

# IDO-EVB3568-V1 Linux5.10(IDO)使用手册

---



## IDO-EVB3568-V1 Linux5.10(IDO)使用手册

深圳触觉智能科技有限公司

[www.industio.cn](http://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、拷贝IDO-EVB3568-V1 Linux5.10使用手册 | TWX |    | 2024/11/14 |
| V1.1 | V1D    | 1、增加RTC定时开机描述                    | TWX |    | 2025/03/04 |
| V1.2 | V1D    | 1、修正Lineout声卡编号                  | TWX |    | 2025/03/13 |
| V1.3 | V1D    | 1、修正Lineout声卡编号                  | LJS |    | 2025/11/20 |
|      |        |                                  |     |    |            |
|      |        |                                  |     |    |            |
|      |        |                                  |     |    |            |
|      |        |                                  |     |    |            |
|      |        |                                  |     |    |            |

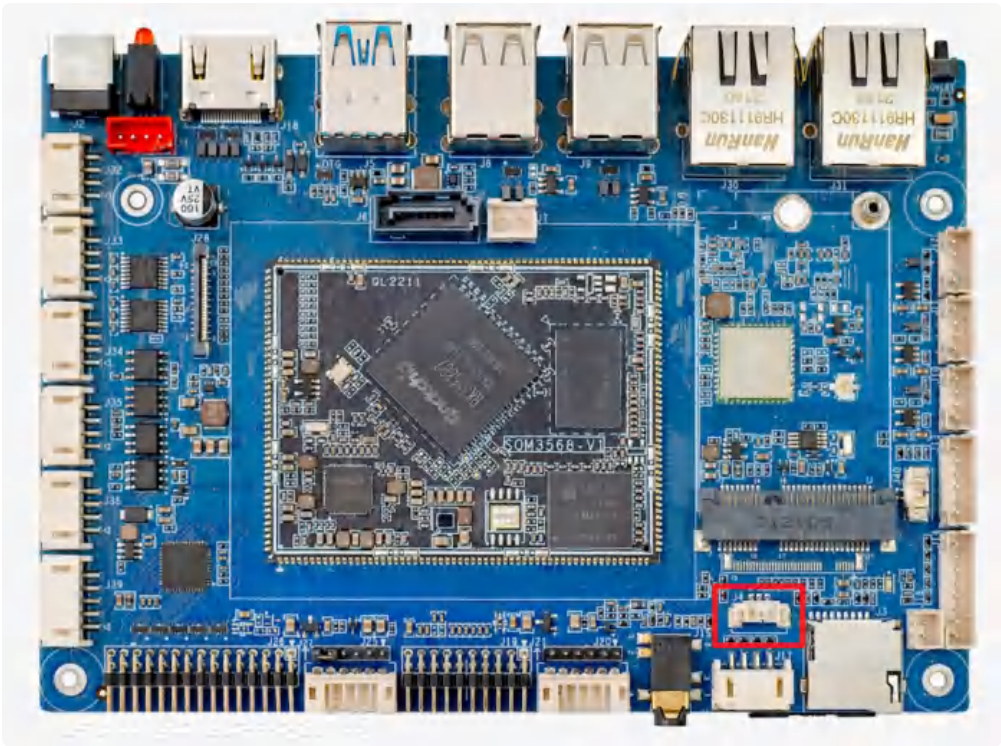
IDO-EVB3568-V1 Ubuntu 系统

1 调试

IDO-EVB3568-V1开发板支持串口调试、ADB调试和远程SSH调试。

# 1.1 串口调试

串口调试接口位于J4端口，见下图。请使用配套的USB串口调试工具。



调试串口为TTL电平，通信参数为1500000 8 N 1。

1. 打开MobaXterm，下载链接如下：

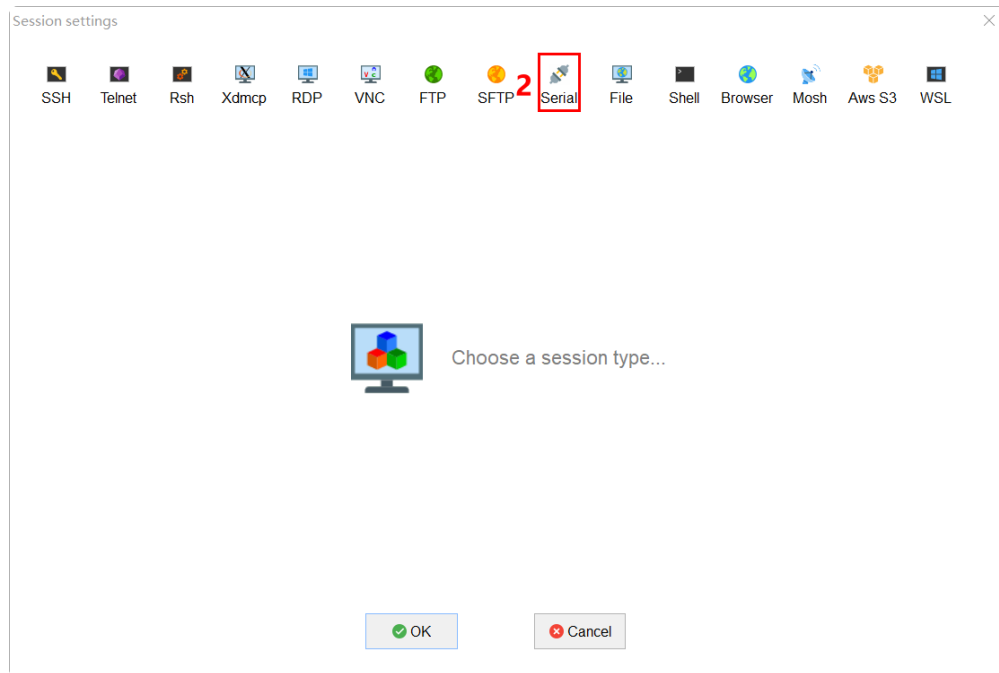
链接：[https://pan.baidu.com/s/11ui4LTd2mq\\_9kiJpeL4bWg?pwd=1234](https://pan.baidu.com/s/11ui4LTd2mq_9kiJpeL4bWg?pwd=1234)

提取码：1234

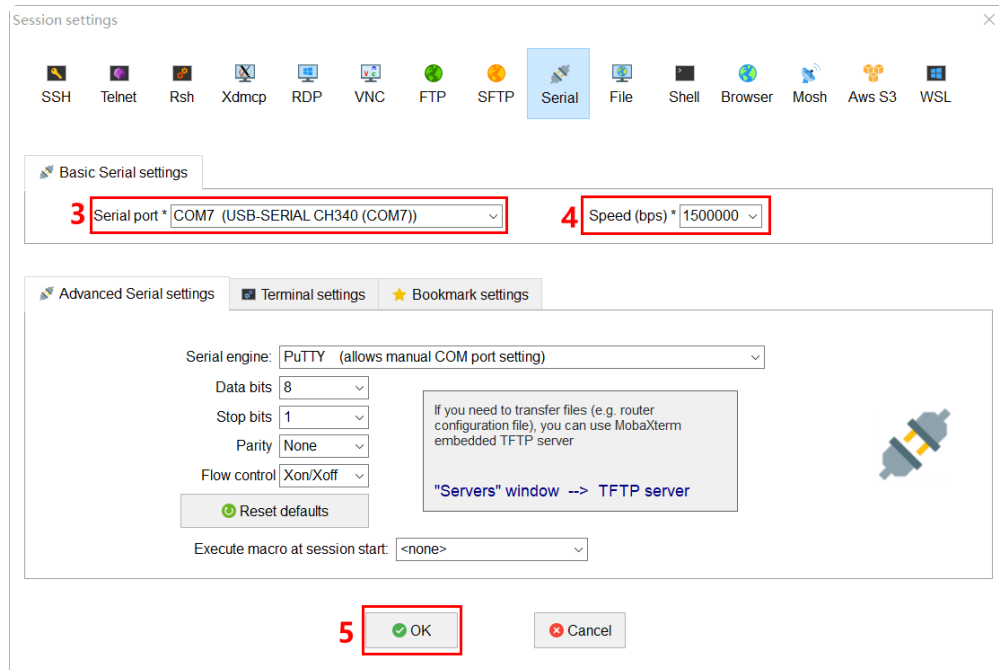
| <input type="checkbox"/> 文件名                          | ↓ 修改时间           | 类型    | 大小      |
|-------------------------------------------------------|------------------|-------|---------|
| <input type="checkbox"/> RKDevTool_Release_v2.95.zip  | 2024-04-24 11:53 | zip文件 | 2.30MB  |
| <input type="checkbox"/> other_tools.txt              | 2024-04-25 15:31 | txt文件 | 44B     |
| <input type="checkbox"/> MobaXterm_Portable_v23.6.zip | 2024-04-24 14:30 | zip文件 | 39.99MB |
| <input type="checkbox"/> DriverAssitant_v5.11.zip     | 2024-04-24 11:52 | zip文件 | 9.36MB  |

1

2. 选择sessionSerial，如下图所示：



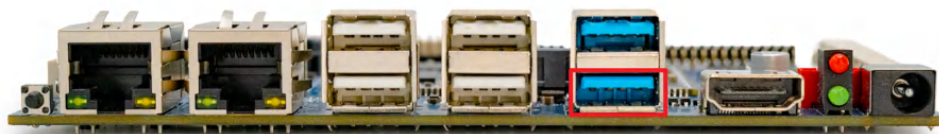
3. 将Serial port修改为在设备管理器中找到的COM端口
4. 设置Speed(bps)为1500000
5. 点击【OK】按钮，如下图所示：



结果如下图所示：

```
root@ido:~# ls /
bin  dev  lib          mnt  root  sdcard  system  usr
boot  etc  lost+found  opt  run  srv     tmp     var
data  home media       proc /sbin  sys     udisk   vendor
root@ido:~#
```

## 1.2 ADB调试



上图红色框内的USB接口为支持OTG模式切换，使用双公头 USB 数据线连接开发板和 PC 端的 USB接口，在PC终端识别到 ADB 设备，即可使用 adb shell 调试。

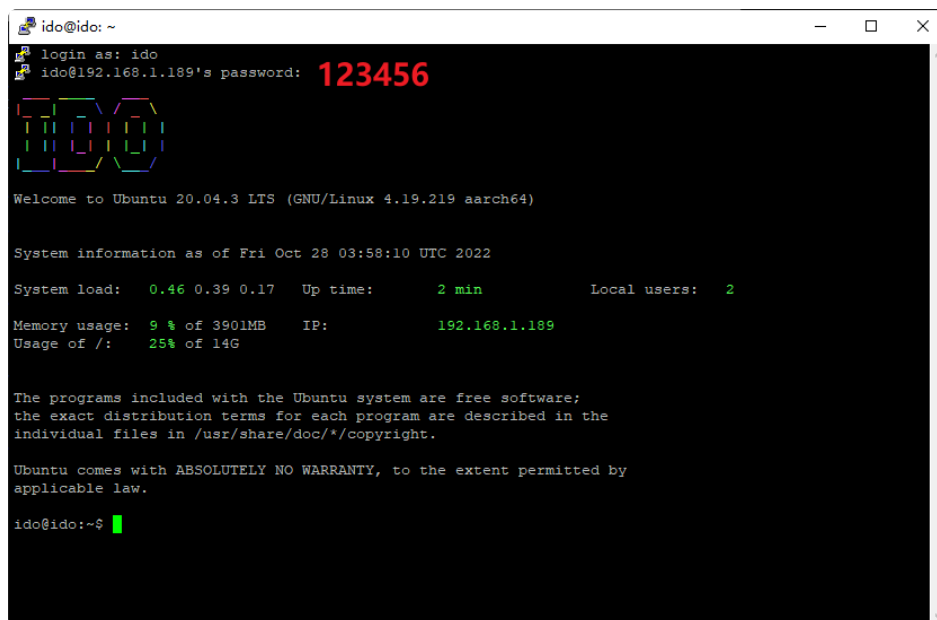
```
root@ido:/# ls
bin    dev    lib      mnt     root    sdcard  system  usr
boot   etc    lost+found  opt     run     srv     tmp     var
data   home   media    proc    sbin    sys     udisk   vendor
root@ido:/#
```

## 1.3 SSH调试

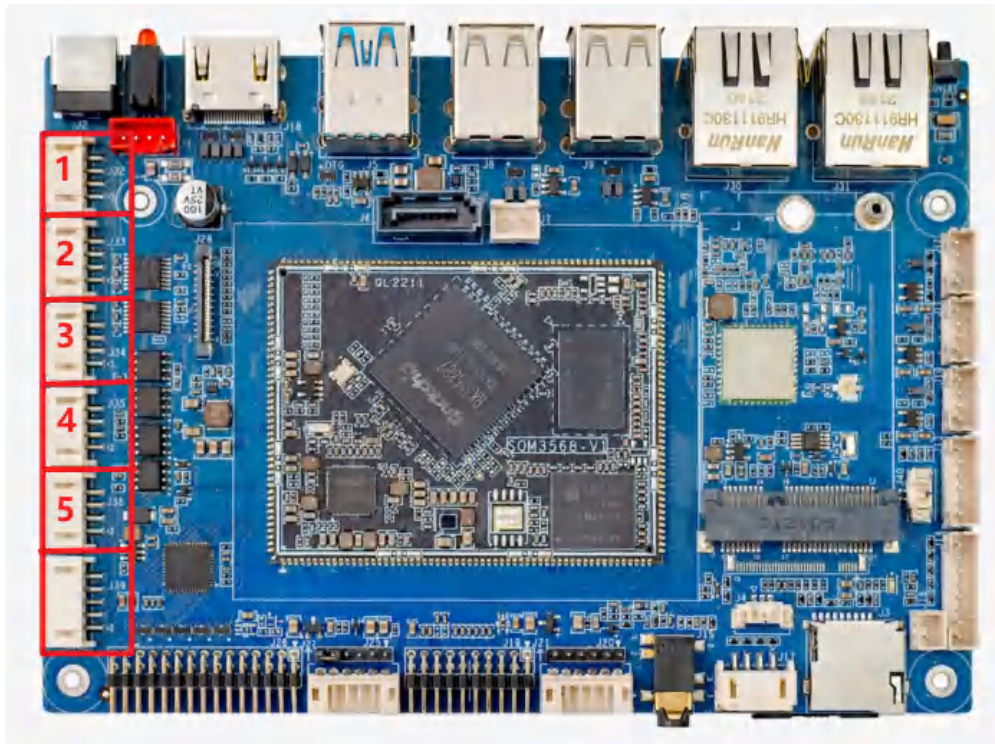
SSH登录账号信息：

账号：ido

密码：123456



## 2 串口



串口接口位置及引脚定义如上图所示，设备节点如下表所示：

| 序号 | 功能    | 设备节点       |
|----|-------|------------|
| 1  | TTL   | /dev/ttyS0 |
| 2  | RS232 | /dev/ttyS3 |
| 3  | RS232 | /dev/ttyS4 |
| 4  | RS485 | /dev/ttyS5 |
| 5  | RS485 | /dev/ttyS7 |

使用工具microcom，可以进行发送和接收测试。

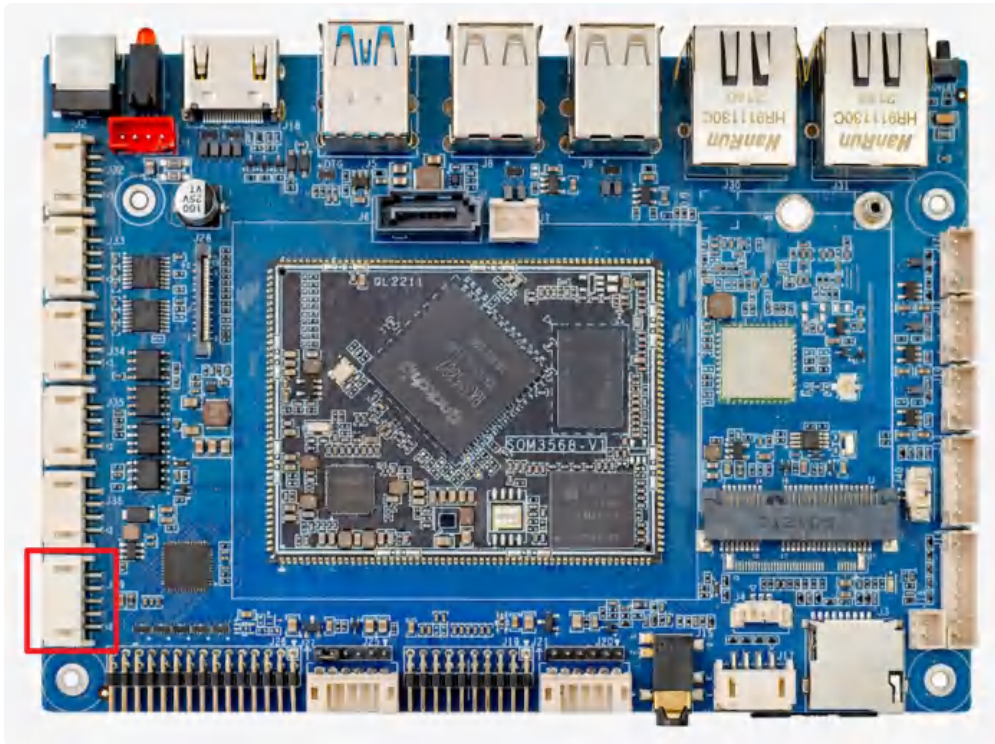
需要先安装microcom工具：

```
▼ Bash |
1  sudo apt-get update
2  sudo apt-get install microcom
```

```
1 root@ido:~# microcom -s 115200 -p /dev/ttyS0
2 [ 44.730195] of_dma_request_slave_channel: dma-names property of node '/s
   erial@fdd50000' missing or empty
3 connected to /dev/ttyS0
4 Escape character: Ctrl-\
5 Type the escape character to get to the prompt.
6 fjskdfjsdfjsdklfjdsfdfsdfdfdsfsdfd
```

按下键盘，将发送对应的字符；而接收的内容，会显示在终端。

### 3 CAN



IDO-EVB3568-V1共配置两路CAN接口，分别为CAN0和CAN1。支持 CAN2.0协议，CAN接口测试方法如下：

```
1  #关闭can0设备
2  ip link set can0 down
3
4  #设置通信波特率为250Kbps
5  ip link set can0 type can bitrate 250000
6
7  #打印can0信息
8  ip -details link show can0
9
10 #启动can0
11 ip link set can0 up
12
13 #执行candump, 阻塞等待can0接收
14 candump can0
15
16 #发送数据
17 cansend can0 123#DEADBEEF
```

