
API 说明文件检索大纲

1. RflySim 工具链核心组件介绍	1
1.1. CopterSim	1
1.1.1. 模型配置区	2
1.1.2. 仿真功能区	3
1.1.3. 状态显示区	4
1.1.4. 仿真控制接口 ReqCopterSim.py	4
1.2. RflySim3D/RflySimUE5	6
1.2.1. 快捷键	7
1.2.2. 快捷命令	8
1.2.3. 地形校准文件—***.txt 获取	8
1.2.4. 地形高程文件—***.png 获取	8
1.3. QGroundControl 地面站	9
1.3.1. QGC 初始界面介绍	9
1.3.2. 载具设置 (Vehicle Setup)	10
1.3.3. 数据分析 (Analyze Tools)	17
1.4. Python38Env	20
1.5. MATLAB 自动代码生成工具箱	20
1.6. SITL/HITL 批处理脚本	21
1.6.1. HITLRun.bat	21
1.6.2. SITLRun.bat	22
1.6.3. HITLPos.bat	22
1.6.4. SITLPos.bat	22
1.6.5. HITLPosStr.bat	22
1.6.6. HITLRunSysID.bat	22
1.6.7. HITLPosSysIDStr.bat	22
1.6.8. HITLPosSysID.bat	23
1.6.9. HITLRunLowGPU.bat	23
1.6.10. SITLRunLowGPU.bat	23
1.6.11. HITLRunMAVLink.bat	23
1.6.12. SITLRunMAVLink.bat	23
1.6.13. HITLRunNoUI.bat	23
1.6.14. SITLRunNoUI.bat	23
1.6.15. HITLPosAlt.bat	23
1.6.16. SITLPosAlt.bat	23
1.6.17. HITLPosAltStr.bat	23
1.6.18. SITLPosAltStr.bat	23
1.6.19. HITLRunChange3D.bat	24
1.6.20. SITLRunChange3D.bat	24
1.6.21. HITLPosStrGPS.bat	24
1.6.22. SITLPosStrGPS.bat	24
1.7. PX4 Firmware 源码	24
1.8. WinWSL 子系统	25

1.9.	Simulink 集群控制接口	25
1.10.	RflySim 配套资料文件	26
2.	RflySim 工具链配套硬件系统	26
2.1.	飞思系列飞机.....	26
2.1.1.	飞思 X150 四旋翼无人机	26
2.1.2.	飞思 X200 四旋翼无人机	28
2.1.3.	飞思 X450 四旋翼无人机	29
2.1.4.	飞思 X680 四旋翼无人机	30
2.2.	PX4 系列飞控.....	32
2.2.1.	Pixhawk 2.4.8(又名 Pixhawk 1).....	32
2.2.2.	Pixhawk 6C.....	33
2.2.3.	Pixhawk 6X.....	34
2.3.	常用遥控器配置.....	35
3.	RflySim 工具链实验流程简介	39
3.1.	底层控制系统开发实验流程	39
3.2.	顶层控制系统开发实验流程	41
3.2.1.	机载板卡硬件在环阶段.....	41
3.2.2.	多机 HIL 仿真阶段	41
3.2.3.	单机自主控制阶段.....	42
3.2.4.	半实物集群控制阶段.....	42
3.2.5.	真机集群控制阶段.....	42
3.2.6.	完全真实环境下的多机协同阶段.....	43

1. RflySim 工具链核心组件介绍

RflySim 工具链包含了众多在进行无人系统建模、仿真、算法验证等开发过程中所涉及到的软件，其中，核心组件有 CopterSim、QGroundControl、RflySim3D/RflySimUE5、Python38Env、WinWSL 子系统、SITL/HITLRun 一键运行脚本、MATLAB 自动代码生成工具箱、Simulink 集群控制接口、PX4 Firmware 源码、RflySim 配套资料文件以及配套硬件系统。用户通过对这些核心组件的学习即可快速上手无人系统的开发和测试工作。

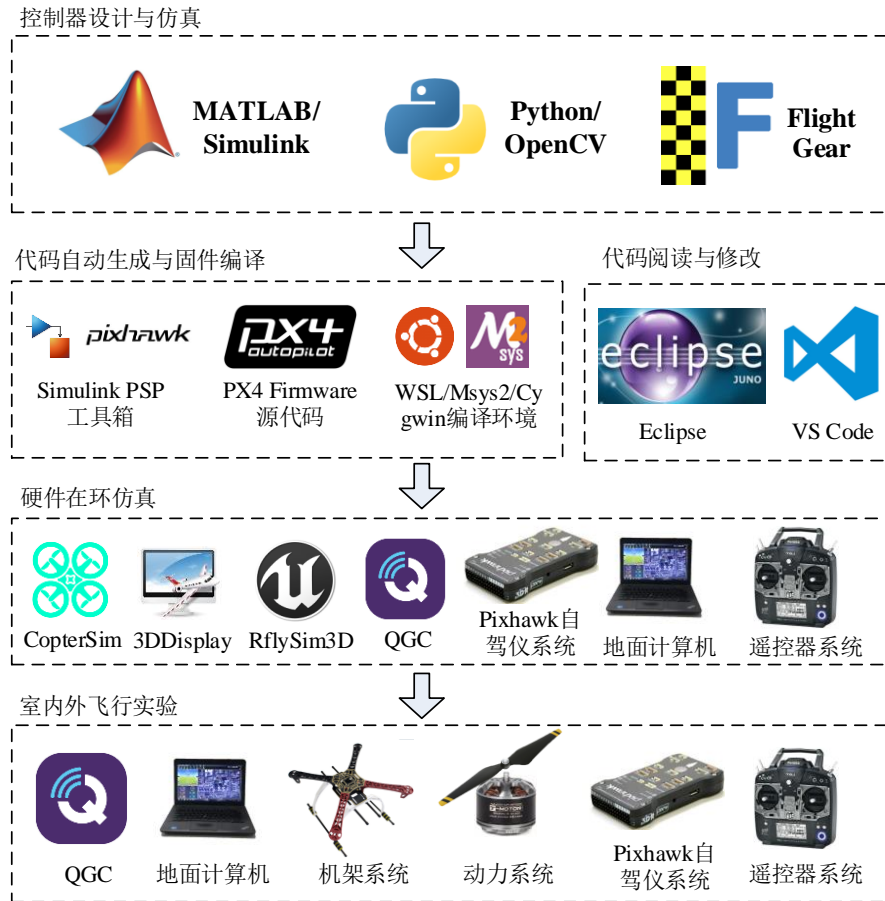


图 1 RflySim 软硬件组件与整体流程的相互关系

1.1. CopterSim

CopterSim 是 RflySim 工具链核心软件之一，它是针对 Pixhawk/PX4 自驾仪平台开发的一款硬件在环仿真软件，可以在软件中配置多旋翼的模型，通过 USB 串口与 Pixhawk 自驾仪连接来实现硬件在环仿真，达到室内模拟室外飞行测试的效果。主要由两大部分组成一模型和通信。模型是指可根据所设置的模型参数，进行计算后直接就可进行仿真；并支持运行动态模型(DLL)，并连同其他软件构成软/硬件在环仿真。CopterSim 是所有数据通信的中心；飞控与 CopterSim 通过串口（硬件在环 HITL）或网络 TCP/UDP（软件在环 SITL）进行连接，使用 MAVLink 进行数据传输，实现控制闭环，模拟室外飞行情形；CopterSim 发送飞机位姿、电机数据到三维引擎，实现可视化展示；转发 MAVLink 消息到 Python 视觉或 QGC 地面站，传输飞机实时状态，实现顶层规划控制；等等。同时，CopterSim 软件

对 MAVLink 数据进行压缩后以 UDP 结构体形式发给集群控制软件，达到通信精简目的（大规模集群需求）。

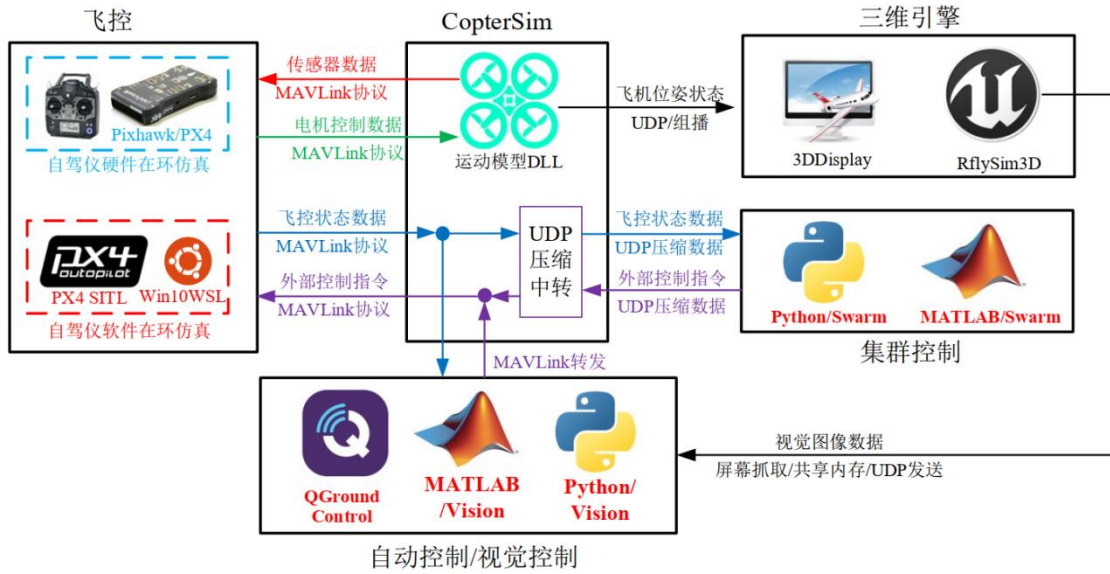


图 2 CopterSim 软件的数据通信结构图

CopterSim 的主界面主要分为模型配置区、仿真功能区、状态显示区三部分，如下图所示。



1.1.1. 模型配置区

可以配置定制多旋翼的构型、尺寸、重量等数据，CopterSim 将计算出指定多旋翼模型参数，实现不同机型的仿真。相关例程可见：[1.BasicExps\el1_CopterSim-Usage\Readme.pdf](#)

1.1.2. 仿真功能区

支持设置飞机的 ID、通信接口、仿真模式、三维场景、分布式联机仿真、地图初始位置、飞控 COM 串口选择、通信模式等配置；同时可以控制仿真的开始、暂停和重新启动；

- **本机 ID：**即每个飞机的标号
- **3DClassID：**飞机三维样式重映射选项，如果设为-1，则会使用 CopterSim 或 DLL 模型中默认的样式（对应 RflySim3D 的 XML 中 ClassID 标号）；如果设为其他值，则会重映射为本 ID。这里的“三维 ClassID”可以设置成“子样式 ID*100000 + 大类 ClassID”的形式。例如，ClassID=3 表示四旋翼大类，四旋翼下有很多的四旋翼样式小类，可以用 100003 表示 ClassID=3 的四旋翼下的 vehicleType=1 的 x330 样式。



- **使用 DLL 模型文件：**该接口用于选择不同的 DLL 文件，选项对应的文件夹地址为：`*PX4PSP\CopterSim\external\model`
- **仿真模式：**用于选择不同的仿真模式，主要分为如下几种：

PX4_HITL：此模式为 PX4 官方硬件在环仿真模式。
PX4_SITL：此模式为 PX4 官方软在环仿真模式。
PX4_SITL_RFLY：此模式为 RflySim 工具链自定义的软件在环仿真模式，由于 PX4_SITL 仅支持 10 架无人机集群仿真，在本模式下可支持上百架无人机进行 SITL 仿真。
Simulink&DLL_SIL：Simulink 中生成的 DLL 文件仿真模式。
PX4_HITL_NET（限完整版以上）：对于 Pixhawk 6x 等带网口的飞控，支持通过局域网连接飞控实现硬件在环仿真。
PX4_SIH_NET（限完整版以上，开发中）：支持飞控 SIH 模式（模型运行在飞控中，降低电脑运算量提升稳定性），NET 表示通过局域网连接飞控。
PX4_SIH_COM（限完整版以上，开发中）：支持飞控 SIH 模式（模型运行在飞控中，降低电脑运算量提升稳定性），COM 表示通过 USB 线连接飞控。
JSON_HITL_COM（限完整版以上，开发中）：支持通过 JSON 协议连接第三方飞控，实现硬件在环仿真，COM 表示使用 USB 串口方式连接飞控。
JSON_HITL_NET（限完整版以上，开发中）：支持通过 JSON 协议连接第三方飞控，实现硬件在环仿真，NET 表示使用网络的方式连接飞控（也可以连接 SITL 的第三方飞控）。
APM_SITL_NET（限完整版以上，开发中）：支持连接 APM 飞控（Ardupilot）进行 SITL 软件在环仿真，NET 表示 CopterSim 与飞控通过网络连接。
FPGA_HITL（限专业版以上，开发中）：支持通过 FPGA 替换飞控硬件的方式，进行硬件在环仿真。
注：在做大规模集群硬件在环时，电脑串口和供电（COM 方式）会影响每台电脑接入飞控数量，推荐使用 NET 后缀的方式。

- **三维显示地图：**用于选择在 RflySim3D 中的三维显示场景，该选项对应的文件夹地址为：`*PX4PSP\CopterSim\external\map`
注意：地形文件包含.png（uint16 灰度图）的二维高程图像，以及.txt 校准文件（描述像素范围对应 xy 宽度，0-uint16_max 对应高度范围等）。
- **联机：**该复选框点击后，将开启局域网内联机仿真。注：勾选本按钮，同时在 RflySim3D/RflySimUE5 中按下 I 键，就能开启局域网联机仿真，支持多台电脑组网显示所有飞机。
- **飞机起点位置：**可设置飞机的初始位置 XY 和 Yaw 角度。也可以通过 GPS 坐标（限完整版编辑）设定飞机的起始位置，适用于全球大地图仿真模式。
- **飞控选择：**此窗口仅在硬件在环仿真阶段进行使用，用于选择插入仿真电脑的飞控。
- **波特率（限完整版编辑）：**支持设定飞控 USB 仿真的波特率，本模式用于支持飞控通过串口连接时，线缆过长，需要使用低波特率的场景，例如，连接转台场景。
- **通信模式：**选择不同的 UDP 通信模式，主要分为如下几种。

UDP_Full: Python 传输完整的 UDP 数据给 CopterSim，传输数据量小；CopterSim 收到数据后，再转换为 Mavlink 后传输给 PX4 飞控；适合中小规模集群（数量小于 10）仿真。

UDP_Simple: 数据包大小与发送频率比 UDP_Full 模式小；适合大规模集群仿真，无人机数量小于 100。

Mavlink_Full: Python 直接发送 MAVLink 消息给 CopterSim，再转发给 PX4，数据量较大适合单机控制；适合单机或少量飞机仿真，无人机数量小于 4；

Mavlink_Simple: 屏蔽部分 MAVLink 消息包，并降低数据频率，发送数据量比 MAVlink_Full 小很多，适合多机集群控制；适合小规模集群仿真，无人机数量小于 8。

Mavlink_NoSend: 模式下 CopterSim 不会向外发送 MAVLink 数据，此模式需要配合硬件在环仿真+数传串口通信，通过有线方式传输 MAVLink，此模式局域网内数据量最小，适合分布式视觉硬件在环仿真，无人机数量不限制

Mavlink_NoGPS: 模式下 CopterSim 不会向外发送 MAVLink 数据和 GPS 数据。注：这种模式适合飞控 SLAM 仿真时，视觉感知算法需要视觉定位的 mavlink 消息发给飞控，形成无 GPS 环境下的视觉控制算法验证。

- **开始仿真/停止仿真/重新仿真。**

1.1.3. 状态显示区

左侧会显示模型和 Pixhawk 回传状态，右边是模型的仿真数据。一个代表性小实验，用于介绍 DLL 模型的导入功能。

1.1.4. 仿真控制接口 ReqCopterSim.py

RflySim 提供了 ReqCopterSim.py 接口库来获取 CopterSim 电脑 IP，发送指令对 CopterSim 相关参数进行信息重置。因此用户在使用 RflySim 工具链进行仿真时，除了通过 SITL/HTLRun.bat 脚本对初始化参数进行设置以外，还可以通过 python 的方式进行初始化参数设置，下面对 ReqCopterSim.py 接口库中关键函数进行介绍。

1) DLL 模型重置函数

CopterSim 所调用 DLL 模型的重置函数有两种，第一种为通过 DLL 模型名称进行重置：

```
2) def sendReSimDllName(self,CopterID=1,name=''):
3) 功能：该函数可以通过 DLL 模型名称修改 CopterSim 所调用的 DLL。
4) 说明：
5) self: 表明 ReqCopterSim 共享对象会影响该函数的行为；
6) CopterID: CopterSim ID 号，默认为 1。
```

- 7) name: DLL 模型名称。
- 8) 注: 要注意的是, 这里默认搜索的 DLL 模型名称路径为 C:\PX4PSP\CopterSim\external\model (以工具链安装在 C 盘为例进行说明), 因此在使用该接口修改 DLL 模型时要确保 C:\PX4PSP\CopterSim\external\model 路径下是否存在新的 DLL 模型。

