IDO-EVB3568-V2 - Buildroot系统使用说明

Industio 触觉智能

IDO-EVB3568-V2

Buildroot 系统使用说明

深圳触觉智能科技有限公司

www.industio.cn

文档修订历史

版本	修订内容	修订	审核	日期
V1.0	创建文档	何伟聪		2024/1/4
	问题： 1. otg 切换到host后无法切换回device模式 2. j20 底座，摄像头接入，无节点			

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。

串口调试默认无登录账号密码为：

```Bash
root@rk3568-buildroot:/# ls
S36load_wifi_modules  etc      lost+found  opt           sbin    udisk
bin                   info     media       proc          sdcard  userdata
busybox.fragment      lib      misc        rockchip-test sys     usr
data                  lib64    mnt         root          system  var
dev                   linuxrc  oem         run           tmp     vendor
```

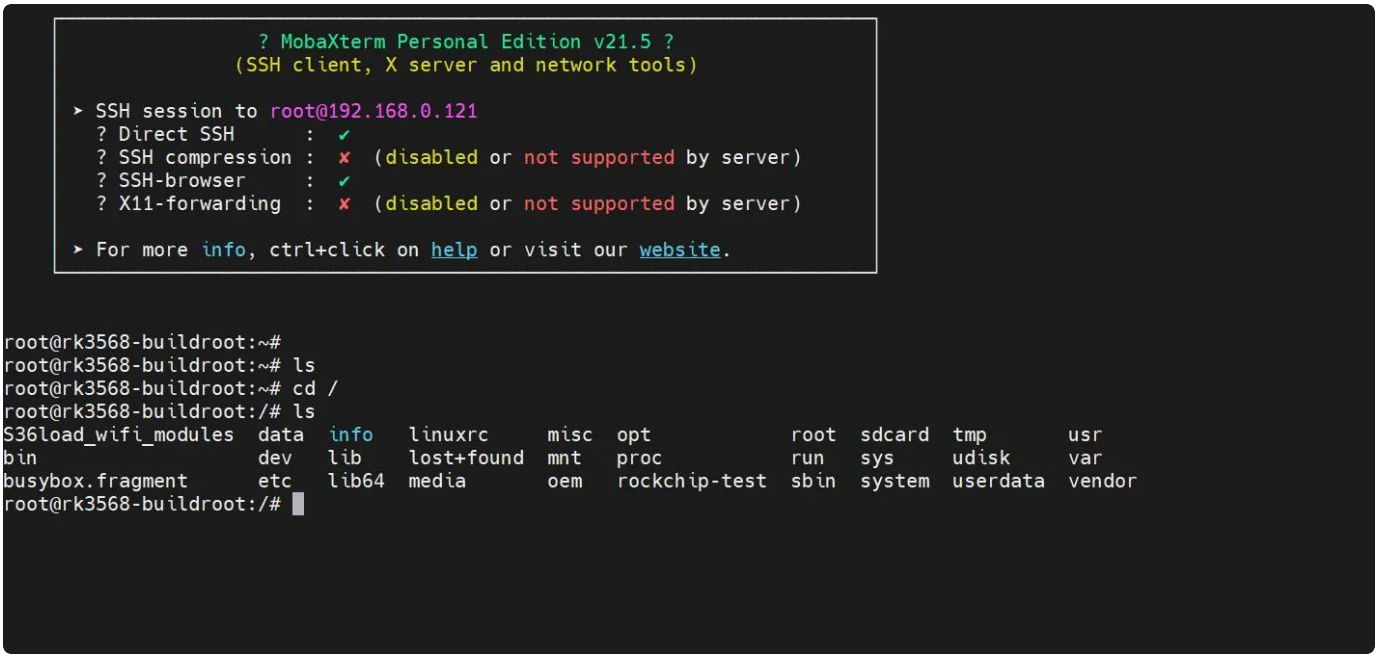
1.2 ADB调试

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。

```Bash
C:\Users\industio_1>adb shell
root@rk3568-buildroot:/# ls
S36load_wifi_modules  etc      lost+found  opt            sbin    udisk
bin                   info     media       proc           sdcard  userdata
busybox.fragment      lib      misc        rockchip-test  sys     usr
data                  lib64    mnt         root           system  var
dev                   linuxrc  oem         run            tmp     vendor
root@rk3568-buildroot:/#
```

