

IDO-Purple Pi OH RK3566-V1 Linux5.10使用手册



IDO-Purple Pi OH RK3566-V1 Linux5.10使用手册

深圳触觉智能科技有限公司

www.industio.cn

文档修订历史

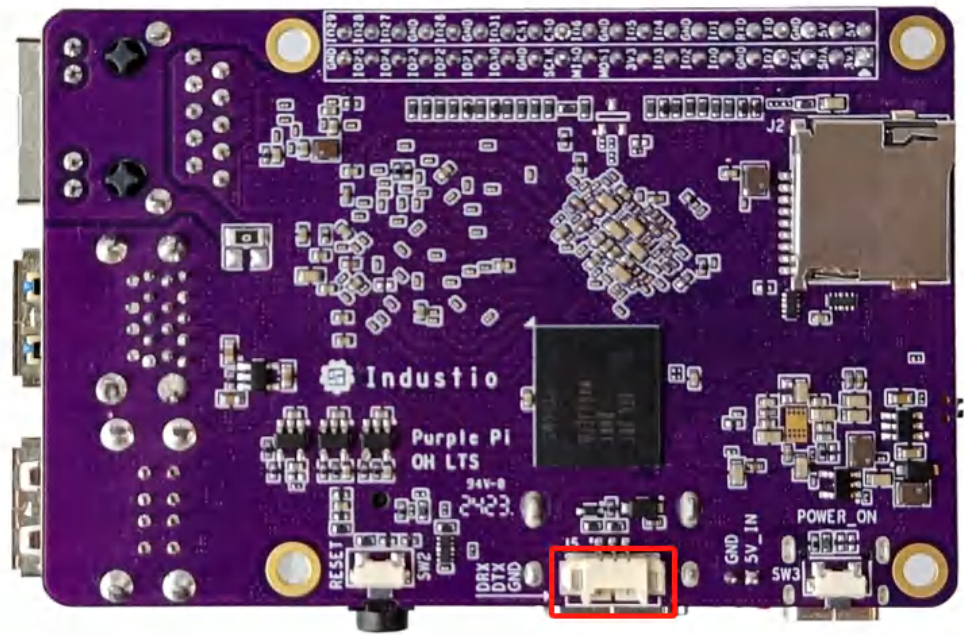
版本	PCBA版本	修订内容	修订	审核	日期
V1.0	V1B	创建文档；	TWX	IDO	2023/03/17
V1.1	V1B	修改USB1描述为USB-HOST2.0	TWX	IDO	2023/03/28
V1.2	V1B	修改耳机描述为"3.5mm 美标"	TWX	IDO	2023/04/26
V1.3	V1B	删除蓝牙连接方式二； 删除音量调节方式二； 修正功能接口序号；	TWX	IDO	2023/05/15
V1.4	V1B	优化文档	MCC	IDO	2024/05/16
V1.5	V1B	修正文档格式；	TWX		2024/08/09
V1.6	V1B	修改Ubuntu 摄像头预览命令；	TWX		2024/10/25

Purple Pi OH RK3566–Debian系统

1 调试

1.1 串口调试

主板调试串口位于板子背面（J5），建议使用配套的USB转串口工具，如下图所示：



1. 打开MobaXterm，下载链接如下：

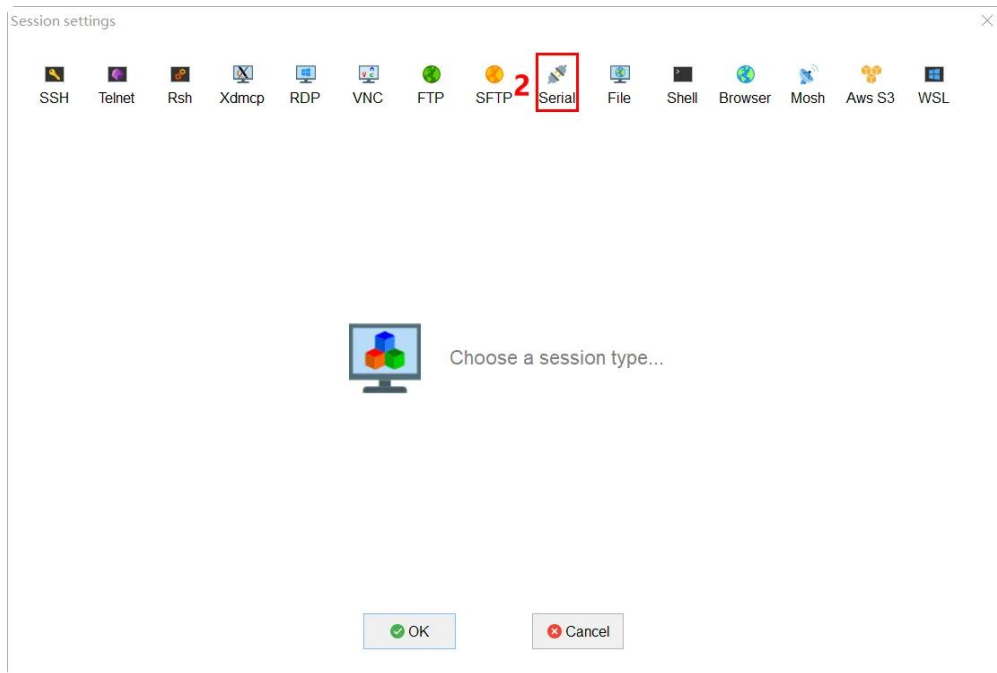
链接：<https://pan.baidu.com/s/1EY5Dces19B3c2oblq0rlfA?pwd=1234>

提取码：1234

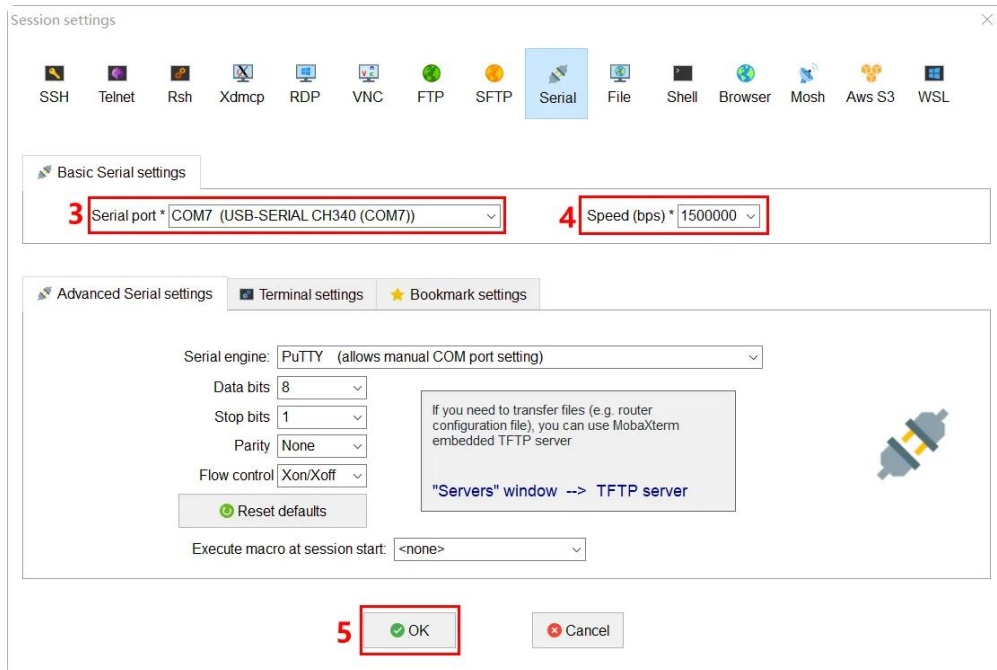
☐		mk-image.sh	2023-04-11 20:38	sh文件	950B
☐		SDDiskTool_v1.69.zip	2023-07-14 16:25	zip文件	467KB
☐		ido-pack-tools.tar.gz	2023-04-11 20:38	gz文件	1.89MB
☐		RKDevTool_Release_v2.95.zip	2023-04-11 20:38	zip文件	2.30MB
☐		RKDevInfoWriteTool_Setup_V1.1.4_210527.7z	2023-04-11 20:38	7z文件	3.50MB
☐		DriverAssitant_v5.11.zip	2023-04-11 20:38	zip文件	9.36MB
☐		MobaXterm_Portable_v23.6.zip	2024-05-22 16:55	zip文件	39.99MB
☐		RKTools-for-mac.zip	2023-06-21 14:07	zip文件	58.94MB

1

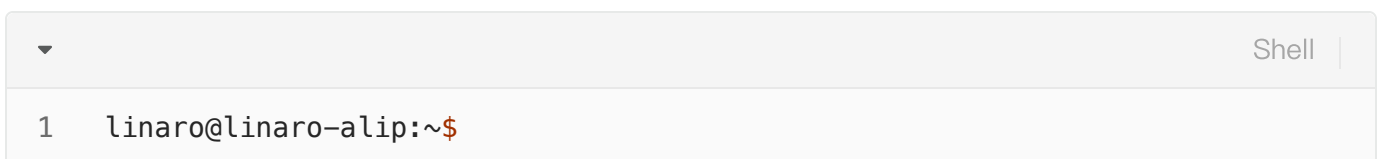
2. 选择session为【Serial】，如下图所示：



3. 将Serial port修改为在设备管理器中找到的COM端口
4. 设置Speed(bps)为1500000
5. 点击【OK】按钮，如下图所示：

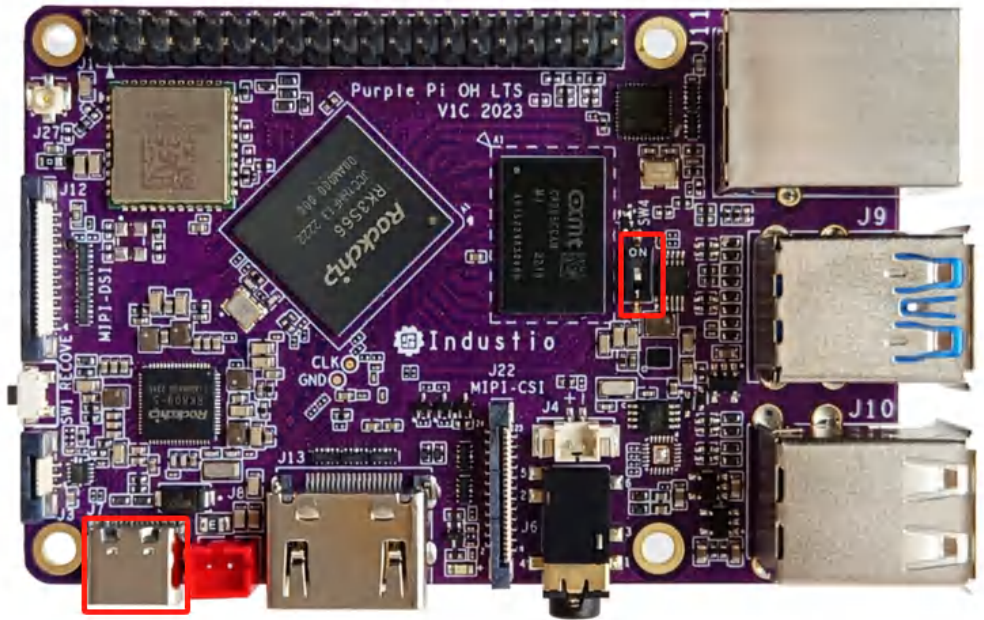


默认登录账号密码为 `linaro @ linaro`



1.2 ADB调试

ADB调试端口位于（J7），将SW4切换到【1】档，使用USB-C，连接主板和电脑即可在电脑上使用ADB调试，如下图所示：



ADB调试界面如下图所示：

```
C:\Users\ronnie>adb shell
* daemon not running. starting it now on port 5037 *
* daemon started successfully *
root@linaro-alip:/# ls
ls
bin  dev  lib      media  opt   run    srv    tmp    usr
boot etc  lost+found mnt    proc  sbin   sys    udisk  var
data home md5sum.txt oem    root  sdcard system userdata vendor
root@linaro-alip:/#
```

1.3 ssh调试

系统默认ssh调试，默认登陆账号密码为：linaro @ linaro。

ssh调试，如下图所示：

```
linaro@linaro-alip: ~  
login as: linaro  
linaro@192.168.1.123's password:  
Linux linaro-alip 4.19.219 #57 SMP Fri Nov 4 11:14:32 CST 2022 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.  
Last login: Fri Nov 4 07:00:47 2022  
linaro@linaro-alip:~$ ls  
Desktop  
linaro@linaro-alip:~$
```

2 双排针功能及接口

主板预留了40Pin的双排针接口，包含一路UART、一路I2C、一路SPI及19个GPIO，如下图所示：



2.1 双排针Pin功能定义

双排针Pin功能定义见下表：

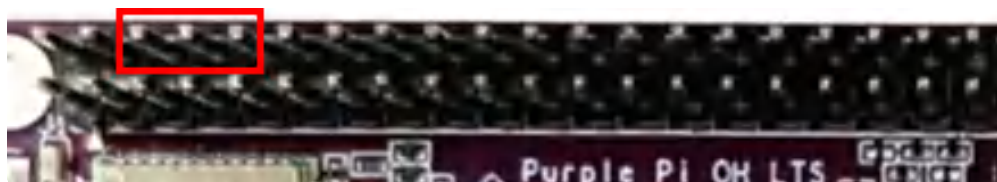
序号	定义	电平	说明
1	VDD_3V3	3.3V	供电输出3.3V
2	VDD_5V	5V	可做5V输入供电，也可输出5V
3	I2C_SDA	3.3V	I2C数据信号
4	VDD_5V	5V	可做5V输入供电，也可输出5V
5	I2C_SCL	3.3V	I2C时钟信号
6	GND	GND	电源地
7	GPIO_7	3.3V	gpio23
8	UART_TXD	3.3V	/dev/ttyS0 发送

9	GND	GND	电源地
10	UART_RXD	3.3V	/dev/ttyS0 接收
11	GPIO_0	3.3V	gpio15
12	GPIO_1	3.3V	gpio22
13	GPIO_2	3.3V	gpio20
14	GND	GND	电源地
15	GPIO_3	3.3V	gpio21
16	GPIO_4	3.3V	gpio124
17	VDD_3V3	3.3V	供电输出3.3V
18	GPIO_5	3.3V	gpio125
19	SPI_MOSI	3.3V	SPI数据信号
20	GND	GND	电源地
21	SPI_MISO	3.3V	电源地
22	GPIO_6	3.3V	gpio130
23	SPI_SCLK	3.3V	SPI时钟信号
24	SPI_CS0	3.3V	SPI片选信号0
25	GND	GND	电源地
26	SPI_CS1	3.3V	SPI片选信号1
27	GPIO_30	3.3V	gpio138
28	GPIO_31	3.3V	gpio139
29	GPIO_21	3.3V	gpio132
30	GND	GND	电源地
31	GPIO_22	3.3V	gpio121

32	GPIO_26	3.3V	gpio120
33	GPIO_23	3.3V	gpio122
34	GND	GND	电源地
35	GPIO_24	3.3V	gpio127
36	GPIO_27	3.3V	gpio123
37	GPIO_25	3.3V	gpio133
38	GPIO_28	3.3V	gpio126
39	GND	GND	电源地
40	GPIO_29	3.3V	gpio131

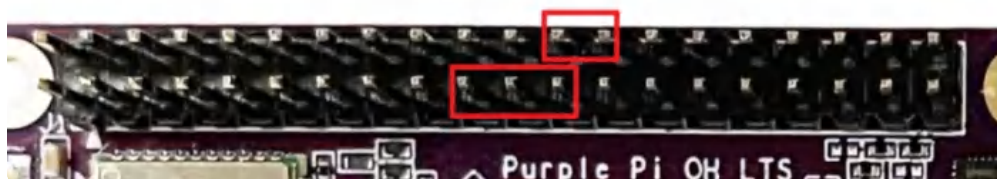
2.2 UART

设备节点为 /dev/ttyS0，串口接口位置及引脚定义，如下图所示：



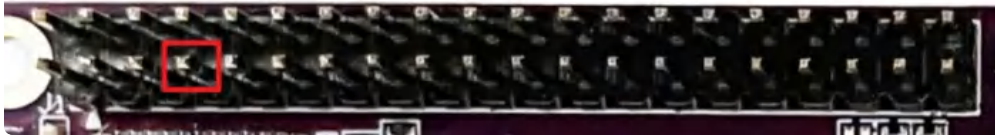
2.3 SPI

设备节点：/dev/spidev3.0，SPI接口位置如下图所示：



2.4 GPIO

GPIO引脚如下图所示：



以Pin7的GPIO_7为例，对应系统的GPIO number为23，可通过sysctl的方式控制此路GPIO的方向和电平，设置为输出，命令如下：

```
1 #导出gpio23
2 echo 23 > /sys/class/gpio/export
3 # 设置GPIO方向为输出
4 echo out > /sys/class/gpio/gpio23/direction
5 # 设置输出高电平
6 echo 1 > /sys/class/gpio/gpio23/value
7 # 设置输出低电平
8 echo 0 > /sys/class/gpio/gpio23/value
```

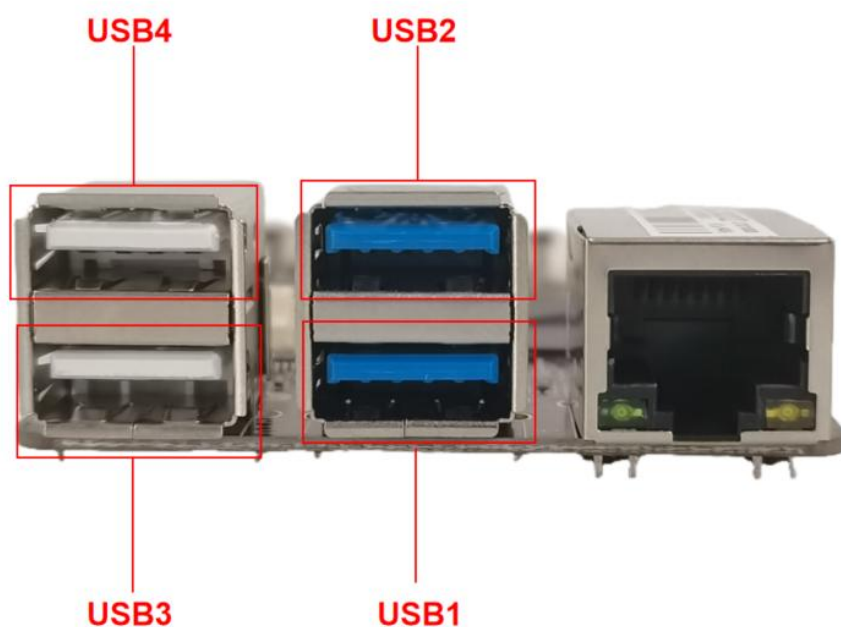
输出高电平时，使用万用表可测量到的电压值为3.3V；输出低电平时，使用万用表可测量到的电压值为0V。设置为输入，命令如下：

```
1 #导出gpio23，如果已经导出过可以不用此操作
2 echo 23 > /sys/class/gpio/export
3 # 设置GPIO方向为输入
4 echo in > /sys/class/gpio/gpio23/direction
5 # 读取GPIO接口电平
6 cat /sys/class/gpio/gpio23/value
```

GPIO连接3.3V的电平，cat到的数值为1；将此GPIO连接到GND，cat将读取到结果为0。

3 USB

主板共配置4路USB接口，分别记作USB1-4。如下图所示：



编号	类型	位置
usb1	host-2.0	J9 (下)
usb2	host-3.0	J9 (上)
usb3	host-2.0	J10 (下)
usb4	host-2.0	J10 (上)

注意：如果要使用USB1，需要将SW4切换至 **【ON】** 档，且ADB功能无法使用。

当USB1-4插入U盘后，会自动挂载/media/linaro/目录下，命令如下：

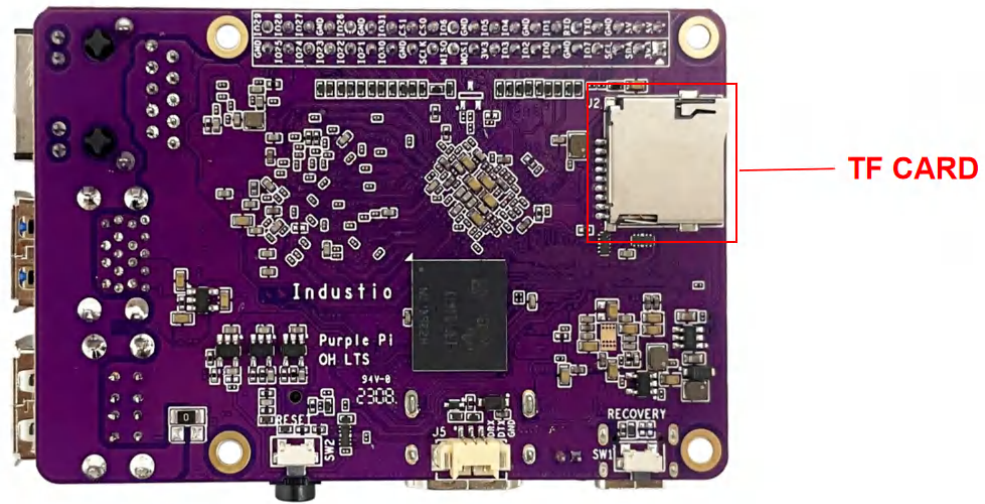
```

1  linaro@linaro-alip:mount
2  /dev/sda1 on /media/linaro/51AE-DF5C type vfat (rw,nosuid,nodev,relatime,uid=1001,gid=1001,mask=0022,dmask=0022,codepage=936,iocharset=utf8,shortname=mixed,showexec=utf8,flush,errors=remount-ro,uhelper=udisks2)

```

4 TF CARD

主板配置了一个TF 卡接口，位于（J2），当TF CARD接口插入TF卡后，会自动挂载到/media/linaro/目录下。



```

1  root@linaro-alip:/# fdisk -l
2  Disk /dev/ram0: 4 MiB, 4194304 bytes, 8192 sectors
3  Units: sectors of 1 * 512 = 512 bytes
4  Sector size (logical/physical): 512 bytes / 4096 bytes
5  I/O size (minimum/optimal): 4096 bytes / 4096 bytes
6
7
8  Disk /dev/mmcblk0: 29.12 GiB, 31268536320 bytes, 61071360 sectors
9  Units: sectors of 1 * 512 = 512 bytes
10 Sector size (logical/physical): 512 bytes / 512 bytes
11 I/O size (minimum/optimal): 512 bytes / 512 bytes
12 Disklabel type: gpt
13 Disk identifier: 1B430000-0000-4C26-8000-284C000003AE
14
15 Device          Start      End  Sectors  Size Type
16 /dev/mmcblk0p1   16384     24575    8192     4M unknown
17 /dev/mmcblk0p2   24576     32767    8192     4M unknown
18 /dev/mmcblk0p3   32768    163839   131072    64M unknown
19 /dev/mmcblk0p4  163840    425983   262144   128M unknown
20 /dev/mmcblk0p5  425984    491519    65536    32M unknown
21 /dev/mmcblk0p6  491520    493567    2048     1M unknown
22 /dev/mmcblk0p7  493568   1542143  1048576   512M unknown
23 /dev/mmcblk0p8 1542144   1804287    262144   128M unknown
24 /dev/mmcblk0p9 1804288  61071326 59267039 28.3G unknown
25
26
27 Disk /dev/mmcblk1: 28.89 GiB, 31016878080 bytes, 60579840 sectors
28 Units: sectors of 1 * 512 = 512 bytes
29 Sector size (logical/physical): 512 bytes / 512 bytes
30 I/O size (minimum/optimal): 512 bytes / 512 bytes
31 Disklabel type: dos
32 Disk identifier: 0xef5a598e
33
34 Device          Boot Start      End  Sectors  Size Id Type
35 /dev/mmcblk1p1             16 60579839 60579824 28.9G  7 HPFS/NTFS/exFAT
36 root@linaro-alip:/#

```

挂载

```

1 root@linaro-alip:/# mkdir sdcard
2 root@linaro-alip:/# mount /dev/mmcblk1p1 /sdcard/
3 root@linaro-alip:/# df
4 文件系统          1K-块    已用    可用  已用%  挂载点
5 /dev/root         29077992 4168800 23664348 15% /
6 devtmpfs          988528    8    988520    1% /dev
7 tmpfs             1000528    0   1000528    0% /dev/shm
8 tmpfs             400212    1496   398716    1% /run
9 tmpfs             5120      4     5116     1% /run/lock
10 /dev/mmcblk0p8    126376    11952   110888   10% /oem
11 /dev/mmcblk0p7    523672    536   512432    1% /userdata
12 tmpfs            200104    40   200064    1% /run/user/1000
13 /dev/mmcblk1p1   30289908 64692 30225216    1% /sdcard
14 root@linaro-alip:/#

