

READING THE ESP32 PINOUT

Not every pin can do every job: some are input-only, some are strapping, some are tied to USB or the ADC. How to read the pinout before you wire, and pick the right pin.

ONE THOUSAND DRONES ENGINEERING TEAM · VERIFIED 2026-07

Not every pin on a microcontroller can do every job. Some can only read, some are sampled at reset as strapping pins, some are tied to USB or to the chip's flash, and some reach only one of the ADC blocks. Reading the pinout before you wire a board saves you from a design that cannot work, or a re-spin to fix it. The rule is simple: match each function to a pin that can actually do it.

THE PINOUT IS THE MAP

The datasheet's pinout, and a board's printed pin diagram, list every pin with what it is allowed to be: a plain GPIO, an ADC channel, a strapping pin, a USB line, a power or ground pin, and so on. Before assigning a signal to a pin, check the map that the pin actually supports that job. It is two minutes that saves a bad board.

- [Espressif. ESP32-S3 Series Datasheet \(pin definitions and per-pin functions\).](#) [espressif.com](#)

PINS WITH RESTRICTIONS

Several kinds need care. On many microcontrollers some pins are input-only: they can read a signal but never drive an output. The original ESP32's [GPIO34](#) to [GPIO39](#) are the classic example; the ESP32-S3 happens to have none, so the caution moves to its other reserved pins. Strapping pins must be free at reset, or the board will not boot (an earlier lesson in this cluster). The native USB pins are spoken for if you use USB. On a module with built-in flash or PSRAM, the pins wired to that memory are reserved and off-limits. And only some pins reach the ADC. Everything else is a general-purpose pin you can assign freely.

A WHICH-PIN-FOR-WHICH-JOB CHECKLIST

Put an output on a full GPIO, never on an input-only pin on a chip that has them. Keep hard pulls and always-driven signals off the strapping pins. Land an analog input on an ADC-capable pin, and prefer the [ADC1](#) pins so a running Wi-Fi radio does not clash with your reading. Leave the USB pins and the on-module flash pins alone. Then confirm each choice against the pinout one more time before you route the board.

- [Espressif. ESP32-S3-DevKitC-1 User Guide \(a real board's pin Name/Function tables, flash-reserved pins\).](#)-
[docs.espressif.com](#)

DEEP DIVE · WHY ADC2 FIGHTS THE WI-FI RADIO

The ESP32 has two ADC blocks, and one of them, **ADC2**, is also used by the Wi-Fi driver. When Wi-Fi is running, an **ADC2** reading can fail or be made to wait, so a sensor sampled on an ADC2 pin returns garbage or stalls exactly when the board is connected. The fix is a pin choice: put any analog input that has to work while connected on an **ADC1** pin, and the clash never happens. It is a classic 'why is my sensor fine on the bench but broken over Wi-Fi' trap, and reading the pinout up front avoids it.

- Espressif. ESP-IDF Programming Guide: ADC Oneshot Mode (ADC2 is also used by Wi-Fi). docs.espressif.com

THE PINOUT AS A MAP: STRAPPING, ADC, USB, AND FLASH-RESERVED PINS EACH FLAGGED BEFORE YOU WIRE.

READ THE PINOUT BEFORE YOU ROUTE THE BOARD

Two minutes with the pin map before you assign signals saves a board re-spin. On your schematic, flag the strapping pins, the USB pair, the on-module flash pins, and the **ADC1** pins first, then wire your signals to what is left. It turns pin selection from a source of subtle bugs into a checklist.

CHECKPOINT**1. Why check the pinout before assigning a function to a pin?**

- To make the board layout look neater
- Because pins are all identical, so it never matters
- Because some pins are input-only, strapping, or reserved for USB or the ADC**

ANSWER · C

Pins differ: a strapping, input-only, or USB pin cannot take an arbitrary function.

2. An input-only pin can do what?

- Read a signal but never drive an output**
- Drive high-current loads directly
- Only carry power, never signals

ANSWER · A

Input-only pins read a level but have no output driver, so never assign an output to one.

3. Why prefer an ADC1 pin for an analog input on a Wi-Fi board?

- ADC1 samples much faster than ADC2
- The other ADC block is shared with Wi-Fi and can fail while connected**
- ADC1 always has more bits of resolution

ANSWER · B

ADC2 is shared with the Wi-Fi driver, so readings on it can fail while connected; ADC1 avoids the clash. The two blocks match in speed and resolution.

- Prerequisite: reading a datasheet
- Related: the ADC, reading an analog voltage
- Related: boot and strapping pins