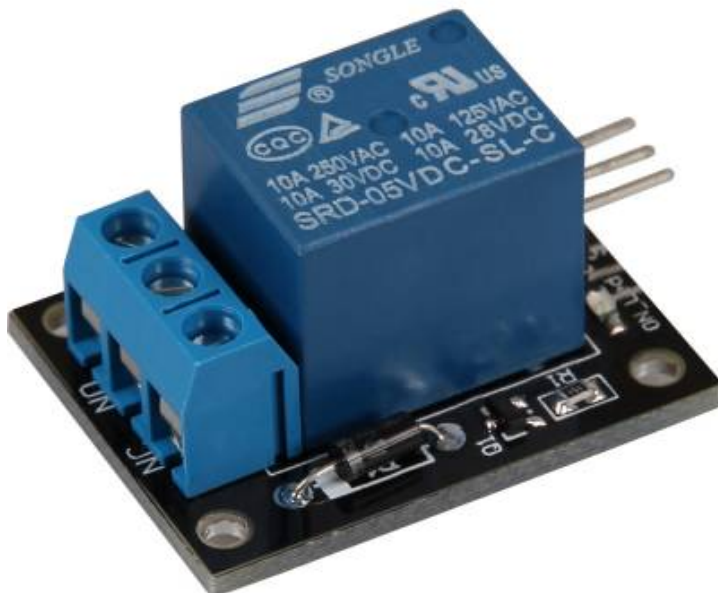


## KY-019 5V Relais Modul

### Inhaltsverzeichnis

1 Bild .....	1
2 Technische Daten / Kurzbeschreibung .....	1
3 Pin-Belegung .....	2
4 Codebeispiel Arduino .....	2
5 Codebeispiel Raspberry Pi .....	3

### Bild



### Technische Daten / Kurzbeschreibung

Spannungsbereich: 240VAC / 10A | 28VDC / 10A Ein Relais zum schalten von höherer Spannungen mittels eines 5V Ausgangs.

**!!!! Warnung !!!!**

**Das Arbeiten mit Spannungen >30v und vor allem bei Netzspannung (230V) kann zu körperlichen Schäden führen bis sogar tödlich sein.**

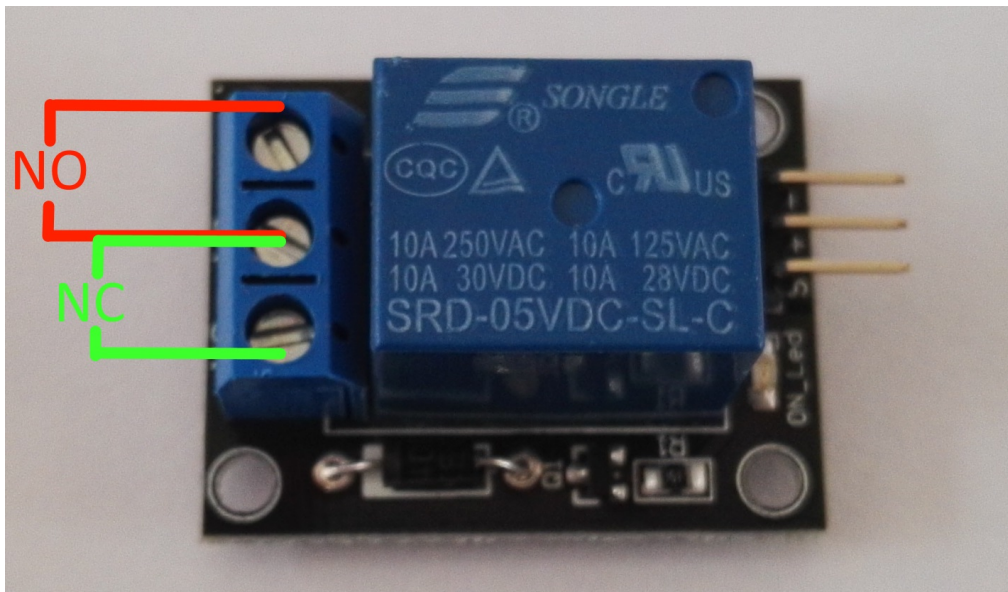
**Wir raten dringend dazu erst mit höheren Spannungen zu Arbeiten, wenn Sie die fachliche Kompetenz dazu besitzen.**

