Design and Implementation of an Autonomous Quadcopter System

Yikun Guo, Jinming Ren, Chun Wang, Qian Xia, Haichuan Gao, Jinhao Cao
(UESTC & UofG Joint School, Team16: Golden Snitch)

***Abstract*—This article presents the design and implementation of an autonomous quadcopter system capable of indoor navigation, object detection, and package retrieval using computer vision techniques. The system integrates multiple subsystems, including a downward-facing camera for real-time image processing, a flight controller with PID stabilization, and OrangePi running ROS2 for flight logic control.**

***Index Terms*—UAV, PID Control, Computer vision, OrangePi**

I. Introduction

UNMANNED aerial vehicles (UAVs) have great potential for indoor inspection and delivery tasks, but they face challenges in navigation and manipulation without GPS [1]. Our project addresses these challenges by developing an indoor autonomous drone system for object detection, precision landing, and package retrieval. The design integrates multiple subsystems in a modular framework to achieve real-time performance:

1) **Computer vision for object detection and landing:** Downward-facing camera and onboard image processing to recognize target objects and landing pads.
2) **Flight control with PID and IMU:** Low-level controller stabilizes the drone with PID loops using IMU feedback.
3) **ROS2-based autonomy:** A companion computer (OrangePi) running ROS2 orchestrates high-level navigation and landing logic, communicating with the flight controller.
4) **Ultrasonic sensor and passive gripper:** Ultrasonic rangefinder assists altitude control for landing; a passive mechanical grabber enables package pickup without active actuators.

Computer vision enables the drone to perceive its environment and locate targets for landing and pickup, which is critical indoors without GPS. Fiducial markers like AprilTag or ArUco are a common solution for precise autonomous landing. For example, Chang *et al.* landed a quadrotor on a moving platform by fusing ArUco visual feedback with onboard inertial data [2]. In our system, the onboard computer detects a designated visual marker or the package itself in real time using lightweight vision algorithms.

Stable flight is maintained by the quadrotor's autopilot using cascaded PID controllers with IMU feedback. PID control is a standard method for quadrocopter stabilization due to its simplicity and robustness [3]. In our design, a Crazepony flight controller runs the low-level attitude and altitude loops, while the OrangePi processor handles higher-level decision making. The OrangePi (running ROS2) computes navigation and mission commands and sends setpoints to the the Crazepony via UART [4]. This separation of high-level autonomy from the real-time control loop improves system modularity and flexibility.

To aid in landing, an ultrasonic rangefinder provides short-range altitude measurements. Integrating an ultrasonic altimeter can improve precision during final descent [5], though such sensors have limited range and a narrow field of view. In our implementation, the sonar is used only below a threshold height to help the drone hover steadily just before touchdown.

For payload retrieval, the drone uses a passive mechanical gripper that requires no motors. Passive grippers reduce weight and energy consumption, as shown by drones that grasp objects by impact without actuators [6]. Our spring-loaded claw remains open in flight and snaps shut on the package upon contact. This solution is energy-efficient but limits the shapes and weights of objects that can be grasped securely.

The scenario for this project is a controlled indoor environment and a small payload. There are several challenges to overcome, such as clear and consistent computer vision under different lighting conditions, stable and robust flight control algorithm, and reliable object detection and retrieval. In the following sections, we will describe the overall system design, the results of our experiments, and the analysis of the system performance

II. Overall system design approach

In order to maximize development efficiency, our team adopts highly modularized design. To achieve this, our plan was to first specify the input and output of every module, then wire them up with the OrangePi central processor. The overall system architecture is shown in Figure 1. Each person in our group is responsible for one of the blocks in Figure 1. We also collaboratively design and plan the connections and interactions between different modules.

Our choice of components is shown in Figure 2, which lists the hardware modules and physical connections. The main control is an orangePi 5, which runs ROS2 and the whole logic of mission. It also directly handles the data

