

JarzRover iPhone-ESP32 - Installation Guide

Basic-kit customers: check [What to Buy](#) for quantities, Amazon shopping links and missing-parts checks.

This guide is for the **iPhone mounted on the rover**, using a **classic ELEGOO ESP-WROOM-32 / ESP32 Dev Module** over **BLE**. Documentation/kit revision v1.1.0, 7 October 2026. The handheld remote is a separate phone running Controller.

A Mac with Xcode, CocoaPods and Flutter, an Apple account, source assistance, and two compatible iPhones are needed for the two-iPhone setup. No signed public iOS apps are included.

Read [Hardware Reference](#) and [Wiring Reference](#) first. Photos below show the original chassis build; use them for mechanical placement only. They are not an ESP32 header or power-wiring map.

Work together safely

An adult checks wiring, soldering, power supplies and powered motor tests. Children can label parts, read steps and record results. Disconnect battery and USB before moving wires. Keep all wheels raised during motor tests; keep fingers, hair and loose clothing clear. Stop immediately for buzzing, a stalled motor, heat, smoke or a damaged battery. Never charge a damaged battery. Use its specified charger.

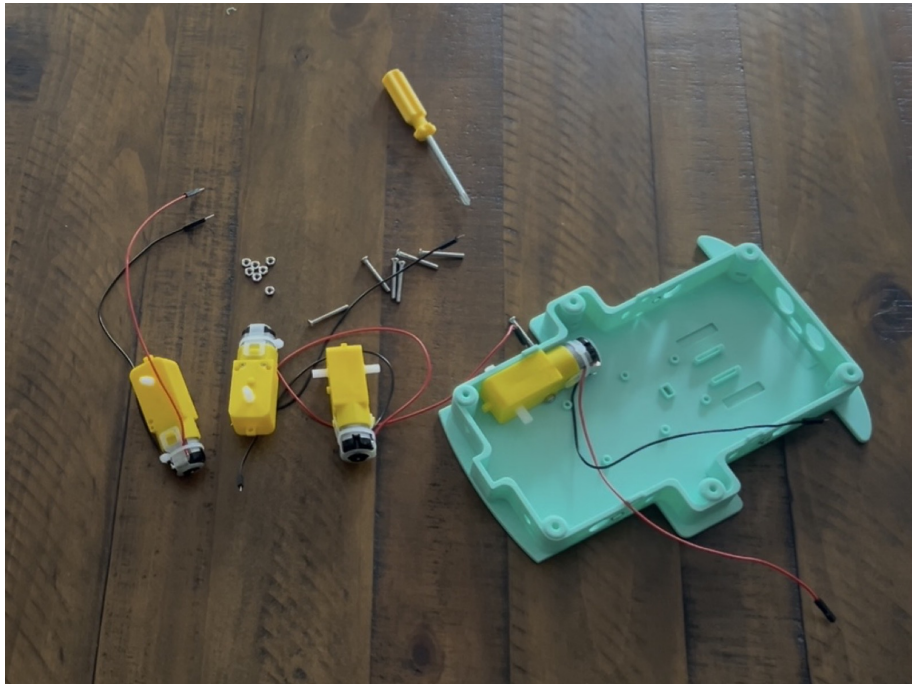
Do not apply raw battery voltage to 5V, 3V3 or signal pins. Never feed motor current through the controller board, USB cable or a breadboard. Photos show mechanical placement; board labels and the wiring table determine electrical connections.

Check parts and software before assembly

Part	What you need / check
Rover computer	iPhone; runs Robot and mounts on chassis
Low-level board	classic ELEGOO ESP-WROOM-32 / ESP32 Dev Module; verify printed module/board label
Remote	Separate phone running Controller; this guide describes a second iPhone
Motor system	L298N module, four TT motors, four matching wheels; check module and motor ratings
Front distance sensor	HC-SR04 with compatible regulated 5 V supply
Wheel sensors	Two front digital speed sensors and matching encoder disks; identify output type
Power	Rated battery, correct charger, switch, matching pigtail/connector and insulated wiring
Printed parts	One-piece body OR bottom + top; both phone-mount parts
Fasteners	8 M3 motor screws with nuts, 2 M3 phone-mount screws with nuts, 2 long back-cover screws, 2 small speed-sensor screws; confirm your order
Tools	Computer with Arduino IDE, USB data cable, screwdrivers, multimeter; adult soldering/insulation tools where needed
ESP32 input protection	ECHO divider or level shifter, separate appropriate speed-input conditioning; see wiring guide
ESP32 power	Verified USB supply for bench tests; an identified regulated mobile supply for untethered driving

