Set up auto-startup

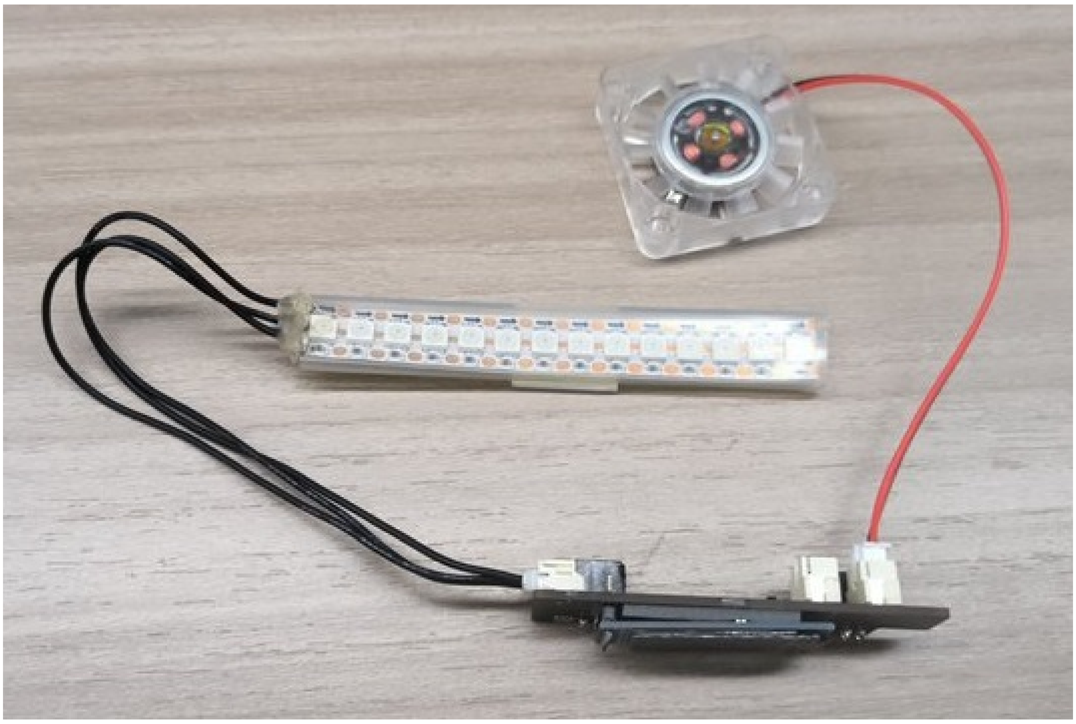
1. Learning objectives

- Understand the basic use of service
- Set up a custom startup program at startup
- Master the addition and cancellation of file execution permissions

Use the relevant files that have been written to set the relevant status of the OLED display, fan and RGB light strip after the Raspberry Pi series motherboard is turned on. The specific principle is not analyzed here. Read the relevant code to understand it yourself.