```

5 Ethernet

主板配置了一个1000M以太网接口，位于（J11），在系统中其对应的网络设备节点为eth0。



5.1 查看以太网IP地址

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，命令如下：

```
1  linaro@linaro-alip:~$ sudo ifconfig eth0
2  eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
3      inet 192.168.1.133 netmask 255.255.255.0 broadcast 192.168.1.255
4      inet6 fe80::3b43:b691:ded5:c497 prefixlen 64 scopeid 0x20<link>
5      ether 82:4c:21:62:f5:35 txqueuelen 1000 (Ethernet)
6      RX packets 29 bytes 4592 (4.4 KiB)
7      RX errors 0 dropped 0 overruns 0 frame 0
8      TX packets 43 bytes 4146 (4.0 KiB)
9      TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
10     device interrupt 40
11
12  linaro@linaro-alip:~$
```

5.2 设置以太网临时IP地址

设置以太网临时IP地址，命令如下：

```
1  linaro@linaro-alip:~$ sudo ifconfig eth0 192.168.1.123
```

5.3 设置以太网永久静态IP

修改配置文件：/etc/network/interfaces，设置静态IP 192.168.1.123为例，命令如下：

```
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  auto eth0
5      iface eth0 inet static
6      address 192.168.1.123
7      netmask 255.255.255.0
8      gateway 192.168.1.1
9      nameserver 192.168.1.1
```

修改配置文件后重启网络，命令如下：

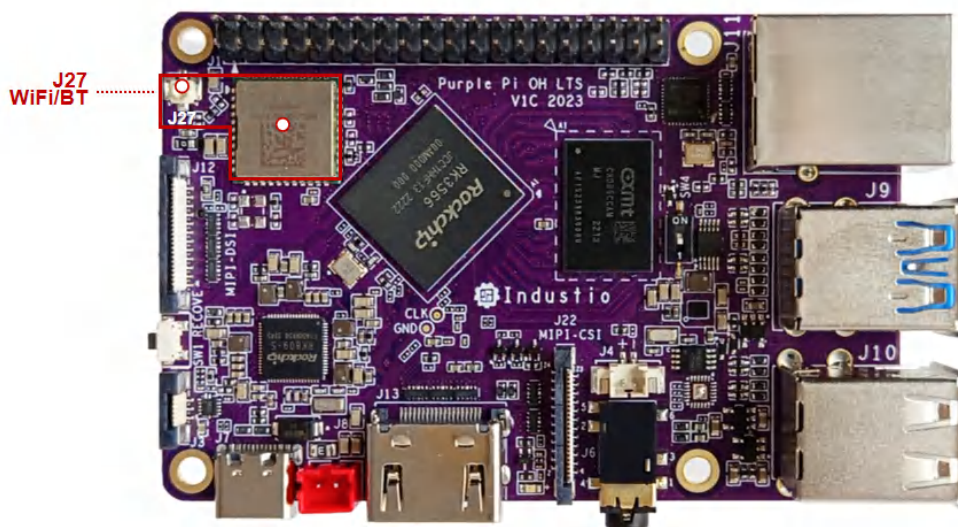
```
1 sudo systemctl restart networking
```

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

低配主板配置一个 2.4G WiFi模块，高配主板配置了 2.4G/5G 双频WiFi模块，如下图所示：



系统上电默认会打开WiFi，对应的网络设备节点为wlan0，命令如下：

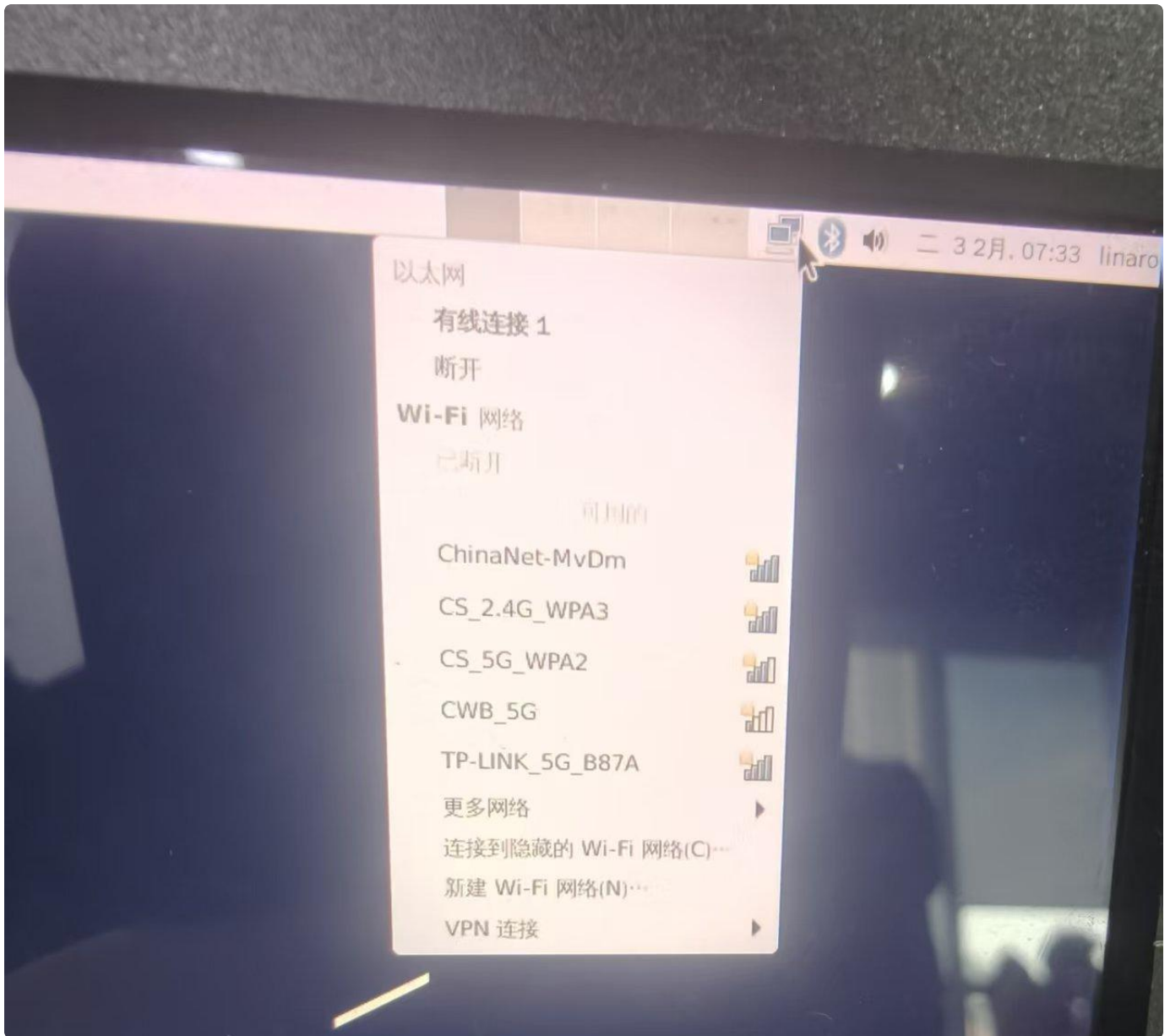
```
1 linaro@linaro-alip:~$ sudo ifconfig wlan0
2 wlan0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
3         ether 2c:d2:6b:10:ea:4d  txqueuelen 1000  (Ethernet)
4         RX packets 0  bytes 0 (0.0 B)
5         RX errors 0  dropped 0  overruns 0  frame 0
6         TX packets 0  bytes 0 (0.0 B)
7         TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

6.1连接热点

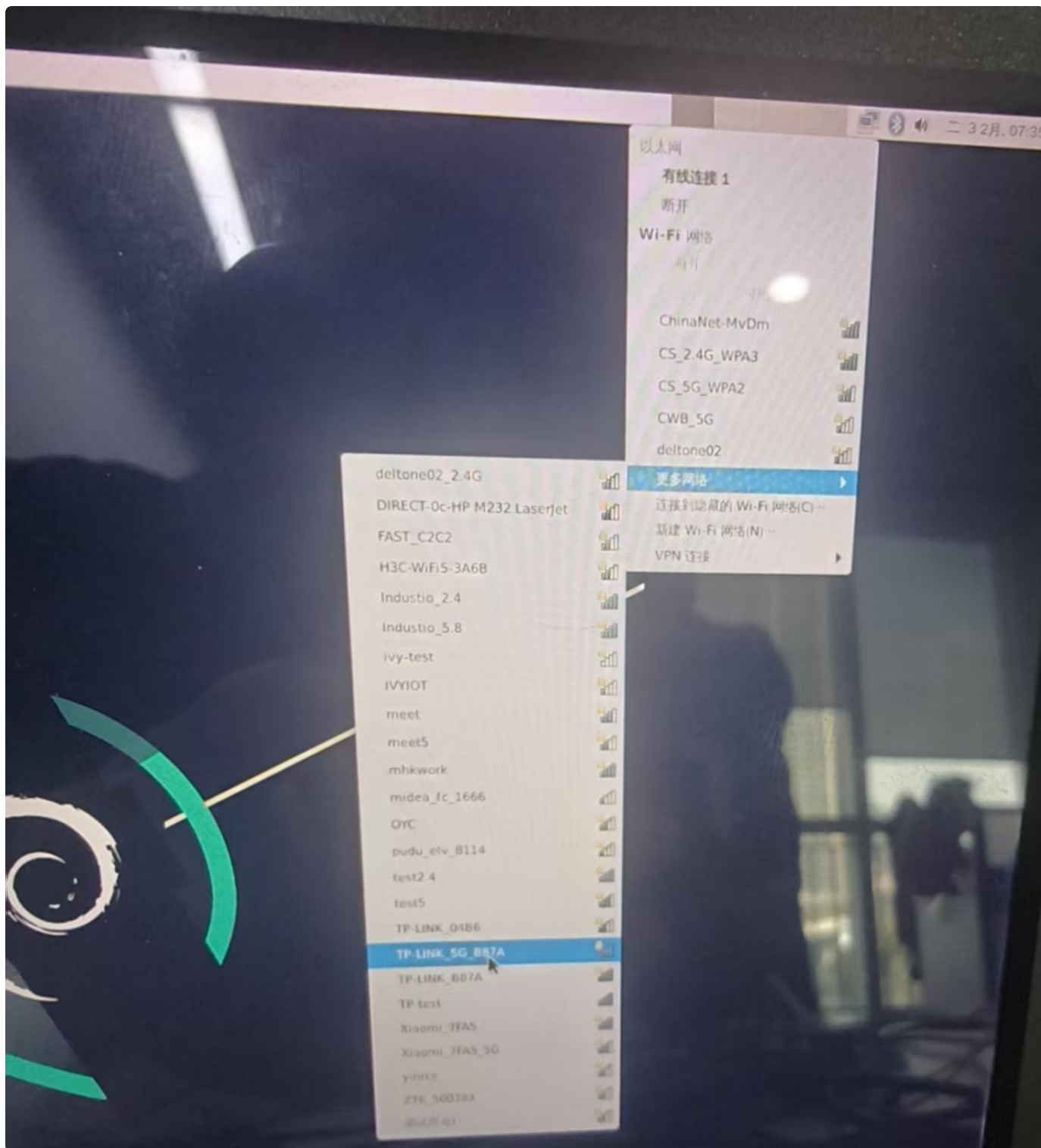
连接热点可以在桌面上操作，也可以使用命令行操作。

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，如下图所示：



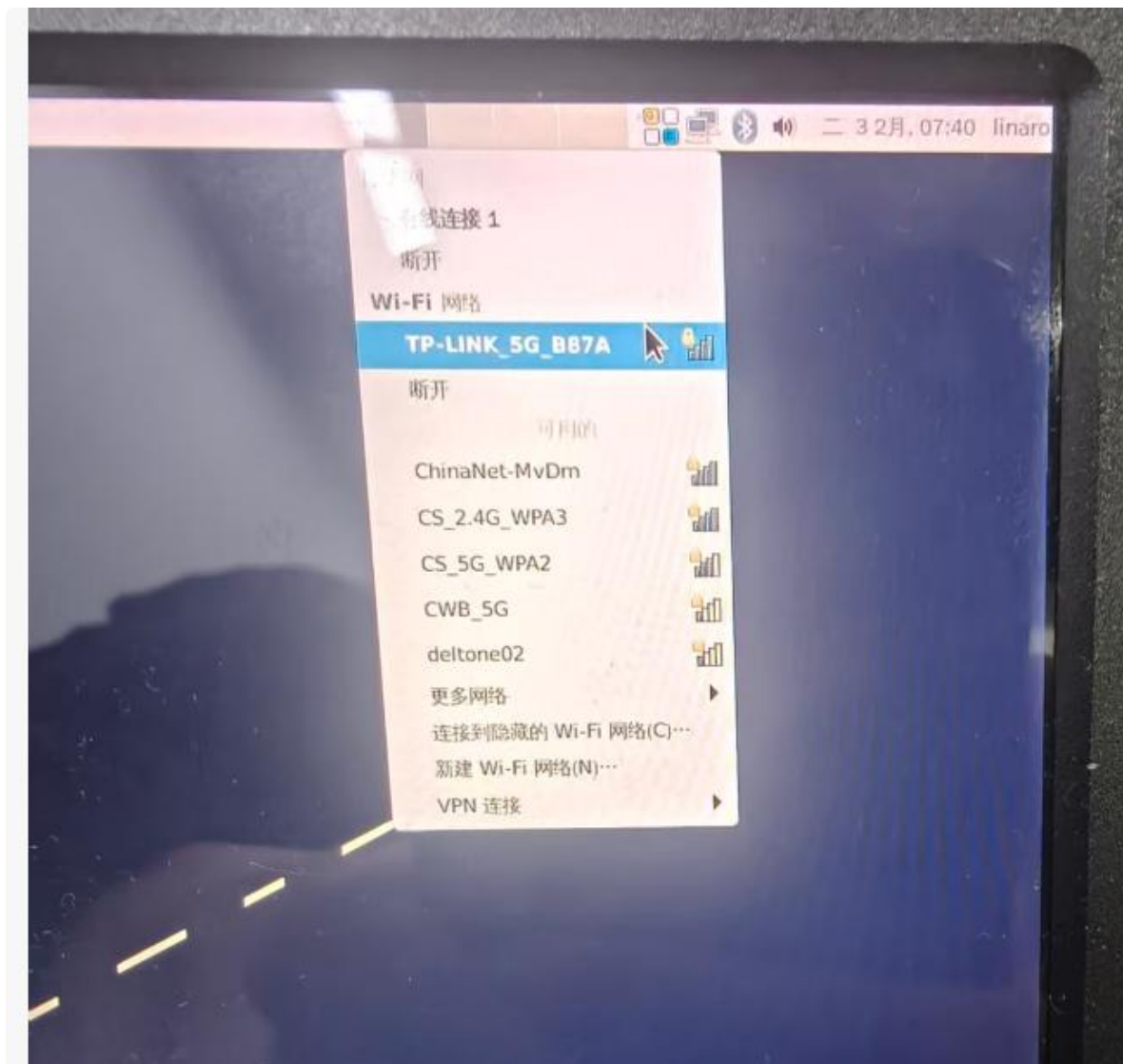
点击要连接的热点，弹出密码输入窗口，如下图所示：



使用键盘输入密码，确认



连接成功后，如下图所示：



6.1.2 方式二

```

1 # 扫描
2 root@linaro-alip:/# nmcli dev wifi list
3 IN-USE  BSSID                SSID                MODE  CHAN  RATE
4         22:A3:8E:D0:51:42  mhkwork            Infra  1     270 Mb
5 it/s >
6         44:F7:70:38:A7:19  CS_2.4G_WPA3       Infra  1     130 Mb
7 it/s >
8         4A:F7:70:38:A7:19  --                 Infra  1     130 Mb
9 it/s >
10        70:C6:DD:3B:2F:C4  Industio_2.4       Infra  11    130 Mb
11 it/s >
12        94:D9:B3:A6:B8:7A  TP-LINK_B87A       Infra  13    270 Mb
13 it/s >
14        94:D9:B3:A6:B8:7F  TP-LINK_5G_B87A    Infra  157   270 Mb
15 it/s >
16        78:60:5B:83:67:AE  test2.4            Infra  6     540 Mb
17 it/s >
18        7A:60:5B:13:67:AE  TP-test            Infra  6     540 Mb
19 it/s >
20        7A:60:5B:73:67:AE  --                 Infra  6     540 Mb
21 it/s >
22        B8:A1:4A:0A:F1:43  ChinaNet-MvDm      Infra  8     130 Mb
23 it/s >
24        70:C6:DD:3B:2F:C7  Industio_5.8       Infra  153   270 Mb
25 it/s >
26        24:CF:24:50:7F:A6  Xiaomi_7FA5        Infra  2     130 Mb
27 it/s >
28        22:A3:8E:D0:51:46  mhkwork            Infra  44    270 Mb
29 it/s >
30        F6:FC:49:36:BC:1C  --                 Infra  7     130 Mb
31 it/s >
32        44:F7:70:38:A7:18  CS_5G_WPA2         Infra  40    270 Mb
33 it/s >
34        F4:FC:49:06:BC:1C  meet               Infra  7     130 Mb
35 it/s >
36        4E:F7:70:38:A7:18  --                 Infra  40    270 Mb
37 it/s >
38        74:39:89:63:90:4C  deltone02_2.4G     Infra  1     270 Mb
39 it/s >
40        78:60:5B:83:67:B0  test5              Infra  48    540 Mb
41 it/s >
42        7A:60:5B:F3:67:AE  --                 Infra  48    540 Mb
43 it/s >
44

```

```
25          4E:49:68:D0:EA:12  --          Infra  6      270 Mb
    it/s >
26          24:CF:24:50:7F:A7  Xiaomi_7FA5_5G          Infra  157    270 Mb
    it/s >
27    lines 1-23
28
29    # 连接
```

6.1.3 方式三

使用命令 字母界面配置

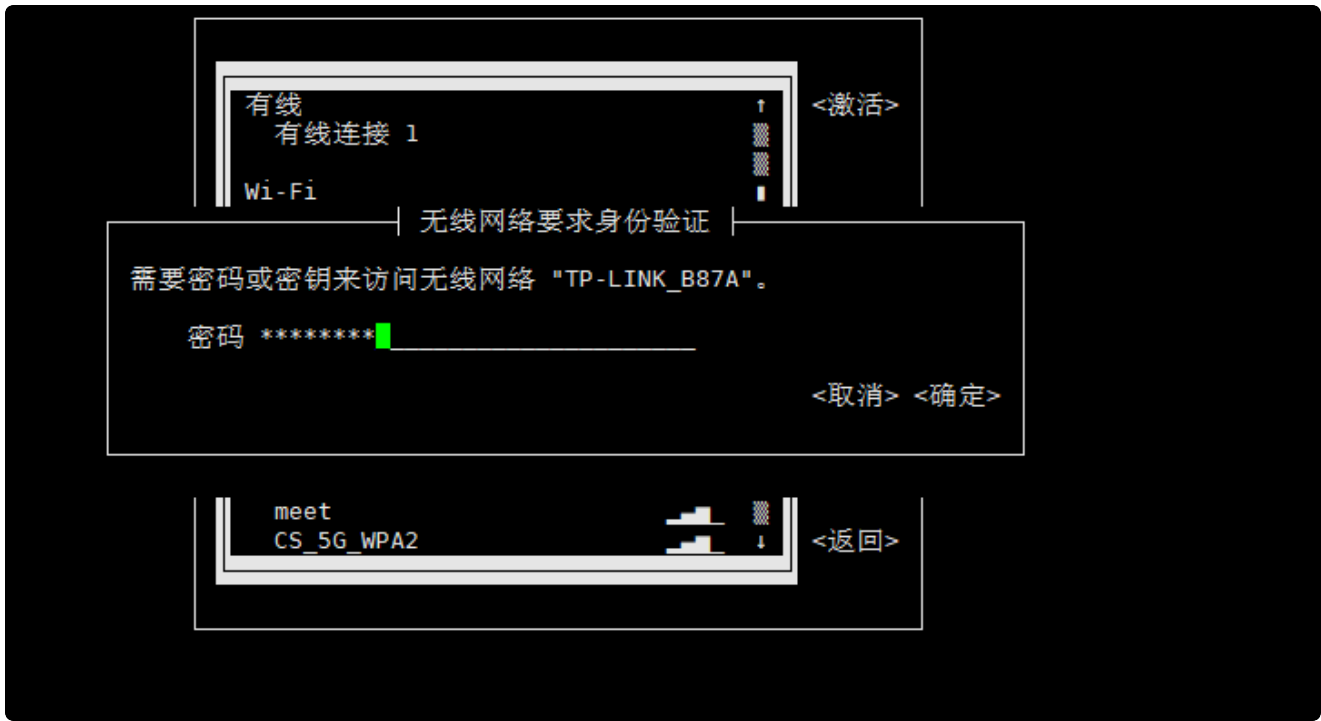
终端执行命令

```
▼ Bash |
1  nmtui
```

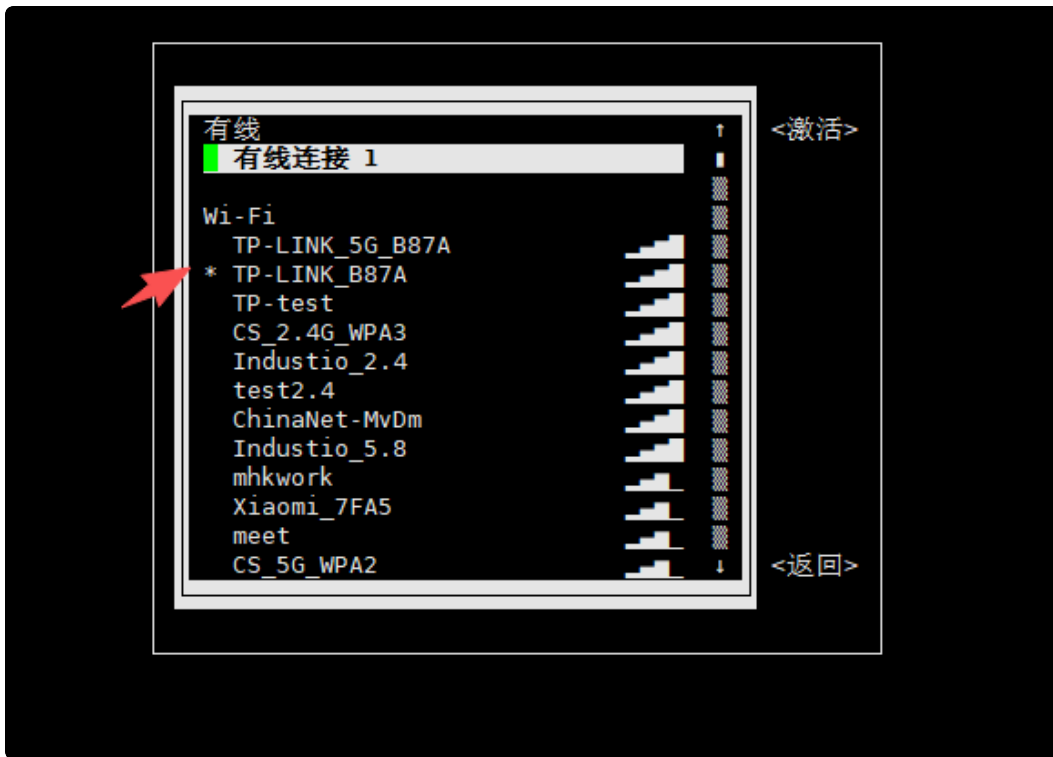
```
| 网络管理器文 |
|               |
| 请选择选项   |
| 编辑连接     |
| 启用连接     |
| 设置系统主机名 |
| 退出         |
|               |
| <确定>       |
```



输入密码



连接成功



连接成功后，键盘输入ESC 退出配置界面

7 Bluetooth

系统开机默认打开蓝牙，对应的网络节点为hci0，命令如下：

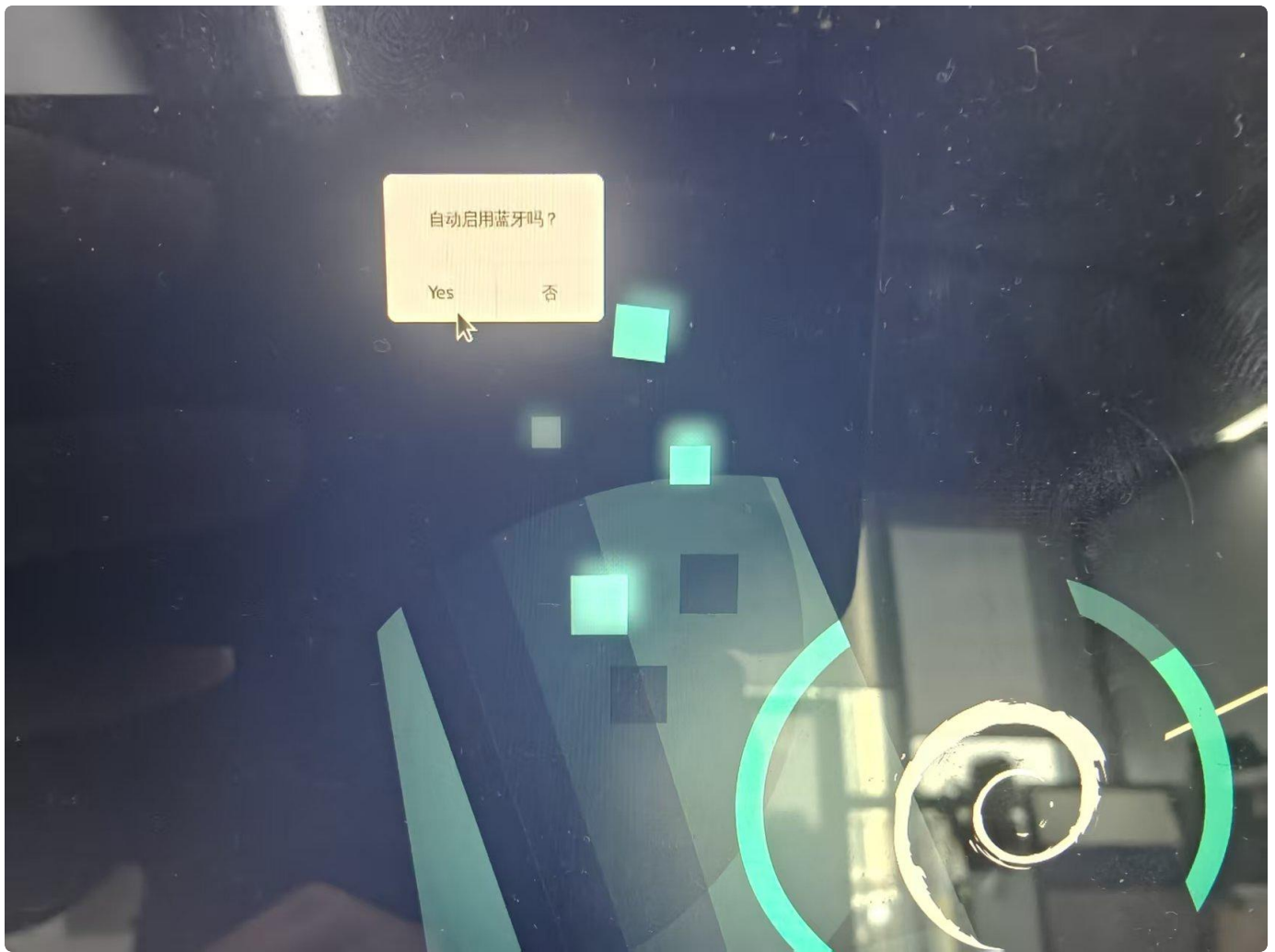
```