9) 第二种为通过 DLL 模型序号进行重置:

- ```
10) def sendReSimDllIdx(self,CopterID=1,index=-1):
```
- 11) 功能: 该函数可以通过 DLL 序号来修改 CopterSim 所调用的 DLL。
- 12) 说明:
- 13) Self: 表明 ReqCopterSim 共享对象会影响该函数的行为;
- 14) CopterID: CopterSim ID 号, 默认为 1。
- 15) index: 选项 ID 序号, 这里对应了 DLL 的序号。

2) 三维场景重置函数

三维场景重置函数有两种, 一种为通过三维场景名称进行重置:

- ```
def sendReSimMapName(self,CopterID=1,name=''):
```
- 功能: 该函数可以通过地图名称来修改三维引擎仿真场景。
- 说明:
- Self: 表明 ReqCopterSim 共享对象会影响该函数的行为;
- CopterID: CopterSim ID 号, 默认为 1。
- name: 地图名称。

第二种为通过地图序号进行重置:

- ```
def sendReSimMapIdx(self,CopterID=1,index=-1):
```
- 功能: 该函数可以通过地图序号来修改三维引擎仿真场景。
- 说明:
- Self: 表明 ReqCopterSim 共享对象会影响该函数的行为;
- CopterID: CopterSim ID 号, 默认为 1。
- index: 选项 ID 序号, 这里对应了地图的序号。

3) 指定 CopterID 数据获取函数

该函数能指定本机或联机的 CopterID, 并请求该 Copter ID 将数据回传至本电脑。

- ```
def sendReSimIP(self,CopterID=1):
```
- 功能: 请求指定 CopterSim 将数据回传到本电脑, 适用于联机设置 Coptersim 消息。
- 说明:
- Self: 表明 ReqCopterSim 共享对象会影响该函数的行为;
- CopterID: 指定 CopterSim ID 号。

4) xyYaw 重设函数

- ```
def sendReSimXYyaw(self,CopterID=1,xyYaw=[0,0,0]):
```
- 功能: 请求指定 CopterSim 更换 x、y、yaw 为指定值, x、y 单位为米, 北东地, yaw, 单位为度
- 说明:
- Self: 表明 ReqCopterSim 共享对象会影响该函数的行为;
- CopterID: 指定 CopterSim ID 号。
- xyYaw=[0,0,0]: 分别为期望 x、y、yaw。

5) 初始化位置、姿态重设函数

- ```
def sendReSimXyzRPYaw(self,CopterID=1,Xyz=[0,0,0],RPYaw=[0,0,0]):
```
- 功能: 请求指定 CopterSim 更换 xyz 为指定值, 单位 m, 北东地, roll、pitch、yaw 为指定值, 单位度。
- Self: 表明 ReqCopterSim 共享对象会影响该函数的行为;
- CopterID: 指定 CopterSim ID 号。
- Xyz=[0,0,0]: 期望 xyz。
- RPYaw=[0,0,0]: 期望滚转、俯仰、偏航。

该软件的相关例程有:

[1.BasicExps\el_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. RflySim3D/RflySimUE5

Unreal Engine 具有强大的图形引擎，支持高品质的 3D 图形和视觉效果；内置的蓝图可视化脚本系统，使得开发者可以使用图形化的方式来创建复杂的逻辑和交互行为，而无需编写代码；拥有庞大的社区支持和资源库，包括模型、纹理、音效、插件等等，这些资源可以帮助开发者加快开发进程和提高模型品质；支持多个平台，包括 PC、主机、移动设备和虚拟现实设备等等；开发者可以根据自己的需求来自定义和扩展引擎的功能和工具，使得 Unreal Engine 适用于各种类型的游戏和应用程序开发。

RflySim3D/RflySimUE5 是基于 Unreal Engine 引擎开发的无人系统高逼真仿真软件，继承了 Unreal Engine 引擎的各种优势，通过 UDP 的形式与工具链其他软件进行通信，实现高逼真的无人系统仿真，同时，可通过屏幕抓取、共享内存等方式将视觉图像数据传输到 QGroundControl、MATLAB、Python 等软件，实现无人系统的视觉算法验证仿真，如图 3 所示。

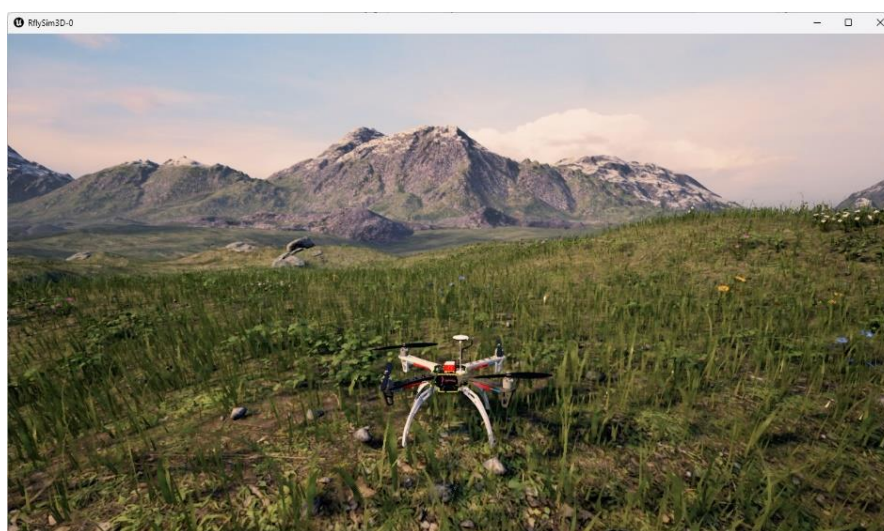


图 3 RflySim3D/RflySimUE5 显示主界面

同时，针对电脑配置较低的用户，RflySim 工具链提供另两种三维仿真软件，分别为：FlightGear 和 3DDisplay。FlightGear 的开发团队来自世界各地，包括程序员、飞行员、物理学家和飞机制造商等领域的专家，提供了多种不同类型的飞机模型和场景，包括各种民用和军用飞机模型，以及多种不同的场景和环境模拟。它是一款非常受欢迎的开源飞行模拟器软件，可以通过 UDP 接收 Simulink 发送的飞行状态，方便地观测 Simulink 仿真时飞机的飞行状态。3DDisplay 是由北航可靠飞行控制研究组开发的虚拟飞行模拟器软件，提供了三维模型和虚拟环境，支持多种飞机模型和场景。用户可根据个人电脑的配置情况，自由切换 RflySim3D/RflySimUE5、FlightGear、3DDisplay 三款仿真软件。

1.2.1. 快捷键

- **F1**: 弹出帮助菜单提示;
- **ESC**: 清除所有飞机
- **S**: 显示/隐藏飞机 ID;
- **H**: 隐藏/显示所有屏幕文字;
- **D**: 显示/隐藏当前飞机数据;
- **M**: 切换地图(先关闭所有 CopterSim);
- **M+数字***: 切换到第*号地图;
- **B**: 在不同飞机间切换视角焦点;
- **B+数字***: 切换到第*号飞机
- **C**: 切换当前飞机三维样式;
- **C+数字***: 切换到第*号三维样式;
- **CTRL + C**: 切换全部飞机三维样式
- **P**: 开启物理碰撞引擎(会与场景物体和地面发生碰撞, 本功能仅支持完整版)
- **V**: 飞机上的视角切换, **0**: 跟随视角、**1**: 前视摄像头、**2**: 右视摄像头、...;
- **V+数字***: 切换到第*号视角
- **N**: 切换到飞机上帝视角, **0**: 跟随飞机视角(不随飞机姿态改变视角角度)、**1**: 固定地面视角且始终看向当前飞机、**2**: 固定地面向北看视角、**3**: 固定地面向南、...;
- **N+数字***: 切换到第*号上帝视角
- 鼠标左键按下拖动: 切换视角角度; 鼠标右键按下拖动: 切换视角所在纵向 **yz** 位置
- 鼠标滚轮: 切换视角所在横向 **x** 位置
- **CTRL+鼠标滚轮**: 缩放所有飞机尺寸(多机时便于观察);
- **ALT+鼠标滚轮**: 缩放当前视角飞机尺寸
- **T**: 开启或关闭飞机轨迹记录功能
- **T+数字***: 开启/更改轨迹粗细为*号
- 鼠标双击: 显示击中点的位置、尺寸、物体等信息。注: 双击后立即按下 **N** 键, 可以快速将视角切换到双击位置, 便于物体创建
- **O+数字***: 在鼠标双击处生成样式 ID 为“*”的物体(障碍物)
- **I**: 开启局域网联机, 支持显示局域网中其他飞机。(限完整版以上)

RflySim3D (64-bit Development PCD3D_SM5)



1.2.2. 快捷命令

按下键盘左上角 ESC 下方的“`”（一撇）按键，输入下列命令可实现场景配置：

- RflyShowTextTime string time //显示字符串 string，持续 time 秒钟
- RflyShowText string //显示字符串 string，持续 5 秒钟
- RflyChangeMapbyID ID //切换到第 ID 号地图
- RflyChangeMapbyName name //切换到名字为 name 的地图
- RflyChangeViewKeyCmd key num // 模拟键盘快捷键 Key+num（例如，B 1）
- RflyCameraPosAngAdd x y z roll pitch yaw // 增量改变视角位置姿态，单位米和度
- RflyCameraPosAng x y z roll pitch yaw //设定当前视角位置姿态，单位米和度
- RflyCameraFovDegrees degrees //改变当前视角的视场角，单位度
- RflyChange3DModel CopterID veTypes //改变 CopterID 号飞机的样式到 veTypes（可以是序号或者名字，例如 RflyChange3DModel 100 Eric_Walking）
- RflyChangeVehicleSize CopterID size //改变 CopterID 号飞机的尺寸到 size
- RflySetPosScale scale //改变飞机的运动轨迹的尺度，在超大地图是可用。
- RflyMoveVehiclePosAng CopterID isFitGround x y z roll pitch yaw // 增量移动 CopterID 号飞机的位置和姿态（单位米和度），如果 isFitGround 设为 1 则飞机会始终贴地。
- RflySetVehiclePosAng CopterID isFitGround x y z roll pitch yaw // 设置 CopterID 号飞机的位置和姿态（单位米和度），如果 isFitGround 设为 1 则飞机会始终贴地。
- RflyScanTerrainH xL yL xR yR H Interval // 扫描地形数据得到 png 和 txt 的高程地形文件，用于导入 CopterSim；地图左下角坐标为 xL yL，右上角坐标为 xR yR，扫描高度为 H，扫描间隔为 Interval，单位都是米。
- RflySetActuatorPWMS CopterID pwm1 ... pwm8 //设置 CopterID 号飞机的前 8 位执行器
- RflySetActuatorPWMSExt CopterID pwm9 ... pwm24 //设置 CopterID 号飞机的 9 到 24 位执行器数值。

注：上述指令也可通过 UDP 发送给 RflySim3D，见 Python 接口 mav.sendUE4Cmd()

注：除了上述 RflySim 工具链指令，还能输入 UE 自带指令，例如 r.setres 等

1.2.3. 地形校准文件—***.txt 获取

地形校准文件是记录三维场景的尺寸大小及 GPS 数据。该文件的各参数的含义如下：

<场景右上角点 x(cm)>，<场景右上角点 y(cm)>，<场景右上角点 z(cm)>，<场景左下角点 x(cm)>，<场景左下角点 y(cm)>，<场景左下角点 z(cm)>，<场景中任意点 x(cm)>，<场景中任意点 y(cm)>，<场景中任意点 z(cm)>，<GPS 经度(度)>，<GPS 纬度(度)>，<GPS 高度(m)>

其中，三维场景中的三点 xy 全为正，z 向上为正，单位均为厘米，前两个点的目的是为了确认地形的范围和中心坐标，第 3 点坐标可自行选取，理论上需要尽量在高度上与前两个点有落差，用于校正高度尺度。最后三位为 GPS 经纬度及高度数据，该数据会发送给 QGC 和 RflySim3D 实现地图统一，GPS 数据仅适用于于 Cesium 的全球大场景的构建。

获取方式：

```
//可在 RflySim3D/RflySimUE5 的快捷命令栏中输入：  
RflyScanTerrainH xL yL xR yR H Interval  
//扫描地形数据得到 png 和 txt 的高程地形文件，用于导入 CopterSim；地图左下角坐标为 xL yL，右上角坐标为 xR yR，扫描高度为 H，扫描间隔为 Interval，单位都是米。
```

使用方式：与地形高程文件***.png 复制到：*PX4PSP\CopterSim\external\model 下，打开 CopterSim 即可使用。

1.2.4. 地形高程文件—***.png 获取

地形高程文件实际上是以图片形式存储的二维矩阵，包含了场景的高程图。以 png 格式存储矩阵能够很好的实现高程矩阵的压缩，便于节省空间。该文件并不包含坐标原点、

缩放尺度、场景范围等信息，因此需要一个校正文件，RflySim 工具链采用 txt 格式输入 9 维数组传入校正信息。

获取方式：

```
//可在 RflySim3D/RflySimUE5 的快捷命令栏中输入：  
RflyScanTerrainH xL yL xR yR H Interval  
//扫描地形数据得到 png 和 txt 的高程地形文件，用于导入 CopterSim；地图左下角坐标为 xL yL，右上角坐标为 xR yR，扫描高度为 H，扫描间隔为 Interval，单位都是米。
```

使用方式：与地形校准文件***.txt 复制到：*PX4PSP\CopterSim\external\model 下，打开 CopterSim 即可使用。

该软件的相关例程有：

[1.BasicExps\3_RflySim3D-Shortcut-Instruct\Readme.pdf](#)

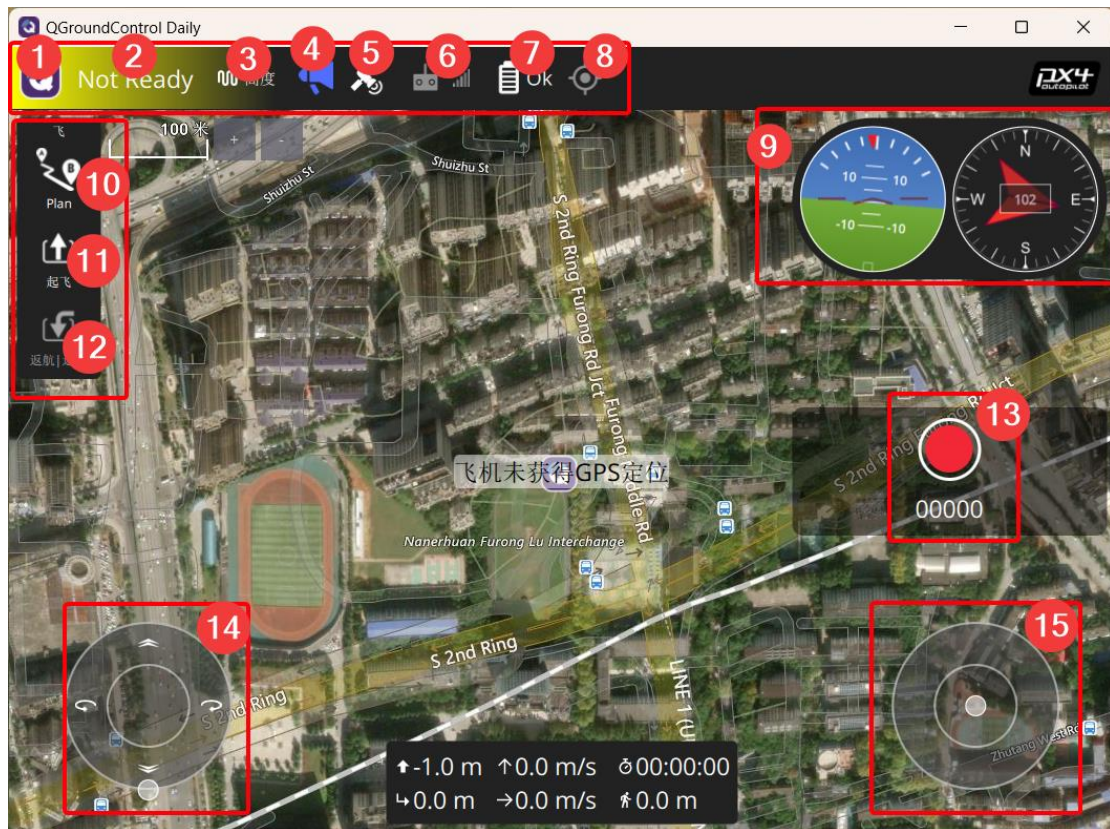
[1.BasicExps\15_Scene-Load\Readme.pdf](#)

1.3. QGroundControl 地面站

无人机地面站是无人机应用控制系统的关键组成部分，操作人员可以通过鼠标、触摸屏、遥控手柄操作地面站以达到控制无人机的目的，并且通过在地面站上设定航点信息以及规划航线，可以使无人机按照预设的路径飞行，并在飞行途中完成航点任务，包括拍照，飞机动作，录像等。目前主流开源地面站为 QGroundControl 和 MissionPlanner，而 QGroundControl 是专为 PX 4 软件最新架构的开源地面站，其使用 QT 编辑器 C++ 语言编写其核心代码，其支持源代码修改和功能二次开发，即适合无人机地面站研究实验也适合无人机地面站功能的定制及修改。相比来说 QGroundControl 的优势有：1) 开源性：QGroundControl 是一个完全开源的软件，这意味着用户可以根据需要自由修改和定制它。2) 易于性：用户界面非常清晰、现代化和易于使用，使用户可以快速进行任务规划和飞行计划。3) 多平台支持：QGroundControl 可在多种操作系统上运行，如 Windows、Linux 和 MacOS 等。4) 模块化架构：QGroundControl 的模块化架构使得开发人员可以轻松地添加和扩展新功能，而不会影响到现有的功能和性能。总体而言，QGroundControl 是一个现代化、易于使用、开源且高度可定制的地面站软件，它在多平台支持、多语言支持、模块化架构等方面具有明显的优势。

