Chapter Outline

Preparation, Installation, First Run, and Maintenance

Chapter 2
Experimental Platform Configuration

2.1 Key Tools

2.2 Installation and First Simulation

2.3 Architecture and Workflows

2.4 API and SDK Interfaces
API: application programming interface
SDK: software development kit

2.5 Examples and Experimental Routes

2.6 Exercises and Discussion

4 Units in This Section

2.2.1 Preinstallation Setup
Textbook: Sec. 2.1.3

2.2.2 Installation and Configuration
Textbook: Sec. 2.1.4

2.2.3 Unified Launcher and First Simulation
Textbook: Sec. 2.1.4

2.2.4 Hardware and Maintenance Options
Textbook: Sec. 2.1.5

2.2.1 Preinstallation Setup: Task Selection

Prepare the Environment for the Experiment

Environment / Tool	Requirements	What It Does
Software-in-the-loop SITL (textbook: SIL)	Windows 10/11 x64 PC meeting performance requirements; RflySim Toolchain	PC runs controller and vehicle; no physical autopilot required.
MATLAB/Simulink Development	MATLAB R2022b or later and the toolboxes required by the model	Model with SITL/HITL; no physical board for modeling alone.
Hardware-in-the-loop HITL (textbook: HIL)	Pixhawk autopilot; radio-control (RC) transmitter; onboard computer as needed; other hardware	Check algorithm execution, communication, and timing on a physical board.
Real-Vehicle Tests	HITL hardware + real vehicle; task sensors (e.g., cameras/lidar)	Sensor data/actuator commands use real-vehicle interfaces.

With adequate PC performance, SITL supports the vast majority of course experiments.

2.2.1 Preinstallation Setup: Recommended Configuration

Recommended RflySim Toolchain Configuration

- **OS:** Windows 10 (1903 or later) or Windows 11, both x64
- **CPU:** 10th-generation Intel Core i5 or newer, or equivalent AMD processor
- **GPU:** NVIDIA GeForce RTX 2060 or higher, or equivalent AMD GPU
- **RAM:** 16 GB or more; DDR4-3200 or equivalent/higher performance recommended
- **Storage:** At least 100 GB free on the installation drive (SSD recommended)
- **Display:** 1080p (1920*1080) or higher (dual monitors recommended)
- **Ports:** At least one USB Type-A port (an adapter may be used)
- **Optional software:** MATLAB R2022b or later

Note: Visual resolution, scene complexity, and vehicle count determine the actual simulation load.

Note: Low-level editing is light; builds/simulation need spare RAM/disk. For vision, simultaneous localization and mapping (SLAM), and multiple vehicles, assess CPU/GPU load by resolution, target frame rate, and count.
Configuration details: HowToInstall.pdf

Editing is light; builds, real-time simulation, and 3D rendering depend on CPU/RAM/GPU.

2.2.1 Preinstallation Setup: MATLAB Option

Choose MATLAB Toolboxes for the Model

Basic RflySim simulation needs no MATLAB. Modeling and low-level control in Chapters 4 and 5, and Simulink swarm control in Chapter 10, require MATLAB R2022b or later and the following toolboxes:

- **Aircraft modeling:** Aerospace Blockset / Toolbox provides aircraft models, environments, and analysis tools.
- **Code generation:** MATLAB Coder and Simulink Coder convert supported functions and models, respectively, to C/C++.
- **Tasks and interfaces:** UAV Toolbox supports autonomous missions; ROS Toolbox connects Robot Operating System (ROS) nodes and messages.
- **Platform support package (PSP):** After registration, the Library Browser shows Pixhawk Target Blocks.
- **Dependencies:** Blocks used by the model determine the toolboxes required; code generation also needs the corresponding products and build environment.

Basic simulation needs no MATLAB. For model development, select toolboxes for modeling, code generation, and external interfaces.

Chapter Outline

Preparation, Installation, First Run, and Maintenance

Chapter 2
Experimental Platform Configuration

2.1 Key Tools

2.2 Installation and First Simulation

2.3 Architecture and Workflows

2.4 API and SDK Interfaces

2.5 Examples and Experimental Routes

2.6 Exercises and Discussion

4 Units in This Section

2.2.1 Preinstallation Setup
Textbook: Sec. 2.1.3

2.2.2 Installation and Configuration
Textbook: Sec. 2.1.4

2.2.3 Unified Launcher and First Simulation
Textbook: Sec. 2.1.4

2.2.4 Hardware and Maintenance Options
Textbook: Sec. 2.1.5

2.2 Installation and First Simulation

2.2.2 Installation and Configuration: Installation Package

Installation Image: Three File Types

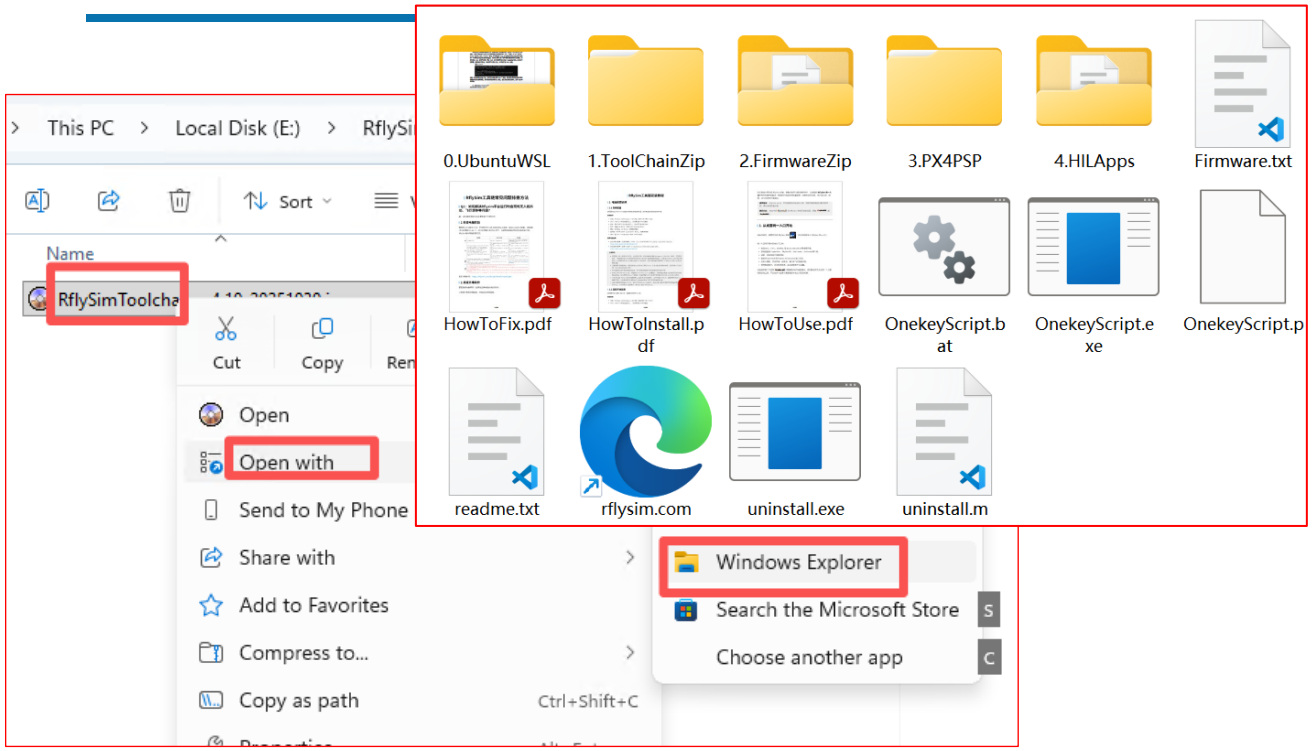


Figure: Current installation package and documentation

Download the ISO image: https://rflysim.com/download. To mount it, right-click > Open with > Windows Explorer:

- **0.UbuntuWSL:** Before installation, configures Windows Subsystem for Linux (WSL), runtime libraries, and device drivers.
- **OnekeyScript.exe:** Installs component packages 1-4 in the installation directory.
- **PDF guides:** HowToInstall: installation; HowToUse: operation; HowToFix: troubleshooting.

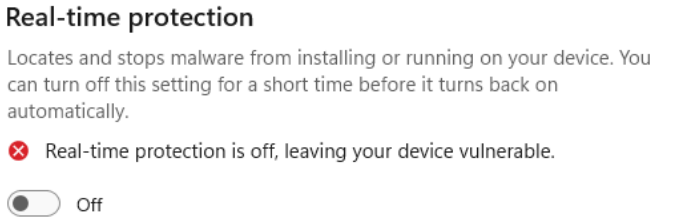
Package: components, installer, guides; OnekeyScript sets up the working environment.

