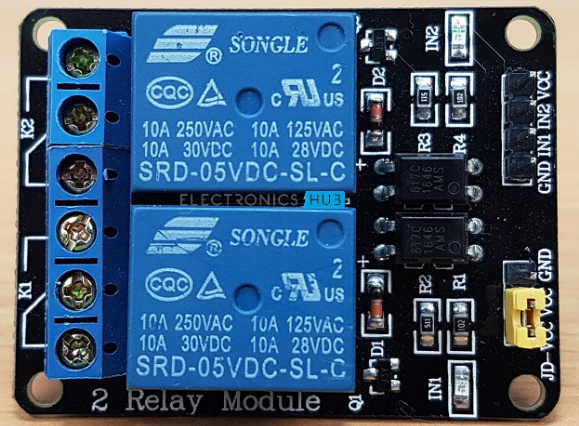
**PRACTICAL NO.10**

**10. Write a program to use Relay to control electrical equipment.**

In this project, we will implement Relay and Relay Module, interface a Relay with Raspberry Pi and see how to control a Relay using Raspberry Pi. This project could be your first step in implementing your own Home Automation Project using Raspberry Pi.

**What is a Relay?**

In layman terms, a relay is a switch. Technically speaking, a relay is an electromagnetic switch where a small control signal (usually from a microcontroller) at the input of the Relay will control a high voltage supply (usually AC mains). Since this is a Raspberry Pi based project, let us talk with respect to Raspberry Pi. The Raspberry Pi computer, although a powerful device, works on a 3.3V Logic. If you want this powerful computer to control your electrical loads, like an LED strip running along your garden or kitchen, you cannot interface them directly as the electrical loads work on AC Mains supply and the Raspberry Pi works on 3.3V DC (technically). Here comes the Relay. A simple electromechanical device that consists of a coil and few electrical contacts. When the coil is energized, it acts as an electromagnet and closes a switch. If the coil is de-energized, the coil loses its magnetic nature and releases the switch. So, by controlling the coil, you can control a switch, which in turn will control an electrical load. You can control the coil of the relay with the help of Raspberry Pi (although not directly, but with additional circuitry) as all you need is a small current to energize the coil.



### Components Required

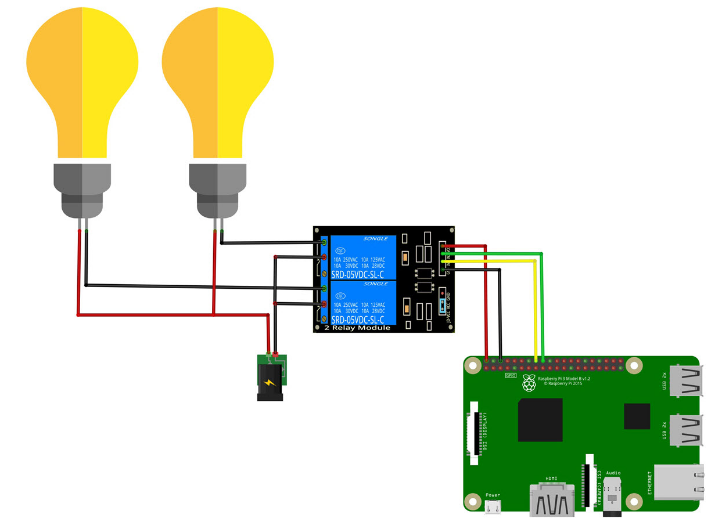
* Raspberry Pi 3 Model B
* 2-Channel Relay Module
* Two Small Incandescent Bulbs (for demonstration in the output)
* Connecting wires
* Power Supply
* Computer

### Working

The main concept behind this project is to understand the working and use of a relay and also control a relay using Raspberry Pi.

There is nothing special going in the project. All you need to do is to control the GPIO pins connected to the Relay Module. If the GPIO Pin is made HIGH, the corresponding load will be switched ON.

To turn OFF the load, make the GPIO pin LOW.



import RPi.GPIO as GPIO

import time

in1 = 16

in2 = 18

GPIO.setmode(GPIO.BOARD)

GPIO.setup(in1, GPIO.OUT)

GPIO.setup(in2, GPIO.OUT)

GPIO.output(in1, False)

GPIO.output(in2, False)

try:

while True:

for x in range(5):

GPIO.output(in1, True)

time.sleep(0.1)

GPIO.output(in1, False)

GPIO.output(in2, True)

time.sleep(0.1)

GPIO.output(in2, False)

GPIO.output(in1,True)

GPIO.output(in2,True)

for x in range(4):

GPIO.output(in1, True)

time.sleep(0.05)

GPIO.output(in1, False)

time.sleep(0.05)

GPIO.output(in1,True)

for x in range(4):

GPIO.output(in2, True)

time.sleep(0.05)

GPIO.output(in2, False)

time.sleep(0.05)

GPIO.output(in2,True)

except KeyboardInterrupt:

GPIO.cleanup()

The relay boards are powered by 5v and the relays are triggered by taking the input pin to ground. Since the Raspberry Pi's GPIO pins output 3.3v I used a set of transistors to take the relay pins to ground.

So when you are setting the GPIO output value:

1) 0 the relay starts ;

2) 1 the relay stops .