Development and practice of intelligent unmanned cluster systemFull-stack development case based on RflySim Toolchain

Lecture 6 External control and trajectory planning

飞思实验室

Outline

1. Experimental platform configuration

2. Key Interface Introduction (Free version)

3. Basic Experiment Case (Free version)

4. Advanced Case Experiment (Collection Edition)

5. Extended Case (Full version)

6. Brief sum-up

1. Experimental platform configuration

1.1 Components to be installed

- Visual Studio 2017（ Both the experience and full versions need to be installed ）
- Configure the C++ compiler for MATLAB （ Both the experience and full versions need to be installed ）
- Matlab 2023a*（ Advanced full version installation ）

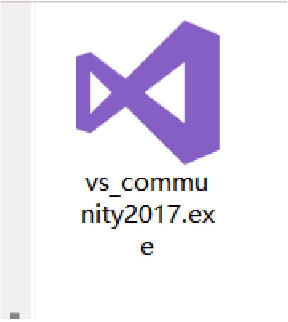
Here is how to install Visual Studio 2017 (requires networking) :

In this platform, the Visual Studio 2017 installation package has been placed

1. Experimental platform configuration

1.2 How to install Visual Studio 2017

- **First, we can open the platform installation location and find the *:\PX4PSP\RflySimAPIs location where some of the routines in the platform and the software installation package are placed**
- **After that, we can open the contents of Chapter 4 and find the basic version of the routine, 4.rflysimmodel \1.BasicExps, where we can find the folder named VS2017Installer, which is the installation package of Visual Studio 2017.**

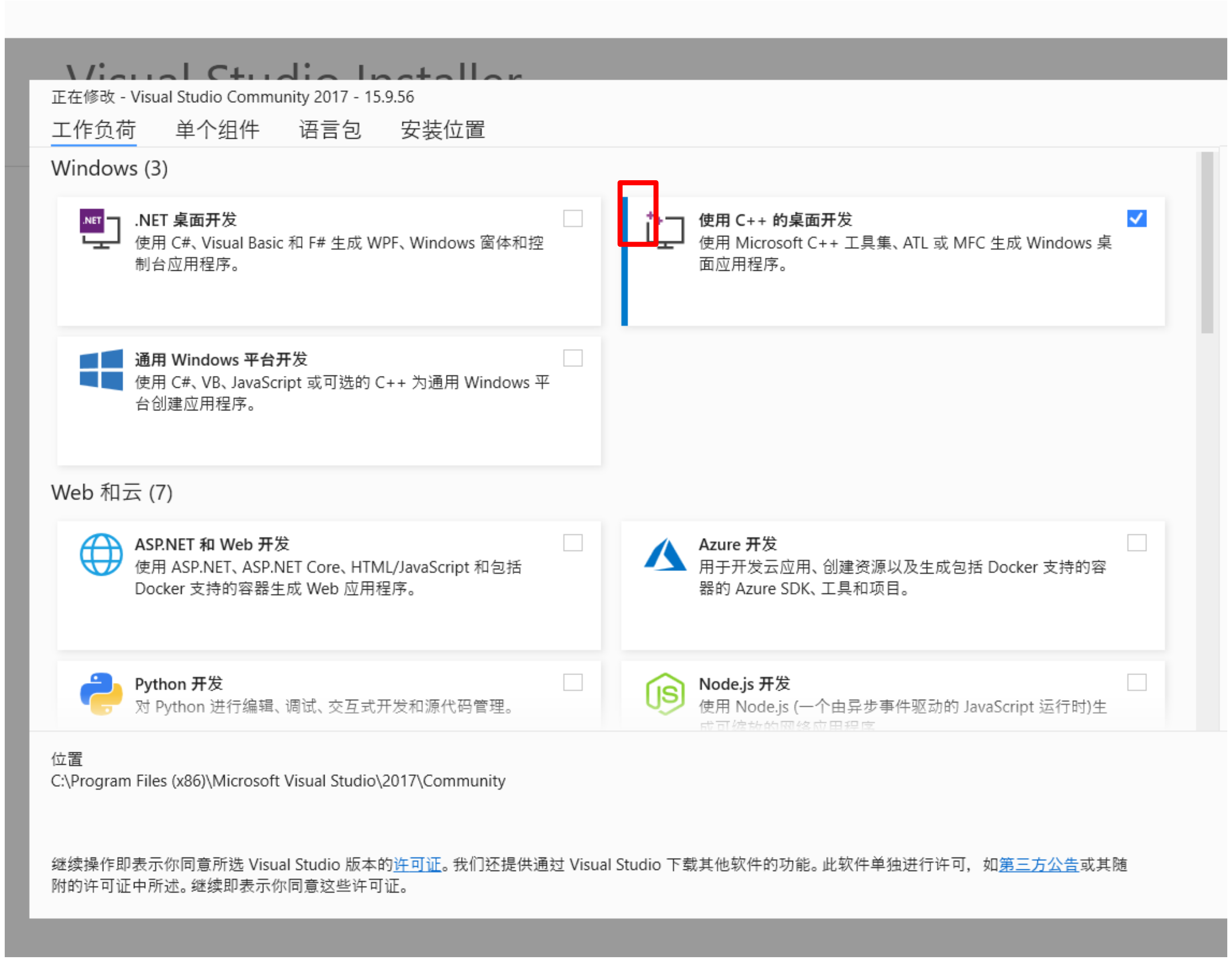


The online installation procedure (requires networking) is as follows: Double-click "RflySimAPIs\SimulinkControlAPI\VS2017Installer\vs_community2017.exe"

1. Experimental platform configuration

1.2 How to install Visual Studio 2017

- **Install Visual Studio 2017 (other versions can also be used, as long as MATLAB can recognize).**
- **In the following courses, Visual Studio compiler is needed in many places, such as the use of MATLAB S-Function Builder module and the automatic generation of C/C++ model code in Simulink**
- **For this course, just check "Desktop Development in C++" on the right.**



1.2 How to install Visual Studio 2017

- Note: VS2019 can also be installed on advanced versions of MATLAB, but MATLAB can only recognize Visual Studio versions lower than its own, so MATLAB 2017b has no way to recognize VS2019.
- Note: Please do not change the VS default installation directory (for example, to disk D), which will cause MATLAB to not recognize.
- You can't use the Mingw compiler, you need VS

1. Experimental platform configuration

- 1.3 Configure the C++ compiler for MATLAB

- Enter the instruction "mex-setup" in the command line window of MATLAB

- In general, the VS 2017 compiler is automatically identified and installed, as shown in the picture on the right, "MEX configuration uses' Microsoft Visual C++ 2017 'for compilation" indicates that the installation is correct

- If there are other compilers, this page can also be toggled to select other compilers such as VS 2013/2015

1. Experimental platform configuration

- **1.4 Matlab 2023a installation method**
- **MATLAB installation package download path:**
- **https://ww2.mathworks.cn/products/matlab.html**

Outline

1. Experimental platform configuration

2. Key Interface Introduction (Free version)

3. Basic Experiment Case (Free version)

4. Advanced Case Experiment (Collection Edition)

5. Extended Case (Full version)

6. Brief sum-up

2. Key interface introduction

2.0 Overview of basic experiments

Including basic function interface "RflySimAPIs / 6. RflySimExtCtrl / 0. ApiExps" as well as the basic routines "RflySimAPIs \ 6. RflySimExtCtrl \ 1. BasicExps"

See API.pdf and Readme.pdf for details

1.PX4MavCtrlAPITest	2023/11/16 10:17	文件夹
2.PX4ComAPITest	2023/11/16 10:59	文件夹
3.PX4MavGPSCtrlTest	2023/11/16 10:17	文件夹
4.PX4RcCtrlAPITest	2023/11/16 10:17	文件夹
5.PX4MultiUavTest	2023/11/16 10:17	文件夹
6.PX4MavAccCtrlTest	2023/11/16 10:17	文件夹
7.PX4MavAttCtrlTest	2023/11/16 10:17	文件夹
8.GeoAPITest	2023/11/16 10:17	文件夹
9.UDPMode1TestShootBall	2023/11/16 10:48	文件夹
10.UDPMode0Test	2023/11/16 10:17	文件夹
11.UDPMode1Test	2023/11/16 10:17	文件夹
12.UDPMode2DefaultTest	2023/11/16 10:17	文件夹
13.UDPMode3Test	2023/11/16 10:51	文件夹
14.UDPMode4Test	2023/11/16 10:58	文件夹
15.CamObjGet	2023/11/16 11:14	文件夹
16.ReadTimeStmpGet	2023/11/16 10:17	文件夹

e0_ExtAPIUsage	2023/11/16 10:17	文件夹
e1_PosCtrl	2023/11/16 10:17	文件夹
e2_VelCtrl	2023/11/16 10:17	文件夹
e3_RCCtrl	2023/11/16 10:17	文件夹
e4_PyOffboardCtrl	2023/11/16 17:41	文件夹
e5_RackFlyCtrl	2023/11/16 10:17	文件夹
e6_PathTrackingCtrl	2023/11/16 18:19	文件夹
e7_MutUAVRemoteCtrl	2023/11/16 10:17	文件夹