2.2.2 Installation and Configuration: Security Software

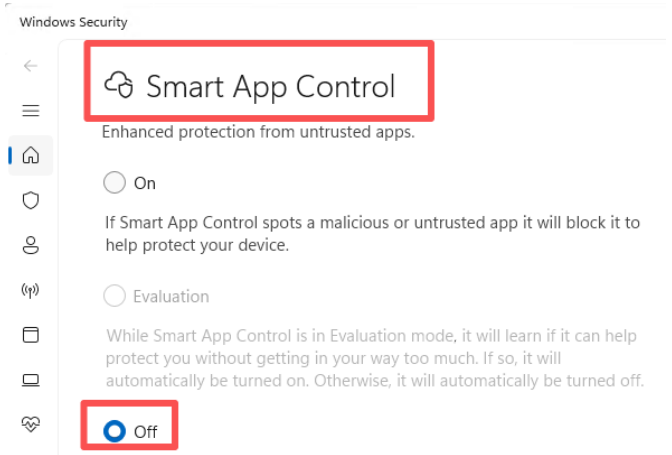
Avoid False Security Blocks During Installation

- **If blocked:** Verify the official source, signature, or official checksum, and identify the exact file that was blocked.
- **Real-time protection:** Keep it on by default; after confirming a false positive, prefer a trusted single-file exclusion.
- **Third-party antivirus:** Keep it on by default; inspect the detection record instead of exiting it without checking.
- **Smart App Control:** Keep it on by default; if blocked, verify the publisher and how it can be re-enabled on the current Windows version.
- **After installation:** Review exclusions and restore any protection settings changed temporarily.

Protection is shown off.
Not a default install step.



Real-time protection settings



Smart App Control settings

Protection on by default. If blocked, verify package and file; make the narrowest adjustment.

2.2 Installation and First Simulation

2.2.2 Installation and Configuration: Environment Initialization

Enable WSL: Authorize, Restart, and Continue

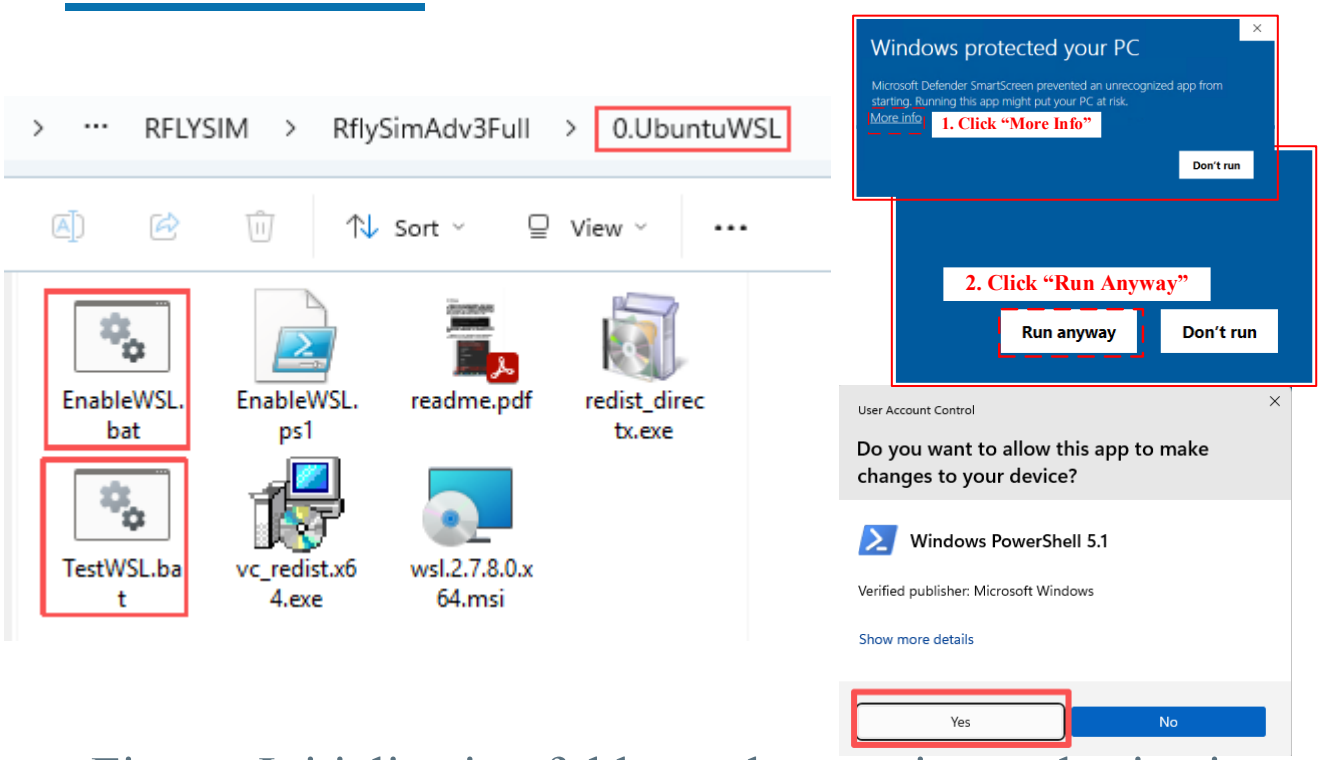


Figure: Initialization folder and execution authorization

- **Start:** Run EnableWSL.bat. Blue warning: verify source and signature/checksum; proceed only if trusted.
- **Restart:** Save work; press Y to restart, then rerun the script to continue.
- **Check:** PASS = passed; assess WARN effects; FAIL: inspect the failed component/log.
- **Verify:** Run TestWSL.bat. "WSL is installed successfully" = WSL available.
- **WSL:** Runs PX4 builds/execution; ROS.

EnableWSL.bat enables WSL and installs runtimes/drivers; restart, continue setup, and verify.

2.2 Installation and First Simulation

2.2.2 Installation and Configuration: Installer

EXE Installer: Quick Installation

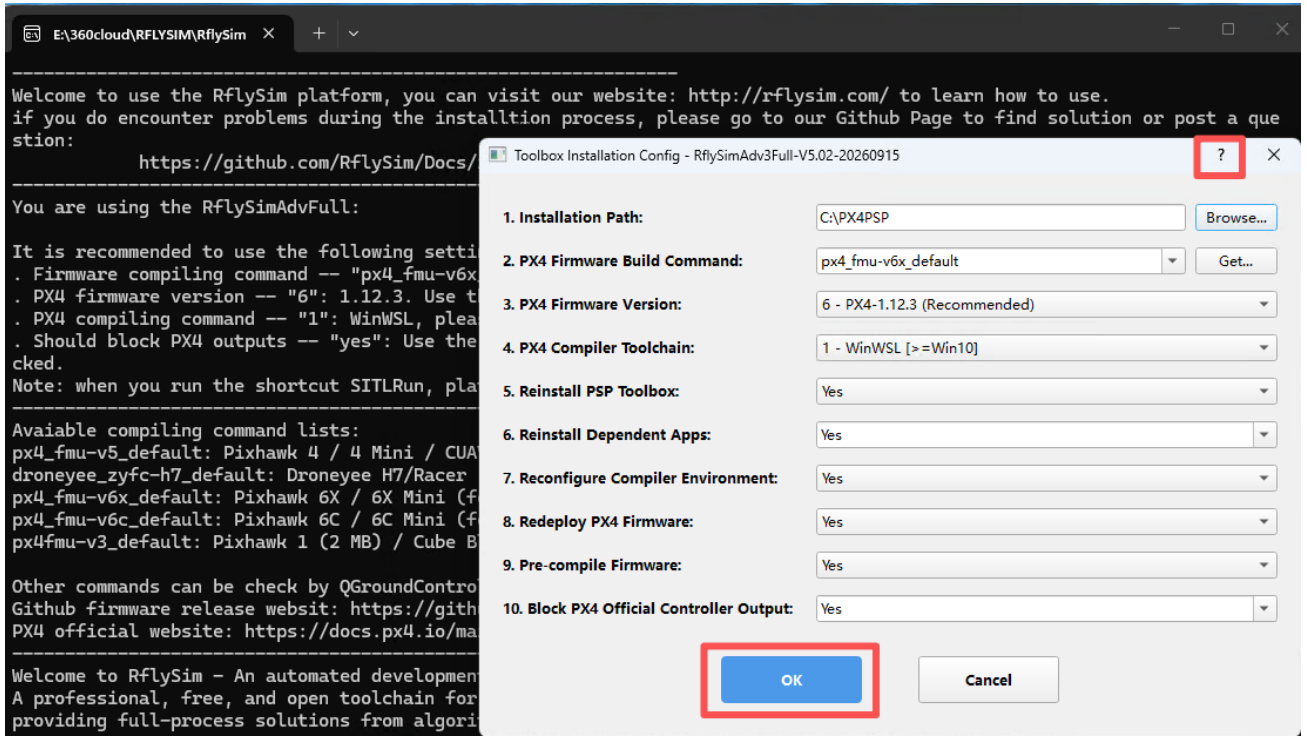
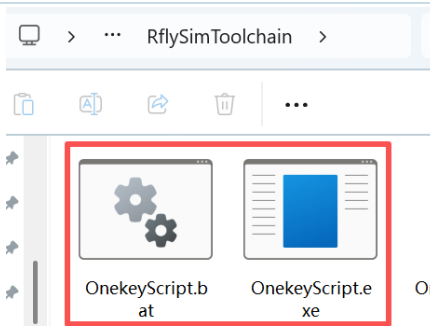
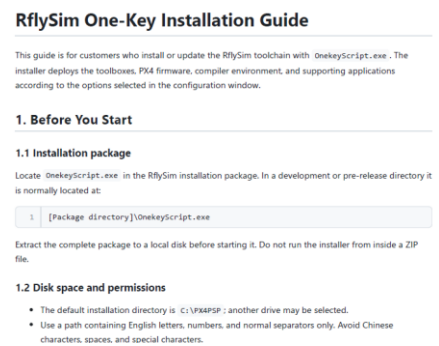


