Multicopter Design and Control Practice
——A Series Experiments Based on MATLAB and Pixhawk

Lesson 02 Experimental Platform Configuration

Dr. Xunhua Dai, dai@buaa.edu.cn

School of Automation Science and Electrical Engineering,

Beihang University, Beijing 100191, China.

Outline

1. Overall Introduction
2. Software Package Installation
3. Hardware Platform Configuration
4. Summary

Overall Introduction

The experimental platform can be divided into two parts: the **hardware platform** and the **software platform**. This lesson shall introduce the basic components of each platform and present the deployment procedure in detail.

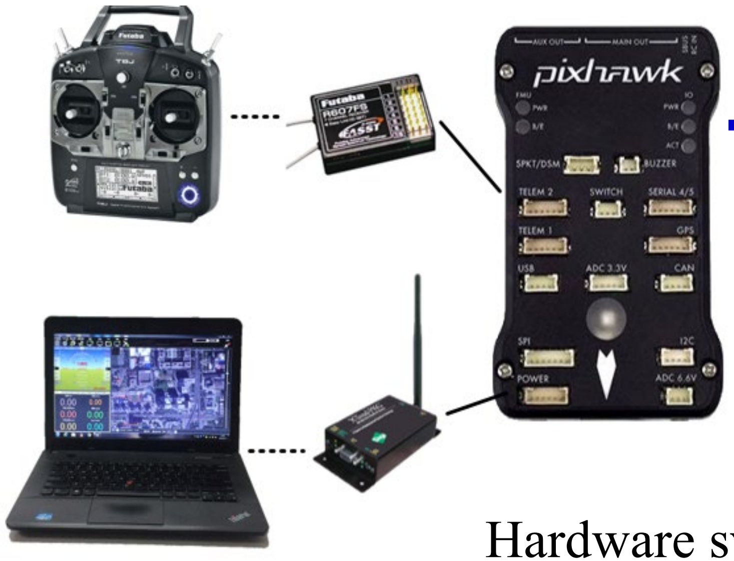
Noteworthy: this lesson is mainly for independent readers or experimental course teachers who need to deploy a code generation environment and prepare a platform for practical flight experiments. If the experimental platform has already been configured, readers can skip this chapter and directly start the experimental courses.

Overall Introduction

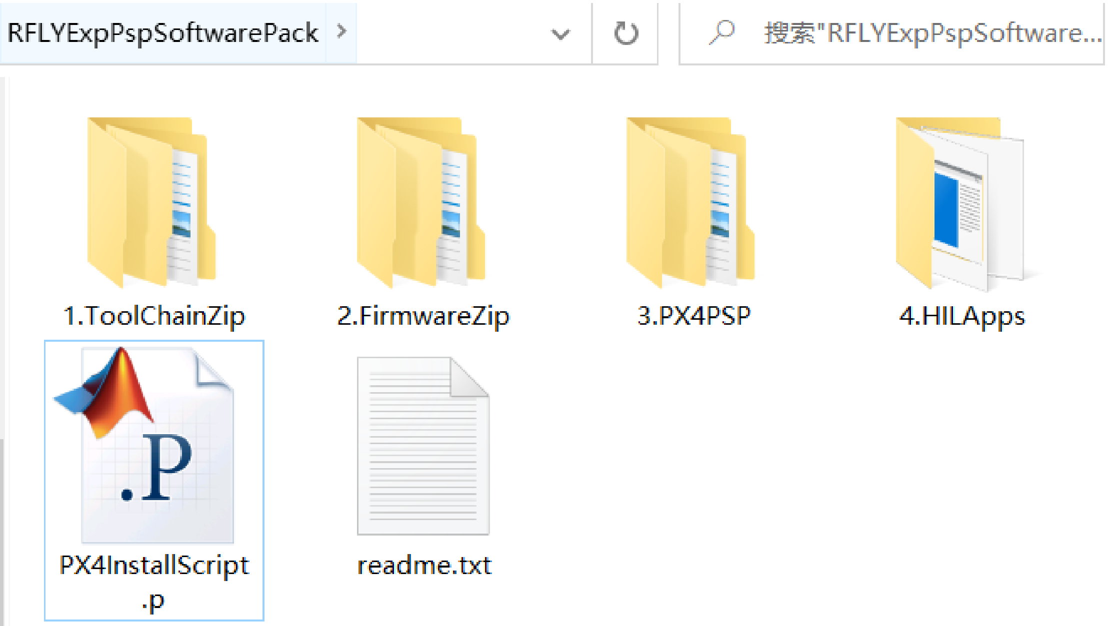
□ Hardware Platform

The hardware platform and the software platform are presented on the right. The basic configuration requirements are:

- Operating System (OS)： Windows 7 ~ 10 x64 system
- Processor: Intel i5 series or above
- Memory: 8G or above
- Graphics: discrete graphics, memory 2G or more
- Storage: 30G available space (solid-state drives are recommended)
- Interface: at least one USB Type-A
- Monitor: screen resolution 1080P or above



Hardware system

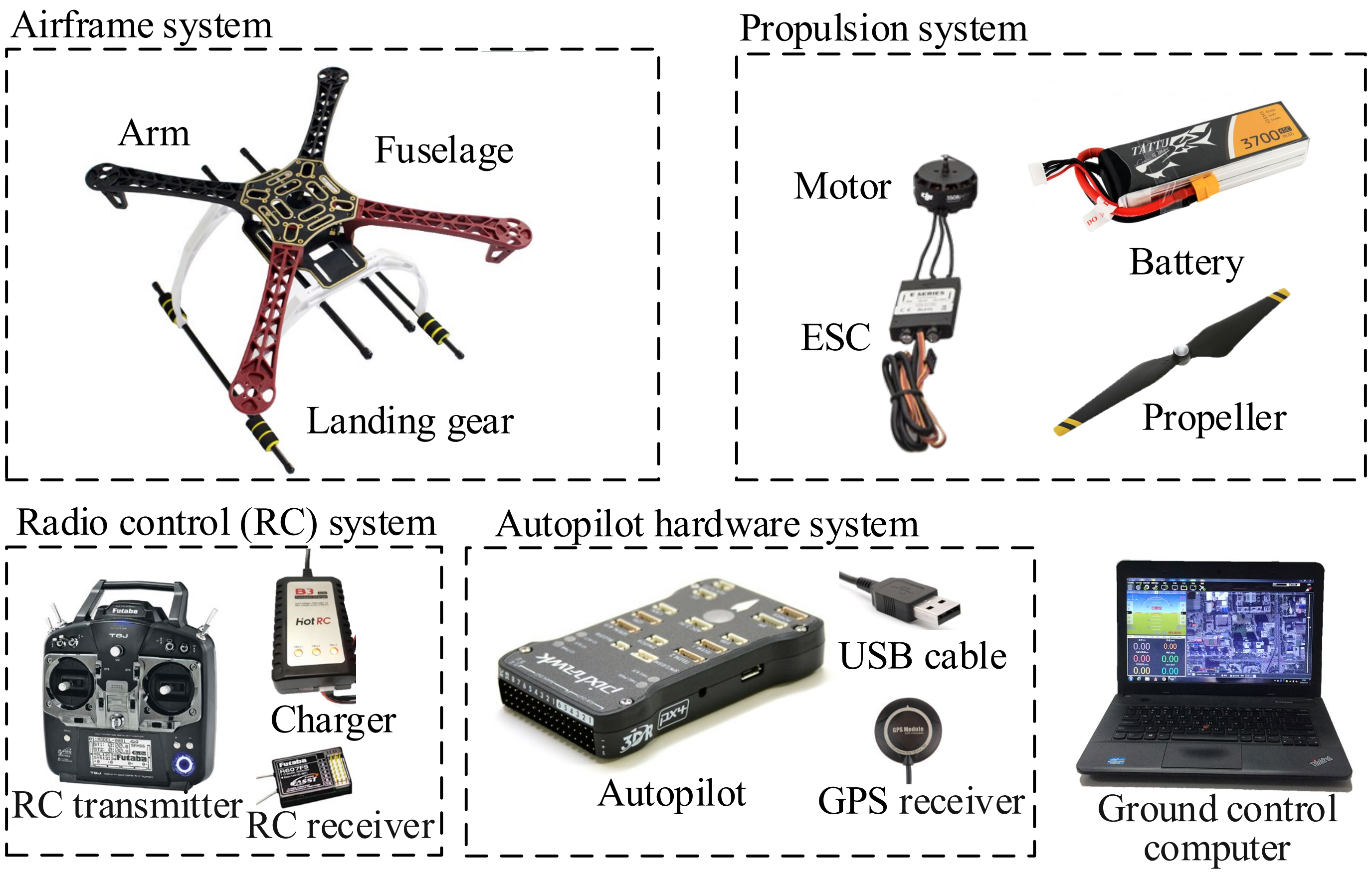


Installation files

Overall Introduction

Hardware Platform

Because all control algorithms are eventually deployed in a real aerial vehicle to perform flight tests, a hardware platform must be prepared for the basic flight test requirements. As shown in the right picture, the experimental hardware platform recommended in this book is composed of **five main parts**.



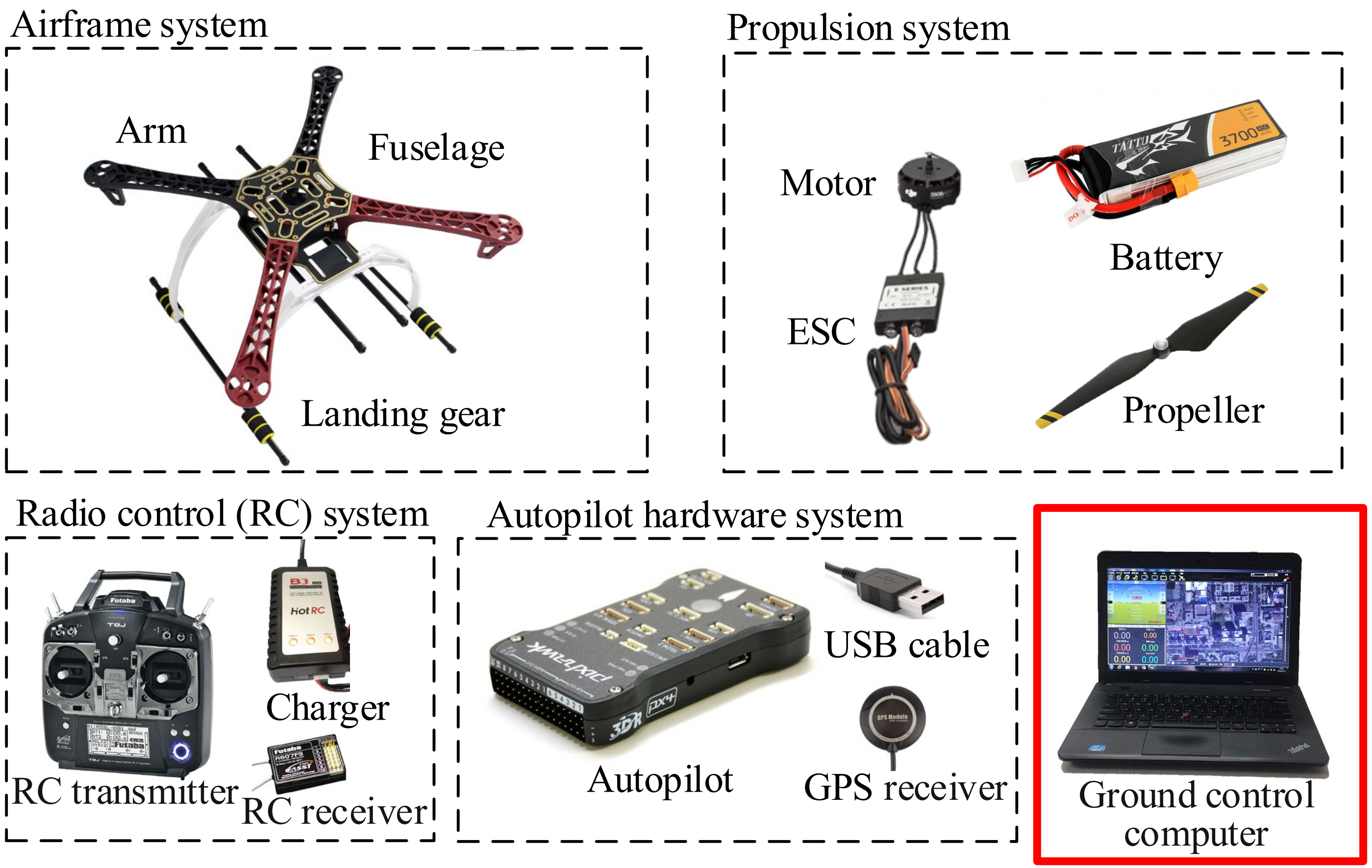
Overall Introduction

☐ Hardware Platform

(1) Ground Computer

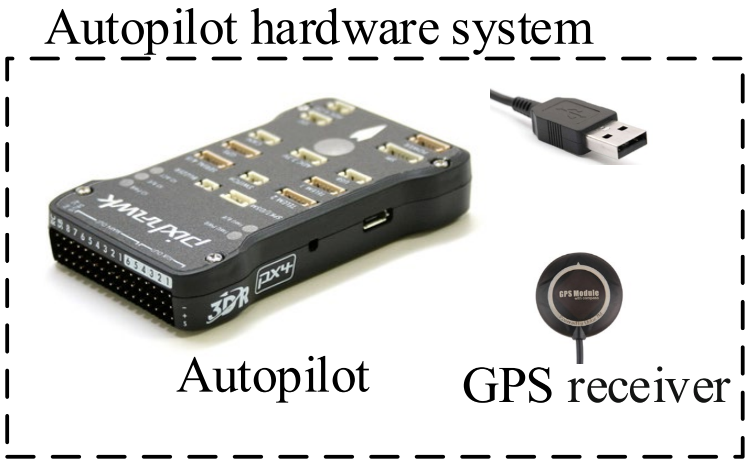
It is a high-performance Personal Computer (PC) with an operating system that performs two main tasks.

- Providing the software operating environment for the simulation software tools to perform functions such as control algorithm development, SIL simulation, automatic code generation, and HIL simulation;
- Working as a ground control station in outdoor flight tests to achieve functions such as sensor calibration, parameter tuning, real-time control, and communication.



Overall Introduction

◻ Hardware Platform



(2) Autopilot System (also called flight control system)

- As the operating platform of control algorithms, it has many sensors and powerful computing capability to estimate flight states and calculate the control signals for the propulsion system to realize the flight control of multicopters.
- This lesson select a widely used open-source autopilot — Pixhawk as the development and experimental autopilot system.
- Pixhawk is an independent open-hardware project that aims to provide standard readily-available, high-quality, and low-cost autopilot hardware for education, amateurs, and developers.
- For different flight mission, performance, and cost requirements, the Pixhawk provides a series of autopilot hardware products that highly promote the development of multicopters.

Overall Introduction

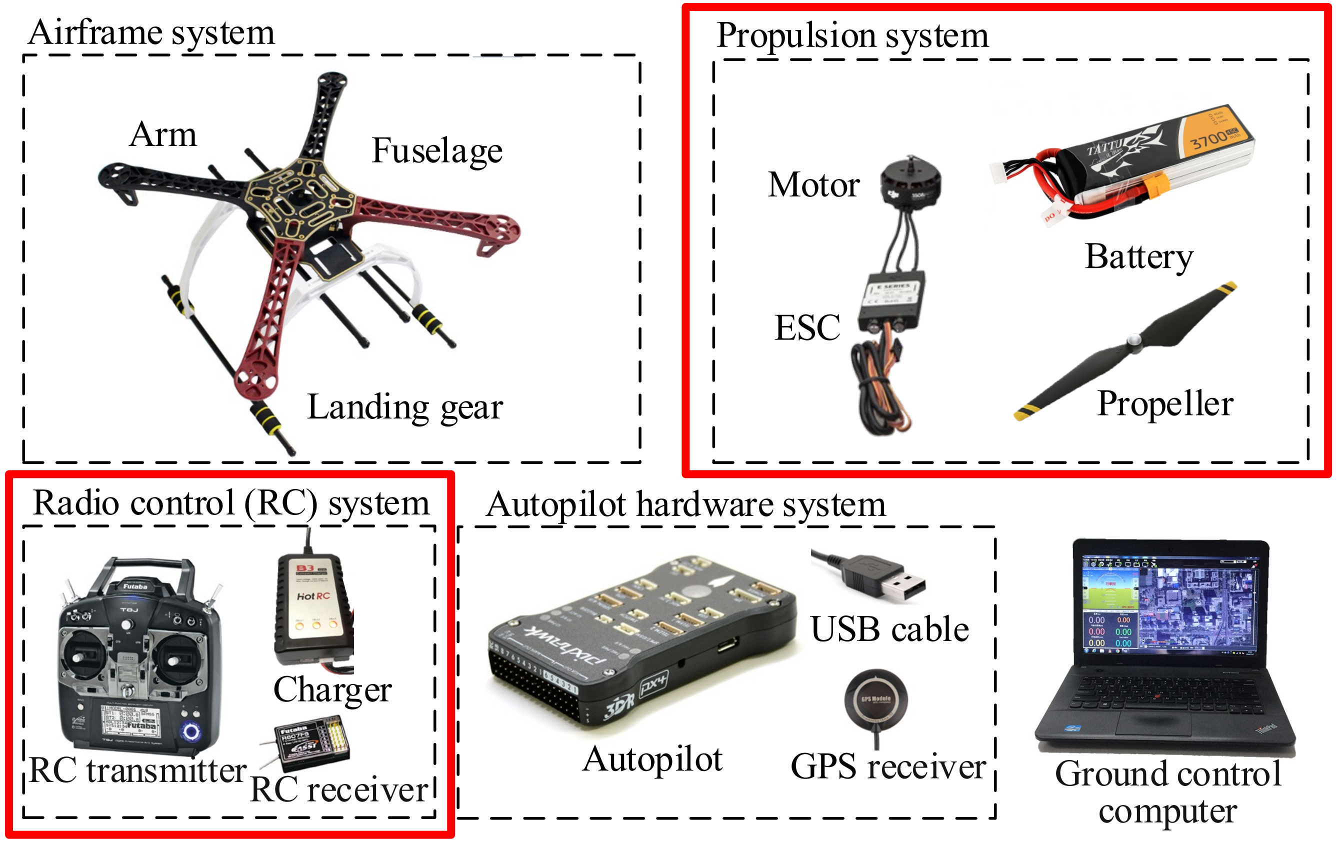
□ Hardware Platform

(3) Radio Control (RC) system

It includes an RC transmitter, an RC receiver, and a battery charger. The RC system is used to send control commands from the pilot on the ground to the autopilot system to realize remote flight control.

(4) Propulsion system

It includes a battery pack and several propellers, Electronic Speed Controllers (ESCs), and motors. The propulsion system is used to receive the Pulse Width Modulation (PWM) control signals from the autopilot system, and control the movement of a multicopter with thrust and torque generated by the rotation of the propellers and motors.

