Welcome!

Thank you for purchasing our *AZ-Delivery KY-019 Relay Module.* On the following pages, you will be introduced to how to use and set up this handy device.

Have fun!

Table of Contents

Introduction

The KY-019 relay module is used to control AC circuits, switching them *ON* or *OFF*. The relay is one of the most important control elements. It is an electrical switch that responds to a signal received from the microcontroller (like in Arduino or Raspberry Pi). Relays are widely used in the remote control, communications, mechatronics devices, power electronic devices, etc. They can also be used to separate power voltage/current electronics (like AC or DC motors, or any AC device, etc.) from low voltage/current electronics (like microcontrollers, sensors, etc.)

The KY-019 module consists of an LED, a resistor, an NPN transistor, a rectifier diode and a *5V DC* relay capable of handling up to *10A 250V AC*.

On the DC side of the board, there are three pins, one for the signal, one for the power supply (*VCC*) and one for the ground (*GND*). On the AC side there are three contacts Normally Closed - *NC* pin, Common pin and Normally Open - *NO* pin.

Specifications

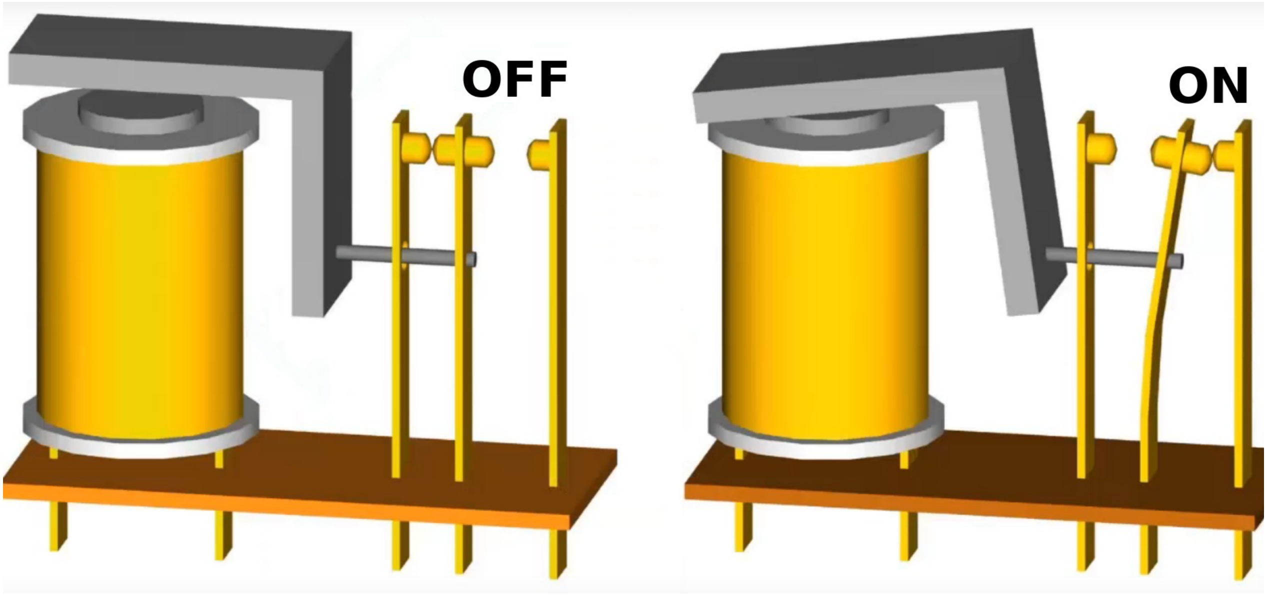
»	TTL control signal:	from 3.3V to 5V DC
»	Max. relay AC:	10A 250V
»	Max. relay DC:	10A 30V
»	Contact type:	Normally Closed – NC, Normally Opened – NO
»	Dimensions:	27 x 34mm [1.1 x 1.4in]

SAFETY WARNING!

When doing projects that are connected to mains voltage, misusage may lead to serious electrical shock!

For the sake of your own safety, be 100% sure what you are doing! Otherwise, ask someone who knows!

