

There are multiple ways for RDK X3 to start up automatically, and one of them is chosen as an example here.

## 1.Add a startup auto start feature

1. Open the desktop self startup script

```
vi .config/autostart/oled.desktop
```

Enter the following content.

A screenshot of a terminal window titled "Terminal - sunrise@ubuntu: ~". The terminal displays the contents of a desktop entry file, likely oled.desktop, which includes fields like Encoding=UTF-8, Type=Application, Name=myprogram, #Exec=bash -c "sh /home/sunrise/temp\_control\_C/temp\_control/start.sh;\$SHELL", and Terminal=false. A red arrow points from the right side of the terminal towards the path "/home/sunrise/temp\_control\_C/temp\_control/start.sh" in the #Exec line. At the bottom of the terminal, it shows ".config/autostart/oled.desktop" 6L, 156B, indicating the file's location and size. The status bar at the very bottom shows "1,1" and "All".

Open, save, and exit the fifth line comment.

```
[Desktop Entry]
Encoding=UTF-8
Type=Application
Name=myprogram
Exec= bash -c "sh /home/sunrise/temp_control_C/temp_control/start.sh;$SHELL"
Terminal=false
~
~
```

**vi editor operation:** After opening the file, enter i to start editing. After editing is complete, press ESC, and then enter wq to save and exit the file

## 2.Reboot RDK X3

Enter the following command on the terminal to restart X3, and the temp\_control program will automatically start after re-start.

Fans, RGB lights, and OLED screens will all have corresponding responses.

```
sudo reboot
```

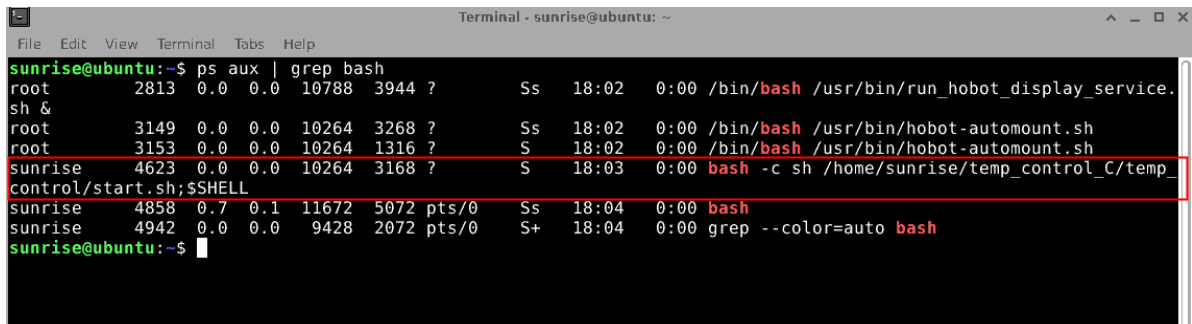
### 3. Exit the program

Due to the fact that the self starting program runs in the background, we are unable to exit the program directly from the open terminal.

If we need to modify the program, but the background process interferes with our debugging results? Of course, it is necessary to check the process ID (PID) of the background program and then end the process.

1. Enter a command on the terminal to open the process list.

```
ps aux | grep bash
```



Sometimes it may be necessary to wait for a while, as the system will prioritize processes that consume high CPU resources.

From the picture, we can see the program temp\_control that starts automatically upon startup, with PID 4623. Therefore, if we kill this process ID, the temp\_control program will no longer run in the background.

2. End process

```
sudo kill -9 4623
```

For example, in the above situation, we can run the command `sudo kill -9 4623` to terminate the temp\_control process running in the background.

If run again, it will prompt that the process does not exist.

```
sunrise@ubuntu:~$ sudo kill -9 4623
sunrise@ubuntu:~$ sudo kill -9 4623
kill: (4623): No such process
sunrise@ubuntu:~$
```

3. Restart the backend operation

If we have already ended a process running in the background but do not want to restart the background program, please use the following steps

We will still run the temp\_control program in the background.

First enter the target folder (determined based on the location of personal saved files)

```
cd ~/temp_control_C/temp_control
```

Add the `&` symbol at the end of the command to indicate background operation

```
sudo ./temp_control &
```

```
pi@raspberrypi:~ $ cd ~/temp_control/  
pi@raspberrypi:~/temp_control $ ls  
a      fan.c      fan_temp.c  oled.c      rgb      rgb_effect  ssd1306_i2c.c  start.desktop  temp_control  
fan    fan_temp  oled        oled_fonts.h  rgb.c    rgb_effect.c  ssd1306_i2c.h  start.sh      temp_control.c  
pi@raspberrypi:~/temp_control $ ./temp_control &  
[1] 1015  
pi@raspberrypi:~/temp_control $ init ok!  
█
```