2. Preparation before the experiment

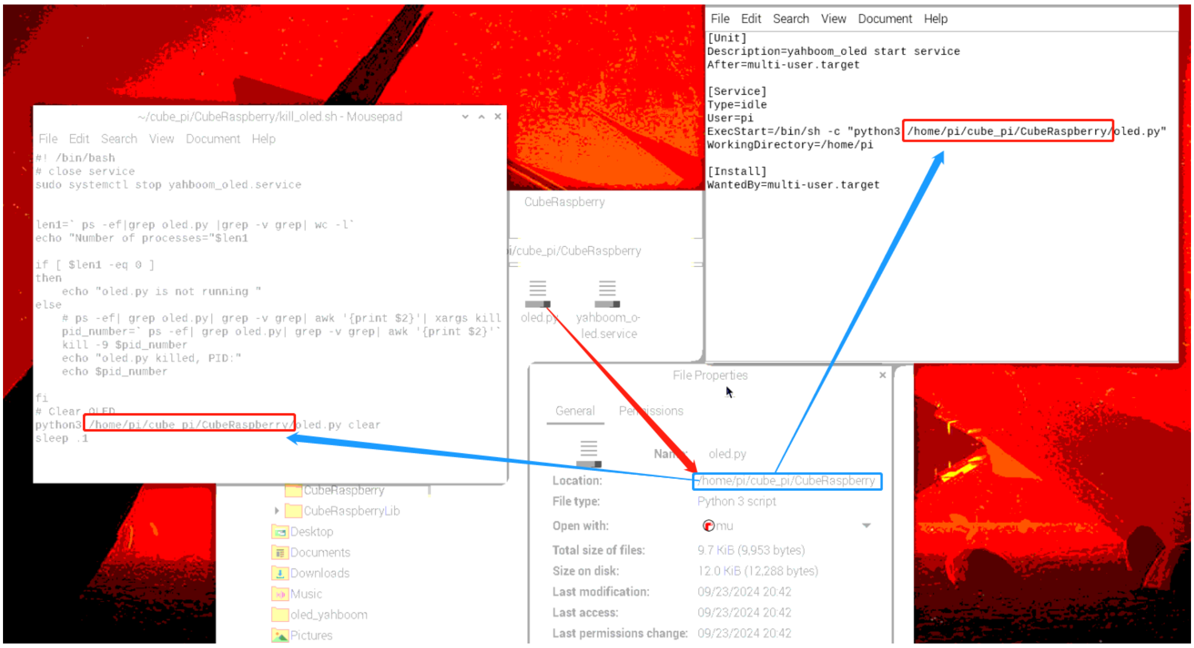
Install the Raspberry Pi chassis according to the assembly video tutorial. You can also refer to the "Raspberry Pi Chassis_Hardware Wiring" tutorial document for installation. The chassis expansion board and related hardware connections are shown here.



Chassis expansion board	3Pin interface	2Pin interface
	RGB light bar	Fan

3. Set up the boot-up service

Copy the CubeRaspberry.zip file in the data to the Raspberry Pi motherboard system. After unzipping the file, you need to change the path names in the kill_oled.sh and yahboom_oled.service files according to your own directory; if the paths are different, modify the location in the figure below.



```
#! /bin/bash
# close service
sudo systemctl stop yahboom_oled.service


len1=` ps -ef|grep oled.py |grep -v grep| wc -l`
echo "Number of processes="$len1

if [ $len1 -eq 0 ]
then
    echo "oled.py is not running "
else
    # ps -ef| grep oled.py| grep -v grep| awk '{print $2}'| xargs kill
    pid_number=` ps -ef| grep oled.py| grep -v grep| awk '{print $2}'`
    kill -9 $pid_number
    echo "oled.py killed, PID:"
    echo $pid_number

fi
# Clear OLED
python3 /home/pi/cube_pi/CubeRaspberry/oled.py clear
sleep .1
```

```
[Unit]
Description=yahboom_oled start service
After=multi-user.target

[Service]
Type=idle
User=pi
ExecStart=/bin/sh -c "python3 /home/pi/cube_pi/CubeRaspberry/oled.py"
WorkingDirectory=/home/pi

[Install]
WantedBy=multi-user.target
```

Name: oled.py
Location: /home/pi/cube_pi/CubeRaspberry
File type: Python 3 script
Open with: mu
Total size of files: 9.7 KiB (9,953 bytes)
Size on disk: 12.0 KiB (12,288 bytes)
Last modification: 09/23/2024 20:42
Last access: 09/23/2024 20:42
Last permissions change: 09/23/2024 20:42

The following command operations may require password input, just remember to enter

1. Copy files

Open the terminal in the CubeRaspberry folder and use the following command to copy yahboom_oled.service to the /etc/systemd/system path.

```
sudo cp yahboom_oled.service /etc/systemd/system
```

2. Manage configuration

Use the systemctl tool to load and manage configuration.

```
systemctl daemon-reload
```

3. Manage service

Manage the yahboom_oled service.