Figure: One-click installation script in operation



Script location



Help screenshot

- **Start:** Run OnekeyScript.exe. Keep first-install defaults; use an English-only path.
- **Components:** Yes = reinstall; No = unchanged; Auto = incremental update.
- **Upgrade:** Back up modified source code, then run the new installer; no uninstall needed.

Note: OnekeyScript.bat batches environment setup and silent one-click installation.

Check first-install defaults; use Help or HowToInstall.pdf for details.

2.2.2 Installation and Configuration: MATLAB Installer

Run OnekeyScript.p if MATLAB Is Installed

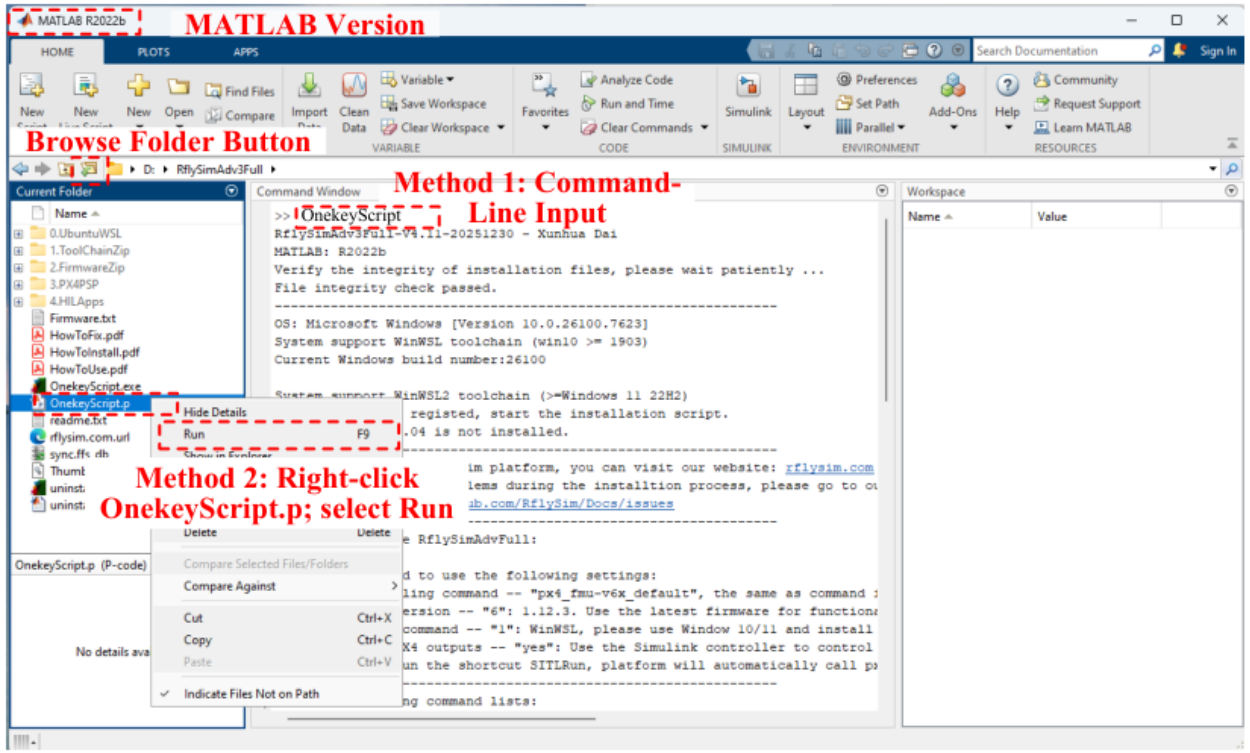


Fig. 2.3 Running the One-Click Installation Script

Supported release: MATLAB R2022b or later.

Locate: Click Browse for Folder and select the mounted image folder.

Run: Enter OnekeyScript in the Command Window,
or right-click OnekeyScript.p and select Run.

Next: The EXE configuration dialog opens.
To uninstall, run uninstall.m.

2.2.2 Installation and Configuration: Deployment Output

Installation Output: Component Locations

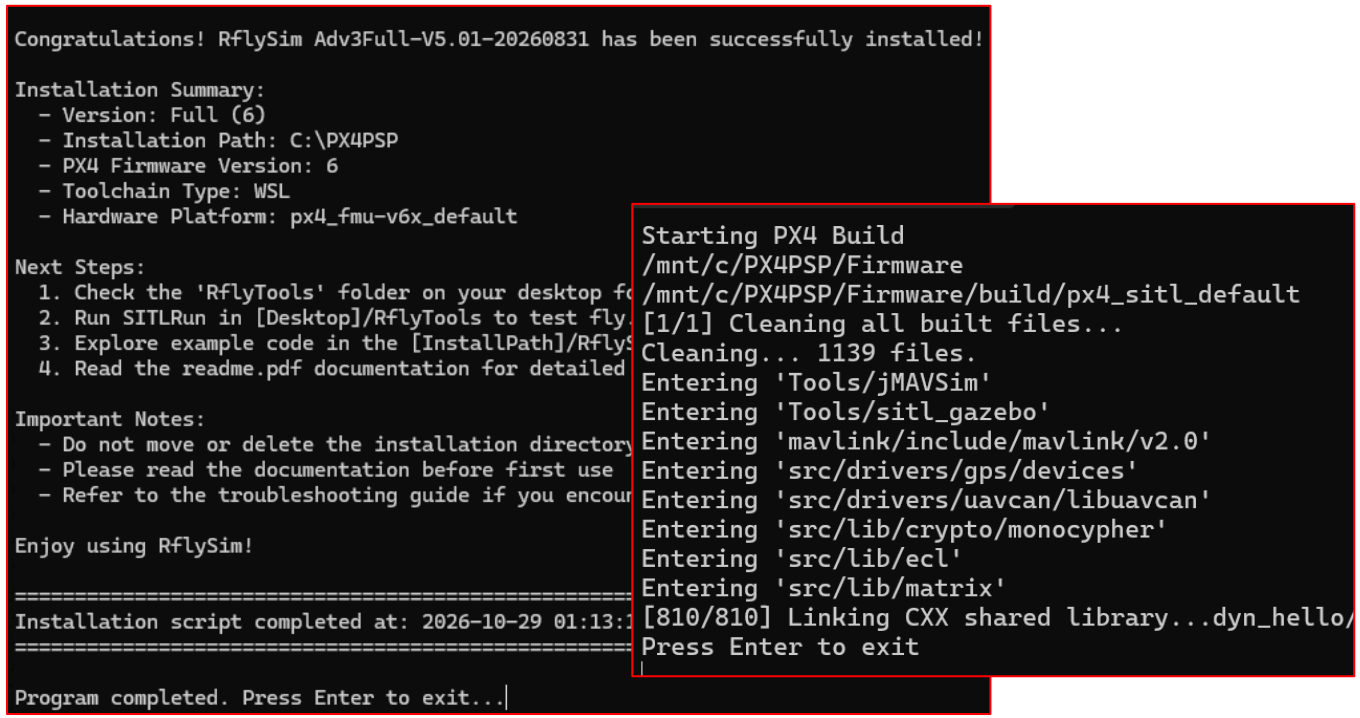


Figure: Successful installation summary (HowToInstall)

Platform installation: When the terminal reports completion, check the summary's paths, firmware version, and board model.

