

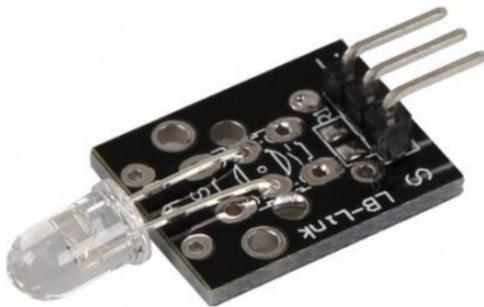
# KY-005 Infrarot Transmitter Modul

Aus Linkerkit.de

## Inhaltsverzeichnis

- 1 Bild
- 2 Technische Daten / Kurzbeschreibung
- 3 Pin-Belegung
- 4 Codebeispiel Arduino
  - 4.1 Codebeispiel ON/OFF
  - 4.2 Codebeispiel Fernbedienung
- 5 Codebeispiel Raspberry Pi
  - 5.1 Codebeispiel ON/OFF
  - 5.2 Codebeispiel Fernbedienung
  - 5.3 Lirc Installation
  - 5.4 IR-Receiver Test
  - 5.5 Fernbedienung anlernen
  - 5.6 Befehle senden mit dem Infrarot Transmitter

## Bild



## Technische Daten / Kurzbeschreibung

Eine Leuchtdiode, die im infraroten Bereich ausstrahlt. Je nach Eingangsspannung, werden Vorwiderstände benötigt

$V_f = 1,1V$

$I_f = 20mA$

emittierende Wellenlänge: 940nm (nicht sichtbares Licht)

### Vorwiderstände:

$R_f (3,3V) = 120\Omega$

[z.B. beim Einsatz mit ARM CPU-Kern basierten Mikrocontrollern wie Raspberry-Pi]

$R_f (5V) = 220\Omega$

Diese Webseite verwendet Cookies, um bestimmte Funktionen zu ermöglichen und das Angebot zu verbessern. **Weitere Informationen**

×



## Pin-Belegung



- \*Auf der Platine gibt es die Möglichkeit den jeweils benötigten Widerstand direkt aufzulöten. In dem Falle kann dann der mittlere Pin genutzt werden, der dann den Widerstand beinhaltet.

## Codebeispiel Arduino

### Codebeispiel ON/OFF

Diese Codebeispiel zeigt auf, wie eine LED mittels eines definierbaren Ausgangspins abwechselnd für Vier Sekunden ein- und danach zwei Sekunden ausgeschaltet werden kann.

```

1  int Led = 13;
2
3  void setup ()
4  {
5    pinMode (Led, OUTPUT); // Initialisierung Ausgangspin für die LED
6  }
7
8  void loop () //Hauptprogrammschleife
9  {
10   digitalWrite (Led, HIGH); // LED wird eingeschaltet
11   delay (4000); // Wartemodus für 4 Sekunden
12   digitalWrite (Led, LOW); // LED wird ausgeschaltet
13   delay (2000); // Wartemodus für weitere zwei Sekunden in denen die LED dann ausge
14 }

```

Beispielprogramm Download:

[LedTestArduino\\_4On\\_2Off.zip](#)

## Codebeispiel Fernbedienung

Mithilfe der beiden Sensormodule KY-005 und KY-022 lässt sich ein Infrarot-Fernbedienung + Infrarot Receiver System aufbauen. Hierzu werden neben den zwei Modulen auch zwei einzelne Arduinos benötigt. Der eine fungiert hierbei dann als Sender und der andere empfängt die Signale und gibt diese dann in der seriellen Konsole aus.

Für das folgende Codebeispiel wird eine zusätzliche Library benötigt:

- [Arduino-IRremote] von Ken Shirriff (<http://z3t0.github.io/Arduino-IRremote/>) | veröffentlicht unter LGPL

Die Library ist im Paket enthalten und muss vor dem Start der Arduino IDE in den "library"-Ordner kopiert werden.

