE 445, Senior Design, Spring 2024

TA: Yanzhao Gong

5th May 2024

Project No. 7

Abstract

In response to the nascent development of the unmanned aerial vehicle (UAV) delivery market, a foundational small-scale drone logistic system is constructed within a Senior Design Project. The system comprises drones, ground terminals, a mobile application, and a cloud server. Specially designed structure is employed in the dynamic structure of the drones under flexible working condition. Real-time-kinematic navigation is also integrated for high-precision landing during the delivery tasks. Cloud server can publish orders to both drones and automatic containers, also processing feedback from them simultaneously. Specialized mobile application is capable of delivery status monitoring. Innovative functionalities and approaches were explored, yielding valuable data and experiences through testing and debugging processes. This endeavor contributes to the understanding and advancement of UAV-based delivery solutions for emerging markets.

Contents

1. Introduction

We designed and realized an automatic 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. 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. Our solution differs from existing solutions since it's more convenient and cheaper. 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. Simultaneously, we also engaged low-cost standard components for large amount production to reduce expenses.

The system we constructed can be divided into 3 subsystems: delivery drone, ground terminals, and communication system. All these subsystems are connected to each other by communication network and cooperate during the delivery tasks.

The specialized drone integrated innovated structure designed by us for flexible working conditions, which 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. 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 server and mobile APP are specially designed 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 publishing command to them. Figure 1 is a general view of them.

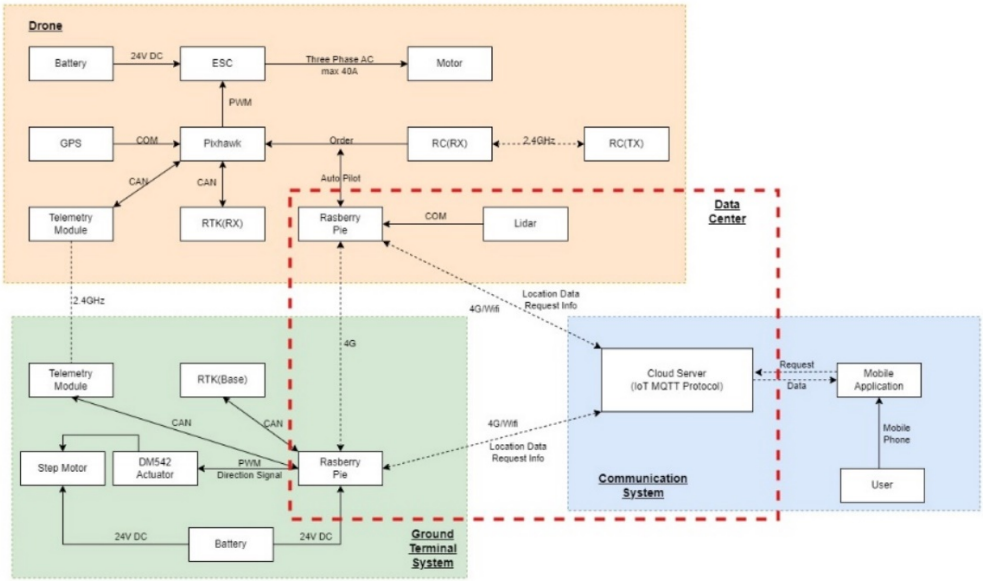


Figure 1: Block Diagram

2. Design

This section is divided into two parts: procedure and details, in which three subsystems (Drone Subsystem, Ground Terminal Subsystem and Communication Subsystem) are described separably.

2.1 Design Procedure

2.1.1 Drone Subsystem

The structure of the drone was specifically designed for the needs of this project by us. Different from the traditional H4 quad-rotor power layout, our design changes the connection between the arm and skeleton from a rigid connection to a hinge connection, so that the drone can adjust the center of gravity in different flight states and flexibly respond to different work scenarios. Simultaneously, the structure for grabbing and releasing delivery is engaged on drone, which can be controlled by the autopilot program to operate the good as expected during tasks.

The main function of the navigation module includes path tracing, message communication and obstacle avoidance. The most basic element is flight control, which is Pixhawk 2.4.8 in our project. The possible choices include DJI, ArduPilot Mega, Speedy Bee etc. The advantage of Pixhawk is that it is an open-source FCU with lots of references on the Internet, which offers both simplicity and extensibility. Naturally, we use Raspberry Pi as the brain of the computer rather than simply STM32. On the one hand, we can use ROS on it to achieve communication between FCU and computer; on the other hand, Linux and GUI interface offers great convenience to code on it. For obstacle avoidance, we chose ToF lidar instead of CV. Since the obstacle avoidance is designed for a drone, fast response and high accuracy is required, which is exactly what ToF good at.

2.1.2 Ground Terminal Subsystem

We plan to design this system as a streamlined communication platform between users and merchants. We enhance efficiency and maximize user service by automating intermediary processes as much as possible. This necessitates an intelligent Ground Terminal subsystem. It will serve not only as a drone landing pad and storage cabinet but also as a picking device (pick up goods and put it down into cabinet), which ensures safe storage of drone-carried goods in designated compartments for easy access by users or merchants. To facilitate communication between the cargo container, the drones, and the cloud service, we use a Raspberry Pi as the ground station's main computer, controlling stepper motors through its GPIO ports' PWM signals. Both STM32 and Arduino are alternative options capable of controlling stepper motors via output ports. However, they require external modules for communication, which presents challenges in debugging and would cause a slower response time. Considering that other subsystems in our project also use Raspberry Pi, the standardization on Raspberry Pi simplifies parts supply and procurement. Thus, it potentially reduces costs due to high volume demand. The cost difference between using microcontrollers with external communication modules, and Raspberry Pi is very small. From an industrial standpoint, microcontrollers' relatively lower reliability also adds to labor and maintenance costs. Therefore, Raspberry Pi is the preferred choice. The subsystem incorporates two linear motion modules, each driven by a timing belt and stepper motor. Our system demands centimeter-level precision without high-speed motion requirements. These requirements make timing belts an economical and convenient choice. Alternatives like gear and rack systems, pneumatic drives,

and screw drives with stepper motors fall short: gear and rack systems require higher motor torque due to greater friction, indirectly increasing costs; pneumatic drives, while ensure an faster motion, do not match the load-bearing performance of timing belt systems and have high maintenance and operational costs due to stringent air tightness requirements; while screw drives offer precise control and higher load capacity, those are over-specifications for this system and come at a significantly unnecessary higher cost. We chose aluminum profiles for the ground station frame for their lightweight, ease of assembly, cost-effectiveness, and structural strength. During assembly, we use standard parts as much as possible, with only one connecting steel part requiring non-standard CNC machining, thus reducing construction and maintenance costs.

2.1.3 Communication Subsystem

In order to let drone and ground terminals to operate correctly, a communication system needs to be established to take the responsibility of data transferring so that the subsystems know the status of each other.

Considering the existing approaches to build a communication system, there are examples like constructing offline servers, Point-to-Point radio communication or employing Mesh Networking. Finally, an online server with MQTT protocol is chosen as it is more suitable for IoT system (low load, low data processing required). Based on that, a mobile application is also developed to send orders to other subsystems and monitor drone status. Raspberry Pis on other subsystem receive the operation code and perform the desired operations. The cloud server is responsible for processing the messages and distribute it to the correct clients.

2.2 Design Details

2.2.1 Drone Subsystem

2.2.1.1 Mechanical Design of Drone

The quadrotor UAV skeleton employed in this project is specially designed for its working conditions. The specialized skeleton is individually designed and manufactured under the joint consultation of our members. The structure of the drone is improved based on the traditional H4 quadrotor drone (Figure 2). Two T-shaped arms are connected to the body by hinges instead of rigid connection. On the premise of ensuring flight stability under variable condition, functions such as transferring center of gravity and adjusting lift configuration can be realized with the hinge. The two forms of the drone are the auxiliary precise landing scene and the high-speed flight scene (Figure3). The former needs high flexibility and the latter needs high efficiency and stability.

