

IDO-EVB3566-V1 Linux使用手册

IDO-EVB3566-V1 Ubuntu系统

1. 调试

1.1. 串口调试

1.2. ADB调试

1.3. ssh调试

2. 串口

2.1. 测试方法

3. USB

3.1. 电源控制

3.2. OTG切换

4. TF CARD

5. 以太网

5.1. 查看以太网IP地址

5.1.1. 使用命令查看

5.2. 设置以太网临时IP地址

5.2.1. 使用命令设置

5.3. 设置以太网永久静态IP

6. WiFi

6.1. 连接热点

6.1.1. 方式一

6.1.2. 方式二

6.1.3. 方式三

6.2 查看WiFi的IP地址

7. 蓝牙

7.1. 扫描设备

7.2. 连接蓝牙设备

8. 音频

8.1. 查看声卡设备

8.2. 播放音频

8.3. 音量的调节

8.3.1. 方式一

8.3.2. 方式二

8.4. 录音

9. 摄像头

9.1. 测试

查看摄像头信息

拍摄并播放图像

10. RTC

10.1. 设置RTC时间

11. 开机自启动

12. 屏幕控制

12.1. 背光调节

12.2. 屏幕旋转

12.2.1. 临时旋转

12.2.2. 永久旋转

13. ADC

13.1. ADC转换方法

13.2. 测试

14. 网络优先级设置

14.1. 查看路由表

14.2. 设置默认路由

14.2.1. 设置WiFi为默认路由

IDO-EVB3566-V1 Debian系统

1. 调试

1.1. 串口调试

1.2. ADB调试

1.3. ssh调试

2. 串口

2.1. 测试方法

3. USB

- 3.1. 电源控制
- 3.2. OTG切换
- 4. TF CARD
- 5. 以太网
 - 5.1. 查看以太网IP地址
 - 5.1.1. 使用命令查看
 - 5.2. 使用命令设置
 - 5.3. 设置以太网永久静态IP
- 6. WiFi
 - 6.1. 连接热点
 - 6.1.1. 方式一
 - 6.1.2. 方式二
 - 6.1.3. 方式三
- 7. 蓝牙
 - 7.1. 连接蓝牙设备
- 8. 音频
 - 8.1. 查看声卡设备
 - 8.2. 播放音频
 - 8.3. 音量的调节
 - 8.4. 录音
- 9. 摄像头
 - 9.1. 测试
 - 9.2. 查看摄像头信息
- 10. RTC
 - 10.1. 获取RTC时间
 - 10.2. 设置RTC时间
- 11. 开机自启动
- 12. 屏幕控制
 - 12.1. 背光调节
 - 12.2. 屏幕旋转
 - 12.2.1. 临时旋转
 - 12.2.2. 永久旋转

13. ADC

13.1. ADC转换方法

13.2. 测试

14. 网络优先级设置

14.1. 查看路由表

14.2. 设置默认路由

14.2.1. 设置WiFi为默认路由

14.2.2. 设置以太网为默认路由

IDO-EVB3566-V1 Buildroot系统

1. 硬件资源概况

1.1. 主板照片,如下图所示:

1.2. 硬件资源及设备节点,如下表所示:

2. 调试

2.1. 串口调试

2.2. ADB调试

2.3. ssh调试

3. UART

3.1. 测试方法

4. USB

5. TF CARD

6. 以太网

6.1. 查看以太网IP地址

6.1.1. 使用命令查看

6.2. 设置以太网临时IP地址

6.2.1. 使用命令设置

6.3. 设置以太网永久静态IP

7. WiFi

8. 连接蓝牙设备

9. 音频

9.1. 查看声卡设备

9.2. 播放音频

9.3. 录音

10. RTC

10.1. 获取RTC时间

10.2. 设置RTC时间

11. 开机自启动

12. 屏幕控制

12.1. 背光调节

13. 按键

14. 网络优先级设置

14.1. 查看路由表

14.2. 设置默认路由

14.2.1. 设置WiFi为默认路由

14.2.2. 设置以太网为默认路由

15. 测试GPIO输入输出



IDO-EVB3566-V1

Linux使用手册

文档修订历史

版本	PCBA版本	修订内容	修订	审核	日期
V1.0	V1A	创建文档	LXH	IDO	2024/07/16

IDO-EVB3566-V1 Ubuntu系统

1. 调试

1.1. 串口调试

通信参数为1500000 8 N 1, 电平状态为TTL电平,如下图所示:

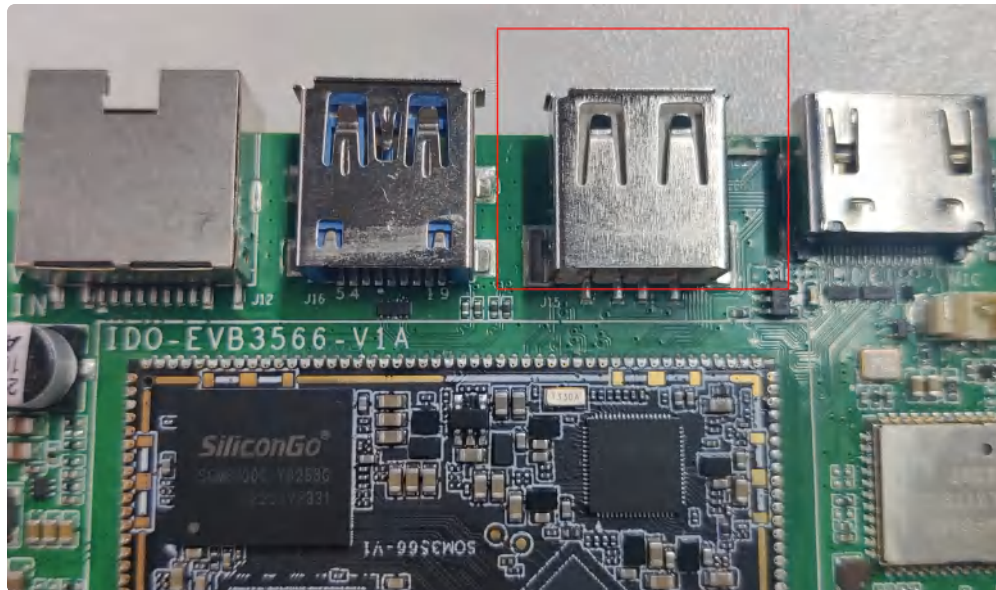


串口引脚参数，如下表所示：

序号	定义	电平/V	说明
1	UART2_RX	3.3V	调试串口信号输入
2	UART2_TX	3.3V	调试串口信号输出
3	GND	电源地	电源地

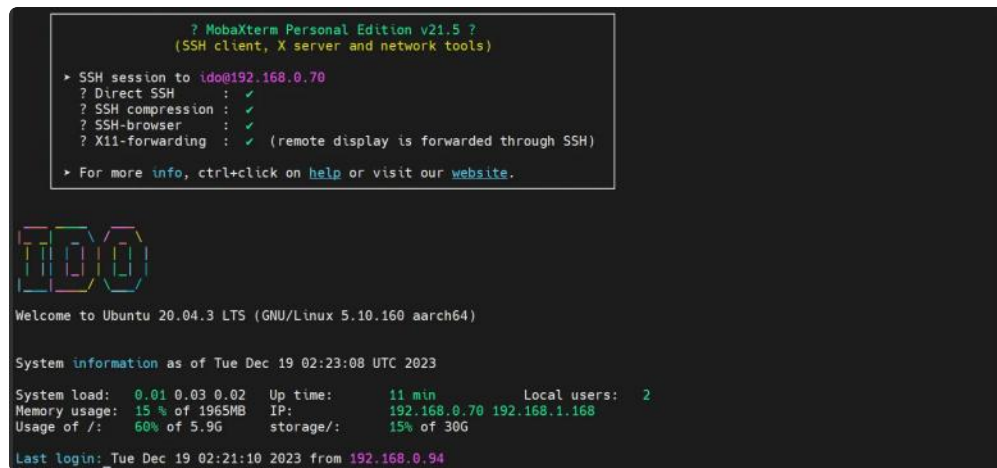
1.2. ADB调试

ADB调试端口位于j29，使用双公头USB线，连接EVB3566和电脑，即可在电脑上使用adb调试,如下图所示：



1.3. ssh调试

系统默认ssh账号和密码为 `ido @ 123456`,如下图所示:

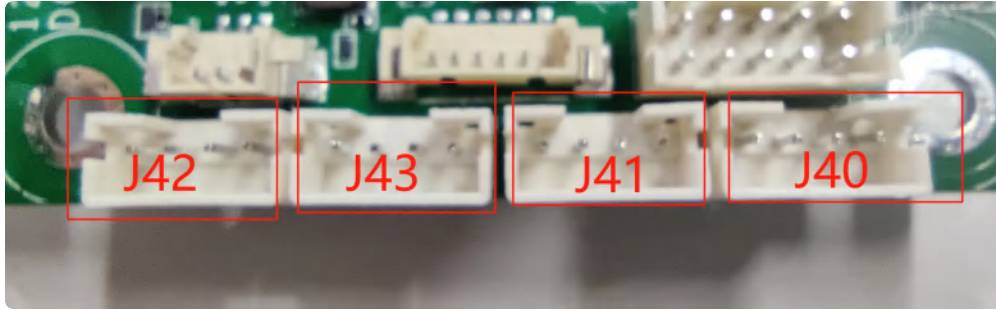


需要修改密码的话, 可以使用`passwd+用户名`设置密码

```
Bash |  
1 root@ido:~# passwd ido  
2 New password:  
3 Retype new password:  
4 passwd: password updated successfully  
5 root@ido:~#
```

2. 串口

IDO-EVB3566-V1主板扩展2路UART和2路RS232（不含调试UART），4路串口通过4个PH2.0-4P立贴座子接出（默认2路UART TTL、2路RS232），如下图所示：



引脚参数，如下表所示：

连接器（设备节点）	UART TTL	RS232
J40 (/dev/ttyS7)	✓	✓（默认功能）
J41 (/dev/ttyS9)	✓	✓（默认功能）
J42 (/dev/ttyS0)	✓（默认功能）	✓
J43 (/dev/ttyS5)	✓（默认功能）	✓

2.1. 测试方法

TTL、RS485和RS232都可以使用microcom工具进行简单的收发测试。

首先安装microcom工具：

```
▼ Bash |  
1 root@ido:/# apt-get update  
2 root@ido:/# apt-get install microcom
```

以测试/dev/ttyS0为例：

```
▼ Bash |
1 root@ido:~# microcom -s 115200 -p /dev/ttyS0
2 [ 2365.210199] of_dma_request_slave_channel: dma-names property of node '/
  serial@fdd50000' missing or empty
3 connected to /dev/ttyS0
4 Escape character: Ctrl-\
5 Type the escape character to get to the prompt.
6 111111
7 111111
8 111111
9 111111
10 111111
```

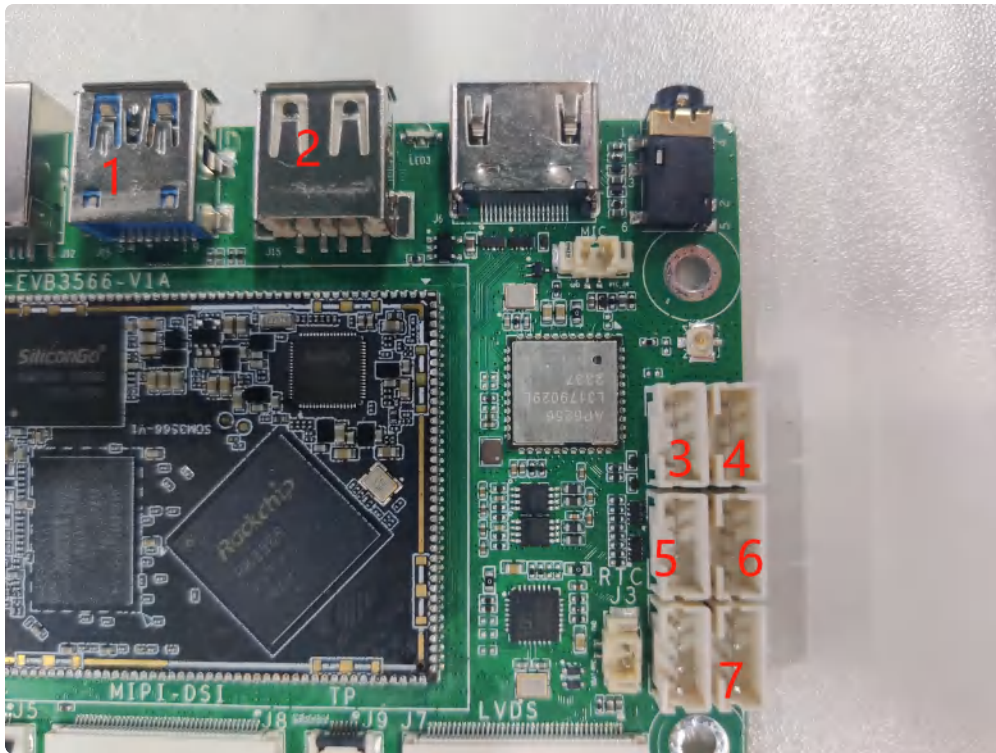
按下键盘任意键会发送对应的字符，而接收的内容会显示在终端。

按【ctrl】和【\】组合键，然后输入quit退出测试。

```
▼ Bash |
1 root@ido:/# ls
2 Desktop
3 root@ido:/# microcom -s 115200 -p /dev/ttyS0
4 [ 754.636312] of_dma_request_slave_channel: dma-names property of node '/
  serial@fdd50000' missing or empty
5 [ 754.636443] ttyS0 - failed to request DMA, use interrupt mode
6 connected to /dev/ttyS0
7 Escape character: Ctrl-\
8 Type the escape character to get to the prompt.
9
10 Enter command. Try 'help' for a list of builtin commands
11 -> quit
12 exiting
```

3. USB

USB接口，如下图所示：



功能说明，如下表所示：

序号	功能	控电节点
1	USB 3.0 HOST	/
2	USB OTG	/
3	USB 2.0 HOST	/sys/class/leds/usb2_host2_pwr/brightness
4	USB 2.0 HOST	/sys/class/leds/usb2_host_pwr/brightness
5	USB 2.0 HOST	/sys/class/leds/usb2_host3_pwr/brightness
6	USB 2.0 HOST	/sys/class/leds/usb2_host1_pwr/brightness
7	USB 2.0 HOST	/sys/class/leds/usb2_host4_pwr/brightness

3.1. 电源控制

供电控制说明，设备节点写0关闭电源，写1开启电源

命令行控制方法如下，以端口7为例

```

1  #关闭
2  echo 0 > /sys/class/leds/usb2_host4_pwr/brightness
3  #开启 (默认状态)
4  echo 1 > /sys/class/leds/usb2_host4_pwr/brightness

```

3.2. OTG切换

USB OTG 切换命令

上电状态说明，如下表所示：

上电外设连接	模式说明
上电前，使用USB Type A 数据线，连接主板和PC	上电后默认为device模式
上电前，插着U盘或者未接USB设备	上电后默认作为host模式

USB OTG 支持host 和device 模式的切换，软件切换方法如下

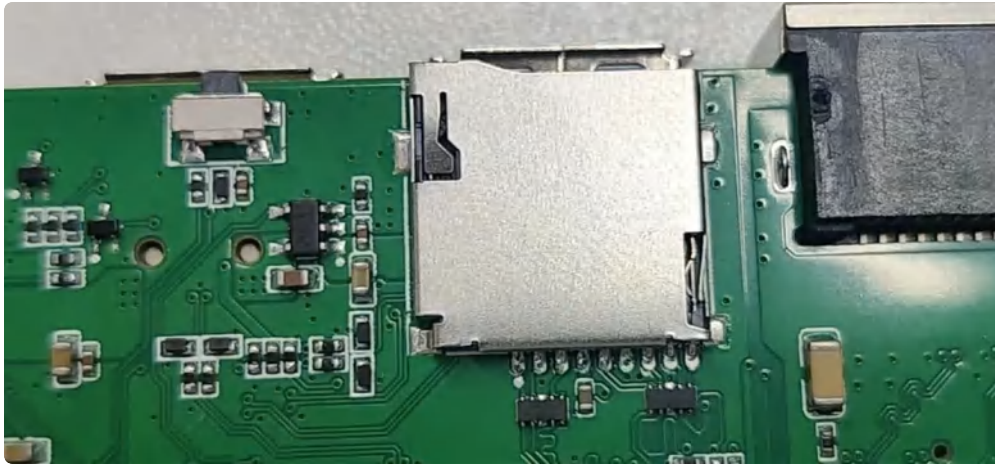
```

1  ## host
2  echo host > /sys/devices/platform/fe8a0000.usb2-phy/otg_mode
3  ## device
4  echo peripheral > /sys/devices/platform/fe8a0000.usb2-phy/otg_mode

```

4. TF CARD

TF Card支持FAT32和NTFS格式分区自动挂载，节点是/dev/mmcblk1p1，TF Card位置如下图所示：



Shell

```
1
2 root@ido:~# df -h
3 Filesystem      Size  Used Avail Use% Mounted on
4 /dev/root        5.9G  3.4G  2.2G  61% /
5 devtmpfs         973M   8.0K  973M   1% /dev
6 tmpfs            983M    0    983M   0% /dev/shm
7 tmpfs            197M   6.1M  191M   4% /run
8 tmpfs            5.0M   4.0K   5.0M   1% /run/lock
9 tmpfs            983M    0    983M   0% /sys/fs/cgroup
10 tmpfs            197M    0    197M   0% /run/user/0
11 tmpfs            197M   8.0K   197M   1% /run/user/1001
12 /dev/mmcblk0p8   23G   36K   22G    1% /media/ido/userdata
13 /dev/mmcblk0p7  121M  12M  101M  11% /media/ido/oem
14 /dev/sda1        30G   4.4G   25G  15% /media/ido/TU100
15 /dev/mmcblk1p1  500M   6.3M  494M   2% /media/ido/F2FA-AD60
16
```

5. 以太网

EVB3566配置了一个100M以太网接口，对应的网络设备节点为eth0。

5.1. 查看以太网IP地址

5.1.1. 使用命令查看

系统默认以太网为动态获取IP，当以太网接口插入网线时，会自动获取IP。

```
▼ | Bash
1
2 root@ido:~# ifconfig eth0
3 eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
4     inet 192.168.0.70 netmask 255.255.255.0 broadcast 192.168.0.255
5     inet6 fe80::e643:14e5:aab:8e74 prefixlen 64 scopeid 0x20<link>
6     ether 12:5d:b6:90:5b:30 txqueuelen 1000 (Ethernet)
7     RX packets 59 bytes 13890 (13.8 KB)
8     RX errors 0 dropped 0 overruns 0 frame 0
9     TX packets 34 bytes 4557 (4.5 KB)
10    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
11    device interrupt 48
12
```

5.2. 设置以太网临时IP地址

5.2.1. 使用命令设置

```
▼ | Bash
1 root@ido:~# ifconfig eth0 192.168.0.123
2 root@ido:~#
```

5.3. 设置以太网永久静态IP

新建/etc/netplan/00-installer-config.yaml, 然后写入如下内容 (注意缩进以Tab为单位) :

```
▼ | Bash
1 network:
2     version: 2
3     renderer: NetworkManager
4     ethernets:
5         eth0:
6             dhcp4: no
7             dhcp4: no
8             addresses: [192.168.0.10/24]
9             gateway4: 192.168.0.1
10            nameservers:
11            addresses: [8.8.8.8, 114.114.114.114]
```

然后重启网络

```
1 root@ido:~# netplan apply
```

重启网络后，eth0的ip地址已经变成刚才设置的静态ip。

```
1 root@ido:~# ifconfig
2 eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
3     inet 192.168.0.10 netmask 255.255.255.0 broadcast 192.168.0.255
4     inet6 fe80::105d:b6ff:fe90:5b30 prefixlen 64 scopeid 0x20<link>
5     ether 12:5d:b6:90:5b:30 txqueuelen 1000 (Ethernet)
6     RX packets 14890 bytes 1524686 (1.5 MB)
7     RX errors 0 dropped 246 overruns 0 frame 0
8     TX packets 123 bytes 12847 (12.8 KB)
9     TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
10    device interrupt 48
11
```