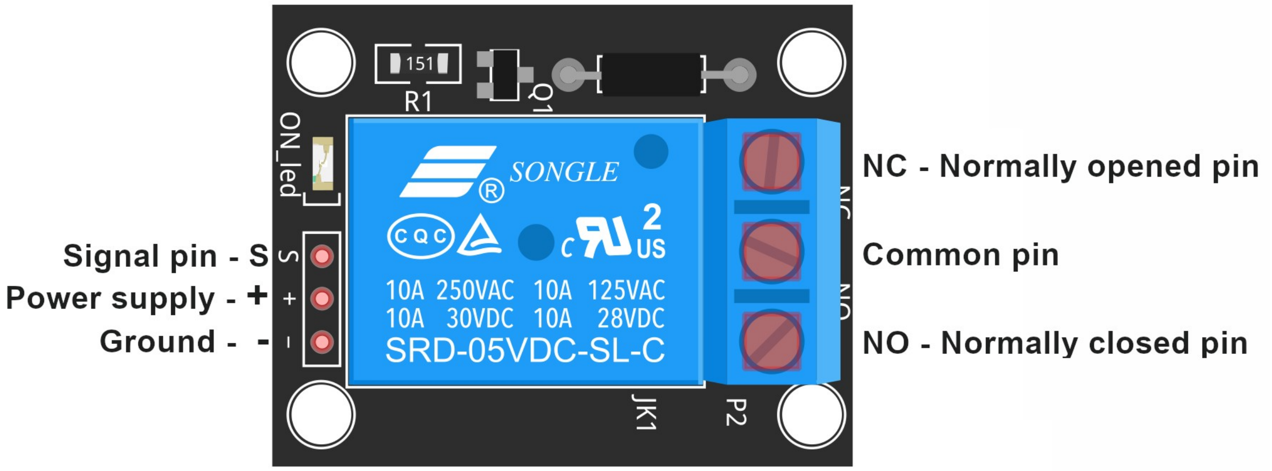
Inside the relay there are one mechanical switch (three lines with middle that moves) which is controlled by the second element, the electromagnet (big cyllinder shaped object on the image), as shown on the following image:



In the rest state, the switch is in the *OFF* state, *NC* pin is connected with a common pin, and *NO* is disconnected. When the power is connected to the electromagnet (using a microcontroller via transistor and rectifier diode), this moves the switch to the active state, thus connecting the common pin to NO pin.

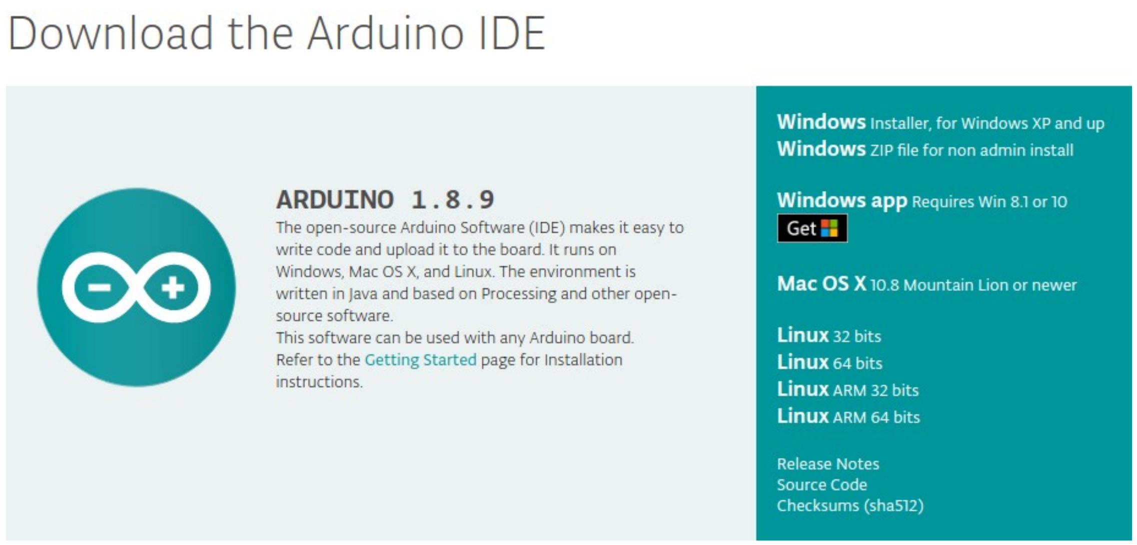
The pinout

The KY-019 relay module has six pins. The pinout diagram is shown on the following image:



How to set-up Arduino IDE

If the Arduino IDE is not installed, follow the *link* and download the installation file for the operating system of choice.



