IDO-EVB3568-V1 - Ubuntu系统使用说明

Industio
触 觉 智 能

IDO-EVB3568-V1

Ubuntu 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	1、创建文档	谭文学		2022/10/26
V1.1	1、增加SSH调试； 2、补充蓝牙部分的描述；	谭文学		2022/10/28

TODO

1、修改ubuntu文件系统的蓝牙固件，修复有时扫描不了的问题；

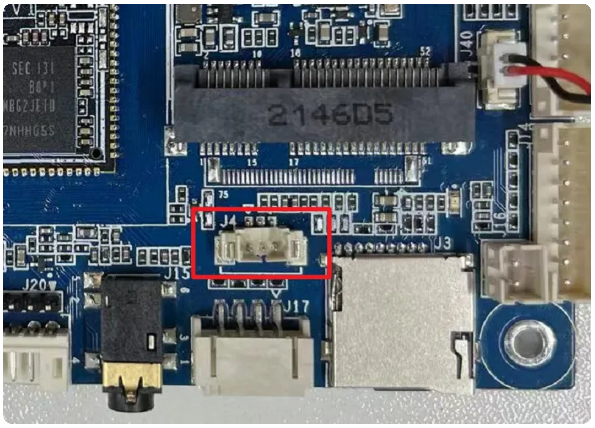
2、重新上传ubuntu系统固件；

调试

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

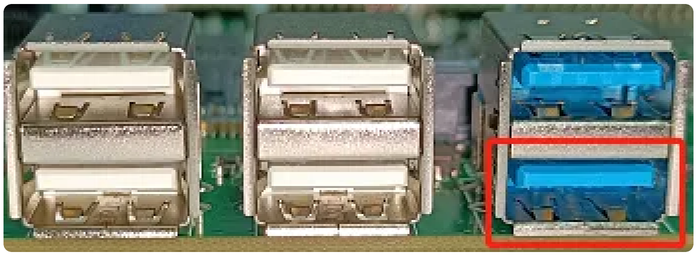
串口调试

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**。

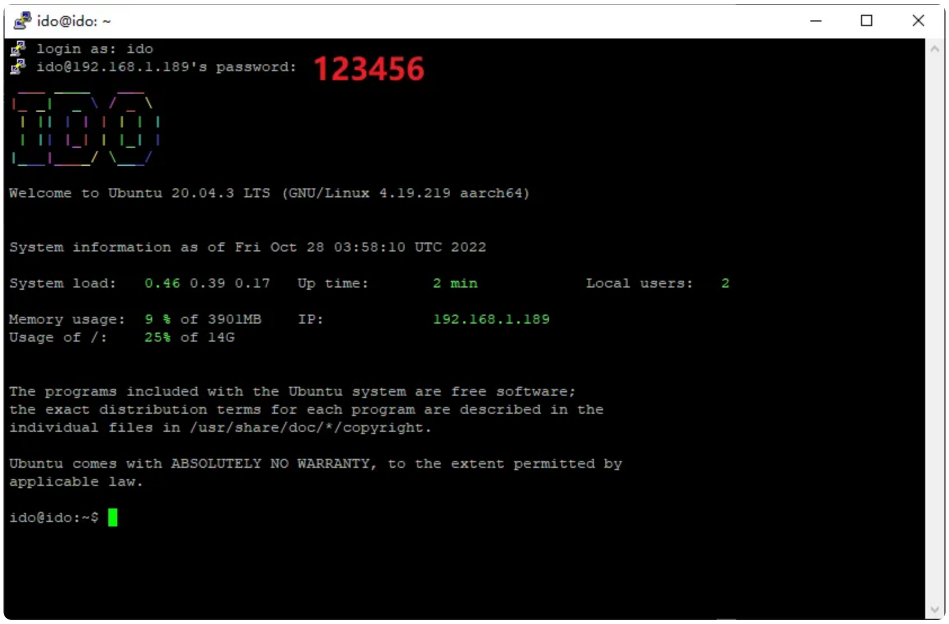
ADB调试



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。

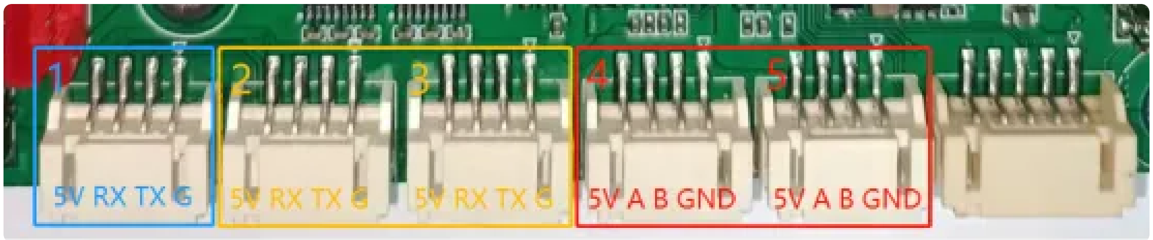
SSH调试

SSH登录账号密码为： **ido @ 123456**。



```
login as: ido
ido@192.168.1.189's password: 123456
 ___ ____   ___  
|_ _|  _ \ / _ \ 
 | || | | | | | |
 | || |_| | |_| |
|___|____/ \___/ 

Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 4.19.219 aarch64)


System information as of Fri Oct 28 03:58:10 UTC 2022

System load:   0.46 0.39 0.17   Up time:       2 min           Local users:   2

Memory usage:  9 % of 3901MB    IP:            192.168.1.189
Usage of /:    25% of 14G


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

ido@ido:~$ 
```

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

使用工具microcom，可以进行发送和接收测试。

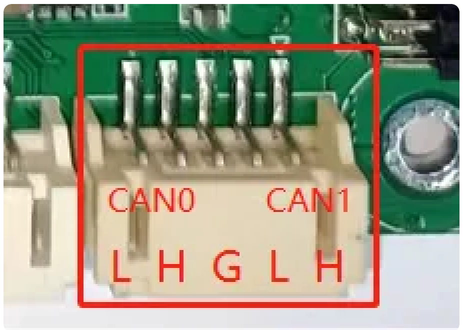
需要先安装microcom工具

```bash
sudo apt-get update
sudo apt-get install microcom
```

```
root@ido:~# microcom -s 115200 -P /dev/ttyS0
[   44.730195] of_dma_request_slave_channel: dma-names property of node '/serial@fdd50000' missing or empty
connected to /dev/ttyS0
Escape character: Ctrl-\
Type the escape character to get to the prompt.
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。

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**

```bash
[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：

```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:/]#
```

