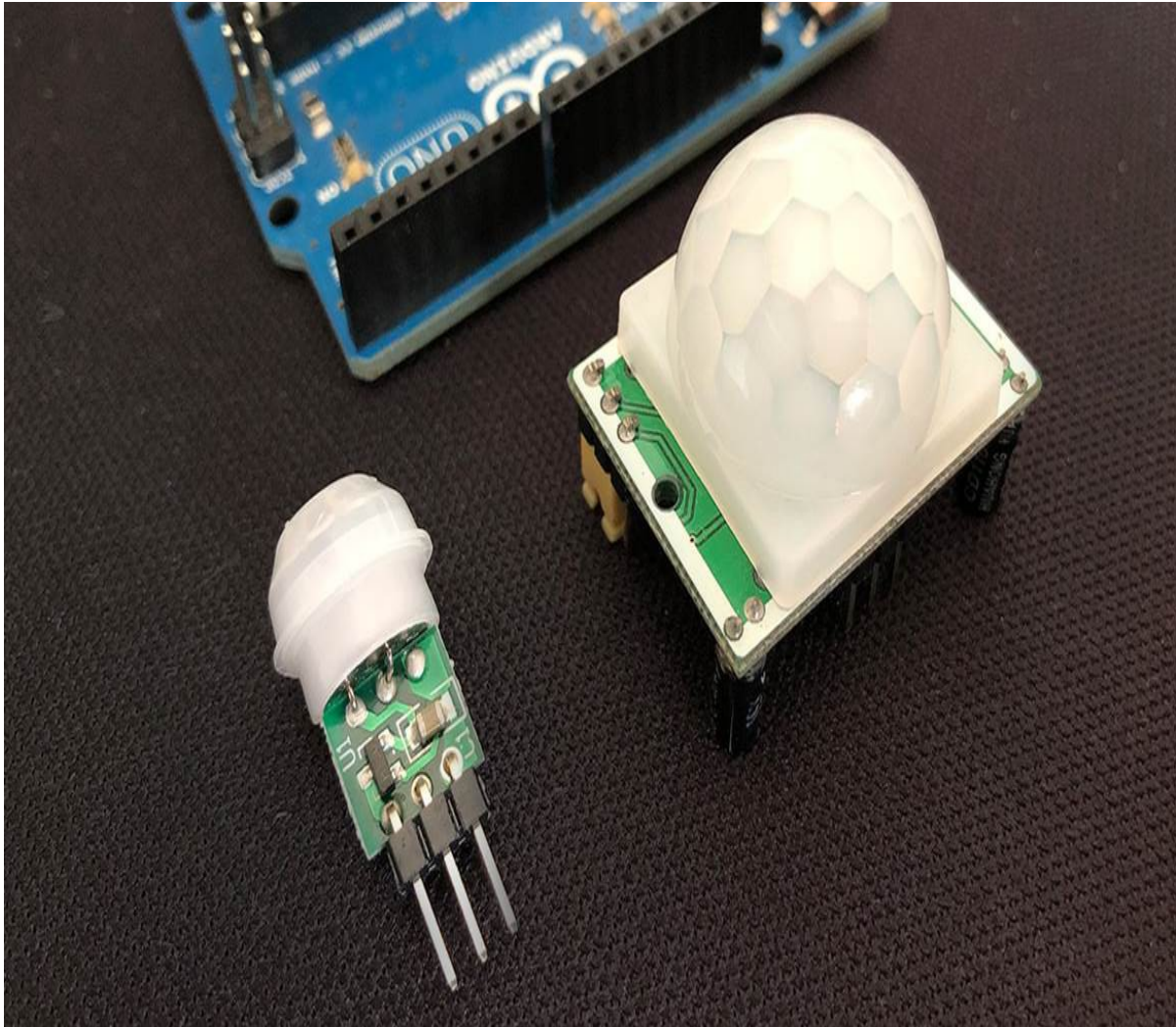


AM-312 oder HC-SR312 PIR Mini Motion Sensor



Arduino Bewegungsmelder – So klappt es

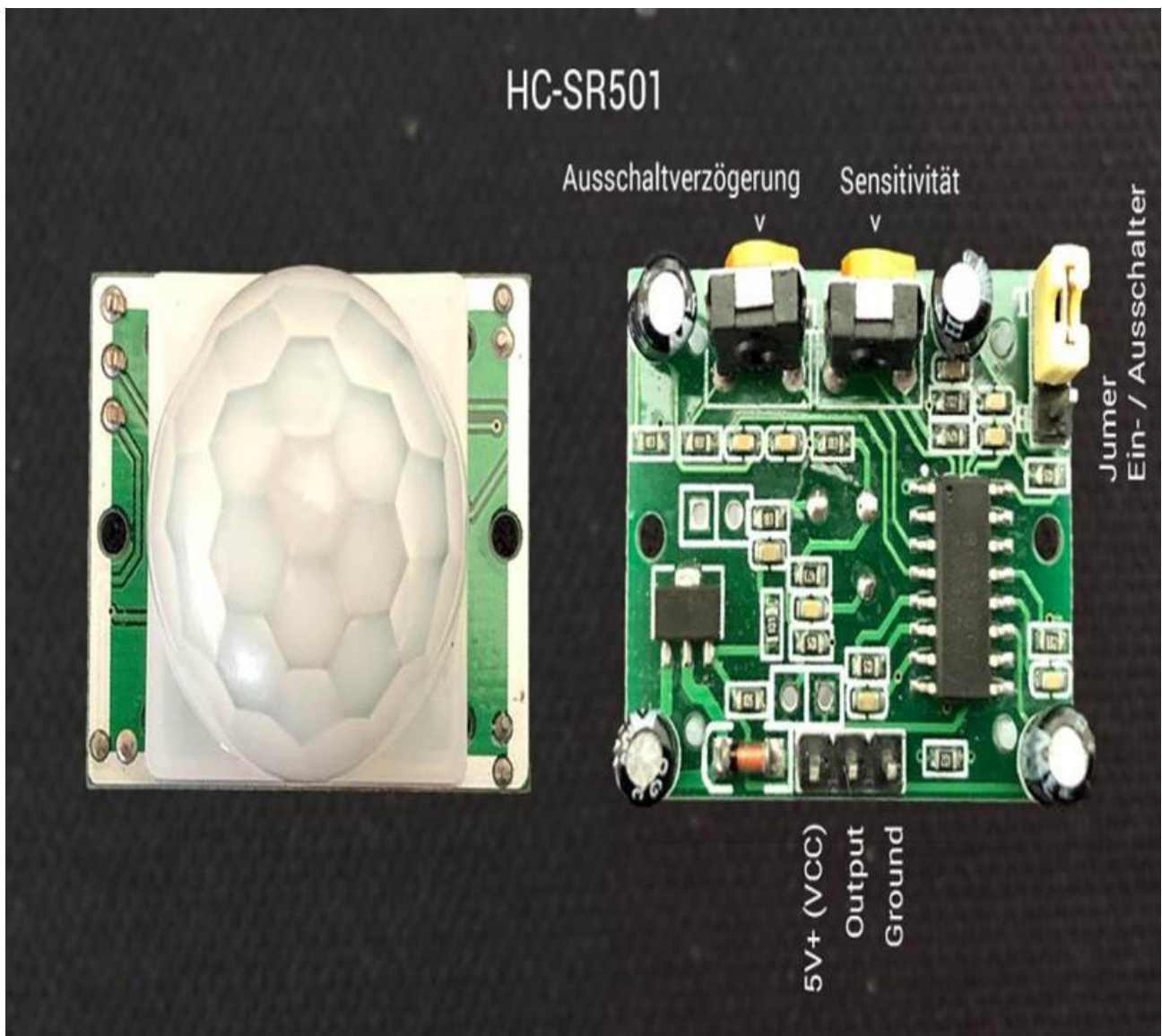
Wie kann man mit Arduino Bewegungsmelder auslesen? Welche Bewegungsmelder sind geeignet? Wie schließt man sie an? Antworten darauf in diesem Artikel.

Bei Bewegungsmelder handelt es sich um [PIR-Sensoren](#). PIR steht dabei für *Passiver Infrarotsensor* oder noch cooler *Pyroelektrischer Infrarotsensor*. Diese PIR-Sensoren reagieren auf Veränderungen der Temperatur. Geht ein Mensch mit seinem warmen Körper durch den Sichtbereich des PIR-Sensors, erzeugt er/sie/es eine Temperaturänderung und löst dadurch den Sensor aus.

Der Einfachheit halber gibt es für Arduino bereits fertige PIR-Module, bei denen die periphere Elektronik bereits integriert ist. Wir können also direkt mit diesen Breakout-Boards arbeiten. Ich empfehle die Verwendung folgender Sensoren:

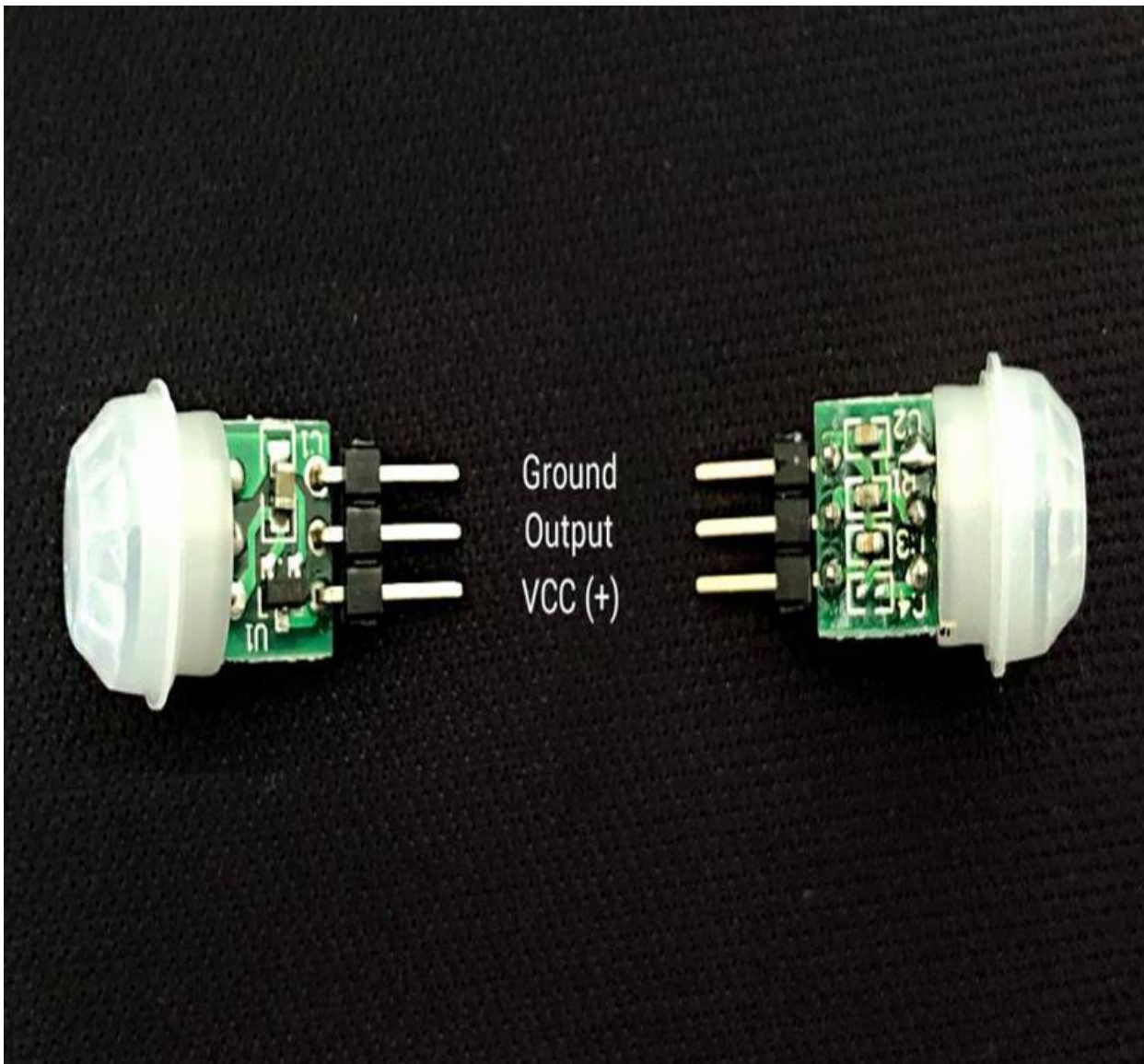
- [HC-SR501 PIR Bewegungssensor](#)*
- [AM312 Bewegungssensor](#)*

