Lesson 6 Relay

Introduction

In this project, we will learn about 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.

Hardware Required

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

Principle

Relay Module

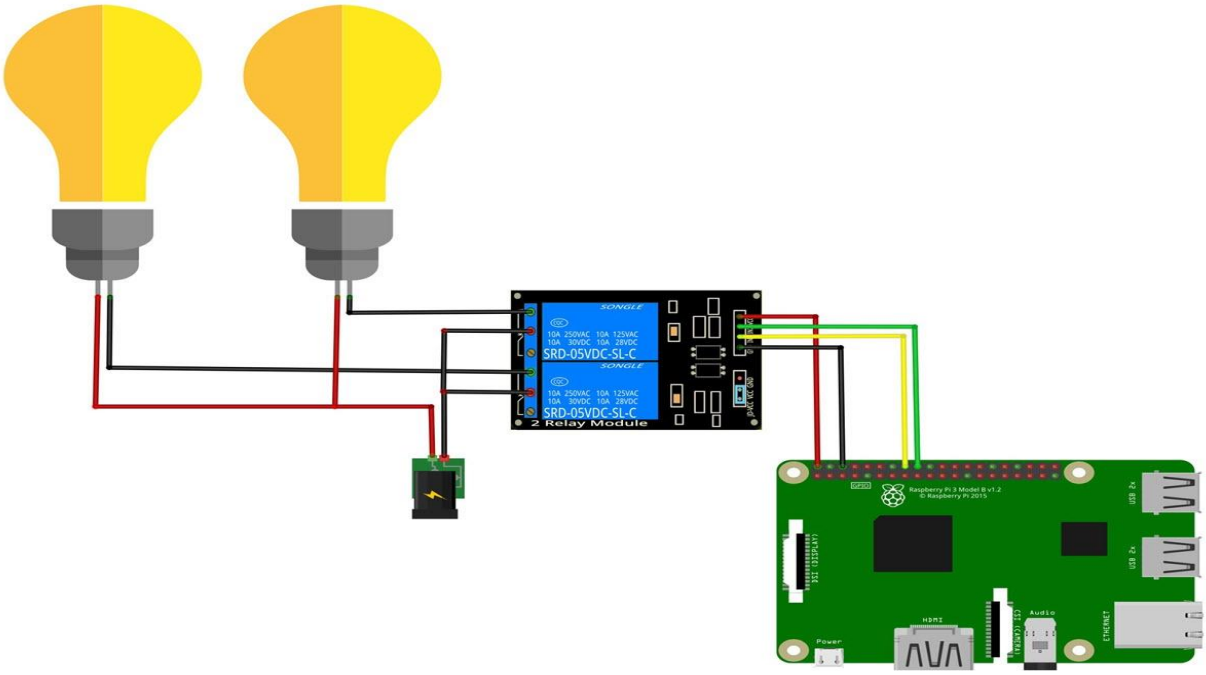
A simple electromechanical device that consists of a coil and few electrical contacts. When the coil is energized, it acts as an electromagnet and closes a switch. If the coil is de-energized, the coil loses its magnetic nature and releases the switch.

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.

The following image shows a typical 5V Relay. It has 5 pins namely: NO (Normally Open), NC (Normally Closed), COMM (Common) and two coil terminals.

Hardware Setup

Connect the inputs to the two relay channels to GPIO16 and GPIO18 of the Raspberry Pi. Then connect the loads as shown in the circuit diagram.



CAUTION: To keep this project simple, I haven't connected any AC Loads (like a CFL Bulb) to the Relay Module. But if you want to control an electrical load, be extremely careful when connecting to AC Mains supply. If needed, take help from an expert.

Python Coding

```
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()
```

Output

If the GPIO Pin is made HIGH, the corresponding load will be switched ON means the light will glow.
To turn OFF the load, make the GPIO pin LOW means light will not glow.

Application

Interfacing a Relay with Raspberry Pi and controlling the triggering of the Relay can be helpful in several applications like:

- Home Automation Appliance Control
- Automated Vehicle
- Robot