START HERE

Make the board appear first

01

Before debugging your program, prove that the computer can see the board. In about **10 minutes**, you will identify a data cable, find the board's current serial port and choose the next check from what you observe.



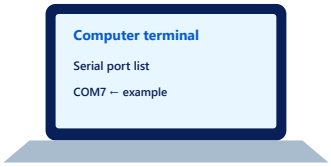
ESP32-S3 board

The device your computer must detect. Keep attached circuits and external power disconnected.



USB data cable

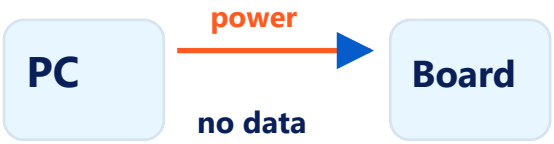
Both ends must fit your board and computer. Connector shape alone does not prove data support.



Computer

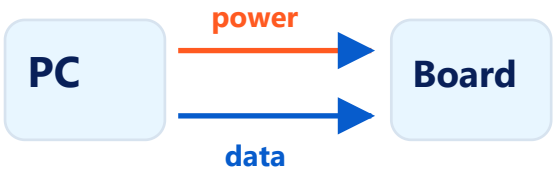
Shows connected USB devices and serial ports. It runs your serial tool later.

Power-only path



Light may turn on

Data-capable path



Can appear as a USB device

Connector shape, cable color and a power light do not prove data.

Conceptual paths, not a cable pin diagram. A known data-capable cable can carry both power and communication.

> **Try it:** choose a cable whose data function has already worked with a device, or whose manufacturer specifies data transfer. A glowing board light proves power arrived; it does not prove communication.

No coding background is needed. Port discovery works before Python firmware is installed. The optional Python reply check on page 6 requires MicroPython from ML-C03.

IDENTIFY THE ROUTE

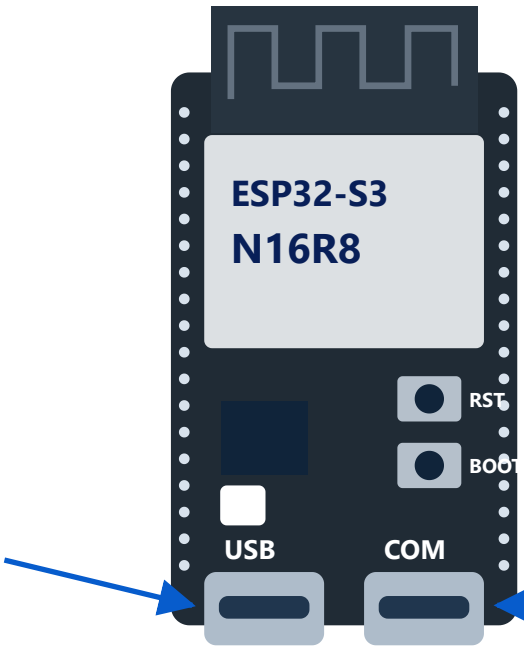
02

Use the native USB socket

Hold the board with its **components facing you, antenna up and sockets down**. This course uses the left socket, named **USB**. It connects directly to the ESP32-S3 USB interface.

Front view

Antenna up
Sockets down



USB · native

Use this socket

Upper: RST
Lower: BOOT

COM · bridge

Component side toward you; back-view left and right are reversed.

The socket names are printed on the back. Turning the board over reverses left and right.

USB · native route

Left in this front view. The chip can present a USB serial interface, which the computer exposes as a serial port.

COM · bridge route

Right in this front view. A separate USB-to-UART bridge carries serial data. Its driver depends on the exact detected bridge.

Keep one route: use one cable to native USB. Leave COM and external power unplugged. Do not change wiring while USB is connected.

Point, then connect: find USB, COM, upper RST and lower BOOT. Read the back labels if needed, then return to the front view before inserting the cable.

UNPLUG → CONNECT → COMPARE

03

Find the port by comparison

Enumeration means the computer detects a USB device and learns what it offers. A **serial port** is the named connection your serial tool opens. Find it by comparing the computer's list before and after plugging in this board.

Before · unplugged

Ports (COM & LPT)

COM4 existing device

After · connected

Ports (COM & LPT)

COM4 existing device

COM7 new board port

Example list: the port that appears and disappears with this board is yours.

Illustrated Windows example. COM7 is not a fixed board address; use the name that actually appears on your computer.

Windows

Open Device Manager. Expand **Ports (COM & LPT)**. Watch for a new entry when USB is connected. Also inspect **Other devices** or USB entries if no serial port appears.

macOS

Compare `ls /dev/cu.*` in Terminal before and after connection. A native USB port often contains `usbmodem`. System Information → USB can help check device detection.

