

IDO-SBC3528-V1 Linux4.19使用手册

IDO-SBC3528-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

3 CAN

2.1 测试

3 USB

2.1 电源控制

2.2 U盘的挂载

2.3 USB OTG 切换命令

3 MicroSD

4 以太网

4.1 测试

5 Lineout

5.1 查看声卡设备

5.2 播放音频

5.3 音量调节

6 MIC

6.1 录音测试

7 WiFi

7.1 连接WiFi热点

7.2 WiFi上网测试

8 Bluetooth

8.1 连接蓝牙设备

9 4G/5G

9.1 4G/5G 拨号	
9.2 4G/5G 上网测试	
10 摄像头	
10.1 测试摄像头是否存在	
10.2 预览摄像头画面	
10.2.1 cheese预览	
10.2.2 gstreamer预览	
11 红外	
11.1 查看红外按键值	
12 ADC	
12.1 计算输入电压	
13 PCIE	
14 DIO	
14.1 DI的使用	
14.2 DO的使用	
15 外部看门狗	
15.1 开启与关闭	
15.2 测试	
16 FAN	
16.1 开启与关闭	
17 LED	
18 RTC	
18.1 读取与设置时间	
18.2 定时开机	
19 输入电压检测	
IDO-SBC3528-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	

3 CAN

2.1 测试

3 USB

2.1 电源控制

2.2 U盘的挂载

2.3 USB OTG 切换命令

3 MicroSD

4 以太网

4.1 测试

5 Lineout

5.1 查看声卡设备

5.2 播放音频

6 MIC

6.1 录音测试

7 WiFi

7.1 连接WiFi热点

7.1.1 命令行连接

7.1.2 桌面连接

8 Bluetooth

8.1 连接蓝牙设备

9 4G/5G

9.1 4G/5G 拨号

9.2 4G/5G 上网测试

10 摄像头

10.1 测试摄像头是否存在

10.2 预览摄像头画面

11 红外

11.1 查看红外按键值

12 ADC

12.1 计算输入电压

13 PCIE

14 DIDO

14.1 DI的使用

14.2 DO的使用

15 外部看门狗

15.1 开启与关闭

15.2 测试

16 FAN

16.1 开启与关闭

17 LED

18 RTC

18.1 读取与设置时间

18.2 定时开机

19 输入电压检测

IDO-SBC3528-V1 Buildroot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

3 CAN

2.1 测试

3 USB

2.1 电源控制

2.2 U盘的挂载

2.3 USB OTG 切换命令

3 MicroSD

4 以太网

4.1 测试

5 Lineout

5.1 查看声卡设备

5.2 播放音频

5.2.1 播放到HDMI

5.2.2 播放到Lineout

5.2.3 播放到耳机

6 MIC

6.1 录音测试

7 WiFi

7.1 连接WiFi热点

8 Bluetooth

8.1 连接蓝牙设备

9 4G/5G

9.1 4G/5G 拨号

9.2 4G/5G 上网测试

10 摄像头

10.1 测试摄像头是否存在

10.2 预览摄像头画面

11 红外

11.1 查看红外按键值

12 ADC

12.1 计算输入电压

13 PCIE

14 DIO

14.1 DI的使用

14.2 DO的使用

15 外部看门狗

15.1 开启与关闭

15.2 测试

16 FAN

16.1 开启与关闭

17 LED

18 RTC

18.1 读取与设置时间

18.2 定时开机

19 输入电压检测

IDO-SBC3528-V1

Linux4.19使用说明

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	V1D	1.创建文档；	TWX	IDO	2024/09/24
V1.1	V1D	1.增加Buildroot系统；	TWX	IDO	2024/09/26

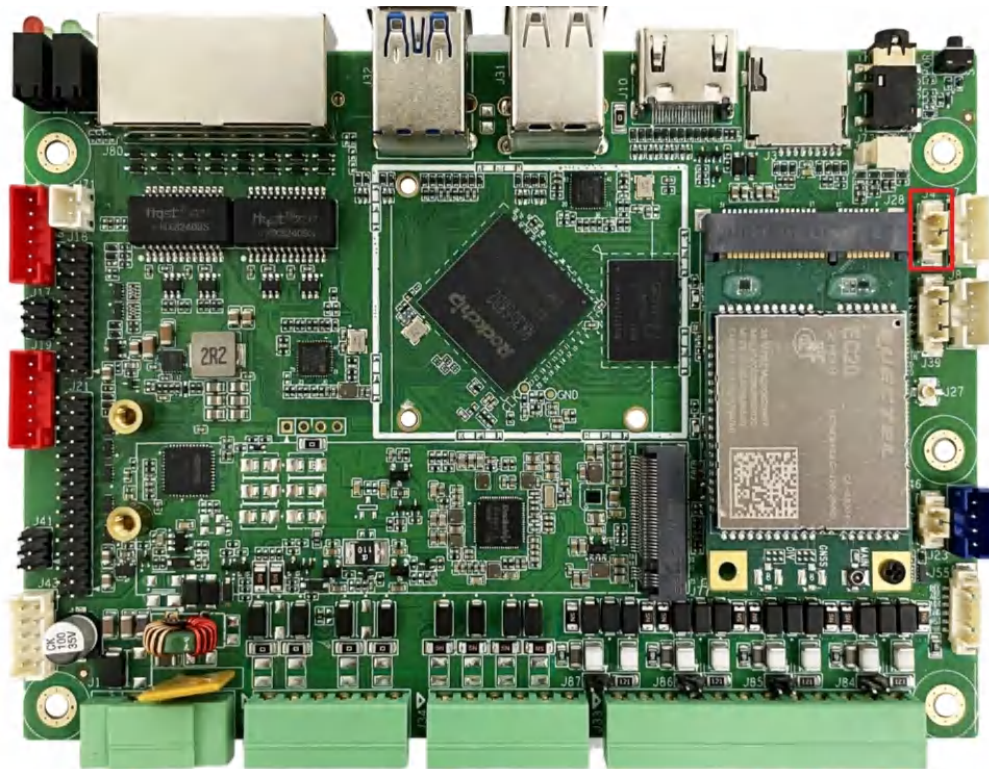
V1.2	V1D	1.修正gstreamer预览摄像头命令；	TWX	IDO	2024/09/27
V1.3	V1D	1.修正OTG host切device指令； 2.修正ADC图片； 3.修正Ubuntu & Buildroot标题；	TWX	IDO	2024/11/18
V1.4	V1D	1.修正debian ssh登录账号密码；	TWX	IDO	2024/11/20

IDO–SBC3528–V1 Debian系统

1 调试

1.1 串口调试

串口调试端口位于J4，通信参数为1500000 8 N 1，电平类型是TTL电平。



默认串口登录账户：linaro ， 密码： linaro。

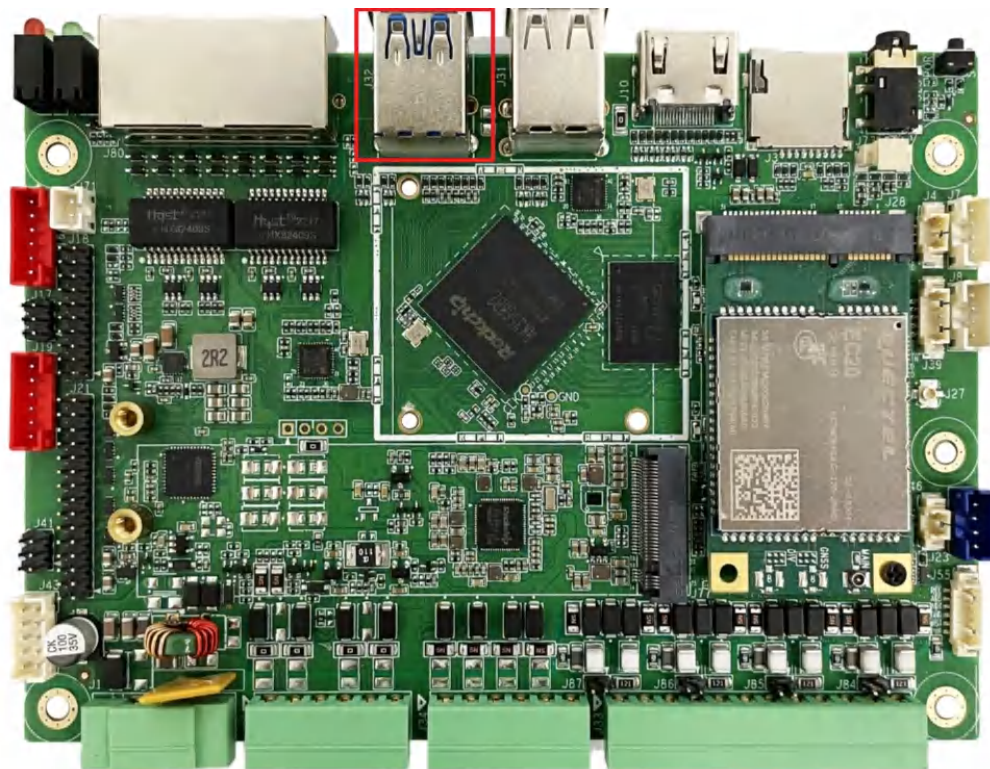
```
linaro-alip login: linaro
密码:
上一次登录: 三 1月 1 12:08:46 UTC 2020ttyFIQ0 上
Linux linaro-alip 4.19.219 #155 SMP Wed Jan 11 13:56:54 CST 2023 aarch64

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:~$
```

1.2 ADB调试

adb调试端口位于J32下方的USB口，即与烧录端口为同一端口。



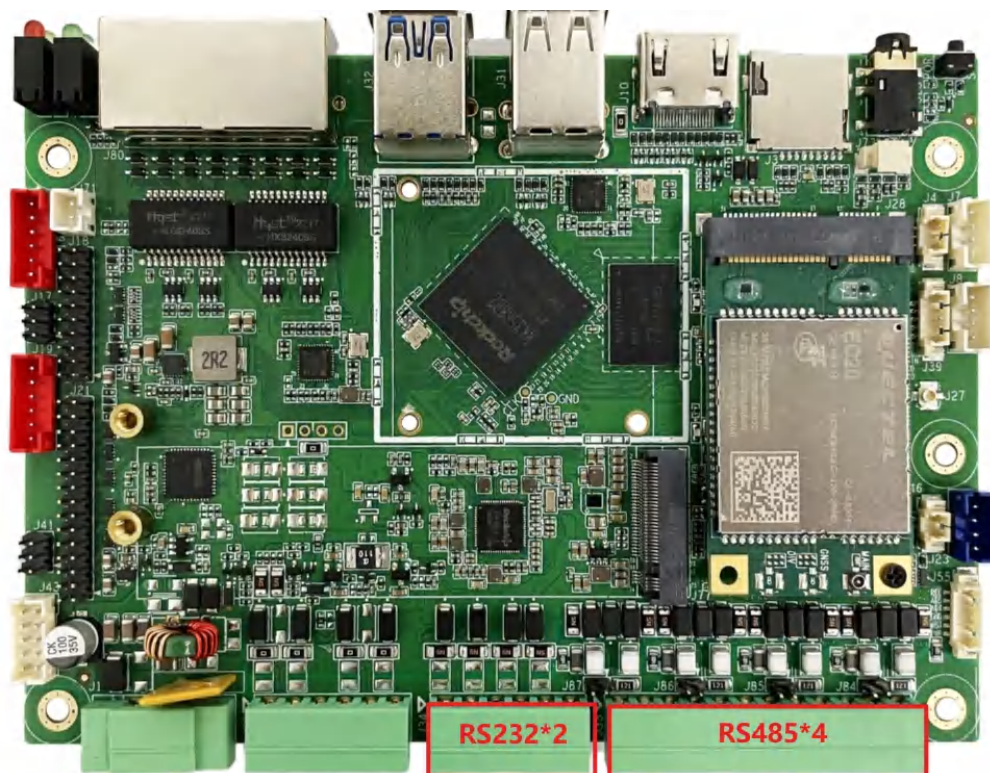
```
D:\>adb shell
* daemon not running. starting it now on port 5037 *
* daemon started successfully *
root@linaro-alip:/# ls
ls
bin  etc      md5sum.txt  opt      run      sys      userdata
boot home   media      proc     sbin     system  usr
data lib    mnt        rockchip-test  sdcard  tmp      var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

1.3 SSH调试

SSH远程登录账号：linaro，密码：linaro。

2 串口

主板共配置6路串口（调试串口除外），其中4路RS485，2路RS232。



各个串口对应的设备节点如下表：

编号	设备节点	类型	位置
1	/dev/ttyS0	RS485	J33,R0-T0
2	/dev/ttyS3	RS485	J33,R3-T3
3	/dev/ttyS4	RS485	J33,,R4-T4
4	/dev/ttyS5	RS485	J33,R5-T5
5	/dev/ttyS7	RS232	J34,R7-T7
6	/dev/ttyS9	RS232	J34,R9-T9

2.1 测试

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：

```

1  linaro@linaro-alip:~$ sudo apt-get update
2  linaro@linaro-alip:~$ sudo apt-get install microcom

```

以测试/dev/ttyS0为例：

```

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。

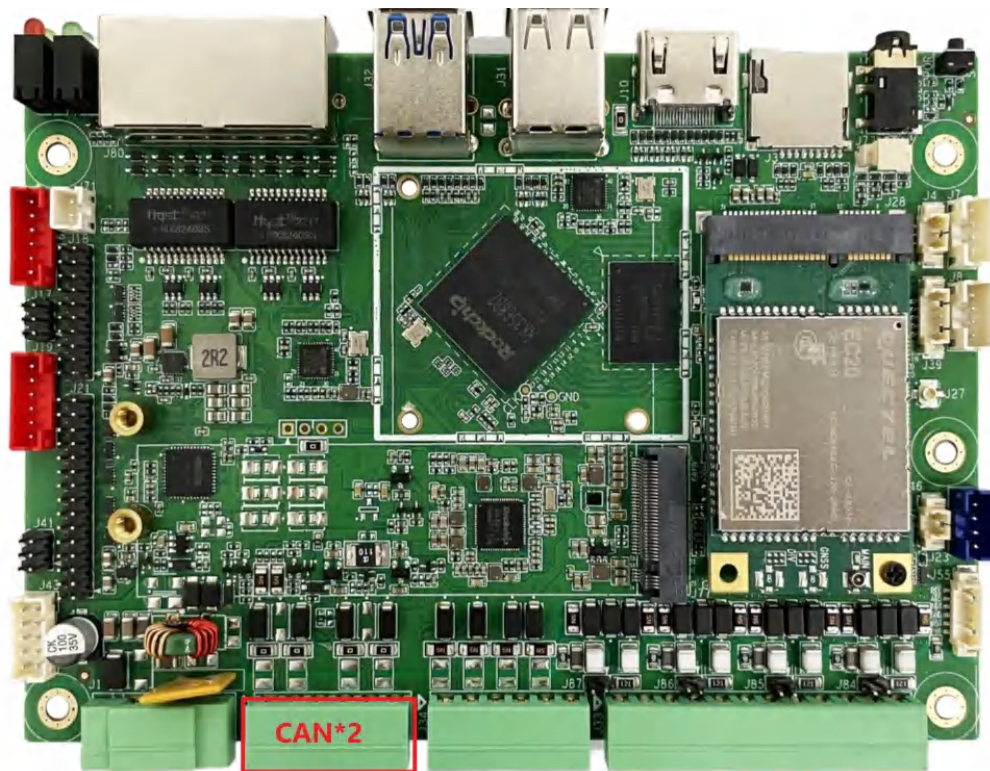
```

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 CAN

主板共配置2路CAN接口，系统中对应的网络节点分别为can0和can1（与DIDO接口复用，二选一）。



2.1 测试

打开can设备，并设置通信波特率为250Kbps：

```
1  sudo ip link set can0 down
2  sudo ip link set can0 type can bitrate 250000
3  sudo ip link set can0 up
4
5  sudo ip link set can1 down
6  sudo ip link set can1 type can bitrate 250000
7  sudo ip link set can1 up
```

发送测试：

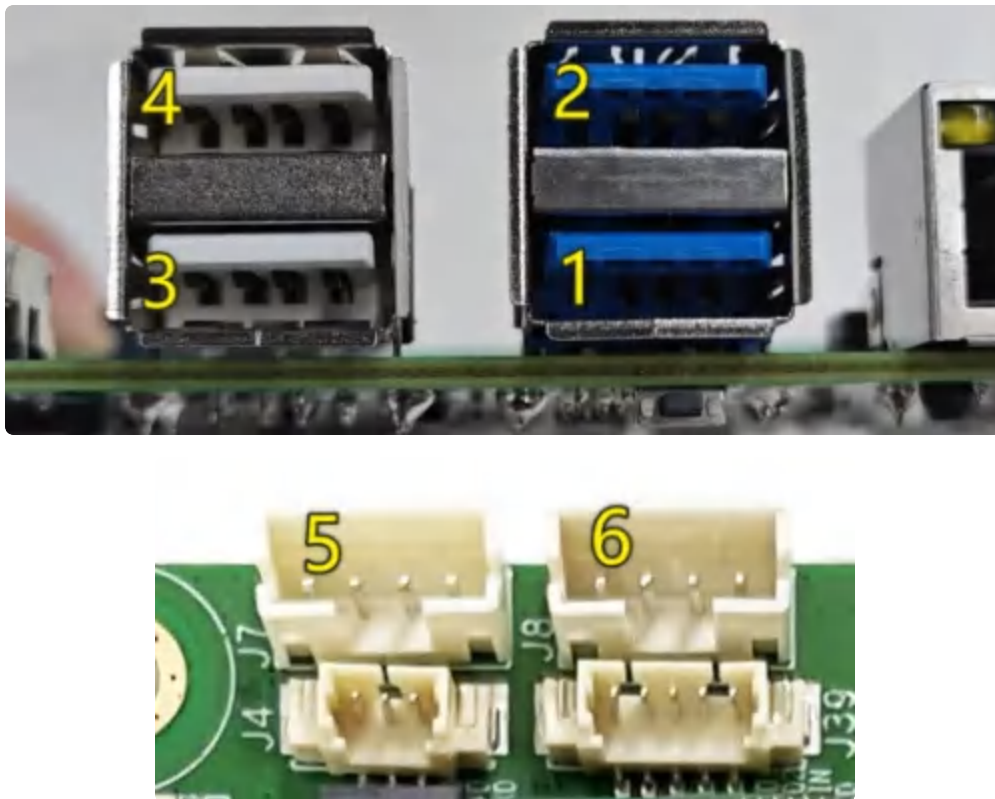
```
1  sudo cansend can0 123#DEADBEEF
```

接收测试：

```
1 sudo candump can0
```

3 USB

主板共配置6路USB接口，这里分别标记USB1-6，其中2路USB3.0（USB1和USB2），4路USB2.0（USB3-6），各个USB口位置如下图所示：



2.1 电源控制

USB2-6支持软件控制电压输出，控制节点见下表：

序号	功能	控电节点
2	USB 3.0 HOST	/sys/class/leds/usb3_host_pwr/brightness
3	USB 2.0 HOST	/sys/class/leds/usb2_host_pwr/brightness
4	USB 2.0 HOST	/sys/class/leds/usb2_fe2_pwr/brightness
5	USB 2.0 HOST	/sys/class/leds/usb2_fe3_pwr/brightness

6	USB 2.0 HOST	/sys/class/leds/usb2_fe4_pwr/brightness
---	--------------	---

供电控制说明，序号1写"1"关闭电源，写"0"开启电源；序号2-序号6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，以序号6为例

```

1  #关闭
2  echo 0 > /sys/class/leds/usb2_fe4_pwr/brightness
3  #开启（默认状态）
4  echo 1 > /sys/class/leds/usb2_fe4_pwr/brightness

```

2.2 U盘的挂载

USB端口插上U盘，会默认挂载到/media/linaro/XXX目录下：

```

1  root@linaro-alip:/home/linaro# mount
2  ...
3  /dev/sda1 on /media/linaro/KINGSTON type vfat ...

```

2.3 USB OTG 切换命令

USB OTG 支持host 和device 模式的切换，软件切换方法如下

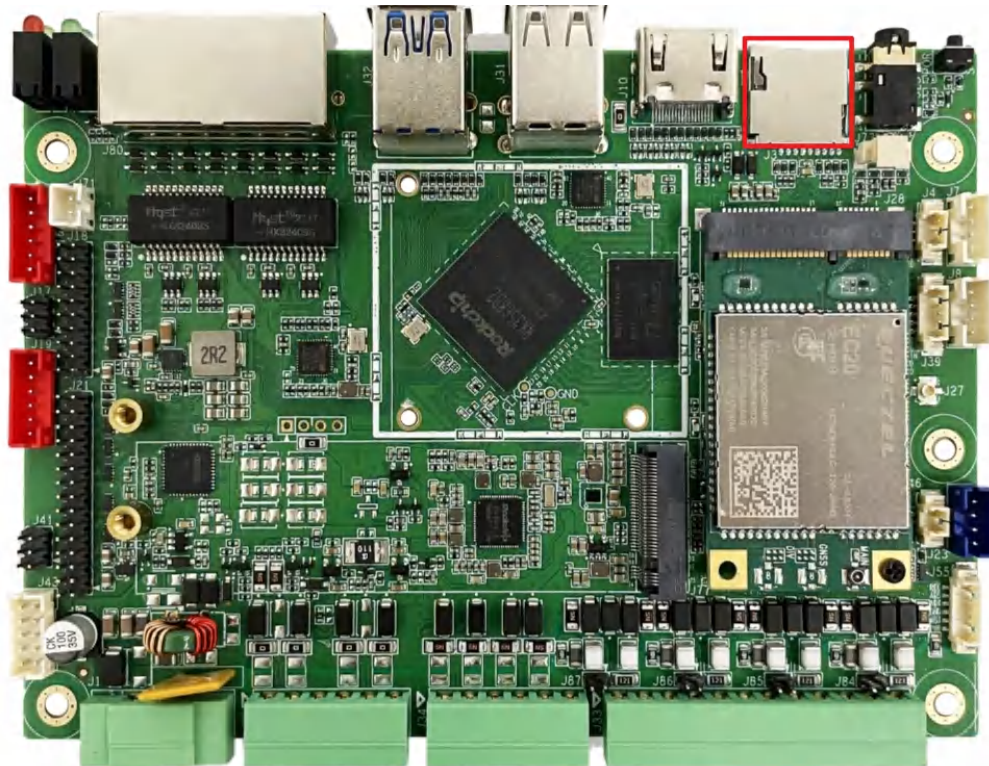
```

1  //切换HOST模式
2  echo host > /sys/devices/platform/fe8a0000.usb2-phy/otg_mode
3  echo 255 > /sys/class/leds/usb3_otg_pwr/brightness
4
5  //切换DEVICE模式
6  echo peripheral > /sys/devices/platform/fe8a0000.usb2-phy/otg_mode
7  echo 0 > /sys/class/leds/usb3_otg_pwr/brightness
8  /usr/bin/usbdevice update

```

3 MicroSD

主板共配置1个SD接口：

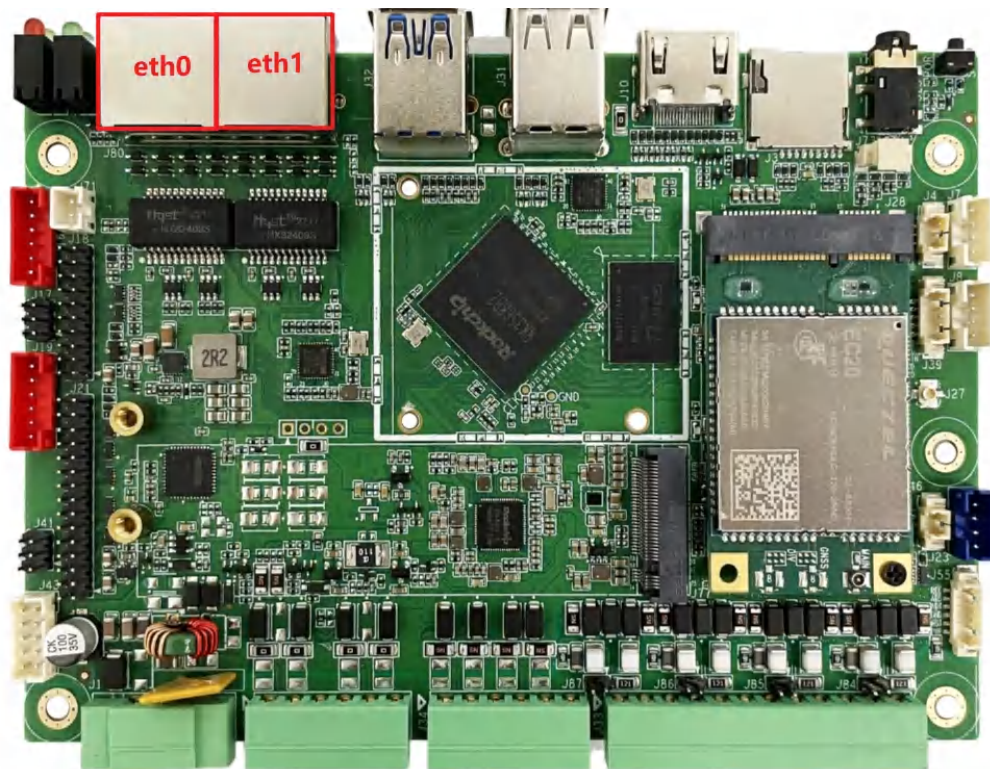


插入SD卡后，自动挂载到/media/ido/xxx目录下，可以使用mount命令确认：

```
1  linaro@linaro-alip:~$ mount
2  ...
3  /dev/mmcblk1p1 on /media/linaro/0403-0201 type vfat
```

4 以太网

主板共配置2路1000M以太网接口，系统中对应的网络节点分别为eth0和eth1：



4.1 测试

以下以eth1为例（eth0同样的操作方法）。

查看eth1的ip:

```
▼ Bash |
1  linaro@linaro-alip:~$ sudo ifconfig eth1
2  eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
3      inet 192.168.1.177  netmask 255.255.255.0  broadcast 192.168.1.255
4      inet6 fe80::4f32:fc2d:eb30:220b  prefixlen 64  scopeid 0x20<link>
5      ether d2:e1:8d:d5:f7:49  txqueuelen 1000  (Ethernet)
6      RX packets 78  bytes 11727 (11.4 KiB)
7      RX errors 0  dropped 0  overruns 0  frame 0
8      TX packets 60  bytes 5822 (5.6 KiB)
9      TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0
10     device interrupt 52
11
```

设置eth1的ip:

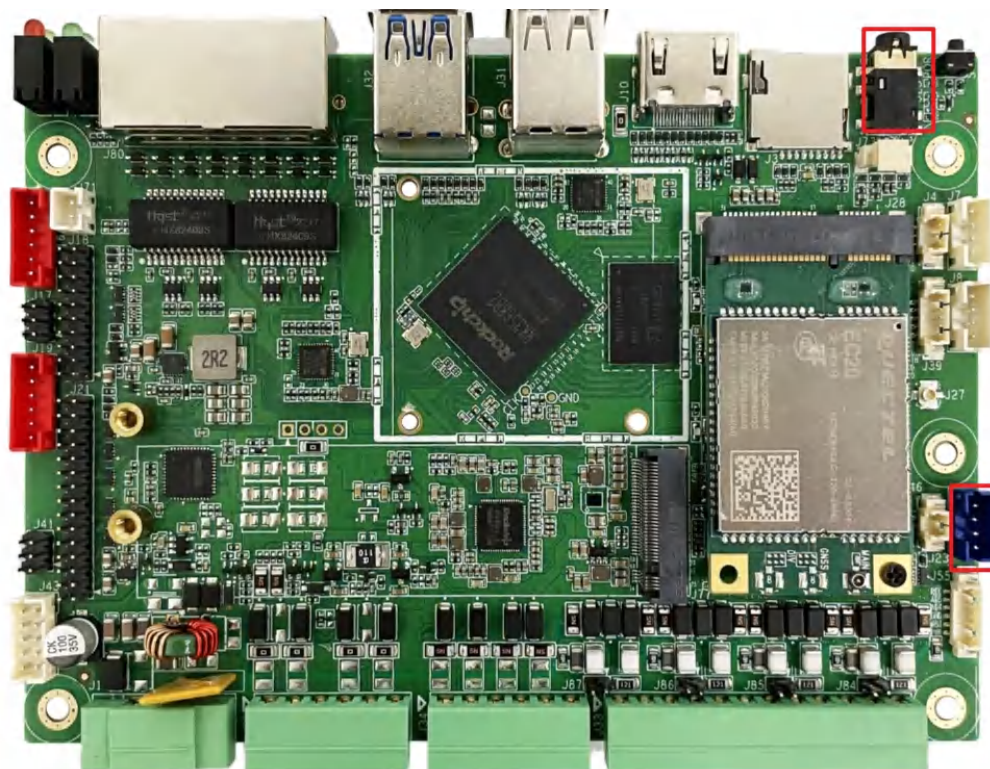
```
1  linaro@linaro-alip:~$ sudo ifconfig eth1 192.168.1.123
2  linaro@linaro-alip:~$
3
```

eth1 ping测试:

```
1  linaro@linaro-alip:~$ sudo ping www.baidu.com -I eth1
2  PING www.a.shifen.com (14.215.177.39) from 192.168.1.177 eth1: 56(84) byte
   s of data.
3  64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=1 ttl=54 time=6.75 ms
4  64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=2 ttl=54 time=6.32 ms
5  64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=3 ttl=54 time=7.01 ms
6  64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=4 ttl=54 time=7.47 ms
7  64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=5 ttl=54 time=7.01 ms
```

5 Lineout

主板支持LineOut音频输出和HDMI音频输出，LineOut音频接口位于J32：



注：默认LineOut和耳机的音频输出是同一路，即LineOut播放同时耳机也会播放。

5.1 查看声卡设备

▼

Bash

```
1  linaro@linaro-alip:~$ aplay -l
2  **** List of PLAYBACK Hardware Devices ****
3  card 0: rockchiphdmi [rockchip,hdmi], device 0: rockchip,hdmi i2s-hifi-0 [r
   rockchip,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  linaro@linaro-alip:~$
```

5.2 播放音频

播放到HDMI:

▼

Bash

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

播放到耳机:

插入耳机，执行以下命令。

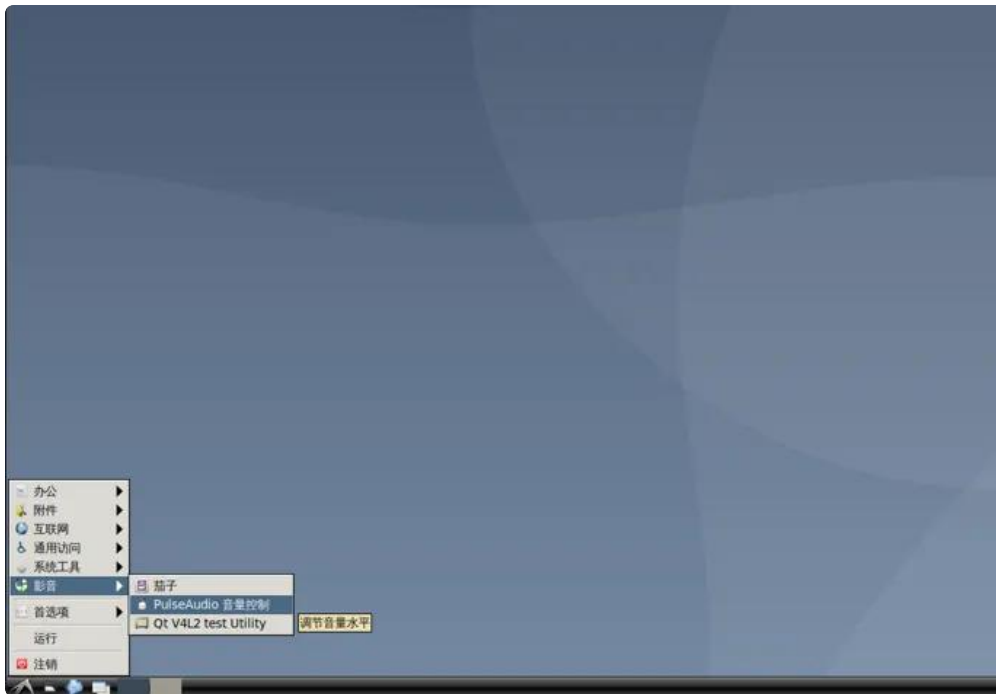
▼

Bash

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

5.3 音量调节

桌面左下角打开PulseAudio程序：



弹出的窗口中，在回放页面中设置系统音量：

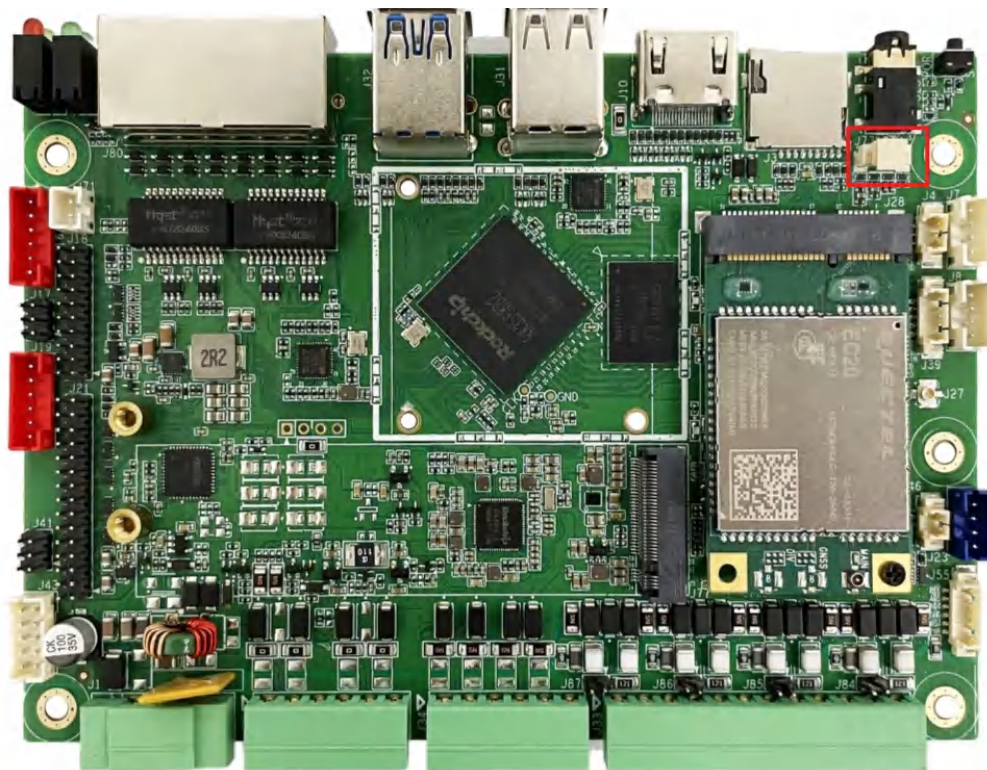


当需要静音时，点击静音按钮即可：



6 MIC

主板配置一路模拟麦克风接口，位于J7：



6.1 录音测试

使用arecord工具可以进行录音测试：

```

1  linaro@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
4  ^CAborted by signal 中断...
5  linaro@linaro-alip:~$

```

录音完后播放测试：

```

1  linaro@linaro-alip:~$ aplay -D plughw:1,0 ./test.wav
2  Playing WAVE './test.wav' : Signed 16 bit Little Endian, Rate 48000 Hz, Stereo

```

7 WiFi

主板配置了一路2.4G WiFi模块，型号为AW-NM372SM。

系统上电默认会打开WiFi，对应的网络设备节点为wlan0。

```

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

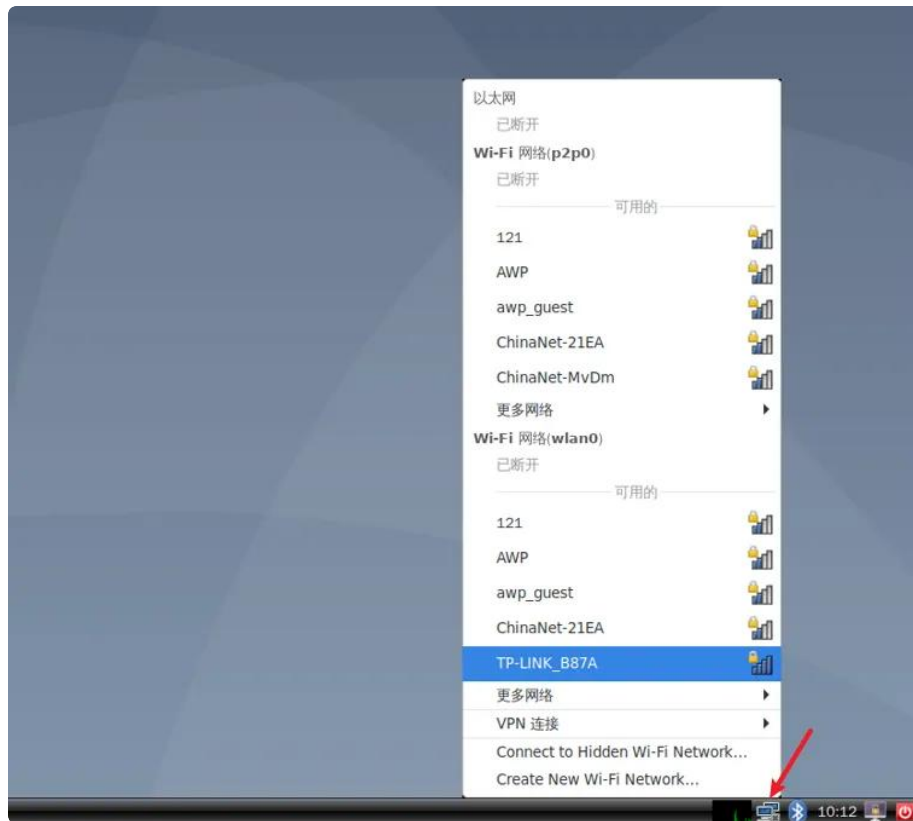
```

7.1 连接WiFi热点

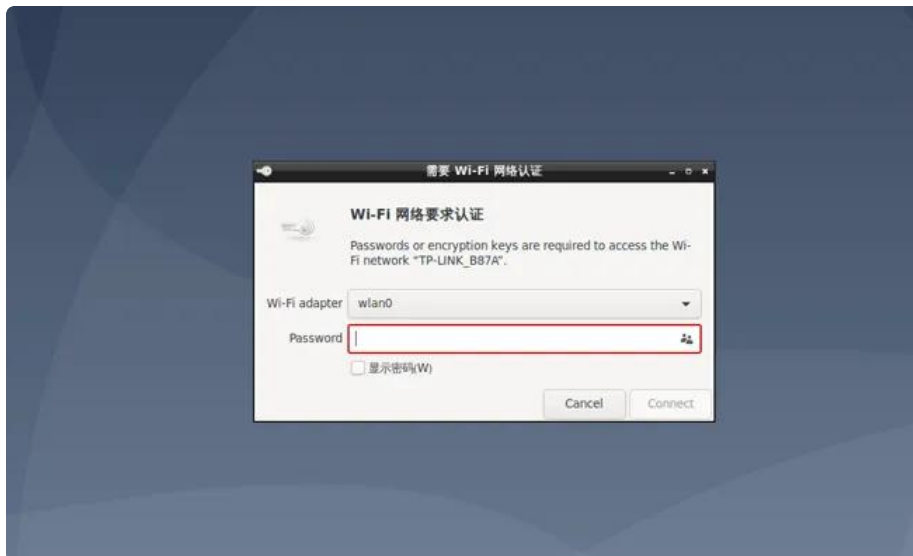
连接热点有2种方式，可以在桌面上操作，也可以使用命令行操作。

方式一

点击桌面右下角的网络图标，即可看到WiFi热点列表：



点击要连接的热点，弹出密码输入窗口：



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。

点击桌面左下角的应用图标，然后依次点击【通用访问】->【Onboard】：



使用软键盘输入密码后，点击【Connect】连接热点：



连接成功后，桌面右下角的网络图标将改变：



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:

```

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

```

7.2 WiFi上网测试

使用ping命令来进行上网测试：

```

1  linaro@linaro-alip:~$ sudo ping www.baidu.com -I wlan0

```

8 Bluetooth

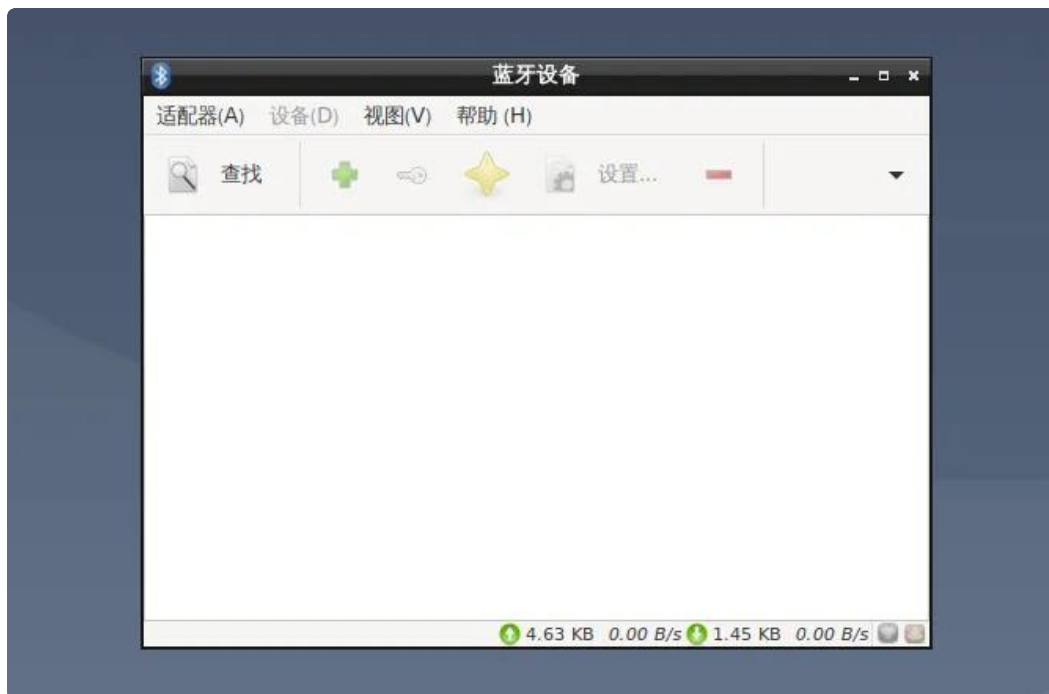
系统开机默认打开蓝牙，对应的网络节点为hci0。

```

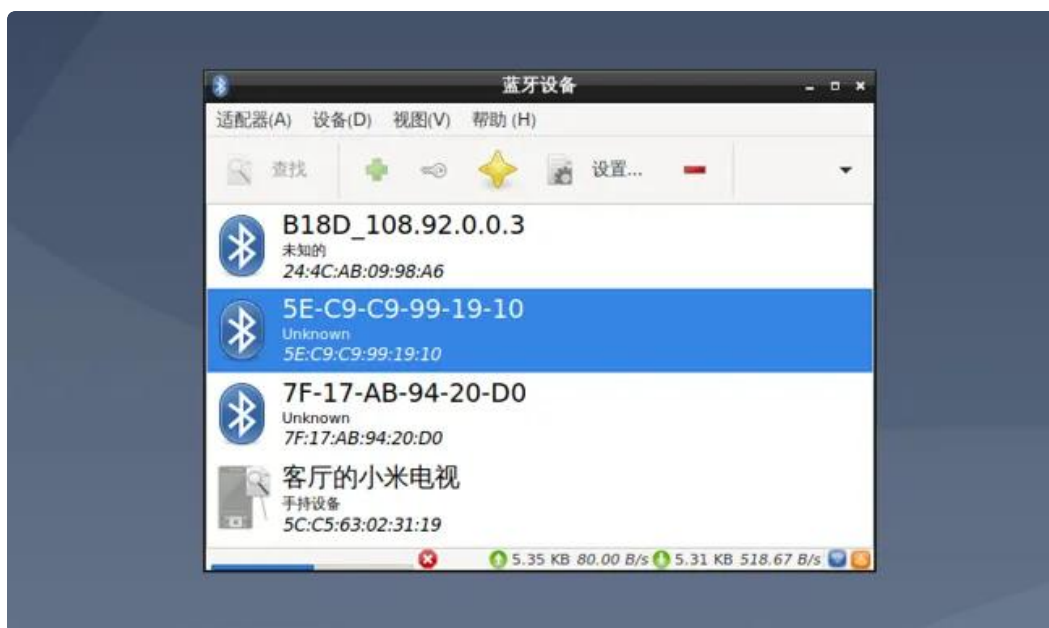
1  linaro@linaro-alip:~$ hciconfig
2  hci0:  Type: Primary  Bus: UART
3      BD Address: 2C:D2:6B:11:AC:77 ACL MTU: 1021:8 SCO MTU: 255:12
4      UP RUNNING
5      RX bytes:1531 acl:0 sco:0 events:51 errors:0
6      TX bytes:5012 acl:0 sco:0 commands:51 errors:0
7
8  linaro@linaro-alip:~$

```

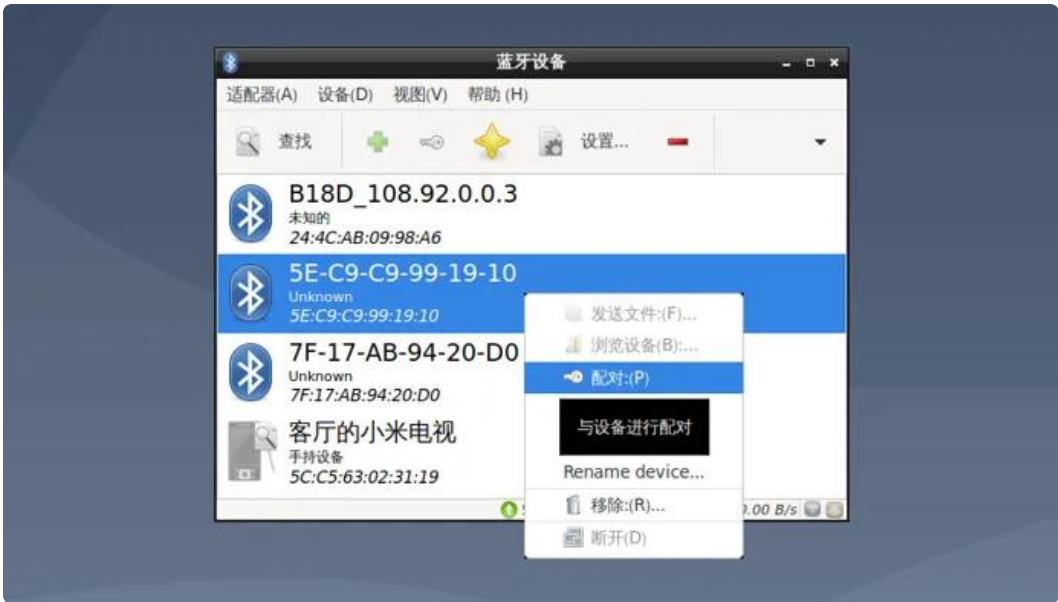
8.1 连接蓝牙设备



将看到附近的蓝牙设备列表：



选中要连接设备，右键->配对，即可连接该设备：



9 4G/5G

主板配置了一路4G/5G接口，默认支持EC20-CN（4G）和EC200U-CN（5G）。

序号	模块名称	说明
1	EC20	4G LTE
2	RG200U-CN	支持 5G NSA 和 SA 模式，支持 TDD 和 FDD 两种模式，向下兼容 4G/3G。

9.1 4G/5G 拨号

上电前，正确安装模块和SIM卡，上电后，系统会自动进行拨号上网。

拨号成功，则会产生wwan0/usb0网络节点：

```
1  linaro@linaro-alip:~$ sudo ifconfig wwan0
2  wwan0: flags=193<UP,RUNNING,NOARP> mtu 1500
3      inet 10.101.61.51 netmask 255.255.255.248
4      inet6 fe80::fc:f6ff:fe8d:bab6 prefixlen 64 scopeid 0x20<link>
5      ether 02:fc:f6:8d:ba:b6 txqueuelen 1000 (Ethernet)
6      RX packets 42 bytes 7013 (6.8 KiB)
7      RX errors 0 dropped 0 overruns 0 frame 0
8      TX packets 57 bytes 4608 (4.5 KiB)
9      TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
10
11  linaro@linaro-alip:~$
```

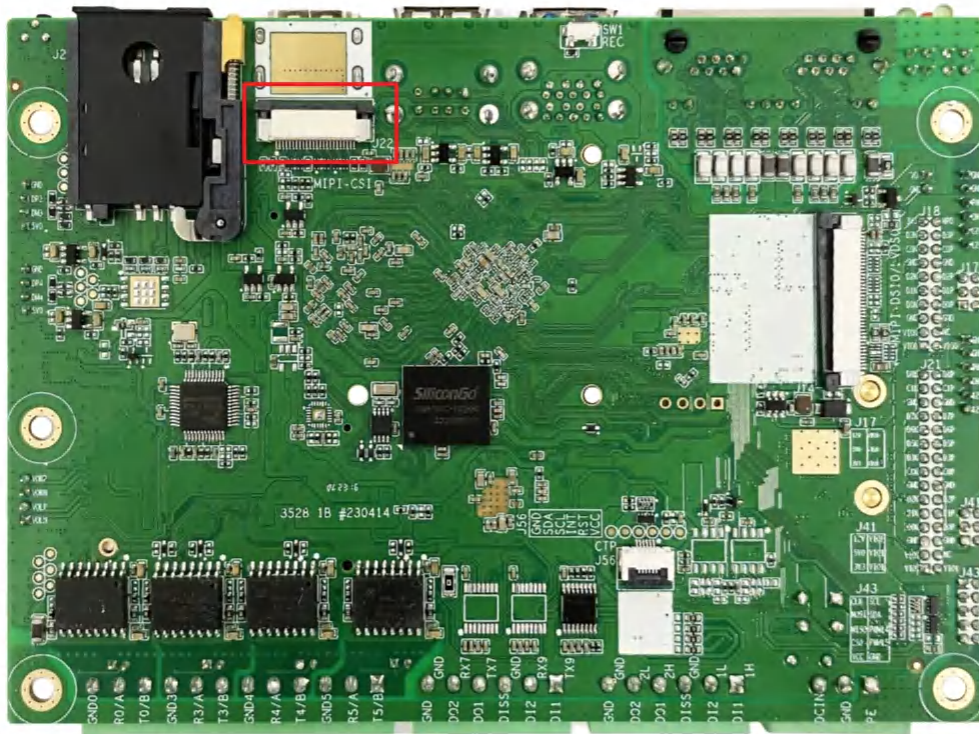
9.2 4G/5G 上网测试

使用ping命令测试4G/5G上网功能是否正常：

```
1  linaro@linaro-alip:~$ sudo ping www.baidu.com -I wwan0
2  PING www.a.shifen.com (183.232.231.174) from 10.101.61.51 wwan0: 56(84) bytes of data.
3  64 bytes from 183.232.231.174 (183.232.231.174): icmp_seq=1 ttl=55 time=33.3 ms
4  64 bytes from 183.232.231.174 (183.232.231.174): icmp_seq=2 ttl=55 time=48.1 ms
5  64 bytes from 183.232.231.174 (183.232.231.174): icmp_seq=3 ttl=55 time=46.2 ms
6  64 bytes from 183.232.231.174 (183.232.231.174): icmp_seq=4 ttl=55 time=45.2 ms
7  64 bytes from 183.232.231.174 (183.232.231.174): icmp_seq=5 ttl=55 time=42.1 ms
```

10 摄像头

主板配置了一路CSI摄像头，默认支持型号OV5648和OV8858，位于J22：



10.1 测试摄像头是否存在

