

IDO-EVB3562-V1 Linux 使用手册

IDO-EVB3562-Debian系统

1 调试

1.1 串口调试

1.2 adb调试

1.3 ssh调试

2 串口

3 USB

4 Micro SD

5 Ethernet

6 WiFi

6.1 连接热点

6.1.1 命令行连接

6.1.2 桌面连接

7 Bluetooth

7.1 开启蓝牙

7.2 扫描设备

7.3 连接蓝牙设备

8 4G

9 音频

9.1 喇叭

9.2 MIC

9.3 耳机

10 Camera

10.1 检测摄像头

10.2 预览摄像头

10.3 抓取视频

11 LED

12 RTC

12.1 读取RTC时间

12.2 设置RTC时间

13 硬盘

14 扩展接口

14.1 UART

14.2 SPI

14.3 ADC输入

14.4 SPI

15 RealTime Linux 实时性系统

15.1 安装测试工具包

15.2 测试实时性性能

15.2.1 开启压力测试：

15.2.2 使用cyclicttest

IDO-EVB3562-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

3 USB

4 Micro SD

5 Ethernet

6 WiFi

6.1 扫描热点

6.2 连接热点

7 Bluetooth

7.1 开启蓝牙

7.2 扫描设备

7.3 连接蓝牙设备

8 4G

9 音频

9.1 喇叭

8.2 MIC

8.3 耳机

10 Camera

10.1 检测摄像头

10.2 预览摄像头

11 LED

12 RTC

12.1 读取RTC时间

12.2 设置RTC时间

13 硬盘

14 扩展接口

14.1 UART

14.2 ADC输入

14.4 SPI

15 RealTime Linux实时性系统

15.1 开启压力测试

15.2 使用cyclicttest测试

IDO-EVB3562-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

3 USB

4 Micro SD

5 Ethernet

6 WiFi

6.1 连接热点

6.1.1 命令行连接

6.1.2 桌面连接

7 Bluetooth

7.1 开启蓝牙

7.2 扫描设备

7.3 连接蓝牙设备

8 4G

9 音频

9.1 喇叭

9.2 MIC

9.3 耳机

10 Camera

10.1 检测摄像头

10.2 预览摄像头

10.3 抓取视频

11 LED

12 RTC

12.1 读取RTC时间

12.2 设置RTC时间

13 硬盘

14 扩展接口

14.1 UART

14.2 SPI

14.3 ADC输入

15 QT

16 RealTime Linux 实时性系统

16.1 安装测试工具包

16.2 测试实时性性能

16.2.1 开启压力测试:

16.2.2 使用cyclictest



IDO-EVB3562-V1

Linux 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	V1B	创建文档	TWX	IDO	2023/08/30

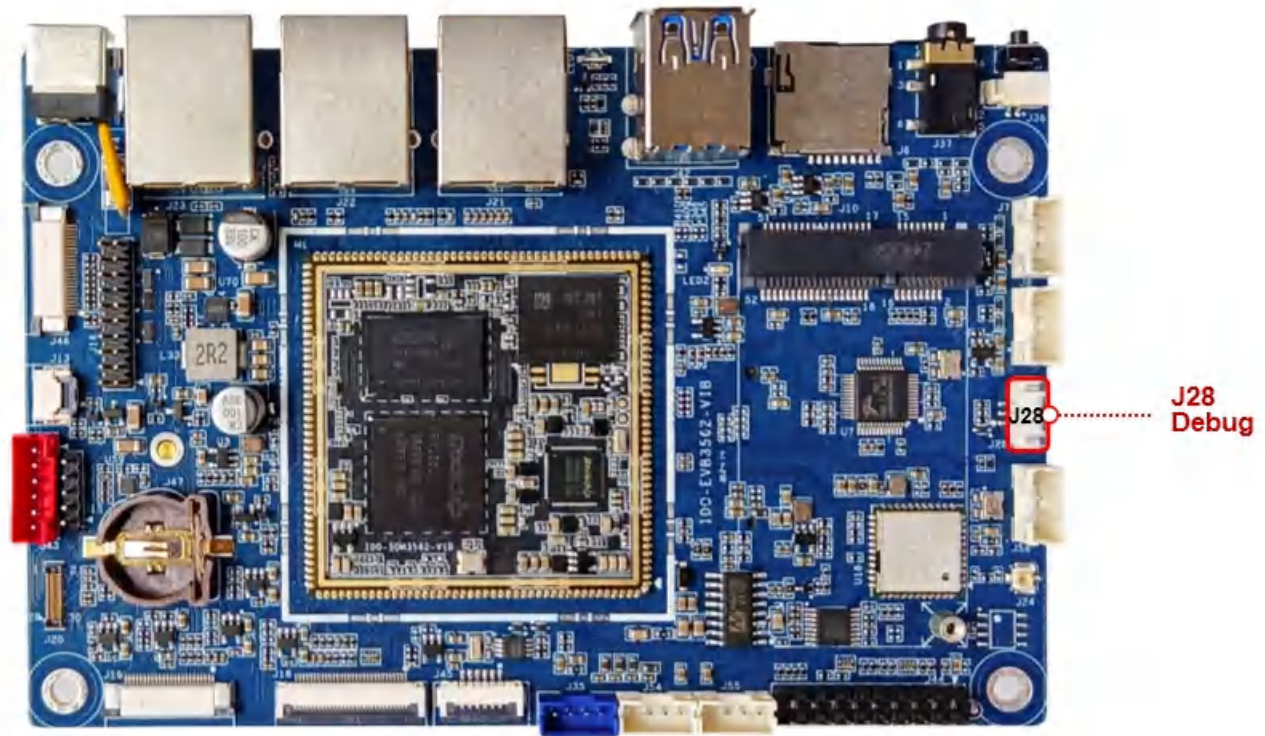
V1.1	V1B	增加"硬盘"一章	TWX	IDO	2023/09/12
V1.2	V1B	文档优化	LZR	IDO	2024/04/16
V1.3	V1B	增加Ubuntu系统使用手册	TWX	IDO	2024/08/14
V1.4	V1B	修正串口设备节点ttyS2为ttyS4； 修改buildroot系统camera测试方法； 文档格式优化； 移除otg device/host切换指令；	TWX	IDO	2024/08/27
V1.5	V1B	增加“RealTime Linux 实时性系统”一章	LJH	IDO	2024/10/9

IDO–EVB3562–Debian系统

1 调试

1.1 串口调试

主板调试串口位于USB座子下方**J28**，建议使用配套的usb转串口工具，如下图所示：



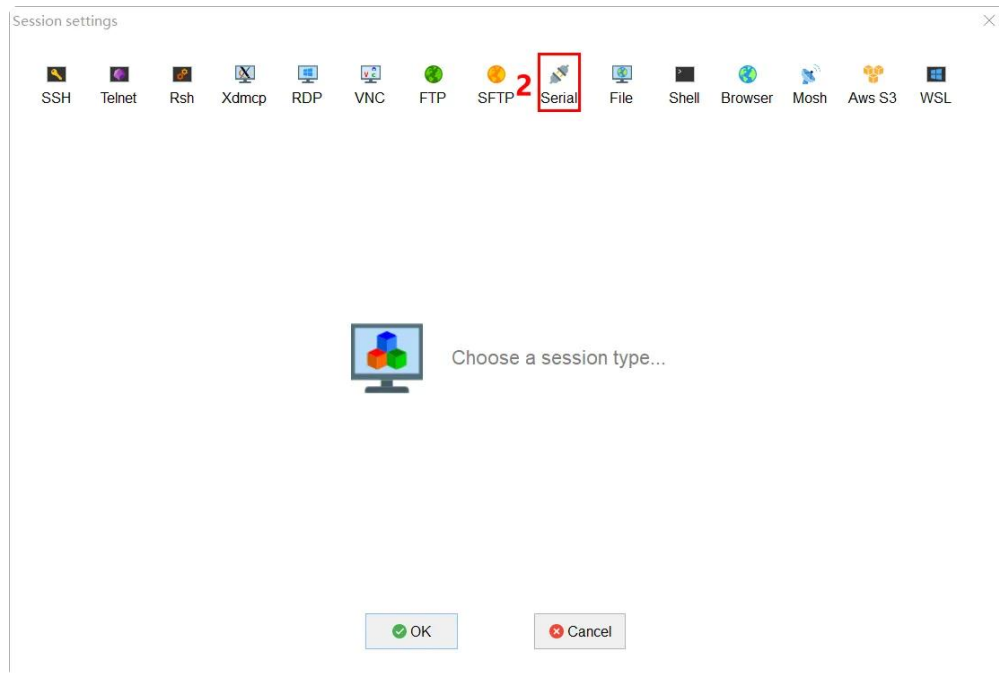
1. 打开MobaXterm, 下载链接如下:

链接: https://pan.baidu.com/s/11ui4LTd2mq_9kiJpeL4bWg?pwd=1234

提取码: 1234

☐ 文件名	↓ 修改时间	类型	大小
☐ RKDevTool_Release_v2.95.zip	2024-04-24 11:53	zip文件	2.30MB
☐ other_tools.txt	2024-04-25 15:31	txt文件	44B
☒ MobaXterm_Portable_v23.6.zip	2024-04-24 14:30	zip文件	39.99MB
☐ DriverAssitant_v5.11.zip	2024-04-24 11:52	zip文件	9.36MB

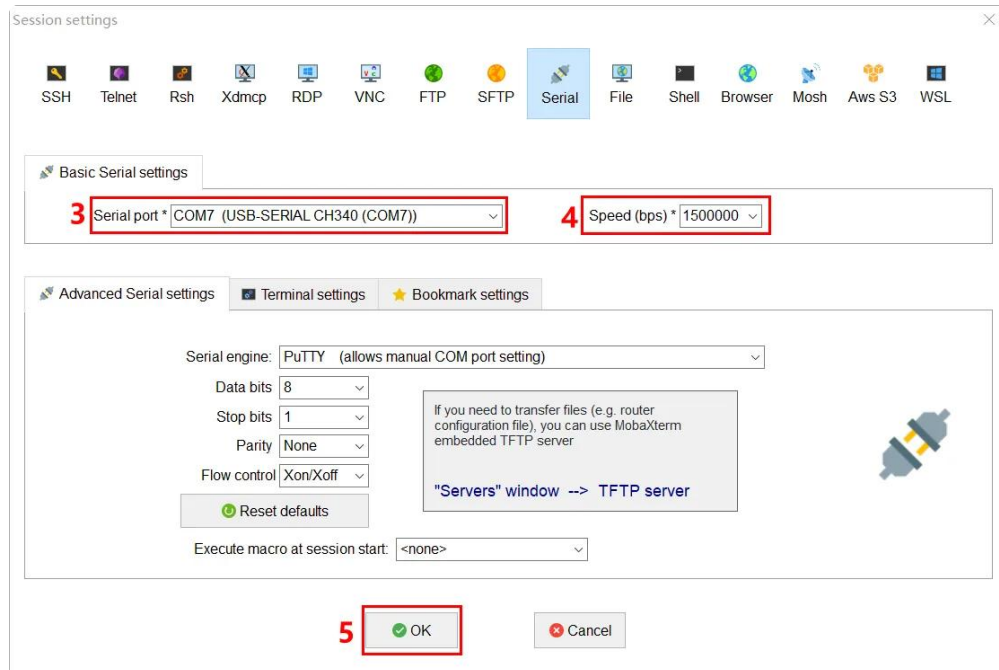
2. 选择sessionSerial, 如下图所示:



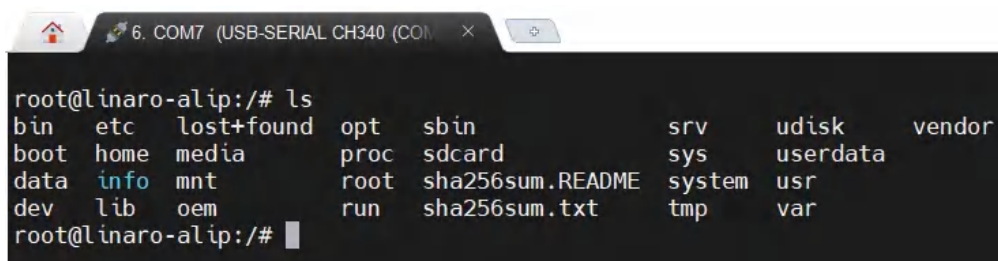
3. 将Serial port修改为在设备管理器中找到的COM端口

4. 设置Speed(bps)为1500000

5. 点击【OK】按钮，如下图所示：

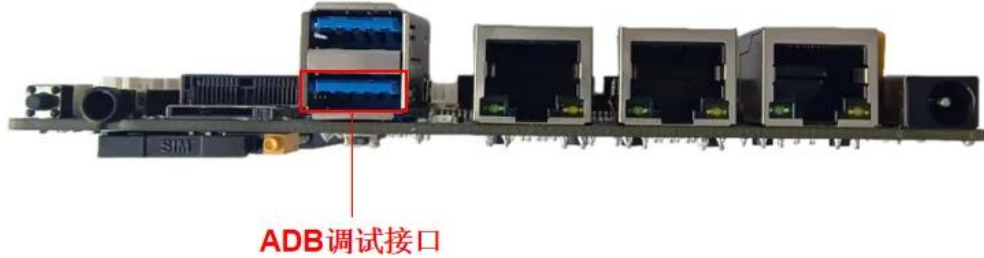


结果如下图所示：



1.2 adb调试

主板adb接口位于下方的USB口，与烧录接口为同一接口，如下图所示：



adb默认使用root账户登录，如下图所示：

```
C:\WINDOWS\system32\cmd.exe - adb shell
root@linaro-alip:/# ls
bin    etc    lost+found  opt    sbin          srv    udisk    vendor
boot  home  media      proc   sdcard        sys    userdata
data  info  mnt        root   sha256sum. README system  usr
dev   lib   oem        run    sha256sum.txt tmp      var
root@linaro-alip:/#
```

1.3 ssh调试

主板支持ssh调试，默认登录账号密码为：

1. 账号：linaro
2. 密码：linaro
3. 端口：22

ssh调试，如下图所示：

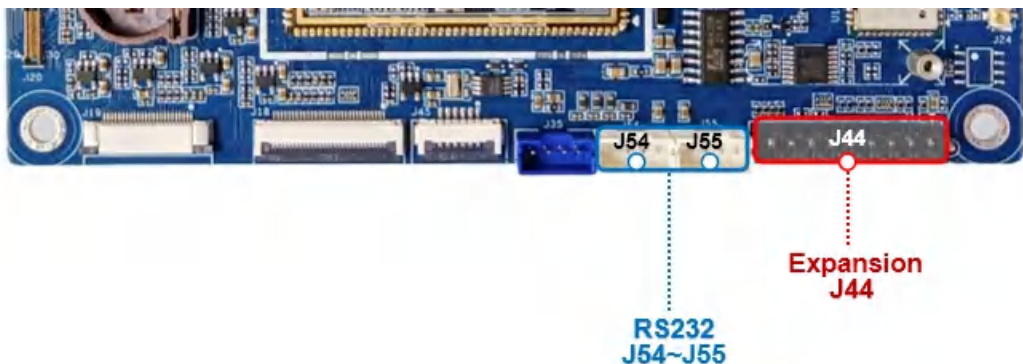
```
linaro@linaro-alip: ~
login as: linaro
linaro@192.168.1.125's password: linaro
Linux linaro-alip 5.10.160 #37 SMP Wed Sep 6 19:34:35 CST 2023 aarch64

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

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
linaro@linaro-alip:~$ ls
Desktop
linaro@linaro-alip:~$
```

2 串口

主板共配置3路串口（除调试串口），其中2路RS232，1路TTL，如下图所示：



序号	接口位置	电平类型	串口设备节点	备注
1	J44	TTL	/dev/ttyS4	/
2	J55	RS232	/dev/ttyS6	连接到扩展端口J44为TTL
3	J54	RS232	/dev/ttyS8	连接到扩展端口J44为TTL

使用microcom可以进行串口收发测试，命令如下：

▼

Shell

```

1 root@linaro-alip:/# apt-get update
2 root@linaro-alip:/# apt-get install microcom
3 root@linaro-alip:/# microcom -s 115200 /dev/ttyS6

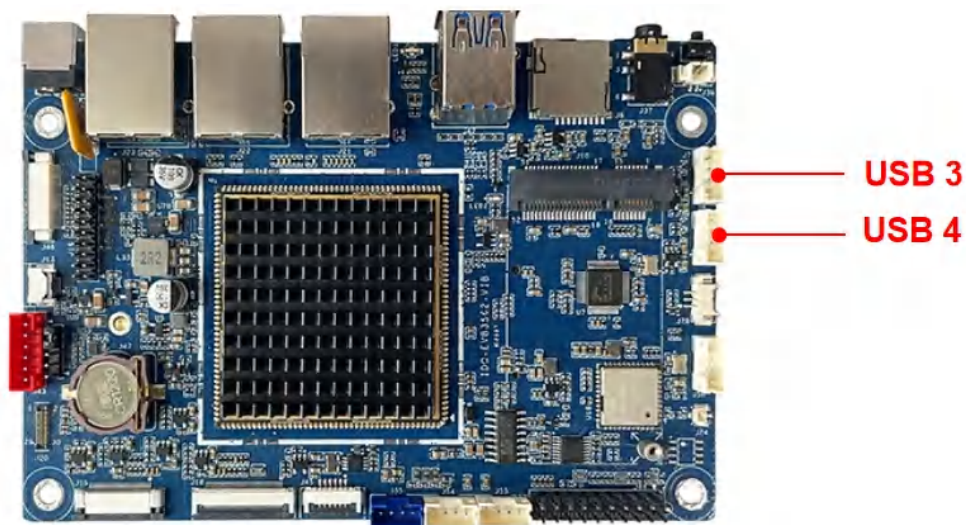
```

注意：测试完成，按Ctrl+x退出。

3 USB

主板共配置4路USB接口，分别记作USB1-4，如下图所示：





序号	接口位置	类型
USB1	J24下	OTG
USB2	J24上	USB2.0 HOST
USB3	J7	USB2.0 HOST
USB4	J8	USB2.0 HOST

其中USB1接电脑上电时，默认为device模式（可使用adb调试）；接U盘或者其他存储设备上电时，默认为host模式（可读取U盘）。

当usb host接上U盘等存储设备时，默认挂载到/mnt/udisk目录，命令如下：

▼

Shell

```

1  root@linaro-alip:/# mount
2  ...
3  /dev/sda1 on /mnt/udisk type vfat ...

