AIRLAB *FPV*

User manual

A browser-based VTOL and FPV flight simulator with ArduPilot flight modes, Betaflight-style drone logic, USB radio support, a configurable OSD, and simulated radio and video links.



Version 1.2 · October 2026

Javier Ruiz-Calderón

Contents

1 What AIRLAB FPV is

2 Requirements and first launch

3 The home menu

4 The in-flight interface

5 Connecting and setting up your radio

6 Flying the airplanes: VTOL, 4+1, fixed wing and flying wing

7 ArduPlane flight modes

8 Flying the FPV drones and the multirotor

9 Rates and expo

10 ArduPilot parameters

11 The OSD

12 Radio and video links

13 Cameras

14 Scenery, time of day and weather

15 Simulated failures and failsafes

16 Learn: lessons and challenges

17 Missions in AUTO mode

18 Photogrammetry and remote sensing

19 Analysis: black box and real logs

20 Sound and voice

21 Graphics and performance

22 Using it on your phone

23 Keyboard shortcuts

24 Troubleshooting

25 Glossary

26 Step-by-step guide: ELRS MAVLink telemetry over WiFi

27 Connections, real ArduPilot (SITL), phone viewer and remote instructor

28 Work missions

29 Pilot logbook, level and certificates

30 Sailplane: aerotow, thermals, slope and variometer

31 ENAIRE UAS zones: legal planning and in-flight warnings

32 Realistic battery and battery failsafe

33 Variable weather during the flight

34 Real terrain anywhere in the world

35 Sailplane contests

36 EASA/AESA drone regulations

37 Real world: weather, traffic and slope

38 Vertical clip for Instagram

39 Operations centre: drone in a box

40 Setting up the Stallion VTOL in ArduPilot

> **How to read this manual**
>
> If this is your first time, read chapters 2, 3, 5 and 6, then work through the lessons in chapter 16 in order. The rest is reference material: each chapter can be read on its own.

1 What AIRLAB FPV is

AIRLAB FPV is a flight simulator that runs in your web browser. It reproduces airplanes using the logic of ArduPlane flight modes (a tilt-rotor VTOL that takes off vertically, tilts its motors forward and flies like an airplane; 4+1 QuadPlanes; a tailsitter that takes off sitting on its tail; fixed wings and flying wings from the Flightory range) plus two FPV drones with Betaflight-style flight logic and three work multirotors (inspection, tethered cleaning and caged) with ArduCopter modes. The entire interface and this manual are available in English and Spanish.

Included aircraft

Aircraft	Type	Flight logic	What to practice
Stallion VTOL	Three-motor tilt-rotor: two tilting front motors and one rear motor	ArduPlane modes (QSTABILIZE, QHOVER, QLOITER, FBWA, AUTO...)	Vertical takeoff, transition to airplane flight and back, failsafes, missions
QuadPlane 4+1	Four lift motors in an X layout and a forward-thrust motor in the nose, on the same wing	ArduPlane modes with QuadPlane	The transition of a "separate-motor" VTOL, the most common type
Fixed wing	Twin tractor-motor airplane with a V-tail, no VTOL	ArduPlane airplane modes, with TAKEOFF	Hand launch, automatic takeoff and belly landing
Super Stingray	Swept flying wing with winglets, a pusher and elevons, no rudder	ArduPlane airplane modes, with TAKEOFF	Flying-wing flight: critical center of gravity, elevons, hand launch
Super Stingray VTOL	The same wing in a 4+1 layout, with two carbon booms	ArduPlane modes with QuadPlane	A light, fast 4+1 with the parameters from the Flightory manual
Mini Plank	650 mm straight flying wing with tip fins and a pusher	ArduPlane airplane modes, with TAKEOFF	A light, lively park flying wing
Lark	Conventional tail on a carbon boom, pusher behind the wing	ArduPlane airplane modes, with TAKEOFF	Long, calm flights with ailerons, elevator and rudder
pico Talon	Compact 905 mm V-tail with a pusher	ArduPlane airplane modes, with TAKEOFF	Light FPV: a V-tail mixing rudder and elevator
Talon 1400	1.3 m V-tail with a pusher	ArduPlane airplane modes, with TAKEOFF	A very stable, trim-free airplane for long missions
Stork	1.62 m, aspect-ratio 8.3 V-tail with a pusher	ArduPlane airplane modes, with TAKEOFF	Efficient gliding, mapping missions
Stork VTOL	The same Stork as a 4+1, with two 500 mm booms	ArduPlane modes with QuadPlane	A high-aspect-ratio V-tail 4+1
Moose	1.6 m twin tractor on the wing with a V-tail	ArduPlane airplane modes, with TAKEOFF	A heavy cargo airplane (up to 4.5 kg)

Riflebird	Flying-wing tailsitter with two fixed motors	ArduPlane modes with tailsitter (Q_TAILSIT_*)	Tail-sitting takeoff, tailsitter transitions
Mario 5	5" freestyle FPV drone, 6S	Betaflight-style: ACRO, ANGLE and HORIZON	Acrobatic flying, rates, throttle
Mario Fold 8	8" long-range FPV drone, 6S2P Li-ion	Betaflight-style, with GPS Rescue	Long-distance flying, video link, return to home
Inspection multirotor	Work quadcopter of about 4 kg, 15" propellers, 6S 16,000 mAh, RGB and thermal gimbal	ArduCopter modes: STABILIZE, ALT_HOLD, LOITER, FLOWHOLD, RTL, LAND and AUTO	Close inspection with position hold, AUTO surveys, flying without GPS, work missions
Tethered cleaning drone	Heavy multirotor of about 60 kg with a hose to a truck and a water lance	ArduCopter modes	Blade cleaning: hose weight, wind and jet thrust
Caged drone	2.4 kg drone inside a 52 cm cage, with lights	ArduCopter modes, no GPS	Confined spaces: tanks, boilers, interiors

What it simulates

- **6-degree-of-freedom physics** with aerodynamics calculated from the aircraft geometry: lift, drag, stall, ground effect and turbulence.
- **Battery** with per-cell voltage, sag from internal resistance, mAh consumption and estimated remaining flight time.
- **Controllers** that mimic the behavior of ArduPilot and Betaflight (they are not their actual code).
- **Real ArduPilot (SITL)**: with AIRLAB Connector, the real ArduPlane flies the Stallion VTOL or the fixed wing with the simulator's physics, with the test `.param` or yours (chapter 27).
- **Work missions**: twelve real jobs (thermal search and rescue; inspection of a power line, a dam, wind turbines, an aircraft in a hangar with damage chart, a solar plant, a tank, a telecom tower and a bridge; blade cleaning with a tethered drone; stockpile volume and wildfire hotspots), with a gimbal camera, a score and a PDF report (chapter 28).
- **Radio link** (LQ and RSSI) and a digital or analog **video link**, both of which degrade with distance and obstacles.
- **Weather**: wind with a height profile, gusts, turbulence and wake behind obstacles; time of day, clouds, fog and rain.

> **Important**
>
> AIRLAB FPV is for practicing modes, transitions, recoveries and flight routines. Its coefficients are estimates and its controllers are imitations: it does not replace real-world testing and it does not guarantee that your real aircraft will behave the same way. With real ArduPilot (chapter 27) the autopilot is the real one, but the physics is still a model.

2 Requirements and first launch

What you need

- **Browser**: an up-to-date Chrome or Edge on a computer. These are the browsers that read your radio over USB (Gamepad API). Firefox and Safari work with a keyboard or gamepad, but your radio may not be detected reliably.
- **Graphics card** with WebGL. Any computer from the last few years will do; on modest machines, turn on the lightweight scenery (chapter 21).
- **Radio** (recommended): an EdgeTX or OpenTX radio connected over USB in joystick mode. A game controller or the keyboard also works.
- **Phone or tablet** (optional): fly with on-screen touch sticks or a Bluetooth controller (chapter 22).

Opening the simulator

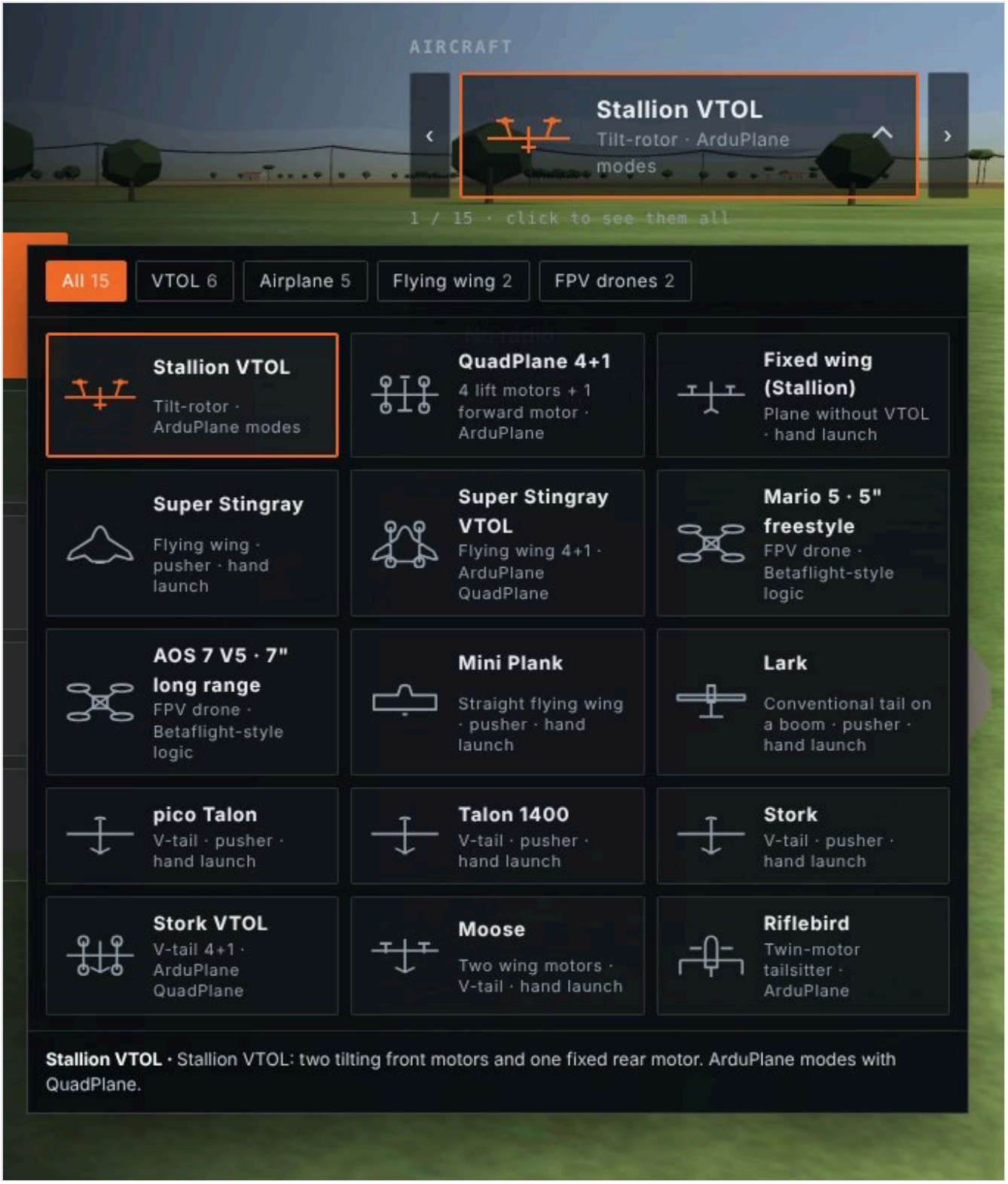
1. Sign in to the website with your account and click **Open the simulator**. The simulator can only be open on one device at a time; if you open it on another, the first one closes.
2. You'll see the logo with a loading bar while the 3D scenery is prepared.
3. The **home menu** appears. The first time, the **interactive guide** starts and walks you through every part of the program in a couple of minutes.
4. Connect your radio over USB and move a stick so the browser detects it (chapter 5).

> **Your settings are saved automatically**
>
> Radio mapping, parameters, rates, OSD, voice and all other options are saved in this browser. If you switch computers or browsers, you'll need to set them up again.

3 The home menu



The home menu. Under “Free flight”, the scenery selector; in the background, the selected aircraft rotates above the flying field; at the bottom right, Captain Pitot’s chat.



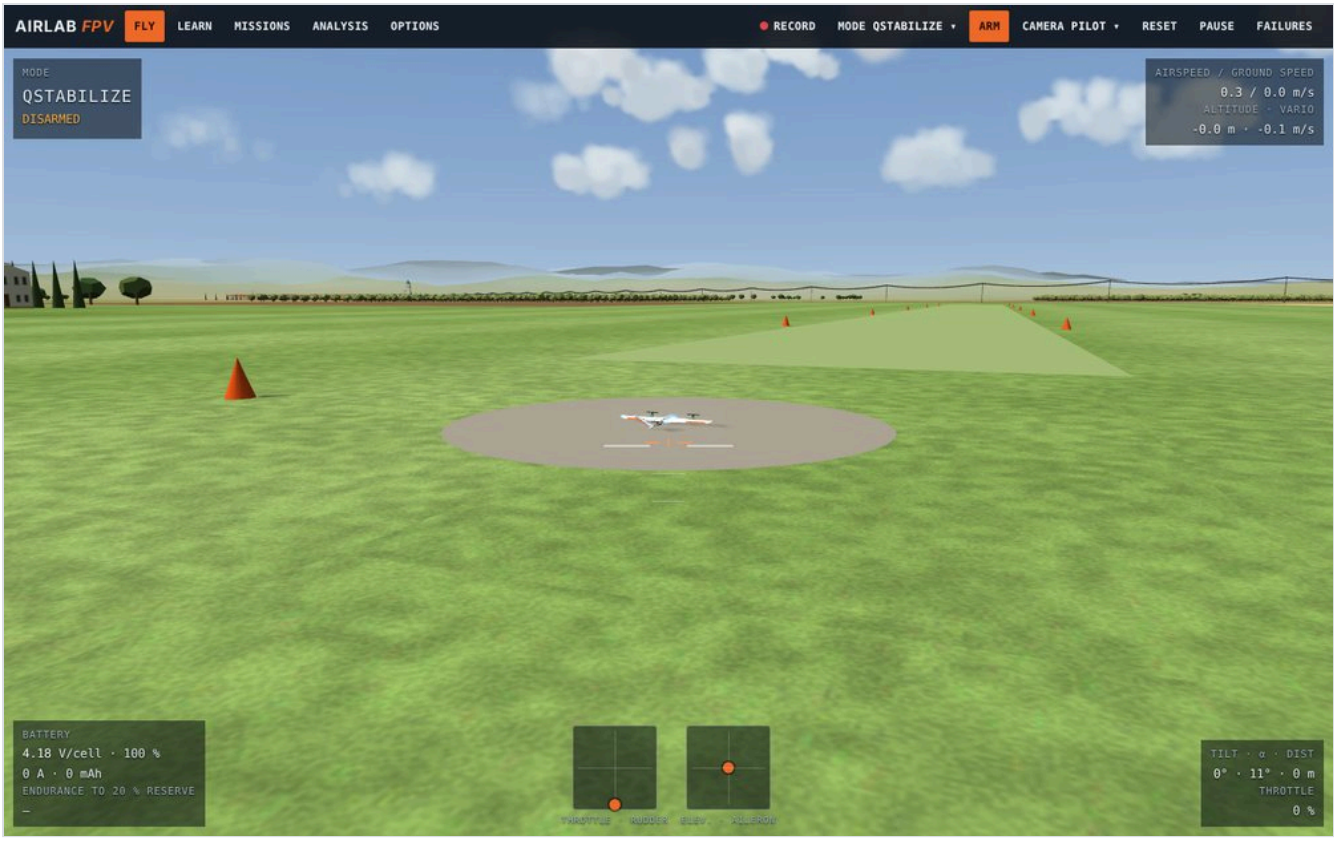
The aircraft selector opened, with the type filters and the details of the aircraft under the mouse.

Item	What it does
Free flight	Starts flying with the current setup. Below it is a summary of the aircraft, scenery, time of day and wind. You can also press `Enter`.
Scenery	Drop-down under “Free flight”: pick where you fly (Osuna countryside, coastal cliff, mountain and valley, river canyon, quarry, or dam and reservoir). Chapter 14.

Lessons	Five step-by-step guided lessons for the Stallion, with your progress.
Challenges	Timed tests scored from 0 to 100, with your personal best.
Missions	Drone jobs with a brief, a score out of 100 and a report.
Plan	Mission planner for AUTO mode, with UAS zones.
Options	Radio, aircraft, rates, OSD, scenery, weather, graphics and sound.
Aircraft	A drop-down selector with the 18 aircraft. The card shows the selected one; the ‹ › arrows take you to the previous or next, and clicking it opens a menu with all of them (Stallion VTOL, QuadPlane 4+1, fixed wing, Super Stingray fixed wing or VTOL, Mini Plank, Lark, pico Talon, Talon 1400, Stork fixed wing or VTOL, Moose, Riflebird, Mario 5 and Mario Fold 8), with filters by type (VTOL, airplane, flying wing and FPV drones) and the details of the one under the mouse. It also works with the keyboard (arrows, Enter and Esc). The one in the background changes instantly.
Captain Pitot	At the bottom right, the instructor's chat: ask him about any feature of the simulator (chapter 19).
Radio	Live radio status. If it's connected but not set up, it offers the setup wizard.
Manual, guide and language	Download this manual as a PDF, replay the interactive guide or switch the simulator to Spanish (the language button, **Español** in this version). Your language choice is remembered.

During flight, `Esc` takes you back to the home menu and pauses the simulator. Clicking the AIRLAB FPV logo in the top bar does the same.

4 The in-flight interface



Flight screen with the Stallion in pilot camera and the classic “AirLab FPV” OSD.

The top bar

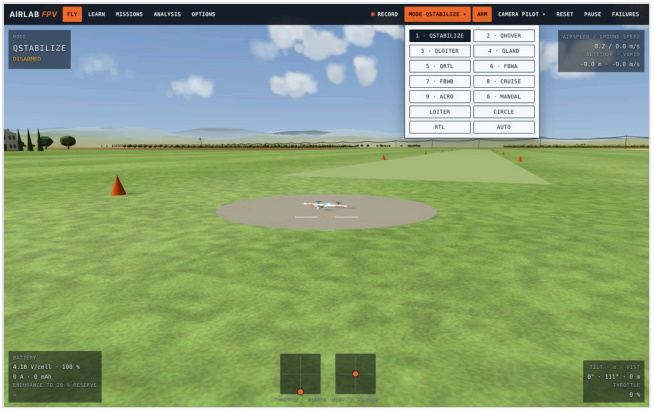
On the left are the program sections, in the order a pilot works. Each one answers a question:

Tab	What for	What is inside
Fly	I want to fly now	The flight, with modes, camera and failures in the bar.
Learn	I want to learn	Lessons, drone regulations (theory, practice and mock exam), challenges, sailplane contests and Captain Pitot (live instructor, custom missions and training plan).
Missions	I want a job with a goal and a score	Work missions and remote sensing exercises.
Plan	I want to prepare a flight	AUTO mission planner with the ENAIRE UAS zones, and the photogrammetry and remote sensing survey.
Analysis	I want to review what I flew	Black box, event log, remote sensing lab and logbook with certificates.
Connect	I want to connect something	AIRLAB Connector, real ArduPilot (SITL), real flight with instructor, phone viewer and remote instructor.
Options	I want to configure	Radio, aircraft and parameters, rates, OSD, scenery, weather and battery, graphics and sound.

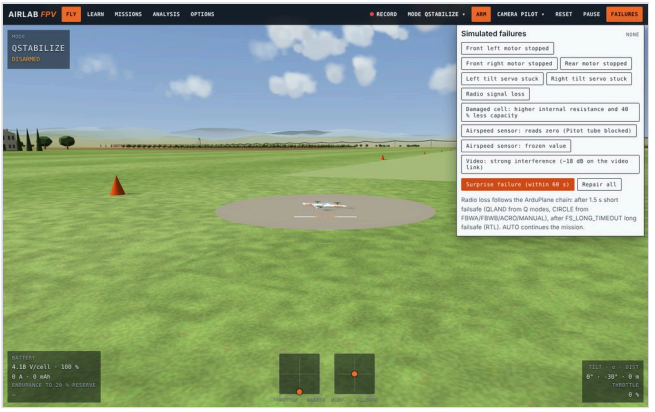
At the top of each tab, buttons take you to each section without scrolling. When you leave Fly, the simulator pauses automatically, and it resumes when you come back.

On the right, only in Fly, are the flight controls:

Button	Function
● **Record**	Records the flight to video (next section). Key `V`.
Mode ▾	Opens the list of the aircraft's modes. The active mode is shown on the button.
Arm / Disarm	Arms or disarms the motors. Red when armed. Key `Space`.
Camera ▾	Pilot, chase, FPV or free. Press `C` to cycle through them.
Reset	Returns the aircraft to the pad, disarmed and with a full battery. Key `R`.
Pause	Freezes the simulation.
Failures ▾	Opens the simulated failures panel. If any failure is active, it's flagged in the bar.

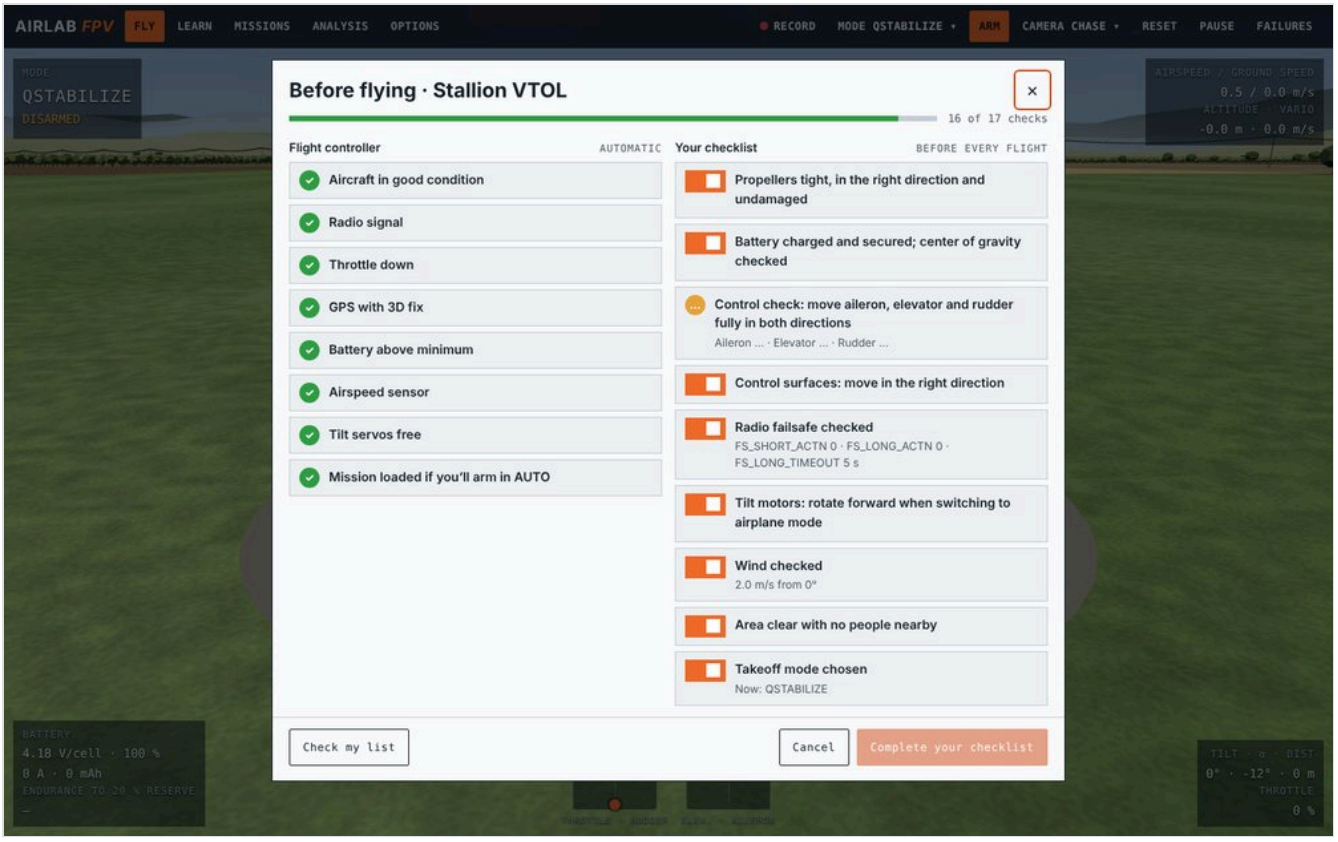


Mode menu. The numbers are the shortcut keys.



Simulated failures panel.

Pre-arm checks



Clicking Arm brings up the “Before flying” list: on the left, what the flight controller checks; on the right, your checklist.

Just like a real flight controller, the simulator won’t arm if something isn’t right. Clicking **Arm** opens the **Before flying** window, with two columns:

- **Flight controller (automatic)**: aircraft undamaged, radio signal, throttle down, GPS with a 3D fix (on startup it takes a few seconds to see 6 satellites), battery above 3.6 V per cell, airspeed sensor, tilt servos (tilt-rotor only) and a loaded mission if you’re arming in AUTO. If something fails, you’ll see the message exactly as ArduPlane would show it (“PreArm: …”), or as Betaflight would on the drones.
- **Your checklist**: propellers, battery and center of gravity, control check (ticked automatically when you move aileron, elevator and rudder to full deflection in both directions), control surfaces, radio failsafe with your `FS_SHORT_ACTN` and `FS_LONG_ACTN` values, tilt motors, current wind, clear area and takeoff mode. **Check my list** ticks them all at once.

The **Arm** button becomes active when everything is green. Your checklist is required **before the first flight** each time you open the simulator; after that only the flight controller checks apply. In Options › Radio › **Arming checks** you can choose flight controller + your checklist before the first flight, flight controller + your checklist before every flight, flight controller only (an on-screen and voice alert tells you what’s failing) or none.

If your radio’s arm switch is already up when you open the simulator, it won’t arm: flick it down and up again. Keyboard shortcuts (such as `Space` to arm) don’t act while you are typing in a text field.

Recording your flight to video

Click ● **Record** in the bar (or press `V`; on a phone, the ● **REC** button). While recording, the button shows the elapsed time. Click it again and the video downloads automatically with a name like `AIRLAB-FPV-`

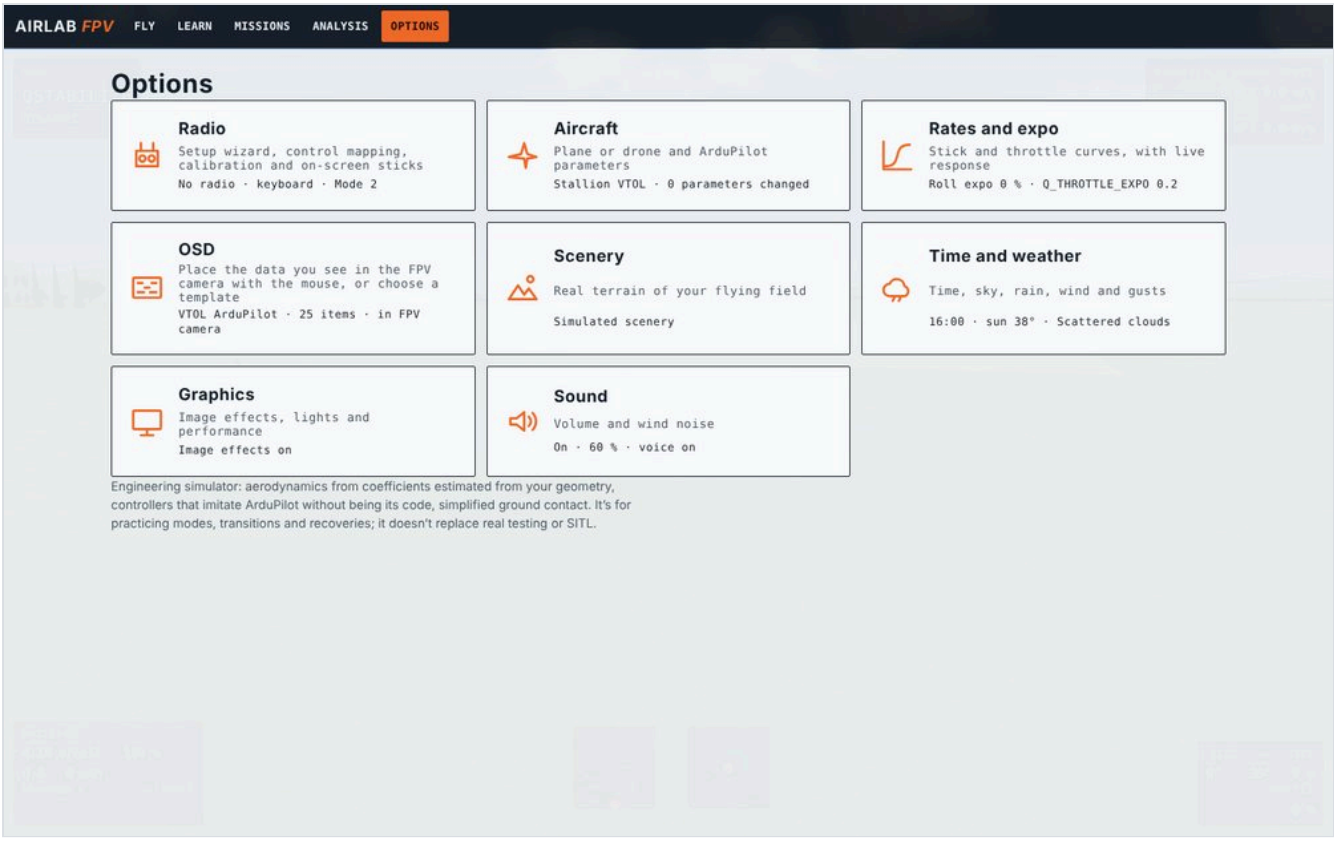
`20261014-183005.mp4`.

- What you see is what gets recorded: the 3D view with video-link effects, the OSD, alerts, on-screen sticks and the sound of motors and wind. The voice is not recorded because the browser doesn't allow it.
- MP4 format if your browser supports it (recent Chrome and Edge, Safari); otherwise WebM.
- In Options › Graphics you choose the quality (up to 1080p at 10 Mb/s, or 720p at 6 Mb/s) and whether to include the AIRLAB FPV watermark in the corner.

The OSD and on-screen sticks

The OSD is overlaid on the image: mode, speeds, altitude, battery and any other data you choose (chapter 11). At the bottom center you can display two boxes showing your stick positions, handy for checking your radio and for recording videos (Options › Radio).

Options



Options: each card summarizes how that section is configured.

5 Connecting and setting up your radio

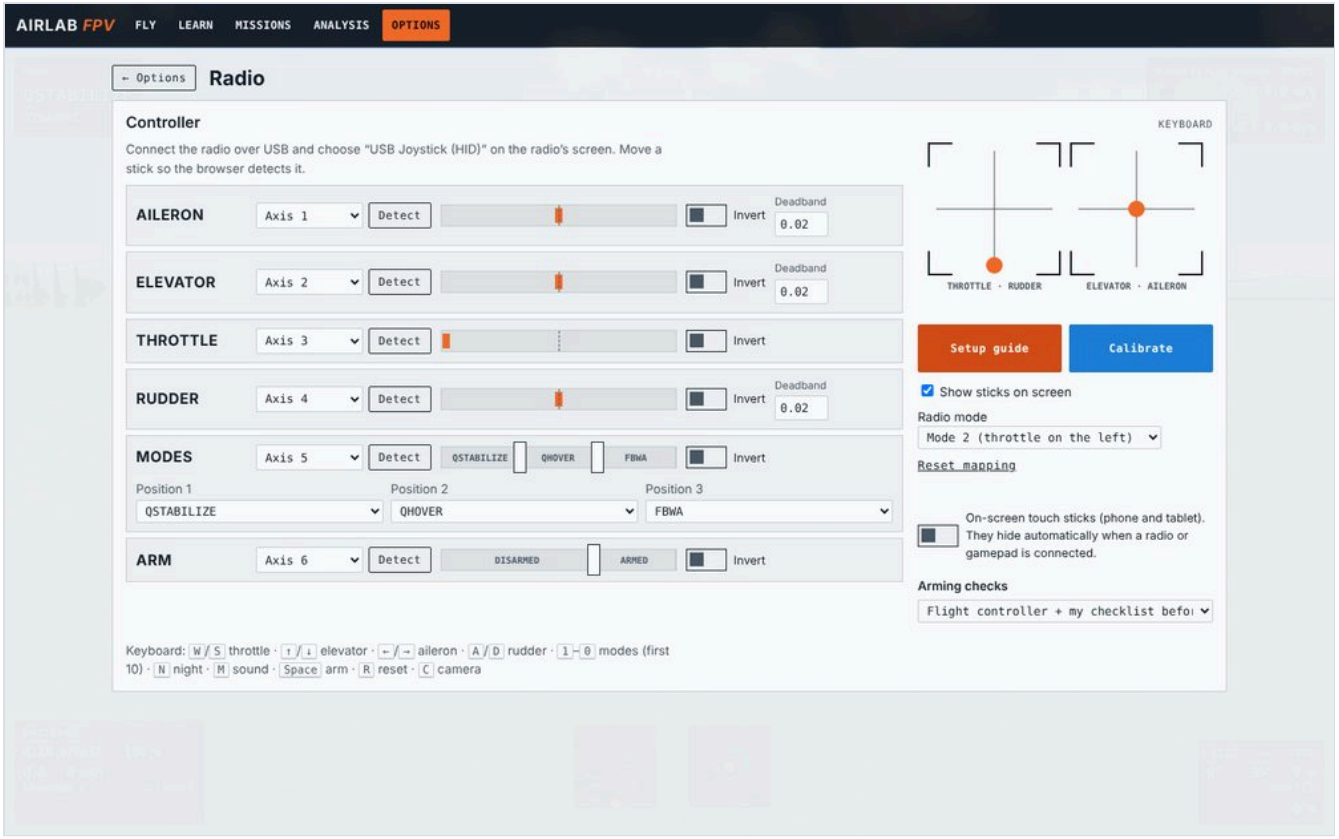
Preparing your EdgeTX radio

1. Connect the radio to your computer with a USB data cable.
2. On the radio screen, choose **USB Joystick (HID)**.
3. In the model, under **SYS › USB Joystick** (or in the model settings, depending on the version), leave the mode set to **Classic** so it sends the channels as they are, or set up **Advanced** mode by assigning each channel as **Axis** or **Btn** (button).
4. Move a stick with the simulator open: the browser only discovers the radio once it receives input.

> **Channels you need**
>
> Aileron, elevator, throttle and rudder, a 3-position switch for modes and a switch for arming. If arming is set up as a button (Btn) in EdgeTX, the simulator accepts that too.

The setup wizard

This is the fastest way. Open it from the home menu or from Options › Radio › **Setup guide**. It asks you to move, one at a time, the throttle, rudder, elevator, aileron, arm switch and mode switch; it detects which axis or button each one is and whether it needs inverting, and finishes with travel calibration.



Options › Radio: channel mapping, switch ranges, sticks and calibration.

Manual setup

Column	What it's for
Channel and **Detect**	Choose the axis or button for each function, or click Detect and move that control for 4 seconds.
Bar	Shows the processed position of the control live.
Invert	Reverses the direction. Elevator usually needs it: pushing the stick forward should lower the nose.
Deadband	Aileron, elevator and rudder. Ignores small movements around center (0.02 by default) and rescales the rest so there are no jumps.
Switch ranges	Drag the limits to decide which switch position activates each mode or arming. The black mark is the current position.
Position 1, 2 and 3	Which flight mode each position of the mode switch activates.

Calibrate

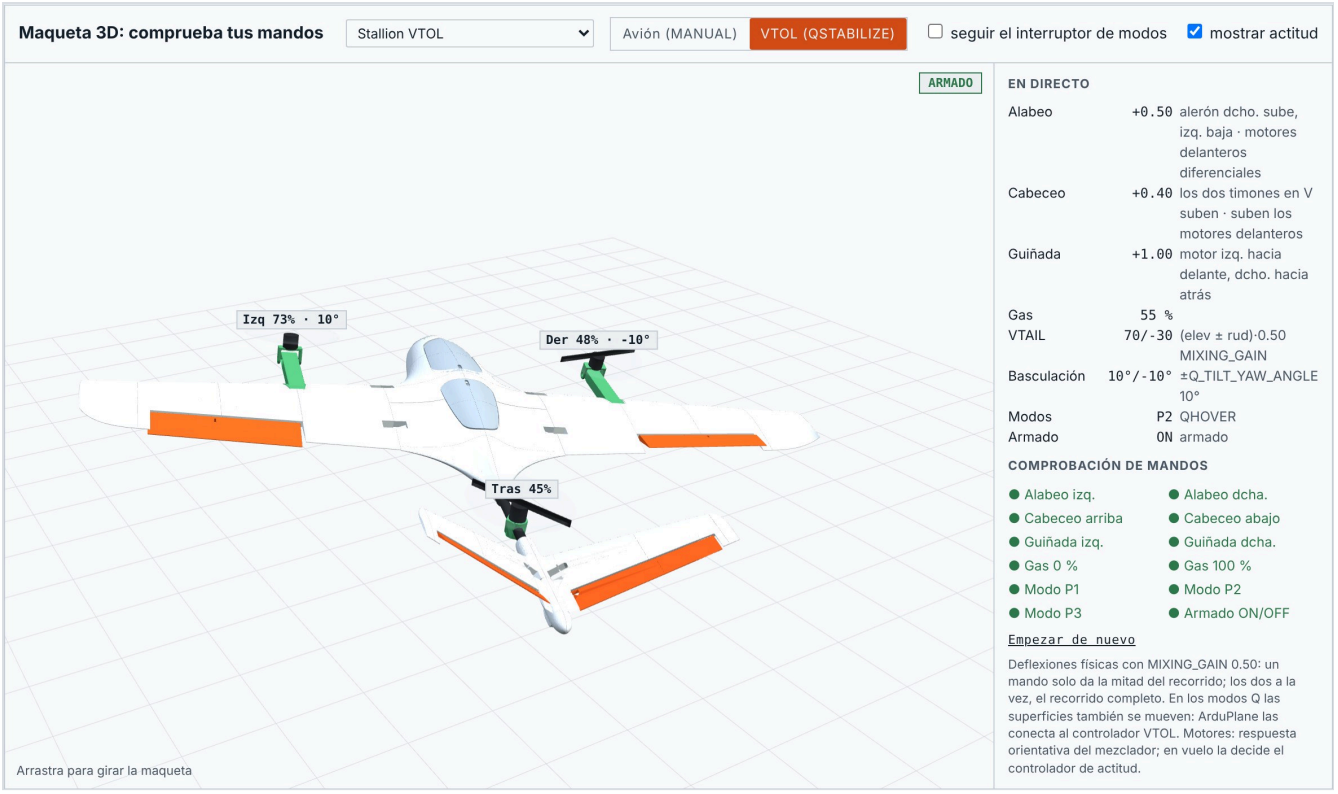
Click **Calibrate**, move all sticks and controls to their endpoints several times, then click it again. This fixes radios whose axes don't reach ±1.

Radio mode

Choose Mode 2 (throttle on the left, the most common) or Mode 1 (throttle on the right). This affects the on-screen sticks and the touch sticks.

3D preview: check your controls

Below the channel assignment there is a 3D aircraft that moves with your sticks. It lets you do in the simulator the same check you do at the field with the aircraft in your hands (props off): that each control moves the right surface or motor, in the right direction. It is the same 3D model that flies in the simulator: pick any of its aircraft at the top (Stallion tilt-rotor, 4+1, flying wings, V-tails, twin, tailsitter, drones…) and, for VTOLs, the mode: **Plane (MANUAL)** or **VTOL (QSTABILIZE)**. With **follow the mode switch** ticked it switches to VTOL by itself when the switch position selects a Q mode. The preview stays still and only moves with your inputs; drag with the mouse or a finger to see it from another angle.



3D preview of the Stallion in VTOL with right roll, pitch up and right yaw: the front motors tilt ±Q_TILT_YAW_ANGLE in opposite directions, the V-tail mixes elevator and rudder with MIXING_GAIN and the checklist ticks itself.

Input	What you should see
Roll right	Right aileron up, left down. In VTOL the left motor also speeds up (tilt-rotor and 4+1). On the tailsitter in hover, roll is done with the motors: the left one speeds up.
Pitch up (stick back)	Both V-tail panels (or both elevons) go up. In VTOL the front motors speed up and the rear one slows.
Yaw right	Right V-tail panel down, left up (with a conventional tail, the rudder to the right). On the tilt-rotor in VTOL the left motor tilts forward and the right one back. On the 4+1 and the drone the counter-clockwise props speed up (orange; clockwise ones are blue). On the twin, the left motor speeds up by RUDD_DT_GAIN %.
Throttle	In MANUAL it drives the forward motors; in VTOL, all the lift motors.

The mixer is ArduPlane's: V-tail = (elevator ± rudder)·MIXING_GAIN and elevons = (elevator ∓ aileron)·MIXING_GAIN, saturated at full travel. With MIXING_GAIN = 0.5 (default) one input alone gives half deflection and both together full deflection, so it never saturates. In Q modes the surfaces also move, because ArduPlane slaves them to the VTOL controller. Motor percentages are the indicative mixer response: in flight the attitude controller decides.

Arming

Outside the flight screen (Options, Learn, Missions...) the radio switches do not arm the simulator aircraft or change its mode; only the preview reflects them. Disarming does work. While the pre-flight checklist is open they don't change the mode either: if you launch a mission it stays in AUTO even if you move sticks and switches for the check. When you go back to flying the mode doesn't change until you move the switch (like ArduPilot, which only reacts to position changes) and, if the arm switch was left up, you have to move it down and up again to arm. The preview does arm with your arm switch (if no switch is assigned, click the ARMED/DISARMED label in the corner). Armed, the lift motors idle even with zero throttle, as in ArduPilot (Q_M_SPIN_ARM, 0.10 by default; MOT_SPIN_ARM in ArduCopter and dshot_idle_value 5.5 % in Betaflight), and each motor shows its percentage. Disarmed, the motors don't spin and the tilt motors don't move in VTOL (on the real aircraft they only do with Q_OPTIONS bit 10, "Disarmed Yaw Tilt"). Surfaces do move while disarmed, as in ArduPlane. The checklist ticks when you take each axis to both ends, throttle to 0 and 100 %, the mode switch through its three positions and the arm switch up and down.

No radio: keyboard or gamepad

With the keyboard: `W`/`S` throttle, up and down arrows elevator, left and right arrows aileron, `A`/`D` rudder. A game controller is mapped the same way as a radio.

6 Flying the airplanes: VTOL, 4+1, fixed wing and flying wing

The Stallion is a tilt-rotor: the two front motors rotate from vertical to horizontal, and the rear motor only works in vertical flight. The autopilot (ArduPlane with QuadPlane) mixes motors and control surfaces according to the mode and flight phase; you just fly with the sticks and the mode switch.

First takeoff

1. Select **QHOVER**. In QHOVER, throttle at center holds altitude, above center climbs and below center descends.
2. Throttle all the way down and **arm**.
3. Raise the throttle above center until you lift off, and stabilize at about 10 m.
4. Use aileron and elevator to tilt the aircraft and move around; use rudder to rotate in place.
5. To land, lower the throttle below center and touch down at less than 1 m/s. Then disarm.

Transition to airplane flight

From a vertical flight mode, switch to an airplane mode (FBWA, for example) with some altitude, at least 25 m:

Phase	What happens	What you see
1. Start	The front motors tilt to `Q_TILT_MAX` and the aircraft accelerates. The motors still carry the weight.	`TRANSITION · waiting for AIRSPEED_MIN`
2. Minimum airspeed	Once above `AIRSPEED_MIN`, the wing is generating lift and the autopilot gradually removes vertical assistance over `Q_TRANSITION_MS`.	`TRANSITION · xx % VTOL`
3. Airplane flight	Motors fully horizontal and rear motor stopped.	“Transition complete: airplane flight”

> **Tip**
> During the transition, keep the nose straight and some throttle in. If airspeed doesn’t build, the aircraft stays waiting in phase 1 and drains a lot of battery.

Back to VTOL

In a Q mode (QHOVER, QLOITER, QLAND, QRTL), the motors return to vertical at `Q_TILT_RATE_UP` while the aircraft slows down. Once they’re vertical, “Motors vertical: VTOL flight” appears. Come in slowly and lined up with the wind.

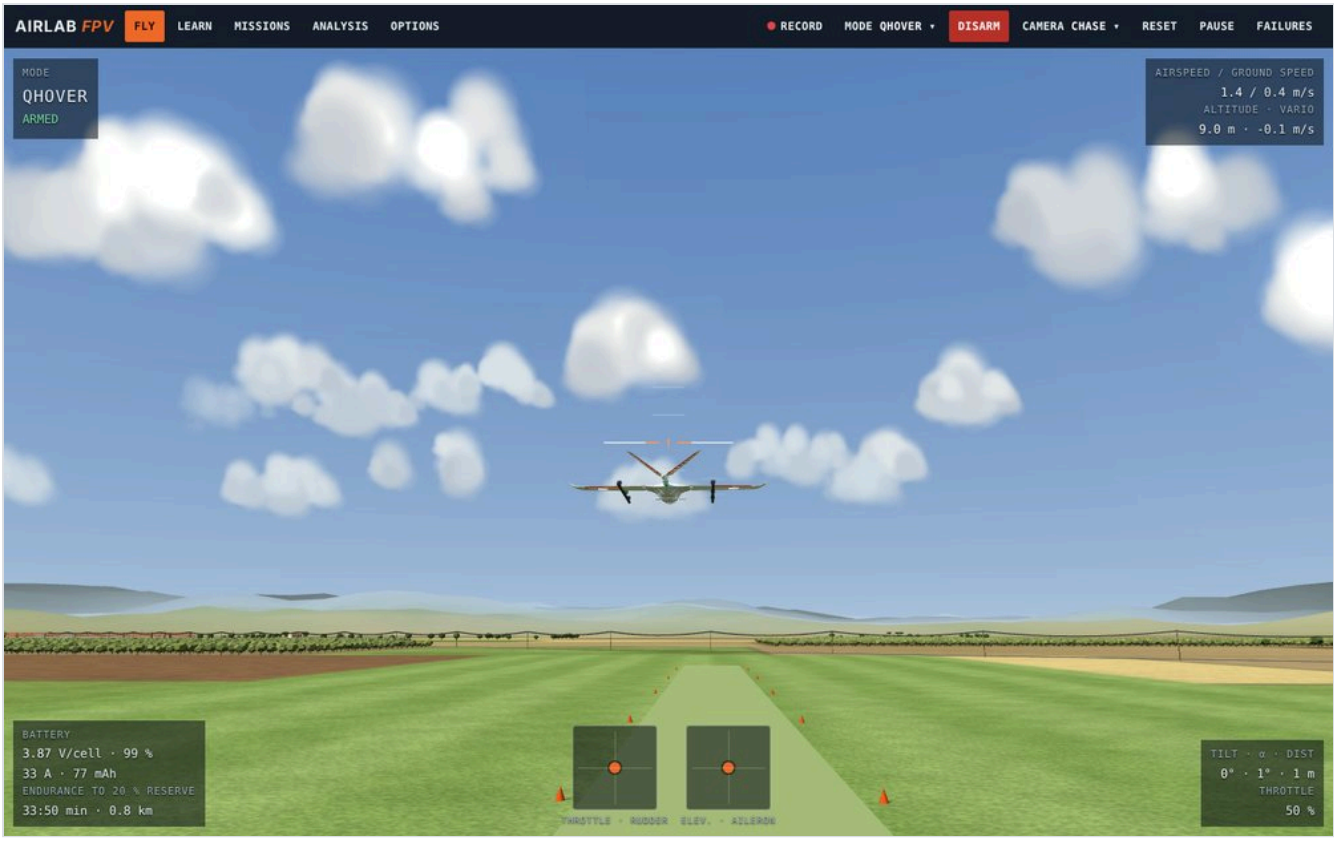
Battery and flight time

The OSD shows per-cell voltage, consumption and remaining flight time down to the 20 % reserve. If the time needed to return home runs out, “RETURN NOW” appears. Vertical flight draws several times more power than airplane flight: transition early.

Common hazards

- **Stall**: in airplane flight, below minimum airspeed or at too high an angle, the wing stops producing lift and “STALL” appears. Lower the nose and add throttle.
- **Vortex ring state**: in vertical flight, descending quickly into your own downwash makes the aircraft drop out of control. Descend slowly or while moving forward.
- **Wind**: in hover, the wind pushes you around; in airplane flight, always land into the wind.

QuadPlane 4+1



QuadPlane 4+1 in QHOVER: four lift motors on two booms and the forward-thrust motor in the nose.

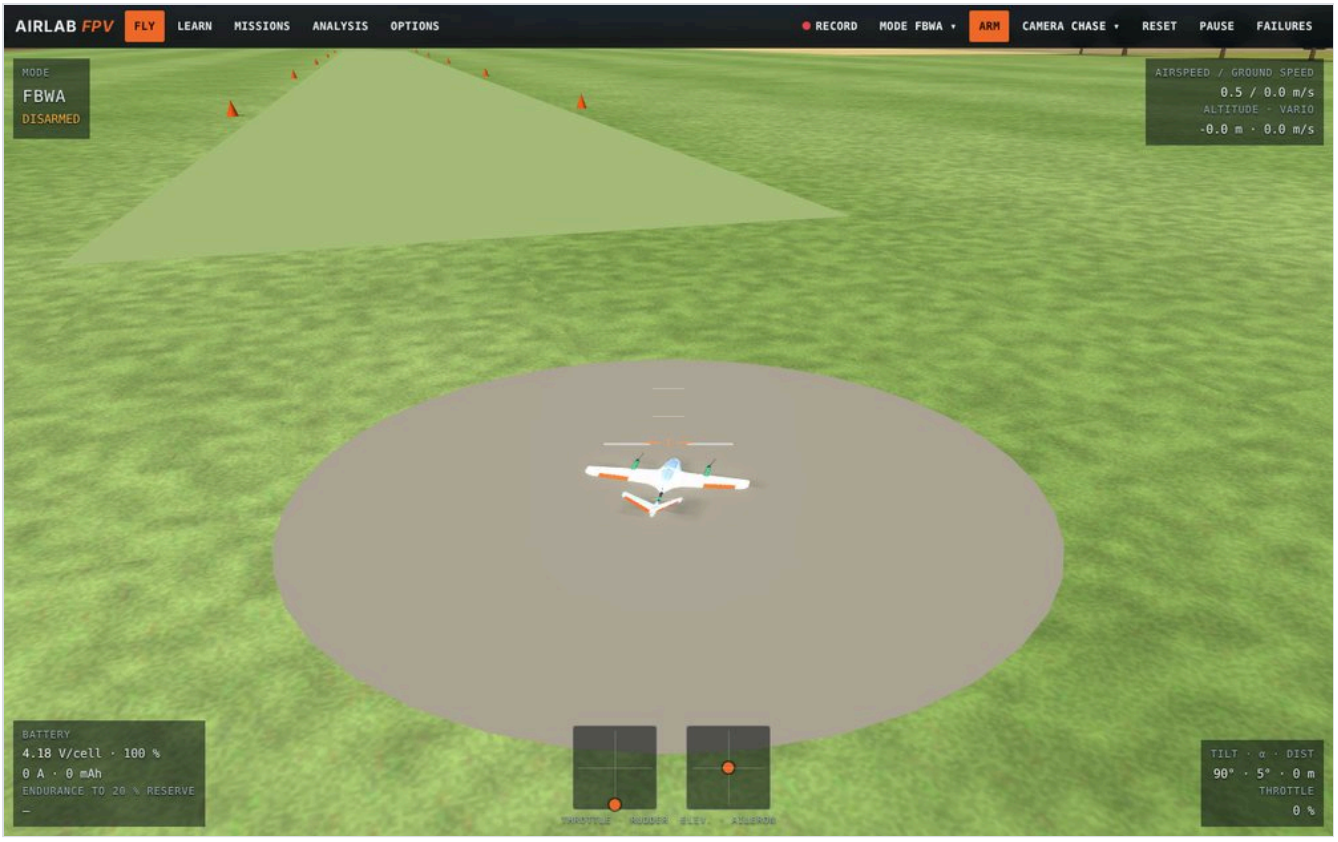
This is the most common VTOL configuration: four fixed motors that only push upward, mounted in an X on two booms under the wing, plus a separate motor for airplane flight. No motors tilt. It uses the same wing as the tilt-rotor, is heavier (an estimated 3.0 kg), and the booms add drag.

- **In vertical flight** it controls like a quadcopter: roll and pitch through differential thrust between motors, and yaw through propeller reaction torque (it yaws more slowly than the tilt-rotor, which yaws by tilting its motors).
- **Transition**: when you switch to an airplane mode, the forward motor spools up while the four lift motors hold altitude. Once above `AIRSPEED_MIN`, the lift motors wind down over `Q_TRANSITION_MS`. You’ll see the same messages as with the tilt-rotor.

- **Back to VTOL**: when you select a Q mode, the forward motor stops and the four lift motors take over immediately: there's no waiting for motors to tilt.

Lessons, challenges and missions with VTOL_TAKEOFF and VTOL_LAND work the same as with the tilt-rotor.

Fixed wing: hand launch, TAKEOFF and landing



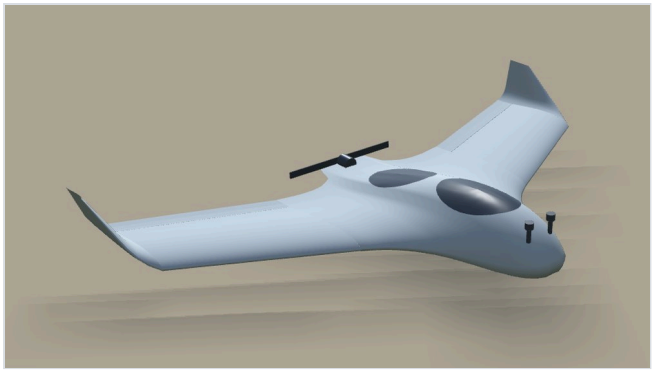
The fixed wing on the ground, ready to launch.

A twin tractor-motor airplane with a V-tail and no vertical-flight motors (2.0 kg). It only has ArduPlane's airplane modes plus **TAKEOFF**; if you select a Q mode on the switch, the simulator uses its equivalent (QSTABILIZE → MANUAL, QHOVER → FBWB, QLOITER → LOITER, QRTL → RTL) and tells you so in the log.

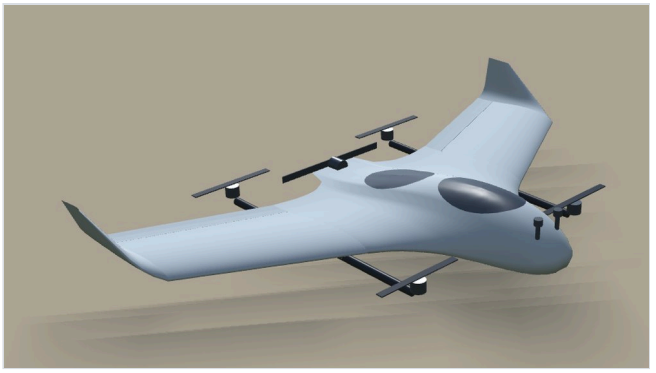
- **Hand launch**: arm in FBWA and raise the throttle above half; after 0.6 s the aircraft is thrown at about 11 m/s with the nose slightly up. Climb at a shallow angle and don't pull elevator until you have airspeed.
- **TAKEOFF mode**: arm in TAKEOFF and, after 2 s, a helper throws it (on the real aircraft, the jolt detected by `TKOFF_THR_MINACC`). It climbs at full throttle at `TKOFF_LVL_PITCH`, with roll limited to `LEVEL_ROLL_LIMIT` up to `TKOFF_LVL_ALT`, up to `TKOFF_ALT`, and then circles `TKOFF_DIST` ahead of the launch point. The default values (15°, 20 m, 50 m, 50 m) are typical for an airplane of this size.
- **Landing**: manually, approach into the wind with low throttle and flare just before touchdown; the aircraft makes a belly landing and slows down by friction. In AUTO, the **LAND** command does it for you (chapter 17).

The lessons and challenges are VTOL-based, so they aren't available with the fixed wing. RTL returns home and circles at `RTL_ALTITUDE` without landing.

Flying wing: Super Stingray and Super Stingray VTOL



Super Stingray: flying wing with a pusher.

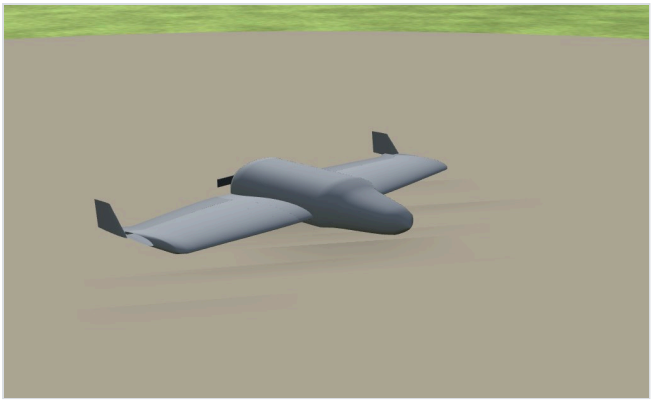


Super Stingray VTOL: 4+1 with carbon booms.

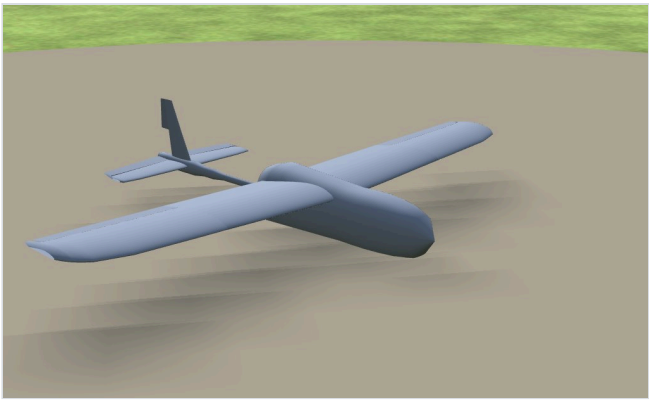
A 1,130 mm swept flying wing with no tail and no rudder: it is controlled with two **elevons**, which act as ailerons and elevator at the same time, and the winglets provide directional stability. A single motor pushes from the tail.

- **Where its data comes from**: planform geometry from the Flightory manual (span, 31 dm², 305 mm root chord, 271 mm mean chord), Selig S5020 airfoil and center of gravity 60 mm behind the leading edge at the root. The aerodynamics are computed on that geometry and matched to the CFD analysis in the manual: lift slope ≈ 3.9 per radian, static margin of 4.7 % of the mean chord and zero pitching moment at 0° angle of attack. Winglet size, elevon throws (±20°) and weights are estimated.
- **Masses**: 1.25 kg for the fixed wing and 1.65 kg for the VTOL, with a 4S2P 7 Ah Li-ion pack (4S in the simulator). Change the mass in Parameters if your build weighs something else: between 1 and 2.5 kg according to the manual.
- **How it flies**: cruise at about 22 m/s (80 km/h) at 40 % throttle and about 6 A, close to Flightory's figures (70-90 km/h, 40 %, 5 A). Stall is around 8 m/s at 1.25 kg. With a small static margin it is very lively in pitch: in MANUAL and ACRO use a gentle hand.
- **Fixed wing**: hand launch it like the fixed wing (throttle above half, TAKEOFF or AUTO) and belly land it. Flightory recommends throwing it backward, from behind your back, holding it by the leading edge with both hands.
- **VTOL**: four 2306 1700 KV motors with 7" props in an X on two 16 × 500 mm booms, as in the conversion pack. The front motors sit closer to the center of gravity than the rear ones, so they work harder in vertical flight. It comes with the parameters from the Flightory manual: `AIRSPEED_MIN` 17, `AIRSPEED_MAX` 35 and `Q_TRANSITION_MS` 3000.
- Each aircraft keeps its own parameters: what you change on one doesn't affect the others.

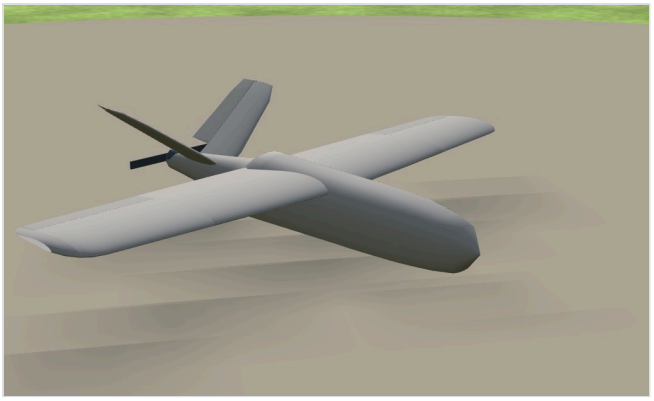
More Flightory airplanes



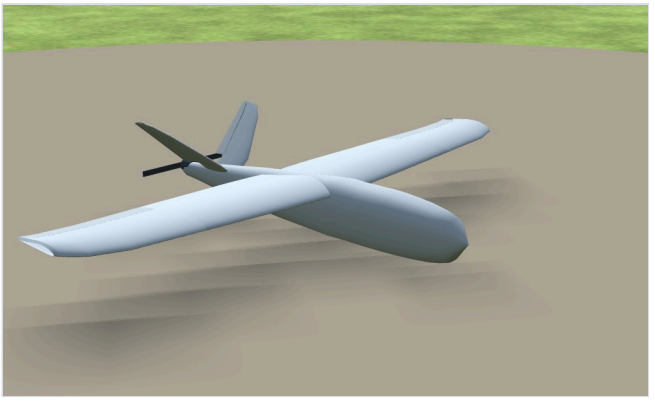
Mini Plank: straight flying wing.



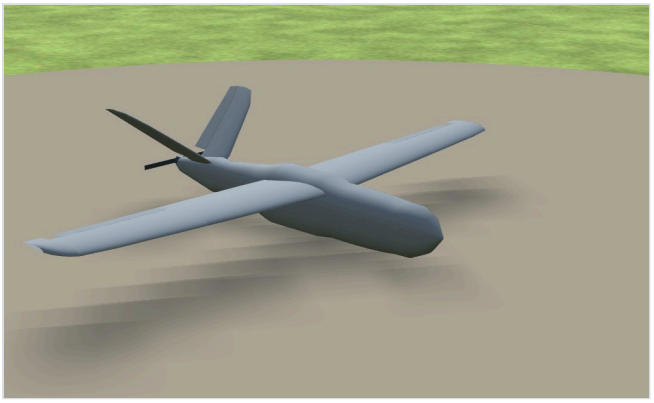
Lark: conventional tail on a boom.



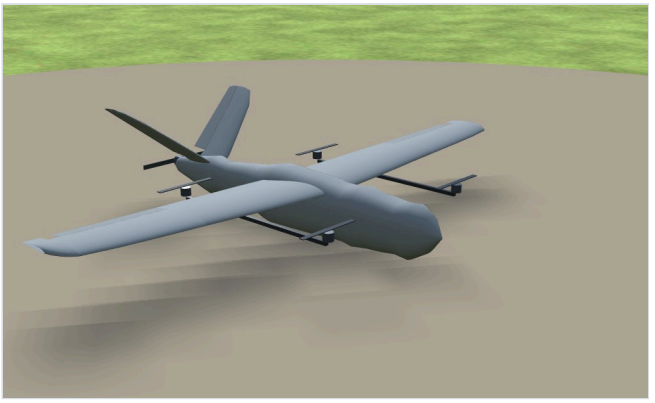
pico Talon: compact V-tail.



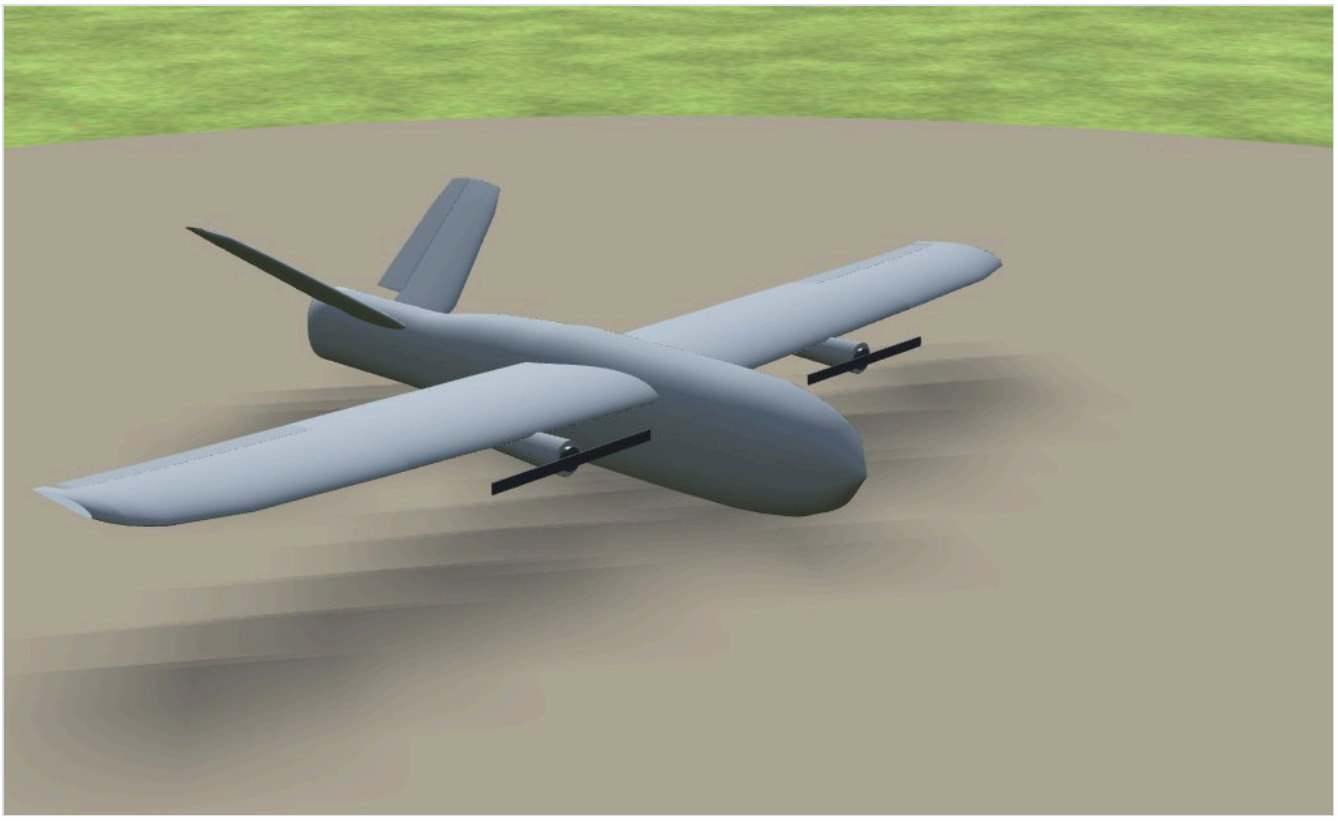
Talon 1400: 1.3 m V-tail.



Stork: aspect ratio 8.3.



Stork VTOL: 4+1 with 500 mm booms.



Moose: two tractor motors on the wing.

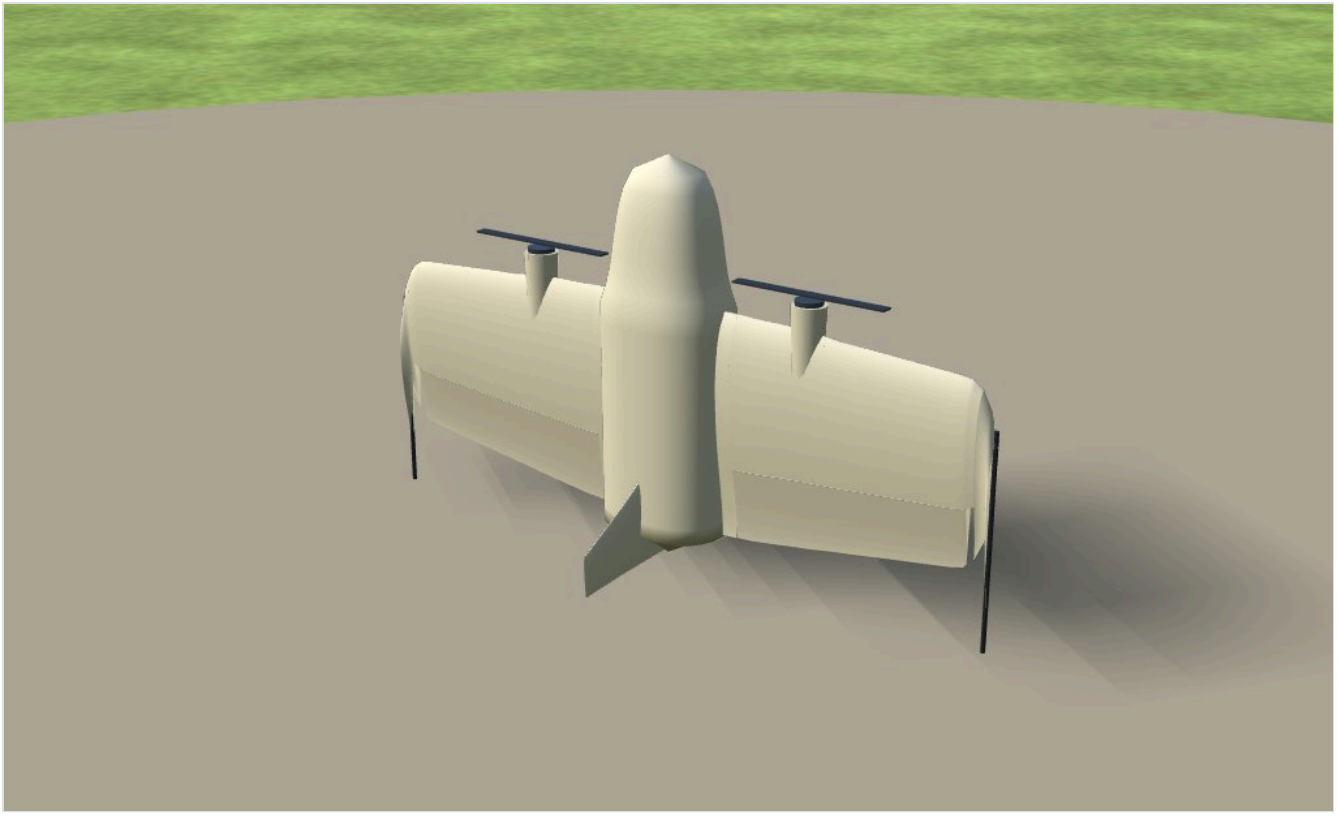
Seven more airplanes, taken from the Flightory manuals with the same method as the Super Stingray: the planform is digitized from the manual's top view (wing, tail and fuselage), the aerodynamics are computed on that geometry (wing, tails and fuselage together) and matched to the CFD analysis the designer publishes: lift slope, static margin, zero pitching moment at 0° and the CL–CD polar at the cruise point, with a little extra drag for antennas, joints and skids. Masses, batteries and V-tail angles are estimates (within each manual's ranges).

