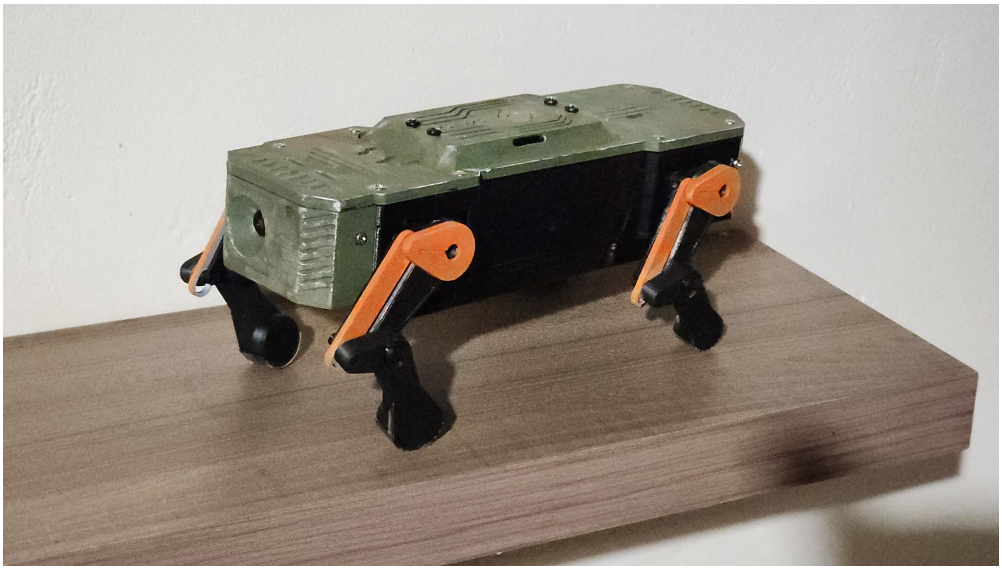
1. Before you begin



Tobias MK5 is a quadruped robot using four MG90S servos and two ESP32 boards. The owner has printed, assembled and tested this physical revision. This guide covers assembly order, component references and software setup.

The package contains one Bambu Lab A1 print project with six plates. Open it as a project in Bambu Studio to preserve placements and settings. No Ender profile, separate STL files or Blender files are included.

Tools: suitable screwdrivers, multimeter, soldering iron, pliers and deburring tools. Disconnect power before changing wiring.

Renders illustrate the overall layout. Some photos and the elastic-band video show earlier colors. Use the supplied 3MF as the source for the parts to print.

2. Printing on the Bambu A1



Settings read from the supplied project: Bambu Lab A1, 0.4 mm nozzle, PLA, 0.2 mm layers, 4 walls and 35% infill. These are saved project settings, not measured print time or material use.

Plates: 1 — Base and brackets; 2 — Sides and rear; 3 — Lid and front; 4 — Black feet; 5 — Orange shoulders; 6 — Struts. Plate 6 retains its historical Portuguese name, “Hastes multicor” (multicolor struts), but the four struts are solid orange parts. A white filament slot does not mean a separate white insert exists in this project.

Open all plates, map your available filaments and preserve the saved orientations. Slice each plate and inspect layers, supports and first-layer placement before printing. Time and material use depend on slicing.

Remove supports and burrs, then dry-fit the parts. Joints must move freely. Do not force a blocked hole into place.

3. Identify parts and fasteners

There are 25 structural parts: base, lid, left side, right side, front, rear, battery tray, cradle and capacitor holder; plus four leg assemblies, each with a shoulder, arm/crank, strut and foot.

FL = front left; FR = front right; RL = rear left; RR = rear right. Left and right are defined looking in the same direction as the robot camera, not facing the robot.

For each servo, use the manufacturer-supplied screws, especially the screw securing the horn to the shaft. Do not substitute an M2 screw simply because it looks similar.