Firmware build: If precompilation was enabled, a separate PX4 build window opens. Wait for the target build to finish without errors and display the exit prompt; then close it as instructed.

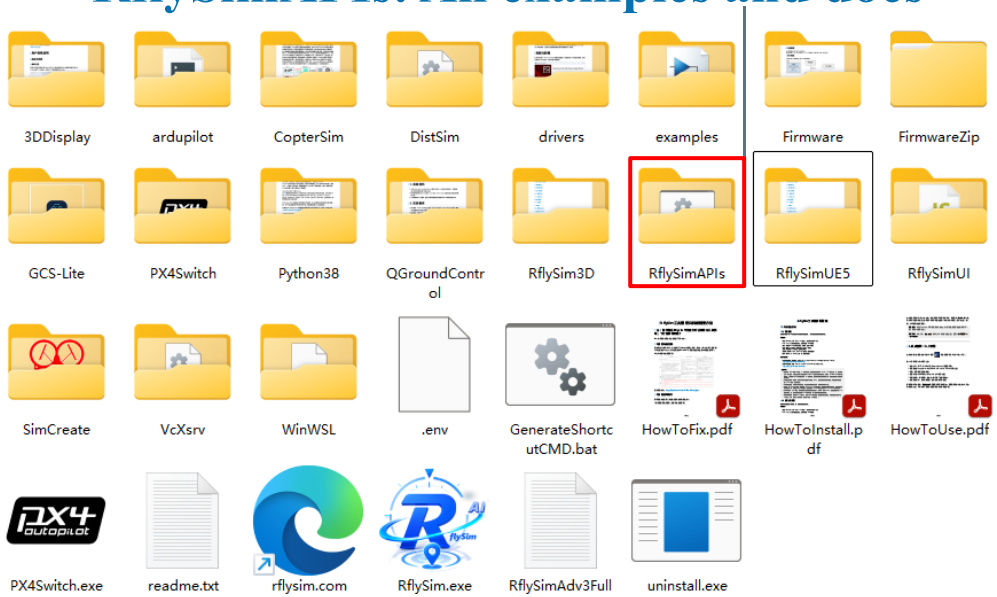
Troubleshooting: Preserve the first error and its context; consult HowToFix.pdf. If security software blocked a file, inspect its detection record.

Logs: [Documents]\Ogre\log. The first error and its context help locate the failed stage.

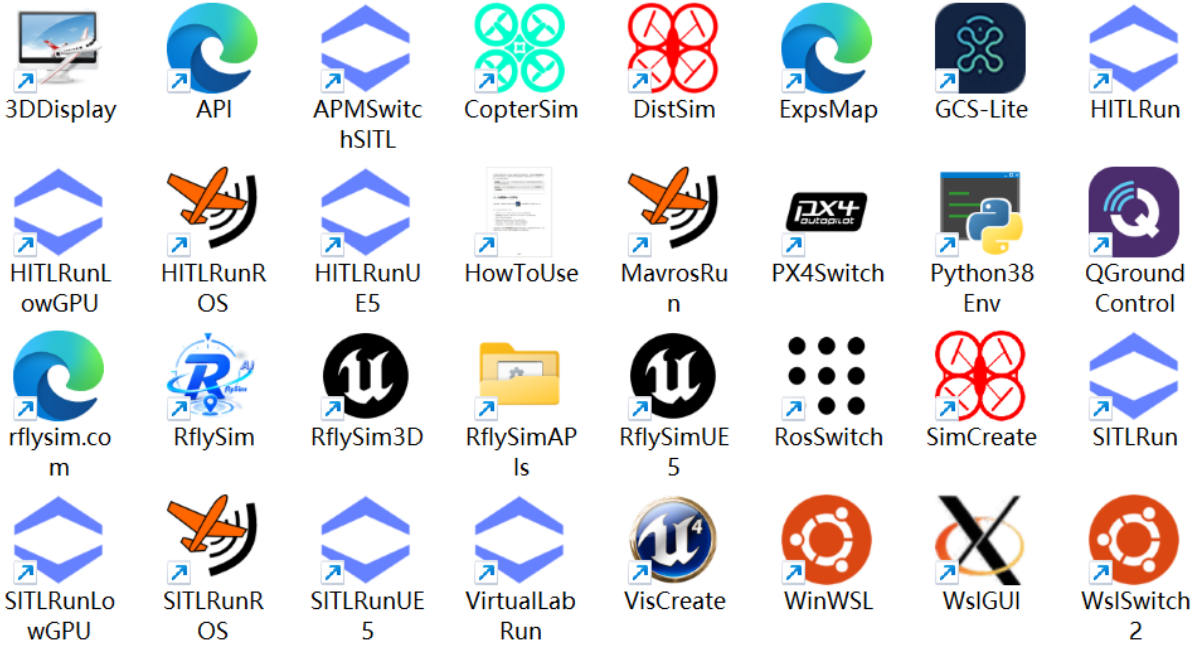
2.2.2 Installation and Configuration: Installation Results

Installation Directory and Desktop Shortcuts

RflySimAPIs: All examples and docs







Installation directory (default: C:\PX4PSP)

Desktop shortcuts

Tool shortcuts in desktop RflyTools

Install folder: tools/components. Desktop RflySim shortcut: unified launcher.

2.2.2 Installation and Configuration: WSL Versions

WSL1 and WSL2: Defaults and Switching

Comparison	WSL1 (Default)	WSL2
Architecture	System-call translation; no separate kernel	Lightweight virtual machine; full Linux kernel
File Access	Fast access to files on Windows drives	Fast within Linux; slower across systems
Networking	Uses Windows networking directly	Mirrored networking from Windows 11 22H2
GPU and Docker	Not supported	Supports AI training and containers

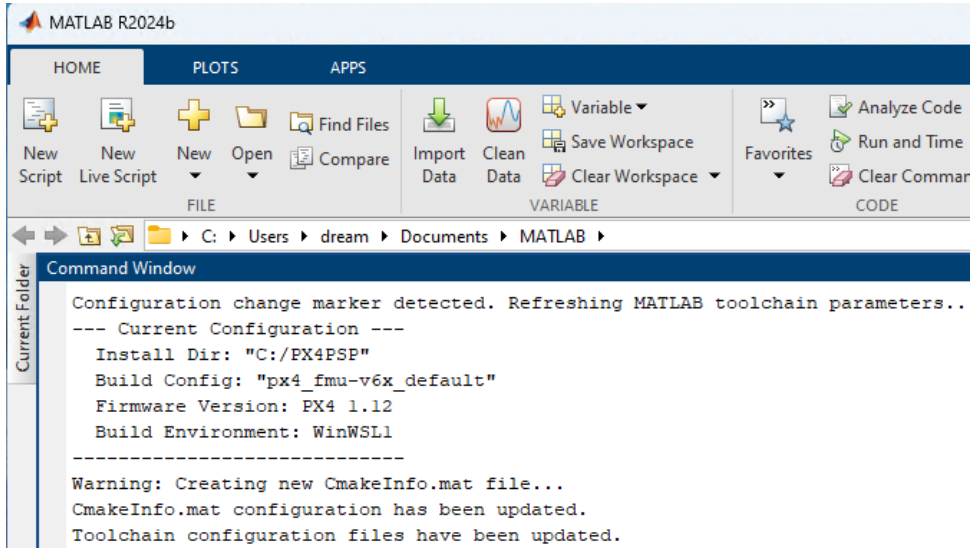
- **Check the version:** Open WinWSL from RflyTools and enter uname -a. WSL2 in the output indicates WSL2. AI = artificial intelligence.
- **Switch:** This release's WslSwitch2 requires Windows 11 22H2+ to configure mirrored networking. WSL2 itself also supports eligible Windows 10 systems.