## 4 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  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
```

### 4.1 连接WiFi热点

#### 4.1.1 命令行连接

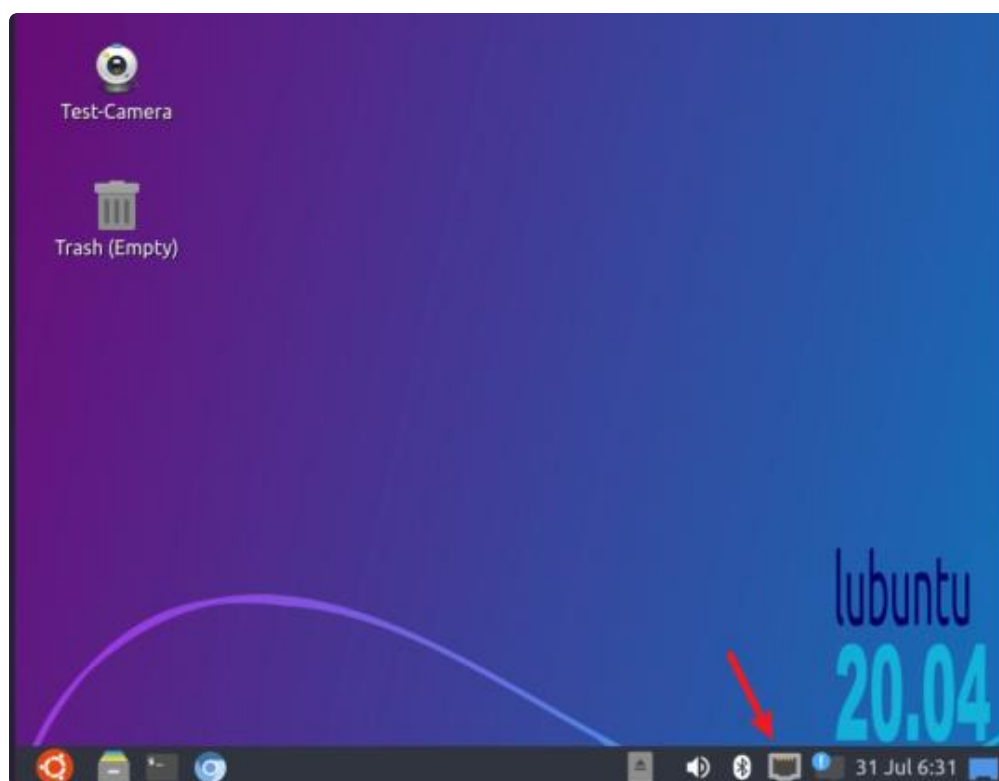
使用wpa\_cli工具连接，命令如下：

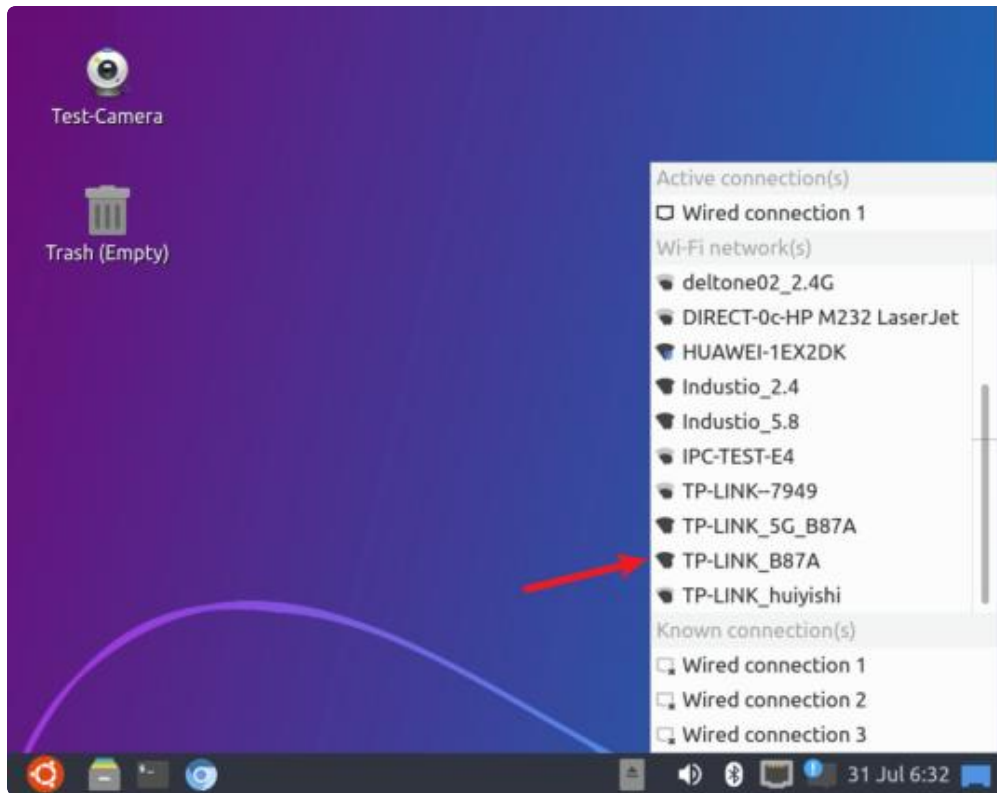
```
1 root@ido:~# wpa_cli -i wlan0 add_network
2 root@ido:~# wpa_cli -i wlan0 set_network 0 ssid '"TP-LINK_B87A"'
3 root@ido:~# wpa_cli -i wlan0 set_network 0 psk '"12345678"'
4 root@ido:~# wpa_cli -i wlan0 enable_network 0
5 root@ido:~# dhclient -i wlan0
```

其中"TP-LINK\_B87A"为热点名称，"12345678"为热点密码。

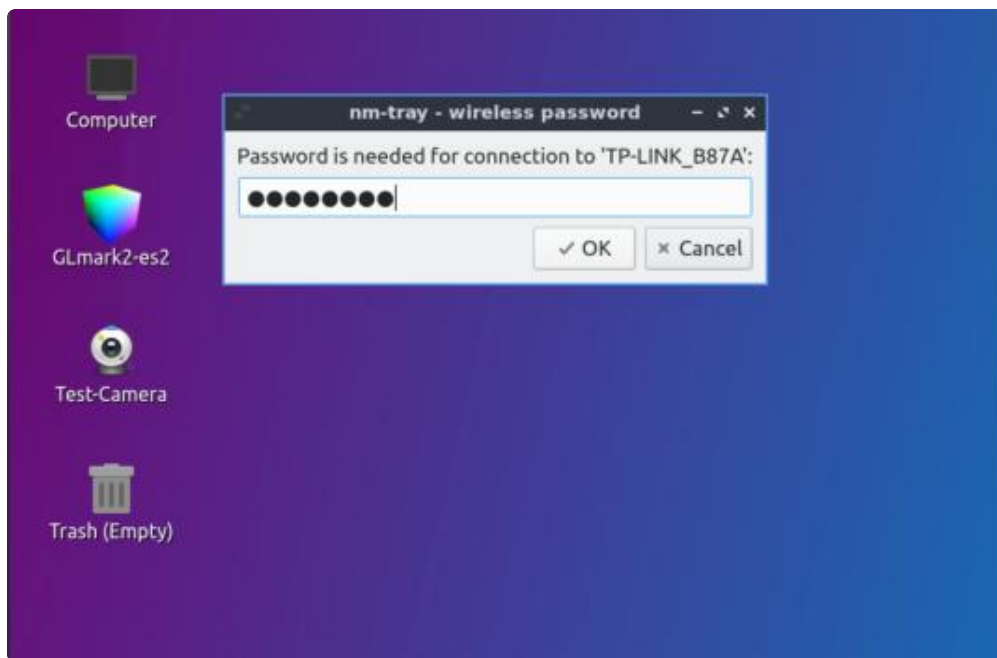
## 4.1.2 桌面连接

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：

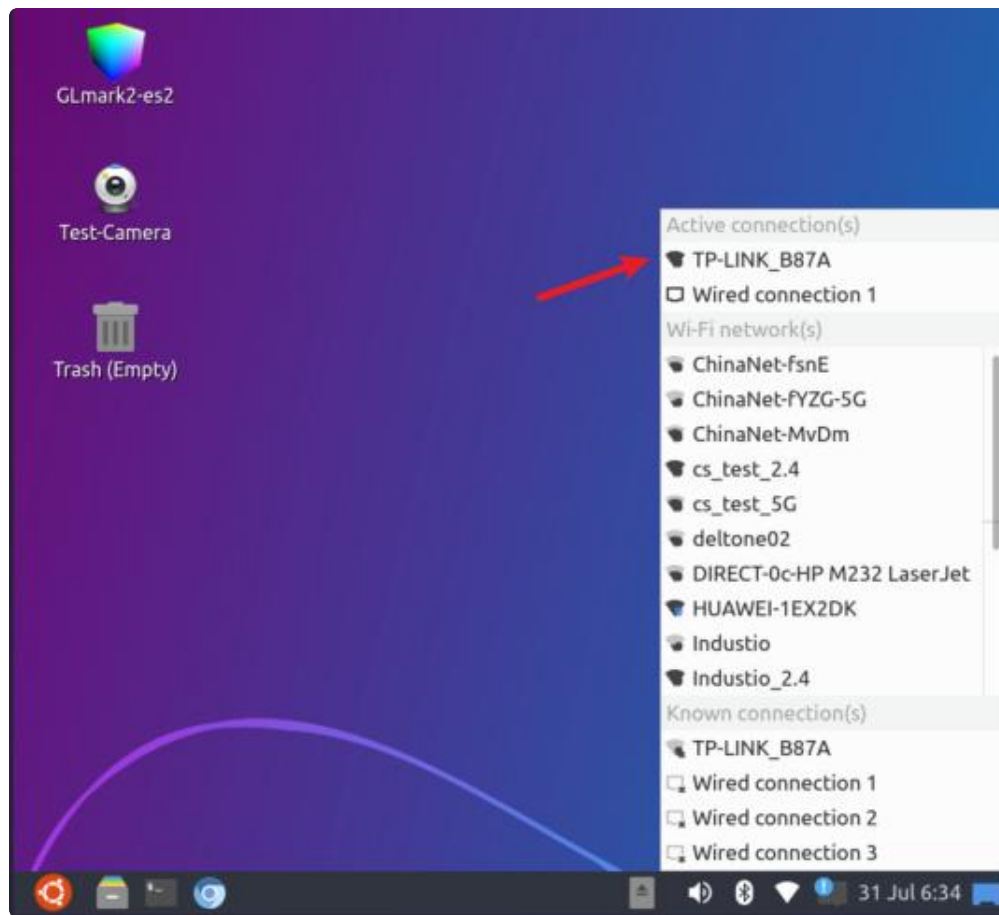




弹出密码输入框，使用键盘输入密码（如果没有接键盘，可以使用软键盘Onboard），如下图所示：



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
```

## 5 Bluetooth

```
▼ Bash |
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
```

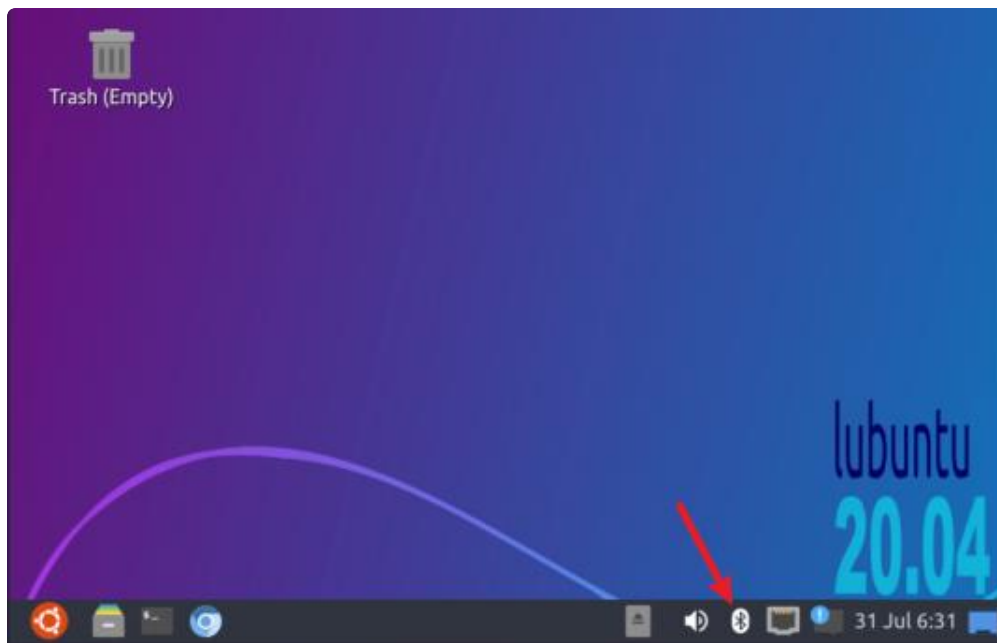
## 5.1 连接蓝牙设备

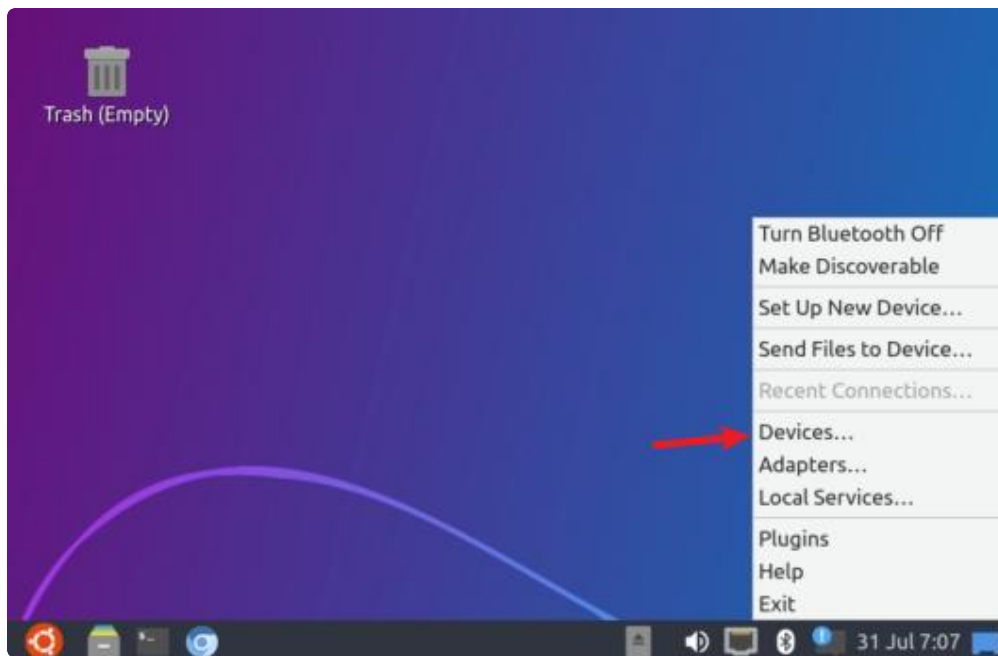
使用命令行连接蓝牙设备，命令如下：

```
▼ Shell |
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
```

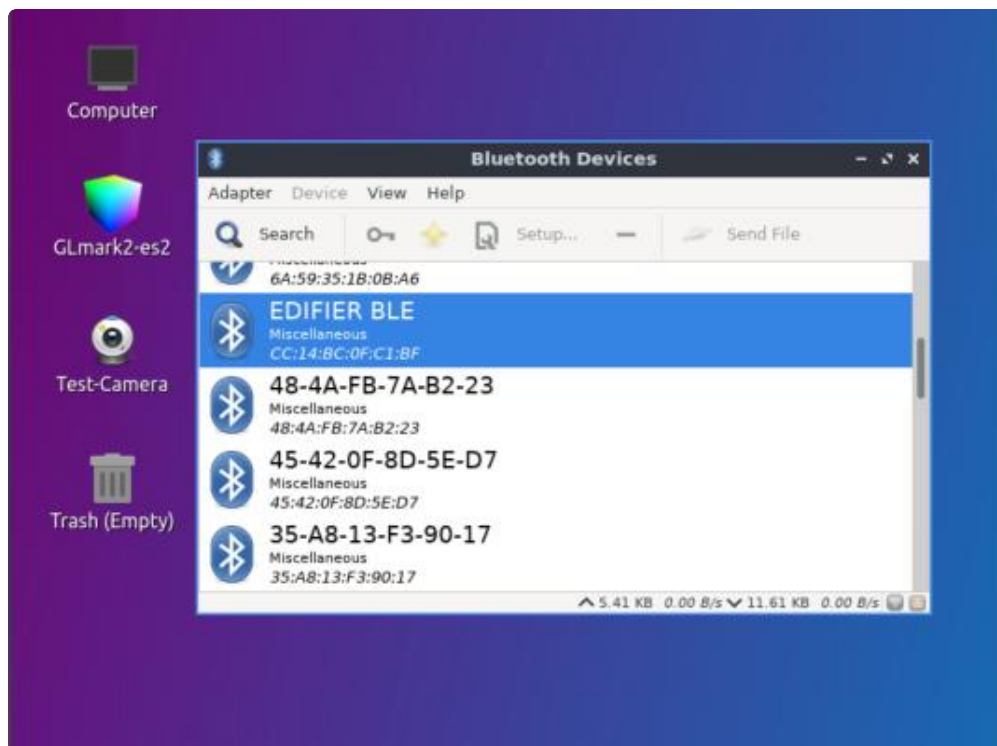
也可在桌面上连接蓝牙设备，步骤如下：

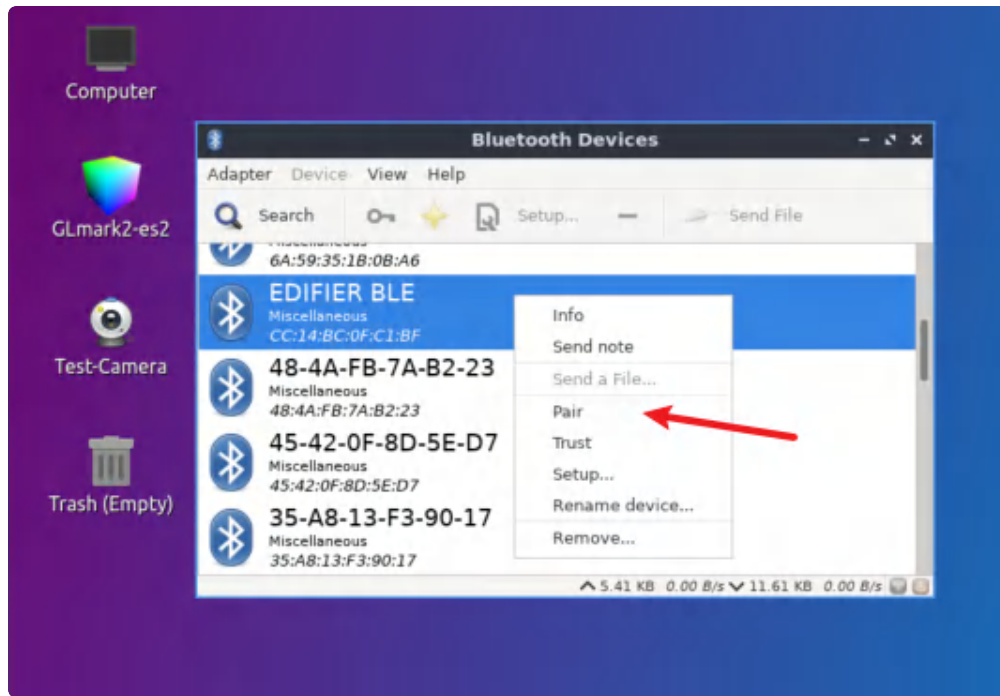
在桌面右下角蓝牙图标右键->Devices:





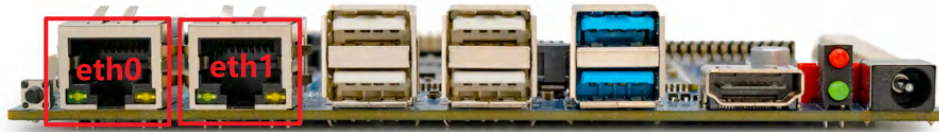
选中要连接的蓝牙设备，右键->pair:





## 6 以太网

开发板两路千兆以太网接口，分别为eth0和eth1。



两路以太网接口默认IP获取方式为 DHCP。

### 6.1 静态IP设置

以eth0设置静态IP地址为例，修改vi /etc/netplan/01-network-manager-all.yaml，在文件中添加如下内容

```
▼ Bash |
1 network:
2     version: 2
3     ethernet:
4         eth0: #配置的网卡的名称
5             addresses: [192.168.31.215/24] #配置的静态ip地址和掩码
6             dhcp4: no #关闭DHCP, 如果需要打开DHCP则写yes
7             dhcp6: no
8             gateway4: 192.168.31.1 #网关地址
9             nameservers:
10                addresses: [192.168.31.1,114.114.114.114] #DNS服务器
地址, 多个DNS服务器地址需要用英文逗号分隔开
11
```

**注意：**间距不是TAB而是空格。

```
▼ Bash |
1 netplan apply
```

执行以上命令生效。

## 7 4G

IDO-EVB3568-V1 默认适配EC20模块, 系统启动完成后, 执行以下指令拨号:

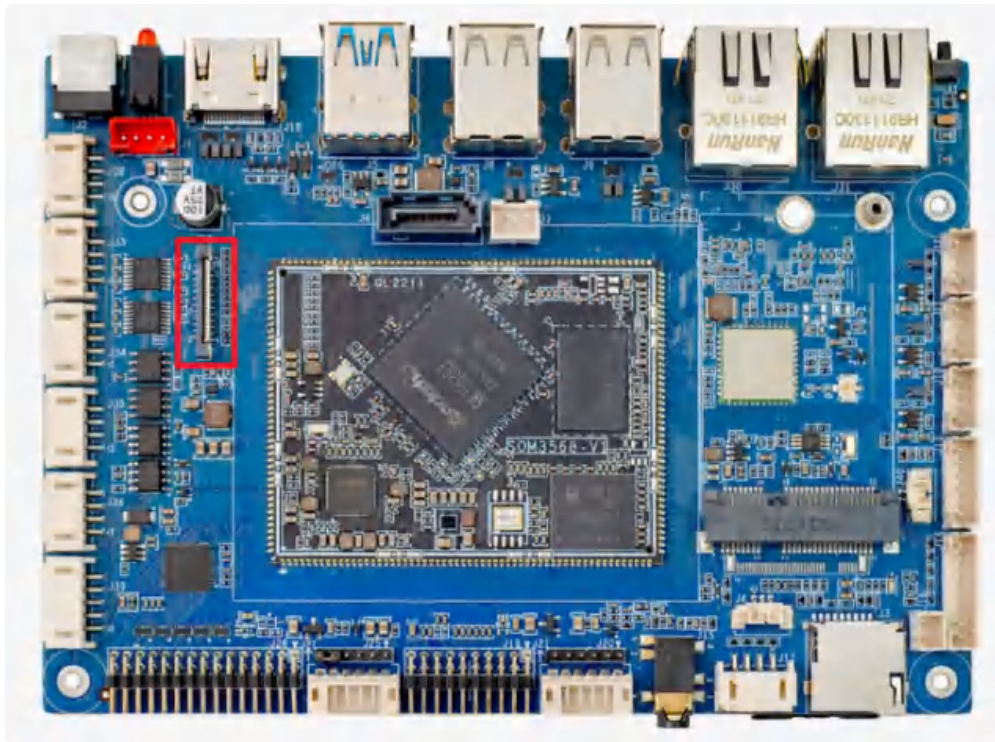
```
▼ Bash |
1 root@ido:~# killall quectel-CM
2 root@ido:~# quectel-CM &
```

正常拨号成功后, wwan0将会分配到ip地址, 此时可以测试是否能够正常ping通外网。

```
▼ Bash |
1 root@ido:~# ping www.baidu.com -I wwan0
```

## 8 摄像头

默认适配OV5648模块, 对应系统中的设备节点为/dev/video0。



## 8.1 测试

### 8.1.1 测试摄像头是否存在

```
▼ Shell |
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。

## 8.1.2 抓取视频流

使用v4l2-ctl工具可以抓取摄像头的视频数据流。

```

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

```

### 8.1.3 预览摄像头

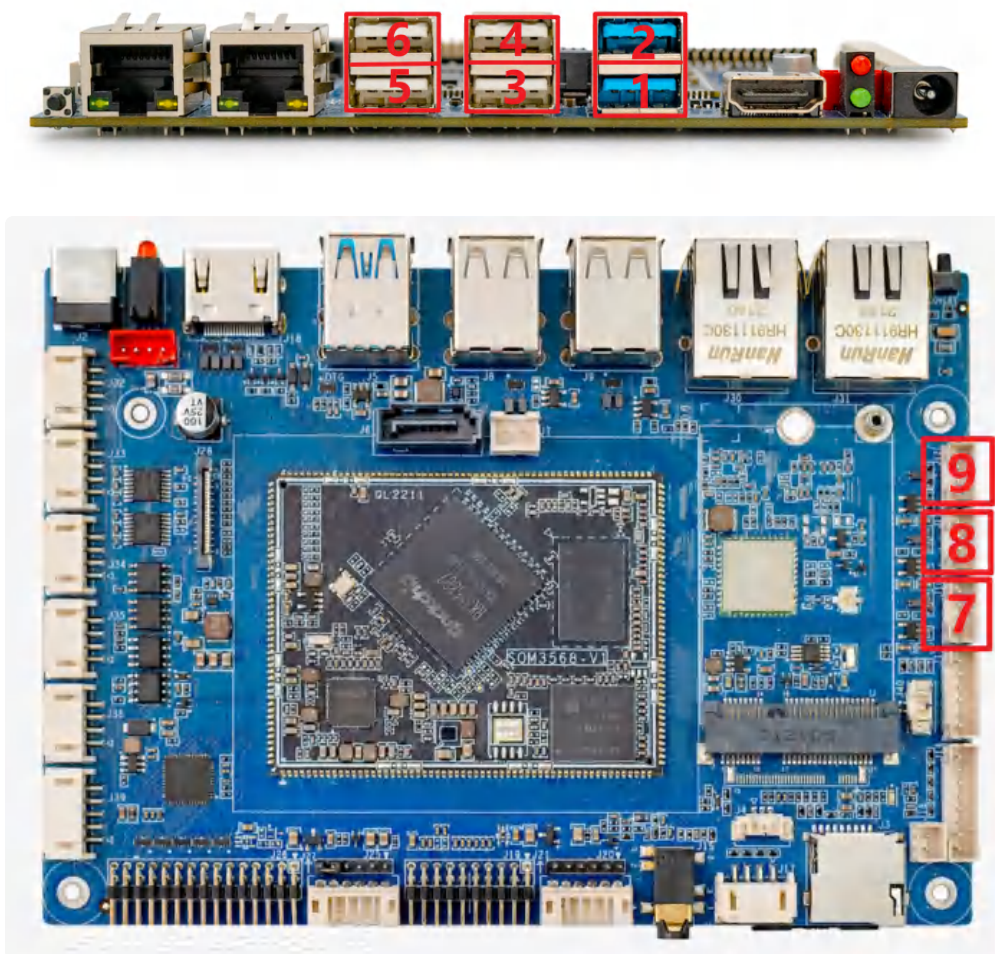
```