Airplane	Span · area	Airfoil · CG	Mass in the simulator	Cruise in AUTO
Mini Plank	650 mm · 12.5 dm²	PW75* · 35 mm	0.35 kg, 3S 3 Ah	14 m/s, 23 % throttle, 1.2 A
Lark	1290 mm · 24 dm²	E193 · 66 mm	2.0 kg, 4S 10.5 Ah	16 m/s, 27 %, 3.3 A (the manual gives about 30 % and 4 A)
pico Talon	905 mm · 15 dm²	S3021 · 40 mm	0.9 kg, 4S 3.5 Ah	17 m/s, 19 %, 1.7 A
Talon 1400	1305 mm · 27.5 dm²	E205 · 63 mm	2.4 kg, 4S 14 Ah	17 m/s, 28 %, 3.7 A
Stork	1620 mm · 31 dm²	S3021 · 47 mm	2.2 kg, 4S 10.5 Ah	17 m/s, 24 %, 3.5 A (the manual: under 50 %)
Stork VTOL	1620 mm · 31 dm²	S3021 · 47 mm	2.8 kg, 4S 10.5 Ah	17 m/s, 32 %, 5.8 A; about 45 A in hover
Moose	1600 mm · 36.5 dm²	S3021 · 48 mm	3.2 kg, 4S 14 Ah	18 m/s, 22 %, 5.8 A (the manual: under 50 %)

* The PW75 airfoil isn't published in the airfoil databases: the MH60, another reflex flying-wing airfoil of the same thickness, is used instead and the result is matched to the manual's CFD. CG: distance from the leading edge at the wing root.

- **They are all hand launched** like the fixed wing (throttle above half, TAKEOFF or AUTO) and belly landed; the Stork VTOL takes off and lands vertically like the 4+1 QuadPlane. The heavier ones (Lark, Talon 1400, Moose) leave the hand close to stall speed: pull gently on the elevator as soon as they're released, or use TAKEOFF.
- **V-tail** (pico Talon, Talon 1400, Stork, Moose): the two ruddervators act as elevator when they move together and as rudder when they move in opposite directions. The simulator does that mixing, as ArduPilot does with the V-tail servo functions.
- **The Stork and the Moose have a large static margin** (33 % and 25 % of the mean chord according to their CFD): they're very stable and need more elevator to flare when landing. The Talon 1400 and the Mini Plank, at 5 %, are much livelier.
- Autopilot gains are scaled to each airplane's control authority, and each one keeps its own parameters.

Tailsitter: Riflebird



Riflebird sitting on its tail, ready for takeoff.

A 777 mm, 16.5 dm² flying wing that takes off and lands vertically **sitting on its tail**, resting on the wingtip legs and the dorsal and ventral fins. Its two motors (T-Motor F60 1750 KV with 7×5 props) are fixed: to fly as an airplane, the whole aircraft tilts 90°. Mass 1.0 kg with a 4S 4 Ah Li-ion pack; aerodynamics matched to the manual's CFD (7.6 % static margin).

- **In VTOL** (QSTABILIZE, QHOVER, QLOITER, QLAND, QRTL) it flies nose up with the belly facing forward, like ArduPilot's "VTOL view": roll comes from **differential thrust** between the two motors, and

pitch and yaw from the **elevons**, which work in the propwash even when the aircraft is standing still. This is the `Q_TAILSIT_ENABLE` = 1 setup with `Q_FRAME_CLASS` = 10 (twin motor).

- **Transition to airplane**: when you switch to an airplane mode, the nose comes down at `Q_TAILSIT_RAT_FW` (50 °/s) while holding altitude, and the transition ends when the pitch passes `Q_TAILSIT_ANGLE` (45° from vertical). It loses a few meters while it picks up speed: do it at least 30 m up.
- **Return to VTOL**: the nose comes up at `Q_TAILSIT_RAT_VT` (50 °/s) at hover throttle until the airplane pitch passes `Q_TAILSIT_ANG_VT` (0 means the same `Q_TAILSIT_ANGLE`); then VTOL control brings it to vertical at the same rate. If you're flying fast, the aircraft climbs a lot as the nose comes up: slow down first.
- The four parameters are under Parameters → Tailsitter. The sticks don't command anything during transitions, as in ArduPilot.

7 ArduPlane flight modes

Keys `1` through `0` select the first ten modes in the list.

Mode	Type	How it responds
QSTABILIZE	Vertical	Stabilized: the sticks command the tilt angle (up to `Q_ANGLE_MAX`). Throttle is direct; you control altitude. The throttle curve depends on `Q_THROTTLE_EXPO`.
QHOVER	Vertical	Like QSTABILIZE, but with altitude hold: throttle at center holds, above climbs and below descends.
QLOITER	Vertical	GPS position and altitude hold. When you release the sticks, it stays put.
QLAND	Vertical	Vertical landing: descends at `Q_WP_SPEED_DN`, and at `Q_LAND_FINAL_SPD` below `Q_LAND_FINAL_ALT`. Disarms on touchdown. You can reposition it with the sticks.
QRTL	Automatic	Returns home (in airplane flight if far away), switches to VTOL on approach and lands vertically.
FBWA	Airplane	Stabilized: the stick commands roll angle (up to `ROLL_LIMIT_DEG`) and pitch angle; release it and the aircraft levels out. Throttle is direct.
FBWB	Airplane	Altitude hold: elevator commands climb or descent at `FBWB_CLIMB_RATE`, and throttle commands airspeed between `AIRSPEED_MIN` and `AIRSPEED_MAX`.
CRUISE	Airplane	Like FBWB, but it also holds heading when you release the aileron stick.
ACRO	Airplane	The stick commands rotation rate (`ACRO_ROLL_RATE`, `ACRO_PITCH_RATE`) and the aircraft does not self-level. With `ACRO_LOCKING` it holds attitude when you release the sticks.
MANUAL	Airplane	No assistance: the sticks move the control surfaces directly (with the expo from `MAN_EXPO_*`).
LOITER	Automatic	Circles of radius `WP_LOITER_RAD` over the current point at a fixed altitude.
CIRCLE	Automatic	Gentle circles. It's the first thing the short failsafe does in airplane modes.
RTL	Automatic	Returns home in airplane flight at `RTL_ALTITUDE` and circles overhead. With `Q_RTL_MODE` it can land vertically.
AUTO	Automatic	Follows the mission from the planner (chapter 17).
TAKEOFF	Automatic	Fixed wing only: automatic takeoff after launch, at `TKOFF_LVL_PITCH` up to `TKOFF_ALT`, then circles and waits (chapter 6).

> **Mode switch**
>
> Moving the switch activates the mode assigned to that position. After a failsafe, ArduPlane does not return to the previous mode on its own: you have to move the switch to regain control, just like on the real aircraft.

8 Flying the FPV drones and the multirotor



Mario 5 in FPV camera with the “Full freestyle” OSD.

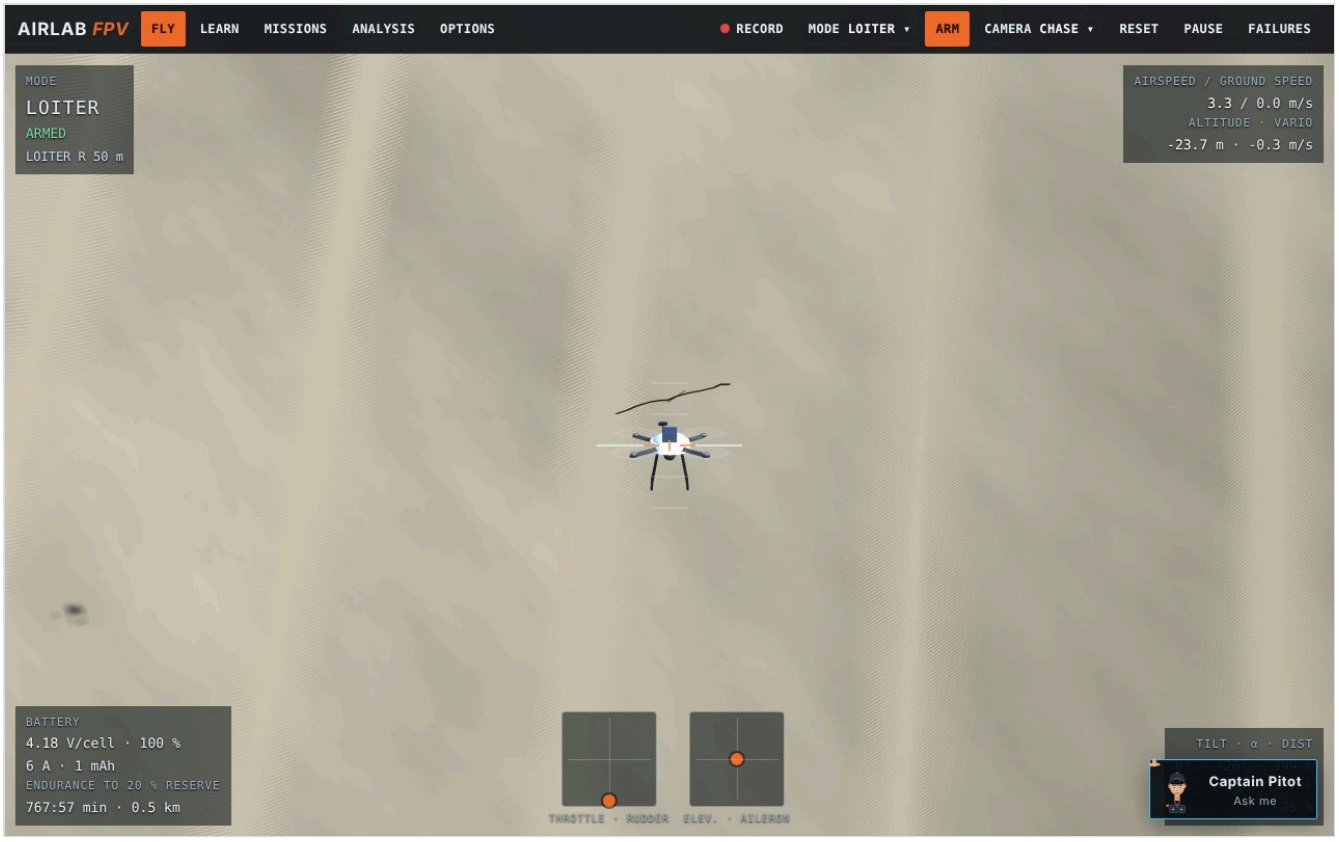
When you choose a drone, the camera automatically switches to FPV and the modes change to Betaflight’s:

Mode	How it responds
ACRO	The sticks command rotation rate according to your rates; when released, it holds attitude. This is the freestyle and racing mode.
ANGLE	Self-leveling: the stick commands the angle (up to the maximum angle set in rates), and when released the drone levels out.
HORIZON	Self-leveling with the stick centered, and acrobatics with the stick at full deflection.

- **Arm**: with the throttle at zero. The drone won’t arm if the throttle is up.
- **Throttle**: there’s no altitude hold. The more you tilt, the more throttle you need to keep from losing altitude.
- **GPS Rescue** (Mario Fold 8): if the radio link is lost far from home, it climbs, returns and lands on its own. Close to home, the failsafe stops the motors.
- **Battery**: the Mario 5 uses a 6S 1300 mAh LiPo; the Mario Fold 8, a 6S2P 8000 mAh Li-ion. The OSD and voice warn you of low battery.

> **Start in ANGLE**
>
> If you've never flown an FPV drone, start in ANGLE with gentle rates and move to ACRO once you have the throttle under control.

8.1 The inspection multirotor (ArduCopter)



The inspection multirotor in LOITER in front of the dam face: body with GPS, gimbal under the nose and landing gear.

A generic work quadcopter of about 4 kg: 15" propellers, 6S 16,000 mAh battery, GPS, a gimbal with RGB and thermal camera, and landing gear. It uses the same quadcopter physics as the FPV drones; what changes is the autopilot, with ArduCopter's modes. When you choose it, the camera switches to chase. The data are estimated: it is not a specific commercial model.

Mode	How it responds
STABILIZE	Self-levelling, up to 30° (ANGLE_MAX). Throttle is direct: you hold the altitude.
ALT_HOLD	Self-levelling with altitude hold. Throttle centred holds altitude; above or below, it climbs or descends at up to 2.5 m/s (PILOT_SPEED_UP). There is a deadzone in the centre (THR_DZ).
LOITER	Position and altitude hold with GPS, also in wind. The sticks command speed (up to 6 m/s) and on release it brakes and stays put. It is the inspection mode and the one set on reset.
FLOWHOLD	Position hold **without GPS**, using the downward optical-flow sensor. It drifts a little with little light, little texture (water) or a lot of height. For interiors and under bridges (8.2).
RTL	Climbs to 15 m if lower (RTL_ALT), flies home at 10 m/s (WPNAV_SPEED), waits 5 s (RTL_LOIT_TIME) and lands.

LAND	Lands where it is: descends at 1.5 m/s (WPNAV_SPEED_DN) and at 0.5 m/s below 10 m (LAND_SPEED, LAND_ALT_LOW). Disarms on touchdown.
AUTO	The planner mission: takeoff, waypoints at 10 m/s with the nose towards the next waypoint (WP_YAW_BEHAVIOR 2), camera triggering, RTL or landing. The planner's **Fly mission** also works for the multirotor.

- **Arming**: with throttle low. In LOITER and ALT_HOLD, raise the throttle above centre to take off and leave it centred to stay put.
- **Yaw**: the rudder turns at up to 202.5°/s (PILOT_Y_RATE). On release it holds heading.
- **Auto disarm**: on the ground with throttle low, after 10 s (DISARM_DELAY).
- **Radio failsafe** (FS_THR_ENABLE 1): on the ground it disarms; in flight, RTL. It stays in RTL even if the radio comes back, until you change mode.
- **Photogrammetry survey**: **Add survey with triggering** inserts a DO_CHANGE_SPEED if photos would blur at 10 m/s.
- **Endurance**: with the example payload, about 40 min of flight to the 20 % reserve.

Defaults from the ArduCopter code: `ArduCopter/config.h`, `libraries/AC_WPNav/AC_WPNav.cpp` and `AC_Loiter.cpp`, `libraries/AC_AttitudeControl/AC_AttitudeControl.cpp`. LOIT_SPEED is set to 6 m/s (the default is 12.5): an inspection setup. ArduPilot · Copter flight modes.

8.2 Flying without GPS: ALT_HOLD, FLOWHOLD and the EKF failsafe

Inside a hangar or a tank, or under a bridge deck, there is no GPS: the multirotor does not know where it is.

- **FLOWHOLD** holds position with the downward optical-flow sensor. It drifts a little more with little light, little texture (water) or a lot of height.
- LOITER, RTL and AUTO need position: without GPS they cannot be selected, and it will not arm in them ("PreArm: needs position").
- If you lose GPS in flight in one of those modes, the **EKF failsafe** triggers and it drops to ALT_HOLD. It is set to `FS_EKF_ACTION 2`, common for work under structures; ArduCopter's default is 1, land.

8.3 Tethered cleaning drone and caged drone

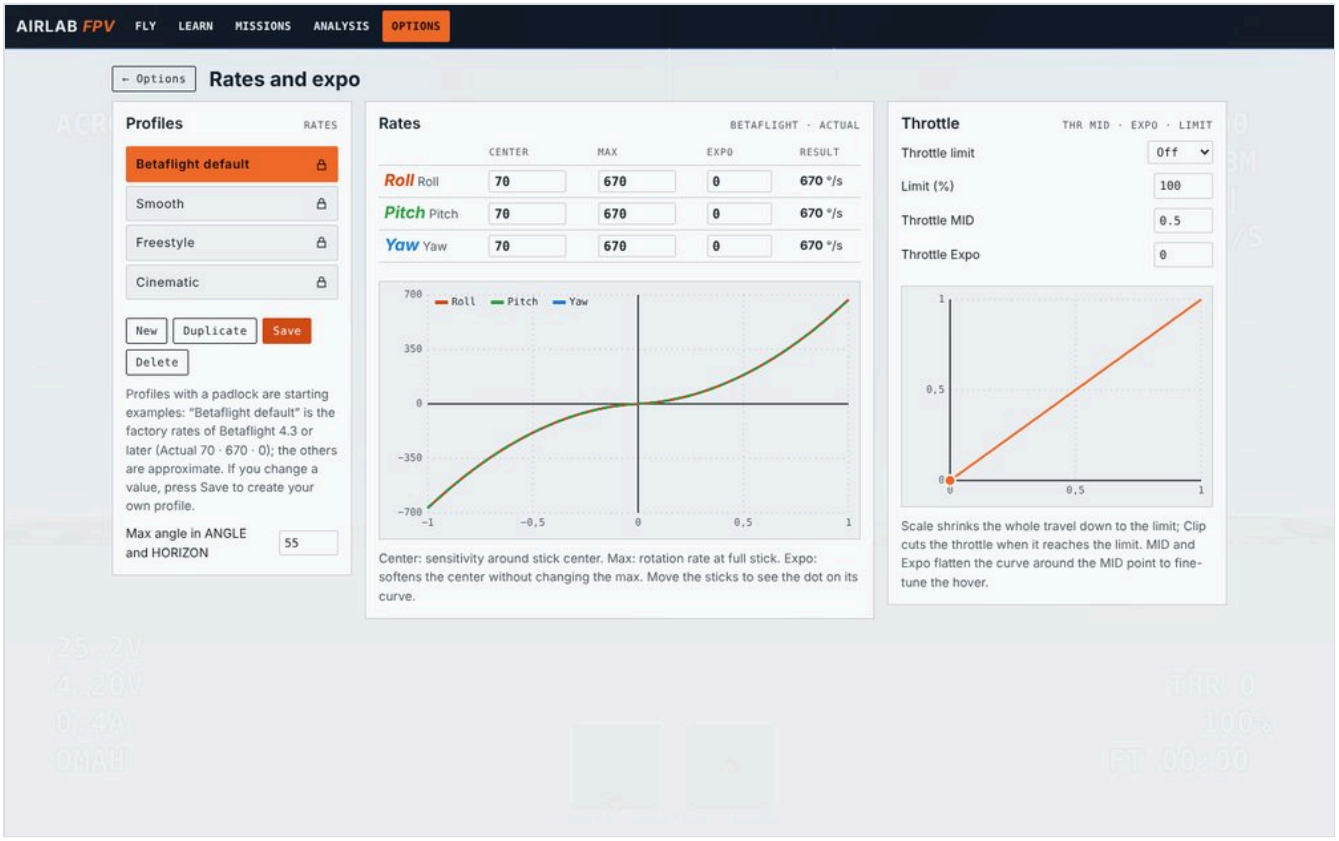
Aircraft	What it is	What is special
Tethered cleaning drone	A generic heavy multirotor of about 60 kg with 40" propellers and a water lance on the gimbal.	It gets power and water through a hose from a truck with generator, pump and tank, so it uses no battery. The full hose weighs 0.35 kg/m, the wind drags it and it cannot go further than 150 m from the truck. Outside the missions, the truck is next to the pad.
Caged drone	A generic drone of about 2.4 kg inside a 52 cm cage, with lights.	For confined spaces: it can touch the walls without falling. It flies without GPS (ALT_HOLD or FLOWHOLD) and lasts about 11 minutes.

Both use the ArduCopter modes in 8.1 and their data are estimated.

9 Rates and expo

In Options › Rates and expo you adjust how the aircraft responds to your sticks. The screen changes depending on the selected aircraft.

Drones: Betaflight “Actual” rates

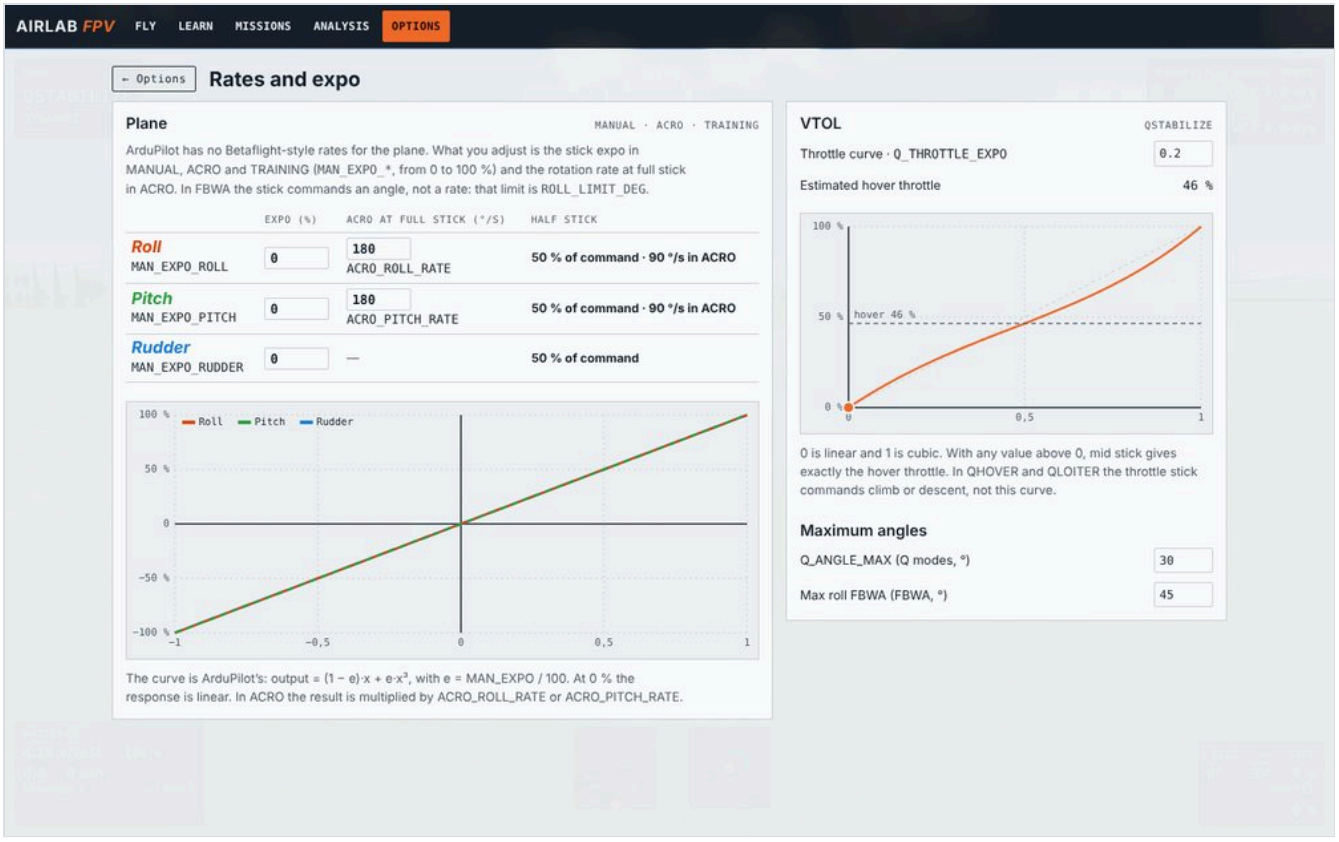


Drone rates: profiles, per-axis table, curves and throttle curve.

Setting	Meaning
Center	Sensitivity around stick center, in °/s.
Max	Rotation rate at full stick, in °/s. The table’s result is this value.
Expo	Softens the center without changing the maximum (0 to 1).
Throttle MID / Expo	Flatten the throttle curve around the MID point to fine-tune hovering.
Throttle limit	Scale reduces the full travel; Clip cuts off when the limit is reached.
Max angle	Maximum tilt in ANGLE and HORIZON.

Profiles with a padlock are starting points: “Betaflight default” (70 · 670 · 0) and three examples (Smooth, Freestyle and Cinematic). When you change a value, **Save** creates your own profile. Move the sticks and you’ll see a dot travel along each curve.

Airplane: what ArduPilot lets you adjust

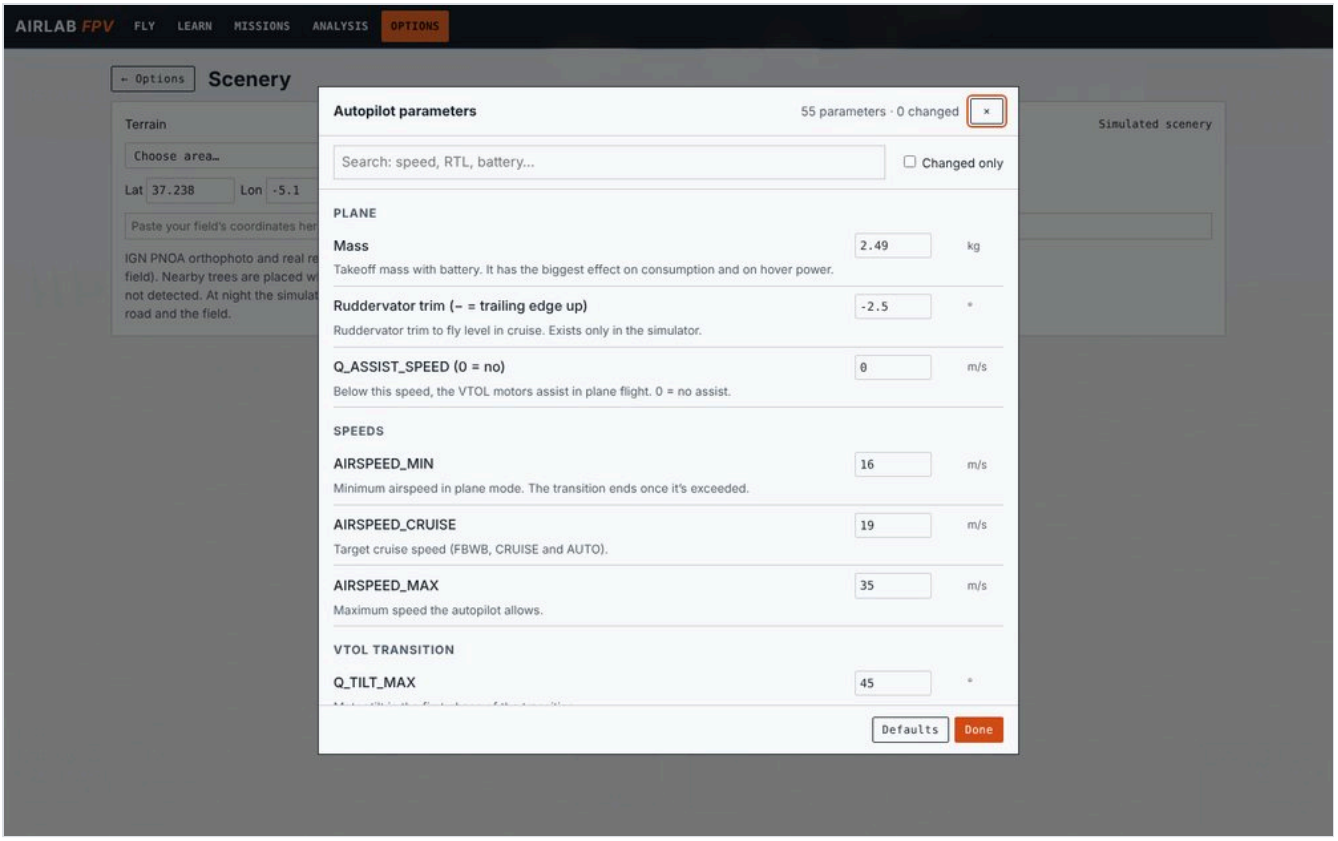


Stallion rates: MANUAL and ACRO expo, ACRO rates and QSTABILIZE throttle curve.

- **MAN_EXPO_ROLL, _PITCH and _RUDDER** (0 to 100 %): stick expo in MANUAL, ACRO and TRAINING. The curve is ArduPilot's: output = (1 − e)·x + e·x³, with e = MAN_EXPO / 100.
- **ACRO_ROLL_RATE and ACRO_PITCH_RATE** (°/s): rotation rate at full stick in ACRO.
- **Q_THROTTLE_EXPO** (0 to 1, 0.2 by default): throttle curve in QSTABILIZE. Above 0, mid-stick gives exactly hover throttle.
- **Q_ANGLE_MAX** and **ROLL_LIMIT_DEG**: maximum tilt in Q modes and in FBWA.

10 ArduPilot parameters

Options › Aircraft › **Configure parameters** opens the list of parameters the simulator uses, grouped (aircraft, airspeeds, transition, hover, airplane, MANUAL and ACRO, navigation, failsafe and battery), with a description of each one, a search box and the option to show only the ones you've changed.



The parameters dialog.

Importing your .param

With **Import Mission Planner .param** you load your flight controller's parameter file (Mission Planner or QGroundControl format). The simulator picks up the ones it understands, converts legacy units (for example, centidegrees in `LIM_ROLL_CD` or cm/s in `Q_WP_SPEED`) and shows you what changed. Everything else (PIDs, sensors, outputs) has no effect on it.

Navigation, TECS and VTOL assistance parameters

On top of the basic ones, the simulator uses these ArduPlane 4.7 parameters with the behaviour described in its documentation. When importing a .param it also recognises the old names in brackets.

Parameter	Default	What it does in the simulator
`NAVL1_PERIOD`, `NAVL1_DAMPING`, `NAVL1_XTRACK_I`	17 s · 0.75 · 0.02	L1 path controller (chapter 17).
`TECS_TIME_CONST`	5 s	How quickly TECS corrects height in AUTO, CRUISE, FBWB, LOITER and RTL. Lower is faster but more aggressive.

`TECS_SPDWEIGHT`	1	0: pitch controls height and throttle controls speed. 2: pitch controls speed (gliders). Without an airspeed sensor, 0 is used.
`STALL_PREVENTION`	1	Limits bank by load factor: below `AIRSPEED_MIN`, to 25°; above it, to the angle the wing can sustain at that speed (never less than 25° or `LEVEL_ROLL_LIMIT`). In the automatic modes it also raises the minimum airspeed in turns.
`MIN_GROUNDSPEED` (`MIN_GNDSPD_CM`)	0 (off)	Minimum ground speed in auto-throttle modes (except CIRCLE): into a headwind it raises the target airspeed, up to `AIRSPEED_MAX`.
`THR_MIN`, `THR_MAX`	0 % · 100 %	Forward-motor throttle limits in every mode except MANUAL. In FBWA the stick is spread between the two.
`TRIM_THROTTLE`, `ARSPD_USE`	45 % · 1	With `ARSPD_USE` = 0 the autopilot ignores the airspeed sensor and TECS sets throttle from `TRIM_THROTTLE` and pitch, as ArduPlane does without a pitot. If `TRIM_THROTTLE` is not the real cruise throttle, the aircraft flies faster or slower than planned. With the sensor, throttle comes from the simulator's thrust model.
`Q_ASSIST_ANGLE`, `Q_ASSIST_ALT`, `Q_ASSIST_DELAY`	30° · 0 (off) · 0.5 s	VTOL motor assistance in aeroplane modes (except MANUAL and ACRO): by airspeed (`Q_ASSIST_SPEED`, immediate), by angle (attitude more than 5° outside the limits and the error above the angle) or by height above home. Angle and altitude wait `Q_ASSIST_DELAY`, and assistance is removed after twice that time. They only work if `Q_ASSIST_SPEED` is positive.
`Q_TRANS_FAIL`, `Q_TRANS_FAIL_ACT`	0 (no limit) · 0	If the forward transition spends more than `Q_TRANS_FAIL` seconds waiting for `AIRSPEED_MIN`: −1 warns only, 0 switches to QLAND, 1 to QRTL. To practise it, set 5 s and transition with low throttle.
`Q_PILOT_SPD_UP`, `Q_PILOT_SPD_DN` (`Q_VELZ_MAX`, `Q_VELZ_MAX_DN`)	2.5 m/s · 0	Climb and descent on the throttle in QHOVER and QLOITER. With `Q_PILOT_SPD_DN` = 0 it descends as fast as it climbs.
`Q_LOIT_SPEED_MS` (`Q_LOIT_SPEED`)	5 m/s	Horizontal speed at full stick in QLOITER.
`RTL_AUTOLAND`	0	1: on reaching home in RTL (within 10 m of the RTL altitude) it continues the mission from the nearest `DO_LAND_START`. 2: it jumps straight there. Without a `DO_LAND_START` it stays in RTL. With a `DO_LAND_START` in the mission and `RTL_AUTOLAND` = 0, ArduPlane won't arm, and neither will the simulator.

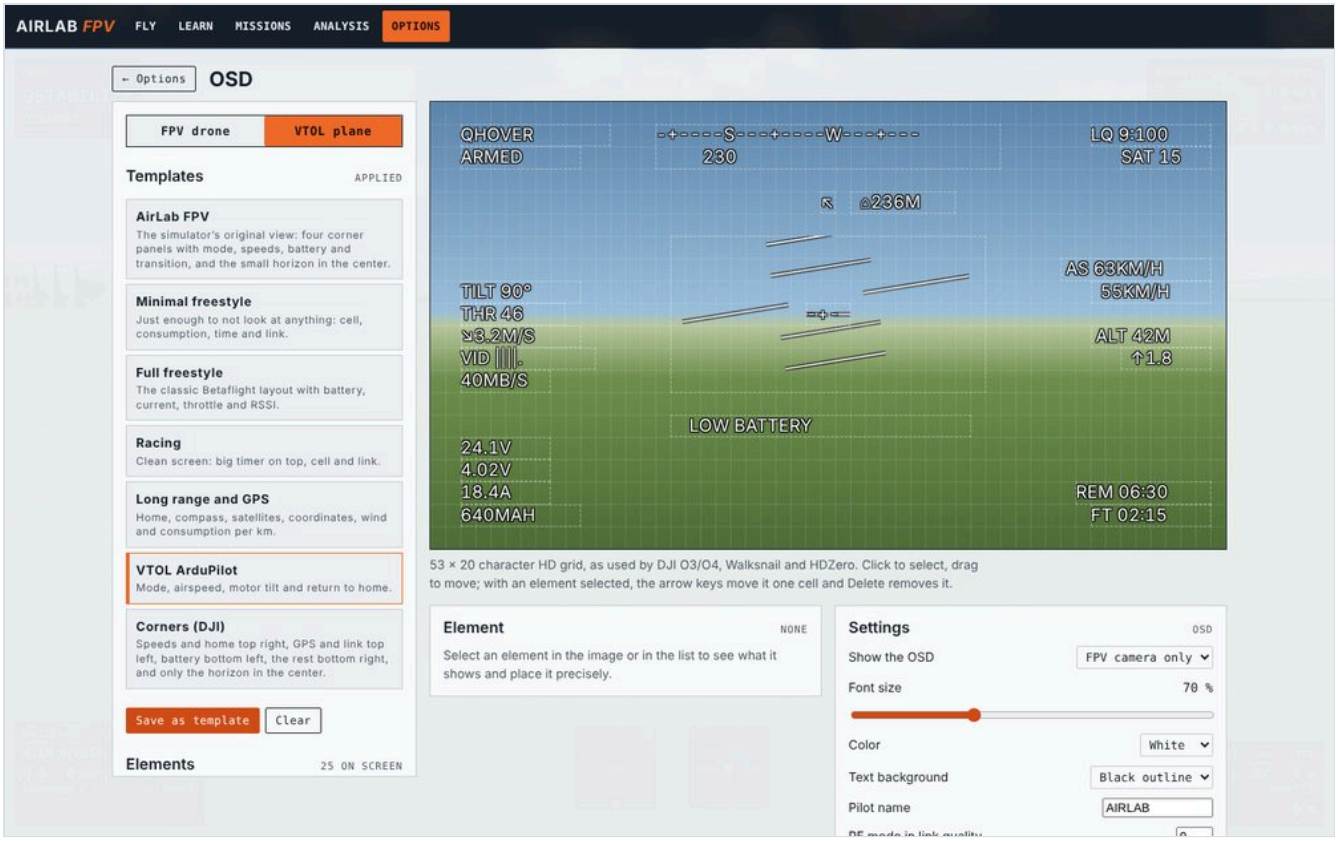
Exporting

Export .param downloads the simulator's parameters in ArduPilot units so you can compare them with your aircraft's.

> **Heads up**
>
> Testing parameters here helps you understand their effect, but don't copy them to your real aircraft without reviewing them: the model is approximate.

11 The OSD

The OSD is the data overlaid on the camera image. In Options › OSD you design it just like in the Betaflight Configurator, on the 53 × 20 character HD grid used by DJI O3/O4, Walksnail and HDZero. The drone and the airplane each have their own OSD.



OSD editor: templates and elements on the left, preview in the center.

How to edit

- **Add**: drag an element from the list onto the image, or turn it on with its switch.
- **Move**: drag it, or select it and use the arrow keys (one cell per press).
- **Remove**: select it and press `Delete`, or click “Remove from OSD”.
- **Exact position**: when it’s selected, type in the column and row.

Templates

Template	Contents
AirLab FPV	The simulator’s original view: four panels in the corners and a small horizon.
Minimal freestyle	Cell voltage, consumption, time, link and mode.
Full freestyle	The classic Betaflight layout, with battery, current, throttle, RSSI and video signal. Default on the drones.
Racing	Large timer at the top, cell voltage and link.

Long range and GPS	Home, compass, satellites, coordinates, wind, consumption per km and horizon.
ArduPilot VTOL	Mode, armed status, airspeed, motor tilt and return to home. Default on the airplane.
Corners (DJI)	Speeds and home at top right, GPS and link at top left, battery at bottom left, everything else at bottom right, and only the horizon in the center.

You can start from any of them, modify it and save it under your own name.

Available elements

Group	Elements
Battery	Cell voltage, total voltage, battery remaining, current, consumption (mAh), power, consumption per km, flight time remaining
Flight	Mode, armed status, flight time, altitude, variometer, ground speed, airspeed, throttle, angles, motor tilt
Navigation	Distance to home, direction to home, compass, heading, satellites, GPS coordinates, distance traveled, wind
Radio and video	Link quality (LQ), RSSI, video signal, bitrate, video latency
Center	Artificial horizon, crosshair, warnings, pilot name

Settings

When it’s shown (FPV camera only, all cameras or never), font size, color (white, green, yellow or cyan), text background (outline, box or none), pilot name and the RF mode digit for link quality.

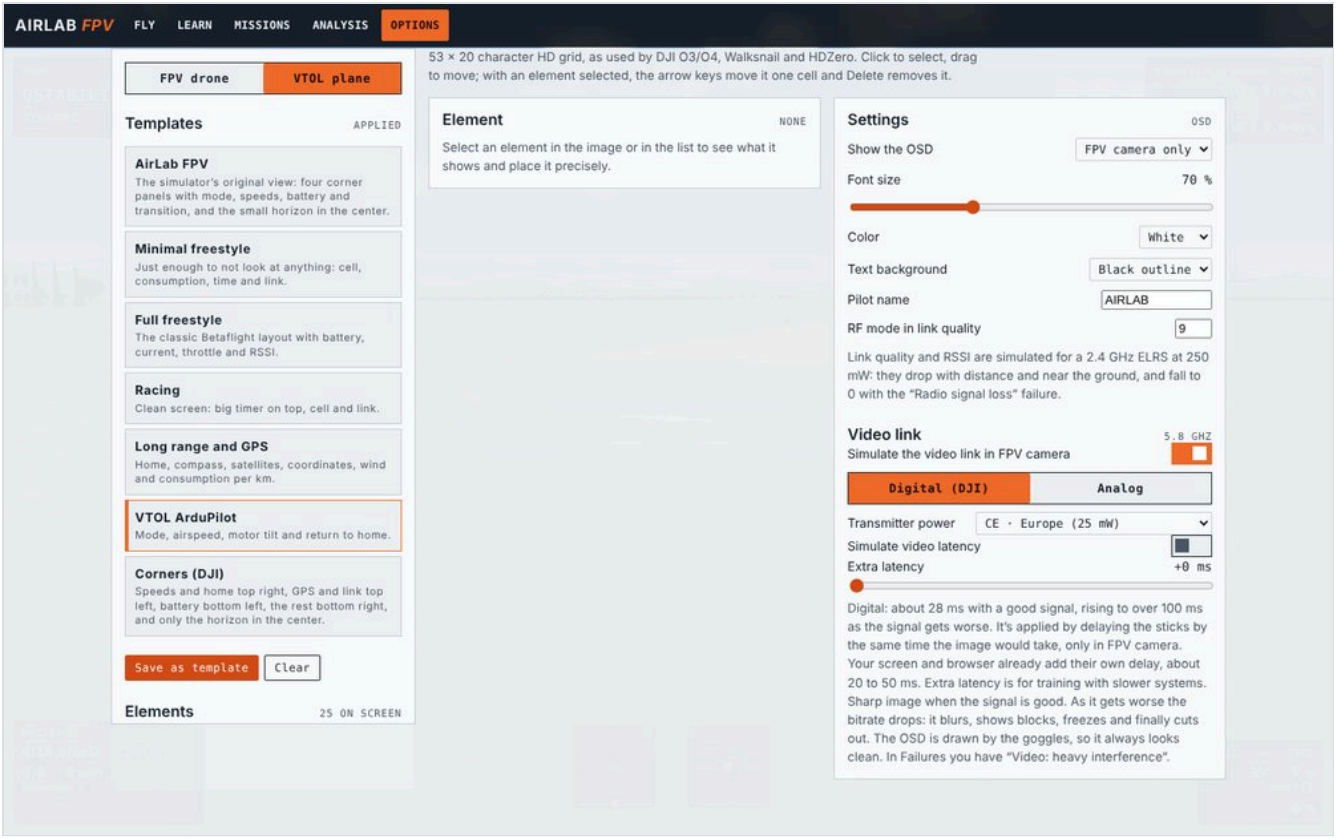
12 Radio and video link

Radio link: LQ and RSSI

The radio sends hundreds of packets per second. **Link quality (LQ)** is the percentage that arrive intact; in `LQ 9:100`, the 9 is the RF mode and the 100 is the quality. **RSSI** is the received signal strength in dBm (−50 up close, −90 or −100 far away). The simulator calculates both for a 2.4 GHz ELRS link at 250 mW: they drop with distance and when flying low and far away, and they fall to 0 with the “Radio signal loss” failure, which triggers the failsafe.

Video link

In Options › OSD › Video link you choose between **Digital (DJI)** and **Analog**. It only affects the FPV camera.



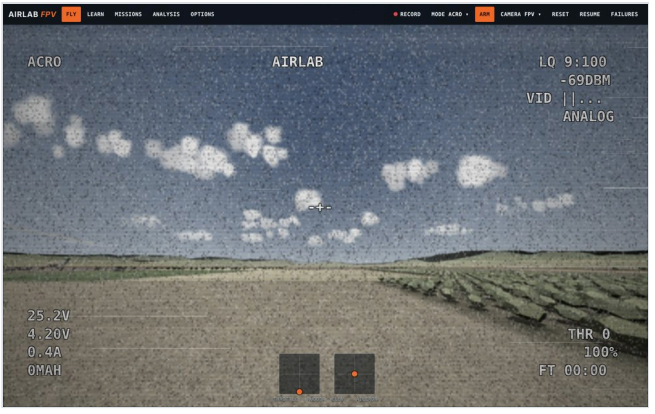
Video link and latency settings.

	Digital (DJI)	Analog
With good signal	Sharp HD image	PAL resolution, somewhat soft, with lines and grain
As the signal drops	The bitrate falls (50 → 4 Mb/s): the image blurs, blocks appear and it freezes now and then	More snow, flashes and a rolling band; it loses color and sync
No signal	Freezes and goes black with “VIDEO SIGNAL LOST”	Never freezes: you are left with snow, and you can still fly

OSD	Drawn by the goggles: always clean	Embedded in the signal: it picks up the noise
Power levels	CE 25 mW · FCC 500 mW	25 mW (legal limit in Europe) · 200 · 400 · 800 mW



Digital with weak signal: blocking and low bitrate.



Analog with medium signal: snow and washed-out colors.

Latency

With “Simulate video latency” enabled, your stick inputs are delayed by the same amount the image would take to arrive: about 5 ms on analog and about 28 ms on digital, rising to over 100 ms as the signal gets worse. “Extra latency” (0 to 150 ms) lets you train for slower systems. Your display and browser already add 20 to 50 ms on their own.

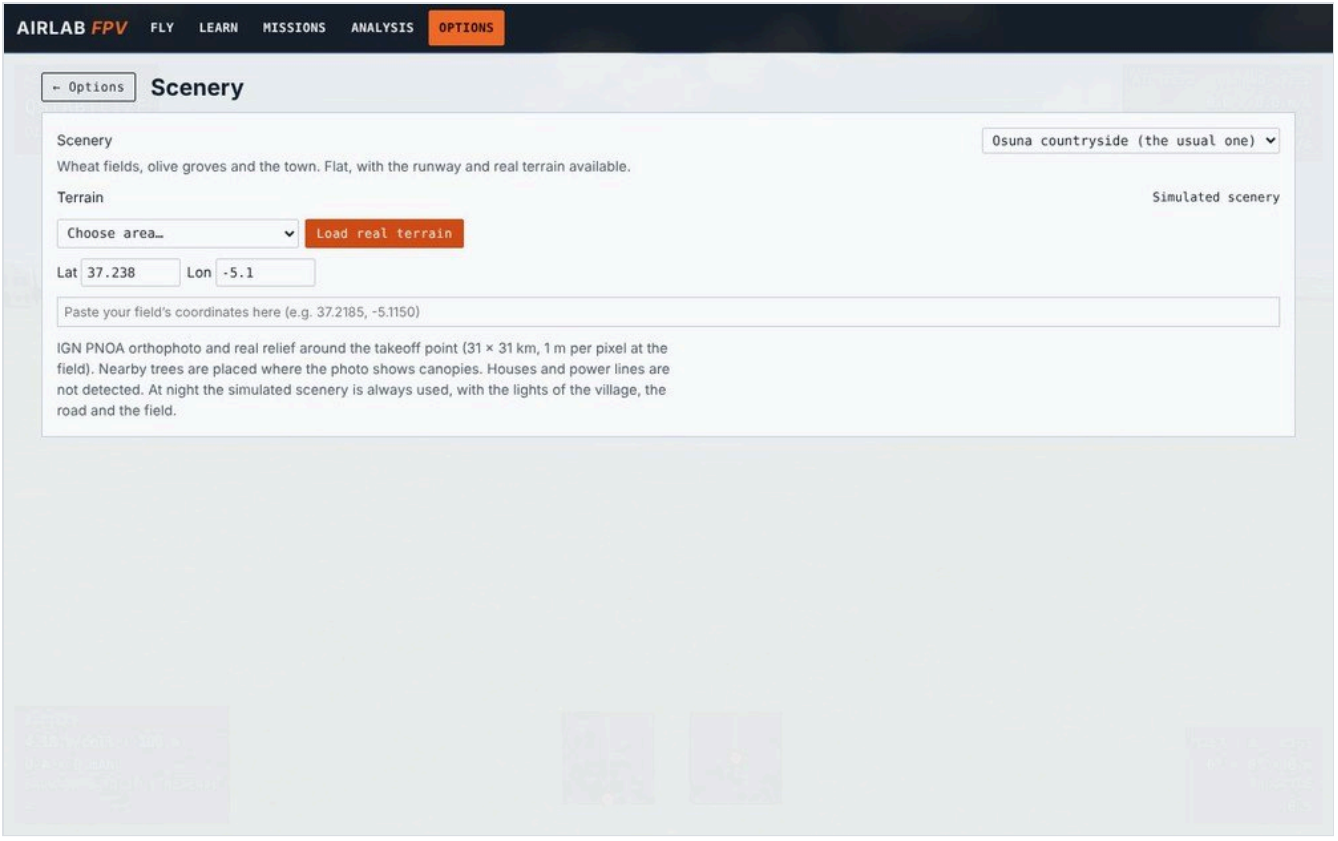
13 Cameras

Camera	Description
Pilot	From the pilot's position next to the landing pad, following the aircraft with your eyes. It's like flying line of sight.
Chase	Behind the aircraft, a few meters back. Useful for learning the transition.
FPV	From the onboard camera, as if wearing goggles. It's the only view where the grid OSD and the video link apply.
Free	Drag with the mouse to orbit around the aircraft.

14 Scenery, time and weather

Scenery

There are six sceneries. Pick one from the drop-down under **Free flight** on the home menu or in **Options › Scenery**; the simulator remembers the last one. When you change scenery the aircraft goes back to the takeoff pad, which is always in the same place, at elevation 0 and on flat ground.



Options › Scenery: the scenery selector with its description and, below, the real terrain.

Scenery	What it is	What for	Wind it sets · runway
Osuna countryside	The usual one: wheat, olive groves, the town, power lines. Flat.	Learning, lessons, challenges and missions. The only one with real terrain and the remote sensing plot.	Yours · yes
Coastal cliff	A pine plateau ending in a sandstone cliff about 80 m above the sea, to the south. Coves with beaches and sea stacks offshore.	Slope soaring with the plane (onshore wind) and dives towards the water with the drone.	South, 7 m/s · yes
Mountain and valley	Takeoff from a saddle. To the north, the summit about 380 m higher; to the south, a slope of about 500 m down to the valley and its river.	Climb and dive hugging the slope (*mountain surfing*). With a south wind, the takeoff slope gives lift.	South, 6 m/s · no (vertical or hand launch)

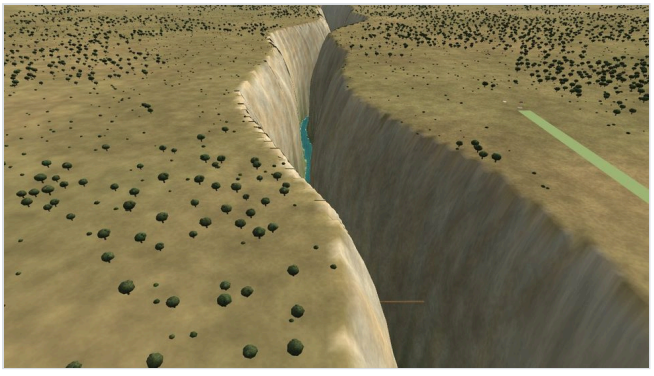
River canyon	A holm-oak plateau cut by a limestone gorge 250 m deep and 30 to 150 m wide, with the river at the bottom and a footbridge hanging halfway down.	Precision flying between walls and under the footbridge.	West, 3 m/s · yes
Quarry	A limestone pit of about 380 × 260 m and 84 m deep, in seven stepped benches, with a pond at the bottom, spoil heaps and trucks.	Dives into the pit and flying between benches.	North-west, 4 m/s · yes
Dam and reservoir	A valley closed by a 95 m high concrete arch dam: upstream, the reservoir; downstream, the valley and its river.	Skim the crest and dive down the face (*dam dive*).	North, 3 m/s · yes



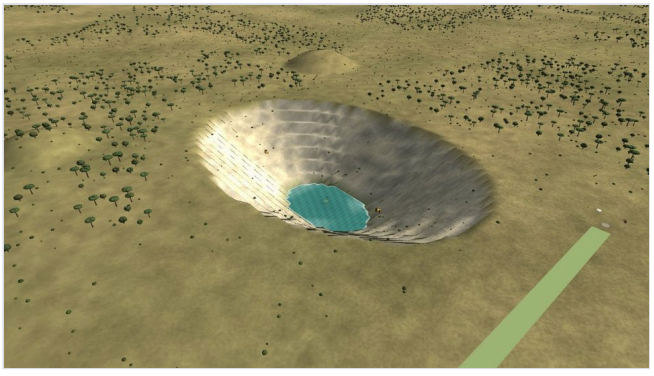
Coastal cliff.



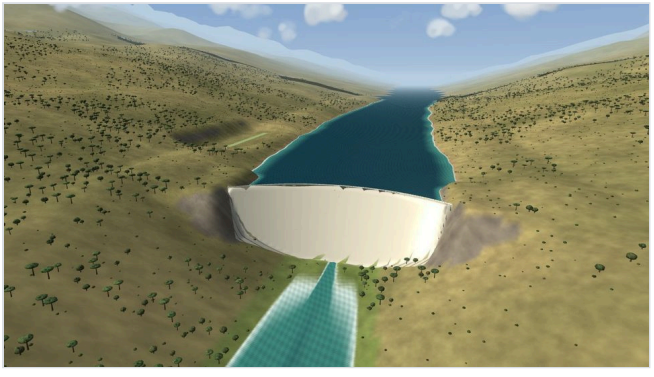
Mountain and valley: the slope and the river.



River canyon.



Quarry.



Dam and reservoir.



Mountain: the takeoff saddle and the summit.

What changes when you fly them

- **The relief is real ground**: the physics uses the same relief that is drawn. You crash into slopes, walls, the dam and the water. Trees and the canyon footbridge are obstacles too.
- **Ridge lift**: with wind, air hitting a slope rises (and sinks behind the crest, with turbulence). The closer to the slope and the more upwind, the stronger the lift; it fades with height. Choosing a scenery sets the wind that gives lift on its main slope; you can change it in Options › Time and weather.
- **Plan**: the planner shows the scenery map. In sceneries with relief, take terrain height into account when setting waypoint altitudes.
- **Real terrain**: if you load your field's real terrain, the simulator goes back to the countryside.

Real terrain of your field

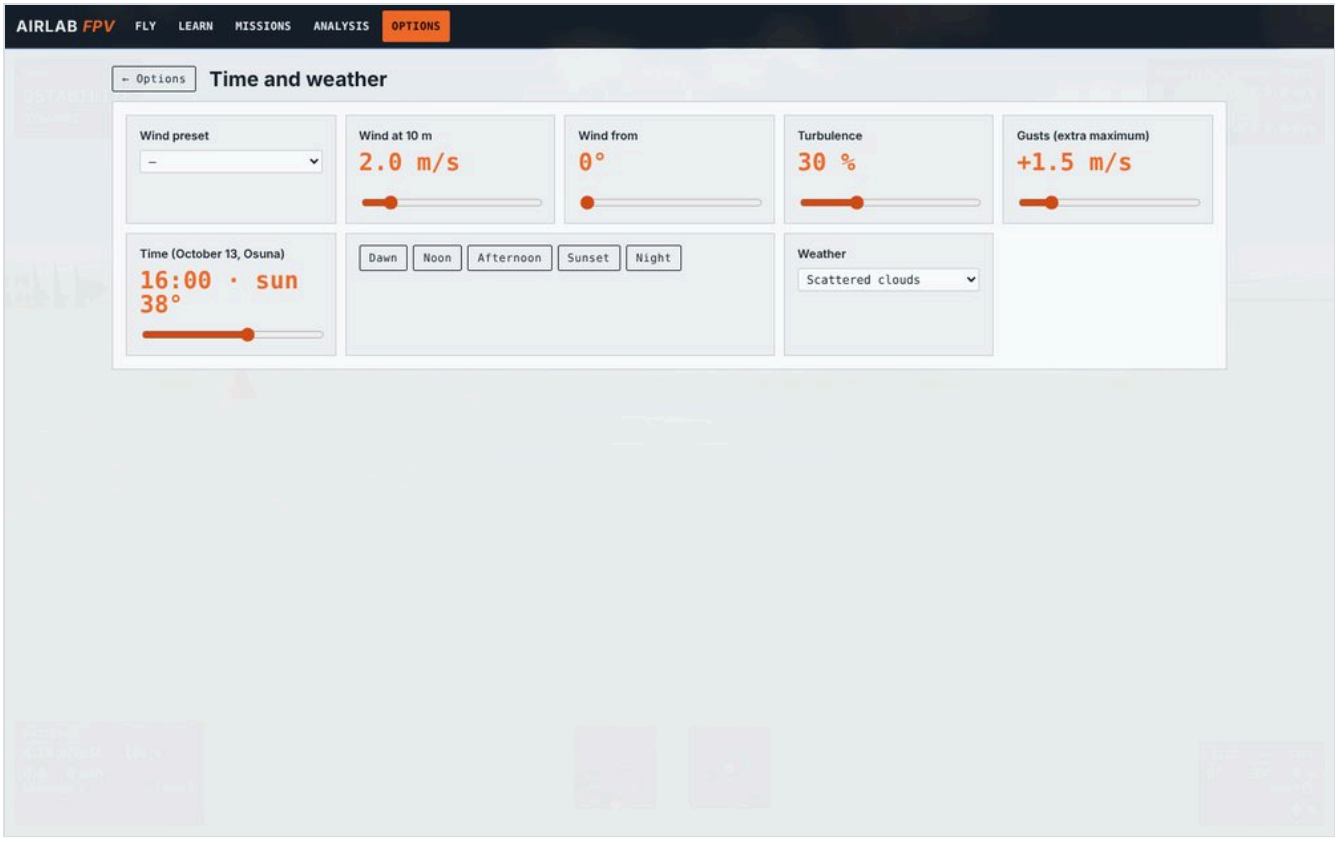
In Options › Scenery you can also load the **real terrain** of your flying field: PNOA orthophotos from the IGN (Spanish National Geographic Institute) and real elevation data covering 31 × 31 km around the takeoff point. Pick a preset area, search anywhere in the world or type or paste your field's coordinates. The orthophoto is PNOA in Spain and open sources in France, Switzerland, the Netherlands and the USA; elsewhere, real elevation with a generated texture (chapter 34).

At night. By default, at night you fly over the simulated scenario, which has the village, runway and car lights (the orthophoto is a daytime photo with no lights); the real terrain comes back at dawn. If you tick **Keep the real terrain at night**, the relief and orthophoto stay at night, darkened by the moonlight and without the simulated village lights. The planner always shows the area orthophoto if it has been downloaded; if at night you are flying over the simulated scenario, it warns you, because the terrain profile is then flat.

More detail near the aircraft. The real terrain loads the orthophoto in three rings around home: the detail ring reaches about 700 m at roughly 1 m per pixel, the surroundings ring about 3 km and the horizon ring about 15 km. With **High-resolution orthophoto around the aircraft**, two patches that follow the aircraft are laid on top. The first covers 1.2 km at orthophoto level 18 (≈0.5 m per pixel). The second covers 480 m at level 19 (≈0.25 m per pixel), only below 300 m above ground. They reload as you move away from their centre. It uses more graphics memory and more downloads, and it is turned off with light graphics. The tiles come from the orthophoto server (in Spain, IGN's PNOA) straight to the browser.

3D buildings. With **3D buildings**, the OpenStreetMap buildings within 1.5 km of home are raised. The footprint is the real one. The height comes from the *height* tag or the number of floors × 3 m, and if neither is present, from a typical height for its type. The roof shows the orthophoto and the façades have windows per floor; sheds and warehouses have none. In Spain many OpenStreetMap buildings come from the Cadastre. They are downloaded once per site (Overpass API) and kept in the browser. Hitting a building is a crash, as with trees. They do not change the height above terrain used by the mission: ArduPilot, with TERRAIN, does not know about buildings either. Buildings © OpenStreetMap contributors (ODbL).

Time and weather



Options › Time and weather.

- **Wind**: preset wind (calm, breeze, moderate, gusty, strong) or fine-tune the speed at 10 m, the direction it blows from, turbulence and gusts. Wind increases with altitude, and there is a wake behind obstacles.
- **Time of day**: from dawn to night. When the sun is below the horizon, the aircraft, field and town lights turn on. The `N` key toggles between 14:00 and 23:00.
- **Weather**: clear, scattered clouds, overcast, fog or rain.

15 Simulated failures and failsafes

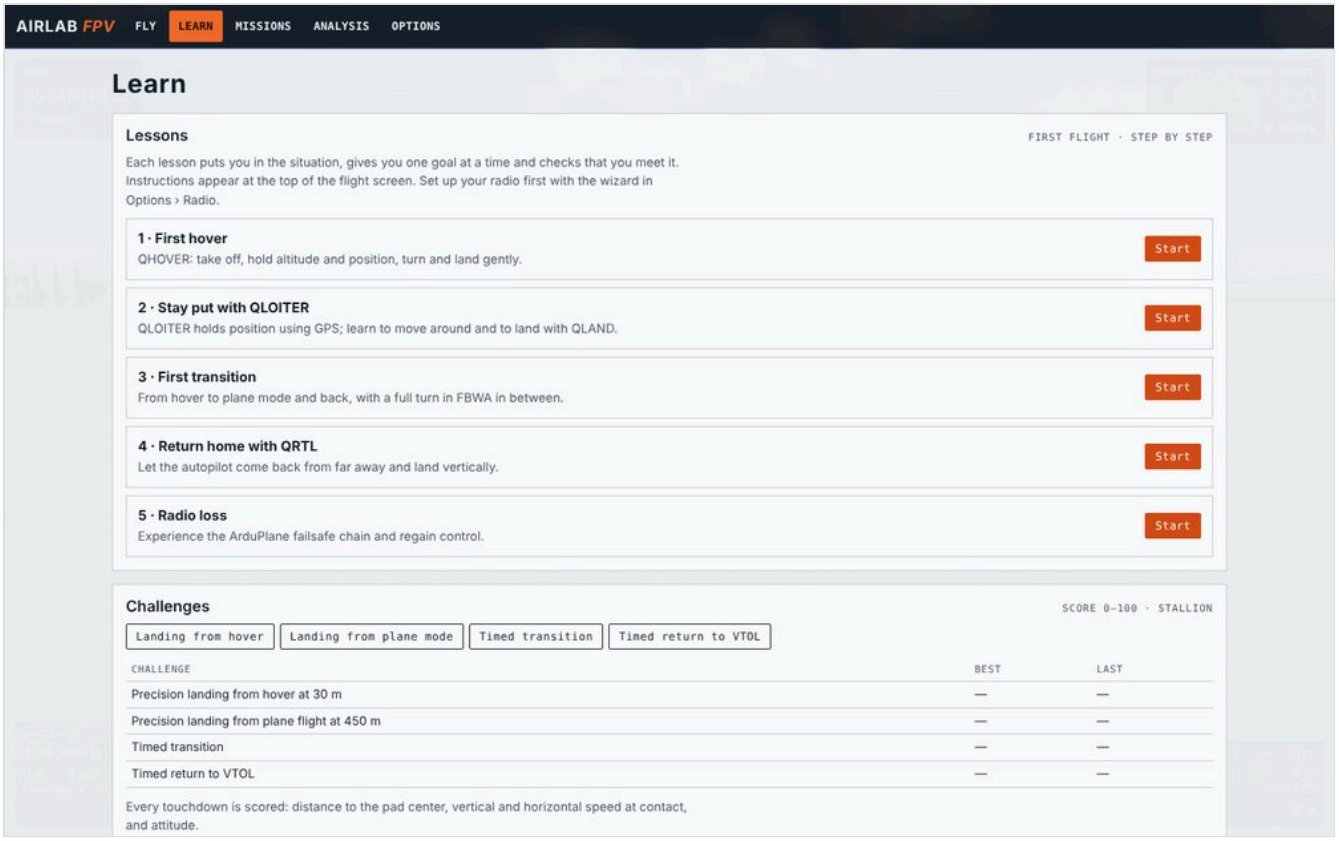
Failure	What to practice
Front left, front right or rear motor stopped	Reacting to a motor loss in hover: switch to airplane flight if you have airspeed, or land as soon as possible.
Left or right tilt servo stuck	Asymmetry during the transition; a controlled return to VTOL.
Radio signal loss	The complete failsafe sequence (see below).
Damaged cell	Higher internal resistance and 40% less capacity: the voltage drops sooner.
Airspeed sensor reading zero or frozen	Blocked or faulty pitot: fly by attitude and throttle without trusting the airspeed.
Video: heavy interference	−18 dB on the video link: fly home with a poor image.
GPS: no position	Like ArduPlane's EKF failsafe: in QLOITER, QRTL or QLAND it switches to QHOVER (you have roll and pitch to fly home); in the VTOL part of an AUTO mission, to QLAND; in airplane modes nothing changes. While it lasts you cannot enter the modes that need a position. On an ArduCopter multirotor it lands (LAND), and in Betaflight GPS Rescue cannot return. (ArduPilot · EKF Failsafe)

Surprise failure triggers a random one within 60 seconds, and **Repair all** clears them.

The ArduPlane failsafe sequence

1. After 1.5 s without radio, the **short failsafe** kicks in: QLAND from the Q modes, and CIRCLE from FBWA, FBWB, ACRO and MANUAL (per `FS_SHORT_ACTN`).
2. After `FS_LONG_TIMEOUT` seconds, the **long failsafe** kicks in: RTL (per `FS_LONG_ACTN`).
3. AUTO continues with the mission.
4. When the radio link comes back, the aircraft stays in the failsafe mode until you move the mode switch.

16 Learn: lessons and challenges



Lessons and challenges.

Each lesson puts you in the situation, gives you one objective at a time at the top of the screen and checks that you complete it. With voice enabled, the instructor reads out each step. The lessons use the Stallion VTOL.

1 · First hover

QHOVER, arm, take off to 10 m, stay between 8 and 12 m near the landing pad for 15 s, yaw 90° with the rudder, land at less than 1 m/s and disarm.

2 · Holding position with QLOITER

QLOITER, climb to 15 m, let go of the sticks and stay within 4 m of the spot for 20 s, move 30 m and stop, then land with QLAND.

3 · First transition

From QHOVER at 30 m, switch to FBWA with throttle above half, fly straight and level for 20 s between 25 and 80 m, make a full circuit without dropping below 20 m, switch back to QHOVER and land on the pad.

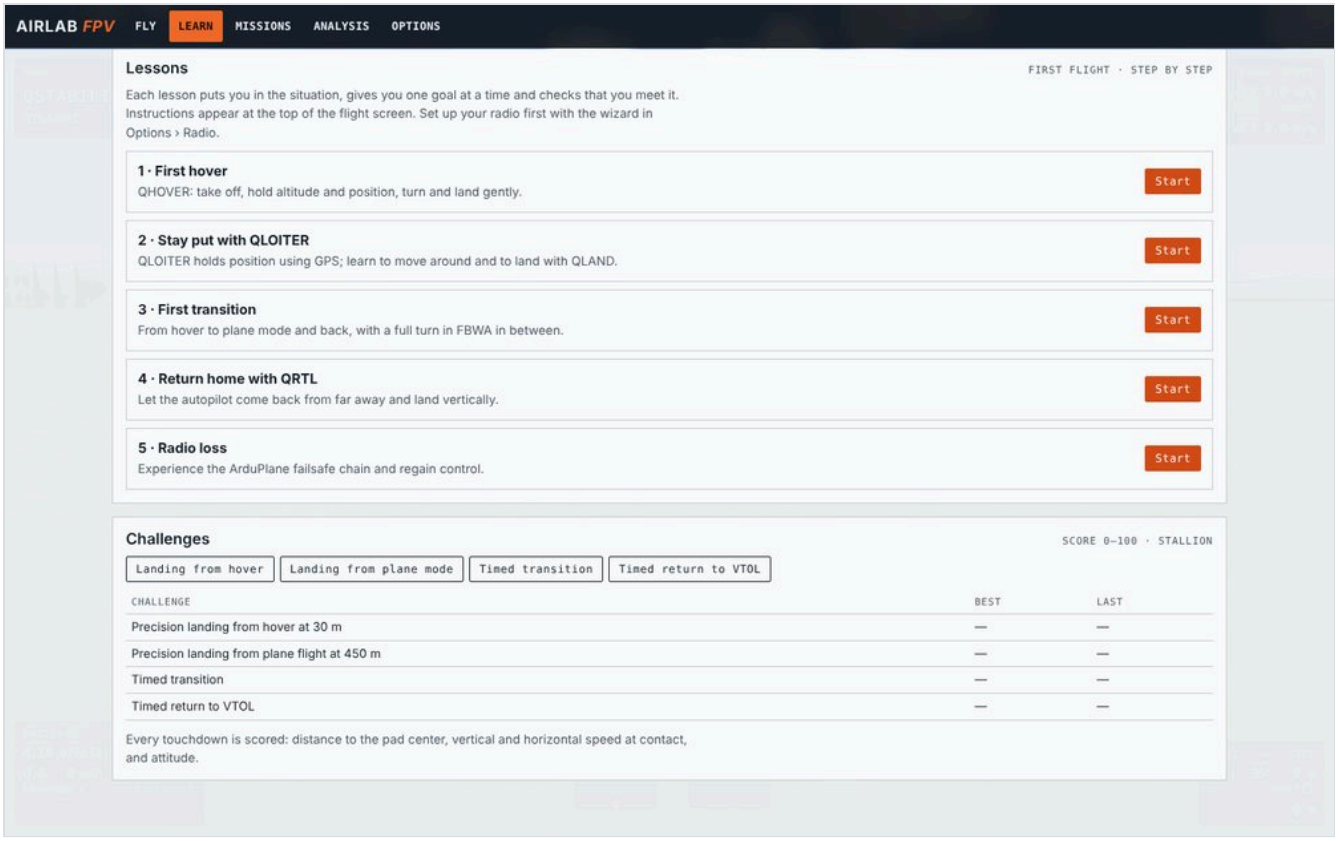
4 · Return to home with QRTL

Flying as an airplane 450 m from home, select QRTL and let it fly back, transition to VTOL and land vertically.

5 · Radio loss

While flying in FBWA the radio cuts out: let go of the controls, watch the short and long failsafes, regain control by moving the mode switch and land.

Challenges



Scored challenges.

Challenge	What is scored
Landing from hover (30 m)	Distance to the center of the pad and vertical and horizontal speed at touchdown.
Landing from airplane flight (450 m)	Same as above, starting in airplane flight far from home.
Timed transition	Time to complete the transition (points deducted after 8 s), altitude lost and maximum roll above 10°.
Timed return to VTOL	Time to come to a stop in hover (points deducted after 10 s), distance traveled beyond 80 m and altitude lost.

Scores range from 0 to 100; a crash scores 0. Your best score for each challenge is saved.

