

IDO-SBC3568-V1B Ubuntu系统使用手册

调试

[串口调试](#)

[ADB调试](#)

[SSH调试](#)

[UART测试](#)

[CAN测试](#)

[WIFI使用](#)

[蓝牙使用](#)

[以太网使用](#)

[静态IP设置](#)

[4G使用](#)

[摄像头使用](#)

[测试](#)

[测试摄像头是否存在](#)

[抓取视频流](#)

[USB](#)

[SD Card](#)

[按键](#)

[ADC](#)

[ADC值读取](#)

[ADC电压转换关系](#)

[时间设置](#)

[RTC时间读取和同步](#)

[NTP时间同步](#)

[时区](#)

[查看时区](#)

[设置时区](#)

[音频](#)

Lineout

耳机

录音

打开mic通道

录音

播放录音

显示屏

显示屏接口说明

显示设置

屏幕背光亮度设置



IDO-SBC3568-V1B

Ubuntu 系统使用手册

深圳触觉智能科技有限公司

www.industio.cn

文档修订历史

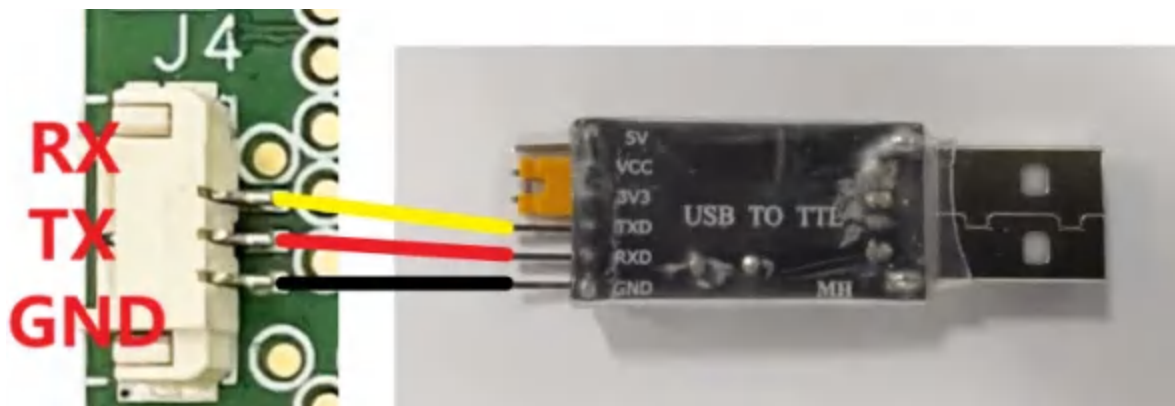
版本	修订内容	修订	审核	日期
V1.0	1、创建文档	刘崇凯		2023/5/16

调试

IDO-SBC3568-V1B开发板支持串口调试、ADB调试和远程SSH调试。

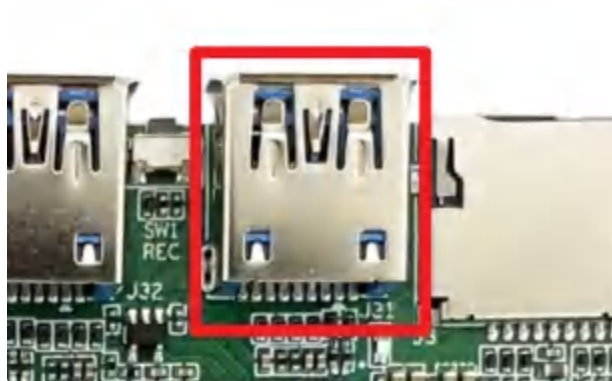
串口调试

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。

```
ido@ido: ~  
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:~$
```

UART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	丝印	功能	设备节点
1	J33	RS485	/dev/ttyS0
2	J34	RS485	/dev/ttyS3
3	J35	RS232	/dev/ttyS4
4	J36	RS232	/dev/ttyS5
5	J37	RS232	/dev/ttyS7
6	J38	RS232	/dev/ttyS9

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

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

▼

Plain Text

```
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。

CAN测试

CAN位于J44的双排针。



序号	编号	描述
19	CAN1_TX	CAN1
20	CAN1_RX	
17	CAN2_TX	CAN2
18	CAN2_RX	

IDO-SBC3568-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 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
```

WiFi使用

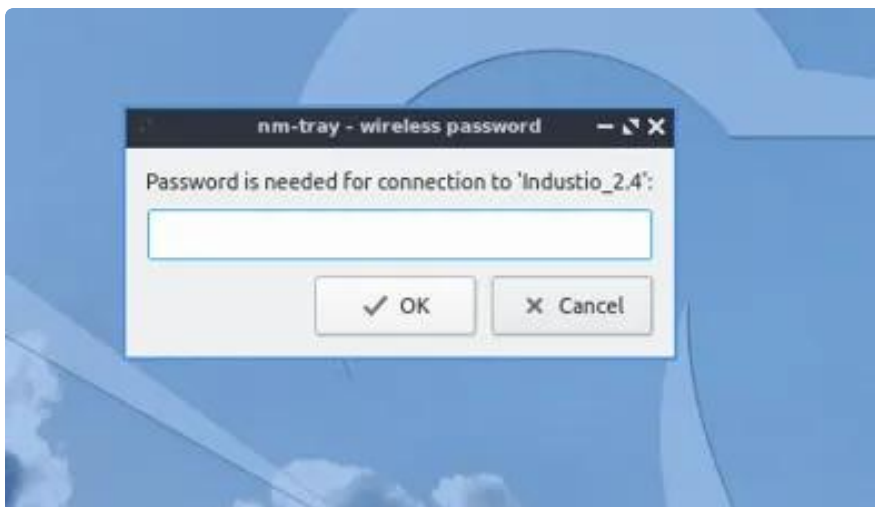
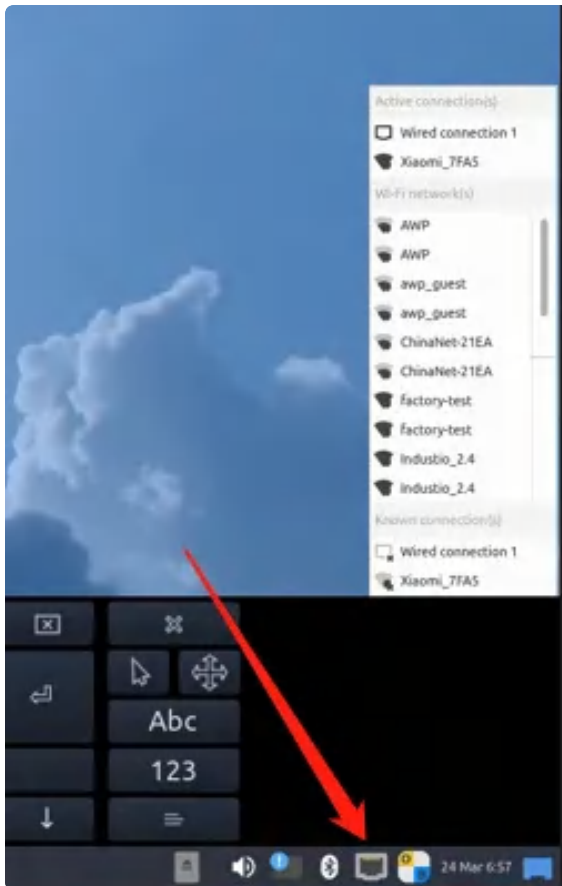
开发板板载USB WiFi 模块为RTL8821CS