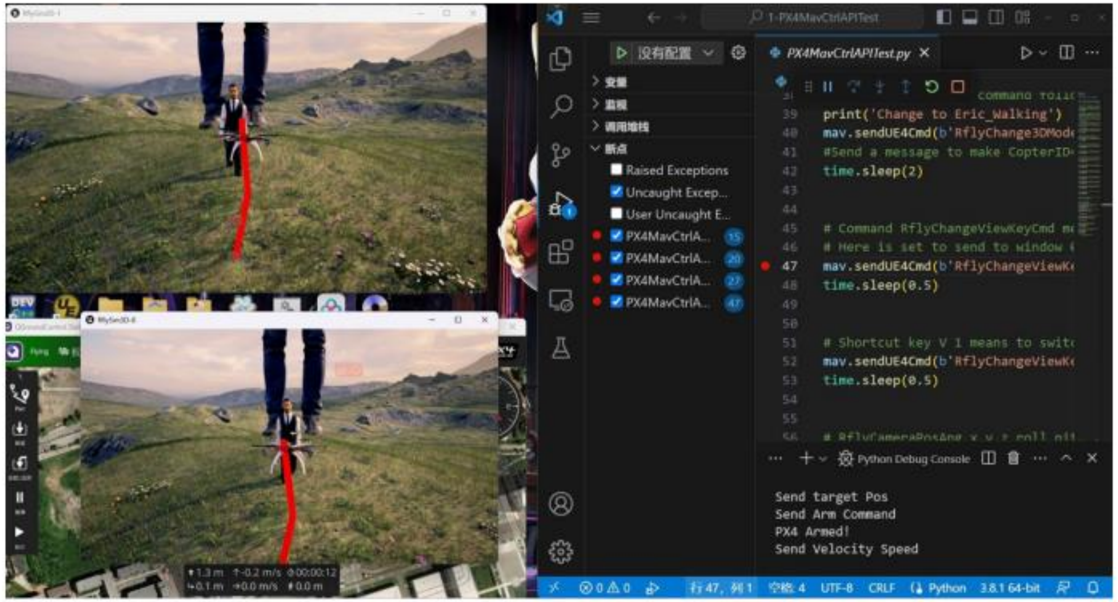
2. Key interface introduction

2.1 Uav control interface debugging experiment

Familiar with UAV offboard mode control, status data acquisition and RflySim3D control interface, understand the SITL communication framework.

See detailed operation and experimental results

[0.ApiExps\1.PX4MavCtrlAPITest\Readme.pdf]()



2. Key interface introduction

2.2 Data transmission connection Pixhawk 6C flight control hardware in the loop simulation experiment

Connect the computer to Pixhawk 6C flight control using MicroUSB wire to open a hardware-in-the-loop simulation of the aircraft.

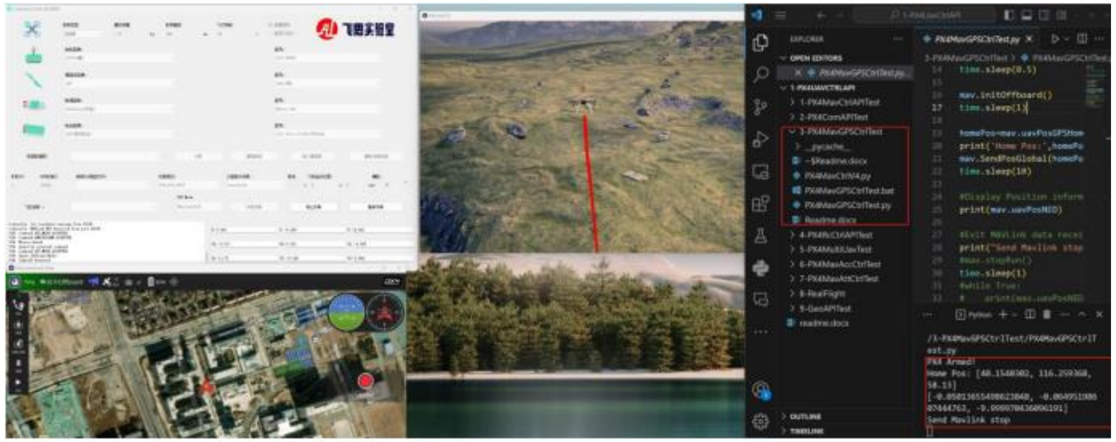
See detailed operation and experimental results 0.ApiExps\2.PX4ComAPITest\Readme.pdf

2. Key interface introduction

2.3 Uav flight control experiment

The SendPosGlobal function interface provided by RflySim platform is used to control the movement of UAV.

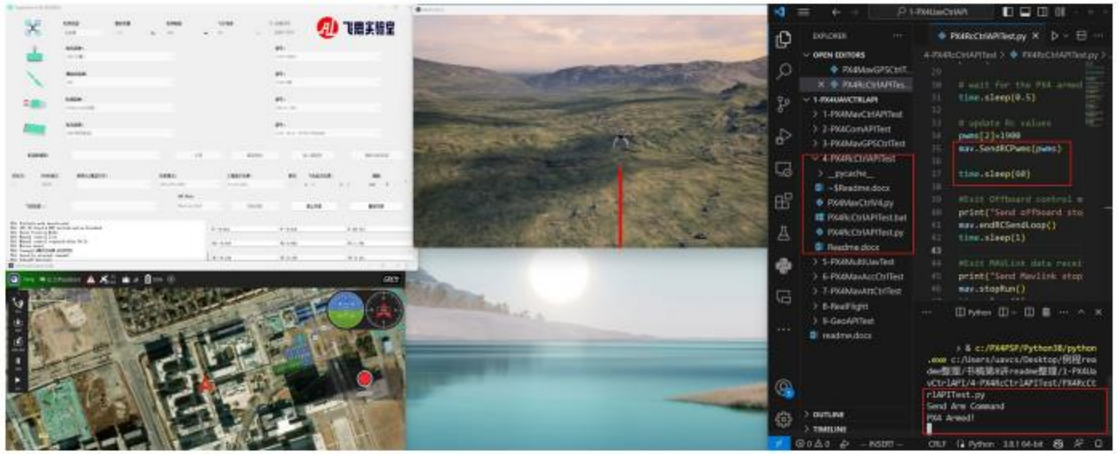
See detailed operation and experimental results [0.ApiExps\3.PX4MavGPSCtrlTest\Readme.pdf]



2. Key interface introduction

2.4 Uav motor speed PWM control experiment

Using the SendRCPwms function interface provided by the platform to control the PWM of the UAV motor, MAVLink is first opened to monitor the CopterSim data and update it in real time. Then set the PWM value, then turn on the RCOverride mode, start sending the RC pwms value, then unlock the drone for control, and finally, send the command to let the flight control out of the Offboard mode and stop listening to the MAVLink data. See detailed operation and experimental results 0.ApiExps\4.PX4RcCtrlAPITest\Readme.pdf

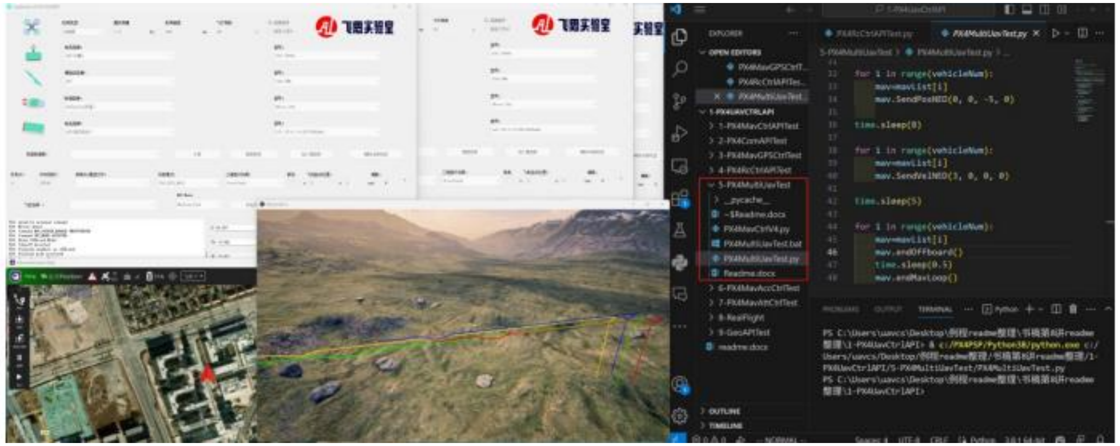


2. Key interface introduction

2.5 Multi-machine SITL software in the loop control experiment

According to the interface function provided by the platform, the position control and speed control SITL software of four aircraft in offboard mode are simulated in the loop.See detailed operation and experimental results [0.ApiExps\5.PX4MultiUavTest\Readme.pdf]

