

嵌入式 Linux 开发快速入门指南

深圳亮钻科技

2024 年 11 月 7 日

Contents

1 Linux 烧录	3
1.1 烧录准备	3
1.2 烧录操作	3
1.2.1 电脑安装主板驱动工具	3
1.2.2 系统烧录	4
1.2.3 常见问题	7
1.3 U 盘刷补丁	7
1.3.1 主板端检查项	7
1.3.2 电脑端检查项	7
2 Linux 主机基本配置	9
2.1 Linux 系统登录说明	9
2.1.1 系统账号和密码	9
2.1.2 ssh 允许 root 用户登录	9
2.1.3 自动登录 (默认普通用户自动登录)	9
2.1.4 禁用/启用桌面	9
2.1.5 电源指示灯	9
2.1.6 登录系统的方式	10
2.2 网络连接	10
2.2.1 以太网连接	10
2.2.2 WiFi 连接测试 (两种: 界面点击右上角、命令方式如下)	10
2.2.3 设置静态 IP	10
2.2.4 主板启动后查看 IP	10
2.2.5 网络连通性测试	11
2.2.6 网络 ADB	11
2.3 其他	11
2.3.1 开启/关闭图形界面	11
2.3.2 显示相关命令	11
3 常用工具及外设测试	12
3.1 PC 端工具	12
3.2 主板端工具	12
3.3 外设测试	12
3.3.1 CAN 测试	12
3.3.2 GPIO 测试	12
3.3.3 声卡	13
3.3.4 蓝牙	15
3.3.5 RTC 测试	16
3.3.6 DOCKER IPTABLES KVM K8S AUDIT	17
3.3.7 以太网、WiFi、热点、网络转发	18
3.3.8 4G 模块上网	18
3.3.9 旋转屏幕方向、触摸方向	19
3.3.10 GPU、DRM、NPU	20
3.3.11 根系统备份	21

3.3.12 自启动	21
3.3.13 视频编解码	21
3.3.14 分区	22
3.3.15 修改网卡名, 以传统方式命名	22
3.3.16 设置中文环境、安装中文输入法	22
3.3.17 开启关闭看门狗	23
3.3.18 日志容量	23
3.3.19 连接 VNC	23
3.3.20 <code>xrandr</code> 设置屏幕	24
3.3.21 禁用 WiFi 蓝牙	25
3.3.22 HDMI-IN 输入测试	25
3.3.23 Debian 禁止待机	25
3.3.24 性能测试	26
4 交叉编译环境的建立 (PC 端)	27
5 目标程序的部署和调试 (主板端)	28
6 常用调试命令	29
7 文档版本	29

1 Linux 烧录

具体请参考: [BD 系列主机系统烧录说明.pdf](#) [RK3288 Linux 系统烧录说明.docx](#) [产品规格书](#) 相关工具请联系销售、FAE 获取。

视频教学链接: https://www.bilibili.com/video/BV1K24y1k7Xm/?spm_id_from=333.999.0.0&vd_source=36dd97c209267d0152eeac24667e5cf0

常见问题及解答: Ubuntu 系统常见问题及解答, 快 Get 起来! (qq.com)

软件开发人员请参考: [Rockchip Linux 软件开发指南.pdf](#)

1.1 烧录准备

硬件: 电脑(笔记本)、双公头 USB 数据线、主机/主板

软件: 驱动包(在本压缩包中)、固件烧录工具包、相关补丁(咨询 FAE 获取)



Figure 1: 烧录准备.png

1.2 烧录操作

1.2.1 电脑安装主板驱动工具

(如果之前安装过其他驱动, 请卸载, 再重新安装)

- 驱动工具解压出来, 双击打开【DriverInstall】应用程序。

📁 ADBDriver	2020/11/10 14:13	文件夹
📁 bin	2020/11/10 14:14	文件夹
📁 Driver	2020/11/10 14:15	文件夹
📁 Log	2022/11/30 14:24	文件夹
📄 config	2014/6/3 15:38	配置设置
📅 DriverInstall	2020/11/10 14:15	应用程序
📄 Readme	2018/1/31 17:44	文本文档



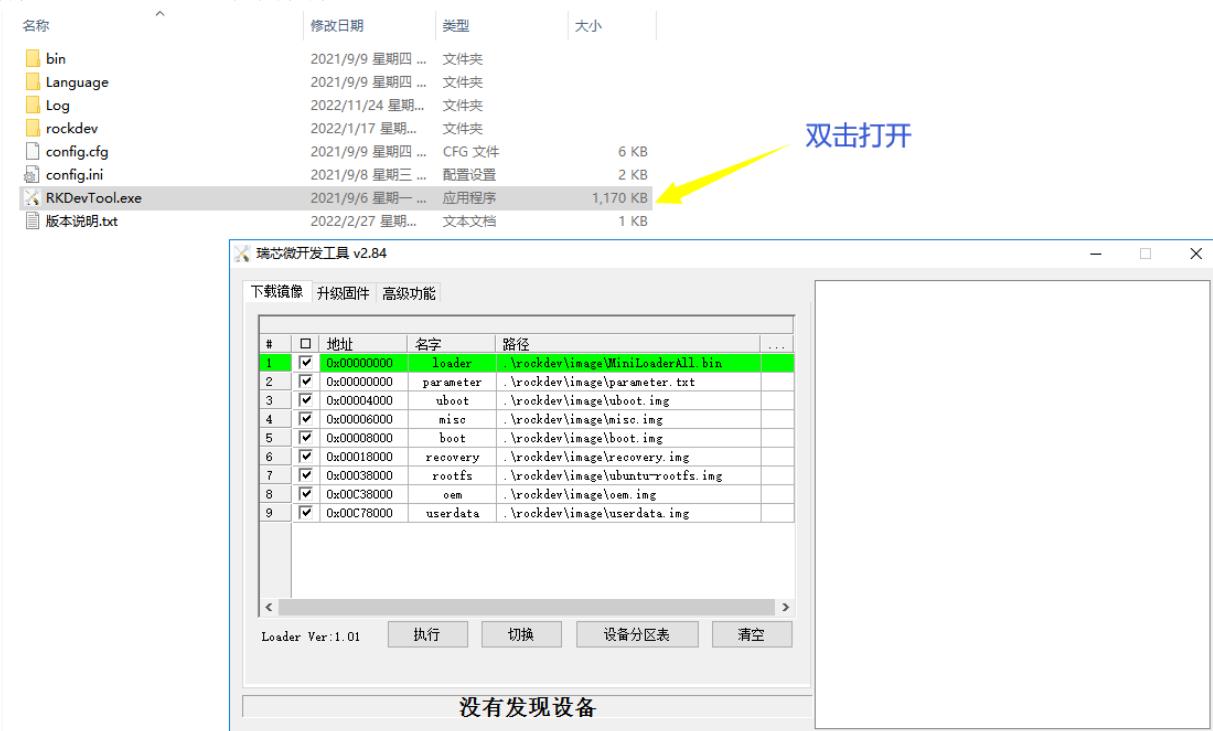
- 点击【驱动安装】，电脑自动安装主板驱动，大约 30s，驱动安装成功。

ADBDriver	2020/11/10 14:13	文件夹
bin	2020/11/10 14:14	文件夹
Driver	2020/11/10 14:15	文件夹
Log	2022/11/30 14:24	文件夹
config	2014/6/3 15:38	配置设置
DriverInstall	2020/11/10 14:15	应用程序
Readme	2018/1/31 17:44	文本文档



