

IDO-EVB3568-V2-Debian 系统使用手册

1、调试

1.1 串口调试

1.2 ADB调试

1.3 ssh调试

2、串口

2.1 测试方法

3、USB

3.1 电源控制

4、TF CARD

5、以太网

5.1 查看以太网IP地址

5.1.1 使用命令查看

5.2 设置以太网临时IP地址

5.2.1 使用命令设置

5.2.2 在桌面上设置

5.3 设置以太网永久静态IP

6、WiFi

6.1 连接热点

6.1.1 方式一

6.1.2 方式二

7、蓝牙

7.1 连接蓝牙设备

8、4G

9、音频

9.1 查看声卡设备

9.2 播放音频

9.3 音量的调节

9.4 录音

9.5 音频（内核5.10）

10、摄像头

10.1 测试

11、RTC

方法一

11.1 获取RTC时间

11.2 设置RTC时间

方法二

11.1 获取RTC时间

11.2 设置RTC时间

11.3 RTC定时开机

12、PWM功能

12.1 测试

13、开机自启动

14、屏幕控制

14.1 背光调节

14.2 屏幕旋转

14.2.1 临时旋转

14.2.2 永久旋转

15、按键

16、ADC

16.1 ADC转换方法

16.2 测试

17、网络优先级设置

17.1 查看路由表

17.2 设置默认路由

17.2.1 设置WiFi为默认路由

17.2.2 设置以太网为默认路由

18、CAN

18.1测试

IDO-EVB3568-V2

Debian系统使用手册

深圳触觉智能科技有限公司

www.industio.cn

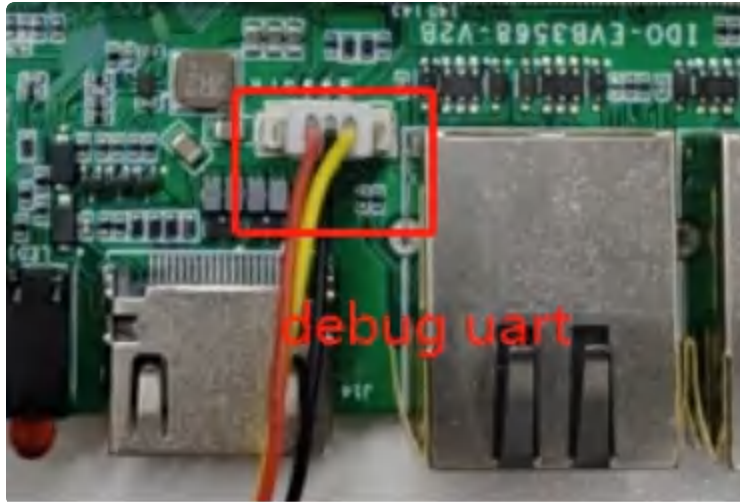
文档修订历史

版本	修订内容	修订	审核	日期
V1.0	创建文档；	谭文学		2022/11/28
V1.1	添加PWM描述和测试方法； 添加ADC功能描述和测试方法； 添加CAN功能描述和测试方法；	刘崇凯		2022/12/09
v1.2	新增 5.10 音频使用方法 新增一种rtc 设置方法			2024/1/5
	问题： rtc定时失败			2024/1/5

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，电平状态为TTL电平。



串口调试默认登录账号密码为 **linaro @ linaro**。

Bash

```
1  Debian GNU/Linux 10 linaro-alip ttyFIQ0
2
3  linaro-alip login: linaro
4  密码:
5  上一次登录: 四 2月 14 10:13:30 UTC 2019ttyFIQ0 上
6  Linux linaro-alip 4.19.219 #57 SMP Fri Nov 4 11:14:32 CST 2022 aarch64
7
8  The programs included with the Debian GNU/Linux system are free software;
9  the exact distribution terms for each program are described in the
10 individual files in /usr/share/doc/*/copyright.
11
12 Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
13 permitted by applicable law.
14 linaro@linaro-alip:~$
15
```

1.2 ADB调试

ADB调试端口位于J5，使用TYPE-C线，连接主板的TYPE-C端口和电脑，即可在电脑上使用adb调试。

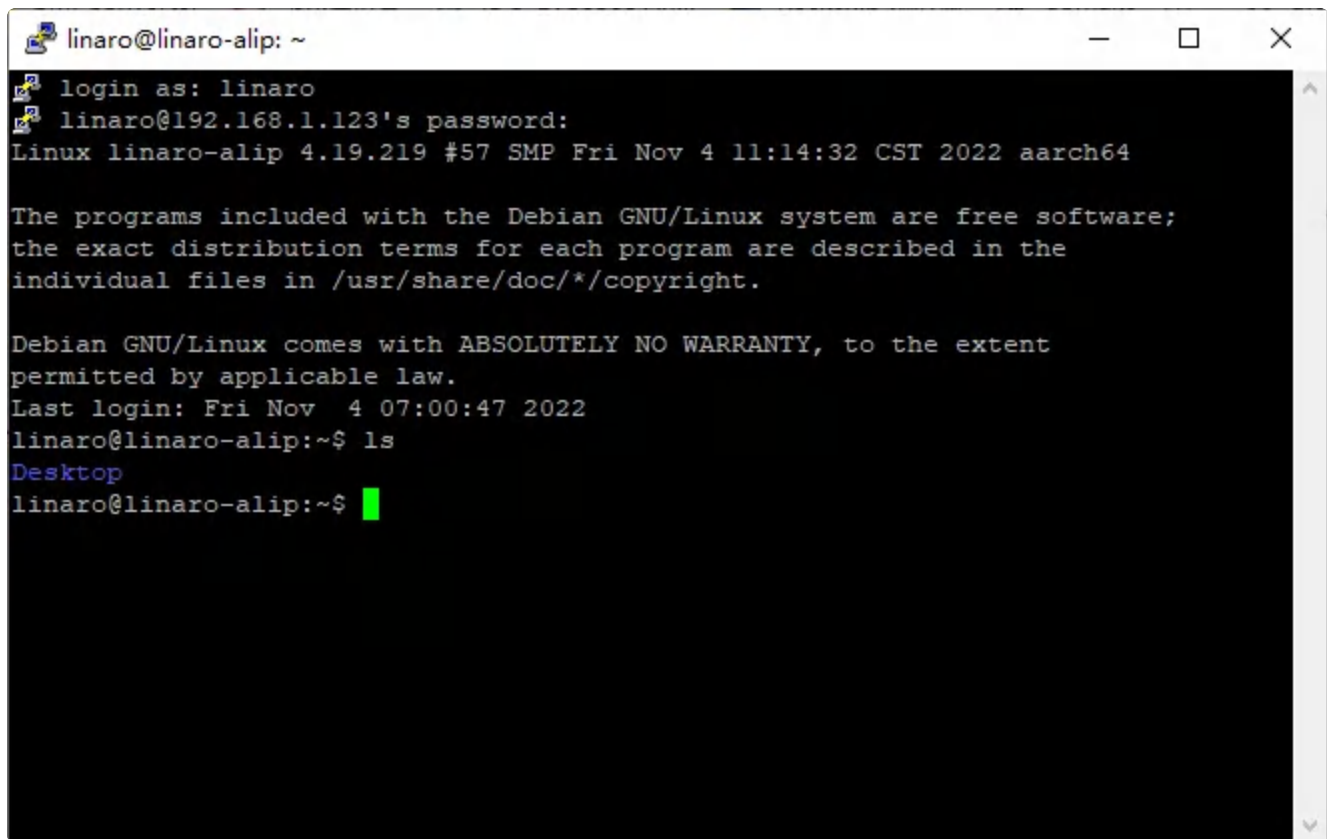


```
C:\Users\ronnie>adb shell
* daemon not running. starting it now on port 5037 *
* daemon started successfully *
root@linaro-alip:/# ls
ls
bin    dev    lib          media  opt    run    srv    tmp    usr
boot  etc    lost+found  mnt    proc   sbin   sys    udisk  var
data  home  md5sum.txt  oem  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

1.3 ssh调试

系统默认登录账号密码为 linaro @ linaro。

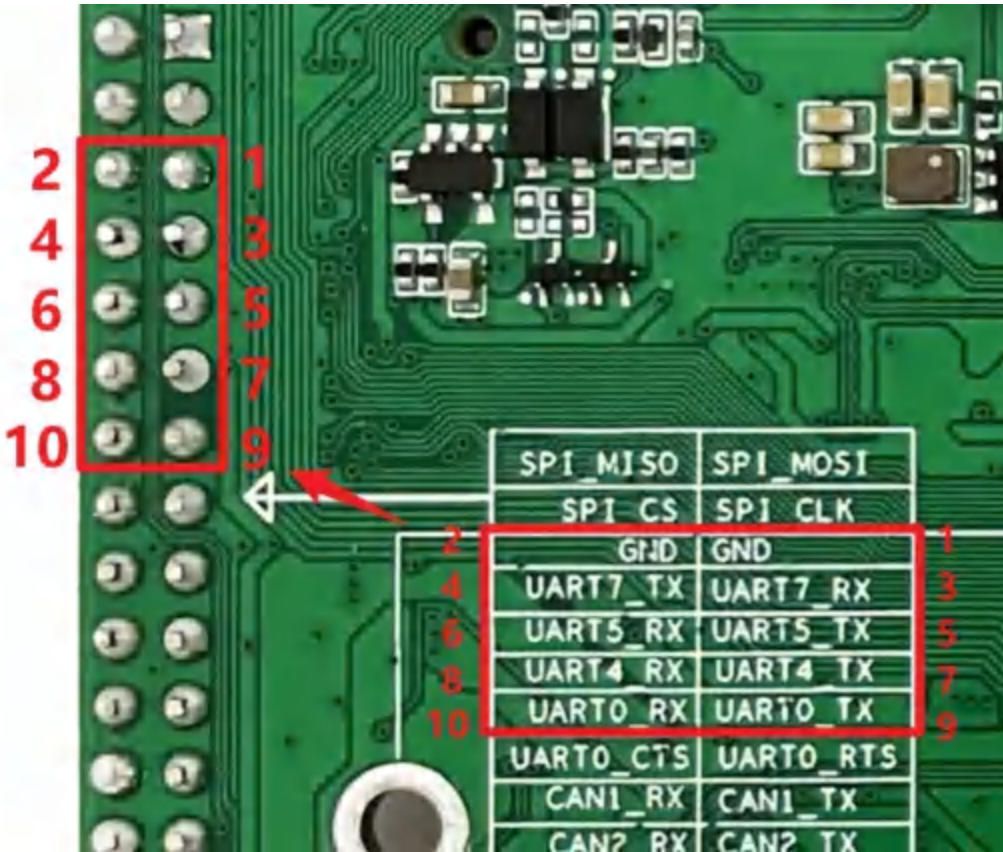
ssh登录需要知道主板的IP，获取IP的方法，请参考第5章以太网的说明。