设置静态IP后，断电重启设备依旧生效。

6. WiFi

系统上电默认会打开WiFi，对应的网络设备节点为wlan0。

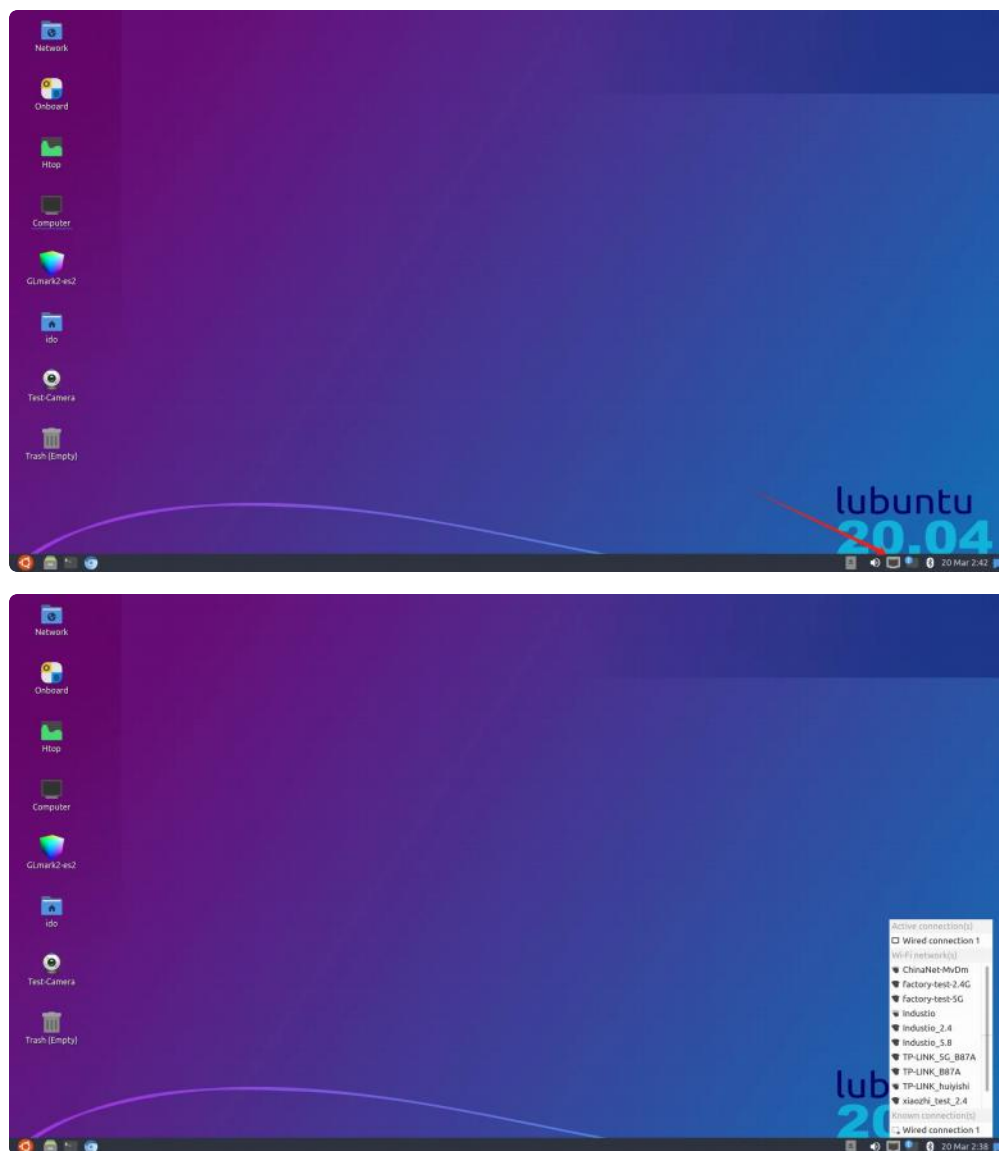
```
1 root@ido:~# ifconfig wlan0
2 wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
3     inet 192.168.1.168 netmask 255.255.255.0 broadcast 192.168.1.255
4     inet6 fe80::2ba8:9c2e:82e4:500f prefixlen 64 scopeid 0x20<link>
5     ether c0:f5:35:12:5d:64 txqueuelen 1000 (Ethernet)
6     RX packets 262 bytes 24114 (24.1 KB)
7     RX errors 0 dropped 1 overruns 0 frame 0
8     TX packets 103 bytes 13602 (13.6 KB)
9     TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
10
```

6.1. 连接热点

连接热点可以在桌面上操作，也可以使用命令行操作。

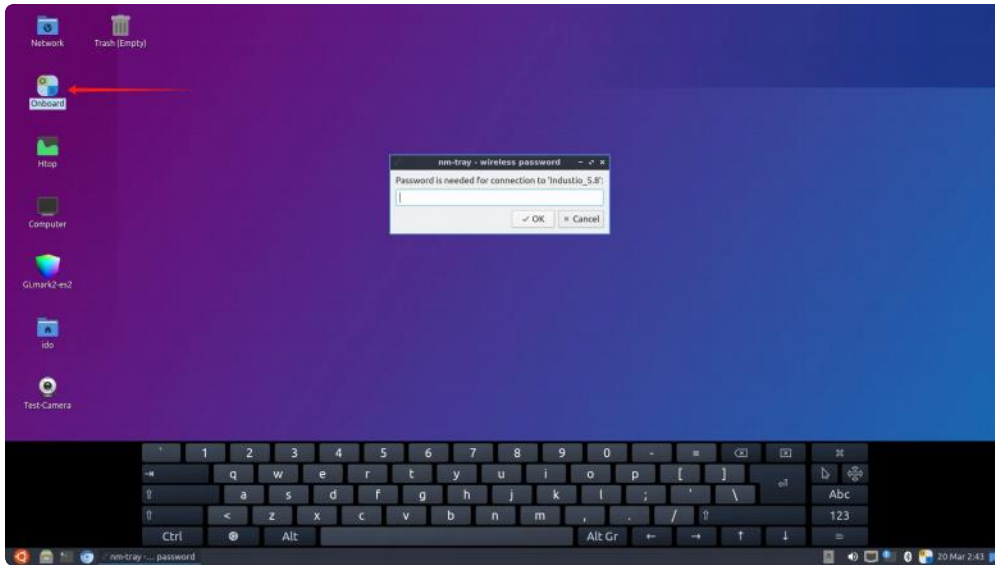
6.1.1. 方式一

鼠标左键点击桌面右下角的网络图标，即可看到WiFi热点列表，如下图所示：

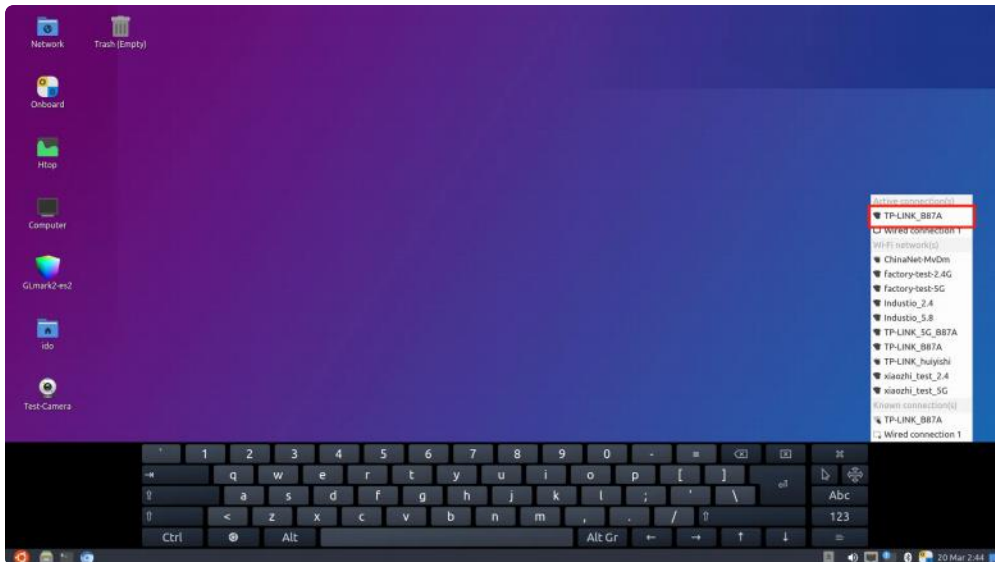


如果有连接键盘，直接输入密码即可；如果没有连接键盘，可以使用系统自带的软键盘。

点击桌面的Onboard图标，即可打开系统自带的软键盘，点击要连接的热点，弹出密码输入窗口，如下图所示：



连接成功后，点击网络图标进去里面会出现连接WiFi的名称，如下图所示：



6.1.2. 方式二

新建/etc/netplan/01-wifi-sta.yaml，并按照以下格式填写内容：

```
Bash |
1 network:
2   wifis:
3     wlan0:
4       dhcp4: true
5       access-points:
6         "TP-LINK_B87A":
7           password: "12345678"
8   version: 2
```

其中TP-LINK_B87A为要连接的热点名称，12345678为连接密码。

修改成功后执行以下命令进行连接：

```
▼ Bash |
1 root@ido:~# killall wpa_supplicant
2 root@ido:~# netplan apply
```

等待几秒钟后，将成功连接WiFi热点。

6.1.3. 方式三

直接使用nmcli工具进行测试

```
▼ Bash |
1 sudo nmcli d wifi connect TP-LINK_B87A password 12345678
```

6.2 查看WiFi的IP地址

使用ifconfig命令可查看连接热点后获取的IP

```
▼ Bash |
1 root@ido:~# ifconfig wlan0
2 wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
3     inet 192.168.1.165 netmask 255.255.255.0 broadcast 192.168.1.255
4     inet6 fe80::984a:9a2f:77b4:e899 prefixlen 64 scopeid 0x20<link>
5     ether 2c:d2:6b:10:ea:4d txqueuelen 1000 (Ethernet)
6     RX packets 83 bytes 10479 (10.4 KB)
7     RX errors 0 dropped 0 overruns 0 frame 0
8     TX packets 35 bytes 5285 (5.2 KB)
9     TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
10
11 root@ido:~#
12
```

测试wifi的网络

```
▼ Bash |
1 root@ido:~# ping www.baidu.com -I wlan0
2 PING www.a.shifen.com (14.215.177.39) from 192.168.1.117 p2p0: 56(84) byte
  s of data.
3 64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=1 ttl=54 time=17.8 ms
4 64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=2 ttl=54 time=9.30 ms
5 64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=3 ttl=54 time=25.7 ms
6 64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=4 ttl=54 time=42.1 ms
7 64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=5 ttl=54 time=13.1 ms
8 64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=6 ttl=54 time=39.8 ms
9 64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=7 ttl=54 time=28.1 ms
```

7. 蓝牙

系统开机默认打开蓝牙，对应的网络节点为hci0。

```
▼ Bash |
1 root@ido:~# hciconfig
2 hci0:  Type: Primary  Bus: UART
3         BD Address: 70:D5:2B:5B:22:22  ACL MTU: 1021:8  SCO MTU: 255:12
4         UP RUNNING
5         RX bytes:1665 acl:0 sco:0 events:57 errors:0
6         TX bytes:6311 acl:0 sco:0 commands:57 errors:0
7
8 root@ido:~#
```

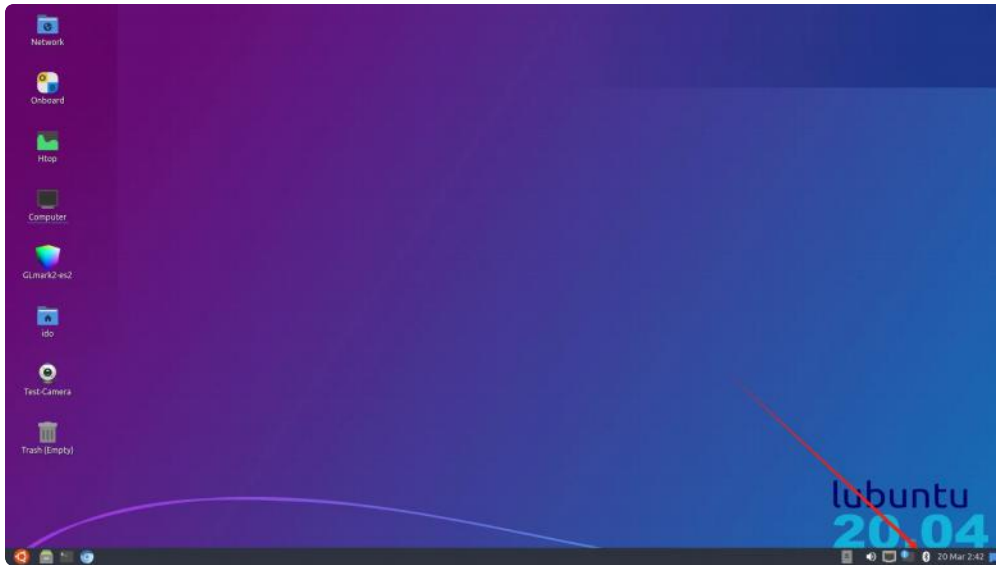
7.1. 扫描设备

使用hcitool scan或hcitool lescan对附近的设备进行扫描：

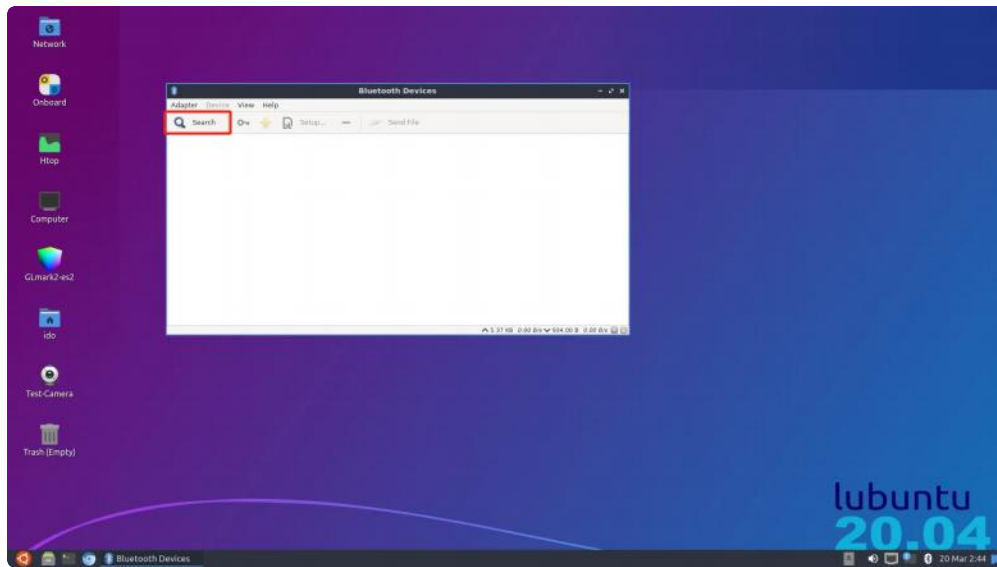
```
▼ Bash |
1 root@ido:~# hcitool scan
2 Scanning ...
3      94:87:E0:9D:14:12      seeyou
4 root@ido:~# hcitool lscan
5 LE Scan ...
6 48:AA:99:9F:BE:D8 (unknown)
7 24:CF:91:F8:E8:1B (unknown)
8 C2:E0:B3:FE:8A:E6 (unknown)
9 50:94:0D:43:77:86 (unknown)
10 72:6B:43:77:F4:27 (unknown)
11 5E:C3:95:C1:D2:F1 (unknown)
12 1A:1B:A6:E9:55:90 (unknown)
```

7.2. 连接蓝牙设备

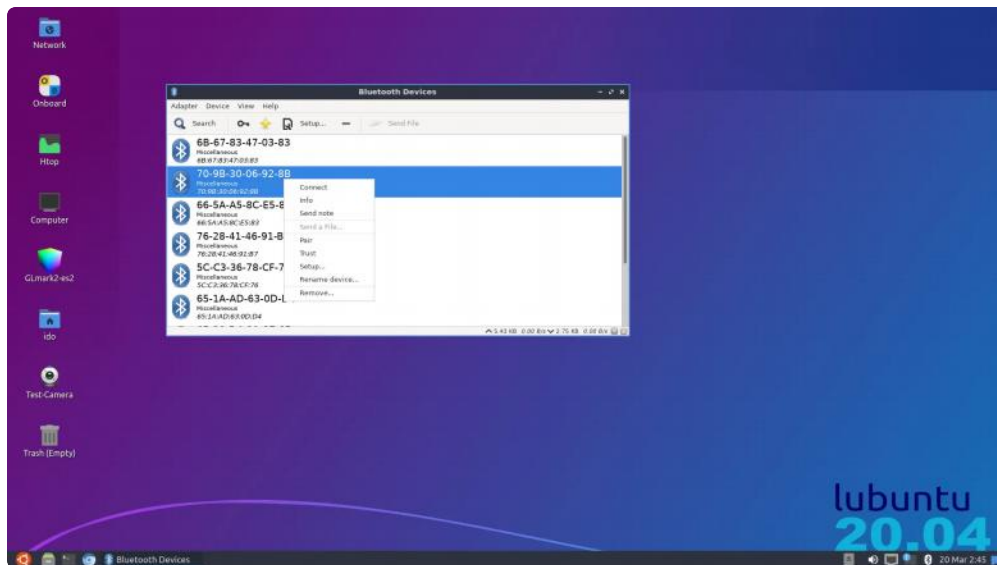
将鼠标放到桌面右下角蓝牙图标，【右键】 -> 【Devices】，如下图所示：



在弹出的窗口中，点击【Search】，如下图所示：



将看到附近的蓝牙设备列表，选中要连接设备，【右键】->【Connect】，即可连接该设备，如下图所示：



8. 音频

8.1. 查看声卡设备

```
▼ Bash |
1 root@ido:/# aplay -l
2 **** List of PLAYBACK Hardware Devices ****
3 ▼ card 0: rockchiphdmi [rockchip,hdmi], device 0: rockchip,hdmi i2s-hifi-0 [r
   ockchip,hdmi i2s-hifi-0]
4     Subdevices: 1/1
5     Subdevice #0: subdevice #0
6 ▼ card 1: rockchiprk809co [rockchip,rk809-codec], device 0: fe410000.i2s-rk81
   7-hifi rk817-hifi-0 [fe410000.i2s-rk817-hifi rk817-hifi-0]
7     Subdevices: 1/1
8     Subdevice #0: subdevice #0
```

这里第一个声卡是hdmi的，第二个声卡是喇叭的，因为我接的hdmi屏幕，如果接的其他屏幕如edp, lvds, mipi, 则只有一个声卡

```
▼ Bash |
1 root@rk3566-buildroot:/# aplay -l
2 **** List of PLAYBACK Hardware Devices ****
3 ▼ card 0: rockchiprk809co [rockchip,rk809-codec], device 0: fe410000.i2s-rk81
   7-hifi rk817-hifi-0 [fe410000.i2s-rk817-hifi rk817-hifi-0]
4     Subdevices: 1/1
5     Subdevice #0: subdevice #0
6
```

8.2. 播放音频

播放到HDMI:

```
▼ Bash |
1 aplay -D plug:spk_c0 /usr/share/sounds/alsa/Rear_Center.wav
```

播放到Lineout:

```
▼ Bash |
1 aplay -D plug:spk_c1 /usr/share/sounds/alsa/Rear_Center.wav
```

播放到耳机（需要插入耳机）：

```
▼ | Bash |
1 aplay -D plug:spk_c1 /usr/share/sounds/alsa/Rear_Center.wav
```