Step 1 - Identify and print the parts

- 1 Unzip the customer kit and open its stl/ folder in your slicer. The files supplied are JarzRover_Body_Full.stl, JarzRover_Body_Bottom.stl, JarzRover_Body_Top.stl, phone_mount_bottom.stl, and phone_mount_top.stl.
- 2 Choose either the one-piece JarzRover_Body_Full.stl or the split JarzRover_Body_Bottom.stl + JarzRover_Body_Top.stl body option; do not print both body options for one rover. Print both phone-mount pieces. The kit does not include wheel STLs; use the wheels supplied with the four TT motors. Arrange parts on the build plate within your printer's limits. If your printer cannot fit the body at 100%, see the FAQ before scaling the STL.
- 3 Slice and print the body pieces and phone mount using settings suitable for your printer, filament, and part geometry. The exact approved printer profile, material, orientation, supports, infill, and print times are not prescribed in this kit. Review each slicer preview and test-fit parts before assembling the electronics.
- 4 Remove supports and strings carefully. Clear screw holes and slots without removing structural material. Check for cracks, warping, or motor mounts that are too tight.
- 5 Lay out the printed components, motors, fasteners, wires, classic ELEGOO ESP-WROOM-32 / ESP32 Dev Module, L298N, sensors, battery, and switch before assembly. Label the left and right motor groups.



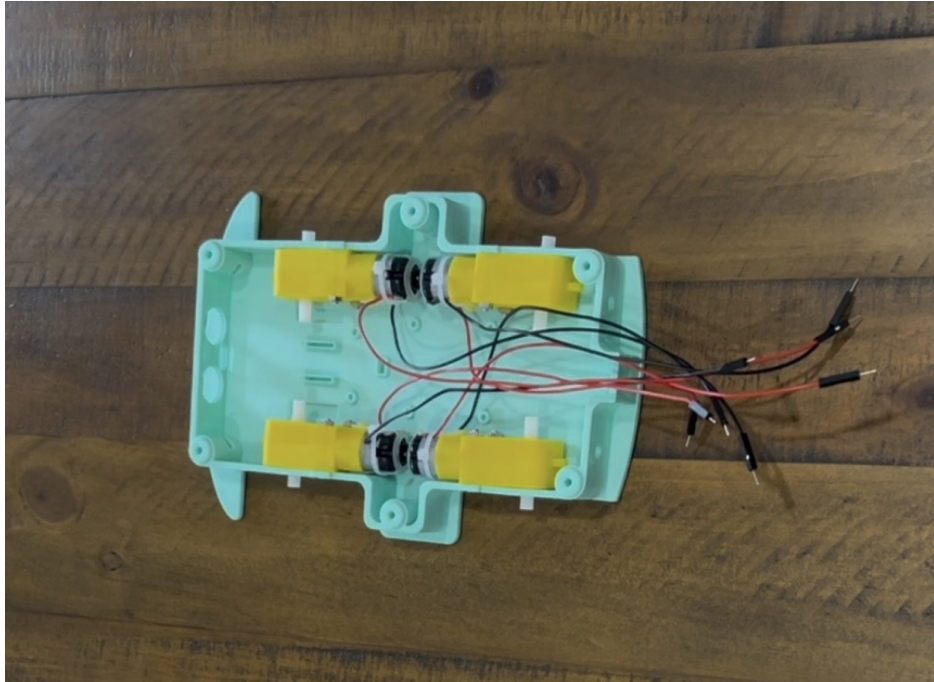
Motors, fasteners, screwdriver, and lower shell laid out before assembly

Figure 1. Parts and lower chassis at the start of the documented reference build.

