SOFTAP WI-FI PROVISIONING

Give the board a Wi-Fi address

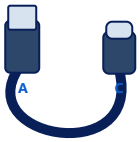
01

Use the board's temporary setup network to send a lab router's details, then prove the board can reconnect after a restart.



ESP32-S3-N16R8 board

Hosts setup Wi-Fi, then joins the router.



USB data cable

Carries power, files and console messages.



Computer + USB terminal

Installs the application and reads results.



Phone or Wi-Fi laptop

Opens the local setup form.



Private lab router

2.4 GHz WPA2-Personal; temporary credentials.

Before you begin: complete ML-C03/C04. Have working MicroPython v1.29.0 for ESP32-S3, a working USB REPL and `mpremote` on your computer. Use a private 2.4 GHz WPA2-Personal lab router with a temporary passphrase.

You will do

Install three files, open a local form, observe the result and test a restart.

You will prove

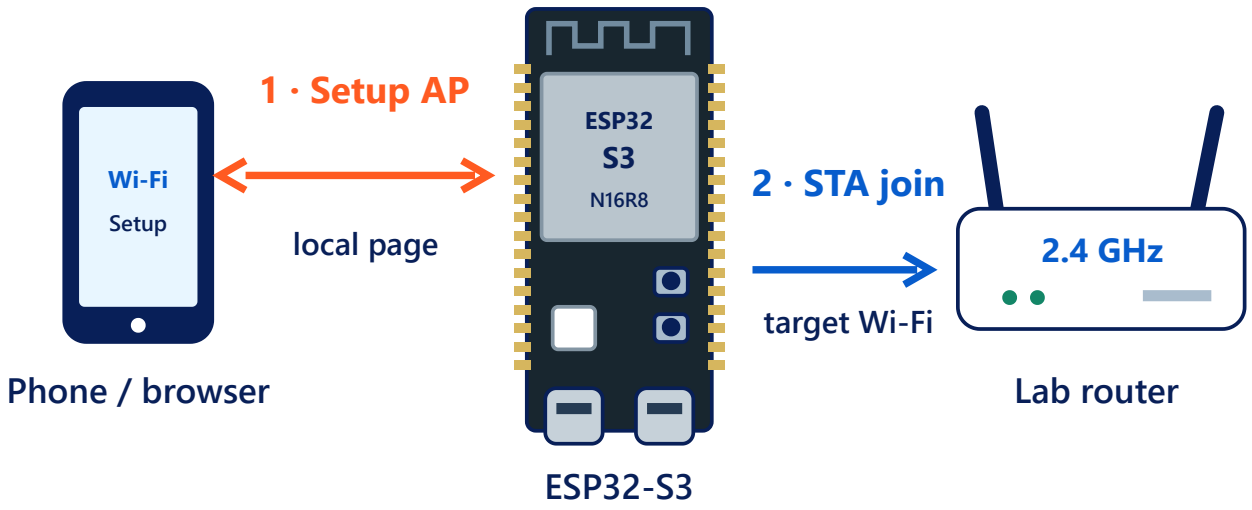
The board has a router connection and an IP address, then repeats that join from its saved file.

The phone and computer may be the same laptop. Keep USB available for the setup key, redacted results and recovery. No extra circuit or external supply is needed.

UNDERSTAND THE ROUTE

One board, two network roles

02



AP = access point: the board hosts a temporary Wi-Fi network. STA = station: the board joins your router as a client.

Provisioning means giving a device the settings it needs. First your browser reaches the board over its setup AP. Next the board uses the submitted SSID (network name) and passphrase to join the lab router.

Setup link

Browser → board. The local page is stored on the board; internet access is unnecessary.

Router link

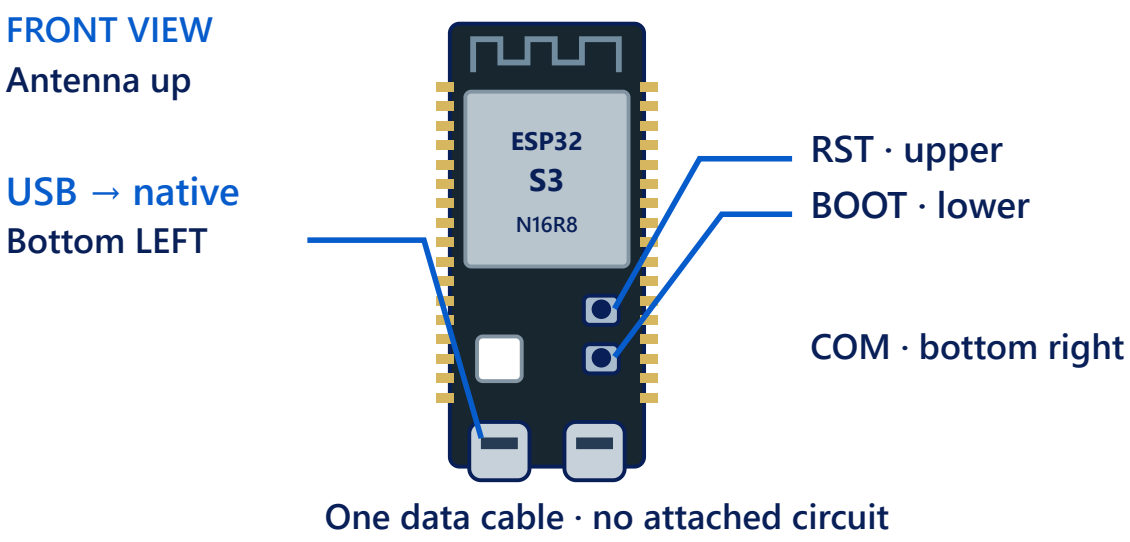
Board → target router. A local IP address shows local connectivity; it does not prove internet access.