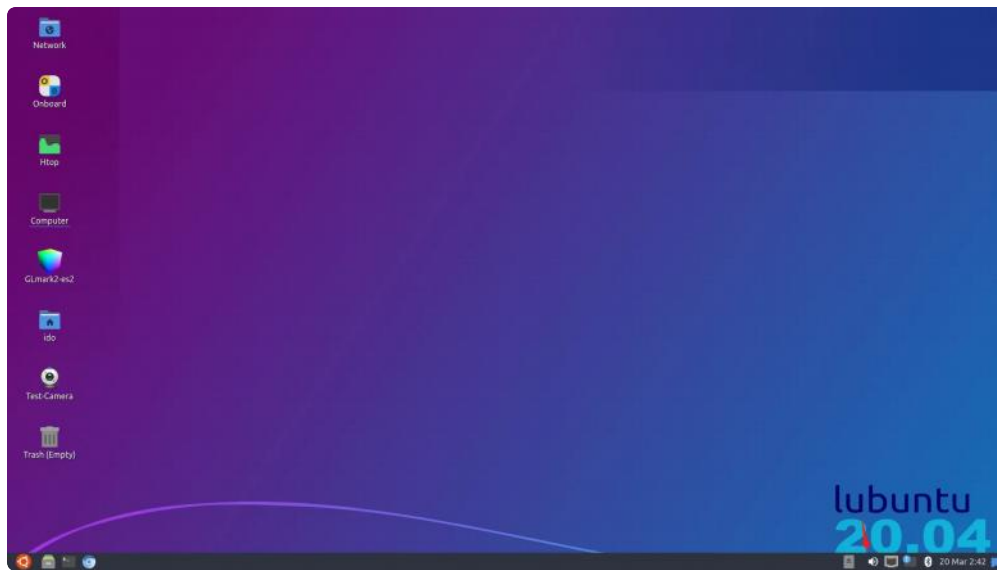
注意：这里是根据你的声卡选择，如果是接的其他屏幕，如mipi，那么只有一个声卡的情况下，喇叭选择的应该是

```
▼ | Bash |
1 aplay -D plug:spk_c0 /usr/share/sounds/alsa/Rear_Center.wav
```

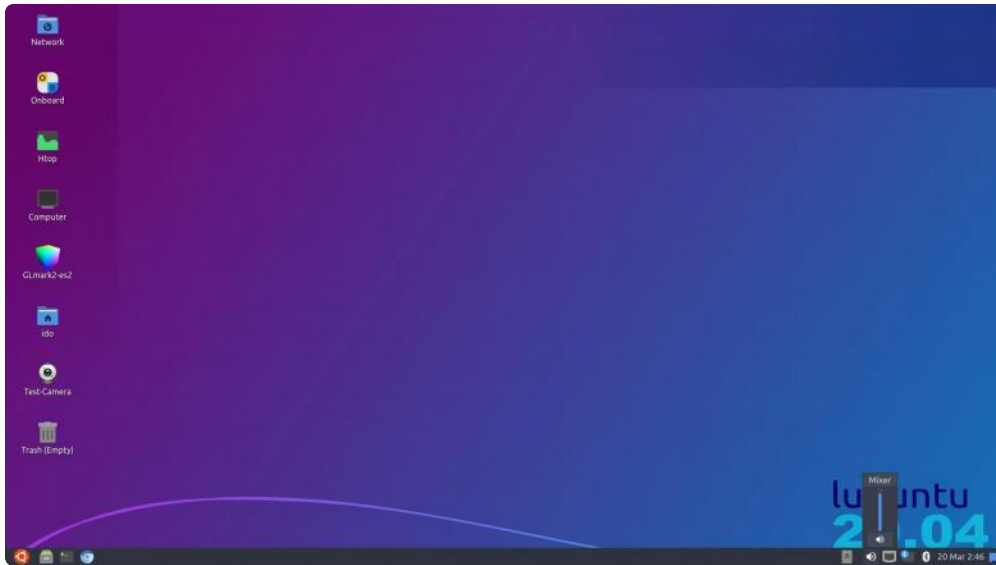
8.3. 音量的调节

8.3.1. 方式一

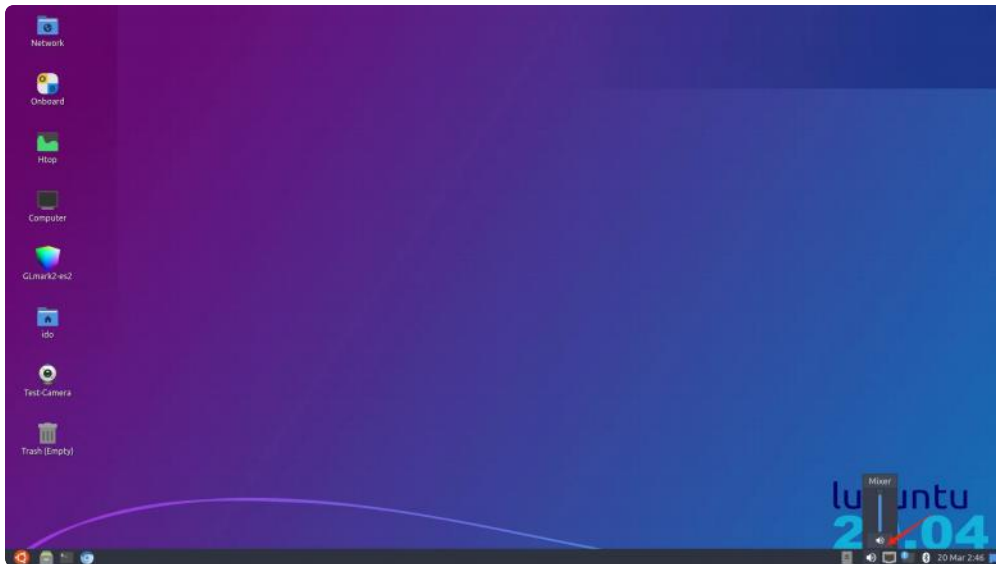
鼠标点击桌面右下角的音量图标，如下图所示：



然后滑动鼠标进行音量调节，如下图所示：



当需要静音是，点击静音按钮，如下图所示：



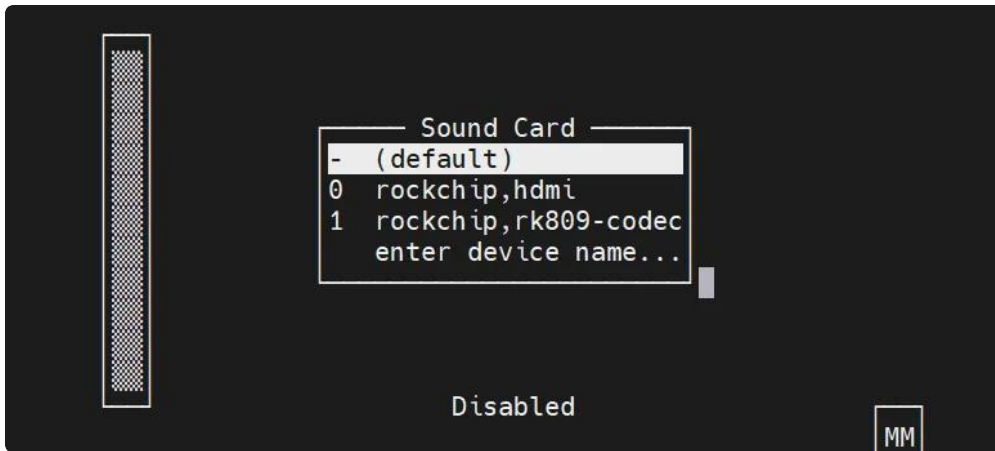
8.3.2. 方式二

执行alsamixer进入图形界面进行调试，如下图所示：

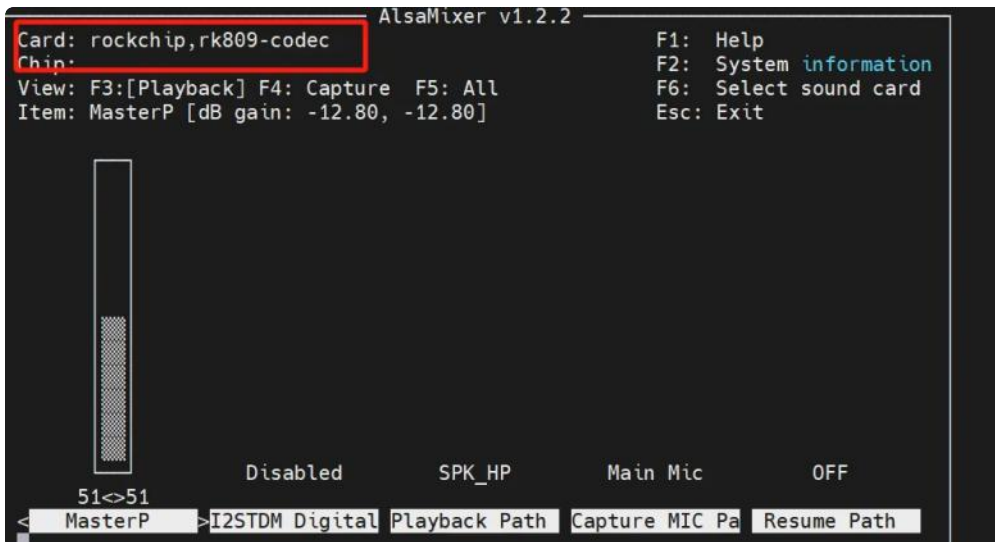
```
▼ Bash |
1 alsamixer
```




进入图形界面，按s键，选择声卡，如果是喇叭或者耳机则选择为1，如果是hdmi音频则选择为0，如下图所示：

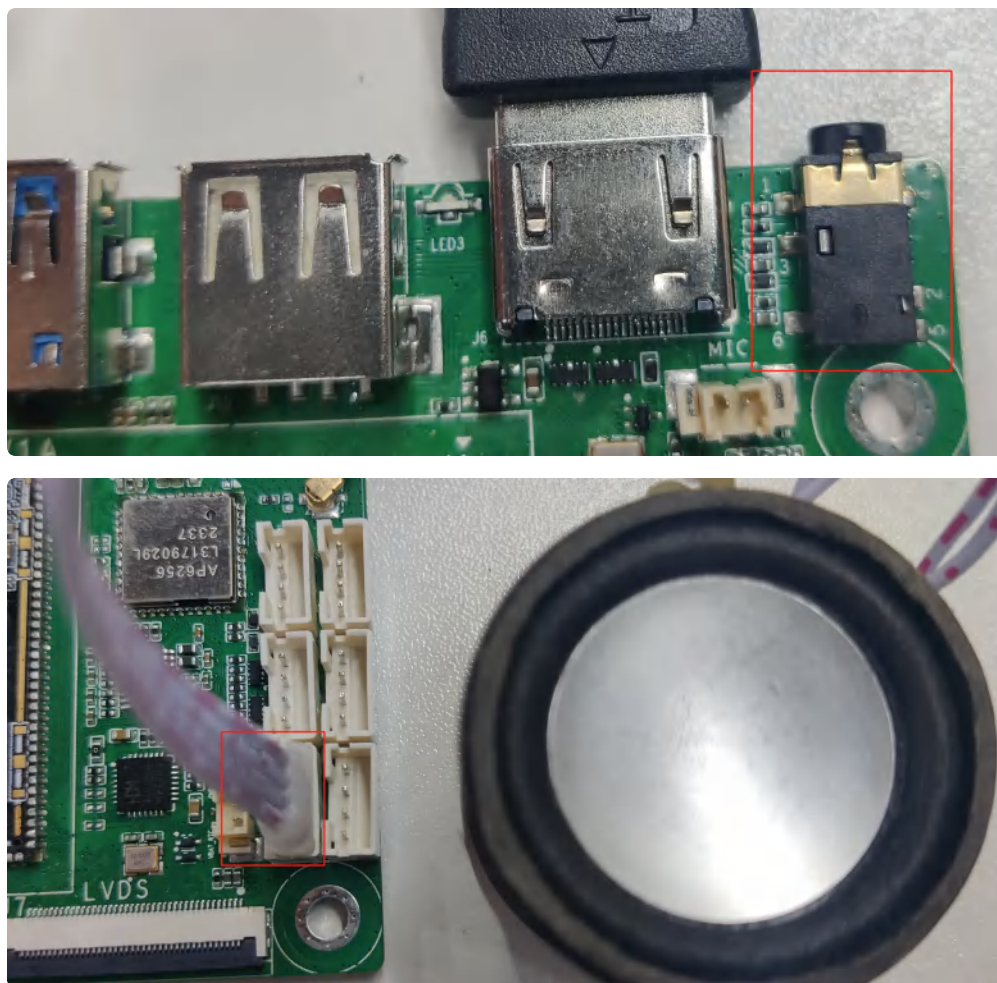


这里可以调节音量为51，如下图所示：



8.4. 录音

我们可以使用麦克风也可以使用耳机自带的麦克风进行录音，如下图所示：



EVB3566 PH2.54 4pin 喇叭接口，一路OTMP标准四节耳机座。喇叭支持最大 $8\Omega@5W$ 。

接口4pin从上到下，如下表所示：

序号	功能
1	VOLN
2	VOLP
3	VORN
4	VORP

使用arecord工具可以进行录音测试：

Bash

```
1 arecord -D hw:1,0 -r 48000 -c 2 -f S16_LE test.wav  
2
```

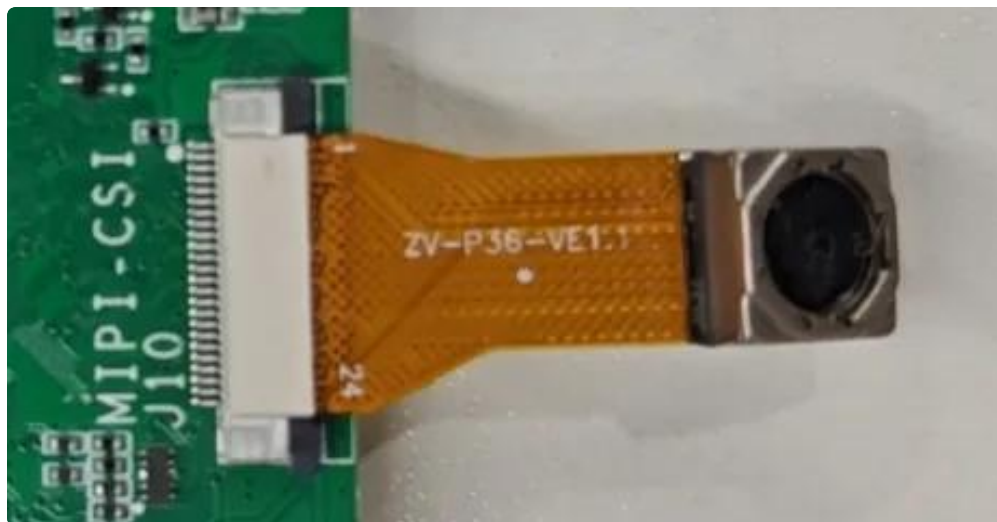
录音完后播放测试:

Bash

```
1 aplay -D plug:spk_c0 test.wav
```

9. 摄像头

EVB3566默认适配OV5648（500万像素）摄像头，如下图所示：



OV8858（800万像素）摄像头，如下图所示：



9.1. 测试

查看摄像头信息

可以看到最后的打印信息，这里使用的ov5648

```

1  root@ido:~# media-ctl -p -d /dev/media0
2
3  Media controller API version 5.10.160
4
5  Media device information
6  -----
7  driver          rkisp-vir0
8  model           rkisp0
9  serial
10 bus info
11 hw revision     0x0
12 driver version  5.10.160
13
14 Device topology
15 - entity 1: rkisp-isp-subdev (4 pads, 7 links)
16     type V4L2 subdev subtype Unknown flags 0
17     device node name /dev/v4l-subdev0
18     pad0: Sink
19         [fmt:SBGGR10_1X10/2592x1944 field:none
20         crop.bounds:(0,0)/2592x1944
21         crop:(0,0)/2592x1944]
22         <- "rkisp-csi-subdev":1 [ENABLED]
23         <- "rkisp_rawrd0_m":0 []
24         <- "rkisp_rawrd2_s":0 []
25     pad1: Sink
26         <- "rkisp-input-params":0 [ENABLED]
27     pad2: Source
28         [fmt:YUYV8_2X8/2592x1944 field:none colorspace:smpte170m q
29         uantization:full-range
30         crop.bounds:(0,0)/2592x1944
31         crop:(0,0)/2592x1944]
32         -> "rkisp_mainpath":0 [ENABLED]
33         -> "rkisp_selfpath":0 [ENABLED]
34     pad3: Source
35         -> "rkisp-statistics":0 [ENABLED]
36     .....
37
38 - entity 61: rkisp-input-params (1 pad, 1 link)
39     type Node subtype V4L flags 0
40     device node name /dev/video8
41     pad0: Source
42         -> "rkisp-isp-subdev":1 [ENABLED]
43
44 - entity 67: rockchip-csi2-dphy0 (2 pads, 2 links)

```

```

45         type V4L2 subdev subtype Unknown flags 0
46         device node name /dev/v4l-subdev2
47     pad0: Sink
48         [fmt:SBGGR10_1X10/2592x1944@10000/150000 field:none]
49         <- "m00_b_ov5648 4-0036":0 [ENABLED]
50     pad1: Source
51         -> "rkisp-csi-subdev":0 [ENABLED]
52
53 - entity 70: m00_b_ov5648 4-0036 (1 pad, 1 link)
54         type V4L2 subdev subtype Sensor flags 0
55         device node name /dev/v4l-subdev3
56     pad0: Source
57         [fmt:SBGGR10_1X10/2592x1944@10000/150000 field:none]
58         -> "rockchip-csi2-dphy0":0 [ENABLED]
59

```

拍摄并播放图像

这边查找到摄像头使用的节点是video0和video1，如下图所示：

```

root@ido:~# grep "" /sys/class/video4linux/*/name
/sys/class/video4linux/v4l-subdev0/name:rkisp-isp-subdev
/sys/class/video4linux/v4l-subdev1/name:rkisp-csi-subdev
/sys/class/video4linux/v4l-subdev2/name:rockchip-csi2-dphy0
/sys/class/video4linux/v4l-subdev3/name:m00_b_ov5648 4-0036
/sys/class/video4linux/video0/name:rkisp_mainpath
/sys/class/video4linux/video1/name:rkisp_selfpath
/sys/class/video4linux/video2/name:rkisp_rawwr0
/sys/class/video4linux/video3/name:rkisp_rawwr2
/sys/class/video4linux/video4/name:rkisp_rawwr3
/sys/class/video4linux/video5/name:rkisp_rawrd0_m
/sys/class/video4linux/video6/name:rkisp_rawrd2_s
/sys/class/video4linux/video7/name:rkisp-statistics
/sys/class/video4linux/video8/name:rkisp-input-params
root@ido:~#

```

```

▼ Bash |
1  gst-launch-1.0 v4l2src device=/dev/video0 ! video/x-raw, format=NV12, width
    =1280, height=720, framerate=30/1 ! autovideoconvert ! autovideosink

```



10. RTC

主板/dev/rtc0为外部RTC（HYM8563），系统默认使用rtc0的时间。

```
▼ Bash |
1 root@ido:~# hwclock
2 2017-08-04 09:01:11.056718+00:00
```

10.1. 设置RTC时间

```
▼ Bash |
1
2 root@ido:~# date -s '2023-12-19 16:15:20'
3 Tue Dec 19 16:15:20 UTC 2023
4 root@ido:~# hwclock -w
5 root@ido:~# hwclock -r
6 2023-12-19 16:15:43.473954+00:00
7 root@ido:~#
```

11. 开机自启动

默认系统开机会运行/etc/rc.local脚本，将要开机执行的程序放到该脚本中即可。

以设置开机运行/usr/sbin/app为例。

编辑/etc/rc.local（如果不存在则创建，且赋予可执行权限）：

```
1  #! /bin/sh
2  export PATH="${PATH:+$PATH:}/usr/sbin:/sbin"
3
4  case "$1" in
5      start)
6          /usr/sbin/app
7      ;;
8  esac
9
10 exit 0
```

这样每次开机后， /usr/sbin/app就会执行。

12. 屏幕控制

12.1. 背光调节

通过修改/sys/class/backlight/backlight/brightness的值，实现背光的调节，范围取0–255，值越大，亮度越高。

设置亮度为100：

```
1  root@ido:~# echo 100 > /sys/class/backlight/backlight/brightness
2  root@ido:~#
```

12.2. 屏幕旋转

使用xrandr工具可以实现屏幕的旋转。

12.2.1. 临时旋转

系统启动后，执行xrandr -o normal,inverted,left,right，可以实现临时旋转屏幕方向，其中normal表示顺时针旋转0度，inverted表示顺时针旋转180度，left表示顺时针旋转270度，right表示顺时针旋转

90度。

```
▼ Bash |  
1 root@ido:~# xrandr -o inverted
```

12.2.2. 永久旋转

修改/etc/default/xrandr启动文件，可以实现永久旋转。

以旋转180度为例：

```
▼ Bash |  
1 root@ido:~# cat /etc/default/xrandr  
2 #!/bin/sh  
3     /usr/bin/xrandr -o inverted  
4  
5 root@ido:~#
```

