6.RGB light special effects

RGB cooling HAT needs to be correctly inserted into the GPIO port of the Raspberry Pi and the I2C function of the Raspberry Pi must be turned on.

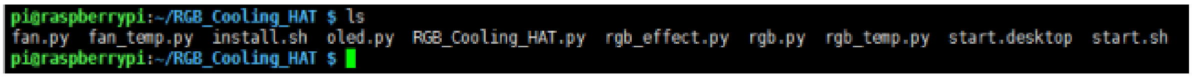
The phenomenon of this experiment is that the breathing light effect of all RGB lights turning purple.

1.Compile and run the program

1.1 Enter the folder and view the files in the current folder

```
cd RGB_Cooling_HAT/
```

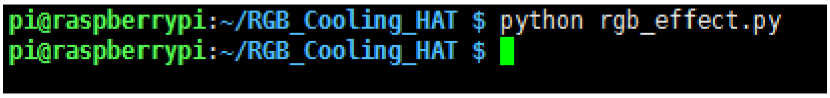
```
ls
```



```
pi@raspberrypi:~/RGB_Cooling_HAT $ ls
fan.py  fan_temp.py  install.sh  oled.py  RGB_Cooling_HAT.py  rgb_effect.py  rgb.py  rgb_temp.py  start.desktop  start.sh
pi@raspberrypi:~/RGB_Cooling_HAT $ 
```

1.2 Run program

```
python rgb_effect.py
```



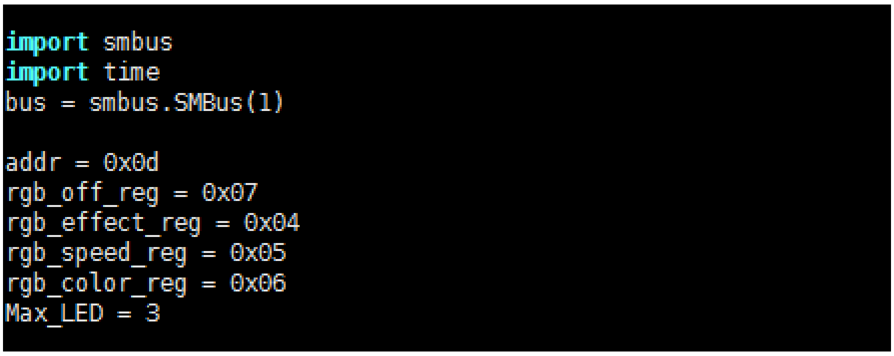
```
pi@raspberrypi:~/RGB_Cooling_HAT $ python rgb_effect.py
pi@raspberrypi:~/RGB_Cooling_HAT $ 
```

At this time, we can see the purple breathing light effect of three RGB lights lighting up at the same time.

2. Code analysis

2.1 Initialize the Raspberry Pi I2C configuration, import the smbus module for I2C communication, and the time module for delay.

There are three RGB lights on the smart personal butler board, so the number of lights is defined as 3, and the register address: rgb_effect is 0x04 , rgb_speed is 0x05, rgb_color is 0x06, rgb_off is 0x07.

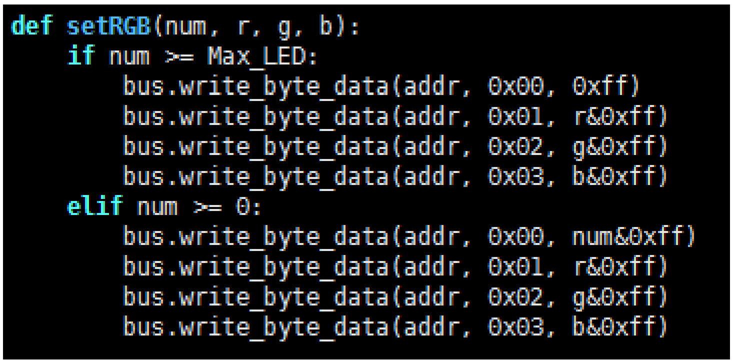


```
import smbus
import time
bus = smbus.SMBus(1)

addr = 0x0d
rgb_off_reg = 0x07
rgb_effect_reg = 0x04
rgb_speed_reg = 0x05
rgb_color_reg = 0x06
Max_LED = 3
```

