Using RGB cooling HAT

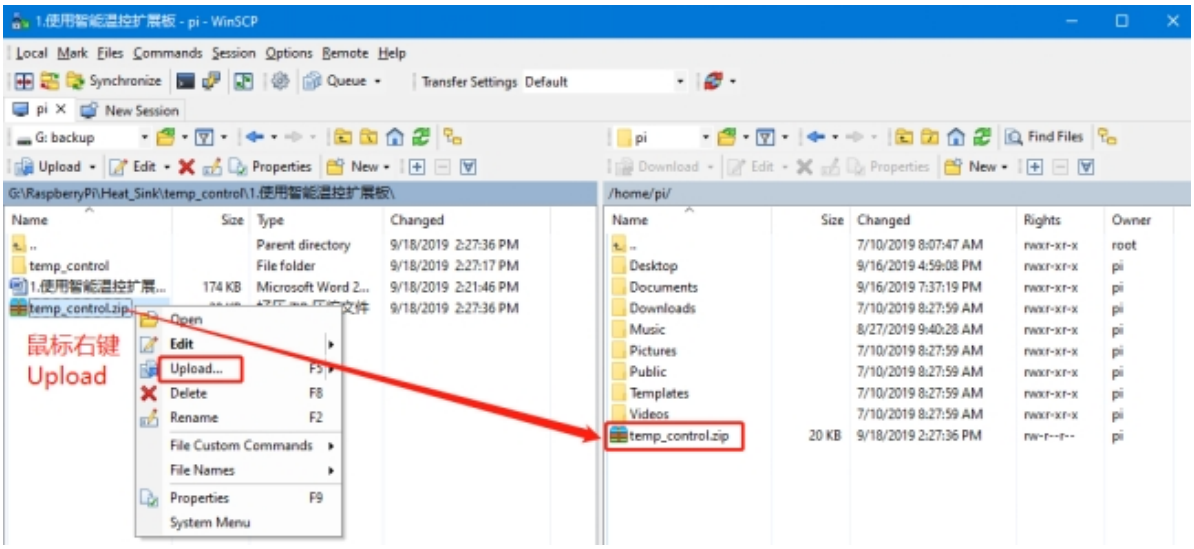
RGB cooling HAT needs to be correctly inserted into the GPIO port of the RDK X3, and the I2C function of the RDK X3 needs to be turned on. **RDK X3 enables I2C0 by default on 40Pin, physical pin numbers 3 and 5, and IO voltage 3.3V.**

The phenomenon of this experiment is: OLED displays the CPU usage, CPU temperature, running memory usage, disk usage and IP address of RDK X3, and the RGB light turns on special effects.

When the CPU temperature of the X3 Pi reaches a certain level, the fan will automatically turn on to dissipate heat for RDK X3, and when the temperature drops, the fan will automatically turn off.

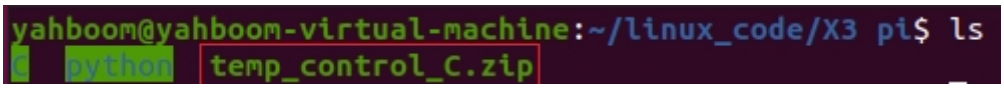
1.File transfer

1.Install WinSCP on your computer, connect to RDK X3, and transfer the **temp_control.zip** package downloaded from the documentation to the pi directory of the RDK X3.



2.Extract file

Open the terminal in RDK X3 system and find **temp_control_C.zip** file.



