5.Change RGB color based on CPU temperature

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 to read and print the Raspberry Pi CPU temperature, and adjust the color change of the RGB light according to the temperature.

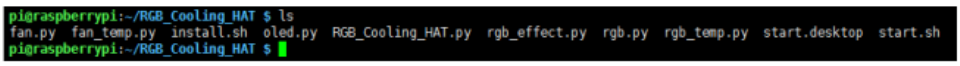
The color distribution is: low temperature turns blue, medium temperature turns yellow, and high temperature turns red.

1.Compile and run the program

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

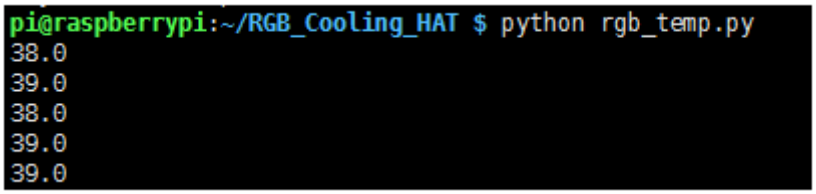
```
cd RGB_Cooling_HAT/
```

```
ls
```



1.2 Run programm

```
python rgb_temp.py
```



At this time, the terminal will print the current CPU temperature value, and the RGB light color will also change with the color change.

2.Code analysis

2.1 Initialize the Raspberry Pi I2C configuration, import the smbus module for I2C communication, the time module for delay, and the os module for accessing operating system service functions.

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

addr = 0x0d
fan_reg = 0x08
state = 0
temp = 0
level_temp = 0
rgb_off_reg = 0x07
Max_LED = 3
```

2.2 Define and set the RGB light function

```
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 the RGB lights first, and then set the RGB lights. If not turned off first, the display effect will sometimes be affected.

```
bus.write_byte_data(addr,rgb_off_reg,0x00)
time.sleep(1)
```

2.4 In the loop, use the command vcgencmd measure_temp to obtain the CPU temperature through os.popen to obtain the temperature, intercept the temperature value and assign it to temp.

```
cmd = os.popen('vcgencmd measure_temp').readline()
CPU_TEMP = cmd.replace("temp=","").replace("'C\n","")
print(CPU_TEMP)
temp = float(CPU_TEMP)
```

2.5 After obtaining the temperature, determine the temperature value and modify the fan speed. It can be modified according to actual needs. For RGB light color correspondence, you can search online to view the RGB color comparison table.

```
    if abs(temp - level_temp) >= 1:
        if temp <= 45:
            level_temp = 45
            setRGB(Max_LED, 0x00, 0x00, 0xff)
        elif temp <= 47:
            level_temp = 47
            setRGB(Max_LED, 0x1e, 0x90, 0xff)
        elif temp <= 49:
            level_temp = 49
            setRGB(Max_LED, 0x00, 0xbf, 0xff)
        elif temp <= 51:
            level_temp = 51
            setRGB(Max_LED, 0x5f, 0x9e, 0xa0)
        elif temp <= 53:
            level_temp = 53
            setRGB(Max_LED, 0xff, 0xff, 0x00)
        elif temp <= 55:
            level_temp = 55
            setRGB(Max_LED, 0xff, 0xd7, 0x00)
        elif temp <= 57:
            level_temp = 57
            setRGB(Max_LED, 0xff, 0xa5, 0x00)
        elif temp <= 59:
            level_temp = 59
            setRGB(Max_LED, 0xff, 0x8c, 0x00)
        elif temp <= 61:
            level_temp = 61
            setRGB(Max_LED, 0xff, 0x45, 0x00)
        elif temp >= 63:
            level_temp = 63
            setRGB(Max_LED, 0xff, 0x00, 0x00)
```