Diesen finden Sie standardmäßig unter dem folgenden Pfad Ihrer Windows-Installation:

C:\Benutzer\[Benutzername]\Dokumente\Arduino\libraries

Bei Infrarot-Sendesystemen, gibt es verschiedene Protokolle, in denen die Daten versendet werden können. In dem folgenden Beispiel wird für das versenden das RC5 Protokoll verwendet - die verwendete Library "Arduino-IRremote" kümmert sich eigenständig um die Konvertierung in die richtige Datenfolge. Es gibt innerhalb der Library jedoch auch andere Protokolle/Kodierungen - diese sind in der Dokumentation/Code der Library gekennzeichnet.



```

3
4 // Hier kann der entsprechende Eingangspin für den Signalausgang
5 // des KY-022 deklariert werden
6 int RECV_PIN = 11;
7
8 // Arduino-IRremote Library wird initialisiert
9 IRrecv irrecv(RECV_PIN);
10 decode_results results;
11
12 void setup()
13 {
14   Serial.begin(9600);
15   irrecv.enableIRIn(); // Infrarot-Receiver wird gestartet
16 }
17
18 // Hauptprogrammschleife
19 void loop() {
20
21   // Es wird geprüft ob am Receiver ein Signal eingegangen ist
22   if (irrecv.decode(&results)) {
23     //Bei Signaleingang wird das empfangene und dekodierte Signal in der seriellen K
24     Serial.println(results.value, HEX);
25     irrecv.resume();
26   }
27 }

```

**Code für den Sender:**

```

1 //Arduino-IRremote Library wird hinzugefügt...
2 #include <IRremote.h>
3
4 //...und hier initialisiert
5 IRsend irsend;
6
7 // Die Einstellungen für den Ausgang werden von der Library übernommen
8 // Die entsprechenden Ausgänge unterscheiden sich je nach verwendeten Arduino
9 // Arduino UNO: Ausgang = D3
10 // Arduino MEGA: Ausgang = D9
11 // Eine komplette Auflistung der entsprechenden Ausgänge finden Sie unter
12 // http://z3t0.github.io/Arduino-IRremote/
13 void setup()
14 {
15 }
16
17 // Hauptprogrammschleife
18 void loop() {
19   // Der Sender verschickt in diesem Beispiel das Signal A90 (in hexdezimaler
20   // Dieses wird dreimal hintereinander gesendet und danach eine Pause für 5
21   for (int i = 0; i < 3; i++) {
22     irsend.sendRC5(0xA90, 12); // [0xA90] zu versendetes Signal | [12] Bit-Läng
23     delay(40);
24   }
25   delay(5000); //Zwischen den Sendeimpulsen gibt es eine Pause von 5 Sekunden
26 }

```

**Beispielprogramm Download:**

Arduino\_Fernbedienung.zip

**Anschlussbelegung Arduino 1 [Empfänger]:**

KY-022

Signal	=	[Pin 11]
+V	=	[Pin 5V]
GND	=	[Pin GND]

**Anschlussbelegung Arduino 2 [Sender]:**

KY-005

Signal	=	[Pin 3 (Arduino Uno)   Pin 9 (Arduino Mega)]
GND+Widerstand	=	[Pin GND*]
GND	=	[Pin GND]

- \*Nur wenn Vorwiderstand auf dem Modul verlötet wurde und nicht vor dem Modul geschaltet ist

**Codebeispiel Raspberry Pi**

Es sind zwei Anwendungsbeispiele für dieses Sensormodul hier vorstellt. Eines, welches die Infrarot-

Diese Webseite verwendet Cookies, um bestimmte Funktionen zu ermöglichen und das Angebot zu verbessern. **Weitere Informationen**

×



entweder Infrarot Receiver für Fernbedienungen zum Steuern von z.B. der Mediacenter-Software OpenElec oder als Infrarot-Transmitter zur Software-gesteuerten Fernbedienung programmiert werden kann.

