



---

# Multicopter Design and Control Practice Experiments

## **RflySim Advanced Courses** **Lesson 04: Other Types of Vehicles**

Dr. Xunhua Dai, Associate Professor,  
School of Computer Science and Engineering,  
Central South University, China;

Email: [dai.xh@csu.edu.cn](mailto:dai.xh@csu.edu.cn) ;

<https://faculty.csu.edu.cn/daixunhua>



# Content

---

## 1. Configuration of software & hardware

## 2. Other types of multicopters

Path of demo source code of this lesson: "RflySimAPIs\OtherVehicleTypes"

## 3. User-defined fixed-wing aircraft

## 4. User-defined vehicle models

## 5. Summary

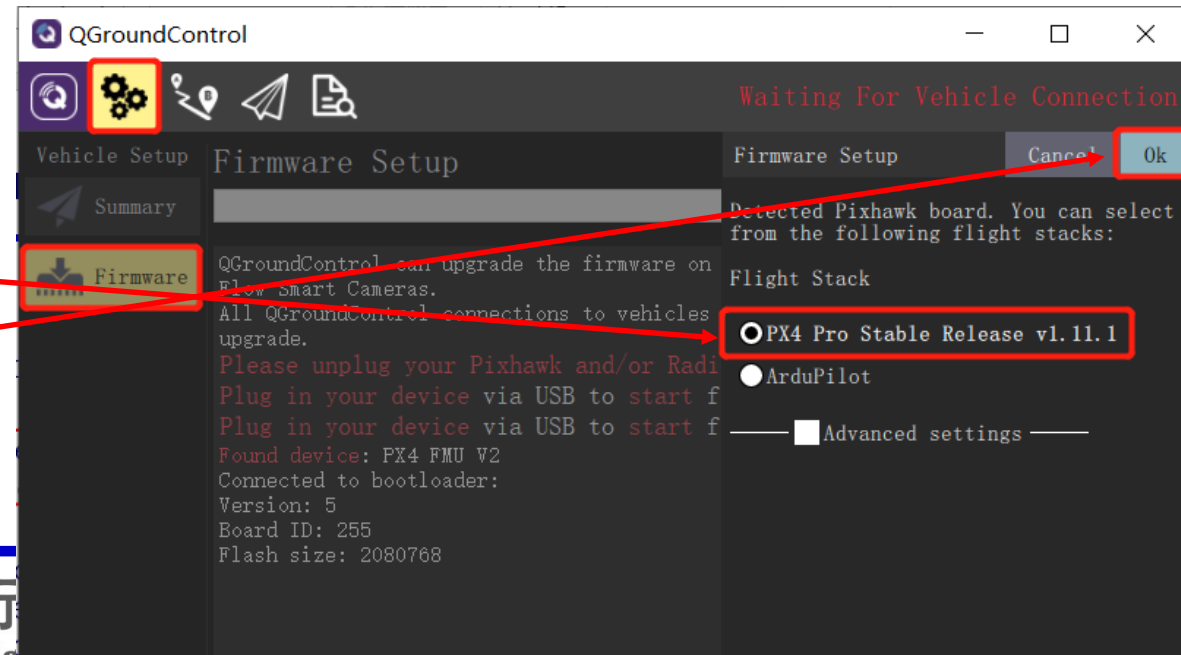
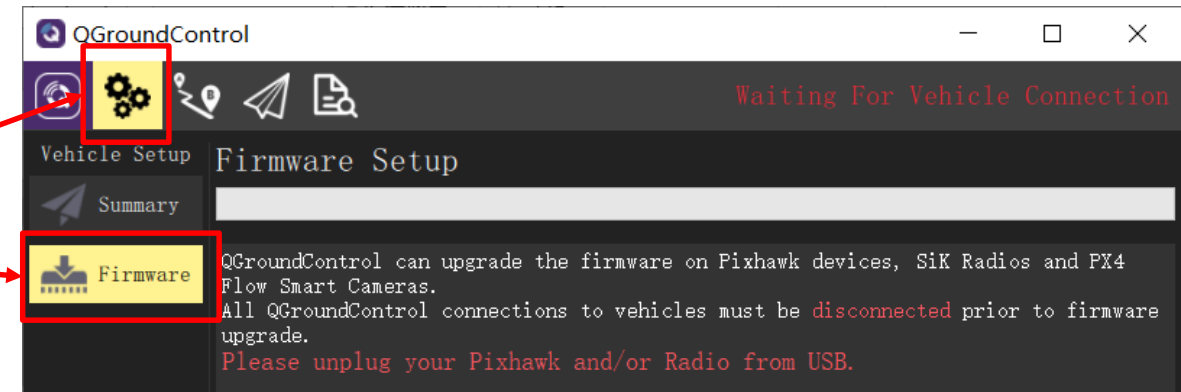


# 1. Configuration of software &

**Note:** The examples in this section need to use the official firmware of PX4, any version is fine, the latest firmware is selected here

## 1.1 Pixhawk hardware configuration (1) — online firmware download

- 1) Open the QGC ground station software;
- 2) As shown in the figure on the right, click the **gear icon** in the toolbar to enter the settings page, and then click the "**Firmware**" label to enter the firmware burning page;
- 3) Connect the Pixhawk autopilot to the computer with a USB cable. The software will automatically recognize the Pixhawk hardware, as shown in the lower right figure, the firmware configuration window will pop up on the right side of the interface, check "**PX4 \*\*\***", and then click "**OK**", QGC will start to download automatically (Internet connection is required; if no internet, please refer to the next page to use the local firmware file) and install the latest PX4 firmware into Pixhawk;



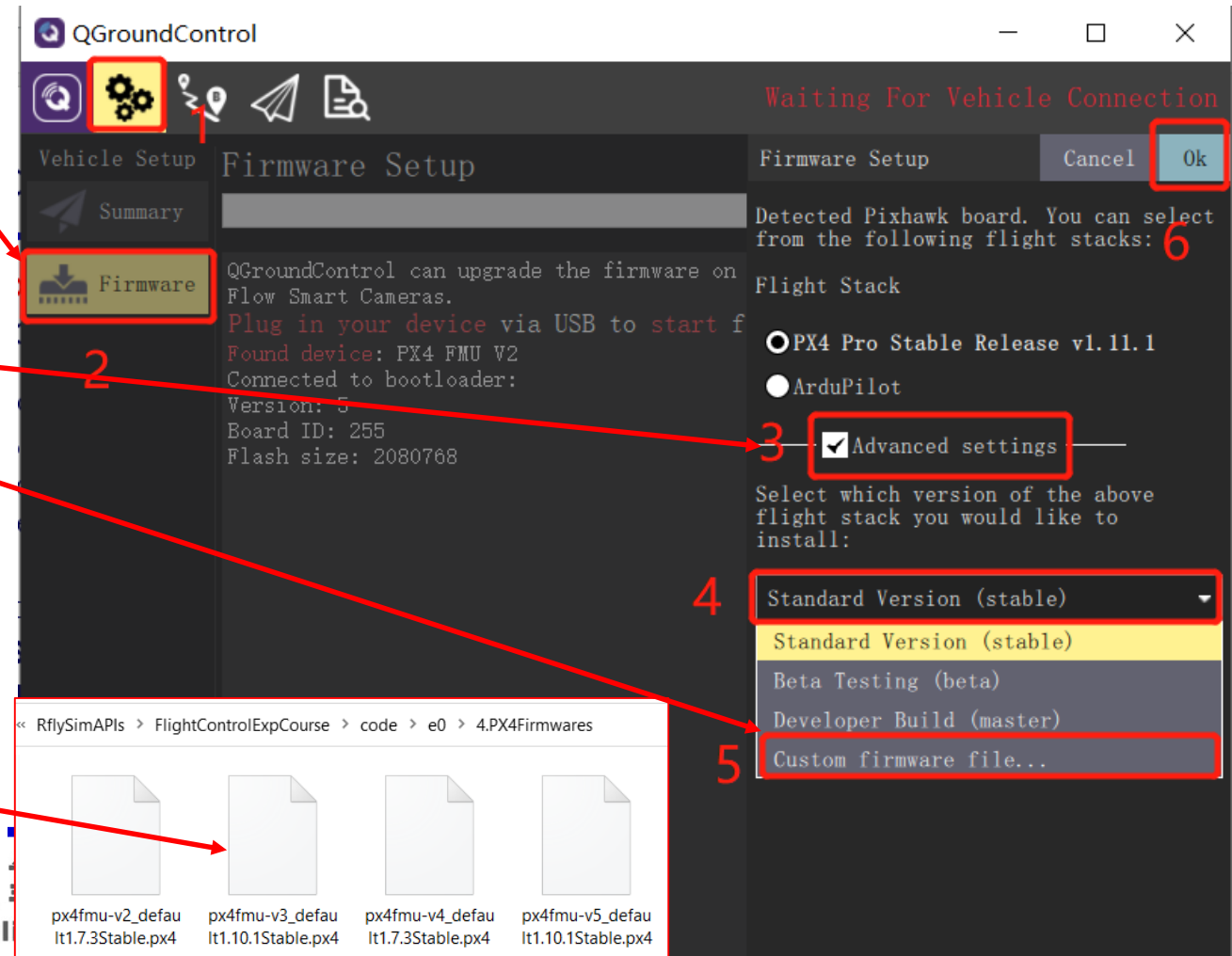


# 1. Configuration of software

Note: If there is no required firmware file in the folder, please visit <https://github.com/PX4/Firmware/releases> to download and upload to Pixhawk

## 1.2 Pixhawk hardware configuration (2) — offline firmware download (when the firmware cannot be downloaded online due to the network)

- 1) Open the QGC ground station software;
- 2) Click the "**Firmware**" tab, and connect the Pixhawk autopilot with the USB data cable. The ground station will automatically detect the autopilot
- 3) Check the "**advanced settings**" checkbox;
- 4) Click **triangle icon** besides the "**Standard Version (stable)**" tab – "**Custom firmware file**", then click "**OK**";
- 5) In the file selection page that pops up, if you use Pixhawk1 flight control, select "**RflySimAPIs \ FlightControlExpCourse \ code \ e0 \ 4.PX4Firmwares \ px4fmu-v3\_default1.10.1Stable.px4**", if you use Pixhawk 4, select **fmv-v5** firmware





# 1. Configuration of software & hardware

## 1.3 Configure C++ Compiler for MATLAB

- Enter the command "**mex -setup**" in the MATLAB command line window
- Generally speaking, the VS 2017 compiler will be automatically recognized and installed. As shown in the right figure, "**MEX configured to use 'Microsoft Visual C++ 2017' for**", indicating that the installation is correct
- This page can also switch to other compilers such as Visual Studio 2013/2015/2017

```
Command Window
>> mex -setup
MEX configured to use 'Microsoft Visual C++ 2017 (C)' for C language compilation.
Warning: The MATLAB C and Fortran API has changed to support MATLAB
variables with more than 2^32-1 elements. You will be required
to update your code to utilize the new API.
You can find more information about this at:
http://www.mathworks.com/help/matlab/matlab\_external/upgrading-mex-files-to-u

To choose a different C compiler, select one from the following:
Microsoft Visual C++ 2013 (C) mex -setup:D:\MATLAB\R2017b\bin\win64\mexopts\msvc2
Microsoft Visual C++ 2015 (C) mex -setup:D:\MATLAB\R2017b\bin\win64\mexopts\msvc2
Microsoft Visual C++ 2017 (C) mex -setup:C:\Users\dream\AppData\Roaming\MathWorks

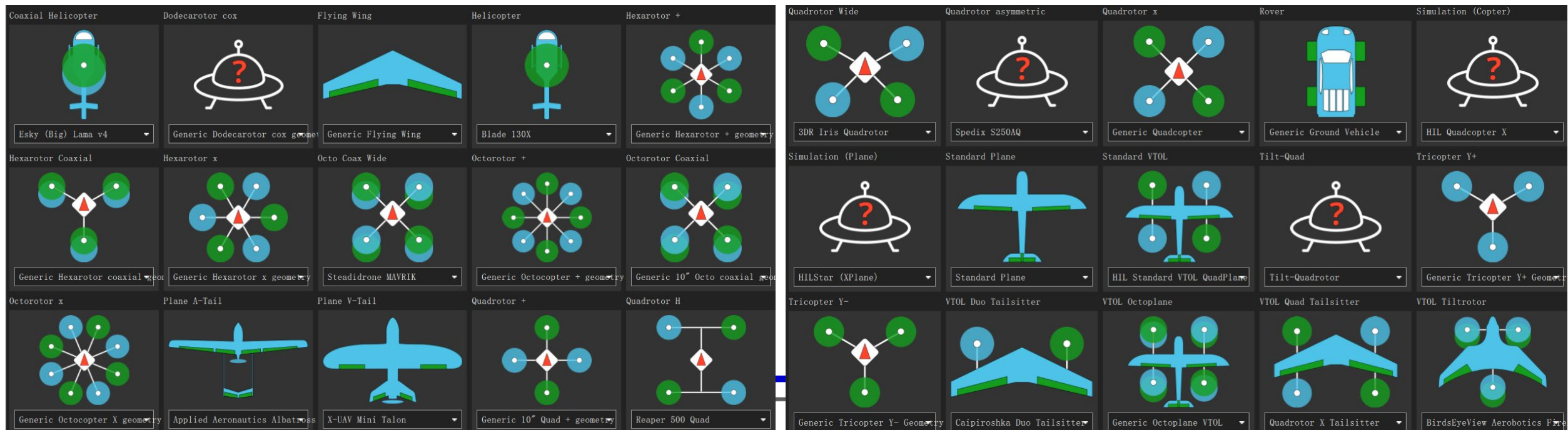
To choose a different language, select one from the following:
mex -setup C++
mex -setup FORTRAN
fx >>
```