```
linaro@linaro-alip: ~  
login as: linaro  
linaro@192.168.1.123's password:  
Linux linaro-alip 4.19.219 #57 SMP Fri Nov 4 11:14:32 CST 2022 aarch64  
  
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
Last login: Fri Nov 4 07:00:47 2022  
linaro@linaro-alip:~$ ls  
Desktop  
linaro@linaro-alip:~$
```

2、串口

主板共配置了4路串口（不包括调试串口），其中1路支持流控。



序号	设备节点	默认电平类型	位置	备注
1	/dev/ttyS0	TTL	J24	4线，支持流控
2	/dev/ttyS4	TTL	J24	2线，不支持流控
3	/dev/ttyS5	TTL	J24	2线，不支持流控
4	/dev/ttyS7	TTL	J24	2线，不支持流控

2.1 测试方法

4路使用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.
```

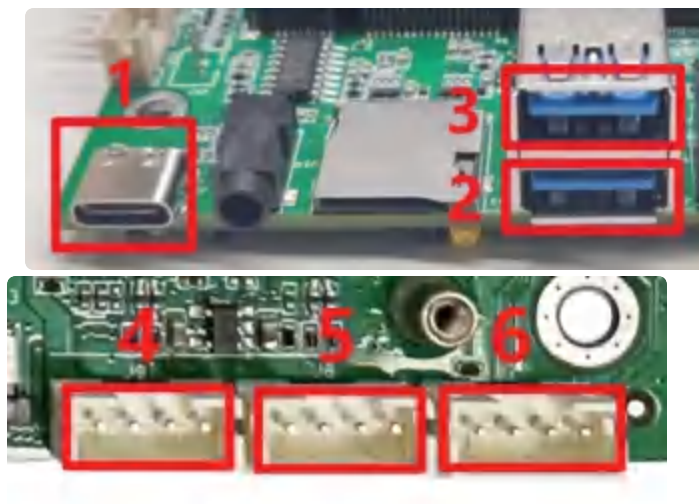
按下键盘任意键会发送对应的字符，而接收的内容会显示在终端。

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、USB

主板共配置了5路USB接口，分别为USB0-5，均为USB-HOST。



名称	编号	类型	位置
1	usb0	USB OTG	J5
2	usb1	host-2.0	J6
3	usb2	host-2.0	J6
4	usb3	host-2.0	J9
5	usb4	host-2.0	J8
6	usb5	host-2.0	J7

USB1默认为device模式，可用于adb调试。如果要切换host模式，执行以下命令：

```

1  linaro@linaro-alip:~$ sudo chmod o+w /dev/otg_mode
2  linaro@linaro-alip:~$
3  linaro@linaro-alip:~$ sudo echo HOST > /dev/otg_mode

```

当要从host切换到device模式，执行以下命令：

```

1  linaro@linaro-alip:~$ sudo echo DEVICE > /dev/otg_mode

```

当USB-HOST插入U盘后，会自动挂载/media/linaro/目录下：

```
▼ Bash |
1  linaro@linaro-alip:~$ ls /media/linaro/
2  KINGSTON
```

3.1 电源控制

默认所有USB-HOST的电源都是开启的，其中USB3-5我们提供了开启/关闭电源的方法。

编号	名称	电源控制节点	位置
4	usb3	/sys/class/leds/usb_fed3_pwr/brightness	J9
5	usb4	/sys/class/leds/usb_fed2_pwr/brightness	J8
6	usb5	/sys/class/leds/usb_fed1_pwr/brightness	J7

打开USB3的电源：

```
▼ Bash |
1  linaro@linaro-alip:~$ sudo echo 255 > /sys/class/leds/usb_fed1_pwr/brightness
   ss
```

关闭USB3的电源：

```
▼ Bash |
1  linaro@linaro-alip:~$ sudo echo 0 > /sys/class/leds/usb_fed1_pwr/brightness
```

USB4-5的电源控制方法类似，如果出现权限不足需要进入超级用户模式再运行上述命令

4、TF CARD

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

```
1  linaro@linaro-alip:~$ ls /media/linaro/  
2  3533-3735  
3  linaro@linaro-alip:~$
```

5、以太网

主板配置了2个1000M以太网接口，对应的网络设备节点为eth0和eth1。



5.1 查看以太网IP地址

5.1.1 使用命令查看

系统默认以太网为动态获取IP，当以太网接口插入网线时，会自动获取IP。

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

5.2 设置以太网临时IP地址

5.2.1 使用命令设置

以设置eth0为例：

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

5.2.2 在桌面上设置

5.3 设置以太网永久静态IP

修改配置文件：/etc/network/interfaces，以设置eth0静态ip 192.168.1.123为例。

```
1  # interfaces(5) file used by ifup(8) and ifdown(8)
2  # Include files from /etc/network/interfaces.d:
3  source-directory /etc/network/interfaces.d
4  auto eth0
5      iface eth0 inet static
6          address 192.168.1.123
7          netmask 255.255.255.0
8          gateway 192.168.1.1
9          nameserver 192.168.1.1
```

