

IDO-EVB3568-V1 Linux4.19使用手册

IDO-EVB3568-V1 Ubuntu 系统

1 调试

1.1 串口调试

1.2 ADB调试

1.3 SSH调试

2 串口测试

3 CAN测试

4 WiFi使用

5 蓝牙使用

6 以太网使用

6.1 静态IP设置

7 4G使用

8 摄像头使用

8.1 测试

8.1.1 测试摄像头是否存在

8.1.2 抓取视频流

9 U盘

10 SD卡

11 开机启动程序

12 按键

13 ADC

13.1 ADC值读取

13.2 ADC电压转换关系

14 时间设置

14.1 RTC时间读取和同步

14.2 NTP时间同步

14.3 时区

15 音频

15.1 Lineout

15.2 耳机

15.3 录音

16 显示屏

16.1 显示屏接口说明

16.2 显示设置

16.2.1 屏幕背光亮度设置

IDO-EVB3568-V1 Buildroot 系统

1 调试

1.1 串口调试

1.2 ADB调试

1.3 SSH调试

2 串口测试

3 CAN测试

4 WiFi使用

5 蓝牙使用

6 以太网使用

6.1 静态IP设置

7 4G

8 摄像头使用

9 U盘

10 SD卡

11 开机启动程序

12 按键

13 ADC

13.1 ADC值读取

13.2 ADC电压转换关系

14 时间设置 RTC

14.1 获取RTC时间

14.2 设置RTC时间

14.2 NTP时间同步

14.3 时区

15 音频

15.1 Lineout

15.2 耳机

15.3 录音

16 显示屏

16.1 显示屏接口说明

16.2 显示设置

16.2.1 屏幕背光亮度设置

17 扩展GPIO

17.1 测试方法

IDO-EVB3568-V1 Debian系统

1 系统用户名及密码

2 调试口使用

2.1 串口调试

2.2 ADB调试

3 串口测试

4 CAN测试

5 WiFi使用

6 蓝牙使用

7 以太网使用

7.1 静态IP设置

7.2 DNS设置

8 4G使用

9 摄像头使用

10 U盘/TF挂载

11 设置开机启动程序

12 按键

13 ADC

14 时间设置

14.1 获取RTC时间

14.2 设置RTC时间

14.3 时区修改

15 音视频

15.1 播放音频的工具

15.2 音量调节

16 显示屏

16.1 显示屏接口说明

16.2 显示设置

16.2.1 屏幕背光亮度设置

17 桌面shell终端

18 浏览器

19 GPU测试



IDO-EVB3568-V1

Linux4.19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	PCBA版本	修订内容	修订	审核	日期
V1.0	V1D	1、创建文档	TWX	IDO	2022/10/26
V1.1	V1D	1、增加SSH调试； 2、补充蓝牙部分的描述；	TWX	IDO	2022/10/28
V1.2	V1D	1、修正文档格式	TWX	IDO	2024/09/06
V1.3	V1D	1、修正文档格式	TWX	IDO	2024/09/14

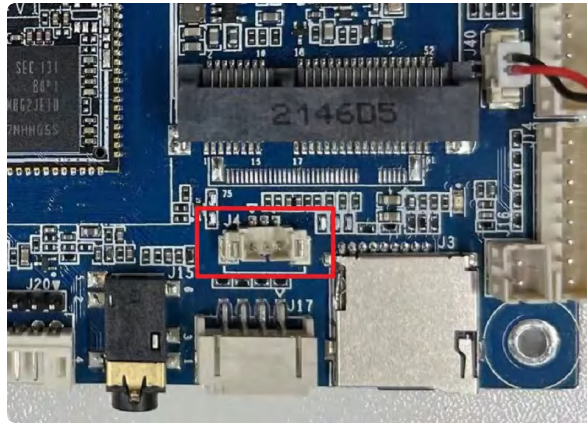
IDO-EVB3568-V1 Ubuntu 系统

1 调试

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

1.1 串口调试

串口调试接口位于J4端口，见下图。请使用配套的USB串口调试工具。



为TTL电平，通信参数为1500000 8 N 1。

1.2 ADB调试



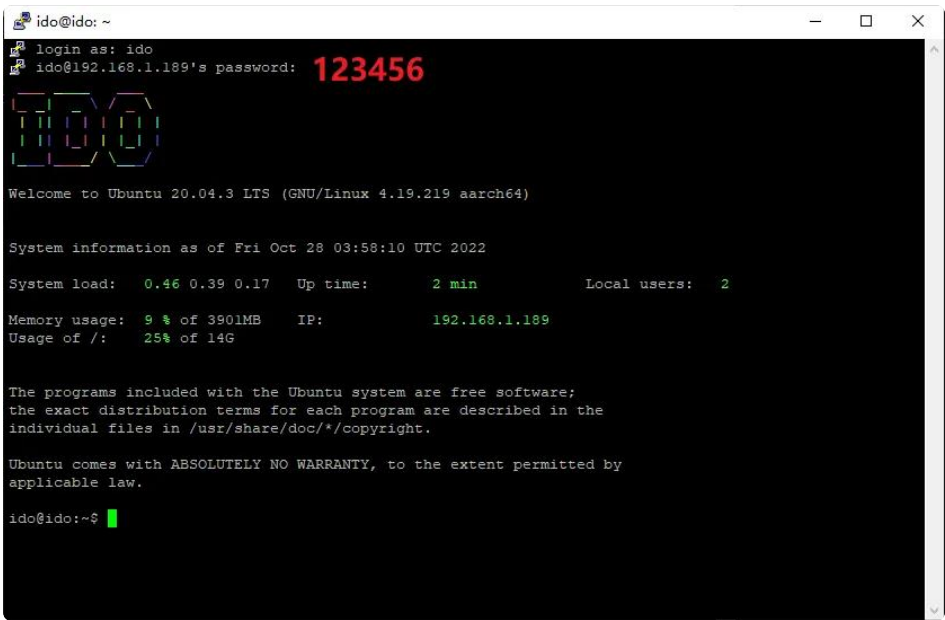
上图红色框内的USB接口为支持OTG模式切换，使用双公头 USB 数据线连接开发板和 PC 端的 USB接口，在PC终端识别到 ADB 设备，即可使用 adb shell 调试。

1.3 SSH调试

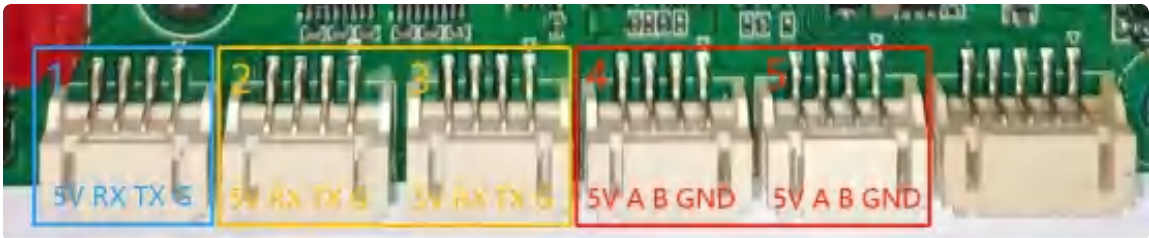
SSH登录账号信息：

账号：ido

密码：123456



2 串口测试



串口接口位置及引脚定义如上图所示，设备节点如下表所示：

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

使用工具microcom，可以进行发送和接收测试。

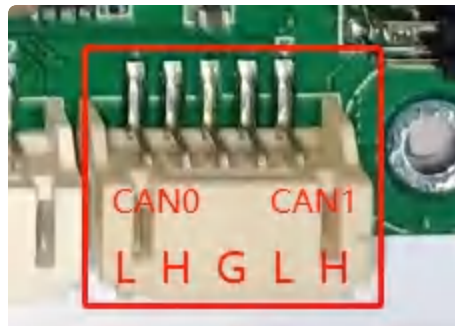
需要先安装microcom工具

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

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

按下键盘，将发送对应的字符；而接收的内容，会显示在终端。

3 CAN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：

```
1  #关闭can0设备
2  ip link set can0 down
3
4  #设置仲裁段1M波特率，数据段3M波特率
5  ip link set can0 type can bitrate 1000000 dbitrade 3000000 fd on
6
7  #打印can0信息
8  ip -details link show can0
9
10 #启动can0
11 ip link set can0 up
12
13 #执行candump，阻塞等待can0接收
14 candump can0
15
16 #canfd格式发送
17 cansend can0 123##1DEADBEEF
18
19 #can格式发送
20 cansend can0 123#1122334455667788
```

4 WiFi使用

在使用 WiFi时连接好WiFi天线，设备节点为wlan0

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

可使用wpa_cli工具连接WiFi热点：

```
1 wpa_cli -i wlan0 set_network 0 ssid '"wifi_ap"'
2 wpa_cli -i wlan0 set_network 0 psk '"12345678"'
3 wpa_cli -i wlan0 enable_network 0
4 dhclient -i wlan0
```

连接成功后，查看wlan0获取的IP地址：

```
1 [root@RK356X:/]# ifconfig wlan0
2 wlan0      Link encap:Ethernet  HWaddr 2C:3B:70:14:17:95
3             inet addr:192.168.1.101  Bcast:192.168.1.255  Mask:255.255.255.0
4             inet6 addr: fe80::220a:b25:4bd:2e3a/64 Scope:Link
5             UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
6             RX packets:26 errors:0 dropped:0 overruns:0 frame:0
7             TX packets:40 errors:0 dropped:0 overruns:0 carrier:0
8             collisions:0 txqueuelen:1000
9             RX bytes:5075 (4.9 KiB)  TX bytes:3913 (3.8 KiB)
10
11 [root@RK356X:/]#
```