## Codebeispiel ON/OFF

Programmierbeispiel in der Programmiersprache Python

```

1  # Benoetigte Module werden importiert und eingerichtet
2  import RPi.GPIO as GPIO
3  import time
4
5  GPIO.setmode(GPIO.BCM)
6
7  # Hier wird der Eingangs-Pin deklariert, an dem der Sensor angeschlossen ist. Zusaetz
8  LED_PIN = 24
9  GPIO.setup(LED_PIN, GPIO.OUT, initial= GPIO.LOW)
10
11 print "LED-Test [druecken Sie STRG+C, um den Test zu beenden]"
12
13 # Hauptprogrammschleife
14 try:
15     while True:
16         print("LED 4 Sekunden an")
17         GPIO.output(LED_PIN,GPIO.HIGH) #LED wird eingeschaltet
18         time.sleep(4) #Wartemodus fuer 4 Sekunden
19         print("LED 2 Sekunden aus")
20         GPIO.output(LED_PIN,GPIO.LOW) #LED wird ausgeschaltet
21         time.sleep(2) #Wartemodus fuer weitere zwei Sekunden, in denen die LE
22
23 # Aufraeumarbeiten nachdem das Programm beendet wurde
24 except KeyboardInterrupt:
25     GPIO.cleanup()

```

### Anschlussbelegung Raspberry Pi:

Sensor Signal	=	GPIO24	[Pin 18]
GND+Widerstand	=	GND*	[Pin 9]
Sensor GND	=	Masse	[Pin 6]

- \*Nur wenn Vorwiderstand auf dem Modul verlötet wurde und nicht vor dem Modul geschaltet ist

### Beispielprogramm Download

LedTest\_RPi\_4On\_2Off.zip

Zu starten mit dem Befehl:

```
1 | sudo python LedTest_RPi_40n_20ff.py
```

## Codebeispiel Fernbedienung

Der Raspberry Pi besitzt mit seiner fortschrittlichen Prozessorarchitektur den Vorteil gegenüber dem Arduino, dass dieser ein komplettes Linux-Betriebssystem betreiben kann. Mit Hilfe eines Infrarot-Receivers können somit nicht nur einfache Datensignale ausgetauscht, sondern es können auch komplette Software-Programme wie z.B. die Mediacenter Software OpenElec per Fernbedienung bedient werden.

Für die Einrichtung eines Infrarot-Steuerungssystem, bietet sich unter Linux bietet die Software "lirc" an (veröffentlicht unter der LGPL - Website (<http://www.lirc.org/>)). Im Folgenden zeigen wir auf, wie lirc eingerichtet, eine Fernbedienung angelernt werden kann und per Infrarotdiode die erlernten Signale per Infrarot versendet werden können (um z.B. aus dem Raspberry Pi eine per Software steuerbare Infrarot Fernbedienung zu machen).

Zu diesem Zwecke werden die Module KY-005 als Infrarot-Transmitter und KY-022 als Infrarot Receiver angewendet.

### Anschlussbelegung Raspberry Pi:

KY-005

Signal	=	GPIO17	[Pin 11]
GND+Widerstand	=	GND*	[Pin 9]
GND	=	GND	[Pin 6]



## KY-022

Signal	=	GPI18	[Pin 12]
+V	=	3,3V	[Pin 17]
GND	=	GND	[Pin 25]

## Lirc Installation

Als erstes öffnen wir auf dem Desktop ein Terminal oder verbinden wir uns per SSH mit dem Raspberry Pi. Dort geben Sie den folgenden Befehl ein, um lirc auf den Raspberry Pi zu installieren:

```
1 | sudo apt-get install lirc -y
```

[Hierzu muss der Raspberry Pi mit dem Internet verbunden sein]

Damit das lirc Modul direkt zum Start des Betriebssystems verfügbar ist, müssen folgende Zeilen am Ende der Datei "/boot/config.txt" hinzugefügt werden:

```
dtoverlay=lirc-rpi,gpio_in_pin=18,gpio_out_pin=17,gpio_in_pull=up
```

Hierbei definiert "gpio\_in\_pin=18" den Eingangspin für den IR-Receiver, sowie "gpio\_out\_pin=17" den Ausgangspin für den IR-Transmitter.

Die Datei kann mit folgenden Befehl editiert werden:

```
1 | sudo nano /boot/config.txt
```

Mit der Tastenfolge [Strg+X -> Y -> Enter] kann die Datei, nach dem Hinzufügen der Zeile am unteren Ende, gespeichert und geschlossen werden.