1.3.1. QGC 初始界面介绍

QGroundControl 的整体界面如下图所示，界面中各按钮的相关解释如下所示：

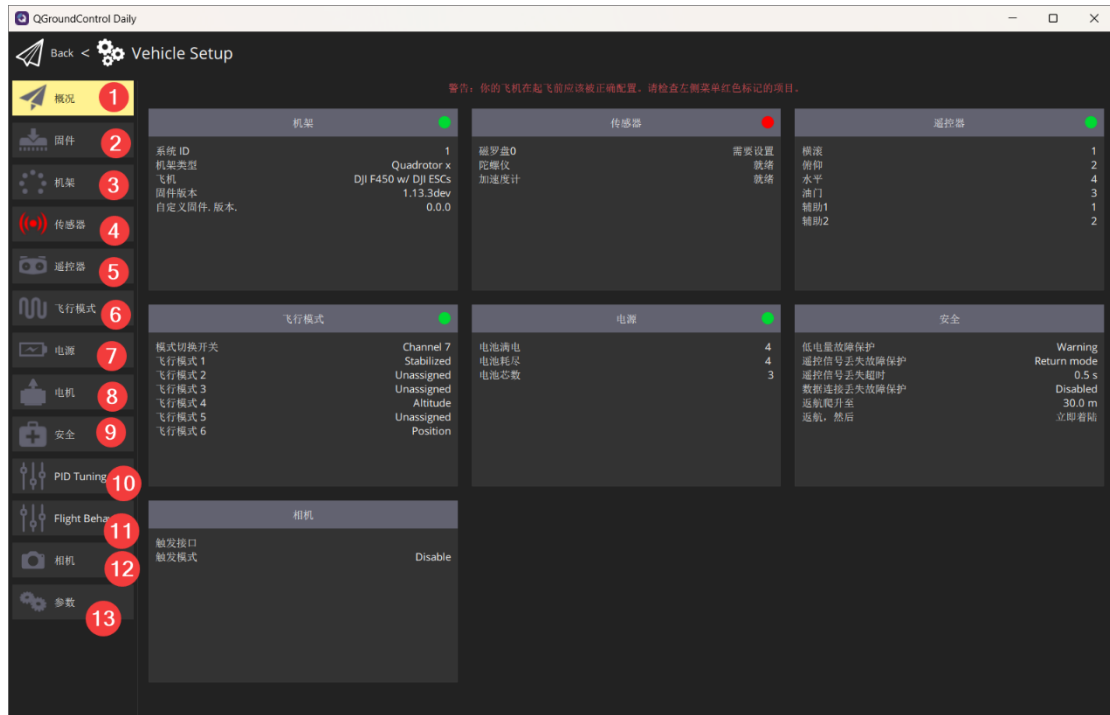


- ① 开始按钮：该按钮可以弹出快捷菜单，可进入载具初始化设置、分析工具使用以及相关的软件属性设置。
- ② 载具状态显示：一般从此处可快速查看载具的整体状态。
- ③ 控制模式选择：该按钮可以切换不同的控制模式，如：手动、自稳、特技等等控制模式。
- ④ 通知：此处可查看载具运行时的信息，如：警告信息、错误信息等。
- ⑤ GPS 状态：显示当前载具所能搜到的卫星数量。
- ⑥ 手柄链接状态显示。
- ⑦ 电池电量显示。
- ⑧ ROI 区域识别。
- ⑨ IMU 状态实时仪表盘。
- ⑩ 航线规划。
- ⑪ 起飞按钮。
- ⑫ 返航按钮。
- ⑬ 录制按钮：可录制 QGC 界面视频。
- ⑭ 虚拟手柄 CH3/CH4 通道。
- ⑮ 虚拟手柄 CH1/CH2 通道。

1.3.2. 载具设置 (Vehicle Setup)

QGC 地面站初始界面可以看到的相关按键命令如下

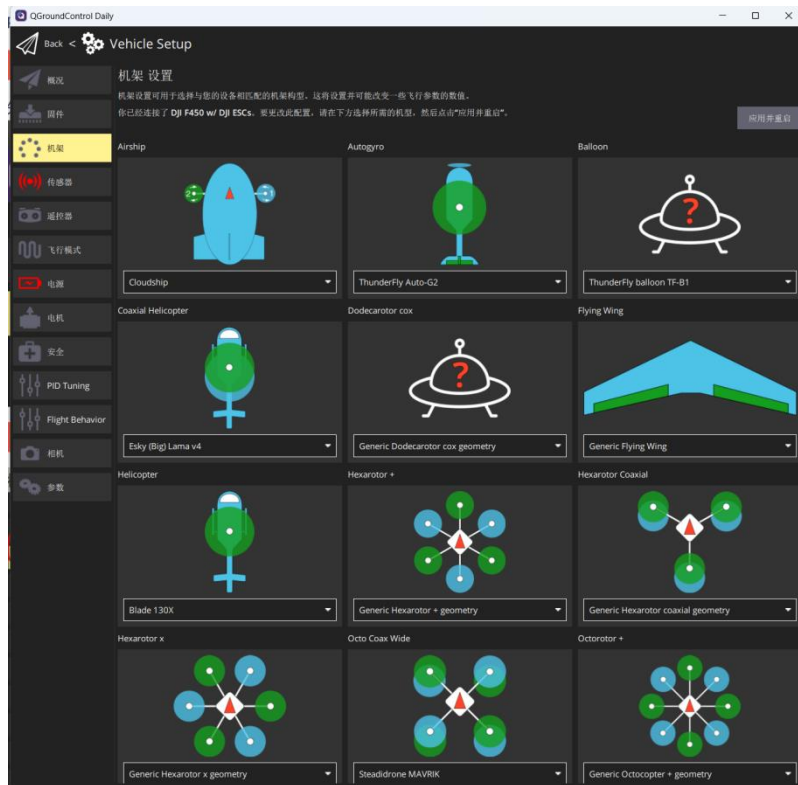
https://docs.qgroundcontrol.com/master/en/qgc-user-guide/getting_started/quick_start.html



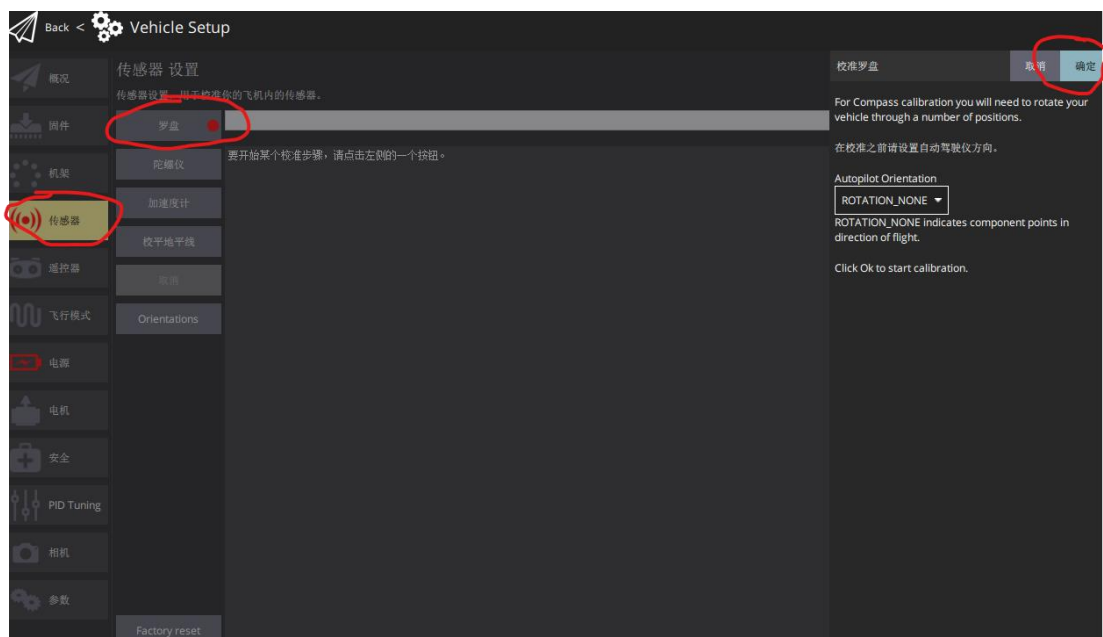
- ① 载具概况：显示当前所连接的载具整体状态情况，如：机架、传感器、遥控器、飞行模式等等。
- ② 固件：先不连接飞控，点击如下页面，然后用 USB 将飞连接电脑，注意飞控不要用电池或其他 USB 以外的设备供电。



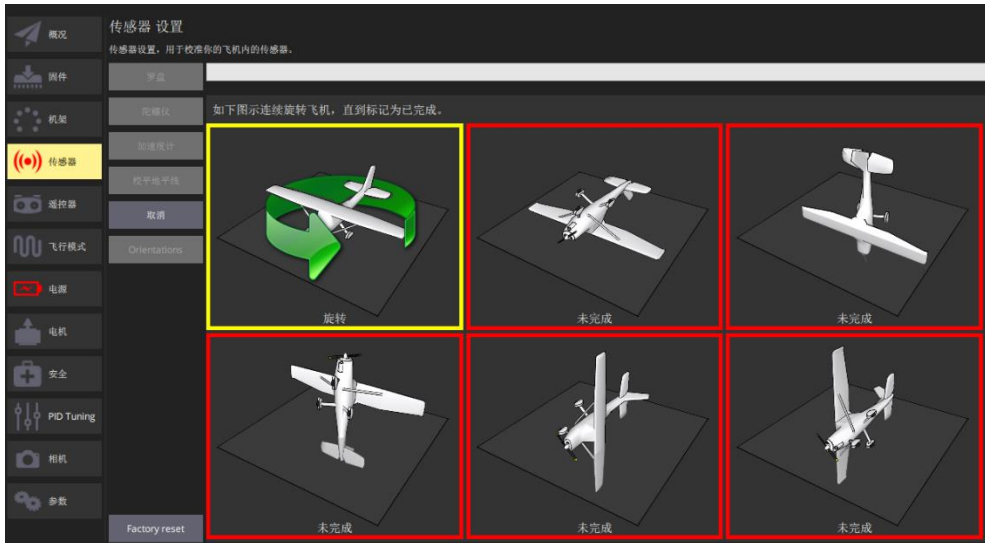
- ③ 机架：将飞控连接至地面站，将机架设置为自己想设置的机型，设置完后请右上方“应用并重启”才可生效。



- ④ 传感器：该传感器中主要包含 IMU 中涉及到的传感器，校准时，一般先校准罗盘，步骤如下：

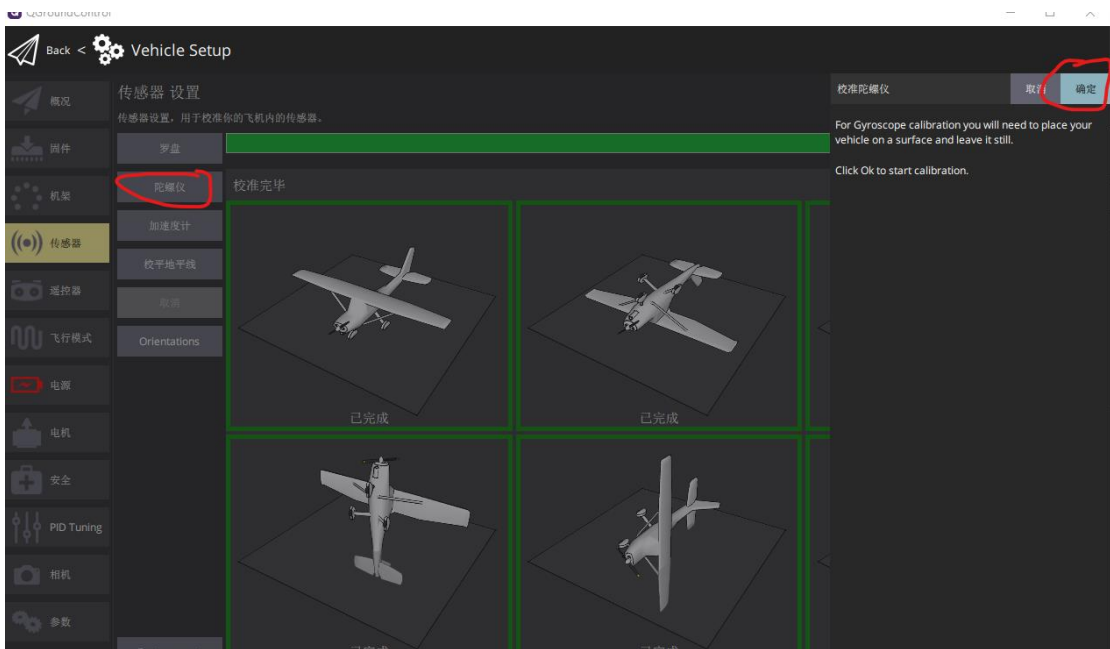


将无人机置于红色所示的任何方向，并保持静止。出现提示后（方向图像变为黄色），沿任意/两个方向绕指定轴旋转车辆。当前方向校准完成后，屏幕上的相关图像将变为绿色。



对所有方向重复校准过程。在所有方向校准完毕后，QGroundControl 将显示 Calibration complete（校准完成）（所有方向图像将显示为绿色，进度条将完全填满）。然后可以继续下一个传感器。

校准陀螺仪：单击陀螺仪传感器按钮，将无人机水平放在地面上，保持静止。单击“确定”开始校准。顶部的条形图满代表校准成功。

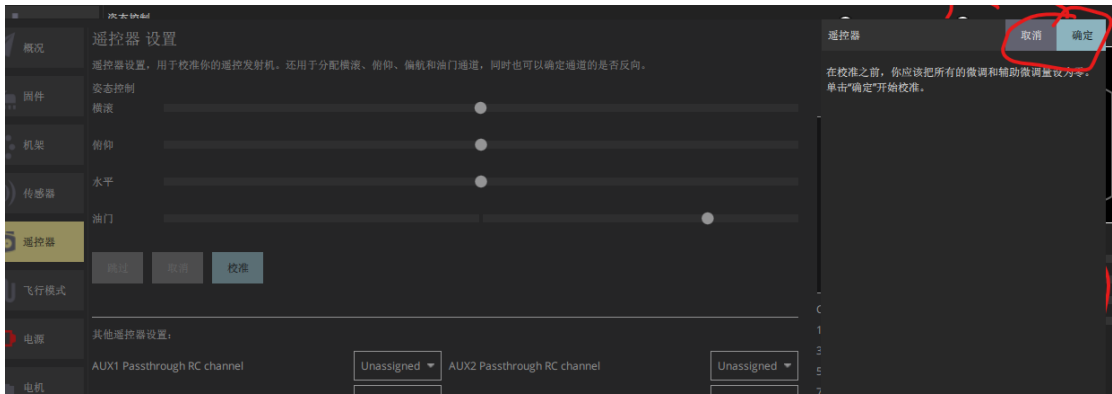


⑤ 遥控器

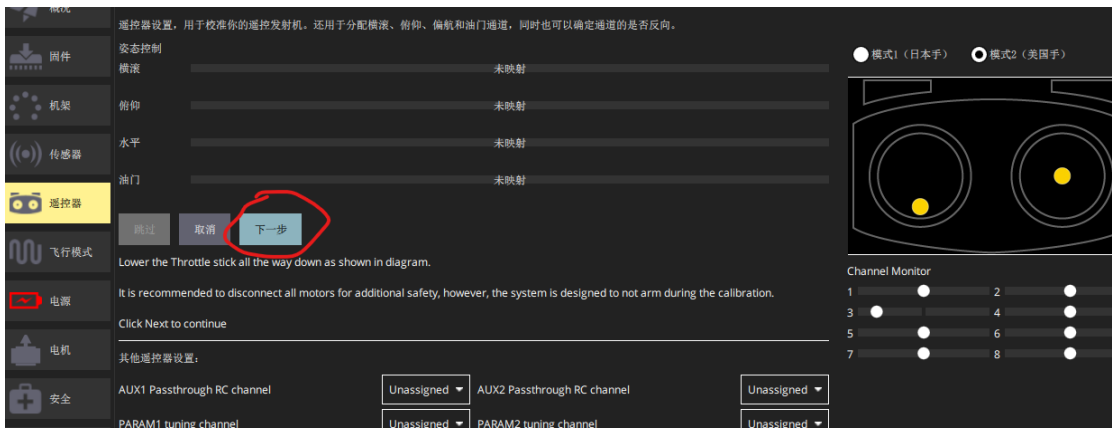
打开遥控器，切换到遥控器页面，检查右下角是否能识别到通道，如果能识别到通道，就可以进行校准，选择右上角的操作方式，然后点击校准



然后点击“确定”