Step 2 - Install the motors in the chassis

- 1 Place the lower shell with its motor pockets accessible. Use the front/rear shape of the printed chassis and the assembly photo to keep the rover's front orientation consistent.
- 2 Seat each TT motor fully in its intended pocket. Do not force a motor into a pocket that is warped or too small; correct the print fit first.
- 3 Secure the motors with the 8 M3 screws and matching nuts/bolts: two fasteners per motor. Tighten until secure without crushing the printed plastic or distorting the motor case.
- 4 Mark the motor leads by physical position (left-front, left-rear, right-front, right-rear) before joining them into the left and right driver groups. The firmware controls left and right sides independently, so a swapped group can make the rover pivot or drive unevenly.

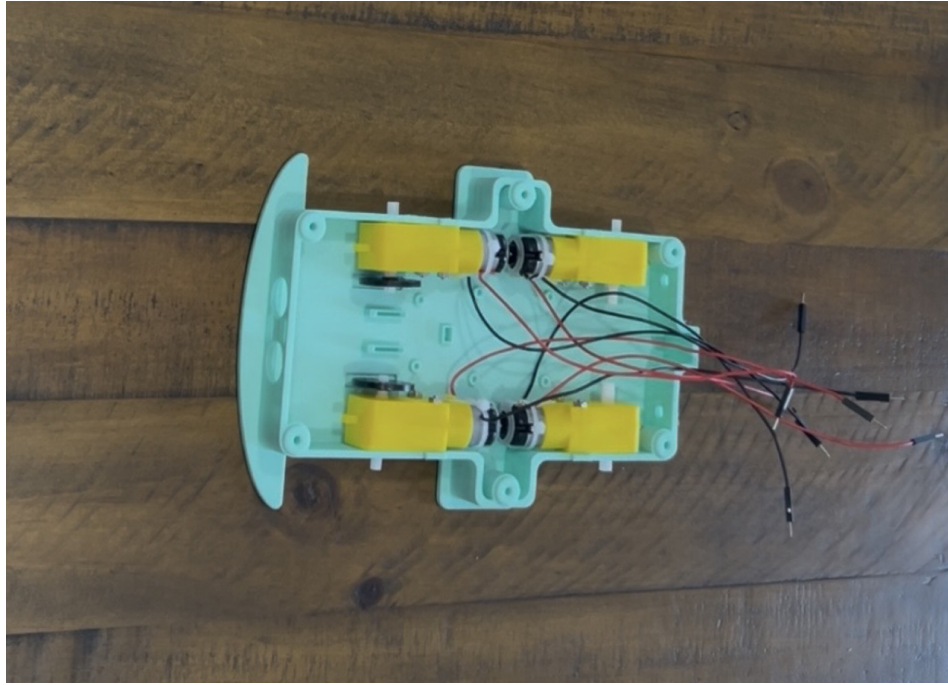
- 5 Turn each wheel/gearbox by hand while power is disconnected. Check that nothing rubs the shell and that the gearboxes are not pinched by the mounts.



