

Launch detection

Launch detection is the moment everything pivots: it starts the flight recording, freezes the pad references, and drops the cellular radio for the airborne phase. It has to satisfy two demands that pull in opposite directions, never fire falsely from handling, wind or a slammed car door, and never miss or meaningfully delay a real launch, and Jupiter resolves that the way good detection always resolves it: by weighing evidence from two independent sensors rather than watching one number cross one line.

Two ways in: fast and slow

There are two independent paths to a launch decision. The fast path needs the accelerometer and the barometer to agree: acceleration must exceed the launch gate for a sustained total, 400 milliseconds' worth, while the altitude sits above the launch threshold for half a second, and when both hold, the launch is declared in well under a second. A second leg of the fast path exists for short, violent motors that burn out in under 200 milliseconds: they don't stay over the gate for long, but they hit hard, so acceleration exceeding twice the gate for 180 milliseconds counts as the same weight of evidence.

The slow path is altitude alone: the rocket above the launch threshold continuously accumulating two and a half seconds. It is always available, and it is deliberately slow, because pressure by itself is one sensor with no corroboration, and a wind gust, a door, or someone picking the rocket up can all look momentarily like a launch to a barometer. One sensor gets patience; two sensors in agreement get speed.

Fast path, under a second

Acceleration over the gate for 400 ms (or twice the gate for 180 ms), and altitude over the threshold for 500 ms. Two sensors agreeing buys speed.

Slow path, 2.5 seconds

Altitude alone over the threshold for a sustained 2.5 seconds. Always available, deliberately patient: one uncorroborated sensor never gets to be hasty.

Either path declares the launch; both must clear the lockout and arming gates first

Evidence, not edges

How the time is counted matters as much as the thresholds. The detector accumulates real elapsed time over its evidence windows rather than demanding one unbroken run, and that accumulation is itself the tolerance for reality: a rocket bouncing around a threshold in turbulence, or a momentary sensor dropout, subtracts from the evidence without resetting it to zero, so marginal conditions delay a decision instead of deadlocking it. Everything is measured in genuine elapsed time from the samples' own timestamps, never in sample counts, so the detection windows mean exactly what they say regardless of what else the processor is doing. And the acceleration test uses the total magnitude rather than any single axis, which means launch detection works with the device mounted at any angle, in any orientation, and depends on nothing else having calibrated or converged first.

The pad reference and the pre-launch buffer

The launch threshold is height above the pad, which raises the question of what "the pad" is. Jupiter computes it continuously while waiting: the pad reference is the average of the altitude readings aged between 10 and 18 seconds old, an eight-second window that deliberately ends ten seconds in the past. That aging is the trick. A real launch is declared within about half a second to three seconds of movement, long before the climb could ever age into the window, so the reference is always a settled, pre-flight baseline that the flight itself cannot pollute. The reference is also rate-limited, so a pressure transient, a nosecone going on or off, a hatch closing, nudges it slowly rather than dragging it, while the height-above-pad calculation itself is deliberately not rate-limited, so a real climb reads through instantly. At the moment of launch the reference freezes for the whole flight. The net effect is that "altitude above the pad" means exactly that, measured against where the rocket actually sat

in the minutes before flight, tracking slow atmospheric drift through a long pad wait and immune to the launch it exists to measure.

Behind detection sits the pre-launch buffer: Jupiter continuously holds at least the last 20 seconds of full-rate sensor history in memory, always, while waiting on the pad. This is what the post-boot LED countdown is filling, and it is why the calibration gate exists, a flight is not allowed to begin until that history is complete. The buffer is what makes everything else on this page possible: the evidence windows are read from it, the pad reference is computed from it, the recording opens with it so the log contains the final pad seconds before the decision, and the search that backdates $t=0$ to true ignition walks through it.

The gates in front of it

Before either path is allowed to declare anything, three gates apply. The launch lockout, `launch_lock_s`, is a hard timer from power-on during which nothing detects, whatever the sensors say, your protection against detection during final assembly and handling. The calibration gate holds detection off until the pre-launch buffers hold their guaranteed history, a countdown shown on the LEDs after boot, so a flight can never begin before the device is ready to record it properly. And arming policy: by default, `detect_unarmed` is on, meaning an unarmed device will still detect and record a flight, a deliberate choice, because logging a flight is not firing a charge, the outputs are gated by arm state entirely separately, and a device that silently refused to record because someone forgot to arm would be losing data for no safety gain.

The settings

Four settings shape detection. `launch_thresh_m` is the altitude above the pad that counts as "flying", settable from 1 to 200 metres. `launch_alp` sets the acceleration gate for the fast path: off, 1.5 g, 2.5 g or 4 g; with it off, only the slow path operates. `launch_lock_s` is the power-on lockout, up to 999 seconds. `detect_unarmed` chooses whether an unarmed device records flights, on by default as above.

The instant it fires

Detection latches one way: nothing un-launches a flight except landing. In the same instant, recording begins, the pad altitude reference freezes for the flight, the cellular radio is commanded off for the airborne phase, and every flight-scoped detector, burnout, apogee, landing, starts clean. Two details mean detection latency never costs you anything. First, the flight log does not begin at the moment of detection: the recording opens with the pre-launch buffer, the final seconds on the pad, so the whole event is in the log including everything before the decision. Second, the log's $t=0$ is not the detection time: after launch, Jupiter walks back through its recorded history to find the true moment of ignition and backdates $t=0$ to it, so every timestamp in your flight is measured from when the motor actually lit, not from when the firmware became certain.

Where it runs

Detection runs on the flight core, checked at 100 Hz directly beside the sensor pipeline, rather than delegated to a background task. The arithmetic explains why: a rocket at 100 metres per second covers 1.7 metres in a single loop pass, so a detector living somewhere with scheduling latency could be many metres late deciding when the flight began. This one cannot be.