

# Chapter10 循光行走

## 实验目的

远程登录树莓派系统之后，运行程序，按下按键K2，启动寻光行走功能，当两路光敏电阻均检测到有光时，前进；当左侧检测到有光时，向左转；当右侧检测到有光时，向右转；左右两侧均未检测到光线时，小车停止。

## 实验原理

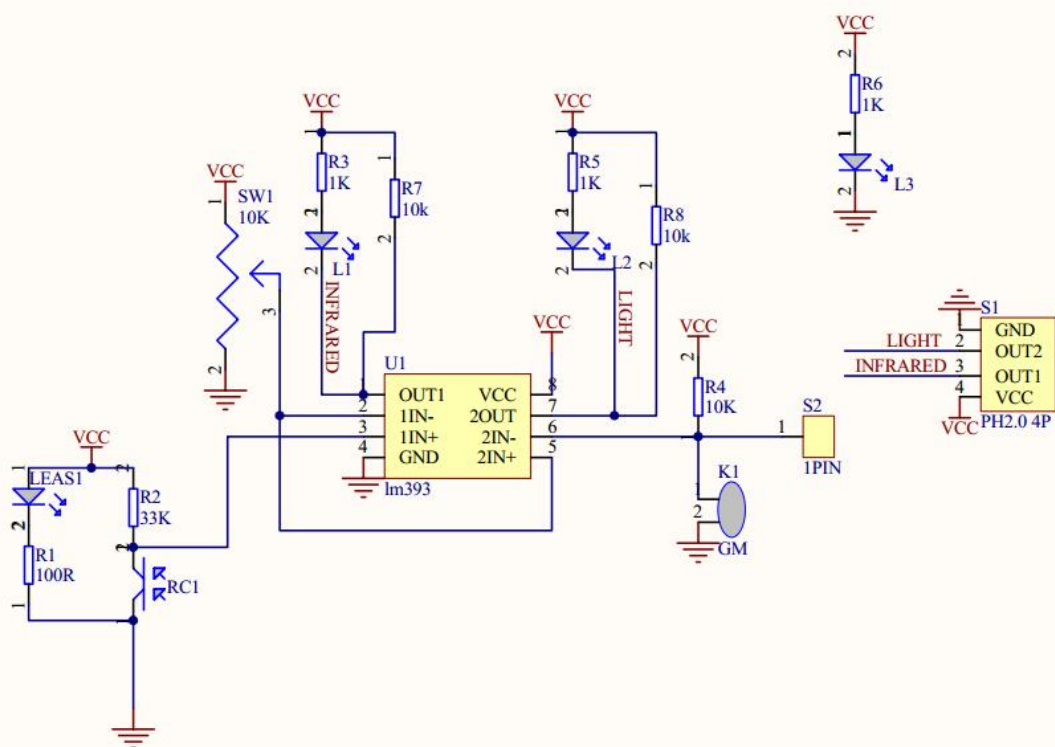
### 光敏原理

光敏电阻器是利用半导体的光电导效应制成的一种电阻值随入射光的强弱而改变的电阻器，又称为光电导探测器；入射光强，电阻减小，入射光弱，电阻增大。我们通过光敏电阻连接到主控板上的引脚的电平变化来决定是否有光。可通过电位器调整循光灵敏度。

### 硬件原理



图一 树莓派接口

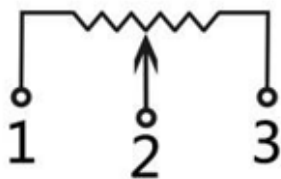


图二 光敏模块接口及电路

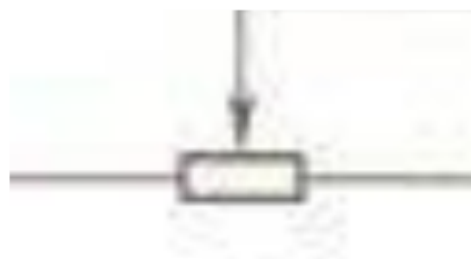
传感器模块提供了四个引脚，其中OUT1控制红外传感测距，OUT2控制循光。

我们来看电路图，可以看到\*\*1IN-和2IN+\*\*引脚都接到了同一个电路模块，该器件叫做电位器(类似于滑动变阻器)。可以用来调整红外测距时电平发生改变的距离。

电位器属于**无极性**器件，**可变电阻**的一种，三个触点，通过旋转旋钮改变2号脚的位置，从而改变阻值的大小，1脚和3脚分别接5V和GND，2脚接模拟输入引脚。



电位器



电路图符号

[https://blog.csdn.net/qq\\_2645653](https://blog.csdn.net/qq_2645653)

## 代码实现

```
#-*- coding:UTF-8 -*-
import RPi.GPIO as GPIO
import time
```

#小车电机引脚定义

```

AIN1 = 21
AIN2 = 20

BIN1 = 26
BIN2 = 19

PWMA = 16
PWMB = 13
#小车按键定义
key = 8

#光敏电阻引脚定义
LdrSensorLeft = 7
LdrSensorRight = 6

#设置GPIO口为BCM编码方式
GPIO.setmode(GPIO.BCM)

#忽略警告信息
GPIO.setwarnings(False)

#电机引脚初始化为输出模式
#按键引脚初始化为输入模式
#光敏电阻引脚初始化为输入模式
def init():
    global pwm_ENA
    global pwm_ENB
    GPIO.setup(PWMA,GPIO.OUT,initial=GPIO.HIGH)
    GPIO.setup(AIN2,GPIO.OUT,initial=GPIO.LOW)
    GPIO.setup(AIN1,GPIO.OUT,initial=GPIO.LOW)
    GPIO.setup(PWMB,GPIO.OUT,initial=GPIO.HIGH)
    GPIO.setup(BIN2,GPIO.OUT,initial=GPIO.LOW)
    GPIO.setup(BIN1,GPIO.OUT,initial=GPIO.LOW)
    GPIO.setup(key,GPIO.IN)
    GPIO.setup(LdrSensorLeft,GPIO.IN)
    GPIO.setup(LdrSensorRight,GPIO.IN)
    #设置pwm引脚和频率为2000hz
    pwm_PWMA = GPIO.PWM(PWMA, 2000)
    pwm_PWMB = GPIO.PWM(PWMB, 2000)
    pwm_ENA.start(0)
    pwm_ENB.start(0)

#延时2s
time.sleep(2)

#try/except语句用来检测try语句块中的错误,
#从而让except语句捕获异常信息并处理。
try:
    init()
    key_scan()
    while True:
        #遇到光线,寻光模块的指示灯灭,端口电平为HIGH
        #未遇光线,寻光模块的指示灯亮,端口电平为LOW
        LdrSensorLeftValue = GPIO.input(LdrSensorLeft)

```

```
LdrSensorRightValue = GPIO.input(LdrSensorRight)

if LdrSensorLeftValue == True and LdrSensorRightValue == True :
    run()          #两侧均有光时信号为HIGH, 光敏电阻指示灯灭, 小车前进
elif LdrSensorLeftValue == True and LdrSensorRightValue == False :
    spin_left()    #左边探测到有光, 有信号返回, 向左转
    time.sleep(0.002)
elif LdrSensorRightValue == True and LdrSensorLeftValue == False:
    spin_right()   #右边探测到有光, 有信号返回, 向右转
    time.sleep(0.002)
elif LdrSensorRightValue == False and LdrSensorLeftValue == False :
    brake()        #均无光, 停止

except KeyboardInterrupt:
    pass
pwm_ENA.stop()
pwm_ENB.stop()
GPIO.cleanup()
```