这样修改后，每次重启设备，桌面将旋转180度。

13. ADC

EVB3566配置了2路ADC，位于J46的第4，5引脚，分别记作ADC1和ADC2。精度为10位。

13.1. ADC转换方法

$$V = (raw/1024)*1.8v$$

其中raw为对应设备节点读取的值，范围为0-1023。

编号	设备节点
ADC1	/sys/bus/iio/devices/iio:device0/in_voltage3_raw
ADC2	/sys/bus/iio/devices/iio:device0/in_voltage2_raw

13.2. 测试

以测试ADC1为例，ADC2测试方法类似。

```
1 root@ido:~# cat /sys/bus/iio/devices/iio:device0/in_voltage3_raw
2 997
```

设备节点读取的raw值为997，代入到公式计算：

$$V=(997/1024)*1.8v=1.75v$$

即ADC1输入的电压为1.75v。

14. 网络优先级设置

EVB3566支持以太网、WiFi和两种网络，通过路由表来设置它们的网络优先级。

14.1. 查看路由表

```
1 root@ido:~# route
2 Kernel IP routing table
3 Destination      Gateway            Genmask           Flags Metric Ref    Use Ifa
4 default          _gateway          0.0.0.0           UG    100   0      0 eth
5 default          _gateway          0.0.0.0           UG    600   0      0 wla
6 192.168.1.0      0.0.0.0           255.255.255.0    U     100   0      0 eth
7 192.168.1.0      0.0.0.0           255.255.255.0    U     600   0      0 wla
```

14.2. 设置默认路由

14.2.1. 设置WiFi为默认路由

当前以太网和WiFi同时使用，设置WiFi优先：

```
▼ Bash |
1 root@ido:~# route
2 Kernel IP routing table
3 Destination Gateway Genmask Flags Metric Ref Use If
ace
4 default _gateway 0.0.0.0 UG 100 0 0 et
h0
5 default _gateway 0.0.0.0 UG 600 0 0 wl
an0
6 192.168.1.0 0.0.0.0 255.255.255.0 U 100 0 0 et
h0
7 192.168.1.0 0.0.0.0 255.255.255.0 U 600 0 0 wl
an0
8 root@ido:~# route del default dev eth0
9 root@ido:~# route
10 Kernel IP routing table
11 Destination Gateway Genmask Flags Metric Ref Use If
ace
12 default _gateway 0.0.0.0 UG 600 0 0 wl
an0
13 192.168.1.0 0.0.0.0 255.255.255.0 U 100 0 0 et
h0
14 192.168.1.0 0.0.0.0 255.255.255.0 U 600 0 0 wl
an0
```

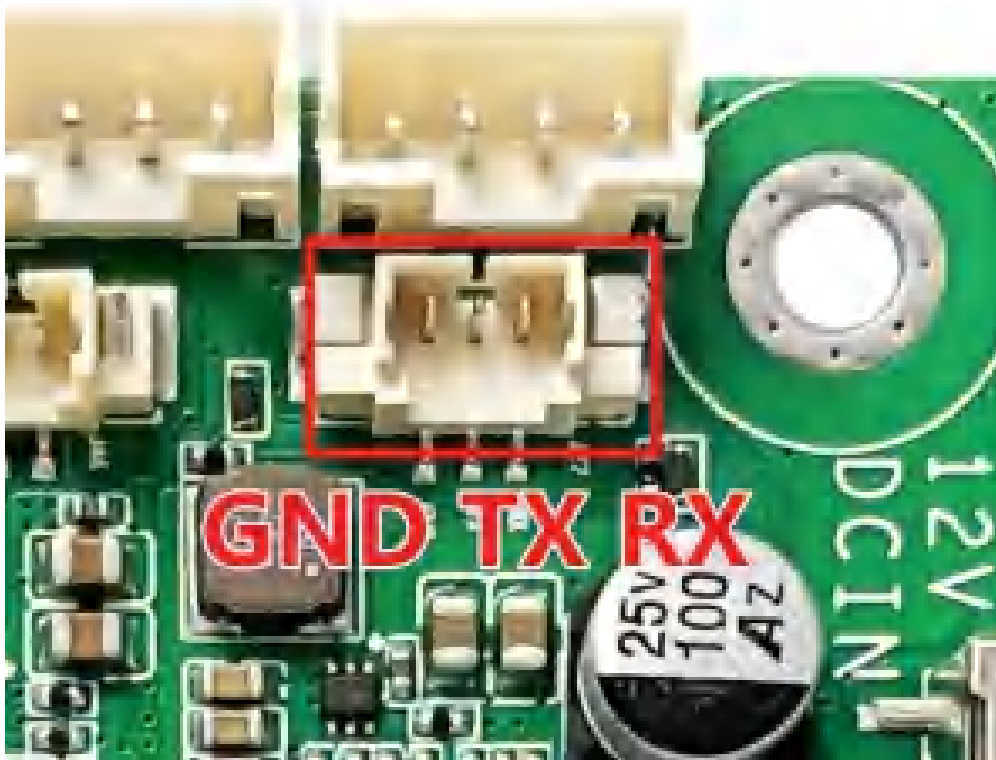
这样默认路由就是wlan0了，即优先使用WiFi进行数据通信。

IDO-EVB3566-V1 Debian系统

1. 调试

1.1. 串口调试

通信参数为1500000 8 N 1，电平状态为TTL电平,默认登录账号密码为 **linaro @ linaro**，如下图所示：

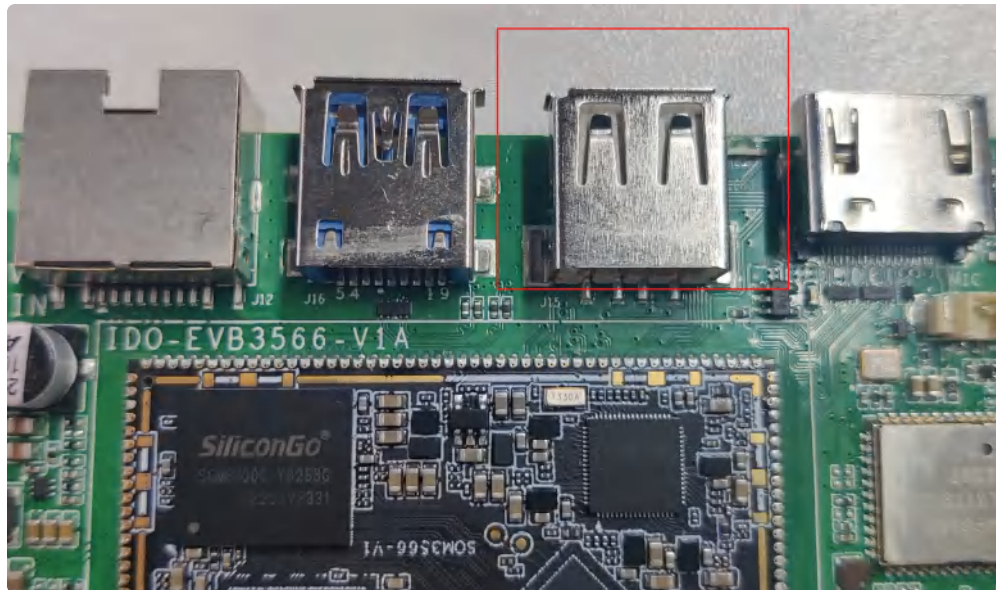


串口引脚参数，如下表所示：

序号	定义	电平/V	说明
1	UART2_RX	3.3V	调试串口信号输入
2	UART2_TX	3.3V	调试串口信号输出
3	GND	电源地	电源地

1.2. ADB调试

ADB调试端口，使用双公头USB线，连接EVB3566和电脑，即可在电脑上使用adb调试,如下图所示：



```

▼ Bash |
1 C:\Users\industio_1>adb shell
2 root@linaro-alip:/# ls
3 bin lib opt run srv userdata
4 boot lost+found out.yuv sbin sys usr
5 dev media proc sdcard system var
6 etc mnt rockchip-test sha256sum.README tets.txt vendor
7 home oem root sha256sum.txt tmp
8 root@linaro-alip:/#

```

1.3. ssh调试

默认登录账号密码为 linaro @ linaro。

```

? MobaXterm Personal Edition v21.5 ?
(SSH client, X server and network tools)

> SSH session to linaro@192.168.0.72
? Direct SSH : ✓
? SSH compression : ✓
? SSH-browser : ✓
? X11-forwarding : ✓ (remote display is forwarded through SSH)

> For more info, ctrl+click on help or visit our website.

Linux linaro-alip 5.10.160 #83 SMP Fri Dec 15 14:55:28 CST 2023 aarch64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

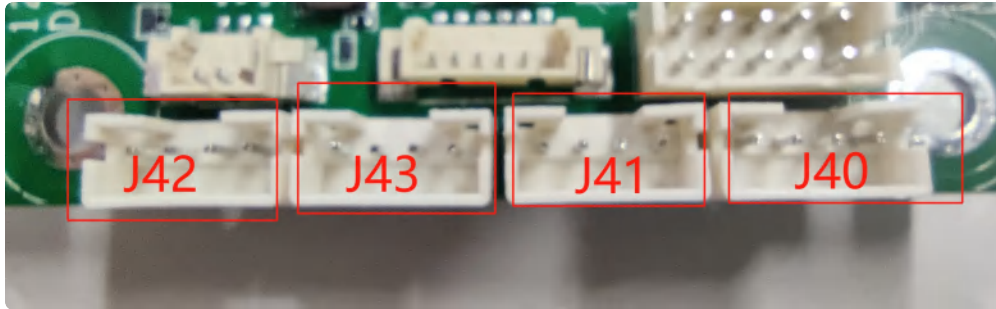
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.

linaro@linaro-alip:~$
linaro@linaro-alip:~$ █

```

2. 串口

IDO-EVB3566-V1主板扩展2路UART和2路RS232（不含调试UART），4路串口通过4个PH2.0-4P立贴座子接出（默认2路UART TTL、2路RS232），如下图所示：



引脚参数，如下表所示：

连接器（设备节点）	UART TTL	RS232
J40 (/dev/ttyS7)	✓	✓（默认功能）
J41 (/dev/ttyS9)	✓	✓（默认功能）
J42 (/dev/ttyS0)	✓（默认功能）	✓
J43 (/dev/ttyS5)	✓（默认功能）	✓

2.1. 测试方法

TTL、RS485和RS232都可以使用microcom工具进行简单的收发测试。

需要先安装microcom工具：

```
▼ Bash |  
1  linaro@linaro-alip:~$ sudo apt-get update  
2  linaro@linaro-alip:~$ sudo apt-get install microcom
```

以测试/dev/ttyS0为例：

```
▼ Bash |
1  linaro@linaro-alip:~$ microcom -s 115200 -p /dev/ttyS0
2  [ 754.636312] of_dma_request_slave_channel: dma-names property of node '/s
   erial@fdd50000' missing or empty
3  [ 754.636443] ttyS0 - failed to request DMA, use interrupt mode
4  connected to /dev/ttyS0
5  Escape character: Ctrl-\
6  Type the escape character to get to the prompt.
```

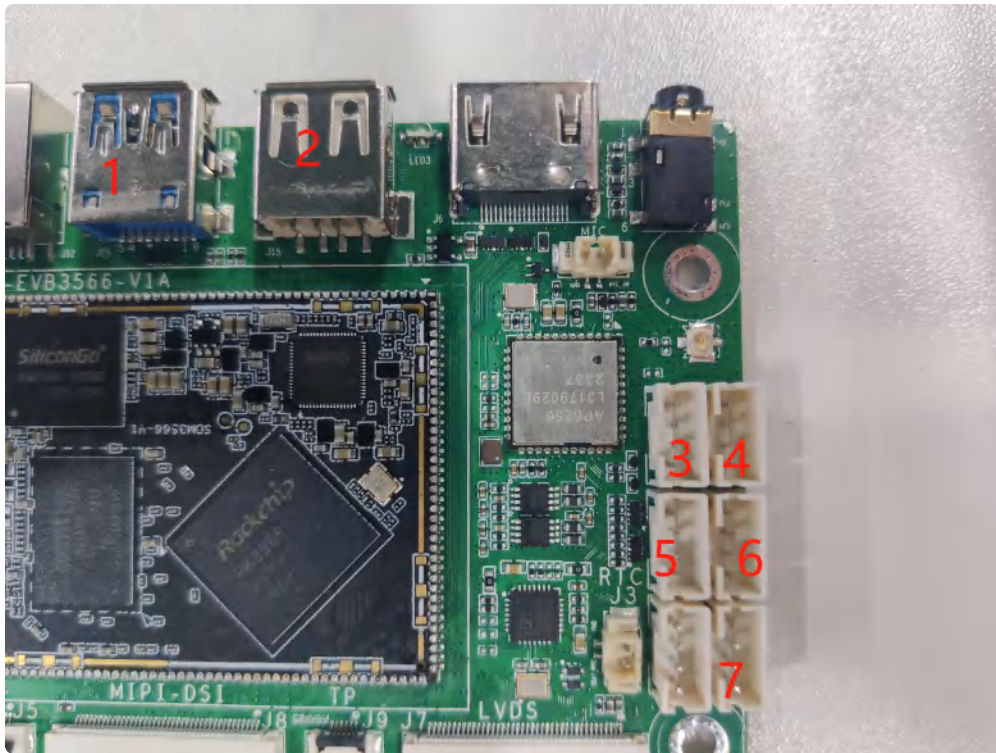
按下键盘任意键会发送对应的字符，而接收的内容会显示在终端。

按【ctrl】和【\】组合键，然后输入quit退出测试。

```
▼ Bash |
1  linaro@linaro-alip:~$ ls
2  Desktop
3  linaro@linaro-alip:~$ microcom -s 115200 -p /dev/ttyS0
4  [ 754.636312] of_dma_request_slave_channel: dma-names property of node '/
   serial@fdd50000' missing or empty
5  [ 754.636443] ttyS0 - failed to request DMA, use interrupt mode
6  connected to /dev/ttyS0
7  Escape character: Ctrl-\
8  Type the escape character to get to the prompt.
9
10 Enter command. Try 'help' for a list of builtin commands
11 -> quit
12 exiting
```

3. USB

USB接口，如下图所示：



功能说明，如下表所示：

序号	功能	控电节点
1	USB 3.0 HOST	/
2	USB OTG	/
3	USB 2.0 HOST	/sys/class/leds/usb2_host2_pwr/brightness
4	USB 2.0 HOST	/sys/class/leds/usb2_host_pwr/brightness
5	USB 2.0 HOST	/sys/class/leds/usb2_host3_pwr/brightness
6	USB 2.0 HOST	/sys/class/leds/usb2_host1_pwr/brightness
7	USB 2.0 HOST	/sys/class/leds/usb2_host4_pwr/brightness

3.1. 电源控制

供电控制说明，设备节点写0关闭电源，写1开启电源

命令行控制方法如下，以端口7为例


```

1  #关闭
2  echo 0 > /sys/class/leds/usb2_host4_pwr/brightness
3  #开启 (默认状态)
4  echo 1 > /sys/class/leds/usb2_host4_pwr/brightness

```

3.2. OTG切换

USB OTG 切换命令

上电状态说明，如下表所示：

上电外设连接	模式说明
上电前，使用USB Type A 数据线，连接主板和PC	上电后默认为device模式
上电前，插着U盘或者未接USB设备	上电后默认作为host模式

USB OTG 支持host 和device 模式的切换，软件切换方法如下

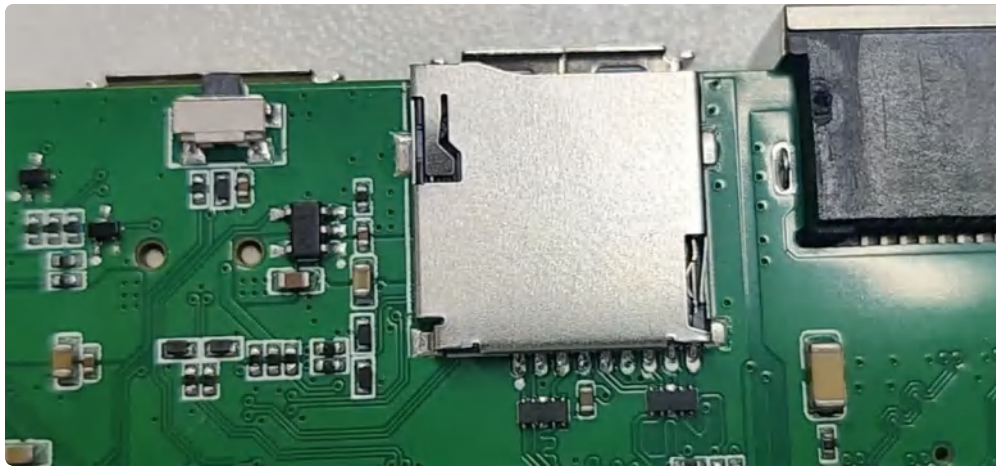
```

1  ## host
2  echo host > /sys/devices/platform/fe8a0000.usb2-phy/otg_mode
3  ## device
4  echo peripheral > /sys/devices/platform/fe8a0000.usb2-phy/otg_mode

```

4. TF CARD

SBC3566配置了一个TF CARD接口，当TF CARD接口插入TF卡后，会自动挂载到/sdcard目录下，如下图所示：



TF Card位置如上图所示，支持FAT32和NTFS格式分区自动挂载。

```
Shell |
1 root@linaro-alip:/mnt# df -h
2 [ 457.763315] FAT-fs (mmcblk1p1): FAT read failed (blocknr 16)
3 文件系统          容量  已用  可用  已用%  挂载点
4 /dev/root          5.9G  3.1G  2.5G   56%  /
5 devtmpfs           973M   8.0K  973M    1%  /dev
6 tmpfs              983M    0    983M    0%  /dev/shm
7 tmpfs              394M   1.5M  392M    1%  /run
8 tmpfs              5.0M   4.0K   5.0M    1%  /run/lock
9 tmpfs              983M   20K   983M    1%  /tmp
10 /dev/mmcblk0p7     121M   12M  101M   11%  /oem
11 /dev/mmcblk0p8     23G    40K   22G    1%  /userdata
12 tmpfs              197M   40K   197M    1%  /run/user/1000
13 /dev/mmcblk1p2     29G    42M   27G    1%  /sdcard
14
```

5. 以太网

EVB3566配置了一个100M以太网接口，对应的网络设备节点为eth0。

5.1. 查看以太网IP地址

5.1.1. 使用命令查看

系统默认以太网为动态获取IP，当以太网接口插入网线时，会自动获取IP。

```
▼ | Bash |
1 root@linaro-alip:/# ifconfig
2 eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
3     inet 192.168.0.72 netmask 255.255.255.0 broadcast 192.168.0.255
4     inet6 fe80::64f5:8dd8:439:6960 prefixlen 64 scopeid 0x20<link>
5     ether ba:28:12:85:ae:52 txqueuelen 1000 (Ethernet)
6     RX packets 4016 bytes 337728 (329.8 KiB)
7     RX errors 0 dropped 119 overruns 0 frame 0
8     TX packets 150 bytes 12297 (12.0 KiB)
9     TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
10    device interrupt 48
11
```

5.2. 使用命令设置

```
▼ | Bash |
1 root@linaro-alip:/# ifconfig eth0 192.168.0.112
2
```

5.3. 设置以太网永久静态IP

修改配置文件：/etc/network/interfaces，以设置静态ip 192.168.1.123为例。

```
▼ | Bash |
1 auto lo
2 iface lo inet loopback
3
4 auto eth0
5 iface eth0 inet static
6 address 192.168.0.234
7 netmask 255.255.255.0
8 gateway 192.168.0.1
9 dns-nameservers 114.114.114.114
```