# 1. Configuration of software & hardware

## 1.4 Supported airframes for SIL and HIL simulation

- RflySim platform support all airframes that defined in PX4 flight controller
- These airframes can be found in QGroundControl Airframe tab
- Currently, we have developed models for multicopters and fixed-wing aircraft. User has to build model in Simulink to simulation other vehicles.





# 1. Configuration of software & hardware

---

## 1.5 Basic steps for SIL/HIL simulation of other vehicle types

- **HIL simulation steps:** configure model parameters in CopterSim or import DLL model from Simulink → configure Pixhawk to desired airframe in QGC → configure Pixhawk to HIL simulation mode in QGC → start simulation in CopterSim
- **SIL simulation steps:** onfigure model parameters in CopterSim or import DLL model from Simulink → configure airframe file in PX4 firmware → select desired airframe in bat one-key script → start simulation with one-key script
  - **Steps to configure airframe file in PX4 firmware:** copy desired airframe file (e.g., 6001\_hexa\_x for hexacopter and 2100\_standard\_plane for fixed-wing aircraft) from folder “Firmware\ROMFS\px4fmu\_common\init.d\airframes” to the folder “Firmware\ROMFS\px4fmu\_common\init.d-posix”
  - **Bat script modification:** copy a SITLRun.bat file, and use txt editor to modify the PX4SITLFrame option to the non-numeric part of the above airframe file. E.g., “set PX4SITLFrame=hexa\_x” for hexacopters and “set PX4SITLFrame=standard\_plane” for fixed-wing aircraft. Similar methods can be applied to other airframe types in PX4 firmware.
  - **Select 3D map:** we recommend to use OldFactory map, which has flat runway for fixed-wing aircraft to takeoff. Three modifications are required: choose a map “SET /a UE4\_MAP=OldFactory”, X coordinate: “SET /a ORIGIN\_POS\_X=-250”, Y coordinate: “SET /a ORIGIN\_POS\_Y=-119”, then you can start simulation at a runway.



# Content

---

1. Configuration of software & hardware
2. Other types of multicopters
3. User-defined fixed-wing aircraft
4. User-defined vehicle models
5. Summary

Path of demo source code of this lesson: "RflySimAPIs\OtherVehicle Types\MulticopterModelCTRL"

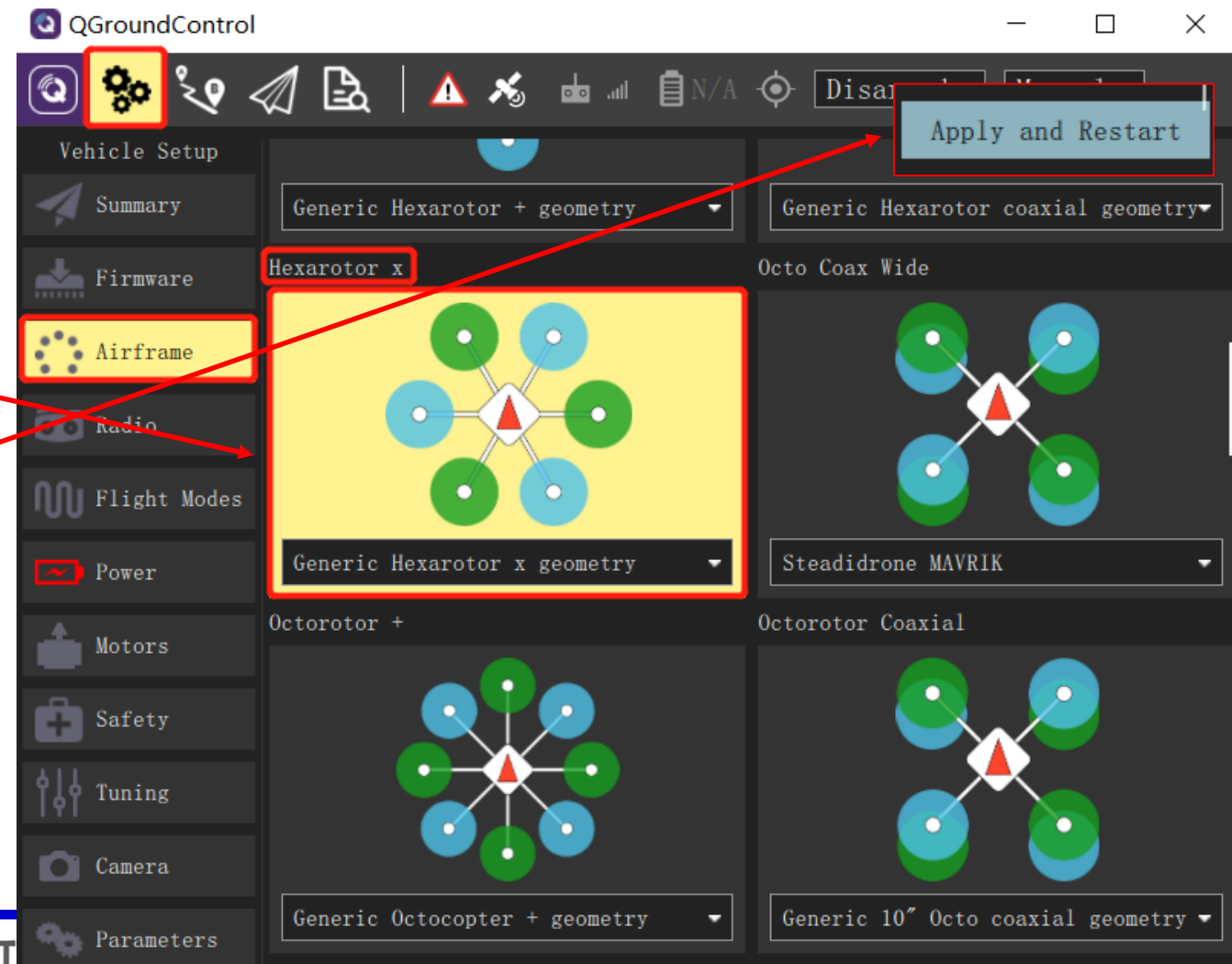




## 2. Other types of multicopters

### 2.1 Hexacopter airframe selection for Pixhawk via QGC

- After the firmware is uploaded, the autopilot will automatically restart and re-connect to the QGC; at this time, as shown on the right, enter the “**Airframe**” tab, select the airframe as “**Hexarotor x**” – “**Generic Hexarotor x geometry**”, and then click on the “**Apply and Restart**” button, the autopilot will restart automatically at this time;
- After restarting, QGC will automatically look for the serial port and connect to Pixhawk. At this time, check each configuration page to ensure that everything is OK.

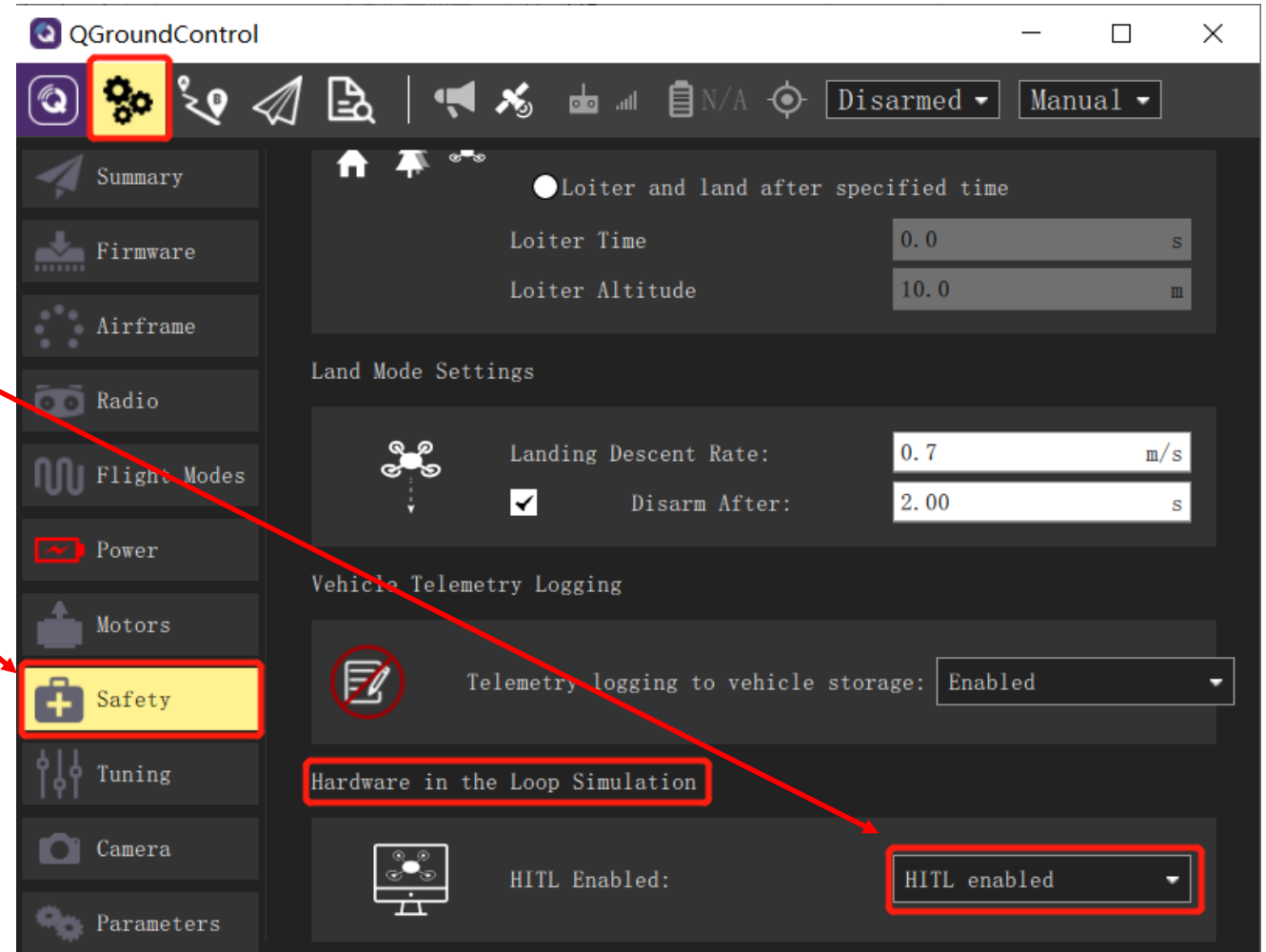




## 2. Other types of multicopters

### 2.2 Set HIL mode for Pixhawk via QGC

- Enter “**Safety**” page, and find “**Hardware in the Loop Simulation**” item to select “**HITL enabled**” to enable HIL simulation mode for Hexacopter.
- After finishing the above steps, re-plug the Pixhawk on the computer, after QGC reconnects to Pixhawk, go to “**Radio**” and “**Flight Modes**” page to check whether the RC is correctly calibrated and the Flight modes are set as required.

