

6 RGB effect

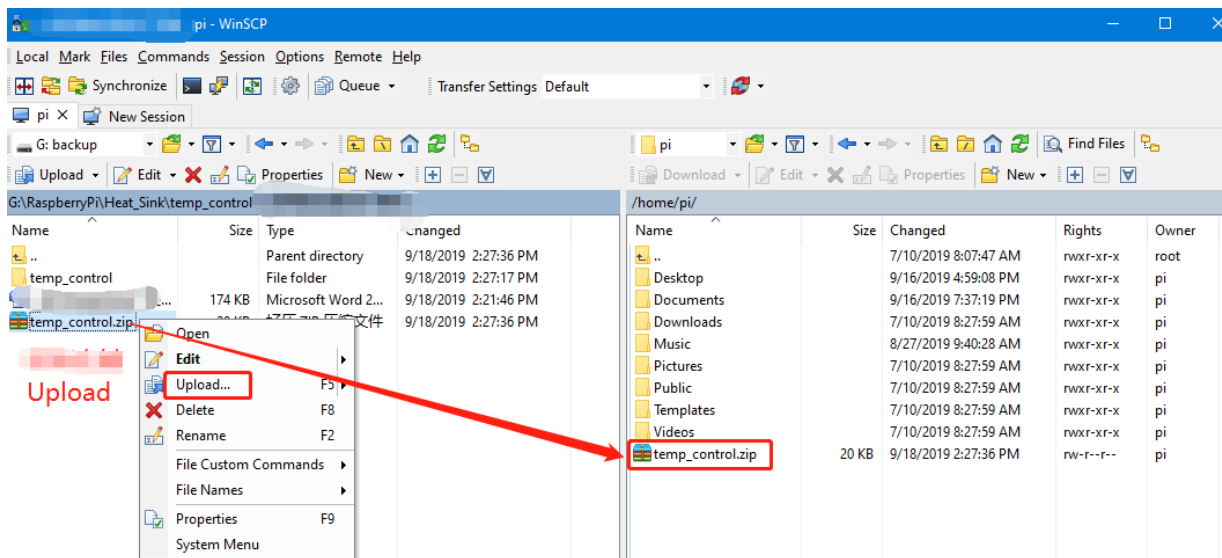
The Raspberry Pi RGB_Cooling_HAT needs to be properly plugged into the GPIO port of the Raspberry Pi and open the Raspberry Pi system **I2C** function.

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 temp_control.zip file. As shown below:

```
pi@raspberrypi:~ $ ls
Desktop  Downloads  Pictures  temp_control.zip  Videos
Documents Music      Public   Templates
```

Input command to extract file:

unzip temp_control.zip

```
pi@raspberrypi:~ $ unzip temp_control.zip
Archive:  temp_control.zip
  creating: temp_control/
  inflating: temp_control/fan
  inflating: temp_control/fan.c
  inflating: temp_control/fan_temp
  inflating: temp_control/fan_temp.c
  inflating: temp_control/oled
  inflating: temp_control/oled.c
  inflating: temp_control/oled_fonts.h
  inflating: temp_control/rgb
  inflating: temp_control/rgb.c
  inflating: temp_control/rgb_effect
  inflating: temp_control/rgb_effect.c
  inflating: temp_control/ssdl306_i2c.c
  inflating: temp_control/ssdl306_i2c.h
  inflating: temp_control/start.desktop
  inflating: temp_control/start.sh
  inflating: temp_control/temp_control
  inflating: temp_control/temp_control.c
pi@raspberrypi:~ $
```

2. Compiling and running program

2.1 Input command to enter temp_control find file:

```
cd temp_control/
ls
```

```
pi@raspberrypi:~ $ cd temp_control/
pi@raspberrypi:~/temp_control $ ls
fan          oled          rgb.c          ssdl306_i2c.h  temp_control.c
fan.c        oled.c        rgb_effect     start.desktop
fan_temp     oled_fonts.h rgb_effect.c   start.sh
fan_temp.c   rgb           ssdl306_i2c.c temp_control
pi@raspberrypi:~/temp_control $
```

