IDO-EVB3568-V1 - Buildroot系统使用说明

Industio 触觉智能

IDO-EVB3568-V1
Buildroot 系统使用说明

文档修订历史

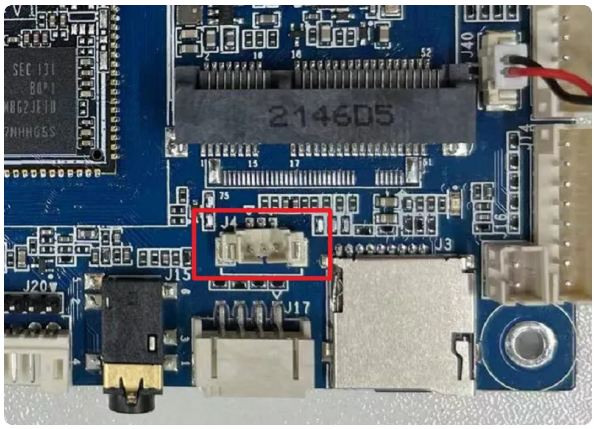
版本	修订内容	修订	审核	日期
V1.0	创建文档	谭文学		2022/10/25
V1.1	修改开机启动程序描述 增加ADC电压获取计算方法 修改时区的描述 增加录音的描述	谭文学		2022/10/26

调试

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

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。



为TTL电平，通信参数为1500000 8 N 1。

ADB调试

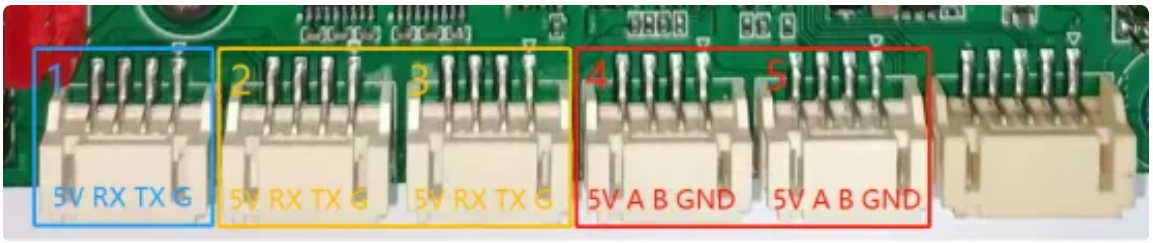
上图红色框内的USB接口为支持OTG模式切换，使用双公头 USB 数据线连接开发板和 PC 端的USB接口，在PC终端识别到 ADB 设备，即可使用 adb shell 调试。

```
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:/]#
```

SSH调试

SSH登录账号密码为： **root @ rockchip**。

串口测试



串口接口位置及引脚定义如上图所示，设备节点列表如下：

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

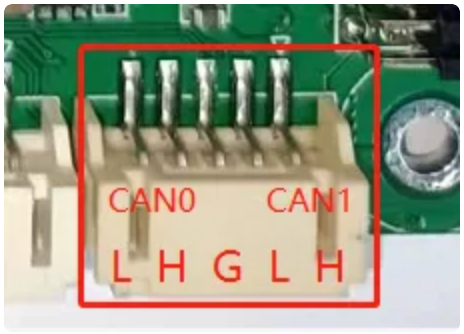
Plain Text

```
[root@RK356X:/]# microcom -s 115200 /dev/ttyS0
[ 2683.415483] of_dma_request_slave_channel: dma-names property of node '/serial@fdd50000' missing or empty
[ 2683.415544] ttyS0 - failed to request DMA, use interrupt mode

```

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+x键停止测试。

CAN测试



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

```bash
#关闭can0设备
ip link set can0 down

#设置仲裁段1M波特率，数据段3M波特率
ip link set can0 type can bitrate 1000000 dbitrate 3000000 fd on

#打印can0信息
ip -details link show can0

#启动can0
ip link set can0 up

#执行candump，阻塞等待can0接收
candump can0

#canfd格式发送
cansend can0 123##1DEADBEEF

#can格式发送
cansend can0 123#1122334455667788
```

WIFI使用

在使用 WIFI时连接好WiFi天线，设备节点为**wlan0**

```
[root@RK356X:/]# ifconfig wlan0
wlan0     Link encap:Ethernet  HWaddr 2C:3B:70:14:17:95
          inet addr:169.254.41.145  Bcast:169.254.255.255  Mask:255.255.0.0
          inet6 addr: fe80::b05:fca4:fb45:9468/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:75 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:21920 (21.4 KiB)
```

系统开机通过/etc/init.d/S80wifireconnect脚本开启WiFi服务，修改/userdata/cfg/wpa_supplicant.conf，填写正确的热点账号和密码：

```
[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:/]#
```

重启后，将自动连接上热点：

```
[root@RK356X:/]# ifconfig wlan0
wlan0     Link encap:Ethernet  HWaddr 2C:3B:70:14:17:95
          inet addr:192.168.1.101  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::220a:b25:4bd:2e3a/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:26 errors:0 dropped:0 overruns:0 frame:0
          TX packets:40 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:5075 (4.9 KiB)  TX bytes:3913 (3.8 KiB)

[root@RK356X:/]#
```