然后重启网络

```
▼ | Bash |
1 systemctl restart networking
```

重启网络后，eth0的ip地址已经变成刚才设置的静态ip。

设置静态IP后，断电重启设备依旧生效。

6. WiFi

系统上电默认会打开WiFi，对应的网络设备节点为wlan0。

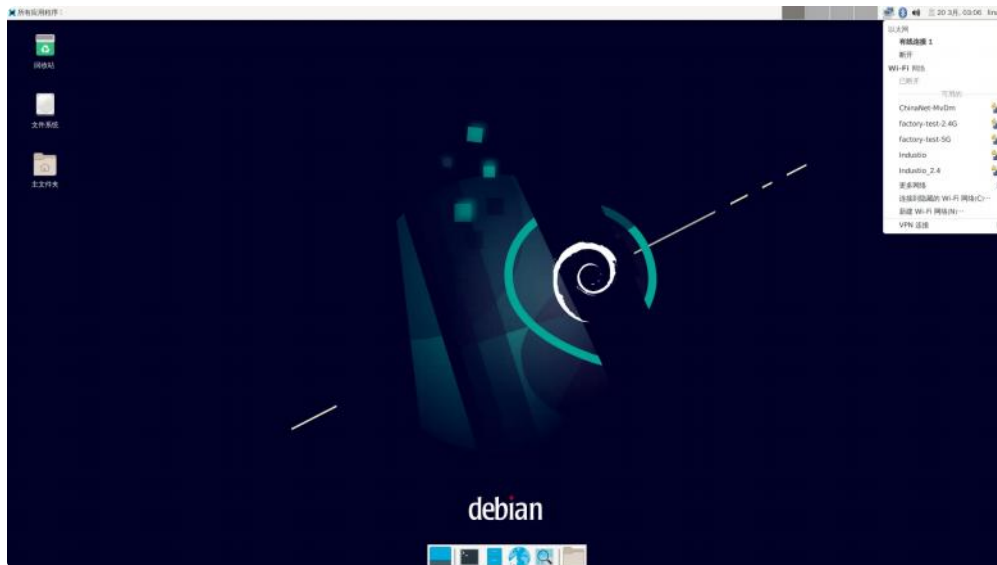
```
root@linaro-alip:~# ifconfig wlan0
wlan0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
        ether c0:f5:35:12:1d:dc txqueuelen 1000 (Ethernet)
        RX packets 0 bytes 0 (0.0 B)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 0 bytes 0 (0.0 B)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions
```

6.1. 连接热点

连接热点可以在桌面上操作，也可以使用命令行操作。

6.1.1. 方式一

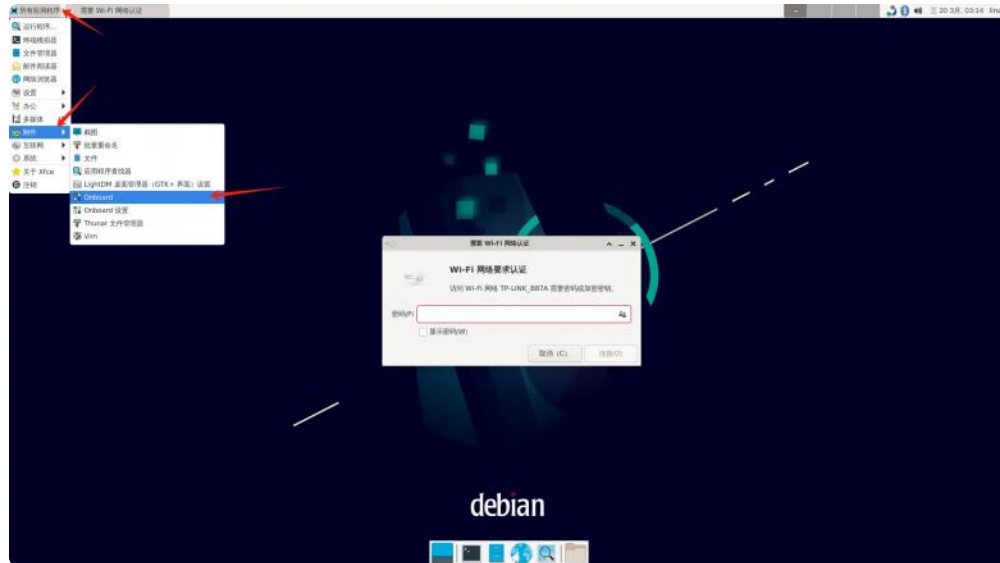
点击桌面右上角的网络图标，即可看到WiFi热点列表，如图所示：



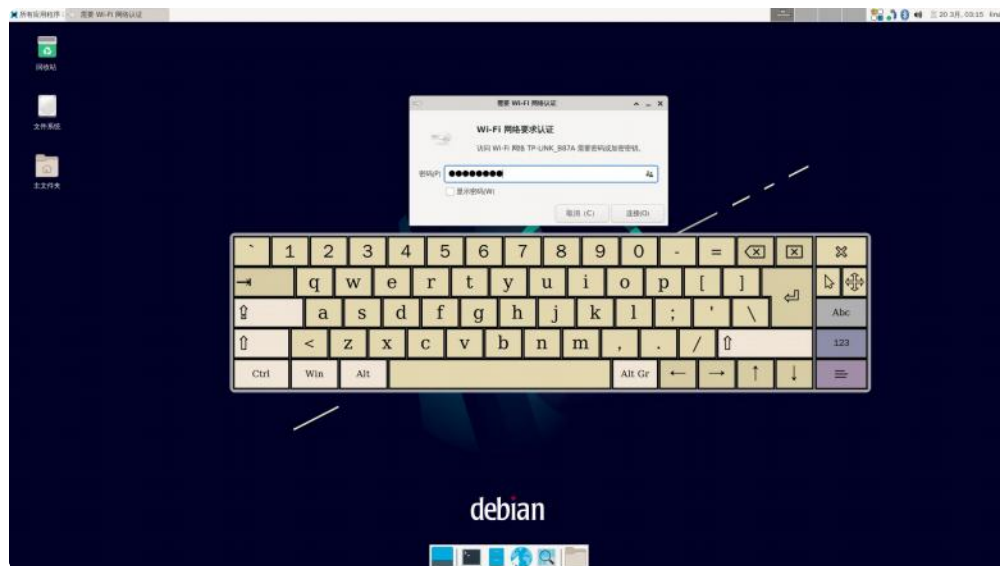
如果有连接键盘，直接输入密码即可；如果没有连接键盘可以下载一个软件

```
1 sudo apt install onboard
```

点击桌面左上角的应用图标，然后依次点击【所有应用程序】->【附件】->【Onboard】，如图所示：



使用软键盘，点击要连接的热点，弹出密码输入窗口，如图所示：



看终端可以知道，已经成功启动了

```

1 root@linaro-alip:/# ifconfig wlan0
2 wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
3     inet 192.168.1.189 netmask 255.255.255.0 broadcast 192.168.1.255
4     inet6 fe80::d8be:24bf:9d2:f95f prefixlen 64 scopeid 0x20<link>
5     ether c0:f5:35:12:1d:dc txqueuelen 1000 (Ethernet)
6     RX packets 47 bytes 5497 (5.3 KiB)
7     RX errors 0 dropped 1 overruns 0 frame 0
8     TX packets 31 bytes 3319 (3.2 KiB)
9     TX errors 0 dropped 0 overruns 0 carrier 0 collisions

```

6.1.2. 方式二

修改/etc/network/interfaces

```

1 # interfaces(5) file used by ifup(8) and ifdown(8)
2 # Include files from /etc/network/interfaces.d:
3 source-directory /etc/network/interfaces.d
4 allow-hotplug wlan0
5 auto wlan0
6 iface wlan0 inet dhcp
7 wpa-conf /etc/wpa.conf

```

新建/etc/wpa.conf

```

1 linaro@linaro-alip:~$ sudo touch /etc/wpa.conf
2 linaro@linaro-alip:~$ sudo chmod a+w /etc/wpa.conf

```

然后使用wpa_passphrase连接WiFi热点:

```

1 linaro@linaro-alip:~$ sudo killall wpa_supplicant
2 linaro@linaro-alip:~$ sudo wpa_passphrase TP-LINK_B87A 12345678 >> /etc/wpa.conf
3 linaro@linaro-alip:~$ sudo ifup wlan0

```

连接成功后使用ifconfig命令可查看wlan0的IP:

```
▼ | Bash
1  linaro@linaro-alip:~$ sudo ifconfig wlan0
2  wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
3      inet 192.168.1.165 netmask 255.255.255.0 broadcast 192.168.1.255
4      inet6 fe80::868:5528:86fe:c9da prefixlen 64 scopeid 0x20<link>
5      ether 2c:d2:6b:10:ea:4d txqueuelen 1000 (Ethernet)
6      RX packets 28 bytes 55800 (54.4 KiB)
7      RX errors 0 dropped 0 overruns 0 frame 0
8      TX packets 40 bytes 64335 (62.8 KiB)
9      TX errors 0 dropped 11 overruns 0 carrier 0 collisions 0
```

6.1.3. 方式三

直接使用nmcli工具进行测试

```
▼ | Bash
1  sudo nmcli d wifi connect cainiaocl password 12345678
```

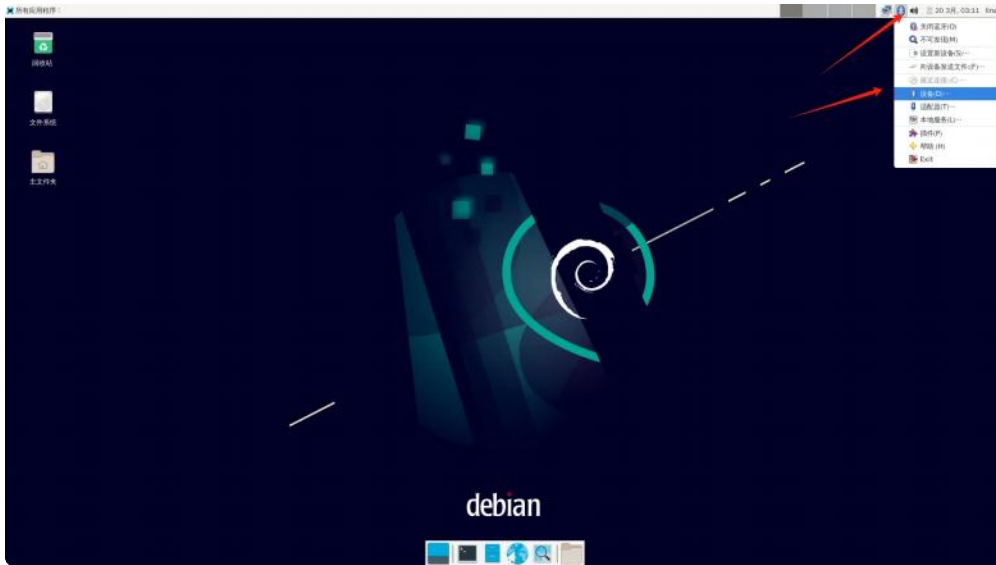
7. 蓝牙

系统开机默认打开蓝牙，对应的网络节点为hci0。

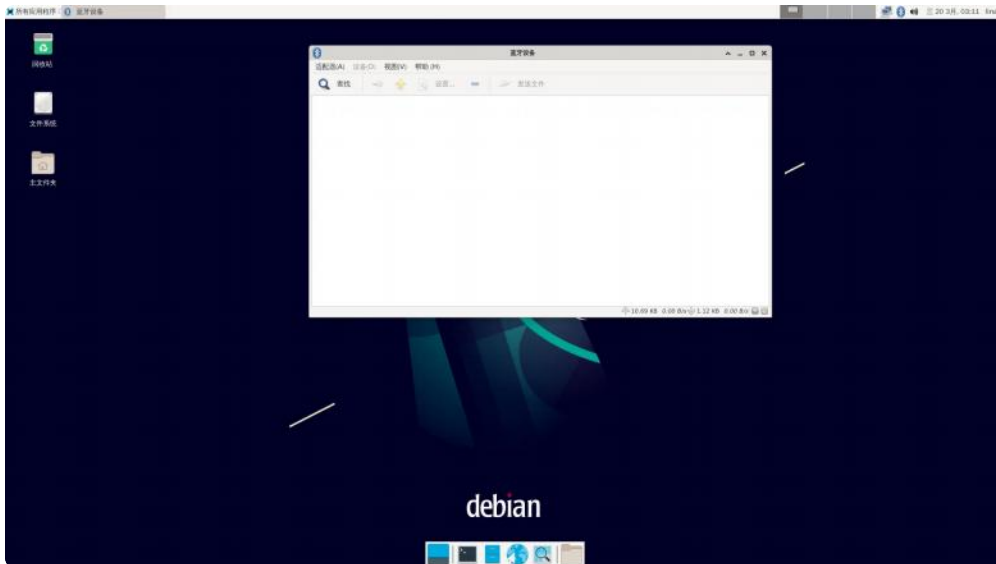
```
▼ | Bash
1  linaro@linaro-alip:~$ hciconfig
2  hci0:  Type: Primary  Bus: UART
3      BD Address: 2C:D2:6B:11:AC:71  ACL MTU: 1021:8  SCO MTU: 255:12
4      UP RUNNING
5      RX bytes:1531 acl:0 sco:0 events:51 errors:0
6      TX bytes:5012 acl:0 sco:0 commands:51 errors:0
7
8  linaro@linaro-alip:~$
```

7.1. 连接蓝牙设备

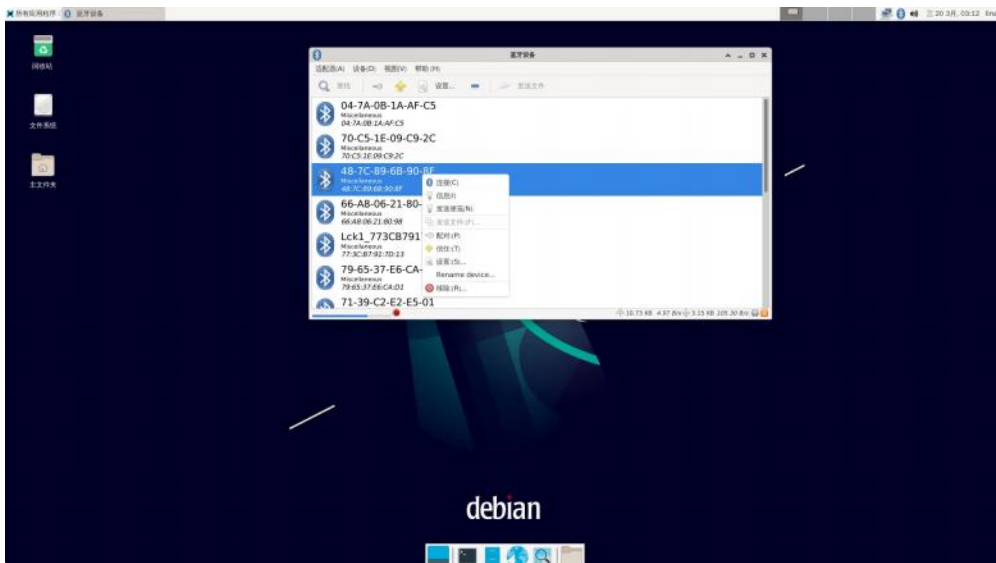
将鼠标放到桌面右上角蓝牙图标，点击设备，如图所示：



在弹出的窗口中，点击【查找】，如图所示：



选中要连接设备，右键->配对，即可连接该设备：



8. 音频

8.1. 查看声卡设备

```
1 root@linaro-alip:/# aplay -l
2 **** List of PLAYBACK Hardware Devices ****
3 card 0: rockchiphdmi [rockchip,hdmi], device 0: rockchip,hdmi i2s-hifi-0 [r
   ockchip,hdmi i2s-hifi-0]
4   Subdevices: 1/1
5   Subdevice #0: subdevice #0
6 card 1: rockchiprk809co [rockchip,rk809-codec], device 0: fe410000.i2s-rk81
   7-hifi rk817-hifi-0 [fe410000.i2s-rk817-hifi rk817-hifi-0]
7   Subdevices: 1/1
8   Subdevice #0: subdevice #0
9
```

上面的第一个声卡是hdmi的，第二个声卡是喇叭的，因为我接的hdmi屏幕，如果接的其他屏幕如edp, lvds, mipi, 则只有一个声卡，如：

```
1 root@rk3566-buildroot:/# aplay -l
2 **** List of PLAYBACK Hardware Devices ****
3 card 0: rockchiprk809co [rockchip,rk809-codec], device 0: fe410000.i2s-rk81
   7-hifi rk817-hifi-0 [fe410000.i2s-rk817-hifi rk817-hifi-0]
4   Subdevices: 1/1
5   Subdevice #0: subdevice #0
6
```

8.2. 播放音频

播放到HDMI：

```
1 aplay -D plug:spk_c0 /usr/share/sounds/alsa/Rear_Center.wav
```

播放到Lineout：

不插入耳机，执行以下命令。

```
▼ | Bash |
1 aplay -D plug:spk_c1 /usr/share/sounds/alsa/Rear_Center.wav
```

播放到耳机：

插入耳机，执行以下命令。

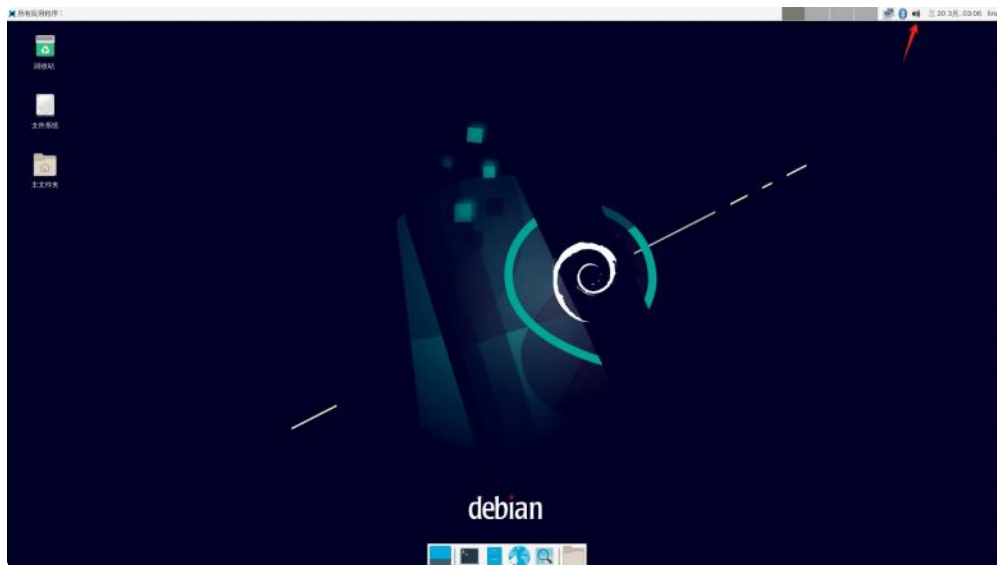
```
▼ | Bash |
1 aplay -D plug:spk_c1 /usr/share/sounds/alsa/Rear_Center.wav
```

注意：这里是根据你的声卡选择，如果是接的其他屏幕，如mipi，那么只有一个声卡的情况下，喇叭选择的应该是

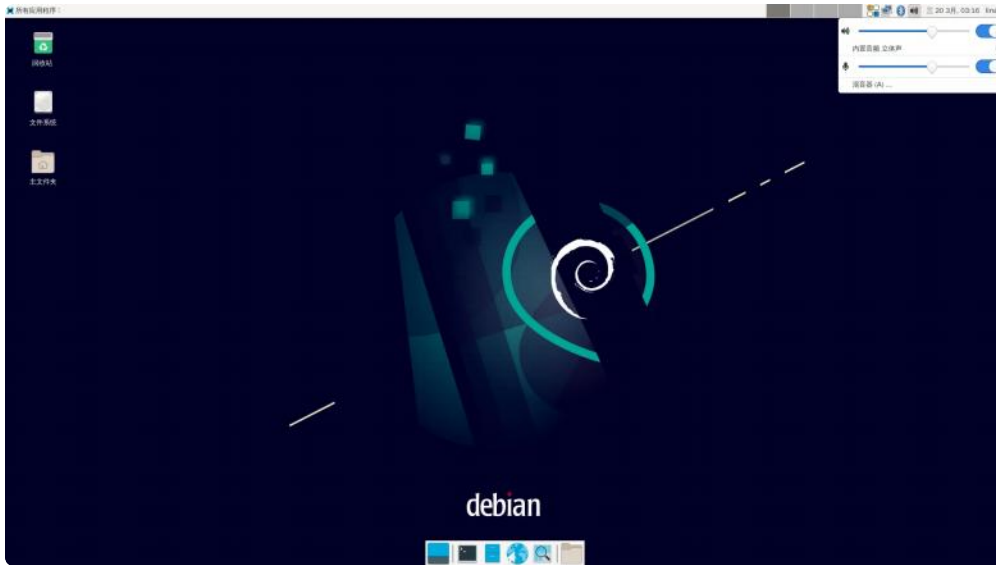
```
▼ | Bash |
1 aplay -D plug:spk_c0 /usr/share/sounds/alsa/Rear_Center.wav
```