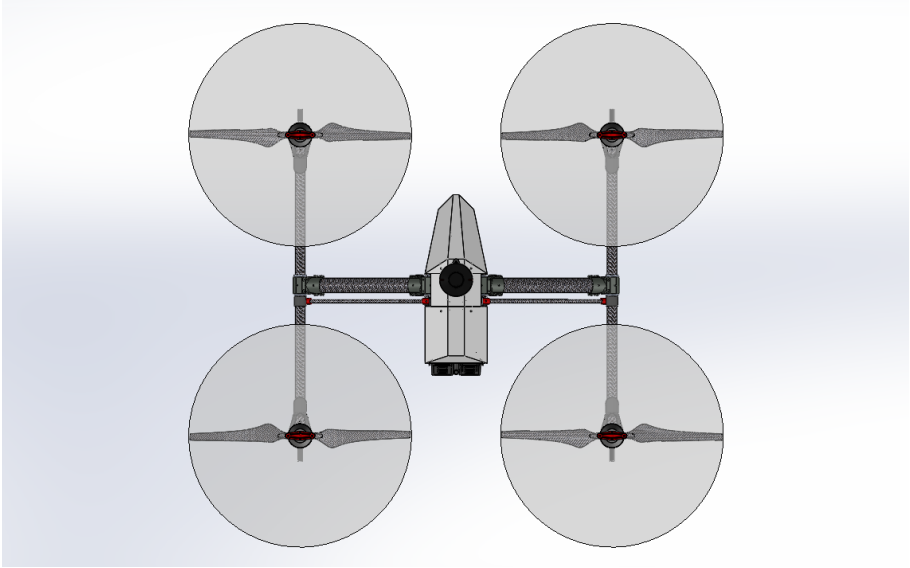


Figure 2 Improved layout based on traditional H4 quadrotor

Figure 3 Transition between two forms

The transformation between the two forms is controlled by a set of four-bar links driven by a screw rod. This enables controlled rotation of the host arm around the rotational axis of the main airframe within a specific angle range, while maintaining optimal connection strength and stiffness in all other directions. Simultaneously, to ensure that there is no deviation in rotor thrust direction, it is essential for the plane where the secondary arm is located to remain parallel to that of the main airframe. Achieving this effect necessitates employing a set of parallelograms connecting rods (Figure 4), thereby requiring design

considerations for a connecting rod aligned parallel with respect to each main arm. Consequently, this connection establishes a four-link mechanism between the main airframe and secondary arm, ensuring consistent angles among their diagonals.

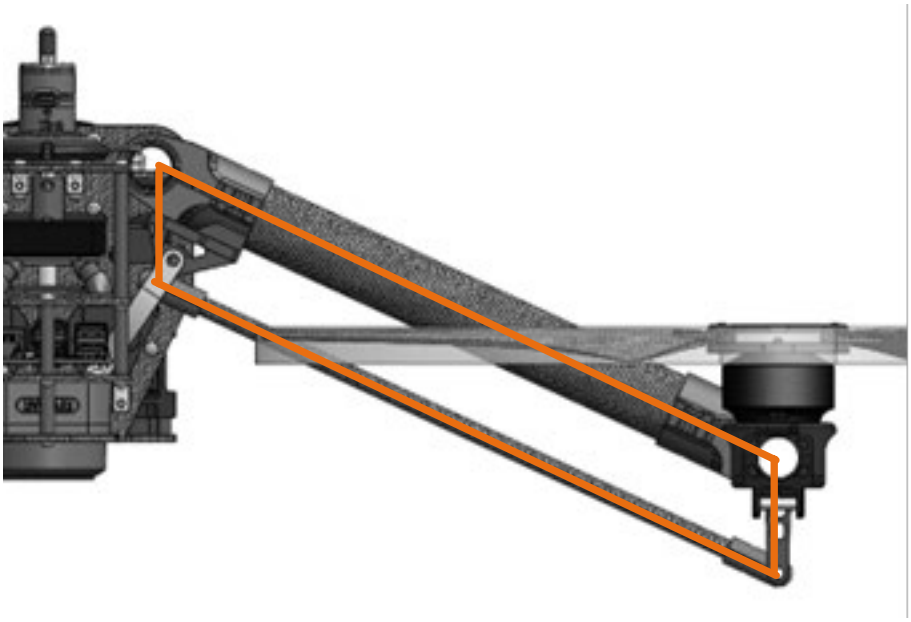


Figure 4 Parallelogram links to keep parallel thrust

The most challenging difficulty during the skeleton design is the innovative hinge. Due to the special layout of the power and center of mass, the hinge on the joint is expected to work smoothly under heavy load and high bearing moment. Simultaneously, the hinge is part of the UAV, which is expected to be light and small-scaled. The initial idea was aluminum alloy parts with graphite as lubricant. The practical test exposed the disadvantage of this structure, where the graphite's thickness of 0.5mm caused the diastema of the hinge about axis other than rotation. The diastema resulted in approximately 2cm waggle at the end of the skeleton arm, which is harmful for the rigidity and reliability during flight.

After trying to use graphite as the lubrication layer failed, we tried to directly contact the aluminum alloy parts with carbon fiber resin tubes. The connected diastema has been successfully solved and the loosening range of skeleton arm has been reduced. However, the rotational friction coefficient of hinge is relatively higher, serious scratch and wear shown on the resin surface. Scratches and dust appeared after a few tests. Experiments have proved that the lubrication effect of graphite is essential, but the thickness of the graphite layer causes loosening. In order to solve the problem, we tried using a layer of resin nesting between the aluminum alloy and carbon fiber resin tubes (Figure 5). To reduce the friction and wear, graphite debris were applied to increase the lubrication effect. After more than 8 hours of flight testing, the hinge joints can withstand the load under normal conditions and achieve smooth rotation eventually.

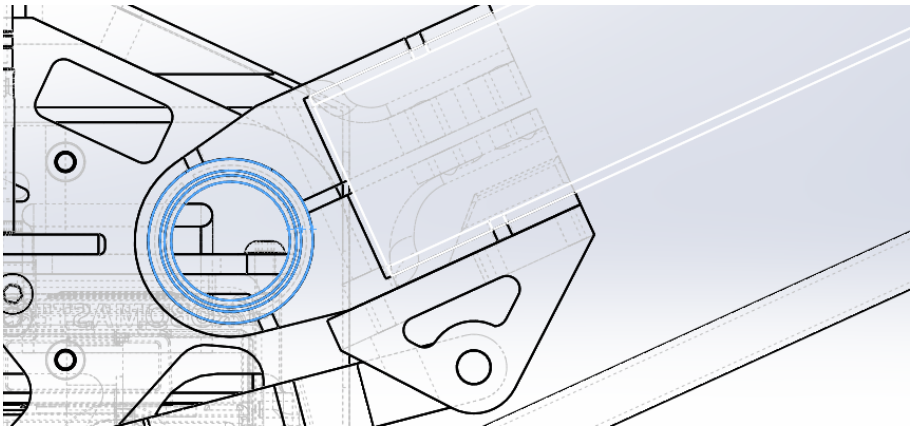


Figure 5 Multilayer Hinge with lubrication

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 have been tested and adjusted for several times to reach the optimal performance. The performance was not ideal when we use 2kg gas springs at first, where the gas springs were overloaded under normal condition. Simply increasing the restoring force resulted in the failure of absorbing shock, which shows no response to the impulse during landing. Finally, we find an optimal solution by replacing the spring with a set of suspension with adjustable restoring force. The shock and impulse from landing support was absorbed as expected eventually.

The load-bearing structure of the body is 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. To optimize the weight control of the drone, most parts' weight is reduced ensuring strong enough to withstand the working load in FEA simulation.

2.2.1.2 Dynamics Analysis of Drone

To complete the transportation task in this project, the design index of the UAV is to mount the cargo of more than 2kg, fly 5km independently at a speed of 40km/h, and accurately land to the destination. Based on the parameters of most of the same type of cargo payload drones on the market and combined with the design experience of multi-rotor aircraft, we set the no-load take-off weight of the UAV within 4kg. No-load endurance of more than 20 minutes; The maximum take-off weight is less than 6kg, the full load endurance is more than 8 minutes, and the cargo can be transported to any destination within 5km in a single flight at the maximum load and maximum flight speed. Therefore, we chose the 340KV 4110 brushless motor with 1760 blades of 17 inches as the power of the UAV. The maximum thrust of a single power group is 2.1kg and working together can provide the maximum thrust of 8.4kg for the UAV. According to experience analysis, the reasonable maximum take-off weight is about 6kg, which meets the task requirements of the project.

The design index of the UAV in this project is that it can load 2kg of cargo in a single flight and transport it to the destination within 5km within 8min. We can get from simple energy calculation, in order to make the endurance of the UAV under the maximum load and maximum flight speed reach 8min. It needs Li-Po lithium batteries with a capacity of at least 9000mAh 6S voltage to power it, so I carried two

4700mAh lithium batteries on the drone. Some data provided by the power equipment manufacturer are used in the calculation, and some other relevant parameters are listed in the following table. (Table ?)

