HC SR501

Overview

In this project, we will make a sense LED, with the human body infrared pyroelectric sensors.When the sensor detects someone passing it, the LED lights up or goes out.

Experimental Materials:

Raspberry Pi *1

5mm red LED light *1

T-type expansion board *1

220 ohm resistor *1

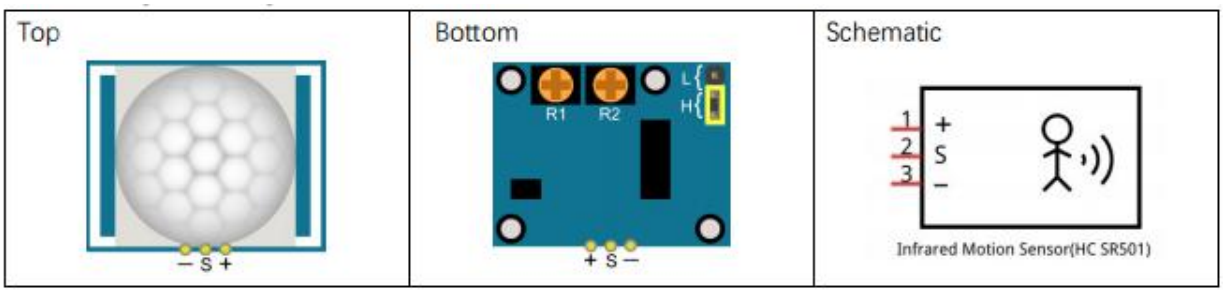
Breadboard*1

HC SR501 *1

Some DuPont lines

Product description：

The following is the diagram of infrared Motion sensor（HC SR-501）:



Technical Parameters:

1. Working voltage: 5v-20v(DC） Static current: 65uA.

2. Automatic trigger. When the body enter into the active area of sensor, the module will output high level (3.3V). When body leave out, it will output high level lasting for time T, then output low level(0V). Delay time T can be adjusted by potentiometer R1.

3. According to the position of jumper cap, you can choose non-repeatable trigger mode or repeatable mode.

L: non-repeatable trigger mode. The module output high level after sensing body, then when delay time is over, the module will output low level. And during high level time, the sensor doesn’t sense body anymore.

H: repeatable trigger mode. The distinction from L is that it can sense body until body leaves during the period of outputting high level. And then it starts to time and output low level after delaying T time.

4. Induction block time: the induction will stay in block condition and do not induce external signal at a little time (less than delay time) after

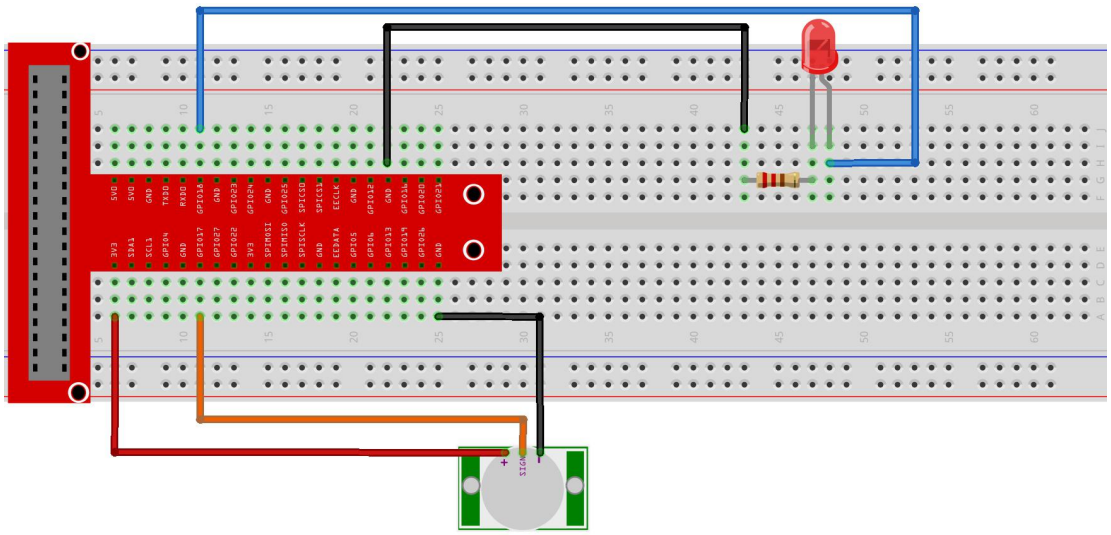
outputting high level or low level

5. Initialization time: the module needs about 1 minute to initialize after powered on. During this period, it will output high or low level alternately.

6. In consideration of feature of this sensor, when body get close or away from edgewise side, the sensor will work with high sensitively. When body get close or away in vertical direction, the sensor can't work well, which should get your attention. Sensing distance is adjusted by potentiometer(R2).

We can regard this sensor as a simple inductive switch when in use.

Wiring diagram：



C code：

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

#define ledPin    1    //define the ledPin
#define sensorPin 0    //define the sensorPin

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

    pinMode(ledPin, OUTPUT);
    pinMode(sensorPin, INPUT);

    while(1){

        if(digitalRead(sensorPin) == HIGH){ //if read sensor for high level
            digitalWrite(ledPin, HIGH);   //led on
            printf("led on...\n");
        }
        else {
            digitalWrite(ledPin, LOW);   //led off
            printf("...led off\n");
        }
    }

    return 0;
}
```

Python code：

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

ledPin = 12    # define the ledPin
sensorPin = 11    # define the sensorPin

def setup():
```

```
    print ('Program is starting...')
    GPIO.setmode(GPIO.BOARD)        # Numbers GPIOs by physical location
    GPIO.setup(ledPin, GPIO.OUT)    # Set ledPin's mode is output
    GPIO.setup(sensorPin, GPIO.IN)    # Set sensorPin's mode is input

def loop():
    while True:
        if GPIO.input(sensorPin)==GPIO.HIGH:
            GPIO.output(ledPin,GPIO.HIGH)
            print ('led on ...')
        else :
            GPIO.output(ledPin,GPIO.LOW)
            print ('led off ...')

def destroy():
    GPIO.cleanup()                     # Release resource

if __name__ == '__main__':     # Program start from here
    setup()
    try:
        loop()
    except KeyboardInterrupt:  # When 'Ctrl+C' is pressed, the child program
destroy() will be  executed.
        destroy()
```

Experimental results:

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

C:

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

Python:

```
python SenseLED.py
```

After the program is executed, try to leave away from or get closed to the Motion Sensor Infrared and observe whether the LED will be turned on or off. The terminal window will print out the state of LED constantly. As is shown below: