

10.Tilt_Ball_Switch

Introduction

This is a ball tilt-switch with a metal ball inside. It is used to detect inclinations of a small angle.

Hardware Required

- ✓ 1 * Raspberry Pi
- ✓ 1 * T-Extension Board
- ✓ 1 * Tilt
- ✓ 2 * LED
- ✓ 1 * 40-pin Cable
- ✓ Several Jumper Wires
- ✓ 1 * Breadboard
- ✓ 2 * Resistor(220 Ω)
- ✓ 1 * Resistor(10K Ω)

Principle

Tilt Ball switch

Tilt Switch with internal ball that will switch to ON state of approx. 15 degrees tilt.

Also great for sensing excessive vibration

Material: Housing and cover: PE heat shrinkable tubing

Ball: Stainless steel

Shape: Round

Color: Black

Contact Rating:12V

Contact Resistance: <10 ohm

Insulation Resistance:>10M ohm

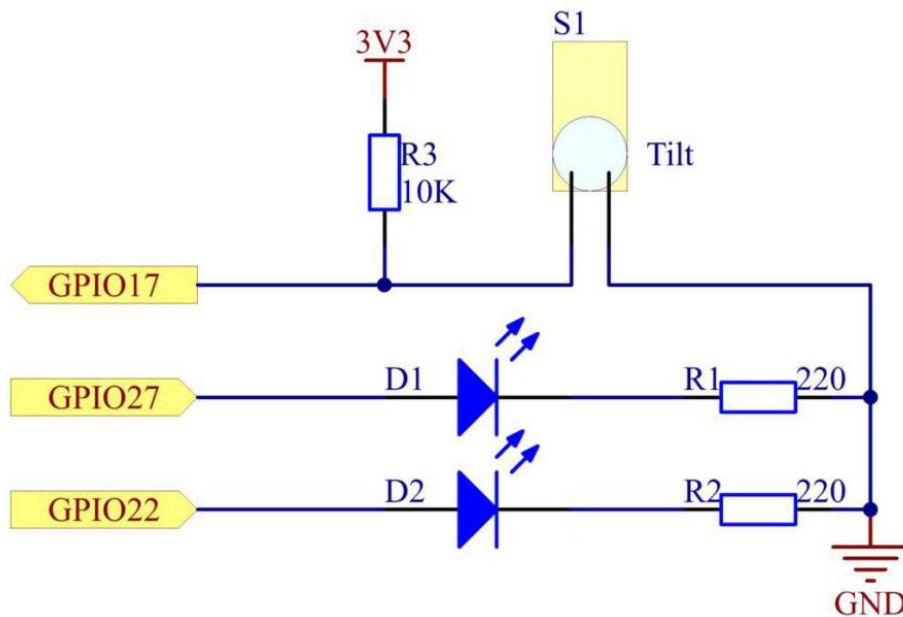
Capacitance:5PF

10.Tilt_Ball_Switch



Schematic Diagram

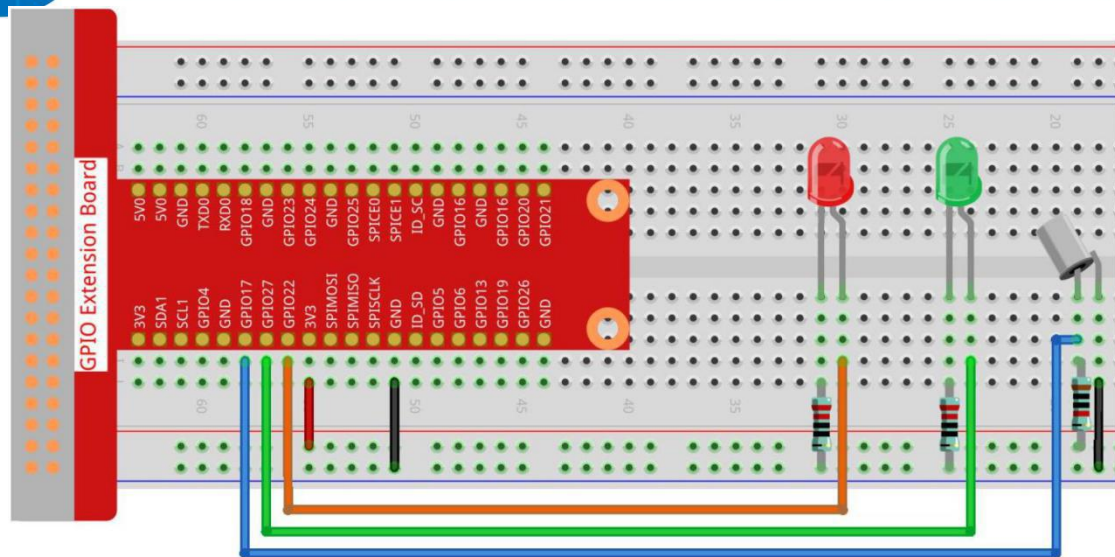
T-Board Name	physical	wiringPi	BCM
GPIO17	Pin 11	0	17
GPIO22	Pin 15	3	22
GPIO27	Pin 13	2	27



Experimental Procedures

Step 1: Build the circuit.

10.Tilt_Ball_Switch



For C Language Users

Step 2: Change directory.

```
cd /home/pi/REXQualis_Raspberry_Pi_Complete_Starter_Kit/C/10.Tilt_Ball_Switch
```

Step 3: Compile.

```
gcc 10.Tilt_Ball_Switch.c -o Tilt_Ball_Switch.out -lwiringPi
```

Step 4: Run.

```
sudo ./Tilt_Ball_Switch.out
```

Place the tilt horizontally, and the green LED will turn on. If you tilt it, "Tilt!" will be printed on the screen and the red LED will light on. Place it horizontally again, and the green LED will turn on again.

Code

```
#include <wiringPi.h>
#include <stdio.h>

#define TiltPin    0 //wiringpi 0
#define Gpin      2 //wiringpi 2
#define Rpin      3 //wiringpi 3
```

10.Tilt_Ball_Switch

```
void LED(char* color)
{
    pinMode(Gpin, OUTPUT);
    pinMode(Rpin, OUTPUT);
    if (color == "RED")
    {
        digitalWrite(Rpin, HIGH);
        digitalWrite(Gpin, LOW);
    }
    else if (color == "GREEN")