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:~$

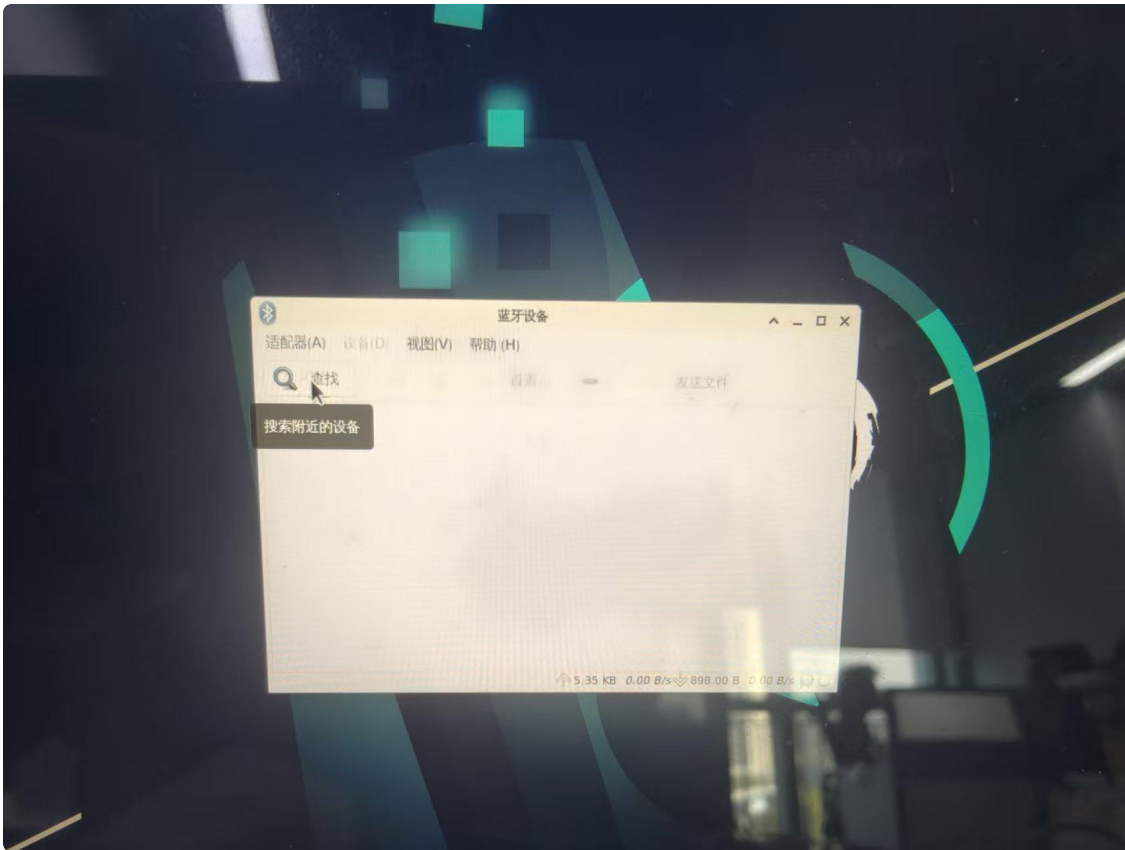
```

7.1 连接蓝牙设备

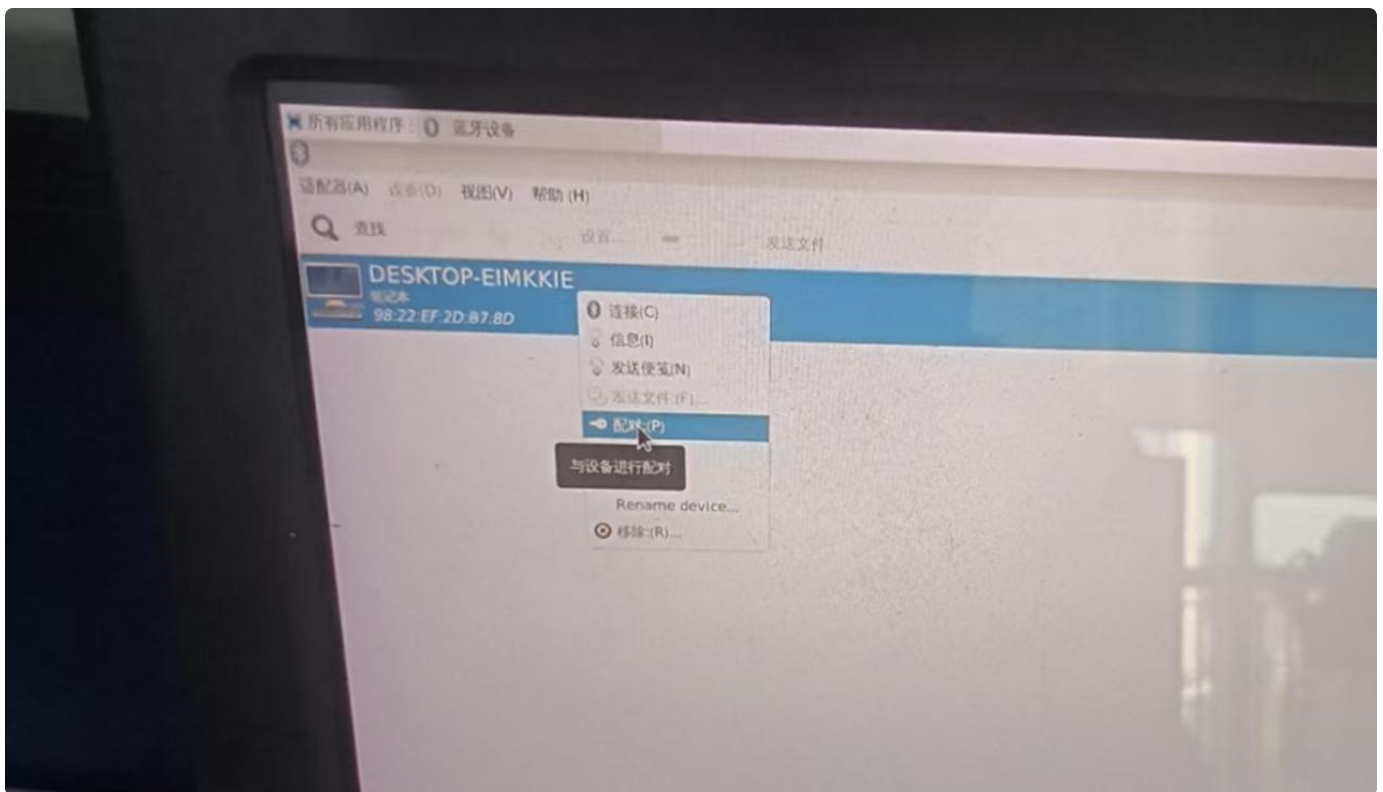
将鼠标放到桌面右上角蓝牙图标，【右键】->【设备】，如下图所示：



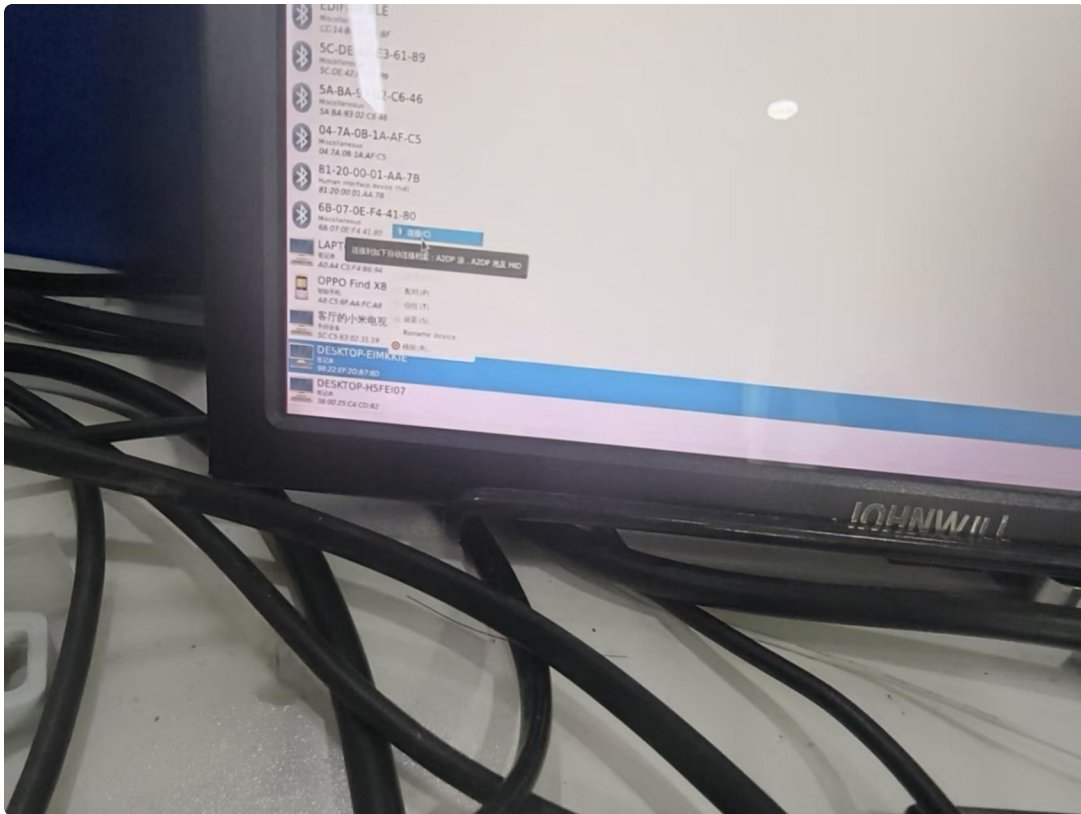
在弹出的窗口中，点击【查找】，如下图所示：



选中要连接设备，【右键】->【配对】，即可连接该设备，如下图所示：



选中要连接设备，【右键】->【配对】，即可连接该设备，如下图所示：



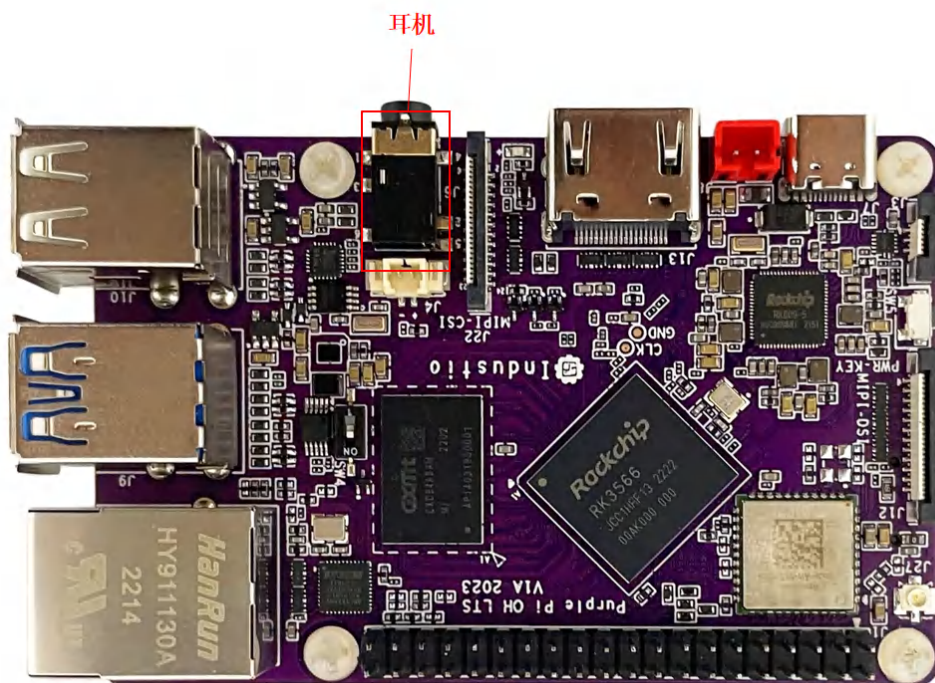
命令连接

Bash |

```
1  bluetoothctl
2  hci0 new_settings: powered bondable ssp br/edr le secure-conn
3  Agent registered
4  [CHG] Controller 4C:38:00:2B:C1:71 Pairable: yes
5  [bluetooth]#
6  #打开蓝牙
7  [bluetooth]# power on
8  #扫描蓝牙
9  [bluetooth]# scan on
10 # # 开启广播
11 [bluetooth]# discoverable on
12 # 允许配对
13 [bluetooth]# pairable on
14 #查看蓝牙设备
15 [bluetooth]# devices
16 #信任蓝牙设备
17 [bluetooth]# trust 7C:C1:80:09:DD:6C
18 #蓝牙配对
19 [bluetooth]# pair 7C:C1:80:09:DD:6C
20 #连接蓝牙
21 [bluetooth]# connect 7C:C1:80:09:DD:6C
```

8 音频

主板配置了一路耳机接口，位于（J6），具备进行音频输出及录音功能，如下图所示：



8.1 查看声卡设备

查看声卡设备，命令如下：

```
1  $ aplay -l
2  **** List of PLAYBACK Hardware Devices ****
3  card 0: Dummy [Dummy], device 0: Dummy PCM [Dummy PCM]
4      Subdevices: 7/8
5      Subdevice #0: subdevice #0
6      Subdevice #1: subdevice #1
7      Subdevice #2: subdevice #2
8      Subdevice #3: subdevice #3
9      Subdevice #4: subdevice #4
10     Subdevice #5: subdevice #5
11     Subdevice #6: subdevice #6
12     Subdevice #7: subdevice #7
13  card 1: Loopback [Loopback], device 0: Loopback PCM [Loopback PCM]
14     Subdevices: 7/8
15     Subdevice #0: subdevice #0
16     Subdevice #1: subdevice #1
17     Subdevice #2: subdevice #2
18     Subdevice #3: subdevice #3
19     Subdevice #4: subdevice #4
20     Subdevice #5: subdevice #5
21     Subdevice #6: subdevice #6
22     Subdevice #7: subdevice #7
23  card 1: Loopback [Loopback], device 1: Loopback PCM [Loopback PCM]
24     Subdevices: 8/8
25     Subdevice #0: subdevice #0
26     Subdevice #1: subdevice #1
27     Subdevice #2: subdevice #2
28     Subdevice #3: subdevice #3
29     Subdevice #4: subdevice #4
30     Subdevice #5: subdevice #5
31     Subdevice #6: subdevice #6
32     Subdevice #7: subdevice #7
33  card 2: rockchiphdmi [rockchip,hdmi], device 0: rockchip,hdmi i2s-hifi-0
    [rockchip,hdmi i2s-hifi-0]
34     Subdevices: 0/1
35     Subdevice #0: subdevice #0
36  card 3: rockchiprk809co [rockchip,rk809-codec], device 0: fe410000.i2s-rk8
    17-hifi rk817-hifi-0 [fe410000.i2s-rk817-hifi rk817-hifi-0]
37     Subdevices: 0/1
38     Subdevice #0: subdevice #0
39
```

注意：MIPI固件仅有rk809-codec一个声卡。

8.2 播放音频

播放到HDMI，命令如下：

▼

Bash

```
1 aplay -D hw:2,0 test.wav
```

插入耳机，执行命令如下：

▼

Bash

```
1 //HDMI固件
2 aplay -D plughw:3,0 /usr/share/sounds/alsa/Rear_Center.wav
3
4 //MIPI固件
5 aplay -D plughw:2,0 /usr/share/sounds/alsa/Rear_Center.wav
```

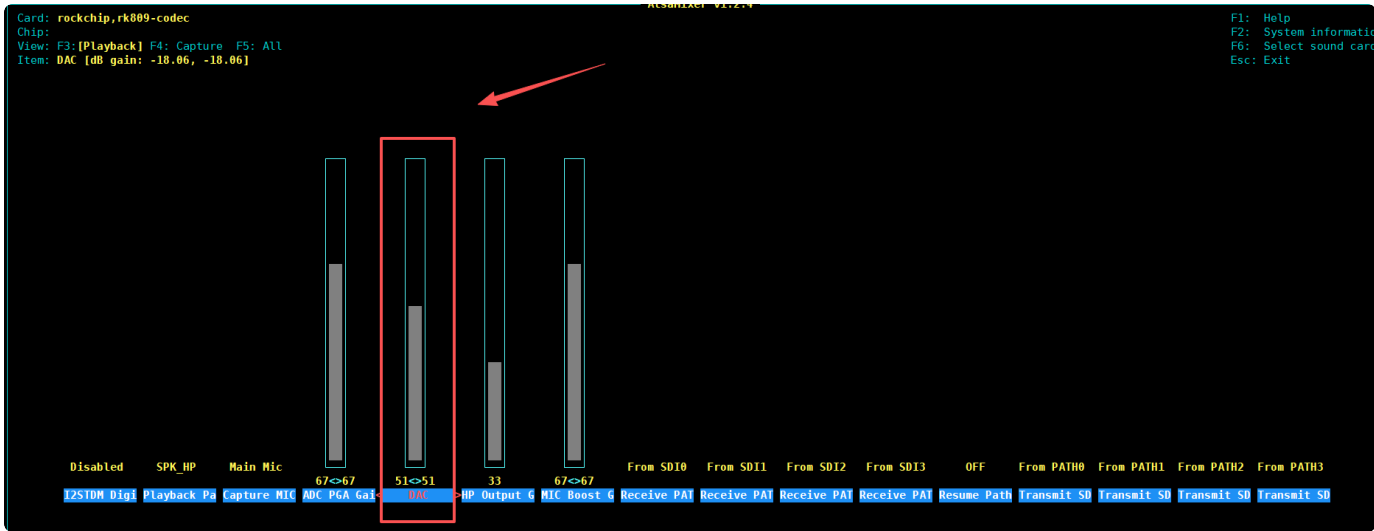
8.3 音量的调节

打开声卡字符配置界面，左右(←, →)选择至DAC，上下调节音量(↑, ↓)

▼

Bash

```
1 # hdmi 屏幕执行
2 alsamixer -c 3
3 # mipi 屏幕执行
4 alsamixer -c 2
```



8.4 录音

需要接入带麦克风的耳机。使用arecord工具可以进行录音测试(0.0才是HDMI, 1.0是MIPI), 命令如下:

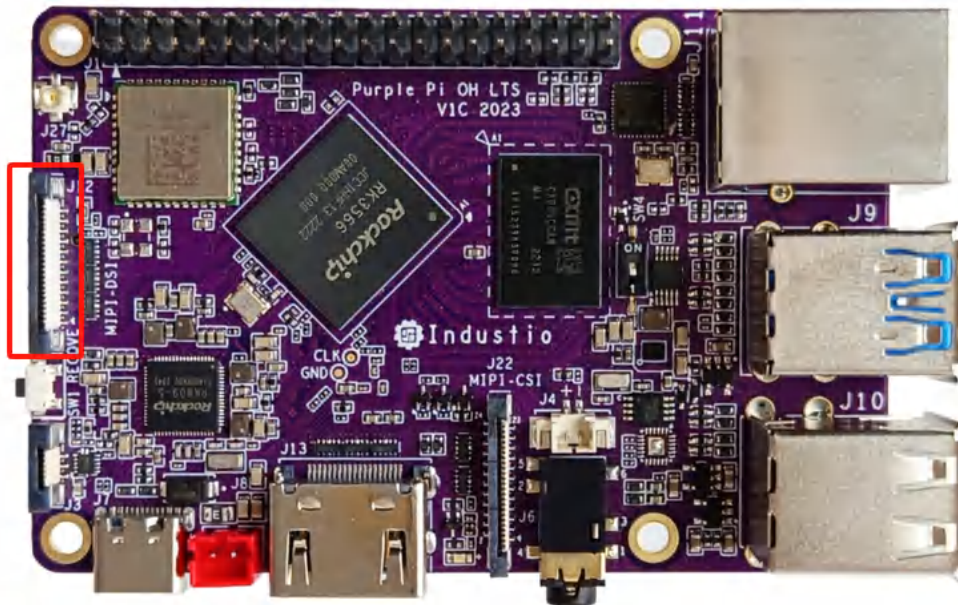
```
▼ Bash |
1 //HDMI固件
2 $ arecord -D hw:3,0 -r 48000 -c 2 -f S16_LE test.wav
3 Recording WAVE 'test.wav' : Signed 16 bit Little Endian, Rate 48000 Hz, Stereo
4
5 ^CAborted by signal 中断...
6 $
7
8 //MIPI固件
9 $ arecord -D hw:2,0 -r 48000 -c 2 -f S16_LE test.wav
10 Recording WAVE 'test.wav' : Signed 16 bit Little Endian, Rate 48000 Hz, Stereo
11
12 ^CAborted by signal 中断...
13 ~$
```

录音完后播放测试, 命令如下:

```
▼ Bash |
1 //HDMI固件
2 linaro@linaro-alip:~$ aplay -D hw:3,0 ./test.wav
3 Playing WAVE './test.wav' : Signed 16 bit Little Endian, Rate 48000 Hz, Stereo
4 //MIPI固件
5 linaro@linaro-alip:~$ aplay -D hw:2,0 ./test.wav
6 Playing WAVE './test.wav' : Signed 16 bit Little Endian, Rate 48000 Hz, Stereo
```

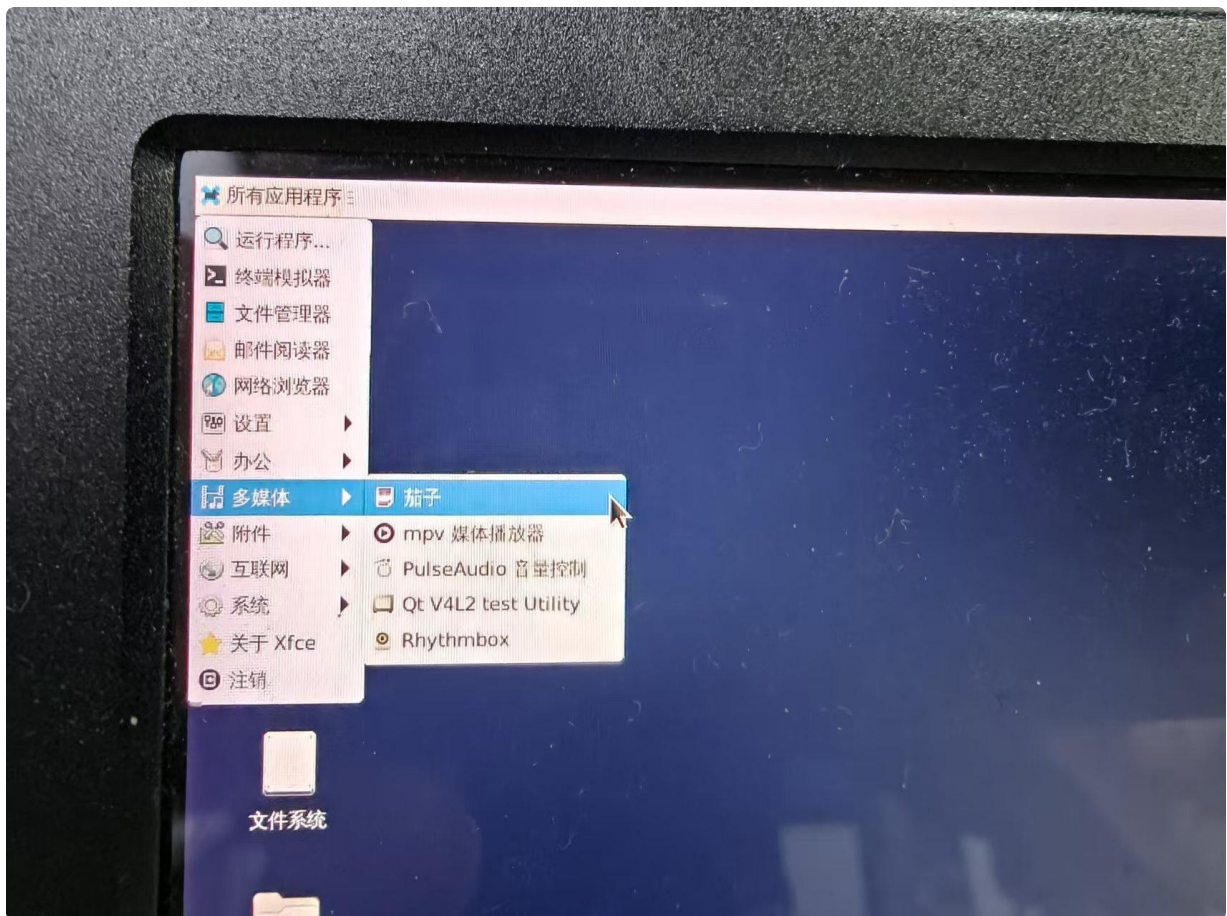
9 Camera

主板默认适配OV5648和OV8858摄像头, 摄像头接口位于(J22), 如下图所示:

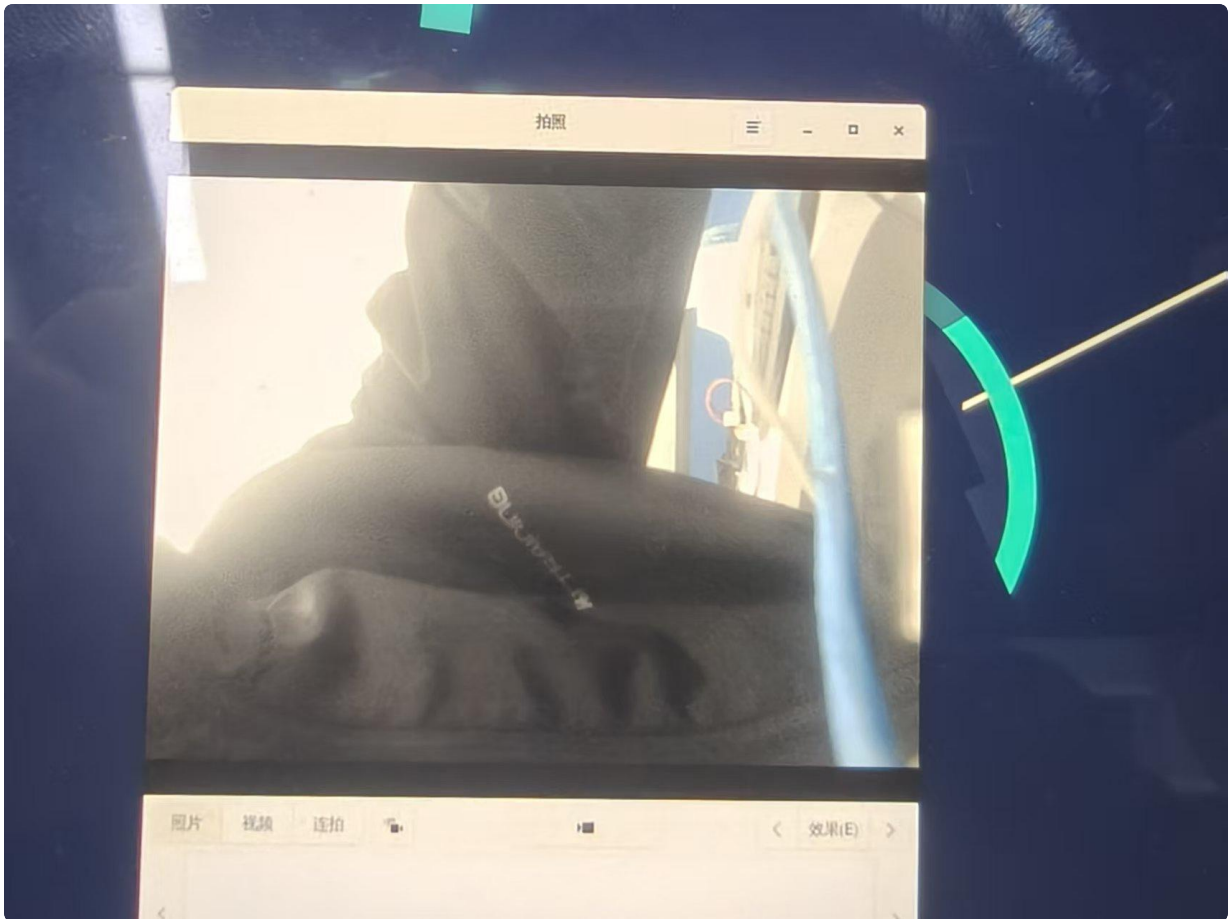


9.1 测试

使用Debian自带的【茄子】程序，可以测试摄像头功能。依次点击桌面左下角的【影音】->【茄子】，如下图所示：



测试摄像头功能如下图所示：



10 RTC

主板包含2个RTC，其中/dev/rtc0为外部RTC（HYM8563），/dev/rtc1为CPU内部的RTC（RK808）。系统默认使用rtc0的时间。

10.1 获取RTC时间

获取RTC时间，命令如下：

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

10.2 设置RTC时间

设置RTC时间，命令如下：

```

1  linaro@linaro-alip:~$ sudo date -s '2022-11-10 10:17:00'
2  2022年 11月 10日 星期四 10:17:00 UTC
3  linaro@linaro-alip:~$ sudo hwclock -w
4  linaro@linaro-alip:~$ sudo hwclock
5  2022-11-10 10:17:10.938039+00:00
6  linaro@linaro-alip:~$

```

10.3 RTC定时开机

CPU内部的rtc1支持支持定时开机功能。以下测试关机60秒定时开机，命令如下：

```

1  linaro@linaro-alip:~$ sudo chmod a+w /sys/class/rtc/rtc1/wakealarm
2  linaro@linaro-alip:~$ sudo echo +60 > /sys/class/rtc/rtc1/wakealarm
3  [ 150.298538] rk809:alarm set RTC date/time 2017-08-04(5) 09:03:38,enable:1
4  linaro@linaro-alip:~$ sudo poweroff
5  Stopping Setup zram based device zram0...
6  ...
7  [ 166.104018] mpp-iep2 fdef0000.iep: shutdown device
8  [ 166.104455] mpp_vepu2 fdee0000.vepu: shutdown device
9  [ 166.104912] mpp_vdpu2 fdea0400.vdpu: shutdown device
10 [ 166.105408] mpp_rkvenc fdf40000.rkvenc: shutdown device
11 [ 166.105875] mpp_rkvenc fdf40000.rkvenc: shutdown success
12 [ 166.107068] Local MAC: 82:4c:21:62:f5:35
13 [ 166.109611] eth0: local mac wol enable
14 [ 166.112687] set rk809 rtc alarm!
15 [ 166.116069] reboot: Power down

```

等待60秒后，主板会开机。

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

修改路径：/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，命令如下：

Bash

```
1  linaro@linaro-alip:~$ sudo chmod a+w /sys/class/backlight/backlight/brightness
2  linaro@linaro-alip:~$ sudo echo 100 > /sys/class/backlight/backlight/brightness
```

12.2 屏幕旋转

使用xrandr工具可以实现屏幕的旋转。

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，命令如下：

Bash

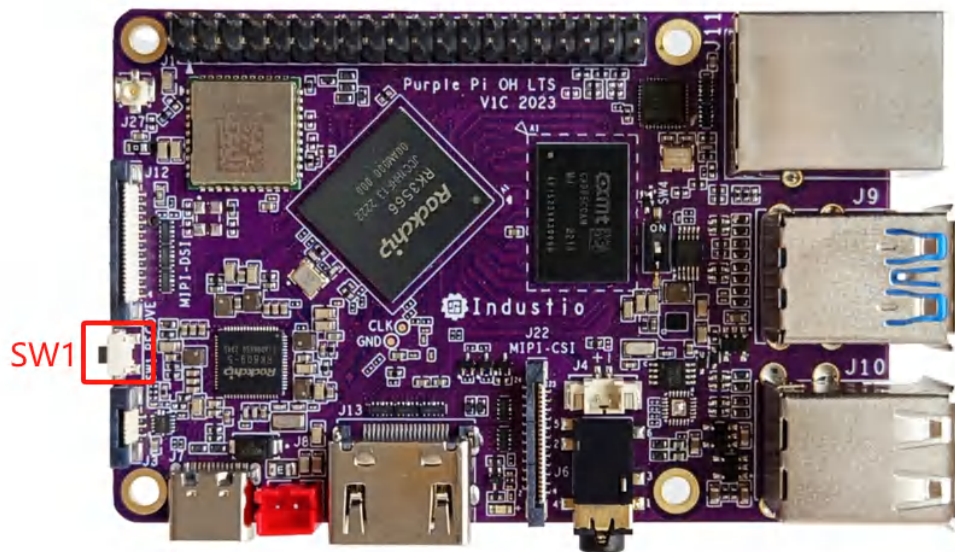
```
1  linaro@linaro-alip:~$ xrandr -o inverted
```

13 按键

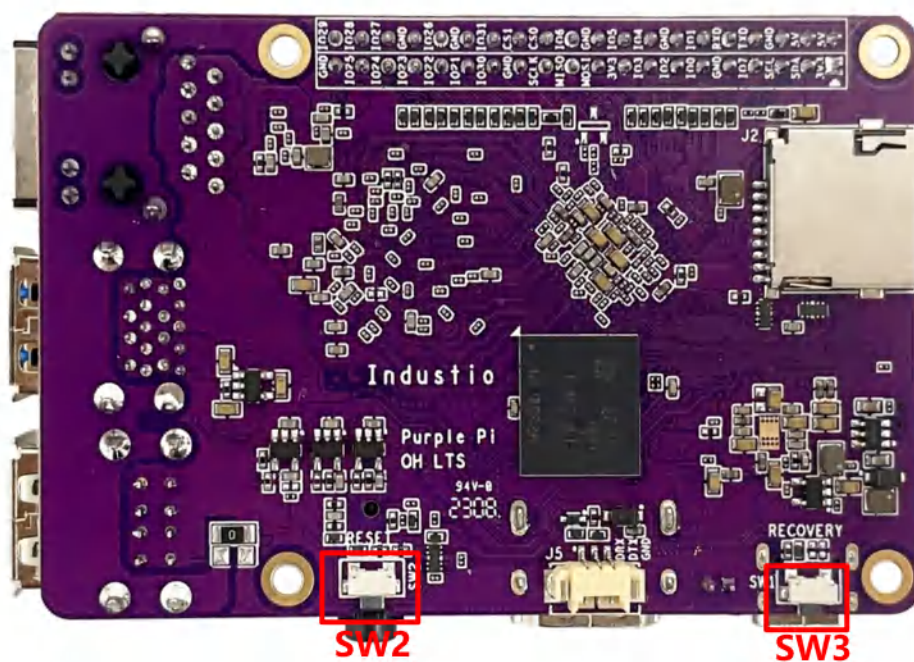
主板共配置3个按键，如下表所示：

标记	名称	功能
SW1	recovery	用于烧录，或上报KEY_VOLUMEUP
SW2	reset	硬件复位
SW3	power_on	电源键，用于开关机及待机唤醒，或上报KEY_POWER

SW1按键位置如下图所示：



SW2, SW3 按键位置如下图所示：

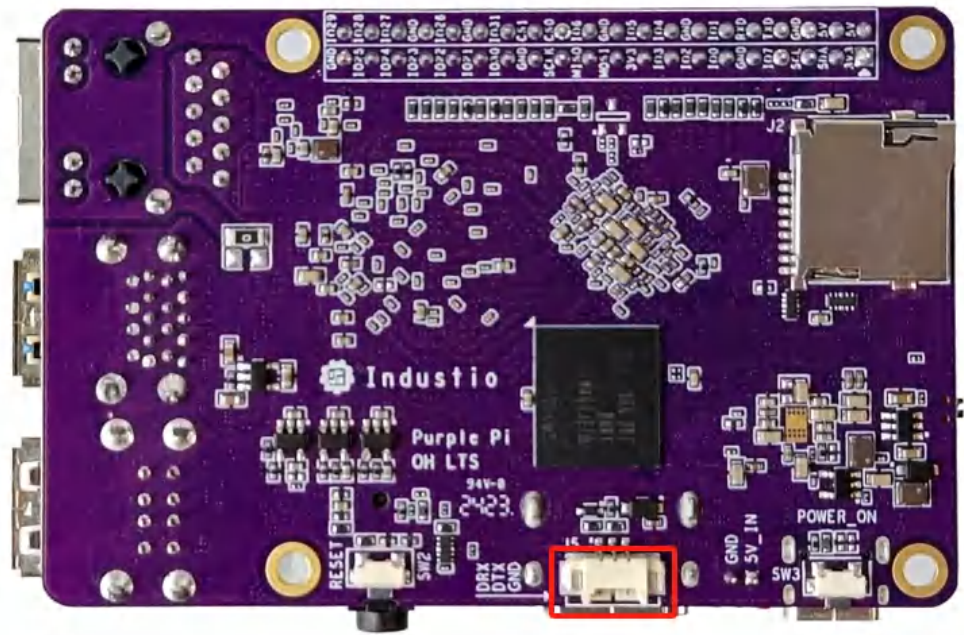


Purple Pi OH RK3566–Buildroot系统

1 调试

1.1 串口调试

主板调试串口位于USB座子下方（J28），建议使用配套的USB转串口工具。如下图所示：



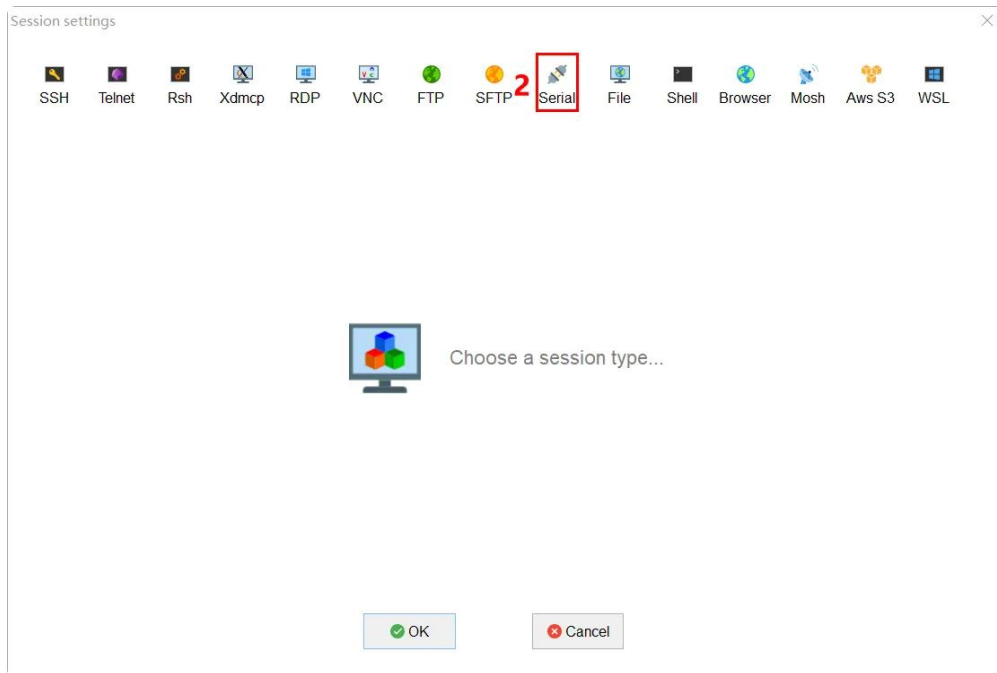
1. 打开MobaXterm, 下载链接如下:

链接: https://pan.baidu.com/s/11ui4LTd2mq_9kiJpeL4bWg?pwd=1234

提取码: 1234

☐ 文件名	↓ 修改时间	类型	大小
☐ RKDevTool_Release_v2.95.zip	2024-04-24 11:53	zip文件	2.30MB
☐ other_tools.txt	2024-04-25 15:31	txt文件	44B
☒ MobaXterm_Portable_v23.6.zip	2024-04-24 14:30	zip文件	39.99MB
☐ DriverAssitant_v5.11.zip	2024-04-24 11:52	zip文件	9.36MB

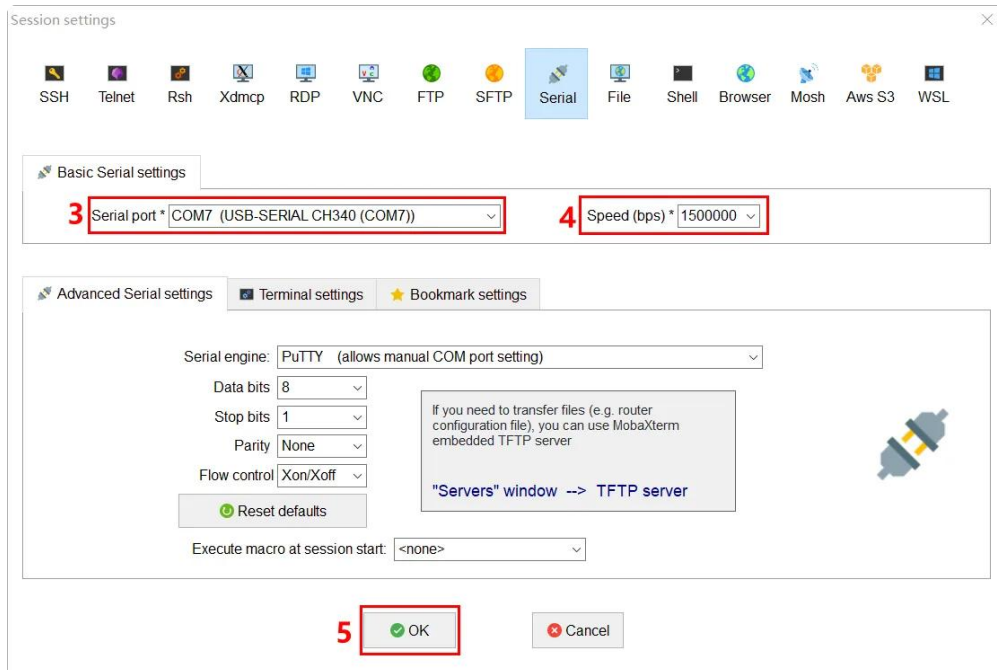
2. 选择sessionSerial, 如下图所示:



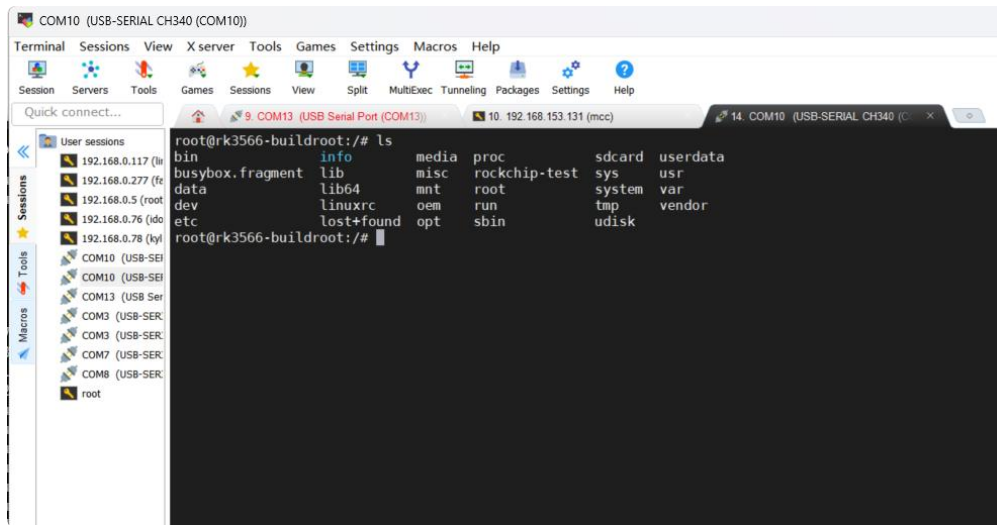
3. 将Serial port修改为在设备管理器中找到的COM端口

4. 设置Speed(bsp)为1500000

5. 点击【OK】按钮，如下图所示：

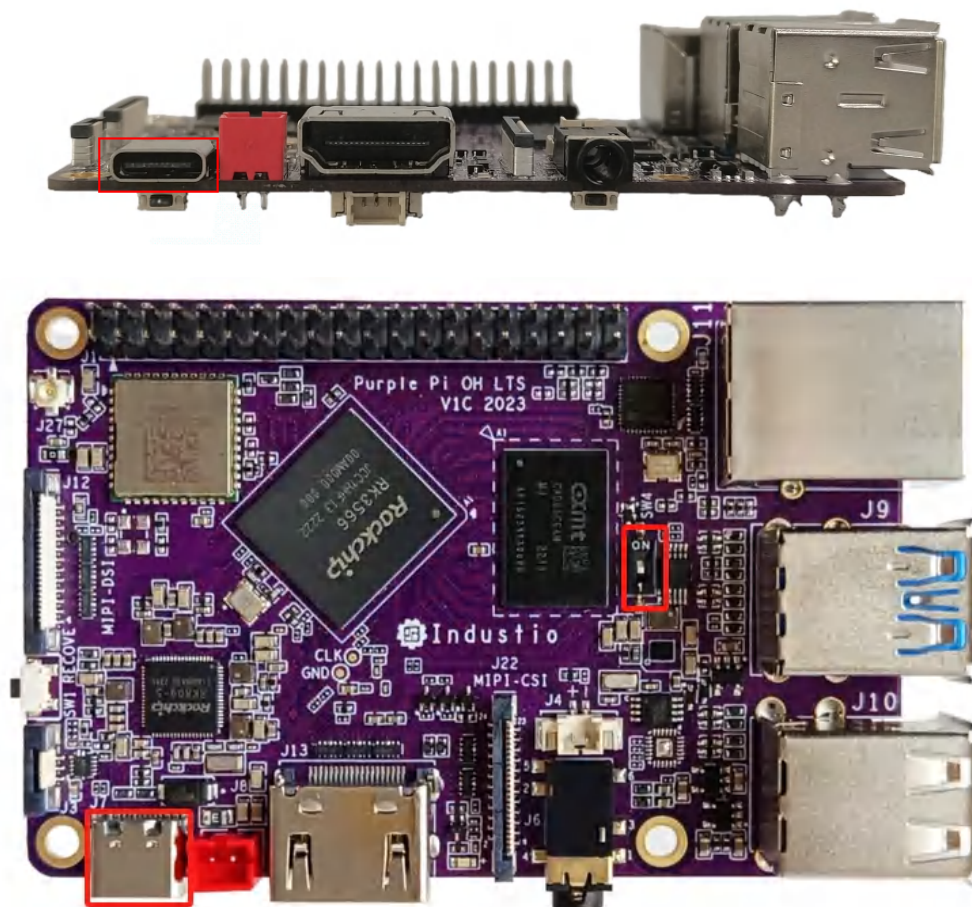


串口登录默认以root用户登录，且没有设置密码，命令如下：



1.2 ADB调试

ADB调试端口位于 (J7) , 将SW4切换到 **【1】** 档, 使用USB-C, 连接主板和电脑即可在电脑上使用 ADB调试, 如下图所示:



```
C:\WINDOWS\system32\cmd. X + v
Microsoft Windows [版本 10.0.22631.3593]
(c) Microsoft Corporation。保留所有权利。