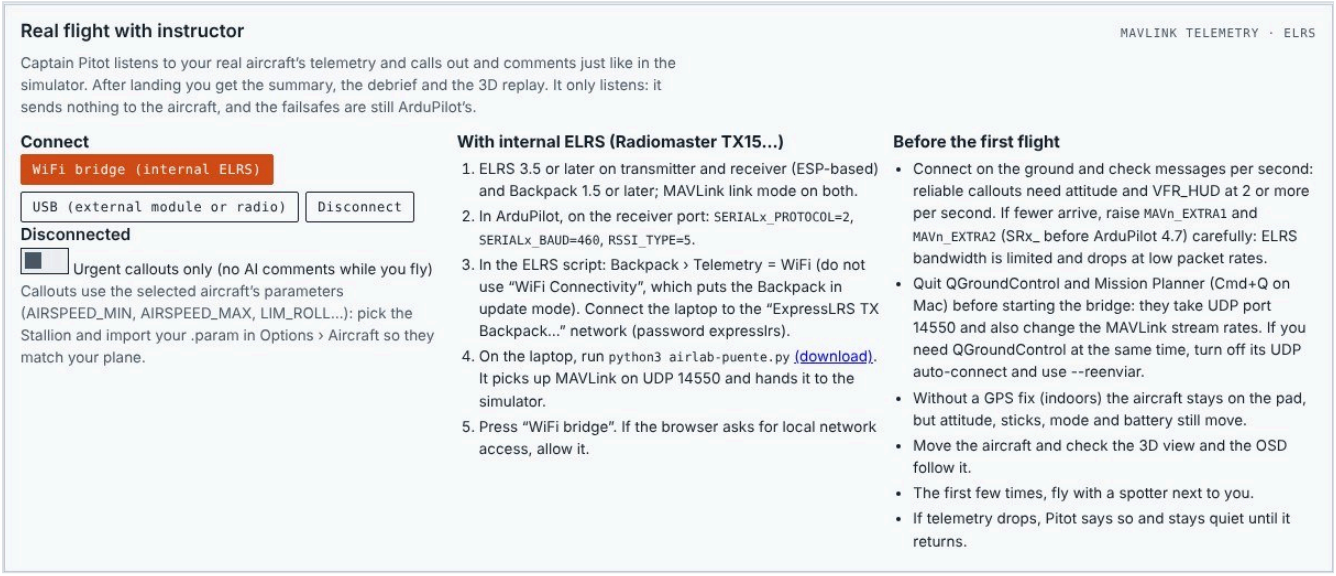
Real flight with instructor

The **Real flight with instructor** panel (Learn) connects the simulator to your real aircraft's MAVLink telemetry. While you fly at the field, Captain Pitot calls out and comments just as in the simulator, the 3D view and the OSD follow the aircraft, and the black box records the flight: after landing you get the

summary, the debrief, the questions to the instructor and the 3D replay. It only listens: it sends nothing to the aircraft, and the failsafes are still ArduPilot's.

Connection	When	How
WiFi bridge	Internal ELRS module in the radio (for example, Radiomaster TX15)	The ELRS Backpack sends MAVLink over WiFi on UDP 14550. A browser cannot open UDP, so **AIRLAB Connector** picks it up: a program on the computer that you install once and open with a double-click (chapter 27). In the ELRS script, set **Backpack › Telemetry** to WiFi (do not use "WiFi Connectivity"), connect the laptop to the "ExpressLRS TX Backpack..." network (password `expresslrs`) open the connector and press **WiFi bridge**; if the browser asks for local network access, allow it. **Connect** tells you live what is missing.
USB	External ELRS module with USB or a telemetry radio	Plug it into the laptop and press **USB**: Chrome or Edge open the port at 460,800 baud (Web Serial). Nothing to install.

ELRS in MAVLink mode needs version 3.5 or later on transmitter and receiver (ESP-based) and 1.5 or later on the Backpack, with **Link Mode** set to MAVLink. In ArduPilot, on the receiver port: `SERIALx_PROTOCOL` = 2, `SERIALx_BAUD` = 460, `RSSI_TYPE` = 5 and `MAV_GCS_SYSID` = 255. Chapter 26 explains the whole process step by step: updating the receiver, setting up radio, Backpack and ArduPilot, the bridge on the Mac and the problems that may come up.

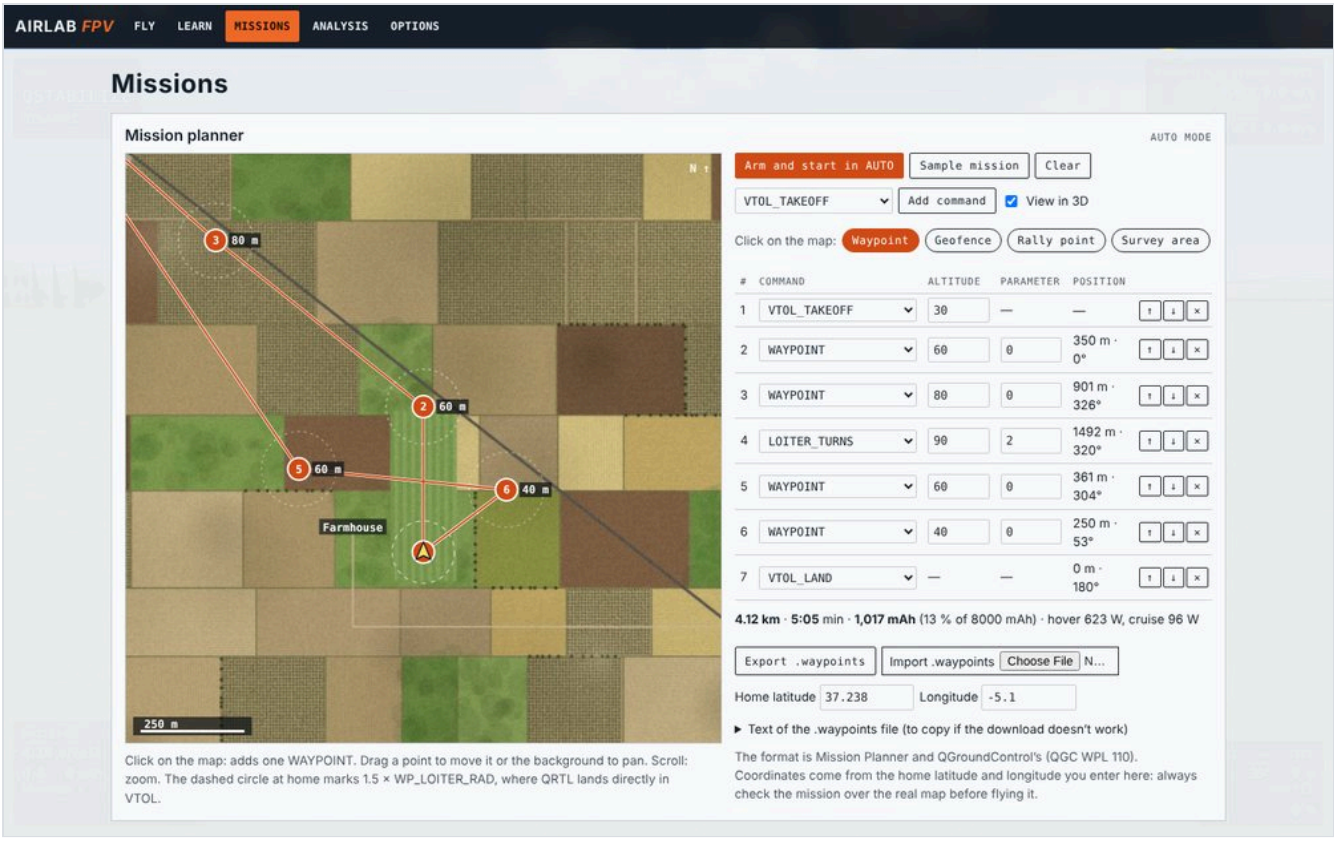


Connect › Real flight with instructor.

- **Before flying**, connect on the ground and check the panel's messages per second: reliable callouts need attitude and VFR_HUD at 2 or more per second (`MAVn_EXTRA1` and `MAVn_EXTRA2` in ArduPilot 4.7; `SRx_` in earlier versions). ELRS bandwidth is limited and drops at low packet rates.
- **Quit QGroundControl and Mission Planner** before opening the connector (`Cmd`+`Q` on Mac): they take UDP port 14550 and also change the MAVLink stream rates. If you need QGroundControl at the same time, turn off its UDP auto-connect and open the connector from Terminal with `python3 ~/AIRLAB/airlab-conector.py --reenviar 14560`.
- **Without a GPS fix** (indoors) the aircraft stays on the pad, but attitude, sticks, mode and battery still move. With a GPS fix, the takeoff point becomes home.

- With the laptop on the Backpack WiFi there is no internet for the AI: share your phone's data over USB (chapter 26).
- Callouts use the selected aircraft's parameters (`AIRSPEED_MIN` , `AIRSPEED_MAX` , `LIM_ROLL` ...): pick the Stallion and import your .param file so they match your plane.
- **Urgent callouts only** turns off the AI comments while you fly.
- If telemetry drops, Pitot says so and stays quiet until it returns.
- With the real terrain of your field loaded, the aircraft appears in place over the orthophoto.
- The first few times, fly with a spotter next to you.

17 Missions in AUTO mode



Planner with the sample mission.

- **Add points**: click on the map. Drag to pan the map or move a waypoint; use the scroll wheel to zoom.
- **Commands**: VTOL_TAKEOFF, TAKEOFF, WAYPOINT (with acceptance radius), LOITER_TURNS (turns), LOITER_TIME (seconds), DO_VTOL_TRANSITION (3 = VTOL, 4 = airplane), VTOL_LAND, LAND, DO_CHANGE_SPEED, DO_JUMP, DO_LAND_START and RTL. The table below explains the new ones.
- **Table**: edit the altitude and parameters of each command, reorder them or delete them.
- **Summary and warnings**: estimated distance, time and energy, with warnings if the mission is too long for the battery or has inconsistencies.
- **View in 3D**: draws the route in the scenery.
- **Arm and start in AUTO**: shows the operation summary (31.6), then arms and starts the mission. You can change mode at any time to take back control.
- **Import and export .waypoints**: Mission Planner format, including the home position (latitude and longitude) so it matches your field.

More commands

Command	Parameters	What it does
DO_CHANGE_SPEED	Airspeed (m/s)	Changes the target speed for the following airplane legs (instead of `AIRSPEED_CRUISE`).

DO_LAND_START	—	Marks where the landing sequence starts (place it before the approach waypoint to the LAND). With `RTL_AUTOLAND` 1 or 2, RTL jumps here and lands. In normal AUTO it does nothing.
DO_JUMP	Target command and repeat count (−1 = forever)	Jumps to another command to repeat part of the mission, for example a circuit. The distance and battery estimate does not count the repeats: the planner warns you about this.
TAKEOFF	Altitude	Airplane takeoff. On the fixed wing, the aircraft is hand launched 2 s after arming and climbs to that altitude with `TKOFF_LVL_PITCH`. On a VTOL it is skipped: use VTOL_TAKEOFF.
LAND	Position	Airplane landing, a belly landing: it descends along a glide path from the previous point to the landing point and, at `LAND_FLARE_ALT` or `LAND_FLARE_SEC` seconds above the ground, stops the motor and flares, aiming for a sink rate of `TECS_LAND_SINK`. Once it comes to a stop, it disarms. Allow a long approach on a 1:10 glide path: from 30 m altitude, place the previous waypoint about 300 m out.

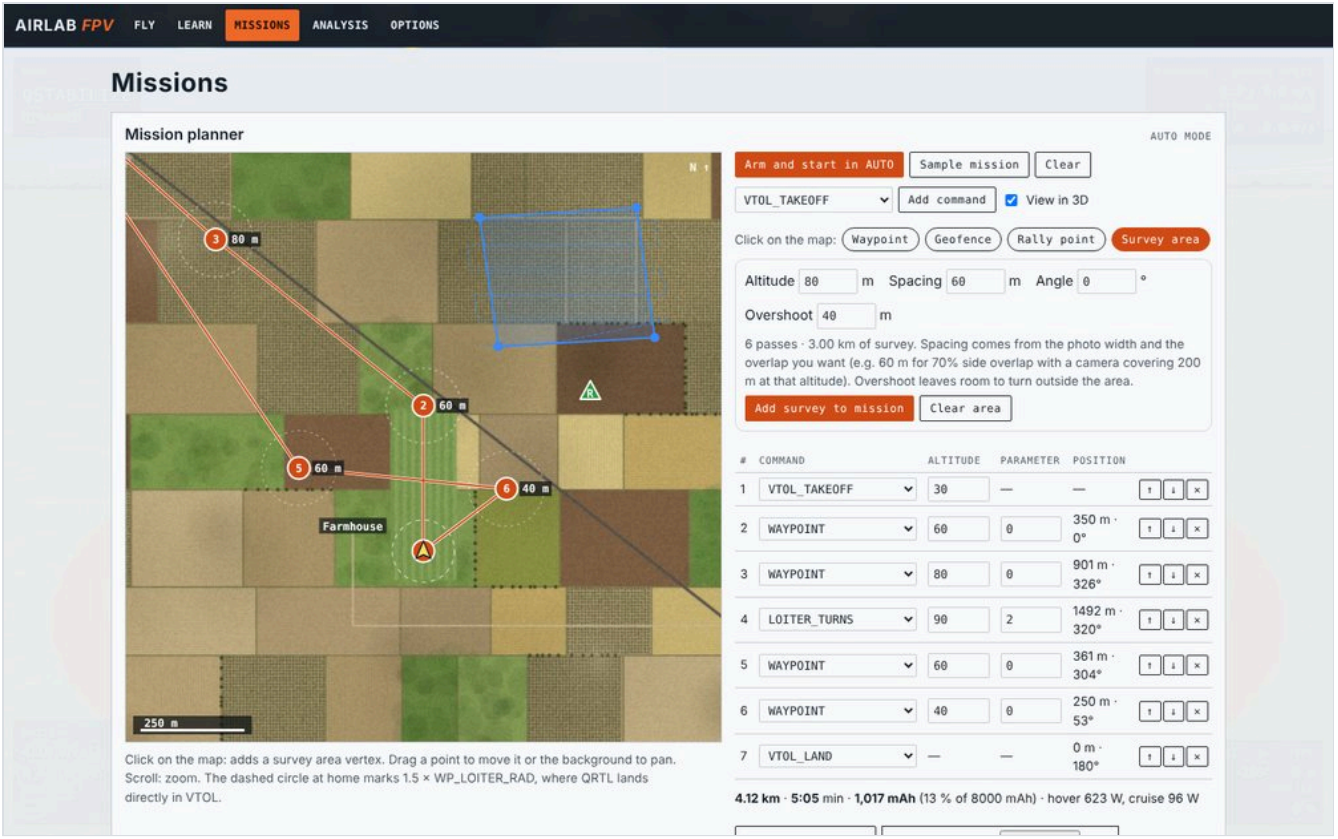
All of them are exported and imported with their parameters in the .waypoints file.

How the route is followed: the L1 controller

On the aeroplane legs, the autopilot follows the route like ArduPlane's L1 controller: it aims at a point on the route an L1 distance ahead and banks towards it. That is why the aircraft's track never quite matches the planned line: it converges on it gradually, cuts the corners and, in a crosswind, sits slightly off it. Tune it in **Parameters › Navigation and return home**, with the same names as in Mission Planner or QGroundControl:

Parameter	Default	What it does
`NAVL1_PERIOD`	17 s	Sets the L1 distance (L1 = damping × period × speed / π; about 77 m at 19 m/s). Lower: tracks the line more tightly and turns earlier. Too low: it weaves. On the real aircraft, lower it 1 at a time on an AUTO circuit until it starts to weave, then raise it one or two steps.
`NAVL1_DAMPING`	0.75	Damping. Rarely changed.
`NAVL1_XTRACK_I`	0.02	Crosstrack error integrator: gradually removes a steady offset. 0 turns it off.

A waypoint counts as reached at `WP_RADIUS` (or the command's radius), but never further out than the L1 distance, as in ArduPlane. The simulator's roll response is more ideal than a real aircraft's, so weaving appears at lower values: use it to see the trend, not as the final value for your aircraft.



Geofence (red circle), a rally point (R) and a survey area (blue) with its passes.

What the map shows besides the mission. The thin yellow line is the track of the last flight and the coloured patch is the coverage of the last photo capture. **Clear** removes the mission and the track; the coverage stays because it is the capture you process in the lab (hide it by unticking *Show coverage of the last capture* in Photogrammetry and remote sensing; it is replaced when another capture starts).

What a map click adds

Above the table you choose what each click adds: **Waypoint**, **Geofence**, **Rally point** or **Survey area**. Each tool opens its own settings panel.

Terrain profile and height above terrain

Below the planner warnings there is the **mission profile**: the terrain along the whole route (the same relief as the scene, SRTM/EU-DEM with real terrain loaded), the dashed safety-margin line and the height the aircraft would fly at. It takes into account how fast the aircraft can climb (`TECS_CLMB_MAX` at cruise speed). Where the aircraft would pass below the margin it is marked in red, and where it would reach the ground, in dark grey; the warnings say between which commands and at what distance it happens. The margin is set at the top (30 m by default) and must cover trees and buildings, which the elevation model does not include.

With a mission in height above home, the aircraft keeps that altitude even if the terrain rises: on a hillside the height above ground drops, photos spread out with no overlap and, if the terrain rises more than the margin, it crashes. To replan:

Button	What it does

Follow terrain	All waypoints switch to **height above terrain**, with the same figure: 83 m above home becomes 83 m above whatever ground is below. It is ArduPilot's *Terrain* frame (MAV_FRAME_GLOBAL_TERRAIN_ALT, 10 in the .waypoints file). Between waypoints the aircraft follows the relief and, with `TERRAIN_LOOKAHD`, looks at the upcoming terrain (not past the next waypoint) to start climbing earlier. On the real aircraft you need `TERRAIN_ENABLE` = 1 and a ground station that sends it terrain data.
Climb just enough	Keeps height above home and raises the waypoints of the legs without margin until the whole route is at least that far from the ground. The photo GSD will change with the terrain.
Height above home	Returns the waypoints to height above home.

In the mission table, each WAYPOINT has a **home / terrain** selector next to its altitude to choose it one by one; in the profile, waypoints above terrain are green. The survey (here and in the photogrammetry panel) has the **height above terrain** box, ticked by default when real terrain is loaded: that way height above ground, GSD and overlap stay the same even if the area slopes. The photogrammetry panel says how much relief the area has and, if you don't follow terrain, between which heights above ground it will fly.

Geofence

- **Maximum radius** from home and **maximum altitude**, like ArduPlane's `FENCE_RADIUS` and `FENCE_ALT_MAX`. A value of 0 disables that limit.
- Optional **inclusion polygon**: click on the map to add vertices; the aircraft must stay inside.
- **On breach**: **RTL** (`FENCE_ACTION` 1) or warn only (`FENCE_ACTION` 0). On drones it only warns. The warning appears on screen, in the log and by voice.
- The planner flags waypoints that fall outside the fence, and the fence is also drawn in the 3D scenery.

Rally points

Click on the map to add rally points. In RTL, the aircraft flies to the nearest one, or home if that is closer, at `RTL_ALTITUDE`, and loiters there (or lands vertically if `Q_RTL_MODE` = 1). They give you alternative places to return to if home is far away or behind an obstacle. They are shown on the map and in 3D as green triangles with an R.

Survey area (photogrammetry)

Click to mark the vertices of the area you want to cover, then adjust:

Setting	What it is
Altitude	Flight altitude for the passes.
Spacing	Distance between passes. It comes from the width covered by each photo at that altitude and the side overlap you want: with a camera covering 200 m and 70% overlap, 60 m.
Angle	Direction of the passes (0° = west to east). It usually works best to run them along the longest side of the area or perpendicular to the wind.
Overshoot	How far each pass extends beyond the area so the aircraft can turn and enter the next one straight.

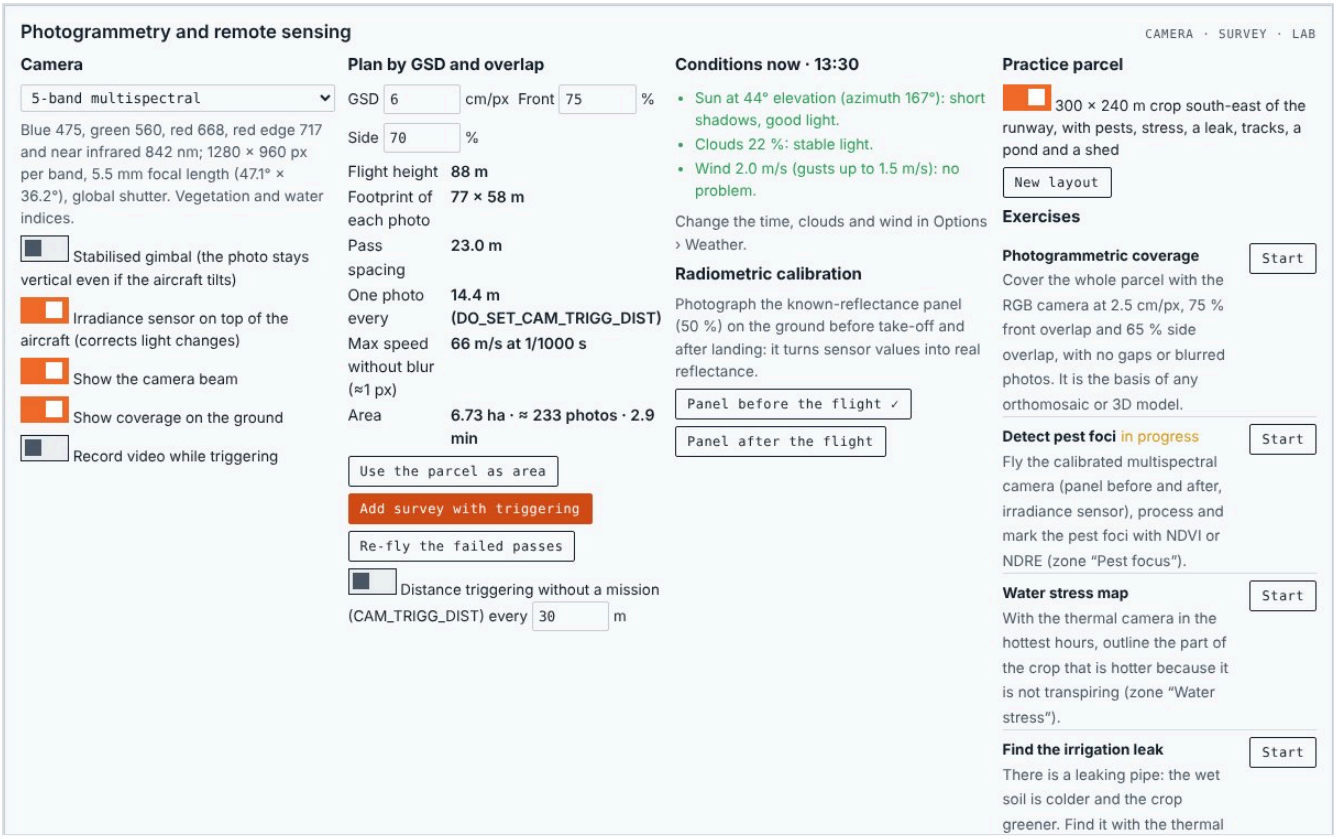
With camera trigger (on by default) makes the survey take photos: altitude and spacing come from the camera, GSD and overlaps in the **Photogrammetry and remote sensing** panel (chapter 18), and **Add survey with trigger** inserts the passes with their `DO_SET_CAM_TRIGG_DIST` (per pass if «Trigger only inside the area» is on), just like that panel's button. Unticked, you set altitude and spacing by hand and **Add survey to the mission** inserts only the waypoints, like Mission Planner's *Simple Grid*: the camera never turns on (unless `CAM1_TRIGG_DIST` is set) and there are no photos or scan. Either way the survey goes before the last RTL, VTOL_LAND or LAND; if the mission is empty, it adds a takeoff before it and the mission end after.

If the mission has a survey without any `DO_SET_CAM_TRIGG_DIST` and `CAM1_TRIGG_DIST` is off, the warnings say **«This survey will not take photos»**. The **Add the camera trigger** button replaces it with the same survey with trigger, in the same place in the mission; whatever came after it (other waypoints, the landing) stays after it.

18 Photogrammetry and remote sensing

The simulator carries three cameras to practise drone mapping jobs: plan a survey by GSD and overlap, fly it in AUTO with distance triggering, check photo quality, process an orthomosaic and study it with vegetation indices, thermal and material classification. There is a crop parcel with hidden problems and five scored exercises. Everything can be saved as a PDF report, an interactive web page, GeoJSON and video.

> **Simulated data**
>
> Values follow real behaviour (reflectance of each band, surface temperature, sunlight, blur), but they are for practice: they do not come from a real field.



Plan › Photogrammetry and remote sensing: camera, plan by GSD and overlap, conditions, calibration, practice parcel, exercises and capture.

Where it is

- **Plan**, below the planner: the **Photogrammetry and remote sensing** panel, with everything you prepare before flying. The exercises can also be started from **Missions › Remote sensing exercises**.
- **The lab**, where captures are studied: open it with **Process and open the lab** in that panel or from **Analysis › Remote sensing**.

The cameras

Camera	Sensor	What for

RGB 20 MP	1″, 5472 × 3648 px, 8.8 mm focal length (73.7° × 53.1°), rolling shutter and auto exposure	Colour orthomosaics, 3D models, visual inspection
5-band multispectral	Blue 475, green 560, red 668, red edge 717 and near infrared 842 nm; 1280 × 960 px per band, 5.5 mm focal length, global shutter at 1/1000 s	Vegetation indices (NDVI, NDRE…), water, materials
Thermal 640 × 512	12 µm microbolometer, 13 mm focal length (32.9° × 26.6°); it takes about 7 ms to respond, so it blurs before the others	Water stress, leaks, moisture, materials

The camera is fixed to the airframe and looks down: when the aircraft banks in a turn or climbs nose-high, the photo is oblique. With **Stabilised gimbal** the photo stays vertical up to 35° of tilt. With the multispectral camera you can carry the **irradiance sensor** on top of the aircraft (see Calibration).

Planning by GSD and overlap

The **GSD** (ground sampling distance) is the size on the ground of each photo pixel: GSD = pixel size × height / focal length. You type the GSD you need and the overlaps, and the panel works out the rest:

Value	How it is worked out
Flight height	The height that gives that GSD with the selected camera. For example, 2.5 cm/px with the RGB is 91 m; 6 cm/px with the multispectral, 88 m; 11 cm/px with the thermal, 119 m. It warns above 120 m.
Footprint of each photo	What one photo covers on the ground (width × length).
Pass spacing	Footprint width × (1 − side overlap).
One photo every	Footprint length × (1 − front overlap). It is the `DO_SET_CAM_TRIGG_DIST` distance.
Max speed without blur	The speed that keeps blur around 1 px with the camera's exposure. If the aircraft flies faster, the survey slows down with `DO_CHANGE_SPEED`.
Area, photos and time	With the survey area drawn: hectares, approximate photos and minutes.

Usual values: 70–80 % front and 60–70 % side overlap for an orthomosaic; more (80/75 %) for 3D models or very uniform crops.

Add survey with triggering puts into the mission, before the last RTL or landing:

1. The first waypoint of the first pass.
2. `DO_CHANGE_SPEED`, if it has to fly slower to avoid blur.
3. `DO_SET_CAM_TRIGG_DIST` with the distance between photos: from here the camera triggers by itself every that many metres.
4. The rest of the passes, alternating direction.
5. `DO_SET_CAM_TRIGG_DIST` = 0, which stops the camera, and `DO_CHANGE_SPEED` back to cruise.

With an airplane, the overshoot is set to about 1.6 turn radii so it enters each pass straight. **Use the parcel as area** sets the practice parcel as the survey area. When exported to .waypoints,

`DO_SET_CAM_TRIGG_DIST` goes as command 206 with the distance in parameter 1, as in Mission Planner and QGroundControl.

Distance triggering without a mission is ArduPilot's `CAM1_TRIGG_DIST` parameter (before 4.5, `CAM_TRIGG_DIST`): the camera triggers every N metres while you fly in any mode. It is also in Options › Aircraft, **Camera** group, with two other real parameters:

Parameter	What it does
`CAM_MAX_ROLL`	Postpones the photo while the roll is greater than that angle and takes it as soon as the aircraft levels. 0 = triggers at any roll. With 15° there are no tilted photos in turns (those over 15° are useless for the mosaic); the panel has a button to set it.
`CAM_AUTO_ONLY`	1 = distance triggering only works in AUTO; 0 = in any mode.

Trigger only inside the area (ticked by default): instead of turning the camera on at the start of the survey and off at the end, each pass has a waypoint half a footprint before entering the area with `DO_SET_CAM_TRIGG_DIST` = the distance between photos and another half a footprint after leaving with `DO_SET_CAM_TRIGG_DIST` = 0. That way there are no photos in the overshoot, the turns or the legs between passes (what QGroundControl does without "Images in turnarounds"). In a test over the parcel with the plane: the usual survey took 99 photos, 40 outside the area; per pass and with `CAM_MAX_ROLL` = 15°, 60 photos, 3 outside, with the same 100 % coverage.

Survey as a plane or in VTOL

A plane cannot turn around into the adjacent pass if passes are closer than its turn diameter (2·r, with r = $v^2/(g \cdot \tan \varphi)$, the ground speed and LIM_ROLL). If it tries, the L1 overshoots the pass, considers the waypoint reached when it crosses its finish line (as ArduPlane does) and finds the next one behind it: it loops towards the opposite side and enters diagonally across the area. That is why, with a plane, the survey is ordered **skipping passes**, in blocks 0, k, 1, k+1, 2, k+2..., with k chosen so each jump is at least 2.2 turn radii (cruise plus 70 % of the wind). The few passes that do not fit change side with a turn-around waypoint three radii beyond the end, outside the area. Multirotors and VTOL surveys stay in zigzag, pass after pass.

The **Survey** selector (tilt-rotor and 4+1 only) chooses how it is flown: **as a plane**, **in VTOL** or **automatic**, which switches to VTOL when the blur-free speed is lower than the plane's minimum safe speed (AIRSPEED_MIN + 1). That is the usual case with the thermal camera: at 1/150 s and 7.7 cm/px, ≈ 1 px of blur requires about 12.7 m/s over the ground, and the Stallion does not fly as a plane below 17 m/s. In VTOL the aircraft reaches the first pass as a plane, switches to VTOL (DO_VTOL_TRANSITION), sweeps at Q_WP_SPEED (lowered if it is above the blur-free speed) and switches back to plane at the end. There is no blur and no loops, but it uses much more battery: when the survey is added the log gives the time and battery percentage, and warns if there is not enough with reserve (reduce the area, raise Q_WP_SPEED up to the blur-free speed, raise the GSD or split it into two flights).

End of the mission. On a VTOL, if the mission had no landing, the survey ends it with `VTOL_LAND` at home: it flies back as a plane, transitions and lands vertically. An `RTL` with `Q_RTL_MODE` = 0 (ArduPlane's default) reaches home as a plane and keeps circling at `WP_LOITER_RAD` without landing; with `Q_RTL_MODE` = 1 it lands in VTOL on arrival. The planner warns if the mission ends with RTL and `Q_RTL_MODE` = 0.

Live scan

As soon as the camera takes the first photo of the flight, the **SCAN** box appears on the left of the flight screen: each photo paints its real footprint on the ground (the four sensor corners projected with the

aircraft attitude) with the terrain seen vertically, so the mosaic grows pass by pass while you fly. Photos that processing would discard (blurred, oblique, no light) are dimmed with a red border, the latest photo has an orange border, and the dashed line is what the camera sees right now. Below it, the number of valid and discarded photos and the GSD of the last one. It is a natural-colour preview: the orthomosaic, the multispectral bands and the thermal layer are computed when you process the capture in Analysis › Remote sensing. The × closes it until the next capture.

Side overlap: planned and measured

Planned. The planner takes the mission legs with the camera on (between a `DO_SET_CAM_TRIGG_DIST` greater than 0 and the next one at 0) that run in the survey direction, pairs each pass with its neighbour and walks along them every few metres. At each point it computes the width the photo covers on the ground at the real height above terrain and the side overlap = 1 − spacing / width. If the worst point is more than 10 points below the side overlap set in Photogrammetry, it warns with the passes (in flight order), the minimum and median overlap, the cause and what to change: if the ground rises under the passes, turn on **Height above terrain** or reduce the spacing to the value shown; if the spacing is large for the footprint, reduce it, fly higher or generate the survey **with camera trigger**. If two passes do not touch at all, it reports a **gap** and how many metres long.

Measured in flight. Valid photos are grouped into passes and, when each one ends, it is compared with its neighbours *in the mission* that have already been flown. In alternate order the adjacent pass is flown later, so until then it does not count as a gap: the scan box says how many passes are still to fly. The measurement cuts each point of the pass with the real photo footprints, with their tilt and the wind drift. If the median overlap is more than 15 points below the target, or there is a gap over more than 10 % of the pass, it warns on screen and with Captain Pitot’s voice, writes it in the log and marks the gaps in red in the scan box. Those passes are repeated with **Repeat the failed passes**.

In flight: the camera beam and coverage



Survey over the parcel: the blue beam shows what the camera sees and the ground is coloured with the coverage.

- A **semi-transparent blue beam** goes from the camera to the ground and marks the photo's footprint; it flashes on every shot.
- The ground is coloured by how many valid photos see each point: **blue** 1, **cyan** 2–3, **green** 4–6, **light green** 7 or more. In **red**, what only has unusable photos.
- Coverage is also shown on the planner map, with a dot per photo.
- With **Record video while triggering**, the flight video starts and stops with the triggering.

Why a photo is discarded

Reason	When it happens	What to do
Blurred	More than 1.5 px of blur: ground speed × exposure / GSD, plus how much the aircraft rotates during the exposure.	Fly slower, raise the GSD or fly with more light (the RGB lengthens the exposure when light is poor).
Oblique	The camera tilts more than 15°: turns, nose-high climbs, gusts.	More overshoot, gimbal, less wind.
No light	Very low sun or night (does not affect the thermal camera).	Fly with the sun high.
Too low / points at the horizon	Under 5 m height, or a photo corner does not hit the ground.	—

Quality control and passes to re-fly

The **Capture** box shows live the photos, the valid ones, the real GSD, the area coverage and the overlaps measured between photos (front between consecutive photos and side between passes), with warnings about what fails: gaps, side overlap broken by wind drift, blurred or oblique photos, missing calibration.

Re-fly the failed passes adds to the mission only the passes where more than 15 % of the track has fewer than 2 valid photos, with their triggering. That completes the capture without repeating the whole flight.

Conditions

The panel rates the current time and weather for the selected camera:

- **Sun**: best above 40° elevation (short shadows). In the scenery, solar noon is at 14:07.
- **Clouds**: with passing clouds, light changes from one photo to the next and the mosaic comes out patchy unless corrected with the irradiance sensor. Overcast gives even but poor light and takes contrast away from thermal.
- **Wind**: from 5 m/s there is drift, tilted photos and broken side overlap. Better to fly the passes upwind and downwind.
- **Thermal**: temperature differences show best at peak heating, between 12:00 and 16:00 (or before dawn for leaks and moisture).

Time, clouds and wind are changed in Options › Weather.

Radiometric calibration

The multispectral camera measures how much light reaches the sensor, which depends on the sun, the clouds and each band's gain. To get **reflectance** (the fraction of light the surface reflects, comparable between flights) two things are used:

- **Reflectance panel** (50 %): photographed on the ground with **Panel before the flight** and **Panel after the flight**. It sets the scale of each band; the one after corrects light drift during the flight.
- **Irradiance sensor**: measures the light at each photo and corrects changes due to clouds.

Calibration	Result
Panel and irradiance sensor	Correct reflectance over the whole mosaic.
Panel only	Correct scale, but with patches where clouds passed.
Irradiance sensor only	No patches, but each band with a small scale error.
Nothing	Values that depend on the light of the moment: they cannot be compared with other flights or with thresholds.

The practice parcel

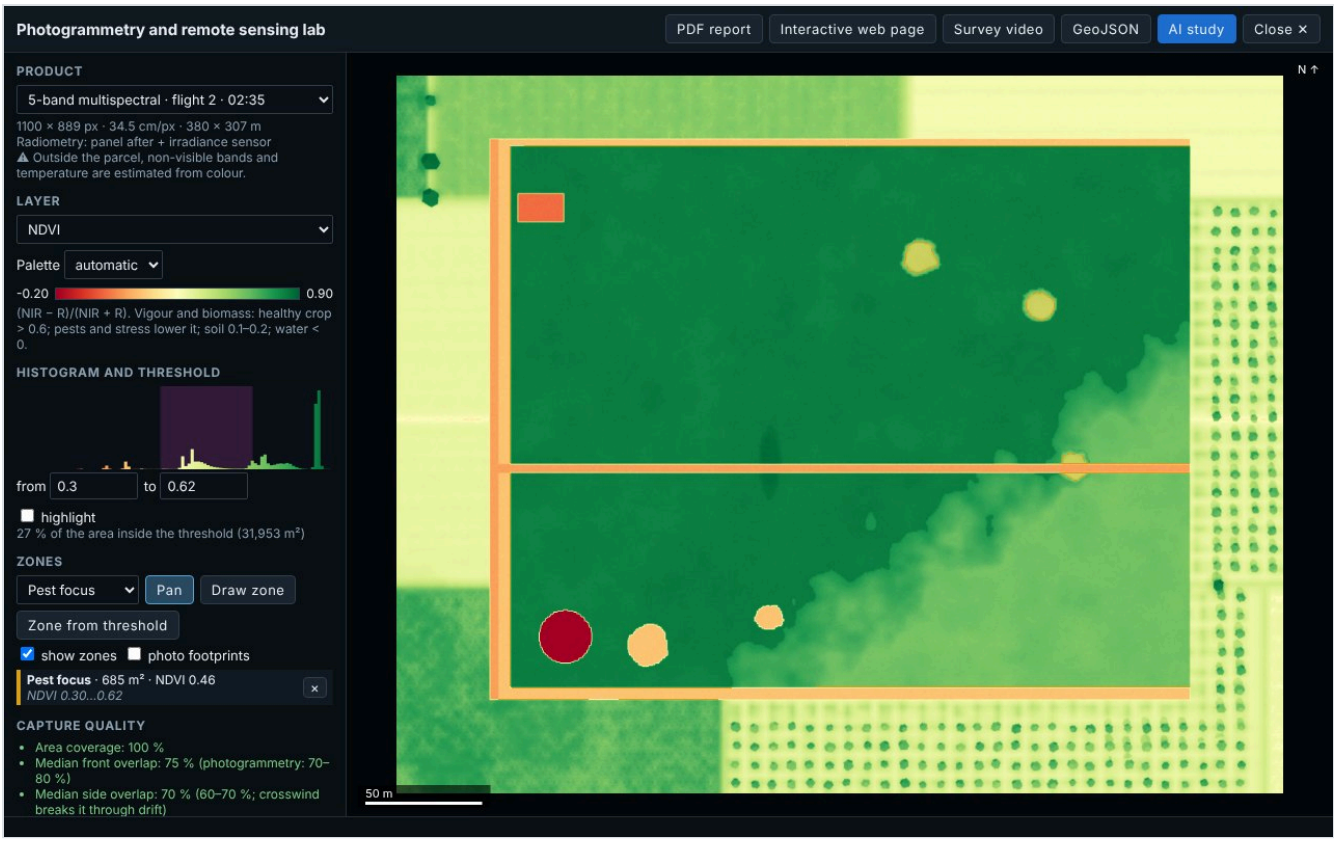
A 300 × 240 m crop south-east of the runway, with tracks, unsown headlands, a pond, a shed with a metal roof and, hidden, pest foci, a water-stressed area at the end of the irrigation sector and a leaking

pipe. **New layout** moves them around (exercises always start with a new one). Trees inside the parcel are removed while it is active.

Processing: the orthomosaic

Process and open the lab builds the orthomosaic from the valid photos: each point comes from the photo that sees it closest to vertical and with the best resolution, with that photo's light, calibration and blur. The mosaic is at most about 1100 pixels per side: over large areas, its pixel is larger than the GSD (shown at the top of the lab). Outside the parcel, the image comes from a top-down view of the scenery and the non-visible bands and temperature are estimated from colour: the lab and the reports say so.

The lab

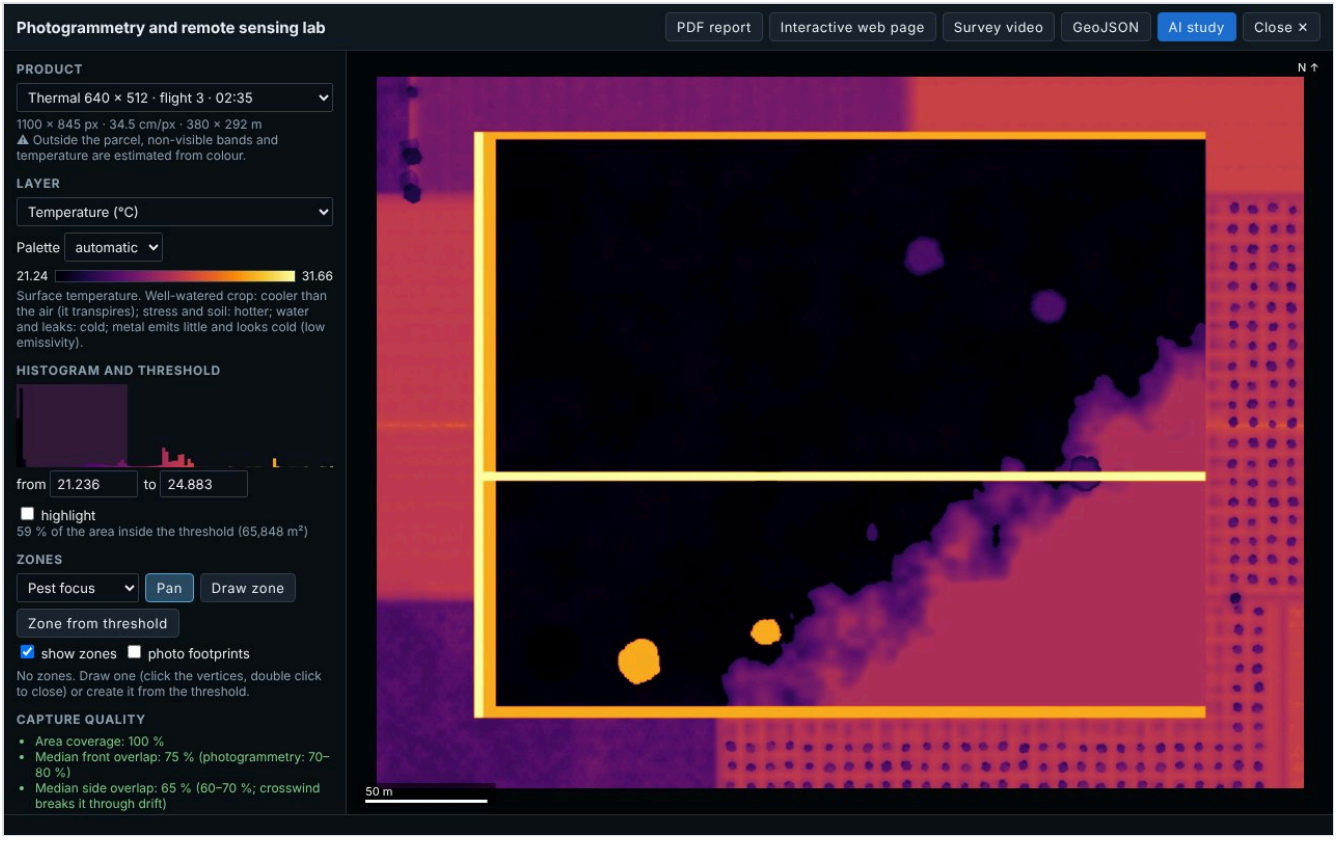


Lab with NDVI: pest foci in yellow, the zone marked from the threshold, the histogram and quality control.

Layer	Formula	What it shows
Natural colour and **false colour**	R, G, B · NIR, R, G	In false colour, healthy vegetation is bright red.
NDVI	(NIR − R)/(NIR + R)	Vigour: healthy crop above 0.6, pests and stress lower, soil 0.1–0.2, water negative.
NDRE	(NIR − RE)/(NIR + RE)	Chlorophyll; does not saturate on dense crops and detects stress earlier.
GNDVI	(NIR − G)/(NIR + G)	Chlorophyll, in late stages.
SAVI	1.5 (NIR − R)/(NIR + R + 0.5)	Like NDVI, correcting for soil when cover is sparse.

NDWI	(G − NIR)/(G + NIR)	Open water (positive). Leaf moisture needs shortwave infrared, which this camera lacks: for moisture, use thermal.
VARI, GLI, ExG	R, G and B only	Greenness with the RGB camera: indicative, uncalibrated.
Temperature	°C	Transpiring crop is cooler than the air; stress and soil, hotter; water and leaks, cold; metal looks cold because it emits little.
Material classification	NDVI, NDWI and brightness thresholds	Vigorous vegetation, weak vegetation, soil, track, water and metal roof. The thresholds can be adjusted.
Photos per point	—	Quality control: 3D needs 5 or more.

- **Probe**: hover to see, at the bottom, the coordinates, the value of each band and index, and how many photos see that point.
- **Histogram and threshold**: pick a value range, highlight it and turn it into a zone with **Zone from threshold** (it drops patches smaller than 4 m²).
- **Zones**: choose the category (pest, stress, leak, soil, track, water, metal or other) and draw with **Draw zone** (click each vertex, double click to close) or create it from the threshold. With the classification, **Create water, metal, track and soil zones** makes them in one go. Each zone shows its area and mean value.
- **Photo footprints**: draws each photo's footprint; discarded ones in red.
- Mouse wheel to zoom, drag to pan, Esc to close.



Thermal at midday: the water-stressed area is hotter than the transpiring crop.

Exercises

Each exercise prepares a new parcel layout, sets it as the survey area and chooses the camera and GSD. You decide the time, plan, fly, process, mark the zones and press **Evaluate** in the lab. The score compares your zones with the parcel's truth (IoU: shared area divided by the union) and rates the capture.

Exercise	Camera and GSD	What is scored
Photogrammetric coverage	RGB, 2.5 cm/px	Coverage, 75/65 % overlaps, unusable photos, GSD and conditions.
Detect pest foci	Multispectral, 6 cm/px	"Pest focus" zones, calibration (panel before and after, irradiance sensor), coverage, GSD and conditions.
Water stress map	Thermal, 11 cm/px	"Water stress" zone, hot time of day, wind, coverage and GSD.
Find the irrigation leak	Thermal (or multispectral), 10.5 cm/px	Your "Irrigation leak" zone within 10 m of the leak, its IoU, conditions, coverage and GSD.
Classify materials	Multispectral, 6 cm/px	Water, metal roof, track and bare soil zones, coverage, calibration, GSD and conditions.

The best score of each exercise is kept in the panel.

Reports and export

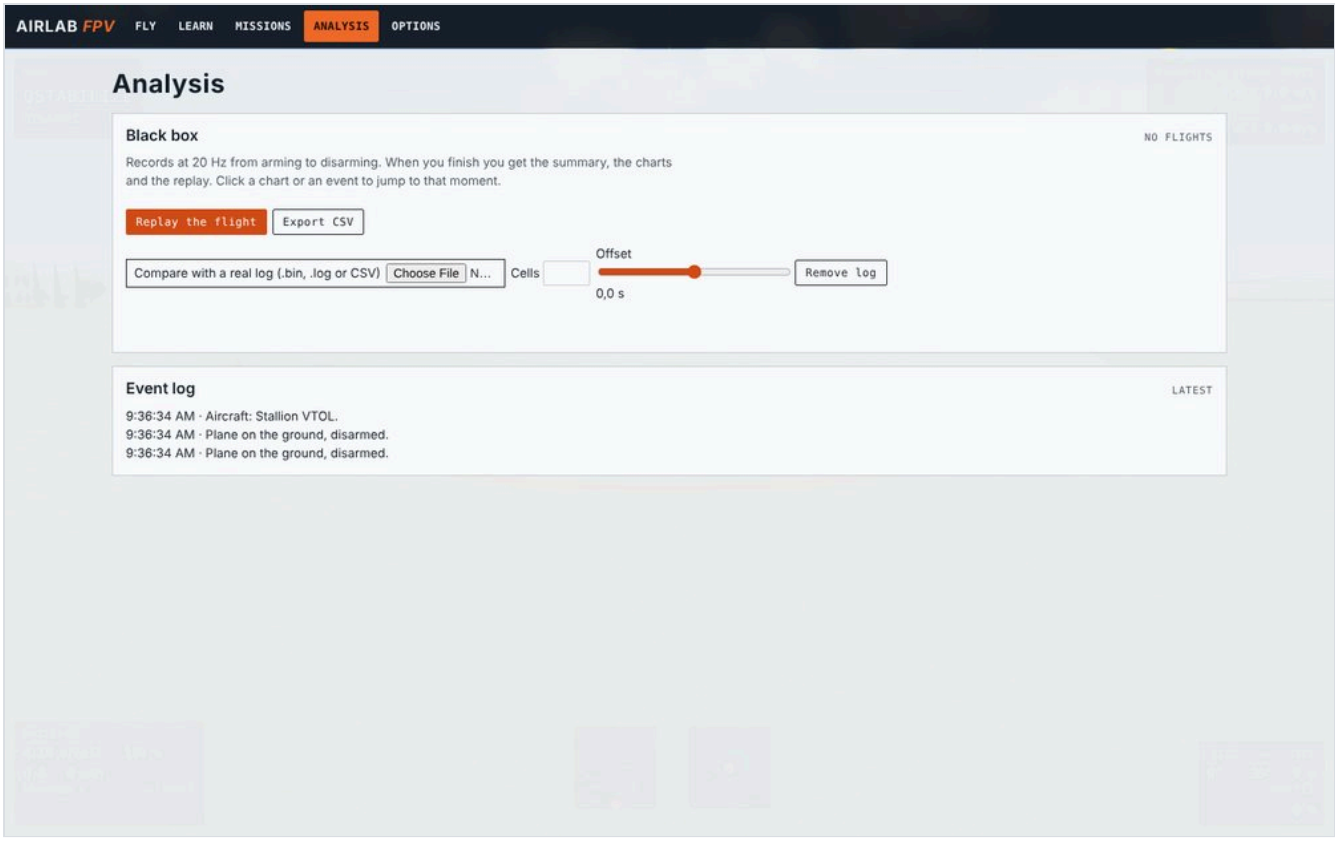
Button	What it saves
PDF report	Mission summary (aircraft, camera, time, sun, clouds, wind, photos, GSD, coverage, overlaps, blur, radiometry, coordinates), quality control and discarded photos, each layer with its scale and explanation, your zones with area and coordinates, the exercise score, the AI study if there is one, and the method.
Interactive web page	A single .html file that opens in any browser offline: layers with legend and opacity, photo footprints with their data, zones, a probe with coordinates and index value, summary, score and a button to download the GeoJSON.
GeoJSON	Zones, photo footprints and centres and the orthomosaic outline, in WGS84 coordinates, to open in a GIS.
Survey video	An animation of about 17 s: the photos cover the orthomosaic in the order they were taken, then the index and finally the zones.

The mission with triggering is exported, as always, from the planner as .waypoints.

AI study

In the web version with your account, **AI study** sends Captain Pitot the orthomosaic, the main index (or thermal), NDRE and the classification at reduced size, together with the statistics, quality, conditions and your zones. He returns what he sees and what it may mean in the field, whether your zones fit, the capture quality and what to change next time. The study appears in the lab, he says it out loud, and it goes into the PDF report and the web page. It counts towards the same daily limit as flight analysis.

19 Analysis: black box and real logs

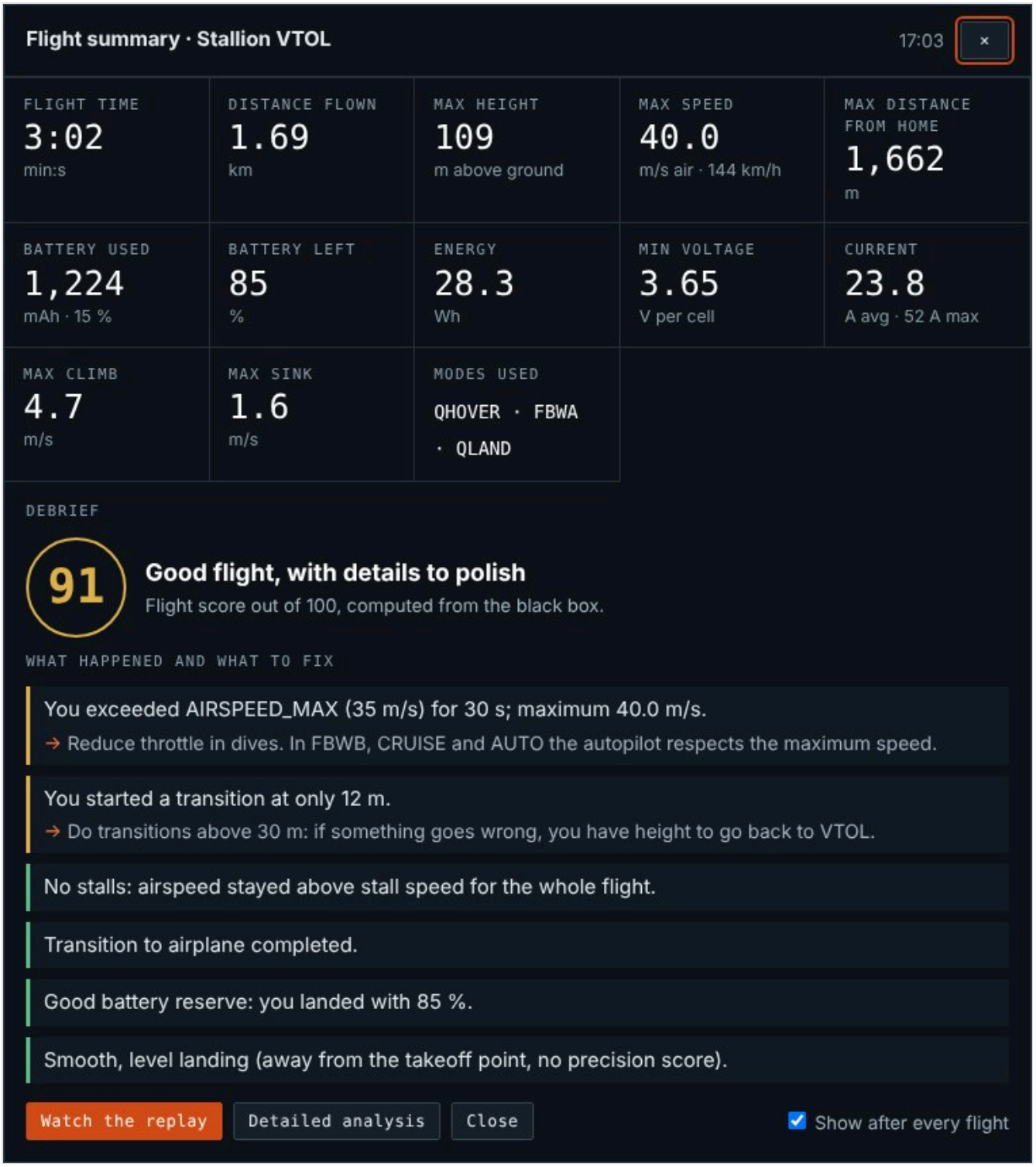


Analysis: black box and event log.

The black box records at 20 Hz from the moment you arm until you disarm. When you finish you will see:

- **Summary**: duration, maximum altitude and speed, distance, consumption, transitions and landing.
- **Graphs** of altitude, speeds, attitude, battery, current and throttle. Click on a graph or an event to jump to that moment.
- **Replay flight**: plays the flight back in 3D with its trail, at several speeds.
- **Export CSV** with all the data.

Summary and debrief after the flight



Flight summary and debrief, on disarming.

When you disarm, or after an impact, a flight summary like the one in DJI goggles appears: flight time, distance flown, maximum height above the ground, maximum speed, maximum distance from home, battery used and left, energy, minimum voltage per cell, average and maximum current, maximum climb and sink, and the modes you used.

Below it comes the **debrief**: a flight score out of 100 and a list of what happened, each point with how to fix it on the next flight. It is computed from the black box and checks:

- **The crash and its cause**: hitting an obstacle, a stall close to the ground, vortex ring state, bank at touchdown, a flat battery, a dive or a descent that was too fast, throttle cut in the air (drones).
- **Stalls** in airplane flight: how many, at what speed and at what height.
- **Limits**: speed above `AIRSPEED_MAX`, too much bank in MANUAL or ACRO, airplane flight below 10 m and height above the 120 m of the EU open category.
- **VTOL**: vortex ring state when descending, transitions started below 30 m and height lost in them.
- **Battery**: reserve at landing (20 % is the recommendation) and heavy voltage sag.
- **Failsafes** that triggered, and the **landing**: vertical and horizontal speed at touchdown, distance to the takeoff point and tilt; for the fixed wing, how hard the belly landing was.

Watch the replay opens the 3D replay and **Detailed analysis** takes you to the black box with the charts. You can turn off the automatic summary with the checkbox at the bottom and open it again with **Summary and debrief** in the black box.

Replay with instructor



Replay with instructor: your flight in grey, the correction in blue; Captain Pitot explains it.

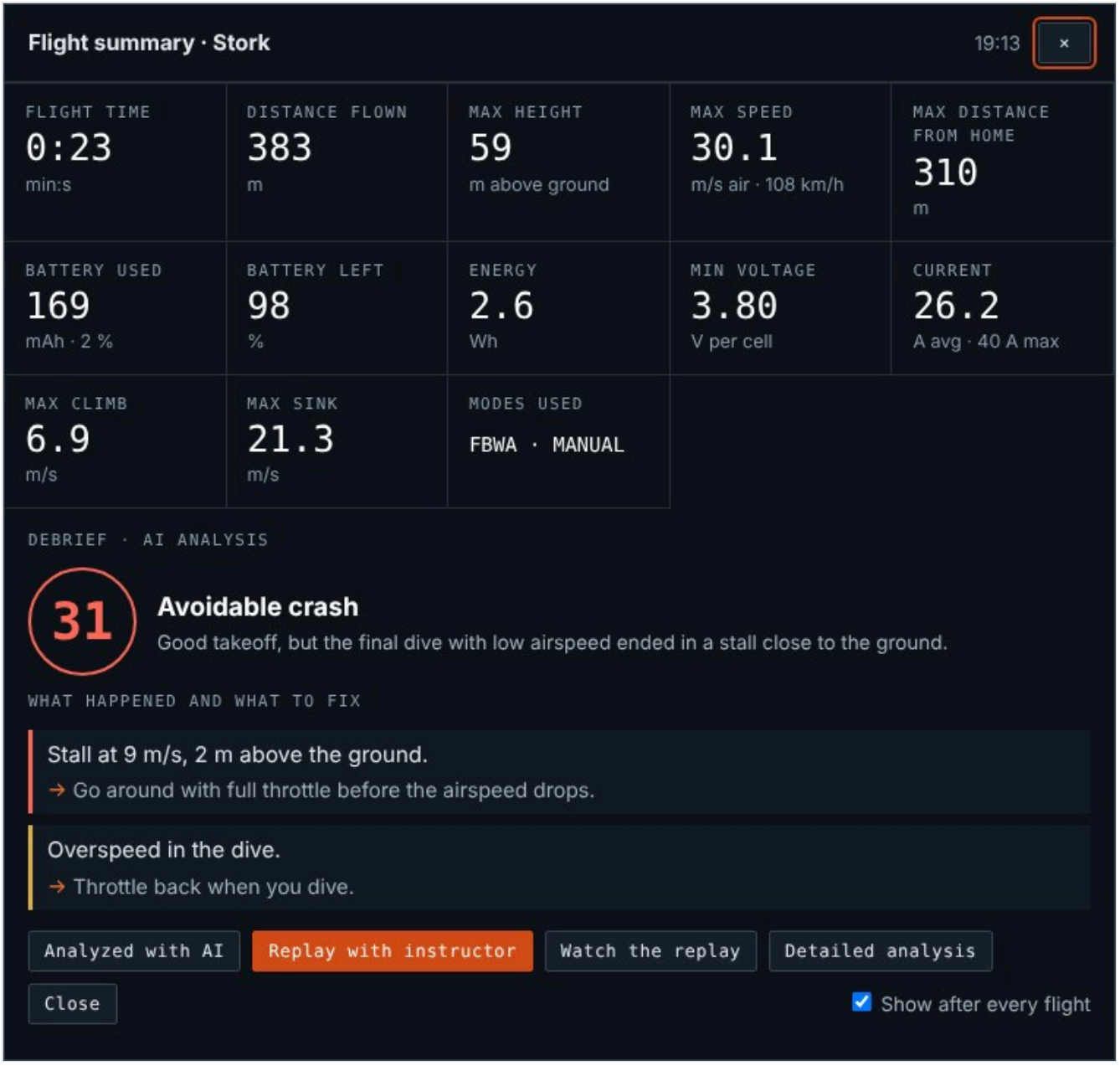
Replay with instructor (in the summary or in the black box) plays back your flight with an instructor who comments on it out loud and with subtitles. Before each mistake you get a warning, at the moment of the mistake the replay stops and the instructor explains what happened, and then shows you the correction.

- **The blue aircraft** is the correction: the simulator flies that part again with its physics, from a few seconds before the mistake, with the manoeuvre that should have been made (throttle back in a dive, level the wings, go around, stretch the transition, descend more slowly in QLAND...). Its trail is blue; yours is green.

- At the top left you have **your sticks** (orange) and those of **the correction** (blue), so you can see what changes in your hands.
- **Continue** goes on after a pause, **Next point** jumps to the next comment and **Exit** takes you back to flying. When there is nothing to comment the replay runs faster, and in the parts with a correction it runs at half speed.

It reviews crashes (obstacle, stall, throttle cut, vortex ring, hard landing), stalls, too much speed or bank, vortex ring state, low transitions or transitions that lose height, and the quality of the landing. If you crash, the correction keeps flying after the impact so you can see how it would have ended.

AI analysis



Debrief written by the AI.

In the web version, with your licence, the summary has an **Analyze with AI** button. It sends Claude (Anthropic) a summary of the black box (flight data, detected events, the aircraft's ArduPilot parameters

and the height, speed, attitude and stick series) and gets back a written debrief, a score, the instructor's script and the corrective manoeuvres. **Replay with instructor** then uses that script and those corrections.

- It usually takes between half a minute and two minutes; meanwhile the summary shows how long it has been running. If the AI doesn't answer, takes longer than 5 minutes or you have reached the daily analysis limit, the rule-based debrief stays and the replay with instructor works the same.
- The AI only uses the data from your flight; the corrections are flown by the simulator with its physics, just like the rule-based ones.
- The offline version has no AI: it always uses the rule-based analysis.

Captain Pitot, your instructor

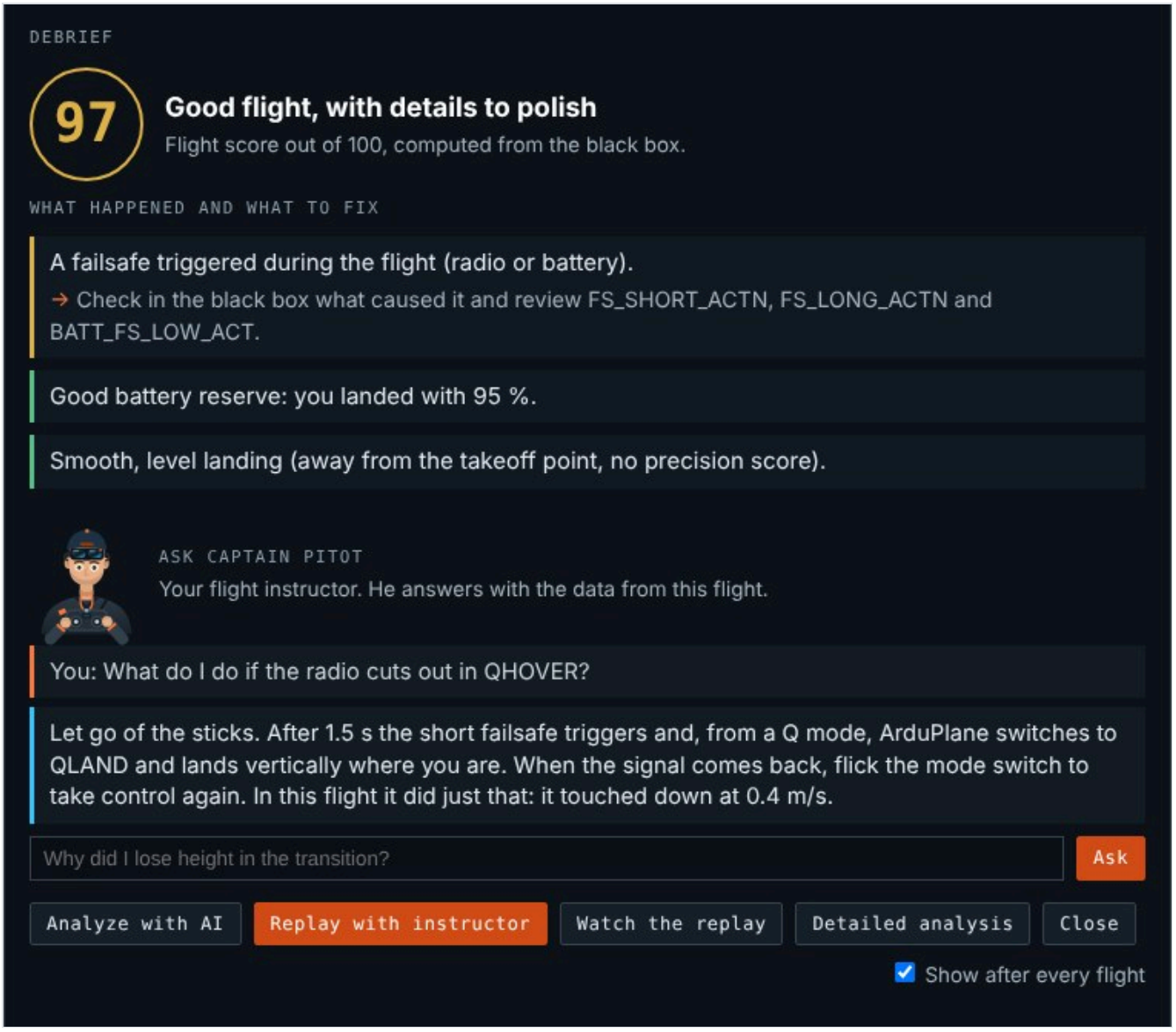


Captain Pitot: at rest, showing a manoeuvre with his hand, pointing at the correction and thinking.

The simulator's instructor is **Captain Pitot**, an FPV pilot with his goggles pushed up on his forehead and the radio hanging from his neck. He appears in the replay with instructor, in the questions in the summary, in the in-flight callouts and in the Learn panel.

- **He moves while he talks**: his mouth follows the voice and he explains with his hands. His hand acts as the airplane: it banks, raises or lowers the nose and acts out the landing.
- When he shows a correction, his hand and the sticks on his radio copy the correction's manoeuvre: aileron, elevator and throttle.
- While the AI prepares the answer, he **thinks**.
- **He has his own voice**, different from the simulator's callouts. In automatic it picks a male voice from the browser. You change it in **Options › Sound › Captain Pitot's voice**, where you can also turn it off and test it. If the browser only has one voice, he uses it with a lower pitch.

Ask Captain Pitot

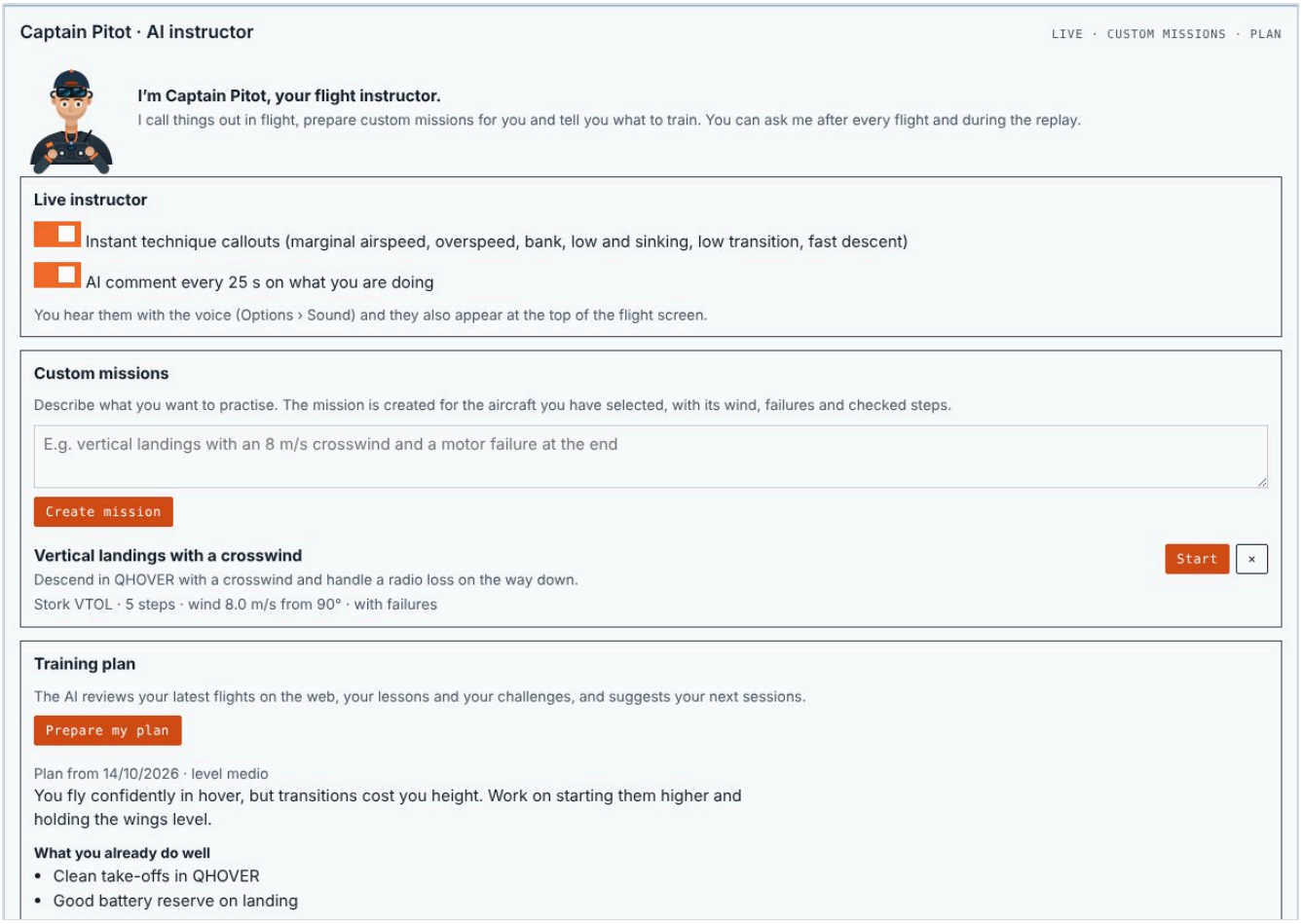


Ask Captain Pitot, in the flight summary.