```
▼ Bash |
1  linaro@linaro-alip:~$ sudo 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]
```

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

10.2 预览摄像头画面

使用cheese预览或者使用gststreamer预览。

10.2.1 cheese预览

鼠标依次点击应用->影音->茄子：

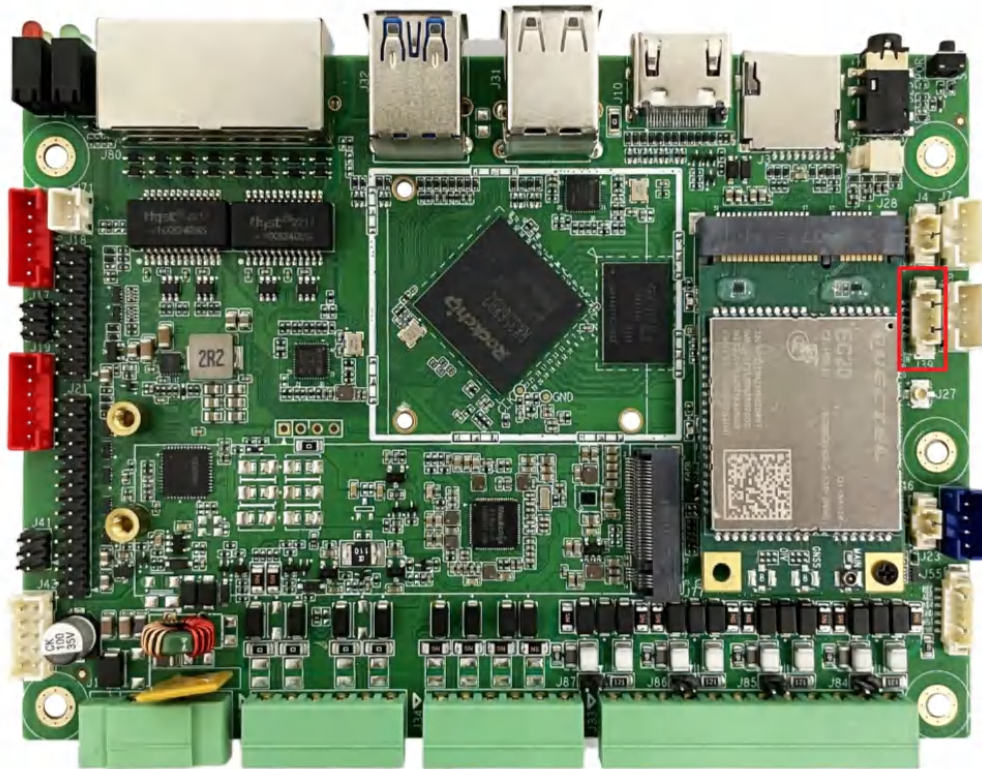


10.2.2 gststreamer预览

```
linaro@linaro-alip:~$ gst-launch-1.0 v4l2src device=/dev/video0 ! video/x-raw, format=NV12, width=1920, height=1080, framerate=15/1 ! autovideoconvert ! autovideosink
```

11 红外

主板配置了一路红外接口，位于J39，支持NEC编码遥控器，默认适配的遥控器型号为HTR-A07，使用时需要安装好配套的红外扩展小板。





HTR-A07的键值如下表所示：

编号	按键	键值	编号	按键	键值
1	电源	KEY_POWER	21	1	KEY_1
2	TV	KEY_SCREEN	22	2	KEY_2
3	橙色	KEY_F1	23	3	KEY_3
4	绿色	KEY_F2	24	4	KEY_4
5	黄色	KEY_F3	25	5	KEY_5
6	紫色	KEY_F4	26	6	KEY_6
7	音量+	KEY_VOLUMEUP	27	7	KEY_7

8	音量-	KEY_VOLUMEDOWN	28	8	KEY_8
9	屏显	KEY_DISPLAY_OFF	29	9	KEY_9
10	静音	KEY_MUTE	30	TVNOW	KEY_DOT
11	上一节目	KEY_VIDEO_PREV	31	0	KEY_0
12	下一节目	KEY_VIDEO_NEXT	32	截屏	KEY_PRINT
13	上	KEY_UP	33	/	/
14	左	KEY_LEFT	34	/	/
15	下	KEY_DOWN	35	/	/
16	右	KEY_RIGHT	36	/	/
17	确认	KEY_ENTER	37	/	/
18	返回	KEY_BACK	38	/	/
19	主页	KEY_HOME	39	/	/
20	菜单	KEY_MENU	40	/	/

11.1 查看红外按键值

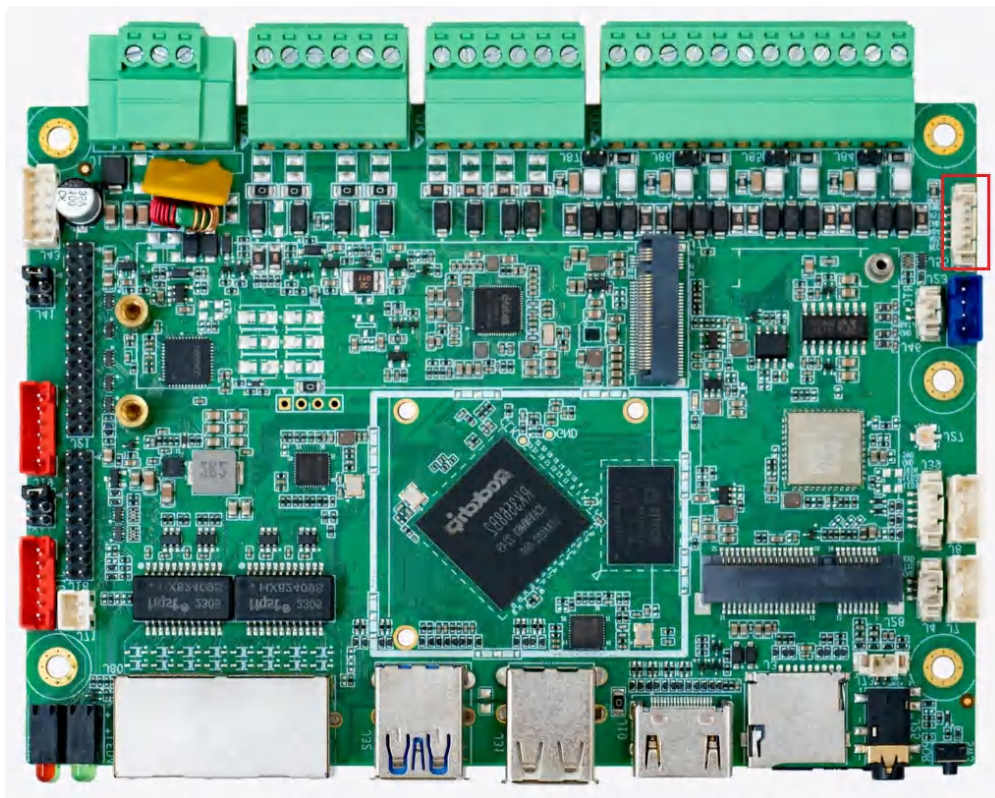
▼

Bash

```
1 echo 1 > /sys/module/rockchip_pwm_remotectl/parameters/code_print
```

12 ADC

主板共配置2路ADC，位于J55，丝印标记为AD4和AD5。



12.1 计算输入电压

使用如下公式计算输入电压：

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

raw为从设备节点读取的数值，两路AD对应的设备节点见下表：

AD4	/sys/bus/iio/devices/iio:device0/in_voltage4_raw
AD5	/sys/bus/iio/devices/iio:device0/in_voltage5_raw

举例：

AD4输入电压计算

```
▼ Bash |
1 cat /sys/bus/iio/devices/iio:device0/in_voltage4_raw
2 258
```

则AD4的输入电压 $V=(258/1024)*1.8\text{v}=0.454\text{v}$ 。

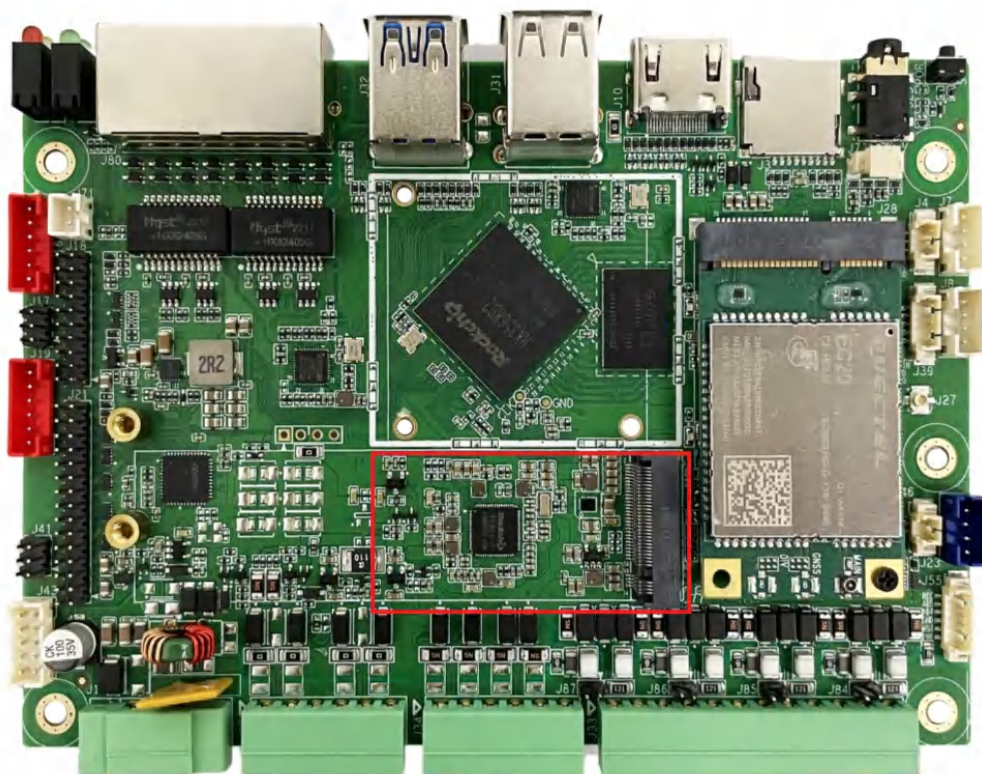
AD5输入电压计算：

```
1 cat /sys/bus/iio/devices/iio:device0/in_voltage5_raw
2 137
```

则AD5的输入电压 $V=(137/1024)*1.8v=0.241v$ 。

13 PCIE

主板配置了一路PCIE3.0接口，位于J77：



断电安装好PCIE固态硬盘，主板上电，系统启动后，硬盘将自动识别：

```

1 [root@RK356X:/]# fdisk -l
2 Found valid GPT with protective MBR; using GPT
3
4 Disk /dev/mmcblk0: 30535680 sectors, 2622M
5 Logical sector size: 512
6 Disk identifier (GUID): c7460000-0000-4b06-8000-5539000009bc
7 Partition table holds up to 128 entries
8 First usable sector is 34, last usable sector is 30535646
9
10 Number  Start (sector)    End (sector)  Size Name
11      1           16384           28671 6144K uboot
12      2           28672           36863 4096K misc
13      3           36864          102399 32.0M boot
14      4          102400          167935 32.0M recovery
15      5          167936          233471 32.0M backup
16      6          233472          757759 256M userdata
17      7          757760          1019903 128M oem
18      8          1019904          30535615 14.0G rootfs
19 Disk /dev/nvme0n1: 119 GB, 128035676160 bytes, 250069680 sectors
20 15566 cylinders, 255 heads, 63 sectors/track
21 Units: cylinders of 16065 * 512 = 8225280 bytes
22
23 Device      Boot StartCHS      EndCHS          StartLBA      EndLBA      Sector
24 s  Size Id Type
25 /dev/nvme0n1p1  1,0,1      1023,254,63      2048  250069646  25006759
26 9 119G  c Win95 FAT32 (LBA)
27 [root@RK356X:/]#

```

可以看到PCIE硬盘在系统中的设备节点为/dev/nvme0n1p1。

使用mount命令可以挂载到系统中进行文件操作：

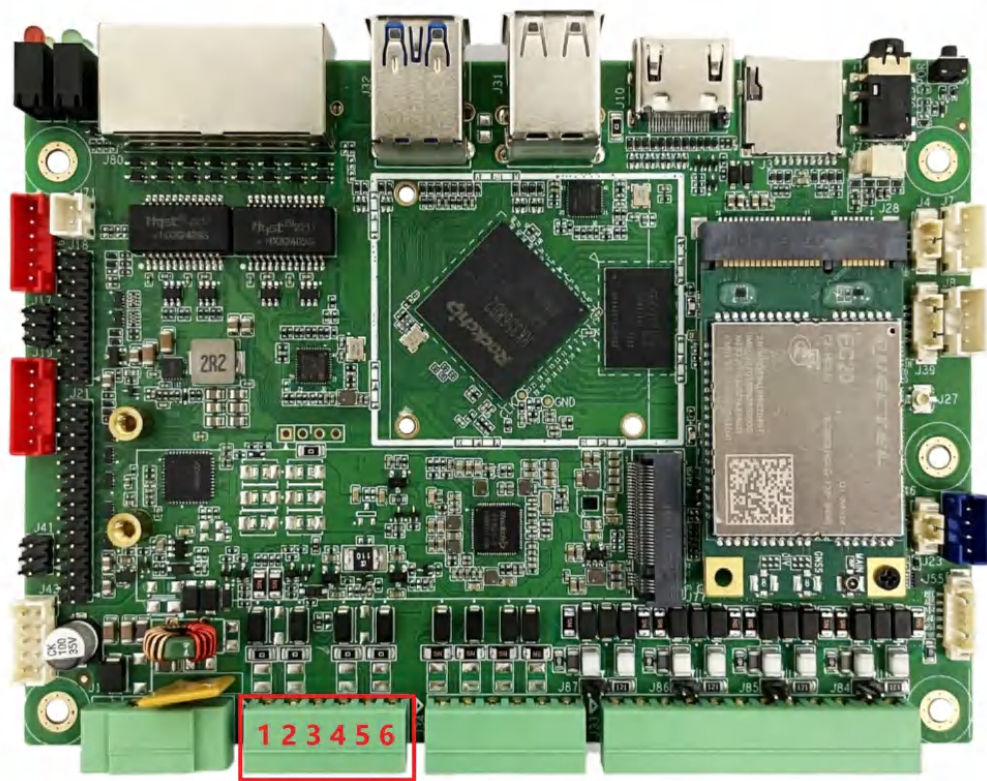
```

1 [root@RK356X:/]# mount /dev/nvme0n1p1 /mnt/sdcard/
2 [root@RK356X:/]# ls /mnt/sdcard/
3 Alarms      DCIM        Download    Movies      Notifications  Podcasts
4 Audiobooks  Documents  LOST.DIR    Music       Pictures       Ringtones
5 [root@RK356X:/]# mount
6 /dev/mmcblk0p8 on / type ext4 (rw,relatime)
7 ...
8 /dev/nvme0n1p1 on /mnt/sdcard type vfat (rw,relatime,fmask=0022,dmask=0022,
9 codepage=936,iocharset=utf8,shortname=mixed,errors=remount-ro)
10 [root@RK356X:/]#

```

14 DIDO

主板共配置2路DI和2路DO，位于J35（与CAN接口复用，二选一）：



各个引脚定义如下表所示：

引脚	功能
1	DI1
2	DI2
3	DI_SS
4	DO1
5	DO2
6	GND

14.1 DI的使用

通过读取设备节点/dev/ido_di，获取DI1和DI2的输入电平状态。打开/dev/ido_di设备文件，读取3个字节，前2个字节为两路DI的输入结果。

通过代码读取举例：

```

1  #include <stdio.h>
2  #include <sys/types.h>
3  #include <sys/stat.h>
4  #include <fcntl.h>
5  #include <unistd.h>
6  #include <stdlib.h>
7
8  int main(void)
9  {
10     char buf[3] = {0, 0, 0};
11
12     int fd;
13     int ret;
14
15     fd = open("/dev/ido_di", O_RDONLY);
16
17     if (fd < 0){
18         exit(1);
19     }
20
21     ret = read(fd, buf, 3);
22     if (ret != 3){
23         perror("read");
24         close(fd);
25         exit(2);
26     }
27
28     printf("DI1输入%s\n", buf[0] == '1' ? "低电平" : "高电平");
29     printf("DI2输入%s\n", buf[1] == '1' ? "低电平" : "高电平");
30
31     close(fd);
32
33     return 0;
34 }

```

通过命令行来获取举例：

```
1 root@linaro-alip:/home/linaro# cat /dev/ido_di
```

返回结果前2个字符与DI1、DI2的输入关系见下表：

结果	DI1输入	DI2输入
00	高	高
10	低	高
01	高	低
11	低	低

14.2 DO的使用

通过向/dev/ido_do设备文件写入特定字符串，控制DO1和DO2的输出。

通过代码控制输出举例：

```

1  #include <stdio.h>
2  #include <sys/types.h>
3  #include <sys/stat.h>
4  #include <fcntl.h>
5  #include <unistd.h>
6  #include <stdlib.h>
7
8  int main(void)
9  {
10     int fd;
11     int ret;
12
13     fd = open("/dev/ido_do", O_WRONLY);
14
15     if (fd < 0){
16         exit(1);
17     }
18
19     //D01输出低电平
20     printf("D01 输出低电平\n");
21     ret = write(fd, "1D", 3);
22     if (ret != 3)
23         goto write_err;
24     sleep(3);
25     //D01输出高电平
26     printf("D01 输出高电平\n");
27     ret = write(fd, "1E", 3);
28     if (ret != 3)
29         goto write_err;
30     sleep(3);
31
32     //D02输出低电平
33     printf("D02 输出低电平\n");
34     ret = write(fd, "2D", 3);
35     if (ret != 3)
36         goto write_err;
37     sleep(3);
38
39     //D01输出高电平
40     printf("D02 输出高电平\n");
41     ret = write(fd, "2E", 3);
42     if (ret != 3)
43         goto write_err;
44     sleep(3);

```

```
45
46         close(fd);
47
48         return 0;
49
50 write_err:
51         close(fd);
52         return -1;
53     }
```

通过命令行来设置输出举例：

```
▼ Shell |
1 //D01输出0V
2 echo 1D > /dev/ido_do
3 //D01输出12V
4 echo 1E > /dev/ido_do
5
6 //D02输出0V
7 echo 2D > /dev/ido_do
8 //D02输出12V
9 echo 2E > /dev/ido_do
```

15 外部看门狗

主板配置了外部看门狗，用于监控系统是否正常运行，当系统出现异常如卡死等情况，外部看门狗将起作用，对系统进行复位重启。

15.1 开启与关闭

开启看门狗功能：

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

关闭看门狗功能：

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

15.2 测试

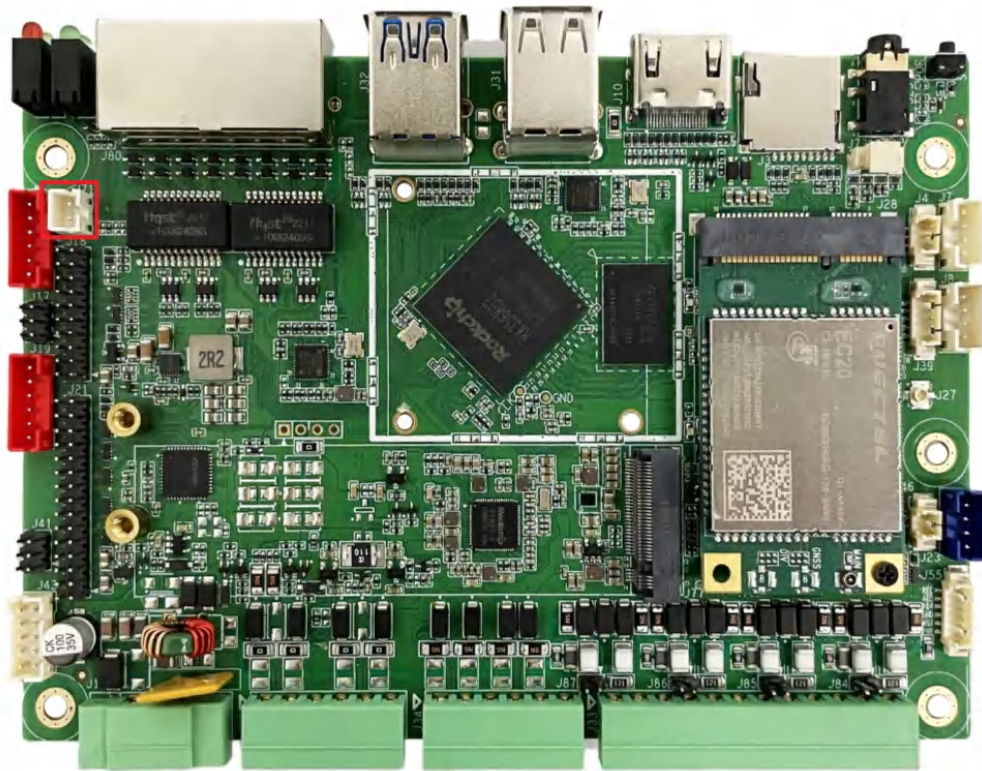
以下指令模拟系统卡死的情况，测试外部看门狗功能：

```
1 echo c > /proc/sysrq-trigger
```

执行完该命令，系统进入卡死状态，约5分钟后，系统将重启。

16 FAN

主板配置了一路风扇电源接口（输出5v/12v可选），位于J71：



16.1 开启与关闭

打开输出：

```

1  sudo echo 27 > /sys/class/gpio/export
2  sudo echo out > /sys/class/gpio/gpio27/direction
3  sudo echo 1 > /sys/class/gpio/gpio27/value

```

关闭输出：

```

1  sudo echo 0 > /sys/class/gpio/gpio27/value

```

17 LED

主板共配置4个LED指示灯：



各个指示灯的作用见下表：

编号	名称	功能
1	POWER_LED	电源指示灯，亮起表示主板供电正常
2	SYS_LED	系统指示灯，闪烁表示系统运行正常
3	USER_LED	预留用户控制
4	4G_LED	4G通信指示灯，闪烁表示4G正在通信

USER_LED控制方法如下：

```
1 //USER_LED灭
2 echo 0 > /sys/class/leds/user_led/brightness
3
4 //USER_LED亮
5 echo 255 > /sys/class/leds/user_led/brightness
```

18 RTC

主板配置2路RTC，一路为HYM8563（rtc0），一路为芯片内部RK809 RTC（rtc1）。

18.1 读取与设置时间

读取RTC时间：

```
1 linaro@linaro-alip:~$ sudo hwclock
2 2020-01-01 12:04:13.276756+00:00
3 linaro@linaro-alip:~$
```

设置RTC时间：

```
1 linaro@linaro-alip:~$ sudo date -s '2023-4-26 17:38:00'
2 2023年 04月 26日 星期三 17:38:00 UTC
3 linaro@linaro-alip:~$ sudo hwclock -w
4 linaro@linaro-alip:~$ sudo hwclock
5 2023-04-26 17:38:07.554898+00:00
6 linaro@linaro-alip:~$
```

18.2 定时开机

RK809 RTC支持定时开机功能，定时开机功能的使用参考以下步骤：

```
1 //设置300秒后开机
2 sudo echo +300 > /sys/class/rtc/rtc1/wakealarm
3
4 //立刻关机
5 sudo poweroff
```

执行以上代码，关机后，等待300秒会自动开机。

19 输入电压检测

输入电压（主板供电电压）读取节点为/sys/bus/iio/devices/iio:device0/in_voltage6_raw，通过读取该节点的值，来计算输入电压的大小。

节点读取方法为：

```
1 cat /sys/bus/iio/devices/iio:device0/in_voltage6_raw
```

输入电压为V与节点读取值W的关系如下：

$$V=(W/1024)*1.8*21$$

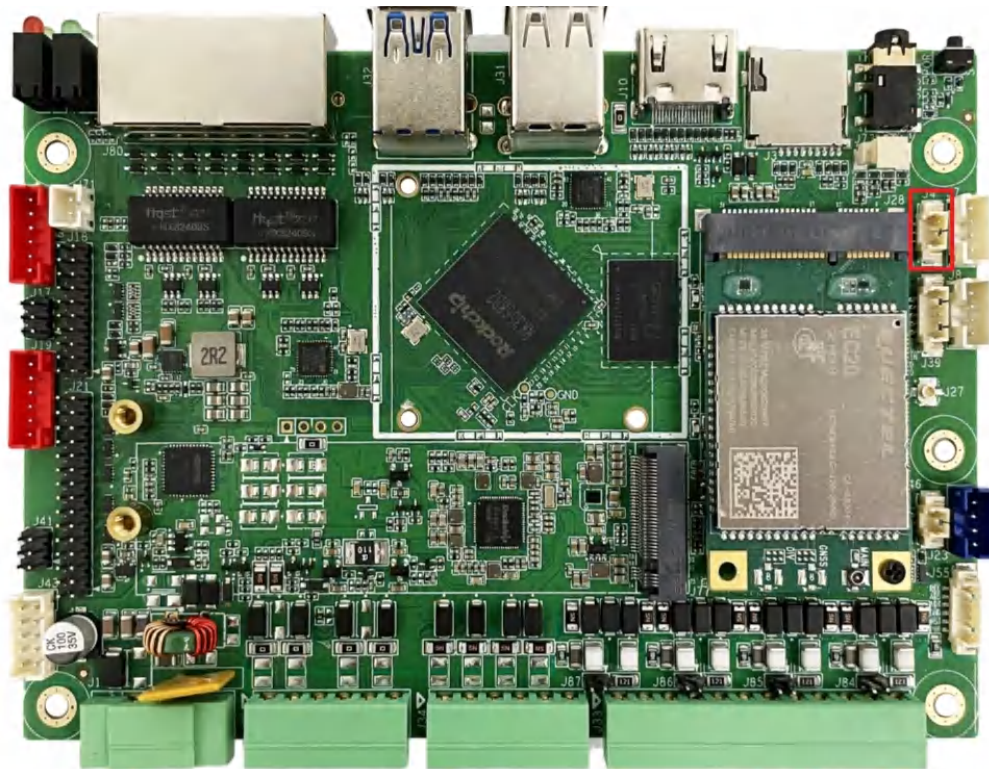
假设节点读取值为333，则主板输入电压为 $V=(333/1024)*1.8*21=12.29v$ 。

IDO-SBC3528-V1 Ubuntu系统

1 调试

1.1 串口调试

串口调试端口位于J4，通信参数为1500000 8 N 1，电平类型是TTL电平。



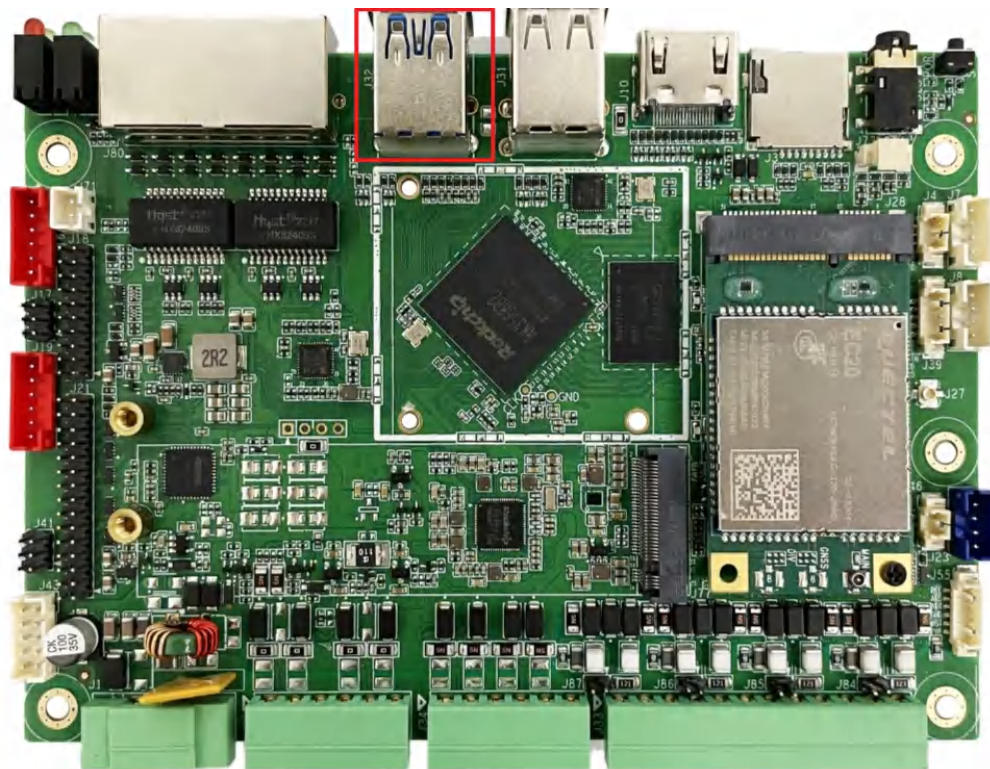
串口登录默认以root用户登录，且没有设置密码。

Bash |