    {
        digitalWrite(Rpin, LOW);
        digitalWrite(Gpin, HIGH);
    }
    else
        printf("LED Error");
}

int main(void)
{
    if(wiringPiSetup() == -1){ //when initialize wiring failed,print message to screen
        printf("setup wiringPi failed !");
        return 1;
    }

    pinMode(TiltPin, INPUT);
    LED("GREEN");

    while(1){ //loop to observe the Tiltpin
```

10.Tilt_Ball_Switch

```
    if(0 == digitalRead(TiltPin)){
        delay(10);
        if(0 == digitalRead(TiltPin)){
            LED("RED");
            printf("Tilt!\n");
        }
    }
    else if(1 == digitalRead(TiltPin)){
        delay(10);
        if(1 == digitalRead(TiltPin)){
            LED("GREEN");
        }
    }
}
return 0;
}
```

Code Explanation

```
void LED(char* color)
{
    pinMode(Gpin, OUTPUT);
    pinMode(Rpin, OUTPUT);
    if (color == "RED")
    {
        digitalWrite(Rpin, HIGH);
        digitalWrite(Gpin, LOW);
    }
    else if (color == "GREEN")
    {
        digitalWrite(Rpin, LOW);
    }
}
```

10.Tilt_Ball_Switch

```

        digitalWrite(Gpin, HIGH);
    }
    else
        printf("LED Error");
}

```

Define a function LED () to turn the two LEDs on or off. If the parameter color is RED, the red LED lights up; similarly, if the parameter color is GREEN, the green LED will turns on.

```

while(1){ //loop to observe the Tiltpin
    if(0 == digitalRead(TiltPin)){
        delay(10);
        if(0 == digitalRead(TiltPin)){
            LED("RED");
            printf("Tilt!\n");
        }
    }
    else if(1 == digitalRead(TiltPin)){
        delay(10);
        if(1 == digitalRead(TiltPin)){
            LED("GREEN");
        }
    }
}
}

```

If the read value of tilt switch is 0, it meas that the tilt switch is tilted then you write the parameter "RED" into function LED to get the red LED lighten up; otherwise, the green LED will lit.