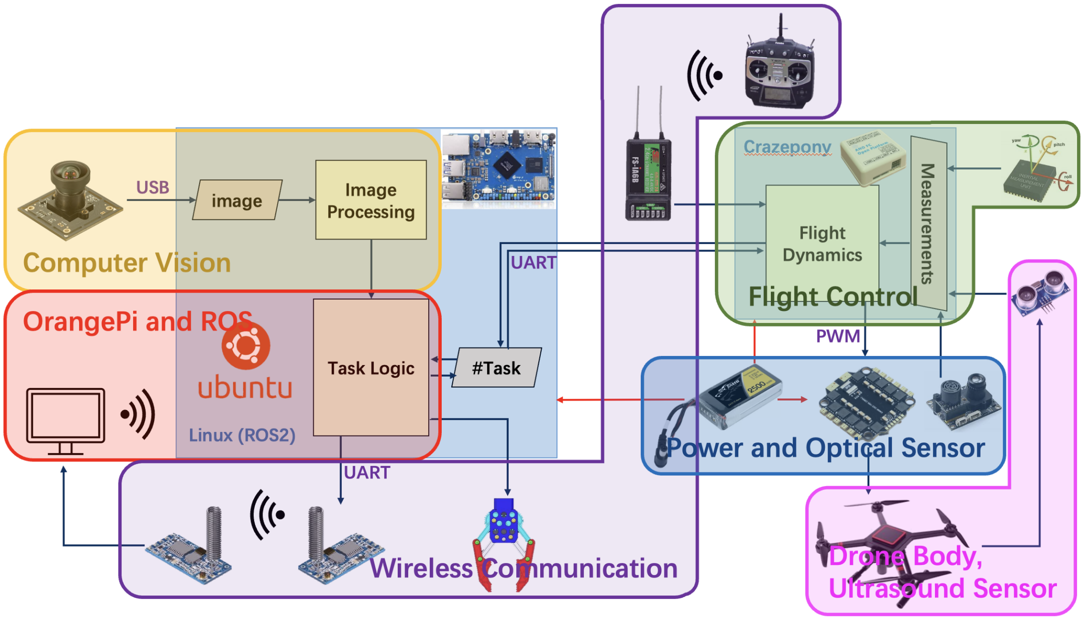


Fig. 1. Overall system architecture and task distribution.

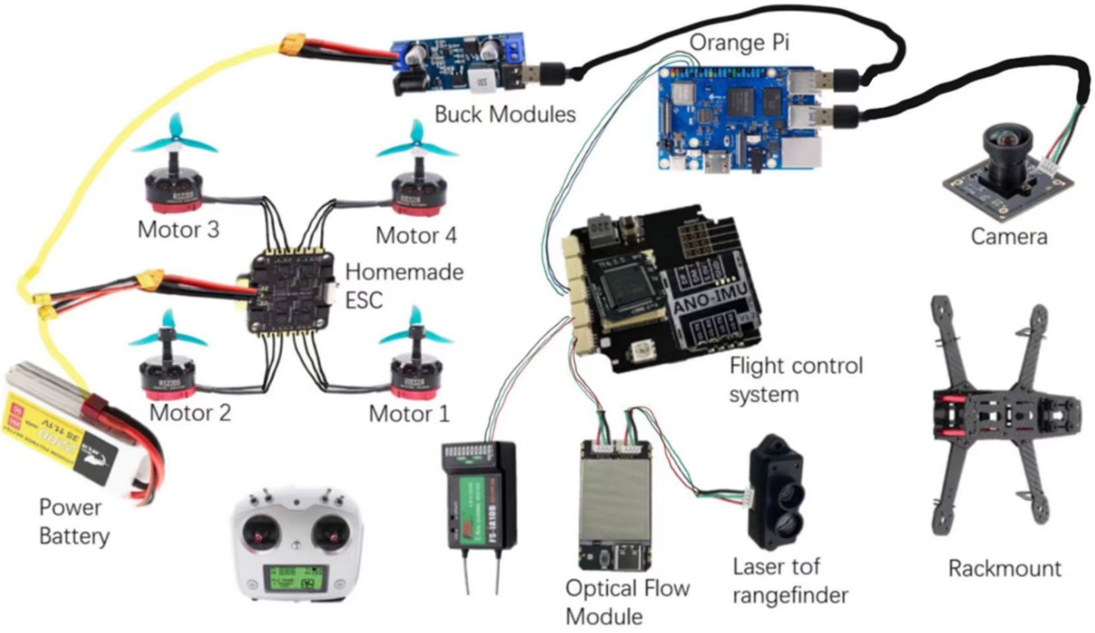


Fig. 2. Physical connections of the individual modules.

from the camera. The flight controller is Crazepony, which is a STM32-based flight controller that runs the low-level PID control loops. The following sections will describe the design and implementation of each module in detail. We provide a complete hardware component breakdown and cost in Table I. The total cost of the hardware is ¥1375, which is within the budget of ¥1500.

The final assembled picture of the drone is shown in Figure 3. This project was started on February 24, 2025 and was completed by June 4, 2025. The project was conducted according to the flowchart shown in Figure 4.

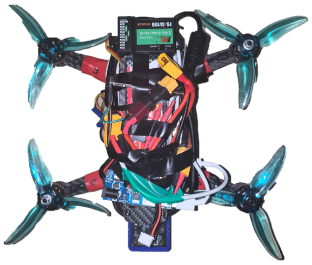


Fig. 3. The final assembled drone used in the TDPS project.

