

## Willkommen!

Und herzlichen Dank für den Kauf unseres AZ-Delivery 8 Kanal Relais Moduls. Auf den folgenden Seiten gehen wir mit dir gemeinsam die ersten Programmierschritte durch.

Viel Spaß!



Mit den 8 Relais können größere Lasten (bis 10A) an einem Arduino, Raspberry Pi usw. betrieben werden.

**WARNUNG!! ES BESTEHT LEBENSGEFAHR DURCH EINEN ELEKTRISCHEN SCHLAG BEI BETRIEB ÜBER 30V ODER 230V NETZSPANNUNG. ACHTEN SIE AUF ENTSPRECHENDE ISOLIERUNG UND SCHUTZVORKEHRUNGEN.**

## Ansteuern des Relais:

Das Relais wird ganz einfach angesteuert, wird der Ausgangspegel auf **LOW** geschaltet, so zieht das Relais an und wird eingeschaltet.

Das Relais hat einen Wechsler-Ausgang mit einem Öffner (NC) und einen Schließer (NO). Je nachdem an welchen Ausgang die Last angeschlossen wird, kann die Last aus bzw. eingeschaltet werden.

## Verwendung der Relais an einem Arduino

### Verdrahten des Moduls mit einem Arduino Uno:

**VCC** wird mit **5V** am Arduino verbunden

**GND** wird mit **GND** verbunden

**IN1** wird mit **PIN 2** verbunden

**IN2** wird mit **PIN 3** verbunden

**IN3** wird mit **PIN 4** verbunden

**IN4** wird mit **PIN 5** verbunden

**IN5** wird mit **PIN 6** verbunden

**IN6** wird mit **PIN 7** verbunden

**IN7** wird mit **PIN 8** verbunden

**IN8** wird mit **PIN 9** verbunden

Rote Leitung

Schwarze Leitung

Grüne Leitung

Orange Leitung

Gelbe Leitung

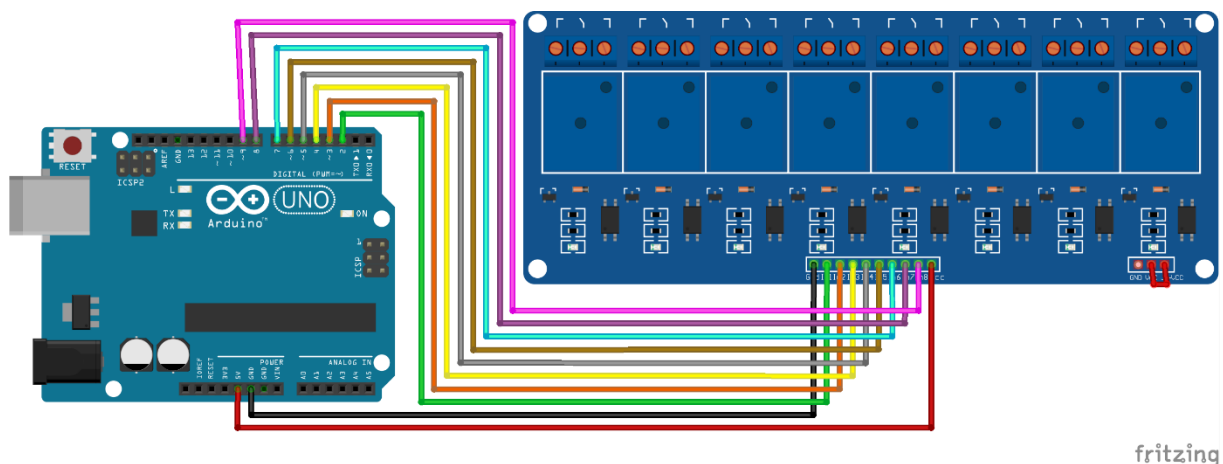
Graue Leitung

Braune Leitung

Hellblaue Leitung

Violette Leitung

Rosa Leitung



### Vorbereiten der Software:

Die Arduino Software sehen wir in diesem Schritt als Installiert an, sollte diese bei dir noch fehlen, so kannst du diese unter <https://www.arduino.cc/en/Main/Software#> herunterladen und auf deinen PC installieren.