WSL1 is the SITL build/run default; switch to WSL2 for GPU/Docker needs or build errors.

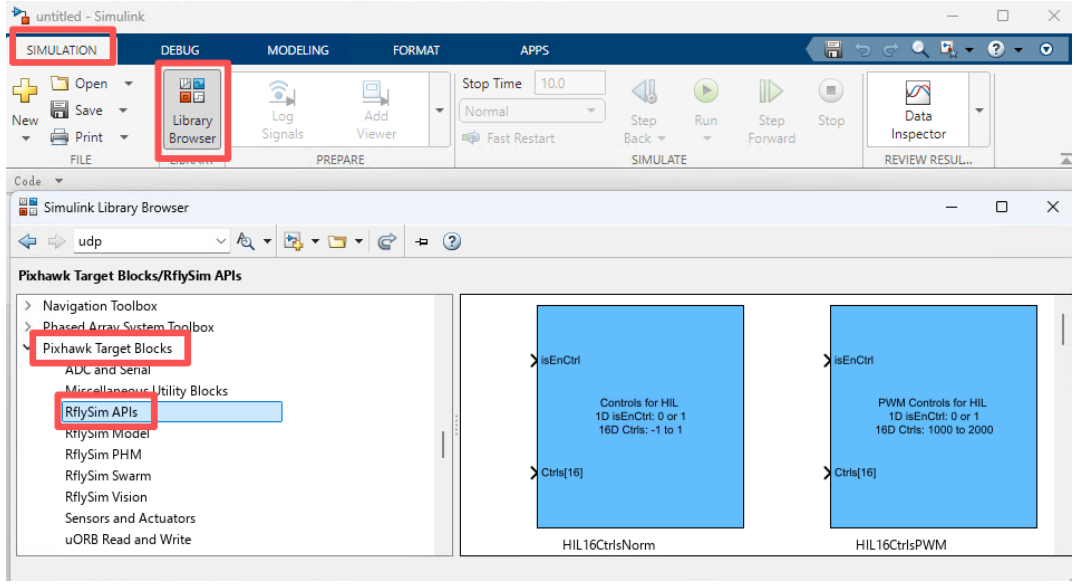
2.2 Installation and First Simulation

2.2.2 Installation and Configuration: MATLAB Integration

MATLAB: Automatic Detection and Registration



MATLAB first-launch registration



Pixhawk Target Blocks in the Library Browser

Note: First MATLAB launch after installation registers the RflySim support package. If it fails, run RflySimAPIs\RflySimSDK\simulink\RegisterMatlab.p in MATLAB from the install folder; restart MATLAB and check the block library.

After auto-registration, Pixhawk Target Blocks provides RflySim's Simulink modeling blocks.

2.2 Installation and First Simulation

2.2.2 Installation and Configuration: PX4 Firmware Selection

Firmware and Build Target



Note: Auto board detection builds and flashes matching HITL firmware.

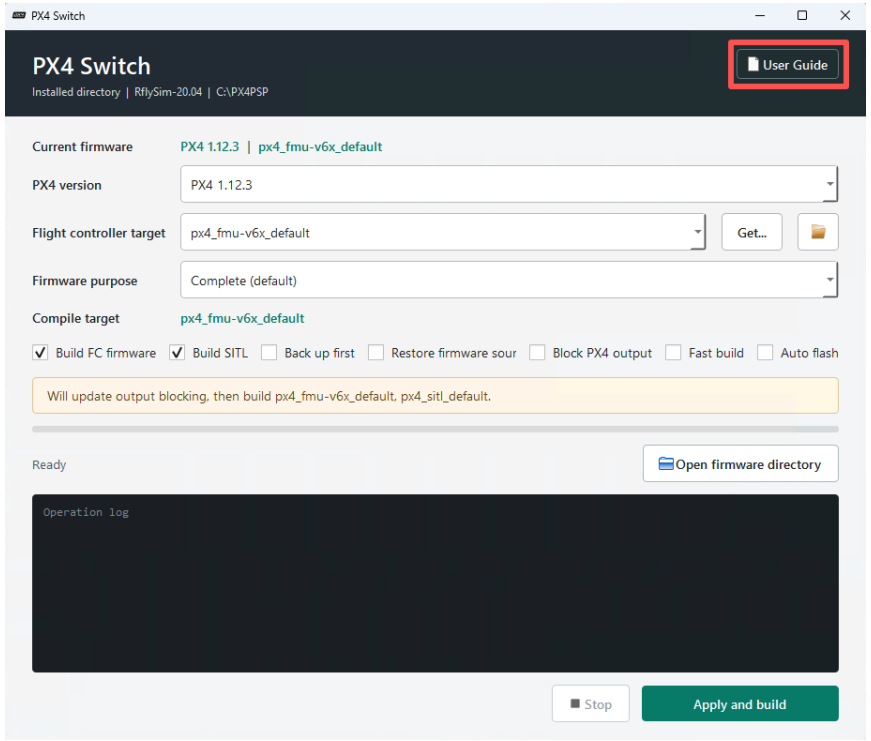


Figure: PX4Switch and User Guide entry

PX4Switch changes the local PX4 version and target board for firmware development and HITL.

Firmware: PX4 1.12 or later; choices depend on local packages. Back up modified source code before switching.

Build target: Selects the board to build for. Pixhawk 6X uses px4_fmu-v6x_default.

Result: Check the build log and firmware. Flashing occurs only with Auto Flash selected and a flight controller connected.

After installation, use PX4Switch to change firmware versions or build commands as needed.

Chapter Outline

Preparation, Installation, First Run, and Maintenance

Chapter 2
Experimental Platform Configuration

2.1 Key Tools

2.2 Installation and First Simulation

2.3 Architecture and Workflows

2.4 API and SDK Interfaces

2.5 Examples and Experimental Routes

2.6 Exercises and Discussion

4 Units in This Section

2.2.1 Preinstallation Setup
Textbook: Sec. 2.1.3

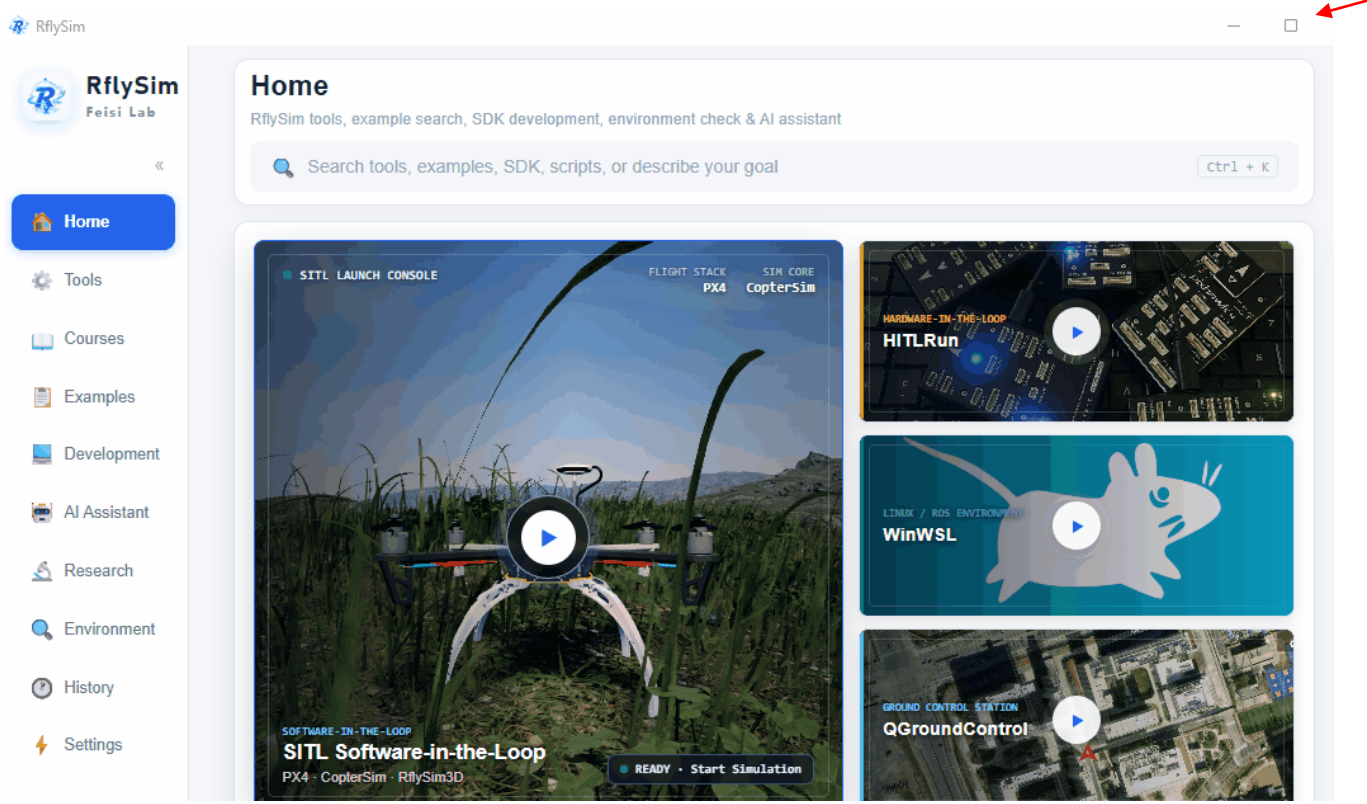
2.2.2 Installation and Configuration
Textbook: Sec. 2.1.4

