

**Teil 2**

# Lektion

# 12

**Infrarotempfängermodul**

## Übersicht

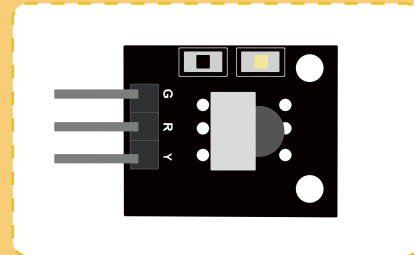
Durch eine IR (Infrarot) Fernbedienung lässt sich Ihr Projekt auf einfache Weise kabellos steuern.

IR Fernbedienungen sind einfach aufgebaut. In dieser Lektion werden wir einen Infrarotempfänger mit dem UNO Board verbinden und ihn dann mittels einer geeigneten Bibliothek einbinden.

In unserem Sketch sind bereits alle hexadezimalen IR Codes für alle Tasten der Fernbedienung eingetragen. Wir werden im Sketch überprüfen, welche Taste gedrückt wurde und ob sie im Moment gedrückt wird.

### Benötigte Bauteile:

- (1) x Elegoo Uno R3
- (1) x IR Empfängermodul
- (1) x IR Fernbedienung
- (3) x W-M Kabel (Weiblich zu Männlich DuPont Jumper Kabel)



## Einführung in die Komponenten

### IR Empfänger (Sensor):

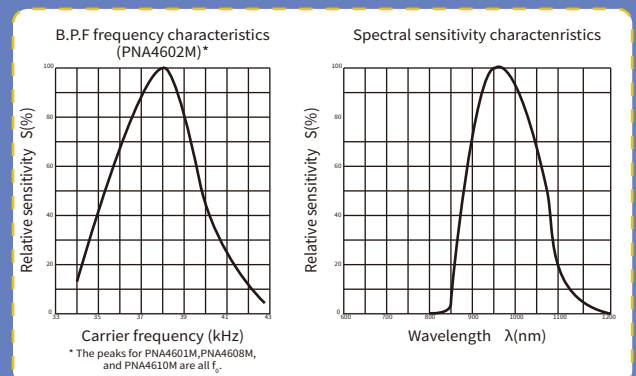
**Infrarotempfänger** sind kleine Mikrochips mit einer Fozelle, die darauf getrimmt ist Infrarotsignale zu erkennen. Die Infrarottechnik wird in zahlreichen Geräten eingesetzt: jeder TV und jeder DVD Player hat einen Infrarotempfänger auf der Vorderseite, der auf das Infrarotsignal der Fernbedienung reagiert. In der dazugehörigen Fernbedienung befindet sich eine passende IR LED, die verschiedene Infrarotimpulse ausgibt, je nachdem welche Taste gedrückt wurde und ob der TV an oder ausgeschaltet oder einfach nur der Kanal gewechselt werden soll. Infrarotlicht ist unsichtbar für das menschliche Auge, daher sehen wir die IR LED nicht leuchten.

**IR** Empfänger unterscheiden sich in einigen Punkten zu normalen, sogenannten „CdS“-Fotzellen. IR Empfänger filtern lediglich Infrarotlicht heraus und können kein sichtbares, normales Licht registrieren. Auf der anderen Seite sind CdS-Fotzellen gut dafür geeignet gelb/grünes sichtbares Licht zu erkennen, funktionieren aber nicht bei Infrarotlicht.