Download the Arduino IDE

ARDUINO 1.8.9

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.

Windows Installer, for Windows XP and up
Windows ZIP file for non admin install

Windows app Requires Win 8.1 or 10
Get

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)

For `Windows` users, double click on the downloaded `.exe` file and follow the instructions in the installation window.

For *Linux* users, download a file with the extension *.tar.xz*, which has to be extracted. When it is extracted, go to the extracted directory and open the terminal in that directory. Two *.sh* scripts have to be executed, the first called *arduino-linux-setup.sh* and the second called *install.sh*.

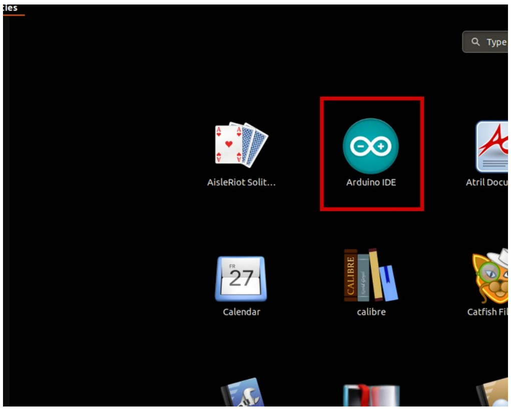
To run the first script in the terminal, open the terminal in the extracted directory and run the following command:

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

user_name - is the name of a superuser in the Linux operating system. A password for the superuser has to be entered when the command is started. Wait for a few minutes for the script to complete everything.

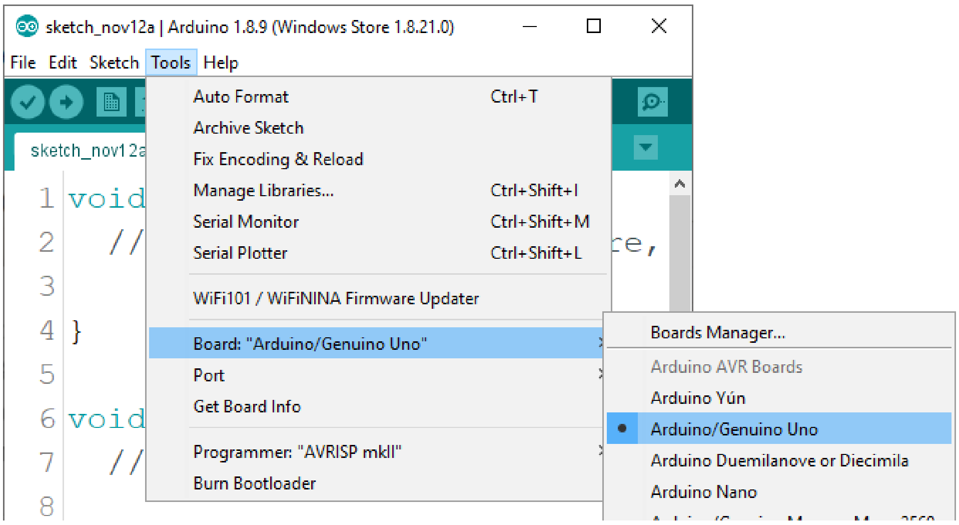
The second script called *install.sh* script has to be used after installation of the first script. Run the following command in the terminal (extracted directory): **sh install.sh**

After the installation of these scripts, go to the *All Apps*, where the *Arduino IDE* is installed.