网络设备节点：wlan0

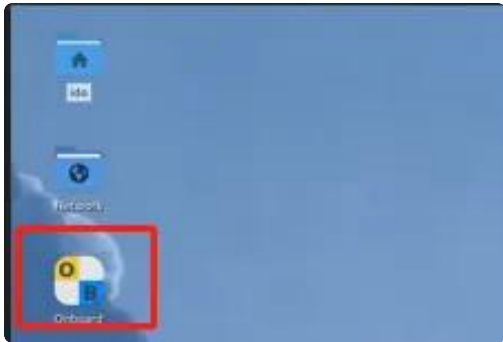
联网方法：Ubuntu-desktop系统可在界面上配置SSID和密码连接附近的WiFi路由。

在Ubuntu系统桌面连接wifi热点：

鼠标右键右下角网口图标，弹出wifi框，点击想要连接的热点。



这里我们可以通过虚拟键盘，进行输入密码。



点击上图图标会弹出虚拟键盘。

用户命令行连接wifi

修改wifi配置文件：

```
Bash |
1 root@ido:~#vi /etc/wpa_supplicant/wpa_supplicant.conf
```

连接wifi：

```
Bash |
1 root@ido:~#vi wpa_supplicant -D nl80211 -i wlan0 -c /etc/wpa_supplicant/wpa
 _supplicant.conf -B &
```