```

4 Micro SD

主板共配置一路Micro SD接口，插入SD卡后，默认挂载到/mnt/sdcard目录，如过为自动挂载可以手动挂载SD卡，命令如下：

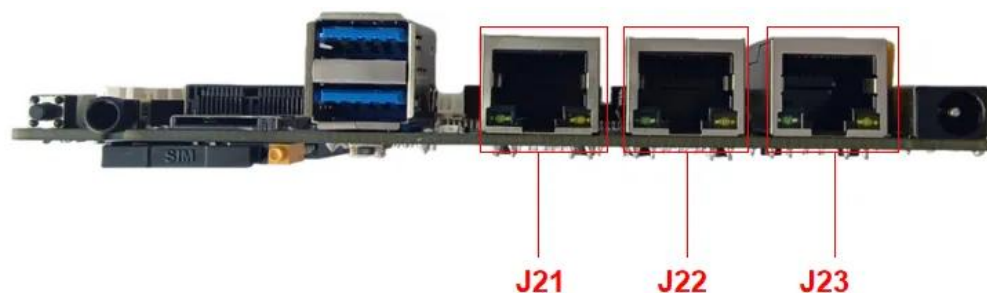
```

1  #找到SD卡节点
2  root@linaro-alip:/# fdisk -l
3  root@linaro-alip:/# mkdir test
4  root@linaro-alip:/# mount /dev/mmcblk1p1 /test/
5  root@linaro-alip:/# df -h
6  文件系统          容量  已用  可用  已用%  挂载点
7  /dev/root          12G   2.7G   8.8G   24%  /
8  devtmpfs            974M     0   974M     0%  /dev
9  tmpfs              985M     0   985M     0%  /dev/shm
10 tmpfs               394M   1.4M   393M     1%  /run
11 tmpfs               5.0M   4.0K   5.0M     1%  /run/lock
12 tmpfs              985M    16K   985M     1%  /tmp
13 /dev/mmcblk2p7      121M    12M   101M    11%  /oem
14 /dev/mmcblk2p6      2.0G    44K   1.8G     1%  /userdata
15 tmpfs              197M    40K   197M     1%  /run/user/1000
16 /dev/mmcblk1p1       15G    8.0K    15G     1%  /test
17
18 #卸载挂载点
19 root@linaro-alip:/# umount /test/

```

5 Ethernet

主板共配置3路以太网接口，包括2路千兆网和1路百兆网，如下图所示：



序号	接口位置	速率	网络节点
1	J21	千兆	eth1
2	J22	千兆	eth0
3	J23	百兆	eth2

系统默认开启DHCP服务，动态获取IP。可通过ifconfig指令设置临时静态IP，命令如下：

▼ Shell

```
1 root@linaro-alip:/# ifconfig eth0 192.168.1.123
```

如需设置永久静态IP，则可通过修改/etc/network/interfaces，命令如下：

▼ Shell

```
1 # interface file auto-generated by buildroot
2
3 auto lo
4 iface lo inet loopback
5
6 auto eth0
7     iface eth0 inet static
8     address 192.168.0.123
9     netmask 255.255.255.0
10    gateway 192.168.0.1
11    nameserver 192.168.0.1
12
13 auto eth1
14     iface eth1 inet static
15     address 192.168.1.123
16     netmask 255.255.255.0
17     gateway 192.168.1.1
18     nameserver 192.168.1.1
19
```

主板重启，命令如下：

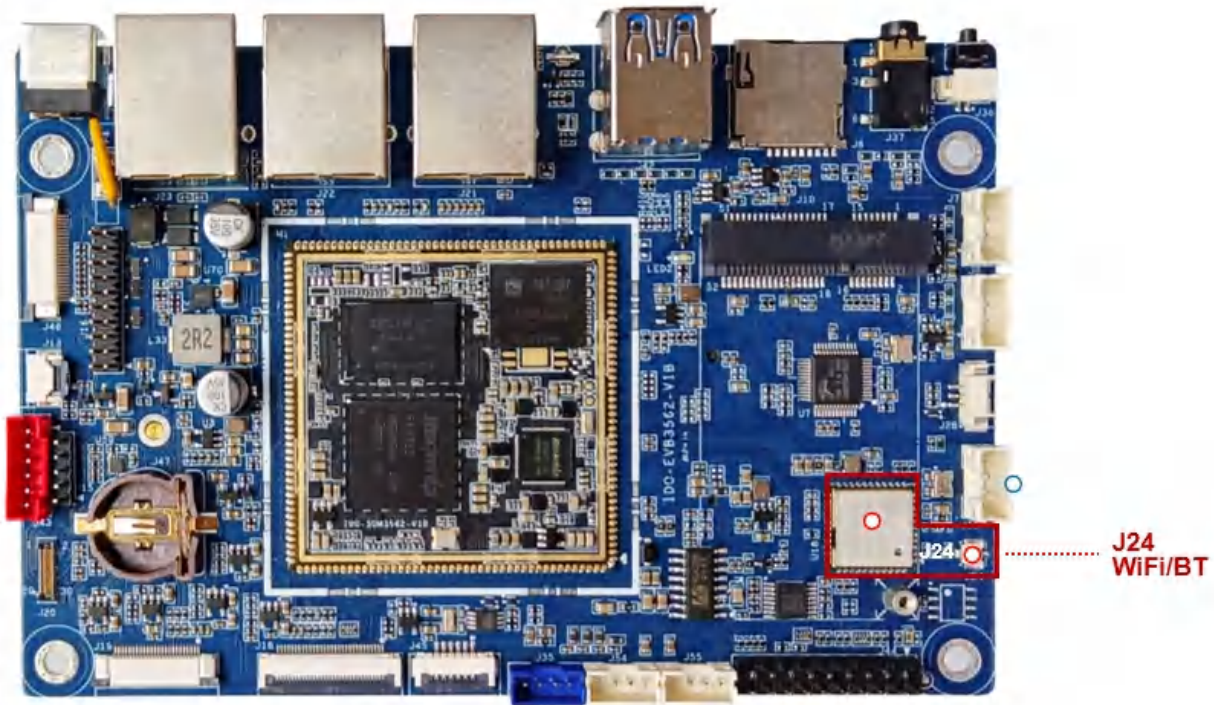
▼ Shell

```
1 root@linaro-alip:/# reboot
```

即使主板断电重启，此静态IP设置仍然生效。

6 WiFi

主板配置一个2.4G/5G 双频WiFi模块，型号是AP6256，如下图所示：



系统启动会默认打开WiFi，对应的网络节点为wlan0，命令如下：

```
root@linaro-alip:/# ifconfig wlan0
wlan0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
        ether 50:41:1c:5d:02:70 txqueuelen 1000 (Ethernet)
        RX packets 0 bytes 0 (0.0 B)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 0 bytes 0 (0.0 B)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

6.1 连接热点

6.1.1 命令行连接

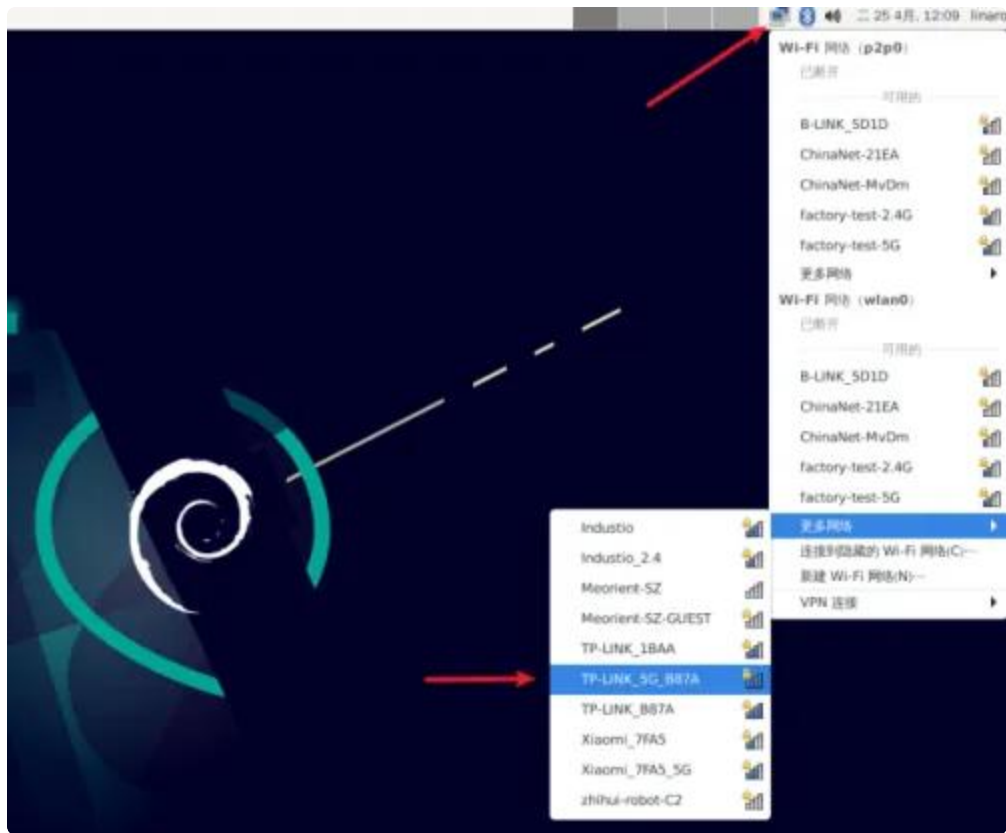
系统提供了一个连接WiFi热点的脚本，位于/usr/bin/wifi_start.sh，命令如下：

```
root@linaro-alip:/# /usr/bin/wifi_start.sh TP-LINK_B87A 12345678
```

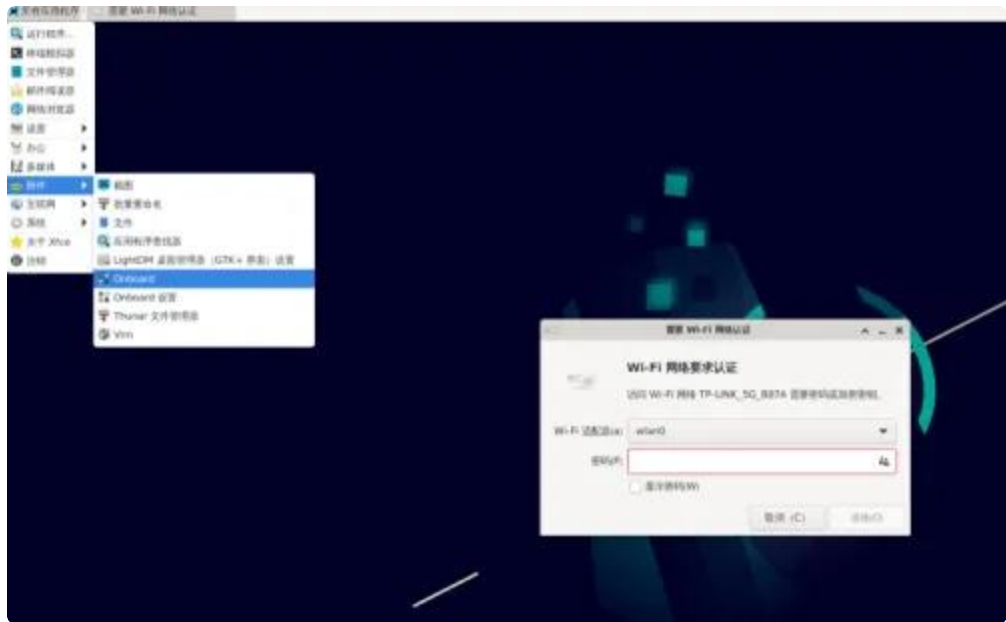
其中"TP-LINK_B87A"为热点名称，"12345678"为热点密码。

6.1.2 桌面连接

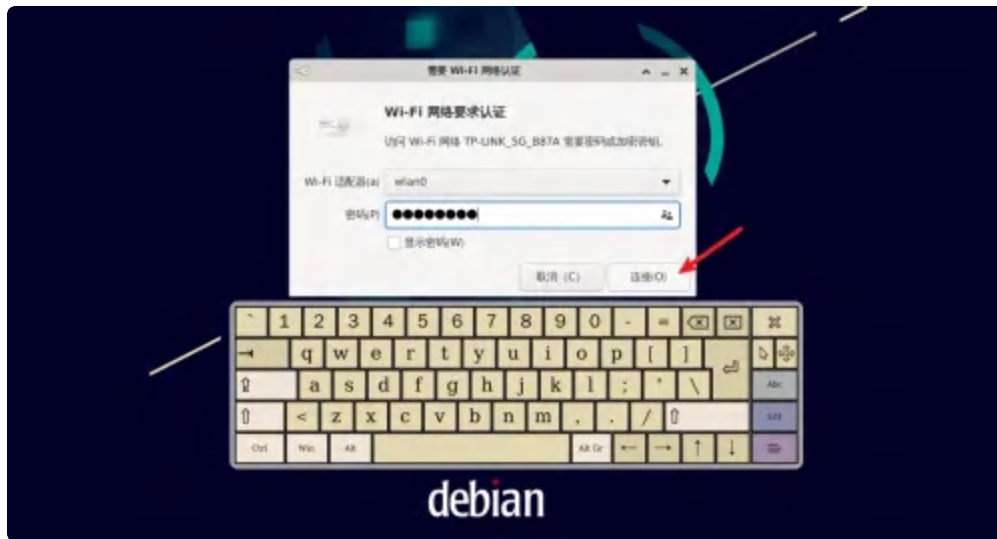
点击桌面右上角的网络按钮，弹出的列表中选择要连接的热点，如下图所示：



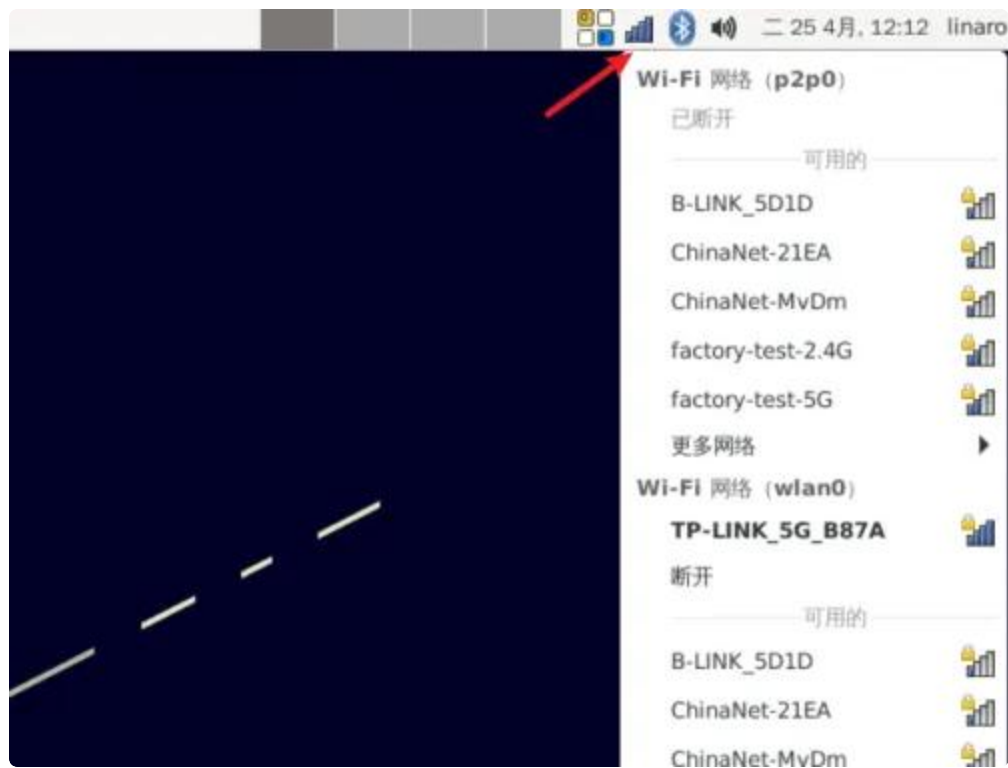
弹出密码输入框，使用键盘输入密码（如果没有接键盘，可以使用软键盘Onboard），如下图所示：



输入密码后，点击【连接】按钮连接热点，如下图所示：



通过再次点击桌面右上角网络按钮，确认是否连接成功，如下图所示：



或者通过ifconfig 命令查看wlan0的IP地址确认，命令如下：

```

1 root@linaro-alip:/# ifconfig wlan0
2 wlan0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
3 ether 50:41:1c:5d:02:70 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

```

7 Bluetooth

主板蓝牙版本为BT5.0。

7.1 开启蓝牙

开启蓝牙，命令如下：

```
▼ Shell |
1 root@linaro-alip:/# hciconfig
2 hci0:   Type: Primary  Bus: UART
3         BD Address: 50:41:1C:5D:02:87  ACL MTU: 1021:8  SCO MTU: 64:1
4         UP RUNNING
5         RX bytes:871 acl:0 sco:0 events:61 errors:0
6         TX bytes:3282 acl:0 sco:0 commands:61 errors:0
7
8
```

7.2 扫描设备

扫描设备，命令如下：

```
▼ Shell |
1 root@linaro-alip:/# hciconfig hci0 iscan
2 root@linaro-alip:/# hcitool scan
3 Scanning ...
```

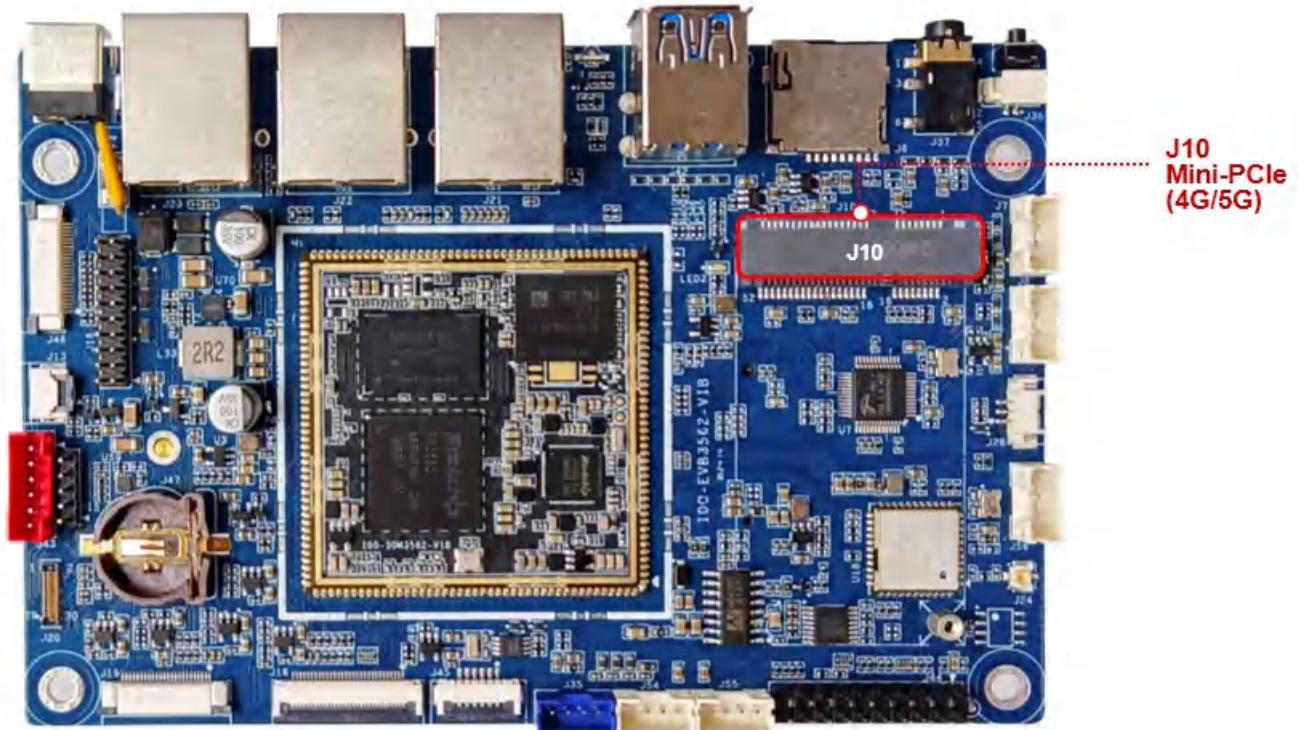
7.3 连接蓝牙设备

连接蓝牙设备，命令如下：

```
▼ Shell |
1 root@linaro-alip:/# bluetoothctl
2 [bluetooth]# scan on
3 ...
4 [bluetooth]# trust 7C:C1:80:09:DD:6C
5 [bluetooth]# pair 7C:C1:80:09:DD:6C
6 [cainiaocl]# exit
```

8 4G

主板配置了一路4G接口，默认适配EC20模块，如下图所示：



正确按照4G模块、SIM卡和4G天线后，使用quectel-CM工具拨号，命令如下：

```
1 root@linaro-alip:/# quectel-CM &
```

拨号成功，将产生wwan0网络节点，命令如下：

```
1 root@linaro-alip:/# ifconfig wwan0
2 wwan0      Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
3            inet addr:10.122.45.92  P-t-P:10.122.45.92  Mask:255.255.255.248
4            UP POINTOPOINT RUNNING NOARP MULTICAST MTU:1500  Metric:1
5            RX packets:12 errors:0 dropped:0 overruns:0 frame:0
6            TX packets:59 errors:0 dropped:0 overruns:0 carrier:0
7            collisions:0 txqueuelen:1000
8            RX bytes:1396 (1.3 KiB)  TX bytes:8214 (8.0 KiB)
9
```

测试4G上网功能，命令如下：

```
1 root@linaro-alip:/# ping www.baidu.com -I wwan0
2 PING www.a.shifen.com (14.119.104.189) from 10.122.45.92 wwan0: 56(84) byte
  s of data.
3 64 bytes from 14.119.104.189 (14.119.104.189): icmp_seq=1 ttl=52 time=95.5
  ms
4 64 bytes from 14.119.104.189 (14.119.104.189): icmp_seq=2 ttl=52 time=99.3
  ms
5 64 bytes from 14.119.104.189 (14.119.104.189): icmp_seq=3 ttl=52 time=97.8
  ms
6
```

9 音频

主板配置了一路声卡Rockchip RK809。

9.1 喇叭

双声道Lineout接口位于J35，如下图所示：



打开/关闭Lineout开关，命令如下：

```
1 //关闭Lineout
2 root@linaro-alip:/# amixer cset numid=4,iface=MIXER,name='Master Playback S
  witch' 0
3
4 //打开Lineout
5 root@linaro-alip:/# amixer cset numid=4,iface=MIXER,name='Master Playback S
  witch' 1
```

播放音频，命令如下：

```
1 root@linaro-alip:/# aplay ./root/8k16bpsStereo.wav
```

调节音量大小，命令如下：

```
1 //音量调节至10%
2 amixer cset numid=5,iface=MIXER,name='Master' 10
3
4 //音量调节至60%
5 amixer cset numid=5,iface=MIXER,name='Master' 60
```

9.2 MIC

MIC如下图所示：



打开MIC开关，命令如下：