In the web version, the flight summary has an **Ask Captain Pitot** box. Type your question (“why did I lose height in the transition?”, “what do I do if the radio cuts out in QHOVER?”) and the AI answers with the data from that flight. The conversation is kept while you stay on the same flight.

In the **Replay with instructor** there is an **Ask** button: it pauses the replay, sends your question with the moment you are at and reads the answer out loud. **Continue** carries on with the replay.

Captain Pitot · AI instructor (Learn)



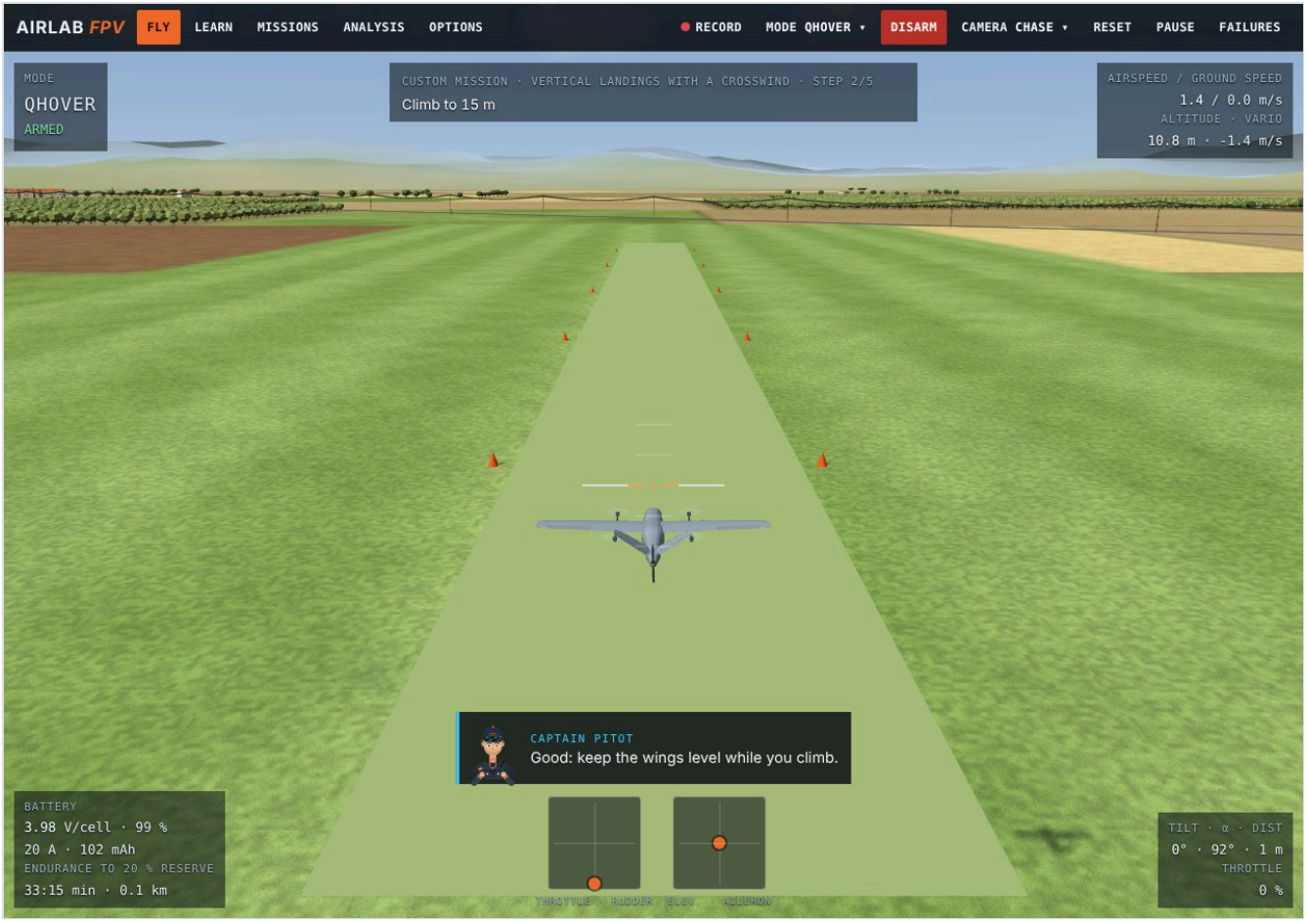
AI instructor panel, in Learn.

In Learn, below the lessons, the **Captain Pitot · AI instructor** panel brings together three tools.

Live instructor. It has two switches:

- **Instant technique callouts**: marginal airspeed or above `AIRSPEED_MAX`, bank above `LIM_ROLL`, low and sinking, a transition started below 30 m and a fast vertical descent, each with what to do. They are rule-based: they come instantly and also work in the offline version.
- **AI comment every 25 s**: the AI looks at the last few seconds of telemetry and the lesson, challenge or mission you are doing, and gives you one technique tip. If there is nothing useful to say, it stays quiet.

Captain Pitot says both with his own voice, and they also appear on the flight screen above the sticks, next to his picture.



Custom mission in progress, with the instructor’s comment.

Custom missions. Describe what you want to practise, for example “vertical landings with an 8 m/s crosswind and a motor failure at the end”, and press **Create mission**. The AI prepares a mission for the aircraft you have selected:

- the starting point, on the ground or in the air;
- the wind, turbulence and gusts;
- the steps, each with its conditions (mode, height, distance, speed, bank, turn, heading or touchdown) and how long to hold them;
- the failures that trigger at each step.

The mission plays like a lesson, with the instructions at the top of the screen. When you finish or leave, the wind and failures go back to how they were. Missions stay saved in the list, with your best time. The AI can only use the conditions, modes and failures the simulator understands: it does not write code.

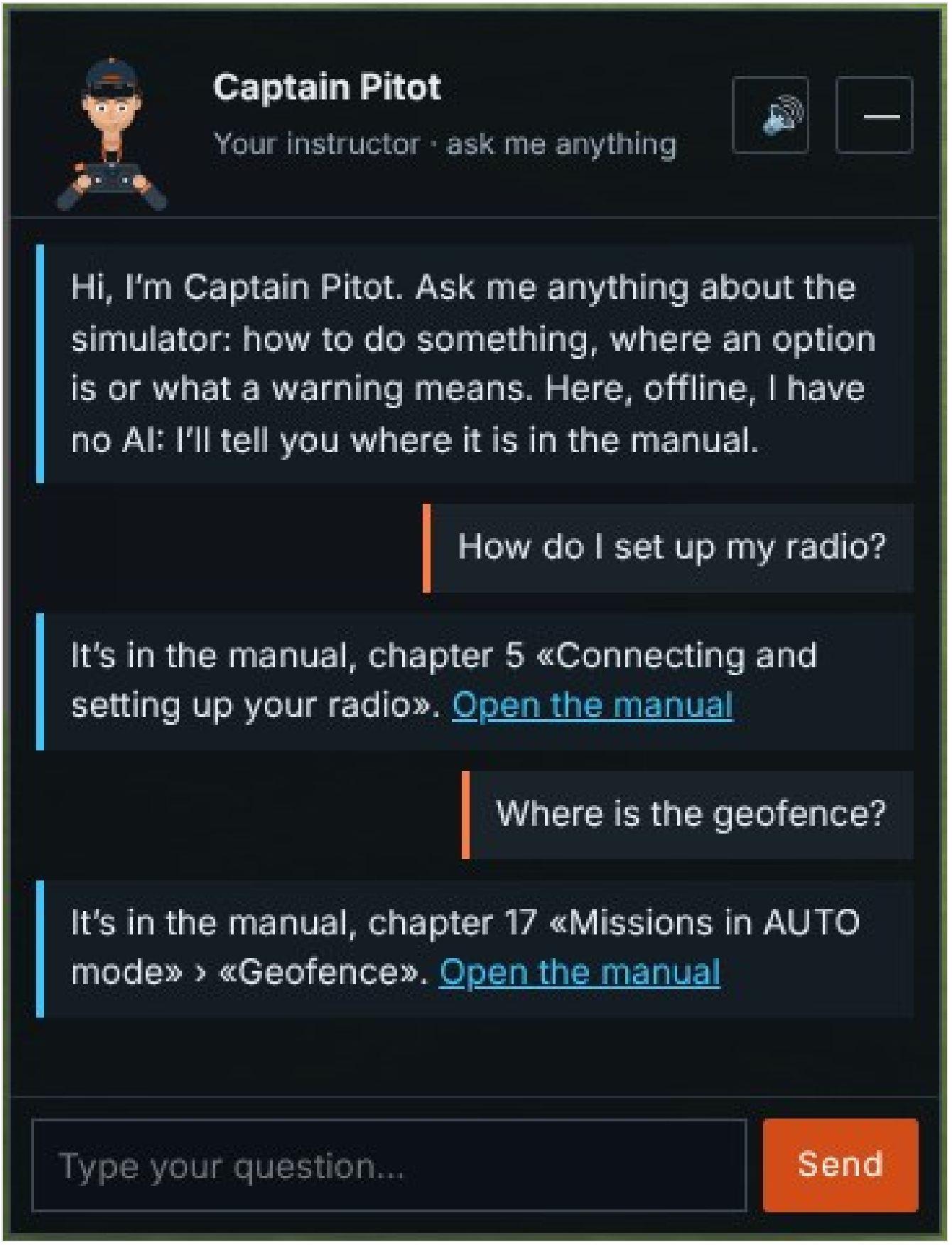
Training plan. Every flight you make on the web saves a summary (aircraft, duration, score, crash and debrief points). **Prepare my plan** sends that history, your lessons and your challenges, and the AI gives you back:

- what you already do well and what to work on most;
- 3 to 5 sessions with a measurable goal.

Each session has its own button: it starts the lesson or the challenge, or creates the custom mission it suggests.

Each AI feature has its own daily limit per account. If there is no connection or the limit is reached, the simulator tells you and keeps working without AI.

Captain Pitot's chat



Captain Pitot's chat, at the bottom right.

Captain Pitot is always at hand in a chat at the bottom right, on every screen. Ask him how to do something, where an option is or what a warning means (“how do I set up my radio?”, “where is the geofence?”).

- **On the web** the AI answers using this manual and what you are doing: the screen, the aircraft, the mode and your last flight. If something isn’t in the manual, he tells you. He can read the answers out loud (speaker button).
- **Without AI** (in the offline version, or with no connection or once the daily limit is reached) he tells you which chapter and section of the manual covers your question, with a link to open it.
- Minimise it with — and open it again with its tab. While you fly armed or watch a replay it moves out of the way so it doesn’t cover the OSD, and comes back when you disarm. It remembers the conversation.

Compare with a real log

Load a log from your real aircraft (ArduPilot .bin, text .log or a CSV from the simulator). Its data is overlaid on the graphs as a dashed line, with an adjustable time offset to line them up and the cell count for the voltage.

Real flight investigation

As an airline does with the black box after an incident: in **Analysis**, above the black box, **Investigate a log (.bin or .log)** loads your aircraft or drone's ArduPilot log (ArduPlane, QuadPlane or ArduCopter) and replays it in 3D on an investigation screen. The log is read on your computer and not uploaded.

- **Top left, third person**: chase, orbit (drag with the mouse and use the wheel to zoom) or from the pilot on the ground.
- **Top right, first person with OSD**: horizon, mode, armed state, total and per-cell voltage, current and consumption, air and ground speed, height and vario, satellites and HDOP, distance to home, vibration, heading and time. When something serious happens, a red warning appears in the middle.
- **Below, the flight parameters**, all on the same time axis: height, speeds, vario, roll and pitch (desired by the autopilot and actual), throttle and motors, voltage per cell, current, vibration (with the 30 and 60 m/s² lines), satellites and the EKF innovation ratio. On top are the modes and the findings as coloured triangles. Click or drag to go to a moment; `Ctrl` + wheel to zoom and double-click to see it all.
- **On the right, Captain Pitot**: he introduces the flight, stops at each finding and explains it out loud. Below are the findings (tap one to go to that moment), the log timeline (arming, modes with their reason, ERR errors, messages and events) and, on the web with your account, **Investigate with AI**.

The rule-based findings look for: the log ending with the aircraft armed in the air, failsafes and mode changes caused by a failure (radio, battery, GPS, EKF, fence...), vibration above 30 and 60 m/s² and accelerometer clipping, the EKF rejecting measurements, weak GPS, low battery under load or with sudden sags, loss of control (actual attitude more than 20° from what the autopilot asks for), motors at their limit, airspeed below the minimum in fixed-wing flight, fast descents near the ground, hard or banked touchdowns and flights above 120 m.

Investigate with AI (counts as one of the day's analyses) only sends a summary: the timeline, the findings and a compact series of the data. Captain Pitot returns the probable cause with its confidence, the factors, the recommendations, what to practise in the simulator and a second-by-second narration that replaces the rule-based one in the replay.

If the log has GPS, the flight is shown **at its real place**: the real terrain (relief and orthophoto) around the log's home is loaded. The terrain button goes back to the simulated scenery. The height is the log's height above the take-off point, so if you land somewhere higher or lower than home, the aircraft can be slightly above or below the ground. It is shown with the aircraft chosen in the simulator: if the log is from a multirotor, choose a drone to see it with its shape. With several flights in the same log, choose which one at the top. `Space` plays or pauses, the arrows move 5 s and `Esc` closes. On closing, everything goes back to how it was (home, terrain, camera and the black box of your last simulated flight).

Fly it yourself: recreate the incident and save it

Each finding in the list has ✈ **Fly it**: the simulator puts you a few seconds before (10, or 15 for a touchdown) at the same place, height, heading, speed and mode as the real flight, with a 3-second countdown, and you take the controls with your radio. At the same moment as in the real flight the equivalent failure kicks in: **radio** (it comes back after 12 s so you can take control again), **battery** or **GPS**; for the others (vibration, stall, hard landing...) the moment is recreated so you can fly it better. It ends if you crash, if you land or 45 s later with no problems, and Captain Pitot compares it with what really happened. ⟲ **Again** repeats the attempt and **Back to the investigation** takes you back to where you were. ✈ **Fly from here**, in the playback bar, gives you the controls at the moment you are watching, with no

failure. Each attempt goes into the logbook as a simulator flight and into the PDF report. It is flown with the aircraft chosen in the simulator, and with its closest mode if it does not have the log's.

Investigation report in PDF

📄 **PDF report**, at the top, creates a report to keep or send (to a client, an insurer or for the aircraft's history): flight data, Captain Pitot's conclusion if you investigated it with AI, the findings explained, the track seen from above with the findings numbered, captures of the key moments in third person and first person with OSD, the parameter charts, your re-flights and the log timeline.

Fleet maintenance (predictive) and pre-flight checklist

Like an airline's component control, for your real aircraft. In **Analysis › Fleet maintenance** each aircraft has its history: every log you investigate is saved in it (the aircraft is recognised by the identifiers of its sensors, which come in the log parameters; the first time you choose which aircraft it belongs to or give it a name). **Add logs** adds several at once. In the investigation you can name the **battery** used in that flight to track its wear.

- **Metrics of each flight**: mean vibration and clipping, hover throttle and current, cruise current, each motor's output in hover (imbalance), battery internal resistance (from the voltage drop on current steps), minimum voltage, consumption, EKF, GPS and the findings.
- **Trends and alerts**: from the fourth flight, the last one is compared with the median of the previous ones: rising vibration (damaged prop or bearing), rising hover throttle or current (more weight or less thrust), one motor working harder than the others, ageing batteries, voltage at its limit and recurring findings. The charts show each value flight by flight with its usual line.
- **Maintenance tasks** by hours, flights or days (props, screws, bearings, tilts, control surfaces, airframe, connectors, Pitot, calibrations, parameter backup and failsafe test), with a usage bar and **Done**. You can add the hours and flights from before AIRLAB.
- **Batteries**: flights, internal resistance from the start to now, minimum voltage and typical consumption of each one.

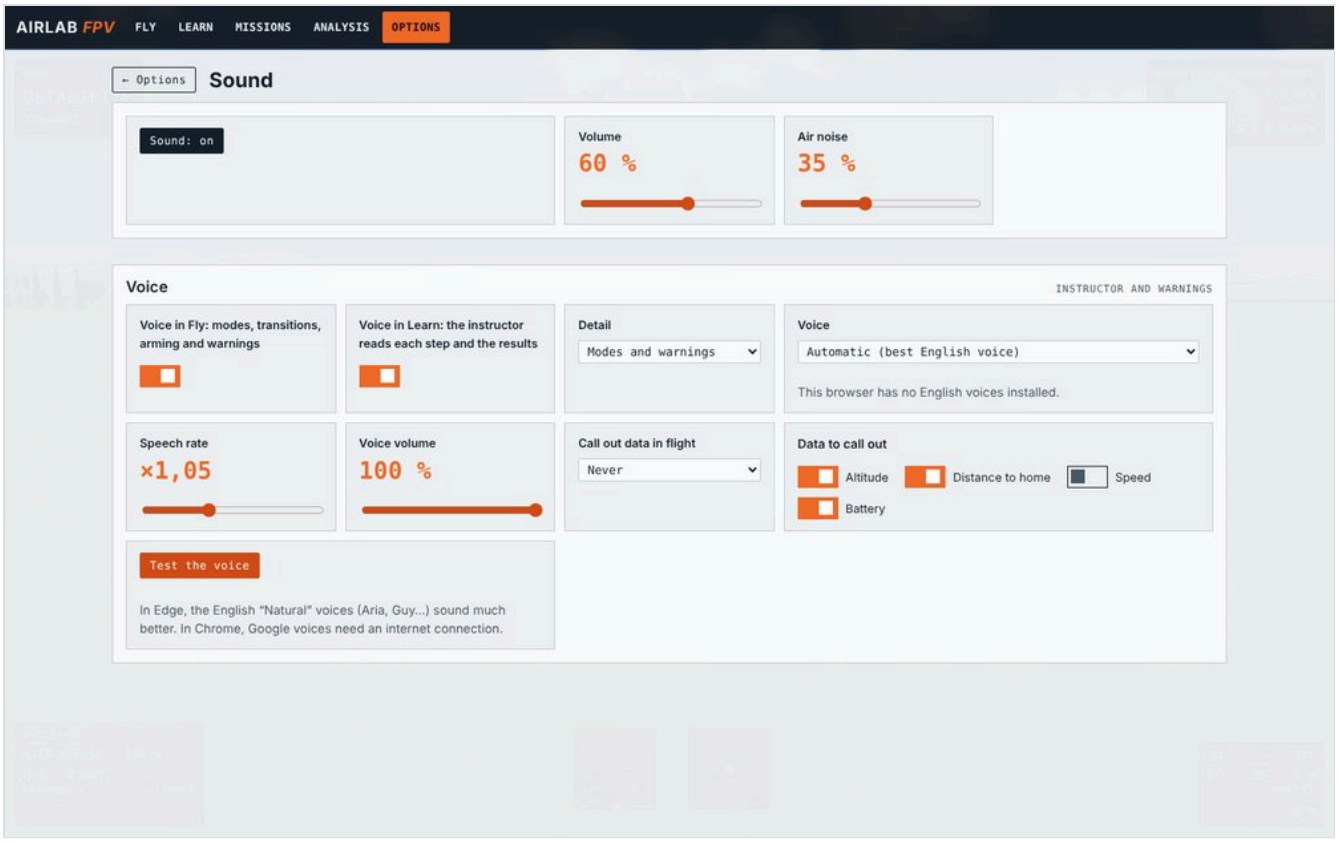
📋 **Pre-flight checklist**: generated for each aircraft from the last flight (for example, after vibration, check the props and motors; after weak GPS, wait for 10 satellites), the alerts, pending maintenance and the before-every-flight items for its type. Tick it on screen or print it as a PDF. On the web with your account, **On the phone** shows a QR that opens the list on the phone to tick it at the field: that QR is always the same for the aircraft and always shows the latest list, so you can print it (it is in the PDF too) and stick it on the aircraft. The history is saved in the browser and, with an account, on the server so you have it on any computer; it can also be exported and imported.

Aircraft logs over WiFi

No cable and no SD card swapping. The **CoreWing F405 Wing V2** (the "APP" one) already has WiFi: its built-in module creates the "CoreWing WING-WiFi" network (password `88888888`, IP `192.168.1.1`, TCP 4278) with SERIAL1's MAVLink, and AIRLAB Connector finds it by itself. On other flight controllers use a small **ESP32** (for example an ESP32-C3 SuperMini) running the free **DroneBridge** firmware, wired to a free serial port of the flight controller (on the CoreWing F405 Wing, UART4: `SERIAL4_PROTOCOL = 2`, `SERIAL4_BAUD = 921`), creates a WiFi network with ArduPilot's MAVLink. **AIRLAB Connector** finds it by itself (at `dronebridge.local` or `192.168.2.1`, TCP 5760) and, with the aircraft disarmed, downloads the new `.bin` logs to **Documents › AIRLAB Logs**; it never downloads the same log twice.

- In **Analysis › Fleet maintenance**, the **Aircraft logs over WiFi** box shows whether the ESP32 is connected, the log being downloaded with its speed and the ones that have arrived. With **Import automatically**, every new log goes into its aircraft's history without a click.
- **At home**: set DroneBridge as a client of your WiFi; power the aircraft near the Mac and the logs arrive by themselves. **At the field**: access point mode; join the laptop to the ESP32's network.
- Only log requests are sent (the same as QGroundControl's Analyze › Log Download), with system ID 253 so the GCS failsafe is not affected. The ESP32 transmits on 2.4 GHz like ExpressLRS: mount it away from the receiver and do a range check with it powered.

20 Sound and voice



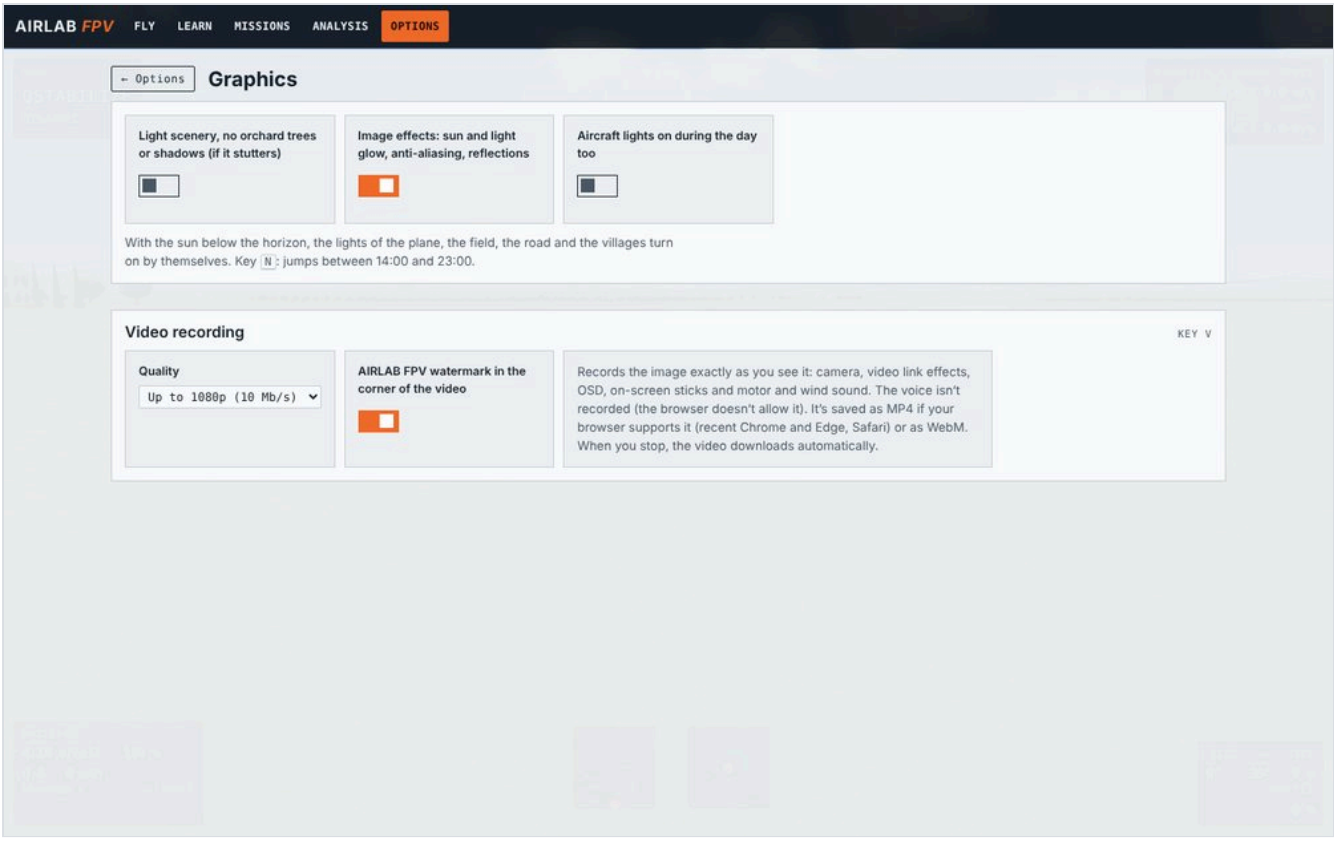
Options › Sound.

Sound: motors, wind and alerts, with master volume and air noise volume. The browser requires a click or a key press before it will play sound. Press `M` to turn it on or off.

Voice

- **In Fly**: announces modes, arming, transition phases, takeoff and landing, and alerts (battery, radio and video signal, failsafes, GPS Rescue, stall, vortex ring state, return now, crash).
- **In Learn**: the instructor reads out each step, corrects you, encourages you and announces results.
- **Detail**: alerts only, modes and alerts, or everything with explanations (the first time you enter a mode, it explains what it does).
- **Call out data** every 15 s, 30 s or 1 minute: altitude, distance to home, speed and battery.
- **Voice, rate and volume**. In Edge, the "Natural" voices sound much better (in English, for example Aria or Guy).

21 Graphics and performance



Options › Graphics.

- **Light scenery**: removes orchard trees and shadows. Turn it on if the flight stutters.
- **Image effects**: glow from the sun and lights, anti-aliased edges and reflections. They use more resources.
- **Daytime lights**: turns the aircraft lights on during the day as well.

The trees are olives, holm oaks, stone pines and cypresses with irregular crowns, silvery leaves (the olives) and sway in the wind. The sceneries with relief have up to about 9,000 trees and 7,000 bushes: if the mountain or the dam runs slowly on your computer, turn on light scenery.

> **If it runs slowly**
>
> Close other tabs, turn on light scenery, turn off effects, and on laptops plug in the charger so the graphics card isn't throttled.

22 Using it on a phone



Flying on a phone in landscape with touch sticks.

- **Hold your phone in landscape**. In portrait you will see “Rotate your phone to fly”. On Android, entering Fly switches to full screen and locks landscape; on iPhone, just rotate it.
- **Touch sticks**: the throttle stick does not return to center and moves by dragging from wherever you place your finger; the other one returns to center when released. They follow Mode 1 or Mode 2.
- **Buttons** at the top center: arm, mode, camera and reset.
- **Radio or gamepad**: on Android you can connect your radio via USB-C with an OTG cable; on Android and iPhone, a Bluetooth gamepad. The touch sticks hide automatically.
- Light scenery turns on automatically the first time so it runs smoothly.

23 Keyboard shortcuts

Key	Action	Key	Action
`W` / `S`	Throttle up and down	`Space`	Arm or disarm
`↑` / `↓`	Elevator	`1`–`0`	Modes (the first ten)
`←` / `→`	Aileron	`C`	Switch camera
`A` / `D`	Rudder	`R`	Reset on the ground
`N`	Day or night	`M`	Sound
`Esc`	Start menu and pause	`Enter`	Free flight (on the start screen)
`V`	Start or stop video recording	`G`	Phone viewer (viewer mode)
`L`	Sailplane: release the tow	`K`	Sailplane: weather map

In the OSD editor: arrow keys move the selected element and `Delete` removes it.

24 Troubleshooting

Problem	Solution
The radio doesn't show up	Use Chrome or Edge on a computer, select “USB Joystick (HID)” on the radio and move a stick with the simulator open. Try another USB cable: many are charge-only.
A channel doesn't move (for example, arming)	In EdgeTX › USB Joystick, check that the channel is enabled as Axis or Btn. The monitor in Options › Radio shows how many axes and buttons the radio is sending.
A stick is reversed	Turn on “Invert” for that channel. Elevator: stick forward = nose down.
The mode doesn't change with the switch	Check the mode switch ranges and which mode is assigned to each position.
It won't arm	Look at the “Before flying” window: whatever is in red is what's failing. Most common causes: throttle not all the way down, waiting for the GPS to see 6 satellites, or needing a reset after a crash.
The recorded video has no voice	This is normal: the browser doesn't allow recording synthesized speech. Motor and wind sounds are recorded.
It stutters	Light scenery, effects off, other tabs closed.
No sound	Click on the view or press a key: the browser requires it. Check Options › Sound.
The voice sounds robotic	Use Edge and choose a “Natural” voice.
Real flight: UDP port 14550 is in use	QGroundControl or Mission Planner are open. Quit them completely (Cmd+Q) and reopen AIRLAB Connector.
Connect says the connector is not running	Double-click “AIRLAB Conector” on the Desktop. If it isn’t there, install it from that same page (chapter 27).
Real ArduPilot: “ArduPilot is not built”	It has to be built once: `bash ~/AIRLAB/compilar-ardupilot.sh` in Terminal (chapter 27).
Real ArduPilot: the arm switch is up and it does not arm	The reason shows in red below. While ArduPilot is not ready (GPS and EKF, about 40 s) it retries by itself every 4 s. If it says “Throttle...”, lower the throttle fully; if it is another PreArm from your `.param`, fix it (chapter 27.6).
Real ArduPilot: “PreArm: Gyro 0 rate ... < loop rate*1.8”	The physics must run at least 1.8 times ArduPilot’s loop: `SCHED_LOOP_RATE 200` and `SIM_RATE_HZ 400`. Loading your `.param` from the panel sets them automatically.
Real ArduPilot: “Duplicate Aux Switch Options”	Two channels with the same function (for example arming on 5 and on 7). Loading your `.param` from the panel with connector 1.2 fixes it automatically.
The connector still shows the old version after installing	Chrome saved the new installer as `airlab-instalar (1).sh` and the old one ran. Delete the `airlab-instalar*.sh` files in Downloads, download it again and run it.
Real ArduPilot: won’t arm after a crash	Press **Reset**: the connector restarts ArduPilot and the aircraft returns to the pad. Wait for GPS again (about 40 s).

Real flight: slow or jerky	Turn off QGroundControl's UDP auto-connect (it changes the MAVLink rates), check `MAVn_*` on the receiver port or raise the packet rate (chapter 26).
Real flight: connected, but the aircraft doesn't move	Without a GPS fix the aircraft stays on the pad; attitude, sticks and mode do move.
A scenery with relief runs slowly	Turn on light scenery in Options › Graphics or go back to the countryside.
"The simulator is open on another device"	It can only be open in one place at a time. Close it on the other device and open it again.

25 Glossary

ArduPilot / ArduPlane	Open-source autopilot; ArduPlane is its version for airplanes, which includes VTOLs (QuadPlane).
Betaflight	Flight controller firmware widely used on freestyle and racing FPV drones.
Tilt-rotor	A VTOL whose motors tilt from vertical to horizontal to go from hover to airplane flight.
Transition	The change from vertical flight to airplane flight, or the other way around.
Failsafe	An automatic action in response to a failure, such as loss of radio or low battery.
Rates	The rotation rate produced by each stick position.
Expo	A curve that softens the stick response around center.
OSD	On-Screen Display: data overlaid on the camera image.
LQ / RSSI	Radio link quality in % and received signal strength in dBm.
VTX	The aircraft's video transmitter.
ELRS	ExpressLRS, an open-source radio system widely used in FPV.
MAVLink	The message protocol ArduPilot and PX4 use between autopilot and ground station.
Backpack	A small WiFi processor inside the radio's ELRS module; in MAVLink mode it forwards telemetry over WiFi.
Link Mode	ELRS link mode: Normal (CRSF) or MAVLink.
Packet rate	Packets per second of the ELRS link (for example 250 Hz). Faster means less range and more bandwidth.
UDP	A connectionless way of sending data over a network; the Backpack sends MAVLink over UDP to port 14550.
Ridge lift	Air that rises when wind hits a slope; it lets you glide without power along it (slope soaring).
Pitot	A sensor that measures airspeed.
QuadPlane	ArduPlane configuration for airplanes with vertical-lift motors.
Elevon	Control surface on flying wings that works as aileron and elevator at the same time.
4+1	A QuadPlane with four fixed lift motors and one forward-thrust motor.
Geofence	A radius, altitude or polygon limit that the autopilot won't let you cross without reacting.
Rally point	An alternative location that RTL returns to if it is closer than home.
Photogrammetry	Measurements and maps made from many overlapping photos taken on parallel passes.
SITL	Software In The Loop: the real ArduPilot code running on the computer; in AIRLAB FPV, with the simulator's physics (chapter 27).

AIRLAB Connector	Program on the computer that links the simulator with the real aircraft's telemetry and with ArduPilot SITL.
GSD	Ground sampling distance: the size on the ground of each photo pixel (cm/px).
Overlap	The share of two consecutive photos (front) or two passes (side) that is the same.
Orthomosaic	An image of the ground built from many photos and corrected so everything is seen vertically and to scale.
NDVI / NDRE	Vegetation indices from near infrared and red or red edge.
Reflectance	The fraction of light a surface reflects in each band; obtained by calibrating with a panel of known reflectance.

AIRLAB FPV is not affiliated with ArduPilot, Betaflight, DJI, ExpressLRS, EdgeTX, Flightory, RadioMaster or QGroundControl; their names are mentioned only to describe compatibility and behavior. Orthophotos: PNOA courtesy of © Instituto Geográfico Nacional (CC BY 4.0). Elevation: Mapzen Terrain Tiles on AWS Open Data.

26 Step-by-step guide: ELRS MAVLink telemetry over WiFi

This chapter walks through how a real aircraft was connected to the simulator using ExpressLRS MAVLink telemetry and the WiFi of the radio's Backpack, including the problems that came up and how they were solved. It is the setup used by the **Real flight with instructor** panel (chapter 16). The values are those of this build: with other gear the steps are the same, but check the names of your receiver and radio.

Part	In this build
Radio	RadioMaster TX15 with internal 2.4 GHz ELRS module and TX Backpack (EdgeTX)
Receiver	RadioMaster RP4TD (2.4 GHz ELRS, true diversity, ESP chip with WiFi)
Flight controller	CoreWing F405 Wing running ArduPlane 4.7 (Stallion VTOL); the receiver is wired to the `SERIAL6` UART
Computer	Mac laptop with QGroundControl, Python 3 and Chrome
Final versions	ELRS 3.5.6 on radio and receiver · Backpack 1.5.9

26.1 How the data travels

1. **Flight controller → receiver.** ArduPilot talks MAVLink on `SERIAL6` at 460,800 baud to the RP4TD.
2. **Receiver → radio.** The 2.4 GHz ELRS link carries the sticks to the aircraft and the MAVLink telemetry back. In MAVLink mode the telemetry ratio is fixed at 1:2.
3. **Internal module → Backpack.** The TX15 module hands the MAVLink to the Backpack, which opens its own WiFi network (access point at `10.0.0.1`) and sends the packets over UDP to port **14550**.
4. **Backpack → Mac.** The laptop, joined to that network, receives the UDP. A browser cannot open UDP, so **AIRLAB Connector** picks it up and serves it at `http://127.0.0.1:14580/stream`.
5. **Mac → simulator.** The **Real flight with instructor › WiFi bridge** panel reads that stream and feeds it into the simulator state: 3D, OSD, callouts, black box and debrief.

> **The sticks travel over MAVLink too**
>
> In MAVLink mode the sticks reach the flight controller as MAVLink messages from a "ground station" with ID 255 (ELRS's default). That is why the port stops being an RC input (`SERIAL6_PROTOCOL` = 23) and becomes MAVLink, and why `MAV_GCS_SYSID` must be 255.

26.2 Requirements

- ELRS radio module and receiver with an **ESP** chip (the ones with WiFi). STM32 hardware cannot run MAVLink.
- **ELRS 3.5.0 or later** on radio and receiver, and **Backpack 1.5.0 or later**.
- With MAVLink on, the switch mode is Hybrid or 16ch/2. Wide is not supported.
- An internal module like the TX15's cannot output telemetry over USB: you need the Backpack over WiFi.

26.3 Step 1 · Check the link and the versions

1. Power on the radio and the aircraft (props off). On the TX15 press SYS and open **Tools › ExpressLRS**.
2. Look at the top right corner of the script: a **C** means radio and receiver are linked. In this build the C was there from the start: the receiver was bound.
3. The script header shows the radio module's version. The receiver appears under **Other Devices**; open it to see its version and options.
4. Look for the receiver's **MAVLink** protocol option. **In this build it was missing**: the receiver had a version older than 3.5 and had to be updated (step 2).

26.4 Step 2 · Update the RP4TD receiver over WiFi

a) Build the firmware first, while online

The receiver does not ship with the firmware file: you build it on the computer with **ExpressLRS Configurator**, the official app (not the receiver's web page). Build it **before** joining the receiver's WiFi, because that network has no internet and the Configurator needs to download the release.

1. Install and open **ExpressLRS Configurator** on the Mac.
2. Pick **Official Releases** and version **3.5.6** (the same branch as the radio).
3. **Device category**: RadioMaster 2.4 GHz. **Device**: the RP4TD (2.4 GHz receiver).
4. **Flashing method**: **WiFi**.
5. **Regulatory domain**: the same as the radio (mandatory; if they differ they will not link). **Binding phrase**: the same as the radio, if you use one.
6. Press **Build**. A temporary folder opens with `firmware.bin.gz`. **Do not unzip it** and keep the folder open.

b) Put the receiver in WiFi mode and open its page

1. Put the receiver in WiFi mode in one of two ways:
   · **Automatic**: power the aircraft with the radio off and wait about 60 s. The LED starts blinking fast.
   · **From the radio** (with the receiver bound): ExpressLRS script › **WiFi Connectivity › Enable RX WiFi**.
2. On the Mac, open the WiFi list and join **ExpressLRS RX** (password `expresslrs`).

> **Don't mix up the two networks**
>
> In this build the one that showed up at first was **ExpressLRS TX Backpack...**: that is the **radio's** Backpack network, not the receiver's. The receiver's is called **ExpressLRS RX** and only appears while the receiver is in WiFi mode.

3. In the browser, open `http://10.0.0.1`. The receiver page shows its version and model. Open the **Update** tab (on older versions, the *Firmware Update* section of the same page).
4. Under **Choose file**, select the `firmware.bin.gz` from the Configurator's temporary folder and press **Update**.
5. Wait one or two minutes without unplugging anything, until the success message. The receiver reboots and the LED returns to its normal pattern.

6. Rejoin your usual WiFi on the Mac. On the radio, ExpressLRS script › Other Devices: the receiver now shows **3.5.6** and the C is back.

26.5 Step 3 · Radio module and Backpack

- **Radio's internal module**: it already had a 3.5.x version, so it was left alone. If yours is older: ExpressLRS script › **WiFi Connectivity › Enable WiFi**, join the Mac to the **ExpressLRS TX** network (password `expresslrs`), open `http://10.0.0.1` › Update and upload the `firmware.bin` built with the Configurator for your radio's internal module (RadioMaster 2.4 GHz category, WiFi method). The first time from a very old factory version, the official guide asks for extra steps: read it first.
- **Backpack**: the script's **Backpack** menu shows its version. In this build it was **1.5.9**, which is fine (1.5.0 or later is needed).

26.6 Step 4 · Receiver protocol MAVLink

1. ExpressLRS script › **Other Devices** › the receiver.
2. Set the serial output protocol to **MAVLink**. In ELRS 3.x it is done here; from ELRS 4 it is set automatically when the radio switches to MAVLink.

26.7 Step 5 · Link Mode on the radio

Link Mode is the link mode of the radio module. **Normal** is the usual one: sticks and telemetry in ELRS's CRSF format, which the receiver outputs as an RC input. **MAVLink** carries sticks and telemetry in MAVLink messages.

1. Power off the aircraft (the receiver must be **disconnected**).
2. ExpressLRS script, main menu › **Link Mode** › **MAVLink**. If the receiver is connected, the radio answers **"Not while connected"**: power off the aircraft and try again.
3. Set power and **Packet Rate**. This build used **250 Hz**. In MAVLink the telemetry ratio is 1:2 and cannot be changed.
4. Power on the aircraft and wait for the **C**.

> **Packet rate and bandwidth**
> More hertz let more telemetry through but shorten range. For reference, the ELRS documentation estimates about 590 B/s downlink at 250 Hz, about 1470 B/s at 333 Hz Full and about 2375 B/s at F1000/K1000. If the simulator stutters, 500 Hz roughly doubles the bandwidth of 250 Hz. Always range-test on the ground after changing it. A handy approach: a high rate on the ground to download parameters or upload missions, and your usual rate to fly.

26.8 Step 6 · ArduPilot (QGroundControl › Parameters)

With the aircraft connected to QGroundControl over USB, change these parameters and reboot the flight controller. They are **ArduPilot 4.7** names; the old name is in brackets.

Parameter	Value	Why
`SERIAL6_PROTOCOL`	2 (MAVLink2)	The receiver port changes from RC input (23) to MAVLink.

`SERIAL6_BAUD`	460 (460,800)	The speed ELRS uses in MAVLink.
`RSSI_TYPE`	5	RSSI from the telemetry radio (MAVLink).
`MAV_GCS_SYSID` (`SYSID_MYGCS`)	255	ID of the station sending the sticks; ELRS uses 255 by default.
`MAV3_*` (`SR6_*`)	see below	Message rates on the receiver port.

Which number your port has. ArduPilot numbers MAVLink channels in order among the ports set to MAVLink. In this build: **MAV1** = USB, **MAV2** = `SERIAL1`, **MAV3** = `SERIAL6` (the receiver). If you set another port to MAVLink, the numbering changes.

Group (receiver port)	Messages the simulator uses	Starting value
`MAV3_EXTRA1`	ATTITUDE	4
`MAV3_EXTRA2`	VFR_HUD (speeds, altitude, throttle)	4
`MAV3_POSITION`	GLOBAL_POSITION_INT (position)	2
`MAV3_EXT_STAT`	SYS_STATUS (battery, sensors), GPS_RAW_INT	1
`MAV3_EXTRA3`	BATTERY_STATUS, WIND, VTOL state	1
`MAV3_RC_CHAN`	RC_CHANNELS (on-screen sticks)	1
`MAV3_RAW_SENS`	—	0 or 1
`MAV3_RAW_CTRL`, `MAV3_PARAMS`, `MAV3_ADSB`	—	**0** (the ELRS documentation asks for 0)

The ELRS documentation suggests 1 for every group except the last three. The values above raise attitude and VFR_HUD, which the instructor needs (2 or more messages per second). If they don't arrive at 250 Hz, lower `MAV3_EXTRA3` and `MAV3_RC_CHAN` or raise the packet rate.

> **QGroundControl changes the rates**
>
> With the Mac on the Backpack WiFi, QGroundControl connected by itself over UDP and raised the receiver port rates (for example `MAV3_EXTRA1` to 10). The link saturated and everything got slow. Fix: in QGroundControl, **Application Settings › General › AutoConnect** (in newer versions, **Comm Links**), turn off **UDP** auto-connect, and set your values again. Optional: ArduPilot 4.7 has `MAV3_OPTIONS` bit 2 ("ignore stream rate requests from the ground station"), which stops any ground station from changing them.

26.9 Step 7 · The Backpack WiFi

1. ExpressLRS script › **Backpack › Telemetry › WiFi**.

> **Don't use "WiFi Connectivity › Enable Backpack WiFi"**
>
> That option puts the Backpack in firmware update mode and it sends no telemetry.

2. On the Mac, join the **ExpressLRS TX Backpack XXXXXX** network (password `expresslrs`). The X's are part of your module's ID.
3. Check: in the browser, `http://10.0.0.1` opens the Backpack page with the MAVLink statistics.
4. Turn off any firewall that blocks UDP. If macOS asks whether Python may accept incoming connections, allow it.

26.10 Step 8 · AIRLAB Connector on the Mac

In this build a standalone program, `airlab-puente.py`, was used at first, started with a command in Terminal. Now **AIRLAB Connector** does it; it also serves the real ArduPilot (chapter 27) and opens with a double-click.

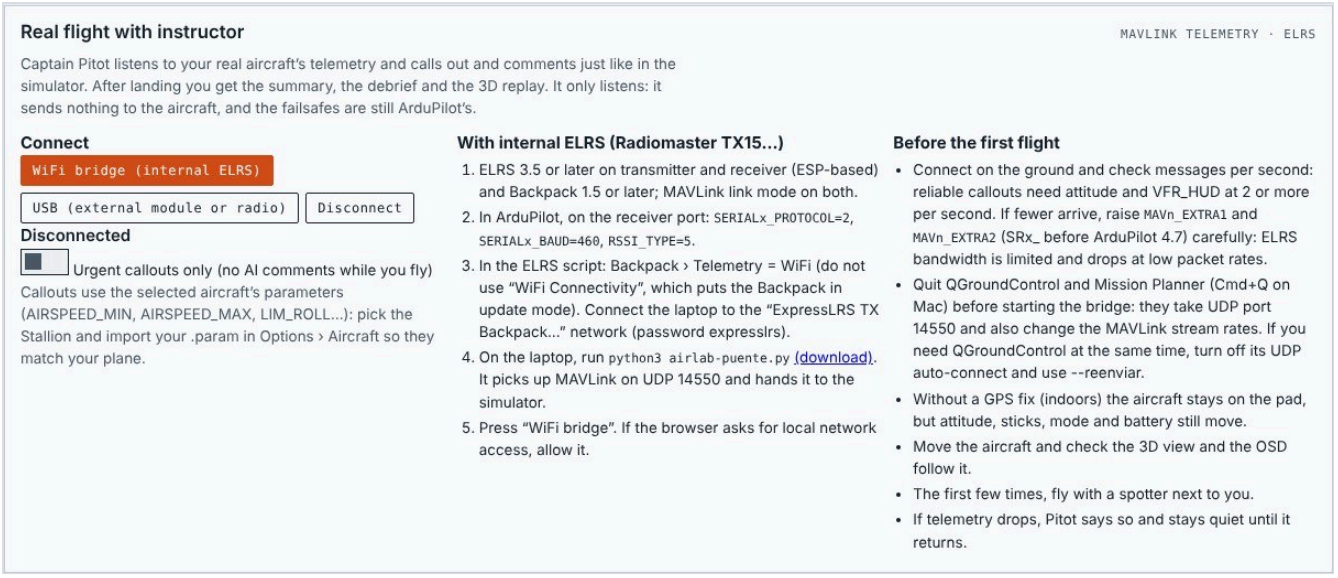
1. **Once:** install the connector from **Connect**: download `airlab-instalar.sh` and paste `bash ~/Downloads/airlab-instalar.sh` into Terminal. "AIRLAB Conector" appears on the Desktop.
2. **Quit QGroundControl and Mission Planner completely** (`Cmd`+`Q`): if they are open they take port 14550.
3. Double-click **AIRLAB Conector**. A Terminal window opens: leave it open while you fly.
4. What you should see (it prints in Spanish): "Vuelo real: escuchando la telemetría del Backpack (UDP 14550)" and, with the aircraft powered, "Vuelo real: recibiendo telemetría de 10.0.0.1:...". Every 10 s, a line with packets per second. In this build 5 or 6 packets per second arrived at 250 Hz.
5. To close it, `Ctrl`+`C` or close the window.

What you see	What it means
"el puerto UDP 14550 lo usa otro programa..." (port in use)	QGroundControl or Mission Planner are still open. Quit them with Cmd+Q and reopen the connector.
No "recibiendo telemetría" (receiving telemetry)	Nothing arrives: check the Mac is on the Backpack WiFi, Backpack › Telemetry is WiFi, Link Mode is MAVLink and there is a C.
Connect: "Simulator not connected"	The connector is receiving but the simulator has not hooked on yet: press **WiFi bridge** in the panel.

QGroundControl at the same time. Open the connector from Terminal with `python3 ~/AIRLAB/airlab-conector.py --reenviar 14560`. In QGroundControl turn off UDP auto-connect and create a manual UDP link on port 14560.

26.11 Step 9 · The simulator

1. Open the simulator in Chrome, go to **Connect › Real flight with instructor** and press **WiFi bridge**. If the browser asks for local network access, allow it.
2. The status changes to "Connected to the bridge, waiting for aircraft data..." and, with data, to ARMED or disarmed. Below it, **Messages/s**: attitude, position, heartbeat, battery and bad frames.
3. Move the aircraft by hand: the 3D view and the OSD follow it. Move the sticks: the on-screen sticks move too (they come from the telemetry).
4. **Without a GPS fix** (indoors), the aircraft stays on the pad: attitude, sticks, mode and battery move. With a GPS fix, the takeoff point becomes home and the aircraft moves around the scenery.



Connect › Real flight with instructor: connection, steps with internal ELRS and checks before the first flight.

26.12 Before flying in MAVLink mode

> **Failsafe test is mandatory**
>
> In MAVLink mode the sticks reach the flight controller differently. Before the first flight, with props off: arm on the ground, switch the radio off and check in QGroundControl that the radio failsafe triggers and does what you configured (QLAND in this build). Switch the radio back on and check how control is recovered. Repeat it whenever you change anything in the link.

- Do a ground range test at the packet rate you will fly with.
- Check in the panel that attitude and VFR_HUD arrive at 2 or more per second.
- Does control get worse? With 1:2 telemetry, one packet in two is telemetry, so the sticks update at half the packet rate (about 125 per second at 250 Hz), plenty for a plane. The connector and the simulator only listen and add no traffic towards the aircraft. What does cut range is raising the packet rate: don't fly at the edge of range with a high packet rate.
- The first few times, fly with a spotter next to you.

26.13 Internet at the field

With the Mac on the Backpack WiFi there is no internet, and the AI instructor needs it. The planned solution: share the iPhone's data **over a USB cable** (Personal Hotspot) and, in **System Settings › Network**, menu **··· › Set Service Order**, put **iPhone USB** above **Wi-Fi**. The WiFi keeps receiving the Backpack telemetry and internet traffic goes through the iPhone. Test it at home before going to the field. The offline version doesn't use AI: the rule-based callouts work the same.

26.14 Problems that came up and fixes

Symptom	Cause	Fix
No MAVLink option on the receiver	Receiver older than ELRS 3.5	Update it to 3.5.6 over WiFi (step 2).

Only “ExpressLRS TX Backpack” shows in the network list	That is the radio’s network; the receiver wasn’t in WiFi mode	Put the receiver in WiFi mode and look for “ExpressLRS RX”.
The receiver page asks for a file you don’t have	The firmware doesn’t ship with the receiver	Build `firmware.bin.gz` with ExpressLRS Configurator (WiFi method).
“Not while connected” when changing Link Mode	The receiver was connected	Power off the aircraft, change Link Mode and power on.
The bridge says “Address already in use” or the port is taken	QGroundControl held UDP 14550	Quit QGroundControl with Cmd+Q and reopen the bridge (now AIRLAB Connector).
Everything is slow	QGroundControl, connected over UDP, raised `MAV3_*` and saturated the link	Turn off QGroundControl’s UDP auto-connect and set the rates again.
Connected, but the aircraft doesn’t move	No GPS fix indoors	Normal: attitude, sticks and mode move; position arrives with a GPS fix.
No internet for the AI at the field	The Mac is on the Backpack WiFi	iPhone over USB with priority over Wi-Fi.

26.15 Summary of values

Where	Setting
Receiver (Other Devices)	ELRS 3.5.6 · MAVLink protocol
Radio (ExpressLRS script)	ELRS 3.5.x · Link Mode MAVLink · 250 Hz (500 Hz optional) · telemetry fixed at 1:2
Backpack	1.5.9 · Telemetry = WiFi · network “ExpressLRS TX Backpack…”, password `expresslrs`, `10.0.0.1`, UDP 14550
ArduPilot	`SERIAL6_PROTOCOL` 2 · `SERIAL6_BAUD` 460 · `RSSI_TYPE` 5 · `MAV_GCS_SYSID` 255 · `MAV3_*` per the table
QGroundControl	UDP auto-connect off
Mac	AIRLAB Connector: double-click on the Desktop (installed once from Connect)
Simulator	Connect › Real flight with instructor › WiFi bridge

Sources: ExpressLRS · MAVLink · ExpressLRS · Updating receivers · ExpressLRS · RadioMaster internal modules · ArduPilot · Crossfire and ELRS · ArduPilot · Parameter name changes (4.7).

27 Connections, real ArduPilot (SITL), phone viewer and remote instructor

A browser cannot open UDP ports or start programs. To connect the simulator to your real aircraft (chapter 16) or to the real ArduPilot you need a program on your computer: **AIRLAB Connector**. You install it once and then open it with a double-click. **Connect** shows live what works and what is missing.

27.1 Installing AIRLAB Connector (once, on the Mac)

1. In the simulator, **Connect** › **airlab-instalar.sh**. It downloads to Downloads.
2. Open Terminal (`Cmd` + `Space`, type “terminal”) and paste this command; the page’s **Copy** button copies it for you:

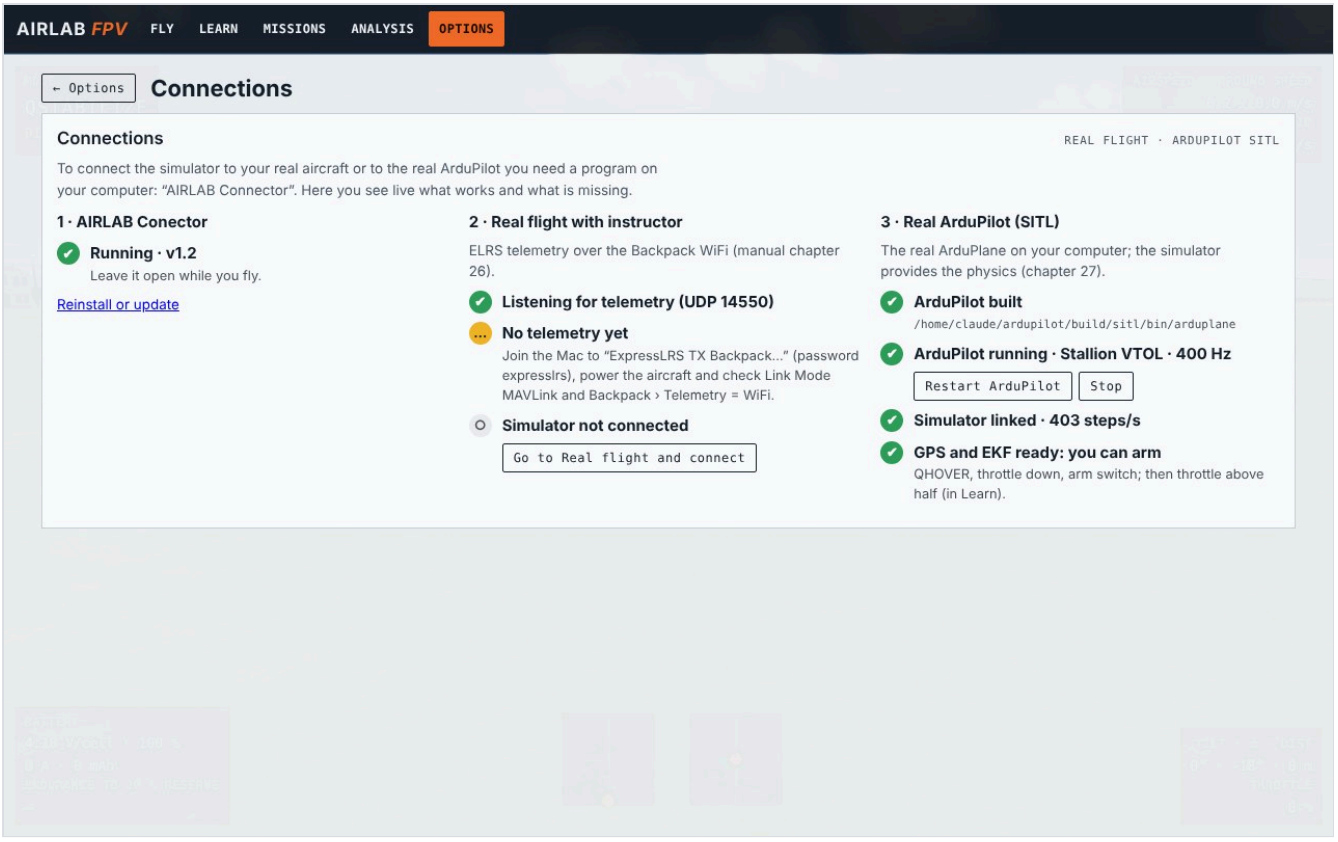
   ```
   bash ~/Downloads/airlab-instalar.sh
   ```

3. The installer puts everything in the `~/AIRLAB` folder, places **AIRLAB Conector** on the Desktop and opens it. If macOS asks to let Terminal access the Desktop, allow it.

After that, double-click **AIRLAB Conector** and leave it open while you fly; the first line of its window shows the version (this guide is for **1.2**). You only need it for real flight and SITL: flying with the simulator’s own autopilot does not use it. Running the installer again updates it and the page tells you when there is a newer version. Before that, close the connector (`Ctrl` + `C`) and delete the old installer from Downloads: otherwise Chrome saves the new one as `airlab-instalar (1).sh` and you would end up running the old one. It only needs the Mac’s Python 3: if it is missing, the installer opens the Apple developer tools installation.

> **It does not touch your aircraft**
>
> The connector sends nothing to the flight controller: with the real aircraft it only listens to telemetry, and with SITL it only talks to the ArduPilot running on your Mac. Its parameters are stored in `~/AIRLAB` . The only thing that changes your flight controller is what you do yourself in QGroundControl connected to it: SITL is TCP `127.0.0.1:5760` , the aircraft is USB or the Backpack’s UDP.

27.2 Connect



The Connect tab with the connector running, ArduPilot running and ready to arm. Each row is green when it is fine, yellow while waiting and red when you need to do something; below it says what.

Block	What it checks
1 · AIRLAB Connector	Whether it is running and its version. If not, it shows the installation steps.
2 · Real flight with instructor	That UDP port 14550 is free (if not, QGroundControl or Mission Planner hold it), that aircraft telemetry is arriving and that the simulator is connected. Button to go to the real-flight panel.
3 · Real ArduPilot (SITL)	That ArduPilot is built, whether it is running and with which aircraft (with **Start**, **Start from scratch**, **Restart** and **Stop**), that the simulator is linked and that ArduPilot already has GPS and EKF to arm.

27.3 Real ArduPilot (SITL)

SITL (*Software In The Loop*) is ArduPilot's code, the same one the flight controller runs, executing on the computer. Here ArduPilot provides the autopilot (modes, PIDs, TECS, transition, limits, missions, failsafes) and the simulator provides the physics, the 3D and the scenery. What you fly is decided by the `.param`: the test one or yours. ArduPilot and the simulator talk over SITL's JSON interface: at every step ArduPilot sends its servo and motor outputs, and the simulator returns sensors, position, velocity and attitude, always in lockstep (400 times per second with the VTOL, 200 with the fixed wing).

Aircraft	What ArduPilot flies	Test .param
Stallion VTOL (tilt-rotor)	Tricopter QuadPlane with vectored tilt: hover, yaw by tilting the front motors, transition, airplane, QRTL and QLAND	`stallion-vtol-sitl.parm`

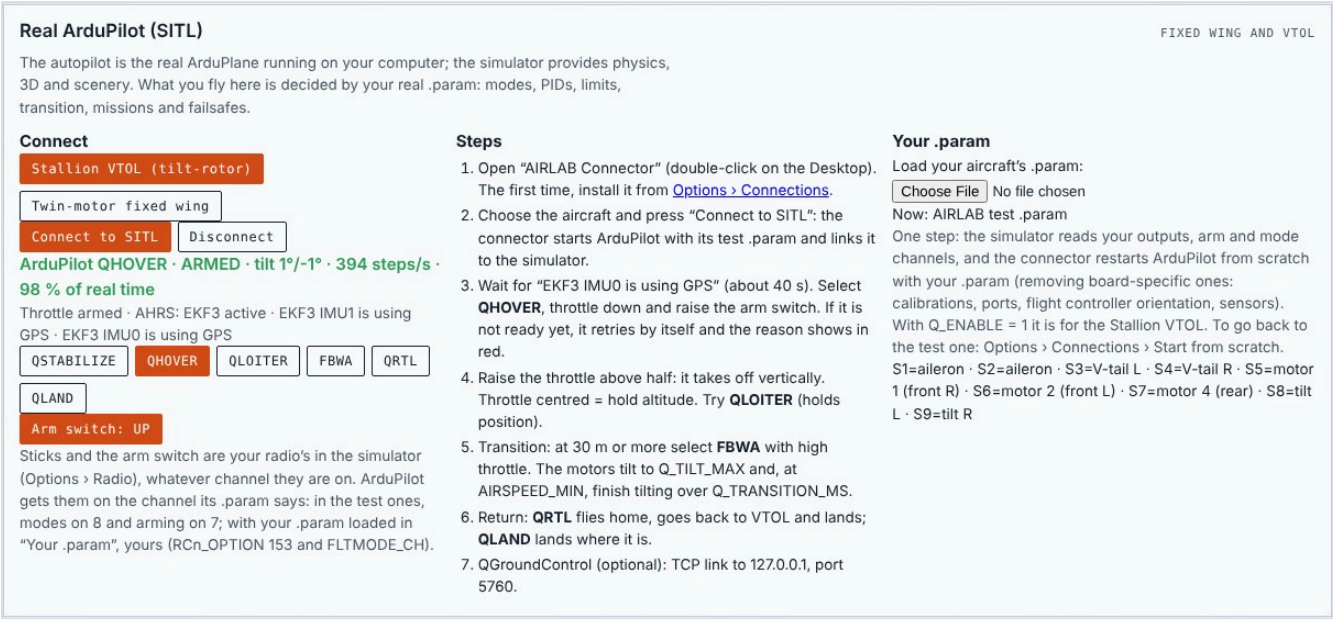
Twin-motor fixed wing	The Stallion without VTOL: hand launch, FBWA, CRUISE, AUTO, RTL, LOITER	`stallion-sitl.parm`

Building ArduPilot (once, 20-40 min)

With the connector installed, in Terminal:

```
bash ~/AIRLAB/compilar-ardupilot.sh
```

The script checks the Apple developer tools and Homebrew, downloads ArduPilot to `~/ardupilot`, installs its dependencies and builds ArduPlane for SITL. The same program is used for the fixed wing and the VTOL: nothing else needs building. If it installs dependencies, at the end it asks you to close Terminal and run it again. Answers to its questions: Python with pyenv → `y` · STM32 toolchain → `n` · everything else → `y`. If something fails, the official guide is *Setting up the Build Environment (MacOSX)* on ardupilot.org.



Connect › Real ArduPilot (SITL) with the Stallion VTOL: aircraft, ArduPilot status (mode, armed, tilt angle, steps per second and % of real time), its messages, the channel 8 modes, the arm switch and "Your .param".

27.4 Flying the Stallion VTOL with ArduPilot

1. Open **AIRLAB Conector**.
2. **Connect › Real ArduPilot (SITL)** › **Stallion VTOL (tilt-rotor)** › **Connect to SITL**. The connector starts ArduPilot with the VTOL `.param` and links it; if the fixed wing was running, it stops it by itself.
3. Select **QHOVER**, throttle fully down and raise the **arm switch**. You can do it right after connecting: while ArduPilot is not ready (GPS and EKF, about 40 s), the simulator lowers and raises the switch every 4 s, as you would, and the reason shows in red below ("AHRS: waiting for home", "not using configured AHRS type"...). It arms as soon as it can: "Throttle armed".
4. Throttle above half: it takes off vertically. Centred throttle holds altitude. With the rudder, ArduPilot tilts one front motor forward and the other back and the aircraft turns (Q_TILT_YAW_ANGLE).
5. **QLOITER** holds position even in wind. **QLAND** descends, lands at Q_LAND_FINAL_SPD, says "Land complete" and disarms.

6. **Transition**: at 20-30 m select **FBWA** with high throttle. “Transition airspeed wait” (motors at Q_TILT_MAX, 45°) → “Transition airspeed reached” at AIRSPEED_MIN → the motors finish tilting over Q_TRANSITION_MS → “Transition FW done”.
7. **QRTL**: flies home as an airplane, brakes (“VTOL airbrake”), goes back to VTOL, approaches at Q_RTL_ALT and lands vertically.

Output	Function	In the simulator
5 · 6 · 7	33 · 34 · 36 (Motor1, Motor2, Motor4)	Front right, front left and rear motor. On a tricopter ArduPilot uses motors 1, 2 and 4.
8 · 9	75 · 76 (TiltMotorFrontLeft / Right)	Angle of each front motor. The output spans SERVOn_MIN…MAX: one end is Q_TILT_YAW_ANGLE behind vertical and the other, horizontal.
1-2 · 3-4	4 · 79/80	Ailerons and V-tail.

- The test `.param` carries the values from the Flightory manual (Q_FRAME_CLASS 7, Q_TILT_MASK 3, Q_TILT_TYPE 2, Q_TILT_YAW_ANGLE 15, Q_TILT_MAX 45, Q_TILT_RATE_UP 15, Q_ASSIST_SPEED −1, Q_OPTIONS 163841, Q_TRANSITION_MS 7000, Q_LAND_FINAL_SPD 0.3), AIRSPEED_MIN 16 / CRUISE 19 / MAX 25, a 6S battery and ArduPilot’s starting VTOL PIDs for a tricopter tilt-rotor. Modes on channel 8: QSTABILIZE · QHOVER · QLOITER · FBWA · QRTL · QLAND.
- The simulator’s motor responds the way ArduPilot assumes (Q_M_SPIN_MIN…MAX and Q_M_THST_EXPO), so Q_M_THST_HOVER learns the model’s real hover throttle.
- The **Failures menu** still works with the real ArduPilot: stop a motor or jam a tilt servo and see what your setup does.
- In FBWA with high throttle the model goes past 35 m/s: that is the propeller and drag of the simulator’s model, not ArduPilot. In CRUISE, AUTO and QRTL the TECS takes over with AIRSPEED_CRUISE.

27.5 Flying the fixed wing

1. **Connect › Real ArduPilot (SITL) › Twin-motor fixed wing › Connect to SITL**.
2. **FBWA**, throttle down and raise the arm switch (it arms by itself as soon as ArduPilot is ready).
3. Throttle to 70-85 % and **Hand launch**. Try RTL (comes back and circles home), LOITER, CRUISE and MANUAL. In automatic modes, throttle above half raises the target airspeed (`THROTTLE_NUDGE`).

27.6 Your .param in one step

In the panel’s **Your .param** card, **Choose file** and your aircraft’s `.param`, exactly as QGroundControl or Mission Planner save it. There is no need to press “Connect” or open QGroundControl. With `Q_ENABLE = 1` it applies to the Stallion VTOL.

- **The simulator** reads your outputs (SERVOn_FUNCTION, MIN, MAX, TRIM, REVERSED), which channel you arm on (`RCn_OPTION 153` ; with ELRS, usually 5) and which channel changes modes (`FLTMODE_CH`). The six mode buttons now show your FLTMODE1…6. Your radio’s arm switch in the simulator (Options › Radio) reaches ArduPilot on your channel, whichever it is.
- **The connector** restarts ArduPilot from scratch with your `.param` . First it removes what belongs to the board: flight controller orientation (`AHRS_ORIENTATION`), IMU, compass and barometer calibrations, serial ports, GPS, airspeed sensor, battery multipliers and radio calibration. On top it adds what SITL

needs: a 200 Hz loop and 400 Hz physics, PWM motors instead of DShot and the ground station failsafe off (without QGroundControl it would trigger at once). Everything else is yours: PIDs, transition, speeds, radio and battery failsafes, modes, tilt.

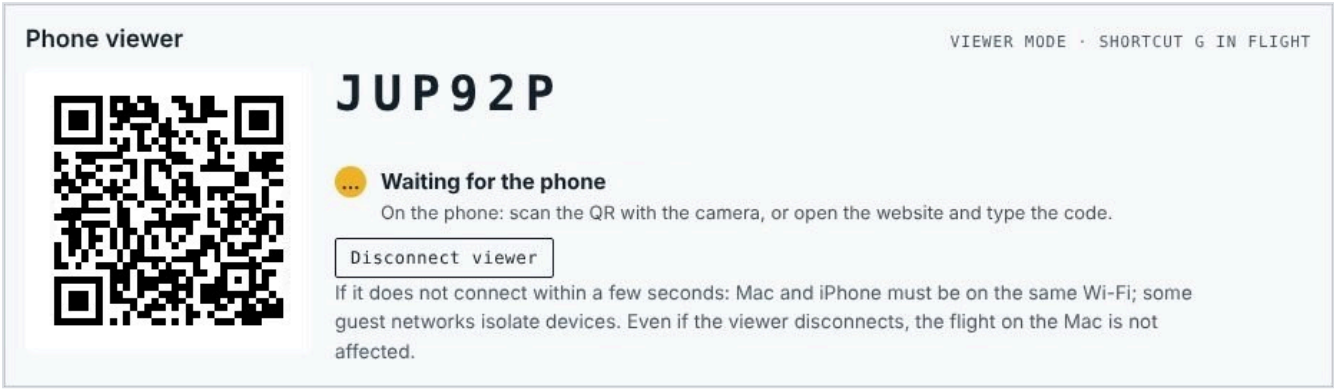
- Your file is not touched: the SITL version is stored separately in `~/AIRLAB/mi-param-vtol.parm` and is used the next times too. To go back to the test one: **Connect › Start from scratch**.
- If a reason shows in red below the arm switch, it is a PreArm from your setup: fix it in QGroundControl (TCP link `127.0.0.1:5760`) or in your `.param` and load it again.