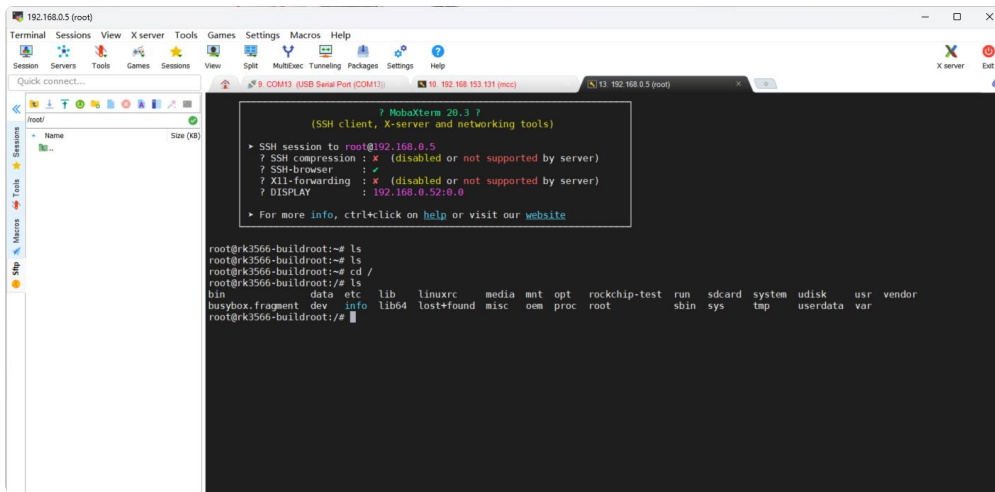
C:\Users\18294>adb shell
root@rk3566-buildroot:/# ls
bin          info         media        proc          sdcard        userdata
busybox.fragment  lib          misc          rockchip-test sys            usr
data         lib64        mnt          root          system        var
dev          linuxrc      oem          run           tmp           vendor
etc          lost+found  opt          sbin          udisk

root@rk3566-buildroot:/#
```

1.3 ssh调试

系统默认登录账号密码为root @ rockchip。

如下图所示：



2 双排针功能及接口

主板预留了40Pin的双排针接口，包含一路UART、一路I2C、一路SPI及19个GPIO，如下图所示：



2.1 双排针Pin功能定义

各个Pin的功能定义见下表：

序号	定义	电平	说明
1	VDD_3V3	3.3V	供电输出3.3V
2	VDD_5V	5V	可做5V输入供电，也可输出5V
3	I2C_SDA	3.3V	I2C数据信号
4	VDD_5V	5V	可做5V输入供电，也可输出5V
5	I2C_SCL	3.3V	I2C时钟信号
6	GND	GND	电源地
7	GPIO_7	3.3V	gpio编号为23
8	UART_TXD	3.3V	/dev/ttyS0 发送
9	GND	GND	电源地
10	UART_RXD	3.3V	/dev/ttyS0 接收
11	GPIO_0	3.3V	gpio编号为15
12	GPIO_1	3.3V	gpio编号为22
13	GPIO_2	3.3V	gpio编号为20
14	GND	GND	电源地
15	GPIO_3	3.3V	gpio编号为21
16	GPIO_4	3.3V	gpio编号为124
17	VDD_3V3	3.3V	供电输出3.3V
18	GPIO_5	3.3V	gpio编号为125
19	SPI_MOSI	3.3V	SPI数据信号
20	GND	GND	电源地
21	SPI_MISO	3.3V	SPI数据信号
22	GPIO_6	3.3V	gpio编号为130
23	SPI_SCLK	3.3V	SPI时钟信号
24	SPI_CS0	3.3V	SPI片选信号0

25	GND	GND	电源地
26	SPI_CS1	3.3V	SPI片选信号1
27	GPIO_30	3.3V	gpio编号为138
28	GPIO_31	3.3V	gpio编号为139
29	GPIO_21	3.3V	gpio编号为132
30	GND	GND	电源地
31	GPIO_22	3.3V	gpio编号为121
32	GPIO_26	3.3V	gpio编号为120
33	GPIO_23	3.3V	gpio编号为122
34	GND	GND	电源地
35	GPIO_24	3.3V	gpio编号为127
36	GPIO_27	3.3V	gpio编号为123
37	GPIO_25	3.3V	gpio编号为133
38	GPIO_28	3.3V	gpio编号为126
39	GND	GND	电源地
40	GPIO_29	3.3V	gpio编号为131

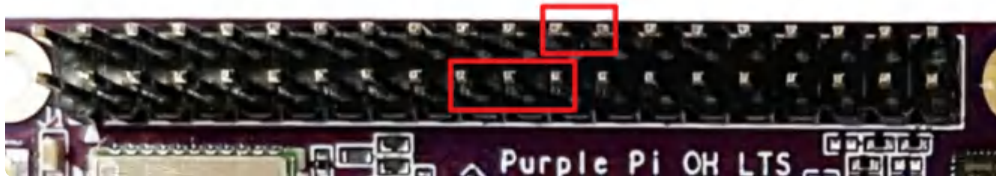
2.1. UART

双排针包含一路UART接口，电平类型为TTL电平，系统中对应的设备节点为/dev/ttyS0，如下图所示：



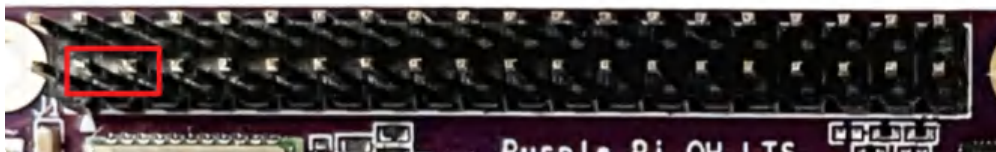
2.2 SPI

双排针包含一路SPI接口，系统中对应的设备节点为/dev/spidev3.0，如下所示：



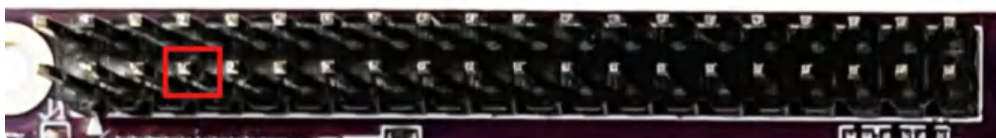
2.3 I2C

双排针包含一路I2C接口，系统种对应的设备节点为/dev/i2c-2，如下如所示：



2.4 GPIO

双排针共包含19个GPIO接口，如下如所示：



以Pin7的GPIO_7为例，对应系统的gpio number 为23，可通过sysctl的方式控制此路GPIO的方向和电平。

输出模式，命令如下：

```
1 #导出gpio23
2 echo 23 > /sys/class/gpio/export
3 # 设置GPIO方向为输出
4 echo out > /sys/class/gpio/gpio23/direction
5 # 设置输出高电平
6 echo 1 > /sys/class/gpio/gpio23/value
7 # 设置输出低电平
8 echo 0 > /sys/class/gpio/gpio23/value
```

注意：输出高指3.3v，输出低指0v。

输入模式，命令如下：

```

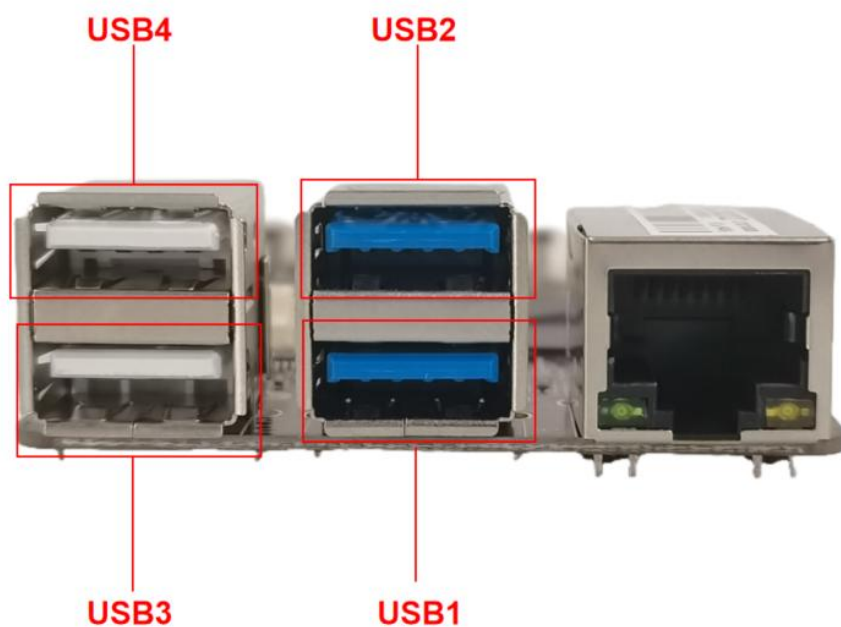
1  #导出gpio23, 如果已经导出过可以不用此操作
2  echo 23 > /sys/class/gpio/export
3  # 设置GPIO方向为输入
4  echo in > /sys/class/gpio/gpio23/direction
5  # 读取GPIO接口电平
6  cat /sys/class/gpio/gpio23/value

```

返回0指输入低电平（0v），返回1指输入高电平（3.3v）。

3 USB

主板共配置4路USB接口，分别记作USB1-4。如下图所示：



编号	类型	位置
usb1	host-2.0	J9（下）
usb2	host-3.0	J9（上）
usb3	host-2.0	J10（下）
usb4	host-2.0	J10（上）

注意：如果要使用USB1，需要将SW4切换至【ON】档，且ADB功能无法使用。

当USB1-4插入U盘后，会自动挂载/media/udisk1/目录下。

```
C:\WINDOWS\system32\cmd. X + v
root@rk3566-buildroot:/media/udisk1# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/root        921M  535M  321M  63% /
devtmpfs         973M    0  973M   0% /dev
tmpfs            984M  240K  983M   1% /tmp
tmpfs            984M  344K  983M   1% /run
tmpfs            984M    0  984M   0% /dev/shm
/dev/mmcblk0p7   121M   12M  101M  11% /oem
/dev/mmcblk0p8   8.0G   96K   7.6G   1% /userdata
/dev/sda1        500M  261M  240M  53% /mnt/udisk
/dev/sda2        29G   13G   16G  45% /media/udisk1
root@rk3566-buildroot:/media/udisk1#
```

3.1 电源控制

默认所有USB的电源都是开启的，我们提供了开启/关闭电源的方法，如有需要，你可以通过该方法控制它们的电源。

编号	电源控制节点
USB1	/sys/class/leds/usb_otg_pwr/brightness
USB2	/sys/class/leds/usb_host1_pwr/brightness
USB3	/sys/class/leds/usb_host2_pwr/brightness
USB4	/sys/class/leds/usb_host3_pwr/brightness

打开USB2的电源，命令如下：

```
▼ Bash |
1 $ echo 255 > /sys/class/leds/usb_host1_pwr/brightness
```

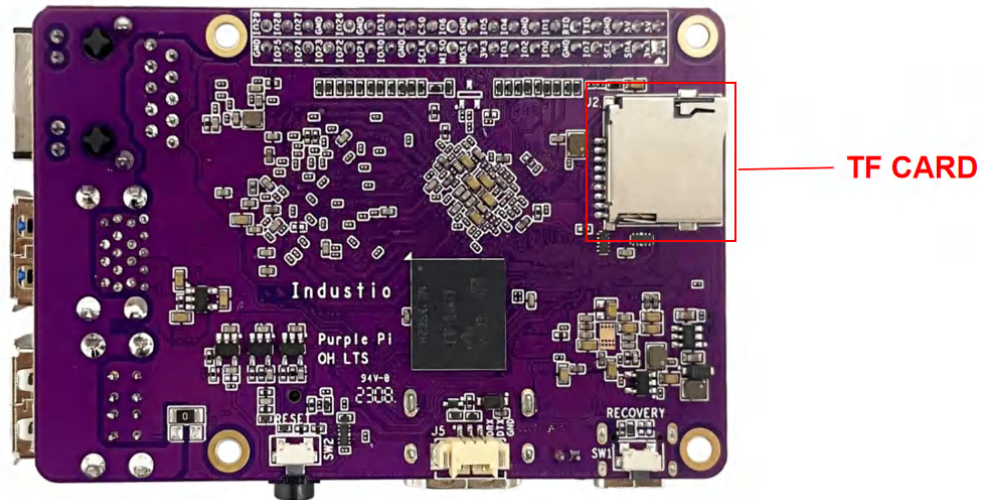
关闭USB2的电源，命令如下：

```
▼ Bash |
1 $ echo 0 > /sys/class/leds/usb_host1_pwr/brightness
```

USB1-4的电源控制方法类似。

4 TF CARD

主板配置了一个TF CARD接口，位于（J2），如下图所示：



当TF CARD接口插入TF卡后，会自动挂载到/media/目录下，如下图所示：

```
C:\WINDOWS\system32\cmd. X + v
root@rk3566-buildroot:/#
root@rk3566-buildroot:/# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/root        921M  535M  321M  63% /
devtmpfs         973M    0  973M   0% /dev
tmpfs            984M  244K  983M   1% /tmp
tmpfs            984M  336K  983M   1% /run
tmpfs            984M    0  984M   0% /dev/shm
/dev/mmcblk0p7   121M   12M  101M  11% /oem
/dev/mmcblk0p8   8.0G   96K   7.6G   1% /userdata
/dev/mmcblk1p1   500M  261M  240M  53% /mnt/sdcard
/dev/mmcblk1p2   29G   13G   16G  45% /media/sdcard1
root@rk3566-buildroot:/# |
```

5 Ethernet

主板配置了一个1000M以太网接口，位于（J11），在系统中其对应的网络设备节点为eth0，如下图所示：



5.1 查看以太网IP地址

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，命令如下：

```
1  $ ifconfig
2  eth0      Link encap:Ethernet  HWaddr 9E:B6:52:71:07:65
3            inet addr:192.168.0.8  Bcast:192.168.0.255  Mask:255.255.255.0
4            inet6 addr: fe80::32ca:9638:fce1:5aa5/64 Scope:Link
5            UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
6            RX packets:5337 errors:0 dropped:100 overruns:0 frame:0
7            TX packets:390 errors:0 dropped:0 overruns:0 carrier:0
8            collisions:0 txqueuelen:1000
9            RX bytes:435963 (425.7 KiB)  TX bytes:36466 (35.6 KiB)
10           Interrupt:40
```

5.2 设置以太网临时IP地址

```
1  $ ifconfig eth0 192.168.1.123
```

5.3 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，命令如下：

```

1  # interface file auto-generated by buildroot
2
3  auto lo
4  iface lo inet loopback
5
6  auto eth0
7  iface eth0 inet static
8  address 192.168.0.18
9  netmask 255.255.255.255
10 gateway 192.168.0.1
11 dns-nameservers 114.114.114.114
12
13

```

其中，dns-nameservers一项为默认dns，如下图所示：

```

eth0      Link encap:Ethernet  HWaddr 9E:B6:52:71:07:65
          inet addr:192.168.0.18  Bcast:0.0.0.0  Mask:255.255.255.255
          inet6 addr: fe80::32ca:9638:fce1:5aa5/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:705  errors:0 dropped:11 overruns:0 frame:0
          TX packets:153  errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:65539 (64.0 KiB)  TX bytes:18478 (18.0 KiB)
          Interrupt:40

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:36  errors:0 dropped:0 overruns:0 frame:0
          TX packets:36  errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2608 (2.5 KiB)  TX bytes:2608 (2.5 KiB)

wlan0     Link encap:Ethernet  HWaddr 14:D4:24:28:D9:2F
          inet addr:169.254.45.50  Bcast:169.254.255.255  Mask:255.255.0.0
          inet6 addr: fe80::dc00:c53:87be:c0bf/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0  errors:0 dropped:0 overruns:0 frame:0
          TX packets:18  errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:2312 (2.2 KiB)

```

6 WIFI

在使用 WIFI时连接好WiFi天线，否则可能会导致WiFi无法连接，设备节点为wlan0，命令如下：

```
▼ Bash |
1 wlan0      Link encap:Ethernet  HWaddr 14:D4:24:28:D9:2F
2            inet addr:169.254.45.50  Bcast:169.254.255.255  Mask:255.255.0.0
3            inet6 addr: fe80::dc00:c53:87be:c0bf/64 Scope:Link
4            UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
5            RX packets:0 errors:0 dropped:0 overruns:0 frame:0
6            TX packets:18 errors:0 dropped:0 overruns:0 carrier:0
7            collisions:0 txqueuelen:1000
8            RX bytes:0 (0.0 B)  TX bytes:2312 (2.2 KiB)
```

系统开机通过 /etc/init.d/S36load_wifi_modules脚本开启WiFi服务 修改/userdata/cfg/wpa_supplicant.conf，填写正确的热点账号和密码，命令如下：

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

使用wpa_supplicant进行连接执行连接，命令如下：

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

如果发现没有自动分配ip，则需要主动分配ip地址，命令如下：

```
▼ Bash |
1 ifconfig wlan0 192.168.1.121
```

尝试ping一下路由器，命令如下：

▼

Bash |

```
1 ping -I wlan0 192.168.1.1
```

有时候ping baidu.com的时候没反应，这可能是因为没把路由表加进去，命令如下：

▼

Bash |

```
1 root@RK3566-buildroot:~# route
2 Kernel IP routing table
3 Destination      Gateway            Genmask           Flags Metric Ref    Use If
4 default          routerlogin.cn    0.0.0.0           UG      202    0      0 et
5 192.168.0.0      *                 255.255.255.0     U       202    0      0 et
6 192.168.1.0      *                 255.255.255.0     U       303    0      0 wl
7
8
9 root@RK3566-buildroot:~# route add default gw 192.168.1.1
10 root@RK3566-buildroot:~#
```

7 Bluetooth

设备节点为hci0，通过/usr/sbin/bt-setup脚本开启蓝牙功能

蓝牙功能开启后，将产生hci0节点，命令如下：

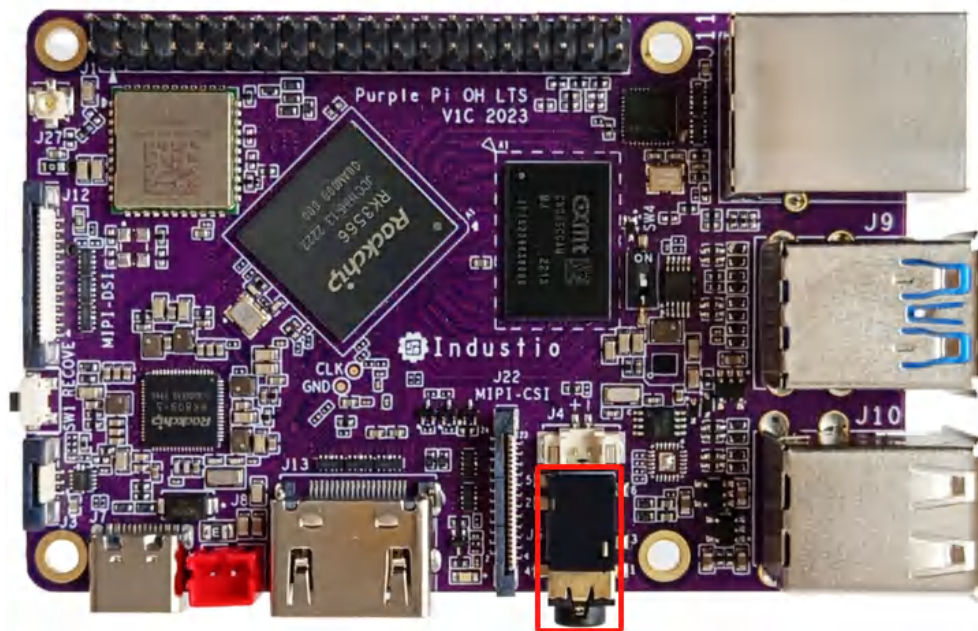
```
1 $ hciconfig -a
2 hci0: Type: Primary Bus: UART
3 BD Address: B8:B3:C9:59:62:DD ACL MTU: 1021:8 SCO MTU: 64:1
4 UP RUNNING
5 RX bytes:884 acl:0 sco:0 events:64 errors:0
6 TX bytes:4979 acl:0 sco:0 commands:64 errors:0
7 Features: 0xbf 0xfe 0xcf 0xfe 0xdb 0xff 0x7b 0x87
8 Packet type: DM1 DM3 DM5 DH1 DH3 DH5 HV1 HV2 HV3
9 Link policy: RSWITCH SNIFF
10 Link mode: SLAVE ACCEPT
11 Name: 'linaro-alip'
12 Class: 0x3c0000
13 Service Classes: Rendering, Capturing, Object Transfer, Audio
14 Device Class: Miscellaneous,
15 HCI Version: 4.2 (0x8) Revision: 0x10d
16 LMP Version: 4.2 (0x8) Subversion: 0x6119
17 Manufacturer: Broadcom Corporation (15)
18
```

测试蓝牙功能，命令如下：

```
1 bluetoothctl
2 hci0 new_settings: powered bondable ssp br/edr le secure-conn
3 Agent registered
4 [CHG] Controller 4C:38:00:2B:C1:71 Pairable: yes
5 [bluetooth]#
6 #打开蓝牙
7 [bluetooth]# power on
8 #扫描蓝牙
9 [bluetooth]# scan on
10 # # 开启广播
11 [bluetooth]# discoverable on
12 # 允许配对
13 [bluetooth]# pairable on
14 #查看蓝牙设备
15 [bluetooth]# devices
16 #信任蓝牙设备
17 [bluetooth]# trust 7C:C1:80:09:DD:6C
18 #蓝牙配对
19 [bluetooth]# pair 7C:C1:80:09:DD:6C
20 #连接蓝牙
21 [bluetooth]# connect 7C:C1:80:09:DD:6C
```

8 音频

主板配置了一路耳机接口，位于（J6），具备进行音频输出及录音功能，如下图所示：



8.1 查看声卡设备

命令如下：

```
1  $ aplay -l
2  **** List of PLAYBACK Hardware Devices ****
3  card 0: Dummy [Dummy], device 0: Dummy PCM [Dummy PCM]
4      Subdevices: 8/8
5      Subdevice #0: subdevice #0
6      Subdevice #1: subdevice #1
7      Subdevice #2: subdevice #2
8      Subdevice #3: subdevice #3
9      Subdevice #4: subdevice #4
10     Subdevice #5: subdevice #5
11     Subdevice #6: subdevice #6
12     Subdevice #7: subdevice #7
13  card 1: Loopback [Loopback], device 0: Loopback PCM [Loopback PCM]
14      Subdevices: 8/8
15      Subdevice #0: subdevice #0
16      Subdevice #1: subdevice #1
17      Subdevice #2: subdevice #2
18      Subdevice #3: subdevice #3
19      Subdevice #4: subdevice #4
20      Subdevice #5: subdevice #5
21      Subdevice #6: subdevice #6
22      Subdevice #7: subdevice #7
23  card 1: Loopback [Loopback], device 1: Loopback PCM [Loopback PCM]
24      Subdevices: 8/8
25      Subdevice #0: subdevice #0
26      Subdevice #1: subdevice #1
27      Subdevice #2: subdevice #2
28      Subdevice #3: subdevice #3
29      Subdevice #4: subdevice #4
30      Subdevice #5: subdevice #5
31      Subdevice #6: subdevice #6
32      Subdevice #7: subdevice #7
33  card 2: rockchiphdmi [rockchip,hdmi], device 0: rockchip,hdmi i2s-hifi-0
    [rockchip,hdmi i2s-hifi-0]
34      Subdevices: 1/1
35      Subdevice #0: subdevice #0
36  card 3: rockchiprk809co [rockchip,rk809-codec], device 0: fe410000.i2s-rk8
    17-hifi rk817-hifi-0 [fe410000.i2s-rk817-hifi rk817-hifi-0]
37      Subdevices: 1/1
38      Subdevice #0: subdevice #0
39
```

注意：MIPI固件仅有rk809-codec一个声卡。

8.2 播放音频

播放到HDMI，命令如下：

```
▼ Bash |
1  aplay -D hw:2,0 /usr/sounds/alsa/Rear_Center.wav
```

播放到耳机，命令如下：

```
▼ Bash |
1  //HDMI固件
2  aplay -D hw:3,0 /usr/sounds/alsa/Rear_Center.wav
3
4  //MIPI固件
5  aplay -D hw:2,0 /usr/sounds/alsa/Rear_Center.wav
```

8.3 录音

使用arecord工具可以进行录音测试，命令如下：

```
▼ Bash |
1  //HDMI固件
2  root@rk3566-buildroot:~# arecord -D hw:3,0 -r 48000 -c 2 -f S16_LE test.wa
v
3  Recording WAVE 'test.wav' : Signed 16 bit Little Endian, Rate 48000 Hz, St
ereo
4  ^CAborted by signal Interrupt...
5  root@rk3566-buildroot:~#
6
7  //MIPI固件
8  root@rk3566-buildroot:~# arecord -D hw:2,0 -r 48000 -c 2 -f S16_LE test.wa
v
9  Recording WAVE 'test.wav' : Signed 16 bit Little Endian, Rate 48000 Hz, St
ereo
10 ^CAborted by signal Interrupt...
11 root@rk3566-buildroot:~#
```

录音完后播放测试，命令如下：

```
1 //HDMI固件
2 root@rk3566-buildroot:~# aplay -D hw:3,0 ./test.wav
3 Playing WAVE './test.wav' : Signed 16 bit Little Endian, Rate 48000 Hz, Stereo
4 root@rk3566-buildroot:~#
5
6 //MIPI固件
7 root@rk3566-buildroot:~# aplay -D hw:2,0 ./test.wav
8 Playing WAVE './test.wav' : Signed 16 bit Little Endian, Rate 48000 Hz, Stereo
9 root@rk3566-buildroot:~#
```

9 RTC

主板包含2个RTC，其中/dev/rtc0为外部RTC（HYM8563），/dev/rtc1为CPU内部的RTC（RK808）。系统默认使用rtc0的时间。

9.1 获取RTC时间

命令如下：