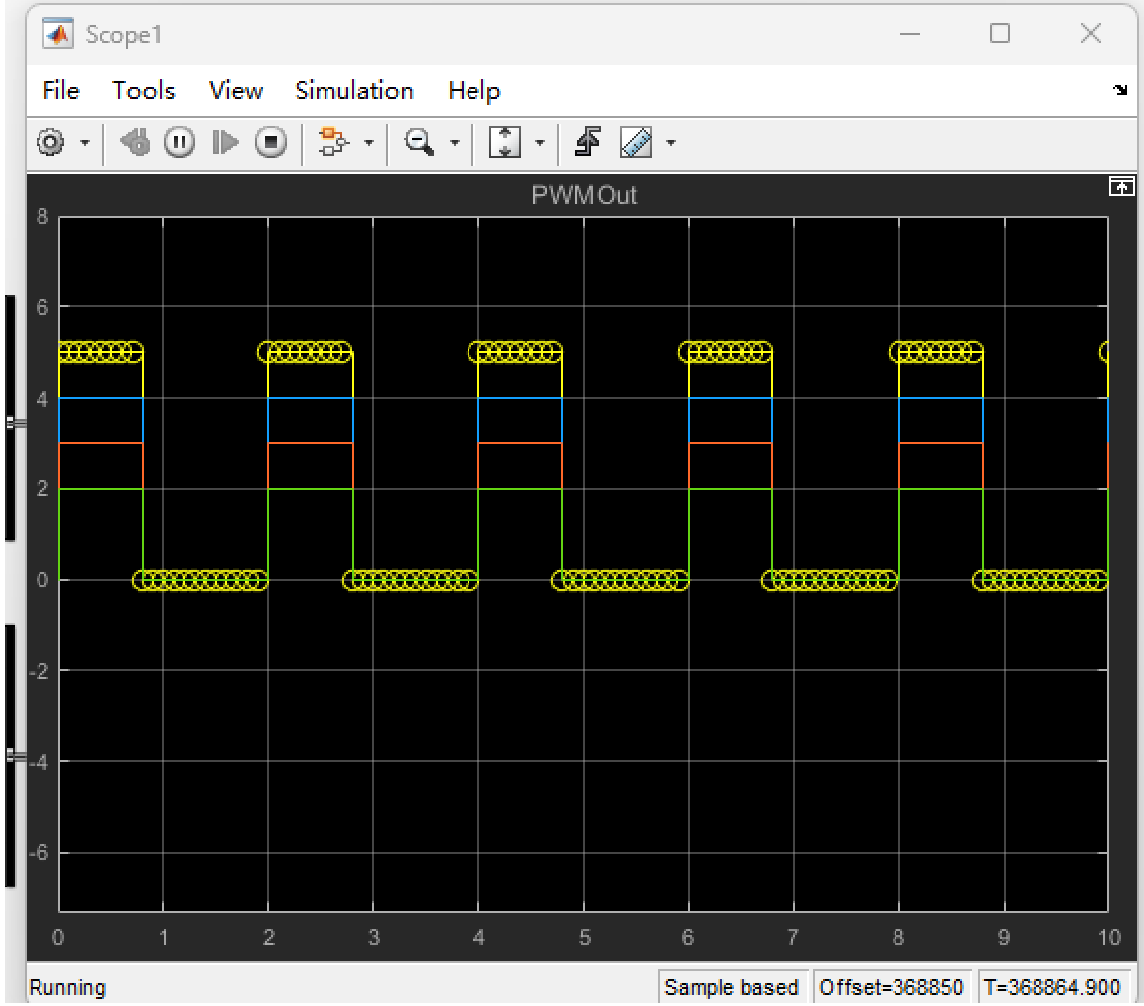
```
mavList[0].sendUE4Cmd(b'RflyChangeViewKeyCmd S')
mavList[0].sendUE4Cmd(b'RflyChangeViewKeyCmd T')

```



2. Key interface introduction

2.6 Uav flight acceleration control experiment

By using the interface function provided by the platform, the acceleration command is sent to the aircraft through the SendAccPX4 interface.See detailed operation and experimental results 0.ApiExps\6.PX4MavAccCtrlTest\Readme.pdf



2. Key interface introduction

2.7 Uav flight control experiment

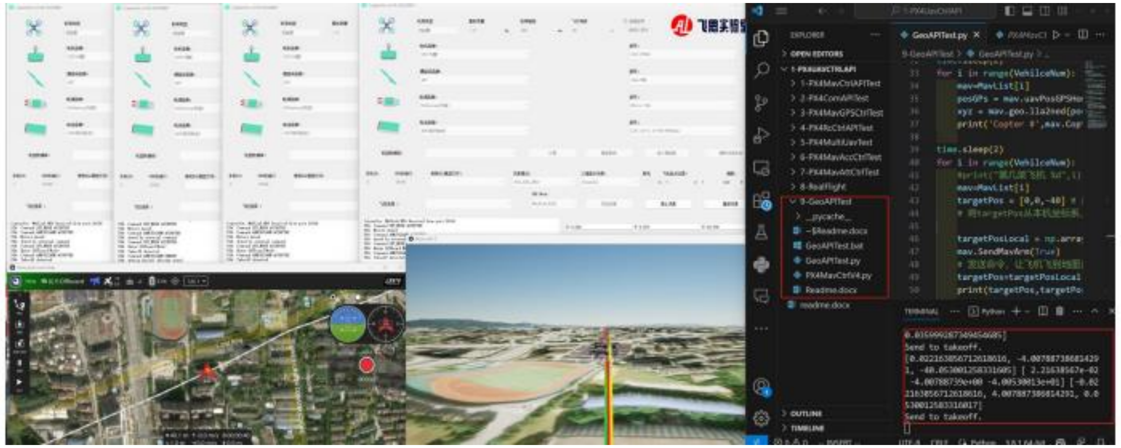
Send desired attitude and throttle data to the aircraft by utilizing the SendAttPX4 interface provided by the RflySim platform.See detailed operation and experimental results

[0.ApiExps\7.PX4MavAttCtrlTest\Readme.pdf]()

```python
mav.InitMavLoop()
time.sleep(0.5)

mav.initOffboard()
time.sleep(1)

mav.SendPosNED(0,0,-20)# 原地起飞，到20米高度
time.sleep(15)
```

```python
print('Current Thrust: ',mav.uavThrust) # 获取当前的悬停油门
mav.SendAttPX4([0,-10,0],mav.uavThrust)# 设置俯仰角为10度，油门为悬停值
print('Send attude command!')
time.sleep(2)
print('Current attitude: ', mav.uavAngEular[0]/math.pi*180, mav.uavAngEular[1]/math.pi*180, mav.uavAngEular[2]/math.pi*1
print('Current altitude: ',mav.uavPosNED[2])
time.sleep(5)


# From PX4 Web: Acceleration setpoint values are mapped to create a normalized
# thrust setpoint (i.e. acceleration setpoints are not "properly" supported).

#Display Position information received from CopterSim
print(mav.uavPosNED)
time.sleep(8)

mav.SendAttPX4([0,-10,0],-5,0,1)# 设置俯仰角为10度，保持高度为-5米
```

2.8 Experiment of UE map coordinate system and UAV coordinate system conversion

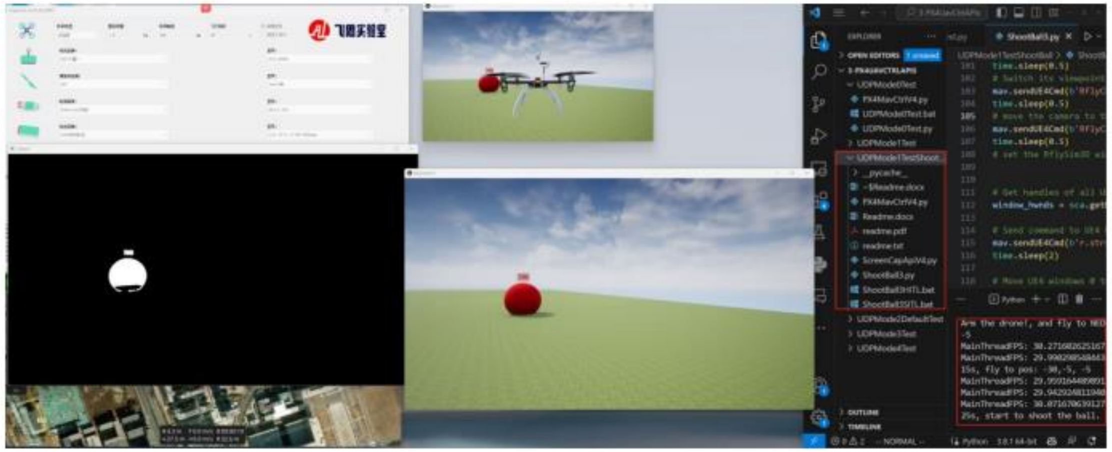


Familiar with UAV control origin and UE map origin coordinate system conversion.See detailed operation and experimental results 0.ApiExps\8.GeoAPITest\Readme.pdf

2. Key interface introduction

2.9 Visually-controlled impact ball experiment

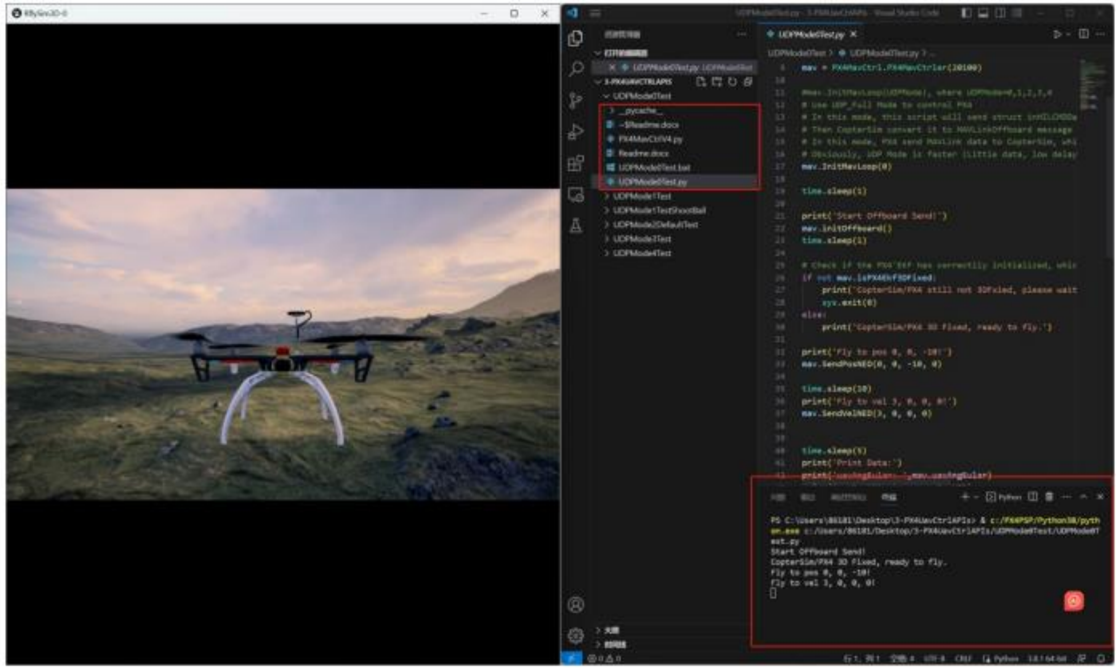
The image in RflySim3D software is captured by calling the platform interface, and the image is processed by opencv, and the control command is solved to control the UAV movement.See detailed operation and experimental results 0.ApiExps\9.UDPMode1TestShootBall\Readme.pdf



