**Touch switch**

Overview

This course will use the Raspberry Pi to control the touch switch and control the LED light on and off by capturing the signal of the touch switch.

Materials Needed

Raspberry Pi \*1

Breadboard \*1

Touch switch \*1

Led \*1

Dupont Line

**The 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

For more details, please refer to the attachment for installing the python interpreter and corresponding libraries in the Raspberry Pi system.

Product Description

Brief Introduction

Touch wall switch is successful application of electronic substitution mechanism. The touch switch has no metal contact, no discharge and no ignition, which make significant savings in copper alloy materials. At the same time, the requirements for the mechanical structure are greatly reduced. It directly replaces traditional switches, providing comfortable operation, excellent feel, precise control and no mechanical wear. At the same time, the touch switch is more humane, you can choose the text prompts on the switch, personalized text labels show the effect of liquid crystal display. Crystal panel issued a faint glimmer, which make late night is no longer completely dark, enough to make people form a sense of position and contour. The principle of touch switch is triggered by the induction voltage of the human body, there are many electromagnetic waves in our environment, which can cause the human body to produce the induced voltage. Because there is an induction voltage to trigger the input of the touch switch (normally a COMS tube of high input impedance) to make it lead to control the circuit breaker.



**Characteristic Parameters**

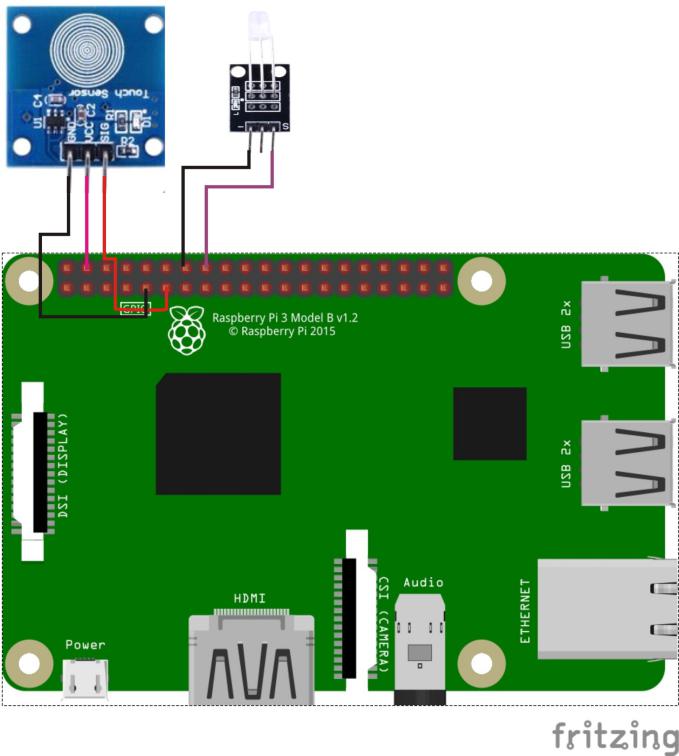
◆Type: Capacitive Point Type.

◆Power Supply Voltage: 2~5.5V.

◆Touch Surface: Both sides can be used as the touch surface.

◆The initial state is low level, the touch is high level, not touch is low level.

Wiring diagram



**The Sample Code**

1. **Python Code**

#!/usr/bin/env python

import RPi.GPIO as GPIO

import time

LedPin = 16

TouchPin = 11

def setup():

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

GPIO.setup(LedPin, GPIO.OUT) # Set LedPin's mode is output

GPIO.setup(TouchPin, GPIO.IN, pull\_up\_down=GPIO.PUD\_UP)

GPIO.output(LedPin, GPIO.LOW) # Set LedPin low to off led

def loop():

while True:

if GPIO.input(TouchPin) == GPIO.HIGH:

print 'touched!'

GPIO.output(LedPin, GPIO.HIGH) # led on

time.sleep(0.2)

else:

GPIO.output(LedPin, GPIO.LOW) # led off

def destroy():

GPIO.output(LedPin, GPIO.HIGH) # led off

GPIO.cleanup() # Release resource

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

setup()

try:

loop()

except KeyboardInterrupt:

destroy()

1. **C Code**

#include <wiringPi.h>

#include <stdio.h>

#define TouchPin 0

#define LedPin 4

int main(void)

{

if(wiringPiSetup() == -1)

{

printf("setup wiringPi failed !");

return -1;

}

pinMode(TouchPin, INPUT);

pinMode(LedPin, OUTPUT);

while(1)

{

if(digitalRead(TouchPin) == HIGH)

{

printf("touched\n");

digitalWrite(LedPin, HIGH); //led on

delay(100);

digitalWrite(LedPin, LOW); //led off

}

}

return 0;

}

**The Experimental Phenomena**

When the sensor is touched, the LED light is on. When the sensor is not touched, the LED lights go out.