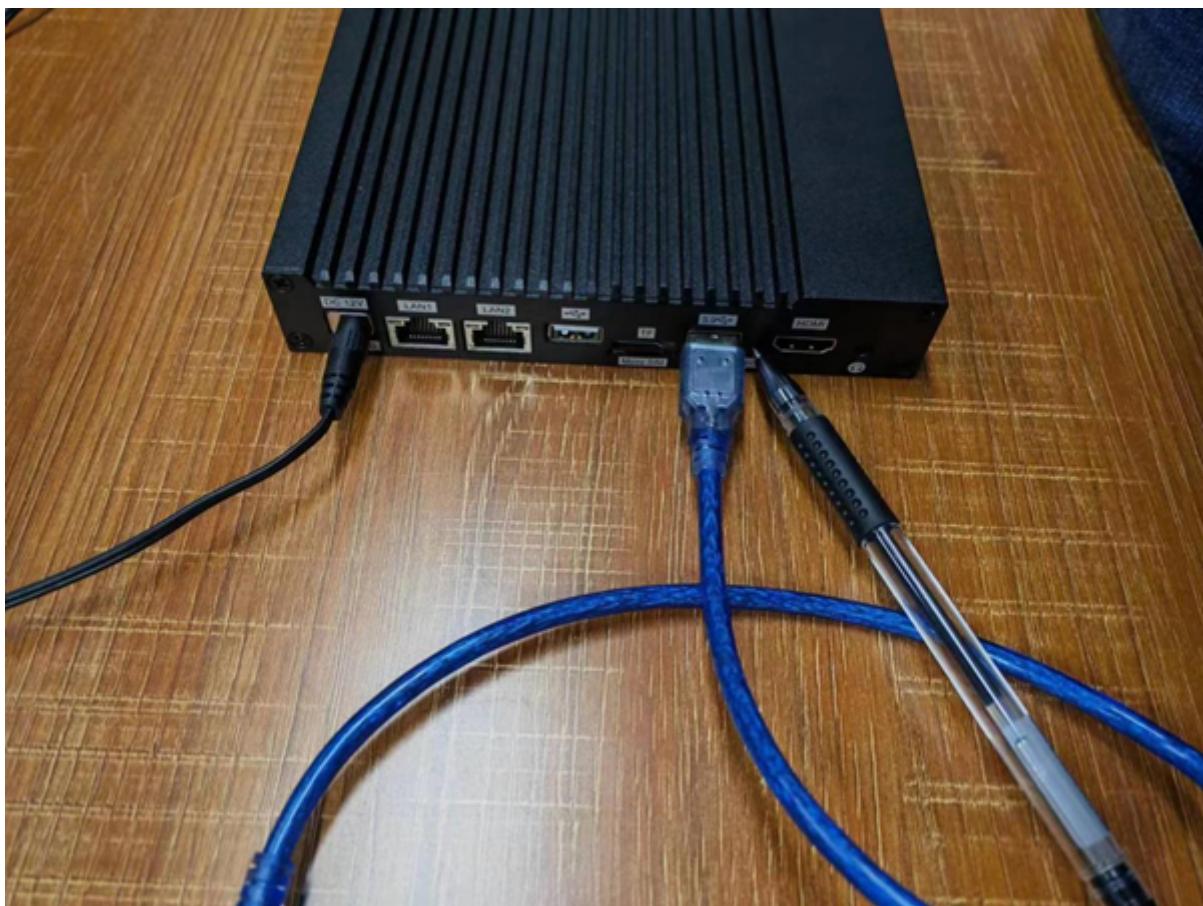
1.2.2 系统烧录

- rk3568dt-l13-oem-ubuntu20-lxqt-240719 (系统包仅为示例，具体请和 FAE 确认) 的烧录工具包解压出来，双击打开【RKDevTool】应用程序。



- 用手或者圆珠笔按住主机/主板的【uboot 按键】不放，主机 DC12V 通电，烧录程序显示“发现一个 LOADER 设备”，松开手，然后点击【执行】，系统自动烧录，等待 2~3 分钟，烧录程序显示“下载完成”。如图接线：
用圆珠笔按住主机的【uboot 按键】不放，主机 DC12V 通电。





烧录程序显示“发现一个 LOADER 设备”，松开手，然后点击【执行】，系统自动烧录

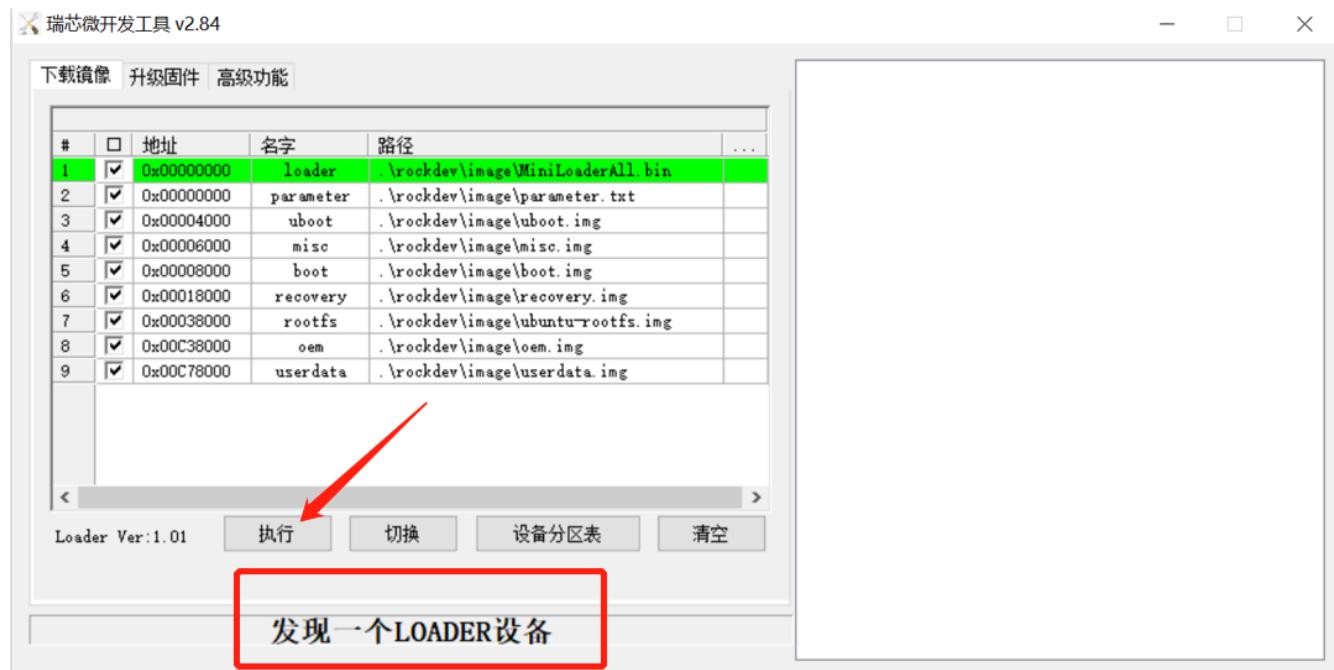


Figure 2: 发现一个 loader 设备.png

- 至此，主板的系统烧录完成，断开数据线，主板重新上电点屏即可。

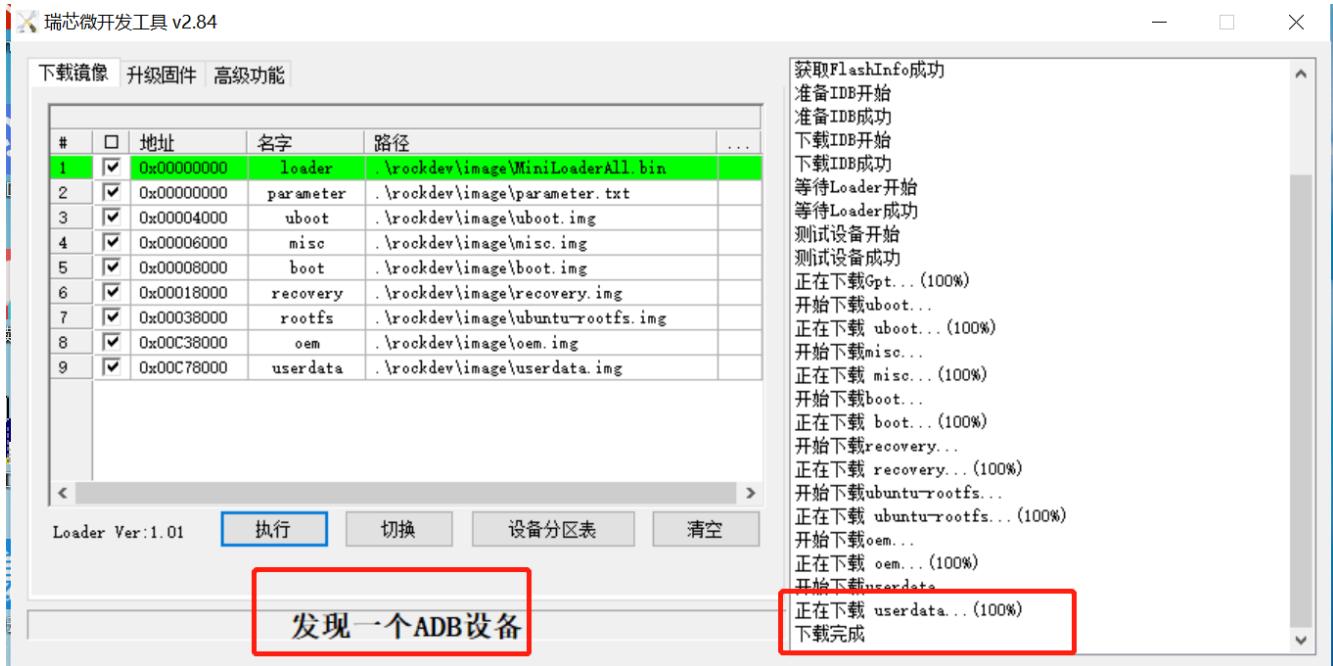


Figure 3: 烧录完成.png

1.2.3 常见问题

Q: 用手或者圆珠笔按住主机的【uboot 按键】不放，主机 DC12V 通电，烧录程序没有“发现一个 LOADER 设备”。

A: 则需要检查硬件是否有问题，电脑的 USB 口是否正常，线材是否正常，直接换个电脑重新安装驱动，更换线材再尝试，数据线有问题的概率最大，能数据传输的线材即可。