Linux

Compare `ls /dev/tty*` before and after connection. Native USB often appears as `/dev/ttyACM0`; a bridge may use `ttyUSB`. These patterns are hints, not proof of which board you selected.

1 · Unplug

Note the existing devices.

2 · Connect

Use native USB and a data cable.

3 · Confirm

Unplug once more: the matching entry should disappear.

Write down: the socket used and current port name. Ignore unrelated ports that remain present when this board is unplugged.

MATCH THE DETECTED DEVICE

04

Install a driver only for a reason

A **driver** lets the operating system work with a device function. A missing serial port may involve a driver, but a bad cable, wrong socket or application state can cause similar symptoms.

Native USB

USB serial class

Often uses the OS class driver

Check the detected device first

JTAG and serial are separate functions

COM / bridge route

Identify the exact bridge

Hardware IDs + device description

Use its official vendor driver

Do not guess a chip from its shape

No random driver packs. Match the device, operating system and function.

Native USB serial and a USB-to-UART bridge are different driver routes. A JTAG driver is not automatically the driver needed for the serial console.

For the native USB route

On current Windows systems, compatible USB serial-class devices can use the included `usbser.sys` driver. macOS and Linux also provide USB serial support. First inspect the actual entry and its error message; do not install a random bridge driver just because there is no Python prompt.

If a device appears with a warning

1. Confirm that the warning entry disappears when this board is unplugged.
2. On Windows, open its **Properties** → **Details** → **Hardware Ids**. Record the device description and IDs.
3. Use the board/device manufacturer's official instructions for that exact interface and your operating system. For the COM route, identify the bridge before choosing its vendor driver.

> **Do not guess:** a similar board photo, cable color or chip outline does not identify a bridge. Avoid unofficial driver bundles and replacement-driver tools unless the exact device instructions require them.

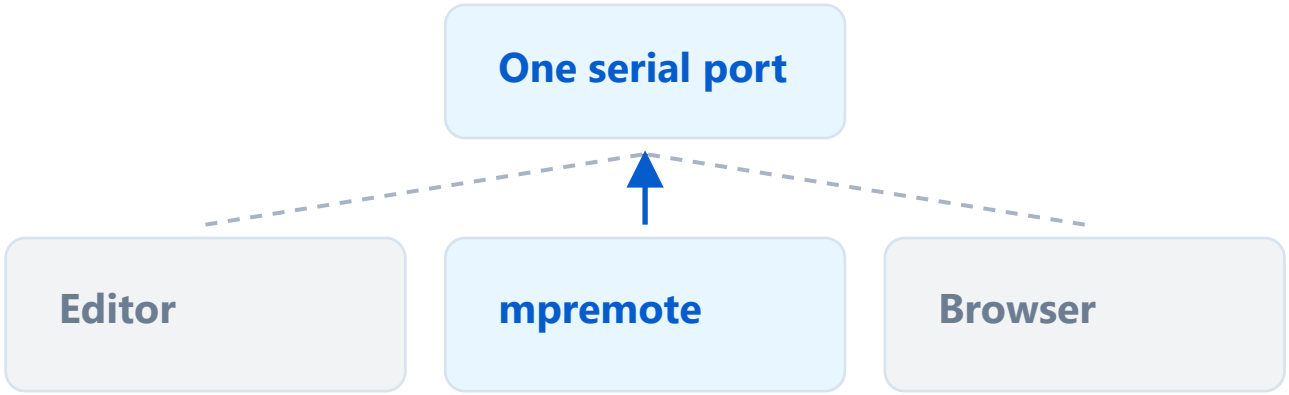
Linux "permission denied" can occur even when the driver and port exist. Follow your distribution's serial-access group instructions, with administrator help if needed. Do not solve it by making every device writable.