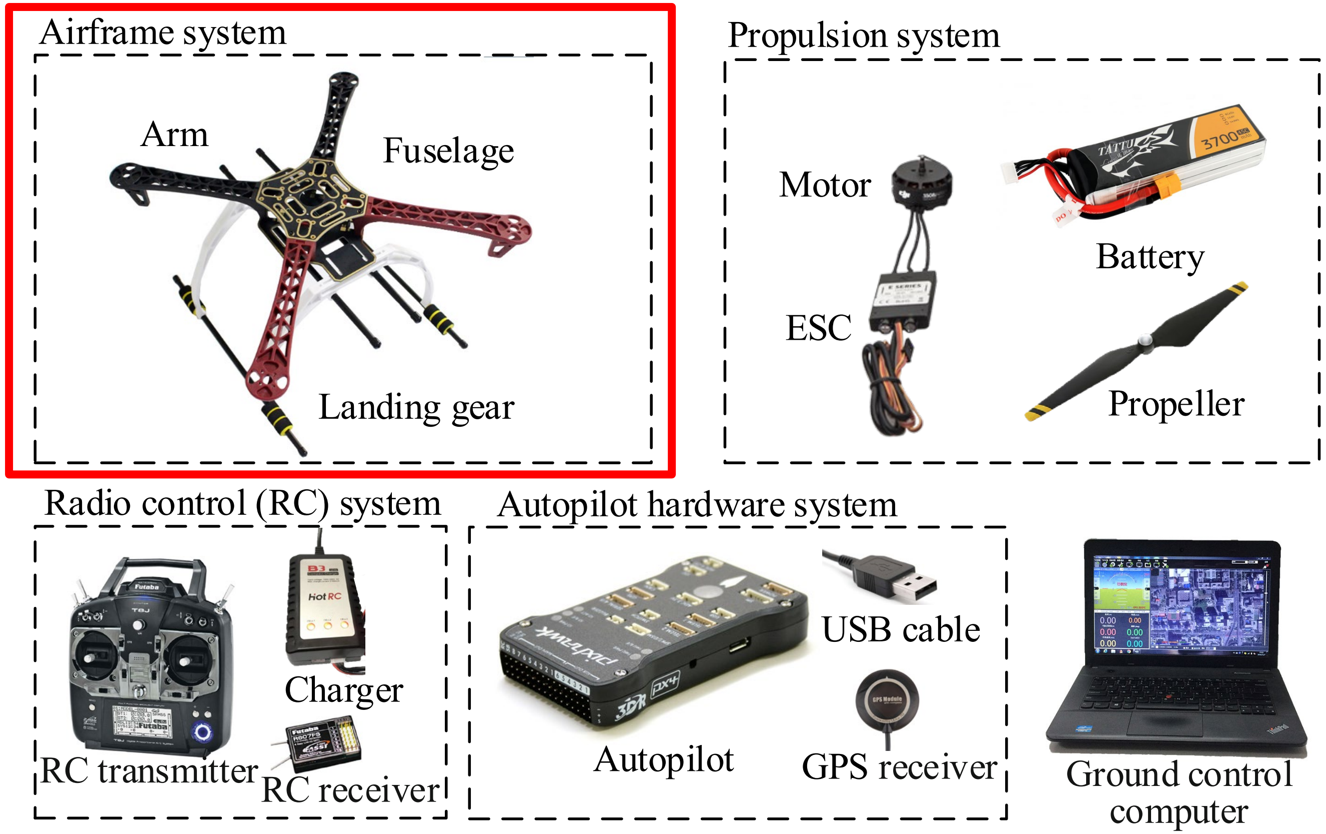


Overall Introduction

Hardware Platform

(5) Airframe system

It includes a fuselage, landing gear, and several arms. The airframe is used to support the propulsion system and the autopilot system and carry a payload; thus, it is required to have excellent aerodynamic performance and structural strength to ensure that flight missions are successfully and reliably accomplished.



Overall Introduction

Software Platform

- This platform supports all versions above MATLAB 2017b, the highest version is currently MATLAB 2020a
- The recommended installation version is MATLAB 2017b
- This lesson do not provide MATLAB installation package or instructions. Please install it by yourself.

It is recommended to install all toolboxes. If partial installation is selected, the toolboxes necessary for this experiment include:

MATLAB/Simulink
Control System Toolbox
Curve Fitting Toolbox
Aerospace Blockset
Aerospace Toolbox
MATLAB Coder
Simulink Coder
Stateflow

Overall Introduction

Software Platform

This experimental platform relies on many software tools to realize controller design, code generation, autopilot code compilation, HIL simulation, and other functions. The Simulation Software Package published along with this book has a one-click installation script. Readers can click the script to finish all the installation and configuration process of the required software environment. The MATLAB/Simulink and the Simulation Software Package comprise the software platform, which contains the following.

Overall Introduction

Software Platform

(1) Pixhawk Support Package (PSP) Toolbox

It is a Simulink toolbox officially released by Mathworks for controller design, code generation, and firmware upload of the Pixhawk autopilot. We have made some updates and optimizations based on the official PSP toolbox to ensure compatibility with the latest Pixhawk and MATLAB versions.

Overall Introduction

Software Platform

(2) FlightGear — Flight Simulator

It is a popular open-source flight simulator that can be used to easily observe the flight states of a simulated aerial vehicle in Simulink by receiving flight data from Simulink via a User Datagram Protocol (UDP) interface.

(3) PX4 Software — Source Code

PX4 is an open-source flight control software system that runs on the Pixhawk hardware platform. The Pixhawk hardware + PX4 software constitutes an integrated autopilot system, which is one of the most widely-used autopilot systems for aerial vehicles.

Overall Introduction

◻ Software Platform

(4) PX4 Toolchain — Compiling Environment

It is used to compile the PX4 source code along with the controller algorithms generated by the PSP toolbox into a ".px4" format firmware file. Then, the firmware file is uploaded and burnt into the Pixhawk autopilot hardware (similar to the process of reinstalling an operating system on a PC). The control algorithm generated by the PSP toolbox will automatically run after Pixhawk restarts.

(5) Eclipse C/C++ — Integrated Development Environment (IDE)

It is used to read and modify the PX4 source code. Eclipse C/C++ is a compact C/C++ IDE with functions similar to those of Microsoft Visual Studio.

Overall Introduction

Software Platform

(6) QGroundControl (QGC) — Ground Control Station

It is used to perform the pre-flight tasks (e.g., sensor calibration and parameter tuning) for the Pixhawk autopilot before the multicopter takes off. The QGC is also used to receive the flight states and send the control commands of the multicopter through wireless radio telemetry during flight tests.

(7) CopterSim—Real-Time Motion Simulation Software

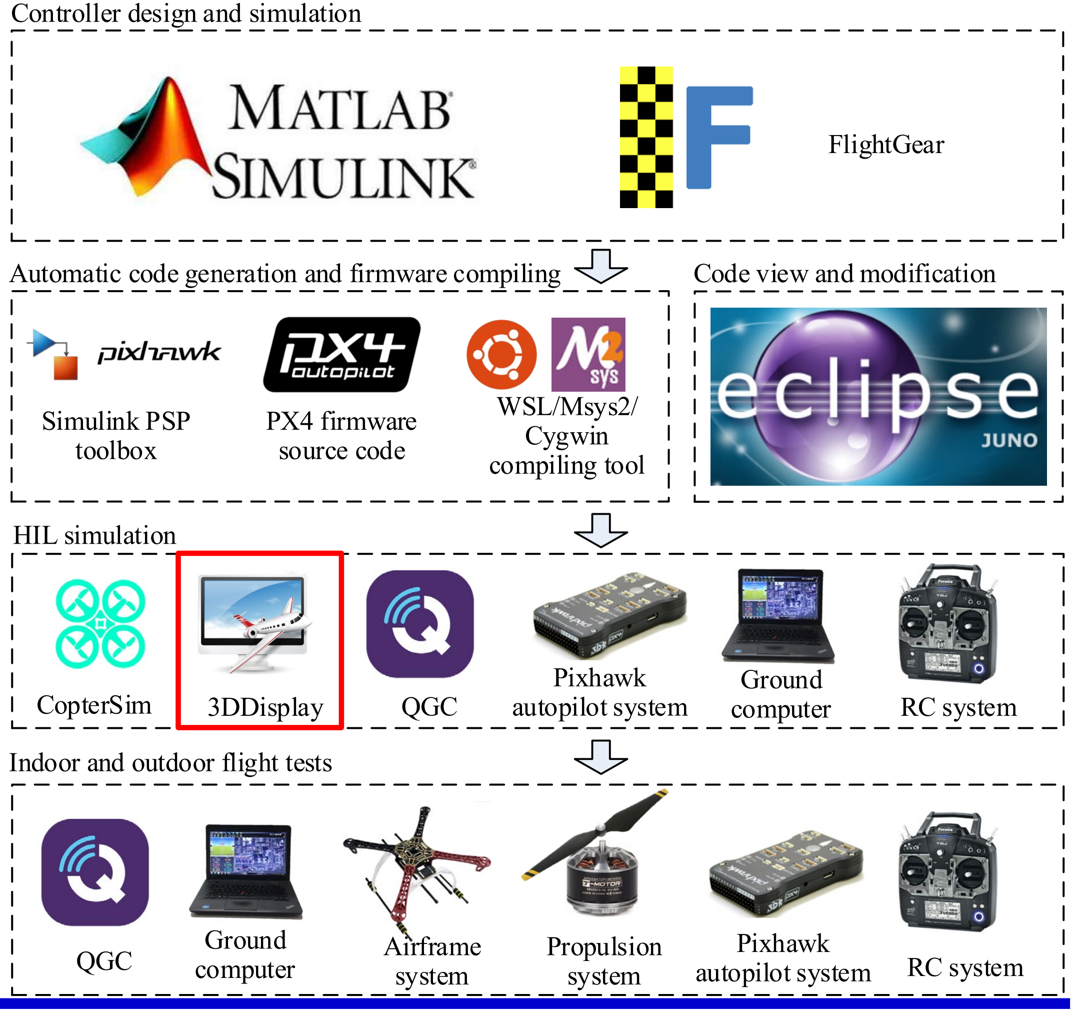
It is a real-time motion simulation software developed for the Pixhawk/PX4 autopilot system. Readers can configure multicopter models in CopterSim, and connect it to the Pixhawk autopilot via the USB serial port to achieve indoor HIL simulations.

Overall Introduction

☐ Software Platform

(8) 3DDisplay — 3D Visual Display Software

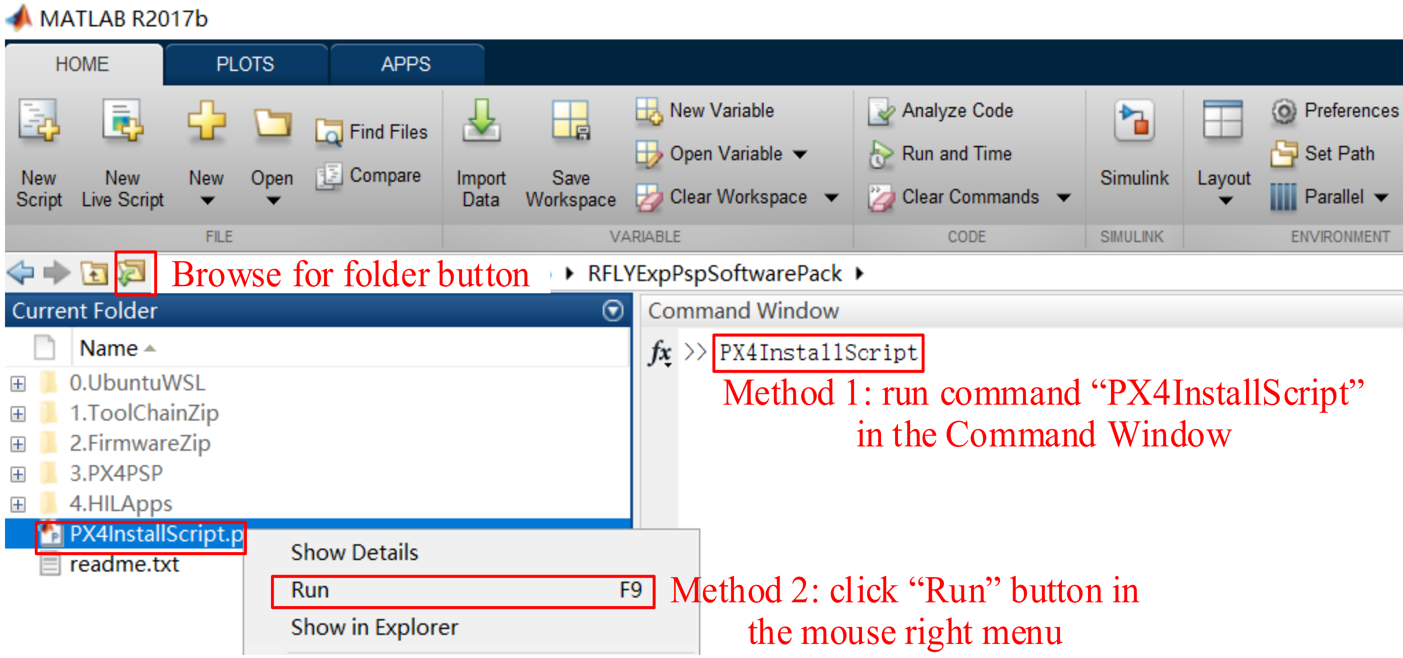
It is a real-time 3D visual display software. It receives the flight data of CopterSim through UDP to display the attitude and position of a multicopter in real-time. CopterSim and 3DDisplay together constitute an integrated HIL simulation platform. The distributed independent operation mechanism of CopterSim and 3DDisplay provides future compatibility for swarm simulations.



Software Package Installation

Installation Steps

1. Obtain the installation ISO image file "RFLYExpPspSoftwarePack.iso" from our website https://rflysim.com/download and unzip it or mount it to a virtual drive folder。
2. Open MATLAB and click the "Browse Folder" button as shown in the figure to locate and enter the "current path" to the "RFLYExpPspSoftwarePack" directory.
3. Select the "PX4InstallScript.p" file, right-click, and click the "Run" option in the pop-up context menu. Note: Another method is to enter the command "PX4InstallScript" in the "Command Window" and press the Enter key.

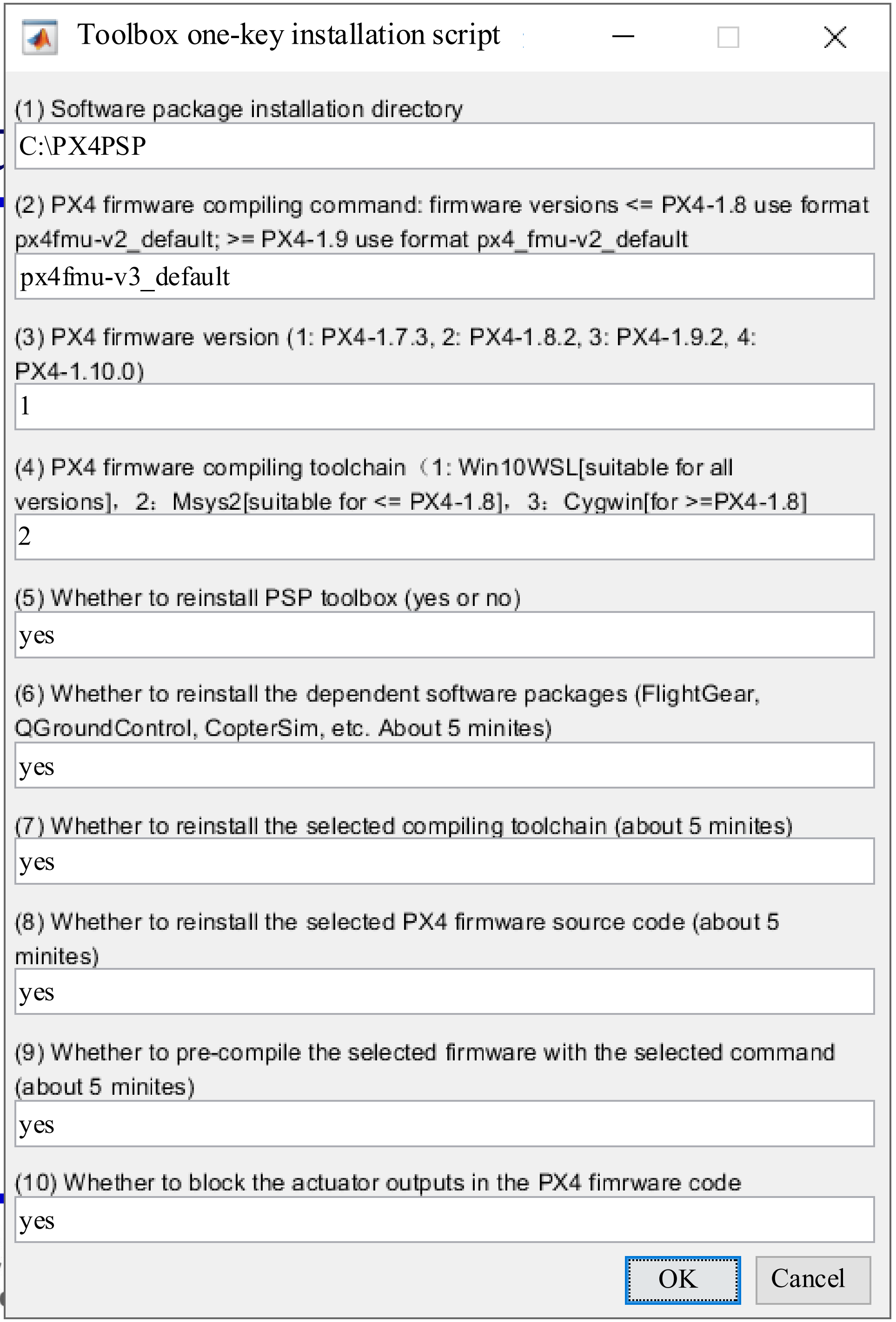


Software Package Installat

Installation Steps

4. In the pop-up configuration window shown in the picture, select the required configuration according to the actual hardware and software requirements (the default configuration is recommended for beginners, where the autopilot hardware is Pixhawk 1 [2MB Flash version, board version 2.4.6 and above] with corresponding compiling command px4fmu-v3_default, the PX4 firmware version is 1.7.3, and the installation directory is C disk, which may occupy around 6G storage), and click the “OK” button.
5. Wait patiently for the package to be successfully installed and deployed, which may take around 30 minutes.

Note: The RflySim Basic Free version only provides PX4 1.7&1.8 firmware + Msys2 toolchain, which is enough for all experiments in this course. You can also contact rflysim@163.com to inquiry about RflySim advanced versions if there is demand.

