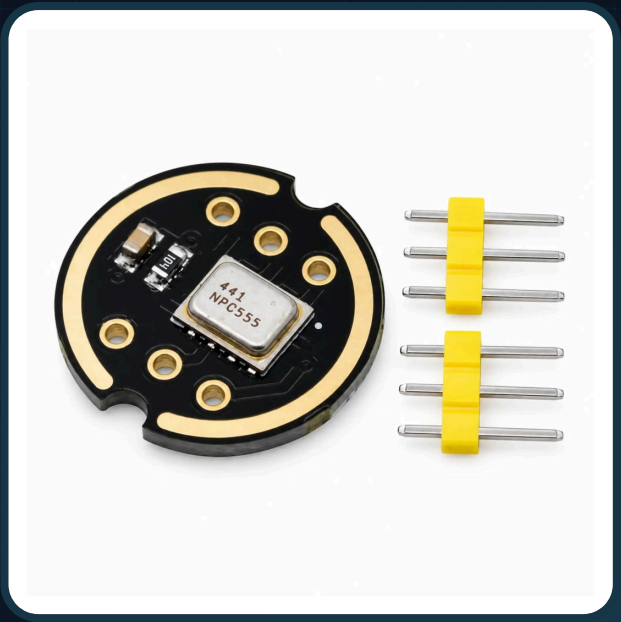
DIGITALES MIKROFON · I2S · ESP32

INMP441 I2S-Mikrofon

Ein MEMS-Mikrofon, das den Ton schon auf der Platine digitalisiert – sauber und ohne Brummen. Erste Pegelmessung in rund 20 Minuten mit dem ESP32 DevKit.



AUFLÖSUNG	24 Bit
ABTASTRATE	8–48 kHz
VERSORGUNG	1,8–3,3 V DC
SCHNITTSTELLE	I2S

01
Verkabeln
6 Leitungen zum ESP32

02
Einrichten
IDE und Board-Paket

03
Hochladen
Beispiel-Sketch

04
Messen
Schallpegel

01 Auf einen Blick

Das INMP441 gibt keinen analogen Pegel aus, sondern einen digitalen Datenstrom über I2S. Ein Arduino UNO hat diese Schnittstelle nicht – der ESP32 schon. Der Sketch misst daraus den Schallpegel und zeigt ihn als Balken.

Technische Daten

Wandler	**MEMS, omnidirektional**
Auflösung	**24 Bit**
Abtastrate	**ca. 8–48 kHz**
Frequenzgang	**ca. 60 Hz – 15 kHz**
Rauschabstand	**ca. 61 dB**
Versorgung	**1,8–3,3 V DC**
Anschluss	**6 Pins, Stiftleiste liegt bei**

Lieferumfang

1 × **INMP441 MEMS-Mikrofon-Modul (I2S) inkl. loser Stiftleiste**

Nur 3,3 V und I2S
An 5 V geht das Mikrofon kaputt. Am Arduino UNO funktioniert es nicht – er hat kein I2S.

Schallöffnung
Das Loch für den Schall sitzt auf der Unterseite der Platine. Nicht zukleben.

Pinbelegung

PIN	FUNKTION	SPANNUNG	HINWEIS
VDD	Versorgung	1,8–3,3 V DC	An 3V3 – nie an 5 V.
GND	Masse	0 V	Gemeinsame Masse.
SCK	I2S Bit Clock	3,3 V	ESP32 GPIO26.
WS	I2S Word Select	3,3 V	ESP32 GPIO25.
SD	I2S Daten	3,3 V	ESP32 GPIO33.
L/R	Kanalwahl	GND oder VDD	An GND = linker Kanal (wie im Sketch).

Zwei Reihen à drei Pins: L/R · WS · SCK und SD · VDD · GND. Die Beschriftung steht auf der Rückseite.

Was du zusätzlich brauchst

ESP32 DevKit USB-C
WROOM-32, CH340C – 3,3-V-Logik
MKR-ESP32-005

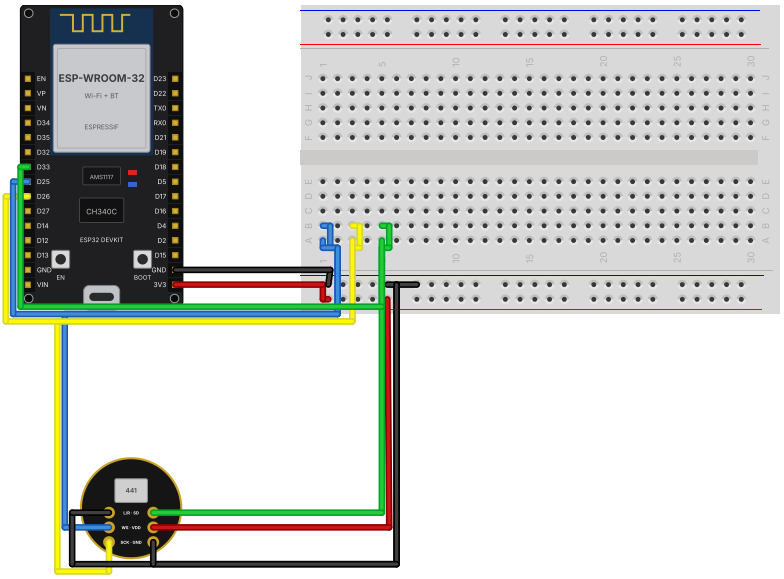
Breadboard + Jumper-Set
400 Kontakte, 40 Dupont-Kabel
MKR-SET-BB400-JUMPER

