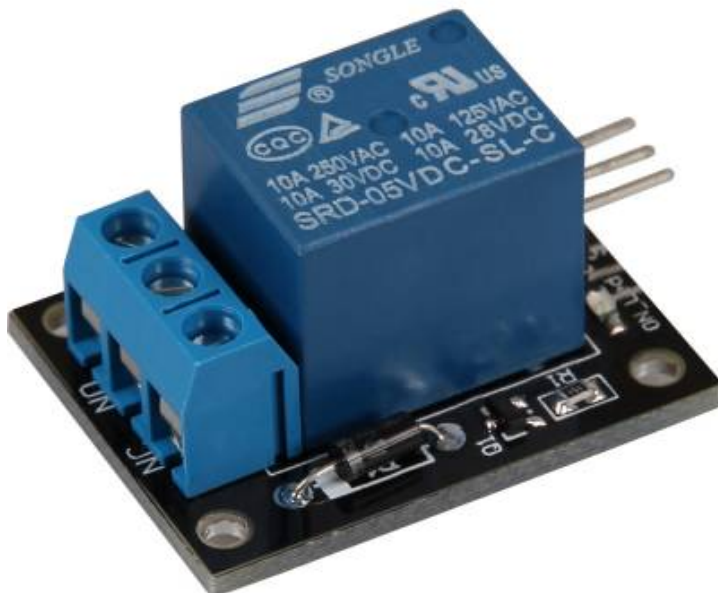


## KY-019 5V Relais module

### Contents

1 Picture .....	1
2 Technical data / Short description .....	1
3 Pinout .....	2
4 Code example Arduino .....	2
5 Code example Raspberry Pi .....	3

## Picture



## Technical data / Short description

Voltage range: 240VAC / 10A | 28VDC / 10A A relay to switch higher voltages via 5V output.

**!!!! Caution !!!!**

**Working with voltages over 30V and a main voltage (230V) can harm your body or kill you. We advise you not to work with higher voltages unless you have the needed experience.**

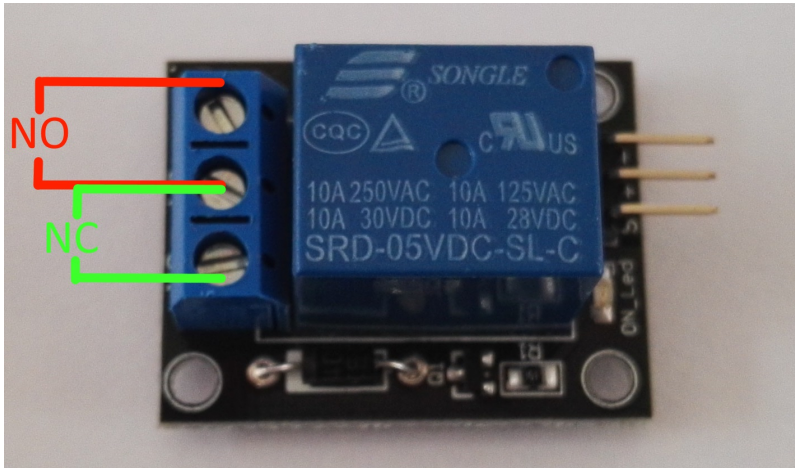
**!!!! Caution !!!!**

The output bar of the Relais has two output terminals.

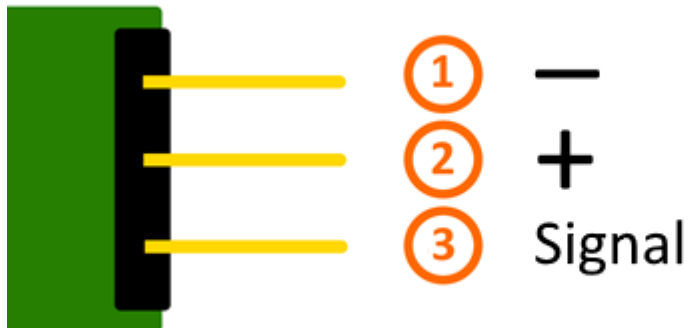
## KY-019 5V Relais module

- The first one is tagged with NC for "normally closed" which means that it's connected through by default.
- The second one is tagged with NO for "normally open" which means it's not connected through by default.

To switch both, you need a signal.



## Pinout



## Code example Arduino

The program imitates a direction indicator - it switches the status of the output terminals in a specific time period (delayTime).

```
int relay = 10; // Declaration of the pin which is connected with the relay
delayTime = 1 // The time which will be waited between the switches of the relay.

void setup ()
{
  pinMode (relay, OUTPUT); // Declaration of the pin to output
}
```

## KY-019 5V Relais module

```
// The program imitates a direction indicator - it switches the status of the output terminal
void loop ()
{
  digitalWrite (relay, HIGH); // "NO" is now connected through
  delay (delayTime * 1000);
  digitalWrite (relay, LOW); // "NC" is now connected through
  delay (delayTime * 1000);
}
```

### Connections Arduino:

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

### Example program download

[KY-019\\_Relais](#)

## Code example Raspberry Pi

The program imitates a direction indicator - it switches the status of the output terminals in a specific time period.

```
# Needed modules will be imported and configured
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BCM)
# Declaration of the break between the changes of the relay status (in seconds)
delayTime = 1

# Declaration of the input pin which is connected with the sensor. Additional to that, the
RELAIS_PIN = 21
GPIO.setup(RELAIS_PIN, GPIO.OUT)
GPIO.output(RELAIS_PIN, False)

print "Sensor-test [press ctrl+c to end]"

# Main program loop
try:
    while True:
        GPIO.output(RELAIS_PIN, True) # NO is now connected through
        time.sleep(delayTime)
        GPIO.output(RELAIS_PIN, False) # NC is now connected through
        time.sleep(delayTime)

# Scavenging work after the end of the program
except KeyboardInterrupt:
    GPIO.cleanup()
```

### Connections Raspberry Pi:

Relais -           = GND           [Pin 06]  
 Relais +           = 5V            [Pin 2]  
 Relais Signal     = GPIO21       [Pin 40]

KY-019 5V Relais module

**Example program download**

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

To start, enter the command:

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