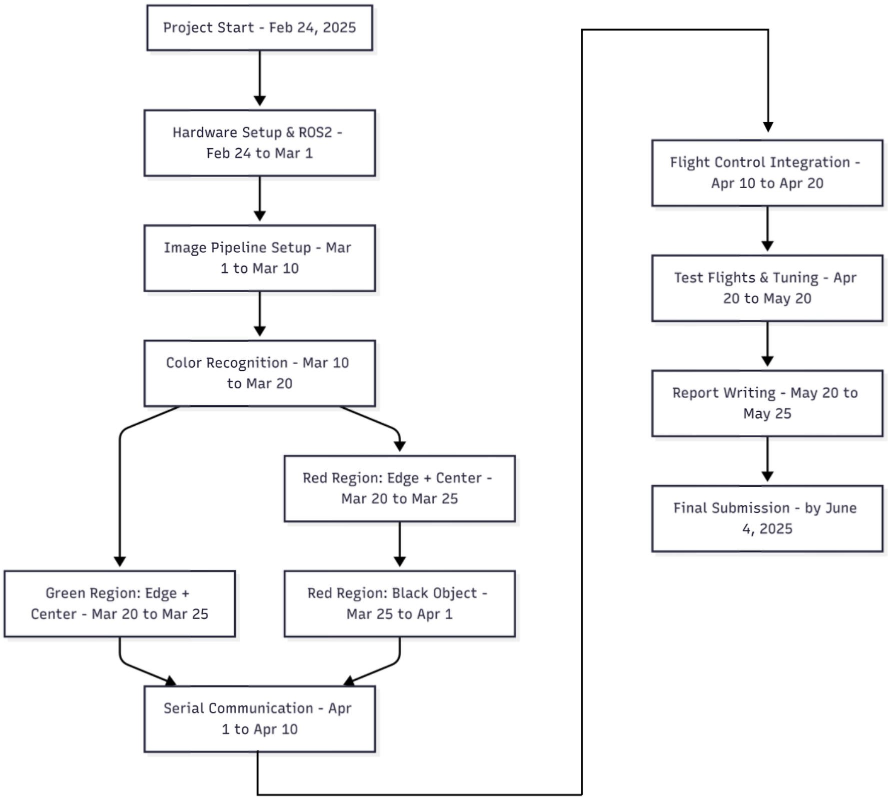

Project Start - Feb 24, 2025
Hardware Setup & ROS2 - Feb 24 to Mar 1
Image Pipeline Setup - Mar 1 to Mar 10
Color Recognition - Mar 10 to Mar 20
Red Region: Edge + Center - Mar 20 to Mar 25
Green Region: Edge + Center - Mar 20 to Mar 25
Red Region: Black Object - Mar 25 to Apr 1
Serial Communication - Apr 1 to Apr 10
Flight Control Integration - Apr 10 to Apr 20
Test Flights & Tuning - Apr 20 to May 20
Report Writing - May 20 to May 25
Final Submission - by June 4, 2025

Fig. 4. Flowchart for the project planning.

TABLE I
Complete Hardware Component Breakdown and Cost

Component	Specification	Cost (RMB)
Core Flight System		
Frame	250mm Wheelbase	¥53
Battery	2300mAh 3S (25C) LiPo	¥58
Motors (x4)	2205 2300KV	¥96
ESC (Custom)	45A	¥20
Propellers (x4)	5-inch, 3-Blade, 4.99” Pitch	¥8
MCU Core	STM32F407	¥20
MCU Extension Board	Development Board	¥30
Closed-Source IMU	Lingxiao IMU	¥360
Onboard Computing & Perception		
Onboard Computer	Orange Pi 3B	¥235
Camera Module	X003 1080p, 2.9mm lens	¥190
Ranging Module	Optical Flow & ToF Laser	¥230
Ultrasonic Module	HC-SR04	¥5
Peripherals & Communication		
Wireless Module	HC-12	¥15
Data Transmission Module	-	¥50
Grasping Module	3D-Printed	¥5
Total		**¥1375**

III. Experimental Design - Subsystem Design and Solutions

A. OrangePi and ROS

This submodule receives vision data via ROS topics, tracks task progress using a flag-based state machine, and sends motion commands to the flight controller over UART. The overall control logic is shown in Figure 5, which include the following main steps:

1) **Take-off and Climb:** The drone initiates the mission by taking off and rising to a fixed altitude of 2.4 meters. This establishes a safe height for tunnel navigation.
2) **Tunnel Navigation:** Using its vision system, the drone detects and flies through a sequence of marked tunnels. This phase requires alignment and forward motion through constrained passageways. Due to the

