API Description File Retrieval Outline

1. RflySim Introduction to core components of the platfor m

RflySim platform includes many software involved in the development process of unmanned system modeling, simulation and algorithm verification. The core components include CopterSim, QGroundControl, RflySim3D/RflySimUE5, Python38Env, WinWSL subsystem, SITL/HITLRun one-click running script, MATLAB automatic code generation toolbox, Simulink cluster control interface, PX4 Firmware source code, RflySim supporting data file and supporting hardware system. By learning these core components, users can quickly start the development and testing of unmanned systems.

Figure1Interrelationship between RflySim hardware and software components and the overall process

1.1. CopterSim

CopterSim is one of the core software of RflySim platform, which is a hardware-in-the-loop simulation software developed for Pixhawk/PX4 autopilot platform. It can configure multi-rotor model in the software and connect with Pixhawk autopilot through USB serial port to realize hardware-in-the-loop simulation. The effect of indoor simulation and outdoor flight test is achieved. It mai nly consists of two parts — model and communication. The model means that the simulation can be carried out directly after calculation according to the set model parameters; it also supports running the dynamic model (DLL), and forms the software/hardware in the loop simulation together with other software. CopterSim is the center of all data communication; the flight control and CopterSim are connected through serial port (HITL) or network TCP/UDP (SITL), and MAVLink is used for data transmission to realize closed-loop control and simulate outdoor flight; CopterSim sends aircraft pose and motor data to the 3D engine for visual display; forwards MAVLink messages to Python Vision or QGC ground station to transmit aircraft real-time status for top-level planning control; and so on. At the same time, CopterSim software compresses MAVLink data and sends it to the cluster control software in the form of UDP structure to achieve the purpose of communication simplification (large-scale cluster requirements).

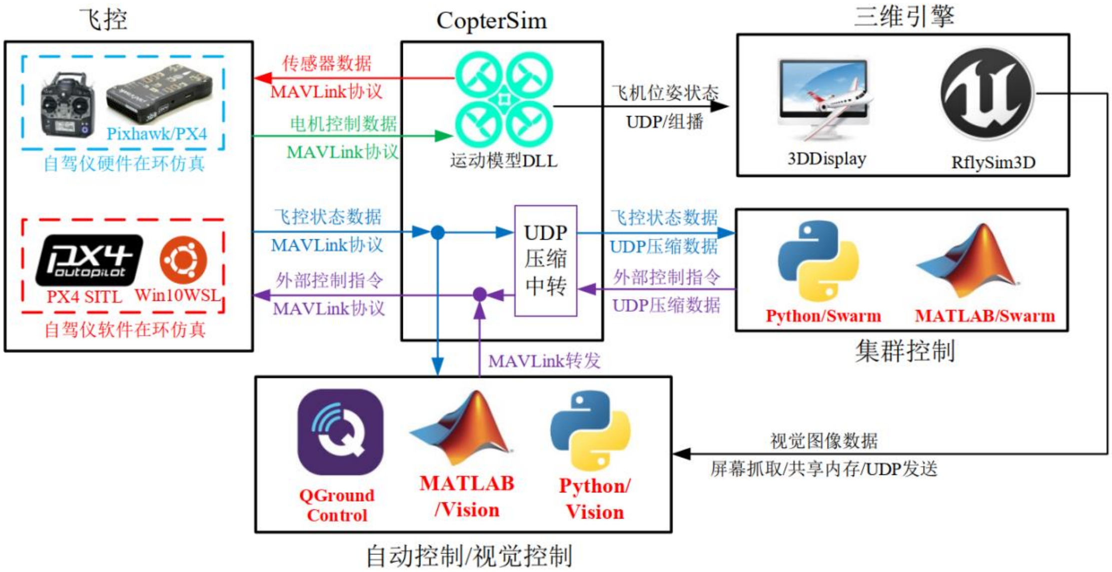


Figure2Data communication structure diagram of CopterSim software

The main interface of CopterSim is mainly divided into three parts: model configuration area, simulation function area and status display area, as shown in the following figure.

1.1.1. Model configuration area

The configuration, size, weight and other data of the customized multi-rotor can be configure d, and CopterSim will calculate the parameters of the specified multi-rotor model to realize the sim ulation of different models.

1.1.2. Simulation ribbon

Support the configuration of aircraft ID, communication interface, simulation mode, 3D scen e, distributed online simulation, map initial position, flight control COM serial port selection, com munication mode, etc.; at the same time, control the start, pause and restart of simulation;

- **Local ID: the label of each aircraft.**
- **UDP receiving port: Simulink/Python and other external programs need to send da ta to this port and return data from port + 1.**
- **Use DLL model file: This interface is used to select different DLL files. The folder a ddress corresponding to the option is: * PX4PSP \ CopterSim \ external \ model**
- **Simulation mode: used to select different simulation modes, which are mainly divid ed into the following types:**

PX4 _ HITL: This mode is the official hardware-in-the-loop emulation mode of PX4.

PX4 _ SITL: This mode is the official PX4 soft-in-the-loop simulation mode.

PX4 _ SITL _ RFLY: This mode is a software-in-the-loop simulation mode customized by the R flySim platform. Since PX4 _ SITL only supports cluster simulation of 10 UAVs, this mode can s upport SITL simulation of hundreds of UAVs.

Simulink & DLL _ SIL: The simulation mode of the DLL file generated in Simulink.

PX4 _ HITL _ NET (limited to complete version or above): For flight control with network port such as Pixhawk 6x, it supports hardware-in-the-loop simulation through LAN connection fl ight control.

PX4 _ SIH _ NET (limited to the full version or above, under development): support the SI H mode of flight control (the model runs in the flight control to reduce the amount of compute r computation and improve stability), and.NET means connecting to the flight control through t he LAN.

PX4 _ SIH _ COM (limited to complete version or above, under development): Support SIH mo de of flight control (the model runs in the flight control to reduce the amount of computer calculation and improve the stability), COM means connecting the flight control through USB cable.

JSON _ HITL _ COM (limited to the full version or above, under development): Support the connection of the third party flight control through the JSON protocol to realize the hardware-in-the-loop simulation, com means to connect the flight control through the USB serial port.

JSON _ HITL _ COM (limited to the complete version or above, under development): Support the connection of the third-party flight control through the JSON protocol to realize the hardware-in-the-loop simulation.NET means to connect the flight control through the network (or connect the third-party flight control of SITL).

APM _ SITL _ NET (limited to the full version or above, under development): Support the connection of APM flight control (Ardupilot) for SITL software in the loop simulation.NET means that CopterSim is connected to the flight control through the network.

FPGA _ HITL (limited to professional version or above, under development): Support the way of replacing the flight control hardware with FPGA to perform hardware-in-the-loop simulation.

Note: In the large-scale cluster hardware-in-the-loop, the computer serial port and power supply (COM mode) will affect the number of flight controllers connected to each computer. It is recommended to use the NET suffix mode.

- **3D display scene: used to select the 3D display scene in RflySim3D. The folder address corresponding to this option is: * PX4PSP \ CopterSim \ external \ map**
  Note: The terrain file contains a two-dimensional elevation image of.png (uint16 grayscale map) and a.txt calibration file (describing the pixel range corresponding to XY width, 0-uint16 _ Max corresponding to height range, etc.).
- **Online: When this check box is clicked, online emulation within the LAN will be enabled. Note: Check this button and press the I key in RflySim3D/RflySimUE5 to enable LAN online simulation, which supports networking of multiple computers to display all aircraft.**
- **Aircraft starting position: the initial position XY and Yaw angle of the aircraft can be set.**
- **Flight control selection: This window is only used in the hardware-in-the-loop simulation phase to select the flight control inserted into the simulation computer.**
- **UDP Mode: Select different UDP communication modes, mainly including the following.**

UDP _ Full: Python transmits complete UDP data to CopterSim. The amount of data transmitted is small. After receiving the data, CopterSim converts it into Mavlink and transmits it to PX4 flight control. It is suitable for simulation of small and medium-sized clusters (less than 10).

UDP _ Simple: The packet size and sending frequency are smaller than UDP _ Full mode; it is suitable for large-scale cluster simulation, and the number of UAVs is less than 100.

Mavlink _ Full: Python directly sends the MAVLink message to CopterSim, and then forwards it to PX4. The message has a large amount of data and is suitable for single machine control. It is suitable for single machine or a small number of aircraft simulation. The number of UAVs is less than 4;

Mavlink _ Simple: shield part of the MAVLink message packets, and reduce the data frequency, the amount of data sent is much smaller than the MAVLink _ Full, suitable for multi-machin

```
e cluster control; suitable for small-scale cluster simulation, the number of UAVs is less than 8.
    Mavlink _ NoSend: CopterSim will not send out MAVLink data in this mode. This mode requires hardware-in-the-loop simulation + data transmission serial communication. MAVLink is transmitted through wired mode. This mode has the smallest amount of data in LAN and is suitable for distributed vision hardware-in-the-loop simulation. There is no limit on the number of UAVs.
    Mavlink _ NoGPS: CopterSim will not send out MAVLink data and GPS data in mode. Note: This mode is suitable for flight control SLAM simulation. The visual perception algorithm needs to send the mavlink message of visual positioning to the flight control to form the visual control algorithm verification in the environment without GPS.
```

- **Start Simulation/Stop Simulation/Re-simulate.**

1.1.3. Status display area

The left side shows the model and Pixhawk return status, and the right side shows the simulation data of the model. A representative small lab to introduce the import functionality of the DLL model.

1.1.4. Emulation control interface Re qCopterSim.py

The CopterSim computer IP can be obtained through the Python interface, and the command is sent to reset the CopterSim online. The key functions are as follows:

```
    def  sendReCopterSim(self,CopterID=1,isReqIP=1,UDP_mode=-1,isXyYaw=0,xyYaw=[0,0,0],isZRP=0,zRollPitch=[0,0,0], otherParams=[0,0,0,0]):
    Explain
    Check sum: Data check bit 1234567, used to check whether the data is correct
    CopterID: The CopterID of the request, which is used to verify whether the request is correct.
    IsReqIP: If < = 0, no response will be made; if > 0, online will be checked, and the flight control data will be sent to the computer IP (the computer sending this request message).
    IsXyYaw: does not respond if < = 0, redeploys the aircraft position if > 0 using the following value of xyYaw
    IsZRP: If < = 0, it does not respond (Z will fit the terrain by default, and rollPitch will become 0). If > 0, the value of zRollPitch is used to deploy the aircraft pose. This interface allows the aircraft to be initialized in the air.
    UDP _ mode: Do not respond if < 0. If > = 0, modify the UDP mode to the specified value.
    OtherParams [4]: reserved.
    ZRollPitch [3]: the initial value of zRollPitch in meters and degrees, with Z positive downward
    Yaw: initial yaw angle in degrees
    XY [2]: value of initial XY, supporting double-precision large map, unit: m, NE
```

```
    def sendReDllMap(self,CopterID=0,dllOrMap=-1,index=-1,name=''):
    Explain
    Check sum: check code, used to confirm the correctness of data, the value here is 1234567895.
    CopterID: target aircraft ID, if 0, no response; if -1, broadcast to all aircraft; if > 0, to the specified aircraft.
    Flag: switch options: if < = 0, no response; if = 1, modify the DLL model; if = 2, modify the map;
    Index: option ID ordinal; does not respond if < 0, uses name to identify option; if > = 0, uses ordinal instead of name to identify option.
    Name [48]: The name of the DLL model or map
```

The relevant routines of the software are:

1.BasicExps\e1_CopterSim-Usage\Readme.pdf

1. BasicExps\e2_DLL-Load\Readme.pdf

1. BasicExps\e14_Log-Get\Readme.pdf

..\Git\2.RflySimUsage\0.ApiExps\e3_ReqCopterSim\Readme.pdf

1.2. RflySim 3 D /RflySim U E5

Unreal Engine has a powerful graphics engine that supports high-quality 3D graphics and visual effects; the built-in blueprint visualization scripting system allows developers to create complex logic and interactive behaviors in a graphical way without writing code; It has a huge community support and resource library, including models, textures, sound effects, plug-ins, etc., which can help developers speed up the development process and improve the quality of models; it supports multiple platforms, including PC, mainframe, mobile devices and virtual reality devices, etc.; Developers can customize and extend the functions and tools of the engine according to their own needs, making Unreal Engine suitable for various types of game and application development.

RflySim3D/RflySimUE5 is a highly realistic simulation software for unmanned system based on Unreal Engine, which inherits the advantages of Unreal Engine and communicates with other software on the platform in the form of UDP to realize highly realistic simulation of unmanned system. Visual image data can be transmitted to QGroundControl, MATLAB, Python and other software by means of screen capture and shared memory to realize visual algorithm verification simulation of unmanned system, as Figure 3shown in.

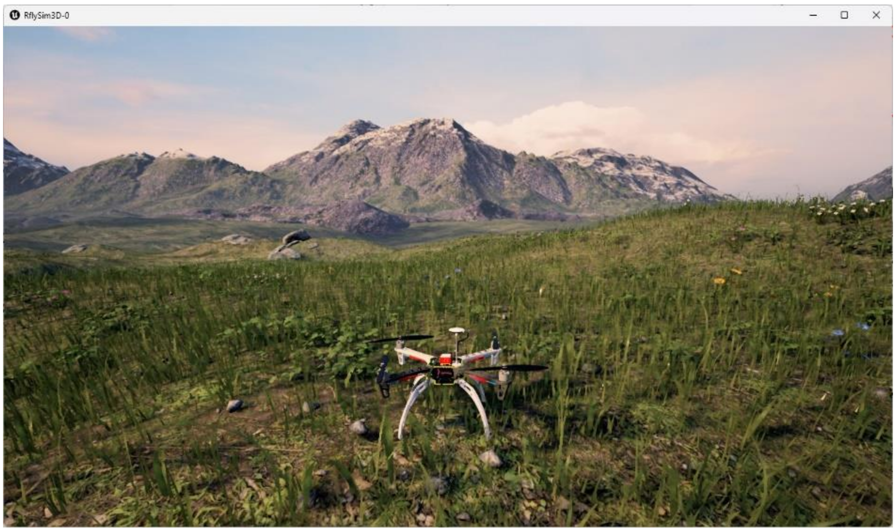


Figure 3RflySim3D/RflySimUE5 display main interface

At the same time, for users with low computer configuration, RflySim platform provides two other three-dimensional simulation software, namely FlightGear and 3D Display. FlightGear's team of developers from all over the world, including programmers, pilots, physicists and aircraft manu facturers, provides many different types of aircraft models and scenarios, including various civil an d military aircraft models, as well as many different scenarios and environmental simulations. It is a very popular open source flight simulator software, which can receive the flight state sent by Sim ulink through UDP, and easily observe the flight state of the aircraft during Simulink simulation. 3 DDisplay is a virtual flight simulator software developed by the Reliable Flight Control Research Group of Beihang University, which provides three-dimensional models and virtual environments, and supports a variety of aircraft models and scenes. Users can freely switch RflySim3D/RflySim UE5, FlightGear and 3DDisplay simulation software according to the configuration of personal co mputers.

