button

Overview

This course will use the key to control the LED light off

Experimental Materials:

Raspberry Pi *1

T-type expansion board *1

Breadboard *1

5mm red led *1

Button *1

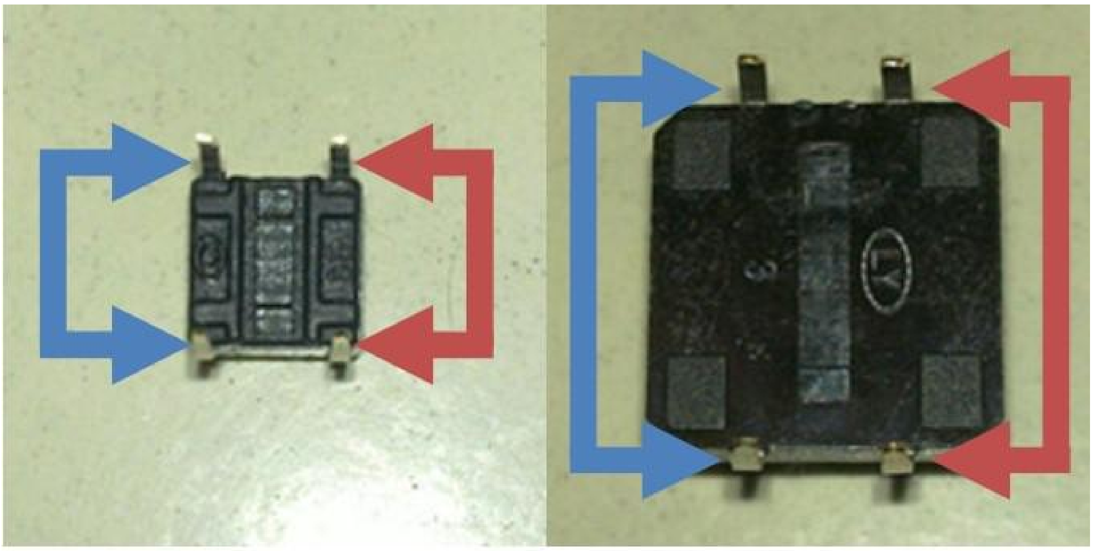
220 ohm resistor *1

Some DuPont lines

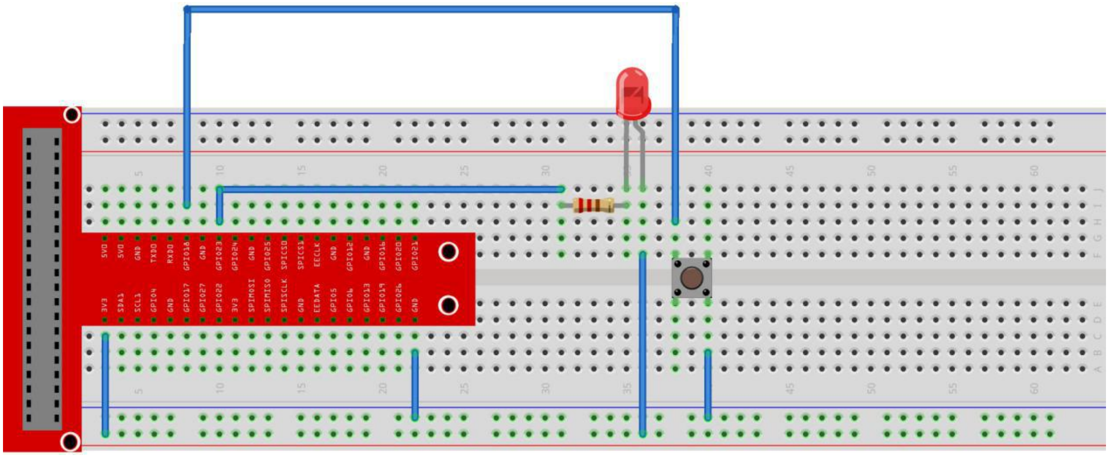
Product description:

● function：Buttons are a common component used to control electronic devices. They are usually used as switches to connect or disconnect circuits.

