

JarzRover iPhone-ESP32 - Wiring Reference

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.

Left/right means looking forward from behind the rover. These are logical circuit diagrams, not physical header-position drawings. Trace the labels printed on your board.

Work together safely

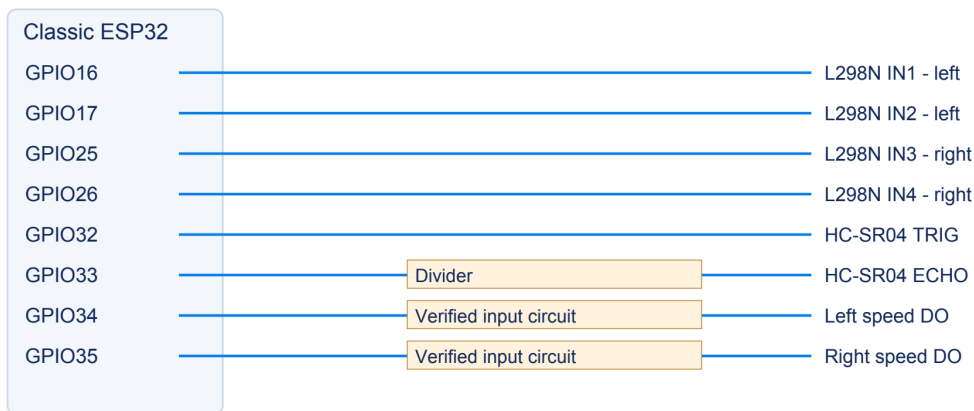
An adult checks wiring, soldering, power supplies and powered motor tests. Children can label parts, read steps and record results. Disconnect battery and USB before moving wires. Keep all wheels raised during motor tests; keep fingers, hair and loose clothing clear. Stop immediately for buzzing, a stalled 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.

Signal connections

iPhone / ESP32: current iOS sketch

Signal diagram only. Power and grounds are shown separately. Read board labels, not header positions.



BLE from iPhone; USB supplies ESP32 for bench tests

Do not route motor current through the controller. Connect controller GND to driver GND.

Current iOS ESP32 signal diagram with level conditioning on returning sensor signals

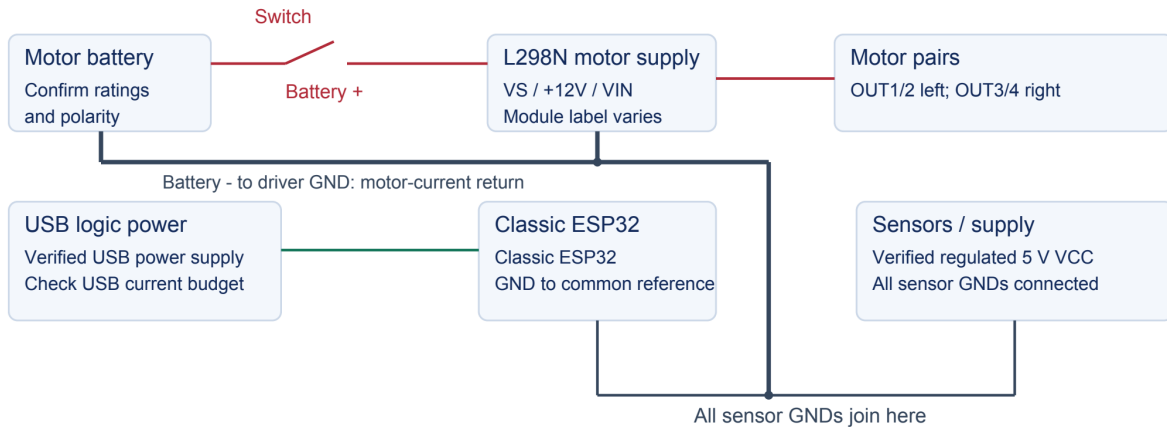
Classic ESP32 GPIO	Connect to	Required interface
16 / 17	L298N IN1 / IN2, left	Verify module accepts 3.3 V logic
25 / 26	L298N IN3 / IN4, right	Verify module accepts 3.3 V logic
32	HC-SR04 TRIG	Check sensor recognizes 3.3 V trigger
33	HC-SR04 ECHO through divider	Never connect 5 V ECHO directly
34 / 35	Left / right speed DO	Use the matching input circuit below
GND	Driver and sensor grounds	Shared reference, separate motor-current return

The dedicated iOS sketch enables BLE, sonar and front speed sensors; disables battery sensing, indicators and status LEDs. GPIO36 is defined for battery sensing but is unused with sensing disabled. GPIO34/35 are input-only with no internal pull-up or pull-down. A firmware INPUT_PULLUP setting cannot create a resistor on these pins. GPIO16/17 must be available on your exact module; do not substitute a PSRAM-equipped board without checking its pin use.