蓝牙使用

设备节点为**hci0**，通过/usr/bin/bt_init.sh脚本开启蓝牙功能

```bash
[root@RK356X:/]# /usr/bin/bt_init.sh
```

蓝牙功能开启后，将产生hci0节点

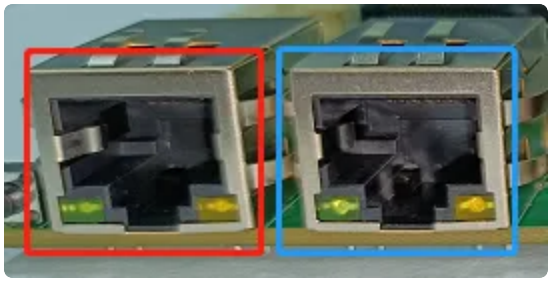
```bash
[root@RK356X:/]# hciconfig  -a
hci0:   Type: Primary  Bus: UART
        BD Address: F3:7A:FA:A4:5E:22  ACL MTU: 1021:8  SCO MTU: 64:1
        DOWN
        RX bytes:668 acl:0 sco:0 events:34 errors:0
        TX bytes:423 acl:0 sco:0 commands:34 errors:0
        Features: 0xbf 0xfe 0xcf 0xfe 0xdb 0xff 0x7b 0x87
        Packet type: DM1 DM3 DM5 DH1 DH3 DH5 HV1 HV2 HV3
        Link policy: RSWITCH SNIFF
        Link mode: SLAVE ACCEPT
```

使用**hcitool**测试蓝牙扫描功能

```bash
[root@RK356X:/]# hciconfig hci0 up
[root@RK356X:/]# hcitool -i hci0 scan
Scanning ...
        94:87:E0:9D:14:12       seeyou
        4C:4F:EE:12:6C:A3       OnePlus 8 Pro
        5C:C5:63:02:31:19       客厅的小米电视
```

以太网使用

开发板两路千兆以太网接口，上图红色框内接口设备节点为 eth0，蓝色框内接口设备节点为 eth1。



两路以太网接口默认IP获取方式为 dhcp。

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

```Bash
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
address 192.168.0.234
netmask 255.255.255.0
gateway 192.168.0.1
dns-nameservers 114.114.114.114
```

其中，dns-nameservers一项为默认dns。

4G

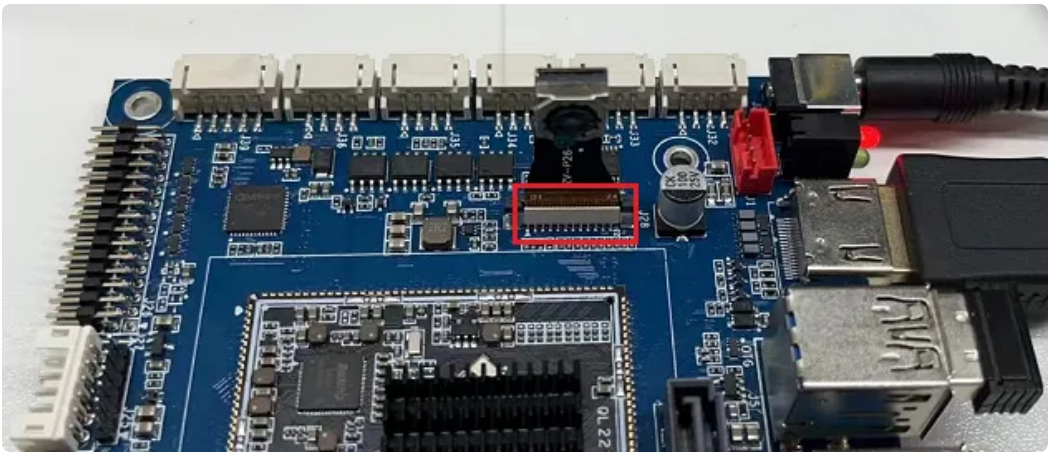
IDO-EVB3568-V1 默认适配EC20模块。

检查拨号上网是否正常

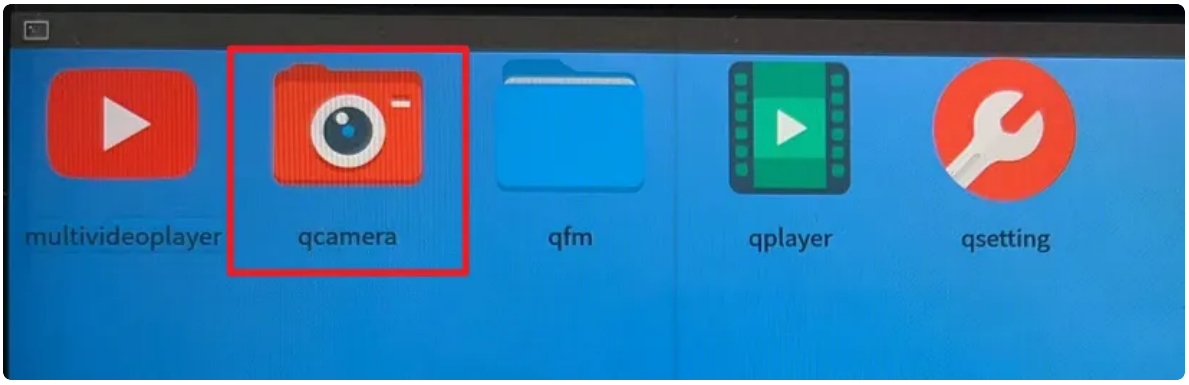
```Bash
ping 114.114.114.114 -I wwan0
```