▼ Bash |
1  gst-launch-1.0 v4l2src device=/dev/video0 ! video/x-raw, format=NV12, width
   =1920, height=1080, framerate=30/1 ! autovideoconvert ! autovideosink

```

## 9 U盘

主板共配置9路USB，其中6路TYPE-A接口，3路PH接口。9路USB分别标记1-9，其中1为USB OTG，其余为USB HOST。



## 9.1 电源控制

USB2-8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/usb_host1_pwr/brightness |
| 3  | USB 2.0 HOST | /sys/class/leds/usb_host3_pwr/brightness |
| 4  | USB 2.0 HOST | /sys/class/leds/usb_fe_d1_pwr/brightness |
| 5  | USB 2.0 HOST | /sys/class/leds/usb_fe_d2_pwr/brightness |
| 6  | USB 2.0 HOST | /sys/class/leds/usb_fe_d3_pwr/brightness |
| 7  | USB 2.0 HOST | /sys/class/leds/usb_fe_d4_pwr/brightness |
| 8  | USB 2.0 HOST | /sys/class/leds/usb_fe_d5_pwr/brightness |

打开/关闭 USB电源方法（以USB2为例）：

```
1 # 打开电源
2 echo 255 > /sys/class/leds/usb_host1_pwr/brightness
3
4 # 关闭电源
5 echo 0 > /sys/class/leds/usb_host1_pwr/brightness
```

## 9.2 U盘的挂载

当接入U盘设备时，默认挂载到/media/ido/目录下

```
1 [root@RK356X:/]# mount
2 ...
3 /dev/sda1 on /media/USB0 type vfat (rw,nodev,noexec,noatime,nodiratime,fmask=0022,dmask=0022,codepage=936,iocharset=utf8,shortname=mixed,errors=remount-ro)
4 ...
```

## 9.3 OTG模式切换

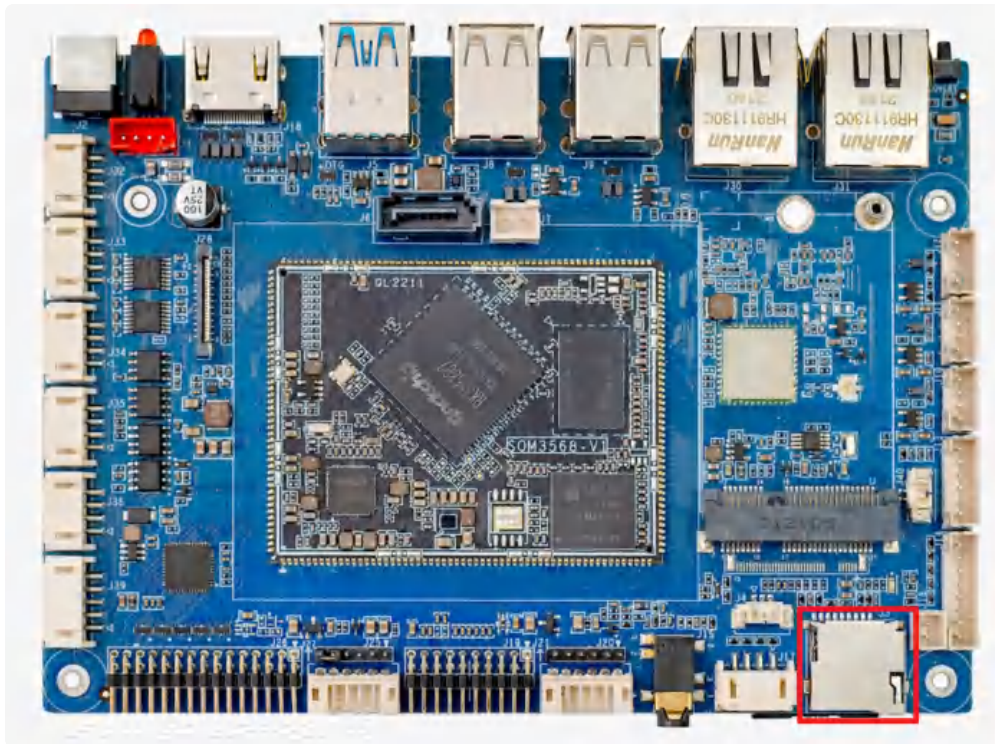
USB1为OTG接口，默认开机为Devices模式，可用于ADB调试；切换至HOST模式时，可接U盘等设备。

OTG模式切换方法如下：

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

## 10 SD卡

将SD卡插入到SD卡槽中，将自动挂载到/mnt/sdcard/目录下。



Bash |

```
1 [root@RK356X:/]# mount
2 ...
3 /dev/mmcblk1p1 on /mnt/sdcard type vfat (rw,noatime,uid=1000,gid=1000,fmask
  =0133,dmask=0022,codepage=936,iocharset=utf8,shortname=mixed,errors=remount
  -ro)
4 ...
```

## 11 按键

主板配置了一个Recovery按键，如下图所示：



在设备断电的情况下，该按键用于烧录固件。在系统正常启动后，则可作为普通按键使用。对应的设备节点为/dev/input/event6，键值为KEY\_VOLUMEUP(115)。

使用evtest进行测试，没有evtest指令则使用以下命令安装：

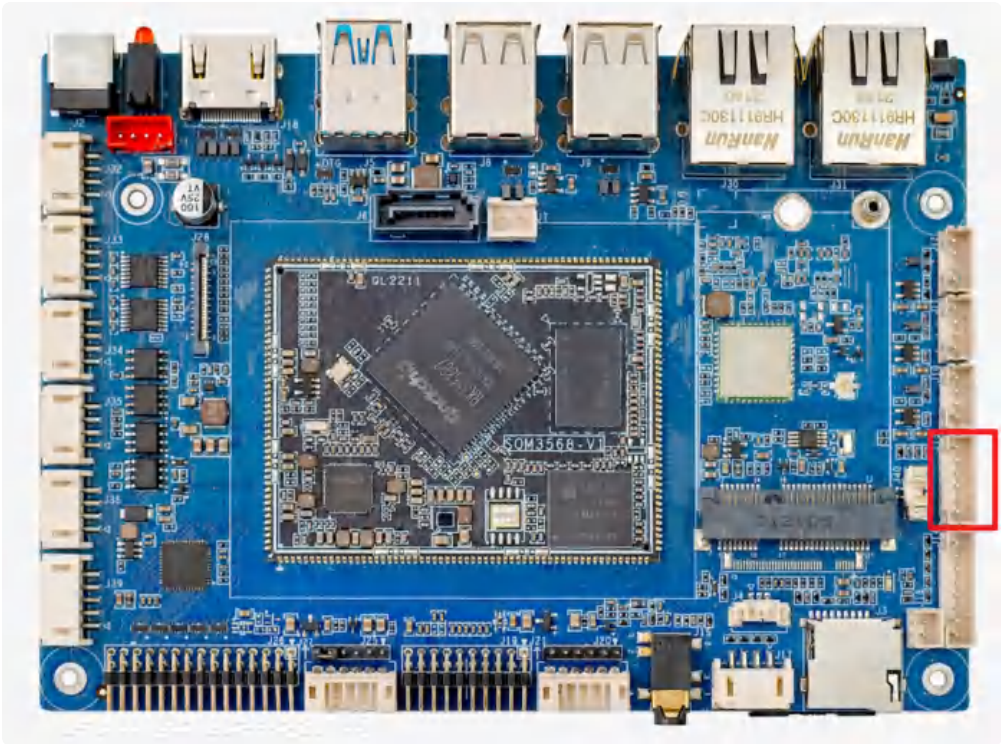
```
1 apt-get install evtest
```

```
1 root@ido:~# evtest
2 No device specified, trying to scan all of /dev/input/event*
3 Available devices:
4 /dev/input/event0:      hdmi_cec_key
5 /dev/input/event1:      fe6e0030.pwm
6 /dev/input/event2:      rockchip,hdmi rockchip,hdmi
7 /dev/input/event3:      bt-powerkey
8 /dev/input/event4:      rk805 pwrkey
9 /dev/input/event5:      rockchip,rk809-codec Headphones
10 /dev/input/event6:      adc-keys
11 Select the device event number [0-6]: 6
12 Input driver version is 1.0.1
13 Input device ID: bus 0x19 vendor 0x1 product 0x1 version 0x100
14 Input device name: "adc-keys"
15 Supported events:
16   Event type 0 (EV_SYN)
17   Event type 1 (EV_KEY)
18     Event code 114 (KEY_VOLUMEDOWN)
19     Event code 115 (KEY_VOLUMEUP)
20     Event code 139 (KEY_MENU)
21     Event code 158 (KEY_BACK)
22 Properties:
23 Testing ... (interrupt to exit)
24 Event: time 1731566093.194520, type 1 (EV_KEY), code 115 (KEY_VOLUMEUP), v
   alue 1
25 Event: time 1731566093.194520, ----- SYN_REPORT -----
26 Event: time 1731566093.297450, type 1 (EV_KEY), code 115 (KEY_VOLUMEUP), v
   alue 0
27 Event: time 1731566093.297450, ----- SYN_REPORT -----
28 Event: time 1731566093.814204, type 1 (EV_KEY), code 115 (KEY_VOLUMEUP), v
   alue 1
29 Event: time 1731566093.814204, ----- SYN_REPORT -----
30 Event: time 1731566093.917579, type 1 (EV_KEY), code 115 (KEY_VOLUMEUP), v
   alue 0
31 Event: time 1731566093.917579, ----- SYN_REPORT -----
```

在选择event number为6后（对应adc-keys），按下Recovery按键，即可看到按下和松开打印的信息。

# 12 ADC

主板共配置了3路ADC 接口（精度为10位），位置如下图所示：



设备节点对应关系如下表：

| 接口   | 设备节点                                              |
|------|---------------------------------------------------|
| ADC2 | /sys/bus/iio/devices/iio\:device0/in_voltage2_raw |
| ADC4 | /sys/bus/iio/devices/iio\:device0/in_voltage4_raw |
| ADC5 | /sys/bus/iio/devices/iio\:device0/in_voltage5_raw |

## 12.1 ADC值读取

▼

Bash

```
1 cat /sys/bus/iio/devices/iio\:device0/in_voltage2_raw
```

## 12.2 ADC电压转换关系



Bash

```
1 V=(in_voltage2_raw/1024)*1.8v
```

假设in\_voltage2\_raw的值为500，则对应的ADC电压为 $V=(500/1024)*1.8v=0.879v$

## 13 时间设置

### 13.1 RTC时间读取和同步

系统时间读取和设置



Bash

```
1 # date
2 Fri Mar 18 12:00:22 CST 2022
3 # date -s "2022-03-18 12:01:00"
```

RTC时间设置



Bash

```
1 # hwclock -r
2 2022-03-18 12:01:06.991425+08:00
3 # hwclock -w
```

### 13.2 RTC定时开机

主板支持RTC定时开机，设置指令如下：



Shell

```
1 echo +0 > /sys/class/rtc/rtc1/wakealarm
2 echo +300 > /sys/class/rtc/rtc1/wakealarm
3
4 poweroff
```

以上指令设置关机300秒后自动开机。

### 13.3 NTP时间同步

系统默认开启了NTP服务，连接网络后，将自动同步网络时间。

## 13.4 时区

查看时区

```
▼ Bash |
1 [root@RK356X:/]# date -R
2 Wed, 26 Oct 2022 03:26:46 +0000
```

+0000表示在0时区。

设置时区

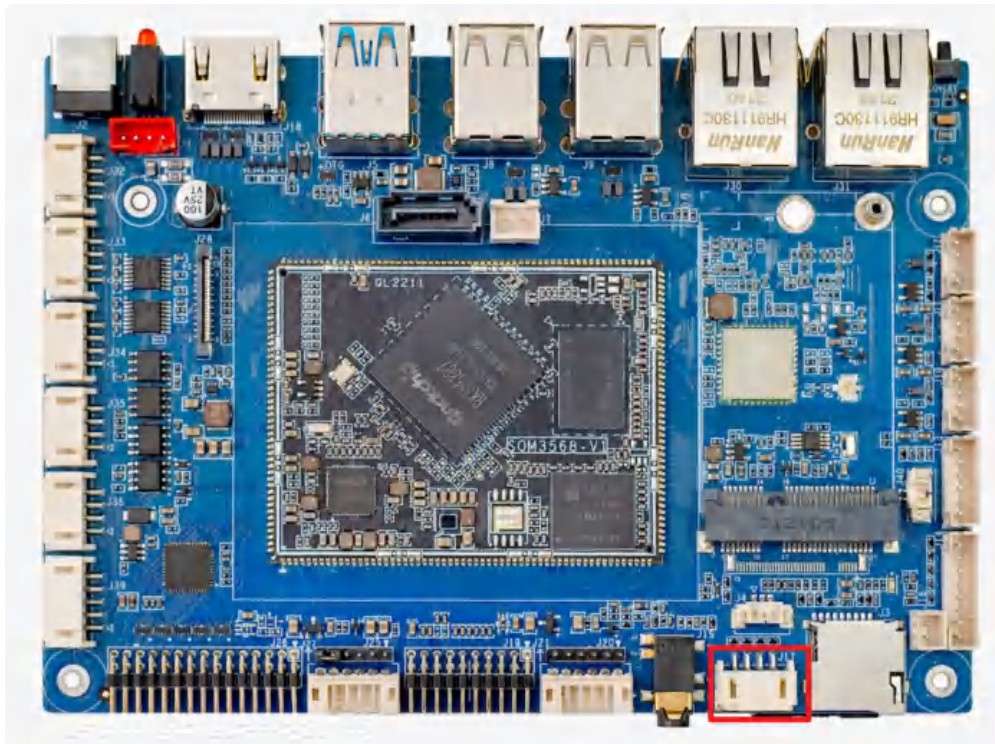
```
▼ Bash |
1 [root@RK356X:/]# export TZ='Asia/Shanghai'
2 [root@RK356X:/]#
3 [root@RK356X:/]# date -R
4 Wed, 26 Oct 2022 11:30:02 +0800
5 [root@RK356X:/]#
```

## 14 音频

使用[aplay](#)工具查看声卡设备

```
1 root@ido:~# 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: rockchiprk809co [rockchip,rk809-codec], device 0: fe410000.i2s-rk8
    17-hifi rk817-hifi-0 [fe410000.i2s-rk817-hifi rk817-hifi-0]
34    Subdevices: 1/1
35    Subdevice #0: subdevice #0
36 ▾ card 3: rockchiphdmi [rockchip,hdmi], device 0: rockchip,hdmi i2s-hifi-0
    [rockchip,hdmi i2s-hifi-0]
37    Subdevices: 1/1
38    Subdevice #0: subdevice #0
```

## 14.1 Lineout



不插入耳机，使用aplay播放wav音频测试

Bash

```
1 $ aplay -Dhw:2,0 ./8k16bpsStereo.wav
2 Playing WAVE '/etc/bsa_file/8k16bpsStereo.wav' : Signed 16 bit Little Endian, Rate 8000 Hz, Stereo
```

 8k16bpsStereo.wav

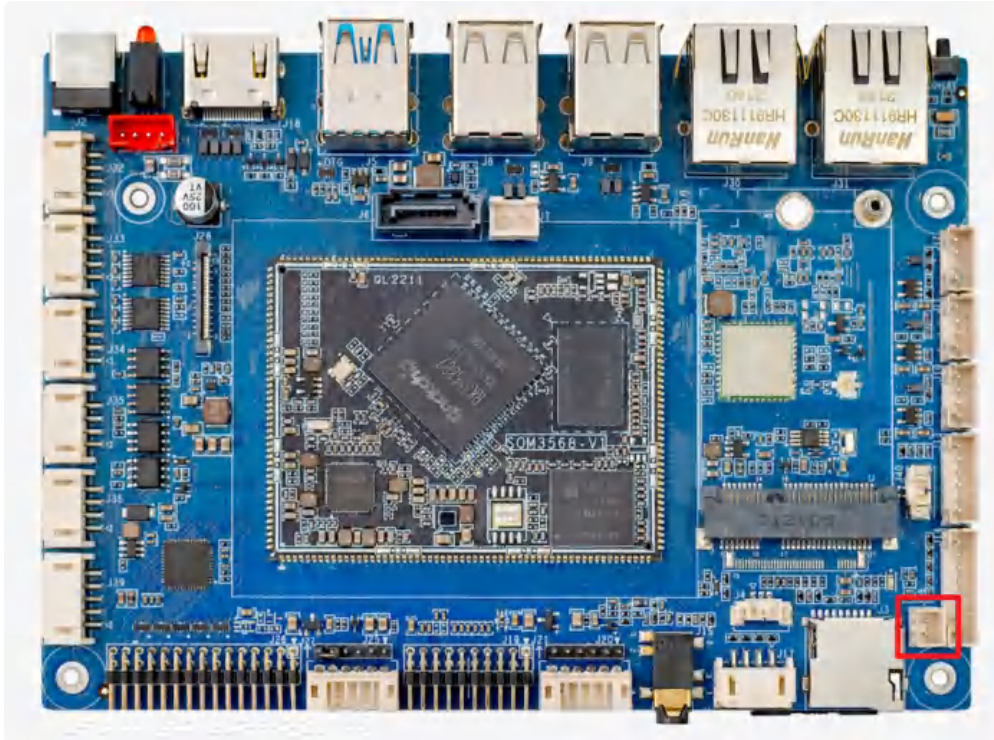
## 14.2 耳机



插入耳机，使用aplay播放wav音频测试

```
1 [root@RK356X:/]# aplay -Dhw:2,0 ./8k16bpsStereo.wav
2 Playing WAVE '/etc/bsa_file/8k16bpsStereo.wav' : Signed 16 bit Little Endian, Rate 8000 Hz, Stereo
```

## 14.3 录音



查看录音设备：

```
1 root@ido:~# arecord -l
2 **** List of CAPTURE 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: rockchiprk809co [rockchip,rk809-codec], device 0: fe410000.i2s-rk8
    17-hifi rk817-hifi-0 [fe410000.i2s-rk817-hifi rk817-hifi-0]
34    Subdevices: 1/1
35    Subdevice #0: subdevice #0
36
```

使用arecord工具进行录音测试。

```
1 $ arecord -D hw:2,0 -r 48000 -c 2 -f S16_LE test.wav
2 Recording WAVE 'test.wav' : Signed 16 bit Little Endian, Rate 48000 Hz, Stereo
3 ^CAborted by signal Interrupt...
```

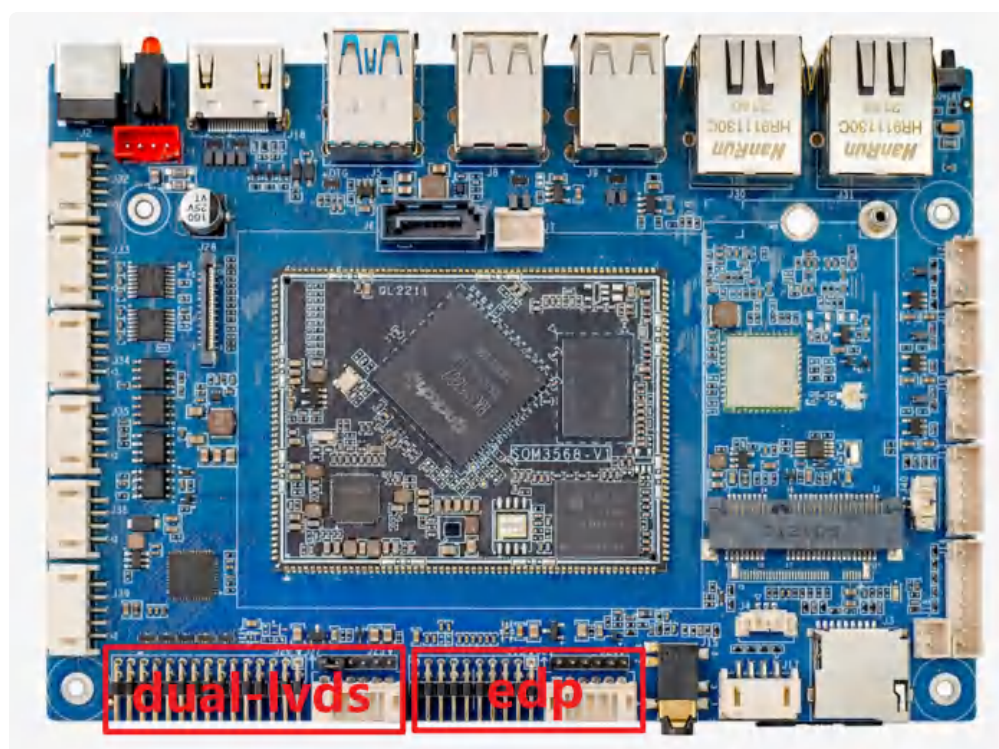
播放录音：

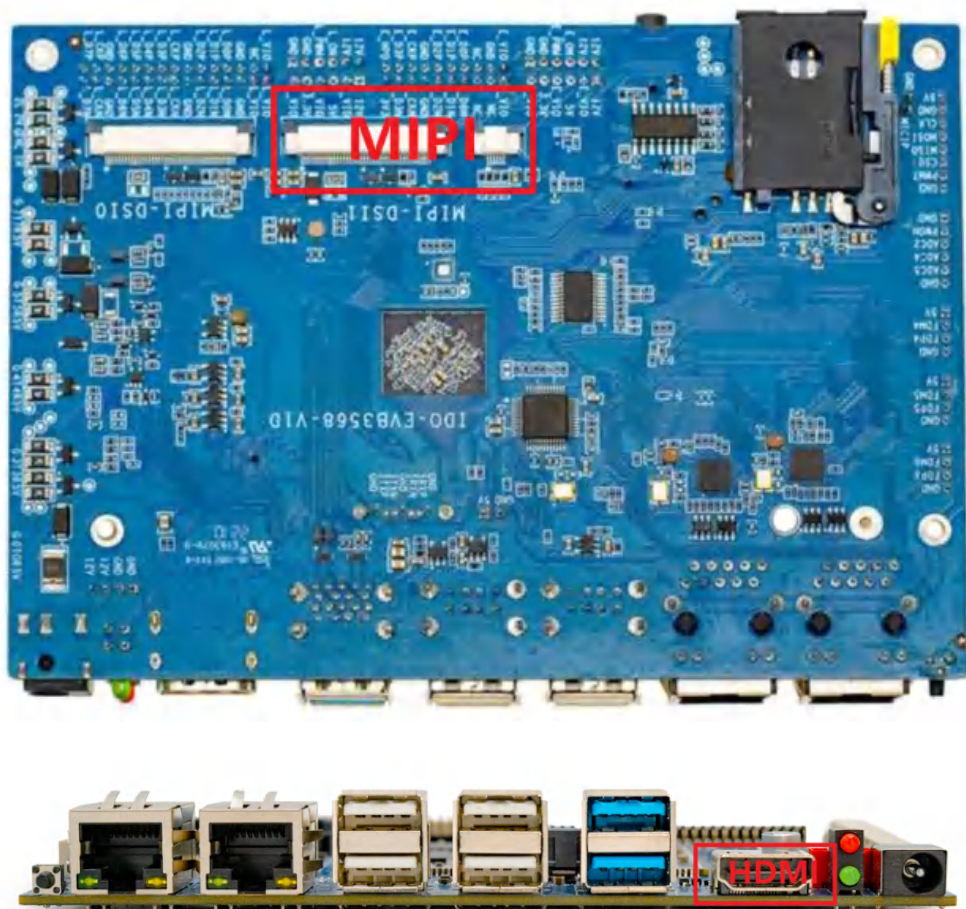
```
1 $ aplay -Dhw:2,0 test.wav
```

## 15 显示屏

### 15.1 显示屏接口说明

主板各个显示接口如下图所示：





## 15.2 显示设置

### 15.2.1 屏幕背光亮度设置

- eDP/MIPI屏背光控制

设备节点: `/sys/class/backlight/backlight/brightness`

设置方法: (支持调节范围 0-255)



Bash

