

HC-SR501 PIR Motion Sensor



Beschreibung:

- Modell: HC-SR501.
- Farbe: wie in den Abbildungen gezeigt.
- Material: Durable Staku laminate (CCL).
- Die Empfindlichkeit und Haltezeit kann eingestellt werden
- Detektionsrang: ca.7m / 23 feet
- Detektionswinkel: niedriger als 100 Grad
- Betriebsspannung Rang: DC 4,5- 20
- PCB Groesse: ca.32mm x 24mm / 1,3 x 0,9 Inch
- 100% nagelneu und hohe Qualität menschlichen Körper Pyroelektrizität Infrarot PIR Motion Sensor-Modul.

Anwendungen

1. Security-Produkte
2. der menschliche Körper-Sensoren-Spielzeug
3. der menschliche Körper Sensor Beleuchtung
4. industrielle Automatisierung und Kontrolle usw..

Notizen

- Das Potentiometer im Uhrzeigersinn, um die Entfernung Erfassungsbereich angepasst erhöht (ca. 7 Meter), im Gegenteil, Erfassungsbereich verringert (ca. 3 Meter).
- Delay Einstellung Potentiometer im Uhrzeigersinn, Sensor Verzögerung länger (über 300 s), die andere Hand, Induktion durch die kurze Verzögerung (ca. 5 s).

Funktionsprinzip

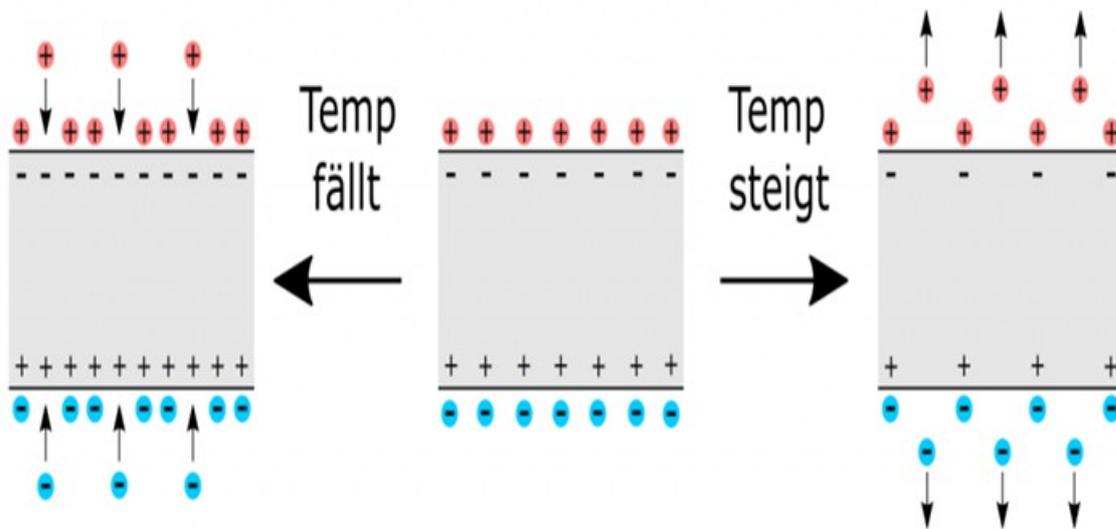


Ich fange mal mit dem Namen an, denn der verrät schon einiges über die Funktionsweise. Für die Abkürzung „PIR“ habe ich zwei Erklärungen gefunden: meistens **P**assive **I**nfrared **S**ensor, seltener **P**yroelectric **I**nfrared **S**ensor. Passiv bedeutet, dass der Sensor selbst keine Signale aussendet, so wie es beispielsweise die Entfernungssensoren [HC-SR04](#) und [VL53L0X](#) tun, die ich in meinen letzten Beiträgen beschrieben habe. Oder der Mikrowellen-Radar Sensor RCWL-0516, über den ich in meinem nächsten Beitrag schreiben werde. Infrarot bedeutet, dass der HC-SR501 Signale im Infrarotwellenlängenbereich, also im Bereich der Wärmestrahlung, empfängt und auswertet. Dadurch ist er besonders zur Detektion von Personen oder Tieren geeignet.

Der pyroelektrische Effekt

Der pyroelektrische Effekt ist etwas schwerer zu erklären bzw. zu verstehen. Es gibt chemische Verbindungen, die aufgrund ihrer Kristallstruktur einen permanenten Dipol besitzen, d.h. die eine Seite ist positiv geladen, die andere negativ. Wer sich an den Chemieunterricht erinnert, der weiß vielleicht noch, dass z.B. Wassermoleküle Dipole sind, da die Wasserstoffatome und der Sauerstoff unterschiedlich stark an den Elektronen „ziehen“. Das ist bei pyroelektrischen Verbindungen so ähnlich, nur dass es hier nicht um Moleküle, sondern um ganze Kristalle geht. Die Natur mag keine Potentialunterschiede, da sie energetisch ungünstig sind. Deshalb sammeln sich auf den Polen entgegengesetzte Ladungen aus der Umgebung.

Und jetzt kommt der relevante Punkt, den man sich beim PIR zu Nutze macht. Der pyroelektrische Effekt ist temperaturabhängig, und zwar unter anderem weil sich der Kristall mit steigender Temperatur ausdehnt und sich damit das Dipolmoment ändert. Dadurch hat man bei Temperaturänderung auf einmal zu viele oder zu wenige Ausgleichladungen und das Gleichgewicht muss sich neu einstellen. Verbindet man nun beide Pole leitend miteinander, stellt man einen Potentialsprung bzw. einen Stromfluss fest.



