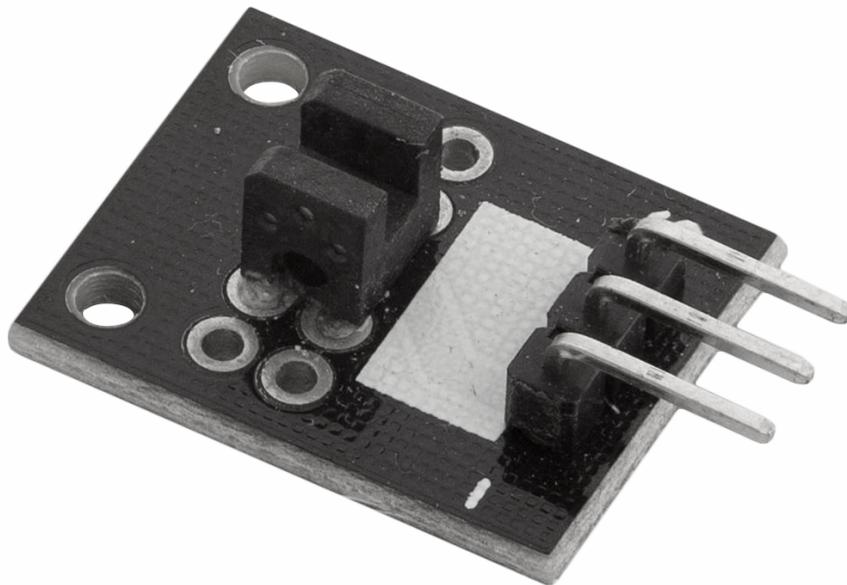


# AZ-Delivery

## Willkommen!

Vielen Dank, dass sie sich für unser *IKY-010 Lichtschrank-Modul* von *AZ-Delivery* entschieden haben. In den nachfolgenden Seiten werden wir Ihnen erklären wie Sie das Gerät einrichten und nutzen können.

**Viel Spaß!**



# Az-Delivery

## Inhaltsverzeichnis

Einführung.....	3
Technische Daten.....	4
Pinbelegung.....	4
Schematische Darstellung.....	5
Wie man die Arduino IDE einrichtet.....	7
Wie man den Raspberry Pi und Python einrichtet.....	11
Verbindung des Moduls mit dem Uno.....	12
Sketch-Beispiel.....	13
Verbindung des Moduls mit dem Raspberry Pi.....	14
Python-Skript.....	15

## Einführung

Das KY-010 light Lichtschranken-Modul besteht aus einem optischen Emitter (eine LED), einem Detektor (ein Fototransistor) und zwei Widerständen ( $10k\Omega$  und  $330\Omega$ ). Der Sensor verwendet einen Lichtstrahl (zwischen Emitter und Detektor), um festzustellen, ob der optische Weg zwischen ihnen durch ein nicht transparentes Objekt blockiert ist.

Das Modul besitzt drei Pins, wobei S der Signal-Pin, der mittlere Pin VCC (zwischen +3.3V und +5V) und – (Minus) ist der Masse-Pin oder 0V.

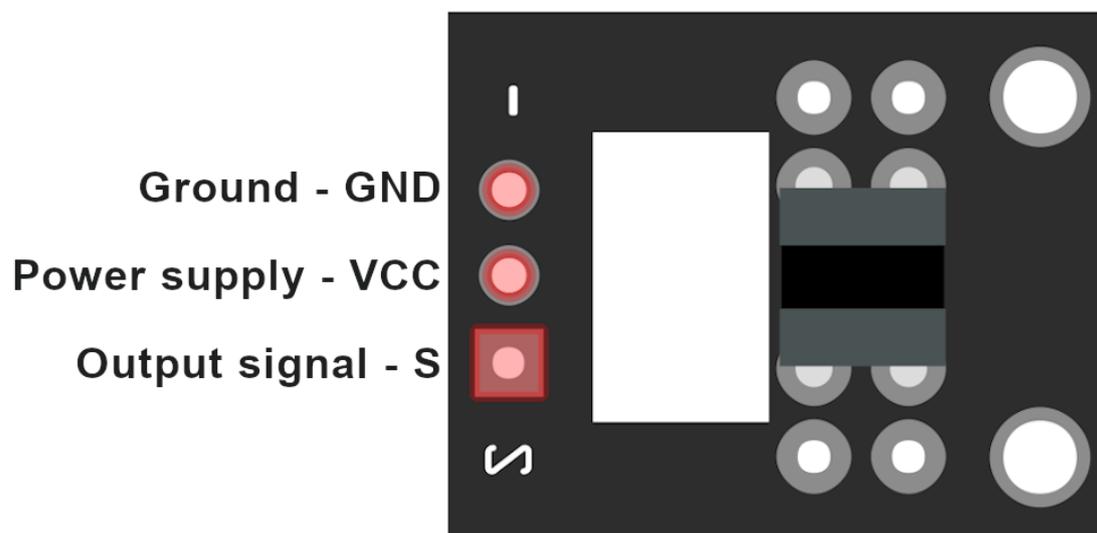
Wenn das KY-010 an die Stromversorgung angeschlossen wird, leuchtet die LED-Diode. Wenn keine Lichtschranke zwischen Emitter und Detektor ist, befindet sich der Modulausgang im Zustand HIGH. Wenn eine Schranke vorhanden ist, befinden sich die Ausgänge im Zustand LOW