然后重启网络

```
1  sudo systemctl restart networking
```

重启网络后，eth0的ip地址已经变成刚才设置的静态ip。

设置静态IP后，断电重启设备依旧生效。

6、WiFi

主板上电默认会打开WiFi，对应的网络设备节点为wlan0。

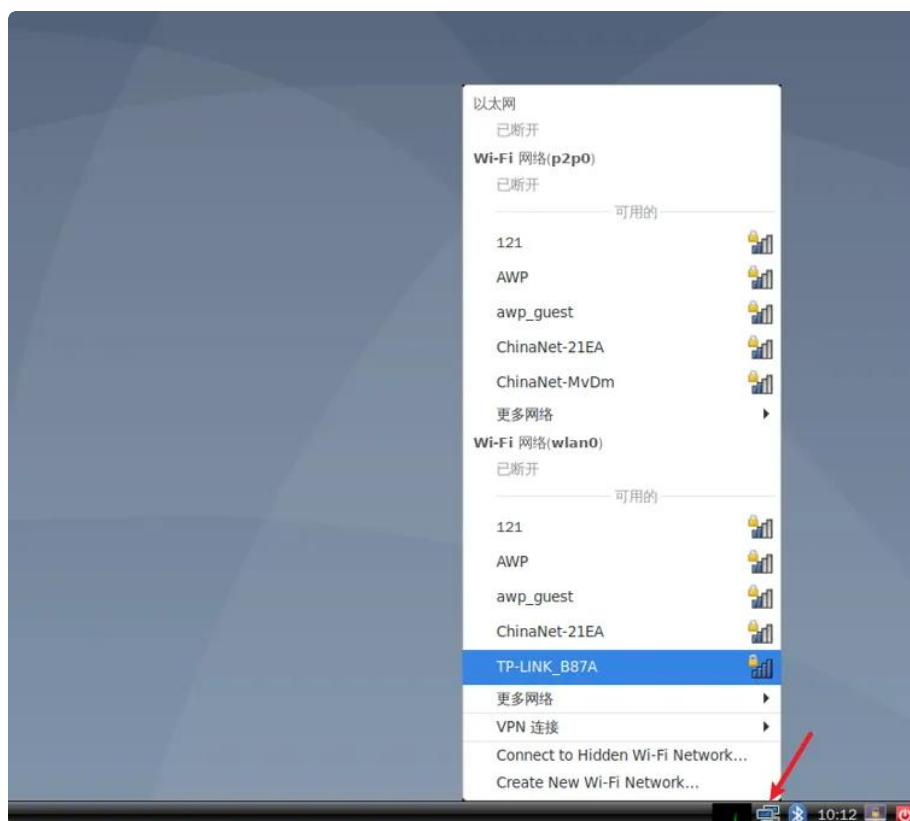
```
1  linaro@linaro-alip:~$ sudo ifconfig wlan0
2  wlan0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
3      ether 2c:d2:6b:10:ea:4d txqueuelen 1000 (Ethernet)
4      RX packets 0 bytes 0 (0.0 B)
5      RX errors 0 dropped 0 overruns 0 frame 0
6      TX packets 0 bytes 0 (0.0 B)
7      TX errors 0 dropped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

6.1 连接热点

连接热点可以在桌面上操作，也可以使用命令行操作。

6.1.1 方式一

点击桌面右下角的网络图标，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：



6.1.2 方式二

修改/etc/network/interfaces

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

新建/etc/wpa.conf

Bash

```
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:

Bash

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

连接成功后使用ifconfig命令可查看wlan0的IP:

Bash

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

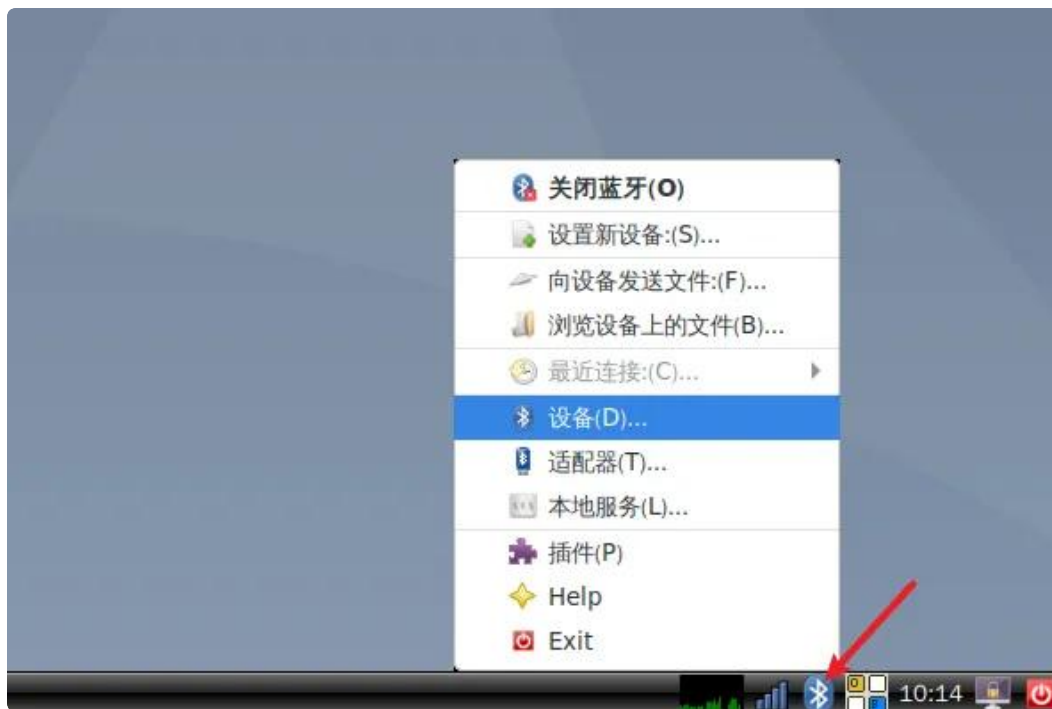
7、蓝牙

主板开机默认打开蓝牙，对应的网络节点为hci0。

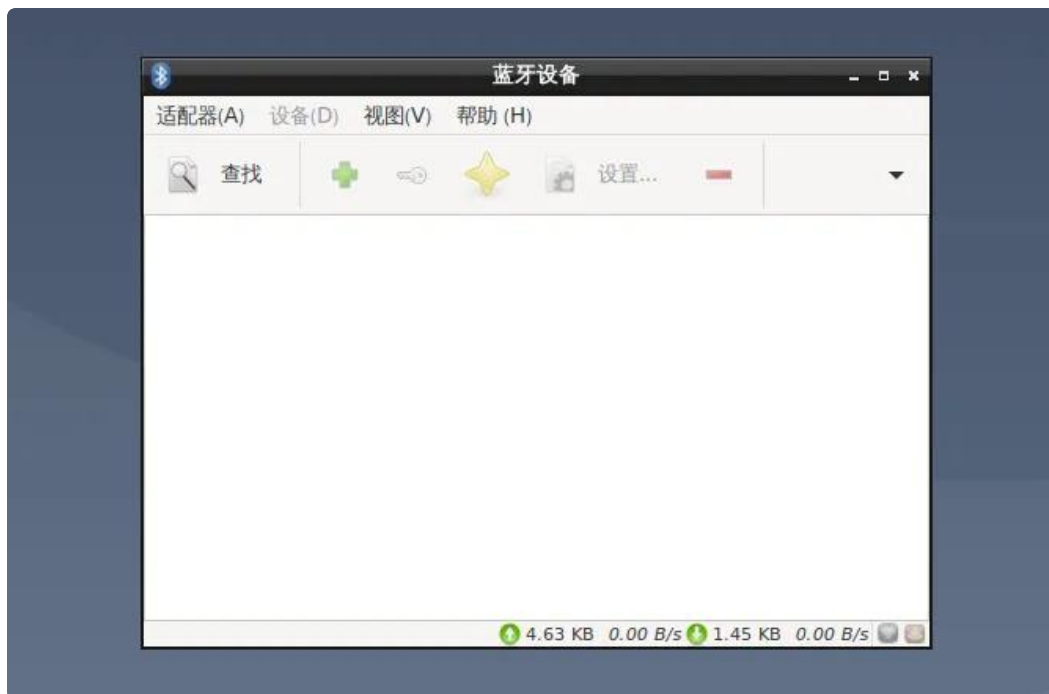
```
1  linaro@linaro-alip:~$ hciconfig
2  hci0:  Type: Primary  Bus: UART
3         BD Address: 2C:D2:6B:11:AC:71  ACL MTU: 1021:8  SCO MTU: 255:12
4         UP RUNNING
5         RX bytes:1531 acl:0 sco:0 events:51 errors:0
6         TX bytes:5012 acl:0 sco:0 commands:51 errors:0
7
8  linaro@linaro-alip:~$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

7.1 连接蓝牙设备

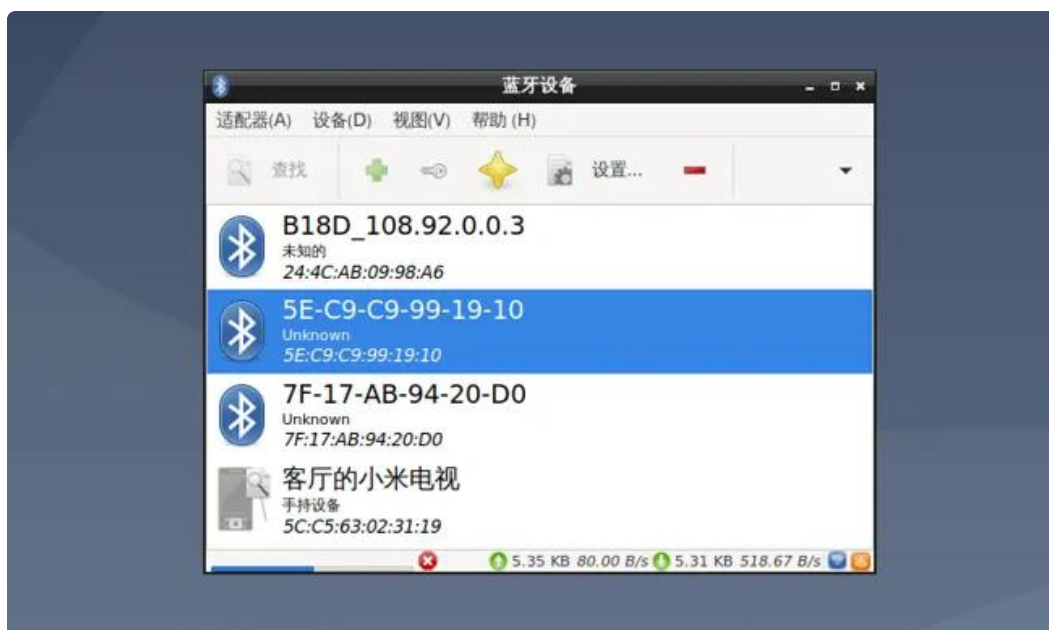
将鼠标放到桌面右下角蓝牙图标，右键->设备：



在弹出的窗口中，点击【查找】：



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：



8、4G

主板默认适配EC20模块（4G），上电前，正确按照模块和SIM卡，上电后，系统会自动进行拨号上网。

序号	模块名称	说明
1	EC20	4G LTE

拨号成功会产生wwan0网络节点：

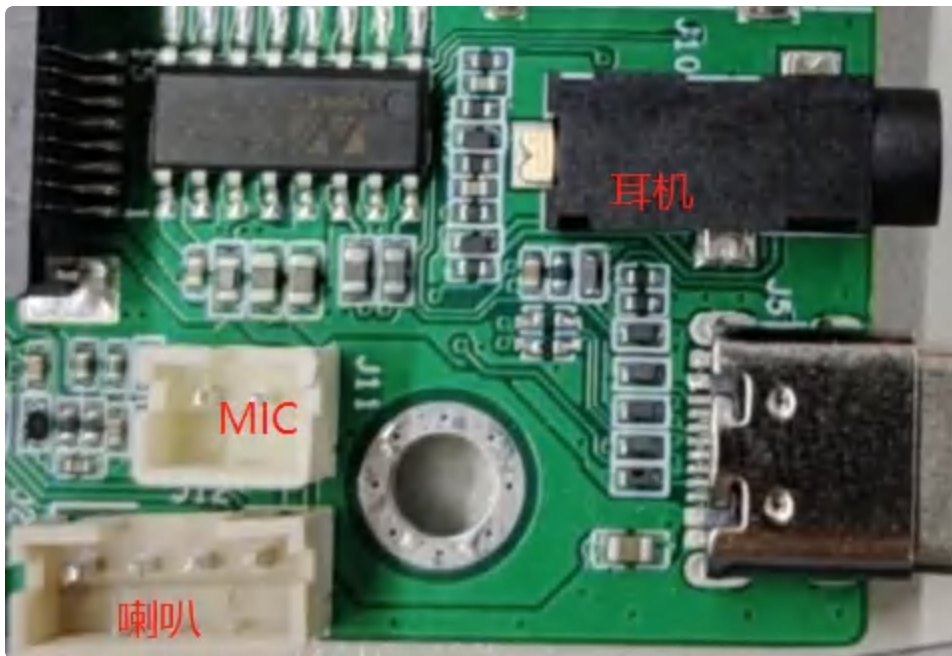
```
linaro@linaro-alip:~$ sudo ifconfig wwan0
wwan0: flags=193<UP,RUNNING,NOARP>  mtu 1500
    inet 10.101.61.51  netmask 255.255.255.248
    inet6 fe80::fc:f6ff:fe8d:bab6  prefixlen 64  scopeid 0x20<link>
    ether 02:fc:f6:8d:ba:b6  txqueuelen 1000  (Ethernet)
    RX packets 42  bytes 7013 (6.8 KiB)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

使用ping命令测试4G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
```

9、音频

喇叭接口位于J12、MIC位于J11、耳机接口位于J10。



9.1 查看声卡设备

```
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:~$
```

9.2 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
```

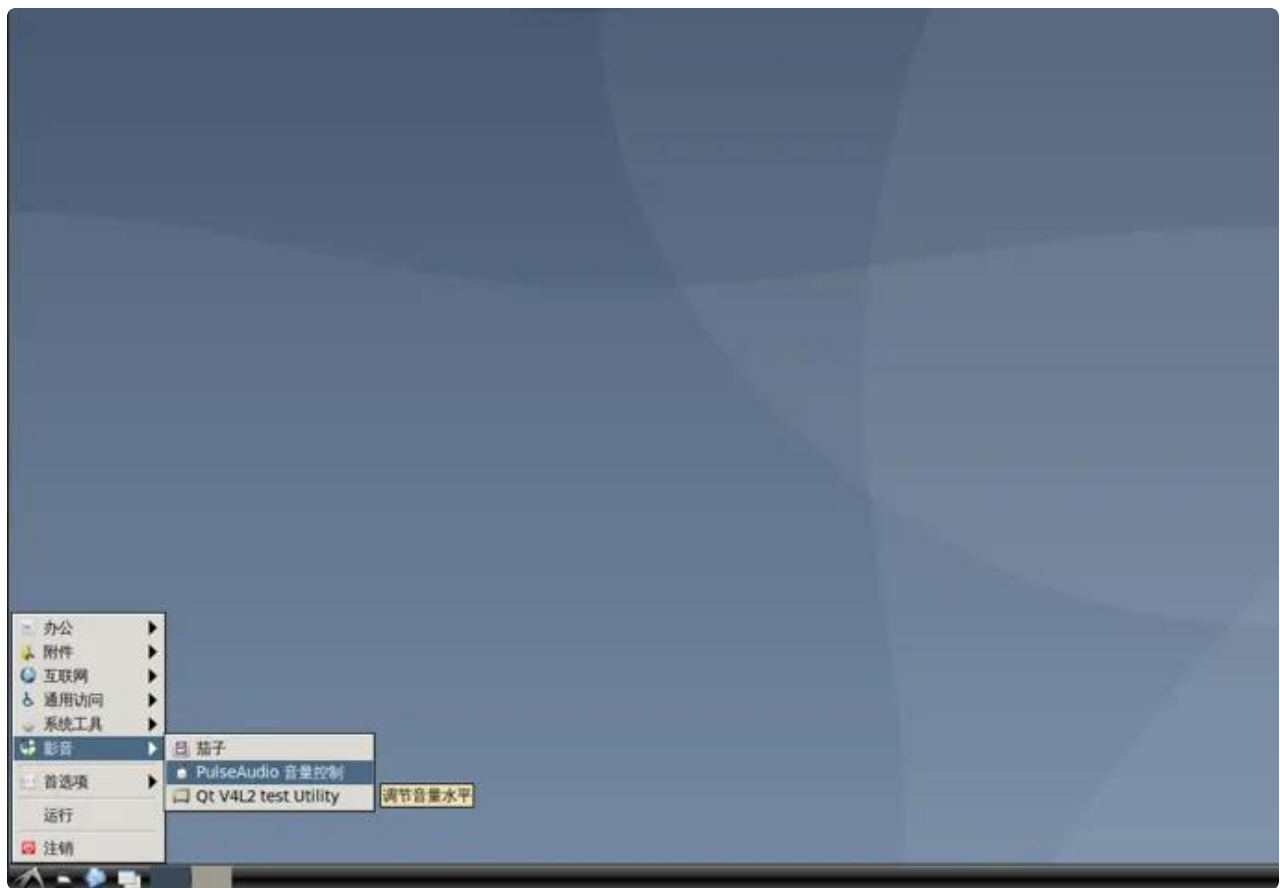
播放到耳机:

插入耳机，执行以下命令。

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

9.3 音量的调节

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：



9.4 录音

将麦克风连接到J11。

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
```

9.5 音频（内核5.10）

播放到HDMI:

```
▼ Bash |
1 root@linaro-alip:/# aplay -D plug:spk_c0 /usr/share/sounds/alsa/Rear_Center.wav
2 Playing WAVE '/usr/share/sounds/alsa/Rear_Center.wav' : Signed 16 bit Little Endian, Rate 48000 Hz, Mono
```

播放到喇叭

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

播放到耳机

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

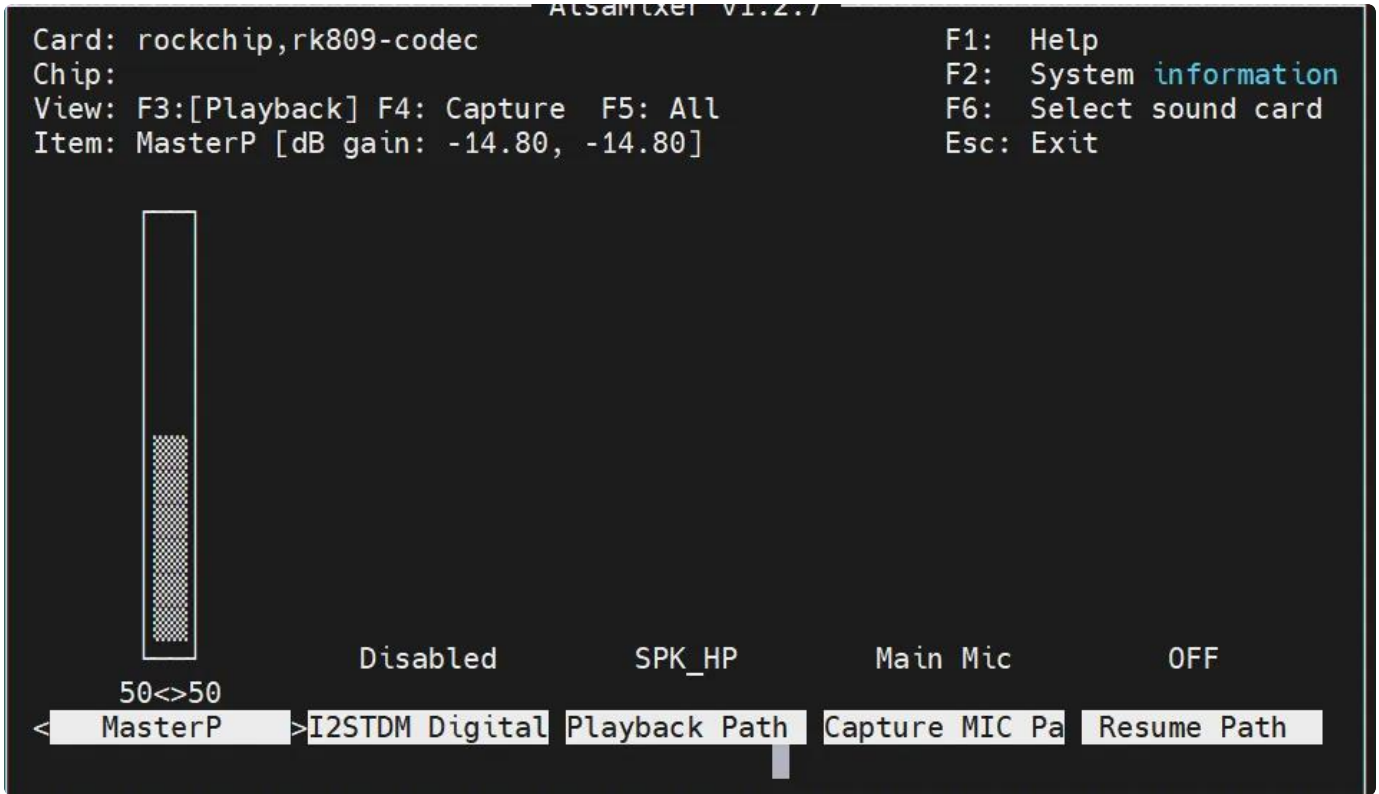
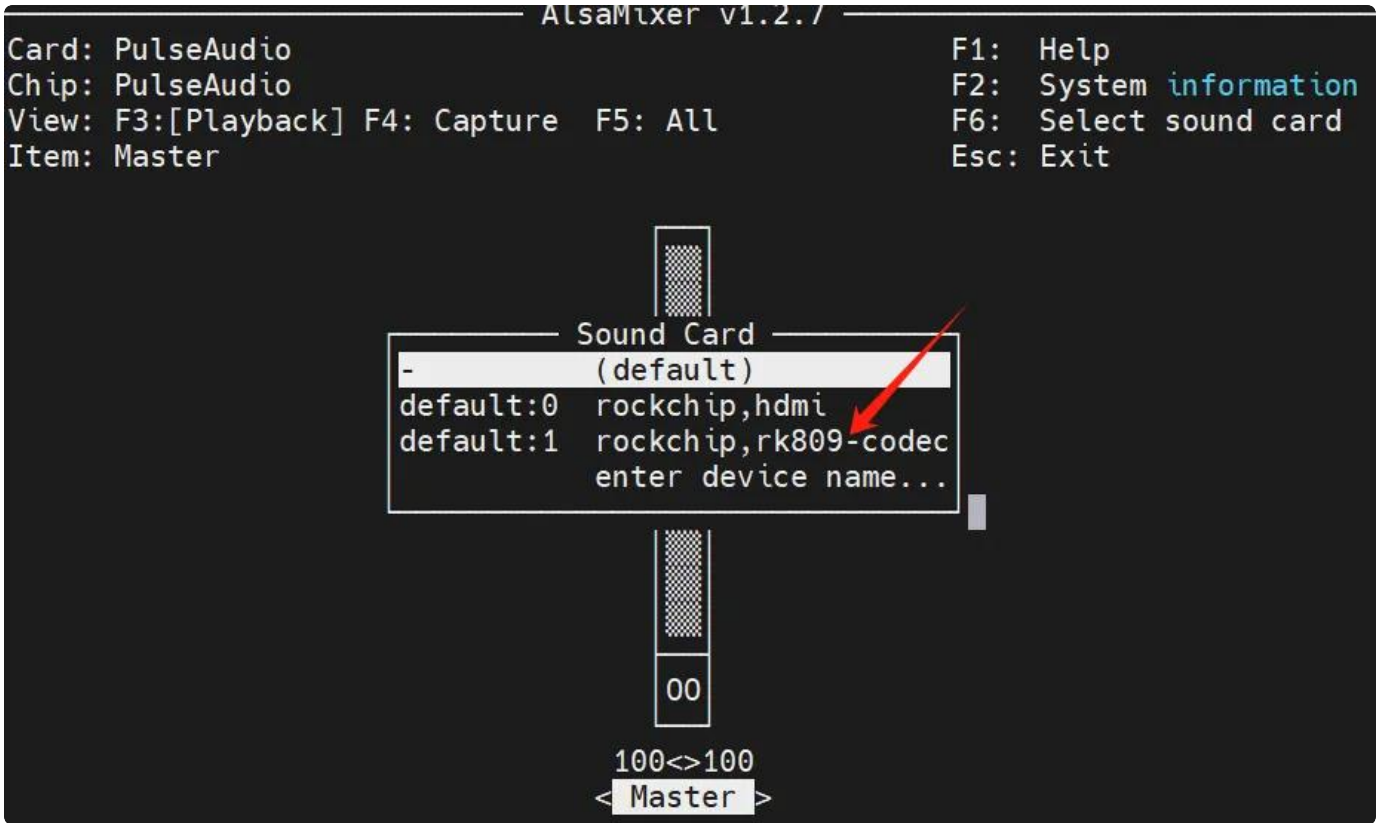
录音

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