1.2.1. Shortcut key

• F1: help menu prompt pops up;
• ESC: Clear all aircraft
• S: show/hide aircraft ID;
• H: Hide/show all screen text;
• D: Display/hide the current aircraft data;
• M: Switch the map (close all CopterSim first);
• M + number *: switch to map No. *;
• B: switch the view focus between different aircraft;
• B + digit *: switch to aircraft No. *
• C: Switch the 3D style of the current aircraft;
• C + number *: switch to 3D style No. *;
• CTRL + C: Toggle all aircraft 3D styles
• P: Turn on the physical collision engine (it will collide with scene objects and the ground. This function only s upports the full version)
• V: Angle of view switching on the aircraft, 0: Following angle of view, 1: Forward camera, 2: Right camera, … ;
• V + number *: Switch to the * perspective
• N: switch to the aircraft God view, 0: follow the aircraft view (do not change the view angle with the aircraft a ttitude), 1: fix the ground view
Angle and always look at the current aircraft, 2: fixed to the north to look at the pe rspective, 3: fixed to the south, … ;
• N + Number *: Switch to God Perspective No. *
• Press and drag the left mouse button to switch the viewing angle; press and drag the right mouse button to switch the vertical YZ position of the viewing angle
• Mouse wheel: switch the horizontal X position of the viewing angle
• CTRL + mouse wheel: zoom all aircraft dimensions (easy to observe in case of multiple aircraft);
• ALT + mouse wheel: zoom the current view plane size
• T: Turn on or off the aircraft track recording function
• T + number *: turn on/change track thickness to *
Double mouse click: display the position, size, object and other information of the hit point. Note: After doubl e-clicking
Press the N key immediately to quickly switch the perspective to the double-click posit ion for easy object creation
• O + number *: An object (obstacle) with style ID "*" is generated at the mouse double click
• I: Enable LAN connection to support the display of other aircraft in the LAN. (Limited to the full version or ab ove)

1.2.2. Shortcut command

Press the "`" (prime) key below ESC in the upper left corner of the keyboard and enter the fol lowing commands to configure the scene:

• RflyShowText Time string time//displays the string string for time seconds
• RflyShowText string//Displays the string string for 5 seconds
• RflyChangeMapbyID ID//Switch to map ID
• RflyChangeMapby Name name//switch to the map with name
• RflyChangeView KeyCmd key num//emulates the keyboard shortcut Key + num (for example, B 1)
• RflyCameraPosAngAdd X y Z roll pitch yaw//Incrementally change the angle of view position attitude in met ers and degrees
• RflyCameraPosAng X y Z roll pitch yaw//Set the current visual angle position attitude in meters and degrees.
• RflyCameraFovD egrees degrees//Change the field angle of the current viewing angle in degrees
• RflyChange3D Model CopterID veTypes//Change the style of CopterID aircraft to veTypes
(This can be a serial number or a name, for example, RflyChange3D Model 100 Eric _ Walkin g)
• RflyChangeVehicle Size CopterID size//Change the size of the CopterID aircraft to size
• RflySetPosScale Scale//Change the scale of the aircraft's trajectory, which is available on oversized maps.
• RflyMoveVehiclePosAng CopterID isFitGround X y Z roll pitch yaw//Incrementally move CopterID
The position and attitude of the aircraft (in meters and degrees). If isFitGround is set t o 1, the aircraft will always be on the ground.
• RflySetVehiclePosAng CopterID isFitGround X y Z roll pitch yaw//set CopterID number aircraft
Position and attitude in meters and degrees. If isFitGround is set to 1, the aircraft will always be on the ground.
• RflyScanTerrainH xL yL xR yR H Interval//scan the terrain data to get elevation terrain text of PNG and txt
Piece, used to import CopterSim; the coordinate of the lower left corner of the map is xL yL, the coordinate of the upper right corner is xR yR, and the scanning height
Is H, the scanning interval is Interval, and the unit is meter.
• RflySetActuatorPWMs CopterID pwm1 ... Pwm8//Set the first 8 actuators of CopterID aircraft
• RflySetActuatorPWMsExt CopterID pwm9 ... PWM 24//Set bit 9 to bit 24 of CopterID aircraft
Actuator value.

```
Note: The above instructions can also be sent to RflySim3D via UDP, see Python interface mav.sendUE4Cmd ()
Note: In addition to the above RflySim platform commands, you can also input the commands that come with the UE, such as R. Setres.
```

1.2.3. Terrain Calibration File —***. t xt Get

The terrain calibration file records the size of the 3D scene and the GPS data. The meanings of the parameters of this document are as follows:

```
< upper right corner of the scene X (cm) >, < upper right corner of the scene y (cm) >, < upper right point of the scene Z (cm)), < lower left corner of scene X (cm ')', < lower left corner of scene y (cm ')', > lower left point of scene Z (cm ")", < any point in the scene X "(cm") ", < any point in the scene y < any point in the scene Z (cm) >, < GPS longitude (degree) >, < GPS latitude (degree) > and < GPS altitude (m) >
```

Among them, the three points XY in the three-dimensional scene are all positive, Z is positive upward, and the unit is centimeter. The purpose of the first two points is to confirm the range and central coordinates of the terrain. The coordinates of the third point can be selected by oneself. In theory, it is necessary to have a difference in height from the first two points as far as possible to correct the height scale. The last three bits are GPS longitude, latitude and altitude data, which will be sent to QGC and RflySim3D for map unification. GPS data is only applicable to the construction of global large scenes in Cesium.

How to get it:

```
//You can enter in the shortcut command bar of RflySim3D/RflySimUE5:
RflyScanTerrainH xL yL xR yR H Interval
//Scan the terrain data to obtain PNG and txt elevation terrain files, which are used to import CopterSim; the coordinates of the lower left corner of the map are xL yL, the coordinates of the upper right corner are xR yR, the scanning height is H, the scanning interval is Interval, and the unit is meter.
```

Usage: Copy the terrain elevation file * * *.png to * PX4PSP \ CopterSim \ external \ model, and open CopterSim to use it.

1.2.4. Terrain Elevation File — ***. png Get

The terrain elevation file is actually a two-dimensional matrix stored in the form of a picture, which contains the elevation map of the scene. Storing the matrix in PNG format can well realize the compression of the elevation matrix, which is convenient for saving space. This file does not contain the coordinate origin, zoom scale, scene range and other information, so a correction file is required. The RflySim platform uses the txt format to input the 9-dimensional array to input the correction information.

How to get it:

```
//You can enter in the shortcut command bar of RflySim3D/RflySimUE5:
RflyScanTerrainH xL yL xR yR H Interval
//Scan the terrain data to obtain PNG and txt elevation terrain files, which are used to import CopterSim; the coordinates of the lower left corner of the map are xL yL, the coordinates of the upper right corner are xR yR, the scanning height is H, the scanning interval is Interval, and the unit is meter.
```

Usage: Copy the terrain calibration file * * *.txt to * PX4PSP \ CopterSim \ external \ model and open CopterSim.

The relevant routines of the software are:

1. BasicExps\e3_RflySim3D-Shortcut-Instruct\Readme.pdf

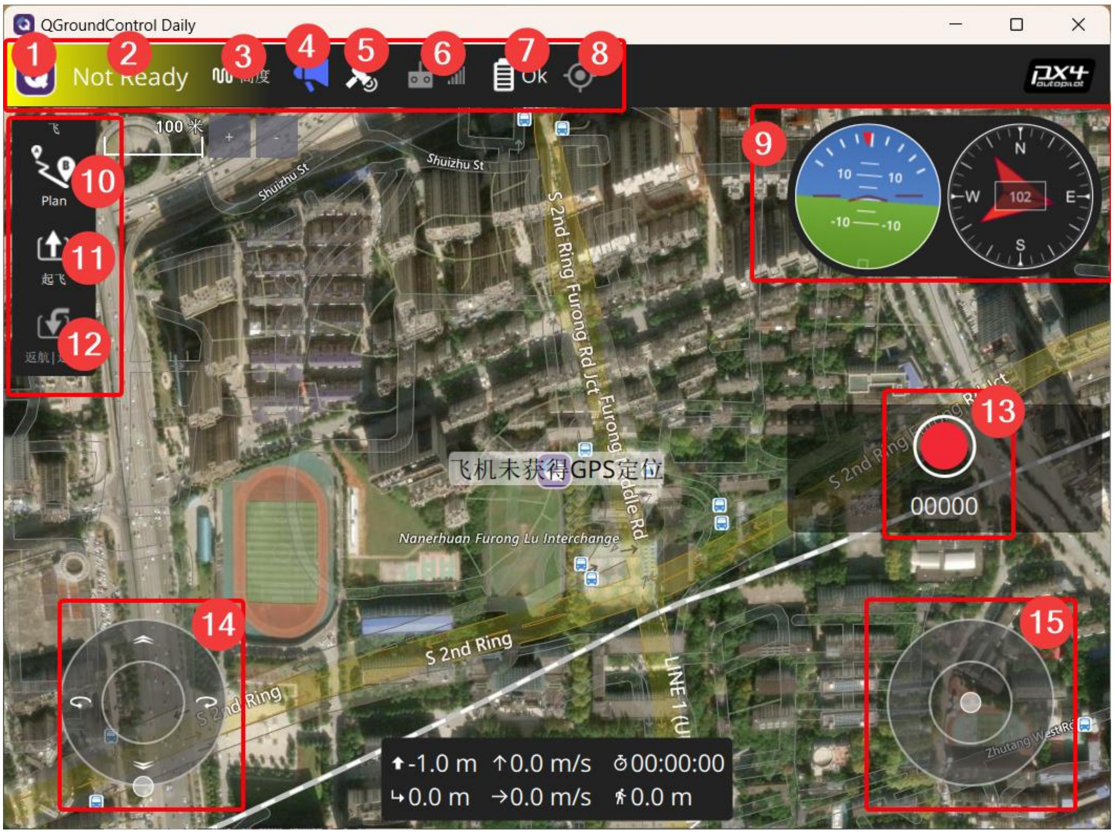
1. BasicExps\e15_Scene-Load\Readme.pdf

1.3. Q G roundControl Ground Station

The UAV ground station is a key component of the UAV application control system. The operator can operate the ground station through the mouse, touch screen and remote control handle to achieve the purpose of controlling the UAV. By setting the waypoint information and planning the route on the ground station, the UAV can fly according to the preset path and complete the waypoint task during the flight, including taking photos and aircraft actions. Video, etc. At present, the mainstream open source ground stations are QGroundControl and MissionPlanner, and QGroundControl is an open source ground station designed for the latest architecture of PX 4 software, which uses QT editor C + + language to write its core code, and supports source code modification and secondary development of functions. It is not only suitable for UAV ground station research and experiment, but also suitable for UAV ground station function customization and modification. In comparison, the advantages of QGroundControl are: 1) Open source: QGroundControl is a completely open source software, which means that users can freely modify and customize it according to their needs. 2) Ease of Use: The user interface is very clear, modern, and easy to use, allowing users to quickly perform mission planning and flight planning. 3) Multi-platform support: QGroundControl runs on multiple operating systems, such as Windows, Linux, and MacOS. 4) Modular architecture: QGroundControl's modular architecture makes it easy for developers to add and extend new functionality without compromising existing functionality and performance. Overall, QGroundControl is a modern, easy to use, open source and highly customizable ground station software, which has obvious advantages in multi-platform support, multi-language support, modular architecture and so on.

1.3.1. QGC Introduction to the initial interface

The overall interface of QGroundContrl is shown in the figure below, and the explanation of each button in the interface is as follows:



① Start button: this button can pop up a shortcut menu to enter the vehicle initialization settings, the use of analysis tools, and the relevant software property settings.

② Vehicle status display: Generally, the overall status of the vehicle can be quickly viewed from here.

③ Control mode selection: This button can switch different control modes, such as manual, self-stabilization, stunt, etc.

④ Notifications: Here you can view vehicle runtime information, such as warning messages, error messages, etc.

⑤ GPS status: displays the number of satellites that can be searched by the current vehicle.

⑥ Handle link status display.

⑦ Battery level display.

⑧ ROI region identification.

⑨ IMU status real-time dashboard.

⑩ Route planning.

⑪ Take off button.

⑫ Return button.

⑬ Record button: QGC interface video can be recorded.

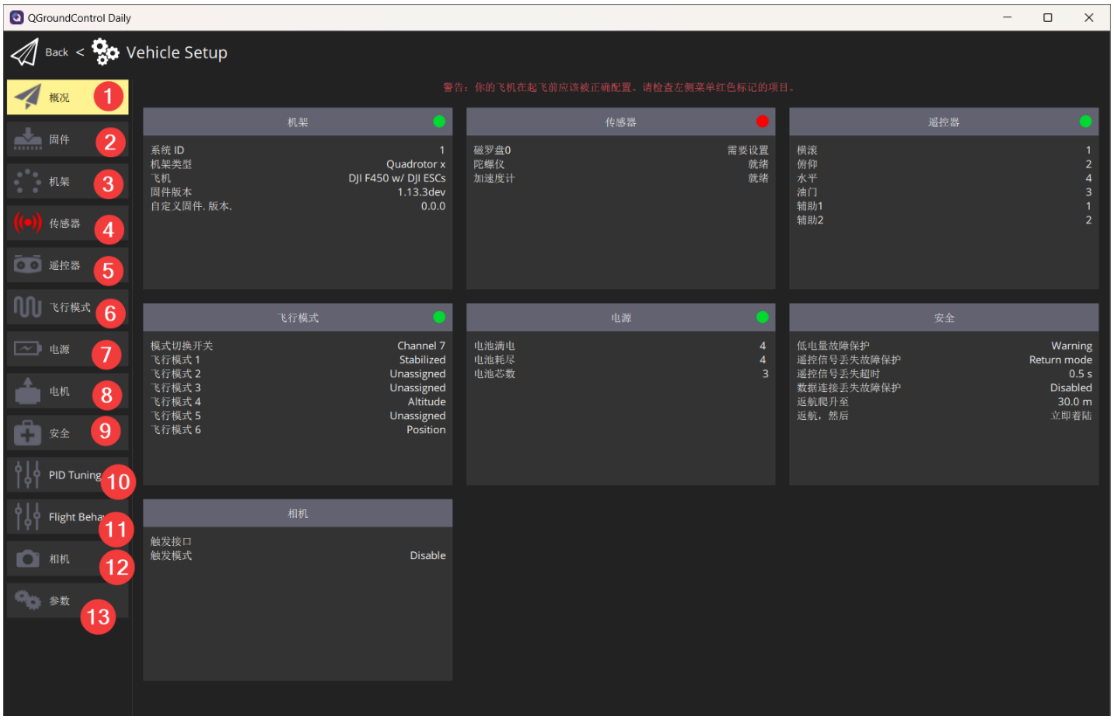
⑭ Virtual handle CH3/CH4 channel.

⑮ Virtual handle CH1/CH2 channel.

