**PRACTICAL – 10**

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

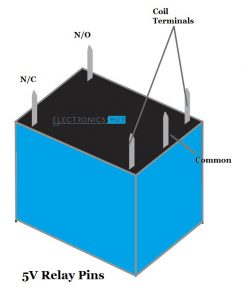
**What is a Relay?**

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).

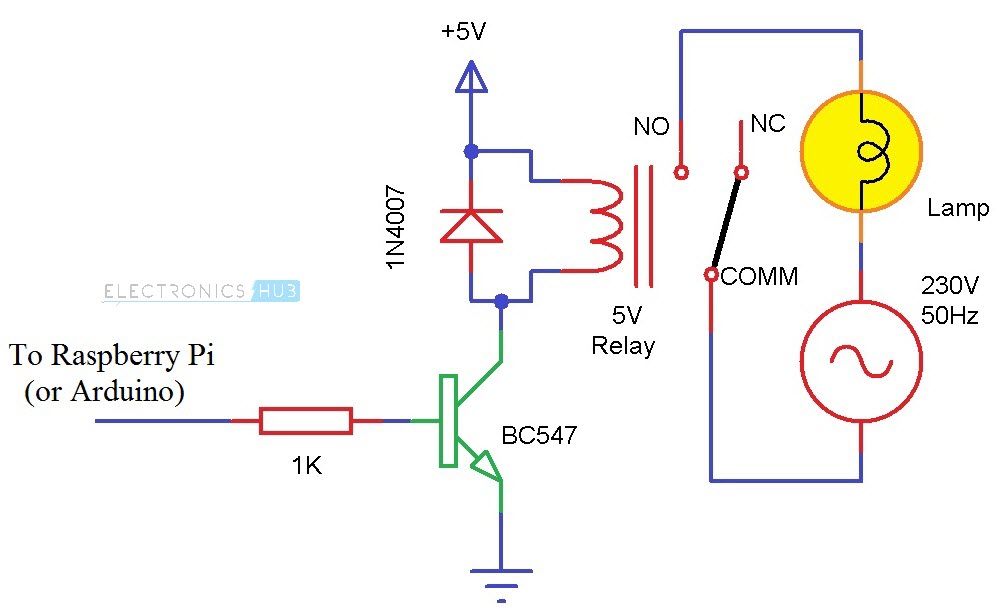
The following image shows a typical 5V Relay. It has 5 pins namely: NO (Normally Open), NC (Normally Closed), COMM (Common) and two coil terminals.



**Relay Module**

Even though the Relay Coil needs a small current in order to get energized, driving it directly from Raspberry Pi (for that matter, any Microcontroller like 8051 or Arduino) is not a good idea.

A simple way is to drive the Relay Coil through a Transistor. The following image shows the connections required with respect to a Relay.

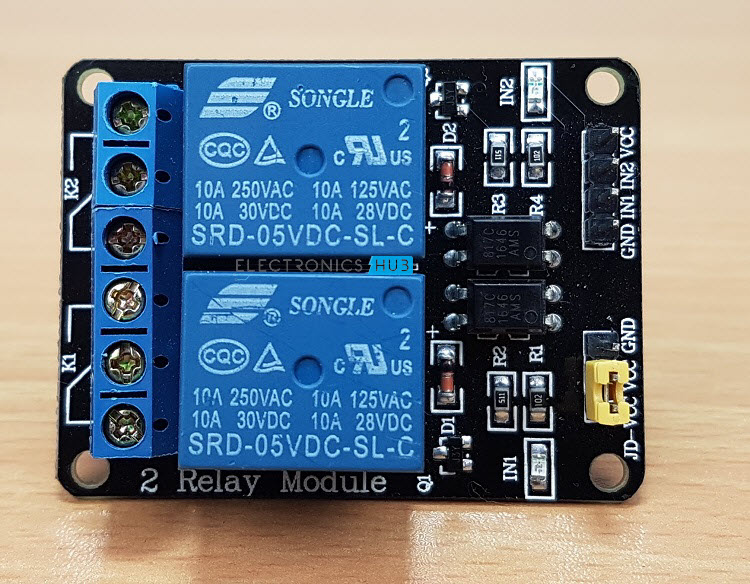


The circuit shown above will drive a relay with minimum number of components (a Transistor, a current limiting resistor, a Relay and a Diode).

If you have all these components, then you can build the circuit yourself. But if you feel that this might be tedious job, we have an alternative: a Relay Module.

A Relay Module is a simple circuit board that consists of the relay itself and all the necessary components that are required to drive a relay and also the required connectors to connect the load.

