RGB

Overview

This lesson will use the RGB module to emit 6 colors of light in sequence

Experimental Materials:

Raspberry Pi *1

T-type expansion board *1

Breadboard *1

RGB *1

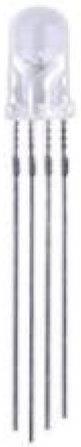
220Ω resistor *3

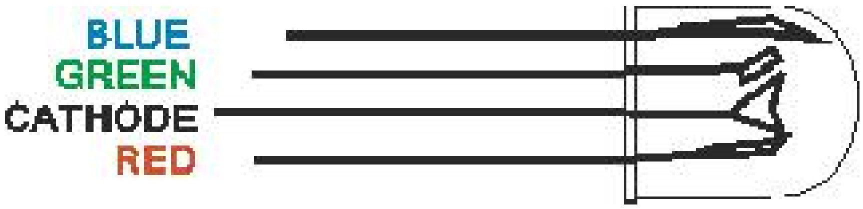
Some DuPont lines

Product description:

● function; three-color RGB module is composed of red, green, blue three led lights, through the color intensity of the three lights can show all the colors.

● Application: Current monitors mostly use the RGB color standard.





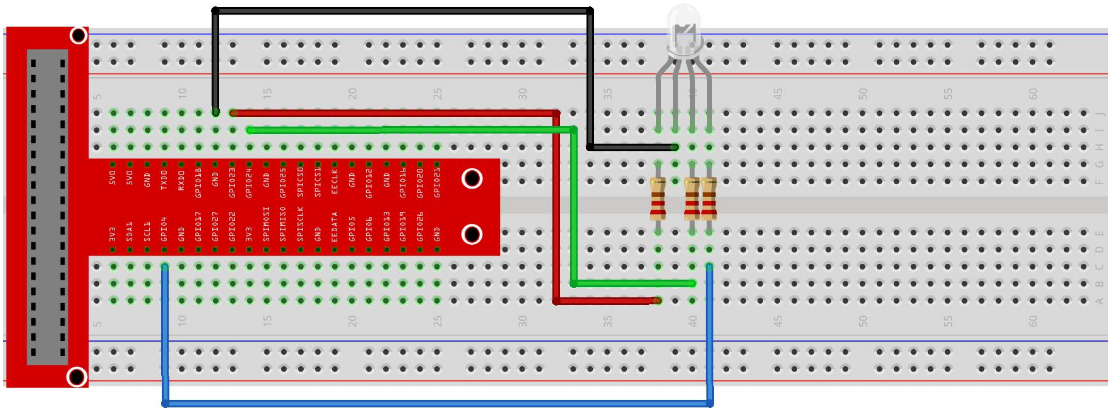
Technical Parameters:

1. Working voltage: 3.3V/5V

2.can access a variety of microcontroller

3. High light LED

Wiring diagram：



C code:

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

#define uchar unsigned char

#define LedPinRed    4
#define LedPinGreen  5
#define LedPinBlue   7

void ledInit(void){
    softPwmCreate(LedPinRed,  0, 100);
    softPwmCreate(LedPinGreen,0, 100);
    softPwmCreate(LedPinBlue, 0, 100);
}

void ledColorSet(uchar r_val, uchar g_val, uchar b_val){
    softPwmWrite(LedPinRed,   r_val);
    softPwmWrite(LedPinGreen, g_val);
    softPwmWrite(LedPinBlue,  b_val);
}

int main(void){

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

    ledInit();
    printf(″\n″);
    printf(″\n″);
    printf(″========================================\n″);
    printf(″|                 RGB                  |\n″);
    printf(″|    ------------------------------    |\n″);
    printf(″|       Red Pin connect to GPIO4       |\n″);
    printf(″|      Green Pin connect to GPIO5      |\n″);
    printf(″|       Blue Pin connect to GPIO7      |\n″);
    printf(″|                                      |\n″);
    printf(″|  Make a RGB LED emits various color  |\n″);
    printf(″========================================\n″);
    printf(″\n″);
    printf(″\n″);

    while(1){
        printf(″Red\n″);
        ledColorSet(0xff,0x00,0x00);   //red
        delay(500);
        printf(″Green\n″);
        ledColorSet(0x00,0xff,0x00);   //green
        delay(500);
        printf(″Blue\n″);
        ledColorSet(0x00,0x00,0xff);   //blue
        delay(500);

        printf(″Yellow\n″);
        ledColorSet(0xff,0xff,0x00);   //yellow
        delay(500);
        printf(″Purple\n″);
        ledColorSet(0xff,0x00,0xff);   //purple
        delay(500);
        printf(″Cyan\n″);
        ledColorSet(0xc0,0xff,0x3e);   //cyan
        delay(500);
    }