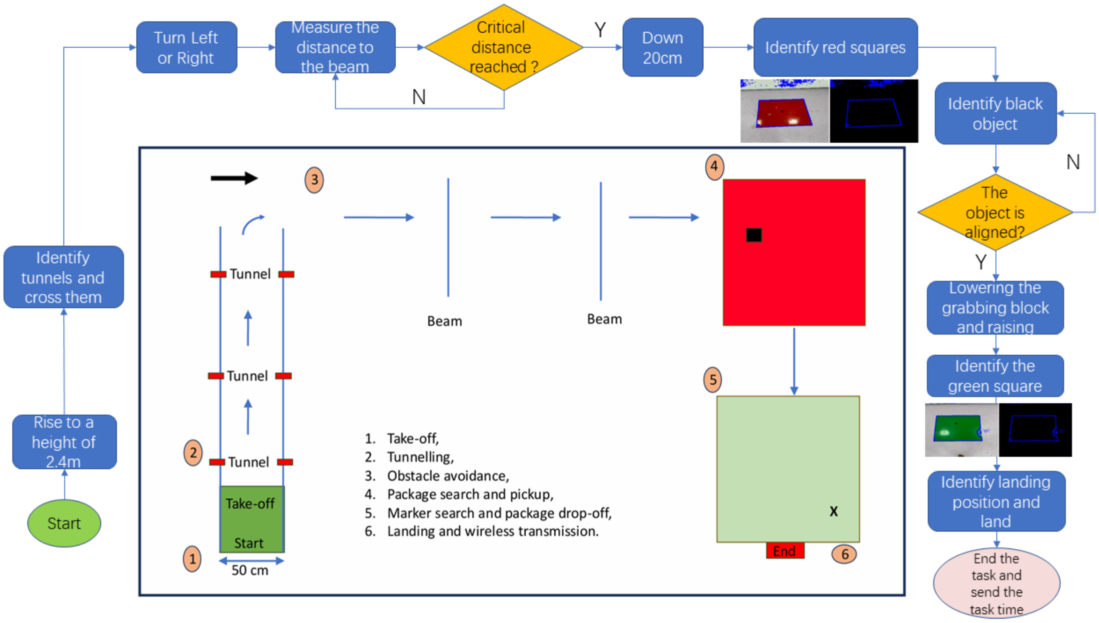

Start
Rise to a height of 2.4m
Identify tunnels and cross them
Turn Left or Right
Measure the distance to the beam
Critical distance reached ?
Y
N
Down 20cm
Identify red squares
Identify black object
The object is aligned?
N
Y
Lowering the grabbing block and raising
Identify the green square
Identify landing position and land
End the task and send the task time
3
Tunnel
Tunnel
Tunnel
2
Take-off
Start
1
50 cm
Beam
Beam
4
5
X
End
6
1. Take-off,
2. Tunnelling,
3. Obstacle avoidance,
4. Package search and pickup,
5. Marker search and package drop-off,
6. Landing and wireless transmission.

Fig. 5. Flowchart of the logic for the autonomous drone.

danger of collision while testing, we omit the actual tunnel navigation in the final implementation.

3) **Beam Avoidance:** The drone flies forward under ceiling beams and continuously measures the vertical distance to them.
   - If a beam is too close (critical distance reached), the drone descends until it is at a safe height.
   - If not, it continues flying or turns left/right as needed.
4) **Package Search and Pickup:** The drone identifies a red square as the pickup zone. Within this region, it searches for a black object (the target item).
   - If the object is not aligned, it continues searching.
   - Once aligned, it lowers the grabbing mechanism to capture the object and then ascends.
5) **Marker Search and Drop-off:** After acquiring the object, the drone searches for a green square, which represents the delivery pad. Upon detection, it calculates the center position for accurate landing.
6) **Landing and Completion:** The drone performs a precision landing on the green marker. Once landed, it ends the task and transmits the task completion time.

B. Flight Control System

The drone's flight control system is a critical subsystem that serves as the bridge between high-level autonomous commands and stable physical flight. The design and implementation of this subsystem involved a multi-stage process of hardware optimization and firmware development.

The core components used in this part of the project are:

1) **Propulsion System:** To meet the project's requirements for flight time and payload capacity within a strict budget, a simulation tool was first used to model the performance of various hardware combinations. The selected hardware provides an optimal balance of performance and cost. The final components include 2205 2300KV motors, 5-inch, 3-blade propellers, and a 2300mAh 3S LiPo battery, all mounted on a 250mm wheelbase frame. This configuration was validated by simulation to be capable of meeting the mission's performance requirements.
2) **Hybrid Flight Control Architecture:** To ensure maximum flight stability, two distinct flight control architectures were evaluated: a fully open-source system and a hybrid system. The project ultimately adopted the hybrid approach, which pairs a programmable microcontroller (MCU) with a specialized, closed-source Inertial Measurement Unit (IMU). This decision was made because the closed-source IMU provided superior, pre-tuned stability and vibration damping, which was more pragmatic given the project's tight schedule. This allowed development efforts to focus on the high-level control logic, which was more critical for the project's success.
3) **Microcontroller (MCU):** The central component of the flight controller is an STM32F407 MCU. It serves as the intelligent intermediary between the high-level Orange Pi computer and the low-level IMU. The most significant contribution was the development of custom firmware for the MCU, centered on a state machine-based command parser. This parser was designed to process both simple (single-byte) and complex (multi-byte) commands sent from the Orange Pi, enabling the execution of

precise, parameterized maneuvers.

4) **Inertial Measurement Unit (IMU):** A closed-source Lingxiao IMU was used for low-level flight control and stabilization. This specialized unit contains proprietary sensor fusion algorithms and offers enhanced vibration resistance, which is critical for a UAV platform. It receives instructions from the MCU, performs complex flight dynamics calculations, and outputs the necessary PWM signals to the motors to achieve controlled flight. The use of this high-performance IMU resulted in a demonstrably stable and highly responsive control system.

C. *Computer Vision*

This submodule was developed to enable the quadcopter to autonomously detect and identify landing zones and packages in real-time, using a GPU-free pipeline on an orangePi embedded system. The system processes video input from an IMX462-H264 USB camera and publishes interpreted environmental data via ROS2.