Table 1 Parameters of power equipment on 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)
Power at max thrust	1600W (400W *4)
Take-off weight (TOW)	4.5 kg
Endurance at TOW	22 min
Max TOW	7 kg
Endurance at Max TOW	8 min
Max diagonal size	1330 mm
Height	360 mm

2.2.1.3 Flight Control and MAVROS based Autopilot

The flight control model is the traditional H4 four-rotor UAV. Four rotors in the same plane rotate in a specific direction to provide thrust. By combining and adjusting the thrust size of the four rotors, 6 degrees of freedom controlling is realized. Our UAV is equipped with Pixhawk series flight control hardware and burned the open-sourced Ardupilot flight control firmware, which is sufficient to complete the tasks of unmanned cruise flight and cargo grabbing and releasing in the project. By counting the mass and position distribution of all parts, we can obtain the dynamic parameters of the whole drone, and plug them into the calculation of acceleration and angular acceleration to obtain appropriate flight control parameters. With the correct installation and setting, the UAV performed normally in the test flight under basically reasonable control parameters and could respond to simple pilot remote control commands. After ensuring the safety and reliability of flight, in order to solve the optimal combination of PID parameters of flight control, we learn and implement the Auto-Tune function of Ardupilot according to the process. During the flights and tunning, we finally get the optimal solution of the parameters of the UAV under different working conditions, so that the attitude response and dynamic efficiency of the UAV reach the best state.

The communication protocol that Pixhawk uses to connect with ground station is MAVLink. MAVLink messages can convey various information such as telemetry data, control commands, and system status. To implement navigation program, C++ and Python is needed with ROS. Thus, MAVROS is introduced as a bridge between code and FCU. MAVROS allows ROS nodes to communicate with MAVLink-enabled drones. MAVROS provides ROS APIs for controlling and monitoring the vehicle's state, sending commands, receiving telemetry data, and more.

In our navigation program, to fetch the GPS coordinates of drone, we subscribed to MAVROS topic named "/mavros/global_position/global". Another MAVROS topic "/mavros/setpoint_position/global" is

used to set the waypoint by publishing the coordinates wanted. In this way, we have realized the position fetching and waypoint setting. The last problem is we need to set the yaw angle, which is also "heading" in our code. Since ENU is used, we can calculate the heading as:

$$angle = arctan\frac{latitude_{target} - latitude\_current}{longtitude_{target} - longtitude_{current}}$$

$$heading = \begin{cases} angle, & positive\ angle \\ angle + 360, & negative\ \text{angle} \end{cases}$$

The flowchart of navigation system is shown in Figure 6:

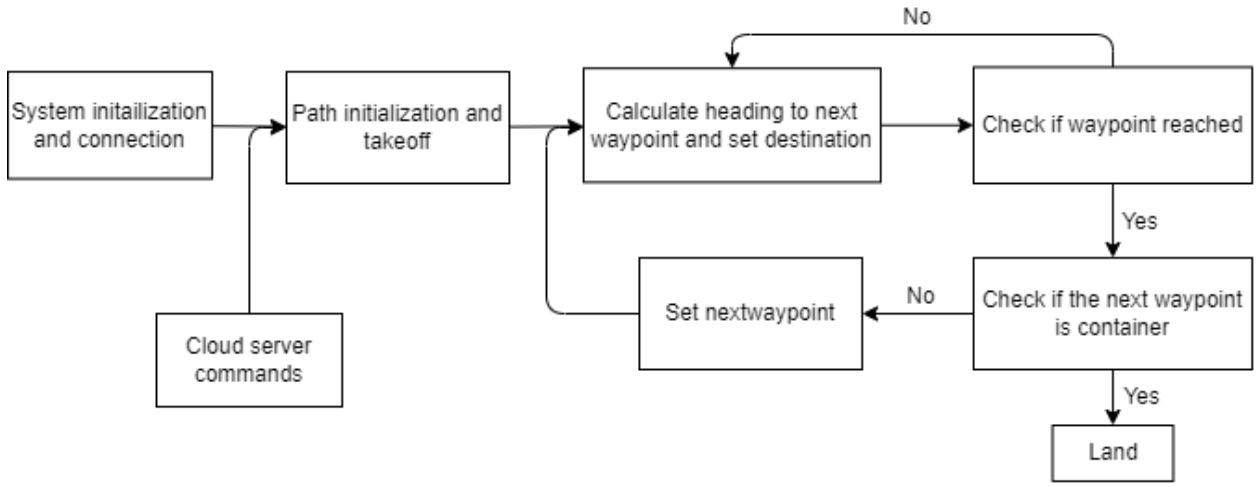


Figure 6 flowchart of navigation system

2.2.2 Ground Terminal Subsystem

2.2.2.1 Mechanical Design and Dynamics Analysis of Ground Terminal

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. The 2020A section aluminum's yielding strength is around 200MPa[1]. Considering the following formula for maximum load of material:

$$P = 0.44 \times (D + t) \times L \times \sigma b \quad (1)$$

Here P is the maximum force, D is the section diameter, t is the thickness, L is the length and is maximal yielding strength. For our supporting section aluminum, L = 1200mm, t = 2mm, D = 50mm. The final P is around 550 KG. Since the platform self-weight is only 5kg, our load capacity is far greater than 3KG.

This subsystem incorporates a vertical linear motion mechanism to control z-axis movement and a horizontal movement mechanism for the x-axis on the platform. A BF5M-15 Timing Pulley & 5M Timing

belt have been selected to manage the lift platform's vertical displacement. This setup, including a 57-step motor adequate for a 100N vertical load as per the manual, involves a 5M-15 tooth pulley. Connection is achieved via a clamping plate and screws, linking the load to the motor. At the lift framework's base, a 57-step motor operates with a DM542 2H Microstep Driver. This system, powered by a 4V-6000mAh li-ion battery, receives direction and PWM pulse signals from a Raspberry Pi to control the motor's operation per the Pi's script.

Figure 7 2-gen ground-based terminal

Since the whole weight of platform is not larger than 10KG, we should choose our step motor based on that requirement. The diameter of BF5M-15 tooth pulley is 22.7mm. The mechanical loss coefficient is 0.6. by formula:

$$T = \mu * r * F \tag{2}$$

The torque would be: 0.72 N*m. With 1A current the 57-step motor could provide 1 N*m torque, while our driver could set maximal current to 3A, which ensures the success of our required load capacity.

For horizontal actions, a 2GT-6 Timing Pulley & 2GT Timing belt ensures the Reaching bar's lateral motion on the platform. Coupled with a 42-step motor, this assembly, as the manual suggests, supports a 20N horizontal force with a 2GT-6 tooth pulley. The motor, mounted on the platform and energized by an identical battery, is driven by a TB6600 2H Microstep Driver. It collaborates with the belt-pulley

system for outward movement to fetch items. Commanded by the Raspberry Pi, the motor executes the programmed actions with the driver supplying the necessary energy.

For a 3KG load on platform, the friction coefficient could be 0.6 (for cast steel material). The diameter of 2GT tooth pulley is 5mm. Based on formula (2), the torque could be 0.05 N*m. With 1A current the 42-step motor could provide 0.5 N*m torque, which is far over our requirement. Since the TB6600 2H Microstep Diver could provide maximal 3A current, which ensures the success of our design.

A Raspberry Pi is set to control the motor based on python script, which enables us to achieve the designed motion in order and control the auto-lifter to pick up and put down the goods. A 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 5hrs, nearly 5000mAh capacity is required. Considering the capacity loss after long-term use, we set a standard of 6000mAh. 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.

2.2.2.2 Dynamics Control of Ground Terminal

The control diagram of the step motor is as shown in Figure 8. The step motor microstep driver receives PWM signals to drive the motor. The physical dip switches on the driver allow for adjustment of the number of pulses required for each full rotation of the motor, thus controlling the output torque and operating speed of the motor under the same power conditions. PWM signals and directional high/low level signals can be obtained through the GPIO ports of a Raspberry Pi. In the Python environment, scripts written using the RPi.GPIO library enable the control of the stepper motor according to the required motion pattern.