Four TT motors installed in the lower chassis

Step 3 - Route and connect motor leads

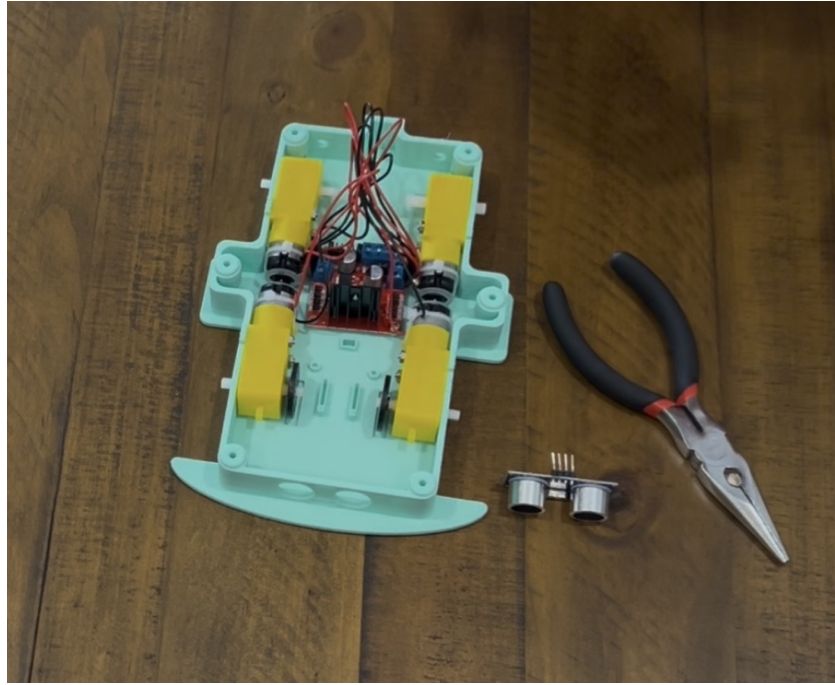
- 1 Route motor leads through the chassis openings shown in the build photos. Keep them away from rotating gears, wheel shafts, screw points, and the edge where the top plate closes.
- 2 Keep the two left-side motor leads grouped for the driver's left channel and the two right-side leads grouped for its right channel, following the working reference assembly.
- 3 Use secure, insulated connections appropriate for motor current. Do not rely on loose twisted wires or leave bare conductors exposed inside the chassis.
- 4 Leave enough slack for the chassis/top plate arrangement, but secure excess wire so it cannot reach a wheel.
- 5 Do not guess the exact L298N output terminal labels or motor polarity from a photo. Trace the specific module labels and the reference assembly; record the resulting left/right polarity before closing the chassis.