8.3. 音量的调节

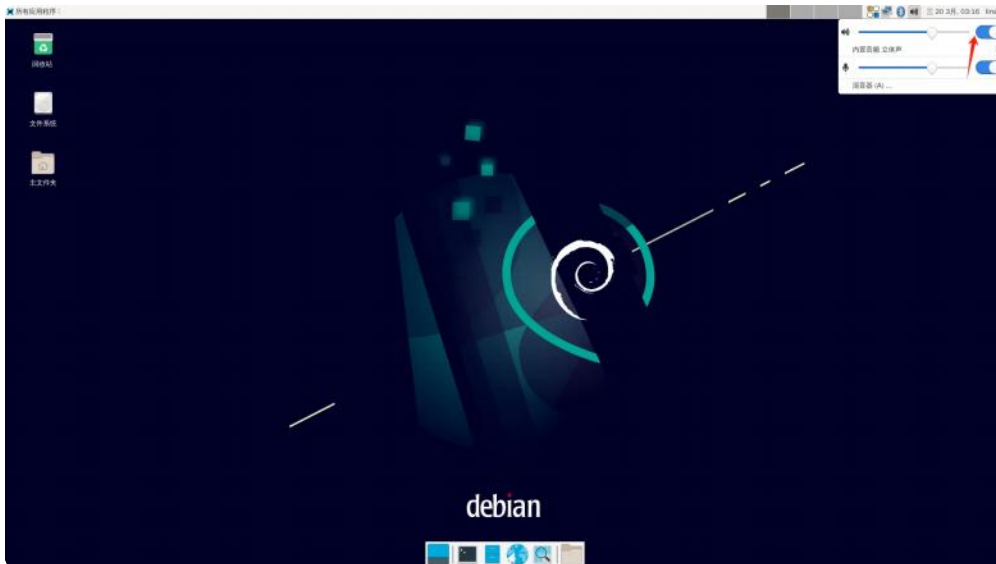
桌面右上角点击蓝牙图标，如图所示：



弹出的窗口中，在回放页面中设置系统音量，如图所示：



当需要静音时，点击静音按钮即可，如图所示：



8.4. 录音

我们可以使用麦克风也可以使用耳机自带的麦克风进行录音，如图所示：





EVB3566 PH2.54 4pin 喇叭接口，一路OTMP标准四节耳机座。喇叭支持最大 $8\Omega@5W$ 。

接口4pin从上到下，如表所示：

序号	功能
1	VOLN
2	VOLP
3	VORN
4	VORP

使用arecord工具可以进行录音测试：

```

▼ Bash |
1 root@linaro-alip:/# arecord -D hw:1,0 -r 48000 -c 2 -f S16_LE test.wav
2 Recording WAVE 'test.wav' : Signed 16 bit Little Endian, Rate 48000 Hz, Stereo
3 ^X^CAborted by signal 中断...
4

```

录音完后播放测试：

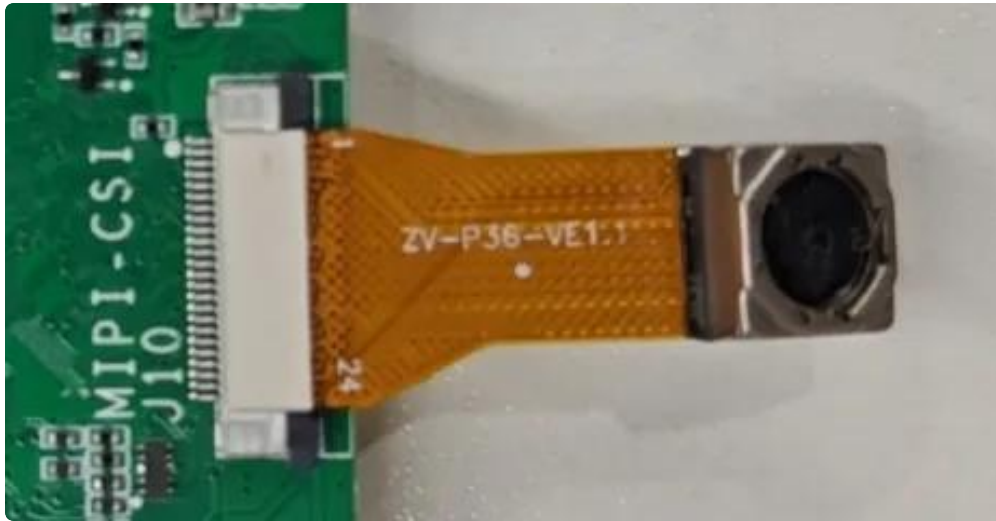
```

▼ Bash |
1 root@linaro-alip:/# aplay -D plug:spk_c1 test.wav
2 Playing WAVE 'test.wav' : Signed 16 bit Little Endian, Rate 48000 Hz, Stereo
3 ^CAborted by signal 中断...

```

9. 摄像头

EVB3566默认适配OV5648（500万像素）摄像头，如下图所示：



OV8858（800万像素）摄像头，如下图所示：



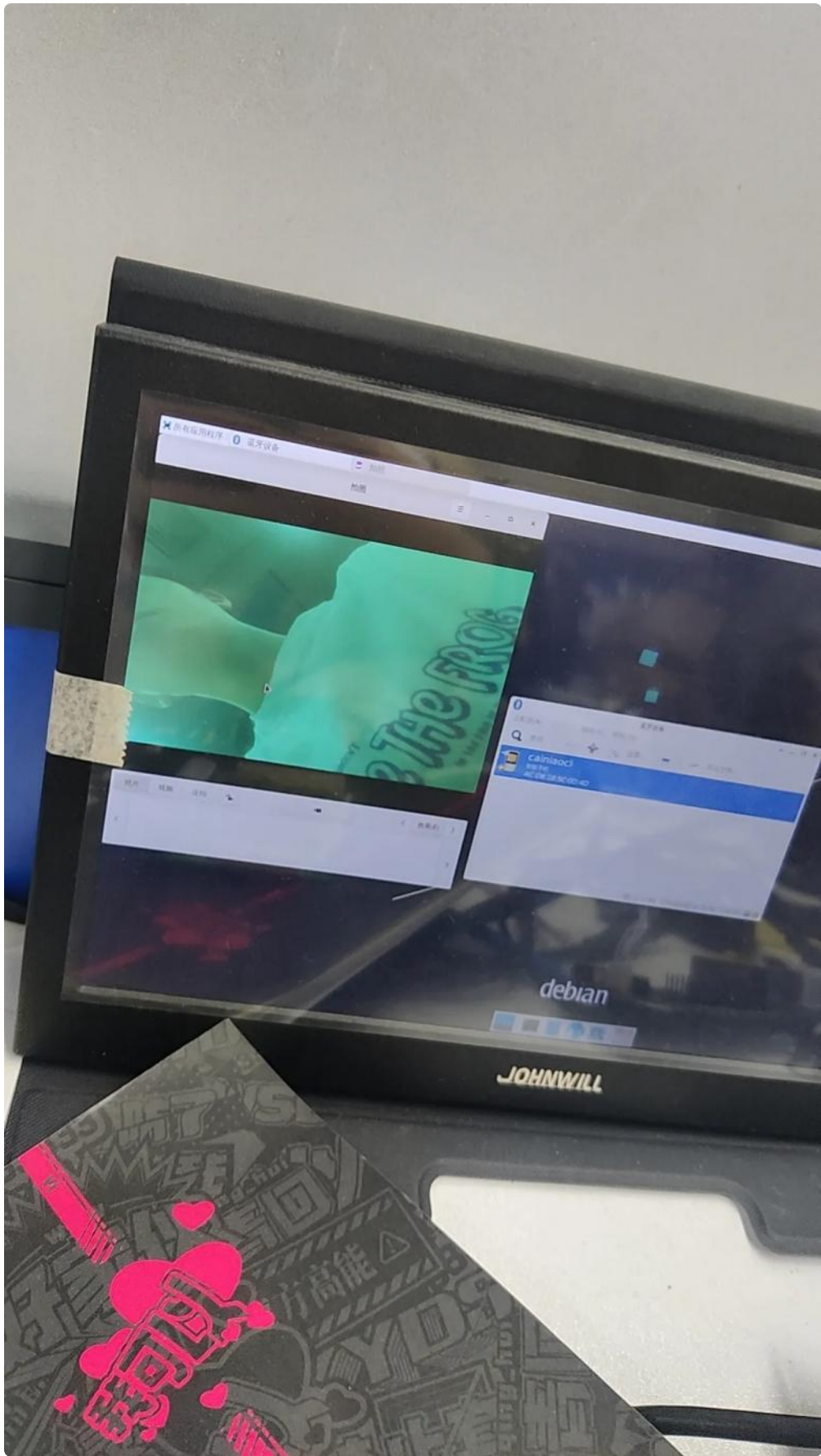
9.1. 测试

使用Debian自带的【茄子】程序，可以测试摄像头功能。

依次点击桌面右上角的【所有程序】->【多媒体】->【茄子】即可启动茄子程序测试摄像头功能。也可以直接输入

```
▼ | Bash |
1  cheese
```

显示效果，如下下图所示：



9.2. 查看摄像头信息

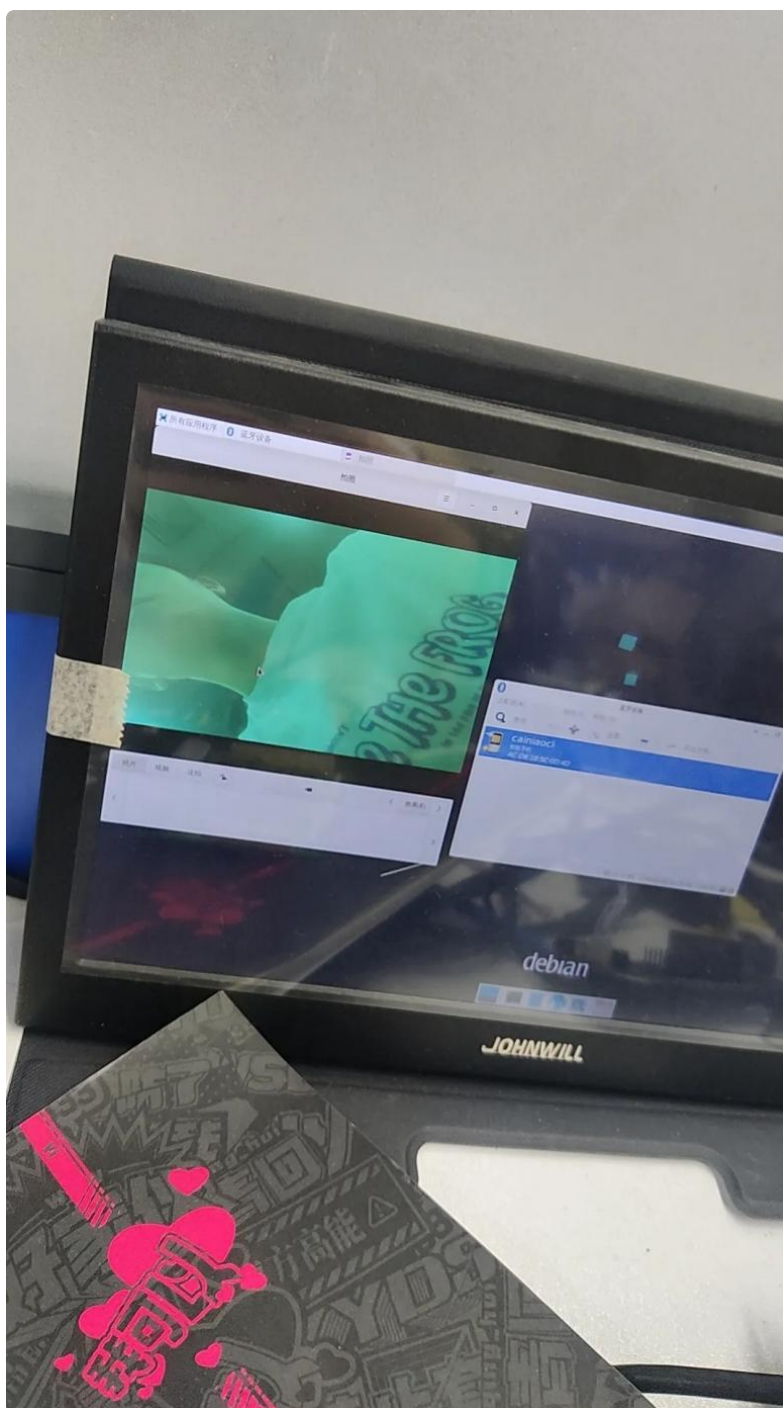
可以看到最后的打印信息，这里使用的ov5648

使用Debian自带的【茄子】程序，可以测试摄像头功能。

依次点击桌面右上角的【所有程序】->【多媒体】->【茄子】即可启动茄子程序测试摄像头功能。
也可以直接输入

```
▼ | Bash |
1 cheese
```

显示效果如下



10. RTC

主板/dev/rtc0为外部RTC（HYM8563），系统默认使用rtc0的时间。

10.1. 获取RTC时间

```
▼ Bash |
1 root@linaro-alip:~$ sudo hwclock
2 2022-11-10 02:16:23.617474+00:00
3 root@linaro-alip:~$
```

10.2. 设置RTC时间

```
▼ Bash |
1 root@linaro-alip:/# date -s '2023-12-15 17:16:03'
2 2023年 12月 15日 星期五 17:16:03 UTC
3 root@linaro-alip:/# hwclock -w //写入rtc中
4 root@linaro-alip:/# hwclock -r
5 2023-12-15 17:17:58.142161+00:00
```

重启过几分钟，读取rtc的时间为：

```
▼ Bash |
1 root@linaro-alip:/# hwclock -r
2 2023-12-15 17:20:12.858640+00:00
```

11. 开机自启动

默认系统开机会运行/etc/rc.local脚本，将要开机执行的程序放到该脚本中即可。

12. 屏幕控制

12.1. 背光调节

通过修改/sys/class/backlight/backlight/brightness的值，实现背光的调节，范围取0–255，值越大，亮度越高。

设置亮度为100：

```
▼ Bash |
1  linaro@linaro-alip:~$ sudo chmod a+w /sys/class/backlight/backlight/brightn
   ess
2  linaro@linaro-alip:~$ sudo echo 100 > /sys/class/backlight/backlight/brigh
   tness
```

12.2. 屏幕旋转

使用xrandr工具可以实现屏幕的旋转。

12.2.1. 临时旋转

系统启动后，执行xrandr -o normal,inverted,left,right，可以实现临时旋转屏幕方向，其中normal表示顺时针旋转0度，inverted表示顺时针旋转180度，left表示顺时针旋转270度，right表示顺时针旋转90度。

```
▼ Bash |
1  root@linaro-alip:~$ xrandr -o inverted
```

12.2.2. 永久旋转

通过修改启动文件/home/linaro/.config/lxsession/LXDE/autostart，将xrandr命令内置，实现永久旋转，主板重启后依旧生效。

```
▼ Bash |
1  root@linaro-alip:~$ cat /home/linaro/.config/lxsession/LXDE/autostart
2  @lxpanel --profile LXDE
3  @pcmanfm --desktop --profile LXDE
4  @xscreensaver -no-splash
5  xrandr -o inverted
6  root@linaro-alip:~$
```

经过以上修改，每次启动主板后，桌面会顺时针旋转180度。

13. ADC

EVB3566配置了2路ADC，位于J46的第4，5引脚，分别记作ADC1和ADC2。精度为10位。

13.1. ADC转换方法

$$V = (\text{raw}/1024)*1.8\text{v}$$

其中raw为对应设备节点读取的值，范围为0–1023，如下表所示：

编号	设备节点
ADC1	/sys/bus/iio/devices/iio:device0/in_voltage3_raw
ADC2	/sys/bus/iio/devices/iio:device0/in_voltage2_raw

13.2. 测试

以测试ADC1为例，ADC2测试方法类似。

```
▼ Bash |  
1  linaro@linaro-alip:~$ cat /sys/bus/iio/devices/iio:device0/in_voltage3_raw  
2  997
```

设备节点读取的raw值为997，代入到公式计算：

$$V=(997/1024)*1.8\text{v}=1.75\text{v}$$

即ADC1输入的电压为1.75v。

14. 网络优先级设置

EVB3566支持以太网、WiFi和两种网络，通过路由表来设置它们的网络优先级。

14.1. 查看路由表

```
1  linaro@linaro-alip:~$ route
2  Kernel IP routing table
3  Destination      Gateway            Genmask           Flags Metric Ref    Use Ifa
4  ce
5  default          _gateway          0.0.0.0           UG    100   0      0 eth
6  0
7  default          _gateway          0.0.0.0           UG    600   0      0 wla
8  n0
9  192.168.1.0      0.0.0.0           255.255.255.0    U     100   0      0 eth
10 0
11 192.168.1.0      0.0.0.0           255.255.255.0    U     600   0      0 wla
12 n0
```

14.2. 设置默认路由

14.2.1. 设置WiFi为默认路由

当前以太网和WiFi同时使用，设置WiFi优先：

```

1  linaro@linaro-alip:~$ route
2  Kernel IP routing table
3  Destination      Gateway            Genmask           Flags Metric Ref    Use Ifa
4  ce
5  default          _gateway          0.0.0.0           UG        100    0      0    et
6  h0
7  default          _gateway          0.0.0.0           UG        600    0      0    wl
8  an0
9  192.168.1.0      0.0.0.0           255.255.255.0    U         100    0      0    et
10 h0
11 192.168.1.0      0.0.0.0           255.255.255.0    U         600    0      0    wl
12 an0
13 linaro@linaro-alip:~$ route del default dev eth0
14 linaro@linaro-alip:~$ route
15 Kernel IP routing table
16 Destination      Gateway            Genmask           Flags Metric Ref    Use Ifa
17 ce
18 default          _gateway          0.0.0.0           UG        600    0      0    wl
19 an0
20 192.168.1.0      0.0.0.0           255.255.255.0    U         100    0      0    et
21 h0
22 192.168.1.0      0.0.0.0           255.255.255.0    U         600    0      0    wl
23 an0

```

这样默认路由就是wlan0了，即优先使用WiFi进行数据通信。

14.2.2. 设置以太网为默认路由

当前以太网和WiFi同时使用，且WiFi优先：

```

1  linaro@linaro-alip:~$ route
2  Kernel IP routing table
3  Destination      Gateway            Genmask           Flags Metric Ref    Use Ifa
4  ce
5  default          _gateway          0.0.0.0           UG        600    0      0    wlan0
6  192.168.1.0      0.0.0.0           255.255.255.0    U         100    0      0    eth0
7  192.168.1.0      0.0.0.0           255.255.255.0    U         600    0      0    wlan0
8  root@ido:~#

```

设置为以太网优先:

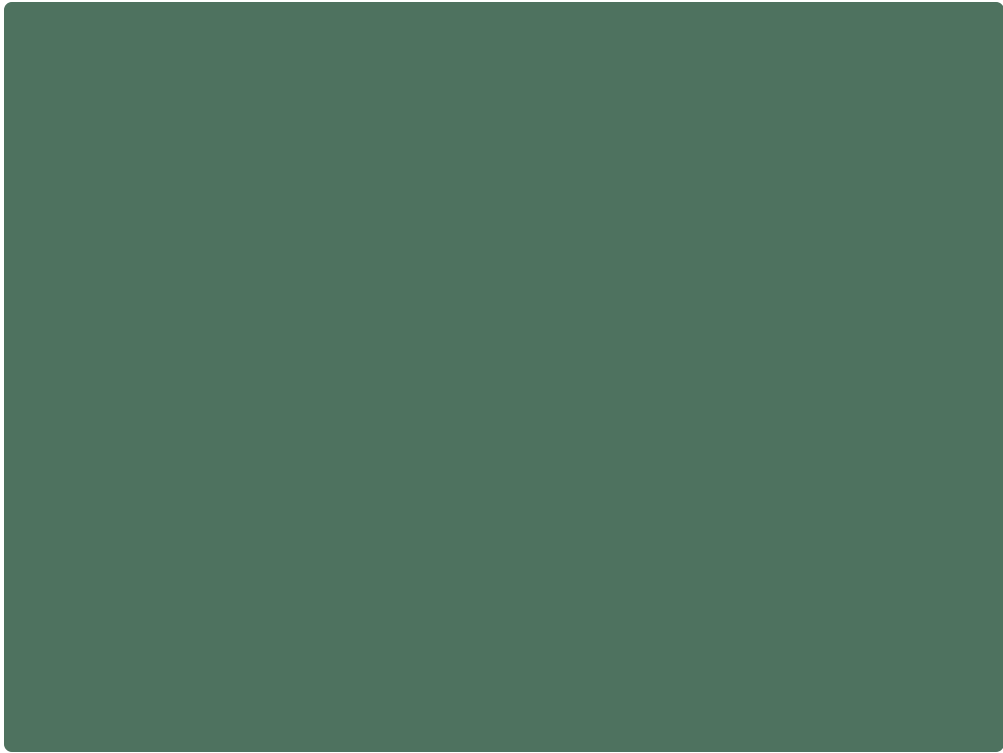
```
▼ Bash |  
1  linaro@linaro-alip:~$ route del default dev wlan0  
2  linaro@linaro-alip:~$ route add default dev eth0  
3  linaro@linaro-alip:~$ route add default gw 192.168.1.1  
4  linaro@linaro-alip:~$ route  
5  Kernel IP routing table  
6  Destination      Gateway            Genmask           Flags Metric Ref    Use If  
7  default          0.0.0.0           0.0.0.0           U        0      0      0 et  
8  192.168.1.0      0.0.0.0           255.255.255.0    U        100     0      0 et  
9  192.168.1.0      0.0.0.0           255.255.255.0    U        600     0      0 wl  
10 root@ido:~#
```