5 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

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

蓝牙功能开启后，将产生hci0节点

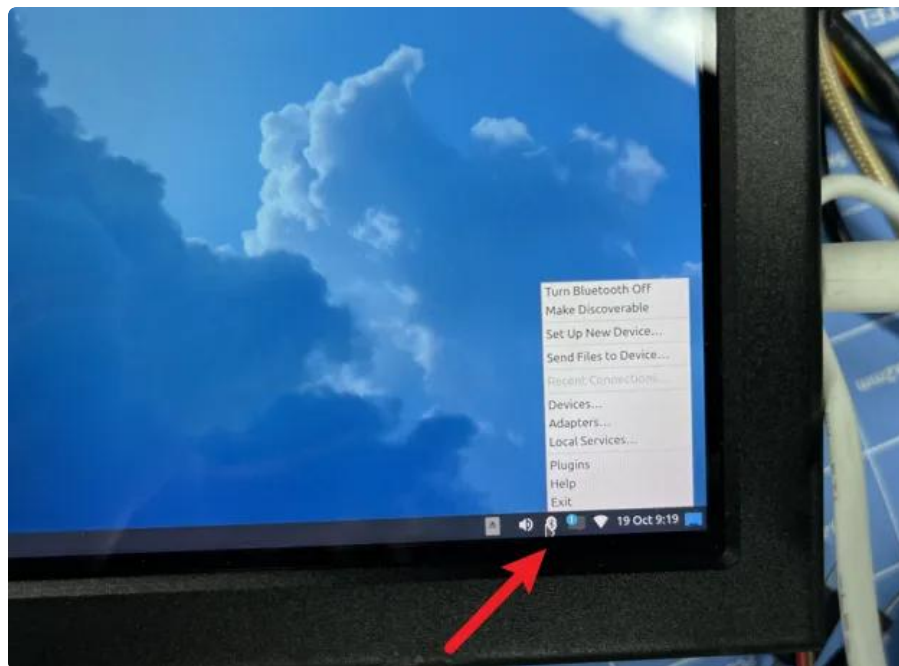
```
▼ Bash |
1 [root@RK356X:/]# hciconfig -a
2 hci0:  Type: Primary  Bus: UART
3       BD Address: F3:7A:FA:A4:5E:22  ACL MTU: 1021:8  SCO MTU: 64:1
4       DOWN
5       RX bytes:668 acl:0 sco:0 events:34 errors:0
6       TX bytes:423 acl:0 sco:0 commands:34 errors:0
7       Features: 0xbf 0xfe 0xcf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

使用hcitool测试蓝牙扫描功能

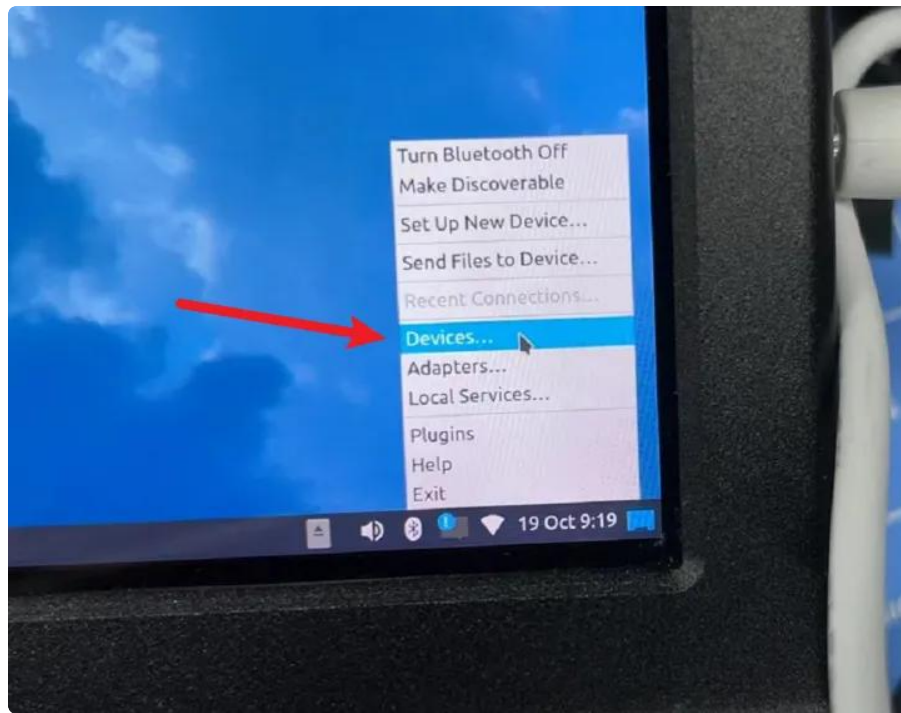
```
▼ Bash |
1 [root@RK356X:/]# hciconfig hci0 up
2 [root@RK356X:/]# hcitool -i hci0 scan
3 Scanning ...
4      94:87:E0:9D:14:12      seeyou
5      4C:4F:EE:12:6C:A3      OnePlus 8 Pro
6      5C:C5:63:02:31:19      客厅的小米电视
```

也可以通过桌面来控制蓝牙设备

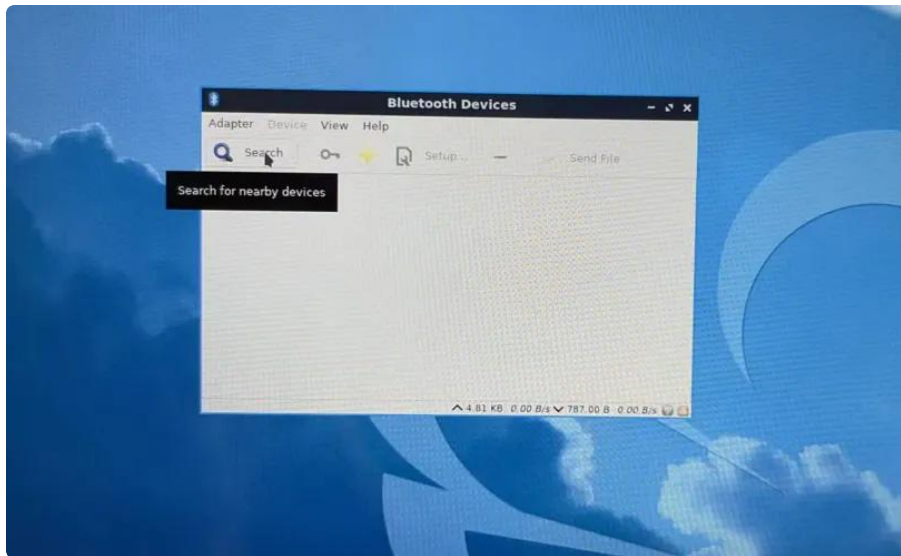
鼠标在桌面右下角的【蓝牙】图标上右键



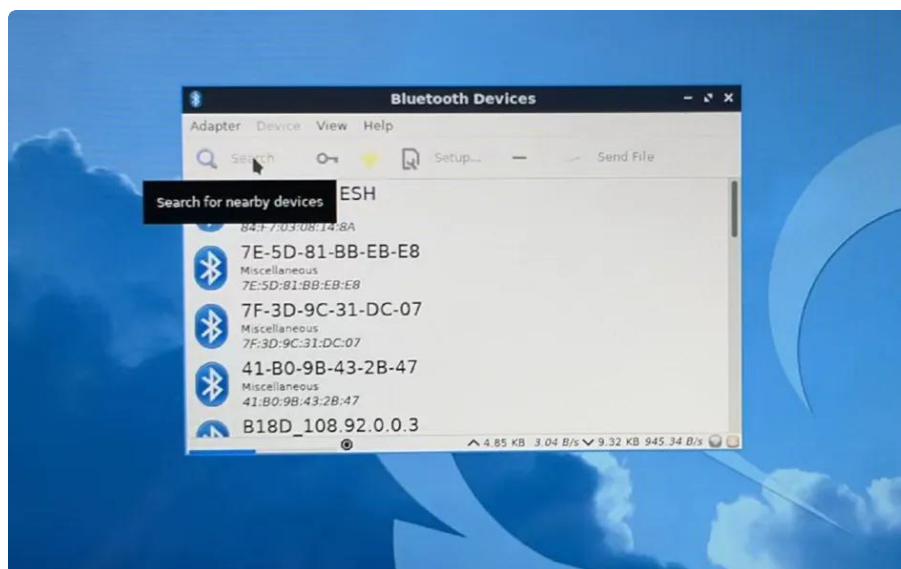
点击"Devices"



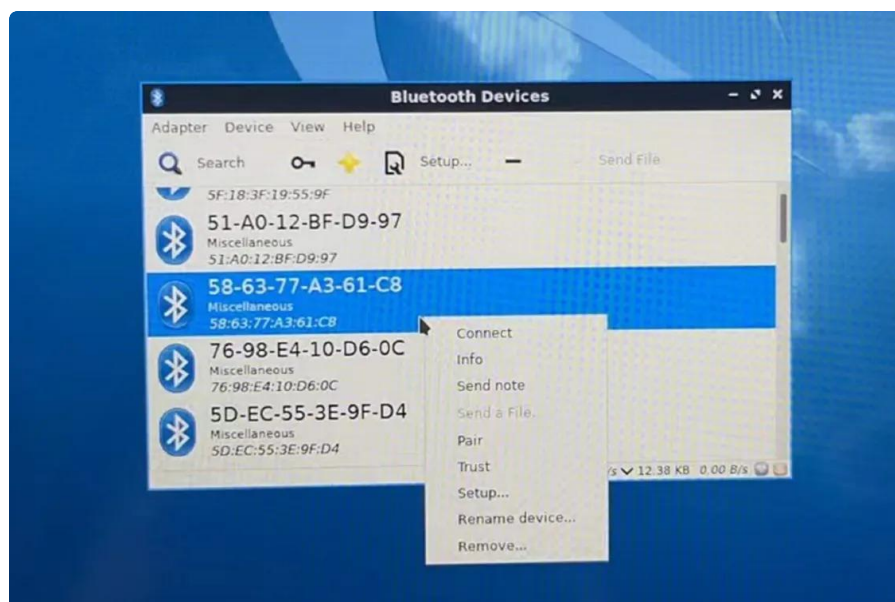
在新弹出的页面中，点击"Search"



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



6 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。

6.1 静态IP设置

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

```
1 network:
2     version: 2
3     ethernets:
4         eth0:      #配置的网卡的名称
5                 addresses: [192.168.31.215/24]    #配置的静态ip地址和掩码
6                 dhcp4: no      #关闭DHCP，如果需要打开DHCP则写yes
7                 dhcp6: no
8                 gateway4: 192.168.31.1    #网关地址
9                 nameservers:
10                    addresses: [192.168.31.1,114.114.114.114] #DNS服务器
11                    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。

```
1 netplan apply
```