27.7 Restart, from scratch and QGroundControl

- **After a crash**, press **Reset**: the connector restarts ArduPilot, like power-cycling the flight controller. The aircraft returns to the pad with new GPS and EKF and the parameters are kept. The aircraft cannot be “teleported” without a restart: the EKF would see a position jump and refuse to arm (“GPS and AHRS differ”).
- **Start from scratch** (in Connections) erases the stored parameters and your loaded `.param`, and goes back to the chosen aircraft’s test one. Each aircraft keeps its own in its folder (`~/AIRLAB/sitl-run` and `sitl-run-vtol`).
- **QGroundControl** is not needed. Open it if you want to see or change parameters, upload missions for AUTO or see messages and the map: TCP link to `127.0.0.1`, port `5760`.
- The panel shows the **steps per second** (400 with the VTOL, 200 with the fixed wing) and the **% of real time**. If it drops well below 100 %, the computer cannot keep up: close tabs or turn on the light scenery.
- Closing the connector (`Ctrl` + `C` or closing the window) also stops the ArduPilot it started.

27.8 Phone viewer (viewer mode)

You fly on the Mac with your radio, as usual, and watch the scene live on an iPhone inside a headset (for example, one you printed yourself), like FPV goggles. **It is not a video**: the phone loads the same simulator, the same aircraft and the same scenery and draws them itself at its own resolution. Only the flight state travels over the network (position, attitude, motors, battery, mode…), about 60 times a second, so the image is sharp and the delay is very small. The Mac still does everything: physics, radio, autopilot, callouts and AI. The phone has no controls or menus, uses no licence or AI queries, and if it disconnects **the flight on the Mac is not affected**.



Connect › Phone viewer, waiting for the phone: QR, code and status. When the phone connects, the row turns green and shows the round-trip time (RTT), packets per second, loss and the phone’s frame rate.

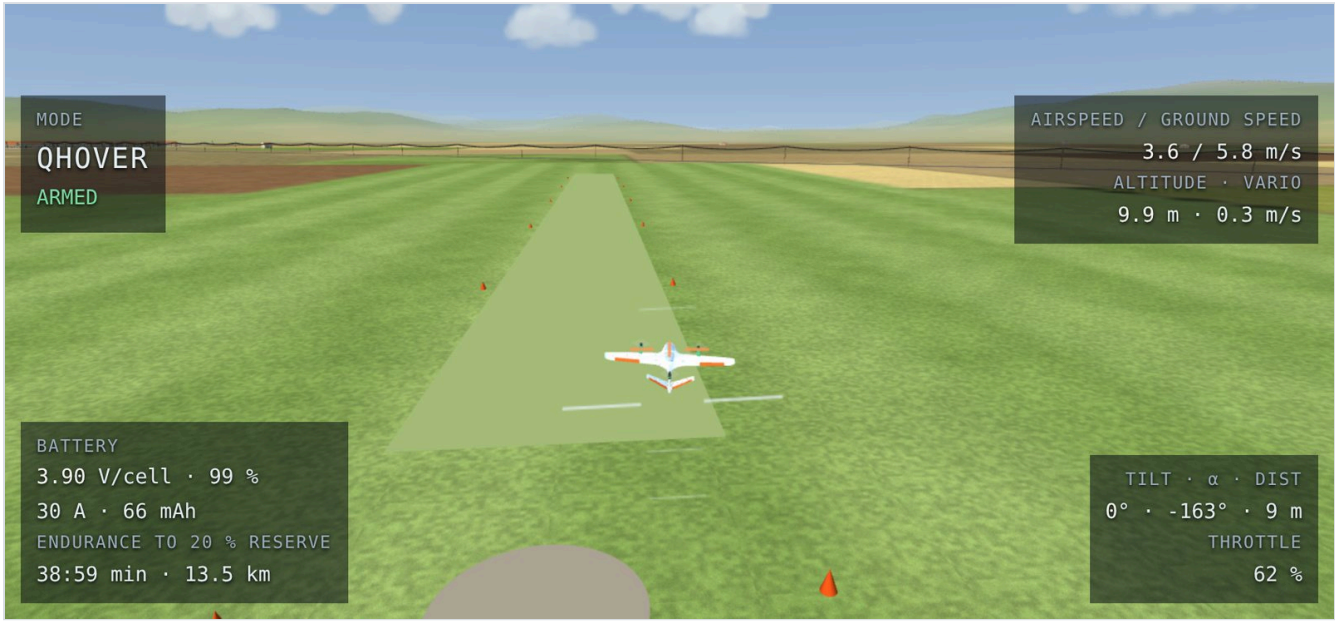
Pairing (the first time, one minute)

1. Mac and iPhone **on the same Wi-Fi**. Some guest networks isolate devices from each other: it will not connect there.
2. On the Mac, in the web version: **Connect › Phone viewer › Turn on phone viewer**, or the `G` key during the flight. A QR and a 6-character code appear (no 0/O or 1/I, to avoid mix-ups).
3. On the iPhone, point the **camera** at the QR and open the link. You can also open the website, go to `/app.html?ver` (or “Watch as viewer” on the start menu) and type the code. The phone does not need your account: the code works while the viewer is on on the Mac and expires 10 minutes after you close it.
4. Choose **Normal screen** or **VR headset (two lenses)** (iOS needs a tap to keep the screen on and for sound), turn the phone sideways and put it in the headset. On the Mac, the row turns green: **Connected**.

The phone follows the Mac all the time: if you change aircraft, scenery, time of day or camera (FPV, chase, free…) or start a work mission, the viewer changes by itself. Replays with the instructor’s blue aircraft are shown too.

Full screen on the iPhone

Safari does not let a page hide its bars. For full screen, the first time tap **Share › Add to Home Screen** and open the viewer from that icon (“AIRLAB visor”): it opens in black, without bars, and remembers the last code. The screen stays on while the viewer is open (in Home Screen apps, from iOS 18.4; on an older iOS, set **Settings › Display & Brightness › Auto-Lock › Never** while you fly). In portrait it shows “Turn the phone sideways”: Safari does not allow locking the orientation.



What the phone shows: the same scene drawn by the iPhone itself, with the OSD. No menus or buttons.

Viewer settings (on the phone)

Double tap or long press on the screen opens the menu; it hides itself after a few seconds. Everything is saved on the phone.

Setting	What for

Sharpness, shadows, draw distance, tree density, max frame rate	The viewer's own graphics quality. Inside a closed box the phone gets hot: by default, sharpness 1.5, no shadows and 60 fps at most. If it gets hot or stutters, go down to 30 fps and fewer trees.
Zoom and horizontal and vertical offset	To centre the image on the headset lens.
Respect the island / notch	Off, the image uses the whole screen (normal inside the headset).
Brightness	Darkens the image (with the headset against your eyes, full brightness is tiring).
OSD and OSD size	The same OSD as on the Mac (including your own custom one), bigger or smaller.
Motor sound	Synthesised on the phone from the throttle and rpm sent by the Mac.
Pitot callouts	Subtitle, voice or none.
VR headset (two lenses)	Headset mode, 3D and its strength, lens separation and height, lens correction (k1, k2), field of view, OSD size in the headset and adjustment grid (see below).
Delay (smoothness)	The phone draws about 40 ms behind so that it always has two states to interpolate between. Less delay = more immediate but more stutter if the Wi-Fi drops packets.



Viewer settings, with the connection status at the bottom.

Reactions, voice and looking around

- **Reactions**: on the normal screen, at the bottom right there is a row of emojis (❤️ 👏 🔥 😮 😂 👍). Each tap rises up the right side of the flight on the Mac, with the name you set in **Viewer settings › Your name**. It lets whoever watches you from another city cheer you on while you fly.
- **Voice**: on the phone, hold 🎤 to talk to the pilot (the first time it asks for microphone permission). On the Mac, "🎤 … is talking" appears and you hear it. The pilot answers with **Talk to the phone** in the

viewer panel (one tap turns it on and off, so your hands stay on the radio); the phone shows “The pilot is talking”.

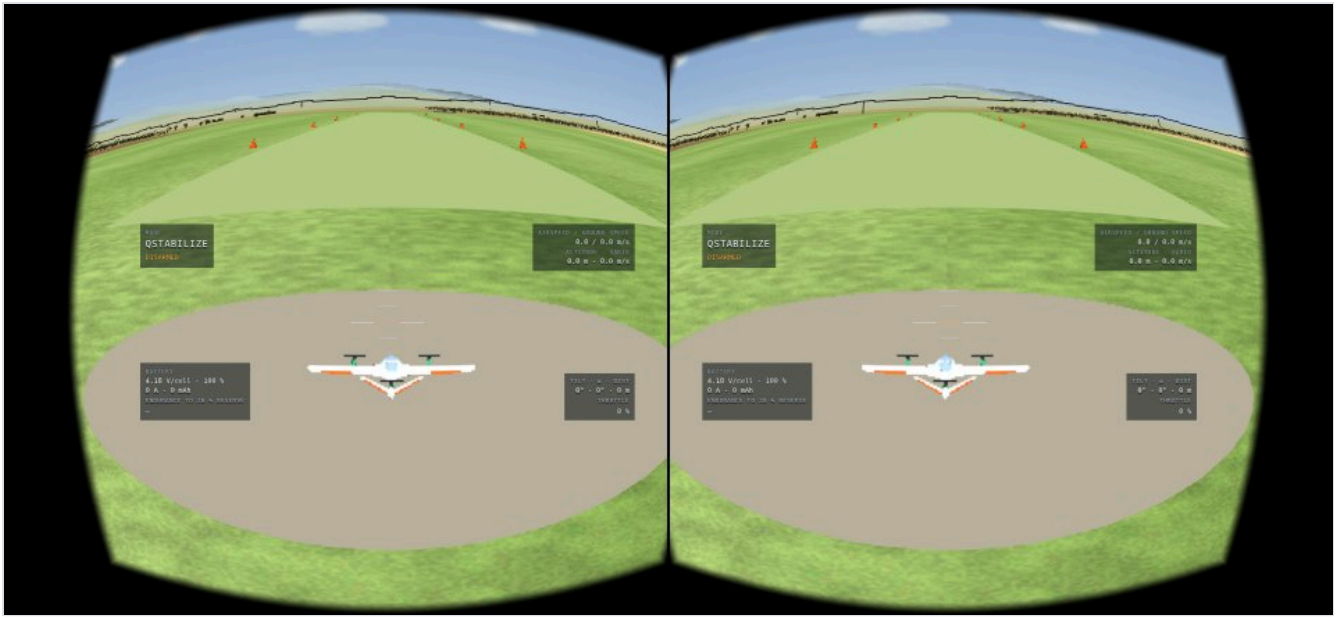
- **Look around**: when you turn the phone, or your head with the VR headset, the camera turns with it, as if you were the co-pilot; the flight on the Mac does not change. **Center the view** in the menu sets the front to where you are looking now. Turn it off with **Look around by moving the phone**. On the iPhone, the first time it asks for permission to use motion.

Reactions and voice can be hidden with **Reaction and voice buttons**. In headset mode the buttons are not shown (the phone is inside the headset), but looking around still works.

Two-lens VR headsets and 3D

With a VR headset for phones (the two-lens ones sold for €15-35), choose **VR headset (two lenses)** when starting, or turn on **Headset mode** in the viewer menu. The image is doubled, one per eye, each centred on its lens and with the barrel distortion that cancels the lenses’ own, so the horizon looks straight. The OSD appears in each eye, smaller, so it falls inside the lens.

3D is on by default: since the phone draws the scene itself, it draws it twice with two cameras spaced like your eyes (6.4 cm) and you see real depth, something real FPV goggles cannot do. It shows most up close (trees, the runway, the pad when landing). You can turn it off and watch in 2D, or raise its strength to exaggerate depth in the chase camera. 3D tires the eyes more: start with short flights and turn it off if you feel dizzy.



Headset mode: one image per eye, with the distortion that cancels the lenses’ own and the OSD inside each lens.

Adjusting your headset (once):

1. Turn on **Grid to adjust the lenses** in the viewer menu and put the phone in the headset, centred.
2. With the headset’s wheels, adjust the **focus** until the grid looks sharp.
3. If you see two red dots or a double image, change **Lens separation** (63 mm to start) until the two red dots merge into one. **Lens height** moves both images up or down if the dot is not centred.
4. If the grid lines bulge outwards, raise **k1**; if they bend inwards, lower it. **k2** corrects the same but only at the edges.
5. **Field of view** widens or narrows the image; **OSD size in the headset** makes it bigger or smaller until you can read it without moving your eyes.

6. Turn the grid off. The settings are saved on the phone.

If something goes wrong

- **It stays on "Waiting for the pilot…"**: check that both are on the same Wi-Fi (not the guest one, not mobile data) and that the viewer is still on on the Mac. Switching to the 5 GHz Wi-Fi usually works better.
- **"The code has expired"**: you pressed "Disconnect viewer" on the Mac or 10 minutes passed with the viewer closed. Turn it on again and scan the new QR.
- **"Another phone is already connected"**: the viewer takes one phone. Disconnect the other one on the Mac.
- **"Connection lost, reconnecting…"**: the image freezes and the phone reconnects by itself as soon as it can; nothing happens on the Mac.
- **Stutter**: check loss and fps in the viewer menu. With loss, raise the delay; with low fps, lower the quality.
- **Double or blurry image in the headset**: adjust focus and lens separation with the grid (above). If it is still tiring, turn 3D off.
- For now it works in the **web version**. It will come to the offline version later.

27.9 Remote instructor

Your instructor watches your flight live from their computer, wherever they are, and from there talks to you, sets failures, changes the wind or the exercise and, if needed, takes the controls with their radio, like a school's trainer cable. It uses the same system as the phone viewer: no video travels, only the flight state, and the instructor's computer draws the same world.

Inviting the instructor (student)

1. In **Connect › Remote instructor**, press **Invite an instructor**.
2. A 6-character code and a link appear. Press **Copy the link** (or **Share…**) and send it by WhatsApp or e-mail.
3. When they join you will see their name in the panel and, during the flight, a label at the top centre.

In the same panel you decide what they can do: **take the controls**; **set failures and change the wind, time and weather**; **change the mode, arm, reset and choose the lesson, challenge, aircraft or scenario**. All three are on by default. **End the lesson** disconnects them.

The instructor station

The instructor opens the link in their computer's browser (or goes to the website with `/app.html?instr` and types the code), enters their name and presses **Connect as instructor**. No account is needed. The flight is on the left and the panel on the right:

- **Flight**: mode, height, speeds, battery, distance home and phase, a map with the track and the camera (like the student, chase, FPV, free or from the ground).
- **Voice and messages**: hold the `space bar` (or the button) to talk, or leave the mic always on. There are quick messages («Climb», «Land now», «Very good»…) and a text field: the student sees them large and the simulator voice reads them out.
- **Controls and mode**: **Take the controls** (key `T`) with their radio connected to the computer (using their browser's calibration) or with the keyboard (arrows, `W`/`S` throttle, `A`/`D` rudder). Mode

buttons, **Arm**, **Disarm** and **Reset**.

- **Failures**: the same as the failure list (chapter 15), including GPS.
- **Wind, time and weather** and **Exercise, aircraft and scenario**: start a lesson or a challenge, change aircraft or scenario (with the student disarmed) and pause the flight.
- **Log**: what happens in the student's flight (landings, transitions, failures, mode changes) and what the instructor does.
- **Assessment to the student's logbook**: a score from 0 to 10 and a comment saved in the student's last flight (chapter 29).



Instructor station: the student's flight live and, on the right, telemetry, track map, camera, voice and messages.



In the student's simulator: banner showing that the instructor has the controls, with the button to take them back, and their name at the top.

From a phone, and voice diagnostics

The instructor station also works on a phone. In portrait, the 3D scene, with the camera you choose (like the student, chase, FPV, free or from the ground), takes the top part and the panel sits below it, scrolling. In landscape the panel is narrower. **Show only the flight** folds the panel and leaves the scene full screen; the **Instructor panel** button opens it again.

Below the talk button, the instructor sees a meter of their own microphone and what is happening with their voice: *closed*, *sending N kbit/s* or *on, but it picks up nothing* (mic muted or used by another app). The student's simulator also reports what reaches it: ✓ *The student hears you*, *The student is not receiving your audio* or that *the student's browser has blocked sound*. That happens because browsers do not play anything until the user touches the page, and to the browser the radio is a game controller, which does not count. The simulator prepares sound when the student presses **Invite an instructor** or the microphone and, if the browser still blocks it, the student sees a large button, **Click here to hear your instructor**.

> **Instructor controls**
>
> When the instructor takes them, the student sees a red banner, «The instructor has the controls», with a **Take the controls back** button: the student can always take them back. If the instructor's sticks stop arriving for half a second (the connection drops), the controls return to the student automatically. Before taking them, the instructor should set their throttle where it is needed: in QHOVER minimum throttle makes it descend.

Voice

The instructor's browser asks for microphone permission when connecting. The student turns theirs on with **Talk to the instructor** or the `I` key, and off the same way. Use headphones so the motor sound does not feed back. Each panel adjusts the volume of the other person's voice.

On different networks

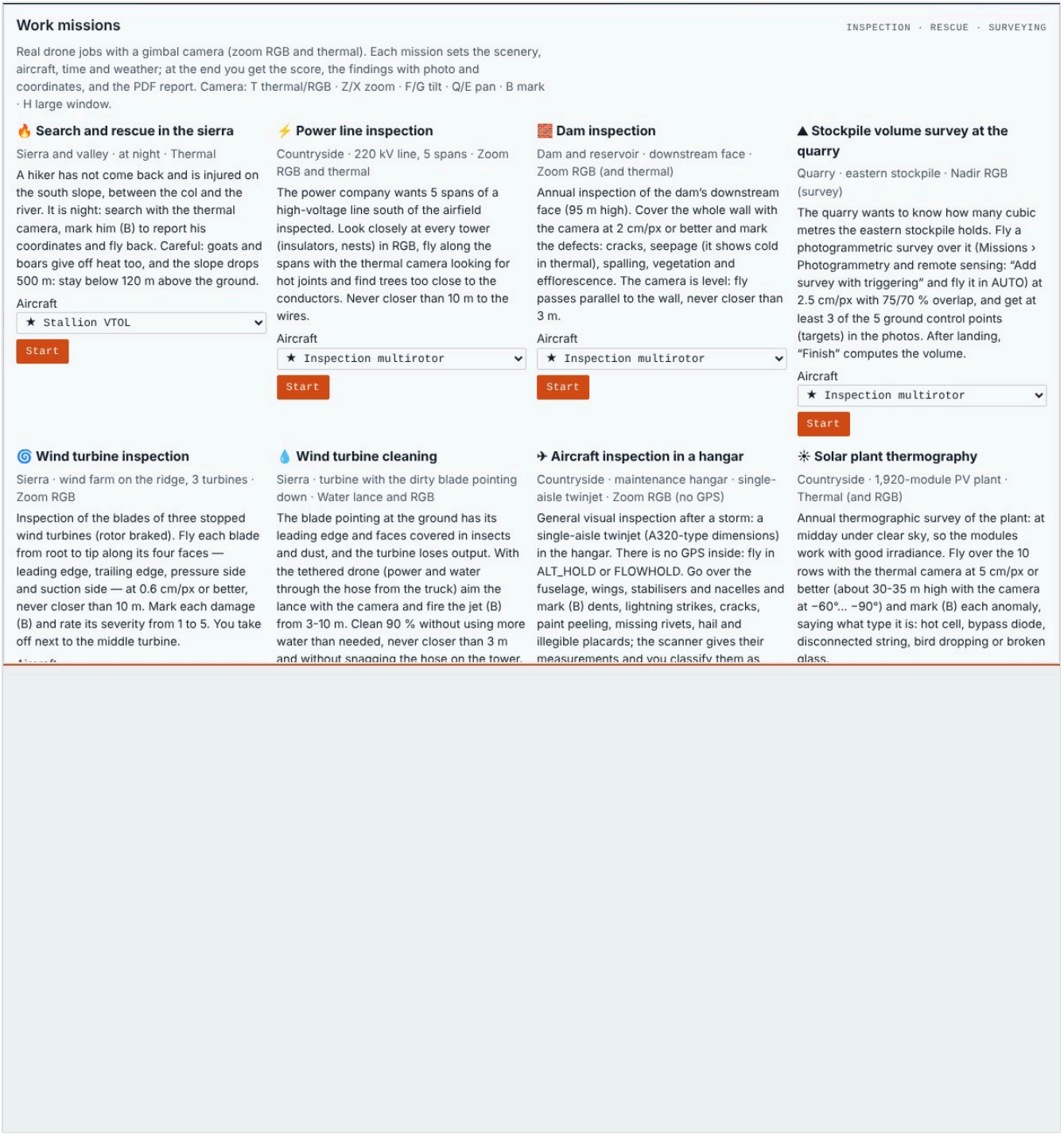
The two computers connect directly. Between two homes that usually works, but some networks (companies, schools, mobile data) block direct connections. For those, the server can provide a **TURN** server, an intermediary that relays the encrypted data. It does so if the variables `CF_TURN_KEY_ID` and `CF_TURN_KEY_TOKEN` are set (Cloudflare Realtime › TURN). With TURN the panel says so, and the phone viewer also works away from your home Wi-Fi.

Very little data travels: the flight state and the voice, about 100–150 MB per hour of lesson.

Sources: ArduPilot · JSON interface · ArduPilot · Tilt Rotor Planes · ArduPilot · Build environment (MacOSX) · `ArduPlane/tiltrotor.cpp` , `libraries/AP_Motors/AP_MotorsTri.cpp` and `libraries/SITL/examples/JSON/readme.md` in the ArduPilot repository · Flightory Stallion VTOL V2 manual (pp. 18-22) · MDN · RTCDataChannel · MDN · Screen Wake Lock API · WebKit · Wake Lock in Home Screen web apps (iOS 18.4). · Cloudflare · Realtime TURN · MDN · Perfect negotiation

28 Work missions

Twelve real drone jobs in the simulator's sceneries: night search and rescue; inspection of a high-voltage power line, a dam, wind turbines, an aircraft in a hangar, a solar plant, the inside of a tank, a telecom tower and a bridge; blade cleaning with a tethered drone, a stockpile volume survey and a hotspot search after a wildfire. They are practice for companies, schools and pilots preparing for their first job: there is a brief, a payload camera, safety rules, a score out of 100 and a PDF report like the one you would hand to a client.



Missions › Work missions: the brief for each job, the scenery, the camera and your best score.

28.1 Starting a mission

1. **Missions › Work missions**. On the job's card, choose the **Aircraft** (recommended ones are marked ★) and press **Start**.
2. The mission sets everything up: scenery, the aircraft chosen on the card, time of day, weather and wind, and the **chase** camera. It places the targets at random, so every attempt is different.
3. Go back to **Fly**. Top left is the mission panel (progress, flight time, height above ground, distance to the wires or the wall) and on the right the camera window.
4. Take off, do the job and fly back to the pad. Land, disarm and press **Finish and score**.

Retry starts again with new targets. **Exit** puts the scenery, aircraft, time and camera back as they were. Changing scenery midway closes the mission. It does not work with real ArduPilot (SITL) connected: disconnect it first.

28.2 Choosing the aircraft

Each card has its **Aircraft** selector. The recommended ones (★) come first: they are what a real job would use. All the others are listed below in case you want to try. The best score is kept per mission and per aircraft.

Mission	Recommended	Why
Search and rescue	Stallion VTOL (or any VTOL); multirotor	A lot of slope has to be covered with endurance; the multirotor is for confirming up close.
Power line	Inspection multirotor	You have to stop in front of each tower and get close precisely. In real work, the VTOL is used for many kilometres of line.
Dam	Inspection multirotor	You need to hold still in front of the wall in wind: LOITER.
Quarry	Inspection multirotor or VTOL	It is an AUTO survey: both can do it. For a small stockpile the multirotor is the usual choice.
Wind turbines, tower, bridge	Inspection multirotor	You need to hold still close to the structure in wind. Under the bridge, the caged drone too.
Blade cleaning	Tethered cleaning drone	It is the only one with a hose and lance.
Aircraft in hangar	Inspection multirotor (or the caged one)	No GPS and close to the aircraft: ALT_HOLD or FLOWHOLD.
Tank	Caged drone	It can touch the walls; any other one crashes when it does.
Solar plant, hotspots	Multirotor or VTOL	Thermal surveys: both work.

Missions next to an installation (wind turbines, hangar, solar plant, tank, tower, bridge) start next to it, as when you arrive by van: that point is their pad and the multirotor's home for RTL.

If you pick a poorly suited aircraft, the card warns you before starting and lets you try: an **airplane without VTOL** cannot stop in front of the wall or tower, so only passes work; an **FPV drone** flies in ANGLE or ACRO, without position or altitude hold and without AUTO, so it is a piloting challenge more than a realistic job.

28.3 The payload camera

It is a camera on a stabilised gimbal under the aircraft: it does not move with roll or pitch, only with heading and with what you command. It has two sensors.

Sensor	Field of view	Resolution	Zoom	Used for
RGB 4K	70°	4000 px wide	up to ×8	Detail: cracks, insulators, nests, vegetation
Thermal IR 640×512	45°	640 px wide	up to ×4	Heat: people and animals at night, hot joints, cold seepage

Key	Button	What it does
T	IR / RGB	Switches sensor
Z · X	+ · −	Zoom in and out
G · F	▲ · ▼	Tilts the camera up or down (from −90°, looking at the ground, to +20°)
Q · E	◀ · ▶	Pans the camera left or right of the nose
B	Mark (B)	Marks a finding at the centre of the crosshair
H	⤢	Large or small window

The top line of the window shows the sensor, zoom, tilt, the distance to the ground or tower along the camera axis (rangefinder) and the **GSD**: how many centimetres of ground fit in one pixel. Closer or more zoom means a smaller GSD and more detail.

In thermal, white is hot and black is cold. The ground warms up with the sun depending on which way it faces, and cools at night. People and animals stand out far more than in daytime.

28.4 Marking findings

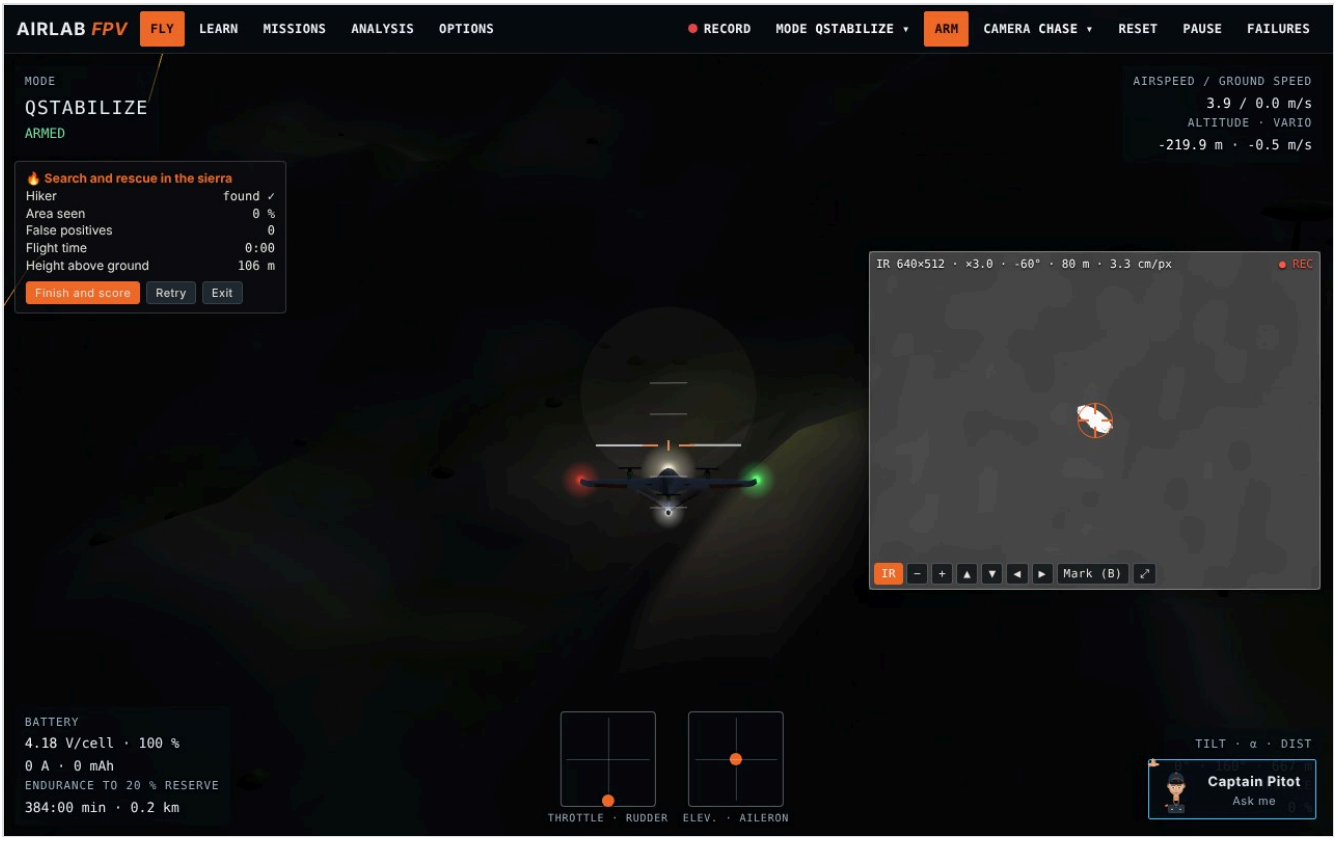
Put the target in the centre of the crosshair and press B. The simulator checks three things, as a reviewer would check your photo:

- **That it is visible**: in the centre of the image and not hidden by terrain.
- **The right sensor**: a hot joint only shows in thermal; a broken insulator, a nest or a crack only in RGB. If it is the wrong one, it tells you.
- **Enough detail**: each finding needs a maximum GSD. If you are short, it tells you what you have and what is needed: get closer or zoom in.

If it is a finding, it saves the photo with its WGS84 position, distance, GSD and sensor. If you mark something that is not (a goat instead of the hiker, a sound piece of wall), it counts as a **false positive** and takes points off.

In some missions, marking opens a small window to **classify** the finding: its severity (wind turbines), its type (solar plant) or whether it is within limits (aircraft). Choose with the mouse or keys 1-5; the score takes it into account.

28.5 Search and rescue in the sierra



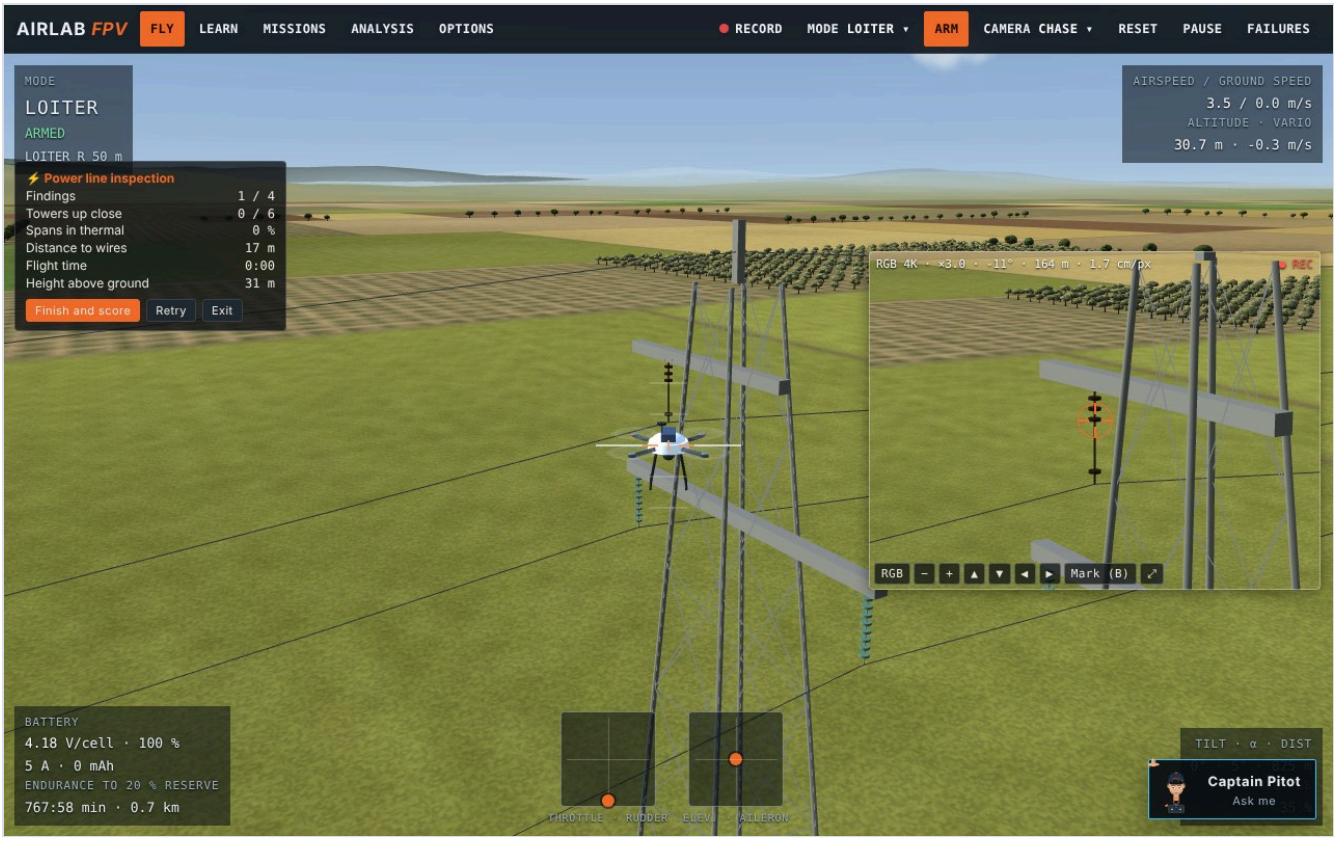
Night rescue: the hiker found with the thermal camera at 96 m and ×3 zoom (4 cm/px). Nothing shows in the normal view.

It is night (21:30). An injured hiker is on the south slope, between the col and the river; the search area is outlined on the ground in orange. Six animals (goats and boars) give off heat too.

- Fly parallel passes over the area with the camera tilted forward (−45° to −60°) in thermal. The slope drops about 500 m: watch the panel's **height above ground**, not the altitude.
- When you see a white spot, get closer or zoom in before marking. A person lying down is longer and narrower than a standing animal.
- Once marked, the position goes in the report for the ground team. Fly back with plenty of battery.

Scored	Points
Hiker found	40
Position accuracy (error of the marked point)	10
Time to find him (full marks under 5 min)	15
No false positives (−5 each)	10
Always below 120 m above ground	10
Return and landing on the pad with ≥ 20 % battery	15

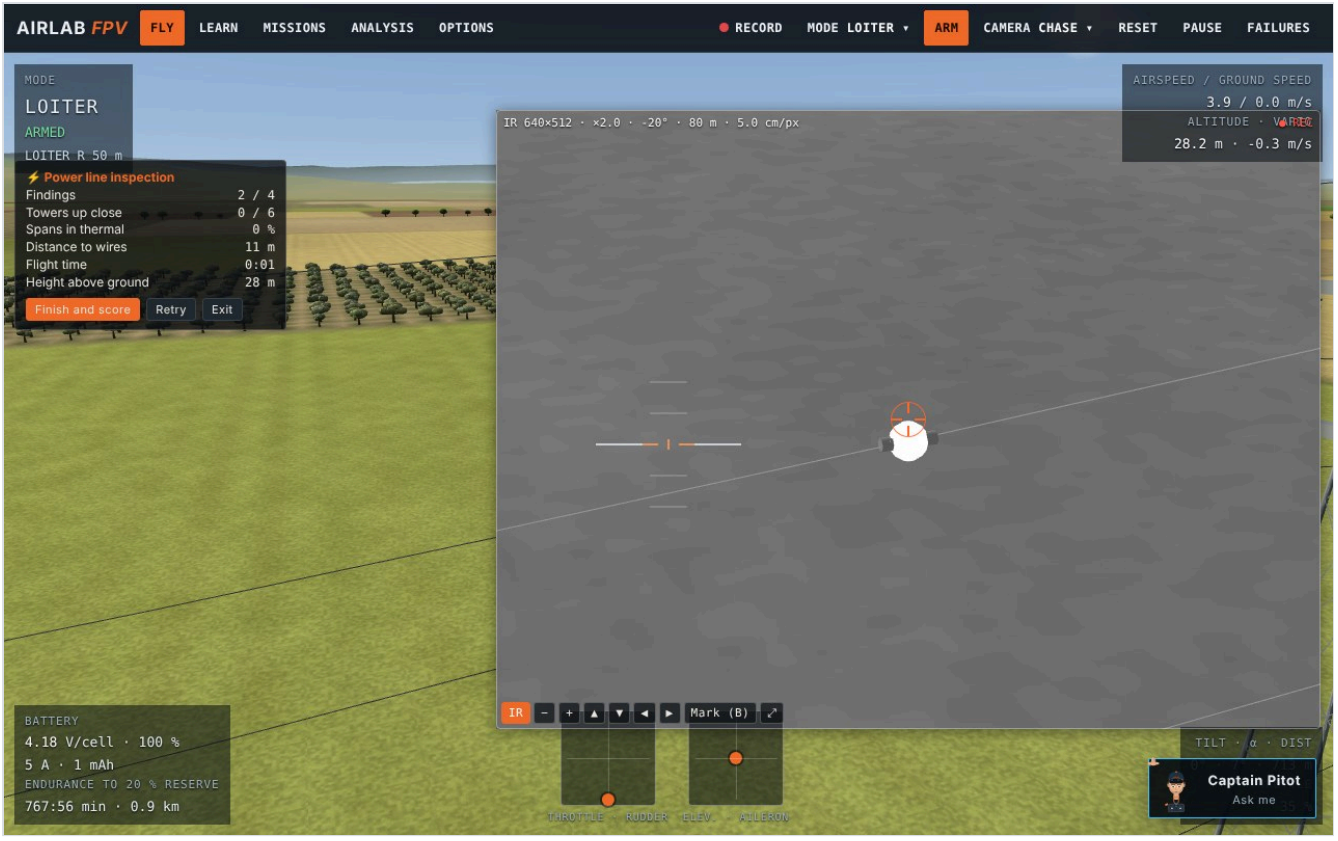
28.6 Power line inspection



Tower inspection in RGB at ×3 zoom: the insulator string and crossarm at 1.5 cm/px.

A 220 kV line with 6 towers and 5 spans, south of the airfield. There are 4 random defects from these:

Defect	Sensor	Maximum GSD
Hot joint (hot spot on a conductor)	Thermal	6 cm/px
Broken insulator (missing or blackened discs)	RGB	1.5 cm/px
Nest on the crossarm	RGB	2.5 cm/px
Tree too close to a conductor	RGB	8 cm/px

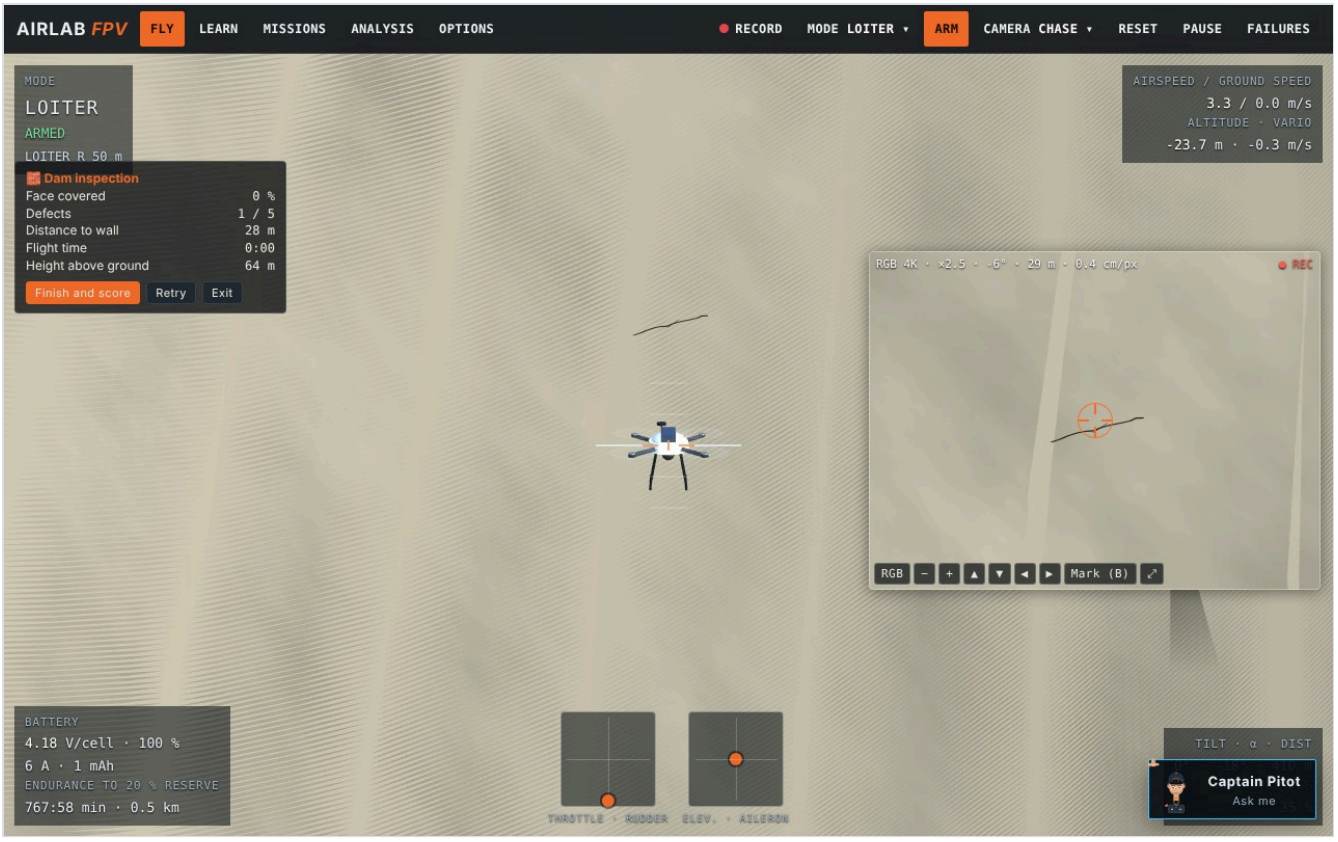


The same line in thermal with the large window (H): a hot joint on a conductor. The conductors show slightly warmer than the background.

- For each tower, get close in RGB and look at the head (insulators and crossarms) at 1.5 cm/px or less: the panel counts the towers inspected.
- Fly along the spans parallel to the line with the thermal camera, looking at the conductors at 8 cm/px or less.
- **Never closer than 10 m to the wires**: the panel shows the distance and warns below 10 m. Below 1.2 m you hit the conductor. This is this exercise's safety distance; in a real job the power company sets it.

Scored	Points
Defects found	40
Towers inspected up close (6)	15
Spans covered in thermal (90 % or more)	15
Safety distance (−3 each time below 10 m)	10
No false positives	5
Return and landing	15

28.7 Dam inspection



Downstream face with the camera level and ×2.5 zoom: a crack marked at 28 m.

The annual inspection of the downstream face (95 m high). There are 5 defects: cracks, seepage (it shows cold in thermal), spalling, vegetation in a joint and efflorescence (white salt stains).

- Set the camera level (0° tilt or slightly down) and fly horizontal passes parallel to the wall, top to bottom. **QLOITER** helps hold the distance in wind.
- The panel shows how much of the face you have covered at 2 cm/px or better; with the RGB camera that is about 55 m at most without zoom. The target is 85 %.
- Never closer than 3 m to the wall.

Scored	Points
Face covered (85 % or more)	30
Defects found	40
Distance to the wall (−4 each time below 3 m)	10
No false positives	5
Return and landing	15

28.8 Stockpile volume survey at the quarry



Quarry: the eastern stockpile with its photo coverage and two of the five control targets, in the nadir camera.

The quarry wants to know how many cubic metres the eastern stockpile holds. Nothing is marked here: you fly a photogrammetric survey (chapter 18) and the simulator computes the volume from the photos you took.

1. On start, the mission prepares the survey area (an octagon around the stockpile) and the RGB camera at **2.5 cm/px** with **75 / 70 %** overlap.
2. **Plan › Photogrammetry and remote sensing › Add survey with triggering**. Check the passes on the planner map.
3. Take off, select **AUTO** and let it fly the survey. The camera triggers by itself.
4. After landing, **Finish and score**.

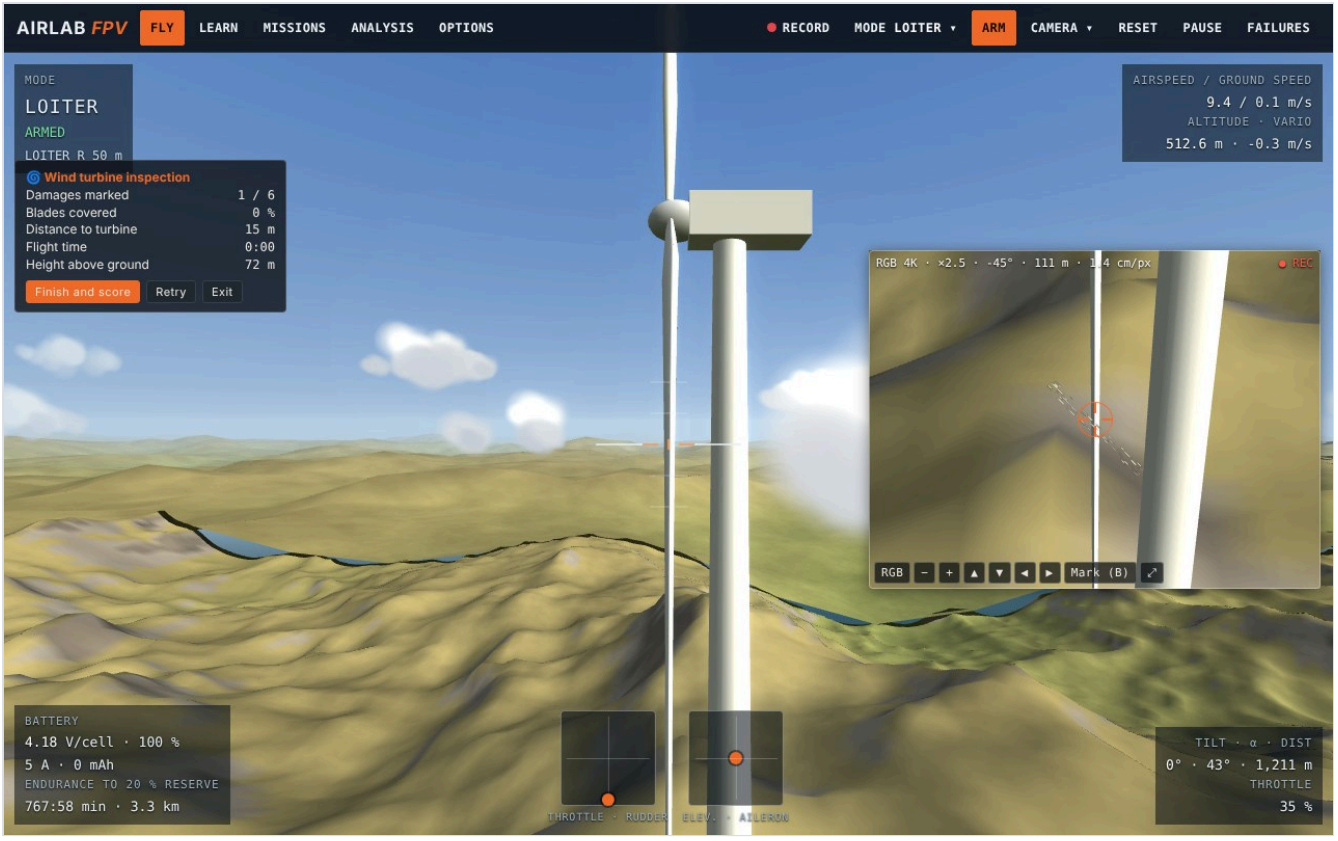
Around the stockpile there are 5 targets (black and white squares): the **ground control points** (GCPs). Each one has to appear in at least 2 photos, and 3 are needed for good georeferencing. Without them, the model carries a height and scale error that cannot be corrected.

How it is computed. Where there are 2 or more valid photos, the surface is reconstructed (the true height plus an error that grows with GSD). Where there are no photos a gap is left and filled with the base plane, so the volume comes out short. The base plane is the mean level around the foot of the stockpile. The result is compared with the real volume of the terrain model.

Scored	Points
Volume error (full marks under 1 %, none from 8 %)	40
Stockpile covered by 2 or more photos (98 % or more)	20

Overlaps: 70 % forward and 60 % side or more	15
GSD of 3 cm/px or better	10
Control points in the photos (5 each, up to 3)	15

28.9 Wind turbine inspection



A blade flown with the multirotor at about 15 m and ×2.5 zoom: a damage mark in the centre of the camera.

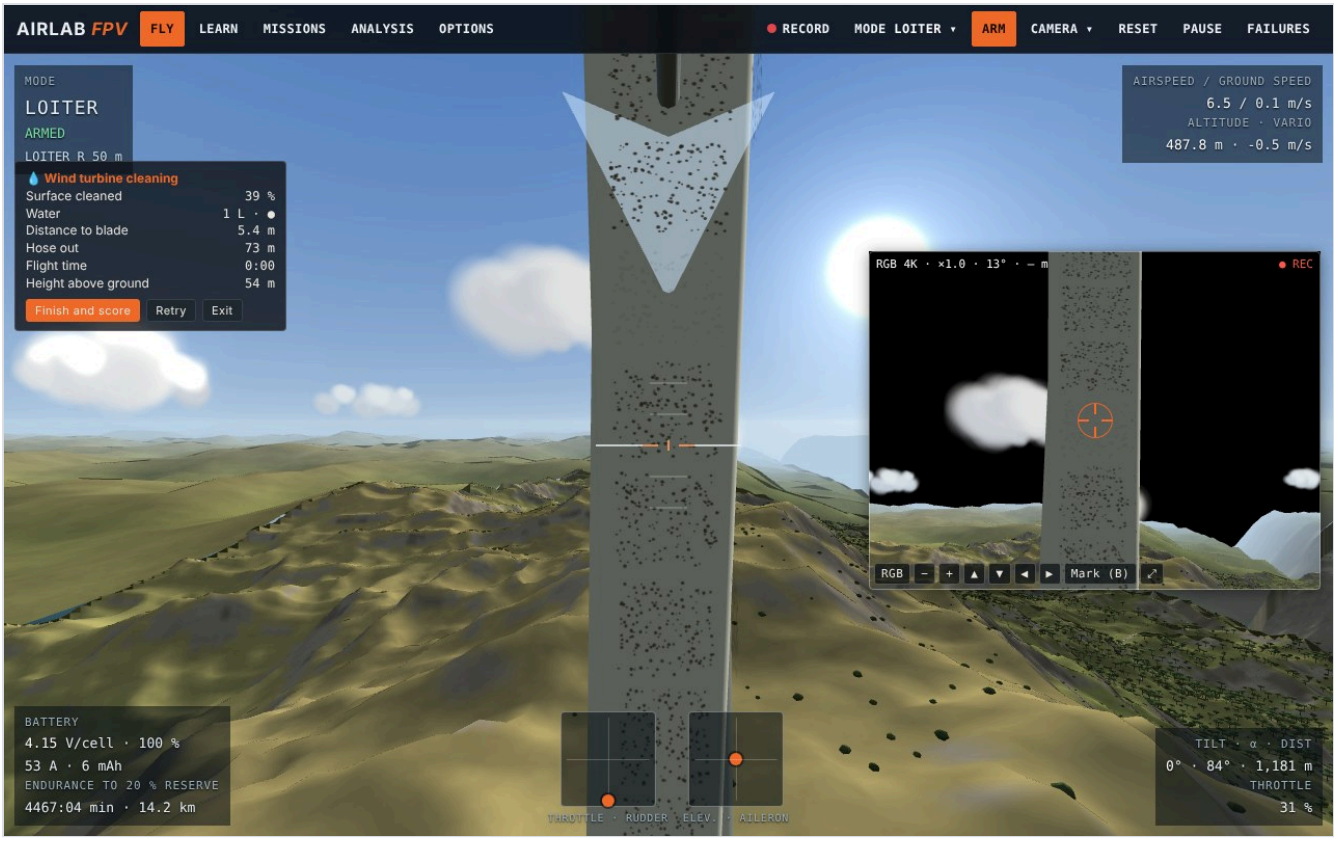
Three wind turbines on the sierra ridge, rotors braked: 80 m tower, hub at 82 m and 44 m blades. You take off next to the middle turbine, with a 7 m/s westerly wind.

- Fly each blade from root to tip along its four faces: **leading edge**, **trailing edge**, **pressure side** (the face towards the wind) and **suction side**, at 0.6 cm/px or better and more than 10 m away.
- There are 6 random damages: leading-edge erosion, laminate crack, delamination, lightning strike near the tip, bird or hail impact, trailing-edge split and coating wear.
- When you mark a damage you choose its **severity** on a 5-level training scale (keys 1-5 also work): 1 cosmetic, 2 minor (monitor), 3 moderate (repair at the next stop), 4 serious (repair soon) and 5 critical (stop the turbine). One level off counts as half.
- Closer than 10 m to the turbine there is a warning; closer than 0.9 m is a crash.

Scored	Points
Damages found	35
Severity correctly rated	15
Blades covered (80 % of the four faces)	25

Distance to the turbine (≥ 10 m; −2 each time)	10
Return and landing	15

28.10 Wind turbine cleaning with a tethered drone



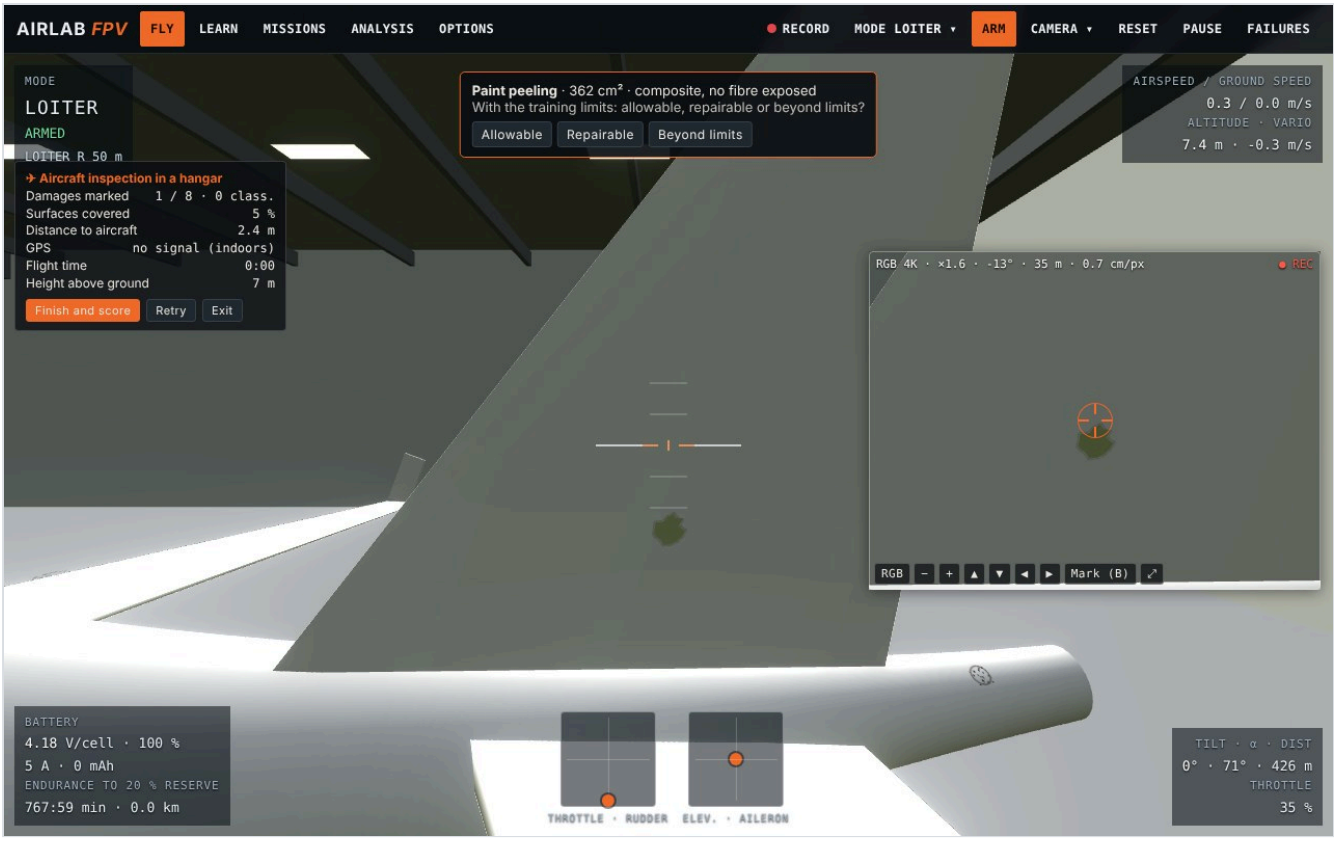
The tethered drone washing the blade that points at the ground. The hose runs down to the truck; the areas already cleaned are free of dirt.

The blade pointing at the ground has its leading edge and faces covered in insects and dust. The **tethered cleaning drone** (ch. 8) gets power and water through the hose from the truck, which carries a 1,000 L tank.

- The lance is on the gimbal: aim with the camera crosshair and turn the jet on or off with B. The jet (25 L/min) cleans from 2 to 11 m, better the closer you are, and it pushes you backwards.
- The full hose weighs 0.35 kg/m and the wind drags it: the drone leans to hold position. You cannot go further than 150 m from the truck.
- Never closer than 3 m to the blade. If the hose rubs the tower or the blades, it counts against you.

Scored	Points
Surface cleaned (target 90 %)	45
Water used (no penalty up to 450 L)	15
Distance to the blade (≥ 3 m; −3 each time)	15
Hose clear (−1 per 2 s rubbing)	10
Return and landing	15

28.11 Aircraft inspection in a hangar, with damage chart



Inside the hangar, with no GPS: when you mark a damage, the scanner gives its measurements and you classify it.

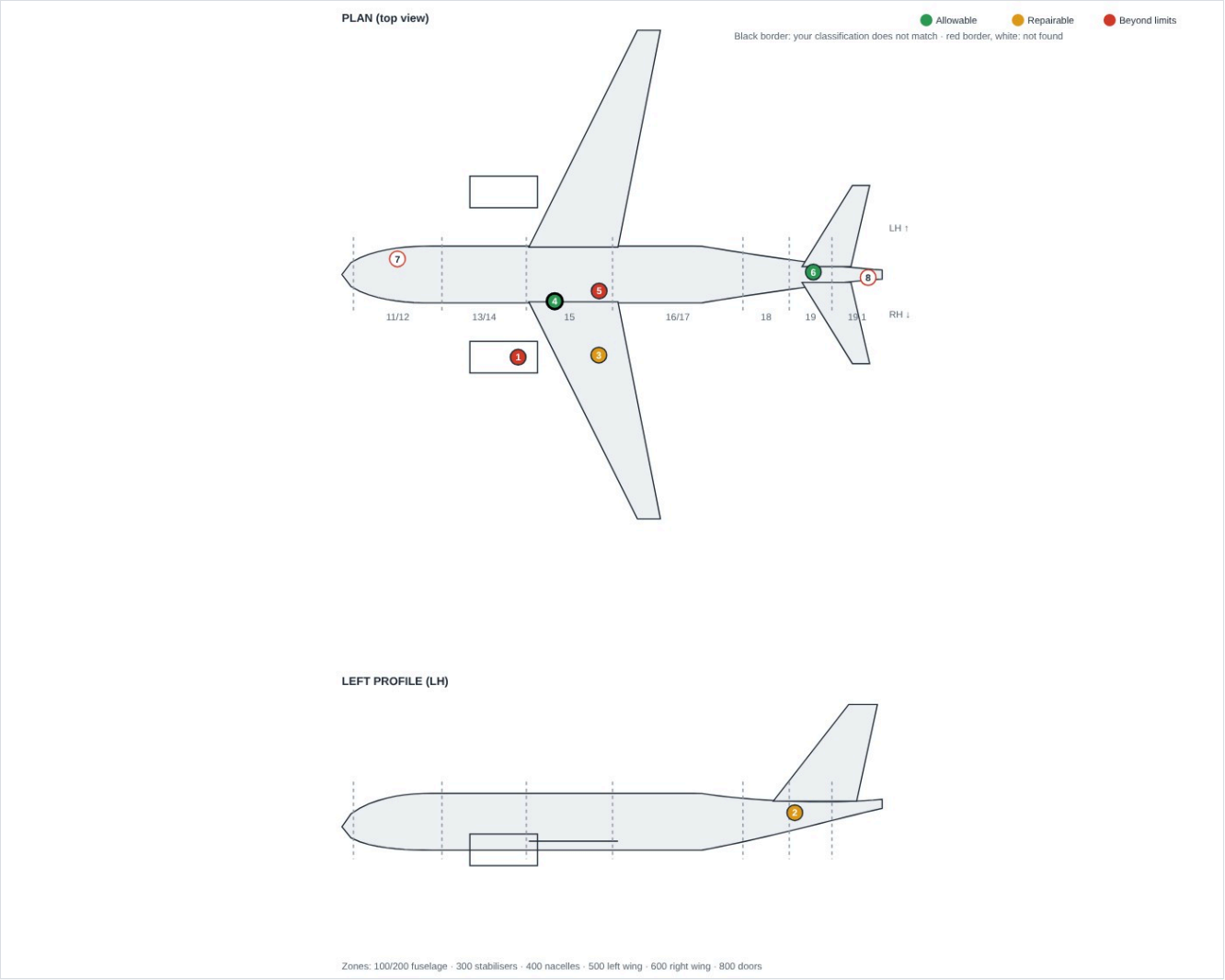
General visual inspection after a storm of a generic single-aisle twinjet with A320-type dimensions: 37.57 m long, 3.95 m fuselage width, 11.76 m high and about 34 m span. It is not any real aircraft and carries no markings.

- **No GPS**: inside the hangar the multirotor has no position. Arm and fly in **ALT_HOLD** or **FLOWHOLD** (ch. 8). LOITER, RTL and AUTO are not available.
- Go over the fuselage, wings, stabilisers and nacelles at 0.3 cm/px or better (0.15 for cracks and rivets), never closer than 2 m to the aircraft. The walls, the roof and the aircraft are crashes.
- There are 8 random damages. When you mark one, the scanner gives its measurements and you classify it as **allowable**, **repairable** or **beyond limits** with these training limits:

Damage	Allowable	Repairable	Beyond limits
Dent	≤ 1.5 mm deep and Ø ≤ 50 mm	≤ 4 mm	> 4 mm. Over a frame, one level up.
Lightning strike	Burnt paint only	Pits ≤ 0.4 mm	Pits > 0.4 mm
Crack	—	In secondary structure (fairing) ≤ 25 mm	In primary structure, or > 25 mm
Paint peeling	On metal, or composite with no fibre exposed	Composite with fibre exposed	—
Missing rivets	—	1	2 or more in a row

Hail	Max depth ≤ 0.8 mm	≤ 2 mm	> 2 mm
Illegible placard	—	Always (it is replaced)	—

> **Training limits**
>
> Real damage limits are in the manufacturer's SRM (Structural Repair Manual), which is not public. These are made up for practice and the report says so.



The report's damage chart: plan and left profile with the fuselage sections, and each damage numbered in the colour of your classification (black border if it does not match the correct one).

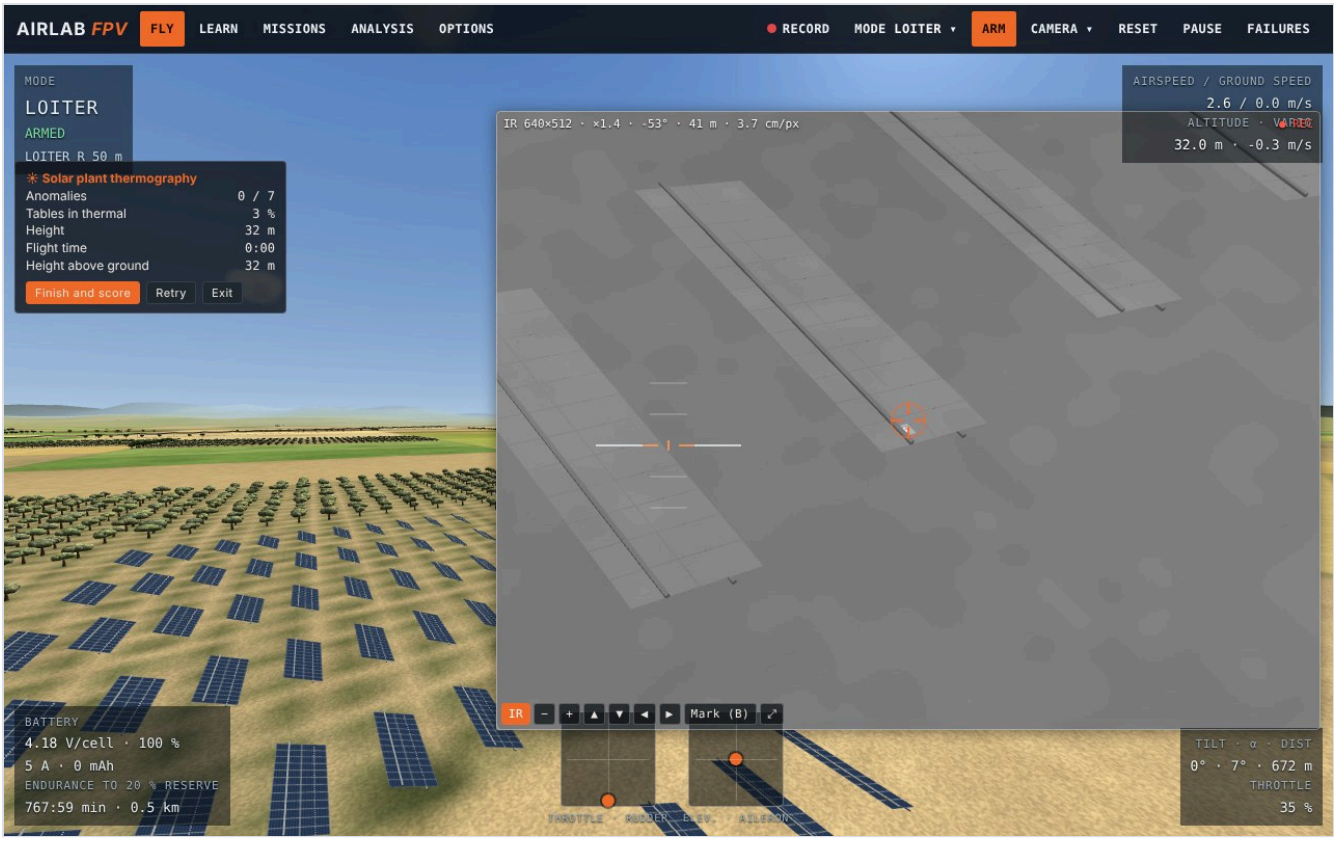
Each damage is located as in the maintenance documentation (ATA 06):

- **Major zone**: 100 and 200 (lower and upper fuselage halves), 300 stabilisers, 400 nacelles, 500 and 600 (left and right wings), 800 doors (831, 841…).
- Fuselage **section**: 11/12 to 19.1.
- **Station** (STA, in mm from STA 0, which is 2540 mm ahead of the radome) and **frame** (FR).
- **Side**: LH or RH.

The damage chart is in the result and in the PDF, with the table of damages, their measurements, your classification and the correct one.

Scored	Points
Damages found	35
Classification against the limits	25
Surfaces covered (target 75 %)	15
Distance to the aircraft (≥ 2 m)	10
Return and landing at the start point	15

28.12 Solar plant thermography



A table of modules in thermal at about 30 m: a disconnected string shows as a warm, uniform row.

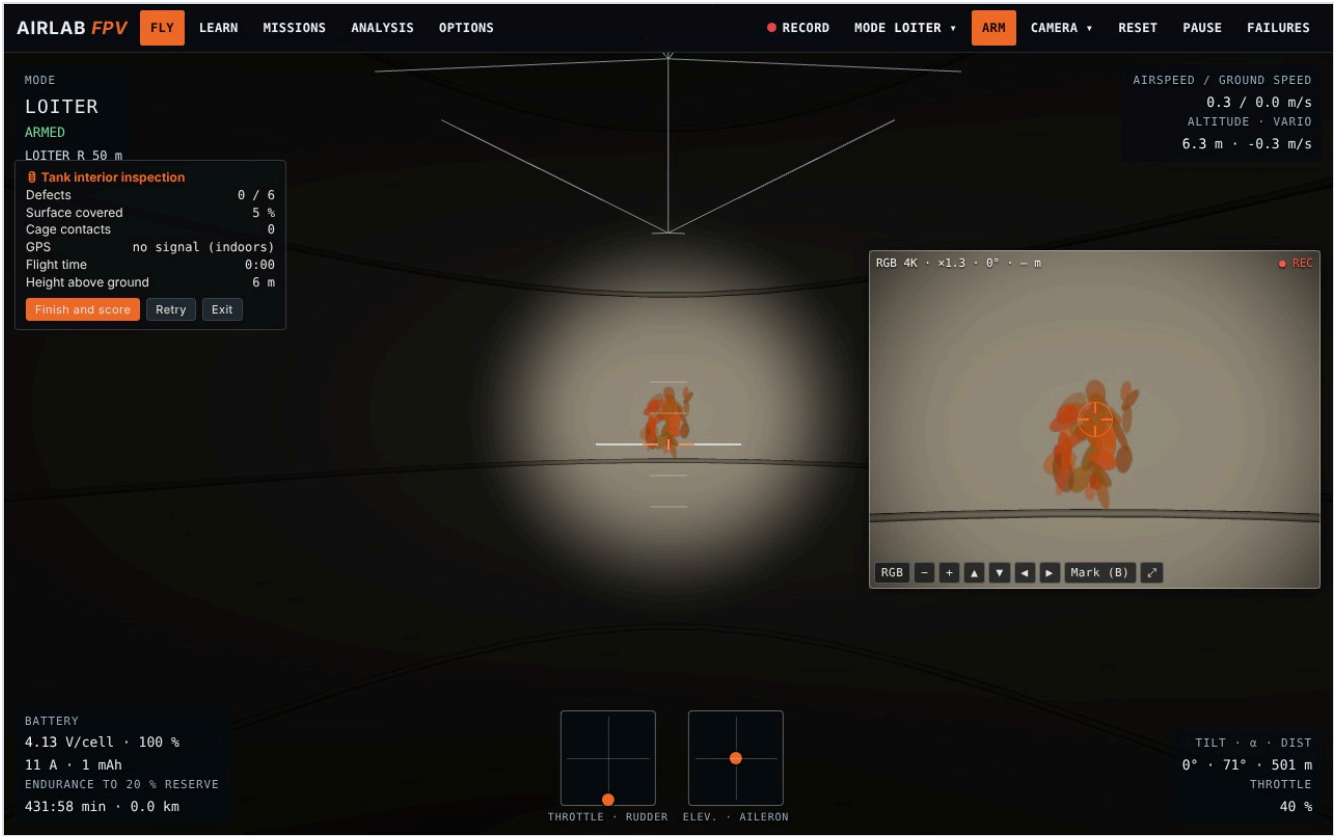
A 1,920-module plant: 10 rows of 8 tables of 2 × 12 modules, tilted 25° to the south. The survey is at midday under clear sky, because the modules have to be working with good irradiance; as a reference, at least about 600 W/m² is usually required.

