

GPIO und sichere Pinwahl

Digitale Eingänge und Ausgänge am klassischen ESP32-DevKitC sicher einsetzen.

ACHTUNG

Diese Pin-Hilfe gilt fuer den klassischen ESP32/ESP32-DevKitC. ESP32-S2, S3, C3, C6 und andere Boards haben andere Pinbelegungen. Vor dem Stecken immer das konkrete Board-Pinout pruefen.

Pin-Gruppe (DevKitC)	Eignung	Wichtiger Hinweis
18, 19, 21, 22, 23	Gute digitale Startpins	Allgemeine Ein-/Ausgänge; Funktionen koennen per GPIO-Matrix zugeordnet werden.
25, 26, 27, 32, 33	Digital, teils ADC/DAC/Touch	Zusatzfunktionen sind chipabhaengig; Board-Dokumentation pruefen.
34, 35, 36, 39	Nur Eingang; ADC1	Kein digitaler Ausgang. Bei 34-39 keine internen Pull-up/-down-Widerstaende einplanen.
0, 2, 5, 12, 15	Boot-/Strapping-Pins	Pegel beim Start beeinflussen Boot-Konfiguration. Fuer Einsteiger besser vermeiden.
1 und 3	UART0 TX/RX	Werden meist fuer Upload und Serial Monitor genutzt; Konflikte vermeiden.
6 bis 11	Flash-Verbindung	Nicht verwenden: auf typischen Modulen intern mit SPI-Flash verbunden.
16 und 17	Modulabhaengig	Bei WROOM oft nutzbar; bei WROVER/PSRAM moeglicherweise reserviert.

INPUT

- `pinMode(pin, INPUT)`
- `digitalRead(pin)`
- Hochohmig ohne Pull
- Signal: LOW oder HIGH

INPUT_PULLUP

- Interner Pull-up ca. 45 kOhm
- Taster nach GND
- offen = HIGH
- gedrueckt = LOW

OUTPUT

- `pinMode(pin, OUTPUT)`
- `digitalWrite(pin, ...)`
- GPIO liefert Signal
- Lasten ueber Treiber

MERKEN

GND ist der gemeinsame Bezug. Ohne gemeinsame Masse sind Pegel unbestimmt. Ein GPIO ist kein Netzteil: Motor, Relais, Streifen oder Summer nicht direkt treiben.

GPIO und sichere Pinwahl

Digitale Eingänge und Ausgänge am klassischen ESP32-DevKitC sicher einsetzen.

TASTER -> LED

```
const int BUTTON_PIN = 21;
const int LED_PIN = 18;

void setup() {
  pinMode(BUTTON_PIN, INPUT_PULLUP);
  pinMode(LED_PIN, OUTPUT);
}

void loop() {
  bool gedrueckt =
    digitalRead(BUTTON_PIN) == LOW;
  digitalWrite(LED_PIN,
    gedrueckt ? HIGH : LOW);
}
```

SCHALTUNG

```
GPIO21 ---- TASTER ---- GND

GPIO18 -- 330 Ohm -- |>|-- GND
                      LED
```

```
INPUT_PULLUP:
offen      = HIGH
gedrueckt  = LOW
```

LED-Polung und Widerstand pruefen.

Thema	Datenblatt/Regel	Sichere Praxis
Logikpegel	GPIOs arbeiten im 3,3-V-Bereich; 5 V sind kein gueltiger Eingang.	5-V-Signale nur ueber passenden Pegelwandler/Spannungsteiler.
Ausgangsstrom	ESP32-Datenblatt nennt bei maximaler Treiberstaerke ca. 40 mA source bzw. 28 mA sink unter Testbedingungen.	Nicht als Dauer-Designziel nutzen. Unterrichts-LED typ. 5-10 mA; Widerstand berechnen.
Mehrere Pins	Verfuegbarer Strom sinkt, wenn mehrere Pins derselben Domaene gleichzeitig liefern.	Verbraucher extern versorgen; Transistor/MOSFET und Schutzbeschaltung nutzen.
Pull-Widerstand	Interne Pull-up/-down-Widerstaende liegen typisch bei etwa 45 kOhm.	Fuer lange Leitungen/stoerige Umgebung externen Widerstand vorsehen.

Fehlerbild	Wahrscheinlich	Pruefung
Startet nicht	Strapping-Pin wird beim Booten falsch gezogen	Schaltung abziehen; Pinwahl aendern
Eingang springt	Pin offen oder Masse fehlt	Pull-up/-down und gemeinsame GND
Ausgang reagiert nicht	Input-only oder falsche Pinnummer	Board-Pinout und pinMode pruefen

PRAXIS

Pinwahl dokumentieren: Nummer, Richtung, angeschlossenes Bauteil und Besonderheit direkt im Schaltplan und als const im Sketch festhalten.