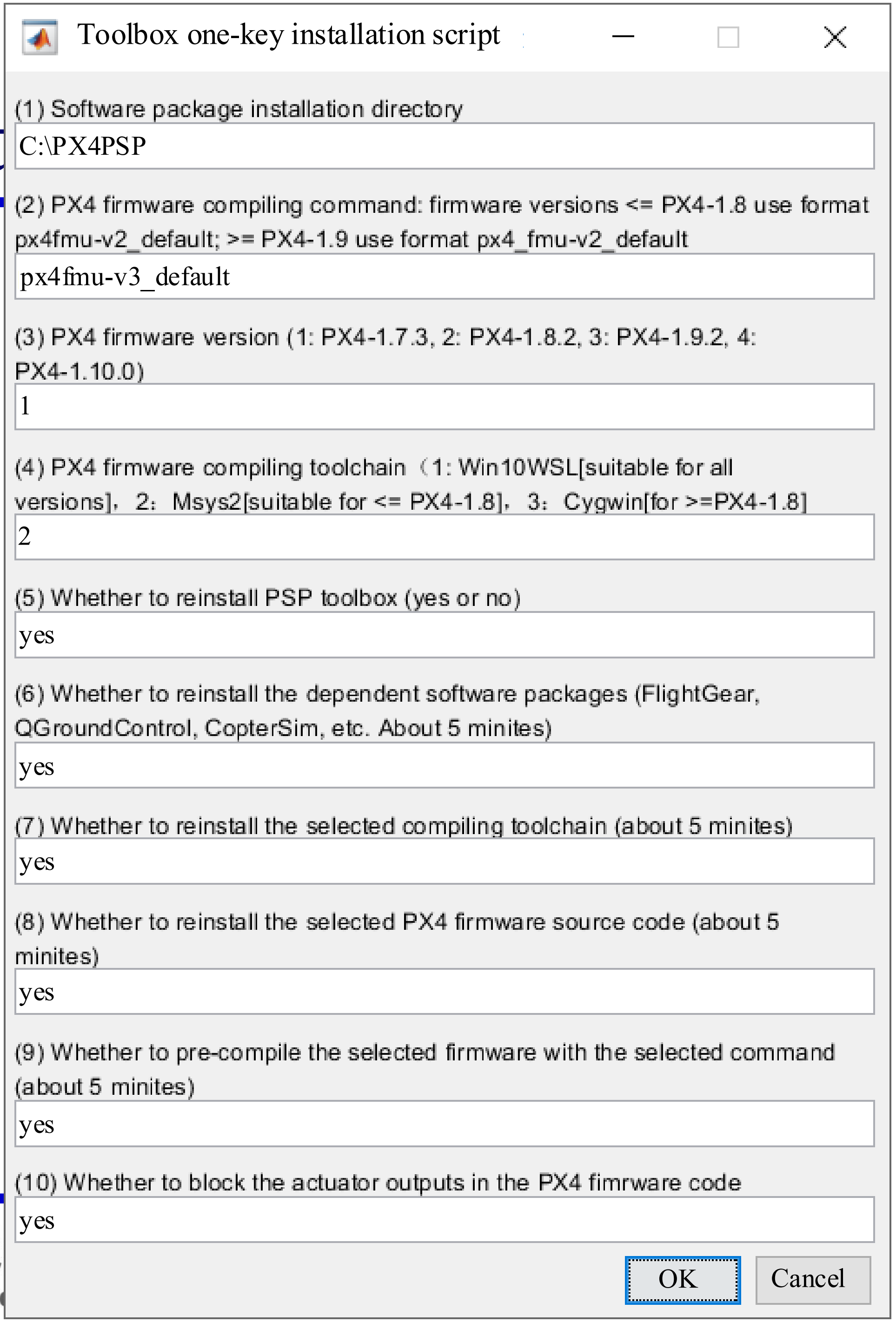


Software Package Installat

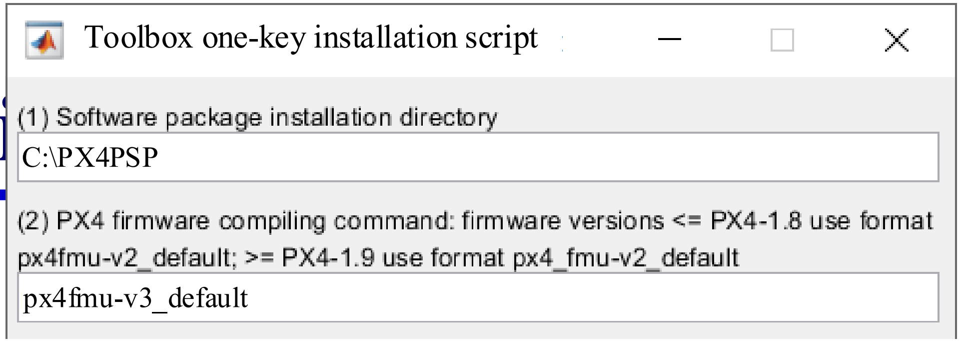
Installation Steps

Noteworthy:

1. Antivirus software may prevent this script from generating desktop shortcuts. If the script prompts that the shortcut generation has failed, please close the antivirus software (Windows 10 should also turn off the "Real-time protection" option in the Settings page) and manually click the “GenerateShortcutCMD.bat” script in the installation directory (the default directory is C:\PX4PSP) to automatically generate all the software shortcuts.
2. 2) If the readers want to change the firmware configurations or restore the compiling environment, just run the “PX4InstallScript” command again and select the required options.



Software Package Installati



☐ Advanced Settings

1. Software package installation directory. All dependent files on the software package are installed in this directory, which requires around 6G storage. The default installation directory is “C: \PX4PSP”. If the C disk space is not sufficient, the readers should choose a directory in other disks; the directory name must be correct with only letter and number to prevent compilation failures.
2. The default compiling command for PX4 is “px4fmu-v3_default”. By selecting this compiling command, the compiling toolchain is automatically called to compile the PX4 source code to a firmware file “px4fmu-v3_default.px4” after the PSP generates the controller code. Then, the file “.px4” is uploaded and burned to the supported hardware to realize the deployment of the control algorithms. Different Pixhawk hardware products must select different PX4 firmware compiling commands. Next picture shows some Pixhawk hardware products, where “px4fmu-v3_default” can be used for three popular products: Pixhawk 1 (2MB Flash version), mRo Pixhawk, and Cube (Pixhawk 2). The command “px4mu-v2_default” corresponds to the most famous Pixhawk 1 (1MB Flash). PX4 also supports other hardware (for example, Intel Aero, Crazyfile, and so on). The corresponding compiling commands are listed in the next page.

Note: after installing the toolbox, if you want to change compiling command (e.g., px4-fmu-v5_default) for different Pixhawk hardware boards, another feasible method is to input command in MATLAB: PX4CMD('px4-fmu-v5_default'), or use command : PX4CMD px4-fmu-v5_default

Software Package Insta

(2) PX4 firmware compiling command: firmware versions <= PX4-1.8 use format px4fmu-v2_default; >= PX4-1.9 use format px4_fmu-v2_default

px4fmu-v3_default

☐ Advanced Settings

- Pixhawk 1: px4fmu-v2_default
- Pixhawk 1 (2MB flash): px4fmu-v3_default
- Pixhawk 4: px4fmu-v5_default
- Pixracer: px4fmu-v4_default
- Pixhawk 3 Pro: px4fmu-v4pro_default
- Pixhawk Mini: px4fmu-v3_default
- Pixhawk 2: px4fmu-v3_default
- mRo Pixhawk: px4fmu-v3_default
- HKPilot32: px4fmu-v2_default
- Pixfalcon: px4fmu-v2_default
- Dropix: px4fmu-v2_default
- MindPX/MindRacer: mindpx-v2_default
- mRo X-2.1: auav-x21_default
- Crazyflie 2.0: crazyflie_default
- Intel® Aero Ready to Fly Drone: aerofc-v1_default

Pixhawk 1 (fmu-v2)
2MB flash Version (fmu-v3)

mRo Pixhawk
(fmu-v3)

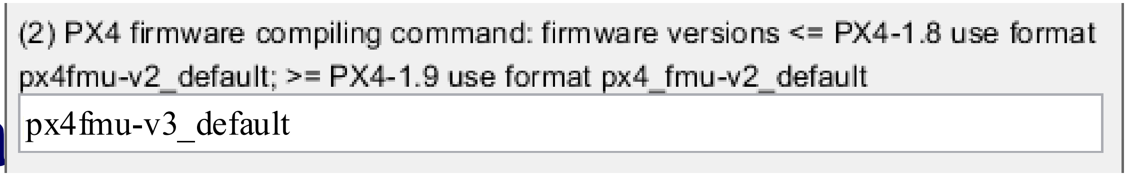
Cube/Pixhawk 2
(fmu-v3)

Pixhawk 3 Pro
(fmu-v4)

Pixhawk 4
(fmu-v5)

Pixhawk 4 Mini
(fmu-v5)

Software Package Insta



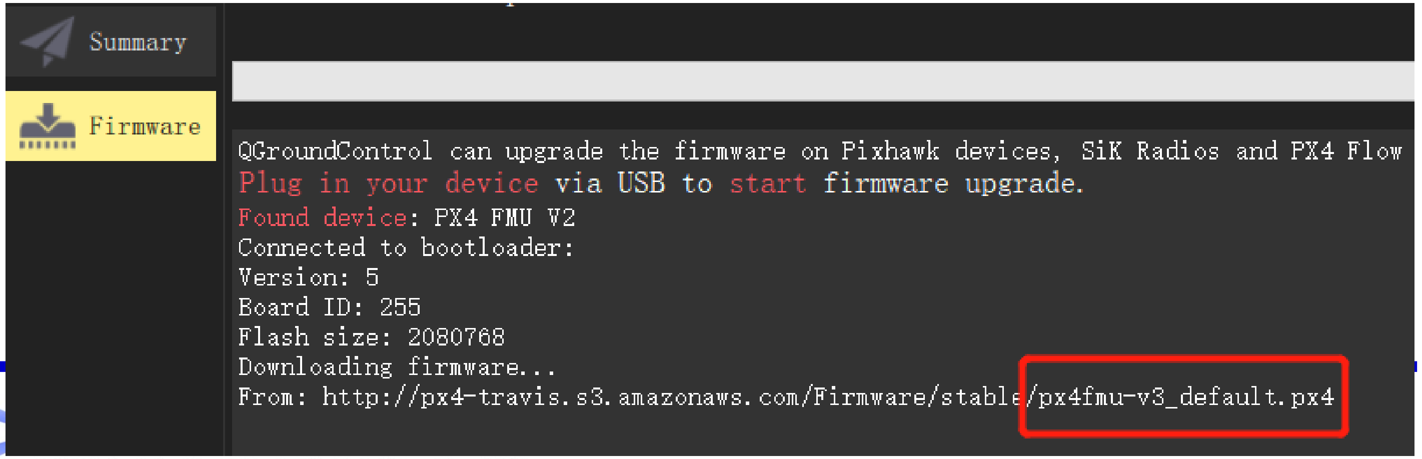
☐ Advanced Settings

Method to find out the desired compiling command for your Pixhawk:

1) Open QGroundControl (QGC) and enter the "Setting" (Gear icon) – "Firmware" Page;

2) Connect Pixhawk with a USB cable, and the QGC will turn to the state in the right figure, then click "OK" to update;

3) QGC will auto download the desired .px4 firmware, so the compiling command can be found in the download link. For example, px4fmu-v3_default is obtained for Pixhawk 1 (2Mb Flash).



Software Package Inst

□ Advanced Settings

3. PX4 firmware version. The version of the PX4 source code is updated constantly, and the latest firmware version was 1.11 when this book was written. As the firmware version is upgraded, new features may be introduced, and more new products will be supported, but the compatibility with some old autopilot hardware will be affected. Because the Pixhawk 1 hardware selected in this book is an old Pixhawk product with LED for better experimental observation effect, the older PX4 firmware version 1.7.3 was selected with the compiling command “px4fmu-v3_default” to achieve better-using effect.
4. PX4 firmware compiling toolchain. Because the compilation of PX4 source code depends on the Linux compiling environment, the software package provides three sets of compiling toolchains to realize the simulation of the Linux compiling environment under the Windows environment.

- Win10WSL based on the Windows Subsystem for Linux (WSL);
- (b) Msys2Toolchain based on toolchain;
- (c) CygwinToolchain based on the toolchain.

Software Package Instal

◻ Advanced Settings

Example 1: if you are using the recommended Pixhawk 1 (2M flash, px4fmu-v3 firmware):

- Configure the RflySim installation script and connect the Pixhawk with RC receiver according to the figure at right
- Use the recommended compiling command px4fmu-v3_default
- Use the recommended PX4 firmware version 1.7.3
- Use the recommended compiling toolchain Msys2
- Pixhawk 1 has LED on the shell, so no external LED is required.

Toolbox one-key installation script

(1) Software package installation directory

C:\PX4PSP

(2) PX4 firmware compiling command: firmware versions <= PX4-1.8 use format px4fmu-v3_default; >= PX4-1.9 use format px4_fmu-v3_default

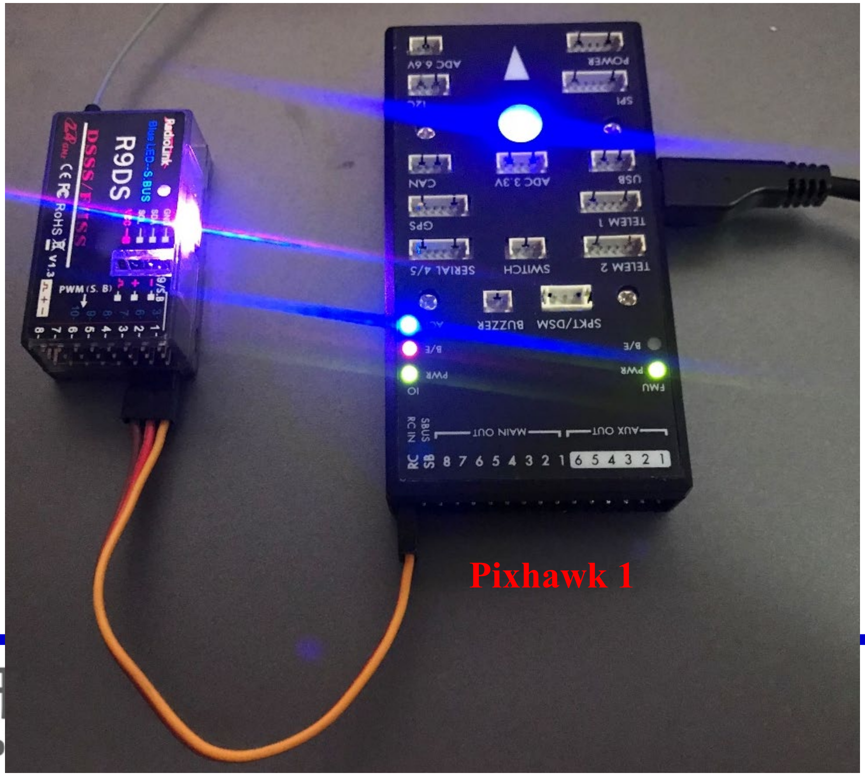
px4fmu-v3_default

(3) PX4 firmware version (1: PX4-1.7.3, 2: PX4-1.8.2, 3: PX4-1.9.2, 4: PX4-1.10.2)

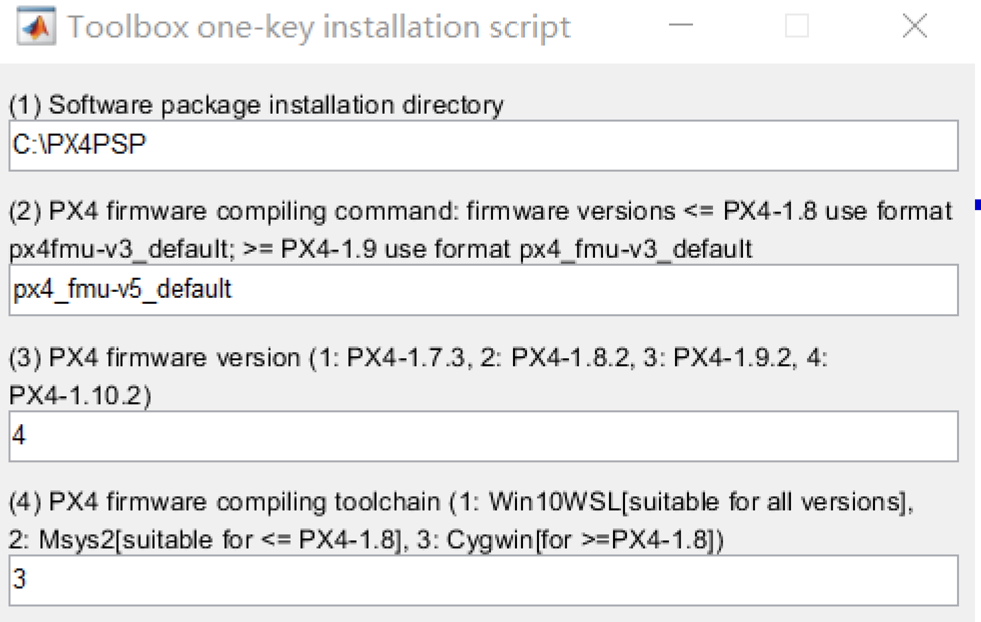
1

(4) PX4 firmware compiling toolchain (1: Win10WSL[suitable for all versions], 2: Msys2[suitable for <= PX4-1.8], 3: Cygwin[for >=PX4-1.8])

2



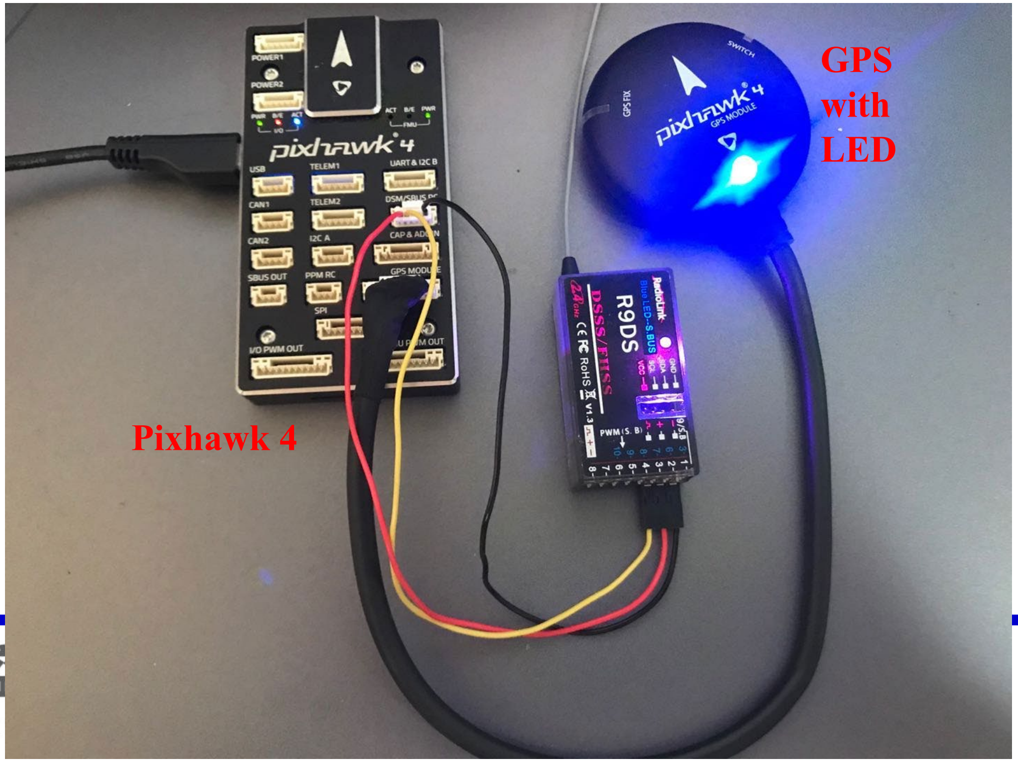
Software Package Insta



☐ Advanced Settings

Example 2: if you are using the Pixhawk 4 hardware (fmu-v5), you should adopt the configuration as shown in the figures at right:

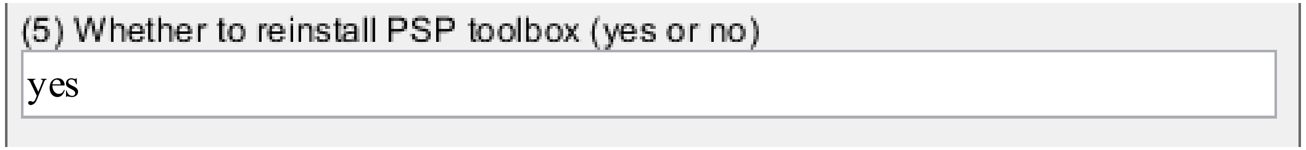
- Please select PX4 firmware >= 1.8.2 (PX4 1.10.2 is recommended), because it is known that PX4 1.7.3 cannot recognize the S.BUS RC, you may need a PPM module to transfer RC signals.
- Use a GPS module with LED integrated, because Pixhawk 4 has no LED on the body. Besides, the you should use JST GH interface to connect RC receiver.
- Use compiling command px4_fmu-v5_default, and compiling toolchain Cygwin or Win10WSL



Note: you should buy a RflySim advanced version to get supports for all PX4 firmware (1.9 to latest) and compiling toolchains (Cygwin and Win10WSL)

Software Package Installation

□ Advanced Settings

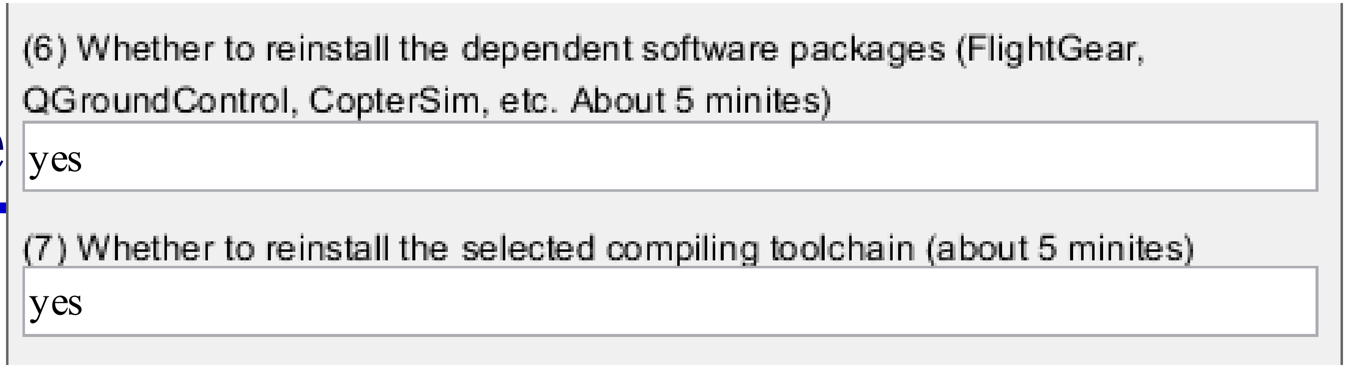


Note that the CygwinToolchain only supports PX4 firmware with version 1.8 or above; the Msys2Toolchain only supports PX4 firmware with version 1.8 or below. Both the CygwinToolchain and the Msys2Toolchain support Windows 7 and above, which are easy to deploy, but the compiling speed is slow. For Windows10 with version 1809 and above, readers can follow the tutorial in “0.UbuntuWSL\Win10UbuntuInstallationStep.docx” to install an Ubuntu subsystem in Windows and then choose the Win10WSL. The Win10WSL toolchain can greatly accelerate the compiling speed and is compatible with all versions of PX4 firmware.

5. Whether to reinstall the PSP Toolbox (yes or no). If this option is set to “yes”, the PSP Toolbox is installed on MATLAB/Simulink. If the PSP toolbox has already been installed, a new installation of the PSP Toolbox is performed. If this option is set to “no”, the script does not do anything on the existing PSP toolbox (it will not uninstall the PSP toolbox or carry out other actions).

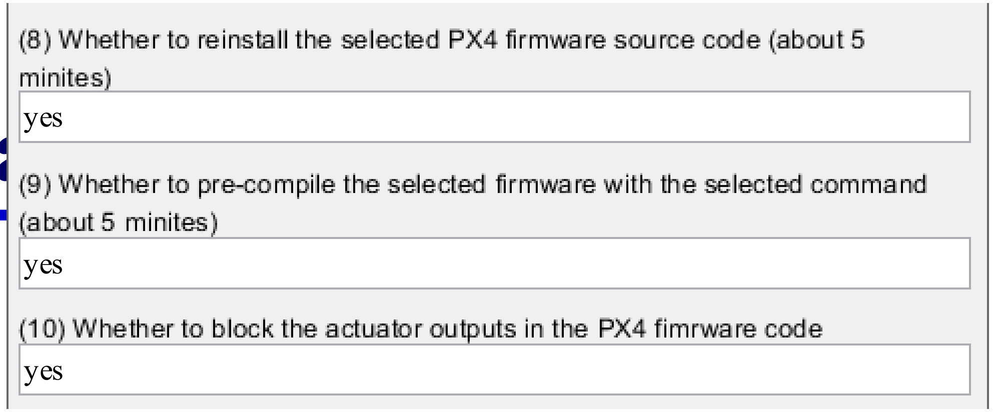
Software Package

☐ Advanced Settings



6. Whether to reinstall the dependent software packages. If this option is set to “yes”, software tools (such as FlightGear, QGC, CopterSim, and 3DDisplay) are deployed to the selected installation directory and shortcuts for them are generated on the desktop. The related drivers for Pixhawk hardware are also installed. If the software tools have already been installed, selecting “yes” will remove the old installation files and reinstall them. If this option is set to “no”, then no change will be made.
7. Whether to reinstall the selected compiling toolchain. If this option is set to “yes”, the selected compiling toolchain (Win10WSL, CygwinToolchain, or Msys2Toolchain) will be deployed to the selected installation directory. If the toolchain already installed, the script will remove the old toolchain files and reinstall it. In contrast, if this option is set to “no”, then no change will be made.

Software Package Installa



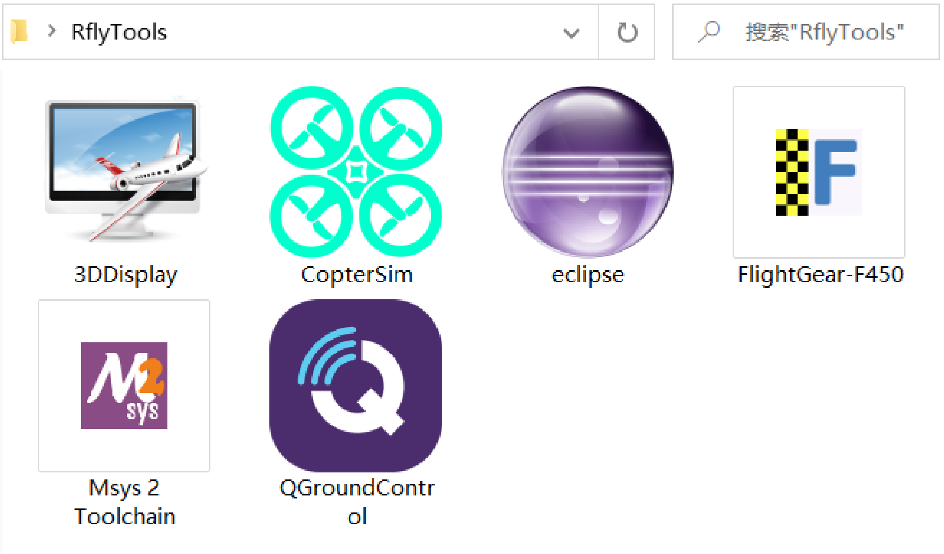
◻ Advanced Settings

8. Whether to reinstall the selected PX4 firmware source code. If this option is set to “yes”, the selected PX4 firmware source code will be deployed to the selected installation directory. If the firmware files already exist, the old firmware folder will be deleted, and a new copy of the source code will be deployed. If this option is set to “no”, then no change will be made.
9. Whether to pre-compile the selected firmware. If this option is set to “yes”, the PX4 source code will be pre-compiled. This can greatly save the compiling time of the subsequent code generation process; whether the compiling environment is installed properly can also be checked. If this option is set to “no”, then no change will be made.
10. Whether to block the actuator outputs of the PX4 original controller. If this option is set to “yes”, the control signals of the PX4 original controller will be blocked to prevent them from conflicting with the generated controller in Simulink. This option must be set to “yes” for the simulations and experiments in this book. If this option is set to “no”, the PX4 outputs will not be blocked, and this mode can be used to test the PX4 original controller.

Software Package Installation

☐ Installation Completion

1) When the above one-key installation scripted is successfully executed, the shortcuts for the core software tools will be generated in the RflyTools folder on the Desktop. according to the specific compiler type selected by the user configuration, only one compiler shortcut, Msys2Toolchain, CygwinToolchain and Win10WSL, will appear.



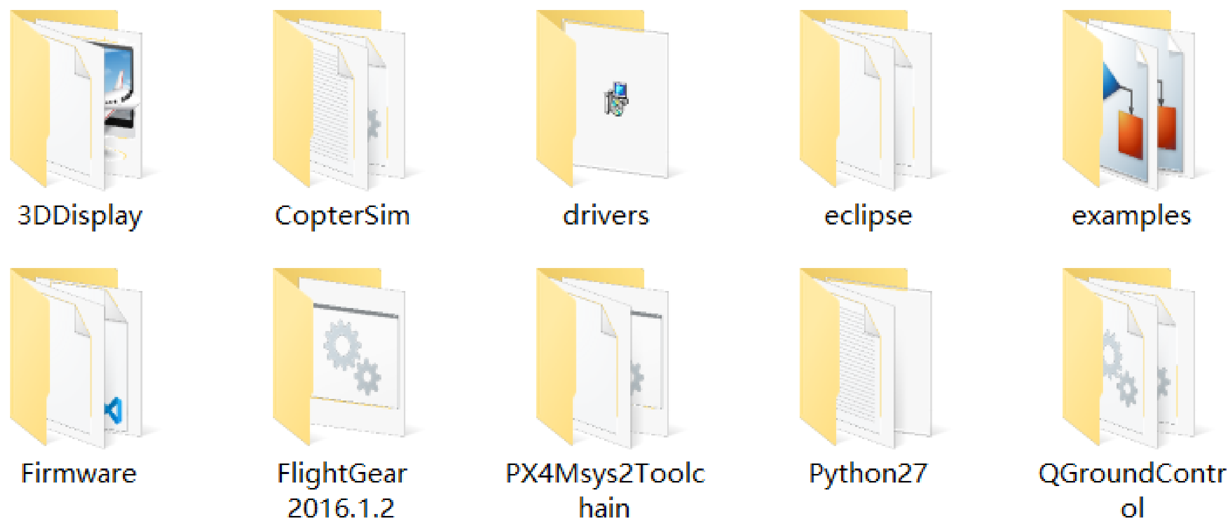
Note: if you encounter any problem during the installation process, please 1) uninstall all antivirus programs or completely close them in task manager; 2) close Win10 system's Real-time protection; 3) run "4.HILApps\MSVCP_2019.07.20_X64.exe" program to fix; 4) restart computer and rerun the script.

Software Package Installation

☐ Installation Completion

2） As shown in right figure, the folders of all software tools are stored in the selected installation directory (the default is "C:\PX4PSP").

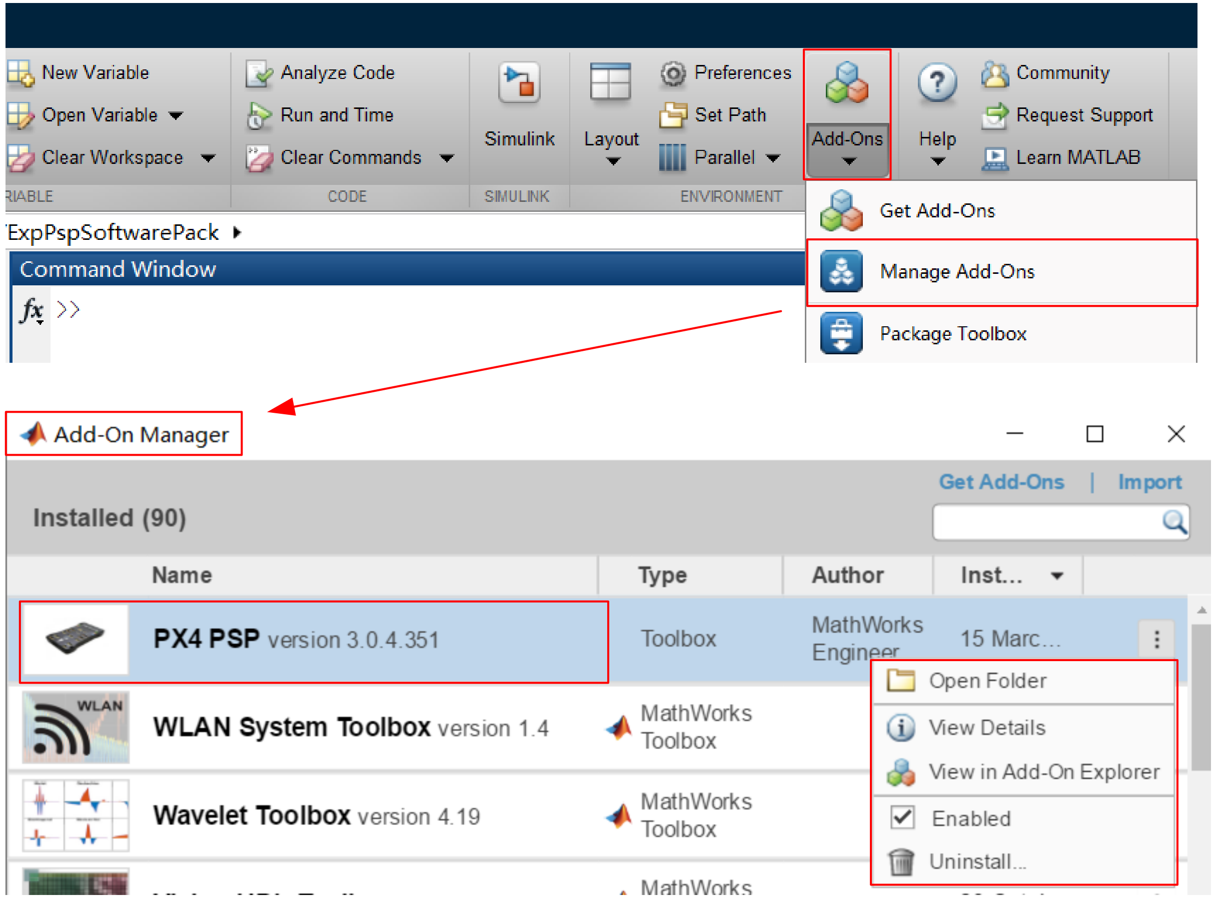
Note that all the software tools are completely portable and independent of the original software (e.g., official versions of QGC and FlightGear) on Windows. The folder "Firmware" stores the PX4 source code. The folder "examples" stores Simulink source code examples of the PSP toolbox; the folder "drivers" stores Pixhawk drivers. The folder "Python27" stores a Python environment for automatic firmware upload.



Software Package Installation

☐ Installation Completion

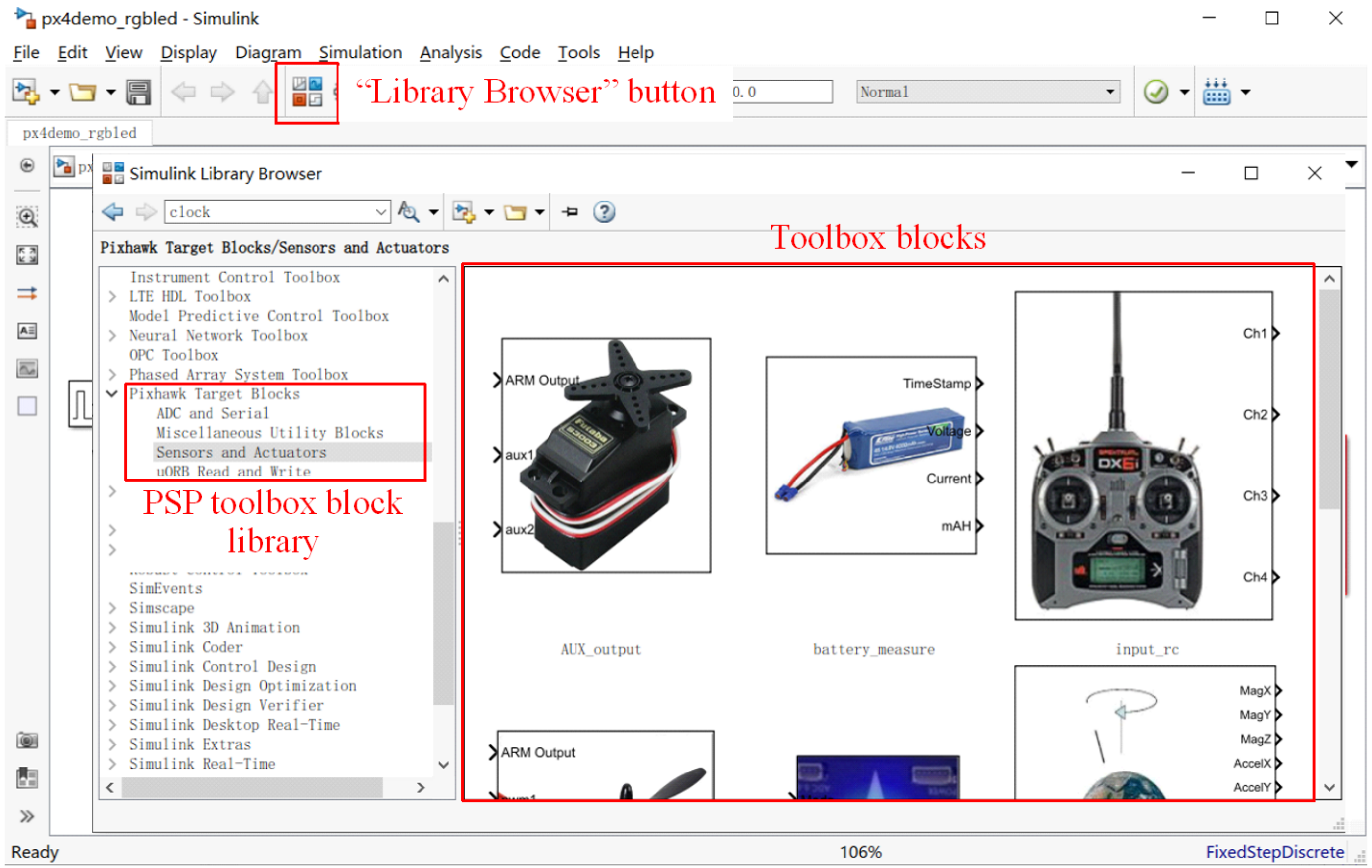
3) As shown in right figure, the installed PSP toolbox can be found on the “Add-Ons” - “Manage Add-Ons” page of MATLAB. On this page, some management operations can be performed for the PSP toolbox that includes disabling, uninstalling, and viewing the installation directory. Note that the PSP toolbox can be installed once for all the MATLAB applications on a computer whose versions are higher than or equal to MATLAB R2017b.