```
▼ Bash |
1 root@linaro-alip:/# aplay -D plug:spk_c1 test.wav
2 Playing WAVE 'test.wav' : Signed 16 bit Little Endian, Rate 48000 Hz, Stereo
3 ^CAborted by signal 中断...
4 aplay: pcm_write:2058: write error: 被中断的系统调用
5
```

音量调节

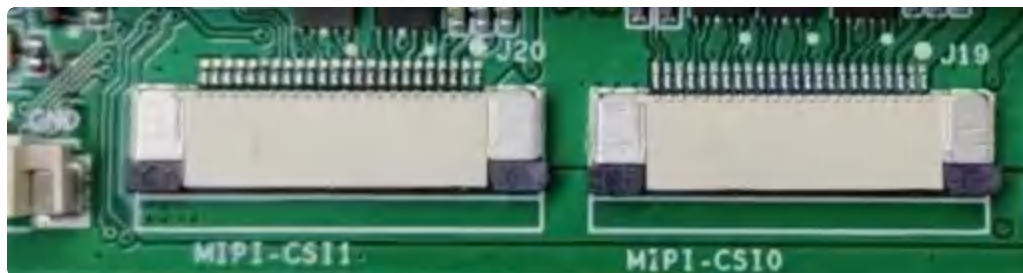
键盘按下"s",选择声卡



左侧柱状图标就是调节音量，其他的不用管即可

10、摄像头

主板默认适配OV5648+OV8858摄像头。

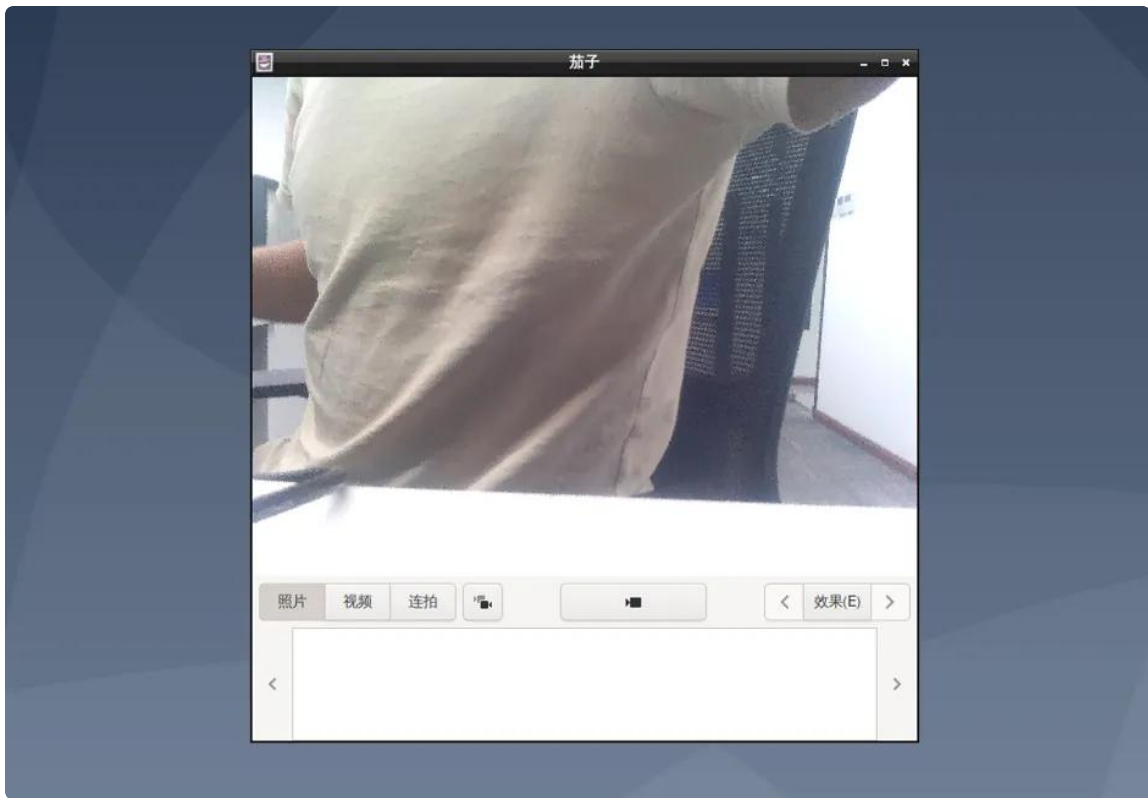


10.1 测试

使用Debian自带的【茄子】程序，可以测试摄像头功能。

依次点击桌面左下角的【影音】->【茄子】即可启动茄子程序测试摄像头功能。





11、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。

方法一

11.1 获取RTC时间

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

11.2 设置RTC时间

▼

Bash

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

方法二

11.1 获取RTC时间

▼

Bash

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

11.2 设置RTC时间

▼

Bash

```
1  root@rk3568-buildroot:/# date -s '2000-01-30 1:1:1'
2  Sun Jan 30 01:01:01 UTC 2000
3  root@rk3568-buildroot:/# hwclock -w -f /dev/rtc1
4  root@rk3568-buildroot:/# hwclock -r -f /dev/rtc1
5  Sun Jan 30 01:01:11 2000  0.000000 seconds
```

断电重新上电，我们可以看到时间又被复原，我们直接

```

1 root@rk3568-buildroot:/# date
2 Fri Aug  4 09:00:17 UTC 2017
3
4 //写入系统时间
5 root@rk3568-buildroot:/# hwclock --hctosys --rtc=/dev/rtc1
6 root@rk3568-buildroot:/# date
7 Sun Jan 30 01:03:58 UTC 2000
8 root@rk3568-buildroot:/# hwclock -r -f /dev/rtc1
9 Sun Jan 30 01:04:17 2000  0.000000 seconds
10

```

11.3 RTC定时开机

CPU内部的rtc1支持支持定时开机功能。

以下测试关机60秒定时开机：