```
1 root@ido:/# 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
5 root@ido:/#
```

1.2 ADB调试

adb调试端口位于J32下方的USB口，即与烧录端口为同一端口。



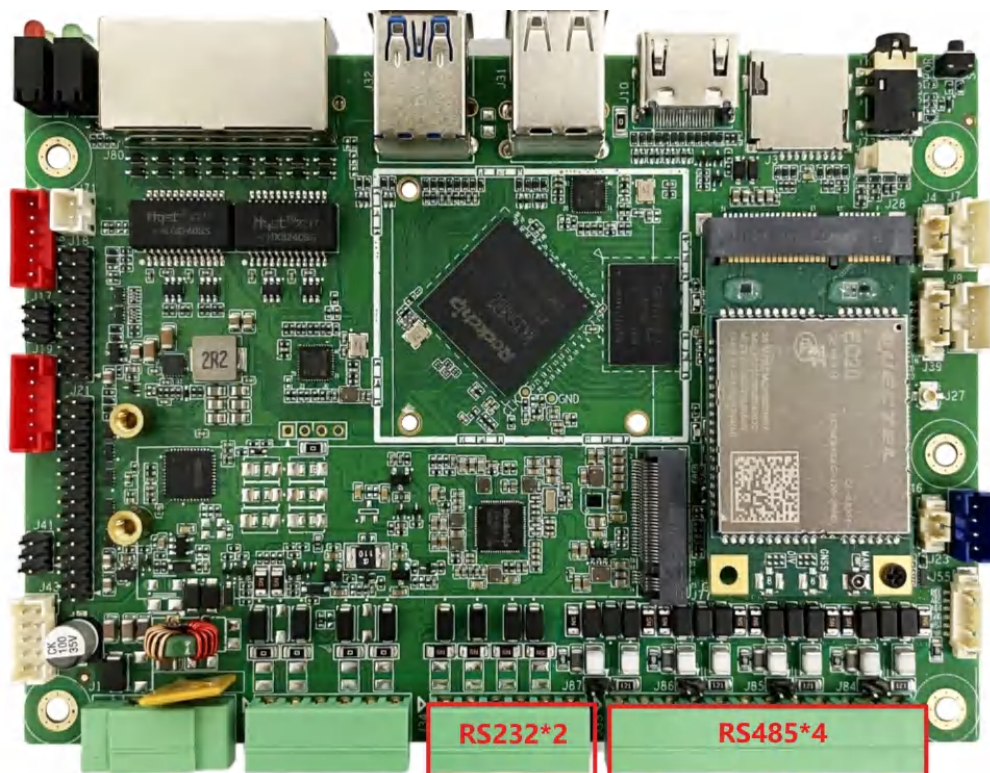
```
C:\Users\ronnie\Downloads>adb shell
adb server is out of date. killing...
* daemon started successfully *
root@ido:/# ls
ls
app.log  data  home  lost+found  opt  run  srv  tmp  var
bin      dev  lib  media      proc sbin sys  udisk vendor
boot    etc  log  mnt        root sdcard system usr
root@ido:/#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

1.3 SSH调试

SSH远程登录账号：ido，密码：123456。

2 串口

主板共配置6路串口（调试串口除外），其中4路RS485，2路RS232。



各个串口对应的设备节点如下表：

编号	设备节点	类型	位置
1	/dev/ttyS0	RS485	J33,R0-T0
2	/dev/ttyS3	RS485	J33,R3-T3
3	/dev/ttyS4	RS485	J33,,R4-T4
4	/dev/ttyS5	RS485	J33,R5-T5
5	/dev/ttyS7	RS232	J34,R7-T7
6	/dev/ttyS9	RS232	J34,R9-T9

2.1 测试

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：

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

以测试/dev/ttyS0为例：

```
1 root@ido:/# microcom -s 115200 -p /dev/ttyS0
2 [ 754.636312] of_dma_request_slave_channel: dma-names property of node '/s
serial@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.
```

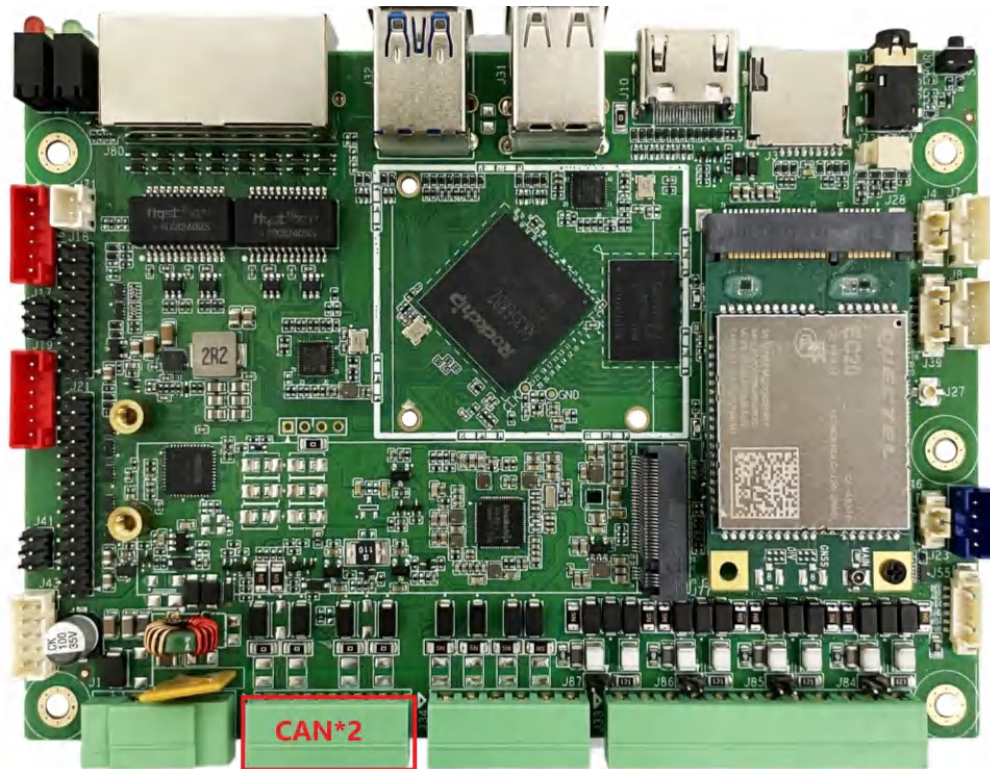
按下键盘任意键会发送对应的字符，而接收的内容会显示在终端。

按【ctrl】和【\】组合键，然后输入quit退出测试。

```
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 CAN

主板共配置2路CAN接口，系统中对应的网络节点分别为can0和can1。



2.1 测试

打开can设备，并设置通信波特率为250Kbps：

```
1  sudo ip link set can0 down
2  sudo ip link set can0 type can bitrate 250000
3  sudo ip link set can0 up
4
5  sudo ip link set can1 down
6  sudo ip link set can1 type can bitrate 250000
7  sudo ip link set can1 up
```

发送测试：

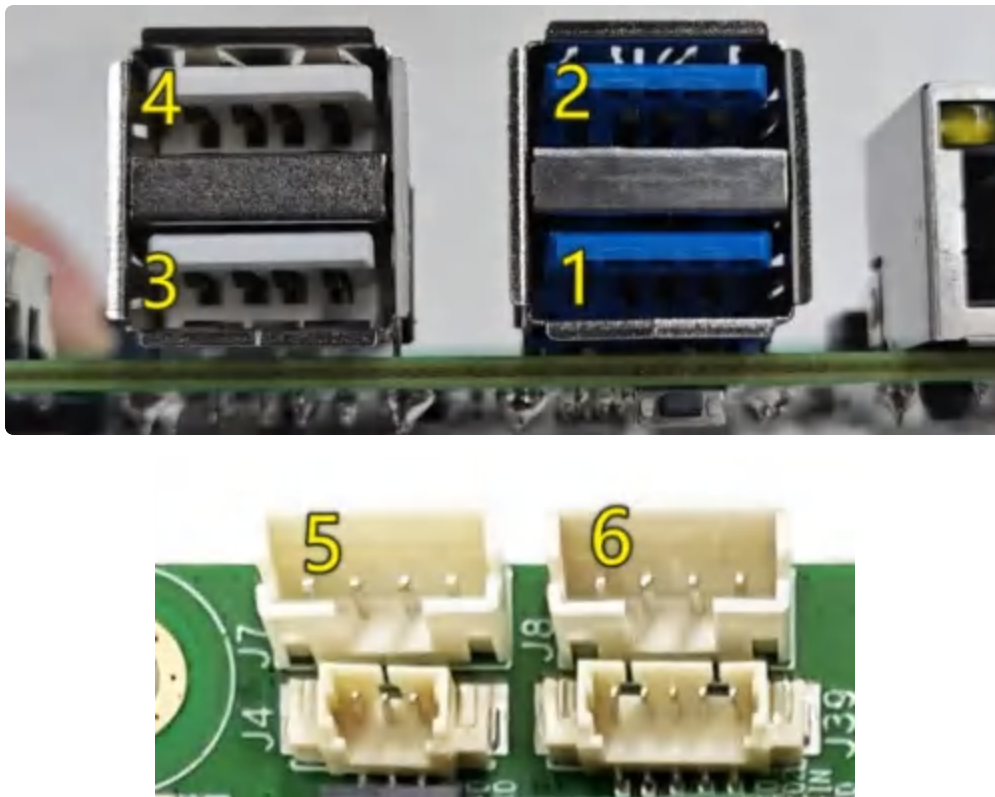
```
1  sudo cansend can0 123#DEADBEEF
```

接收测试：

```
1 sudo candump can0
```

3 USB

主板共配置6路USB接口，这里分别标记USB1-6，其中2路USB3.0（USB1和USB2），4路USB2.0（USB3-6），各个USB口位置如下图所示：



2.1 电源控制

USB2-6支持软件控制电压输出，控制节点见下表：

序号	功能	控电节点
2	USB 3.0 HOST	/sys/class/leds/usb3_host_pwr/brightness
3	USB 2.0 HOST	/sys/class/leds/usb2_host_pwr/brightness
4	USB 2.0 HOST	/sys/class/leds/usb2_fe2_pwr/brightness
5	USB 2.0 HOST	/sys/class/leds/usb2_fe3_pwr/brightness

6	USB 2.0 HOST	/sys/class/leds/usb2_fe4_pwr/brightness
---	--------------	---

供电控制说明，序号1写"1"关闭电源，写"0"开启电源；序号2-序号6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，以序号6为例

```

1  #关闭
2  echo 0 > /sys/class/leds/usb2_fe4_pwr/brightness
3  #开启（默认状态）
4  echo 1 > /sys/class/leds/usb2_fe4_pwr/brightness

```

2.2 U盘的挂载

USB端口插上U盘，会默认挂载到/ido/ido/XXX目录下：

```

1  root@ido:/# mount
2  ...
3  /dev/sda1 on /media/ido/KINGSTON type vfat ...

```

2.3 USB OTG 切换命令

USB OTG 支持host 和device 模式的切换，软件切换方法如下

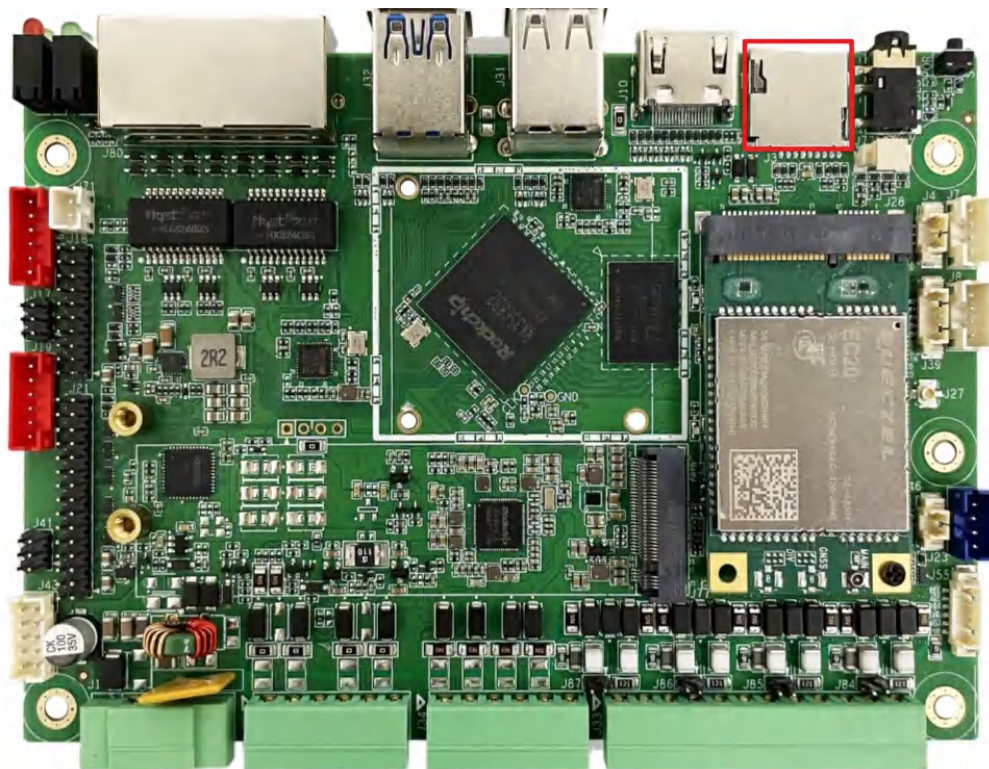
```

1  //切换HOST模式
2  echo host > /sys/devices/platform/fe8a0000.usb2-phy/otg_mode
3  echo 255 > /sys/class/leds/usb3_otg_pwr/brightness
4
5  //切换DEVICE模式
6  echo peripheral > /sys/devices/platform/fe8a0000.usb2-phy/otg_mode
7  echo 0 > /sys/class/leds/usb3_otg_pwr/brightness

```

3 MicroSD

主板共配置1个SD接口：



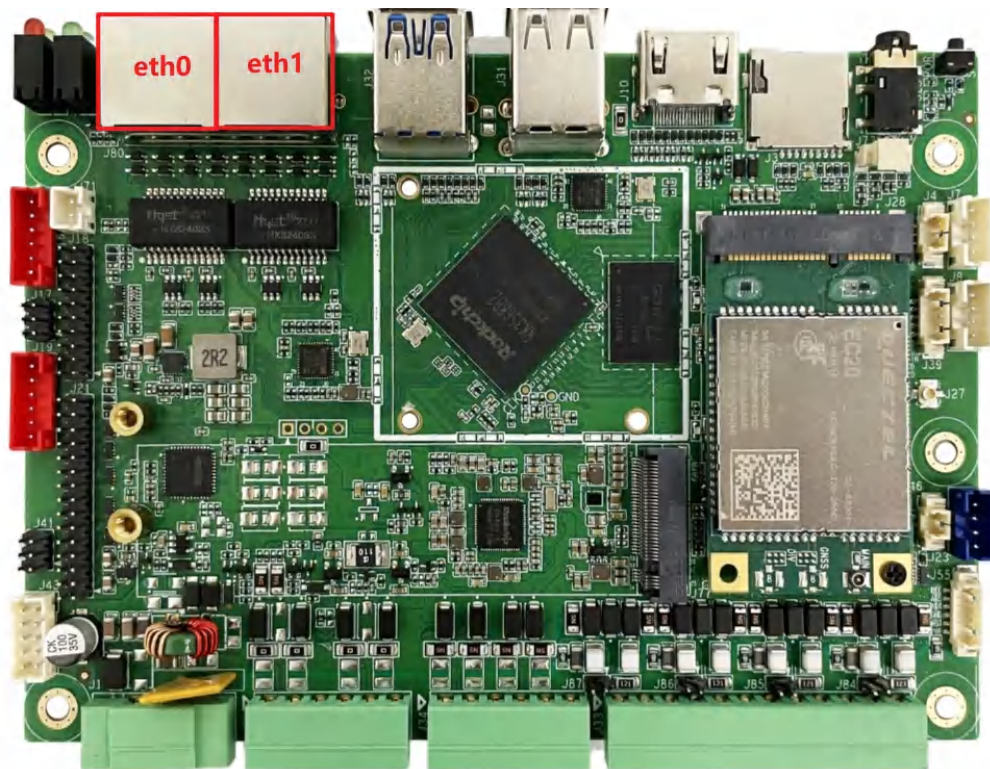
插入SD卡后，自动挂载到/media/ido/xxx目录下，可以使用mount命令确认：

Bash

```
1 root@ido:/# mount
2 ...
3 /dev/mmcblk1p1 on /media/ido/0403-0201 type vfat
```

4 以太网

主板共配置2路1000M以太网接口，系统中对应的网络节点分别为eth0和eth1：



4.1 测试

以下以eth1为例（eth0同样的操作方法）。

查看eth1的ip:

Bash |

```
1 root@ido:/# sudo ifconfig eth1
2 eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
3     inet 192.168.1.177 netmask 255.255.255.0 broadcast 192.168.1.255
4     inet6 fe80::4f32:fc2d:eb30:220b prefixlen 64 scopeid 0x20<link>
5     ether d2:e1:8d:d5:f7:49 txqueuelen 1000 (Ethernet)
6     RX packets 78 bytes 11727 (11.4 KiB)
7     RX errors 0 dropped 0 overruns 0 frame 0
8     TX packets 60 bytes 5822 (5.6 KiB)
9     TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
10    device interrupt 52
11
```

设置eth1的ip:

Bash |

```
1 root@ido:/# ifconfig eth1 192.168.1.123
```

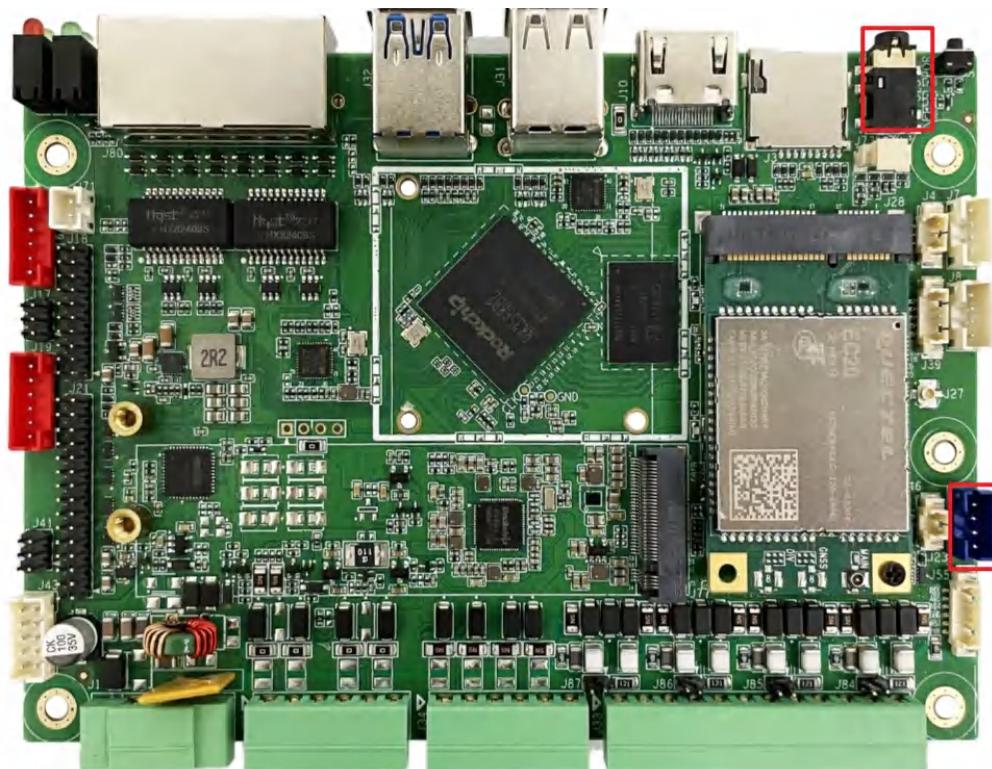
eth1 ping测试:

Bash

```
1 root@ido:/# ping www.baidu.com -I eth1
2 PING www.a.shifen.com (14.215.177.39) from 192.168.1.177 eth1: 56(84) byte
  s of data.
3 64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=1 ttl=54 time=6.75 ms
4 64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=2 ttl=54 time=6.32 ms
5 64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=3 ttl=54 time=7.01 ms
6 64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=4 ttl=54 time=7.47 ms
7 64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=5 ttl=54 time=7.01 ms
```

5 Lineout

主板支持LineOut音频输出和HDMI音频输出，LineOut音频接口位于J32：



注：默认LineOut和耳机的音频输出是同一路，即LineOut播放同时耳机也会播放。

5.1 查看声卡设备

```

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

```

5.2 播放音频

播放到HDMI:

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

播放到Lineout:

不插入耳机，执行以下命令。

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

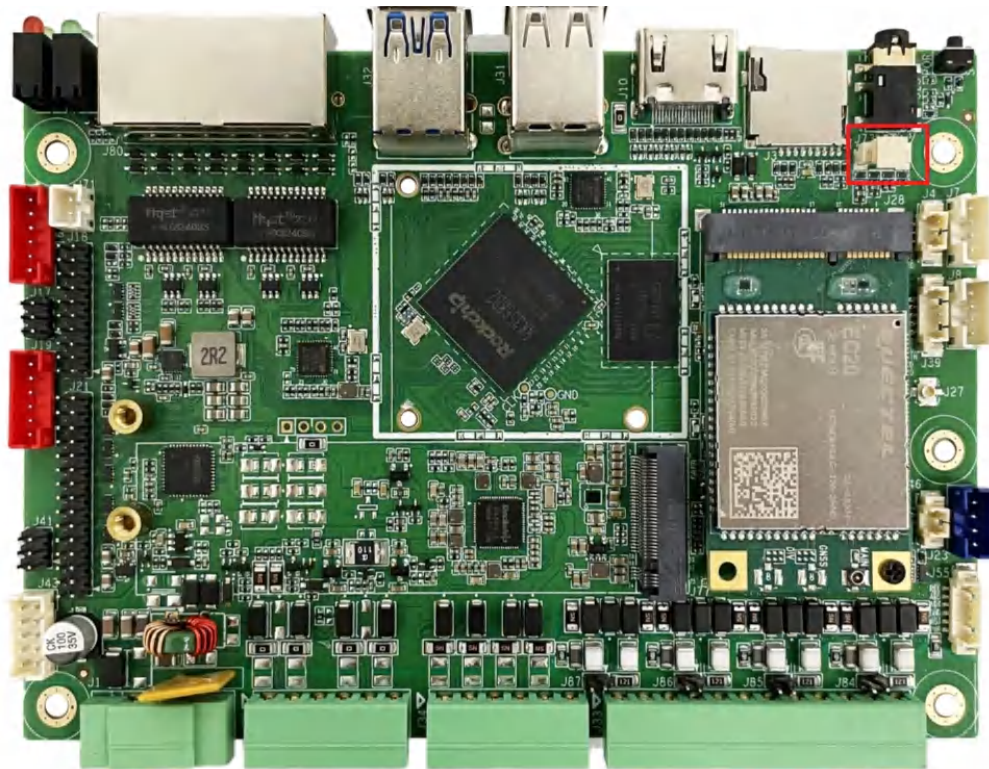
播放到耳机:

插入耳机，执行以下命令。

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

6 MIC

主板配置一路模拟麦克风接口，位于J7:



6.1 录音测试

使用arecord工具可以进行录音测试：

```
▼ Bash |
1 root@ido:~# 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
4 ^CAborted by signal 中断...
```

录音完后播放测试：

```
▼ Bash |
1 root@ido:~# aplay -D plughw:1,0 ./test.wav
2 Playing WAVE './test.wav' : Signed 16 bit Little Endian, Rate 48000 Hz, Stereo
```

7 WiFi

主板配置了一路2.4G WiFi模块，型号为AW-NM372SM。

系统上电默认会打开WiFi，对应的网络设备节点为wlan0。