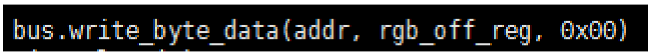
2.2 setRGB(num, r, g, b) function:

Set the RGB light color, num refers to the serial number of the RGB light, 0 is the first light, 1 is the second light, and 2 is the third light.
If greater than or equal to 3, all lights are set at the same time. The value range of R, G, and B values is 0~255.



```
def setRGB(num, r, g, b):
    if num >= Max_LED:
        bus.write_byte_data(addr, 0x00, 0xff)
        bus.write_byte_data(addr, 0x01, r&0xff)
        bus.write_byte_data(addr, 0x02, g&0xff)
        bus.write_byte_data(addr, 0x03, b&0xff)
    elif num >= 0:
        bus.write_byte_data(addr, 0x00, num&0xff)
        bus.write_byte_data(addr, 0x01, r&0xff)
        bus.write_byte_data(addr, 0x02, g&0xff)
        bus.write_byte_data(addr, 0x03, b&0xff)
```

2.3 Turn off RGB light. According to the protocol, the register for turning off RGB is 0x07 and the data is 0x00.

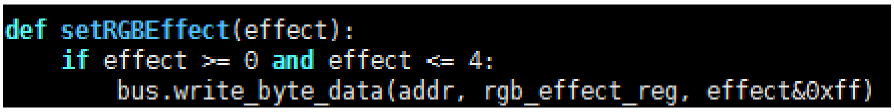


```
bus.write_byte_data(addr, rgb_off_reg, 0x00)
```

2.4 setRGBEffect(effect) function.

First, determine whether the input value corresponds to the protocol.

There are a total of five special effects: 0 running water light, 1 breathing light, 2 marquee light, 3 rainbow light, and 4 colorful light.



```
def setRGBEffect(effect):
    if effect >= 0 and effect <= 4:
        bus.write_byte_data(addr, rgb_effect_reg, effect&0xff)
```

2.5 setRGBSpeed(speed) function.

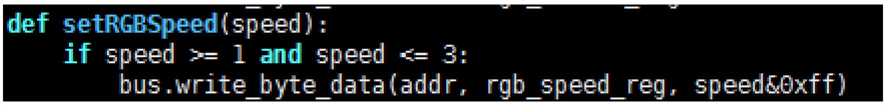
Modify the RGB light switching speed of the above modes.

1 low speed;

2 medium speed (default);

3 high speed;

if not set, the default is medium speed.

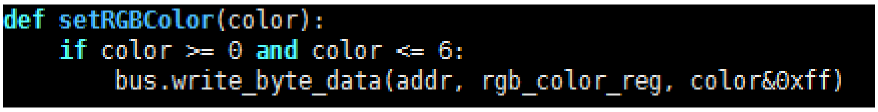


```
def setRGBSpeed(speed):
    if speed >= 1 and speed <= 3:
        bus.write_byte_data(addr, rgb_speed_reg, speed&0xff)
```

2.6 setRGBColor(color) function.

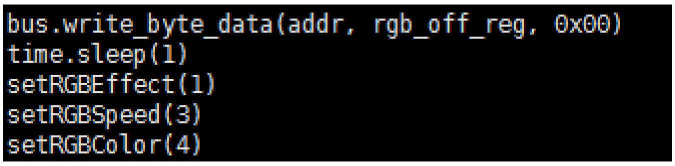
Set the color of the running water light and breathing light in the RGB light effect.

0 red, 1 green (default), 2 blue, 3 yellow, 4 purple, 5 cyan, 6 white. If not set, it defaults to green.



```
def setRGBColor(color):
    if color >= 0 and color <= 6:
        bus.write_byte_data(addr, rgb_color_reg, color&0xff)
```

2.7 We take the purple breathing light that lights up at high speed as an example.



```
bus.write_byte_data(addr, rgb_off_reg, 0x00)
time.sleep(1)
setRGBEffect(1)
setRGBSpeed(3)
setRGBColor(4)
```