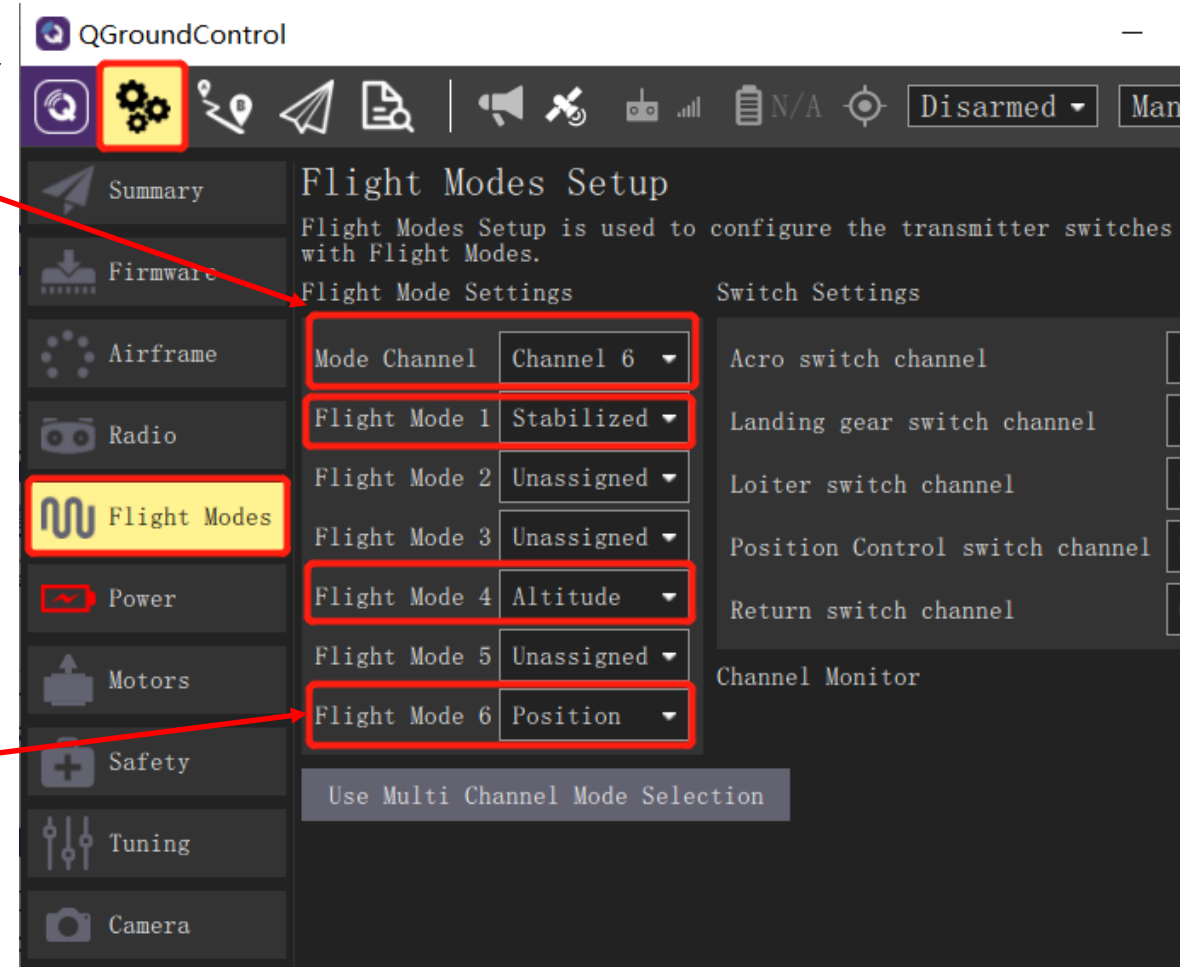




## 2. Other types of multicopters

### 2.3 Flight mode setting for Stabilized Mode

- After the above RC transmitter calibration steps, click on the QGC ground station to enter the "**Flight Modes**" setting page, and select "**Mode Channel**" as the previously tested CH6 channel. Since the CH6 channel is a three-stage switch, the top, middle, and lower positions of the switch correspond to the three labels "**Flight Mode 1, 4, and 6**" respectively.
- As shown in the figure on the right, set these three labels to "**Stabilized**" (self-stabilization mode, only attitude control), "**Altitude**" (fixed height mode, with attitude and height control) and "**Position**" (fixed-point mode, with attitude, fixed height and horizontal position control).
- Push the CH6 stick to ensure Pixhawk enter the first mode "**Stabilized**".

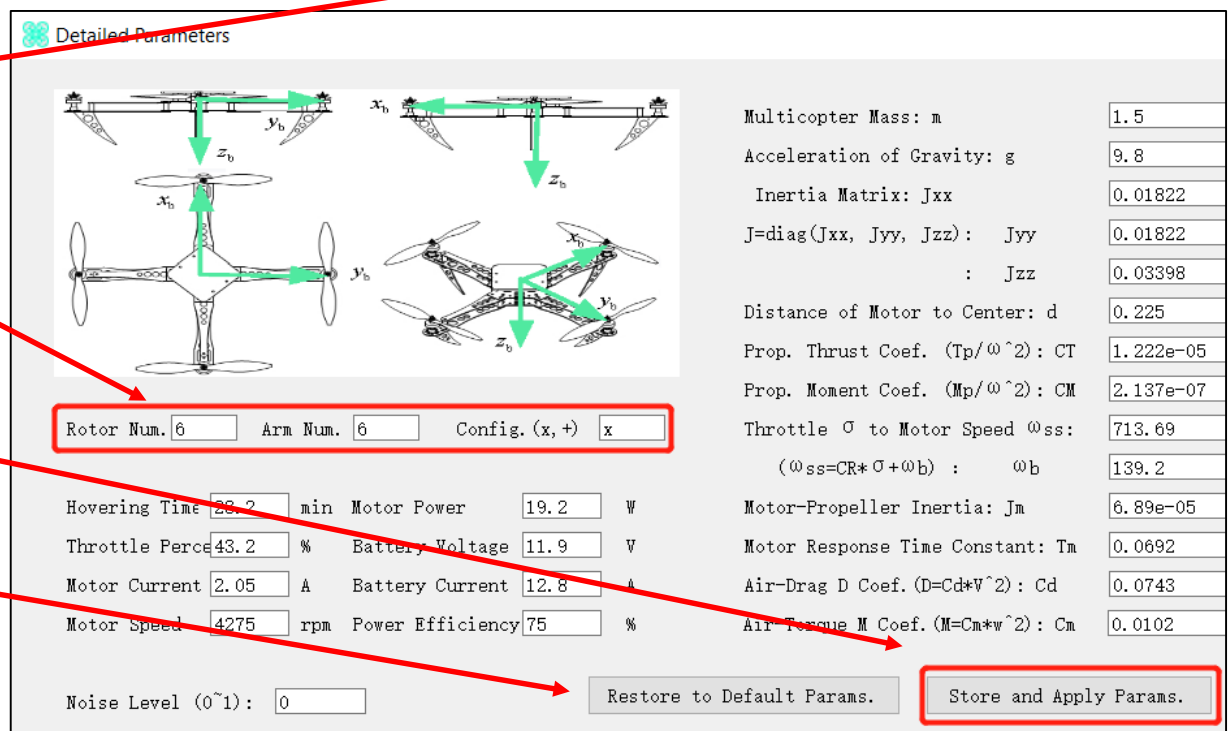
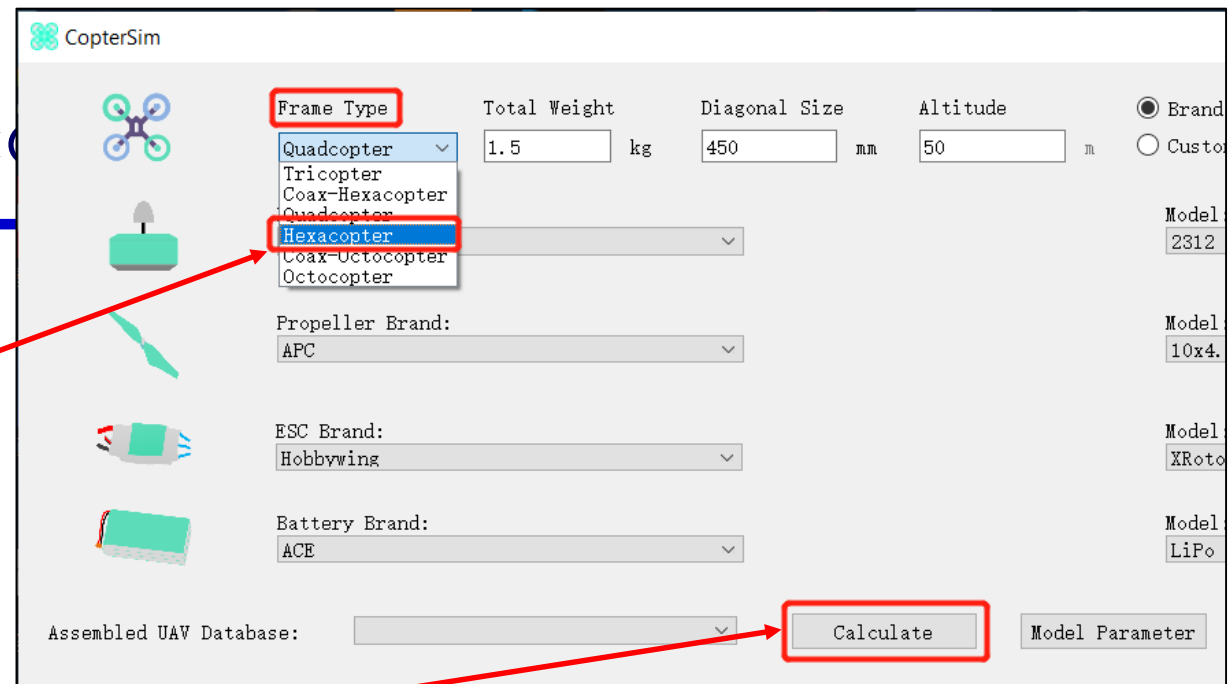




## 2. Other types of multicopter

### 2.4 Set hexacopter model on CopterSim --- Method 1: assemble with component

- Open CopterSim, and select “**Hexacopter**” option in the “**Frame Type**” drop-down list .
- Input appropriate parameters in the CopterSim UI to meet your requirement, including “**Total Weight**”, “**Diagonal Size**”, and motor, ESC components of the propulsion system. (**Note**, you can just use the default parameters for simplicity).
- Click the “**Calculate**” button to obtain model parameters for the desired hexacopter, and the parameters can be viewed and modified in the pop-up window.
- Click “**Store and Apply Params**” button to store parameters for future simulation if the model parameters are OK.
- **Note**: Please click “**Restore to Default Params**” button to restore parameters for future simulation when you want to simulate quadcopter.