2. Key interface introduction

2.10 Drone through UDP Full communication experiment

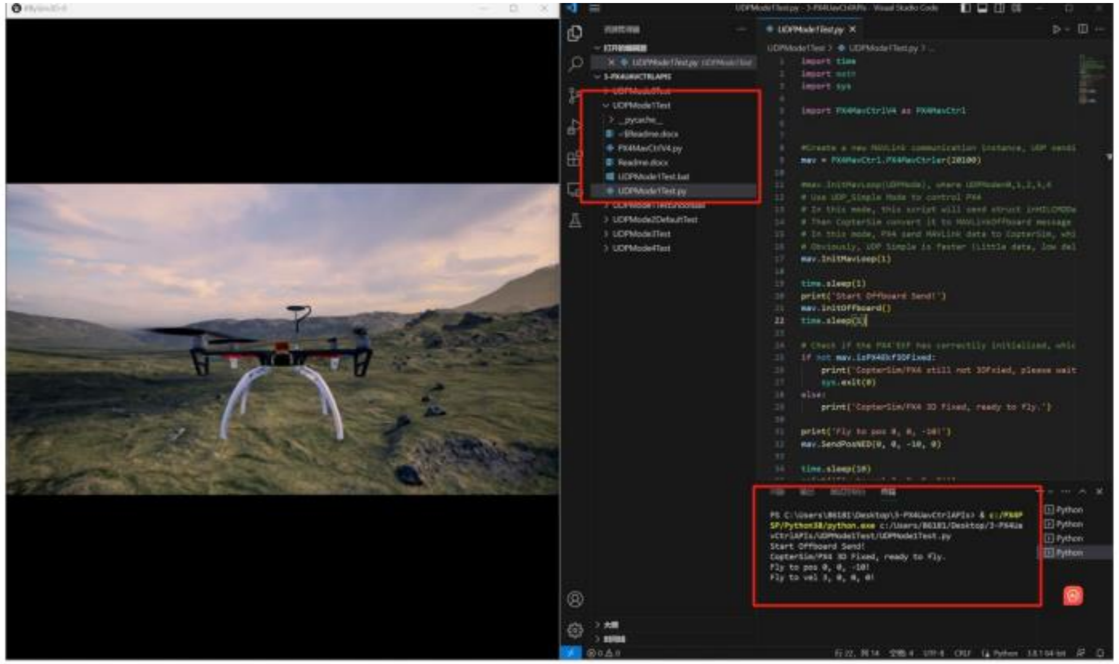
By using the interface function provided by the platform, commands are sent to the aircraft via UDP_Full communication.See detailed operation and experimental results [0.ApiExps\10.UDPMode0Test\Readme.pdf]



2. Key interface introduction

2.11 Uav Simple communication experiment via UDP

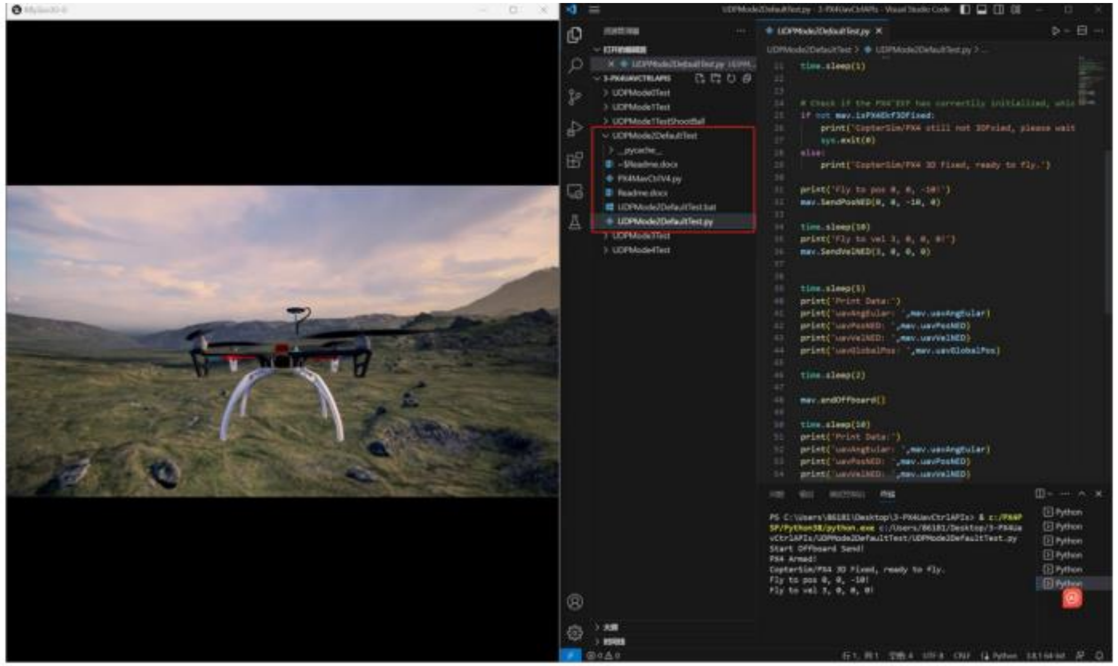
Commands are sent to the aircraft via UDP_Simple communication using interface functions provided by the platform.See detailed operation and experimental results [0.ApiExps\11.UDPMode1Test\Readme.pdf]



2. Key interface introduction

2.12 Drone through MAVLink Full communication experiment

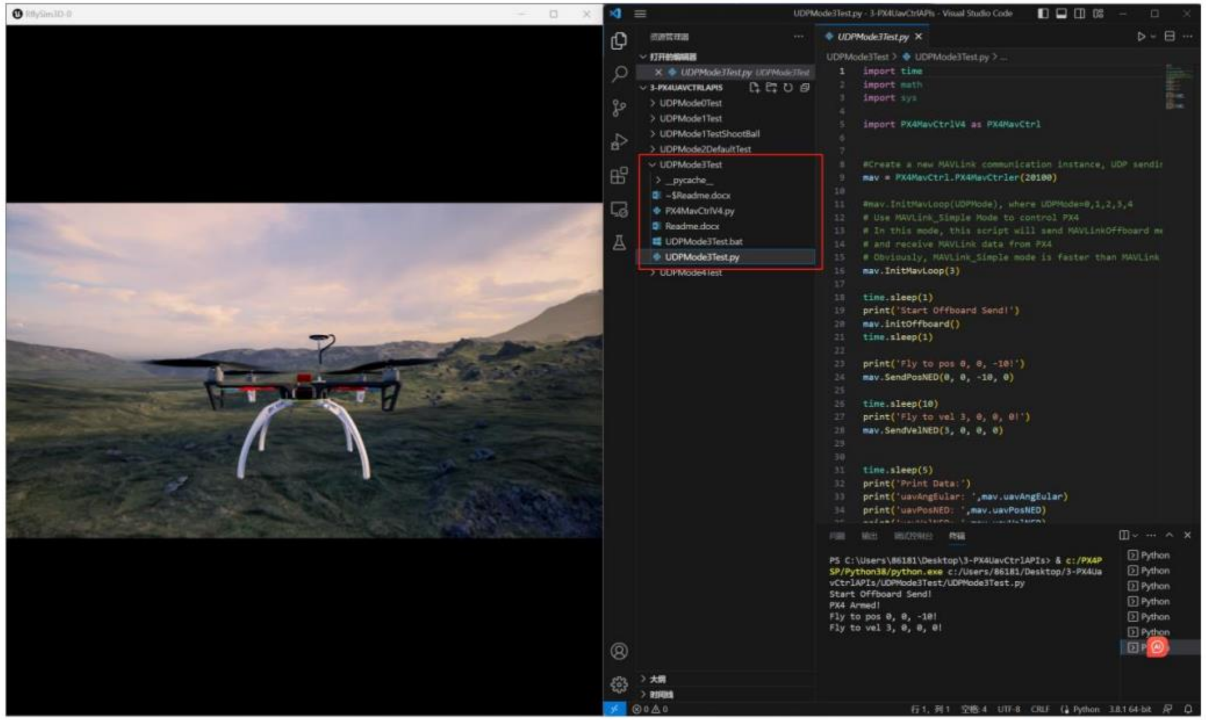
Commands are sent to the aircraft via MAVLink_Full communication using interface functions provided by the platform.See detailed operation and experimental results [0.ApiExps\12.UDPMode2DefaultTest\Readme.pdf]