执行以上命令生效。

7 4G使用

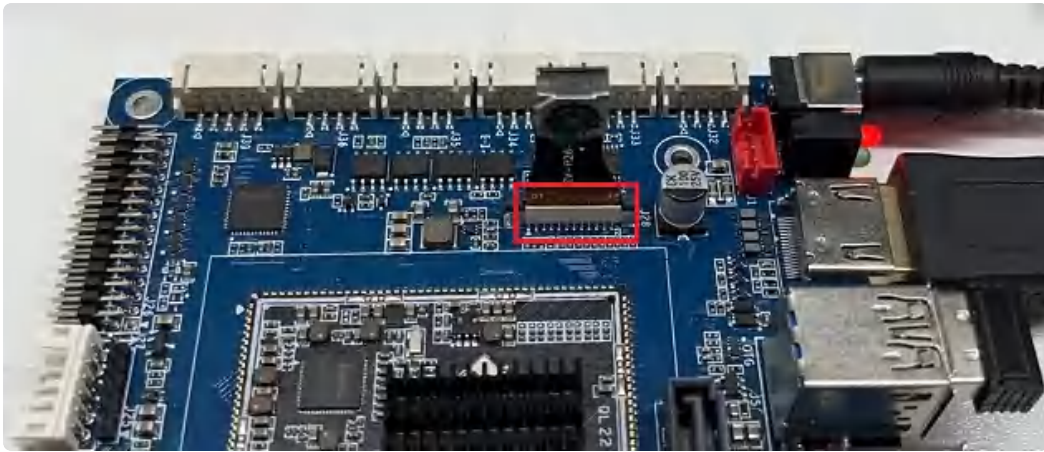
IDO-EVB3568-V1 默认适配EC20模块，系统中已经安装了对应的拨号上网服务，系统启动完成后，执行以下指令拨号：

```
1 systemctl start 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。

8 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。



8.1 测试

8.1.1 测试摄像头是否存在

```
▼ Shell |
1 root@ido:~# media-ctl -p -d /dev/media0
2 ...
3 - entity 67: rockchip-csi2-dphy0 (2 pads, 2 links)
4     type V4L2 subdev subtype Unknown flags 0
5     device node name /dev/v4l-subdev2
6     pad0: Sink
7         [fmt:SBGGR10_1X10/2592x1944@10000/150000 field:none]
8         <- "m00_b_ov5648 2-0036":0 [ENABLED]
9     pad1: Source
10        [fmt:SBGGR10_1X10/2592x1944@10000/150000 field:none]
11        -> "rkisp-csi-subdev":0 [ENABLED]
12
13 - entity 70: m00_b_ov5648 2-0036 (1 pad, 1 link)
14     type V4L2 subdev subtype Sensor flags 0
15     device node name /dev/v4l-subdev3
16     pad0: Source
17        [fmt:SBGGR10_1X10/2592x1944@10000/150000 field:none]
18        -> "rockchip-csi2-dphy0":0 [ENABLED]
19 root@ido:~#
```

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

8.1.2 抓取视频流

使用v4l2-ctl工具可以抓取摄像头的视频数据流。

```

1  root@ido:~# v4l2-ctl --verbose -d /dev/video0 --set-fmt-video=width=1920,height=1080,pixelformat='NV12' --stream-mmap=4 --set-selection=target=crop,flags=0,top=0,left=0,width=1920,height=1080 --stream-to=./out.yuv
2  VIDIOC_QUERYCAP: ok
3  VIDIOC_G_FMT: ok
4  VIDIOC_S_FMT: ok
5  Format Video Capture Multiplanar:
6      Width/Height      : 1920/1080
7      Pixel Format       : 'NV12' (Y/CbCr 4:2:0)
8      Field             : None
9      Number of planes  : 1
10     Flags              :
11     Colorspace         : Default
12     Transfer Function  : Default
13     YCbCr/HSV Encoding: Default
14     Quantization       : Full Range
15     Plane 0            :
16         Bytes per Line : 1920
17         Size Image     : 3110400
18 VIDIOC_G_SELECTION: ok
19 VIDIOC_S_SELECTION: ok
20     VIDIOC_REQBUFS returned 0 (Success)
21     VIDIOC_QUERYBUF returned 0 (Success)
22     VIDIOC_QUERYBUF returned 0 (Success)
23     VIDIOC_QUERYBUF returned 0 (Success)
24     VIDIOC_QUERYBUF returned 0 (Success)
25     VIDIOC_QBUF returned 0 (Success)
26     VIDIOC_QBUF returned 0 (Success)
27     VIDIOC_QBUF returned 0 (Success)
28     VIDIOC_QBUF returned 0 (Success)
29     VIDIOC_STREAMON returned 0 (Success)
30 cap dqbuf: 0 seq:      1 bytesused: 3110400 ts: 1384.549991 (ts-monotonic, ts-src-eof)
31 cap dqbuf: 1 seq:      2 bytesused: 3110400 ts: 1384.616490 delta: 66.499 ms (ts-monotonic, ts-src-eof)
32 cap dqbuf: 2 seq:      3 bytesused: 3110400 ts: 1384.682975 delta: 66.485 ms (ts-monotonic, ts-src-eof)
33 cap dqbuf: 3 seq:      4 bytesused: 3110400 ts: 1384.749486 delta: 66.511 ms (ts-monotonic, ts-src-eof)
34 cap dqbuf: 0 seq:      5 bytesused: 3110400 ts: 1384.816022 delta: 66.536 ms fps: 15.04 (ts-monotonic, ts-src-eof)
35 cap dqbuf: 1 seq:      6 bytesused: 3110400 ts: 1384.882509 delta: 66.487 ms fps: 15.04 (ts-monotonic, ts-src-eof)
36 cap dqbuf: 2 seq:      7 bytesused: 3110400 ts: 1384.949025 delta: 66.516 ms fps: 15.04 (ts-monotonic, ts-src-eof)

```

```

37 cap dqbuf: 3 seq:      8 bytesused: 3110400 ts: 1385.015545 delta: 66.520
   ms fps: 15.04 (ts-monotonic, ts-src-eof)
38 cap dqbuf: 0 seq:      9 bytesused: 3110400 ts: 1385.082051 delta: 66.506
   ms fps: 15.04 (ts-monotonic, ts-src-eof)
39 cap dqbuf: 1 seq:     10 bytesused: 3110400 ts: 1385.148567 delta: 66.516
   ms fps: 15.04 (ts-monotonic, ts-src-eof)
40 cap dqbuf: 2 seq:     11 bytesused: 3110400 ts: 1385.215079 delta: 66.512
   ms fps: 15.04 (ts-monotonic, ts-src-eof)
41 cap dqbuf: 3 seq:     12 bytesused: 3110400 ts: 1385.281594 delta: 66.515
   ms fps: 15.04 (ts-monotonic, ts-src-eof)
42 cap dqbuf: 0 seq:     13 bytesused: 3110400 ts: 1385.348115 delta: 66.521
   ms fps: 15.04 (ts-monotonic, ts-src-eof)
43 cap dqbuf: 1 seq:     14 bytesused: 3110400 ts: 1385.414669 delta: 66.554
   ms fps: 15.03 (ts-monotonic, ts-src-eof)
44 cap dqbuf: 2 seq:     15 bytesused: 3110400 ts: 1385.481133 delta: 66.464
   ms fps: 15.04 (ts-monotonic, ts-src-eof)
45 cap dqbuf: 3 seq:     16 bytesused: 3110400 ts: 1385.547656 delta: 66.523
   ms fps: 15.04 (ts-monotonic, ts-src-eof)
46 cap dqbuf: 0 seq:     17 bytesused: 3110400 ts: 1385.614172 delta: 66.516
   ms fps: 15.04 (ts-monotonic, ts-src-eof)
47 cap dqbuf: 1 seq:     18 bytesused: 3110400 ts: 1385.680680 delta: 66.508
   ms fps: 15.04 (ts-monotonic, ts-src-eof)
48 cap dqbuf: 2 seq:     19 bytesused: 3110400 ts: 1385.747241 delta: 66.561
   ms fps: 15.03 (ts-monotonic, ts-src-eof)
49 cap dqbuf: 3 seq:     20 bytesused: 3110400 ts: 1385.813714 delta: 66.473
   ms fps: 15.03 (ts-monotonic, ts-src-eof)
50 ^C

```

按Ctrl+C停止抓取，视频流保存到文件out.yuv。

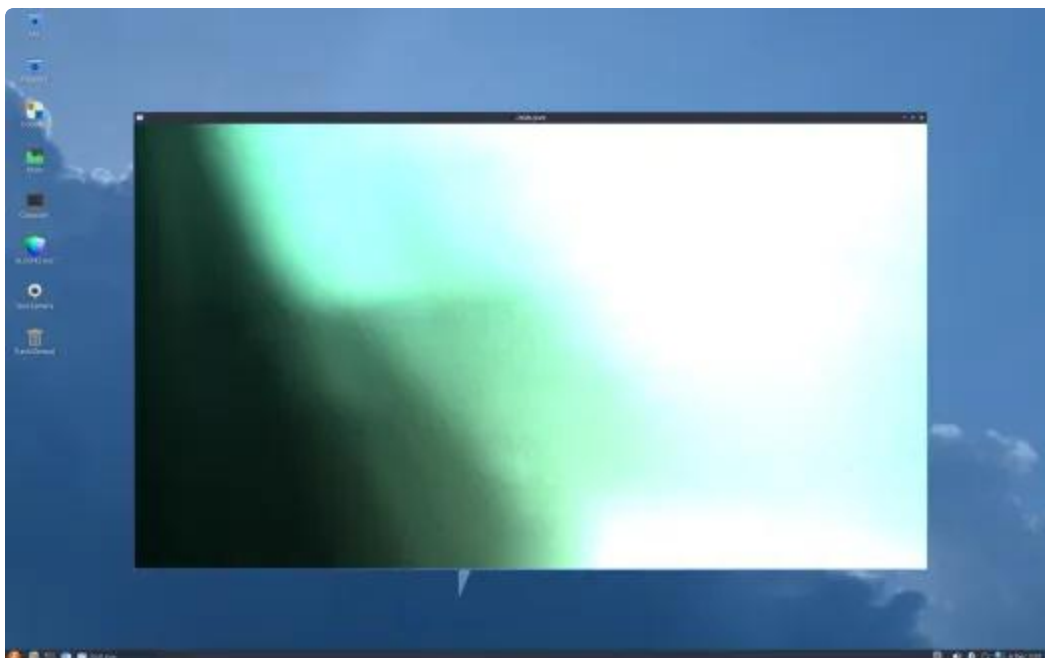
使用FFplay工具播放抓取的视频流：