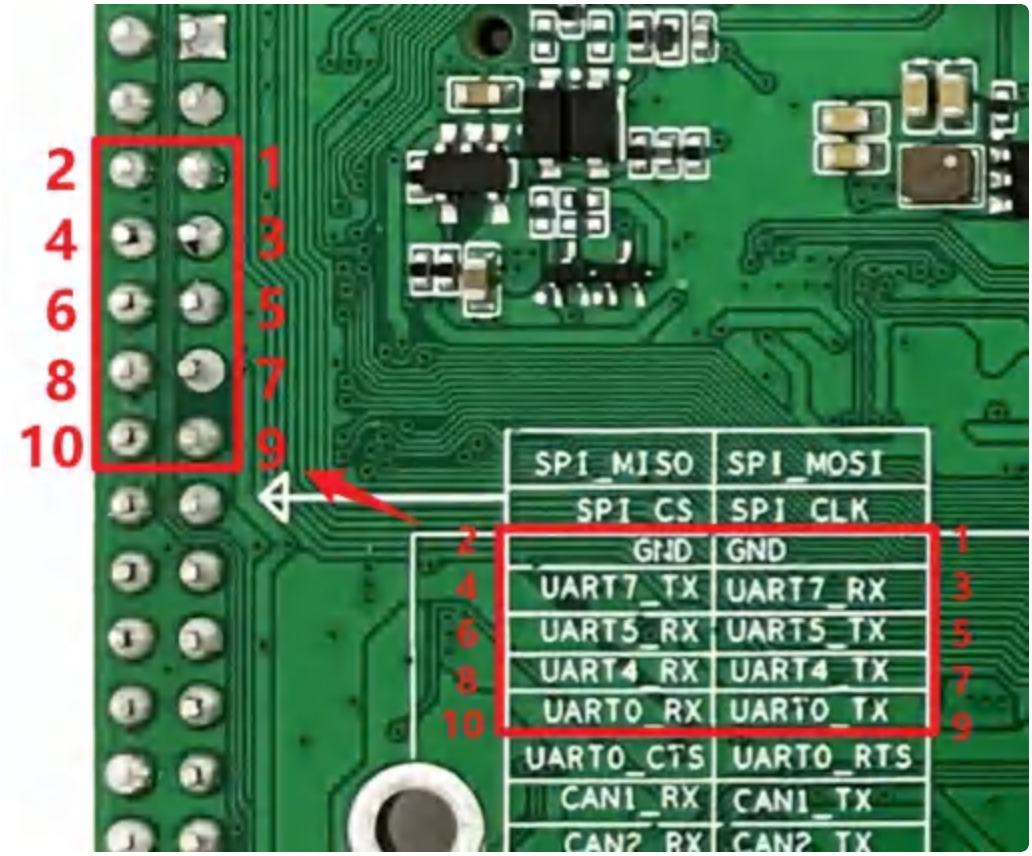
1.3 ssh调试

系统默认ssh账号和密码为 **root@ rockchip**。



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。

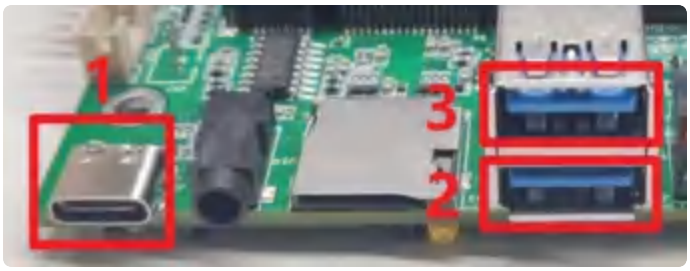
以测试/dev/ttyS0为例：

```
root@rk3568-buildroot:/# microcom  -s 115200 -X /dev/ttyS
ttyS0  ttyS4  ttyS5  ttyS7  ttyS8
root@rk3568-buildroot:/# microcom  -s 115200 -X /dev/ttyS0
[ 9094.238336] of_dma_request_slave_channel: dma-names property of node '/serial@fdd50000' missing or empty
[ 9094.238476] dw-apb-uart fdd50000.serial: failed to request DMA, use interrupt mode
22222222
22222222
22222222
22222222
22222222
22222222
```

buildroot系统如果使用了microcom无法使用ctr+\退出，只能上下电重启

3、USB

主板共配置了5路USB接口，分别为USB0-5，均为USB-HOST。



编号	名称	类型	位置
1	usb0	USB OTG	J5
2	usb1	host-2.0	J6
3	usb2	host-3.0	J6

4	usb3	host-2.0	J9
5	usb4	host-2.0	J8
6	usb5	host-2.0	J7

USB OTG 支持host 和device 模式的切换，软件切换方法如下

```Bash
## host
echo host > /sys/devices/platform/fe8a0000.usb2-phy/otg_mode
## device
echo peripheral > /sys/devices/platform/fe8a0000.usb2-phy/otg_mode
```

当USB-HOST插入U盘后，会自动挂载/media/ido/目录下：

```Bash
root@rk3568-buildroot:/# ls /media/ido/
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。

编写	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
root@rk3568-buildroot:/# echo 255 > /sys/class/leds/usb_fed1_pwr/brightness
```

关闭USB3的电源：

```Bash
root@rk3568-buildroot:/# echo 0 > /sys/class/leds/usb_fed1_pwr/brightness
```

USB4-5的电源控制方法类似。

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/ido/目录下。

```Bash
root@rk3568-buildroot:/# [  494.235146] mmc_host mmc1: Bus speed (slot 0) = 375000Hz (slot req 400000Hz, actual 375000HZ div = 0)
[  494.400868] mmc_host mmc1: Bus speed (slot 0) = 50000000Hz (slot req 50000000Hz, actual 50000000HZ div = 0)
[  494.401109] mmc1: new high speed SDHC card at address e624
[  494.403724] mmcblk1: mmc1:e624 SU04G 3.69 GiB
[  494.420352]  mmcblk1: p1
[  494.870755] FAT-fs (mmcblk1p1): utf8 is not a recommended IO charset for FAT filesystems, filesystem will be case sensitive!
[  494.876127] FAT-fs (mmcblk1p1): Volume was not properly unmounted. Some data may be corrupt. Please run fsck.

root@rk3568-buildroot:/#
root@rk3568-buildroot:/# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/root       1.1G  608M  421M  60% /
devtmpfs        1.9G  8.0K  1.9G   1% /dev
tmpfs           2.0G  252K  2.0G   1% /tmp
tmpfs           2.0G  312K  2.0G   1% /run
tmpfs           2.0G     0  2.0G   0% /dev/shm
/dev/mmcblk0p7  121M   12M  101M  11% /oem
/dev/mmcblk0p8   23G  240K   22G   1% /userdata
/dev/mmcblk1p1  3.7G  1.9M  3.7G   1% /mnt/sdcard

```

5、以太网

5.1 查看以太网IP地址

5.1.1 使用命令查看

系统默认以太网为动态获取IP，当以太网接口插入网线时，会自动获取IP。

```Bash
root@rk3568-buildroot:/# ifconfig  eth0
eth0      Link encap:Ethernet  HWaddr 42:DC:16:A0:6C:1F
          inet addr:192.168.0.121  Bcast:0.0.0.0  Mask:255.255.255.0
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
          Interrupt:49

```

5.2 设置IP地址

5.2.1 临时设置IP

```Plain Text
ifconfig eth1 192.168.1.123
```

5.2.2 设置静态IP

```Bash
vi /etc/network/interfaces
------------------------------------------------------------
# interface file auto-generated by buildroot
auto lo
iface lo inet loopback
auto eth0
iface eth0 inet static
address 192.168.0.121
netmask 255.255.255.0
gateway 192.168.0.1
dns-nameservers 114.114.114.114
```

6、WiFi

系统上电默认会打开WiFi，对应的网络设备节点为**wlan0**。

```Bash
root@rk3568-buildroot:/# ifconfig wlan0
wlan0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether 2c:d2:6b:10:ea:4d  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@rk3568-buildroot:/#
```

6.1 连接热点

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

```Bash
[root@RK356X:/]# cat /userdata/cfg/wpa_supplicant.conf
ctrl_interface=/var/run/wpa_supplicant
ap_scan=1
update_config=1

network={
        ssid="TP-LINK_B87A"
        psk="12345678"
        key_mgmt=WPA-PSK
}
[root@RK356X:/]#
```