2. Key interface introduction

2.13Drone via MAVLink Simple communication experiment

Commands are sent to the aircraft via MAVLink_Simple communication using interface functions provided by the platform.See detailed operation and experimental results 0.ApiExps\13.UDPMode3Test\Readme.pdf

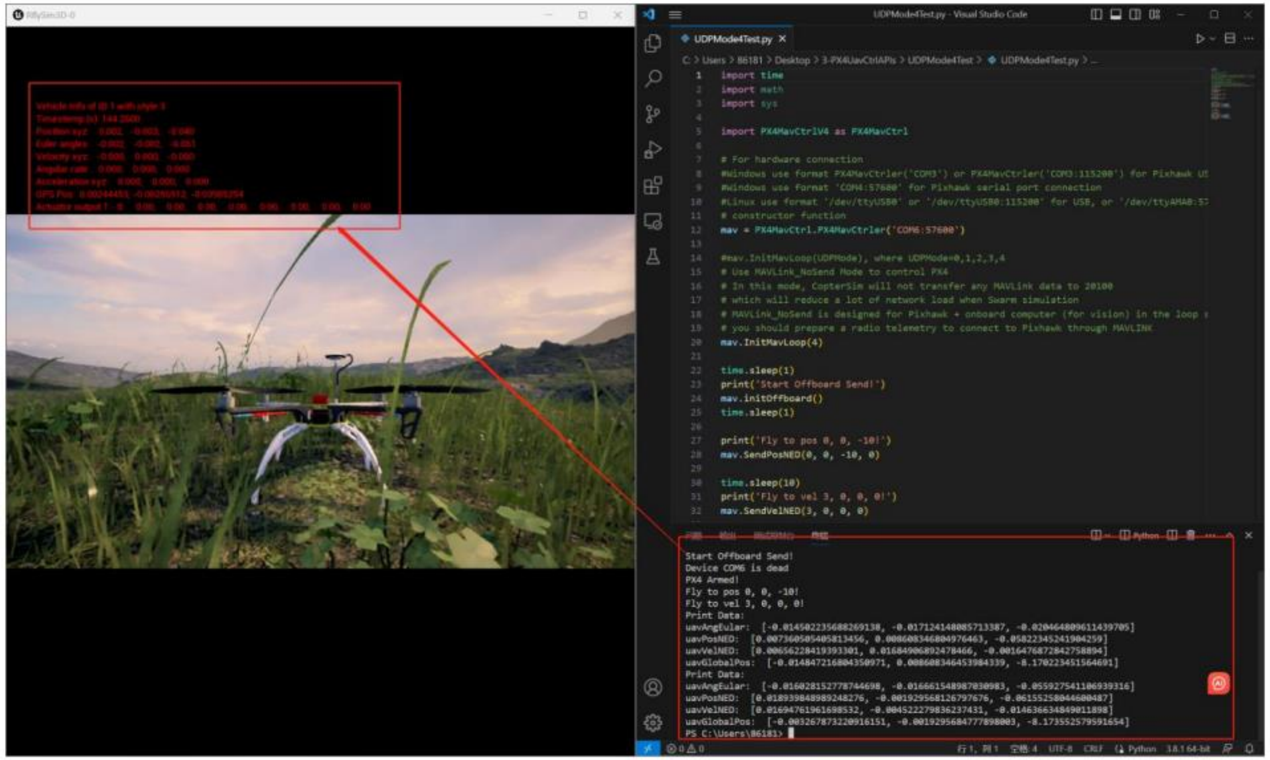


飞思实验室

2. Key interface introduction

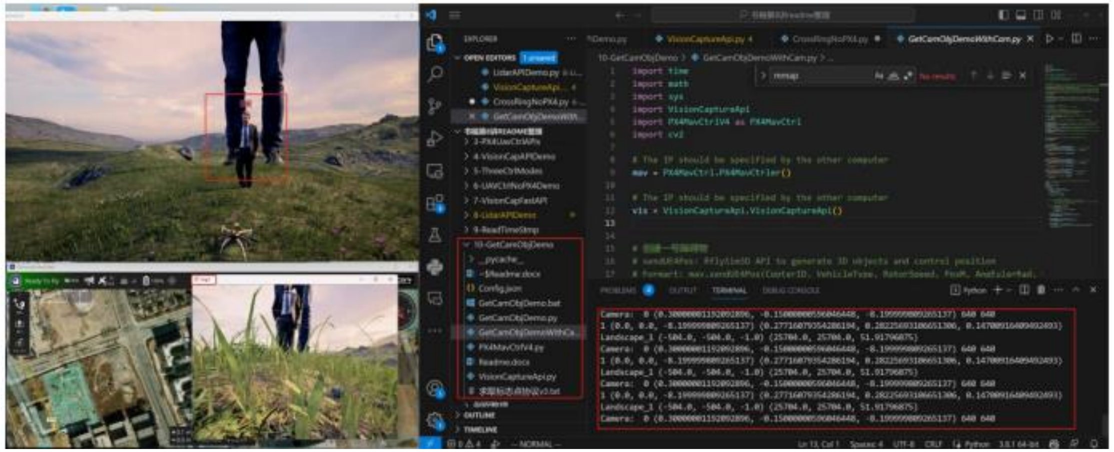
2.14 CopterSim-UDP communication mode

By using the interface function provided by the platform, the CopterSim sends commands to the aircraft through MAVLink_NoSend mode.See detailed operation and experimental results [0.ApiExps\14.UDPMode4Test\Readme.pdf]



2. Key interface introduction

2.15 Airplane, object, camera information acquisition experiment

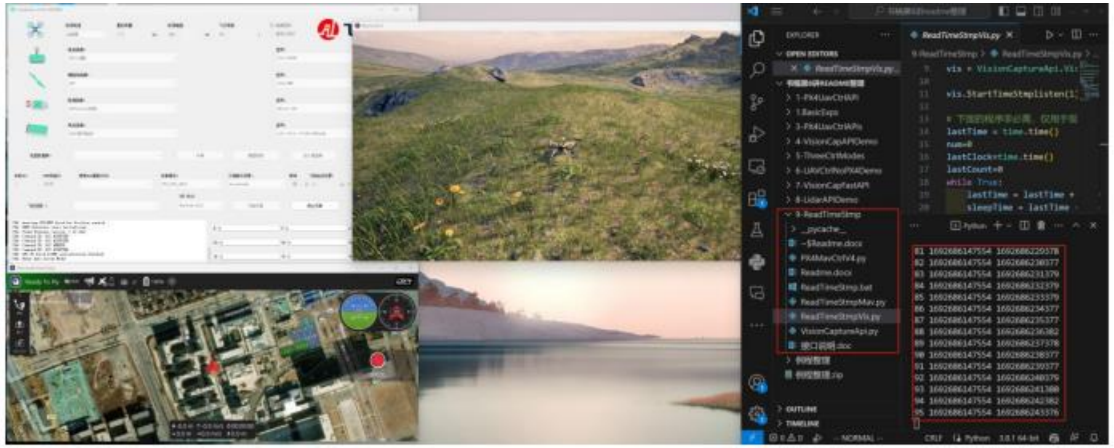


Get information about aircraft, objects, and cameras through the python interface.See detailed operation and experimental results 0.ApiExps\15.CamObjGet\Readme.pdf

2. Key interface introduction

2.16 Time stamp acquisition experiment

Start listening on port 20005 to get the rflytimestamp of CopterID by calling the StartTimeStmplisten(self,cpID=0) interface. If cpID= =0, only the current CopterID is listened to. If cpID >0, the timestamp of the required CopterID is listened for. It then calls the mav. RflyTime. SysCurrentTime attributes for unmanned aerial vehicle (uav) namely the current timestamp. See detailed operation and experimental results 0.ApiExps\16.ReadTimeStmpGet\Readme.pdf



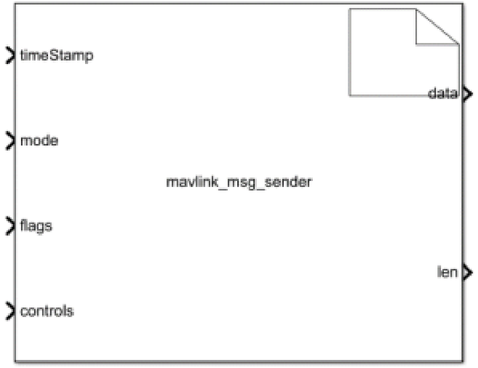
Outline

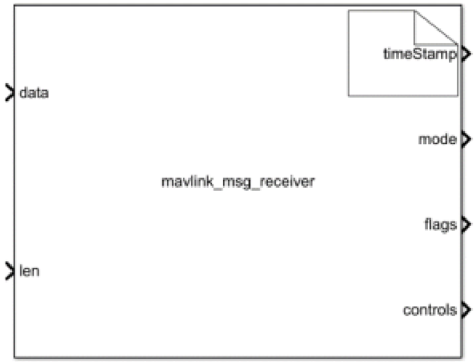
1. Experimental platform configuration
2. Key Interface Introduction (Free version)
3. Basic Experiment Case (Free version)
4. Advanced Case Experiment (Collection Edition)
5. Extended Case (Full version)
6. Brief sum-up