Tempera

turabhängigkeit des pyroelektrischen Effekts

Man kann den Kristall übrigens auch durch Druck verformen und so das Gleichgewicht stören. Das ist der piezoelektrische Effekt. Jeder pyroelektrische Kristall ist deshalb auch piezoelektrisch. Umgekehrt nicht, aber das auszuführen würde dann doch den Rahmen sprengen. Wer mehr über Pyroelektrizität wissen möchte, kann sich [hier](#) auch der Website des Zentrums für Pyroelektrizität informieren (was es alles gibt!) oder auf [Wikipedia](#).

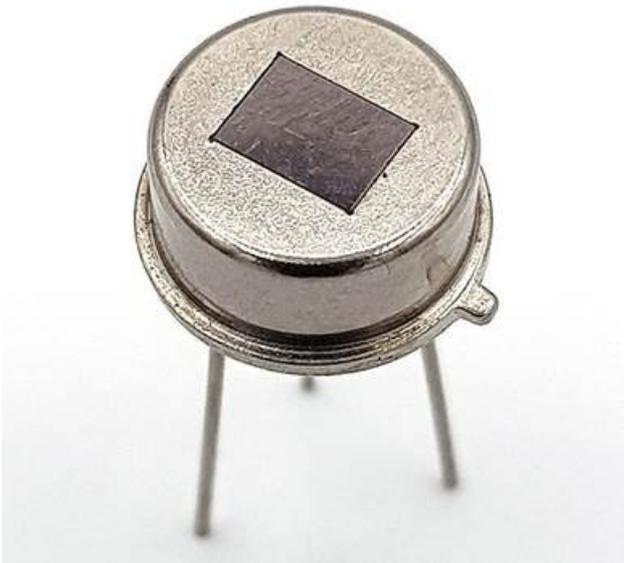
Wie der HC-SR501 den pyroelektrischen Effekt nutzt

Trifft die Infrarotstrahlung von zu detektierenden Lebewesen oder Gegenständen auf einen pyroelektrischen Kristall, dann wärmt sich dieser auf und es kommt zu der oben beschriebenen Potentialänderung. Das allein wäre aber zu fehleranfällig für die Bewegungserkennung. Zwei Maßnahmen verstärken den Effekt im PIR und machen die Detektion sicherer:

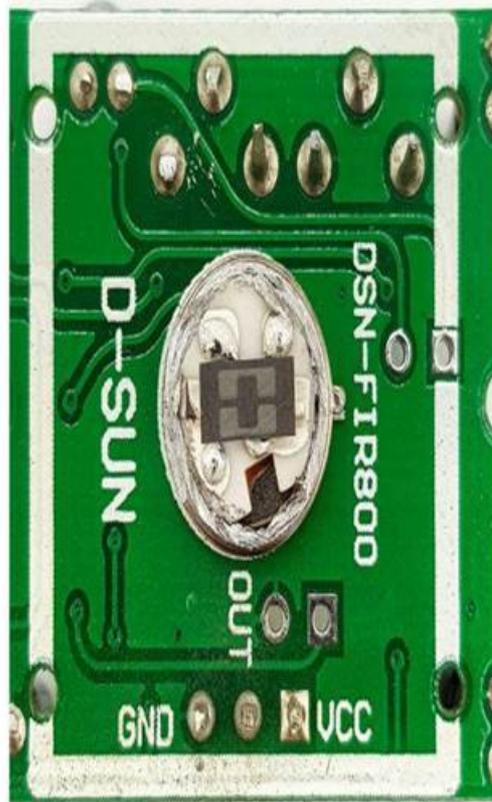
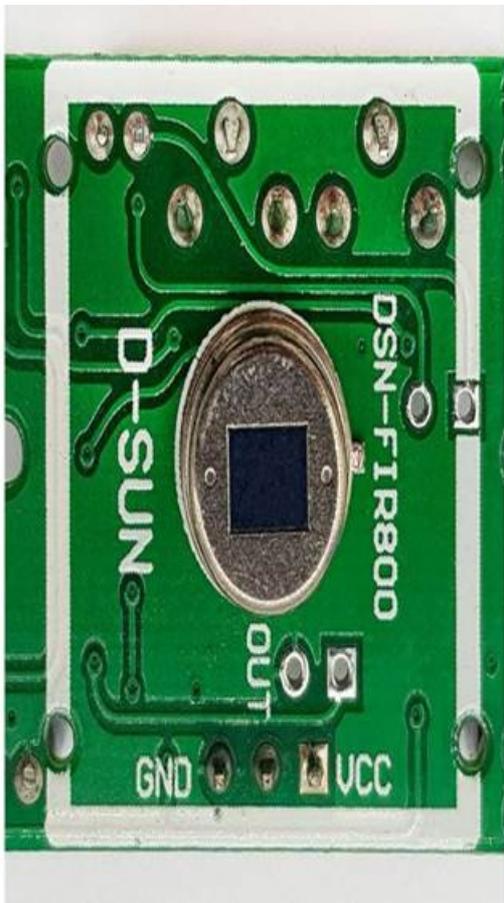
1. Die weiße Kappe auf dem PIR ist keine Schutzkappe, sondern besteht aus einer Reihe von Sammellinsen. Diese konzentrieren die Strahlung, was die Empfindlichkeit erhöht, und sie sorgen für eine Segmentierung des Erfassungsbereichs. Eine Person, die am Sensor vorbeigeht, wird so nacheinander von unterschiedlichen Segmenten erfasst.
2. Man verwendet zwei Kristalle und verschaltet sie so, dass sie Potentialsprünge in unterschiedliche Richtungen verursachen. Zudem sind die Kristalle nebeneinander angeordnet, so dass eine vorbeilaufende Person die Kristalle zeitlich versetzt anregt. Dieser Effekt wird durch die Segmentierung der Kappe verstärkt. Erst so bekommt man Signale die sich deutlich vom Hintergrund bzw. Änderungen des Hintergrunds abheben. Die Auswertung erfolgt über Komparatorschaltungen, die in dem auffälligen [BISS0001](#) Chip auf dem Modul untergebracht sind.