```
1 amixer cset numid=2,iface=MIXER,name='Capture MIC Path' 'Main Mic'
```

录音，命令如下：

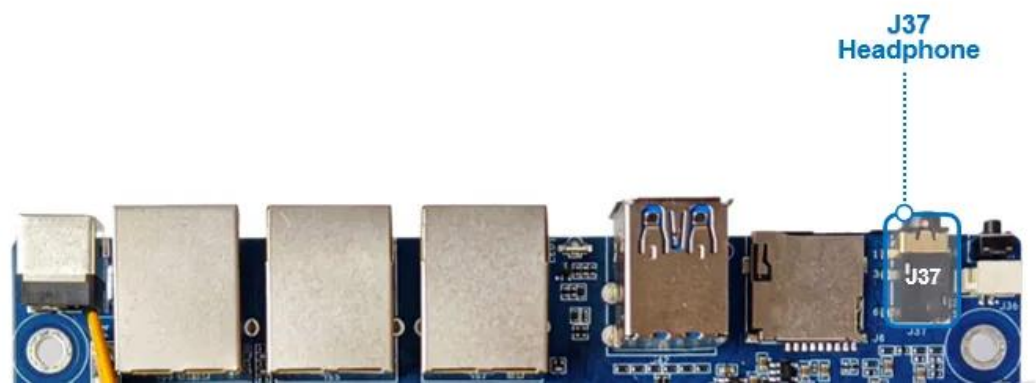
```
1 arecord -D hw:0,0 -r 48000 -c 2 -f S16_LE test.wav
```

播放，命令如下：

```
1 aplay -Dplughw:0,0 ./test.wav
```

9.3 耳机

耳机如下图所示：



打开/关闭耳机音频输出开关，命令如下：

▼ Shell |

```
1 //关闭耳机输出
2 root@linaro-alip:/# amixer cset numid=4,iface=MIXER,name='Master Playback Switch' 0
3
4 //打开耳机输出
5 root@linaro-alip:/# amixer cset numid=4,iface=MIXER,name='Master Playback Switch' 1
```

播放音频，命令如下：

▼ Shell |

```
1 root@linaro-alip:/# aplay ./root/8k16bpsStereo.wav
```

调节音量大小，命令如下：

▼ Shell |

```
1 amixer cset numid=3,iface=MIXER,name='Master Playback Volume' 10000,10000
```

注意：音量大小范围是0-65536。

10 Camera

主板配置了2路摄像头，一路为GC8034（J18，3264x2448），另一路为OV13855（J20，4224x3136），两路摄像头对应的设备节点如下：

GC8034	/dev/video22
--------	--------------

OV13855

/dev/video29

10.1 检测摄像头

```
1 root@linaro-alip:/# media-ctl -p -d /dev/media0 | grep gc8034
2     <- "m01_f_gc8034 2-0037":0 [ENABLED]
3 - entity 63: m01_f_gc8034 2-0037 (1 pad, 1 link)
4 root@linaro-alip:/# media-ctl -p -d /dev/media1 | grep ov13855
5     <- "m00_b_ov13855 1-0010":0 [ENABLED]
6 - entity 63: m00_b_ov13855 1-0010 (1 pad, 1 link)
```

10.2 预览摄像头

预览OV13855摄像头画面：

```
1 root@linaro-alip:/# gst-launch-1.0 v4l2src device=/dev/video29 ! video/x-raw, format=NV12, width=1920, height=1080, framerate=30/1 ! autovideoconvert ! autovideosink
```

10.3 抓取视频

使用v4l2-ctl工具可以抓取摄像头的视频数据流。

```

1 root@linaro-alip:/# v4l2-ctl --verbose -d /dev/video22 --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          : sRGB
12    Transfer Function   : Rec. 709
13    YCbCr/HSV Encoding: Rec. 709
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 [ 2496.916217] rockchip-mipi-csi2 ff380000.mipi0-csi2: stream on, src_sd:
00000000949a962c, sd_name:rockchip-csi2-dphy0
30 [ 2496.916253] rockchip-mipi-csi2 ff380000.mipi0-csi2: stream ON
31     VIDIOC_STREAMON returned 0 (Success)
32 ...

```

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：

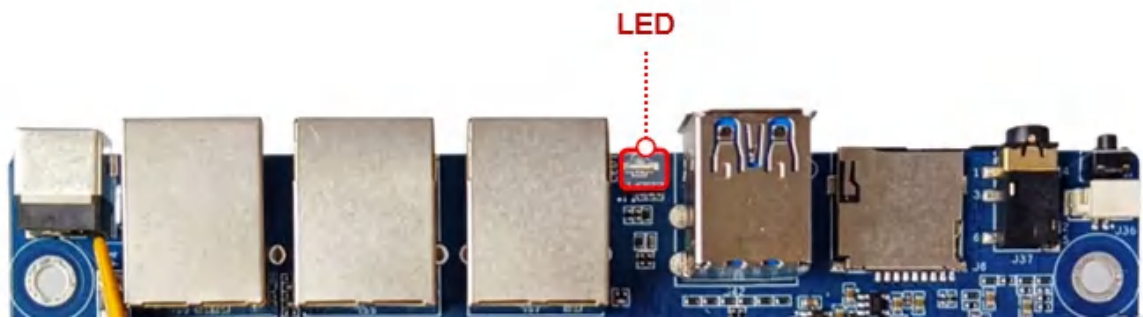
```

1 root@linaro-alip:/# ffplay -f rawvideo -video_size 1920x1080 -pix_fmt nv12
./out.yuv

```

11 LED

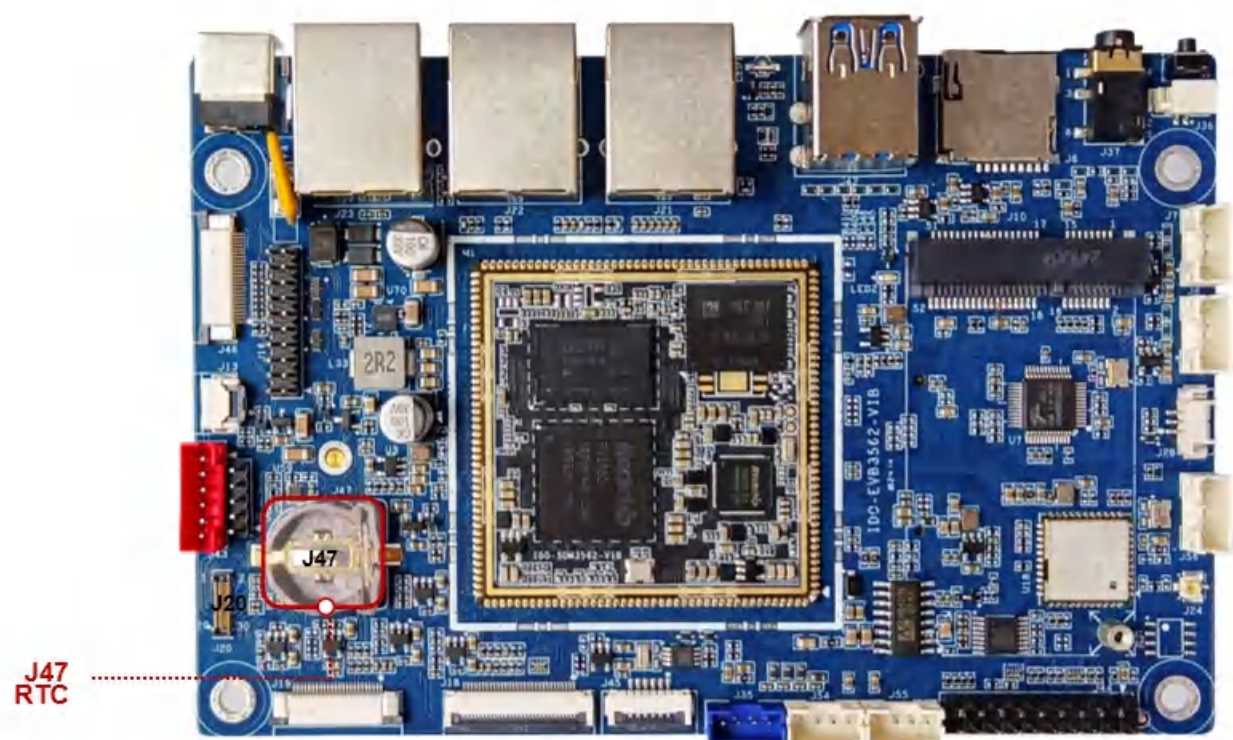
主板配置了1个LED灯，位于USB与网口在中间，如下图所示：



序号	名称	位置	颜色	功能
1	系统指示灯	下	黄色	闪烁表示系统运行正常

12 RTC

主板配置了一个外部RTC，型号为HYM8563，系统中对应的设备节点为/dev/rtc0，如下图所示：



12.1 读取RTC时间

读取RTC时间，命令如下：

```
1 root@linaro-alip:/# hwclock
2 Fri Aug 4 09:02:38 2017 0.000000 seconds
```

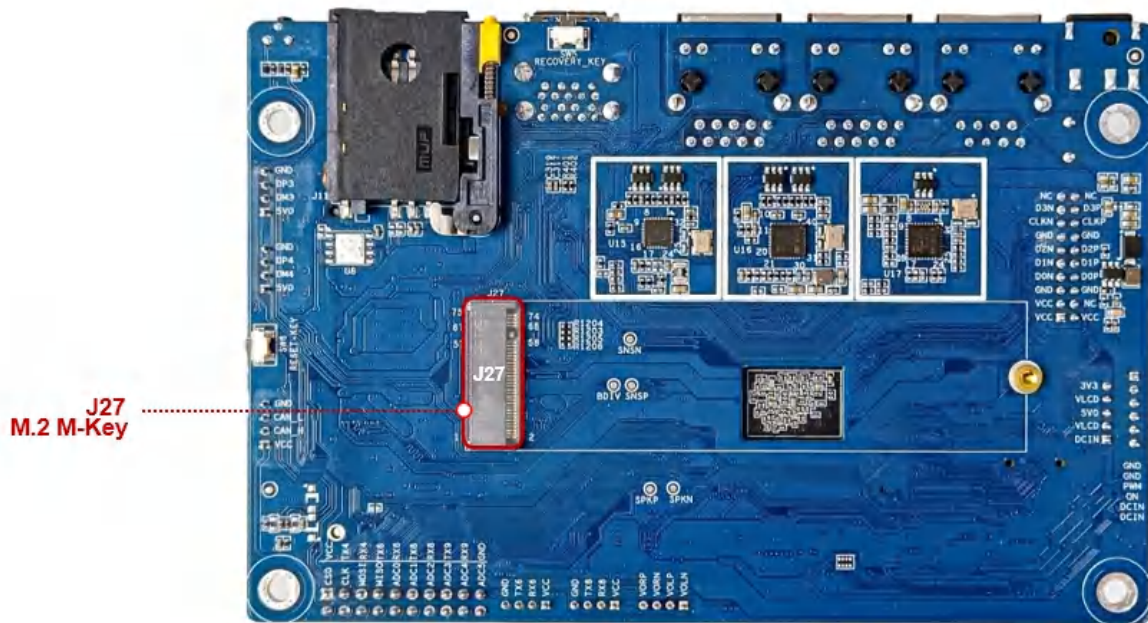
12.2 设置RTC时间

设置RTC时间，命令如下：

```
1 root@linaro-alip:/# date -s '2023-8-22 15:30:00'
2 Tue Aug 22 15:30:00 UTC 2023
3 root@rk3562-buildroot:/# hwclock -w
4 root@rk3562-buildroot:/# hwclock
5 Tue Aug 22 15:30:05 2023 0.000000 seconds
```

13 硬盘

主板配置了一路硬盘接口（M.2），位于主板背面J27，如下图所示：



使用fdisk命令可以查看硬盘设备信息，命令如下：

▼Shell

```
1 root@linaro-alip:/# fdisk -l
2 Disk /dev/nvme0n1: 932 GB, 1000204886016 bytes, 1953525168 sectors
3 121601 cylinders, 255 heads, 63 sectors/track
4 Units: sectors of 1 * 512 = 512 bytes
5
6 Device      Boot StartCHS      EndCHS          StartLBA      EndLBA      Sectors
   Size Id Type
7 /dev/nvme0n1p1  1,0,1      1023,254,63      2048 1953525134 1953523087
   931G  c Win95 FAT32 (LBA)
```

使用mount命令挂载到文件系统中，命令如下：

▼Shell

```
1 root@linaro-alip:/# mount /dev/nvme0n1p1 /mnt
2 补充
3 4GB.bin  Audiobooks  Documents  LOST.DIR  Music      Pictures  Recordin
   gs
4 Alarms   DCIM          Download   Movies     Notifications  Podcasts  Ringtone
   s
```

挂载成功后，即可进行文件读写操作。

14 扩展接口

主板配置了一排扩展接口，位于J44，扩展接口的各个引脚功能，如下表所示：

pin	功能		pin	功能
1	5V输出		2	SPI0_CS
3	UART2_TX		4	SPI0_CLK
5	UART2_RX		6	SPI0_MOSI
7	UART6_TX		8	SPI0_MISO
9	UART6_RX		10	ADC0
11	UART8_TX		12	ADC1
13	UART8_RX		14	ADC2
15	UART9_TX		16	ADC3

17	UART9_RX		18	ADC4
19	GND		20	ADC5

14.1 UART

扩展接口共包含4路UART，各路UART对应的设备节点，如下表所示：

pin	功能	设备节点
j3-2	UART2_TX	/dev/ttyS4
5	UART2_RX	
7	UART6_TX	/dev/ttyS6
9	UART6_RX	
11	UART8_TX	/dev/ttyS8
13	UART8_RX	
15	UART9_TX	/dev/ttyS9
17	UART9_RX	

14.2 SPI

系统中对应的/dev/spidev0.0。

14.3 ADC输入

扩展接口配置了6路ADC输入（参考电压为1.8V，精度10位）如下表所示：

名称	设备节点
ADCIN0	/sys/bus/iio/devices/iio:device0/in_voltage0_raw
ADCIN1	/sys/bus/iio/devices/iio:device0/in_voltage1_raw
ADCIN2	/sys/bus/iio/devices/iio:device0/in_voltage2_raw

ADCIN3	/sys/bus/iio/devices/iio:device0/in_voltage3_raw
ADCIN4	/sys/bus/iio/devices/iio:device0/in_voltage4_raw
ADCIN5	/sys/bus/iio/devices/iio:device0/in_voltage5_raw

ADC输入电压计算公式为： $V = (\text{voltage_raw}/1024) \times 1.8V$ （其中voltage_raw为从设备节点读取的值）。

以ADCIN2为例：

1. 读取in_voltage2_raw的值，命令如下：

▼

Shell

```

1  root@linaro-alip:/# fdisk -lmount /dev/nvme0n1p1 /mntcat /sys/bus/iio/devices/iio:device0/in_voltage2_raw
2  524

```

2. 计算ADCIN2的输入电压V

$$V = (\text{voltage_raw}/1024) \times 1.8V = (524/1024) \times 1.8V = 0.92V$$

14.4 SPI

扩展接口配置了1路SPI，系统中对应的设备节点为/dev/spidev0.0。

15 RealTime Linux 实时性系统

15.1 安装测试工具包

▼

Bash

```

1  $ sudo apt update
2  // 安装cpu压力测试工具
3  $ sudo apt install stress
4  // 安装rt测试工具
5  $ sudo apt-get install rt-tests

```

15.2 测试实时性性能

15.2.1 开启压力测试：

```
1 // 开启性能模式
2 $ sudo echo performance | tee $(find /sys/ -name *governor) /dev/null || true
3 // 开启压力测试
4 $ stress --cpu 4 --io 4 --vm 2 --vm-bytes 128M --timeout 3600s &
```

15.2.2 使用cyclicttest

```
1 $ sudo cyclicttest -t 4 -p 80 -m -S -D 1h
2 #参数详解:
3 -t: 指定测试线程的数量
4 -p: 设置线程的优先级（优先级范围通常在1到99之间（99 为最高优先级））
5 -m: 锁定内存
6 -S: 使用 SCHED_FIFO 调度策略（SCHED_FIFO 是一种先入先出的实时调度策略，它确保高优先级线程在运行时不会被较低优先级的线程打断，直到它主动放弃 CPU。）
7 -D: 指定测试时间（m: 分钟；h: 小时；d: 天）
```

```
linaro@linaro-alip:~$ sudo cyclicttest -t 4 -p 80 -m -S -D 1h
# /dev/cpu_dma_latency set to 0us
policy: fifo: loadavg: 5.84 5.49 5.07 5/360 30070