3. Basic Experiment Case

3.1.1 MAVLink module encapsulation experiment

MAVLink (Micro Air Vehicle Link) is a communication protocol for small unmanned vehicles, first released in 2009. The protocol is widely used in communication between Ground Control stations (GCS) and Unmanned vehicles, as well as in internal communication between the onboard computer and Pixhawk. The protocol defines the rules for parameter transmission in the form of a message library. The MAVLink protocol supports a variety of vehicles such as unmanned fixed-wing aircraft, unmanned rotorcraft, and unmanned vehicles. In this experiment, MAVLINK_MSG_ID_HIL_ACTUATOR_CONTROLS message is divided into two parts: data sending module and data parsing module based on Simulink. See detailed operation and experimental results 1.BasicExps\e0_ExtAPIUsage\1.MavLinkPackSimulink\Readme.pdf

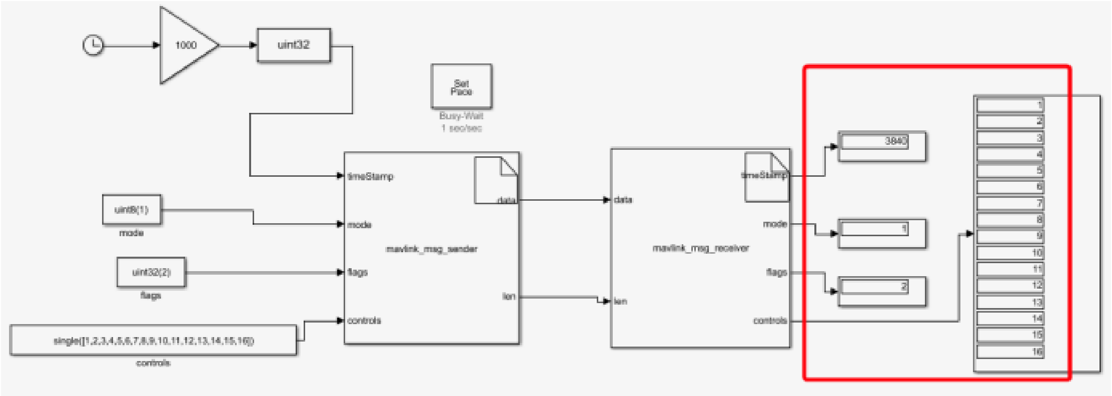




3. Basic Experiment Case

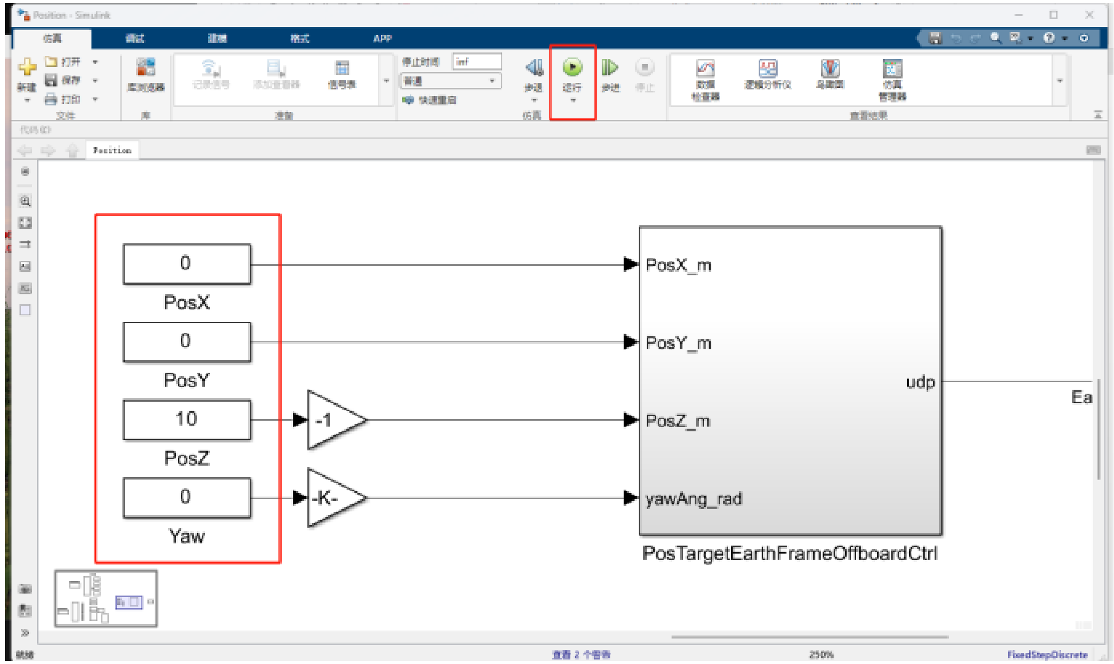
3.1.2 MAVLink data transmission experiment

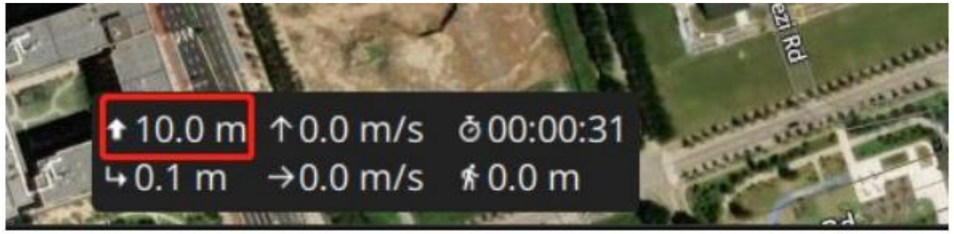
MAVLink (Micro Air Vehicle Link) is a communication protocol for small unmanned vehicles, first released in 2009. The protocol is widely used in the communication between Ground Control stations (GCS) and Unmanned vehicles, as well as in the communication between onboard computers and vehiclesIn the internal communication between Pixhawk, the protocol defines the rules for parameter transmission in the form of message libraries. The MAVLink protocol supports a variety of vehicles such as unmanned fixed-wing aircraft, unmanned rotorcraft, and unmanned vehicles. This experiment will be based on the two modules established in the experiment "*\PX4PSP\RflySimAPIs\7.RflySimExtCtrl\1.BasicExps\e0_ExtAPIUsage\1.MavLinkPackSimulink". Simulate sending and receiving MAVLINK_MSG_ID_HIL_ACTUATOR_CONTROLS messages.See detailed operation and experimental results 1.BasicExps\e0_ExtAPIUsage\2.MavlinkCodeDecode\Readme.pdf



3. Basic Experiment Case

3.2 Offboard mode control UAV position control experiment

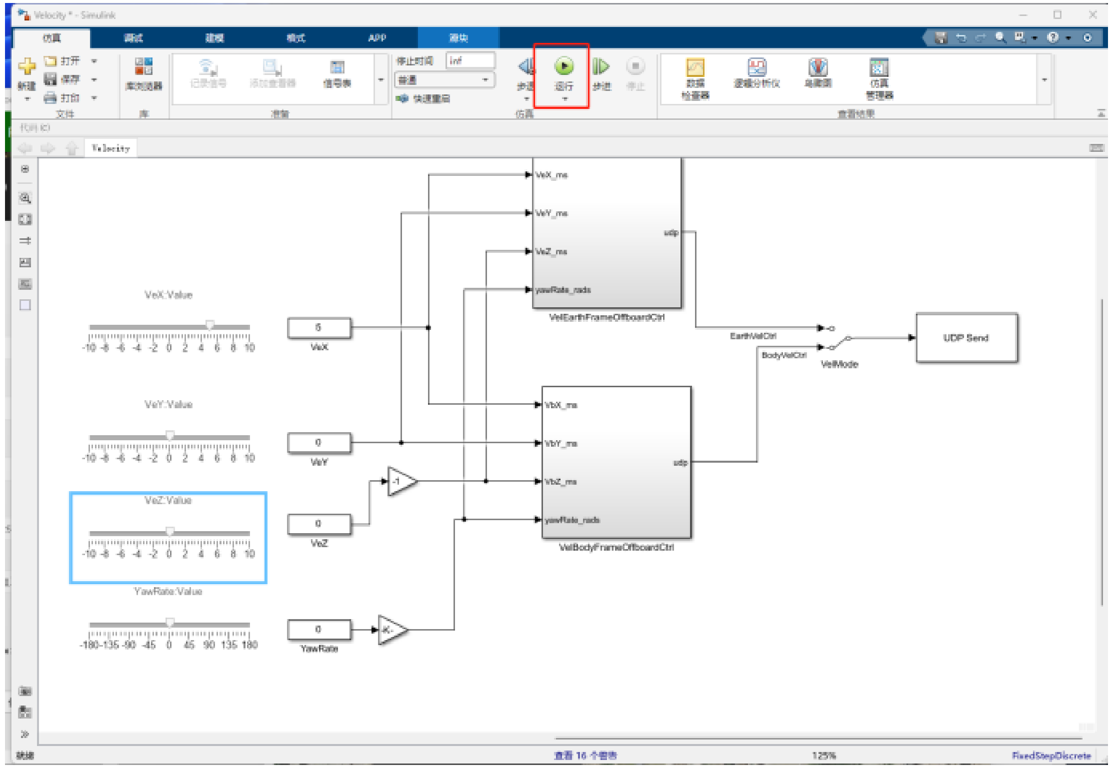




Offboard mode is a control mode of UAV, which usually gives the on-board computer or the ground computer (upper computer) real-time control of the aircraft's speed, position, attitude, etc. The aircraft can be regarded as a whole object, focusing on the top-level vision and cluster algorithm development. This experiment is mainly about position control experiment.See detailed operation and experimental results 1.BasicExps\e1_PosCtrl\readme.pdf