## Der Code für einen Arduino:

```
const int RELAIS1 = 2;           //Arduino Pin 2
const int RELAIS2 = 3;           //Arduino Pin 3
const int RELAIS3 = 4;           //Arduino Pin 4
const int RELAIS4 = 5;           //Arduino Pin 5
const int RELAIS5 = 6;           //Arduino Pin 6
const int RELAIS6 = 7;           //Arduino Pin 7
const int RELAIS7 = 8;           //Arduino Pin 8
const int RELAIS8 = 9;           //Arduino Pin 9

void setup() {
  pinMode(RELAIS1, OUTPUT);
  pinMode(RELAIS2, OUTPUT);
  pinMode(RELAIS3, OUTPUT);
  pinMode(RELAIS4, OUTPUT);
  pinMode(RELAIS5, OUTPUT);
  pinMode(RELAIS6, OUTPUT);
  pinMode(RELAIS7, OUTPUT);
  pinMode(RELAIS8, OUTPUT);
}

void loop() {
  digitalWrite(RELAIS1, LOW);
  digitalWrite(RELAIS2, HIGH);
  digitalWrite(RELAIS3, HIGH);
  digitalWrite(RELAIS4, HIGH);
  digitalWrite(RELAIS5, HIGH);
  digitalWrite(RELAIS6, HIGH);
  digitalWrite(RELAIS7, HIGH);
  digitalWrite(RELAIS8, HIGH);
  delay(1000);


  digitalWrite(RELAIS1, HIGH);
  digitalWrite(RELAIS2, LOW);
  digitalWrite(RELAIS3, HIGH);
  digitalWrite(RELAIS4, HIGH);
  digitalWrite(RELAIS5, HIGH);
  digitalWrite(RELAIS6, HIGH);
  digitalWrite(RELAIS7, HIGH);
  digitalWrite(RELAIS8, HIGH);
  delay(1000);

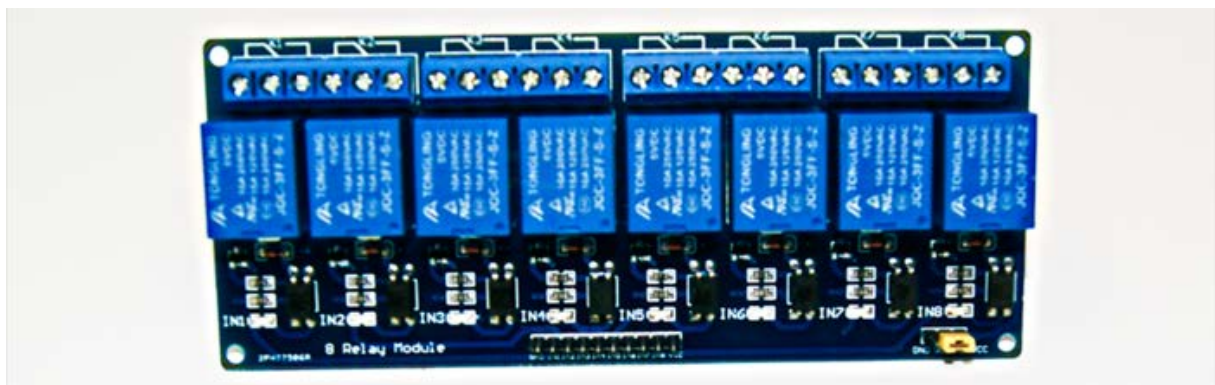
  digitalWrite(RELAIS1, HIGH);
  digitalWrite(RELAIS2, HIGH);
  digitalWrite(RELAIS3, LOW);
  digitalWrite(RELAIS4, HIGH);
  digitalWrite(RELAIS5, HIGH);
  digitalWrite(RELAIS6, HIGH);
  digitalWrite(RELAIS7, HIGH);
  digitalWrite(RELAIS8, HIGH);
  delay(1000);

  digitalWrite(RELAIS1, HIGH);
  digitalWrite(RELAIS2, HIGH);
  digitalWrite(RELAIS3, HIGH);
  digitalWrite(RELAIS4, LOW);
  digitalWrite(RELAIS5, HIGH);
  digitalWrite(RELAIS6, HIGH);
  digitalWrite(RELAIS7, HIGH);
  digitalWrite(RELAIS8, HIGH);
  delay(1000);
```

