

## Touch Sensor



### Overview

Touch sensors are switches, in the electronics sense, that—unlike conventional push-buttons or throw switches—have no moving parts, no metal contacts, and no electrical discharge requirement. In turn this greatly simplifies their mechanical structure, saves copper alloys in their construction, and reduces the impact of wear-and-tear. They are often combined with, or embedded in, liquid crystal display panels, allowing them to be augmented by local illumination, dynamically-changing text and graphical prompts, and other visual “packaging.” (Modern smartphone and tablet touchscreens are examples of such combinations.) Touch sensors are triggered by the induction voltage of the human body itself. In this experiment, you’ll use the Raspberry Pi switch an LED light on and off in response to “touches” sensed by the touch switch.

### Experimental Materials

Raspberry Pi	x1
Breadboard	x1
Touch Sensor	x1
LED (3 pin)	x1
Resistor (330Ω)	x1
Dupont jumper wires	

### Experimental Procedure

1. If you have not done so already, prepare your development system by installing the Python interpreter, RPi.GPIO library, and wiringPi library as described in READ\_ME\_FIRST.TXT.
2. Install the touch sensor and three-pin LED on your breadboard, connecting them to the Raspberry Pi using the resistor and Dupont jumper wires as illustrated in the Wiring Diagram below. Note you will connect only two of the three pins on the LED.

3. Execute the sample stored in this experiment's subfolder.  
If using C, compile and execute the C code:

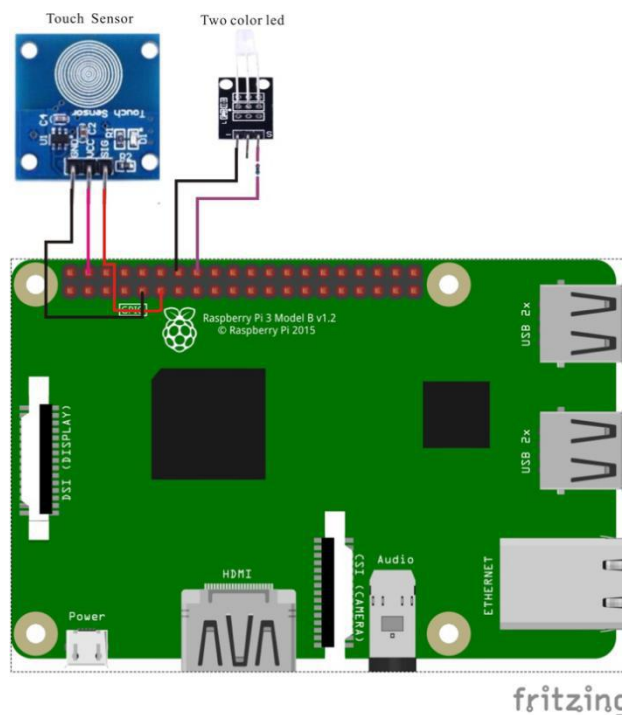
```
cd Code/C
gcc touchSensor.c -o touchSensor.out -lwiringPi
touchSensor.out
```

If using Python, launch the Python script:

```
cd Code/Python
python touchSensor.py
```

4. Make experimental observations. When you touch the sensor, the LED illuminates. When you stop touching the sensor, it goes dark.

## Wiring Diagram



Touch Sensor pin position:

SIG	↔	Raspberry Pi pin 11
VCC	↔	Raspberry Pi +5V
GND	↔	Raspberry Pi GND

LED pin position:

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

"-" ↔ Raspberry Pi GND

## Technical Background

◆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

## Sample Code

### 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()
```

## 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)
    {
        digitalWrite(LedPin, digitalRead(TouchPin));
        delay(200);
    }

    return 0;
}
```