重启后，将自动连接上热点：

```bash
[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

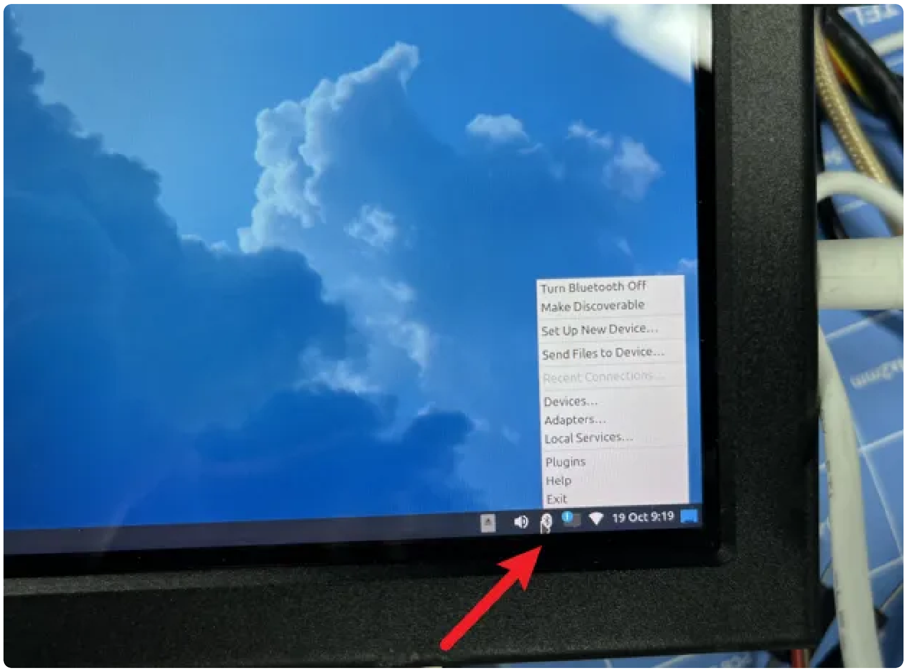
```Bash
[root@RK356X:/]# hciconfig  -a
hci0:   Type: Primary  Bus: UART
        BD Address: F3:7A:FA:A4:5E:22  ACL MTU: 1021:8  SCO MTU: 64:1
        DOWN
        RX bytes:668 acl:0 sco:0 events:34 errors:0
        TX bytes:423 acl:0 sco:0 commands:34 errors:0
        Features: 0xbf 0xfe 0xcf 0xfe 0xdb 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

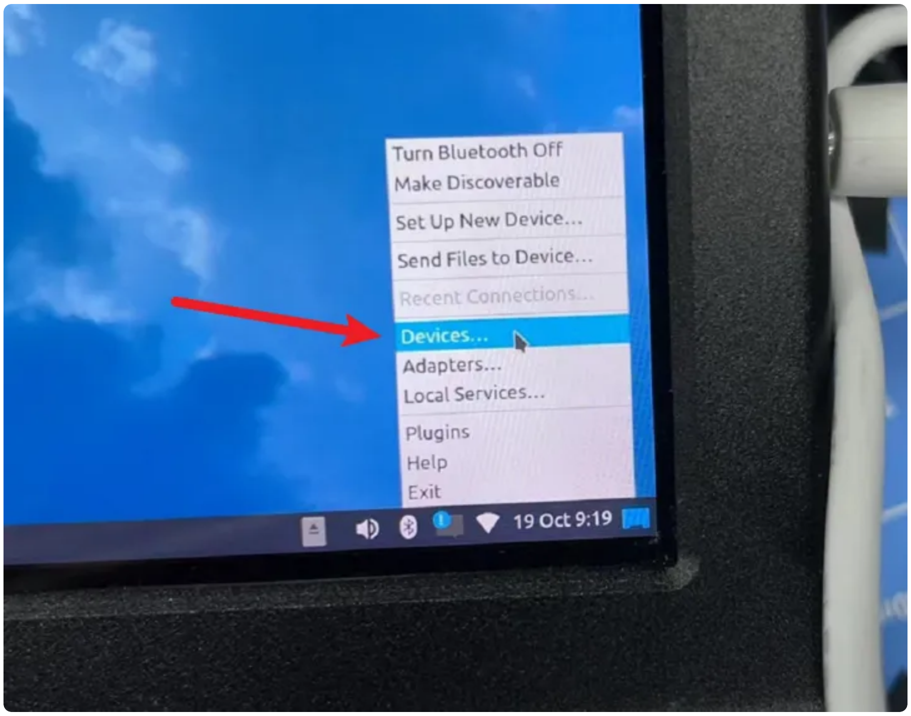
```Bash
[root@RK356X:/]# hciconfig hci0 up
[root@RK356X:/]# hcitool -i hci0 scan
Scanning ...
        94:87:E0:9D:14:12       seeyou
        4C:4F:EE:12:6C:A3       OnePlus 8 Pro
        5C:C5:63:02:31:19       客厅的小米电视
```