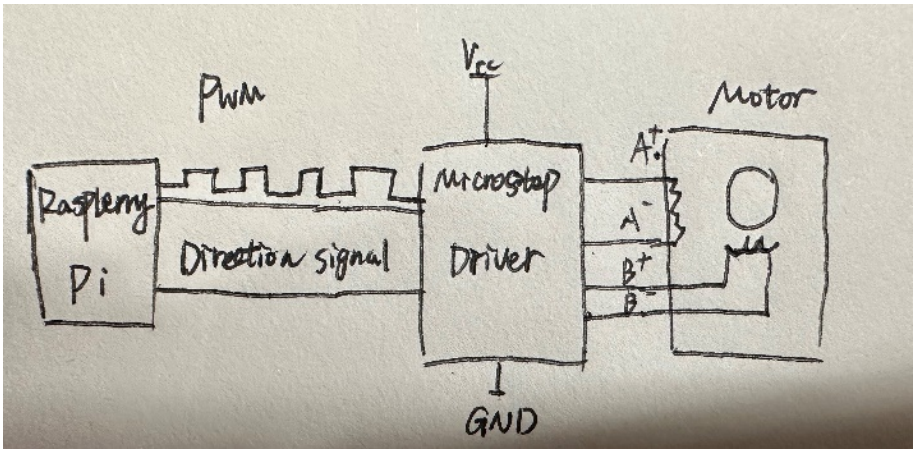


Figure 8 Control Diagram of Terminal

2.2.2.3 Communication with Drone and Cloud Service

Raspberry Pi receives commands from the drone and cloud service, and runs the script accordingly. The detailed content would be explained in the following paragraph.

2.2.3 Communication Subsystem

The complete communication subsystem can be separated into four parts: Raspberry Pi in drone, Raspberry in ground terminal, mobile application and cloud server. While Raspberry Pis in drone and terminal have many other functions in other subsystems, only communication part would be introduced in this sub-section.

2.2.3.1 Raspberry Pi in Delivery Drone

As for part of the communication system, Raspberry Pi on the drone would be responsible to tell the drone the starting point and the destination point of the delivery so that the drone can correctly pick up and drop the goods. It should receive the operation code from the cloud server and interpret it into the correct operation. Also, when approaching the departure terminal or destination terminal, a message should be sent so that the terminals can open the container in time.

During the progress of the delivery, location (GPS) data, velocity and other status data would be processed and send to cloud server according to a preset time interval (5s).

The main protocol we are using is MQTT protocol. A short introduction to it is attached in to Appendix B

Since cellular network is used on our delivery drone, a communication module is implemented, which is HUAWEI ME909s, enabling the drone to be connected to the network almost everywhere. A short introduction is also in Appendix B.

2.2.3.2 Raspberry Pi in Ground Terminals

Raspberry Pi in our ground terminal would play an important role during our delivery. Similar to the Raspberry Pi in the delivery drone, it also needs to communicate with cloud platform to know which terminal is the sender and which terminal is the receiver. However, since it is a “terminal”, it is simpler as we do not necessarily need an LTE module for communication. Most time a Wi-Fi connection would fit.

The design for the functions of these Raspberry Pis are the followings:

- Receive operation from the cloud platform, invoking active mode.
- Correctly interpret operation code and determine the departure and destination terminals.
- Be ready to receive signal from drone to open container.
- Invoke the script for motor running to open container.

2.2.3.3 Cloud Platform

The cloud server is developed 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.

In recent years, Aliyun has expanded its offerings to include artificial intelligence (AI), machine learning, Internet of Things (IoT), and blockchain services, positioning itself as a leading cloud provider driving digital transformation and innovation across various industries which gives us a motivation to apply it as our cloud platform.

By creating devices in the platform and implement correct code in mobile application (java code) and Raspberry Pis (python code), a simple MQTT based data transfer can be established. Figure 9 shows our created devices.

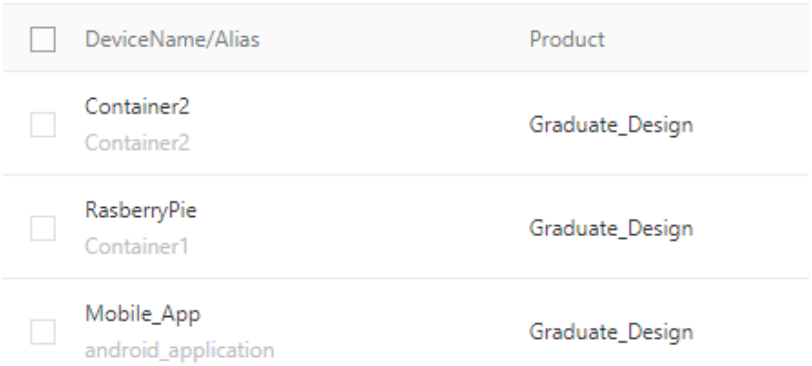


Figure 9: Examples of Our Devices

However, due to privacy issues, Aliyun has a service called “Data Forwarding”. For example, Mobile application cannot subscribe topic belonging to other devices, say Container1. In this case, we need to forward the data from container1 to mobile application. Figure 10 shows an example of Data Forwarding.

Rule Name	Rule ID	Data Type	Rule Description
Container1ToApp	1771447	JSON	Container1ToApp
AppToContainer1	1771439	JSON	AppToContainer1

Figure 10: Example of Our Data Forwarding Methods

Aliyun Cloud Platform also has a function named “Device Log” to allow users to trace the published message. In this case, during verification we can quickly locate the bug to see the problem is on which device.

A pseudocode with flow chart is attached in Appendix B to explain how our MQTT code works with cloud platform.

2.2.3.4 Mobile Application

The mobile application, “***Talon Eat***” is also a crucial part of the communication system. It intergrades functions of sending requests, receiving other device information, displaying information together into one app. Figure 11 shows the application icon developed by Yanbing.

Figure 11: Application Icon by Yanbing Yang

Due to the complexity of iOS (developed by Apple™), *Talon Eat* is first developed and tested on Android™ using Android Studio™. BaiduMap API is used to display map on the application, a brief introduction to it is attached in Appendix B as well.

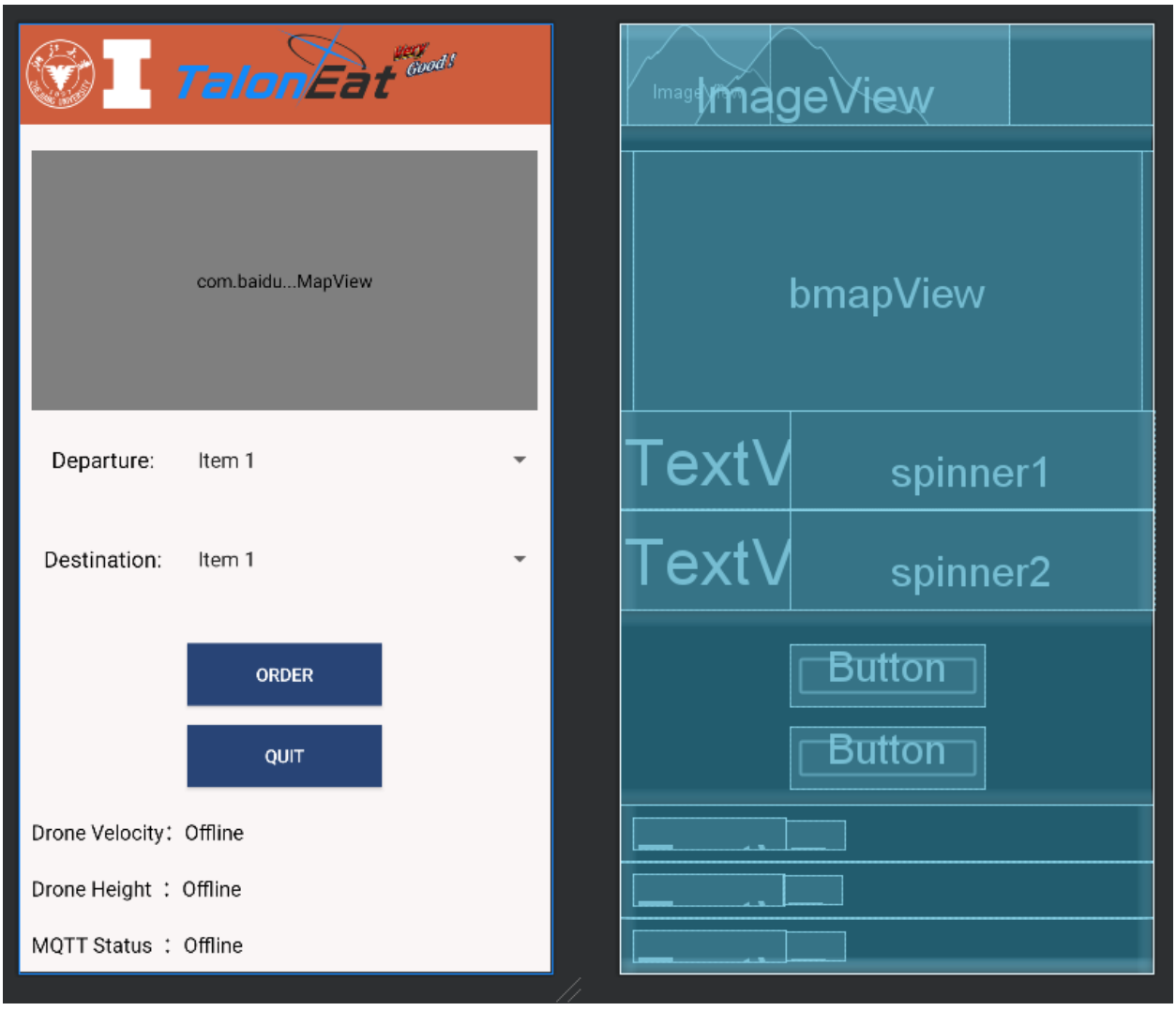


Figure 12: Layout Design of *Talon Eat*

Figure 12 above is the general layout design of our mobile application *Talon Eat* (before compilation).