```
1 #关闭
2 echo 0 > /sys/class/backlight/backlight/brightness
3 #最亮
4 echo 255 > /sys/class/backlight/backlight/brightness
```

- Dual LVDS屏幕背光控制

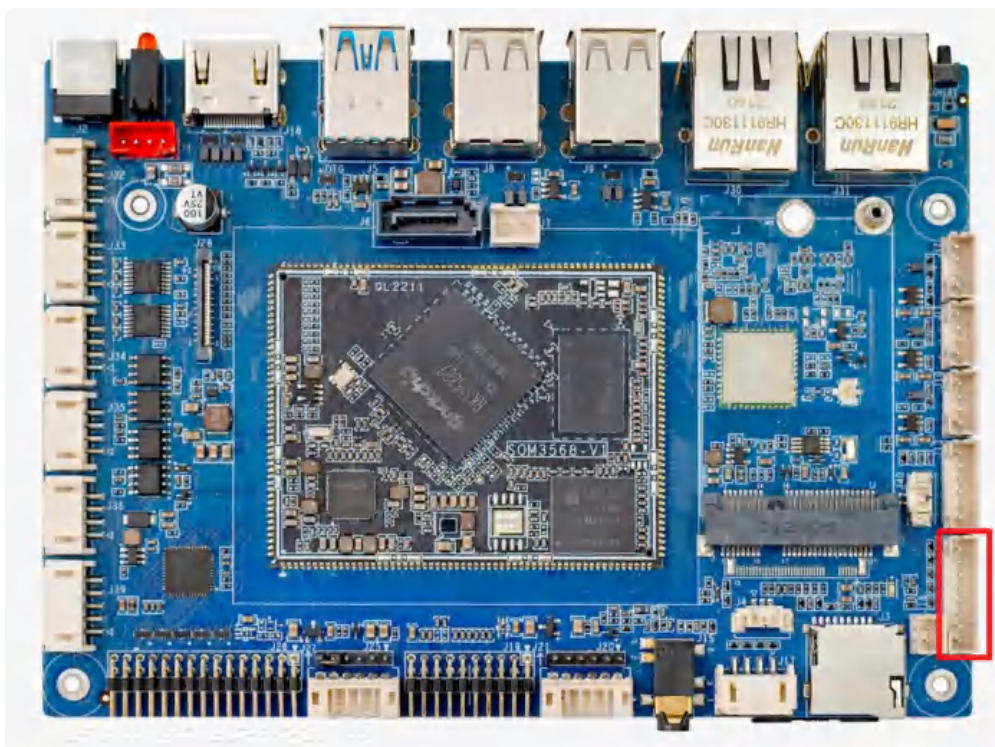
设备节点: `/sys/class/backlight/backlight1/brightness`

设置方法: (支持调节范围 0-255)

```
1 #关闭
2 echo 0 > /sys/class/backlight/backlight1/brightness
3 #最亮
4 echo 255 > /sys/class/backlight/backlight1/brightness
```

## 16 扩展GPIO

主板扩展了5路GPIO，位于J14：



各个引脚的功能描述如下表所示：

| 序号 | 定义                                | 电平/V | 说明                  |
|----|-----------------------------------|------|---------------------|
| 1  | 5V                                | 5V   | 电源5V输出              |
| 2  | GND                               | GND  | 电源地                 |
| 3  | SPI3_CLK_M1/GPIO4_C2/PWM14_M1     | 3.3V | 默认配置为GPIO，GPIO编号146 |
| 4  | SPI3_MOSI_M1/GPIO4_C3/PWM15_IR_M1 | 3.3V | 默认配置为GPIO，GPIO编号147 |

|   |                                |      |                      |
|---|--------------------------------|------|----------------------|
| 5 | SPI3_MISO_M1/GPIO4_C5/PWM12_M1 | 3.3V | 默认配置为GPIO, GPIO编号149 |
| 6 | SPI3_CS0_M1/GPIO4_C6/PWM13_M1  | 3.3V | 默认配置为GPIO, GPIO编号150 |
| 7 | PWM7_IR/GPIO0_C6               | 3.3V | 默认配置为GPIO, GPIO编号22  |
| 8 | GND                            | GND  | 电源地                  |

## 17.1 测试方法

以GPIO4\_C2为例。

作为输入：

```

1  echo 146 > /sys/class/gpio/export
2  echo in > /sys/class/gpio/gpio146/direction
3  cat /sys/class/gpio/gpio146/value

```

value值为0, 说明该gpio输入低电平; value值为1, 说明该gpio输入高电平。

作为输出：

```

1  echo 146 > /sys/class/gpio/export
2  echo out > /sys/class/gpio/gpio146/direction
3  #输出高电平
4  echo 1 > /sys/class/gpio/gpio146/value
5  #输出低电平
6  echo 0 > /sys/class/gpio/gpio146/value

```

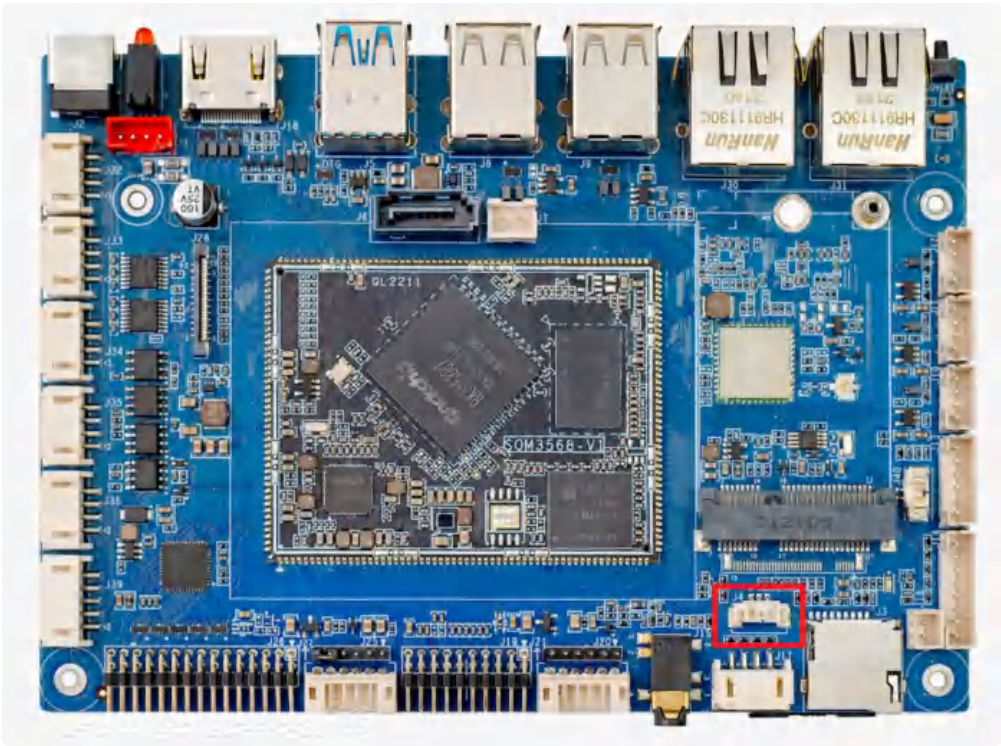
## IDO-EVB3568-V1 Buildroot 系统

### 1 调试

IDO-EVB3568-V1开发板支持串口调试、ADB调试和远程SSH调试。

# 1.1 串口调试

串口调试接口位于J4端口，见下图。请使用配套的USB串口调试工具。



调试串口为TTL电平，通信参数为1500000 8 N 1。

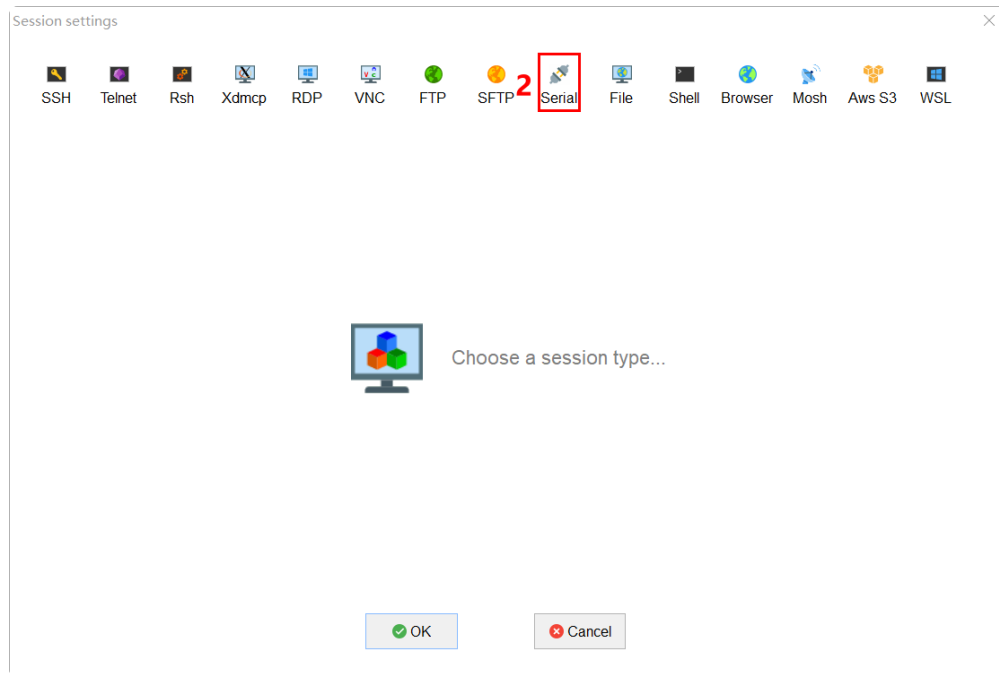
1. 打开MobaXterm，下载链接如下：

链接：[https://pan.baidu.com/s/11ui4LTd2mq\\_9kiJpeL4bWg?pwd=1234](https://pan.baidu.com/s/11ui4LTd2mq_9kiJpeL4bWg?pwd=1234)

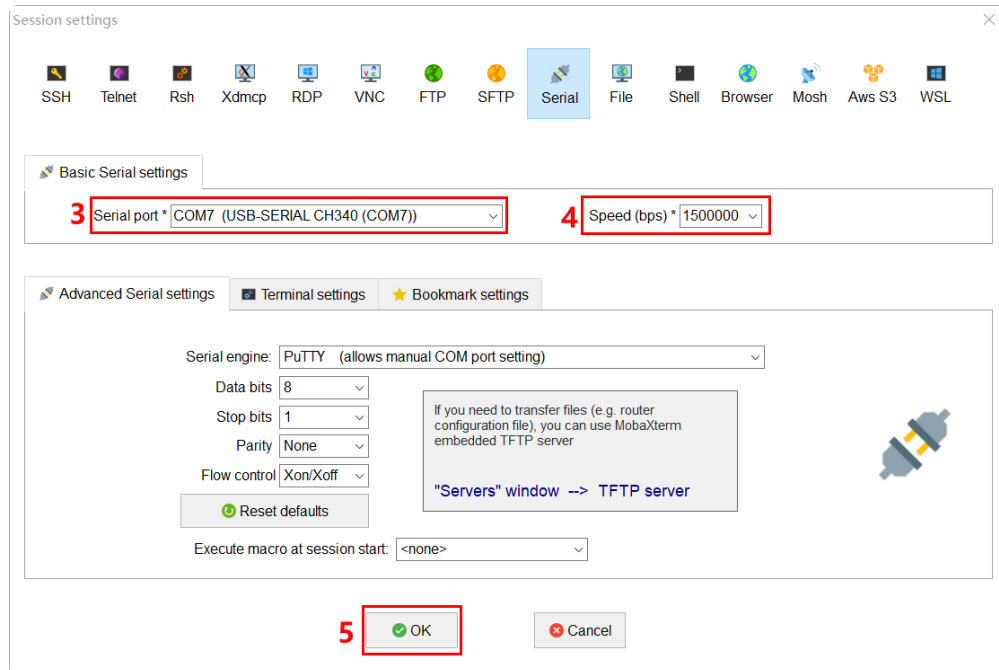
提取码：1234

| <input type="checkbox"/> 文件名                          | ↓ 修改时间           | 类型    | 大小      |
|-------------------------------------------------------|------------------|-------|---------|
| <input type="checkbox"/> RKDevTool_Release_v2.95.zip  | 2024-04-24 11:53 | zip文件 | 2.30MB  |
| <input type="checkbox"/> other_tools.txt              | 2024-04-25 15:31 | txt文件 | 44B     |
| <input type="checkbox"/> MobaXterm_Portable_v23.6.zip | 2024-04-24 14:30 | zip文件 | 39.99MB |
| <input type="checkbox"/> DriverAssitant_v5.11.zip     | 2024-04-24 11:52 | zip文件 | 9.36MB  |

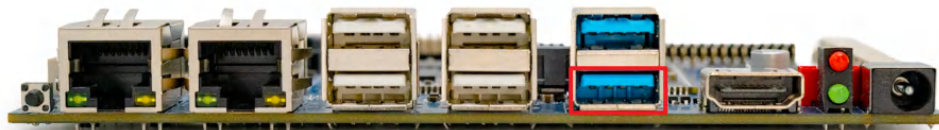
2. 选择sessionSerial，如下图所示：



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



## 1.2 ADB调试



上图红色框内的USB接口为支持OTG模式切换，使用双公头 USB 数据线连接开发板和 PC 端的 USB接口，在PC终端识别到 ADB 设备，即可使用 adb shell 调试。

```
C:\Users\ronnie>adb shell
* daemon not running. starting it now on port 5037 *
* daemon started successfully *
[root@RK356X:/]# ls
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:/]#
```

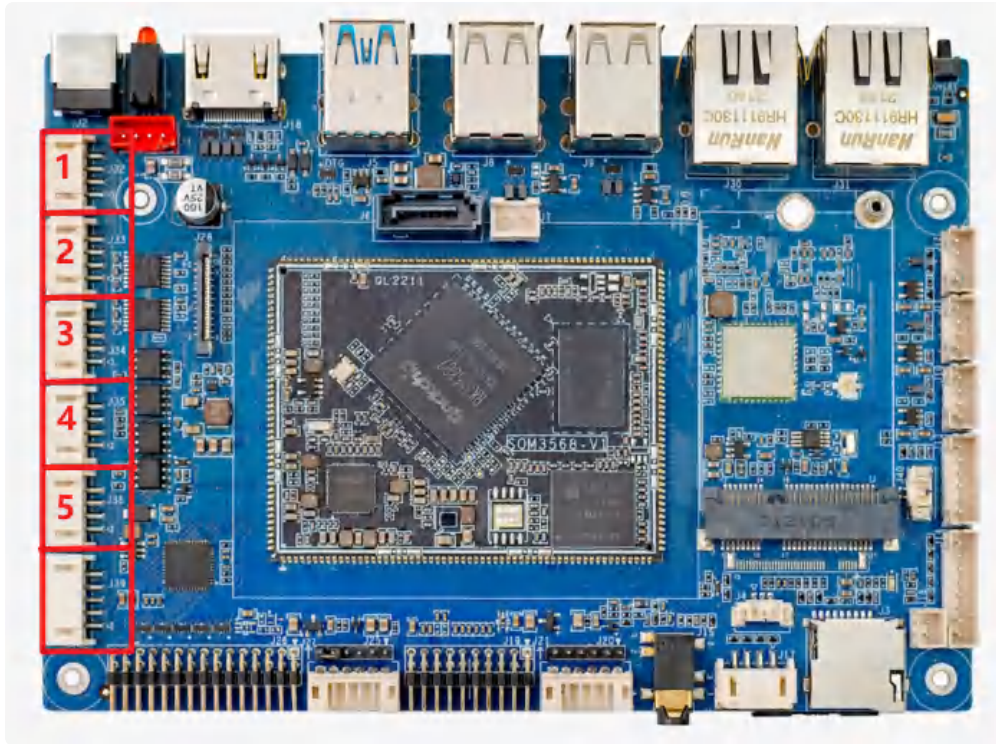
## 1.3 SSH调试

SSH登录账号信息：

账号：root

密码：rockchip

## 2 串口



串口接口位置及引脚定义如上图所示，设备节点如下表所示：

| 序号 | 功能    | 设备节点       |
|----|-------|------------|
| 1  | TTL   | /dev/ttyS0 |
| 2  | RS232 | /dev/ttyS3 |
| 3  | RS232 | /dev/ttyS4 |
| 4  | RS485 | /dev/ttyS5 |
| 5  | RS485 | /dev/ttyS7 |

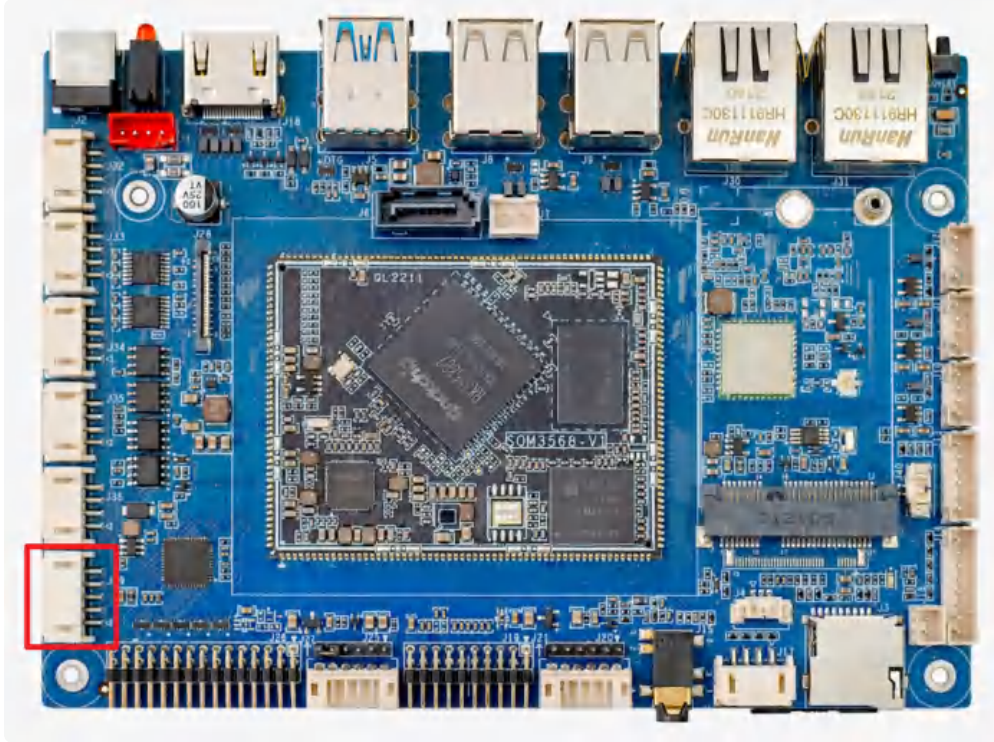
以上串口均可以使用microcom工具进行测试

Bash

```
1 $ microcom -s 115200 /dev/ttyS0
2 [ 2683.415483] of_dma_request_slave_channel: dma-names property of node '/serial@fdd50000' missing or empty
3 [ 2683.415544] ttyS0 - failed to request DMA, use interrupt mode
4
```

当按下键盘时，串口会发送对应的字符，而接收的内容会显示在终端。Ctrl+\键,然后输入 quit 停止测试。

### 3 CAN



IDO-EVB3568-V1共配置两路CAN接口，分别为CAN0和CAN1。支持 CAN2.0协议，CAN接口测试方法如下：

```
1  #关闭can0设备
2  ip link set can0 down
3
4  #设置通信波特率为250Kbps
5  ip link set can0 type can bitrate 250000
6
7  #打印can0信息
8  ip -details link show can0
9
10 #启动can0
11 ip link set can0 up
12
13 #执行candump，阻塞等待can0接收
14 candump can0
15
16 #发送数据
17 cansend can0 123#DEADBEEF
```

Bash |

## 4 WiFi

在使用 WiFi时连接好WiFi天线，设备节点为wlan0