也可以通过桌面来控制蓝牙设备

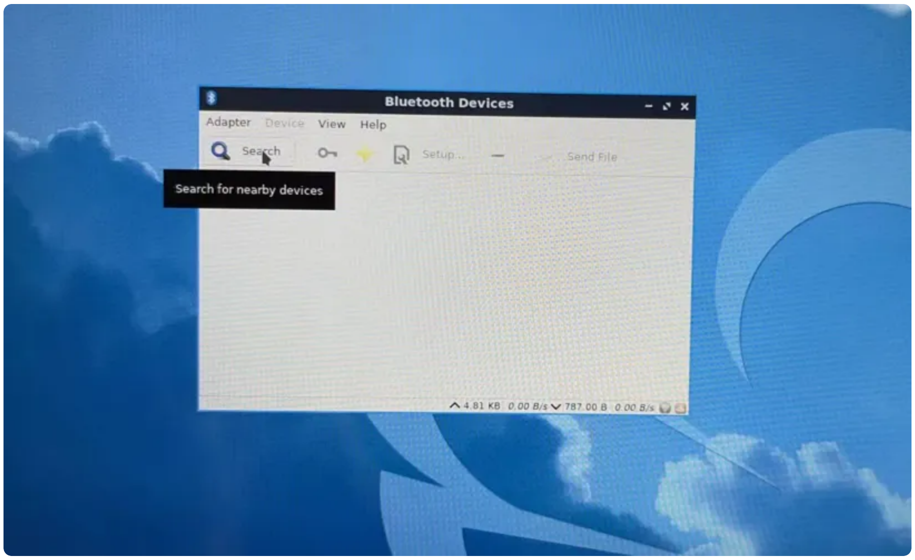
鼠标在桌面右下角的【蓝牙】图标上右键



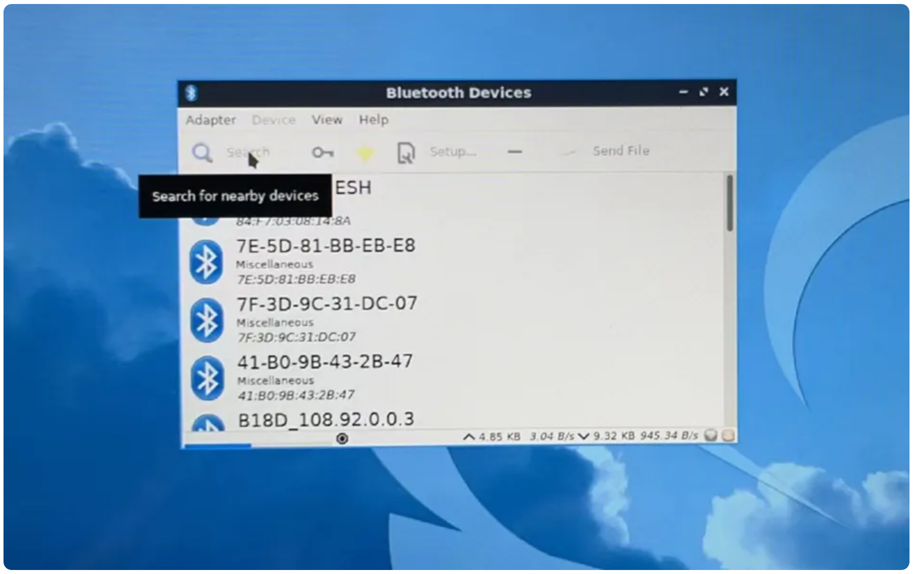
点击"Devices"



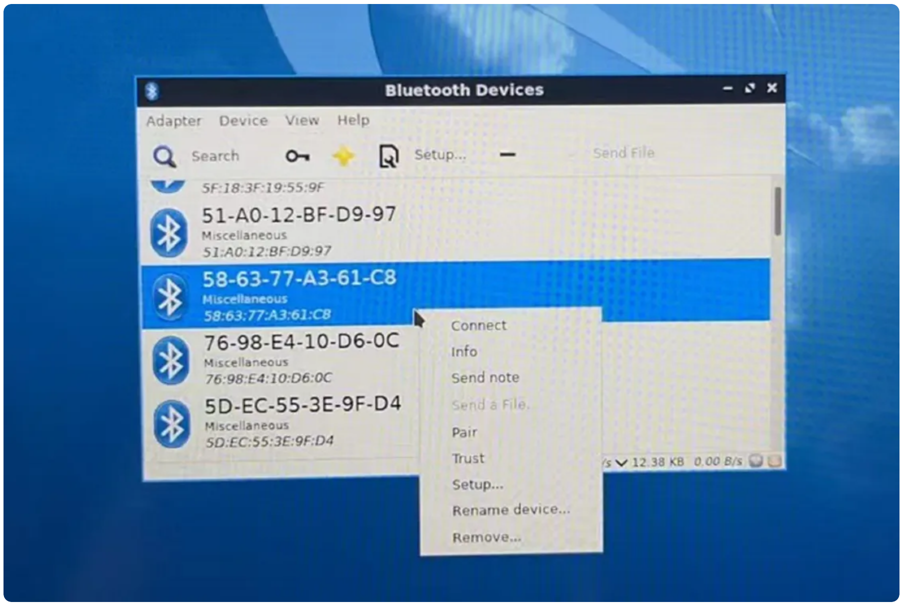
在新弹出的页面中，点击"Search"



列表中可以看到扫描的蓝牙设备

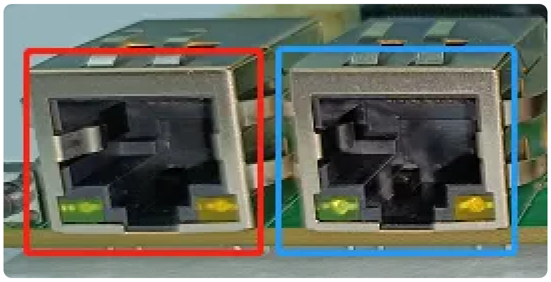


在扫描列表中选中要连接的蓝牙设备，右键–>connect即可连接该设备



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vi /etc/netplan/01-network-manager-all.yaml，在文件中添加如下内容

```bash
network:
      version: 2
      ethernets:
            eth0:     #配置的网卡的名称
                addresses: [192.168.31.215/24]    #配置的静态ip地址和掩码
                dhcp4: no    #关闭DHCP，如果需要打开DHCP则写yes
                dhcp6: no
                gateway4: 192.168.31.1    #网关地址
                nameservers:
                     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

```

注意：间距不是TAB而是空格。

```bash
netplan apply
```

执行以上命令生效。

4G使用

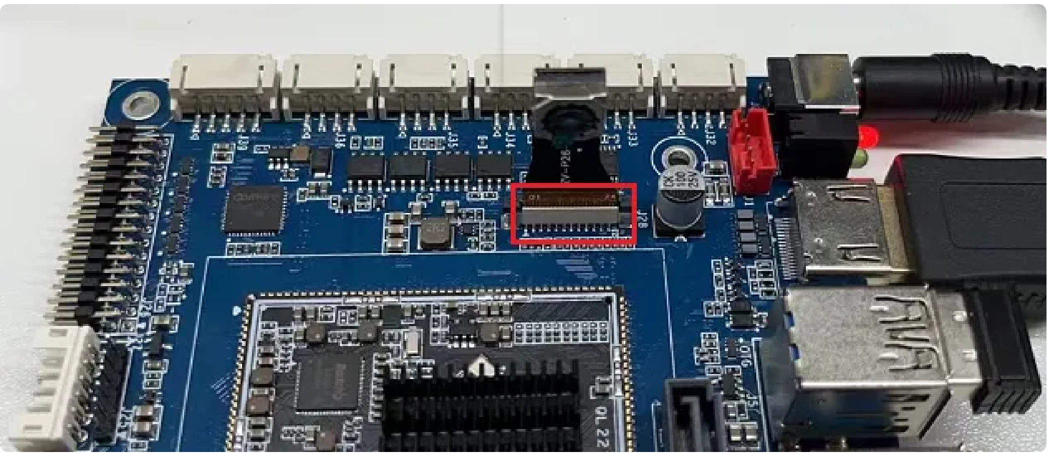
IDO-EVB3568-V1 默认适配EC20模块，系统中已经安装了对应的拨号上网服务。

```bash
/lib/systemd/system/ec20.service
```

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。

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。



10.1 测试

10.1.1 测试摄像头是否存在

```shell
root@ido:~# media-ctl -p -d /dev/media0
...
- entity 67: rockchip-csi2-dphy0 (2 pads, 2 links)
             type V4L2 subdev subtype Unknown flags 0
             device node name /dev/v4l-subdev2
        pad0: Sink
                [fmt:SBGGR10_1X10/2592x1944@10000/150000 field:none]
                <- "m00_b_ov5648 2-0036":0 [ENABLED]
        pad1: Source
                [fmt:SBGGR10_1X10/2592x1944@10000/150000 field:none]
                -> "rkisp-csi-subdev":0 [ENABLED]

- entity 70: m00_b_ov5648 2-0036 (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]
root@ido:~#
```