HC-SR501 PIR Bewegungssensor



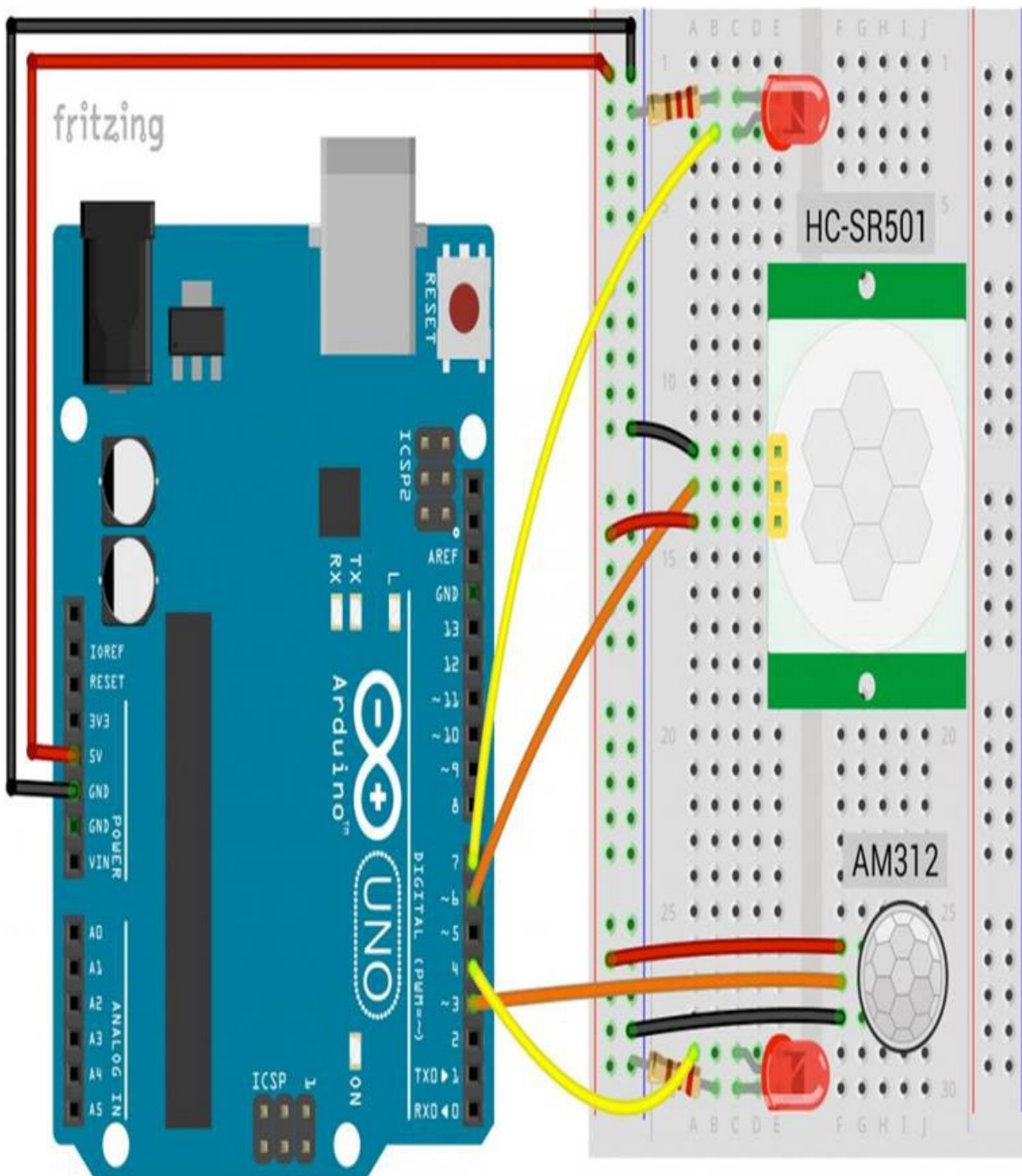
Der HC-SR501 ist ein PIR Bewegungssensor, der über zwei Potentiometer zur Einstellung von Empfindlichkeit und Schaltdauer verfügt. Des Weiteren kann man ihn per Jumper als Einschalter oder Ausschalter konfigurieren. Er ist für eine Betriebsspannung von 5V ausgelegt und verbraucht 65mA.

AM312 Bewegungssensor



Wesentlich kleiner als der HC-SR501 bietet der AM312 weniger Einstellmöglichkeiten. Er lässt sich dafür in kleinere Aufbauten integrieren, funktioniert in einem Spannungsbereich zwischen 2.7 – 12V und verbraucht weniger als 1mA, was ihn besonders in Batterie gestützter Verwendung interessant macht.

Arduino Bewegungsmelder Schaltplan



Die Schaltung ist für beide Sensoren sehr einfach. Sie werden nur mit GND, 5V+ und einem digitalen Input verbunden.

Code

```
const int PIR1_PIN = 3;
const int LED1_PIN = 4;

const int PIR2_PIN = 6;
const int LED2_PIN = 7;

void setup() {
  pinMode(PIR1_PIN, INPUT);
  pinMode(LED1_PIN, OUTPUT);
  pinMode(PIR2_PIN, INPUT);
  pinMode(LED2_PIN, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  digitalWrite(LED1_PIN,digitalRead(PIR1_PIN));
  digitalWrite(LED2_PIN,digitalRead(PIR2_PIN));
  delay(20);
}
```

Der Code zum Testen ist überschaubar. Im Grunde wird der digitale Pin ausgelesen und dessen Wert an einen anderen digitalen Pin übergeben. Somit leuchten die jeweiligen LEDs, sobald die Bewegungsmelder ausschlagen. Die PIRs sind also wie digitale Inputs zu verstehen. Sie benötigen keine [Pull-Up Widerstände](#).