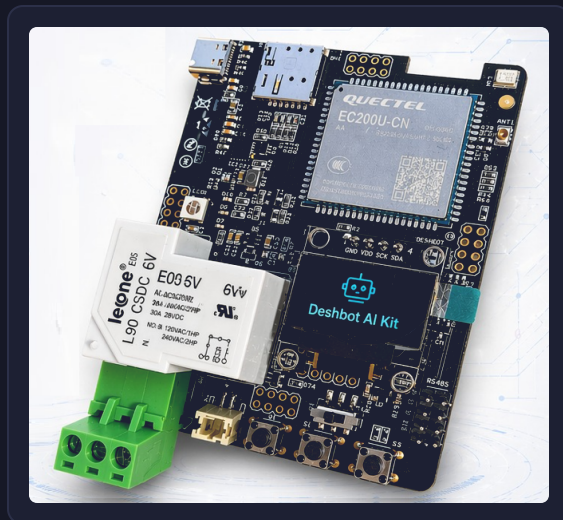


Power it on. Open Serial Monitor. Meet Deshbot AI.

Deshbot AI Kit is a compact, AI-enabled embedded hardware platform built around the ESP32-S3 — compute, sensing, audio, display, connectivity, and high-power switching all on one board, for Edge AI and IoT prototyping without a cloud dependency.

To get started: connect the board over USB-C, flash the firmware, and open the Serial Monitor in the Arduino IDE, ESP-IDF, or PlatformIO — everything you need prints right there on screen.

github.com/Deshbot23/Deshbot-AI-Advanced



BEFORE YOU START

What's in the box

Check every part against this list before your first power-up.

ITEM	QTY
ESP Development Kit (Deshbot AI Kit board)	1
Battery	1
Mod02	1
Jumper wires	10
USB-C cable	1
ArduCam	1
Memory card (32 GB)	1
OLED	1
Speaker	1
LAN module	1
3D printed case	1

One board, every subsystem

The ESP32-S3 sits at the center, with sensing, audio, display, storage, communication, and actuation all wired directly to it.

ESP32-S3, dual-core LX7 @ 240MHz

Wi-Fi 802.11 b/g/n

Bluetooth LE 5.0

Quectel EC200U-CN 4G LTE Cat-1

SHT40 Temp/Humidity

LIS3DH Accelerometer

INMP441 I2S Microphone

MAX98357 Class-D Amp (~3W)

SSD1306 OLED, 128×64

RS485 (industrial)

30A SPDT Relay, 6V coil

Camera Interface

MicroSD Slot

LAN Module Interface

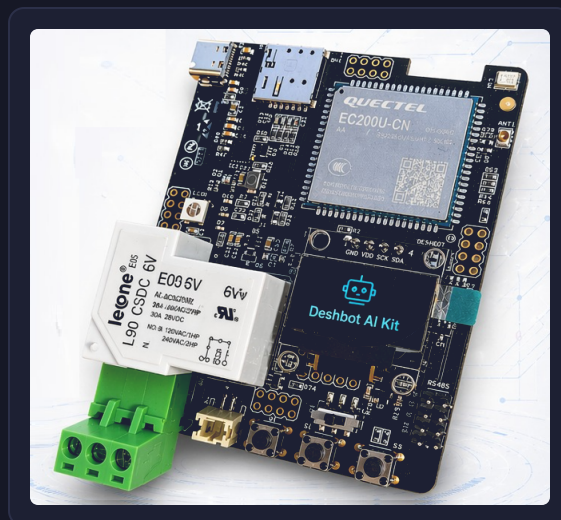
LoRa Module Support

Onboard Li-ion Charging

3 Navigation Buttons

RGB Status LED

Board dimensions: ~85mm × 60mm, PCB mounting holes. Some interfaces (camera, LAN, LoRa, GNSS on the EC200 variant) are present on the board but are not exercised by the demos in this guide — see the GitHub repo as those layers mature.



GETTING STARTED

From box to first boot

1

Power up

USB Type-C, or the onboard battery.

2

Install board package

Add ESP32 support in Arduino IDE, or use ESP-IDF / PlatformIO directly.

3

Select the board

ESP32-S3 Dev Module.

4

Flash the firmware

Build and upload from the repo.

5

Open Serial Monitor

115200 baud on the console UART (GPIO43/44) — see the boot banner.

6

Pick a demo

SW1 = up, SW2 = down, SW3 = select, from the OLED menu.



```
=====
                DESHBOT AI ADVANCED
        Modular ESP32-S3 AI + IoT Development Kit
=====
Name           : Deshbot_AI/yourname
Serial No      : DSB-AC29
MAC Address    : 48:27:E2:07:38:C0
Firmware       : led_relay_oled
-----
GitHub Repo    : https://github.com/Deshbot23/Deshbot-AI-Advanced
Website        : https://deshbot.com/
=====
        Hold SW1 for ~1.5s at any time to return to the menu.
=====
```

This banner prints once, first thing after reset — an example, not a live capture; your board's serial number and MAC will differ.

THE DEMO MENU

Start with these — verified on real hardware

The full menu has 26 entries, several still code-complete but not yet hands-on confirmed. These are the ones worth starting with.