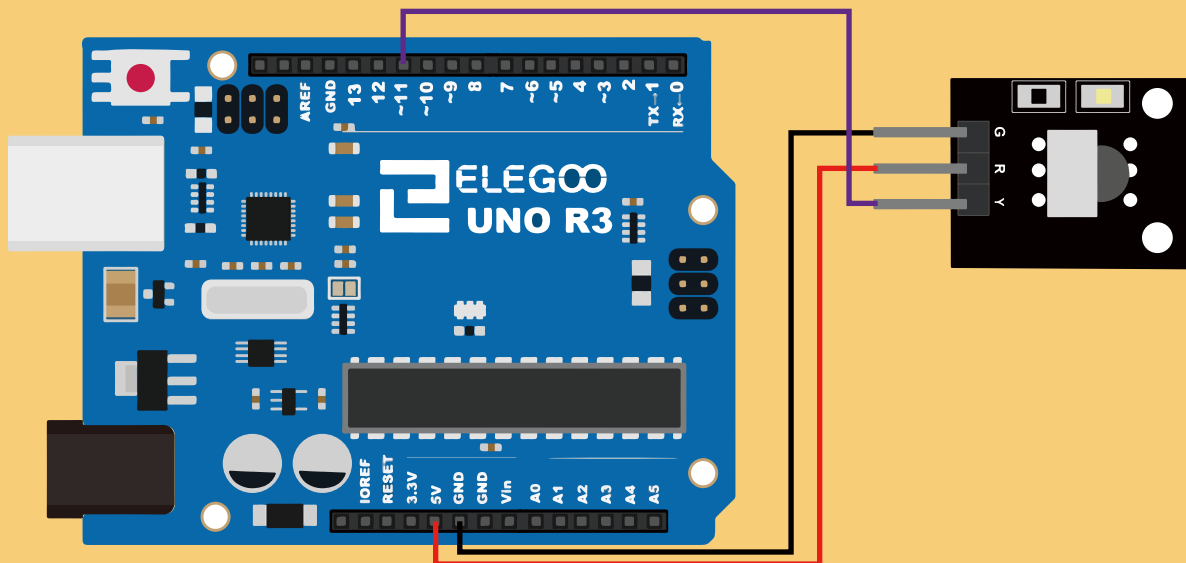
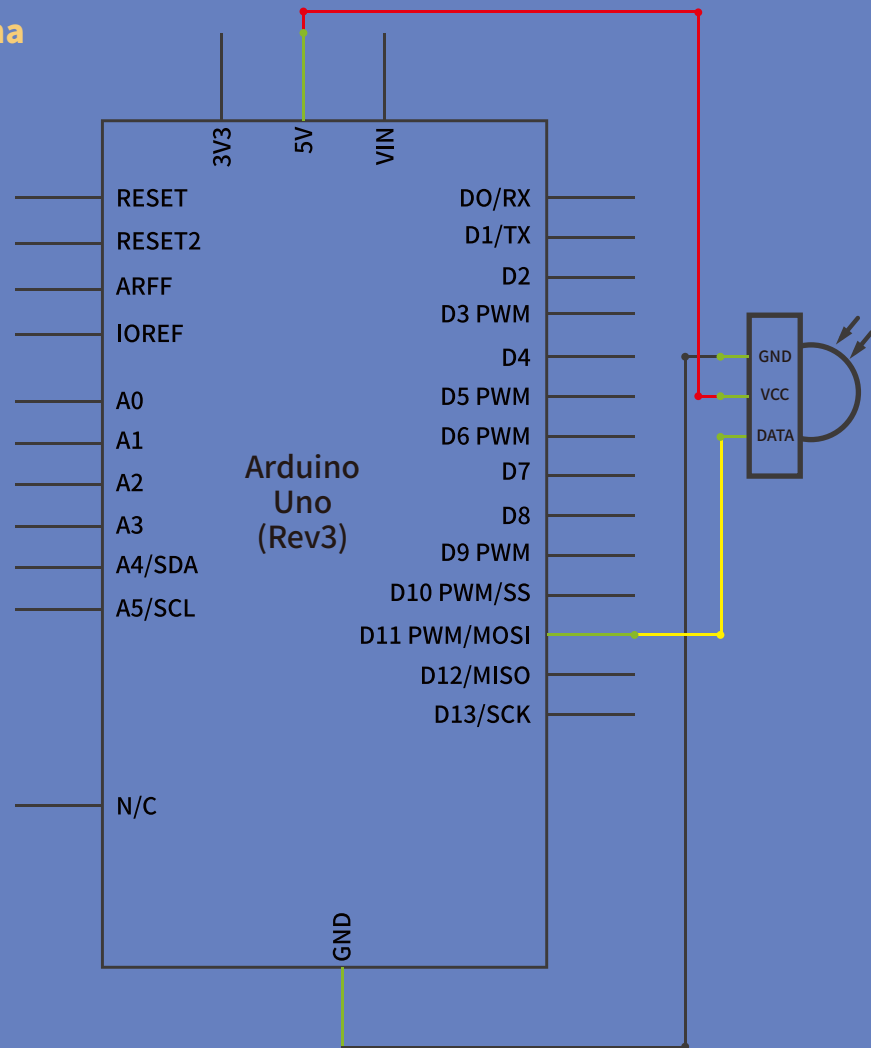
**Infrarotempfänger** haben einen Demodulator eingebaut, der auf ein moduliertes Infrarotsignal bei 38kHz reagiert. Eine einfach blinkende Infrarot-LED würde nichts bewirken, das Licht muss PWM-moduliert sein und eine Frequenz von 38kHz haben. Fotzellen haben dagegen gar keinen eingebauten Demodulator und können jede Frequenz feststellen, die in ihrer Frequenzzeit liegt (etwa 1kHz).