```
1  $ ifconfig wlan0
2  wlan0      Link encap:Ethernet  HWaddr 2C:3B:70:14:17:95
3              inet addr:169.254.41.145  Bcast:169.254.255.255  Mask:255.255.0.0
4              inet6 addr: fe80::b05:fca4:fb45:9468/64 Scope:Link
5              UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
6              RX packets:0 errors:0 dropped:0 overruns:0 frame:0
7              TX packets:75 errors:0 dropped:0 overruns:0 carrier:0
8              collisions:0 txqueuelen:1000
9              RX bytes:0 (0.0 B)  TX bytes:21920 (21.4 KiB)
```

### 4.1 连接WiFi热点

修改/etc/wpa\_supplicant.conf，填写正确的热点账号和密码：

```
1  ctrl_interface=/var/run/wpa_supplicant
2  ap_scan=1
3  update_config=1
4
5  network={
6      ssid="TP-LINK_B87A"
7      psk="12345678"
8      key_mgmt=WPA-PSK
9  }
```

使用wpa\_supplicant工具连接WiFi热点：

```
1  wpa_supplicant -i wlan0 -c /etc/wpa_supplicant.conf -B
```

连接成功后，wlan0节点会自动分配IP：

```

1  $ ifconfig wlan0
2  wlan0      Link encap:Ethernet  HWaddr 2C:3B:70:14:17:95
3              inet addr:192.168.1.101  Bcast:192.168.1.255  Mask:255.255.255.0
4              inet6 addr: fe80::220a:b25:4bd:2e3a/64 Scope:Link
5              UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
6              RX packets:26 errors:0 dropped:0 overruns:0 frame:0
7              TX packets:40 errors:0 dropped:0 overruns:0 carrier:0
8              collisions:0 txqueuelen:1000
9              RX bytes:5075 (4.9 KiB)  TX bytes:3913 (3.8 KiB)

```

## 5 Bluetooth

系统开机默认会打开蓝牙，对应的节点为hci0：

```

1  $ hciconfig
2  hci0:      Type: Primary  Bus: UART
3              BD Address: F3:7A:FA:A4:5E:22  ACL MTU: 1021:8  SCO MTU: 64:1
4              DOWN
5              RX bytes:668 acl:0 sco:0 events:34 errors:0
6              TX bytes:423 acl:0 sco:0 commands:34 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

```

使用hcitool测试蓝牙扫描功能

```

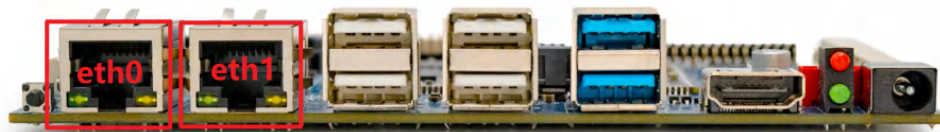
1  $ hciconfig hci0 up
2  $ hcitool -i hci0 scan
3  Scanning ...
4          94:87:E0:9D:14:12      seeyou
5          4C:4F:EE:12:6C:A3      OnePlus 8 Pro
6          5C:C5:63:02:31:19      客厅的小米电视

```

使用bluetoothctl工具进行蓝牙配对连接。

## 6 以太网

开发板两路千兆以太网接口，分别为eth0和eth1。



两路以太网接口默认IP获取方式为 DHCP。

### 6.1 静态IP设置

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

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

其中，dns-nameservers一项为默认dns。

## 7 4G

IDO-EVB3568-V1 默认适配EC20模块，系统启动完成后，执行以下命令拨号：

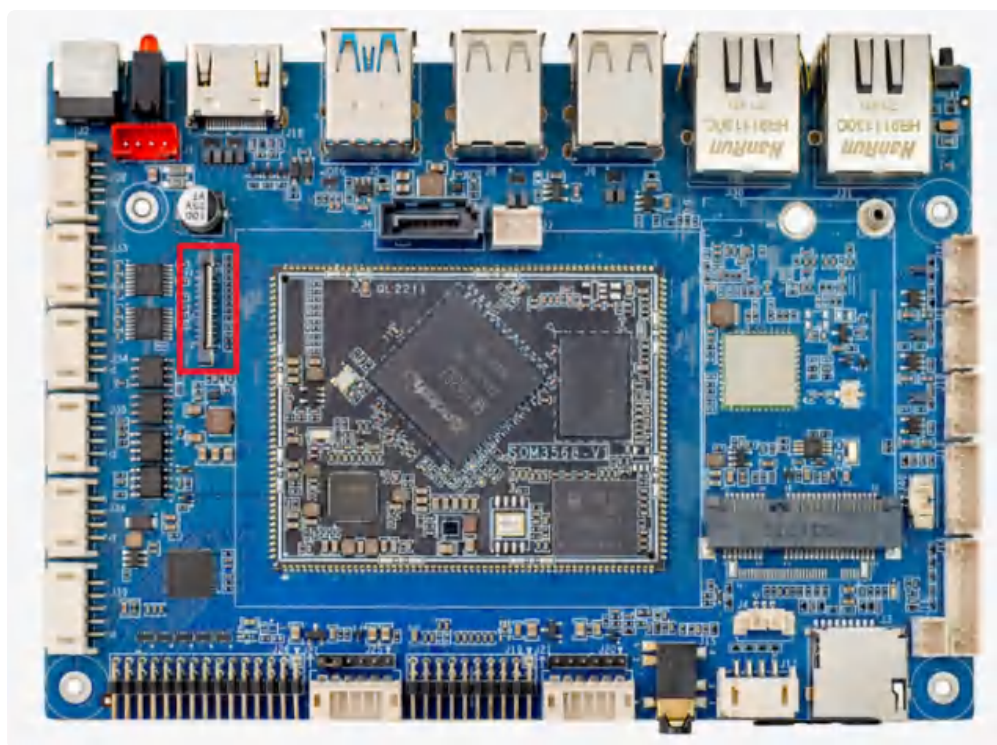
```
1  $ quectel-CM &
```

检查拨号上网是否正常

```
1 $ ping www.baidu.com -I wwan0
```

## 8 摄像头

默认适配OV5648模块，对应系统中的设备节点为/dev/video0。



### 8.1 测试

#### 8.1.1 测试摄像头是否存在

```

1  $ 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

```

结果显示m00\_b\_ov5648，说明摄像头存在，最高分辨率支持2592x1944。

## 8.1.2 抓取视频流

使用v4l2-ctl工具可以抓取摄像头的视频数据流。

```

1  $ v4l2-ctl --verbose -d /dev/video0 --set-fmt-video=width=1920,height=1080
    ,pixelformat='NV12' --stream-mmap=4 --set-selection=target=crop,flags=0,to
    p=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-monotoni
    c, 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

```

### 8.1.3 预览摄像头

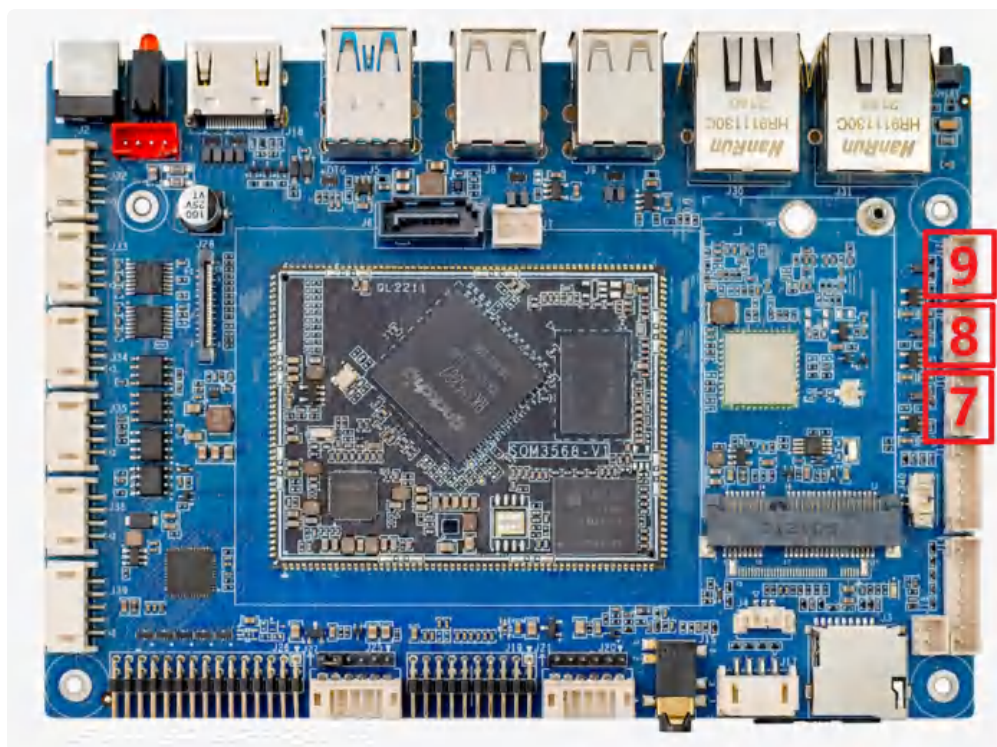
```

▼ Bash |
1 $ gst-launch-1.0 v4l2src device=/dev/video0 ! video/x-raw, format=NV12, width=1920, height=1080, framerate=30/1 ! videoconvert ! autovideosink

```

## 9 U盘

主板共配置9路USB，其中6路TYPE-A接口，3路PH接口。9路USB分别标记1-9，其中1为USB OTG，其余为USB HOST。



## 9.1 电源控制

USB2-8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/usb_host1_pwr/brightness |
| 3  | USB 2.0 HOST | /sys/class/leds/usb_host3_pwr/brightness |
| 4  | USB 2.0 HOST | /sys/class/leds/usb_fe_d1_pwr/brightness |
| 5  | USB 2.0 HOST | /sys/class/leds/usb_fe_d2_pwr/brightness |
| 6  | USB 2.0 HOST | /sys/class/leds/usb_fe_d3_pwr/brightness |
| 7  | USB 2.0 HOST | /sys/class/leds/usb_fe_d4_pwr/brightness |
| 8  | USB 2.0 HOST | /sys/class/leds/usb_fe_d5_pwr/brightness |

打开/关闭 USB电源方法（以USB2为例）：

```
1 # 打开电源
2 echo 255 > /sys/class/leds/usb_host1_pwr/brightness
3
4 # 关闭电源
5 echo 0 > /sys/class/leds/usb_host1_pwr/brightness
```

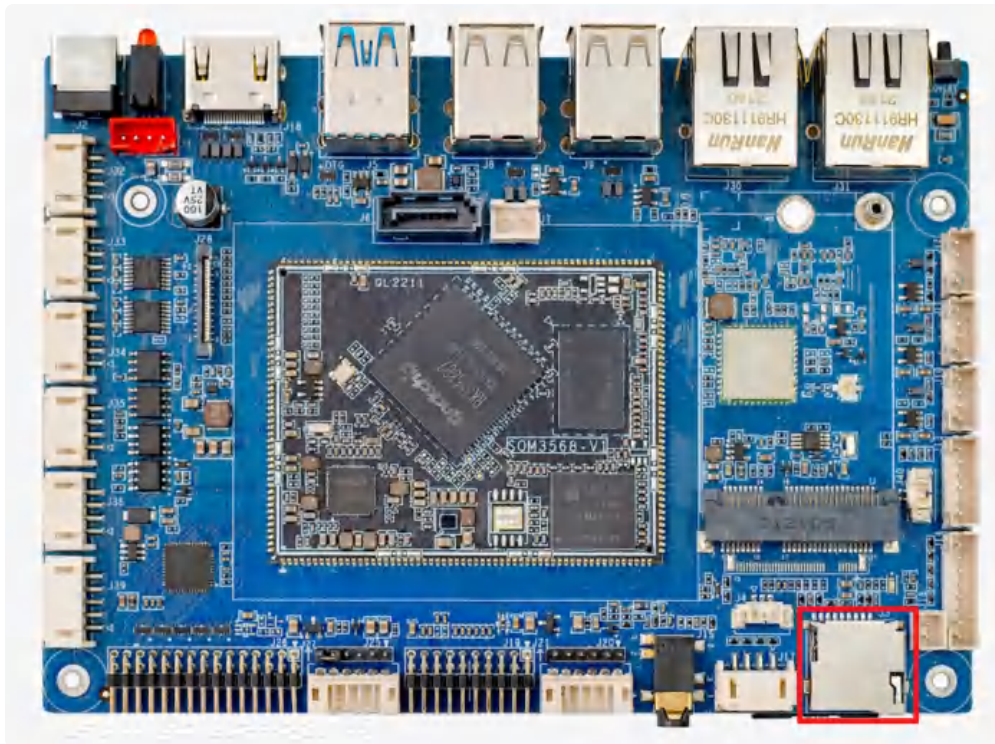
## 9.2 U盘的挂载

当接入U盘设备时，默认挂载到/media/USBX/目录下(X=0,1,2,3,4,5,6,7)

```
1 $ mount
2 ...
3 /dev/sda1 on /media/USB0 type vfat (rw,nodev,noexec,noatime,nodiratime,fmask=0022,dmask=0022,codepage=936,ioccharset=utf8,shortname=mixed,errors=remount-ro)
4 ...
```

## 10 SD卡

将SD卡插入到SD卡槽中，将自动挂载到/mnt/sdcard/目录下。



Bash |

```
1  $ mount
2  ...
3  /dev/mmcblk1p1 on /mnt/sdcard type vfat (rw,noatime,uid=1000,gid=1000,fmask
   =0133,dmask=0022,codepage=936,iocharset=utf8,shortname=mixed,errors=remount
   -ro)
4  ...
```

## 11 开机启动程序

将需要启动的脚本放置到/etc/init.d/目录下，且名字以S开头，可以参考/etc/init.d/目录下的其他启动脚本格式。如S49ntp：

```
1  $ cat /etc/init.d/S49ntp
2  #! /bin/sh
3
4  NAME=ntpd
5  DAEMON=/usr/sbin/$NAME
6
7  # Gracefully exit if the package has been removed.
8  test -x $DAEMON || exit 0
9
10 # Read config file if it is present.
11 if [ -r /etc/default/$NAME ]
12 then
13     . /etc/default/$NAME
14 fi
15
16 case "$1" in
17     start)
18         printf "Starting $NAME: "
19         start-stop-daemon -S -q -x $DAEMON -- -g
20         [ $? = 0 ] && echo "OK" || echo "FAIL"
21         ;;
22     stop)
23         printf "Stopping $NAME: "
24         start-stop-daemon -K -q -n $NAME
25         [ $? = 0 ] && echo "OK" || echo "FAIL"
26         ;;
27     restart|reload)
28         echo "Restarting $NAME: "
29         $0 stop
30         sleep 1
31         $0 start
32         ;;
33     *)
34         echo "Usage: $0 {start|stop|restart|reload}" >&2
35         exit 1
36         ;;
37 esac
38
39 exit 0
```

在开机的时候，会进入到`start`)；在关机的时候，会进入到`stop`)。

## 12 按键

主板配置了一个Recovery按键，如下图所示：



在设备断电的情况下，该按键用于烧录固件。在系统正常启动后，则可作为普通按键使用。对应的设备节点为`/dev/input/event6`，键值为`KEY_VOLUMEUP(115)`。

使用`evtest`进行测试：

```

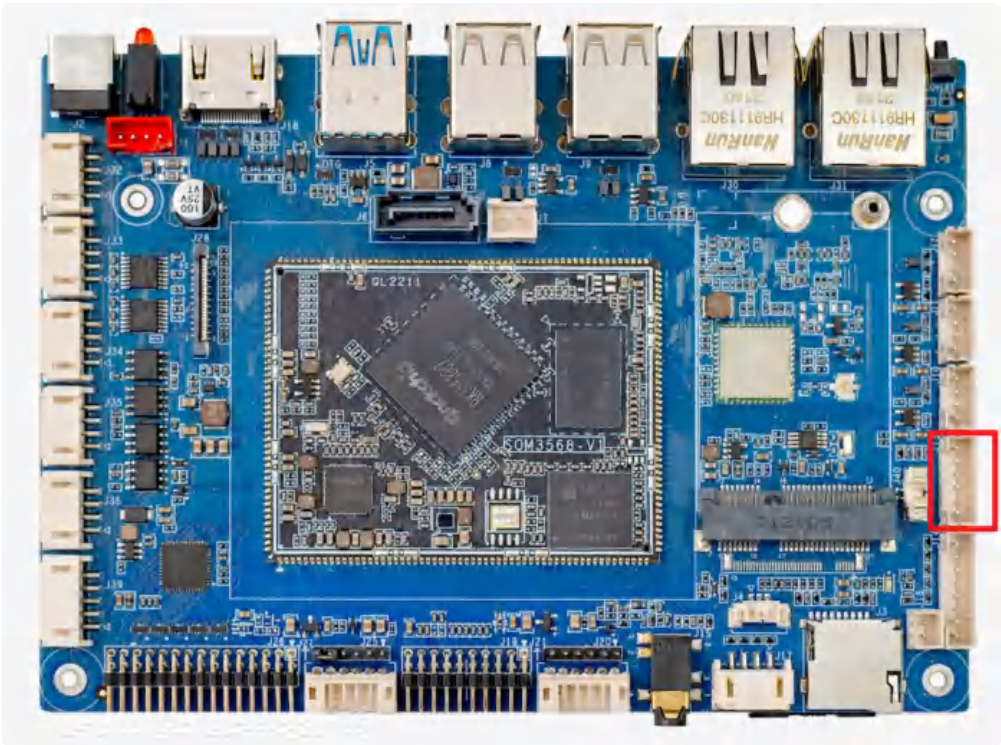
1 ▾ [root@RK356X:/]# evtest
2 No device specified, trying to scan all of /dev/input/event*
3 Available devices:
4 /dev/input/event0:      hdmi_cec_key
5 /dev/input/event1:      fe6e0030.pwm
6 /dev/input/event2:      rockchip,hdmi rockchip,hdmi
7 /dev/input/event3:      bt-powerkey
8 /dev/input/event4:      rk805 pwrkey
9 /dev/input/event5:      rockchip,rk809-codec Headphones
10 /dev/input/event6:      adc-keys
11 ▾ Select the device event number [0-6]: 6
12 Input driver version is 1.0.1
13 Input device ID: bus 0x19 vendor 0x1 product 0x1 version 0x100
14 Input device name: "adc-keys"
15 Supported events:
16   Event type 0 (EV_SYN)
17   Event type 1 (EV_KEY)
18     Event code 114 (KEY_VOLUMEDOWN)
19     Event code 115 (KEY_VOLUMEUP)
20     Event code 139 (KEY_MENU)
21     Event code 158 (KEY_BACK)
22 Properties:
23 Testing ... (interrupt to exit)
24 Event: time 1731566093.194520, type 1 (EV_KEY), code 115 (KEY_VOLUMEUP), v
    alue 1
25 Event: time 1731566093.194520, ----- SYN_REPORT -----
26 Event: time 1731566093.297450, type 1 (EV_KEY), code 115 (KEY_VOLUMEUP), v
    alue 0
27 Event: time 1731566093.297450, ----- SYN_REPORT -----
28 Event: time 1731566093.814204, type 1 (EV_KEY), code 115 (KEY_VOLUMEUP), v
    alue 1
29 Event: time 1731566093.814204, ----- SYN_REPORT -----
30 Event: time 1731566093.917579, type 1 (EV_KEY), code 115 (KEY_VOLUMEUP), v
    alue 0
31 Event: time 1731566093.917579, ----- SYN_REPORT -----

```

在选择event number为6后（对应adc-keys），按下Recovery按键，即可看到按下和松开打印的信息。

## 13 ADC

IDO-EVB3568-V1共配置了3路ADC 接口（精度为10位），位置如下图所示：



设备节点对应关系如下表：

| 接口   | 设备节点                                              |
|------|---------------------------------------------------|
| ADC2 | /sys/bus/iio/devices/iio\:device0/in_voltage2_raw |
| ADC4 | /sys/bus/iio/devices/iio\:device0/in_voltage4_raw |
| ADC5 | /sys/bus/iio/devices/iio\:device0/in_voltage5_raw |

### 13.1 ADC值读取

▼

Bash

1 cat /sys/bus/iio/devices/iio\:device0/in\_voltage2\_raw

### 13.2 ADC电压转换关系

▼

Bash

1 V=(in\_voltage2\_raw/1024)\*1.8v

假设in\_voltage2\_raw的值为500，则对应的ADC电压为 $V=(500/1024)*1.8v=0.879v$

## 14 时间设置 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。

### 14.1 获取RTC时间

```
1 root@rk3568-buildroot:/# hwclock
2 Fri Aug  4 09:00:53 2017  0.000000 seconds
```

### 14.2 设置RTC时间

```
1 root@rk3568-buildroot:/# hwclock
2 Fri Aug  4 09:07:26 2017  0.000000 seconds
3
4 //设置系统时间
5 root@rk3568-buildroot:/# date -s '2023-12-19 16:15:20'
6 Tue Dec 19 16:15:20 UTC 2023
7
8 //把系统时间写入rtc
9 root@rk3568-buildroot:/# hwclock -w
10 root@rk3568-buildroot:/# hwclock -r
11 Tue Dec 19 16:15:25 2023  0.000000 seconds
12
13 //断电重启后，直接hwclock就会把rtc时间写入系统
14 root@rk3568-buildroot:/# hwclock
15 Tue Dec 19 16:16:55 2023  0.000000 seconds
16
17
```

### 14.3 RTC定时开机

主板支持RTC定时开机，设置指令如下：

```
1 echo +0 > /sys/class/rtc/rtc1/wakealarm
2 echo +300 > /sys/class/rtc/rtc1/wakealarm
3
4 poweroff
```

以上指令设置关机300秒后自动开机。

## 14.4 NTP时间同步

系统默认开启了NTP服务，连接网络后，将自动同步网络时间。

## 14.5 时区

查看时区