执行命令，连接wifi

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

测试

```Bash
root@rk3568-buildroot:/# ifconfig wlan0
wlan0     Link encap:Ethernet  HWaddr 10:BB:F3:15:5A:25
          inet addr:192.168.1.113  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::723e:b72c:ccf:3282/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:198 errors:0 dropped:0 overruns:0 frame:0
          TX packets:110 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:25864 (25.2 KiB)  TX bytes:12986 (12.6 KiB)


root@rk3568-buildroot:/# ping baidu.com -I wlan0
64 bytes from 110.242.68.66 (110.242.68.66): icmp_seq=1 ttl=49 time=47.2 m
s
64 bytes from 110.242.68.66 (110.242.68.66): icmp_seq=2 ttl=49 time=48.2 m
s
64 bytes from 110.242.68.66 (110.242.68.66): icmp_seq=3 ttl=49 time=47.3 m
s

```

7、连接蓝牙设备

设备节点为hci0，我们已经做了开机自启动开启蓝牙功能

蓝牙功能开启后，将产生hci0节点

```Bash
root@rk3568-buildroot:/# hciconfig
hci0:   Type: Primary  Bus: UART
        BD Address: 10:BB:F3:15:CF:4B  ACL MTU: 1021:8  SCO MTU: 255:12
        UP RUNNING
        RX bytes:1486 acl:0 sco:0 events:49 errors:0
        TX bytes:3990 acl:0 sco:0 commands:49 errors:0
```

7.1 扫描设备

使用hcitool工具，扫描蓝牙设备

```Bash
root@rk3568-buildroot:/# hcitool -i hci0 scan
Scanning ...

        BC:64:D9:BF:92:84       Ace 2
        AC:D6:18:5C:0D:4D       cainiaocl
```

8、4G

主板默认适配EC20模块（4G）和 RG200U-CN（5G），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
2	RG200U-CN	支持 5G NSA 和 SA 模式，支持 TDD 和 FDD 两种模式，向下兼容 4G/3G。

拨号成功会产生wwan0网络节点：

```Bash

root@rk3568-buildroot:/# ifconfig wwan0
wwan0     Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          inet addr:10.2.155.147  P-t-P:10.2.155.147  Mask:255.255.255.248
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:45 errors:0 dropped:0 overruns:0 frame:0
          TX packets:84 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:4829 (4.7 KiB)  TX bytes:9410 (9.1 KiB)

```

使用ping命令测试4G/5G上网功能是否正常：

```Bash
root@rk3568-buildroot:/# ping baidu.com -I wwan0
PING baidu.com (110.242.68.66) from 10.2.155.147 wwan0: 56(84) bytes of data.
64 bytes from 110.242.68.66 (110.242.68.66): icmp_seq=1 ttl=49 time=107 ms
64 bytes from 110.242.68.66 (110.242.68.66): icmp_seq=2 ttl=49 time=79.4 ms
```

9、音频

9.1 查看声卡设备

```Bash
root@rk3568-buildroot:/# aplay -l
**** List of PLAYBACK Hardware Devices ****
card 0: rockchiphdmi [rockchip,hdmi], device 0: fe400000.i2s-i2s-hifi i2s-hifi-0 [fe400000.i2s-i2s-hifi i2s-hifi-0]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
card 1: rockchiprk809co [rockchip,rk809-codec], device 0: fe410000.i2s-rk817-hifi rk817-hifi-0 [fe410000.i2s-rk817-hifi rk817-hifi-0]
  Subdevices: 1/1
  Subdevice #0: subdevice #0

```

9.2 播放音频

注意：如果接入的是其他显示器只有一个声卡，这时候spk_c0就是播放到耳机或者喇叭上

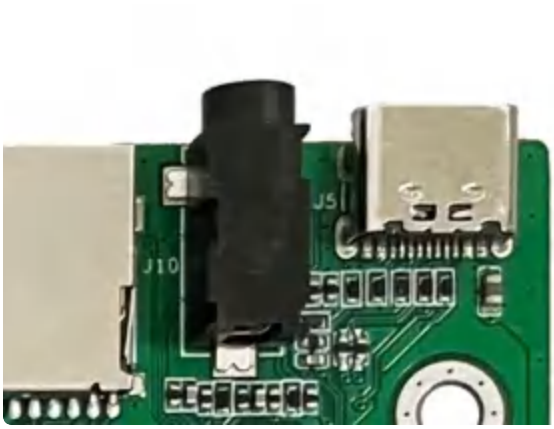
播放到HDMI：

```Bash
rk3568-buildroot:/root# aplay -D plug:spk_c0 Rear_Center.wav
Playing WAVE 'Rear_Center.wav' : Signed 16 bit Little Endian, Rate 48000 Hz, Mono

```

播放到Lineout：

```Bash
rk3568-buildroot:/root# aplay -D plug:spk_c1 Rear_Center.wav
Playing WAVE 'Rear_Center.wav' : Signed 16 bit Little Endian, Rate 48000 Hz, Mono

