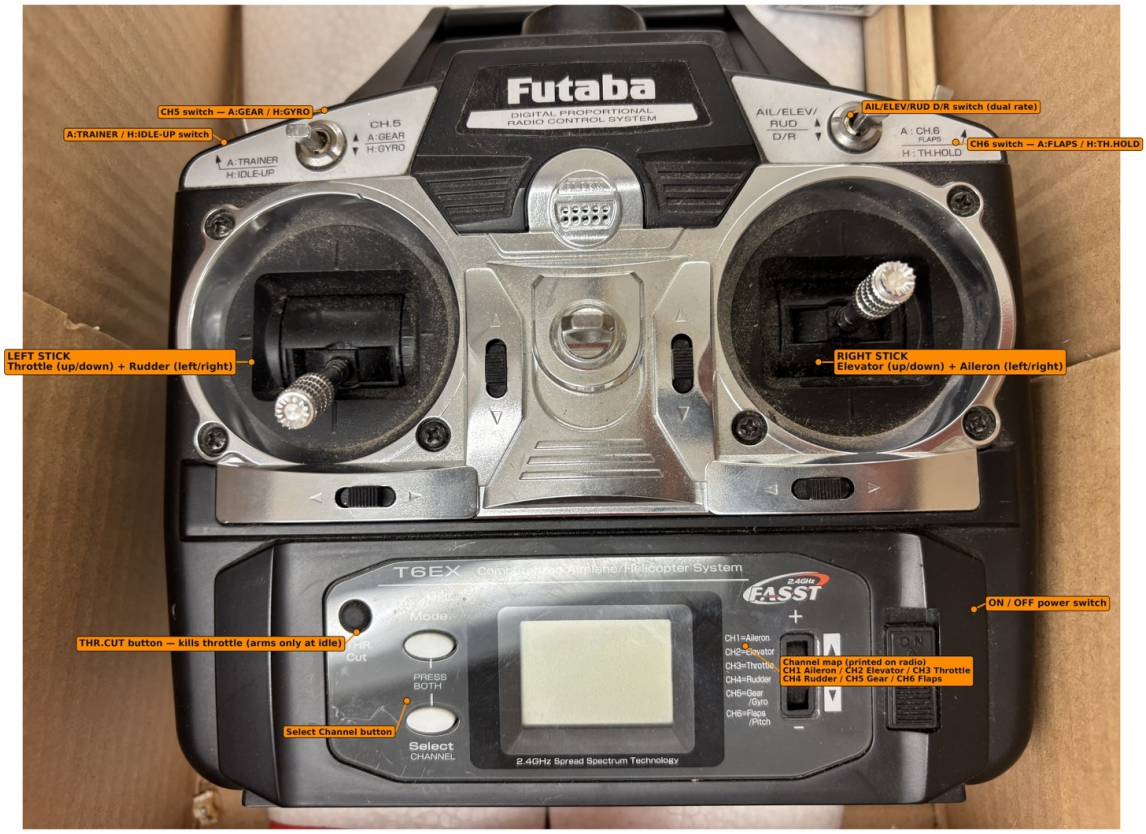
Fixed Wing RC Autonomous Drone — Transmitter

Futaba T6EX Transmitter

Reference summary — channel map, stick mode, switches, and telemetry limits

Download PDF



Futaba T6EX — stick functions, switches, and controls, as wired for this build.

Channel Map (printed on the radio itself)

Channel	Airplane Function	Helicopter Function
CH1	Aileron	Aileron
CH2	Elevator	Elevator
CH3	**Throttle**	Throttle
CH4	Rudder	Rudder
CH5	Gear	Gyro
CH6	Flaps	Pitch

Stick Mode

- Ships in **Mode 2** by default — confirmed factory setting for this transmitter.
- Mode 2 = **left stick:** throttle (up/down, no spring return) + rudder (left/right, spring-centered); **right stick:** elevator (up/down) + aileron (left/right), both spring-centered.
- This already matches "throttle on the left stick" — no mode change needed.
- To verify: hold **MODE + SELECT** while powering on; the active mode shows on the LCD.
- Moving throttle to a different stick isn't purely a menu toggle — the stick's physical throttle-detent/ratchet hardware has to move too (a service-center job). Not relevant here since it's already set the way you want it.

Switches Near the Sticks

- **AIL/ELEV/RUD D/R** — dual-rate switch, adjusts control throw/sensitivity on aileron, elevator, and rudder.
- **CH5 switch** — A:GEAR (airplane) / H:GYRO (heli). No retracts on this build, so it's free to repurpose — e.g. a camera trigger or an aux input for the flight controller later.
- **CH6 switch** — A:FLAPS (airplane) / H:TH.HOLD (heli). Also free to repurpose if the wing has no flaps.
- **A:TRAINER / H:IDLE-UP switch** — trainer (buddy-box) function in airplane mode; useful if an instructor ever wants link-in override capability during student flights.
- **THR.CUT button** — forces throttle fully off; only arms when the throttle stick is at idle, so it can't cut the motor mid-flight by accident. Worth setting up as a deliberate "kill the motor now" control on the ground.

Telemetry

This is **FASST** — a one-way (transmit-only) protocol. The T6EX does **not** receive telemetry back from the receiver: no RSSI, battery voltage, or sensor data shown on this transmitter's screen. Futaba's telemetry-capable systems are a different product line (S.Bus2 / T-FHSS / FASSTest-series radios).

A battery failsafe exists, but it watches the **receiver pack's own voltage** (trips below ~3.8V) — see the receiver page for why that doesn't double as a main-flight-battery warning if the receiver is powered through an ESC's BEC.

> **Compatibility note for the ArduPilot phase:** FASST receivers like the R606FS output individual PWM per channel, not a combined PPM/SBUS stream. That's fine for direct manual RC flight, but it is **not** directly compatible with a modern single-RCIN flight controller (e.g. Matek F405-Wing) planned for the autopilot phase. Bridging it would need a PPM encoder module — an extra part and failure point best avoided. When the project moves to ArduPilot, plan on swapping to an SBUS/CRSF-capable radio (e.g. ExpressLRS) rather than carrying this system forward into that phase.

Sources: Futaba 6EX-2.4GHz / T6EX Instruction Manual · Futaba 6EX 2.4GHz User Manual — receiver & servo connections