LIBRARY · REFERENCE GUIDE

BOOT AND STRAPPING PINS

At reset the ESP32 reads a few strapping pins to decide how to boot. Why a stray pull on one can stop a board booting, and which pins to keep free.

ONE THOUSAND DRONES ENGINEERING TEAM · VERIFIED 2026-07

When the ESP32 comes out of reset, before it runs a line of your code, it reads a handful of strapping pins to decide how to boot: run the firmware in flash, or wait in the bootloader to be flashed. If your circuit holds one of those pins the wrong way at that instant, the board will not boot. A few pins are effectively sacred, and it pays to know which.

WHAT A STRAPPING PIN IS

A strapping pin is an ordinary GPIO that the chip samples once, at the moment of reset, to read a configuration choice. After that instant it goes back to being a normal pin you can use however you like. But the level it happens to sit at during that sample is latched, so whatever your circuit does to it at power-up is what the chip acts on.

RUN MODE VERSUS DOWNLOAD MODE

The key strapping choice is the boot mode. Left at its default level, the chip loads and runs the firmware already in flash. Held the other way, on the ESP32 the boot pin pulled low at reset, it enters the download bootloader and waits for a flashing tool to send a new program. That is exactly what pressing the `BOOT` button does while you reset the board.

- Espressif. esptool documentation: Boot Mode Selection (strapping pins, download vs run). docs.espressif.com

WHY A STRAY PULL BRICKS BOOT

Hang a hard pull-up or pull-down, an LED, or a sensor that drives the line onto a strapping pin, and you can override its boot-mode level at reset. The chip then boots the wrong way, or hangs, and the board looks dead even though it is fine. The fix is discipline: keep strapping pins free of hard pulls, or use them only for signals that are safely idle at reset. Check the datasheet's strapping-pin table before you assign one.

- Espressif. ESP32-S3 Series Datasheet (strapping pins and their reset-time levels). espressif.com

DEEP DIVE · THE AUTO-RESET CIRCUIT: WHY YOU DON'T PRESS BUTTONS TO FLASH

On most ESP32 boards you never touch a button to upload, thanks to a small circuit. The USB-to-serial bridge's DTR and RTS handshake lines are cross-wired, through two transistors, to the chip's reset (EN) pin and its boot strapping pin. The flashing tool wiggles those two lines in the right order to drop the chip into the bootloader and reset it, then lets it run again afterward. That is why esptool seems to flash on its own, and why a broken or missing auto-reset circuit means flashing by hand with the BOOT button.

AT RESET THE CHIP READS THE STRAPPING PINS AND PICKS ONE OF TWO BOOT PATHS.

CHECKPOINT

1. When does the ESP32 read its strapping pins?

a. Continuously while it runs

b. Only when you press a key on your keyboard

c. Once, at reset, before your code runs

ANSWER · C

The level is sampled and latched at reset; after that the pins are ordinary GPIO again.

2. Why can a resistor on the wrong pin stop an ESP32 from booting?

a. It draws too much power for the regulator

b. It can override a strapping pin's boot-mode level at reset

c. It slows the system clock down

ANSWER · B

A hard pull on a strapping pin changes the level the chip latches at reset, so it boots the wrong way.

3. What does holding the BOOT pin low during reset do?

a. Puts the chip into its download bootloader to be flashed

b. Permanently erases the firmware

c. Overclocks the CPU

ANSWER · A

That strapping level selects download mode, where the bootloader waits for a flashing tool.

- See it on a real board: the L1.01 build (its boot and reset circuit)
- Next: flashing firmware