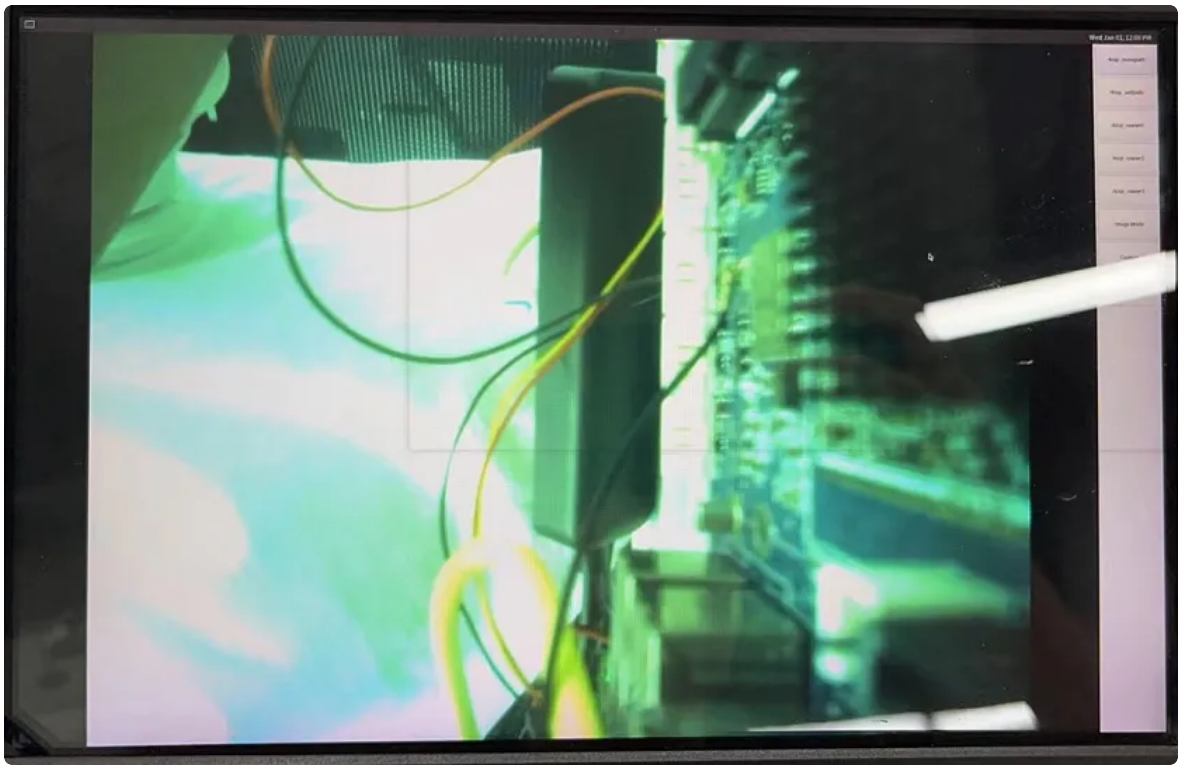
摄像头使用

默认适配OV5648模块，对应系统中的设备节点为/dev/video0。



使用qcamera工具来进行测试。





U盘

除了红框的接口，其余均为USB-HOST。

红框USB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：

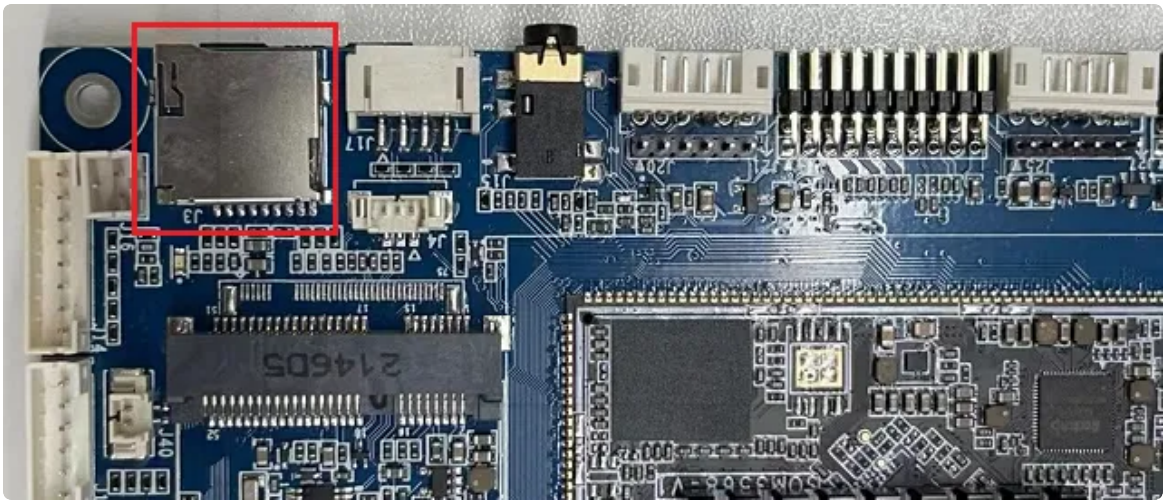
```Java
host:
echo host > /sys/devices/platform/fe8a0000.usb2-phy/otg_mode
device:
echo peripheral > /sys/devices/platform/fe8a0000.usb2-phy/otg_mode
```