```

播放到耳机（需要插入耳机）：

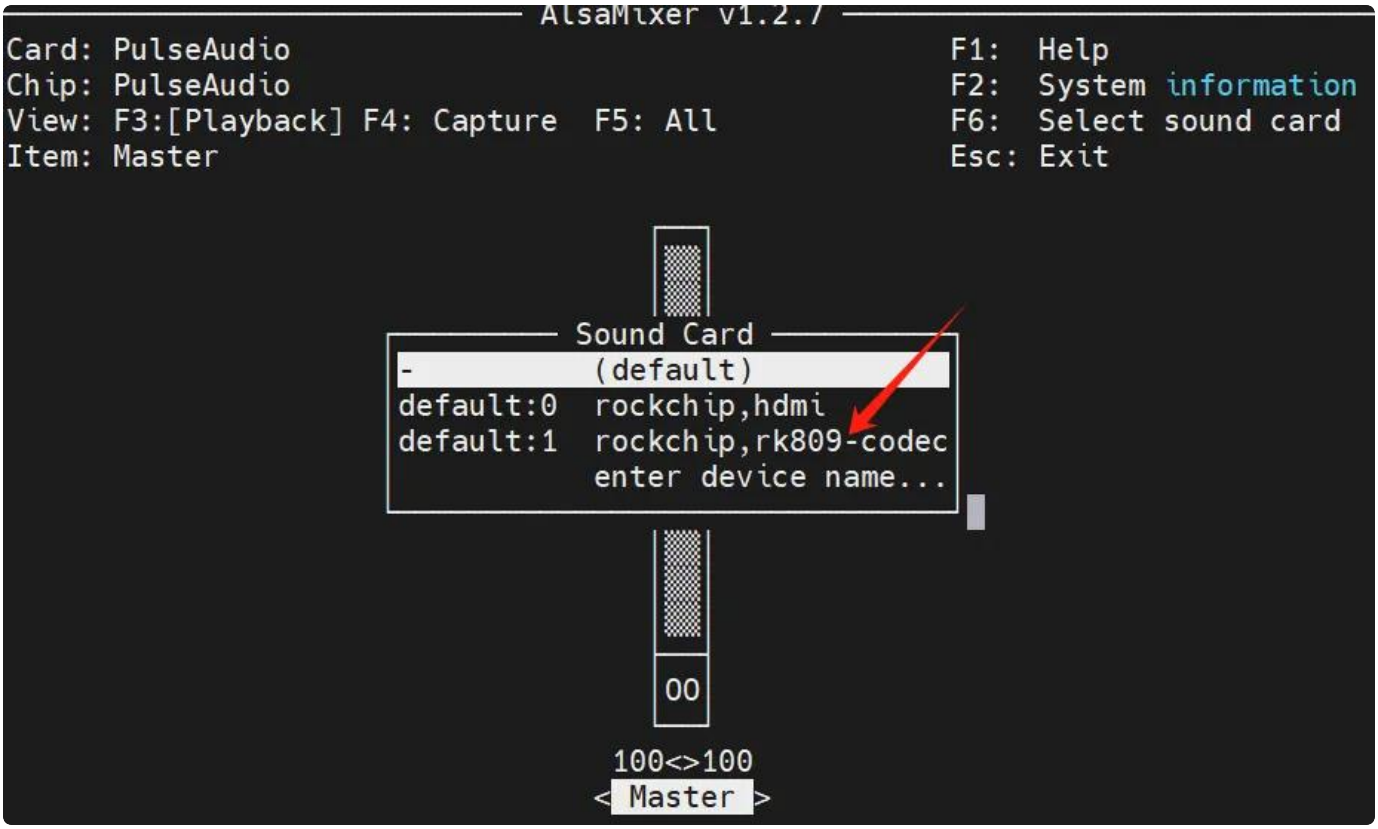


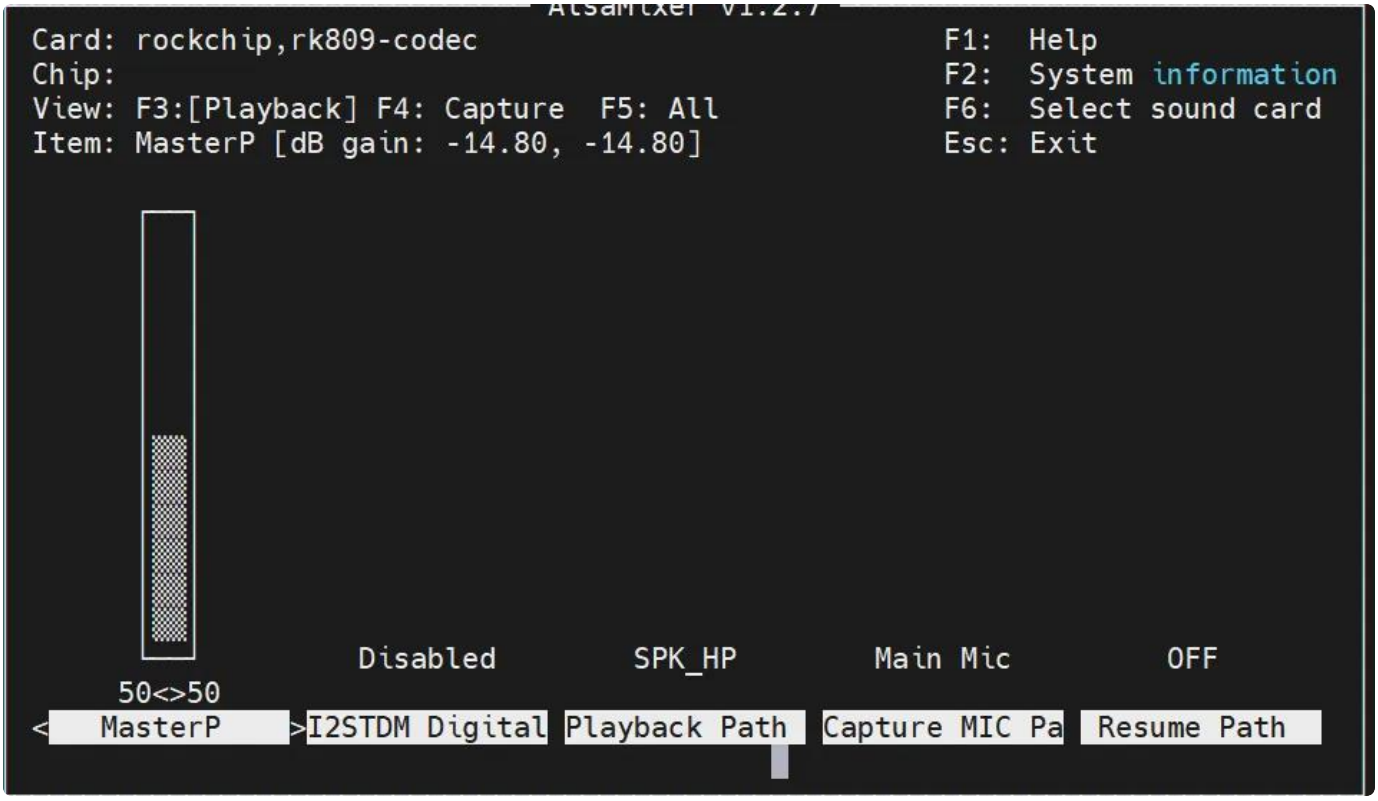
```Bash
rk3568-buildroot:/root# aplay -D plug:spk_c1 Rear_Center.wav
Playing WAVE 'Rear_Center.wav' : Signed 16 bit Little Endian, Rate 48000 Hz, Mono
```

