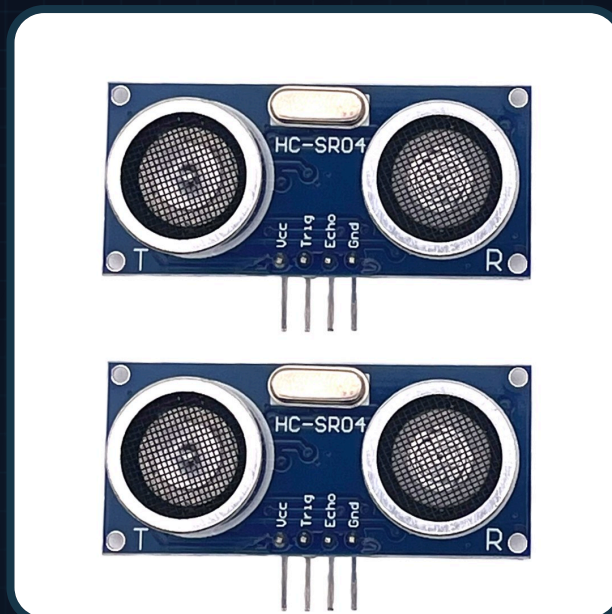


ULTRASCHALL-ENTFERNUNG

HC-SR04 Ultraschall

Zwei Ultraschallsensoren für Entfernungen von 2 cm bis 4 m. Erste Messung in rund 15 Minuten – mit UNO R3, Steckbrett und vier Kabeln.



MESSBEREICH

2 cm ... 4 m

AUFLÖSUNG

ca. 3 mm

VERSORGUNG

5 V DC

SCHNITTSTELLE

Trig + Echo

01

Verkabeln

4 Leitungen zum UNO

02

Einrichten

IDE und Treiber

03

Hochladen

Beispiel-Sketch

04

Messen

Serieller Monitor

01 Auf einen Blick

Der HC-SR04 misst Entfernungen per Ultraschall: Er sendet einen 40-kHz-Impuls aus und misst die Laufzeit des Echos. Ein 10-µs-Impuls an Trig startet die Messung, die Länge des Echo-Impulses verrät die Entfernung.

Technische Daten

Messbereich	ca. 2 cm bis 4 m
Auflösung	ca. 3 mm
Messwinkel	ca. 15°
Frequenz	40 kHz Ultraschall
Versorgungsspannung	5 V DC
Ruhestrom	< 2 mA
Signale	Trig-Eingang, Echo-Ausgang (5 V)

Lieferumfang

2 × HC-SR04 Ultraschallsensor



Echo liefert 5 V

Am UNO kein Problem. An ESP32 oder Raspberry Pi gehört ein Spannungsteiler oder Pegelwandler an den Echo-Pin.



Gut zu wissen

Weiche Stoffe schlucken den Schall, schräge Flächen lenken ihn weg – beides führt zu Fehlmessungen.

Pinbelegung

PIN	FUNKTION	SPANNUNG	HINWEIS
● VCC	Versorgung POWER	5 V DC	Braucht 5 V – auch am ESP32.
● Trig	Trigger-Eingang DIGITAL	5 V Logik	10-µs-HIGH-Impuls startet eine Messung.
● Echo	Echo-Ausgang DIGITAL	5 V Ausgang	HIGH-Dauer = Laufzeit des Schalls hin und zurück.
● GND	Masse GND	0 V	Gemeinsame Masse.

Die Pins sind auf der Vorderseite beschriftet: Vcc, Trig, Echo, Gnd.

Was du zusätzlich brauchst

UNO R3 mit USB-C

ATmega328P, CH340G-USB-Wandler
[MKR-UNO-R3-013](#)