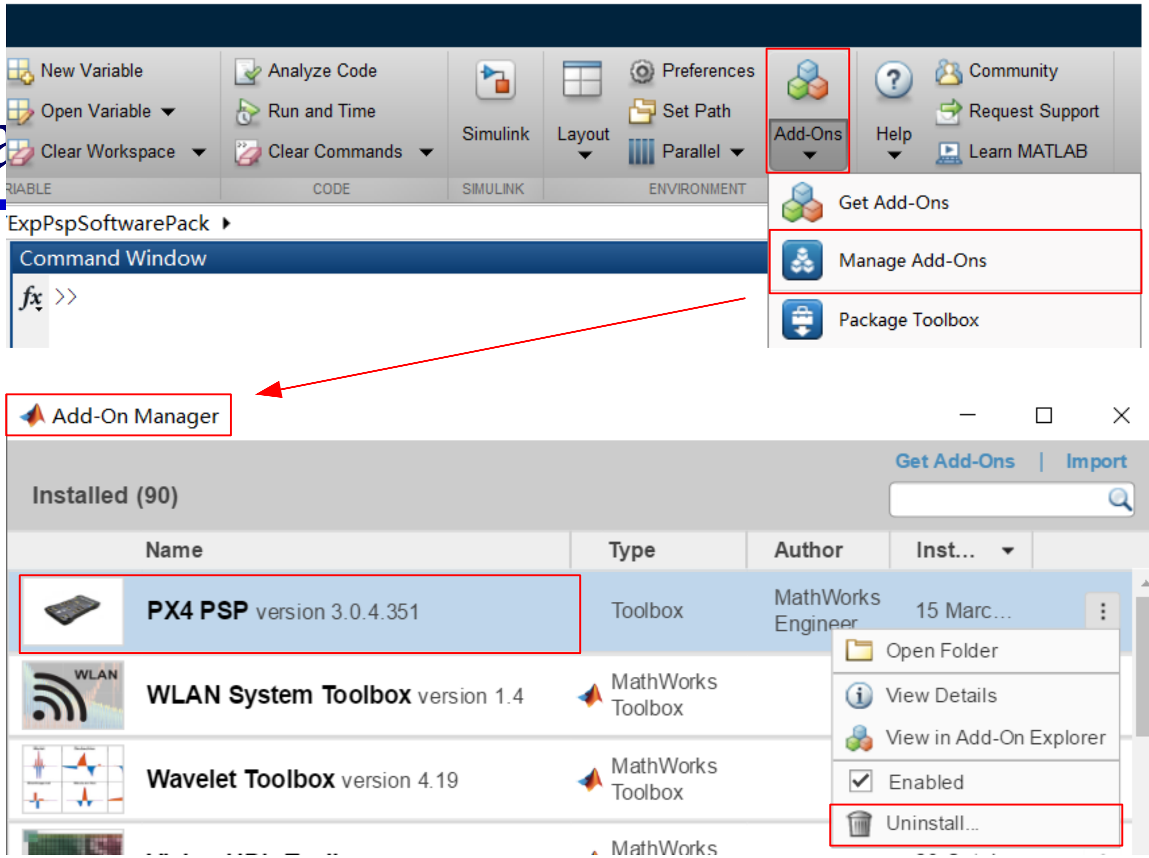
Software Package Installation

□ Installation Completion

4）As shown in right figure, the readers can open any Simulink file and click the “Simulink Library Brower” button to open the Simulink library browser, and then find the “Pixhawk Target Blocks” library generated by the PSP toolbox.



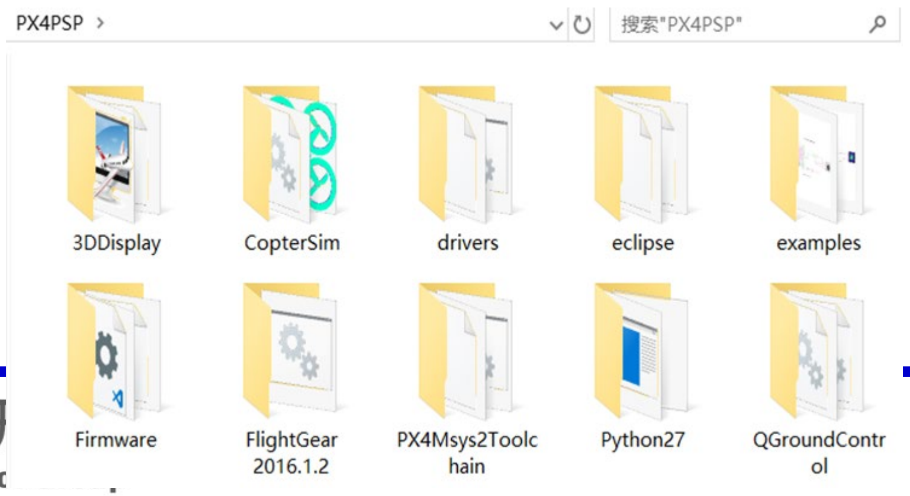
Software Package Insta



☐ Installation Completion

If the readers want to uninstall the simulation software package, they can carry out the following steps.

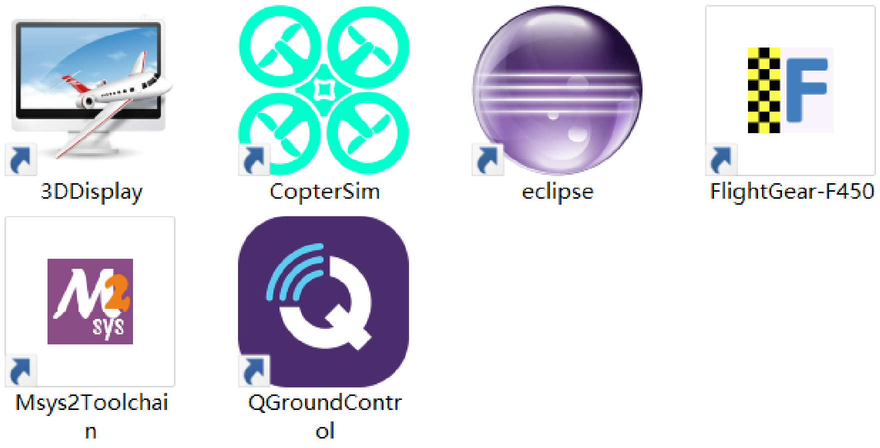
1. Delete all the Desktop\RflyTools shortcuts;
2. Delete all files and folders in the installation directory (the default is C:\PX4PSP);
3. In the “Management Additional Functions” page of MATLAB, click the “Uninstall” button to uninstall the PSP toolbox. (if you failed to uninstall it, you can also close MATLAB and then delete the [Documents]\MATLAB\Add-Ons\Toolboxes\PX4PSP folder to uninstall it)
4. Restart MATLAB and enter "edit pathdef.m" to check and delete any possibly residual PX4 items
5. [Documents]\Ogre folder stores files such as serial number, you can keep them on your computer.

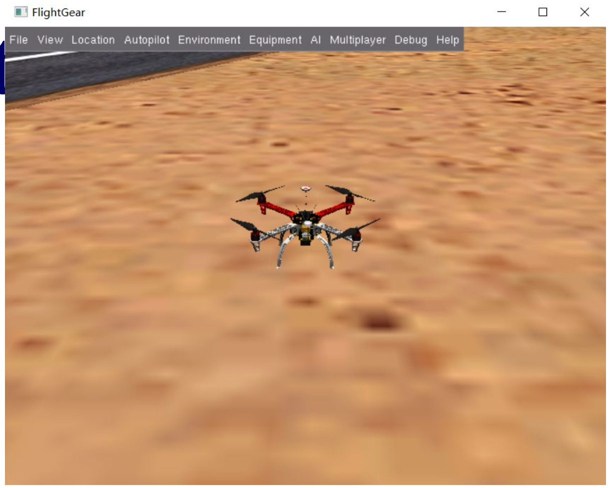


Software Package Installation

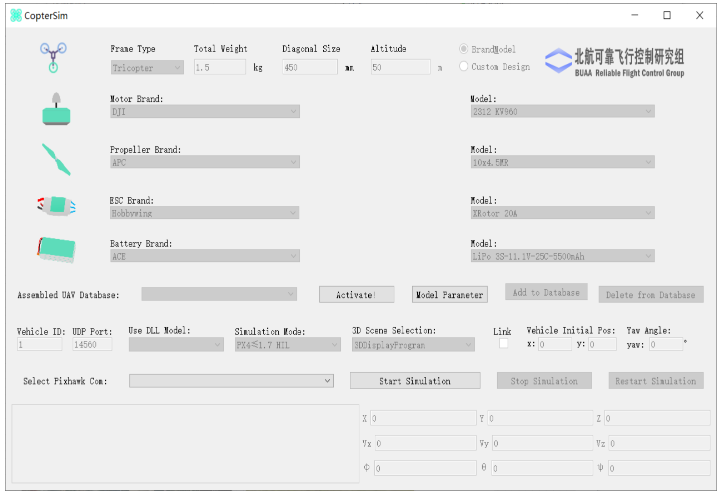
□ Brief Introduction to Soft

1) Double-click the desktop shortcuts, which include "FlightGear-F450", "CopterSim", "QGroundControl" and "3DDisplay". Then, check the software User Interface (UI) one by one with right figure to confirm that each software can operate correctly.

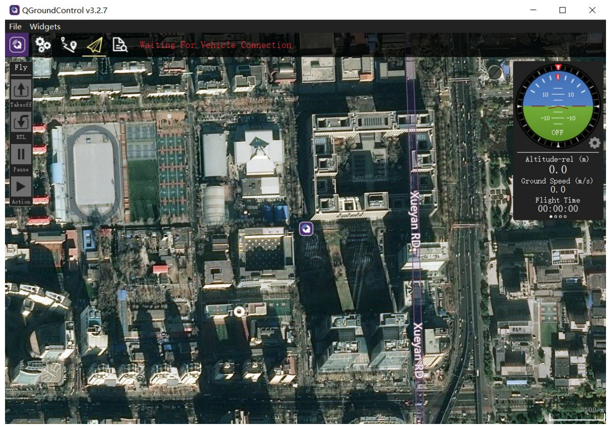




(a) FlightGear-F450



(b) CopterSim



(c) QGroundControl

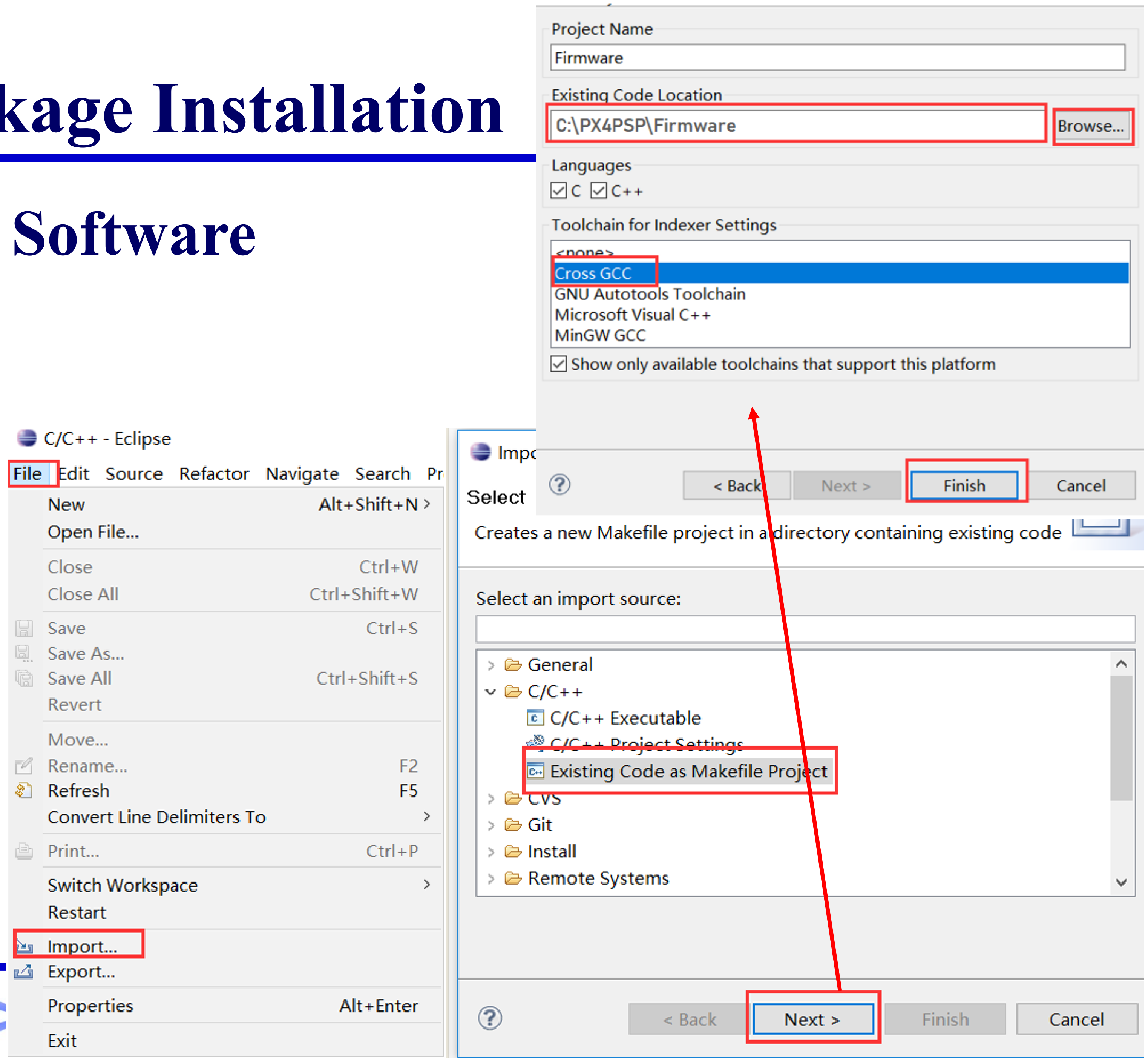
(d) 3DDisplay

Software Package Installation

□ Brief Introduction to Software

2) Double-click the desktop shortcut “Eclipse” to open the Eclipse software. As shown in right figure, click “File” – “Import...” – “C/C++” – “Existing Code as Makefile Project” from the menu bar of the Eclipse, and then click “next”. In the “Existing Code Location” section of the pop-up window, click the “Browse” button to locate the “Firmware” folder in the installation directory (default is “C:\PX4PSP”), then choose “Cross GCC” and click the “Finish” button.

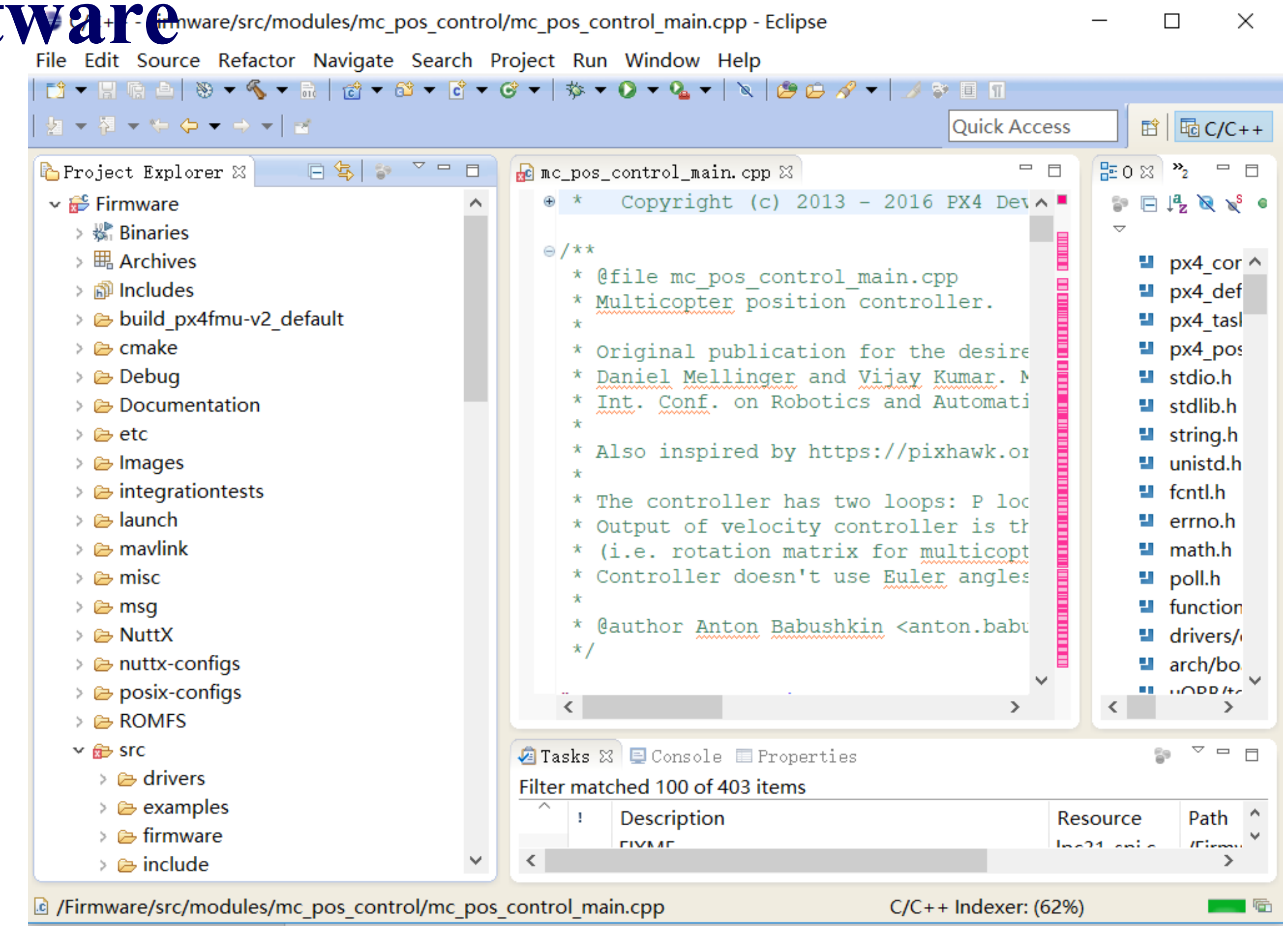
Note: Eclipse is only for reading and modifying source code, not affecting normal operations of the whole platform. If Eclipse can not operate properly in your computer, you can use VS Code or other IDEs