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)

```Bash
[root@RK356X:/]# mount
...
/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)
...
```

SD卡

将SD卡插入到SD卡槽中，将自动挂载到/mnt/sdcard/目录下。



```Bash
[root@RK356X:/]# mount
...
/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)
...
```

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：

```bash
[root@RK356X:/]# cat /etc/init.d/S49ntp
#! /bin/sh

NAME=ntpd
DAEMON=/usr/sbin/$NAME

# Gracefully exit if the package has been removed.
test -x $DAEMON || exit 0

# Read config file if it is present.
if [ -r /etc/default/$NAME ]
then
  . /etc/default/$NAME
fi

case "$1" in
  start)
    printf "Starting $NAME: "
    start-stop-daemon -S -q -x $DAEMON -- -g
    [ $? = 0 ] && echo "OK" || echo "FAIL"
    ;;
  stop)
    printf "Stopping $NAME: "
    start-stop-daemon -K -q -n $NAME
    [ $? = 0 ] && echo "OK" || echo "FAIL"
    ;;
  restart|reload)
    echo "Restarting $NAME: "
    $0 stop
    sleep 1
    $0 start
    ;;
  *)
    echo "Usage: $0 {start|stop|restart|reload}" >&2
    exit 1
    ;;
esac

exit 0
```

在开机的时候，会进入到start)；在关机的时候，会进入到stop)。

按键

IDO-EVB3568-V1 配置了一个Recovery按键，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/event2**，键值为**KEY_VOLUMEUP(115)**。

使用evtest进行测试：

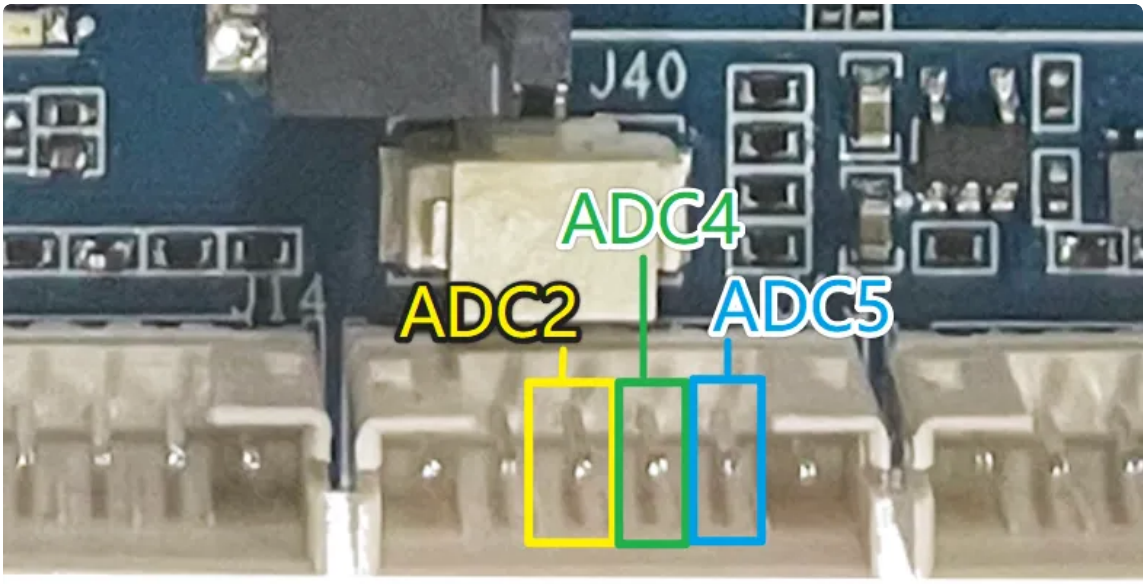
```Bash
[root@RK356X:/]# evtest
No device specified, trying to scan all of /dev/input/event*
Available devices:
/dev/input/event0:      fe6e0030.pwm
/dev/input/event1:      rk805 pwrkey
/dev/input/event2:      adc-keys
/dev/input/event3:      rockchip,rk809-codec Headphones
Select the device event number [0-3]: 2
Input driver version is 1.0.1
Input device ID: bus 0x19 vendor 0x1 product 0x1 version 0x100
Input device name: "adc-keys"
Supported events:
  Event type 0 (EV_SYN)
  Event type 1 (EV_KEY)
    Event code 114 (KEY_VOLUMEDOWN)
    Event code 115 (KEY_VOLUMEUP)
    Event code 139 (KEY_MENU)
    Event code 158 (KEY_BACK)
Properties:
Testing ... (interrupt to exit)
Event: time 1666752551.345149, type 1 (EV_KEY), code 115 (KEY_VOLUMEUP), value 1
Event: time 1666752551.345149, ---------- SYN_REPORT ------------
Event: time 1666752551.551624, type 1 (EV_KEY), code 115 (KEY_VOLUMEUP), value 0
Event: time 1666752551.551624, ---------- SYN_REPORT ------------
Event: time 1666752552.274980, type 1 (EV_KEY), code 115 (KEY_VOLUMEUP), value 1
Event: time 1666752552.274980, ---------- SYN_REPORT ------------
Event: time 1666752552.688312, type 1 (EV_KEY), code 115 (KEY_VOLUMEUP), value 0
Event: time 1666752552.688312, ---------- SYN_REPORT ------------