- Title is our application name and the format may be changed later.
- bmapView is map display area applying Baidu Map API.

- Departure and Destination points are chosen by users, they can choose from the spinners, listing all available starting and destination points.
- Two buttons control the functions of sending requests and disconnecting and quitting the application. Duplicate requests should be avoided.
- Other status of the drone (velocity or height etc.).
- Display drone track on the map.
- Other buttons to differ from “Send Message” to help users better choose the destination.

3. Verification

During verification and experimental tests of all the subsystems, we have gone through approximately three steps: designing and simulation, manufacture and assembling, practical test and debugging.

Firstly, the drone subsystem was CAD-ed and simulated to verify the mechanical properties. After the low-cost prototype assembly test, the initial version drone was built to carry all the equipment containing the flight control module and Raspberry Pie. Ground based terminal was also designed with mostly standard materials and servo motors by CAD. The required function of cloud server and mobile application were also started to be analyzed and constructed.

Secondly, basic flying ability of drone is ensured by debugging on the firmware and tuning the parameters of dynamic control system under manually flight tests. Autopilot executed by the Raspberry Pie was integrated on drone and tested in practical flights several times after simulated flight verification in ROS. The mobile application "***Talon Eat***" is first developed and tested in Android Studio with both virtual and actual devices. Cloud server is also first tested with MQTTX. Prototype the ground-based terminal is built to verify the structure and dynamics properties. Several mechanism and software errors are discovered and collected to feedback to each subsystem for improvement.

Thirdly, each subsystem was verified the independent function under practical tests and debugging. Complete function of the drone's structure was realized by integrating the specialized autopilot program. Designated flight task could be fully automatically conducted by the drone with high-precision navigation. Communication subsystem is verified to enable all subsystems to know the status of each other, ensuring all messages are correctly delivered. We also verified each part of the ground terminal, which will be shown in following content.

3.1 Verification of Drone Subsystem

After completing the theoretical calculation and CAD modeling, the initial step is to conduct computer-aided finite element analysis simulation testing in order to verify whether the structural strength in the design is sufficient to withstand heavy loads and enhance the structure by addressing material failure and risk factors. Subsequently, we proceed with constructing the first-generation assembly verification and structural function testing machines, ensuring their compatibility with power equipment through circuit wiring design while identifying any design errors or deficiencies. Currently, we have accomplished finite element analysis and assembly verification, and are presently engaged in flight testing and debugging of the testing machine which evaluates both structural strength rationality and dynamic aircraft layout. Throughout this process, it is crucial to continuously document shortcomings and loopholes within our design, constantly supplementing and improving them until achieving optimal performance for the final aircraft.

After completing a certain degree of CAD modeling, finite element analysis on some of the important structural parts was performed, which is the lowest cost simulation test before production. Through the calculation and analysis of the stress and impact generated in the working scene of the UAV, the stress or torque of most parts in the frame can be obtained. By setting the constraint conditions and load conditions of the response as well as the nature of the material, we can carry out FEA simulation test of

each part to ensure that the strength of the part is enough to support it to complete the task requirements of the project. If the part shows too much strength redundancy in the simulation test, the topology optimization built into the CAD software can optimize the part to ensure the appropriate weight loss under the strength (Figure 13). Some of the parts fail and almost break in FEA. According to the result distribution obtained by simulation, the structure of the parts can be adjusted or stronger materials can be selected (Figure 14).

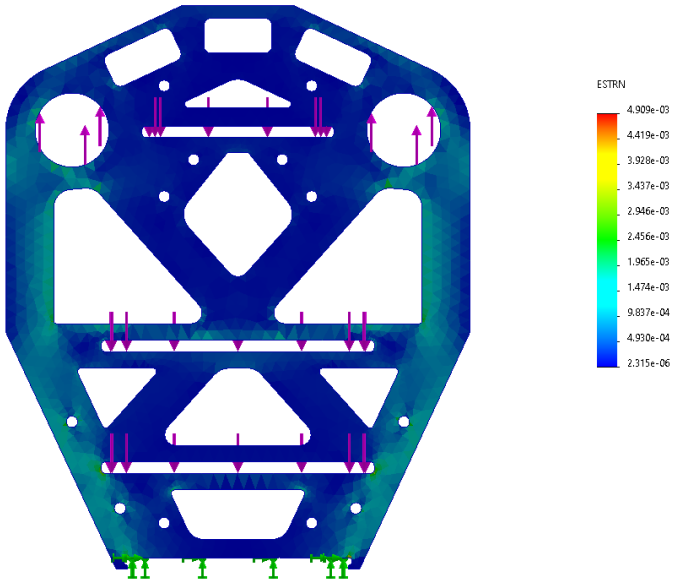


Figure 13 Topological optimized board after weight reduction

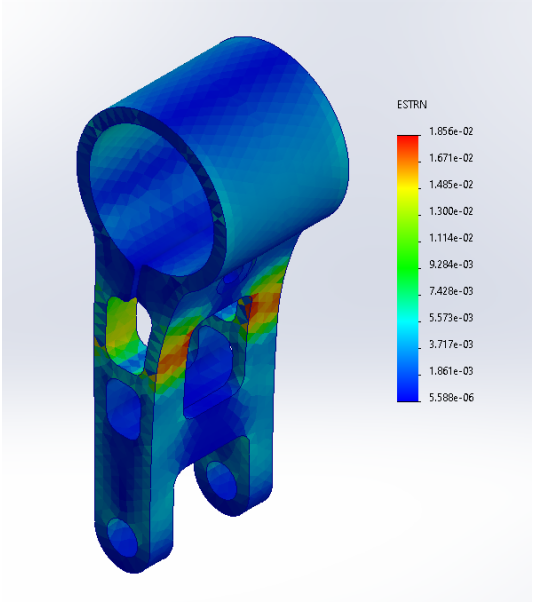


Figure 14 Parts required improving due to understrength

After the successful simulation test, we manufactured some of the key parts with lower cost processing technology, and carried out verification assembly with the purchased standard parts and functional modules to check the structural interference and logic errors in the assembly process. The model can be further modified and refined according to the records in the assembly process. After the verification of the assembly and the adjustment of the model, the layout design and testing of the electronic equipment can be carried out, and these steps are carried out by our team members in a standardized and safe operation and successfully tested.

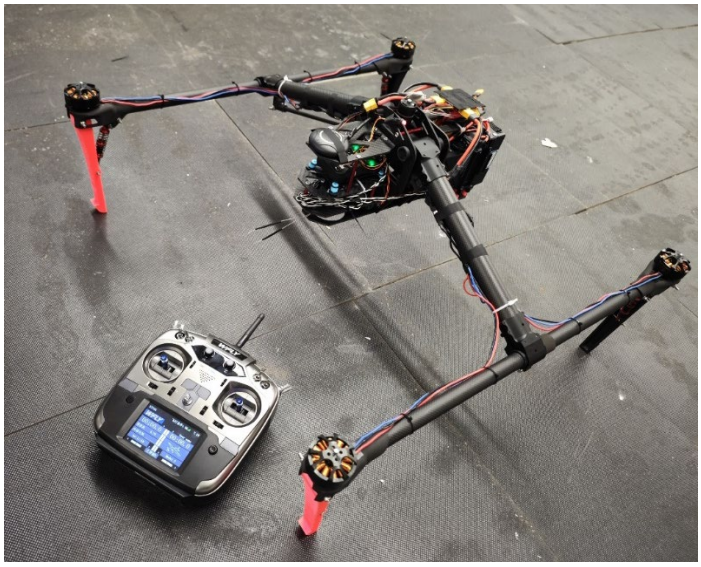


Figure 15 First prototype of the drone with electrical equipment online

After confirming the rationality of the design, we built the first version of the testing machine, and most of the parts were processed according to the final version. After the testing machine was built, we could start the testing of the functional structure, such as the attitude conversion of the UAV and the working conditions of the functional module. After basically confirming the stability and reliability of the structure, the electronic equipment needed were installed and connected on the UAV. After flight control configuration and parameters setting up, we conducted a test flight ensuring the absolute safety of personnel, verifying the structural strength, power configuration, flight duration and other indicators of the UAV. The test was successfully controlled, and valuable details are recorded for analysis and improvement. By the help of autotune function built in Ardupilot, we adjust the PID parameters of the drone by tens of trial. Since the action of autopilot could be aggressive comparing with human, we adjusted the PID little bit milder to ensure safety.