Breadboard + Jumper-Set

400 Kontakte, 40 Dupont-Kabel
[MKR-SET-BB400-JUMPER](#)

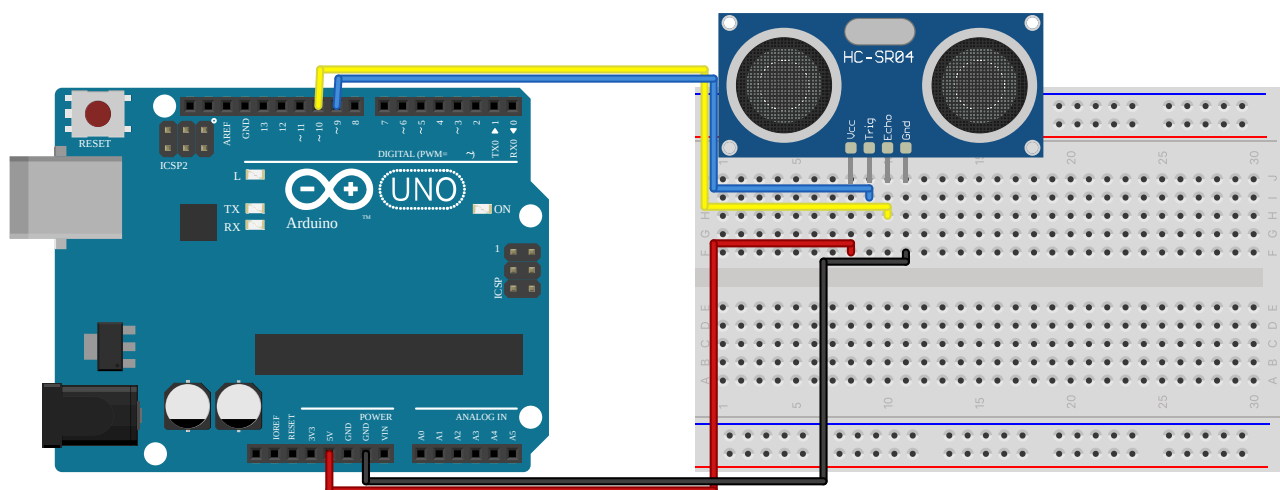
USB-C-Datenkabel

Kein reines Ladekabel – es muss Daten übertragen

02 Verkabeln

Vier Leitungen genügen. Zieh vorher das USB-Kabel ab – verkabelt wird immer stromlos.

STECKPLATINEN-ANSICHT



Verbindungen

SENSOR		UNO R3	LEITUNG
VCC	→	5V	Versorgung
Trig	→	D9	Messung starten
Echo	→	D10	Echo-Laufzeit
GND	→	GND	Masse

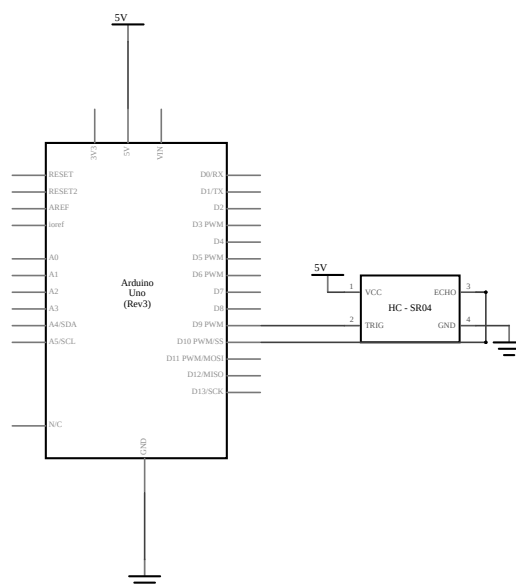
i Zwei Sensoren?

Der zweite HC-SR04 bekommt eigene Trig/Echo-Pins (z. B. D11/D12) und teilt sich 5V und GND mit dem ersten.

! Erst prüfen, dann einstecken

Vertauschte VCC und GND können den Sensor beschädigen. Erst alle Leitungen prüfen, dann USB anstecken.

SCHALTPLAN



Verdrahtung erstellt mit Fritzing

03 Einrichten & Hochladen

Schritte 1 bis 3 machst du nur ein einziges Mal – danach ist dein UNO R3 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 Treiber für den CH340G prüfen

Der UNO R3 nutzt den CH340G als USB-Seriell-Wandler. Aktuelle Systeme erkennen ihn meist automatisch. Taucht in Schritt 3 kein Port auf, installiere den Treiber „CH341SER“ des Chipherstellers WCH (wch-ic.com).

3 Board und Port wählen

UNO per USB-C anschließen. Dann **Werkzeuge** → **Board** → **Arduino AVR Boards** → **Arduino Uno** wählen und unter **Werkzeuge** → **Port** den neu aufgetauchten COM-Port (Windows) bzw. `/dev/ttyUSB0` (Linux) / `/dev/cu.wchusbserial...` (macOS).

4 Keine Bibliothek nötig

Dieser Sketch kommt ohne Zusatzbibliothek aus – du kannst direkt mit Schritt 5 weitermachen.

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 **9600 Baud** einstellen. Zweimal pro Sekunde erscheint die gemessene Entfernung.