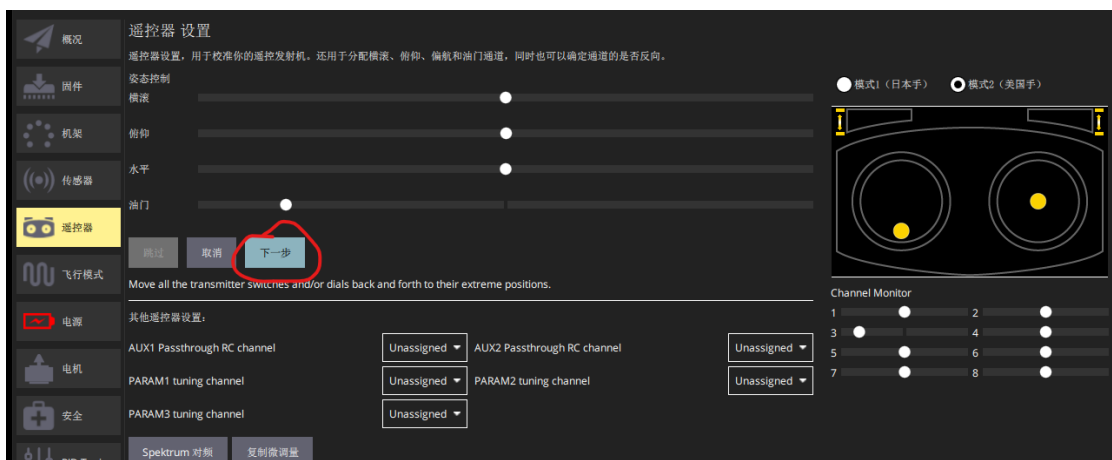
再点击“下一步”



将遥控器摇杆移动到下图指示的位置。

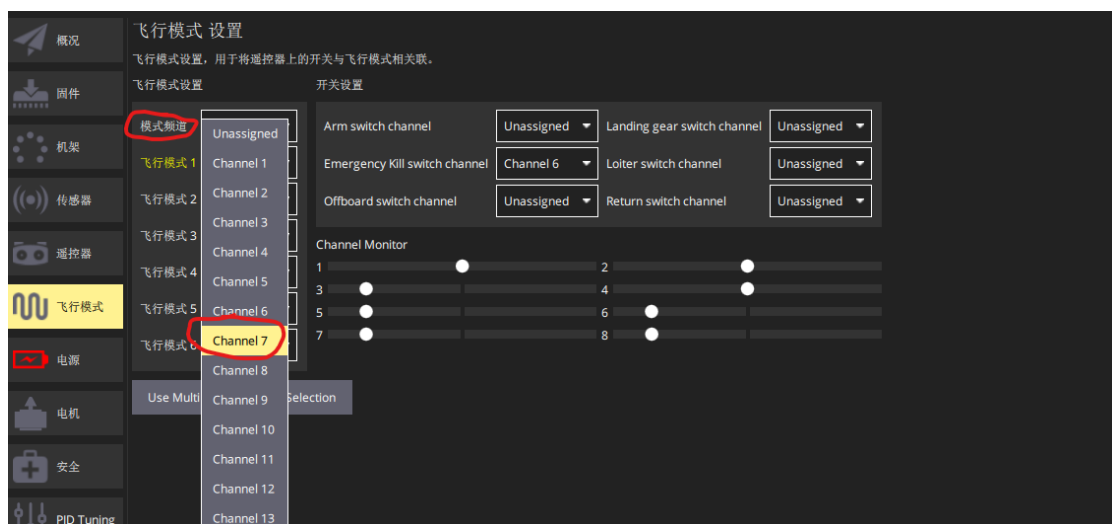


当杆就位时，地面站会提示下一个需要拨的位置，拨完所有位置后，按两次“下一步”保存设置。

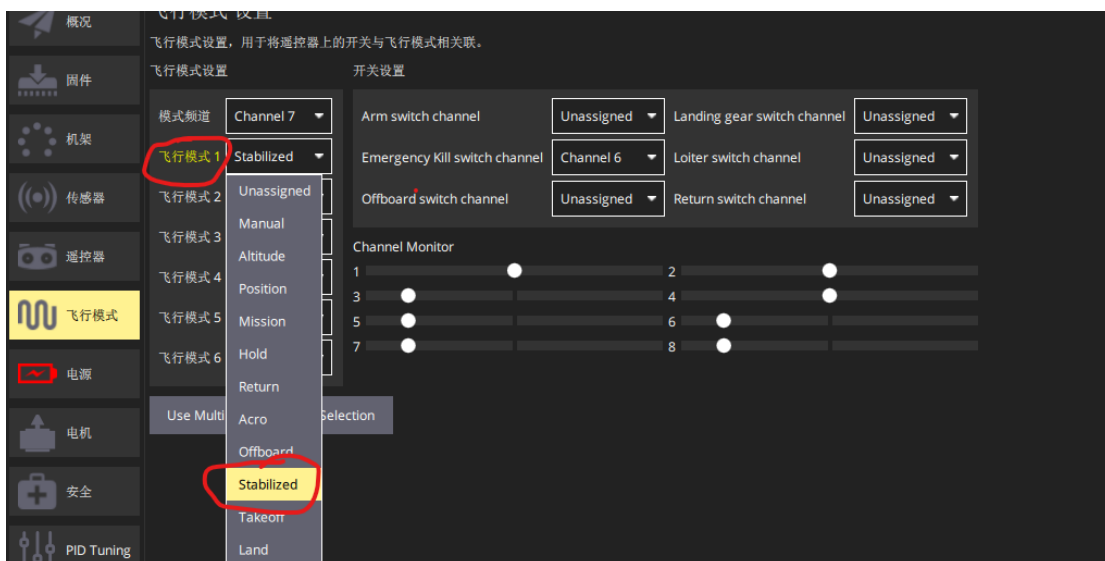


⑥ 飞行模式切换开关

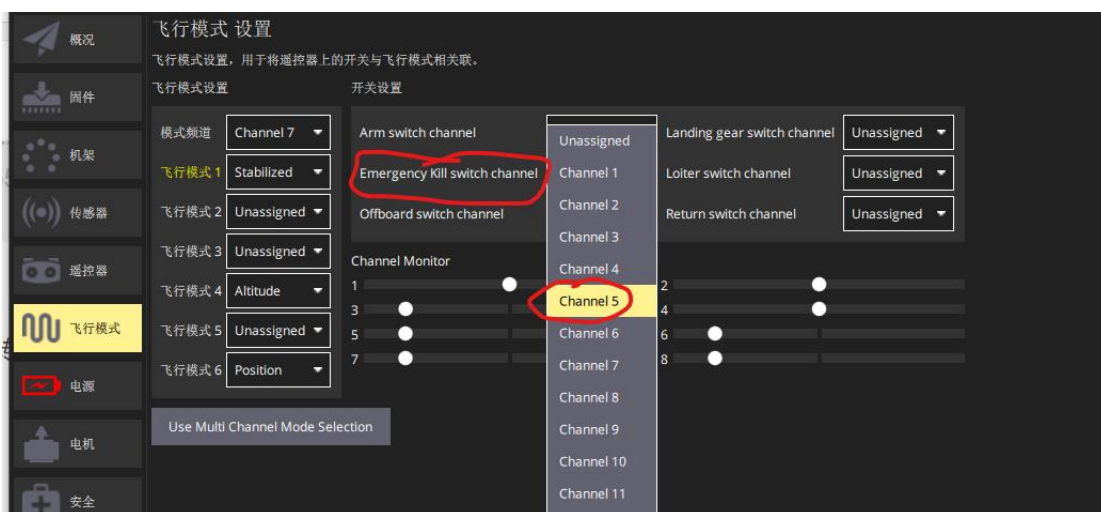
点击“模式频道”右侧的复选框，设置相应的遥控器拨码开关通道。



然后分别设置三档对应的飞行模式。

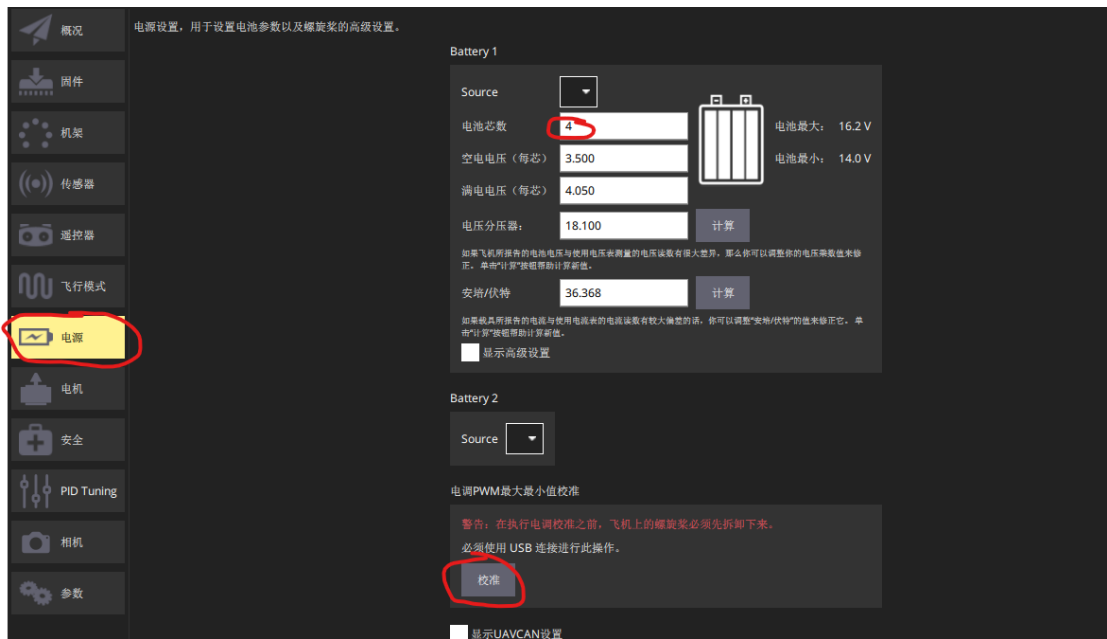


其他的开关通道在飞行模式右侧，如下，需要设置哪个，就把这个开关右侧的遥控器通道进行设置即可，我这里设置了一个刹车（Kill switch），通道为遥控器的第五个通道。刹车的作用是使电机直接停转，可根据需要进行设置



⑦ 电源

校准电调时，用 USB 将飞控连接到地面站，不接电池，不装浆叶，电调的信号线接到飞控上。切换到“电源”页面，输入电池芯数并回车，点击“校准”，然后插上电池即可校准。

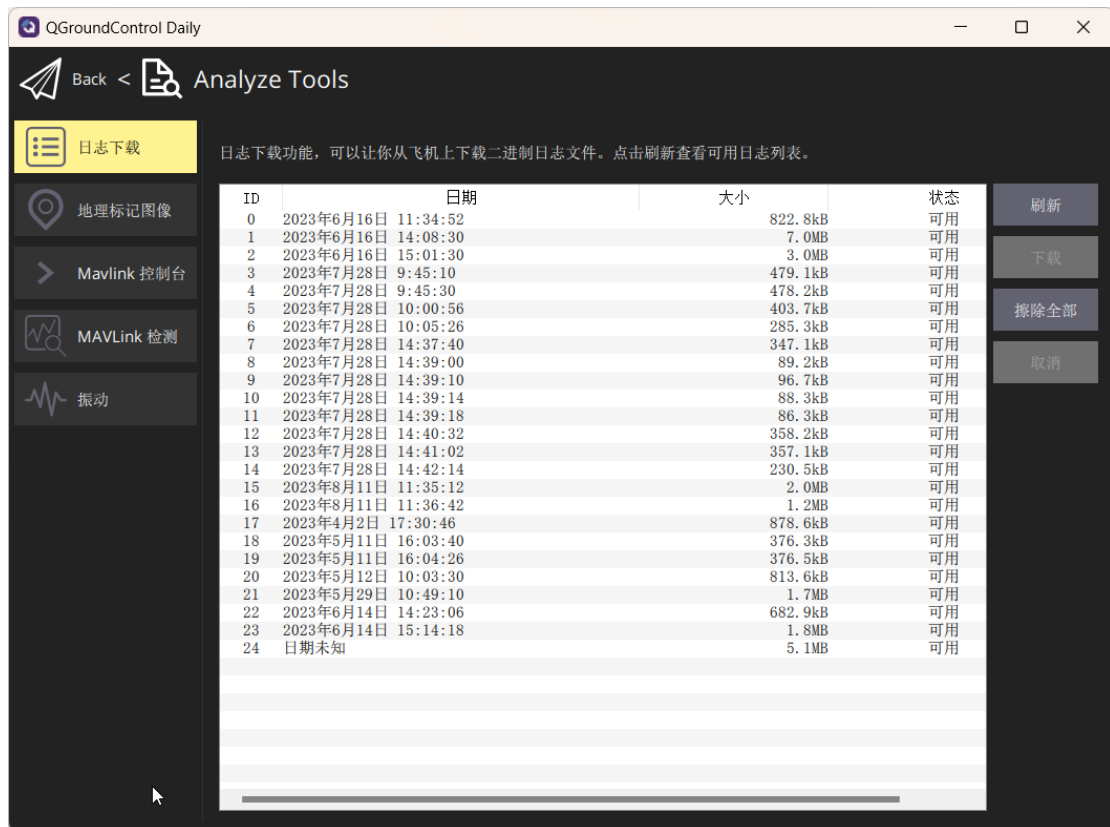


- ⑧ 电机：显示电机的 PWM
- ⑨ 安全：该菜单下可对载具的低电量故障保护触发器、物体探测、遥控器信号丢失故障进行相关设置。
- ⑩ PID 整定：可对载具的 PID 控制参数进行整定。
- ⑪ Flight Behavior。
- ⑫ 相机设置
- ⑬ 参数：此处可修改 PX4 软件中定义的任意参数，修改完成后需重启才可生效。

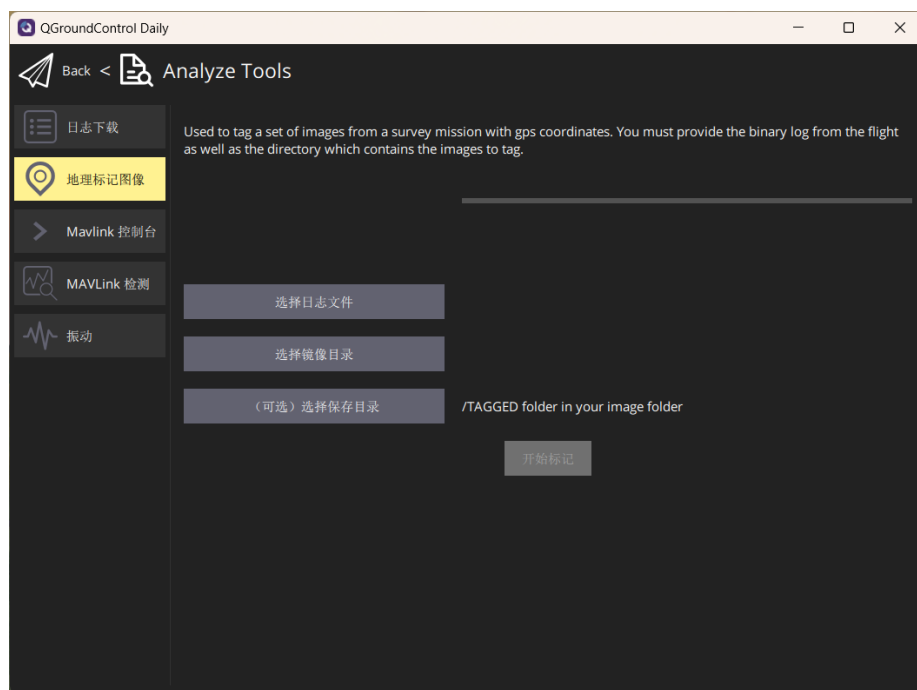
1.3.3. 数据分析 (Analyze Tools)

QGC 提供丰富的数据分析工具，主要有日志下载、地理标记图像、MAVlink 控制台、MAVLink 检测、振动。

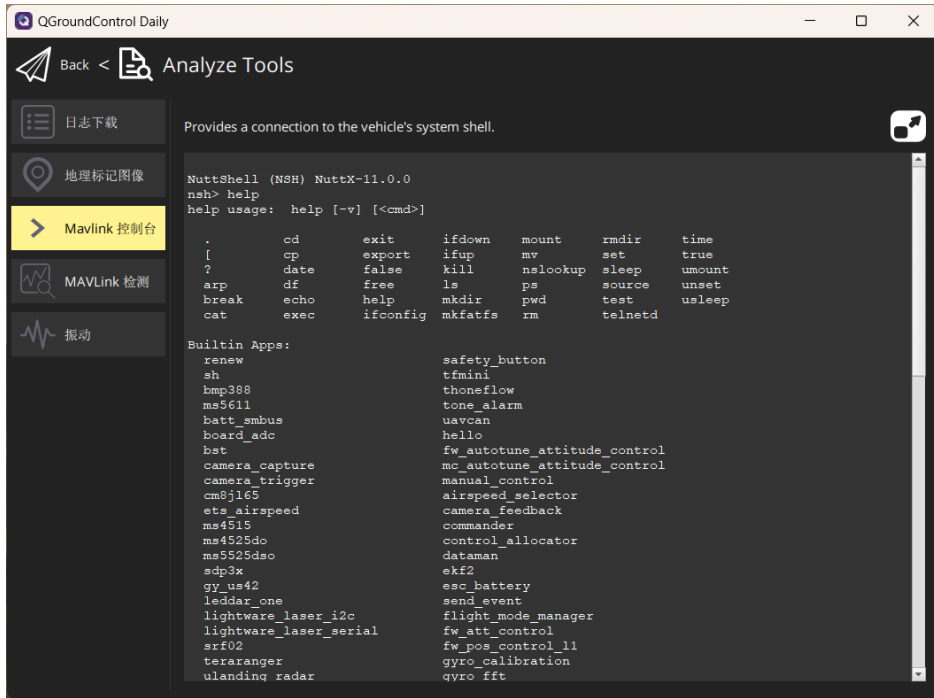
日志下载：可以在链接飞控的情况下，再在当前飞控内存卡内存储的日志信息，选择任意一条即可下载到 .ulg 格式的文件，该文件可通过网站：https://docs.px4.io/main/zh/log/flight_log_analysis.html 进行分析日志。