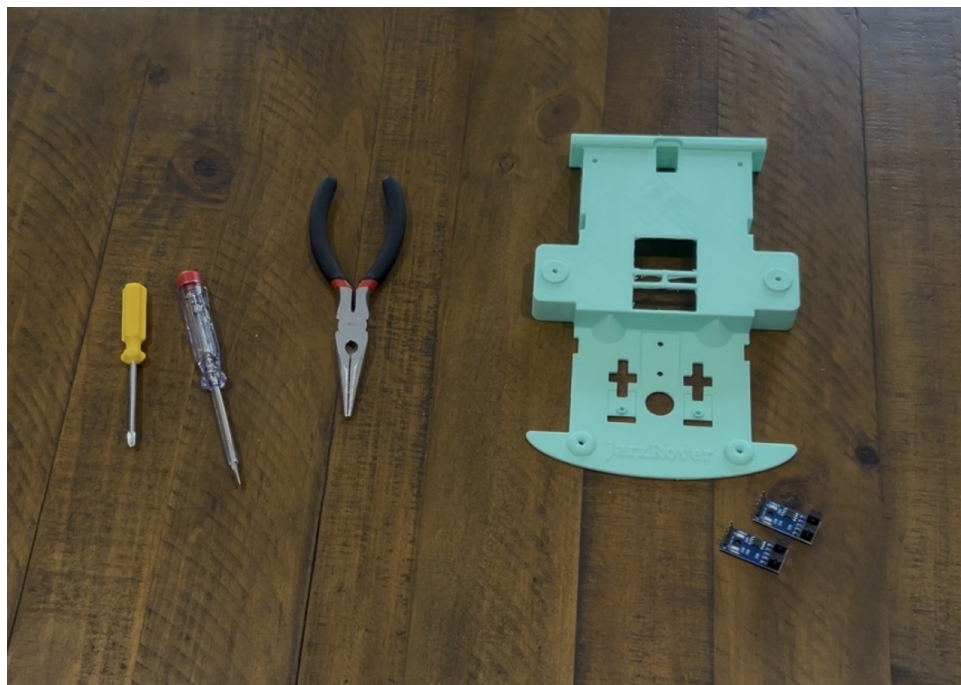
Motor wiring routed inside the chassis

Step 4 - Mount the controller board, motor driver, and sensors

- 1 Place the ESP32 and L298N so that wiring stays accessible and neither board can short against the chassis or battery. Use the included mounts/fasteners where the print provides them.
- 2 Mount the HC-SR04 at the front with both transducers facing forward and unobstructed. Secure it so it cannot swing into the wheels or chassis.
- 3 If fitting the two front speed sensors, place them so each encoder disk passes the sensor without scraping. Use the 2 small speed-sensor screws, but adjust their position for alignment, then secure them so they cannot move into the disks. Turn the wheels/encoder disks by hand to check clearance before tightening anything further.
- 4 Keep high-current motor wiring separated and secured from small signal wires where practical.
- 5 Arrange the battery and switch so the battery is restrained, the switch remains accessible, and wires cannot be trapped beneath the cover. Verify that the battery connector cannot be plugged in with reversed polarity.



Driver and front sonar mounted in the chassis



Top plate and sensor components shown before fitting

Step 5 - Wire your board

Keep all power disconnected. Follow the full circuit diagrams in [Wiring Reference](#) and have an adult verify supplies, jumpers, input circuits and common ground. Use this signal table as a final cross-check:

Board pin	Connect to	Purpose / check
GPIO16	L298N IN1	Left motor input 1
GPIO17	L298N IN2	Left motor input 2
GPIO25	L298N IN3	Right motor input 1
GPIO26	L298N IN4	Right motor input 2
GPIO32	HC-SR04 TRIG	Verify 3.3 V trigger is recognized
GPIO33	Conditioned HC-SR04 ECHO	Divider or verified level shifter required
GPIO34	Conditioned left speed DO	Input only; no internal pull resistor
GPIO35	Conditioned right speed DO	Input only; no internal pull resistor
GND	L298N GND and sensor grounds	Common signal reference

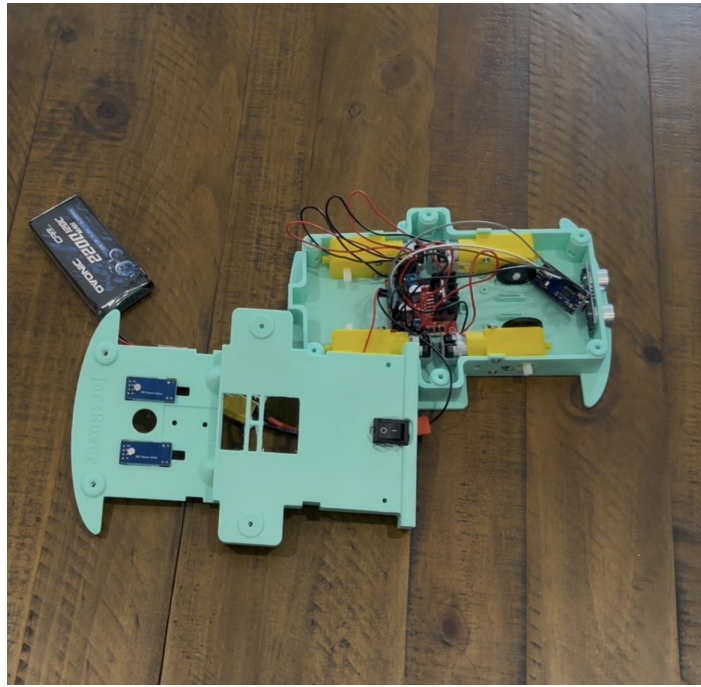
Step 6 - Test components and restore firmware

In Arduino IDE Preferences, add https://espressif.github.io/arduino-esp32/package_esp32_index.json to Additional Boards Manager URLs. In Boards Manager install **esp32 by Espressif Systems version 2.0.17** and choose **ESP32 Dev Module**. This firmware uses the 2.x LEDC APIs; do not select 3.x for this release. Choose the correct USB port. No extra BLE library is needed.

