8 Raspberry Pi boot self-starting setting

There are many ways to start the Raspberry Pi boot. Choose one of them (new .desktop file) as a demonstration.

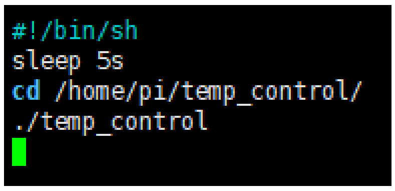
1. Add boot self-starting

1.1 New startup script start.sh

```
nano /home/pi/temp_control/start.sh
```

Input following content：

```
#!/bin/sh
sleep 5s
cd /home/pi/ temp_control /
./ temp_control
```



```
#!/bin/sh
sleep 5s
cd /home/pi/temp_control/
./temp_control
```

Press **ctrl+X**, press **Y** to save, press **Enter**.

2. New create boot startup program

2.1 Input following command to open**.config** folder

```
cd /home/pi/.config
```

2.2 Input following command to new create autostart folder

```
mkdir autostart
```

2.3 Input following command to enter autostart folder

```
cd autostart
```

2.4 Input following command to new create shortcut for self-starting

```
nano start.desktop
```

Input following content:

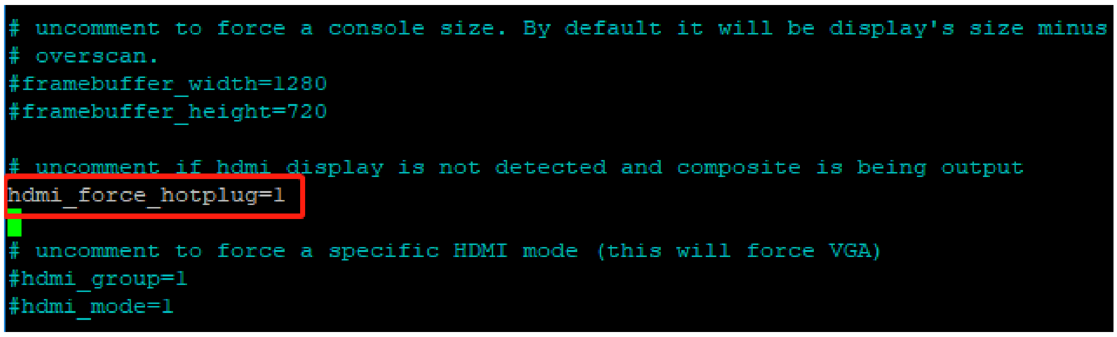
```
[Desktop Entry]
Type=Application
Exec=sh /home/pi/cpu_show_v3/start.sh
```

Press **ctrl+X**, press **Y** to save, press **Enter**.

Exec=Start command。

!!!Note:

Since this self-starting method needs to be started after the desktop is started, the startup will be slower. If it is found that it cannot be self-started after adding, please check if there is a ## in front of **hdmi_force_hotplug=1** in the **/boot/config.txt** file. If there is a # number, please delete the # number. The picture will prevail.



```
# uncomment to force a console size. By default it will be display's size minus
# overscan.
#framebuffer_width=1280
#framebuffer_height=720

# uncomment if hdmi display is not detected and composite is being output
hdmi_force_hotplug=1

# uncomment to force a specific HDMI mode (this will force VGA)
#hdmi_group=1
#hdmi_mode=1
```

3. Restart Raspberry Pi

After restarting, the temp_control program will start automatically, and the fan, RGB light and oled screen will have corresponding responses.

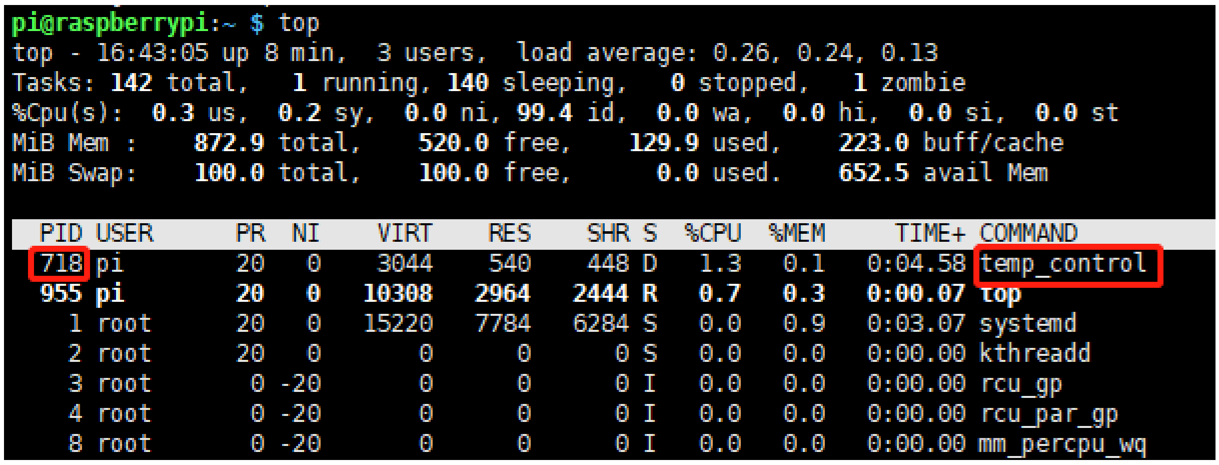
Input following command to restart Raspberry Pi:

sudo reboot

4. Exit the program

Since the self-starting program is running in the background, we cannot directly exit the program in the open terminal. If you need to modify the program, but the background process interferes with our debugging results, so you need to see the process number (PID) of the daemon, and then end this process.

4.1 Input top in the terminal to open the process list



```
pi@raspberrypi:~ $ top
top - 16:43:05 up 8 min,  3 users,  load average: 0.26, 0.24, 0.13
Tasks: 142 total,   1 running, 140 sleeping,   0 stopped,   1 zombie
%Cpu(s):  0.3 us,  0.2 sy,  0.0 ni, 99.4 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :    872.9 total,    520.0 free,    129.9 used,    223.0 buff/cache
MiB Swap:    100.0 total,    100.0 free,      0.0 used.    652.5 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
  718 pi        20   0    3044    540    448 D   1.3   0.1   0:04.58 temp_control
  955 pi        20   0   10308   2964   2444 R   0.7   0.3   0:00.07 top
    1 root      20   0   15220   7784   6284 S   0.0   0.9   0:03.07 systemd
    2 root      20   0       0      0      0 S   0.0   0.0   0:00.00 kthreadd
    3 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_gp
    4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_par_gp
    8 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 mm_percpu_wq
```

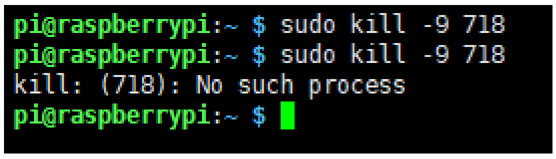
Sometimes it may take a while for the system to queue up processes that take up a lot of CPU resources. From the picture we can see the boot **temp_control** program, its PID is 718, so we kill this process number, temp_control program will not run in the background.

Press **Ctrl+C** to exit top.

4.2 Input following command to end the process

sudo kill -9 PID

For example: In the above case, we can run the sudo kill -9 718 command to end the temp_control process running in the background. If you run it again, it will be prompted that the process does not exist.



```
pi@raspberrypi:~ $ sudo kill -9 718
pi@raspberrypi:~ $ sudo kill -9 718
kill: (718): No such process
pi@raspberrypi:~ $
```

4.3 Restart the background operation

If we have finished the process running in the background, but we want to restart the background process.

First method: restart the Raspberry Pi,

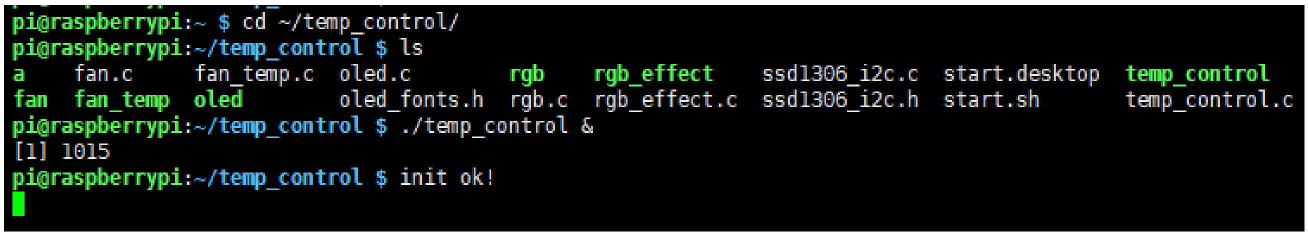
Second method: Add an “&” to the running program.

For example, we still run the **temp_control** program in the background:

We need to input following command to enter the target folder (based on the location of the file saved by the individual)

cd ~/temp_control

./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!
```

At this point, the system will prompt the PID of this process (1015).

Press Ctrl+C again. You can see that the terminal can enter other commands, and the program is running in the background.