- Fly over the rows with the thermal camera at 5 cm/px or better, about 30-35 m high with the camera between −60° and −90°. It can be done in AUTO with a survey (ch. 18).
- There are 7 anomalies. When you mark one, you say what type it is:
  - **Hot cell**: a very hot spot.
  - **Bypass diode**: a third of the module warm.
  - **Disconnected string**: the 12 modules of a row warm and uniform.

- **Bird dropping**: a hot spot that also shows in RGB.
- **Broken glass**: it shows in RGB and is hot.

Scored	Points
Anomalies found	35
Anomaly type	20
Plant covered in thermal (target 90 %)	25
No false positives	5
Return and landing	15

28.13 Tank interior



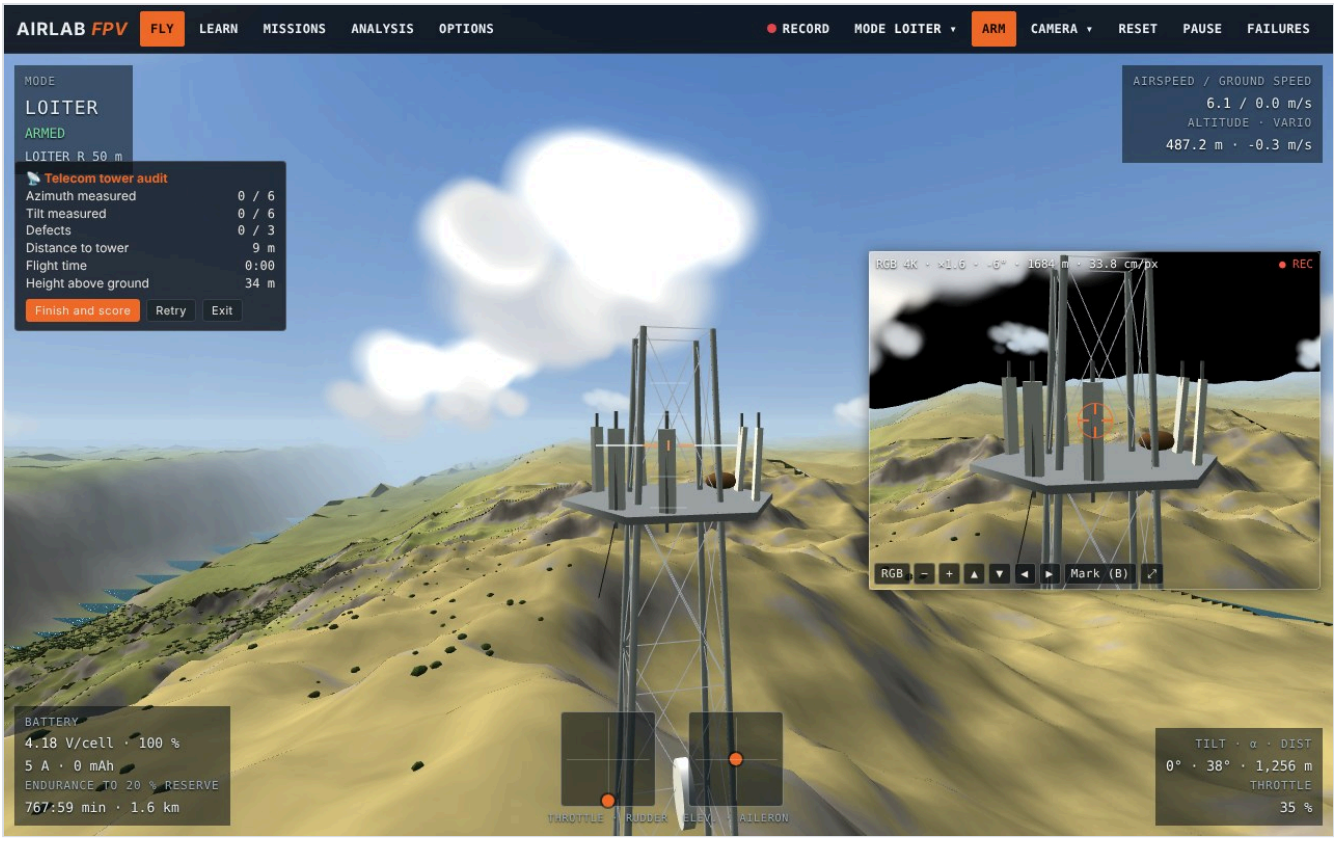
Inside the tank with the caged drone: the lights are the only light.

An empty tank, 20 m in diameter and 14 m high. The inspection is done without anyone entering. The **caged drone** (ch. 8) starts inside, on the floor, and flies with its own camera (FPV view).

- It is dark: the only light is your lights. There is no GPS: ALT_HOLD or FLOWHOLD, and optical flow is so-so with little light.
- Look for corrosion on the plates and the roof, pitting, a weld crack, coating blisters and sediment on the floor, at 0.5 cm/px or better (0.2 for the crack).
- With the cage you can touch the wall and you bounce off; a hit at more than 3.5 m/s is a crash. With another drone, touching it is a crash.

Scored	Points
Defects found	40
Interior surface covered (target 70 %)	30
Wall contacts (−1 each)	15
Return and landing at the start point	15

28.14 Telecom tower audit



A 40 m lattice tower with three sectors of two antennas: measuring an antenna from the front.

The operator wants to check the orientation of its 6 antennas (3 sectors) and the state of the tower.

- To measure an antenna, put it in the crosshair and press B at 1.5 cm/px or better:
  - **from the front** (±15°) for **azimuth**;
  - **side-on** (±15°) for **tilt**.
- The report compares each antenna with the design sheet. More than ±5° of azimuth or ±2° of tilt is “out of tolerance”.
- Also look for rust on a bracket, a loose jumper cable and a nest.
- Never closer than 3 m to the tower, and never within 5 m in front of an antenna. It is a training exposure zone: the real one depends on each antenna’s power.

Scored	Points
Antennas measured (azimuth and tilt)	35

Misaligned antennas detected	10
Tower defects	25
Safety	15
Return and landing	15

28.15 Bridge inspection



Under the deck of the valley bridge: with no GPS, the multirotor is in ALT_HOLD.

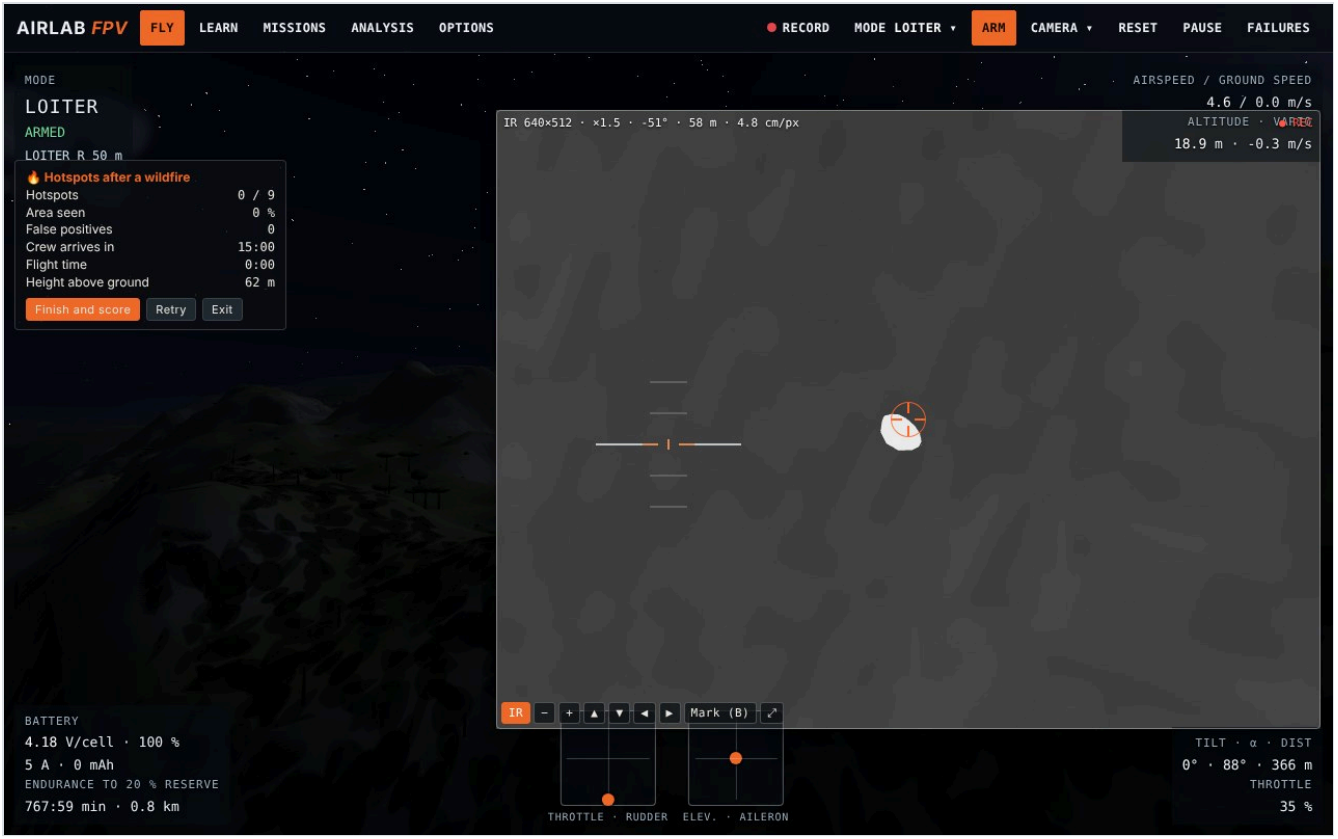
A three-span concrete bridge (25 + 40 + 25 m) over the valley river, with a 10 m deck on three girders and two-column piers.

- Under the deck the GPS is lost and the multirotor drops to **ALT_HOLD** (ch. 8). You will have to hold it yourself or use FLOWHOLD; over water, optical flow barely works.
- There are 6 random defects on the slab, girders, piers, bearings and joints: cracks, spalls with exposed rebar, efflorescence, a damaged bearing and joint leaks. They have to be seen at 0.5 cm/px or better.
- Never closer than 2 m to the structure. The deck, girders and piers are crashes.

Scored	Points
Defects found	40
Deck underside and piers (target 70 %)	25
Distance to the structure (≥ 2 m)	15

Return and landing	20

28.16 Hotspots after a wildfire



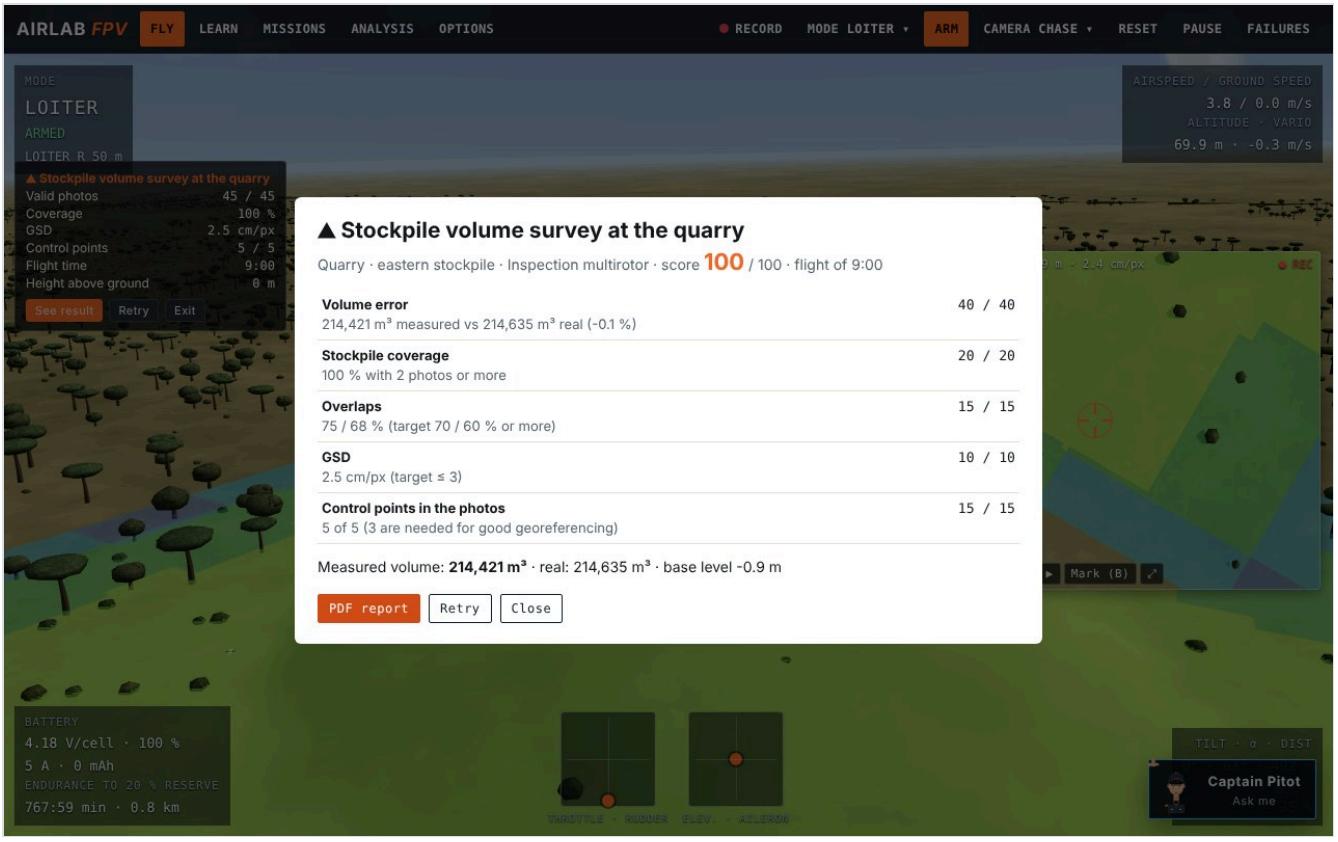
At night over the burnt slope: a hotspot in thermal.

At night (22:00), over a burnt area of about 520 × 360 m on the south slope. The fire is under control but could flare up again, and the crew arrives in 15 minutes.

- Fly over the area with the thermal camera at 15 cm/px or better and mark the 9 hotspots (embers under stumps and roots).
- Rocks keep the heat of the day and show warm, not hot: marking them is a false positive, as is the boar.
- Stay below 120 m above the ground.

Scored	Points
Hotspots marked	45
Burnt area seen in thermal (target 90 %)	20
False positives (−3 each)	10
Before the crew arrives	10
Legal height	5
Return and landing	10

28.17 Result and PDF report



Volume survey result: each item with its points and the measured volume against the real one.

Finish and score shows the score out of 100 with each item, the findings with their photo, coordinates, distance, GSD and sensor (and those not found) and, at the quarry, the volume. **PDF report** downloads it with the brief, aircraft, camera, flight time, results, the photo of each finding and the false positives. Each mission's best score is kept in the browser and shown on its card.

> **For schools and companies**
>
> Since the targets change on every attempt, it works for assessment: students cannot learn where they are. The PDF report is the record of the exercise. The data are simulated and the report says so on every page.

Sources: 120 m maximum height above ground in the open category, Commission Implementing Regulation (EU) 2019/947 (EUR-Lex). Volume, GSD and overlaps: chapter 18. UAV blade inspection, more than 10 m away and below 2 m/s: arXiv 2507.04922. Blade damage types: Fluix. Tethered drone cleaning (cable, hose, truck with generator and pump): The Drive. Drone aircraft inspection (AMM task for the A320 family; dent & buckle charts): AviTrader, Mainblades. A320 dimensions, zones, sections and stations: Lufthansa Technical Training A319/320/321 ATA 06-10 training manual. PV thermography (typical irradiance ≥ 600 W/m²): Sinovoltaics. Caged drones in tanks: RUAS. FS_EKF_ACTION and FLOWHOLD: ArduCopter code (`ArduCopter/Parameters.cpp` , `mode_flowhold.cpp`).

29 Pilot logbook, level and certificates

The simulator keeps your **flight log** without you having to write anything down. Every flight, from arming to disarming, is recorded in **Analysis › Pilot logbook**. You can also get there from the **Logbook** button on the home screen, which shows your level and hours.



Analysis › Pilot logbook: level and what is left for the next one, totals, hours per week, per aircraft and per mode, courses and flight list.

29.1 What is logged for each flight

- **Date and time**, **aircraft**, **scenario** and whether it was at night.
- **Airborne time** and armed time, and the **time in each mode** (QHOVER, FBWA, AUTO...).

- **Landings** (a touchdown counts if you had been airborne for at least 3 s; it is *soft* if you were descending at 1.5 m/s or less with less than 20° of tilt), **transitions** to airplane and back to VTOL, and whether it ended in a **crash**.
- Maximum height, distance flown, the **debrief score** and the **activity**: lesson completed, challenge with its score, work mission or custom AI mission.
- The **source**: simulator, ArduPilot SITL, **real aircraft** (with «Real flight with instructor», chapter 16; it also stores the takeoff coordinates) or **real, logged by hand**.

Flights with less than 3 s in the air (arming and disarming on the ground) are not logged. Replays are not logged either.

29.2 Simulator hours and real hours

Simulator (and SITL) hours and real hours are always kept **separate**: in the totals, in the weekly chart (blue and orange) and in the exports. For a flight of your aircraft without telemetry, press **+ Real flight by hand** and fill in date, time, airborne time, aircraft, place, landings, transitions, modes and notes. Only manually logged flights can be deleted; the others are logged by the simulator and cannot be edited.

CSV downloads every flight for Excel (semicolon separated). **PDF** downloads the full logbook on landscape A4, with the totals and space for the pilot's and the instructor's signatures.

> **Is it an official logbook?**
>
> It is a very complete personal record and helps you keep yours, but it does not replace any records the regulations may require from an operator.

29.3 Pilot level

Your level does not go up with hours alone: you also have to show skills. Under your level you can see what is left for the next one.

Level	What it takes
0 · Student	To get started.
1 · Pilot in training	30 min of flight, 5 soft landings and the «First hover» lesson.
2 · VTOL pilot	The «VTOL fundamentals» course complete.
3 · Advanced pilot	10 h of flight, 50 transitions, the four challenges at 80 or more and the «AUTO missions» course.
4 · Professional pilot	25 h of flight and the «Drone inspection» and «Photogrammetry and remote sensing» courses.

Hours for levels and courses include the simulator, SITL and real flights with telemetry; manually logged flights do not count. **Badges** hold fourteen more achievements: first transition, landing of 95 or more, night flight, radio loss handled, first AUTO or work mission, flight with SITL, real flight, 100 landings, 20 flights in a row without a crash...

29.4 Courses and certificates

Course	Requirements
VTOL fundamentals	The five «Learn» lessons; «Landing from hover» and «Timed transition» challenges at 70 or more; 1 h with VTOL aircraft; 10 transitions; 10 soft landings.
AUTO missions with ArduPilot	5 flights with at least 1 min in AUTO finished without a crash; «Return home with QRTL» and «Radio loss» lessons; 2 h of flight.
Drone inspection	6 different work missions at 70 or more; 2 h of flight.
Photogrammetry and remote sensing	The five remote sensing exercises at 70 or more; 1 h of flight.
EASA/AESA drone regulations	A1/A3 and A2 mock exams at 75% or more; «120 m ceiling» and «A2: distance to people» practices and the UAS zones exercise at 70 or more (chapter 36).
Drone-in-a-box operator	Three different operations centre services at 70 or more; one at level 4 or 5 at 70 or more; six contingencies handled correctly; 1 h of flight (chapter 39).

When a course is complete, **Get the certificate (PDF)** becomes active. The first time it asks for **your first and last name**, exactly as you want them shown; it is saved in your account and used on every certificate (change it with **Change** in «Courses and certificates»).



Certificate for the «VTOL fundamentals» course: requirements met, flight record, date, verification code and QR.

The certificate carries a **verification code** (like AL-7K2Q-M9XD) and a QR. Anyone can check it at **www.airlabfpv.es/verificar.html**: it shows the name, course, date and flight record. The hours, flights,

landings and transitions on the certificate are computed by the server from the flights stored in your account. If you download it again, it keeps the same code.

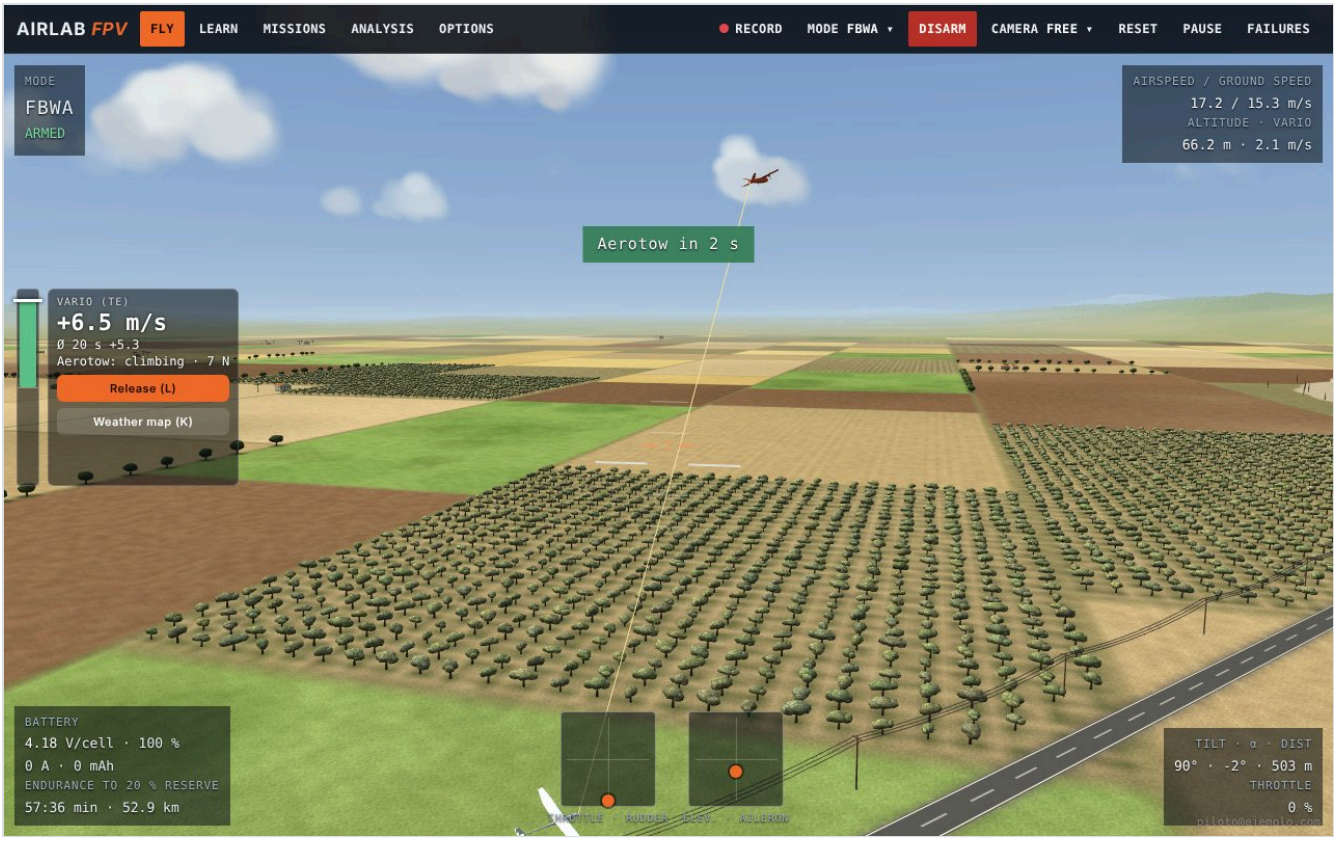
> **Not an official qualification**
> It is a simulator training certificate and says so. It is not an AESA or EASA qualification (A1/A3, A2, STS) and does not authorise you to operate unmanned aircraft.

29.5 Where it is stored

With an account, the logbook, your name, your progress (lessons, challenges, missions and exercises) and your certificates are saved in your account: sign in on another computer and it is all there. If a flight could not be uploaded it shows an ↑ arrow and is uploaded automatically after the next flight. In the offline version everything is stored in the browser; export the CSV now and then to keep a copy. Certificates issued by the offline version have no verification code.

30 Sailplane: aerotow, thermals, slope and variometer

The **2.5 m sailplane** is a radio-controlled glider with no motor: 51 dm², aspect ratio 12.3, SD7037 airfoil, cruciform tail and 1.45 kg (30 g/dm²). It stalls at about 6 m/s and glides about 18 times its height at 9 m/s. It is a generic design, not a commercial model: its aerodynamics come from AeroBuildup and NeuralFoil, like the other aircraft, but without a manufacturer's CFD to calibrate it.



Aerotow: the sailplane behind the tug, with the rope, and on the left the variometer with the tow status and the Release button.

30.1 Taking off on tow

1. Choose the sailplane on the home screen. It appears on the runway with the tug (an orange twin) 20 m ahead and the rope slack.
2. **Arm** (space bar or the radio switch). After 2 s the tug takes up the slack and rolls.
3. Keep the wings level. A helper holds the wingtip up to 5 m/s. At about 10–12 m/s pull a little elevator: the sailplane lifts off before the tug.
4. During the climb (12.5 m/s, about 2 m/s climb) the tug flies a circuit: a straight leg and a 180° turn at 15° of bank. Follow it with aileron and rudder, right behind and just above its wake.
5. At the **release height** (200 m by default) you hear «Release height». Press L or the **Release** button. If you do not release within 45 s, the tug releases you.

> **Just like the real thing**
>
> If you get too high (more than 25° above), too far to the side (more than 45°) or overtake, the tug releases for safety: a high sailplane lifts its tail. If the rope is jerked above 38 N, it breaks. The rope force is shown on the indicator.

In the weather map you can switch the launch to **by hand** (for slopes): when you arm, the sailplane is launched after 2 s.

It releases you next to a thermal. If there are thermals, the tug does not fly at random: it heads for the best nearby thermal, circles it and, on reaching release height next to it, releases you by itself. Captain Pitot tells you where it is («150 m away, at 10 o'clock») so you can turn towards it. With no thermals (at night, overcast or thermals off) it flies the usual circuit. You can release earlier with L.

30.2 Modes

With no motor, the sailplane flies in **MANUAL**, **FBWA** and **ACRO**. The modes that need throttle (FBWB, CRUISE, LOITER, AUTO…) cannot be selected; failsafe modes still act.

With the motor glider's motor out and charged (30.7), Captain Pitot may tell you to add throttle; with the motor retracted, his advice is for a glider: lower the nose to gain speed, never «add throttle».

30.3 Variometer

The indicator on the left is a **total energy** variometer: it measures how much the sailplane climbs or sinks counting its speed too. So it does not beep when you pull the elevator and trade speed for height; only when the air really rises. Below it is the average of the last 20 s.

- **Climbing**: beeps, higher and faster the more you climb.
- **Between 0 and −0.8 m/s** (normal gliding sink): silence.
- **Sinking more**: a continuous low tone, lower the faster you sink. The threshold can be changed in the map (−0.5 to −2 m/s).

30.4 Thermals

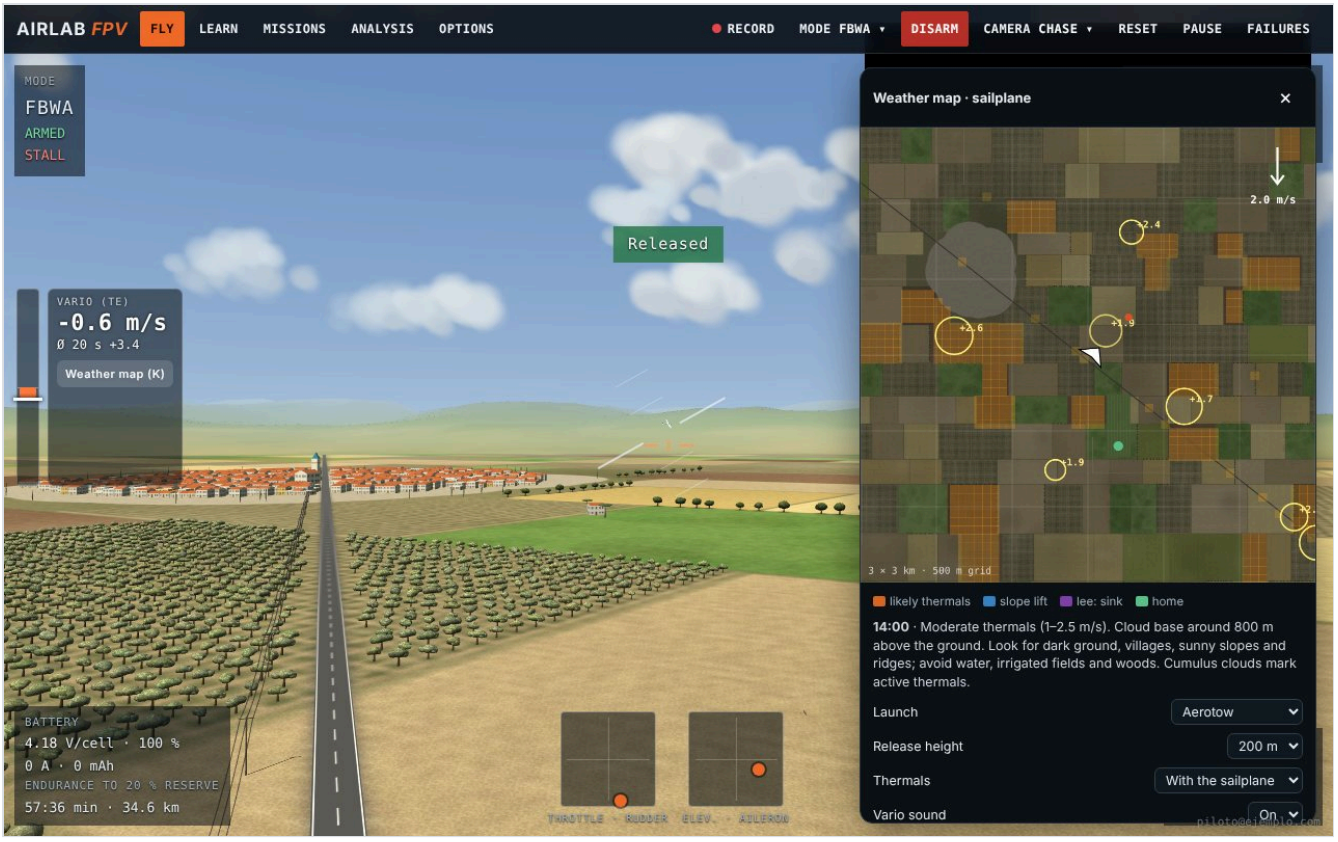
Thermals are bubbles and columns of warm air rising from the ground when the sun heats it. In the simulator they are born more often where the ground heats up most:

- **Sun**: with the sun below 8° there are none. They are strongest at midday and in the afternoon, weak with clouds and absent when overcast, foggy or raining.
- **Ground**: dark soil, villages and stubble heat up more than irrigated green fields, woods or water.
- **Relief**: slopes facing the sun, and ridges and hills, which is where the bubbles break away.

Each thermal has a core of 1 to 4 m/s and a ring of sinking air around it. It lasts 6 to 15 minutes, drifts and leans with the wind and reaches the cloud base (350 to 1,200 m above the ground, depending on the day), where a **cumulus** forms. Between thermals the air sinks a little. To use one, turn as soon as the vario beeps and shift the circle towards where it climbs most.

Each thermal core has a radius of 85 to 160 m, with the strongest lift in the middle: there is room to circle inside without falling out. **In the 3D flight** (map option, on by default) each thermal shows as a translucent column the size of its core: yellow when weak, orange and red the more it lifts, leaning with the wind and

with bands rising slowly; on the ground, a ring marks where it starts. To fly as in reality, turn it off and use the cumulus and the vario.



Weather map (K key): likely thermal areas in orange, slope lift in blue and purple, wind, home and sailplane settings; with «training» the actual thermals are shown.

30.5 Weather map (K key)

It shows 3 × 3 km around the sailplane:

- In **orange**, the areas where thermals are most likely with the sun and ground at that time.
- In **blue**, slopes facing the wind, which give lift. In **purple**, lee slopes, with sink and turbulence.
- The wind arrow, home, the tug and a forecast: thermal strength and cloud base.

In the same panel you choose the launch, the release height, whether there are thermals (only with the sailplane, with every aircraft or never), the vario sound and its threshold. There is also **Show the actual thermals**, for training: it draws each thermal with its strength at that moment.

There are also **Show the thermals in the 3D flight**, the **Motor glider motor** (yes or no) and the **Motor battery** (2, 3 or 5 minutes at full power).

30.6 Slope

With wind blowing against a slope the air rises along it, and the lift is strongest close to the slope and in front of the ridge. In the scenarios with relief (sierra, coast, dam, quarry) and on the real terrain of your area, fly figure-eights in front of the windward face. Do not go to the lee side: the air sinks there and there are rotors.

Sources: SD7037 airfoil from the UIUC database (via aerosandbox) · ArduPilot · Soaring · sailplane speeds and polar computed with `tools/aeronaves/generar.py`.

30.7 Motor glider: retractable motor

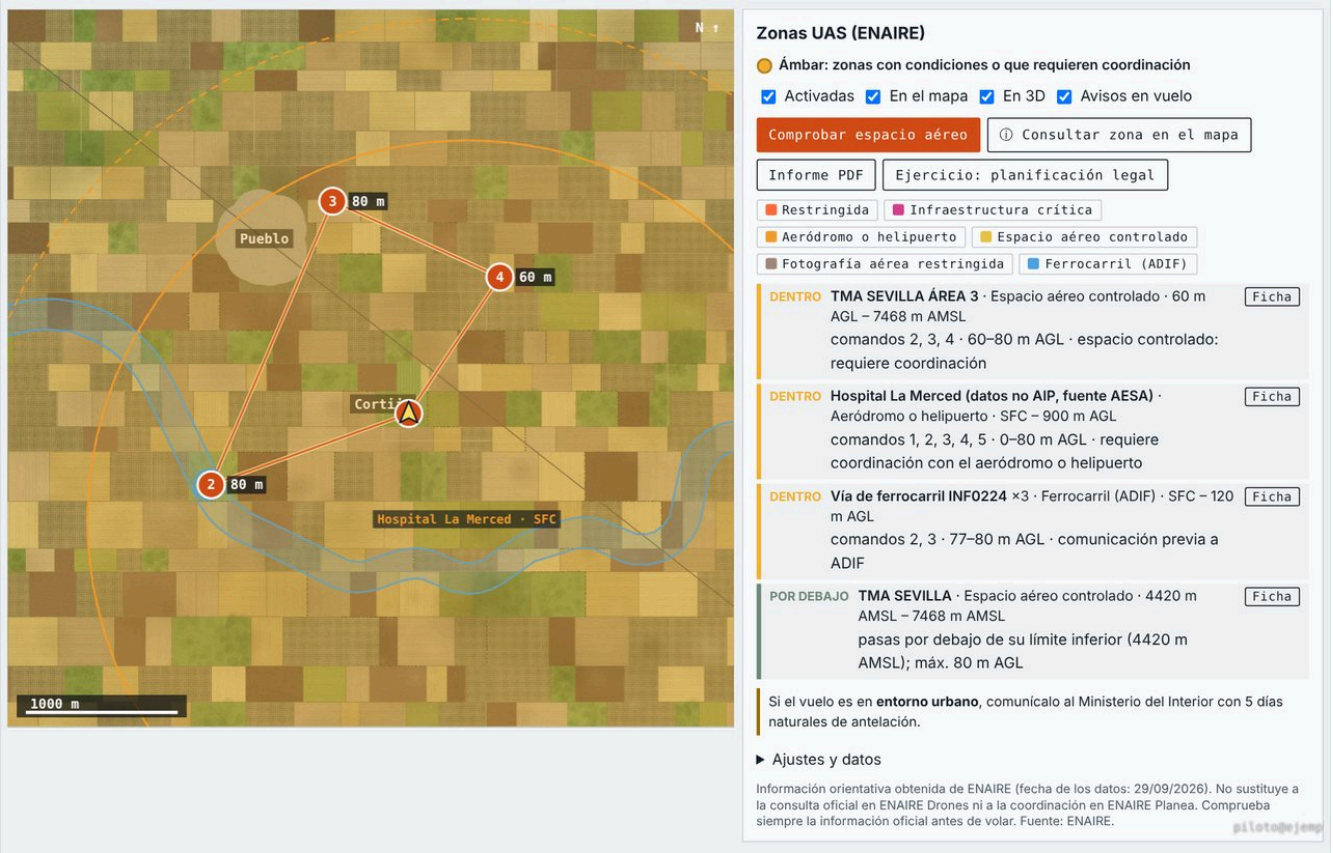
Like many sailplanes, the glider carries an electric motor on a mast that comes out of its back, behind the wing. With the `J` key or the **Deploy motor (J)** button in the vario box, the mast comes out in 1.5 s and the throttle makes it climb (about 2–3 m/s). `J` again retracts it.

- **Motor battery**: 2, 3 or 5 minutes at full power; it drains with the throttle. The box shows what is left (bars and time). Captain Pitot warns at 25 % and when it runs out. Every reset, a full battery.
- With the motor out and stopped, the propeller causes a lot of drag: if you are not using it, retract it.
- Launch **With the motor (motor glider)** in the map: self-launch from the strip; the mast comes out when you arm.
- Use it as in reality: to climb when you cannot find thermals or to get home if you are low and far away.

31 ENAIRE UAS zones: legal planning and in-flight warnings

The simulator knows Spain's **official UAS geographical zones**, the same ones shown by the ENAIRE Drones app: aerodromes and heliports, controlled airspace, prohibited, restricted and reserved areas, critical infrastructure, railway lines and urban environments. You see them on the planner map and in 3D, they check your mission before flying and they warn you during the flight, including real flights with telemetry.

> **Guidance information**
>
> The data comes from ENAIRE and the app always shows its date. It does not replace the official check in ENAIRE Drones or coordination through ENAIRE Planea: always check before flying. The simulator never says a flight "is authorised"; at most, that no zones were found in the data consulted.



Planner with the UAS zones around Osuna: the Hospital La Merced heliport (orange), the railway line (blue), the traffic light and the list of zones that affect the mission.

31.1 Where the data comes from

From ENAIRE's official files in ED-318 format (the European standard for UAS zones), published in the AIP. AIRLAB bundles them in three layers:

- **Aeronautical**: aerodromes, heliports, microlight sites, CTR, ATZ, FIZ, TMA and CTA, P (prohibited), R (restricted) and TSA/TRA (reserved) areas, model flying areas and restricted aerial photography areas. Updated every AIRAC cycle (28 days).

- **Infrastructure**: critical infrastructure and ADIF railway lines.
- **Urban environment**: a general notice for the whole territory.

They are downloaded by area around the planner home and saved in the browser, so they keep working offline. In **Settings and data** you can save the zones within 50 km of home in advance and choose the **live query to ENAIRE** (experimental): if ENAIRE does not answer, the bundled data is used. If there is a newer AIRAC cycle than the data, the app says so.

> **What the data does not include**
>
> ENAIRE does not publish the **environmental zones** (nature parks, SPAs…) in this data, nor structured **schedules**: CTR hours and NOTAM activation of TSA/TRA are written in each zone's message. Check them in ENAIRE Drones.

31.2 The zones on the planner map

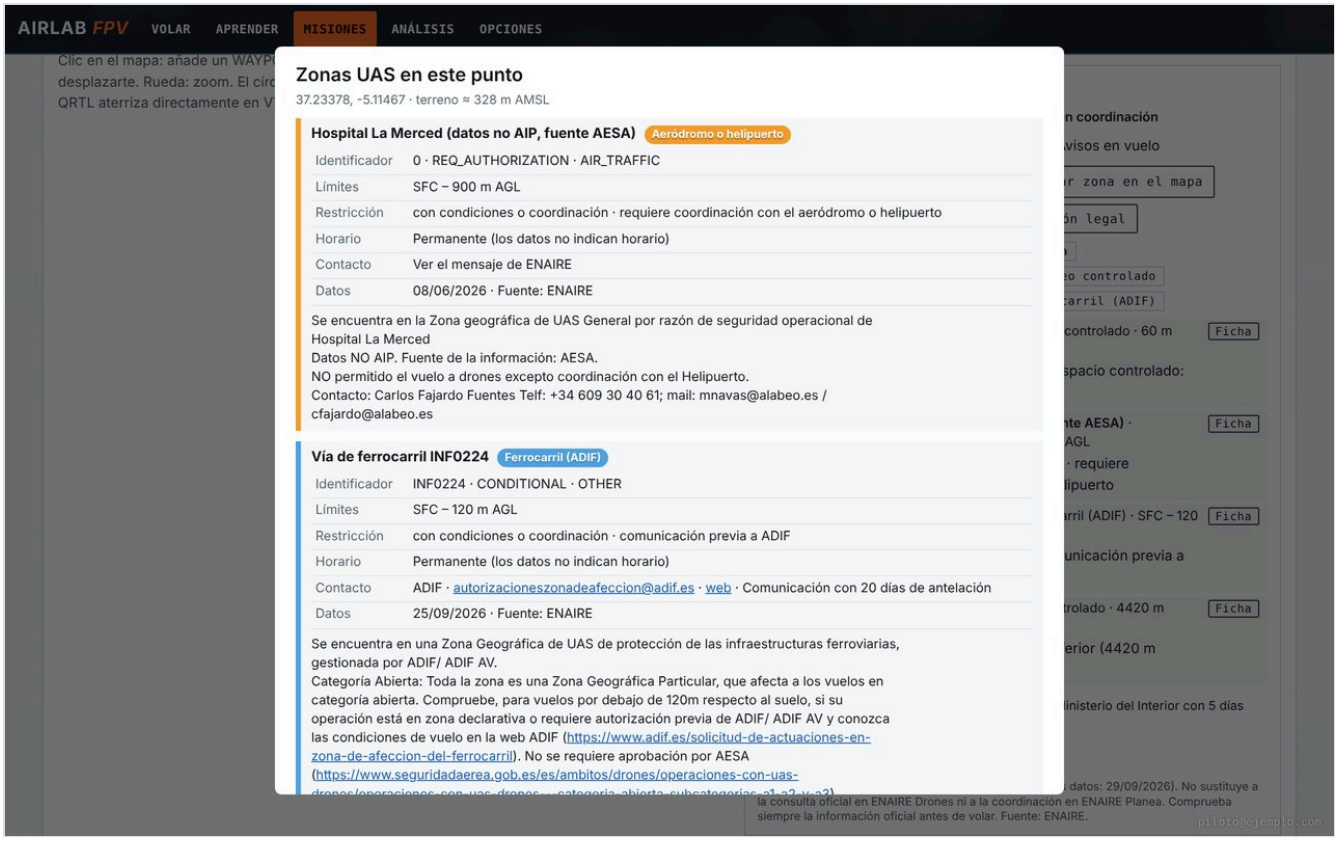
In **Plan › Planner**, below the list of warnings, is the **UAS zones (ENAIRE)** box. With **On the map** the zones are drawn with one colour per type; the legend is also a filter (tap a type to hide it). A zone with a **dashed outline** starts above the ground: you can fly below its lower limit without entering it.

How they look on the map. Zones that reach drone flight height (lower limit up to 150 m above the ground) are filled with a **diagonal hatch** of their colour, denser in prohibited and restricted ones, with a wide band inside the edge, like the in-flight curtains, so you can see which side the zone is on. Those that start very high (a TMA at 4420 m, for example) are only outlined with a dashed line. If the zone is larger than the view and the centre is inside, **"This whole view is inside:"** appears at the bottom left with their names. And the **mission route is painted in the zone's colour** on the legs that pass inside a zone at its flight height, with a ring on the waypoints that fall inside: red in a restricted zone, orange at an aerodrome, yellow in controlled airspace…

Type	What it means
Prohibited (P) · Restricted (R)	No flying, or only with express authorisation and its conditions.
Critical infrastructure	Prior express permission from the facility owner.
Aerodrome or heliport	Coordination with the aerodrome or heliport (contact in the details).
Controlled airspace (CTR, ATZ, FIZ, TMA, CTA)	Coordination with air traffic control, usually through ENAIRE Planea.
Reserved (TSA/TRA)	It may be active: check the NOTAMs.
Railway (ADIF)	Prior notice to ADIF (usually 20 days) below 120 m.
Restricted aerial photography	CECAF technical conditions for capturing images.
Model flying	For information: area with an exemption for model flying.
Urban environment	Notice to the Ministry of the Interior 5 calendar days in advance if you fly in an urban environment.

Press **Look up a zone on the map** and tap a point: you get the **details** of every zone at that point with its limits (AGL or AMSL), the restriction, the schedule, the coordination contact, ENAIRE's message as

published, the data date and links to ENAIRE Drones and ENAIRE Planea. Press the button again to go back to adding waypoints.



Details of the zones at a point: limits, restriction, contact and ENAIRE's message.

31.3 Checking the mission's airspace

Every time you change the mission, the planner checks each waypoint and each leg between waypoints, every 20 m, with its height. For zones with **AGL** limits it uses the height above the ground at that point; for **AMSL** ones, the home ground elevation (from the real terrain or an elevation model) plus the mission height. The result is a **traffic light**:

- **Green**: no zones detected in the data consulted.
- **Amber**: you enter zones with conditions or that require coordination.
- **Red**: you enter a prohibited zone or one that requires express authorisation.

The list shows for each zone whether you are **inside** (with which commands and at what heights), **near** (within 150 m) or **below** its lower limit. The zones you enter also appear in the planner's list of warnings. **PDF report** creates a report with the map, the traffic light, each zone with its contact and the legal notice, to take to the field or attach to a request.

31.4 Warnings during the flight

With the aircraft armed (in the simulator, with real ArduPilot in SITL and in a **real flight with telemetry**), the simulator warns you:

- when you **approach** a zone (200 m horizontally and 20 m vertically by default; change it in **Settings and data**);

- when you **enter**: “Caution: entering aerodrome or heliport zone: Hospital La Merced. Requires coordination…”, on screen and with Captain Pitot’s voice;
- when you **leave**.

While you are inside, a box at the bottom of the screen shows the zone, its type and its limits. Entries and exits are stored in the black box and in the flight summary, which reminds you what a real flight would require. In a real flight with telemetry it only warns: it never acts on the aircraft.



In flight: the zone volumes in 3D (the railway in blue), the approach warning and the current-zone box.

With **In 3D** the volumes of nearby zones are drawn between their lower and upper limits (cut at 160 m above the ground). The pre-flight checklist includes **Legal pre-flight**, with the home zones and the mission traffic light, and Captain Pitot’s chat knows the zone you are in to answer questions about it.

31.5 Exercise: legal flight planning

Exercise: legal flight planning gives you a job: photograph a building at 80 m near home, chosen with ENAIRE’s data (next to a railway line if there is one nearby). The target is marked on the map and in 3D.

1. Plan the mission: take off at home, fly over the target no higher than 120 m and land back.
2. Press **Check airspace** and review the details.
3. Press **Evaluate** and answer: which zones require something for your flight (passing below a zone’s lower limit does not put you in it), what must be done for each type and whether the Ministry of the Interior must be notified.

The score out of 100 counts the plan (25), checking the airspace with the final mission (10), identifying the zones (25), what to do for each one (30) and not entering zones you do not need (10). The marking explains every answer.

31.6 Operation summary before flying

When you press **Arm and start in AUTO**, before arming, the **Operation summary** appears. It follows the order in which a real operation is prepared in Spain:

1. **The mission**: commands and route length, duration and battery, maximum height above ground and maximum distance from the pilot. For a survey, also: area, passes, photos, GSD and planned side overlap.
2. **Category (guidance)**. Open A3 for a privately built aircraft under 25 kg without a class mark: away from people, 150 m from urban areas, in VLOS and up to 120 m. It becomes **specific** if it goes above 120 m, beyond the pilot's sight, over an urban environment or weighs 25 kg or more. The specific category needs an AESA operational authorisation, because standard scenarios STS-01 and STS-02 require classes C5 and C6. The rule sets no distance for VLOS, only that you see the aircraft unaided at all times; the summary warns from 500 m as a practical criterion for an aircraft this size.
3. **Airspace and permissions**: every UAS zone the route enters at its flight height, with what RD 517/2024 requires:
   - Aerodrome or heliport: coordinate with its manager (with AENA, through the ENAIRE Planea / ENAIRE Drones contact).
   - Controlled airspace: coordinate with ENAIRE through ENAIRE Planea 10 days ahead and at most one month before, except in VLOS, at 60 m AGL or less and outside aerodrome zones.
   - Critical infrastructure: permission from the owner.
   - Railway: notify ADIF.
   - TSA/TRA: check the NOTAMs.
   - Prohibited zone or one with express authorisation: no flying.
   - Urban environment: notify the Ministry of the Interior 5 days ahead.

   For each zone it also shows who (contact on the ENAIRE sheet) and the lead time. Zones you fly below their lower limit are listed separately.
4. **Requirements**: operator registered with AESA with the number on the aircraft, pilot, insurance, authorisation in the specific category, NOTAMs and weather for the day, permission for the site and data protection if the photos may capture people.
5. **Before arming**: what is still to resolve.

Simulate the request opens a form with the operation details: operator, pilot, phone, aircraft, date, time window, area and height. The request is «sent» today for the planned flight date, chosen at the top. If the lead time is not met, it is refused. If it is met, the reply arrives with a number and example conditions: height and time window, start and end calls, landing if a manned aircraft appears and that coordination is not an authorisation. In reality each manager sets their own conditions, and no reply means refusal. The calls and a helicopter warning are practised in the **Coordination in a UAS zone** exercise (36.5).

Fly is enabled when everything is resolved. **Fly anyway (practice)** starts the mission and writes what was missing in the log; in reality you could not fly like that. In flight it warns if you go above the coordinated height inside the zone. The tick box at the bottom turns the summary off.

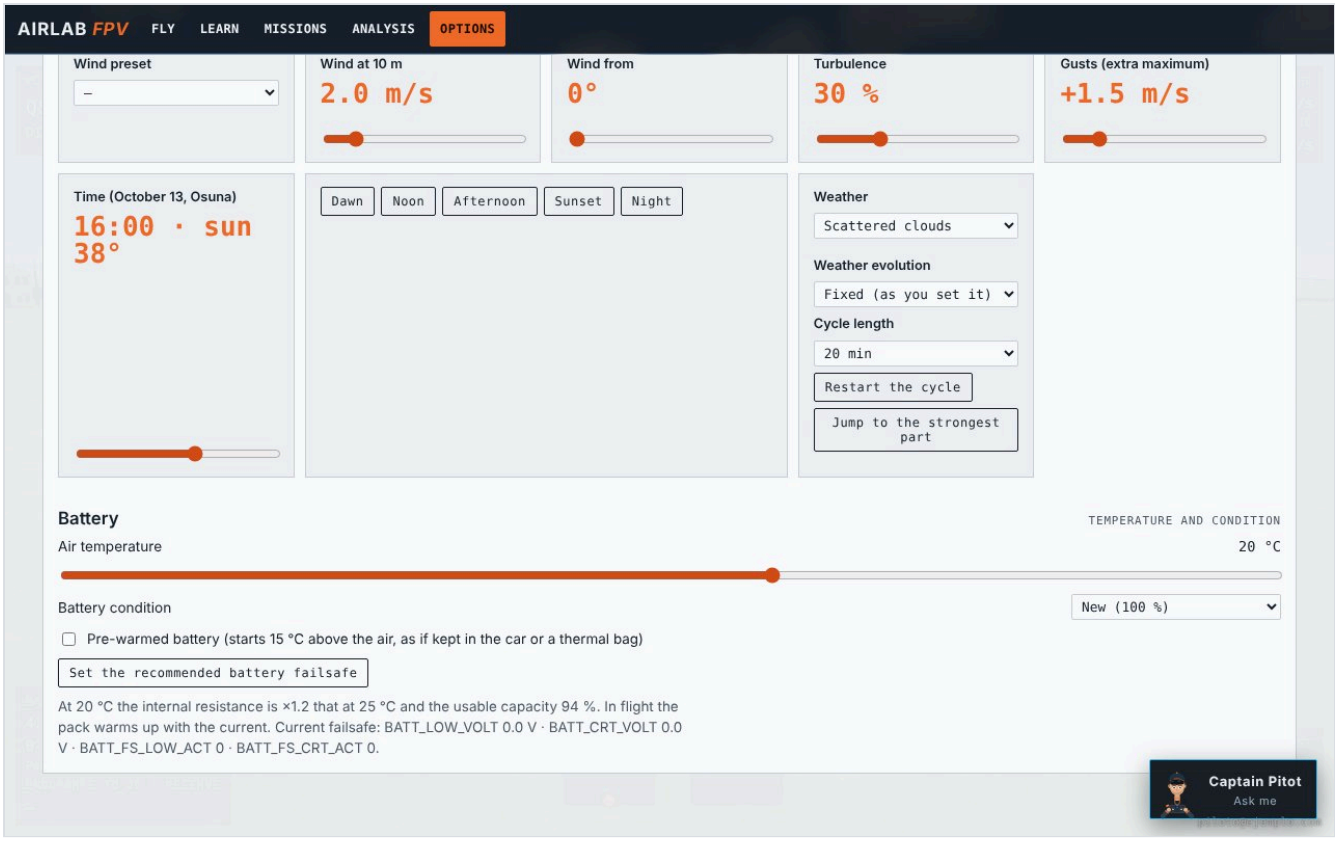
Sources: ENAIRE · UAS geographical zones (AIP, ED-318) · ENAIRE Drones · ENAIRE Planea · Royal Decree 517/2024 · AESA guide UAS-OPS-DT01 on UAS geographical zones.

32 Realistic battery and battery failsafe

The battery is no longer a fixed voltage minus a drop. It now behaves like a real LiPo:

- **Cold**: internal resistance rises in the cold (a little over twice as much at 0 °C as at 25 °C) and usable capacity falls (around 70% at 0 °C). At high throttle the voltage sags much more and the low-battery warning comes sooner.
- **Load**: on top of the immediate drop with current, there is a slow sag under sustained load (polarisation) that recovers within seconds when you throttle back.
- **Warming**: the pack heats up with current and cools in the air. A cold battery performs better after a few minutes of flight.
- **Wear**: a used or worn battery has more resistance and less capacity.
- If the battery cannot deliver the power the motors demand, the voltage drops and so does thrust.

In **Options › Time and weather › Battery** you choose the **air temperature** (with variable weather the weather's temperature is used), the **battery condition** (new, used or worn) and whether it is **pre-warmed** (it starts 15 °C above the air, as if kept in the car or a thermal bag). In the OSD, below the voltage, you see the pack temperature and **VOLTAGE SAG** when the sag under load is large.



Options › Time and weather › Battery: temperature, condition, pre-warming and the recommended failsafe.

32.1 Battery failsafe as in ArduPilot

It works like the ArduPlane battery monitor, with its parameters (in **Parameters › Battery**):

Parameter	What it does

BATT_LOW_VOLT · BATT_LOW_MAH	Low battery: by voltage (below it for BATT_LOW_TIMER seconds) or by mAh used. 0 = disabled.
BATT_CRT_VOLT · BATT_CRT_MAH	Critical battery, with values lower than the low ones.
BATT_FS_LOW_ACT · BATT_FS_CRT_ACT	Action: 0 warn, 1 RTL, 2 Land, 3 Terminate, 4 QLand, 6 Loiter to QLand. Critical overrides low.
BATT_LOW_TIMER	Seconds continuously below the voltage before triggering (10 by default): avoids triggering on a throttle punch.
BATT_FS_VOLTSRC	0: measured voltage. 1: sag-compensated voltage (what it would read at rest), steadier at high throttle.

When it triggers you get the ArduPilot messages («Battery 1 is low 21.40V used 6400 mAh») and Captain Pitot announces it. On an aeroplane without VTOL, QLand becomes RTL. **Set the recommended battery failsafe** sets 3.5 V per cell or 25% of mAh → RTL and 3.3 V per cell or 10% → QLand (Land on an aeroplane). Betaflight drones have no battery failsafe: they warn in the OSD with **BATT LOW** and **LAND NOW**. With real ArduPilot (SITL) or in real flight, the real ArduPilot is in charge.

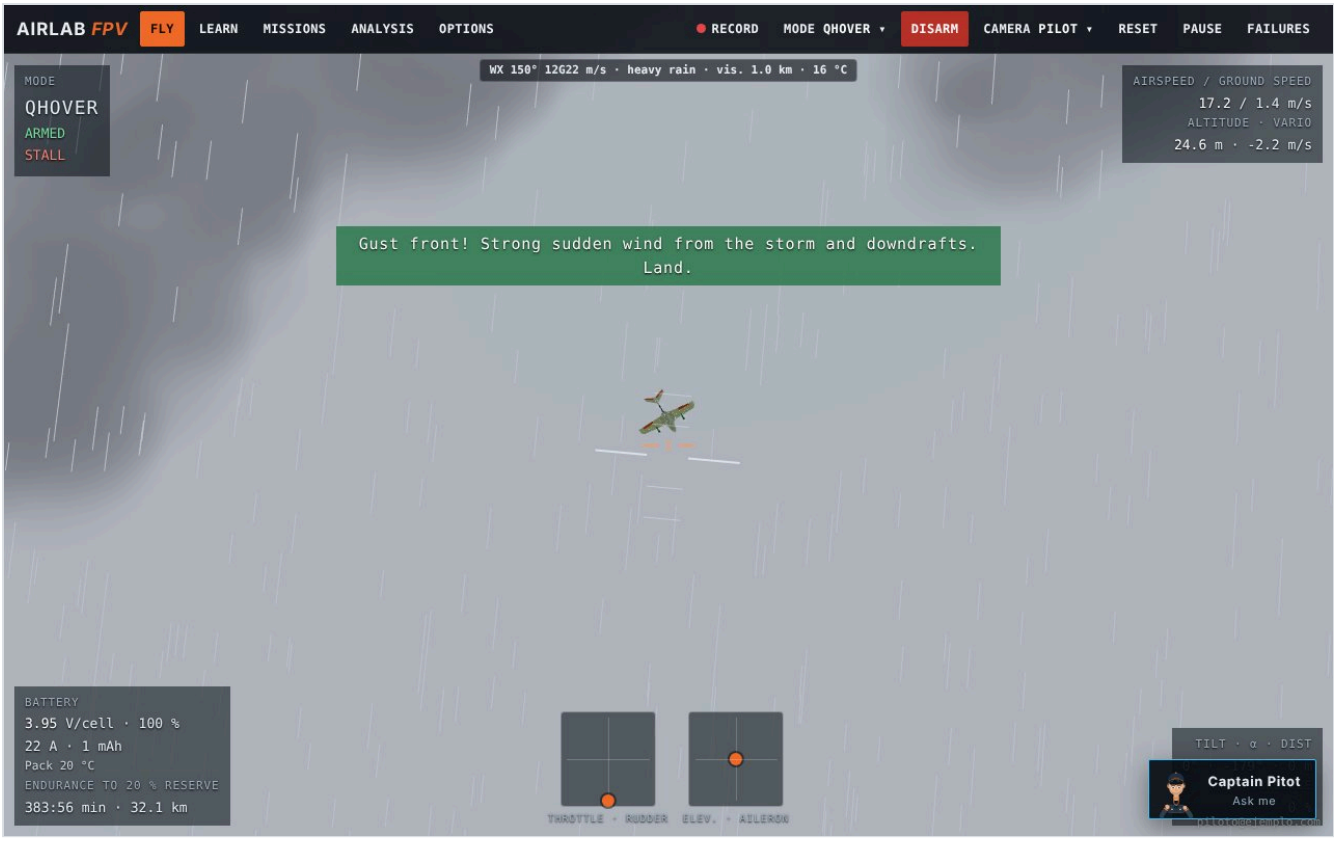
> **Try it**
>
> Set 0 °C and a used battery, take off in QHOVER and climb at high throttle: you will see the voltage sag and the warning trigger much earlier than at 25 °C. Repeat with the battery pre-warmed.

33 Variable weather during the flight

In **Options › Time and weather › Weather evolution** the weather can change while you fly:

- **Fixed**: as you set it (as before).
- **Variable**: clouds, wind speed and direction change slowly and every now and then a shower passes.
- **Front passing**: the sky clouds over, the wind backs south-west and strengthens, heavy rain lowers the visibility, the temperature drops about 6 °C and then it clears with a gusty north-westerly.
- **Summer thunderstorm**: heat and growing cumulus; then the **gust front** (strong, sudden wind out of the storm, with downdrafts), torrential rain with lightning and thunder, and then it clears.

You choose the **cycle length** (8, 15, 20 or 40 minutes), and can restart it or **jump to the strongest part**. At the top centre, the **WX** line gives wind and gusts, rain and visibility, like a weather report. Captain Pitot announces each change («Gust front! … Land.»).



Summer thunderstorm: rain, reduced visibility and the WX line with wind and gusts.

Rain reduces visibility, darkens the ground and, on the FPV camera, leaves **drops on the lens** that clear with speed. Lightning lights up the scene and the thunder arrives with the delay of the distance. The battery uses the weather's temperature (chapter 32). Going back to «Fixed» restores your wind and weather settings.

> **What it teaches**
>
> The gust front arrives before the rain and can double the wind in seconds. If you see cumulus towers growing, land before it arrives.

34 Real terrain anywhere in the world

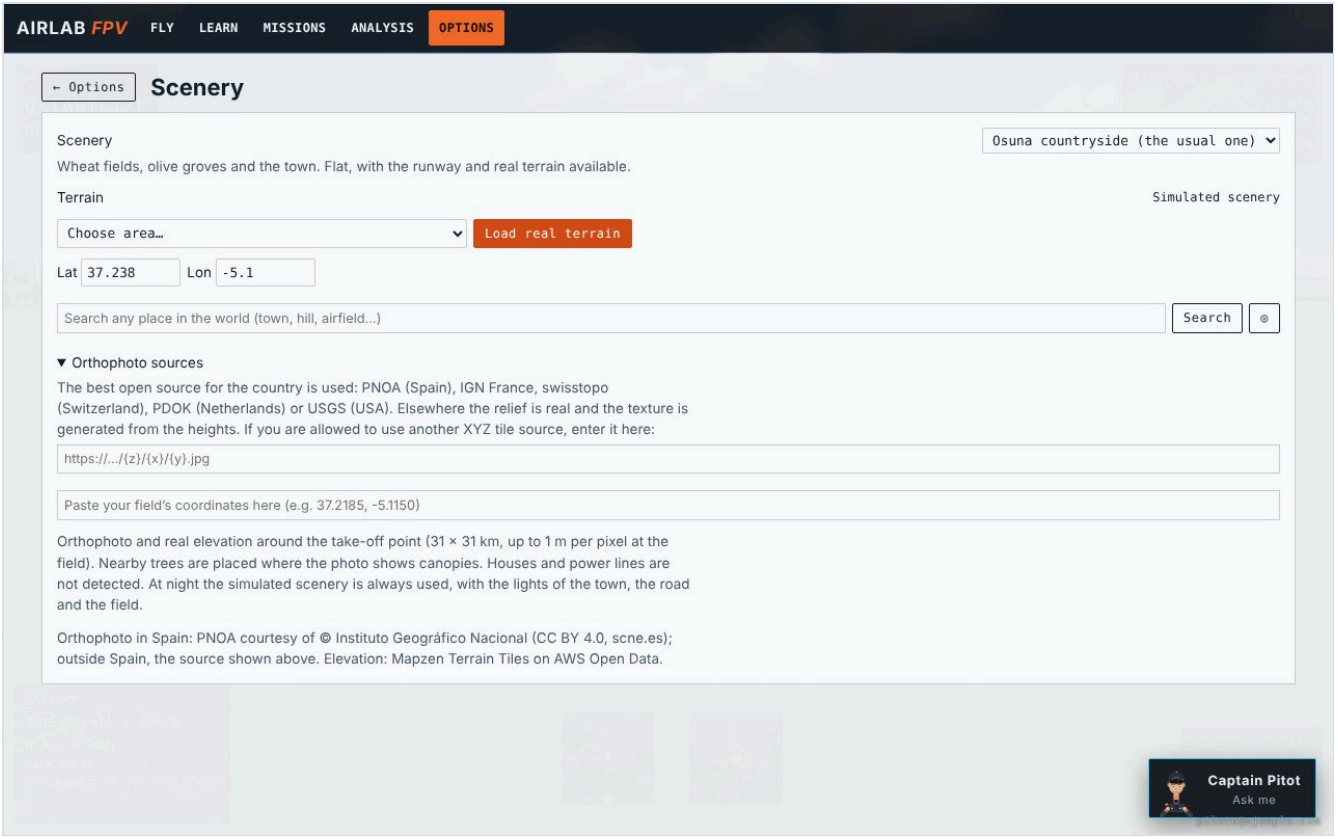
Real terrain is no longer limited to Spain. In **Options › Scenery › Real terrain**:

- **Search**: type a town, a mountain or an airfield in any country (OpenStreetMap search). There is also **Use my location** and the recent places.
- **Known flying sites**: Dune of Pilat (France), Torrey Pines (USA), Mam Tor (UK), Algodonales and Piedrahíta, Àger, Interlaken (Switzerland) and Texel (Netherlands), plus the Osuna areas.

The **elevation** is real worldwide (SRTM, EU-DEM). The **orthophoto** comes from open sources by country:

Area	Orthophoto
Spain	IGN PNOA
France	IGN France (BD ORTHO)
Switzerland	swisstopo SWISSIMAGE
Netherlands	PDOK Luchtfoto
United States	USGS The National Map
Rest of the world	Texture generated from the elevation: crops and meadows on flat ground, rock on slopes, snow on high peaks and sea at sea level, with hillshading.

If a source does not respond or returns blank tiles, the next one is tried. You can add your own XYZ tile source (respect its terms of use). The data attribution is always shown below. The **sun** rises and sets according to the site's latitude and the date. ENAIRE UAS zones only exist in Spain: elsewhere the traffic light is grey («No data»).



Options › Scenery › Real terrain: search, known flying sites and orthophoto sources.

35 Sailplane contests

From **Learn › Sailplane contests**, or with the sailplane chosen from the **Contests** button in the variometer box, you open two tasks inspired by RC sailplane competitions (not the full FAI rules):

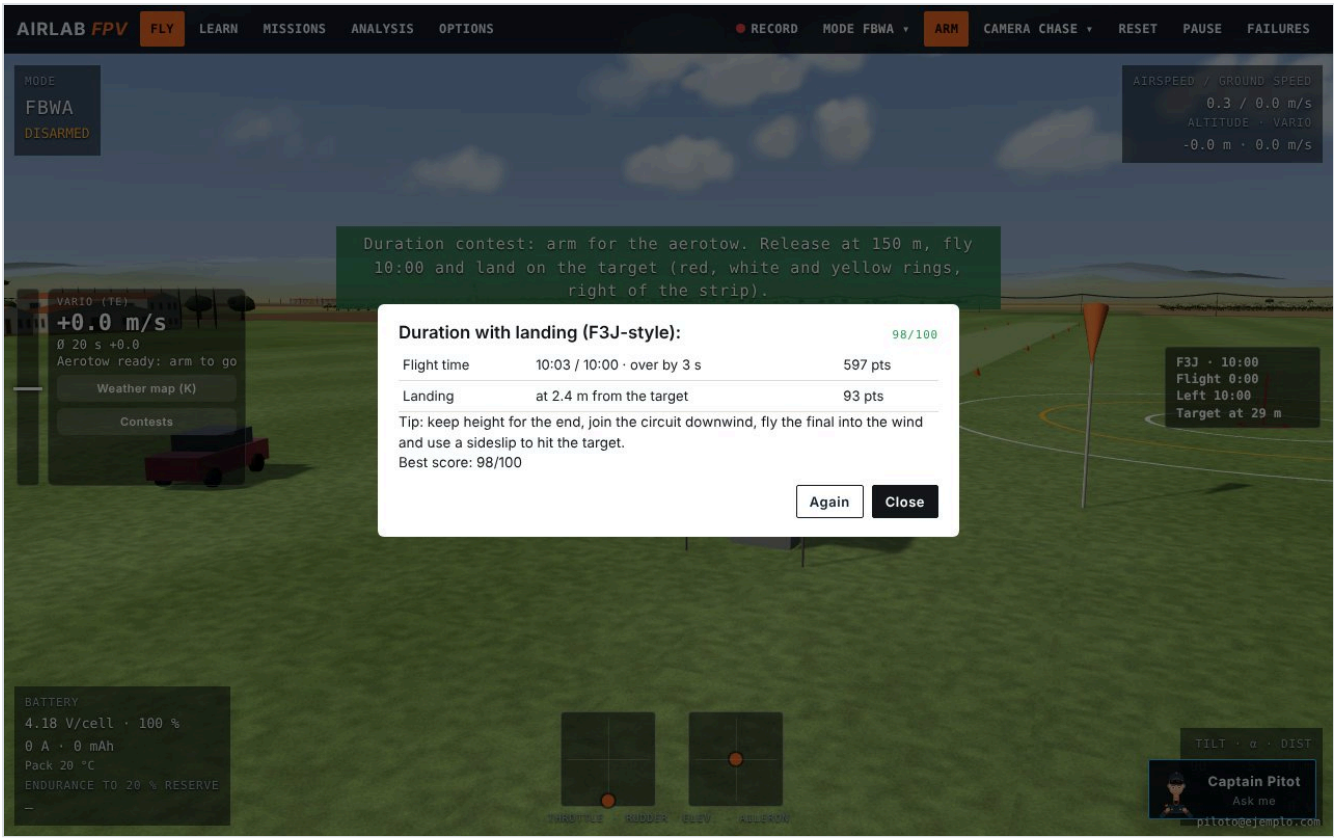
35.1 Duration with precision landing (F3J-style)

1. Choose the working time: 5, 8 or 10 minutes. You go up on aerotow and release at 150 m.
2. The clock starts at release. Find thermals to stay up exactly the target time.
3. Land on the **target**: the red (1 m), white (5 m), yellow (10 m) and white (15 m) rings, right of the strip.