如遇到其他问题，请咨询技术支持。

1.3 U 盘刷补丁

- 若时间紧迫，不便自行操作，可发向日葵、ToDesk 远程识别码、开语音视频等，远程协助操作。(具体咨询 FAE)
- 首先参考补丁里面的使用须知即可，若仍刷补丁失败，再详细排查。
- FAT 格式工具下载地址 [<http://9ioldgame.com/software/guiformat.html>]。使用时需仔细检查要格式化的盘符，以防造成数据损失。
- 鼠标停在优盘图标上面，右键选择属性，查看优盘格式类型，非 FAT/FAT32 时，该工具用于格式化优盘。
- 启动盘格式工具为 DiskGenius.exe 若有需要，可提供。特别注意：数据宝贵，操作之前，需反复确认。

1.3.1 主板端检查项

- U 盘必须是 FAT 或 FAT32 格式、不能是其他格式 (其他格式会出现概率可行)
- U 盘不能是当作启动盘的优盘 (若 U 盘是启动盘，则一般有 2 个分区，需换其他 U 盘刷补丁，或格式工具处理)
- 压缩包解压后的文件按照原本的目录结构拷贝到 U 盘根目录，如果解压后的文件夹跟压缩包同名，要把该文件夹的下级目录内容放在优盘根目录
- 主板断电后把 U 盘接到主板上的 USB 接口上，然后重新开机；等 1~2 分钟 (正常情况下为 20 秒左右)，主板上的灯变成红色或熄灭或屏幕变黑或 ADB 断开，出现以上标志，迅速拔掉优盘，然后断电重新上电开机。屏幕变黑不一定代表刷补丁完成，因为可能会出现重启的情况。以其他方式为优先判断方式；优先参考‘使用须知.txt’
- ADB 的状态 (连接、断开) 可通过瑞芯微工具查看

1.3.2 电脑端检查项

- 电脑端若能够连接 ADB，可通过执行 adb shell，验证补丁是否成功刷入。
- 若刷的是内核补丁，既补丁中含有 image\boot.img 文件，通过执行 adb shell，然后 uname -a 查看内核版本时间，并和补丁包中的-说明-内核版本对比。
- 若刷的是屏参补丁，可通过执行 adb shell，然后 dmesg | grep mode= 查看屏参信息

- 尽量不插在具有 OTG 功能的 USB 口，因为当该端口是 OTG 模式时，无法识别优盘，该端口靠近按键在左侧 (Fi 系列一般是韦线上面的 USB 口为 OTG HOST)。
- 若接有屏幕，插上优盘，一般由文件夹弹出

注：24 年 8 月份之后的 Linux 补丁基本都为关机不重启，可查阅补丁里面的“使用须知.txt”
为了刷补丁后有更直观的现象，近期已改为关机亮红灯/灭灯-不重启 (灯：指电源指示灯)。

2 Linux 主机基本配置

2.1 Linux 系统登录说明

2.1.1 系统账号和密码

系统	普通用户	密码	超级用户	密码
Ubuntu20.04/22.04	teamhd	root	root	root
Debian10/11	linaro	linaro	root	root
Ubuntu18.04	teamhd		root	

输入密码时，屏幕上不会显示输入的密码，输入完后直接回车即可。

注: Ubuntu18.04 默认账户: teamhd, 密码为空, 可以自行更改账户密码和账户添加。

修改密码命令: sudo passwd teamhd

Root 账号修改密码: sudo passwd root

2.1.2 ssh 允许 root 用户登录

```
sudo sed -i 's/#\!PermitRootLogin prohibit-password/PermitRootLogin yes/' /etc/ssh/sshd_config  
sudo systemctl restart ssh
```

2.1.3 自动登录 (默认普通用户自动登录)

```
sudo nano /etc/gdm3/custom.conf  
  
# Enabling automatic login  
# AutomaticLoginEnable = true  
# AutomaticLogin = user1
```

取消注释 `AutomaticLoginEnable` 和 `AutomaticLogin` 行，并将 `AutomaticLogin` 的值更改为您要自动登录的用户名。例如:

```
AutomaticLoginEnable = true  
AutomaticLogin = your_username
```

如果使用 root 用户登录桌面系统是无法使用右上角的 pulseaudio 来管理音频设备，因为 pulseaudio 不允许在 root 用户下运行

2.1.4 禁用/启用桌面

禁用桌面
sudo systemctl disable lightdm.service
sudo reboot

启用桌面
sudo systemctl start lightdm.service
sudo systemctl enable lightdm.service

2.1.5 电源指示灯

- Ubuntu20.04 最新系统电源指示灯为呼吸灯模式，可以作为系统是否正常运行的判断方式。
- 24 年 8 月份之后的 Linux 补丁基本都为关机不重启，刷补丁后常亮绿灯；为了更直观，近期已改为关机亮红灯/灭灯-不重启 (灯：指电源指示灯)。

2.1.6 登录系统的方式

- 可以通过 ssh 远程登录 Linux 系统
- 如果主板连接好了调试串口，可以使用串口终端登录 Linux 系统（3588 默认启用调试串口；3568 默认不启用）
- 如果主板连接 HDMI 显示器，可以通过 HDMI 显示的终端登录到 Linux 系统

2.2 网络连接

2.2.1 以太网连接

- 有线网络默认是 DHCP 获取地址，插上网络自动获取地址，桌面状态栏右下角有网络设置的图标，点击“Connection Information”可以查看相关网络的连接信息。

2.2.2 WiFi 连接测试（两种：界面点击右上角、命令方式如下）

```
# 查看网络设备  
sudo nmcli dev  
  
# 扫描周围的 WiFi 热点  
sudo nmcli dev wifi  
  
# 开启 WiFi(可选)  
sudo nmcli r wifi on  
  
# 关闭 WiFi(可选)  
sudo nmcli r wifi off  
  
# 连接 WiFi(wifi_name 改为实际的 WiFi 名 wifi_passwd 改为实际的密码)  
sudo nmcli dev wifi connect "wifi_name" password "wifi_passwd" ifname wlan0
```

2.2.3 设置静态 IP

分为 Netplan 和 nmcli 两种。以 nmcli 为例：

```
# 查看网络  
nmcli con show  
  
# 设置静态 IP "Wired connection 1" 表示设置以太网口的静态 IP 地址  
sudo nmcli con mod "Wired connection 1" \  
    ipv4.addresses "192.168.1.110" \  
    ipv4.gateway "192.168.1.1" \  
    ipv4.dns "8.8.8.8" \  
    ipv4.method "manual"  
  
# 重启系统  
sudo reboot  
  
# 查看 Ip  
ip addr show
```

2.2.4 主板启动后查看 IP

- 接 HDMI 显示器，然后登录系统使用 ip addr show 命令查看
- 在调试串口终端输入 ip addr show 命令来查看
- 如果没有调试串口，也没有 HDMI 显示器，还可以通过路由器的管理界面来查看开发板网口的 IP 地址。

2.2.5 网络连通性测试

```
ping www.baidu.com
```

2.2.6 网络 ADB

```
adb connect 主板 ip 地址
```

2.3 其他

2.3.1 开启/关闭图形界面

```
# 关机图形界面  
sudo systemctl set-default multi-user.target  
sudo reboot  
  
# 开启用户图形界面。  
sudo systemctl set-default graphical.target  
sudo reboot
```

2.3.2 显示相关命令

```
# 查看屏参  
dmesg | grep mode=  
  
# 查看显示状态  
xrandr  
  
# 正常显示/90/180/270  
xrandr -o normal  
xrandr -o left  
xrandr -o inverted  
xrandr -o right  
  
# 查看显示日志  
cat /var/log/Xorg.0.log
```

3 常用工具及外设测试

3.1 PC 端工具

工具	获取
串口调试工具 MobaXterm	https://mobaxterm.mobatek.net
ADB 调试工具	公司-网盘 > 工具-安卓 > ADB 驱动 (网盘链接, 咨询 FAE)
系统包、补丁	咨询 FAE
ssh	https://www.putty.org/

3.2 主板端工具

请勿执行 `apt upgrade`, 主板目前不支持联网更新内核!!!