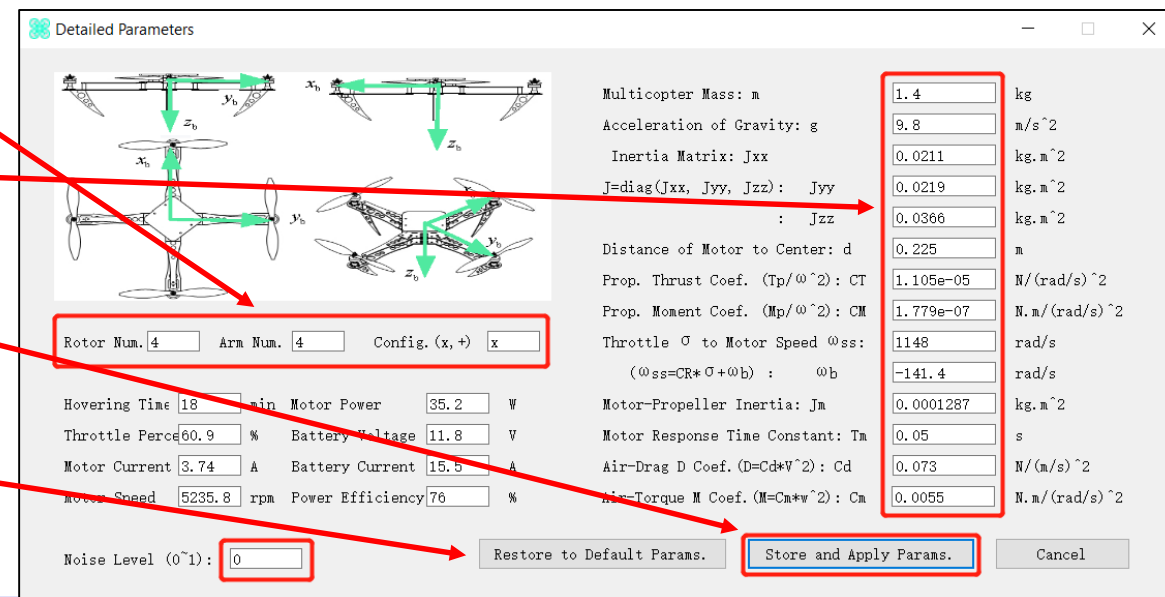
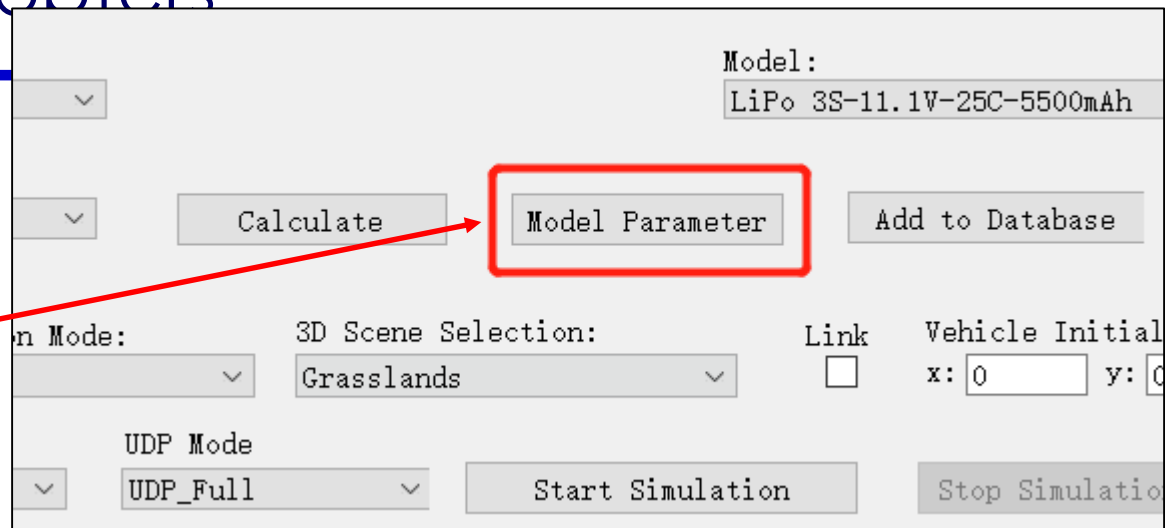


## 2. Other types of multicopters

### 2.4 Set hexacopter model on CopterSim

#### --- Method 2: input model parameters

- Open CopterSim, and click the “**Model Parameter**” button.
- Manually input appropriate hexacopter parameters in the parameter window to meet your requirement, including “**Rotor Num.**”, “**Arm Num.**” to configure a hexacopter frame
- And parameters “**Total mass**”, “**moment of inertia**” to configure a vehicle model.
- Click “**Store and Apply Params**” button to store parameters for future hexacopter simulation.
- **Note:** Please click “**Restore to Default Params**” button to restore parameters for future simulation when you want to simulate quadcopter.

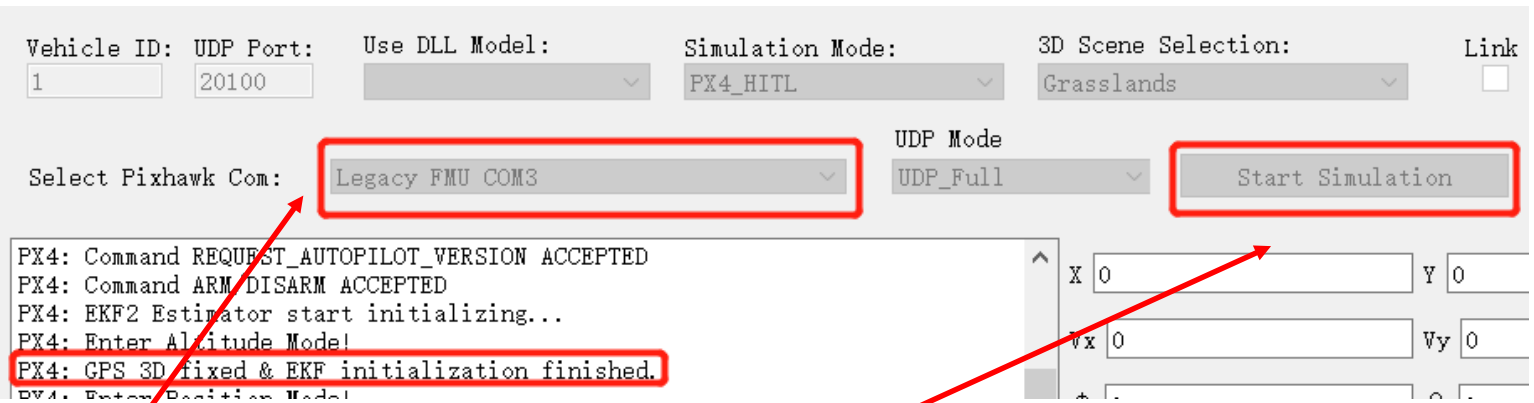




## 2. Other types

### 2.5 HIL simulation for hexacopter

- Assuming Pixhawk and CopterSim are both set to hexacopter mode, open CopterSim, connect Pixhawk on the computer. Then, close QGC for short (prevent occupying COM port) and select Pixhawk COM port on CopterSim, and you can start HIL simulation by click the “**Start Simulation**” button.
- Open RflySim3D, and now you can see a hexacopter on the grassland. Open QGC here, you can connect to Pixhawk to observe states and control the hexacopter to takeoff and fly.
- When CopterSim shows “**GPS 3D fixed & ...**” on the lower left message box, you can use RC to arm and control the vehicle.

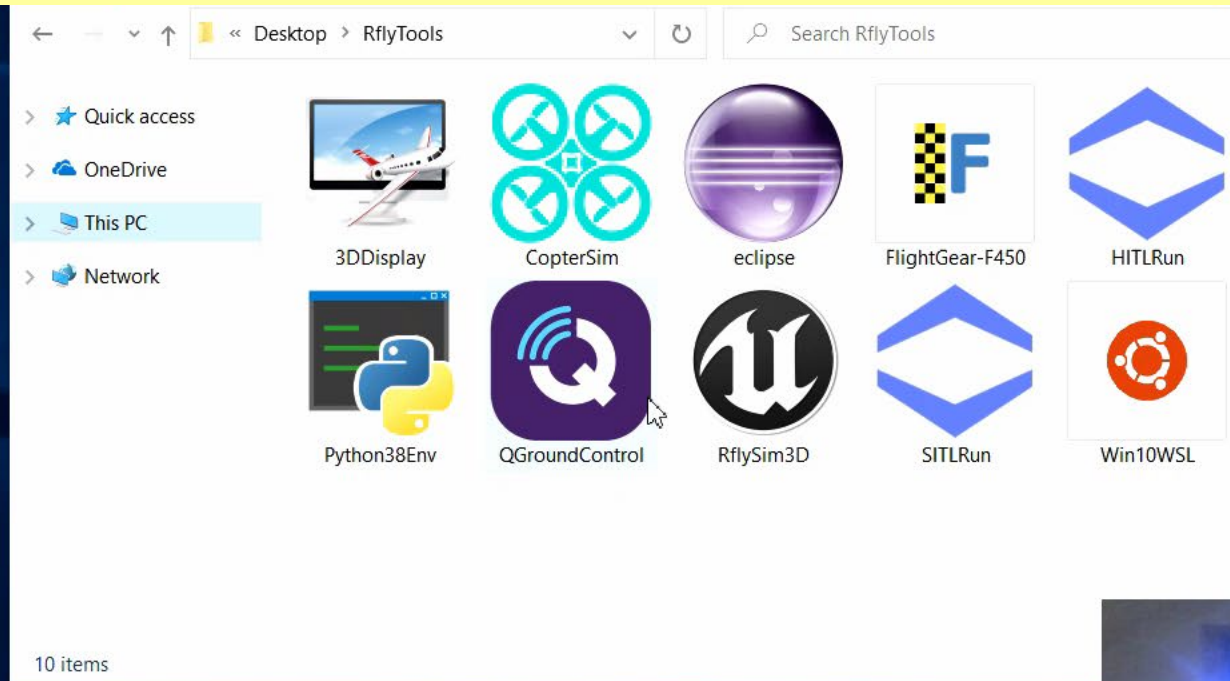


## RflySim: How to simulate other types of multicopters

Watch this video by clicking the following links:

YouTube: <https://youtu.be/bryZXxaJE5M>

Youku: [https://v.youku.com/v\\_show/id\\_XNDcwNjA4MDY2MA==.html](https://v.youku.com/v_show/id_XNDcwNjA4MDY2MA==.html)



# Connect Pixhawk and open QGC

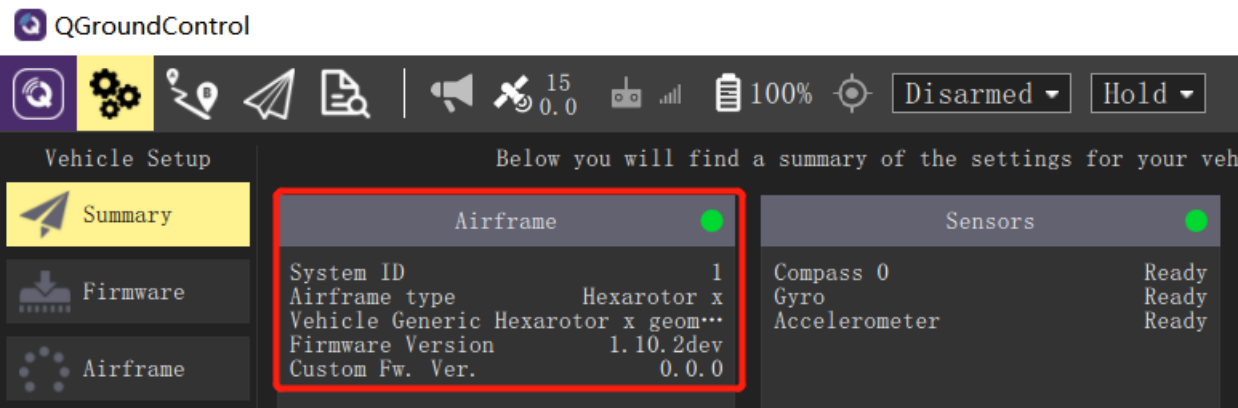
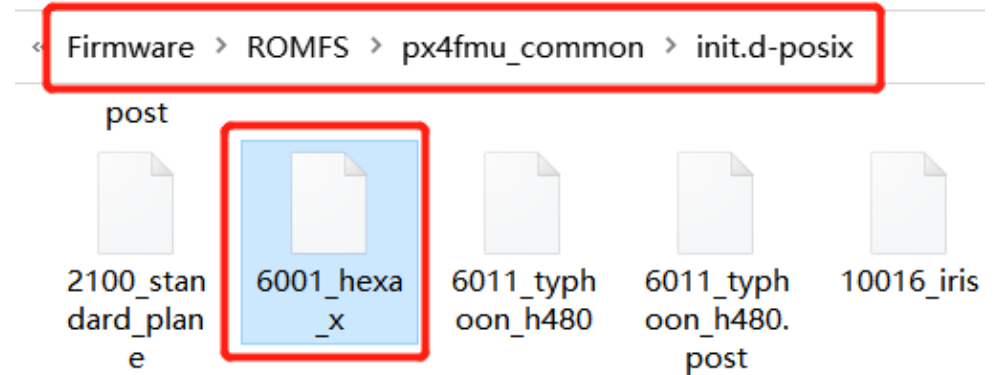