2.2.3 Unified Launcher and First Simulation
Textbook: Sec. 2.1.4

2.2.4 Hardware and Maintenance Options
Textbook: Sec. 2.1.5

2.2.3 Unified Launcher and First Simulation: Main Interface

RflySim Unified Launcher and Navigation





- **Home:** Start SITL or HITL. Run Environment Check before first use.
- **Tool Navigation:** Find tools by function, read their purpose, and launch them.
- **Example Gallery:** Examples for all ten course chapters; corresponds to RflySimAPIs.
- **Custom Development:** SDK and API documentation for extending the platform.
- **Intelligent Assistant:** Ask about platform use and learning; verify answers locally.

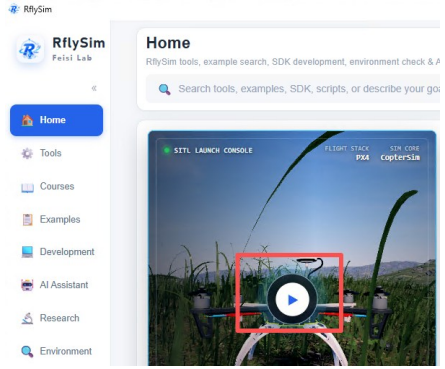
Launcher: tool/learning-resource index. Linked programs: development and simulation.

2.2 Installation and First Simulation

2.2.3 Unified Launcher and First Simulation: Six-Step Overview

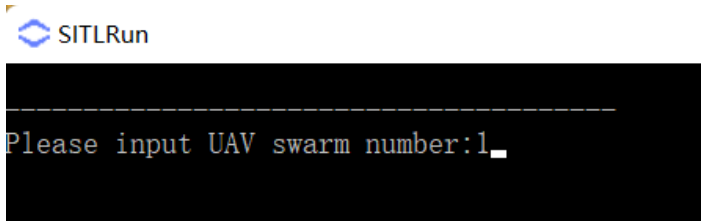
First Takeoff: Six Steps to a Stable Hover

Note: EKF = extended Kalman filter estimator; 3DFixed = 3D position ready. Help: HowToFix.pdf.





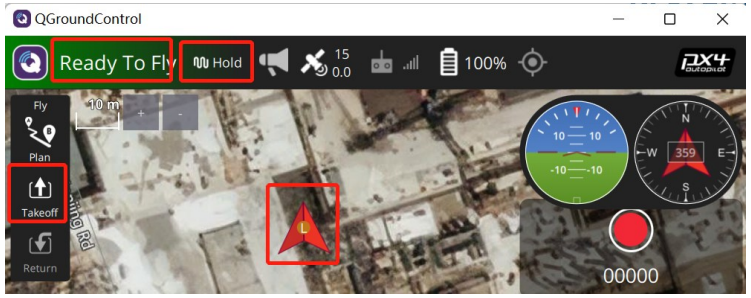
1. Run SITLRun: Home or Tool Navigation



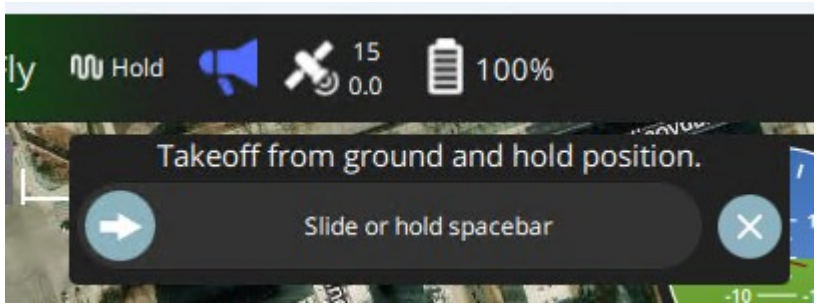
2. Enter vehicle count 1



3. Check 3D view: EKF 3DFixed



4. QGroundControl (QGC): Hold, then Takeoff



5. Drag the confirmation slider right



6. Observe takeoff and hover

Stable single-vehicle takeoff/hover shows connected control/motion/display links in this run.

2.2.3 Unified Launcher and First Simulation: First-Run Sequence

After the Windows Open, When Can Takeoff Begin?

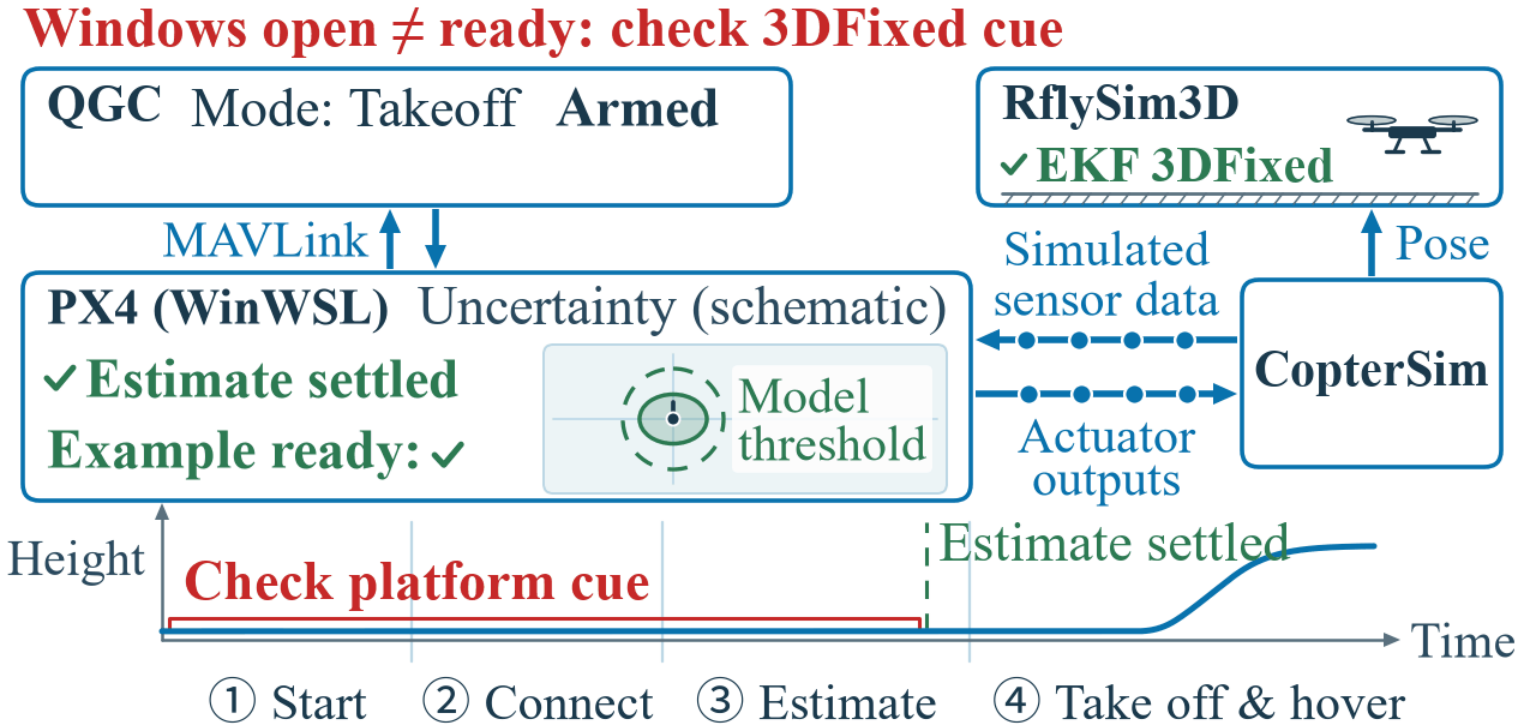


Figure: SITL first-run stages, takeoff, and hover (teaching simulation)

- **Stages:** Start, connect, estimate, then take off and hover.
- **Indicators:** 3DFixed is a platform message; ring thresholds are teaching illustrations only.
- **Action:** In Hold (position hold), select Takeoff (launch); slide to confirm.