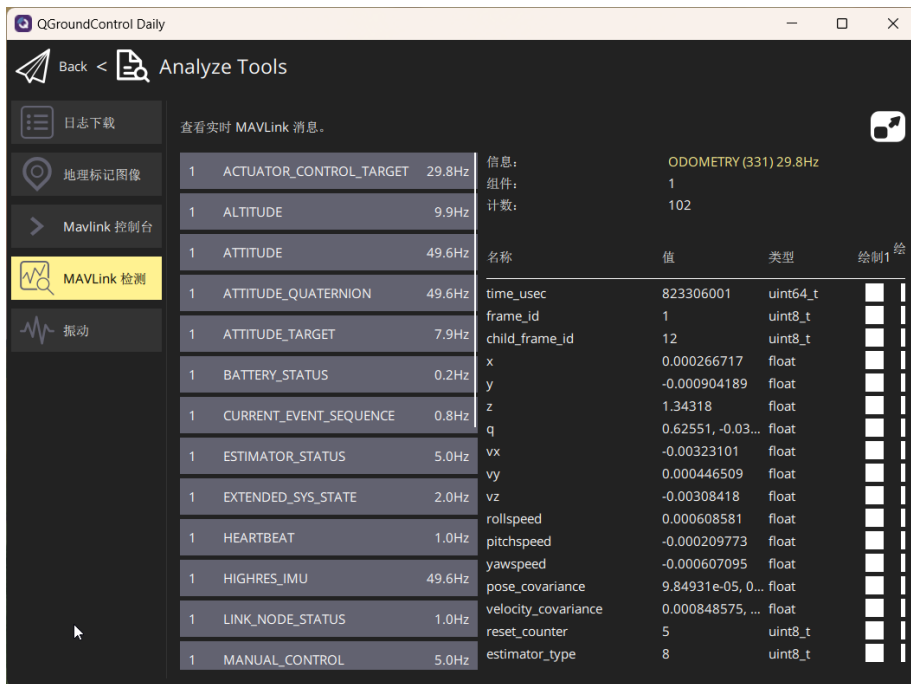
地理标记图像：用于用 `gps` 标记一组调查任务的图像，但必须提供航点的二进制日志以及包含要标记的图像的目录。



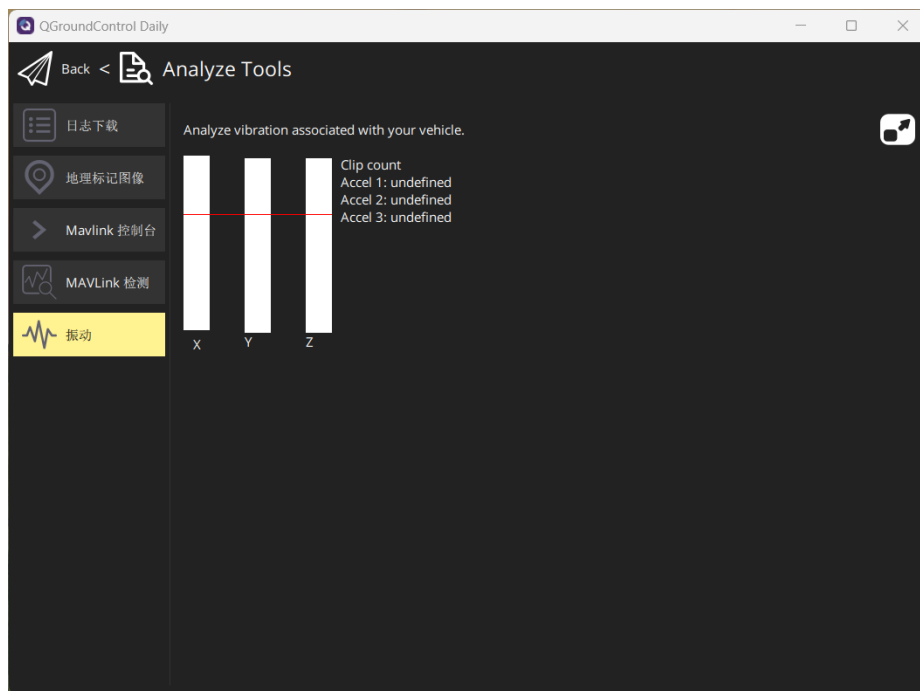
MAVLink 控制台：它提供一个可与载具上的飞控运行系统 Nuttx 的 Shell 进行数据通信链接。



MAVLink 检测:



振动: 分析与车辆相关的振动



该软件的相关例程有：

[1.BasicExps\10_Firmware-Upload\Readme.pdf](#)

[1.BasicExps\16_Identify-Hardware-Command\Readme.pdf](#)

[1.BasicExps\17_RoutePlanning\Readme.pdf](#)

1.4. Python38Env

Python 是一种高级的、面向对象的解释型编程语言。它最初由 Guido van Rossum 于 1989 年创建，现在已经成为一种流行的编程语言，用于开发 Web 应用程序、数据分析、人工智能、科学计算、网络编程等等。Python 是一种简单易学、易读、易编写的语言，因此也被广泛用于教学和入门级编程。

Python38Env 是一个 Python 3.8 的编程语言的虚拟环境，包含了 numpy、pymavlink、OpenCV、pyulog 等库，可快速进行无人系统的相关算法开发，无需用户部署 python 运行环境和各种功能库。

该软件的相关例程有：

[1.BasicExps\14_Log-Reads-Python38Env\Readme.pdf](#)

1.5. MATLAB 自动代码生成工具箱

MATLAB 自动代码生成工具箱是一个 MATLAB 的扩展工具包，用于从 Simulink 模型中生成 C 代码、可执行文件、静态库和动态库等各种形式的可执行文件。这些可执行文件可以直接在嵌入式平台上运行，而不需要进行手动编写和调试。支持多种嵌入式平台，包括 ARM Cortex-M 和 A 系列处理器、NXP MPC55xx 和 MPC56xx 系列、Pixhawk 系列等。

该模块库中包含 GPS 数据模块、电池数据模块、uORB 模块等众多模块。基于 RflySim 和 Pixhawk Support Package 平台，用户可实现：① 在 Simulink 中设计和模拟控制算法；② 从 Simulink 模型自动生成 C 代码和 PX4 固件，并将其直接烧录到 Pixhawk 板上；③ 使用 MATLAB 脚本和函数配置和校准 Pixhawk 板及其外围设备；④ 实时读写与 Pixhawk 板的数据等等。

该软件的相关例程有：

[1.BasicExps\7 Code-Generation\Readme.pdf](#)

1.6. SITL/HITL 批处理脚本

批处理技术是指计算机可分组处理收集到的若干任务，整个过程完全自动化，无需人工干预，这也可以称为工作负载自动化 (WLA) 和作业调度。它具有速度和成本节约、准确性、操作简单等优点。

RflySim 基于批处理技术开发了众多批处理脚本，让用户可以快速一键启动部署多架、多种、多样无人系统组合式仿真。提高了无人系统开发和仿真速度。平台较常用的批处理脚本：① SITLRun.bat：是开启多机软件在环仿真的批处理文件，其本质上是脚本方式启动并配置 RflySim 工具链的部分软件和选项，② HITLRun.bat：是开启多机硬件在环仿真的批处理文件，插入多个飞控之后，再双击该批处理文件，根据提示输入想要参与仿真的 Pixhawk 串口号，即可开启多机的硬件在环仿真（以输入串口顺序进行飞机 ID 排序）。除此之外 RflySim 工具链还提供众多批处理脚本文件，如：SITLRunPos.bat、SITLRunLowGPU.bat、SITLRunMAVLink.bat、HITLRunPos.bat、HITLPosSysID.bat、HITLPosStr.bat 等等，部分脚本文件见 *PX4PSP\RflySimAPIs\BatScripts。用户可通过编辑器打开这些文件，根据个人需求修改其中的参数，实现自定义开发，快速开始仿真或算法的验证。

1.6.1. HITLRun.bat

```
常规硬件在环仿真脚本，支持输入串口序列（英文逗号","分隔），来启动多机的硬件在环仿真
注：REM 打头的行是注释语句，不会被执行。其他的 bat 脚本语法规则可自行搜索学习。
注：本脚本的飞机位置是由脚本按矩形队列自动生成的，控制变量包括：
SET /a START_INDEX=1（初始飞机序号，本脚本生成的飞机的 CopterID，以此 START_INDEX 为初始值，依次递增
1）
SET /a IsSysID=0（根据飞控的 sysID 自动设置）
SET /a TOTOAL_COPTER=8（总飞机数量，仅在多机联机仿真才需要幅值，告诉本脚本实际的飞机总数，以此来确定
矩形队列的边长）
SET /a CLASS_3D_ID=-1（设置载具的 ClassID 显示在 RflySim3D 中）
SET /a BaudRate=921600（设置飞控的波特率）
SET UE4_MAP=Grasslands（设置地图名字）
SET /a ORIGIN_POS_X=0（矩形编队的原点 X 位置，单位米，只支持整数输入）
SET /a ORIGIN_POS_Y=0（矩形编队的原点 Y 位置，单位米，只支持整数输入）
SET /a ORIGIN_YAW=0（矩形编队的原点 yaw 角度，单位度，只支持整数输入）
SET /a VEHICLE_INTERVAL=2（矩形编队的飞机间隔，单位米，只支持整数输入）
SET /a UDP_START_PORT=20100（接收外部控制数据的 UDP 通信接口，与 CopterID 对应会自动加 2，这里通常不
需要修改，仅在电脑端口被占用时才修改）
SET /A DLLModelVal=DLLModel（是否使用 DLL 模型，DLL 模型的名字。这里支持将 Simulink 的飞行器模型生成
```

DLL 来导入工具链，此模式支持固定翼、无人车等模型)

SimMode (仿真模式，这里设置为 **0** 或 **PX4_HITL**，表示硬件在环仿真)

SET IS_BROADCAST=0 (是否联机仿真，这里可输入目标 IP 地址序列)

SET UDPSIMMODE=0 (**UDP_START_PORT** 端口接收的数据协议，UDP 模式传输的是工具链私有结构体，支持 **Simulink** 控制；**MAVLink** 模式传输的是 **MAVLink** 协议，支持 **Python** 和 **mavros** 等控制模式)

1.6.2. SITLRun.bat

常规软件在环脚本，支持输入飞机数量，自动开启多机软件在环仿真

与 **HITLRun.bat** 相比，关键代码如下

set SimMode=2 (这里设置为软件在环模式，对应 **CopterSimUI** 的值)

set PX4SITLFrame=iris (这里对应了 **PX4** 飞控的机架模式，这里是一个四旋翼)

1.6.3. HITLPos.bat

开启硬件在环仿真，支持输入 **PosX**、**PosY** 和 **Yaw** 的值来初始化飞机位置偏航角

注：输入的位置字符串，以英文逗号“,”分隔

关键代码如下：

SET /P PosXStr=Please enter the PosX (m) list: (输入 X 的初始位置序列，单位米，可以为浮点数)

SET /P PosYStr=Please enter the PosY (m) list: (输入 Y 的初始位置序列，单位米，可以为浮点数)

SET /P YawStr=Please enter the Yaw (degree) list: (输入偏航的初始值序列，单位度，可以为浮点数)

1.6.4. SITLPos.bat

开启软件在环仿真，支持输入 **PosX**、**PosY** 和 **Yaw** 的值来初始化飞机位置偏航角

注：其余注意事项同上

1.6.5. HITLPosStr.bat

开启硬件在环仿真，**PosX**、**PosY** 和 **Yaw** 的值以字符串的形式写在 **bat** 文件的 **PosXStr**、**PosYStr**、**YawStr** 变量中，不需要手动输入

注：使用时，需要手动去修改 **PosXStr**、**PosYStr**、**YawStr** 字符串的值，来设置生成飞机的位置和偏航

注：输入的位置字符串，以英文逗号“,”分隔

注：位置序列的数量可以大于插入飞控数量，会依次取前面的位置进行赋值，例如本例子位置字符串包含 **10** 组飞机初始姿态，因此支持仿真 **1** 到 **10** 个飞机的硬件在环仿真

关键代码如下：

SET PosXStr=1,2,3,4,5,6,7,8,9,10 (X 的初始位置序列，单位米，可以为浮点数)

SET PosYStr=0,0,0,0,0,0,0,0,0,0 (Y 的初始位置序列，单位米，可以为浮点数)

SET YawStr=0,0,0,0,0,0,0,0,0,0 (偏航的初始值序列，单位度，可以为浮点数)

SITLPosStr.bat

开启软件在环仿真，**PosX**、**PosY** 和 **Yaw** 的值以字符串的形式写在 **bat** 文件的 **PosXStr**、**PosYStr**、**YawStr** 变量中，不需要手动输入

注：其余注意事项同上

1.6.6. HITLRunSysID.bat

开启硬件在环仿真，但是 **CopterID** 不是自动增 **1** 排序，而是根据 **SysID** 的值 (在 **QGroundControl** 中可配置) 来自自动确定

注：如果使用常规模式，**CopterID** 是和电脑串口序列先后顺序绑定的，每次插拔飞控后串口号会变化，因此无法通过 **CopterID** 定位到确定的飞控硬件

注：使用本方法，**CopterID** 可以直接和飞控硬件绑定，便于故障时快速确定飞控序号。

注：如果要再使用常规模式的 **HITLRun.bat** 脚本，建议将 **SysID** 改回默认值 **1**

关键代码如下：

SET /a IsSysID=1 (开启通过 **SysID** 自动计算 **CopterID** 的功能)

1.6.7. HITLPosSysIDStr.bat

开启硬件在环仿真，通过 **SysID** 自动确定 **CopterID** 取值，且支持配置初始位置序列

注：本脚本会以 **SysID - START_INDEX** 的值，去索引 **PosXStr**、**PosYStr**、**YawStr** 列表来确定最终位置

注：例如 **SysID** 为 **15**，**START_INDEX** 为 **11**，**PosXStr=1,2,3,4,5,6,7,8,9,10**，则飞机最终的 x 取值应该是 **PosXStr** 的第 **4** 位 (从 **0** 开始数)，也就是 **PosX=5**

关键代码如下：

```
SET /a IsSysID=1 (开启通过 SysID 自动计算 CopterID 的功能)
SET /a START_INDEX=1 (位置列表的初始飞机的值)
SET PosXStr=1,2,3,4,5,6,7,8,9,10 (X 的初始位置序列, 单位米, 可以为浮点数)
SET PosYStr=0,0,0,0,0,0,0,0,0,0 (Y 的初始位置序列, 单位米, 可以为浮点数)
SET YawStr=0,0,0,0,0,0,0,0,0,0 (偏航的初始值序列, 单位度, 可以为浮点数)
```

