ISSN: 1813-4890

# X-shaped Quadcopter Drone

Yang Li, Xiaohong Wang [a], Jie Wang, Xuehui Lü, Gaofeng Pan

College of Information engineering, Henan University of Science and Technology, Luoyang 471023, China.

[a]wxhong2006@163.com

## Abstract

**This paper describes a novel lightweight X-shaped quadcopter drone, specifically a model controlled by a remote control handle, including sensor calibration, frequency pairing, wireless communication, OLED display, attitude control, and altitude hold hovering. It can be precisely controlled for free flight within a low-altitude range of 100m. This paper summarizes and compares commonly used flight control algorithms, discussing their respective advantages, disadvantages, and application scenarios to facilitate selection for different situations. Four algorithms are included: complementary filtering, Mahony algorithm, Madgwick algorithm, and linear Kalman filtering, with quaternion calculations used in the process. The RT-Thread real-time operating system divides system tasks into multiple threads, facilitating task management and later maintenance. Simultaneously, all components, including the flight controller, ESC, and motors, are integrated into a single four-layer PCB board, resulting in a more compact and lightweight design, which is beneficial for debugging, learning, and mass production..**

## Keywords

**Quadcopter, Wireless remote control, Algorithm summary, RT-Thread, Component integration.**

## 1. Introduction

With the rapid development of microelectrome-chanical systems (MEMS) and automatic control technology, small unmanned aerial vehicles (UAVs) are increasingly widely used in aerial photography, monitoring, and personal entertainment. To achieve stable flight of UAVs in complex environments, high-performance IMU attitude calculation algorithms and stable power control systems are crucial. This paper, addressing independent development needs, designs a small quadcopter based on the STM32 architecture and the RT-Thread real-time operating system[4,8]. It reads data from nine-axis sensors and a barometer through a highly integrated sensor array, uses advanced quaternions to calculate attitude transformations, and summarizes common flight control algorithms on the market, discussing and comparing their advantages and disadvantages in detail to adapt to various complex scenarios and different design requirements[7]. An optimized cascade PID control algorithm solves problems such as susceptibility to interference and attitude drift in small UAVs[5]. Attitude control and altitude hovering achieve high-precision wireless control. The highly integrated and compact PCB design is more conducive to learning for beginners and mass production.

## 2. Overall System Design

The quadcopter drone consists of three parts: the Flight controller(FC)、the Electronic speed controller(ESC) and the remote control(RC), the main function can be achieved are:

ISSN: 1813-4890

1) Frequency Pairing: Establish stable and continuous communication between the drone and remote control.

2) Sensor Calibration: Calibrate the sensor to obtain accurate raw data on Acceleration,angular velocity,and magnetic field intensity.

3) Indicator lights: Different colors and flashing frequencies of lights are used to indicate the real-time status of different modules at various stages of drone and remote control operation.

4) wireless remote: Send various command to drone via remote control handle.

5) Attitude mode: The drone's attitude can be freely controlled ,including moving left、right、backward、forward and spin.

6) IMU attitude detection: The attitude of drone is calculated for PID control and fed back to the remote control handle.

7) Hovering at a constant altitude: Controlling the drone to fly stably at a specified altitude with its attitude remaining almost unchanged.

8) OLED display:Displaying modules of drone that running and sent commands in real time.

9) TYPE-C port: Design for charging, both drone and remote handle are set.

10) Serial port: Add two serial port printing information and facilitate debug.

Its system block diagram is shown in Figure 1.