```

1 root@ido:~# ffplay -f rawvideo -video_size 1920x1080 -pix_fmt nv12 ./out.yuv
2 ffplay version 4.2.4-1ubuntu1.0firefly1 Copyright (c) 2003-2020 the FFmpeg
  developers
3   built with gcc 9 (Ubuntu 9.3.0-17ubuntu1~20.04)
4   configuration: --prefix=/usr --extra-version=1ubuntu1.0firefly1 --toolcha
  in=hardened --libdir=/usr/lib/aarch64-linux-gnu --incdir=/usr/include/aar
  ch64-linux-gnu --arch=arm64 --enable-gpl --disable-stripping --enable-avre
  sample --disable-filter=resample --enable-avisynth --enable-gnutls --enabl
  e-ladspa --enable-libaom --enable-libass --enable-libbluray --enable-libbs
  2b --enable-libcaca --enable-libcdio --enable-libcodec2 --enable-libflite
  --enable-libfontconfig --enable-libfreetype --enable-libfribidi --enable-l
  ibgme --enable-libgsm --enable-libjack --enable-libmp3lame --enable-libmys
  ofa --enable-libopenjpeg --enable-libopenmpt --enable-libopus --enable-lib
  pulse --enable-librsvg --enable-librubberband --enable-libshine --enable-l
  ibsnappy --enable-libsoxr --enable-lbspeex --enable-libssh --enable-libth
  eora --enable-libtwolame --enable-libvidstab --enable-libvorbis --enable-l
  ibvpx --enable-libwavpack --enable-libwebp --enable-libx265 --enable-libxm
  l2 --enable-libxvid --enable-libzmq --enable-libzvbi --enable-lv2 --enable
  -omx --enable-openal --enable-openc1 --enable-opengl --enable-sdl2 --enabl
  e-libdc1394 --enable-libdrm --enable-libiec61883 --enable-chromaprint --enab
  le-frei0r --enable-libx264 --enable-libdrm --enable-librga --enable-rkmp
  p --enable-version3 --disable-libopenh264 --disable-vaapi --disable-udpau
  --disable-decoder=h264_v4l2m2m --disable-decoder=vp8_v4l2m2m --disable-dec
  oder=mpeg2_v4l2m2m --disable-decoder=mpeg4_v4l2m2m --disable-muxer='ac3,ea
  c3,mlp,truehd' --disable-encoder='ac3_fixed,ac3,mlp,spdif,truehd' --disabl
  e-demuxer='ac3,eac3,mlp,truehd,dts,dtsd' --disable-parser='aac,ac3,mlp' --
  -disable-decoder='ac3,eac3,mlp,dolby_e' --enable-shared --disable-doc
5   libavutil      56. 31.100 / 56. 31.100
6   libavcodec     58. 54.100 / 58. 54.100
7   libavformat    58. 29.100 / 58. 29.100
8   libavdevice    58.  8.100 / 58.  8.100
9   libavfilter     7. 57.100 / 7. 57.100
10  libavresample   4.  0.  0 / 4.  0.  0
11  libswscale      5.  5.100 / 5.  5.100
12  libswresample   3.  5.100 / 3.  5.100
13  libpostproc    55.  5.100 / 55.  5.100
14 Option -pix_fmt is deprecated, use -pixel_format.
15 libGL error: failed to create dri screen
16 libGL error: failed to load driver: rockchip
17 libGL error: failed to create dri screen
18 libGL error: failed to load driver: rockchip
19 [rawvideo @ 0x7f3c000ba0] Estimating duration from bitrate, this may be in
  accurate
20 Input #0, rawvideo, from './out.yuv':

```

21 Duration: 00:00:04.00, start: 0.000000, bitrate: 622075 kb/s
22 Stream #0:0: Video: rawvideo (NV12 / 0x3231564E), nv12, 1920x1080, 622
23 080 kb/s, 25 tbr, 25 tbn, 25 tbc



9 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：

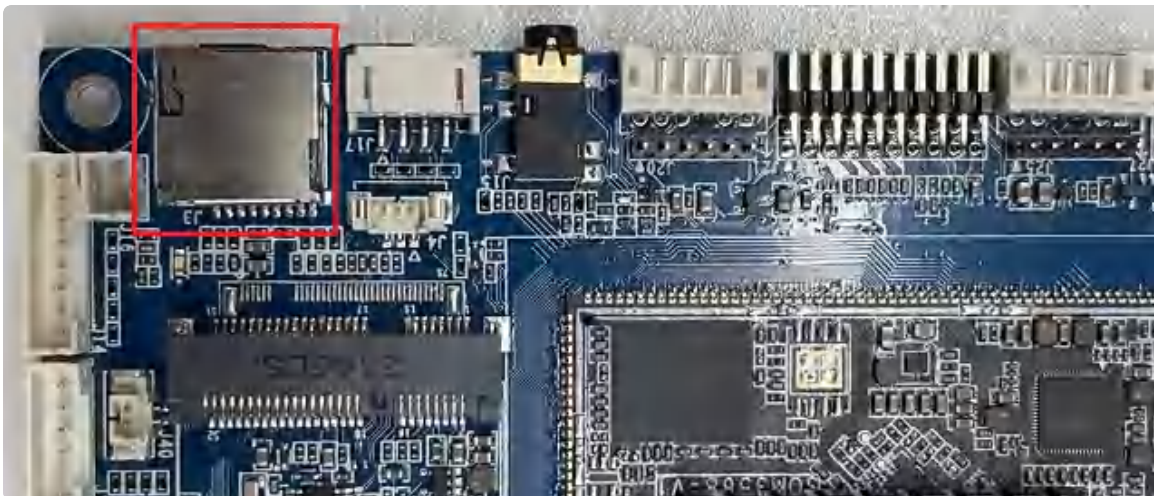
```
1 host:
2 echo host > /sys/devices/platform/fe8a0000.usb2-phy/otg_mode
3 device:
4 echo peripheral > /sys/devices/platform/fe8a0000.usb2-phy/otg_mode
```

当接入U盘设备时，默认挂载到/media/ido/目录下

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

10 SD卡

将SD卡插入到SD卡槽中，将自动挂载到/mnt/sdcard/目录下。



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

11 开机启动程序

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

```
1  [root@RK356X:/]# cat /etc/init.d/S49ntp
2  #!/bin/sh
3
4  NAME=ntpd
5  DAEMON=/usr/sbin/$NAME
6
7  # Gracefully exit if the package has been removed.
8  test -x $DAEMON || exit 0
9
10 # Read config file if it is present.
11 if [ -r /etc/default/$NAME ]
12 then
13     . /etc/default/$NAME
14 fi
15
16 case "$1" in
17     start)
18         printf "Starting $NAME: "
19         start-stop-daemon -S -q -x $DAEMON -- -g
20         [ $? = 0 ] && echo "OK" || echo "FAIL"
21         ;;
22     stop)
23         printf "Stopping $NAME: "
24         start-stop-daemon -K -q -n $NAME
25         [ $? = 0 ] && echo "OK" || echo "FAIL"
26         ;;
27     restart|reload)
28         echo "Restarting $NAME: "
29         $0 stop
30         sleep 1
31         $0 start
32         ;;
33     *)
34         echo "Usage: $0 {start|stop|restart|reload}" >&2
35         exit 1
36         ;;
37 esac
38
39 exit 0
```

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

12 按键

IDO-EVB3568-V1 配置了一个Recovery按键，如下图所示：



在设备断电的情况下，该按键用于烧录固件。在系统正常启动后，则可作为普通按键使用。对应的设备节点为`/dev/input/event2`，键值为`KEY_VOLUMEUP(115)`。

使用`evtest`进行测试，`evtest`工具使用以下命令安装：



Bash |

```
1 [root@RK356X:/]# apt-get install evtest
```

```

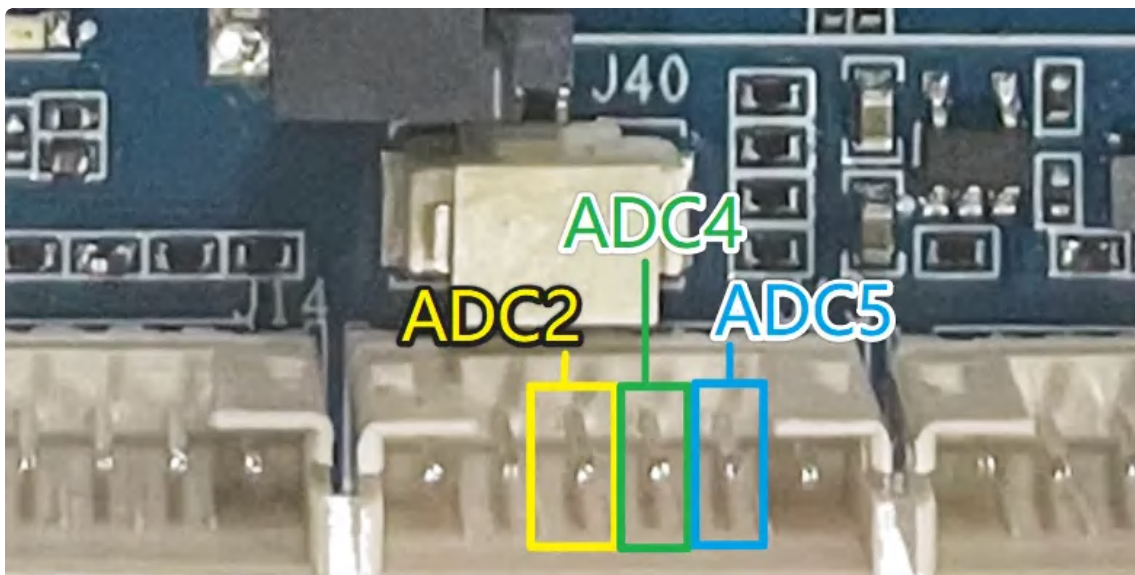
1 [root@RK356X:/]# evtest
2 No device specified, trying to scan all of /dev/input/event*
3 Available devices:
4 /dev/input/event0:      fe6e0030.pwm
5 /dev/input/event1:      rk805 pwrkey
6 /dev/input/event2:      adc-keys
7 /dev/input/event3:      rockchip,rk809-codec Headphones
8 Select the device event number [0-3]: 2
9 Input driver version is 1.0.1
10 Input device ID: bus 0x19 vendor 0x1 product 0x1 version 0x100
11 Input device name: "adc-keys"
12 Supported events:
13   Event type 0 (EV_SYN)
14   Event type 1 (EV_KEY)
15     Event code 114 (KEY_VOLUMEDOWN)
16     Event code 115 (KEY_VOLUMEUP)
17     Event code 139 (KEY_MENU)
18     Event code 158 (KEY_BACK)
19 Properties:
20 Testing ... (interrupt to exit)
21 Event: time 1666752551.345149, type 1 (EV_KEY), code 115 (KEY_VOLUMEUP), v
    alue 1
22 Event: time 1666752551.345149, ----- SYN_REPORT -----
23 Event: time 1666752551.551624, type 1 (EV_KEY), code 115 (KEY_VOLUMEUP), v
    alue 0
24 Event: time 1666752551.551624, ----- SYN_REPORT -----
25 Event: time 1666752552.274980, type 1 (EV_KEY), code 115 (KEY_VOLUMEUP), v
    alue 1
26 Event: time 1666752552.274980, ----- SYN_REPORT -----
27 Event: time 1666752552.688312, type 1 (EV_KEY), code 115 (KEY_VOLUMEUP), v
    alue 0
28 Event: time 1666752552.688312, ----- SYN_REPORT -----
29

```

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。

13 ADC

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



设备节点对应关系如下表：

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

13.1 ADC值读取

▼

Bash

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

13.2 ADC电压转换关系

▼

Bash

1 V=(in_voltage2_raw/1024)*1.8v

假设in_voltage2_raw的值为500，则对应的ADC电压为 $V=(500/1024)*1.8v=0.879v$

14 时间设置

14.1 RTC时间读取和同步

系统时间读取和设置

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

RTC时间设置

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

14.2 NTP时间同步

系统默认开启了NTP服务，连接网络后，将自动同步网络时间。

14.3 时区

查看时区

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

+0000表示在0时区。

设置时区

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

15 音频

使用aplay工具查看声卡设备

