

# Infrarothindernissensor

## Benötigtes Material

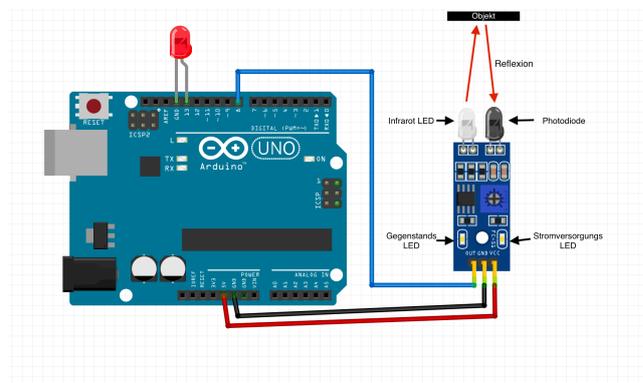
- Arduino Uno
- USB-Kabel für den Arduino
- Infrarotsensor (FC-51)
- LED
- 3x MW-Kabel
- Kleiner Schraubenzieher
- Kleines Stück dunkles Tape

## Beschreibung

Der Infrarotsensor „Flying fish“ ist ein Hindernissensor. Das bedeutet, er kann mithilfe einer Infrarot-LED und einer Photodiode wahrnehmen, wenn sich ein Hindernis in einem gewissen Bereich vor ihm befindet. Die Infrarot-LED sendet hierbei ein Infrarotsignal aus, welches von einem Gegenstand reflektiert und von der Photodiode wieder aufgefangen wird. Der Sensor verfügt darüber hinaus noch über zwei kleine grüne LED's. Die LED, die sich über dem „Vcc“ Pin befindet zeigt an, ob der Sensor mit Strom versorgt wird. Die LED, die sich über dem „Out“ Pin befindet leuchtet auf, wenn der Sensor einen Gegenstand wahrnimmt. Auf dem Sensor befindet sich ein Potentiometer, welches genutzt wird, um die Reichweite des Sensors einzustellen. Laut Spezifikation erkennt der Sensor Hindernisse zwischen 2-30 cm. Da der Sensor nur über einen digitalen Input verfügt, können wir lediglich anzeigen lassen, ob sich ein Gegenstand in der eingestellten Reichweite befindet, aber nicht wie weit dieser entfernt ist.

## Verkabelung

IR-Sensor	Arduino
GND	GND
Vcc	5V
OUT	8



Schaltplan

## Code

Mithilfe des Programms kann man beispielhaft eine LED aufleuchten lassen, wenn der IR-Sensor ein Hindernis wahrnimmt.

```
int IR_SENSOR = 8; // Setze den Pin 8 als Anschluss für den Sensor
int LED       = 13; // Setze den Pin 13 als Anschluss für die LED

void setup() {
  pinMode(IR_SENSOR, INPUT); // Setze den Sensor-Pin als INPUT, da dieser ein
                             // Signal an den Arduino senden soll
  pinMode(LED, OUTPUT);     // Setze die LED als OUTPUT
}
```

```

void loop() {
  if (digitalRead(IR_SENSOR) == 0) { // Auslesen, ob IR-Sensor das Signal gibt,
                                     // dass ein Hindernis vor ihm ist
    digitalWrite(LED, HIGH); // Wenn ja, soll die LED leuchten
  } else {
    digitalWrite(LED, LOW); // Wenn nicht, soll die LED nicht leuchten
  }
}

```

## Aufgaben

### ① Beschreibe, was der Code tut!

Der Arduino bekommt den IR-Sensor als Inout und die LED als Output im Setup gesetzt. In der Loop wird geprüft, ob der Infrarotsensor ein Hindernis vor sich hat, dann leuchte die LED, wenn nicht bleibt die LED aus.

### ② Wie funktioniert ein Infrarotsensor?

Siehe in die Beschreibung.