Body screw buying reference: an M2 × 4, 6 and 8 mm assortment. M2 × 6 mm is a starting option to try where depth allows. The builder used small screws already available and did not record all lengths. These are selection references, not verified lengths for every hole.

Check the joined thickness and available depth. A screw must grip without bottoming out in a blind hole, protruding through the opposite wall, touching the battery or locking a joint. Adjust length or washers as needed. Tighten by hand without crushing plastic. The model list contains 32 screw positions, including four servo-shaft screws, and eight M2 nuts.

4. Assemble the body and brackets



1. Lay out the base, battery tray, cradle and capacitor holder. Check holes and insert nuts into their pockets before closing access areas.

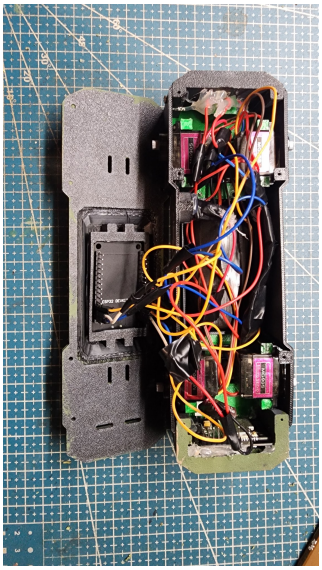
2. Dry-fit these supports to the base. Check clearance for your actual battery and both converters; dimensions can vary between manufacturers.

3. Prepare the front, rear and side panels. Seat the camera in the front opening without pressing its lens or sharply bending the ribbon cable. Keep programming connectors accessible.

4. Seat the four servos in their matching locations, with shafts aligned to the openings. Use the mounting tabs and suitable screws for the servo and printed parts.

5. Route and label the four servo cables FL/FR/RL/RR. Do not pinch wires between panels. Leave the lid off until wiring and servo alignment are complete.

5. Prepare power and firmware



1. With boards and servos disconnected, adjust both converters and measure their outputs. This project separates the servo branch (within the MG90S specification, 4.8–6 V) from the logic branch (regulated 5 V). Never connect the 2S battery directly to the MG90S servos.

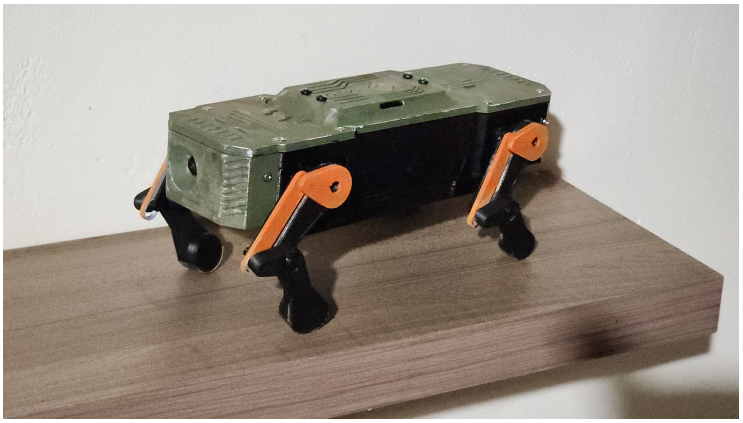
2. Connect a common GND between converters, ESP32 DevKit, ESP32-CAM and speaker driver. Observe capacitor polarity: 1000 µF near the servo branch and 470 µF near each board.

3. Before attaching cranks to servo shafts, flash the leg firmware over USB, configure the network and start the dashboard. Electronic centering is needed in the next step.

4. The Li-ion pack is 2S: 7.4 V nominal and 8.4 V fully charged. Use a compatible 2S charger. Jack and switch terminal layouts depend on the purchased components; verify continuity and polarity instead of copying terminal positions from a photo.

5. Test each power branch before connecting all servos. Resets during motion can indicate voltage drops: stop and inspect power delivery, wiring and connectors.