- Start yahboom_oled service

```
systemctl start yahboom_oled
```

- Stop yahboom_oled service

```
systemctl stop yahboom_oled
```

- Restart yahboom_oled service

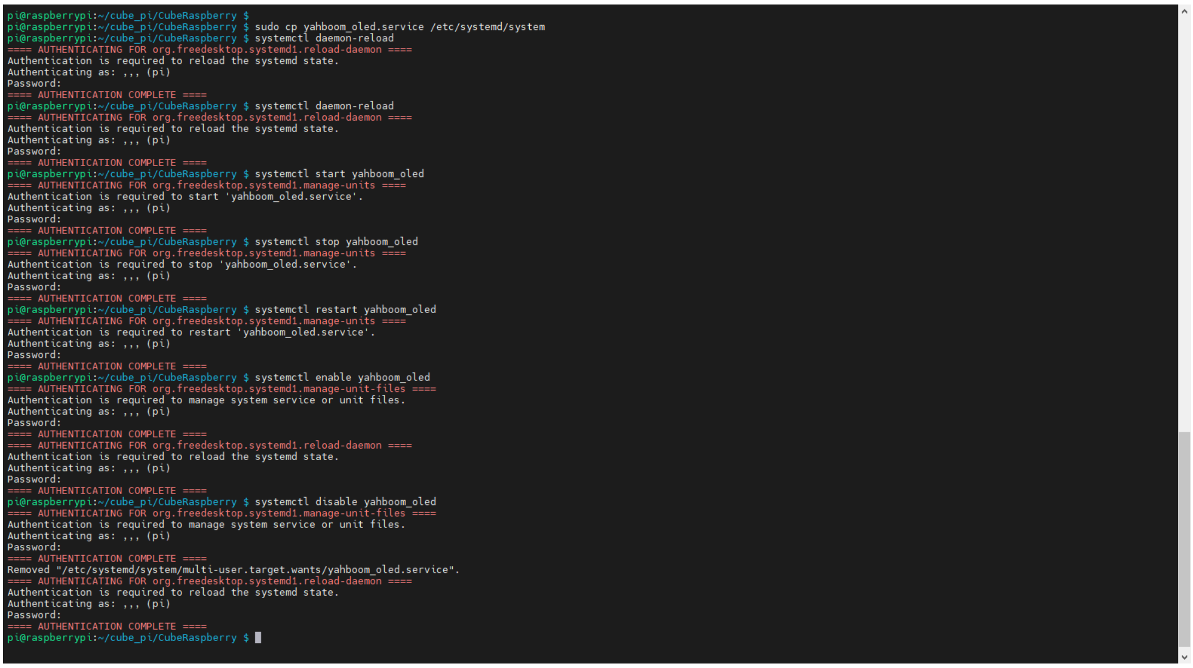
```
systemctl restart yahboom_oled
```

- Set yahboom_oled service to start automatically at boot

```
systemctl enable yahboom_oled
```

- Disable yahboom_oled service to start automatically at boot

```
systemctl disable yahboom_oled
```



```
pi@raspberrypi:~/cube_pi/CubeRaspberry $
pi@raspberrypi:~/cube_pi/CubeRaspberry $ sudo cp yahboom_oled.service /etc/systemd/system
pi@raspberrypi:~/cube_pi/CubeRaspberry $ systemctl daemon-reload
==== AUTHENTICATING FOR org.freedesktop.systemd1.reload-daemon ====
Authentication is required to reload the systemd state.
Authenticating as: ,,, (pi)
Password:
==== AUTHENTICATION COMPLETE ====
pi@raspberrypi:~/cube_pi/CubeRaspberry $ systemctl daemon-reload
==== AUTHENTICATING FOR org.freedesktop.systemd1.reload-daemon ====
Authentication is required to reload the systemd state.
Authenticating as: ,,, (pi)
Password:
==== AUTHENTICATION COMPLETE ====
pi@raspberrypi:~/cube_pi/CubeRaspberry $ systemctl start yahboom_oled
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ====
Authentication is required to start 'yahboom_oled.service'.
Authenticating as: ,,, (pi)
Password:
==== AUTHENTICATION COMPLETE ====
pi@raspberrypi:~/cube_pi/CubeRaspberry $ systemctl stop yahboom_oled
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ====
Authentication is required to stop 'yahboom_oled.service'.
Authenticating as: ,,, (pi)
Password:
==== AUTHENTICATION COMPLETE ====
pi@raspberrypi:~/cube_pi/CubeRaspberry $ systemctl restart yahboom_oled
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ====
Authentication is required to restart 'yahboom_oled.service'.
Authenticating as: ,,, (pi)
Password:
==== AUTHENTICATION COMPLETE ====
pi@raspberrypi:~/cube_pi/CubeRaspberry $ systemctl enable yahboom_oled
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-unit-files ====
Authentication is required to manage system service or unit files.
Authenticating as: ,,, (pi)
Password:
==== AUTHENTICATION COMPLETE ====
==== AUTHENTICATING FOR org.freedesktop.systemd1.reload-daemon ====
Authentication is required to reload the systemd state.
Authenticating as: ,,, (pi)
Password:
==== AUTHENTICATION COMPLETE ====
pi@raspberrypi:~/cube_pi/CubeRaspberry $ systemctl disable yahboom_oled
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-unit-files ====
Authentication is required to manage system service or unit files.
Authenticating as: ,,, (pi)
Password:
==== AUTHENTICATION COMPLETE ====
Removed "/etc/systemd/system/multi-user.target.wants/yahboom_oled.service".
==== AUTHENTICATING FOR org.freedesktop.systemd1.reload-daemon ====
Authentication is required to reload the systemd state.
Authenticating as: ,,, (pi)
Password:
==== AUTHENTICATION COMPLETE ====
pi@raspberrypi:~/cube_pi/CubeRaspberry $
```