注：如果播放到耳机或者喇叭没有声音，那么需要在alsamixer配置一下

```
rk3568-buildroot:/root# alsamixer

```

键盘按下"s",选择声卡

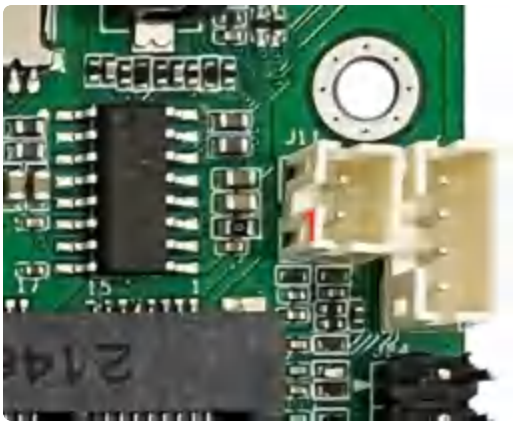




如图所示，最左边是控制播放音量大小的，hdmi也是一样，选择声卡进入就可以控制音量大小，playback path 何capture mic pa 必须选择，否则会播放没有声音

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：

```
root@rk3568-buildroot:/root# arecord -D hw:1,0 -r 48000 -c 2 -f S16_LE test.wav
Recording WAVE 'test.wav' : Signed 16 bit Little Endian, Rate 48000 Hz, Stereo
^CAborted by signal Interrupt...
root@rk3568-buildroot:/root#
```

注：因为我们的card 1，rk809集成了mic所以，我们在选择录音的时候选择的是声卡1即hw:1,0

录音完后播放到喇叭测试：

```
root@rk3568-buildroot:/root# aplay -D plug:spk_c1 test.wav
Playing WAVE 'test.wav' : Signed 16 bit Little Endian, Rate 48000 Hz, Stereo
^CAborted by signal Interrupt...
```

10、摄像头

主板默认适配OV5648+OV8858摄像头。

10.1 查看摄像头是否存在

```Bash
root@rk3568-buildroot:/# media-ctl -pd /dev/media0
Media controller API version 5.10.160

Media device information
------------------------
driver          rkisp-vir0
model           rkisp0
serial
bus info
hw revision     0x0
driver version  5.10.160

Device topology
- entity 1: rkisp-isp-subdev (4 pads, 7 links)
            type V4L2 subdev subtype Unknown flags 0
            device node name /dev/v4l-subdev0
        pad0: Sink
                [fmt:SBGGR10_1X10/2592x1944 field:none
                 crop.bounds:(0,0)/2592x1944
                 crop:(0,0)/2592x1944]
                <- "rkisp-csi-subdev":1 [ENABLED]
                <- "rkisp_rawrd0_m":0 []
                <- "rkisp_rawrd2_s":0 []
        pad1: Sink
                <- "rkisp-input-params":0 [ENABLED]
        pad2: Source
                [fmt:YUYV8_2X8/2592x1944 field:none colorspace:smpte170m q
uantization:full-range
                 crop.bounds:(0,0)/2592x1944
                 crop:(0,0)/2592x1944]
                -> "rkisp_mainpath":0 [ENABLED]
                -> "rkisp_selfpath":0 [ENABLED]
        pad3: Source
                -> "rkisp-statistics":0 [ENABLED]

......
......
......

- entity 70: m00_b_ov5648 4-0036 (1 pad, 1 link)
             type V4L2 subdev subtype Sensor flags 0
             device node name /dev/v4l-subdev3
        pad0: Source
                [fmt:SBGGR10_1X10/2592x1944@10000/150000 field:none]
                -> "rockchip-csi2-dphy0":0 [ENABLED]
```

```
45
```

10.2 抓取视频

```
root@rk3568-buildroot:/# 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
VIDIOC_QUERYCAP: ok
VIDIOC_G_FMT: ok
VIDIOC_S_FMT: ok
Format Video Capture Multiplanar:
        Width/Height      : 1920/1080
        Pixel Format      : 'NV12' (Y/CbCr 4:2:0)
        Field             : None
        Number of planes  : 1
        Flags             :
        Colorspace        : Default
        Transfer Function : Default
        YCbCr/HSV Encoding: Default
        Quantization      : Full Range
        Plane 0           :
           Bytes per Line : 1920
           Size Image     : 3110400
VIDIOC_G_SELECTION: ok
VIDIOC_S_SELECTION: ok
                VIDIOC_REQBUFS returned 0 (Success)
                VIDIOC_QUERYBUF returned 0 (Success)
                VIDIOC_QUERYBUF returned 0 (Success)
                VIDIOC_QUERYBUF returned 0 (Success)
                VIDIOC_QUERYBUF returned 0 (Success)
                VIDIOC_QBUF returned 0 (Success)
                VIDIOC_QBUF returned 0 (Success)
                VIDIOC_QBUF returned 0 (Success)
                VIDIOC_QBUF returned 0 (Success)
[  113.761757] rkisp_hw fdff0000.rkisp: set isp clk = 297000000Hz
[  113.774118] rockchip-csi2-dphy0: dphy0, data_rate_mbps 420
[  113.774173] rockchip-csi2-dphy csi2-dphy0: csi2_dphy_s_stream stream on:1, dphy0
[  113.774186] ov5648 4-0036: ov5648_s_stream(878) enter!
[  113.774193] ov5648 4-0036: stream on!!!
[  113.790858] ov5648 4-0036: ov5648_set_ctrl Unhandled id:0x9f0905, val:0x400
                VIDIOC_STREAMON returned 0 (Success)
cap dqbuf: 0 seq:      0 bytesused: 3110400 ts: 166.994238 (ts-monotonic, ts-src-eof)
cap dqbuf: 1 seq:      1 bytesused: 3110400 ts: 167.060730 delta: 66.492 ms (ts-monotonic, ts-src-eof)