# Az-Delivery

```
digitalWrite(RELAIS1, HIGH);  
digitalWrite(RELAIS2, HIGH);  
digitalWrite(RELAIS3, HIGH);  
digitalWrite(RELAIS4, HIGH);  
digitalWrite(RELAIS5, LOW);  
digitalWrite(RELAIS6, HIGH);  
digitalWrite(RELAIS7, HIGH);  
digitalWrite(RELAIS8, HIGH);  
delay(1000);  
  
digitalWrite(RELAIS1, HIGH);  
digitalWrite(RELAIS2, HIGH);  
digitalWrite(RELAIS3, HIGH);  
digitalWrite(RELAIS4, HIGH);  
digitalWrite(RELAIS5, HIGH);  
digitalWrite(RELAIS6, LOW);  
digitalWrite(RELAIS7, HIGH);  
digitalWrite(RELAIS8, HIGH);  
delay(1000);  
  
digitalWrite(RELAIS1, HIGH);  
digitalWrite(RELAIS2, HIGH);  
digitalWrite(RELAIS3, HIGH);  
digitalWrite(RELAIS4, HIGH);  
digitalWrite(RELAIS5, HIGH);  
digitalWrite(RELAIS6, HIGH);  
digitalWrite(RELAIS7, LOW);  
digitalWrite(RELAIS8, HIGH);  
delay(1000);  
  
digitalWrite(RELAIS1, HIGH);  
digitalWrite(RELAIS2, HIGH);  
digitalWrite(RELAIS3, HIGH);  
digitalWrite(RELAIS4, HIGH);  
digitalWrite(RELAIS5, HIGH);  
digitalWrite(RELAIS6, HIGH);  
digitalWrite(RELAIS7, HIGH);  
digitalWrite(RELAIS8, LOW);  
delay(1000);  
}
```

Nach dem Übertragen  werden die Relais für 1 Sekunde nacheinander eingeschaltet und wieder ausgeschaltet.