6. Center servos and fit the legs



1. Connect servo signals to the ESP32 DevKit: FL to GPIO14, FR to GPIO15, RL to GPIO13 and RR to GPIO2. Servo positive and GND come from the regulated supply, not the 3V3 pin.

2. Support the body with legs clear and shafts unloaded. In the dashboard, open Settings and the leg adjustment controls, then select the command to align all legs to 90°. Confirm only when ready for movement.

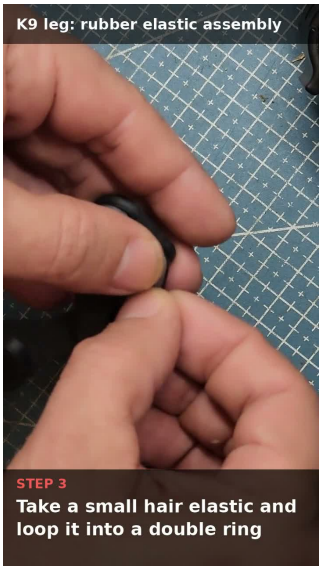
3. Wait for centering. Fit cranks according to FL/FR/RL/RR and the mechanical assembly reference. The 90° command is an assembly reference; these servos do not measure or confirm their physical position back to the controller.

4. Fit the strut onto the foot pin and assemble the leg mechanism. Check mirrored left/right placement and that all pins are seated. Add the elastic bands as described on the next page.

5. Secure each crank with the original servo-shaft screw. Do not force a powered shaft by hand. If a crank needs repositioning, switch off first, reposition and check the reference again.

6. Check that joints are free and cables stay outside their travel. Test small movements with the body supported before placing it on the floor.

7. Elastic bands and foot return



1. Fit the strut to the foot and check that the mechanism moves freely.

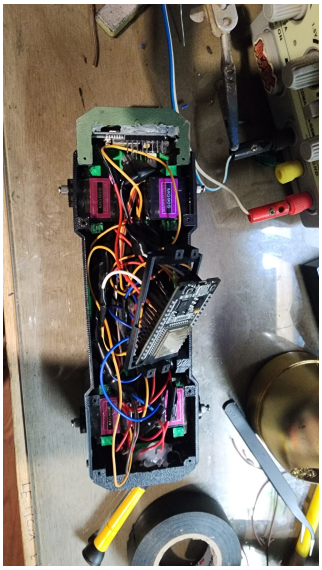
2. Form a small elastic band into a double loop and route it around the retaining points shown in the supporting video. Tension holds the assembly together and provides passive joint return.

3. The project reference is three small hair elastic bands per leg. Start with this number and check the mechanism: it should not be loose or require excessive force to flex.

4. With power off, flex the foot by hand. It should return by itself. Compare all four legs and replace cracked bands or bands with noticeably different tension.

Supporting video: 04_Leg_Elastic_Bands_9x16.mp4. Its white strut is from an earlier color version. The supplied 3MF uses solid orange struts. The clip explains the joint and band routing, not color separation.

8. Electronics and speaker



ESP32 DevKit handles servo signals and sound. ESP32-CAM handles video, flashlight and strobe. The boards communicate over Wi-Fi; the project wiring does not require a wired data connection between them.

Speaker wiring: GPIO25 → 220 Ω resistor → TIP41C base. Emitter → common GND. Collector → speaker negative. Speaker positive → regulated 5 V.

With the TIP41C printed face toward you and leads pointing down, the pins are Base, Collector, Emitter. The metal tab is connected to the Collector; insulate it from other conductors.

Fit the 1N4007 across the speaker: striped cathode to positive, anode to the collector. Use the included wiring diagram at full size.

Secure wires without pulling connectors. Do not compress the battery, camera ribbon or components when closing the lid.

9. Software and first connection

Public code: https://github.com/lizard-wizard42/K9-TOBIAS . Follow each board guide in k9v2-1_firmware/boards. ESP32 DevKit and ESP32-CAM use different builds; do not interchange their firmware files.

