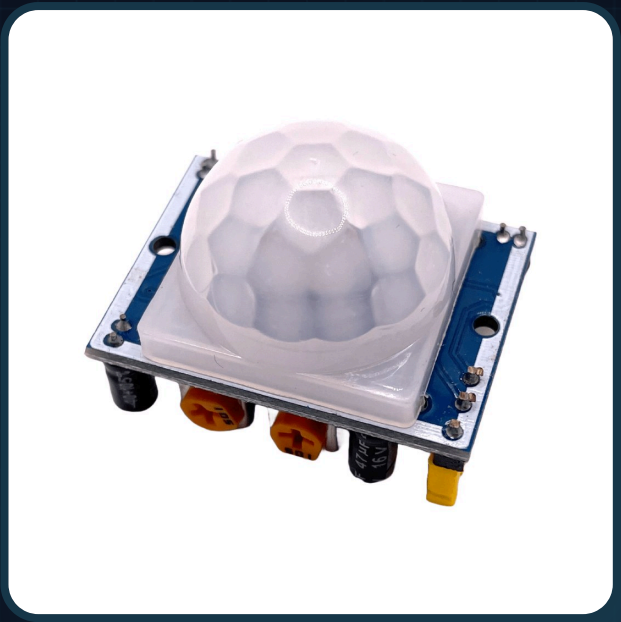
BEWEGUNGSMELDER (PIR)

HC-SR501 PIR

Passiv-Infrarot-Bewegungsmelder mit einstellbarer Reichweite. Erste Auslösung in rund 15 Minuten – mit UNO R3 und drei Kabeln.



REICHWEITE 3 … 7 m	ERFASSUNG ca. 110°
VERSORGUNG 4,5 … 20 V DC	AUSGANG 3,3 V digital

01
Verkabeln
3 Leitungen zum UNO

02
Einrichten
IDE und Treiber

03
Hochladen
Beispiel-Sketch

04
Messen
Serieller Monitor

01 Auf einen Blick

Der HC-SR501 registriert die Wärmestrahlung eines Körpers, der sich durch sein Blickfeld bewegt. Erkennt er Bewegung, legt er seinen Ausgang auf HIGH – mehr Auswertung braucht es nicht.

Technische Daten

Versorgungsspannung	**4,5 bis 20 V DC**
Ausgang	**3,3 V HIGH bei Bewegung**
Erfassung	**ca. 110°, 3 bis 7 m**
Nachlaufzeit	**ca. 3 s bis 5 min (Poti)**
Trigger-Modus	**wiederholend (H) / einmalig (L)**
Ruhestrom	**ca. 65 µA**
Aufwärmzeit	**30 bis 60 s**

Lieferumfang

1 × **HC-SR501 PIR-Bewegungsmelder (blaue Platine)**

! Erst aufwärmen lassen

Nach dem Einschalten braucht der Sensor rund eine Minute. In dieser Zeit meldet er gern Fehlalarme.

i Bewegung, nicht Anwesenheit

Wer still sitzt, verschwindet für den Sensor. Er reagiert nur auf Wärme, die sich bewegt.

Pinbelegung

PIN	FUNKTION	SPANNUNG	HINWEIS
VCC	Versorgung POWER	4,5–20 V DC	Am UNO an 5V.
OUT	Digitalausgang DIGITAL	3,3 V HIGH	HIGH bei Bewegung – direkt an UNO oder ESP32.
GND	Masse GND	0 V	Gemeinsame Masse.

Die Beschriftung VCC · OUT · GND steht auf der Unterseite neben den Pins (unter der Linse). OUT liegt immer in der Mitte – prüfe die Außenpins vor dem Einschalten.

Was du zusätzlich brauchst

UNO R3 mit USB-C
ATmega328P, CH340G-USB-Wandler
MKR-UNO-R3-013

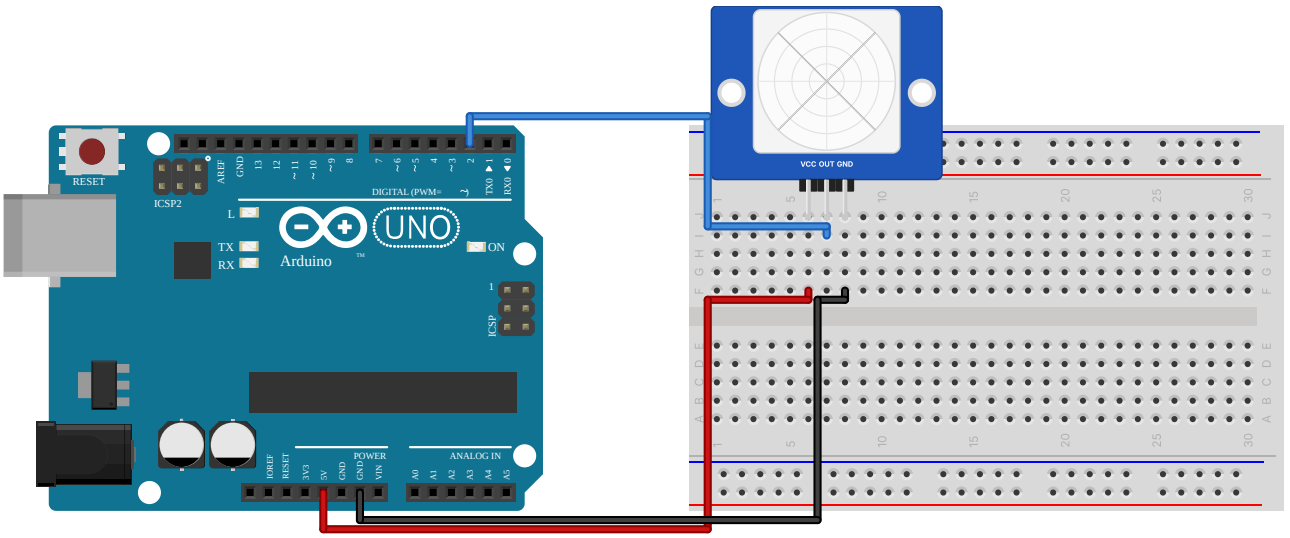
Breadboard + Jumper-Set
400 Kontakte, 40 Dupont-Kabel
MKR-SET-BB400-JUMPER