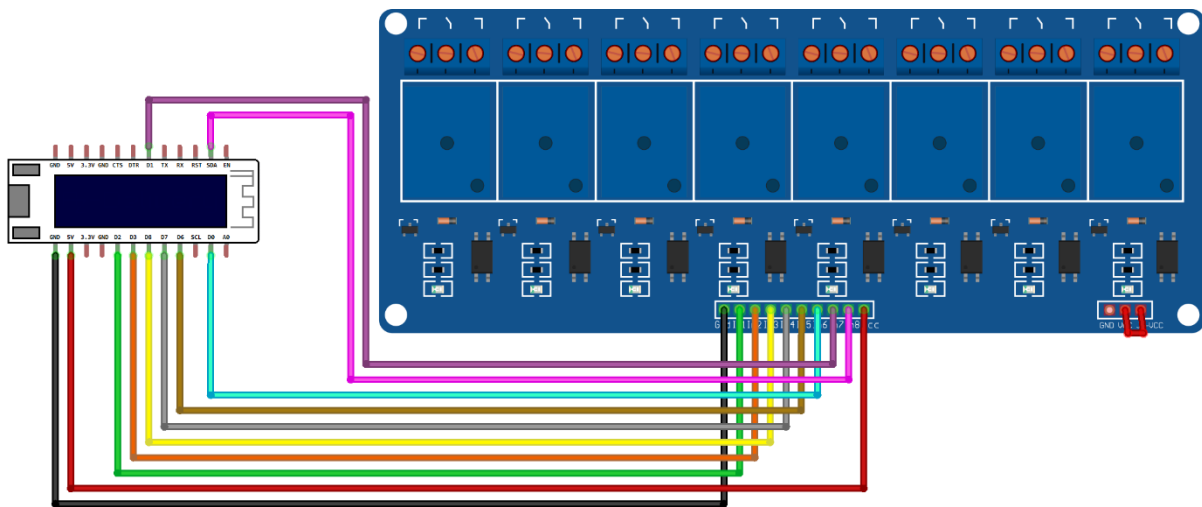


## Verwenden der Relais an einem ESP8266

### Verdrahten des Moduls mit einem ESP8266 (mit OLED):

**VCC** wird mit **5V** am ESP8266 verbunden  
**GND** wird mit **GND** verbunden  
**IN1** wird mit **PIN D2** verbunden  
**IN2** wird mit **PIN D3** verbunden  
**IN3** wird mit **PIN D8** verbunden  
**IN4** wird mit **PIN D7** verbunden  
**IN5** wird mit **PIN D6** verbunden  
**IN6** wird mit **PIN D0** verbunden  
**IN7** wird mit **PIN D1** verbunden  
**IN8** wird mit **PIN SDA** verbunden

Rote Leitung  
Schwarze Leitung  
Grüne Leitung  
Orange Leitung  
Gelbe Leitung  
Graue Leitung  
Braune Leitung  
Hellblaue Leitung  
Violette Leitung  
Rosa Leitung



fritzing

## Der Code für ESP8266:

```
const int RELAIS1 = D2;           //ESP GPIO Pin 2
const int RELAIS2 = D3;           //ESP GPIO Pin 3
const int RELAIS3 = D8;           //ESP GPIO Pin 8
const int RELAIS4 = D7;           //ESP GPIO Pin 7
const int RELAIS5 = D6;           //ESP GPIO Pin 6
const int RELAIS6 = D0;           //ESP GPIO Pin 0
const int RELAIS7 = D1;           //ESP GPIO Pin 1
const int RELAIS8 = D4;           //ESP GPIO Pin SDA

void setup() {
  pinMode(RELAIS1, OUTPUT);
  pinMode(RELAIS2, OUTPUT);
  pinMode(RELAIS3, OUTPUT);
  pinMode(RELAIS4, OUTPUT);
  pinMode(RELAIS5, OUTPUT);
  pinMode(RELAIS6, OUTPUT);
  pinMode(RELAIS7, OUTPUT);
  pinMode(RELAIS8, OUTPUT);
}

void loop() {
  digitalWrite(RELAIS1, LOW);
  digitalWrite(RELAIS2, HIGH);
  digitalWrite(RELAIS3, HIGH);
  digitalWrite(RELAIS4, HIGH);
  digitalWrite(RELAIS5, HIGH);
  digitalWrite(RELAIS6, HIGH);
  digitalWrite(RELAIS7, HIGH);
  digitalWrite(RELAIS8, HIGH);
  delay(1000);

  digitalWrite(RELAIS1, HIGH);
  digitalWrite(RELAIS2, LOW);
  digitalWrite(RELAIS3, HIGH);
  digitalWrite(RELAIS4, HIGH);
  digitalWrite(RELAIS5, HIGH);
  digitalWrite(RELAIS6, HIGH);
  digitalWrite(RELAIS7, HIGH);
  digitalWrite(RELAIS8, HIGH);
  delay(1000);

  digitalWrite(RELAIS1, HIGH);
  digitalWrite(RELAIS2, HIGH);
  digitalWrite(RELAIS3, LOW);
  digitalWrite(RELAIS4, HIGH);
  digitalWrite(RELAIS5, HIGH);
  digitalWrite(RELAIS6, HIGH);
  digitalWrite(RELAIS7, HIGH);
  digitalWrite(RELAIS8, HIGH);
  delay(1000);

  digitalWrite(RELAIS1, HIGH);
  digitalWrite(RELAIS2, HIGH);
  digitalWrite(RELAIS3, HIGH);
  digitalWrite(RELAIS4, LOW);
  digitalWrite(RELAIS5, HIGH);
  digitalWrite(RELAIS6, HIGH);
  digitalWrite(RELAIS7, HIGH);
}
```