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

10.1.2抓取视频流

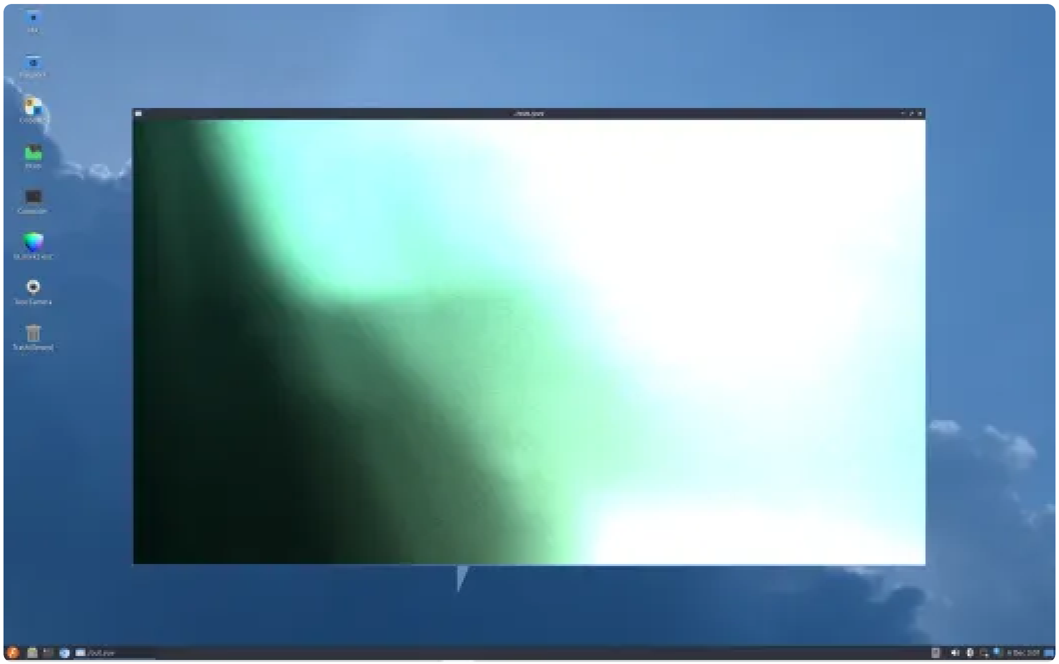
使用v4l2-ctl工具可以抓取摄像头的视频数据流。

```
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
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)
                VIDIOC_STREAMON returned 0 (Success)
cap dqbuf: 0 seq:      1 bytesused: 3110400 ts: 1384.549991 (ts-monotonic, ts-src-eof)
cap dqbuf: 1 seq:      2 bytesused: 3110400 ts: 1384.616490 delta: 66.499 ms (ts-monotonic, ts-src-eof)
cap dqbuf: 2 seq:      3 bytesused: 3110400 ts: 1384.682975 delta: 66.485 ms (ts-monotonic, ts-src-eof)
cap dqbuf: 3 seq:      4 bytesused: 3110400 ts: 1384.749486 delta: 66.511 ms (ts-monotonic, ts-src-eof)
cap dqbuf: 0 seq:      5 bytesused: 3110400 ts: 1384.816022 delta: 66.536 ms fps: 15.04 (ts-monotonic, ts-src-eof)
cap dqbuf: 1 seq:      6 bytesused: 3110400 ts: 1384.882509 delta: 66.487 ms fps: 15.04 (ts-monotonic, ts-src-eof)
cap dqbuf: 2 seq:      7 bytesused: 3110400 ts: 1384.949025 delta: 66.516 ms fps: 15.04 (ts-monotonic, ts-src-eof)
cap dqbuf: 3 seq:       8 bytesused: 3110400 ts: 1385.015545 delta: 66.520 ms fps: 15.04 (ts-monotonic, ts-src-eof)
cap dqbuf: 0 seq:       9 bytesused: 3110400 ts: 1385.082051 delta: 66.506 ms fps: 15.04 (ts-monotonic, ts-src-eof)
cap dqbuf: 1 seq:      10 bytesused: 3110400 ts: 1385.148567 delta: 66.516 ms fps: 15.04 (ts-monotonic, ts-src-eof)
cap dqbuf: 2 seq:      11 bytesused: 3110400 ts: 1385.215079 delta: 66.512 ms fps: 15.04 (ts-monotonic, ts-src-eof)
cap dqbuf: 3 seq:      12 bytesused: 3110400 ts: 1385.281594 delta: 66.515 ms fps: 15.04 (ts-monotonic, ts-src-eof)
cap dqbuf: 0 seq:      13 bytesused: 3110400 ts: 1385.348115 delta: 66.521 ms fps: 15.04 (ts-monotonic, ts-src-eof)
cap dqbuf: 1 seq:      14 bytesused: 3110400 ts: 1385.414669 delta: 66.554 ms fps: 15.03 (ts-monotonic, ts-src-eof)
cap dqbuf: 2 seq:      15 bytesused: 3110400 ts: 1385.481133 delta: 66.464 ms fps: 15.04 (ts-monotonic, ts-src-eof)
cap dqbuf: 3 seq:      16 bytesused: 3110400 ts: 1385.547656 delta: 66.523 ms fps: 15.04 (ts-monotonic, ts-src-eof)
cap dqbuf: 0 seq:      17 bytesused: 3110400 ts: 1385.614172 delta: 66.516 ms fps: 15.04 (ts-monotonic, ts-src-eof)
cap dqbuf: 1 seq:      18 bytesused: 3110400 ts: 1385.680680 delta: 66.508 ms fps: 15.04 (ts-monotonic, ts-src-eof)
cap dqbuf: 2 seq:      19 bytesused: 3110400 ts: 1385.747241 delta: 66.561 ms fps: 15.03 (ts-monotonic, ts-src-eof)
cap dqbuf: 3 seq:      20 bytesused: 3110400 ts: 1385.813714 delta: 66.473 ms fps: 15.03 (ts-monotonic, ts-src-eof)
^C
```