1.3.2. Carrier setup （ Vehicle Setup ）

The related key commands that can be seen in the initial interface of QGC ground station are as follow

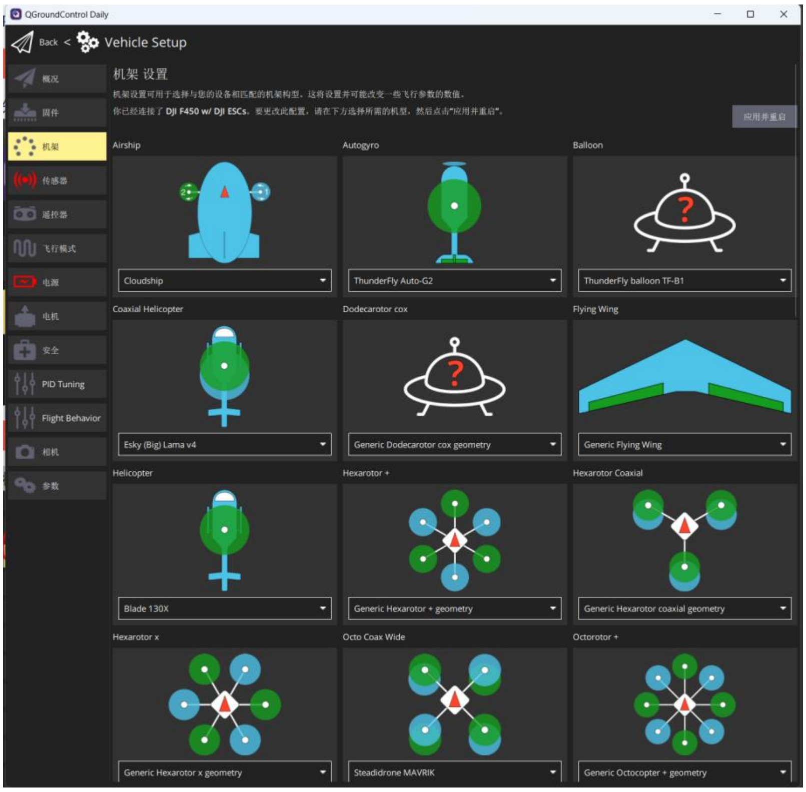
https://docs.qgroundcontrol.com/master/en/qgc-user-guide/getting_started/quick_start.html



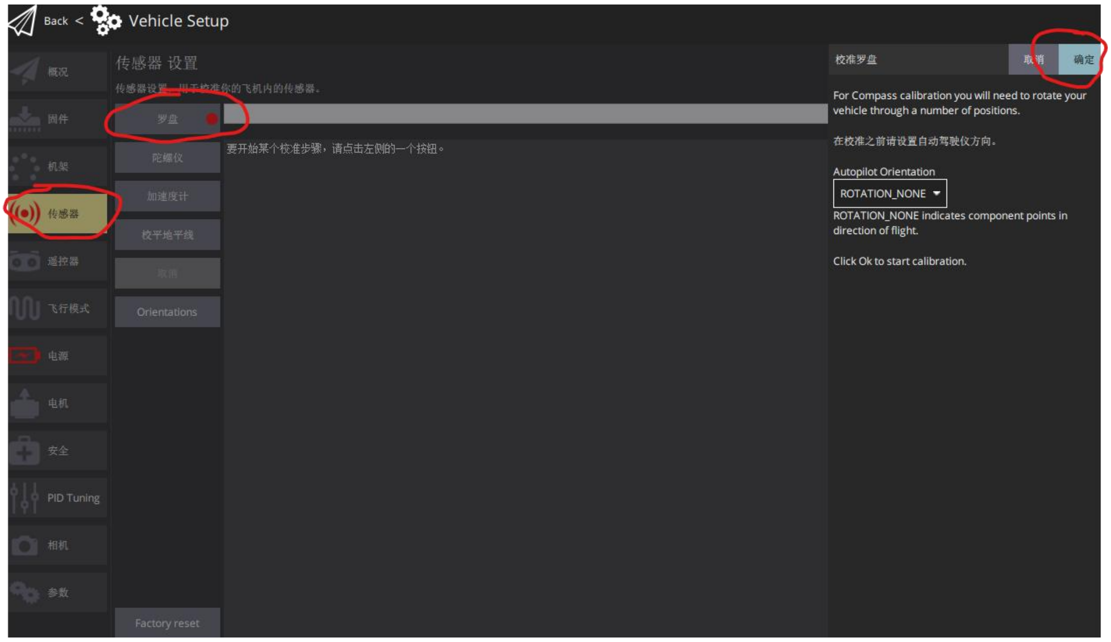
① Vehicle overview: displays the overall status of the currently connected vehicle, such as rack, sensor, remote control, flight mode, etc.

② Firmware: Do not connect the flight control first, click the following page, and then connect t he flight control to the computer with USB. Note that the flight control should not be powere d by batteries or other devices other than USB.

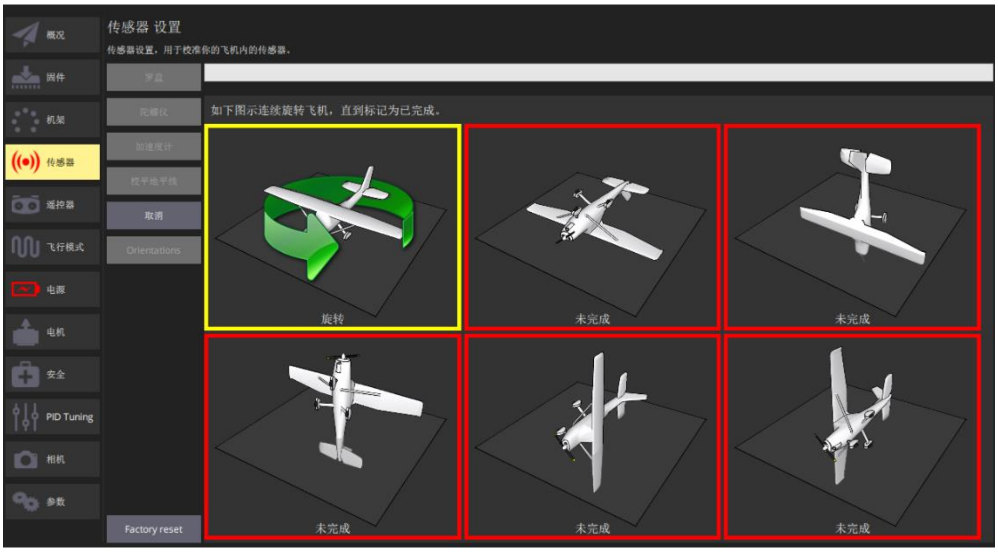
③ Rack: connect the flight control to the ground station, and set the rack to the model you want to set. After setting, please click "Apply and restart" on the upper right to take effect.



④ Sensor: The sensor mainly includes the sensors involved in the IMU. When calibrating, the co mpass is generally calibrated first, and the steps are as follows:

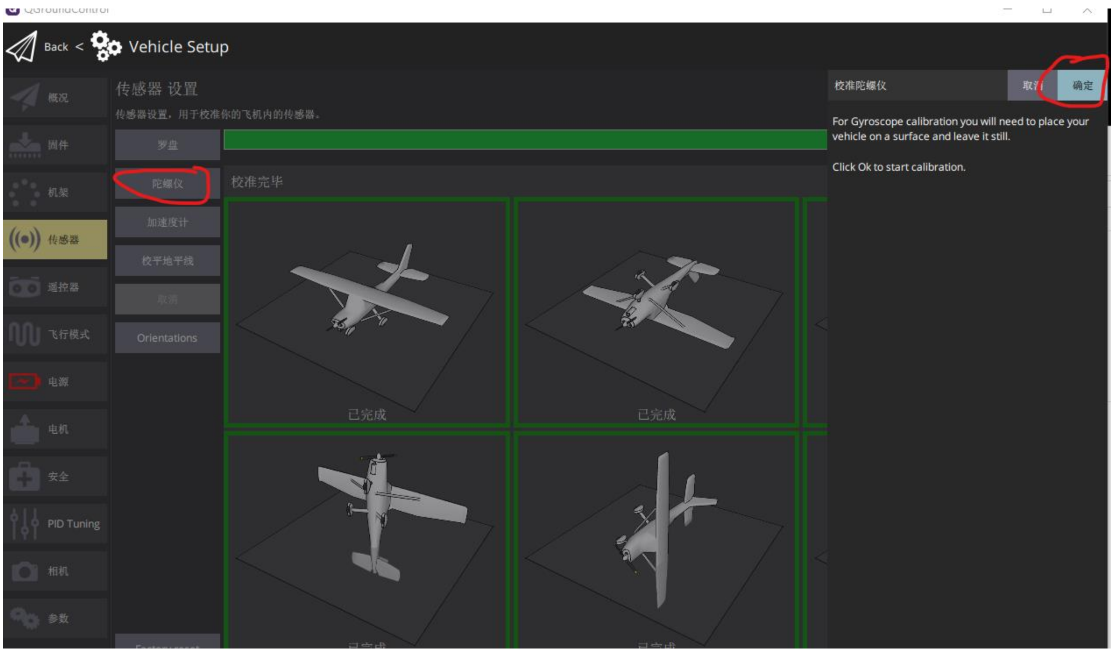


Position the drone in any direction shown in red and remain stationary. When prompted (direc tion image turns yellow), rotate the vehicle about the specified axis in any/both directions. W hen the current orientation calibration is complete, the associated image on the screen will tur n green.



Repeat the calibration process for all directions. After all orientations have been calibrated, Q GroundControl will display Calibration complete (all orientation images will be displayed in green and the progress bar will be completely filled). You can then move on to the next senso r.

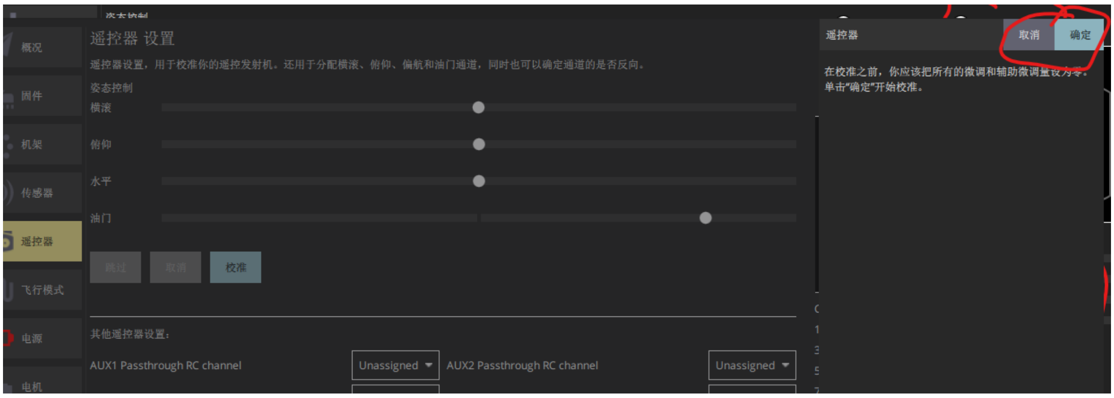
Calibrate the gyroscope: Click the gyroscope sensor button to place the drone horizonta lly on the ground and keep it stationary. Click OK to start the calibration. The bar grap h at the top is full for a successful calibration.



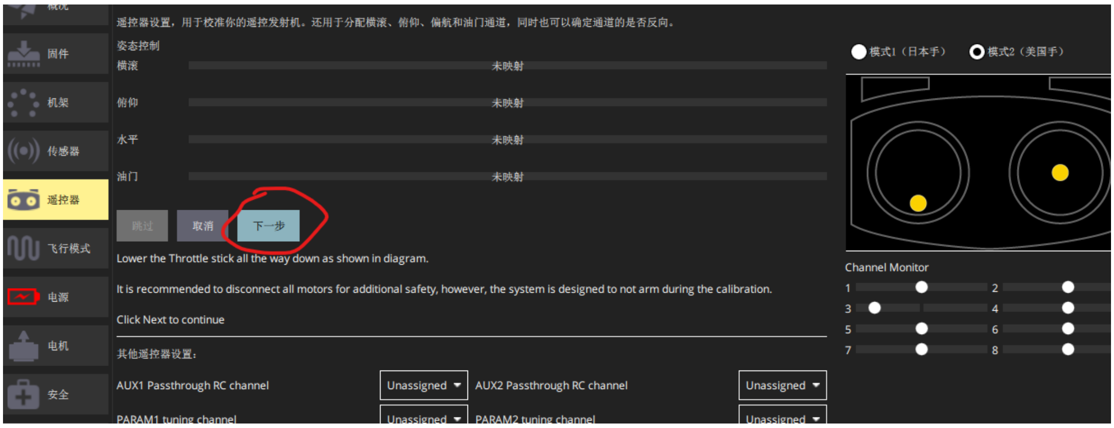
⑤ Remote control

Open the remote controller, switch to the remote controller page, check whether the channel c an be identified in the lower right corner, if so, you can calibrate, select the operation mode in the u pper right corner, and then click Calibrate

Then click OK

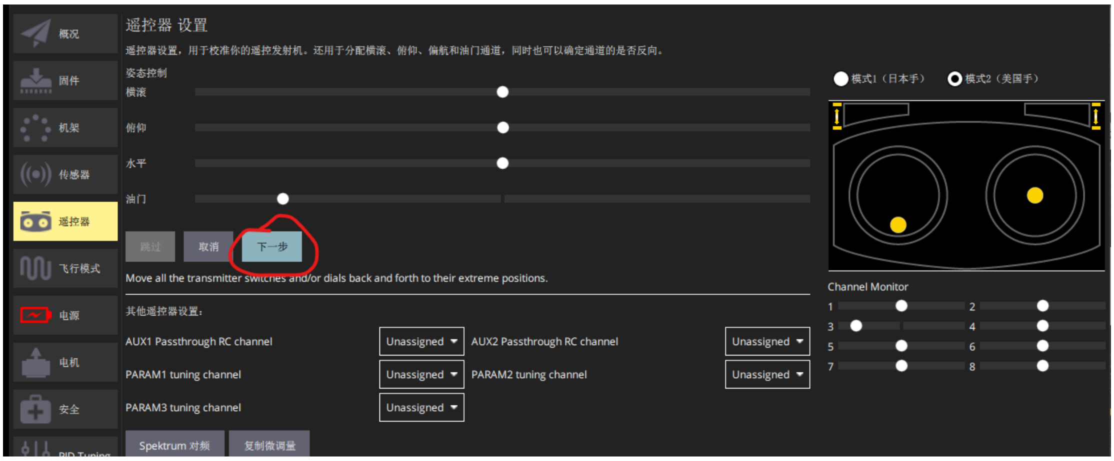


Then click "Next"



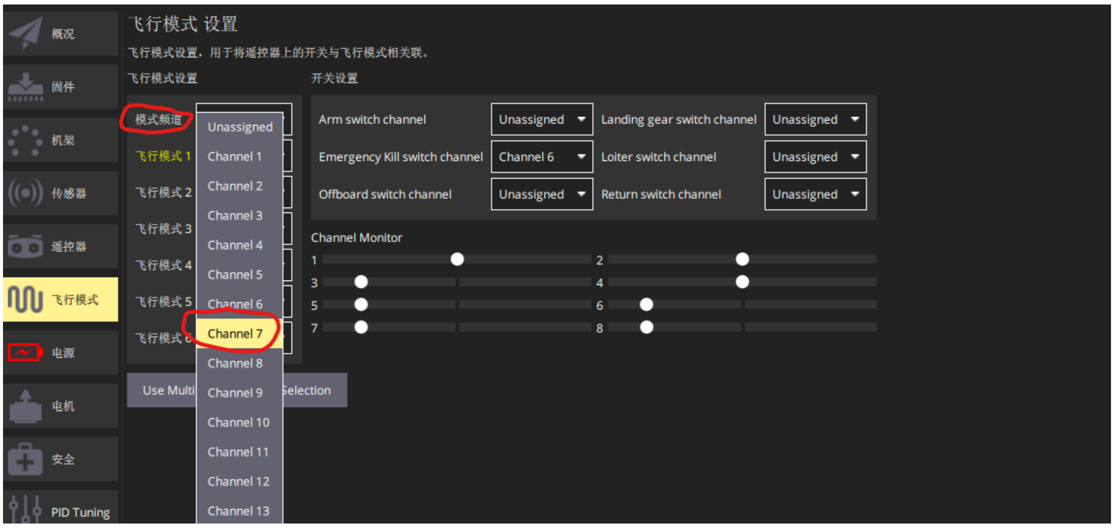
Move the remote control joystick to the position indicated in the following illustration.