The algorithm consists of three main stages:

1) **Color Recognition:** The Excess Red Index (ExR) method was used over simple R-channel thresholding and HSV methods for its robustness against noise and lighting variations. Hard-coded thresholds were selected to reliably distinguish red and green markers from the background.
2) **Edge Detection:** OpenCV's findContours() function was employed to identify contours of colored regions. False edges were filtered based on contour area, ensuring that only the most significant shapes were considered as valid targets.
3) **Vertices Detection:** The Ramer-Douglas-Peucker (RDP) algorithm was applied to approximate the contours and extract vertices, which were then used to determine the presence of a square/box, color type, center coordinates, and orientation angle.

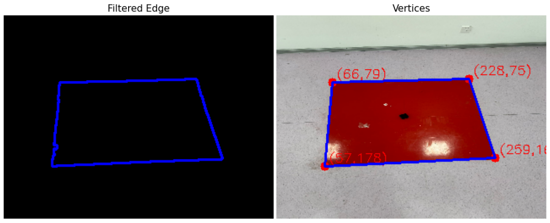


Fig. 6. Vertices detection results for red square.

These results were published as a ROS2 topic (`CV_status`), which the drone's control logic used to perform navigation and manipulation tasks. The module achieved a 90% success rate in detecting landing areas under varied lighting conditions. However, inaccuracies occasionally arose due to the fixed focal length affecting the reliability of the RDP algorithm.

D. *Wireless Communication*

This subsystem is mainly responsible for the communication part of the unmanned aerial vehicle (UAV), and is divided into two key components: the communication between the remote control and flight control systems, and the wireless communication between the UAV and the ground station. Firstly, the communication between remote control and the flight control system ensures that in the event of a flight control failure, the operator can immediately switch to manual mode and directly control the unmanned aerial vehicle through the remote control to guarantee flight safety. To achieve this goal, an interrupt function has been designed in the flight control system, which can quickly switch to manual mode when receiving the switching signal from the remote control.

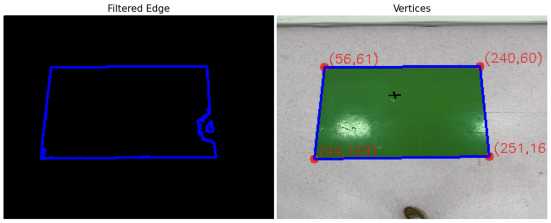
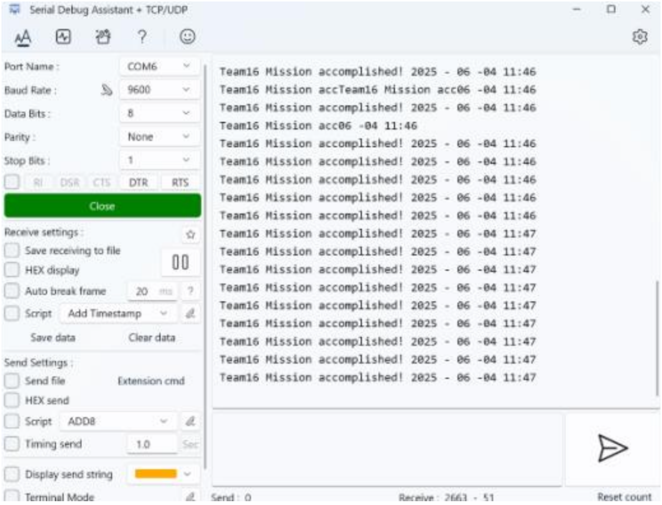


Fig. 7. Vertices detection results for green square.

Secondly, the HC-12 wireless communication module is used for data transmission after completing the task. When the unmanned aerial vehicle (UAV) completes the mission and lands safely, the HC-12 module transmits key data (such as mission completion confirmation and accurate landing time) to the ground station to ensure the timely feedback of mission information. Through the combination of the orangePi computing platform and the HC-12 module, the stable and reliable data transmission after the task is completed is ensured. The test results show that the system can operate stably under different test conditions and provide a solid foundation for the future expansion of the communication system.



Port Name : COM6
Baud Rate : 9600
Data Bits : 8
Parity : None
Stop Bits : 1

Receive settings :
□ Save receiving to file
□ HEX display
□ Auto break frame 20 ms
□ Script Add Timestamp

Send Settings :
□ Send file Extension cmd
□ HEX send
□ Script ADD8
□ Timing send 1.0 Sec
□ Display send string
□ Terminal Mode

```
Team16 Mission accomplished! 2025 - 06 -04 11:46
Team16 Mission accTeam16 Mission acc06 -04 11:46
Team16 Mission accomplished! 2025 - 06 -04 11:46
Team16 Mission acc06 -04 11:46
Team16 Mission accomplished! 2025 - 06 -04 11:46
Team16 Mission accomplished! 2025 - 06 -04 11:46
Team16 Mission accomplished! 2025 - 06 -04 11:46
Team16 Mission accomplished! 2025 - 06 -04 11:46
Team16 Mission accomplished! 2025 - 06 -04 11:46
Team16 Mission accomplished! 2025 - 06 -04 11:47
Team16 Mission accomplished! 2025 - 06 -04 11:47
Team16 Mission accomplished! 2025 - 06 -04 11:47
Team16 Mission accomplished! 2025 - 06 -04 11:47
Team16 Mission accomplished! 2025 - 06 -04 11:47
Team16 Mission accomplished! 2025 - 06 -04 11:47
Team16 Mission accomplished! 2025 - 06 -04 11:47
Team16 Mission accomplished! 2025 - 06 -04 11:47
Team16 Mission accomplished! 2025 - 06 -04 11:47
```

Send : 0 Receive : 2663 - 51

Fig. 8. The final received data from the ground station.