● Application: For example, any button on the mobile phone, the backlight will light up



Wiring diagram:



C code：

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

#define LedPin     4
#define ButtonPin  1

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

    pinMode(LedPin, OUTPUT);
    pinMode(ButtonPin, INPUT);
    // Pull up to 3.3V,make GPIO1 a stable level
    pullUpDnControl(ButtonPin, PUD_UP);

    digitalWrite(LedPin, HIGH);
    printf("LED off\n");

    while(1){
        // Indicate that button has pressed down
        if(digitalRead(ButtonPin) == 0){
            // Led on
            digitalWrite(LedPin, LOW);
            printf("LED on\n");
        }
        else{
            // Led off
            digitalWrite(LedPin, HIGH);
            printf("LED off\n");
        }
    }
    return 0;
}
```

Python code：

```
#!/usr/bin/env python

import RPi.GPIO as GPIO
import time
# Set #23 as LED pin
LedPin = 23
# Set #18 as button pin
ButtonPin = 18

# Set Led status to True(OFF)
Led_status = True

# Define a function to print message at the beginning
def print_message():
    print ("========================================")
    print ("|          Button control LED          |")
    print ("|         LED connect to GPIO23        |")
    print ("|        Button connect to GPIO18      |")
    print ("|                                      |")
    print ("|   Press button to turn on/off LED.   |")
    print ("|                                      |")
    print ("========================================\n")
    print ’Program is running...’


# Define a setup function for some setup
def setup():
    # Set the GPIO modes to BCM Numbering
    GPIO.setmode(GPIO.BCM)
    # Set LedPin’s mode to output,
    # and initial level to high (3.3v)
    GPIO.setup(LedPin, GPIO.OUT, initial=GPIO.HIGH)
    # Set BtnPin’s mode to input,
    # and pull up to high (3.3V)
    GPIO.setup(ButtonPin, GPIO.IN, pull_up_down=GPIO.PUD_UP)
    # Set up a falling detect on BtnPin,
    # and callback function to swLed
    GPIO.add_event_detect(ButtonPin, GPIO.FALLING, callback=swLed)

# Define a callback function for button callback
def swLed(ev=None):
    global Led_status
    # Switch led status(on-->off; off-->on)
    Led_status = not Led_status
    GPIO.output(LedPin, Led_status)
    if Led_status:
        print ’LED OFF’


    else:
        print ’LED ON’



# Define a main function for main process
def main():
    # Print messages
    print_message()
    while True:
        # Don’t do anything.
        time.sleep(1)

# Define a destroy function for clean up everything after
# the script finished
def destroy():
    # Turn off LED
    GPIO.output(LedPin, GPIO.HIGH)
    # Release resource
    GPIO.cleanup()