```
▼ Bash |
1 root@ido:~# 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
```

7.1 连接WiFi热点

7.1.1 命令行连接

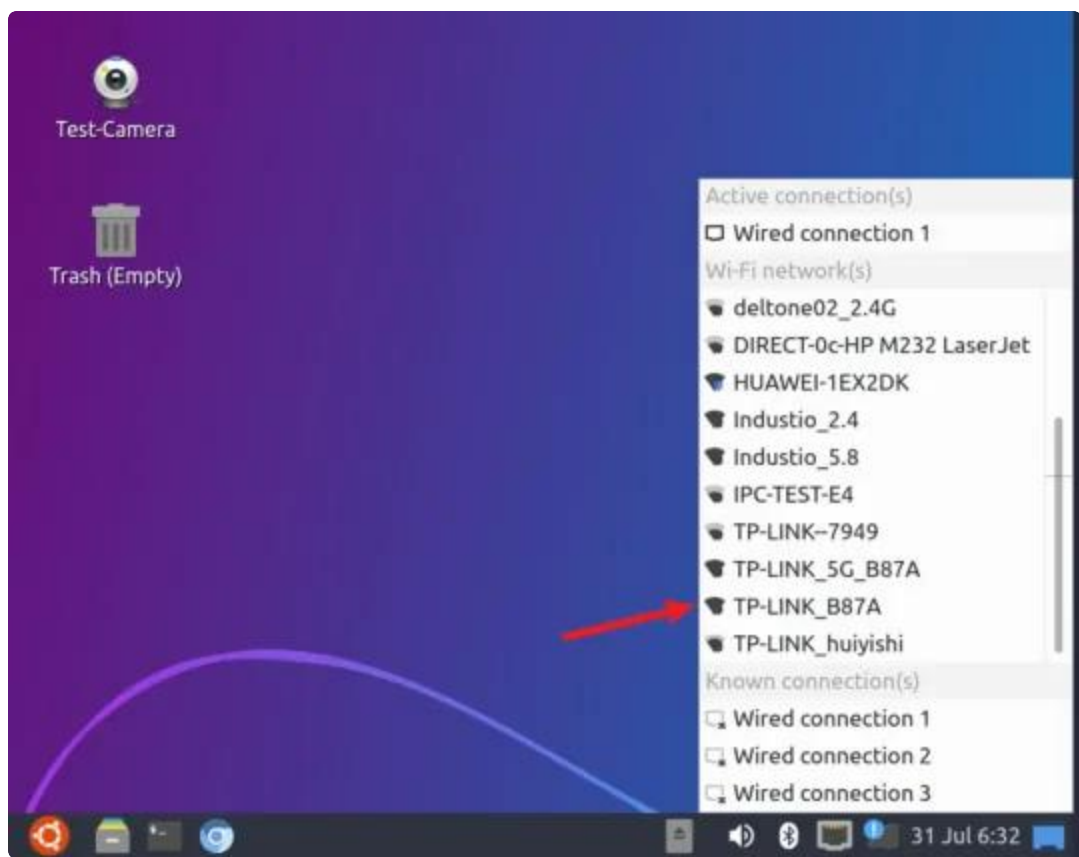
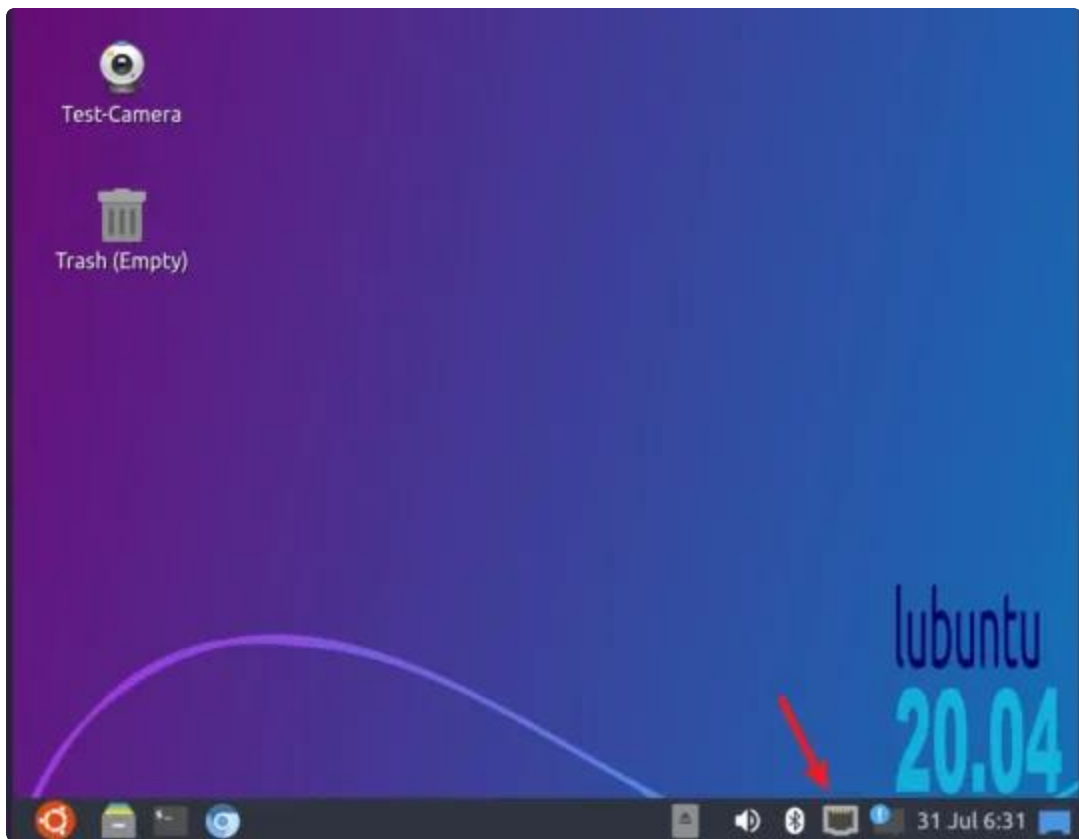
使用wpa_cli工具连接，命令如下：

```
▼ Shell |
1 root@ido:~# wpa_cli -i wlan0 add_network
2 root@ido:~# wpa_cli -i wlan0 set_network 0 ssid '"TP-LINK_B87A"'
3 root@ido:~# wpa_cli -i wlan0 set_network 0 psk '"12345678"'
4 root@ido:~# wpa_cli -i wlan0 enable_network 0
5 root@ido:~# dhclient -i wlan0
```

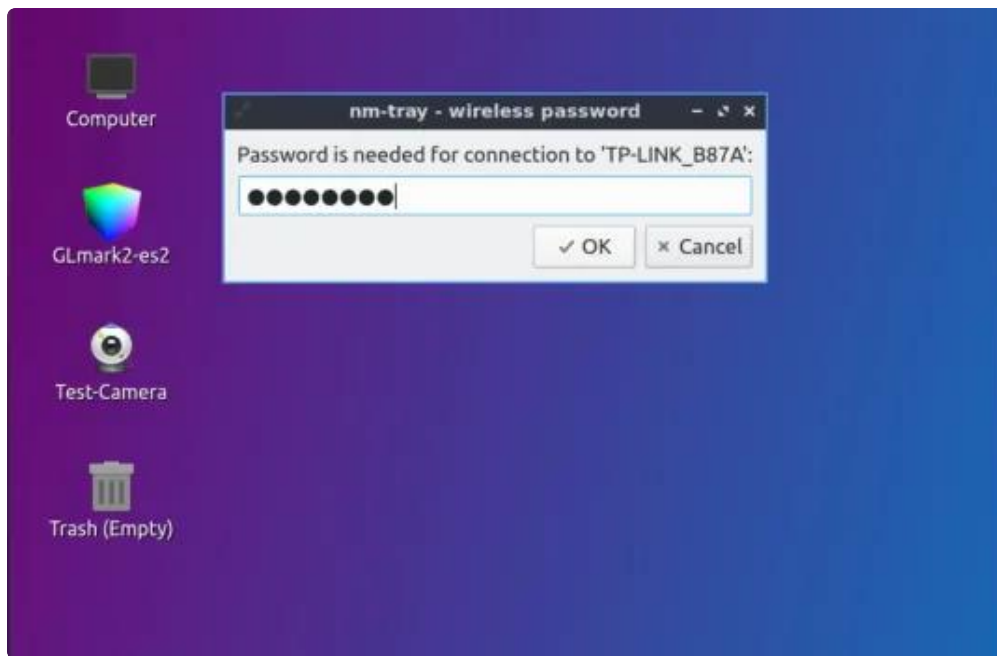
其中"TP-LINK_B87A"为热点名称，"12345678"为热点密码。

7.1.2 桌面连接

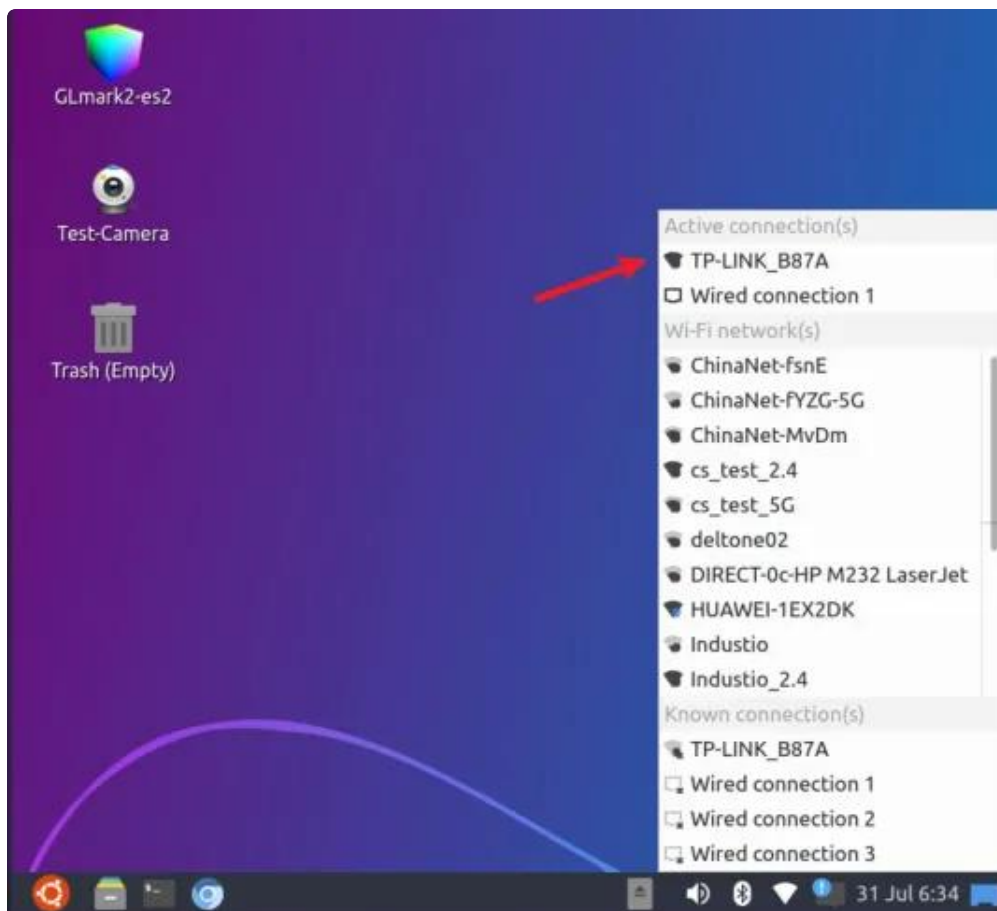
点击桌面右上角的网络按钮，弹出的列表中选择要连接的热点，如下图所示：



弹出密码输入框，使用键盘输入密码（如果没有接键盘，可以使用软键盘Onboard），如下图所示：



输入密码完成后，点击【OK】按钮，即可连接WiFi热点。



```

1 root@ido:~# ifconfig wlan0
2 wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
3         inet 192.168.1.127 netmask 255.255.255.0 broadcast 192.168.1.255
4         inet6 fe80::5ac:29b:8a95:2bf0 prefixlen 64 scopeid 0x20<link>
5         ether 50:41:1c:b0:13:e2 txqueuelen 1000 (Ethernet)
6         RX packets 1399 bytes 824772 (824.7 KB)
7         RX errors 0 dropped 0 overruns 0 frame 0
8         TX packets 977 bytes 128063 (128.0 KB)
9         TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

8 Bluetooth

系统开机默认打开蓝牙，对应的网络节点为hci0。

```

1 root@ido:~# hciconfig hci0 up
2 root@ido:~# hciconfig
3 hci0:   Type: Primary   Bus: UART
4         BD Address: 50:41:1C:5D:02:87 ACL MTU: 1021:8 SCO MTU: 64:1
5         UP RUNNING
6         RX bytes:871 acl:0 sco:0 events:61 errors:0
7         TX bytes:3282 acl:0 sco:0 commands:61 errors:0

```

8.1 连接蓝牙设备

使用命令行连接蓝牙设备，命令如下：

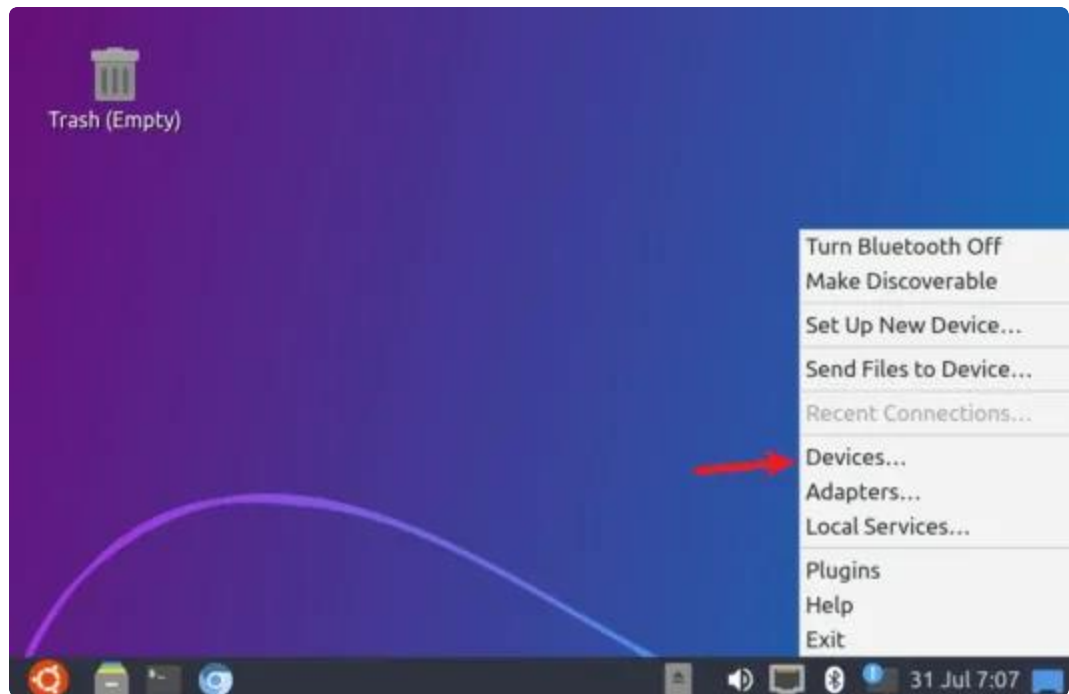
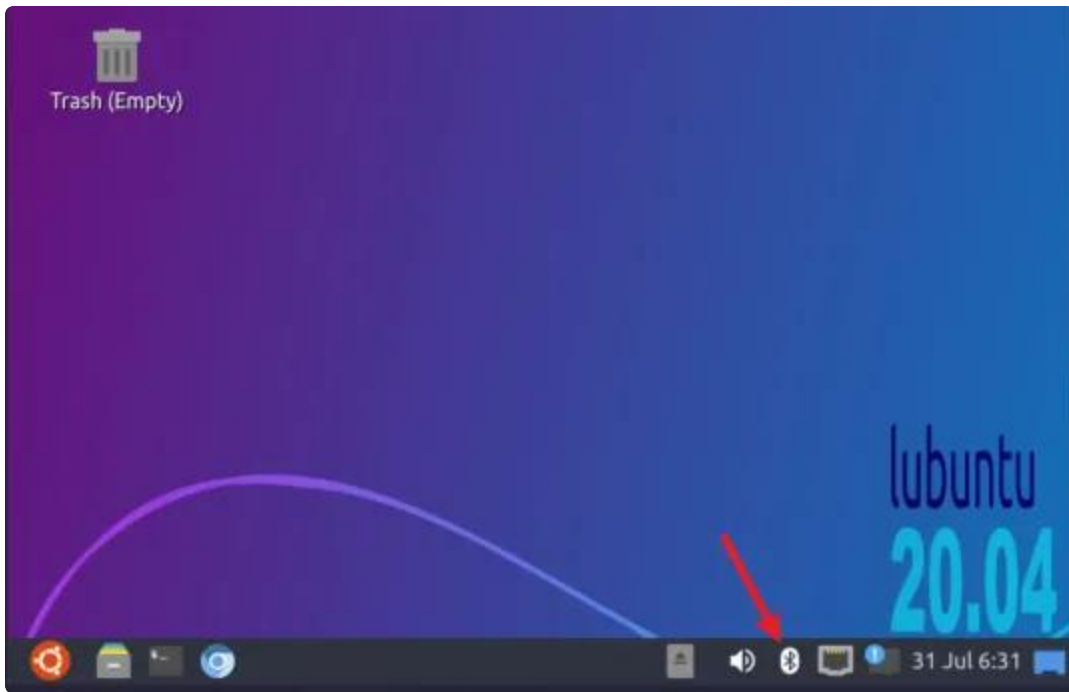
```

1 root@ido:~# bluetoothctl
2 [bluetooth]# scan on
3 ...
4 [bluetooth]# trust 7C:C1:80:09:DD:6C
5 [bluetooth]# pair 7C:C1:80:09:DD:6C
6 [cainiaocl]# exit

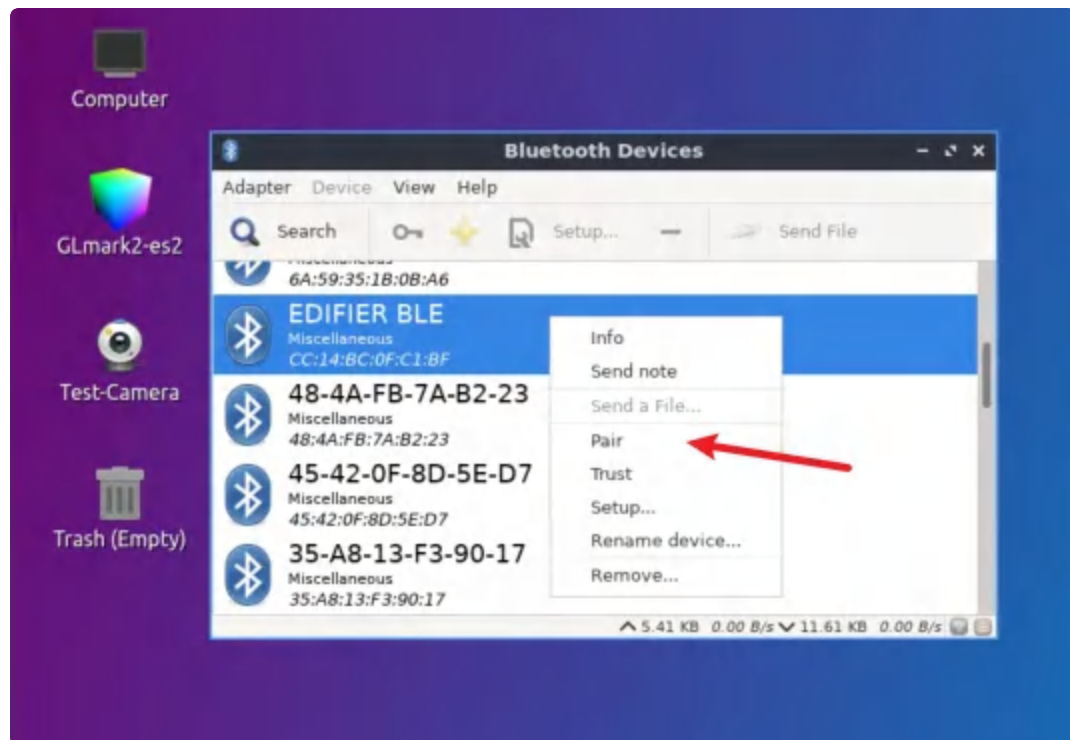
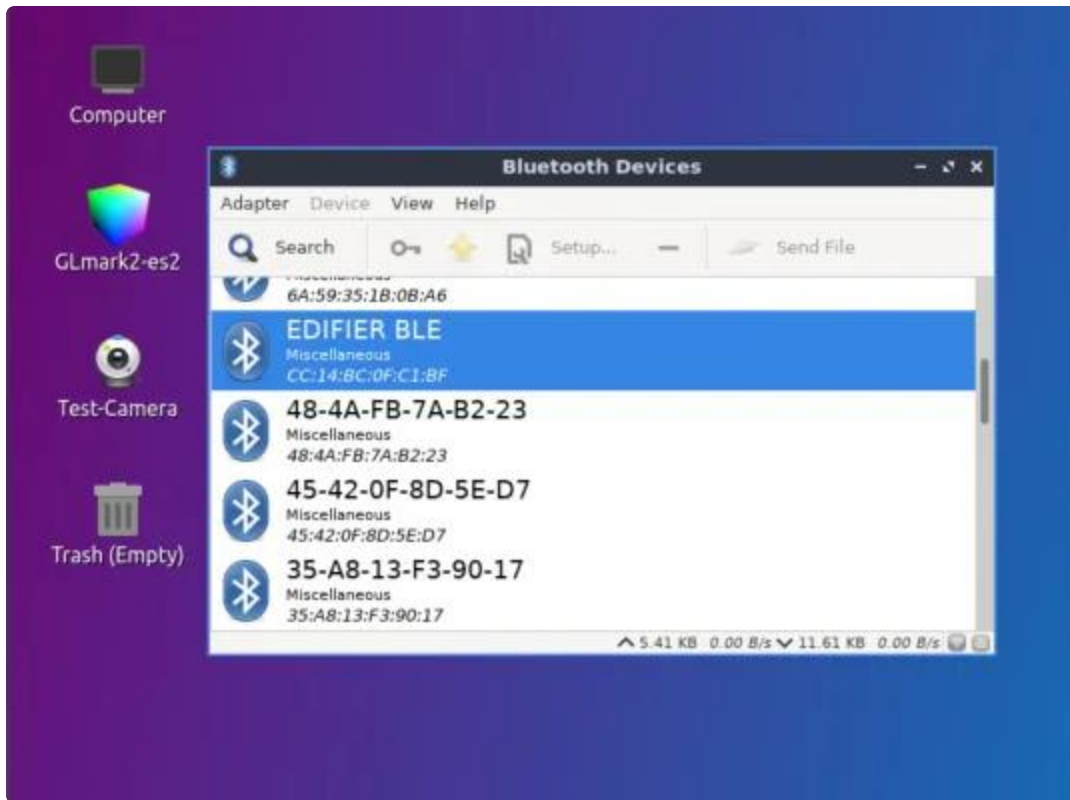
```

也可在桌面上连接蓝牙设备，步骤如下：

在桌面右下角蓝牙图标右键->Devices:



选中要连接的蓝牙设备，右键->pair:



9 4G/5G

主板配置了一路4G/5G接口，默认支持EC20-CN（4G）和EC200U-CN（5G）。

序号	模块名称	说明
----	------	----

1	EC20	4G LTE
2	RG200U-CN	支持 5G NSA 和 SA 模式，支持 TDD 和 FDD 两种模式，向下兼容 4G/3G。

9.1 4G/5G 拨号

正确按照4G模块、SIM卡和4G天线后，使用quectel-CM工具拨号，命令如下：

```

▼ Bash |
1 root@ido:~# /usr/bin/quectel-CM &

```

拨号成功，则会产生wwan0/usb0网络节点：

```

▼ Bash |
1 root@ido:~# ifconfig wwan0
2 wwan0      Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
3           inet addr:10.122.45.92  P-t-P:10.122.45.92  Mask:255.255.255.248
4           UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
5           RX packets:12 errors:0 dropped:0 overruns:0 frame:0
6           TX packets:59 errors:0 dropped:0 overruns:0 carrier:0
7           collisions:0 txqueuelen:1000
8           RX bytes:1396 (1.3 KiB)  TX bytes:8214 (8.0 KiB)

```

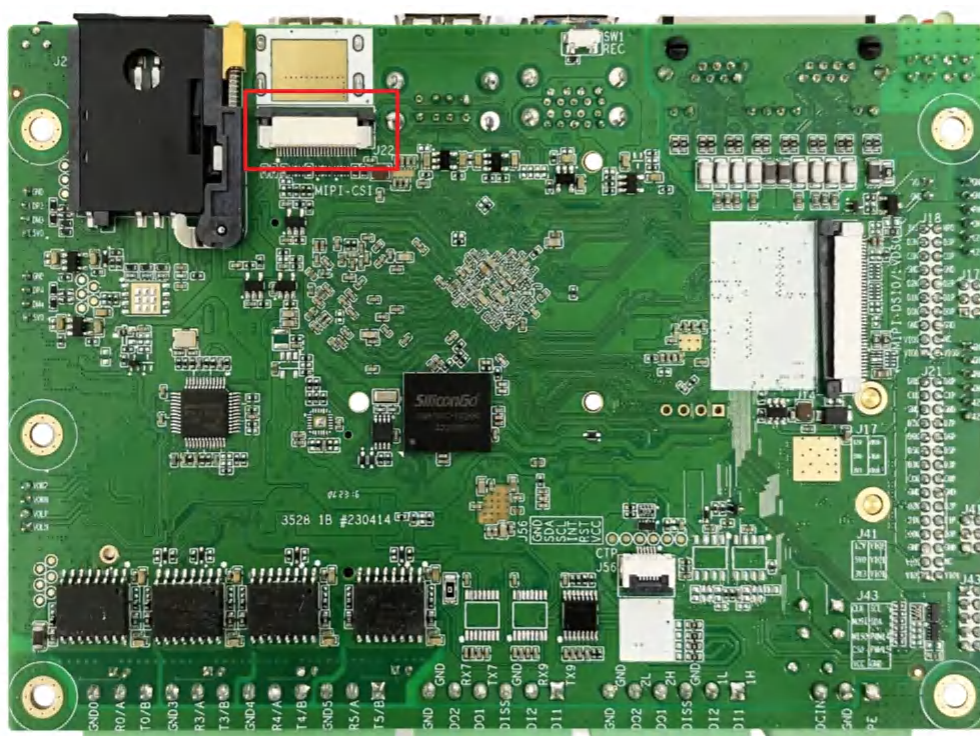
9.2 4G/5G 上网测试

使用ping命令测试4G/5G上网功能是否正常：

```
1 root@ido:~# ping www.baidu.com -I wwan0
2 PING www.a.shifen.com (14.119.104.189) from 10.122.45.92 wwan0: 56(84) byte
  s of data.
3 64 bytes from 14.119.104.189 (14.119.104.189): icmp_seq=1 ttl=52 time=95.5
  ms
4 64 bytes from 14.119.104.189 (14.119.104.189): icmp_seq=2 ttl=52 time=99.3
  ms
5 64 bytes from 14.119.104.189 (14.119.104.189): icmp_seq=3 ttl=52 time=97.8
  mss
```