## Technische Daten

- » Betriebsspannungsbereich: 3.3V bis 5V DC
- » Betriebstemperatur: -25°C bis 80°C [-13°F bis 176°F]
- » Dimensionen: 19 x 15mm [0.73 x 0.6in]

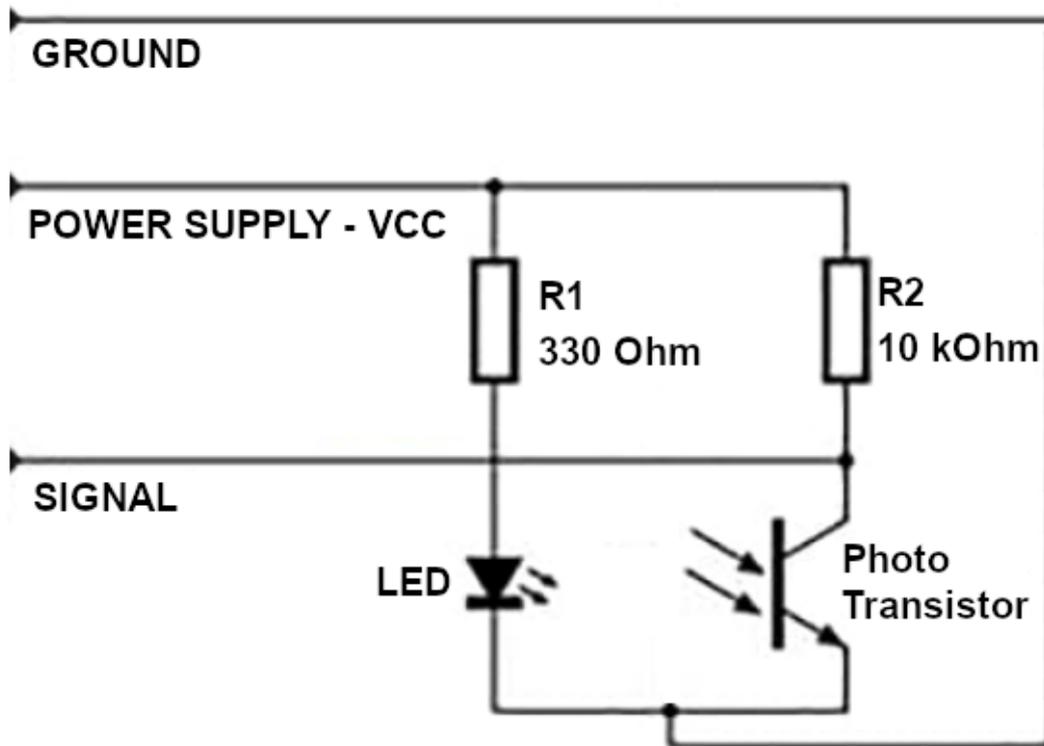
## Pinbelegung

Das KY-010 Lichtschranken-Modul besitzt drei Pins. Die Pinbelegung ist wie folgt:



## Schematische Darstellung

Die schematische Darstellung des KY-010-Moduls ist wie folgt:



# Az-Delivery

## Wie man die Arduino IDE einrichtet

Falls die Arduino-IDE nicht installiert ist, folgen Sie dem [link](#) und laden Sie die Installationsdatei für das Betriebssystem Ihrer Wahl herunter.