Enter the following command to extract the file.

```
unzip temp_control_C.zip
```

```
yahboom@yahboom-virtual-machine:~/linux_code/X3 pi$ unzip temp_control_C.zip
Archive:  temp_control_C.zip
   creating: temp_control_C/i2c_fan/
  inflating: temp_control_C/i2c_fan/fan
  inflating: temp_control_C/i2c_fan/fan_i2c_dan(┤e║└╩║└├).c
  inflating: temp_control_C/i2c_fan/i2c_fan.c
  inflating: temp_control_C/i2c_fan/myi2c.c
  inflating: temp_control_C/i2c_fan/myi2c.h
   creating: temp_control_C/i2c_OLED/
  inflating: temp_control_C/i2c_OLED/myi2c.c
  inflating: temp_control_C/i2c_OLED/myi2c.h
  inflating: temp_control_C/i2c_OLED/oled
  inflating: temp_control_C/i2c_OLED/oled.c
  inflating: temp_control_C/i2c_OLED/oled_fonts.h
  inflating: temp_control_C/i2c_OLED/ssd1306_i2c.c
  inflating: temp_control_C/i2c_OLED/ssd1306_i2c.h
   creating: temp_control_C/i2c_RGB/
  inflating: temp_control_C/i2c_RGB/I2C_RGB.c
  inflating: temp_control_C/i2c_RGB/myi2c.c
  inflating: temp_control_C/i2c_RGB/myi2c.h
  inflating: temp_control_C/i2c_RGB/RGB
   creating: temp_control_C/temp_control/
  inflating: temp_control_C/temp_control/install.sh
  inflating: temp_control_C/temp_control/myi2c.c
  inflating: temp_control_C/temp_control/myi2c.h
  inflating: temp_control_C/temp_control/oled_fonts.h
 extracting: temp_control_C/temp_control/readme.txt
  inflating: temp_control_C/temp_control/rgb_temp.c
  inflating: temp_control_C/temp_control/rgb_temp.h
  inflating: temp_control_C/temp_control/ssd1306_i2c.c
  inflating: temp_control_C/temp_control/ssd1306_i2c.h
  inflating: temp_control_C/temp_control/start.sh
  inflating: temp_control_C/temp_control/temp_control
  inflating: temp_control_C/temp_control/temp_control.c
  inflating: temp_control_C/temp_control/temp_control.h
```

2. Compiling and running the program

0.Input the following command to check the IIC peripherals.

```
python3 /app/40pin_samples/test_i2c.py
```

Input 0， press **Enter**

```
sunrise@ubuntu:~$ python3 /app/40pin_samples/test_i2c.py
Starting demo now! Press CTRL+C to exit
List of enabled I2C controllers:
/dev/i2c-0  /dev/i2c-1
Please input I2C BUS num:0
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:                         -- -- -- -- -- 0d -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- 3c -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
Please input I2C device num(Hex):
```

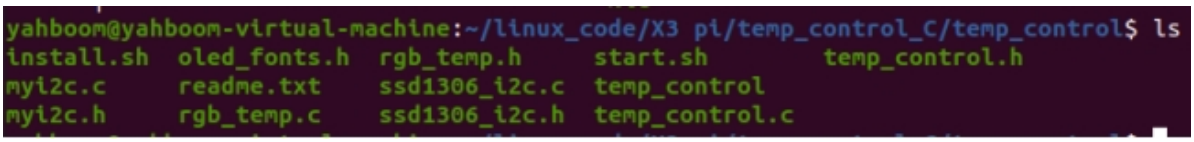
We can see that the system recognizes the IIC device address: 0x0d.

1.Enter the folder and view the files in the current folder

```
cd temp_control_C/temp_control
ls
```

2.Compile program files

```
gcc -o temp_control *.c
```

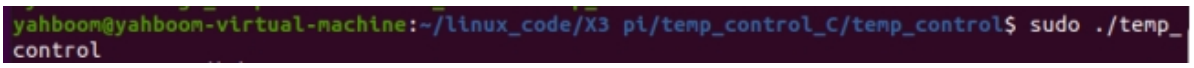


Among them, the gcc compiler is called, -o means to generate files, temp_control is the generated file name, temp_control.c is the source program, ssd1306_i2c.c is the library to drive oled, and myi2c.c is to drive the device on the I2C bus on RDK X3.