**Infrarotempfänger** geben ein digitales Signal aus. Wenn sie ein 38kHz IR-Signal empfangen, geben sie ein LOW-Signal (0V) aus, während sie wenn sie nichts empfangen HIGH (5V) ausgeben. Fotzellen dagegen funktionieren wie Widerstände. Ihr Widerstand ändert sich abhängig davon, wie viel Licht sie empfangen.

**Wie** Sie in den Graphen erkennen können, liegt die Spitzenfrequenz bei 38kHz und die optimale Wellenlänge des Lichtes beträgt 940nm. Es können zwar alle Frequenzen im Bereich von 35kHz bis 41kHz und alle Wellenlänge von 850 bis 1100nm benutzt werden, aber die Empfangsgenauigkeit nimmt ab und die Signale würden nicht so gut erkannt werden. Daher gehen Sie bitte sicher, passende LEDs zum senden zu benutzen. Ob Ihre LED die richtige Wellenlänge hat, können Sie im Datenblatt der LED überprüfen. Der optimale Wert wäre 940nm (nicht sichtbar).



## Verbindungsschema



Es gibt drei Verbindungen zum IR-Empfänger.

Die Anschlüsse sind: Signal, Spannung und GND.

Das “-” steht für GND, “S” ist das Signal, und der mittlere Pin ist die Spannung (5V).

**Schaltplan**

## Code

- Nach dem Verbinden der Komponenten öffnen Sie bitte den Sketch im Code-Ordner unter „IR\_Receiver\_Module“ und laden ihn auf Ihr UNO Board hoch. Bei Fragen zum Hochladen eines Sketches schauen Sie sich bitte Lektion 5 in part 1 noch einmal an.
- Bevor Sie diesen Sketch hochladen können, müssen Sie die „IRremote“-Bibliothek installiert haben. Sonst wird sich Ihr Sketch nicht hochladen lassen.
- Für Hinweise wie man eine Bibliothek einbindet, gehen Sie zurück zu Lektion 5 in part 1.
- Als nächstes verschieben wir zeitweise den Ordner „RobotIRremote“ aus dem Bibliotheksordner der IRremote-Bibliothek, da wir zum Programmieren eine andere Version dessen benutzen werden, was sonst zu Konflikten führen würde.
- Sie können den Ordner nach dem Hochladen wieder zurück in den Bibliotheksordner

```
switch(results.value)

{
case 0xFFA25D: Serial.println("POWER"); break;
case 0xFFE21D: Serial.println("FUNC/STOP"); break;
.....
default:
Serial.println(" other button ");

} // End Case
```

switch...case

### [Control Structure]

#### Beschreibung

- Wie if-Anweisungen steuert “switch case” den Programmfluss, in dem es Programmierern ermöglicht, unterschiedlichen Code anzugeben, der unter verschiedenen Bedingungen ausgeführt werden soll. Insbesondere vergleicht eine switch-Anweisung den Wert einer Variablen mit den in case-Anweisungen angegebenen Werten. Wenn eine case-Anweisung gefunden wird, deren Wert mit dem der Variablen übereinstimmt, wird der Code in dieser case-Anweisung ausgeführt.
- Das Schlüsselwort break beendet die switch-Anweisung und wird normalerweise am Ende eines jeden Falls verwendet. Ohne eine break-Anweisung führt die switch-Anweisung die folgenden Ausdrücke ("falling-through") weiter aus, bis eine break-Anweisung oder das Ende der switch-Anweisung erreicht ist.

## Syntax

```
switch (var) {  
  case label1:  
    // statements  
    break;  
  case label2:  
    // statements  
    break;  
  default:  
    // statements  
    break;  
}
```

## Parameters

**var:** Eine Variable, deren Wert mit verschiedenen Fällen verglichen werden soll. Zulässige Datentypen: int, char.

**label1, label2:** Konstanten. Zulässige Datentypen: int, char.

## Returns

Nothing