Points: 1 per second of flight up to the target; each extra second subtracts 1. Landing: 100 at 1 m or less, falling to 30 at 15 m; further away, 0. If you break it on touchdown, 0. The score out of 100 is 80% flight plus 20% landing. On the right, an indicator gives the flight time, time left and distance to the target.

Indicator and ranking: top right, the large countdown (green, yellow in the last minute and red if you go over), the time flown and the distance to the target. Below, the live **group ranking**: you fly at the same time as 7 fictional pilots, who add points while flying and their landing on touchdown; some land early. At the end you get the final ranking with your place.

Captain Pitot's comments: every half minute or so he tells you your place and gives a tip based on what you are doing (you are in lift, you are low with plenty of time left, you have height to spare…); he calls half time, one minute and 25 s left, when a rival lands and when you move up. If you use the motor glider's motor after release, the flight time does not count: F3J has no motor.



Duration contest result: flight time, landing and score.

35.2 Slope distance (F3B distance-style)

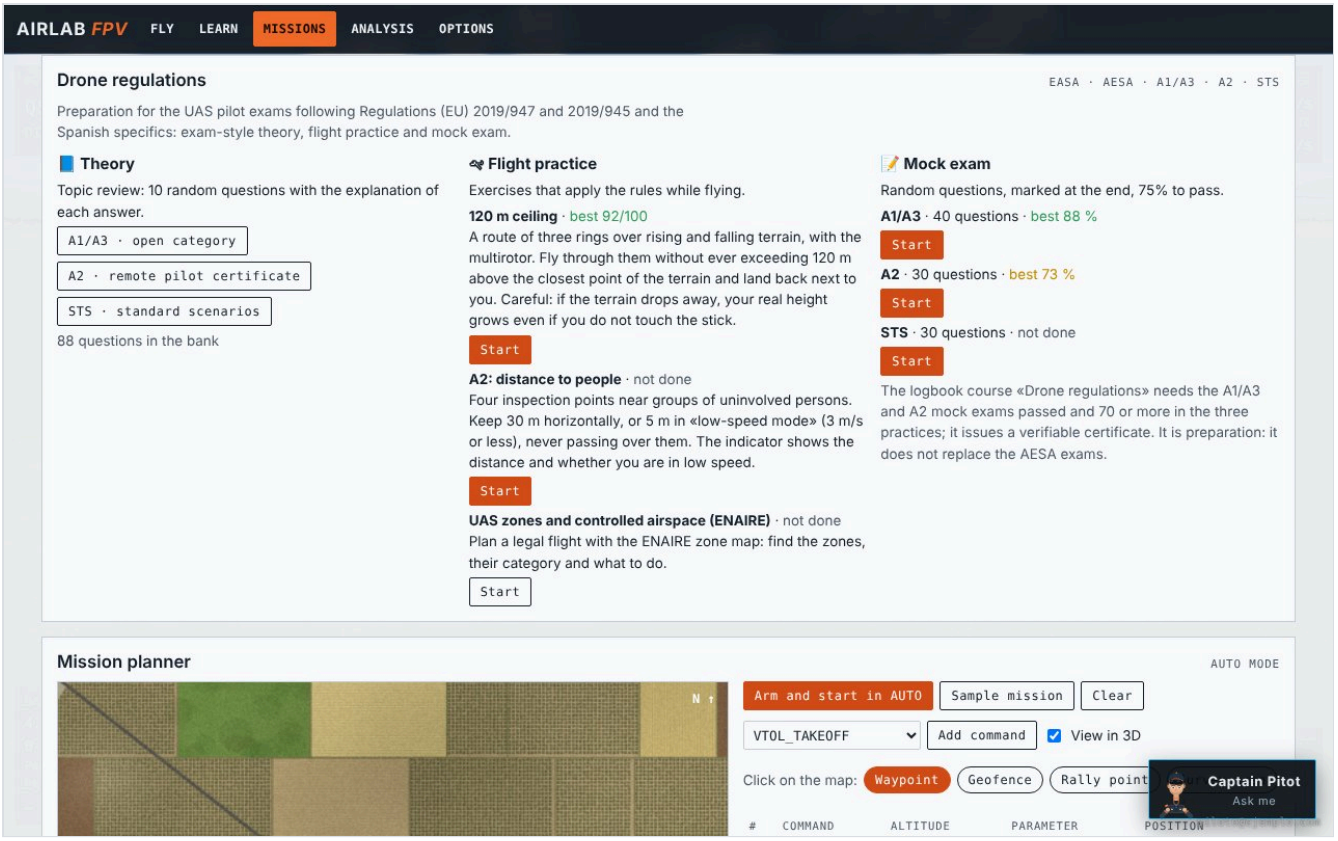
Hand launch on a slope with the wind in your face. Two bases, A (orange) and B (blue), 150 m apart along the slope. Complete base-to-base **legs** in **4 minutes** count; the score is legs / 16 × 100 (16 legs are 2,400 m, about 10 m/s on average). You need a slope and 6–9 m/s of wind against it: the cliff, the sierra or a real slope site (Dune of Pilat, Algodonales…).

The best score of each task is saved and shown in the contests menu.

In distance there is also a countdown and a live ranking with each rival’s legs.

36 EASA/AESA drone regulations

In **Learn**, the **Drone regulations** panel prepares you for the UAS remote pilot exams in the open category (A1/A3 and A2) and for the standard scenarios (STS) of the specific category, following Regulations (EU) 2019/947 and 2019/945 and the Spanish specifics (AESA, ENAIRE geographical zones).



Learn › Drone regulations: theory, flight practice and mock exam.

36.1 Theory

A bank of exam-style questions on three topics: **A1/A3** (open category: heights, distances, classes C0–C4, registration, VLOS and observer, zones, night, responsibilities), **A2** (meteorology, UAS flight performance and ground-risk mitigation) and **STS** (standard scenarios, controlled ground area, contingency volume, declaration, SORA, PDRA and LUC). The review gives 10 random questions and, as you answer, the explanation of each one.

36.2 Flight practice

- **120 m ceiling**: with the multirotor, a route of three rings over the terrain that rises and falls most around home. Fly through the rings without ever exceeding 120 m above the **closest point** of the terrain and land back next to you. The indicator shows the height above terrain and above take-off: over a valley or off a cliff, the real one is greater than what the drone shows.
- **A2: distance to people**: four inspection points near groups of uninvolved persons. Keep 30 m horizontally or, in **low-speed mode** (3 m/s or less), 5 m; never above them. On the ground, the red circle marks 30 m and the yellow one 5 m. The indicator shows the distance to the nearest person, your speed and whether you are in low speed.

- **UAS zones and controlled airspace**: the legal planning exercise with the ENAIRE map (chapter 31.5).
- **Coordination in a UAS zone**: a complete flight inside a zone that requires coordination (section 36.5).

Each practice gives a score out of 100: the rings, landing next to the pilot and compliance (time above 120 m or breaking the distance). Flying over uninvolved persons caps the score at 40.



A2 practice: groups of people with the 30 m and 5 m circles, the inspection ring and the distance and speed indicator.

36.3 Mock exam

Mock exam	Questions	Pass mark
A1/A3	40	75%
A2	30 (mostly A2, with some A1/A3)	75%
STS	30 (mostly STS)	75%

There is no marking until the end; then you get the score, whether you passed and the missed questions with the correct answer and its explanation.

36.4 Certificate

The logbook course **EASA/AESA drone regulations** requires: A1/A3 and A2 mock exams at 75% or more, the «120 m ceiling» and «A2: distance to people» practices at 70 or more and the UAS zones exercise at 70 or more. It issues a verifiable certificate like the others.

> **It is preparation**
>
> It does not replace the AESA exams and does not authorise you to fly. For the open category, the A1/A3 training and exam are taken on the AESA website, and the A2 exam at AESA or a recognised entity (in other EU countries, at your national aviation authority).

36.5 Coordination in a UAS zone

Practise what is needed to fly inside a UAS geographical zone that requires coordination. Choose the case: **your real zone** (if the take-off point is inside the zone of an aerodrome, a heliport or controlled airspace), or an example: **hospital heliport**, **AENA airport** or **ENAIRE CTR**. Six steps:

1. **Identify**: three questions on who to coordinate with, how far ahead and what coordination is.
2. **Request**: operator, pilot, mobile during the flight, date, times, height, radius, category, aircraft and description. If something essential is missing, the date is too late or you ask for more than 120 m, it is returned.
3. **Reply**: coordination number and conditions (maximum height, radius, call at the start and at the end, time window, phone operational, land if a manned aircraft appears, check the NOTAMs).
4. **Start call**: the manager's phone rings and you speak into the microphone (in Chrome) or type. Say who you are, the coordination number and that you are starting the flight; if something is missing, they ask for it.
5. **Flight**: with the inspection multirotor, fly through the two blue points without leaving the yellow circle or exceeding the maximum height. The indicator on the right shows height, distance, points and the remaining window. At some point the manager warns of a medical helicopter: land immediately.
6. **End call**: after landing and disarming next to you, call to close the flight.

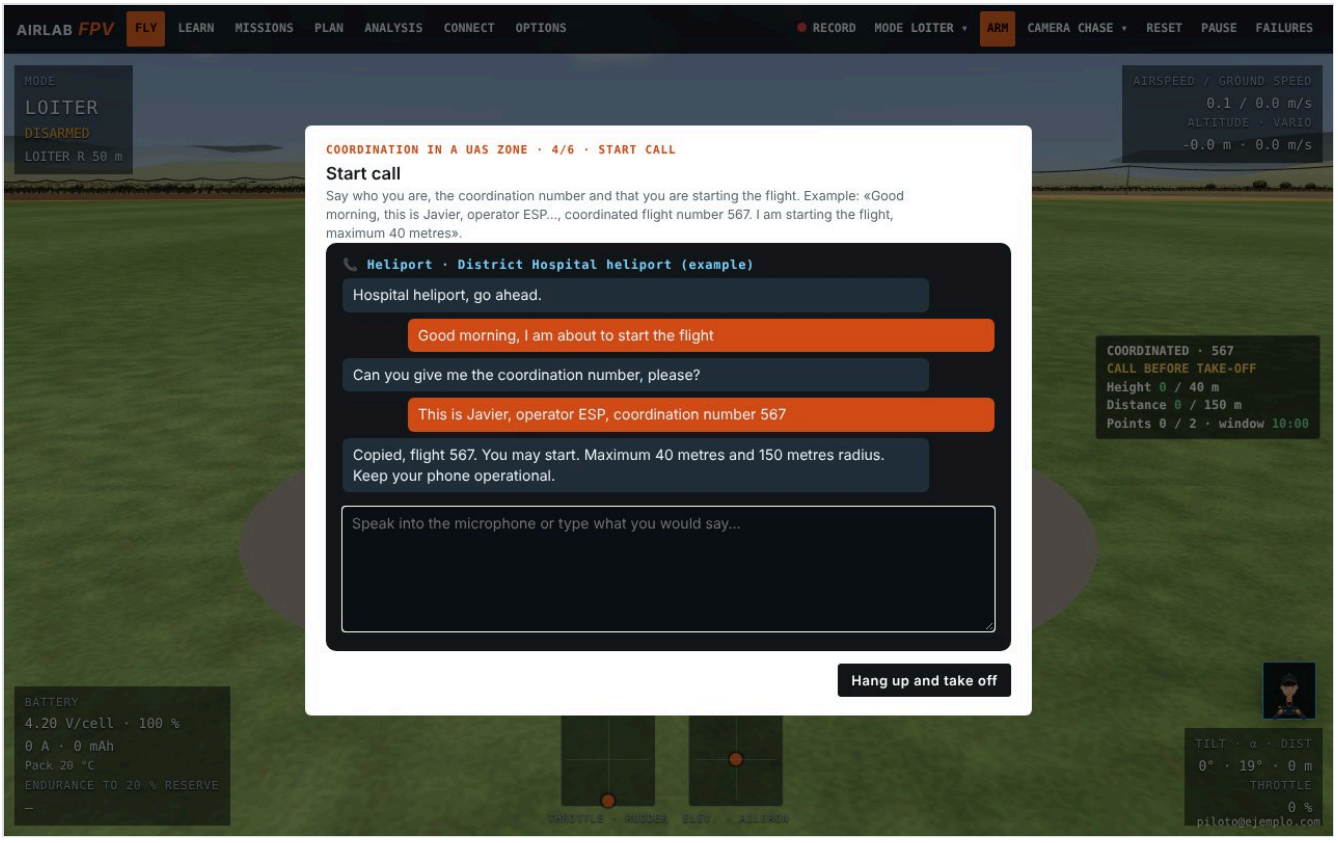
Item	Points
Identification	15
Request (−4 for each return)	20
Start call	15
Maximum height	15
Operation area	10
Landing for the helicopter (≤30 s full, ≤60 s half)	15
Inspection points	5
End call	10

Deductions: taking off before the start call (−15), going past the window (−5) and not ticking the NOTAM check (−5). A crash caps the score at 30.

If you leave the conditions (maximum height or area), the manager calls you straight away: «descend below 40 metres» or «return to the area», and the indicator counts down 15 s. If you correct, it is noted (−3 if you take 5 s or less, −6 if longer). If you don't correct within 15 s, or go past twice the coordinated height or radius, the manager **cancels the coordination** and orders you to land: land, disarm, and the

score is capped at 40. In reality, breaking the coordination conditions is an infringement and the coordination is no longer valid.

> **How it works in reality**
>
> In ENAIRE controlled airspace no coordination is needed if you fly VLOS, outside aerodrome and heliport zones and at 60 m AGL or less; otherwise the request goes through ENAIRE Planea at least 10 calendar days ahead (and at most one month), and silence means refusal. Near AENA airports, through the contact listed in ENAIRE Planea or ENAIRE Drones, 20 days ahead. Each heliport has its own procedure (for example, the Hospital General de Segovia one asks for 7 calendar days). Coordination is not an authorisation and each manager sets their own conditions: the exercise ones are an example.



Start call to the heliport manager: the manager asks for the missing coordination number and clears the flight.

Sources: Implementing Regulation (EU) 2019/947 and its AMC/GM · Delegated Regulation (EU) 2019/945 · AESA · Drones · Royal Decree 517/2024. · ENAIRE · flying in ENAIRE zones · Hospital General de Segovia heliport coordination procedure

37 Real world: weather, traffic and slope

Three features that bring into the simulator what is happening right now at your flying site: the real weather, the aircraft and helicopters flying nearby and the slope lift of the real relief. They need internet; the offline version requests the data from www.airlabfpv.es.

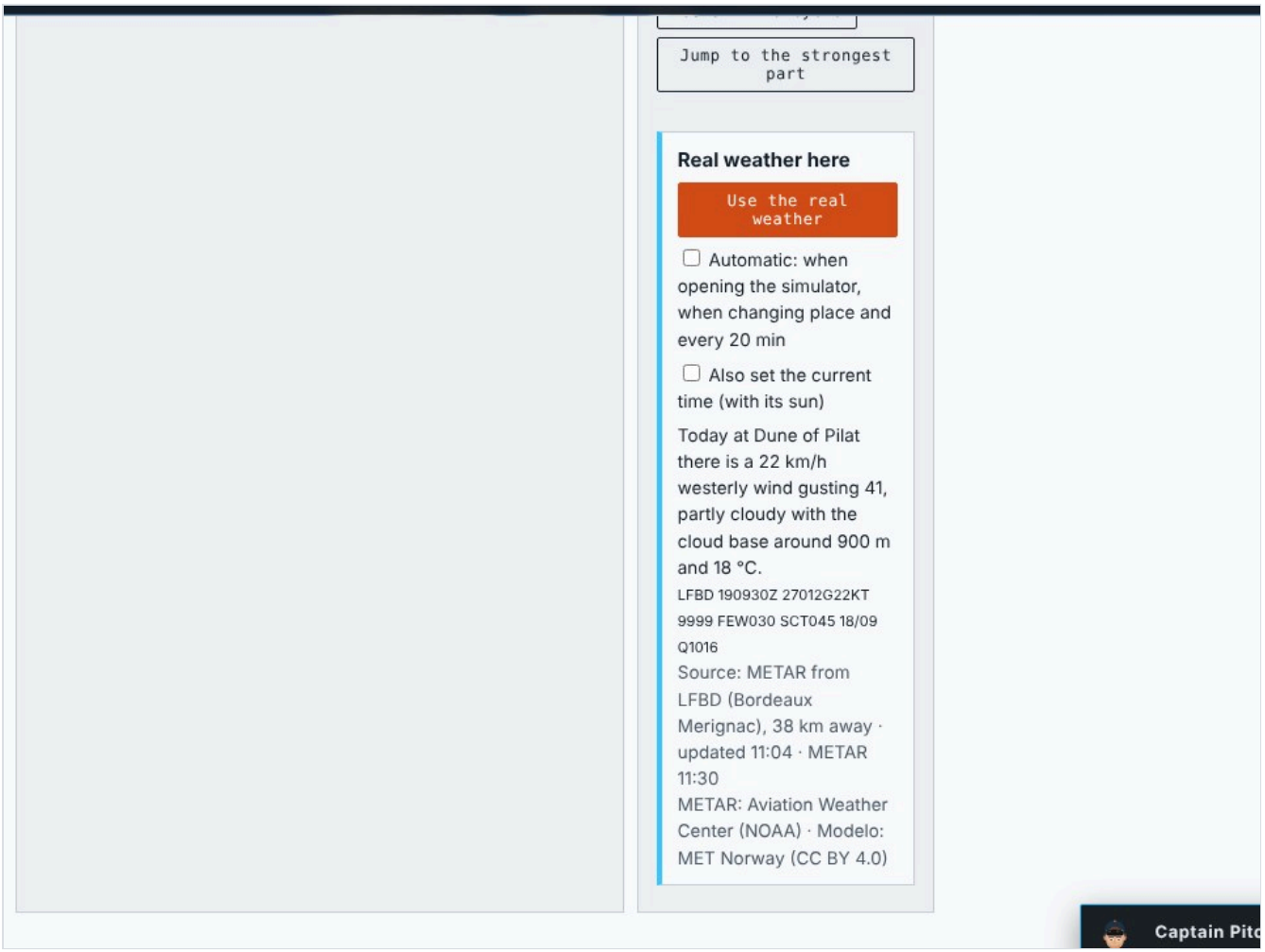
37.1 Real weather here

In **Options › Time and weather**, the **Real weather here** box (and, next to the real terrain, the **Real weather for this place** button). Pressing **Use the real weather** fetches the report for the take-off point:

- **METAR** from the nearest aerodrome (up to 60 km), from NOAA's Aviation Weather Center. If it is 45 km or closer, the METAR is used.
- If there is no aerodrome nearby, the **MET Norway model** at the exact point.

It is applied to the simulator: mean wind, direction and gusts; cloud cover and cloud base (also the top of the sailplane's thermals); visibility (fog); rain; temperature (used by the battery) and QNH (used by real traffic for altitude). Captain Pitot sums it up, for example: «Today at Osuna there is a 25 km/h westerly wind gusting 40, a few clouds and 18 °C». At the top, in flight, the **REAL WX** indicator shows the aerodrome, the wind and the temperature until you change them by hand.

With **Automatic** it updates when opening the simulator, when changing place and every 20 minutes. **Also set the current time** moves the sun to the time now. The simulator allows up to 54 km/h (15 m/s) of wind: if there is more, it stays there and warns you.



Options › Time and weather › Real weather here: summary, raw METAR and sources.

37.2 Real air traffic

In **Options › Scenery**, the **Real air traffic** box. With **Show the real traffic right now**, every 10 s the aircraft and helicopters transmitting ADS-B within the chosen radius (9 to 46 km) arrive and are drawn in 3D: light aircraft, jet, heavy jet, helicopter with rotor, glider, ultralight or balloon, by category. Between two updates they move with their speed, track and climb rate.

- **Height**: barometric altitude corrected with the real-weather QNH, minus the height of your take-off point.
- **Visual aids**: a dot visible from far away and a label with callsign, type, height difference and distance. Without them, you see them as in reality: very small until they are close.
- **Traffic indicator** (left): the four nearest with distance, height difference and clock position. Yellow, closer than 3 km and 450 m in height; red, closer than 800 m and 200 m.
- **See and avoid**: with the drone in the air, Captain Pitot warns of approaching traffic («traffic at 3 o'clock, 1.2 km, 150 m above: helicopter») and, if it passes close, raises the alert so you descend and move away.
- **Practice traffic**: a simulated helicopter or light aircraft crossing near you every minute and a half, anywhere and at any time.

In the **coordination in a UAS zone** practice (36.5), if a helicopter is nearby the manager's warning comes earlier and gives its callsign.



3D traffic with label and indicator: a helicopter crosses in front of the drone.

> **It does not replace watching the sky**
>
> Only aircraft with ADS-B appear, a few seconds late. Many light aircraft, helicopters and gliders do not carry it. Data: adsb.lol (ODbL licence).

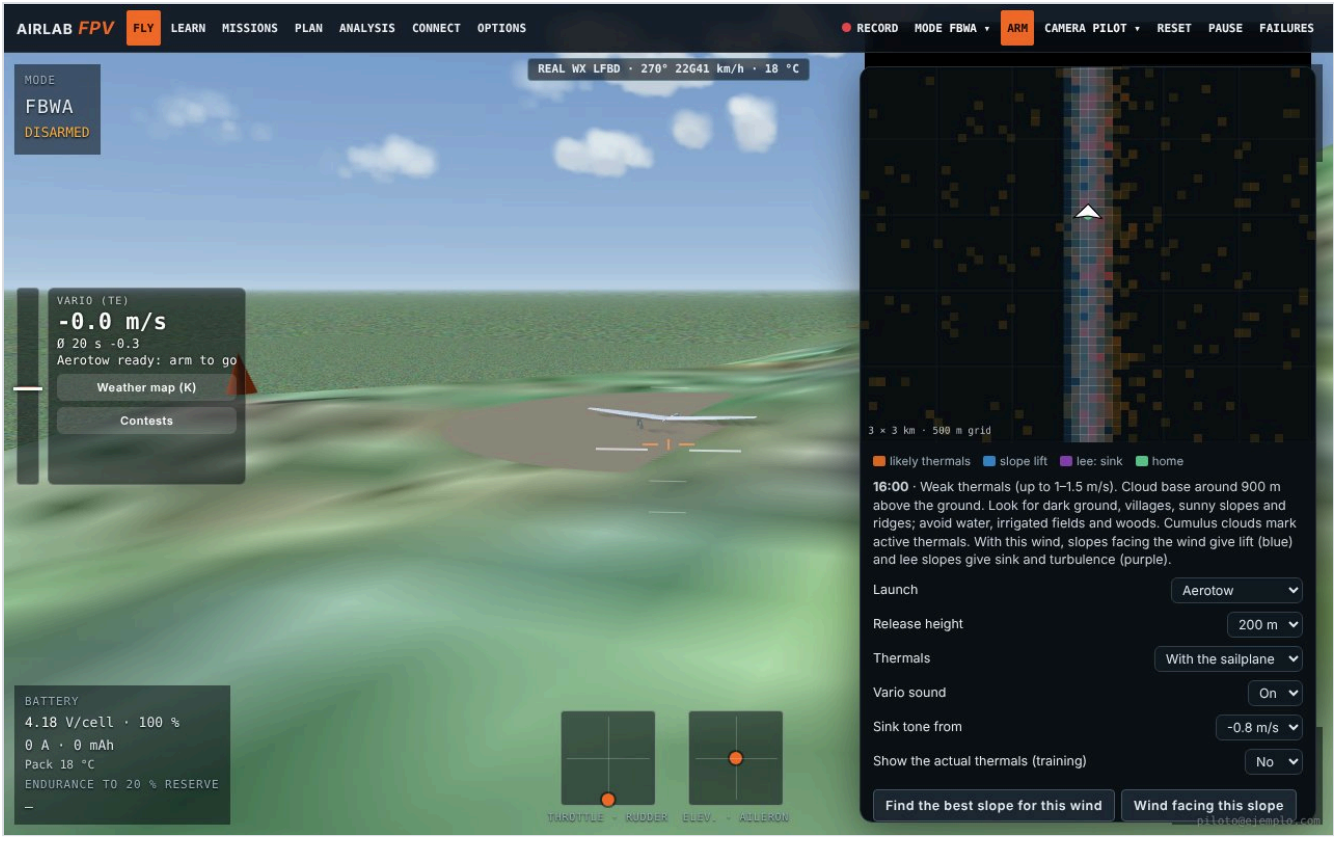
37.3 Slope and thermals on the real terrain

With the sailplane over the real terrain, wind hitting a slope rises: the lift is computed from the gradient of the real relief upwind and reaches higher the taller the slope (up to about 1.5 times its height above the crest). Behind the crest, in the lee, the air sinks. Thermals start where the sun heats most: each slope's orientation and the colour of the orthophoto (dry soil, stubble and towns heat more than woodland, irrigated fields or water). With real weather, the cloud base is the METAR one.

In the sailplane weather map (K key) and next to the real terrain:

- **Find the best slope for this wind**: scans 1.5 km around and picks the slope edge that best faces the wind. Moves take-off there, reloads the terrain, selects the sailplane and hand launch.
- **Wind facing this slope**: sets the wind blowing up the slope at your take-off point, to practise even when the real weather does not help.

With hand launch, on reset the sailplane faces the wind. Fly in front of the edge, upwind, and always turn away from the slope.



Sailplane weather map with slope lift (blue), lee sink (purple) and the slope buttons.

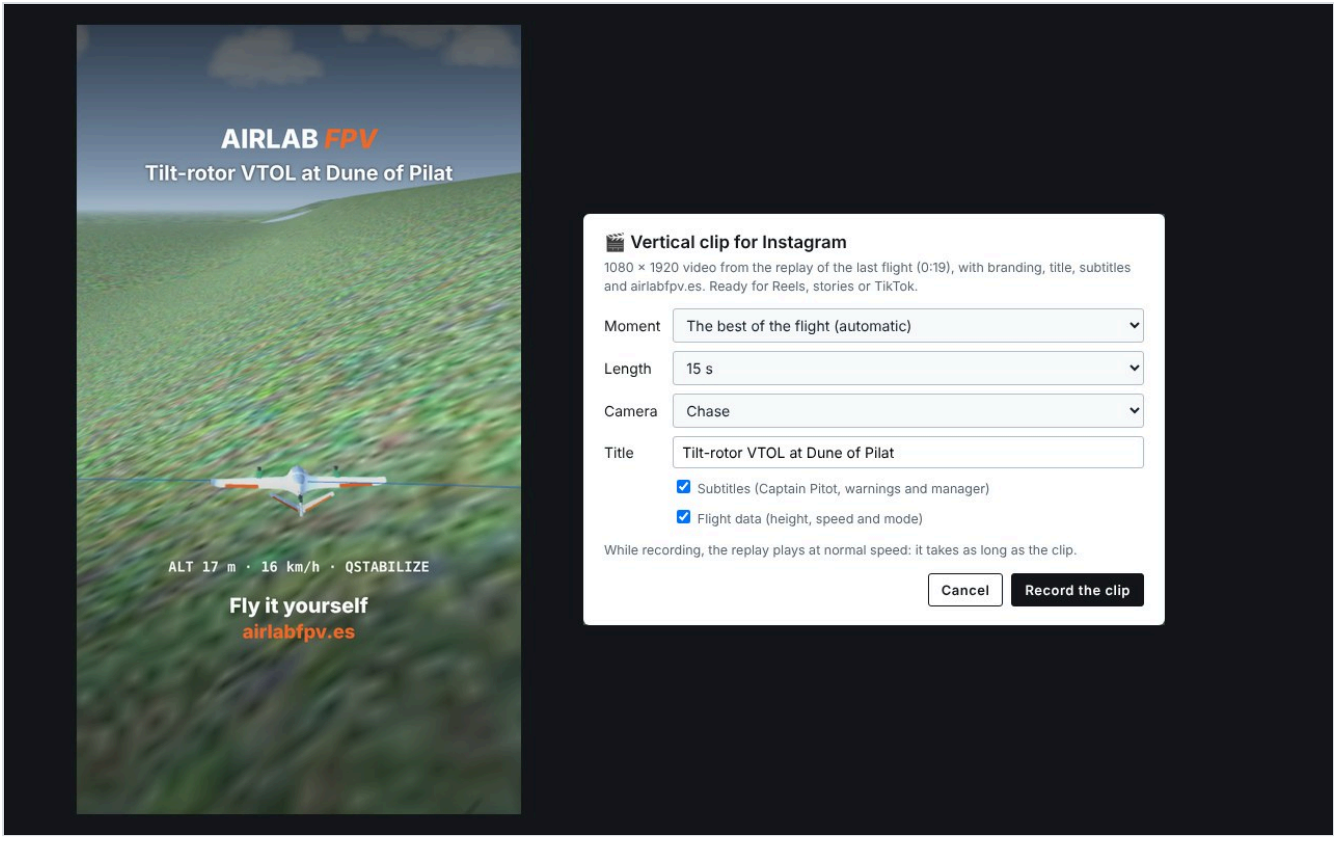
Sources: NOAA Aviation Weather Center · Data API · MET Norway · terms (CC BY 4.0) · adsb.lol (open ADS-B data, ODbL).

38 Vertical clip for Instagram

The 🎬 **Clip for Instagram** button (in the replay controls, in the Analysis black box and in the «Share it?» prompt at the end of a flight) creates a 1080 × 1920 vertical video from the replay of the last flight, ready for Reels, stories or TikTok.

- **Moment**: **mix of key moments** (automatic: take-off, transitions, top speed and height, manoeuvres, Captain Pitot warnings and landing, joined with a short fade and a label naming each moment), **choose the moments** (a list with tick boxes and the flight timeline with the height; click the line to add another moment; the clip length is shared among the chosen ones), **the best continuous stretch**, the end of the flight or from the current replay position.
- **Style**: **full-screen video**; **video on top, telemetry below** (big height, speed and vario, artificial horizon, the track on a map, mode, heading and battery); **chase on top, FPV below**, both cameras at once; **chase on top, FPV with OSD below** (a drone OSD: mode, voltage, mAh, speed, height, vario, artificial horizon, time and home arrow with distance); or **cinematic** (cinema bars, the camera orbits the aircraft and the replay runs in ×0.5 slow motion).
- **Hook**: a big sentence for the first two seconds so people stop scrolling (one is suggested for the aircraft; empty, no hook). **End card** of 1.6 s with the brand and airlabfpv.es.
- **Length**: 10, 15 or 30 s. **Camera**: chase, FPV, free or from the pilot.
- **Title** (aircraft and place, editable), **subtitles** with what Captain Pitot, the warnings and the coordination manager said (the browser voice cannot be recorded, so it is written) and **flight data** (height, speed and mode).
- It always carries the AIRLAB FPV brand at the top and «Fly it yourself · airlabfpv.es». Texts stay inside the Reels safe zone: away from the header, the buttons on the right and the footer.

While recording, the replay plays at normal speed with engine and wind sound: it takes as long as the clip. When done it downloads by itself. Instagram needs **MP4**: up-to-date Chrome and Edge and Safari record it; if your browser only records WebM, it tells you.



Vertical clip: brand, title, Captain Pitot subtitle, flight data and airlabfpv.es.

39 Operations centre: drone in a box

A **drone-in-a-box** system (dock) keeps the multirotor in a box on site: the box protects it, charges it and opens its lid so it can take off on its own. The operator flies it remotely from an operations centre, almost always with commands (go to, orbit, follow, return) and the gimbal camera, and only takes the sticks when needed. This is how facility and port security, wildfires, search and rescue, coastal surveillance and emergencies such as a flood are worked. This chapter explains the complete exercise.

39.1 How to open it

In **Missions › Operations centre**, press **Open the operations centre**. You choose:

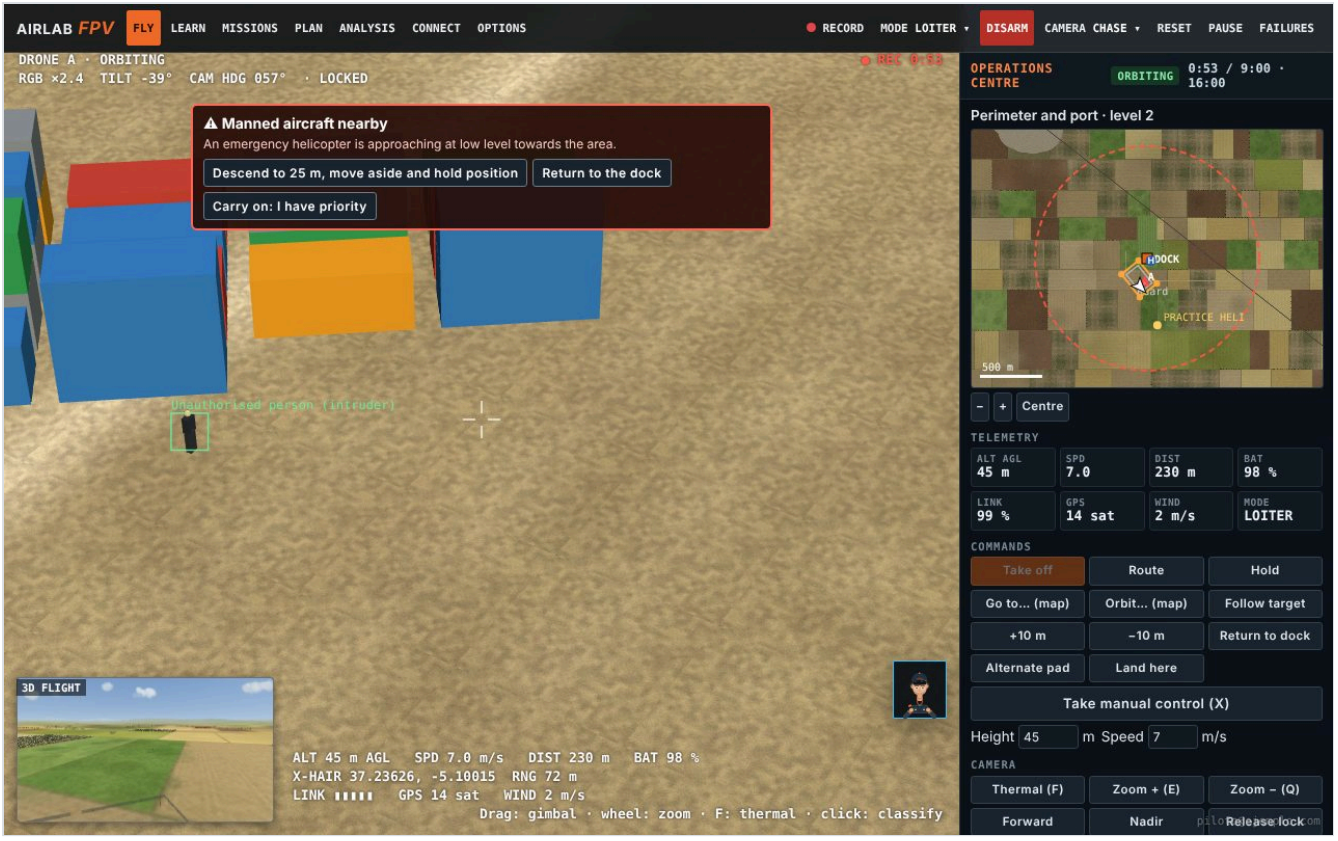
- **Service**: Perimeter and port, Wildfire, Mountain rescue, Coast and border or Flood (DANA).
- **Level** 1 to 5: at levels 1 and 2 an «AI» assist marks things in the video that could be something (dashed blue box); from level 3 there is no assist and there are more contingencies and more targets.
- **Site**: the recommended one (with real terrain: the port of Algeciras, Sierra de Grazalema, Sierra de Cazorla, Tarifa or Paiporta) or the scenery you are in.
- **Drones**: one, or two from level 4 (drone B is virtual, with its own dock and camera). **Rounds**: 1 to 3, with a battery swap and a time change between rounds (the last one at night if the scenery allows it).

39.2 Briefing and checklist

Before deploying you read the mission and go through the checklist: weather (wind and gusts against the drone limit, 12 m/s), NOTAMs of the day, UAS geographical zone, geofence and ceiling (120 m AGL), C2 link, GPS and battery, and the alternate landing point. For each item you decide whether it is fine or an issue that needs coordination. This gives up to 7 of the 10 preparation points.

39.3 Flight plan

The console opens on the right with drone A's plan: route templates for the service (perimeter, orbit, sweep, coastline...), **Edit points on the map** (drag, click to add, double-click to remove), loops, height and speed. Below you see the estimate: route time, endurance and **reserve**; below 25 % it turns red. If a point is outside the geofence, it warns you too. **Plan ready: to the console** closes the planning (3 more points if the plan is correct).



Operations centre in automatic: drone video big, 3D flight small and the console with map, telemetry and commands.

39.4 The console

- **Map**: geofence (dashed red), route, docks, alternate pad (blue H), the drone with its camera footprint, detections, teams on their way and air traffic. Wheel or −/+ to zoom, drag to move it, **Centre** to follow the drone.
- **Telemetry**: height above ground, speed, distance to the dock, battery, link, GPS, wind and mode.
- **Commands**: Take off, Route, Hold, Go to... and Orbit... (click on the map), Follow target, +10 m / −10 m, Return to dock, Alternate pad, Land here and Take manual control. With two drones, **Drone A/B** switches which one you command and whose camera you see.
- **Camera**, **Alerts** (the contingencies, with their options), **Detections** (with the photo) and **Log**. Alerts also appear over the video.

The drone flies on top of ArduCopter's LOITER: the console sends it the point, the height above the terrain (it follows the relief) and the heading. On take-off the dock lid opens; on return it descends slowly over the dock, disarms and the lid closes.

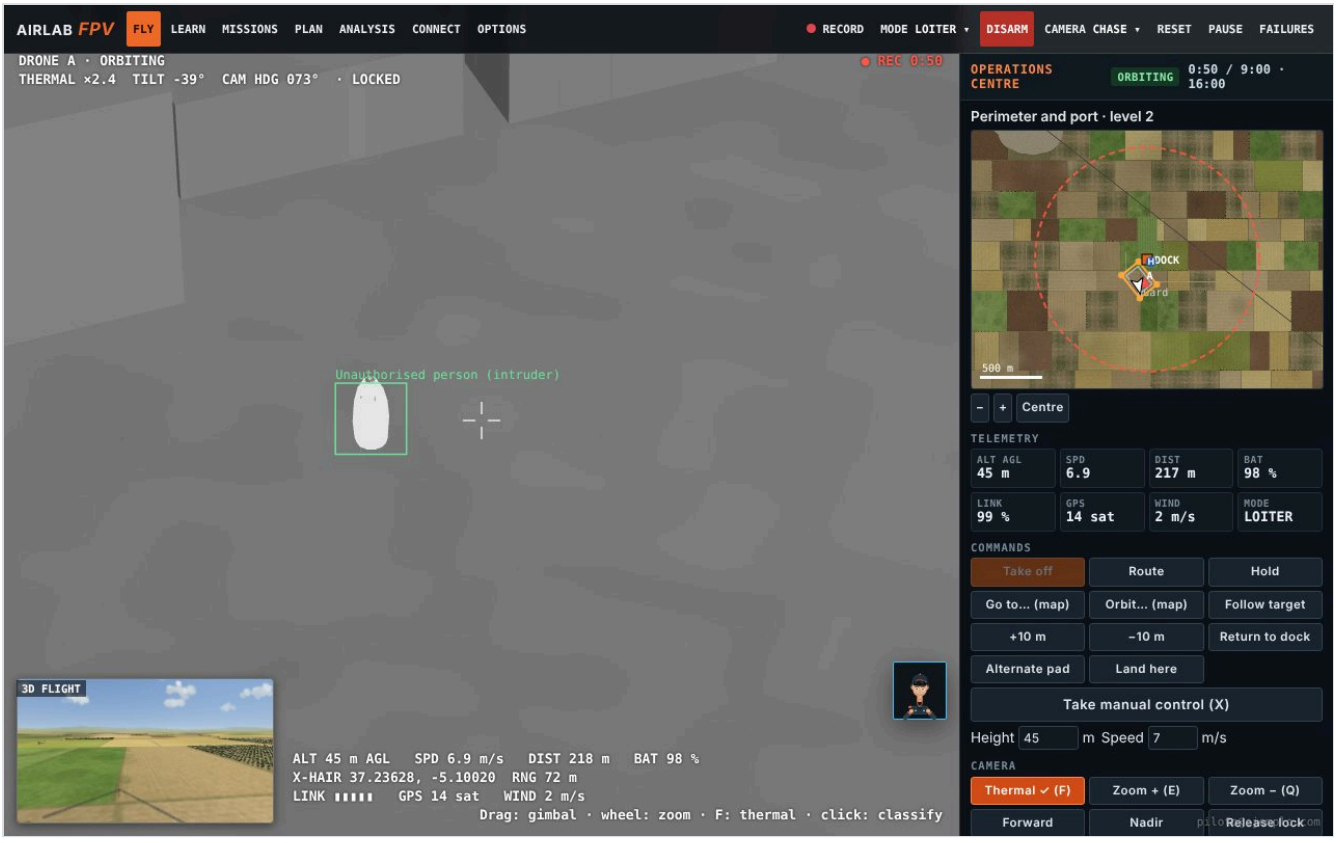
39.5 The drone camera

- **Drag** moves the gimbal (it follows the nose with the pan you give it); **wheel** or **Q**/**E** to zoom (×20 in RGB, ×8 in thermal); **F** switches to **thermal**, where hot things show white (people, animals, engines, fire) and smoke does not hide them.
- **Camera joystick** (in the console **Camera** section): **on screen**, a circle over the video you drag with the mouse or a finger, with + / − to zoom; **second gamepad** (a game controller apart from the radio: left stick to move, right stick or triggers to zoom, A/✕ thermal, B/○ lock); or **radio sticks in automatic** (aileron and elevator move the camera and rudder zooms; the sticks then no longer switch to manual:

use X, the button or the radio mode switch). With a radio connected, this is the starting mode. Speed drops with zoom for fine aiming.

- **Forward**, **Nadir** (straight down) and **Lock on crosshair**: the camera keeps looking at that point even if the drone moves.
- **Click on something**: if it is big enough, you choose what it is and what you do (whom you alert). If it is small it asks you to get closer or zoom in. **Click on the ground**: look, go or orbit there, or mark a finding.

When you alert a service, its team sets off towards the target and you see it arrive (guard, patrol, firefighting helicopter, rescue team, Salvamar, firefighters' boat). If the order was to follow it, keep it on camera until they arrive: it counts for the score.



Thermal camera: the person shows white; the green box is a target already classified.

39.6 Manual control

Take manual control, the **X** key, flipping the radio mode switch or moving the sticks decisively (more than half travel, when they are not driving the camera) hand the drone to your sticks (in LOITER, or ALT_HOLD without GPS). Then the view swaps: the **3D flight big** with the OSD and the **drone camera small**, top right, where you can keep moving the gimbal and classifying. **Back to automatic** (or X) leaves the drone holding position and returns to the camera view.



Manual control: 3D flight big and the drone camera small.

39.7 Contingencies

Alert	What happens	What is expected
C2 link degraded	Signal drops and the video gets noisy; if it is lost, the video freezes and the drone returns to the dock on its own (failsafe). The link depends on distance and on the real line of sight between the dock antenna and the drone.	Climb to improve line of sight; if it does not improve, return.
Strong wind gust	11 m/s with gusts above the limit.	Suspend and return to the dock.
Abnormal battery drain	One cell sags: the battery drops much faster.	Return now (at 22 % it returns on its own).
Manned aircraft nearby	An emergency helicopter at low level.	Descend, move aside and wait: manned aviation always has priority.
GPS loss	The drone switches to ALT_HOLD and drifts with the wind; no guided flight.	Take manual control and bring it within view of the dock.
Request outside the geofence	The command post asks to follow a target leaving the area.	Do not leave: report last position and heading.

Thermal camera failure	The thermal camera stops giving an image.	Continue with RGB closer, or return if the task needs thermal.
The dock does not open	On return the lid does not open and the drone waits over it.	Land at the alternate pad (blue H).

Each alert gives you 45 s to decide. There are also automatic protections: geofence (leaving it sends the drone back to the dock), 120 m ceiling and battery return at 22 %.

39.8 The services

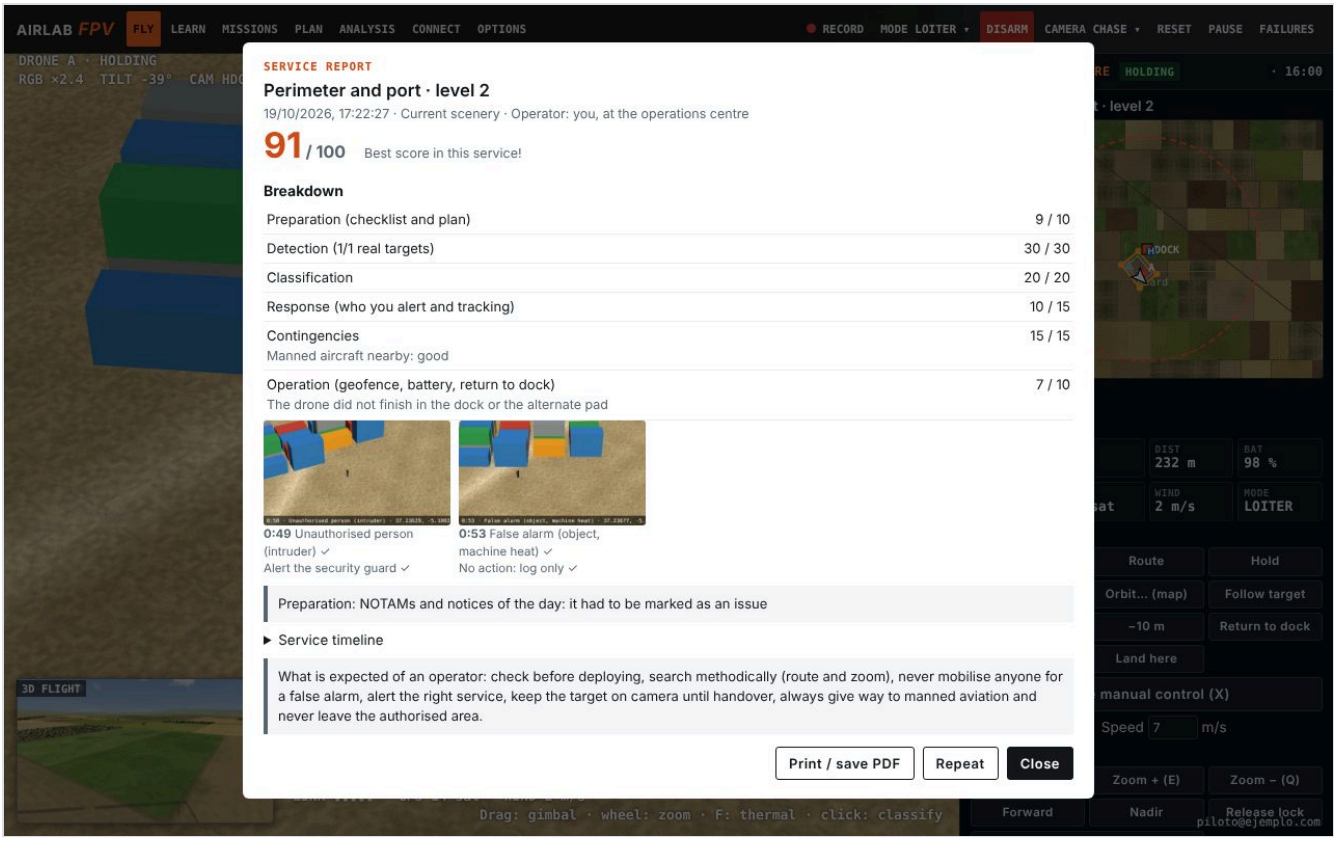
Service	What you look for	What it is confused with	Whom you alert
Perimeter and port	Intruders getting in through a gap in the fence; at level 4, a van by the fence.	Guard in a vest, fox, dog, hot transformer.	Security guard (or Guardia Civil if a vehicle is involved).
Wildfire	The fire seat (it grows over time); a person near the fire.	Sun-heated rocks, goats, an authorised agricultural burn (that one goes to the environmental officer).	112 / forest firefighters. When the helicopter arrives, move more than 300 m away or land.
Mountain rescue	Injured hiker among trees; his companion.	Deer, ibex, hot rocks, a camping group.	Rescue team (112); orbit over the victim until it arrives.
Coast and border	Inflatable with people; person in the water; landed group.	Fishing boat, surfers, buoy.	Salvamento Marítimo first (and Guardia Civil): human life at sea comes first.
Flood (DANA)	Person on a car roof, person in the water, people on roofs.	Empty cars, a dog on a roof, floating debris.	Firefighters / UME, by priority: water and cars before roofs.

39.9 Score and service report

Item	Points
Preparation (checklist and plan)	10
Detection of the real targets	30
Classification	20
Response (whom you alert, tracking until handover; −3 per false alarm)	15
Contingencies	15
Operation (geofence, battery, return to dock)	10

A crash caps the score at 30. When you finish (back in the dock in the last round or with **End service**) the **service report** appears: score, breakdown by round, what you missed, the photos of each detection with its classification and what it really was, and the timeline. It can be printed or saved as PDF. The best

scores count for the **Drone-in-a-box operator** course in the logbook: three different services at 70 or more, one at level 4 or 5 at 70 or more, six contingencies handled correctly and 1 h of flight.



Service report with the score, the breakdown and the photos of the detections.

> **How it is in reality**
>
> Drone-in-a-box flights are usually BVLOS and fall in the specific category (standard scenario or AESA authorisation with a risk assessment), with a registered operator, trained pilots and emergency procedures. The functions of this operations centre are generic and for training: they do not reproduce any specific commercial system.

40 Setting up the Stallion VTOL in ArduPilot, parameter by parameter

This chapter is a guide to configuring the **real Flightory Stallion VTOL** (a tilt-rotor tricopter with vectored yaw) in ArduPlane, assuming nothing. For each parameter it explains what it does internally, with the formulas from the code itself where they matter; its default value; what Flightory suggests; and what the example Stallion uses. That Stallion is the author's: CoreWing F405 Wing V2, 6S2P 8000 mAh Li-ion battery, Emax ECO II 2807 1300 KV motors with 7" props, Flycolor Raptor5 45 A ESCs, ELRS receiver, Foxeer M10Q-250 GPS, MS4525DO pitot and DJI O4 Pro. It is meant to be read with QGroundControl open.

> **Where each fact comes from**
>
> Definitions, ranges, units and defaults come from the **ArduPlane 4.7.2** source (the current stable release; the wiki parameter list is generated from it) and from the official ArduPilot Plane wiki. The Stallion values come from the Flightory manuals (Stallion V2, VTOL PACK and "ArduPilot config for a 4+1 QuadPlane") and from the author's parameter file up to his first hover test flight. Estimates and personal criteria are labelled as such. **Before flying, always compare with the file on your flight controller.**

> **A badly configured VTOL breaks in seconds**
>
> Configure with the props off, check each block on the bench before moving to the next, and fly in phases: hover first, then transition. The simulator (chapter 10 and chapter 27, SITL) lets you rehearse most of this first.

40.1 Working with parameters in QGroundControl

- **Where:** Vehicle Setup (gear) › **Parameters**. The search box filters by name; each parameter shows its description, units and range.
- **Loading a file:** Tools › **Load from file for review…** shows which parameters will change before writing. Always review it: someone else's .param can carry accelerometer or compass calibrations from another aircraft.
- **Reboots:** some parameters only apply at boot: `Q_ENABLE`, `Q_FRAME_CLASS`, `AHRS_ORIENTATION`, `SERIALx_*`, `SCHED_LOOP_RATE` and the DShot output ones. Use Tools › **Reboot Vehicle** after changing them.
- **Parameters that appear when others are enabled:** the `Q_*` ones don't exist until `Q_ENABLE = 1` and a reboot; the `Q_TILT_*` ones, until `Q_TILT_ENABLE = 1`. That is why setup takes several passes.
- **Renamed parameters:** ArduPlane renames parameters between versions and converts the values automatically. In 4.7 they include `ROLL_LIMIT_DEG` (formerly `LIM_ROLL_CD`, in centidegrees), `AIRSPEED_CRUISE` (formerly `TRIM_ARSPD_CM`), `ARMING_SKIPCHK` (replaces `ARMING_CHECK`), `Q_A_ANGLE_MAX` (formerly `Q_ANGLE_MAX`), `Q_WP_SPD` (formerly `Q_WP_SPEED`, in cm/s), `Q_LOIT_SPEED_MS` and `MIN_GROUNDSPEED`. If a tutorial uses an old name, look for the new one.
- **Keep copies:** Tools › **Save to file** after every session. It is your change history.

Reading a bitmask

Many parameters (`Q_OPTIONS` , `RC_OPTIONS` , `ARMING_SKIPCHK` , `SERVO_BLH_*MASK` …) are a sum of powers of two: each option is a bit and the value is $\Sigma\ 2^{bit}$ of the options you enable. QGroundControl usually shows them as check boxes, but it pays to be able to decompose them when reading a file:

A bitmask is the sum of 2^bit for every enabled option

Q_OPTIONS = 163841

bit 0	2^0 = 1	Level Transition: wings level (LEVEL_ROLL_LIMIT)
bit 15	2^15 = 32768	ThrLandControl: throttle sets the landing rate
bit 17	2^17 = 131072	EnableLandReposition: reposition while landing
	Σ = 163841	

RC_OPTIONS = 8480

bit 5	2^5 = 32	Arming check: throttle at 0
bit 8	2^8 = 256	CRSF telemetry passthrough (Yaapu)
bit 13	2^13 = 8192	ELRS at 420 kbaud
	Σ = 8480	

ARMING_SKIPCHK = 1024

bit 10	2^10 = 1024	Skip "Logging Available" (no microSD)
	Σ = 1024	

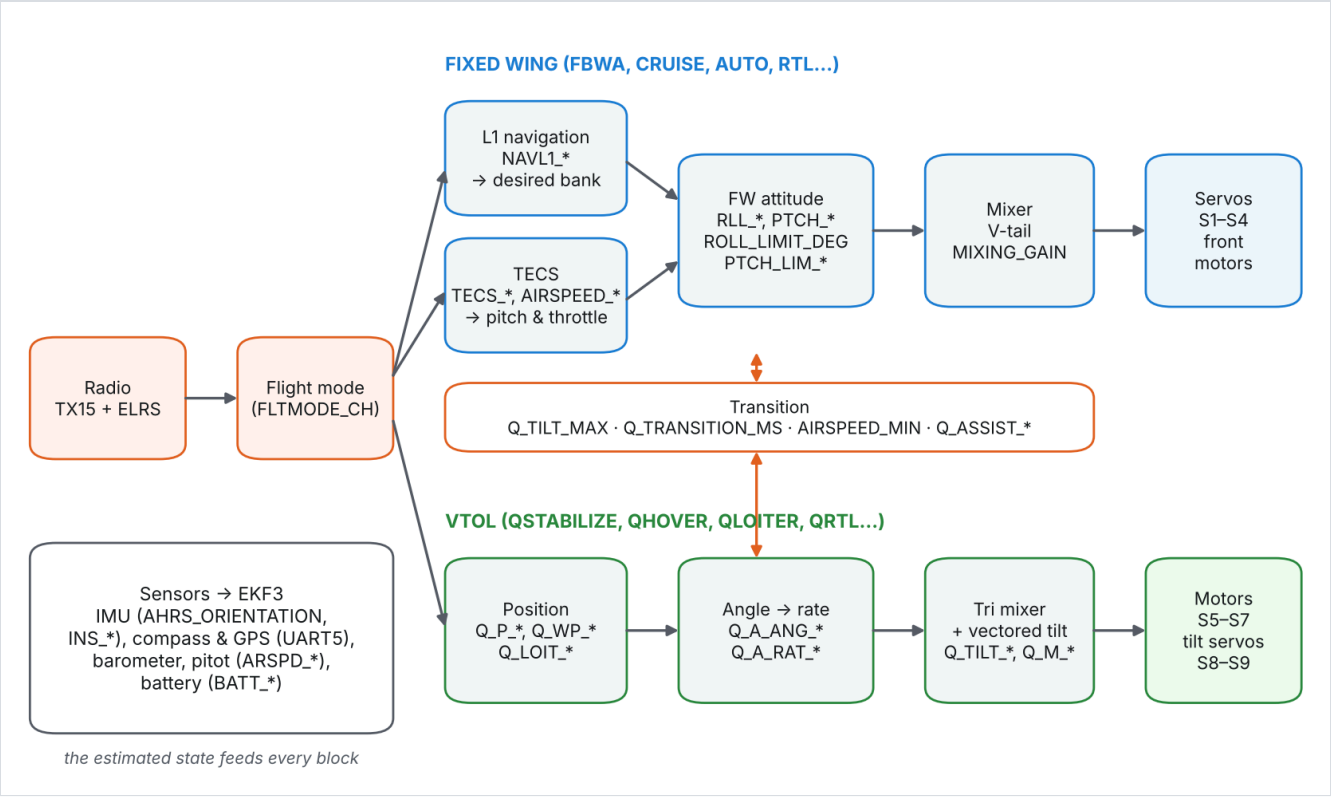
SERVO_BLH_RVMASK = 64

bit 6	2^6 = 64	Reverse the channel 7 ESC (S7, rear motor)
	Σ = 64	

The Stallion's four bitmasks, decomposed. The meanings are the official ArduPlane 4.7 definition of each bit.

40.2 How QuadPlane control is organised

ArduPlane treats the Stallion as a **tilt-rotor QuadPlane**: two control stacks living side by side. In the aeroplane modes (FBWA, CRUISE, AUTO, RTL…), L1 navigation computes the desired bank and TECS computes pitch and throttle from the energy balance. Those demands go to the fixed-wing attitude controllers and the V-tail mixer. In the Q modes (QSTABILIZE, QHOVER, QLOITER…), the multicopter controller takes over: position, angle, rate and motor mixing, with differential tilt for yaw. The transition is the bridge between the two, and both work at once during it.

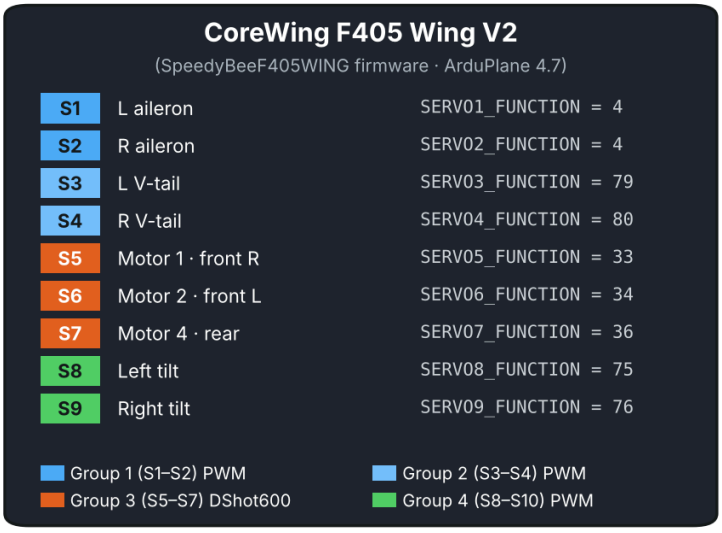


Control blocks and the parameter families that affect each one.

Knowing which block a parameter belongs to tells you when it acts. For example, `ROLL_LIMIT_DEG` has no effect in QHOVER, and `Q_A_ANGLE_MAX` none in FBWA.

40.3 Firmware, board and serial ports

The CoreWing F405 Wing V2 runs the **SpeedyBee F405 Wing** firmware (there is no dedicated build; Flightory has tested it and it works). Flash `arduplane_with_bl.hex` from the stable release on firmware.ardupilot.org in DFU mode with STM32CubeProgrammer. What differs between the two boards is the UART order, fixed with `SERIALx_PROTOCOL`.



Serial ports (UART)

UART1 · SERIAL1
MAVLink2 (2) · telemetry

UART2 · SERIAL2
free (−1)

UART3 · SERIAL3
DisplayPort (42) · DJI O4 Pro, 115200

UART4 · SERIAL4
free (−1)

UART5 · SERIAL5
GPS (5) · Foxeer M10Q-250

UART6 · SERIAL6
RCIN (23) · ELRS RP4TD receiver

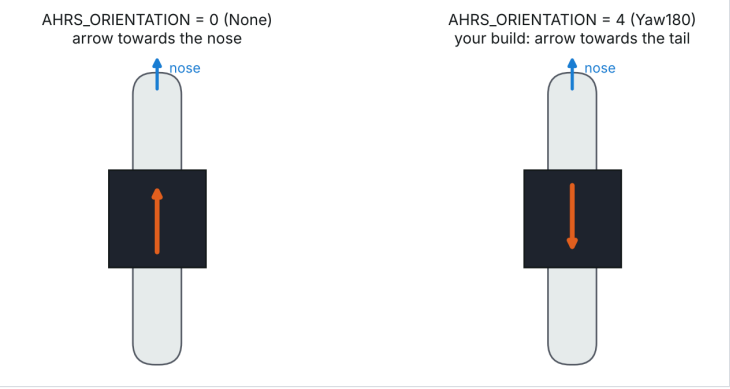
I²C (GPS pads): QMC5883 compass and MS4525DO sensor

Outputs, timer groups and serial ports of the author's Stallion.

Parameter	What it does	Default	Stallion
`SERIALx_PROTOCOL`	What each UART speaks. Useful values: 2 MAVLink2, 5 GPS, 23 RCIN (receiver), 42 DisplayPort (digital OSD), −1 none.	board specific	UART1 = 2 · UART3 = 42 · UART5 = 5 · UART6 = 23
`SERIALx_BAUD`	Port speed in thousands of baud (115 = 115200). An ELRS receiver on RCIN negotiates by itself; DisplayPort runs at 115.	57 (38 on SERIAL3, 115 on SERIAL6)	`SERIAL3_BAUD` = 115
`BRD_ALT_CONFIG`	Alternative board configuration. On this family, 1 frees UART2 (meant for SBUS) as a normal UART.	0	0
`GPS1_TYPE`	GPS type. 1 = AUTO detects the u-blox M10.	1	1
`COMPASS_EXTERNAL`	Tells ArduPilot the compass is off-board (in the GPS). With an external compass its orientation (`COMPASS_ORIENT`) is independent of the board's.	auto-detected	1

> **If you use the ELRS link in MAVLink mode**
> With the WiFi telemetry of chapter 26, the receiver UART changes from RCIN (23) to MAVLink2 (2), and `RSSI_TYPE` and the message rates change too. That chapter has the full table; don't mix the two setups.

40.4 Orientation, inertial sensors and levelling



On the author's Stallion the board is rotated 180° in yaw (arrow towards the tail), so AHRS_ORIENTATION = 4.

Parameter	What it does	Default	Stallion
`AHRS_ORIENTATION`	Board rotation relative to its standard position (arrow to the nose, components up). Rotates the IMU and internal compass readings. Applies at boot and **requires recalibrating the accelerometer and levelling again**.	0 (None)	4 (Yaw180)
`AHRS_TRIM_X` , `AHRS_TRIM_Y`	Roll and pitch difference (rad, ±0.1745) between board and airframe. Written by the "Level Horizon" calibration with the aircraft in its level-flight attitude. Positive Y = nose up.	0	from your levelling

`INS_GYRO_FILTER`	Cut-off frequency (Hz) of the gyro low-pass filter. Lower = less noise but more phase lag in the rate loop. The wiki gives starting values by prop size: 80 Hz (5"), 40 Hz (10"), 20 Hz (20").	20	60 (interpolated for 7")
`INS_ACCEL_FILTER`	The same for the accelerometer. The wiki recommends 10 to 20 Hz.	20	20
`INS_LOG_BAT_MASK`	Which IMU logs batch samples, used to view the vibration spectrum and set up the notch filter. 1 = IMU1. Needs a microSD card.	0	1

Procedure

1. **Check the orientation before calibrating.** With `AHRS_ORIENTATION` set and rebooted, watch the QGroundControl horizon (or MAVLink Inspector, ATTITUDE): nose down must give negative pitch and right wing down positive roll. If an axis is swapped, the orientation is wrong.
2. **Calibrate the accelerometer** (Sensors › Accelerometer) in all six positions, with the board already in the aircraft.
3. **Level** (Level Horizon) with the aircraft resting in its cruise attitude: wing at 0–2° angle of attack, where the Flightory manual places the E205's best efficiency.
4. **Calibrate the compass** outdoors, away from metal, with the GPS in its final position and the battery connected if it sits nearby.

40.5 Airspeed sensor (pitot)

On a tilt-rotor the airspeed sensor is not optional in practice: the transition ends when `AIRSPEED_MIN` is passed, and without a pitot ArduPilot uses a **synthetic** airspeed estimated from GPS and wind. ArduPilot computes indicated airspeed (EAS) from the pitot differential pressure:

$$V_{EAS} = \sqrt{(ARSPD_RATIO \cdot \Delta p)} \quad \text{and} \quad V_{TAS} = V_{EAS} \cdot \sqrt{(\rho_0/\rho)}$$

The theoretical ratio would be $2/\rho_0 \approx 1.63\ m^3/kg$; the default of 2 accounts for the typical losses of a real pitot. All ArduPlane speeds (`AIRSPEED_MIN`, `AIRSPEED_CRUISE` …) are **indicated**: on a 40 °C day the stall happens at the same indicated speed, but at a higher ground speed.

Parameter	What it does	Default	Stallion
`ARSPD_TYPE`	Sensor model. 1 = I²C MS4525DO (the PX4 kit one).	board specific	1
`ARSPD_USE`	0 = measures and logs it but **doesn't use it for control**. 1 = uses it in auto-throttle modes (TECS) and in the transition. 2 = only at zero throttle (gliders with the pitot behind the prop).	0	0 until validated in flight; then 1
`ARSPD_RATIO`	Factor in the formula above. Raising it increases indicated airspeed for the same pressure.	2	calibrate
`ARSPD_AUTOCAL`	Adjusts the ratio in flight against GPS and estimated wind; saves every 2 minutes if it changes by more than 5 %. **Must not be left enabled.**	0	1 only on the calibration flight

`ARSPD_OPTIONS`, `ARSPD_WIND_MAX`	Health check: if airspeed and ground speed differ by more than `ARSPD_WIND_MAX`, the sensor is marked unhealthy and, with the default options, stops being used and is re-enabled when it recovers. Protects against a blocked pitot, **but with WIND_MAX = 0 the check is off**.	11 / 0	WIND_MAX ≈ highest expected wind + margin (e.g. 15 m/s)

> **On the author's Stallion**
>
> The static ports sit about 16 mm ahead of a wide nose (the O4 Pro camera), so a systematic error from nose overpressure is likely. Hence the plan: fly with `ARSPD_USE` = 0, compare ARSP.Airspeed with GPS speed in the log on out-and-back legs, do a calibration flight with `ARSPD_AUTOCAL` = 1 (wide circuits in FBWA or LOITER) and, once it agrees, set `ARSPD_USE` = 1. And always remove the "Remove before flight" cover before take-off.

40.6 Battery and power measurement

Parameter	What it does	Default	Stallion
`BATT_MONITOR`	Monitor type. 4 = analog voltage and current (the CoreWing PDB sensor).	board specific	4
`BATT_VOLT_PIN`, `BATT_CURR_PIN`	Analog sensor inputs. On this board, 10 and 11.	board specific	10 and 11
`BATT_VOLT_MULT`	$V_{battery} = V_{pin} \cdot$ VOLT_MULT. Correct it with a multimeter: new = current · $V_{multimeter} / V_{read}$.	board specific	11.05
`BATT_AMP_PERVLT`	$I = (V_{pin} -$ AMP_OFFSET$) \cdot$ AMP_PERVLT. Calibrate with a wattmeter: new = current · $I_{wattmeter} / I_{read}$. In-flight check: after recharging, charger mAh / logged mAh.	board specific	64 (CoreWing manual; Flightory uses 50): **to verify with a wattmeter**
`BATT_AMP_OFFSET`	Sensor voltage at zero current.	0	0
`BATT_CAPACITY`	Capacity (mAh). Reference for the remaining percentage and the mAh thresholds.	3300	8000
`BATT_ARM_VOLT`	Minimum voltage to allow arming. 0 = no limit.	0	optional: 22.8 (3.8 V/cell) prevents taking off on a half-used pack

Voltage compensation and current limit in VTOL

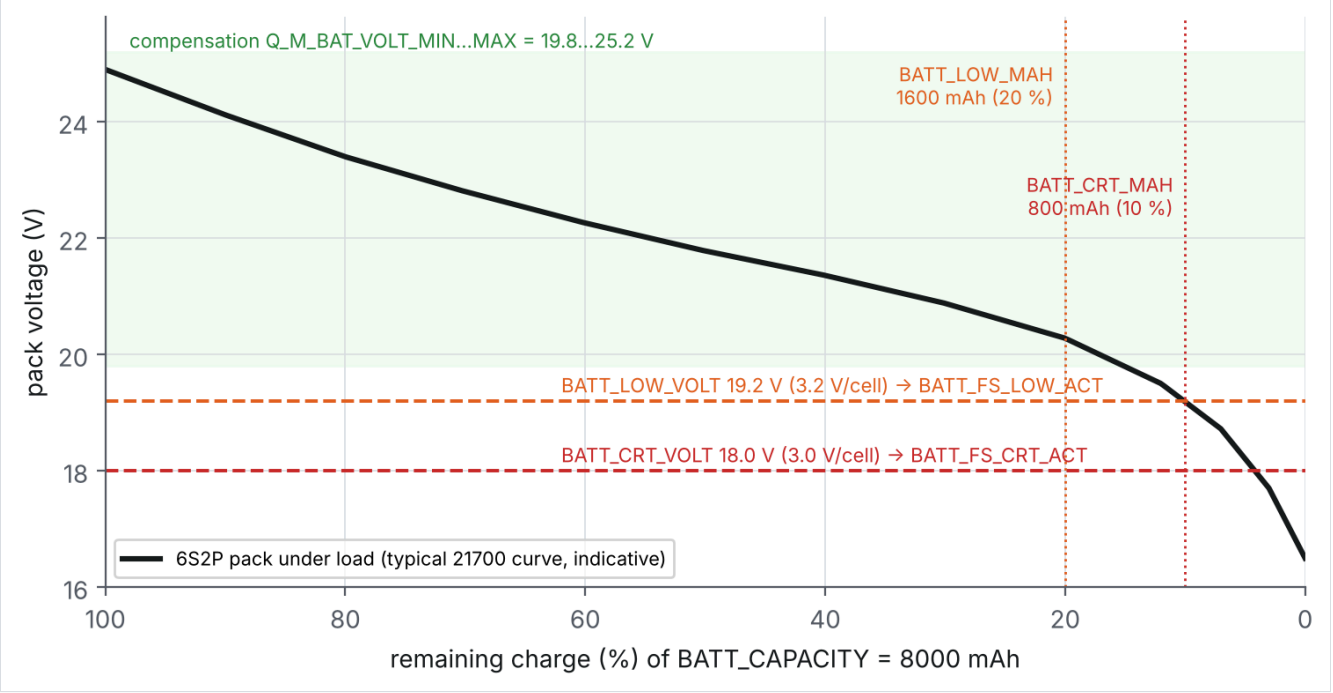
The thrust of a prop at constant command falls with voltage. `Q_M_BAT_VOLT_MAX` and `Q_M_BAT_VOLT_MIN` define the range over which the motor controller scales the command so that the same stick (and the same PID gains) produce the same thrust with a full or nearly empty battery. The official definition recommends 4.2 V and 3.3 V per cell. Below the minimum there is no further compensation.