联网后执行如下命令更新软件包列表 `sudo apt update -y`

工具/软件	获取
摄像头测试工具	<code>sudo apt-get install -y cheese</code>
串口工具	<code>sudo apt-get install -y cutecom</code>
CAN 工具	<code>sudo apt install -y can-utils</code>

3.3 外设测试

测试之前请先查阅[主板产品规格书.pdf](#) (咨询 FAE)

3.3.1 CAN 测试

测试脚本参考 `can-utils.md`(咨询 FAE)

功能	命令
查询当前网络设备 (一般可显示 1 个 CAN 设备节点)	
刷对应补丁可显示 2 个; CAN 功能需确保硬件有焊接 CAN 芯片)	<code>ifconfig -a</code>
关闭 CAN0	<code>ip link set can0 down</code>
设置比特率 1MHz	<code>ip link set can0 type can bitrate 1000000</code>
打印 can0 信息	<code>ip -details link show can0</code>
启动 CAN0	<code>ip link set can0 up</code>
发送 (标准帧, 数据帧, ID:123, date:DEADBEEF)	<code>cansend can0 123#DEADBEEF</code>
开启打印, 等待接收	<code>candump can0</code>

3.3.2 GPIO 测试

功能/说明	命令/信息
3588 GPIO 引脚编号	#J38 K1=#116、K2=#125、K3=#135、K4=#27、K5=#152、K6=#153、K7=#154、K8=#155
3568 GPIO 引脚编号	#J38 K1=#88、K2=#107、K3=#89、K4=#108、K5=#90、K6=#109、K7=#91、K8=#110
查看方向 (以 3588 K8 为例)	cat /sys/class/gpio/gpio110/direction
设置为输出	echo out > /sys/class/gpio/gpio110/direction
设置为输入	echo in > /sys/class/gpio/gpio110/direction
读取当前的电平状态	cat /sys/class/gpio/gpio110/value
输出低电平	echo 0 >/sys/class/gpio/gpio110/value
输出高电平	echo 1 >/sys/class/gpio/gpio110/value

测试举例:

以 3588 K8 为例 (测试其他 gpio, 需修改 pio 为对应的编号) :

```
pio=110
echo $pio > /sys/class/gpio/export
echo out > /sys/class/gpio/gpio$pio/direction

# 设置高电平
echo 1 > /sys/class/gpio/gpio$pio/value

# 查看所有 IO 状态
cat /sys/kernel/debug/gpio

# 设置低电平
echo 0 > /sys/class/gpio/gpio$pio/value

# 查看所有 IO 状态
cat /sys/kernel/debug/gpio
```

引脚计算:

```
# 引脚计算
bank = 0;      // GPIO0_B5 => 0, bank [0,4]
group = 1;      // GPIO0_B5 => 1, group {(A=0), (B=1), (C=2), (D=3)}
X = 5;          // GPIO0_B5 => 5, X [0,7]
number = group * 8 + X = 1 * 8 + 5 = 13;
pin = bank * 32 + number = 0 * 32 + 13 = 13;
```

3.3.3 声卡

ubuntu 系统

在 Linux 系统上, 通常是通过 PulseAudio 来管理声音设备和流的, 默认声卡通常也是由 PulseAudio 来管理。普通用户下, 执行如下命令查看进程:

```
ps aux | grep pulseaudio
```

查看系统上可用的声卡及其标识符:

```
pacmd list-sinks | grep -e 'name:' -e 'index:'
```

临时设置声卡 (以实际名称为准):

```
pactl set-default-sink alsa_output.platform-es7144-sound.stereo-fallback
```

执行测试命令 (也可界面打开浏览器输入 music.taihe.com 播放音乐测试):

```
aplay /usr/share/sounds/alsa/Rear_Right.wav
```

用 pactl 来调节音量。音量的值以百分比表示。这个百分比是相对于最大音量的百分比 (一般是 200%)。

例如, 要增加音量, 可以使用以下命令 (1 代表当前声卡索引号为 1):

```
pactl set-sink-volume alsa_output.platform-rk809-sound.stereo-fallback +5% 或
```

```
pactl set-sink-volume 1 +5%
```

要增加音量, 可以使用以下命令:

```
pactl set-sink-volume alsa_output.platform-rk809-sound.stereo-fallback +5% 或
```

```
pactl set-sink-volume 1 -5%
```

直接设置音量, 可以使用以下命令:

```
pactl set-sink-volume alsa_output.platform-rk809-sound.stereo-fallback 50% 或
```

```
pactl set-sink-volume 1 50%
```

设置声卡重启后生效: 普通用户下创建声卡服务 (仅为示例, 默认提供补丁解决, 相关参考文件可联系 FAE 获取)

```
mkdir -p /home/teamhd/.config/systemd/user/default.target.wants
```

```
chmod -R 777 /home/teamhd/.config/systemd
```

```
cp -f /mnt/usb_storage/set-default-audio.service \
/home/teamhd/.config/systemd/user/set-default-audio.service
chmod 666 /home/teamhd/.config/systemd/user/set-default-audio.service
```

```
cp -f /mnt/usb_storage/set-default-audio /usr/local/bin/set-default-audio
chmod 777 /usr/local/bin/set-default-audio
```

```
sudo -u teamhd XDG_RUNTIME_DIR=/run/user/$(id -u teamhd) systemctl --user \
enable /home/teamhd/.config/systemd/user/set-default-audio.service
```

声卡相关命令:

```
# 获取录音设备
```

```
arecord -l
```

```
# 获取播放设备
```

```
aplay -l
```

查看声卡驱动目录

```
ls /dev/snd/
```

查看声卡驱动目录

```
teamhd@teamhd:~$ ls /dev/snd
by-path controlC0  controlC1  pcmC0D0p  pcmC1D0c  pcmC1D0p  seq  timer
```

controlC0: 用于声卡控制, C0 表示声卡 0, 对应上文的 HDMI 声音输出。

controlC1: 用于声卡控制, C1 表示声卡 1, 对应上文的板载音频处理芯片 rockchip-es8388。

pcmC0D0p: 用于播放的 pcm 设备, 最后面的 “p” 是 playback 的缩写, 表示放音。

pcmC1D0c: 用于录音的 pcm 设备, 最后面的 “c” 是 capture 的缩写, 表示录音

pcmC1D0p: 用于播放的 pcm 设备, 最后面的 “p” 是 playback 的缩写, 表示放音。

timer : 定时器

by-path: 保存着设备的对应关系

```
teamhd@teamhd:~$ ls -l /dev/snd/by-path/
```

总用量 0

```
lrwxrwxrwx 1 root root 12 9月 26 01:58 platform-es8323-sound -> ../controlC1  
lrwxrwxrwx 1 root root 12 9月 26 01:58 platform-hdmi-sound -> ../controlC0
```

可以看到每个 control 之间都有一个对应关系，而这些对应关系正好对应上文的 card 0 , card 1

```
# 获取对应的控制器配置的值  
amixer -c $card_id cget $id  
# 设置对应的控制器配置的值  
amixer -c $card_id 1 cset $id $value  
  
# 例：获取 card1 的声音播放音量大小  
root@lubancat:~# amixer -c 1 cget numid=3  
numid=3,iface=MIXER,name='Playback Volume'  
; type=INTEGER,access=rw---R--,values=2,min=0,max=252,step=0  
: values=252,252  
| dBScale-min=-95.00dB,step=0.37dB,mute=0  
  
1. values=2 : 需要配置两个值  
2. min=0 : 可配置的最小值为 0  
3. max=252 : 可配置的最大值为 252  
4. values=252,252: 当前的值为 252,252(左右声道)  
  
# 设置 card1 声音播放音量大小为 150,150  
root@lubancat:~# amixer -c 1 cset numid=3 "150,150"  
numid=3,iface=MIXER,name='Playback Volume'  
; type=INTEGER,access=rw---R--,values=2,min=0,max=252,step=0  
: values=150,150  
| dBScale-min=-95.00dB,step=0.37dB,mute=0
```

3.3.4 蓝牙

蓝牙经典模式测试

可接 HDMI 显示器，在界面点击 bluetooth 图标-> 设备-> 搜索 来进行扫描连接。

蓝牙 BLE 模式测试手机下载 BLE 调试助手，切换到从机模式。

相关命令：

```
sudo rfkill block bluetooth # 关闭蓝牙  
sudo rfkill unblock bluetooth # 开启蓝牙  
hciconfig -a # 查看版本  
bluetoothctl # 进入蓝牙调试工具  
power on # 打开蓝牙适配器  
scan on # 扫描到手机蓝牙地址  
scan off # 关闭扫描
```

注意事项：

```
# 注意天线是否接上、手机和主板的距离  
# 使用手机安装 BLE 调试助手进行 BLE 测试，软件从应用市场下载  
# BLE 验证的步骤：  
1. 开关蓝牙  
2. hciconfig -a 保证 Features 字段的十六制不能全是'0x00'  
3. 若全为 0x00，执行 hciconfig hci0 down && sleep 5 && hciconfig hci0 up && hciconfig -a  
4. 5 秒后上面的指令执行完毕，看显示的 Features 字段  
5. 显示不是全'0x00'后，代表 ble 可用。  
6. 开始执行测试脚本（非必需，若需 demo，可联系 FAE 获取）
```

```
获取 uuid  
main-get-uuid.py  
循环发送数据  
main-send-message-loop.py  
发送数据  
main-send-message.py  
循环发送数据并可接收  
main-send-receive-loop.py
```

```
# 手机打开 BLE 调试助手，点击左上方菜单，选切换到设备模式，点击‘开启广播’，关闭两个 hex 按钮，  
等待自动连接并接收到数据，发送数据框填入数据点击发送后，adb shell 命令行也可接收到数据。
```

