3-Color led

Overview

This course will use the Raspberry Pi GPIO pin to control the 3-Color led lamp.

Experimental Materials

RaspberryPi \*1

breadboard \*1

3-color led \*1

Dupont Line

Ready to work

1. Install python interpreter in your Raspberry Pi system

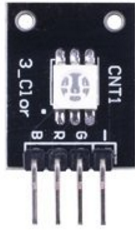
2. Install the RPi.GPIO library in your Raspberry Pi system

3. Install the wiringPi library in your Raspberry Pi system

Refer to the attached "Installing a Python Interpreter and Corresponding Libraries in a Raspberry Pi System" for details.

product description

The 3-color LED module is made of a LED that patch full-color. When you adjust the PWM voltage input of R, G and B pins, the intensity of the three primary colors (red/blue/green) will be changed, you could get full-color effect. It is often used in KTV, stage, building and other environments. You can use the Raspberry Pi to control the module for cool lighting effects.



characteristic parameters

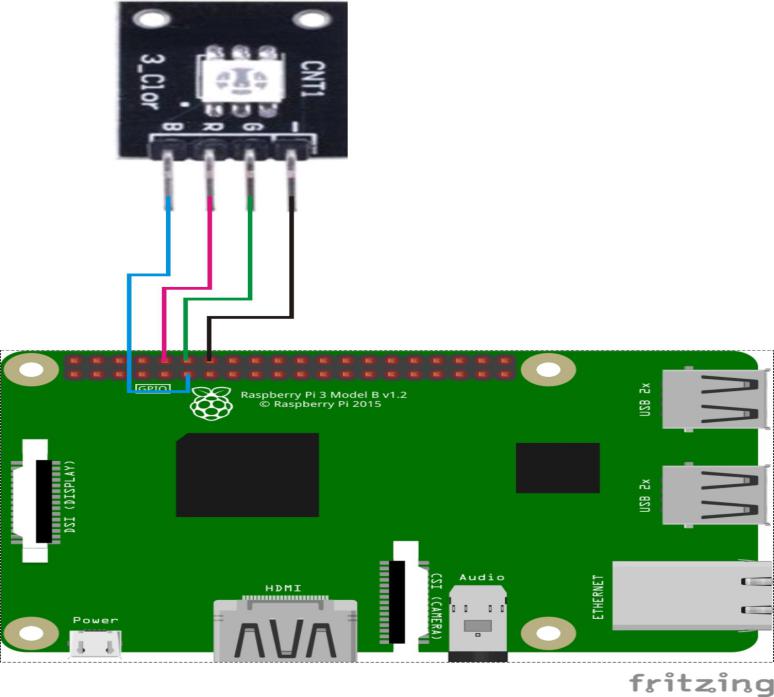
◆ use full color LED patch

◆ Adjust the three primary colors by PWM can get different colors

◆Working voltage: 5V

◆Drive mode: Common cathode drive

Wiring diagram



Sample code

1. python code

#!/usr/bin/env python # to declare Interpreter that this script used

import RPi.GPIO as GPIO # import library “RPi.GPIO”

import time #import library “time”

colors = [0xFF0000, 0x00FF00, 0x0000FF, 0xFFFF00, 0xFF00FF, 0x00FFFF]

pins = {'pin\_R':10, 'pin\_G':12, 'pin\_B':11} # pins is a dict

GPIO.setmode(GPIO.BOARD) # Numbers GPIOs by physical location

for i in pins:

GPIO.setup(pins[i], GPIO.OUT) # Set pins' mode is output

p\_R = GPIO.PWM(pins['pin\_R'], 2000) # set Frequece to 2KHz

p\_G = GPIO.PWM(pins['pin\_G'], 2000)

p\_B = GPIO.PWM(pins['pin\_B'], 2000)

p\_R.start(0) # Initial duty Cycle = 0(leds off)

p\_G.start(0)

p\_B.start(0)

def map(x, in\_min, in\_max, out\_min, out\_max):

return (x - in\_min) \* (out\_max - out\_min) / (in\_max - in\_min) + out\_min

def setColor(col):

R\_val = (col & 0xFF0000) >> 16

G\_val = (col & 0x00FF00) >> 8

B\_val = (col & 0x0000FF) >> 0

R\_val = map(R\_val, 0, 255, 0, 100)

G\_val = map(G\_val, 0, 255, 0, 100)

B\_val = map(B\_val, 0, 255, 0, 100)

p\_R.ChangeDutyCycle(R\_val) # Change duty cycle

p\_G.ChangeDutyCycle(G\_val)

p\_B.ChangeDutyCycle(B\_val)

try:

while True:

for col in colors:

setColor(col)

time.sleep(0.5)

except KeyboardInterrupt:

p\_R.stop()

p\_G.stop()

p\_B.stop()

for i in pins:

GPIO.output(pins[i], GPIO.HIGH) # Turn off all leds

GPIO.cleanup()

1. C code

#include <wiringPi.h>

#include <softPwm.h>

#include <stdio.h>

typedef unsigned char uchar;

#define LedPinRed 16

#define LedPinGreen 1

#define LedPinBlue 0

void ledInit(void)

{

softPwmCreate(LedPinRed, 0, 100);

softPwmCreate(LedPinGreen,0, 100);

softPwmCreate(LedPinBlue, 0, 100);

}

uchar map(uchar val, uchar in\_min, uchar in\_max, uchar out\_min, uchar out\_max)

{

uchar tmp = 0;

tmp = (val - in\_min) \* (out\_max - out\_min) / (in\_max - in\_min) + out\_min;

return tmp;

}

void ledColorSet(uchar r\_val, uchar g\_val, uchar b\_val)

{

uchar R\_val, G\_val, B\_val;

R\_val = map(r\_val, 0, 255, 0, 100);

G\_val = map(g\_val, 0, 255, 0, 100);

B\_val = map(b\_val, 0, 255, 0, 100);

softPwmWrite(LedPinRed, R\_val);

softPwmWrite(LedPinGreen, G\_val);

softPwmWrite(LedPinBlue, B\_val);

}

int main(void)

{

int i;

if(wiringPiSetup() == -1)

{

printf("setup wiringPi failed !");

return 1;

}

ledInit();

while(1)

{

ledColorSet(0xff,0x00,0x00); //red

delay(1000);

ledColorSet(0x00,0xff,0x00); //green

delay(1000);

ledColorSet(0x00,0x00,0xff); //Blue

delay(1000);

ledColorSet(0xff,0xff,0x00); //yellow

delay(1000);

ledColorSet(0xff,0x00,0xff); //Magenta

delay(1000);

ledColorSet(0x00,0xff,0xff); //Cyan

delay(1000);

}

return 0;

}

Experimental phenomena

Three-color LED lights emit red, green, blue, yellow, magenta, and cyan lights.