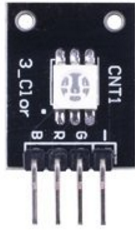
Three-Color LED  


Overview

The Three-Color LED is an RGB LED in a surface-mount device package (SMD). Like the glass bulb RGB LED, three-color LEDS contain three separate light-emitting diodes in Red, Green, and Blue colors. In different intensities, these three component colors fuse together to form millions of separately perceivable colors. In this experiment, you’ll learn to connect and program the three-color LED to produce a flashing sequence of six different colors.

Experimental Materials

Raspberry Pi x1

Breadboard x1

Three-color LED x1

Dupont jumper wires

Experimental Procedure

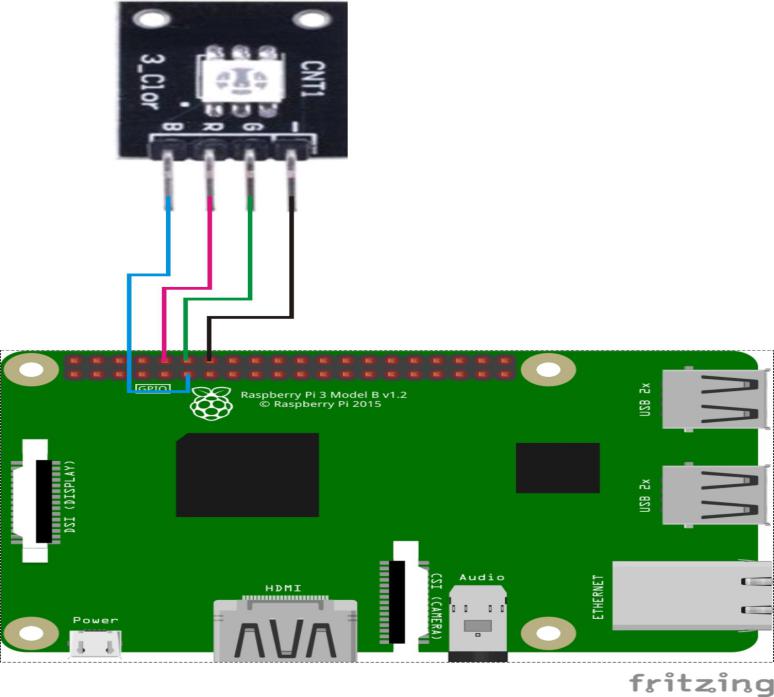
1. If you have not done so already, prepare your development system by installing the Python interpreter, RPi.GIO library, and wiringPi library as described in READ\_ME.TXT.
2. Install the three-color LED in your breadboard and use Dupont jumper wires to connect it to your Raspberry Pi as illustrated in the Wiring Diagram below. (The three-color LED module in this kit includes onboard series resistors, so no additional resistors are needed.)
3. Execute the sample stored in this experiment’s subfolder.

If using C, compile and execute the C code:  
cd Code/C  
gcc 3colorLED.c -o 3colorLED.out –lwiringPi  
./3colorLED.out

If using Python, launch the Python script:  
cd Code/Python  
python 3colorLED.py

1. Make experimental observations. The LED should alternately flash red, green, blue, yellow, magenta and cyan in a cycle.   
     
   The sample code uses pulse-width modulation (PWM) to vary the output voltage of the red, green, and blue GPIO pins, varying the intensity of light on the LED. In software, each individual component can be set to 101 different values (using Python’s ChangeDutyCycle() or C’s softPwmWrite() routines), so the LED can produce 1013—over a million—distinct color combinations. LEDs are ubiquitous illumination sources for their durability and energy efficiency; and RGB is an additive color model drawn from the human perception of colors and widely used in electronic color imaging.

Wiring Diagram



Three-color LED pin position:

R ↔ Raspberry Pi pin 10

G ↔ Raspberry Pi pin 12

B ↔ Raspberry Pi pin 11

- ↔ Raspberry Pi GND

Sample code

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()

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;

}