6 RGB effect

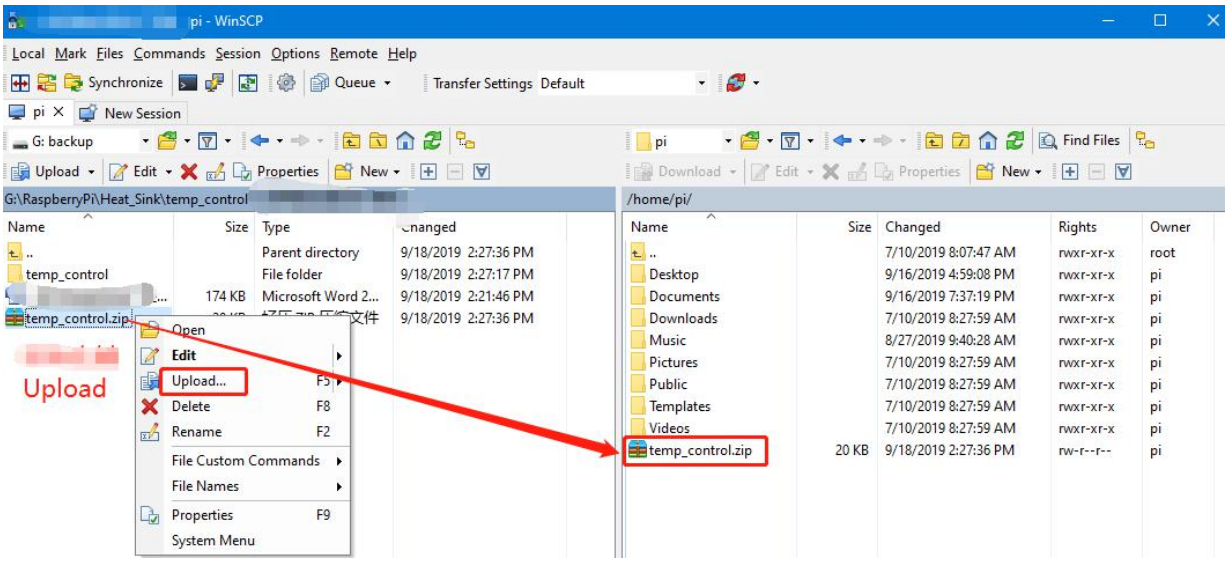
The Raspberry Pi RGB_Cooling_HAT needs to be properly plugged into the GPIO port of the Raspberry Pi and open the Raspberry Pi system **I2C** function.

This experimental phenomenon shows that all RGB light become purple breathing light.

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]

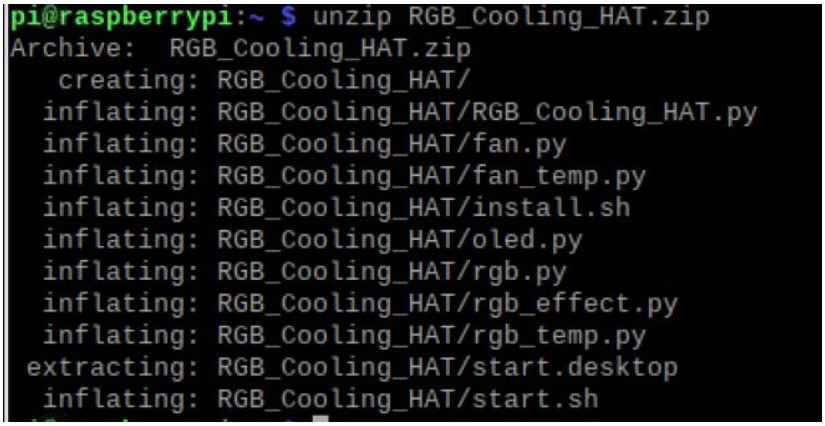


1.2 Extract file

Open the Raspberry Pi terminal and input command ls to find the RGB_Cooling_HAT.zip file. As shown below:

Input command to extract file:

unzip RGB_Cooling_HAT.zip



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 following command ti run program

```
python rgb_effect.py
```

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

We can see that all RGB light become purple breathin light.

3. About code

3.1 Initialize the I2C configuration of the Raspberry Pi, import the smbus module for I2C communication, import the time module for delay.

There are three RGB lights on the Raspberry-Pi-RGB-Cooling-HAT, so define the number of lights as 3. Define the register address: rgb_effect is 0x04, rgb_speed Is 0x05, rgb_color is 0x06, and 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. setRGB(num, r, g, b) function:

Set the RGB lamp color, num refers to which lamp.

0 is the first lamp, 1 is the second lamp, 2 is the third lamp.

If greater than or equal to 3, all lamps are set at the same time. The range of R, G, 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)
```

3.3 Turn off RGB. According to the protocol, the register to turn off RGB is 0x07, and the data is

0x00.

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

3.4 setRGBEffect(effect) function.

First,we need to judge the input value. There are five kinds of special effects to choice, 0-water light, 1-breathing light, 2-marquee, 3-rainbow lights, 4-colorful lights.

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

3.5 setRGBSpeed(speed) function.

Modify the RGB light switching speed of the mode. 1-low speed, 2-medium speed (default), 3-high speed. If you do not set, it is medium speed(default).

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

6. setRGBColor(color) function.

Set the color of the water light and breathing light, 0-red, 1-green (default), 2-blue, 3-yellow, 4-purple, 5-cyan, 6-white. If not set, the default is green.

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

7. An example of setting a purple high-speed breathing light.

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