```

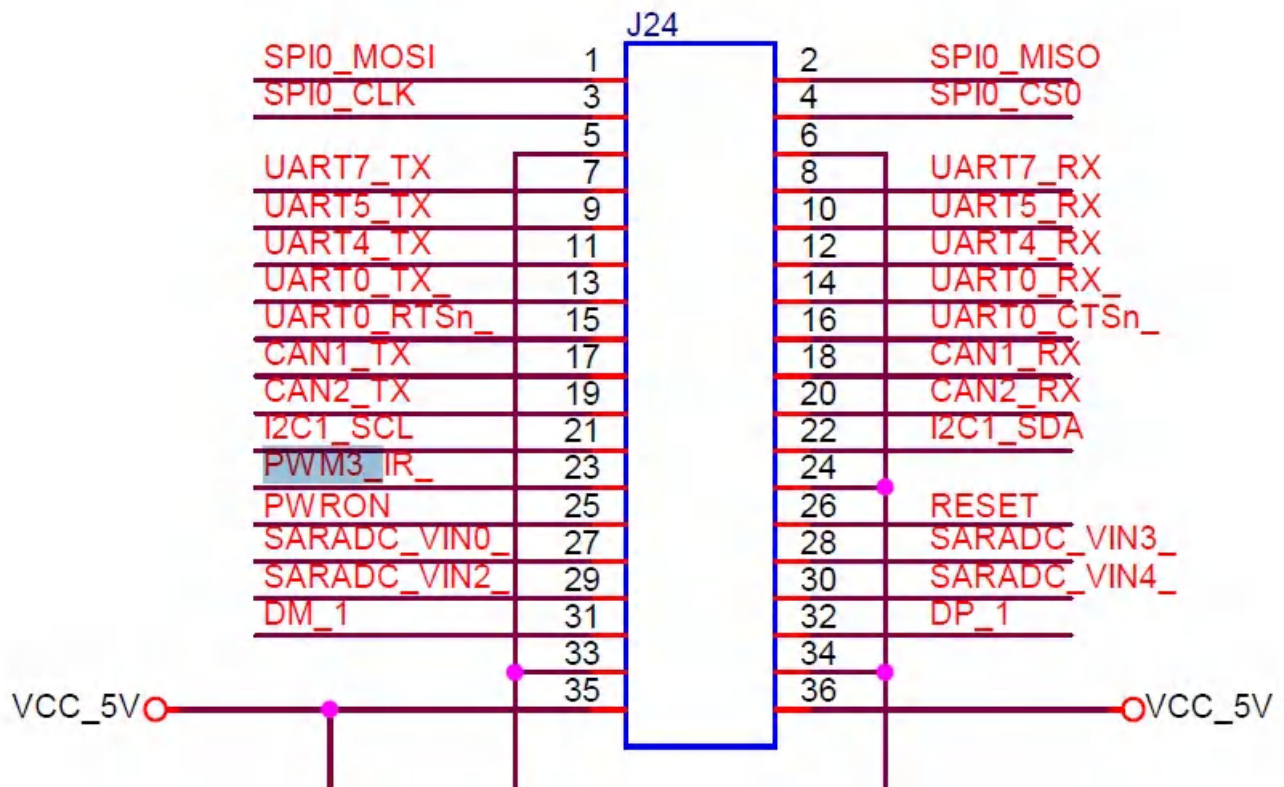
1 linaro@linaro-alip:~$ sudo chmod a+w /sys/class/rtc/rtc1/wakealarm
2 linaro@linaro-alip:~$ sudo echo +60 > /sys/class/rtc/rtc1/wakealarm
3 [ 150.298538] rk809:alarm set RTC date/time 2017-08-04(5) 09:03:38,enabl
   e:1
4 linaro@linaro-alip:~$ sudo poweroff
5     Stopping Setup zram based device zram0...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

```

等待60秒后，主板会开机。

12、PWM功能

PWM位于J24双排针。



《双排针图》

序号	定义	电平	说明
23	PWM3_IR_	3.3V	

12.1 测试

```
cd /sys/class/pwm/pwmchip0/
```

开启PWM:

```
echo 0 > export
```

```
cd pwm0
```

设置周期:

```
echo 330000 > period
```

设置占空比:

```
echo 150000 > duty_cycle
```

使能PWM:

```
echo 1 > enable
```

失能PWM:

```
echo 0 > enable
```

13、开机自启动

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

14、屏幕控制

14.1 背光调节

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

设置亮度为100:

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

14.2 屏幕旋转

使用xrandr工具可以实现屏幕的旋转。

14.2.1 临时旋转

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

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

14.2.2 永久旋转

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

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

经过以上修改，每次启动主板后，桌面会顺时针旋转180度。

15、按键

主板配置了一个ADC按键SW2，对应的设备节点为/dev/input/event0。

系统运行时，短按该按键上报KEY_POWER，并且进入待机状态。

系统待机时，短按该按键，系统恢复正常运行。

系统运行时，长按该按键5秒关机。

系统关机时，短按该按键开机。

16、ADC

主板配置了4路ADC，位于J24的第7.8.9,10引脚，分别记作ADC0、ADC2、ADC3、ADC4。精度为10位。

16.1 ADC转换方法

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

其中raw为对应设备节点读取的值，范围为0–1023。

序号	编号	设备节点
9	SARADC VINU0_	/sys/bus/iio/devices/iio:device0/in_voltage0_raw
7	SARADC VINU2_	/sys/bus/iio/devices/iio:device0/in_voltage2_raw
10	SARADC VINU3_	/sys/bus/iio/devices/iio:device0/in_voltage3_raw
8	SARADC VINU4_	/sys/bus/iio/devices/iio:device0/in_voltage4_raw

16.2 测试

以测试ADC2为例，其余ADC测试方法类似。

```
linaro@linaro-alip:~$ cat /sys/bus/iio/devices/iio:device0/in_voltage0_raw
1023
```

设备节点读取的raw值为1023，代入到公式计算：

$$V=(1023/1024)*1.8v=1.79v$$

即ADC1输入的电压为1.79v。

17、网络优先级设置

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

17.1 查看路由表

▼

Bash |

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

17.2 设置默认路由

17.2.1 设置WiFi为默认路由

当前以太网和WiFi同时使用，设置WiFi优先：

▼

Bash

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

这样默认路由就是wlan0了，即优先使用WiFi进行数据通信。

17.2.2 设置以太网为默认路由

当前以太网和WiFi同时使用，且WiFi优先：

▼

Bash

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

设置为以太网优先：

▼

Bash

```

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

```

其他情况按照类似的方法进行处理即可。

18、CAN

CAN位于J24的双排针。共有3路CAN可供使用。

序号	编号	描述
19	CAN1_TX	CAN1
20	CAN1_RX	
17	CAN2_TX	CAN2
18	CAN2_RX	
15	CAN0_TX (I2C_SCL)	CAN0 当作为CAN0时需要关闭I2C1
16	CAN0_RX (I2C_SDA)	

注意：由于开发板未带有CAN芯片，所以这里需要用到转接板进行测试。

18.1测试

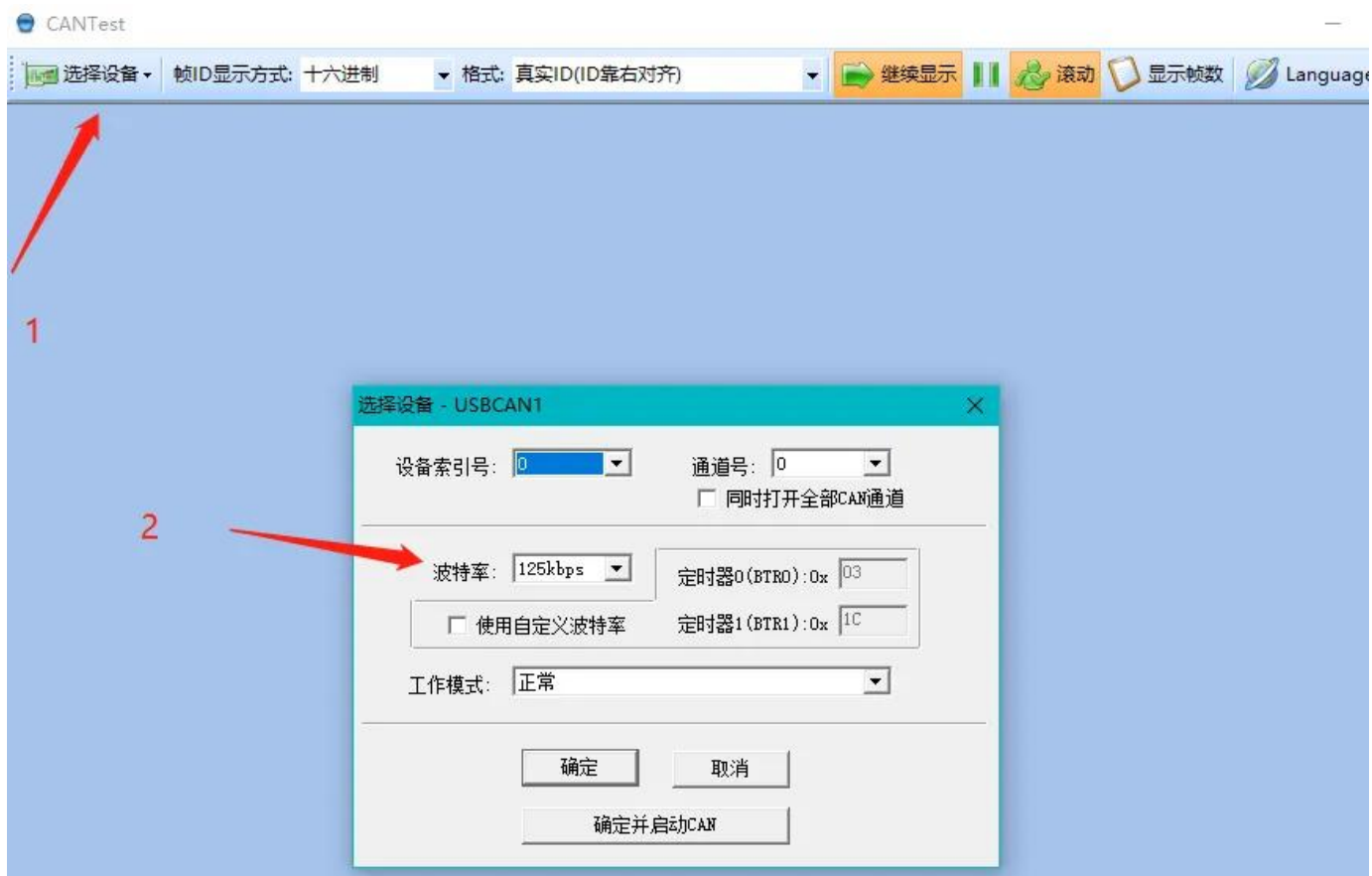
测试需要用USB转CAN工具，通过PC与板上CAN通信。

这里以CAN0为例，其余节点测试方法相同：