```

```
cap dqbuf: 2 seq:      2 bytesused: 3110400 ts: 167.127250 delta: 66.520 ms (ts-monotonic, ts-src-eof)
cap dqbuf: 3 seq:      3 bytesused: 3110400 ts: 167.193752 delta: 66.502 ms (ts-monotonic, ts-src-eof)
cap dqbuf: 0 seq:      4 bytesused: 3110400 ts: 167.260267 delta: 66.515 ms fps: 15.04 (ts-monotonic, ts-src-eof)
cap dqbuf: 1 seq:      5 bytesused: 3110400 ts: 167.326780 delta: 66.513 ms fps: 15.04 (ts-monotonic, ts-src-eof)
cap dqbuf: 2 seq:      6 bytesused: 3110400 ts: 167.393286 delta: 66.506 ms fps: 15.04 (ts-monotonic, ts-src-eof)
cap dqbuf: 3 seq:      7 bytesused: 3110400 ts: 167.459802 delta: 66.516 ms fps: 15.04 (ts-monotonic, ts-src-eof)
cap dqbuf: 0 seq:      8 bytesused: 3110400 ts: 167.526312 delta: 66.510 ms fps: 15.04 (ts-monotonic, ts-src-eof)
cap dqbuf: 1 seq:      9 bytesused: 3110400 ts: 167.592825 delta: 66.513 ms fps: 15.04 (ts-monotonic, ts-src-eof)
cap dqbuf: 2 seq:     10 bytesused: 3110400 ts: 167.659336 delta: 66.511 ms fps: 15.04 (ts-monotonic, ts-src-eof)
cap dqbuf: 3 seq:     11 bytesused: 3110400 ts: 167.725846 delta: 66.510 ms fps: 15.04 (ts-monotonic, ts-src-eof)
cap dqbuf: 0 seq:     12 bytesused: 3110400 ts: 167.792353 delta: 66.507 ms fps: 15.04 (ts-monotonic, ts-src-eof)
cap dqbuf: 1 seq:     13 bytesused: 3110400 ts: 167.858870 delta: 66.517 ms fps: 15.04 (ts-monotonic, ts-src-eof)
cap dqbuf: 2 seq:     14 bytesused: 3110400 ts: 167.925377 delta: 66.507 ms fps: 15.04 (ts-monotonic, ts-src-eof)
cap dqbuf: 3 seq:     15 bytesused: 3110400 ts: 167.991892 delta: 66.515 ms fps: 15.04 (ts-monotonic, ts-src-eof)
cap dqbuf: 0 seq:     16 bytesused: 3110400 ts: 168.058402 delta: 66.510 ms fps: 15.04 (ts-monotonic, ts-src-eof)
cap dqbuf: 1 seq:     17 bytesused: 3110400 ts: 168.124914 delta: 66.512 ms fps: 15.04 (ts-monotonic, ts-src-eof)
cap dqbuf: 2 seq:     18 bytesused: 3110400 ts: 168.191427 delta: 66.513 ms fps: 15.04 (ts-monotonic, ts-src-eof)
cap dqbuf: 3 seq:     19 bytesused: 3110400 ts: 168.257938 delta: 66.511 ms fps: 15.04 (ts-monotonic, ts-src-eof)
cap dqbuf: 0 seq:     20 bytesused: 3110400 ts: 168.324442 delta: 66.504 ms fps: 15.04 (ts-monotonic, ts-src-eof)

```

11、RTC

主板包含2个RTC，其中/dev/rtc1为外部RTC（HYM8563），/dev/rtc0为CPU内部的RTC（RK808）。系统默认使用rtc0的时间。所以这里有两种解决方法，如果rtc-hym8563是rtc0，则直接

设置即可

```
root@rk3568-buildroot:/# dmesg | grep rtc
[    1.583028] rk808-rtc rk808-rtc: registered as rtc0
[    1.584797] rk808-rtc rk808-rtc: setting system clock to 2017-08-04T09:00:03 UTC (1501837203)
[    1.601112] rtc-hym8563 5-0051: registered as rtc1

```

方法一

11.1 获取RTC时间

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

11.2 设置RTC时间

```
root@rk3568-buildroot:/# date -s '2000-01-30 1:1:1'
Sun Jan 30 01:01:01 UTC 2000
root@rk3568-buildroot:/# hwclock -w -f /dev/rtc1
root@rk3568-buildroot:/# hwclock -r -f /dev/rtc1
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

```Bash
root@rk3568-buildroot:/# date
Fri Aug  4 09:00:17 UTC 2017

//写入系统时间
root@rk3568-buildroot:/# hwclock --hctosys --rtc=/dev/rtc1
root@rk3568-buildroot:/# date
Sun Jan 30 01:03:58 UTC 2000
root@rk3568-buildroot:/# hwclock -r -f /dev/rtc1
Sun Jan 30 01:04:17 2000  0.000000 seconds

```

方法二

如果不想这么麻烦的话，在内核中找到CONFIG_RTC_DRV_RK808把他关掉就行

```Bash
#CONFIG_RTC_DRV_RK808
```

这时，外部rtc的节点就是外部RTC（HYM8563），也是系统默认使用的rtc，我们常规设置就可以

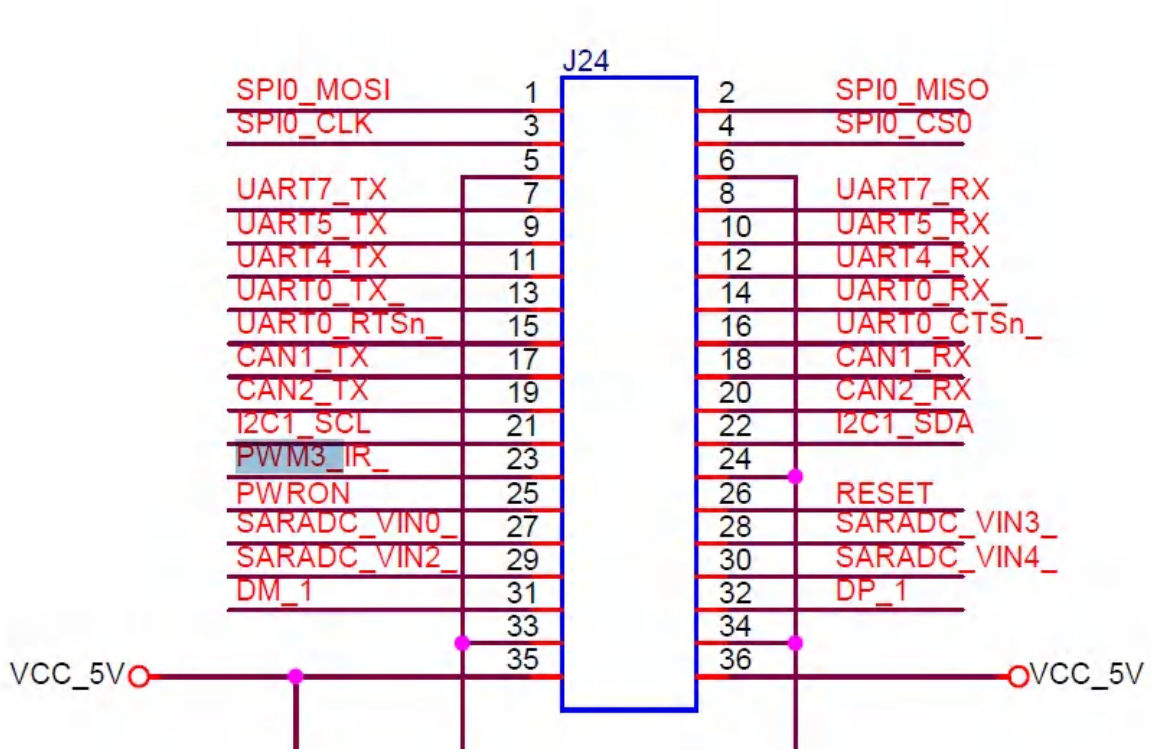
```Bash
root@rk3568-buildroot:/# hwclock
Fri Aug  4 09:07:26 2017  0.000000 seconds