When the pole is in place, the ground station will prompt for the next position to dial. After dialing all positions, press "Next" twice to save the settings.

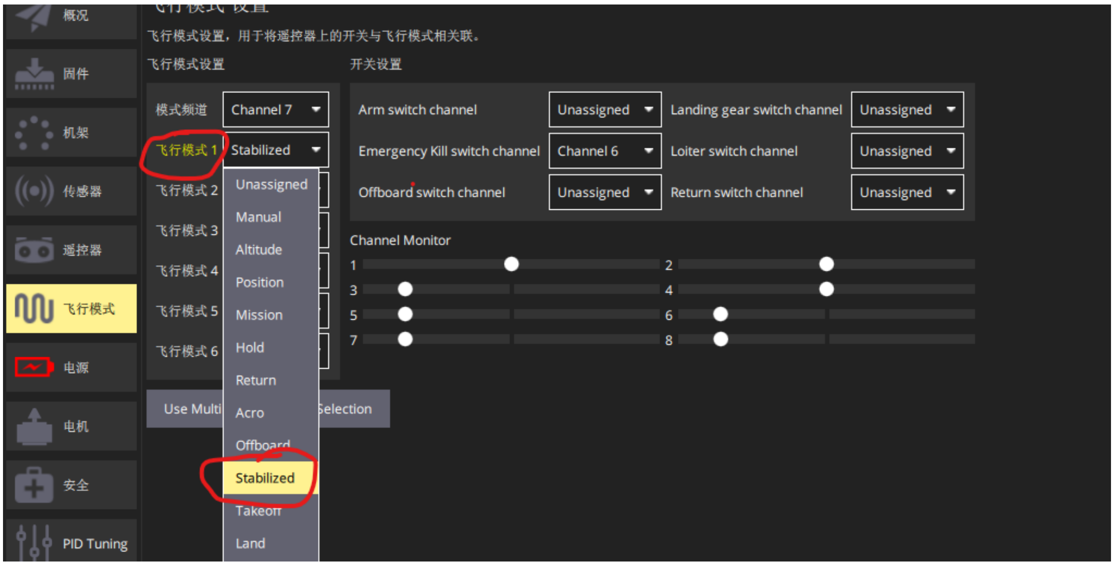


⑥　Flight mode switch

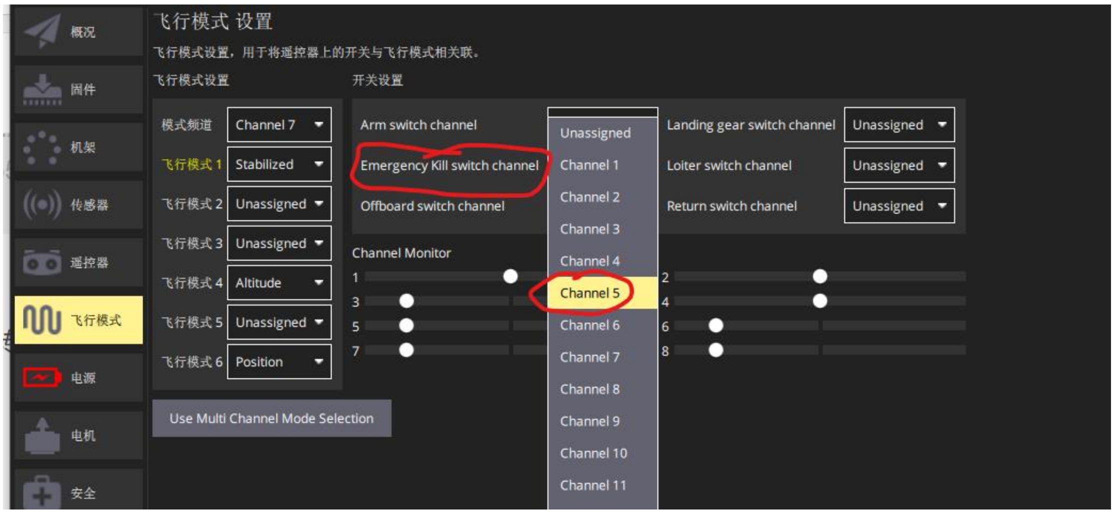
Click the check box on the right side of "Mode Channel" to set the corresponding remote control dial switch channel.



Then set the flight mode corresponding to the third gear respectively.



The other switch channels are on the right side of the flight mode, as shown below. Which one needs to be set, just set the remote control channel on the right side of this switch. I set a Kill switch here, and the channel is the fifth channel of the remote control. The function of the brake is to stop the motor directly, which can be set as required.



⑦ Power source

When calibrating the electric tuning, connect the flight control to the ground station with USB, without connecting the battery and installing the blade, and connect the signal line of the electric tuning to the flight control. Switch to the "Power" page, enter the number of battery cells and press Enter, click "Calibrate", and then plug in the battery to calibrate.

⑧ Motor: Displays the PWM of the motor

⑨ Safety: Under this menu, you can set the low battery fault protection trigger, object detection and remote control signal loss fault of the vehicle.

⑩ PID tuning: The PID control parameters of the vehicle can be tuned.

⑪ Flight Behavior。

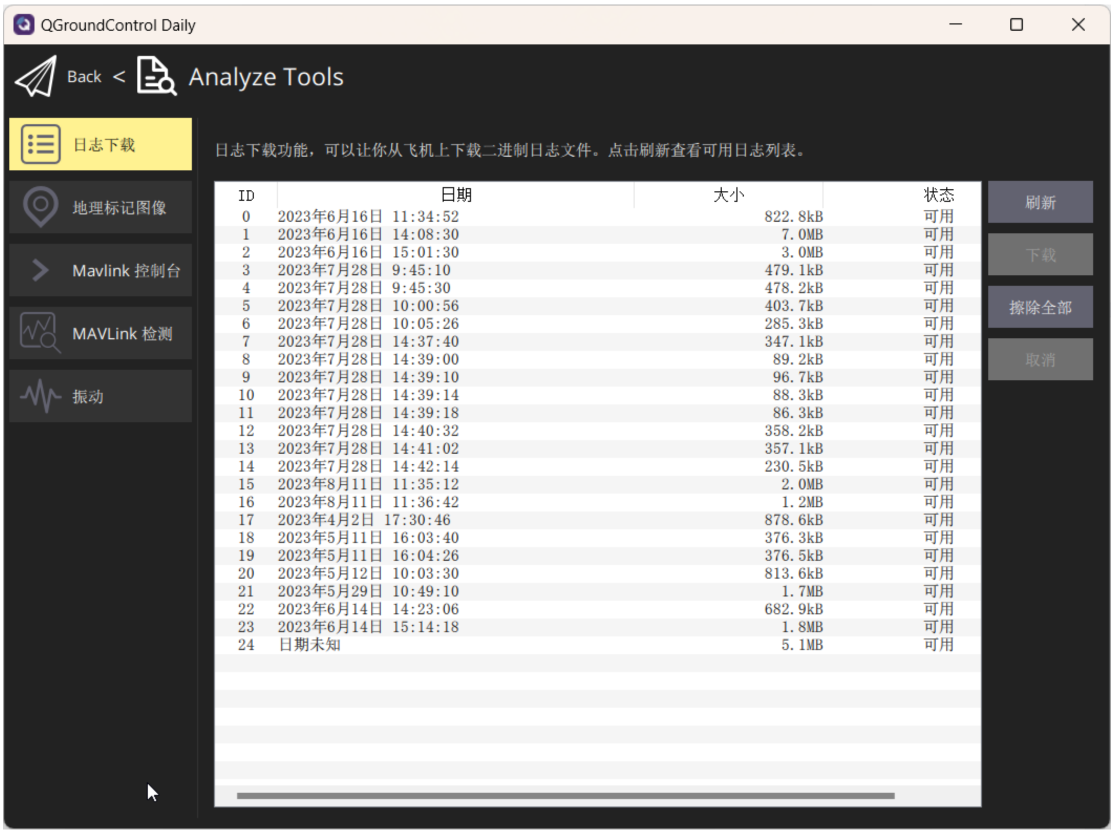
⑫ Camera settings

⑬ Parameter: Any parameter defined in the PX4 software can be modified here. After modificati on, it can take effect after restarting.

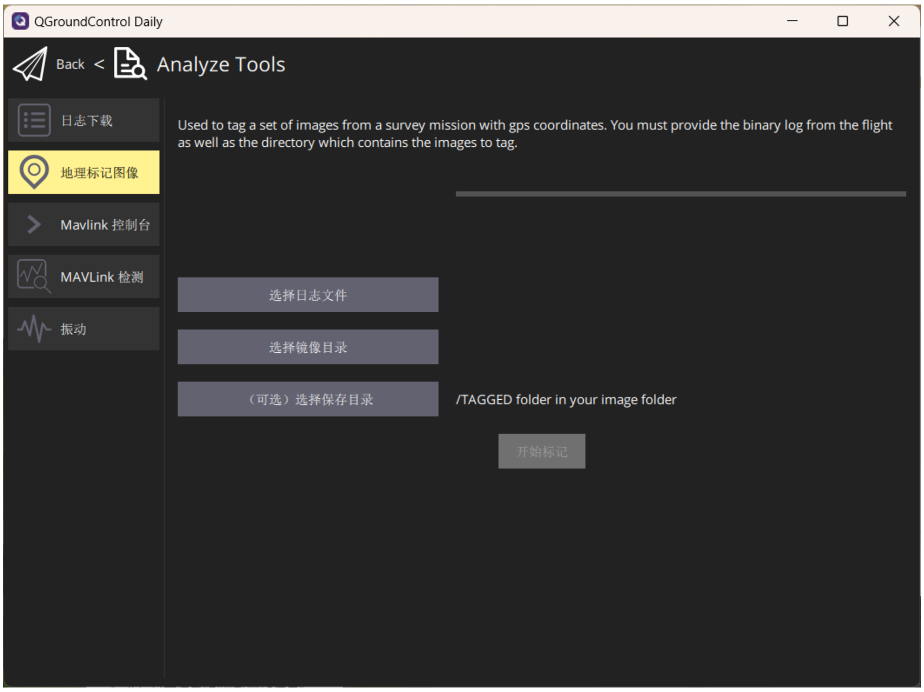
1.3.3. Data analysis（ Analyze Tools ）

QGC provides rich data analysis tools, mainly including log download, geotagged images, M AVlink console, MAVLink detection and vibration.

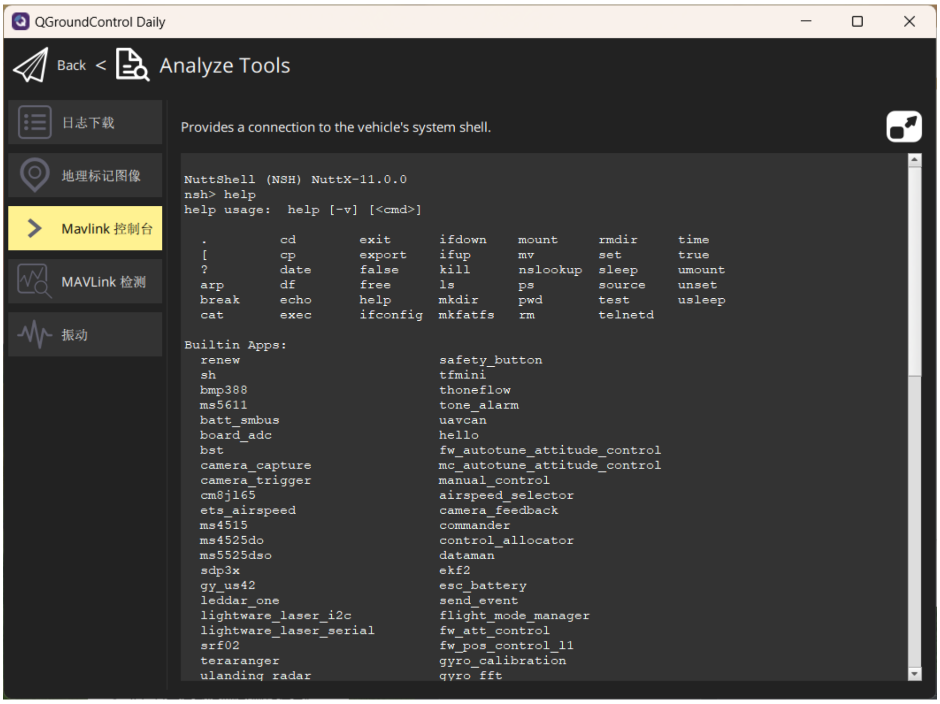
Log download: When the flight control is linked, select any one of the log information st ored in the current flight control memory card to download a file in.ulg format, which can be used to analyze the log through the website:https://docs.px4.io/main/zh/log/flight_log_analysis.html.



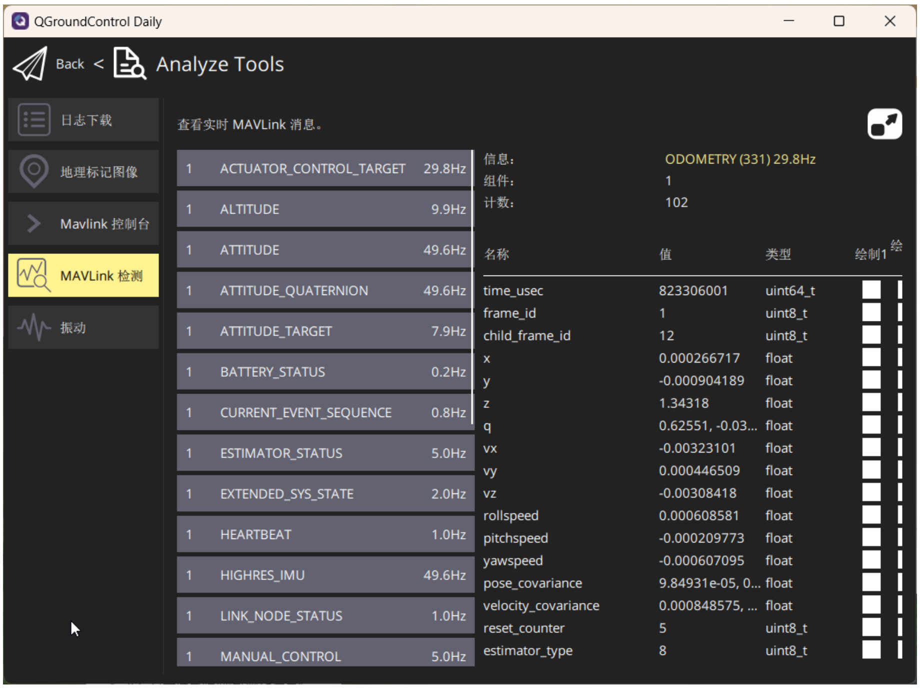
Geotagged images: images used to tag a group of survey missions with GPS, but must provide a binary log of waypoints and a directory containing the images to be tagged.