*.c: refers to all .c source files in the current directory.

3.Run program

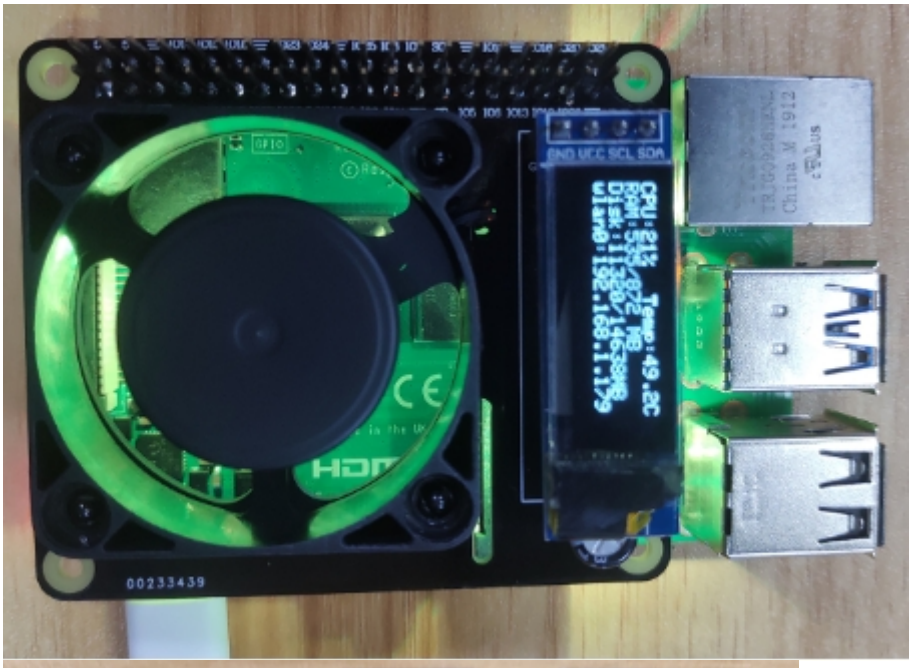
```
sudo ./temp_control
```

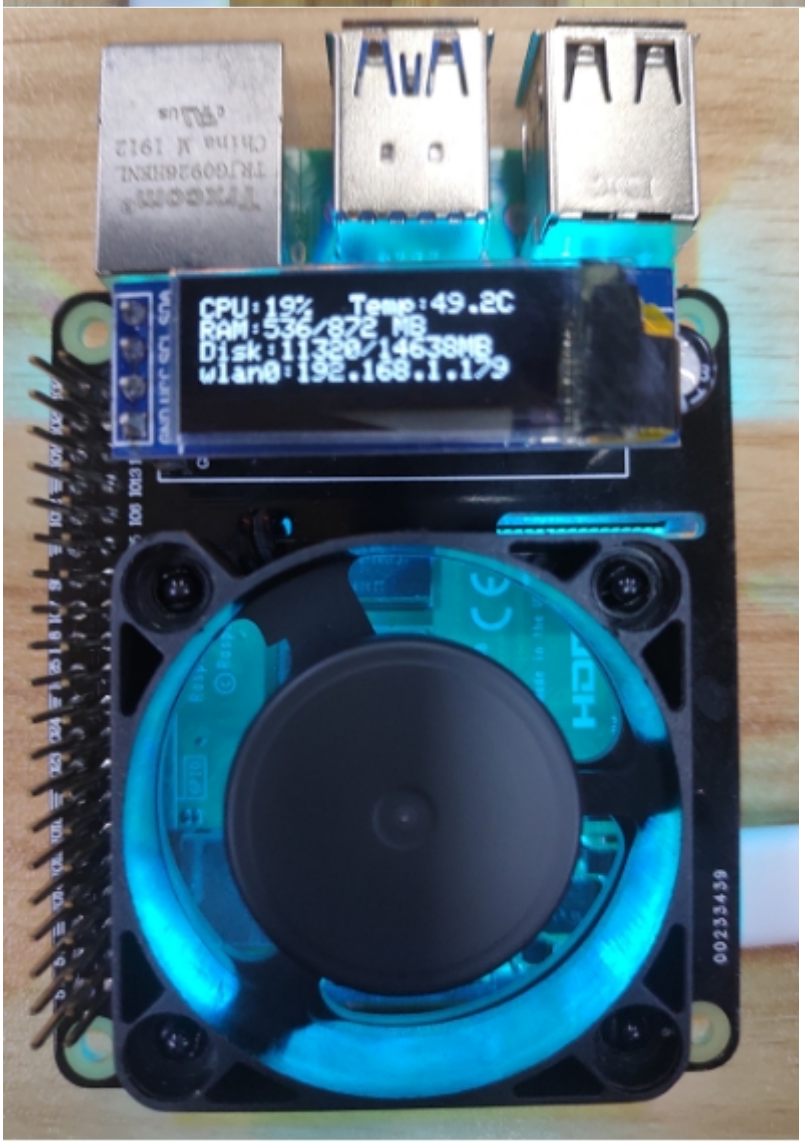


At this time, the system will prompt that the initialization is OK, and the RGB light will display special effects according to the temperature. At the same time, the OLED screen will display RDK X3 CPU usage, CPU temperature, running memory usage, disk usage, IP address and other information.

When the CPU temperature reaches 45 degrees, the fan will automatically turn on to dissipate heat for RDK X3. When the temperature drops to 42 degrees, the fan will automatically turn off.