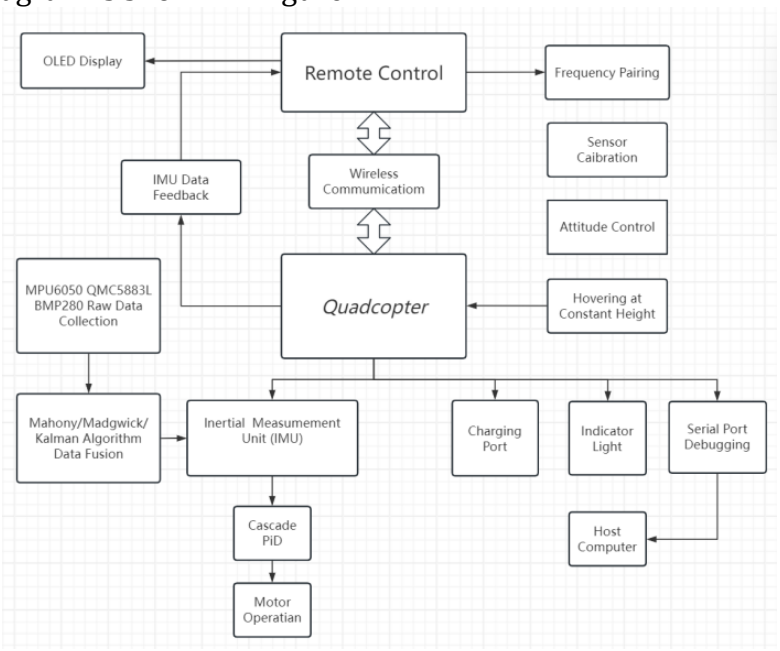


Fig. 1 System Block Diagram

## 3. Integrated Design of Quadrotor UAV Systems

### 3.1. Hardware Architecture and Flight Control System

The system is centered around the ARM Cortex-M3-based STM32F103C8T6 microcontroller, operating at the frequency of 72 MHz with 16 KB SRAM and 64 KB Flash memory. To ensure data persistence, critical parameters such as sensor calibration offsets and PID gains are stored in the terminal sectors of the Flash memory.

By leveraging the MCU's rich peripheral resources, a 16-bit advanced-control timer (TIM1) and multiple general-purpose timers generate high-frequency Pulse Width Modulation (PWM) signals. These signals drive enhanced MOSFETs (SI2302A) for precise motor speed regulation. Furthermore, an integrated 12-bit ADC module facilitates real-time battery voltage monitoring, with telemetry data transmitted back to the ground station handle for power management.

ISSN: 1813-4890

The physical prototype of quadcopter and circuit schematic are presented in Figure 2 and Figure 3, respectively.

Fig. 1 physical prototype

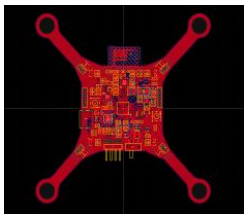

Fig. 3 circuit schematic

## 3.2. Attitude Estimation and Algorithms Fusion

### 3.2.1. Attitude Determination Module

The sensing layer incorporates a nine-axis inertial measurement unit (IMU) and a barometric altimeter:

Attitude Sensing: An MPU6050 six-axis gyroscope/accelerometer provides Pitch and Roll data, while a QMC5883L magnetometer compensates for the long-term drift of the Yaw axis.

Altitude Observation: A BMP280 high-precision barometer provides real-time altitude data with integrated temperature compensation.

### 3.2.2. Euler Angles And Quaternions

Traditional Euler angles use pitch, roll, and yaw angles in a fixed rotational order (e.g., Z-Y-X) to represent the real-time attitude of a drone. However, this has a fatal problem—singularity, also known as "gimbal lock." If the pitch angle reaches ±90° (vertically upward or downward), the original x-axis (Roll axis) of the drone coincides with the current z-axis (Yaw axis). At this point, the drone loses a degree of freedom to change its heading, causing the algorithm to crash and the drone to spin erratically. This makes continuous rolling and rotating functions almost impossible for the drone and also poses significant safety hazards. The advent of quaternions solves this problem.

Quaternions are a mathematical tool used to represent rotations in three-dimensional space, belonging to the hyper complex number system. They are the "standard language" for modern high-order attitude calculations (such as the Mahony algorithm and 3D graphics). A quaternion consists of one real part and three imaginary parts, as shown in Equations 1 and 2.

$$q = w + xi + yj + zk, \quad q = [w, v] \tag{1}$$

$$q = [q_0, q_1, q_2, q_3]^T \tag{2}$$

The quaternion q is used to represent the rotation of coordinate system b (body coordinate system) relative to coordinate system n (geographic coordinate system, usually NED or ENU). It satisfies the normalization condition $q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1$. It describes the rotation as “rotating by a specific angle θ about any defined axis (vector u) in space,” and completes the transformation from the body coordinate system to the geographic coordinate system through the matrix $R_n^b$. The relationship between the rate of change of attitude and the angular velocity $\omega = [\omega_x, \omega_y, \omega_z]^T$ is obtained using quaternion differential equations, as shown in equations 3 and 4.