按Ctrl-C停止抓取，视频流保存到文件out.yuv。

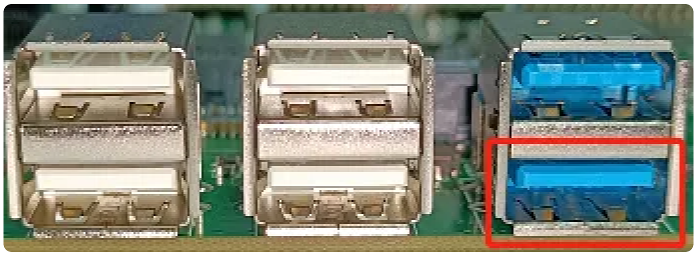
使用ffplay工具播放抓取的视频流：

```
root@ido:~# ffplay -f rawvideo -video_size 1920x1080 -pix_fmt nv12 ./out.yuv
ffplay version 4.2.4-1ubuntu1.0firefly1 Copyright (c) 2003-2020 the FFmpeg developers
  built with gcc 9 (Ubuntu 9.3.0-17ubuntu1~20.04)
  configuration: --prefix=/usr --extra-version=1ubuntu1.0firefly1 --toolchain=hardened --libdir=/usr/lib/aarch64-linux-gnu --incdir=/usr/include/aarch64-linux-gnu --arch=arm64 --enable-gpl --disable-stripping --enable-avresample --disable-filter=resample --enable-avisynth --enable-gnutls --enable-ladspa --enable-libaom --enable-libass --enable-libbluray --enable-libbs2b --enable-libcaca --enable-libcdio --enable-libcodec2 --enable-libflite --enable-libfontconfig --enable-libfreetype --enable-libfribidi --enable-libgme --enable-libgsm --enable-libjack --enable-libmp3lame --enable-libmysofa --enable-libopenjpeg --enable-libopenmpt --enable-libopus --enable-libpulse --enable-librsvg --enable-librubberband --enable-libshine --enable-libsnappy --enable-libsoxr --enable-libspeex --enable-libssh --enable-libtheora --enable-libtwolame --enable-libvidstab --enable-libvorbis --enable-libvpx --enable-libwavpack --enable-libwebp --enable-libx265 --enable-libxml2 --enable-libxvid --enable-libzmq --enable-libzvbi --enable-lv2 --enable-omx --enable-openal --enable-opencl --enable-opengl --enable-sdl2 --enable-libdc1394 --enable-libdrm --enable-libiec61883 --enable-chromaprint --enable-frei0r --enable-libx264 --enable-libdrm --enable-librga --enable-rkmpp --enable-version3 --disable-libopenh264 --disable-vaapi --disable-vdpau --disable-decoder=h264_v4l2m2m --disable-decoder=vp8_v4l2m2m --disable-decoder=mpeg2_v4l2m2m --disable-decoder=mpeg4_v4l2m2m --disable-muxer='ac3,eac3,mlp,truehd' --disable-encoder='ac3_fixed,ac3,mlp,spdif,truehd' --disable-demuxer='ac3,eac3,mlp,truehd,dts,dtshd' --disable-parser='aac,ac3,mlp' --disable-decoder='ac3,eac3,mlp,dolby_e' --enable-shared --disable-doc
  libavutil      56. 31.100 / 56. 31.100
  libavcodec     58. 54.100 / 58. 54.100
  libavformat    58. 29.100 / 58. 29.100
  libavdevice    58.  8.100 / 58.  8.100
  libavfilter     7. 57.100 /  7. 57.100
  libavresample   4.  0.  0 /  4.  0.  0
  libswscale      5.  5.100 /  5.  5.100
  libswresample   3.  5.100 /  3.  5.100
  libpostproc    55.  5.100 / 55.  5.100
Option -pix_fmt is deprecated, use -pixel_format.
libGL error: failed to create dri screen
libGL error: failed to load driver: rockchip
libGL error: failed to create dri screen
libGL error: failed to load driver: rockchip
[rawvideo @ 0x7f3c000ba0] Estimating duration from bitrate, this may be inaccurate
Input #0, rawvideo, from './out.yuv':
  Duration: 00:00:04.00, start: 0.000000, bitrate: 622075 kb/s
    Stream #0:0: Video: rawvideo (NV12 / 0x3231564E), nv12, 1920x1080, 622080 kb/s, 25 tbr, 25 tbn, 25 tbc

```



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：

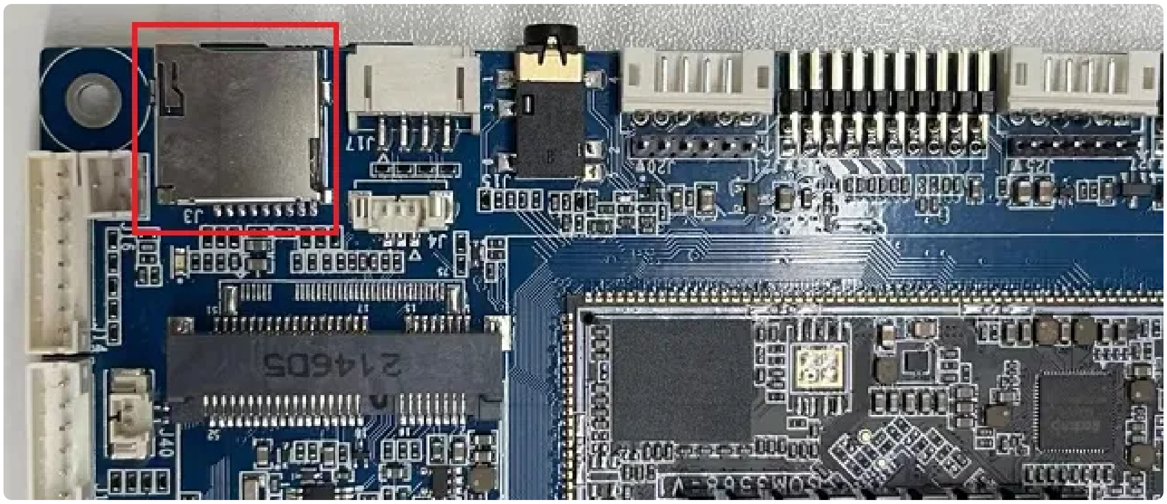
```java
host:
echo host > /sys/devices/platform/fe8a0000.usb2-phy/otg_mode
device:
echo peripheral > /sys/devices/platform/fe8a0000.usb2-phy/otg_mode
```

当接入U盘设备时，默认挂载到/media/ido/目录下

```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)
...
```