```
▼ Bash |
1 //关掉can
2 ifconfig can0 down
3
4 //配置can通信的波特率
5 ip link set can0 type can bitrate 125000 triple-sampling on
6
7 //开启can通信
8 ifconfig can0 up
9
10 //作为接收端接收数据
11 candump can0
12
13 //作为发送端发送数据
14 cansend can0 5A1#1122334455667788
15
```

PC软件端的接收与发送：

(1) 选择USBCAN1



(2) 启动CAN测试

The screenshot displays the CANTest application window. The top menu bar includes options like '选择设备' (Select Device), '帧ID显示方式' (Frame ID Display Method), and '格式' (Format). Below the menu bar, there's a toolbar with icons for '滤波设置' (Filter Settings), '启动' (Start), '停止' (Stop), '关闭' (Close), '定位' (Position), '清空' (Clear), '保存' (Save), '设备操作' (Device Operation), '接收时间标识' (Receive Time Identifier), '隐藏发送帧' (Hide Transmit Frames), '显示发送帧' (Show Transmit Frames), 'DBC', and '实时保存' (Real-time Save).

The main area shows a table of received CAN frames. A red arrow points to the '停止' (Stop) button in the toolbar. The table has columns for '序号' (Serial Number), '传输方向' (Transmission Direction), '时间标识' (Time Identifier), '帧ID' (Frame ID), '帧格式' (Frame Format), '帧类型' (Frame Type), '数据长度' (Data Length), and '数据(HEX)' (Data (HEX)).

序号	传输方向	时间标识	帧ID	帧格式	帧类型	数据长度	数据(HEX)
00000073	接收	17:54:33.4...	0x000005a1	数据帧	标准帧	0x08	11 22 33 44 55 66 77 88
00000074	接收	17:54:33.7...	0x000005a1	数据帧	标准帧	0x08	11 22 33 44 55 66 77 88
00000075	接收	17:54:34.1...	0x000005a1	数据帧	标准帧	0x08	11 22 33 44 55 66 77 88
00000076	接收	17:54:34.5...	0x000005a1	数据帧	标准帧	0x08	11 22 33 44 55 66 77 88
00000077	发送	19:15:27.3...	0x00000000	数据帧	标准帧	0x08	00 01 02 03 04 05 06 07
00000078	发送	19:15:27.8...	0x00000000	数据帧	标准帧	0x08	00 01 02 03 04 05 06 07
00000079	发送	19:15:28.0...	0x00000000	数据帧	标准帧	0x08	00 01 02 03 04 05 06 07
00000080	发送	19:15:28.2...	0x00000000	数据帧	标准帧	0x08	00 01 02 03 04 05 06 07
00000081	发送	19:15:28.4...	0x00000000	数据帧	标准帧	0x08	00 01 02 03 04 05 06 07
00000082	接收	19:15:34.1...	0x000005a1	数据帧	标准帧	0x08	11 22 33 44 55 66 77 88
00000083	接收	19:15:34.4...	0x000005a1	数据帧	标准帧	0x08	11 22 33 44 55 66 77 88
00000084	接收	19:15:35.2...	0x000005a1	数据帧	标准帧	0x08	11 22 33 44 55 66 77 88
00000085	接收	19:15:35.7...	0x000005a1	数据帧	标准帧	0x08	11 22 33 44 55 66 77 88
00000086	发送	19:16:07.6...	0x00000000	数据帧	标准帧	0x08	00 01 02 03 04 05 06 07
00000087	发送	19:16:08.6...	0x00000000	数据帧	标准帧	0x08	00 01 02 03 04 05 06 07
00000088	发送	19:16:08.9...	0x00000000	数据帧	标准帧	0x08	00 01 02 03 04 05 06 07
00000089	发送	19:16:09.5...	0x00000000	数据帧	标准帧	0x08	00 01 02 03 04 05 06 07
00000090	发送	19:16:16.0...	0x00000000	数据帧	标准帧	0x08	00 01 02 03 04 05 06 07
00000091	发送	19:16:16.2...	0x00000000	数据帧	标准帧	0x08	00 01 02 03 04 05 06 07
00000092	发送	19:16:16.4...	0x00000000	数据帧	标准帧	0x08	00 01 02 03 04 05 06 07
00000093	发送	19:16:16.6...	0x00000000	数据帧	标准帧	0x08	00 01 02 03 04 05 06 07
00000094	发送	19:16:16.8...	0x00000000	数据帧	标准帧	0x08	00 01 02 03 04 05 06 07
00000095	发送	19:16:17.0...	0x00000000	数据帧	标准帧	0x08	00 01 02 03 04 05 06 07
00000096	发送	19:16:40.3...	0x00000000	数据帧	标准帧	0x08	00 01 02 03 04 05 06 07
00000097	发送	19:16:41.0...	0x00000000	数据帧	标准帧	0x08	00 01 02 03 04 05 06 07
00000098	发送	19:16:41.2...	0x00000000	数据帧	标准帧	0x08	00 01 02 03 04 05 06 07
00000099	发送	19:16:41.7...	0x00000000	数据帧	标准帧	0x08	00 01 02 03 04 05 06 07
00000100	发送	19:16:47.5...	0x00000000	数据帧	标准帧	0x08	00 01 02 03 04 05 06 07
00000101	发送	19:16:47.7...	0x00000000	数据帧	标准帧	0x08	00 01 02 03 04 05 06 07
00000102	发送	19:16:47.9...	0x00000000	数据帧	标准帧	0x08	00 01 02 03 04 05 06 07
00000103	发送	19:16:48.1...	0x00000000	数据帧	标准帧	0x08	00 01 02 03 04 05 06 07
00000104	发送	19:16:48.3...	0x00000000	数据帧	标准帧	0x08	00 01 02 03 04 05 06 07
00000105	接收	19:16:54.9...	0x000005a1	数据帧	标准帧	0x08	11 22 33 44 55 66 77 88
00000106	接收	19:16:55.4...	0x000005a1	数据帧	标准帧	0x08	11 22 33 44 55 66 77 88
00000107	接收	19:16:56.1...	0x000005a1	数据帧	标准帧	0x08	11 22 33 44 55 66 77 88
00000108	接收	19:16:56.7...	0x000005a1	数据帧	标准帧	0x08	11 22 33 44 55 66 77 88

At the bottom, there's a '基本操作' (Basic Operation) section with controls for sending frames. It includes a '发送方式' (Send Method) dropdown set to '正常发送' (Normal Send), radio buttons for '每次发送单帧' (Send single frame each time) and '每次发送 10 帧' (Send 10 frames each time), and a checkbox for '帧ID每发送一帧递增' (Frame ID increments by one frame sent). Below these are fields for '帧类型' (Frame Type) set to '标准帧' (Standard Frame), '帧ID(HEX)' (Frame ID (HEX)) set to '00000000', '数据(HEX)' (Data (HEX)) set to '00 01 02 03 04 05 06 07', '发送次数' (Send count) set to '1', and '每次发送间隔(ms)' (Send interval (ms)) set to '0'. There are '发送' (Send) and '停止' (Stop) buttons. A red arrow points to the '发送' button.

板端发送过来的数据可以在CANtest上打印出来。