```
1 $ date -R
2 Wed, 26 Oct 2022 03:26:46 +0000
```

+0000表示在0时区。

设置时区

```
1 $ export TZ='Asia/Shanghai'
2 $
3 $ date -R
4 Wed, 26 Oct 2022 11:30:02 +0800
5 $
```

## 15 音频

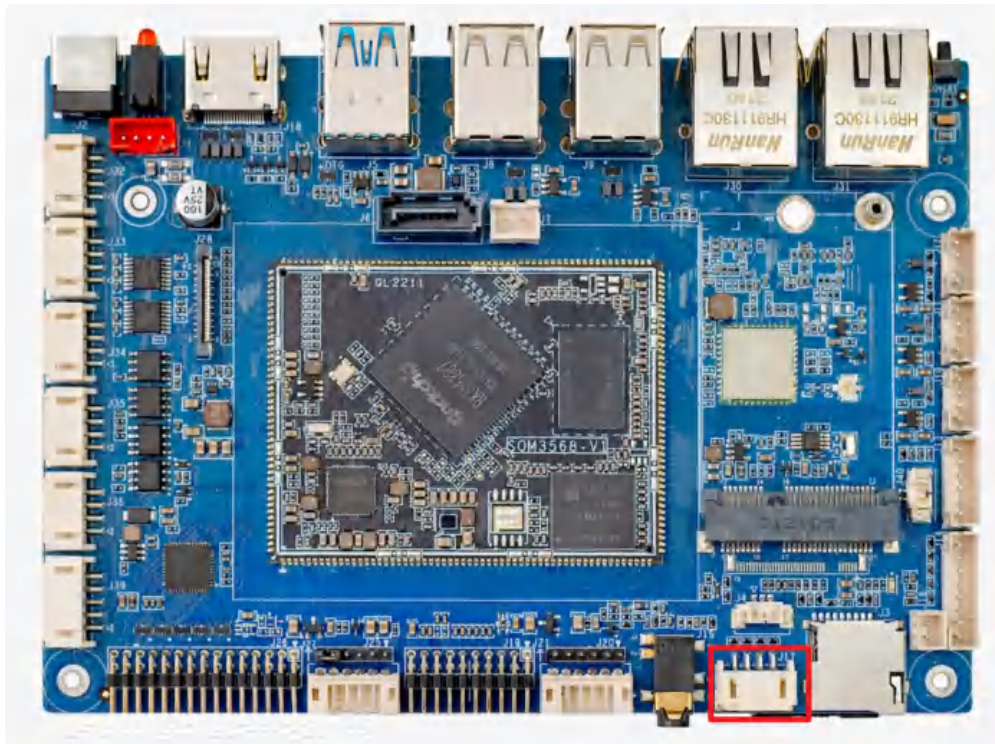
使用aplay工具查看声卡设备

```

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: rockchiprk809co [rockchip,rk809-codec], device 0: fe410000.i2s-rk8
17-hifi rk817-hifi-0 [fe410000.i2s-rk817-hifi rk817-hifi-0]
34      Subdevices: 1/1
35      Subdevice #0: subdevice #0
36  card 3: rockchiphdmi [rockchip,hdmi], device 0: rockchip,hdmi i2s-hifi-0
[rockchip,hdmi i2s-hifi-0]
37      Subdevices: 1/1
38      Subdevice #0: subdevice #0
39  $

```

## 14.1 Lineout



打开Lineout开关：

```
▼ Shell |
1 $ amixer -c 2 cset numid=12,iface=MIXER,name='Playback Path' 'SPK'
```

不插入耳机，使用aplay播放wav音频测试

```
▼ Bash |
1 $ aplay -Dhw:2,0 ./8k16bpsStereo.wav
2 Playing WAVE './8k16bpsStereo.wav' : Signed 16 bit Little Endian, Rate 8000
  Hz, Stereo
```

[8k16bpsStereo.wav](#)

HDMI 音频

```
▼ Bash |
1 $ aplay -Dhw:3,0 ./8k16bpsStereo.wav
```

## 14.2 耳机

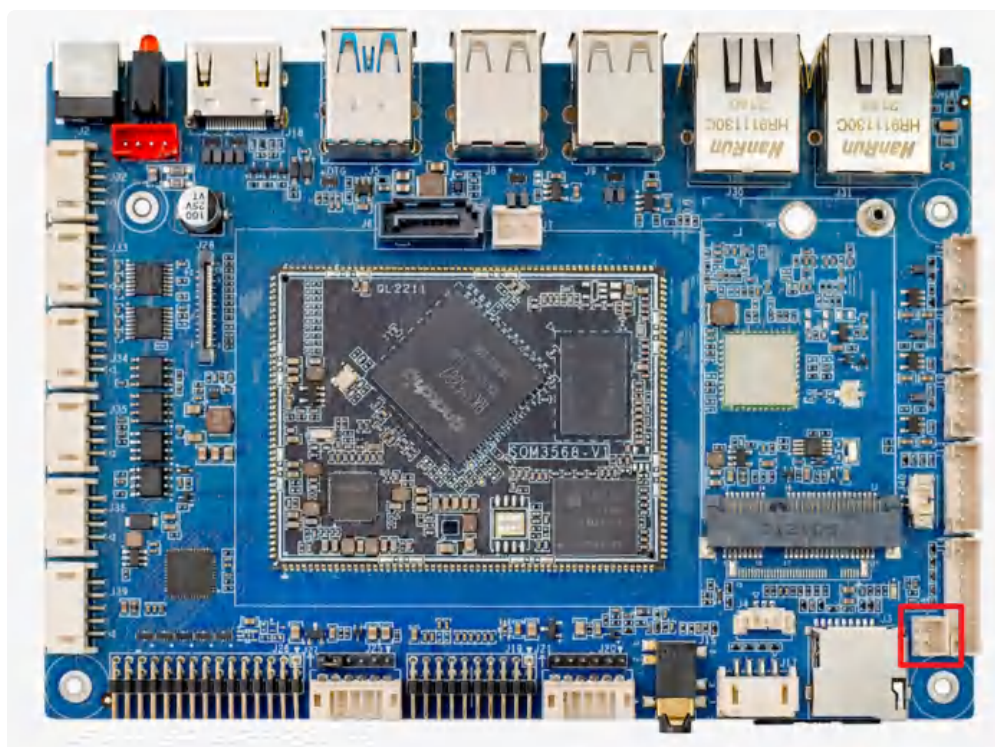


插入耳机，使用aplay播放wav音频测试：

Bash |

```
1 $ aplay -Dhw:2,0 ./8k16bpsStereo.wav
2 Playing WAVE './8k16bpsStereo.wav' : Signed 16 bit Little Endian, Rate 8000
   Hz, Stereo
```

## 14.3 录音



查看录音设备：

```
1  $ arecord -l
2  **** List of CAPTURE 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: rockchiprk809co [rockchip,rk809-codec], device 0: fe410000.i2s-rk8
    17-hifi rk817-hifi-0 [fe410000.i2s-rk817-hifi rk817-hifi-0]
34     Subdevices: 1/1
35     Subdevice #0: subdevice #0
36  $
37
```

打开Mic开关:

```
1 $ amixer -c 2 cset numid=13,iface=MIXER,name='Capture MIC Path' 'Main Mic'
```

使用arecord工具进行录音测试。

```
1 $ arecord -D hw:2,0 -r 48000 -c 2 -f S16_LE test.wav
2 Recording WAVE 'test.wav' : Signed 16 bit Little Endian, Rate 48000 Hz, Stereo
3 ^CAborted by signal Interrupt...
```

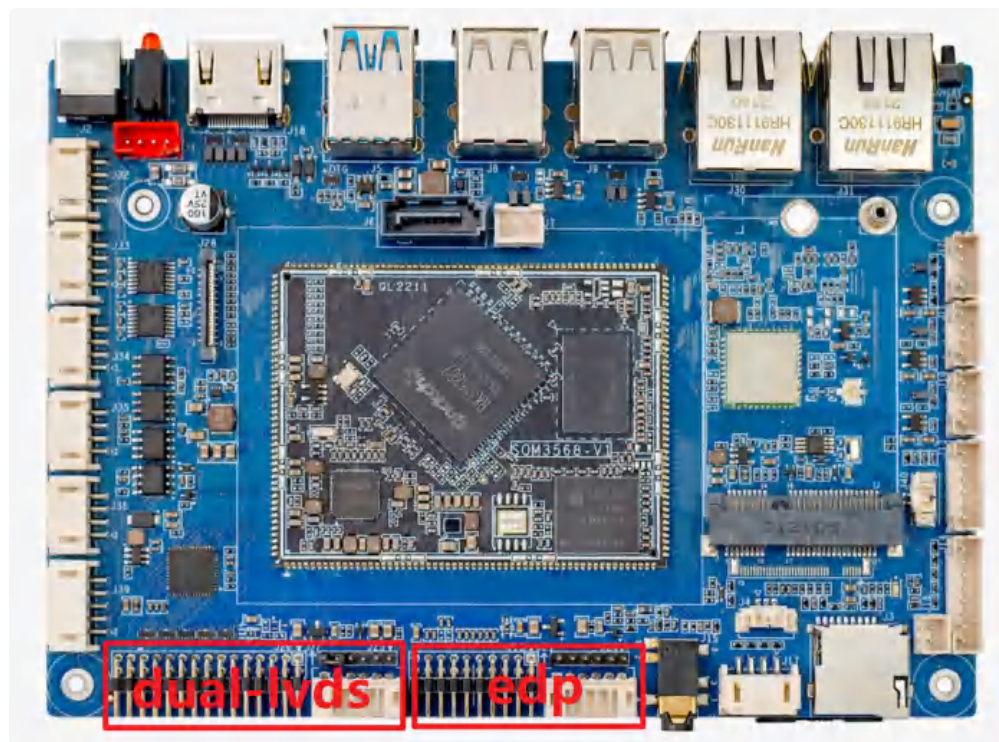
播放录音：

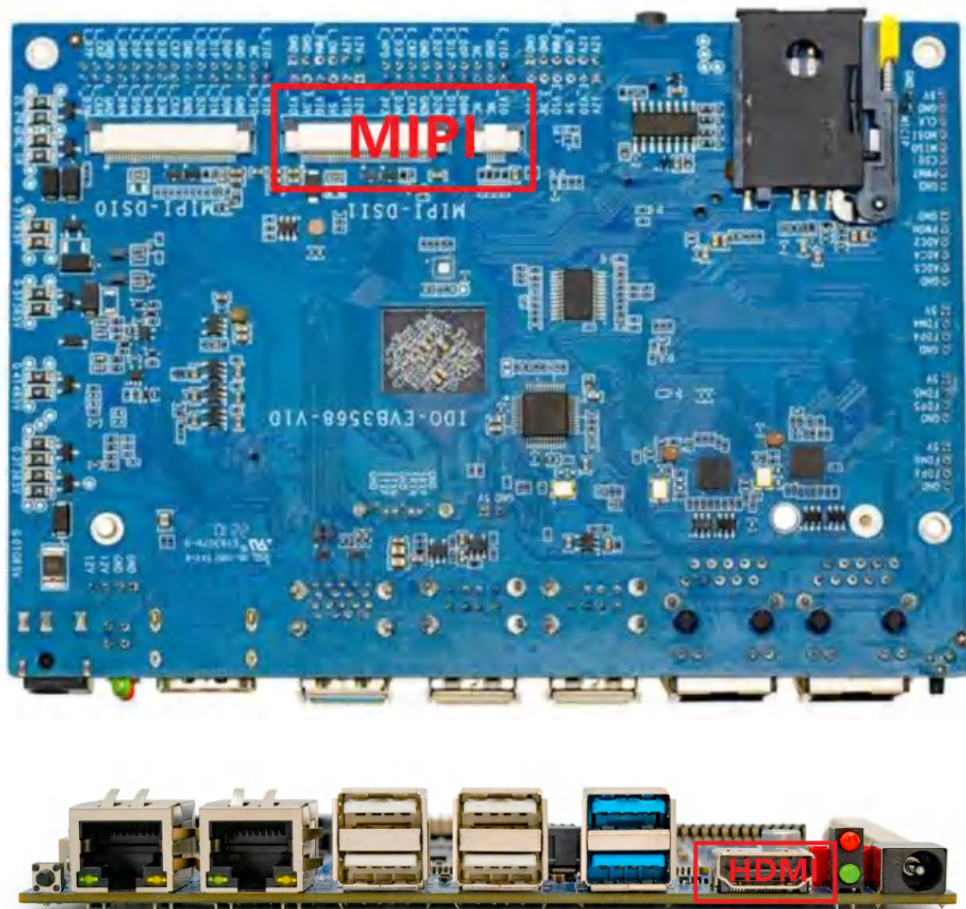
```
1 $ aplay -Dhw:2,0 test.wav
```

## 16 显示屏

### 16.1 显示屏接口说明

主板各个显示接口如下图所示：





## 16.2.1 屏幕背光亮度设置

- eDP/MIPI屏背光控制

设备节点: `/sys/class/backlight/backlight/brightness`

设置方法: (支持调节范围 0-255)

▼

Bash |

```
1 #关闭
2 echo 0 > /sys/class/backlight/backlight/brightness
3 #最亮
4 echo 255 > /sys/class/backlight/backlight/brightness
```

- Dual LVDS屏幕背光控制

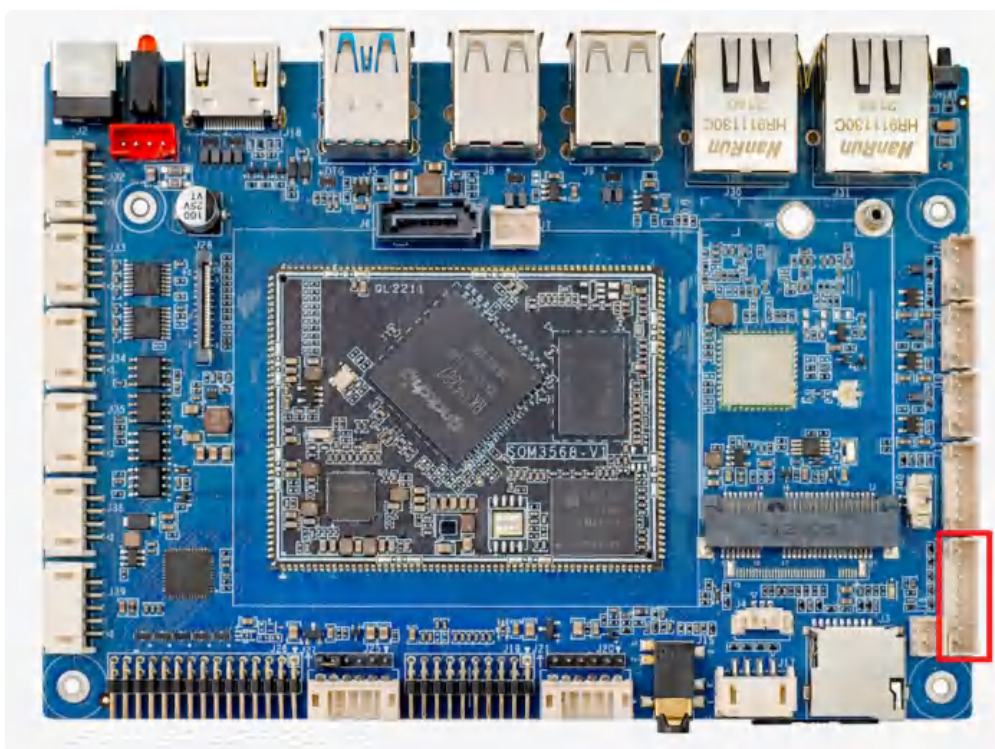
设备节点: `/sys/class/backlight/backlight1/brightness`

设置方法: (支持调节范围 0-255)

```
1 #关闭
2 echo 0 > /sys/class/backlight/backlight1/brightness
3 #最亮
4 echo 255 > /sys/class/backlight/backlight1/brightness
```

## 17 扩展GPIO

主板扩展了5路GPIO，位于J14：



各个引脚的功能描述如下表所示：

| 序号 | 定义                                | 电平/V | 说明                  |
|----|-----------------------------------|------|---------------------|
| 1  | 5V                                | 5V   | 电源5V输出              |
| 2  | GND                               | GND  | 电源地                 |
| 3  | SPI3_CLK_M1/GPIO4_C2/PWM14_M1     | 3.3V | 默认配置为GPIO，GPIO编号146 |
| 4  | SPI3_MOSI_M1/GPIO4_C3/PWM15_IR_M1 | 3.3V | 默认配置为GPIO，GPIO编号147 |

|   |                                |      |                      |
|---|--------------------------------|------|----------------------|
| 5 | SPI3_MISO_M1/GPIO4_C5/PWM12_M1 | 3.3V | 默认配置为GPIO, GPIO编号149 |
| 6 | SPI3_CS0_M1/GPIO4_C6/PWM13_M1  | 3.3V | 默认配置为GPIO, GPIO编号150 |
| 7 | PWM7_IR/GPIO0_C6               | 3.3V | 默认配置为GPIO, GPIO编号22  |
| 8 | GND                            | GND  | 电源地                  |

## 17.1 测试方法

以GPIO4\_C2为例。

作为输入：

```

1 echo 146 > /sys/class/gpio/export
2 echo in > /sys/class/gpio/gpio146/direction
3 cat /sys/class/gpio/gpio146/value

```

value值为0, 说明该gpio输入低电平; value值为1, 说明该gpio输入高电平。

作为输出：

```

1 echo 146 > /sys/class/gpio/export
2 echo out > /sys/class/gpio/gpio146/direction
3 #输出高电平
4 echo 1 > /sys/class/gpio/gpio146/value
5 #输出低电平
6 echo 0 > /sys/class/gpio/gpio146/value

```

## IDO-EVB3568-V1 Debian系统

### 1 系统用户名及密码

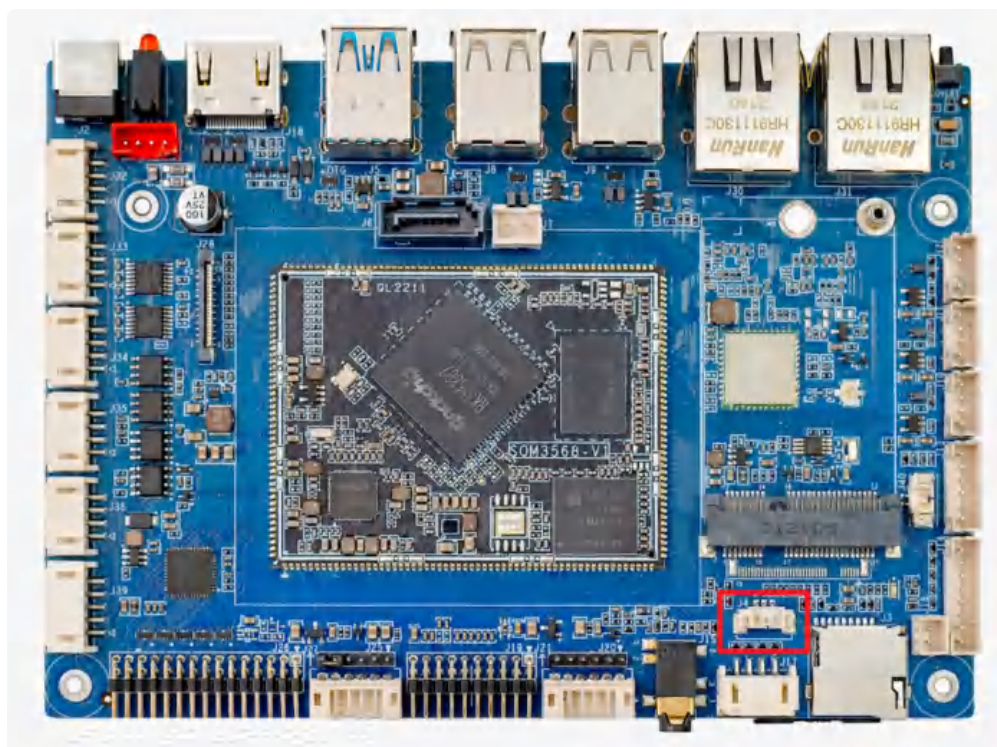
系统用户名及密码如下表所示：

| 文件系统版本   | 账户     | 密码     |
|----------|--------|--------|
| Debian11 | root   | 未设置密码  |
|          | linaro | linaro |

## 2 调试口使用

### 2.1 串口调试

串口调试接口位于J4端口，见下图。请使用配套的USB串口调试工具。



调试串口为TTL电平，通信参数为1500000 8 N 1。

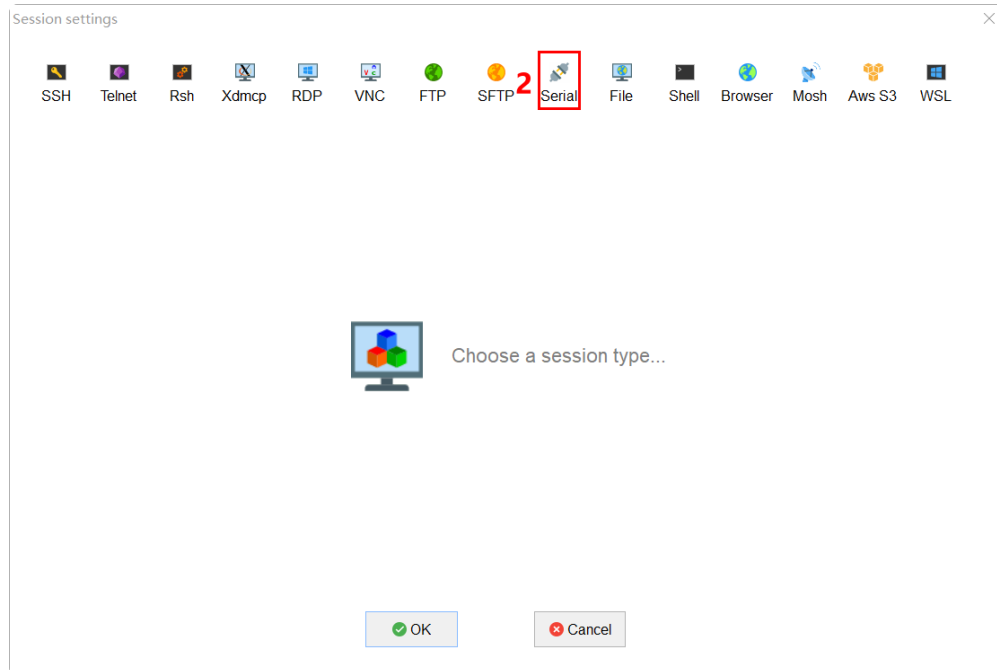
1. 打开MobaXterm，下载链接如下：

链接：[https://pan.baidu.com/s/11ui4LTd2mq\\_9kiJpeL4bWg?pwd=1234](https://pan.baidu.com/s/11ui4LTd2mq_9kiJpeL4bWg?pwd=1234)

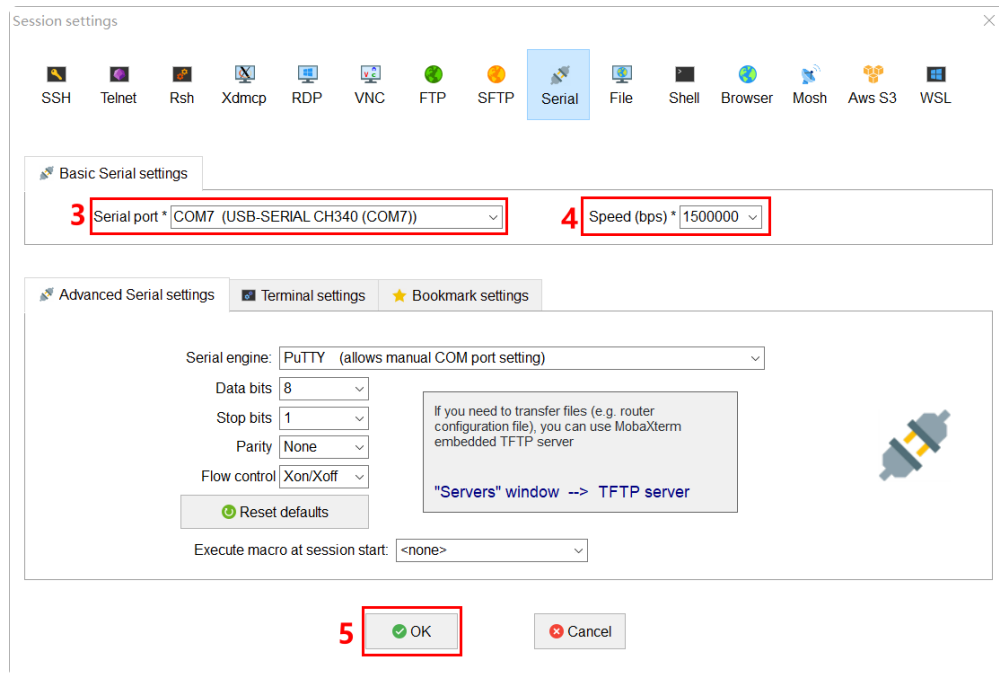
提取码：1234

| <input type="checkbox"/> 文件名                          | ↓ 修改时间           | 类型    | 大小      |
|-------------------------------------------------------|------------------|-------|---------|
| <input type="checkbox"/> RKDevTool_Release_v2.95.zip  | 2024-04-24 11:53 | zip文件 | 2.30MB  |
| <input type="checkbox"/> other_tools.txt              | 2024-04-25 15:31 | txt文件 | 44B     |
| <input type="checkbox"/> MobaXterm_Portable_v23.6.zip | 2024-04-24 14:30 | zip文件 | 39.99MB |
| <input type="checkbox"/> DriverAssitant_v5.11.zip     | 2024-04-24 11:52 | zip文件 | 9.36MB  |

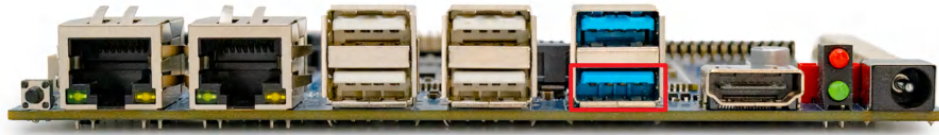
2. 选择sessionSerial，如下图所示：



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



## 2.2 ADB调试



上图红色框内的USB接口为支持OTG模式切换，在系统上电前，使用双公头 USB 数据线连接开发板和 PC 端的USB接口，在PC终端识别到 ADB 设备，即可使用 adb shell 调试。