Ein tieferer Blick in den HC-SR501

Das Herzstück des Moduls ist der eigentliche Infrarotsensor, den man auch separat kaufen kann.



Da ich neugierig war habe ich ein Modul geopfert und habe den Sensor mit meiner Dremel-Miniflex geöffnet:



Der HC-

SR501, rechts: geöffnet

Man kann die beiden pyroelektrischen Kristalle deutlich erkennen. Meistens kommt dabei übrigens Lithiumtantalat (LiTaO_3) zum Einsatz.

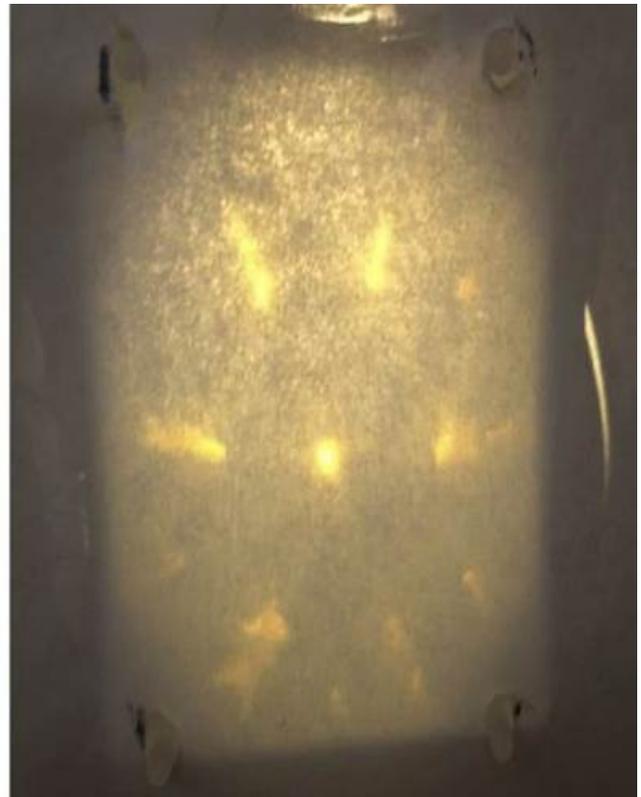
Dann habe ich mir noch die Kappen näher angeschaut. Hier gibt es zwei Versionen. Bei der einen erinnert die Struktur an einen Fußball, bei der anderen handelt es sich um eine [Fresnelsche Stufenlinse](#), wie man sie z.B. von Scheinwerfern oder Projektoren her kennt.



PI

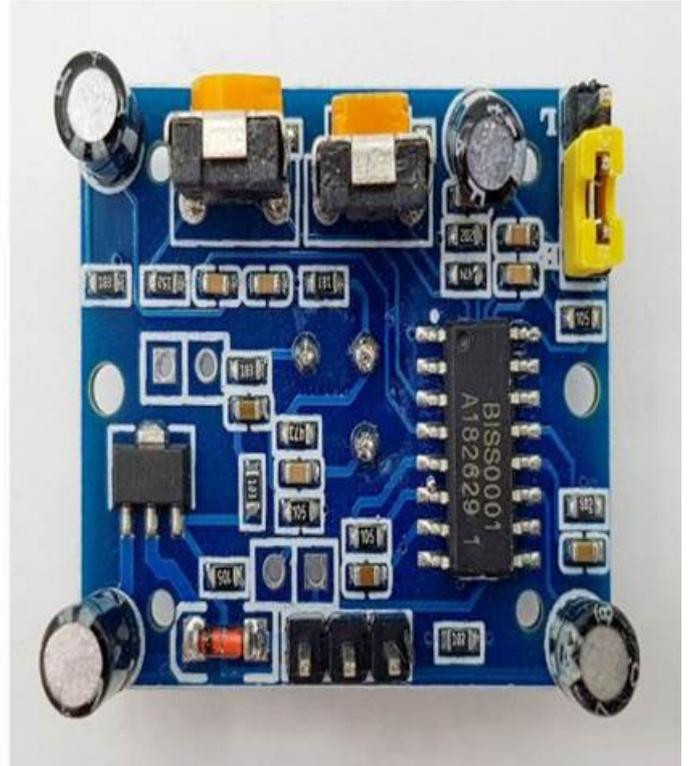
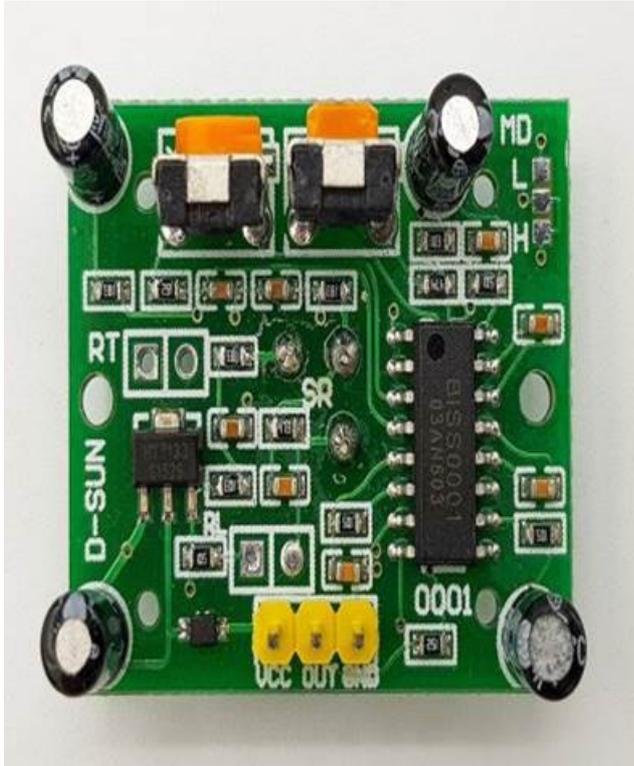
R Sammellinsen, rechts erkennt man die Fresnel-Linsenstruktur