Software Package Installation

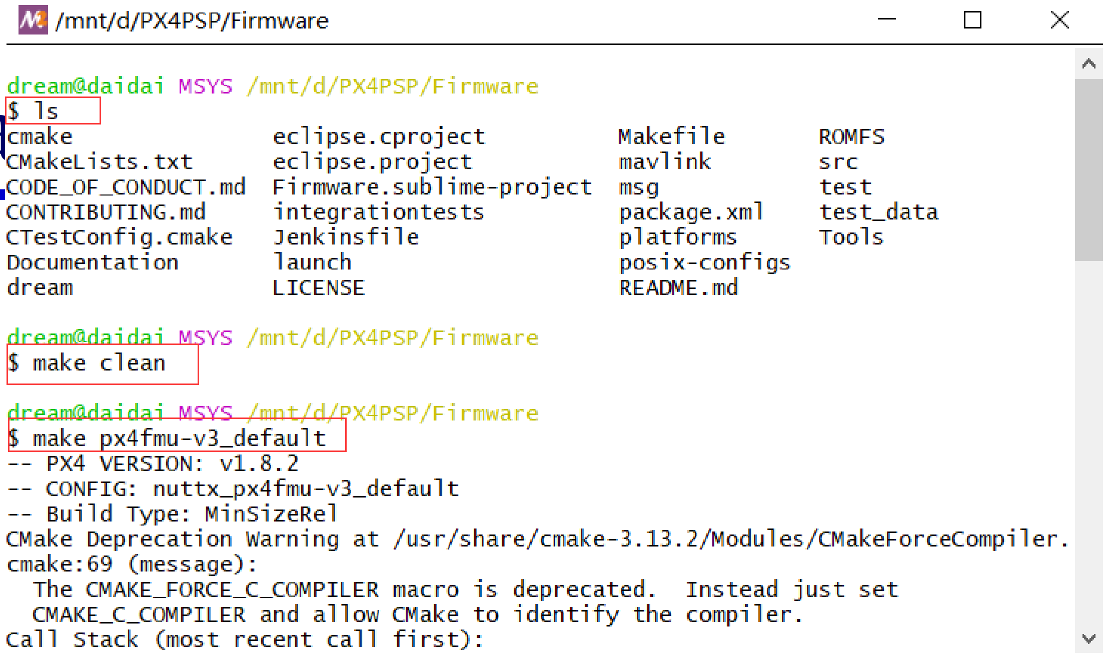
□ Brief Introduction to Software

3) After completing the above steps, as shown in right figure, the source code and directory structure of the “Firmware” can be found in the “Project Explorer” window, where readers can read the PX4 source code and try to modify it. Readers can also view the PX4 developer documentation to clearly understand the architecture and implementation principles of the PX4 algorithms and deepen the understanding of an actual flight control system. Note that: a “Welcome” tab will cover the content when readers first open Eclipse, it must be closed manually.



Software Package Insta

□ Brief Introduction to Software



4） Double-click one of the three shortcuts “Win10WSL”, “Msys2Toolchain” or “CygwinToolchain” on the desktop to pop up the command window interface shown in right figure (the original UI has a pure black background, and the image color has been reversed for reading). Because the compiling toolchains are essentially Linux emulation software, the basic Linux commands (such as “ls”, “pwd”, and “gcc --version”) can be tapped on the command line. For readers who are unfamiliar with Linux operations, this compiling toolchain can also be used as a Linux learning and practice tool. The most important function of this toolchain is to compile the source code of PX4 and generate the “.px4” firmware file. As shown in the figure, “make clean” can be tapped on the command line to clear the old compiling files, and the “make px4fmu-v3_default” command is used to compile the source code to the firmware file “C:\PX4PSP\Firmware\build\px4fmu-v3_default\px4fmu-v3_default.px4” for Pixhawk 1 (2MB flash). Because the PSP toolbox will automatically call this compiling command after the code is generated, the readers do not need to know how to use it.

Hardware Platform Configura

☐ RC System Configuration

- There are two RC system products presented in this book, which are RadioLink AT9S and Futaba T14SG. RC transmitters with “Left-hand throttle (Mode 2)” configuration are selected in this book, whose left stick is the throttle lever without the auto-return function.The receivers of these RC systems have the S.BUS output function that can transmit the PWM signals of all channels to the flight control through one data line.
- Radio Link AT9S is relatively inexpensive, and it is more suitable for indoor experiments; Futaba T14SG is relatively expensive, but it offers better performance and reliability, which makes it suitable for actual outdoor flight tests.
- The following subsections detail the configuration steps of the two RC systems. Other RC systems can be configured in a similar way.

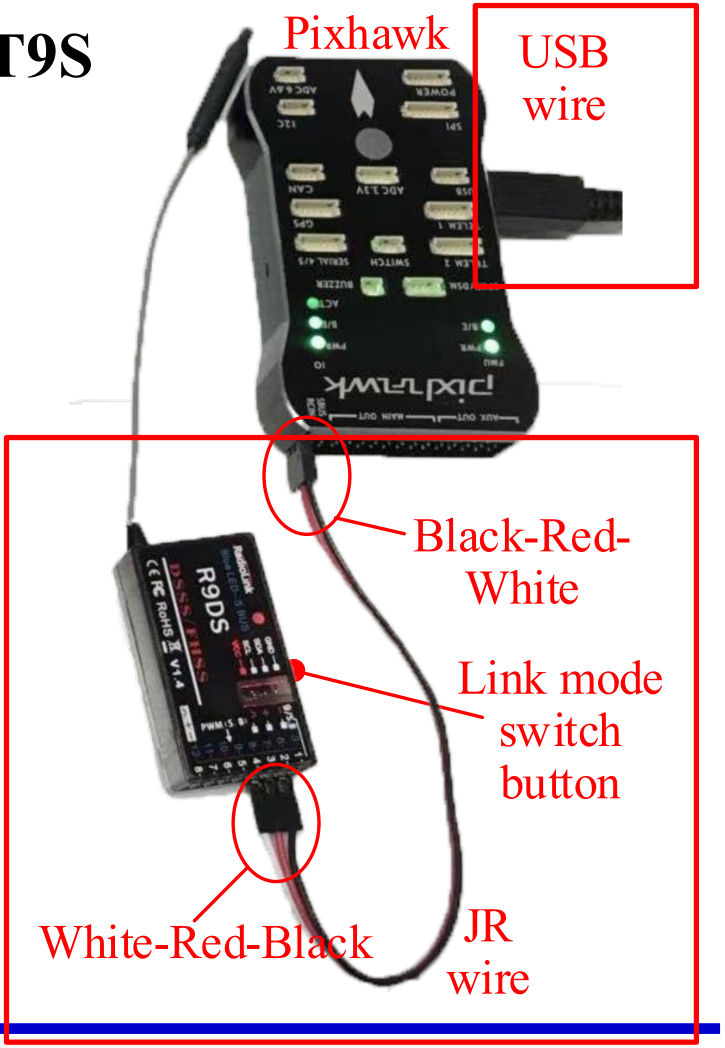
LadioLink AT9S

Futaba T14SG

Hardware Platform Configuration

□ RC System Configuration (1) RadioLink AT9S

The preferred RC transmitter for this experiment is the RadioLink AT9S RC transmitter and the matching R9DS receiver. The necessary accessories include: a battery (LiPo battery, 3S, 11.1V) and a matching charger for powering RC transmitter; the JR cable (also can be replaced by DuPont cable), used to connect the receiver with Pixhawk autopilot; MicroUSB data cable, used to connect Pixhawk autopilot and computer.

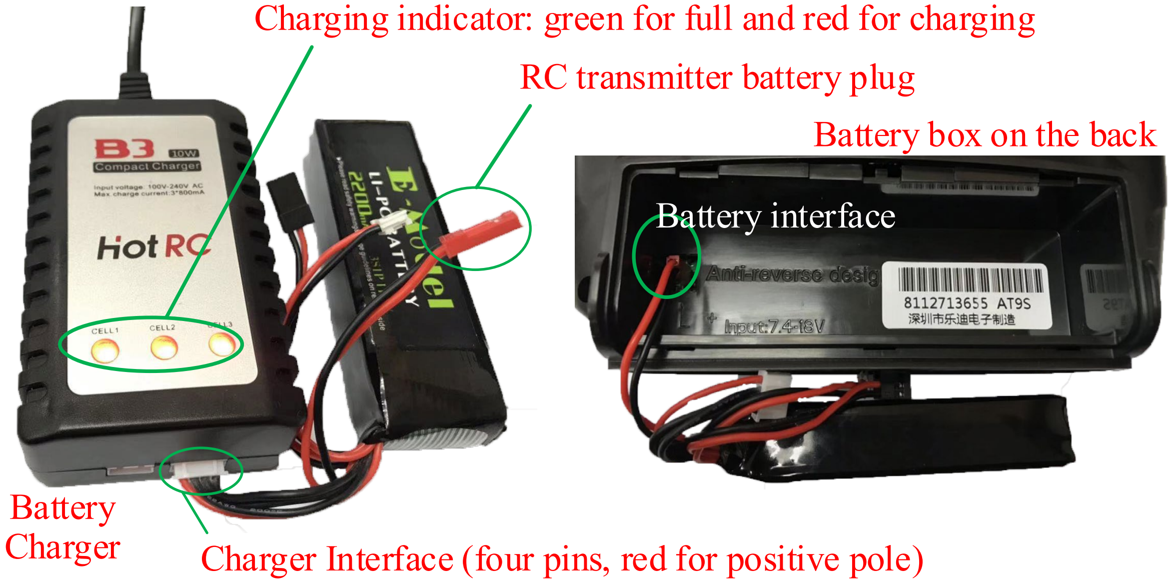


Hardware Platform Configuration

◻ RC System Configuration (1) RadioLink AT9S

1) Battery and charger instructions

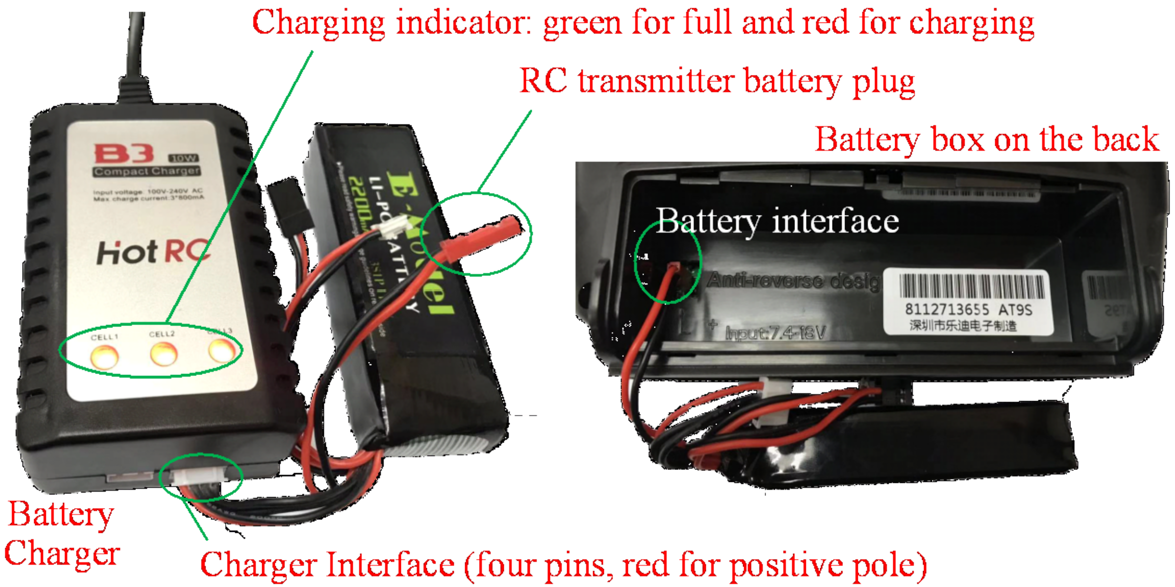
1. The left side of figure shows the battery and the charger. The battery begins to charge when the four-port charging head of the battery is inserted into the socket on the charger. Red and green indicator colors represent the "charging" and "fully charged" status, respectively. The RC transmitter battery is installed as follows. Open the battery slot on the back of the RC transmitter and insert the two-line battery power supply interface (red line for the battery positive pole) into the battery power slot. Note: Unplug the battery from the RC transmitter while charging. Avoid using the RC transmitter while charging.



☐ RC System Configuration (1) RadioLink AT9S

1) Battery and charger instructions

2. The battery installation method is as follows: open the lower battery slot on the back of the RC transmitter (push it down), and insert the battery-powered head (red connector, red and black wires corresponding to the positive and negative electrodes) to the left of the battery slot of the RC transmitter plug in the two port on the side, make sure the positive electrode (red wire) is facing up.



Hardware Platform Configuration

□ RC System Configuration

(1) RadioLink AT9S

2) RC receiver initial setting

1. Connect the RC receiver and the Pixhawk according to the figure. The horizontal pin on the downside of the tail face of the receiver must be connected to the left-most RC pin on the tail face of the Pixhawk with a three-line JR line (pay attention to the color direction), and the Pixhawk MicroUSB port must be connected to the computer USB Type-A interface to supply power to the receiver and the Pixhawk.

Note: the arrows point to the increasing direction of PWM

Hardware Platform Configur

□ RC System Configuration

Note: the arrows point to the increasing direction of PWM

(1) RadioLink AT9S

2) RC receiver initial setting

2. How to rematch the RC transmitter with an RC receiver (the connection has been completed by default, and this step needs to be performed only when problems occur in the connection of the receiver and the transmitter). Turn on the power of the RC transmitter (all other RC transmitters should be turned off), and correctly connect the receiver with the Pixhawk and the computer. Then, press the matching switch on the right side of the receiver with a pen tip or needle (see the right) for more than one second. At this time, the LED of the receiver starts to flash, which indicates that it is searching for the nearest RC transmitter. When the receiver LED flashes seven or eight times and then remains constant, it means that the matching process is finished and a connection has been successfully established between the RC transmitter and receiver.

Hardware Platform Configur

◻ RC System Configuration

(1) RadioLink AT9S

Note: the arrows point to the increasing direction of PWM

2) RC receiver initial setting

3. S.BUS signal mode selection (the receiver is in this mode by default; thus, this step is typically not performed). The S.BUS mode allows the Pixhawk to transmit all channel PWM signals through one JR line. If the Pixhawk is powered up and connected to the receiver, the LED on the receiver is blue-white, which indicates that it is already in S.BUS mode and no setup is required. If the receiver LED is red, the readers need to double-press the matching switch on the right side of the receiver (press twice within one second). If the receiver LED turns blue-white, then the S.BUS mode has been successfully set.

Hardware Platform Configur

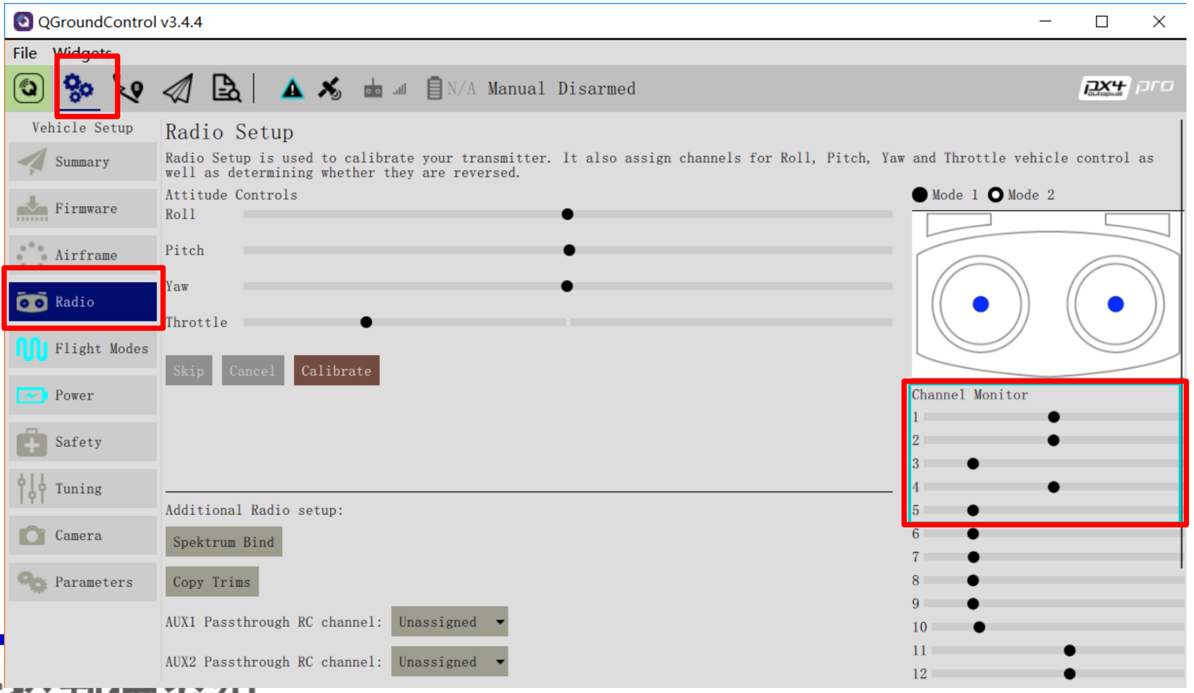
◻ RC System Configuration

(1) RadioLink AT9S

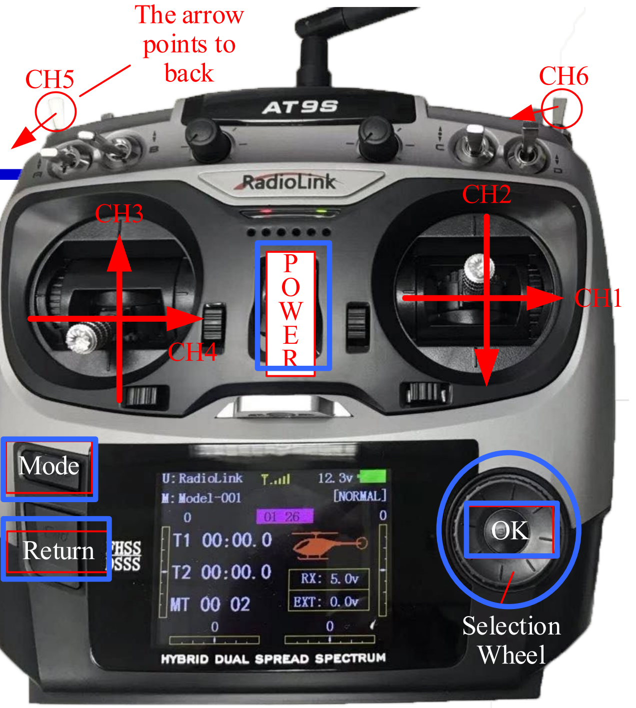
Method to check connection of RC system and Pixhawk

1. Connect the Pixhawk autopilot and the RC receiver with a JR wire.
2. Connect the Pixhawk autopilot with the computer with a Micro USB cable.
3. Power on the RC transmitter, and open the QGC.
4. When the QGC has successfully connected the Pixhawk, click the "Gear" button and then "Radio" button to enter Radio setting page
5. If the channels and slide bars can be observed on the "Channel Monitor" of the figure, it means the RC receiver has been connected successfully. Otherwise, please reconfigure it.