//设置系统时间
root@rk3568-buildroot:/# date -s '2023-12-19 16:15:20'
Tue Dec 19 16:15:20 UTC 2023

//把系统时间写入rtc
root@rk3568-buildroot:/# hwclock -w
root@rk3568-buildroot:/# hwclock -r
Tue Dec 19 16:15:25 2023  0.000000 seconds

//断电重启后，直接hwclock就会把rtc时间写入系统
root@rk3568-buildroot:/# hwclock
Tue Dec 19 16:16:55 2023  0.000000 seconds


```

12、开机自启动

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

13、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	

13.1 测试

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

开启PWM:

```Bash
echo 0 > export
cd pwm0
```

设置周期:

```Bash
echo 330000> period
```

设置占空比:

```Bash
echo 150000 > duty_cycle
```

使能PWM:

```Bash
echo 1 > enable
```

失能PWM:

```Bash
echo 0 > enable
```

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。（/backlight/brightness没有这个） (backlight/brightness没找到)

设置亮度为100：

```Bash
root@rk3568-buildroot:/#  echo 100 >  /sys/class/backlight/backlight/brightness
```

15、按键

主板配置了一个按键SW2，作为电源键使用，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 = (raw/1024)*1.8v

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。

```
root@rk3568-buildroot:/# cat /sys/bus/iio/devices/iio:device0/in_voltage3_raw
997
```

设备节点读取的raw值为997，代入到公式计算：

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

即ADC1输入的电压为1.75v。

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

```
root@rk3568-buildroot:/# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         _gateway        0.0.0.0         UG    100    0        0 eth0
default         _gateway        0.0.0.0         UG    600    0        0 wlan0
192.168.1.0     0.0.0.0         255.255.255.0   U     100    0        0 eth0
192.168.1.0     0.0.0.0         255.255.255.0   U     600    0        0 wlan0
```

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：

```
root@rk3568-buildroot:/# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         _gateway        0.0.0.0         UG    100    0        0 eth0
default         _gateway        0.0.0.0         UG    600    0        0 wlan0
192.168.1.0     0.0.0.0         255.255.255.0   U     100    0        0 eth0
192.168.1.0     0.0.0.0         255.255.255.0   U     600    0        0 wlan0
root@rk3568-buildroot:/# route del default dev eth0
root@rk3568-buildroot:/# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         _gateway        0.0.0.0         UG    600    0        0 wlan0
192.168.1.0     0.0.0.0         255.255.255.0   U     100    0        0 eth0
192.168.1.0     0.0.0.0         255.255.255.0   U     600    0        0 wlan0
```

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：

```
root@rk3568-buildroot:/# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         _gateway        0.0.0.0         UG    600    0        0 wlan0
192.168.1.0     0.0.0.0         255.255.255.0   U     100    0        0 eth0
192.168.1.0     0.0.0.0         255.255.255.0   U     600    0        0 wlan0
root@rk3568-buildroot:/#
```

设置为以太网优先：