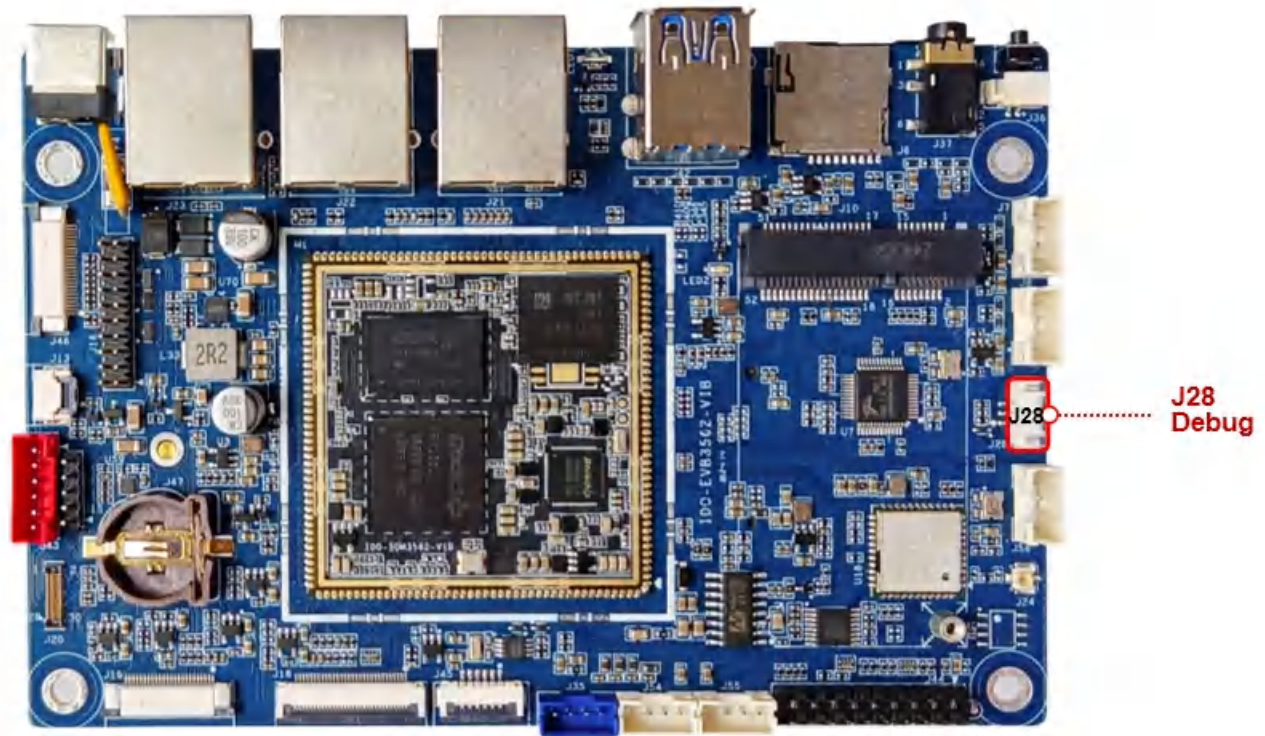
T: 0 (17789) P:80 I:1000 C:2235230 Min:      4 Act:      5 Avg:      4 Max:      25
T: 1 (17790) P:80 I:1500 C:1490143 Min:      4 Act:      4 Avg:      4 Max:      34
T: 2 (17791) P:80 I:2000 C:1117591 Min:      4 Act:      4 Avg:      4 Max:      36
T: 3 (17792) P:80 I:2500 C: 894062 Min:      4 Act:      5 Avg:      5 Max:      26
```

IDO-EVB3562-Buildroot系统

1 调试

1.1 串口调试

主板调试串口位于USB座子下方J28，建议使用配套的usb转串口工具，如下图所示：



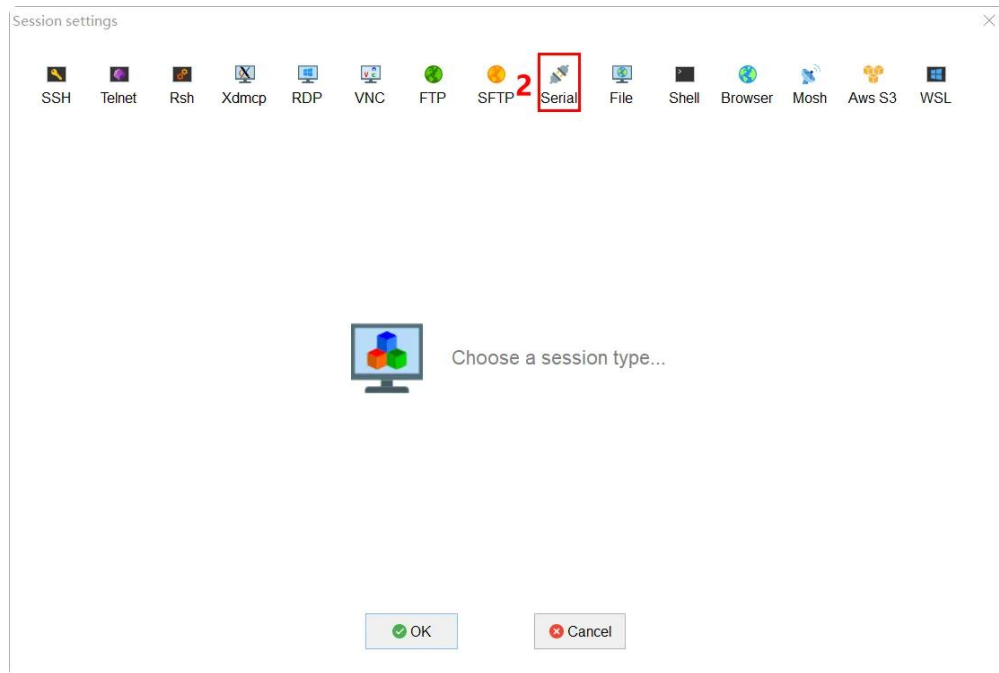
1. 打开MobaXterm, 下载链接如下:

链接: https://pan.baidu.com/s/11ui4LTd2mq_9kiJpeL4bWg?pwd=1234

提取码: 1234

☐ 文件名	修改时间	类型	大小
☐ RKDevTool_Release_v2.95.zip	2024-04-24 11:53	zip文件	2.30MB
☐ other_tools.txt	2024-04-25 15:31	txt文件	44B
☒ MobaXterm_Portable_v23.6.zip	2024-04-24 14:30	zip文件	39.99MB
☐ DriverAssitant_v5.11.zip	2024-04-24 11:52	zip文件	9.36MB

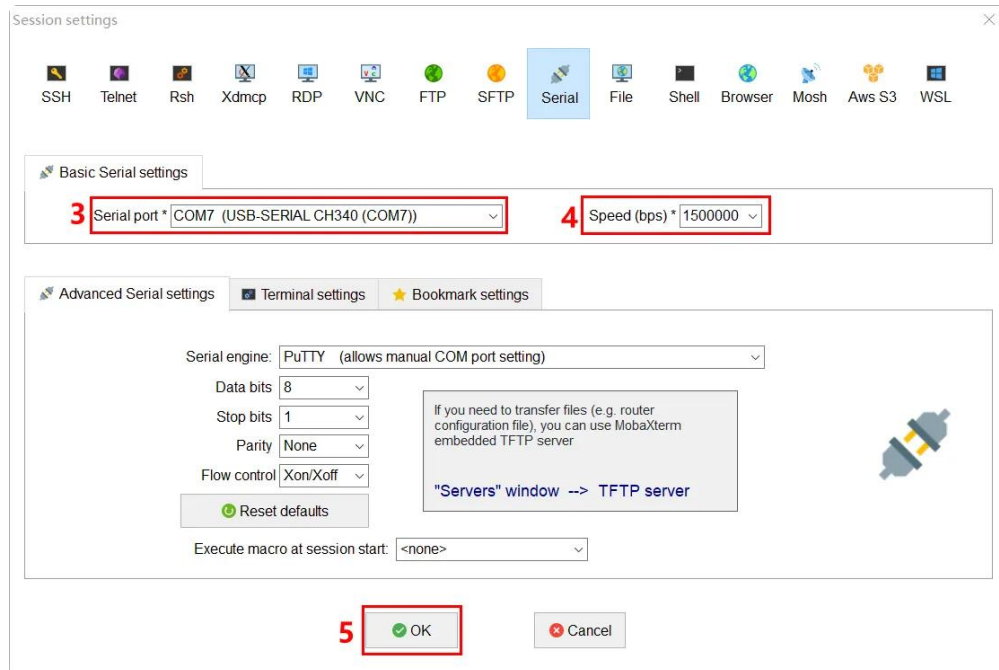
2. 选择session为Serial, 如下图所示:



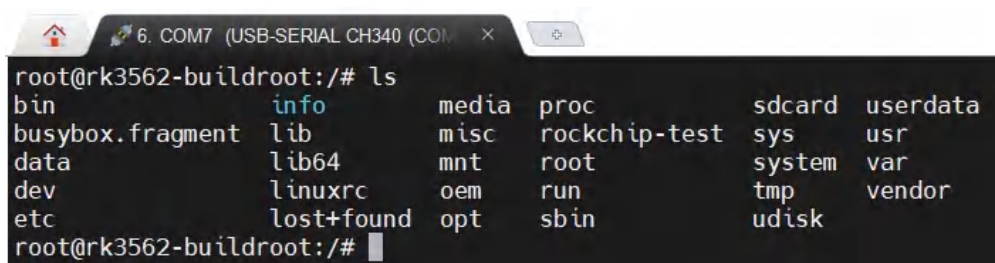
3. 将Serial port修改为在设备管理器中找到的COM端口

4. 设置Speed(bps)为1500000

5. 点击【OK】按钮，如下图所示：

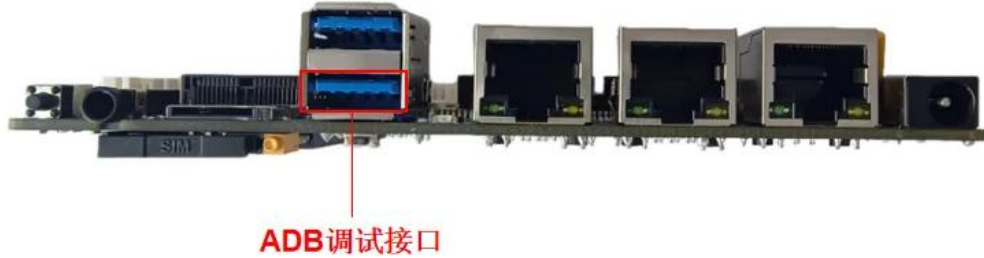


结果如下图所示：



1.2 adb调试

主板adb接口位于J24下方的USB口，与烧录接口为同一接口，如下图所示：



adb默认使用root账户登录，如下图所示：

```
C:\WINDOWS\system32\cmd.exe - adb shell
root@rk3562-buildroot:/# ls
bin          info         media        proc         sdcard      userdata
busybox.fragment  lib         misc         rockchip-test sys         usr
data         lib64        mnt          root         system      var
dev          linuxrc      oem          run          tmp         vendor
etc          lost+found  opt          sbin         udisk
root@rk3562-buildroot:/#
```

1.3 ssh调试

主板支持ssh调试，默认登录账号密码为：

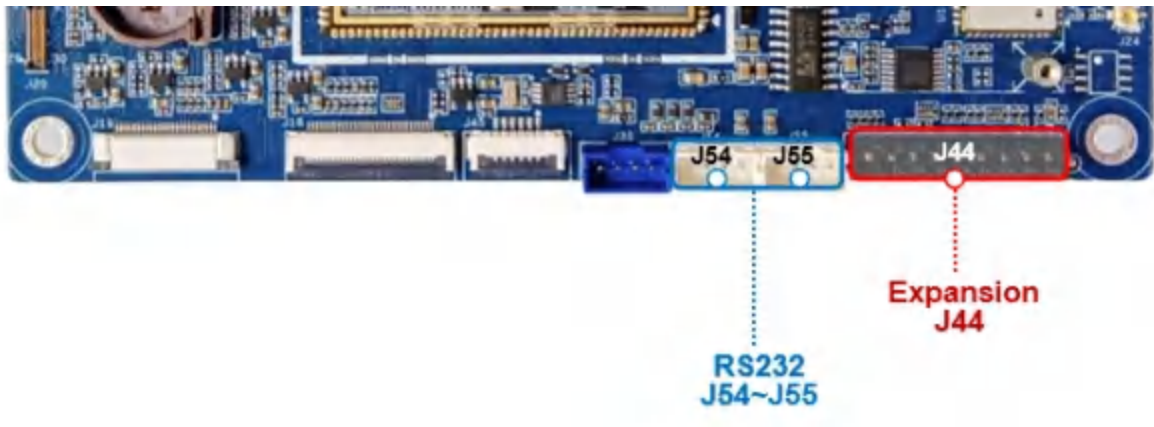
1. 账号：root
2. 密码：rockchip
3. 端口：22

ssh调试，如下图所示：

```
192.168.1.179 - PuTTY
login as: root
root@192.168.1.179's password: rockchip
root@rk3562-buildroot:~# ls
bcmhdh.ko      fw_bcm43456c5_ag.bin  iperf3      nvram_ap6256.txt
dhd_static_buf.ko  fw_bcm43456c5_ag_mfg.bin  libiperf.so.0
```

2 串口

主板共配置3路串口（除调试串口），其中2路RS232，1路TTL，如下图所示：



序号	接口位置	电平类型	串口设备节点	备注
1	J44	TTL	/dev/ttyS4	/
2	J55	RS232	/dev/ttyS6	连接到扩展端口J44为TTL
3	J54	RS232	/dev/ttyS8	连接到扩展端口J44为TTL

使用microcom可以进行串口收发测试，命令如下：

▼

Shell

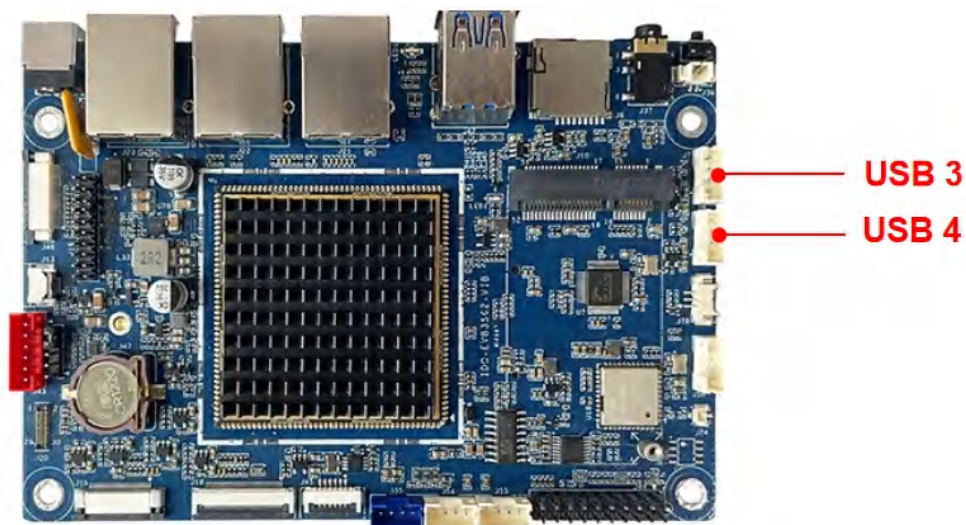
```
1 root@rk3562-buildroot:/# microcom -s 115200 /dev/ttyS6
```

注意：测试完成，按Ctrl+x退出。

3 USB

主板共配置4路USB接口，分别记作USB1-4，如下图所示：





序号	接口位置	类型
USB1	J24下	OTG
USB2	J24上	USB2.0 HOST
USB3	J7	USB2.0 HOST
USB4	J8	USB2.0 HOST

其中USB1接电脑上电时，默认为device模式（可使用adb调试）；接U盘或者其他存储设备上电时，默认为host模式（可读取U盘）。

当usb host接上U盘等存储设备时，默认挂载到/mnt/udisk目录，命令如下：

▼

Shell

```

1  root@rk3562-buildroot:/# mount
2  ...
3  /dev/sda1 on /mnt/udisk type vfat ...

```

4 Micro SD

主板共配置一路Micro SD接口。插入SD卡后，默认挂载到/mnt/sdcard目录，命令如下：

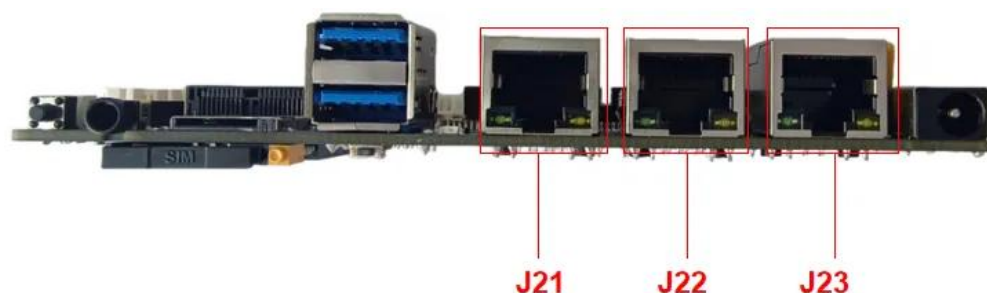
```

1 root@rk3562-buildroot:/# df -h
2 Filesystem      1M-blocks  Used Available Use% Mounted on
3 /dev/root        667      421      194   69% /
4 devtmpfs         974        1      974    1% /dev
5 tmpfs            985        1      985    1% /tmp
6 tmpfs            985        1      985    1% /run
7 tmpfs            985        0      985    0% /dev/shm
8 /dev/mmcblk2p7   121      12       101   11% /oem
9 /dev/mmcblk2p6   1946        1     1828    1% /userdata
10 /dev/mmcblk1p1  14892        1    14892    1% /mnt/sdcard

```

5 Ethernet

主板共配置3路以太网接口，包括2路千兆网和1路百兆网，如下图所示：



序号	接口位置	速率	网络节点
1	J21	千兆	eth1
2	J22	千兆	eth0
3	J23	百兆	eth2

系统默认开启DHCP服务，动态获取IP，可通过ifconfig指令设置临时静态IP，命令如下：

```

1 root@rk3562-buildroot:/# ifconfig eth0 192.168.1.123

```

如需设置永久静态IP，则可通过修改/etc/network/interfaces，命令如下：

▼ Shell

```
1  # interface file auto-generated by buildroot
2
3  auto lo
4  iface lo inet loopback
5
6  auto eth0
7      iface eth0 inet static
8      address 192.168.0.123
9      netmask 255.255.255.0
10     gateway 192.168.0.1
11     nameserver 192.168.0.1
12
13  auto eth2
14     iface eth2 inet static
15     address 192.168.1.123
16     netmask 255.255.255.0
17     gateway 192.168.1.1
18     nameserver 192.168.1.1
19
```

开发板重启，命令如下：

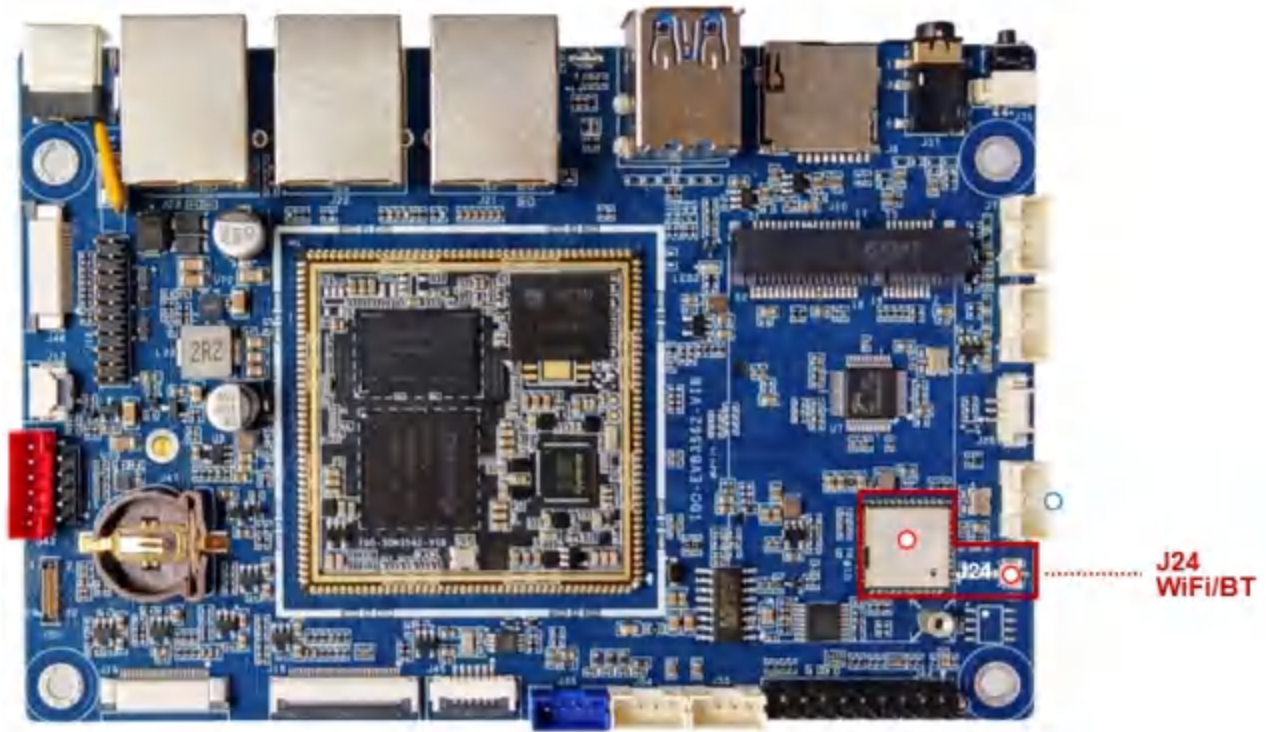
▼ Shell

```
1  root@rk3562-buildroot:/# /etc/init.d/S40network restart
```

即使主板断电重启，此静态IP设置仍然生效。

6 WiFi

主板配置一个2.4G/5G 双频WiFi模块，型号是AP6256，如下图所示：



6.1 扫描热点

扫描热点，命令如下：

```
1 root@rk3562-buildroot:/# iwlist wlan0 scan
```

6.2 连接热点

系统提供了一个连接WiFi热点的脚本，位于/usr/bin/wifi_start.sh，使用方法如下，命令如下：

```
1 root@rk3562-buildroot:/# /usr/bin/wifi_start.sh TP-LINK_B87A 12345678
```

其中"TP-LINK_B87A"为热点名称，"12345678"为热点密码。

7 Bluetooth

主板蓝牙版本为BT5.0。

7.1 开启蓝牙

开启蓝牙，命令如下：

```
▼ Shell |
1 root@rk3562-buildroot:/# hciconfig
2 hci0:   Type: Primary   Bus: UART
3         BD Address: 50:41:1C:5D:02:87   ACL MTU: 1021:8   SCO MTU: 64:1
4         UP RUNNING
5         RX bytes:871 acl:0 sco:0 events:61 errors:0
6         TX bytes:3282 acl:0 sco:0 commands:61 errors:0
7
8
```

7.2 扫描设备

扫描设备，命令如下：

```
▼ Shell |
1 root@rk3562-buildroot:/# hciconfig hci0 iscan
2 root@rk3562-buildroot:/# hcitool scan
3 Scanning ...
```

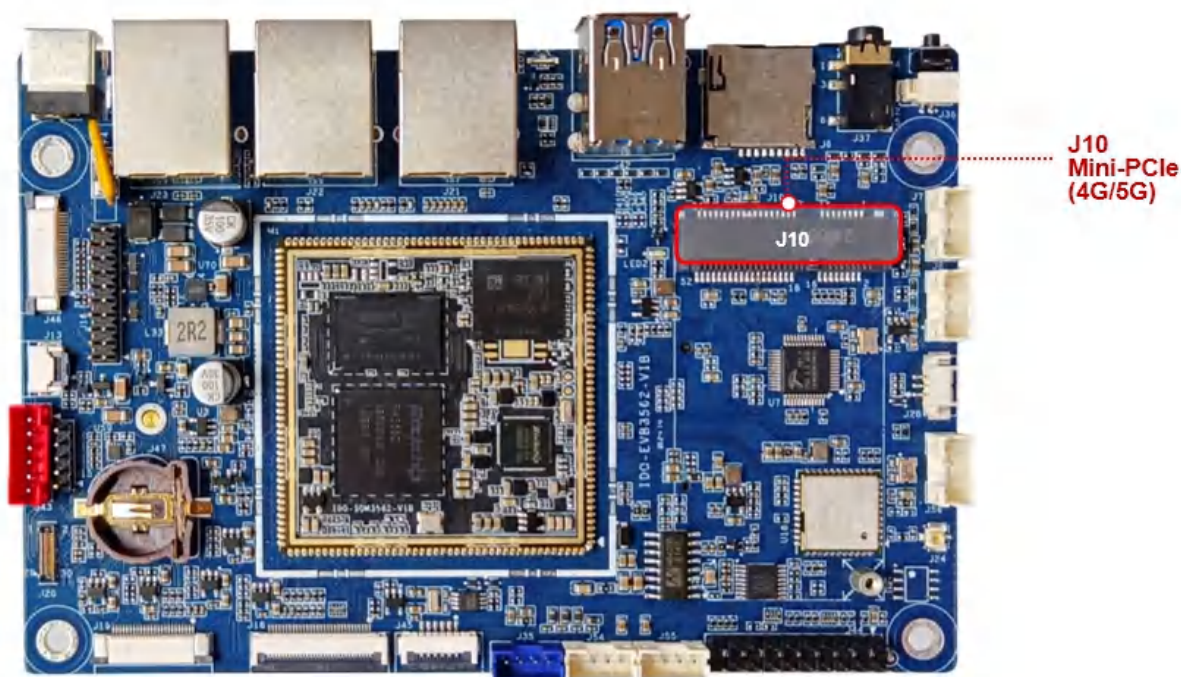
7.3 连接蓝牙设备

连接蓝牙设备，命令如下：