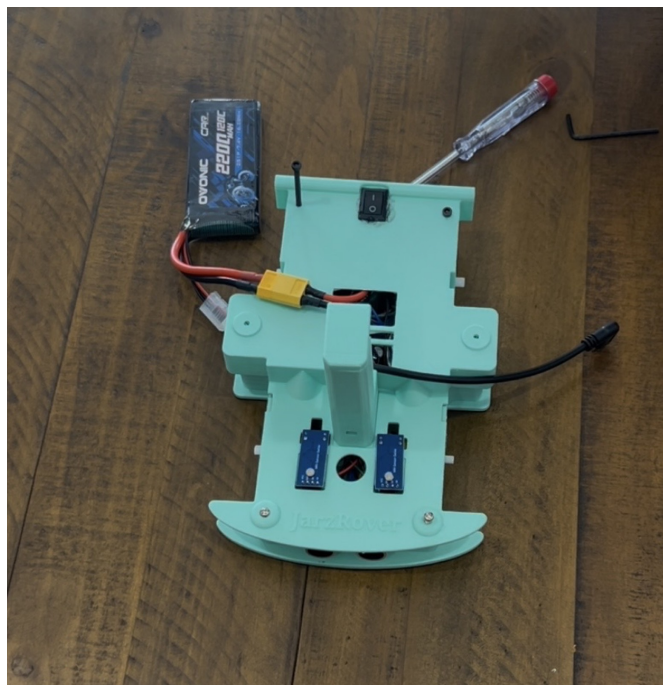
- 1 With motor power off, connect the board to the computer using a USB data cable.
- 2 Open [Component Tests](#). Run serial, sonar and hand-turned speed checks first, followed by the adult-supervised motor test. Then run the BLE test. Use the included sketches from tests/iphone/; an upload alone does not mean a component passed.
- 3 Disconnect motor power before changing sketches. Each uploaded test replaces the board's normal program.
- 4 When tests pass, open firmware/openbot_ios/openbot_ios.ino from this ZIP in Arduino IDE. It selects OPENBOT DIY_ESP32 with the JarzRover-specific pins above; do not choose the old general DIY_ESP32 sketch. Click Verify, then Upload, keeping motor power off.
- 5 Disconnect the programming USB cable and connect the verified ESP32 supply. Keep all wheels raised for app testing.

Step 7 - Fit the cover and phone mount

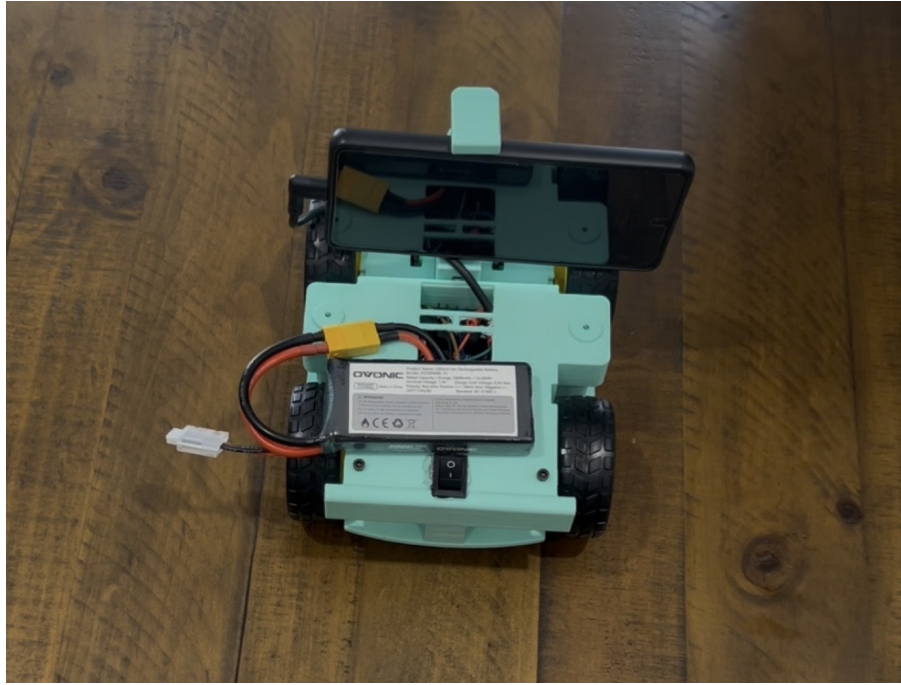
- 1 Before closing the body, take a final photo of the open chassis and verify all connectors are secure and insulated.
- 2 Move all wires away from motor gears, wheels, and cover edges. Check that the switch and battery remain reachable.
- 3 Fit the top plate without trapping wires. If it does not seat easily, stop and find the obstruction; do not tighten screws to force it closed.
- 4 Attach the phone mount with the 2 M3 screws and matching nuts/bolts. Install the 2 long M3 screws for the back cover where they align cleanly.
- 5 Skip the middle screw positions unless they align easily. Do not force middle screws into a scaled or slightly shifted print; the rover is designed to work without them when the main motor, mount, and back-cover fasteners are secure.
- 6 Mount the iPhone in landscape orientation as intended by the app. Secure the phone against sliding and keep the forward camera unobstructed.
- 7 Keep the two phone roles distinct: the mounted rover phone runs Robot; a second phone runs Controller.