$$R_n^b = \begin{bmatrix} 1-2(q_2^2+q_3^2) & 2(q_1q_2-q_0q_3) & 2(q_1q_3+q_0q_2) \\ 2(q_1q_2+q_0q_3) & 1-2(q_1^2+q_3^2) & 2(q_2q_3-q_0q_1) \\ 2(q_1q_3-q_0q_2) & 2(q_2q_3+q_0q_1) & 1-2(q_1^2+q_2^2) \end{bmatrix} \tag{3}$$

ISSN: 1813-4890

$$\dot{q} = \frac{1}{2} q \otimes \omega = \frac{1}{2} \begin{bmatrix} -q_1\omega_x - q_2\omega_y - q_3\omega_z \\ q_0\omega_x + q_2\omega_z - q_3\omega_y \\ q_0\omega_y - q_1\omega_z + q_3\omega_x \\ q_0\omega_z + q_1\omega_y - q_2\omega_x \end{bmatrix} \quad (4)$$

This approach completely solves the gimbal lock problem and is also more computationally efficient. Euler angle calculations involve numerous trigonometric function operations (sin, cos, atan[2]), which are very time-consuming on a microcontroller.

### 3.2.3. Complementary Filtering

Complementary filtering is the simplest and most intuitive method for processing MPU6050 data. Due to the complementary characteristics of the two sensors: the gyroscope is accurate in the short term but drifts in the long term; the accelerometer is greatly affected by vibration (high-frequency noise) in the short term but has stable angles in the long term. We obtain accurate data by fixing the trust weights of the gyroscope and accelerometer.as shown in equations 5.

$$\theta_n = \alpha \cdot (\theta_{n-1} + \omega \cdot \Delta t) + (1 - \alpha) \cdot a \quad (5)$$

Here, a represents the weight assigned to the gyroscope's angle, typically ranging from 0.95 to 0.98. A larger value indicates higher trust in the gyroscope, resulting in a faster system response, but also more severe integral drift. Conversely, a larger value indicates higher trust in the accelerometer, leading to a slower system response and more significant suppression of integral drift.

It's worth noting that complementary filtering essentially acts as a delay suppressor of gyroscope integral drift, but it doesn't truly solve the problem; it's essentially a low-end version of linear Kalman filtering. Both its response speed and long-term stability need improvement, therefore it's generally not used in UAV attitude estimation. It's only used as a reference to introduce the linear Kalman filter later.

### 3.2.4. Mahony Filtering

Compared to complementary filtering, the Mahony algorithm truly solves the zero-drift problem[1]. It uses gravity and magnetic force vectors as references, transforms the coordinate system to obtain a predicted data vector, and compares it with the actual gyroscope data for feedback correction. Its essence lies in the “cross product error” and “PI feedback.” The overall flowchart is shown in Figure 4:

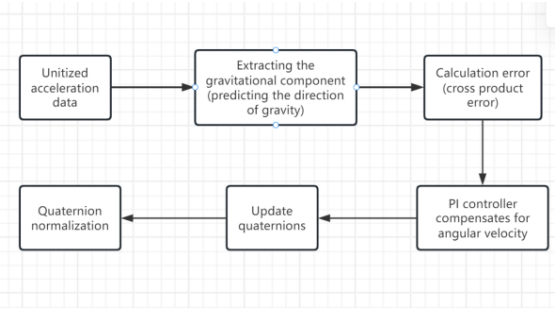


Fig. 4. Mahony Filter overall flowchart

The raw accelerometer data $[a_x, a_y, a_z]$ read by the MPU6050 contains the gravity vector. After normalizing it to $\vec{a} = \frac{a}{|a|}$, the gravity component (predicted gravity direction) $\vec{g} = [0,0,1]^T$T is extracted. Using the current quaternion estimate, we can transform it to the body coordinate system to obtain the "theoretical gravity direction (as shown in Equation 6) $\vec{v}$:

ISSN: 1813-4890

$$\vec{v} = (R_b^n)^T \cdot \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} = \begin{bmatrix} 2(q_1 q_3 - q_0 q_2) \\ 2(q_0 q_1 + q_2 q_3) \\ d_0^2 - d_1^2 - d_2^2 + d_3^2 \end{bmatrix} \tag{6}$$

Note: Here, $\vec{v}$ is the gravity vector of the system based on the attitude of the previous moment.

