

MORPHEUS

I put this drone together from scratch, debugged all the issues, and finally got it in the air.

ERNESTO GUEMEZ / 5-INCH FPV QUADCOPTER / PHASE 2 OF 3

SUMMARY

A drone I built to understand every part of it

Morpheus is a 5-inch FPV quadcopter that I assembled and configured myself. It weighs about 650 g, runs on a 4S LiPo battery, and flies today under manual control. I had never soldered before this project, so the build was also my first course in electronics, firmware and debugging.

The long-term goal is an autonomous flight: takeoff, a GPS waypoint route, return to launch and landing, commanded by a Python program on a Raspberry Pi through MAVLink. That goal has **not** been reached. This report separates what is built and flying from what is planned, and it gives the most space to the failures I had to diagnose along the way.

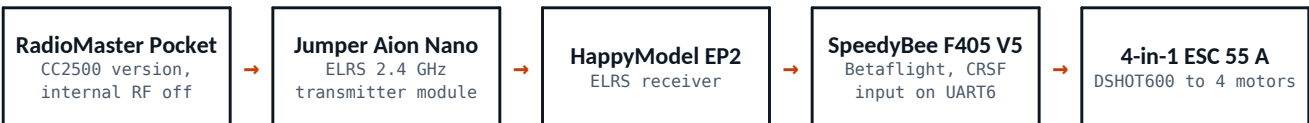
PHASE	SCOPE	STATUS
Phase 1	Build: frame, motors, soldering, electronics, configuration	Done
Phase 2	Manual flight and tuning: drift, oscillation, PID, RPM filtering	In progress
Phase 3	Autonomy: ArduPilot, GPS, telemetry, Raspberry Pi, Python missions	Planned, not built

SYSTEM ARCHITECTURE

Radio to motors, one link at a time

Control flows in one direction, from my hands to the motors. Each link uses a defined protocol, and most of the debugging in this project happened at the boundaries between links.

BUILT TODAY



PLANNED, PHASE 3 (NOT INSTALLED)

Beitian BN-880 GPS and compass, UART4 and I2C	SiK radio 915 MHz Ground telemetry, UART2	Raspberry Pi Zero 2 W Python, MAVLink over USB	ArduPilot Replaces Betaflight on the F405 V5
---	---	--	--

PART	WHAT I USED
Frame	Mark4 5 in carbon fiber, 5 mm arms, 16 x 16 mm motor mount
Motors	EMAX ECO II 2207 2400KV x 4
Flight controller / ESC	SpeedyBee F405 V5 stack, 55 A 4-in-1 ESC, 30.5 x 30.5 mm
Receiver	HappyModel EP2, ELRS 2.4 GHz
Radio	RadioMaster Pocket (CC2500) with external Jumper Aion ELRS Nano module
Propellers	HQProp 5 x 4.3 x 3 V1S
Battery	GNB 4S 1300 mAh LiPo, 120C, XT60
Weight	About 650 g

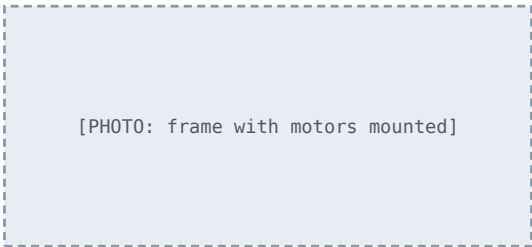
THE BUILD

Turning a box of loose parts into a flight controller that actually listens to my radio

FRAME AND MOTORS

Structure first

I started with a Mark4 5-inch carbon frame and four EMAX ECO II 2207 2400KV motors on 16x16 mm mounts. This is exactly where I realized my first batch of motors was the wrong size—log entry 1.



SOLDERING

Twelve motor wires, one capacitor, one battery lead

I ended up soldering the motor wires and a capacitor to the ESC, the XT60 lead to the power pads, and the receiver to UART6. Power and ground connect straight through and transmit and receive cross over. You have to be super careful since reversing power will instantly fry your parts, so I double-checked every single joint.



ELECTRONICS STACK

Stack, insulate, test for shorts

The SpeedyBee F405 flight controller sits right on top of the ESC. Since carbon fiber conducts electricity, I had to insulate the receiver under a plastic cover. Before plugging in the LiPo, I ran a bulb test and checked for continuity just to be safe (log entry 5).

[PHOTO: assembled stack with receiver]

CONFIGURATION

Betaflight, checked against measurements

It's running Betaflight 2026.6.1 with DSHOT600 and a 4S profile. I tuned the voltage scale until Betaflight read 15.2 V against 15.3 V on my multimeter, and two of the motors were spinning backward, but I fixed that in the software (I reversed M1 and M4) so I wouldn't have to resolder anything.

[SCREENSHOT: Betaflight motor direction screen]

I completely skipped the landing legs at first because the provided screws were too short. They're optional anyway, so the drone just landed on its battery strap. I eventually figured out a way to attach them though.

ENGINEERING LOG

Five things that went wrong and how I actually figured them out

I chose to highlight these five because they all required a totally different way of thinking. Fixing them was usually quick, but the diagnosis is the part I really found interesting.

1. The wrong motors

SOURCING DILIGENCE

SYMPTOM	The motors showed up, but they were size 2006 instead of the 2207s I needed.
DIAGNOSIS	I noticed it right in the middle of assembly when the specs didn't match what I had planned.
ROOT CAUSE	I trusted the title on AliExpress instead of checking the specific variant I was clicking on.
FIX	I opened a dispute, ordered the right 2207 motors, and waited for them to arrive.

LESSON The title is marketing; the variant you select is what you're actually buying.