Keep the two networks separate. Do not enter `Molo-Setup-C11` as the target router name. Its key gets your browser to the form; the router's passphrase gets the board onto your lab network.

PREPARE OVER USB

Install a small, known application

03



Place lab_config.py, softap_lab.py and main.py from this course's `examples` folder in one computer folder. The included example guide repeats these steps. Leave unrelated programs and configuration backed up privately.

1. Connect the native USB socket. Close other serial tools. Use `mpremote connect list` to find the runtime port.
2. If `main.py` already exists, copy it to a new private backup filename before replacing it. If it is absent, skip that backup command.
3. Run the commands below from the folder containing the three files. Replace `PORT` with your actual port, such as `COM7`.

```
mpremote connect PORT fs cp :main.py ./main-before-c11.py
mpremote connect PORT fs cp lab_config.py :lab_config.py
mpremote connect PORT fs cp softap_lab.py :softap_lab.py
mpremote connect PORT fs cp main.py :main.py
mpremote connect PORT repl
```

Install `main.py` last. In the REPL, press Ctrl+C twice, then Ctrl+D to restart and see startup messages. Rediscover the port if USB reappears under a different name.

OPEN A LOCAL PAGE

Join the temporary setup Wi-Fi

04

TEMPORARY SETUP AP



Molo-Setup-C11

Join with USB key

No internet is normal

Browser → board

TARGET LAB ROUTER



Classroom-Lab

2.4 GHz · WPA2

Board joins here

Board → router

1. Read the computer's USB console. It shows `SETUP SSID: Molo-Setup-C11`, a fresh **SETUP KEY** and `http://192.168.4.1/`.
2. On your phone or laptop, open Wi-Fi settings and join `Molo-Setup-C11` using that temporary key. Keep the key private.
3. Choose to stay connected if the operating system says “No internet.” Temporarily disable an automatic switch to mobile data or another Wi-Fi network.
4. Open the exact **http://192.168.4.1/** address in the browser's address bar. Use HTTP, not a search box or an assumed HTTPS address.

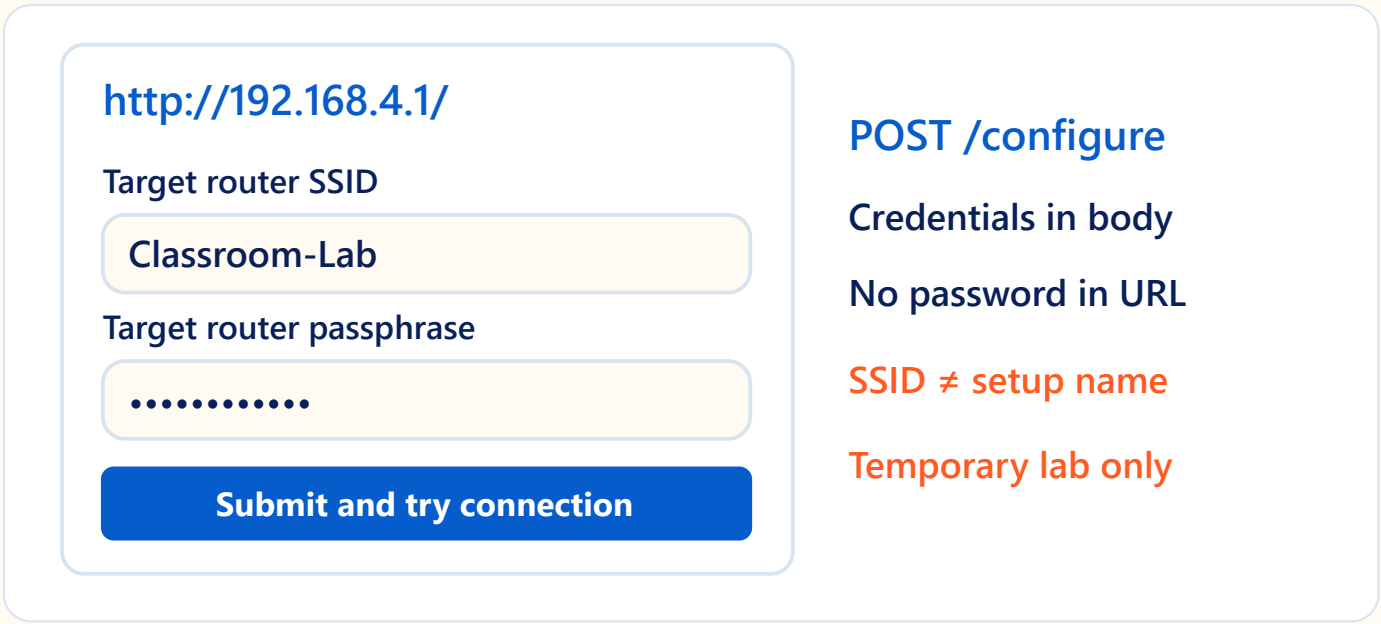
> **Expected result:** a page headed “Connect the board.” There is no captive-portal detection or internet redirect in this example. Enter the address manually.