DEMO	WHAT IT DOES
LED	Onboard LED blinks; OLED mirrors its live ON/OFF state.
Relay	LED + relay toggle together with an audible click; OLED shows both live states.
Sensors	Auto-refreshing temperature/humidity (SHT40) and accelerometer (LIS3DH) status on OLED and serial, every 3s.
Voice	Mic + speaker only, fully offline — speaks "Hello" at boot, then a sustained sound triggers a spoken "How can I help you?" reply.
Boot Report	Robot-eyes intro, then a live system check narrated one subsystem at a time — sensors, OLED, RGB, relay, switches — ending with SW1/SW2/SW3 presses reported back.
Games	Three OLED pixel games — a space shooter, a Tetris clone, and a 3-lane traffic dodger. SW1/SW2 steer/move, SW3 fires/rotates/honks.
BLE Scanner	<i>Deshbot_AI_BLE_Scanner</i> — lists nearby BLE devices with live RSSI on the OLED; SW1/SW3 scroll, SW2 opens detail view. User-confirmed working on real hardware.
Mic Wave	Live oscilloscope trace of the microphone signal, plus FREQ/PEAK/RMS readouts, fully offline.
Temp Graph	Scrolling strip-chart of live temperature readings with labeled axes, fully offline.
Voice Control Relay	Fully offline, speaker-dependent — enroll "relay on"/"relay off" once, then switch the relay by voice. SW3 re-trains.
Cam Capture	Arducam Mega 3MP still capture — SW2 takes a photo (RGB flash), saves a numbered JPEG to the SD card. Verified working on hardware.
LAN & WiFi	Wired W5500 Ethernet + WiFi with automatic no-reboot failover; SW2/SW3 force a mode, SW1-tap resets the active interface. Driver and link-up verified; a full DHCP/failover run is not yet confirmed live.

There's more once you're ready

Voice commands, a Deshbot chatbot, BLE sensor broadcasting, Wi-Fi/Ethernet failover, and OTA updates all exist in the firmware — most need a Wi-Fi network and a free Groq API key in `main/credentials.h`. Several of these are code-complete but not yet confirmed end-to-end on hardware; check the repo's `AGENT_HANDOFF.md` for current status before relying on one.

PIN REFERENCE

Onboard peripheral pin map

Source of truth in code is `main/board_pins.h`; this table is its human-readable mirror.

SIGNAL	PIN(S)	NOTES
LED	42	Push-pull output, active high.
RGB LED (WS2812)	1	Driven via RMT, 1 LED.
Relay coil	39	Output; true = energised (audible click).
OLED SDA / SCL	19 / 20	SSD1306, addr 0x3C (0x3D if SA0 tied high).
Accelerometer	I2C 0x19	Shared I2C bus with the OLED.
Temp/Humidity (SHT40)	I2C 0x44	Shared I2C bus with the OLED.
Mic (INMP441) WS / SCK / SD	8 / 17 / 18	I2S RX, 16kHz mono.
Speaker (MAX98357) BCLK / LRCLK / DIN	15 / 16 / 7	I2S TX, 16kHz mono 16-bit; channel enabled only per clip.
SW1 / SW2 / SW3	35 / 36 / 37	Active-low, internal pull-up. Up / Down / Select.
Console UART TX / RX	43 / 44	115200 baud.
EC200U 4G module TX / RX	5 / 6	UART, AT commands.

Reserved — don't repurpose

GPIO43/44 are the console UART. GPIO19/20 carry the shared I2C bus for four devices (OLED, accelerometer, temp/humidity). The two I2S peripherals own six pins between them.

SAFETY

Real electricity, a battery, and a 30A relay

Use the relay carefully with high voltage

The onboard relay is rated 30A on a 6V coil, SPDT contact. Ensure proper grounding and avoid powering high loads without isolation.

Do

- ✓ Power over USB-C or the battery circuit as documented.
- ✓ Ground high-voltage loads switched through the relay properly.
- ✓ Keep the board on a non-conductive surface.

Don't

- ✗ Switch high loads through the relay without proper isolation.
- ✗ Leave the battery charging unattended.
- ✗ Touch relay contacts or terminals while a load is live.

TROUBLESHOOTING

If something's not working

SYMPTOM	LIKELY CAUSE / FIX
Upload fails	Confirm board is set to <code>ESP32-S3 Dev Module</code> and the correct COM port is selected.
Serial Monitor shows garbled text	Baud rate mismatch — set it to 115200 .
Nothing prints at all	Wrong COM port, or the board needs a reset press after opening the monitor.
Sensors show "unidentified"	SHT40/LIS3DH not detected on the I2C bus — check the shared SDA/SCL wiring (GPIO19/20) isn't disturbed.
Voice/chat demos report "NO WIFI" / "CHECK API KEY"	Fill in real values in <code>main/credentials.h</code> (copied from <code>credentials.example.h</code>).

GitHub Repository [deshbot.com](#)

Deshbot AI Kit — First-Time User Guide. See the repository's `README.md` and `AGENT_HANDOFF.md` for full technical and verification detail.