Lötkolben
Stiftleiste einlöten (z. B. TOOLCRAFT LS-30E)
MKR-LS30E-70

02 Verkabeln

Sechs Leitungen: L/R kommt mit an GND und legt den Kanal fest.

STECKPLATINEN-ANSICHT



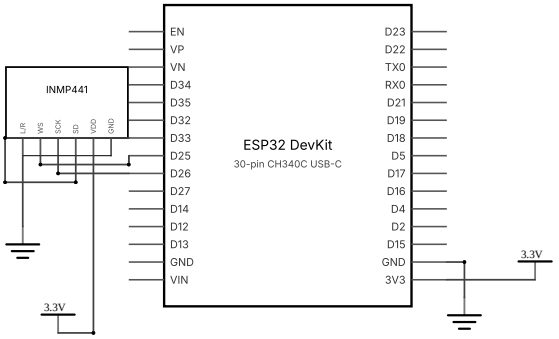
Verbindungen

MODUL		ESP32 DEVKIT	LEITUNG
VDD	→	3V3	Versorgung
GND	→	GND	Masse
L/R	→	GND	linker Kanal
SCK	→	D26	Bit Clock
WS	→	D25	Word Select
SD	→	D33	Daten

Stiftleisten einlöten
Zwei 3er-Leisten von unten einsetzen, sodass die Schallöffnung frei bleibt.

L/R nicht offen lassen
Ohne festen Pegel an L/R liefert das Mikrofon mal links, mal rechts – oder gar nichts.

SCHALTPLAN



Verdrahtung erstellt mit Fritzing

03 Einrichten & Hochladen

Schritte 1 bis 3 machst du nur ein einziges Mal – danach ist dein ESP32 für alle weiteren Projekte startklar.

1 Arduino IDE installieren

Lade die aktuelle Arduino IDE 2 von `arduino.cc/en/software` herunter und installiere sie.

2 ESP32-Board-Paket installieren

Im Boardverwalter (Board-Symbol links) nach **esp32** suchen und **esp32 von Espressif Systems** installieren. Den CH340C-Treiber bringen Windows 10/11 und macOS meist mit. Ausführlich mit Bildern: unser Guide zum ESP32 DevKit.

3 Board und Port wählen

ESP32 per USB-C anschließen. In der Board-Auswahl **ESP32 Dev Module** wählen und den neuen COM-Port (Windows) bzw. `/dev/ttyUSB0` (Linux) / `/dev/cu.wchusbserial…` (macOS).

4 Keine Bibliothek nötig

Die I2S-Bibliothek *ESP_I2S* steckt bereits im ESP32-Board-Paket – weiter mit Schritt 5.

5 Sketch hochladen

Neuen Sketch anlegen, den Code von Seite 5 einfügen und mit dem Pfeil-Symbol hochladen (`Strg` + `U`). Unten erscheint „Hochladen abgeschlossen“.

6 Seriellen Monitor öffnen

`Strg` + `Umschalt` + `M`, unten rechts **115200 Baud** einstellen. Zehnmal pro Sekunde erscheint der Pegel in dBFS mit Balken. Klatsch in die Hände!

BEISPIELAUSGABE · SERIELLER MONITOR · 115200 BAUD

```
Pegel: -63.8 dBFS ########
Pegel: -61.2 dBFS #########
Pegel: -21.5 dBFS ##############################
Pegel: -58.9 dBFS ##########
```

i Schnelltest

Leise: um −60 dBFS. Sprechen: −40 bis −30. Klatschen: über −25. Steht der Wert fest bei −180 oder ändert sich nicht, kommen keine Daten an.

04 Der erste Sketch

Liest 256 Abtastwerte, berechnet daraus den Effektivwert und zeigt den Pegel in dBFS als Zahl und Balken.