```
▼ Shell |
1 root@rk3562-buildroot:/# bluetoothctl
2 [bluetooth]# scan on
3 ...
4 [bluetooth]# trust 7C:C1:80:09:DD:6C
5 [bluetooth]# pair 7C:C1:80:09:DD:6C
6 [cainiaocl]# exit
```

8 4G

主板配置了一路4G接口，默认适配EC20模块，如下图所示：



正确按照4G模块、SIM卡和4G天线后，使用quectel-CM工具拨号，命令如下：

```
▼ Shell |
1 root@rk3562-buildroot:/# quectel-CM &
```

拨号成功，将产生wwan0网络节点，命令如下：

```
▼ Shell |
1 root@rk3562-buildroot:/# ifconfig wwan0
2 wwan0      Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
3           inet addr:10.122.45.92  P-t-P:10.122.45.92  Mask:255.255.255.248
4           UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
5           RX packets:12 errors:0 dropped:0 overruns:0 frame:0
6           TX packets:59 errors:0 dropped:0 overruns:0 carrier:0
7           collisions:0 txqueuelen:1000
8           RX bytes:1396 (1.3 KiB)  TX bytes:8214 (8.0 KiB)
9
```

测试4G上网功能，命令如下：

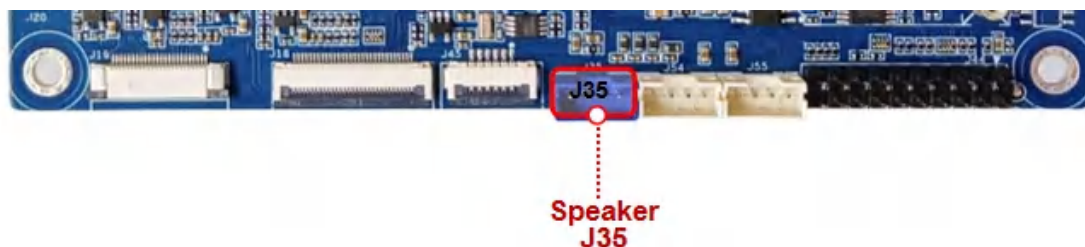
```
1 root@rk3562-buildroot:/# ping www.baidu.com -I wwan0
2 PING www.a.shifen.com (14.119.104.189) from 10.122.45.92 wwan0: 56(84) bytes of data.
3 64 bytes from 14.119.104.189 (14.119.104.189): icmp_seq=1 ttl=52 time=95.5 ms
4 64 bytes from 14.119.104.189 (14.119.104.189): icmp_seq=2 ttl=52 time=99.3 ms
5 64 bytes from 14.119.104.189 (14.119.104.189): icmp_seq=3 ttl=52 time=97.8 ms
6
```

9 音频

主板配置了一路声卡Rockchip RK809。

9.1 喇叭

双声道Lineout接口位于J35，如下图所示：



打开/关闭Lineout开关，命令如下：

```
1 //关闭Lineout
2 root@rk3562-buildroot:/# amixer cset numid=4,iface=MIXER,name='Master Playback Switch' 0
3
4 //打开Lineout
5 root@rk3562-buildroot:/# amixer cset numid=4,iface=MIXER,name='Master Playback Switch' 1
```

播放音频，命令如下：

```
1 root@rk3562-buildroot:/# aplay ./root/8k16bpsStereo.wav
```

调节音量大小，命令如下：

```
1 //音量调节至10%
2 amixer cset numid=5,iface=MIXER,name='Master' 10
3
4 //音量调节至60%
5 amixer cset numid=5,iface=MIXER,name='Master' 60
```

8.2 MIC

MIC如下图所示：



打开MIC开关，命令如下：

```
1 amixer cset numid=2,iface=MIXER,name='Capture MIC Path' 'Main Mic'
```

录音，命令如下：

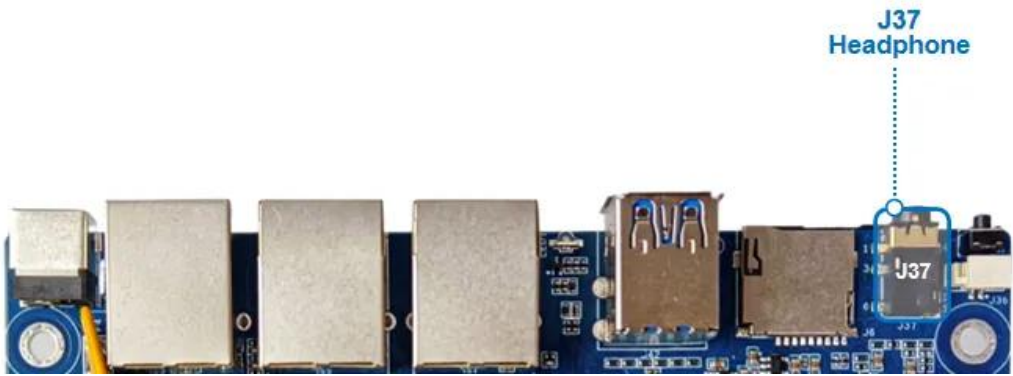
```
1 arecord -D hw:0,0 -r 48000 -c 2 -f S16_LE test.wav
```

播放，命令如下：

```
1 aplay -Dplughw:0,0 ./test.wav
```

8.3 耳机

耳机如下图所示：



打开/关闭耳机音频输出开关，命令如下：

▼ Shell

```
1 //关闭耳机输出
2 root@rk3562-buildroot:/# amixer cset numid=4,iface=MIXER,name='Master Playback Switch' 0
3
4 //打开耳机输出
5 root@rk3562-buildroot:/# amixer cset numid=4,iface=MIXER,name='Master Playback Switch' 1
```

播放音频，命令如下：

▼ Shell

```
1 root@rk3562-buildroot:/# aplay ./root/8k16bpsStereo.wav
```

调节音量大小，命令如下：

▼ Shell

```
1 amixer cset numid=3,iface=MIXER,name='Master Playback Volume' 10000,10000
```

注意：音量大小范围是0-65536。

10 Camera

主板配置了2路摄像头，一路为GC8034（J18，3264x2448），另一路为OV13855（J20，4224x3136），两路摄像头对应的设备节点如下：

GC8034	/dev/video22
--------	--------------

OV13855

/dev/video29

10.1 检测摄像头

```
1 root@rk3562-buildroot:/# media-ctl -p -d /dev/media0 | grep gc8034
2         <- "m01_f_gc8034 2-0037":0 [ENABLED]
3 - entity 63: m01_f_gc8034 2-0037 (1 pad, 1 link)
4 root@rk3562-buildroot:/# media-ctl -p -d /dev/media1 | grep ov13855
5         <- "m00_b_ov13855 1-0010":0 [ENABLED]
6 - entity 63: m00_b_ov13855 1-0010 (1 pad, 1 link)
```

10.2 预览摄像头

预览GC8034摄像头画面：

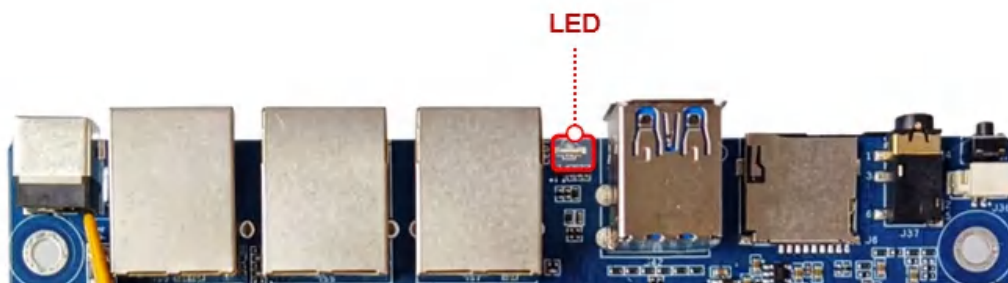
```
1 root@rk3562-buildroot:/# gst-launch-1.0 v4l2src device=/dev/video22 ! vide
o/x-raw, format=NV12, width=1920, height=1080, framerate=30/1 ! videoconver
t ! autovideosink
```

预览OV13855摄像头画面：

```
1 root@rk3562-buildroot:/# gst-launch-1.0 v4l2src device=/dev/video29 ! vide
o/x-raw, format=NV12, width=1920, height=1080, framerate=30/1 ! videoconver
t ! autovideosink
```

11 LED

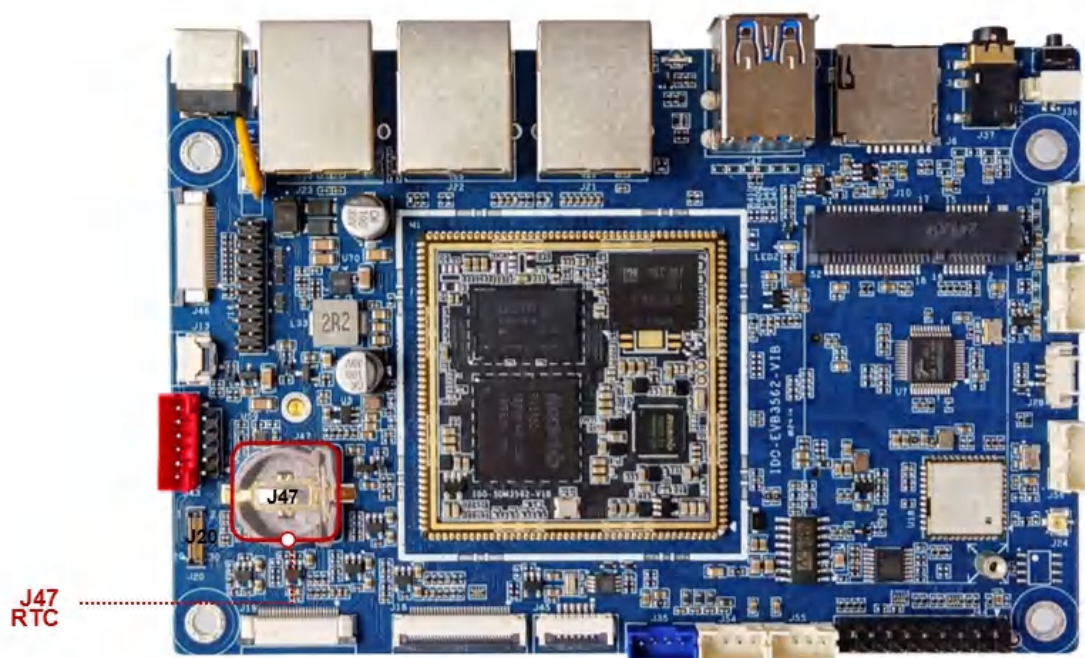
主板配置了1个LED灯，位于USB与网口在中间，如下图所示：



序号	名称	位置	颜色	功能
1	系统指示灯	下	黄色	闪烁表示系统运行正常

12 RTC

RTC如下图所示：



主板配置了一个外部RTC，型号为HYM8563，系统中对应的设备节点为/dev/rtc0。

12.1 读取RTC时间

读取RTC时间，命令如下：

```

1 root@rk3562-buildroot:/# hwclock
2 Fri Aug  4 09:02:38 2017  0.000000 seconds

```

12.2 设置RTC时间

设置RTC时间，命令如下：

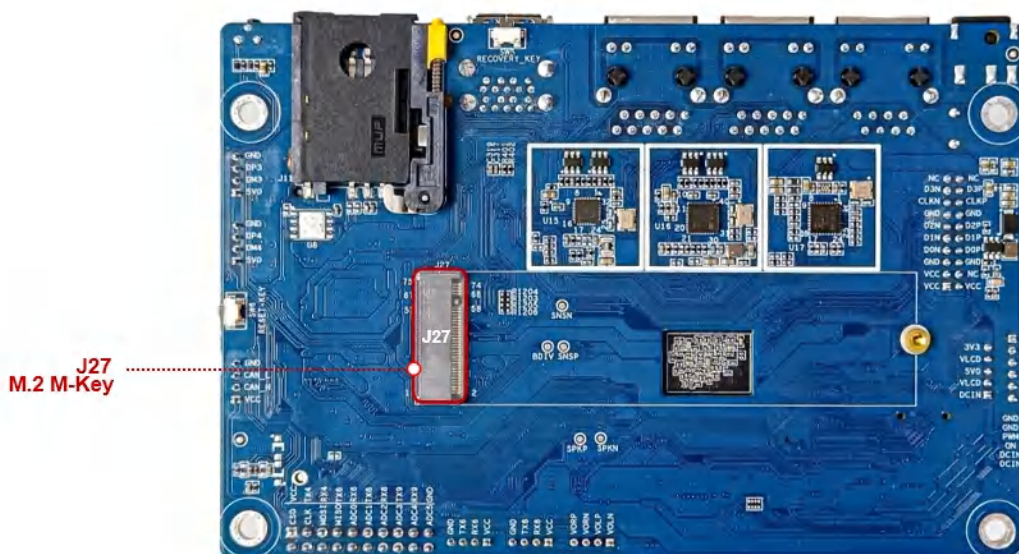
```

1 root@rk3562-buildroot:/# date -s '2023-8-22 15:30:00'
2 Tue Aug 22 15:30:00 UTC 2023
3 root@rk3562-buildroot:/# hwclock -w
4 root@rk3562-buildroot:/# hwclock
5 Tue Aug 22 15:30:05 2023  0.000000 seconds

```

13 硬盘

主板配置了一路硬盘接口（M.2），位于主板背面J27，如下图所示：



使用fdisk命令可以查看硬盘设备信息，命令如下：

▼Shell

```
1 root@rk3562-buildroot:/# fdisk -l
2 Disk /dev/nvme0n1: 932 GB, 1000204886016 bytes, 1953525168 sectors
3 121601 cylinders, 255 heads, 63 sectors/track
4 Units: sectors of 1 * 512 = 512 bytes
5
6 Device      Boot StartCHS      EndCHS          StartLBA      EndLBA      Sectors
   Size Id Type
7 /dev/nvme0n1p1  1,0,1      1023,254,63      2048 1953525134 1953523087
   931G  c Win95 FAT32 (LBA)
```

使用mount命令挂载到文件系统中，命令如下：

▼Shell

```
1 root@rk3562-buildroot:/# mount /dev/nvme0n1p1 /mnt
2 root@rk3562-buildroot:/# ls /mnt
3 4GB.bin  Audiobooks  Documents  LOST.DIR  Music      Pictures  Recordin
  gs
4 Alarms   DCIM        Download   Movies     Notifications  Podcasts  Ringtone
  s
```

挂载成功后，即可进行文件读写操作。

14 扩展接口

主板配置了一排扩展接口，位于J44，扩展接口的各个引脚功能，如下表所示：

pin	功能	pin	功能
1	5V输出	2	SPI0_CS
3	UART2_TX	4	SPI0_CLK
5	UART2_RX	6	SPI0_MOSI
7	UART6_TX	8	SPI0_MISO
9	UART6_RX	10	ADC0
11	UART8_TX	12	ADC1
13	UART8_RX	14	ADC2
15	UART9_TX	16	ADC3

17	UART9_RX	18	ADC4
19	GND	20	ADC5

14.1 UART

扩展接口共包含4路UART，各路UART对应的设备节点，如下表所示：

pin	功能	设备节点
3	UART2_TX	/dev/ttyS2
5	UART2_RX	
7	UART6_TX	/dev/ttyS6
9	UART6_RX	
11	UART8_TX	/dev/ttyS8
13	UART8_RX	
15	UART9_TX	/dev/ttyS9
17	UART9_RX	

14.2 ADC输入

扩展接口配置了6路ADC输入（参考电压为1.8V，精度10位）如下表所示：

名称	设备节点
ADCIN0	/sys/bus/iio/devices/iio:device0/in_voltage0_raw
ADCIN1	/sys/bus/iio/devices/iio:device0/in_voltage1_raw
ADCIN2	/sys/bus/iio/devices/iio:device0/in_voltage2_raw
ADCIN3	/sys/bus/iio/devices/iio:device0/in_voltage3_raw
ADCIN4	/sys/bus/iio/devices/iio:device0/in_voltage4_raw
ADCIN5	/sys/bus/iio/devices/iio:device0/in_voltage5_raw

ADC输入电压计算公式为： $V = (\text{voltage_raw}/1024) \times 1.8V$ （其中voltage_raw为从设备节点读取的值）。

以ADCIN2为例：

1. 读取in_voltage2_raw的值，命令如下：

```
▼ Shell |
1 root@rk3562-buildroot:/# cat /sys/bus/iio/devices/iio:device0/in_voltage2_r
  aw
2 524
```

2. 计算ADCIN2的输入电压V

$$V = (\text{voltage_raw}/1024) \times 1.8V = (524/1024) \times 1.8v = 0.92V$$

14.4 SPI

扩展接口配置了1路SPI，系统中对应的设备节点为/dev/spidev0.0。

15 RealTime Linux实时性系统

15.1 开启压力测试

```
▼ Bash |
1 // 开启性能模式
2 $ echo performance | tee $(find /sys/ -name *governor) /dev/null || true
3 // 开启压力测试
4 $ stress --cpu 4 --io 4 --vm 2 --vm-bytes 128M --timeout 3600s &
```

15.2 使用cyclicttest测试

```

1  $ cyclicttest -t 4 -p 80 -m -S -D 1h
2  #参数详解:
3  -t:指定测试线程的数量
4  -p:设置线程的优先级 (优先级范围通常在1到99之间 (99 为最高优先级) )
5  -m:锁定内存
6  -S:使用 SCHED_FIFO 调度策略(SCHED_FIFO 是一种先入先出的实时调度策略, 它确保高优先级线程在运行时不会被较低优先级的线程打断, 直到它主动放弃 CPU。)
7  -D:指定测试时间 (m:分钟; h:小时; d:天)

```

```

root@rk3562-buildroot:~# cyclicttest -t 5 -p 80 -m -S -D 1h
# /dev/cpu_dma_latency set to 0us
policy: fifo: loadavg: 9.87 12.24 12.42 1/211 1288

T: 0 ( 1234) P:80 I:1000 C:3599993 Min:      4 Act:      4 Avg:      7 Max:      47
T: 1 ( 1235) P:80 I:1500 C:2399976 Min:      4 Act:      5 Avg:      7 Max:      29
T: 2 ( 1236) P:80 I:2000 C:1799981 Min:      4 Act:      4 Avg:      7 Max:     186
T: 3 ( 1237) P:80 I:2500 C:1439982 Min:      4 Act:      5 Avg:      7 Max:      35