其他情况按照类似的方法进行处理即可。

IDO-EVB3566-V1 Buildroot系统

1. 硬件资源概况

1.1. 主板照片,如下图所示:



IDO-EVB3566-V1A正面实物图



IDO-EVB3566-V1A背面实物图

1.2. 硬件资源及设备节点,如下表所示:

基本参数

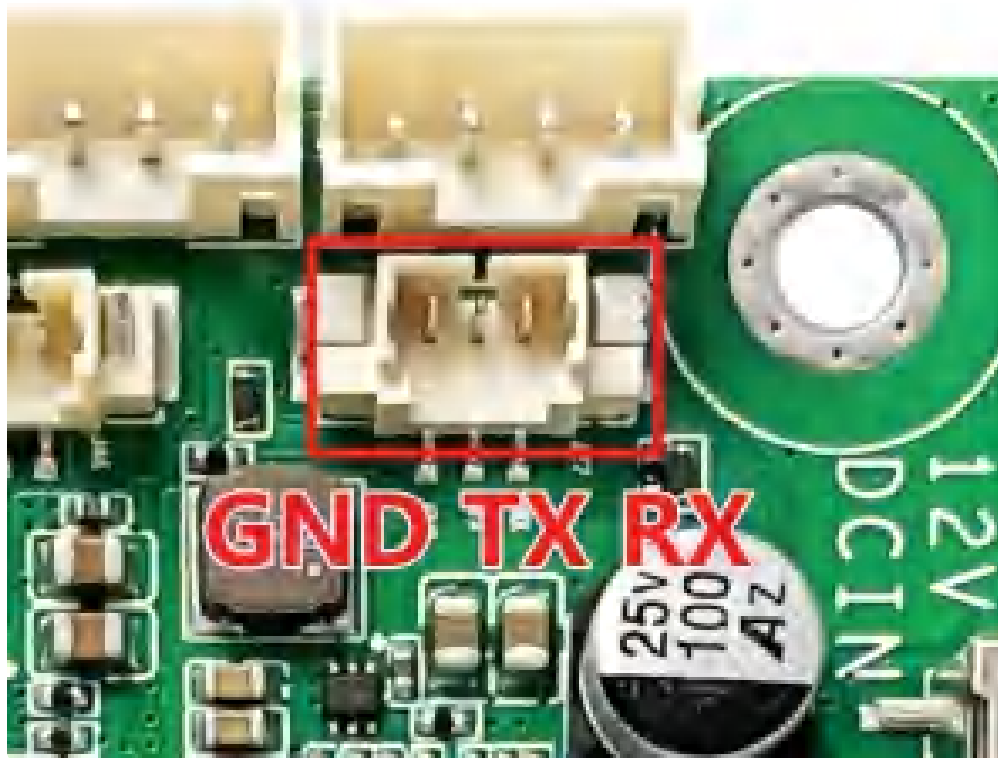
SOC	RockChip RK3566
CPU	四核 64 位Cortex-A55 处理器，主频最高1.8GHz
GPU	Mali-G52 1-Core-2EE 支持 OpenGL ES 1.1/2.0/3.2, OpenCL 2.0, Vulkan 1.1 内嵌高性能2D 加速硬件
NPU	神经网络加速引擎，处理性能高达1个TOPS 支持INT8/INT16/FP16/BFP16 MAC混合操作 支持深度学习框架TensorFlow, TF-lite, Pytorch, Caffe, ONNX, MXNet, Keras, Darknet等模型
VPU	支持4K 60fps H.264/H.265/VP9 视频解码 支持1080P 100fps H.264/H.265视频编码 支持8M ISP
内存	LPDDR4/LPDDR4X 默认2GB（最高支持8GB）
存储	eMMC 默认8GB（可选16GB/32GB/64GB）
硬件参数	
以太网	1 × 百兆以太网（1000 Mbps）
无线网络	支持双频2.4G/5.8G Wifi 支持BT4.2及以上
显示接口	支持双屏同显 1 × HDMI2.0, 支持4K@60fps 输出 1 × MIPI_DSI_TX, 支持1920*1080@60fps 输出 1 × LVDS, 支持 1920*1080@60fps 输出 1 × eDP1.3, 支持 2560*1600@60fps 输出
摄像头	1 × MIPI_CSI 摄像头接口, 支持单4-Lane 13M Sensor。
4G/5G	1 × Mini PCIe 扩展 4G LTE（或1 × M.2 扩展5G）

音频接口	1 × HDMI 音频输出 2 × Speaker, 最大4Ω@10W喇叭输出 1 × 麦克风音频输入 1 × 耳机MIC音频输入 1 × 耳机音频输出
扩展接口	1 × USB 3.0 HOST (标准TYPE-A母座) 4 × USB 2.0 HOST (PH-2.0mm-4P线对板连接器) 1 × USB 2.0 OTG (标准TYPE-A母座) 1 × Debug (UART2) 2 × RS232 (可设置为UART/RS485) 2 × RS232 (可设置为UART) 1 × RS485 (可设置为UART) 1 × SPI 1 × I2C 2 × ADC 2 × PWM 2 × Speaker (4Ω@10W) 1 × MIC 1 × IR 1 × TP 座 (I2C)

2. 调试

2.1. 串口调试

通信参数为1500000 8 N 1, 电平状态为TTL电平,如下图所示:

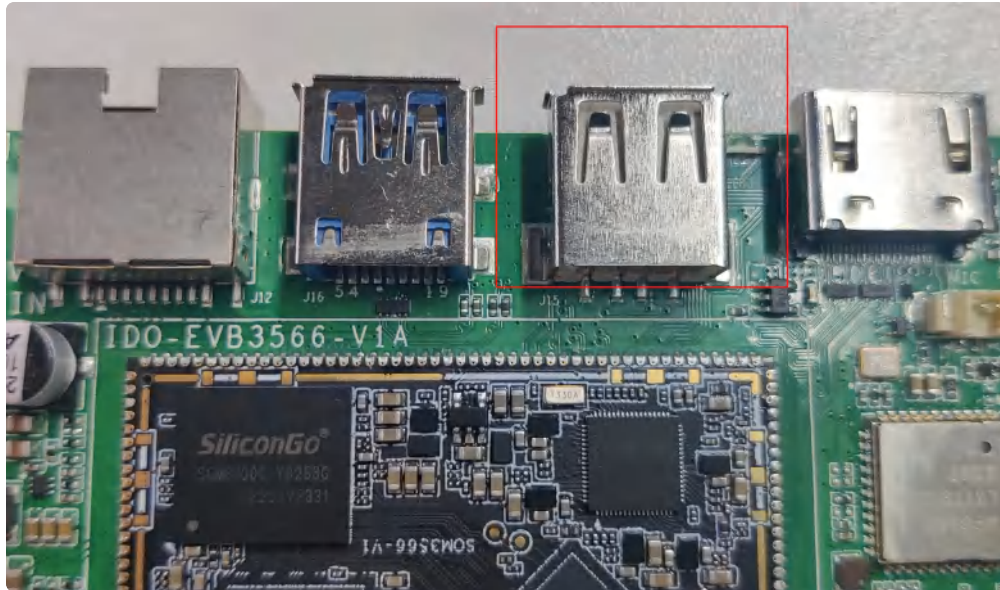


串口引脚参数，如下表所示：

序号	定义	电平/V	说明
1	UART2_RX	3.3V	调试串口信号输入
2	UART2_TX	3.3V	调试串口信号输出
3	GND	电源地	电源地

2.2. ADB调试

ADB调试端口位于j29，使用双公头USB线，连接EVB3566和电脑，即可在电脑上使用adb调试,如下图所示：



```
▼ Bash |  
1  
2 C:\Users\industio_1>adb shell  
3 root@rk3566-buildroot:/# ls  
4 bin info misc root test.wav vendor  
5 busybox.fragment lib mnt run tmp  
6 cset lib64 oem sbin udisk  
7 data linuxrc opt sdcard userdata  
8 dev lost+found proc sys usr  
9 etc media rockchip-test system var  
10 root@rk3566-buildroot:/#
```

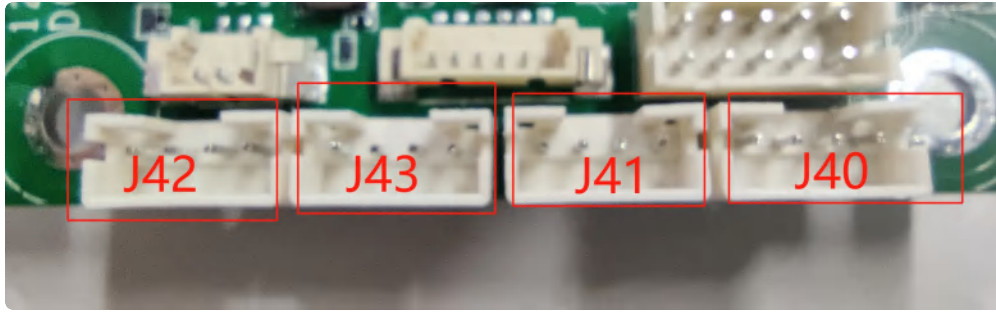
2.3. ssh调试

系统默认登录账号密码为root @ rockchip。

ssh登录需要知道主板的IP，获取IP的方法，请参考第5章以太网的说明。

3. UART

如下图所示：



IDO-EVB3566-V1主板扩展2路UART和2路RS232（不含调试UART），4路串口通过4个PH2.0-4P立贴座子接出（默认2路UART TTL、2路RS232），如下表所示：

连接器（设备节点）	UART TTL	RS232
J40 (/dev/ttyS7)	✓	✓（默认功能）
J41 (/dev/ttyS9)	✓	✓（默认功能）
J42 (/dev/ttyS0)	✓（默认功能）	✓
J43 (/dev/ttyS5)	✓（默认功能）	✓

3.1. 测试方法

以上串口均可以使用microcom工具进行测试

```

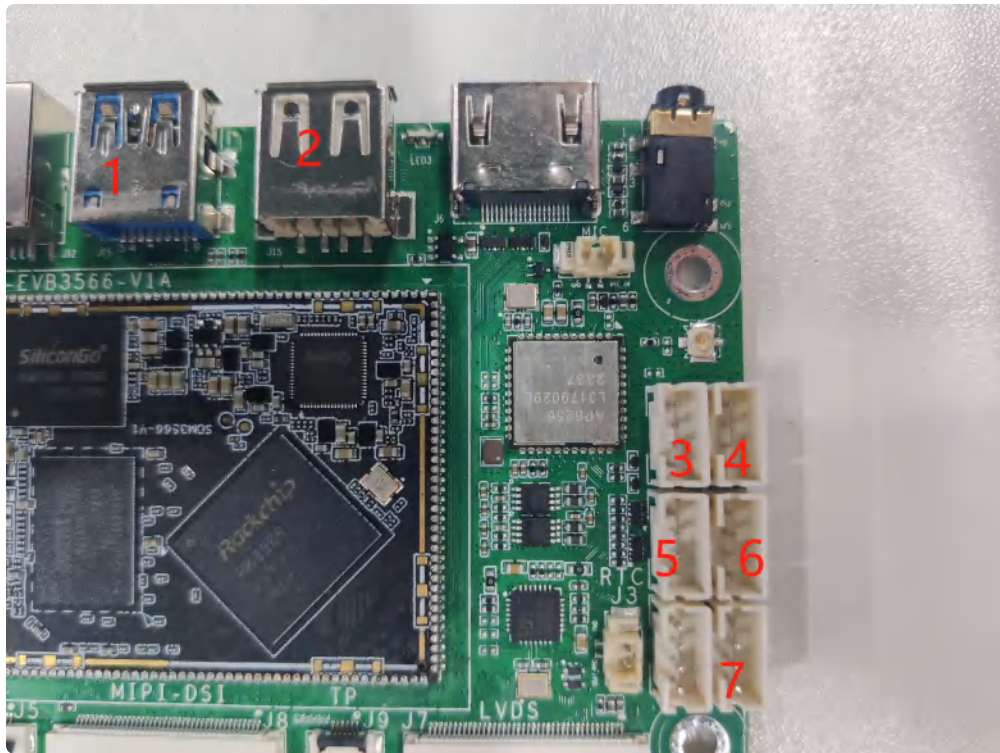
1 [root@RK356X:/]# microcom -s 115200 -X /dev/ttyS7

```

当按下键盘时，串口会发送对应的字符，而接收的内容会显示在终端，这里由于buildroot带的microcom工具的关系，需要重启，才能退出。

4. USB

USB接口，如下图所示：



功能说明，如下表所示：

序号	功能	控电节点
1	USB 3.0 HOST	/
2	USB OTG	/
3	USB 2.0 HOST	/sys/class/leds/usb2_host2_pwr/brightness
4	USB 2.0 HOST	/sys/class/leds/usb2_host_pwr/brightness
5	USB 2.0 HOST	/sys/class/leds/usb2_host3_pwr/brightness
6	USB 2.0 HOST	/sys/class/leds/usb2_host1_pwr/brightness
7	USB 2.0 HOST	/sys/class/leds/usb2_host4_pwr/brightness

供电控制说明，设备节点写0关闭电源，写1开启电源

命令行控制方法如下，以端口7为例

```

Shell |
1 #关闭
2 echo 0 > /sys/class/leds/usb2_host4_pwr/brightness
3 #开启（默认状态）
4 echo 1 > /sys/class/leds/usb2_host4_pwr/brightness

```

USB OTG 切换命令

上电状态说明

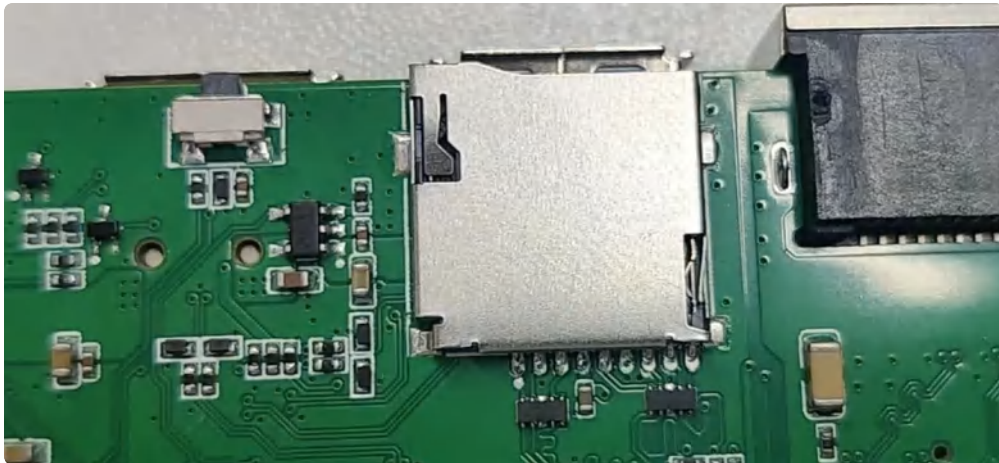
上电外设连接	模式说明
上电前，使用USB Type A 数据线，连接主板和PC	上电后默认为device模式
上电前，插着U盘或者未接USB设备	上电后默认作为host模式

USB OTG 支持host 和device 模式的切换，软件切换方法如下

```
▼ Shell |
1  ## host
2  echo host > /sys/devices/platform/fe8a0000.usb2-phy/otg_mode
3  ## device
4  echo peripheral > /sys/devices/platform/fe8a0000.usb2-phy/otg_mode
```

5. TF CARD

TF Card支持FAT32和NTFS格式分区自动挂载，TF Card位置如下图所示：



```

1 root@rk3566-buildroot:/# df -h
2 Filesystem      Size  Used Avail Use% Mounted on
3 /dev/root       1.4G  702M  555M  56% /
4 devtmpfs        972M   8.0K  972M   1% /dev
5 tmpfs           983M  172K  983M   1% /tmp
6 tmpfs           983M  288K  983M   1% /run
7 tmpfs           983M     0  983M   0% /dev/shm
8 /dev/mmcblk0p7  121M   12M  101M  11% /oem
9 /dev/mmcblk0p8  23G   244K   22G   1% /userdata
10 /dev/mmcblk1p2  29G   42M   27G   1% /mnt/sdcard
11

```

6. 以太网

EVB3566配置了一个100M以太网接口，对应的网络设备节点为eth0。

6.1. 查看以太网IP地址

6.1.1. 使用命令查看

系统默认以太网为动态获取IP，当以太网接口插入网线时，会自动获取IP。

```

1 root@rk3566-buildroot:/# ifconfig
2 eth0      Link encap:Ethernet  HWaddr BA:28:12:85:AE:52
3          inet addr:192.168.0.138  Bcast:192.168.0.255  Mask:255.255.255.0
4          inet6 addr: fe80::8c31:1099:2ddf:b3aa/64 Scope:Link
5          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
6          RX packets:101 errors:0 dropped:0 overruns:0 frame:0
7          TX packets:14 errors:0 dropped:0 overruns:0 carrier:0
8          collisions:0 txqueuelen:1000
9          RX bytes:8617 (8.4 KiB)  TX bytes:1508 (1.4 KiB)
10         Interrupt:48
11

```

6.2. 设置以太网临时IP地址

6.2.1. 使用命令设置

```

1 root@rk3566-buildroot:/# ifconfig eth0 192.168.0.112
2

```

6.3. 设置以太网永久静态IP

以eth0设置静态IP地址为例，修改/etc/network/interfaces，在文件中添加如下内容

```

1 auto lo
2 iface lo inet loopback
3
4 auto eth0
5 iface eth0 inet static
6 address 192.168.0.234
7 netmask 255.255.255.0
8 gateway 192.168.0.1
9 dns-nameservers 114.114.114.114

```

其中，dns-nameservers一项为默认dns。然后重启即可

7. WiFi

在使用 WIFI时连接好WiFi天线，设备节点为wlan0

```

1 [root@RK356X:/]# ifconfig wlan0
2 wlan0      Link encap:Ethernet  HWaddr C0:F5:35:12:1D:E6
3           inet addr:169.254.148.169  Bcast:169.254.255.255  Mask:255.255.
4           0.0
5           inet6 addr: fe80::a760:26e4:b3ba:c681/64 Scope:Link
6           UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
7           RX packets:0 errors:0 dropped:0 overruns:0 frame:0
8           TX packets:16 errors:0 dropped:0 overruns:0 carrier:0
9           collisions:0 txqueuelen:1000
10          RX bytes:0 (0.0 B)  TX bytes:3120 (3.0 KiB)

```

修改/userdata/cfg/wpa_supplicant.conf，填写正确的热点账号和密码：

```
Bash |
1 [root@RK356X:/]# cat /userdata/cfg/wpa_supplicant.conf
2 ctrl_interface=/var/run/wpa_supplicant
3 ap_scan=1
4 update_config=1
5
6 network={
7     ssid="TP-LINK_B87A"
8     psk="12345678"
9     key_mgmt=WPA-PSK
10 }
11 [root@RK356X:/]#
```

执行命令，连接wifi

```
Bash |
1 wpa_supplicant -D nl80211 -i wlan0 -c /userdata/cfg/wpa_supplicant.conf -B
```