### Download the Arduino IDE



The screenshot shows the Arduino IDE download page. On the left, there is a teal circle containing the Arduino logo (an infinity symbol with a minus sign on the left and a plus sign on the right). To the right of the logo, the text reads: **ARDUINO 1.8.9**. Below this, it says: "The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software. This software can be used with any Arduino board. Refer to the [Getting Started](#) page for Installation instructions." On the right side of the page, there is a teal sidebar with the following options: **Windows** Installer, for Windows XP and up; **Windows** ZIP file for non admin install; **Windows app** Requires Win 8.1 or 10 with a "Get" button; **Mac OS X** 10.8 Mountain Lion or newer; **Linux** 32 bits; **Linux** 64 bits; **Linux ARM** 32 bits; **Linux ARM** 64 bits; Release Notes; Source Code; Checksums (sha512).

Für Windows Benutzer: Doppelklicken Sie auf die heruntergeladene .exe-Datei und folgen Sie den Anweisungen im Installationsfenster.

# Az-Delivery

Für *Linux* Benutzer, laden Sie eine Datei mit der Erweiterung *.tar.xz* herunter, die extrahiert werden muss. Wenn sie extrahiert ist, gehen Sie in das extrahierte Verzeichnis und öffnen Sie das Terminal in diesem Verzeichnis. Zwei *.sh* Skripte müssen ausgeführt werden, das erste namens *arduino-linux-setup.sh* und das zweite heißt *install.sh*.

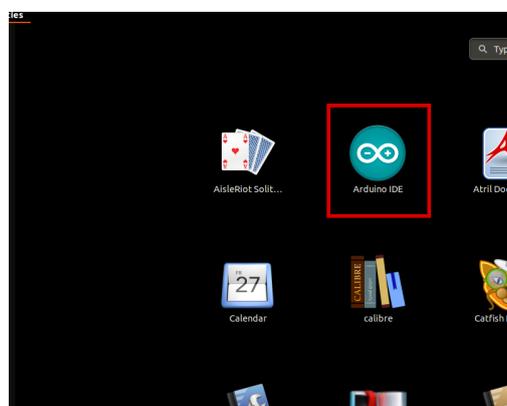
Um das erste Skript im Terminal auszuführen, öffnen Sie das Terminal im extrahierten Ordner und führen Sie den folgenden Befehl aus:

```
sh arduino-linux-setup.sh user_name
```

**user\_name** - ist der Name eines Superusers im Linux-Betriebssystem. Ein Passwort für den Superuser muss beim Start des Befehls eingegeben werden. Warten Sie einige Minuten, bis das Skript vollständig abgeschlossen ist.

Das zweite Skript mit der Bezeichnung *install.sh*-Skript muss nach der Installation des ersten Skripts verwendet werden. Führen Sie den folgenden Befehl im Terminal (extrahiertes Verzeichnis) aus: **sh install.sh**

Nach der Installation dieser Skripte gehen Sie zu *All Apps*, wo die *Arduino-IDE* installiert ist.