Note: the arrows point to the increasing direction of PWM



Hardware Platform Configuration



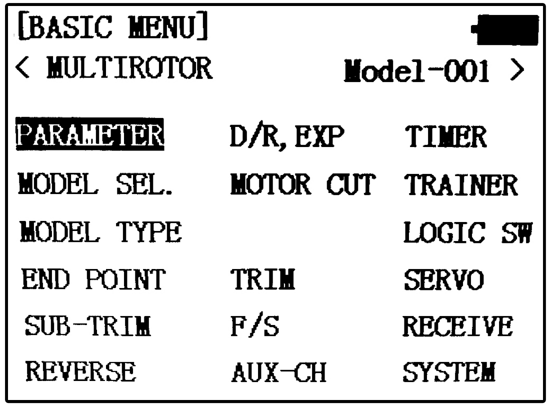
□ RC System Configuration (1) RadioLink AT9S

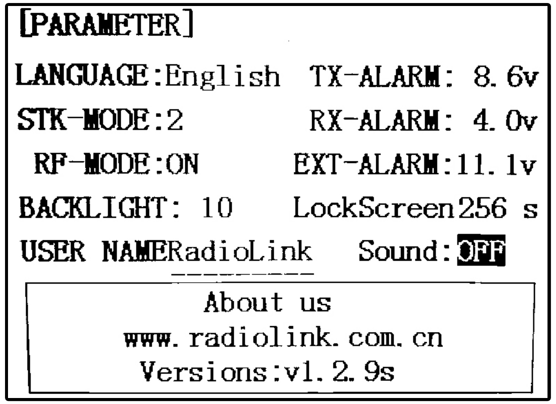
3) RC transmitter setting

1. Pull up the "POWER" switch to open the RC transmitter.

2 . Setting the Language and Turning off the Sound

- Press the "Mode" button on the RC transmitter for several seconds to enter the model setting page.
- Roll the "Selection Wheel" on the RC transmitter
- move the cursor to "PARAMETER", and press the "OK" button on the RC transmitter to enter the RC transmitter parameter setting page.
- Scroll the "Selection Wheel" to select the "English" item





Hardware Platform Configuration

RC System Configuration

(1) RadioLink AT9S

3) RC transmitter setting

- click the “OK” button, and then
- scroll the “Selection Wheel” again to select the desired display language
- Then click the “OK” button to confirm the selection
- modify the “Sound” option from “ON” to “OFF” by the same method

[BASIC MENU]

< MULTIROTOR Model-001 >

PARAMETER	D/R, EXP	TIMER
MODEL SEL.	MOTOR CUT	TRAINER
MODEL TYPE		LOGIC SW
END POINT	TRIM	SERVO
SUB-TRIM	F/S	RECEIVE
REVERSE	AUX-CH	SYSTEM

[PARAMETER]

LANGUAGE:English	TX-ALARM: 8.6v
STK-MODE:2	RX-ALARM: 4.0v
RF-MODE:ON	EXT-ALARM:11.1v
BACKLIGHT: 10	LockScreen256 s
USER NAMERadioLink	Sound:OFF

About us
www.radiolink.com.cn
Versions:v1.2.9s

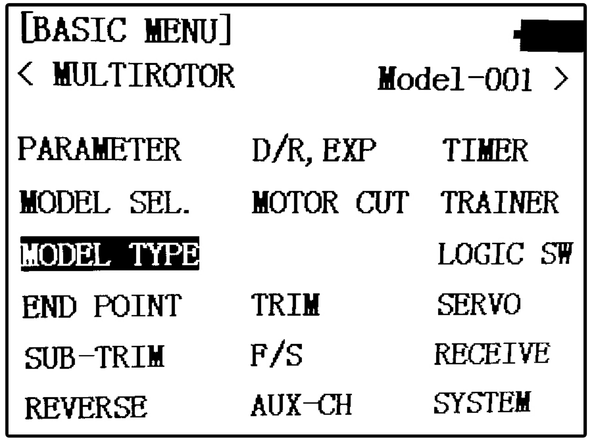
Hardware Platform Configuration

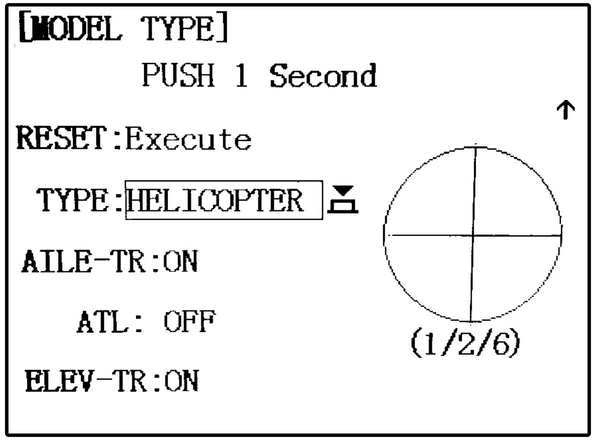
◻ RC System Configuration (1) RadioLink AT9S

3) RC transmitter setting

3. Multicopter mode setting

- Press the “Mode” button for several seconds to enter the “BASIC MENU” page, and click the “MODEL TYPE” item to enter the model type selection page shown in right figure.
- Change the “TYPE” item from “HELICOPTER” to “MULTICOPTER”, and then press the “OK” button for several seconds to set the control mode to “Multicopter”.



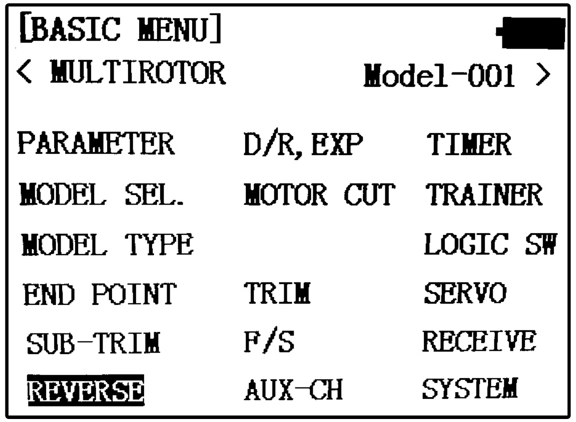


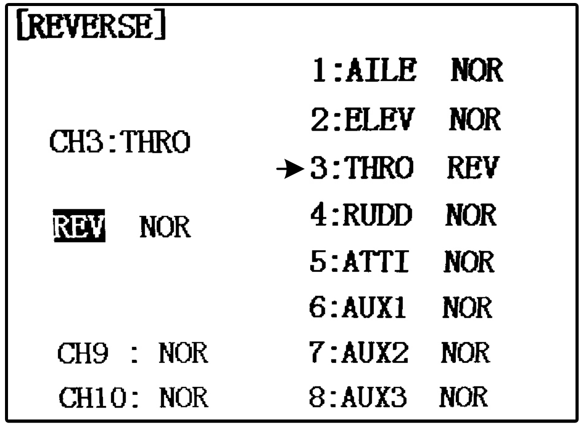
Hardware Platform Configuration

◻ RC System Configuration (1) RadioLink AT9S

4) Throttle channel reverse setting

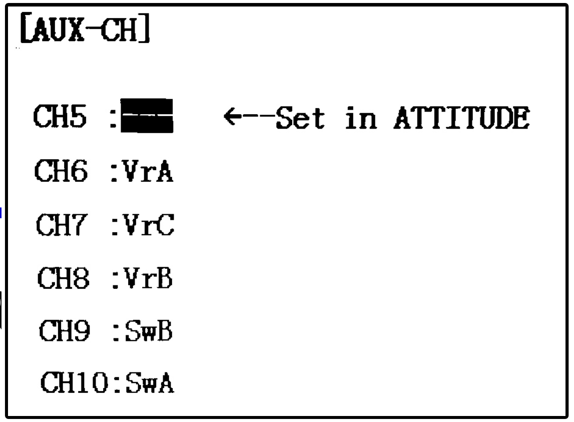
- The throttle channel of the RadioLink transmitter is opposite to normal RC transmitters, and the throttle channel reverse needs to be set.
- Press the "Mode" button for several seconds to enter the "REVERSE" setting page shown in right figure.
- Change the throttle channel from "NOR" to "REV".

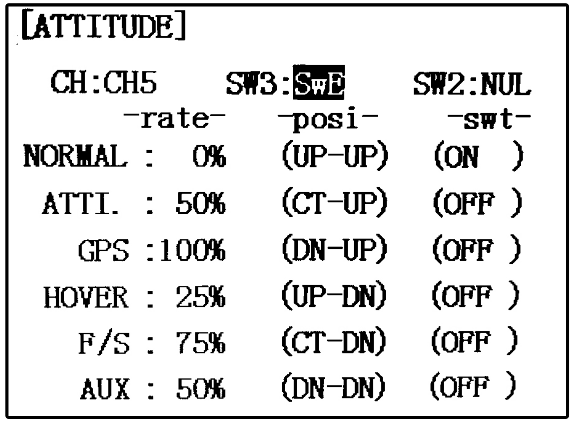




Hardware Platform C

◻ RC System Configuration (1) R





5) CH5-CH6 mode switching channel setting

- Because of experimental requirements, CH5 of the RC transmitter needs to be mapped to a three-position switch in the upper left corner for mode switching of the Pixhawk.
- Press the “Mode” button for several seconds, and click the “AUX-CH” item next to the “REVERSE” item.
- On the “AUX-CH” setting page, click the “CH5” item to enter the channel setting page, and map CH5 to a three-position switch “SwE” on the RC transmitter (switch “E” is located in the top-left corner of the RC transmitter.
- Similarly, the “CH6” item must be modified to a three-position switch “SwG” of the RC transmitter (switch “G” is located in the upper-right corner of the RC transmitter in right figure.

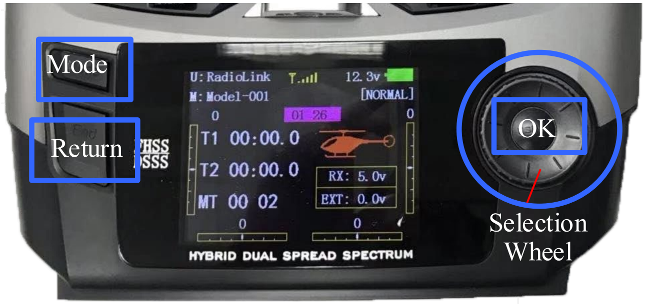
Hardware Platform Configuration

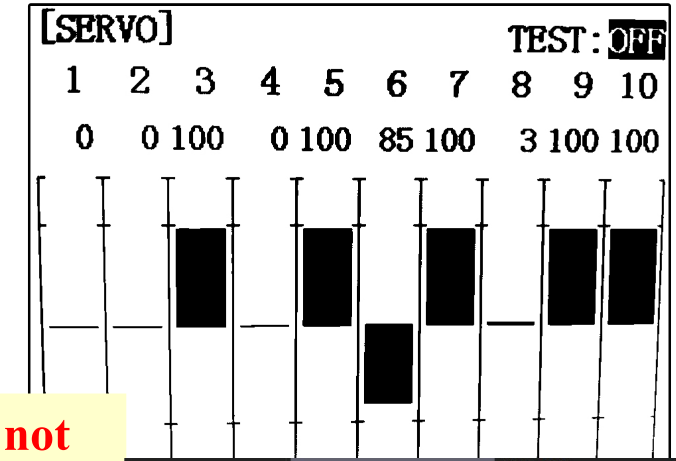
RC System Configuration (1) RadioLink AT9S

6) Channel confirmation

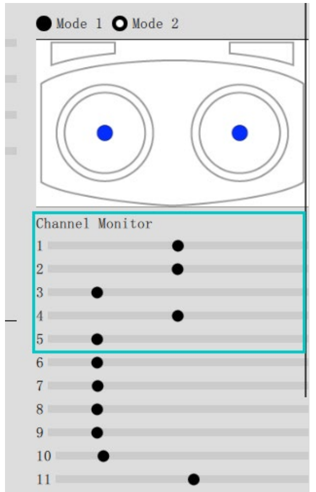
Restart the RC transmitter, and press the “Return” button (see right figure) on the RC transmitter to enter the “SERVO” page.

- In this page, the PWM value of each channel can be verified by moving sticks and switches on the RC transmitter.
- Note that, as shown in right figure, the channel value reaches an upper limit of 100 corresponding to the desired PWM value of 1100 (left position in Channel Monitor of QGC); the channel value reaches the lower limit 100, corresponding to the desired output PWM value of 1900 (right position in Channel Monitor of QGC).

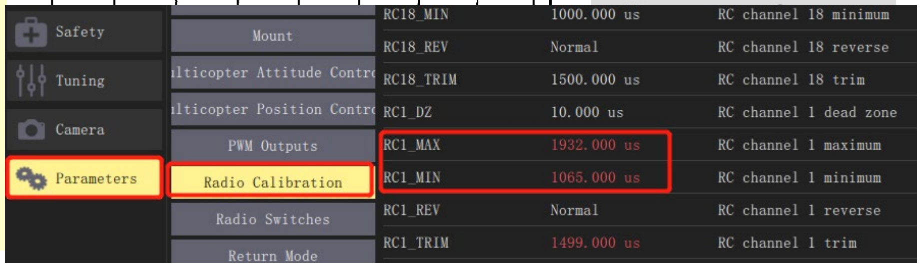




Channel Monitor in QGC



Note that the actual PWM value range received by the RC receiver may not equal to 1100 - 1900 due to various errors. For example, RadioLink AT9S usually ranges from 1065 – 1933 (can be observed in QGC’s Setting – Parameters – Radio Calibration page). So RC calibration is important for autopilots to correctly recognize the control commands from the ground pilot.



Hardware Platform Configuration

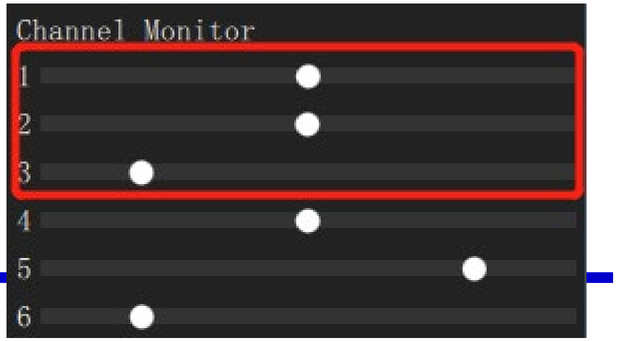
□ RC System Configuration (1) RadioLink AT9S

6) Channel confirmation

Move each channel stick in right figure to confirm that each channel corresponds correctly to the following rules.

1. CH1：this corresponds to the horizontal movement of the right-hand stick of the RC transmitter. The right-hand stick moves from left to right, corresponding to a PWM value that changes from 1100 to 1900 (QGC's Channel Monitor moves from left to right).
2. CH2：this corresponds to the vertical movement of the right-hand stick of the RC transmitter. The right-hand stick moves from top to bottom, corresponding to a PWM value that changes from 1100 to 1900 (QGC's Channel Monitor moves from left to right).
3. CH3：this corresponds to the vertical movement of the left-hand stick of the RC transmitter. The left-hand stick moves from top to bottom, corresponding to a PWM value that changes from 1900 to 1100 (opposite to CH2, right to left in QGC's Channel Monitor).

Note: the arrows point to the increasing direction of PWM

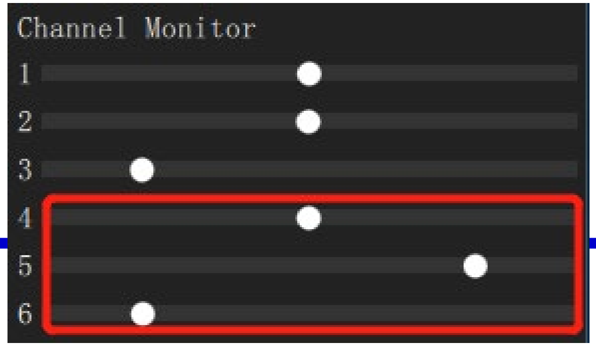


◻ RC System Configuration (1) RadioLink AT9S

6) Channel confirmation

4. CH4: this corresponds to the horizontal movement of the left-hand stick of the RC transmitter. The left-hand stick moves from left to right, corresponding to a PWM value that changes from 1100 to 1900 (QGC's Channel Monitor moves from left to right).
5. CH5: this corresponds to the three-position switch on the upper-left side of the RC transmitter. The switch moves from the top position (the farthest position to the user), middle position, and bottom position (the closest position to the user), corresponding to PWM values of 1100, 1500, and 1900, respectively. (left to right in QGC)
6. CH6: this corresponds to the three-position switch on the upper-right side of the RC transmitter. The switch moves to the top position (the farthest position to the user), middle position, and bottom position (the closest position to the user), corresponding PWM values of 1100, 1500, and 1900, respectively. (left to right in QGC)

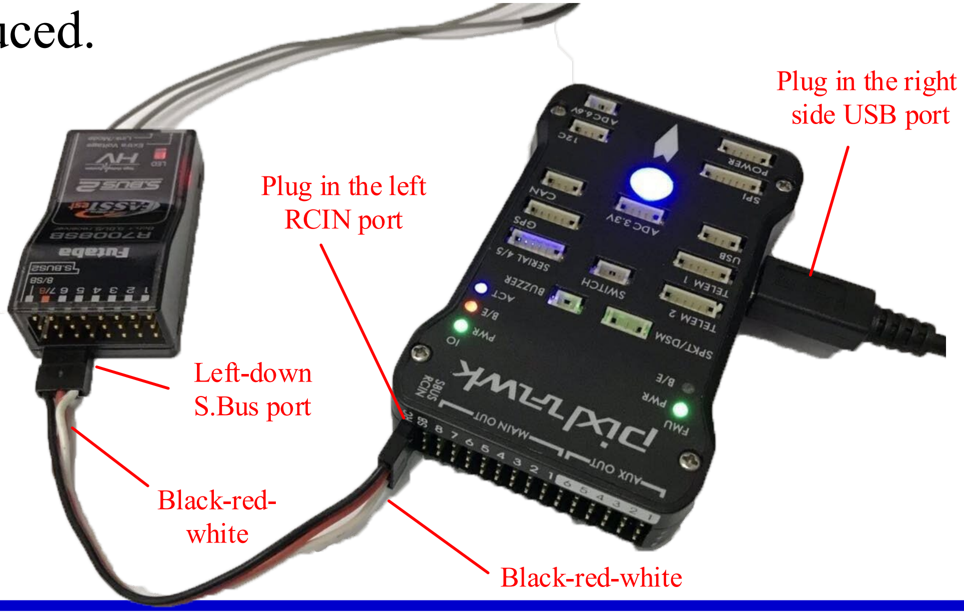
Note: the arrows point to the increasing direction of PWM



Hardware Platform Configuration

☐ RC System Configuration (2) Futaba T14SG

1. The connection between the Futaba receiver and the Pixhawk autopilot is slightly different from that of the RadioLink receiver.
2. The specific connection is shown in down figure. In the following paragraphs, the setup process for the Futaba RC transmitter is introduced.