BEISPIELAUSGABE · SERIELLER MONITOR · 9600 BAUD

MAKER00 HC-SR04-Test startet ...

Entfernung: 23.4 cm

Entfernung: 23.5 cm

Entfernung: 11.8 cm

i Schnelltest

Halte die Hand vor den Sensor und bewege sie langsam vor und zurück – der Wert muss mitgehen.

04 Der erste Sketch

Löst zweimal pro Sekunde eine Messung aus und rechnet die Echo-Laufzeit in Zentimeter um.

```
MAKER00_HCSR04_Start.ino

1 // MAKER00 Getting Started - HC-SR04 Ultraschallsensor am UNO R3
2 // Misst zweimal pro Sekunde die Entfernung in Zentimetern.
3
4 const int TRIG_PIN = 9; // Trig an D9
5 const int ECHO_PIN = 10; // Echo an D10
6
7 void setup() {
8   Serial.begin(9600);
9   pinMode(TRIG_PIN, OUTPUT);
10  pinMode(ECHO_PIN, INPUT);
11  Serial.println(F("MAKER00 HC-SR04-Test startet ..."));
12 }
13
14 void loop() {
15   // 10-µs-Impuls an Trig startet eine Messung
16   digitalWrite(TRIG_PIN, LOW);
17   delayMicroseconds(2);
18   digitalWrite(TRIG_PIN, HIGH);
19   delayMicroseconds(10);
20   digitalWrite(TRIG_PIN, LOW);
21
22   // Dauer des Echo-Impulses in µs (max. 30 ms ≈ 5 m)
23   unsigned long daver = pulseIn(ECHO_PIN, HIGH, 30000UL);
24
25   if (daver == 0) {
26     Serial.println(F("Kein Echo - außer Reichweite?"));
27   } else {
28     float cm = daver * 0.0343 / 2.0; // Schall: 343 m/s, hin und zurück
29     Serial.print(F("Entfernung: "));
30     Serial.print(cm, 1);
31     Serial.println(F(" cm"));
32   }
33   delay(500);
34 }
```

Zeile 4-5

Pins

Hier stehen Trig- und Echo-Pin.
Andere Pins? Nur diese Zahlen
ändern.

Zeile 23

Laufzeit messen

`pulseIn()` misst, wie lange Echo auf
HIGH steht – max. 30 ms.

Zeile 28

Umrechnung

Schall legt 0,0343 cm pro µs zurück;
geteilt durch 2, weil er hin und
zurück läuft.

Geprüft: kompiliert für Arduino Uno (3.8 kB Flash = 11 %, 200 Byte RAM).

05

Wenn es nicht klappt

SYMPTOM	LÖSUNG
Immer „Kein Echo“	Trig an D9 und Echo an D10? Nicht vertauscht? Der Sensor braucht 5 V, nicht 3,3 V.
Werte springen stark	Glatte, gerade Fläche anpeilen. Weiche oder schräge Flächen streuen den Schall.
Unter 2 cm falsche Werte	Zu nah – der Sensor misst erst ab ca. 2 cm zuverlässig.
Kein Port in der IDE	Anderes USB-C-Kabel probieren – viele Kabel laden nur. Danach CH340-Treiber installieren (Schritt 2).
Nur wirre Zeichen im Monitor	Baudrate im seriellen Monitor auf 9600 stellen.

06

Wie geht's weiter?

Mit ESP32 & ESPHome

VCC an **5V** (VIN), Trig an z. B. **GPIO5**. Echo an **GPIO18** – aber nur über einen Spannungsteiler (z. B. 1 kΩ/2 kΩ). In ESPHome:

```
sensor:
  - platform: ultrasonic
    trigger_pin: GPIO5
    echo_pin: GPIO18
    name: "Abstand"
    update_interval: 1s
```

Projektideen

Einparkhilfe

Abstand per LED-Ring in Ampelfarben anzeigen

Füllstand

Wasserstand in Zisterne oder Regentonne messen

Roboter

Hindernisse erkennen und ausweichen

Besucherzähler

Durchgänge an einer Tür zählen

Passt dazu

0,96" OLED-Display

Messwerte direkt anzeigen

[MKR-0.96-OLED-015](#)

ESP32 DevKit USB-C

WLAN, ESPHome, Home

Assistant

[MKR-ESP32-005](#)

WS2812 LED-Ring

12 RGB-LEDs als Anzeige

[MKR-WS2812-RING12-67](#)

2x SG90 Servo 360°

Bewegung per PWM

[MKR-SG90-360G-003](#)

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.