E. Ultrasonic Sensor and Grabbing System

Two key subsystems were implemented in the drone to address indoor mission requirements: an ultrasonic-based altitude sensing module and a passive grabbing mechanism. The ultrasonic module, built around the HC-SR04 sensor, was mounted on top of the drone to measure ceiling distance in real-time. It provided vertical positioning data to the ROS-based flight controller, enabling dynamic obstacle avoidance and altitude stabilization. Filtered range data was published at 10 Hz to ensure robust control even under noisy conditions. The grabbing mechanism consisted of a spring-latched acrylic plate with a rubber surface, fixed beneath the drone. Upon contacting a target object, the latch released and allowed the plate to hang freely, using friction during ascent to lift lightweight packages. This passive design required no electrical components and enabled reliable object pickup with minimal complexity.

F. Data Transmission Module

We used LR24 data transmission module to receives the data from the quadrocopter and transmits it to the ground station. The data includes the flight status, the current position of the quadrocopter, and the status of the payload. The first step is to connect one of the LR24 data transmission modules to the flight control module at its port TELEM1 with the PIN sequence of the LR24 data transmission.

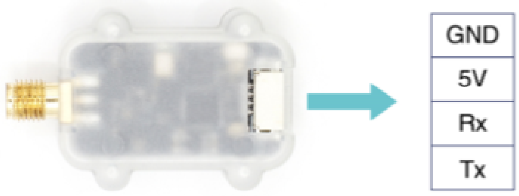


Fig. 9. Pins of LR24 data transmission module.

The corresponding pins on the TELEM1 of the flight control module will be connected to pins TX, RX to achieve UART communication between flight control module and LR24. Then, another module will be connected to the ground station through USB connection.

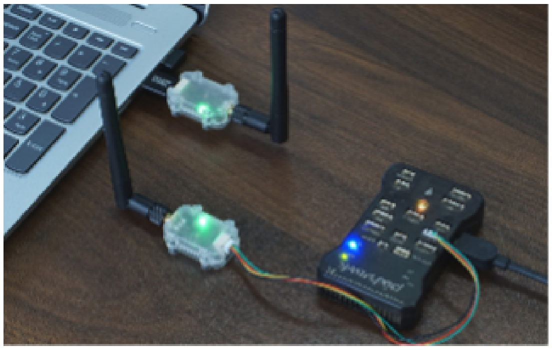


Fig. 10. The connection of LR24 data transmission module.

Whenever the flight control module is on operation, the data it collects or other module collects like flight attitude, altitude, speed, position, sensor data will be transmitted to the connected LR24 data transmission module though UART communication, then the LoRa spread spectrum within the module modulates these data signals, converting them into signals suitable for transmission on radio waves, and extends them to a wider frequency band to improve the anti- interference ability and transmission distance of the signals. Then, LR24 transmit them in the form of electromagnetic waves through the transmitting antenna, and after the LR24 data transmission radio of the ground control station receives these electromagnetic wave signals through the receiving antenna, the receiving part demodulates and decodes the signals, restores the received analog signals to the original data, and then sends them to the computer or other equipment of the ground control station through USB interface, thereby achieving data communication between the drone and the ground control station. During this process, FHSS frequency hopping technology enables radio stations to continuously switch frequency points during transmission, further enhancing anti-interference capabilities and communication security.

G. Homemade Brushless Electronic Speed Controller (ESC)

In the design of brushless ESC, its schematic was first drawn, which shows a microsystem composed of multiple sub circuits with unique functions. The power supply sub circuit uses the XT30PW-M connector to receive power from a 6S lithium battery (2700mAh). This connector can withstand a maximum input voltage of over 100V and a constant current of over 20A, meeting the power transmission requirements of electrical regulation. The power regulation sub circuit adopts the LDO AS78L05RTR-E1 chip, which stabilizes the battery voltage to 5V to support the operation of other sub circuits. The main control sub circuit and power output sub circuit are based on the STC15W408AS-35I-SOP16 microcontroller, which receives PWM signals from the flight control module and converts them into three-phase signals to drive the motor. The PWM signal is amplified before being transmitted to the MOSFET driver circuit to ensure effective driving of the MOSFET. Each power output sub circuit uses two MOSFETs (AON6407 and AON6354), whose largest drain current is much greater than 20A at 25 ℃, and their on resistance is less than 4.5 mΩ, which helps to reduce power consumption and improve operational safety. The test sub circuit is constructed through the GH125-S04CCA-00 connector to achieve UART communication, allowing for the simulation of input PWM signals through the host computer and Host Message Interface (HMI), thereby safely testing the function of the brushless ESC without relying on the actual flight control module or other UAV modules. Finally, it was verified through actual testing that the brushless ESC can stably drive the motor and was successfully applied in unmanned aerial vehicles, proving the effectiveness and safety of the design.