# Az-Delivery

```
digitalWrite(RELAIS8, HIGH);
delay(1000);
digitalWrite(RELAIS1, HIGH);
digitalWrite(RELAIS2, HIGH);
digitalWrite(RELAIS3, HIGH);
digitalWrite(RELAIS4, HIGH);
digitalWrite(RELAIS5, LOW);
digitalWrite(RELAIS6, HIGH);
digitalWrite(RELAIS7, HIGH);
digitalWrite(RELAIS8, HIGH);
delay(1000);

digitalWrite(RELAIS1, HIGH);
digitalWrite(RELAIS2, HIGH);
digitalWrite(RELAIS3, HIGH);
digitalWrite(RELAIS4, HIGH);
digitalWrite(RELAIS5, HIGH);
digitalWrite(RELAIS6, LOW);
digitalWrite(RELAIS7, HIGH);
digitalWrite(RELAIS8, HIGH);
delay(1000);

digitalWrite(RELAIS1, HIGH);
digitalWrite(RELAIS2, HIGH);
digitalWrite(RELAIS3, HIGH);
digitalWrite(RELAIS4, HIGH);
digitalWrite(RELAIS5, HIGH);
digitalWrite(RELAIS6, HIGH);
digitalWrite(RELAIS7, LOW);
digitalWrite(RELAIS8, HIGH);
delay(1000);

digitalWrite(RELAIS1, HIGH);
digitalWrite(RELAIS2, HIGH);
digitalWrite(RELAIS3, HIGH);
digitalWrite(RELAIS4, HIGH);
digitalWrite(RELAIS5, HIGH);
digitalWrite(RELAIS6, HIGH);
digitalWrite(RELAIS7, HIGH);
digitalWrite(RELAIS8, LOW);
delay(1000);
}
```

Nach dem übertragen  werden die Relais für 1 Sekunde nacheinander eingeschaltet und wieder ausgeschaltet.

.