```
▼ Bash |
1 [root@RK356X:/]# aplay -l
2 ***** List of PLAYBACK Hardware Devices *****
3 card 0: rockchiprk809co [rockchip,rk809-codec], device 0: fe410000.i2s-rk81
  7-hifi rk817-hifi-0 [fe410000.i2s-rk817-hifi rk817-hifi-0]
4   Subdevices: 1/1
5   Subdevice #0: subdevice #0
```

15.1 Lineout

不插入耳机，使用aplay播放wav音频测试

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

15.2 耳机

插入耳机，使用aplay播放wav音频测试

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

15.3 录音

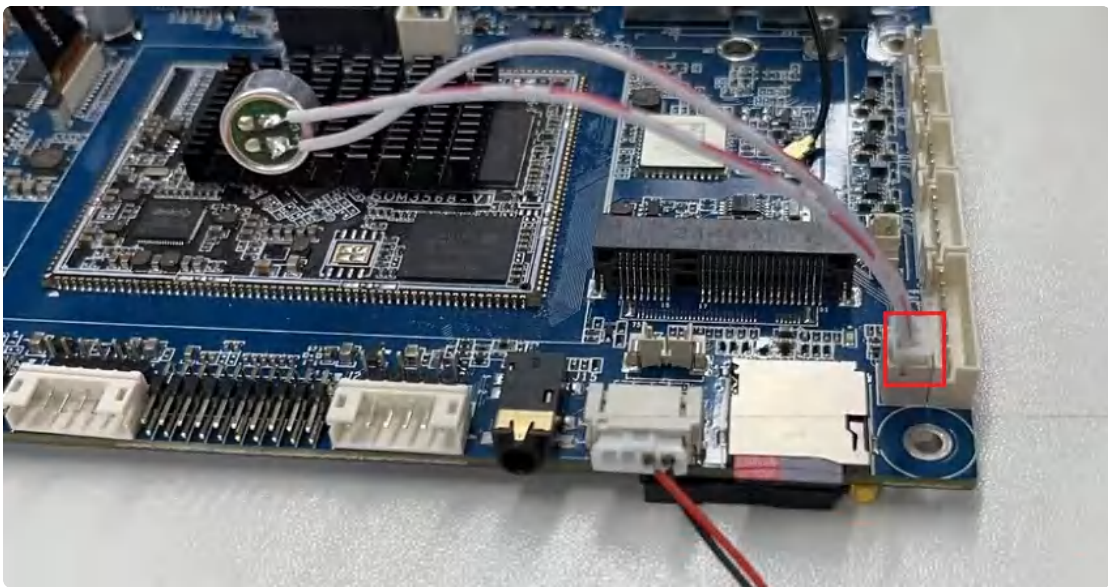
打开MIC通道

```
▼ Bash |
1 alsamixer
```

Capture MIC Path选择Main Mic

[illegible]

参考如下示意图接上麦克风。



使用arecord工具进行录音测试。

```
1 [root@RK356X:/]# arecord -D hw:0,0 -r 48000 -c 2 -f S16_LE test.wav
2 Recording WAVE 'test.wav' : Signed 16 bit Little Endian, Rate 48000 Hz, Stereo
3 ^CAborted by signal Interrupt...
```

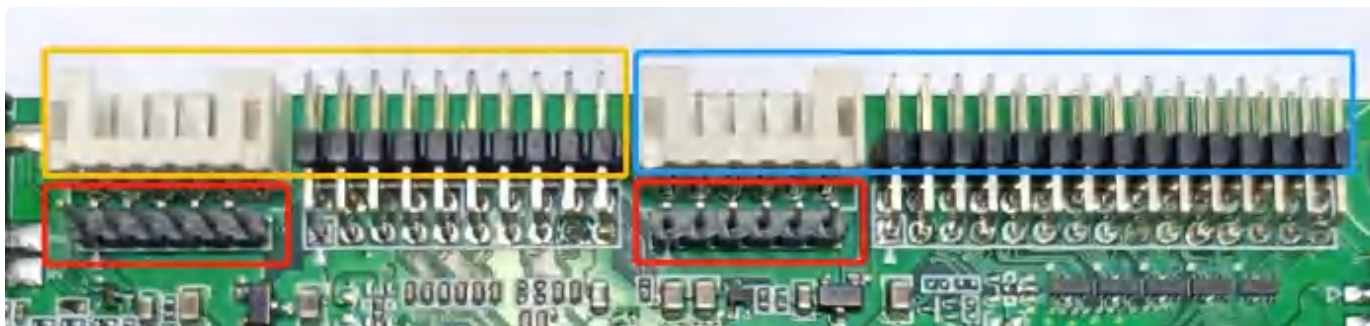
播放录音

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

16 显示屏

16.1 显示屏接口说明

主板各个显示接口如下图所示：



黄色框是eDP屏接口，蓝色框是Dual LVDS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。

16.2 显示设置

16.2.1 屏幕背光亮度设置

- eDP/MIPI屏背光控制

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

设置方法：(支持调节范围 0-255)

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

- Dual LVDS屏幕背光控制

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

设置方法: (支持调节范围 0-255)

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

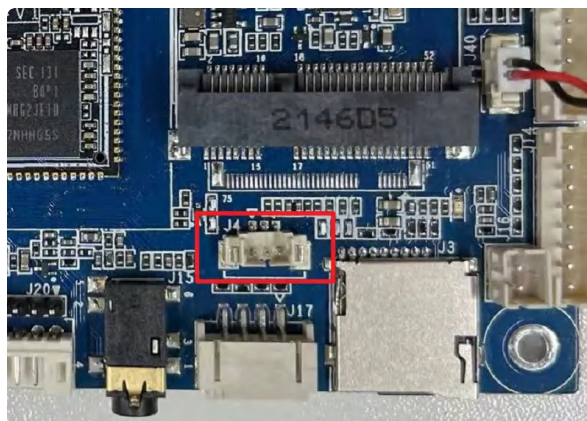
IDO-EVB3568-V1 Buildroot 系统

1 调试

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

1.1 串口调试

串口调试接口位于J4端口, 见下图。请使用配套的USB串口调试工具。



为TTL电平，通信参数为1500000 8 N 1。

1.2 ADB调试



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

```
C:\Users\ronnie>adb shell
* daemon not running. starting it now on port 5037 *
* daemon started successfully *
[root@RK356X:/]# ls
ls
bin          init         media  proc        sdcard      udisk
busybox.config  lib         misc   rockchip_test  sys         userdata
data         lib64        mnt     root         system      usr
dev          linuxrc      oem     run          timestamp   var
etc          lost+found  opt     sbin         tmp         vendor
[root@RK356X:/]#
```

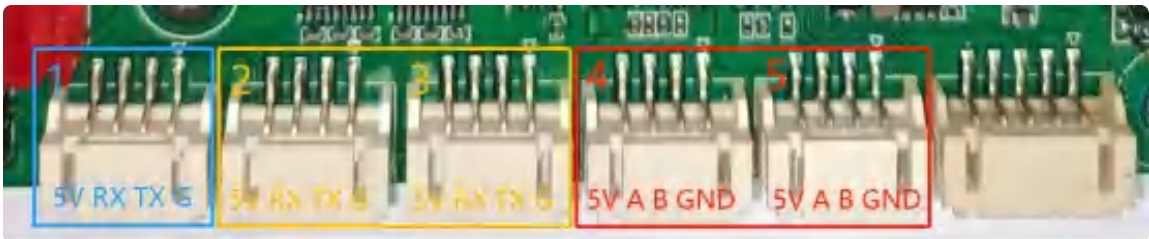
1.3 SSH调试

SSH登录账号信息：

账号：root

密码：rockchip

2 串口测试



串口接口位置及引脚定义如上图所示，设备节点如下表所示：

序号	功能	设备节点
1	TTL	/dev/ttyS0

2	RS232	/dev/ttyS3
3	RS232	/dev/ttyS4
4	RS485	/dev/ttyS5
5	RS485	/dev/ttyS7

以上串口均可以使用microcom工具进行测试

▼
Plain Text

```

1 [root@RK356X:/]# microcom -s 115200 /dev/ttyS0
2 [ 2683.415483] of_dma_request_slave_channel: dma-names property of node '/s
  erial@fdd50000' missing or empty
3 [ 2683.415544] ttyS0 - failed to request DMA, use interrupt mode
4

```

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

3 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：

```

1  #关闭can0设备
2  ip link set can0 down
3
4  #设置仲裁段1M波特率，数据段3M波特率
5  ip link set can0 type can bitrate 1000000 dbitrade 3000000 fd on
6
7  #打印can0信息
8  ip -details link show can0
9
10 #启动can0
11 ip link set can0 up
12
13 #执行candump，阻塞等待can0接收
14 candump can0
15
16 #canfd格式发送
17 cansend can0 123##1DEADBEEF
18
19 #can格式发送
20 cansend can0 123#1122334455667788

```

4 WiFi使用

在使用 WiFi时连接好WiFi天线，设备节点为wlan0

```

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

```

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：

```

1 [root@RK356X:/]# cat /userdata/cfg/wpa_supplicant.conf
2 ctrl_interface=/var/run/wpa_supplicant
3 ap_scan=1
4 update_config=1
5
6 network={
7     ssid="TP-LINK_B87A"
8     psk="12345678"
9     key_mgmt=WPA-PSK
10 }
11 [root@RK356X:/]#

```

重启后，将自动连接上热点：