Note: The picture shows the effect of each command after running. Run the command according to your needs.

4. Run kill_oled.sh

Running the kill_oled.sh file will stop the yahboom_oled service and the yahboom_oled service process, and the OLED display will not display anything.

- Add executable permissions

```
chmod +x kill_oled.sh
```

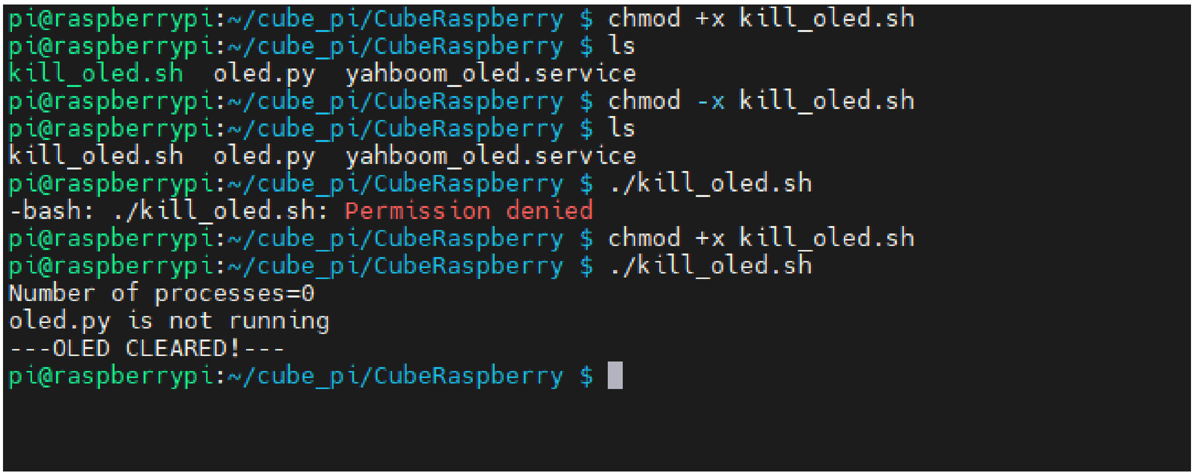
- Cancel executable permissions

```
chmod -x kill_oled.sh
```