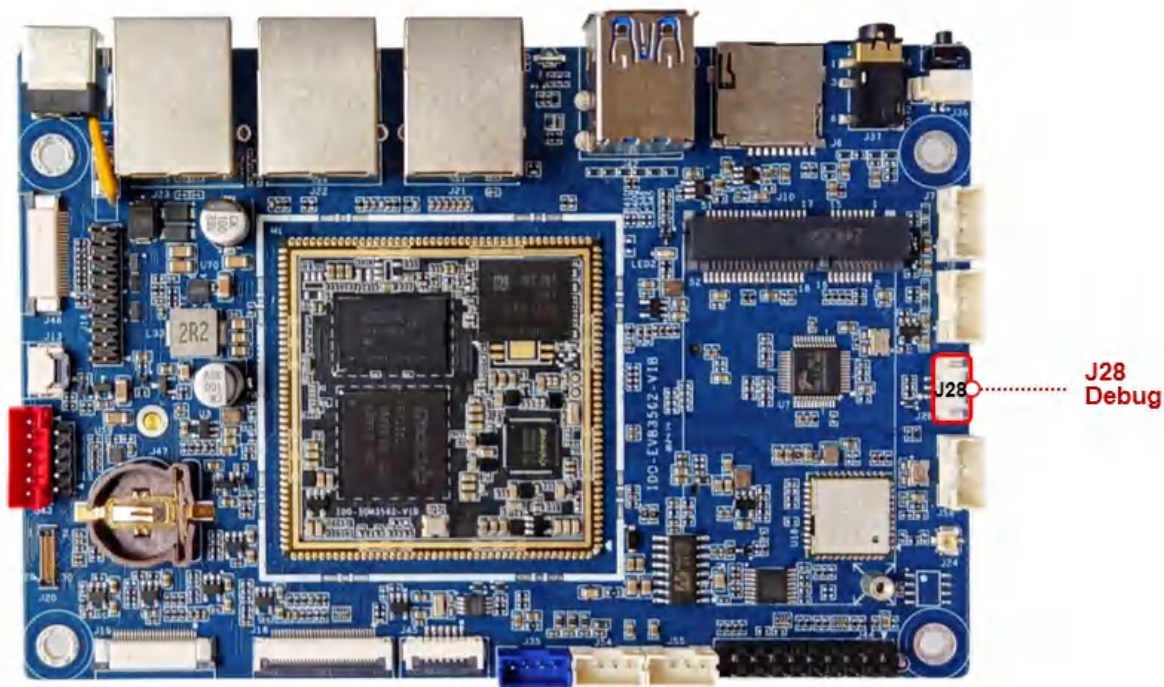
```

IDO-EVB3562-Ubuntu系统

1 调试

1.1 串口调试

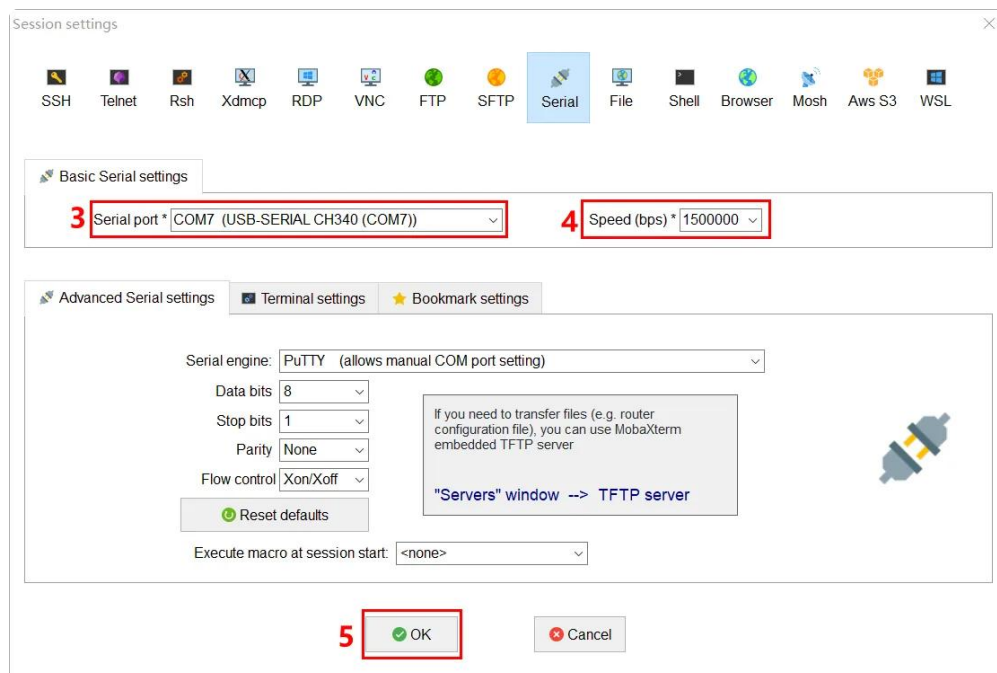
主板调试串口位于USB座子下方J28, 建议使用配套的usb转串口工具, 如下图所示:



1. 打开MobaXterm, 下载链接如下:

链接: https://pan.baidu.com/s/11ui4LTd2mq_9kiJpeL4bWg?pwd=1234

提取码: 1234



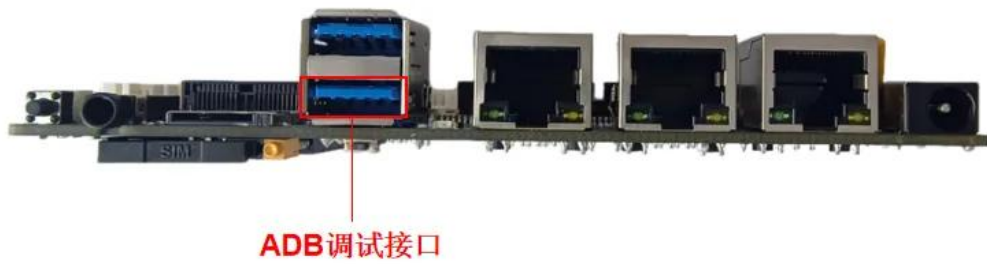
结果如下图所示:

```
root@ido:~# ls /
bin  dev  lib      mnt  root  sdcard  system  usr
boot  etc  lost+found  opt  run  srv     tmp     var
data  home media    proc /sbin  sys     udisk   vendor
root@ido:~#
```

测试项目	要求	测试结果
调试串口	升级后 调试串口能收发	✓

1.2 adb调试

主板adb接口位于下方的USB口，与烧录接口为同一接口，如下图所示：



adb默认使用root账户登录，如下图所示：

```
root@ido:/# ls
bin  dev  lib      mnt  root  sdcard  system  usr
boot etc  lost+found  opt  run  srv     tmp     var
data home media    proc sbin sys     udisk   vendor
root@ido:/#
```

测试项目	要求	测试结果
adb串口	升级完成后adb串口正常	✓

1.3 ssh调试

主板支持ssh调试，默认登录账号密码为：

- 1. 账号：ido
- 2. 密码：123456
- 3. 端口：22

ssh调试，如下图所示：

```
login as: ido
ido@192.168.1.102's password:

Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.10.160 aarch64)

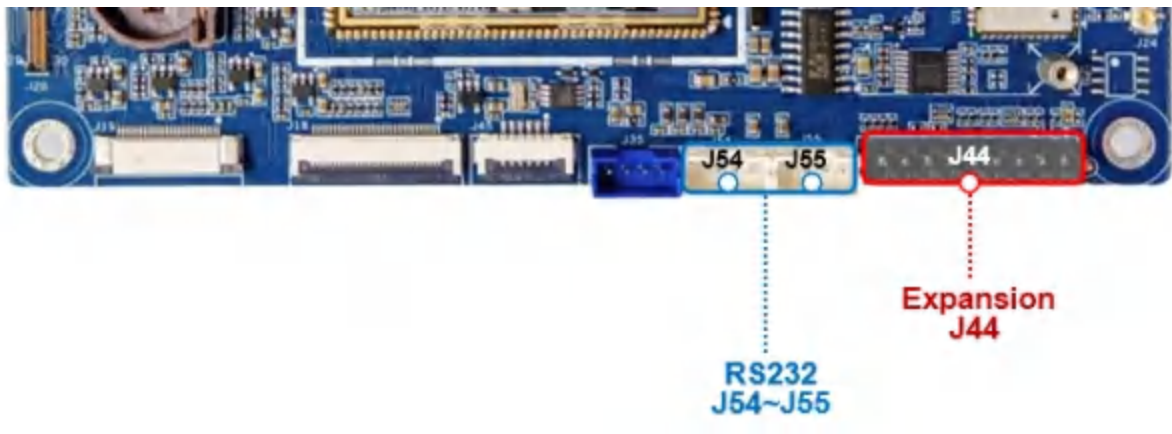
System information as of Wed Jul 31 05:42:20 UTC 2024

System load:  0.01 0.03 0.05   Up time:         2:36 hours           Local us
ers:    2
Memory usage: 22 % of 1970MB   IP:           192.168.1.102 192.168.0.186
Usage of /:   32% of 12G

ido@ido:~$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  Templates  Videos
ido@ido:~$
```

2 串口

主板共配置3路串口（除调试串口），其中2路RS232，1路TTL，如下图所示：



序号	接口位置	电平类型	串口设备节点	备注
1	J44	TTL	/dev/ttyS4	/
2	J55	RS232	/dev/ttyS6	连接到扩展端口J44为TTL
3	J54	RS232	/dev/ttyS8	连接到扩展端口J44为TTL

使用microcom可以进行串口收发测试，命令如下：

```

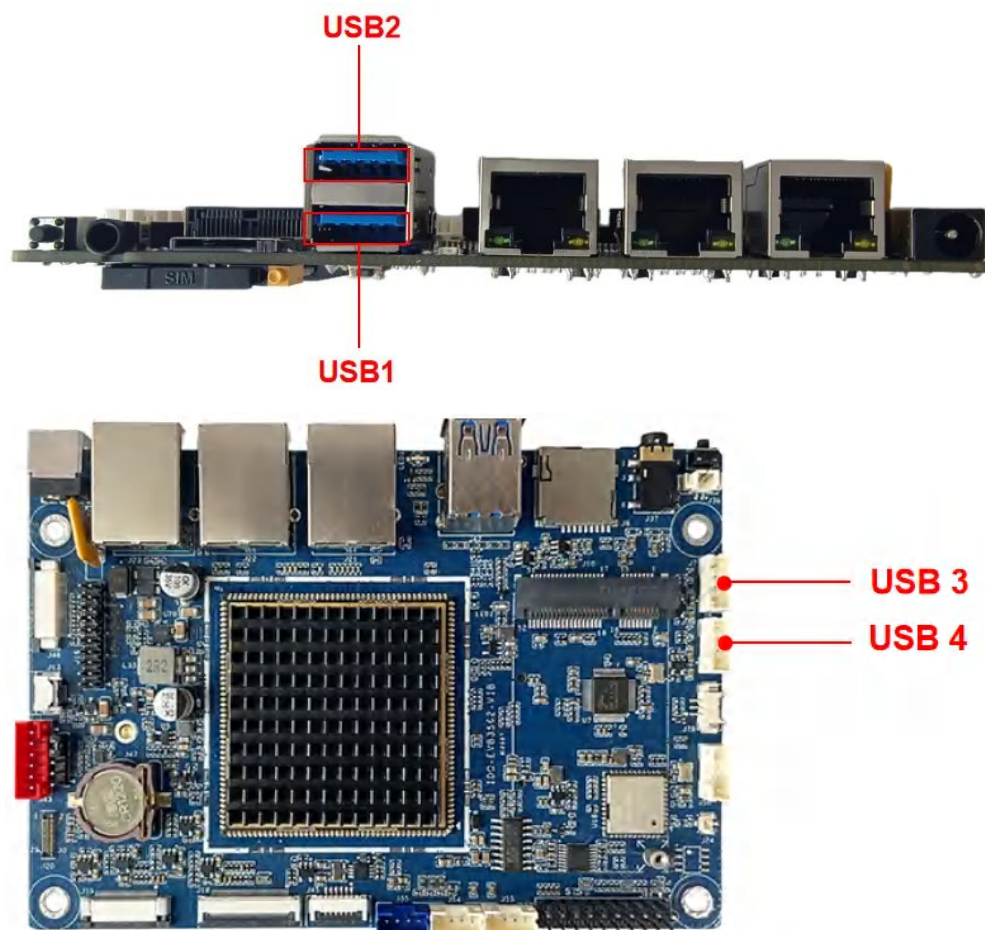
1 root@ido:~# apt-get update && apt-get install microcom
2 root@ido:~# microcom -s 115200 -p /dev/ttyS6
3 [ 9563.763274] of_dma_request_slave_channel: dma-names property of node '/serial@ff6c0000' missing or empty
4 connected to /dev/ttyS6
5 Escape character: Ctrl-\
6 Type the escape character to get to the prompt.

```

注意：测试完成，按Ctrl+\和Ctrl+C退出。

3 USB

主板共配置4路USB接口，分别记作USB1-4，如下图所示：



序号	接口位置	类型
USB1	J24下	OTG
USB2	J24上	USB2.0 HOST

USB3	J7	USB2.0 HOST
USB4	J8	USB2.0 HOST

其中USB1接电脑上电时，默认为device模式（可使用adb调试）；接U盘或者其他存储设备上电时，默认为host模式（可读取U盘）。

USB端口插上U盘，会默认挂载到/media/ido/XXX目录下：

▼

Bash

```
1 root@ido:/# mount
2 ...
3 /dev/sda1 on /media/ido/KINGSTON type vfat ...
```

测试项目	要求	测试结果
USB3.0	USB1	✓
	USB2	✓
	USB3	✓
	USB3	✓

4 Micro SD

接上SD卡后，自动挂载到/media/ido/xxx目录下，可以使用mount命令确认：

▼

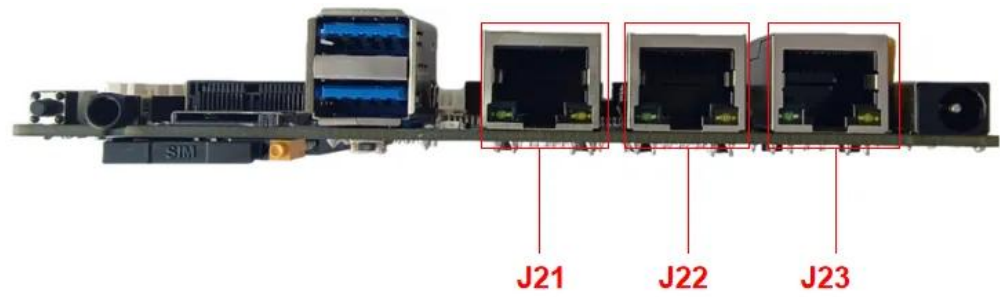
Shell

```
1 root@ido:/# mount
2 ...
3 /dev/mmcblk1p1 on /media/ido/0403-0201 type vfat
```

测试项目	要求	测试结果
挂载SD	可以自动挂载fat32格式sd卡	✓
	可以手动挂载NTFS的sd卡	✓

5 Ethernet

主板共配置3路以太网接口，包括2路千兆网和1路百兆网，如下图所示：



序号	接口位置	速率	网络节点
1	J21	千兆	enp1s0
2	J22	千兆	eth0
3	J23	百兆	eth2

系统默认开启DHCP服务，动态获取IP。可通过ifconfig指令设置临时静态IP，命令如下：

▼

Shell

```
1 root@ido:~# ifconfig eth0 192.168.1.123
```

如需设置永久静态IP，则可通过修改/etc/network/interfaces，命令如下：

▼Shell

```
1  # interface file auto-generated by buildroot
2
3  auto lo
4  iface lo inet loopback
5
6  auto eth0
7      iface eth0 inet static
8      address 192.168.0.123
9      netmask 255.255.255.0
10     gateway 192.168.0.1
11     nameserver 192.168.0.1
12
13 auto eth1
14     iface eth1 inet static
15     address 192.168.1.123
16     netmask 255.255.255.0
17     gateway 192.168.1.1
18     nameserver 192.168.1.1
19
```

主板重启，命令如下：

▼Shell

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

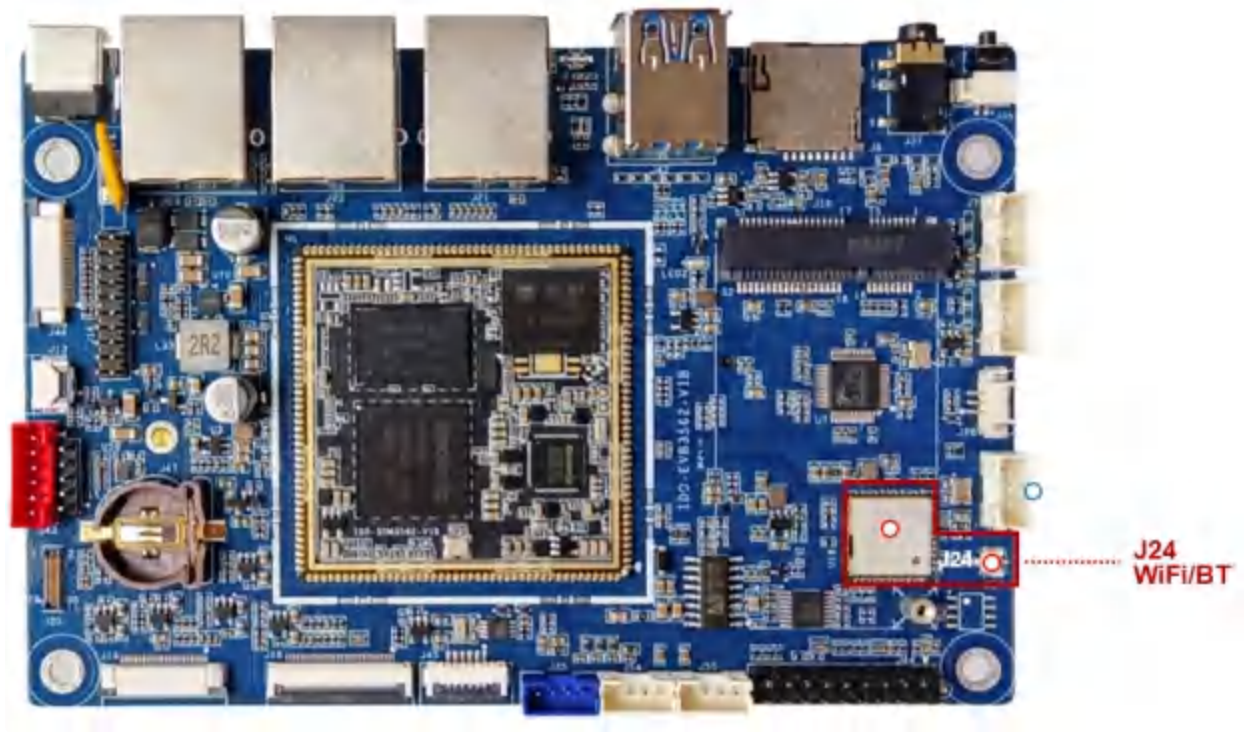
即使主板断电重启，此静态IP设置仍然生效。

测试项目	要求	测试结果
enp1s0	识别正常，并成功上网	✓
	自动获取IP地址	✓
	手动修改IP地址	✓
eth0	识别正常，并成功上网	✓
	自动获取IP地址	✓
	手动修改IP地址	✓
	识别正常，并成功上网	✓

eth2	自动获取IP地址	✓
	手动修改IP地址	✓

6 WiFi

主板配置一个2.4G/5G 双频WiFi模块，型号是AP6256，如下图所示：



系统启动会默认打开WiFi，对应的网络节点为wlan0，命令如下：

▼ Shell |