## 2. Other types of multicopters

### 2.5-2 SIL simulation for hexacopters

- Follow the steps in the above section to configure hexacopter X mode in CopterSim
- Copy the airframe file “6001\_hexa\_x” from folder “Firmware\ROMFS\px4fmw\_common\init.d\airframes” to the folder “Firmware\ROMFS\px4fmw\_common\init.d-posix”
- Copy and save a SITLRun.bat file, and modify the airframe code to “set PX4SITLFrame=hexa\_x”
- Run the bat script and enter vehicle number (support swarm simulation) to start simulation



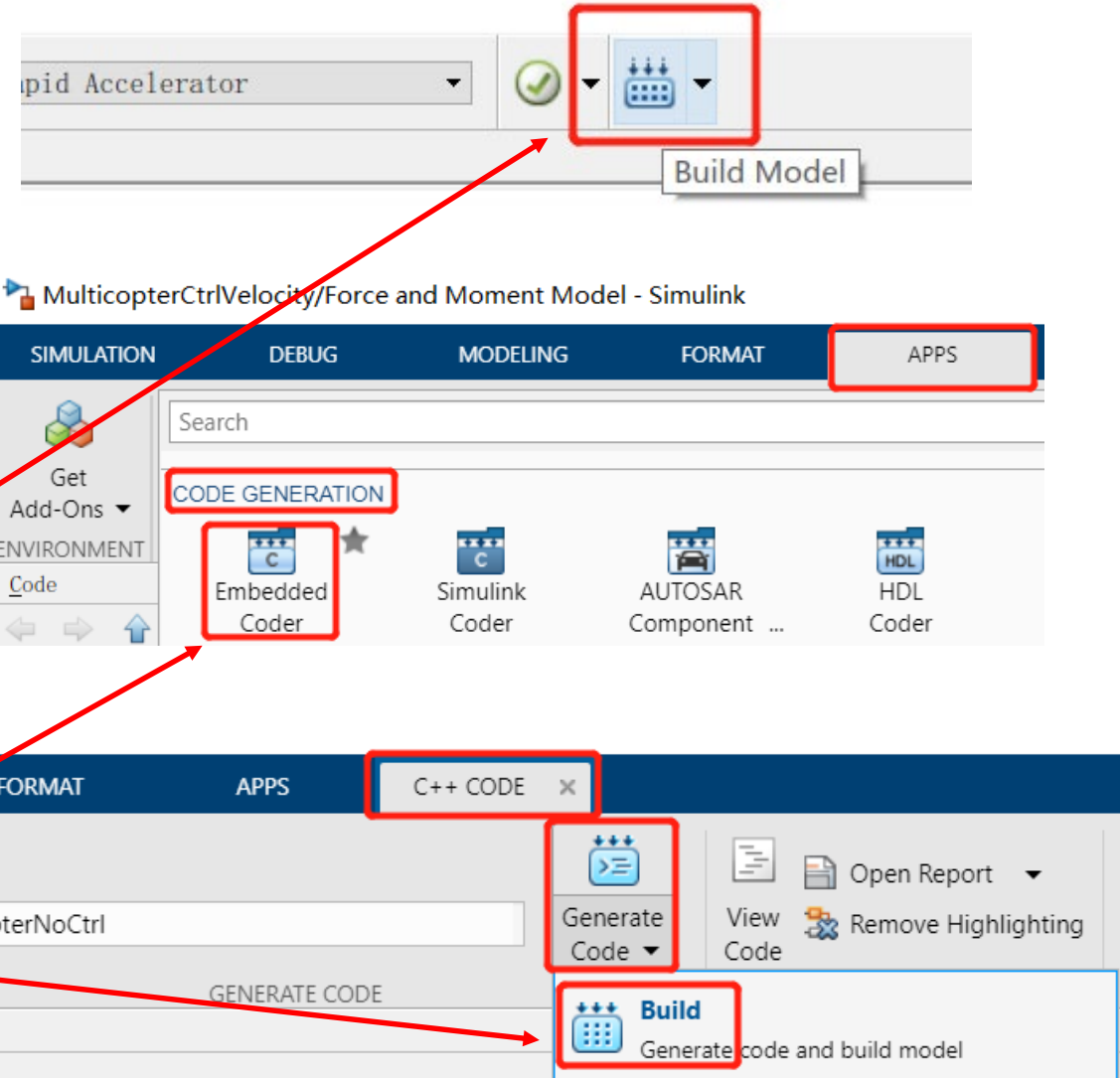




## 2. Other types of multicopters

### 2.6 Develop other multicopter models in Simulink

- Open MATLAB, enter the demo path “**RflySimAPIs\OtherVehicleTypes\MulticopterModelCTRL**”, open the “**MulticopterNoCtrl.slx**” file to obtain a unified multicopter model template.
- Click the “**Build**” button on Simulink then the C/C++ code for the model can be obtained, the steps are as follows.
- For **MATLAB 2019a** and before, the “**Build**” button is on the Simulink toolbar as presented on the figure upper right
- For MATLAB 2019b and above, you should click “**APPS - CODE GENERATION - Embedded Coder**” to get the “**C++ CODE**” page, where you can find the “**Build**” button to click to generate code.

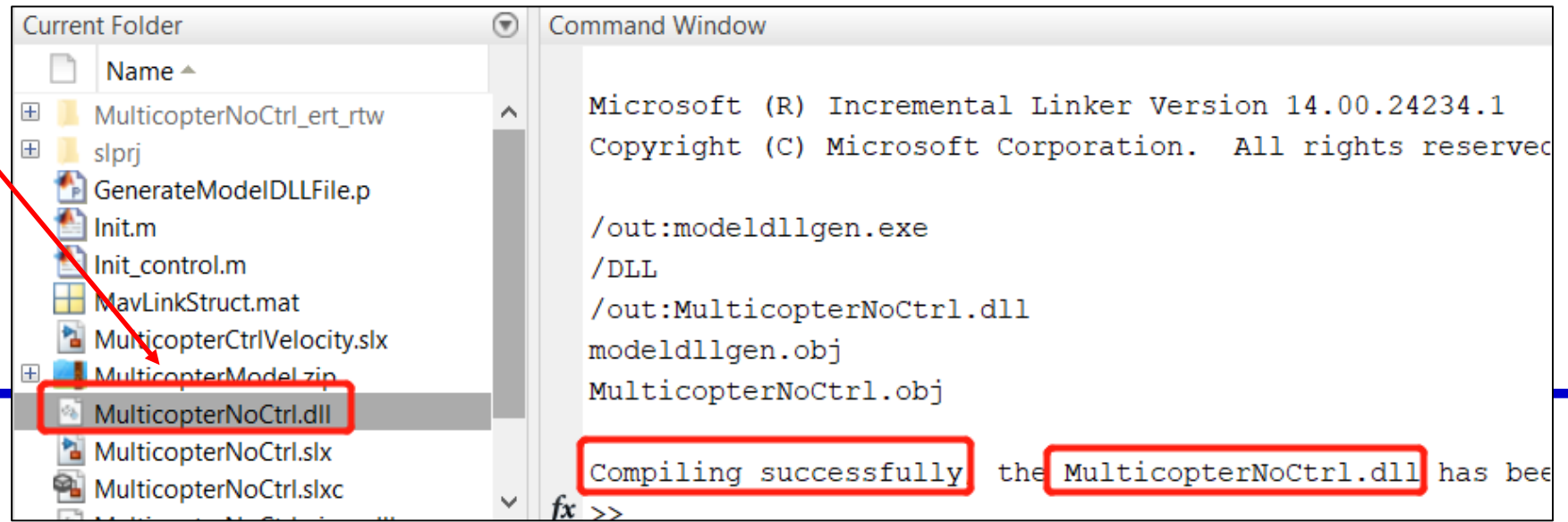
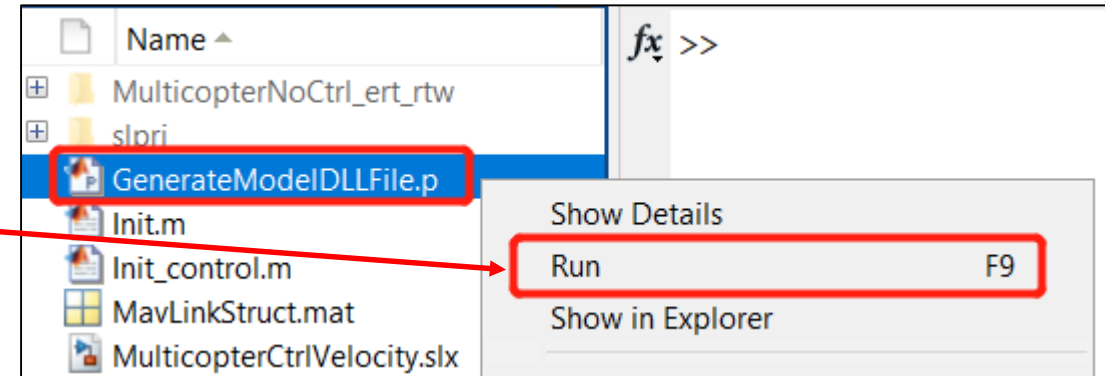




## 2. Other types of multicopters

### 2.7 Generate DLL model file for CopterSim

- In MATLAB, right-click the “**GenerateModelDLLFile.p**” file with the mouse, and click the “**Run**” button to compile the C/C++ code to DLL model file for CopterSim simulation.
- The obtained **.dll** file name is the same as the **.slx** file name. For example, in this demo is “**MulticopterNoCtrl.dll**”.



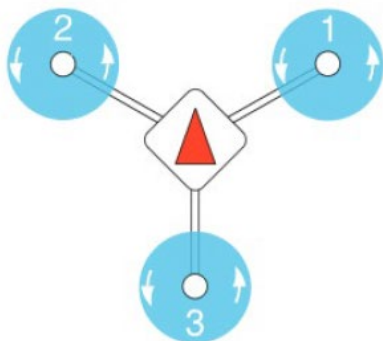


## 2. Other types of multicopters

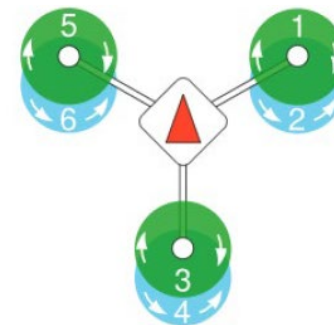
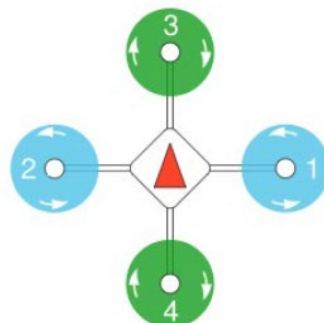
### 2.8 Develop model for other multicopters

- This demo is a quadcopter in default, but it can be modified to other types of multicopter by changing parameter “**ModelParam\_uavType**” in the “**init.m**” file. Some of the supported multicopter types are listed as follows. Please read the “**readme.docx**” for details.