## Verwendung der Relais an einem Raspberry Pi

### Verdrahten des Moduls mit einem Raspberry Pi:

**VCC** wird mit **5V** am Raspberry verbunden

**GND** wird mit **GND** verbunden

**IN1** wird mit **GPIO 23** verbunden

**IN2** wird mit **GPIO 24** verbunden

**IN3** wird mit **GPIO 25** verbunden

**IN4** wird mit **GPIO 8** verbunden

**IN5** wird mit **GPIO 7** verbunden

**IN6** wird mit **GPIO 1** verbunden

**IN7** wird mit **GPIO 12** verbunden

**IN8** wird mit **GPIO 16** verbunden

Rote Leitung

Schwarze Leitung

Grüne Leitung

Orange Leitung

Gelbe Leitung

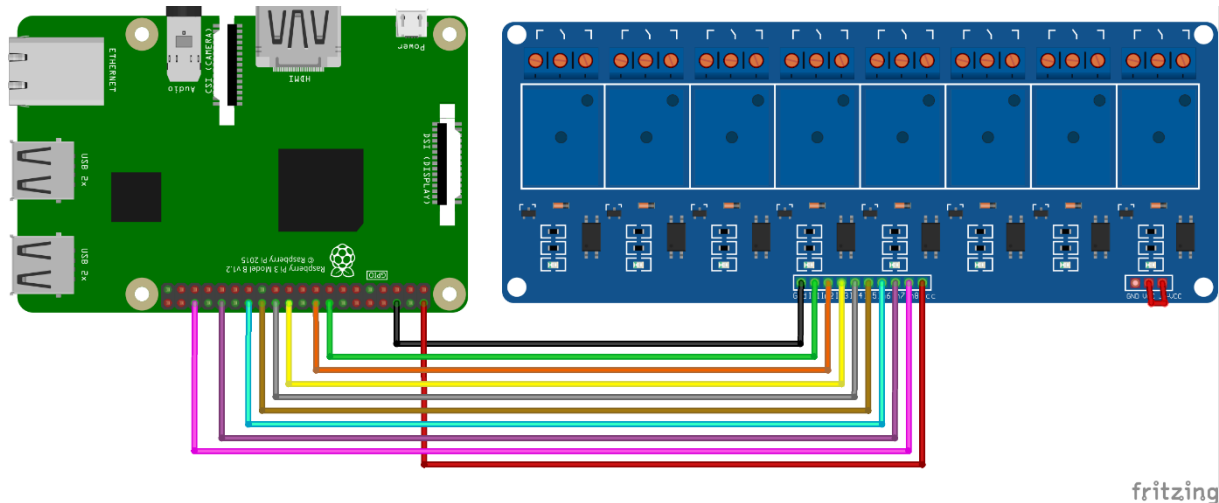
Graue Leitung

Braune Leitung

Hellblaue Leitung

Violette Leitung

Rosa Leitung



### Vorbereiten der Software:

Der Raspberry sollte entsprechend dem eBook für Raspberry Pi vorbereitet werden und aktualisiert werden.

Installation von wiringPi falls **gpio -v** keine Informationen ausgibt:

```
gpio -v
sudo apt-get purge wiringPi
sudo apt-get install git-core
git clone git://git.drogon.net/wiringPi
cd ~/wiringPi
git pull origin
cd ~/wiringPi
./build
sudo ./build
sudo reboot
```

Version prüfen  
Deinstallieren von alter Version  
Installieren von git  
wiringPi herunterladen  
in das Verzeichnis wechseln  
prüfen auf aktuelle Version  
in das Verzeichnis wechseln  
wiringPi Kompilieren  
wiringPi Installieren  
Raspberry Pi neustarten



Nun legen wir ein neues Programm an:

**touch relais.py**

Anlegen einer neuen Datei „relais.py“

**nano relais.py**

Datei im Editor öffnen

Dann fügen wir folgenden Code in den Editor ein:

```
#!/usr/bin/python
from time import sleep
import RPi.GPIO as GPIO          # Module einbinden
GPIO.setmode(GPIO.BCM)          # Pin Beschreibung auf BCM
GPIO.setup(23,GPIO.OUT)          # GPIO4 als Ausgang festlegen
GPIO.setup(24,GPIO.OUT)          # GPIO5 als Ausgang festlegen
GPIO.setup(25,GPIO.OUT)          # GPIO6 als Ausgang festlegen
GPIO.setup(8,GPIO.OUT)           # GPIO10 als Ausgang festlegen

GPIO.setup(7,GPIO.OUT)           # GPIO4 als Ausgang festlegen
GPIO.setup(1,GPIO.OUT)           # GPIO5 als Ausgang festlegen
GPIO.setup(12,GPIO.OUT)          # GPIO6 als Ausgang festlegen
GPIO.setup(16,GPIO.OUT)          # GPIO10 als Ausgang festlegen
while True:                      # Schleife generieren
    GPIO.output(23,GPIO.LOW)      # Relais 1 EIN
    GPIO.output(24,GPIO.HIGH)    # Relais 2 AUS
    GPIO.output(25,GPIO.HIGH)    # Relais 3 AUS
    GPIO.output(8,GPIO.HIGH)     # Relais 4 AUS
    GPIO.output(7,GPIO.HIGH)     # Relais 5 AUS
    GPIO.output(1,GPIO.HIGH)     # Relais 6 AUS
    GPIO.output(12,GPIO.HIGH)    # Relais 7 AUS
    GPIO.output(16,GPIO.HIGH)    # Relais 8 AUS

    sleep(1)                     # warte 1s
    GPIO.output(23,GPIO.HIGH)    # Relais 1 AUS
    GPIO.output(24,GPIO.LOW)     # Relais 2 EIN
    GPIO.output(25,GPIO.HIGH)    # Relais 3 AUS
    GPIO.output(8,GPIO.HIGH)     # Relais 4 AUS
    GPIO.output(7,GPIO.HIGH)     # Relais 5 AUS
    GPIO.output(1,GPIO.HIGH)     # Relais 6 AUS
    GPIO.output(12,GPIO.HIGH)    # Relais 7 AUS
    GPIO.output(16,GPIO.HIGH)    # Relais 8 AUS

    sleep(1)                     # warte 1s
    GPIO.output(23,GPIO.HIGH)    # Relais 1 AUS
    GPIO.output(24,GPIO.HIGH)    # Relais 2 AUS
    GPIO.output(25,GPIO.LOW)     # Relais 3 EIN
    GPIO.output(8,GPIO.HIGH)     # Relais 4 AUS
    GPIO.output(7,GPIO.HIGH)     # Relais 5 AUS
    GPIO.output(1,GPIO.HIGH)     # Relais 6 AUS
    GPIO.output(12,GPIO.HIGH)    # Relais 7 AUS
    GPIO.output(16,GPIO.HIGH)    # Relais 8 AUS
    sleep(1)                     # warte 1s
```