Check the platform indicator and full preflight checks before initiating Takeoff in Hold.

2.2.3 Unified Launcher and First Simulation: Exit

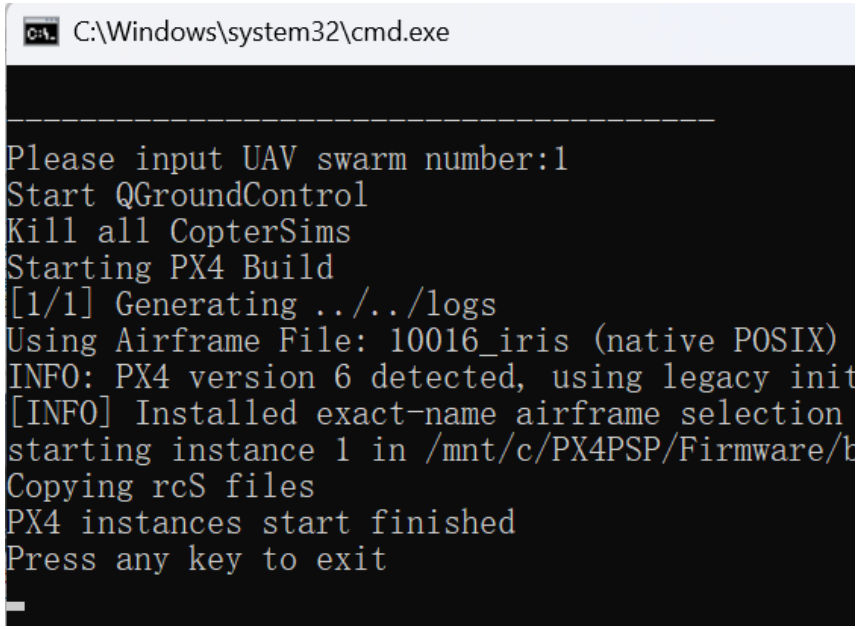
Land and Disarm Before Exit

Note: In QGC, flash compatible PX4 firmware; select an HIL (HITL) airframe. Then run HITLRun; enter the serial-port number.



Figure: Next-unit preview: starting HITL

Figure: SITL exit prompt



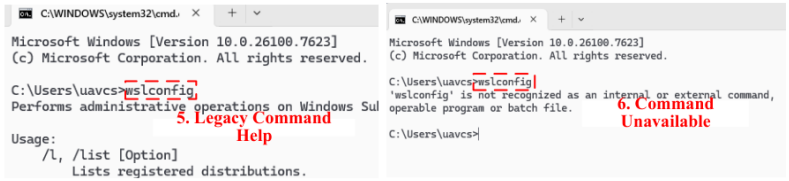
Land, disarm, then press any key to exit.

After flight, press any key in the startup terminal to end simulation and close its programs.

2.2.3 Unified Launcher and First Simulation: Troubleshooting

Trace Each Symptom to the Component

Fig. 2.2(c)(d) Legacy wslconfig output



(c) Legacy wslconfig Help Output (d) Legacy wslconfig Command Unavailable

Note: Help output does not prove readiness. Use wsl -l -v to check distributions/versions; start the required distribution to verify it.

Symptom	Check First	Check and Recovery
WSL unavailable	Linux runtime environment	Check distribution/version; start it to test operation.
QGC not connected	PX4 process and MAVLink link	Check PX4, duplicate QGC instances, and ports.
Not fully initialized	Sensor feedback and estimator	Check logs/links; allow only verified false blocks.
Port occupied	Duplicate or leftover processes	Find the owner; exit it normally, then recheck.
Choppy display	Rendering and instance load	Reduce vehicle count or use SITLRunLowGPU.

Find the failed link first; address communication, estimation, and rendering issues separately.

Chapter Outline

Preparation, Installation, First Run, and Maintenance

Chapter 2
Experimental Platform Configuration

2.1 Key Tools

2.2 Installation and First Simulation

2.3 Architecture and Workflows

2.4 API and SDK Interfaces

2.5 Examples and Experimental Routes

2.6 Exercises and Discussion

4 Units in This Section

2.2.1 Preinstallation Setup
Textbook: Sec. 2.1.3

2.2.2 Installation and Configuration
Textbook: Sec. 2.1.4

2.2.3 Unified Launcher and First Simulation
Textbook: Sec. 2.1.4

2.2.4 Hardware and Maintenance Options
Textbook: Sec. 2.1.5

2.2.4 Hardware and Maintenance Options: HITL Preparation

HITL Adds an Autopilot, RC Transmitter, and More

Note: Pixhawk 6X is recommended. Flash compatible firmware, select an HIL (HITL) airframe, and check the serial port. On a real vehicle, remove propellers and disconnect propulsion power first.