3.3 Offboard mode control UAV speed control experiment

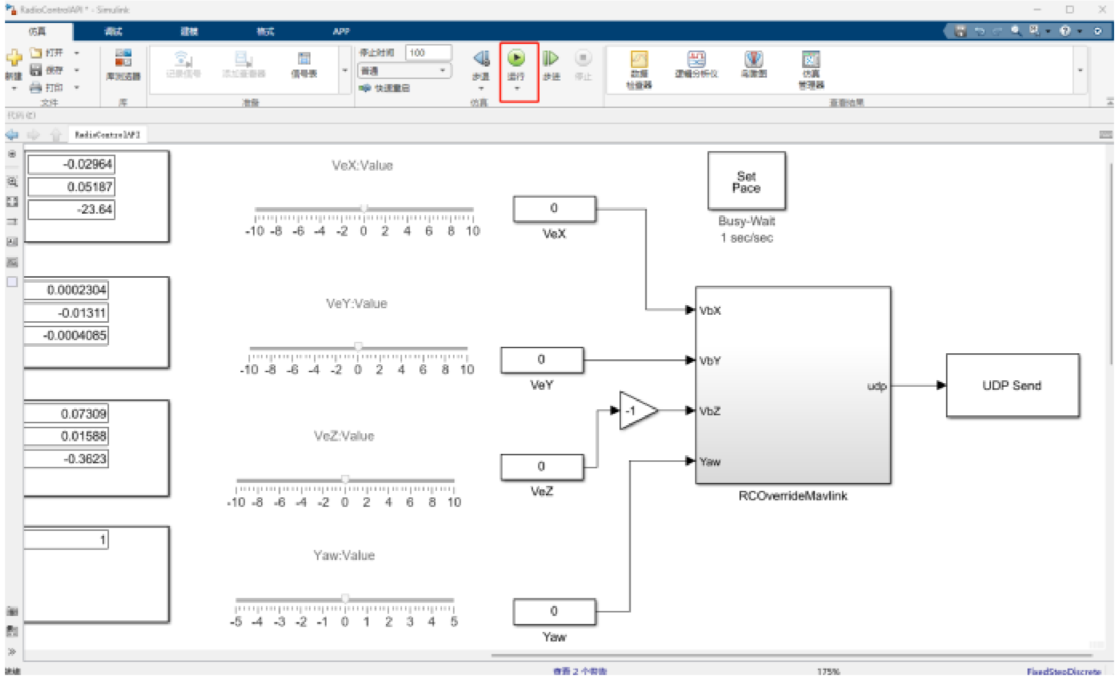
Offboard mode is a control mode of UAV, which usually gives the on-board computer or the ground computer (upper computer) real-time control of the aircraft's speed, position, attitude, etc. The aircraft can be regarded as a whole object, focusing on the top-level vision and cluster algorithm development. This experiment is mainly about speed control experiment.See detailed operation and experimental results [1.BasicExps\e2_VelCtrl\readme.pdf]



3. Basic Experiment Case

3.4 Remote control mode single machine control

Remote control mode is a kind of control mode for human operation of UAV, which has a good effect in some UAV stunt performances. The remote control used in this section is the operation mode of "American hand", that is, the throttle and yaw control quantity corresponding to the left rocker, while the right rocker corresponds to roll and pitch. This experiment is conducted by controller instead of remote control.See detailed operation and experimental results [1.BasicExps\e3_RCCtrl\readme.pdf]

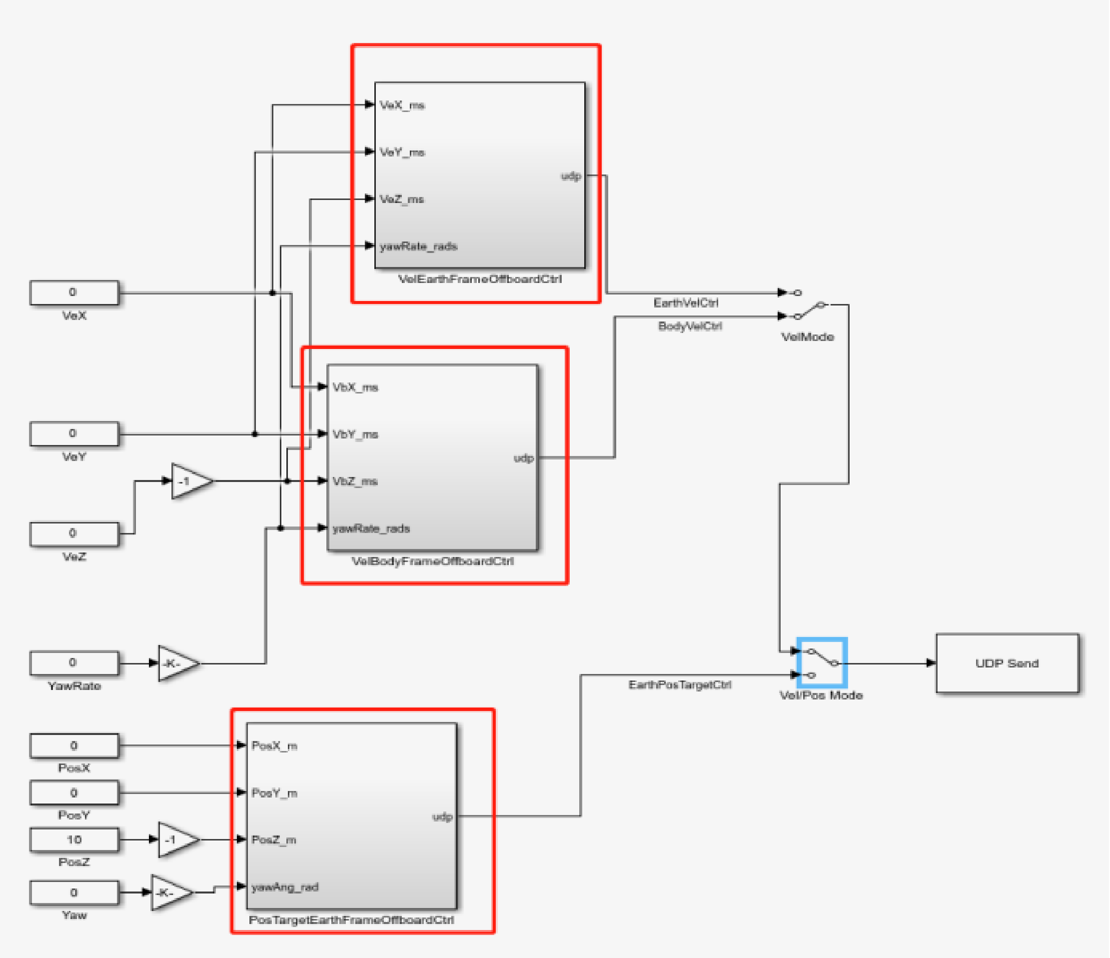


3. Basic Experiment Case

3.5 Python-Offboard single-machine control experiment

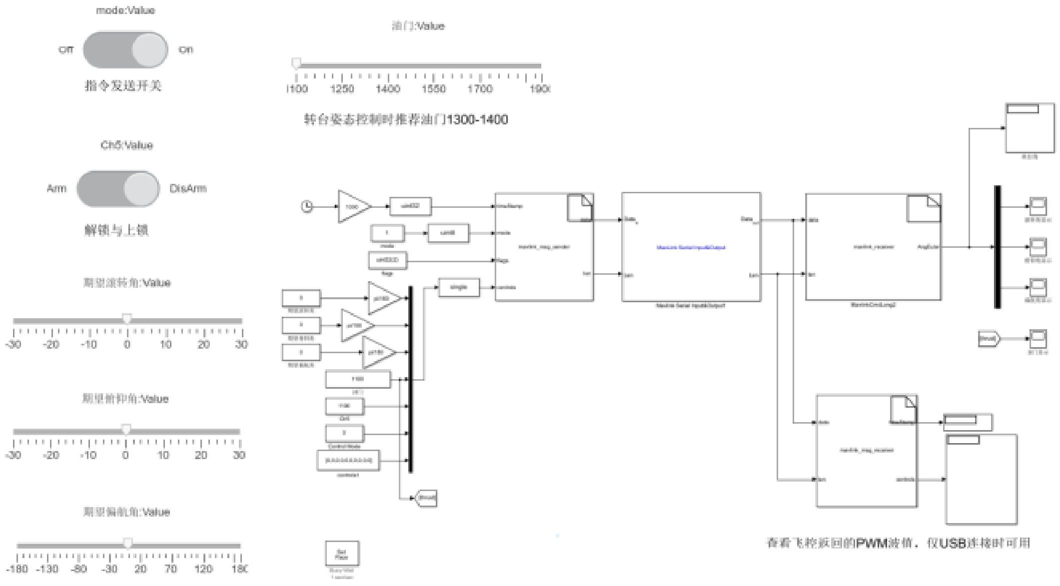
Offboard mode is a control mode of UAV, which usually gives the on-board computer or the ground computer (upper computer) real-time control of the aircraft's speed, position, attitude, etc. The aircraft can be regarded as a whole object, focusing on the top-level vision and cluster algorithm development. Python control UAV is to communicate with the UAV through a programming language, the basic principle is to establish communication through a serial port or network connection to the UAV to obtain the status information of the UAV and execute commands. A demonstration program that uses PX4's OffboardAPI to control the vehicle's expected speed and position.See detailed operation and experimental results