```
1 root@rk3566-buildroot:~# hwclock
2 Thu Aug 17 10:58:47 2023 0.000000 seconds
3 root@rk3566-buildroot:~#
```

9.2 设置RTC时间

命令如下：

```
▼ Bash |
1 $ date -s "2022-02-02 20:20:20"
2 Wed Feb 2 20:20:20 UTC 2022
3
4 $ hwclock -w
5
6 $ hwclock -r
7 Wed Feb 2 20:20:33 2022 0.000000 seconds
```

10 开机自启动

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

11 屏幕控制

11.1 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，命令如下

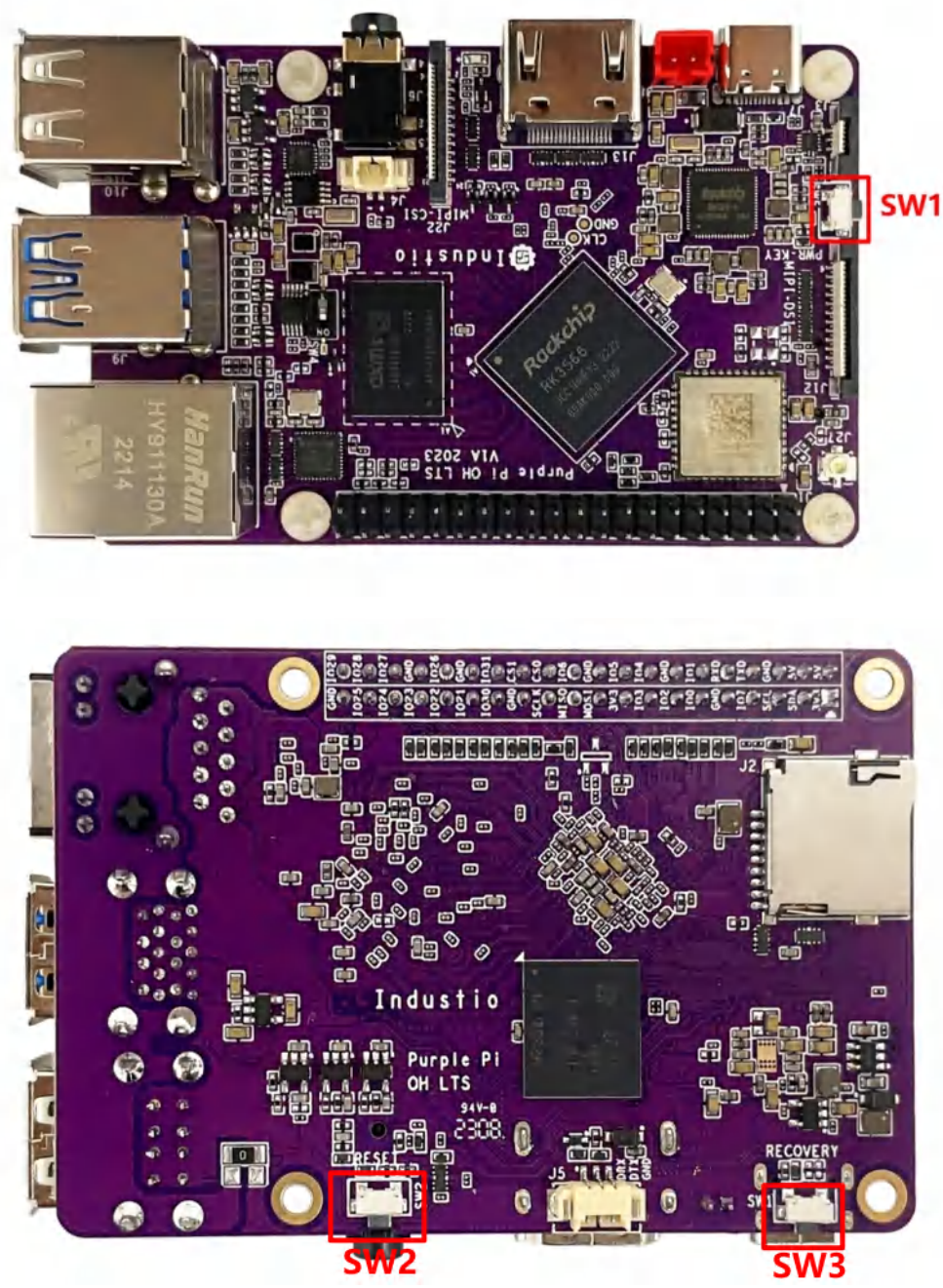
```
▼ Bash |
1 $ chmod a+w /sys/class/backlight/backlight/brightness
2 $ echo 100 > /sys/class/backlight/backlight/brightness
```

12 按键

主板共配置3个按键：

标记	名称	功能
SW1	recovery	用于烧录，或上报KEY_VOLUMEUP
SW2	reset	硬件复位
SW3	power_on	电源键，用于开关机及待机唤醒，或上报KEY_POWER

如下图所示：

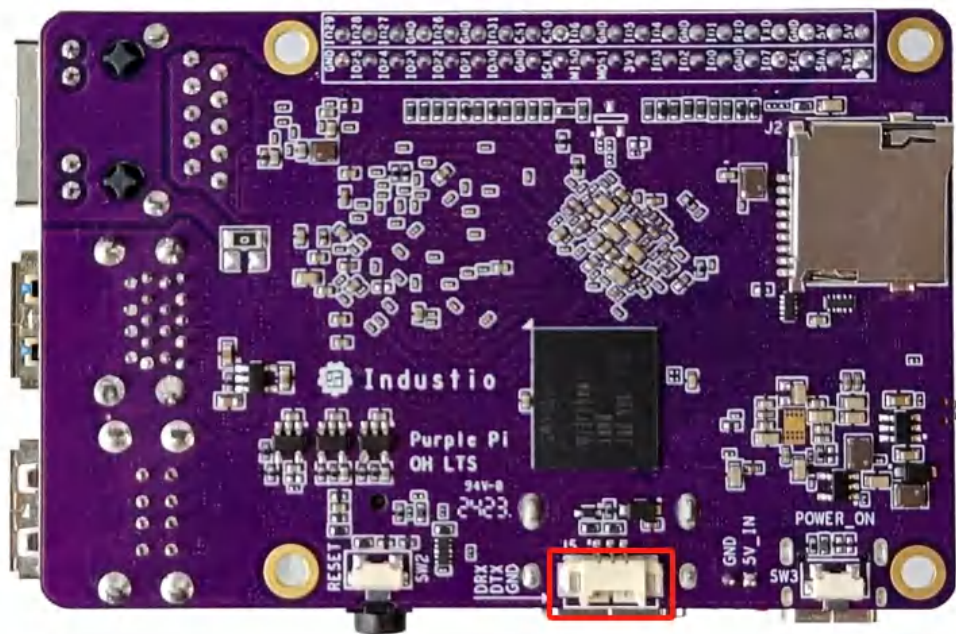


Purple Pi OH RK3566–Ubuntu系统

1 调试

1.1 串口调试

主板调试串口位于USB座子下方（J28），建议使用配套的USB转串口工具。如下图所示：



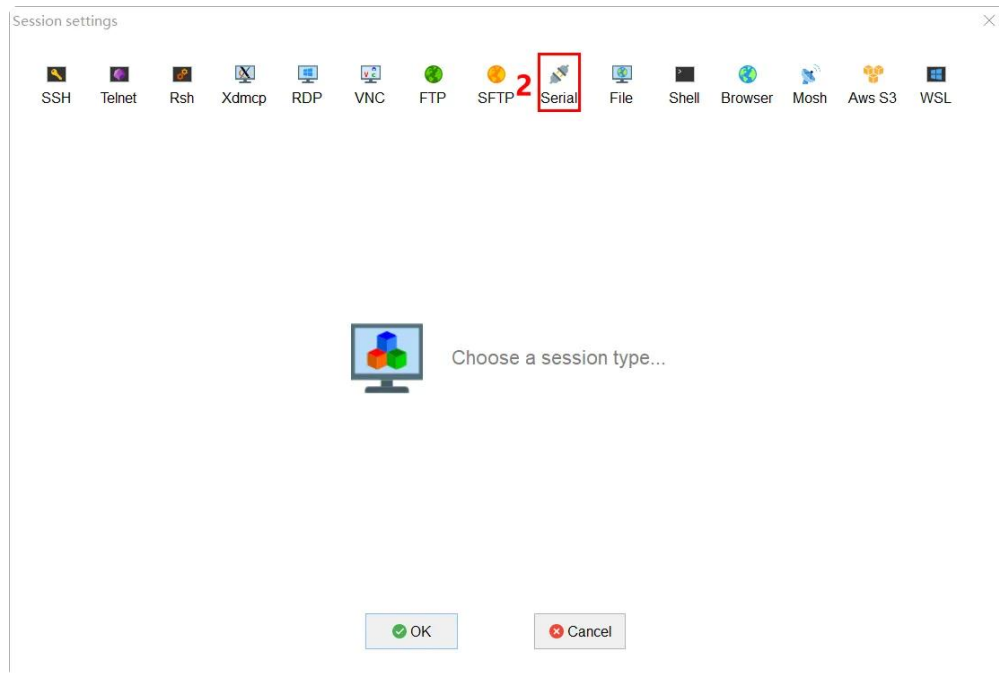
1. 打开MobaXterm，下载链接如下：

链接：https://pan.baidu.com/s/11ui4LTd2mq_9kiJpeL4bWg?pwd=1234

提取码：1234

☐ 文件名	↓ 修改时间	类型	大小
☐ RKDevTool_Release_v2.95.zip	2024-04-24 11:53	zip文件	2.30MB
☐ other_tools.txt	2024-04-25 15:31	txt文件	44B
☒ MobaXterm_Portable_v23.6.zip	2024-04-24 14:30	zip文件	39.99MB
☐ DriverAssitant_v5.11.zip	2024-04-24 11:52	zip文件	9.36MB

2. 选择sessionSerial，如下图所示：



3. 将Serial port修改为在设备管理器中找到的COM端口
4. 设置Speed(bsp)为1500000
5. 点击【OK】按钮，如下图所示

串口登录默认以root用户登录，且没有设置密码。

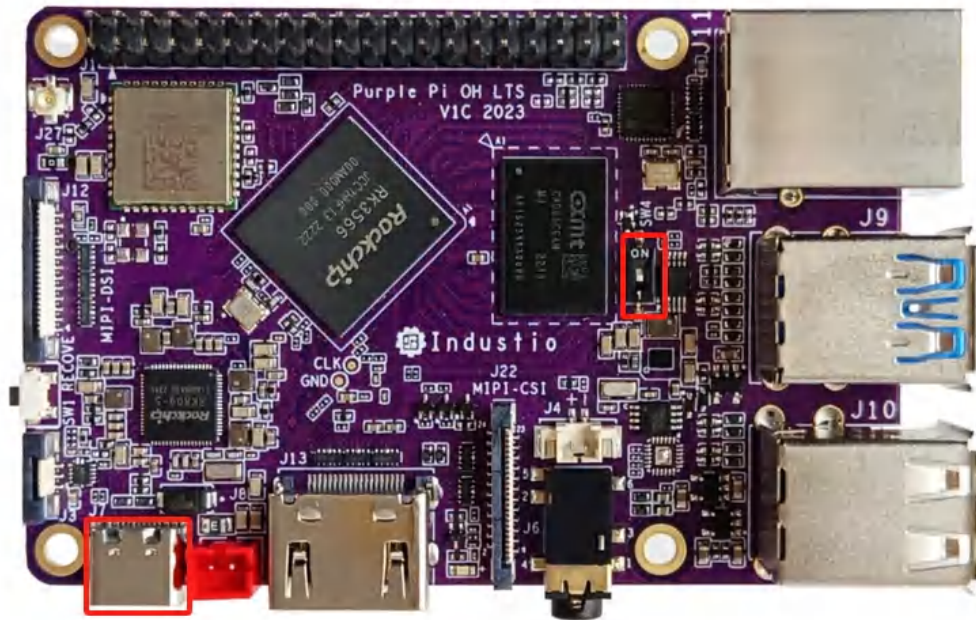
```

▼ Bash
1  $ ls
2  app.log  data  home  lost+found  opt  run  srv  tmp  var
3  bin      dev   lib   media      proc  sbin  sys  udisk  vendor
4  boot     etc   log   mnt        root  sdcard  system  usr

```

1.2 ADB调试

ADB调试端口位于（J7），将SW4切换到【1】档，使用USB-C，连接主板和电脑即可在电脑上使用ADB调试，如下图所示：



```
C:\Users\ronnie\Downloads>adb shell
adb server is out of date. killing...
* daemon started successfully *
root@ido:/# ls
ls
app.log  data  home  lost+found  opt  run  srv  tmp  var
bin      dev  lib   media      proc sbin sys  udisk vendor
boot     etc  log   mnt        root sdcard system usr
root@ido:/#
```

1.3 ssh调试

系统默认ssh账号和密码为 ido @ 123456。

```
ido@ido: ~
login as: ido
ido@192.168.1.149's password: 123456

Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 4.19.219 aarch64)

System information as of Fri Oct 28 03:29:24 UTC 2022

System load:  0.02 0.02 0.00  Up time:      40 min  Local users:  4
Memory usage: 17 % of 1971MB  IP:         192.168.1.149
Usage of /:   25% of 14G      storage/:   2% of 29G

Last login: Fri Oct 28 03:28:07 2022 from 192.168.1.147
ido@ido:~$
```

2 双排针功能及接口

主板预留了40Pin的双排针接口，包含一路UART、一路I2C、一路SPI及19个GPIO，如下图所示



2.1 双排针Pin功能定义

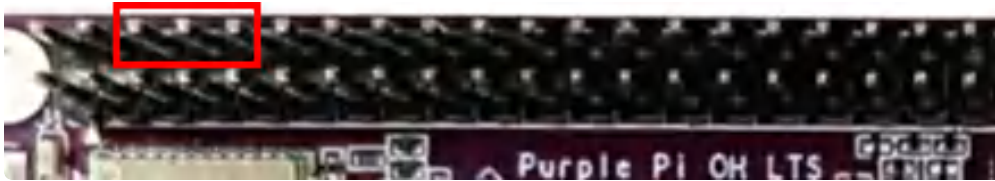
各个Pin的功能定义见下表：

序号	定义	电平	说明
1	VDD_3V3	3.3V	供电输出3.3V
2	VDD_5V	5V	可做5V输入供电，也可输出5V
3	I2C_SDA	3.3V	I2C数据信号
4	VDD_5V	5V	可做5V输入供电，也可输出5V
5	I2C_SCL	3.3V	I2C时钟信号
6	GND	GND	电源地
7	GPIO_7	3.3V	gpio编号为23
8	UART_TXD	3.3V	/dev/ttyS0 发送
9	GND	GND	电源地
10	UART_RXD	3.3V	/dev/ttyS0 接收
11	GPIO_0	3.3V	gpio编号为15
12	GPIO_1	3.3V	gpio编号为22
13	GPIO_2	3.3V	gpio编号为20
14	GND	GND	电源地
15	GPIO_3	3.3V	gpio编号为21
16	GPIO_4	3.3V	gpio编号为124
17	VDD_3V3	3.3V	供电输出3.3V
18	GPIO_5	3.3V	gpio编号为125

19	SPI_MOSI	3.3V	SPI数据信号
20	GND	GND	电源地
21	SPI_MISO	3.3V	SPI数据信号
22	GPIO_6	3.3V	gpio编号为130
23	SPI_SCLK	3.3V	SPI时钟信号
24	SPI_CS0	3.3V	SPI片选信号0
25	GND	GND	电源地
26	SPI_CS1	3.3V	SPI片选信号1
27	GPIO_30	3.3V	gpio编号为138
28	GPIO_31	3.3V	gpio编号为139
29	GPIO_21	3.3V	gpio编号为132
30	GND	GND	电源地
31	GPIO_22	3.3V	gpio编号为121
32	GPIO_26	3.3V	gpio编号为120
33	GPIO_23	3.3V	gpio编号为122
34	GND	GND	电源地
35	GPIO_24	3.3V	gpio编号为127
36	GPIO_27	3.3V	gpio编号为123
37	GPIO_25	3.3V	gpio编号为133
38	GPIO_28	3.3V	gpio编号为126
39	GND	GND	电源地
40	GPIO_29	3.3V	gpio编号为131

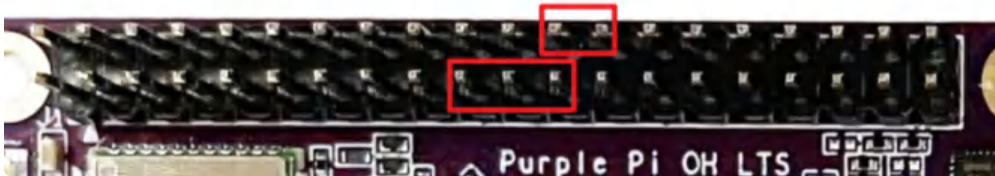
2.1.1 UART

双排针包含一路UART接口，电平类型为TTL电平，系统中对应的设备节点为/dev/ttyS0，如下图所示：



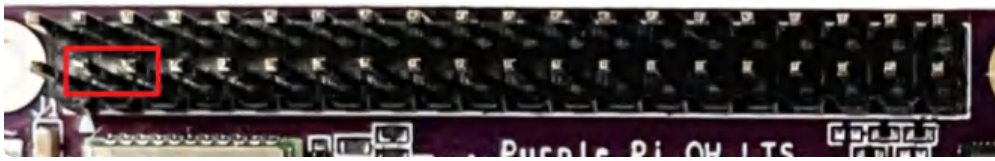
2.1.2 SPI

双排针包含一路SPI接口，系统中对应的设备节点为/dev/spidev3.0，如下如所示：



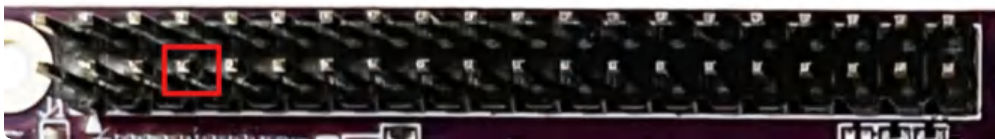
2.1.3 I2C

双排针包含一路I2C接口，系统种对应的设备节点为/dev/i2c-2，如下图所示：



2.1.4 GPIO

双排针共包含19个GPIO接口，如下图所示：



以Pin7的GPIO_7为例，对应系统的gpio number 为23，可通过sysctl的方式控制此路GPIO的方向和电平。

输出模式，命令如下：

```
1 #导出gpio23
2 echo 23 > /sys/class/gpio/export
3 # 设置GPIO方向为输出
4 echo out > /sys/class/gpio/gpio23/direction
5 # 设置输出高电平
6 echo 1 > /sys/class/gpio/gpio23/value
7 # 设置输出低电平
8 echo 0 > /sys/class/gpio/gpio23/value
```

注意：输出高指3.3v，输出低指0v。

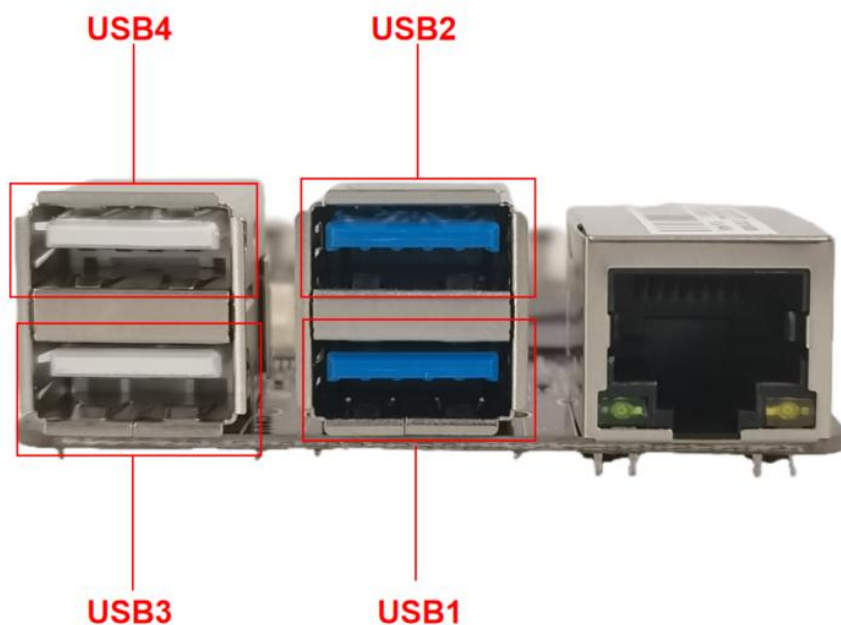
输入模式，命令如下：

```
1 #导出gpio23，如果已经导出过可以不用此操作
2 echo 23 > /sys/class/gpio/export
3 # 设置GPIO方向为输入
4 echo in > /sys/class/gpio/gpio23/direction
5 # 读取GPIO接口电平
6 cat /sys/class/gpio/gpio23/value
```

返回0指输入低电平（0v），返回1指输入高电平（3.3v）。

3 USB

主板共配置4路USB接口，分别记作USB1-4。如下图所示：



编号	类型	位置
usb1	host-2.0	J9（下）
usb2	host-3.0	J9（上）
usb3	host-2.0	J10（下）
usb4	host-2.0	J10（上）

注意：如果要使用USB1，需要将SW4切换至OFF档，且ADB功能无法使用。

当USB1-4插入U盘后，会自动挂载/media/ido/目录下。

▼

Bash

```
1 root@ido:~# ls /media/ido/
2 KINGSTON
```

3.1 电源控制

默认所有USB的电源都是开启的，我们提供了开启/关闭电源的方法，如有需要，你可以通过该方法控制它们的电源。

编号	电源控制节点
USB1	/sys/class/leds/usb_otg_pwr/brightness

USB2	/sys/class/leds/usb_host1_pwr/brightness
USB3	/sys/class/leds/usb_host2_pwr/brightness
USB4	/sys/class/leds/usb_host3_pwr/brightness

打开USB2的电源，命令如下：

```

1 root@ido:~# echo 255 > /sys/class/leds/usb_host1_pwr/brightness

```

关闭USB2的电源，命令如下：

```

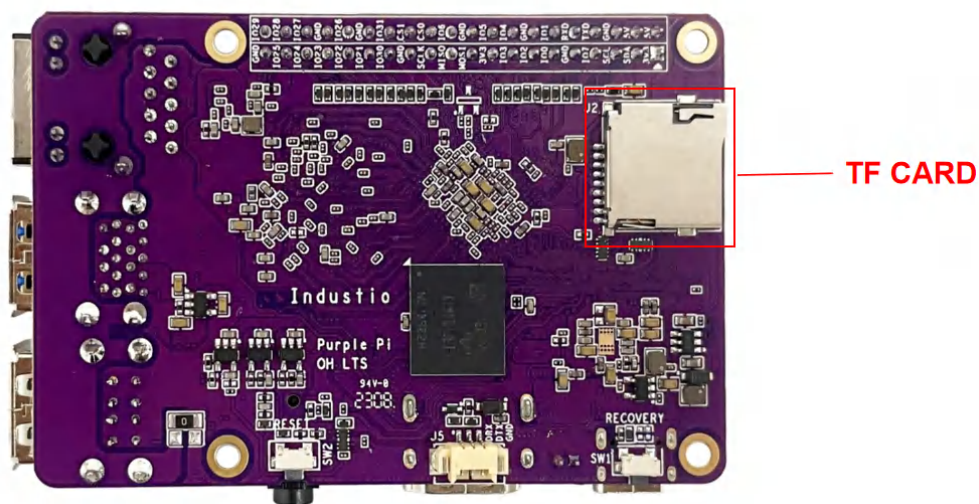
1 root@ido:~# echo 0 > /sys/class/leds/usb_host1_pwr/brightness

```

USB1-4的电源控制方法类似。

4 TF CARD

主板配置了一个TF CARD接口，位于（J2），当TF CARD接口插入TF卡后，会自动挂载到/media/ido/目录下，如下图所示：



```
1 root@ido:~# ls /media/ido/  
2 3533-3735  
3 root@ido:~#
```

5 Ethernet

主板配置了一个1000M以太网接口，位于J11，在系统中其对应的网络设备节点为eth0，如下图所示：



5.1 查看以太网IP地址

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，命令如下：

```
1 root@ido:~# ifconfig eth0  
2 eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
3     inet 192.168.1.133 netmask 255.255.255.0 broadcast 192.168.1.255  
4     inet6 fe80::3b43:b691:ded5:c497 prefixlen 64 scopeid 0x20<link>  
5     ether 82:4c:21:62:f5:35 txqueuelen 1000 (Ethernet)  
6     RX packets 29 bytes 4592 (4.4 KiB)  
7     RX errors 0 dropped 0 overruns 0 frame 0  
8     TX packets 43 bytes 4146 (4.0 KiB)  
9     TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
10    device interrupt 40  
11  
12 root@ido:~#
```

5.2 设置以太网临时IP地址

