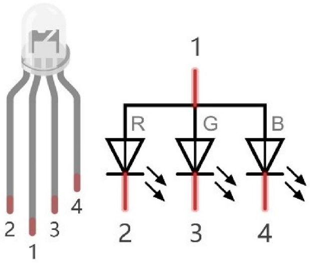
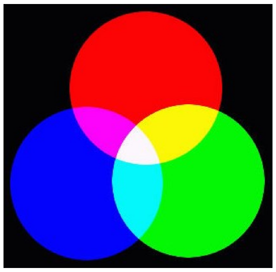
Chapter 5 RGBLED

In this chapter, we will learn how to control a RGBLED.
RGB LED has integrated 3 LEDs that can respectively emit red, green and blue light. And it has 4 pins. The long pin (1) is the common port, that is, 3 LED 's positive or negative port. The RGB LED with common positive port and its symbol are shown below. We can make RGB LED emit various colors of light by controlling these 3 LEDs to emit light with different brightness,



Red, green, and blue light are called 3 primary colors. When you combine these three primary-color light with different brightness, it can produce almost all kinds of visible lights. Computer screens, single pixel of cell phone screen, neon, and etc. are working under this principle.



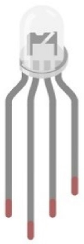
RGB

If we use three 8 bit PWM to control the RGBLED, in theory, we can create $2^8*2^8*2^8$=16777216 (16 million) color through different combinations.
Next, we will use RGBLED to make a colorful LED.

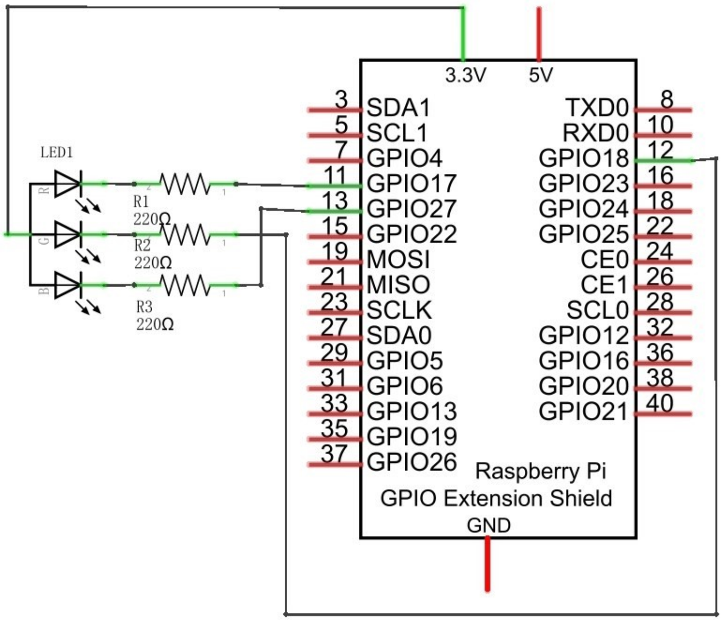
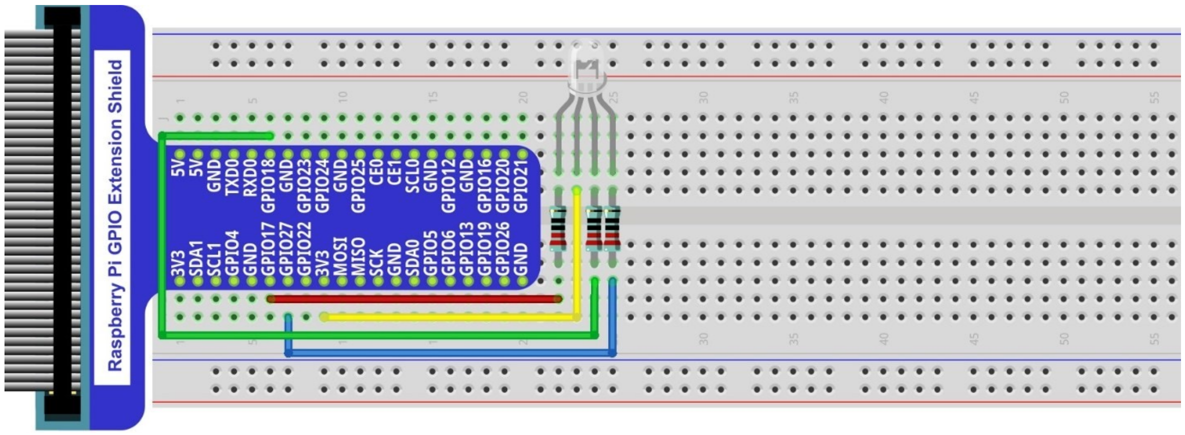
Project 5.1 Colorful LED

In this project, we will make a colorful LED. And we can control RGBLED to switch different colors automatically.

Component List

Raspberry Pi 3B x1 GPIO Extension Board & Wire x1 BreadBoard x1	RGBLED x1 	Resistor 220Ω x3
Jumper 		

Circuit

Schematic diagram 
Hardware connection. If you need any support, pleasefeel free to contact us via: **support@freenove.com** 

Code

Since this project requires 3 PWM, but in RPi, only one GPIO has the hardware capability to output PWM, we need to use the software to make the ordinary GPIO output PWM.

C Code 5.1.1 ColorfulLED

First observe the project result, and then analyze the code.