Parameter	What it does	Default	Stallion
`Q_M_BAT_VOLT_MAX`	Voltage above which there is no extra scaling. 4.2 · cells.	0 (disabled)	25.2 (6S)

`Q_M_BAT_VOLT_MIN`	Voltage below which there is no more scaling. 3.3 · cells.	0	19.8
`Q_M_BAT_CURR_MAX`	Total current above which the maximum throttle of the VTOL motors is limited. 0 = no limit.	0	80 A: 10 C × 8 Ah, the pack's continuous limit

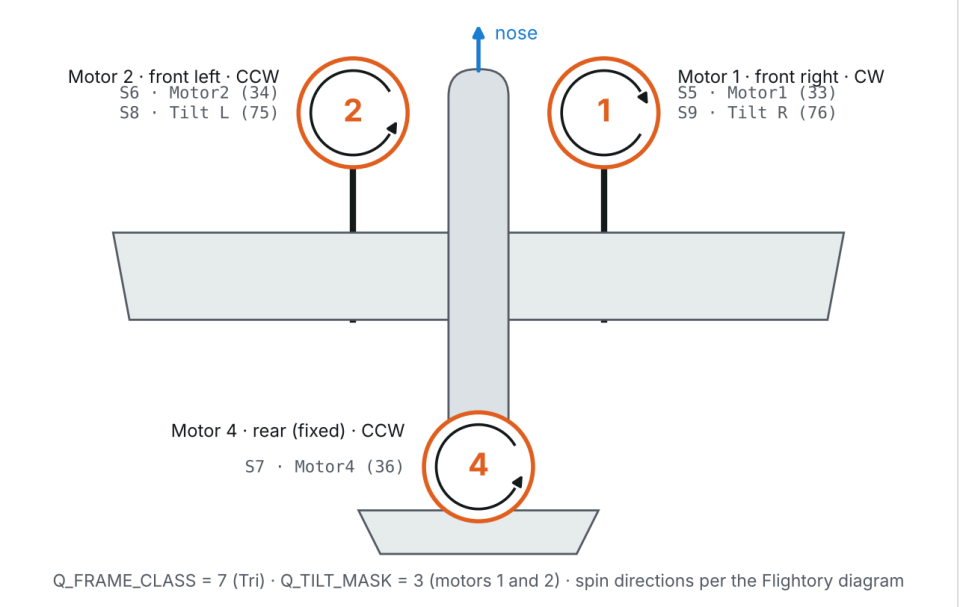
Watch out for 6S

The VTOL PACK manual is designed for 4S. On 6S, the ESCs (3–6S), the board (up to 28 V) and the video (9 V BEC) are within range, and the 1300 KV Emax is sold for 6S, but current with a 7" prop rises a lot. With Emax's figures, hover takes about 28 A in total, while full VTOL throttle would exceed 130 A. Hence the 80 A limit and the wattmeter check: under 40 A per ESC at full throttle for 2–3 s, with the Raptor5's 55 A for 10 s peak as the absolute limit.



Compensation range and failsafe thresholds on a typical curve for a 6S2P pack of 21700 cells. The curve is indicative: the real one depends on cell, temperature and current.

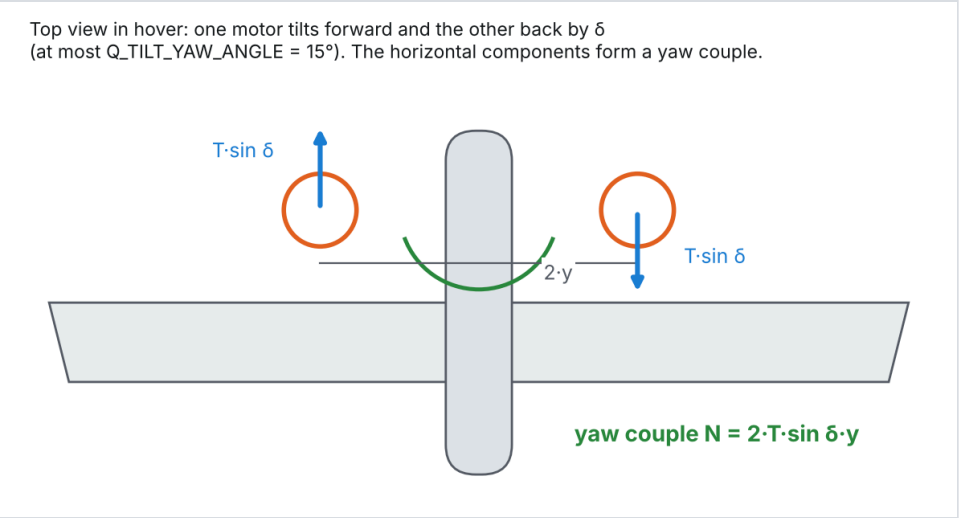
40.7 Enabling the QuadPlane and defining the tricopter



ArduPilot tricopter motor numbering and the Stallion's outputs. Yaw does not come from the rear motor but from the differential tilt of the front ones.

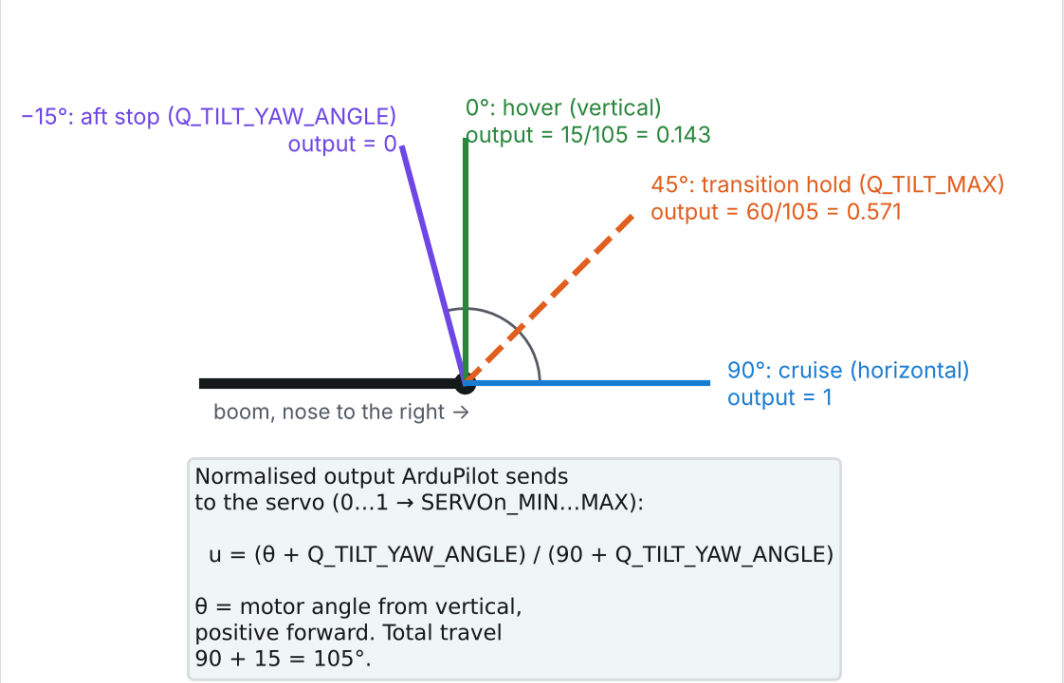
Parameter	What it does	Default	Stallion
`Q_ENABLE`	Enables the QuadPlane. 2 = also starts AUTO in VTOL. Then Write and reboot so the `Q_*` parameters appear.	0	1
`Q_FRAME_CLASS`	Motor layout in hover. 7 = Tri. For Tri and Y6, `Q_FRAME_TYPE` is ignored.	1 (Quad)	7
`Q_TILT_ENABLE`	Enables the tilt-rotor logic.	0	1
`Q_TILT_MASK`	Bitmask of the motors that tilt, in the frame's standard order. Motors 1 and 2 → $2^0 + 2^1 = 3$.	0	3
`Q_TILT_TYPE`	0 continuous · 1 binary (retract servo) · 2 **vectored yaw** (left and right tilt independently) · 3 bicopter.	0	2

> **Why "vectored yaw"**
>
> In a classic tri, yaw comes from a servo tilting the tail motor. On the Stallion the tail motor is fixed and yaw comes from tilting one front motor forward and the other back: the horizontal components of their thrust form a couple. This saves a servo and weight at the tail, at the cost of yaw authority depending on the front motors' thrust and tilt.



Yaw couple in hover: N = 2·T·sin δ·y, with δ ≤ Q_TILT_YAW_ANGLE and y the motors' half-span.

40.8 The tilt mechanism



Tilt angles and the normalised output ArduPilot sends to the servo at each one (Q_TILT_YAW_ANGLE = 15).

Parameter	What it does	Default	Stallion
`Q_TILT_YAW_ANGLE`	How many degrees the motor can go **past vertical, backwards**, which is the servo's minimum output position. With `Q_TILT_TYPE` = 2 it is what enables vectored yaw. Range 0–30°. More angle gives more yaw authority but needs more mechanical travel.	0	15 (Flightory)
`Q_TILT_MAX`	Maximum angle with multicopter control. During the transition wait the motors stay here (if throttle is above 50 %) while the aircraft accelerates. Range 20–80°.	45	45

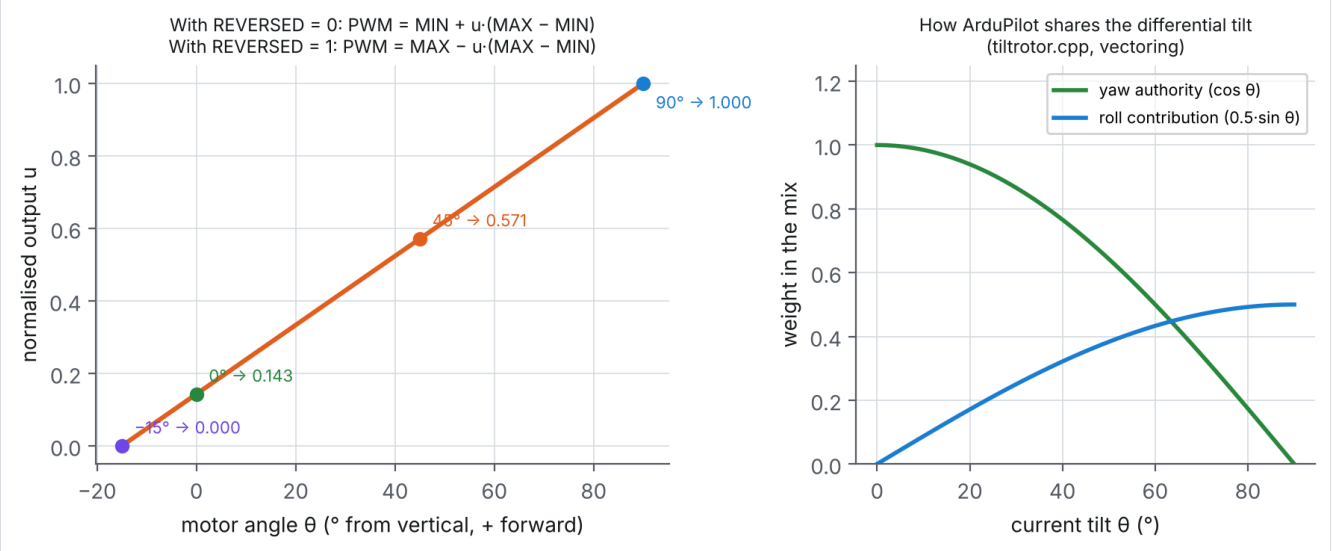
`Q_TILT_RATE_UP`	Maximum tilt rate (°/s) when going from aeroplane back to hover.	40	15
`Q_TILT_RATE_DN`	Rate when going from hover to aeroplane. 0 = uses the up rate. Automatic exceptions: in MANUAL, or armed in aeroplane flight without assistance, at least 90 °/s.	0	0 (15 °/s both ways)
`Q_TILT_FIX_ANGLE`, `Q_TILT_FIX_GAIN`	Use the tilt to help roll and pitch in aeroplane flight. Disabled with 0.	0	0

How ArduPilot turns the angle into PWM

The tilt outputs (`SERVOn_FUNCTION` 75 left and 76 right) move over a normalised range from 0 to 1 covering 90° + `Q_TILT_YAW_ANGLE` . With θ = angle from vertical (positive forward):

`u = (θ + Q_TILT_YAW_ANGLE) / (90 + Q_TILT_YAW_ANGLE)` · `PWM = MIN + u·(MAX − MIN)` (or `MAX − u·(MAX − MIN)` if REVERSED = 1)

In hover, yaw adds to and subtracts from that base value: $u_{left/right} = u_{base} \pm k \cdot u_0$, with u_0 = 15/105 (the margin to the aft stop) and k the yaw demand. That demand is multiplied by cos θ, so it loses effect as the motors tilt forward, and by `Q_M_THST_HOVER` /throttle (limited to 0.5–2), so authority doesn't change with throttle.



Left: normalised output versus angle. Right: weight of yaw and roll in the differential tilt versus tilt angle.

Author's Stallion	MIN	MAX	Hover (u = 0.143) with REV = 0 / 1	Horizontal (u = 1) with REV = 0 / 1
`SERVO8` (left tilt)	1075	2175	1232 / 2017 µs	2175 / 1075 µs
`SERVO9` (right tilt)	900	2025	1060 / 1864 µs	2025 / 900 µs

Bench setup procedure (wiki, Tilt Rotor Setup Tips)

1. Props off. Set `Q_TILT_YAW_ANGLE` first, because it changes the nominal position.
2. With `SERVO8/9_MIN` = `MAX` = 1500, mount the servo arm so the travel from horizontal to vertical is evenly split. Then set MIN/MAX to 1300/1700.
3. Disarmed, switch to **MANUAL** from QGroundControl: the motors must go forward. In **QSTABILIZE** they must go vertical. If one goes the wrong way, reverse it with `SERVOn_REVERSED` .

4. Iterate: adjust the “forward” end in MANUAL and the “vertical” end in QSTABILIZE. In MANUAL both motors must be exactly aligned: any difference is a rolling moment in aeroplane flight. If there are mechanical stops, rest on them lightly so the servo doesn’t overheat.
5. To check the aft stop (yaw), temporarily enable `Q_OPTIONS` bit 10 (Disarmed Yaw Tilt: 163841 + 1024 = 164865) and move the rudder fully in QSTABILIZE, disarmed: each servo must reach its end without straining. **Remove the bit afterwards.**
6. Optional: `Q_TILT_RATE_UP` = 2 °/s temporarily (below the documented 10–300 range, for this test only) to see whether both move in sync (in FBWA, not MANUAL, which forces 90 °/s). Restore the value afterwards.

40.9 Outputs, servos and ESCs

Output	Item	`SERVOn_FUNCTION`	Notes
S1, S2	Left and right ailerons	4 (Aileron)	One is usually reversed.
S3, S4	Left and right V-tail	79 (VTailLeft), 80 (VTailRight)	ArduPilot does the mixing; never mix on the radio.
S5, S6, S7	Motors 1, 2 and 4	33, 34, 36	The tri uses motors 1, 2 and 4 (not 3). All three in group 3 (DShot).
S8, S9	Left and right tilt	75, 76	Group 4, PWM.

Outputs come in **timer groups** that share a protocol: S1–S2, S3–S4, S5–S7, S8–S10 (PWM or DShot) and S11–S12 (PWM only). Enabling DShot for the motors switches their whole group to DShot. That is why the three ESCs sit together on S5–S7 and no servo shares their group. ESCs connect signal and ground only: the red wire doesn’t go to the VX pin, which carries the servo BEC.

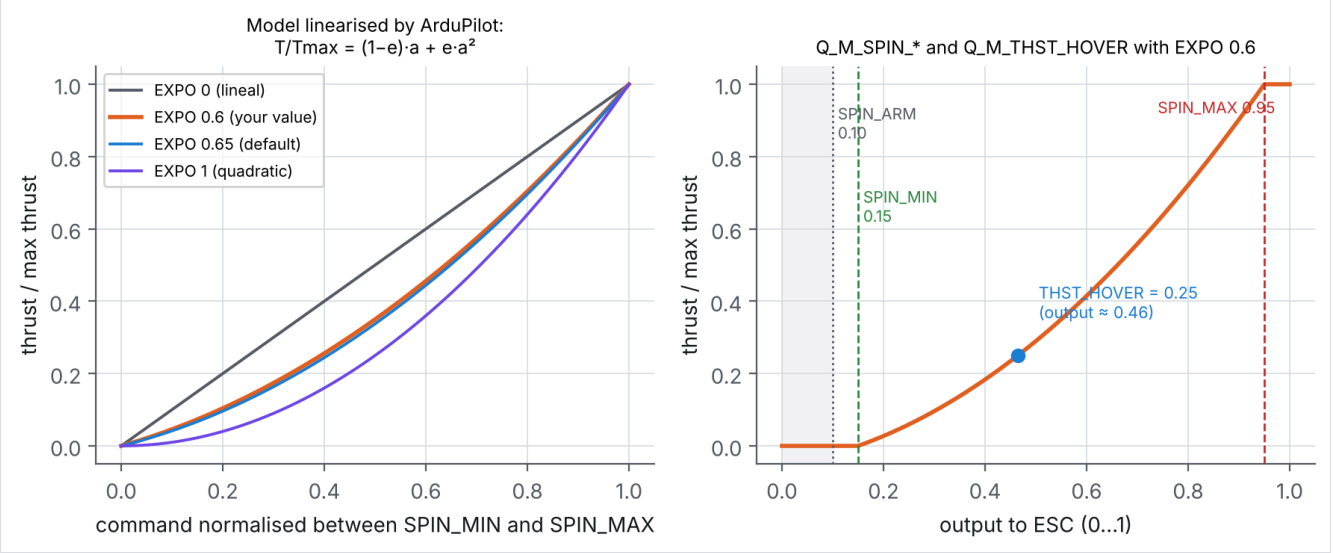
Parameter	What it does	Default	Stallion
`SERVOn_MIN/MAX/TRIM`	Minimum, maximum and neutral pulse width (µs). On surfaces, TRIM is the mechanical neutral; on the tilts, the range sets the ends (40.8).	1100/1900/1500	from your setup
`SERVOn_REVERSED`	Reverses the output direction.	0	from the bench test (Flightory starts with S1, S2, S4 and S8 reversed)
`Q_M_PWM_TYPE`	VTOL motor protocol: 0 PWM, 4/5/6/7 DShot150/300/600/1200. With DShot **no ESC calibration is needed** and the command arrives without drift.	0	6 (DShot600, the one in Flightory’s guide for this board)
`SERVO_DSHOT_ESC`	ESC type for DShot commands (reverse, beacon…). 1 = BLHeli32/Kiss/AM32.	0	1
`SERVO_BLH_AUTO`	Passthrough to configure the motor ESCs through the board (BLHeliSuite32 or similar).	0	1

`SERVO_BLH_RVMASK`	Bitmask of channels whose ESC reverses direction by DShot command. Bit 6 = channel 7.	0	64 (rear motor on S7)
`SERVO_BLH_MASK`, `SERVO_BLH_OTYPE`	Put DShot on channels that aren't VTOL motors (for example a separate pusher). Not needed on the Stallion.	0	0

ESC low-voltage protection

The Raptor5 can cut power below a set voltage per cell. On a VTOL that is dangerous: if a single ESC cuts in hover, the aircraft becomes unbalanced and ArduPilot doesn't know why. Disable it on all three; ArduPilot's failsafe already protects the battery.

Thrust curve and spin points



Left: the thrust model ArduPilot inverts to linearise the command. Right: where SPIN_ARM, SPIN_MIN, SPIN_MAX and hover throttle fall.

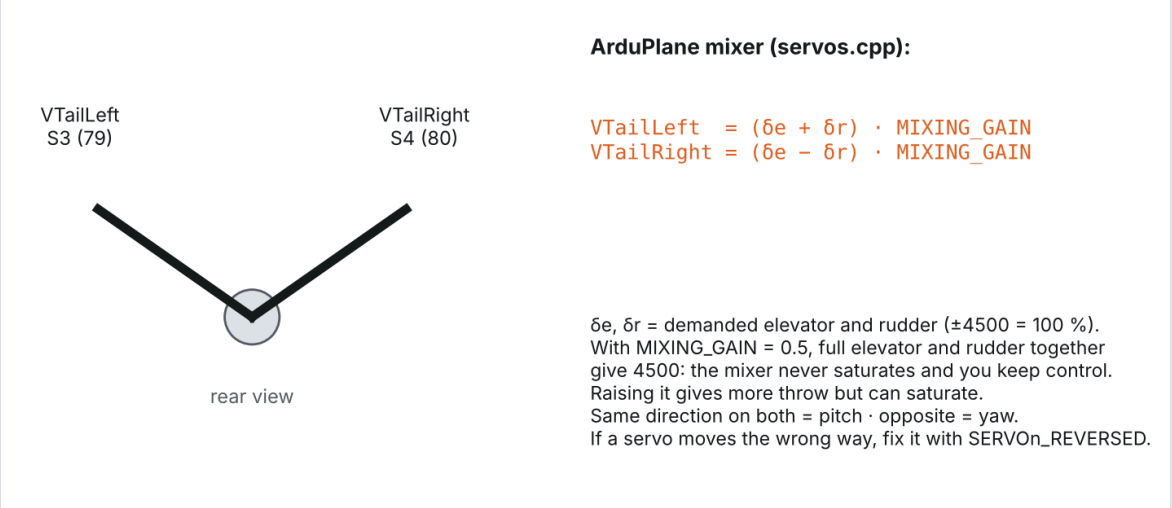
Parameter	What it does	Default	Stallion
`Q_M_THST_EXPO`	Exponent of the thrust-command curve, from 0 (linear) to 1 (quadratic): $T/T_{max} = (1-e)\cdot a + e\cdot a^2$. ArduPilot inverts this curve so the loop sees a linear plant. Wiki: 0.55 (5"), 0.65 (10"), 0.75 (20"); don't exceed 0.75 without thrust-stand data.	0.65	0.6 (7")
`Q_M_SPIN_ARM`	Output (0–1) at which the motors spin when armed, without thrust. Below SPIN_MIN.	0.10	0.10: adjust with the motor test
`Q_M_SPIN_MIN`	Output where useful thrust begins; the controller's "zero" in flight.	0.15	measure with the motor test
`Q_M_SPIN_MAX`	Output where thrust saturates.	0.95	0.95
`Q_M_THST_HOVER`	Thrust (0–1) needed to hover. Reference for the altitude controller and the yaw scaling.	0.35	0.25 to start (wiki), then the learned value

`Q_M_HOVER_LEARN`	Learns THST_HOVER in flight (1) and also saves it (2). It is a filter with a 10 s time constant that only runs in a steady hover: roll and pitch below 5°, vertical speed below 0.6 m/s and no climb demand, and airspeed below 0.3·AIRSPEED_MIN.	2	2
`Q_M_SPOOL_TIME`	Time to go from 0 to SPIN_MIN on arming.	0.25 s	0.25

Reading the learned THST_HOVER

The wiki expects 0.3 to 0.6; higher means too little thrust. As a practical rule: up to 0.45 leaves comfortable margin for gusts and arresting descents; 0.45 to 0.55 flies, but tight; above 0.55, lose weight or add thrust. It matters more with Li-ion, whose voltage sags more under load so available thrust drops late in the flight.

40.10 Control surfaces and V-tail



V-tail mixing as implemented in ArduPlane.

Parameter	What it does	Default	Stallion
`MIXING_GAIN`	Gain of the V-tail (and elevon) mixer. With 0.5, full elevator and rudder together don't saturate. Range 0.5–1.2.	0.5	0.5
`MIXING_OFFSET`	Biases the mix towards one input. Left at 0 for a V-tail.	0	0
`KFF_RDDRMIX`	Rudder added with aileron to counter adverse yaw.	0.5	default; tune in flight
`SERVO_AUTO_TRIM`	Trims the servos in level flight and saves it (at most 20 % from centre).	0	0 until the aircraft is trimmed by hand

Bench check (FBWA, disarmed)

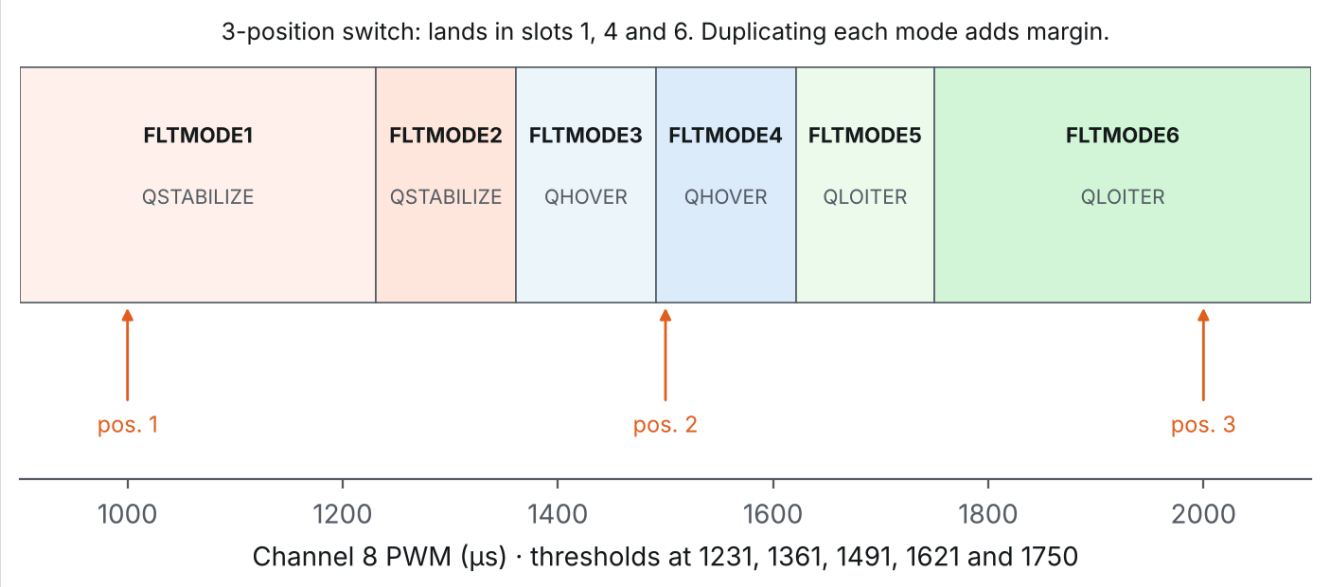
- **Surfaces from the radio** (in MANUAL): roll right → right aileron up, left down. Elevator back → both V surfaces up. Rudder right → the V yaws the nose right (seen from behind, right surface down and left surface up).
- **Surfaces from the stabilisation** (in FBWA, sticks centred): bank the aircraft right → the ailerons must try to raise the right wing. Nose down → the V goes up to raise the nose. If the stabilisation corrects

the wrong way, the fault is in `SERVOn_REVERSED`, not on the radio.

- If a stick is reversed on the radio (for example elevator), fix it with the input channel's `RCn_REVERSED`; on the author's Stallion, `RC2_REVERSED` = 1.

40.11 Radio, flight modes and arming

Parameter	What it does	Default	Stallion
`RCMAP_*`	Which channel is roll, pitch, throttle and yaw. With ELRS in AETR: 1, 2, 3, 4.	1/2/3/4	1/2/3/4
`RC_OPTIONS`	Receiver options (bitmask, figure in 40.1). 8480 = ELRS at 420 kbaud + CRSF telemetry passthrough (Yaapu) + arm only with throttle at 0 (the last one is on by default).	32	8480 (Flightory)
`RC5_OPTION`	Channel 5 function. 153 = ArmDisarm: arm/disarm switch, safer than the rudder gesture.	0	153
`ARMING_RUDDER`	Rudder arming: 0 off · 1 arm only · 2 arm and disarm. ArduPlane won't disarm by rudder while it detects it is flying; the risk with 2 is right after touchdown or if flight detection fails.	1	1
`FLTMODE_CH`	Flight-mode channel.	8	8
`FLTMODE1…6`	Mode for each of the six PWM bands (figure). Numbers: 0 MANUAL, 5 FBWA, 6 FBWB, 7 CRUISE, 10 AUTO, 11 RTL, 12 LOITER, 17 QSTABILIZE, 18 QHOVER, 19 QLOITER, 20 QLAND, 21 QRTL, 22 QAUTOTUNE.	—	phase 1: 17, 17, 18, 18, 19, 19
`ARMING_SKIPCHK`	Pre-arm checks to skip (bitmask). Recommended: 0. 1024 = skip "Logging Available".	0	1024 (no microSD)
`ARMING_REQUIRE`	Requires arming before the motor gets output. Essential on electrics.	1	1



Flight-mode channel bands per the code (RC_Channel::read_6pos_switch) and the phase-1 setup.

> **No microSD, no logs**
>
> Skipping the logging check lets you fly, but without a log you can't diagnose a failure or set up the notch filter (it needs `INS_LOG_BAT_MASK`). An 8–32 GB class 10 card in FAT32 with the default `LOG_BITMASK` (or 65535 for everything) is the cheapest part of the project.

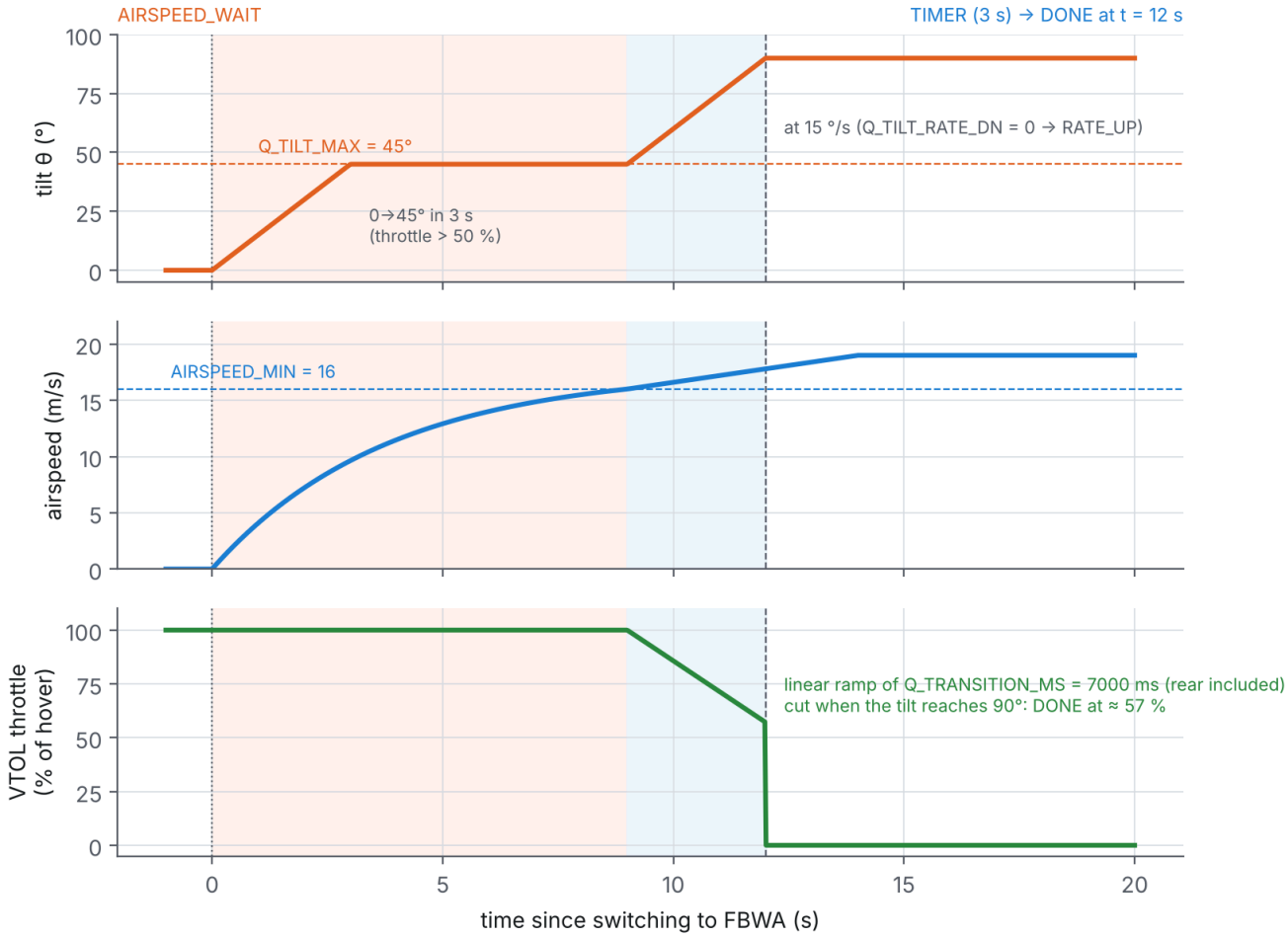
Recommended modes by phase

Phase	3-position switch	Why
1. First hover	QSTABILIZE · QHOVER · QLOITER	No aeroplane mode: passing through FBWA in hover would start tilting the motors.
2. Transitions	QHOVER · FBWA · QLOITER (or QRTL)	QHOVER to FBWA performs the transition, and back to a Q mode the reverse. QSTABILIZE stays available as a recovery from the ground station.
3. Aeroplane and missions	A 6-position switch with QLOITER, QHOVER, FBWA, CRUISE, AUTO, RTL	Flightory's guide suggests MANUAL, QLOITER, QLAND, FBWA, CRUISE and RTL; add AUTOTUNE when it's time to tune.

40.12 The transition, step by step

A tilt-rotor's forward transition goes through three states in the code (SLT_Transition). Understanding them explains almost every parameter in this section:

1. **AIRSPEED_WAIT.** On switching to an aeroplane mode, the front motors tilt to `Q_TILT_MAX` at the tilt rate. In FBWA the angle follows throttle: θ = min(1; 0.02·throttle %)·Q_TILT_MAX, so above 50 % throttle they reach 45°. The rear motor and the vertical component of the front ones keep holding the aircraft up, and the VTOL controller stabilises. It waits until airspeed exceeds `AIRSPEED_MIN`.
2. **TIMER.** Once the speed is reached, the motors go horizontal at the tilt rate and the VTOL motors' throttle (rear motor included) ramps down **linearly** over `Q_TRANSITION_MS`. Meanwhile the fixed-wing controllers take over the surfaces.
3. **DONE.** On a tilt-rotor the transition ends as soon as the motors reach horizontal, **even if `Q_TRANSITION_MS` hasn't elapsed** ("Transition FW done"). The rear motor stops and the front ones work as tractor motors. At 15 °/s, going from 45° to 90° takes 3 s: the 7 s ramp is cut when VTOL throttle is at about 57 % (1 − 3/7). For the ramp to complete, the tilt would have to take at least Q_TRANSITION_MS to cover those 45°.



Timeline of a transition with the Stallion's values. The segment up to AIRSPEED_MIN is illustrative.

Parameter	What it does	Default	Stallion
`AIRSPEED_MIN`	Minimum demanded airspeed in auto-throttle modes and end of the transition wait. Official definition: 20 % above level-flight stall.	9	16 (the manual gives 12 for a Stallion of about 2 kg)
`AIRSPEED_CRUISE`	Target (indicated) airspeed in CRUISE, AUTO, RTL… The Stallion manual gives best efficiency at 60–70 km/h (16.7–19.4 m/s).	12	19
`AIRSPEED_MAX`	Maximum demanded. Official: slightly below level-flight speed at `THR_MAX`, and at least 50 % above AIRSPEED_MIN.	22	35 (Flightory 4+1)
`AIRSPEED_STALL`	1 g stall speed, if known. Refines protection in turns and allows landings slower than AIRSPEED_MIN. 0 = uses AIRSPEED_MIN.	0	0
`Q_TRANSITION_MS`	Duration of the VTOL throttle ramp-down after reaching AIRSPEED_MIN (500–30 000 ms). On a tilt-rotor the ramp is cut short if the tilt reaches horizontal first (see DONE).	5000	7000

`Q_BACKTRANS_MS`	On the way back to VTOL in position-controlled modes, time for the pitch limit to go from 0 to maximum. 0 = no ramp.	3000	5000 (Flightory)
`Q_TRANS_DECEL`	Deceleration (m/s²) used to compute the stopping distance when returning to VTOL: d ≈ V²/(2a). At 19 m/s and 2 m/s², about 90 m.	2	2
`LEVEL_ROLL_LIMIT`	Maximum bank when level flight is wanted: transition with Level Transition, take-off and final landing.	5	5
`Q_OPTIONS`	QuadPlane options (bitmask). Stallion: bit 0 Level Transition (wings within LEVEL_ROLL_LIMIT during the transition; that bit's climb limit doesn't apply to tilt-rotors), bit 15 ThrLandControl (throttle sets the landing rate), bit 17 EnableLandReposition (reposition with the sticks during auto landing).	0	163841
`Q_ASSIST_SPEED`	Below this airspeed the VTOL motors assist in aeroplane modes (except MANUAL and ACRO). 0 = no assistance, with a PreArm warning. −1 = no assistance and no warning, except during the transition. If used: 3 m/s below the minimum you fly at (5 without a pitot).	0	−1 (Flightory)
`Q_ASSIST_ANGLE`, `Q_ASSIST_ALT`, `Q_ASSIST_DELAY`	Assistance on attitude error (more than 5° outside the limits and the error above the angle), on height above home, and its delay. Only active with a positive Q_ASSIST_SPEED.	30° · 0 · 0.5 s	inactive (ASSIST_SPEED = −1)
`Q_TRANS_FAIL`, `Q_TRANS_FAIL_ACT`	Maximum airspeed wait (s, 0 = no limit) and what to do if exceeded: −1 warn, 0 QLAND, 1 QRTL.	0 · 0	0 · 0
`TKOFF_THR_SLEW`	Throttle ramp (%/s) in auto take-off and in the transition.	0	40 (Flightory)

Why AIRSPEED_MIN = 16 on the author's Stallion

With 2.49 kg, S = 26.5 dm² and a wing CL_{max} of 1.0 (estimate for the E205 at Re ≈ $2 \cdot 10^5$), $V_s = \sqrt{(2mg/(\rho \cdot S \cdot CL_{max}))} \approx 12.3$ m/s. The official 20 % margin gives 14.7; 16 m/s leaves a bit more margin for the 94 g/dm² wing loading (the manual allows up to 115) and for the uncalibrated pitot error. Cruise at 19 m/s falls inside the manual's efficiency window.

The first transition

Do it high (at least 30–40 m), into wind, with 150–200 m clear ahead, and touch nothing until you see "Transition done". If anything goes wrong, switch back to QHOVER: the motors return to vertical and you get multicopter control back.

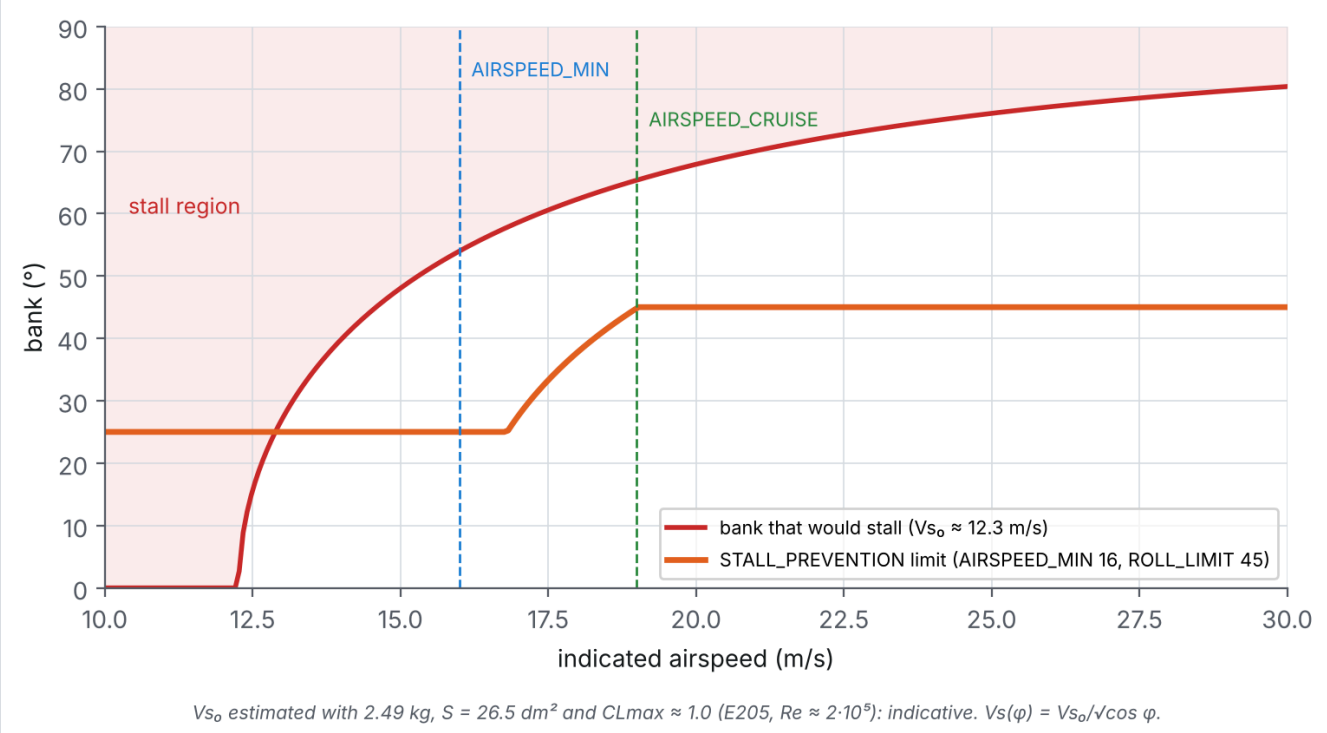
40.13 Aeroplane flight: limits, TECS, stall protection and navigation

Parameter	What it does	Default	Stallion

`ROLL_LIMIT_DEG`	Maximum bank in stabilised and automatic modes.	45	45
`PTCH_LIM_MAX_DEG` , `PTCH_LIM_MIN_DEG`	Maximum pitch up and down.	20 / −25	20 / −25
`STALL_PREVENTION`	Limits bank at low airspeed by load factor, and raises the minimum airspeed in turns (figure). Without a pitot it uses the synthetic airspeed.	1	1
`STAB_PITCH_DOWN`	Degrees of nose-down added in FBWA when throttle is below TRIM_THROTTLE, to keep speed in a glide.	2	2
`TRIM_THROTTLE`	Cruise throttle (%) that holds AIRSPEED_CRUISE. Without an airspeed sensor it is TECS's throttle baseline.	45	45: adjust from the first flight's log
`THR_MAX` , `THR_MIN`	Maximum and minimum forward-motor throttle (%) in every mode except MANUAL.	100 / 0	100 / 0; THR_MAX could cap aeroplane current after the wattmeter test
`THR_SLEWRATE`	Maximum throttle change (%/s).	100	100

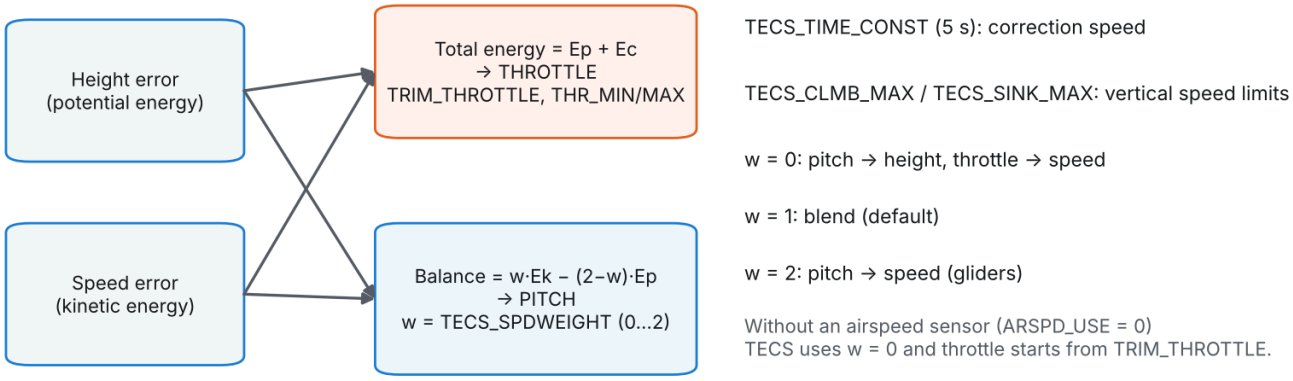
Stall protection (STALL_PREVENTION)

In a coordinated turn the load factor is n = 1/cos φ and the stall speed rises to $V_s \cdot \sqrt{n}$. ArduPlane computes the maximum sustainable load factor as $n_{max} = (V/V_{ref})^2$, with V_{ref} = AIRSPEED_STALL or, if 0, AIRSPEED_MIN. If $n_{max} \le 1$, it limits bank to 25°. Otherwise it limits to $\varphi = \arccos(1/n_{max})$, never below 25° or LEVEL_ROLL_LIMIT. In automatic modes TECS also raises the minimum airspeed to AIRSPEED_MIN·√n.



The Stallion's estimated stall bank versus the limit imposed by STALL_PREVENTION.

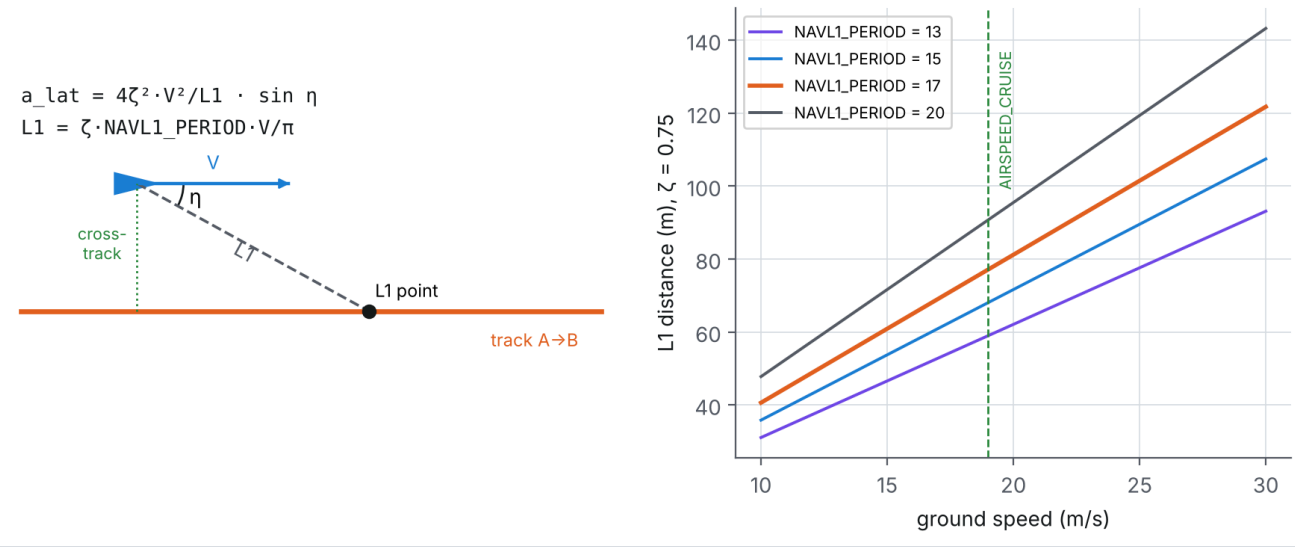
TECS: height and speed



TECS splits energy control: throttle governs total energy and pitch the balance between potential and kinetic.

Parameter	What it does	Default	Stallion
`TECS_TIME_CONST`	Time constant (s) of height and speed corrections. Lower = faster, with overshoot risk. Range 3–10.	5	5
`TECS_SPDWEIGHT`	Speed weighting (0–2): 0 pitch → height, throttle → speed; 2 pitch → speed (gliders); 1 blend.	1	1
`TECS_CLMB_MAX`	Maximum demanded climb (m/s). Official: no more than the climb at THR_MAX and AIRSPEED_CRUISE with a low battery.	5	5: review with the log
`TECS_SINK_MAX`, `TECS_SINK_MIN`	Maximum demanded descent, and sink rate at THR_MIN and AIRSPEED_CRUISE.	5 / 2	5 / 2

L1 navigation and waypoints



L1 geometry and L1 distance versus NAVL1_PERIOD and speed.

Parameter	What it does	Default	Stallion

`NAVL1_PERIOD`	Path controller period (s). Sets L1 = ζ·P·V/π; lateral acceleration is 4ζ²·V²/L1·sin η. Lower = tighter; too low and it weaves. Lower it 1 at a time on an AUTO circuit.	17	17 to start
`NAVL1_DAMPING`	Damping ζ. Rarely changed.	0.75	0.75
`NAVL1_XTRACK_I`	Crosstrack integrator: removes a steady offset.	0.02	0.02
`WP_RADIUS`	Distance at which a waypoint counts as reached (at most the L1 distance).	90	default
`WP_LOITER_RAD`	Radius of LOITER and RTL circles. Negative = anticlockwise. Too small and ROLL_LIMIT_DEG limits it: $r_{min} \approx V^2/(g \cdot \tan \varphi) \approx 37$ m at 19 m/s and 45°.	60	50
`MIN_GROUNDSPEED`	Minimum ground speed in auto-throttle modes (formerly MIN_GNDSPD_CM). In a strong headwind it raises the target airspeed.	0	0

The simulator implements these parameters with the same formulas: it is the place to see their effect before touching them on the aircraft (chapter 17).

40.14 VTOL flight: hover, QLOITER and tuning

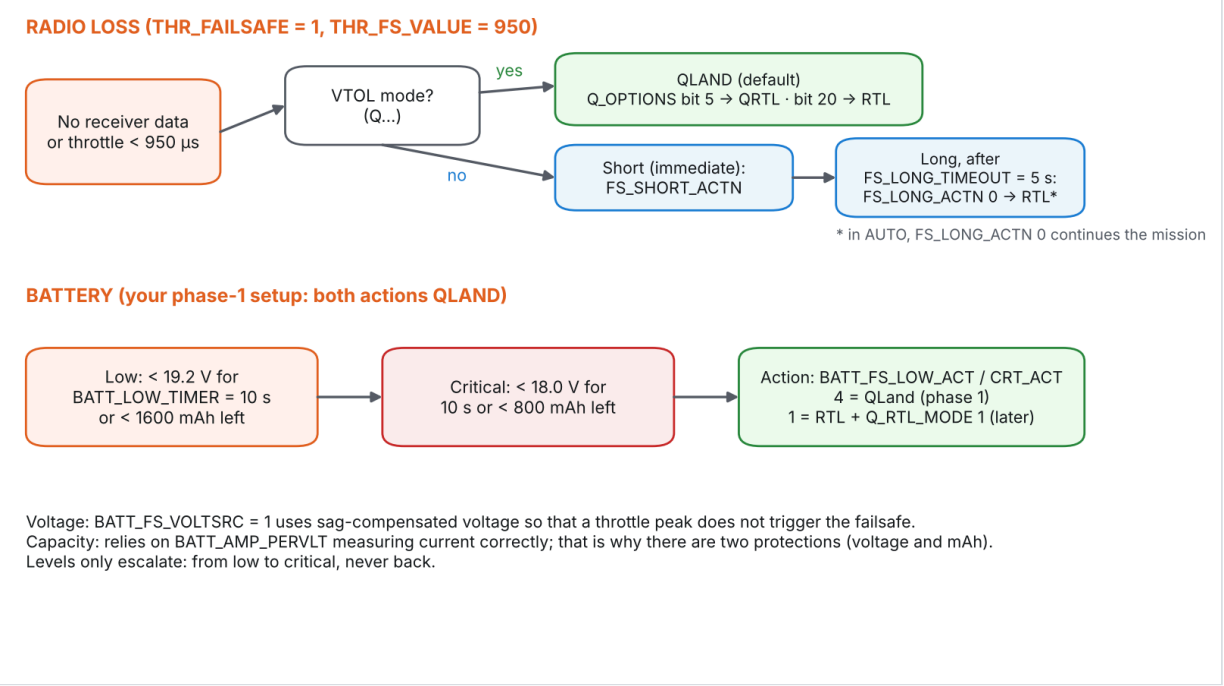
Parameter	What it does	Default	Stallion
`Q_A_ANGLE_MAX`	Maximum lean in VTOL modes (°).	30	30
`Q_PILOT_SPD_UP`, `Q_PILOT_SPD_DN`	Maximum climb and descent on the throttle in QHOVER and QLOITER (m/s). DN = 0 uses the climb value.	2.5 / 0	default
`Q_LOIT_SPEED_MS`	Maximum horizontal speed in QLOITER.	5	5
`Q_WP_SPD`, `Q_WP_SPD_UP`, `Q_WP_SPD_DN`	VTOL navigation speeds (QRTL, VTOL legs of AUTO).	5 / 2.5 / 1.5	default
`Q_LAND_FINAL_ALT`, `Q_LAND_FINAL_SPD`	Height at which the slow final descent starts, and its speed.	6 m / 0.5 m/s	6 / 0.3
`Q_A_RAT_RLL_P/I/D`, `Q_A_RAT_PIT_*`	VTOL rate loop. The QuadPlane defaults roll and pitch P and I to 0.25.	0.25 / 0.25 / —	default until tuned
`Q_A_RAT_YAW_FLTE`	Yaw rate-loop error filter. The wiki says **0 for vectored yaw** (2.5 otherwise).	2.5	0
`Q_A_RAT_*_FLTD`, `FLTT`	Derivative and target filters. Wiki starting value: INS_GYRO_FILTER/2.	FLTD 10 (roll and pitch), 20 (yaw) · FLTT 20	30 with the 60 Hz filter
`SCHED_LOOP_RATE`	Main loop rate (Hz). 300 suits a QuadPlane of this size; 400 only helps aircraft under 1 kg.	300 on a QuadPlane	300
`Q_FWD_THR_USE`, `Q_FWD_THR_GAIN`	Use motor tilt to move forward in QLOITER instead of pitching. On a tilt-rotor the gain ≈ total rotors / tilting rotors = 3/2.	0 / 2.0	default to start

VTOL tuning sequence (wiki, QuadPlane Tuning Process)

1. **Before the first flight:** battery and expo (40.6, 40.9), initial filters (`INS_GYRO_FILTER` , FLTD/FLTT, FLTE = 0) and `Q_M_THST_HOVER` = 0.25. Minimum take-off weight and a full battery; light wind and 15–25 °C.
2. **First hover in QSTABILIZE.** If it oscillates, cut roll and pitch P, I and D by 50 % and repeat. Don't take off in QHOVER until altitude control has been tested in flight.
3. **QHOVER for 30 s** at 5 m with `Q_M_HOVER_LEARN` = 2. On the ground: `Q_P_D_ACC_I` = 0.2·THST_HOVER and `Q_P_D_ACC_P` = 0.1·THST_HOVER (4.7 names).
4. **Notch filter:** 2 minutes of hands-off hover with `INS_LOG_BAT_MASK` = 1, then set up the harmonic notch from the spectrum.
5. **Vectored yaw** (wiki, Tilt Vectored Yaw Tuning): short, sharp rudder inputs, set `Q_A_RAT_YAW_FF` by comparing PIQY.FF+P+D with X·PIQY.Act, set I = FF, then raise D and then P until oscillation and keep 1/2–1/3.
6. **Fine tuning:** manual (D and P in steps) or automatic: QuickTune, built in from 4.6, or QAUTOTUNE one axis at a time (5–8 min each; don't use it with less than 8 min of VTOL battery).

For the fixed wing, **AUTOTUNE** (`AUTOTUNE_LEVEL` 6 recommended) tunes roll and pitch in flight: AUTOTUNE mode on the switch and wide rolls and pitch inputs for a few minutes. On QuadPlanes with a high `INS_GYRO_FILTER` , `AUTOTUNE_OPTIONS` stops it changing FLTD and FLTT.

40.15 Failsafes and return to home

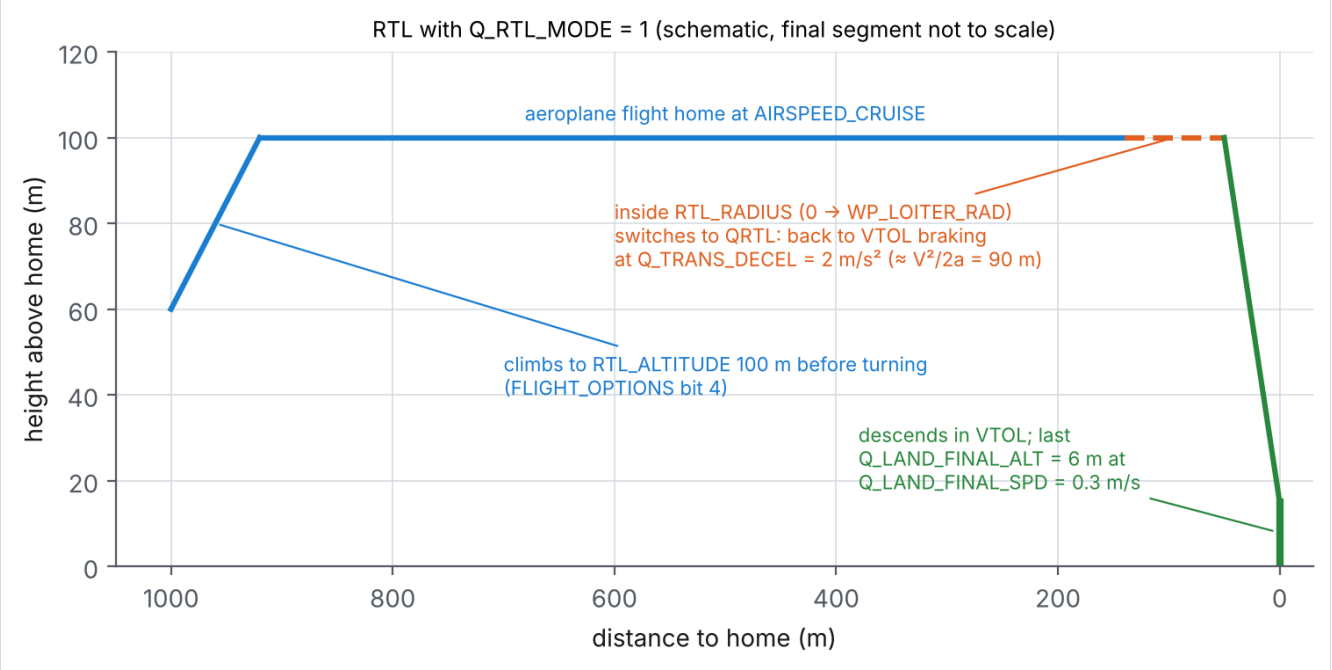


Radio and battery failsafe logic with the phase-1 setup.

Parameter	What it does	Default	Stallion
`THR_FAILSAFE`	1 = failsafe on radio loss (no receiver data, FS bit or throttle below THR_FS_VALUE). 2 = ignores the radio without an action.	1	1

`THR_FS_VALUE`	Throttle PWM below which the signal is considered lost. Well below the normal minimum.	950	950
`FS_SHORT_ACTN`	Short failsafe action in aeroplane modes. In Q modes it goes to QLAND unless Q_OPTIONS bit 5 (QRTL) or 20 (RTL) is set.	0	0
`FS_LONG_ACTN`, `FS_LONG_TIMEOUT`	Action after FS_LONG_TIMEOUT seconds: 0 continue in AUTO or RTL from other modes · 1 RTL · 2 glide…	0 · 5 s	0 · 5 s
`FS_GCS_ENABL`	Failsafe on loss of the ground station (MAVLink heartbeat). On an electric, only with ARMING_REQUIRE.	0	0
`BATT_LOW_VOLT`, `BATT_CRT_VOLT`	Low and critical failsafe voltages; must persist for BATT_LOW_TIMER seconds.	0	19.2 / 18.0 (3.2 and 3.0 V/cell, criterion for Li-ion)
`BATT_LOW_MAH`, `BATT_CRT_MAH`	Remaining capacity for each failsafe.	0	1600 / 800 (20 % and 10 %)
`BATT_FS_LOW_ACT`, `BATT_FS_CRT_ACT`	Actions: 0 warn · 1 RTL · 2 Land · 3 terminate · 4 QLand · 6 Loiter to QLand · 7 AUTOLAND or RTL.	0 / 0	4 / 4 in phase 1
`BATT_FS_VOLTSRC`	0 raw voltage · 1 sag-compensated voltage.	0	1
`BATT_LOW_TIMER`	Seconds below the threshold before triggering.	10	10

RTL and QRTL



Profile of an RTL with Q_RTL_MODE = 1.

Parameter	What it does	Default	Stallion
`RTL_ALTITUDE`	Return height above home (m). −1 = keep current.	100	100

`FLIGHT_OPTIONS`	Mode options (bitmask). Bit 4 = climb to RTL_ALTITUDE before turning for home.	0	16
`RTL_RADIUS`	RTL loiter radius (0 = WP_LOITER_RAD). With Q_RTL_MODE = 1, the radius at which it switches from aeroplane to VTOL to land.	0	0
`Q_RTL_MODE`	0 aeroplane RTL (circles over home) · 1 switch to QRTL inside RTL_RADIUS · 2 spiral VTOL approach · 3 always QRTL.	0	1
`Q_RTL_ALT`, `Q_RTL_ALT_MIN`	QRTL height, and minimum climb when starting in VTOL.	15 / 10	15 / 10
`RTL_AUTOLAND`	Continue the mission landing sequence from a DO_LAND_START (1 after reaching home, 2 directly). With a DO_LAND_START and 0, it won't arm.	0	0
`RTL_CLIMB_MIN`	Metres of initial climb with bank limited to LEVEL_ROLL_LIMIT.	0	0

> **Battery "Land" and aeroplane RTL are not enough**
>
> With `Q_RTL_MODE` = 0 (Flightory's value), a battery RTL ends up circling at 100 m over home until the pack runs out, and the "Land" action without a mission landing sequence ends in RTL. On a VTOL it makes more sense to use RTL with `Q_RTL_MODE` = 1 (fly back as an aeroplane and land vertically) for low battery, and QLand for critical. In phase 1, close in and without a validated transition, both on QLand.

Geofence. `FENCE_ENABLE` = 1 with `FENCE_TYPE` (bits: max altitude, circle, polygons, min altitude), `FENCE_RADIUS`, `FENCE_ALT_MAX` and `FENCE_ACTION` (0 report, 1 RTL, 8 AUTOLAND or RTL). To respect the 120 m of the open category, a 110–115 m altitude fence with RTL is a good safety net (chapter 36).

40.16 DJI O4 Pro OSD

Parameter	What it does	Default	Stallion
`OSD_TYPE`	5 = MSP DisplayPort (DJI O3/O4, Walksnail, HDZero).	board specific	5
`MSP_OPTIONS`	Bit 2 = use the Betaflight font, which is what the O4 displays.	0	4
`OSD1_*_EN/_X/_Y`	Elements and position on the HD grid.	—	the author's corner layout

40.17 The author's Stallion at a glance

Summary of the first-hover configuration by block. Items marked ⚠ are still to be measured or validated in flight.

Block	Parameters
Board and ports	AHRS_ORIENTATION 4 · SERIAL1_PROTOCOL 2 · SERIAL3_PROTOCOL 42, BAUD 115 · SERIAL5_PROTOCOL 5 · SERIAL6_PROTOCOL 23 · GPS1_TYPE 1 · COMPASS_EXTERNAL 1 · SCHED_LOOP_RATE 300 · OSD_TYPE 5 · MSP_OPTIONS 4

QuadPlane and tilt	Q_ENABLE 1 · Q_FRAME_CLASS 7 · Q_TILT_ENABLE 1 · Q_TILT_MASK 3 · Q_TILT_TYPE 2 · Q_TILT_YAW_ANGLE 15 · Q_TILT_MAX 45 · Q_TILT_RATE_UP 15 · Q_TILT_RATE_DN 0
Outputs	SERVO1/2_FUNCTION 4 · SERVO3 79 · SERVO4 80 · SERVO5 33 · SERVO6 34 · SERVO7 36 · SERVO8 75 (1075–2175 ⚠) · SERVO9 76 (900–2025 ⚠) · MIXING_GAIN 0.5 · RC2_REVERSED 1
Motors and ESCs	Q_M_PWM_TYPE 6 · SERVO_DSHOT_ESC 1 · SERVO_BLH_AUTO 1 · SERVO_BLH_RVMASK 64 · Q_M_THST_EXPO 0.6 · Q_M_THST_HOVER 0.25 (learned) · Q_M_HOVER_LEARN 2 · Q_M_SPIN_ARM 0.1 ⚠
Battery	BATT_MONITOR 4 · BATT_VOLT_MULT 11.05 · BATT_AMP_PERVLT 64 ⚠ · BATT_CAPACITY 8000 · Q_M_BAT_VOLT_MAX 25.2 · Q_M_BAT_VOLT_MIN 19.8 · Q_M_BAT_CURR_MAX 80
Filters	INS_GYRO_FILTER 60 · Q_A_RAT_YAW_FLTE 0 · INS_LOG_BAT_MASK 1
Pitot	ARSPD_TYPE 1 · ARSPD_USE 0 ⚠ (set to 1 after calibration)
Speeds and transition	AIRSPEED_MIN 16 · AIRSPEED_CRUISE 19 · AIRSPEED_MAX 35 · Q_TRANSITION_MS 7000 · Q_TRANS_DECEL 2 · Q_OPTIONS 163841 · Q_ASSIST_SPEED −1 · TKOFF_THR_SLEW 40
Radio and modes	RC_OPTIONS 8480 · RC5_OPTION 153 · FLTMODE_CH 8 · FLTMODE1–6 17, 17, 18, 18, 19, 19 · ARMING_SKIPCHK 1024 ⚠
Failsafes	THR_FAILSAFE 1 · THR_FS_VALUE 950 · FS_LONG_TIMEOUT 5 · FS_LONG_ACTN 0 · BATT_LOW_VOLT 19.2 · BATT_CRT_VOLT 18.0 · BATT_LOW_MAH 1600 · BATT_CRT_MAH 800 · BATT_FS_LOW_ACT 4 · BATT_FS_CRT_ACT 4 · BATT_FS_VOLTSRC 1
Return to home	RTL_ALTITUDE 100 · FLIGHT_OPTIONS 16 · Q_RTL_MODE 1 · WP_LOITER_RAD 50 · Q_LAND_FINAL_ALT 6 · Q_LAND_FINAL_SPD 0.3

40.18 Work order, from bench to first flight

1. **Firmware** and connection to QGroundControl. Save the factory parameter file.
2. **Ports** (40.3) and reboot. Check that the GPS gets a fix, the receiver comes through and the OSD shows in the goggles.
3. **QuadPlane and tilt** (40.7) in two passes with reboots. **Outputs** (40.9).
4. **Orientation**, accelerometer, levelling and compass (40.4).
5. **Radio:** calibration, modes, arm switch and `RC_OPTIONS` (40.11).
6. **Surfaces:** directions and throws in MANUAL and FBWA (40.10).
7. **Tilt:** directions, ends and alignment (40.8), props off.
8. **Motors, props off:** protocol, motor test (A front right, B rear, D front left), spin directions and SPIN_ARM/SPIN_MIN.
9. **Battery:** voltage and current calibration with a multimeter and wattmeter, and failsafes (40.6, 40.15).
10. **Radio failsafe on the bench:** armed without props in QSTABILIZE, switch the radio off and check it goes to QLAND and what it does when the signal returns.
11. **Centre of gravity** at 60 mm from the root leading edge, with everything installed.
12. **First hover** in QSTABILIZE at low height; then QHOVER and QLOITER. Review the log: vibration, learned THST_HOVER, consumption.

13. **Notch, VTOL tuning and yaw** (40.14).
14. **Pitot:** validate and calibrate; ARSPD_USE = 1.
15. **First transition** high and into wind; then fixed-wing AUTOTUNE, L1 and TECS, and finally missions.

> **References**
>
> ArduPilot Plane wiki: Tilt Rotor Planes, Tilt Rotor Setup Tips, Yaw Tuning for Tilt Vectored Yaw Configurations, QuadPlane VTOL Tuning Process, Assisted Fixed-Wing Flight and the complete parameter list. ArduPlane 4.7.2 source (quadplane.cpp, tiltrotor.cpp, Attitude.cpp, servos.cpp, AP_TECS, AP_L1_Control, AP_MotorsMulticopter, AP_BattMonitor). Flightory manuals: Stallion V2, Stallion VTOL PACK and “ArduPilot config for a 4+1 QuadPlane VTOL”.