Note: Generally, the file name of the executable file is in green font

- Run the kill_oled.sh file

```
./kill_oled.sh
```



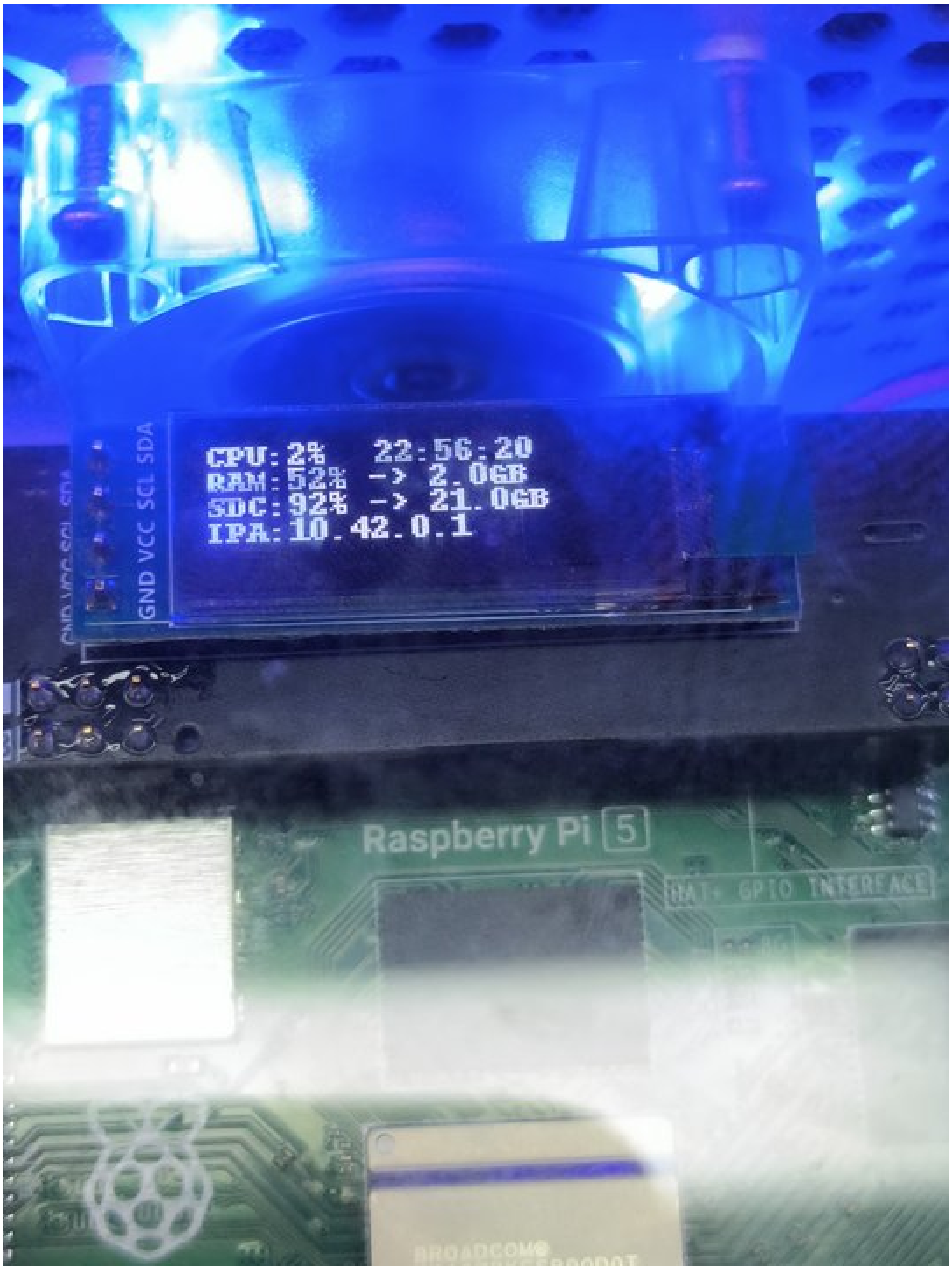
```
pi@raspberrypi:~/cube_pi/CubeRaspberry $ chmod +x kill_oled.sh
pi@raspberrypi:~/cube_pi/CubeRaspberry $ ls
kill_oled.sh  oled.py  yahboom_oled.service
pi@raspberrypi:~/cube_pi/CubeRaspberry $ chmod -x kill_oled.sh
pi@raspberrypi:~/cube_pi/CubeRaspberry $ ls
kill_oled.sh  oled.py  yahboom_oled.service
pi@raspberrypi:~/cube_pi/CubeRaspberry $ ./kill_oled.sh
-bash: ./kill_oled.sh: Permission denied
pi@raspberrypi:~/cube_pi/CubeRaspberry $ chmod +x kill_oled.sh
pi@raspberrypi:~/cube_pi/CubeRaspberry $ ./kill_oled.sh
Number of processes=0
oled.py is not running
---OLED CLEARED!---
pi@raspberrypi:~/cube_pi/CubeRaspberry $ 
```

- Running effect

4. Experimental phenomenon

Follow the steps in the tutorial to realize the automatic startup of the yahboom_oled service.

After the Raspberry Pi series motherboard is turned on, the OLED display will display relevant information of the system, and the fan and RGB light strip will also work in the corresponding state.



```
CPU:2%    22:56:20
RAM:52%  -> 2.0GB
SDC:92%  -> 21.0GB
IPA:10.42.0.1
```