```
>adb.exe devices
List of devices attached
a4b21b7e423198ac    device

>adb.exe shell
root@linaro-alip:/# ls
bin    etc    lost+found  mnt    rockchip_test  sbin    system  var
boot  home  md5sum.txt  opt    root           srv     tmp     vendor
dev    lib    media      proc   run            sys     usr
```

## 2.3 SSH调试

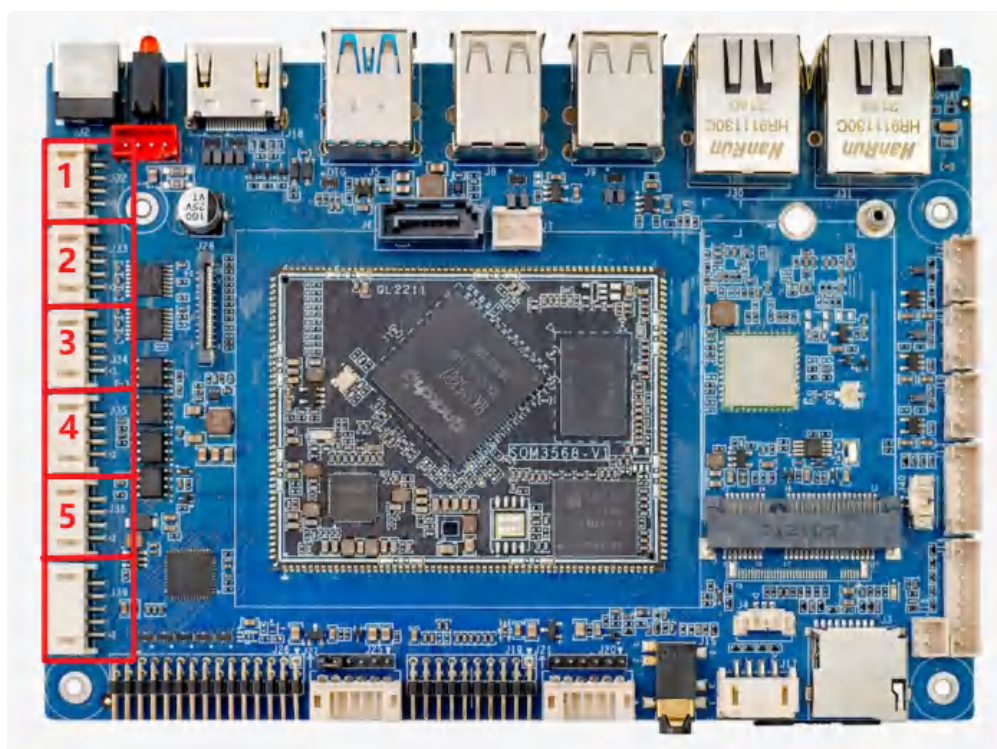
SSH登录账号信息：

账号：linaro

密码：linaro

```
linaro@linaro-alip: ~  
login as: linaro  
linaro@192.168.1.160's password:  
Linux linaro-alip 5.10.209 #98 SMP Fri Oct 25 11:37:05 CST 2024 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:~$ ls  
Desktop  
linaro@linaro-alip:~$
```

### 3 串口



串口接口位置及引脚定义如上图所示，设备节点如下表所示：

| 序号 | 功能    | 设备节点       |
|----|-------|------------|
| 1  | TTL   | /dev/ttyS0 |
| 2  | RS232 | /dev/ttyS3 |
| 3  | RS232 | /dev/ttyS4 |
| 4  | RS485 | /dev/ttyS5 |
| 5  | RS485 | /dev/ttyS7 |

使用工具microcom, 可以进行发送和接收测试。

需要先安装microcom工具:

▼

Bash |

```
1 sudo apt-get update
2 sudo apt-get install microcom
```

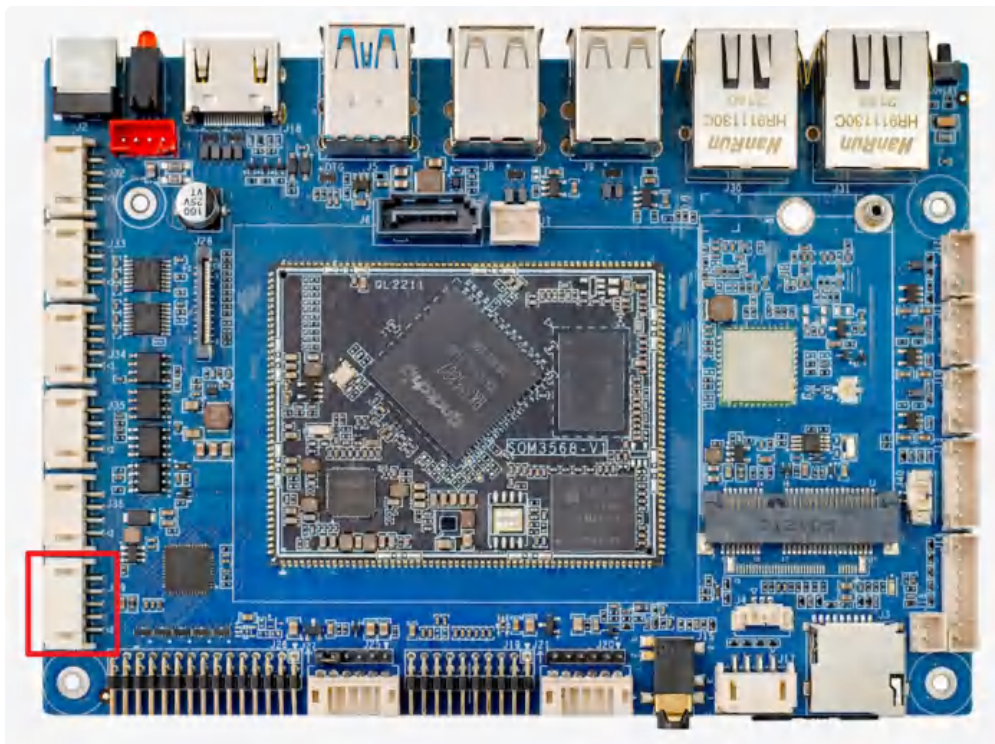
▼

Plain Text |

```
1 root@ido:~# microcom -s 115200 -p /dev/ttyS0
2 [ 44.730195] of_dma_request_slave_channel: dma-names property of node '/s
   erial@fdd50000' missing or empty
3 connected to /dev/ttyS0
4 Escape character: Ctrl-\
5 Type the escape character to get to the prompt.
6 fjskdfjsdfjsdklfjdsfdsdffdfdsfsdfd
```

按下键盘, 将发送对应的字符; 而接收的内容, 会显示在终端。

## 4 CAN



IDO-EVB3568-V1共配置两路CAN接口, 分别为CAN0和CAN1。支持 CAN2.0协议, CAN接口测试方法如下:

```

1  #关闭can0设备
2  ip link set can0 down
3
4  #设置通信波特率为250Kbps
5  ip link set can0 type can bitrate 250000
6
7  #打印can0信息
8  ip -details link show can0
9
10 #启动can0
11 ip link set can0 up
12
13 #执行candump, 阻塞等待can0接收
14 candump can0
15
16 #发送数据
17 cansend can0 123#DEADBEEF

```

## 5 WiFi使用

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

```

### 5.1 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

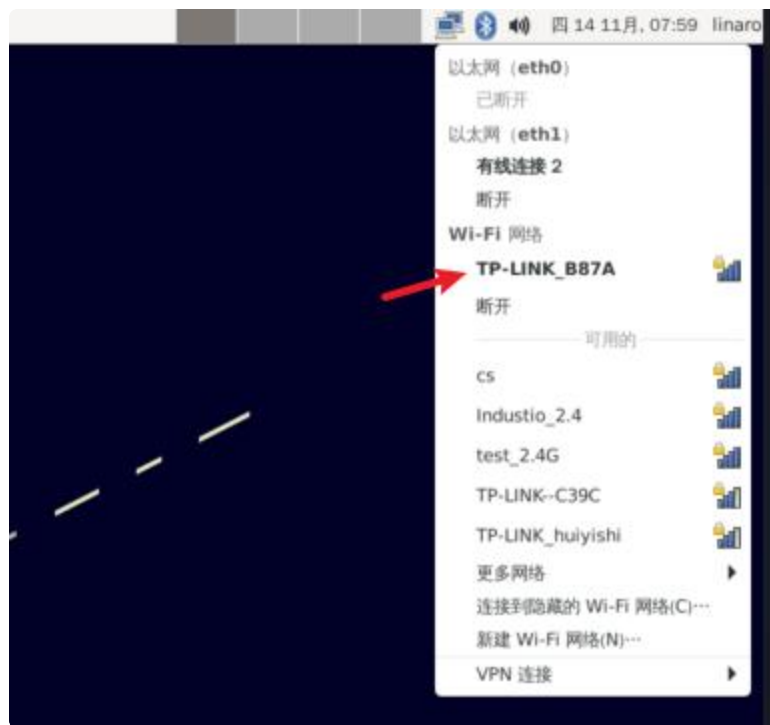
点击桌面右上角的网络图标，即可看到WiFi热点列表：



点击要连接的热点，弹出密码输入窗口：



使用键盘输入密码后，点击【连接】连接热点：



## 方式二

使用wpa\_cli工具连接。

```

1 root@linaro-alip:~# wpa_cli -i wlan0 add_network
2 0
3 root@linaro-alip:~# wpa_cli -i wlan0 set_network 0 ssid '"TP-LINK_B87A"'
4 OK
5 root@linaro-alip:~# wpa_cli -i wlan0 set_network 0 psk '"12345678"'
6 OK
7 root@linaro-alip:~# wpa_cli -i wlan0 enable_network 0
8 OK
9 root@linaro-alip:~# dhclient -i wlan0

```

连接成功后，查看wlan0获取的IP地址：

```

1 root@linaro-alip:~# ifconfig wlan0
2 wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
3     inet 192.168.1.191 netmask 255.255.255.0 broadcast 192.168.1.255
4     ether e8:fb:1c:88:4d:ab txqueuelen 1000 (Ethernet)
5     RX packets 82 bytes 8843 (8.6 KiB)
6     RX errors 0 dropped 665 overruns 0 frame 0
7     TX packets 66 bytes 7257 (7.0 KiB)
8     TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

## 6 蓝牙使用

主板开机默认会打开蓝牙，设备节点为hci0：

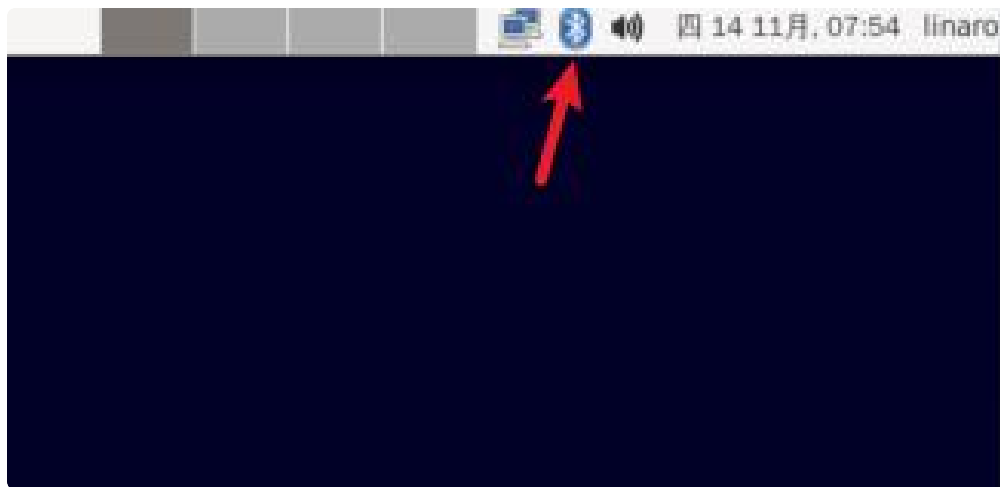
```

1 root@linaro-alip:~# hciconfig
2 hci0: Type: Primary Bus: UART
3     BD Address: E8:FB:1C:88:4D:AA ACL MTU: 1021:8 SCO MTU: 64:1
4     UP RUNNING
5     RX bytes:3772 acl:0 sco:0 events:129 errors:0
6     TX bytes:5539 acl:0 sco:0 commands:73 errors:0

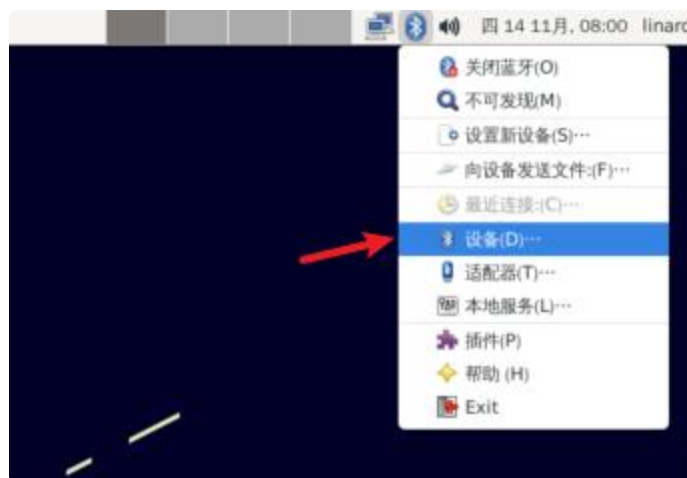
```

### 6.1 连接蓝牙设备

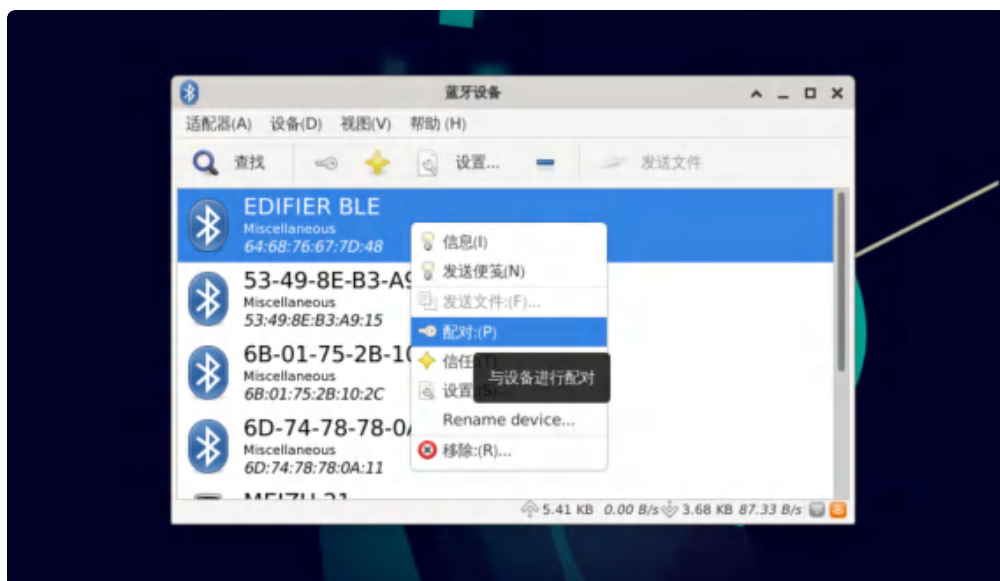
点击桌面右上角蓝牙图标：



点击【设备】：

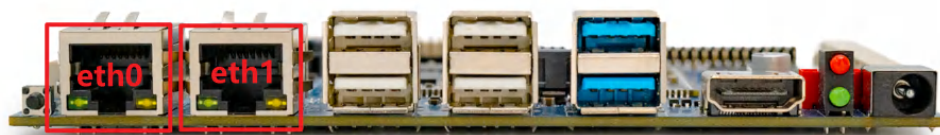


将看到附近的蓝牙设备列表，选中要连接设备，右键->配对，即可连接该设备：



## 7 以太网使用

开发板两路千兆以太网接口，分别为eth0和eth1。



两路以太网接口默认IP获取方式为 DHCP。

### 7.1 静态IP设置

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

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

### 7.2 DNS设置

为开发添加114.114.114.114的DNS，修改/etc/network/interfaces

```
1  dns-nameservers 114.114.114.114
```

## 8 4G使用

开发板带有PCIe座子，可连接 EC20 4G模块。默认系统支持移远拨号管理软件 quectel-CM系统默认适配了 EC20 miniPCIe 4G模块，支持自动拨号上网。使用 4G 功能前，需要先在板载 mini PCIe 接

口插入 EC20 模块，在卡槽插入 SIM 卡，并连接好 4G 天线以保证信号的稳定。

系统已添加 EC20 网络连接管理工具 quectel-CM 用于拨号服务，并自动识别模块插入状态启动服务程序，手动开关服务的方法如下：

```
▼ Bash |
1  #停止拨号服务
2  quectel-CM &
3  #开启拨号服务
4  killall quectel-CM
```

正常拨号成功后，wwan0将会分配到ip地址，此时可以测试是否能够正常ping通外网。

```
▼ Bash |
1  ping www.baidu.com -I wwan0
```

## 9 摄像头使用

### 9.1 测试

#### 9.1.1 测试摄像头是否存在

```
1 root@linaro-alip:/# 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 抓取视频流

使用v4l2-ctl工具可以抓取摄像头的视频数据流。

```

1  root@linaro-alip:/# 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