[PHOTO: the wrong motors]

2. A radio that could not speak to the receiver

SYSTEMS THINKING

SYMPTOM	My RadioMaster Pocket didn't have an ELRS option, but my receiver only speaks ELRS.
DIAGNOSIS	I checked the box again and realized ELRS was a product option, and I had accidentally bought the CC2500 version.
ROOT CAUSE	Since they run on completely different radio chips, no firmware update was ever going to bridge them.
FIX	I fixed it by snapping a Jumper Aion ELRS module into the external bay and switching the settings to CRSF. But then the new module got stuck on "Loading". Turns out I needed to reseat the module properly on its data pins.

LESSON Always check the actual chips your hardware uses, since menus and packaging can be misleading.

[PHOTO: radio menu, Internal RF options]

[PHOTO: module stuck on Loading]

3. The drone that would not arm

METHODICAL DEBUGGING

SYMPTOM	The radio kept saying "armed", but the motors weren't doing anything.
DIAGNOSIS	Instead of messing with random settings, I checked the arming flags with the CLI status command. Only CLI and MSP were set, which just come from being plugged into a computer over USB, so they're harmless. There was no RXLOSS or FAILSAFE, so the receiver link was completely fine. Output: Arming disable flags: CLI MSP
ROOT CAUSE	The problem was that the controller wasn't getting the arm command because my ARM switch range got messed up when I swapped radio modules.
FIX	I tweaked the ARM range in the Modes tab, and the motors finally started up.

LESSON Just because the radio says it's armed doesn't mean the drone agrees. You have to check the drone's actual flags.

[SCREENSHOT: CLI status output]

[SCREENSHOT: Modes tab, ARM range]

4. Motors screaming, no lift

SYSTEMATIC ELIMINATION

SYMPTOM	On my very first try, the motors spun up at full speed but the drone just sat on the ground.
DIAGNOSIS	Since the electronics and radio were working, it had to be a thrust issue. All four motors were failing the exact same way, which pointed straight to the propellers.
ROOT CAUSE	Turns out, I had mounted all four propellers upside down.
FIX	I flipped them over, double-checked the spin directions, and it took off.

LESSON Rule out what's working first, then find the common denominator. Checking prop direction is definitely on my checklist now.

[PHOTO OR CLIP FRAME: the no-lift attempt]

5. A smoke stopper from a car bulb

SAFETY AND RESOURCEFULNESS

SYMPTOM	I was about to plug in the battery for the first time, but I didn't have a smoke stopper, and a short circuit would easily melt my wiring.
DIAGNOSIS	I needed a way to limit the current just in case, and I didn't want to wait days for a tool to ship.
ROOT CAUSE	It wasn't a failure, just a missing tool and a risk I didn't want to take.
FIX	So I wired a 12V car bulb in series with the battery. If there's a short, the bulb lights up and saves the board. I also ran a multimeter continuity test before connecting power.

LESSON Always make sure your first power-up is survivable before you actually do it.

[PHOTO: bulb in series with the battery lead]

ALSO FIXED

PROBLEM	CAUSE AND FIX
Charge-only USB cable	Betaflight couldn't see the drone because I was using a charge-only cable. Swapping to a data cable fixed it instantly.
CRSF missing from firmware	The factory firmware didn't support CRSF, so I reflashed it to 2026.6.1 with CRSF enabled.

CONFIGURATION AND VERIFICATION

Every number I could check against the real world, I checked

ITEM	SETTING OR RESULT
Firmware	Betaflight 2026.6.1, SPEEDYBEEF405V5 target, CRSF enabled
Receiver link	ELRS 2.4 GHz, CRSF on UART6. Bound radio and receiver, all channels respond (AETR).
ESC	DSHOT600, motor idle 5.5 percent, motor stop off
Mixer and motors	QUAD X. M1 CW, M2 CCW, M3 CCW, M4 CW. M1 and M4 reversed in software.
Battery	4S, voltage scale 110, minimum 3.3 V and warning 3.5 V per cell
Voltage check	15.2 V in Betaflight against 15.3 V on a multimeter
Sensors	ICM42688P gyro and accelerometer, DPS310 barometer. Accelerometer calibrated, pitch and roll near zero and stable.
Modes	ARM and Angle
Failsafe	Tested. Motors cut about one second after the radio is switched off.

Configuration is always done with the propellers removed. The motors spin freely when disarmed, which is normal, and propellers go on last.

WHAT'S NEXT

Finish tuning, then teach it to fly itself

The order matters. Betaflight does not support waypoint missions, so autonomy needs ArduPilot. Flashing ArduPilot erases my Betaflight settings, and there is no one-to-one migration of PIDs or modes. So I will finish tuning and flying in Betaflight first, then switch firmware, then add hardware, then prove everything in simulation before the first autonomous flight.

PORT	PLANNED USE
UART2	SiK telemetry radio, 915 MHz, air and ground pair for monitoring
UART4	GPS (Beitian BN-880). The compass connects separately over I2C.
UART5	ESC telemetry (already assigned)
UART6	RC receiver (already assigned)
USB	Raspberry Pi Zero 2 W, MAVLink over USB, so no UART is consumed

Failsafes to configure before any autonomous flight: geofence, return to launch on low battery, and a defined response to signal loss. **Power:** I plan a dedicated 5 V 2 A regulator for the Raspberry Pi, so a current spike from the Pi cannot brown out the flight controller in the air. It is not yet purchased or installed.

Scope decision. I cut a documentation camera and digital FPV video from Phase 3. Neither helps the drone fly itself. They would add cost, weight and a new piloting skill. I will reconsider only if basic autonomy works and time remains.

LINKS

ITEM

WHERE

Project page	morpheus-drone.pages.dev
Contact	ernesto.guemeza@gmail.com