1. Use cd command to enter 05.1.1_ColorfulLED directory of C code.

```
cd ~/Freenove_Ultimate_Starter_Kit_for_Raspberry_Pi/Code/C_Code/05.1.1_ColorfulLED
```

2. Use following command to compile "ColorfulLED.c" and generate executable file "ColorfulLED". Note: in this project, the software PWM uses a multi-threading mechanism. So "-lpthread" option need to be add the compiler.

```
gcc ColorfulLED.c -o ColorfulLED -lwiringPi -lpthread
```

3. And then run the generated by "ColorfulLED".

```
sudo ./ColorfulLED
```

After the program is executed, you will see that the RGBLED shows light of different color randomly.

The following is the program code:

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

#define ledPinRed    0
#define ledPinGreen  1
#define ledPinBlue   2

void ledInit(void)
{
    softPwmCreate(ledPinRed,  0, 100);
    softPwmCreate(ledPinGreen,0, 100);
    softPwmCreate(ledPinBlue, 0, 100);
}

void ledColorSet(int r_val, int g_val, int b_val)
{
    softPwmWrite(ledPinRed,   r_val);
    softPwmWrite(ledPinGreen, g_val);
    softPwmWrite(ledPinBlue,  b_val);
}

int main(void)
{
    int r,g,b;
    if(wiringPiSetup() == -1){ //when initialize wiring failed,print message to screen
        printf("setup wiringPi failed !");
        return 1;
    }
    printf("Program is starting ...\n");
    ledInit();

    while(1){
        r=random()%100;
        g=random()%100;
        b=random()%100;
        ledColorSet(r,g,b);
        printf("r=%d,  g=%d,  b=%d \n",r,g,b);
        delay(300);
    }
    return 0;
}
```

First, in subfunction of ledInit(), create the software PWM control pins used to control the R, G, B pin respectively.

```
void ledInit(void)
{
    softPwmCreate(ledPinRed,  0, 100);
    softPwmCreate(ledPinGreen,0, 100);
    softPwmCreate(ledPinBlue, 0, 100);
}
```

Then create subfunction, and set the PWM of three pins.

```
void ledColorSet(int r_val, int g_val, int b_val)
{
    softPwmWrite(ledPinRed,   r_val);
    softPwmWrite(ledPinGreen, g_val);
    softPwmWrite(ledPinBlue,  b_val);
}
```

Finally, in the "while" cycle of main function, get three random numbers and specify them as the PWM duty cycle, which will be assigned to the corresponding pins. So RGBLED can switch the color randomly all the time.

```
    while(1){
        r=random()%100;
        g=random()%100;
        b=random()%100;
        ledColorSet(r,g,b);
        printf("r=%d,  g=%d,  b=%d \n",r,g,b);
        delay(300);
    }
```

The related function of Software PWM can be described as follws:

long random();
This function will return a random number.

For more details about Software PWM, please refer to: http://wiringpi.com/reference/software-pwm-library/

Python Code 5.1.1 ColorfulLED

First observe the project result, and then analyze the code.

1. Use cd command to enter 05.1.1_ColorfulLED directory of Python code.

```
cd ~/Freenove_Ultimate_Starter_Kit_for_Raspberry_Pi/Code/Python_Code/05.1.1_ColorfulLED
```

2. Use python command to execute python code “ColorfulLED.py”.

```
python ColorfulLED.py
```

After the program is executed, you will see that the RGBLED shows light of different color randomly.

The following is the program code:

```
import RPi.GPIO as GPIO
import time
import random

pins = {’pin_R’:11, ’pin_G’:12, ’pin_B’:13}  # pins is a dict

def setup():
    global p_R,p_G,p_B
    print (’Program is starting ... ’)
    GPIO.setmode(GPIO.BOARD)       # Numbers GPIOs by physical location
    for i in pins:
        GPIO.setup(pins[i], GPIO.OUT)   # Set pins’ mode is output
        GPIO.output(pins[i], GPIO.HIGH) # Set pins to high(+3.3V) to off led
    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
    p_G.start(0)
    p_B.start(0)

def setColor(r_val,g_val,b_val):
    p_R.ChangeDutyCycle(r_val)      # Change duty cycle
    p_G.ChangeDutyCycle(g_val)
    p_B.ChangeDutyCycle(b_val)

def loop():
    while True :
        r=random.randint(0,100)#get a random in (0,100)
        g=random.randint(0,100)
        b=random.randint(0,100)
        setColor(r,g,b)#set random as a duty cycle value
        print ('r=%d, g=%d, b=%d ' %(r ,g, b))
        time.sleep(0.3)

def destroy():
    p_R.stop()
    p_G.stop()
    p_B.stop()
    GPIO.cleanup()

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

In last chapter, we have learned how to use python language to make a pin output PWM. In this project, we let three pins output PWM, and the usage is exactly the same as last chapter. In the “while” cycle of “loop” function, we first obtain three random numbers, and then specify these three random numbers as the PWM value of the three pins.o that the RGBLED switching of different colors randomly.

```
def loop():
    while True :
        r=random.randint(0,100)
        g=random.randint(0,100)
        b=random.randint(0,100)
        setColor(r,g,b)
        print ('r=%d, g=%d, b=%d ' %(r ,g, b))
        time.sleep(0.3)
```

About function randint():

random.randint(a, b)
The function can returns a random integer within the specified range (a, b).