Figure 16 First test flight of the drone without load

One more thing we need to do is to verify the autopilot program in simulation. Same as the real case, ROS is used in simulation as well. The ROS package we used is an open source one called iq_sim. This package contains gazebo environments representing different drone scenarios and configurations. It's tailored to be compatible with the Ardupilot control system, leveraging the Ardupilot gazebo plugin to enable seamless interaction between the Ardupilot control software and simulated drones in gazebo. Gazebo is a widely used open-source robotics simulator. It provides a virtual environment where users can simulate and test robotic systems, including drones. Gazebo allows users to create detailed 3D models of robots and their environments, simulate physics-based interactions, and run control algorithms in a simulated environment (Figure 17). All the functions including path tracing, obstacle avoidance and communication with server are tested before implementing on drone.

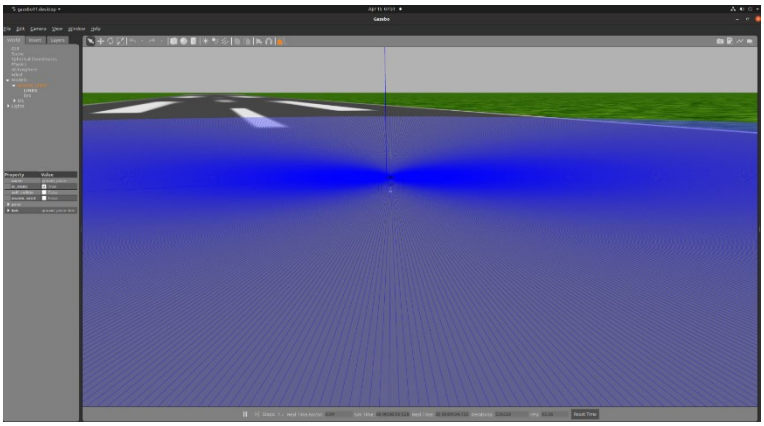


Figure 17 Interface of Simulation Program

Making sure both the drone and program are reliable, we tested the autopilot on our drone. At the very beginning, only GPS and pose reporting were tested to ensure the communication between FCU and Raspberry Pi is stable. Then, arming, takeoff and landing commands were tried. A few parameters, such as the climbing rate, were adjusted based on observations to improve stability and efficiency. Finally, path tracing, heading and altitude decision were verified. It is important that the simulation and real flight were conducted in between to solve the problem observed.

3.2 Verification of Ground Terminal Subsystem

Regarding all aforementioned parts, we set verification procedures for each. All of them could be found in the tables in Appendix. The following content in this paragraph should be read accompanied by the according tables.

During prototype experiment, the section aluminum keeps stable while setting 25KG, which has already satisfied 2.5 safety factor.

During test, the BF5M-15 belt section keeps stable while setting 300N, which has already satisfied 3 safety factors. The 2GT-6 belt section keeps stable while setting 50N, which has already satisfied 3 safety factors.

By testing, for 57-step motor the minimum advance distance is around 0.37mm, which satisfies the minimum requirement of 1cm. what's more, the maximum torque is around 2.73 N*m, which satisfy our requirement.

For 42-step motor, the minimum advance distance is around 0.078mm, which satisfies the minimum requirement of 1cm. what's more, the maximum torque is around 1.28 N*m, which satisfy our requirement.

Both pulley-belt system could be driven by Raspberry Pi if successfully connected.

Inspecting oscilloscope, the output voltage of voltage regulator is accurate 5V.

With 1A current load, the Ground Terminal battery takes 7hours to be out of charge, which satisfies the requirement.

With 25N load, the vertical displacement of reach bar is 1.88cm, which satisfies the requirement.

3.3 Verification of Communication subsystem

According the Requirement & Verification Table, the four parts of communication subsystem can be verified separately.

Python code scripts are implemented on both Raspberry Pis in drone and ground terminals since the code skeleton of them are similar. The verification is first using MQTTX before the mobile application was developed. The software MQTTX was used to play the role of mobile application and send orders to Raspberry Pis. With the help of the "monitoring" function on cloud platform. As figure 18 shows, the status including the latency can be shown in the cloud platform.

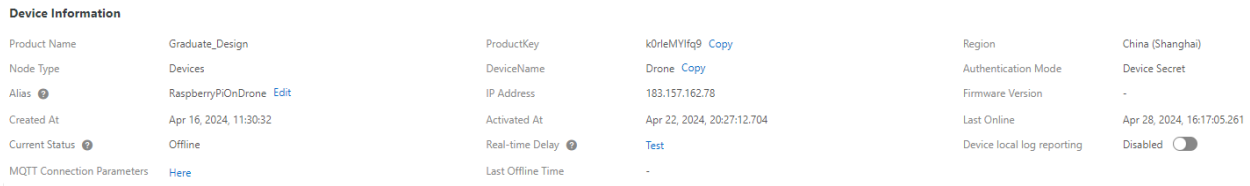


Figure 18: Status of Device

Mobile application can also be verified independently. The first test was also using MQTTX to ensure the correct connection is established. After the completion of other subsystems, the mobile application takes the responsibility of MQTTX and the status table in Figure 18 is also made use again.

The verification between application and ground terminals is achieved by sending an order on the phone then monitor the action of the cabinet of the terminals. More detail is in the R&V table.

The verification between mobile application and delivery drone is also achieved by sending an order on the phone the monitor whether the status data on the phone is correctly updated. Figure 19 shows the correct update of the location of the drone.

Figure 19: Drone Location Data

The last verification is quite simple, send the order on the phone, see whether the correct cabinet is open, whether the drone goes to the right starting terminal and whether it is guided to the right destination terminal. If so, given all other subsystems are verified, then the communication subsystem is verified too.

4. Costs

4.1 Parts

The parts we have purchased were all sold at retail price, so the column of Bulk Purchase Price is replaced by part quantity.

Table 2 Parts Costs

Part	Manufacturer	Retail Cost ($)	Quantity	Actual Cost ($)
Aluminum Extrusions	Hang Zhou Jin En Aluminum Industry Co	$6	26 parts (10m length in total)	$60
Standard Fasteners	Hang Zhou Jin En Aluminum Industry Co	$10	N/A	$10
57-step motor & DM542 motor driver	Shen Zheng electromechanical shop	$25	2	$50
42-step motor & TB6600 motor driver	Shen Zheng electromechanical shop	$12	2	$24
BF5M-15 Timing Pulley & 5M Timing belt	Far East Belt Co., Ltd	$2.5	2	$5
2GT-6 Timing Pulley & 2GT Timing belt	Far East Belt Co., Ltd	$2	2	$4
24V-6000mAh li-ion battery	Dong Guan Qi Suo Electronics Co., Ltd	$10	2	$20
Power Charger	Dong Guan Qi Suo Electronics Co., Ltd	$3	1	$3
Raspberry Pi	Raspberry Pi Foundation	$54	3	$160
Slider rail& Slider table	Su Zhou Hao Cheng Industry Co	$10	4	$40
KP08 Bearing	Japan KIF Company	$1	4	$4
Voltage Regulator	Raspberry Pi Foundation	$2	2	$4
Non-standard custom Standard parts	Hang Zhou En Da CNC company	$15	2	$30
Pixhawk	Holybro	$160	1	$160
Slamtec Rplidar C1	SLAMTEC	$62	1	$62
Carbon fiber sheet	Fusheng Carbon Fiber Industry	$35	N/A	$35
Carbon fiber tube	Kuaijie-Jingmi CNC industry	$40	N/A	$40
DJI M2006	DJI	$28	1	$28
DJI C610		$15	1	$15
DJI TB 47 Battery		$400	2	$400
17'' Propeller	Yibang Carbon Fiber Industry	$9	4	$18
Motor	SunnySky Motor	$28.5	4	$114
ME909s-821	HUAWEI	$30	1	$30
HobbyWing ESC 40A	HobbyWing	$11.25	4	$45

HDMI Video Capture Card	HaGiBis	$10	1	$10
Here3 RTK Navigation Module with Base	Cube Pilot	$378	1	$378

4.2 Labor

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

5. Conclusion

5.1 Accomplishments

The full function of the drone has been verified available to achieve the task in project. Mechanical structures are tested during several hours flight and fragile ones are improved and replaced.