```

在选择event number为2后，按下RECOVERY按键，即可看到按下和松开打印的信息。

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

ADC值读取

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

ADC电压转换关系

```Bash
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

时间设置 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

```Bash
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

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

11.2 设置RTC时间

```Bash
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


```

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。

时区

查看时区

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

+0000表示在0时区。

设置时区

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

音频

使用aplay工具查看声卡设备

```Bash
[root@RK356X:/]# aplay  -l
**** List of PLAYBACK Hardware Devices ****
card 0: 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
```

Lineout

不插入耳机，使用aplay播放wav音频测试

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

耳机

插入耳机，使用aplay播放wav音频测试

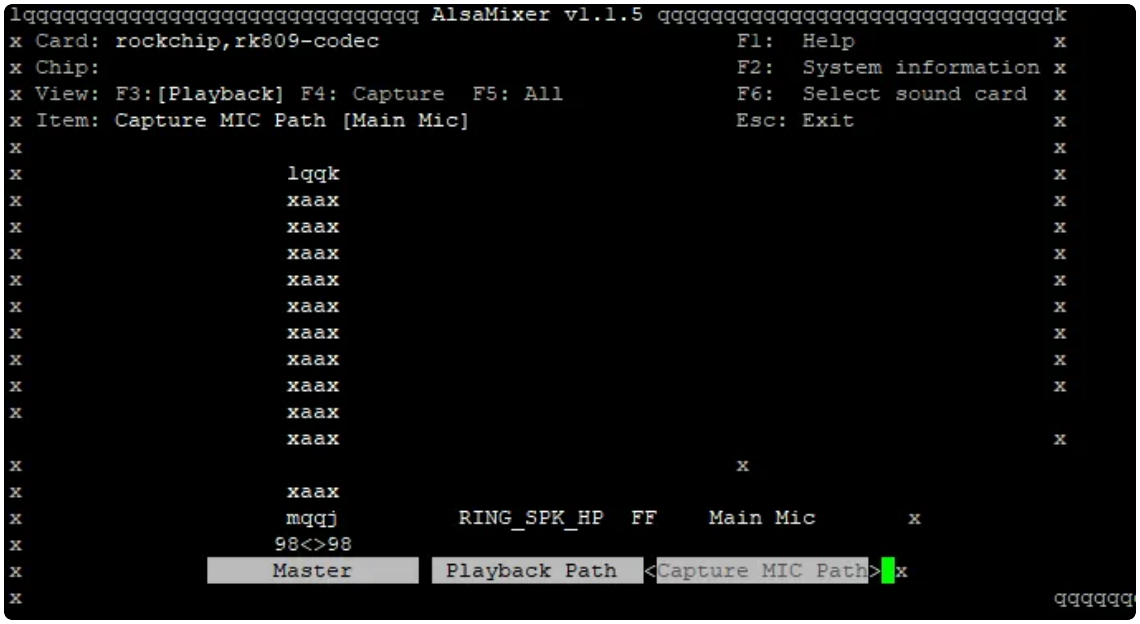
```Bash
[root@RK356X:/]# aplay  /etc/bsa_file/8k16bpsStereo.wav
Playing WAVE '/etc/bsa_file/8k16bpsStereo.wav' : Signed 16 bit Little Endian, Rate 8000 Hz, Stereo
```