MAKEROO_INMP441_Start.ino

```
// MAKEROO · INMP441 I2S-Mikrofon · Lautstaerke messen (ESP32)
#include <ESP_I2S.h>

const int SCK_PIN = 26, WS_PIN = 25, SD_PIN = 33;   // L/R an GND = linker Kanal
I2SClass i2s;
int32_t puffer[256];

void setup() {
  Serial.begin(115200);
  i2s.setPins(SCK_PIN, WS_PIN, -1, SD_PIN);         // BCLK, WS, DOUT, DIN
  if (!i2s.begin(I2S_MODE_STD, 16000, I2S_DATA_BIT_WIDTH_32BIT,
                 I2S_SLOT_MODE_MONO, I2S_STD_SLOT_LEFT)) {
    Serial.println("I2S-Start fehlgeschlagen!");
    while (true) delay(10);
  }
}

void loop() {
  size_t n = i2s.readBytes((char*)puffer, sizeof(puffer)) / 4;
  if (n == 0) return;
  double summe = 0;
  for (size_t i = 0; i < n; i++) {
    float s = (puffer[i] >> 8) / 8388608.0;         // 24 Bit -> -1 ... +1
    summe += s * s;
  }
  float dbfs = 20 * log10(sqrt(summe / n) + 1e-9); // 0 dBFS = Vollaussteuerung
  int balken = constrain(map((int)dbfs, -80, 0, 0, 40), 0, 40);
  Serial.printf("Pegel: %6.1f dBFS  ", dbfs);
  for (int i = 0; i < balken; i++) Serial.print('#');
  Serial.println();
  delay(100);
}
```

Zeile 4
Pins
SCK, WS und SD sind am ESP32 frei wählbar.

Zeile 10
Pins zuweisen
Kein Ausgang (−1) – wir lesen nur.

Zeile 11–12
I2S starten
16 kHz, 32-Bit-Wörter, nur der linke Kanal.

Zeile 19
Einlesen
Füllt den Puffer mit bis zu 256 Werten.

Zeile 23
24 Bit
Die oberen 24 Bit enthalten den Messwert – auf −1 bis +1 normiert.

Zeile 26
Pegel in dB
0 dBFS ist Vollaussteuerung, Stille liegt um −60 bis −70.

Geprüft: kompiliert für ESP32 Dev Module (Board-Paket esp32 3.3.12) (290.3 kB Flash = 22 %, 23 kB RAM).

05 Wenn es nicht klappt

SYMPTOM	LÖSUNG
Pegel bleibt bei −180	Keine Daten: SD an GPIO33? L/R an GND? VDD an 3V3?
Werte springen wild	SCK und WS vertauscht oder Kabel zu lang – unter 15 cm bleiben.
„I2S-Start fehlgeschlagen!“	ESP32-Board-Paket ab Version 3.0 nötig (*ESP_I2S*). Unter `Werkzeuge → Board` die Version prüfen.
Nur Rauschen um −50	Mikrofon zu nah am ESP32 oder an WLAN-Antenne – ein paar Zentimeter Kabel helfen.
Upload bricht ab	Beim „Connecting...“ die BOOT-Taste des ESP32 gedrückt halten, bis der Upload startet.

06 Wie geht's weiter?

Mit ESP32 & ESPHome

Gleiche Verdrahtung wie hier – in ESPHome als Mikrofon für Sprachassistenten:

```
i2s_audio:
  i2s_lrclk_pin: GPIO25
  i2s_bclk_pin: GPIO26
microphone:
  - platform: i2s_audio
    id: mic
    i2s_din_pin: GPIO33
    adc_type: external
    channel: left
```

Projektideen

Lärmampel
Pegel als Farbe auf dem LED-Ring

Klatschschalter
Zweimal klatschen, Relais schaltet

Sprachassistent
Mit ESPHome und Home Assistant

Babyphone
Benachrichtigung bei Geräusch

Passt dazu

WS2812 LED-Ring
12 RGB-LEDs als Anzeige
`MKR-WS2812-RING12-67`

Relais KY-019
Verbraucher schalten
`MKR-KY019-034`

ESP32 Gehäuse aus Acryl
Schützt das Board
`MKR-ESP32-HOUSING-045`

0,96" OLED-Display
Werte direkt anzeigen
`MKR-0.96-OLED-015`

Sicherheitshinweise. Elektronikmodul für Ausbildungs-, Entwicklungs- und Experimentierzwecke. Nur mit Kleinspannung betreiben, niemals an Netzspannung. Nicht geeignet für Kinder unter 14 Jahren; Nutzung nur unter Aufsicht. Elektroaltgeräte gehören nicht in den Hausmüll – Rückgabe kostenlos bei kommunalen Sammelstellen.

Verantwortlicher Wirtschaftsakteur (GPSR): Alexander Prez · makeroo · Hinterm Bahnhof 4 · 90513 Zirndorf · support@makeroo.de
Fragen? Schreib uns mit einem Foto deiner Verkabelung.
Arduino ist eine Marke der Arduino SA und dient nur der Kompatibilitätsangabe.