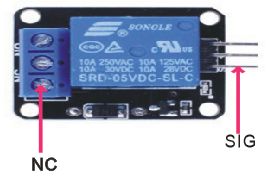
Relay  


Overview

We will learn to use the Raspberry Pi to control relay module and control the opening and closing of the relay output circuit by giving the relay different signals.

Materials Needed

Raspberry Pi x1

Breadboard x1

Relay module x1

LED (3-pin) x1

Resistor(330Ω) x1

Dupont jumper wires

 Preparatory work

1. Install python interpreter in your Raspberry Pi system

2. Install the RPi.GPIO library in your Raspberry Pi system

3. Install the wiringPi library in your Raspberry Pi system

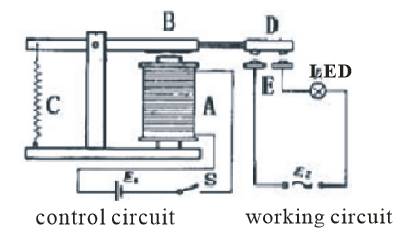
See the attached “Installing a Python Interpreter and Corresponding Libraries in a Raspberry Pi System”for details.

Product Description

Brief Introduction

Electromagnetic relay circuit can be divided into low voltage control circuit and high voltage working circuit. The control circuit is composed of electromagnet A, armature B, low-voltage power supply E1 and switches; the working circuit is composed of LED lamps, power supply E2 and the static contacts and moving contacts equaling to the switches.

After the working circuit is connected,in the normal state, D and E are disconnected,the working circuit is off-state.When the switch S is closed, the armature B is sucked down by the electromagnet, and the movable contact D is in contact with the two stationary contacts at the same time, so that the D and the E connect with each other. At this time, the spring is stretched and the working circuit is observed to be turned on and the LED lamp is turned on. Disconnect switch S, the electromagnet loses its magnetic properties and it won’t suck the armature. The armature returns to its original position under the tension of the spring. The movable contact is separated from the stationary contact. The working circuit is cut off and the LED lamp turns off.



Experimental phenomena

With the work of the relay, the LED light will be on and off in circle.

Characteristic Parameters

◆Operation Voltage: 5V

◆Operation Time: 10ms(max)

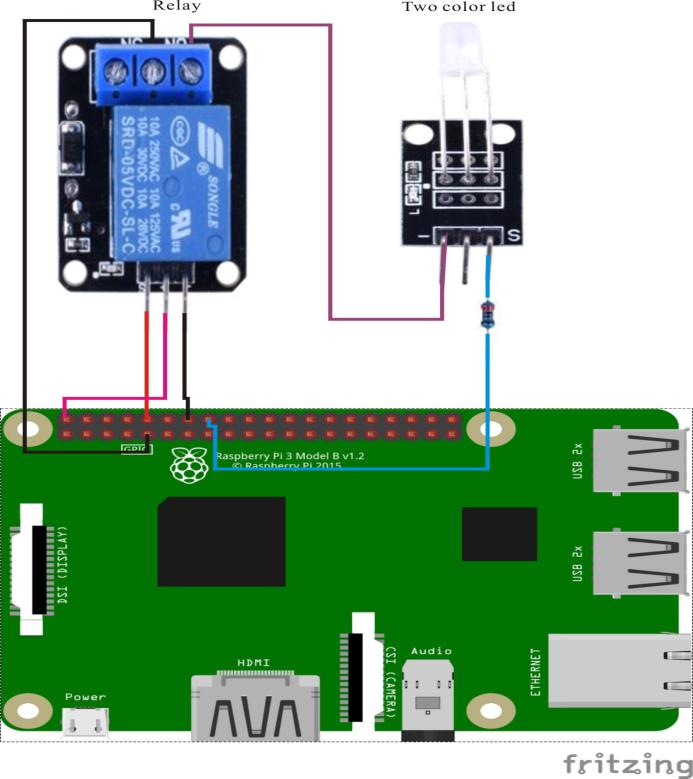
◆Insulation Resistance: 100MΩ(min)

◆Ambient Temperature: -25°C~+70°C

◆Ambient Humidity: 45%~85%RH

◆Electrical Level Type: Active High

Wiring Diagram



Relay pin position:

"S" ↔ Raspberry Pi pin 10

"+" ↔ Raspberry Pi +5V

"-" ↔ Raspberry Pi GND

LED pin position:

"S” ↔ Raspberry Pi pin 16 (through resistor)

"-" ↔ Raspberry Pi GND

Sample Code

Python code

#!/usr/bin/env python

import RPi.GPIO as GPIO

import time

RelayPin = 10 # pin10

LedPin = 16

def setup():

GPIO.setmode(GPIO.BOARD) # Numbers GPIOs by physical location

GPIO.setup(RelayPin, GPIO.OUT)

GPIO.setup(LedPin, GPIO.OUT)

GPIO.output(LedPin, GPIO.HIGH)

def loop():

while True:

print '...relayd on'

GPIO.output(RelayPin, GPIO.HIGH)

time.sleep(0.5)

print 'relay off...'

GPIO.output(RelayPin, GPIO.LOW)

time.sleep(0.5)

def destroy():

GPIO.output(RelayPin, GPIO.HIGH)

GPIO.cleanup() # Release resource

if \_\_name\_\_ == '\_\_main\_\_': # Program start from here

setup()

try:

loop()

except KeyboardInterrupt:

destroy()

C code

#include <wiringPi.h>

#include <stdio.h>

#define RelayPin 16

#define LedPin 4

int main(void)

{

if(wiringPiSetup() == -1)

{

printf("setup wiringPi failed !");

return 1;

}

pinMode(RelayPin, OUTPUT);

pinMode(LedPin, OUTPUT);

digitalWrite(LedPin, HIGH);

while(1)

{

digitalWrite(RelayPin, HIGH);

delay(500);

digitalWrite(RelayPin, LOW);

delay(500);

}

return 0;

}