Open chassis and electronics before final closure



Front assembly and phone-mount arrangement



Phone mounted on the completed rover

Install Robot on the rover iPhone

A signed public iOS release is not available yet. JARzLabs is working on Apple developer distribution and will share a signed iOS binary when available. No date is promised. This ZIP contains no installable iPhone Robot or Controller binary. The current workflow needs a Mac, Xcode, an Apple account and source access. A Windows PC alone cannot perform these Xcode steps.

These steps follow the repository's `ios/README.md`, `ios/OpenBot/Podfile` and app target. Ask an adult familiar with Mac software setup to help.

- 1 Request the matching jarz-development source from JARzLabs. Install Xcode and CocoaPods on the Mac. Use a rover iPhone running iOS 15.5 or later, the actual Robot target minimum; the older iOS 13 line in the repository README is outdated.
- 2 Open Terminal in the downloaded source folder, enter the `ios/OpenBot` folder, and run `pod install`. Wait for dependency installation to finish.
- 3 Open `ios/OpenBot/OpenBot.xcworkspace` in Xcode. Use the workspace, which includes the app's dependencies.
- 4 Add your Apple account in **Xcode > Settings > Accounts**. Select the **OpenBot** app target, open **Signing & Capabilities**, enable **Automatically manage signing**, and select your own Team. Follow Xcode's signing/device-registration prompts. Ask for help if your account cannot sign the project.
- 5 Connect and unlock the iPhone. Trust the Mac when asked. Enable **Developer Mode** if requested by iOS and trust the development account if prompted.
- 6 Select the **OpenBot** scheme and your physical iPhone as the destination, then click the triangular **Run** button. Wait for installation and open the app. A simulator build cannot be installed on the phone.
- 7 Allow Bluetooth, camera and local-network permissions when needed. Restore the normal ESP32 firmware before connecting in the app. The normal firmware advertises **OpenBot: DIY_ESP32**. Use the app's BLE connection screen to select it; close any connected BLE scanner first.

The USB cable from Mac to iPhone is for app installation. Rover control is iPhone Robot to ESP32 over BLE. Development signing depends on your account and device; Xcode may require renewal/reinstallation. A raw IPA is not a universal iPhone installer. No Firebase service credential is needed for basic control or Creature Lab.

Install Controller on the handheld iPhone

This is the separate **Flutter Controller** project, not the native Robot project. You need a second phone for this two-phone control setup.

- 1 On the Mac, install Flutter using its official setup instructions, then run `flutter doctor` and resolve the reported iOS toolchain requirements with your adult helper.
- 2 In the source folder `controller/flutter`, run `flutter pub get` to download dependencies.
- 3 Connect the second iPhone. Run `flutter devices` to identify it. For signing, open `controller/flutter/ios/Runner.xcworkspace` in Xcode, select the **Runner** target and set your own Team under **Signing & Capabilities**. Follow the device trust/Developer Mode prompts as for Robot.
- 4 Return to `controller/flutter` and run `flutter run -d DEVICE_ID`, replacing `DEVICE_ID` with that phone's identifier. Keep the phone unlocked. The repository's `controller/flutter/README.md` is the maintained setup reference; ask JARzLabs if build requirements differ on your Mac.
- 5 Keep Robot open on the rover iPhone and Controller open on the handheld iPhone. Put both on the same local Wi-Fi network and select **Phone** control in Robot where offered. Controller communicates with the Robot phone over the network; it does not connect directly to the motor board.