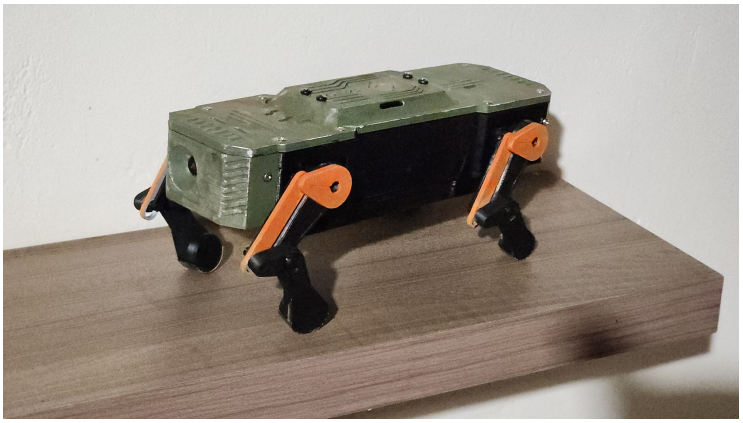
Install PlatformIO and Node.js 20 or later on your computer. In the firmware directory, copy src/secrets.example.h to src/secrets.h and enter your network credentials and local passwords. Do not share secrets.h or binaries containing your credentials.

Build using “pio run -e esp32dev” and “pio run -e esp32cam” in the firmware directory. Follow each board’s USB instructions for initial flashing. ESP32-CAM may require GPIO0 grounded during reset/flashing and 3.3 V UART signals.

In k9_control_hub, run “npm ci”, then “npm start”. Open http://localhost:3080 on the computer. On a phone, open http://YOUR-COMPUTER-IP:3080 on the same network.

Keep the computer and both boards on the same 2.4 GHz Wi-Fi network. Check board status, camera image and Stop response. Voice/AI is optional and configured separately.

10. Test, close and maintain



1. With the body supported, test centering, small movements and Stop. Check that wires avoid cranks and joints and that all four channels match their intended legs.

2. Place the robot on a clear, level surface. Start slowly and observe its gait. If a foot catches, a servo buzzes continuously or a board resets, stop and inspect the assembly.

3. Check the camera and speaker. Volume runs from 0 to 100 in steps of 5; zero/mute also silences emotions. Standard poses and actions do not play automatic sounds.

4. Close the lid only after testing. Tighten screws gradually and make sure the lid does not press against boards or the battery.

5. Before each use, check crank screws, elastic wear, connectors and battery charge. Disconnect power before mechanical adjustments or soldering.

Quick checks: uneven gait → mechanical reference, leg orientation and elastic tension; resets → power and GND; camera offline → camera firmware/network/power; no sound → volume, GPIO25 and driver.

11. Links, videos and credits

Code, both firmware builds, dashboard and contribution guide: https://github.com/lizard-wizard42/K9-TOBIAS

Hardware and wiring: https://github.com/lizard-wizard42/K9-TOBIAS/tree/main/docs/hardware

Leg board: https://github.com/lizard-wizard42/K9-TOBIAS/blob/main/k9v2-1_firmware/boards/esp32dev.md

Camera board: https://github.com/lizard-wizard42/K9-TOBIAS/blob/main/k9v2-1_firmware/boards/esp32cam.md

The full video and supporting clips are linked from the MakerWorld description. The main video covers the project, assembly and testing; use this PDF and the wiring diagram for individual connections.

Tobias MK5 was inspired by and based on miniRobo K9V2.1: https://www.minirobo.io/projects/k9v2-1 . Tobias modifications and integration: Lizard Wizard — https://www.lizardwizard.io/ .

For print-file licensing, see the MakerWorld listing. Code licensing and third-party terms are documented in LICENSE and THIRD_PARTY.md on GitHub. Screw references do not replace checking available depth at each fastening point.