录音

打开mic通道

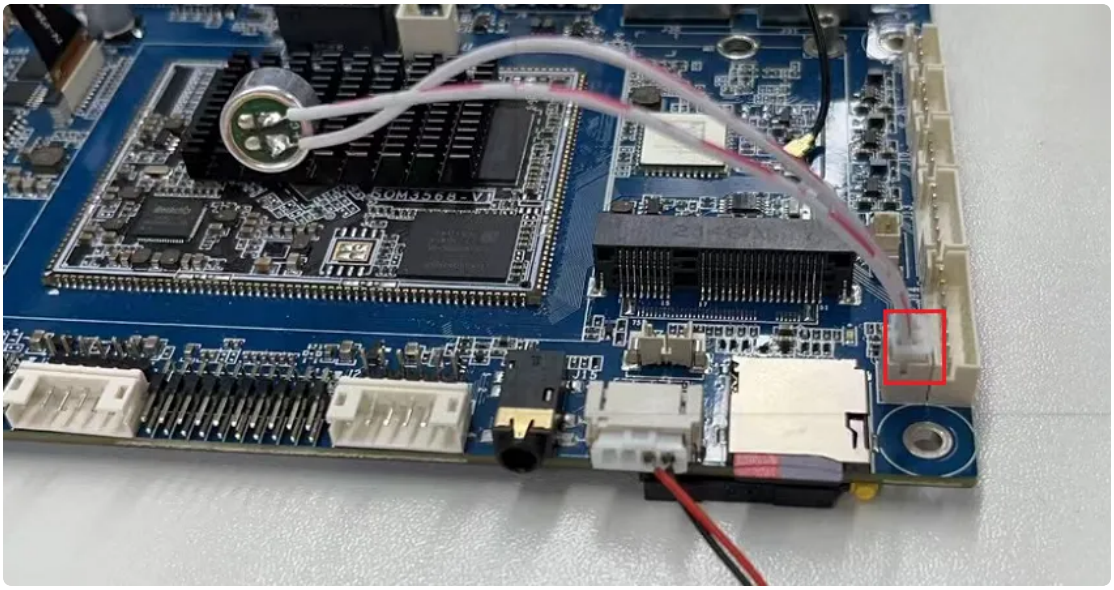
```Bash
alsamixer
```

Capture MIC Path选择**Main Mic**



录音

参考如下示意图接上麦克风。



使用arecord工具进行录音测试。

```
[root@RK356X:/]# arecord -D hw:0,0 -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...
```

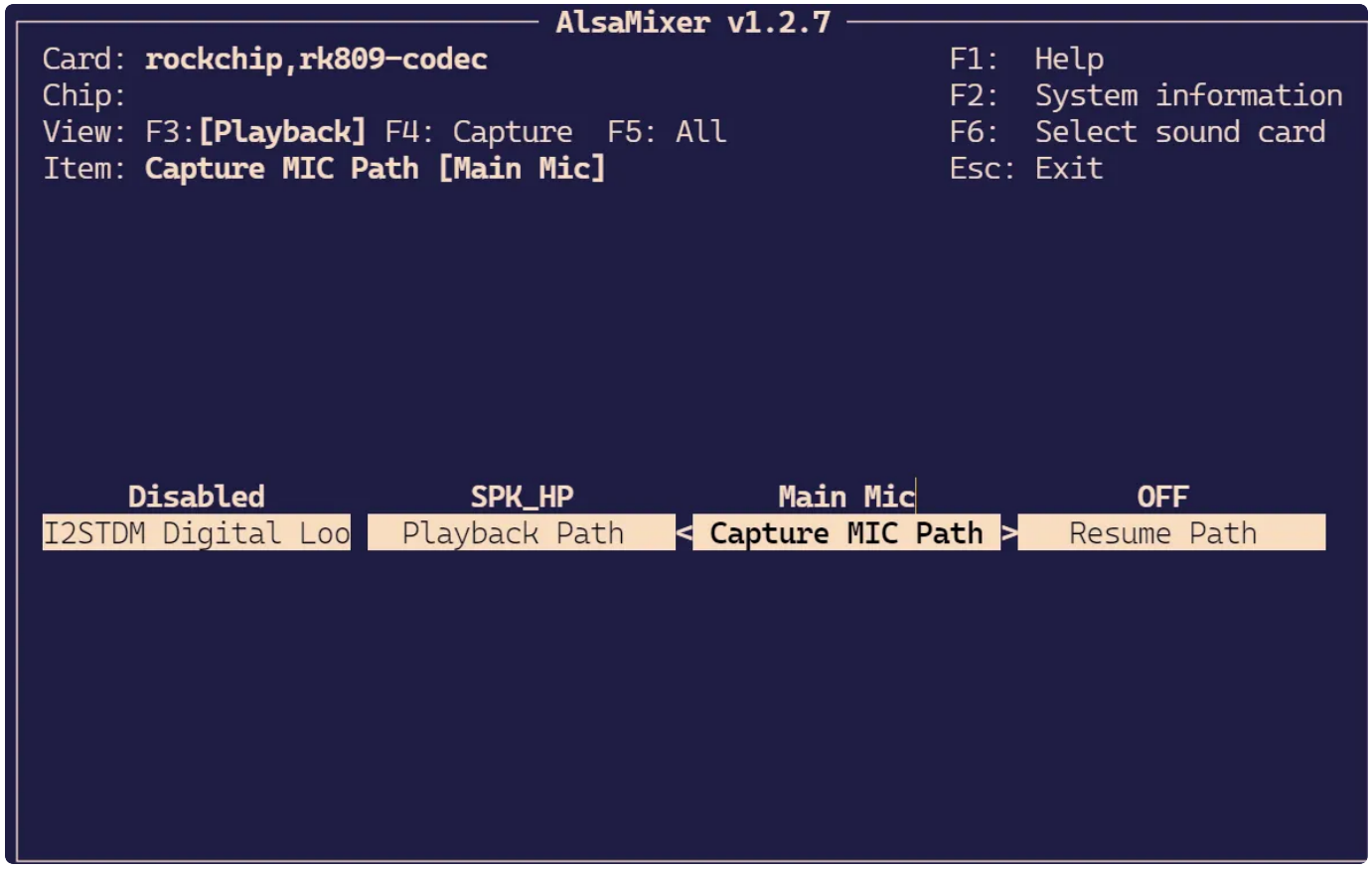
播放录音

```Bash
[root@RK356X:/]# aplay test.wav
```