The accelerometer measures the real-time direction of gravity, $\vec{a}$. The deviation between $\vec{a}$ and $\vec{v}$ is the attitude error. We use the cross product of vectors to measure this error, as shown in Equation 7.

$$\mathbf{e} = \vec{a} \times \vec{v} = \begin{bmatrix} e_x \\ e_y \\ e_z \end{bmatrix} = \begin{bmatrix} a_y v_z - a_z v_y \\ a_z v_x - a_x v_z \\ a_x v_y - a_y v_x \end{bmatrix} \tag{7}$$

Next comes the most crucial step in the Mahony algorithm. It uses the previously calculated error e to correct the gyroscope's angular velocity $\omega_{gyro}$. Where: the proportional term (P) corrects the current attitude deviation; the integral term (I) eliminates the gyroscope's zero-point drift (Bias). The corrected angular velocity (as shown in Equations 8 and 9) is:

$$e_{int} = e_{int} + e \cdot \Delta t \tag{8}$$

$$\omega_{corr} = \omega_{gyro} + K_p \cdot e + K_i \cdot e_{int} \tag{9}$$

Kp controls the compensation velocity, and Ki controls the elimination of drift.

The attitude is updated using the corrected angular velocity $\omega_{corr}$ according to the quaternion differential equation (as shown in Equation 10):

$$q_{new} = q_{old} + \dot{q} \cdot \Delta t \tag{10}$$

Specifically, for each component (as in Equations 11 and 12):

$$q_0 = q_0 + \frac{1}{2}(-q_1\omega_x - q_2\omega_y - q_3\omega_z)\Delta t \tag{11}$$

$$q_1 = q_1 + \frac{1}{2}(q_0\omega_x + q_2\omega_z - q_3\omega_y)\Delta t \tag{12}$$

Finally, since numerical computation (integration and rounding errors) can destroy the normalization property of quaternions, normalization must be performed (as shown in Equation 13):

$$q = \frac{q}{||q||} \tag{13}$$

The above process uses the gravity vector $\vec{g}$ as the reference direction to correct the gyroscope values. Since the gravity vector is vertically downwards, its projection in the sensor coordinate system does not change with Yaw. This makes it impossible to correct for Yaw[6] (yaw angle). Next, the geomagnetic vector $\vec{m}$ is introduced into the calculation to provide an absolute Yaw direction reference.

It is worth noting that because the Earth's magnetic field has not only a horizontal component but also a vertical component (magnetic inclination), and the vertical component varies in different regions, we cannot simply assume that the magnetometer points directly forward.

First, the measured values are projected onto the “geographic coordinate system” and an “ideal reference vector” is constructed. Finally, the system is switched back to the “body coordinate system” for cross-product correction to correct attitude errors.

### 3.2.5. Madgwick Filtering

The Madgwick algorithm, proposed by Sebastian Madgwick in 2010, is renowned for its low computational cost and excellent performance at low sampling rates. It is one of the most

ISSN: 1813-4890

mainstream algorithms for processing MPU6050 (accelerometer + gyroscope) data in self-balancing scooters, drones, and wearable devices[2].

Essentially, both Madgwick and Mahony use the direction of gravity observed by the accelerometer as a reference to correct the drift error caused by gyroscope integration. The difference lies in the filtering mechanism: Mahony uses a complementary filtering approach, introducing a proportional-integral (PI) controller for feedback control (to eliminate errors). Madgwick, on the other hand, performs numerical optimization (finding the minimum value of the function), essentially shifting the current quaternion one step towards the direction where gravity is aligned, much like moving downhill.

The overall flowchart is shown in Figure 5.

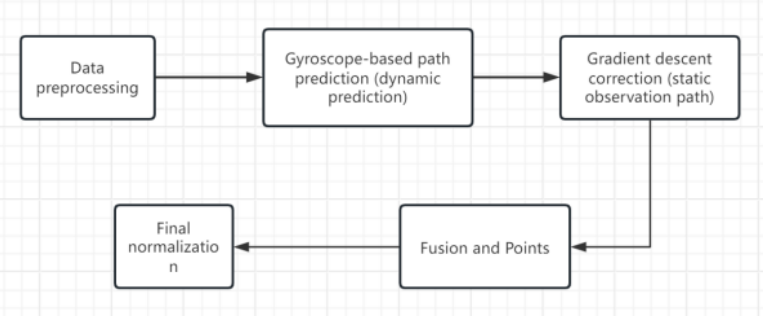


Fig. 5. Madgwick Filter overall flowchart

The gyroscope integral dynamic prediction, quaternion representation, and accelerometer conversion to theoretical gravity vectors are the same as in Mahony, so I won't go into detail here. I'll only explain the core of gradient descent correction: Define an objective function (error function):

