**Photosensitive Resistance**

Overview

In this course, we will use the Raspberry Pi to capture the signal of the photosensitive resistance module and to control the LED on or off according to the captured signals.

The Experimental Materials

Raspberry Pi \* 1

Breadboard \* 1

Photoresistance \* 1

ADC0832 \*1

Led \* 1

Dupont Line

The Preparatory Work

1. Install the python interpreter in your Raspberry Pi system.

2. Install the rpi.gpio library in your Raspberry Pi system.

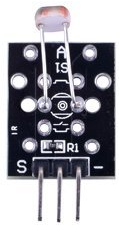
3. Install wiring Pi library in your Raspberry Pi system.

For more details, please refer to the attachment for installing the python interpreter and corresponding libraries in the Raspberry Pi system.

Product Description

Brief Introduction

Photoresistor is a type of resistance that changes with the change of light intensity and the resistance of the photosensitive resistor used in our experiment will decrease with the increase of light intensity. In this experiment, the photosensitive resistor and the 103 resistor in the module form a series circuit, and the S pin is connected between the photosensitive resistance and the 103 resistor. When the light attenuates, the resistance value of the photosensitive resistor will increase and the partial pressure of the photosensitive resistor becomes larger. As the photosensitive resistance is connected to the negative electrode, the potential of S pin increases and the amount of ADC conversion received by the Raspberry Pi increases. We can control the work of other devices by receiving this numeric value. Photosensitive resistors are mainly used in various photoelectric control systems, Such as the photoelectric automatic switch portal, the navigation lights, street lights, other lighting systems automatically light out and automatic feed water and automatic water shutoff device, the automatic protection device on the machine and the position detector, the thickness of the thin parts detector, automatic camera exposure device, photoelectric counter, smoke alarm, photoelectric tracking system, etc.



Characteristic Parameters:

◆Maximum voltage (V-dc) : 150

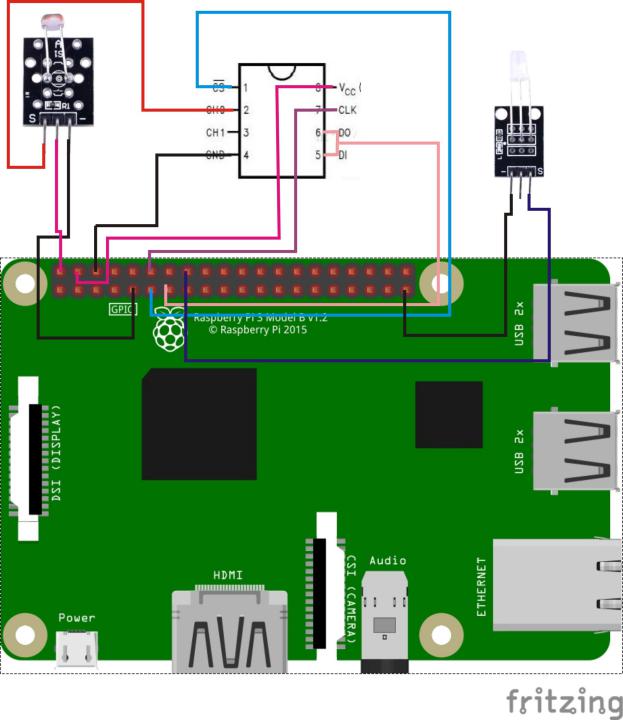
◆ Max Power (mW) : 100

◆Ambient Temperature (°C ): + 70-30

◆ Spectrum Peak (nm) : 540

◆Response Time (ms) : rise: 20 drops: 30

Wiring diagram



**The Sample Code**

1. **Python Code**

#!/usr/bin/env python

import RPi.GPIO as GPIO

import ADC0832

import time

LedPin = 16

threshold = 120

def init():

ADC0832.setup()

GPIO.setup(LedPin, GPIO.OUT)

def loop():

while True:

res = ADC0832.getResult(0)

print 'res = %d' % res

if(res > threshold):

print 'It is night, light on...'

GPIO.output(LedPin, GPIO.HIGH)

else:

print 'It is already dawn, light off'

GPIO.output(LedPin, GPIO.LOW)

time.sleep(0.2)

if \_\_name\_\_ == '\_\_main\_\_':

init()

try:

loop()

except KeyboardInterrupt:

ADC0832.destroy()

print 'The end !'

**2、C Code**

#include <wiringPi.h>

#include <stdio.h>

typedef unsigned char uchar;

typedef unsigned int uint;

#define ADC\_CS 0

#define ADC\_CLK 1

#define ADC\_DIO 2

#define LedPin 4

#define threshold 120

uchar get\_ADC\_Result(uchar channel)

{

uchar i;

uchar dat1=0, dat2=0;

digitalWrite(ADC\_CS, 0);

digitalWrite(ADC\_CLK,0);

digitalWrite(ADC\_DIO,1); delayMicroseconds(2);

digitalWrite(ADC\_CLK,1); delayMicroseconds(2);

digitalWrite(ADC\_CLK,0);

digitalWrite(ADC\_DIO,1); delayMicroseconds(2);

digitalWrite(ADC\_CLK,1); delayMicroseconds(2);

digitalWrite(ADC\_CLK,0);

digitalWrite(ADC\_DIO,channel); delayMicroseconds(2);

digitalWrite(ADC\_CLK,1);

digitalWrite(ADC\_DIO,1); delayMicroseconds(2);

digitalWrite(ADC\_CLK,0);

digitalWrite(ADC\_DIO,1); delayMicroseconds(2);

for(i=0;i<8;i++)

{

digitalWrite(ADC\_CLK,1); delayMicroseconds(2);

digitalWrite(ADC\_CLK,0); delayMicroseconds(2);

pinMode(ADC\_DIO, INPUT);

dat1=dat1<<1 | digitalRead(ADC\_DIO);

}

for(i=0;i<8;i++)

{

dat2 = dat2 | ((uchar)(digitalRead(ADC\_DIO))<<i);

digitalWrite(ADC\_CLK,1); delayMicroseconds(2);

digitalWrite(ADC\_CLK,0); delayMicroseconds(2);

}

digitalWrite(ADC\_CS,1);

pinMode(ADC\_DIO, OUTPUT);

return(dat1==dat2) ? dat1 : 0;

}

int main(void)

{

uchar analogVal;

if(wiringPiSetup() == -1)

{

printf("setup wiringPi failed !");

return -1;

}

pinMode(ADC\_CS, OUTPUT);

pinMode(ADC\_CLK, OUTPUT);

pinMode(ADC\_DIO, OUTPUT);

pinMode(LedPin, OUTPUT);

while(1)

{

analogVal = get\_ADC\_Result(0);

printf("analogVal is %d.\n", analogVal);

if(analogVal > threshold)

{

printf("It is night,light on!\n");

digitalWrite(LedPin, HIGH);

}

else

{

printf("It is already dawn, light off!\n");

digitalWrite(LedPin, LOW);

}

delay(200);

}

return 0;

}

**The Experimental Phenomena**

The LED light will be lit when the light intensity detected by the Photoresistor is lower than the set threshold. The LED light will be extinguished when the light intensity detected by the Photoresistor is higher than the set threshold.