I have used a two channel relay module in this project. It is basically, two relays with all the circuitry on a single board.



I suggest you to pick up a relay module like this one as you might be dealing with AC Supply in a future project and connecting with screw terminals will make your job easy.

**How to Control a Relay using Raspberry Pi?**

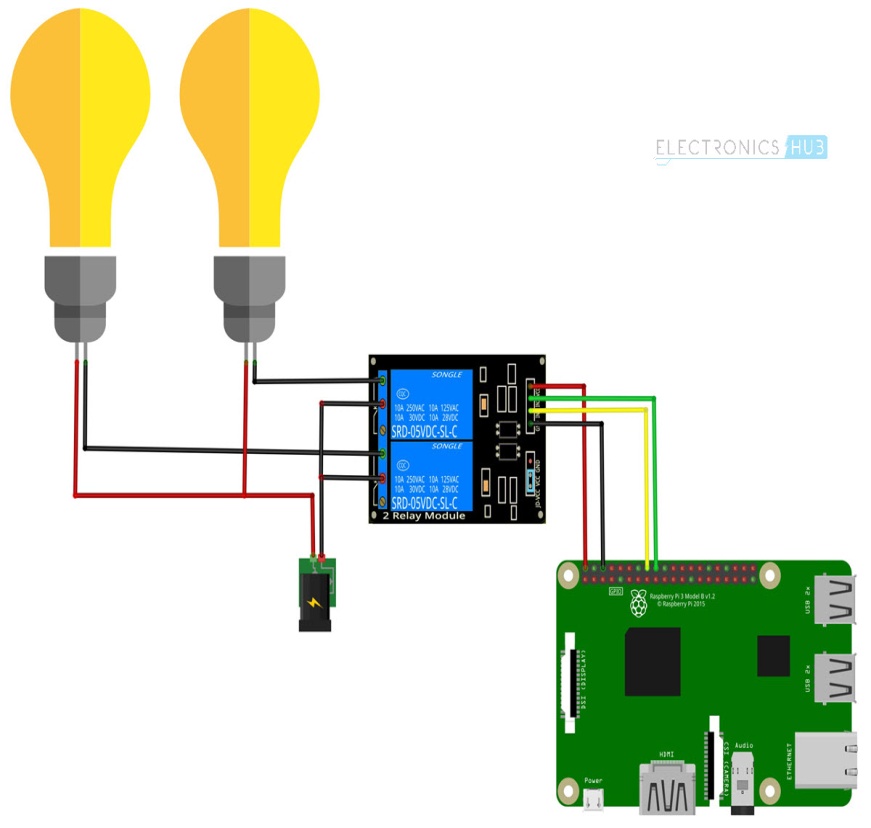
Till now, we have seen about relay, what is the necessity of relays, how relay works and also about relay modules. Now comes the interesting part i.e. How to Control a Relay using Raspberry Pi?

If you understood the concept of a relay, then it might be clear that all you need to do is to control the coil of the relay i.e. if the Raspberry Pi wants a load to be turned ON, then activate the relay by energizing the coil (sending a HIGH signal from Raspberry Pi).

Similarly, if the Raspberry Pi wants to turn the load OFF, then de-energize the coil by sending a LOW signal.

**Circuit Diagram**

The following image shows the connections with respect to the project of How to Control a Relay using Raspberry Pi.



**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

**Circuit Design**

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.

**Code**

The Python Script for the project How to Control a Relay using Raspberry Pi is given below.

|  |
| --- |
| 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() |

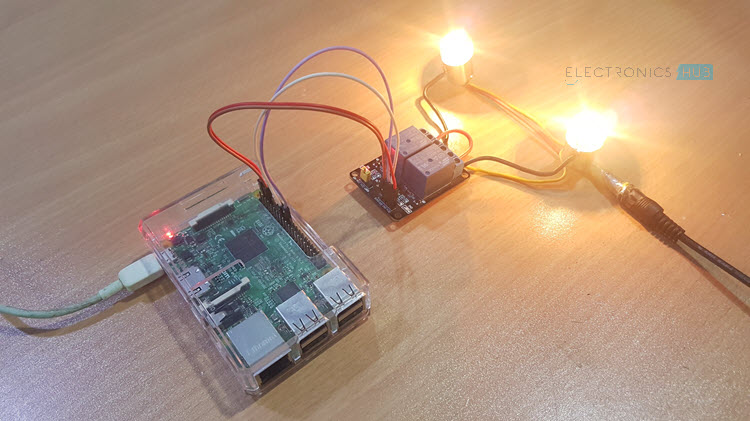
**Working**

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.

**Output:-**



|  |
| --- |
|  |
|  |  |