```

1 [root@RK356X:/]# ifconfig wlan0
2 wlan0      Link encap:Ethernet  HWaddr 2C:3B:70:14:17:95
3             inet addr:192.168.1.101  Bcast:192.168.1.255  Mask:255.255.255.0
4             inet6 addr: fe80::220a:b25:4bd:2e3a/64 Scope:Link
5             UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
6             RX packets:26 errors:0 dropped:0 overruns:0 frame:0
7             TX packets:40 errors:0 dropped:0 overruns:0 carrier:0
8             collisions:0 txqueuelen:1000
9             RX bytes:5075 (4.9 KiB)  TX bytes:3913 (3.8 KiB)
10
11 [root@RK356X:/]#

```

5 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

```

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

```

蓝牙功能开启后，将产生hci0节点

```

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

```

使用hcitool测试蓝牙扫描功能

```

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

```

6 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。

6.1 静态IP设置

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

▼ Bash |

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

其中，dns-nameservers一项为默认dns。

7 4G

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

▼ Bash |

```
1 quectel-CM &
```

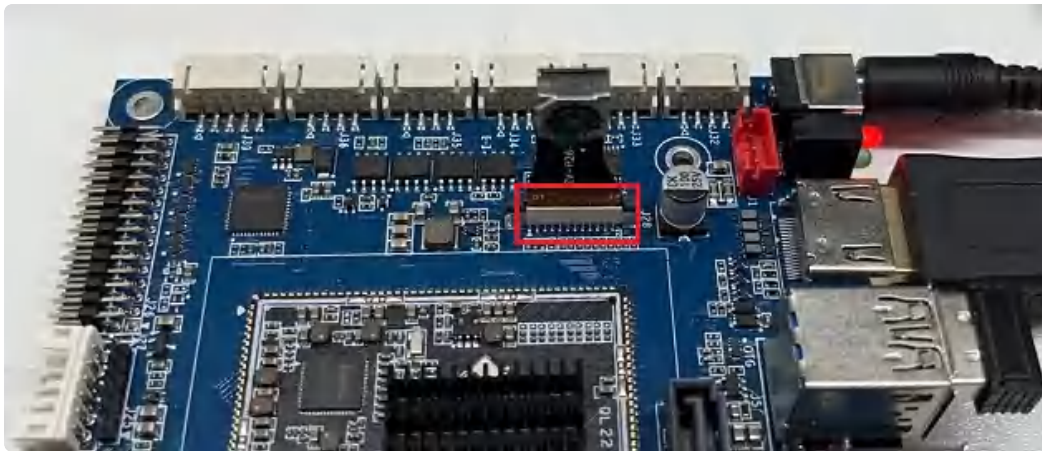
检查拨号上网是否正常

▼ Bash |

```
1 ping 114.114.114.114 -I wwan0
```

8 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。



9 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：

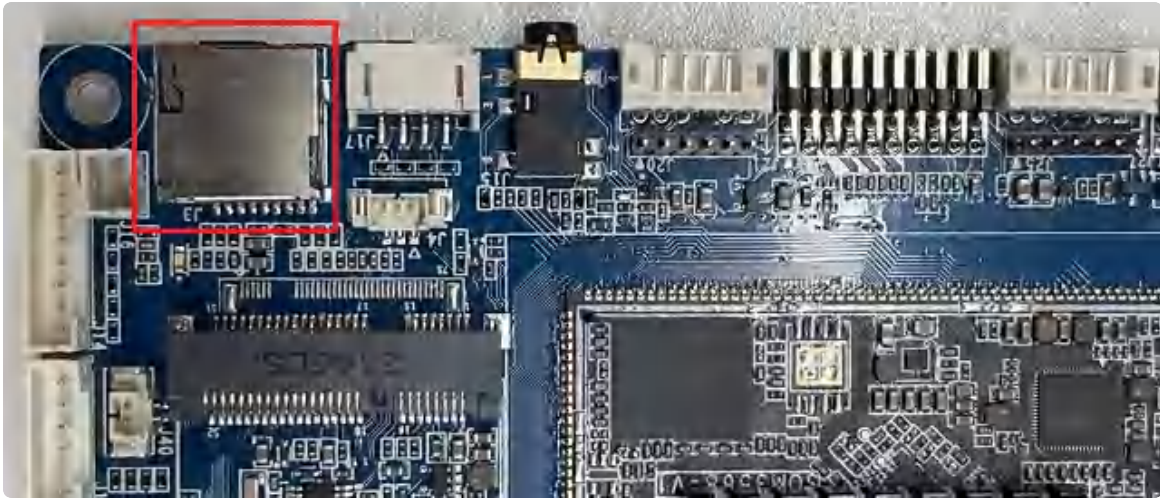
```
▼ Java |
1 host:
2 echo host > /sys/devices/platform/fe8a0000.usb2-phy/otg_mode
3 device:
4 echo peripheral > /sys/devices/platform/fe8a0000.usb2-phy/otg_mode
```

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

10 SD卡

将SD卡插入到SD卡槽中，将自动挂载到/mnt/sdcard/目录下。



Bash

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

11 开机启动程序

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

```
1 [root@RK356X:/]# cat /etc/init.d/S49ntp
2 #!/bin/sh
3
4 NAME=ntpd
5 DAEMON=/usr/sbin/$NAME
6
7 # Gracefully exit if the package has been removed.
8 test -x $DAEMON || exit 0
9
10 # Read config file if it is present.
11 if [ -r /etc/default/$NAME ]
12 then
13     . /etc/default/$NAME
14 fi
15
16 case "$1" in
17     start)
18         printf "Starting $NAME: "
19         start-stop-daemon -S -q -x $DAEMON -- -g
20         [ $? = 0 ] && echo "OK" || echo "FAIL"
21         ;;
22     stop)
23         printf "Stopping $NAME: "
24         start-stop-daemon -K -q -n $NAME
25         [ $? = 0 ] && echo "OK" || echo "FAIL"
26         ;;
27     restart|reload)
28         echo "Restarting $NAME: "
29         $0 stop
30         sleep 1
31         $0 start
32         ;;
33     *)
34         echo "Usage: $0 {start|stop|restart|reload}" >&2
35         exit 1
36         ;;
37 esac
38
39 exit 0
```

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

12 按键

IDO-EVB3568-V1 配置了一个Recovery按键，如下图所示：



在设备断电的情况下，该按键用于烧录固件。在系统正常启动后，则可作为普通按键使用。对应的设备节点为`/dev/input/event2`，键值为`KEY_VOLUMEUP(115)`。

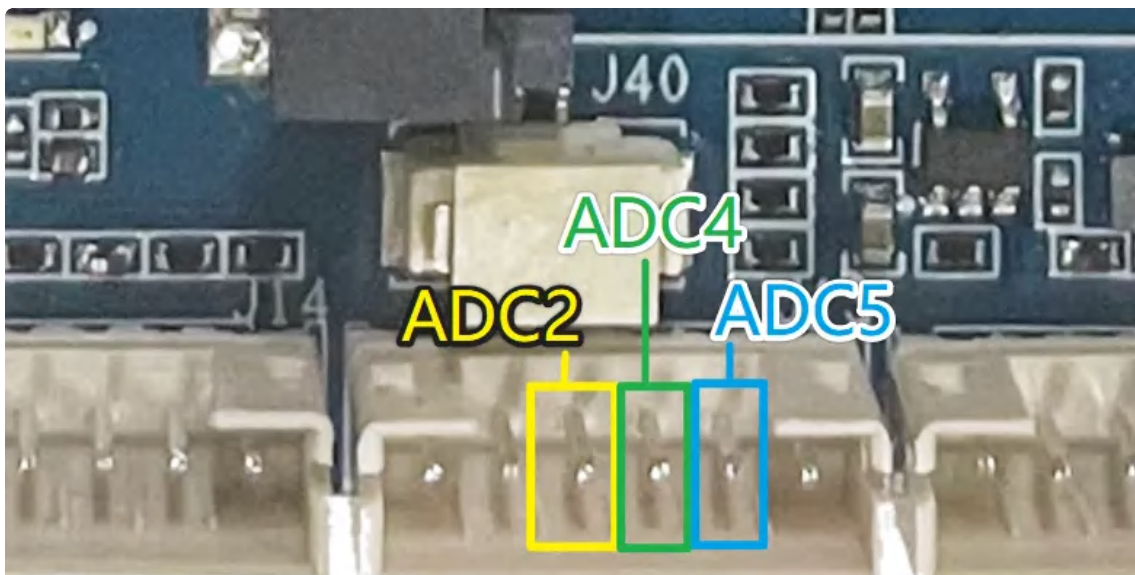
使用`evtest`进行测试：

```
1 [root@RK356X:/]# evtest
2 No device specified, trying to scan all of /dev/input/event*
3 Available devices:
4 /dev/input/event0:      fe6e0030.pwm
5 /dev/input/event1:      rk805 pwrkey
6 /dev/input/event2:      adc-keys
7 /dev/input/event3:      rockchip,rk809-codec Headphones
8 Select the device event number [0-3]: 2
9 Input driver version is 1.0.1
10 Input device ID: bus 0x19 vendor 0x1 product 0x1 version 0x100
11 Input device name: "adc-keys"
12 Supported events:
13   Event type 0 (EV_SYN)
14   Event type 1 (EV_KEY)
15     Event code 114 (KEY_VOLUMEDOWN)
16     Event code 115 (KEY_VOLUMEUP)
17     Event code 139 (KEY_MENU)
18     Event code 158 (KEY_BACK)
19 Properties:
20 Testing ... (interrupt to exit)
21 Event: time 1666752551.345149, type 1 (EV_KEY), code 115 (KEY_VOLUMEUP), v
    alue 1
22 Event: time 1666752551.345149, ----- SYN_REPORT -----
23 Event: time 1666752551.551624, type 1 (EV_KEY), code 115 (KEY_VOLUMEUP), v
    alue 0
24 Event: time 1666752551.551624, ----- SYN_REPORT -----
25 Event: time 1666752552.274980, type 1 (EV_KEY), code 115 (KEY_VOLUMEUP), v
    alue 1
26 Event: time 1666752552.274980, ----- SYN_REPORT -----
27 Event: time 1666752552.688312, type 1 (EV_KEY), code 115 (KEY_VOLUMEUP), v
    alue 0
28 Event: time 1666752552.688312, ----- SYN_REPORT -----
29
```

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。

13 ADC

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



设备节点对应关系如下表：

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

13.1 ADC值读取

▼

Bash

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

13.2 ADC电压转换关系

▼

Bash

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

假设in_voltage2_raw的值为500，则对应的ADC电压为 $V=(500/1024)*1.8v=0.879v$

14 时间设置 RTC

主板包含2个RTC，其中/dev/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 |
1 root@rk3568-buildroot:/# dmesg | grep rtc
2 [ 1.583028] rk808-rtc rk808-rtc: registered as rtc0
3 [ 1.584797] rk808-rtc rk808-rtc: setting system clock to 2017-08-04T09:00:03 UTC (1501837203)
4 [ 1.601112] rtc-hym8563 5-0051: registered as rtc1
5
```