蓝牙使用

开发板板载USB WiFi+蓝牙二合一模块RTL8821CS

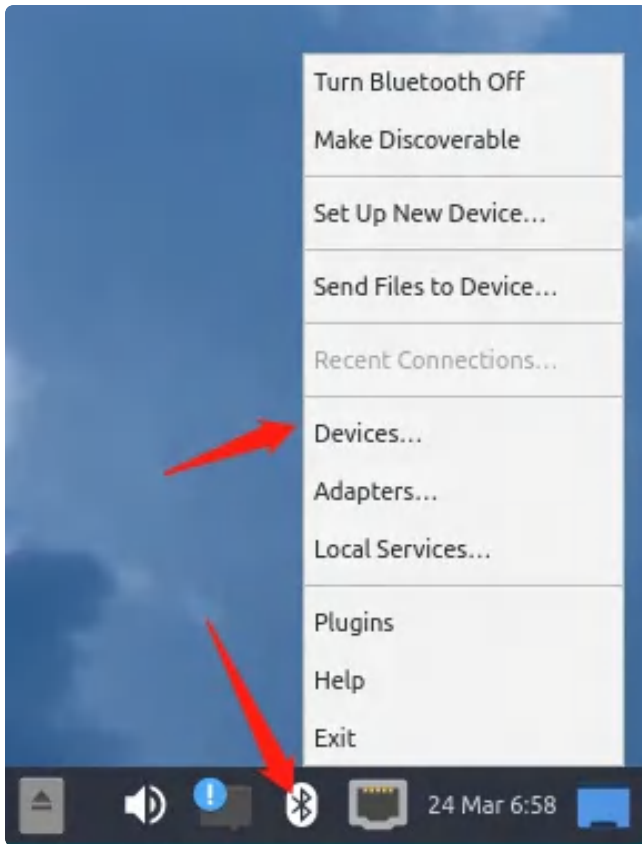
设备节点：hci0

蓝牙标准：蓝牙4.2双模

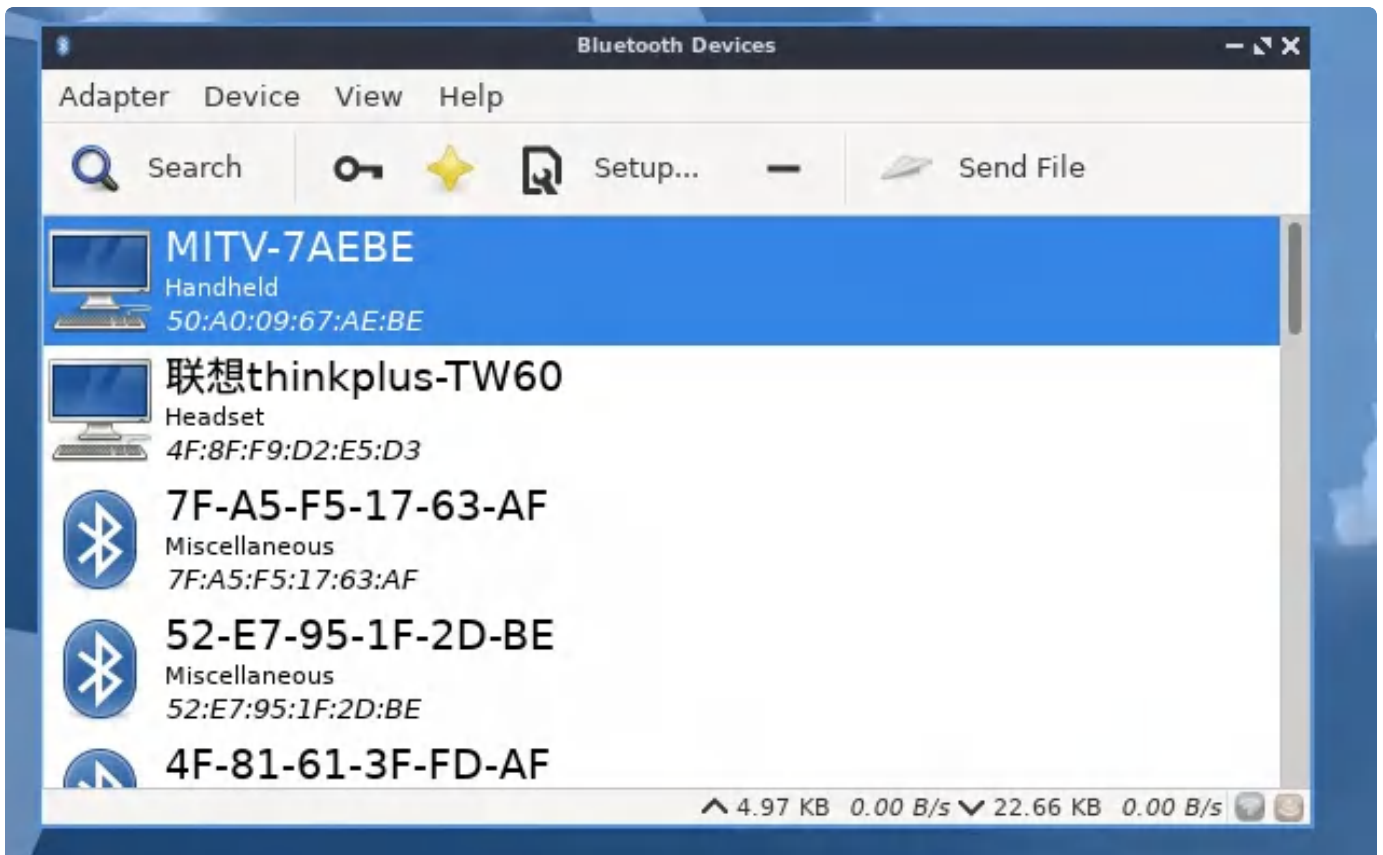
连接方法：Ubuntu系统可在界面上可以搜索附近的蓝牙并连接，

在Ubuntu系统桌面连接蓝牙：

鼠标右键点击右下角蓝牙图标，弹出选项框后，点击“Devices”



弹出蓝牙搜索界面。



右键点击相应的蓝牙设备，连接即可。

命令行开启蓝牙

蓝牙功能开启后，将产生hci0节点

```
▼ Bash |
1 root@ido:~# 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@ido:~# hciconfig hci0 up
2 root@ido:~# 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
```

以太网使用

主板配置了1个1000M以太网接口，对应的网络设备节点为eth0。



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

```
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。

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。