**!!!! Warnung !!!!**

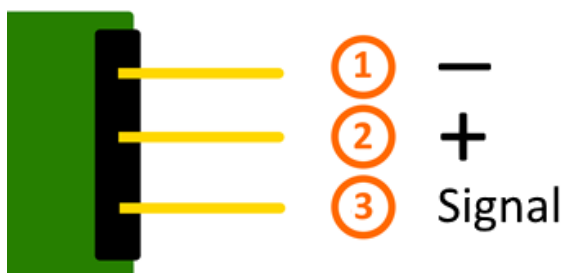
## KY-019 5V Relais Modul

Die Ausgangsleiste des Relais besitzt zwei Ausgangsterminals:

- Das eine welches mit "NC" für "normally closed" gekennzeichnet ist, was bedeutet dass dieser Durchgang ohne elektrische Umschaltung am Relais standardmäßig kurzgeschlossen ist.
- Das andere welches mit "NO" für "normally open" gekennzeichnet ist, was bedeutet dass dieser Durchgang ohne elektrische Umschaltung am Relais standardmäßig offen bzw. getrennt ist.



## Pin-Belegung



## Codebeispiel Arduino

Das Programm bildet einen Blinker nach - es schaltet das Relais in vorher definierter Zeit (delayTime) zwischen den beiden Zuständen (bzw. Ausgangsterminals) um.

## KY-019 5V Relais Modul

```
int relay = 10; // Hier wird der Pin deklariert, an dem das Relay angeschlossen ist
int delayTime = 1; // Wert in Sekunden, wie lange zwischen den Umschaltungen gewartet wird
void setup ()
{
  pinMode (relay, OUTPUT); // Der Pin wird als Ausgang deklariert
}

// Das Programm bildet einen Blinker nach - es schaltet das Relay in vorher definierter
// Zeit (delayTime) zwischen den beiden Zuständen (bzw. Ausgangsterminals) um.
void loop ()
{
  digitalWrite (relay, HIGH); // "NO" ist nun kurzgeschlossen;
  delay (delayTime * 1000);
  digitalWrite (relay, LOW); // "NC" ist nun kurzgeschlossen;
  delay (delayTime * 1000);
}
```

### Anschlussbelegung Arduino:

Sensor -	= [Pin GND]
Sensor +	= [Pin 5V]
Sensor Signal	= [Pin 10]

### Beispielprogramm Download

[KY-019\\_Relais.zip](#)

## Codebeispiel Raspberry Pi

Das Programm bildet einen Blinker nach - es schaltet das Relay in vorher definierter Zeit (delayTime) zwischen den beiden Zuständen (bzw. Ausgangsterminals) um.

```
# Benötigte Module werden importiert und eingerichtet
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BCM)
# Hier wird die Pause (in Sekunden) zwischen dem Umschalten deklariert
delayTime = 1

# Hier wird der Eingangs-Pin deklariert, an dem der Sensor angeschlossen ist. Zusätzlich
RELAIS_PIN = 24
GPIO.setup(RELAIS_PIN, GPIO.OUT)
GPIO.output(RELAIS_PIN, False)

print ("Sensor-Test [druecken Sie STRG+C, um den Test zu beenden]")

# Hauptprogrammsschleife
try:
  while True:
    GPIO.output(RELAIS_PIN, True) # NO ist nun kurzgeschlossen
    time.sleep(delayTime)
    GPIO.output(RELAIS_PIN, False) # NC ist nun kurzgeschlossen
    time.sleep(delayTime)
```

## KY-019 5V Relais Modul

```
# Aufräumarbeiten nachdem das Programm beendet wurde
except KeyboardInterrupt:
    GPIO.cleanup()
```

### Anschlussbelegung Raspberry Pi:

Relais -	=	GND	[Pin 06]
Relais +	=	5V	[Pin 2]
Relais Signal	=	GPIO24	[Pin 18]

### Beispielprogramm Download

[KY-019\\_RPi\\_Relais.zip](#)

Zu starten mit dem Befehl:

```
sudo python KY-019_RPi_Relais.py
```