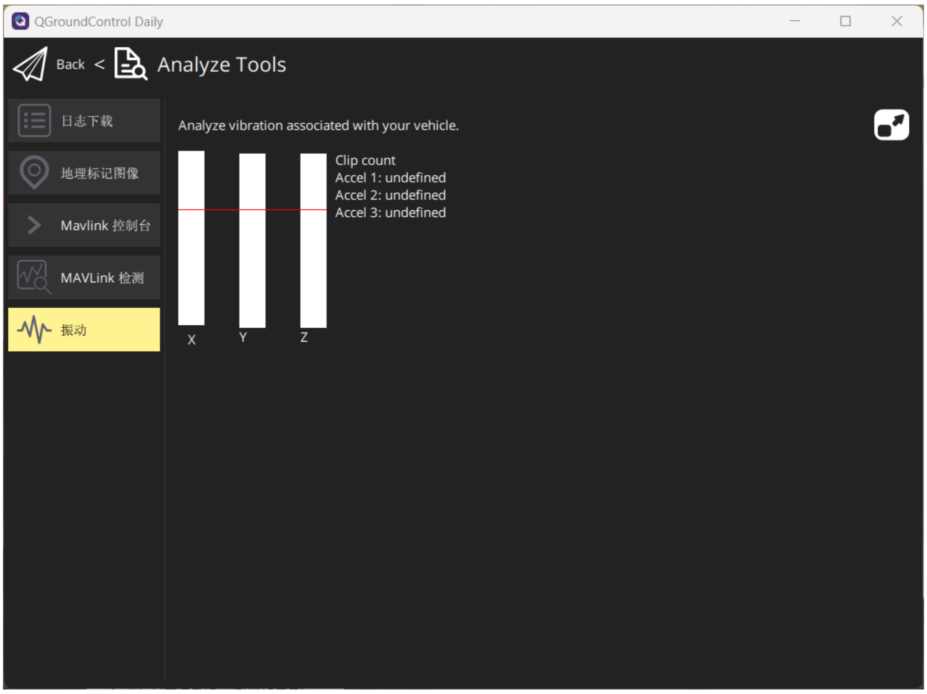
MAVLink Console: It provides a data communication link with the Shell of Nuttx, the flight control operation system on the vehicle.



MAVLink detection:



Vibration: Analyze the vibration associated with the vehicles



The relevant routines of the software are:

1. BasicExps\e10_Firmware-Upload\Readme.pdf

1. BasicExps\e16_Identify-Hardware-Command\Readme.pdf

1. BasicExps\e17_RoutePlanning\Readme.pdf

1.4. Python38Env

Python is a high-level, object-oriented, interpreted programming language. Originally created by Guido van Rossum in 1989, it has become a popular programming language for developing We b applications, data analysis, artificial intelligence, scientific computing, network programming, an d more. Python is a language that is easy to learn, easy to read, and easy to write, so it is also widel y used for teaching and entry-level programming.

Python38Env is a Python 3.8 programming language virtual environment, including numpy, p ymavlink, OpenCV, pyulog and other libraries, which can quickly develop algorithms related to un manned systems without requiring users to deploy python runtime environment and various functi on libraries.

The relevant routines of the software are:

1. BasicExps\e4_Log-Reads-Python38Env\Readme.pdf

1.5. MATLAB Automatic Code Generation Toolbox

MATLAB Automatic Code Generation Toolbox is an extended toolkit of MATLAB, which is used to generate C code, executable files, static libraries, dynamic libraries and other forms of exec utable files from Simulink models. These executables can be run directly on embedded platforms without the need for manual writing and debugging. Supports multiple embedded platforms includ ing ARM Cortex-M and A-series processors, NXP MPC55xx and MPC56xx series, Pixhawk serie s, and more.

The module library includes GPS data module, battery data module, uORB module and many other modules. Based on RflySim and the Pixhawk Support Package platform, users can design an d simulate control algorithms in Simulink, automatically generate C code and PX4 firmware from Simulink models, and burn them directly to the Pixhawk board, automatically generate C code and PX4 firmware from Simulink models, and directly burn them to the Pixhawk board; ③ Use MAT LAB scripts and functions to configure and calibrate the Pixhawk board and its peripheral devices; ④ Read and write data with the Pixhawk board in real time, etc.

The relevant routines of the software are:

1. BasicExps\e7_Code-Generation\Readme.pdf

1.6. SITL/HITL Batch processing Script

Batch processing technology means that the computer can process several tasks collected in g roups, and the whole process is completely automated without human intervention, which can also be called workload automation (WLA) and job scheduling. It has the advantages of speed, cost sa ving, accuracy and simple operation.

RflySim has developed a number of batch scripts based on batch processing technology, allow ing users to quickly deploy multiple, multiple, and multiple unmanned system combined simulatio ns with one click. And improve that development and simulation speed of the unmanned system. T he common batch processing scripts of the platform are as follows: (1) SITLRun. Bat: a batch proc essing file for starting multi-machine software-in-the-loop simulation, which essentially starts and configures part of the software and options of the RflySim platform in a script mode; (2) HITLRu n. Bat: a batch processing file for starting multi-machine hardware-in-the-loop simulation, Double-click the batch file and input the Pixhawk serial port number you want to participate in the simulati on according to the prompt to start the hardware-in-the-loop simulation of multiple machines (sort the aircraft IDs in the order of input serial ports). In addition, the RflySim platform provides a num ber of batch script files. Uch as SITLRunPos. Bat, SITLRunLowGPU. Bat, SITLRunMAVLink. B at, HITLRunPos. Bat, HITLPosSysID. Bat, HITLPosStr. Bat, etc., See * \ PX4PSP \ RflySimAPIs \ Bat Scripts for some script files, and see for C:\PX4PSP\RflySimAPIs\BatScripts\readme.txt instr uctions. Users can open these files through the editor, modify the parameters according to their per sonal needs, realize custom development, and quickly start simulation or algorithm verification.

1.6.1. HITLRun.bat

The conventional hardware-in-the-loop simulation script supports the input of serial port sequence (English comma "," separated) to start the multi-computer hardware-in-the-loop simulation.

Note: Lines beginning with REM are comment statements and will not be executed. Other bat script syntax rules can be searched and learned by themselves.

Note: The aircraft position of this script is automatically generated by the script according to the rectangular queue, and the control variables include:

SET/a START _ INDEX = 1 (the initial aircraft serial number, the CopterID of the aircraft generated by this script, with this START _ INDEX as the initial value, incremented by 1 in turn)

SET/a TOTOAL _ COPTER = 8 (total number of aircraft, amplitude is required only for multi-aircraft online simulation. Tell the actual total number of aircraft in this script to determine the side length of the rectangular queue)

SET UE4 _ MAP = Grasslands (Set Map Name)

SET/a ORIGIN _ POS _ X = 0 (origin X position of the rectangular formation in meters, only integer input is supported)

SET/a ORIGIN _ POS _ Y = 0 (Y position of the origin of the rectangular formation in meters, only integer input is supported)

SET/a ORIGIN _ YAW = 0 (the yaw angle at the origin of the rectangular formation, in degrees, only integer inputs are supported)

SET/a VEHICLE _ INTERVAL = 2 (aircraft separation for rectangular formation in meters, integer input only)

SET/a UDP _ START _ PORT = 20100 (the UDP communication interface for receiving external control data will automatically add 2 corresponding to CopterID, which usually does not need to be modified here, and can be modified only when the computer port is occupied)

SET/A DLLModelVal = DLLModel (Whether to use the DLL model, the name of the DLL model. It is supported here to generate the aircraft model of Simulink into DLL to import into the platform, and this mode supports fixed-wing, unmanned vehicle and other models.)

SimMode (here set to 0 or PX4 _ HITL for hardware-in-the-loop simulation)

SET IS _ BROADCAST = 0 (whether to simulate online or not, the target IP address sequence can be input here)

SET UDPSIMMOde = 0 (for the data protocol received by the UDP _ START _ PORT port, the UDP mode transmits the private structure of the platform and supports Simulink control; the MAVLink mode transmits the MAVLink protocol and supports Python and mavros control modes)

1.6.2. SITLRun.bat

Conventional software in-the-loop script supports inputting the number of aircraft and automatically starts multi-aircraft software in-the-loop simulation.

Compared with the HITLRun. Bat, the key codes are as follow

Set SimMode = 2 (software in-loop mode, corresponding to the value of CopterSimUI)

Set PX4SitlFrame = iris (here corresponds to the rack mode of PX4 flight control, here is a quadrotor)

1.6.3. HITLPos.bat

Enable hardware-in-the-loop simulation, support input of PosX, PosY and Yaw values to initialize the aircraft position yaw angle

Note: The input position strings are separated by commas ",".

The key codes are as follows:

SET/P PosXStr = Please enter the PosX (m) list: (input the initial position sequence of X in meter, which can be a floating point number)

SET/P PosYStr = Please enter the PosY (m) list: (input the initial position sequence of Y in meter, which can be a floating point number)

SET/P YawStr = Please enter the Yaw (degree) list: (input the initial value sequence of yaw, unit degree, can be floating point number)

1.6.4. SITLPos.bat

Enable the software in-loop simulation, and support the input of the values of PosX, PosY and Yaw to initialize the aircraft position yaw angle.

Note: Other precautions are the same as above.

1.6.5. HITLPosStr.bat

When HILS is enabled, the values of PosX, PosY, and Yaw are written in the form of strings in the PosXStr, PosYStr, and YawStr variables of the bat file without manual input

Note: When using, you need to manually modify the values of PosXStr, PosYStr and YawStr strings to set the position and yaw of the generated aircraft.

Note: The input position strings are separated by commas ",".

Note: The number of position sequences can be greater than the number of inserted flight controls, and the previous positions will be taken in turn for amplitude. For example, in this example, the position string contains 10 groups of aircraft initial postures, so it supports hardware-in-the-loop simulation of 1 to 10 aircraft.

The key codes are as follows:

SET PosXStr = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 (the initial position sequence of X in meters, which can be a floating point number)

SET PosYStr = 0,0,0,0,0,0,0,0,0,0,0 (the initial position sequence of Y, in meters, can be a floating point number)

SET YawStr = 0,0,0,0,0,0,0,0 (initial value sequence of yaw, unit degree, can be floating point number)

SITLPosStr.bat

Start the software in-loop simulation. The values of PosX, PosY and Yaw are written in the variables of PosXStr, PosYStr and YawStr in the bat file in the form of character strings, without manual input.

Note: Other precautions are the same as above.

1.6.6. HITLRunSysID.bat

Hardware-in-the-loop simulation is enabled, but CopterID is not sorted by automatically increasing by 1, but is automatically determined according to the value of SysID (configurable in QGroundControl)

Note: If the conventional mode is used, the CopterID is bound to the serial port sequence of the computer in sequence. The serial port number will change after each insertion and extraction of the flight control, so it is impossible to locate the determined flight control hardware through the CopterID.

Note: With this method, the CopterID can be directly bound to the flight control hardware, so that the flight control serial number can be quickly determined in case of failure.

Note: If you want to reuse the HITLRun. Bat script in regular mode, it is recommended that you change the SysID back to the default value of 1

The key codes are as follows:

SET/a Is SysID = 1 (turn on automatic calculation of CopterID from SysID)

1.6.7. HITLPosSysIDStr.bat

Enable hardware-in-the-loop simulation, automatically determine CopterID value through SysID, and support configuration of initial position sequence

Note: This script will index the PosXStr, PosYStr, and YawStr lists with the value of the SysID- START _ INDEX to determine the final location

Note: For example, if SysID is 15, START _ INDEX is 11, and PosXStr = 1,2,3,4,5,6,7,8,9,10, then the final X value of the aircraft should be the fourth digit of PosXStr (counted from 0), that is, PosX = 5

The key codes are as follows:

SET/a Is SysID = 1 (turn on automatic calculation of CopterID from SysID)

SET/a START _ INDEX = 1 (value of the initial aircraft of the position list)

SET PosXStr = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 (the initial position sequence of X in meters, which can be a floating point number)
SET PosYStr = 0,0,0,0,0,0,0,0,0,0,0 (the initial position sequence of Y, in meters, can be a floating point number)
SET YawStr = 0,0,0,0,0,0,0,0,0 (initial value sequence of yaw, unit degree, can be floating point number)

1.6.8. HITLPosSysID.bat

Turn on the hardware-in-the-loop simulation and automatically determine the CopterID value through SysID (same as the HITLPosSysIDStr. Bat), but the initial position sequence is manually input

1.6.9. HITLRunLowGPU.bat

Turn on conventional hardware-in-the-loop simulation and use Low GPU scenarios to ensure that low-profile computers can run on the platform
Note: The core modification is to replace GrassLands characters with LowGPU HITLRun. Bat.
The key codes are as follows:
SET UE4_MAP=LowGPU

1.6.10. SITLRunLowGPU.bat

Open the conventional software in-loop simulation, do not use the GrassLands scenario but use the LowGPU scenario to ensure that the low-profile computer can run the platform
Note: Other precautions are the same as above.

1.6.11. HITLRunMAVLink.bat

Open the conventional hardware-in-the-loop simulation, use the UDPSIMMODE value of the MAVLink _ Full, and support Python visual control and real flight control
Note: Only the value of UDPSIMMODE is modified here
Note: UDPSIMMODE can be a number or a character string, corresponding to the UI interface of CopterSim.
The key codes are as follows:
SET UDPSIMMOde = Mavlink _ Full (communication mode using Mavlink _ Full for 20100 series ports)

1.6.12. SITLRunMAVLink.bat

Open the conventional software in-loop simulation, use the UDPSIMMOde value of the MAVLink _ Full, and support Python visual control and real flight control
Note: Other precautions are the same as above.

1.6.13. HITLRunNoUI.bat

The hardware-in-the-loop simulation is started, and the UI-free version CopterSimNoUI. Exe is used, so that the calculation amount can be reduced, and the method is suitable for large-scale cluster use.
Note: The core modification is to replace the string of the CopterSim. Exe with the CopterSimNoUI. Exe.
Note: When the number of cluster routines of the platform exceeds 4, the NoUI version of CopterSim will be automatically used to improve flight stability.

1.6.14. SITLRunNoUI.bat

The software in-loop simulation is started, and the CopterSimNoUI. Exe without UI is used, so that the calculation amount can be reduced, and the method is suitable for large-scale cluster use.
Note: Other precautions are the same as above.

1.6.15. HITLPosAlt.bat

Hardware-in-the-loop simulation is enabled, and the Alt identifier is added to support the

configuration of the initial altitude, pitch angle, and roll angle of the aircraft.
Note: Other precautions are the same as above.

1.6.16. SITLPosAlt.bat

Open the software in-loop simulation, add the Alt identifier, and support the configuration of the aircraft's initial altitude, pitch angle and roll angle.
Note: Other precautions are the same as above.

1.6.17. HITLPosAltStr.bat

Hardware-in-the-loop simulation is enabled, the Alt identifier is added, the initial altitude, pitch angle and roll angle of the aircraft can be configured, and multiple aircraft can be started.
Note: Other precautions are the same as above.

1.6.18. SITLPosAltStr.bat

Open the software in-loop simulation, add the Alt identifier, support the configuration of the initial altitude, pitch angle and roll angle of the aircraft, and support the start of multiple aircraft.
Note: Other precautions are the same as above.