The setup window lasts five minutes and accepts one Wi-Fi client at a time. If it expires, press RST alone to start again and read the new temporary key. A successful saved configuration goes straight to the router on later starts.

A FORM, NOT A URL

Submit the router details

05



Enter the **target lab router's** network name and temporary WPA2 passphrase. Use the exact capitalization and spaces. The example supports a 1–32 byte UTF-8 SSID and an 8–63 character printable ASCII passphrase. It does not handle open networks, enterprise login or a 64-digit raw key.

1. Check that the name belongs to your 2.4 GHz lab router.
2. Select **Submit and try connection** once. The browser sends an HTTP POST body to `/configure`.
3. Wait for “Submission received,” then read the USB result. The form never echoes the password.

> **Use temporary lab credentials.** The setup AP uses a fresh WPA2 key, but this simple local page uses HTTP, without TLS. POST keeps values out of the URL; it is not end-to-end encryption. Do not use home, workplace or shared-account secrets.

The server checks its session token and validates the same field limits again. Do not paste credentials into browser URLs, screenshots, prompts, terminal commands or support logs.

READ THE RESULT

A receipt is only the first signal

06

SUBMISSION RECEIVED

HTTP 202

Form accepted

Connection still pending

CONNECTION RESULT

isconnected() → True

valid IP → save file

USB: connected

First send the receipt; then try the router.

The application sends and closes the HTTP response **before** it starts the router join. This gives the browser its receipt before the board's radio changes roles.

The station attempt lasts at most 15 seconds. The code checks `sta.isconnected()` and a nonzero address from `sta.ifconfig()[0]`. Only then does it save `/lab_wifi.json` and close the setup AP.

```
{"state": "connected", "configured": true,
 "ip_assigned": true}
```

This is the connection result. If setup Wi-Fi disappears, read USB first. AP and STA share a radio; a channel change may interrupt your browser. Rejoin the router on your phone after success. Internet access remains a separate test.

PROVE THE SAVED CONFIGURATION

Restart without the form

07



/lab_wifi.json

schema 1 · SSID · password

RST alone · BOOT released

USB: reconnected + IP assigned

A restart must work without resubmitting the form.

1. After a `connected` result, keep the lab router available. Leave BOOT released.
2. Press the upper **RST** button once. If the runtime port disappears, rediscover it and reopen your USB terminal.
3. If you missed startup output, use Ctrl+C and Ctrl+D in the REPL to make a clean application restart while watching.
4. Observe the `reconnected` result below. Do not submit the web form again.

```
{"state": "reconnected", "configured": true,
 "ip_assigned": true}
```

Saved file

Exactly `schema`, `ssid` and `password`, with schema value `1`. ML-C09 uses this same lab file format.

Keep it private

The file contains readable credentials. Do not print it or add it to Git. Protect backups, and delete lab settings when finished.

At the REPL, `print(softap_lab.status())` reports only saved-state, link and IP booleans. A status read alone does not prove that a restart loaded the file.

RECOVER A FAILED ATTEMPT

Retry one layer at a time

08

JOIN FAILED

Read redacted USB result

Rejoin setup AP

Reopen 192.168.4.1

Correct target details

CLEAR / RETRY

Ctrl+C → REPL

softap_lab.clear()

RST → fresh setup key

Remove lab file only

Observation	Next action
No setup network	Check USB power and the current key. A saved configuration may already have joined the router. Read USB before forcing setup.
No local page	Confirm the phone is still on setup Wi-Fi. Type the HTTP address manually; disable automatic Wi-Fi/mobile-data switching for the lab.
`join_failed`	Rejoin setup Wi-Fi and reopen the form. Check the target SSID, temporary key, 2.4 GHz band, WPA2 mode and router availability.
`save_failed`	The code disconnects STA. Keep USB, back up private files and check free filesystem space before retrying.
`setup_expired` / `setup_error`	Return to the USB REPL. Check installation and supported firmware, then restart the application.