5.10音频

进入音频设置的图形界面确保以下配置打开

```Bash
alsamixer
```



播放到HDMI：

```Bash
aplay -D plug:spk_c0 /usr/share/sounds/alsa/Rear_Center.wav
```

播放到Lineout：

```Bash
aplay -D plug:spk_c1 /usr/share/sounds/alsa/Rear_Center.wav
```

播放到耳机（需要插入耳机）：

```Bash
aplay -D plug:spk_c1 /usr/share/sounds/alsa/Rear_Center.wav
```

注意：这里是根据你的声卡选择，如果是接的其他屏幕，如mipi，那么只有一个声卡的情况下，喇叭选择的应该是

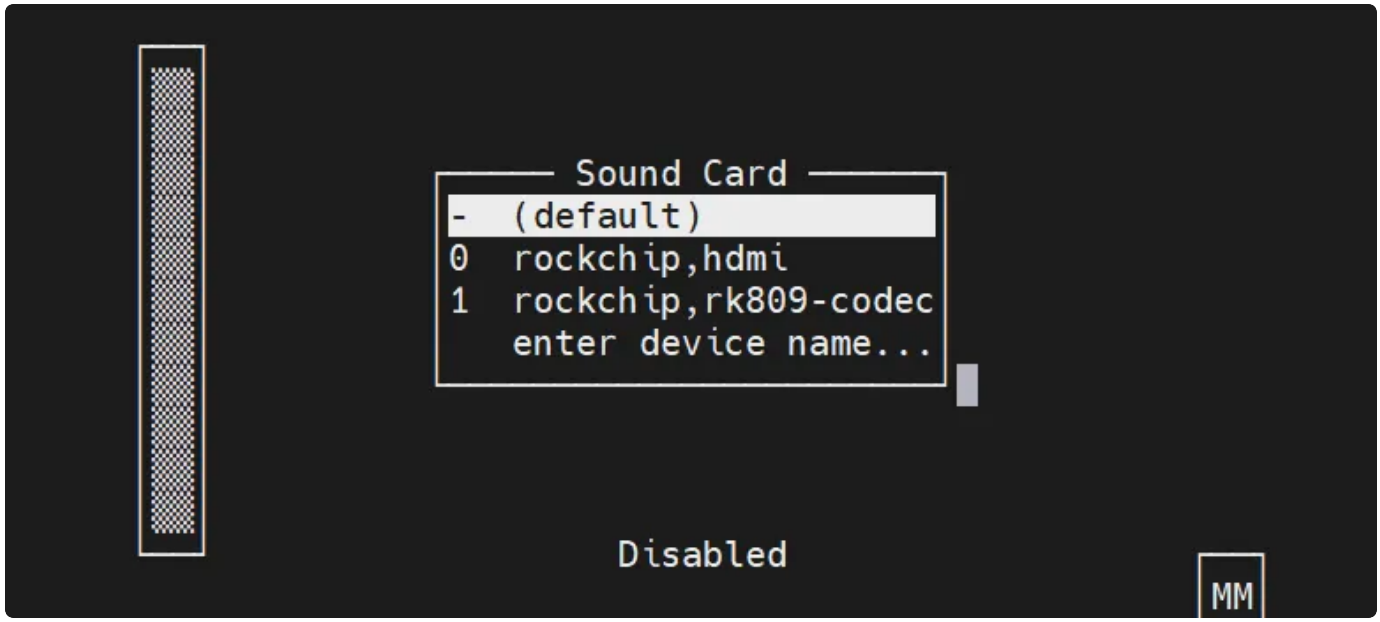
```Bash
aplay -D plug:spk_c0 /usr/share/sounds/alsa/Rear_Center.wav
```

