Lesson 10 Control RGB Light Flashing Color

1. Working Principle

RGB represents red, green and blue three colors and the parameter of three colors ranges from 0 to 255. The color can be changed by setting RGB color parameter. The source code of program is located in:

/home/ubuntu/armpi_pro/src/armpi_pro_demo/expansion_board_demo/RGB_Change.py

```
#Turn off all lights
Board.RGB.setPixelColor(0, Board.PixelColor(0, 0, 0))
Board.RGB.setPixelColor(1, Board.PixelColor(0, 0, 0))
Board.RGB.show()

signal.signal(signal.SIGINT, Stop)

while True:
    #Set two lights as red. 0 is RGB1 and 1 is RGB2.
    Board.RGB.setPixelColor(0, Board.PixelColor(255, 0, 0))
    Board.RGB.setPixelColor(1, Board.PixelColor(255, 0, 0))
    Board.RGB.show()
    time.sleep(1)

    #Set two lights as green. 0 is RGB1 and 1 is RGB2.
    Board.RGB.setPixelColor(0, Board.PixelColor(0, 255, 0))
    Board.RGB.setPixelColor(1, Board.PixelColor(0, 255, 0))
    Board.RGB.show()
    time.sleep(1)

    #Set two lights as blue. 0 is RGB1 and 1 is RGB2.
    Board.RGB.setPixelColor(0, Board.PixelColor(0, 0, 255))
    Board.RGB.setPixelColor(1, Board.PixelColor(0, 0, 255))
    Board.RGB.show()
    time.sleep(1)

    if not start:
        #Turn off all lights
        Board.RGB.setPixelColor(0, Board.PixelColor(0, 0, 0))
        Board.RGB.setPixelColor(1, Board.PixelColor(0, 0, 0))
        Board.RGB.show()
        print('Closed')
        break
```

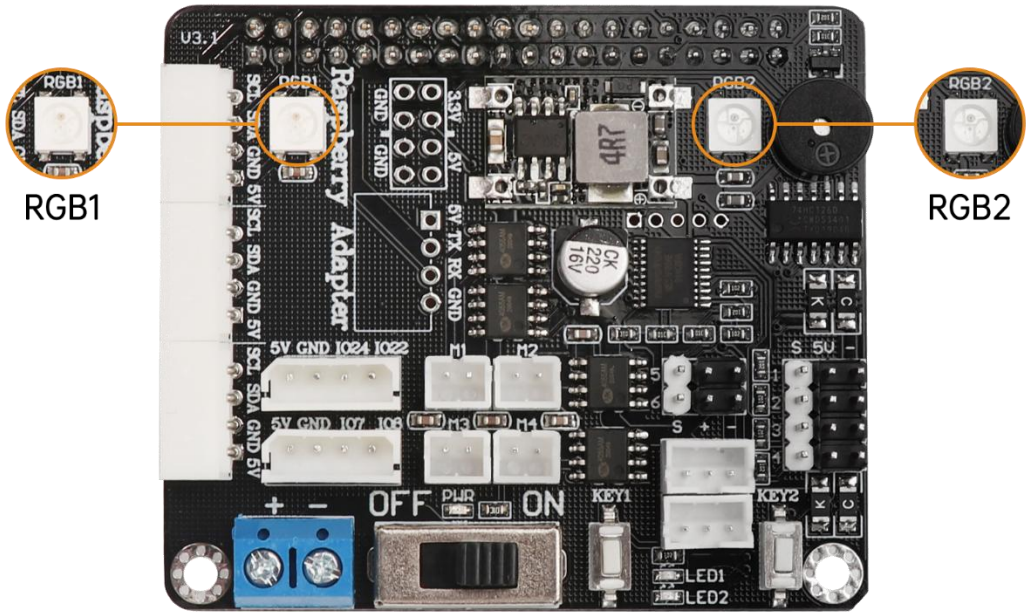
Control RGB light by calling Board.RGB.setPixelColor() function in Board library. Take the code "**Board.RGB.setPixelColor(0, Board.PixelColor(255, 0, 0))**" as example.