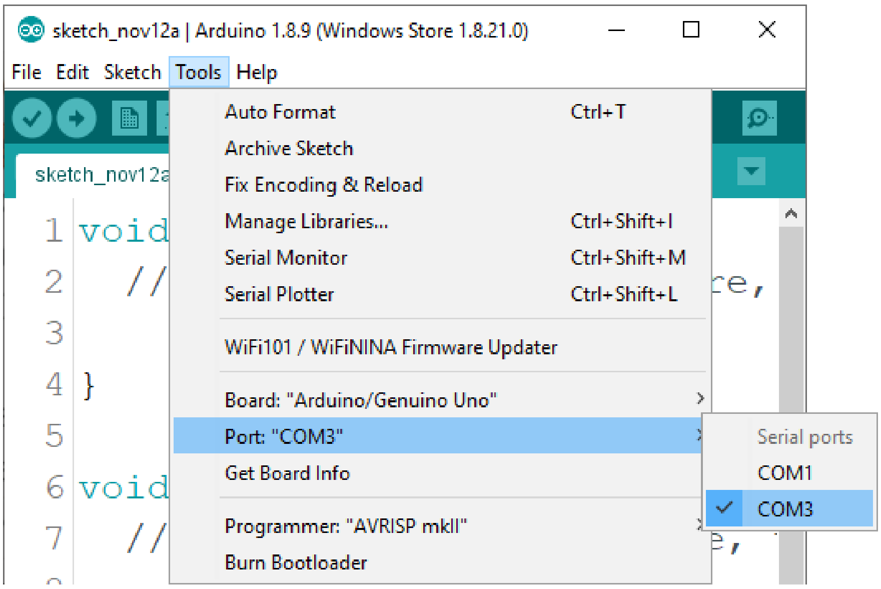
Almost all operating systems come with a text editor preinstalled (for example, *Windows* comes with *Notepad*, *Linux Ubuntu* comes with *Gedit*, *Linux Raspbian* comes with *Leafpad*, etc.). All of these text editors are perfectly fine for the purpose of the eBook.

Next thing is to check if your PC can detect an Arduino board. Open freshly installed Arduino IDE, and go to:
Tools > Board > {your board name here}
{your board name here} should be the *Arduino/Genuino Uno*, as it can be seen on the following image:


sketch_nov12a | Arduino 1.8.9 (Windows Store 1.8.21.0)
File Edit Sketch Tools Help
sketch_nov12a
1 void
2 //
3
4 }
5
6 void
7 //
8
ce,
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
Board: "Arduino/Genuino Uno"
Port
Get Board Info
Programmer: "AVRISP mkII"
Burn Bootloader
Boards Manager...
Arduino AVR Boards
Arduino Yún
● Arduino/Genuino Uno
Arduino Duemilanove or Diecimila
Arduino Nano

The port to which the Arduino board is connected has to be selected. Go to:
Tools > Port > {port name goes here}
and when the Arduino board is connected to the USB port, the port name can be seen in the drop-down menu on the previous image.

If the Arduino IDE is used on Windows, port names are as follows:


sketch_nov12a | Arduino 1.8.9 (Windows Store 1.8.21.0)
File Edit Sketch Tools Help
sketch_nov12a
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
Get Board Info
Burn Bootloader
Serial ports
COM1
✓ COM3

Board: "Arduino/Genuino Uno"

Port: "COM3"

Programmer: "AVRISP mkII"

For *Linux* users, for example port name is */dev/ttyUSBx*, where *x* represents integer number between *0* and *9*.

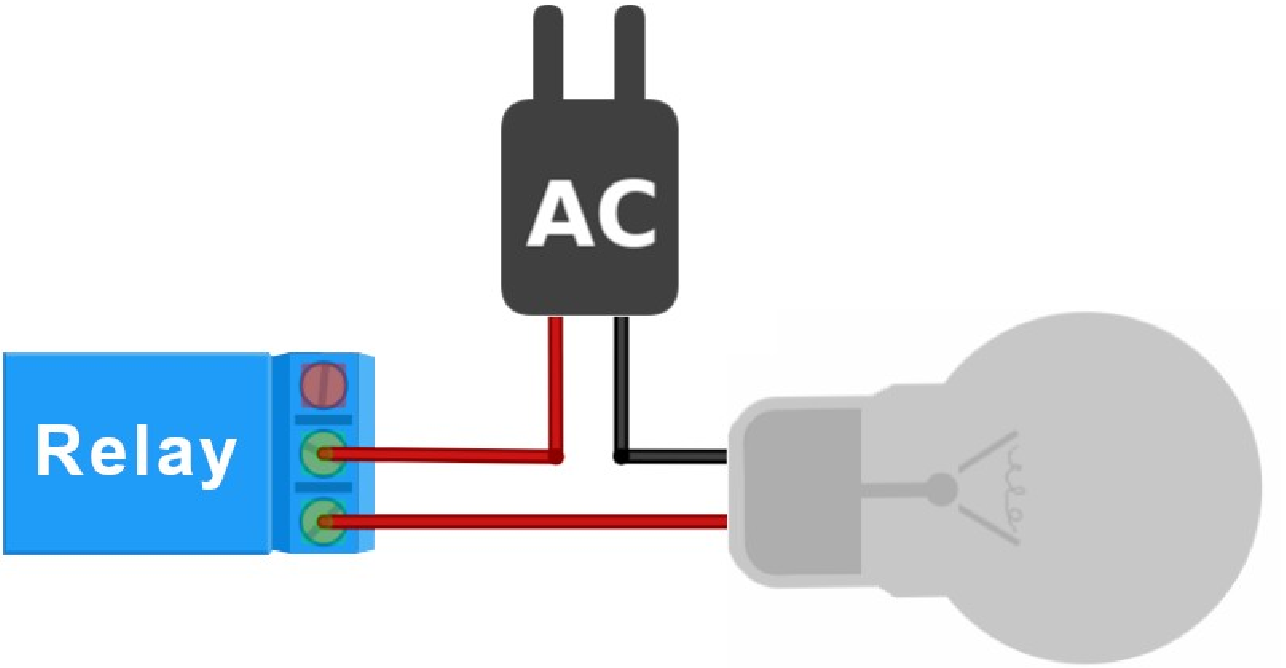
How to set-up the Raspberry Pi and Python