3.3.5 RTC 测试

注意：复制 date 设置时间的命令前面有个 '#' 号要去掉

参考文档：Linux 设置时间-240802.md (咨询 FAE)

显示当前的系统时间和日期设置

```
timedatectl
```

timedatectl 返回参数的含义：

- Local time 显示的是当前时区的时间。
- Universal time 显示的是 UTC 时间，与本地时间有 8 小时的差异。
- RTC time 显示的是从硬件时钟读取的时间，通常与 UTC 时间一致。
- Time zone 显示系统当前的时区设置及其偏移量。
- System clock synchronized 显示系统时钟是否与 NTP 服务器同步。
- NTP service 显示 NTP 服务的运行状态。
- RTC in local TZ: yes 表示使用本地时区时间 no 表示 RTC 使用 UTC 时间

设置时区为上海

```
sudo ln -sf /usr/share/zoneinfo/Asia/Shanghai /etc/localtime  
或
```

```
sudo timedatectl set-timezone Asia/Shanghai
```

```
##### 不联网设置时间 #####
```

1. 先设置准确时间

```
root@teamhd:/# date -s "20230303 17:10:00"  
Fri Mar 3 17:10:00 UTC 2023
```

2. 执行 hwclock -w 将时间保存到硬件 rtc 中

```
root@teamhd:/# hwclock -w
```

断 2 分钟多以后，重启主板，重新查看时间，能准确无误更新：

```
C:\Users\Mooer>adb shell  
root@teamhd:/# date  
Fri Mar 3 17:12:44 UTC 2023
```

```
##### 联网设置时间不成功（需禁用对应服务，可不测试） #####
```

1、若未安装 ntp 软件，在联网状态下，要设置时间，需关闭 systemd-timesyncd 服务

```
sudo systemctl stop systemd-timesyncd  
sudo systemctl disable systemd-timesyncd
```

2、若安装 ntp 软件，在联网状态下，要设置时间，关闭 ntp 服务

```
sudo systemctl stop ntp  
sudo systemctl disable ntp
```

3、设置时间

```
date -s "20230303 17:10:00"
hwclock -w
```

3.3.6 DOCKER IPTABLES KVM K8S AUDIT

功能	命令/信息
查看主板所需内核配置是否已开启	<code>zcat /proc/config.gz grep < 内核配置项 ></code>
查看主板所有内核配置项	<code>zcat /proc/config.gz</code>
桥接相关内核配置项	<code>zcat /proc/config.gz \ grep CONFIG_BRIDGE</code>
AUDIT 相关内核配置项	<code>CONFIG_AUDIT=y</code> <code>CONFIG_AUDITSYSCALL=y</code> <code># CONFIG_NETFILTER_XT_TARGET_AUDIT is not set</code> <code>CONFIG_AUDIT_GENERIC=y</code> <code>CONFIG_AUDIT_COMPAT_GENERIC=y</code>
K8S	<code>redmine-9626 redmine-9087 (内部链接, 咨询 FAE)</code>
IPTABLES 及网络转发	<code>CONFIG_NF_NAT_MASQUERADE=y</code> <code>CONFIG_NETFILTER_XT_TARGET_MASQUERADE=y</code> <code>CONFIG_IP_NF_TARGET_MASQUERADE=y</code> <code>CONFIG_NF_CONNTRACK=y</code> <code>CONFIG_NF_NAT=y</code> <code>CONFIG_IP_NF_NAT=y</code>

DOCKER 参考 <https://redmine.lztek.cn/issues/11057> (内部链接, 咨询 FAE)

```
# DOCKER 验证
简易验证:
root@teamhd:/# apt update -y
root@teamhd:/# apt install -y docker.io
root@teamhd:/# docker ps
CONTAINER ID   IMAGE      COMMAND   CREATED     STATUS      PORTS      NAMES
root@teamhd:/# docker --version
Docker version 24.0.7, build 24.0.7-0ubuntu2~20.04.1
```

DOCKER 官方不再支持国内直接拉取镜像, `docker run hello-world` 执行失败是正常现象。
改用 `docker ps` 和 `docker --version` 验证 (无报错即可)。
客户实际使用时, 可参考网络搜索设置 (可用的) 国内源, 或代理。

注: ubuntu (如:22.04) 和 debian 系统如果刷完 DOCKER 补丁仍无法使用 DOCKER(执行如下命令):

```
sudo update-alternatives --set iptables /usr/sbin/iptables-legacy
sudo update-alternatives --set ip6tables /usr/sbin/ip6tables-legacy
sudo systemctl restart docker
```

IPTABLES 参考 <https://redmine.lztek.cn/issues/9066> (内部链接, 咨询 FAE)

```
# IPTABLES 验证, iptables -L 执行成功代表支持
iptables -L
```

网络转发测试

按照如下的测试环境:

- 主板左侧网口 `eth0` 配置为静态 ip(无需接网线), 用于配合创建 `tcp` 服务;
- 使用 `python` 创建一个 `192.168.0.100:80` `tcp`
- 右侧网口 `enP2p3s0` 连接有线网络 (外网 `192.168.1.156`)
- 电脑通过 `putty` 连接 `192.168.1.156:8080` 能够连接成功, 并发数据给 `tcp` 服务

测试命令：

```
sudo ip addr add 192.168.0.100/24 dev eth0

sudo iptables -t nat -A PREROUTING -p tcp --dport 8080 -j DNAT --to-destination 192.168.0.100:80

sudo iptables -t nat -A POSTROUTING -p tcp -d 192.168.0.100 --dport 80 -j SNAT \
--to-source 192.168.1.156

sudo iptables -t nat -A OUTPUT -p tcp -d 192.168.1.156 --dport 8080 -j DNAT \
--to-destination 192.168.0.100:80
```

KVM 参考 <https://redmine.lztek.cn/issues/10045> (内部链接, 咨询 FAE)

K8S 参考 <https://redmine.lztek.cn/issues/9626> (内部链接, 咨询 FAE)

AUDIT 参考 <https://redmine.lztek.cn/issues/9325> (内部链接, 咨询 FAE)

相关链接: https://wiki.t-firefly.com/zh_CN/Firefly-Linux-Guide/first_use.html#docker-zhi-chi

3.3.7 以太网、WiFi、热点、网络转发

参考链接: https://wiki.t-firefly.com/zh_CN/Firefly-Linux-Guide/first_use.html#wang-luo-pei-zhi

3.3.8 4G 模块上网

用串口发 AT 指令操作模块

发 AT 命令：

```
cat /dev/ttyUSB2 &
echo -e "ATI\r\n" > /dev/ttyUSB2
```

查看模块节点: `lsusb` 拨号程序位置: `cat /etc/init.d/gobinet_boot.sh` 拨号程序 `/sbin/quectel-CM` 自启动拨号位置: `cat /etc/rc.local` 文件中 `/etc/init.d/gobinet_boot.sh &`

4G 模块复位测试, 控制模块通断重启 (需确认补丁开放对应 gpio、硬件已改焊、gpio 引脚操作正确)

```
echo 30 > /sys/class/gpio/export
echo out > /sys/class/gpio/gpio30/direction
echo 0 > /sys/class/gpio/gpio30/value
sleep 3
echo 1 > /sys/class/gpio/gpio30/value # 15 秒后/dev/ttyUSB* 设备节点出现
```

注意: D-3568 v8.0 是反相的, 高电平触发复位、低电平正常工作。V8 增加了一个三极管 3904, 若焊接复位三极管就需要补丁开放复位 io 并在开机后配置该 IO 输出低电平, 4g 模块才能正常工作

断网重连 `/etc/init.d/gobinet_boot.sh` 修改成如下代码, 可初始启动、循环检测与复位重连 `/dev/ttyUSB2` 或 `/dev/ttyUSB1` 取决于实际模块

```
#!/bin/bash
sleep 60
/sbin/quectel-CM &

check_reboot(){
    sleep 60
    DEVICE="/dev/ttyUSB2"
    while true; do
        if [ -e "$DEVICE" ];then
            if ! ifconfig usb0 | grep -q "inet "; then
                pid=$(pidof /sbin/quectel-CM)

                if [ -n "$pid" ]; then
                    echo $pid
```

```

        kill -9 $pid
        sleep 15
    fi

    echo -e "AT+CFUN=1,1\r\n" > $DEVICE

    sleep 30
    /sbin/quectel-CM &
    sleep 60
fi
fi
sleep 5
done
}

check_reboot