1.6.19. HITLRunChange3D.bat

Turn on the regular hardware in the ring simulation function, using the appearance of 100003, that is, quadcopter ClassID=3, + the first aircraft style, that is, Droneyee X330 style
The key code is as follows:
SET /a CLASS_3D_ID=100003

1.6.20. SITLRunChange3D.bat

Turn on the regular software in the ring simulation function, using the appearance of 100003, that is, quadcopter ClassID=3, + the first aircraft style, that is, Droneyee X330 style
The key code is as follows:
SET /a CLASS_3D_ID=100003

1.6.21. HITLPosStrGPS.bat

Turn on hardware-in-the-loop simulation to initialize the aircraft position using global GPS coordinates
The key code is as follows:
SET isPosGps=1 (If isPosGps is set to 1, PosXStr and PosYStr can be entered in latitude and precision format (in degrees); If there is no isPosGps or if it is set to 0, enter the initial position using the xy northeasterly (in meters) format.
SET isBatLLAOrin=1 (If isBatLLAOrin is set to 1 and LatLongAlt is set to the latitude and longitude, the latitude and longitude of LatLongAlt will be used as the origin of the GPS coordinates in the northeast Earth coordinate system. Instead, use the GPS Settings inside the model, or the values specified in the png+txt terrain file.)
Note: This feature only supports the full version and above, the free version cannot configure the GPS initial location.

1.6.22. SITLPosStrGPS.bat

Turn on the software in the loop simulation and initialize the aircraft position using global GPS coordinates
The key code is as follows:
SET isPosGps=1 (If isPosGps is set to 1, PosXStr and PosYStr can be entered in latitude and precision format (in degrees); If there is no isPosGps or if it is set to 0, enter the initial position using the xy northeasterly (in meters) format.
SET isBatLLAOrin=1 (If isBatLLAOrin is set to 1 and LatLongAlt is set to the latitude and longitude, the latitude and longitude of LatLongAlt will be used as the origin of the GPS coordinates in the northeast Earth coordinate system. Instead, use the GPS Settings inside the mod

```
el, or the values specified in the png+txt terrain file.)
     Note: This feature only supports the full version and above, the free version cannot confi
gure the GPS initial location.
```

The relevant routines of the software are:

1. BasicExps\e6_BAT-Startup\Readme.pdf

1.7. PX4 Firmware Source Code

PX4 evolved from PIXHAWK, a software and hardware project at the Computer Vision and Geometry Laboratory of the Federal Institute of Technology (ETH) in Zurich, Switzerland. The flight control system is completely open source, providing a low-cost and high-performance high-end autopilot for flight control enthusiasts and research teams around the world. After years of development and improvement by world-class developers from industry and academia, the PX4 flight control system has formed a perfect and reasonable software architecture, and with the Pixhawk series autopilot hardware platform, it constitutes the Pixhawk PX4 autopilot software and hardware platform, which can control multi-rotor, fixed-wing, airship and other vehicles. It is an open source UAV autopilot software and hardware platform widely used in the world.

The RflySim platform supports one-click deployment of the PX4 compilation environment. Different PX4 firmware compilation commands and firmware versions can be selected by customization. The platform will deploy the selected PX4 Firmware source code on the set installation path. If the firmware exists, the old firmware folder will be deleted and a new deployment will be made. Greatly improves the efficiency of PX4 environment deployment.

The relevant routines of the software are:

1. BasicExps\e12_PX4-App\Readme.pdf

1.8. WinWSL Subsystem

The Win10 WSL subsystem is a subsystem on the Windows operating system where users can run Linux applications, use the Linux command line interface (CLI), and install Linux distributions. The Linux system installed on the RflySim platform with one click is Ubuntu 18.04.5, which is mainly used for compiling the PX4 source code.

This platform also provides two other compiling environments to realize the simulation of Linux compiling environment under Windows platform, which are Msys2Toolchain compiling environment based on Msys2 and CygwinToolchain compiler based on Cygwin. Users can select different compilation environments according to their own PX4 version, and different selections can be made in the one-click deployment and installation interface to complete the switching of different compilation environments.

The relevant routines of the software are:

1. BasicExps\e9_Build-Firmware\Readme.pdf

1.9. Simulink Cluster Control Interface

The RflySim platform develops a cluster control interface based on Simulink S functions, as Figure 4shown in. The interface is implemented by C + + mixed programming through Simulink S functions, and cooperates with the advantages of Simulink's own UDP module. It has the advantages of high efficiency, small operation, low delay, more reliability and strong expansibility. Users can load the module into their own control system by copying and pasting to help users quickly realize the development of unmanned system cluster control.

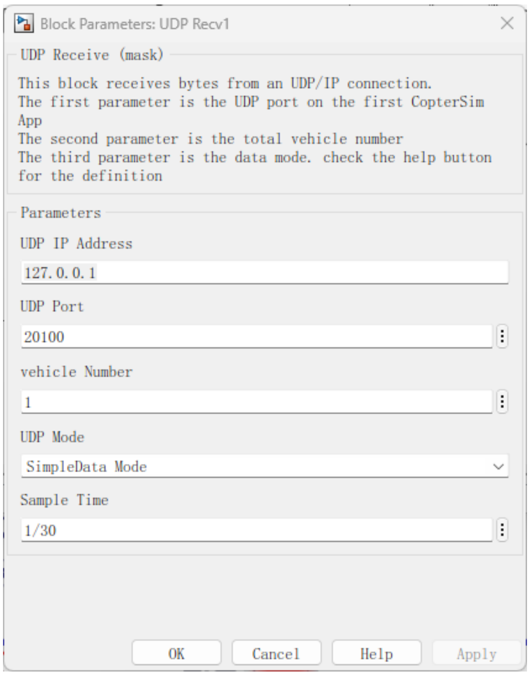


Figure 4Cluster control interface interface

- The first item "UDP IP Address" is the IP address of the target computer. If "127.0.0.1" is input, it can only accept the Pixhawk autopilot status forwarded by the local CopterSim and control it; "255.255.255.255" can receive and control the CopterSim program running in all computers in the LAN (the CopterSim of other computers needs to check the "Online" button); the designated IP such as "192.168.1.12" will only send control instructions to the host of the IP address. Generally speaking, the "255.255.255.255" broadcast can meet the demand in a small-scale cluster. When the number of aircraft continues to increase, the designated IP needs to be enabled to reduce the network load and improve the communication speed and reliability.
- The second "UDP Port" "is the initial port number of the first aircraft, and the default start po

rt is the 20100. Each CopterSim needs to occupy one port to send and receive messages. For example, this module needs to simulate aircraft with aircraft ID of 10 ~ 15, and this item need s to be filled with 20100 + 10 * 2 = 20120, and the latter item "Vehicle number" aircraft quan tity "needs to be filled with 5.

- The third item "Vehicle number" indicates the number of CopterSim to be connected and cont rols the number of input and output ports of the module. If 10 is input, the module will autom atically generate 10 pairs of input and output interfaces.
- The fourth "UDP mode" is the data mode protocol of the input and output interface, mainly in cluding the FullData complete mode (the most complete data, but the amount of transmitted d ata is large); SimpleData reduced data mode (more aircraft > 8, to avoid network congestion with too large data) and UltraSimple ultra-reduced mode (more than 20 aircraft per computer), with less latency.
- The fifth term is the Sample Time "sampling time, which should correspond to the Simulink s imulation time.

The relevant routines of the software are:

1. BasicExps\e8_SwarmAPI\Readme.pdf

1.10. RflySim Supporting information documents

RflySim platform provides very perfect learning materials and routine files, through PPT cour seware materials and RflySim APIs routine files, so that users can gradually and progressively lear n from the bottom control algorithm of unmanned system to the middle decision-making algorithm to the top learning algorithm. Build and develop the unmanned system you need in a one-stop wa y.

2. RflySim Platform Supporting hardware system

RflySim platform provides a complete set of supporting hardware system, including four-roto r UAV, flight control, remote control and other components. These components are perfectly comp atible with the platform, and can realize the software and hardware in the loop simulation experim ent in the RflySim platform, and realize the flight of UAV in the real environment based on the gen erated firmware.

2.1. Freescale series aircraft

At present, the supported aircraft include Feisi X150, Feisi X200, Feisi X450 and other four-r otor UAVs, among which Feisi X150 is a newly designed micro-four-rotor UAV for indoor cluster control research.

2.1.1. Freescale X 150 Four Rotary-wing UAV

The newly designed micro four-rotor UAV for indoor cluster control research has a symmetrical motor wheelbase of 140mm, an innovative full-protection structure design, abandons the previous complex wiring of carbon plates, prints the body with high strength and light weight in 3D, adopts laser fixed height and optical flow fixed point, and integrates the whole machine to comprehensively improve the efficiency of indoor cluster research.

Research direction: development of navigation and positioning of optical positioning system; development of centralized/distributed cluster formation algorithm; development of cooperative formation control of vehicle-machine combination; secondary development of ROS; secondary development of MATLAB;

Versions and performance

Product configuration	Standard Edition	The flagship edition
Basic configuration	Optical flow fixed point, laser fixed height, external magnetic compass	
Onboard card	ZYpi-3566	
Board performance	CPU: RK3566 Memory: 4GB, DDR4 Storage: 32GB WIFI: Integrated wifi 6	
Vision Sensor	None	Monocular sensor * 22MP
Positioning system	Indoor optical positioning system	Indoor optical positioning system /GPS
Means of communication	WIFI	
Basic software environment	Each sensor is driven	

Functional features	Focus on centralized and distributed cluster formation capabilities	On the basis of realizing centralized and distributed cluster formation functions, general visual function development and application can be carried out, and flight can be carried out based on GPS positioning.

Aircraft indicators

Freescale X150 intelligent UAV	
Size (including paddle)	200*200*85mm
Symmetrical motor wheel base	140mm
Aircraft weight	205g
Battery	3s，1300mAh 105g
Weight of complete machine (including battery)	310g
Maximum rising speed	2m/s
Maximum descent speed	2m/s
Maximum horizontal flight speed	5m/s
Maximum takeoff altitude	3500m
Endurance (no load)	8 minutes
Operating ambient temperature	-20 ° C to 50 ° C

Application scenario: Perfect indoor micro UAV cluster collaborative formation research solution, suitable for teaching and research in colleges and universities, as well as research in military units It is mainly used in the field of indoor UAV cluster control and distributed cluster algorithm verification.

2.1.2. Freescale X 200 Four Rotary-wing UAV

Indoor small intelligent UAV, symmetrical motor wheelbase 200mm, full carbon fiber protective body design, propeller sinking installation mode, internal laser fixed height and optical flow fixed point, suitable for indoor UAV cluster cooperative formation application, with distributed cluster UAV cooperative control capability. It can be equipped with visible light camera and airborne visual processing board, and has the ability of visual navigation, target recognition and target following.

Scientific research direction

Model-based design and development;

ROS control development;

Development of MATLAB control;

Centralized/distributed cluster control algorithm development;

Carry out visual navigation, target recognition and target following algorithm verification;

Product version

Product configuration	Standard Edition	Monocular version	Model design version
Basic configuration	Optical flow fixed point, laser fixed height, external magnetic compass		
Flight control	Racer flight control		
Onboard card	NX Xavier		NX Xavier/ZYpi3566
Vision Sensor	T265	Monocular camera	None
Means of communication	WIFI		
Basic software environment	Each sensor is driven		
Functional features	T265 is used for positioning, and high-precision indoor centralized/distributed cluster control algorithm is developed.	Centralized/distributed cluster control algorithm development; target recognition, target following algorithm verification	Model-based design and development; ROS control development;

Aircraft indicators

Freescale X200 intelligent UAV	
Size (including paddle)	300*300*160mm
Symmetrical motor wheel base	200mm
Aircraft weight	580g
Battery	4s，5300mAh，469g
Weight of complete machine (including battery)	1049g

Additional maximum load	200g
Maximum rising speed	2m/s
Maximum descent speed	2m/s
Maximum horizontal fligh t speed	10m/s
Maximum takeoff altitude	4000m
Endurance (no load)	20 minutes
Operating ambient temper ature	-20 ° C to 50 ° C

Application scenario

As a professional intelligent aircraft product for universities and research institutes, the indoo r small UAV cluster cooperative formation research solution is mainly applied in the following res earch fields: model-based design and development; indoor centralized/distributed cluster algorithm development; visual navigation; target following; and target recognition.

2.1.3. Freescale X 450 Four Rotary-wing UAV

Professional outdoor small intelligent four-rotor UAV, symmetrical motor wheelbase 450 mm, modular design of the whole machine, while carrying the airborne computer, equipped with functi onal modules such as depth camera and laser radar, forming a perfect outdoor intelligent aircraft, e xcellent product performance can cope with complex outdoor flight environment. It is an intelligen t aircraft research platform for outdoor cluster formation algorithm development, slam navigation a nd other research fields.


Scientific research direction

Model-based design and development;

Secondary development of ROS;

Secondary development of MATLAB;

Centralized/distributed cluster control of UAV;

Development of visual slam navigation and laser slam navigation;

Product version

Product configuration	Pilot version	The flagship edition	Advanced version

Basic configuration	Optical flow fixed point, laser fixed height, external magnetic compass		
Flight control	Racer flight control		
Onboard card	NX Xavier		
Visual odometer	T265 camera		
Space exploration	D435i depth camera	Silan S1 lidar	D435i depth camera Silan S1 lidar
Positioning system	GPS/RTK		
Communication link	Within 200 m-Onboard WiFi; 3km-ZY-H3; 10km-ZY-H12		
Basic software environment	Each sensor is driven UAV offboard control sample program		
Functional features	Outdoor group formation flight control lasting for more than 20 minutes; verification and development of visual slam navigation algorithm realized by single machine	Outdoor swarming formation flight control; Single machine implementation; Laser slam navigation algorithm validation and development	On the basis of the cluster function, the functions of visual slam navigation and laser slam navigation algorithm verification and development are realized simultaneously by a single machine.

Aircraft indicators

Freescale X450 intelligent UAV	
Size (without paddle)	420*420*240mm
Symmetrical motor wheel base	450mm
Aircraft weight	1200g
Battery	6s，6000mAh，862g
Weight of complete machine (including battery)	2062g
Additional maximum load	1000g
Positioning accuracy	GPS: vertical: ± 0.5m; horizontal: ± 2m
	RTK: vertical: ± 3cm; horizontal: ± 5cm
Maximum rising speed	2m/s
Maximum descent speed	2m/s
Maximum horizontal flight speed	8m/s
Maximum takeoff altitude	4000m
Endurance (no load)	30min
Operating ambient temperature	-20 ° C to 50 ° C

Application scenario

The perfect outdoor small UAV cluster cooperative formation research solution is suitable for teaching and scientific research in colleges and universities, as well as scientific research in military units, and is applied to Slam algorithm development/verification, path planning/obstacle avoida nce algorithm development, AI algorithm development/verification and other fields.

2.1.4. Freescale X680 Four Rotary-wing UAV

Medium-sized intelligent four-rotor UAV symmetrical motor wheelbase 680 mm, the whole machine adopts industrial design, high-strength fuselage can be used as a multi-mission load flight platform, using laser fixed-height optical flow fixed-point, and equipped with depth camera and laser radar and other functional modules, with visual navigation development and target following development conditions at the same time, it can carry out a larger load mission flight. It is a multi-functional intelligent UAV with load, long-term endurance and scientific research and development.

Scientific research direction:

Model-based design and development;

Centralized/distributed cluster control of UAV;

Outdoor airborne cluster control algorithm development;

ROS control development, support MATLAB control development;

Combine with that unmanned vehicle to carry out the coordinate formation control of the integration of the sky and the earth;

Carry out visual navigation, target recognition and target following algorithm verification;

Product Version:

Product configuration	Standard Edition	Customized version
Basic configuration	Optical flow fixed point, laser fixed height, external magnetic compass	
Flight control	H7 flight control	
Onboard card	NX Xavier	
Space exploration	D435i	Laser radar
Pod	None	G1 PTZ pod
Other functional modules	None	Custom carry
Positioning system	GPS/RTK	
Communication link	3km—ZY-H3； 10km—ZY-H12	

Basic software environment	Each sensor is driven	
Functional features	Large load, long endurance; outdoor multi-mission load group formation flight; carrying NX board, while carrying out a variety of complex algorithm verification, single machine to achieve target recognition, visual navigation and other artificial intelligence application development.	Sensor or functional module equipment can be customized according to specific application requirements to meet various functional requirements such as image recognition and target following. It is recommended to carry G1 PTZ pod, laser radar, RTK high-precision positioning module, customized PTZ pod, etc.