```
1 root@ido:~# ifconfig eth0 192.168.1.123
```

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），命令如下：

```
1 network:
2     version: 2
3     renderer: NetworkManager
4     ethernets:
5         eth0:
6             dhcp4: no
7             dhcp6: no
8             addresses: [192.168.1.10/24]
9             gateway4: 192.168.1.1
10            nameservers:
11                addresses: [8.8.8.8, 114.114.114.114]
```

然后重启网络，命令如下：

```
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，命令如下：

```
1 root@ido:~# ifconfig eth0
2 eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
3     inet 192.168.1.123 netmask 255.255.255.0 broadcast 192.168.1.255
4     inet6 fe80::b7c2:552b:4127:ee1c prefixlen 64 scopeid 0x20<link>
5     ether 82:4c:21:62:f5:35 txqueuelen 1000 (Ethernet)
6     RX packets 19 bytes 3281 (3.2 KB)
7     RX errors 0 dropped 0 overruns 0 frame 0
8     TX packets 36 bytes 5592 (5.5 KB)
9     TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
10    device interrupt 40
```

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，命令如下：

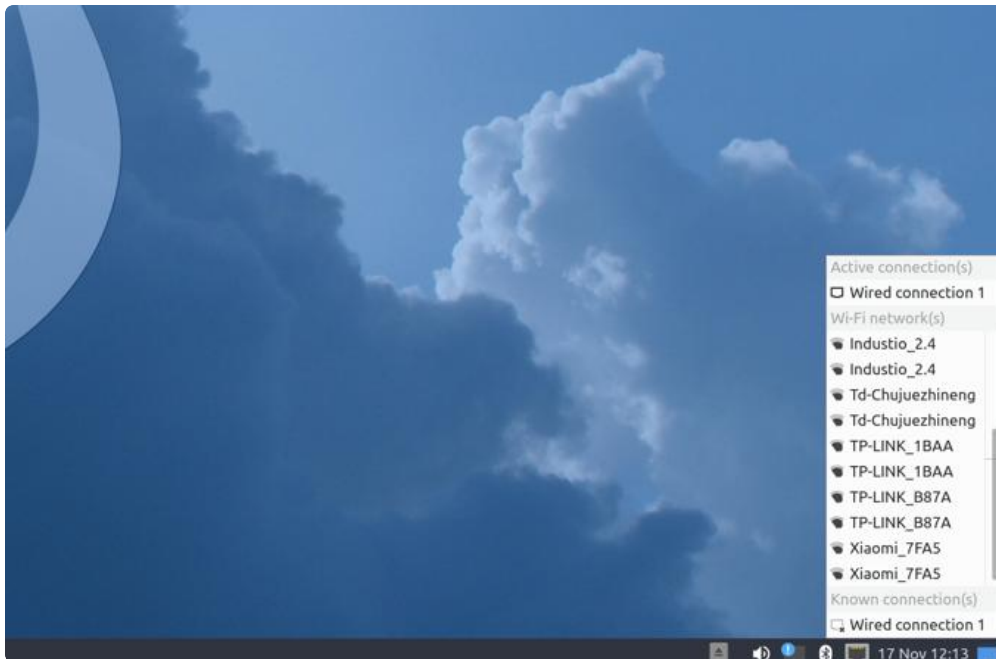
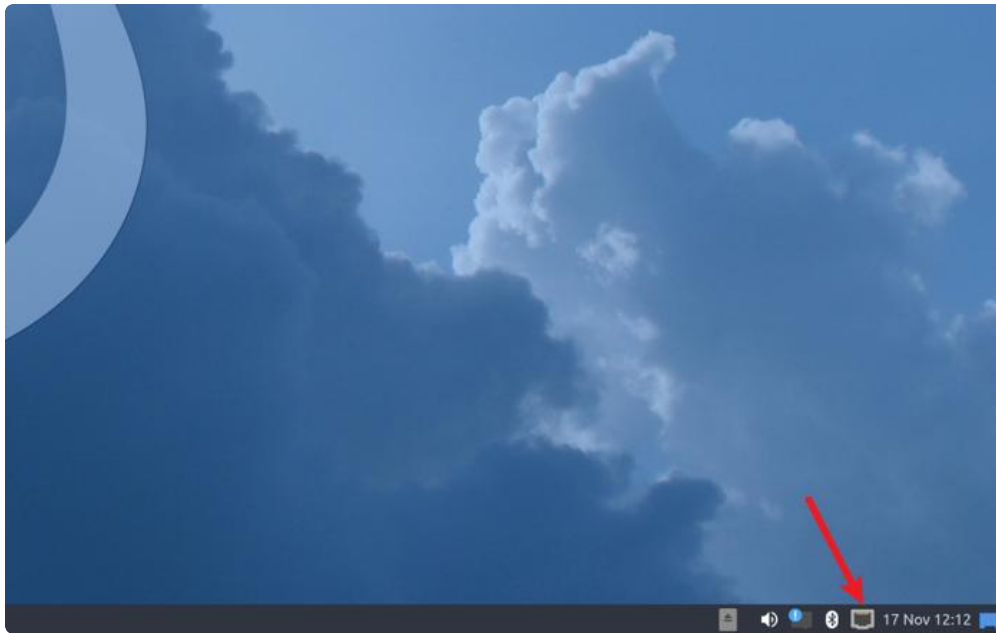
```
1 root@ido:~# ifconfig wlan0
2 wlan0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
3     ether 2c:d2:6b:10:ea:4d txqueuelen 1000 (Ethernet)
4     RX packets 0 bytes 0 (0.0 B)
5     RX errors 0 dropped 0 overruns 0 frame 0
6     TX packets 0 bytes 0 (0.0 B)
7     TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
8
9 root@ido:~#
```

6.1 连接热点

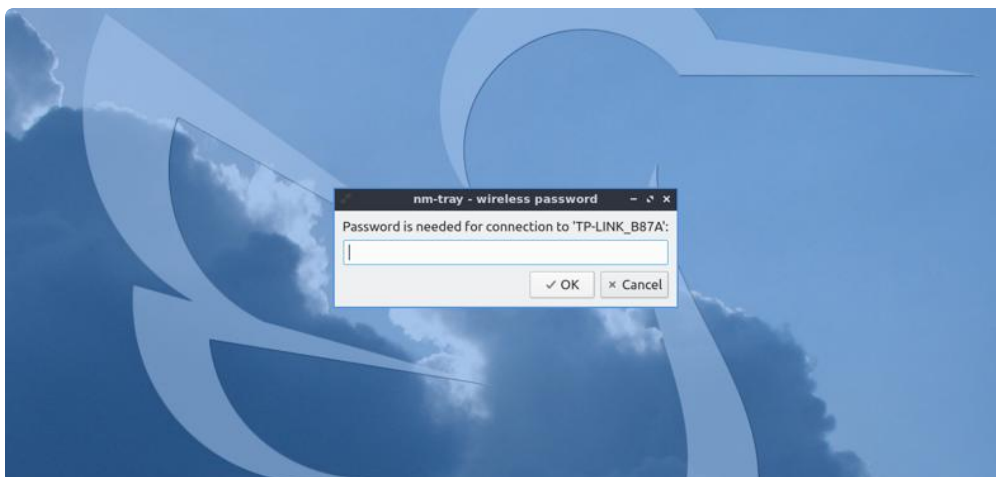
连接热点可以在桌面上操作，也可以使用命令行操作。

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：

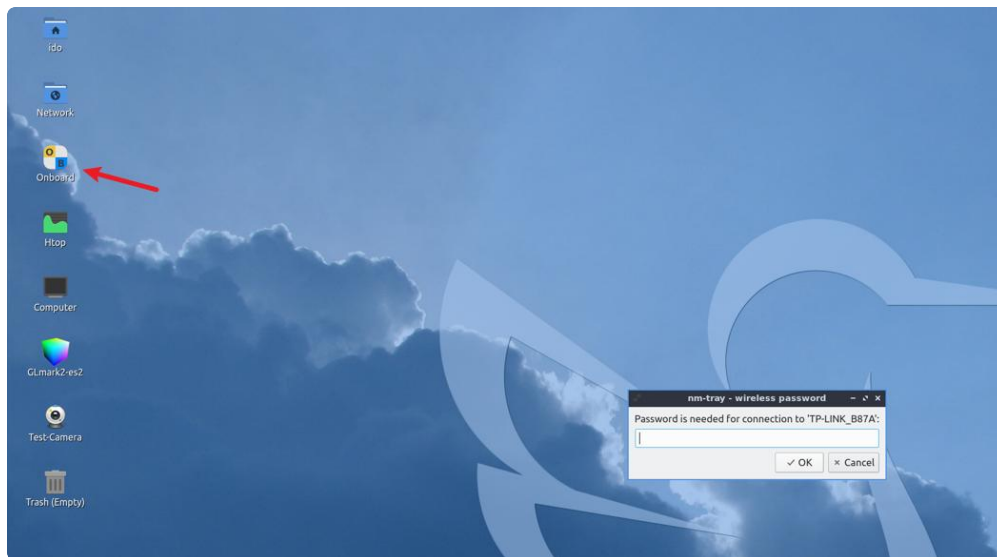


点击要连接的热点，弹出密码输入窗口，如下图所示：

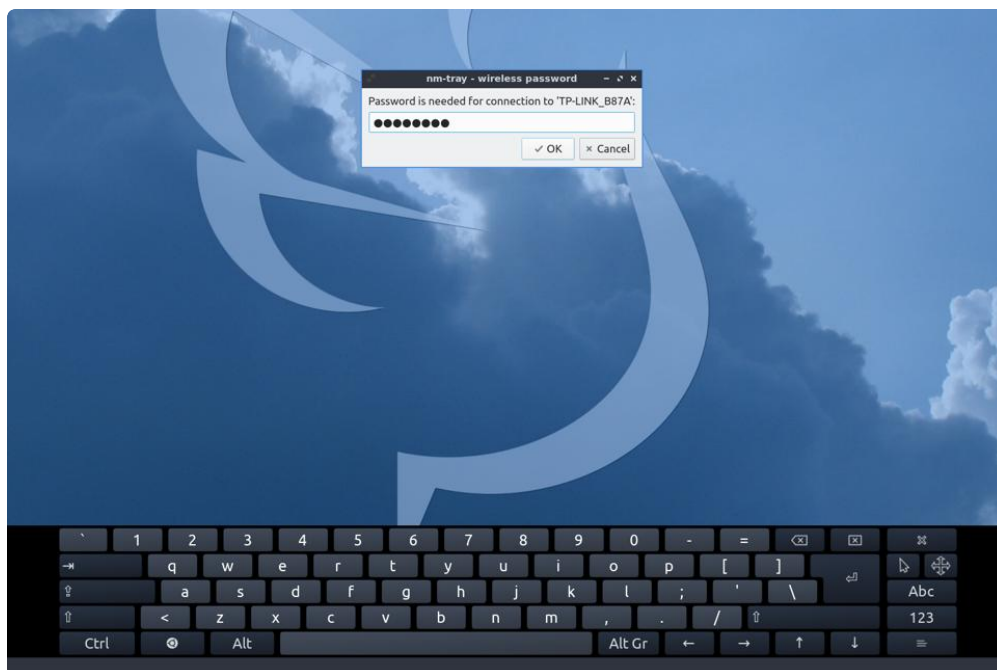


如果有连接键盘，直接输入密码即可；如果没有连接键盘，可以使用系统自带的软键盘。

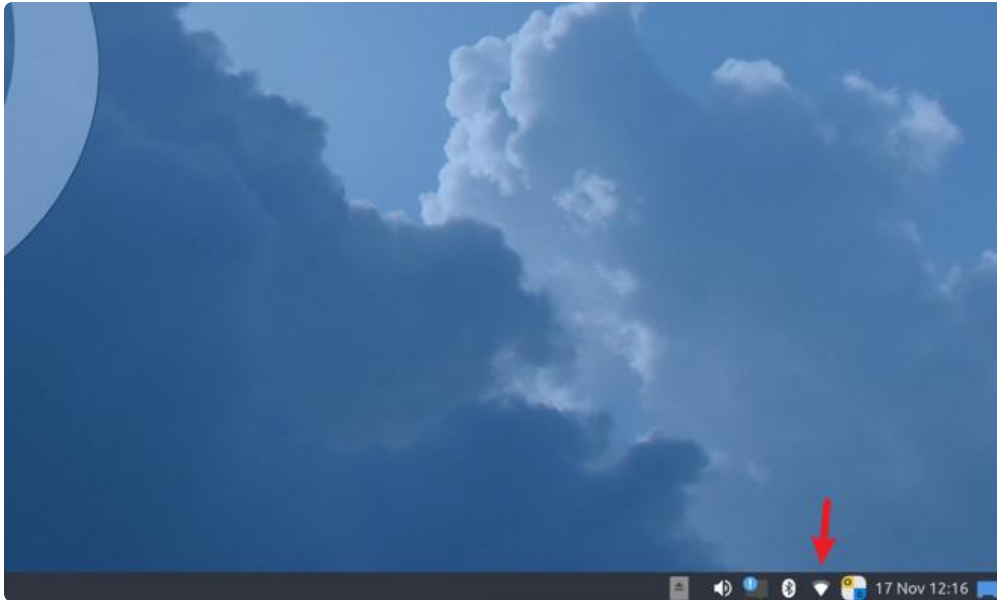
点击桌面的Onboard图标，即可打开系统自带的软键盘，如下图所示：



使用软键盘输入密码后，点击【OK】连接热点，如下图所示：



连接成功后，桌面右下角的网络图标将改变，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，命令如下：

```
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，命令如下：

```
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

1. 输入该命令是否能搜索到你的WiFi设备，命令如下：

```
1 sudo iwlist scanning
```

2. 使用下方命令进行连接WiFi网络，命令如下：

```
1 sudo nmcli d wifi connect cainiaocl password 12345678
```

其中cainiaocl, 12345678, 是你的账号密码，命令如下：

```
1 ifconfig
```

如果你的wlan0 出现了IP地址，则说明连接成功

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，命令如下：

```
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
```

6.3 测试WiFi的网络

命令如下：

```
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
```

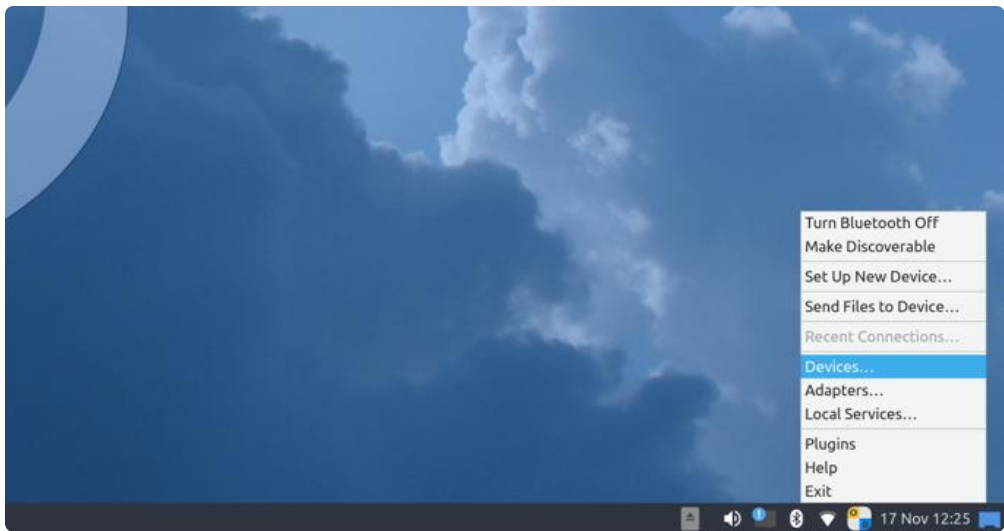
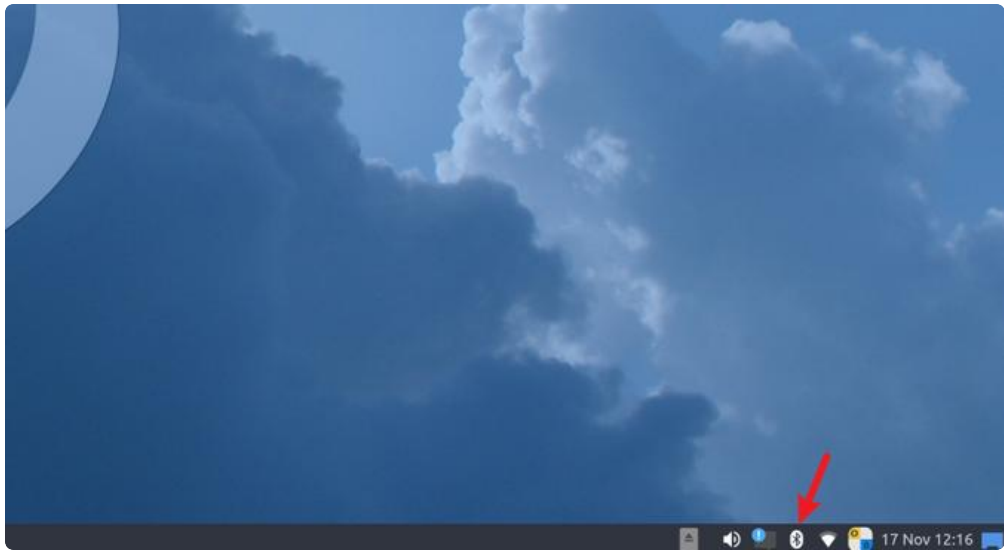
7 Bluetooth

系统开机默认打开蓝牙，对应的网络节点为hci0，命令如下：

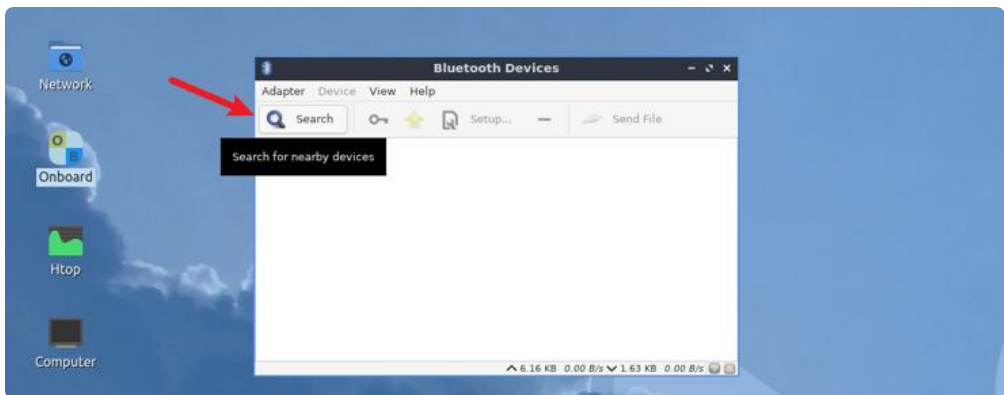
```
1 root@ido:~# hciconfig
2 hci0:   Type: Primary   Bus: UART
3         BD Address: 70:D5:2B:5B:22:22  ACL MTU: 1021:8  SCO MTU: 255:12
4         UP RUNNING
5         RX bytes:1665 acl:0 sco:0 events:57 errors:0
6         TX bytes:6311 acl:0 sco:0 commands:57 errors:0
7
8 root@ido:~#
```

7.1 连接蓝牙设备

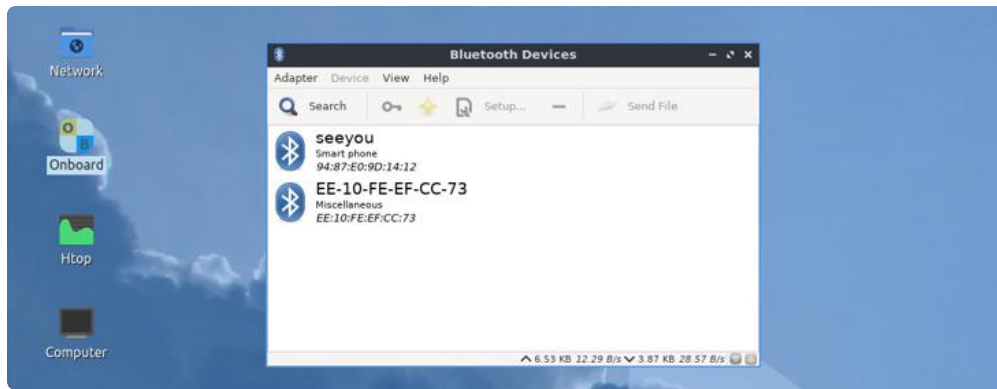
将鼠标放到桌面右下角蓝牙图标，右键->Devices，如下图所示：



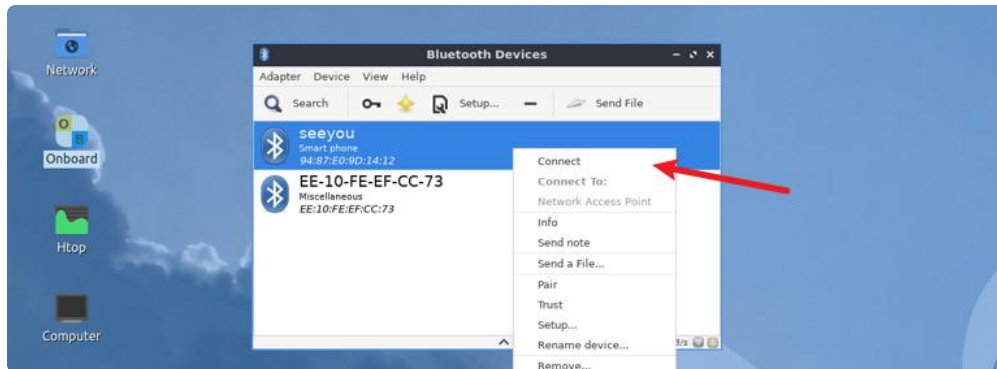
在弹出的窗口中，点击【Search】，如下图所示：



将看到附近的蓝牙设备列表，如下图所示：

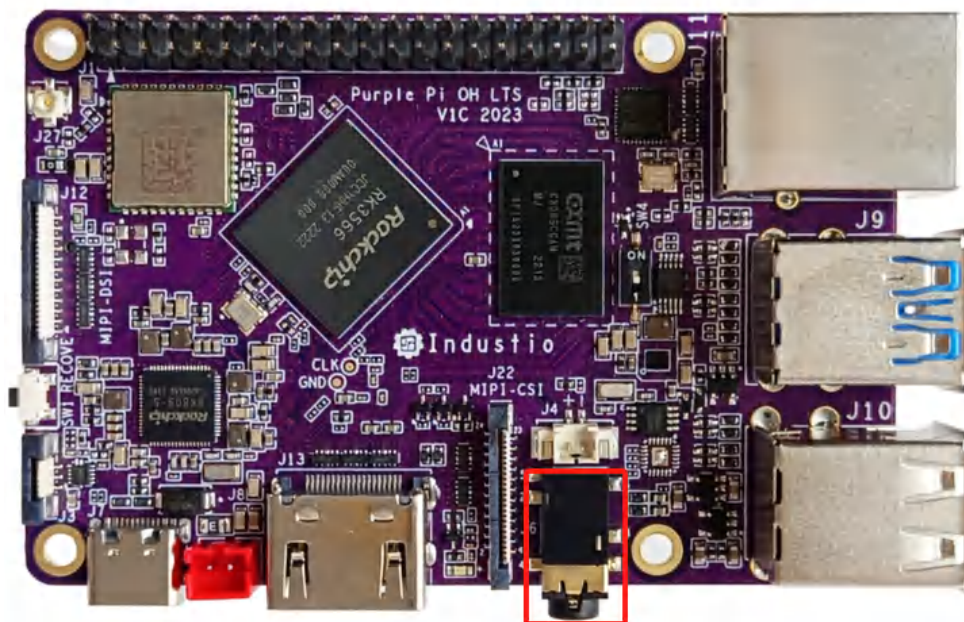


选中要连接设备，右键->Connect，即可连接该设备，如下图所示：



8 音频

主板配置了一路耳机接口，位于（J6），具备进行音频输出及录音功能，如下图所示：



8.1 查看声卡设备

命令如下：

```
▼ 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
9 root@ido:~#
```

注意：MIPI固件仅有rk809-codec一个声卡。

8.2 播放音频

播放到HDMI，命令如下：

```
▼ Bash |
1 aplay -D plughw:0,0 /usr/share/sounds/alsa/Rear_Center.wav
```