Computer Simulation

Simulate sensor readings
Send motion feedback

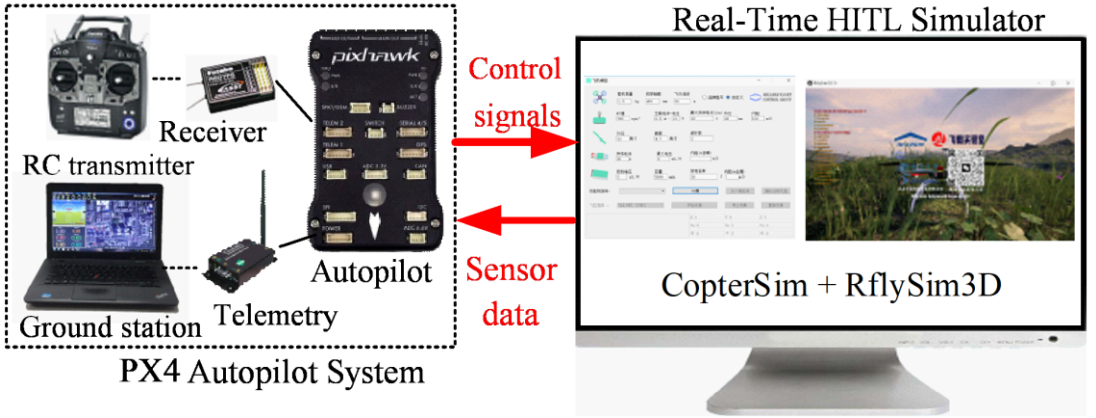
Physical Autopilot

State estimation and control
Generate actuator outputs

Computer Simulation

Use actuator outputs
to update vehicle motion

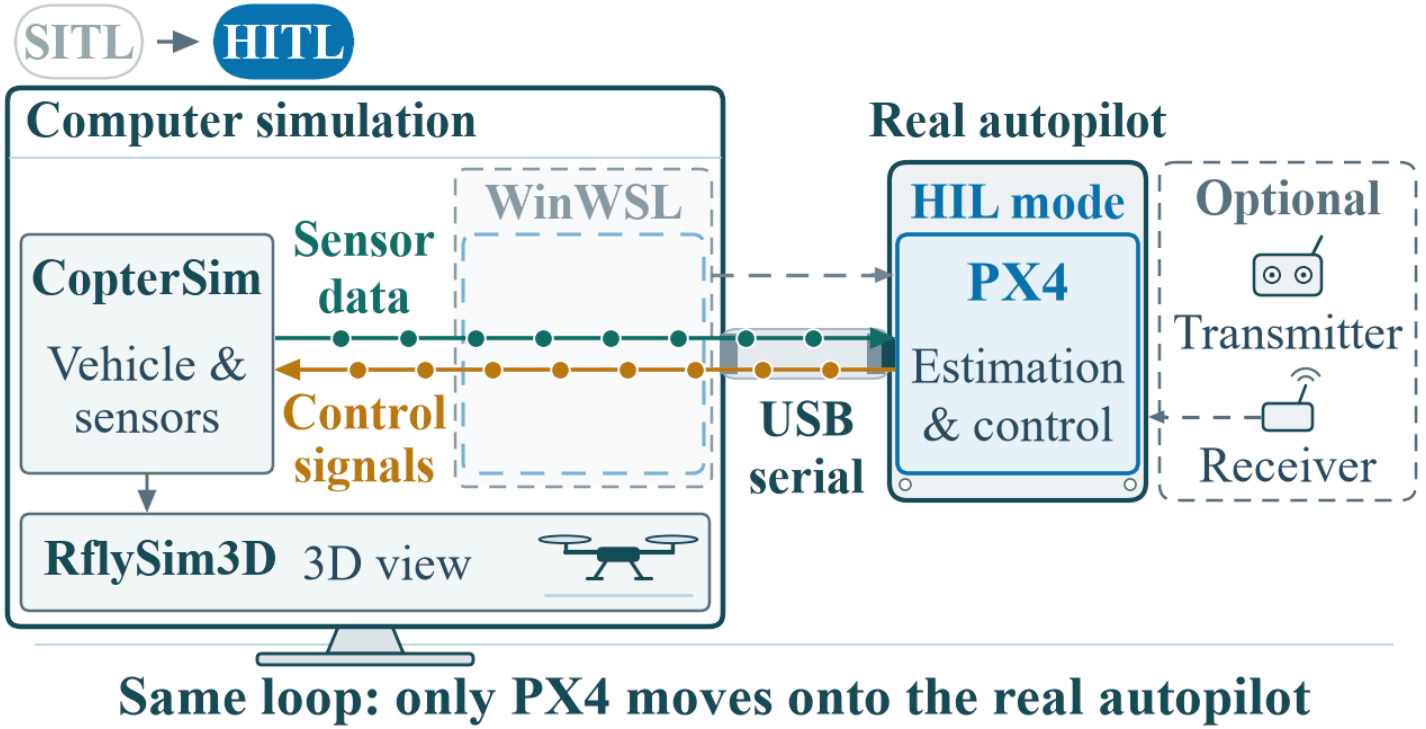




HITL: physical autopilot controls; PC simulates vehicle motion and sensor feedback.

2.2.4 Hardware and Maintenance Options: HITL Preparation

SITL to HITL: New Controller Location, Same Closed Loop



- **Location:** PX4 moves from WinWSL to the physical autopilot.
- **Loop:** Simulated sensor data goes to PX4; control outputs return to CopterSim.
- **Boundary:** The computer still simulates vehicle motion and sensor feedback.

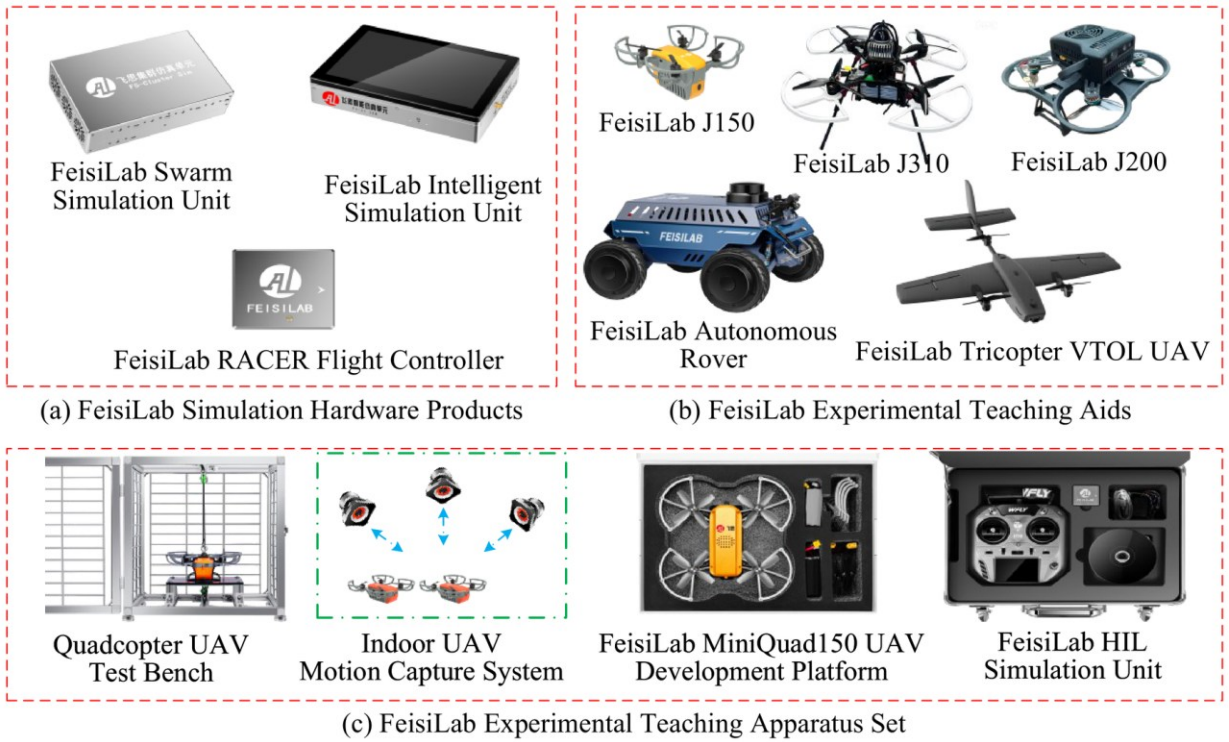
Figure: Controller location in SITL and HITL (conceptual diagram)

HITL: real autopilot controls; computer simulates vehicle and sensor feedback.

2.2.4 Hardware and Maintenance Options: Teaching Hardware Overview

Hardware: Experiment Roles

Real tests can use PX4 vehicles. Match model, control setup, and hardware interfaces; build your own or use FeisiLab teaching kits.



- **Simulation hardware (a):** Connect control boards and onboard computers to the closed loop.
- **Experimental vehicles (b):** Multirotor, ground-motion, and vertical-takeoff transition platforms.
- **Experiment kits (c):** Add test benches, positioning, development, and HITL capabilities.
- **Selection:** Define the experiment first, then match interfaces, positioning, and the site.

Fig. 2.8 Teaching-Aid Products. Details: https://rflysim.taobao.com/

2.2.4 Hardware and Maintenance Options: Uninstallation

Uninstall: Back Up First, Then Clean Up Each Part

- **Back up:** Save your work, parameters, logs, and WSL data elsewhere. Keep sn*.txt in [Documents]\Ogre.
- **Automatic uninstall:** Prefer [InstallationDirectory]\uninstall.exe. The package's uninstaller or MATLAB script is an alternative.
- **Manual toolbox cleanup:** Delete shortcuts, then remove [Documents]\MATLAB\Add-Ons\Toolboxes\PX4PSP and its path entries.
- **Manual environment cleanup:** Use wsl -l -v to check the distribution name.

Run wsl --unregister <Name> to remove it; its data is permanently deleted.
Clean temporary folders and map caches as needed.

- **Delete last:** Verify backups, then delete the actual installation directory, such as C:\PX4PSP, separately from the Documents toolbox folder.

Back up data/license files first; clean the toolbox, WSL, and installation folder separately.

Lesson Self-Check

Four Questions to Explain After the First Run

- **Why no flight controller is plugged in:** SITL runs the PX4 controller on the computer; the simulator provides the vehicle model and sensor feedback.
- **Why wait after windows open:** Process startup, component connection, and EKF initialization are separate stages. Position control needs usable state estimates.
- **How to confirm the action:** Altitude increases after takeoff, position settles into a stable hover, and ground-station readings agree with the 3D view.
- **How to exit and reproduce:** Land, confirm touchdown and disarming, exit as prompted in the terminal, and save versions, configurations, and logs.

Estimated states guide takeoff/landing; versions, configurations, and logs preserve reproducibility.

Learning Resources and Next Steps

Learning Resources and the Transition to 2.3

Installation / Upgrade / Uninstall	https://rflysim.com/doc/zh/1.Start/install.html
First Single-Vehicle SITL/HITL	https://rflysim.com/doc/zh/1.Start/quick.html
Tool Manuals and SDK Index	https://rflysim.com/doc/zh/index.html
Ch. 2 Materials and Examples	https://rflysim.com/doc/zh/2.Course/ch2/index.html

Local manuals: HowToInstall.pdf for installation, HowToUse.pdf for operation, and HowToFix.pdf for troubleshooting.

Next: 2.3 Architecture and Development Workflows

First run must show state/motion responses; open windows alone do not prove a closed loop.