10 摄像头

主板配置了一路CSI摄像头，默认支持型号OV5648和OV8858，位于J22：



10.1 测试摄像头是否存在

```

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@100000/150000 field:none]
8         <- "m00_b_ov5648 2-0036":0 [ENABLED]
9     pad1: Source
10        [fmt:SBGGR10_1X10/2592x1944@100000/150000 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@100000/150000 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。

10.2 预览摄像头画面

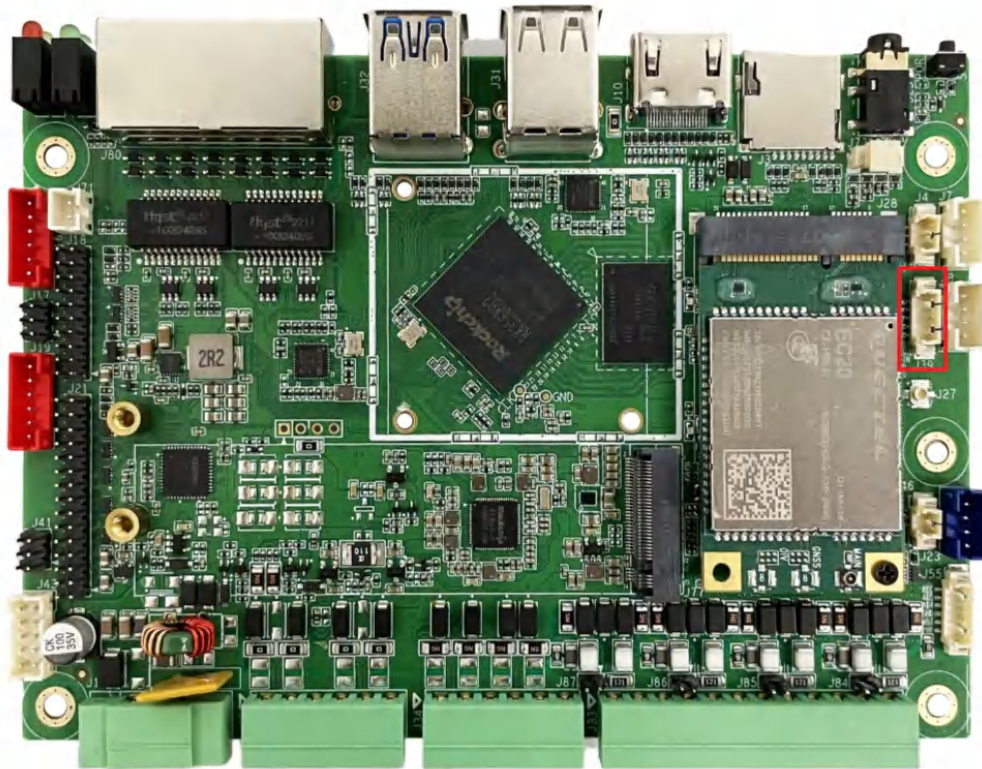
```

1 root@ido:~# gst-launch-1.0 v4l2src device=/dev/video0 ! video/x-raw, format
  =NV12, width=1920, height=1080, framerate=15/1 ! autovideoconvert ! autovid
  eosink

```

11 红外

主板配置了一路红外接口，位于J39，支持NEC编码遥控器，默认适配的遥控器型号为HTR-A07，使用时需要安装好配套的红外扩展小板。





HTR-A07的键值如下表所示：

编号	按键	键值	编号	按键	键值
1	电源	KEY_POWER	21	1	KEY_1
2	TV	KEY_SCREEN	22	2	KEY_2
3	橙色	KEY_F1	23	3	KEY_3
4	绿色	KEY_F2	24	4	KEY_4
5	黄色	KEY_F3	25	5	KEY_5
6	紫色	KEY_F4	26	6	KEY_6
7	音量+	KEY_VOLUMEUP	27	7	KEY_7

8	音量-	KEY_VOLUMEDOWN	28	8	KEY_8
9	屏显	KEY_DISPLAY_OFF	29	9	KEY_9
10	静音	KEY_MUTE	30	TVNOW	KEY_DOT
11	上一节目	KEY_VIDEO_PREV	31	0	KEY_0
12	下一节目	KEY_VIDEO_NEXT	32	截屏	KEY_PRINT
13	上	KEY_UP	33	/	/
14	左	KEY_LEFT	34	/	/
15	下	KEY_DOWN	35	/	/
16	右	KEY_RIGHT	36	/	/
17	确认	KEY_ENTER	37	/	/
18	返回	KEY_BACK	38	/	/
19	主页	KEY_HOME	39	/	/
20	菜单	KEY_MENU	40	/	/

11.1 查看红外按键值

▼

Bash

```
1 echo 1 > /sys/module/rockchip_pwm_remotectl/parameters/code_print
```

12 ADC

主板共配置2路ADC，位于J55，丝印标记为AD4和AD5。



12.1 计算输入电压

使用如下公式计算输入电压：

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

raw为从设备节点读取的数值，两路AD对应的设备节点见下表：

AD4	/sys/bus/iio/devices/iio:device0/in_voltage4_raw
AD5	/sys/bus/iio/devices/iio:device0/in_voltage5_raw

举例：

AD4输入电压计算

```
▼ Bash |
1 cat /sys/bus/iio/devices/iio:device0/in_voltage4_raw
2 258
```

则AD4的输入电压 $V=(258/1024)*1.8\text{v}=0.454\text{v}$ 。

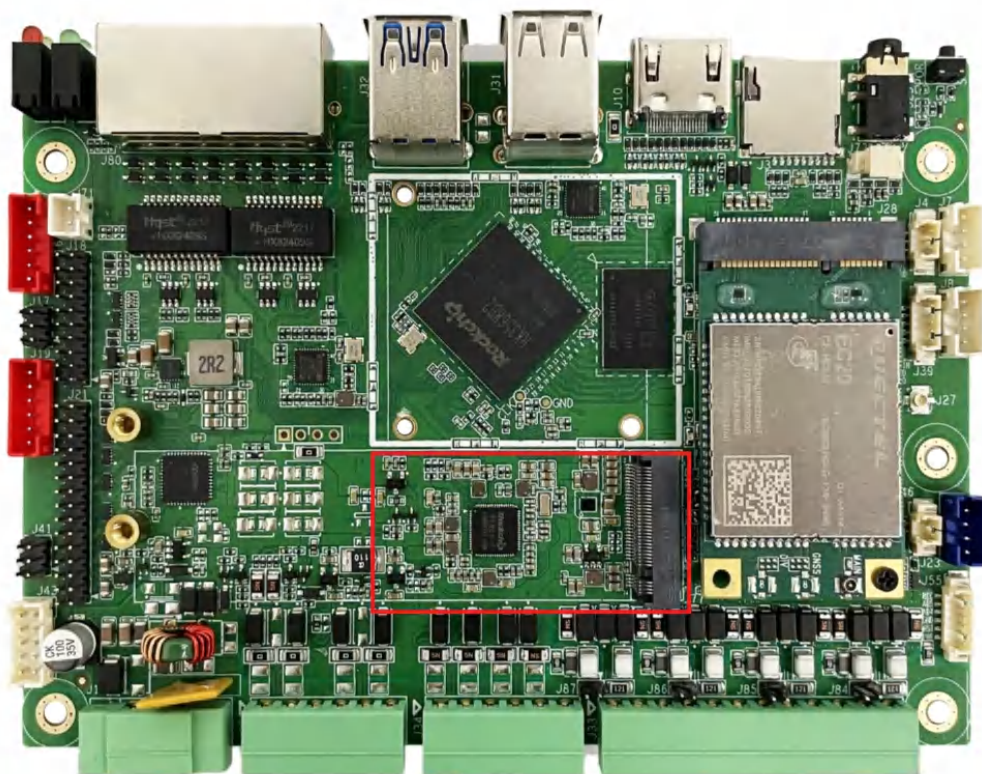
AD5输入电压计算：

```
1 cat /sys/bus/iio/devices/iio:device0/in_voltage5_raw
2 137
```

则AD5的输入电压 $V=(137/1024)*1.8v=0.241v$ 。

13 PCIE

主板配置了一路PCIE3.0接口，位于J77：



断电安装好PCIE固态硬盘，主板上电，系统启动后，硬盘将自动识别：

```

1 root@ido:~# fdisk -l
2 Found valid GPT with protective MBR; using GPT
3
4 Disk /dev/mmcblk0: 30535680 sectors, 2622M
5 Logical sector size: 512
6 Disk identifier (GUID): c7460000-0000-4b06-8000-5539000009bc
7 Partition table holds up to 128 entries
8 First usable sector is 34, last usable sector is 30535646
9
10 Number  Start (sector)    End (sector)  Size Name
11      1           16384             28671 6144K uboot
12      2           28672             36863 4096K misc
13      3           36864            102399 32.0M boot
14      4          102400            167935 32.0M recovery
15      5          167936            233471 32.0M backup
16      6          233472            757759 256M userdata
17      7          757760            1019903 128M oem
18      8          1019904           30535615 14.0G rootfs
19 Disk /dev/nvme0n1: 119 GB, 128035676160 bytes, 250069680 sectors
20 15566 cylinders, 255 heads, 63 sectors/track
21 Units: cylinders of 16065 * 512 = 8225280 bytes
22
23 Device      Boot StartCHS      EndCHS          StartLBA      EndLBA      Sector
24 s  Size Id Type
25 /dev/nvme0n1p1  1,0,1      1023,254,63      2048  250069646  25006759
    9 119G  c Win95 FAT32 (LBA)
25 root@ido:~#

```

可以看到PCIE硬盘在系统中的设备节点为/dev/nvme0n1p1。

使用mount命令可以挂载到系统中进行文件操作：

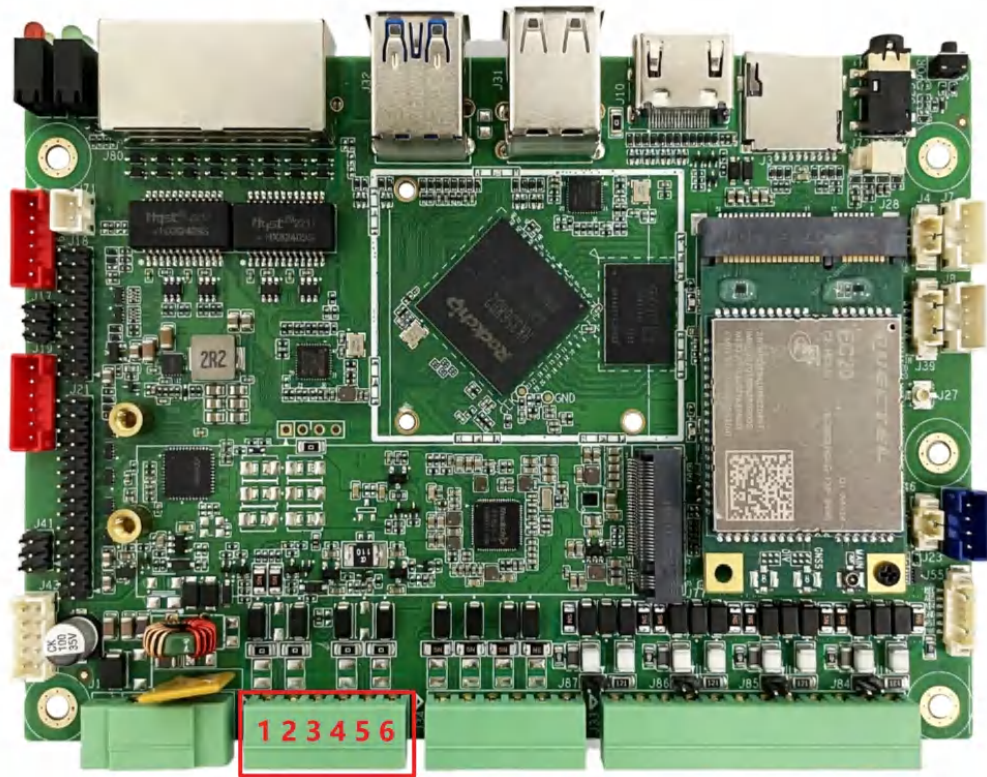
```

1 root@ido:~# mount /dev/nvme0n1p1 /mnt/sdcard/
2 [root@RK356X:/]# ls /mnt/sdcard/
3 Alarms      DCIM        Download    Movies      Notifications  Podcasts
4 Audiobooks  Documents  LOST.DIR    Music       Pictures       Ringtones
5 root@ido:~# mount
6 /dev/mmcblk0p8 on / type ext4 (rw,relatime)
7 ...
8 /dev/nvme0n1p1 on /mnt/sdcard type vfat (rw,relatime,fmask=0022,dmask=0022,
9 codepage=936,iocharset=utf8,shortname=mixed,errors=remount-ro)
10 root@ido:~#

```

14 DIDO

主板共配置2路DI和2路DO，位于J35（与CAN接口复用，二选一）：



各个引脚定义如下表所示：

引脚	功能
1	DI1
2	DI2
3	DI_SS
4	DO1
5	DO2
6	GND

14.1 DI的使用

通过读取设备节点/dev/ido_di，获取DI1和DI2的输入电平状态。打开/dev/ido_di设备文件，读取3个字节，前2个字节为两路DI的输入结果。

通过代码读取举例：

```

1  #include <stdio.h>
2  #include <sys/types.h>
3  #include <sys/stat.h>
4  #include <fcntl.h>
5  #include <unistd.h>
6  #include <stdlib.h>
7
8  int main(void)
9  {
10     char buf[3] = {0, 0, 0};
11
12     int fd;
13     int ret;
14
15     fd = open("/dev/ido_di", O_RDONLY);
16
17     if (fd < 0){
18         exit(1);
19     }
20
21     ret = read(fd, buf, 3);
22     if (ret != 3){
23         perror("read");
24         close(fd);
25         exit(2);
26     }
27
28     printf("DI1输入%s\n", buf[0] == '1' ? "低电平" : "高电平");
29     printf("DI2输入%s\n", buf[1] == '1' ? "低电平" : "高电平");
30
31     close(fd);
32
33     return 0;
34 }

```

通过命令行来获取举例：

[illegible]

返回结果前2个字符与DI1、DI2的输入关系见下表：

结果	DI1输入	DI2输入
00	高	高
10	低	高
01	高	低
11	低	低

14.2 DO的使用

通过向/dev/ido_do设备文件写入特定字符串，控制DO1和DO2的输出。

通过代码控制输出举例：

```

1  #include <stdio.h>
2  #include <sys/types.h>
3  #include <sys/stat.h>
4  #include <fcntl.h>
5  #include <unistd.h>
6  #include <stdlib.h>
7
8  int main(void)
9  {
10     int fd;
11     int ret;
12
13     fd = open("/dev/ido_do", O_WRONLY);
14
15     if (fd < 0){
16         exit(1);
17     }
18
19     //D01输出低电平
20     printf("D01 输出低电平\n");
21     ret = write(fd, "1D", 3);
22     if (ret != 3)
23         goto write_err;
24     sleep(3);
25     //D01输出高电平
26     printf("D01 输出高电平\n");
27     ret = write(fd, "1E", 3);
28     if (ret != 3)
29         goto write_err;
30     sleep(3);
31
32     //D02输出低电平
33     printf("D02 输出低电平\n");
34     ret = write(fd, "2D", 3);
35     if (ret != 3)
36         goto write_err;
37     sleep(3);
38
39     //D01输出高电平
40     printf("D02 输出高电平\n");
41     ret = write(fd, "2E", 3);
42     if (ret != 3)
43         goto write_err;
44     sleep(3);

```

```
45
46         close(fd);
47
48         return 0;
49
50 write_err:
51         close(fd);
52         return -1;
53     }
```

通过命令行来设置输出举例：

```
▼ Shell |
1 //D01输出0V
2 echo 1D > /dev/ido_do
3 //D01输出12V
4 echo 1E > /dev/ido_do
5
6 //D02输出0V
7 echo 2D > /dev/ido_do
8 //D02输出12V
9 echo 2E > /dev/ido_do
```

15 外部看门狗

主板配置了外部看门狗，用于监控系统是否正常运行，当系统出现异常如卡死等情况，外部看门狗将起作用，对系统进行复位重启。

15.1 开启与关闭

开启看门狗功能：

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

关闭看门狗功能：

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

15.2 测试

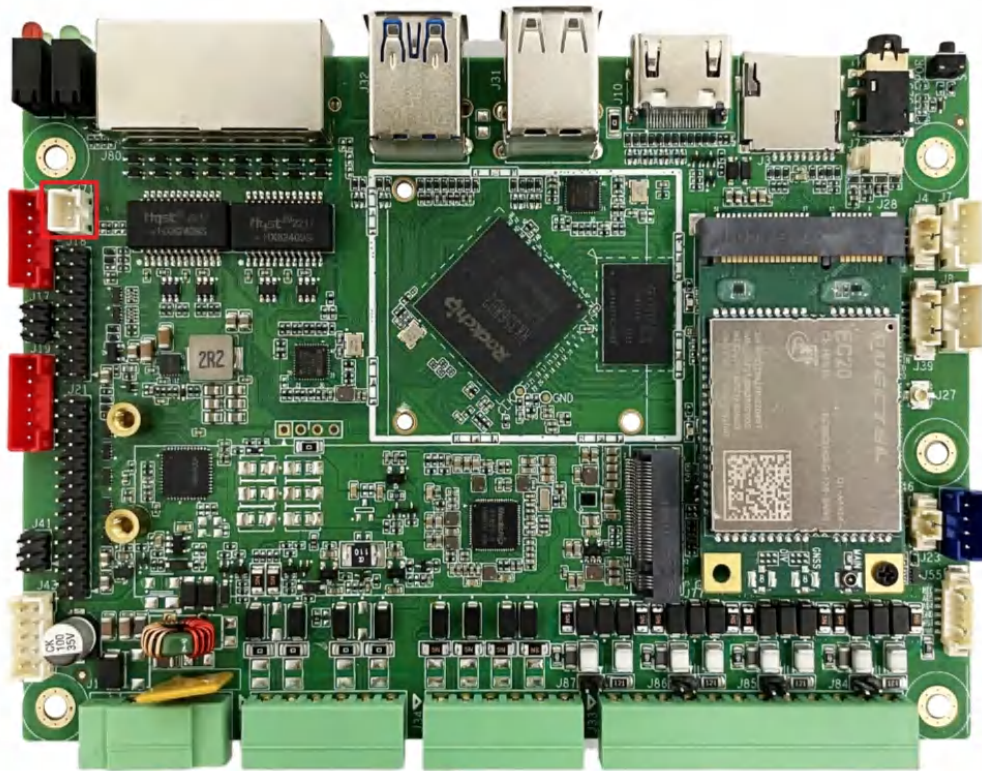
以下指令模拟系统卡死的情况，测试外部看门狗功能：

```
1 echo c > /proc/sysrq-trigger
```

执行完该命令，系统进入卡死状态，约5分钟后，系统将重启。

16 FAN

主板配置了一路风扇电源接口（输出5v/12v可选），位于J71：



16.1 开启与关闭

打开输出：

```

1  sudo echo 27 > /sys/class/gpio/export
2  sudo echo out > /sys/class/gpio/gpio27/direction
3  sudo echo 1 > /sys/class/gpio/gpio27/value

```

关闭输出：

```

1  sudo echo 0 > /sys/class/gpio/gpio27/value

```

17 LED

主板共配置4个LED指示灯：



各个指示灯的作用见下表：

编号	名称	功能
1	POWER_LED	电源指示灯，亮起表示主板供电正常
2	SYS_LED	系统指示灯，闪烁表示系统运行正常
3	USER_LED	预留用户控制
4	4G_LED	4G通信指示灯，闪烁表示4G正在通信

USER_LED控制方法如下：

```
1 //USER_LED灭
2 echo 0 > /sys/class/leds/user_led/brightness
3
4 //USER_LED亮
5 echo 255 > /sys/class/leds/user_led/brightness
```

18 RTC

主板配置2路RTC，一路为HYM8563（rtc0），一路为芯片内部RK809 RTC（rtc1）。

18.1 读取与设置时间

读取RTC时间：

```
1 root@ido:~# hwclock
2 2020-01-01 12:04:13.276756+00:00
```

设置RTC时间：

```
1 root@ido:~# date -s '2023-4-26 17:38:00'
2 2023年 04月 26日 星期三 17:38:00 UTC
3 root@ido:~# hwclock -w
4 root@ido:~# hwclock
5 2023-04-26 17:38:07.554898+00:00
```

18.2 定时开机

RK809 RTC支持定时开机功能，定时开机功能的使用参考以下步骤：

```

1 //设置300秒后开机
2 sudo echo +300 > /sys/class/rtc/rtc1/wakealarm
3
4 //立刻关机
5 sudo poweroff

```

执行以上代码，关机后，等待300秒会自动开机。

19 输入电压检测

输入电压（主板供电电压）读取节点为/sys/bus/iio/devices/iio:device0/in_voltage6_raw，通过读取该节点的值，来计算输入电压的大小。

节点读取方法为：

```

1 cat /sys/bus/iio/devices/iio:device0/in_voltage6_raw

```

输入电压为V与节点读取值W的关系如下：

$$V=(W/1024)*1.8*21$$

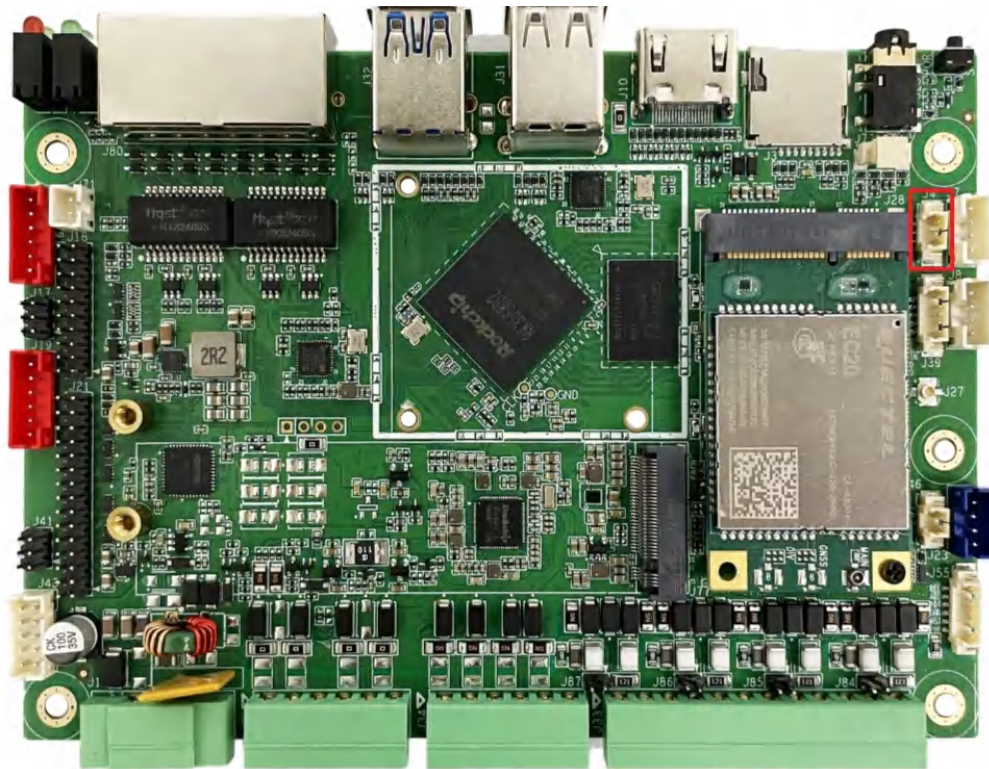
假设节点读取值为333，则主板输入电压为 $V=(333/1024)*1.8*21=12.29v$ 。

IDO-SBC3528-V1 Buildroot系统

1 调试

1.1 串口调试

串口调试端口位于J4，通信参数为1500000 8 N 1，电平类型是TTL电平。



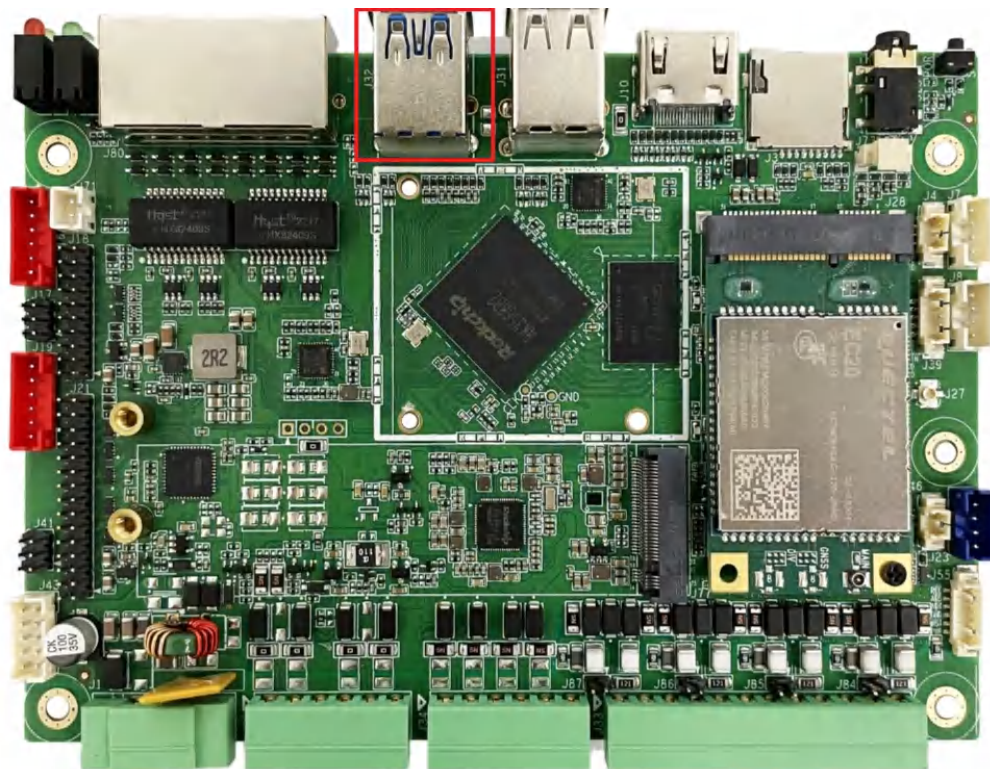
串口登录默认以root用户登录，且没有设置密码。

Bash |