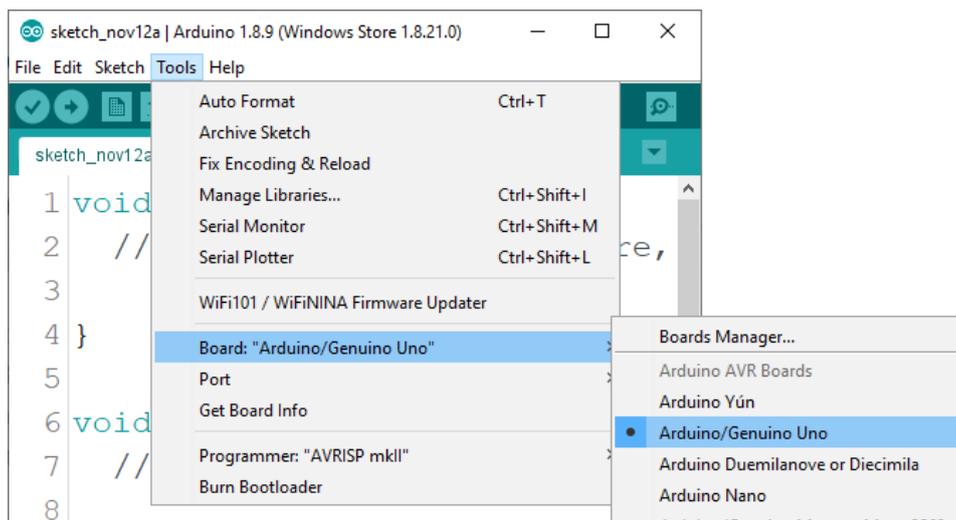
# Az-Delivery

Fast alle Betriebssysteme werden mit einem vorinstallierten Texteditor ausgeliefert (z.B. *Windows* mit *Notepad*, *Linux Ubuntu* mit *Gedit*, *Linux Raspbian* mit *Leafpad* usw.). Alle diese Texteditoren sind für den Zweck des eBooks vollkommen in Ordnung.

Zunächst ist zu prüfen, ob Ihr PC ein Arduino-Board erkennen kann. Öffnen Sie die frisch installierte Arduino-IDE, und gehen Sie zu:

*Tools > Board > {your board name here}*

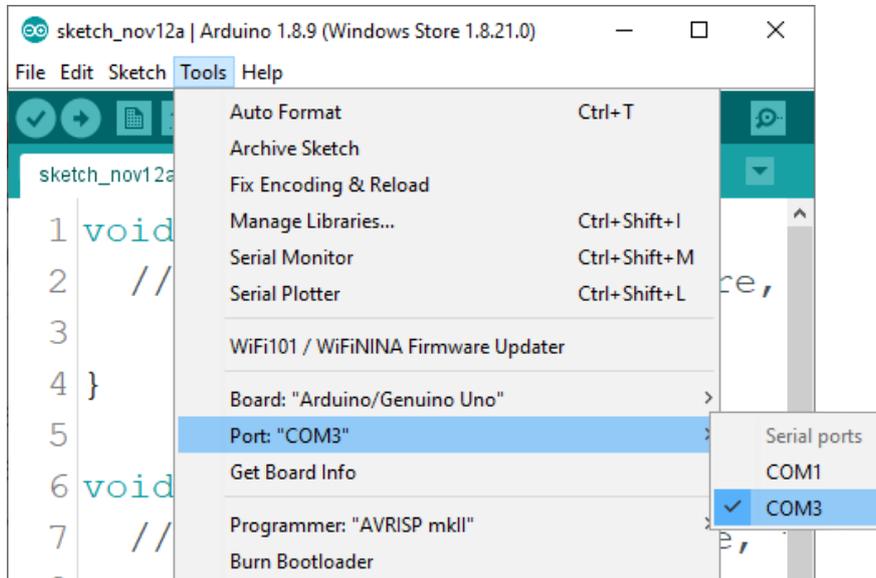
*{your board name here}* sollte der *Arduino/Genuino Uno* sein, wie es auf dem folgenden Bild zu sehen kann:



Der Port, an den das Arduino-Board angeschlossen ist, muss ausgewählt werden. Gehe zu: *Tools > Port > {port name goes here}* und wenn das Arduino-Board an den USB-Port angeschlossen ist, ist der Portname im Drop-down Menü auf dem vorherigen Bild zu sehen.

# Az-Delivery

Wenn die Arduino-IDE unter Windows verwendet wird, lauten die Portnamen wie folgt:



Für *Linux* Benutzer, ist zum Beispiel der Portname `/dev/ttyUSBx`, wobei *x* für eine ganze Zahl zwischen 0 und 9 steht.