2.2 Input command to compile:

```
gcc -o rgb_effect rgb_effect.c -lwiringPi
```

```
pi@raspberrypi:~/temp_control $ gcc -o rgb rgb.c -lwiringPi
pi@raspberrypi:~/temp_control $ ls
fan          fan_temp.c    oled_fonts.h  rgb_effect     ssdl306_i2c.h  temp_control
fan.c        oled          rgb           rgb_effect.c   start.desktop  temp_control.c
fan_temp     oled.c       rgb.c         ssdl306_i2c.c start.sh
pi@raspberrypi:~/temp_control $
```

Among them, the gcc compiler is called, -o means to generate the file, **rgb_effect** is the generated file name, **rgb_effect.c** is the source program, **-lwiringPi** is the wiringPi library that references the Raspberry Pi.

2.3 Input command to run the program

```
./rgb_effect
```

```
pi@raspberrypi:~/temp_control $ ./rgb_effect
pi@raspberrypi:~/temp_control $
```

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

3. About code

3.1 There are three RGB lights on the expansion board, so the number of defined lights is 3, Define the register address: RGB_Effect is 0x04, RGB_Speed is 0x05, and RGB_Color is 0x06. Declare the function you need to use.

```
#define Max_LED    3
#define RGB_Effect 0x04
#define RGB_Speed  0x05
#define RGB_Color  0x06

int fd_i2c;
void setRGB(int num, int R, int G, int B);
void closeRGB();

void setRGBEffect(int effect);
void setRGBSpeed(int speed);
void setRGBColor(int color);
```

3.2 void setRGB(int num, int R, int G, int B) function:

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

```
void setRGB(int num, int R, int G, int B)
{
    if (num >= Max_LED)
    {
        wiringPiI2CWriteReg8(fd_i2c, 0x00, 0xff);
        wiringPiI2CWriteReg8(fd_i2c, 0x01, R);
        wiringPiI2CWriteReg8(fd_i2c, 0x02, G);
        wiringPiI2CWriteReg8(fd_i2c, 0x03, B);
    }
    else if (num >= 0)
    {
        wiringPiI2CWriteReg8(fd_i2c, 0x00, num);
        wiringPiI2CWriteReg8(fd_i2c, 0x01, R);
        wiringPiI2CWriteReg8(fd_i2c, 0x02, G);
        wiringPiI2CWriteReg8(fd_i2c, 0x03, B);
    }
}
```

3.3 Turn off RGB. According to the protocol, the RGB register is turned off to 0x07 and the data is 0x00.

```
void closeRGB()
{
    wiringPiI2CWriteReg8(fd_i2c, 0x07, 0x00);
}
```

4. void setRGBEffect(int effect) function:

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

```
void setRGBEffect(int effect)
{
    if (effect >= 0 && effect <= 4)
    {
        wiringPiI2CWriteReg8(fd_i2c, RGB_Effect, effect);
    }
}
```

5. void setRGBSpeed(int 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).

```
void setRGBSpeed(int speed)
{
    if (speed >= 1 && speed <= 3)
    {
        wiringPiI2CWriteReg8(fd_i2c, RGB_Speed, speed);
    }
}
```

6. void setRGBColor(int 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.

```
void setRGBColor(int color)
{
    if (color >= 0 && color <= 6)
    {
        wiringPiI2CWriteReg8(fd_i2c, RGB_Color, color);
    }
}
```

7. Initialize the I2C configuration

```
wiringPiSetup();
fd_i2c = wiringPiI2CSetup(0x0d);
if (fd_i2c < 0)
{
    fprintf(stderr, "fail to init I2C\n");
    return -1;
}
```

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

```
closeRGB();  
delay(1);  
  
setRGBEffect(1);  
setRGBSpeed(3);  
setRGBColor(4);
```