音量调节：

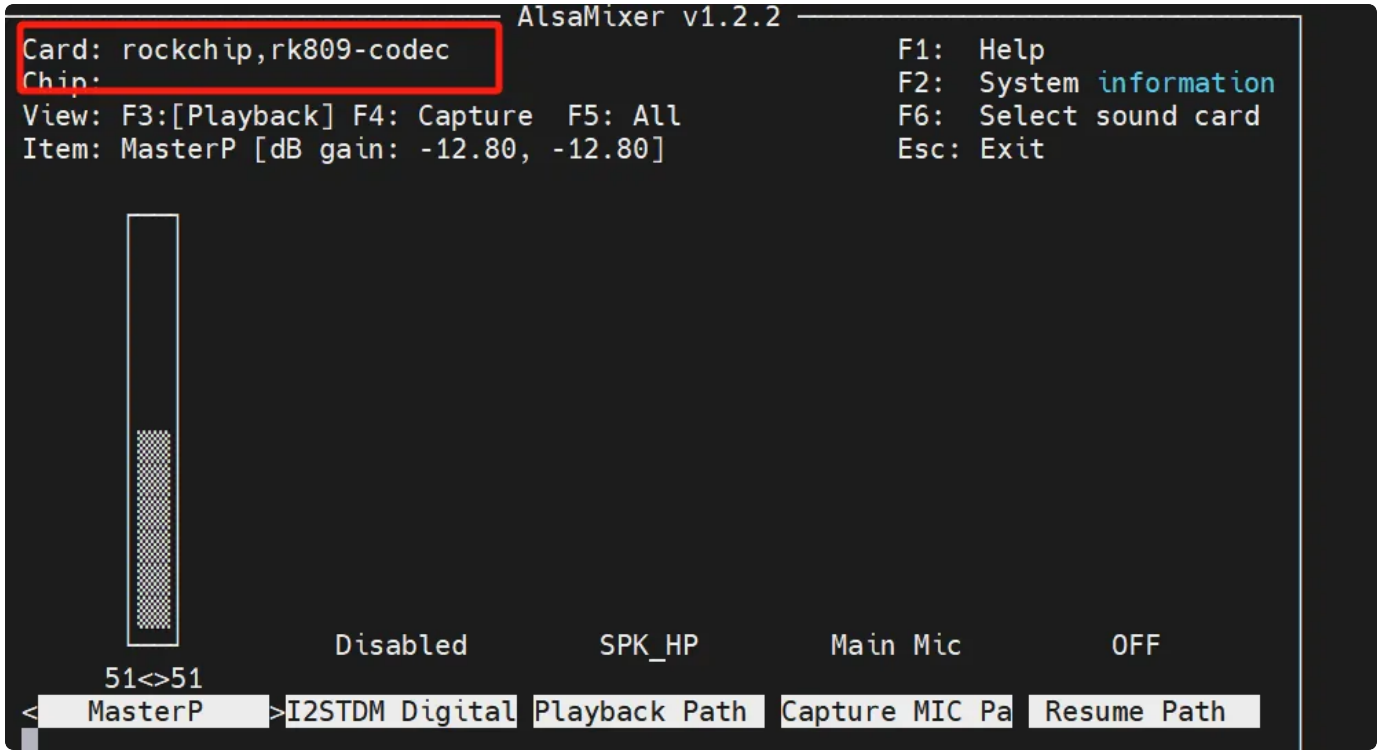
执行alsamixer进入图形界面进行调试

```Bash
alsamixer
```

进入图形界面，按s键，选择声卡，如果是喇叭或者耳机则选择为1，如果是hdmi音频则选择为0

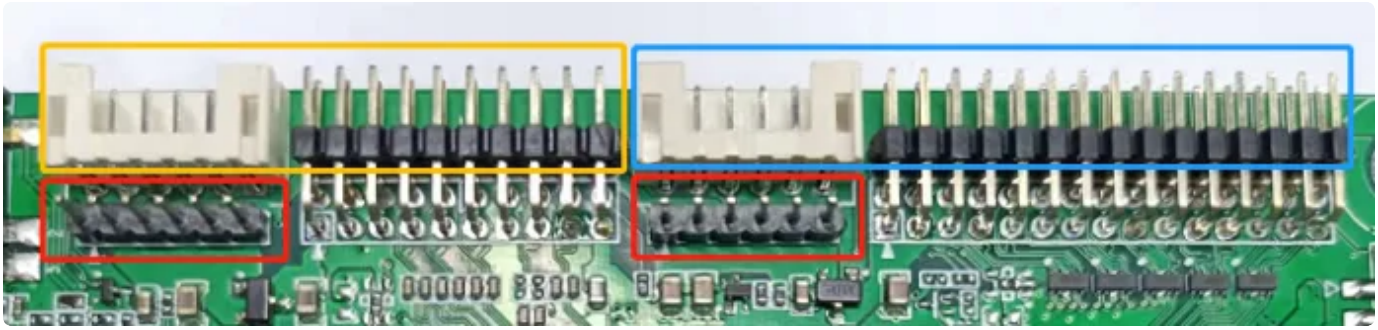


这里可以调节音量为51



显示屏

显示屏接口说明



黄色框是eDP屏接口，蓝色框是duallvds屏接口。
红色框分别是两者的供电口，提供12/5/3.3V。

开发板背面，其中红色框是mipi屏接口，蓝色框I2C触摸屏接口。

显示设置

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
#关闭
echo 0 > /sys/class/backlight/backlight/brightness
#最亮
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)

```bash
#关闭
echo 0 > /sys/class/backlight/backlight1/brightness
#最亮
echo 255 > /sys/class/backlight/backlight1/brightness
```

扩展GPIO

主板扩展了5路GPIO，位于J14：

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

测试方法

以GPIO4_C2为例。

作为输入：

```bash
echo 146 > /sys/class/gpio/export
echo in > /sys/class/gpio/gpio146/direction
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：

```bash
echo 146 > /sys/class/gpio/export
echo out > /sys/class/gpio/gpio146/direction
#输出高电平
echo 1 > /sys/class/gpio/gpio146/value
#输出低电平
echo 0 > /sys/class/gpio/gpio146/value
```