Um zu prüfen wie die Linsen wirken, habe ich eine der Kappen passend mit ihrer Kuppel durch eine Pappe gesteckt und hinten eine „Leinwand“ aus Butterbrotpapier angebastelt. Von vorne angeleuchtet kann man gut die Brennpunkte der Einzellinsen erkennen:



Abbildungsleistung der Linsen – für ein Stück Plastik und den Zweck OK
Mehr über PIR Sensoren gibt es hier auf [Wikipedia](https://de.wikipedia.org/wiki/Passiv_infrarotstrahlung).

Moduleigenschaften



HC-SR501 Module

Egal aus welcher Quelle man sich einen HC-SR501 besorgt – rein äußerlich sind alle ziemlich gleich. Außer der Farbe und der Ausführung der Kappe ist der größte sichtbare Unterschied die Ausführung des Jumpers oben rechts, nämlich mal mit Pins zum Umstecken und mal ohne Pins zum Löten. Zu den nicht sichtbaren Unterschieden komme ich später.

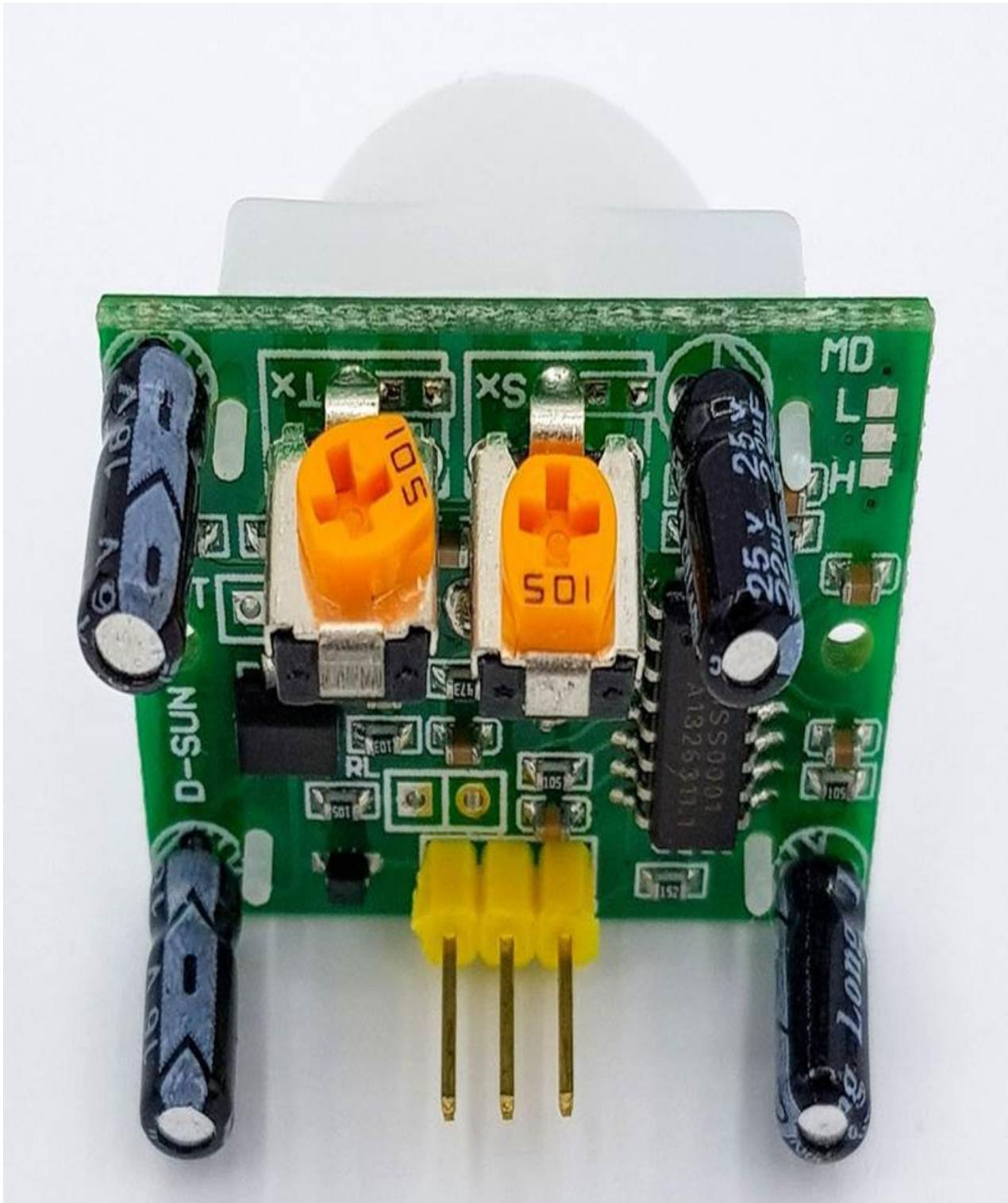
Technische Daten des HC-SR501

- **Spannungsversorgung:** 4.5 – 20 Volt an VCC und GND (manchmal leicht abweichende Angaben)
- **Reichweite:** max ca. 7 Meter, herunterregelbar auf ca. 4 Meter
- **Signallevel:** 3.3 Volt (Bewegung) / 0 Volt (keine Bewegung) an OUT
- **Signallänge:** meist wird 5 – 300 Sekunden angegeben, nach meiner Erfahrung ca. 2 – 500 Sekunden
- **Stromverbrauch:** im Wartezustand genügsame 60 – 70 Mikroampere; bei Bewegungsdetektion kurzfristig mehr, aber immer < 1 Milliampere