Auch die Datei "/etc/lirc/hardware.conf" muss modifiziert werden. Hierbei ist der Befehl...

```
1 | sudo nano /etc/lirc/hardware.conf
```

... zu verwenden. Hier müssen die folgenden Zeilen modifiziert werden. An den entsprechenden Stellen muss dann ...

```
DRIVER="UNCONFIGURED"
--->
DRIVER="default"
```

```
DEVICE=""
--->
DEVICE="/dev/lirc0"
```

```
MODULES=""
--->
MODULES="lirc-rpi"
```

...geändert werden.

Die modifizierte Datei muss dann wie folgt aussehen:

```
1 | # /etc/lirc/hardware.conf
2 | #
3 | # Arguments which will be used when launching lircd
4 | LIRC_ARGS=""
5 |
6 | #Don't start lircmd even if there seems to be a good config file
7 | #START_LIRCMD=false
8 |
9 | #Don't start irexec, even if a good config file seems to exist.
10 | #START_IEXEC=false
11 |
12 | #Try to load appropriate kernel modules
13 | LOAD_MODULES=true
14 |
15 | # Run "lircd --driver=help" for a list of supported drivers.
16 | DRIVER="default"
17 | # usually /dev/lirc0 is the correct setting for systems using udev
```



```

21 | # Default configuration files for your hardware if any
22 | LIRCD_CONF=""
23 | LIRCMD_CONF=""

```

Danach starten wir den Raspberry Pi mit dem folgenden Befehl neu:

```
1 | sudo reboot
```

## IR-Receiver Test

Um den angeschlossenen Receiver zu testen, muss vorab lirc mit dem Befehl...

```
1 | sudo /etc/init.d/lirc stop
```

beendet werden. Danach kann mit...

```
1 | mode2 -d /dev/lirc0
```

...getestet werden, ob am Raspberry Pi Signale detektiert werden können. Hierzu nehmen Sie eine Infrarot Fernbedienung und drücken eine beliebige Taste - es sollten Zeilen in der folgenden Form auftauchen:

```

1 | space 95253
2 | pulse 9022
3 | space 2210
4 | pulse 604
5 | space 95246
6 | pulse 9019
7 | space 2211
8 | pulse 601
9 | space 95252
10 | pulse 9019
11 | space 2210
12 | pulse 603
13 | space 95239
14 | pulse 9020
15 | space 2208
16 | pulse 603
17 | ...

```

Mit dem Befehl...

```
1 | sudo /etc/init.d/lirc start
```

... kann der lirc-Dienst wieder gestartet werden.

## Fernbedienung anlernen

Um eine Infrarot-Fernbedienung im lirc System zu registrieren, muss für die Fernbedienung die Datei "/etc/lirc/lircd.conf" konfiguriert werden. In dieser sind die jeweiligen Zuordnungen von Befehl zu empfangenen Infrarot-Codes gespeichert.

Um eine entsprechend richtig formatierte lircd.conf erstellen zu können, gibt es in der lirc-Software einen Assistenten, der die Datei automatisch erstellt. Die Vorgehensweise für diese Konfiguration ist wie folgt:

Zu aller erst, muss der lirc-Service beendet werden

```
1 | sudo /etc/init.d/lirc stop
```

Mit dem folgenden Befehl starten wir dann den Assistenten:

```
1 | irrecord -d /dev/lirc0 ~/MeineFernbedienung.conf
```

Dieser Assistent führt vorab eine erste Initialisierung der Fernbedienung durch - in dieser müssen mehrere Tasten abwechselnd gedrückt werden, damit das lirc System die entsprechende Kodierung der Fernbedienung erlernen kann. Bitte folgen Sie dazu die entsprechenden Anweisungen des Assistenten. Nach der Initialisierung fragt dann der Assistent nach dem Namen der Knopfzuordnung, die mit einem neuen Infrarotcode aufgezeichnet werden soll. Sie können sich hierzu die Knopfzuordnungen aus der folgenden Datei auswählen

FernbedienungsCodes.txt

Diese müssen dann in den Assistenten eingegeben und mit Enter bestätigt werden. Hiernach startet dann die Aufzeichnung des Infrarot-Codes, für die ausgewählte Taste.