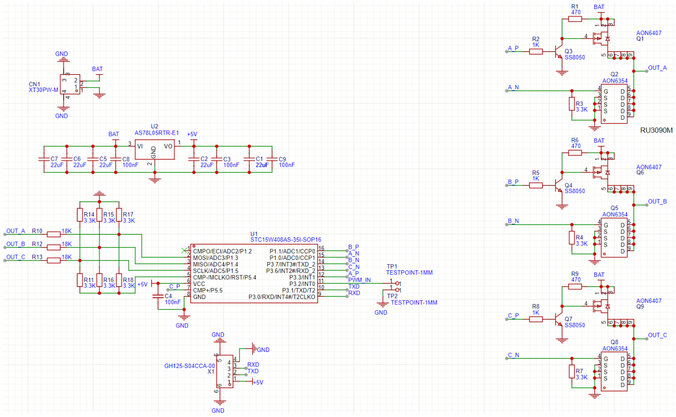


Fig. 11. Schematic of the brushless ESC.

H. Optical Flow Sensor

After determining the principle of optical flow module, the integrated module of optical flow and laser ranging leveraging the TOF principle to perform laser ranging was chosen so that by analyzing the motion of the pixels in the image sequence shot by the camera on the optical flow, the two-dimensional optical flow data would be gotten, and with the laser ranging module determining the height data of the drone, the three-dimentional optical data can be combined and gotten so that the displacement and velocity of the drone and its relative location towards the surrounding environment can be sensed by the flight control module, enabling hovering in low attitude and automatedly modulate the flight path and the attitude. After this,the integration work of the optical flow module was done, ensuring that it can effectively communicate with the flight control module and provide stable data during flight. As shown in the timing diagram, the flight control module communicates with the MTF-01 TOF ranging module via UART. Initially, the sensor is in a low - power idle state, waiting for an activation command. Upon receiving the ”Start Ranging” command, it transitions to Active Ranging mode, initiating the initialization sequence to configure parameters like measurement frequency and ranging range through UART register writes. Then, it enters the Ranging cycle, where the 850 nm LED emits pulses and the receiver captures reflected signals for TOF calculation using Distance $= (c \times \Delta t)/2$, with c as light speed and Δt as pulse round $-$ trip time. After computing the distance, the sensor activates the GPIO1 interrupt line. The flight control module reads the measurement results from the sensor’s data registers via UART and clears the interrupt flag. During the Inter-Measurement Period, the system can reconfigure parameters or prepare for next measurements. In continuous ranging mode, the sensor repeats the ranging cycle at a frequency determined by the timing budget, up to 100 Hz. The cycle stops when the flight control module sends a “Stop Ranging” command or pulls the XShut pin low. This workflow creates a closed-loop control architecture via UART commands and interrupt-driven synchronization. By adjusting the Timing Budget and Inter-Measurement Period, the system can be optimized for various applications.

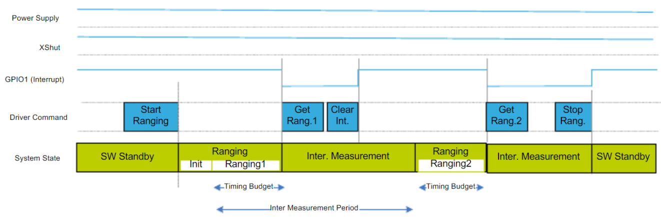


Fig. 12. Timing diagram of the optical flow sensor.

IV. Results and Analysis

The successful integration of multiple subsystems in our drone project was demonstrated through a series of structured indoor flight trials. These trials cover the full autonomous mission cycle: **takeoff, navigation, object search, pickup, delivery, and landing**. The evaluation focused on both functional correctness and system-level reliability, providing evidence that the team’s modular architecture and design choices achieved their intended objectives. The execution timings and success rates of each task were recorded across 10 flight trials, as shown in Table II.

TABLE II
Performance Metrics for Each Mission Task

Mission Task	Mean Time	Success Rate	Trials
Take-off	13s	100%	10
Avoidance Navigation	15s	100%	10
Search for Target	9s	100%	10
Pick Up Object	40s	60%	10
Land on Pad	25s	90%	10
Other Navigation	30s	100%	10

As shown in Table II, searching for the target takes least of the time, which means the computer vision module is highly effective. Picking up the object takes the longest time and has the lowest success rate, the possible reason is as follows: Due to the strong downwash generated by the drone’s propellers, the object had to be fixed to the ground. Lifting it required significant upward thrust, but our drone’s lift capability was limited. Even when the pickup was successful, the added weight shifted the drone’s center of gravity, making stable flight difficult afterward.