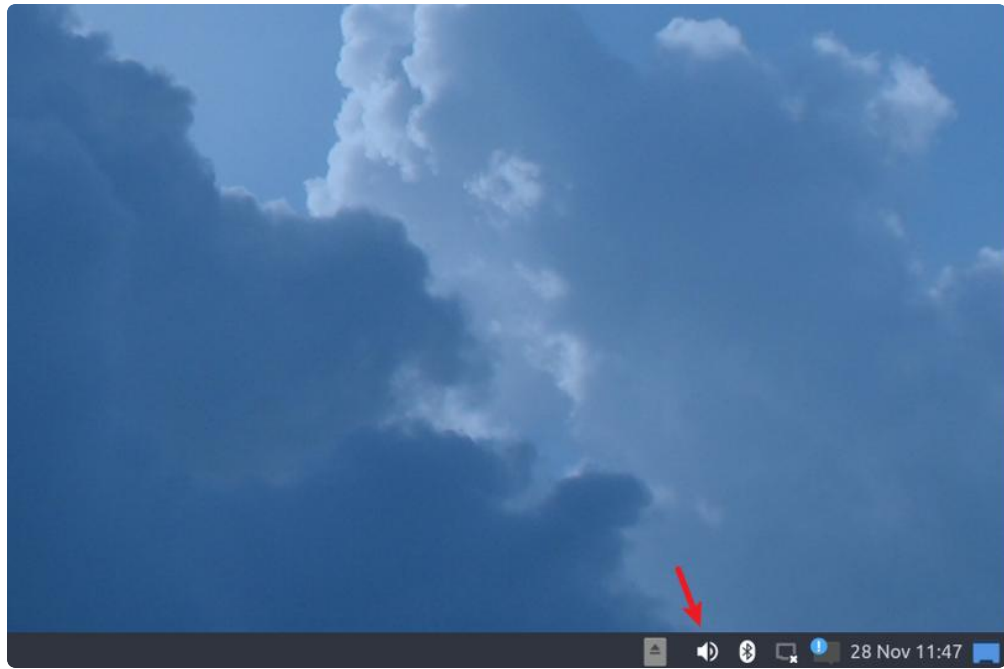
播放到耳机

插入耳机，执行以下命令：

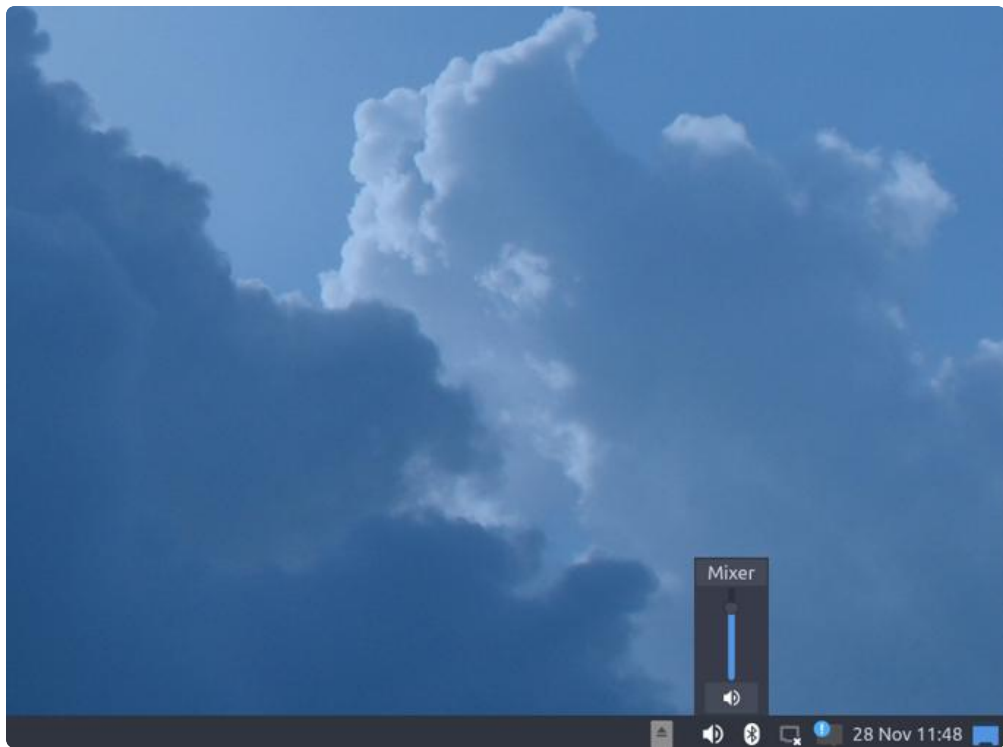
```
▼ Bash |
1 //HDMI固件
2 aplay -D plughw:1,0 /usr/share/sounds/alsa/Rear_Center.wav
3
4 //MIPI固件
5 aplay -D plughw:0,0 /usr/share/sounds/alsa/Rear_Center.wav
```

8.3 音量的调节

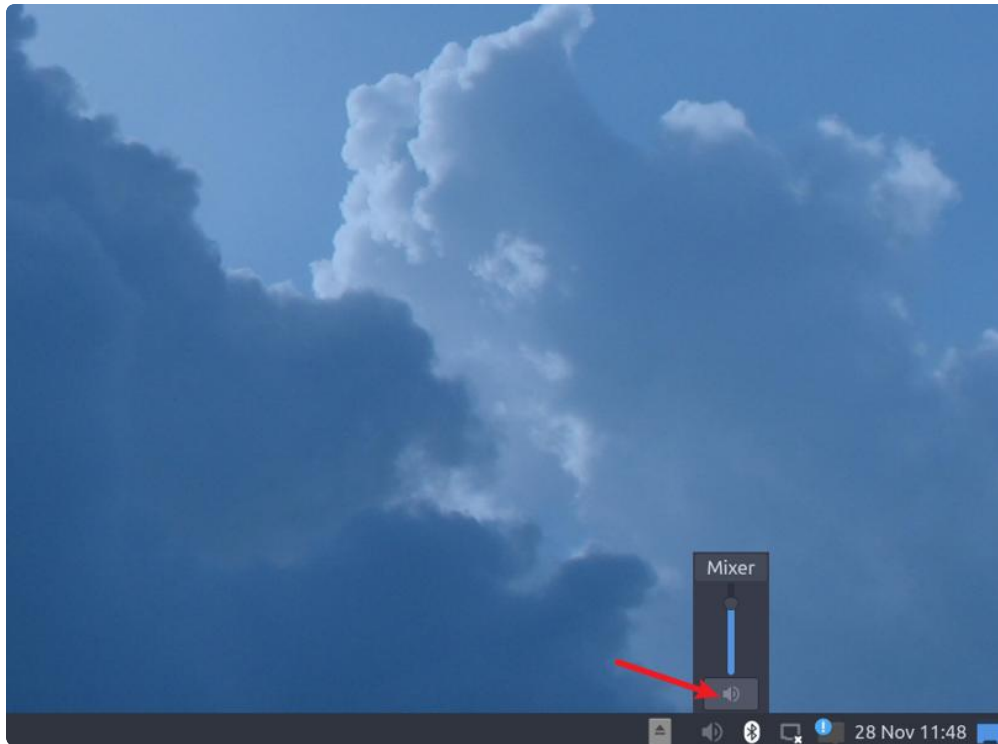
鼠标点击桌面右下角的音量图标，如下图所示：



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：



8.4 录音

需要接入带麦克风的耳机。

使用arecord工具可以进行录音测试，命令如下：

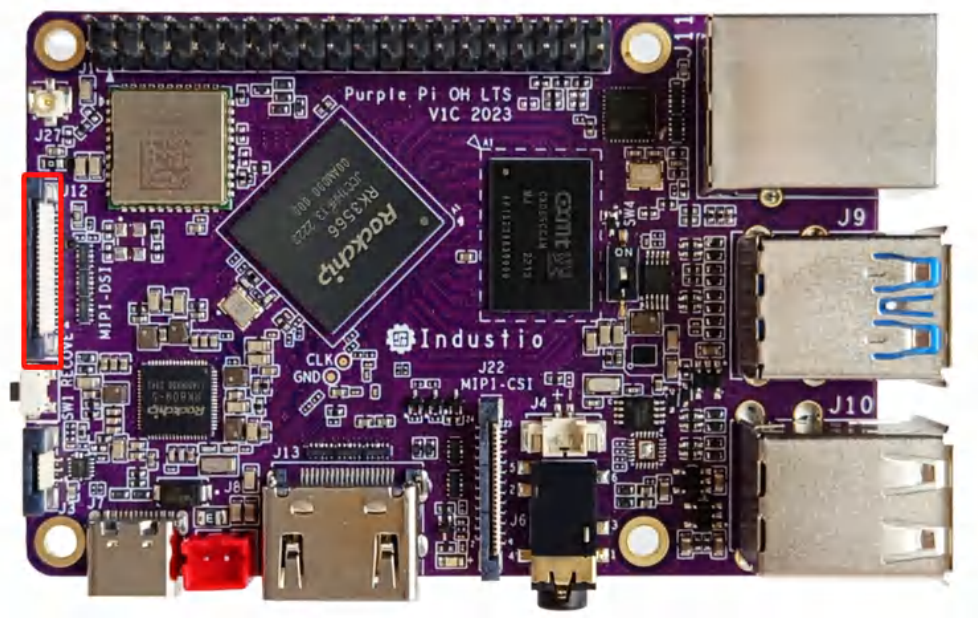
```
1 //HDMI固件
2 root@ido:~# arecord -D hw:1,0 -r 48000 -c 2 -f S16_LE test.wav
3 Recording WAVE 'test.wav' : Signed 16 bit Little Endian, Rate 48000 Hz, Stereo
4 ^CAborted by signal Interrupt...
5 root@ido:~#
6
7 //MIPI固件
8 root@ido:~# arecord -D hw:0,0 -r 48000 -c 2 -f S16_LE test.wav
9 Recording WAVE 'test.wav' : Signed 16 bit Little Endian, Rate 48000 Hz, Stereo
10 ^CAborted by signal Interrupt...
11 root@ido:~#
```

录音完后播放测试，命令如下：

```
1 //HDMI固件
2 root@ido:~# aplay -D plughw:1,0 ./test.wav
3 Playing WAVE './test.wav' : Signed 16 bit Little Endian, Rate 48000 Hz, Stereo
4 root@ido:~#
5
6 //MIPI固件
7 root@ido:~# aplay -D plughw:0,0 ./test.wav
8 Playing WAVE './test.wav' : Signed 16 bit Little Endian, Rate 48000 Hz, Stereo
9 root@ido:~#
```

9 Camera

主板默认适配OV5648和OV8858摄像头，摄像头接口位于（J22）,如下图所示：



9.1 测试

9.1.1 测试摄像头是否存在

命令如下：

```
1 root@ido:~# media-ctl -p -d /dev/media0
2 ...
3 - entity 67: rockchip-csi2-dphy0 (2 pads, 2 links)
4     type V4L2 subdev subtype Unknown flags 0
5     device node name /dev/v4l-subdev2
6     pad0: Sink
7         [fmt:SBGGR10_1X10/2592x1944@10000/150000 field:none]
8         <- "m00_b_ov5648 2-0036":0 [ENABLED]
9     pad1: Source
10        [fmt:SBGGR10_1X10/2592x1944@10000/150000 field:none]
11        -> "rkisp-csi-subdev":0 [ENABLED]
12
13 - entity 70: m00_b_ov5648 2-0036 (1 pad, 1 link)
14     type V4L2 subdev subtype Sensor flags 0
15     device node name /dev/v4l-subdev3
16     pad0: Source
17        [fmt:SBGGR10_1X10/2592x1944@10000/150000 field:none]
18        -> "rockchip-csi2-dphy0":0 [ENABLED]
19 root@ido:~#
```

结果显示m00_b_ov5648，说明摄像头存在，最高分辨率支持2592x1944。

9.1.2 抓取视频

使用v4l2-ctl工具可以抓取摄像头的视频数据流，命令如下：

```

1  root@ido:~# v4l2-ctl --verbose -d /dev/video0 --set-fmt-video=width=1920,height=1080,pixelformat='NV12' --stream-mmap=4 --set-selection=target=crop,flags=0,top=0,left=0,width=1920,height=1080 --stream-to=./out.yuv
2  VIDIOC_QUERYCAP: ok
3  VIDIOC_G_FMT: ok
4  VIDIOC_S_FMT: ok
5  Format Video Capture Multiplanar:
6      Width/Height      : 1920/1080
7      Pixel Format       : 'NV12' (Y/CbCr 4:2:0)
8      Field              : None
9      Number of planes   : 1
10     Flags               :
11     Colorspace          : Default
12     Transfer Function   : Default
13     YCbCr/HSV Encoding  : Default
14     Quantization        : Full Range
15     Plane 0             :
16         Bytes per Line  : 1920
17         Size Image      : 3110400
18 VIDIOC_G_SELECTION: ok
19 VIDIOC_S_SELECTION: ok
20     VIDIOC_REQBUFS returned 0 (Success)
21     VIDIOC_QUERYBUF returned 0 (Success)
22     VIDIOC_QUERYBUF returned 0 (Success)
23     VIDIOC_QUERYBUF returned 0 (Success)
24     VIDIOC_QUERYBUF returned 0 (Success)
25     VIDIOC_QBUF returned 0 (Success)
26     VIDIOC_QBUF returned 0 (Success)
27     VIDIOC_QBUF returned 0 (Success)
28     VIDIOC_QBUF returned 0 (Success)
29     VIDIOC_STREAMON returned 0 (Success)
30 cap dqbuf: 0 seq:      1 bytesused: 3110400 ts: 1384.549991 (ts-monotonic, ts-src-eof)
31 cap dqbuf: 1 seq:      2 bytesused: 3110400 ts: 1384.616490 delta: 66.499 ms (ts-monotonic, ts-src-eof)
32 cap dqbuf: 2 seq:      3 bytesused: 3110400 ts: 1384.682975 delta: 66.485 ms (ts-monotonic, ts-src-eof)
33 cap dqbuf: 3 seq:      4 bytesused: 3110400 ts: 1384.749486 delta: 66.511 ms (ts-monotonic, ts-src-eof)
34 cap dqbuf: 0 seq:      5 bytesused: 3110400 ts: 1384.816022 delta: 66.536 ms fps: 15.04 (ts-monotonic, ts-src-eof)
35 cap dqbuf: 1 seq:      6 bytesused: 3110400 ts: 1384.882509 delta: 66.487 ms fps: 15.04 (ts-monotonic, ts-src-eof)
36 cap dqbuf: 2 seq:      7 bytesused: 3110400 ts: 1384.949025 delta: 66.516 ms fps: 15.04 (ts-monotonic, ts-src-eof)

```

```

37 cap dqbuf: 3 seq:      8 bytesused: 3110400 ts: 1385.015545 delta: 66.520
   ms fps: 15.04 (ts-monotonic, ts-src-eof)
38 cap dqbuf: 0 seq:      9 bytesused: 3110400 ts: 1385.082051 delta: 66.506
   ms fps: 15.04 (ts-monotonic, ts-src-eof)
39 cap dqbuf: 1 seq:     10 bytesused: 3110400 ts: 1385.148567 delta: 66.516
   ms fps: 15.04 (ts-monotonic, ts-src-eof)
40 cap dqbuf: 2 seq:     11 bytesused: 3110400 ts: 1385.215079 delta: 66.512
   ms fps: 15.04 (ts-monotonic, ts-src-eof)
41 cap dqbuf: 3 seq:     12 bytesused: 3110400 ts: 1385.281594 delta: 66.515
   ms fps: 15.04 (ts-monotonic, ts-src-eof)
42 cap dqbuf: 0 seq:     13 bytesused: 3110400 ts: 1385.348115 delta: 66.521
   ms fps: 15.04 (ts-monotonic, ts-src-eof)
43 cap dqbuf: 1 seq:     14 bytesused: 3110400 ts: 1385.414669 delta: 66.554
   ms fps: 15.03 (ts-monotonic, ts-src-eof)
44 cap dqbuf: 2 seq:     15 bytesused: 3110400 ts: 1385.481133 delta: 66.464
   ms fps: 15.04 (ts-monotonic, ts-src-eof)
45 cap dqbuf: 3 seq:     16 bytesused: 3110400 ts: 1385.547656 delta: 66.523
   ms fps: 15.04 (ts-monotonic, ts-src-eof)
46 cap dqbuf: 0 seq:     17 bytesused: 3110400 ts: 1385.614172 delta: 66.516
   ms fps: 15.04 (ts-monotonic, ts-src-eof)
47 cap dqbuf: 1 seq:     18 bytesused: 3110400 ts: 1385.680680 delta: 66.508
   ms fps: 15.04 (ts-monotonic, ts-src-eof)
48 cap dqbuf: 2 seq:     19 bytesused: 3110400 ts: 1385.747241 delta: 66.561
   ms fps: 15.03 (ts-monotonic, ts-src-eof)
49 cap dqbuf: 3 seq:     20 bytesused: 3110400 ts: 1385.813714 delta: 66.473
   ms fps: 15.03 (ts-monotonic, ts-src-eof)
50 ^C

```

按Ctrl-C停止抓取，视频流保存到文件out.yuv。

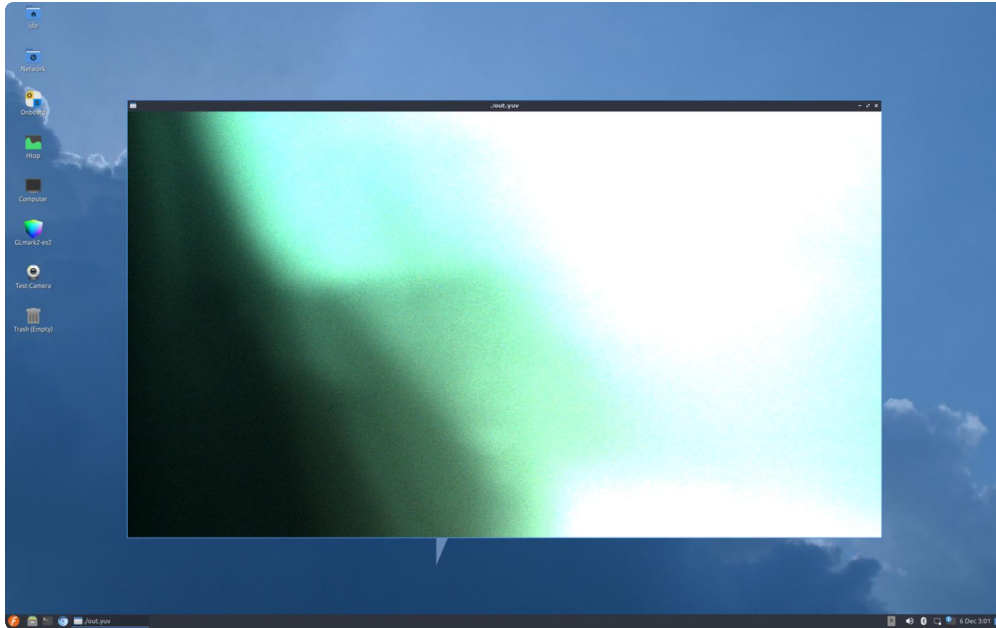
使用ffplay工具播放抓取的视频流，命令如下：

```

1 root@ido:~# ffplay -f rawvideo -video_size 1920x1080 -pix_fmt nv12 ./out.yuv
2 ffplay version 4.2.4-1ubuntu1.0firefly1 Copyright (c) 2003-2020 the FFmpeg
  developers
3   built with gcc 9 (Ubuntu 9.3.0-17ubuntu1~20.04)
4   configuration: --prefix=/usr --extra-version=1ubuntu1.0firefly1 --toolcha
  in=hardened --libdir=/usr/lib/aarch64-linux-gnu --incdir=/usr/include/aar
  ch64-linux-gnu --arch=arm64 --enable-gpl --disable-stripping --enable-avre
  sample --disable-filter=resample --enable-avisynth --enable-gnutls --enabl
  e-ladspa --enable-libaom --enable-libass --enable-libbluray --enable-libbs
  2b --enable-libcaca --enable-libcdio --enable-libcodec2 --enable-libflite
  --enable-libfontconfig --enable-libfreetype --enable-libfribidi --enable-l
  ibgme --enable-libgsm --enable-libjack --enable-libmp3lame --enable-libmys
  ofa --enable-libopenjpeg --enable-libopenmpt --enable-libopus --enable-lib
  pulse --enable-librsvg --enable-librubberband --enable-libshine --enable-l
  ibsnappy --enable-libsoxr --enable-lbspeex --enable-libssh --enable-libth
  eora --enable-libtwolame --enable-libvidstab --enable-libvorbis --enable-l
  ibvpx --enable-libwavpack --enable-libwebp --enable-libx265 --enable-libxm
  l2 --enable-libxvid --enable-libzmq --enable-libzvbi --enable-lv2 --enable
  -omx --enable-openal --enable-openc1 --enable-opengl --enable-sdl2 --enabl
  e-libdc1394 --enable-libdrm --enable-libiec61883 --enable-chromaprint --enab
  le-frei0r --enable-libx264 --enable-libdrm --enable-librga --enable-rkmp
  p --enable-version3 --disable-libopenh264 --disable-vaapi --disable-udpau
  --disable-decoder=h264_v4l2m2m --disable-decoder=vp8_v4l2m2m --disable-dec
  oder=mpeg2_v4l2m2m --disable-decoder=mpeg4_v4l2m2m --disable-muxer='ac3,ea
  c3,mlp,truehd' --disable-encoder='ac3_fixed,ac3,mlp,spdif,truehd' --disabl
  e-demuxer='ac3,eac3,mlp,truehd,dts,dtsd' --disable-parser='aac,ac3,mlp' --
  -disable-decoder='ac3,eac3,mlp,dolby_e' --enable-shared --disable-doc
5   libavutil      56. 31.100 / 56. 31.100
6   libavcodec     58. 54.100 / 58. 54.100
7   libavformat    58. 29.100 / 58. 29.100
8   libavdevice    58.  8.100 / 58.  8.100
9   libavfilter     7. 57.100 / 7. 57.100
10  libavresample   4.  0.  0 / 4.  0.  0
11  libswscale      5.  5.100 / 5.  5.100
12  libswresample   3.  5.100 / 3.  5.100
13  libpostproc    55.  5.100 / 55.  5.100
14 Option -pix_fmt is deprecated, use -pixel_format.
15 libGL error: failed to create dri screen
16 libGL error: failed to load driver: rockchip
17 libGL error: failed to create dri screen
18 libGL error: failed to load driver: rockchip
19 [rawvideo @ 0x7f3c000ba0] Estimating duration from bitrate, this may be in
  accurate
20 Input #0, rawvideo, from './out.yuv':

```

```
21 Duration: 00:00:04.00, start: 0.000000, bitrate: 622075 kb/s
22 Stream #0:0: Video: rawvideo (NV12 / 0x3231564E), nv12, 1920x1080, 622
23 080 kb/s, 25 tbr, 25 tbn, 25 tbc
```



9.1.3 打开摄像头

命令如下：

```
▼ Bash |
1  gst-launch-1.0 v4l2src device=/dev/video0 ! video/x-raw, format=NV12, width
   =1920, height=1080, framerate=15/1 ! autovideoconvert ! autovideosink
```

10 RTC

主板包含2个RTC，其中/dev/rtc0为外部RTC（HYM8563），/dev/rtc1为CPU内部的RTC（RK808）。系统默认使用rtc0的时间。

10.1 获取RTC时间

命令如下：

```
▼ Bash |
1  root@ido:~# hwclock
2  2022-11-10 02:16:23.617474+00:00
```

10.2 设置RTC时间

```
1 root@ido:~# hwclock
2 2022-11-18 08:30:40.778874+00:00
3 root@ido:~# date -s '2022-11-10 10:17:00'
4 Thu Nov 10 10:17:00 UTC 2022
5 root@ido:~# hwclock -w
6 root@ido:~# hwclock
7 2022-11-18 08:31:06.829691+00:00
8 root@ido:~#
```

注：如果接入网络的话可能会设置失败，因为网络每隔一段时间会直接获取最新时间。

10.3 RTC定时开机

CPU内部的rtc1节点支持定时开机功能。

以下测试关机60秒定时开机，命令如下：

```
1 root@ido:~# echo +60 > /sys/class/rtc/rtc1/wakealarm
2 root@ido:~# poweroff
3          Stopping Session c1 of user ido.
4          Stopping Session c2 of user root.
5 ▾ [ OK ] Removed slice system-modprobe.slice.
6 ▾ [ OK ] Stopped target Bluetooth.
7 ...
8 ▾ [ OK ] Reached target Shutdown.
9 ▾ [ OK ] Reached target Final Step.
10 ▾ [ OK ] Finished Power-Off.
11 ▾ [ OK ] Reached target Power-Off.
12 ▾ [ 1195.829009] Local MAC: 82:4c:21:62:f5:35
13 ▾ [ 1195.834186] set rk809 rtc alarm!
14 ▾ [ 1195.837515] reboot: Power down
15
```

等待60秒后，主板会开机。

11 开机自启动

默认系统开机后会运行/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

11.1 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，命令如下：

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

11.2 屏幕旋转

使用xrandr工具可以实现屏幕的旋转。

11.2.1 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，命令如下：

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

11.2.2 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，命令如下：

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

这样修改后，每次重启设备，桌面将旋转180度。

13 按键

主板共配置3个按键：

标记	名称	功能
SW1	Recovery 键	用于进入烧录模式，上报键值：KEY_VOLUMEUP
SW2	Reset键	硬件复位
SW3	电源键	用于开关机、待机唤醒，上报键值：KEY_POWER