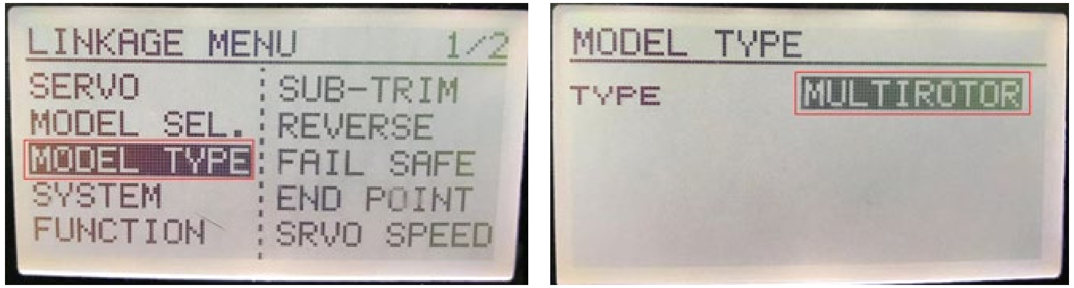
Note: the arrows point to the PWM increasing directions

Hardware Platform Co

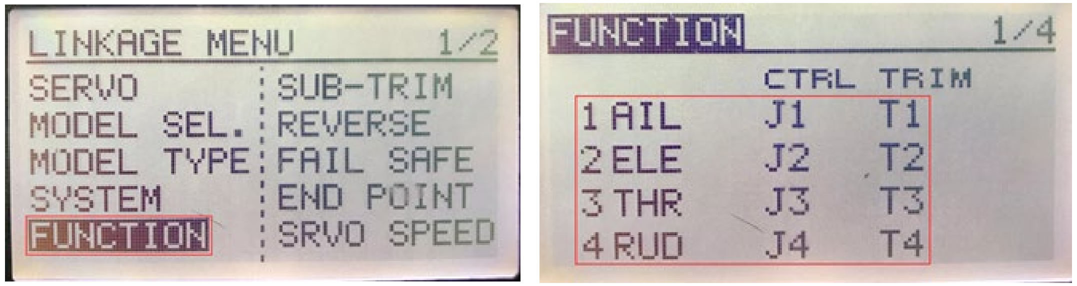
□ RC System Configuration (2) Fut

The Futaba T14SG RC transmitter needs to use six channels: the J1 stick (CH1, roll channel), J2 stick (CH2, pitch channel), J3 stick (CH3, throttle channel), J4 stick (CH4, yaw channel), SE three-position switch (upper-left switch, CH5 mode channel), and SG three-position switch (upper-right switch, CH6 mode channel). The basic process of setting the Futaba T14SG RC transmitter is summarized as follows.

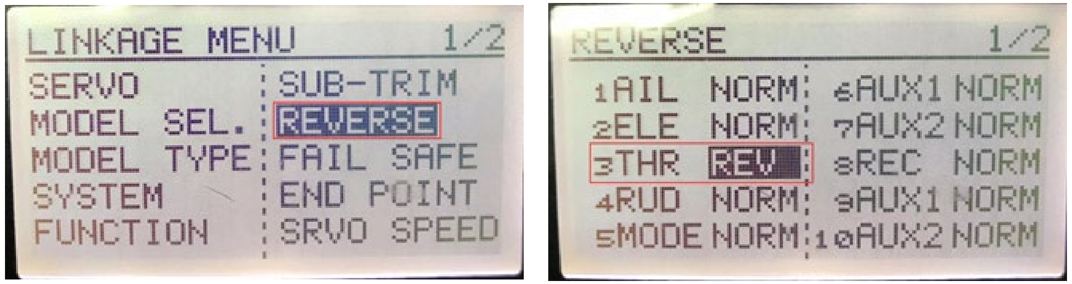
1. Double-click the “LINK” button on the RC transmitter to enter the “LINKAGE MENU” link setting page. Then, enter the “MODEL TYPE” page and change the “TYPE” to “MULTICOPTER”;



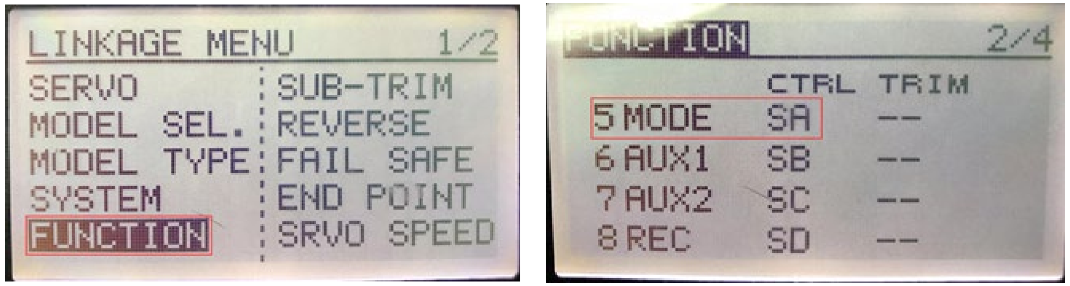
(a) Model type for multicopter



(b) Check channel definitions



(c) Reverse throttle channel



(d) Mode switch channel selection

Hardware Platform Co

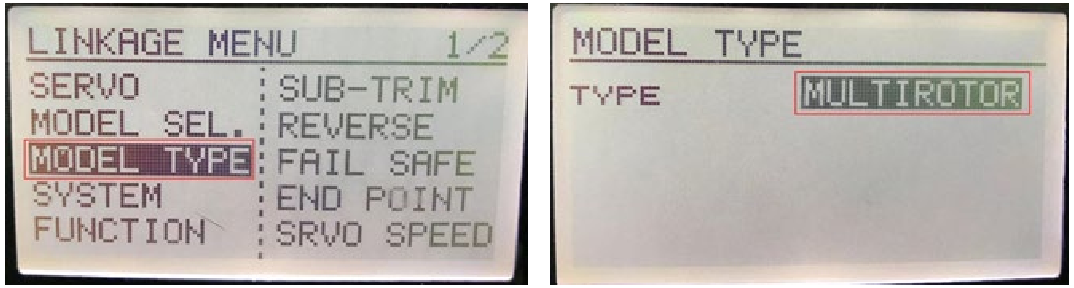
◻ RC System Configuration (2) Fut

2. Go back to the “LINKAGE MENU” page, and enter the “FUNCTION” page to confirm the channel mapping is in correct order J1 to J4 and T1 to T4;

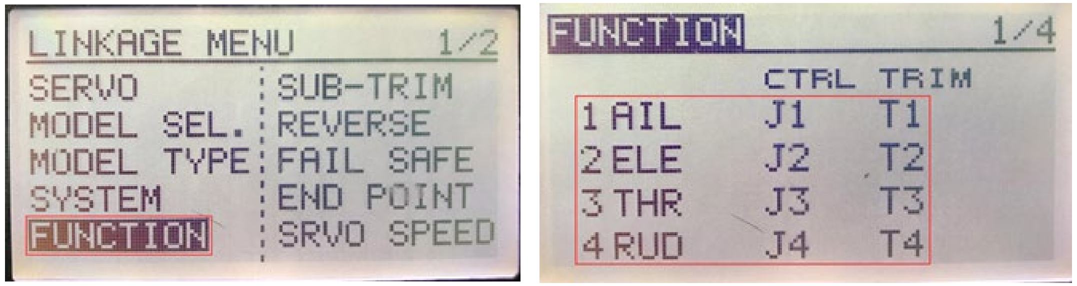
3. Go back to the “LINKAGE MENU” page, and enter the “REVERSE” page to confirm only the third channel (throttle) is reversed;

4. Go back to the “LINKAGE MENU” page, and enter the “FUNCTION”, and scroll to the second page for the setting of CH5 to CH8. Then, set the “CTRL” option of the “5 MODE” channel to “SE” stick (the upper-left stick of the RC transmitter).

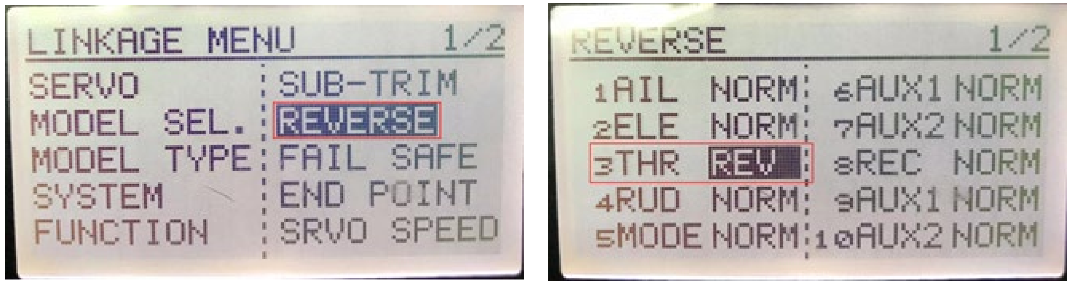
5. As in the previous step, set the “6 AUX1” channel to the “SG” stick (the upper-right stick of the RC transmitter).



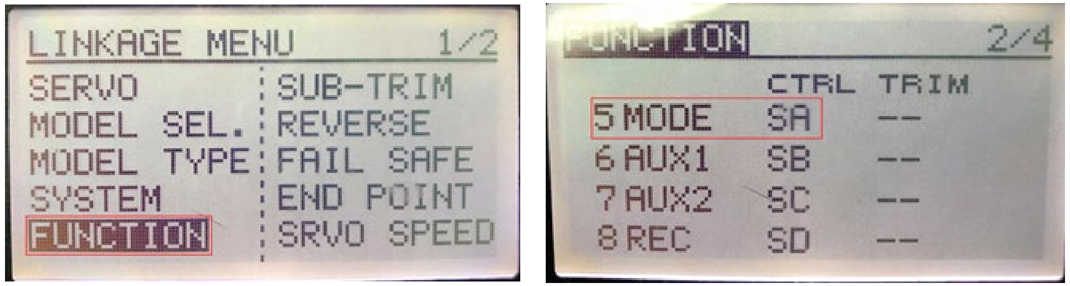
(a) Model type for multicopter



(b) Check channel definitions



(c) Reverse throttle channel



(d) Mode switch channel selection

Hardware Platform (

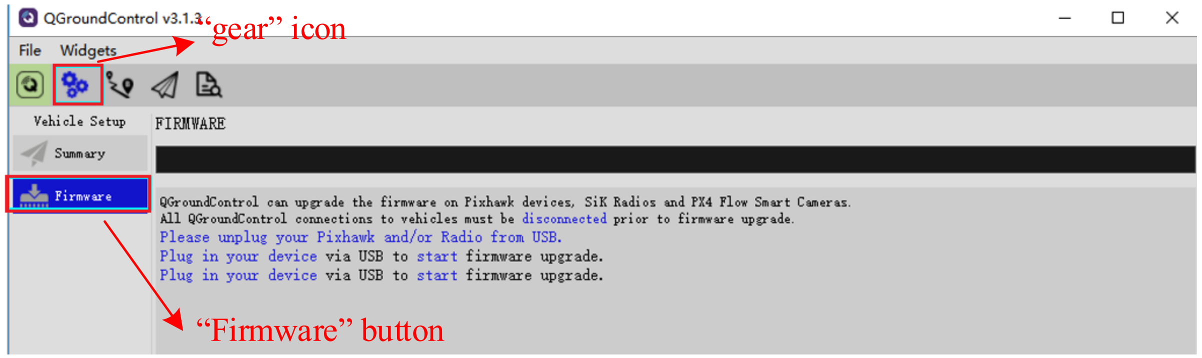
□ Pixhawk Configuration

Several basic firmware uploading and configuration operations are required for the brand-new Pixhawk to ensure that the Pixhawk autopilot meets the experimental requirements and ensure that the operation and configuration of Pixhawk are correct. The configuration method is summarized below.

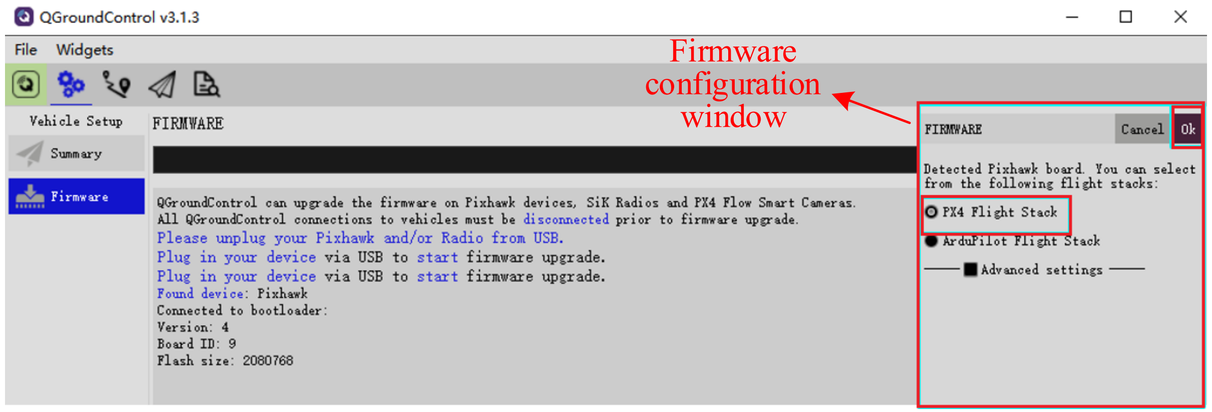
1. Open the QGC software.

2. As shown in the figure on the right, click the “gear” icon to enter the setting page; then, click the “Firmware” tab to enter the firmware uploading page.

3. Connect the Pixhawk autopilot and the computer using a USB cable. At this time, the software will automatically recognize the Pixhawk hardware.



(a) Setting page



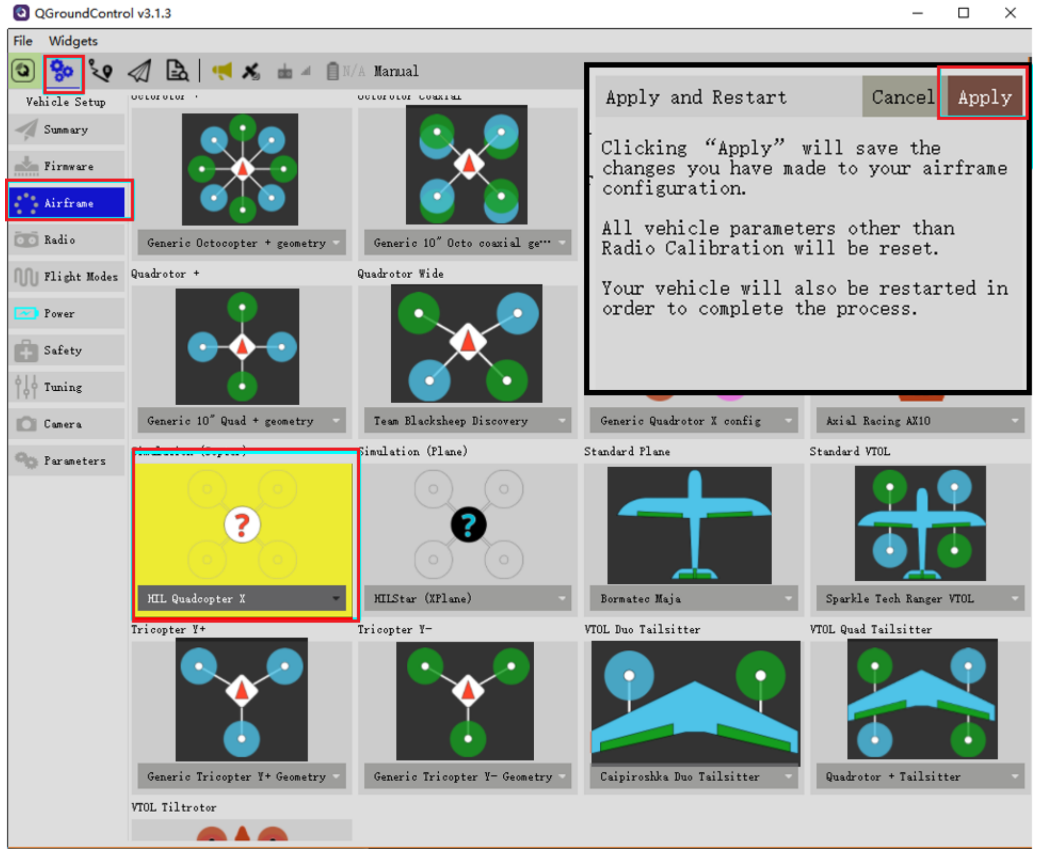
(b) Select “PX4 Flight Stack” item

Then, a firmware configuration window will pop up on the right side of the UI. Select the “PX4 Flight Stack” item, and click “OK”; then, QGC begins to automatically download and burn the latest PX4 firmware into the Pixhawk autopilot hardware.

Hardware Platform Configuration

□ Pixhawk Configuration

4. After the firmware is burned, the Pixhawk will automatically restart and reconnect to the QGC software. Then, as shown in figure on the right, enter the “Airframe” tab, select “HIL Quadcopter X” airframe type , and click the “Apply and Restart” button in the upper-right corner. Then the autopilot will automatically restart to finish the configuration for HIL simulation.



5. After rebooting, QGC will automatically reconnect to Pixhawk. Check each configuration page to ensure that the Pixhawk autopilot has been in the HIL simulation mode.

Hardware Platform Config

☐ Airframe Configuration

For the subsequent flight experiments, it is necessary to ensure that the configuration of the multicopter is as close as possible to the simulation model. The experiments presented in this book select the most popular F450 multicopter with the following configuration:

1. Airframe: DJI Flame Wheel F450 airframe

2. Propulsion system: DJI E310 propulsion suite (four motors, four ESCs, and four propellers)

3. Battery: GENS ACE LiPo 4000mAh-3S(11.1V)-25C

4. Autopilot: Pixhawk 1 (2M flash version), px4fmu-v3_default

5. GPS module: UBlox NEO-M8N GPS

6. Other accessories: power module, buzzer, safety switch, connector, and anti-vibration damper

The assembling methods can refer to PX4 website https://docs.px4.io

Resource

All course PPTs, videos, and source code will be released on our website

https://rflysim.com/en

For more detailed content, please refer to the textbook:

Quan Quan, Xunhua Dai, Shuai Wang. *Multicopter Design and Control Practice*. Springer, 2020

https://www.springer.com/us/book/9789811531378

If you encounter any problems, please post question at Github page

https://github.com/RflySim/RflyExpCode/issues

If you are interested in RflySim advanced platform and courses for rapid development and testing of UAV Swarm/Vision/AI algorithms, please visit:

https://rflysim.com/en/4_Pro/Advanced.html

Thanks