```
1 root@ido:~# ifconfig wlan0
2 wlan0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
3     ether 50:41:1c:5d:02:70 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
```

6.1 连接热点

6.1.1 命令行连接

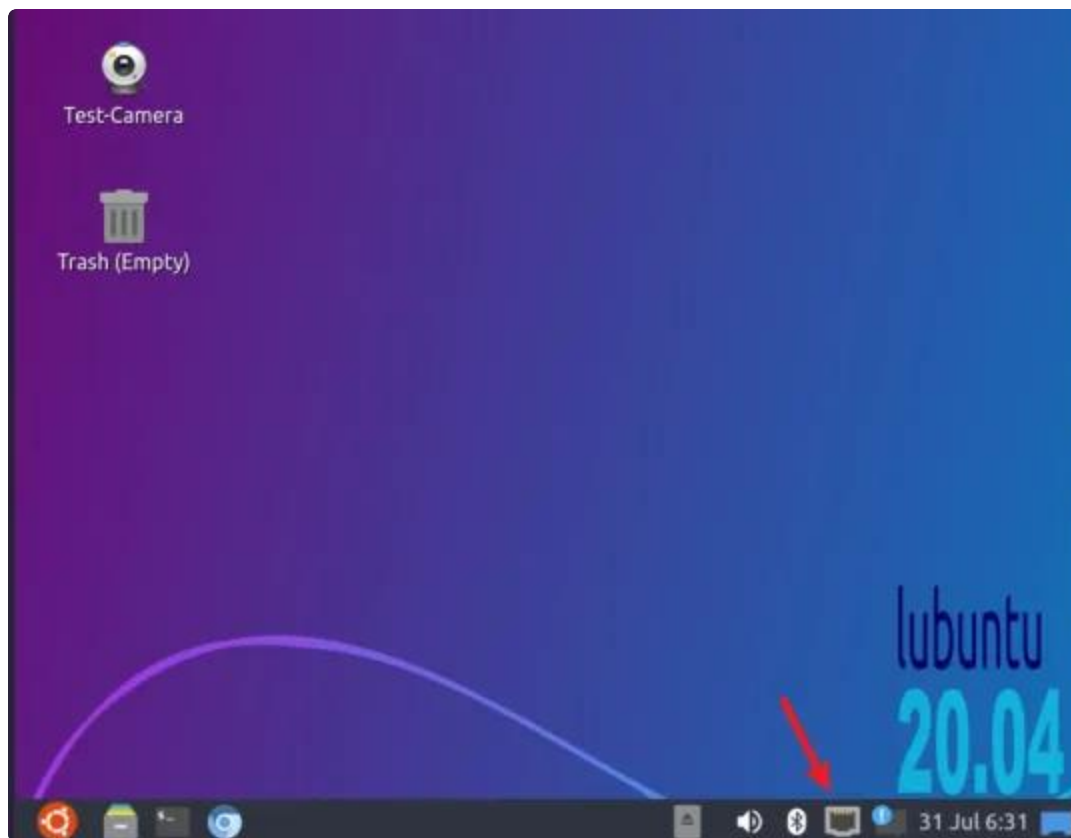
使用wpa_cli工具连接，命令如下：

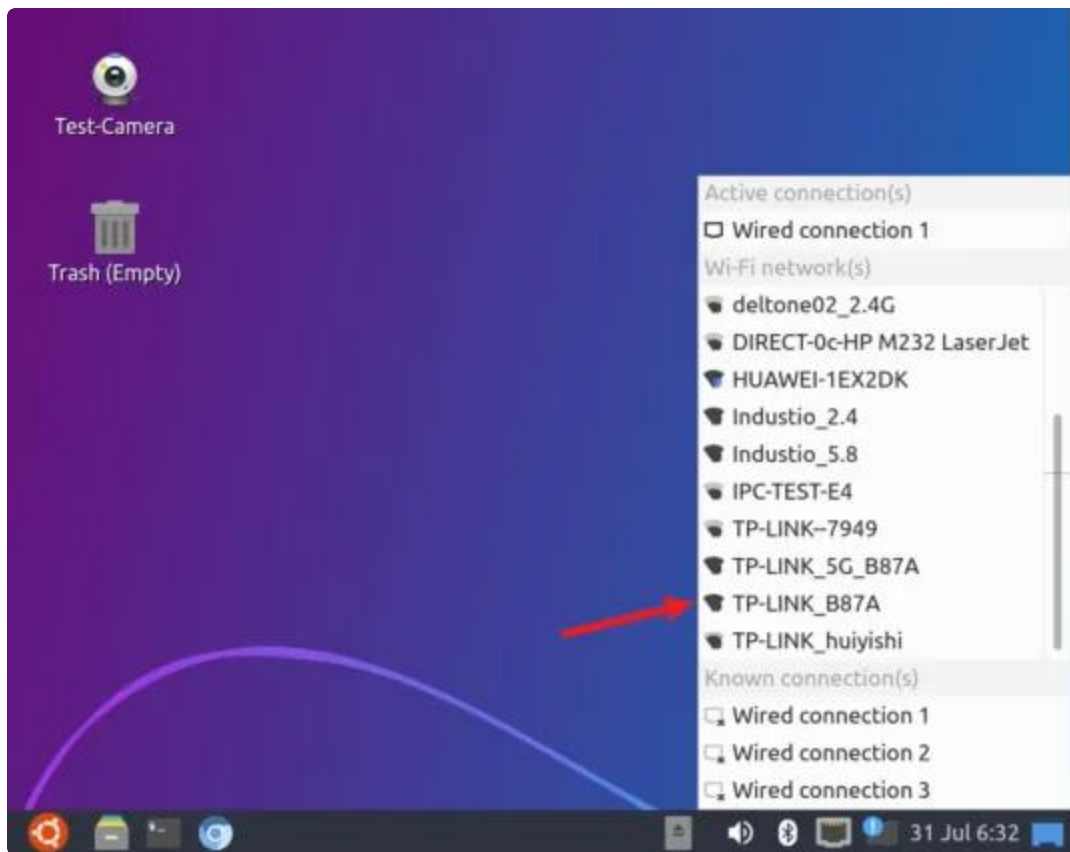
```
1 root@ido:~# wpa_cli -i wlan0 add_network
2 root@ido:~# wpa_cli -i wlan0 set_network 0 ssid '"TP-LINK_B87A"'
3 root@ido:~# wpa_cli -i wlan0 set_network 0 psk '"12345678"'
4 root@ido:~# wpa_cli -i wlan0 enable_network 0
5 root@ido:~# dhclient -i wlan0
```

其中"TP-LINK_B87A"为热点名称，"12345678"为热点密码。

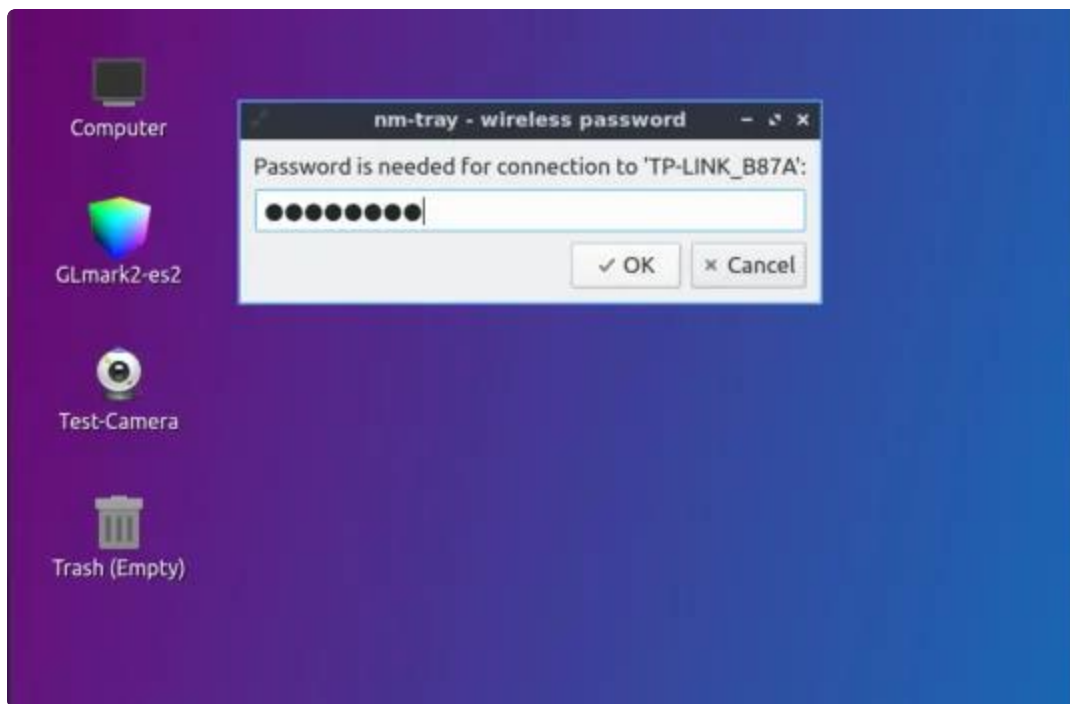
6.1.2 桌面连接

点击桌面右上角的网络按钮，弹出的列表中选择要连接的热点，如下图所示：

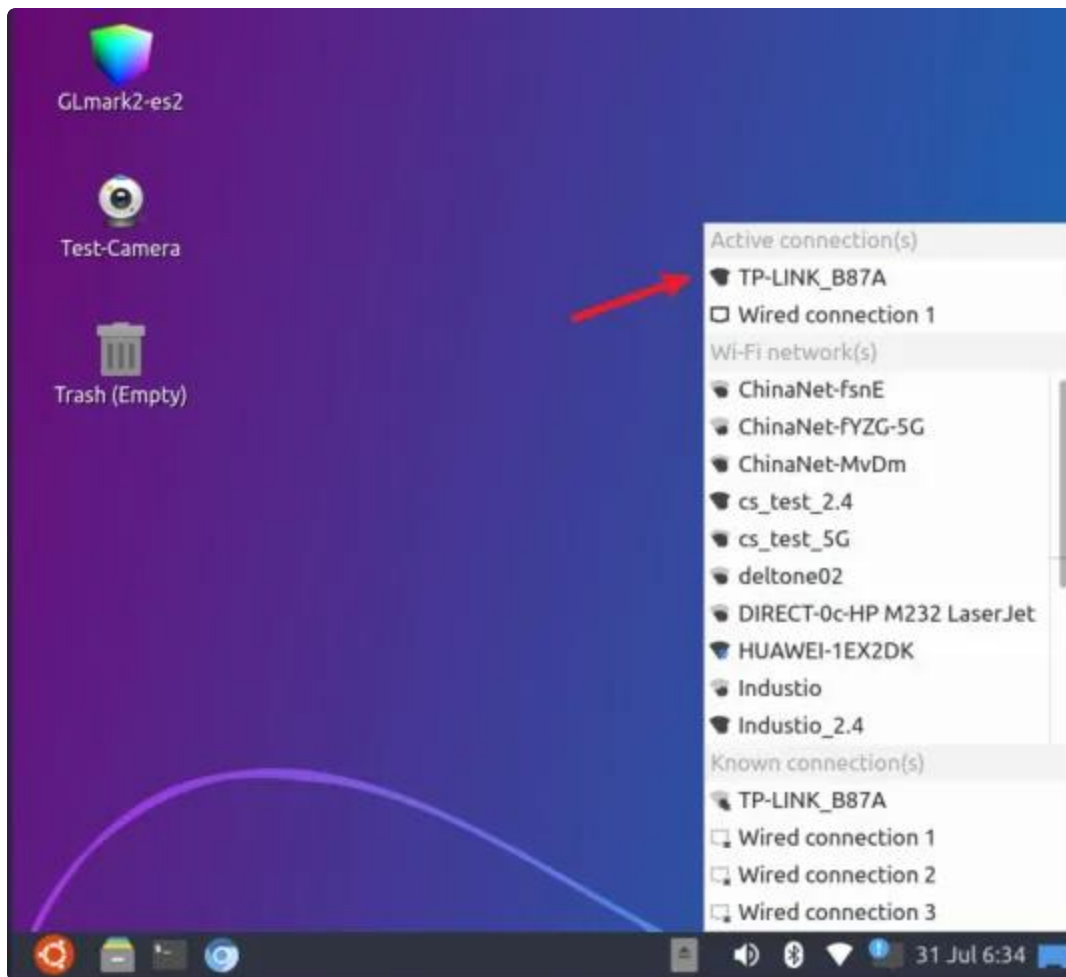




弹出密码输入框，使用键盘输入密码（如果没有接键盘，可以使用软键盘Onboard），如下图所示：



输入密码完成后，点击【OK】按钮，即可连接WiFi热点。



```
Shell |
1 root@ido:~# ifconfig wlan0
2 wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
3     inet 192.168.1.127 netmask 255.255.255.0 broadcast 192.168.1.255
4     inet6 fe80::5ac:29b:8a95:2bf0 prefixlen 64 scopeid 0x20<link>
5     ether 50:41:1c:b0:13:e2 txqueuelen 1000 (Ethernet)
6     RX packets 1399 bytes 824772 (824.7 KB)
7     RX errors 0 dropped 0 overruns 0 frame 0
8     TX packets 977 bytes 128063 (128.0 KB)
9     TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

7 Bluetooth

主板蓝牙版本为BT5.0。

7.1 开启蓝牙

开启蓝牙，命令如下：

▼Shell

```
1 root@ido:~# hciconfig hci0 up
2 root@ido:~# hciconfig
3 hci0:   Type: Primary  Bus: UART
4         BD Address: 50:41:1C:5D:02:87  ACL MTU: 1021:8  SCO MTU: 64:1
5         UP RUNNING
6         RX bytes:871 acl:0 sco:0 events:61 errors:0
7         TX bytes:3282 acl:0 sco:0 commands:61 errors:0
```

测试项目	要求	测试结果
蓝牙	蓝牙开启	✓
	蓝牙关闭	✓

7.2 扫描设备

扫描设备，命令如下：

▼Shell

```
1 root@ido:~# hciconfig hci0 iscan
2 root@ido:~# hcitool scan
3 Scanning ...
```

测试项目	要求	测试结果
蓝牙	蓝牙扫描	✓

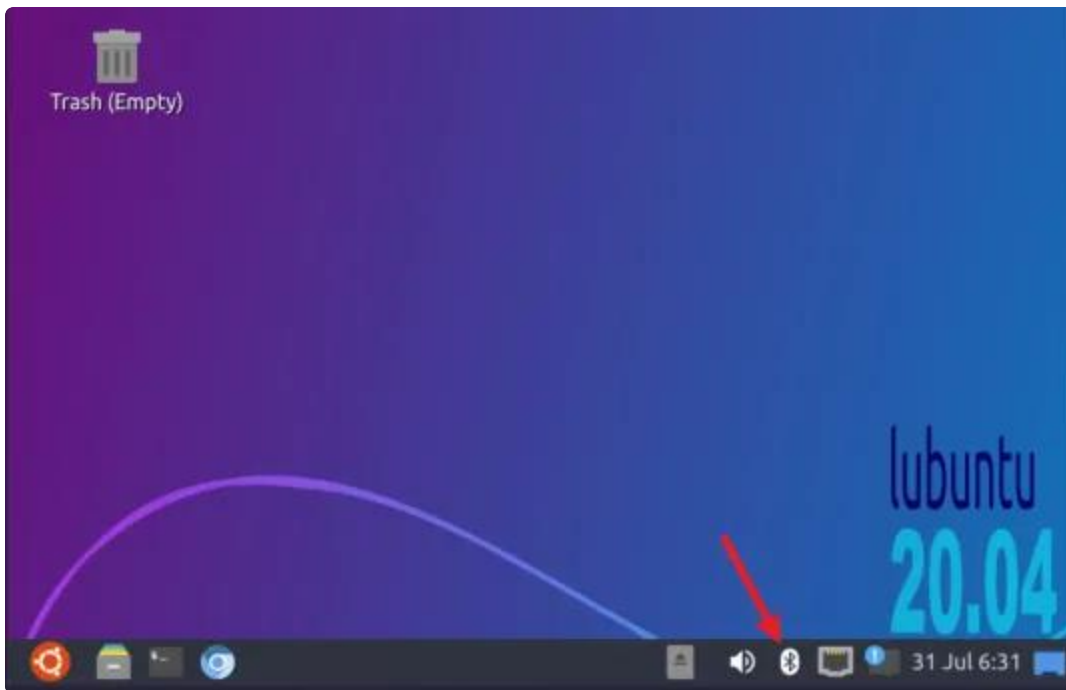
7.3 连接蓝牙设备

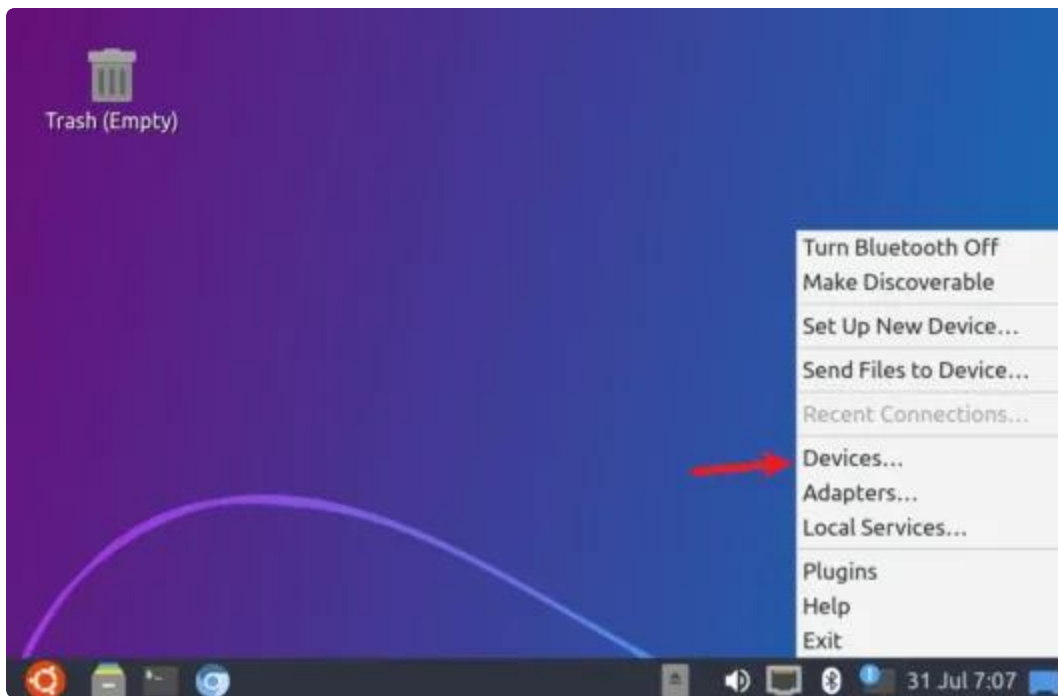
使用命令行连接蓝牙设备，命令如下：

```
▼ Shell |
1 root@ido:~# bluetoothctl
2 [bluetooth]# scan on
3 ...
4 [bluetooth]# trust 7C:C1:80:09:DD:6C
5 [bluetooth]# pair 7C:C1:80:09:DD:6C
6 [cainiaocl]# exit
```

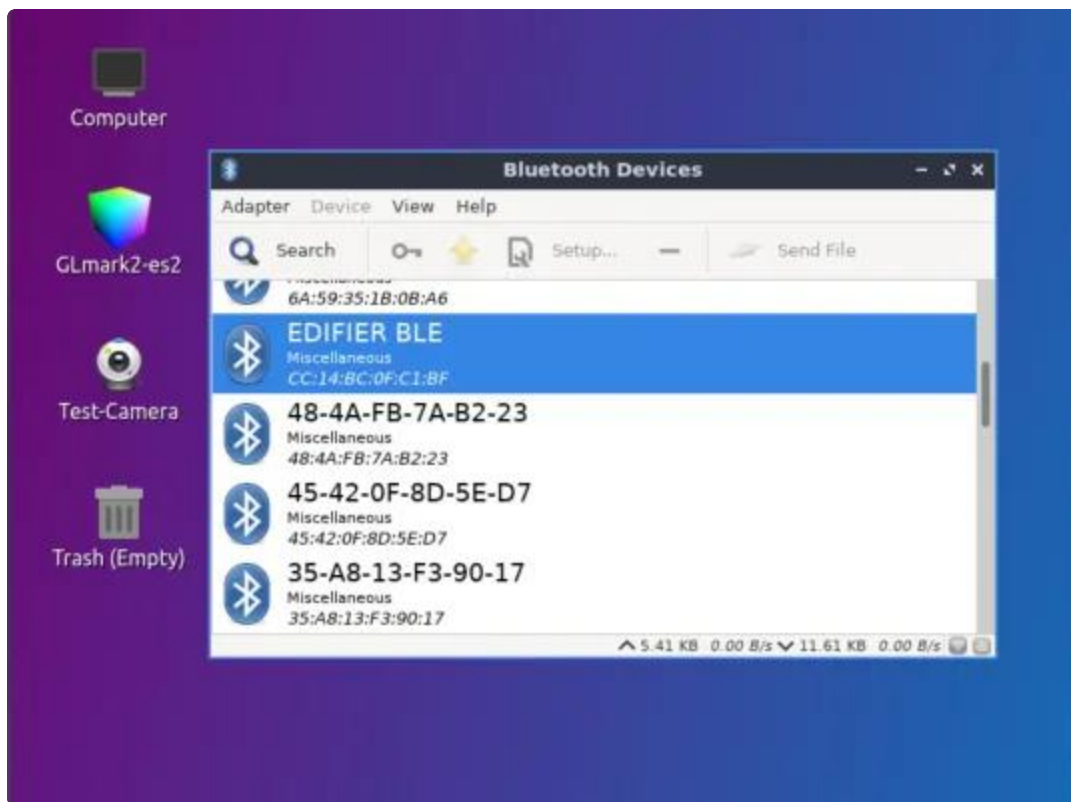
也可在桌面上连接蓝牙设备，步骤如下：

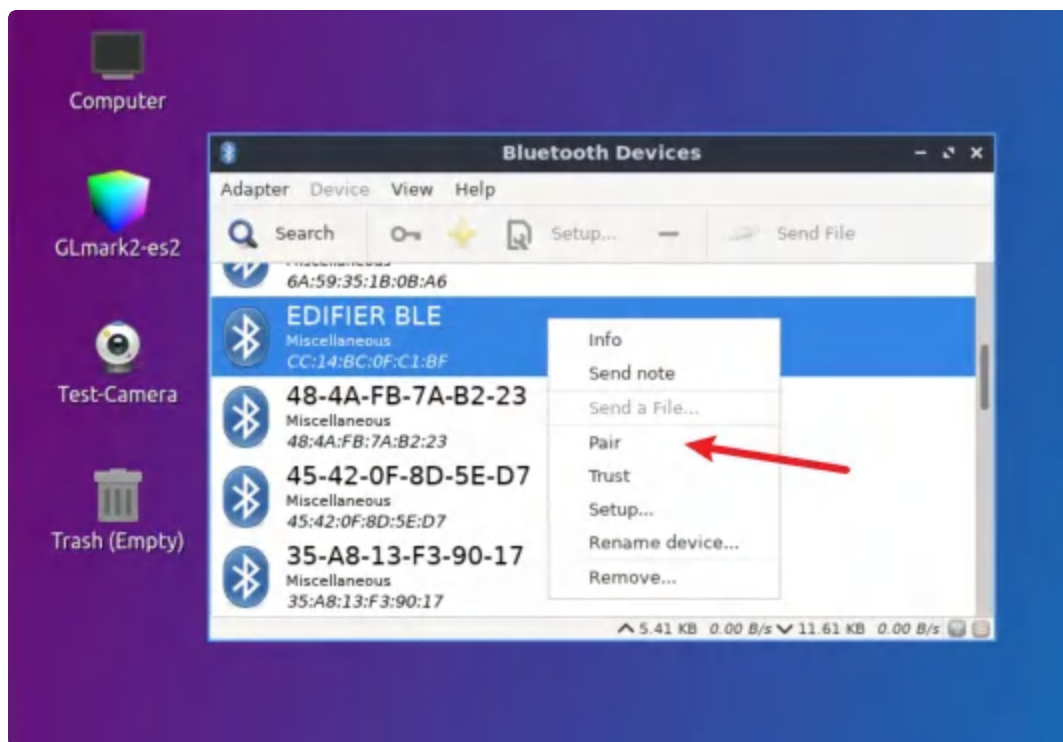
在桌面右下角蓝牙图标右键->Devices：





选中要连接的蓝牙设备，右键->pair:

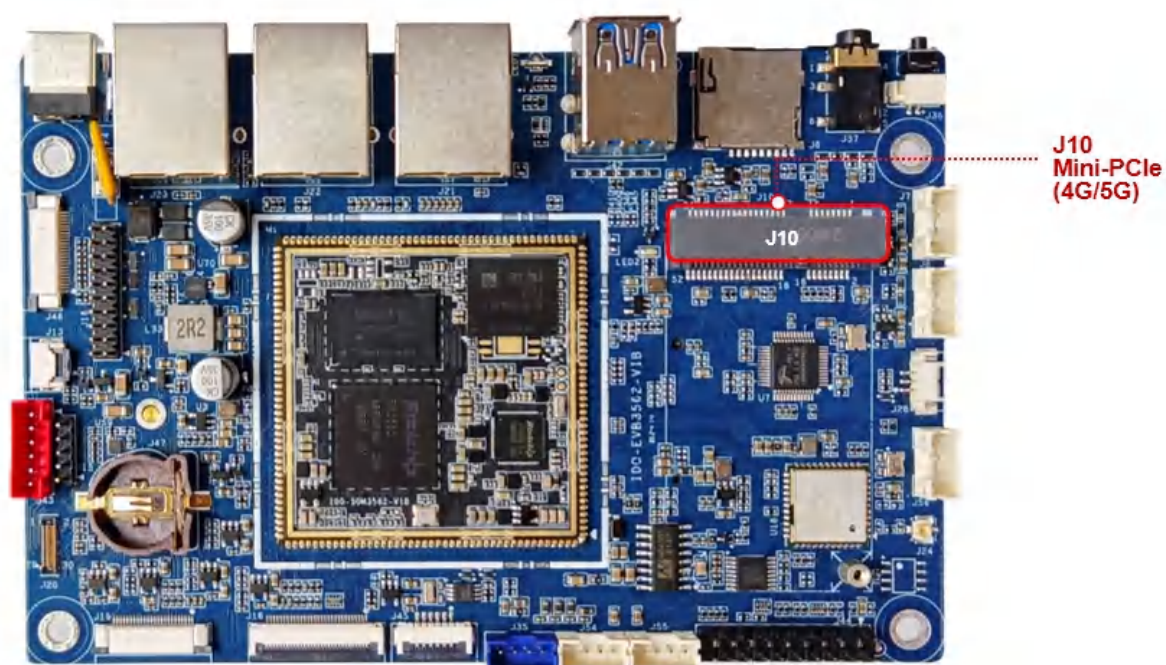




测试项目	要求	测试结果
蓝牙	蓝牙 连接	☐

8 4G

主板配置了一路4G接口，默认适配EC20模块，如下图所示：



正确按照4G模块、SIM卡和4G天线后，使用quectel-CM工具拨号，命令如下：

▼ Shell

```
1 root@ido:~# quectel-CM &
```

拨号成功，将产生wwan0网络节点，命令如下：

▼ Shell

```
1 root@ido:~# ifconfig wwan0
2 wwan0      Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
3            inet addr:10.122.45.92  P-t-P:10.122.45.92  Mask:255.255.255.248
4            UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
5            RX packets:12 errors:0 dropped:0 overruns:0 frame:0
6            TX packets:59 errors:0 dropped:0 overruns:0 carrier:0
7            collisions:0 txqueuelen:1000
8            RX bytes:1396 (1.3 KiB)  TX bytes:8214 (8.0 KiB)
```

测试4G上网功能，命令如下：

▼ Shell

```
1 root@ido:~# ping www.baidu.com -I wwan0
2 PING www.a.shifen.com (14.119.104.189) from 10.122.45.92 wwan0: 56(84) byte
  s of data.
3 64 bytes from 14.119.104.189 (14.119.104.189): icmp_seq=1 ttl=52 time=95.5
  ms
4 64 bytes from 14.119.104.189 (14.119.104.189): icmp_seq=2 ttl=52 time=99.3
  ms
5 64 bytes from 14.119.104.189 (14.119.104.189): icmp_seq=3 ttl=52 time=97.8
  ms
6
```

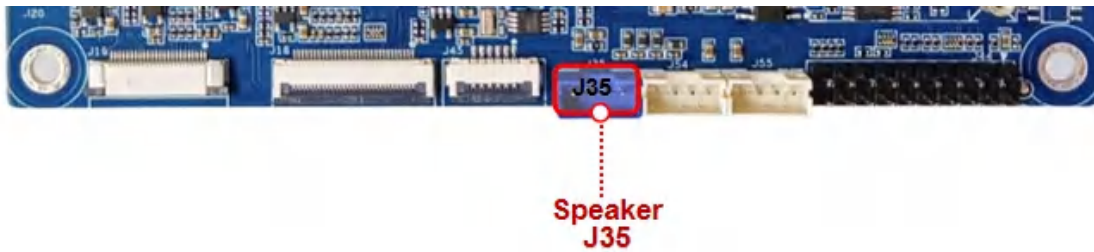
测试项目	要求	测试结果
4G	4G模块识别	☐
	4G上网	☐

9 音频

主板配置了一路声卡Rockchip RK809。

9.1 喇叭

双声道Lineout接口位于J35，如下图所示：



打开/关闭Lineout开关，命令如下：

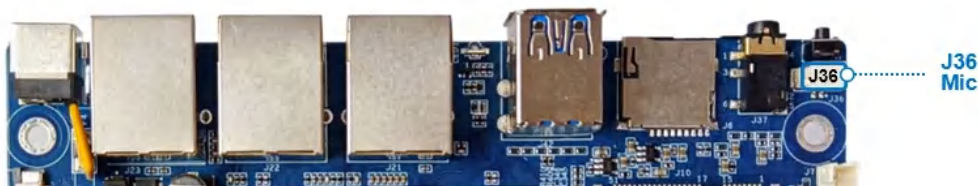
```
▼ Shell |
1 //关闭Lineout
2 root@ido:~# amixer cset numid=45,iface=MIXER,name='Playback Path' 'OFF'
3
4 //打开Lineout
5 root@ido:~# amixer cset numid=45,iface=MIXER,name='Playback Path' 'SPK'
```

播放音频，命令如下：

```
▼ Shell |
1 root@ido:~# aplay /usr/share/sounds/alsa/Front_Center.wav
```

9.2 MIC

MIC如下图所示：



打开MIC开关，命令如下：

```
▼ Shell |
1 amixer cset numid=46,iface=MIXER,name='Capture MIC Path' 'Main Mic'
```

录音，命令如下：

▼Shell

```
1 arecord -D hw:0,0 -r 48000 -c 2 -f S16_LE test.wav
```

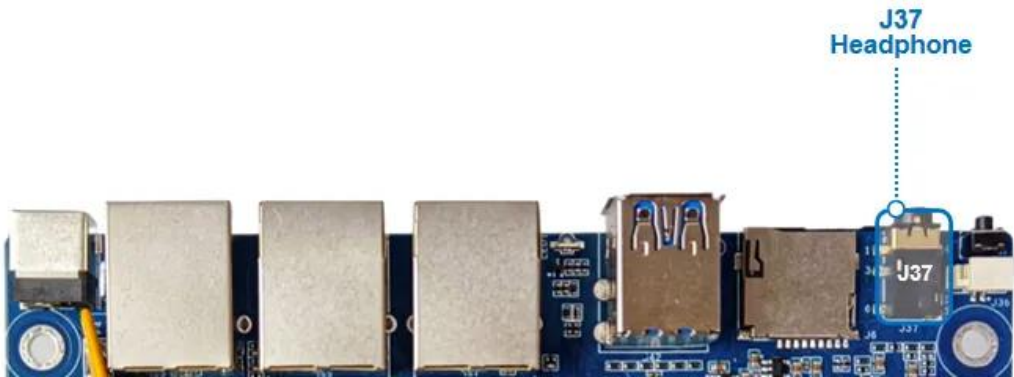
播放，命令如下：

▼Shell

```
1 aplay -Dplughw:0,0 ./test.wav
```

9.3 耳机

耳机如下图所示：



打开/关闭耳机音频输出开关，命令如下：

▼Shell

```
1 //关闭耳机输出
2 root@ido:~# amixer cset numid=45,iface=MIXER,name='Playback Path' 'OFF'
3
4 //打开耳机输出
5 root@ido:~# amixer cset numid=45,iface=MIXER,name='Playback Path' 'SPK'
```

10 Camera

主板配置了2路摄像头，一路为GC8034（J18，3264x2448），另一路为OV13855（J20，4224x3136），两路摄像头对应的设备节点如下：

GC8034	/dev/video22
OV13855	/dev/video29

10.1 检测摄像头

▼

Bash |

```
1 root@ido:~# media-ctl -p -d /dev/media0 | grep gc8034
2         <- "m01_f_gc8034 2-0037":0 [ENABLED]
3 - entity 63: m01_f_gc8034 2-0037 (1 pad, 1 link)
4 root@ido:~# media-ctl -p -d /dev/media1 | grep ov13855
5         <- "m00_b_ov13855 1-0010":0 [ENABLED]
6 - entity 63: m00_b_ov13855 1-0010 (1 pad, 1 link)
```

10.2 预览摄像头

预览GC8034摄像头画面：

▼

Bash |

```
1 root@ido:~# gst-launch-1.0 v4l2src device=/dev/video22 ! video/x-raw, format=NV12, width=1920, height=1080, framerate=30/1 ! autovideoconvert ! autovideodeosink
```

预览OV13855摄像头画面：

▼

Bash |

```
1 root@ido:~# gst-launch-1.0 v4l2src device=/dev/video29 ! video/x-raw, format=NV12, width=1920, height=1080, framerate=30/1 ! autovideoconvert ! autovideodeosink
```

10.3 抓取视频

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/video22 --set-fmt-video=width=1920,
height=1080,pixelformat='NV12' --stream-mmap=4 --set-selection=target=cro
p,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          : sRGB
12    Transfer Function   : Rec. 709
13    YCbCr/HSV Encoding: Rec. 709
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 [ 2496.916217] rockchip-mipi-csi2 ff380000.mipi0-csi2: stream on, src_sd:
00000000949a962c, sd_name:rockchip-csi2-dphy0
30 [ 2496.916253] rockchip-mipi-csi2 ff380000.mipi0-csi2: stream ON
31     VIDIOC_STREAMON returned 0 (Success)
32 ...

```

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：

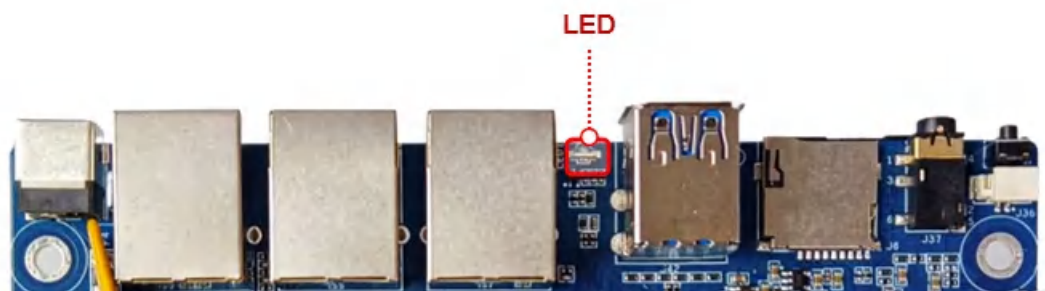
```

1 root@ido:~# ffplay -f rawvideo -video_size 1920x1080 -pix_fmt nv12 ./out.yu
v

```

11 LED

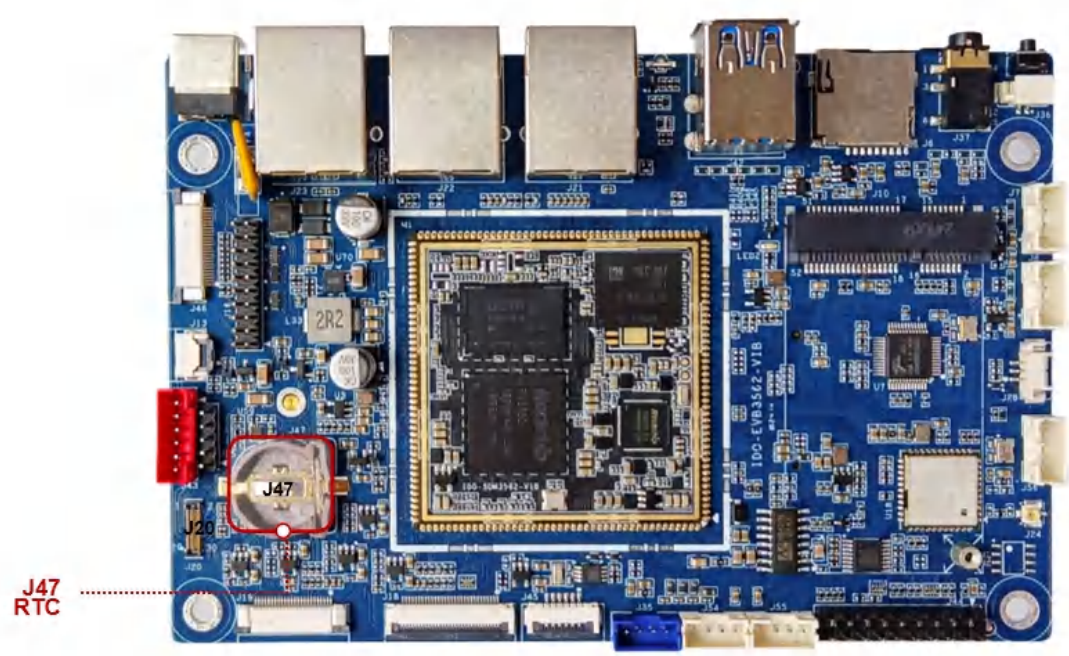
主板配置了1个LED灯，位于USB与网口在中间，如下图所示：



序号	名称	位置	颜色	功能
1	系统指示灯	下	黄色	闪烁表示系统运行正常

12 RTC

主板配置了一个外部RTC，型号为HYM8563，系统中对应的设备节点为/dev/rtc0，如下图所示：



12.1 读取RTC时间

读取RTC时间，命令如下：

▼Shell

```
1 root@ido:~# hwclock
2 Fri Aug 4 09:02:38 2017 0.000000 seconds
```

测试项目	要求	测试结果
RTC时间	读取时间	✓
	设备断电重启后时间保存	✓

12.2 设置RTC时间

设置RTC时间，命令如下：

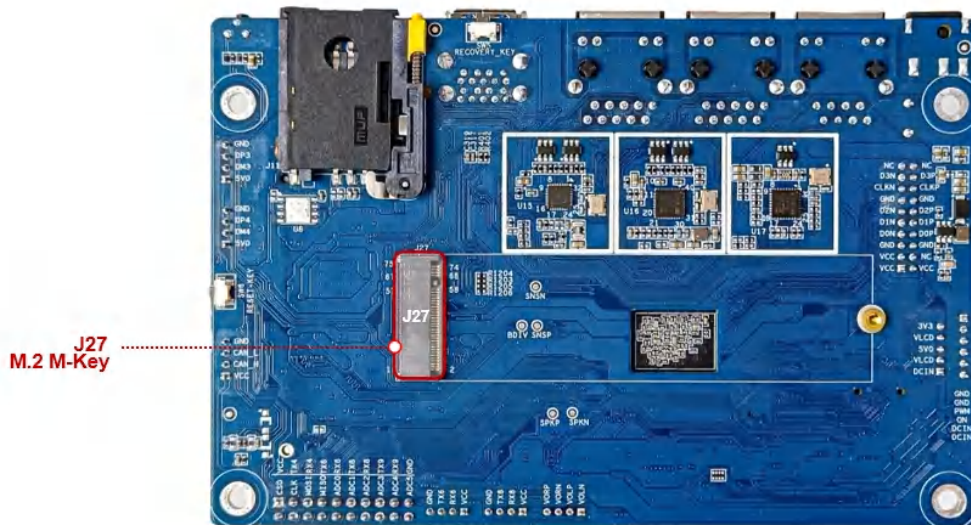
▼Shell

```
1 root@ido:~# date -s '2023-8-22 15:30:00'
2 Tue Aug 22 15:30:00 UTC 2023
3 root@ido:~# hwclock -w
4 root@ido:~# hwclock
5 Tue Aug 22 15:30:05 2023 0.000000 seconds
```

测试项目	要求	测试结果
RTC时间	设置RTC时间并保存成功	✓

13 硬盘

板配置了一路硬盘接口（M.2），位于主板背面J27，如下图所示：



使用fdisk命令可以查看硬盘设备信息，命令如下：

```

1 root@ido:~# fdisk -l
2 Disk /dev/nvme0n1: 932 GB, 1000204886016 bytes, 1953525168 sectors
3 121601 cylinders, 255 heads, 63 sectors/track
4 Units: sectors of 1 * 512 = 512 bytes
5 Device      Boot StartCHS      EndCHS          StartLBA      EndLBA      Sectors
   Size Id Type
6 /dev/nvme0n1p1  1,0,1      1023,254,63      2048 1953525134 1953523087
   931G  c Win95 FAT32 (LBA)
  
```

使用mount命令挂载到文件系统中，命令如下：

```

1 root@ido:~# mount /dev/nvme0n1p1 /udisk
  
```

挂载成功后，即可进行文件读写操作。

14 扩展接口

主板配置了一排扩展接口，位于J44，扩展接口的各个引脚功能，如下表所示：

pin	功能	pin	功能
1	5V输出	2	SPI0_CS
3	UART2_TX	4	SPI0_CLK

5	UART2_RX	6	SPI0_MOSI
7	UART6_TX	8	SPI0_MISO
9	UART6_RX	10	ADC0
11	UART8_TX	12	ADC1
13	UART8_RX	14	ADC2
15	UART9_TX	16	ADC3
17	UART9_RX	18	ADC4
19	GND	20	ADC5

14.1 UART

扩展接口共包含4路UART，各路UART对应的设备节点，如下表所示：

pin	功能	设备节点
j3-2	UART2_TX	/dev/ttyS2
5	UART2_RX	
7	UART6_TX	/dev/ttyS6
9	UART6_RX	
11	UART8_TX	/dev/ttyS8
13	UART8_RX	
15	UART9_TX	/dev/ttyS9
17	UART9_RX	

14.2 SPI

系统中对应的/dev/spidev0.0。

14.3 ADC输入

扩展接口配置了6路ADC输入（参考电压为1.8V，精度10位）如下表所示：

名称	设备节点
ADCIN0	/sys/bus/iio/devices/iio:device0/in_voltage0_raw
ADCIN1	/sys/bus/iio/devices/iio:device0/in_voltage1_raw
ADCIN2	/sys/bus/iio/devices/iio:device0/in_voltage2_raw
ADCIN3	/sys/bus/iio/devices/iio:device0/in_voltage3_raw
ADCIN4	/sys/bus/iio/devices/iio:device0/in_voltage4_raw
ADCIN5	/sys/bus/iio/devices/iio:device0/in_voltage5_raw

ADC输入电压计算公式为： $V = (\text{voltage_raw}/1024) \times 1.8V$ （其中voltage_raw为从设备节点读取的值）。

以ADCIN2为例：

读取in_voltage2_raw的值，命令如下：

▼ Shell

```
1 root@ido:~# cat /sys/bus/iio/devices/iio:device0/in_voltage2_raw
2 524
```

计算ADCIN2的输入电压V

$$V = (\text{voltage_raw}/1024) \times 1.8V = (524/1024) \times 1.8V = 0.92V$$

15 QT

运行QT测试程序：

▼ Shell

```
1 root@ido:~# source /usr/local/qt5/env.sh
2 root@ido:~# /usr/local/qt5/demo/widgets/animation/moveblocks/moveblocks
```

测试QT虚拟键盘：

```
1 root@ido:~# source /usr/local/qt5/env.sh
2 root@ido:~# /usr/local/qt5/demo/virtualkeyboard/basic/basic
```

16 RealTime Linux 实时性系统

16.1 安装测试工具包

```
1 $ sudo apt update
2 // 安装cpu压力测试工具
3 $ sudo apt install stress
4 // 安装rt测试工具
5 $ sudo apt-get install rt-tests
```

16.2 测试实时性性能

16.2.1 开启压力测试:

```
1 // 开启性能模式
2 $ sudo echo performance | tee $(find /sys/ -name *governor) /dev/null || true
3 // 开启压力测试
4 $ stress --cpu 4 --io 4 --vm 2 --vm-bytes 128M --timeout 3600s &
```

16.2.2 使用cyclicttest

```
1 $ sudo cyclicttest -t 4 -p 80 -m -S -D 1h
2 #参数详解:
3 -t:指定测试线程的数量
4 -p:设置线程的优先级 (优先级范围通常在1到99之间 (99 为最高优先级) )
5 -m:锁定内存
6 -S:使用 SCHED_FIFO 调度策略 (SCHED_FIFO 是一种先入先出的实时调度策略, 它确保高优先级线
   程在运行时不会被较低优先级的线程打断, 直到它主动放弃 CPU。)
7 -D:指定测试时间 (m:分钟; h:小时; d:天)
```

```
ido@ido:~$ sudo cyclicttest -t 4 -p 99 -m -S -D 1h
# /dev/cpu_dma_latency set to 0us
policy: fifo: loadavg: 5.13 5.24 5.05 5/382 17820

T: 0 ( 2243) P:99 I:1000 C:2768073 Min:      4 Act:      5 Avg:      4 Max:      34
T: 1 ( 2244) P:99 I:1500 C:1845369 Min:      4 Act:      4 Avg:      4 Max:      31
T: 2 ( 2245) P:99 I:2000 C:1384013 Min:      4 Act:      7 Avg:      5 Max:      29
T: 3 ( 2246) P:99 I:2500 C:1107197 Min:      4 Act:      5 Avg:      5 Max:      32
```