- **Detektionsmodi:** Single Trigger oder Repeatabl Trigger (Erklärung weiter unten)
- **„Aufwärmzeit“:** Nach Anschluss an die Stromversorgung braucht das Modul 30 – 60 Sekunden bis es einsatzbereit ist. In dieser Zeit geht OUT ein paar Male auf HIGH.
- **Sperrzeit:** 0.2 oder 3 Sekunden, je nach Modell

Zu der Reichweite ist noch zu sagen, dass diese natürlich von der Größe und der Wärmestrahlung der zu detektierenden Objekte abhängt. Man kann z.B. nicht die Reichweite auf 4 Meter herunterregeln und dann sicher sein, dass unterhalb dieses Wertes alles erkannt wird und oberhalb nichts. Man muss ein bisschen herumspielen und die Einstellung finden, die am besten zur Anwendung passt.

Aufgrund der lang einstellbaren Signalzeit braucht man für die typische Anwendung „Lampe geht bei Bewegung für x Minuten an“ keinen Microcontroller sondern lediglich noch ein Relais (wie langweilig!).

Einstellungen am HC-SR501



Zeit- (T) und Empfindlichkeitseinstellung (S) an den Potis des HC-SR501

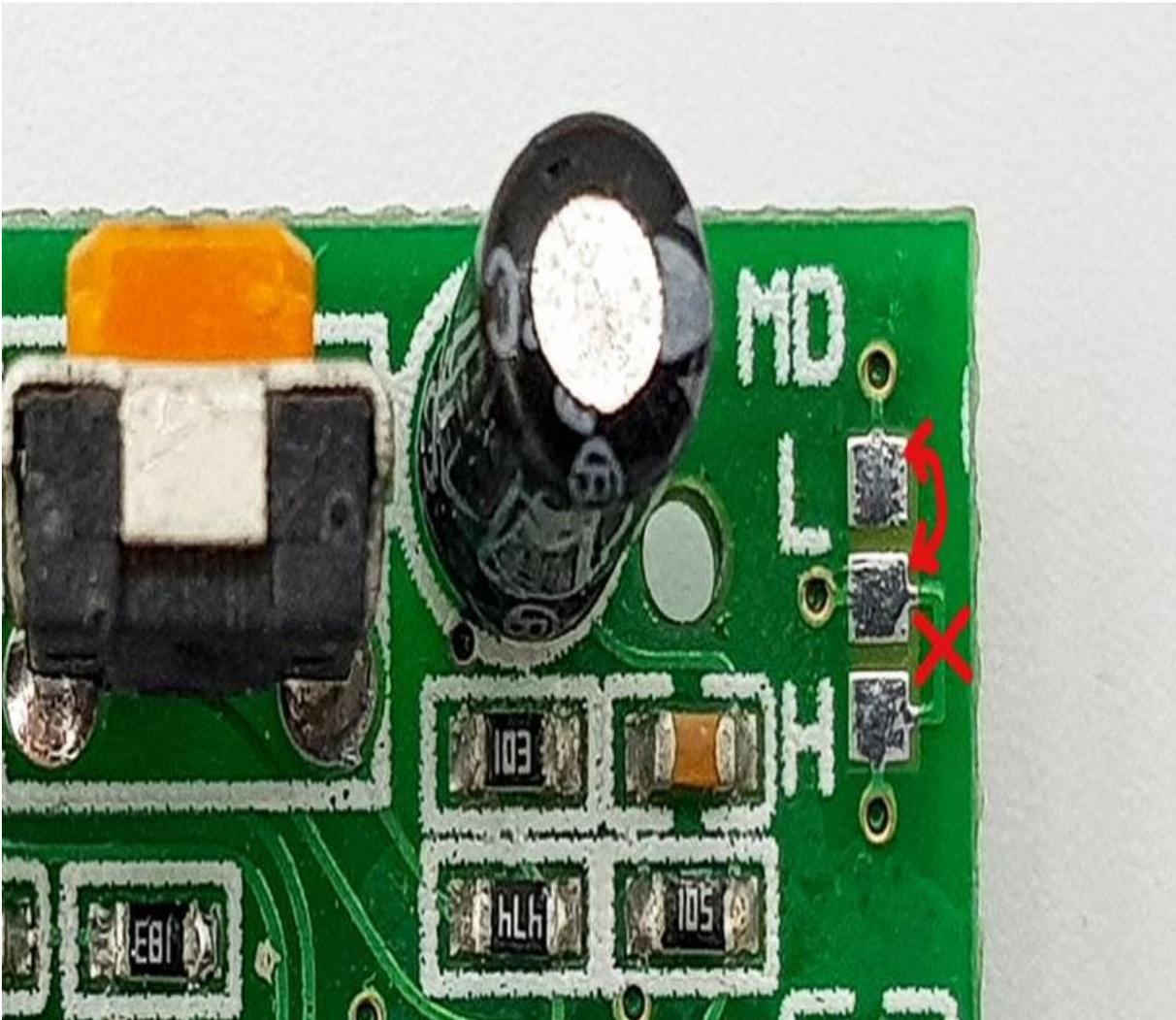
Die Länge des Signals wird am mit „T“ (Time) gekennzeichneten Poti eingestellt. Eine Drehung im Uhrzeigersinn verlängert das Signal. Das mit „S“ (Sensitivity) gekennzeichnete Poti ist für die Empfindlichkeit zuständig. Auch hier erhöht eine Drehung im Uhrzeigersinn den Wert. Bei einigen Modulen gibt es keine Kennzeichnung, die Potis sind aber immer wie oben abgebildet angeordnet.