```
1 [root@RK356X:/]# ls
2 bin          init          media  proc          sdcard        udisk
3 busybox.config lib          misc   rockchip_test sys           userdata
4 data         lib64        mnt    root          system        usr
5 dev          linuxrc      oem    run           timestamp     var
6 etc          lost+found  opt    sbin          tmp           vendor
7 [root@RK356X:/]#
```

1.2 ADB调试

adb调试端口位于J32下方的USB口，即与烧录端口为同一端口。



```
[root@RK356X:/]# ls
bin          init         media  proc        sdcard      udisk
busybox.config  lib         misc   rockchip_test sys          userdata
data         lib64        mnt    root        system      usr
dev          linuxrc      oem    run         timestamp   var
etc          lost+found  opt    sbin        tmp         vendor
[root@RK356X:/]#
```

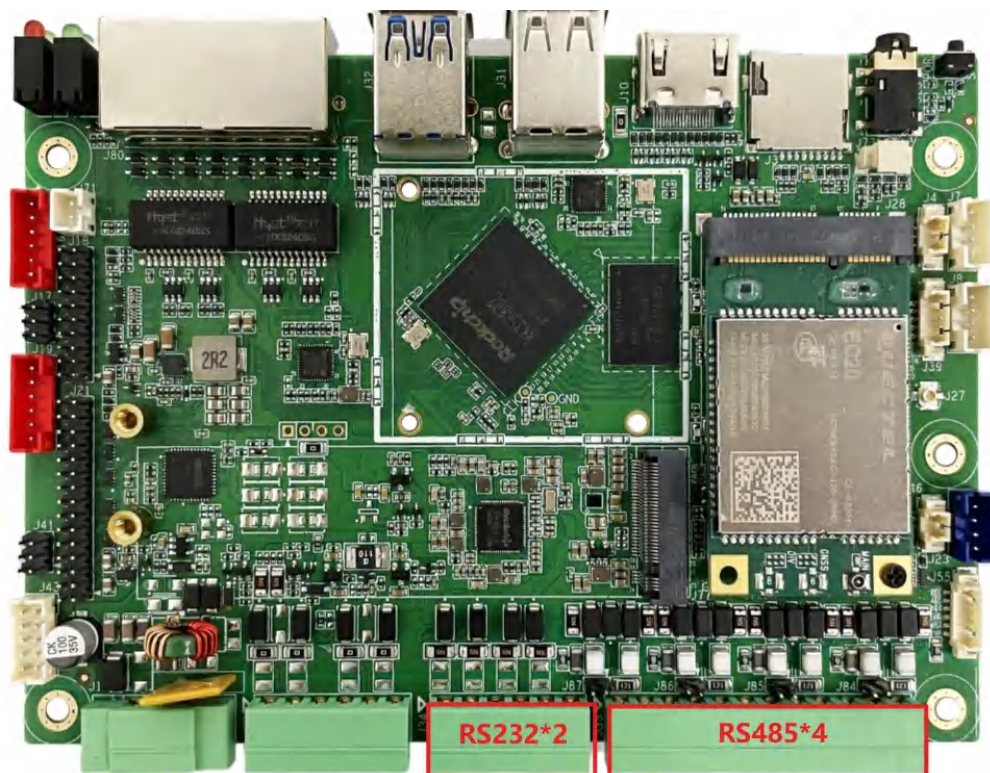
1.3 SSH调试

ssh远程调试账号:root, 密码:rockchip。

```
login as: root
root@192.168.1.122's password:
[root@RK356X:~]# ls
[root@RK356X:~]# ls /
bin          init         media  proc        sdcard      udisk
busybox.config  lib         misc   rockchip_test sys          userdata
data         lib64        mnt    root        system      usr
dev          linuxrc      oem    run         timestamp   var
etc          lost+found  opt    sbin        tmp         vendor
[root@RK356X:~]#
```

2 串口

主板共配置6路串口（调试串口除外），其中4路RS485，2路RS232。



各个串口对应的设备节点如下表：

编号	设备节点	类型	位置
1	/dev/ttyS0	RS485	J33,R0-T0
2	/dev/ttyS3	RS485	J33,R3-T3
3	/dev/ttyS4	RS485	J33,,R4-T4
4	/dev/ttyS5	RS485	J33,R5-T5
5	/dev/ttyS7	RS232	J34,R7-T7
6	/dev/ttyS9	RS232	J34,R9-T9

2.1 测试

RS485和RS232都可以使用microcom工具进行简单的收发测试。

以测试/dev/ttyS0为例：

```
1 [root@RK356X:/]# microcom -s 115200 /dev/ttyS0
```

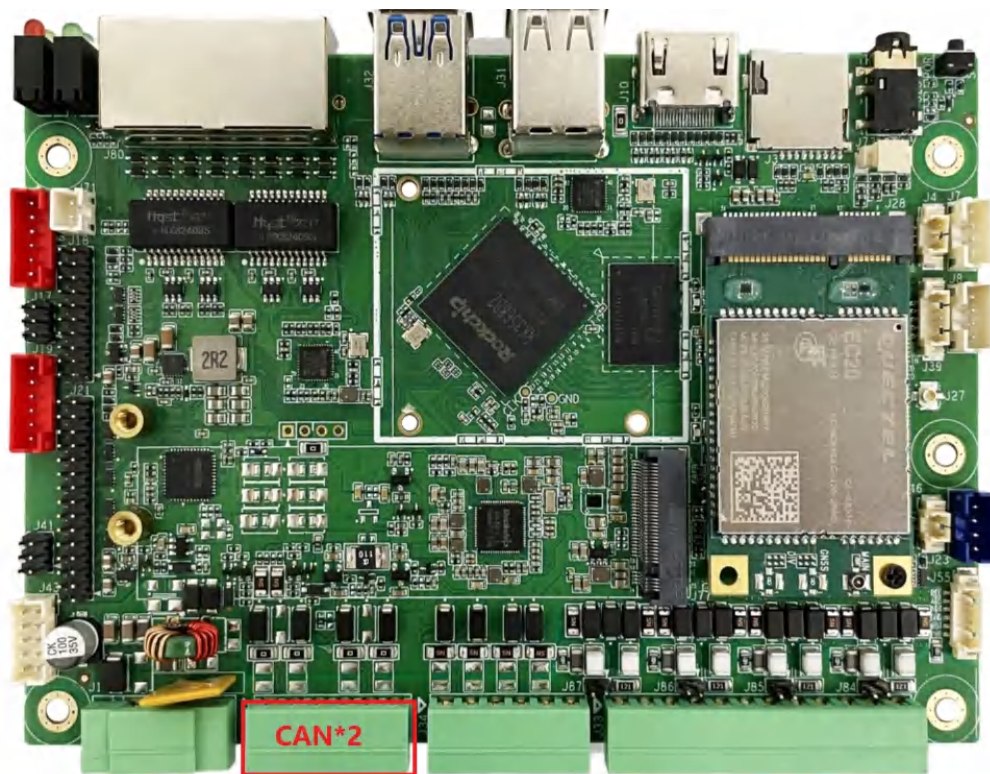
按下键盘任意键会发送对应的字符，而接收的内容会显示在终端。

按【ctrl】和【x】组合键，然后输入quit退出测试。

```
1 [root@RK356X:/]# microcom -s 115200 -p /dev/ttyS0
2 [root@RK356X:/]#
```

3 CAN

主板共配置2路CAN接口，系统中对应的网络节点分别为can0和can1。



2.1 测试

打开can设备，并设置通信波特率为250Kbps：

```
1 ip link set can0 down
2 ip link set can0 type can bitrate 250000
3 ip link set can0 up
4
5 ip link set can1 down
6 ip link set can1 type can bitrate 250000
7 ip link set can1 up
```

发送测试:

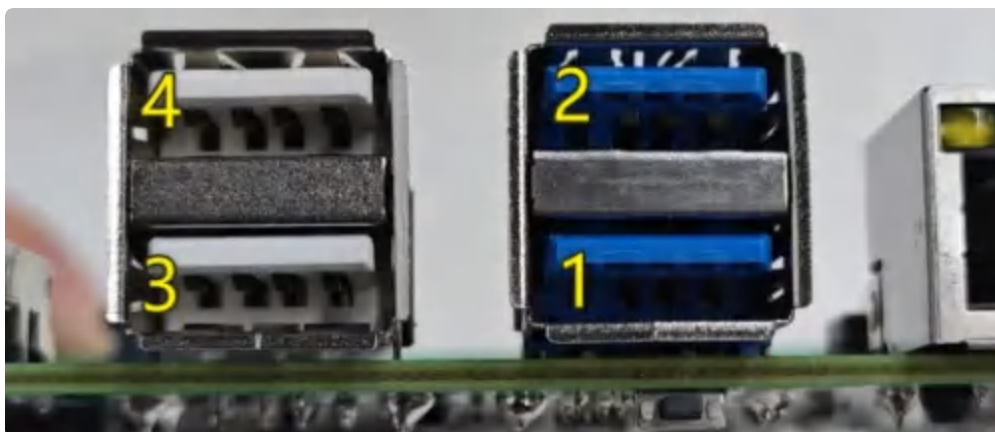
```
1 cansend can0 123#DEADBEEF
```

接收测试:

```
1 candump can0
```

3 USB

主板共配置6路USB接口，这里分别标记USB1-6，其中2路USB3.0（USB1和USB2），4路USB2.0（USB3-6），各个USB口位置如下图所示：





2.1 电源控制

USB2-6支持软件控制电压输出，控制节点见下表：

序号	功能	控电节点
2	USB 3.0 HOST	/sys/class/leds/usb3_host_pwr/brightness
3	USB 2.0 HOST	/sys/class/leds/usb2_host_pwr/brightness
4	USB 2.0 HOST	/sys/class/leds/usb2_fe2_pwr/brightness
5	USB 2.0 HOST	/sys/class/leds/usb2_fe3_pwr/brightness
6	USB 2.0 HOST	/sys/class/leds/usb2_fe4_pwr/brightness

供电控制说明，序号1写"1"关闭电源，写"0"开启电源；序号2-序号6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，以序号6为例

▼Shell

```
1 #关闭
2 echo 0 > /sys/class/leds/usb2_fe4_pwr/brightness
3 #开启（默认状态）
4 echo 1 > /sys/class/leds/usb2_fe4_pwr/brightness
```

2.2 U盘的挂载

USB端口插上U盘，会默认挂载到/media/usbx目录下：

```
1 [root@RK356X:/]# mount
2 ...
3 /dev/sda1 on /media/usb0 type fuseblk (rw,nosuid...)
```

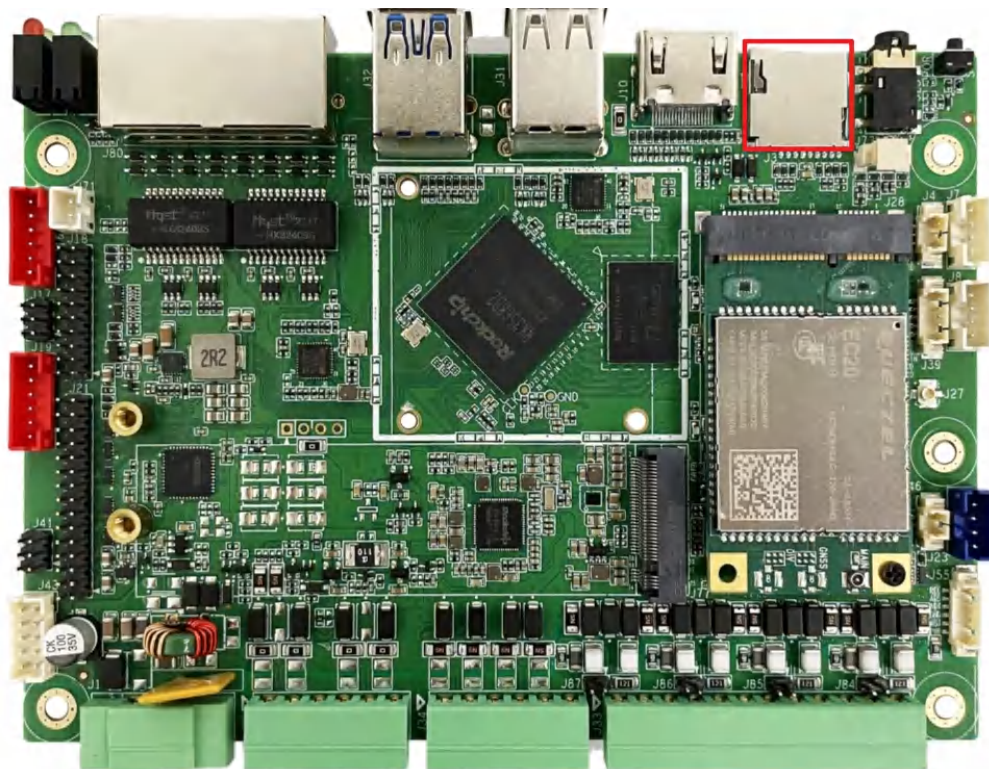
2.3 USB OTG 切换命令

USB OTG 支持host 和device 模式的切换，软件切换方法如下

```
1 //切换HOST模式
2 echo host > /sys/devices/platform/fe8a0000.usb2-phy/otg_mode
3 echo 255 > /sys/class/leds/usb3_otg_pwr/brightness
4
5 //切换DEVICE模式
6 echo peripheral > /sys/devices/platform/fe8a0000.usb2-phy/otg_mode
7 echo 0 > /sys/class/leds/usb3_otg_pwr/brightness
```

3 MicroSD

主板共配置1个SD接口：



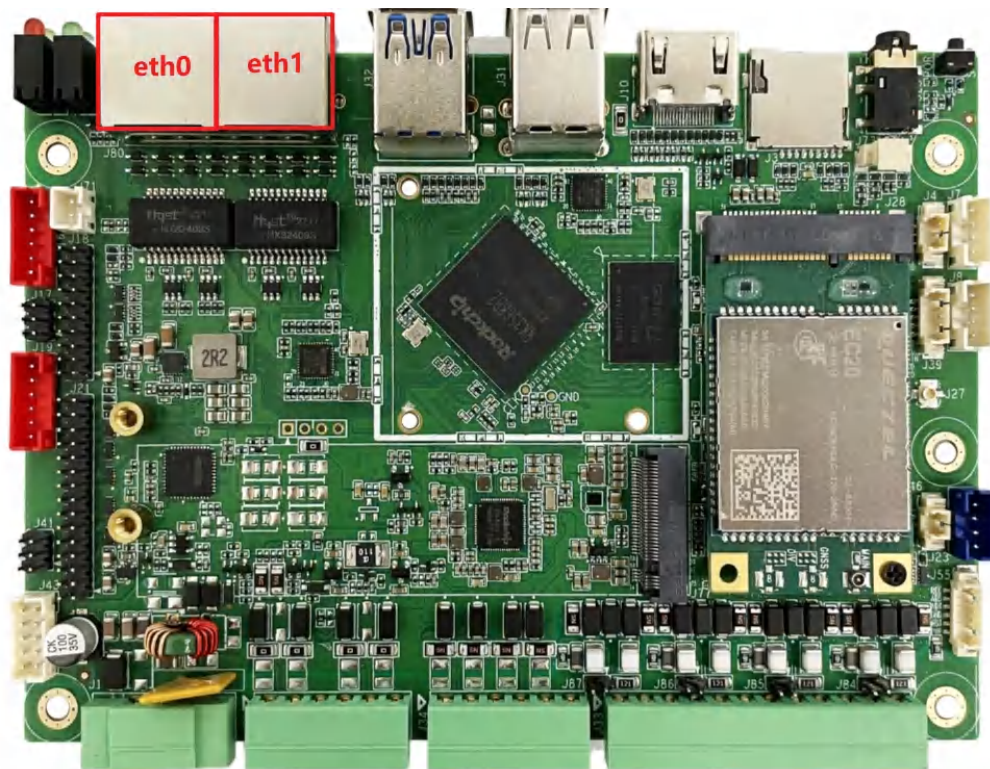
插入SD卡后，自动挂载到/mnt/sdcard目录下，可以使用mount命令确认：

Bash

```
1 [root@RK356X:/]# mount
2 ...
3 /dev/mmcblk1p1 on /mnt/sdcard type vfat (rw,noatime...)
```

4 以太网

主板共配置2路1000M以太网接口，系统中对应的网络节点分别为eth0和eth1：



4.1 测试

以下以eth1为例（eth0同样的操作方法）。

查看eth1的ip:

Bash |

```
1 [root@RK356X:~]# ifconfig eth1
2 eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
3     inet 192.168.1.177 netmask 255.255.255.0 broadcast 192.168.1.255
4     inet6 fe80::4f32:fc2d:eb30:220b prefixlen 64 scopeid 0x20<link>
5     ether d2:e1:8d:d5:f7:49 txqueuelen 1000 (Ethernet)
6     RX packets 78 bytes 11727 (11.4 KiB)
7     RX errors 0 dropped 0 overruns 0 frame 0
8     TX packets 60 bytes 5822 (5.6 KiB)
9     TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
10    device interrupt 52
11
```

设置eth1的ip:

Bash |

```
1 [root@RK356X:~]# ifconfig eth1 192.168.1.123
```

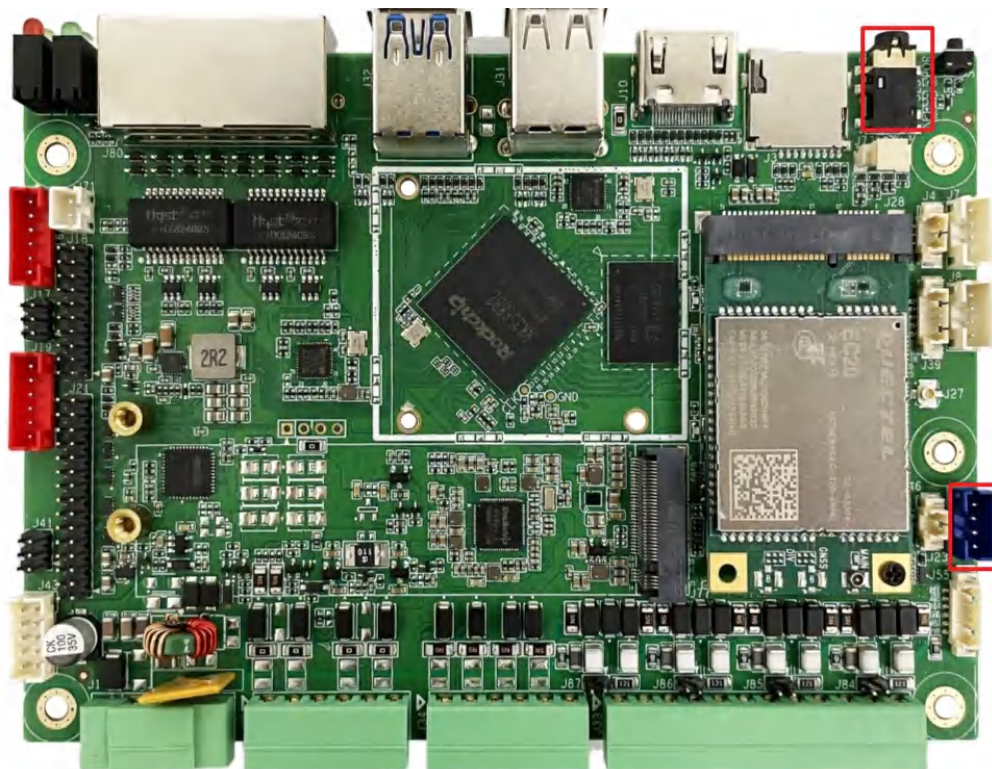
eth1 ping测试:

Bash

```
1 [root@RK356X:~]# ping www.baidu.com -I eth1
2 PING www.a.shifen.com (14.215.177.39) from 192.168.1.177 eth1: 56(84) byte
  s of data.
3 64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=1 ttl=54 time=6.75 ms
4 64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=2 ttl=54 time=6.32 ms
5 64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=3 ttl=54 time=7.01 ms
6 64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=4 ttl=54 time=7.47 ms
7 64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=5 ttl=54 time=7.01 ms
```

5 Lineout

主板支持LineOut音频输出和HDMI音频输出，LineOut音频接口位于J32:



注：默认LineOut和耳机的音频输出是同一路，即LineOut播放同时耳机也会播放。

5.1 查看声卡设备

```

1 [root@RK356X:~]# aplay -l
2 **** List of PLAYBACK Hardware Devices ****
3 card 0: rockchipdmi [rockchip,dmi], device 0: fe400000.i2s-i2s-hifi i2s-hifi-0 [fe400000.i2s-i2s-hifi i2s-hifi-0]
4   Subdevices: 1/1
5   Subdevice #0: subdevice #0
6 card 1: rockchiprk809co [rockchip,rk809-codec], device 0: fe410000.i2s-rk817-hifi rk817-hifi-0 [fe410000.i2s-rk817-hifi rk817-hifi-0]
7   Subdevices: 1/1
8   Subdevice #0: subdevice #0

```

5.2 播放音频

5.2.1 播放到HDMI

```

1 aplay -D plughw:0,0 /etc/bsa_file/8k16bpsStereo.wav

```

5.2.2 播放到Lineout

打开Lineout开关：

```

1 amixer -c 1 cset numid=2,iface=MIXER,name='Playback Path' 'SPK'

```

不插入耳机，执行以下命令播放音频到Lineout：

```

1 aplay -D plughw:1,0 /etc/bsa_file/8k16bpsStereo.wav

```

5.2.3 播放到耳机

设置只输出到耳机：

Bash

```
1 amixer -c 1 cset numid=2,iface=MIXER,name='Playback Path' 'HP'
```

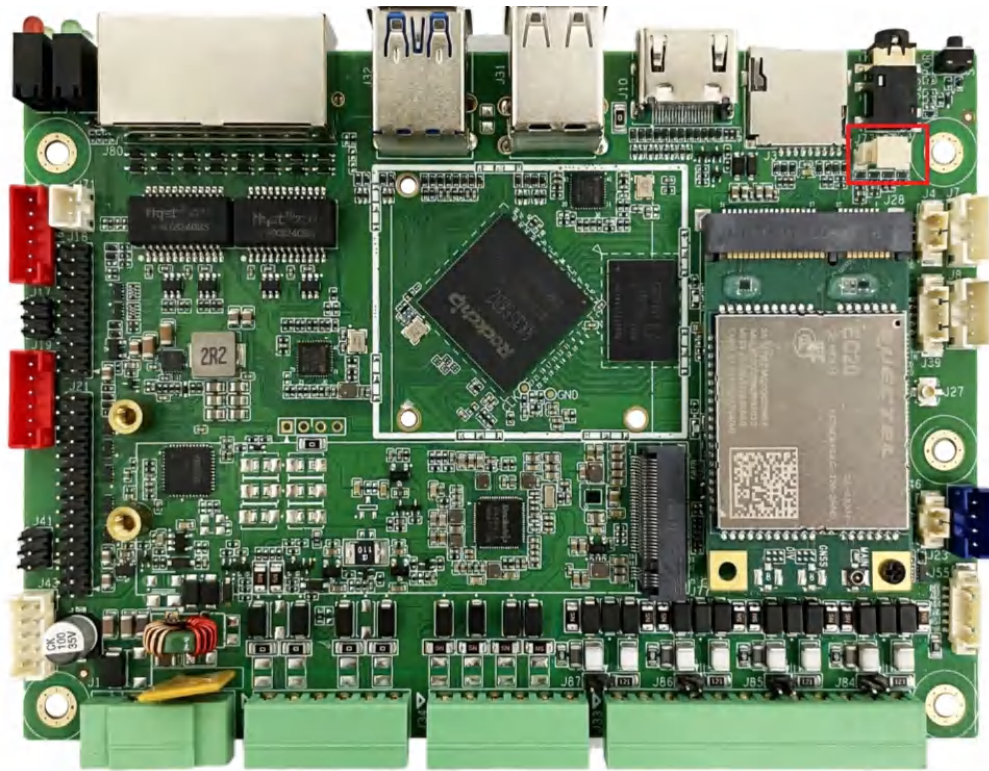
插入耳机，执行以下命令播放音频到耳机：

Bash

```
1 aplay -D plughw:1,0 /etc/bsa_file/8k16bpsStereo.wav
```

6 MIC

主板配置一路模拟麦克风接口，位于J7：



6.1 录音测试

打开录音开关：

Bash

```
1 amixer -c 1 cset numid=3,iface=MIXER,name='Capture MIC Path' 'Main Mic'
```

使用arecord工具可以进行录音测试：

```
▼ Bash |
1 [root@RK356X:/]# 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
4 ^CAborted by signal 中断...
```

录音完后播放测试：