$$f(q,a) = h(q) - a \tag{14}$$

To minimize this error, gradient descent is introduced. The direction of the attitude gradient is defined as:

$$\nabla f = J^T(\mathbf{q}) f(\mathbf{q},\mathbf{a}) \tag{15}$$

Where $J^T$ is the Jacobian matrix, i.e., the partial derivative of f with respect to q. The calculated gradient $\nabla f$ represents the degree to which the current attitude deviates from the direction of gravity. The corrected quaternion rate of change $\dot{q}_\nabla$ is:

$$\dot{q}_{\epsilon,t} = \frac{\nabla f}{||\nabla f||} \tag{16}$$

Weighted merging:

$$\dot{q}_t = \dot{q}_{\omega,t} - \beta \dot{q}_{\epsilon,t} \tag{17}$$

Here, β is a crucial gain parameter: a larger β indicates greater trust in the accelerometer, weaker resistance to vibration interference, but faster deviation correction. A smaller β indicates greater trust in the gyroscope, resulting in smoother attitude control, but long-term drift is possible. The final quaternion iteration formula is:

$$\mathbf{q}_t = \mathbf{q}_{t-1} + \dot{\mathbf{q}}_t \Delta t \tag{18}$$

Of course, $\mathbf{q}_t$ must be normalized at the end to ensure the validity of its rotation.

### 3.2.6. Linear Kalman Filter

The Linear Kalman Filter (LKF) is a mathematical algorithm based on “prediction + update” to estimate the closest approximation to the true state from noisy data. Its cleverness lies in the fact that its weights (K) are automatically calculated based on P and R, rather than being hard-coded manually[3]. Compared to complementary filters (fixed trust weights), Madgwick (fixed

ISSN: 1813-4890

PI feedback weights), and Mahony (fixed β gyroscope drift weights), it is more flexible and accurate, but computationally more complex.

It assumes all errors are Gaussian white noise and dynamically adjusts the weights (Kalman gain) of the physical model's predictions and sensor measurements to calculate the optimal estimate. Compared to the UKF (Extended Kalman Filter), it is linear, representing a single-dimensional rotation. It does not require quaternions and extensive matrix operations, resulting in extremely fast operation, making it ideal for microcontrollers (such as STM32 and Arduino). It excels in precisely controlling the attitude and angles of drones, but suffers from gimbal lock-up.

The overall flowchart is shown in Figure 6:

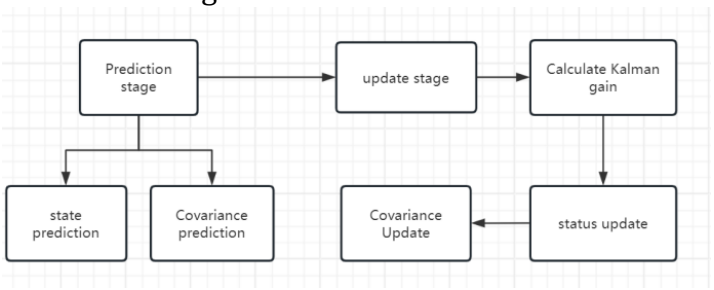


Figure 6. Kalman Filter overall flowchart

Prediction Phase: This phase does not rely on external measurements; it derives the current state entirely from physical laws and the state at the previous moment.

State Prediction Equation:

$$\hat{x}_{k|k-1} = F\hat{x}_{k-1|k-1} + Bu_k \tag{19}$$

In MPU6050: Angle = Old Angle + (Current Angular Velocity - Zero Bias) × Δt. This step is a purely mathematical derivation. Due to gyroscope noise, this calculated result (prior estimate) will accumulate errors over time.

Covariance Prediction Equation:

$$P_{k|k-1} = FP_{k-1|k-1}F^T + Q \tag{20}$$

Calculates the propagation of "uncertainty." P represents our "confidence" in guessing the current state. Even if we were very certain at the previous moment ( $P_{k-1}$ was small), due to the imperfections of the prediction model itself and the presence of process noise (Q), the uncertainty P increases slightly with each prediction. The larger Q is, the higher the distrust of the gyroscope model.