### ③ Baut den Code so um, dass wenn der Sensor ein Hindernis entdeckt eine rote LED leuchtet, und bei freier Bahn eine grüne.

```

int IR_SENSOR = 8; // Setze den Pin 8 als Anschluss für den Sensor
int LED_rot   = 13; // Setze den Pin 13 als Anschluss für die LED bei Hindernis
int LED_grün  = 12; // Setze den Pin 12 als Anschluss für die LED bei freier Bahn

void setup() {
  pinMode(IR_SENSOR, INPUT); // Setze den Sensor-Pin als INPUT, da dieser ein
                              // Signal an den Arduino senden soll
  pinMode(LED, OUTPUT);      // Setze die LED als OUTPUT
}

void loop() {
  if (digitalRead(IR_SENSOR) == 0) { // Auslesen, ob IR-Sensor das Signal gibt,
                                     // dass ein Hindernis vor ihm ist
    digitalWrite(LED_rot, HIGH); // Wenn ja, soll die LED leuchten
  } else {
    digitalWrite(LED_grün, HIGH); // Wenn nicht, soll die LED nicht leuchten
  }
}

```

## Häufige Fragen und Probleme

### Der Infrarotsensor sieht nicht weit genug/ sieht zu weit!

Das Potentiometer kann mit einem kleinen Kreuzschraubenzieher eingestellt werden. Nach rechts drehen steigert die Entfernung, bis ein Objekt wahrgenommen wird. Nach links drehen verringert die Entfernung.

### Der Infrarotsensor leuchtet auf, bevor ein Objekt vor ihm ist.

Da der IR-Sensor mit einer IR-LED und Reflexion arbeitet, kann es zu Störungen von außen kommen. Zum einen spielt bei Reflexion von Wellen immer die Farbe des Objektes eine Rolle, von dem reflektiert wird. Das bedeutet, dass wenn gegenüber von dem Sensor eine helle Wand ist, er diese, abhängig vom Licht, ebenfalls als Objekt wahrnehmen kann. Es empfiehlt sich, einen schwarzen Hintergrund zu wählen. Ebenso kann es vorkommen, dass die IR-LED streut und so eine unerwünschte Reflexion das Signal auslöst. Um dies zu verhindern, kann um die beiden Dioden seitlich ein schwarzes Tape gebunden werden, so dass es nach vorne hin offen ist. Der Sensor sollte dabei eingeschaltet sein, um zu verhindern, dass das Tape zu weit vorne angebracht wird und der Sensor das Tape als Hindernis erkennt.

**Der Sensor leuchtet, aber die angeschlossene LED nicht.**

Es ist darauf zu achten, dass die LED als Output und der Sensor als Input festgelegt ist. Der Sensor soll ein Signal an den Arduino senden und dieser wiederum an die LED.

**Konkretes Problem zwischen Potentiometer und Entfernung.**

Der Verkäufer sichert eine Reichweite zwischen 2cm und 30 cm zu. Diese ist nicht realistisch. Beim Kalibrieren des Sensors kann es zu einigen Komplikationen kommen, die dazu führen, dass der Sensor automatisch aufleuchtet, sobald man das Potentiometer bis zu einem gewissen Punkt nach rechts dreht. Wenn man nun das Potentiometer wieder so weit zurück dreht, dass der Sensor gradeso nicht aufleuchtet, fällt einem auf, dass die Reichweite dann weit unter dem Maximum liegt. Dies ergibt sich aus den bereits angesprochenen Problemen der Reflexion und der Bauweise des Sensors und lässt sich nur bis zu einem gewissen Punkt verhindern. Daher gilt es, bei diesem Sensor auszuprobieren, unter welchen Bedingungen das beste und leicht replizierbare Ergebnis erzielt werden kann.

**Die angeschlossene LED als Feedbacksignal ersetzen.**

Wenn das visuelle Feedback z.B. durch ein LCD-Display ersetzt werden soll, lässt sich dieses ganz einfach umsetzen. Einfach nochmal das Material für das Display anschauen und es wie gewohnt anschließen. Danach den Code für den IR-Sensor um den Code bis zu dem Loop für das LCD-Display ergänzen. Im Loop dann klar machen, wann das LCD Display etwas anzeigen soll und unter welcher Bedingung. Anschließend die passenden Zeilen im Code ergänzen und gegebenenfalls etwas umschreiben.

Das Material und dessen Inhalte sind - sofern nicht anders angegeben - lizenziert unter der Creative Commons Lizenz CC BY-NC-SA 4.0 (für den vollständigen Lizenztext siehe <https://creativecommons.org/licenses/by-sa/4.0/legalcode>)