Betriebsmodi Single und Repeatable Trigger

Am Jumper oben rechts lassen sich die Betriebsmodi „Single Trigger“ und „Repeatable Trigger“ einstellen. Ist der mittlere Kontakt mit „H“ verbunden, ist der Repeatable Trigger Modus eingestellt. Entsprechend ist der Single Trigger Modus eingestellt, wenn er mit „L“ verbunden ist. Nach einer

Bewegungsdetektion geht der OUT Pin für die eingestellte Zeit auf HIGH. Ist der Repeatable Trigger Modus aktiv, dann können während dieser Zeit der Signalausgabe weitere Bewegungen detektiert werden. Bei jeder Bewegung beginnt die Ausgabzeit wieder bei Null. Bei kontinuierlicher Bewegung bleibt das OUT Signal also kontinuierlich HIGH. Beim Single Trigger Modus kann hingegen während der Signalausgabe keine Bewegung detektiert werden. Dementsprechend geht bei dieser Einstellung das OUT Signal auch bei kontinuierlicher Bewegung zwischenzeitlich auf LOW.

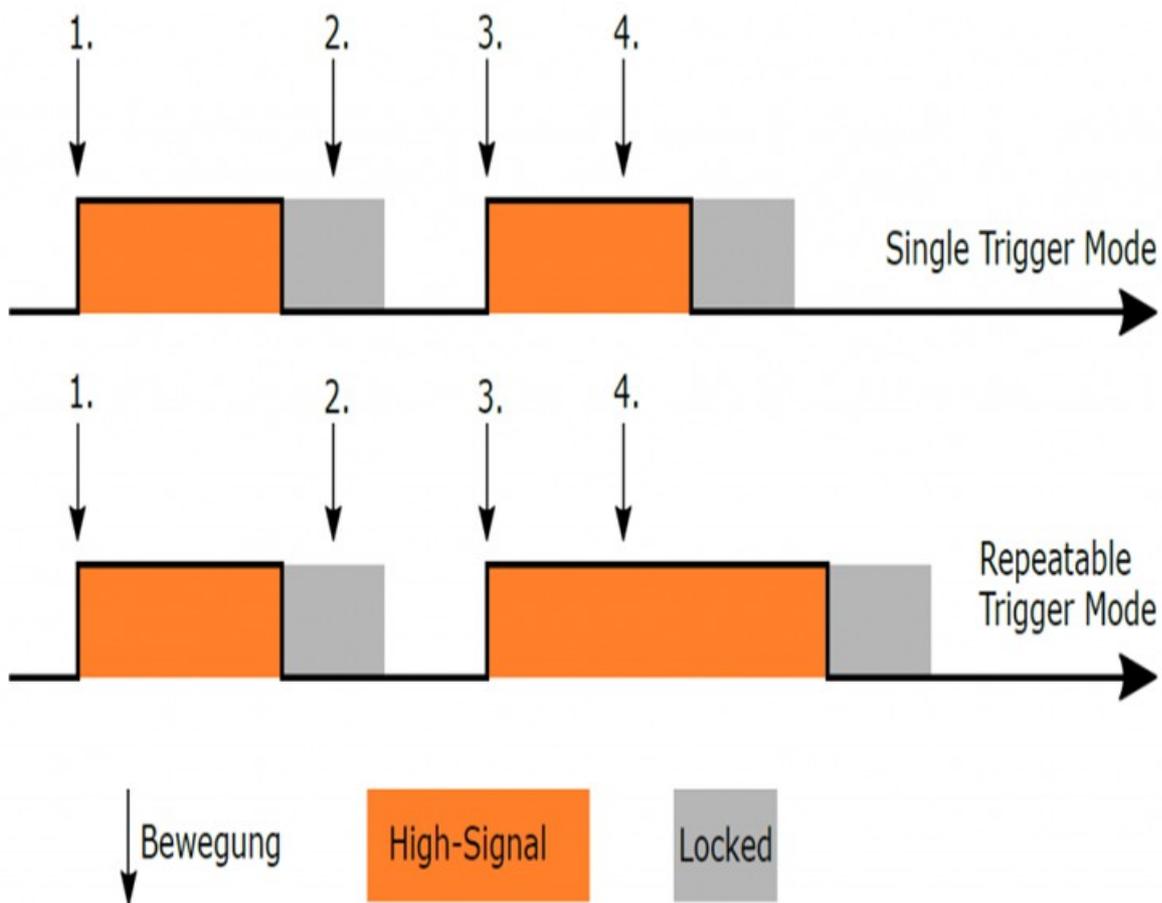
Bei den Modulen ohne Pins am Jumper ist der Repeatable Trigger Modus voreingestellt. Um zu wechseln muss man nicht nur den mittleren Kontakt mit „L“ verbinden, sondern auch die Leitung zu „H“ kappen. Das kann man mit einer Rasierklinge oder einem Cuttermesser mit frischer Klinge machen (Vorsicht!).



Vom Repeatable in den Single Trigger Modus

Ist das OUT Signal auf LOW gegangen, dann ist der HC-SR501 für eine kurze Zeit gesperrt und kann keine Bewegungen detektieren. Dieses Verhalten ist unabhängig vom Modus.