Update Phase:

Calculate Kalman Gain:

$$K_k = \frac{P_{k|k-1}H^T}{HP_{k|k-1}H^T + R} \tag{21}$$

P represents the noise of the predicted value (model uncertainty). R represents the noise of the measured value (sensor jitter). If R≫P, the sensor noise is large (e.g., during vibration), and K approaches 0. In this case, the system almost ignores the measured value and tends to trust the prediction. If P≫R, the prediction error is large (e.g., gyroscope drift), and K approaches 1. In this case, the system almost ignores the predicted value and tends to trust the measured value.

State Update Equation:

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k(z_k - H\hat{x}_{k|k-1}) \tag{22}$$

Correcting the error, ( $z_k - H\hat{x}_{k|k-1}$) is the measurement margin. The final result = prediction result + K×error.

ISSN: 1813-4890

Covariance Update Equation:

$$P_{k|k} = (I - K_k H)P_{k|k-1} \quad (23)$$

Updates the confidence index. Because external measurements are introduced for correction, the system is now "more confident" in its state estimate than in the prediction stage. $P_{k|k}$ will decrease, serving as the starting point for the next cycle.

### 3.2.7. Summary

In summary, each of the four filtering algorithms has its own advantages and disadvantages. For toy/educational drones, complementary filtering is recommended. It is simple to develop and has low hardware costs. For applications requiring extremely high speed and low latency (such as racing drones), the Mahony algorithm is the preferred choice. It allows the control loop to operate at frequencies of several kilohertz while maintaining accuracy. For consumer-grade aerial photography drones and long-range cruise drones that require precise hovering and stable flight but do not need such high control frequencies, the Madgwick algorithm is recommended. For industrial drones, swarm aircraft, and other drones requiring automated flight paths and high-precision mapping, the linear/extended Kalman EKF is the best choice, as its extremely high accuracy and response speed can meet most situations and needs.

A comprehensive comparison of the four discussed algorithms is summarized in Table 1.

Table 1 A comprehensive comparison of the four attitude estimation algorithms

| Feature | Complementary Filter | Mahony Filter | Madgwick Filter | Kalman Filter (LKF) |
|---|---|---|---|---|
| Core Algorithm | Weighted Average | PI Control + Complementary | Gradient Descent | Minimum Variance Estimation |
| Computational Complexity | Extremely Low | Low | Medium-Low | High |
| Noise Immunity | Poor | Good | Excellent | Superior (with proper tuning) |
| Convergence Speed | Fast | Extremely Fast | Moderate | Depends on Initial Values |
| MCU it is sufficient | Requirements 8-b | 32-bit / FPU recommended | 32-bit /FPU recommended | High Clock Speed + FPU |

## 3.3. Cascade PID control

A cascade PID control architecture is implemented to ensure flight stability.Outer Loop (Angular Position) Employs a proportional (P) controller to translate angular deviation into target angular velocities for the inner loop.On this basis, Inner Loop (Angular Rate) Utilizes a full PID controller, where the integral (I) term eliminates steady-state errors and the derivative (D) term suppresses high-frequency oscillations.

Compared to conventional single-loop or full-state PID controllers, this cascaded approach provides faster dynamic response, superior disturbance rejection (especially against wind), and enhanced error-correction robustness.

## 3.4. Wireless communication module

Wireless communication module utilizes SI24R1 chip(Domestic compatible version of NRF24R1). Operating in the 2.4G band with 2Mbps data rate,+7dBm output power and theoretical effective outdoor distance 100m. The software implements a circular buffer to

ISSN: 1813-4890

optimize memory allocation and employs CRC (Cyclic Redundancy Check) to ensure data integrity. A "heartbeat" mechanism is maintained post-pairing to prevent link disconnection.

The WS2812B RGB LED, powered by a 5V DC rail, is incorporated as a real-time visual feedback interface. Changing into different colors for unlocking, locking, calibration, frequency matching, low power, and communication interruption, which greatly facilitates debugging.

### 3.5. Power System

The TC3608 chip is employ to stablize the battery voltage into 5V supplying for WS2812B .Currently, the ME6206 chip declines 5V to 3.3V for other system components.The TP4056 charging circuit integrates a Type-C interface with backflow protection to enhance electrical safety.

### 3.6. RT-Thread real-time operating system

Comparing with traditional super-loop paradigm, this system is built upon the RT-Thread real-time operating system (RTOS). By partitioning functionalities—such as attitude estimation, control logic, and communication into independ- ent concurrent threads, the architecture effectively resolves the issues of high data coupling and global variable proliferation. This modular design significantly enhances determinism and real-time performance, facilitating system maintainability and simplified troubleshooting.