For the Raspberry Pi, first the operating system has to be installed, then everything has to be set-up so that it can be used in the *Headless* mode. The *Headless* mode enables remote connection to the Raspberry Pi, without the need for a *PC* screen Monitor, mouse or keyboard. The only things that are used in this mode are the Raspberry Pi itself, power supply and internet connection. All of this is explained minutely in the free eBook:
Raspberry Pi Quick Startup Guide

The *Raspbian* operating system comes with *Python* preinstalled.

Connecting the AC side of the relay

For this purpose we will be using one light bulb, a two wire cable and an AC power plug. The AC part of the connection diagram is the same for all 16 relays on-board 16 relays module. Connect the relay with light bulb and power plug as shown on the connection diagram below:

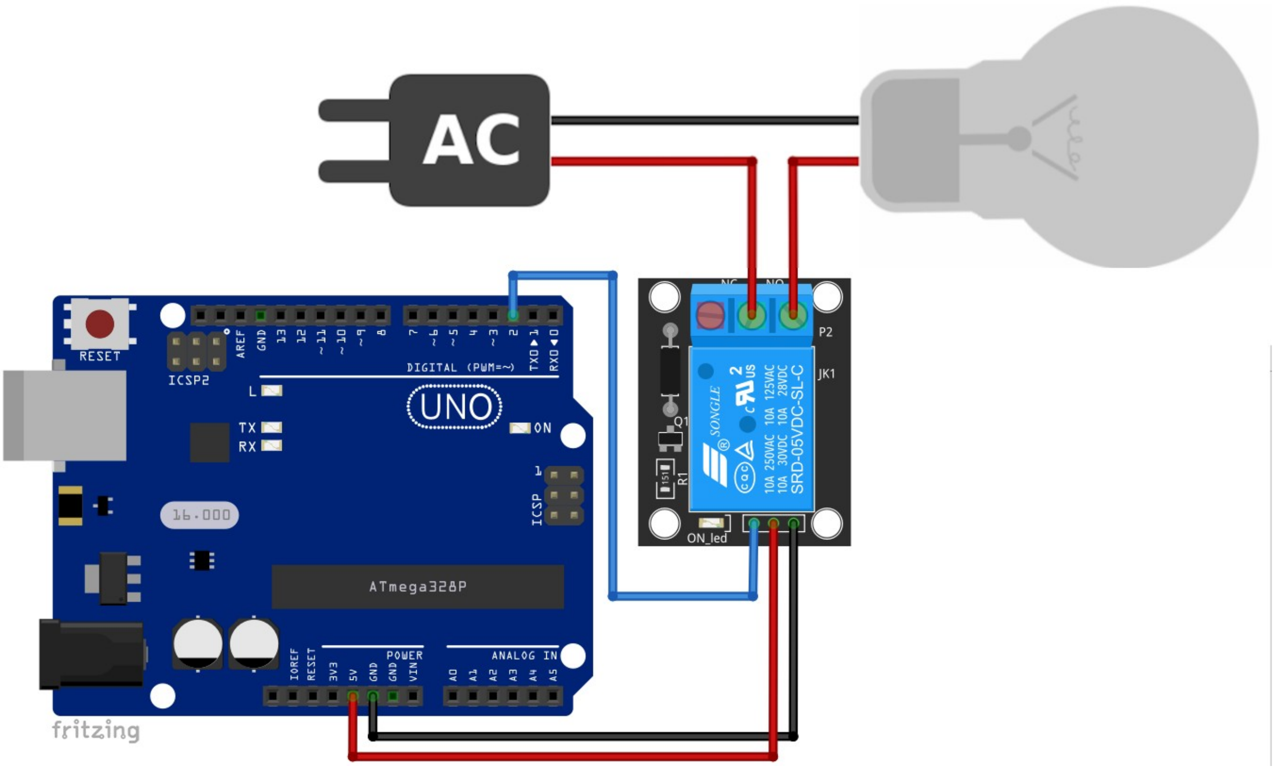


Module pin	**>**	**Power plug, light bulb**	