USB-C-Datenkabel
Kein reines Ladekabel – es muss Daten übertragen

02 Verkabeln

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

STECKPLATINEN-ANSICHT



Verbindungen

SENSOR		UNO R3	LEITUNG
● VCC	→	5V	Versorgung
● OUT	→	D2	Bewegung
● GND	→	GND	Masse

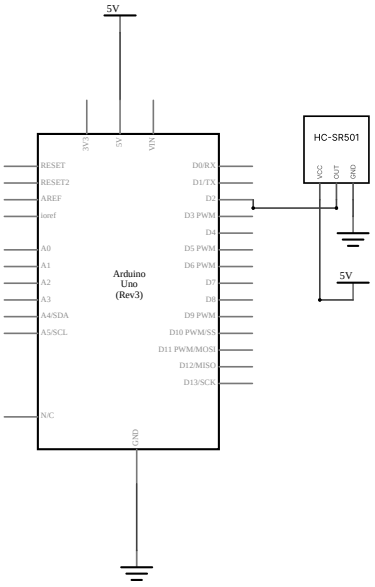
Einstellen

Linkes und rechtes Poti stellen Empfindlichkeit und Nachlaufzeit ein. Für den Test: Nachlaufzeit-Poti ganz gegen den Uhrzeigersinn (kürzeste Zeit).

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. Bei jeder Änderung erscheint eine Zeile – zusätzlich leuchtet die LED „L“ auf dem UNO.

BEISPIELAUSGABE · SERIELLER MONITOR · 9600 BAUD

```
MAKEROO PIR-Test - bitte ca. 60 s aufwärmen lassen ...
Bewegung erkannt! (74 s nach Start)
Ruhe (79 s nach Start)
Bewegung erkannt! (112 s nach Start)
```

i Schnelltest

Nach der Aufwärmzeit einmal mit der Hand vor der Linse winken – die LED „L“ geht an und nach der eingestellten Nachlaufzeit wieder aus.

04 Der erste Sketch

Liest den Ausgang des Melders, spiegelt ihn auf die Board-LED und meldet jede Änderung.

MAKEROO_PIR_Start.ino

```
1  // MAKEROO Getting Started – HC-SR501 Bewegungsmelder am UNO R3
2  // Meldet Bewegung im seriellen Monitor und schaltet die Board-LED.
3
4  const int PIR_PIN = 2;  // OUT an D2
5  bool letzterZustand = LOW;
6
7  void setup() {
8    Serial.begin(9600);
9    pinMode(PIR_PIN, INPUT);
10   pinMode(LED_BUILTIN, OUTPUT);
11   Serial.println(F("MAKEROO PIR-Test – bitte ca. 60 s aufwärmen lassen ..."));
12 }
13
14 void loop() {
15   bool zustand = digitalRead(PIR_PIN);
16   digitalWrite(LED_BUILTIN, zustand);  // LED "L" zeigt den Ausgang
17
18   if (zustand ≠ letzterZustand) {     // nur Änderungen ausgeben
19     if (zustand == HIGH) {
20       Serial.print(F("Bewegung erkannt!  ("));
21     } else {
22       Serial.print(F("Ruhe  ("));
23     }
24     Serial.print(millis() / 1000);
25     Serial.println(F(" s nach Start)"));
26     letzterZustand = zustand;
27   }
28 }
```

Zeile 4
Pin
OUT hängt an D2. Anderer Pin? Nur diese Zahl ändern.

Zeile 16
Board-LED
Die LED „L“ zeigt den Ausgang direkt an – praktisch zum Einstellen der Potis.

Zeile 18
Nur Änderungen
So erscheint nicht jede Millisekunde eine Zeile, sondern nur beim Wechsel.

Geprüft: kompiliert für Arduino Uno (2.3 kB Flash = 7 %, 189 Byte RAM).

05 Wenn es nicht klappt

SYMPTOM	LÖSUNG
Löst ständig aus	Aufwärmzeit abwarten. Empfindlichkeit reduzieren, Sensor nicht auf Heizung oder Fenster richten.
Bleibt lange HIGH	Nachlaufzeit am Poti kürzer stellen. Jumper auf L = einmalige Auslösung.
Reagiert gar nicht	OUT an D2? VCC an 5V (nicht 3,3 V)? Mindestens 4,5 V nötig.
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), OUT direkt an z. B. **GPIO27** – der Ausgang hat schon 3,3 V. In ESPHome:

```
binary_sensor:
  - platform: gpio
    pin: GPIO27
    name: "Bewegung Flur"
    device_class: motion
```

Projektideen

Treppenlicht
LED-Streifen bei Bewegung einschalten

Alarmanlage
Summer und Push-Nachricht auslösen

Kamera-Trigger
ESP32-CAM bei Bewegung fotografieren lassen

Anwesenheit
Raumnutzung in Home Assistant auswerten

Passt dazu

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

ESP32 DevKit USB-C
WLAN, ESPHome, Home Assistant
`MKR-ESP32-005`

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

0,96" OLED-Display
Messwerte 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.