5 - Change RGB color according to CPU temperature

The Raspberry Pi RGB_Cooling_HAT needs to be properly plugged into the GPIO port of the Raspberry Pi and open the Raspberry Pi system **I2C** function.

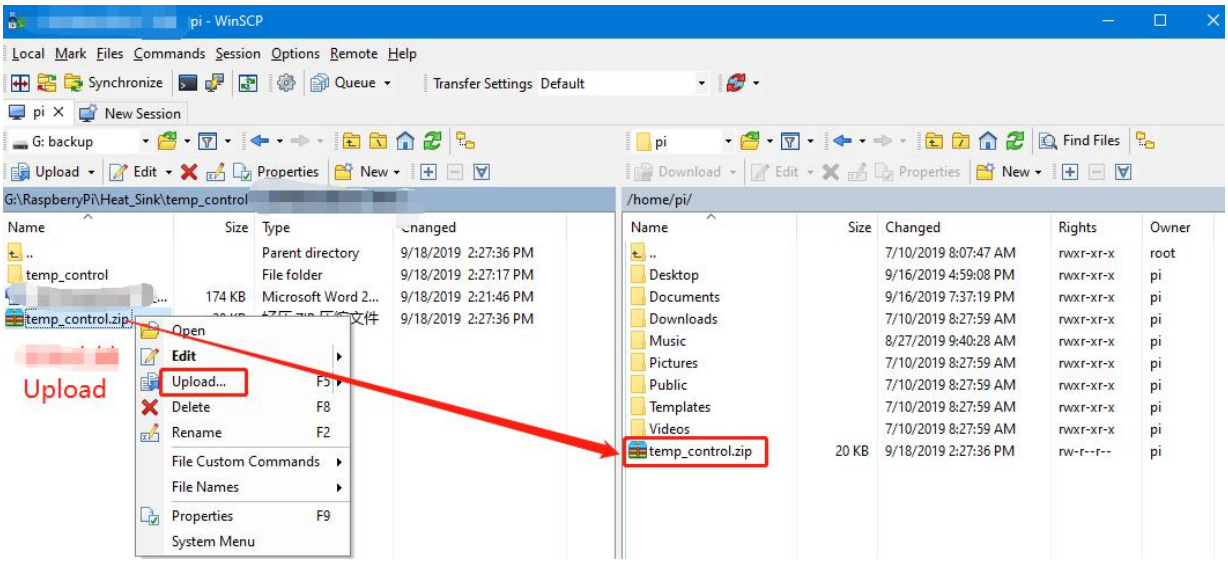
The experimental phenomenon is to read and print the Raspberry Pi CPU temperature, adjust the color change of the RGB lamp according to the temperature.

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

1. File transfer

1.1 Install **WinSCP** tool on the computer side, connect the Raspberry Pi and transfer the **temp_control.zip** package to the pi directory of the Raspberry Pi.

Path of WinSCP:[Raspberry Pi RGB_Cooling_HAT]---[Tools]---[winscp556_setup.1416364912.exe]



1.2 Extract file

Open the Raspberry Pi terminal and input command ls to find the RGB_Cooling_HAT.zipfile.

As shown below:

Input command to extract file:

unzip RGB_Cooling_HAT.zip

```
pi@raspberrypi:~ $ unzip RGB_Cooling_HAT.zip
Archive:  RGB_Cooling_HAT.zip
   creating: RGB_Cooling_HAT/
  inflating: RGB_Cooling_HAT/RGB_Cooling_HAT.py
  inflating: RGB_Cooling_HAT/fan.py
  inflating: RGB_Cooling_HAT/fan_temp.py
  inflating: RGB_Cooling_HAT/install.sh
  inflating: RGB_Cooling_HAT/oled.py
  inflating: RGB_Cooling_HAT/rgb.py
  inflating: RGB_Cooling_HAT/rgb_effect.py
  inflating: RGB_Cooling_HAT/rgb_temp.py
 extracting: RGB_Cooling_HAT/start.desktop
  inflating: RGB_Cooling_HAT/start.sh
```

2. Compiling and running program

2.1 Input command to enter temp_control find file:

```
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 $
```

2.2 Input command to run program

```
python rgb_temp.py
```

```
pi@raspberrypi:~/RGB_Cooling_HAT $ python rgb_temp.py
38.0
39.0
38.0
39.0
39.0
```

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

3. About code

3.1 Initialize the Raspberry Pi I2C configuration, import smbus module for I2C communication, import time module for delay, and import os module for access to operating system services.

```
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
```

3.2 Define 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)
```

3.3 Turn off the RGB lights first, and then set the RGB lights. If you do not turn off the lights, it will affect the display effect.

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

3.4 In the loop, use os.popen to obtain the temperature using the command vcgencmd measure_temp to obtain the CPU temperature, and 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)
```

3.5 Obtain the temperature, determine the temperature value, and modify the fan speed.
It can be modified according to actual needs. The RGB lamp color correspondence can be searched 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)
```