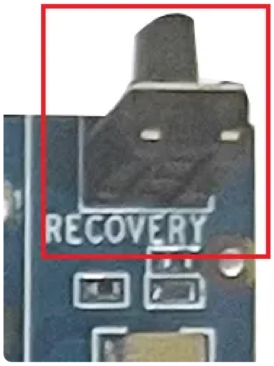
开机启动程序

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)**。

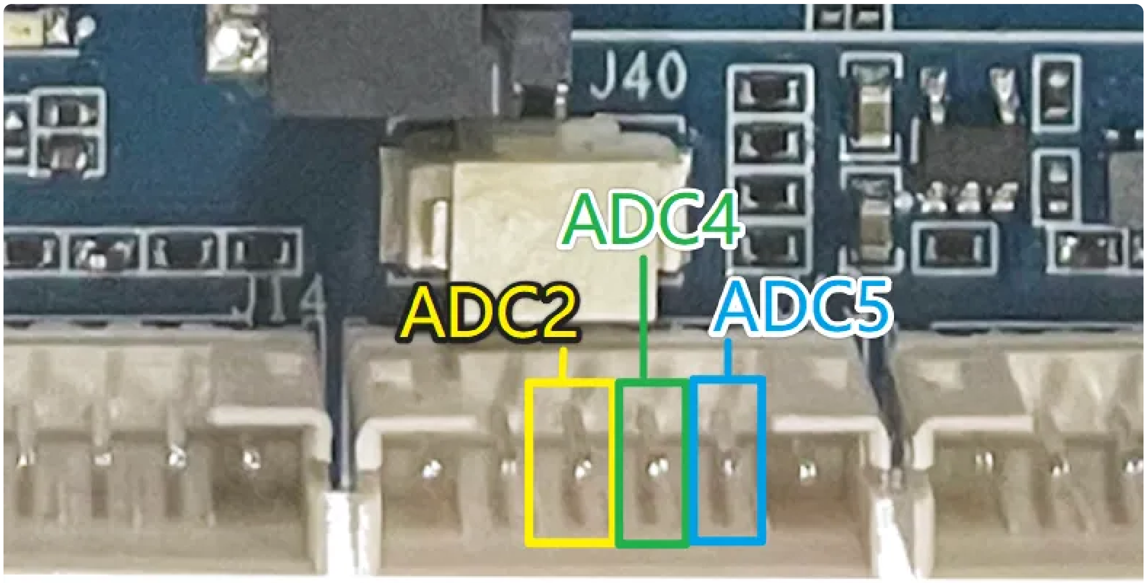
使用evtest进行测试：

```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时间读取和同步

系统时间读取和设置

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

rtc时间设置

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

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

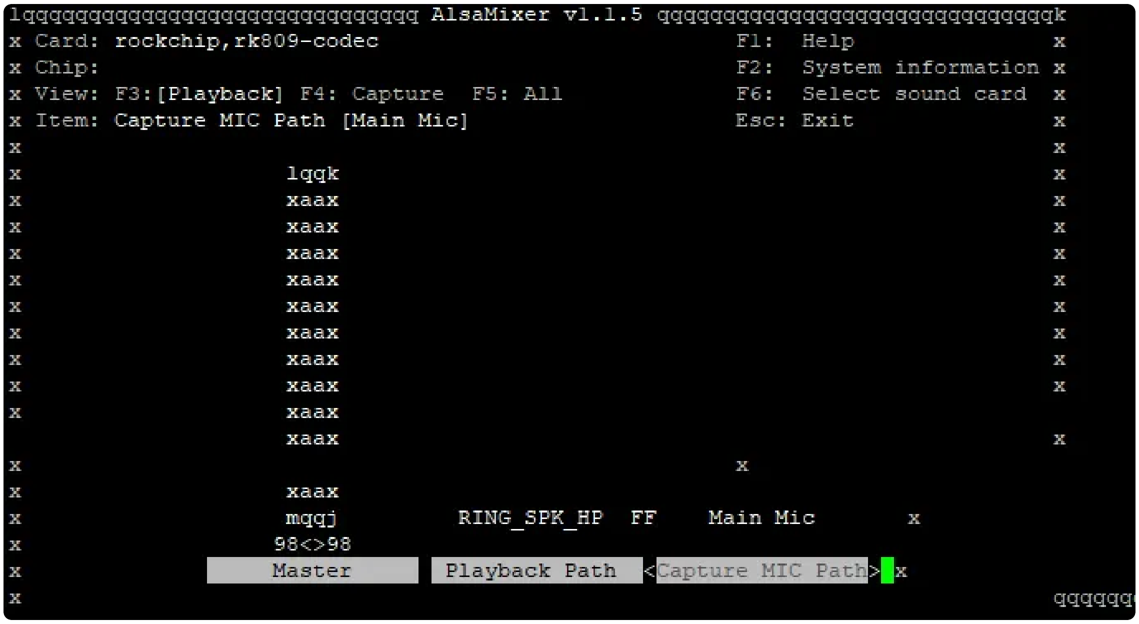
```bash
[root@RK356X:/]# aplay  /etc/bsa_file/8k16bpsStereo.wav
Playing WAVE '/etc/bsa_file/8k16bpsStereo.wav' : Signed 16 bit Little Endian, Rate 8000 Hz, Stereo
```

录音

打开mic通道

```bash
alsamixer
```

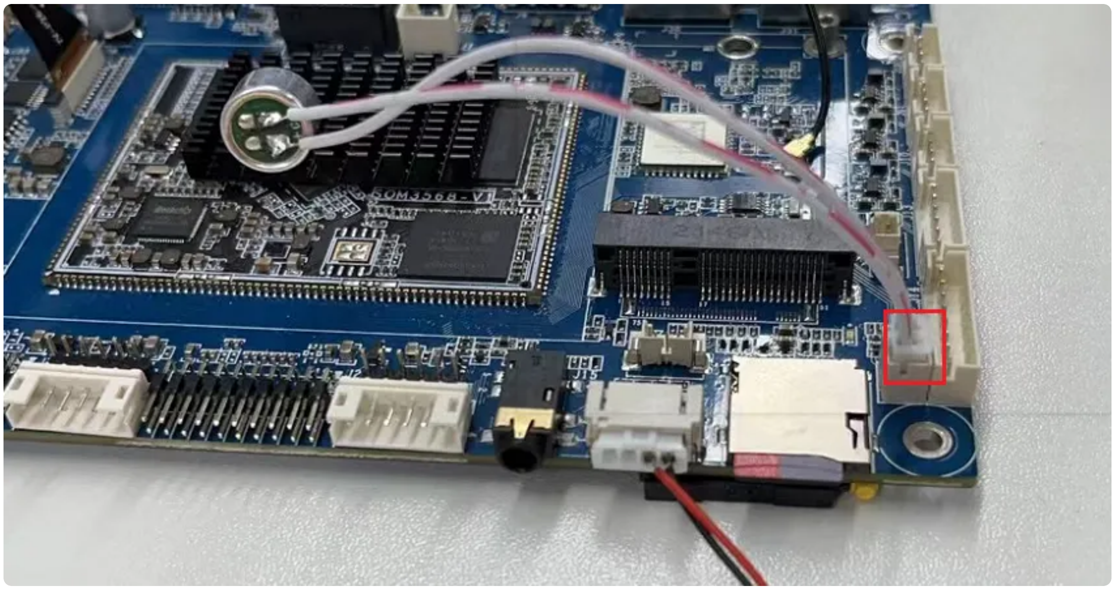
Capture MIC Path选择**Main Mic**



```
lqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqq AlsaMixer v1.1.5 qqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqk
x Card: rockchip,rk809-codec                          F1:  Help                   x
x Chip:                                               F2:  System information     x
x View: F3:[Playback] F4: Capture  F5: All            F6:  Select sound card      x
x Item: Capture MIC Path [Main Mic]                   Esc: Exit                   x
x                                                                                 x
x                   lqqk                                                          x
x                   xaax                                                          x
x                   xaax                                                          x
x                   xaax                                                          x
x                   xaax                                                          x
x                   xaax                                                          x
x                   xaax                                                          x
x                   xaax                                                          x
x                   xaax                                                          x
x                   xaax
                    xaax                                                          x
x                                                     x
x                   xaax
x                   mqqj         RING_SPK_HP   FF     Main Mic        x
x                  98<>98
x                  Master        Playback Path  <Capture MIC Path> x
x                                                                                 qqqqqq
```

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。