The the first parameter represents RGB light. “0” is RGB1 and “1” is RGB2.

The second parameter is color channel parameter set by “Board.PixelColor()” function. The first parameter “225” represents “R” channel (red component). The second and third parameters are the value of “G” and “B” channels.

2. Preparation

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



3. Operations 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 RGB_Change.py" to start the game and press "Enter".

```
ubuntu@ubuntu:~/armpi_pro/src/armpi_pro_demo/expansion_board_demo$
sudo python3 RGB_Change.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, the two RGB on the Raspberry Pi expansion board flash in different colors in cycles, and the color flashing sequence is red, green, and blue. Every lights flash 1 second.

5. Function Extension

The RGB light defaults to light up red. If want to modify its color.

In program, RGB lights up red light by default. If want to modify its color, we can modify the program. Take changing green light as example.

1) Enter command **"cd armpi_pro/src/armpi_pro_demo/expansion_board_demo/"** and press "Enter" to access to the game programmings directory.

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

2) Enter command "sudo vim RGB_Change.py" and press "Enter" to open the program file.

```
ubuntu@ubuntu:~/armpi_pro/src/armpi_pro_demo/expansion_board_demo$
sudo vim RGB_Change.py
```

3) Find the code shown in the figure below.

```
35      #Set two lights as red. 0 is RGB1 and 1 is RGB2.
36      Board.RGB.setPixelColor(0, Board.PixelColor(255, 0, 0))
37      Board.RGB.setPixelColor(1, Board.PixelColor(255, 0, 0))
38      Board.RGB.show()
39      time.sleep(1)
40
41      #Set two lights as green. 0 is RGB1 and 1 is RGB2.
42      Board.RGB.setPixelColor(0, Board.PixelColor(0, 255, 0))
43      Board.RGB.setPixelColor(1, Board.PixelColor(0, 255, 0))
44      Board.RGB.show()
45      time.sleep(1)
46
47      #Set two lights as blue. 0 is RGB1 and 1 is RGB2.
48      Board.RGB.setPixelColor(0, Board.PixelColor(0, 0, 255))
49      Board.RGB.setPixelColor(1, Board.PixelColor(0, 0, 255))
50      Board.RGB.show()
51      time.sleep(1)
```

4) Press "i" key. When "INSERT" word appears, which means it has been switched to the editing mode.

```
 59          break
-- INSERT --
```

5) We can change color by modifying RGB value. (Board.PixelColor (0, 0, 0) corresponds to RGB value. The first parameter is R, the second parameter is G and the third parameter is B, as the figure shown below.

```
33
34 while True:
35     #设置2个灯为红色,0为RGB1,1为RGB2
36     Board.RGB.setPixelColor(0, Board.PixelColor(0, 0, 255))
37     Board.RGB.setPixelColor(1, Board.PixelColor(0, 0, 255))
38     Board.RGB.show()
39     time.sleep(1)
40
41     #设置2个灯为绿色,0为RGB1,1为RGB2
42     Board.RGB.setPixelColor(0, Board.PixelColor(0, 255, 0))
43     Board.RGB.setPixelColor(1, Board.PixelColor(0, 255, 0))
44     Board.RGB.show()
45     time.sleep(1)
46
47     #设置2个灯为蓝色,0为RGB1,1为RGB2
48     Board.RGB.setPixelColor(0, Board.PixelColor(255, 0, 0))
49     Board.RGB.setPixelColor(1, Board.PixelColor(255, 0, 0))
50     Board.RGB.show()
```

6) After modifying, press “Esc” and input “:wq”. Then press “Enter” to save and exit.

```
59         break
:wq
```

7) After saving program, please refer to “3.Operation Steps” to check the outcome.