1.6.8. HITLPosSysID.bat

开启硬件在环仿真, 通过 SysID 自动确定 CopterID 取值 (同 HITLPosSysIDStr.bat), 但初始位置序列是手动输入的

1.6.9. HITLRunLowGPU.bat

开启常规硬件在环仿真, 使用 LowGPU 场景, 来保证低配电脑能运行工具链

注: 核心修改是 HITLRun.bat 将 GrassLands 字符替换为 LowGPU

关键代码如下:

```
SET UE4_MAP=LowGPU
```

1.6.10. SITLRunLowGPU.bat

开启常规软件在环仿真, 不使用 GrassLands 场景但是使用 LowGPU 场景, 来保证低配电脑能运行工具链

注: 其余注意事项同上

1.6.11. HITLRunMAVLink.bat

开启常规硬件在环仿真, 使用 MAVLink_Full 的 UDPSIMMODE 取值, 支持 Python 视觉控制和真机飞行控制

注: 这里仅修改了 UDPSIMMODE 的值

注: UDPSIMMODE 可以取数字或字符串, 对应了 CopterSim 的 UI 界面

关键代码如下:

```
SET UDPSIMMODE=Mavlink_Full (使用 Mavlink_Full 的通信模式于 20100 系列端口)
```

1.6.12. SITLRunMAVLink.bat

开启常规软件在环仿真, 使用 MAVLink_Full 的 UDPSIMMODE 取值, 支持 Python 视觉控制和真机飞行控制

注: 其余注意事项同上

1.6.13. HITLRunNoUI.bat

开启硬件在环仿真, 使用无 UI 版 CopterSimNoUI.exe, 可以减小计算量, 适合大规模集群使用

注: 核心修改是将 CopterSim.exe 的字符串替换为 CopterSimNoUI.exe

注: 工具链的集群例程, 超过数量 4 时, 会自动使用 NoUI 版 CopterSim 以提高飞行稳定性。

1.6.14. SITLRunNoUI.bat

开启软件在环仿真, 使用无 UI 版 CopterSimNoUI.exe, 可以减小计算量, 适合大规模集群使用

注: 其余注意事项同上

1.6.15. HITLPosAlt.bat

开启硬件在环仿真, 增加了 Alt 标识符, 支持配置飞机的初始高度、俯仰角和滚转角。

注: 其余注意事项同上

1.6.16. SITLPosAlt.bat

开启软件在环仿真, 增加了 Alt 标识符, 支持配置飞机的初始高度、俯仰角和滚转角。

注: 其余注意事项同上

1.6.17. HITLPosAltStr.bat

开启硬件在环仿真, 增加了 Alt 标识符, 支持配置飞机的初始高度、俯仰角和滚转角, 支持启动多架飞机。

注: 其余注意事项同上

1.6.18. SITLPosAltStr.bat

开启软件在环仿真, 增加了 Alt 标识符, 支持配置飞机的初始高度、俯仰角和滚转角, 支持启动多架飞机。

注: 其余注意事项同上

1.6.19. HITLRunChange3D.bat

开启常规硬件在环仿真功能，使用 100003 的外观，即 ClassID=3 的四旋翼，+第 1 个飞机样式，即 Droneeye X330 样式

关键代码如下：

```
SET /a CLASS_3D_ID=100003
```

1.6.20. SITLRunChange3D.bat

开启常规软件在环仿真功能，使用 100003 的外观，即 ClassID=3 的四旋翼，+第 1 个飞机样式，即 Droneeye X330 样式

关键代码如下：

```
SET /a CLASS_3D_ID=100003
```

1.6.21. HITLPosStrGPS.bat

开启硬件在环仿真，使用全球 GPS 坐标初始化飞机位置

关键代码如下：

SET isPosGps=1（如果 isPosGps 设为 1，则 PosXStr 和 PosYStr 可使用纬度和精度格式（单位度）输入；若无 isPosGps 或设为 0，则使用 xy 的北东地（单位米）的格式输入初始位置。）

SET isBatLLAOrin=1（如果 isBatLLAOrin 设置为 1，且 LatLongAlt 设定了经纬高的值，则会使用 LatLongAlt 的经纬高作为北东地坐标系下，GPS 坐标原点。反之，使用模型内部 GPS 设定值，或者 png+txt 地形文件中指定的值。）

注：本功能仅支持完整版及以上，免费版无法配置 GPS 初始位置。

1.6.22. SITLPosStrGPS.bat

开启软件在环仿真，使用全球 GPS 坐标初始化飞机位置

关键代码如下：

SET isPosGps=1（如果 isPosGps 设为 1，则 PosXStr 和 PosYStr 可使用纬度和精度格式（单位度）输入；若无 isPosGps 或设为 0，则使用 xy 的北东地（单位米）的格式输入初始位置。）

SET isBatLLAOrin=1（如果 isBatLLAOrin 设置为 1，且 LatLongAlt 设定了经纬高的值，则会使用 LatLongAlt 的经纬高作为北东地坐标系下，GPS 坐标原点。反之，使用模型内部 GPS 设定值，或者 png+txt 地形文件中指定的值。）

注：本功能仅支持完整版及以上，免费版无法配置 GPS 初始位置。

该软件的相关例程有：

[1.BasicExps\06_BAT-Startup\Readme.pdf](#)

1.7. PX4 Firmware 源码

PX4 由瑞士苏黎世联邦理工学院(ETH)计算机视觉与几何实验室的一个软硬件项目 PIX HAWK 演变而来，该飞控系统完全开源，为全球各地的飞控爱好者和研究团队提供一款低成本高性能的高端自驾仪。经过来自工业界和学术界的世界级开发人员多年的开发与完善，目前 PX4 飞控系统已经形成完善合理的软件架构，配合 Pixhawk 系列自驾仪硬件平台上，构成了 Pixhawk PX4 自驾仪软硬件平台，可控制多旋翼、固定翼、飞艇等多种载具，是目前世界范围内广泛应用的开源无人机自驾仪软硬件平台。

RflySim 工具链支持一键部署 PX4 编译环境，可自定义选择不同的 PX4 固件编译命令和固件版本，工具链会将选定的 PX4 Firmware 源代码部署在设定的安装路径上，如果固件存在，会删除旧的固件文件夹，并进行全新部署，极大的提高了 PX4 环境部署的效率。

该软件的相关例程有：

[1.BasicExps\012_PX4-App\Readme.pdf](#)

1.8. WinWSL 子系统

WinWSL 子系统是一种 Windows 操作系统上的子系统，用户可在 Windows 系统中运行 Linux 应用程序，使用 Linux 命令行界面（CLI）并安装 Linux 发行版，RflySim 工具链一键安装的 Linux 系统为 Ubuntu20.04，主要是用于 PX4 源代码的编译，

本工具链还提供另两套编译环境来实现 Windows 平台下对 Linux 编译环境的模拟，分别是：基于 Msys2 的 Msys2Toolchain 编译环境和基于 Cygwin 的 CygwinToolchain 编译器。用户可根据自己的 PX4 版本选择不同的编译环境，且在一键部署安装界面不同的选择即可完成不同编译环境的切换。

该软件的相关例程有：

[1.BasicExps\le9_Build-Firmware\Readme.pdf](#)

1.9. Simulink 集群控制接口

RflySim 工具链基于 Simulink S 函数开发了集群控制接口，如图 4 所示，该接口是通过 Simulink S 函数通过 C++混编实现，配合 Simulink 自带的 UDP 模块优点，具有效率高、运算小、延迟低、更可靠、扩展性强等优点。用户可通过复制粘贴的方式，将该模块加载到自己的控制系统中，帮助用户快速实现无人系统集群控制开发。

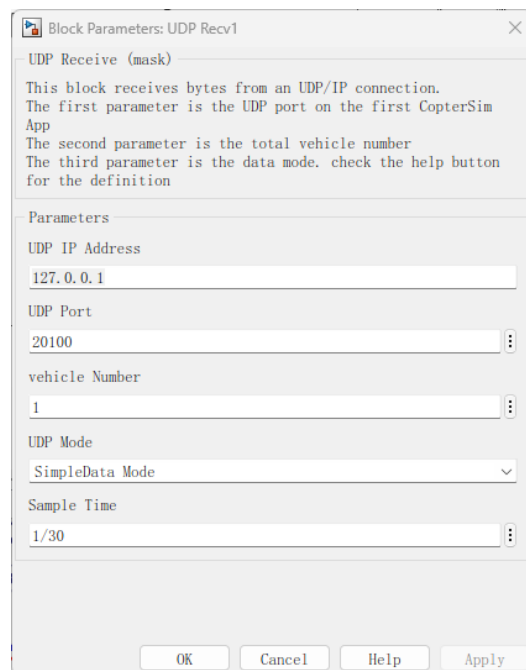


图 4 集群控制接口界面

- 第一项“UDP IP Address”是目标电脑的 IP 地址，输入“127.0.0.1”则只能接受本机的 CopterSim 转发的 Pixhawk 自驾仪状态并进行控制；“255.255.255.255”则能接收并控制局域网内所有电脑中运行的 CopterSim 程序（其他电脑的 CopterSim 需要勾选“联

机”按钮)；“192.168.1.12”之类的指定 IP 则只会向该 IP 地址的主机发送控制指令。

一般而言，在小范围集群的时候“255.255.255.255”广播已经能够满足需求，当飞机数量继续增多，则需要启用指定 IP 来减小网络负载，提高通信速度和可靠性。

- 第二项“UDP Port”是第一个飞机的初始端口号，默认起始端口是 20100。每个 CopterSim 的收发消息需要各占用一个端口。例如，本模块需要仿真飞机 ID 为 10~15 的飞机，这一项则需要填 $20100+10*2=20120$ ，后面一项“Vehicle number”飞机数量需要填 5。
- 第三项“Vehicle number”飞机数量表示需要连接的 CopterSim 数量，同时控制该模块的输入输出端口数量，如果输入 10，则模块会自动生成 10 对输入输出接口。
- 第四项“UDP mode”是输入输出接口的数据模式协议，主要包含 FullData 完整模式（数据最全，但传输数据量较大）；SimpleData 精简数据模式（较多飞机 >8 ，避免数据过大网络阻塞）和 UltraSimple 超精简模式（单电脑飞机数 >20 ），延迟更小。
- 第五项是“Sample Time”采样时间，该时间应该与 Simulink 仿真时间对应。

该软件的相关例程有：

[1.BasicExps\8_SwarmAPI\Readme.pdf](#)

1.10. RflySim 配套资料文件

RflySim 工具链提供非常完善的学习资料及例程文件，通过 PPT 课件资料和 RflySimAPIs 例程文件，使用户循序通过渐进、层层递进的学习方式，从无人系统底层控制算法→中层决策算法→顶层学习算法的开发与验证，一站式搭建、开发出自己所需的无人系统。

2. RflySim 工具链配套硬件系统

RflySim 工具链提供了一套完整的配套硬件系统，包括四旋翼无人机、飞控、遥控器等组件。这些组件均能与工具链完美兼容，可以在 RflySim 工具链中是实现软、硬件在环仿真实验，基于生成的固件实现无人机在真实环境下的飞行。

2.1. 飞思系列飞机

目前支持的飞机有飞思 X150、飞思 X200、飞思 X450 等四旋翼无人机，其中飞思 X150 为室内集群控制科研全新设计的微型四旋翼无人机，

2.1.1. 飞思 X150 四旋翼无人机

为室内集群控制科研全新设计的微型四旋翼无人机，对称电机轴距 140mm，创新式全防护结构设计，摒弃以往碳板冗杂布线，碳尼 3D 打印高强度、轻质量机体，采用激光定高与光流定点，整机一体化方案，全面提升室内集群科研效率。



科研方向：光学定位系统导航定位开发；集中式/分布式集群编队算法开发车机组合天地一体协同编队控制开发；ROS 二次开发；matlab 二次开发；

版本与性能

产品配置	标准版	旗舰版
基础配置	光流定点、激光定高、外置磁罗盘	
机载板卡	ZYpi-3566	
板卡性能	CPU: RK3566 内存: 4GB, DDR4 存储: 32GB WIFI: 集成 wifi6	
视觉传感器	无	单目传感器*2, 200 万像素
定位系统	室内光学定位系统	室内光学定位系统 /GPS
通讯方式	WIFI	
基础软件环境	各传感器驱动	
功能特点	专注于实现集中式和分布式集群编队功能	在实现集中式和分布式集群编队功能基础上，并可以进行一般性视觉功能开发应用；可基于 GPS 定位进行飞行

飞行器指标

飞思 X150 智能无人机	
尺寸 (含桨)	200*200*85mm
对称电机轴距	140mm
飞行器重量	205g
电池	3s, 1300mAh 105g
整机重量 (含电池)	310g
最大上升速度	2m/s
最大下降速度	2m/s
最大水平飞行速度	5m/s

最大起飞海拔高度	3500m
续航时间（空载）	8 分钟
工作环境温度	-20° C 至 50° C

应用场景：完善的室内微型无人机集群协同编队科研解决方案，适合高等院校教学科研，以及军工单位科研，主要应用于室内无人机集群控制、分布式集群算法验证领域。

2.1.2. 飞思 X200 四旋翼无人机

室内小型智能无人机，对称电机轴距 200mm，全碳纤维防护机体设计，螺旋桨下沉式安装方式，采用内部激光定高与光流定点，适用于室内无人机集群协同编队应用，具有分布式集群无人机协同控制能力。可搭载可见光摄像头与机载视觉处理板卡，具有进行视觉导航、目标识别、目标跟随能力。



科研方向

- 基于模型设计开发；
- ROS 控制开发；
- matlab 控制开发；
- 集中式/分布式集群控制算法开发；
- 进行视觉导航、目标识别、目标跟随算法验证；

产品版本

产品配置	标准版	单目版	模型设计版
基础配置	光流定点、激光定高、外置磁罗盘		
飞控	Racer 飞控		
机载板卡	NX Xavier		NX Xavier/ZYpi3566
视觉传感器	T265	单目相机	无
通讯方式	WIFI		
基础软件环境	各传感器驱动		
功能特点	采用 T265 进行定位，进行高精度室内集中式/分布式集群控制算法开发	集中式/分布式集群控制算法开发；目标识别、目标跟随算法验证	基于模型设计开发；ROS 控制开发；

飞行器指标

飞思 X200 智能无人机	
尺寸 (含桨)	300*300*160mm
对称电机轴距	200mm
飞行器重量	580g
电池	4s, 5300mAh, 469g
整机重量 (含电池)	1049g
额外最大载重	200g
最大上升速度	2m/s
最大下降速度	2m/s
最大水平飞行速度	10m/s
最大起飞海拔高度	4000m
续航时间 (空载)	20 分钟
工作环境温度	-20° C 至 50° C

应用场景

室内小型无人机集群协同编队科研解决方案，作为专业面向高校及科研院所的智能飞行器产品，主要应用以下科研领域:基于模型设计开发；室内集中式/分布式集群算法开发；视觉导航；目标跟随；目标识别。

2.1.3. 飞思 X450 四旋翼无人机

专业的室外小型智能四旋翼无人机，对称电机轴距 450mm，整机模块化设计，在搭载机载计算机的同时，配备搭载深度相机和激光雷达等功能模块，形成完善的室外智能飞行器，优异的产品性能可以应对复杂的室外飞行环境，是面向室外集群编队算法开发、slam 导航等研究领域的智能飞行器科研平台。



科研方向

- 基于模型设计开发；
- ROS 二次开发；
- matlab 二次开发；
- 无人机集中式/分布式集群控制；
- 视觉 slam 导航、激光 slam 导航开发；

产品版本

产品配置	领航版	旗舰版	高阶版
基础配置	光流定点、激光定高、外置磁罗盘		
飞控	Racer 飞控		
机载板卡	NX Xavier		
视觉里程计	T265 相机		
空间探测	D435i 深度相机	思岚 S1 激光雷达	D435i 深度相机 思岚 S1 激光雷达
定位系统	GPS/RTK		
通讯链路	200m 范围内—板载 WiFi; 3km—ZY-H3; 10km—ZY-H12		
基础软件环境	各传感器驱动 无人机 offboard 控制示例程序		
功能特点	可进行长达二十多分钟续航的室外集群编队飞行控制; 单机实现视觉 slam 导航算法验证与开发	室外集群编队飞行控制; 单机实现激光 slam 导航算法验证与开发	在集群的功能基础上, 单机同时实现视觉 slam 导航、激光 slam 导航算法验证与开发功能

飞行器指标

飞思 X450 智能无人机	
尺寸 (不含桨)	420*420*240mm
对称电机轴距	450mm
飞行器重量	1200g
电池	6s, 6000mAh, 862g
整机重量 (含电池)	2062g
额外最大载重	1000g
定位精度	GPS: 垂直: $\pm 0.5m$; 水平: $\pm 2m$ RTK: 垂直: $\pm 3cm$; 水平: $\pm 5cm$
最大上升速度	2m/s
最大下降速度	2m/s
最大水平飞行速度	8m/s
最大起飞海拔高度	4000m
续航时间 (空载)	30min
工作环境温度	-20° C 至 50° C

应用场景

完善的室外小型无人机集群协同编队科研解决方案, 适合高等院校教学科研, 以及军工单位科研, 应用于 Slam 算法开发/验证; 路径规划/避障算法开发; AI 算法开发/验证等领域。

2.1.4. 飞思 X680 四旋翼无人机

中型智能四旋翼无人机对称电机轴距 680mm, 整机采用工业化设计, 高强度机身可作

为于多任务载荷飞行平台，采用激光定高光流定点，并配备深度相机和激光雷达等功能模块，在具备视觉导航开发与目标跟随等开发条件的同时，可以进行较大载荷任务飞行，是一款兼顾载重、长时续航和科研开发的多功能智能无人机。



科研方向：

- 基于模型设计开发；
- 无人机集中式/分布式集群控制；
- 室外机载集群控制算法开发；
- ROS 控制开发、支持 matlab 控制开发；
- 与无人车结合进行天地一体协同编队控制；
- 进行视觉导航、目标识别、目标跟随算法验证；

产品版本：

产品配置	标准版	定制版
基础配置	光流定点、激光定高、外置磁罗盘	
飞控	H7 飞控	
机载板卡	NX Xavier	
空间探测	D435i	激光雷达
吊舱	无	G1 云台吊舱
其他功能模块	无	定制搭载
定位系统	GPS/RTK	
通讯链路	3km—ZY-H3；10km—ZY-H12	
基础软件环境	各传感器驱动	
功能特点	载重较大，长时续航；室外可进行多任务载荷集群编队飞行；搭载 NX 板卡，同时进行多种复杂算法验证，单机实现目标识别、视觉导航等人工智能应用开发。	可根据具体的应用需求进行定制化搭载传感器或功能模块设计，满足图像识别、目标跟随等多种功能需求，推荐选择搭载 G1 云台吊舱、激光雷达、RTK 高精度定位模块、定制版云台吊舱等。