For Python Language Users

Step 2: Change directory.

10.Tilt_Ball_Switch

```
cd /home/pi/REXQualis_Raspberry_Pi_Complete_Starter_Kit/Python
```

Step 3: Run.

```
sudo python3 10.Tilt_Ball_Switch.py
```

Place the tilt horizontally, and the green LED will turns on. If you tilt it, "Tilt!" will be printed on the screen and the red LED will turns on. Place it horizontally again, and the green LED will lights on.

Code

The code here is for Python3, if you need for Python2, please open the code with the suffix py2 in the attachment.

```
#!/usr/bin/env python3
import RPi.GPIO as GPIO

TiltPin = 11 #Tile pin
Gpin    = 13 #red pin
Rpin    = 15 #green pin

#notion BOARD

def setup():
    GPIO.setmode(GPIO.BOARD)      # Numbers GPIOs by physical location
    GPIO.setup(Gpin, GPIO.OUT)    # Set Green Led Pin mode to output
    GPIO.setup(Rpin, GPIO.OUT)    # Set Red Led Pin mode to output
    GPIO.setup(TiltPin, GPIO.IN, pull_up_down=GPIO.PUD_UP) # Set
    BtnPin's mode is input, and pull up to high level(3.3V)
    GPIO.add_event_detect(TiltPin, GPIO.BOTH, callback=detect, bouncetime=200)

def Led(x):
    if x == 0:
        GPIO.output(Rpin, 1)
```

10.Tilt_Ball_Switch

```
GPIO.output(Gpin, 0)

if x == 1:
    GPIO.output(Rpin, 0)
    GPIO.output(Gpin, 1)

def Print(x):
    if x == 0:
        print ('*****')
        print ('*   Tilt!   *')
        print ('*****')

def detect(chn):
    Led(GPIO.input(TiltPin))
    Print(GPIO.input(TiltPin))

def loop():
    while True:
        pass

def destroy(): #leave the program
    GPIO.output(Gpin, GPIO.HIGH) # Green led off
    GPIO.output(Rpin, GPIO.HIGH) # Red led off
    GPIO.cleanup() # Release resource

if __name__ == '__main__': # Program start from here
    setup()
    try:
        loop()
```


10.Tilt_Ball_Switch

```

except KeyboardInterrupt: # When 'Ctrl+C' is pressed, the child program
destroy() will be executed.
    destroy()

```

Code Explanation

```
GPIO.add_event_detect(TiltPin, GPIO.BOTH, callback=detect, bouncetime=200)
```

Set up a detect on TiltPin, and callback function to detect.

```

def Led(x):
    if x == 0:
        GPIO.output(Rpin, 1)
        GPIO.output(Gpin, 0)
    if x == 1:
        GPIO.output(Rpin, 0)
        GPIO.output(Gpin, 1)

```

Define a function Led() to turn the two LEDs on or off. If x=0, the red LED lights up; otherwise, the green LED will be lit.

```

def Print(x):
    if x == 0:
        print ('*****')
        print (' * Tilt! *')
        print ('*****')

```

Create a function, Print() to print the characters above on the screen.

```

def detect(chn):
    Led(GPIO.input(TiltPin))
    Print(GPIO.input(TiltPin))

```

Define a callback function for tilt callback. Get the read value of the tilt switch then the function Led () controls the turning on or off of the two LEDs that is depended on the read value of the tilt switch.

10.Tilt_Ball_Switch

Phenomenon Picture