```
root@rk3568-buildroot:/# route del default dev wlan0
root@rk3568-buildroot:/# route add default dev eth0
root@rk3568-buildroot:/# route add default gw 192.168.1.1
root@rk3568-buildroot:/# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         0.0.0.0         0.0.0.0         U     0      0        0 eth0
192.168.1.0     0.0.0.0         255.255.255.0   U     100    0        0 eth0
192.168.1.0     0.0.0.0         255.255.255.0   U     600    0        0 wlan0
root@rk3568-buildroot:/#
```

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。

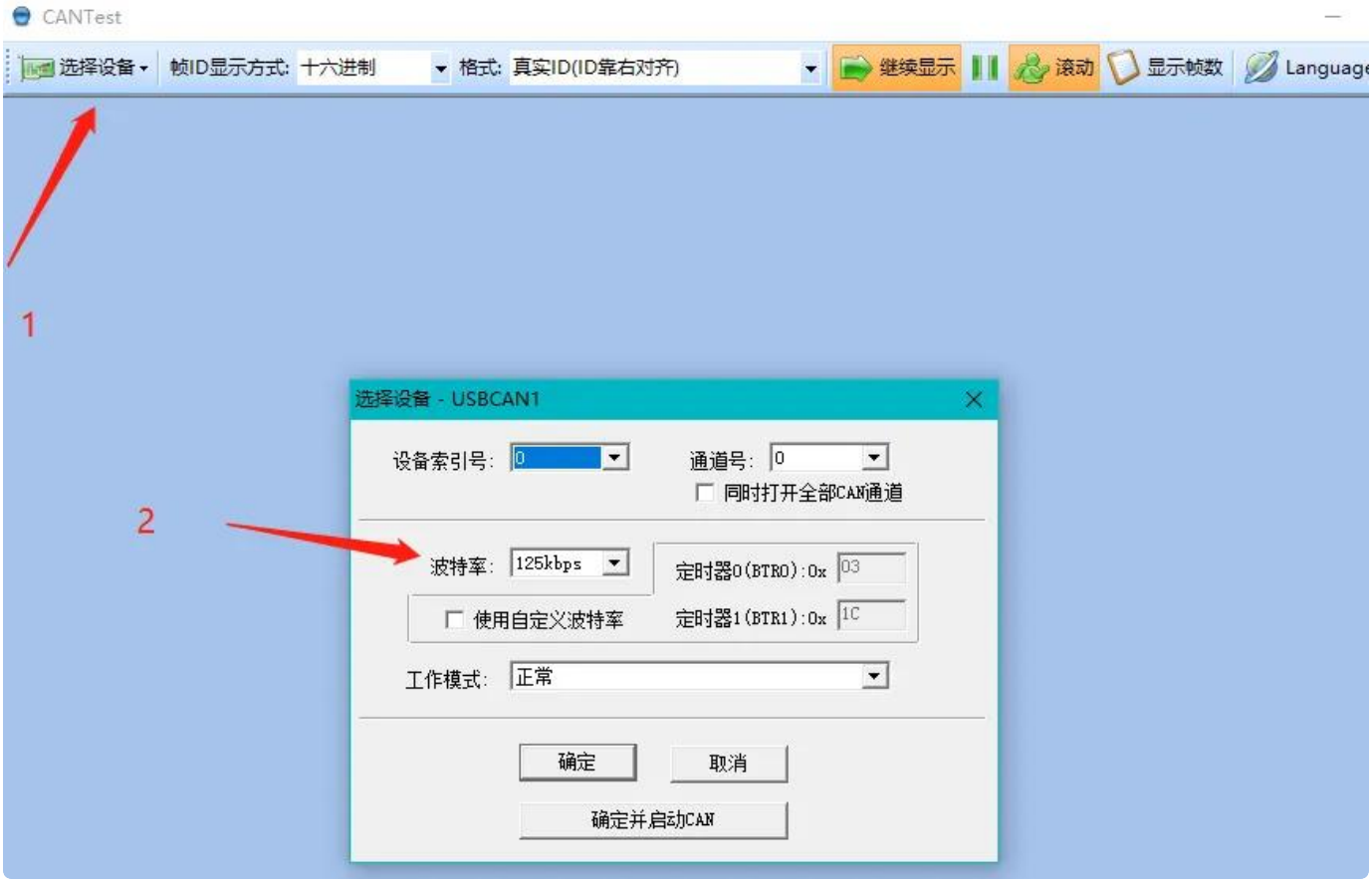
18.1测试

测试需要用USB转CAN工具，通过PC与板上CAN通信。

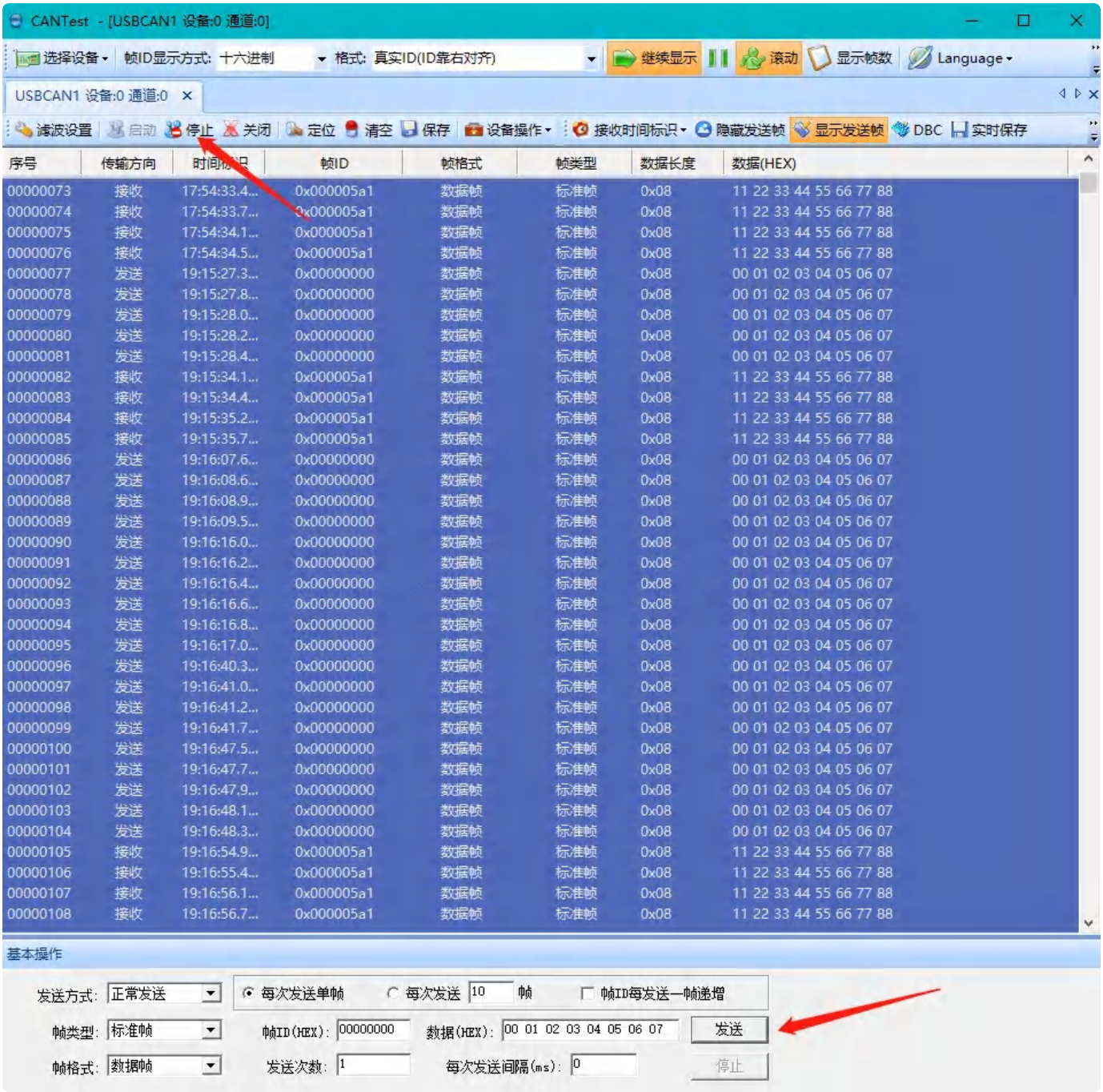
这里以CAN0为例，其余节点测试方法相同：

```Bash
//关掉can
ifconfig can0 down

//配置can通信的波特率
ip link set can0 type can bitrate 125000 triple-sampling on

//开启can通信
ifconfig can0 up

//作为接收端接收数据
candump can0

//作为发送端发送数据
cansend can0 5A1#1122334455667788

```

(1) 选择USBCAN1



（2）启动CAN测试



板端发送过来的数据可以在CANtest上打印出来。