

2.Control fan

The 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.

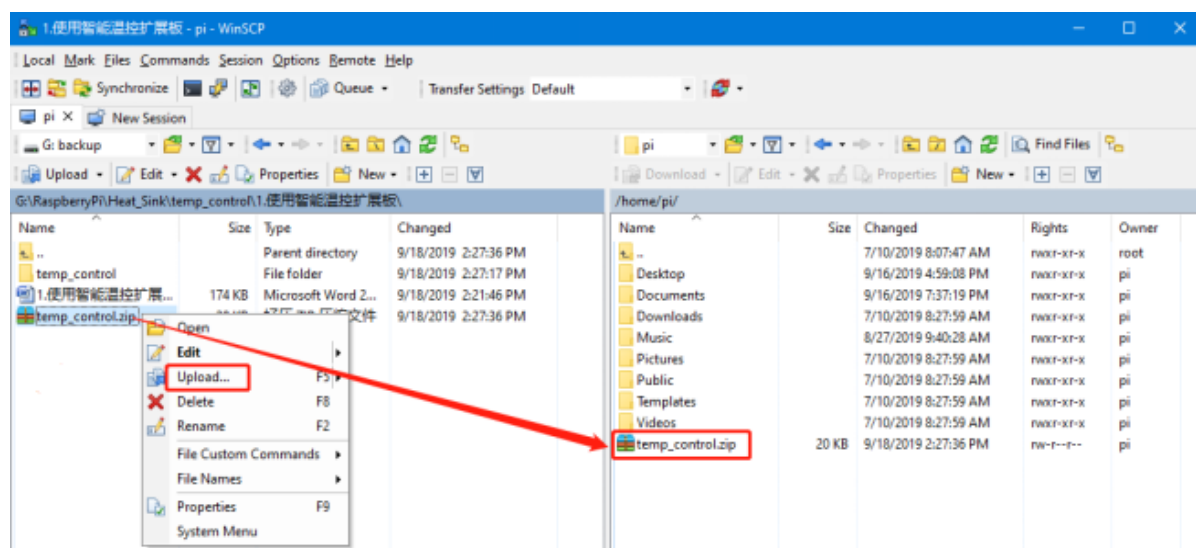
The phenomenon of this experiment is to control the speed of the fan.

After 2 seconds, increases the fan speed every second, when it up to highest speed, run 2seconds. Then, it will stop.

1. File transfer (If the Raspberry Pi system already has files, you can ignore this step)

1.1 Install the WinSCP tool on the computer.

Connect to the Raspberry Pi and transfer the temp_control.zip compressed package downloaded in the data to the pi directory of the Raspberry Pi.



1.2 Unzip files

Open the terminal of Raspberry Pi and find the temp_control.zip file just transferred to Raspberry Pi.

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

Enter the following 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. Compile and run the program

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

```

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 Compile program files

```

gcc -o fan fan.c -lwiringPi

```

```

pi@raspberrypi:~/temp_control $ gcc -o fan fan.c -lwiringPi
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 $ █

```

2.3 Run

```

./fan

```

```

pi@raspberrypi:~/temp_control $ ./fan
█

```

After 2 seconds, increases the fan speed every second, when it up to highest speed, run 2seconds. Then, it will stop.

3.Code Analysis

3.1 Initialize Raspberry Pi I2C configuration

```
#include <stdio.h>
// 导入wiringPi/I2C库
#include <wiringPi.h>
#include <wiringPiI2C.h>

int main(void)
{
    int state = 0;
    // 定义I2C相关参数
    int fd_i2c;
    wiringPiSetup();
    fd_i2c = wiringPiI2CSetup(0x0d);
    if (fd_i2c < 0)
    {
        fprintf(stderr, "fail to init I2C\n");
        return -1;
    }
}
```

3.2 Cyclically control the fan speed.

According to the protocol, the fan speed level is known, 0x00 off, 0x01 full speed, 0x02: 20% speed, 0x03: 30% speed,..., 0x09: 90% speed

```
// 循环让state自加1, 每次加1都发一个命令调节调节风扇的速度
while (1)
{
    switch (state)
    {
        case 0:
            wiringPiI2CWriteReg8(fd_i2c, 0x08, 0x00);
            break;
        case 1:
            wiringPiI2CWriteReg8(fd_i2c, 0x08, 0x02);
            break;
        case 2:
            wiringPiI2CWriteReg8(fd_i2c, 0x08, 0x03);
            break;
        case 3:
            wiringPiI2CWriteReg8(fd_i2c, 0x08, 0x04);
            break;
        case 4:
            wiringPiI2CWriteReg8(fd_i2c, 0x08, 0x05);
            break;
        case 5:
            wiringPiI2CWriteReg8(fd_i2c, 0x08, 0x06);
            break;
        case 6:
            wiringPiI2CWriteReg8(fd_i2c, 0x08, 0x07);
            break;
        case 7:
            wiringPiI2CWriteReg8(fd_i2c, 0x08, 0x08);
            break;
        case 8:
            wiringPiI2CWriteReg8(fd_i2c, 0x08, 0x09);
            break;
        case 9:
            wiringPiI2CWriteReg8(fd_i2c, 0x08, 0x01);
            break;
        default:
            break;
    }
}
```

3.3 Limit the state size. If it is greater than 9, set it to 0 to achieve a loop effect.

```
if (state == 0)
{
    delay(1000);
}

state++;

if (state > 9)
{
    delay(1000);
    state = 0;
}

delay(1000);
```