Beispiel: [KEY\_0] eingeben -> mit Enter bestätigen -> Taste "0" auf der Fernbedienung drücken -> warten bis der Assistent die Aufnahme bestätigt.

Sollen keine weiteren Tasten angelernt werden, so kann der Assistent mit der Enter-Taste beendet werden.



```
1 | sudo nano ~/MeineFernbedienung.conf
```

Hier kann dann die Zeile 17 von

```
name /home/pi/MeineFernbedienung.conf
```

in

```
name MeineFernbedienung
```

geändert werden. Beachten Sie hierbei keine Leerzeichen und Sonderzeichen innerhalb des Namens zu verwenden. Mit der Tastenfolge [Strg+X -> Y -> Enter] kann die Datei, nach dem hinzufügen der Zeile am unteren Ende, gespeichert und geschlossen werden.

Nach erstellen der Konfigurationsdatei können Sie vorab für die original lircd.conf mit folgenden Befehl ein Backup erstellen:

```
1 | sudo mv /etc/lirc/lircd.conf /etc/lirc/lircd.conf.bak
```

und mit dem Befehl...

```
1 | sudo cp ~/MeineFernbedienung.conf /etc/lirc/lircd.conf
```

...wird die vorab neu erstellte Konfigurationsdatei für das lirc-System eingesetzt.

Nun kann mit...

```
1 | sudo /etc/init.d/lirc start
```

...das lirc System wieder gestartet werden.

Ab nun ist die neu angelegte Fernbedienung im System registriert und kann in kompatibler Software, wie z.B. dem Mediacenter OpenElec verwendet werden. Alternativ kann mit dem Befehl...

```
1 | irw
```

...die Zuordnungen und die Funktion der Fernbedienung getestet werden.

Will man die Fernbedienung nun z.B. in OpenElec verwenden, kann es je nach Konfiguration der Medienssoftware sein, dass lirc in den Optionen von OpenElec vorab noch aktiviert werden muss.

## Befehle senden mit dem Infrarot Transmitter

Möchte man mit dem Raspberry Pi nun Geräte wie z.B. den Fernseher per Infrarot steuern, so können nun die vorab angelegten Befehle mit Hilfe des Infrarot Transmitters wieder versendet werden. So lässt sich z.B. eine Software gesteuerte Infrarot-Steuerung aufbauen oder einzelne Geräte mittels Netzwerk/Internet ein und Ausschalten.

Zuerst überprüfen wir mit folgendem Befehl...

```
1 | irsend LIST MeineFernbedienung ""
```

...welche Zuordnungen für die jeweilige gespeicherte Fernbedienung verfügbar sind.

Nun können wir z.B. den Befehl [KEY\_0] versenden, indem wir den folgenden Befehl verwenden:

```
1 | irsend SEND_ONCE MeineFernbedienung KEY_0
```

Auf dem Fernseher bzw. dem entsprechenden Empfänger-Endgerät, sollte sich nun eine Reaktion zeigen. Zu dem o.g. Befehl gibt es auch die Variation, dass das Signal wiederholend ausgegeben wird

```
1 | irsend SEND_START MeineFernbedienung KEY_0
```

Hier nach wird der gesendete Code [KEY\_0] so oft wiederholt bis mit...

```
1 | irsend SEND_STOP MeineFernbedienung KEY_0
```

...wieder beendet wird. Dies wird z.B. bei Befehlen wie "Lautstärke Hoch" oder "Helligkeit runter" angewendet.

Von „[https://www.linkerkit.de/index.php?title=KY-005\\_Infrarot\\_Transmitter\\_Modul&oldid=571](https://www.linkerkit.de/index.php?title=KY-005_Infrarot_Transmitter_Modul&oldid=571)“





Diese Webseite verwendet Cookies, um bestimmte Funktionen zu ermöglichen und das Angebot zu verbessern. **Weitere Informationen**