```bash
[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...
```

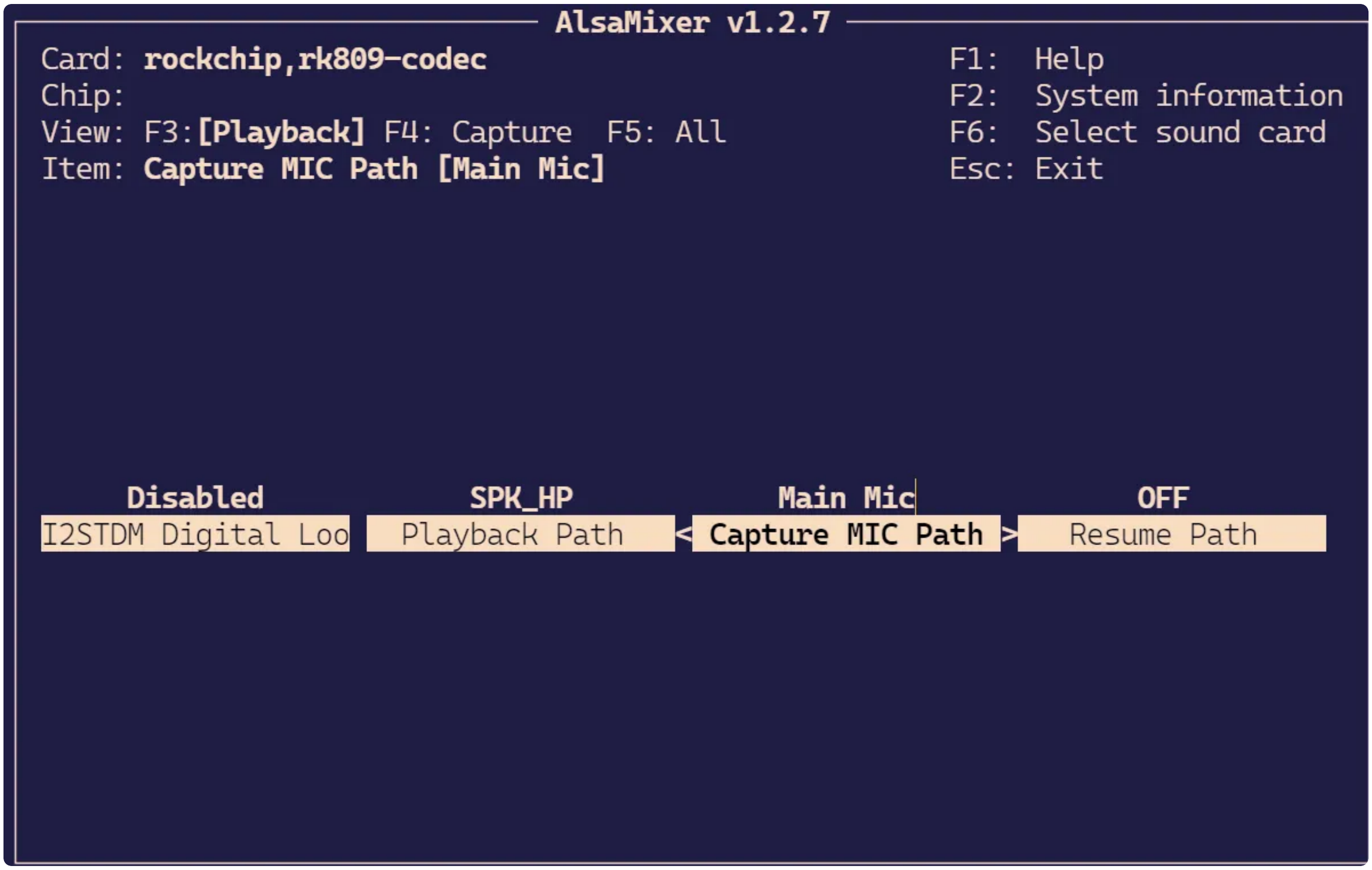
播放录音

```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
```



```
                              AlsaMixer v1.2.7
Card: rockchip,rk809-codec                          F1:  Help
Chip:                                               F2:  System information
View: F3:[Playback] F4: Capture  F5: All            F6:  Select sound card
Item: Capture MIC Path [Main Mic]                   Esc: Exit