# Az-Delivery

## Wie man den Raspberry Pi und Python einrichtet

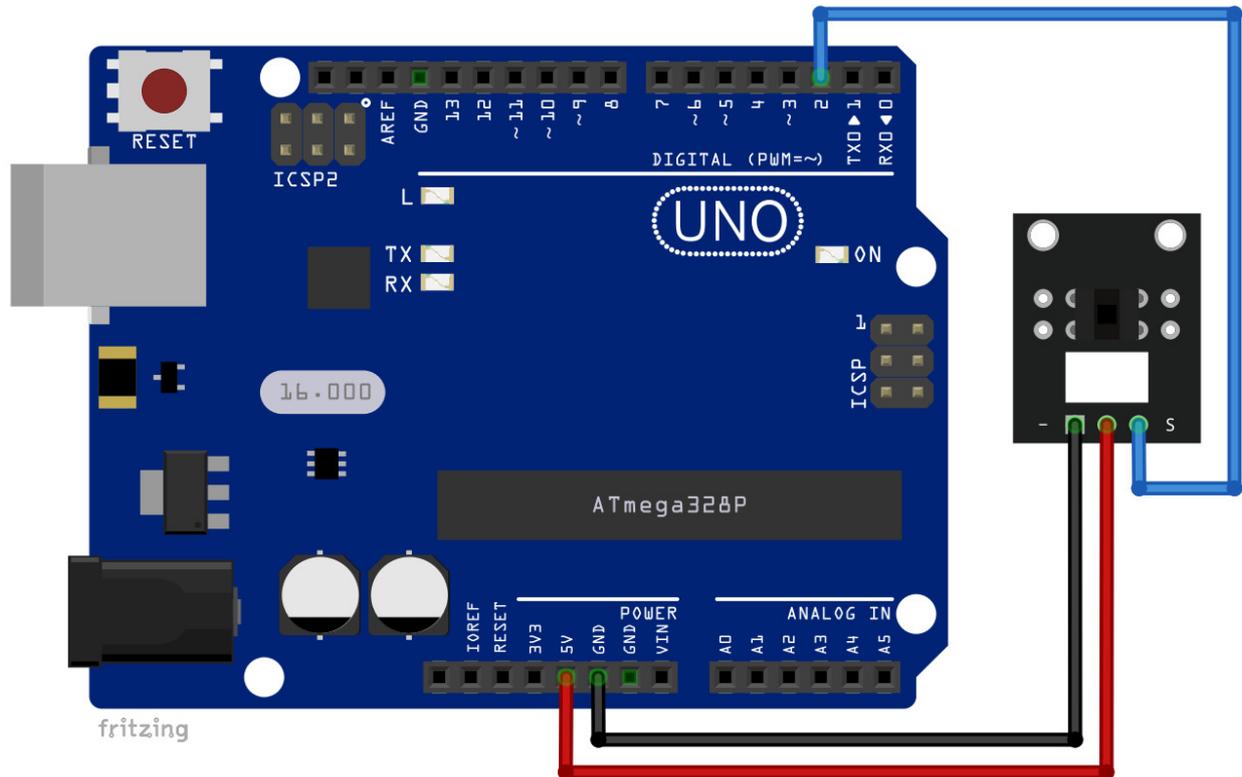
Für den Raspberry Pi muss zuerst das Betriebssystem installiert werden, dann muss alles so eingerichtet werden, dass es im *Headless*-Modus verwendet werden kann. Der *Headless*-Modus ermöglicht eine Fernverbindung zum Raspberry Pi, ohne dass ein PC-Bildschirm, eine Maus oder eine Tastatur erforderlich ist. Die einzigen Dinge, die in diesem Modus verwendet werden, sind der Raspberry Pi selbst, die Stromversorgung und die Internetverbindung. Das alles wird in dem kostenlosen eBook ausführlich erklärt:

[Raspberry Pi Quick Startup Guide](#)

Das Betriebssystem Raspbian wird mit vorinstalliertem *Python* ausgeliefert.

## Verbindung des Moduls mit dem Uno

Verbinden Sie das KY-010-Modul mit dem Uno, wie unten abgebildet:



KY-010 pin	>	Uno pin	
S	>	D2	Blauer Draht
- (GND)	>	GND	Schwarzer Draht
Middle pin (VCC)	>	5V	Roter Draht