Das folgende Schema veranschaulicht das noch einmal. Bei beiden Modi führt die Bewegung 1 zum Auslösen eines Signals. Während der anschließenden Sperrzeit („locked“) bleibt die Bewegung 2 wirkungslos. Bewegung 3 verursacht wieder eine normale Auslösung. Den Unterschied macht die Bewegung 4. Im Single Trigger Modus bleibt sie folgenlos, im Repeatable Trigger Modus wird die Signalausgabe verlängert.



Au

löse- und Signalschema in Abhängigkeit des Betriebsmodus Soweit die Theorie....

So wie oben beschrieben liest man es überall. Nach meiner Erfahrung ist der Sachverhalt aber doch noch ein bisschen komplexer. Wenn nämlich während der Sperrzeit eine Bewegung im Erfassungsbereich auftritt, geht OUT manchmal nach Beendigung der Sperrzeit auf „HIGH“ obwohl in diesem Moment keine Bewegung mehr stattfindet. Das Phänomen tritt insbesondere dann auf, wenn man sich nah am Sensor befindet. Beim Herumspielen mit den Sensoreinstellungen auf meinem Schreibtisch fiel mir das auf und ich hatte das nirgends eindeutig beschrieben gefunden. Also wundert euch nicht. Woran es liegt weiß ich nicht genau, ich würde aber vermuten, dass der durch die Bewegung aufgewärmte Kristall eine gewisse Zeit braucht, bis er wieder seine Gleichgewichtstemperatur erreicht hat. Ein Schatten der Bewegung oder ein Nachleuchten sozusagen.

HC-SR501 Module mit unterschiedlichen Sperrzeiten

Einige der von mir getesteten Module haben eine Sperrzeit von ca. 3 Sekunden, bei anderen beträgt sie lediglich 0.2 Sekunden. Tendenziell scheint es so zu sein, dass die Modelle mit Pins am Jumper zu der Gruppe mit den 3 Sekunden Sperrzeit gehören und die anderen zu denen mit 0.2 Sekunden Sperrzeit. Das heißt aber nicht, dass man mit Letzteren eine besonders hohe Detektionsfrequenz erreichen kann. Der eben beschriebene Nachleuchteffekt führt hier sogar dazu, dass man bei Einstellung minimaler Signalzeit Doppelauslösungen hervorrufen kann. Das äußert sich so, dass eine Bewegung ein HIGH an OUT auslöst, kurz in den Sperrmodus geht (also LOW) und dann wieder HIGH obwohl die Bewegung längst vorbei ist.

Überaus verwirrend finde ich dazu auch die im Umlauf befindlichen Datenblätter (z. B. dieses [hier](#)), deren Inhalt sich wortwörtlich herauskopiert überall in Produktbeschreibungen und z. T. auch in Blogs wiederfindet:

- Delay time: Adjustable (.3->5min)
- Lock time: 0.2 sec

...

Delay time	5-300S(adjustable) Range (approximately .3Sec -5Min)
Block time	2.5S(default)Can be made a range(0.xx to tens of seconds)

Auszug aus einem typischen Datenblatt

Wahrscheinlich versteht das niemand und deswegen wird es einfach kopiert. Oder könnt ihr mir das erklären?

Aufgrund dieser Eigenheiten ist es am sichersten, wenn man den LOW zu HIGH Wechsel an OUT ausliest und dann für mindestens 5 Sekunden alles was an OUT passiert ignoriert. Also ungefähr so:

Movement_Detection.ino

```
int triggerPin = 2;
void setup() {
  pinMode(triggerPin, INPUT);
}
void loop() {
  if(digitalRead(triggerPin)==HIGH){
    deineAktionen();
    delay(5000);
  }
}
void deineAktionen(){
  // deine Aktionen, z.B. Lampe anschalten, Alarm an o.ä.
}
```

oder so:

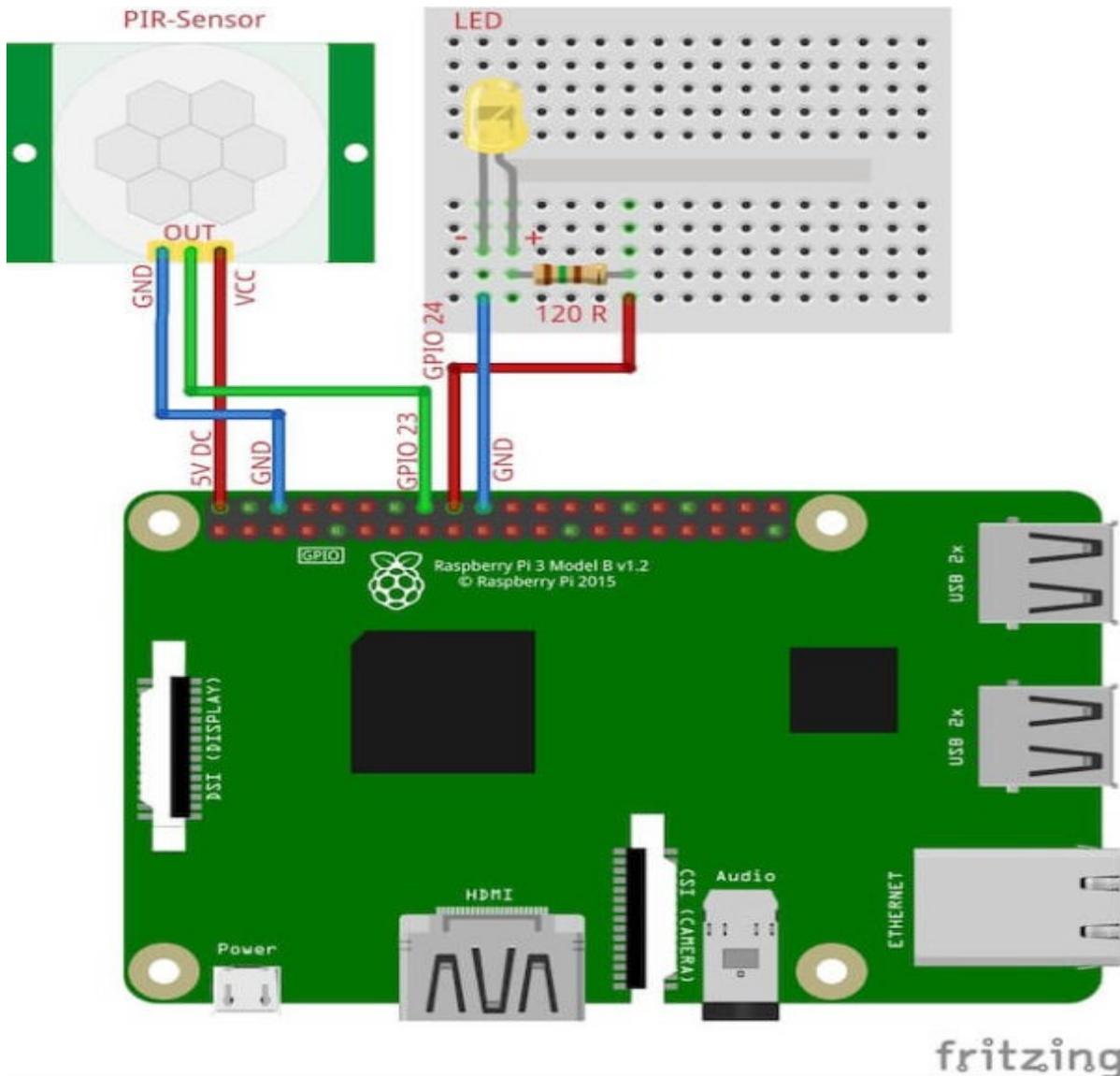
Movement_Detection_mit _Interrupt.ino

```
byte interruptPin=2;
volatile bool movement = false;
void setup() {
  pinMode(interruptPin, INPUT);
  attachInterrupt(digitalPinToInterrupt(interruptPin), movementDetected, RISING);
}
void loop() {
  if(movement){
    deineAktionen();
    delay(5000);
    movement=false;
  }
}
void deineAktionen(){
  // dein Code
}
void movementDetected(){
  movement=true;
}
```


Installation und Anschluss

Wie oben schon einmal erwähnt, bauen wir eine einfache LED-Lichtschaltung für den HC-SR-501 auf, um die zu erklären, wie man den Sensor anschließt und programmiert. Was du dazu benötigst, findest du weiter oben unter "Was brauchst du?".

Die Schaltung kannst du nach folgender Zeichnung aufbauen:



Erklärung zur Schaltung:

Der Sensor wird an VCC mit 5V DC über den Raspberry Pi versorgt. GND des Sensors wird auf einen GND des Pi geschlossen.

Der OUT-Ausgang des Sensors wird an einen beliebigen GPIO Anschluss geschlossen.

Die LED schaltest du in Reihe mit dem Widerstand auf einen anderen GPIO und GND.

Der Sensor hängt bei uns im Beispiel am GPIO 23 und die LED am GPIO 24.

Programmierung des Raspberry PIR

Im Folgenden findest du den Programmcode für die Bewegungserkennung. Die Kommentare im Code, geben dir Hinweise zu den einzelnen Befehlszeilen.

Erstelle als erstes eine neue Datei mit dem Namen „*electreeks_bewegungssensor.py*“

Öffne die Datei mit einem Editor oder IDLE deiner Wahl und trage dort den Programmcode ein:

```
#Import der Python Module
import RPi.GPIO as GPIO
import time

#Variablen definieren
PIR_Sensor = 23
LED = 24

#GPIOs mit GPIO-Nummer ansprechen
GPIO.setmode(GPIO.BCM)

#Ein- und Ausgaenge definieren
GPIO.setup(PIR_Sensor, GPIO.IN)
GPIO.setup(LED, GPIO.OUT)

print("Das Programm wurde gestartet.")

#Callback Funktion erstellen
def Bewegung(PIR_Sensor):
    print("Dein Electreeks Sensor hat eine Bewegung erkannt!")
    GPIO.output(LED, True)
    time.sleep(5)
    GPIO.output(LED, False)

#Hauptprogramm starten
if __name__ == "__main__":
    try:
        #Sobald eine Bewegung erkannt wurde, fuehre die Funktion aus.
        GPIO.add_event_detect(PIR_Sensor, GPIO.RISING, callback=Bewegung)
        while True:
            time.sleep(0)

    #Programm beenden mit Strg + c
    except KeyboardInterrupt:
        print ("Das Programm wurde beendet.")
        GPIO.cleanup()
```

Wenn du fertig bist, speicher deine Änderungen. Führe nun mit dem Terminal auf deinem Raspberry Pi das Programm mit dem Befehl: **sudo python electreeks_bewegungssensor.py** aus.

Wenn alles geklappt hat, muss sich bei einer Bewegung die LED für 5 Sekunden einschalten und im Terminal der Kommentar „Dein Electreeks Sensor hat eine Bewegung erkannt!“ erscheinen. Das geht solange, bis der PIR keine Bewegung mehr erkennt. Du kannst das Programm jederzeit mit Strg + c beenden.