```

参考链接: <https://redmine.lztek.cn/issues/8276> (内部链接, 咨询 FAE)

参考链接: <https://redmine.lztek.cn/issues/8846> (内部链接, 咨询 FAE)

3.3.9 旋转屏幕方向、触摸方向

Ubuntu20.04 参考链接: <https://redmine.lztek.cn/issues/11039>

- 屏幕旋转: 创建/编辑/etc/X11/Xsession.d/55gnome-session_gnomerc 文件屏幕旋转指定角度。

```

旋转 90   xrandr -o left
旋转 180   xrandr -o inverted
旋转 270   xrandr -o right
旋转 0     xrandr -o normal

```

- 触摸旋转: 创建/编辑/etc/X11/xorg.conf.d/05-touchscreen.conf 文件指定触摸旋转角度。

```

正常情况下是"1 0 0 0 1 0 0 0 1"
向左旋转 90 度则应该设置为"0 -1 1 1 0 0 0 0 1"
向右旋转 90 度则应该设置为"0 1 0 -1 0 1 0 0 1"
上下颠倒(旋转 180 度)则应该设置为"-1 0 1 0 -1 1 0 0 1"

```

Ubuntu18.04 参考链接: <https://redmine.lztek.cn/issues/9755>

- 屏幕旋转: 创建/编辑/etc/X11/Xsession.d/55gnome-session_gnomerc 文件屏幕旋转指定角度。

```

旋转 90   xrandr -o left
旋转 180   xrandr -o inverted
旋转 270   xrandr -o right
旋转 0     xrandr -o normal

```

- 触摸旋转: 编辑/etc/X11/xorg.conf 文件末尾对应的语句

```

旋转 90
    Option "SwapAxes" "true"
    Option "InvertY" "false"
    Option "InvertX" "true"
旋转 180
    Option "SwapAxes" "false"
    Option "InvertY" "true"
    Option "InvertX" "true"
旋转 270
    Option "SwapAxes" "true"
    Option "InvertY" "true"

```

```
    Option "InvertX" "false"
旋转 0
    Option "SwapAxes" "false"
    Option "InvertY" "false"
    Option "InvertX" "false"
```

Debian 参考链接: <https://redmine.lztek.cn/issues/10942>

- 触摸旋转: 创建/编辑/etc/X11/xorg.conf.d/05-touchscreen.conf 文件指定触摸旋转角度。

正常情况下是"1 0 0 0 1 0 0 0 1"
向左旋转 90 度则应该设置为"0 -1 1 1 0 0 0 0 1"
向右旋转 90 度则应该设置为"0 1 0 -1 0 1 0 0 1"
上下颠倒(旋转 180 度)则应该设置为"-1 0 1 0 -1 1 0 0 1"

- 屏幕旋转: 创建 /usr/local/bin/rotate_screen.sh 和 /etc/xdg/autostart/rotate_screen.desktop

LinuxQT 参考链接: <https://redmine.lztek.cn/issues/9567>

- 参考 https://blog.csdn.net/qq_37858386/article/details/132356738, 解决/etc/xdg/weston/weston.ini 屏幕旋转问题, 新版本 transform 的参数要加"rotate-"
 - 修改 weston.ini 文件末尾 transform 旋转屏幕与触摸 默认参数 rotate-90 可选参数 # normal|rotate-90|rotate-180|rotate-270|flipped|flipped-rotate-90|flipped-rotate-180|flipped-rotate-270
 - 注意新旧 weston 版本 transform 参数的差异

3.3.10 GPU、DRM、NPU

GPU 负责处理图形渲染任务, 提供高效的图形计算能力。它能加速图形相关的计算, 支持 2D 和 3D 图形的绘制, 适用于游戏、图形界面和视频播放等场景。

DRM (Direct Rendering Manager) 是 Linux 内核中的一个子系统, 主要负责对 GPU 资源的管理和调度。它提供了一组接口, 允许用户空间的图形应用程序直接与 GPU 进行交互, 从而实现高效的图形渲染。

NPU 专为深度学习和人工智能计算优化, 能够高效处理神经网络的特定操作, 如矩阵乘法和卷积等。主要用于边缘设备和嵌入式系统中的 AI 推理, 例如智能手机、摄像头和物联网设备。

glmark2 (GPU 跑分工具) `glmark2-es2`

查看 GPU 频率 `cat /sys/class/devfreq/fb000000.gpu/cur_freq`

查看 NPU 频率 `sudo cat /sys/kernel/debug/clk/clk_summary | grep clk_npu_dsu0`

运行系统内置的 `clinfo` 获取平台 OpenCL 相关参数

webGL 支持在 Chromium 浏览器中, 在地址栏输入: `chrome://gpu` 可以查看 chromium 下硬件加速的支持。

RK DRM 相关文档 Rockchip_Developer_Guide_DRM_Display_Driver_CN.pdf

DRM 相关文档 https://blog.csdn.net/hexiaolong2009/category_9281458.html

DRM 相关文档 <https://blog.csdn.net/dearsq/article/details/78394388>

DRM 相关文档 <https://events.static.linuxfound.org/sites/events/files/slides/brezillon-drm-kms.pdf>

DRM 相关文档 <https://blog.csdn.net/wending1986/article/details/106837597/>

RK NPU 相关文档 Rockchip_Quick_Start_RKNN_SDK_V1.3.0_CN.pdf

RK NPU 相关文档 Rockchip_RKNPU_User_Guide_RKNN_API_V1.3.0_CN.pdf

RK NPU 相关文档 Rockchip_RV1106_Quick_Start_RKNN_SDK_V1.3.0b0_CN

CPU 设置高性能模式: `echo performance | sudo tee /sys/devices/system/cpu/cpu*/cpufreq/scaling_governor`

设置 GPU 最大频率: `echo 1000000000 | sudo tee /sys/class/devfreq/fb000000.gpu/max_freq`

查看 GPU 占用率: `while true; do cat /sys/class/devfreq/fb000000.gpu/load; sleep 1; done`

```
连续运行 glmark2-es2 --run-forever 指定显示设备 DISPLAY=:0 glmark2-es2 指定分辨率(默认 800x600)
glmark2-es2 -s 1920x1080 关闭图形输出 glmark2-es2 --off-screen
```

GPU 驱动名称 libmali-valhall-g610-g13p0-x11-gbm_1.9-1_arm64.deb

RK3588 主板版本信息:

L10 系统: librknrt.so 1.4.0 rknn_server 1.4.0 rknpu driver 0.8.2

L12 系统: librknrt.so 1.5.0 rknn_server 1.5.0 rknpu driver 0.8.8

L14 系统: librknrt.so 1.5.2 rknn_server 1.5.2 rknpu driver 0.9.3

查询 NPU 驱动版本: cat /sys/kernel/debug/rknpu/version

查看 rknn 版本:

```
strings /usr/lib/librknrt.so | grep version
```

```
strings /usr/bin/rknn_server | grep build
```

rknn-toolkit2 最新版本 1.6.0 链接 <https://github.com/rockchip-linux/rknn-toolkit2>

rknpu2 最新版本 1.5.2 链接 <https://github.com/rockchip-linux/rknpu2/tree/v1.5.2>

yolov8 链接 https://github.com/airockchip/ultralytics_yolov8

使用 yolov5 测试正常。

GCC 交叉编译工具链使用的 SDK 里面的 10.3.1

也可参考 01_Rockchip_RKNPU_Quick_Start_RKNN_SDK_V1.6.0_CN.pdf 获取。

查看 rga 版本

```
root@teamhd:/usr/lib/aarch64-linux-gnu# strings librga.so |grep rga_api |grep version
rga_api version 1.9.3_[2]
```

GPU 参考链接: <https://redmine.lztek.cn/issues/10465> (内部链接, 咨询 FAE)

NPU 参考链接:

<https://redmine.lztek.cn/issues/11058> (内部链接, 咨询 FAE)

https://wiki.t-firefly.com/Core-3568J/usage_npu.html

https://wiki.t-firefly.com/zh_CN/iCore-3588Q/usage_npu.html

3.3.11 根系统备份

根系统备份功能暂不做支持, 具体情况需和 FAE 沟通确认。

3.3.12 自启动

实测过的自启动有 4 种方式:

- /etc/init.d/创建所需脚本并在/etc/rcS.d 目录下创建符号链接指向.sh 文件。
- /etc/systemd/system/目录下创建 xx.service 服务, ExecStart=.sh 路径。并执行 systemctl enable xx.service 使能该服务后重启主板测试(也可直接重启测试)。
- /etc/xdg/autostart/目录下创建.desktop 桌面文件, Exec=.sh 路径。.sh 文件一般放在/usr/local/bin/目录下即可。
- /etc/rc.local 文件末尾 exit 0 语句之前添加
若采用/etc/rc.local 自启动, 可直接运行 bash /etc/rc.local 测试。注意.sh 文件权限, 执行 bash /etc/rc.local 时是 root 用户还是普通用户。首行的'-e' 选项可去掉, 忽略可能的加载错误, 保证所有语句均执行。若客户的.sh 包含例如 Qt 的二进制可执行程序, 使用方式 3 创建桌面文件最合适。

