

# Lesson 9 Control RGB Light Flashing

## 1. Working Principle

In program, control the flashing frequency of RGB lights by modifying the delay time of on and off.

The source code of program is located in:

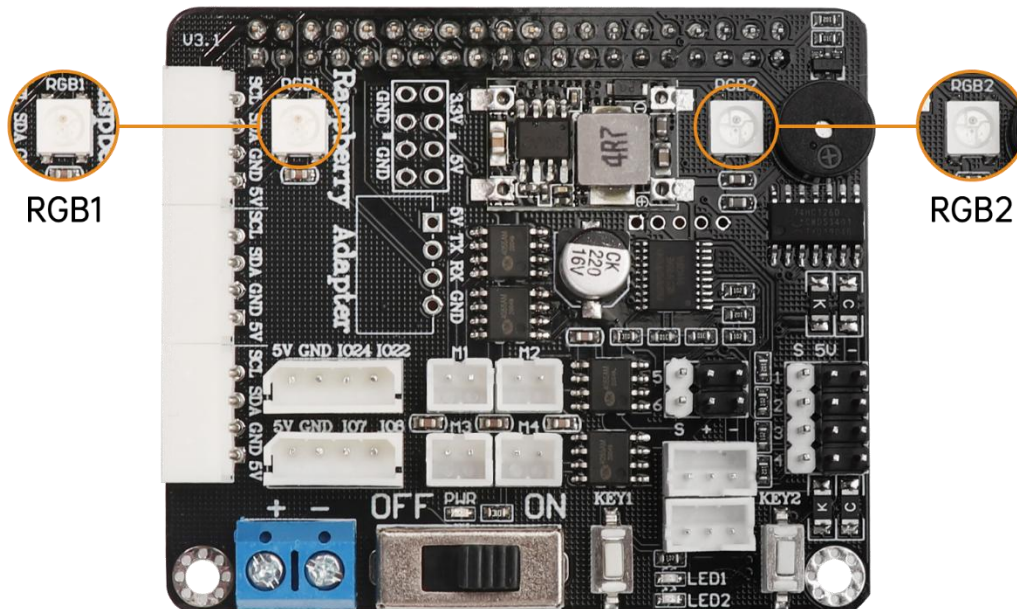
`/home/ubuntu/arduino_pro/src/arduino_pro_demo/expansion_board_demo/RGB_`

**Blink.py**

```
24 def start():
25     Board.RGB.setPixelColor(0, Board.PixelColor(255, 0, 0)) # Set the RGB1 on expansion board as red.
26     Board.RGB.setPixelColor(1, Board.PixelColor(255, 0, 0)) # Set the RGB2 on expansion board as red
27     Board.RGB.show()
28     time.sleep(1)
29
30     Board.RGB.setPixelColor(0, Board.PixelColor(0, 0, 0)) # Turn off the RGB1 on expansion board
31     Board.RGB.setPixelColor(1, Board.PixelColor(0, 0, 0)) # Turn off the RGB2 on expansion board
32     Board.RGB.show()
33     time.sleep(1)
34
35 for i in range(3):
36     Board.RGB.setPixelColor(0, Board.PixelColor(255, 0, 0)) # Set the RGB1 on expansion board as red.
37     Board.RGB.show()
38     time.sleep(0.5)
39     Board.RGB.setPixelColor(0, Board.PixelColor(0, 0, 0)) # Turn off the RGB1 on expansion board
40     Board.RGB.show()
41     time.sleep(0.5)
42
43     Board.RGB.setPixelColor(1, Board.PixelColor(255, 0, 0)) # Set the RGB2 on expansion board as red
44     Board.RGB.show()
45     time.sleep(0.5)
46     Board.RGB.setPixelColor(1, Board.PixelColor(0, 0, 0)) # Turn off the RGB2 on expansion board
47     Board.RGB.show()
48     time.sleep(0.5)
49
50     Board.RGB.setPixelColor(0, Board.PixelColor(255, 0, 0)) # Set the RGB1 on expansion board as red.
51     Board.RGB.setPixelColor(1, Board.PixelColor(255, 0, 0)) # Set the RGB2 on expansion board as red
52     Board.RGB.show()
53     time.sleep(1)
54
55     Board.RGB.setPixelColor(0, Board.PixelColor(0, 0, 0)) # Turn off the RGB1 on expansion board
56     Board.RGB.setPixelColor(1, Board.PixelColor(0, 0, 0)) # Turn off the RGB2 on expansion board
57     Board.RGB.show()
58     time.sleep(1)
```

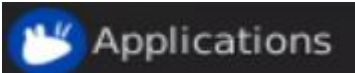

## 2. Preparation

There are two RGB lights on expansion board, as the figure shown below:



## 3. Operation Steps

1) Please refer to the tutorial in “6.Raspberry Pi and Expansion Board Lessons/2.Raspberry Pi Expansion Board/Lesson 4 Set Environment Development” to remotely connect through NoMachine.

2) Open the terminal. Click  in the lower left corner and select  to enter the terminal.

3) In the opened interface, enter the command “cd armpi\_pro/src/armpi\_pro\_demo/expansion\_board\_demo/” to access to game programmings directory and then press “Enter”.

```
ubuntu@ubuntu:~$ cd armpi_pro/src/armpi_pro_demo/expansion_board_demo/
```

4) Then enter command “sudo python3 BusServo\_Speed.py” to start the

game and press “Enter”.

```
ubuntu@ubuntu:~/armpi_pro/src/armpi_pro_demo/expansion_board_demo$  
sudo python3 RGB_Blink.py
```

5) If want to exit the program, you can press “Ctrl+C”. If fail to exit, you can try multiple times.

## 4. Project Outcome

After running program, two TGB lights on Raspberry Pi expansion board will enter the following status circularly.

Status	Outcome
State 1	Two RGB lights flash once at the same time. The RGB light color is red and its flashing is at a frequency of on and off with 1-second intervals.
State 2	Two RGB lights flash third times in turns. The RGB light color is red and its flashing is at a frequency of on and off with 0.5-second intervals.
Status 3	Two RGB lights flash once at the same time. The RGB light color is red and its flashing is at a frequency of on and off with 1-second intervals.