     Disabled            SPK_HP            Main Mic             OFF
I2STDM Digital Loo   Playback Path   < Capture MIC Path >   Resume Path
```

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

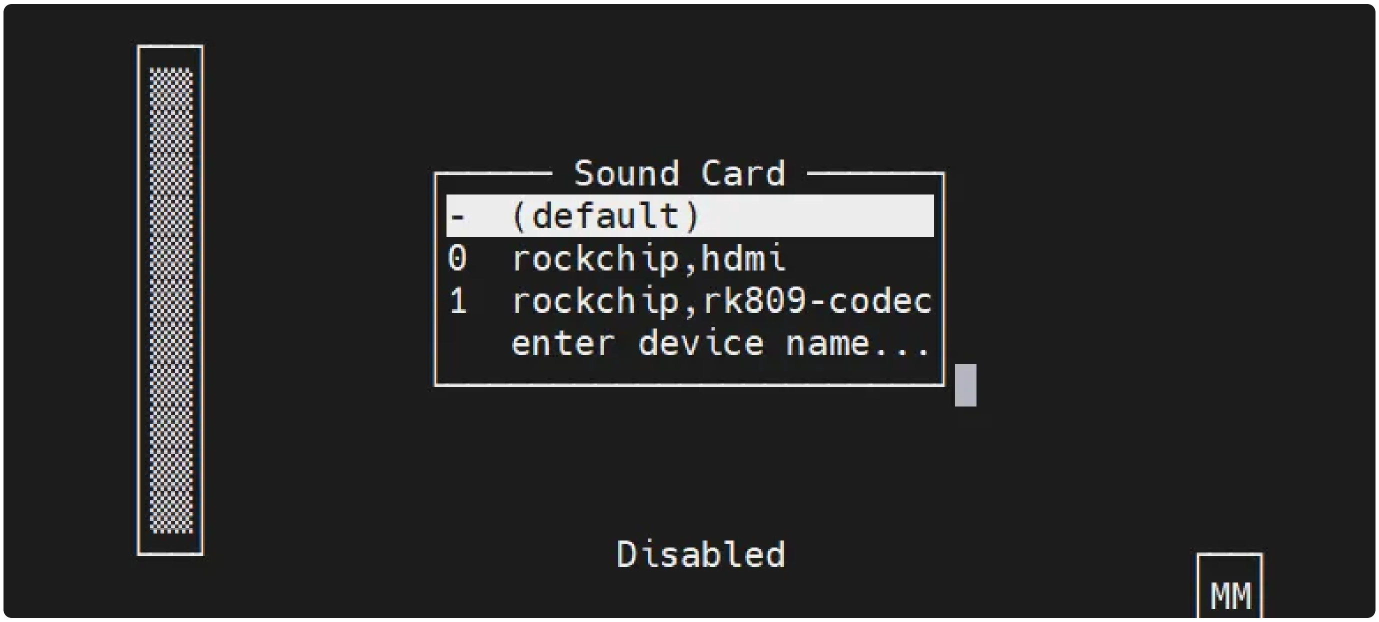
```bash
aplay -D plug:spk_c0 /usr/share/sounds/alsa/Rear_Center.wav
```

音量调节：

执行alsamixer进入图形界面进行调试

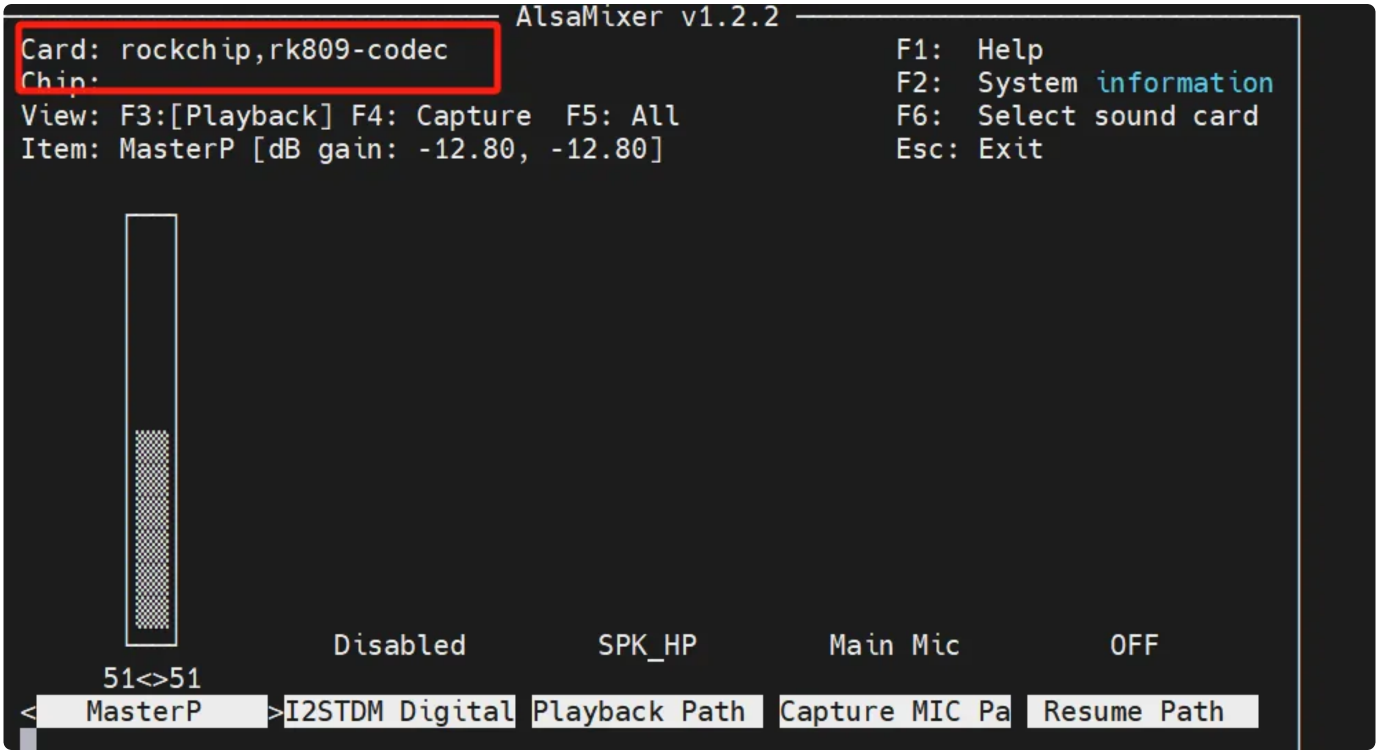
```bash
alsamixer
```

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



```
                 Sound Card
            -   (default)
            0   rockchip,hdmi
            1   rockchip,rk809-codec
                enter device name...

                 Disabled
                                            MM
```

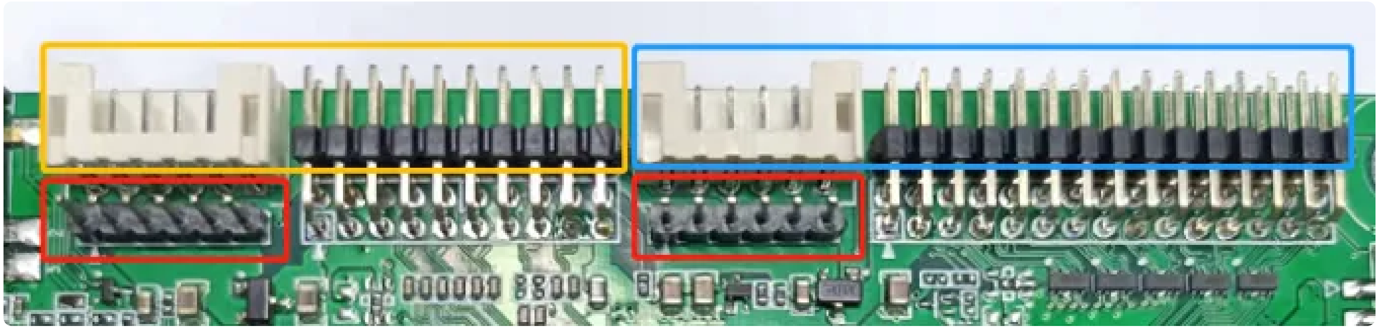
这里可以调节音量为51



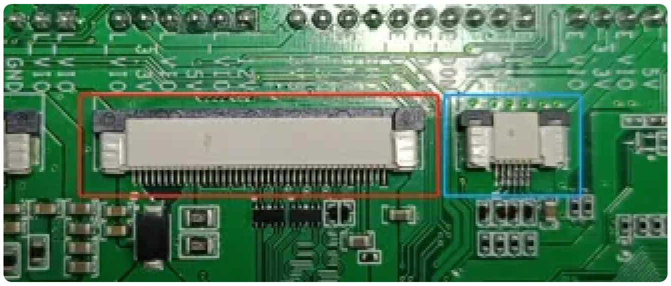
```
──────────────────────── AlsaMixer v1.2.2 ──────────────────────
Card: rockchip,rk809-codec                    F1:  Help
Chip:                                         F2:  System information
View: F3:[Playback] F4: Capture  F5: All      F6:  Select sound card
Item: MasterP [dB gain: -12.80, -12.80]       Esc: Exit

                Disabled      SPK_HP       Main Mic       OFF
    51<>51
<   MasterP   >I2STDM Digital Playback Path Capture MIC Pa Resume Path
```

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
```