```

### 9.1.3 预览摄像头

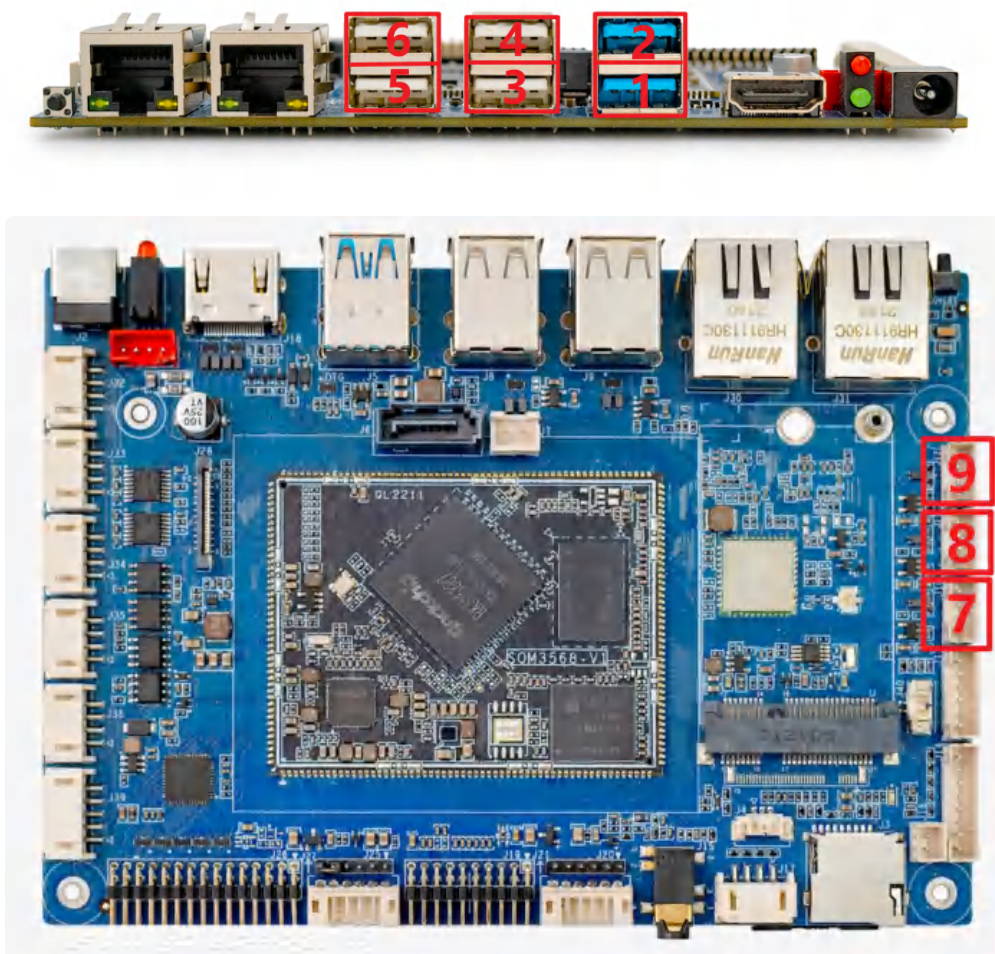
```

▼ Bash |
1  gst-launch-1.0 v4l2src device=/dev/video0 ! video/x-raw, format=NV12, width
   =1920, height=1080, framerate=30/1 ! autovideoconvert ! autovideosink

```

## 10 U盘

主板共配置9路USB，其中6路TYPE-A接口，3路PH接口。9路USB分别标记1-9，其中1为USB OTG，其余为USB HOST。



## 10.1 电源控制

USB2-8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/usb_host1_pwr/brightness |
| 3  | USB 2.0 HOST | /sys/class/leds/usb_host3_pwr/brightness |
| 4  | USB 2.0 HOST | /sys/class/leds/usb_fe_d1_pwr/brightness |
| 5  | USB 2.0 HOST | /sys/class/leds/usb_fe_d2_pwr/brightness |
| 6  | USB 2.0 HOST | /sys/class/leds/usb_fe_d3_pwr/brightness |
| 7  | USB 2.0 HOST | /sys/class/leds/usb_fe_d4_pwr/brightness |
| 8  | USB 2.0 HOST | /sys/class/leds/usb_fe_d5_pwr/brightness |

打开/关闭 USB电源方法（以USB2为例）：

```
1 # 打开电源
2 echo 255 > /sys/class/leds/usb_host1_pwr/brightness
3
4 # 关闭电源
5 echo 0 > /sys/class/leds/usb_host1_pwr/brightness
```

## 10.2 U盘的挂载

当接入U盘设备时，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 ...
```

## 11 设置开机启动程序

- 非界面程序自启动

通过srevice方式添加开机自启动服务。

## 12 按键

主板配置了一个Recovery按键，如下图所示：



在设备断电的情况下，该按键用于烧录固件。在系统正常启动后，则可作为普通按键使用。对应的设备节点为/dev/input/event6，键值为KEY\_VOLUMEUP(115)。

使用evtest进行测试，没有evtest指令则使用以下命令安装：

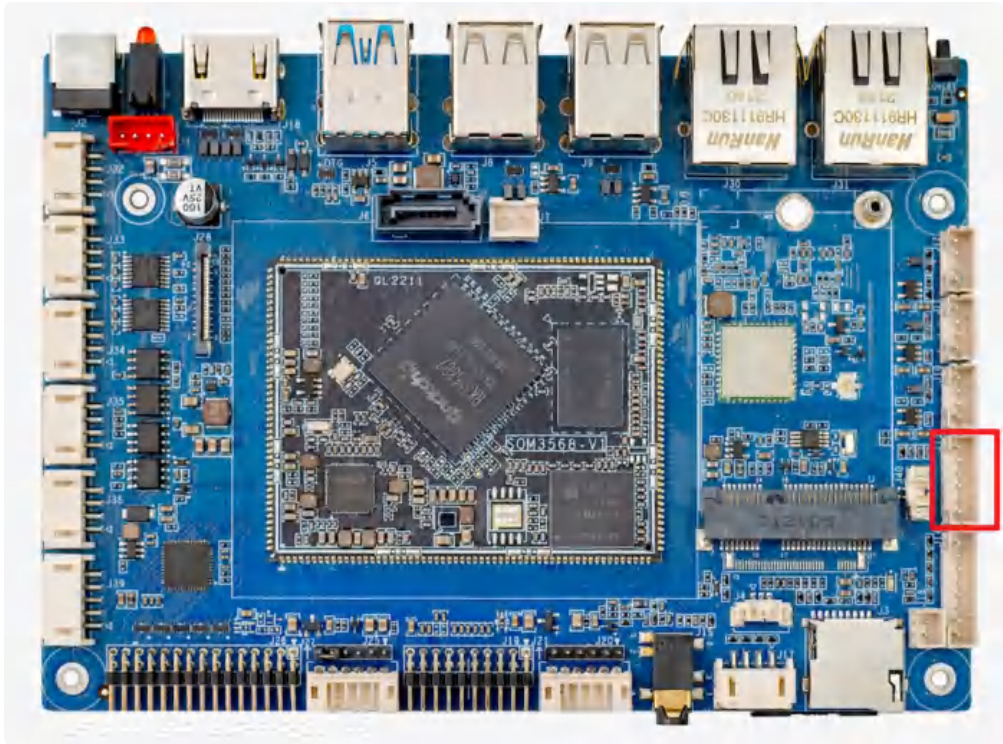
```
1 apt-get install evtest
```

```
1 root@linaro-alip:/# evtest
2 No device specified, trying to scan all of /dev/input/event*
3 Available devices:
4 /dev/input/event0: rk805 pwrkey
5 /dev/input/event1: bt-powerkey
6 /dev/input/event2: hdmi_cec_key
7 /dev/input/event3: adc-keys
8 /dev/input/event4: rockchip,hdmi rockchip,hdmi
9 Select the device event number [0-4]: 3
10 Input driver version is 1.0.1
11 Input device ID: bus 0x19 vendor 0x1 product 0x1 version 0x100
12 Input device name: "adc-keys"
13 Supported events:
14   Event type 0 (EV_SYN)
15   Event type 1 (EV_KEY)
16     Event code 114 (KEY_VOLUMEDOWN)
17     Event code 115 (KEY_VOLUMEUP)
18     Event code 139 (KEY_MENU)
19     Event code 158 (KEY_BACK)
20 Properties:
21 Testing ... (interrupt to exit)
22 Event: time 1763696496.675245, type 1 (EV_KEY), code 115 (KEY_VOLUMEUP), v
    alue 1
23 Event: time 1763696496.675245, ----- SYN_REPORT -----
24 Event: time 1763696496.881894, type 1 (EV_KEY), code 115 (KEY_VOLUMEUP), v
    alue 0
25 Event: time 1763696496.881894, ----- SYN_REPORT -----
26
```

在选择event number为6后（对应adc-keys），按下Recovery按键，即可看到按下和松开打印的信息。

## 13 ADC

主板共配置了3路ADC 接口（精度为10位），位置如下图所示：



设备节点对应关系如下表：

| 接口   | 设备节点                                              |
|------|---------------------------------------------------|
| ADC2 | /sys/bus/iio/devices/iio\:device0/in_voltage2_raw |
| ADC4 | /sys/bus/iio/devices/iio\:device0/in_voltage4_raw |
| ADC5 | /sys/bus/iio/devices/iio\:device0/in_voltage5_raw |

### 13.1 ADC值读取

▼

Bash

1 cat /sys/bus/iio/devices/iio\:device0/in\_voltage2\_raw

### 13.2 ADC电压转换关系

▼

Bash

1 V=(in\_voltage2\_raw/1024)\*1.8v

假设in\_voltage2\_raw的值为500，则对应的ADC电压为 $V=(500/1024)*1.8v=0.879v$

# 14 时间设置

## 14.1 RTC时间读取和同步

系统时间读取和设置

▼ Bash

```
1 root@linaro-alip:/home/linaro# date
2 Fri Mar 18 12:00:22 CST 2022
3 root@linaro-alip:/home/linaro# date -s "2022-03-18 12:01:00"
```

RTC时间设置

▼ Bash

```
1 root@linaro-alip:/home/linaro# hwclock -r
2 2022-03-18 12:01:06.991425+08:00
```

## 14.2 RTC定时开机

主板支持RTC定时开机，设置指令如下：

▼ Shell

```
1 echo +0 > /sys/class/rtc/rtc1/wakealarm
2 echo +300 > /sys/class/rtc/rtc1/wakealarm
3
4 poweroff
```

以上指令设置关机300秒后自动开机。

## 14.3 NTP时间同步

系统默认开启了NTP服务，连接网络后，将自动同步网络时间。

## 14.4 时区

查看时区

▼ Bash |

```
1 root@linaro-alip:/home/linaro# date -R
2 Wed, 26 Oct 2022 03:26:46 +0000
```

+0000表示在0时区。

设置时区

▼ Bash |

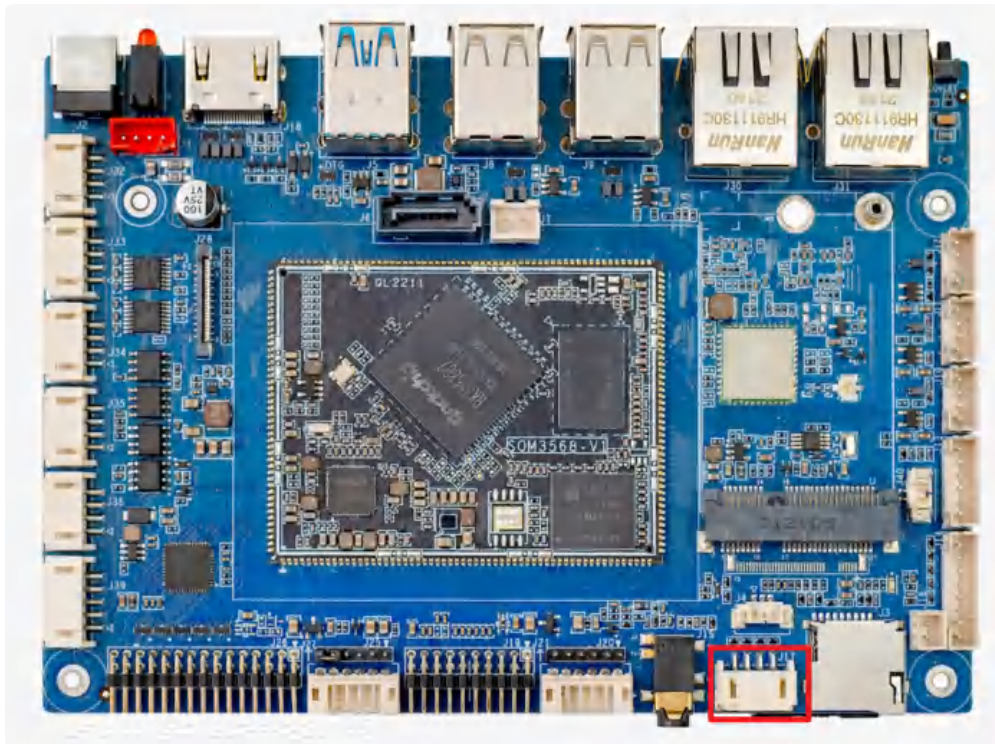
```
1 root@linaro-alip:/home/linaro# export TZ='Asia/Shanghai'
2 root@linaro-alip:/home/linaro# date -R
3 Wed, 26 Oct 2022 11:30:02 +0800
```

## 15 音频

使用[aplay](#)工具查看声卡设备

```
1 root@linaro-alip:/# 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: rockchiprk809co [rockchip,rk809-codec], device 0: fe410000.i2s-rk8
    17-hifi rk817-hifi-0 [fe410000.i2s-rk817-hifi rk817-hifi-0]
34     Subdevices: 1/1
35     Subdevice #0: subdevice #0
36 ▾ card 3: rockchiphdmi [rockchip,hdmi], device 0: rockchip,hdmi i2s-hifi-0
    [rockchip,hdmi i2s-hifi-0]
37     Subdevices: 1/1
38     Subdevice #0: subdevice #0
39
```

## 15.1 Lineout



不插入耳机，使用aplay播放wav音频测试

Bash

```
1 root@linaro-alip:/# aplay -Dhw:2,0 ./8k16bpsStereo.wav
2 Playing WAVE './8k16bpsStereo.wav' : Signed 16 bit Little Endian, Rate 8000 Hz, Stereo
```

 8k16bpsStereo.wav

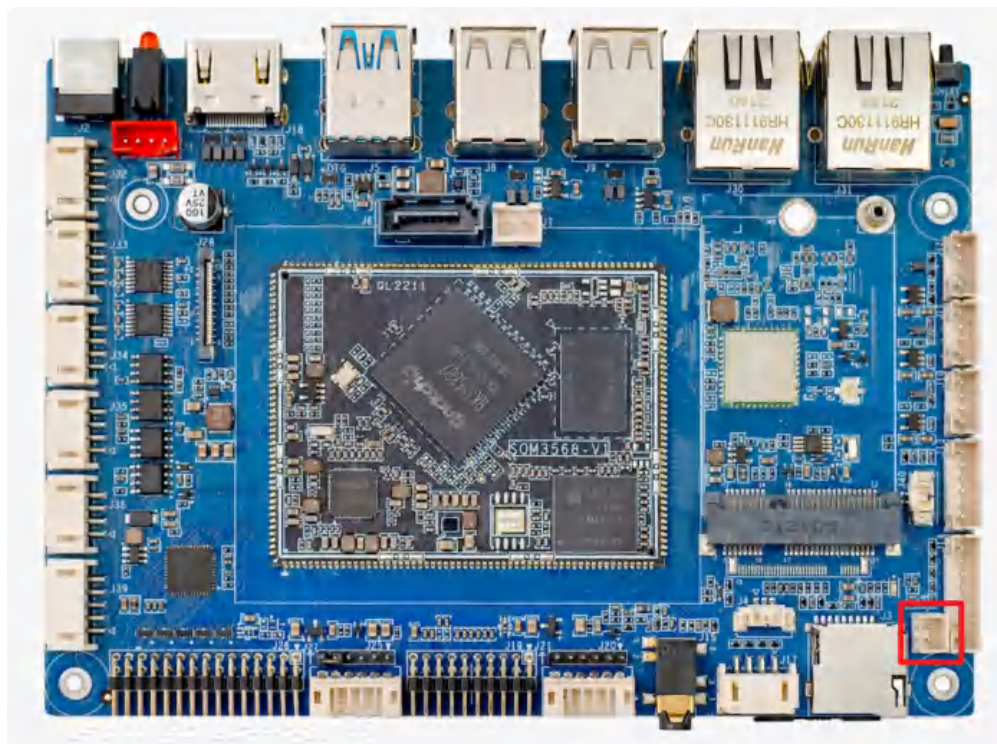
## 15.2 耳机



插入耳机，使用aplay播放wav音频测试

```
1 root@linaro-alip:/# aplay -Dhw:2,0 ./8k16bpsStereo.wav
2 Playing WAVE './8k16bpsStereo.wav' : Signed 16 bit Little Endian, Rate 8000
  Hz, Stereo
```

## 15.3 录音



查看录音设备：

```
1 root@linaro-alip:/# arecord -l
2 **** List of CAPTURE 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: rockchiprk809co [rockchip,rk809-codec], device 0: fe410000.i2s-rk8
    17-hifi rk817-hifi-0 [fe410000.i2s-rk817-hifi rk817-hifi-0]
34     Subdevices: 1/1
35     Subdevice #0: subdevice #0
36
```

使用arecord工具进行录音测试。

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

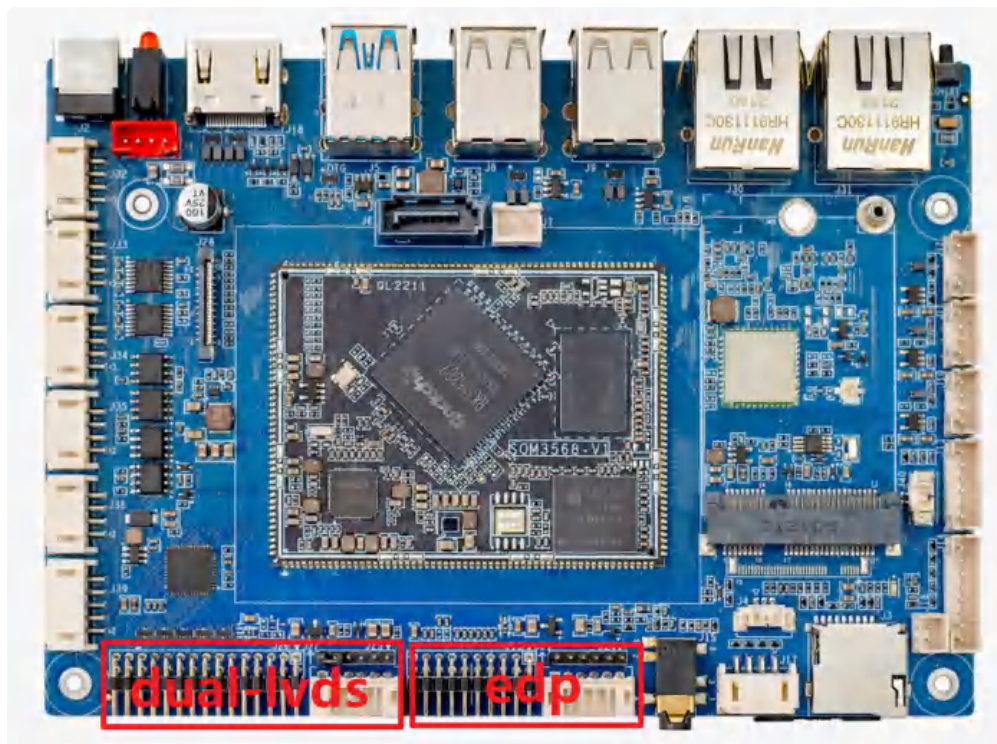
播放录音：

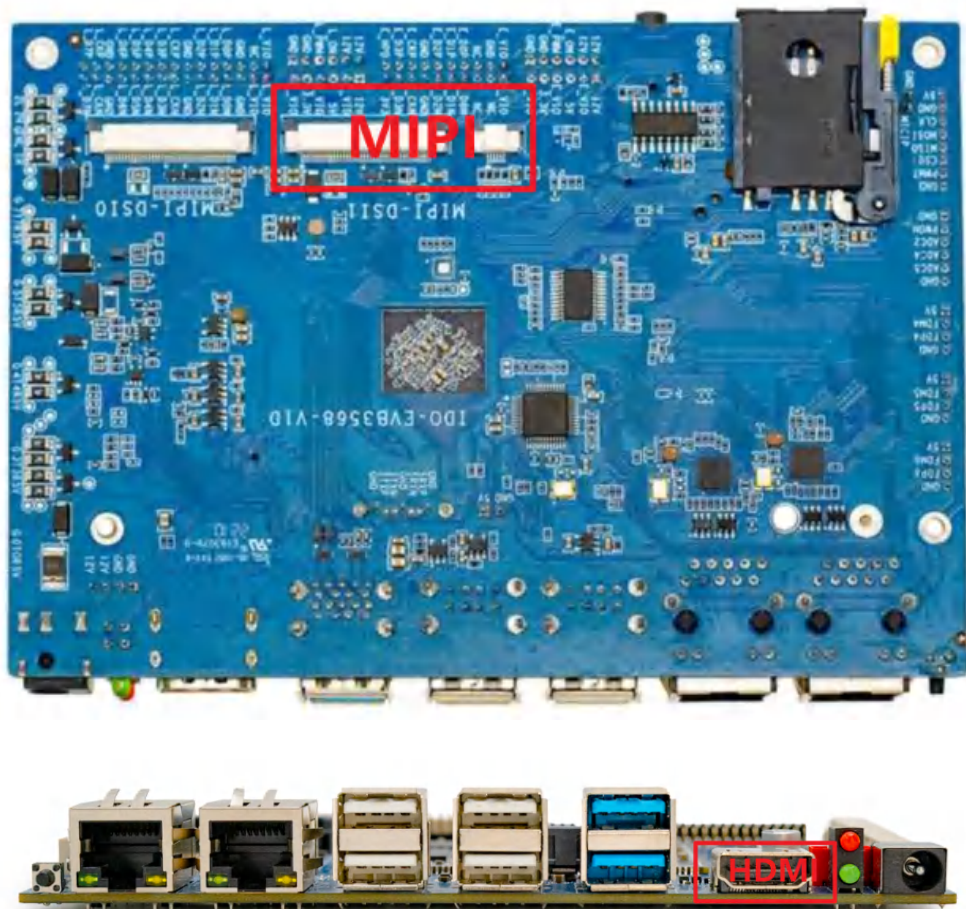
```
1 root@linaro-alip:/# aplay -Dhw:2,0 test.wav
```

## 16 显示屏

### 16.1 显示屏接口说明

主板各个显示接口如下图所示：





## 16.2 显示设置

### 16.2.1 屏幕背光亮度设置

- eDP/MIPI屏背光控制

设备节点: `/sys/class/backlight/backlight/brightness`

设置方法: (支持调节范围 0-255)



Bash

```
1 #关闭
2 echo 0 > /sys/class/backlight/backlight/brightness
3 #最亮
4 echo 255 > /sys/class/backlight/backlight/brightness
```

- Dual LVDS屏幕背光控制

设备节点: `/sys/class/backlight/backlight1/brightness`

设置方法: (支持调节范围 0-255)

```
1 #关闭
2 echo 0 > /sys/class/backlight/backlight1/brightness
3 #最亮
4 echo 255 > /sys/class/backlight/backlight1/brightness
```

## 17. SD卡

将SD卡插入到SD卡槽中，将自动挂载到/mnt/sdcard/目录下。