```
1  /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。



MIPI CSI 接口于J22，如上图所示，支持OV5648 摄像头模组，连接方法如下



测试

测试摄像头是否存在

```
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。

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 --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-lspspeex --enable-libssh --enable-libtheora --enable-libtwolame --enable-libvidstab --enable-libvorbis --enable-libvpx --enable-libwavpack --enable-libwebp --enable-libx265 --enable-libx264 --enable-libx265 --enable-libx264 --enable-libxvid --enable-libzmq --enable-libzvbi --enable-lv2 --enable-omx --enable-opengl --enable-opencore --enable-opencore --enable-opengl --enable-sdl2 --enable-libdc1394 --enable-libdrm --enable-libiec61883 --enable-chromaprint --enable-frei0r --enable-libx264 --enable-libdrm --enable-librga --enable-rkmp --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
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 inaccurate
20 Input #0, rawvideo, from './out.yuv':

```

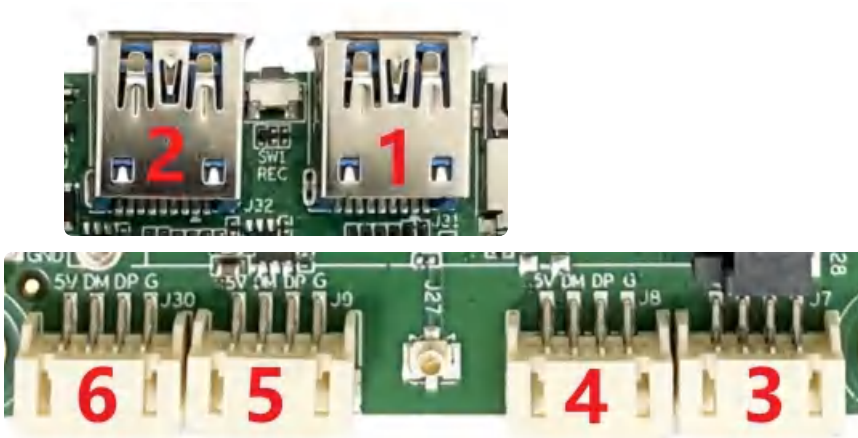
```

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

```



USB



USB接口如上图所示，功能说明如下

序号	丝印	功能	控电节点
2	J31	USB 2.0 HOST	/sys/class/leds/usb3_host_pwr/brightness
3	J7	USB 2.0 HOST	/sys/class/leds/usb2_host_pwr/brightness

4	J8	USB 2.0 HOST	/sys/class/leds/usb2_fe2_pwr/brightness
5	J9	USB 2.0 HOST	/sys/class/leds/usb2_fe3_pwr/brightness
6	J30	USB 2.0 HOST	/sys/class/leds/usb2_fe4_pwr/brightness

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。

打开USB6的电源：

```

1  linaro@linaro-alip:~$ sudo echo 255 > /sys/class/leds/usb2_fe4_pwr/brightness

```

关闭USB6的电源：

```

1  linaro@linaro-alip:~$ sudo echo 0 > /sys/class/leds/usb2_fe4_pwr/brightness

```

其他USB的电源控制方法类似。

USB OTG 切换命令

USB OTG 支持host 和device 模式的切换，软件切换方法如下

```

1  host:
2  echo host > /sys/devices/platform/fe8a0000.usb2-phy/otg_mode
3  echo HOST > /dev/otg_mode
4  device:
5  echo peripheral > /sys/devices/platform/fe8a0000.usb2-phy/otg_mode
6  echo DEVICE > /dev/otg_mode

```

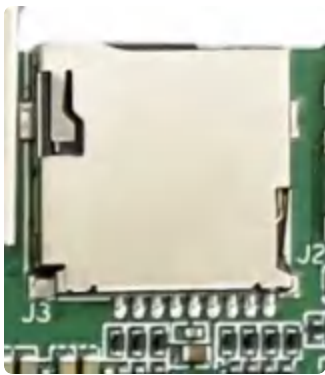
当接入U盘设备时，默认挂载到/media/ido/目录下

Bash

```
1 root@ido:~# mount
2 ...
3 /dev/sda1 on /media/ido/KINGSTON type vfat (rw,nosuid,nodev,relatime,uid=1001,gid=1001,mask=0022,dmask=0022,codepage=936,ioccharset=utf8,shortname=mixed,showexec,utf8,flush,errors=remount-ro,uhelper=udisks2)
4 ...
```