1.BasicExps\e4_PyOffboardCtrl\readme.pdf



3.6 Single control bench experiment

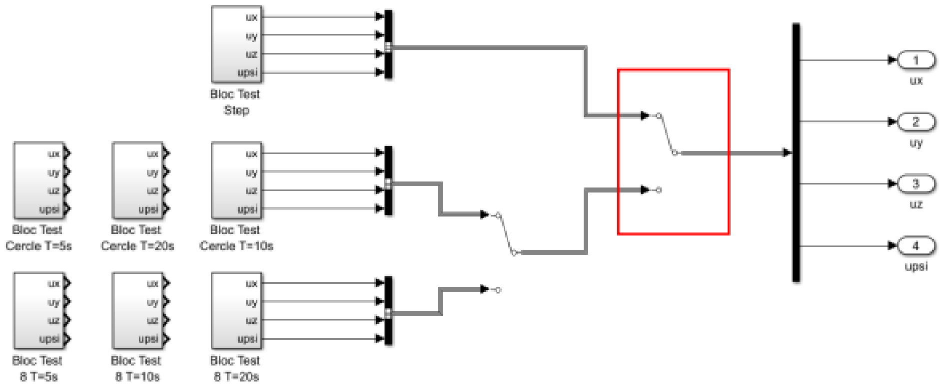
In this experiment, a multi-rotor flight controller was built in MATLAB/Simulink, and control commands were sent through Simulink to control the attitude of the four-rotor UAV on the rotary table. Proficient in MAVLINK communication application, proficient in four-rotor UAV attitude control and parameter setting. See detailed operation and experimental results 1.BasicExps\e5_RackFlyCtrl\Readme.pdf



飞思实验室

3.7 Simulation experiment of multi-rotor path tracking controller

Understand the given multi-rotor three-channel linearized transfer function simulation model and the corresponding trajectory tracking controller, and track tracking.See detailed operation and experimental results 1.Basic Exps\e6_PathTrackingCtrl\Readme.pdf



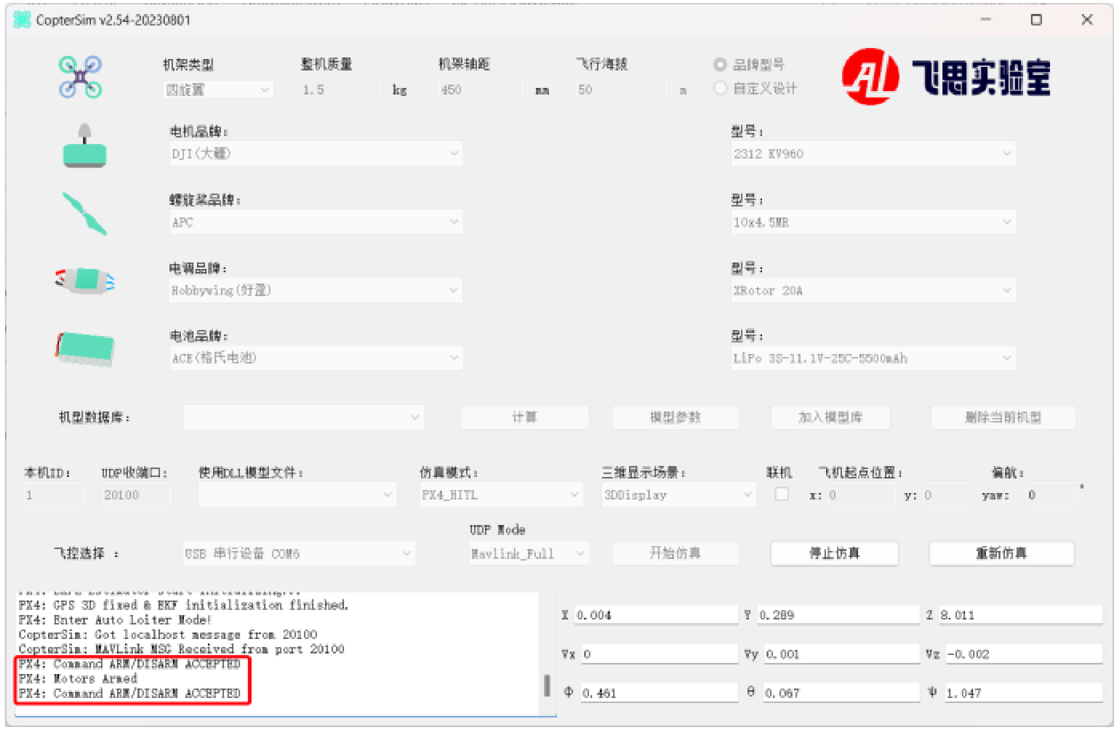
Outline

1. Experimental platform configuration
2. Key Interface Introduction (Free version)
3. Basic Experiment Case (Free version)
4. Advanced Case Experiment (Collection Edition)
5. Extended Case (Full version)
6. Brief sum-up

4. Advanced Case Experiment

4.1MAVSfun unlocks HIL experiments

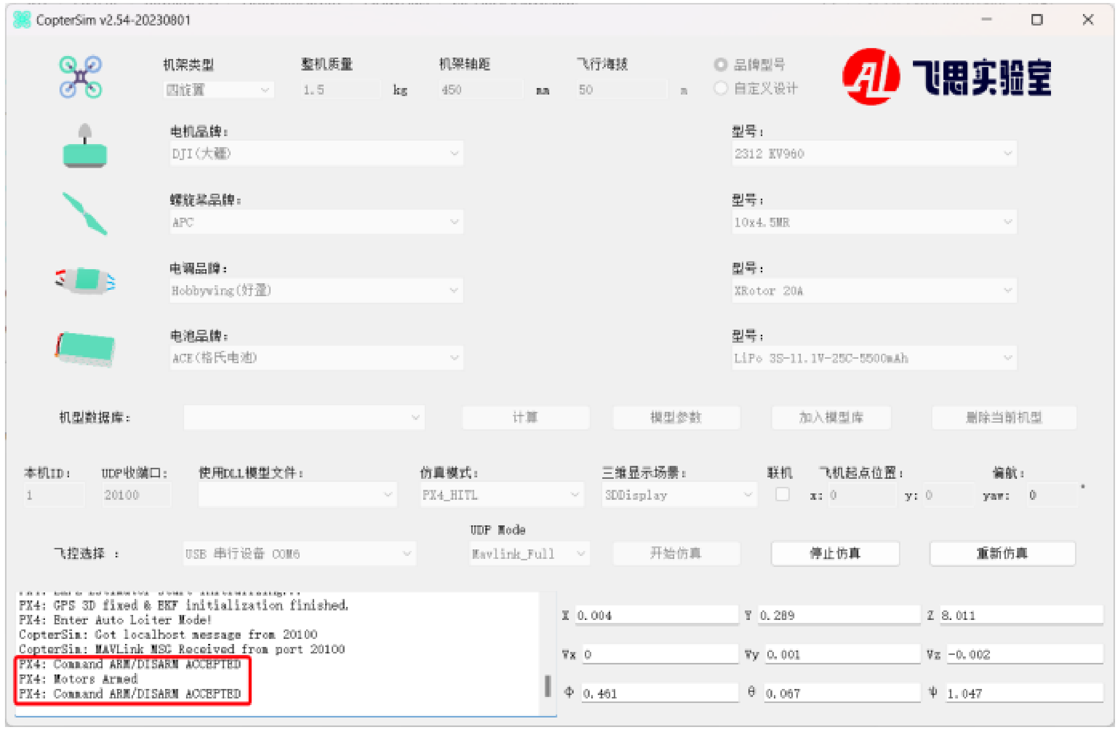
MAVLink (Micro Air Vehicle Link) is a communication protocol for small unmanned vehicles, first released in 2009. The protocol is widely used in communication between Ground Control stations (GCS) and Unmanned vehicles, as well as in internal communication between the onboard computer and Pixhawk. The protocol defines the rules for parameter transmission in the form of a message library. The MAVLink protocol supports a variety of vehicles such as unmanned fixed-wing aircraft, unmanned rotorcraft, and unmanned vehicles. In this experiment, the unlocking information will be displayed in CopterSim software through MAVLink encapsulation module and UDP during the hardware-in-the-loop simulation.See detailed operation and experimental results 2.AdvExps\e1_ExtAPIAdvUsage\1.MavSfunTest_Arm\Readme.pdf



4. Advanced Case Experiment

4.2 MAVLink controlled HIL experiment

MAVLink (Micro Air Vehicle Link) is a communication protocol for small unmanned vehicles, first released in 2009. This experiment will be based on CopterSim software in the hardware in the loop simulation, through MAVLink encapsulation module UDP, to achieve UAV attitude control. See detailed operation and experimental results 2.AdvExps\e1_ExtAPIAdvUsage\2.MavSfunTest_Con\Readme.pdf



Outline

1. Experimental platform configuration
2. Key Interface Introduction (Free version)
3. Basic Experiment Case (Free version)
4. Advanced Case Experiment (Collection Edition)
5. Extended Case (Full version)
6. Brief sum-up

5. Extended Case

Being developed

Outline

1. Experimental platform configuration

2. Key Interface Introduction (Free version)

3. Basic Experiment Case (Free version)

4. Advanced Case Experiment (Collection Edition)

5. Extended Case (Full version)

6. Brief sum-up

6. Brief sum-up

- **This lecture mainly explains the external control and trajectory planning of UAV system, which is divided into three parts: basic experiment, advanced experiment and extended case, which can realize model fault injection and flight control source code injection course.**
- **If in doubt, please visit https://doc.rflysim.com/ for more information.**

RflySim更多教程

扫码咨询与交流

飞思RflySim技术交流群

飞思实验室

Thanks！