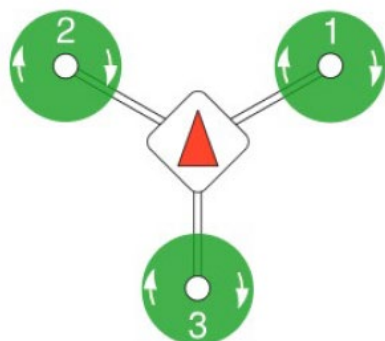
ModelParam\_uavType = 1: Tricopter Y+



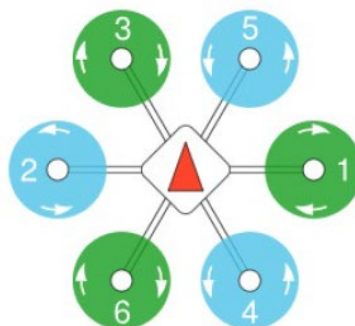
ModelParam\_uavType = 4: Quadrotor +



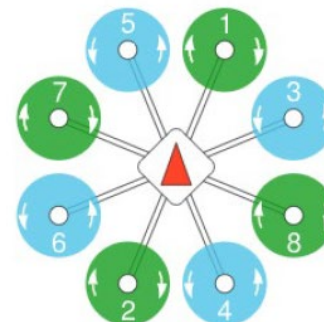
ModelParam\_uavType = 2: Tricopter Y-



ModelParam\_uavType = 5: Hexarotor x



ModelParam\_uavType = 8: Octorotor x





## 2. Other types of multicopters

```
%Initial condition
ModelInit_PosE=[0, 0, 0];
ModelInit_AngEuler=[0, 0, 0];
ModelInit_Inputs = [0 0 0 0 0 0 0 0];
```

### 2.9 Extend model to any other vehicles

- “MulticopterNoCtrl.slx” can be treated as a unified modeling template for all vehicle systems. Users can paste vehicle models into it by following the interface of input/output/parameters, then the DLL model file can be obtained for CopterSim.
- **Four essential inputs:** **1) inPWMS**, eight-dimensional (8-D) float vector signal from autopilot, ranges from 0 to 1; **2) TerrainZ**, 1-D ground height signal from CopterSim; **3) inSILFloats**, 20-D float vector; **4) inSILInts**, 8-D int vector. **Note:** inSILFloats and inSILInts are interfaces to communicate with other external programs (Simulink/Python/C++) through UDP for the requirement of fault injection, external sensor data generation, external control, etc.
- **Four essential outputs:** **1) MavHILSensor**, struct/bus signal, onboard sensor data; **2) MavVehicle3DInfo**, struct/bus signal, vehicle states for RflySim3D; **3) MavHILGPS**, struct/bus signal, GPS raw data; **4) MavHILStateQuat**, struct/bus signal, simulation states for Pixhawk.
- **Three essential parameters (see “init.m”):** **1) ModelInit\_PosE**, 3-D vector for vehicle initial position, see the “**6DOF**” module for details; **2) ModelInit\_AngEuler**, 3-D vector for vehicle initial attitude, see the “**6DOF**” module for details; **3) ModelInit\_Inputs**, 8-D vector for motor initial pwm, see the “**motor model**” module.

Model Info  
MulticopterNoCtrl

Matlab version is MATLAB R2017b and above.  
Coordinate frame is NED frame.  
This model simulates the behavior of a multicopters based on performance parameters and RPM inputs

Input 1: Eight-dimensional PWM inputs, from autopilot

1  
inPWMs

PWM inputs from autopilot (8-dimensional float vector, range from 0-1)

Input 2: one-dimensional ground height, from CopterSim

2  
TerrainZ

Current ground height at this location (1-dimensional float value, get positive value when downwards ground). This input is to tell the vehicle to stay at the surface of the ground.

4  
inSILFloats

20-dimensional external input float signals that can be transferred from other programs through UDP network.

3  
inSILInts

8-dimensional external input int signals that can be transferred from other programs through UDP network.

Note 1: inSILFloats and inSILInts make sure that CopterSim can receive data from other programs (Simulink/Python/C&C++) during real-time HIL or SIL simulation for fault injection, external control, generating data for external sensors, etc.

Input 3-4: 20-D float vector + 8-D int vector inputs, from external programs via UDP network

Motor dynamics model

Motor Model

Force and moment model

Force and Moment Model

Six-degree of freedom motion model

6DOF

Note2: There are three parameters essential for DLL model generation. They are "ModelInit\_inputs" (8D vector for input pwm signal initialization) in the "Motor Model" module, "ModelInit\_PosE" (3D vector for position initialization) and "ModelInit\_AngEuler" (3D vector for attitude initialization) in the "6DOF" module

Note: there are three essential parameter vectors, for CopterSim initializing vehicle position, attitude, motor pwm, etc.

Environment model

Environment Model

Output 1: onboard sensor data

MavLinkSensorData  
MavHILSensor

Output 2: Simulation state data

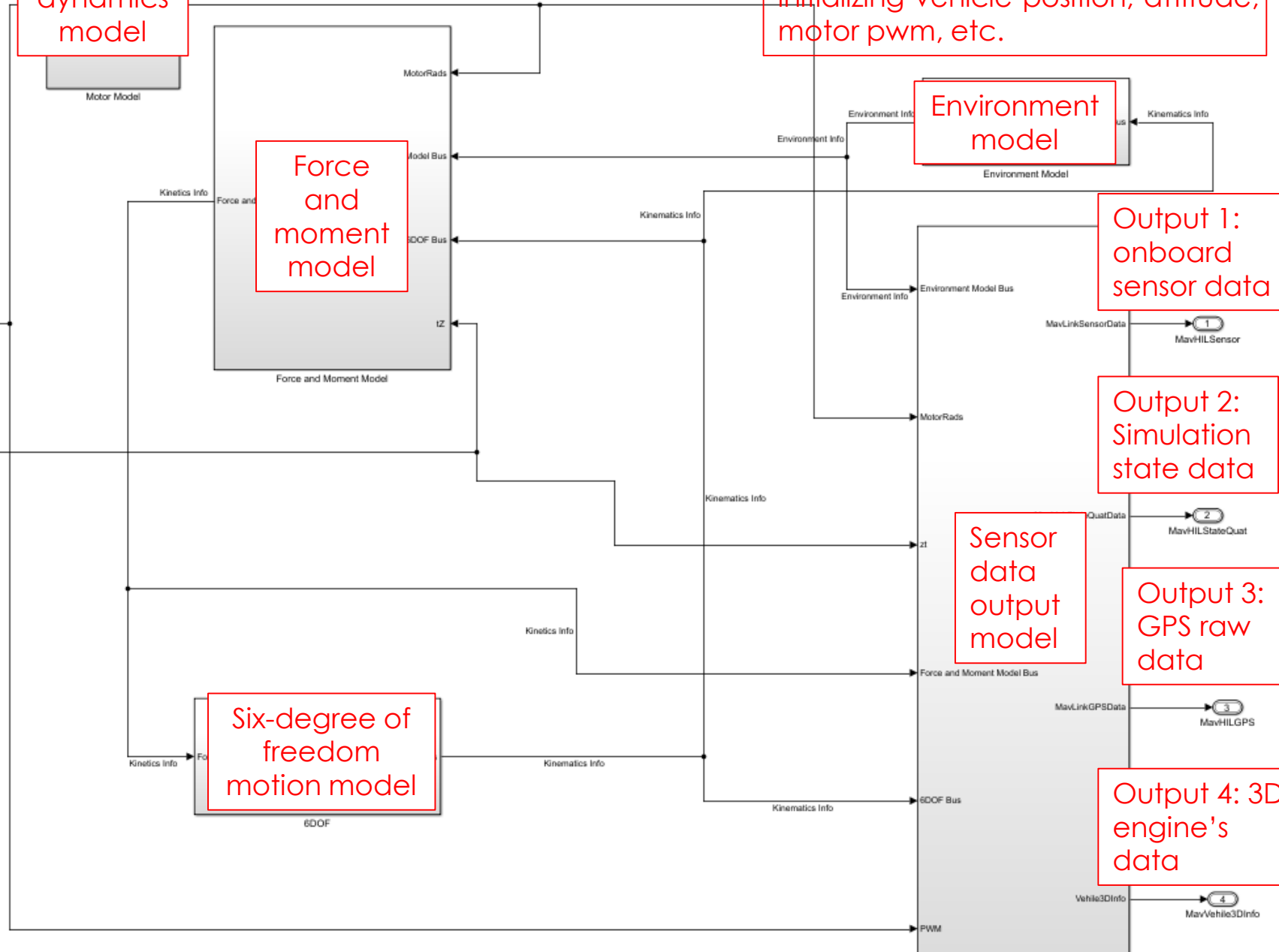
QuatData  
MavHILStateQuat

Output 3: GPS raw data

MavLinkGPSData  
MavHILGPS

Output 4: 3D engine's data

Vehicle3DInfo  
MavVehicle3DInfo





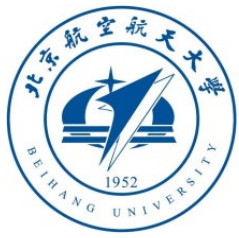
## 2. Other types of multicopters

Note: you can also use bat one-key script to start SIL/HIL simulations. Just modify the code to "set DLLModel=MulticopterNoCtrl"

### 2.10 CopterSim call DLL model file for simulations

- In the previous steps, a "**MulticopterNoCtrl.dll**" file is obtained, which is a quadcopter by default. Therefore, please change the Pixhawk airframe type to "**HIL Quadcopter X**" in QGC with steps similar to **Section 2.1**.
- Copy the DLL file "**MulticopterNoCtrl.dll**" to "**C:\PX4PSP\CopterSim\external\model**", where "**C:\PX4PSP**" should be modified to your installation path.
- Open CopterSim, and select "**MulticopterNoCtrl**" item in the drop-down list of the "**Use DLL Model**" option. Then, you can plug your Pixhawk in the computer and start HIL simulations

The screenshot shows the CopterSim software interface. On the left, a file explorer window displays the directory path (C:) > PX4PSP > CopterSim > external > model, containing a file named "MulticopterNoCtrl.dll". The main interface has several fields: "Vehicle ID" (1), "UDP Port" (20100), "Use DLL Model" (MulticopterNoCtrl), "Simulation Mode" (PX4\_HITL), "3D Scene Selection" (Grasslands), "Link" (checkbox), "Select Pixhawk Com" (Legacy FMU COM3), "UDP Mode" (UDP\_Full), and a "Start Simulation" button. A terminal window at the bottom shows the following output: "PX4: Command REQUEST\_AUTOPILOT\_VERSION ACCEPTED", "PX4: Command ARM/DISARM ACCEPTED", "PX4: EKF2 Estimator start initializing...", "PX4: GPS 3D fixed & EKF initialization finished.", "PX4: Enter Altitude Mode!", and "PX4: Enter Position Mode!".



# Content

---

1. Configuration of software & hardware
2. Other types of multicopters
3. User-defined fixed-wing aircraft
4. User-defined vehicle models
5. Summary