# Az-Delivery

```
GPIO.output(23,GPIO.HIGH) # Relais 1 AUS
GPIO.output(24,GPIO.HIGH) # Relais 2 AUS
GPIO.output(25,GPIO.HIGH) # Relais 3 AUS
GPIO.output(8,GPIO.LOW)   # Relais 4 EIN
GPIO.output(7,GPIO.HIGH)  # Relais 5 AUS
GPIO.output(1,GPIO.HIGH)  # Relais 6 AUS
GPIO.output(12,GPIO.HIGH) # Relais 7 AUS
GPIO.output(16,GPIO.HIGH) # Relais 8 AUS
sleep(1)                  # warte 1s
```

```
GPIO.output(23,GPIO.HIGH) # Relais 1 AUS
GPIO.output(24,GPIO.HIGH) # Relais 2 AUS
GPIO.output(25,GPIO.HIGH) # Relais 3 AUS
GPIO.output(8,GPIO.HIGH)  # Relais 4 AUS
GPIO.output(7,GPIO.LOW)   # Relais 5 EIN
GPIO.output(1,GPIO.HIGH)  # Relais 6 AUS
GPIO.output(12,GPIO.HIGH) # Relais 7 AUS
GPIO.output(16,GPIO.HIGH) # Relais 8 AUS
sleep(1)                  # warte 1s
```

```
GPIO.output(23,GPIO.HIGH) # Relais 1 AUS
GPIO.output(24,GPIO.HIGH) # Relais 2 AUS
GPIO.output(25,GPIO.HIGH) # Relais 3 AUS
GPIO.output(8,GPIO.HIGH)  # Relais 4 AUS
GPIO.output(7,GPIO.HIGH)  # Relais 5 AUS
GPIO.output(1,GPIO.LOW)   # Relais 6 EIN
GPIO.output(12,GPIO.HIGH) # Relais 7 AUS
GPIO.output(16,GPIO.HIGH) # Relais 8 AUS
sleep(1)                  # warte 1s
```

```
GPIO.output(23,GPIO.HIGH) # Relais 1 AUS
GPIO.output(24,GPIO.HIGH) # Relais 2 AUS
GPIO.output(25,GPIO.HIGH) # Relais 3 AUS
GPIO.output(8,GPIO.HIGH)  # Relais 4 AUS
GPIO.output(7,GPIO.HIGH)  # Relais 5 AUS
GPIO.output(1,GPIO.HIGH)  # Relais 6 AUS
GPIO.output(12,GPIO.LOW)  # Relais 7 EIN
GPIO.output(16,GPIO.HIGH) # Relais 8 AUS
sleep(1)                  # warte 1s
```

```
GPIO.output(23,GPIO.HIGH) # Relais 1 AUS
GPIO.output(24,GPIO.HIGH) # Relais 2 AUS
GPIO.output(25,GPIO.HIGH) # Relais 3 AUS
GPIO.output(8,GPIO.HIGH)  # Relais 4 AUS
GPIO.output(7,GPIO.HIGH)  # Relais 5 AUS
GPIO.output(1,GPIO.HIGH)  # Relais 6 AUS
GPIO.output(12,GPIO.HIGH) # Relais 7 AUS
GPIO.output(16,GPIO.LOW)  # Relais 8 EIN
sleep(1)                  # warte 1s
```