ONE OWNER; REDISCOVER AFTER CHANGES

Keep the right port free

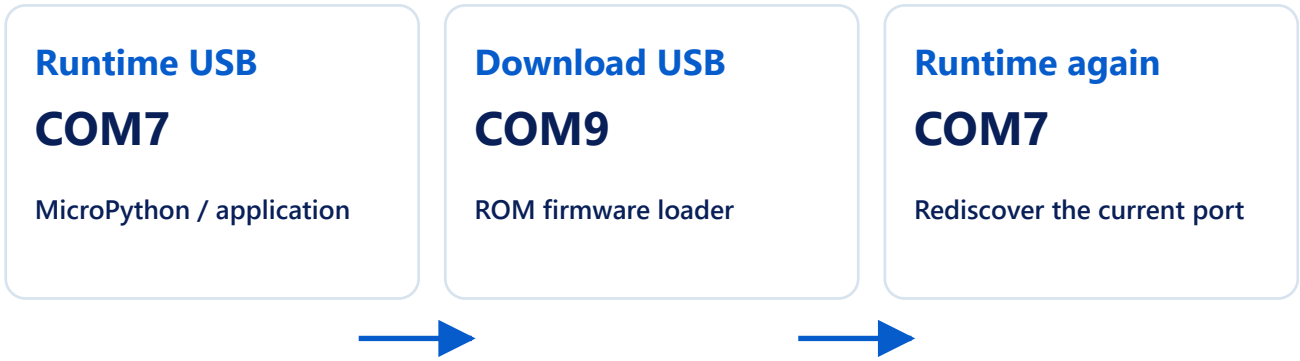
05

A port can be visible but unavailable because another program holds it. Exit the other connection before opening your next tool.



Close other serial sessions before the next tool takes over.

Close the serial session inside an editor or browser. A disconnected browser tab and a closed serial session are not always the same action.



Example names only. A changed mode can produce a different port.

Illustrated names only. Resetting or changing between download and normal runtime can make USB disappear and reappear under a different port name.

After a reset or firmware change: wait for USB to reconnect, compare the list again, then select the current port. Do not keep retrying an old COM number.

If normal USB does not appear: hold the lower BOOT button, press and release upper RST, then release BOOT. Compare the device list again. This enters the ROM download loader; it does not erase flash. To return to installed firmware, release BOOT and tap RST alone.

A port seen in download mode proves a different checkpoint from a working MicroPython prompt. Use ML-C03 to install firmware; this lesson does not erase or flash the board.

OPTIONAL CHECK · MICROPYTHON ALREADY INSTALLED

06

Ask a known console for a reply

Use this check only after installing compatible MicroPython in ML-C03. **mpremote** is a computer tool that opens the board's Python console. An unknown application may speak a different protocol; do not send Python commands blindly.

Prepare mpremote on the computer

With desktop Python 3.10 or newer, open a terminal in a lesson folder. On Windows PowerShell:

```
python -m venv .venv
.venv\Scripts\python.exe -m pip install mpremote
.venv\Scripts\python.exe -m mpremote connect list
.venv\Scripts\python.exe -m mpremote connect PORT repl
```

Replace `PORT` with the runtime port you identified. On macOS/Linux, create the environment with your installed Python command and use `.venv/bin/python` instead of the Windows executable path. These commands run on the computer, not after `>>>`.

Example MicroPython console

```
>>> 2 + 3
5
>>> print("USB link ready")
USB link ready
```

Type only Python statements after >>>. Ctrl-] exits mpremote.

Expected example, not a saved board transcript. Type the statements without the prompt symbols, one line at a time.

Check: press Enter, or Ctrl-C if a Python program is running. At `>>>`, `2 + 3` should return `5`. The greeting should match what you typed.

Exit cleanly: Ctrl-] leaves mpremote and releases the port. A UART terminal, if used later, normally needs 115200 baud; changing baud cannot fix a USB device that never appears.

PRACTICE AND SUCCESS CHECKS

07

Work from the first missing step

Do one check at a time. A power light, a USB entry, a serial port and a readable reply are four different observations.

1	2	3	4
Cable + socket	**USB device**	**Serial port**	**Application**
Data reaches the PC	Computer detects it	Driver exposes a port	Your tool opens it

Fix the first missing observation before changing Python code.

Stop at the first missing checkpoint. A program edit cannot repair a power-only cable.

Light on; no USB change

Check the native USB socket. Try a known data cable and a direct computer USB port. Remove an optional hub or adapter from the test path when possible.

USB entry; no serial port

Inspect the detected device and error. Check driver/function and current mode. A USB device may expose a function other than the serial console you expected.

Port exists; open fails

Close other tools. Check your account's serial-access permission. Rediscover the port after reconnecting.

Port opens; no Python reply

Confirm normal MicroPython runtime, current port and prompt. Try Enter or Ctrl-C. A blank terminal alone does not mean a missing driver.

Your turn

Unplug the board, note the list, reconnect and find its entry twice. Then explain what you would check if the power light came on but no USB device appeared.

> **Answer:** check the socket and a known data cable before debugging Python or installing arbitrary drivers.

- ☐ Identify native USB in the front view.
- ☐ Find the port by appearance and disappearance.
- ☐ Explain which observation points to the next check.
- ☐ Release the serial tool and record the current port.