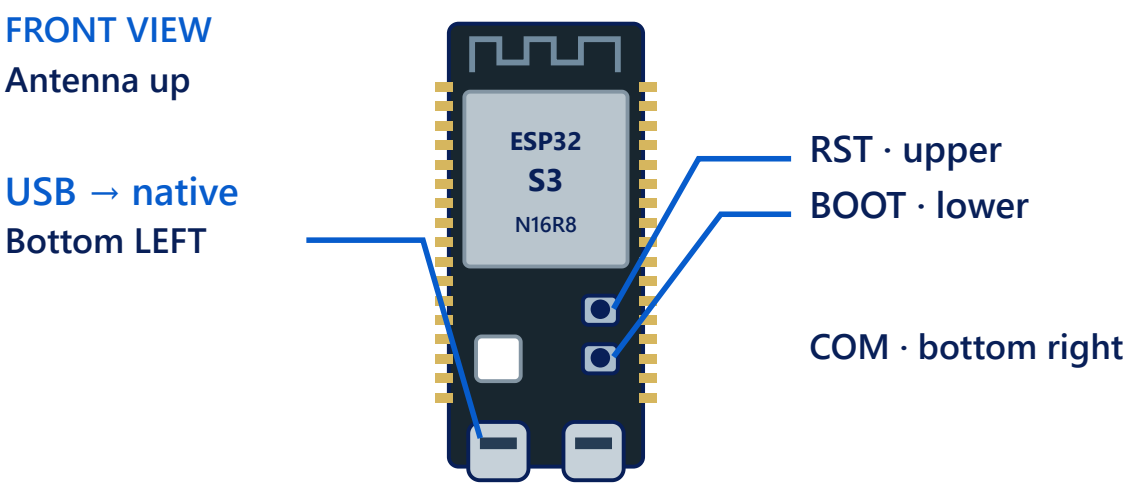
A failed new attempt does not replace a previously saved valid file. On the next restart, the application tries that file first; if it cannot join, it opens setup again.

> **Share only the layer and state:** for example, “form receipt received; USB says join_failed.” Remove the temporary setup key and any network details before sharing a screenshot.

A DELIBERATE RECOVERY PATH

Clear the lab and start fresh

09



Clearing is a separate action. It removes this lab's saved settings; it does not erase the firmware or unrelated programs.

1. In the USB terminal, press Ctrl+C twice to stop an active setup loop and reach `>>>`.
2. Run the commands below. The result says `cleared` and `configured: False`.
3. Press RST alone. The board should open a fresh setup AP and print a new key. Rejoin using that new key.

```
import softap_lab
print(softap_lab.clear())
```

Exactly what is cleared: `/lab_wifi.json` and its named temporary save file. The helper also disconnects the station and closes the setup AP. It does not claim to erase credentials belonging to every other firmware or application.

Open setup without deleting the saved file

```
import softap_lab
softap_lab.run(force_setup=True)
```

Use this after stopping the application at the REPL. A successful new submission replaces the lab file. An unsuccessful attempt leaves the previous file available.

When the lab ends, clear the device file and remove any private computer backups you no longer need. Restore your previous `main.py` only from your own checked backup.

PRACTICE AND SUCCESS CRITERIA

10

Run a complete provisioning check

1 · Receive	2 · Connect	3 · Restart
202 receipt	Link + valid IP	Saved-file join
No success yet	Save config	reconnected
Temporary setup AP	Local network only	No form resubmission

1. Start with no saved lab configuration. Join setup Wi-Fi and open the form.
2. Use an intentionally wrong temporary lab passphrase. Confirm `join_failed` after the bounded attempt; rejoin setup if needed.
3. Submit correct lab details. Confirm `connected` with `ip_assigned: true`. The setup AP closes.
4. Restart with BOOT released. Confirm `reconnected` without resubmitting.
5. Clear the configuration and restart. Confirm fresh setup is available again.

> **Success record:** write only the observed state sequence, whether an IP was assigned and whether restart worked. Do not record either passphrase.

Check your understanding

1. Your phone says “No internet” on setup Wi-Fi. Is setup necessarily broken?
2. The browser shows HTTP 202. Has the router join succeeded?
3. What proves that the saved file works?

Answers

1. No. The page is local to the board.
2. No. Read the link and IP result over USB.
3. A clean restart reports reconnected without another form submission.

Keep USB recovery available as you compare SoftAP with BLE and SmartConfig in the next courses.