**Practical 10**

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

**What is a Relay?**

A relay is an [electrically](https://en.wikipedia.org/wiki/Electric) operated [switch](https://en.wikipedia.org/wiki/Switch). It consists of a set of input terminals for a single or multiple control signals, and a set of operating contact terminals. The switch may have any number of contacts in multiple [contact forms](https://en.wikipedia.org/wiki/Electrical_contact#Contact_form), such as make contacts, break contacts, or combinations thereof.

Relays are used where it is necessary to control a circuit by an independent low-power signal, or where several circuits must be controlled by one signal. Relays were first used in long-distance [telegraph](https://en.wikipedia.org/wiki/Electrical_telegraph) circuits as signal repeaters: they refresh the signal coming in from one circuit by transmitting it on another circuit. Relays were used extensively in telephone exchanges and early computers to perform logical operations.

The traditional form of a relay uses an [electromagnet](https://en.wikipedia.org/wiki/Electromagnet) to close or open the contacts, but other operating principles have been invented, such as in [solid-state relays](https://en.wikipedia.org/wiki/Solid-state_relay) which use [semiconductor](https://en.wikipedia.org/wiki/Semiconductor) properties for control without relying on [moving parts](https://en.wikipedia.org/wiki/Moving_parts). Relays with calibrated operating characteristics and sometimes multiple operating coils are used to protect electrical circuits from overload or faults; in modern electric power systems these functions are performed by digital instruments still called [protective relays](https://en.wikipedia.org/wiki/Protective_relay).

Firstly we will connect the raspberry and the 4 array module and test it locally. The next step is to make a web app that will control our module . We will make it with apache server , php , javascript , jquery , bootstrap , bootstrap- switch buttons and some python.

**Step 1: What You Need**



We need :

1. Raspberry Pi
2. 4 Array Relay Module
3. 6 Wires
4. Wifi Dongle
5. Filezilla (optional)
6. SHH (optional)

## Step 2: Wiring

## Wiring

## Wiring

## Wiring

We need to connect our Raspberry and the module .We will get the 5V from the Raspberry (not very recommended but its easy ) .So lets begin :

PI GPIO | Relay module

Pin 4 ---------VCC

Pin 6 ---------GND

Pin 8 ----------IN4

Pin 10 ---------IN3

Pin 11 ---------IN2

Pin 12 ---------IN1

## Step 3: Testing

## Testing

## Testing

## Now we need to start the raspberry and to connect to the power supply and connect to it over FTP with **Filezilla** or start it on monitor . Now write the following code and save the code with .py extension.

## # getting the main GPIO libraly

## import RPi.GPIO as GPIO

## # getting the time libraly

## import time

## # setting a current mode

## GPIO.setmode(GPIO.BCM)

## #removing the warings

## GPIO.setwarnings(False)

## #creating a list (array) with the number of GPIO's that we use

## pins = [18,17,15,14]

## #setting the mode for all pins so all will be switched on

## GPIO.setup(pins, GPIO.OUT)

## #for loop where pin = 18 next 17 ,15, 14

## for pin in pins :

## #setting the GPIO to HIGH or 1 or true

## GPIO.output(pin, GPIO.HIGH)

## #wait 0,5 second

## time.sleep(0.5)

## #setting the GPIO to LOW or 0 or false

## GPIO.output(pin, GPIO.LOW)

## #wait 0,5 second

## time.sleep(0.5)

## #Checking if the current relay is running and printing it

## if not GPIO.input(pin) :

## print("Pin "+str(pin)+" is working" )

## 

## #same but the difference is that we have

## #for loop where pin = 14 next 15,17,18

## # backwards

## for pin in reversed(pins) :

## GPIO.output(pin, GPIO.HIGH)

## time.sleep(0.5)

## GPIO.output(pin, GPIO.LOW)

## time.sleep(0.5)

## #cleaning all GPIO's

## GPIO.cleanup()

## print "Shutdown All relays"

## Then start the python script by typing :-

## sudo python test\_relay.py

## If leds start flashing from IN1 to IN4 and shutdown from IN4 to IN1 then everything is okey and the wiring is like currect .We can move to the next step .

**Explanation:-**

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 .

## Step 4: Installing Apache and Php:-

## We will need them for the server part .Helpfull link.

## To install apache2 we need to run :

## sudo apt-get install apache2 -y

## To test the webserver we can you on the raspberry browser

## http://localhost/

## or

## http://192.168.1.10

## (raspberry ip address) .Type

## ifconfig

## to get the current IP .

## Next install PHP5 module for APACHE :

## sudo apt-get install php5 libapache2-mod-php5 -y

## And remove index.html file :

## sudo rm index.html

## The next thing is to navigate to the default directory:

## For Raspbian Wheezy is /var/www ,

## For Raspbian Jessie is /var/www/html

## So we type :

## cd /var/www/html<br>ls -al

## And this will output :

## total 12<br>drwxr-xr-x 2 root root 4096 Jan 8 01:29 .<br>drwxr-xr-x 12 root root 4096 Jan 8 01:28 ..<br>-rw-r--r-- 1 root root 177 Jan 8 01:29 index.html

## We need to change the permission if we want to upload or modify files :

## sudo chown pi: index.html

## Delete the index.html file from the dir.

## Step 5: Uploading the Needed Files for Your Web App:-

## 

You need to download the .rar file , extract it and copy the files with Filezilla to the main dir of Raspberry home/pi and to the var/www. For pasting the files into var/www you need to set a permision .

at var/www<br>bootstrap.css<br>bootstrap-switch.css<br>bootstrap-switch.js<br>changeState.php<br>firstCheck.php<br>index.html<br>jquery.js

at home/pi

firstCheck.py

relay\_off.py

relay\_on.py

When you are done coping the files , just go to your browser , type RaspberryIP/index.html (ex: 192.168.1.23/index.html) and the webpage will be opened .And also you can test it and remotely control the relays from any device that is connected to your local network(router) .