Complete the raised-wheel tests before driving. App installation and BLE discovery alone do not establish working motor control.

First complete rover test

- 1 With power off, turn each wheel by hand. Check for rubbing, loose wires and pinched leads. Recheck the board-specific wiring table and common ground with an adult.
- 2 Restore the normal firmware after component tests. Secure the rover with every wheel raised and leave automatic movement off.
- 3 Connect the Robot phone to its board, then connect the handheld Controller phone to Robot over the same local Wi-Fi network. Allow requested Bluetooth/local-network/camera permissions appropriate to your platform.
- 4 Ask for a brief low-speed forward movement, then release the control. Check both sides move in the expected direction and stop. Repeat reverse and gentle left/right turns. Stop immediately if a wheel stalls or buzzes.
- 5 Test stop on release, then app closure and connection loss while raised. Keep the battery switch within reach. If motion does not stop, switch off and contact JARzLabs; do not use the rover on the floor until resolved.
- 6 Move a flat target in front of sonar and turn each sensed wheel by hand. Confirm corresponding readings/counts change using the component diagnostics if the app has no relevant display.
- 7 Close the cover carefully, then repeat the raised-wheel check. A working phone connection or sonar reading does not prove the motor battery path is connected.
- 8 Only after these checks pass, try a short slow drive in a clear floor area with an adult present. Recheck connection and stop behavior after restarting the apps or power.

Write down expected result, observed result and PASS or NEED HELP for every step. These instructions are a customer commissioning checklist, not a claim that this unit has already passed.

Troubleshooting one thing at a time

What happens	First checks
Board lights up but motors do not move	Stop; check motor battery, switch, driver supply, ENA/ENB and common ground. Logic power alone does not power motors.
Motors stop after closing the cover	Disconnect all power; inspect loose terminals, pinched motor leads and changed left/right connections.
One side is slow or buzzes	Stop; with power removed compare hand rotation, motor mounts, encoder clearance and channel wiring. Do not increase voltage by guesswork.
Sonar says NO ECHO	Check TRIG/ECHO, supply, target shape and board-specific input circuit. No echo is not a measured zero distance.
Speed counts stay still	Turn the wheel by hand; check disk alignment, DO rather than AO, supply, ground and the correct left/right pin. A blinking sensor LED alone does not verify the return signal.
Controller cannot find Robot	Keep both apps open on the same local network, allow local-network access and avoid guest-network isolation. Check Phone control mode where offered.

Previous builds reported slow right wheels and pivot buzzing, and motors stopping after enclosure work. Those reports do not establish a single repair or a universal battery choice. Resolve the failing component test before trying the full rover again.

Platform connection checks

If iPhone cannot install the app, check Xcode Team, signing, selected physical device, trust and Developer Mode. If BLE fails, confirm Bluetooth permission, ESP32 power, normal firmware and close any connected scanner. The test service is not the Robot service. If sensor values are wrong, check safe input levels and external bias on GPIO34/35 before changing pins.

Source-code access and development assistance

For source access, iOS development, Creature Lab or ball tracking, contact Jarzlabs24@gmail.com. Tell us your kit version, board, phone and the feature you want to explore. The ZIP includes component-test sketches and the matching low-level firmware for restoring the board; the complete Robot, Controller and Creature Lab source projects are supplied separately with assistance.

Creature Lab image generation needs a local server on your own computer, internet access and your own private image-generation credential. Keep credentials out of email and shared files. Basic driving does not require that server. Start with manual motion and stop tests before exploring vision features.

Contact Jarzlabs24@gmail.com for help. Include your kit version, board label, app version, test name and result. A parent or adult helper can send the request.