Das Programm führen mit dem Befehl

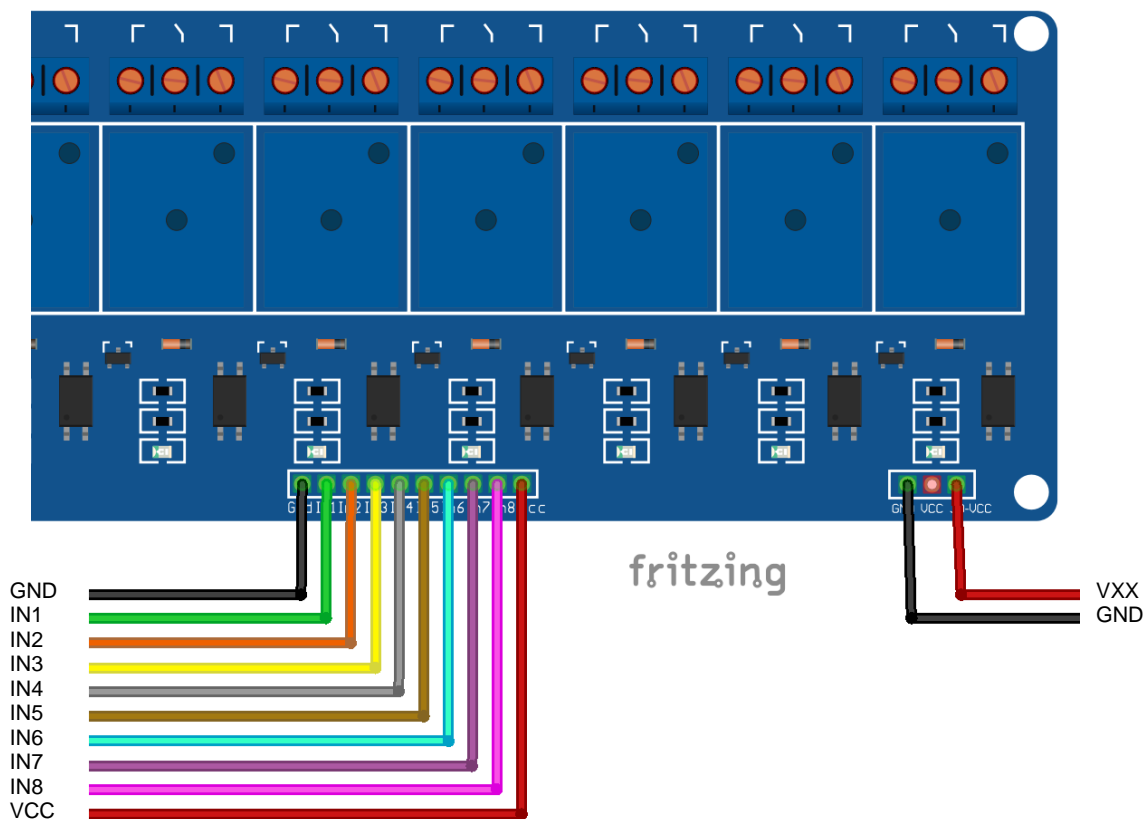
```
python relais.py
```

aus. Die Relais für 1 Sekunde nacheinander eingeschaltet und wieder ausgeschaltet, solange bis wir das Programm mit STRG+C beenden.

**Du hast es geschafft, du kannst nun in deinen Projekten ein Relais verwenden und größere Verbraucher schalten!**

## Externe Stromversorgung für das Modul:

Sollte die Stromversorgung über einen Raspberry Pi, Arduino, ESP usw. nicht ausreichend sein, so auch eine externe Stromversorgung von 5V verwendet werden.



**Arduino / Raspberry Pi / ESP**

**Externe Stromversorgung  
mit 5V**

Es besteht keine Galvanische Trennung zwischen dem Raspberry Pi, Arduino, ESP oder ähnlichem. Die Masse (GND) ist verbunden. JD-VCC und VCC dürfen nicht miteinander verbunden werden und der Jumper muss entfernt werden.

Ein Relais benötigt 0.45W, was einen Strombedarf von 0,09A bedeutet. Das Relaismodul benötigt bei beiden eingeschalteten Relais somit 0,72A. Das ist für manche µC-Module schon zu viel!



Ab jetzt heißt es Experimentieren.

Und für mehr Hardware sorgt natürlich dein Online-Shop auf:

<https://az-delivery.de>

Viel Spaß!

Impressum

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