Common pin	>	One side of power plug	**Red wire**
Normally opened pin	>	One side of bulb	**Red wire**

Light bulb	**>**	**Power plug**	
The other side of bulb	>	The other side of power plug	**Black wire**

Connecting the module with Uno

Connect the KY-019 module with the Uno as shown on the following connection diagram:



KY-019 pin	**>**	**Uno pin**	
S	>	D2	**Blue wire**
- (GND)	>	GND	**Black wire**
+ (VCC)	>	5V	**Red wire**

Sketch example

```
#define RELAY_PIN 2
void setup() {
  pinMode(RELAY_PIN, OUTPUT);
}
void loop() {
  digitalWrite(RELAY_PIN, HIGH);
  delay(1000);

  digitalWrite(RELAY_PIN, LOW);
  delay(1000);
}
```

When the sketch is uploaded to the Uno, the clicks from the relay could be heard. When the relay change state from active to rest and vice versa, the clicks of the switch can be heard.

The light bulb should blink every second.

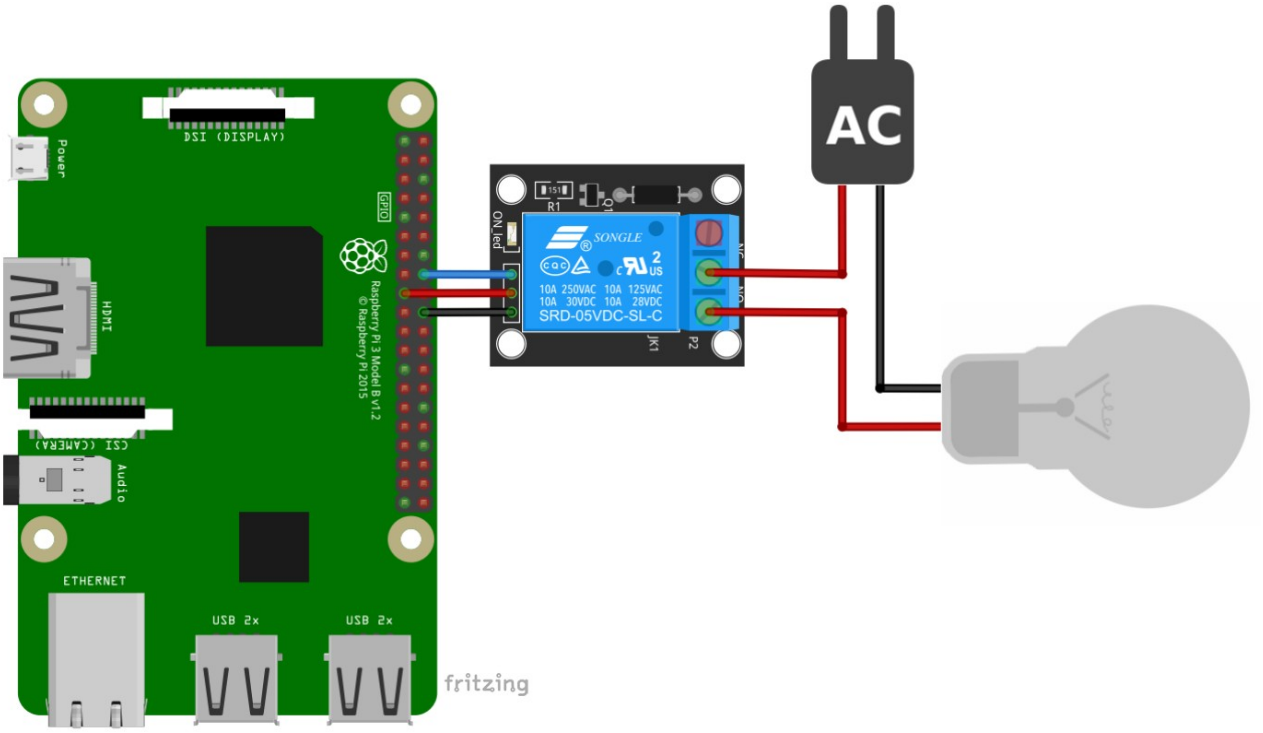
We can change NO/NC pin states by these lines of code:

`digitalWrite(2, HIGH);` – **NC** pin is not connected with the c. pin
NO pin is connected with the c. pin

`digitalWrite(2, LOW);` – **NC** pin is connected with the c. pin
NO pin is not connected with the c. pin

Connecting the module with Raspberry Pi

Connect the KY-019 module with the Raspberry Pi as shown on the following connection diagram:



KY-019 pin	**>**	**Raspberry Pi pin**		
S	>	GPIO23	[pin 16]	**Blue wire**
+ (VCC)	>	3V3	[pin 17]	**Red wire**
- (GND)	>	GND	[pin 20]	**Black wire**

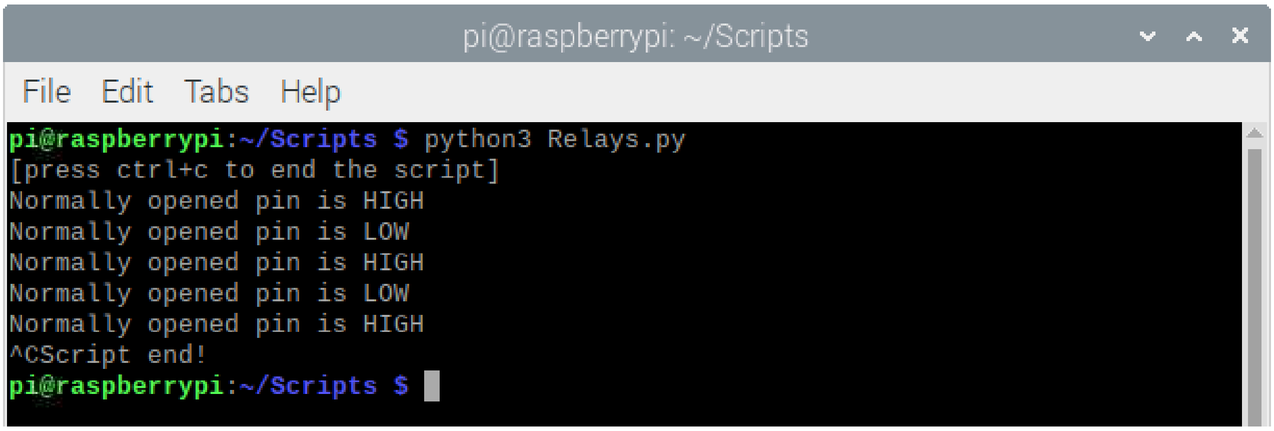
Python script

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

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

# Pin setup for the module
Relay_PIN = 23
GPIO.setup(Relay_PIN, GPIO.OUT)

print('[Press CTRL + C to end the script!]')
try: # Main program loop
    while True:
        GPIO.output(Relay_PIN, GPIO.HIGH)
        print('Normally opened pin is HIGH')
        sleep(1) # Wait for a second
        GPIO.output(Relay_PIN, GPIO.LOW)
        print('Normally opened pin is LOW')
        sleep(1) # Wait for a second

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

finally:
    GPIO.cleanup()
```

Save the script by name *Relays.py*. To run the script open the terminal in the directory where you saved the script and run the following command:

python3 Relays.py

The result should look like the output on the following image:



```
pi@raspberrypi:~/Scripts $ python3 Relays.py
[press ctrl+c to end the script]
Normally opened pin is HIGH
Normally opened pin is LOW
Normally opened pin is HIGH
Normally opened pin is LOW
Normally opened pin is HIGH
^CScript end!
pi@raspberrypi:~/Scripts $ 
```

To stiop the script press CTRL + C on the keyboard.

Now it is the time to learn and make your own projects. You can do that with the help of many example scripts and other tutorials, which you can find on the internet.

If you are looking for the high quality products for Arduino and Raspberry Pi, AZ-Delivery Vertriebs GmbH is the right company to get them from. You will be provided with numerous application examples, full installation guides, eBooks, libraries and assistance from our technical experts.

https://az-delivery.de

Have Fun!

Impressum

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