By now, we have achieved the most of objects for the drone, includes autopilot based on server command, elementary obstacle avoidance, and flight information feedback.

Moreover, all the objectives of the Ground Terminal subsystem have been successfully achieved, including picking up and putting down the goods from cabinet to the drone landing pad, communicating with drone and cloud service, and executing the according motor motions based on commands.

The objectives of the Communication Subsystem have also been achieved. Now mobile application can control the operation of other subsystems and let them work properly.

5.2 Uncertainties

In the aspect of mechanical structure of the drone, the diastema of the skeleton is always the unavoidable but potentially lethal. The skeleton's manufacturing process can still be improved with higher precision and rigidity. Simultaneously, the experimental results show the necessity of laying the center of mass at the geometric center, which can reduce the burden on flight control and motors.

In our proposal, the mounting should be automatic as well, which is involved with both landing accuracy and form transition. Considering the time limits, we simplified these features without affecting the functionality of the whole system.

For the ground terminal, due to the external boost module and battery transformer module, the connections between cables are relatively complex. This increases maintenance costs to some extent. For vertical slide rails, the pressure between the guide rail and the slider is greater when moving upwards, which increases the risk of the machine stopping due to a higher friction coefficient between the slider and the rail after prolonged use.

For the communication system, the current problem is for the mobile application ***Talon Eat***, the MQTT connection may be lost due to undetermined issue. Also, data collision may also happen if multiple users are sending the same order. A potential solution may be adding log in data base system as mentioned in 5.4.

5.3 Ethical considerations

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.

Addressing the IEEE Code of Ethics, which stipulates the paramountcy of safeguarding human safety, my design and operational practices reflect a commitment to this principle. In particular, the IEEE Code of Ethics urges us to "avoid real or perceived conflicts of interest whenever possible, and to disclose them to affected parties when they do exist"[8]. It also emphasizes the need to "reject bribery in all its forms"[9]. In alignment with this guidance, I have installed protective plastic pads on all sharp-edged aluminum profiles to prevent accidental injury. The two sets of operating mechanisms are enclosed within the aluminum framework, with battery replacement and power switching performed externally. This design minimizes the risk of accidental injury during operation. Further adhering to the IEEE Code of Ethics, specifically the call to "be realistic in stating claims or estimates based on available data"[2], I have set forth a clear and achievable timeline for completing the tasks at hand.

Moreover, to reinforce safety, warning signs have been placed to alert users of the potential hazards during the subsystem's operation. Additionally, a series of safety-compliant operational procedures have been established, requiring power disconnection during battery changes and maintenance, in compliance with ethical guidelines that mandate prioritizing the safety and welfare of workers and the public in our engineering solutions.

5.4 Future work

For the drone, there's still a large potential for improving the physical properties. The weight reduction and mass distribution of the parts on the drone can be conducted with more workload. As we expected, the drone which is going to be produce in large quantities, should be optimized before conducting.

For ground terminal, firstly we will consider simplifying the wiring in subsequent iterations. A strategy could be to redesign the boost module and the functionalities required by the Raspberry Pi onto a new PCB. The advantage of this approach is that it allows complete control over the changes and iterations of the upper-level machine, while significantly optimizing the wiring. However, the time cost and technical difficulty of redevelopment are relatively high, which necessitates a certain level of investment. Secondly, by altering the rail design and adding force compensation, we can reduce the operational stress on the rails, thereby enhancing the long-term durability of the machinery.

For the communication subsystem, in the future, two versions of *Talon Eat* for senders (Merchants) and receivers (Customers) to help them better use the application. Also, consider the potential high load of customers, a log in interface with database may be added to avoid abuse of application. Also, automatic spinner by clicking the location of terminals on the map may also be added.

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

[8] “Basic variable of 2020A aluminum.” Available at: https://wenku.baidu.com/view/35a926527075a417866fb84ae45c3b3566ecdd4e.html

[9] IEEE. "IEEE Code of Ethics." [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: 10th March 2024].

Appendix A Requirement and Verification Table

Drone Subsystem

Table 3 Drone Subsystem Requirements and Verifications

Requirement	Verification	Verification status (Y or N)
1. Max take-off weight should be larger than 7 kg.	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.	Y
2. Flight endurance at max TOW should be longer than 10 min	2. Obtain the endurance at different load and find the max TOW which satisfies the requirement.	Y
3. The flight form can be transformed back and force with a stable attitude.	3. Test the transforming structure during flight and record the drone's attitude.	Y
4. HERE3 module should enable accurate take-off and landing.	4. Check if the drone can take off from the terminal and land on it automatically, within about 10cm error.	Y
5. Remote control needs to be a qualified backup	5. Try to take over control during the flight	Y
6. a. Pixhawk should be able to get command from Raspberry Pi and execute. b. Raspberry Pi should be able to fetch order information from cloud server and plan the path.	6. 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.	Y

Ground Terminal Subsystem

Table 4 Ground Terminal Subsystem Requirements and Verifications

Requirement	Verification	Verification status (Y or N)
7. Ground Terminal Frame is capable of withstanding a maximum platform load of 10 kilograms, with a minimum of 2 safety factor.	7. a. Build the frame in simulation software for load testing; construct a 1:1 prototype. Gradually add load until failure to test the load capacity.	Y
8. BF5M-15 Timing Pulley & 5M Timing belt can bear a vertical load of 100N with a minimum of 2 safety factor.	8. 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.	Y
9. In conjunction with the timing belt, the 57 Step Motor the minimum movement increment does not exceed 1cm.	9. 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 .	Y
10. The 57 Step Motor can provide 1 N*m of torque.	10. 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	Y

	output torque.	
11. The 2GT-6 Timing Pulley & 2GT Timing belt can bear a vertical load of 20N with a minimum of 2 safety factor.	11. 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.	Y
12. In conjunction with the timing belt, the minimum movement increment of 2HS28 Step Motor does not exceed 1cm.	12. 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	Y
13. Can provide 1N*m of torque.	13. 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.	Y
14. DM542 2H Microstep Driver can drive the 57-step motor	14. 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	Y

	driver. c. Turn on the power and observe if the motor can be driven.	
15. TB6600 2H Microstep Driver can drive the 42-step motor	15. 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.	Y
16. Raspberry Pi can successfully output PWM signal and 3.3v high voltage signal.	16. 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.	Y
17. Voltage Regulator can provide 5V from a 3.3V source. Voltage Regulator can operate at current within 300mA	17. 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.	Y
18. 24V-6000mAh li-ion battery can support our motor to use 5hrs.	18. 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 5hrs.	Y
19. The vertical displacement of the Reach Bar strips does not exceed 3cm when applying 50N vertical force at one side.	19. a. Fix a bar at one side. b. Applying 25N force at the other side vertically. c. Record the vertical displacement.	Y

	Make sure it does not exceed 3cm.	