However, the success rate of the rest missions are 100%, which means that our system achieved relatively robust and consistent performance in these basic functions.

In this project, we have successfully solved the folowing problems:

1) **Stable and responsive flight control:** According to Table II, the success rate for basic take-off and landing tasks is almost 100%, which means this setup enabled fast and stable PID-based attitude control, validated through pitch-rate step response tests with minimal overshoot. Moreover, we found

that the optical sensor module plays a crucial role in maintaining stable flight.

2) **Real-time, GPU-free computer vision:** The vision system implemented on the OrangePi achieved robust object and landing pad detection using lightweight algorithms (Excess Red index and RDP). This allowed accurate localization in real time without GPU acceleration, achieving a 90% detection success rate.
3) **Navigation and control logic:** The drone successfully executed all basic navigation tasks, including obstacle avoidance and landing square search, using a ROS2-based architecture. This demonstrated the effectiveness of modular software design and the strong capabilities of ROS2 operating system.

V. Conclusion and Future Work

This report details the design and implementation of an autonomous quadcopter system, focusing on its tunneling navigation, object detection, and package retrieval capabilities. The system is composed of a downward-facing camera for real-time image processing, a flight controller with PID stabilization, and an OrangePi running ROS2 for flight logic control. The key findings and conclusions are as follows:

The quadcopter system successfully integrates multiple subsystems, including flight control, computer vision, wireless communication, and object grasping. The modular design approach allows for efficient development and effective collaboration among team members. Each subsystem plays a vital role in the overall functionality of the drone.

In terms of hardware integration, the combination of the Crazepony flight controller and the OrangePi provides a stable and responsive platform for autonomous flight. The flight controller with the help of optical flow module handles low-level attitude and altitude control using PID loops and IMU feedback, while the OrangePi manages high-level navigation and decision-making through ROS2. This separation of responsibilities enhances system modularity and flexibility. The propulsion system, carefully selected through simulation and testing, delivers adequate thrust and flight time for the mission requirements. The hybrid flight control architecture, incorporating a programmable MCU and a specialized closed-source IMU, ensures maximum flight stability and reliability.

The computer vision subsystem demonstrates effective object and landing pad detection using lightweight algorithms on the OrangePi. The Excess Red Index method for color recognition and the Ramer-Douglas-Peucker algorithm for vertices detection enable robust performance under varied lighting conditions. The vision system achieves a 90% success rate in detecting landing areas, highlighting its reliability in real-time applications. The wireless communication subsystem provides stable data transmission between the drone and the ground station, ensuring timely feedback of mission information. The HC-12 module and the LR24 data transmission module work together to facilitate reliable communication under different test conditions.

The ultrasonic sensor and passive gripper subsystem addresses indoor mission requirements effectively. The ultrasonic module assists in altitude control and obstacle avoidance during landing, while the passive mechanical gripper enables energy-efficient package pickup without the need for active actuators.

The flight trials conducted demonstrate the successful integration of the quadcopter system, with high success rates in tasks such as take-off, navigation, target search, and landing. However, challenges remain in object pickup due to the drone's limited lift capability and the impact of added weight on stability. Despite these issues, the system still completes all the tasks above effectively and achieves relatively robust and consistent performance in basic functions.

Although our project was quite successful, some improvements can still be made to make our drone more operates more flexibly and stably with higher task success rate in the future. Adopting more advanced control algorithms (such as model predictive control or sliding mode control) to improve the stability and response speed of the aircraft can be an example, and developing adaptive control strategies to cope with changes in flight conditions. Integrating 3D perception technology (such as binocular cameras or LiDAR) to enhance obstacle avoidance and navigation capabilities in the future is also in consideration, while exploring more advanced machine learning techniques to improve the accuracy and robustness of object detection and recognition. Moreover, we can develop energy recovery systems and intelligent power management systems to extend flight time and increase the number of times that the drone is tested with one charge.

In summary, this project successfully develops an autonomous quadcopter system capable of indoor navigation, object detection, and package retrieval. The integration of hardware and software components, along with the modular design approach, enables the drone to perform complex tasks with a high degree of autonomy and reliability.

References

[1] S. Kumar and H. Gupta. Vision-based navigation for indoor uavs: A survey. *Journal of Intelligent and Robotic Systems*, 110(1):1–25, 2024.

[2] Y. Chang and T. Kim. Visual-inertial fusion for autonomous drone landing using aruco markers. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1432–1439, 2022.

[3] A. Lopez and M. Garcia. A review of pid control schemes for quadcopters. *Control Engineering Practice*, 138:105397, 2023.

[4] D. Morales and E. Navarro. Ros 2 integration of companion computers with mavlink-based flight controllers. *Robotics and Autonomous Systems*, 167:104508, 2023.

[5] M. Delbene and F. Russo. Ultrasonic sensor fusion for low-altitude uav landing. *Sensors*, 22(3):1012, 2022.

[6] J. Hsiao and M. Buehler. Design of a passive gripper for aerial object retrieval. In *Proceedings of the 2022 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8945–8951, 2022.