# Az-Delivery

## Sketch-Beispiel

```
#define SIGNAL_PIN 2
uint8_t value = 0;

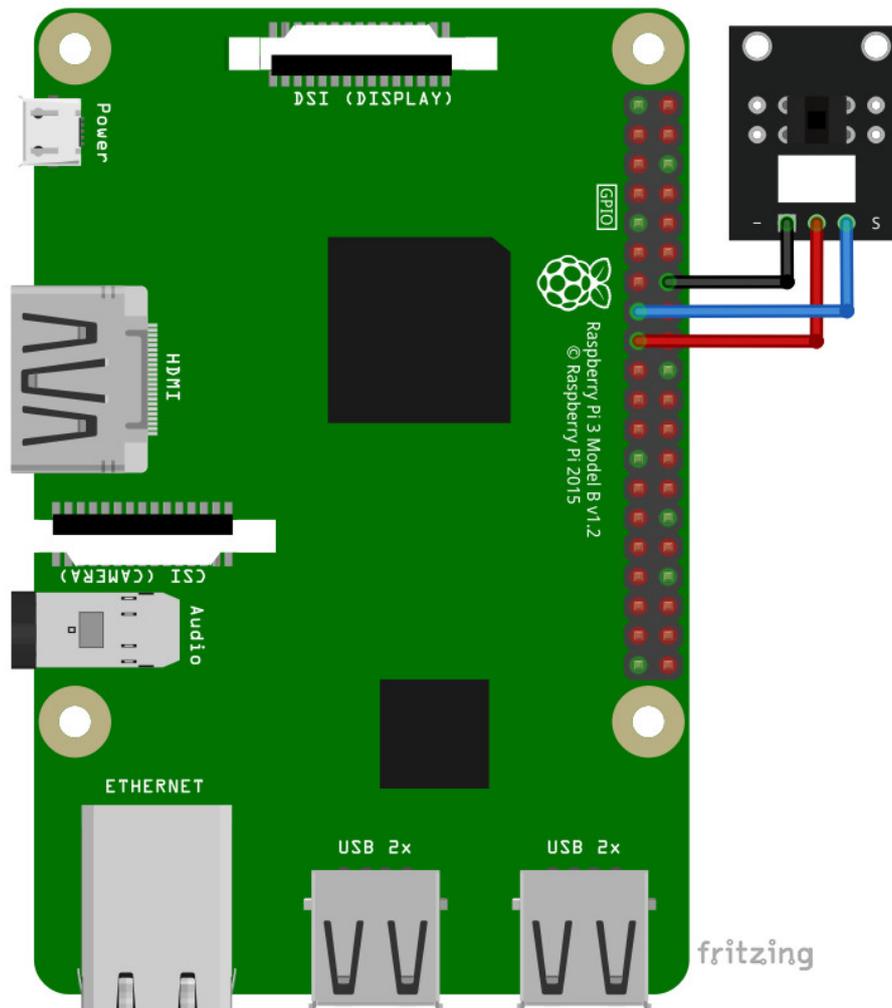
void setup() {
  Serial.begin(9600);
  pinMode(LED_BUILTIN, OUTPUT);
  pinMode(SIGNAL_PIN, INPUT);
}

void loop() {
  value = digitalRead(SIGNAL_PIN);
  Serial.println(value); // for debugging
  if(value == HIGH) {
    digitalWrite(LED_BUILTIN, HIGH);
  }
  else {
    digitalWrite(LED_BUILTIN, LOW);
  }
  delay(1000);
}
```

Wenn der Sketch auf den Uno geladen wird, leuchtet die integrierte LED, die mit dem digitalen Pin 13 verbunden ist, auf, sobald ein Hindernis zwischen Emitter und Detektor erkannt wird.

## Verbindung des Moduls mit dem Raspberry Pi

Verbinden Sie das KY-010-Modul mit dem Raspberry Pi, wie unten abgebildet:



KY-010 pin	>	Raspberry Pi pin	
- (GND)	>	GND [pin 14]	<b>Schwarzer Draht</b>
S	>	GPIO22 [pin 15]	<b>Blauer Draht</b>
Middle pin (VCC)	>	3V3 [pin 17]	<b>Roter Draht</b>