飞行器指标

飞思 X680 智能无人机	
尺寸（不含桨）	567*567*400mm
对称电机轴距	680mm
飞行器重量	2550g
电池	6s, 16000mAh, 1475g
整机重量（含电池）	4025g
额外最大载重	2000g
定位精度	GPS: 垂直: $\pm 0.5\text{m}$; 水平: $\pm 2\text{m}$ RTK: 垂直: $\pm 3\text{cm}$; 水平: $\pm 5\text{cm}$
最大上升速度	2m/s
最大下降速度	2m/s
最大水平飞行速度	12m/s
最大起飞海拔高度	5000m
续航时间（空载）	40 分钟
工作环境温度	-20° C 至 50° C

应用场景

完善的室外中型智能无人机解决方案，适合高等院校教学科研，以及军工单位科研，主要应用于室内无人机集群控制、分布式集群算法验证领域。

2.2. PX4 系列飞控

由于 RflySim 工具链是基于 PX4 软件系统开发而成，正常情况下，只要支持 PX4 软件系统的飞控均可在 RflySim 工具链上使用。目前长期支持的飞控 Pixhawk 2.4.8(又名 Pixhawk 1)、Pixhawk 6C、Pixhawk 6X。

2.2.1. Pixhawk 2.4.8(又名 Pixhawk 1)

Pixhawk 2.4.8 是一种先进的自动驾驶仪，由 PX4 开放硬件项目设计和 3D 机器人制造。它具有来自 ST 公司先进的处理器和传感器技术，以及 NuttX 实时操作系统，能够实现惊人的性能，灵活性和可靠性控制任何自主飞行器。其特点有：

- 1.先进的 32 位 ARM CortexM4 高性能处理器，可运行 NuttX RTOS 实时操作系统。
- 2.14 个 PWM/舵机输出(其中 8 个具有安全和手动控制功能，另外 6 个辅助，兼容高功率)，外设丰富(UART,I2C,SPI,CAN)。
- 3.冗余设计，集成备份电源和基本安全飞行控制器，主控制器失效时可安全切换到备份控制。
- 4.备份系统集成混控功能，提供自动和手动混控模式。
- 5.冗余电源输入和自动故障转移，外部安全按钮以容易启动电机。

- 6.多色 LED 灯, 高功率, 多音蜂鸣器。
- 7. Micro SD, 长时间高速率记录飞行数据。



如果使用的是 Pixhawk 2.4.8 (2M flash) 的飞控硬件(对应固件为 px4_fmuv3), 推荐使用下图所示软件安装配置, 和右下图所示硬件连接配置。



- 使用 px4_fmuv3_default 编译命令。
- 使用“6”: PX4 1.12.3 版本固件。
- 使用“1”: WinWSL 编译器。
- Pixhawk 1 上自带 LED 灯, 不需要外接模块, 只需按右图连接遥控器接收机。

注: Pixhawk 2/3/4 开始都不自带 LED 等模块, 需要购买外接 LED 模块。

2.2.2. Pixhawk 6C

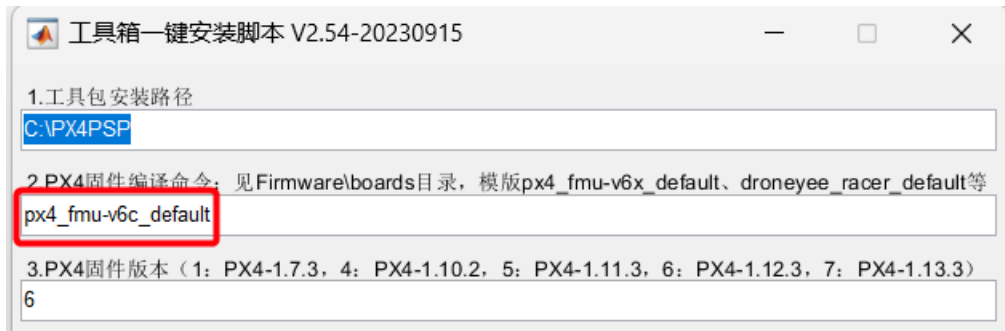
Pixhawk 6C 是基于 Pixhawk FMUv6C 开放标准和连接标准的成功无人机控制器家族的最新更新。它配备了 PX4 自动驾驶仪。在 Pixhawk 6C 内部, 可以找到由 STM 制造的基于 STM32H743 的芯片, 与来自 Bosch 和 InvenSense 的传感器技术配对, 为任何自主车辆的控制提供灵活性和可靠性, 适用于学术和商业应用。其特点有:

1. 高性能 STM32H743 处理器, 具有更多的计算能力和内存容量;
2. 新的成本效益设计, 采用低底盘尺寸;
3. 新设计的集成振动隔离系统, 过滤高频振动并降低噪声, 确保准确的读数;

4. 惯性测量单元（IMUs）由内置的加热电阻进行温度控制，确保 IMUs 的最佳工作温度。



如果使用的是 Pixhawk 6C 的飞控硬件，推荐使用下图所示软件安装配置，硬件连接配置同 Pixhawk 2.4.8 相同。



- 使用 `px4_fmuv6c_default` 编译命令。
- 使用“7”：PX4 1.13.3 版本固件。
- 使用“1”：WinWSL 编译器。

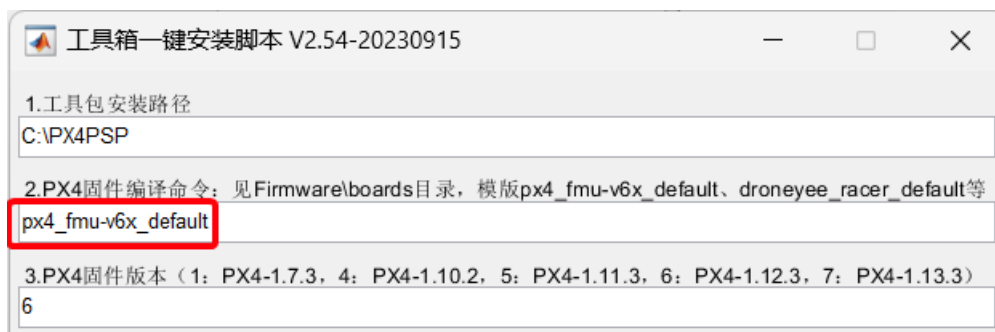
2.2.3. Pixhawk 6X

在 Pixhawk 6X 内部，您可以找到由 STM 制造的基于 STM32H753 的芯片，与 Bosch、InvenSense 提供的传感器技术配对，为任何自主车辆的控制提供灵活性和可靠性，适用于学术和商业应用。Pixhawk 6X 的 H7 微控制器包含运行最高达 480MHz 的 Arm® Cortex®-M7 核心，具有 2MB 闪存存储和 1MB RAM。PX4 自动驾驶仪利用了增强的处理能力和 RAM。由于更新的处理能力，开发人员可以更加高效和生产力，使他们的开发工作变得更加复杂和模型。FMUv6X 开放标准包括内置的高性能、低噪声惯性测量单元（IMU），旨在提高稳定性。独立的 LDO 为每个传感器组供电，具有独立的电源控制。一种过滤高频振动并降低噪声的振动隔离系统，以确保准确的读数，使车辆能够实现更好的总体飞行性能。外部传感器总线（SPI5）具有两个芯片选择线和数据就绪信号，用于与 SPI 接口连接的附加传感器和载荷，并配有内置的微芯片以太网 PHY，可以通过以太网实现高速通信。Pixhawk 6X 完美适用于企业研究实验室、初创企业、学术研究（包括教授、研究生和学生）以及商业应用。其特点有：

1. 高性能 STM32H753 处理器；
2. 可拆卸的飞控板：独立的 IMU、FMU 和基础系统通过 100 个 Pin 和 50 个 Pin Pixhawk 自动驾驶仪总线连接器相连。
3. 冗余：在各自的总线上具有三倍 IMU 传感器和双倍气压传感器。
4. 三倍冗余区域：具有各自的总线和各自的电源控制的完全隔离的传感器区域。
5. 新设计的振动隔离系统过滤高频振动和减少噪声，以确保准确的读数。
6. 以太网接口用于高速任务计算机集成。
7. 惯性测量单元由内置的加热电阻进行温度控制，以确保 IMU 的最佳工作温度。



如果使用的是 Pixhawk 6X 的飞控硬件，推荐使用下图所示软件安装配置，硬件连接配置同 Pixhawk 2.4.8 相同。



- 使用 px4_fmu-v6c_default 编译命令。
- 使用“7”：PX4 1.13.3 版本固件。
- 使用“1”：WinWSL 编译器。

2.3. 常用遥控器配置

本工具链使用的遥控器推荐使用“美国手”的操纵方式，即左侧摇杆对应的油门与偏

航控制量，而右侧摇杆对应滚转与俯仰。遥控器中滚转、俯仰、油门和偏航分别对应了接收机的 CH1~CH4 通道，左右上侧拨杆对应了 CH5/CH6 号通道，用于触发飞行模式切换。

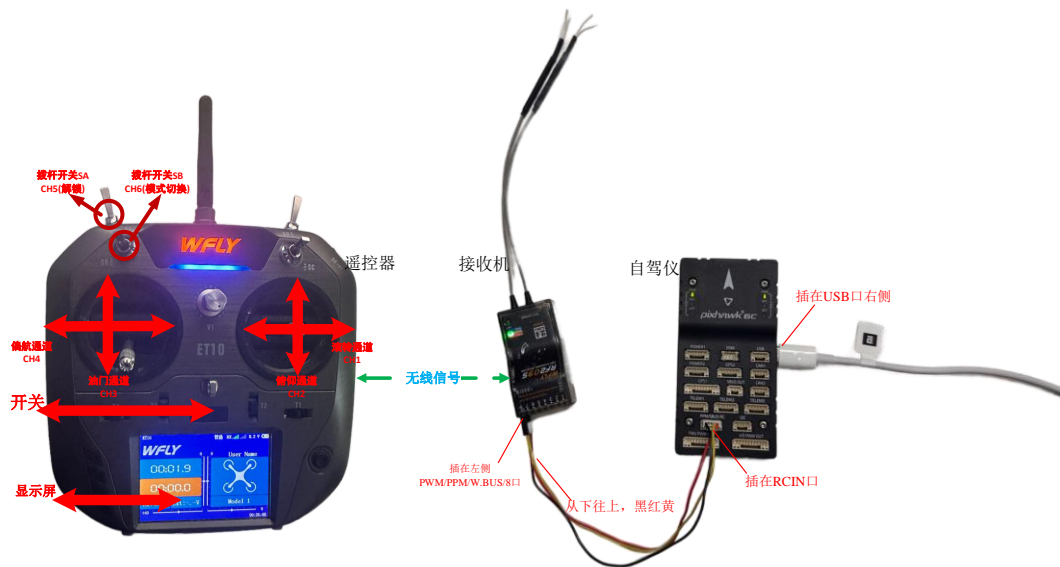
油门杆（CH3 通道）从最下端和最上端分别对应了 PWM 信号从 1100 到 1900 附近波动（通道不同或遥控器不同都会存在区别，因此需要校准）；滚转（CH1 通道）和偏航（CH4 通道）摇杆从最左端到最右端对应 PWM 信号从 1100 到 1900；俯仰（CH2 通道）摇杆从最下端到最上端对应 PWM 信号从 1900 到 1100；CH5/6 为三段开关，从顶部（最远离使用者的档位）到底部（最靠近使用者的档位）档位对应 PWM 信号为 1100、1500 和 1900。



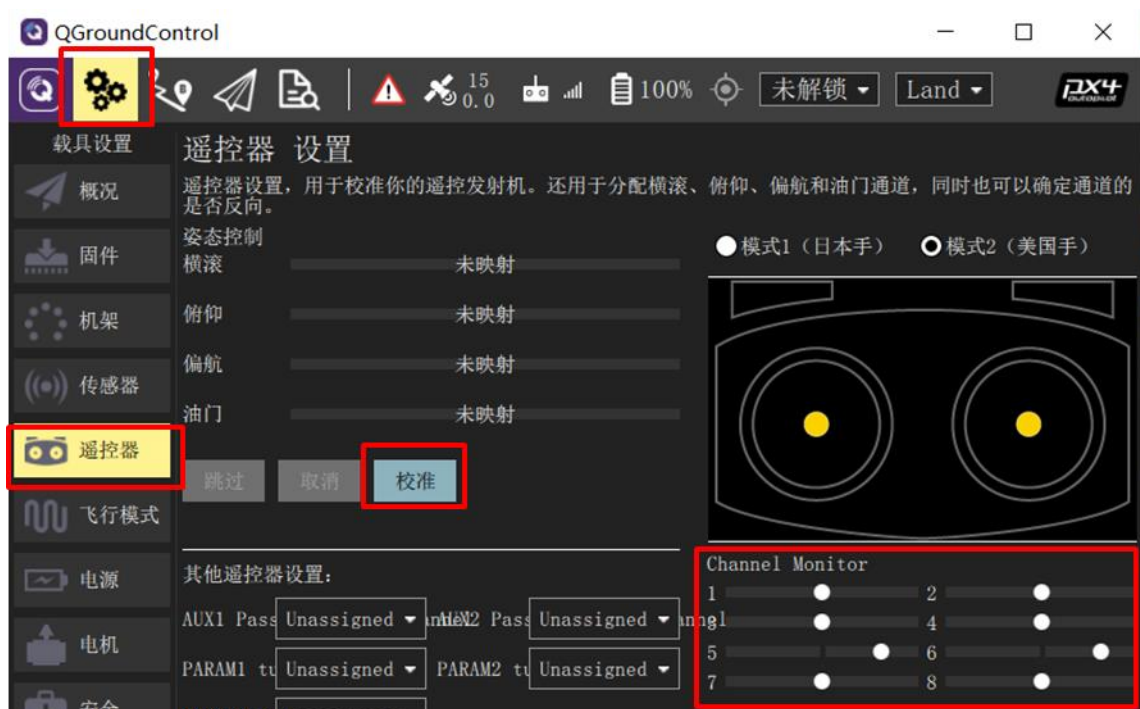
油门：控制上下运动，对应固定翼油门杆
偏航：控制机头转向，对应固定翼方向舵
俯仰：控制前后运动，对应固定翼升降舵
滚转：控制左右运动，对应固定翼副翼

配置与校准方法如下：

1. 正确连接 Pixhawk 与接收机，用 USB 数据线连接 Pixhawk 与电脑，打开遥控器，打开 QGroundControl 地面站软件，点击右下图所示的“Radio”（遥控器）标签页。



- 依次从左到右（或从上到下）拨动遥控器的 CH1 到 CH5 通道（见右上图），观测右下图地面站右侧红框区域中各个通道的白点。如果观测到：1、2、4、5、6 号小圆点从左到右移动（PWM 从 1100 到 1900）；3 号圆点从右向左移动，说明遥控器设置正确。否则需要重新配置遥控器。
- 点击右下图的“Calibrate”（校准）按钮，按提示可以校准遥控器。

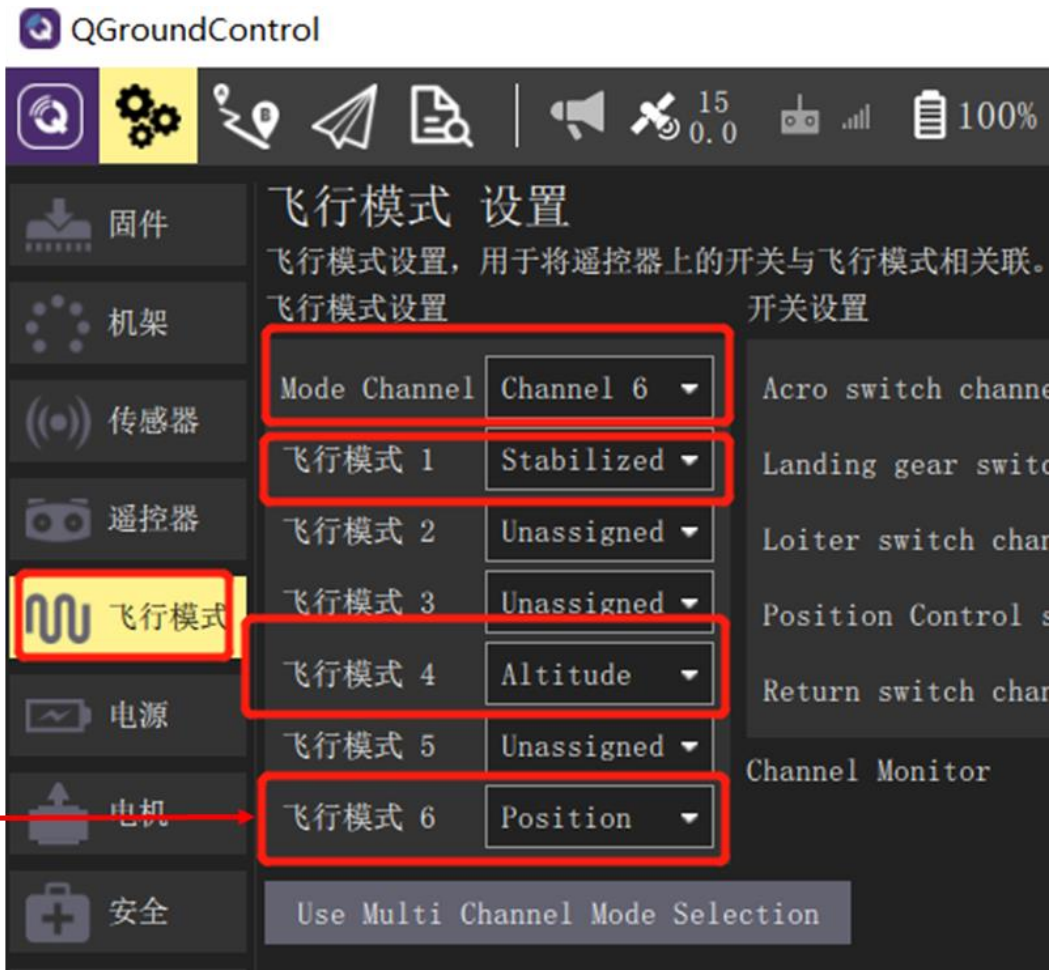


4. 点击 QGC 地面站上的“Calibrate” - “Next”按钮，然后依次将摇杆置于右图所示位置（根据 QGC 页面的实时提示）即可完成遥控器校准。



飞行模式设置

1. 经过上面的遥控器校准步骤后，点击地面站进入“Flight Modes”（飞行模式）设置页面，选择“Mode Channel”（模式通道）为前面测试过的 Channel 6 通道。由于 CH6 通道是一个三段开关，开关的顶部、中部、下部档位分别对应了“Flight Mode（飞行模式）1、4、6”三个标签。
2. 将这三个标签分别设置为“Stabilized”（自稳模式，只有姿态控制）、“Altitude”（定高模式，姿态和高度控制）和“Position”（定点模式，有姿态、定高和水平位置控制）。在后续的硬件在环仿真中，可以通过切换不同的模式来体验不同的控制效果。