# If run this script directly, do:
if __name__ == ’__main__’:
    setup()
    try:
        main()
    # When ’Ctrl+C’ is pressed, the child program
    # destroy() will be  executed.
    except KeyboardInterrupt:
        destroy()
```

Experimental results:

In the directory where the code file is located, execute the following command

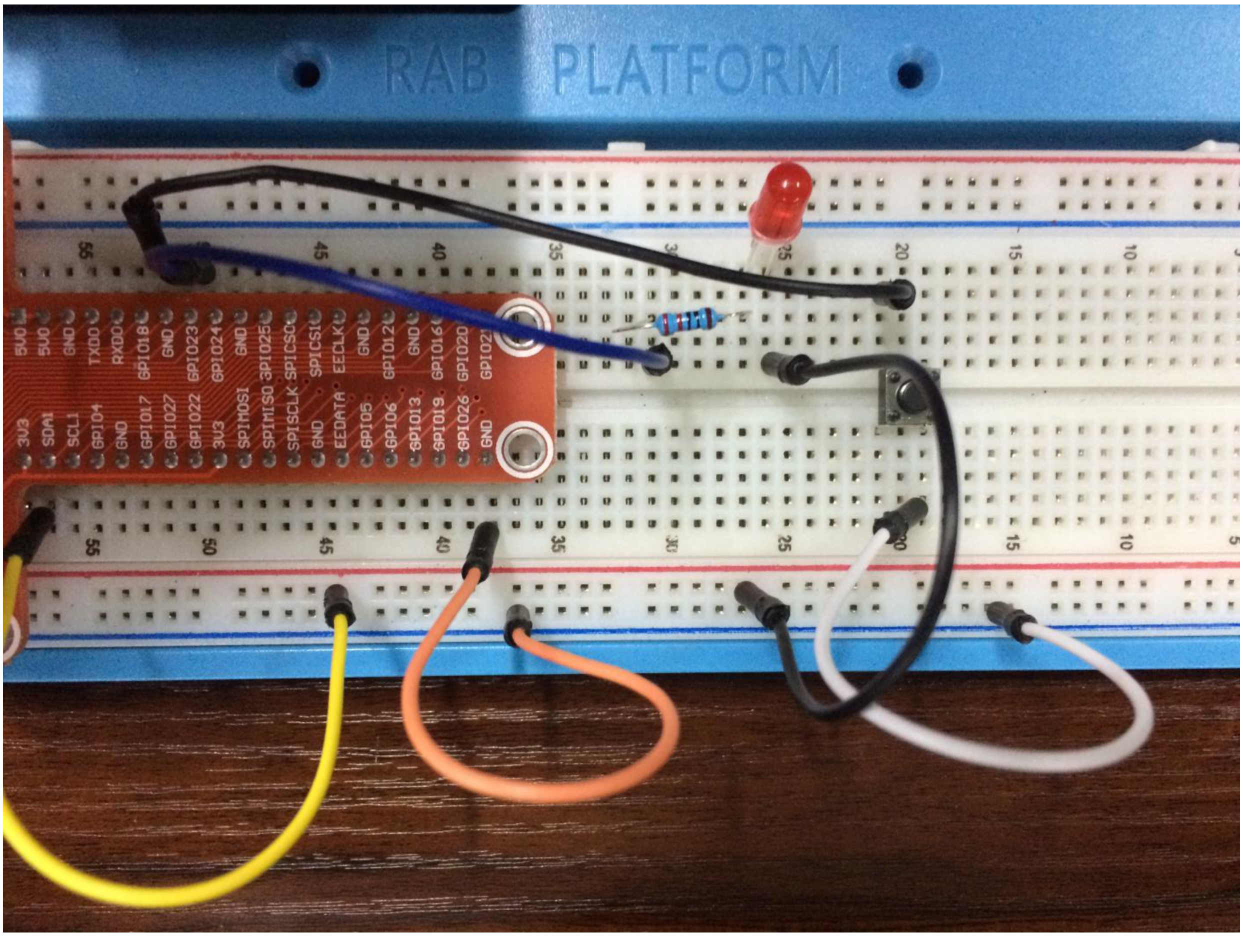
C：

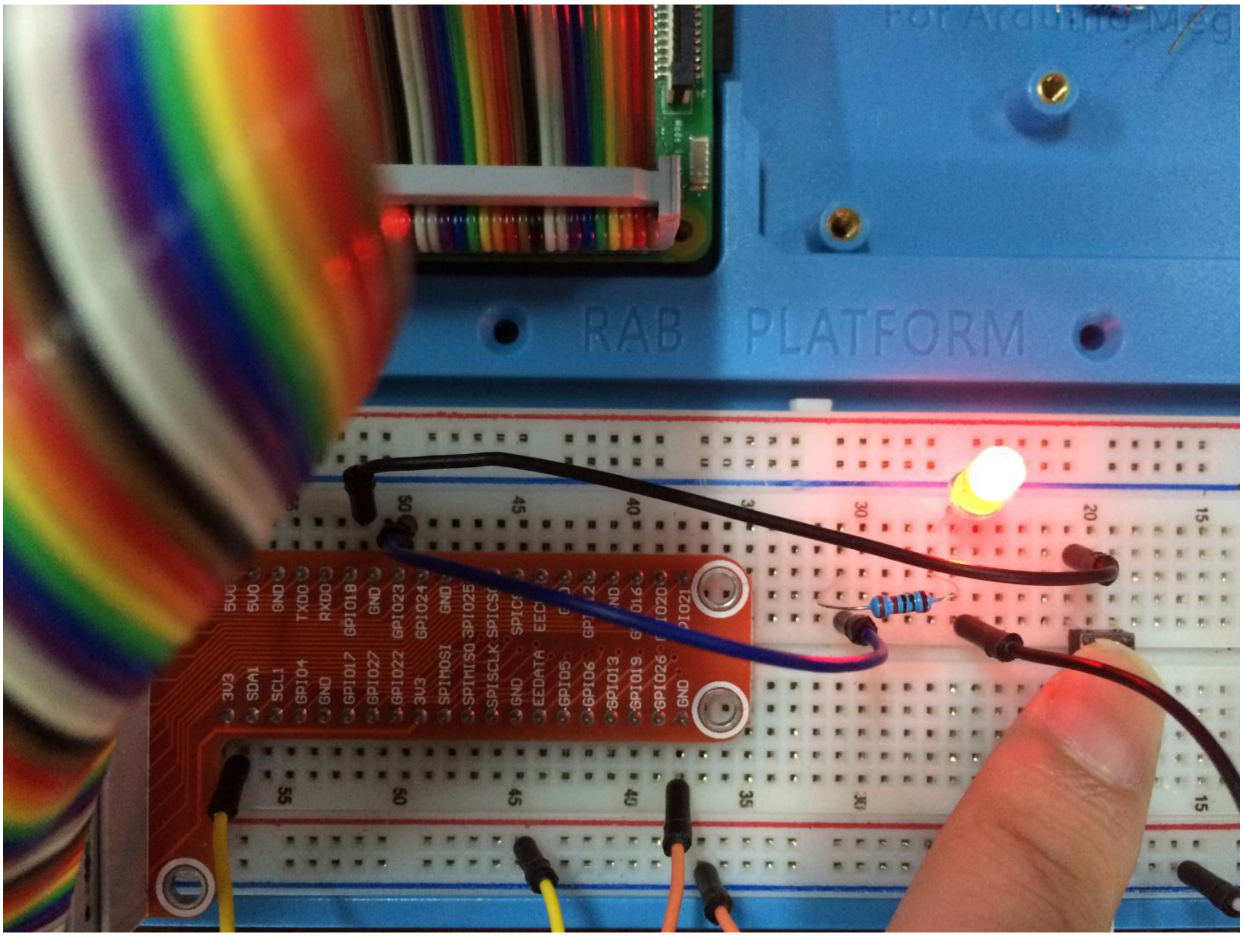
```
gcc -Wall -o button button.c -lwiringPi
sudo ./button
```

Python:

```
python button.py
```

After the instruction is executed, press the button to control the on and off of the led lamp




RAB PLATFORM
For Arduino Meg
5V0
5V0
GND
TXD0
RXD0
GPIO18
GND
GPIO23
GPIO24
GND
SPIMISO
3PIO25
SPICS0
3V3
SDA1
SCL1
GPIO4
GND
GPIO17
GPIO27
GPIO22
3V3
SPIMOSI
SPISCLK
GND
EEDATA
GPIO5
GPIO6
GPIO13
GPIO19
GPIO26
GND
GPIO20
GPIO21
55
50
45
40
35
30
20
15