# Az-Delivery

## Python-Skript

```
import RPi.GPIO as GPIO
from time import sleep

GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)

# Pin setup for the module
Signal_PIN = 22
GPIO.setup(Signal_PIN, GPIO.IN)

print('[Press CTRL + C to end the script!]\n')
try: # Main program loop
    while True:
        state = GPIO.input(Signal_PIN)
        if state == 1:
            print('BARRIER DETECTED')

        else:
            print('No barrier')

        sleep(2)

# Scavenging work after the end of the program
except KeyboardInterrupt:
    print('\nScript end!')

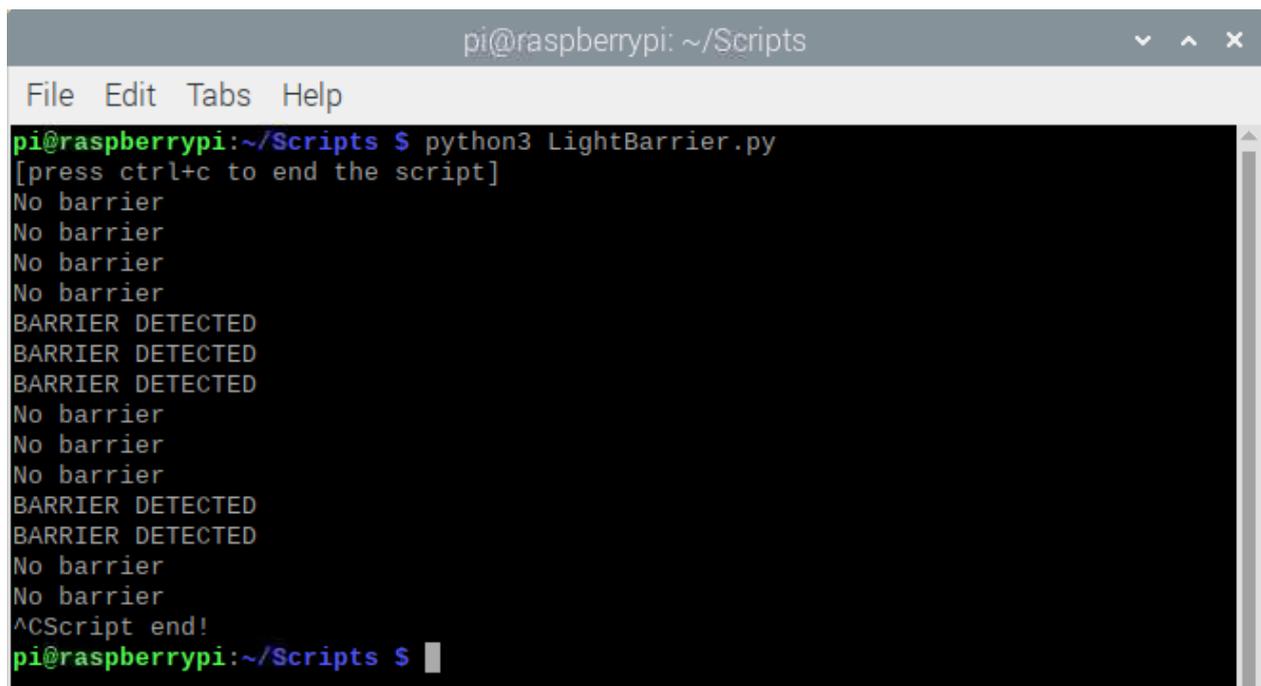
finally:
    GPIO.cleanup()
```

# Az-Delivery

Speichern Sie das Skript unter dem Namen *LightBarrier.py*. Um das Skript auszuführen, öffnen Sie das Terminal in dem Verzeichnis, in dem das Skript gespeichert ist, und führen Sie den folgenden Befehl aus:

```
python3 LightBarrier.py
```

Die Ausgabe sollte wie folgt aussehen:



```
pi@raspberrypi: ~/Scripts
File Edit Tabs Help
pi@raspberrypi:~/Scripts $ python3 LightBarrier.py
[press ctrl+c to end the script]
No barrier
No barrier
No barrier
No barrier
BARRIER DETECTED
BARRIER DETECTED
BARRIER DETECTED
No barrier
No barrier
No barrier
BARRIER DETECTED
BARRIER DETECTED
No barrier
No barrier
^CScript end!
pi@raspberrypi:~/Scripts $
```

Um das Skript zu beenden, drücken Sie “STRG + C” auf der Tastatur.

**Sie haben es geschafft. Sie können jetzt unser Modul für Ihre Projekte nutzen.**

# Az-Delivery

Jetzt sind Sie dran! Entwickeln Sie Ihre eigenen Projekte und Smart-Home Installationen. Wie Sie das bewerkstelligen können, zeigen wir Ihnen unkompliziert und verständlich auf unserem Blog. Dort bieten wir Ihnen Beispielskripte und Tutorials mit interessanten kleinen Projekten an, um schnell in die Welt der Mikroelektronik einzusteigen. Zusätzlich bietet Ihnen auch das Internet unzählige Möglichkeiten, um sich in Sachen Mikroelektronik weiterzubilden.

**Falls Sie nach weiteren hochwertigen Produkten für Arduino und Raspberry Pi suchen, sind Sie bei AZ-Delivery Vertriebs GmbH goldrichtig. Wir bieten Ihnen zahlreiche Anwendungsbeispiele, ausführliche Installationsanleitungen, E-Books, Bibliotheken und natürlich die Unterstützung unserer technischen Experten.**

<https://az-delivery.de>

Viel Spaß!

Impressum

<https://az-delivery.de/pages/about-us>