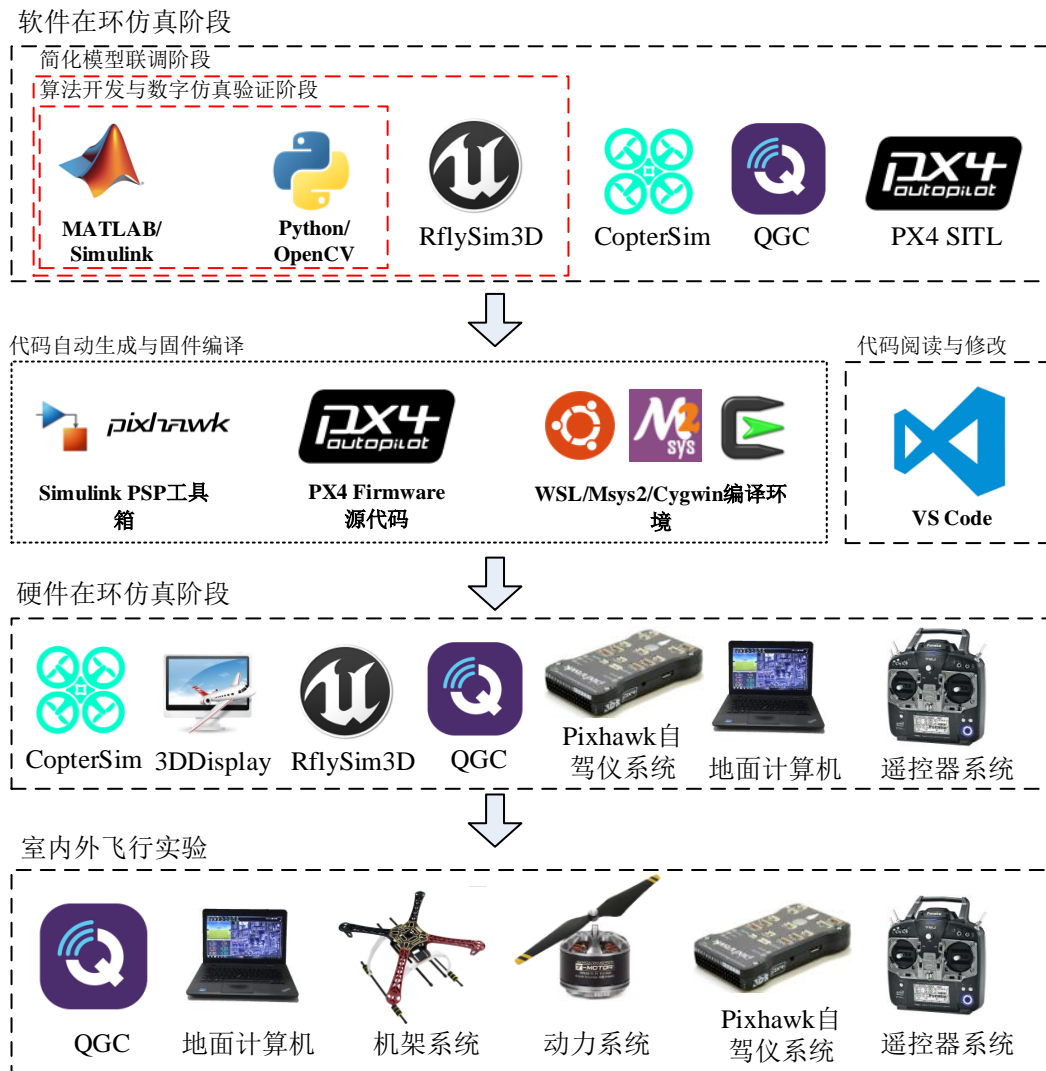
此外，工具链还支持的遥控器有乐迪 AT9S Pro、天地飞 ET07、福斯 i6s、Futuba T14S G 等。更多遥控器的详细配置请见：[1.BasicExps\el1_RC-Config\Readme.pdf](#)。

3. RflySim 工具链实验流程简介

3.1. 底层控制系统开发实验流程

底层控制系统开发依据从易到难的依次顺序分为：算法开发与数字仿真验证阶段、简化模型联调阶段、软件在环仿真阶段、硬件在环仿真阶段、室内外飞行实验；算法开发与数字仿真验证阶段主要任务是基于简化的无人系统模型，进行底层控制算法的开发及验证，使所开发的算法满足最初的需求。基于上一阶段的开发，简化模型联调阶段的主要任务是通过外部控制的方式，实现无人系统的顶层控制，使其无人系统的仿真形成一个闭环，达到无人系统联调的目的。软件在环仿真(Software-in-the-loop Simulation, SIL)是指在主机上编译生成的源代码并将其作为单独的进程执行。通过比较普通模式仿真结果和 SIL 仿真结果，测试模型与生成的代码之间是否存在数值等效性。RflySim 工具链的 SIL 仿真整个阶段可在 MATLAB/Python 环境下进行，利用给定无人载具仿真模型和例程，在 Simulink/Python 中进行控制算法设计，并正确连接模型和控制器，确保输入输出信号与实际无人系统一致。如：多旋翼无人机系统：多旋翼模型将传感器数据或状态估计信息（例如，姿态角、

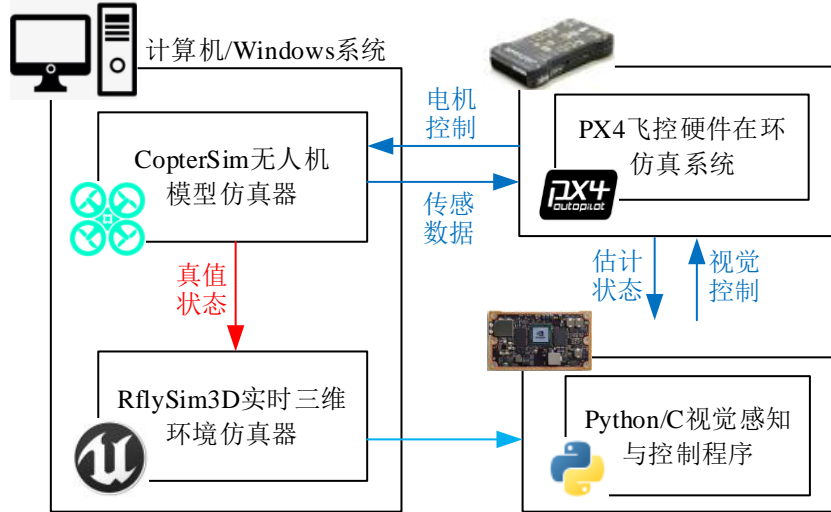
角速率、位置和速度等) 发送给控制器, 控制器将每个电机 PWM 控制指令发回给模型, 从而形成一个 SIL 仿真闭环系统。用户可以观察控制性能, 自行修改或设计控制器来达到期望的性能需求。硬件在环仿真(Hardware-in-the-loop Simulation, HIL)阶段是一种用于实时嵌入式系统的开发和测试技术。HIL 仿真提供动态系统模型, 可以模拟真实的系统环境, 加入相关动态系统的数学表示法, 并通过嵌入式系统的输入输出将其与仿真系统平台相连 [3]。RflySim 工具链可将 Simulink 无人载具模型参数导入到 CopterSim 中, 并将 Simulink 控制器算法生成代码下载到 Pixhawk 自驾仪, 然后用 USB 实体信号线替代 Simulink 中的虚拟信号线。CopterSim 将传感器数据 (例如, 加速度计、气压计、磁力计等) 通过 USB 数据线发送给 Pixhawk 系统; Pixhawk 系统中的 PX4 自驾仪软件将收到传感器数据进行滤波和状态估计, 将估计的状态信息通过内部的 uORB 消息总线发送给控制器; 控制器再通过 USB 数据线将每个电机的 PWM 控制指令发回给 CopterSim, 从而形成一个硬件在环仿真闭环。



3.2. 顶层控制系统开发实验流程

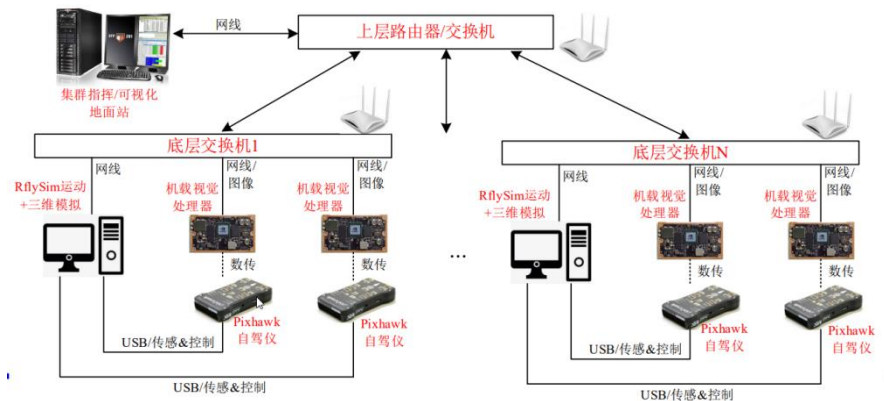
3.2.1. 机载板卡硬件在环阶段

基于 HIL 仿真阶段，本阶段加入更多的硬件，如：组网通信模块、视觉处理模块、数据采集模块等。在这个阶段，我们需将系统部署到实际的硬件设备上，对不同的硬件进行集成和调试，以确保它们能够相互配合，以实现整个系统的高效运行。这个阶段是整个开发过程中的重要环节，也是确保系统最终能够在实际场景中正常运行的关键阶段。



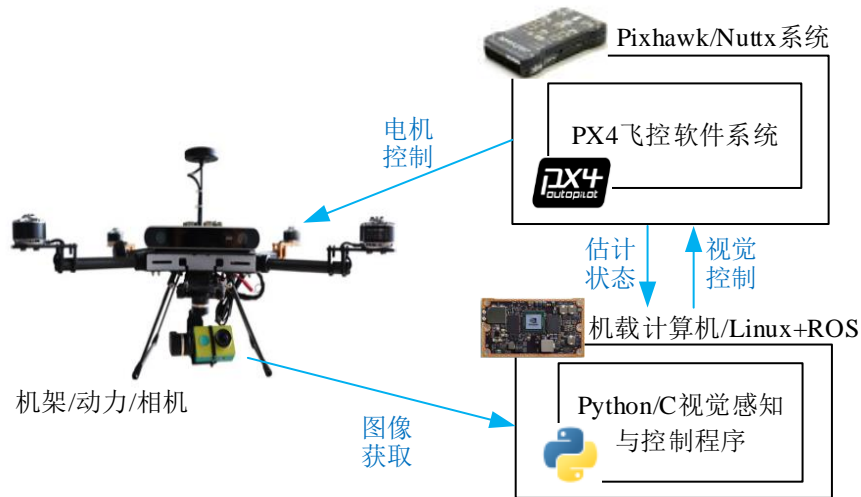
3.2.2. 多机 HIL 仿真阶段

本阶段的整个仿真系统可看作是由多个 HIL 仿真的子系统组成，但不是简单的子系统叠加，需要考虑不同硬件之间的系统拓扑和配置、网络和通信的模型结构以及仿真主机的资源调度和管理等。仿真过程中需对各个硬件设备进行适当的配置和调试，确保各个硬件设备可以正常工作。无论从多机 HIL 仿真的角度，还是从无人系统真实集群控制的角度，通信带宽和计算性能永远是制约集群数量增加的重要瓶颈。由于仿真计算机的性能存在瓶颈，且单台计算机只能够连接数量有限的 Pixhawk 进行 HIL 仿真。同时，随着无人机数量的增加，飞机间相互通信的数据量暴增，直到通信带宽达到饱和。因此，RflySim 工具链通过多台电脑组网方式，实现无人机数量的任意扩展，将无人机集群整体分为若干子群，采用网络分层的方式，实现更大规模的集群仿真，如[错误!未找到引用源。](#)所示。



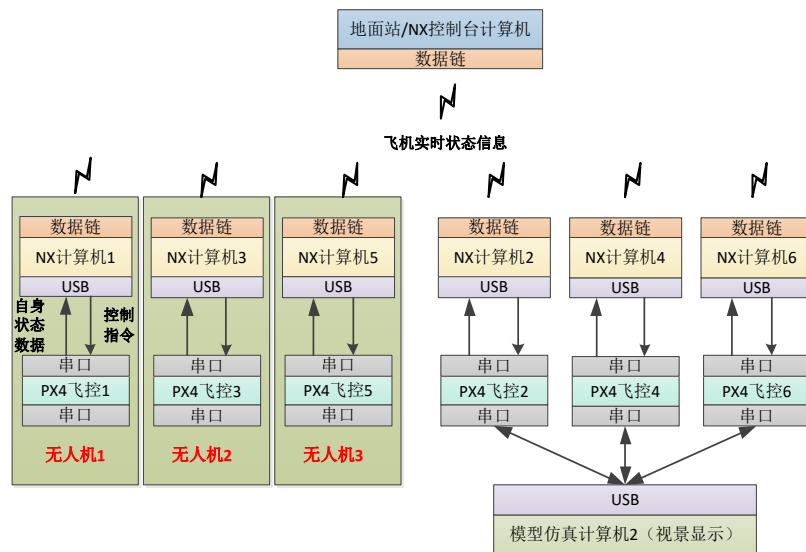
3.2.3. 单机自主控制阶段

单机自主控制即单机在没有人干预的情况下独立自主进行任务规划和执行的能力。如一架自主式无人机应该具备内部和外部状态感知系统、机载系统之间的内部通讯链、机载故障管理系统和面对战场环境变化的任务重规划系统^{错误!未找到引用源。}。基于 RflySim 工具链可快速开发出无人系统领域的顶层控制算法，如：SLMA 算法、轨迹规划算法、避障算法等。



3.2.4. 半实物集群控制阶段

半实物集群控制阶段是指在仿真实验系统的仿真回路中接入部分实物的实时仿真。它是一种在复杂控制系统开发过程中的典型应用，可以用来验证和优化集群协同控制决策算法。本书第 10.3.5 小节即为多机半实物仿真案例，实验以无人机集群超低空避碰算法的开发和验证为依据，采用“3 实 3 虚”的固定翼无人机半实物仿真实验，完成无人机密集集群超低空避碰算法移植验证、无人机数字孪生模型开发、虚实协同等实验任务。



3.2.5. 真机集群控制阶段

真机集群控制阶段是指在真实的环境中，使用真实的设备，进行集群控制的实验或测

试，可以验证系统的性能和可靠性。真机集群控制阶段需要考虑很多实际的因素，如通信、干扰、故障、安全等。例如：使用光学动捕系统进行无人机的室内动捕，来捕捉无人机的位置、姿态、速度等运动信息。可以用于研究和验证无人机的运动控制、导航、编队、协同等功能。一般需要使用高速、高分辨率的摄像头，以及在无人机上贴上反光标记，来实现高精度、高实时性、高稳定性的动捕效果。



3.2.6. 完全真实环境下的多机协同阶段

本阶段是指在真实的环境中，使用多架无人机，通过机间通信和群体智能，合作分工完成某一共同任务的阶段，是无人机集群协同技术的最高水平，用于研究和验证无人机集群的路径规划、态势感知、任务协作等功能^{错误!未找到引用源。}。无人机完全真实环境下的多机协同阶段需要使用高性能、高可靠性、高安全性的无人机、通信、控制系统，来实现高效率、高灵活性、高鲁棒性的协同效果。如^{错误!未找到引用源。}所示为无人机集群虚实结合仿真构架。