```
▼ Bash |
1 [root@RK356X:/]# aplay -D plughw:1,0 ./test.wav
2 Playing WAVE './test.wav' : Signed 16 bit Little Endian, Rate 48000 Hz, Stereo
```

7 WiFi

主板配置了一路2.4G WiFi模块，型号为AW-NM372SM。

系统上电默认会打开WiFi，对应的网络设备节点为wlan0。

```
▼ Bash |
1 [root@RK356X:/]# 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
```

7.1 连接WiFi热点

使用脚本wifi_start.sh连接WiFi热点：

```
▼ Bash |
1 wifi_start.sh "TP-LINK_B87A" "12345678"
```

其中"TP-LINK_B87A"为热点名称, "12345678"为热点密码。

```
1 [root@RK356X:/]# ifconfig wlan0
2 wlan0      Link encap:Ethernet  HWaddr A8:41:F4:4F:B1:6D
3            inet addr:192.168.1.178  Bcast:192.168.1.255  Mask:255.255.255.0
4            inet6 addr: fe80::af5b:f8e4:625e:91f9/64 Scope:Link
5            UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
6            RX packets:5 errors:0 dropped:27 overruns:0 frame:0
7            TX packets:38 errors:0 dropped:0 overruns:0 carrier:0
8            collisions:0 txqueuelen:1000
9            RX bytes:1010 (1010.0 B)  TX bytes:5221 (5.0 KiB)
```

8 Bluetooth

系统开机默认打开蓝牙, 对应的网络节点为hci0。

```
1 [root@RK356X:/]# hciconfig hci0 up
2 [root@RK356X:/]# hciconfig
3 hci0:      Type: Primary  Bus: UART
4            BD Address: 28:51:03:8D:A2:01  ACL MTU: 1021:8  SCO MTU: 64:1
5            UP RUNNING
6            RX bytes:668 acl:0 sco:0 events:34 errors:0
7            TX bytes:423 acl:0 sco:0 commands:34 errors:0
```

8.1 连接蓝牙设备

使用命令行连接蓝牙设备, 命令如下:

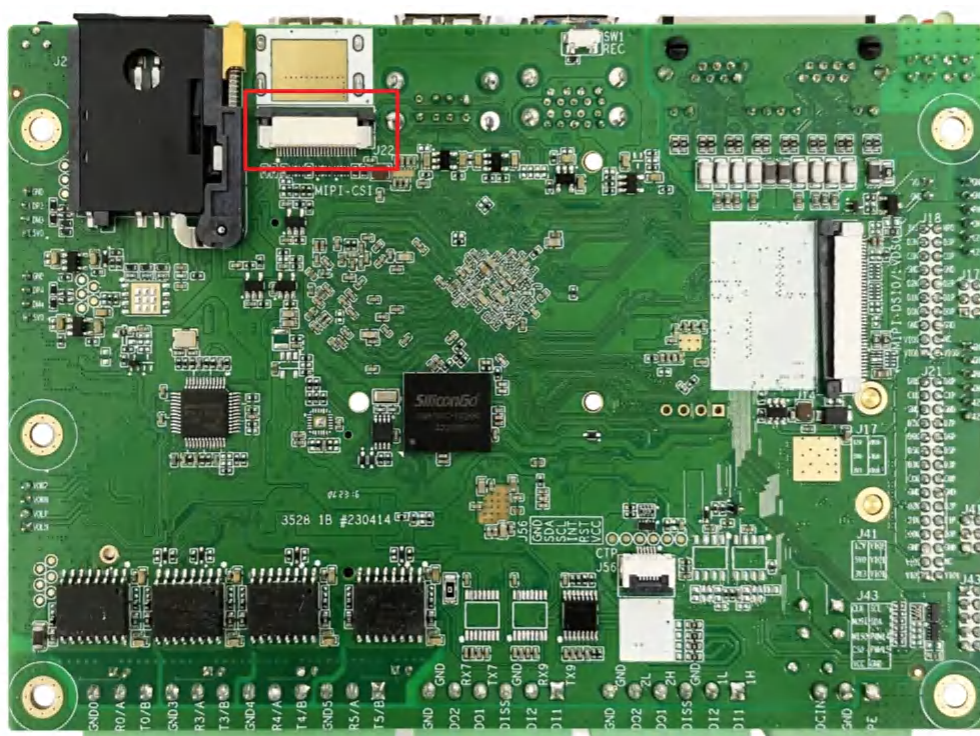
```
1 [root@RK356X:/]# bluetoothctl
2 [bluetooth]# scan on
3 ...
4 [bluetooth]# trust 7C:C1:80:09:DD:6C
5 [bluetooth]# pair 7C:C1:80:09:DD:6C
6 [cainiaocl]# exit
```



```
1 [root@RK356X:/]# ping www.baidu.com -I wwan0
2 PING www.a.shifen.com (14.119.104.189) from 10.122.45.92 wwan0: 56(84) byte
  s of data.
3 64 bytes from 14.119.104.189 (14.119.104.189): icmp_seq=1 ttl=52 time=95.5
  ms
4 64 bytes from 14.119.104.189 (14.119.104.189): icmp_seq=2 ttl=52 time=99.3
  ms
5 64 bytes from 14.119.104.189 (14.119.104.189): icmp_seq=3 ttl=52 time=97.8
  mss
```

10 摄像头

主板配置了一路CSI摄像头，默认支持型号OV5648和OV8858，位于J22：



10.1 测试摄像头是否存在

```

1 [root@RK356X:/]# 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 4-0036":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 4-0036 (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]

```

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

10.2 预览摄像头画面

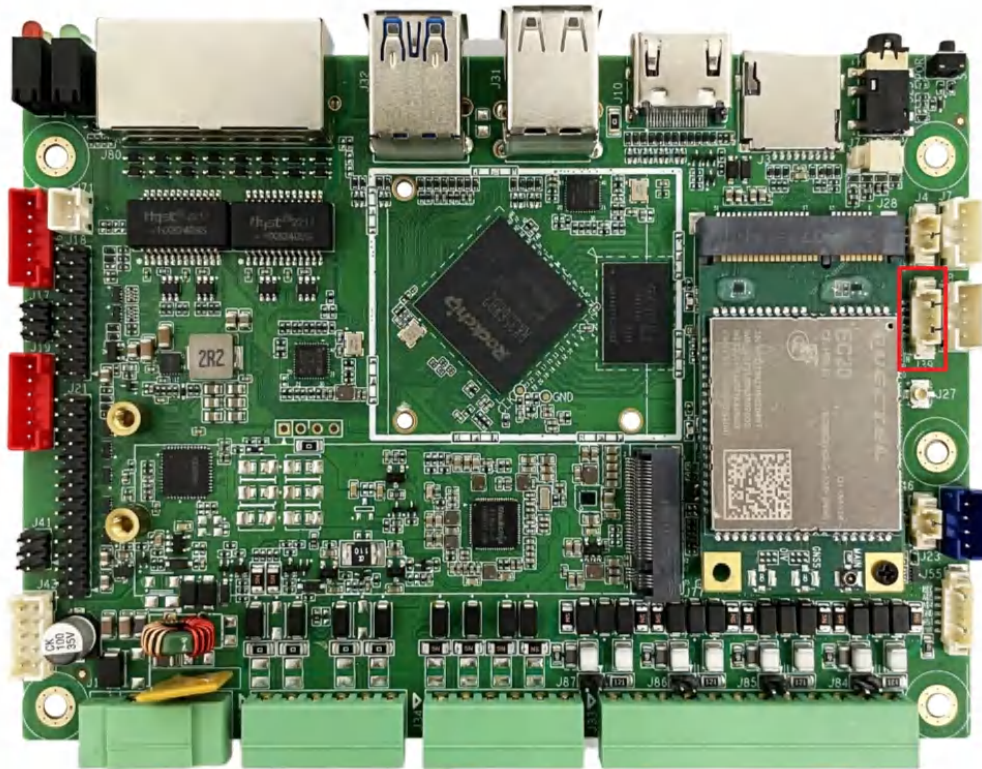
```

1 [root@RK356X:/]# gst-launch-1.0 v4l2src device=/dev/video0 ! video/x-raw, format=NV12, width=1920, height=1080, framerate=15/1 ! videoconvert ! autovideosink

```

11 红外

主板配置了一路红外接口，位于J39，支持NEC编码遥控器，默认适配的遥控器型号为HTR-A07，使用时需要安装好配套的红外扩展小板。





HTR-A07的键值如下表所示：

编号	按键	键值	编号	按键	键值
1	电源	KEY_POWER	21	1	KEY_1
2	TV	KEY_SCREEN	22	2	KEY_2
3	橙色	KEY_F1	23	3	KEY_3
4	绿色	KEY_F2	24	4	KEY_4
5	黄色	KEY_F3	25	5	KEY_5
6	紫色	KEY_F4	26	6	KEY_6
7	音量+	KEY_VOLUMEUP	27	7	KEY_7

8	音量-	KEY_VOLUMEDOWN	28	8	KEY_8
9	屏显	KEY_DISPLAY_OFF	29	9	KEY_9
10	静音	KEY_MUTE	30	TVNOW	KEY_DOT
11	上一节目	KEY_VIDEO_PREV	31	0	KEY_0
12	下一节目	KEY_VIDEO_NEXT	32	截屏	KEY_PRINT
13	上	KEY_UP	33	/	/
14	左	KEY_LEFT	34	/	/
15	下	KEY_DOWN	35	/	/
16	右	KEY_RIGHT	36	/	/
17	确认	KEY_ENTER	37	/	/
18	返回	KEY_BACK	38	/	/
19	主页	KEY_HOME	39	/	/
20	菜单	KEY_MENU	40	/	/

11.1 查看红外按键值

▼

Bash

```
1 echo 1 > /sys/module/rockchip_pwm_remotectl/parameters/code_print
```

12 ADC

主板共配置2路ADC，位于J55，丝印标记为AD4和AD5。



12.1 计算输入电压

使用如下公式计算输入电压：

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

raw为从设备节点读取的数值，两路AD对应的设备节点见下表：

AD4	/sys/bus/iio/devices/iio:device0/in_voltage4_raw
AD5	/sys/bus/iio/devices/iio:device0/in_voltage5_raw

举例：

AD4输入电压计算

```
▼ Bash |
1 cat /sys/bus/iio/devices/iio:device0/in_voltage4_raw
2 258
```

则AD4的输入电压 $V=(258/1024)*1.8\text{v}=0.454\text{v}$ 。

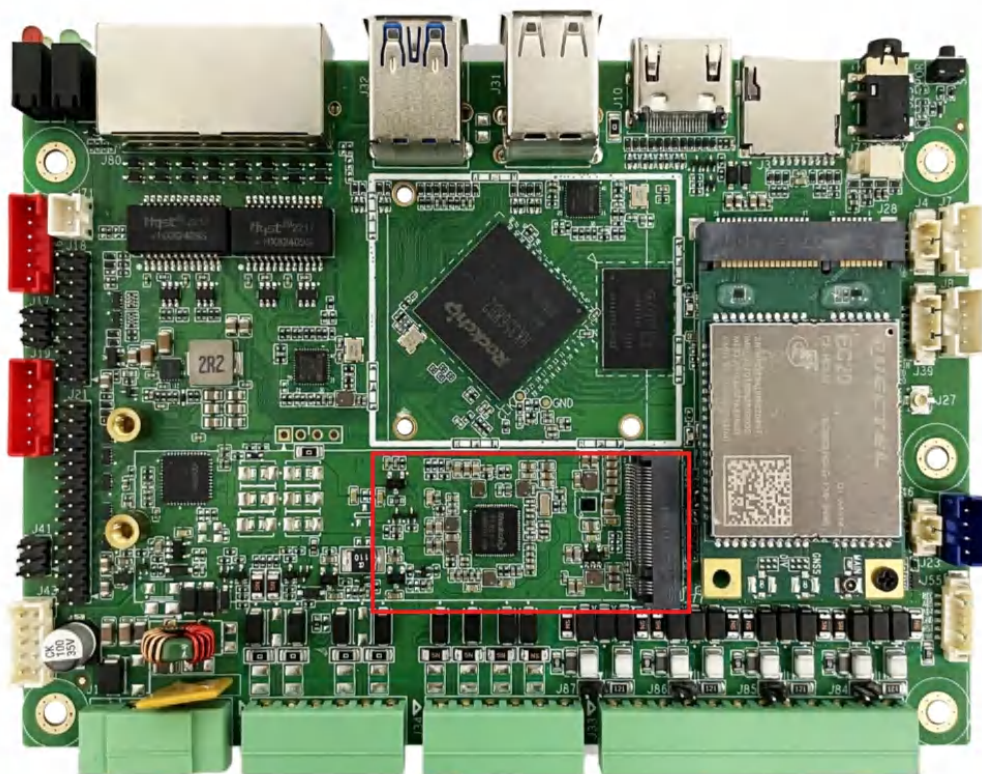
AD5输入电压计算：

```
1 cat /sys/bus/iio/devices/iio:device0/in_voltage5_raw
2 137
```

则AD5的输入电压 $V=(137/1024)*1.8v=0.241v$ 。

13 PCIE

主板配置了一路PCIE3.0接口，位于J77：



断电安装好PCIE固态硬盘，主板上电，系统启动后，硬盘将自动识别：

```

1 [root@RK356X:/]# fdisk -l
2 Found valid GPT with protective MBR; using GPT
3
4 Disk /dev/mmcblk0: 30535680 sectors, 2622M
5 Logical sector size: 512
6 Disk identifier (GUID): c7460000-0000-4b06-8000-5539000009bc
7 Partition table holds up to 128 entries
8 First usable sector is 34, last usable sector is 30535646
9
10 Number  Start (sector)    End (sector)  Size Name
11     1           16384           28671 6144K uboot
12     2           28672           36863 4096K misc
13     3           36864          102399 32.0M boot
14     4          102400          167935 32.0M recovery
15     5          167936          233471 32.0M backup
16     6          233472          757759 256M userdata
17     7          757760          1019903 128M oem
18     8          1019904          30535615 14.0G rootfs
19 Disk /dev/nvme0n1: 119 GB, 128035676160 bytes, 250069680 sectors
20 15566 cylinders, 255 heads, 63 sectors/track
21 Units: cylinders of 16065 * 512 = 8225280 bytes
22
23 Device      Boot StartCHS      EndCHS          StartLBA      EndLBA      Sector
24 /dev/nvme0n1p1  1,0,1      1023,254,63      2048  250069646  25006759
    9 119G   c Win95 FAT32 (LBA)

```

可以看到PCIE硬盘在系统中的设备节点为/dev/nvme0n1p1。

使用mount命令可以挂载到系统中进行文件操作：

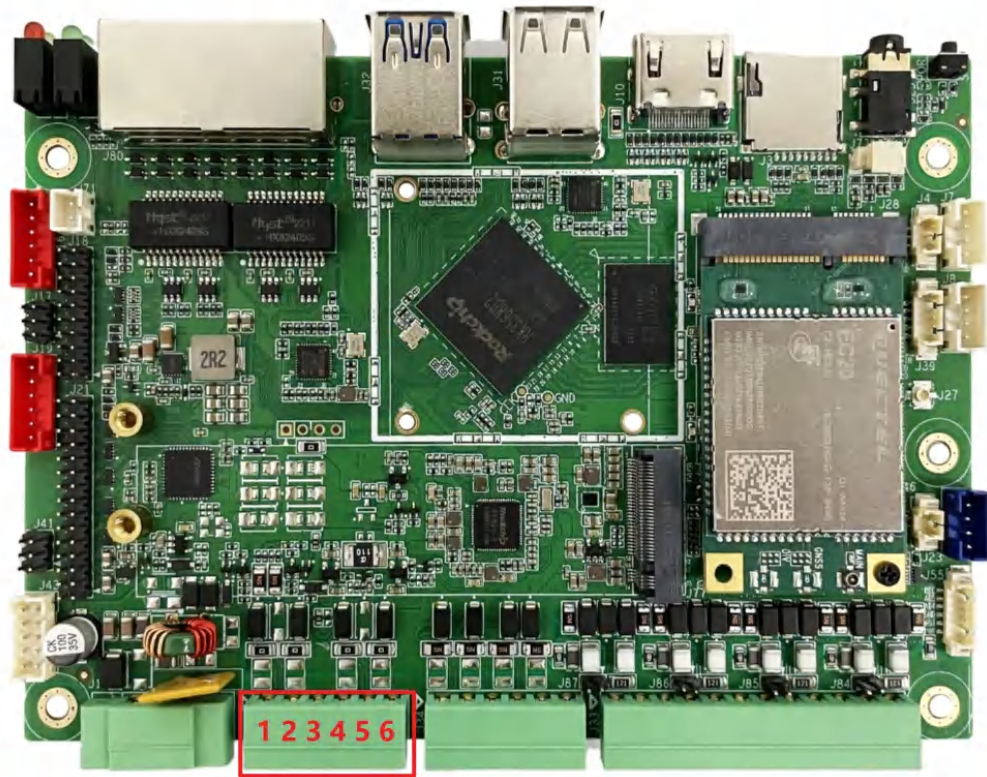
```

1 [root@RK356X:/]# mount /dev/nvme0n1p1 /mnt/sdcard/
2 [root@RK356X:/]# ls /mnt/sdcard/
3 Alarms      DCIM        Download  Movies  Notifications  Podcasts
4 Audiobooks  Documents  LOST.DIR  Music   Pictures        Ringtones
5 [root@RK356X:/]# mount
6 /dev/mmcblk0p8 on / type ext4 (rw,relatime)
7 ...
8 /dev/nvme0n1p1 on /mnt/sdcard type vfat (rw,relatime...)

```

14 DIDO

主板共配置2路DI和2路DO，位于J35（与CAN接口复用，二选一）：



各个引脚定义如下表所示：

引脚	功能
1	DI1
2	DI2
3	DI_SS
4	DO1
5	DO2
6	GND

14.1 DI的使用

通过读取设备节点/dev/ido_di，获取DI1和DI2的输入电平状态。打开/dev/ido_di设备文件，读取3个字节，前2个字节为两路DI的输入结果。

通过代码读取举例：

```

1  #include <stdio.h>
2  #include <sys/types.h>
3  #include <sys/stat.h>
4  #include <fcntl.h>
5  #include <unistd.h>
6  #include <stdlib.h>
7
8  int main(void)
9  {
10     char buf[3] = {0, 0, 0};
11
12     int fd;
13     int ret;
14
15     fd = open("/dev/ido_di", O_RDONLY);
16
17     if (fd < 0){
18         exit(1);
19     }
20
21     ret = read(fd, buf, 3);
22     if (ret != 3){
23         perror("read");
24         close(fd);
25         exit(2);
26     }
27
28     printf("DI1输入%s\n", buf[0] == '1' ? "低电平" : "高电平");
29     printf("DI2输入%s\n", buf[1] == '1' ? "低电平" : "高电平");
30
31     close(fd);
32
33     return 0;
34 }

```

通过命令行来获取举例：

[illegible]

返回结果前2个字符与DI1、DI2的输入关系见下表：

结果	DI1输入	DI2输入
00	高	高
10	低	高
01	高	低
11	低	低

14.2 DO的使用

通过向/dev/ido_do设备文件写入特定字符串，控制DO1和DO2的输出。

通过代码控制输出举例：

```

1  #include <stdio.h>
2  #include <sys/types.h>
3  #include <sys/stat.h>
4  #include <fcntl.h>
5  #include <unistd.h>
6  #include <stdlib.h>
7
8  int main(void)
9  {
10         int fd;
11         int ret;
12
13         fd = open("/dev/ido_do", O_WRONLY);
14
15         if (fd < 0){
16             exit(1);
17         }
18
19         //D01输出低电平
20         printf("D01 输出低电平\n");
21         ret = write(fd, "1D", 3);
22         if (ret != 3)
23             goto write_err;
24         sleep(3);
25         //D01输出高电平
26         printf("D01 输出高电平\n");
27         ret = write(fd, "1E", 3);
28         if (ret != 3)
29             goto write_err;
30         sleep(3);
31
32         //D02输出低电平
33         printf("D02 输出低电平\n");
34         ret = write(fd, "2D", 3);
35         if (ret != 3)
36             goto write_err;
37         sleep(3);
38
39         //D01输出高电平
40         printf("D02 输出高电平\n");
41         ret = write(fd, "2E", 3);
42         if (ret != 3)
43             goto write_err;
44         sleep(3);

```

```

45
46         close(fd);
47
48         return 0;
49
50 write_err:
51         close(fd);
52         return -1;
53     }

```

通过命令行来设置输出举例：

```

▼ Shell |
1 //D01输出0V
2 echo 1D > /dev/ido_do
3 //D01输出12V
4 echo 1E > /dev/ido_do
5
6 //D02输出0V
7 echo 2D > /dev/ido_do
8 //D02输出12V
9 echo 2E > /dev/ido_do

```

15 外部看门狗

主板配置了外部看门狗，用于监控系统是否正常运行，当系统出现异常如卡死等情况，外部看门狗将起作用，对系统进行复位重启。

15.1 开启与关闭

开启开门狗功能：

```

▼ Bash |
1 echo 255 > /sys/class/leds/ido_wdt_en/brightness

```

关闭看门狗功能：

```

▼ Bash |
1 echo 0 > /sys/class/leds/ido_wdt_en/brightness

```

15.2 测试

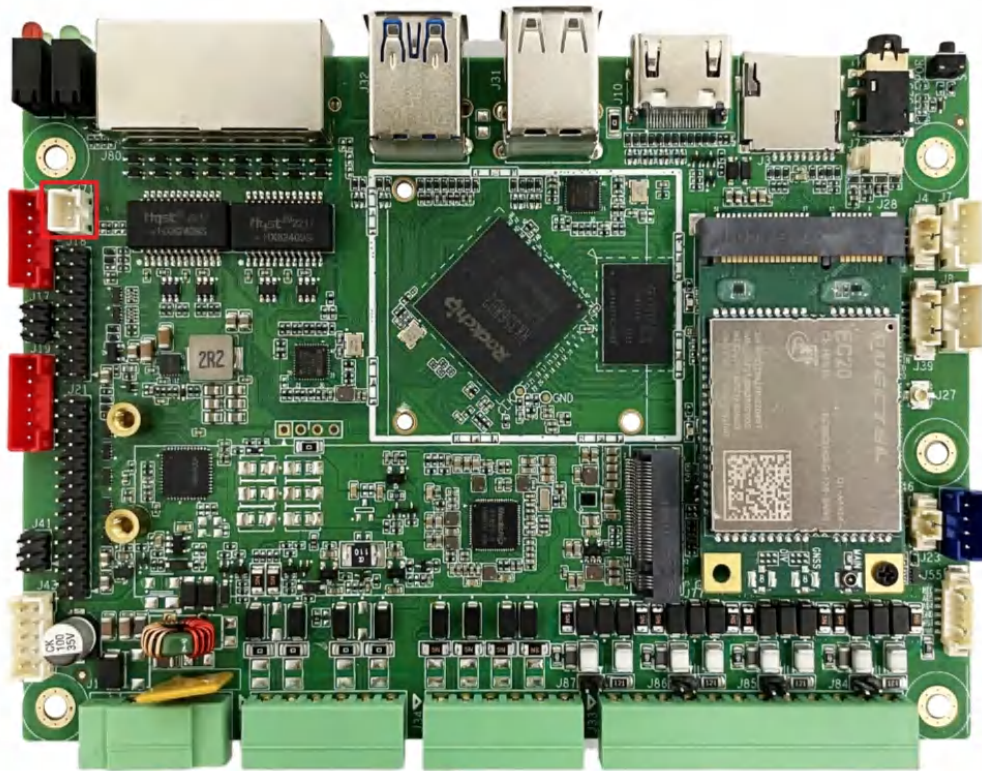
以下指令模拟系统卡死的情况，测试外部看门狗功能：

```
1 echo c > /proc/sysrq-trigger
```

执行完该命令，系统进入卡死状态，约5分钟后，系统将重启。

16 FAN

主板配置了一路风扇电源接口（输出5v/12v可选），位于J71：



16.1 开启与关闭

打开输出：

```

1 echo 27 > /sys/class/gpio/export
2 echo out > /sys/class/gpio/gpio27/direction
3 echo 1 > /sys/class/gpio/gpio27/value

```

关闭输出：

```

1 echo 0 > /sys/class/gpio/gpio27/value

```

17 LED

主板共配置4个LED指示灯：



各个指示灯的作用见下表：

编号	名称	功能
1	POWER_LED	电源指示灯，亮起表示主板供电正常
2	SYS_LED	系统指示灯，闪烁表示系统运行正常
3	USER_LED	预留用户控制
4	4G_LED	4G通信指示灯，闪烁表示4G正在通信

USER_LED控制方法如下：

```
1 //USER_LED灭
2 echo 0 > /sys/class/leds/user_led/brightness
3
4 //USER_LED亮
5 echo 255 > /sys/class/leds/user_led/brightness
```

18 RTC

主板配置2路RTC，一路为HYM8563（rtc0），一路为芯片内部RK809 RTC（rtc1）。

18.1 读取与设置时间

读取RTC时间：

```
1 [root@RK356X:/]# hwclock
2 2020-01-01 12:04:13.276756+00:00
```

设置RTC时间：

```
1 [root@RK356X:/]# date -s '2023-4-26 17:38:00'
2 2023年 04月 26日 星期三 17:38:00 UTC
3 [root@RK356X:/]# hwclock -w
4 [root@RK356X:/]# hwclock
5 2023-04-26 17:38:07.554898+00:00
```

18.2 定时开机

RK809 RTC支持定时开机功能，定时开机功能的使用参考以下步骤：

```
1 //设置300秒后开机
2 echo +300 > /sys/class/rtc/rtc1/wakealarm
3
4 //立刻关机
5 poweroff
```

执行以上代码，关机后，等待300秒会自动开机。

19 输入电压检测

输入电压（主板供电电压）读取节点为/sys/bus/iio/devices/iio:device0/in_voltage6_raw，通过读取该节点的值，来计算输入电压的大小。

节点读取方法为：

```
1 cat /sys/bus/iio/devices/iio:device0/in_voltage6_raw
```

输入电压为V与节点读取值W的关系如下：

$$V=(W/1024)*1.8*21$$

假设节点读取值为333，则主板输入电压为 $V=(333/1024)*1.8*21=12.29v$ 。