测试

```
Bash |
1 root@rk3566-buildroot:/# ping baidu.com
2 PING baidu.com (110.242.68.66) 56(84) bytes of data.
3 64 bytes from 110.242.68.66 (110.242.68.66): icmp_seq=1 ttl=49 time=46.5 m
  s
4 64 bytes from 110.242.68.66 (110.242.68.66): icmp_seq=2 ttl=49 time=52.6 m
  s
5 64 bytes from 110.242.68.66 (110.242.68.66): icmp_seq=3 ttl=49 time=59.8 m
  s
6 64 bytes from 110.242.68.66 (110.242.68.66): icmp_seq=6 ttl=49 time=154 ms
7 64 bytes from 110.242.68.66 (110.242.68.66): icmp_seq=7 ttl=49 time=50.8 m
  s
8 64 bytes from 110.242.68.66 (110.242.68.66): icmp_seq=8 ttl=49 time=51.2 m
  s
9 64 bytes from 110.242.68.66 (110.242.68.66): icmp_seq=9 ttl=49 time=60.5 m
  s
10 64 bytes from 110.242.68.66 (110.242.68.66): icmp_seq=10 ttl=49 time=63.1
   ms
11 64 bytes from 110.242.68.66 (110.242.68.66): icmp_seq=11 ttl=49 time=62.0
   ms
12 64 bytes from 110.242.68.66 (110.242.68.66): icmp_seq=12 ttl=49 time=52.7
   ms
13
```


8. 连接蓝牙设备

设备节点为hci0，我们已经做了开机自启动开启蓝牙功能

蓝牙功能开启后，将产生hci0节点

```
▼ Bash |  
1 root@rk3566-buildroot:/# hciconfig  
2 hci0: Type: Primary Bus: UART  
3 BD Address: C0:F5:35:12:1D:E7 ACL MTU: 1021:8 SCO MTU: 64:1  
4 UP RUNNING  
5 RX bytes:871 acl:0 sco:0 events:61 errors:0  
6 TX bytes:3282 acl:0 sco:0 commands:61 errors:
```

使用hcidool测试蓝牙扫描功能

```
▼ Bash |  
1 root@rk3566-buildroot:/# hcidool -i hci0 scan  
2 Scanning ...  
3 AC:D6:18:5C:0D:4D cainiaocl
```

9. 音频

耳机接口，如下图所示：



EVB3566 PH2.54 4pin 喇叭接口，一路OTMP标准四节耳机座。喇叭支持最大8Ω@5W，如下图所示：



接口4pin从上到下，如下表所示：

序号	功能
1	VOLN
2	VOLP
3	VORN
4	VORP

9.1. 查看声卡设备

```

▼ Bash |
1 root@rk3566-buildroot:/# aplay -l
2 **** List of PLAYBACK Hardware Devices ****
3 ▼ card 0: rockchiphdmi [rockchip,hdmi], device 0: rockchip,hdmi i2s-hifi-0 [r
   ockchip,hdmi i2s-hifi-0]
4   Subdevices: 1/1
5   Subdevice #0: subdevice #0
6 ▼ card 1: rockchiprk809co [rockchip,rk809-codec], device 0: fe410000.i2s-rk81
   7-hifi rk817-hifi-0 [fe410000.i2s-rk817-hifi rk817-hifi-0]
7   Subdevices: 1/1
8   Subdevice #0: subdevice #0
9

```

这里第一个声卡是hdmi的，第二个声卡是喇叭的，因为我接的hdmi屏幕，如果接的其他屏幕如edp, lvds, mipi, 则只有一个声卡

```
▼ | Bash |
1 root@rk3566-buildroot:/# aplay -l
2 **** List of PLAYBACK Hardware Devices ****
3 ▼ card 0: rockchiprk809co [rockchip,rk809-codec], device 0: fe410000.i2s-rk81
   7-hifi rk817-hifi-0 [fe410000.i2s-rk817-hifi rk817-hifi-0]
4   Subdevices: 1/1
5   Subdevice #0: subdevice #0
6
```

9.2. 播放音频

播放到HDMI:

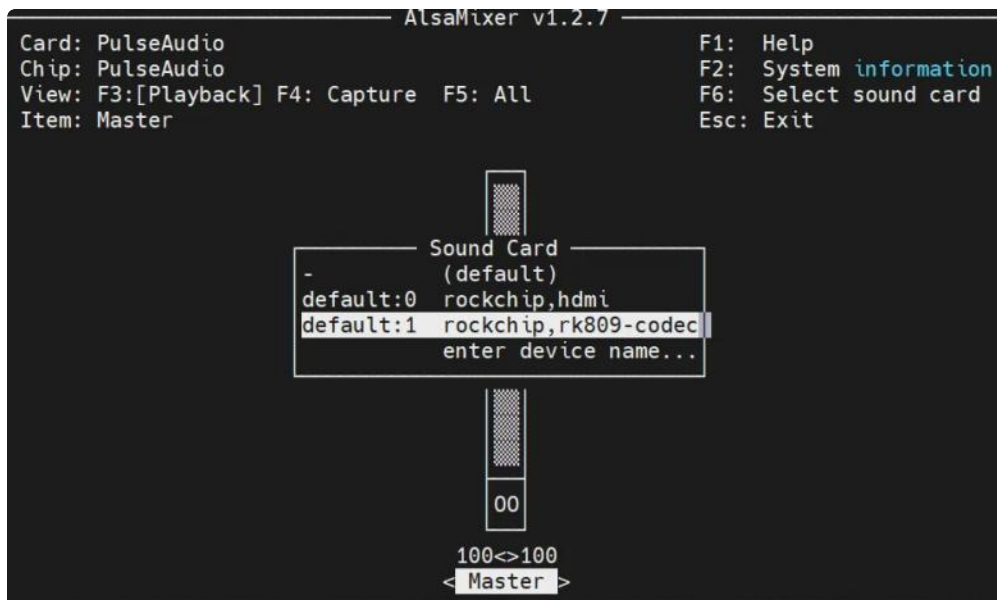
```
▼ | Bash |
1 aplay -D plug:spk_c0 /usr/share/sounds/alsa/Rear_Center.wav
```

不插入耳机，播放到Lineout:

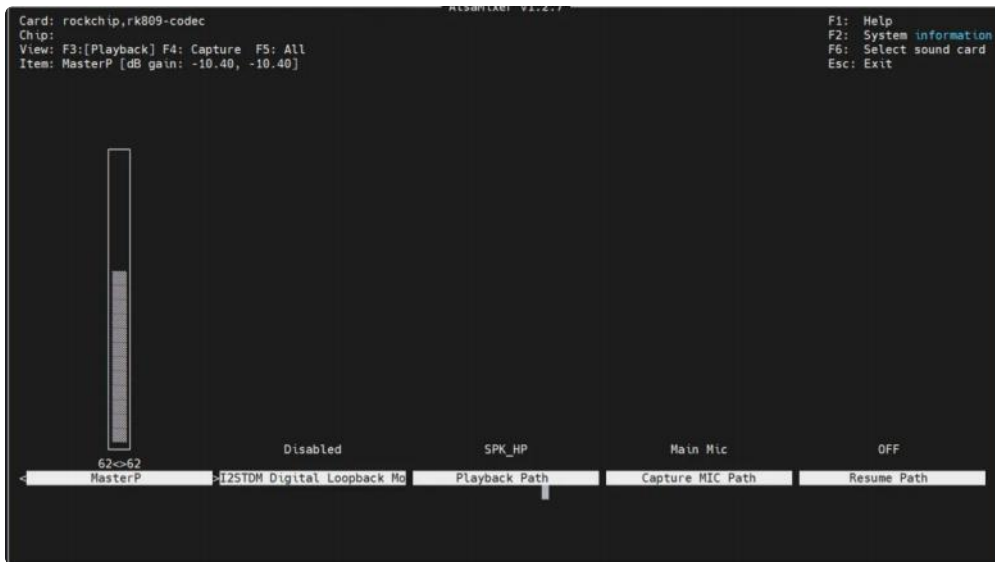
进入alsa调试页面

```
▼ | Bash |
1 alsamixer
```

按s选择需要配置的声卡，这里选择配置耳机喇叭default: 1，如下图所示:



然后进入，配置成这种模式，如下图所示:



按Esc退出，自动保存，然后执行播放命令

```

1 aplay -D plug:spk_c1 /usr/share/sounds/alsa/Rear_Center.wav
  
```

播放到耳机：

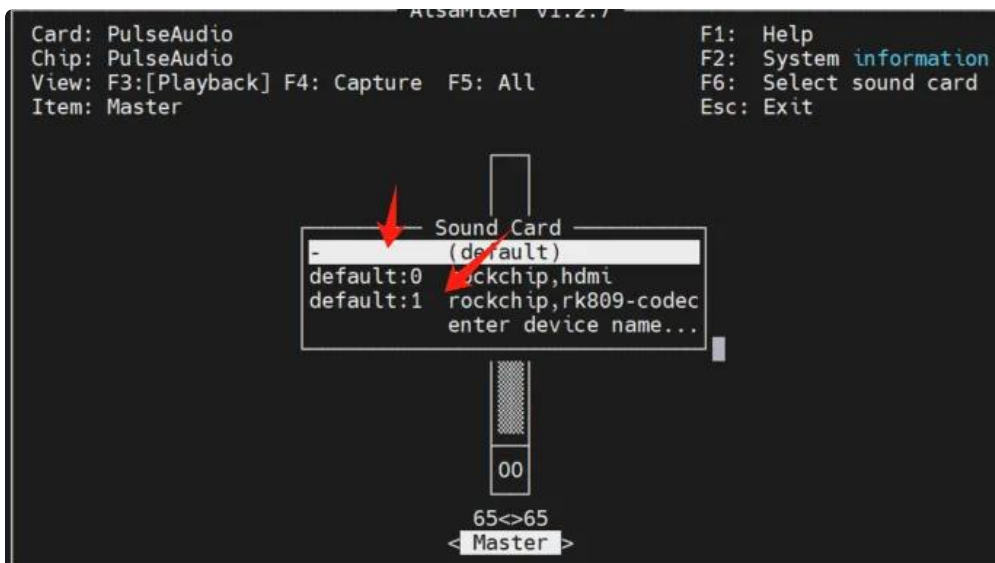
插入耳机，执行以下命令。

```

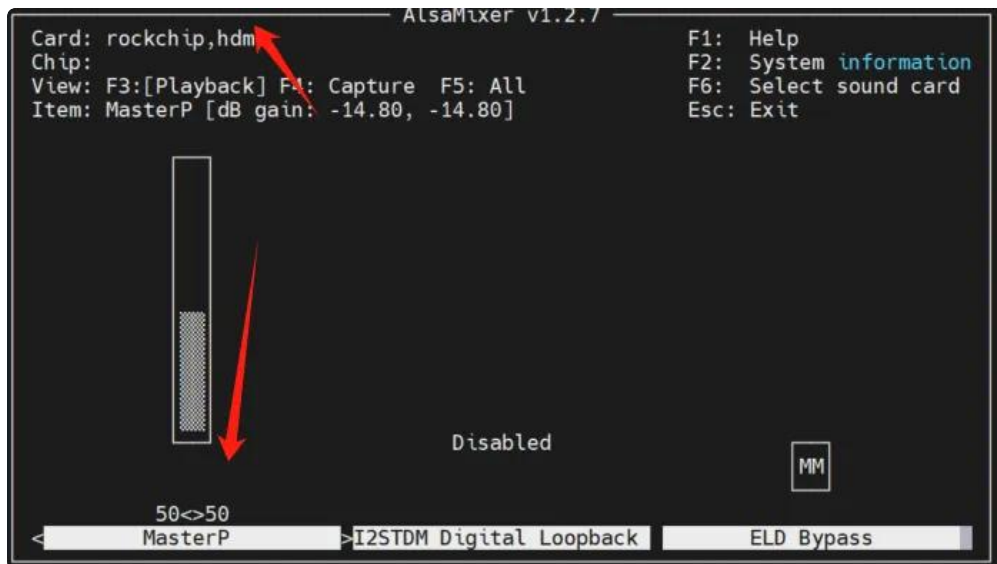
1 aplay -D plug:spk_c1 /usr/share/sounds/alsa/Rear_Center.wav
  
```

音量调节

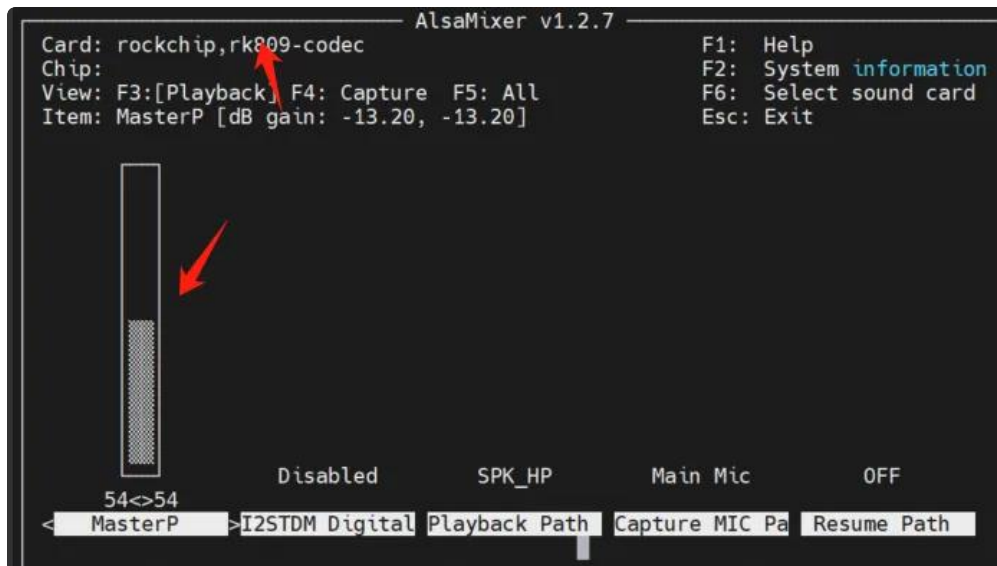
我们可以选择声卡对音量进行调节,通过上下左右箭头进行声卡拉高拉低，如下图所示：



hdmi声卡音量调节，如下图所示：



喇叭/耳机声卡音量调节，如下图所示：



注意：这里是根据你的声卡选择，如果是接的其他屏幕，如mipi，那么只有一个声卡的情况下，喇叭选择的应该是

```

1  aplay -D plug:spk_c0 /usr/share/sounds/alsa/Rear_Center.wav

```

9.3. 录音

使用的是耳机录音测试，当然也可以使用麦克风进行测试

使用arecord工具可以进行录音测试：

```
▼ Bash |  
1 root@rk3566-buildroot:/# arecord -D hw:1,0 -r 48000 -c 2 -f S16_LE test.wav  
2 Recording WAVE 'test.wav' : Signed 16 bit Little Endian, Rate 48000 Hz, Stereo  
3 ^CAborted by signal Interrupt...
```

录音完后播放测试:

播放到耳机/linkout

```
▼ Bash |  
1 aplay -D plug:spk_c1 /usr/share/sounds/alsa/Rear_Center.wav  
2 Playing WAVE 'test.wav' : Signed 16 bit Little Endian, Rate 48000 Hz, Stereo
```

播放到hdmi

```
▼ Bash |  
1 aplay -D plug:spk_c0 /usr/share/sounds/alsa/Rear_Center.wav  
2
```

10. RTC

主板/dev/rtc0为外部RTC (HYM8563) , 系统默认使用rtc0的时间。

10.1. 获取RTC时间

```
▼ Bash |  
1 root@rk3566-buildroot:/# hwclock -r
```

10.2. 设置RTC时间

```
▼ Bash |  
1 root@rk3566-buildroot:/# date -s '2023-12-14 17:19:10'  
2 root@rk3566-buildroot:/# hwclock -w  
3 root@rk3566-buildroot:/# hwclock -r  
4 Thu Dec 14 17:19:39 2023 0.000000 seconds  
5 root@rk3566-buildroot:/# sync  
6 root@rk3566-buildroot:/# reboot
```

11. 开机自启动

默认系统开机会运行/etc/init.d/Sxxx脚本，默认是执行S开头的脚本，将要开机执行的程序放到该脚本中即可。

12. 屏幕控制

12.1. 背光调节

通过修改/sys/class/backlight/backlight/brightness的值，实现背光的调节，范围取0-255，值越大，亮度越高。

设置亮度为100：

```
▼ Bash |  
1 [root@RK356X:/]# chmod a+w /sys/class/backlight/backlight/brightness  
2 [root@RK356X:/]# echo 100 > /sys/class/backlight/backlight/brightness
```

13. 按键

EVB3566配置了1个recovery按键，这个按键，插入电前长按可以进入烧录模式。

14. 网络优先级设置

EVB3566支持以太网、WiFi和4G/5G三种网络，通过路由表来设置它们的网络优先级。

14.1. 查看路由表

```
▼ Bash |  
1 root@ido:~# route  
2 Kernel IP routing table  
3 Destination      Gateway           Genmask          Flags Metric Ref    Use Ifa  
4 default          _gateway         0.0.0.0          UG    100   0      0 eth  
5 default          _gateway         0.0.0.0          UG    600   0      0 wla  
6 192.168.1.0      0.0.0.0          255.255.255.0   U     100   0      0 eth  
7 192.168.1.0      0.0.0.0          255.255.255.0   U     600   0      0 wla
```

14.2. 设置默认路由

14.2.1. 设置WiFi为默认路由

当前以太网和WiFi同时使用，设置WiFi优先：


```

▼ | Bash
1 root@ido:~# route
2 Kernel IP routing table
3 Destination Gateway Genmask Flags Metric Ref Use If
4 default _gateway 0.0.0.0 UG 100 0 0 et
5 default _gateway 0.0.0.0 UG 600 0 0 wl
6 192.168.1.0 0.0.0.0 255.255.255.0 U 100 0 0 et
7 192.168.1.0 0.0.0.0 255.255.255.0 U 600 0 0 wl
8 root@ido:~# route del default dev eth0
9 root@ido:~# route
10 Kernel IP routing table
11 Destination Gateway Genmask Flags Metric Ref Use If
12 default _gateway 0.0.0.0 UG 600 0 0 wl
13 192.168.1.0 0.0.0.0 255.255.255.0 U 100 0 0 et
14 192.168.1.0 0.0.0.0 255.255.255.0 U 600 0 0 wl

```

这样默认路由就是wlan0了，即优先使用WiFi进行数据通信。

14.2.2. 设置以太网为默认路由

当前以太网和WiFi同时使用，且WiFi优先：

```

▼ | Bash
1 root@ido:~# route
2 Kernel IP routing table
3 Destination Gateway Genmask Flags Metric Ref Use Ifa
4 default _gateway 0.0.0.0 UG 600 0 0 wla
5 192.168.1.0 0.0.0.0 255.255.255.0 U 100 0 0 eth
6 192.168.1.0 0.0.0.0 255.255.255.0 U 600 0 0 wla
7 root@ido:~#

```

设置为以太网优先:

```
▼ Bash |
1 root@ido:~# route del default dev wlan0
2 root@ido:~# route add default dev eth0
3 root@ido:~# route add default gw 192.168.1.1
4 root@ido:~# route
5 Kernel IP routing table
6 Destination      Gateway            Genmask           Flags Metric Ref    Use If
7 default          0.0.0.0           0.0.0.0           U         0       0      0 et
8 h0
9 192.168.1.0      0.0.0.0           255.255.255.0    U        100      0      0 et
10 h0
11 192.168.1.0      0.0.0.0           255.255.255.0    U        600      0      0 wl
12 an0
13 root@ido:~#
```

其他情况按照类似的方法进行处理即可。

15. 测试GPIO输入输出

以GPIO96为例，输入输出方向控制方法如下

1. 设置方向为输入，并读取接口电平

```
▼ Shell |
1 echo 96 > /sys/class/gpio/export
2 # 设置方向为输入
3 echo in > /sys/class/gpio/gpio96/direction
4 # 读取IO口电平值
5 cat /sys/class/gpio/gpio96/value
```

2. 设置方向为输出，设置输出电平

```
1 echo 96 > /sys/class/gpio/export
2 # 设置方向为输出
3 echo out > /sys/class/gpio/gpio96/direction
4 # 设置IO口输出高电平
5 echo 1 > /sys/class/gpio/gpio96/value
6 # 设置IO口输出低电平
7 echo 0 > /sys/class/gpio/gpio96/value
```