3.3.13 视频编解码

参考链接: <https://redmine.lztek.cn/issues/10247> (内部链接, 咨询 FAE)

参考链接: <https://redmine.lztek.cn/issues/10226> (内部链接, 咨询 FAE)

参考链接: https://wiki.t-firefly.com/zh_CN/Firefly-Linux-Guide/manual_ubuntu.html#shi-pin-ying-jian-bian-jie-ma-zhi-chi

RK 相关文档: Rockchip_Developer_Guide_MPP_CN.pdf
RK 相关文档: Rockchip_User_Guide_Linux_Gstreamer_CN.pdf

FFmpeg 对于 Rockchip 暂时只支持通过 Mpp 实现硬件解码, 暂时没有硬件编码的支持

确认 rkmpp 解码器 `ffmpeg -decoders | grep "rkmpp"`

RK 官方 MPP 库地址: <https://github.com/rockchip-linux/mpp>

安装相关依赖工具

```
sudo apt install -y git cmake
```

拉取 RK 官方 MPP 仓库

```
git clone https://github.com/rockchip-linux/mpp.git
```

编译

进入 aarch64 相应的编译路径

```
cd mpp/build/linux/aarch64/
```

修改交叉编译配置文件, 指定编译器 gcc 和 g++ (一般默认就好)

```
vim arm.linux.cross.cmake
```

运行 bash 脚本后编译 (编译过程大概需要 15 分钟)

```
./make-Makefiles.bash
```

```
make
```

视频解码

解码器 demo 为 mpi_dec_test 系列程序

视频编码

编码器 demo 为 mpi_enc_test 系列程序

3.3.14 分区

可 `df -h` 查看分区大小, 默认 14G 分区。替换系统包 rockdev/image/parameter.txt 文件可以改分区大小。(咨询 FAE)

3.3.15 修改网卡名, 以传统方式命名

```
mv /usr/lib/udev/rules.d/80-net-setup-link.rules /usr/lib/udev/rules.d/80-net-setup-link.rules.bak
```

3.3.16 设置中文环境、安装中文输入法

参考链接: <https://redmine.lztek.cn/issues/10125>

参考链接: <https://redmine.lztek.cn/issues/10597>

设置中文环境

```
echo "LANG=\"zh_CN.UTF-8\"" > /etc/default/locale
echo "LANGUAGE=\"zh_CN:zh\"" >> /etc/default/locale
chmod 664 /etc/default/locale
```

```
locale-gen zh_CN.UTF-8
```

安装中文输入法

```
sudo apt install -y fcitx
sudo apt install -y fcitx-googlepinyin # 以谷歌拼音为例
nano ~/.xprofile
```

在文件中添加以下内容, 以启用 fcitx:

```
export XMODIFIERS="@im=fcitx"
export QT_IM_MODULE="fcitx"
export GTK_IM_MODULE="fcitx"
```

```
# 重启  
sudo reboot  
  
# 配置  
fcitx-configtool  
  
# 在配置工具中添加、删除输入法、设置快捷键等
```

3.3.17 开启关闭看门狗

RK 相关文档: Rockchip_Developer_Guide_Linux_WDT_CN.pdf (咨询 FAE)
参考链接: https://wiki.t-firefly.com/zh_CN/iCore-3588Q/usage_watchdog.html

```
# 写入任意内容(大写字母‘V’除外),开启看门狗,每44秒内需要写入一次(喂狗)  
echo A > /dev/watchdog  
  
# 开启看门狗,并且内核会每隔22秒自动喂一次狗  
echo V > /dev/watchdog  
  
# 查看看门狗状态和设置超时时间  
wdctl  
  
# 开启看门狗:在/etc/rc.local文件末尾exit0之前,  
wdctl &
```

3.3.18 日志容量

注: 根据实际情况适当做限制可以防止磁盘不足引起的主板无法启动、屏幕不显示、无法进入系统停留在登录界面等问题

如下是日志相关命令,为防止主板长时间运行出现磁盘空间不足的情况,可选适当的命令添加到/etc/rc.local 自启动文件。

```
sudo du -h -m /var/log # 查看日志容量大小  
sudo shutdown -r 00:00 # 设置每天0点自动重启一次  
sudo chmod -R 777 /var/log/ # 添加权限  
sudo echo > /var/log/syslog # 清除日志  
sudo echo > /var/log/kern.log # 清除日志  
sudo journalctl --vacuum-size=100M # 限制日志文件大小
```

3.3.19 连接 VNC

第一步:

安装 tightvncserver:
sudo apt-get update
sudo apt-get install -y gnome-session-flashback tightvncserver gnome-terminal nautilus \
gnome-settings-daemon gnome-panel lrzs

第二步:

```
su root # 切换到root用户下  
sudo tightvncserver # 设置密码,选n  
vncserver -kill :1
```

第三步:

xstartup 文件(覆盖的方式)拷贝到 /root/.vnc 目录下 (若使用 Xshell 执行 rz 上传文件)
chmod 755 xstartup

第四步:

```
vncserver -geometry 1920x1080 -depth 24
```

注意事项：

最好先在 root 用户下执行，保证.vnc 文件夹安装在/root 目录下，测试通过后，再自行测试/home/teamhd 目录下放.vnc 的情况

1、如果不是在 root 用户下启动 tightvncserver 则需要设置 HOME USER 变量，
并通过 echo \$HOME;echo \$USER 查看

2、注意启动 vncserver，如果前面加了 sudo，则会在/root/.vnc/目录下找 xstartup 文件

3、电脑端使用 tightvnc 的时候，端口要和主板 vnc 启动的端口一致。

4、如果一样要在普通用户下使用，则需要设置 HOME USE 环境变量，

只有在/etc/bash.bashrc 末尾添加后重启才生效，tightvncserver 会在 HOME 设置的家目录下寻找 xstartup，若显示灰屏，大概率是家目录不对

xstartup 文件，注意权限和文件编码格式 (需 unix)

```
#!/bin/sh

unset SESSION_MANAGER
unset DBUS_SESSION_BUS_ADDRESS
export XKL_XMODMAP_DISABLE=1
export XDG_CURRENT_DESKTOP="GNOME-Flashback:GNOME"
export XDG_MENU_PREFIX="gnome-flashback-"
[ -x /etc/vnc/xstartup ] && exec /etc/vnc/xstartup
[ -r $HOME/.Xresources ] && xrdb $HOME/.Xresources
xsetroot -solid grey
vncconfig -iconic &
#gnome-terminal &
#nautilus &
gnome-session --session=gnome-flashback-metacity --disable-acceleration-check &
```

3.3.20 xrandr 设置屏幕

列出所有显示设备和可用的分辨率

```
xrandr
```

设置主显示器

```
xrandr --output <DISPLAY> --primary
```

设置显示器的分辨率

```
xrandr --output <DISPLAY> --mode <RESOLUTION>
```

旋转显示器 方向可以是 normal、left、right 或 inverted

```
xrandr --output <DISPLAY> --rotate <direction>
```

设置显示器的位置

```
xrandr --output <DISPLAY1> --auto --right-of <DISPLAY2>
```

关闭显示器

```
xrandr --output <DISPLAY> --off
```

添加自定义分辨率

首先，使用 `cvt` 生成模式线：

```
cvt 1920 1080 60
```

然后使用输出的模式行添加新模式：

```
xrandr --newmode "1920x1080_60.00" 1920 2080 2112 2200 1080 1083 1088 1120 -hsync +vsync
```

```
xrandr --addmode <DISPLAY> "1920x1080_60.00"
```

```
xrandr --output <DISPLAY> --mode "1920x1080_60.00"
```

```
# 双屏设置命令  
xrandr --output eDP-1 --auto --primary --output HDMI-1 --auto --right-of eDP-1  
  
# 双屏镜像  
xrandr --output HDMI-1 --auto --same-as eDP-1
```

3.3.21 禁用 WiFi 蓝牙

方法一：删除 ko 驱动文件

```
rm -f /system/lib/modules/*.ko
```

方法二：注释 ko 驱动文件加载命令

编辑/etc/rc.local，注释掉相关命令

若是 WiFi6 模块，对应的驱动名为 aic8800_bsp.ko 和 aic8800_fdrv.ko 注释如下语句

```
if [ -f /system/lib/modules/aic8800_bsp.ko ]; then  
    insmod /system/lib/modules/aic8800_bsp.ko  
    sleep 2  
fi  
  
if [ -f /system/lib/modules/aic8800_fdrv.ko ]; then  
    insmod /system/lib/modules/aic8800_fdrv.ko  
    sleep 2  
fi
```