Aircraft indicators

Freescale X680 intelligent UAV	
Size (without paddle)	567*567*400mm
Symmetrical motor wheel base	680mm
Aircraft weight	2550g
Battery	6s， 16000mAh， 1475g
Weight of complete machine (including battery)	4025g
Additional maximum load	2000g
Positioning accuracy	GPS: vertical: ± 0.5m; horizontal: ± 2m
	RTK: vertical: ± 3cm; horizontal: ± 5cm
Maximum rising speed	2m/s
Maximum descent speed	2m/s
Maximum horizontal flight speed	12m/s
Maximum takeoff altitude	5000m
Endurance (no load)	40 minutes
Operating ambient temperature	-20 ° C to 50 ° C

Application scenario

The perfect outdoor medium-sized intelligent UAV solution is suitable for teaching and scientific research in colleges and universities, as well as scientific research in military units. It is mainly used in the field of indoor UAV cluster control and distributed cluster algorithm verification.

2.2. PX 4 Series flight control

Since the RflySim platform is developed based on the PX4 software system, under normal circumstances, any flight control that supports the PX4 software system can be used on the RflySim platform. Pixhawk 2.4.8 (also known as Pixhawk 1), Pixhawk 6C and Pixhawk 6X are currently supported for a long time.

2.2.1. Pixhawk 2.4.8(Also known as Pixhawk 1)

Pixhaw 2.4.8 is an advanced autopilot designed by the PX4 Open Hardware Project and built by 3D Robotics. It features advanced processor and sensor technology from ST, as well as the Nutt X real-time operating system, enabling amazing performance, flexibility and reliability to control a ny autonomous vehicle. Its characteristics are as follows:

1. Advanced 32-bit ARM CortexM4 high-performance processor running the NuttX RTOS re al-time operating system.

2.14 PWM/steering gear outputs (8 of them have safety and manual control functions, and the other 6 are auxiliary and compatible with high power), with rich peripherals (UART, I2C, SPI, CA N).

3. Redundancy design, integrated backup power supply and basic safety flight controller, can safely switch to backup control when the main controller fails.

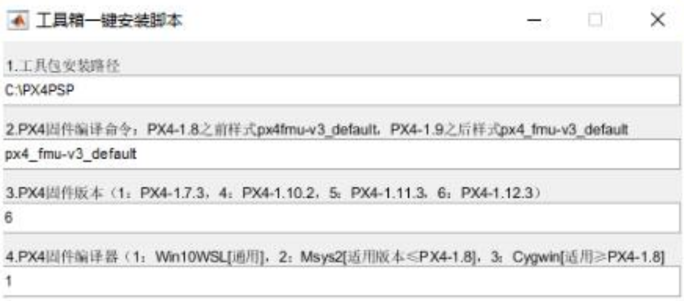
4. The backup system integrates the mixed control function and provides automatic and manu al mixed control modes.

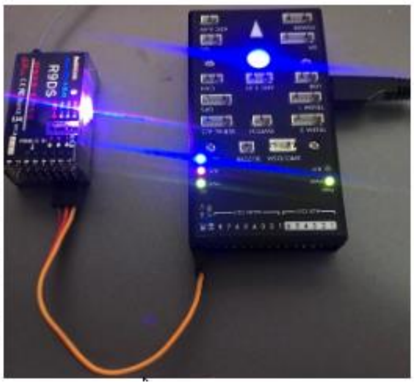
5. Redundant power input and automatic fail-over, external safety button for easy motor start.

6. Multi-color led lights, high power, multi-tone buzzer.

7. Micro SD records flight data at a high rate for a long time.

If Pixhawk 2.4.8 (2m flash) flight control hardware (corresponding firmware is px4 _ fmu-v3) is used, it is recommended to use the software installation configuration shown in the following fi gure and the hardware connection configuration shown in the lower right figure.





- Compile the command with px4 _ fmu -v3 _ default.
- Use "6": PX4 version 1.12.3 firmware.
- Use "1": Win10 WSL compiler.
- Pixhawk 1 comes with its own LED light, which does not require an external module. Just connect the remote control receiver as shown in the figure on the right.

Note: Pixhawk 2/3/4 do not come with LED modules at first, so you need to buy external LED modules.

2.2.2. Pixhawk 6C

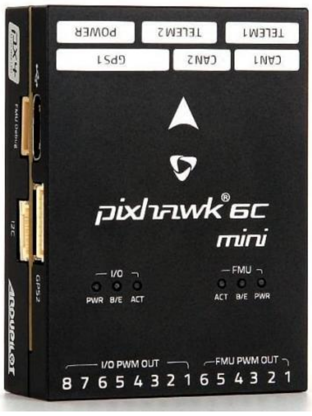
The Pixhawk 6C is the latest update to the successful family of drone controllers based on the Pixhawk FMUv6C open and connectivity standard. It is equipped with PX4 autopilot. Inside the Pixhawk 6C, an STM32H743-based chip manufactured by STM can be found, paired with sensor technology from Bosch and InvenSense to provide flexibility and reliability for the control of any autonomous vehicle, suitable for both academic and commercial applications. Its characteristics are as follows:

1. High-performance STM32H743 processor with more computing power and memory capacity;

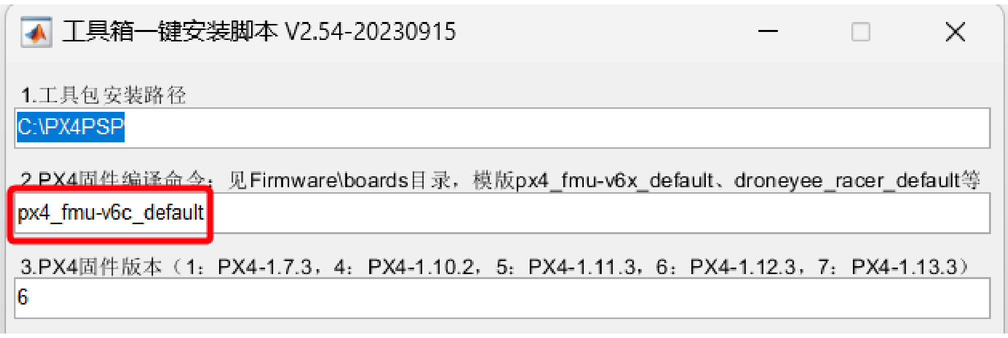
2. New cost-effective design with low chassis dimensions;

3. Newly designed integrated vibration isolation system filters high frequency vibration and reduces noise to ensure accurate readings;

4. Inertial Measurement Units (IMUs) are temperature controlled by a built-in heating resistor, ensuring optimum operating temperature for the IMUs.



If the flight control hardware of Pixhawk 6C is used, it is recommended to use the software installation configuration shown in the figure below. The hardware connection configuration is the same as Pixhawk 2.4.8.



- Compile the command using px4 _ fmu -v6c _ default.
- Use "7": Firmware for PX4 version 1.13.3.
- Use "1": Win10 WSL compiler.

2.2.3. Pixhawk 6X

Inside the Pixhawk 6X, you'll find an STM32H753 based chip manufactured by STM, paired with sensor technology provided by Bosch, InvenSense, to provide flexibility and reliability for the control of any autonomous vehicle, suitable for both academic and commercial applications. The Pixhawk 6X's H7 microcontroller contains an Arm ® Cortex ® -M7 core running at up to 480MHz, with 2MB of flash storage and 1MB of RAM. The PX4 Autopilot takes advantage of enhanced processing power and RAM. Because of the newer processing power, developers can be more efficient and productive, making their development work more complex and model. The FMUv6X open standard includes a built-in high-performance, low-noise inertial measurement unit (IMU) designed to improve stability. Separate LDOs power each sensor group with independent power supply control. A vibration isolation system that filters high frequency vibrations and reduces noise to ensure accurate readings, enabling the vehicle to achieve better overall flight performance. The external sensor bus (SPI5) has two chip-select lines and data-ready signals for additional sensors and loads connected to the SPI interface, and is equipped with a built-in microchip Ethernet PHY that enables high-speed communication via Ethernet. Pixhawk 6X is perfect for corporate research labs, startups, academic research (including professors, graduate students, and students), and business applications. Its characteristics are as follows:

1. High performance STM32H753 processor;

2. Removable flight control board: The independent IMU, FMU, and base systems are connected via the 100 Pin and 50 Pin Pixhawk autopilot bus connectors.

3. Redundancy: Triple IMU sensors and double barometric pressure sensors on their respective buses.

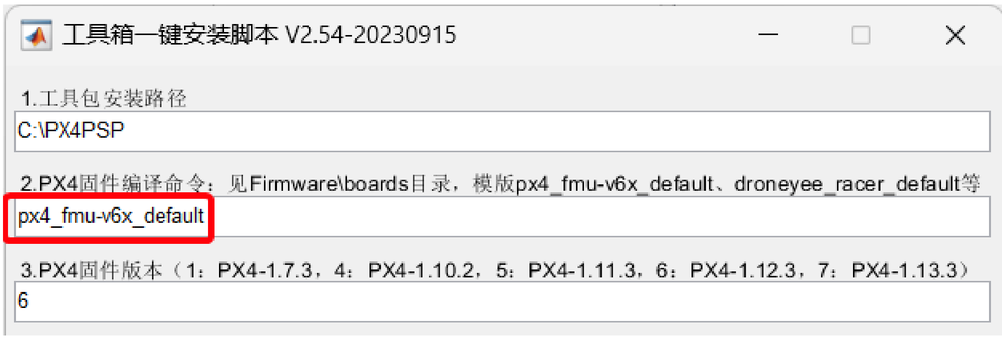
4. Triple Redundant Area: Fully isolated sensor area with respective busses and respective power controls.

5. Newly designed vibration isolation system filters high frequency vibrations and reduces noise to ensure accurate readings.

6. The Ethernet interface is used for high-speed mission computer integration.

7. The IMU is temperature controlled by a built-in heating resistor to ensure the optimum operating temperature of the IMU.

If the flight control hardware of Pixhawk 6X is used, it is recommended to use the software installation configuration shown in the figure below. The hardware connection configuration is the same as Pixhawk 2.4.8.



- Compile the command using px4 _ fmu -v6c _ default.
- Use "7": Firmware for PX4 version 1.13.3.
- Use "1": Win10 WSL compiler.

2.3. Common remote control Configuration

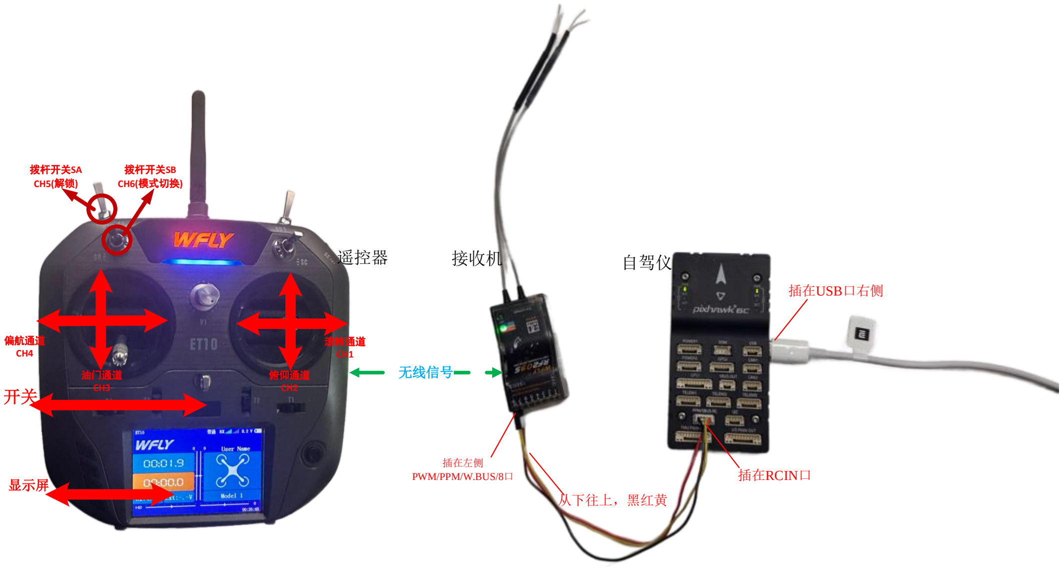
The remote controller used on this platform is recommended to use the "American hand" control mode, that is, the left joystick corresponds to the throttle and yaw control, while the right joystick corresponds to the roll and pitch. In the remote controller, roll, pitch, throttle and yaw respectively correspond to CH1 ~ CH4 channels of the receiver, and the left and right upper levers correspond to CH5/CH6 channels, which are used to trigger flight mode switching.

The throttle lever (CH3 channel) corresponds to the fluctuation of the PWM signal from 1100 to 1900 from the lowest end and the highest end respectively (different channels or different remote controllers will have differences, so calibration is required); the roll (CH1 channel) and yaw (CH4 channel) rockers correspond to the PWM signal from 1100 to 1900 from the leftmost end to the rightmost end; The pitch (CH2 channel) rocker corresponds to the PWM signal from 1900 to 1100 from the lowest end to the highest end; CH5/6 is a three-stage switch, and the PWM signals from the top (the position farthest from the user) to the bottom (the position closest to the user) are 1100, 1500 and 1900.

油门：控制上下运动，对应固定翼油门杆
偏航：控制机头转向，对应固定翼方向舵
俯仰：控制前后运动，对应固定翼升降舵
滚转：控制左右运动，对应固定翼副翼

The configuration and calibration methods are as follows:

1. Connect the Pixhawk and the receiver correctly, connect the Pixhawk and the computer with t he USB cable, open the remote control, open the QGroundControl ground station software, a nd click on the "Radio" tab as shown in the lower right figure.