14.1 获取RTC时间

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

14.2 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

```

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

```

1 #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

```

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

```

14.2 NTP时间同步

系统默认开启了NTP服务，连接网络后，将自动同步网络时间。

14.3 时区

查看时区

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

+0000表示在0时区。

设置时区

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

15 音频

使用[aplay](#)工具查看声卡设备

```
▼ Bash |
1 [root@RK356X:/]# aplay -l
2 **** List of PLAYBACK Hardware Devices ****
3 card 0: rockchiprk809co [rockchip,rk809-codec], device 0: fe410000.i2s-rk817-hifi rk817-hifi-0 [fe410000.i2s-rk817-hifi rk817-hifi-0]
4 Subdevices: 1/1
5 Subdevice #0: subdevice #0
```

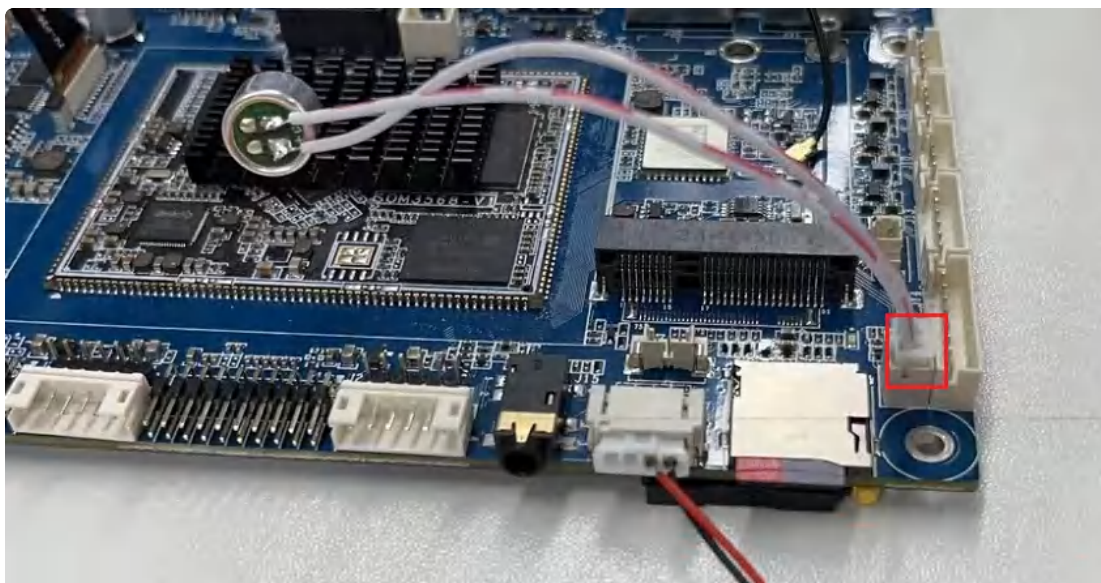
15.1 Lineout

不插入耳机，使用[aplay](#)播放wav音频测试

1

1

1



使用arecord工具进行录音测试。

Bash

```
1 [root@RK356X:/]# arecord -D hw:0,0 -r 48000 -c 2 -f S16_LE test.wav
2 Recording WAVE 'test.wav' : Signed 16 bit Little Endian, Rate 48000 Hz, Stereo
3 ^CAborted by signal Interrupt...
```

播放录音

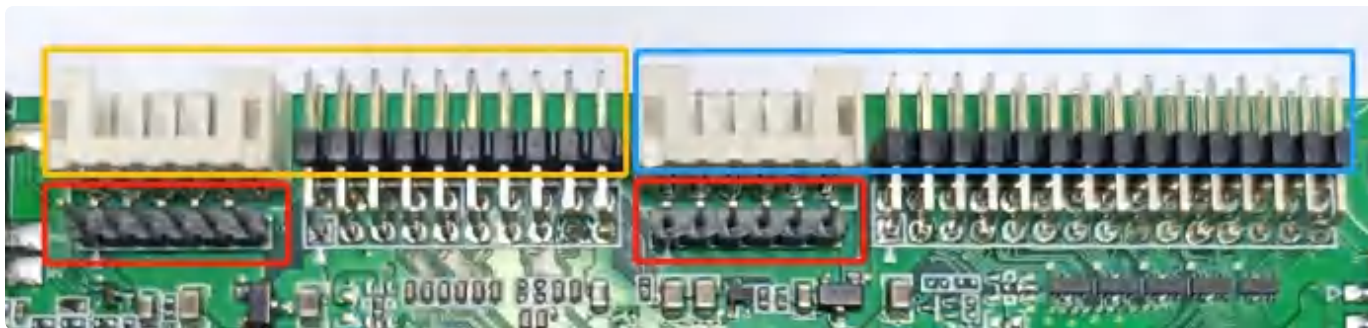
Bash

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

16 显示屏

16.1 显示屏接口说明

主板各个显示接口如下图所示：



黄色框是eDP屏接口，蓝色框是Dual LVDS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。

16.2 显示设置

16.2.1 屏幕背光亮度设置

- eDP/MIPI屏背光控制

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

设置方法：(支持调节范围 0-255)

```

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

```

- Dual LVDS屏幕背光控制

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

设置方法：(支持调节范围 0-255)

```

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

```

17 扩展GPIO

主板扩展了5路GPIO，位于J14：



各个引脚的功能描述如下表所示：

序号	定义	电平/V	说明
1	5V	5V	电源5V输出
2	GND	GND	电源地
3	SPI3_CLK_M1/GPIO4_C2/PWM14_M1	3.3V	默认配置为GPIO，GPIO编号146
4	SPI3_MOSI_M1/GPIO4_C3/PWM15_IR_M1	3.3V	默认配置为GPIO，GPIO编号147
5	SPI3_MISO_M1/GPIO4_C5/PWM12_M1	3.3V	默认配置为GPIO，GPIO编号149
6	SPI3_CS0_M1/GPIO4_C6/PWM13_M1	3.3V	默认配置为GPIO，GPIO编号150

7	PWM7_IR/GPIO0_C6	3.3V	默认配置为GPIO，GPIO编号22
8	GND	GND	电源地

17.1 测试方法

以GPIO4_C2为例。

作为输入：

```

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

```

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

作为输出：

```

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

```

IDO-EVB3568-V1 Debian系统

1 系统用户名及密码

系统用户名及密码如下表所示：

文件系统版本	账户	密码
Debian10	root	未设置密码

2 调试口使用

2.1 串口调试



调试串口为 TTL 电平，开发板接口为 MX1.25 接线端子，使用 USB 转串口模块连接 PC 调试终端。

串口参数：波特率 1500000、数据位 8bit、无校验位、停止位 1bit。

Baud rate:	1500000	Flow control
Data bits:	8	☐ DTR/DSR
Parity:	None	☐ RTS/CTS
Stop bits:	1	☐ XON/XOFF

2.2 ADB调试

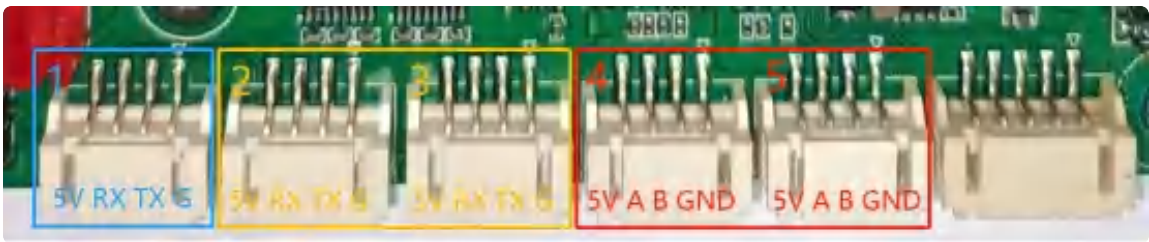


上图红色框内的USB接口为支持OTG模式切换，在系统上电前，使用双公头 USB 数据线连接开发板和 PC 端的USB接口，在PC终端识别到 ADB 设备，即可使用 adb shell 调试。

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

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

3 串口测试



串口接口位置及引脚定义如上图所示，设备节点如下表所示：

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

- Debian10 系统串口测试

系统已安装 microcom 工具，可使用此工具给串口节点设置串口参数，例如 /dev/ttyS0 设置波特率为 115200，命令执行后在终端中输入任意字符（输入的字符不会显示），与 /dev/ttyS0 接口相连的上位机软件将会接收到终端输入的字符。

Java

```
1 # microcom -p /dev/ttyS0 -s 115200
2 connected to /dev/ttyS0
3 Escape character: Ctrl-\
4 Type the escape character to get to the prompt.
```

输入 Ctrl-\ 进入 microcom 的命令终端，再输入quit 即可退出microcom 软件。

```
1 Enter command. Try 'help' for a list of builtin commands
2 -> quit
3 exiting
```

4 CAN测试



开发板共有两路CAN接口，支持 CANFD 协议，CAN接口测试方法如下：

```
1 #关闭can0设备
2 ip link set can0 down
3
4 #设置仲裁段1M波特率，数据段3M波特率
5 ip link set can0 type can bitrate 1000000 dbitrate 3000000 fd on
6
7 #打印can0信息
8 ip -details link show can0
9
10 #启动can0
11 ip link set can0 up
12
13 #执行candump，阻塞等待can0接收
14 candump can0
15
16 #canfd格式发送
17 cansend can0 123##1DEADBEEF
18
19 #can格式发送
20 cansend can0 123#1122334455667788
```

5 WiFi使用

在使用 WiFi 时需要在开发板的 WiFi 天线座上连接好外接天线。

WiFi 设备节点为：wlan0

```
▼ Bash |
1  # ifconfig wlan0
2  wlan0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
3          ether 90:e8:68:b3:9b:53  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
```