Power, common ground and sensor distribution

Power and common ground

Conceptual bench arrangement. Verify exact module supply ratings and regulator/enable jumpers first.



No connection shown between L298N 5V and the USB-powered board: do not join regulator outputs.

Sensor supply source must be verified for your board. No raw battery to 5V, 3V3, A7 or GPIO.

Power circuit: switched motor supply, separate USB logic supply, and common ground star

This diagram deliberately keeps the L298N 5V terminal disconnected from the USB-powered board. Use the ESP32 USB input and a verified USB supply for bench tests. A mobile ESP32 build needs a separately verified supply arrangement. Do not feed a 2S pack into 5V, 3V3, A7 or a GPIO.

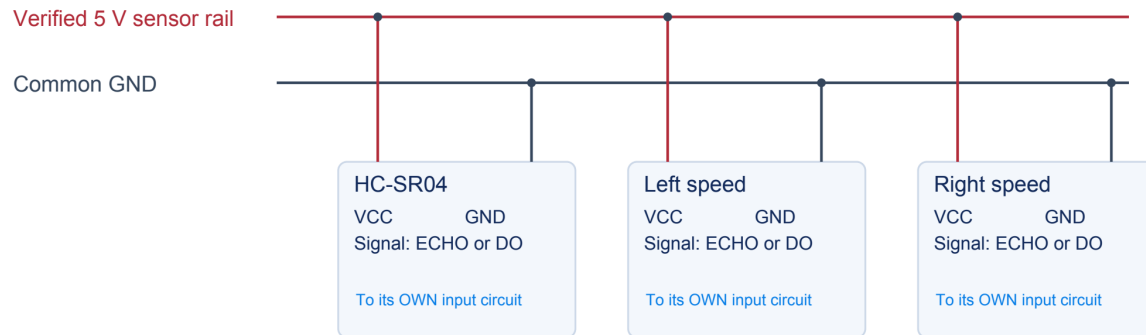
The battery feeds the driver's **motor supply** through the switch; battery negative returns directly to driver GND with suitable motor-current wire. The controller and sensors join that ground reference with signal-ground wires. Do not route motor return current through the controller or thin sensor leads. A label such as +12V on an L298N module names its supply terminal; it is not an instruction to use a 12 V battery.

Use one verified regulated 5 V sensor source. On this ESP32, verify whether the exact board exposes USB 5 V and its available current; never assume a pin marked VIN is a safe 5 V output. Use a verified regulated supply when required. Do not join independent regulator outputs or back-feed USB.

Three-way wire splits

Three sensors: separate VCC and GND splits

Use insulated splices or terminal blocks. This is a distribution diagram, not physical connector order.



Power lines do not contain signal-divider resistors. Each divider connects its lower resistor to GND.

For 3.3 V-powered speed modules, use a separate verified 3.3 V feed instead of the 5 V branch shown.

Separate sensor VCC and GND branches with individual return-signal circuits

Keep the power split and ground split separate. Use insulated soldered splices, suitable connectors or terminal blocks with strain relief; the working reference does not require a breadboard. Each sensor gets VCC and GND. A resistor divider goes on an individual **signal return**, not in the shared VCC feed. Its lower resistor joins common ground. Sonar, left speed and right speed each need their own signal circuit where conditioning is required.

Check the L298N module before powering it

- IN1-4 carry control signals; OUT1-4 carry motor power. Never connect a motor to a GPIO.
- ENA/ENB enable the two bridges. This reference uses PWM on IN1-4, so both bridges must be enabled as specified by the exact module. Verify its enable jumpers; do not confuse them with the regulator jumper.
- The module's 5V-EN/regulator jumper determines how its 5V terminal is used. The correct position depends on the regulator, battery range and external supply. An adult must identify the board and measure the intended rail before connecting it.
- Check motor stall current, both motors' combined channel demand, driver cooling, wire/switch/connector ratings and battery protection. A generic L298 chip rating does not certify a low-cost module or a four-motor build.
- Record battery type, full-charge voltage, connector polarity, enable/regulator settings and actual motor terminal assignments. These details are not fully established by the project photos.

Sensor return circuits

ESP32: two different signal circuits

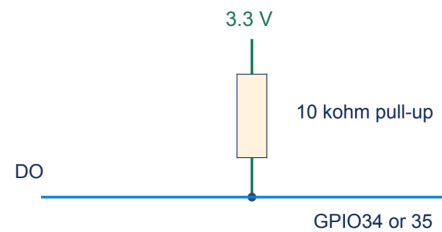
Choose by actual output type. Duplicate separately for each applicable signal. Never put these in the shared VCC feed.

A. Confirmed 5 V push-pull output



1% resistors; regulated input ≤ 5.1 V.
Nominal 5.00 V becomes about 3.21 V.
HC-SR04 return goes to GPIO33.