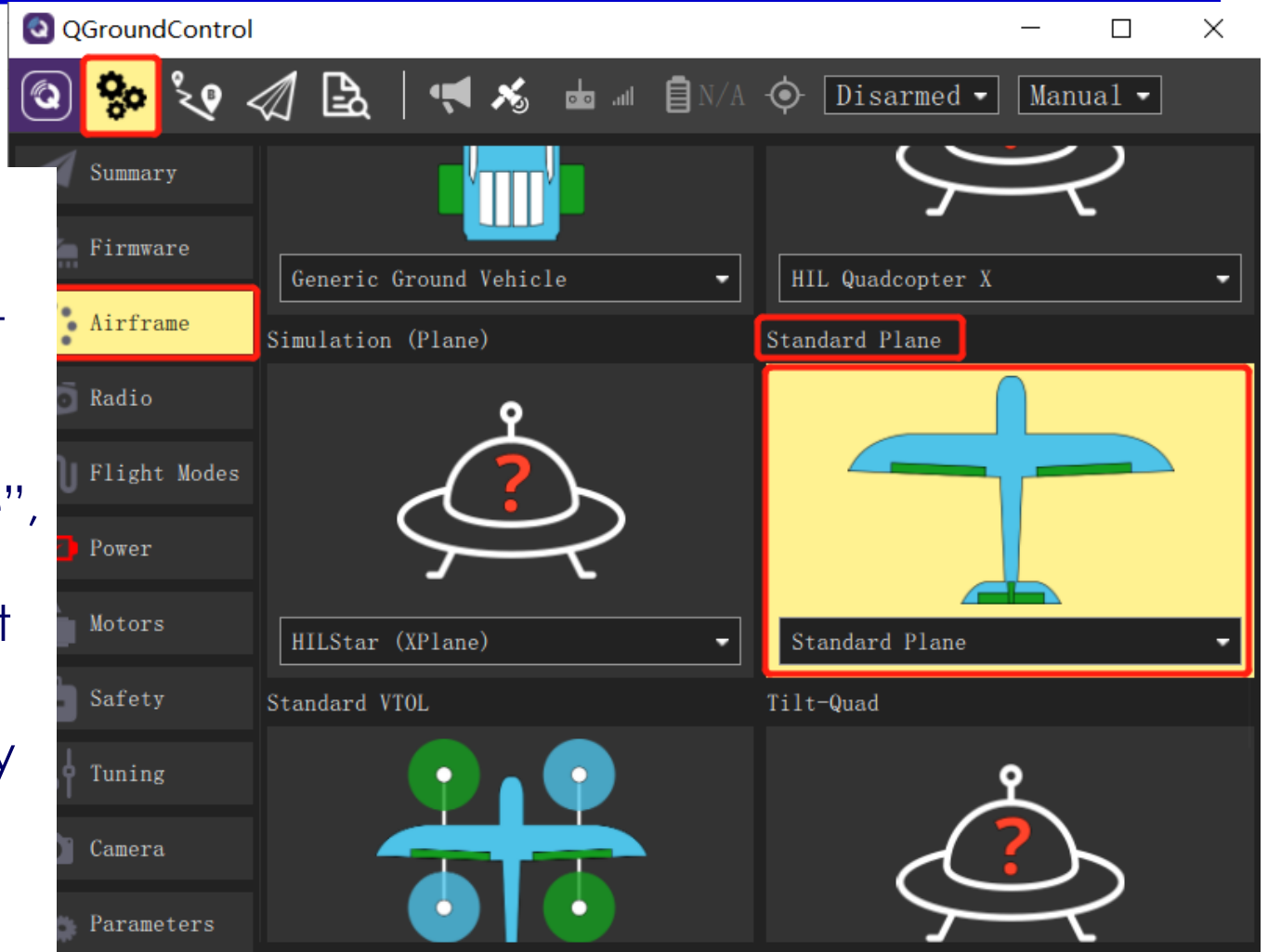
Path of demo source code of this lesson: "RflySimAPIs\OtherVehicle Types\AircraftModelCTRL"



### 3. User-defined fixed-wing aircraft

#### 3.1 Fixed-wing airframe selection for Pixhawk via QGC

- Start QGC and plug Pixhawk in the computer, and the Pixhawk will automatically connect to the QGC; at this time, as shown on the right, enter the “**Airframe**” tab, select the airframe as “**Standard Plane**” – “**Standard Plane**”, and then click on the “**Apply and Restart**” button, the autopilot will restart automatically at this time;
- After restarting, QGC will automatically look for the serial port and connect to Pixhawk. At this time, check each configuration page to ensure that everything is OK.



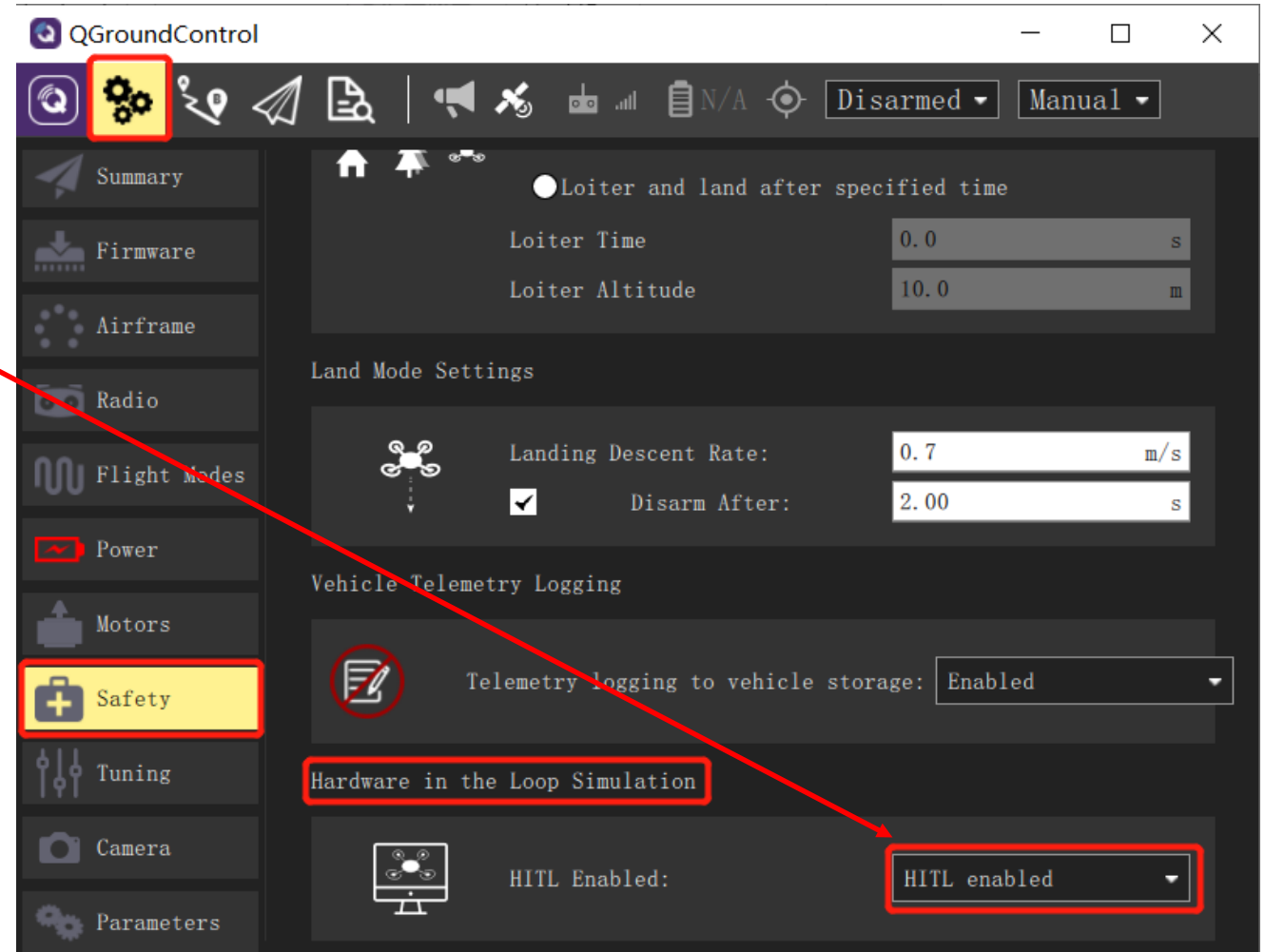




### 3. User-defined fixed-wing aircraft

#### 3.2 Set HIL mode for Pixhawk via QGC

- Enter “**Safety**” page, and find “**Hardware in the Loop Simulation**” item to select “**HITL enabled**” to enable HIL simulation mode for fixed-wing aircraft.
- After finishing the above steps, re-plug the Pixhawk on the computer, after QGC reconnects to Pixhawk, go to “**Radio**” and “**Flight Modes**” page to check whether the RC is correctly calibrated and the Flight modes are set as required.

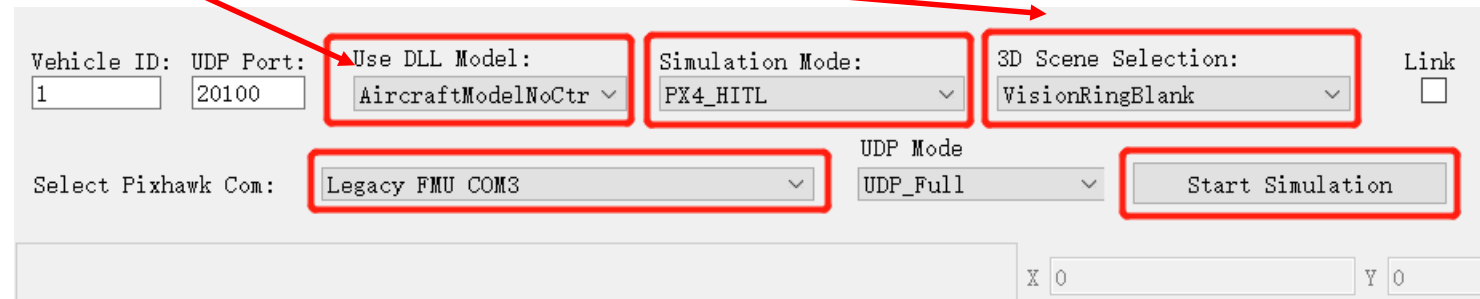
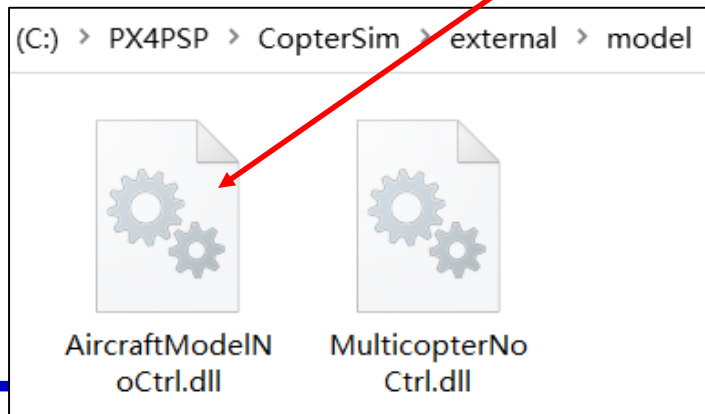




## 3. User-defined fixed-wing aircraft

### 3.3 Simulink model for fixed-wing aircraft

- Open MATLAB and enter the path “**RflySimAPIs\OtherVehicleTypes\AircraftModelCTRL**” to open the fixed-wing aircraft model demo “**AircraftModelNoCtrl.slx**”.
- Note: the first four pwm inputs in PX4 for fixed-wing aircraft are different from multicopter, the channels and ranges are 1) roll channel (**-1 to 1**); pitch channel (**-1 to 1**); throttle channel (**-1 to 1**); yaw/rudder channel (**-1 to 1**). The initial states of RC sticks are throttle at lowest position and pitch/roll/yaw at middle position, so the initial pwm vector for **ModelInit\_Inputs** is **[0,0,-1,0]**.
- Please follow the steps in **Section 2.5** to generate DLL model file for “**AircraftModelNoCtrl.slx**”, which yields “**AircraftModelNoCtrl.dll**”. Then, copy it to the folder “**CopterSim\external\model**” and select a grass flat map “**VisionKingBlank**” for fixed-wing takeoff and fly tests.