- Debian10桌面系统WiFi设置

点击桌面右下角的电脑图标，大弹出的界面中，选择路由并填写密码上网。

6 蓝牙使用

设备节点: hci0

Debian系统下的蓝牙管理工具开机自动启动，并将蓝牙设备开启，可以点击桌面的右下蓝牙图标，连接蓝牙设备。

命令行查看蓝牙节点状态如下：

```

1 # hciconfig -a
2 hci0:   Type: Primary   Bus: UART
3         BD Address: 67:DF:2D:A3:09:10   ACL MTU: 1021:8   SCO MTU: 64:1
4         UP RUNNING
5         RX bytes:1130 acl:0 sco:0 events:63 errors:0
6         TX bytes:4968 acl:0 sco:0 commands:63 errors:0
7         Features: 0xbf 0xfe 0xcf 0xfe 0xdb 0xff 0x7b 0x87
8         Packet type: DM1 DM3 DM5 DH1 DH3 DH5 HV1 HV2 HV3
9         Link policy: RSWITCH SNIFF
10        Link mode: SLAVE ACCEPT
11        Name: 'linaro-alip'
12        Class: 0x1c0000
13        Service Classes: Rendering, Capturing, Object Transfer
14        Device Class: Miscellaneous,
15        HCI Version: 4.2 (0x8) Revision: 0x0
16        LMP Version: 4.2 (0x8) Subversion: 0x2209
17        Manufacturer: Broadcom Corporation (15)

```

扫描周边的BLE蓝牙

```

1 # hcitool scan
2 Scanning ...
3         7C:03:AB:A2:DF:BE           n/a
4         94:87:E0:9D:14:12          seeyou
5         B4:20:5B:5A:2F:50          OPPO Reno6 5G
6         04:E5:98:75:D4:D3          Redmi 9
7         7C:2A:DB:01:48:F9          K00KV
8         D0:97:FE:61:46:0C          GT Neo
9         A4:50:46:9C:DC:23          小米手机8

```

7 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。

以太网接口默认IP获取方式为 DHCP，只需要将以太网接口连接路由器即可为开发板动态分配 IP 地址。

7.1 静态IP设置

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

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

7.2 DNS设置

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

```
1 dns-nameservers 114.114.114.114
```

8 4G使用

开发板带有PCIe座子，可连接 EC20 4G模块。默认系统支持移远拨号管理软件 quectel-CM系统默认适配了 EC20 miniPCIe 4G模块，支持自动拨号上网。使用 4G 功能前，需要先在板载 mini PCIe 接口插入 EC20 模块，在卡槽插入 SIM 卡，并连接好 4G 天线以保证信号的稳定。

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

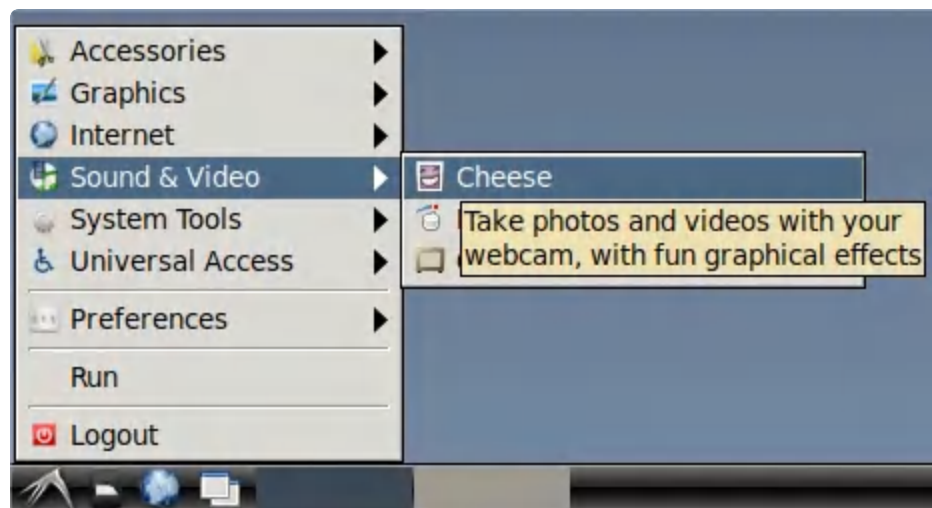
```
1 #停止拨号服务
2 systemctl stop ec20-dialup.service
3 #开启拨号服务
4 systemctl start ec20-dialup.service
```

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。

9 摄像头使用



节点：/dev/video0可用系统自带软件Cheese进行测试，如下图所示：



10 U盘/TF挂载



红色框内的接口默认为OTG模式，device和host模式的切换方法如下：

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

- Debian10 系统

支持FAT32格式U盘/TF卡的自动挂载，默认挂载目录为 /media/linaro/

11 设置开机启动程序

- 非界面程序自启动

在 /etc/rc.local 脚本中添加需要执行的应用程序命令。

- 带界面应用程序自启动

在 /home/linaro/.config/lxsession/LXDE/autostart 文件中添加需要执行的程序。

12 按键

开发板有一个Recovery按键，如下图所示：



在系统正常启动后，可作为普通按键使用，设备节点： /dev/input/event3，键值： KEY_VOLUMEUP。
在可使用 hexdump 命令确认按键是否有效

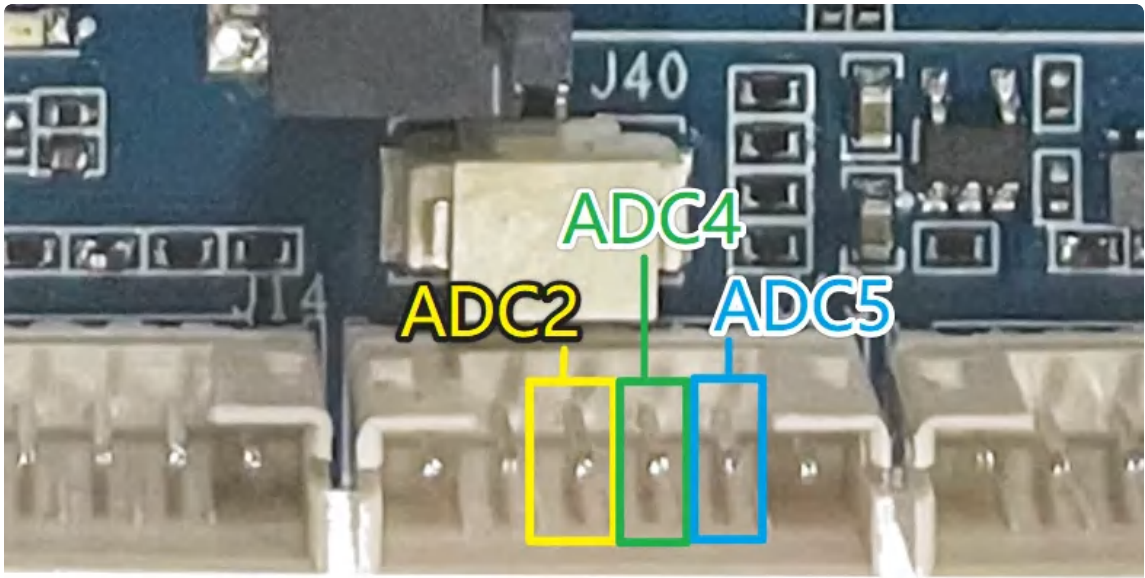
▼

Bash |

1 hexdump /dev/input/event3

13 ADC

开发板引出了3路 ADC 接口，位置如下图所示：



支持10位的SAR-ADC，标准电压位1.8V，ADC接口列表如下所示

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

```

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

```

14 时间设置

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

```

1 root@rk3568-buildroot:/# dmesg | grep rtc
2 [ 1.583028] rk808-rtc rk808-rtc: registered as rtc0
3 [ 1.584797] rk808-rtc rk808-rtc: setting system clock to 2017-08-04T09:00:03 UTC (1501837203)
4 [ 1.601112] rtc-hym8563 5-0051: registered as rtc1
5

```

14.1 获取RTC时间

```

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

```

14.2 设置RTC时间

▼ 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

▼ Bash |

```
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
```

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 |

```
1 #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

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

14.3 时区修改

系统默认时区为 Asia/Shanghai

可以在命令执行 `tzselect` 命令修改系统时区

15 音视频

15.1 播放音频的工具

使用 `aplay -l` 命令可查看当前开发板的声卡信息，如下图所示：

```
1 $ aplay -l
2 List of PLAYBACK Hardware Devices
3 card 0: rockchiprk809co [rockchip,rk809-codec], device 0: fe410000.i2s-rk81
  7-hifi rk817-hifi-0 [fe410000.i2s-rk817-hifi rk817-hifi-0]
4 Subdevices: 1/1
5 Subdevice #0: subdevice #0
```

播放wav音频文件

```
1 $ aplay -Dhw:0,0 demo.wav
```

- Debian10 系统播放音视频

界面上可以使用 MPV 播放器播放音视频文件。

15.2 音量调节

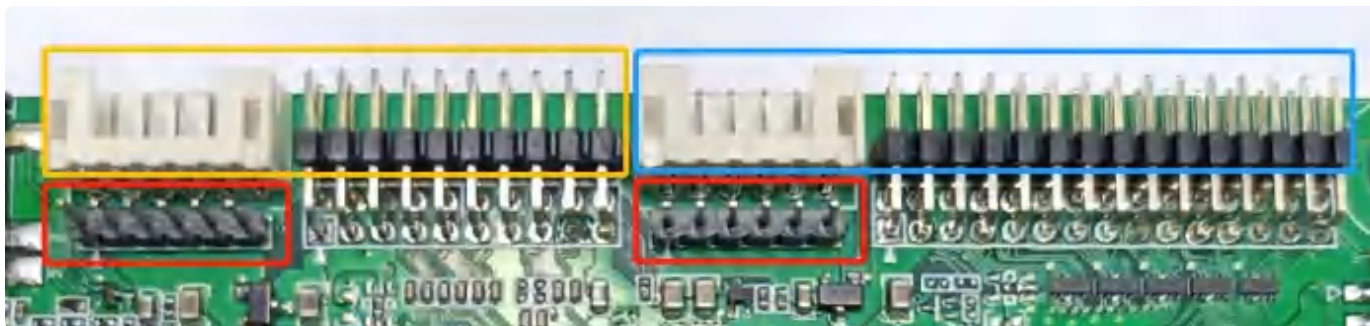
点击系统桌面左下角图标，选择 [Sound Video]→[PulseAudio Volume Control]

在软件界面中调整音量的大小。

16 显示屏

16.1 显示屏接口说明

主板屏幕接口如下图所示：



黄色框是eDP屏接口，蓝色框是Dual LVDS屏接口。

红色框分别是两者的供电电口，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。

16.2 显示设置

16.2.1 屏幕背光亮度设置

- eDP/MIPI屏幕背光控制

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

设置方法: (支持调节范围 0–255)

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

- Dual LVDS屏幕背光控制

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

设置方法: (支持调节范围 0–255)

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

17 桌面shell终端

- Debian10系统

点击系统桌面左下角图标, 选择[System Tools]→[LXTerminal]

18 浏览器

- Debian10系统自带Chromium浏览器

点击系统桌面左下角图标, 选择 [Internet]→[Chromium Browser]

19 GPU测试

命令行执行 test_glmark2.sh 脚本测试