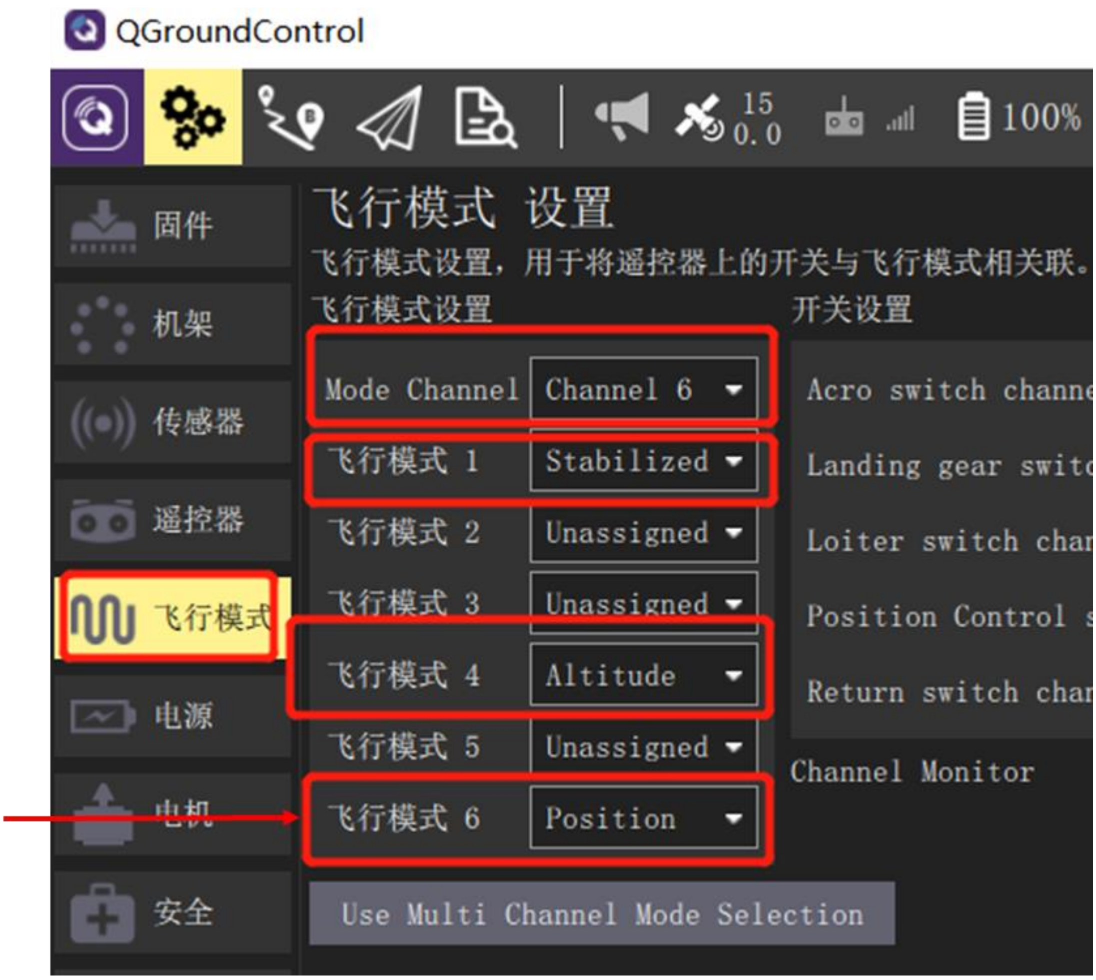


2. Dial the channels from CH1 to CH5 of the remote controller from left to right (or from top to bottom) in turn (see the upper right figure), and observe the white dots of each channel in the red box area on the right side of the ground station in the lower right figure. If small dots 1, 2, 4, 5 and 6 move from left to right (PWM from 1100 to 1900) and dot 3 moves from right to left, the remote control is set correctly. Otherwise, the remote control needs to be reconfigured.
3. Click the "Calibrate" button in the lower right figure and follow the prompts to calibrate the remote control.

4. Click the "Calibrate" – "Next" button on the QGC ground station, and then place the joystick in the position shown in the right figure (according to the real-time prompt on the QGC page) to complete the remote control calibration.

Airplane mode settings

1. After the above remote control calibration steps, click the ground station to enter the "Flight Modes" (flight mode) setting page, and select "Mode Channel" (mode channel) as Channel 6 tested previously. Since the CH6 channel is a three-stage switch, the top, middle and lower po sitions of the switch correspond to three labels of "Flight Mode 1, 4 and 6" respectively.
2. Set the three labels to Stabilized (only attitude control), Altitude (attitude and height control), and Position (attitude, altitude, and horizontal position control). In the subsequent hardware-i n-the-loop simulation, different control effects can be experienced by switching different mod es.



In addition, the platform also supports remote controls such as Ledi AT9S Pro, Tiandifei ET07, Fox i6s, Futuba T14SG, etc. For more detailed configuration of the remote control, please see: 1.BasicExps\e11_RC-Config\Readme.pdf .

3. RflySim Introduction to Platform Experiment Process

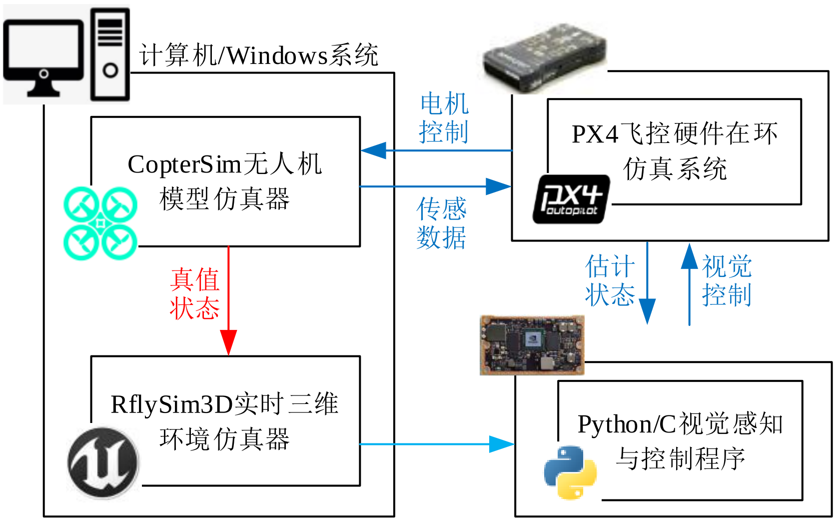
3.1. Bottom control system development Experimental process

According to the order from easy to difficult, the development of the bottom control system is divided into: algorithm development and digital simulation verification stage, simplified model joint debugging stage, software in the loop simulation stage, hardware in the loop simulation stage, and indoor and outdoor flight test; The main task of the algorithm development and digital simulation verification stage is to develop and verify the underlying control algorithm based on the simplified unmanned system model, so that the developed algorithm meets the initial requirements. Based on the previous stage of development, the main task of the simplified model joint debugging stage is to realize the top-level control of the unmanned system through external control, so that the simulation of the unmanned system can form a closed loop to achieve the purpose of unmanned system joint debugging. Software-in-the-loop Simulation (SIL) refers to compiling the generated source code on the host and executing it as a separate process. The numerical equivalence between the model and the generated code is tested by comparing the normal mode simulation results with the SIL simulation results. The SIL simulation of RflySim platform can be carried out in MATLAB/Python environment. The control algorithm is designed in Simulink/Python using the given unmanned vehicle simulation model and routines, and the model and controller are correctly connected to ensure that the input and output signals are consistent with the actual unmanned system. For example, multi-rotor UAV system: the multi-rotor model sends sensor data or state estimation information (such as attitude angle, angular rate, position and velocity) to the controller, and the controller sends each motor PWM control command back to the model, thus forming a SIL simulation closed-loop system. Users can observe the control performance and modify or design the controller to meet the desired performance requirements. Hardware-in-the-loop Simulation (HIL) phase is a technology used for the development and testing of real-time embedded systems. HIL simulation provides a dynamic system model, which can simulate the real system environment, add the mathematical representation of the relevant dynamic system, and connect it to the simulation system platform through the input and output of the embedded system [3]. The RflySim platform can import the Simulink unmanned vehicle model parameters into CopterSim, download the code generated by the Simulink controller algorithm to the Pixhawk autopilot, and then replace the virtual signal line in Simulink with the USB physical signal line. CopterSim sends the sensor data (such as accelerometer, barometer, magnetometer, etc.) to the Pixhawk system through the USB data line; the PX4 autopilot software in the Pixhawk system filters and estimates the state of the received sensor data, and sends the estimated state information to the controller through the internal uORB message bus; The controller sends the PWM control command of each motor back to CopterSim through the USB data line, thus forming a hardware-in-the-loop simulation closed loop.

3.2. Top-level control system development Experimental process

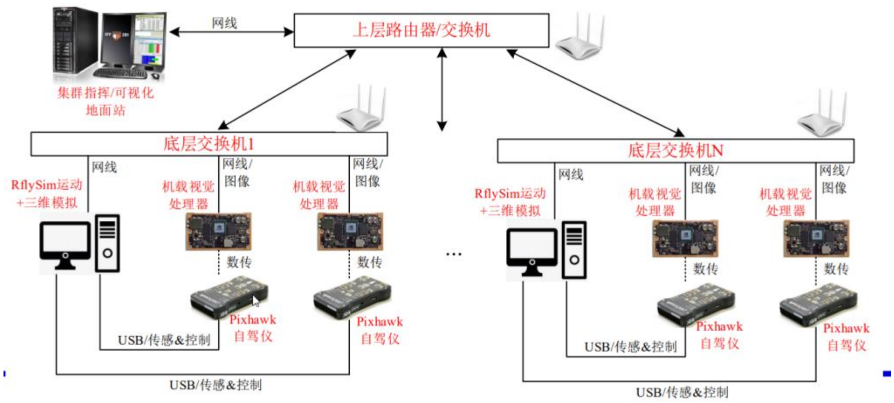
3.2.1. Airborne board hardware-in-the-loop phase

Based on HIL simulation stage, more hardware is added in this stage, such as networking communication module, vision processing module, data acquisition module and so on. At this stage, we need to deploy the system to the actual hardware devices, integrate and debug different hardware to ensure that they can cooperate with each other to achieve the efficient operation of the whole system. This stage is an important part of the whole development process, and it is also the key stage to ensure that the system can finally run normally in the actual scenario.



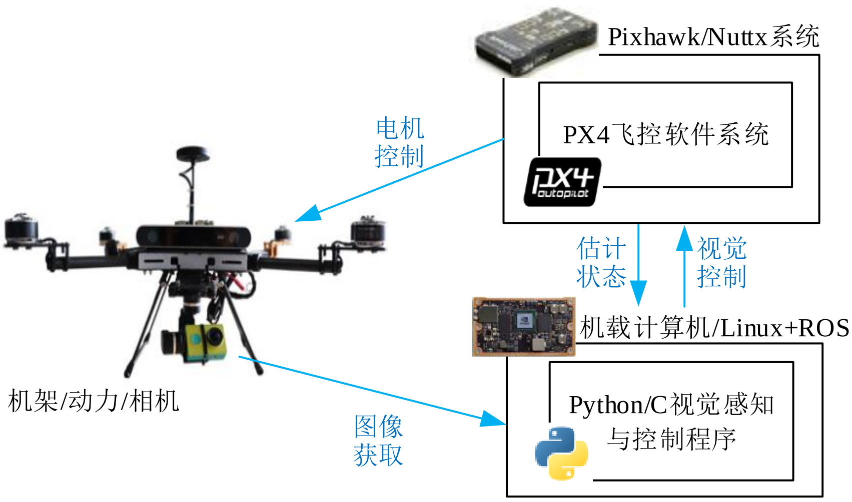
3.2.2. Multi-machine HIL Simulation phase

The whole simulation system at this stage can be regarded as composed of multiple HIL simulation subsystems, but it is not a simple superposition of subsystems. It is necessary to consider the system topology and configuration between different hardware, the model structure of network and communication, and the resource scheduling and management of the simulation host. In the process of simulation, it is necessary to properly configure and debug each hardware device to ensure that each hardware device can work normally. Whether from the perspective of multi-machine HIL simulation or from the perspective of real cluster control of unmanned systems, communication bandwidth and computing performance are always important bottlenecks restricting the increase of the number of clusters. Because of the performance bottleneck of the simulation computer, and a single computer can only connect a limited number of Pixhawks for HIL simulation. At the same time, with the increase of the number of UAVs, the amount of communication data between aircraft increases dramatically until the communication bandwidth reaches saturation. Therefore, the RflySim platform realizes arbitrary expansion of the number of UAVs by networking multiple computers, divides the whole UAV cluster into several subgroups, and realizes larger-scale cluster simulation by network layering, as 错误!未找到引用源。shown in.



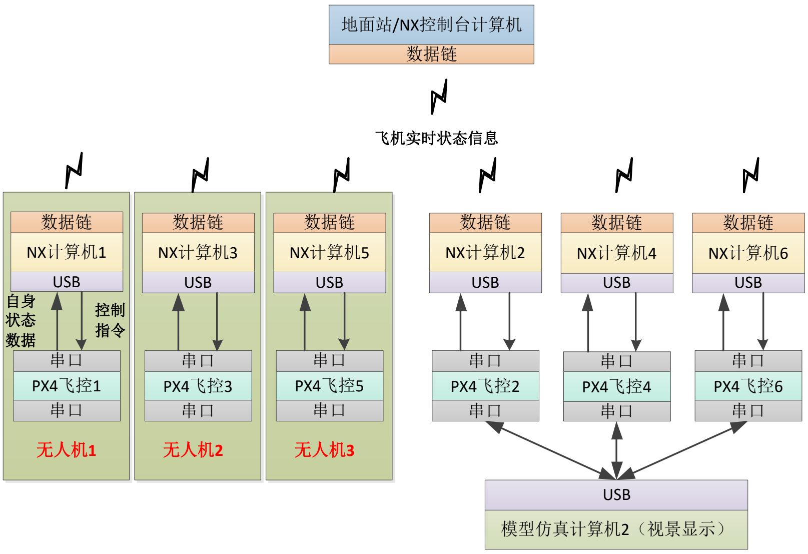
3.2.3. Single Machine Autonomous Control Phase

Single machine autonomous control refers to the ability of a single machine to independently plan and execute a mission without human intervention. For example, an autonomous UAV should have an internal and external state awareness system, an internal communication link between airborne systems, an airborne fault management system, and a mission re-planning system [错误!未找到引用源。] in the face of changes in the battlefield environment. Based on the RflySim platform, top-level control algorithms in the field of unmanned systems can be quickly developed, such as SLMA algorithm, trajectory planning algorithm, obstacle avoidance algorithm and so on.



3.2.4. Hardware-in-the-loop cluster control phase

The hardware-in-the-loop cluster control stage refers to the real-time simulation of part of the physical objects in the simulation loop of the simulation experiment system. It is a typical application in the development process of complex control system, and can be used to verify and optimize the decision-making algorithm of cluster collaborative control. Section 10.3.5 of this book is a multi-UAV hardware-in-the-loop simulation case. The experiment is based on the development and verification of UAV cluster ultra-low altitude collision avoidance algorithm. The fixed-wing UAV hardware-in-the-loop simulation experiment of "3 real and 3 virtual" is used to complete the experimental tasks of UAV cluster ultra-low altitude collision avoidance algorithm transplantation verification, UAV digital twin model development, virtual and real collaboration, etc.



3.2.5. Real machine cluster control stage

The real machine cluster control stage refers to the experiment or test of cluster control using real equipment in a real environment, which can verify the performance and reliability of the syste m. Many practical factors need to be considered in the real machine cluster control stage, such as c ommunication, interference, failure, safety and so on. For example, the optical motion capture syst em is used for indoor motion capture of the UAV to capture the motion information of the UAV, su ch as position, attitude, speed, etc. It can be used to study and verify the motion control, navigatio n, formation, coordination and other functions of UAV. Generally, it is necessary to use high-speed and high-resolution cameras and paste reflective marks on UAVs to achieve high-precision, real-ti me and high-stability dynamic capture effects.

3.2.6. Multi-machine Cooperation Phase in Completely Real Environment

This stage refers to the stage in which multiple UAVs are used in a real environment to compl ete a common task through inter-UAV communication and swarm intelligence. It is the highest lev el of UAV cluster collaboration technology and is used to study and verify the functions [错误!未找到引用]

源。 of UAV cluster, such as path planning, situational awareness, and task collaboration. High-performance, high-reliability and high-security UAVs, communication and control systems are needed to achieve high efficiency, high flexibility and high robustness in the stage of multi-UAV cooperation in the fully real environment. As 错误!未找到引用源。shown in the figure, it is the virtual and real combination simulation framework of UAV cluster.