Communication Subsystem

Table 5 Communication Subsystem Requirements and Verifications

Requirement	Verification	Verification status (Y or N)
20. Drone Raspberry Pi can send location data and other status data to cloud server every preset time interval, say 5s.	20. 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. See whether the location of the drone and other status data on Talon Eat have been correctly updated. d. (Alternative) Using "Device Log" module on Aliyun Cloud Platform to check the data delivery rate.	Y
21. Drone Raspberry Pi can alert ground terminal when approaching in 50m.	21. a. Change drone into delivery mode. (not necessarily to take off) b. Put the drone within 50m range of the terminal. c. See whether the ground terminal opens the container. d. (Alternative) Using "Device Log" module on Aliyun Cloud Platform to check whether the message is delivered.	Y
22. Drone Raspberry Pi can interpret the operation code form cloud server correctly.	22. a. Let the drone be ready to take off. b. Send an order on Talon Eat. c. See whether the drone go to the right departure terminal. d. (Alternative) If other subsystem works correctly, see whether the drone autopilots to the right destination terminal.	Y
23. All required functions can be performed under cellular (non-Wi-	23. b. Disconnect Wi-Fi on Raspberry Pi and launch ME909s network	Y

Fi) network.	module. c. Perform 1-3 requirement again to see succeed or not. d. (Alternative) If b fails, try ping command in terminal.	
24. Terminal Raspberry Pi can open or close the correct cabinet.	24. a. Turn on the Raspberry Pi in the terminal and connect it to the motors. b. Send a test message using mobile application. c. See whether the correct container is open or closed. d. (Alternative) Using “Device Log” module on Aliyun Cloud Platform to check operation message.	Y
25. Terminal Raspberry Pi can be alerted when the drone is approaching in 50m.	25. a. Change drone into delivery mode. (not necessarily to take off) b. Put the drone within 50m range of the terminal. c. See whether the container is ready to open. d. (Alternative) Using “Device Log” module on Aliyun Cloud Platform to check whether the message is delivered.	Y
26. Terminal Raspberry Pi can react to mobile application request in 100ms.	26. a. Send a test request on Talon Eat. b. Using “Device Log” module on Aliyun Cloud Platform to check whether the latency is under 100ms.	Y
27. Cloud Platform can handle messages transfer bidirectionally	27. 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.	Y
28. Cloud Platform’s “Data Forwarding” can correctly forward data to correct topic.	28. a. Send a test message from mobile application. b. Check whether other devices receive the messages.	Y

	c. (Alternative) Check “Data Forwarding” on cloud platform to see whether the data is correctly forwarded.	
29. Mobile Application can send order message to ground terminal within 100ms (ideally) under both cellular and Wi-Fi environment.	29. a. Send a message using mobile application. 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.	Y
30. Mobile Application can display drone location with error less or equal than 5m.	30. 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.	Y
31. (If 20. cannot work) bMapview is correctly displayed and user location is displayed with error in 5m (outdoors).	31. a. Launch application outdoors with GPS and Wi-Fi permission on. b. Check whether the location is accurate. c. (Alternative) if not, open Baidu Map to see whether the locations are similar.	Y
32. Mobile Application permissions (data storage, internet and location) are correctly authorized.	32. a. Launch the application on a device that does not have Talon Eat (i.e., not update). b. Check whether prompt boxes pump out to require permissions. c. Check whether MQTT connection is established and map is displayed.	Y
33. Operation code correctly interpreted given the spinner input.	33. a. Correctly launch the application and choose a departure point and a destination point. b. Send the request. c. Check whether the Raspberry Pis on the corresponding terminals receive the message.	Y
34. Mobile Application can display velocity and height data of the	34. a. Let the drone take off. b. Check whether the data on the	Y

drone every 30s.	application is displayed and updated. c. (Alternative) If b fails, use "Device Log" module on Aliyun Cloud Platform to check the message.	
35. Mobile Application can avoid duplicate orders to the same starting or destination container.	35. a. Let the drone take off to perform a simple delivery. b. Send a new delivery request, see whether an error message pop out.	Y

Appendix B Other Content

Introduction to MQTT Protocol

MQTT (Message Queuing Telemetry Transport) is a "lightweight" communication protocol based on the publish/subscribe model. It is built on the TCP/IP protocol and was initially released by IBM in 1999.

MQTT offers several key features. Firstly, it utilizes the publish/subscribe information model, enabling one-to-many message distribution and decoupling applications. Secondly, it shields the transmission of payload content, focusing on the delivery of messages rather than their specific format. Thirdly, it leverages TCP/IP for network connectivity, ensuring reliable data transmission.

The main advantage of MQTT lies in its ability to provide real-time and reliable messaging services for connecting remote devices with minimal code and limited bandwidth. The use cases of MQTT are diverse and span across various industries. In the IoT domain, it facilitates communication between a vast number of devices operating in complex network environments. In scientific research, MQTT ensures stable and complete data transmission for experimental data collection. In the financial industry, it enables real-time monitoring and secure transmission of transaction information. In healthcare, MQTT handles the transmission of large volumes of patient vital sign data, ensuring its integrity. Furthermore, it finds applications in energy management, logistics, and other fields where real-time monitoring and control are crucial.

Introduction to HUAWEI ME909s-821

HUAWEI ME909s is a state-of-the-art LTE module that offers exceptional performance and reliability for a wide range of applications. With its advanced technology and features, it is an ideal choice for those seeking a robust and efficient wireless communication solution.

ME909s boasts impressive LTE capabilities, enabling high-speed data transmission and seamless connectivity. Whether it's for mobile computing, IoT devices, or industrial automation, the module ensures reliable and efficient communication, even in challenging network environments.

Moreover, the ME909s is designed with a compact form factor, making it easy to integrate into various devices and systems. Its small size and lightweight design allow for seamless integration without compromising on performance.

In terms of performance, the ME909s delivers excellent data throughput and low latency, ensuring smooth and responsive communication. Its robust signal reception and transmission capabilities make it reliable even in areas with weak or unstable network coverage.

After consideration of both usage and cost, we finally chose this module and after some testing, it is now working well together with the Raspberry Pi.

MQTT Protocol Pseudocode with Flowchart

Initialize MQTT client

Connect to MQTT broker

IF connection successful THEN

IF Subscriber THEN

Subscribe to specific topic

Wait for update

IF update THEN

React to operation code

END IF

IF Publisher THEN

Wait for publishing

IF need publishing THEN

Publish

Data forwarding on

cloud platform

END IF

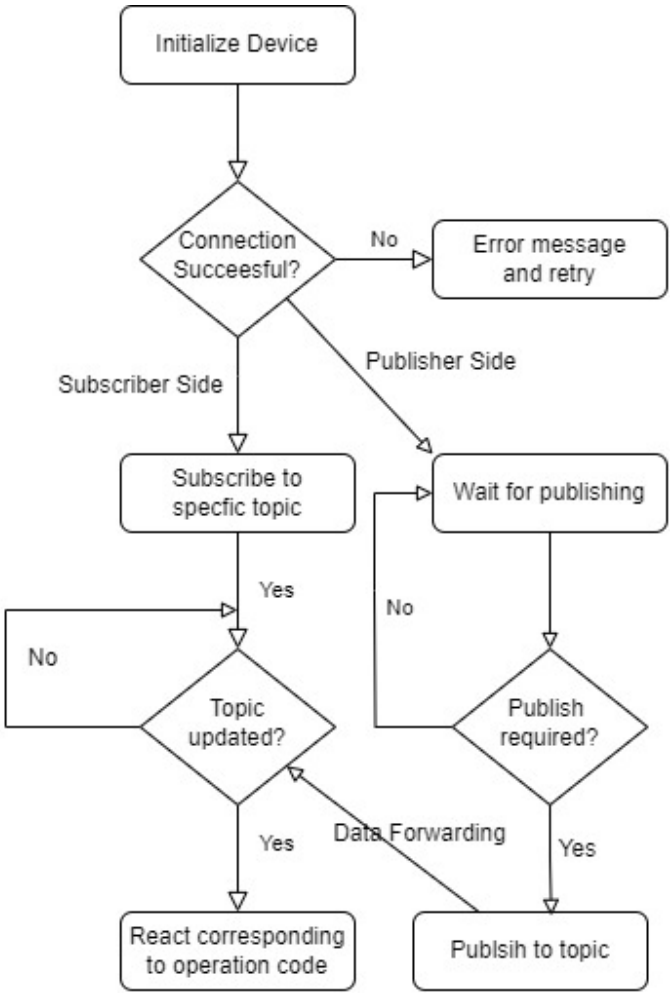
END IF

Disconnect from MQTT broker

ELSE

Display error message

END IF



Introduction and Application of Baidu Map API

Baidu Map Open Platform is a comprehensive location-based services (LBS) platform offered by Baidu, a leading technology company in China. This platform provides developers with a wide range of APIs and SDKs to integrate mapping and location-based functionalities into their applications seamlessly.

Key features of the Baidu Map Open Platform include interactive map display with various viewing modes, geocoding and reverse geocoding services for address-to-coordinate conversions, and route planning with real-time traffic updates. Additionally, it offers location-based services like location search, real-time location sharing, and tracking, enabling developers to build applications for navigation, logistics, and asset tracking.

The platform supports indoor mapping for venues like shopping malls and airports, allowing developers to create indoor navigation solutions. Customization options are available for map styles, markers, and overlays, ensuring a personalized user experience. Baidu Map SDKs are compatible with multiple platforms, including Android, iOS, and web browsers, facilitating cross-platform development [4].

In our application design, we have location listener function to help the users to locate themselves. Also, given the longitude and latitude, the track of the drone is also marked on the map.

During my implementation, I also tried to use the location function by Android itself, however the location is not accurate so finally Baidu's method is applied.