Press Ctrl+C to exit the program.





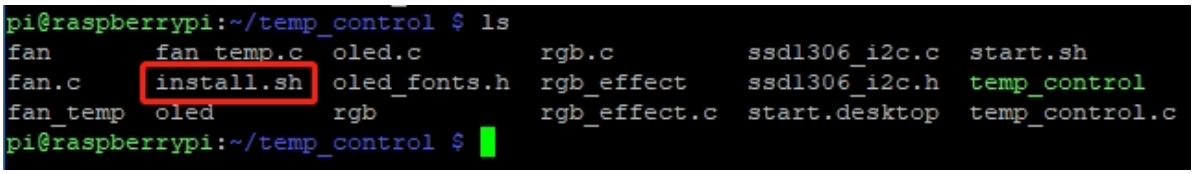
4. Add auto-startup

1.Enter the folder

```
cd /home/sunrise/temp_control_C/temp_control
```

2.View the installed script files

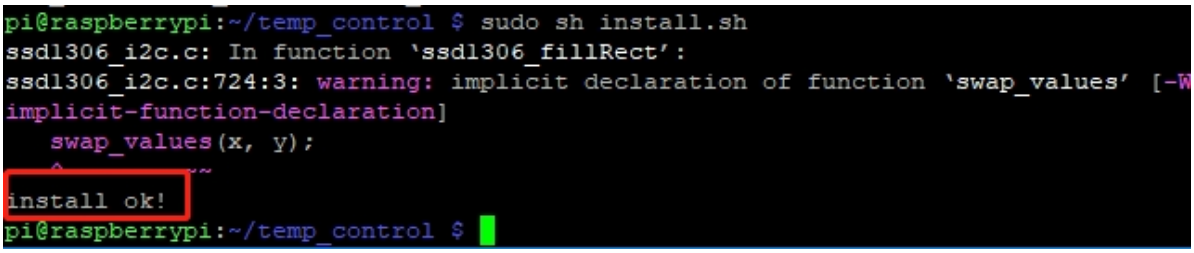
```
ls
```



3.Run the following command to install the script

```
sudo sh install.sh
```

The system will prompt that the installation is successful.



5. Reboot RDK X3

Enter the sudo reboot command to restart RDK X3, and you will see that the program starts up when the X3 Pi starts up.