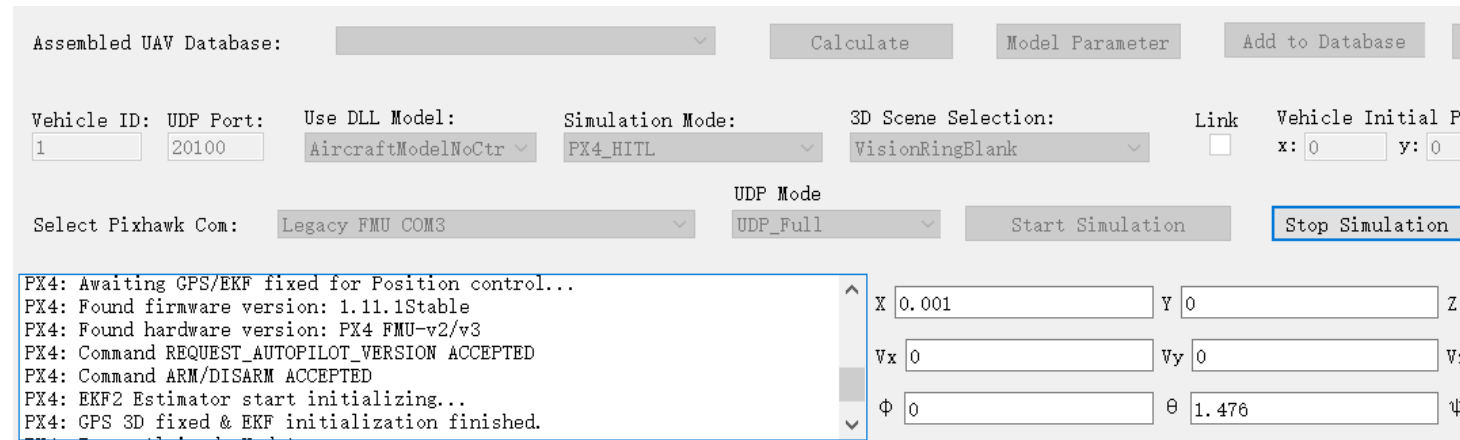
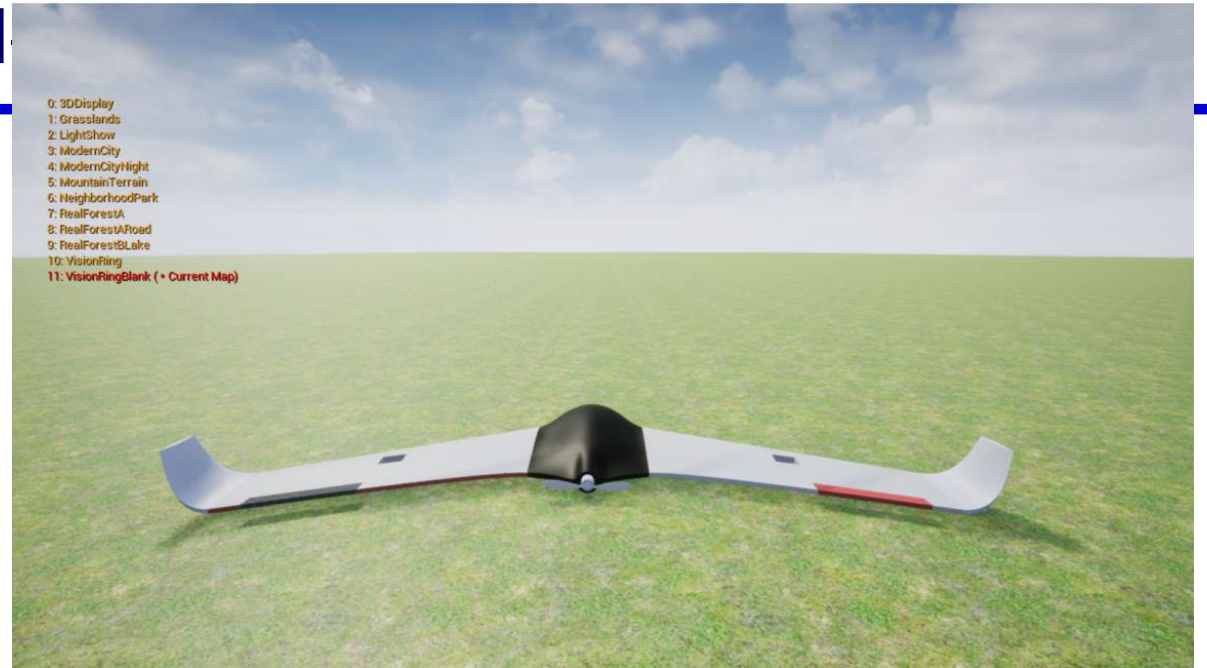




### 3. User-defined fixed

#### 3.4 Simulation effect for fixed-wing aircraft

- Open **RflySim3D**, and you can see a fly-wing aircraft on a grass land.
- Use RC transmitter to arm the vehicle, push the throttle stick to takeoff, and then control it to fly free.
- Open **QGC**, and you can observe the trajectory of the vehicle, or send commands to it.
- You can also use Simulink to design controller for fixed-wing aircraft and download it to Pixhawk to perform HIL simulation. Please try it yourself.



RflySim: How to simulate other types of vehicles, for example fixed-wing aircraft

Watch this video by clicking the following links:

YouTube: <https://youtu.be/gt7IIUx7JRC>

Youku: [https://v.youku.com/v\\_show/id\\_XNDcwNjA4MTQxMg==.html](https://v.youku.com/v_show/id_XNDcwNjA4MTQxMg==.html)

Vehicle Setup

- Summary
- Firmware
- Airframe**
- Radio
- Flight Modes
- Power
- Motors
- Safety
- Tuning
- Camera
- Parameters

Hexarotor Coaxial

Generic Hexarotor coaxial geometry | Generic Hexarotor x geometry | Steadidrone MAVRIK | Generic Octocopter + geometry | Generic 10" Octo coaxial geometry

Octorotor x | Plane A-Tail | Quadrotor + | Quadrotor H | Quadrotor Wide

Generic Octocopter X geometry | Applied Aeronautics Albatross | Generic 10" Quad + geometry | Reaper 500 Quad | Team Blacksheep Discovery

Quadrotor x | Simulation (Copter) | Simulation (Plane) | Standard Plane | Standard VTOL

Generic Quadplane VTOL

VTOL Duo Tailsitter | VTOL Quad Tailsitter | VTOL Tiltrotor



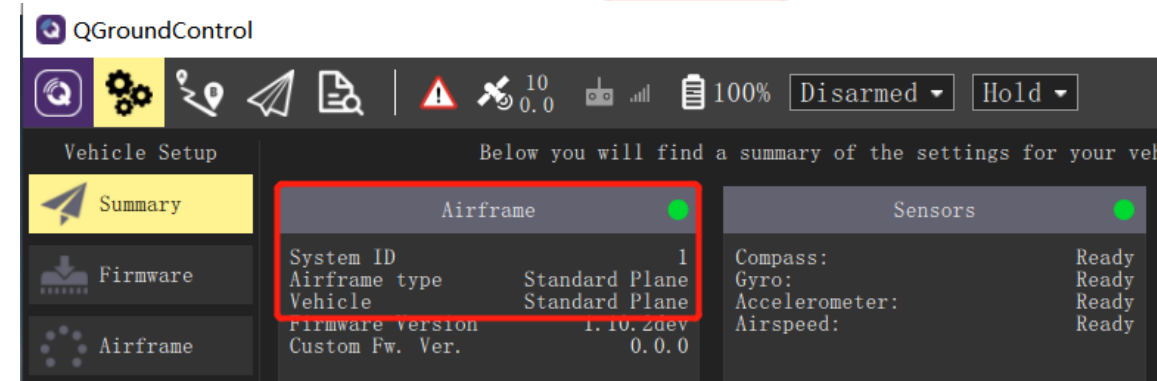
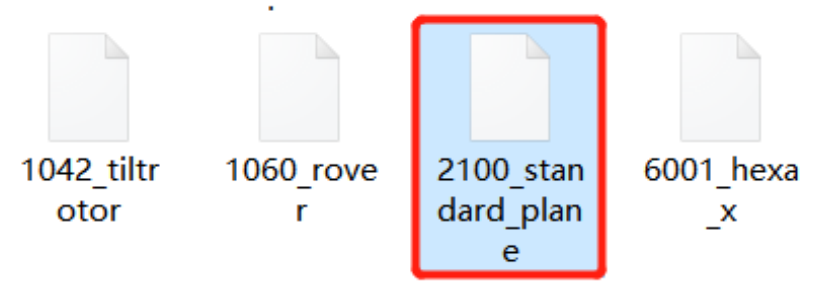


### 3. User-defined fixed-wing aircraft

#### 3.5 SIL simulation for fixed-wing aircraft

- Follow the steps in the above section to build aircraft mode in Simulink and generate DLL model to CopterSim
- Copy the airframe file “2100\_standard\_plane” from folder “Firmware\ROMFS\px4fmu\_common\init.d\airframes” to the folder “Firmware\ROMFS\px4fmu\_common\init.d-posix”
- Copy and save a SITLRun.bat file,
  - set DLLModel=AircraftModelNoCtrl
  - set PX4SITLFrame=standard\_plane
  - SET UE4\_MAP=OldFactory
  - SET /a ORIGIN\_POS\_X=-250
  - SET /a ORIGIN\_POS\_Y=-119
- Run the bat script and enter vehicle number (support swarm simulation) to start simulation

Firmware > ROMFS > px4fmu\_common > init.d-posix





# Content

---

1. Configuration of software & hardware
2. Other types of multicopters
3. User-defined fixed-wing aircraft
4. User-defined vehicle models
5. Summary

Path of demo source code of this lesson: "RflySimAPIs\OtherVehicle Types\AircraftMathworks"

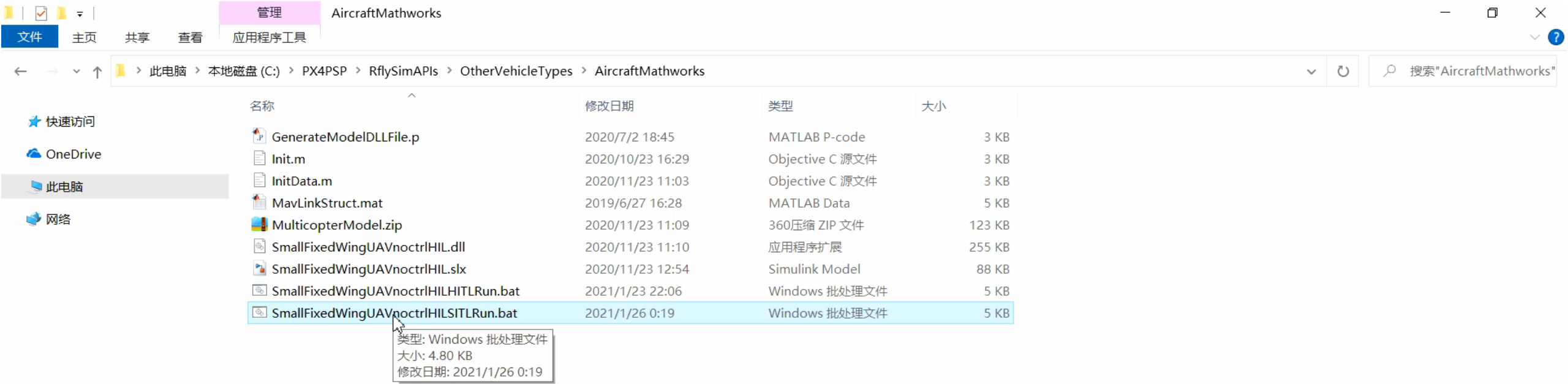


## 4. User-defined vehicle models

---

### 4.1 Import Simulink model for other vehicle types

- A demo in the folder “**RflySimAPIs \ OtherVehicleTypes \ AircraftMathworks**” is from a fixed-wing aircraft model from Mathworks Github page. The model file is “**SmallFixedWingUAVnoctrlHIL.slx**” which is realized by pasting body motion, aerodynamics, propulsion system models to our template file “**MulticopterNoCtrl.slx**”.
- The Mathworks’s official Github page URL is  
<https://github.com/mathworks/simulinkDroneReferenceApp>
- In this toolbox, you can also find attitude controller, path planning controller, etc. for fixed-wing aircraft. So you can follow the steps in Lesson 2 to import this controller in Simulink and download it to Pixhawk.
- Please try the above functions by yourself. Note: you can run “\*\*HITLRun.bat” or “\*\*SITLRun.bat” in this folder to start hardware/software-in-the-loop simulation, and use QGC to send mission to automatically takeoff, fly route, and land.



RflySim: How to start SITL simulation, and use QGC to control aircraft to takeoff and land automatically.

Watch this video by clicking the following links:

YouTube: <https://youtu.be/riEVvwZBE4Y>

Youku: [https://v.youku.com/v\\_show/id\\_XNTA4Njg3NzU0NA==.html](https://v.youku.com/v_show/id_XNTA4Njg3NzU0NA==.html)





## 4. User-defined vehicle models

---

### 4.2 How do external programs communicate with DLL model

- The demo is  
“**RflySimAPIs\OtherVehicleTypes\ExternalCtrlAPI\DllModelExtCtrlAPI.slx**”
- You can receive true states (the same data as to RflySim3D) from CopterSim by using UDP port **30101**.
- You can send “inSILFloats” and “inSILInts” data to DLL model by using UDP port **30100**.
- You can also send any MAVLink message to Pixhawk by using UDP port **30100**.
- The above API allows you to inject faults to model, or send new sensor data (optical flow, range finder, ADC, etc.) to Pixhawk.



## 4. User-defined vehicle models

---

More demos are coming soon ...



---

# Thanks