

ADC, Potentiometer und Messwerte

Analoge Spannungen sicher einlesen, Rohwerte richtig deuten und Messungen stabil auswerten.

ACHTUNG

Analoge Eingänge sind nicht 5-V-tolerant. Ein Potentiometer fuer den Versuch liegt zwischen 3V3 und GND; der Schleifer geht an einen passenden ADC-Pin. Andere Boards und SoCs unterscheiden sich.

POTENTIOMETER

```
3V3 ----[ Potentiometer ]---- GND
      |
      +---- GPIO34 (ADC1)

aeusserer Pin -> 3V3
aeusserer Pin -> GND
mittlerer Pin  -> Signal

Vor dem Einschalten Schleifer-
spannung gegen GND messen.
```

ROHWERT LESEN

```
const int POT_PIN = 34;

void setup() {
  Serial.begin(115200);
  analogReadResolution(12);
}

void loop() {
  int raw = analogRead(POT_PIN);
  Serial.println(raw);
  delay(100);
}
```

12 Bit ergeben standardmaessig 0-4095. Der Rohwert ist noch keine exakte Spannung.

Begriff	Bedeutung	Praxis
ADC	Analog-Digital-Wandler	Wandelt eine Eingangsspannung in einen Zahlenwert.
12 Bit	$2^{12} = 4096$ Stufen	Standardbereich von <code>analogRead()</code> : 0 bis 4095.
Rohwert	Nicht kalibrierter Messwert	Nicht einfach linear mit 3,3 V gleichsetzen.
mV-Wert	<code>analogReadMilliVolts(pin)</code>	Nutzt Kalibrierung und liefert Millivolt; Genauigkeit bleibt hardwareabhaengig.
ADC1	Beim klassischen ESP32 GPIO32-39	Fuer Projekte mit Wi-Fi bevorzugen; GPIO34-39 nur Eingang.
ADC2	Mehrere andere ADC-Pins	Beim klassischen ESP32 Konflikt mit aktivem Wi-Fi moeglich.

MESSEN

- Rohwert anzeigen
- Minimum/Maximum beobachten
- Spannung kontrollieren

GLAETTEN

- Mehrfach messen
- Mittelwert bilden
- Kleine Schwankungen erwarten

KALIBRIEREN

- Referenzpunkte erfassen
- Nicht ideale Endwerte nutzen
- Einheit erst danach berechnen

MERKEN

Ein analoger Rohwert ist nur eine Zahl. Temperatur, Prozent oder Lux entstehen erst durch Sensor-Kennlinie und Kalibrierung.

ADC, Potentiometer und Messwerte

Analoge Spannungen sicher einlesen, Rohwerte richtig deuten und Messungen stabil auswerten.

ACHT MESSUNGEN MITTELN UND AUF GEMESSENEN BEREICH ABBILDEN

```
const int POT_PIN = 34;
const int RAW_MIN = 120; // am eigenen Aufbau messen
const int RAW_MAX = 3980; // am eigenen Aufbau messen

int readAverage() {
  uint32_t sum = 0;
  for (int i = 0; i < 8; i++) {
    sum += analogRead(POT_PIN);
  }
  return sum / 8;
}

void setup() {
  Serial.begin(115200);
}

void loop() {
  int raw = readAverage();
  int percent = map(raw, RAW_MIN, RAW_MAX, 0, 100);
  percent = constrain(percent, 0, 100);

  Serial.printf("raw=%d percent=%d%%\n", raw, percent);
  delay(100);
}
```

RAW_MIN/RAW_MAX am echten Aufbau bestimmen. Fuer Spannungen analogReadMilliVolts() und die SoC-spezifische Daempfung beachten.

Fehlerbild	Wahrscheinlich	Naechster Test
Immer 0	Signal/GND fehlt oder falscher Pin	Spannung am ADC-Pin gegen GND messen
Immer Maximum	Signal nahe Messbereichsende/falsch verdrahtet	Schleifer und 3V3/GND pruefen
Springt stark	Rauschen, lange Leitung, instabile Quelle	Kurze Leitung, Mittelwert, Kondensator nach Sensor-Datenblatt
Mit Wi-Fi unbrauchbar	ADC2-Konflikt beim klassischen ESP32	Auf ADC1-Pin wechseln, z.B. GPIO34
Prozent falsch	0/4095 sind nicht reale Endwerte	RAW_MIN und RAW_MAX messen und map() anpassen

SERIAL MONITOR

- Beschriftete Einzelwerte
- Fehlertext
- Grenzen pruefen

SERIAL PLOTTER

- Verlauf sichtbar
- Rauschen erkennen
- Eine Zahl pro Zeile

MULTIMETER

- Eingangsspannung
- GND-Bezug
- 5 V sicher ausschliessen

PRAXIS

Erst Rohdaten dokumentieren, dann skalieren. ADC-Messbereich, Daempfung und geeignete Pins immer fuer den verwendeten ESP32-SoC pruefen.