SD Card

将SD卡插入到SD卡槽中，将自动挂载到/media/ido/目录下。



Bash

```
1 root@ido:~# mount
2 ...
3 /dev/mmcblk0p1 on /media/ido/E2C4-11A5 type vfat (rw,nosuid,nodev,relatime,uid=1001,gid=1001,mask=0022,dmask=0022,codepage=936,ioccharset=utf8,shortname=mixed,showexec,utf8,flush,errors=remount-ro,uhelper=udisks2)
4 ...
```

按键

丝印	功能
SW1	Recovery
SW3	Power

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。

使用evtest进行测试：

```
▼ Bash |
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。

ADC

IDO-SBC3568-V1共配置了2路ADC 接口（精度为10位）位于J55的第4、5引脚，分别记作ADC4、ADC5。精度为10位，位置如下图所示：



设备节点对应关系如下表：

序号	编号	设备节点
4	SARADC VINU4_	/sys/bus/iio/devices/iio:device0/in_voltage4_raw
5	SARADC VINU5_	/sys/bus/iio/devices/iio:device0/in_voltage5_raw

ADC值读取

▼

Bash

```
1 cat /sys/bus/iio/devices/iio\:device0/in_voltage4_raw
```

ADC电压转换关系

▼

Bash

```
1 V=(in_voltage4_raw/1024)*1.8v
```

假设in_voltage4_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

```
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
```

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

```
1 root@ido:~# 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@ido:~# export TZ='Asia/Shanghai'
2 root@ido:~#
3 root@ido:~# date -R
4 Wed, 26 Oct 2022 11:30:02 +0800
5 root@ido:~#
```

音频

使用aplay工具查看声卡设备

▼

Bash |

```
1 root@ido:~# 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
```

Lineout

不插入耳机，使用aplay播放wav音频测试

▼

Bash |

```
1 root@ido:~# aplay -D plughw:1,0 /usr/share/sounds/alsa/Rear_Center.wav
2 Playing WAVE '/etc/bsa_file/8k16bpsStereo.wav' : Signed 16 bit Little Endia
  n, Rate 8000 Hz, Stereo
```

耳机

插入耳机，使用aplay播放wav音频测试

▼

Bash |

```
1 root@ido:~# aplay -D plughw:1,0 /usr/share/sounds/alsa/Rear_Center.wav
2 Playing WAVE '/etc/bsa_file/8k16bpsStereo.wav' : Signed 16 bit Little Endia
  n, Rate 8000 Hz, Stereo
```

录音

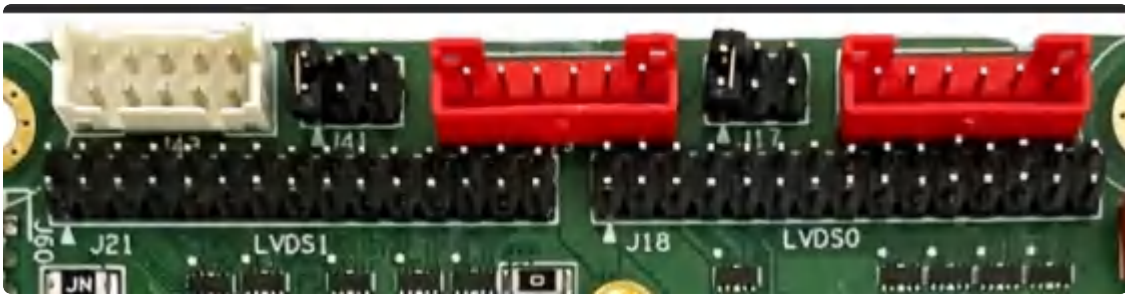
打开mic通道

▼

Bash |

```
1 alsamixer
```

Capture MIC Path选择Main Mic



上图为LVDS接口。

红色座子是两者的供电口，提供12/5/3.3V。



开发板背面，其中J14是mipi屏接口，J56为I2C触摸屏接口。

显示设置

屏幕背光亮度设置

- 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
```