## 4. Design of the Remote Control Terminal

The remote controller serves as the ground control station (GCS) of the UAV system, also built upon the STM32F103C8T6 MCU and the SI24R1 RF module for robust bidirectional data exchange.

For signal acquisition, the system employs four 12-bit ADC channels to sample analog joystick inputs, which are subsequently digitized and mapped into specific throttle and attitude control parameters. The Human-Machine Interface (HMI) features a 1.2-inch OLED display for real-time telemetry visualization, supplemented by four tactile switches. These buttons facilitate menu navigation, command execution (Confirm/Cancel), and system arming/disarming.

At the operational level, the terminal enables users to initiate sensor calibration, link pairing, and IMU data requests. Real-time feedback from each operational phase is transmit-Ted back to the controller and displayed on the OLED screen. In Attitude Control Mode, the mechanical displacement of the dual joysticks is linearly mapped to Throttle, Yaw, Pitch, and Roll values. Specifically, the Pitch, Roll, and Yaw parameters are normalized to a range of 0–100, while the Throttle maintains a higher resolution of 0–1000 for precise altitude regulation.

The functional flowchart and the physical prototype are illustrated in Figure 7 and Figure 8, respectively.

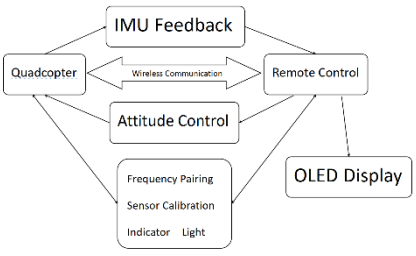


Fig. 7 functional flowchart

Fig. 8 physical prototype

ISSN: 1813-4890

## 5. Conclusion

This paper describes a lightweight quadcopter drone that achieves functions such as frequency matching, calibration, and attitude control flight. It also features wireless control via a remote control handle and an OLED display, significantly reducing the difficulties of the debugging phase. Furthermore, it summarizes the most common and classic flight control algorithms, allowing for flexible selection in various situations, making it very convenient for beginners. Due to the device's high level of integration, the purchase of components and the soldering process are also more convenient. If mass-produced and deployed in social production, it will undoubtedly generate a very positive response.

## Acknowledgements

This work was financially supported by the Natural Science Foundation of Henan Provincial Department of Science and Technology under Grant No. 252300421807, and the research results of Henan University of Science and Technology 2025 College Students' Innovation and Entrepreneurship Training Program (2025125).

## References

Xu, E. S., Lu, W. H., Liu, Y. F., et al. (2019). Attitude calculation and application research based on Mahony filter algorithm. *Intelligent Computer and Application*, *9*(5), 80–83.

Shi, J. K., Zhang, S. Y., Jian, K. W., et al. (2024). A satellite attitude measurement method based on Madgwick-EKF fusion algorithm. *Shanghai Aerospace (Chinese and English)*, *41*(2), 95–103, 120.

Zhao, J. L., Li, S. Z., & Zhao, Z. L. (2023). UAV attitude algorithm based on extended Kalman filter. *Journal of North China University (Natural Science Edition)*, *44*(3), 305–310.

Ren, J. Q., Zhong, X. Y., & Zhang, X. H. (2021). Design of multi-sensor quadrotor attitude control system based on STM32. *47*(5), 97–101, 107.

Dai, L. J., Xu, Y. H., Xu, Z. Y., et al. (2025). Design of a quadrotor UAV control system based on cascade PID algorithm. *China New Technologies & New Products*, (12), 8–10.

Sun, L. (2019). Research on magnetic declination adaptive algorithm in geomagnetic navigation of quadrotor UAV. *Aviation Weapons*, *26*(4), 52–57.

Zhou, J. X., Chen, W., Chen, X., et al. (2025). Integrated modeling and control of pose of quadrotor UAV based on torque. *Electro-Optics and Control*, *32*(4), 8–16.

Zhang, W., Wang, K., & Wang, C. (2021). Design and implementation of UAV based on STM32. *Computer and Digital Engineering*, *49*(9), 1919–1923.

Biographies

Xiaohong Wang(1985.04-), female, Han nationality, born in Luoyang city, Henan Province, Ph. D., lecturer. Her research interests include nonlinear system stability and control applications.