若是 ap6212 模块，对应的驱动名为 bcmdhd.ko，注释如下语句

```
if [ -f /system/lib/modules/bcmdhd.ko ]; then  
    insmod /system/lib/modules/bcmdhd.ko  
    sleep 5  
fi
```

方法三：执行关闭 WiFi 的命令（如需开启，执行 sudo nmcli r wifi on）

```
sudo nmcli r wifi off
```

3.3.22 HDMI-IN 输入测试

参考链接: <https://redmine.lztek.cn/issues/10908> (内部链接, 咨询 FAE)

参考链接: <https://redmine.lztek.cn/issues/10085> (内部链接, 咨询 FAE)

参考链接: <https://redmine.lztek.cn/issues/9523> (内部链接, 咨询 FAE)

参考链接: <https://redmine.lztek.cn/issues/11246> (内部链接, 咨询 FAE)

相关命令:

1. 确认是否读取到输入源参数

```
v4l2-ctl -d /dev/video0 -V
```

2. 查看帧率范围

```
v4l2-ctl --list-frameintervals width=1920,height=1080,pixelformat=NV12 -d /dev/video0
```

3. 查看当前设置的视频格式

```
v4l2-ctl --get-fmt-video -d /dev/video0
```

4. 设置 60 帧并输出调试信息的方式启动

```
gst-launch-1.0 -v v4l2src device=/dev/video0 ! video/x-raw,format=NV12,width=1920,height=1080 \  
! videorate ! video/x-raw,framerate=60/1 ! videoconvert ! autovideosink
```

3.3.23 Debian 禁止待机

```
sudo xset -dpms  
sudo xset s off  
xset dpms force off (立即关闭屏幕)
```

3.3.24 性能测试

参考链接: <https://redmine.lztek.cn/issues/11159> (内部链接, 咨询 FAE)

参考文档: Rockchip_Developer_Guide_Linux_USB_Performance_Analysis_CN.pdf

安卓平台使用 `adb root`、回车、`adb shell` 登录, 然后 `df -h` 获取 U 盘名称, 下列命令改成实际的 U 盘命令。

使用 USB3.0 的 U 盘接在 USB3.0 接口

安卓系统下测试写 U 盘命令:

先清缓存 `echo 3 > /proc/sys/vm/drop_caches`

真实的写入速度 `busybox dd if=/dev/zero of=/mnt/media_rw/2A9B-BDD1/test1 bs=1M count=1024 conv=sync`

安卓系统下测试读 U 盘命令:

先清缓存 `echo 3 > /proc/sys/vm/drop_caches`

真实的读取速度 `busybox dd if=/mnt/media_rw/2A9B-BDD1/test1 of=/dev/null conv=sync`

ubuntu 系统下测试写 U 盘命令:

先清缓存 `echo 3 > /proc/sys/vm/drop_caches`

真实的写入速度 `dd if=/dev/zero of=/mnt/media_rw/2A9B-BDD1/test bs=128K count=8K`

ubuntu 系统下测试读 U 盘命令:

先清缓存 `echo 3 > /proc/sys/vm/drop_caches`

真实的读取速度 `dd if=/media/teamhd/2A9B-BDD1/test of=/dev/null bs=128K count=8K`

其他相关工具: `hdparm`

功能	命令
查看分区	<code>fdisk -l</code>
写磁盘	<code>dd if=/dev/zero of=/dev/mmcblk0p6 bs=1M count=2000 oflag=direct,nonblock</code>
读磁盘	<code>dd if=/dev/mmcblk0p6 of=/dev/null bs=1M count=2000 iflag=direct,nonblock</code>
U 盘写	<code>dd if=/dev/zero of=/dev/sda1 bs=1M count=2000 oflag=direct,nonblock</code>
U 盘读	<code>dd if=/dev/sda1 of=/dev/null bs=1M count=2000 iflag=direct,nonblock</code>
设置性能模式	<code>echo performance tee \$(find /sys/ -name *governor)</code>

4 交叉编译环境的建立 (PC 端)

```
# 安装基础环境  
apt update  
apt install -y qtcreator qtbase5-dev  
  
# 安装额外组件与开发环境, 例如  
apt install -y libqt5multimedia5 qtmultimedia5-dev libqt5quick5 qtdeclarative5-dev
```

QT 5.12.2 版本

- Qt 版本: 5.12.2
- 开发依赖的系统: x86-64 / Ubuntu 18.04
- 支持的主板型号: RK3568 RK3566 RK3399 RK3328 PX30 / **Ubuntu 18.04** Minimal&Desktop

QT 5.15 版本

- Qt 版本: 5.15
- 开发依赖的系统: x86-64 / Ubuntu 20.04
- 支持的主板型号: Firefly RK3588 RK3568 RK3566 / **Ubuntu 20.04** Desktop

下载: <https://pan.baidu.com/s/1luAwk9eqFtYGILFcZSN1jw> 提取码: ae7w

详情参见工具链中的 **Qt5.1x.x_Release.md** 文件 编译: 在 host 端, 进入 Qt 工程目录, `qmake && make` 即可. 运行: 工具链中含有例程, 用户在部署完成后, 可以在 host 端 build demo,
在 target 端运行 demo 以测试部署是否成功。后端: 工具链完整支持 wenEngine, 支持 EGLFS LinuxFB XCB 等后端。

确定了使用哪个后端, 就可以修改设备中 /etc/profile.d/target_qtEnv.sh 文件,
去除对应平台环境变量前面的 # 使其一直生效

例如, 使用 XCB , 则将文件内 XCB 部分前面的 # 删除

```
#XCB  
export QT_QPA_PLATFORM=XCB  
export QT_QPA_EGLFS_INTEGRATION=XCB_EGL
```

注意, qt 文档中所有路径的名称不可更改, 否则会导致编译或者运行出错。

相关链接:

Qt 介绍: https://wiki.t-firefly.com/zh_CN/Firefly-Linux-Guide/first_use.html#qt-zhi-chi

Qt 开发文档: <https://doc.embedfire.com/linux/qt/embed/zh/latest/README.html>

Qt 开发视频: <https://www.bilibili.com/video/BV1sF411L7cz/?t=2.0>

5 目标程序的部署和调试 (主板端)

参考链接: <https://www.cnblogs.com/loongson-artc-lyc/p/16149936.html>

以 qt5.12.2 在 rk3568 上部署为例: **PC** 端

```
下载 : firefly_arm64_qt5.12.2_18.04_20210414.tgz
解压到 PC 主机: tar -zxvf firefly_arm64_qt5.12.2_18.04_20210414.tgz
下载 demo: https://github.com/guodongxiaren/QCounter
参考 firefly_arm64_qt5.12.2_18.04_20210414.tgz 里面的 Qt5.12_Release.md 进行配置
cd /home/teamhd/firefly_arm64_qt5.12.2_18.04/example/
解压 QCounter-master 到该目录
进入该目录, 执行:
/opt/firefly_arm64_qt5.12.2_18.04/host/bin/qmake -project
/opt/firefly_arm64_qt5.12.2_18.04/host/bin/qmake -makefile
make
生成可执行程序, 传送到主板
```

主板端

```
下载: firefly_arm64_qt5.12.2_18.04_20210414.tgz
解压到主板
参考 firefly_arm64_qt5.12.2_18.04_20210414 里面的 Qt5.12_Release.md 进行配置
运行程序 (参考 Qt5.15_Release_CN.md ): LD_LIBRARY_PATH=/opt/firefly_qt5.12/lib ./Counter
```

6 常用调试命令

注：若命令执行未成功，尝试在命令前面增加 sudo，sudo 和命令之间要有空格。
命令因系统不同，稍有差异，以实测为准。

功能	命令
查看日志	dmesg
查看显示日志	cat /var/log/xorg.0.log
查看异常日志	journalctl -xe
查看系统信息	cat /etc/os-release
查看 IP	ip
查看 IP	ip addr show
测试网络连通性	ping www.baidu.com
GPU 跑分工具	glmark2-es2
CPU 信息	cat /proc/cpuinfo
查看磁盘空间使用情况	df -h 或 lsblk
挂载设备的路径	df -h
USB 设备信息	lsusb
列出系统中可用音频设备	aplay -l
桌面环境	echo \$XDG_CURRENT_DESKTOP
显示架构	echo \$XDG_SESSION_TYPE
显示管理器	cat /etc/X11/default-display-manager
内核版本/内核时间	uname -a 或 uname -v
gcc 版本	gcc -v
OpenCL	clinfo
RGA	strings librga.so grep rga_api grep version
查找文件 (注意：请勿在根目录下执行,请勿在/sys /system /proc 等目录下执行)	find ./ -name *.sh
查内存	cat /proc/meminfo 或 free -h

7 文档版本

当前文档版本是：(1.0)