B. Verified open-collector speed DO



Only if no onboard path pulls DO to 5 V.
Sensor ground joins common GND.
34/35 have NO internal pull resistors.

Do not use A on an unknown weak open-collector output. Do not use B with an unknown 5 V pull-up.

Adult verifies supply, HIGH and LOW levels before connecting GPIO. Use a verified level shifter if uncertain.

Two separate sensor-input circuits: a 5 V push-pull divider and a verified open-collector 3.3 V pull-up

Circuit A: confirmed 5 V push-pull signal, including the usual HC-SR04 ECHO. Use $R1 = 10$ kohm from signal to GPIO junction and $R2 = 18$ kohm from junction to ground, both 1%. The design calculation is $V_{gpio} = V_{in} \times 18 / (10 + 18)$: about 3.21 V at 5.00 V. With a regulated signal supply no higher than 5.1 V and worst-case 1% resistor tolerances, the calculated maximum is about 3.30 V. Verify actual HIGH/LOW levels with a meter before attaching ESP32; a meter is not a complete transient check. A divider does not regulate sensor power.

Circuit B: verified open-collector speed output. Use a pull-up to 3.3 V, typically 10 kohm, only after confirming the module output can safely be referenced to 3.3 V and has no pull-up/indicator path to 5 V. A module powered at 5 V may contain such a path. Do not simply add a 3.3 V pull-up to an unknown module. Identify the circuit or use a suitable verified level shifter. A module that works completely at 3.3 V can use its confirmed safe DO output; an external pull-up is needed if that output is open collector and lacks one.

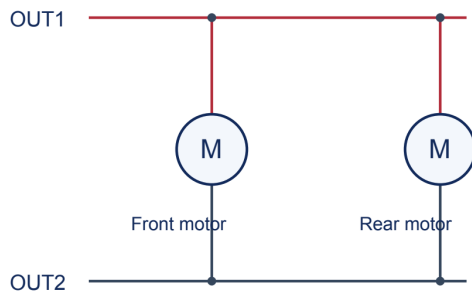
For a confirmed 5 V push-pull speed output, use a separate Circuit A for each DO. Do not apply that divider blindly to a weak open-collector output: the loaded HIGH may become too low. Use DO rather than AO. Turn the wheel by hand and check both states. Earlier chat advice proposed 1k/2k dividers; those produce about 3.33 V at nominal 5 V and are not the conservative example used here. No resistor change or new sensor circuit was physically tested during this documentation update.

Motor pairs and direction

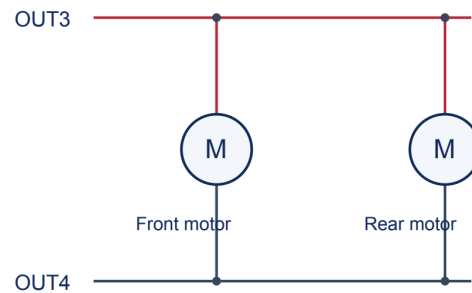
L298N: two parallel motors on each side

Logical motor circuit. Installed polarity depends on mirrored motor orientation; test with wheels raised.

LEFT channel A



RIGHT channel B



Motor leads go only to driver outputs. Do not connect them directly to ground or a GPIO.

Preserve the reference polarity. Disconnect battery before reversing leads or swapping output groups.

Motor-pair circuit showing two parallel motors per L298N bridge, with polarity checked by raised-wheel testing

Each left motor connects across OUT1/OUT2; each right motor across OUT3/OUT4. Connect motors in parallel within a side, not in series and not between left/right channels. The physical motors are mirrored, so wire colors alone do not prove forward direction. The inherited OpenBot DIY instructions use OUT1/OUT2 for left and OUT4/OUT3 for right polarity; the actual JarzRover polarity must be checked on the assembled unit.

Test one side at a time with all wheels raised. Both wheels on that side must propel the rover in the same direction. The native iOS app applies direction inversion in its BLE command path; it has no Android-style Reverse motor direction setting. Check the matching app build before changing a working motor pair. Power off and disconnect the battery before changing motor leads.

Adult check before powered tests

- 1 Disconnect all power, trace each wire against this guide and photograph the completed connections.
- 2 Confirm battery polarity, regulated logic/sensor supplies, common ground and driver jumper settings for your exact module. Never measure resistance on a powered circuit.
- 3 Verify each sensor return remains within safe 3.3 V levels before attaching ESP32 inputs. Do not connect 5 V ECHO directly.
- 4 Run [Component Tests](#), then restore the normal firmware and follow [Installation Guide](#).

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.

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.

Source verification covers firmware definitions and diagram consistency, not electrical measurements or a physical test of your kit. Pin assignments are matched to the dedicated firmware/openbot_ios/openbot_ios.ino source. Module power and jumper details still need checking on the actual supplied parts.