    return 0;
}
```

Python code：

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

import RPi.GPIO as GPIO
import time

# Set up a color table in Hexadecimal
COLOR = [0xFF0000, 0x00FF00, 0x0000FF, 0xFFFF00, 0xFF00FF, 0x00FFFF]
# Set pins’ channels with dictionary
pins = {’Red’:23, ’Green’:24, ’Blue’:4}

def print_message():
    print (″========================================″)
    print (″|             RGB LED                  |″)
    print (″|    ------------------------------    |″)
    print (″|       Red Pin connect to GPIO23      |″)
    print (″|       Green Pin connect to GPIO24    |″)
    print (″|       Blue Pin connect to GPIO4      |″)
    print (″|                                      |″)
    print (″|  Make a RGB LED emits various color  |″)
    print (″========================================\n″)
    print ’Program is running...’
    print ’Please press Ctrl+C to end the program...’
    raw_input (″Press Enter to begin\n″)

def setup():
    global p_R, p_G, p_B
    # Set the GPIO modes to BCM Numbering
    GPIO.setmode(GPIO.BCM)
    # Set all LedPin’s mode to output,
    # and initial level to High(3.3v)
    for i in pins:
        GPIO.setup(pins[i], GPIO.OUT, initial=GPIO.HIGH)

    # Set all led as pwm channel,
    #  and frequece to 2KHz
    p_R = GPIO.PWM(pins[’Red’], 2000)
    p_G = GPIO.PWM(pins[’Green’], 2000)
    p_B = GPIO.PWM(pins[’Blue’], 2000)

    # Set all begin with value 0
    p_R.start(0)
    p_G.start(0)
    p_B.start(0)
```

```
# Define a MAP function for mapping values.
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

# Define a function to set up colors
# input color should be Hexadecimal with
# red value, blue value, green value.
def setColor(color):
    # Devide colors from ’color’ veriable
    R_val = (color & 0xFF0000) >> 16
    G_val = (color & 0x00FF00) >> 8
    B_val = (color & 0x0000FF) >> 0

    # Map color value from 0~255 to 0~100
    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)

    # Change the colors
    p_R.ChangeDutyCycle(R_val)
    p_G.ChangeDutyCycle(G_val)
    p_B.ChangeDutyCycle(B_val)

    print ″color_msg: R_val = %s,  G_val = %s, B_val = %s″%(R_val, G_val,
B_val)

def main():
    print_message()
    while True:
        for color in COLOR:
            setColor(color)
            time.sleep(0.5)

def destroy():
    # Stop all pwm channel
    p_R.stop()
    p_G.stop()
    p_B.stop()
    # Turn off all LEDs
    GPIO.output(pins, 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

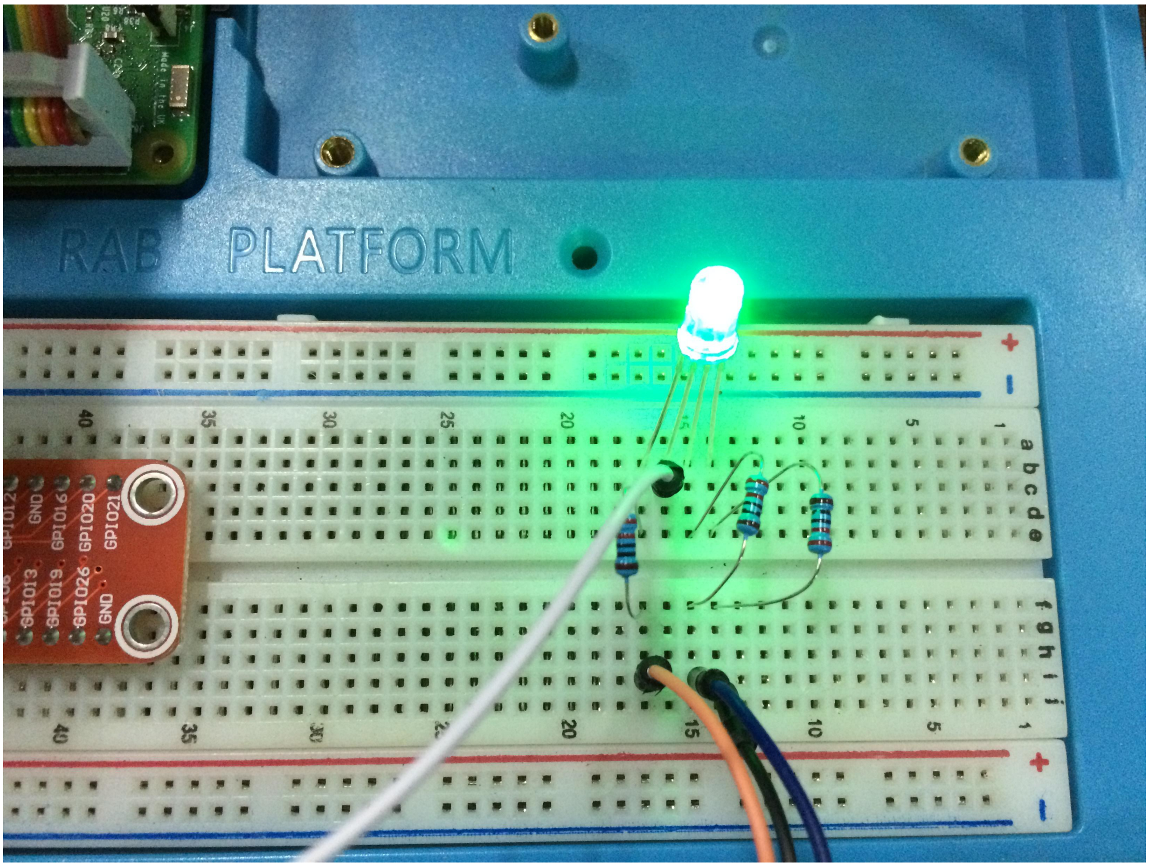
C：

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

Python:

```
python rgb.py
```

After the instruction is executed, RGB emits six colors of light in sequence


RAB PLATFORM
GPIO12
GND
GPIO16
GPIO20
GPIO21
GPIO13
GPIO19
GPIO26
GND
40
35
30
25
20
15
10
5
1
a b c d e
f g h i j
+
-