

IDO-EVB3562-V2 Linux使用手册

IDO-EVB3562-Ubuntu 系统

1 调试

1.1 调试串口

1.2 adb调试

1.3 ssh调试

2 UART

2.1 RS232

2.2 TTL

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 Buletooth

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 ov13855(J32)

预览

抓图

10.2 ov13855(J27)

预览

抓图

11 LED

12 RTC

12.1 读取时间

12.2 设置时间

13 硬盘

14 CAN

15 扩展接口

15.1 GPIO

15.2 ADC

16 系统使用说明

16.1 QT5

16.1.1 3d测试

16.2 glmark2

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

2.1 RS232

2.2 TTL

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 Buletooth

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 ov13855(J32)

预览

抓图

10.2 ov13855(J27)

预览

抓图

11 LED

12 RTC

12.1 读取时间

12.2 设置时间

13 硬盘

14 CAN

15 扩展接口

15.1 GPIO

15.2 ADC

16 系统使用说明

16.1 QT5

16.1.1 3d测试

16.2 glmark2

IDO-EVB3562-Buildroot系统

1 调试

1.1 串口调试

1.2 adb调试

www.industio.cn

- 1.3 ssh调试
- 2 串口
 - 2.1 RS232
 - 2.2 TTL
- 3 USB
- 4 Micro SD
- 5 Ethernet
- 6 WIFI
 - 6.1 连接热点
 - 6.1.1 命令行连接
- 7 Buletooth
 - 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 OV13855(J32)
 - 预览
 - 抓图
 - 10.2 OV13855(J27)
 - 预览
 - 抓图
- 11 LED
- 12 RTC
 - 12.1 读取时间
 - 12.2 设置时间
- 13 硬盘
- 14 CAN

www.industio.cn

15 扩展接口

15.1 GPIO

15.2 ADC

16 系统使用说明

16.1 QT5

16.1.1 3d测试

16.2 glmark2



IDO-EVB3562-V2A 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

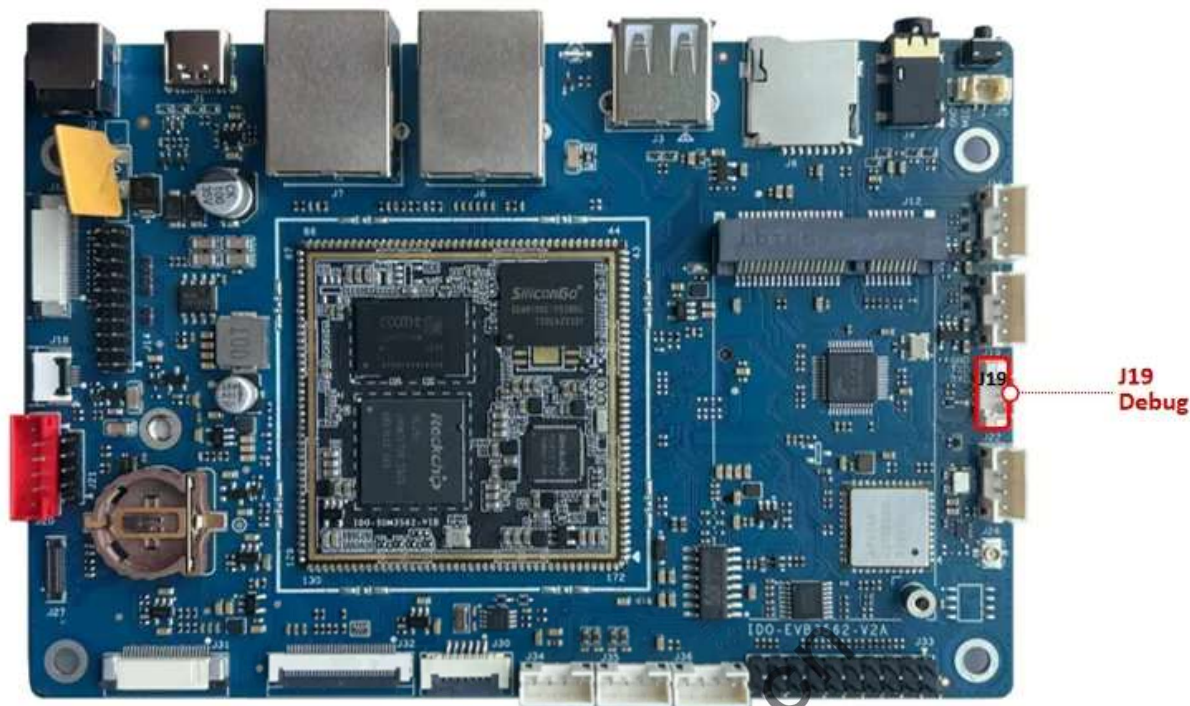
版本	修订内容	修订	审核	日期
V1.0	创建文档	MHK	IDO	2025/04/06
V1.1	更新文档	MHK	IDO	2025/04/25

IDO-EVB3562-Ubuntu 系统

1 调试

1.1 调试串口

主板调试串口位于J19，建议使用配套的usb转串口工具，如下图所示：



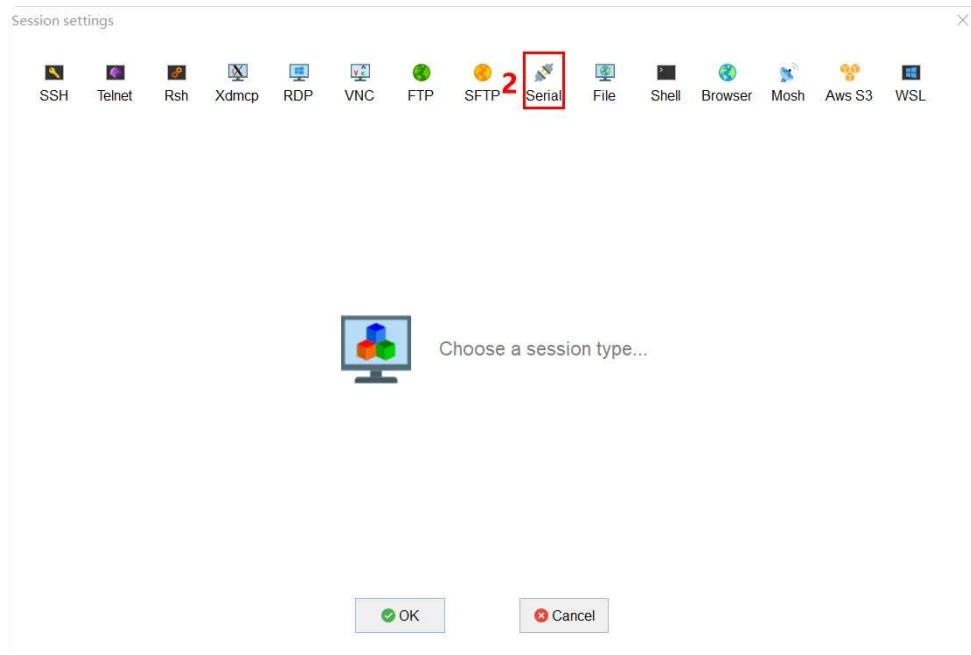
1. 打开MobaXterm，下载链接如下：

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_v3.6.zip	2024-04-24 14:30	zip文件	39.99MB
☐ DriverAssitant_v3.1.zip	2024-04-24 11:52	zip文件	9.36MB

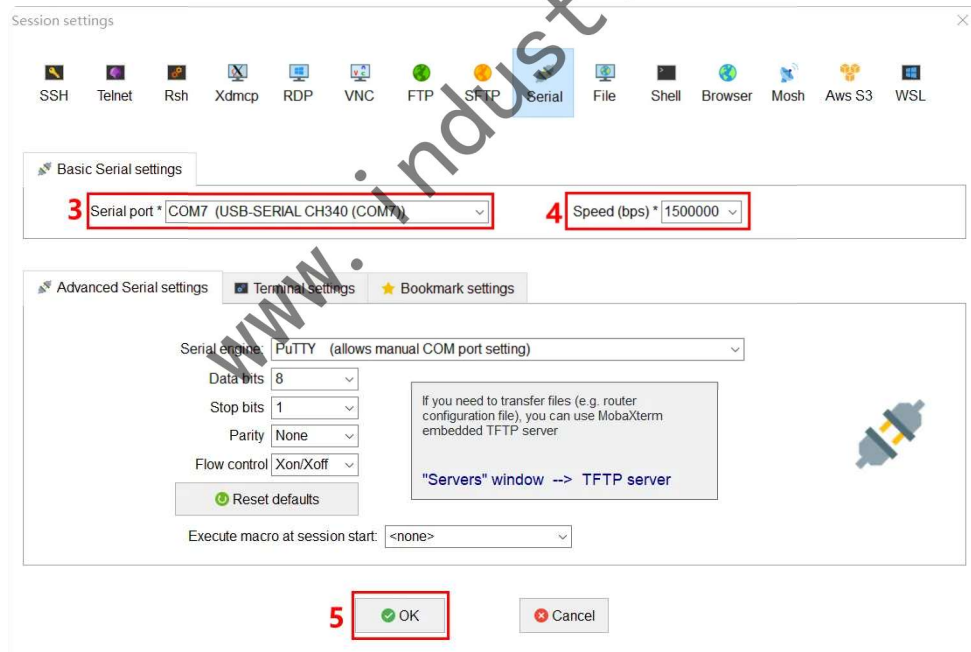
2. 选择sessionSerial，如下图所示：



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：

```
root@ido:~# cd /
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:/#
```

1.2 adb调试

主板adb接口位于J1，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:\Users\industio_mhk>adb shell
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:/#
```

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：123456
3. 端口：22

ssh调试，如下图所示：

```
[C:\~]$ ssh root@192.168.0.143
```

```
Connecting to 192.168.0.143:22...  
Connection established.  
To escape to local shell, press 'Ctrl+Alt+J'.
```



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

```
System information as of Mon Dec 23 16:36:44 CST 2024
```

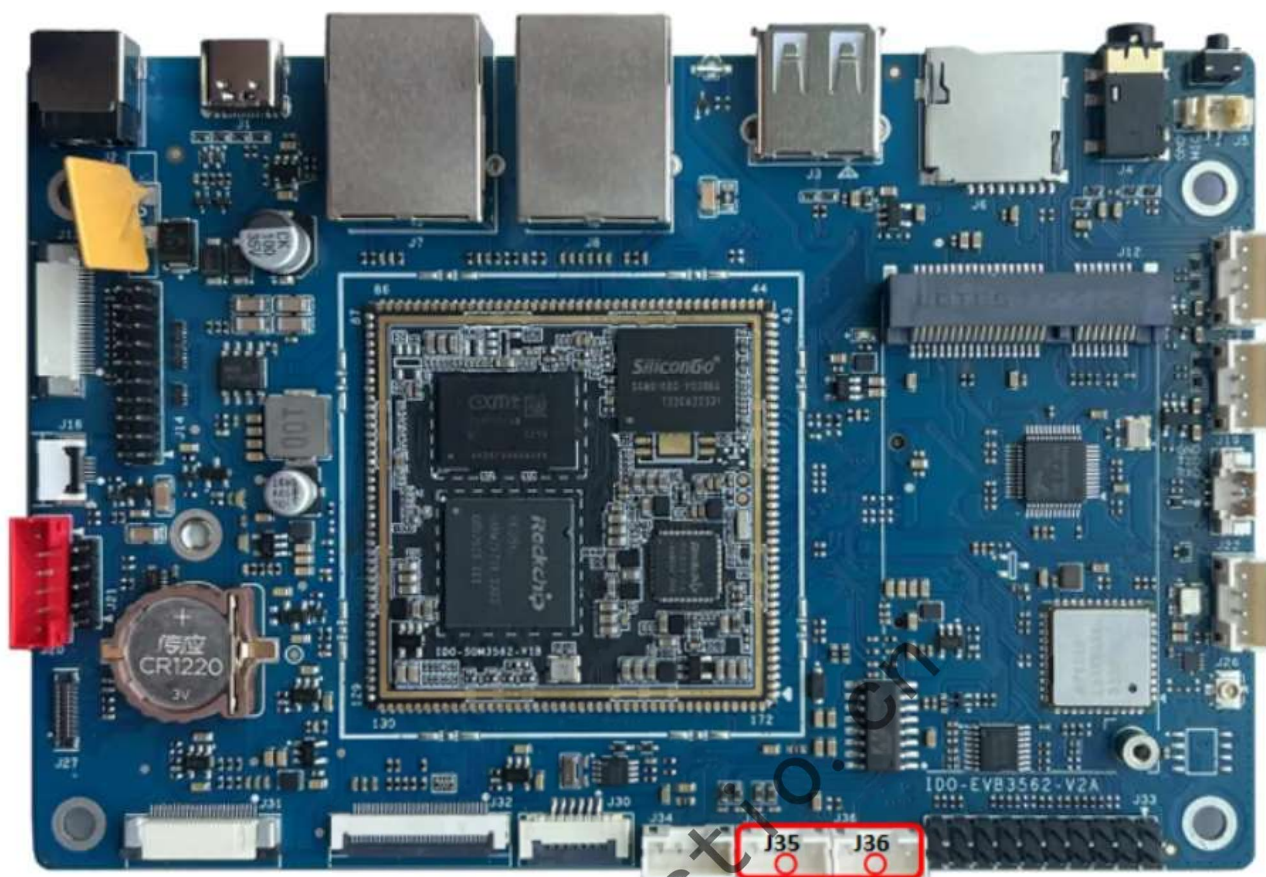
```
System load:  0.06 0.05 0.03   Up time:       11 min           Local users:  2  
Memory usage: 8 % of 3914MB   IP:           192.168.0.143  
Usage of /:   84% of 4.7G
```

```
Last login: Mon Dec 23 16:25:18 2024  
root@ido:~#
```

2 UART

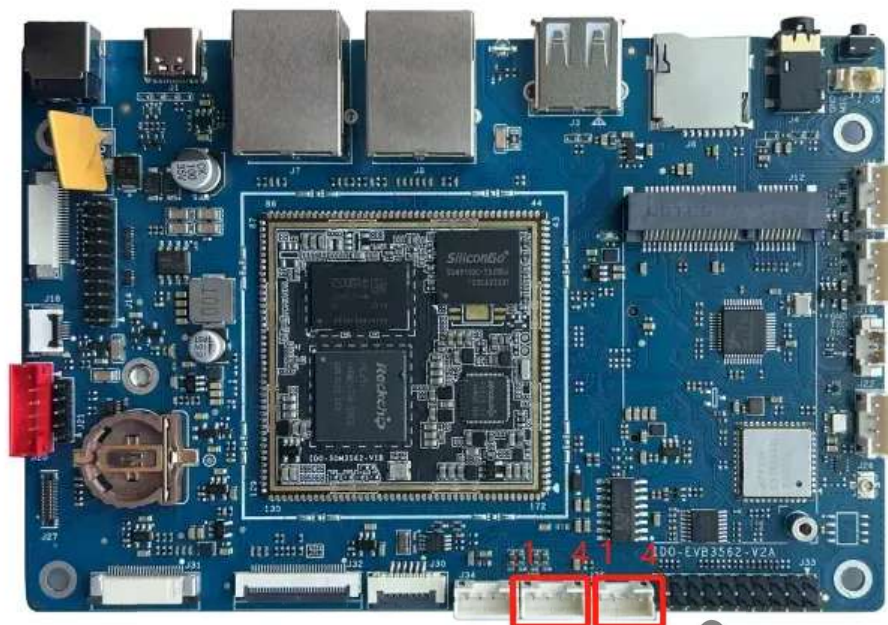
主板共配置4路串口（除调试串口），如下图所示：

2.1 RS232



2x RS232
J35~J36

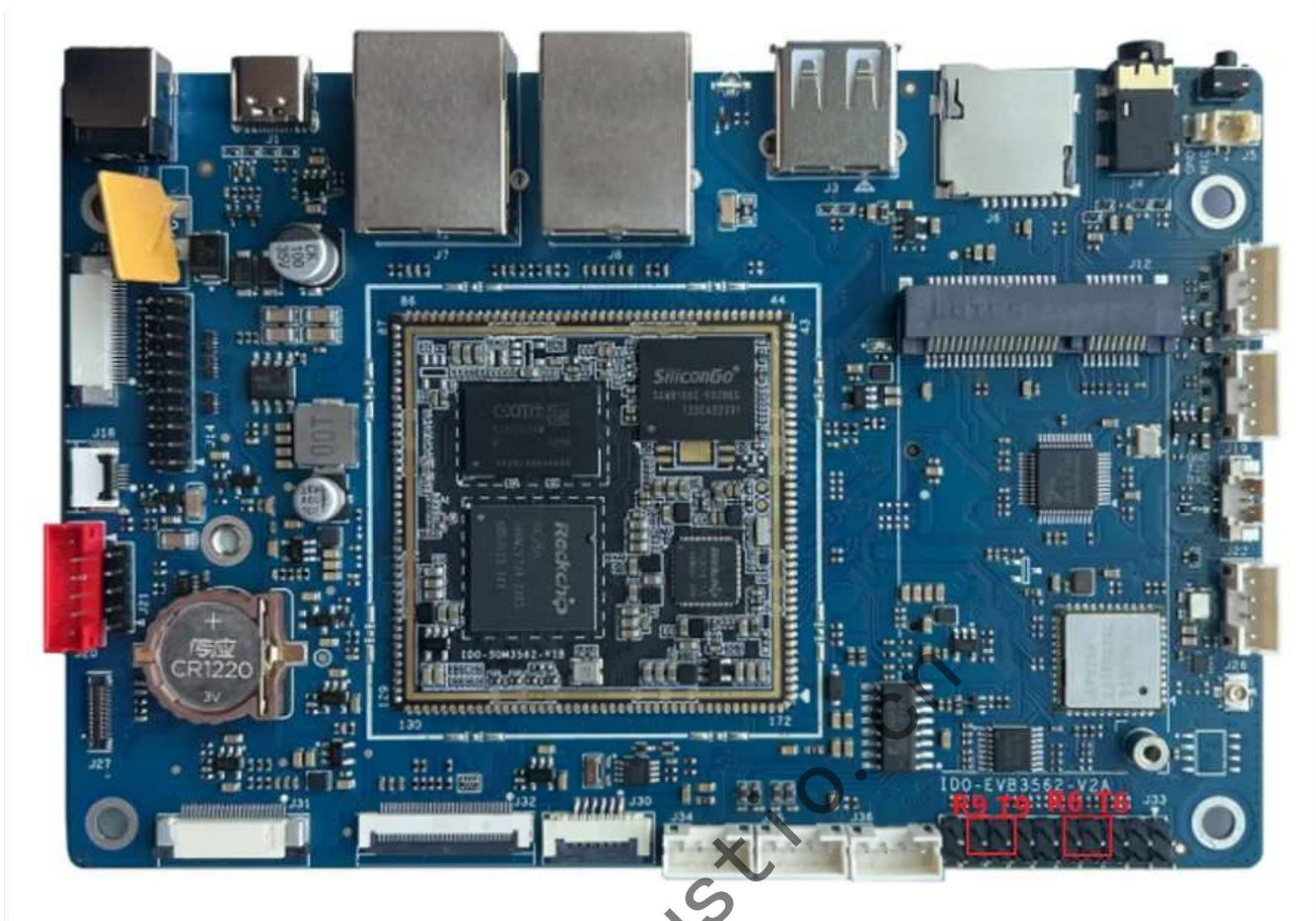
丝印	电平类型	设备节点	说明
J35	RS232	/dev/ttyS8	/
J36	RS232	/dev/ttyS3	/



引脚定义:

序号	定义	电平/V	说明
1	VCC	5V/3.3V	默认5V电源输出，可配置为3.3V
2	RX	/	RS232-接收
3	TX	/	RS232-发送
4	GND	GND	电源地

2.2 TTL



设备节点，如下表所示：

序号	电平类型	设备节点	说明
1	TTL	/dev/ttyS9	/
2	TTL	/dev/ttyS6	默认配置CAN0，此接口无效

使用microcom可以进行串口收发测试，命令如下：

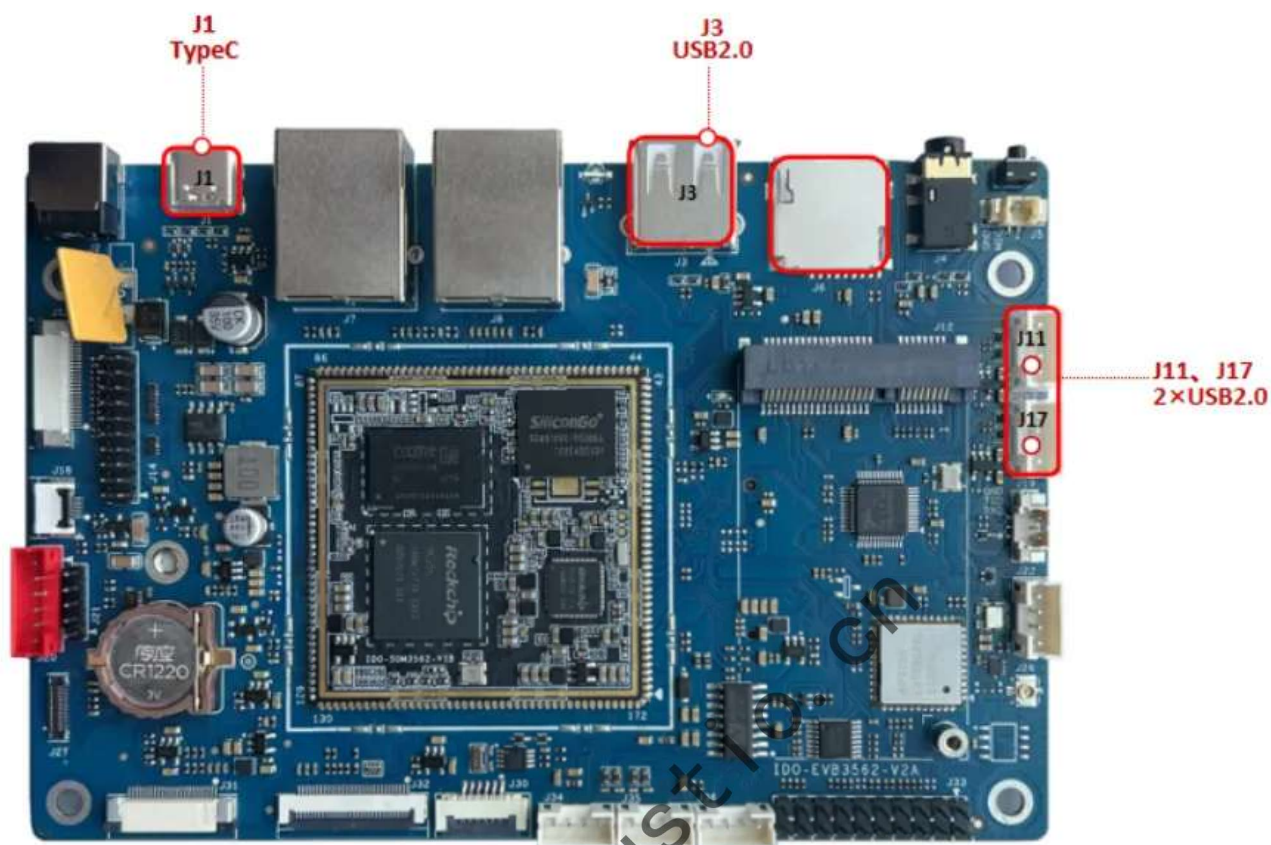
```
1 root@ido:/# microcom -s 115200 -p /dev/ttyS3
```

注意：

- 1.测试完成，按Ctrl+x退出。
- 2./dev/ttyS6 默认不通，此功能需要修改软硬件。

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	J1	OTG
USB2	J3	USB2.0 HOST
USB3	J11	USB2.0 HOST
USB4	J17	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@ido:/# mount
2 ...
3 /dev/sda1 on /mnt/udisk/5FD90FD95657F045 type vfat ...
```

USB电源控制，如下表所示：

USB端口	默认状态	动作	命令
USB2	开启	关闭电源	echo 0 > /sys/class/leds/usb_host2/brightness
		开启电源	echo 1 > /sys/class/leds/usb_host2/brightness
USB3	开启	关闭电源	echo 0 > /sys/class/leds/usb_host3/brightness
		开启电源	echo 1 > /sys/class/leds/usb_host3/brightness
USB4	开启	关闭电源	echo 0 > /sys/class/leds/usb_host4/brightness
		开启电源	echo 1 > /sys/class/leds/usb_host4/brightness

供电控制说明：设备节点写"0"关闭电源，写"1"开启电源。

4 Micro SD

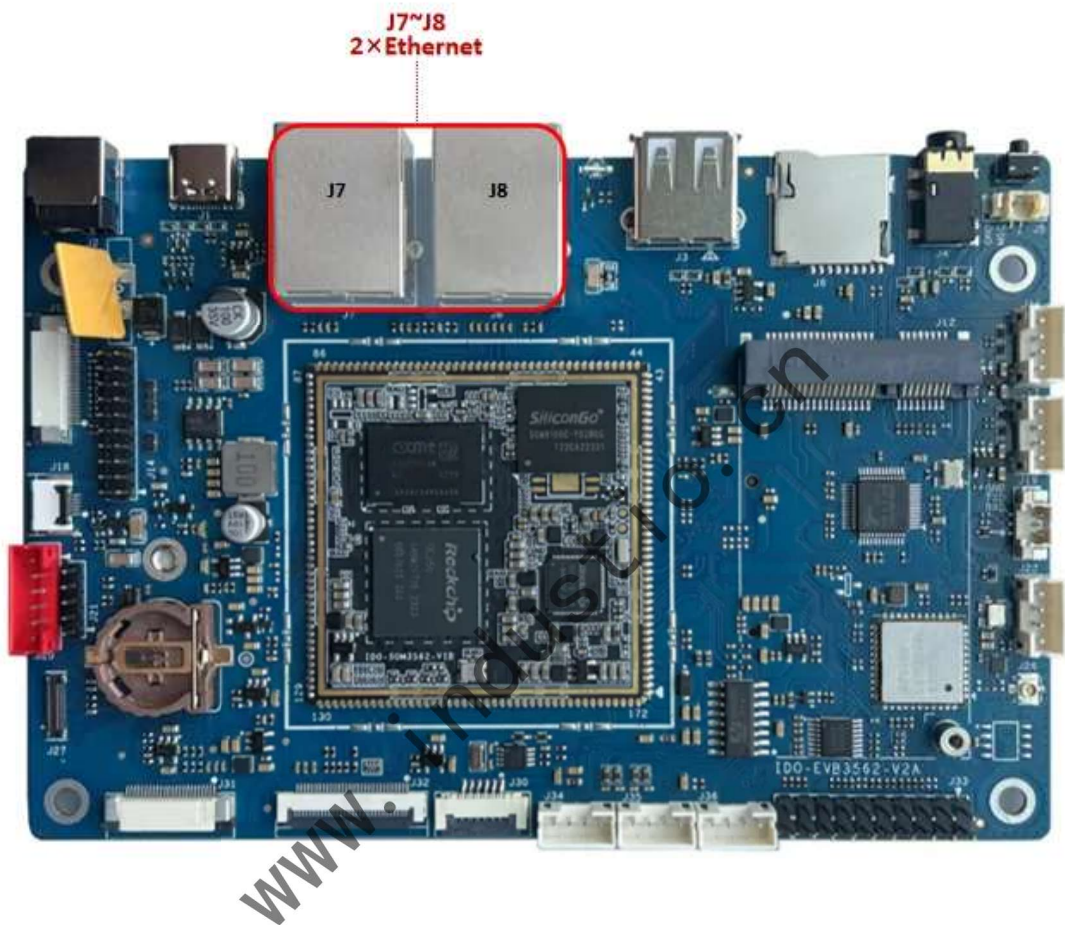
主板共配置一路Micro SD接口，插入SD卡后，默认挂载到/mnt/sdcard目录，如下所示：

▼ Shell

```
1 root@ido:/# fdisk -l
2 Disk /dev/ram0: 4 MiB, 4194304 bytes, 8192 sectors
3 Units: sectors of 1 * 512 = 512 bytes
4 Sector size (logical/physical): 512 bytes / 4096 bytes
5 I/O size (minimum/optimal): 4096 bytes / 4096 bytes
6 ...
7 Device          Boot Start      End  Sectors  Size Id Type
8 /dev/mmcbk1p1    8192 30449663 30441472 14.5G  c W95 FAT32 (LBA)
9
10 root@ido:/# mount
11 /dev/mmcbk0p8 on / type ext4 (rw,relatime)
12 devtmpfs on /dev type devtmpfs (rw,relatime,size=1988328k,nr_inodes=497082,mode=755)
13 sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)
14 proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
15 ...
16 /dev/mmcbk1p1 on /mnt/sdcard type vfat (rw,relatime,async,uid=1000,gid=1000,fmask=0022,dmask=0022,codepage=936,iocharset=utf8,shortname=mixed,errors=remount-ro)
```

5 Ethernet

主板共配置2路千兆网接口，如下图所示：



序号	接口位置	速率	网络节点
1	J7	千兆	eth0
2	J8	千兆	eth1

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@ido:/# ifconfig eth0 192.168.3.200 netmask 255.255.255.0 up
2 root@ido:/# route add default gw 192.168.3.1
3 root@ido:/# ping 8.8.8.8
4 PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
5 64 bytes from 8.8.8.8: icmp_seq=1 ttl=113 time=12.7 ms
6 64 bytes from 8.8.8.8: icmp_seq=2 ttl=113 time=14.1 ms

```

如需设置永久静态IP，则可通过添加/etc/netplan/01-netcfg.yaml，命令如下：

```

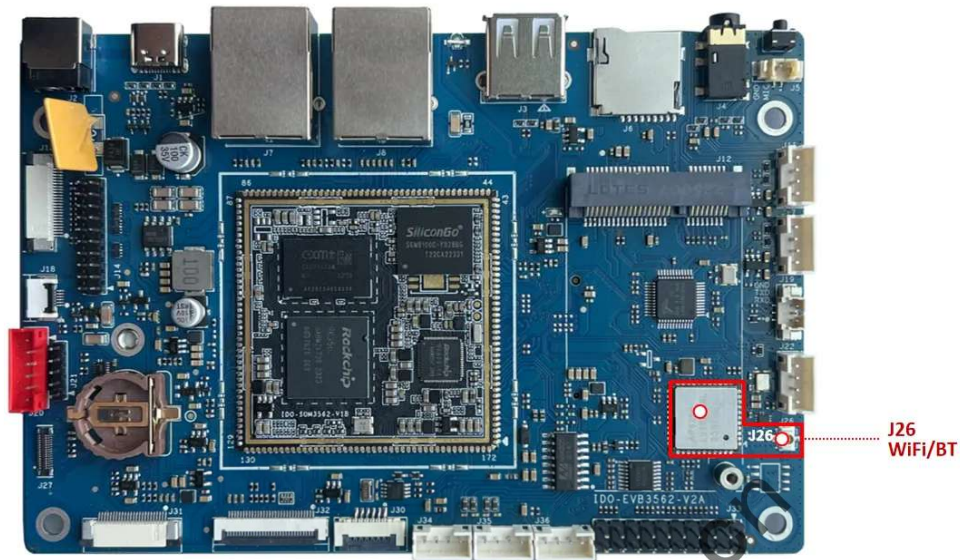
1 root@ido:/# touch /etc/netplan/01-netcfg.yaml
2 root@ido:/# vim /etc/netplan/01-netcfg.yaml
3 root@ido:/# cat /etc/netplan/01-netcfg.yaml
4 network:
5     version: 2
6     ethernet:
7         eth0:
8             dhcp4: no
9             addresses: [192.168.3.121/24]
10            nameservers:
11                addresses: [192.168.3.1, 8.8.8.8, 114.114.1
12                    14.114]
13            routes:
14                - to: 0.0.0.0/0
15                  via: 192.168.3.1
16                  metric: 100
17 #开启netplan服务
18 root@ido:/# systemctl enable systemd-networkd.service
19 Created symlink /etc/systemd/system/dbus-org.freedesktop.network1.service
20 → /lib/systemd/system/systemd-networkd.service.
21 Created symlink /etc/systemd/system/multi-user.target.wants/systemd-networ
22 kd.service → /lib/systemd/system/systemd-networkd.service.
23 Created symlink /etc/systemd/system/sockets.target.wants/systemd-networkd.
24 socket → /lib/systemd/system/systemd-networkd.socket.
25 Unit /etc/systemd/system/systemd-networkd-wait-online.service is masked, i
26 gnoring.
27 root@ido:/# systemctl start systemd-networkd.service
28
29 #立即生效
30 root@ido:/# netplan apply

```

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，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：

```
1 root@ido:/# ifconfig wlan0
2 wlan0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
3 ether c0:f5:35:12:c5:f8 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

命令如下：

```

1  #打开WiFi
2  root@ido:/# nmcli r wifi on
3  #扫描附近热点
4  root@ido:/# nmcli dev wifi
5  IN-USE BSSID SSID MODE CHAN RATE SIGN
AL B>
6      22:A3:8E:D0:51:42 HUAWEI-1EX2DK Infra 1 130 Mbit/s 100
7      ->
8      94:D9:B3:A6:B8:7A TP-LINK_B87A Infra 7 270 Mbit/s 100
9      ->
10     22:A3:8E:D0:51:46 HUAWEI-1EX2DK Infra 44 270 Mbit/s 100
11     ->
12     70:C6:DD:3B:2F:C4 Industio_2.4 Infra 11 130 Mbit/s 97
13     ->
14     78:60:5B:83:67:AE test_2.4G Infra 6 540 Mbit/s 95
15     ->
16     7A:60:5B:13:67:AE TPGuest_67AE Infra 6 540 Mbit/s 92
17     ->
18     7A:60:5B:73:67:AE -- Infra 6 540 Mbit/s 90
19     ->
20     44:F7:70:38:A7:18 CS_5G Infra 44 270 Mbit/s 87
21     ->
22     4E:F7:70:38:A7:18 -- Infra 44 270 Mbit/s 87
23     ->
24     78:60:5B:83:67:B0 test_5G Infra 40 540 Mbit/s 84
25     ->
26     7A:60:5B:F3:67:AE -- Infra 40 540 Mbit/s 84
27     ->
28     70:C6:DD:3B:2F:C7 Industio_5.8 Infra 153 270 Mbit/s 82
29     ->
30     9A:2C:0D:B3:E0:07 jjh Infra 11 130 Mbit/s 79
31     ->
32     44:F7:70:38:A7:19 CS Infra 1 270 Mbit/s 75
33     ->
34     B8:A1:4A:0A:F1:43 ChinaNet-MvDm Infra 4 130 Mbit/s 74
35     ->
36     ...
37  #连接热点
38  root@ido:/# nmcli dev wifi connect Industio_2.4 password industio666 ifnam
e wlan0
39  Device 'wlan0' successfully activated with 'a01dc09e-6e54-47b3-baf8-f91129
9a4482'.
40  root@ido:/# ifconfig wlan0
41  wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
42      inet 192.168.0.90 netmask 255.255.255.0 broadcast 192.168.0.255

```

```

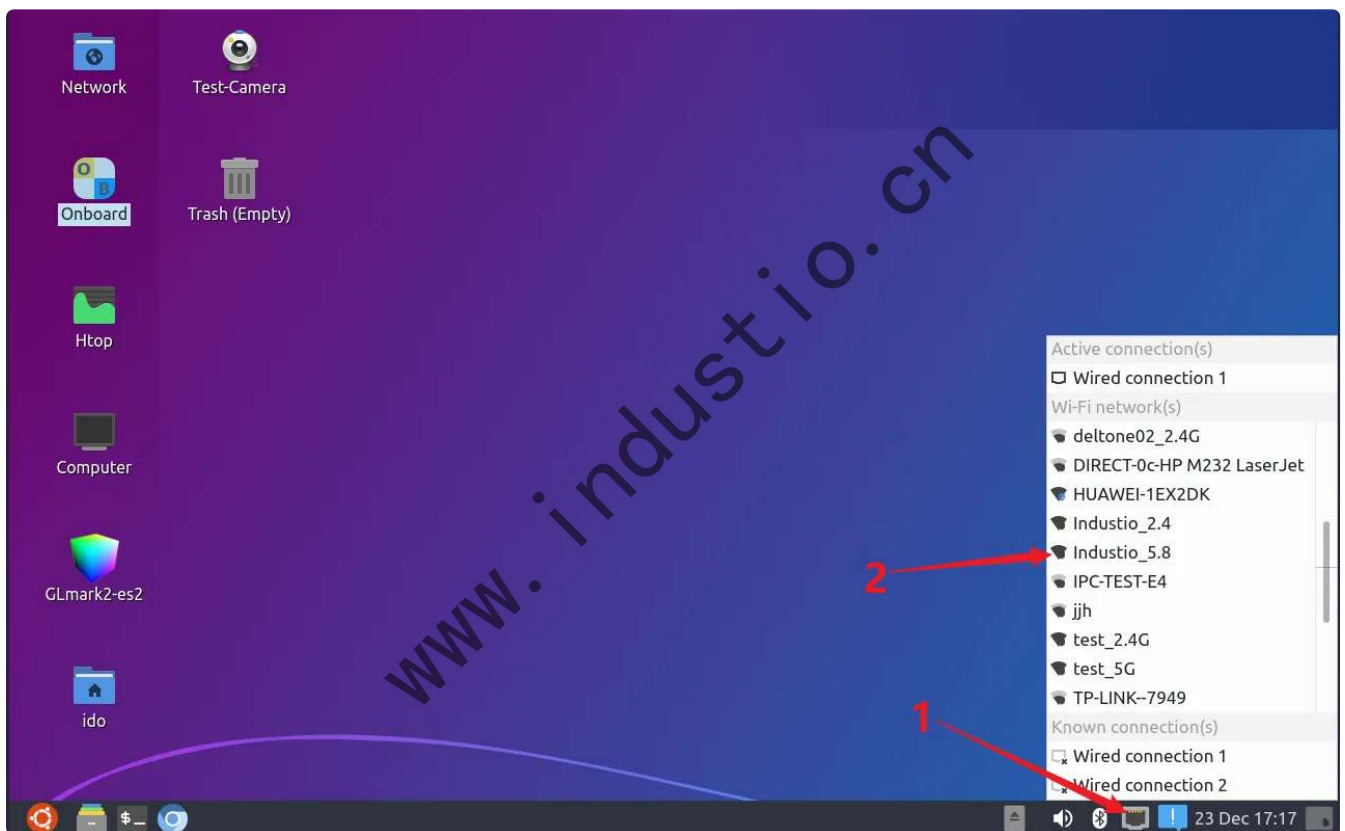
28      inet6 fe80::8b1a:eccc:ac59:639a prefixlen 64 scopeid 0x20<link>
29      ether c0:f5:35:12:c5:f8 txqueuelen 1000 (Ethernet)
30      RX packets 414 bytes 35675 (35.6 KB)
31      RX errors 0 dropped 9 overruns 0 frame 0
32      TX packets 40 bytes 6866 (6.8 KB)
33      TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
34

```

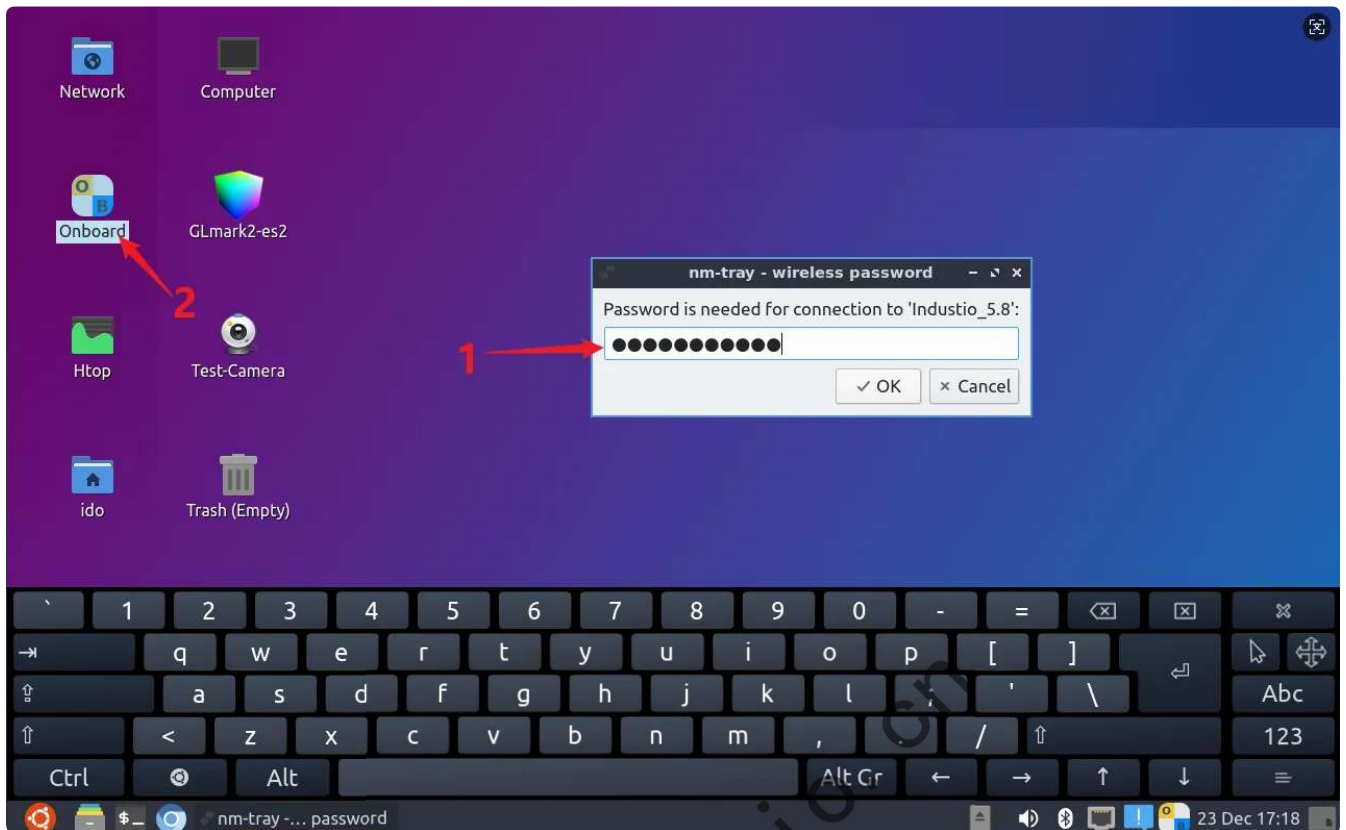
其中"Industio_2.4"为热点名称, "industio666"为热点密码。

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:

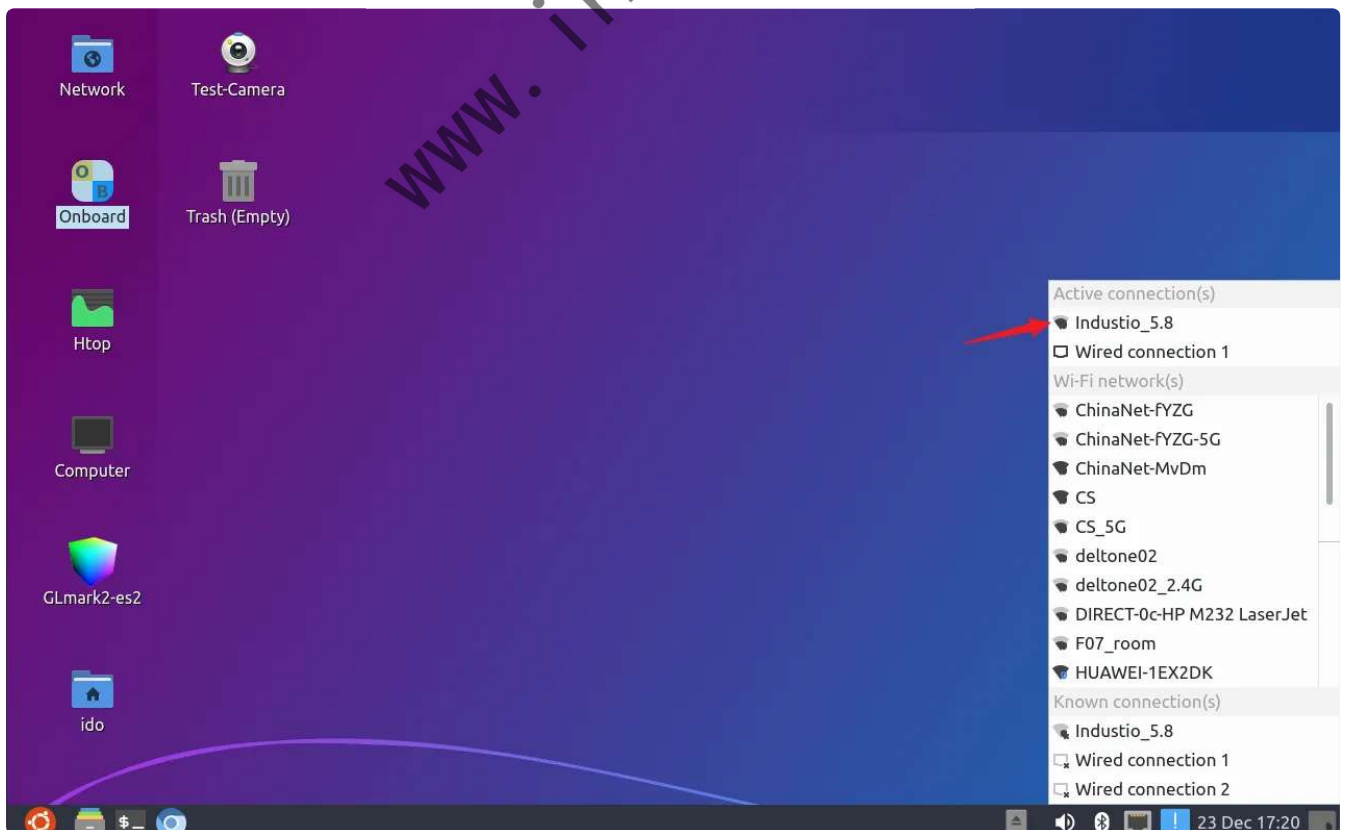


弹出密码输入框, 使用键盘输入密码 (如果没有接键盘, 可以使用软键盘Onboard), 如下图所示:



输入密码后，点击【OK】按钮连接热点。

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：



通过ifconfig 命令查看wlan0的IP地址正确:

```
1 root@ido:/# ifconfig wlan0
2 wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
3         inet 192.168.0.90 netmask 255.255.255.0 broadcast 192.168.0.255
4         inet6 fe80::bd5e:dd45:459b:2f31 prefixlen 64 scopeid 0x20<link>
5         ether c0:f5:35:12:c5:f8 txqueuelen 1000 (Ethernet)
6         RX packets 1470 bytes 117107 (117.1 KB)
7         RX errors 0 dropped 19 overruns 0 frame 0
8         TX packets 58 bytes 9434 (9.4 KB)
9         TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

7 Buletooth

7.1 开启蓝牙

开启蓝牙, 命令如下:

```
1 root@ido:/# hciconfig -a
2 hci0: Type: Primary Bus: UART
3     BD Address: F8:A6:61:58:22:A8 ACL MTU: 1021:8 SCO MTU: 64:1
4     UP RUNNING
5     RX bytes:934 acl:0 sco:0 events:70 errors:0
6     TX bytes:5494 acl:0 sco:0 commands:70 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: 'ido'
12    Class: 0x1c0000
13    Service Classes: Rendering, Capturing, Object Transfer
14    Device Class: Miscellaneous,
15    HCI Version: 5.0 (0x9) Revision: 0x50
16    LMP Version: 5.0 (0x9) Subversion: 0x6606
17    Manufacturer: Broadcom Corporation (15)
```

7.2 扫描设备

扫描设备，命令如下：

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

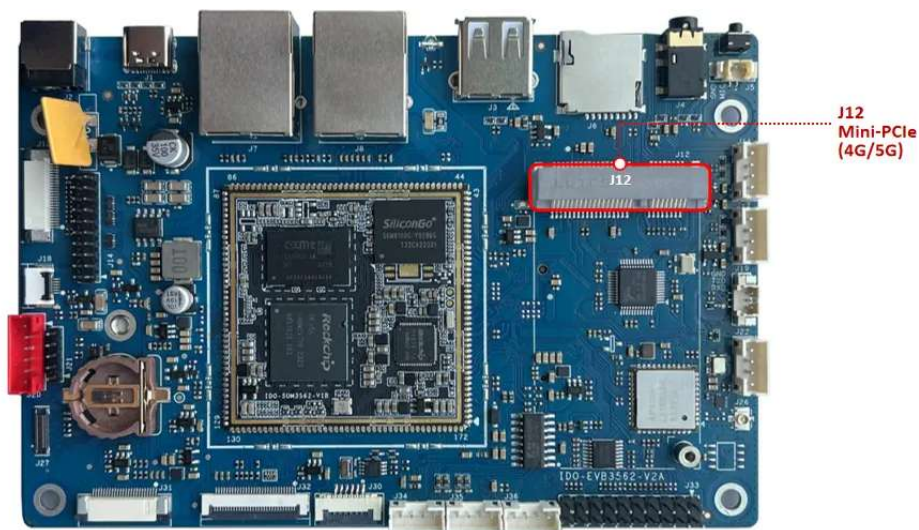
7.3 连接蓝牙设备

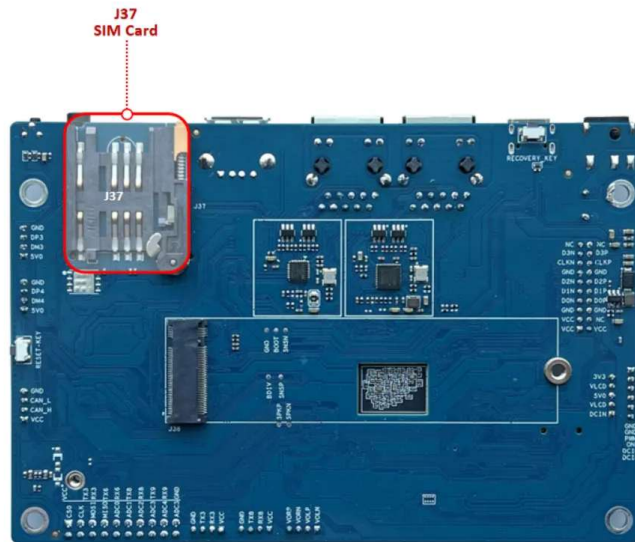
连接蓝牙设备，命令如下：

```
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 [bluetooth]# connect 7C:C1:80:09:DD:6C
7 [cainiaocl]#
```

8 4G

主板配置了一路4G接口，默认适配EC20模块，如下图所示：





开机自动拨号，拨号成功后，产生wwan0网络节点：

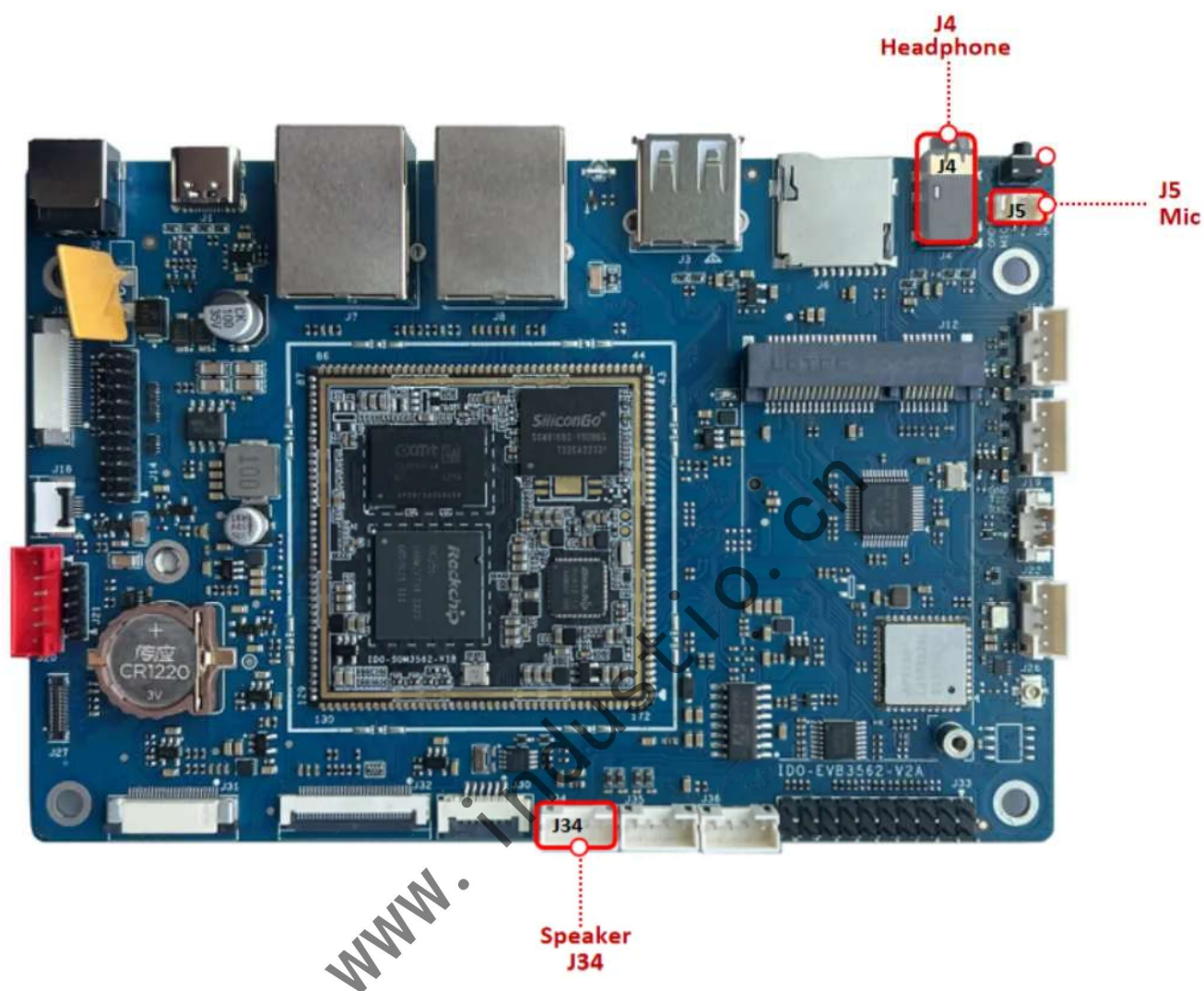
```
root@ido:/# ifconfig wwan0
wwan0    Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
        inet addr:10.122.45.92  P-t-P:10.122.45.92  Mask:255.255.255.248
        UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
        RX packets:12 errors:0 dropped:0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

测试4G上外网功能，命令如下：

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

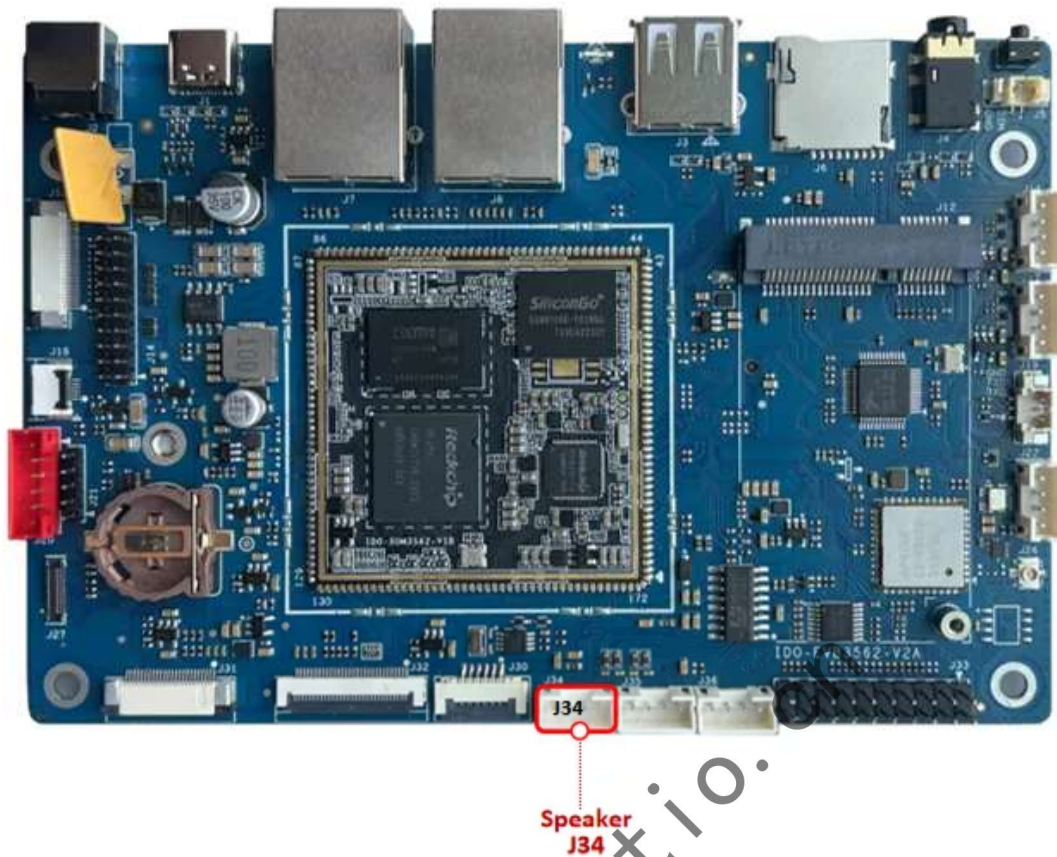
9 音频

主板音频接口如下所示：



9.1 喇叭

双声道Lineout接口位于J34，如下图所示：

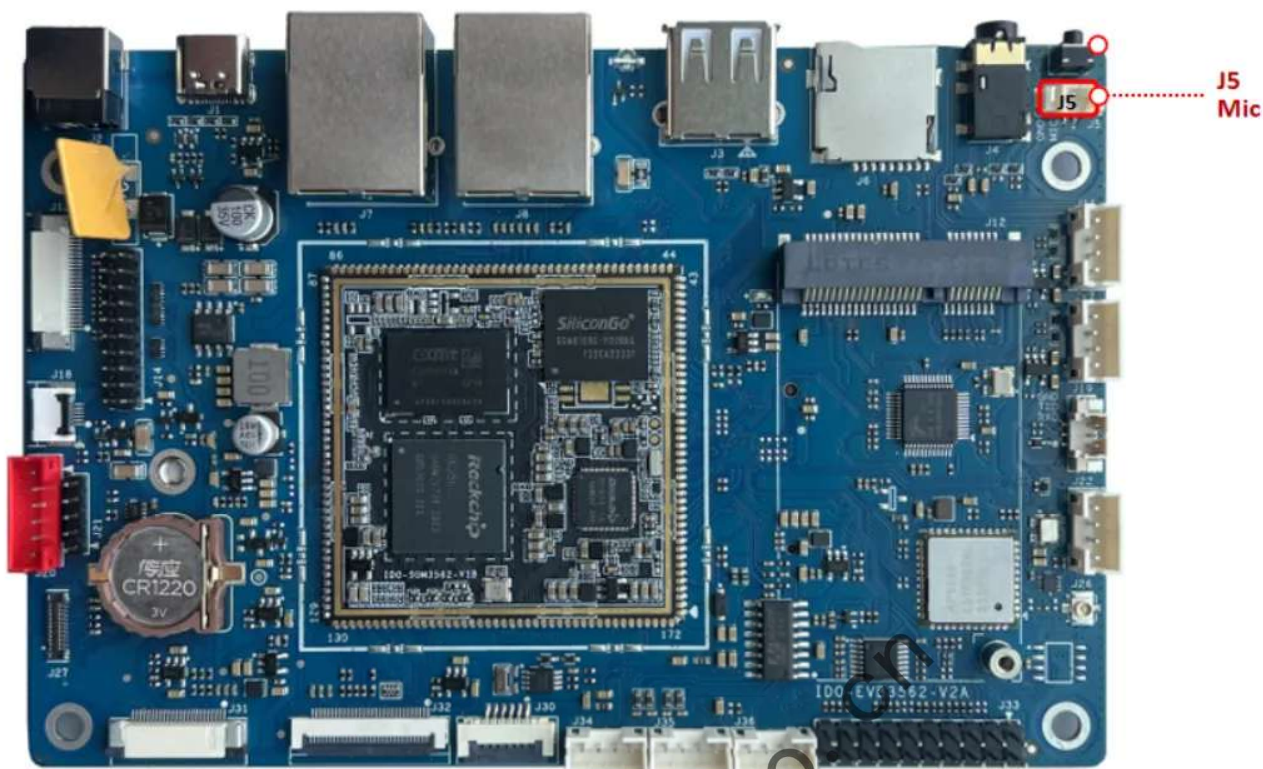


播放音频，命令如下：

```
1 #打开spk switch
2 root@ido:/# amixer cset numid=37 2 -c 1
3 #打开mic switch
4 root@ido:/# amixer cset numid=38 1 -c 1
5
6 #播放音频
7 root@ido:/# aplay -D hw:1,0 ./xinhuan.wav
```

9.2 MIC

MIC如下图所示：

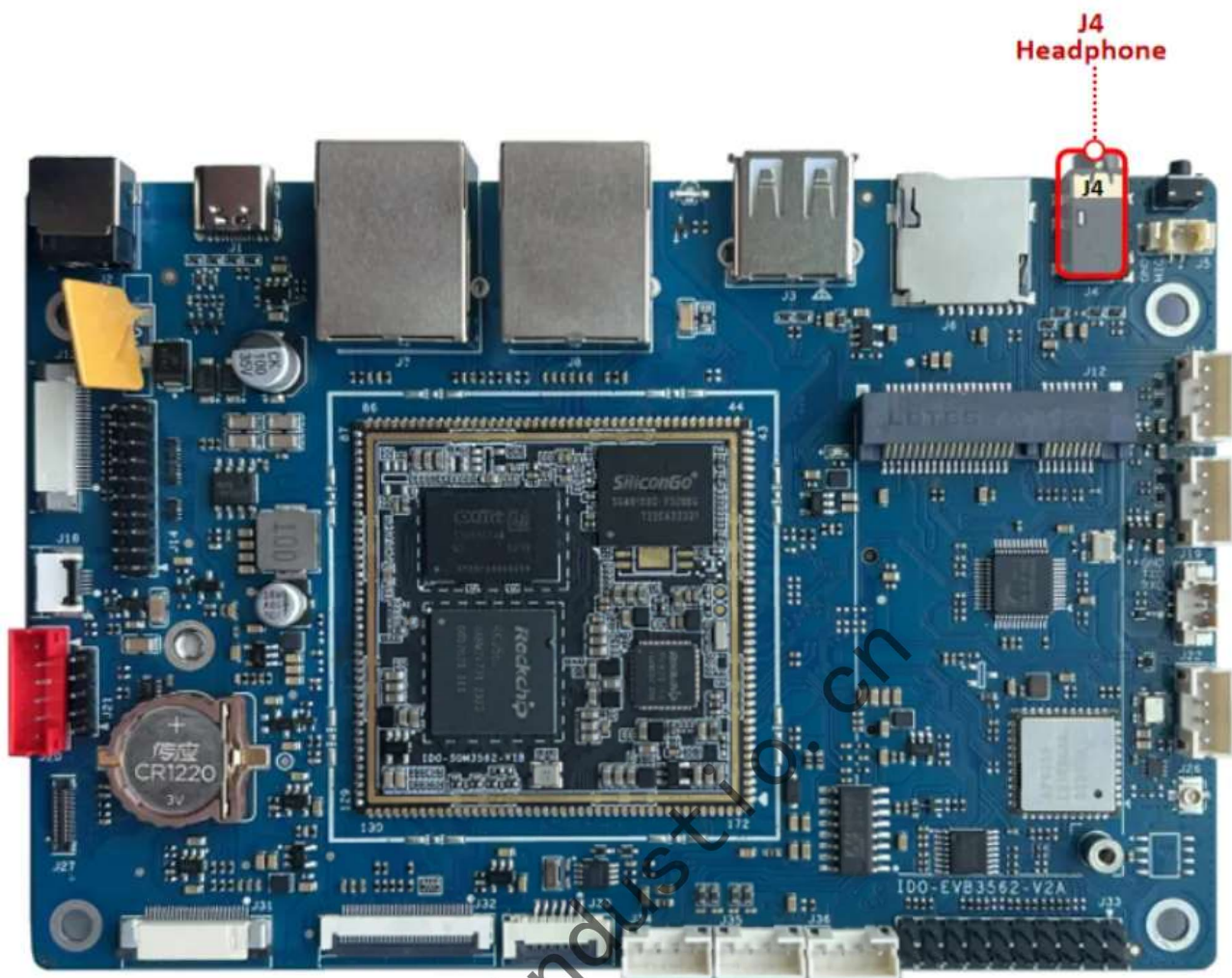


录音，命令如下：

```
1 root@ido:/# arecord -D hw:1,0 -r 48000 -c 2 -f S16_LE test.wav
```

9.3 耳机

耳机如下图所示：

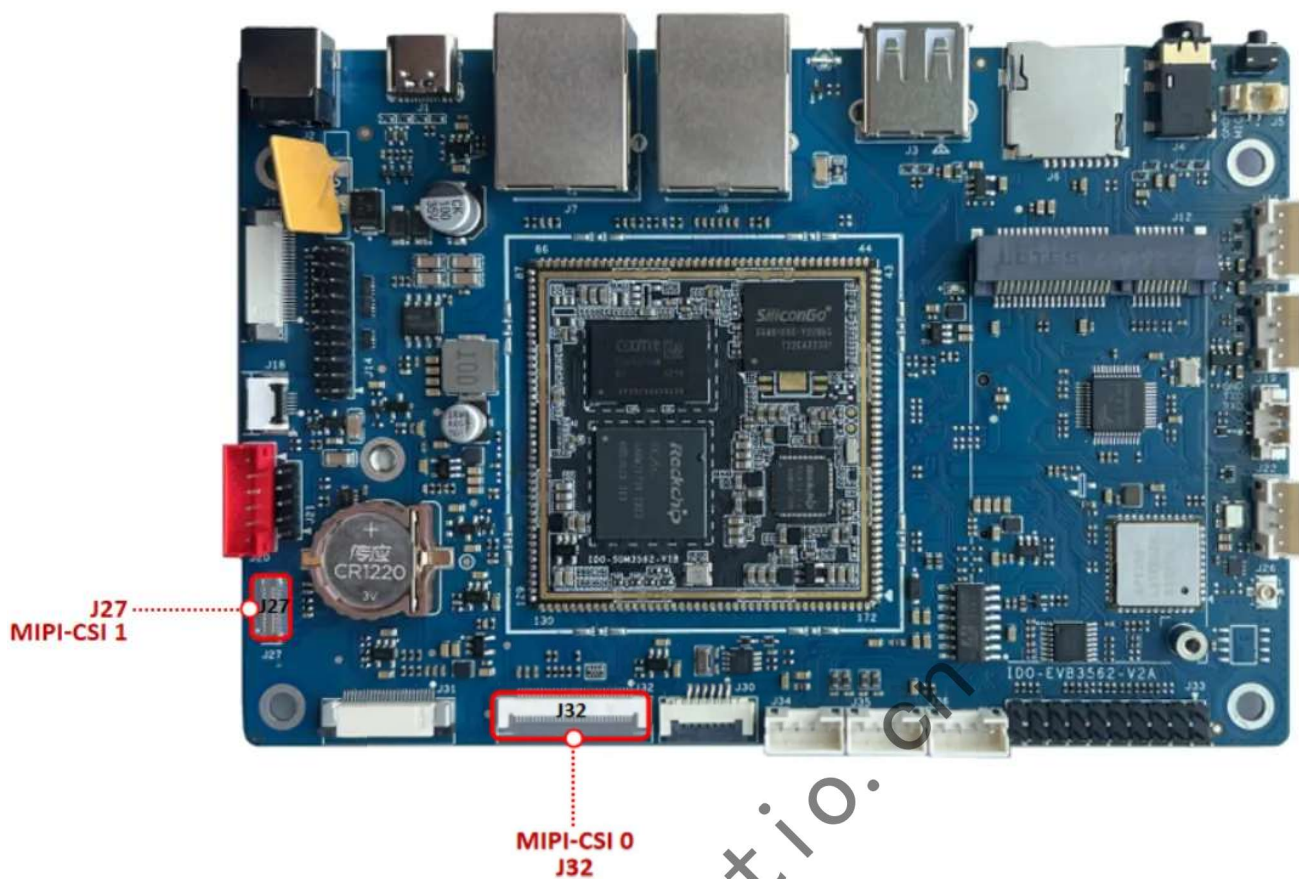


播放音频，命令如下：

```
1 root@ido:/# aplay -D hw:1,0 ./xinhuan.wav
```

10 Camera

主板配置2路摄像头，型号为ov13855（J27）和ov13855（J32），如下所示：



10.1 ov13855(J32)

预览

```
▼ Shell |
1 root@ido:/# export DISPLAY=:0
2 root@ido:/# gst-launch-1.0 v4l2src device=/dev/video22 ! video/x-raw, format=NV12, width=1920, height=1080, framerate=30/1 ! videoconvert ! autovideosink sync=false
```

抓图

▼ Shell

```
1 root@ido:/# export DISPLAY=:0
2 root@ido:/# v4l2-ctl -d /dev/video22 --set-fmt-video=width=1920,height=1080,pixelformat=NV12 --stream-mmap=4 --stream-to=/tmp/test_vide22_nv12.yuv --stream-count=1 --stream-skip=30
```

10.2 ov13855(J27)

预览

▼ Shell

```
1 root@ido:/# export DISPLAY=:0
2 root@ido:/# gst-launch-1.0 v4l2src device=/dev/video29 ! video/x-raw, format=NV12, width=1920, height=1080, framerate=30/1 ! videoconvert ! autovideosink sync=false
```

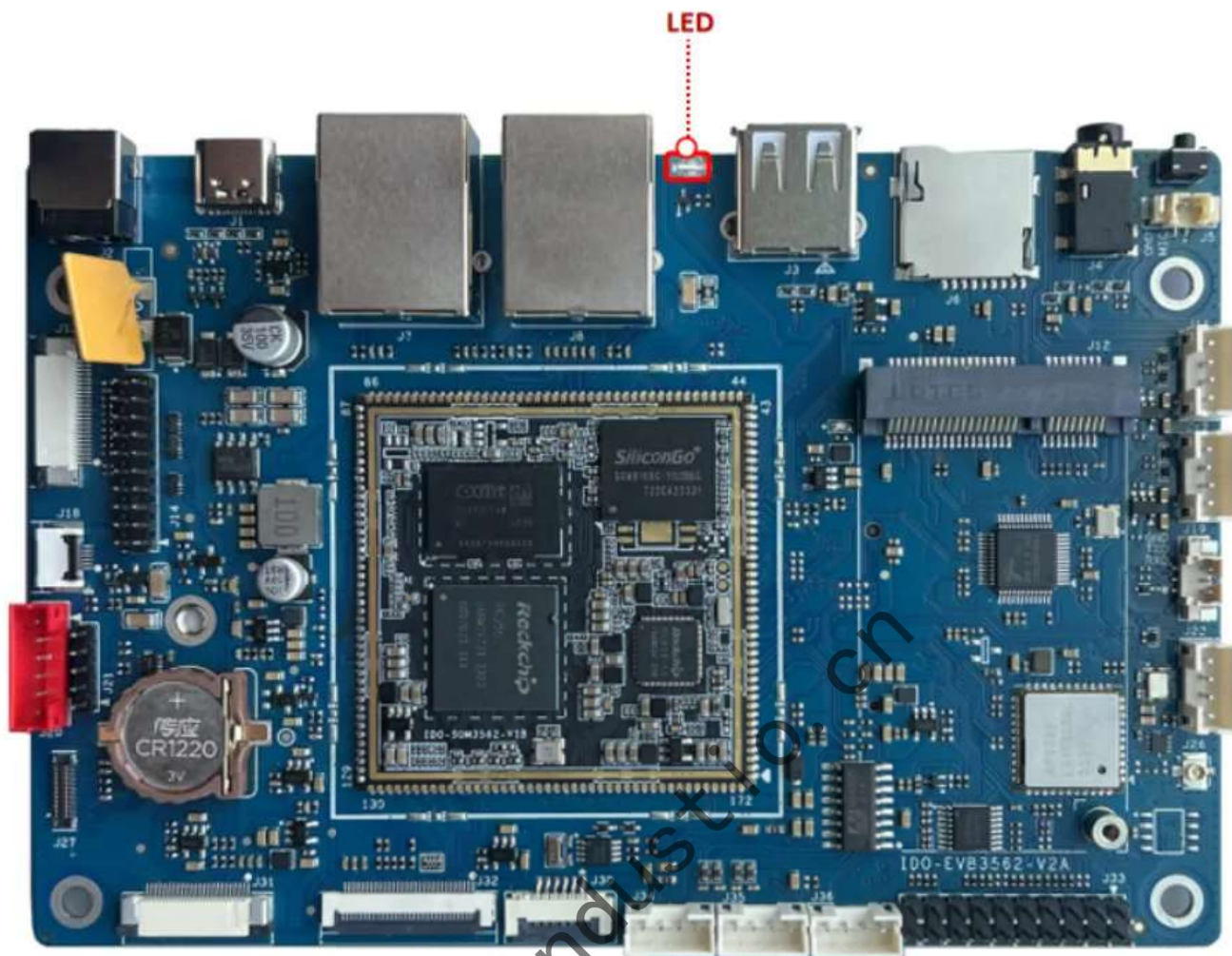
抓图

▼ Shell

```
1 root@ido:/# export DISPLAY=:0
2 root@ido:/# v4l2-ctl -d /dev/video29 --set-fmt-video=width=1920,height=1080,pixelformat=NV12 --stream-mmap=4 --stream-to=/tmp/test_vide22_nv12.yuv --stream-count=1 --stream-skip=30
```

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，系统中对应的设备节点为/dev/rtc0，如下图所示：



12.1 读取时间

读取RTC时间，命令如下：

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

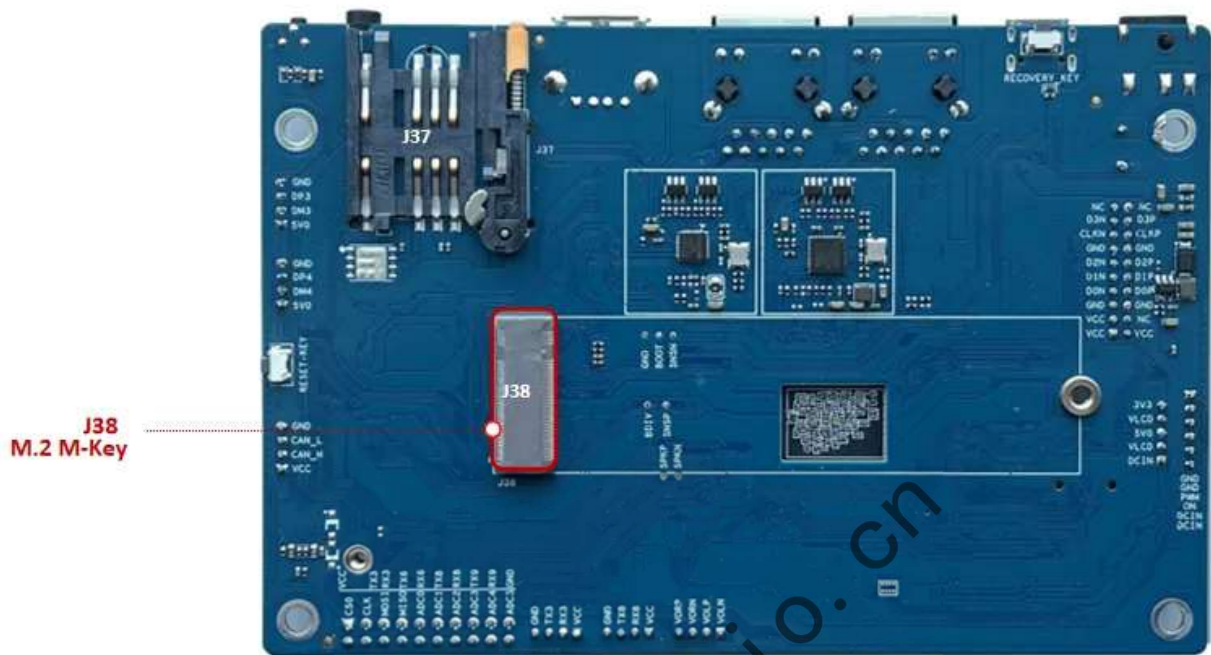
12.2 设置时间

设置RTC时间，命令如下：

```
1 root@ido:/# date -s '2024-8-22 15:30:00'
2 Tue Aug 22 15:30:00 UTC 2024
3 root@ido:/# hwclock -w
4 root@ido:/# hwclock -r
5 Tue Aug 22 15:30:05 2024 0.000000 seconds
```

13 硬盘

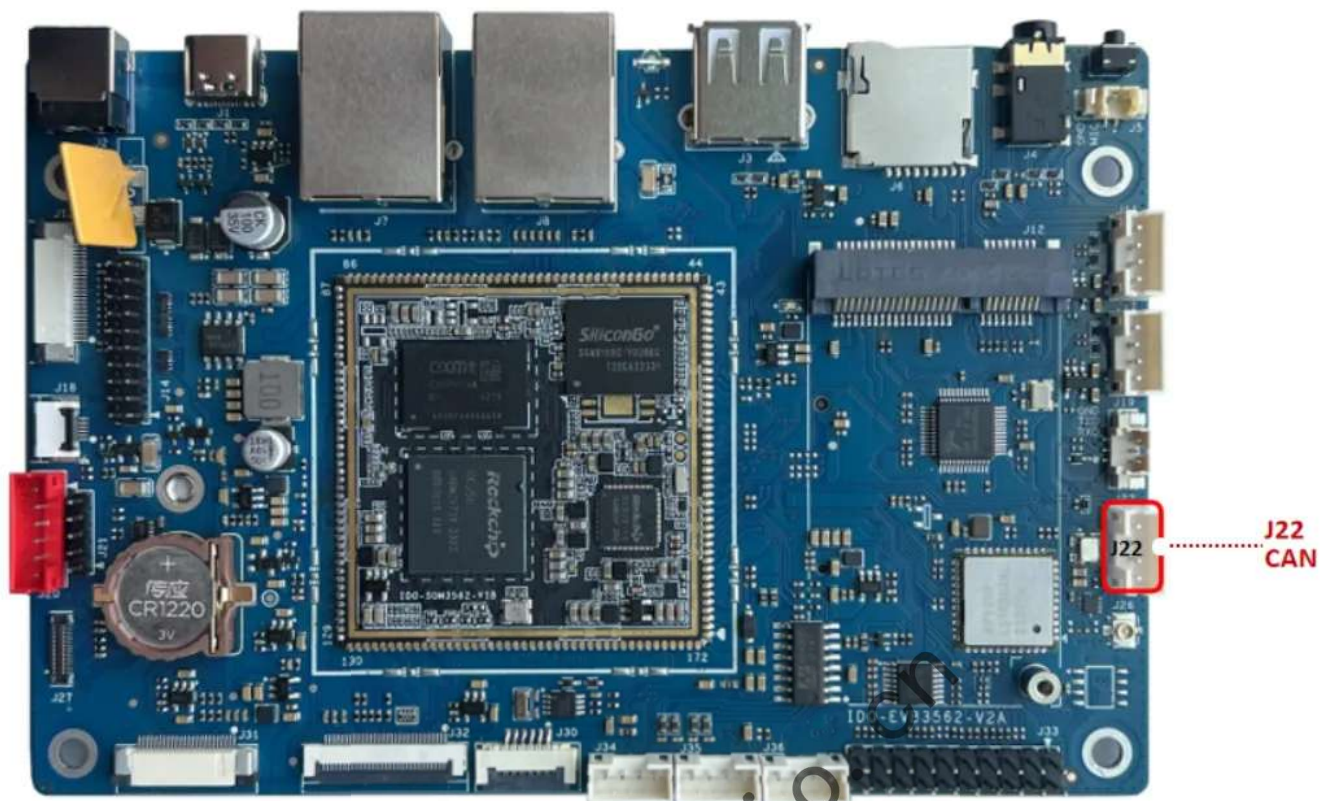
主板配置了一路硬盘接口（M.2），位于主板背面J38，如下图所示：



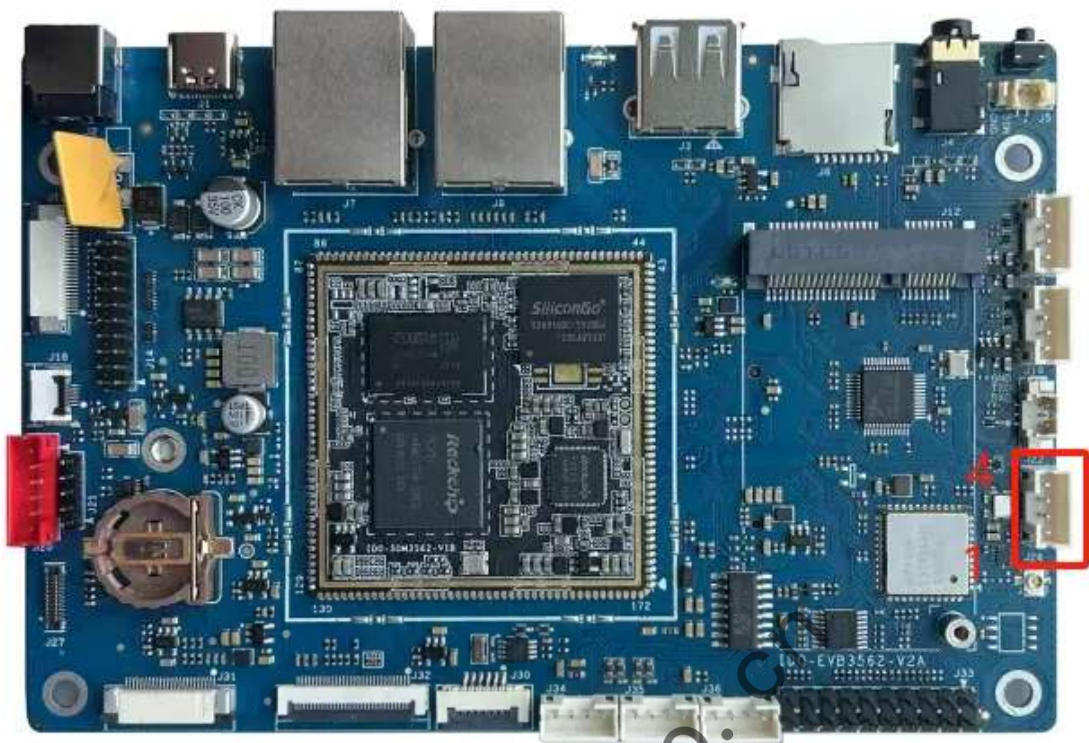
注意：PCIE功能目前和J8(eth1)冲突，如果需要此功能需要更改软硬件。

14 CAN

主板配置一路CAN接口，如下图所示。



序号	丝印	设备节点
1	J22	CAN0



引脚定义：

序号	定义	电平/V	说明
1	VCC	5V/3.3V	默认5V供电输出，可配置为3.3V
2	CAN0_H	/	CAN信号
3	CAN0_L	/	
4	GND	GND	电源地

CAN命令行测试命令：

```
1  #关闭can0设备
2  ip link set can0 down
3
4  #普通can协议 (1.0)
5  ip link set can0 type can bitrate 125000 triple-sampling on
6
7  #查看can信息, 波特率等等
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 #can格式发送
17 cansend can0 123#1122334455667788
```

15 扩展接口

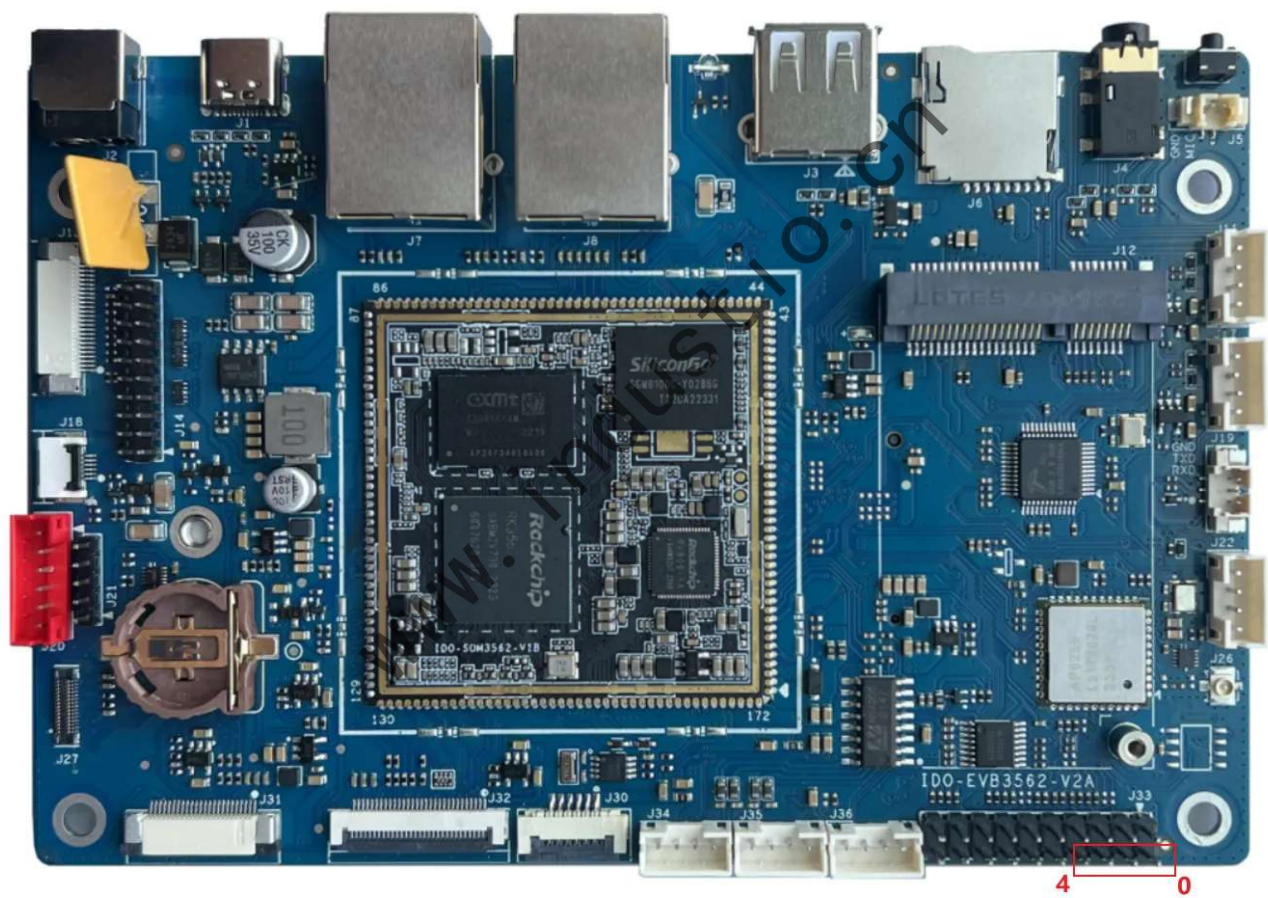
主板配置了一排扩展接口，位于J33，扩展接口的各个引脚功能，如下表所示：

pin	功能	pin	功能
1	5V输出	2	SPI0_CSN0_M0/PWM5_M0
3	UART3_TX	4	SPI0_CLK_M0/PWM0_M0
5	UART3_RX	6	SPI0_MOSI_M0/PWM1_M0
7	UART6_TX	8	SPI0_MISO_M0/PWM2_M0
9	UART6_RX	10	ADC0
11	UART8_TX	12	ADC1
13	UART8_RX	14	ADC2

15	I2C5_SCL_M0/UART 9_TX		16	ADC3
17	I2C5_SDA_M0/UART 9_RX		18	ADC4
19	GND		20	ADC5

15.1 GPIO

主板扩展接口有四路GPIO，如下图所示：



序号	管脚标号	GPIO 标号	LIBGPIO节点
1	SPI0_CSN0_M0/PW M5_M0	GPIO0_C2	gpiochip0 18
2	SPI0_CLK_M0/PWM0 _M0	GPIO0_C3	gpiochip0 19

3	SPI0_MOSI_M0/PWM 1_M0	GPIO0_C4	gpiochip0 20
4	SPI0_MISO_M0/PWM 2_M0	GPIO0_C5	gpiochip0 21

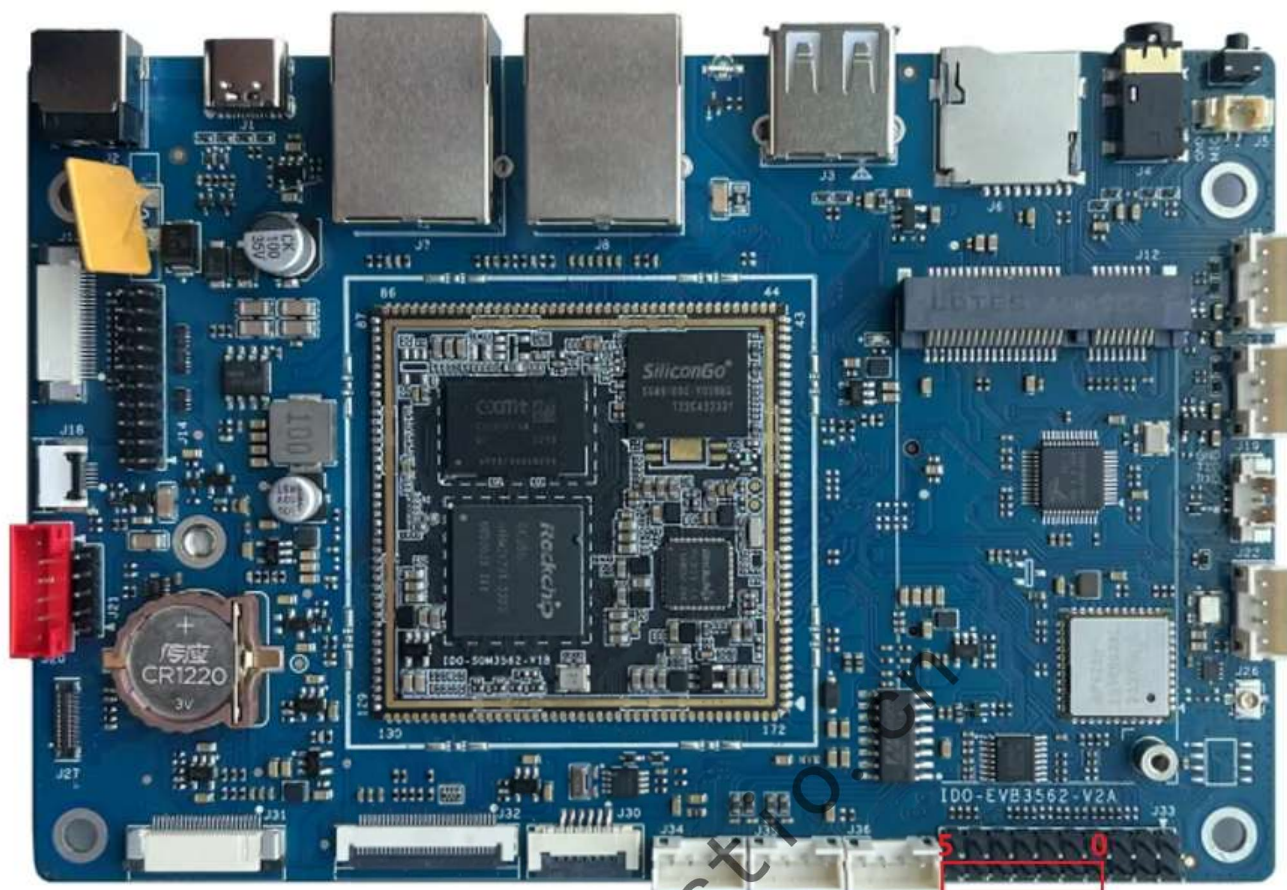
GPIO控制，根据上表【LIBGPIO节点】进行命令操作，以GPIO0_C2为例：

▼Shell

```
1  #输出：
2  #1为输出高电平(3.3V)，0为输出低电平(0V)。
3  root@ido:~# gpioset gpiochip0 18=1
4
5  root@ido:~# gpioset gpiochip0 18=0
6
7  #输入：
8  #0为外部输入低电平，1为外部输入高电平。
9  root@ido:~# gpioget gpiochip0 18
10 0
11 root@ido:~# gpioget gpiochip0 18
12 1
```

15.2 ADC

主板扩展接口有六路ADC，如下图所示：



扩展接口配置了6路ADC输入（参考电压为1.8v，支持10bit ADC采样）如下表所示：

名称	设备节点
ADCIN0	/sys/bus/iio/devices/iio:device1/in_voltage0_raw
ADCIN1	/sys/bus/iio/devices/iio:device1/in_voltage1_raw
ADCIN2	/sys/bus/iio/devices/iio:device1/in_voltage2_raw
ADCIN3	/sys/bus/iio/devices/iio:device1/in_voltage3_raw
ADCIN4	/sys/bus/iio/devices/iio:device1/in_voltage4_raw
ADCIN5	/sys/bus/iio/devices/iio:device1/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ADCIN5为例：

硬件连接：ADC5外接0V。

1.读取in_voltage5_raw的值，命令如下：

```
1 root@ido:/# cat /sys/bus/iio/devices/iio:device1/in_voltage5_raw
2 0
```

2.计算ADC5的输入电压V

$$V = (\text{voltage_raw}/1024) * 1.8V = (0/1024) * 1.8V = 0V$$

硬件连接：ADC5外接1.8V。

1.读取in_voltage5_raw的值，命令如下：

```
1 root@ido:~# cat /sys/bus/iio/devices/iio:device1/in_voltage5_raw
2 1020
```

2.计算ADC5的输入电压：

$$V = (\text{voltage_raw}/1024) * 1.8V = (1020/1024) * 1.8V = 1.79V$$

注意：ADC脚默认悬空状态，需要外部提供0~1.8v电压。

16 系统使用说明

16.1 QT5

16.1.1 3d测试

执行qt5 3d demo：

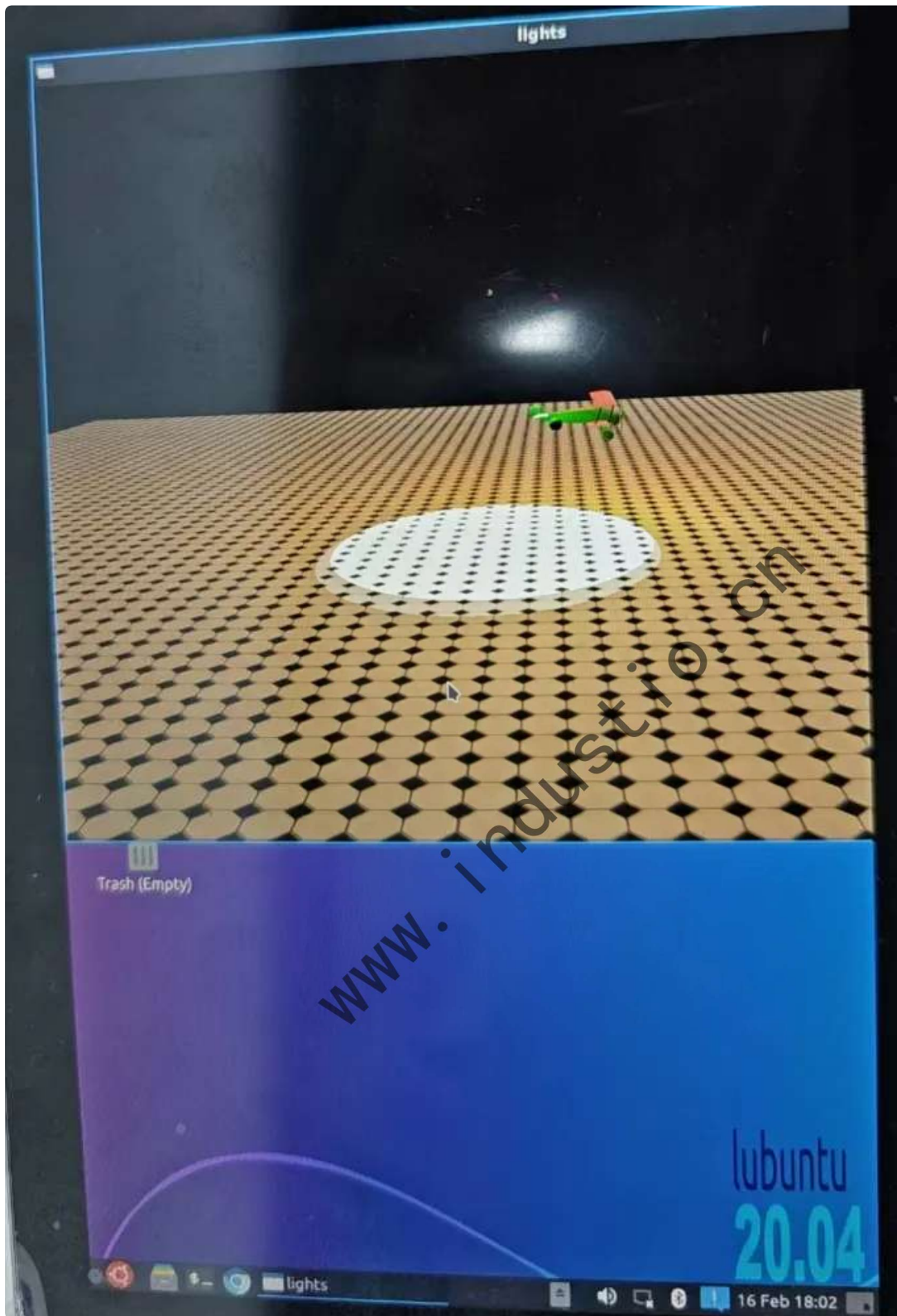
▼

Bash

```
1 root@ido:~# chmod a+x lights
2 root@ido:~#
3 root@ido:~# ./lights
4 arm_release_ver: g13p0-01eac0, rk_so_ver: 3
5 QOpenGLContext::swapBuffers() called with non-exposed window, behavior is undefined
```

如图所示：

www.industio.cn



16.2 glmark2

```

1  #设置cpu性能模式
2  root@ido:/# echo performance > /sys/devices/system/cpu/cpufreq/policy0/scaling_governor
3
4  #测试系统性能
5  root@ido:/# glmark2-es2
6  glmark2-es2          glmark2-es2-drm          glmark2-es2-wayland
7  root@ido:/# glmark2-es2
8  arm_release_ver: gl3p0-01eac0, rk_so_ver: 3
9  arm_release_ver: gl3p0-01eac0, rk_so_ver: 3
10 =====
11      glmark2 2021.02
12 =====
13      OpenGL Information
14      GL_VENDOR:      ARM
15      GL_RENDERER:    Mali-G52
16      GL_VERSION:     OpenGL ES 3.2 v1.gl3p0-01eac0.0fd2effaec483a5f4c440d2ffa25eb7a
17 =====
18 [build] use-vbo=false: FPS: 272 FrameTime: 3.676 ms
19 [build] use-vbo=true: FPS: 282 FrameTime: 3.546 ms
20 [texture] texture-filter=nearest: FPS: 306 FrameTime: 3.268 ms
21 [texture] texture-filter=linear: FPS: 306 FrameTime: 3.268 ms
22 [texture] texture-filter=mipmap: FPS: 306 FrameTime: 3.268 ms
23 [shading] shading=gouraud: FPS: 275 FrameTime: 3.636 ms
24 [shading] shading=blinn-phong-inf: FPS: 278 FrameTime: 3.597 ms
25 [shading] shading=phong: FPS: 266 FrameTime: 3.759 ms
26 [shading] shading=cel: FPS: 260 FrameTime: 3.846 ms
27 [bump] bump-render=high-poly: FPS: 195 FrameTime: 5.128 ms
28 [bump] bump-render=normals: FPS: 304 FrameTime: 3.289 ms
29 [bump] bump-render=height: FPS: 296 FrameTime: 3.378 ms
30 [effect2d] kernel=0,1,0;1,-4,1;0,1,0;: FPS: 240 FrameTime: 4.167 ms
31 [effect2d] kernel=1,1,1,1,1;1,1,1,1,1;1,1,1,1,1;: FPS: 146 FrameTime: 6.849 ms
32 [pulsar] light=false:quads=5:texture=false: FPS: 308 FrameTime: 3.247 ms
33 [desktop] blur-radius=5:effect=blur:passes=1:separable=true:windows=4: FPS: 135 FrameTime: 7.407 ms
34 [desktop] effect=shadow:windows=4: FPS: 262 FrameTime: 3.817 ms
35 [buffer] columns=200:interleave=false:update-dispersion=0.9:update-fraction=0.5:update-method=map: FPS: 127 FrameTime: 7.874 ms
36 [buffer] columns=200:interleave=false:update-dispersion=0.9:update-fraction=0.5:update-method=subdata: FPS: 128 FrameTime: 7.812 ms
37 [buffer] columns=200:interleave=true:update-dispersion=0.9:update-fraction=0.5:update-method=map: FPS: 162 FrameTime: 6.173 ms
38 [ideas] speed=duration: FPS: 232 FrameTime: 4.310 ms

```

```

39 [jellyfish] <default>: FPS: 226 FrameTime: 4.425 ms
40 [terrain] <default>: FPS: 39 FrameTime: 25.641 ms
41 [shadow] <default>: FPS: 188 FrameTime: 5.319 ms
42 [refract] <default>: FPS: 79 FrameTime: 12.658 ms
43 [conditionals] fragment-steps=0:vertex-steps=0: FPS: 304 FrameTime: 3.289
ms
44 [conditionals] fragment-steps=5:vertex-steps=0: FPS: 279 FrameTime: 3.584
ms
45 [conditionals] fragment-steps=0:vertex-steps=5: FPS: 304 FrameTime: 3.289
ms
46 [function] fragment-complexity=low:fragment-steps=5: FPS: 294 FrameTime: 3
.401 ms
47 [function] fragment-complexity=medium:fragment-steps=5: FPS: 264 FrameTim
e: 3.788 ms
48 [loop] fragment-loop=false:fragment-steps=5:vertex-steps=5: FPS: 294 Frame
Time: 3.401 ms
49 [loop] fragment-steps=5:fragment-uniform=false:vertex-steps=5: FPS: 294 Fr
ameTime: 3.401 ms
50 [loop] fragment-steps=5:fragment-uniform=true:vertex-steps=5: FPS: 277 Fra
meTime: 3.610 ms
51 =====
52                               glmark2 Score: 240
53 =====
54

```

如图所示：

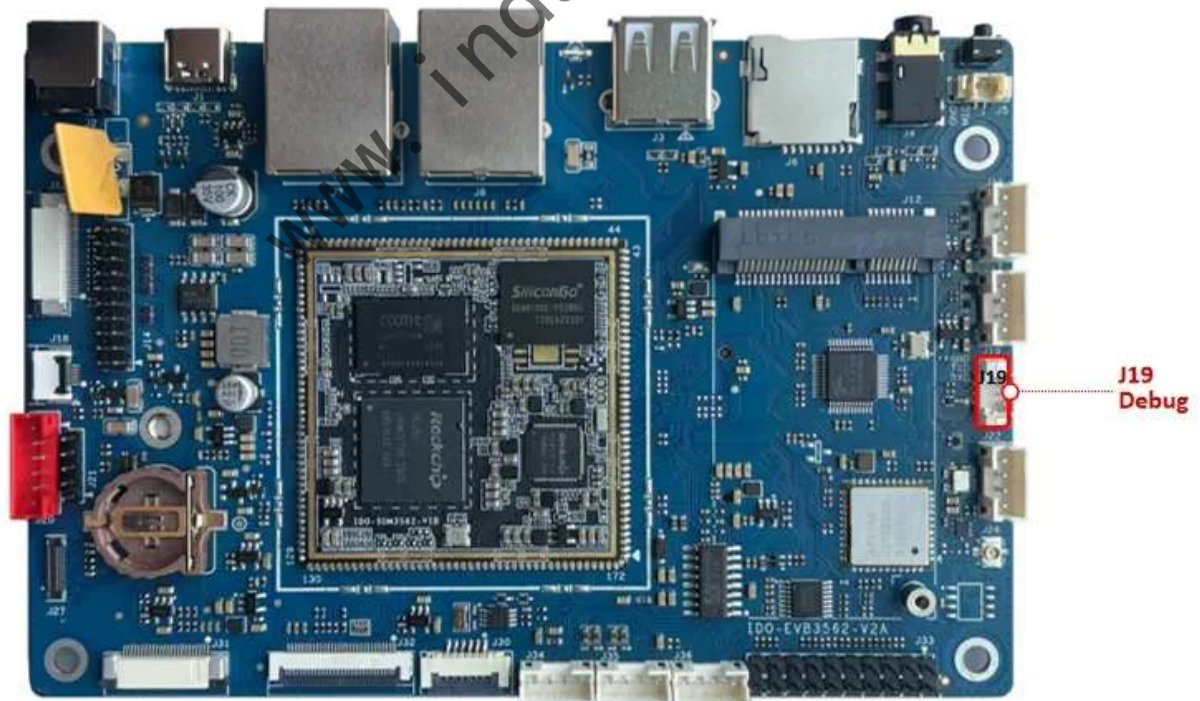


IDO-EVB3562-Debian 系统

1 调试

1.1 串口调试

主板调试串口位于J19，建议使用配套的usb转串口工具，如下图所示：



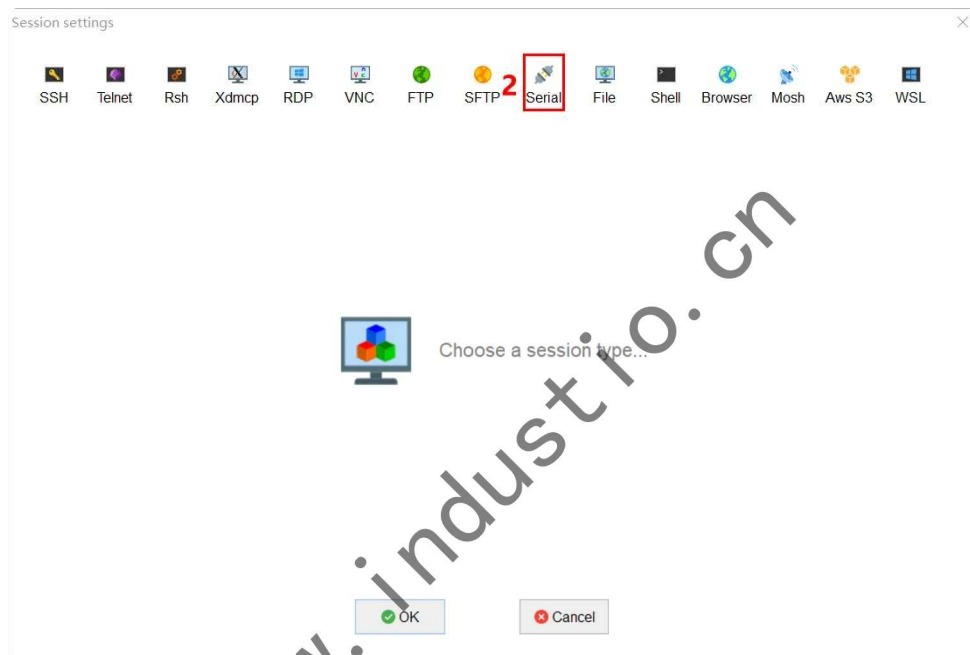
1. 打开MobaXterm，下载链接如下：

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

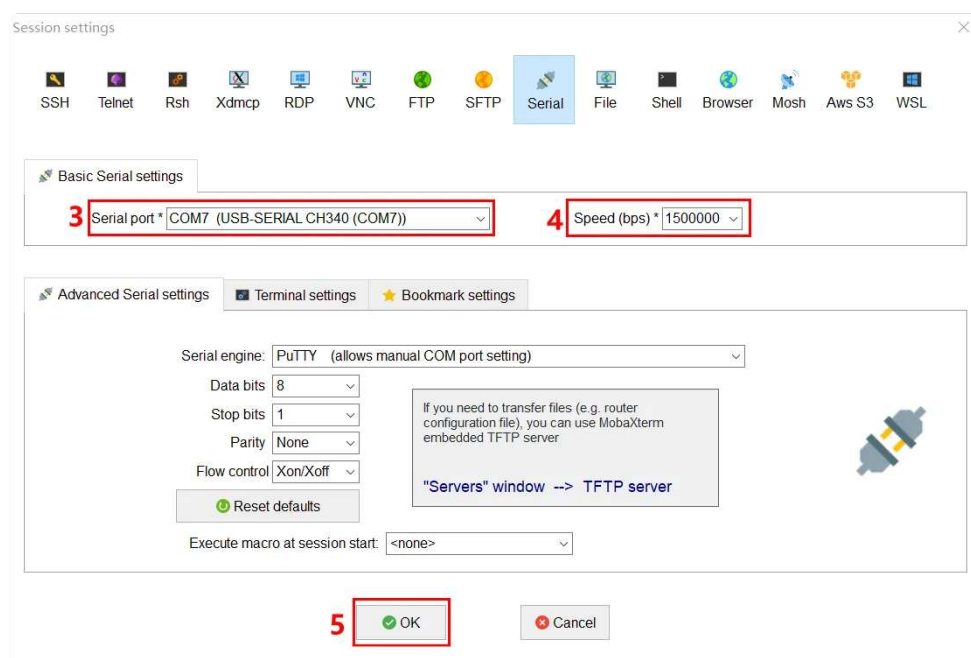
2. 选择sessionSerial，如下图所示：



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：

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

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J1，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:\Users\industio_mhk>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：root
2. 密码：123456
3. 端口：22

ssh调试，如下图所示：

```
[C:\~]$ ssh root@192.168.0.131
```

```
Connecting to 192.168.0.131:22...
```

```
Connection established.
```

```
To escape to local shell, press 'Ctrl+Alt+]'.
```

```
Linux linaro-alip 5.10.160 #5 SMP Mon Dec 23 10:34:20 CST 2024 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.
```

```
Last login: Tue Dec 24 15:05:47 2024 from 192.168.0.15
```

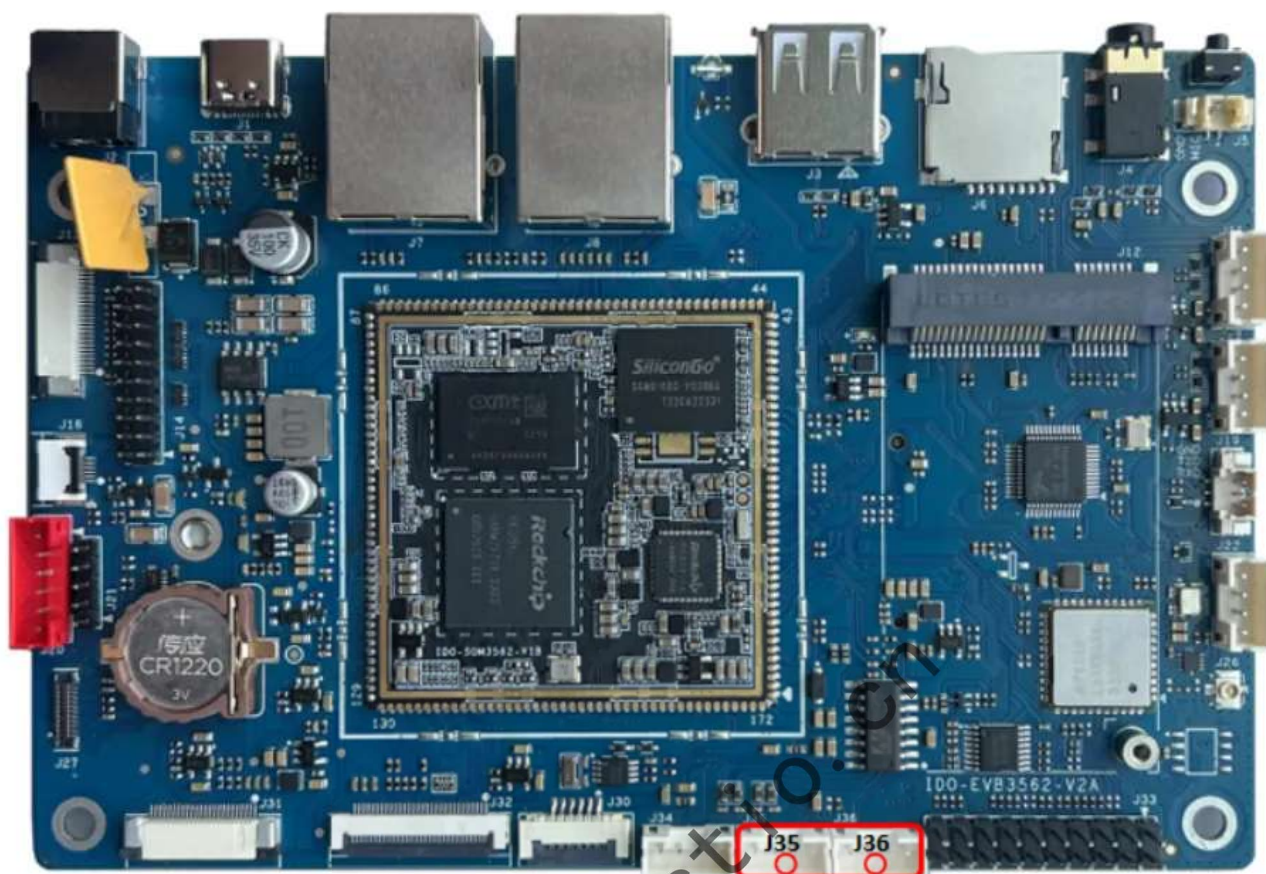
```
root@linaro-alip:~#
```

2 串口

主板共配置4路串口（除调试串口），如下图所示：

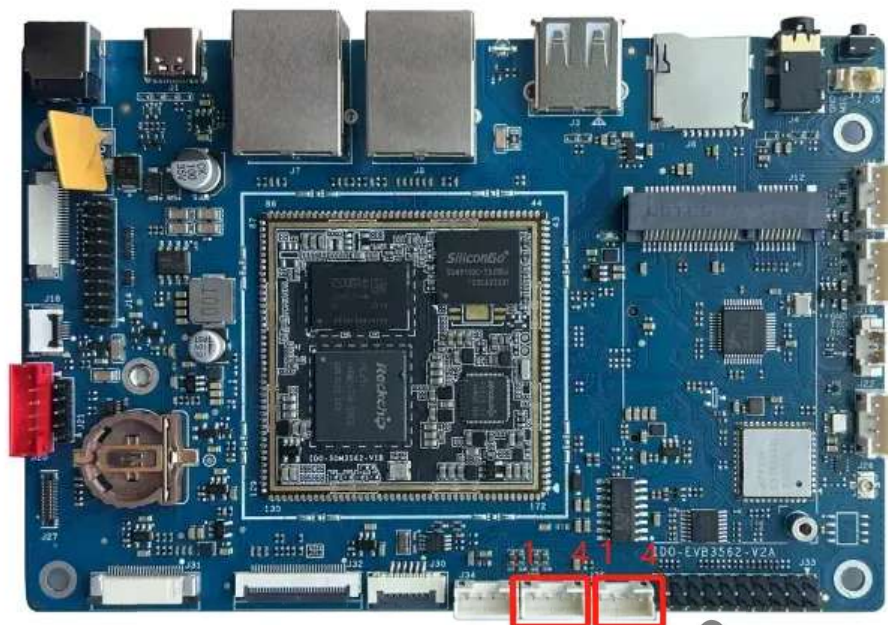
2.1 RS232

www.industio.cn



**2x RS232
J35~J36**

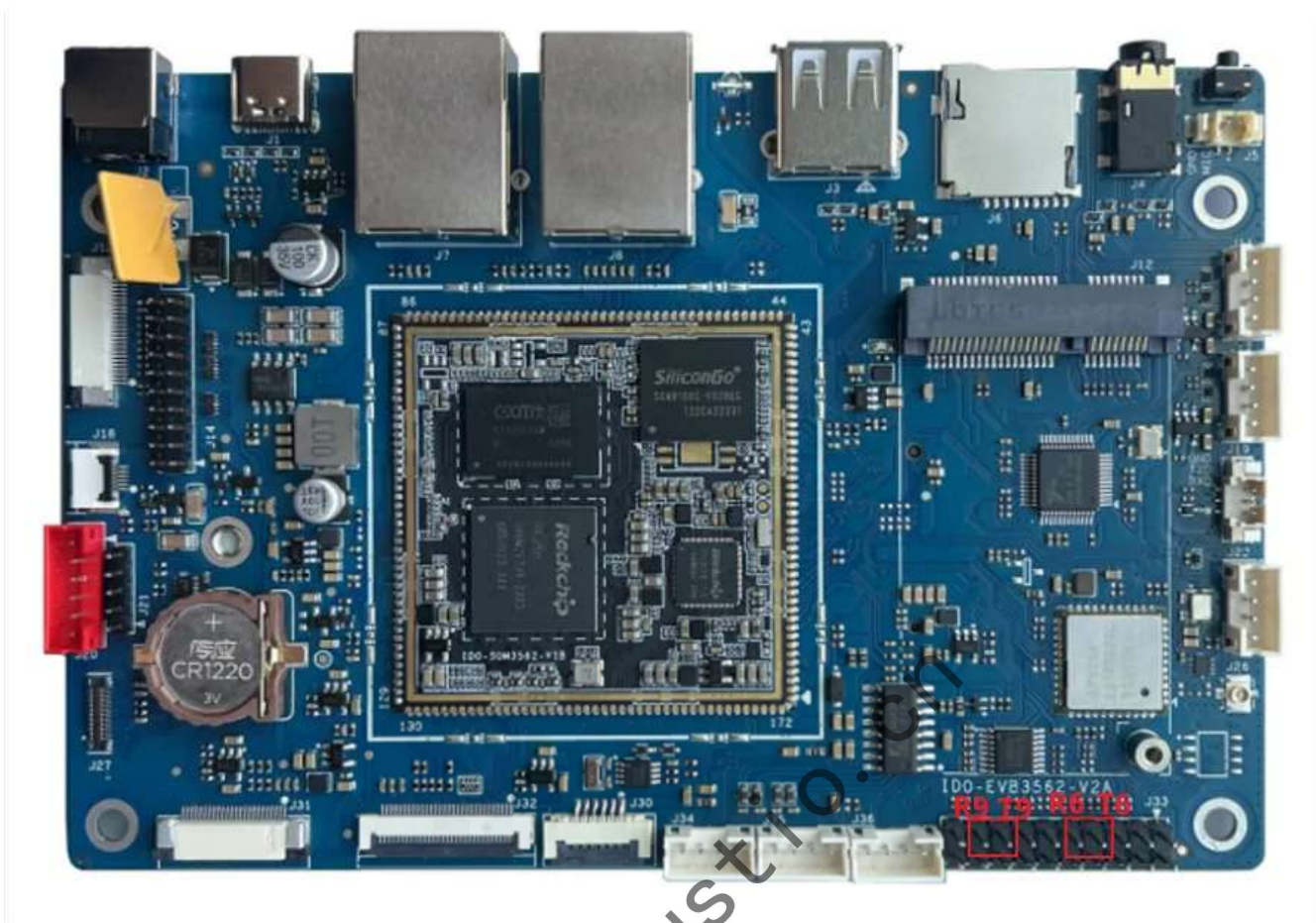
丝印	电平类型	设备节点	说明
J35	RS232	/dev/ttyS8	/
J36	RS232	/dev/ttyS3	/



引脚定义:

序号	定义	电平/V	说明
1	VCC	5V/3.3V	默认5V电源输出，可配置为3.3V
2	RX	/	RS232-接收
3	TX	/	RS232-发送
4	GND	GND	电源地

2.2 TTL



设备节点，如下表所示：

序号	电平类型	设备节点	说明
1	TTL	/dev/ttyS9	/
2	TTL	/dev/ttyS6	默认配置CAN0，此接口无效

使用microcom可以进行串口收发测试，命令如下：

▼Shell

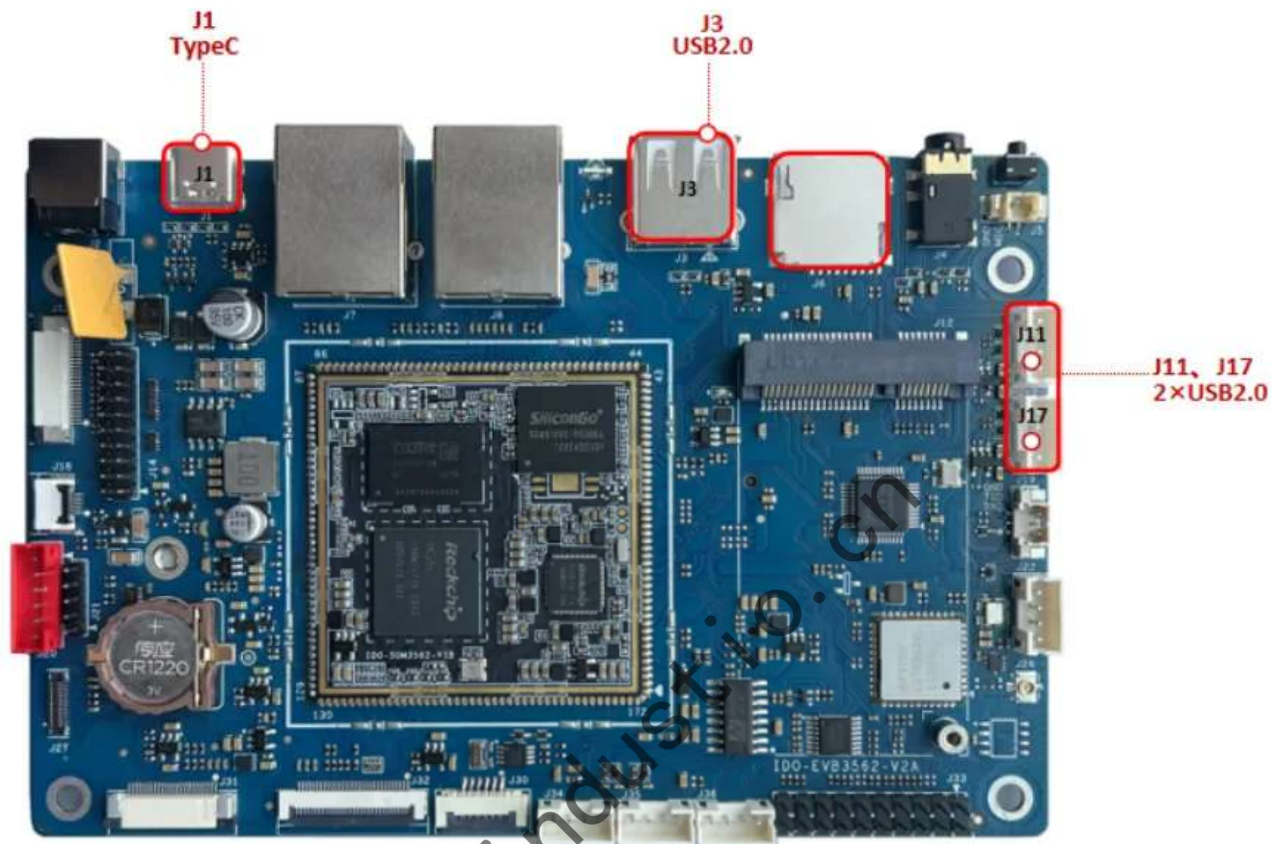
```
1 root@linaro-alip:/# microcom -s 115200 -p /dev/ttyS3
```

注意：

- 1.测试完成，按Ctrl+x退出。
- 2./dev/ttyS6 默认不通，此功能需要修改软硬件。

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	J1	OTG
USB2	J3	USB2.0 HOST
USB3	J11	USB2.0 HOST
USB4	J17	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 /dev/mmcblk2p8 on / type ext4 (rw,relatime)
3 ...
4 /dev/sda1 on /mnt/udisk/USBSTORAGE type vfat (rw,relatime,uid=1000,gid=1000
,fmask=0022,dmask=0022,codepage=936,iocharset=utf8,shortname=mixed,errors=r
emount-ro)
```

USB电源控制，如下表所示：

USB端口	默认状态	动作	命令
USB2	开启	关闭电源	echo 0 > /sys/class/leds/usb_host2/brightness
		开启电源	echo 1 > /sys/class/leds/usb_host2/brightness
USB3	开启	关闭电源	echo 0 > /sys/class/leds/usb_host3/brightness
		开启电源	echo 1 > /sys/class/leds/usb_host3/brightness
USB4	开启	关闭电源	echo 0 > /sys/class/leds/usb_host4/brightness
		开启电源	echo 1 > /sys/class/leds/usb_host4/brightness

供电控制说明：设备节点写"0"关闭电源，写"1"开启电源。

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，如下所示：

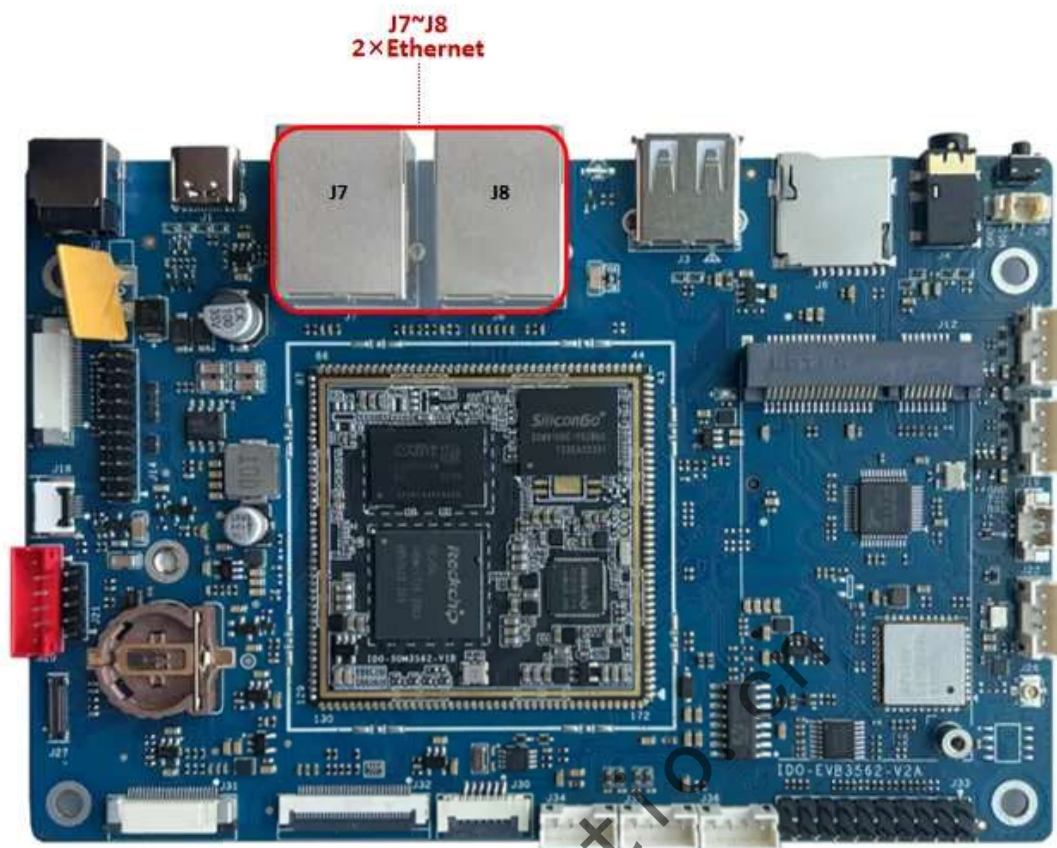
```

1  root@linaro-alip:~# fdisk -l
2  Disk /dev/ram0: 4 MiB, 4194304 bytes, 8192 sectors
3  Units: sectors of 1 * 512 = 512 bytes
4  Sector size (logical/physical): 512 bytes / 4096 bytes
5  I/O size (minimum/optimal): 4096 bytes / 4096 bytes
6  ...
7  Device          Boot Start      End  Sectors  Size Id Type
8  /dev/mmcbk1p1    8192 30449663 30441472 14.5G  c W95 FAT32 (LBA)
9
10 root@linaro-alip:~# mount
11 /dev/mmcbk0p8 on / type ext4 (rw,relatime)
12 devtmpfs on /dev type devtmpfs (rw,relatime,size=1988328k,nr_inodes=497082
13 ,mode=755)
14 sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)
15 proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
16 ...
17 /dev/mmcbk1p1 on /mnt/sdcard type vfat (rw,relatime,async,uid=1000,gid=100
18 0,fmask=0022,dmask=0022,codepage=936,iocharset=utf8,shortname=mixed,errors
19 =remount-ro)

```

5 Ethernet

主板共配置2路千兆网接口，如下图所示：



序号	接口位置	速率	网络节点
1	J7	千兆	eth0
2	J8	千兆	eth1

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 192.168.3.200 netmask 255.255.255.0 up
2 root@linaro-alip:~# route add default gw 192.168.3.1
3 root@linaro-alip:~# ping 8.8.8.8
4 PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
5 64 bytes from 8.8.8.8: icmp_seq=1 ttl=113 time=12.7 ms
6 64 bytes from 8.8.8.8: icmp_seq=2 ttl=113 time=14.1 ms

```

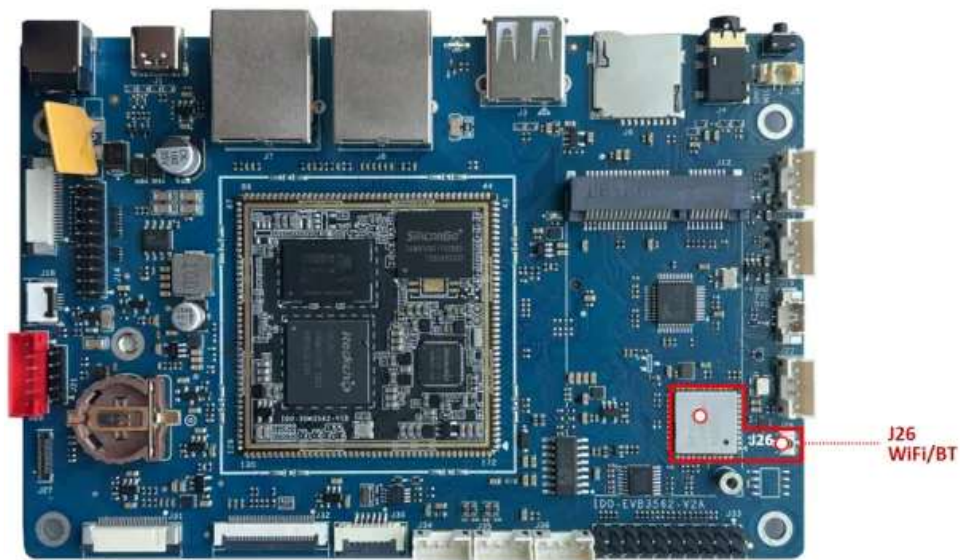
如需设置永久静态IP，则可通过添加/etc/netplan/01-netcfg.yaml，命令如下：

```
1 root@linaro-alip:~# touch /etc/netplan/01-netcfg.yaml
2 root@linaro-alip:~# vim /etc/netplan/01-netcfg.yaml
3 root@linaro-alip:~# cat /etc/netplan/01-netcfg.yaml
4 network:
5     version: 2
6     ethernet:
7         eth0:
8             dhcp4: no
9             addresses: [192.168.3.121/24]
10            nameservers:
11                addresses: [192.168.3.1, 8.8.8.8, 114.114.1
12                    14.114]
13            routes:
14                - to: 0.0.0.0/0
15                  via: 192.168.3.1
16                  metric: 100
17 #开启netplan服务
18 root@linaro-alip:~# systemctl enable systemd-networkd.service
19 Created symlink /etc/systemd/system/dbus-org.freedesktop.network1.service
20 → /lib/systemd/system/systemd-networkd.service.
21 Created symlink /etc/systemd/system/multi-user.target.wants/systemd-networ
22 kd.service → /lib/systemd/system/systemd-networkd.service.
23 Created symlink /etc/systemd/system/sockets.target.wants/systemd-networkd.
24 socket → /lib/systemd/system/systemd-networkd.socket.
25 Unit /etc/systemd/system/systemd-networkd-wait-online.service is masked, i
26 gnoring.
27 root@linaro-alip:~# systemctl start systemd-networkd.service
28 #立即生效
29 root@linaro-alip:~# netplan apply
```

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，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：

```
1 root@linaro-alip:/# ifconfig wlan0
2 wlan0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
3         ether c0:f5:35:12:c5:f8 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

命令如下：

```

1  #打开WiFi
2  root@linaro-alip:/# nmcli r wifi on
3  #扫描附近热点
4  root@linaro-alip:/# nmcli dev wifi
5  IN-USE  BSSID          SSID          MODE  CHAN  RATE
SIGNAL  BARS  SECURITY
6          22:A3:8E:D0:51:42  HUAWEI-1EX2DK  Infra  1     130 Mbit/s
100      ██████  --
7          94:D9:B3:A6:B8:7A  TP-LINK_B87A   Infra  7     270 Mbit/s
100      ██████  WPA1 WPA2
8          22:A3:8E:D0:51:46  HUAWEI-1EX2DK  Infra  44    270 Mbit/s
100      ██████  --
9          94:D9:B3:A6:B8:7F  TP-LINK_5G_B87A  Infra  157   270 Mbit/s
100      ██████  WPA1 WPA2
10         78:60:5B:83:67:AE  test_2.4G      Infra  6     540 Mbit/s
92       ██████  WPA1 WPA2
11         70:C6:DD:3B:2F:C4  Industio_2.4   Infra  11    130 Mbit/s
92       ██████  WPA1 WPA2
12         70:C6:DD:3B:2F:C7  Industio_5.8   Infra  153   270 Mbit/s
92       ██████  WPA1 WPA2
13         7A:60:5B:13:67:AE  TPGuest_67AE   Infra  6     540 Mbit/s
90       ██████  --
14         7A:60:5B:73:67:AE  --             Infra  6     540 Mbit/s
90       ██████  WPA1 WPA2
15         9A:2C:0D:B3:E0:07  jjh            Infra  11    130 Mbit/s
84       ██████  WPA2
16         7A:60:5B:F3:67:AE  --             Infra  40    540 Mbit/s
82       ██████  WPA1 WPA2
17         44:F7:70:38:A7:18  CS_5G          Infra  44    270 Mbit/s
82       ██████  WPA2
18         44:F7:70:38:A7:19  CS             Infra  1     270 Mbit/s
80       ██████  WPA3
19         B8:A1:4A:0A:F1:43  ChinaNet-MvDm  Infra  4     130 Mbit/s
80       ██████  WPA1 WPA2
20         78:60:5B:83:67:B0  test_5G        Infra  40    540 Mbit/s
80       ██████  WPA1 WPA2
21         4E:F7:70:38:A7:18  --             Infra  44    270 Mbit/s
80       ██████  WPA2
22         F6:6D:2F:E0:46:03  --             Infra  11    270 Mbit/s
67       ██████  WPA1 WPA2
23         1C:D5:E2:D6:0D:FA  ChinaNet-fsnE  Infra  1     130 Mbit/s
62       ██████  WPA1 WPA2
24         24:CF:24:50:7F:A6  Xiaomi_7FA5     Infra  10    130 Mbit/s
60       ██████  WPA1 WPA2
25

```

```

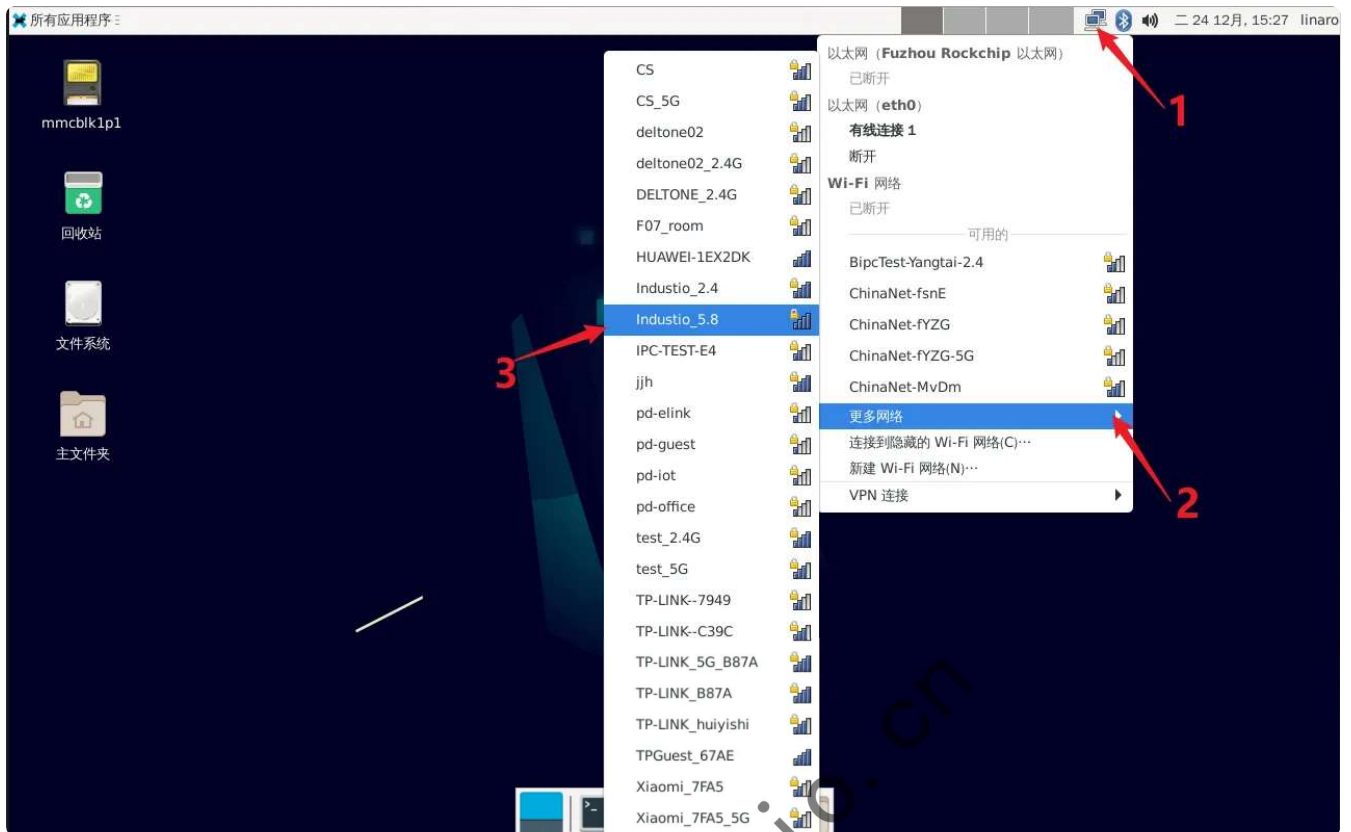
26      57      68:77:24:4B:C3:9C TP-LINK--C39C Infra 6 270 Mbit/s
27      55      24:CF:24:50:7F:A7 Xiaomi_7FA5_5G Infra 157 270 Mbit/s
28      52      F4:6D:2F:D0:46:03 TP-LINK_huiyishi Infra 11 270 Mbit/s
29      50      76:39:89:63:90:4C -- Infra 6 270 Mbit/s
30      49      A0:10:77:22:60:A0 ChinaNet-fYZG Infra 1 130 Mbit/s
31      47      74:39:89:63:90:4C deltone02_2.4G Infra 6 270 Mbit/s
32      45      3C:BC:D0:2E:5A:D4 ChinaNet-fYZG Infra 1 130 Mbit/s
33      #连接热点
34      root@linaro-alip:/# nmcli dev wifi connect Industio_5.8 password industio6
35      66 ifname wlan0
36      成功用 "wlan0568b5c10-d4d5-445a-824f-5d640690fc83" 激活了设备 ""。
37      root@linaro-alip:/# ifconfig wlan0
38      wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
39      inet 192.168.0.90 netmask 255.255.255.0 broadcast 192.168.0.255
40      inet6 fe80::b352:13af:f613:b532 prefixlen 64 scopeid 0x20<link>
41      ether c0:f5:35:12:c5:f8 txqueuelen 1000 (Ethernet)
42      RX packets 69 bytes 6416 (6.2 KiB)
43      RX errors 0 dropped 3 overruns 0 frame 0
      TX packets 15 bytes 2077 (2.0 KiB)

```

其中"Industio_5.8"为热点名称，"industio666"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：

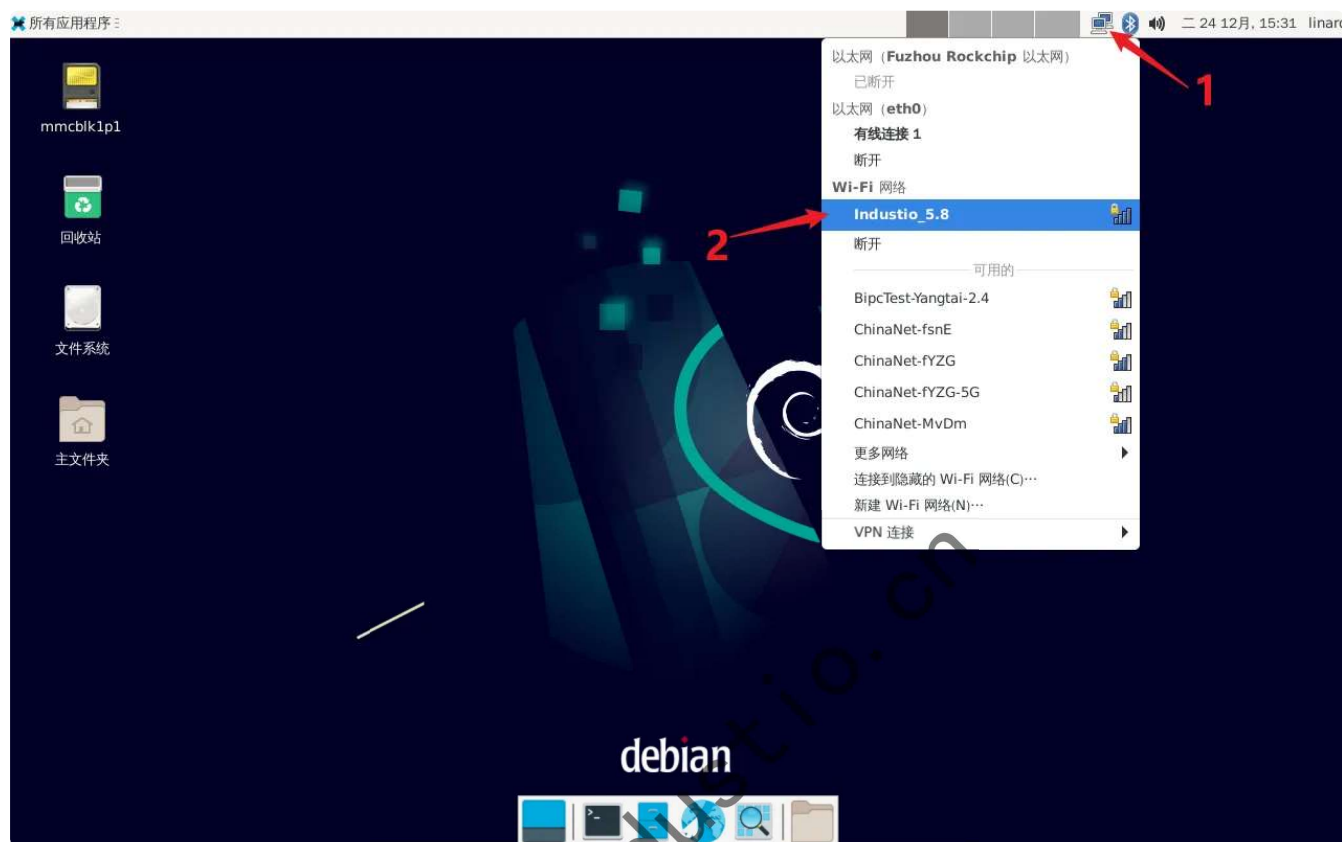


弹出密码输入框，使用键盘输入密码，如下图所示：



输入密码后，点击【连接】按钮连接热点。

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：

```
root@linaro-alip:/# ifconfig wlan0
wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.0.90 netmask 255.255.255.0 broadcast 192.168.0.255
    inet6 fe80::ffcc:fe37:fa1c:c336 prefixlen 64 scopeid 0x20<link>
    ether c0:f5:35:12:c5:f8 txqueuelen 1000 (Ethernet)
    RX packets 894 bytes 66801 (65.2 KiB)
    RX errors 0 dropped 26 overruns 0 frame 0
    TX packets 22 bytes 3330 (3.2 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

7 Buletooth

7.1 开启蓝牙

开启蓝牙，命令如下：

```
1 root@linaro-alip:/# hciconfig -a
2 hci0: Type: Primary Bus: UART
3   BD Address: 4A:F1:56:1D:BB:CF ACL MTU: 1021:8 SCO MTU: 64:1
4   UP RUNNING
5   RX bytes:927 acl:0 sco:0 events:69 errors:0
6   TX bytes:5004 acl:0 sco:0 commands:69 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: 0x3c0000
13  Service Classes: Rendering, Capturing, Object Transfer, Audio
14  Device Class: Miscellaneous,
15  HCI Version: 5.0 (0x9) Revision: 0x50
16  LMP Version: 5.0 (0x9) Subversion: 0x6606
17  Manufacturer: Broadcom Corporation (15)
```

7.2 扫描设备

扫描设备，命令如下：

```
1 root@linaro-alip:/# bluetoothctl
2 Agent registered
3 [bluetooth]# power on
4 Changing power on succeeded
5 [bluetooth]# scan on
```

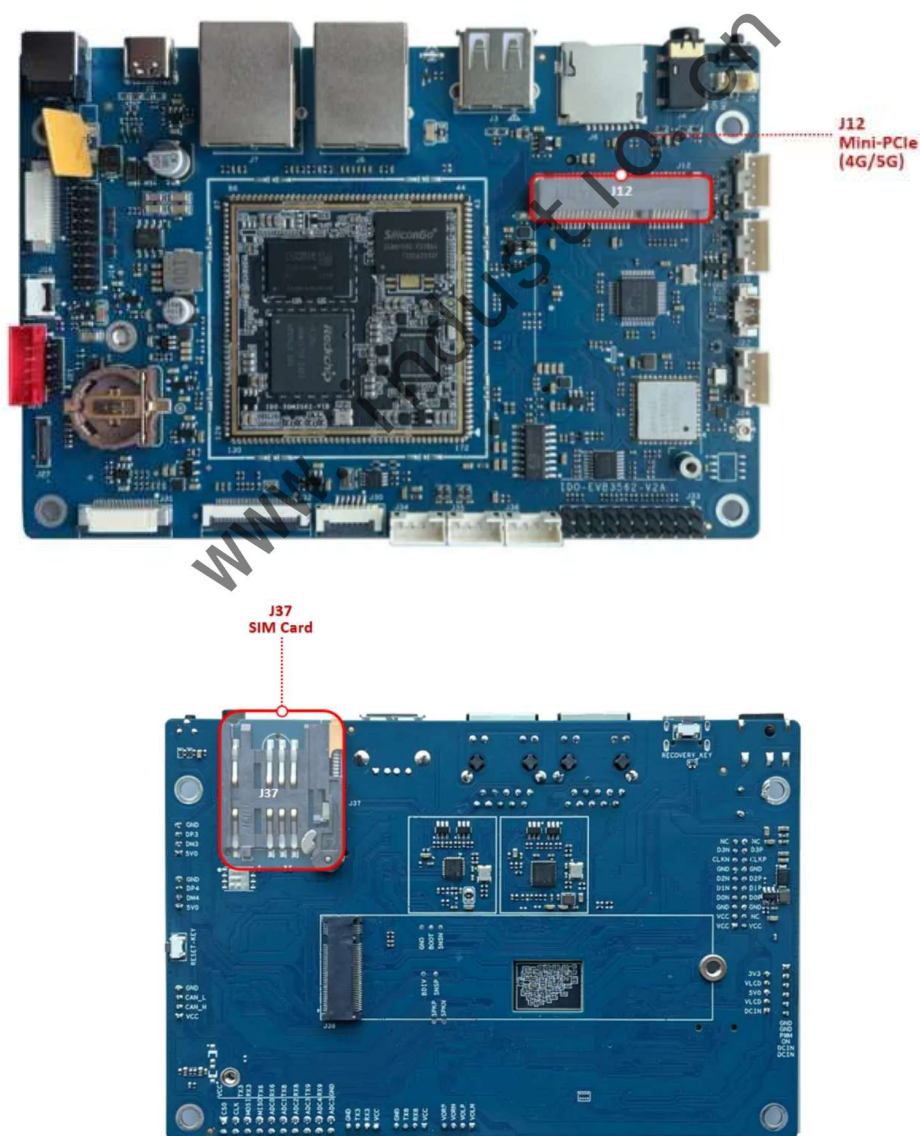
7.3 连接蓝牙设备

连接蓝牙设备，命令如下：

```
1 root@linaro-alip:/# bluetoothctl
2 [bluetooth]# devices
3 ...
4 [bluetooth]# trust 70:55:BB:0B:E0:AA
5 [bluetooth]# pair 70:55:BB:0B:E0:AA
6 [bluetooth]# connect 70:55:BB:0B:E0:AA
7 [A17]#
```

8 4G

主板配置了一路4G接口，默认适配EC20模块，如下图所示：



开机自动拨号，拨号成功后，产生wwan0网络节点：

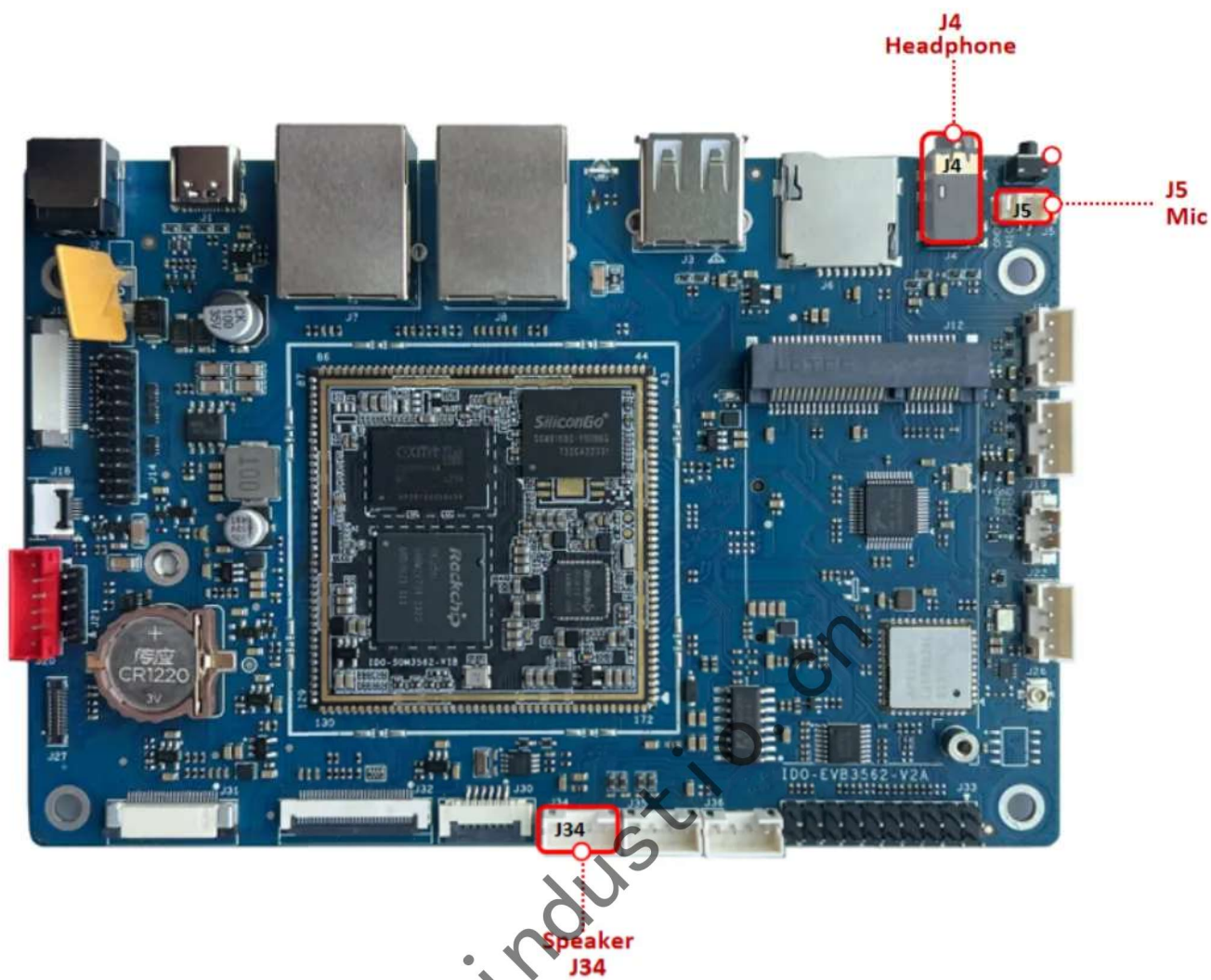
```
▼ Shell |
1 root@linaro-alip:/# ifconfig wwan0
2 wwan0      Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
3           inet addr:10.122.45.92  P-t-P:10.122.45.92  Mask:255.255.255.248
4           UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
5           RX 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：

```
▼ Shell |
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) 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
```

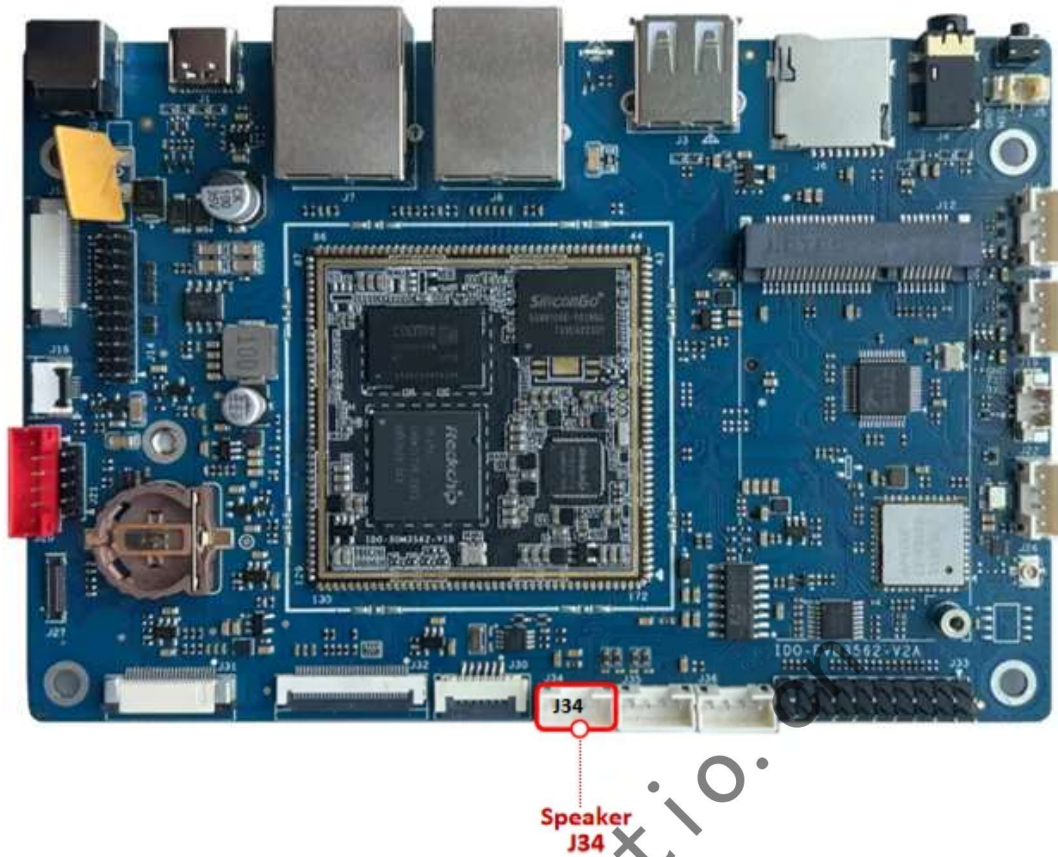
9 音频

主板音频接口如下所示：



9.1 喇叭

双声道Lineout接口位于J34，如下图所示：

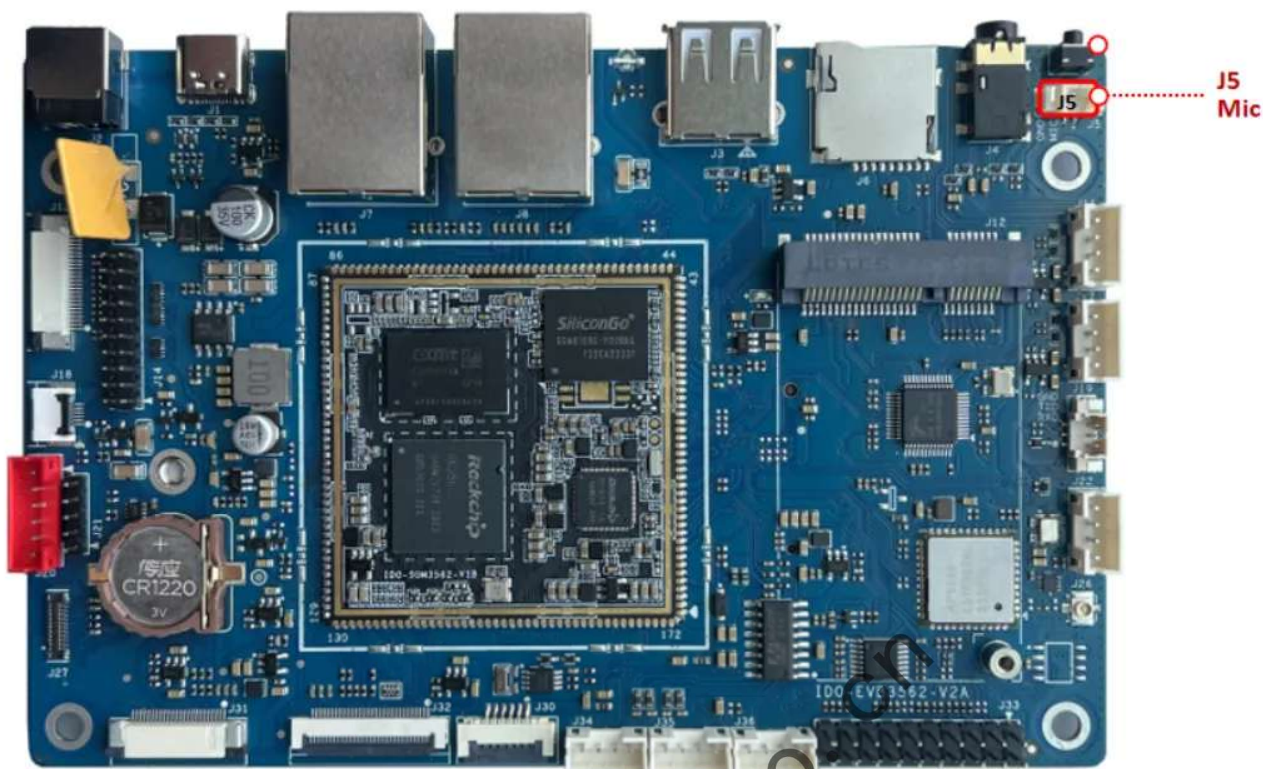


播放音频，命令如下：

```
1 #打开spk switch
2 root@linaro-alip:/# amixer cset numid=37 2 -c 1
3 #打开mic switch
4 root@linaro-alip:/# amixer cset numid=38 1 -c 1
5
6 root@linaro-alip:/# aplay -D hw:1,0 ./xinhuan.wav
```

9.2 MIC

MIC如下图所示：

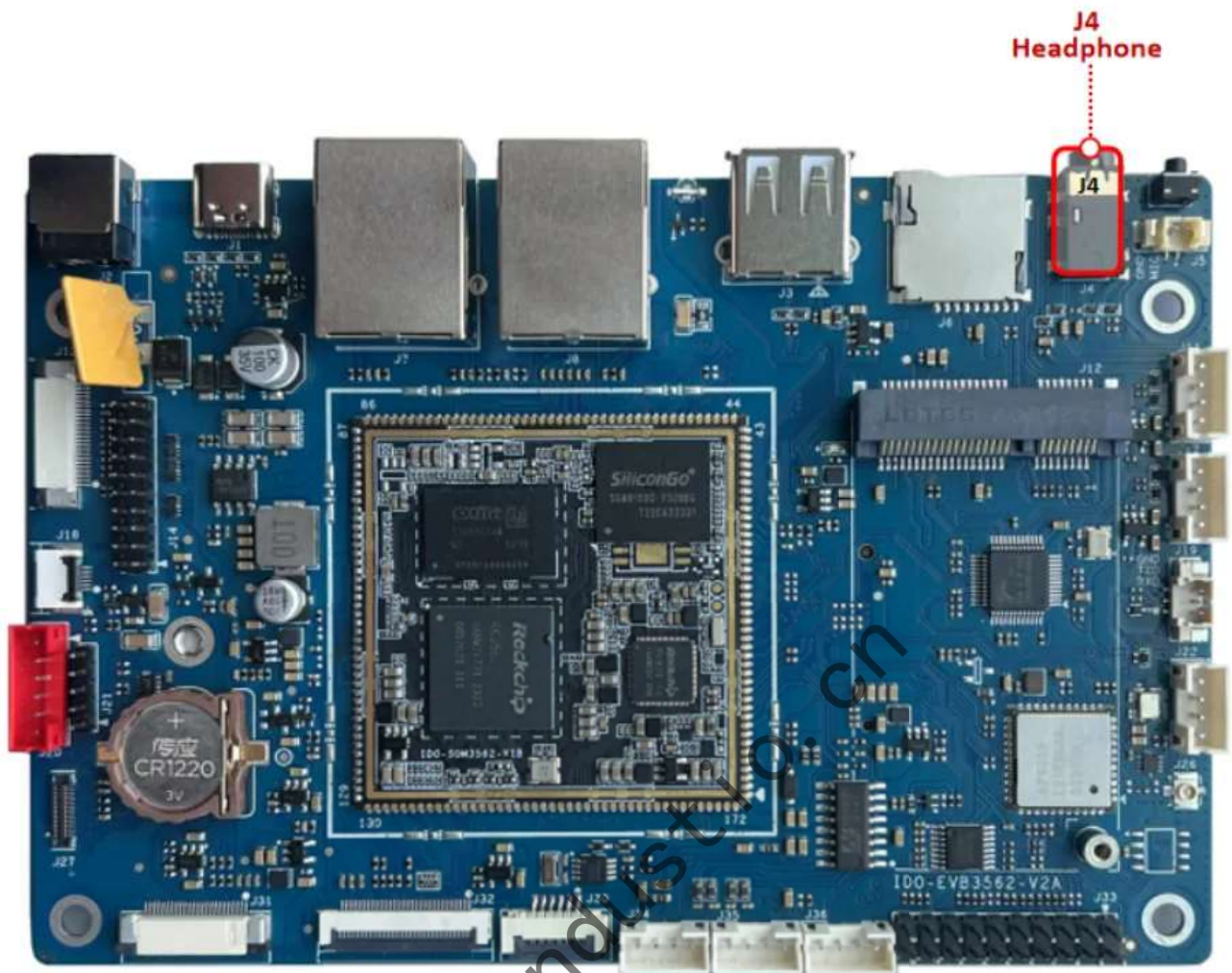


录音，命令如下：

```
1 root@linaro-alip:/# arecord -D hw:1,0 -r 48000 -c 2 -f S16_LE test.wav
```

9.3 耳机

耳机如下图所示：

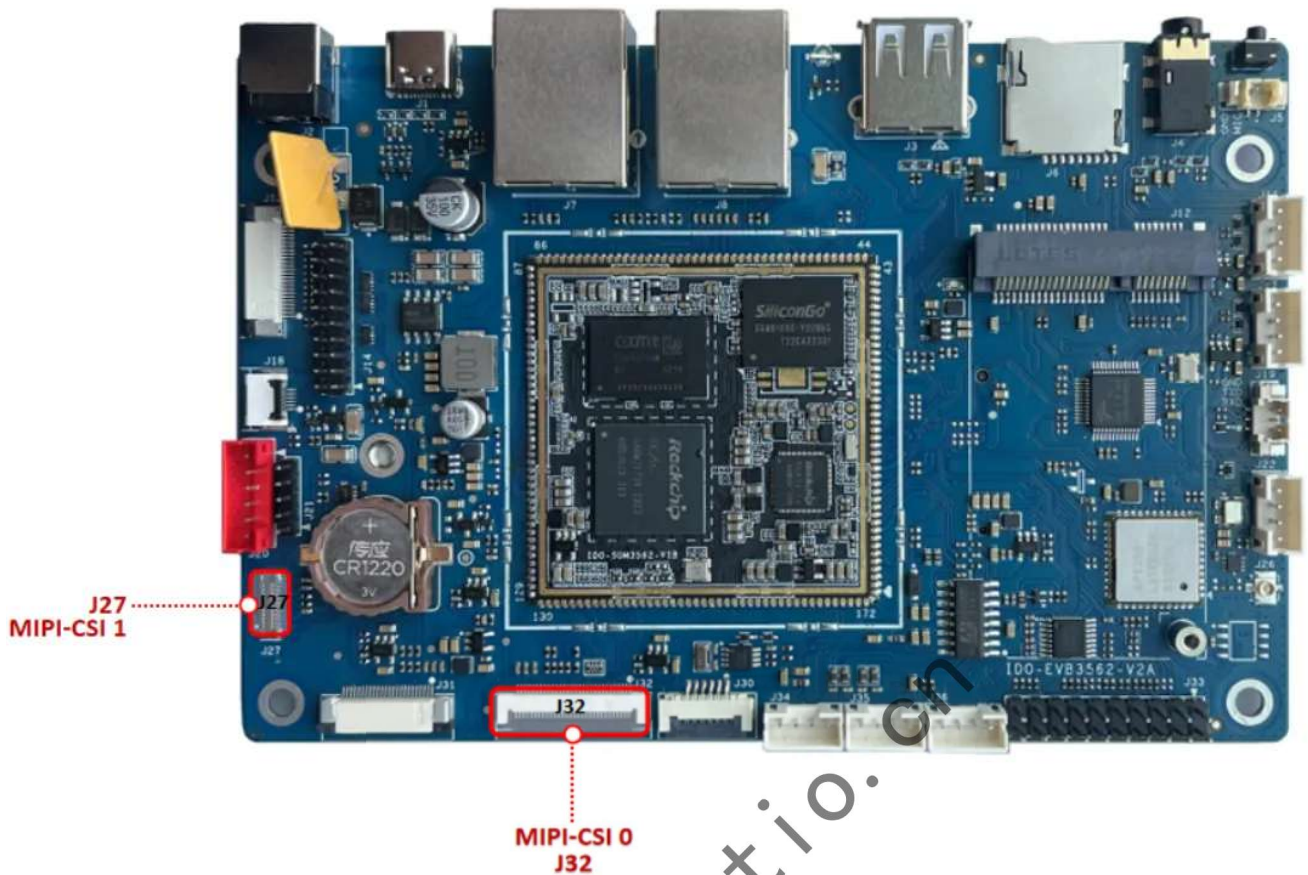


播放音频，命令如下：

```
1 root@linaro-alip:/# aplay -D hw:1,0 ./xinhuan.wav
```

10 Camera

主板配置2路摄像头，型号为ov13855（J27）和ov13855（J32），如下所示：



10.1 ov13855(J32)

预览

▼ Shell

```
1 root@linaro-alip:/# export DISPLAY=:0
2 root@linaro-alip:/# gst-launch-1.0 v4l2src device=/dev/video22 ! video/x-raw, format=NV12, width=1920, height=1080, framerate=30/1 ! videoconvert ! autovideosink sync=false
```

抓图

▼ Shell

```
1 root@linaro-alip:/# export DISPLAY=:0
2 root@linaro-alip:/# v4l2-ctl -d /dev/video22 --set-fmt-video=width=1920,height=1080,pixelformat=NV12 --stream-mmap=4 --stream-to=/tmp/test_vid22_nv12.yuv --stream-count=1 --stream-skip=30
```

10.2 ov13855(J27)

预览

▼ Shell

```
1 root@linaro-alip:/# export DISPLAY=:0
2 root@linaro-alip:/# gst-launch-1.0 v4l2src device=/dev/video29 ! video/x-raw, format=Nv12, width=1920, height=1080, framerate=30/1 ! videoconvert ! autovideosink sync=false
```

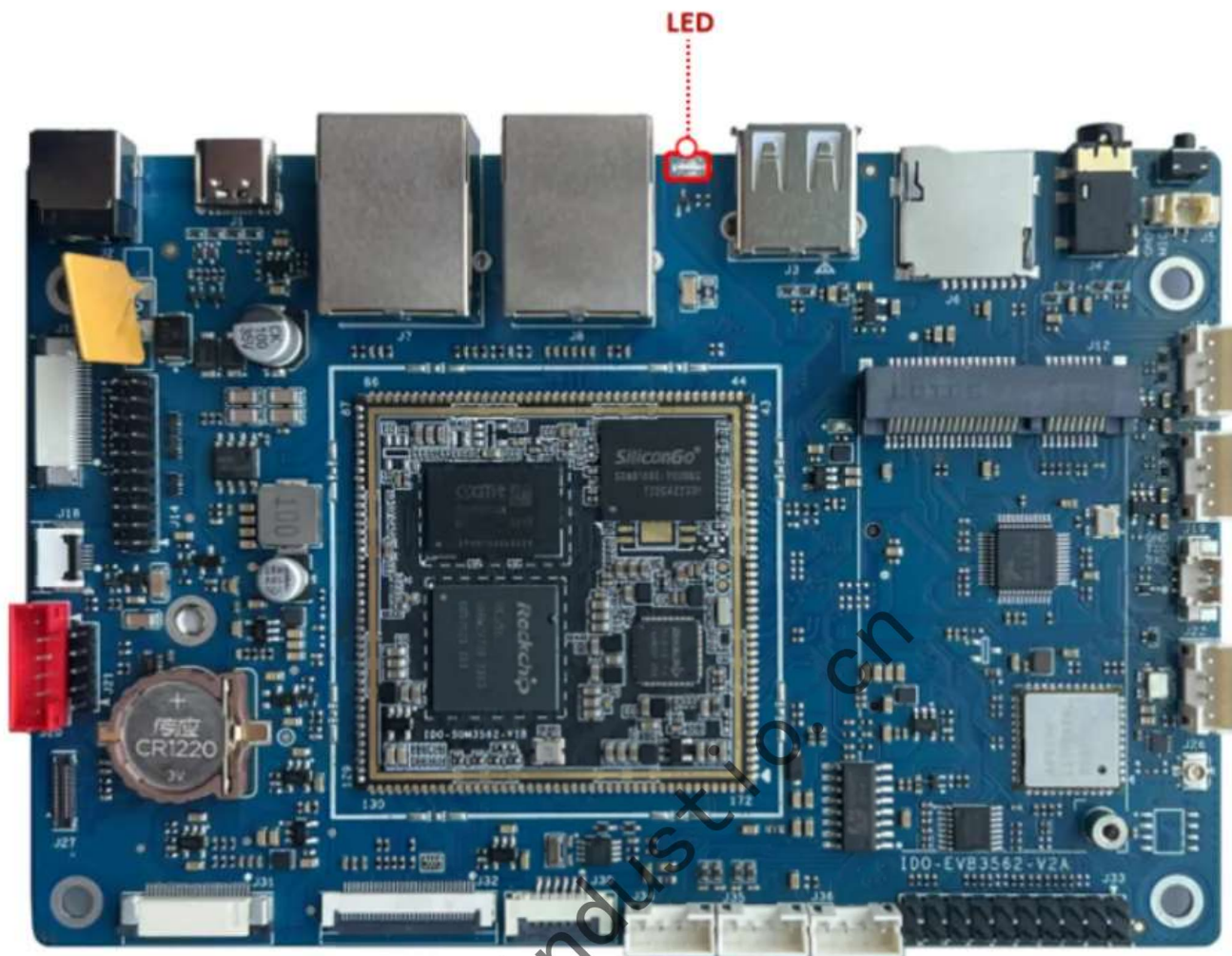
抓图

▼ Shell

```
1 root@linaro-alip:/# export DISPLAY=:0
2 root@linaro-alip:/# v4l2-ctl -d /dev/video29 --set-fmt-video=width=1920,height=1080,pixelformat=Nv12 --stream-mmap=4 --stream-to=/tmp/test_vide22_nv12.yuv --stream-count=1 --stream-skip=30
```

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，系统中对应的设备节点为/dev/rtc0，如下图所示：



12.1 读取时间

读取RTC时间，命令如下：

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

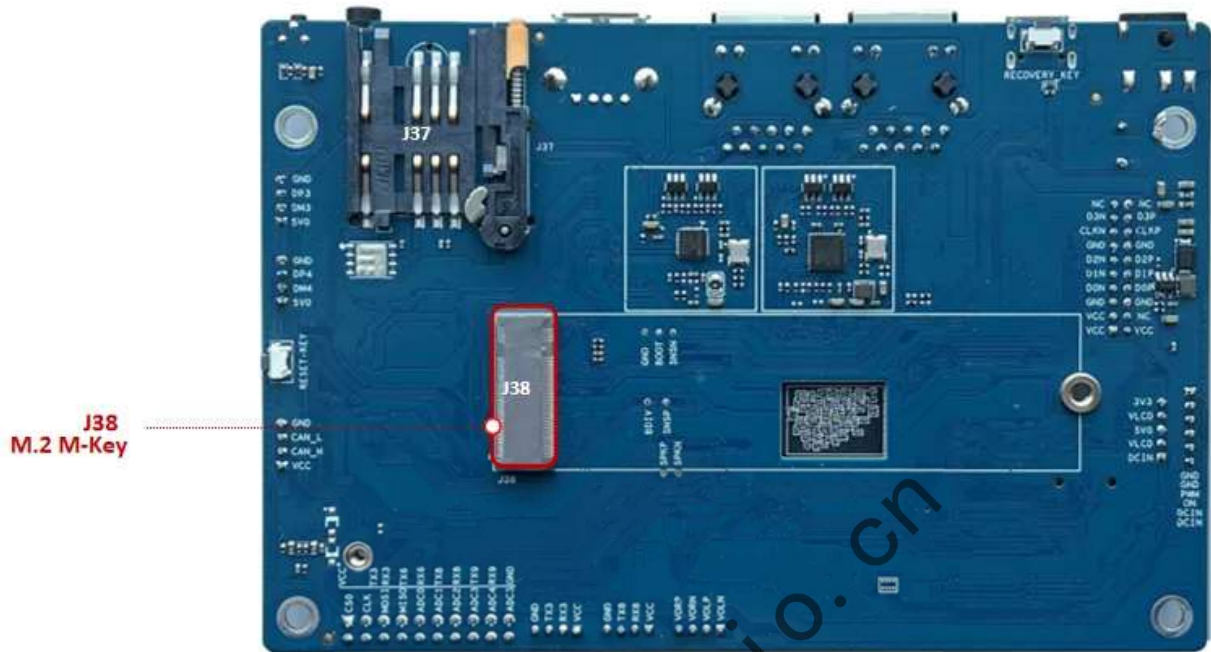
12.2 设置时间

设置RTC时间，命令如下：

```
1 root@linaro-alip:/# date -s '2024-8-22 15:30:00'
2 Tue Aug 22 15:30:00 UTC 2024
3 root@linaro-alip:/# hwclock -w
4 root@linaro-alip:/# hwclock -r
5 Tue Aug 22 15:30:05 2024 0.000000 seconds
```

13 硬盘

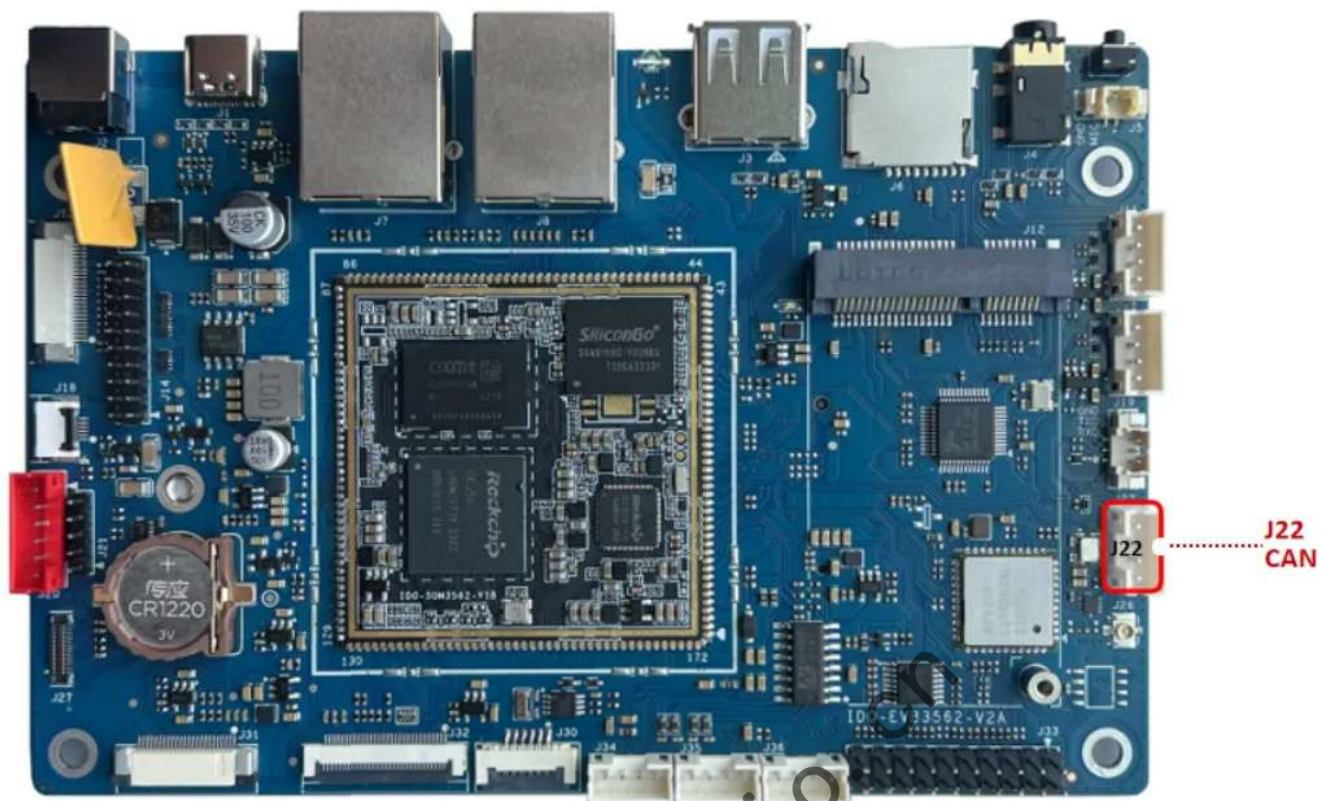
主板配置了一路硬盘接口（M.2），位于主板背面J38，如下图所示：



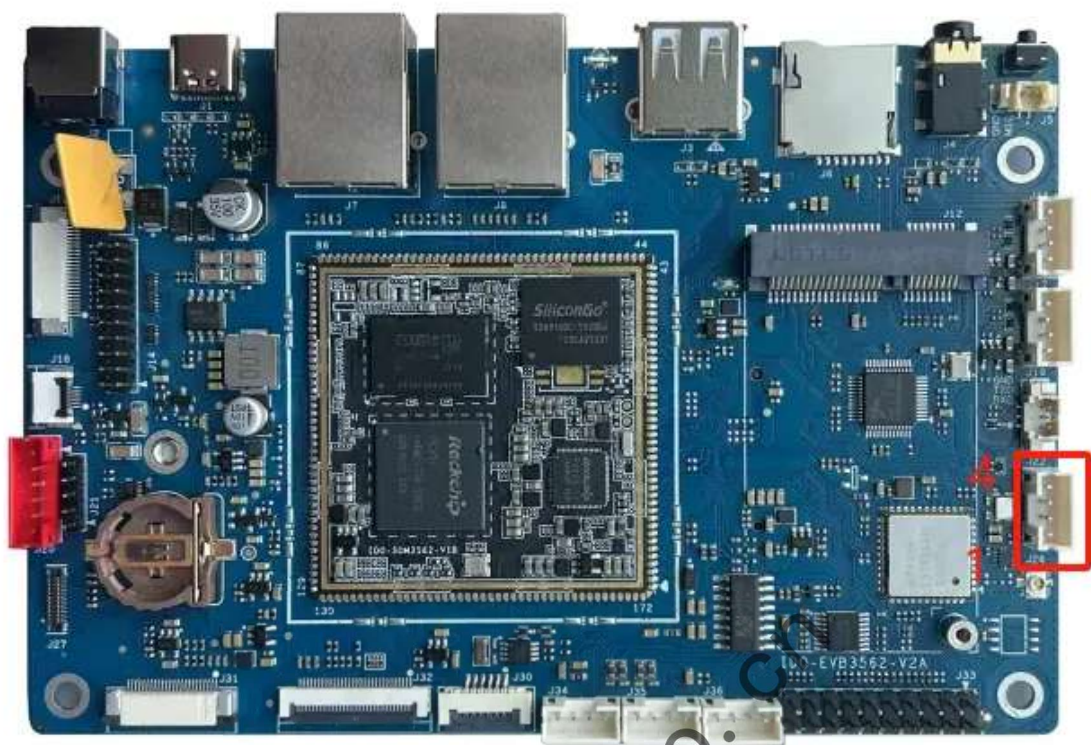
注意：PCIE功能目前和J8网卡(eth1)冲突，如果需要此功能需要更改软硬件。

14 CAN

主板配置一路CAN接口，如下图所示。



序号	丝印	设备节点
1	J22	CAN0



引脚定义:

序号	定义	电平/V	说明
1	VCC	5V/3.3V	默认5V供电输出，可配置为3.3V
2	CAN0_H	/	CAN信号
3	CAN0_L	/	
4	GND	GND	电源地

CAN命令行测试命令:

▼

C

```
1  #关闭can0设备
2  ip link set can0 down
3
4  #普通can协议 (1.0)
5  ip link set can0 type can bitrate 125000 triple-sampling on
6
7  #查看can信息, 波特率等等
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 #can格式发送
17 cansend can0 123#1122334455667788
```

15 扩展接口

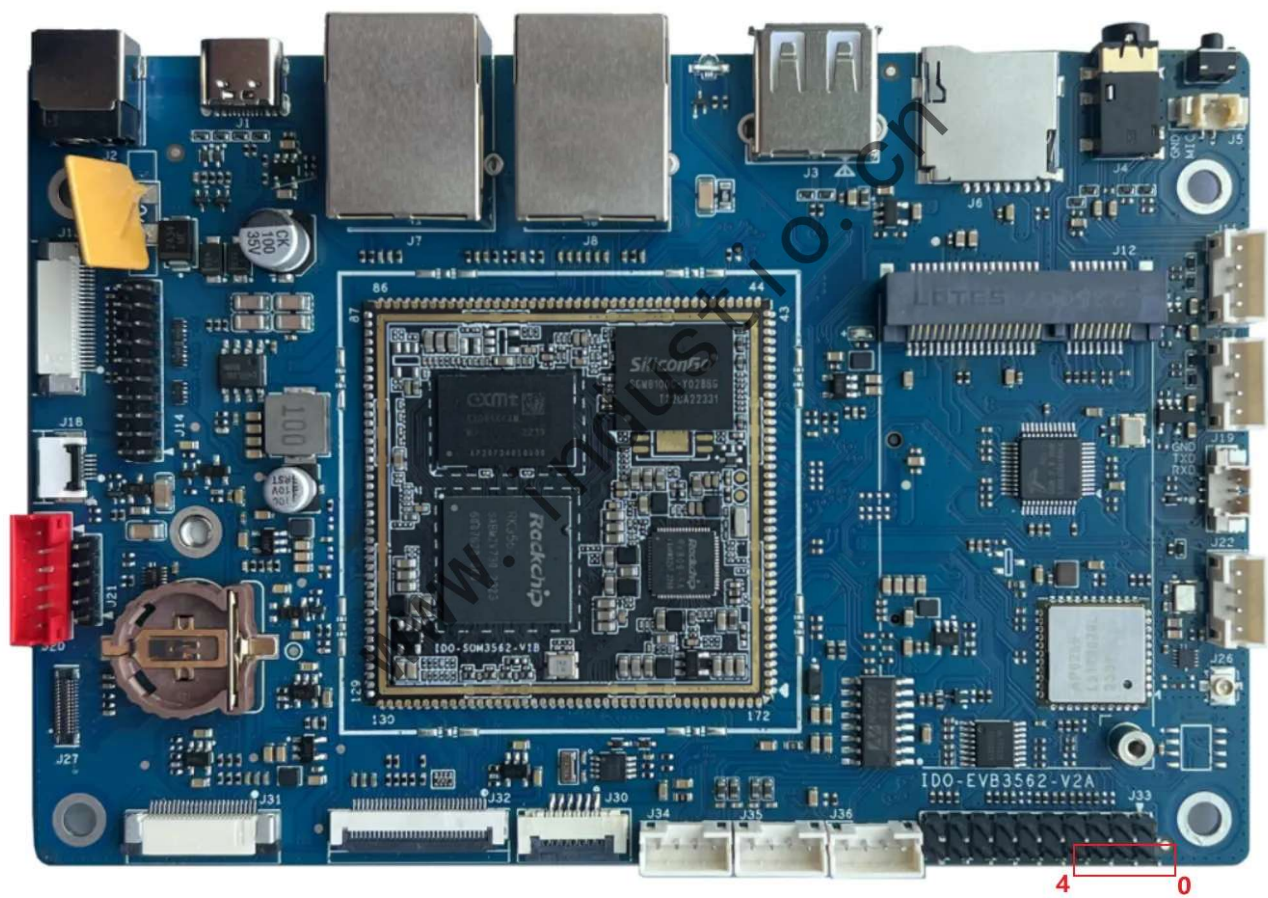
主板配置了一排扩展接口，位于J33，扩展接口的各个引脚功能，如下表所示：

pin	功能	pin	功能
1	5V输出	2	SPI0_CSN0_M0/PWM5_M0
3	UART3_TX	4	SPI0_CLK_M0/PWM0_M0
5	UART3_RX	6	SPI0_MOSI_M0/PWM1_M0
7	UART6_TX	8	SPI0_MISO_M0/PWM2_M0
9	UART6_RX	10	ADC0
11	UART8_TX	12	ADC1
13	UART8_RX	14	ADC2

15	I2C5_SCL_M0/UART 9_TX		16	ADC3
17	I2C5_SDA_M0/UART 9_RX		18	ADC4
19	GND		20	ADC5

15.1 GPIO

主板扩展接口有四路GPIO，如下图所示：



序号	管脚标号	GPIO 标号	LIBGPIO节点
1	SPI0_CSN0_M0/PW M5_M0	GPIO0_C2	gpiochip0 18
2	SPI0_CLK_M0/PWM0 _M0	GPIO0_C3	gpiochip0 19

3	SPI0_MOSI_M0/PWM 1_M0	GPIO0_C4	gpiochip0 20
4	SPI0_MISO_M0/PWM 2_M0	GPIO0_C5	gpiochip0 21

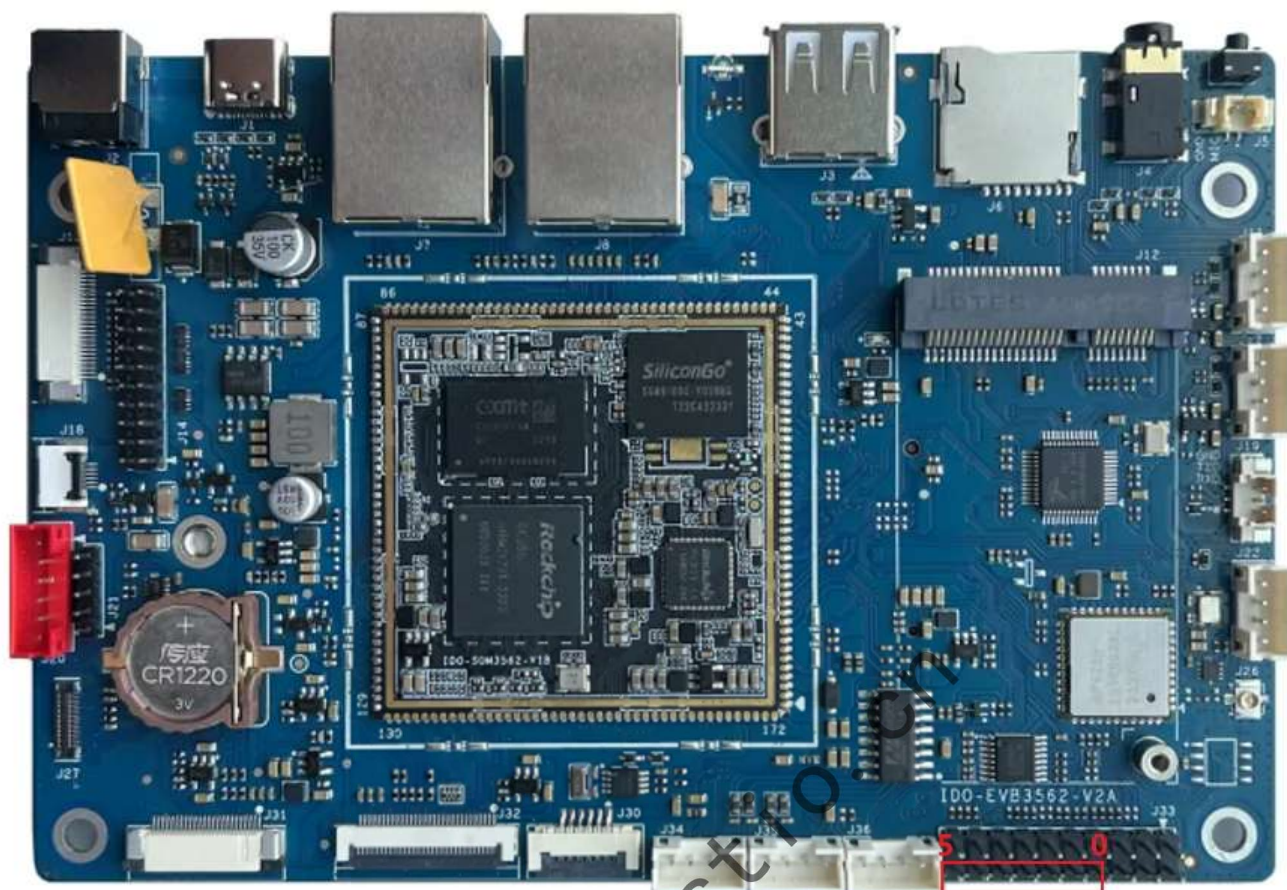
GPIO控制，根据上表【LIBGPIO节点】进行命令操作，以GPIO0_C2为例：

▼Shell

```
1  #输出：
2  #1为输出高电平(3.3V)，0为输出低电平(0V)。
3  root@linaro-alip:/# gpioset gpiochip0 18=1
4
5  root@linaro-alip:/# gpioset gpiochip0 18=0
6
7  #输入：
8  #0为外部输入低电平，1为外部输入高电平。
9  root@linaro-alip:/# gpioget gpiochip0 18
10 0
11 root@linaro-alip:/# gpioget gpiochip0 18
12 1
```

15.2 ADC

主板扩展接口有六路ADC，如下图所示：



扩展接口配置了6路ADC输入（参考电压为1.8v，支持10bit ADC采样）如下表所示：

名称	设备节点
ADCIN0	/sys/bus/iio/devices/iio:device1/in_voltage0_raw
ADCIN1	/sys/bus/iio/devices/iio:device1/in_voltage1_raw
ADCIN2	/sys/bus/iio/devices/iio:device1/in_voltage2_raw
ADCIN3	/sys/bus/iio/devices/iio:device1/in_voltage3_raw
ADCIN4	/sys/bus/iio/devices/iio:device1/in_voltage4_raw
ADCIN5	/sys/bus/iio/devices/iio:device1/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ADCIN5为例：

硬件连接：ADC5外接0V。

1.读取in_voltage5_raw的值，命令如下：

```
1 root@linaro-alip:/# cat /sys/bus/iio/devices/iio:device1/in_voltage5_raw
2 0
```

2.计算ADC5的输入电压V

$$V = (\text{voltage_raw}/1024) * 1.8V = (0/1024) * 1.8V = 0V$$

硬件连接：ADC5外接1.8V。

1.读取in_voltage5_raw的值，命令如下：

```
1 root@linaro-alip:/# cat /sys/bus/iio/devices/iio:device1/in_voltage5_raw
2 1020
```

2.计算ADC5的输入电压：

$$V = (\text{voltage_raw}/1024) * 1.8V = (1020/1024) * 1.8V = 1.79V$$

注意：ADC脚默认悬空状态，需要外部提供0~1.8v电压。

16 系统使用说明

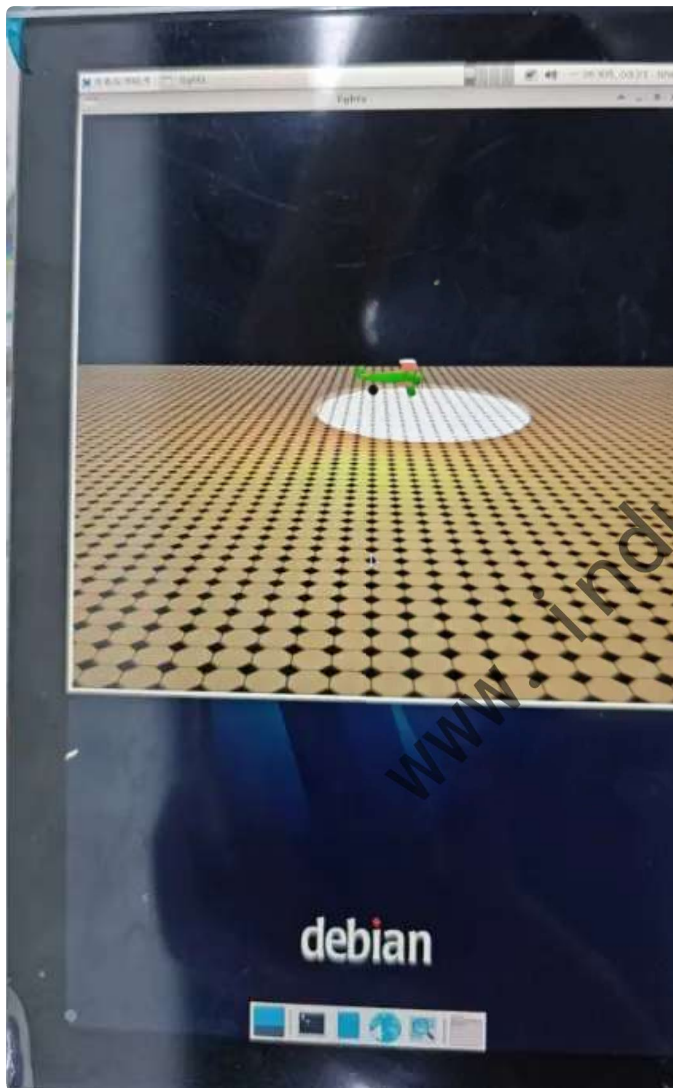
16.1 QT5

16.1.1 3d测试

执行qt5 3d demo：

```
▼ Bash |
1 root@linaro-alip:/# chmod a+x lights
2 root@linaro-alip:/#
3 root@linaro-alip:/# ./lights
4 QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-root'
5 QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-root'
6 arm_release_ver: g24p0-00eac0, rk_so_ver: 7
```

如图所示：



16.2 glmark2

```

1  #设置cpu性能模式
2  root@linaro-alip:/# echo performance > /sys/devices/system/cpu/cpufreq/policy0/scaling_governor
3
4  #测试系统性能
5  root@linaro-alip:/# glmark2-es2
6  arm_release_ver: g24p0-00eac0, rk_so_ver: 7
7  =====
8      glmark2 2023.01
9  =====
10     OpenGL Information
11     GL_VENDOR:      ARM
12     GL_RENDERER:    Mali-G52
13     GL_VERSION:     OpenGL ES 3.2 v1.g24p0-00eac0.a6b5f2be9252f82510fcbe88
14     Surface Config: buf=32 r=8 g=8 b=8 a=8 depth=24 stencil=0 samples=0
15     Surface Size:   800x600 windowed
16     =====
17  [build] use-vbo=false: FPS: 332 FrameTime: 3.021 ms
18  [build] use-vbo=true: FPS: 403 FrameTime: 2.485 ms
19  [texture] texture-filter=nearest: FPS: 450 FrameTime: 2.227 ms
20  [texture] texture-filter=linear: FPS: 448 FrameTime: 2.237 ms
21  [texture] texture-filter=mipmap: FPS: 450 FrameTime: 2.224 ms
22  [shading] shading=gouraud: FPS: 336 FrameTime: 2.983 ms
23  [shading] shading=blinn-phong-inf: FPS: 364 FrameTime: 2.752 ms
24  [shading] shading=phong: FPS: 335 FrameTime: 2.987 ms
25  [shading] shading=cel: FPS: 337 FrameTime: 2.972 ms
26  [bump] bump-render=high-poly: FPS: 192 FrameTime: 5.222 ms
27  [bump] bump-render=normals: FPS: 462 FrameTime: 2.166 ms
28  [bump] bump-render=height: FPS: 449 FrameTime: 2.228 ms
29  [effect2d] kernel=0,1,0;1,-4,1;0,1,0,: FPS: 301 FrameTime: 3.331 ms
30  [effect2d] kernel=1,1,1,1,1;1,1,1,1,1;1,1,1,1,1,: FPS: 151 FrameTime: 6.66
31  3 ms
32  [pulsar] light=false:quads=5:texture=false: FPS: 423 FrameTime: 2.367 ms
33  [desktop] blur-radius=5:effect=blur:passes=1:separable=true:windows=4: FPS: 164 FrameTime: 6.105 ms
34  [desktop] effect=shadow:windows=4: FPS: 334 FrameTime: 2.996 ms
35  [buffer] columns=200:interleave=false:update-dispersion=0.9:update-fraction=0.5:update-method=map: FPS: 120 FrameTime: 8.339 ms
36  [buffer] columns=200:interleave=false:update-dispersion=0.9:update-fraction=0.5:update-method=subdata: FPS: 121 FrameTime: 8.332 ms
37  [buffer] columns=200:interleave=true:update-dispersion=0.9:update-fraction=0.5:update-method=map: FPS: 153 FrameTime: 6.558 ms
38  [ideas] speed=duration: FPS: 206 FrameTime: 4.871 ms
39  [jellyfish] <default>: FPS: 262 FrameTime: 3.825 ms

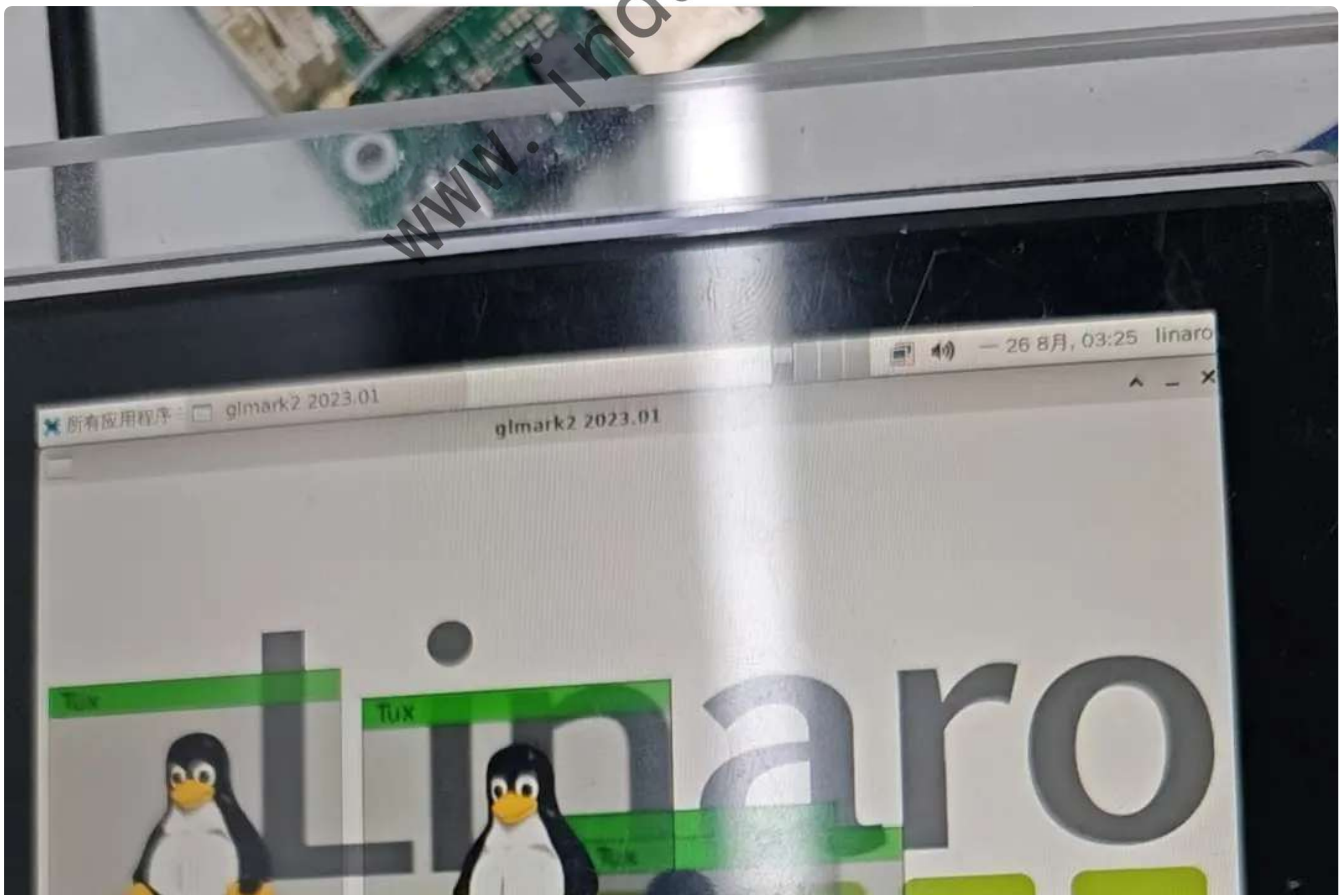
```

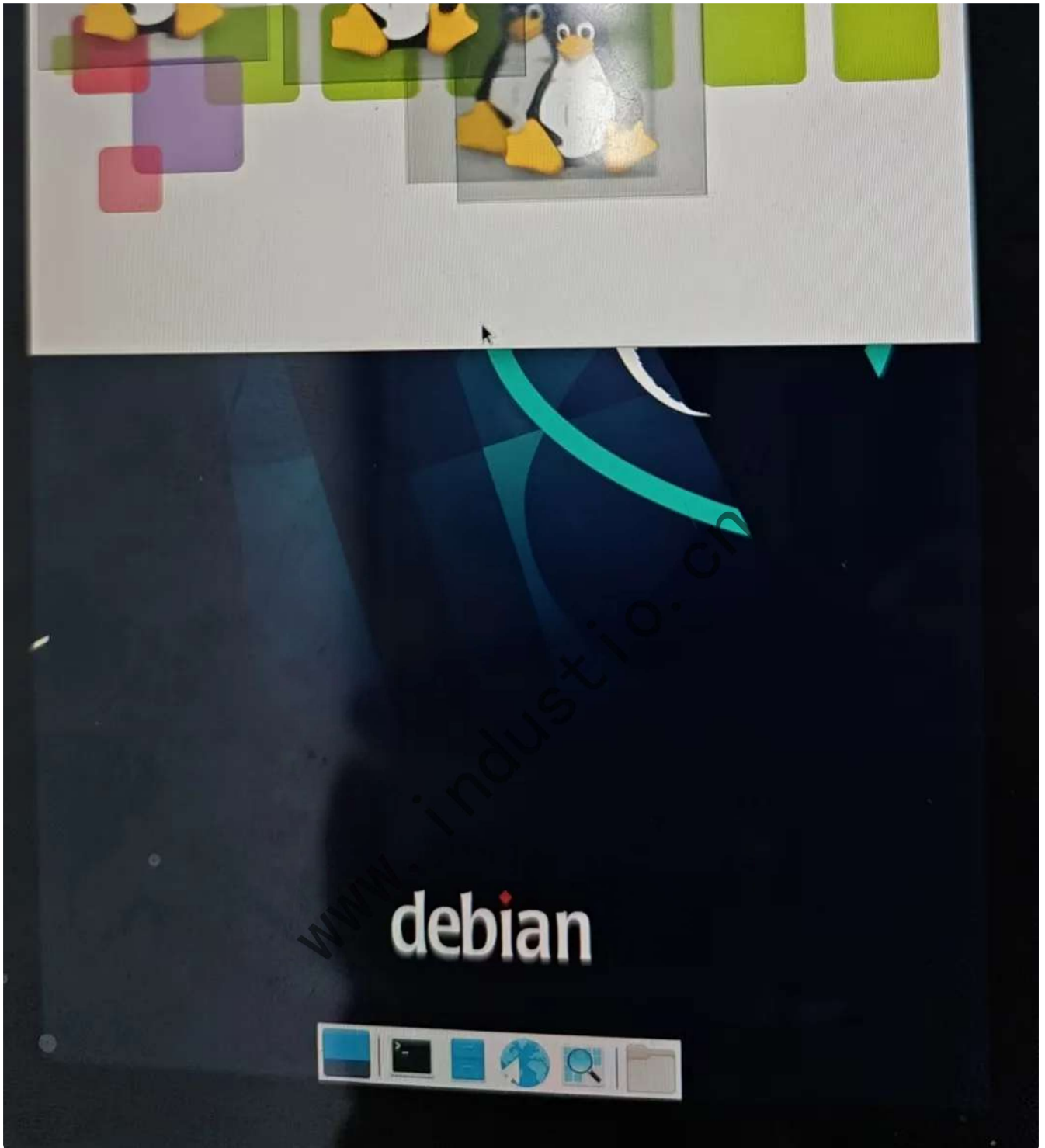
```

39 [terrain] <default>: FPS: 38 FrameTime: 26.619 ms
40 [shadow] <default>: FPS: 215 FrameTime: 4.657 ms
41 [refract] <default>: FPS: 62 FrameTime: 16.133 ms
42 [conditionals] fragment-steps=0:vertex-steps=0: FPS: 420 FrameTime: 2.386
ms
43 [conditionals] fragment-steps=5:vertex-steps=0: FPS: 360 FrameTime: 2.783
ms
44 [conditionals] fragment-steps=0:vertex-steps=5: FPS: 419 FrameTime: 2.387
ms
45 [function] fragment-complexity=low:fragment-steps=5: FPS: 390 FrameTime: 2
.571 ms
46 [function] fragment-complexity=medium:fragment-steps=5: FPS: 331 FrameTim
e: 3.026 ms
47 [loop] fragment-loop=false:fragment-steps=5:vertex-steps=5: FPS: 389 Frame
Time: 2.575 ms
48 [loop] fragment-steps=5:fragment-uniform=false:vertex-steps=5: FPS: 390 Fr
ameTime: 2.569 ms
49 [loop] fragment-steps=5:fragment-uniform=true:vertex-steps=5: FPS: 357 Fra
meTime: 2.806 ms
50 =====
51 glmark2 Score: 307
52 =====

```

如图所示：



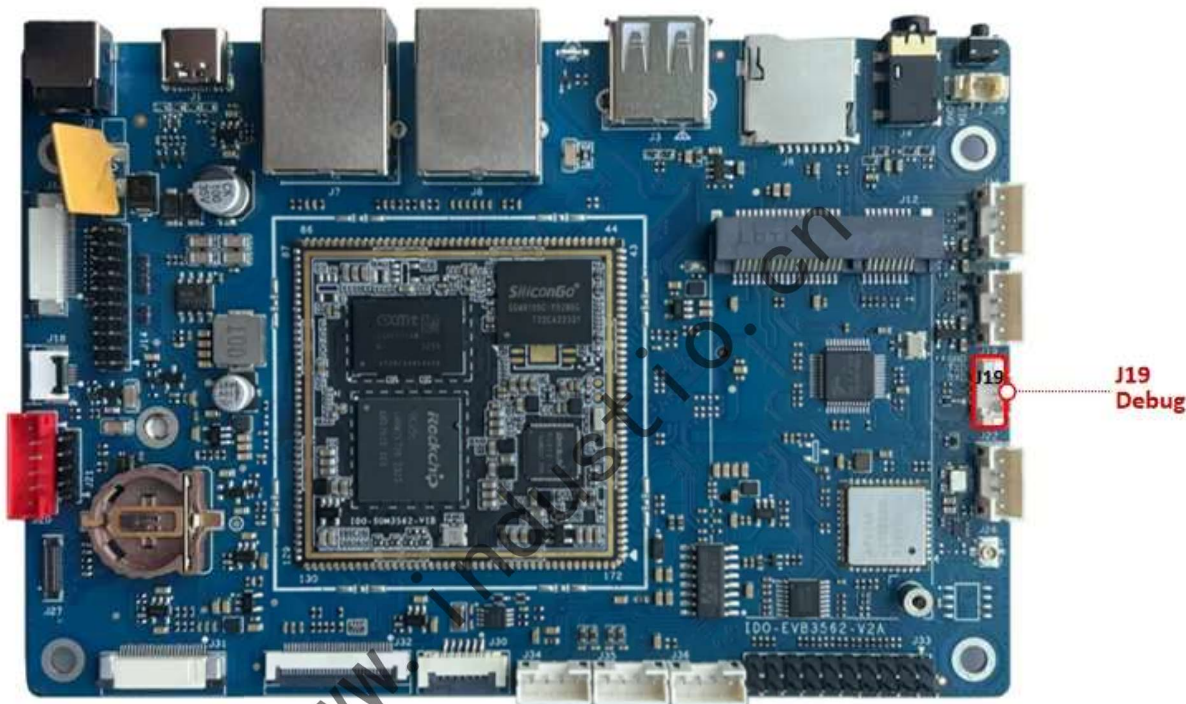


IDO-EVB3562-Buildroot系统

1 调试

1.1 串口调试

主板调试串口位于J19，建议使用配套的usb转串口工具，如下图所示：



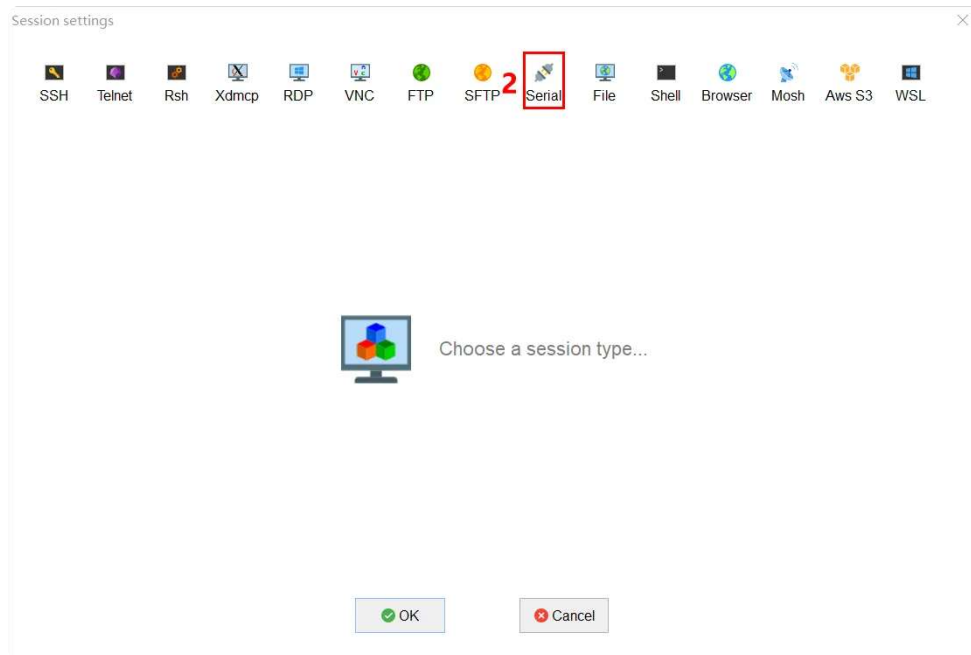
1. 打开MobaXterm，下载链接如下：

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

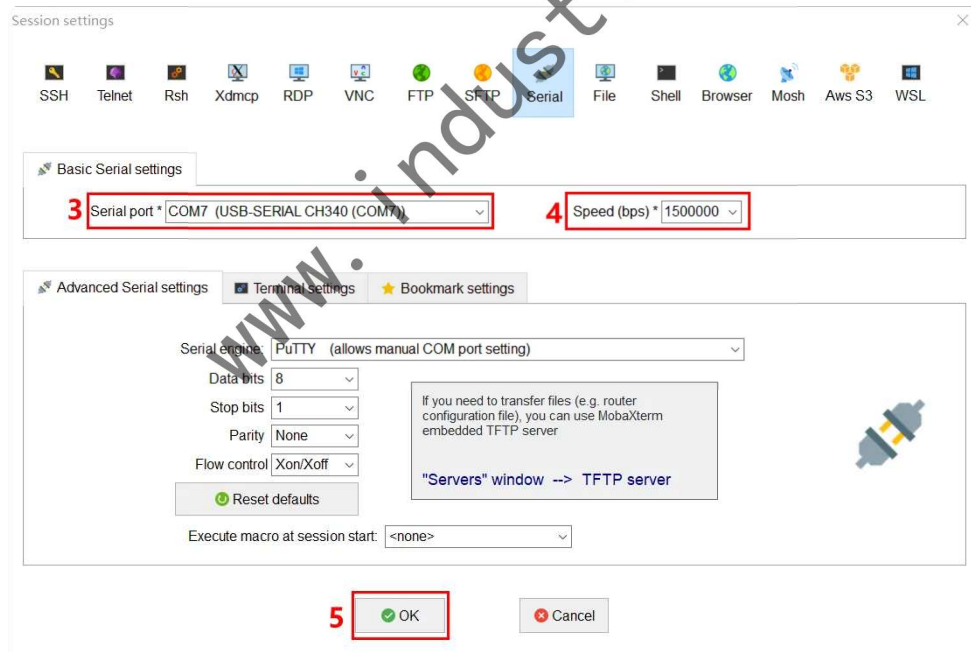
1. 选择sessionSerial，如下图所示：



2. 将Serial port修改为在设备管理器中找到的COM端口

3. 设置Speed(bps)为1500000

4. 点击【OK】按钮，如下图所示：



结果如下图所示：

```
root@rk3562:/# ls
bin                home               lost+found         proc               sdcard            userdata
busybox.fragment  info              media              rockchip-test     sys               usr
data              lib               mnt               root              system           var
dev              lib64            oem               run               tmp              vendor
etc              linuxrc          opt               sbin              udisk
root@rk3562:/# |
```

1.2 adb调试

主板adb接口位于J1，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:\Users\industio_mhk>adb shell
root@rk3562:/# ls
bin          dev          info         linuxrc      mnt          proc         run          sys          udisk        var
busybox.fragment  etc         lib         lost+found  oem          rockchip-test sbin         system      userdata    vendor
data         home        lib64       media       opt          root         sdcard      tmp         usr
root@rk3562:/#
```

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：123456
3. 端口：22

ssh调试，如下图所示：

```
[C:\~]$ ssh root@192.168.0.115

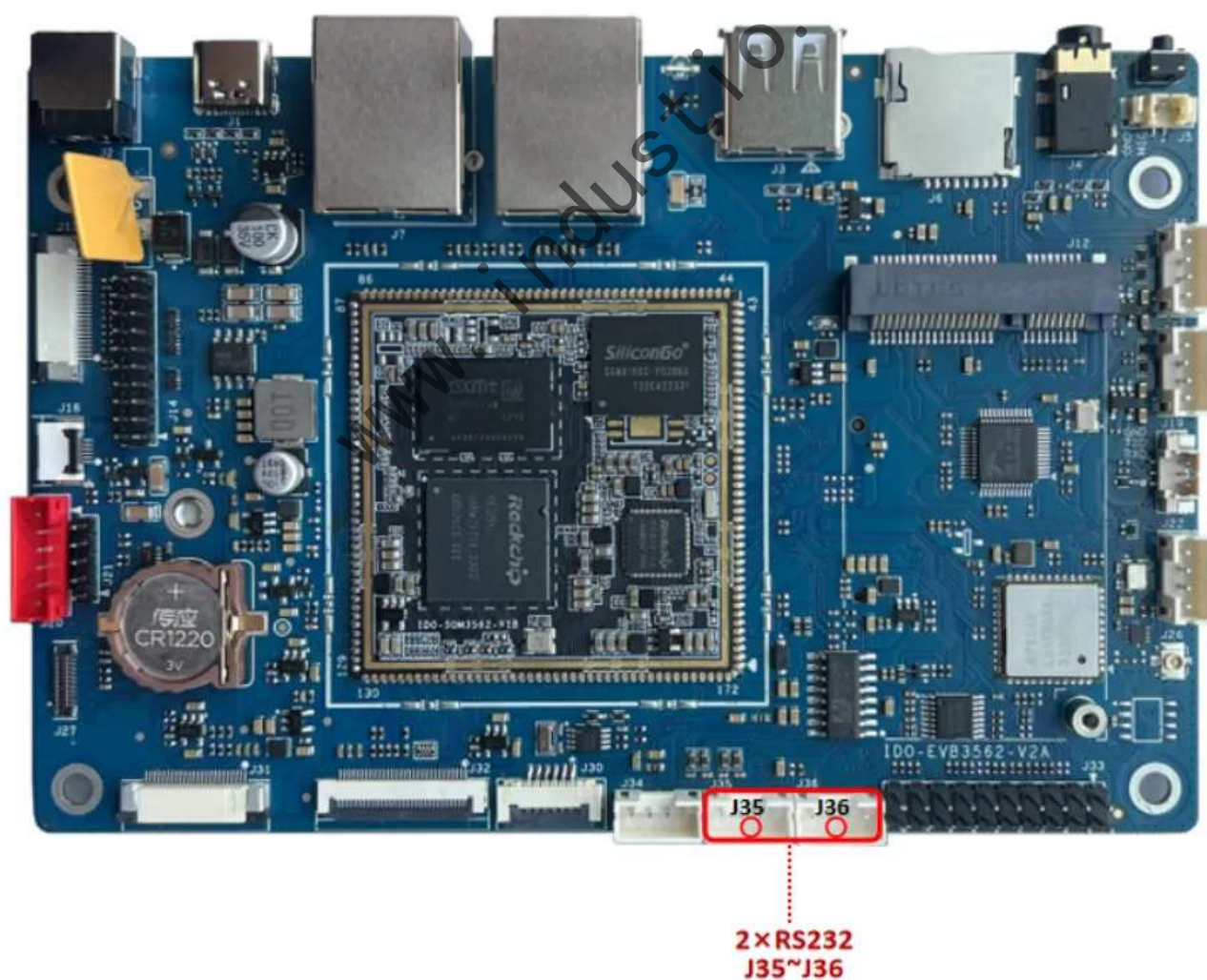
Connecting to 192.168.0.115:22...
Connection established.
To escape to local shell, press 'Ctrl+Alt+J'.

root@rk3562:~#
```

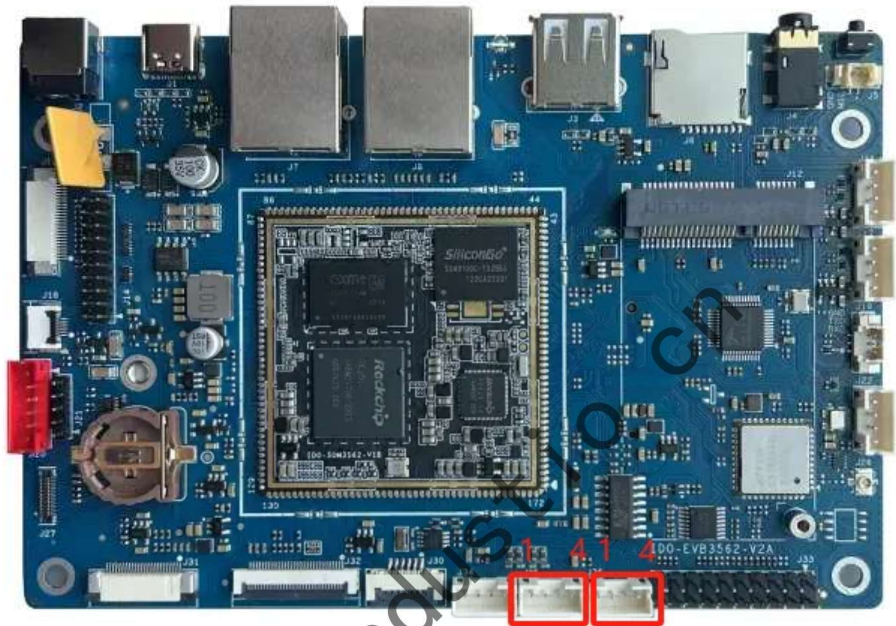
2 串口

主板共配置4路串口（除调试串口），如下图所示：

2.1 RS232



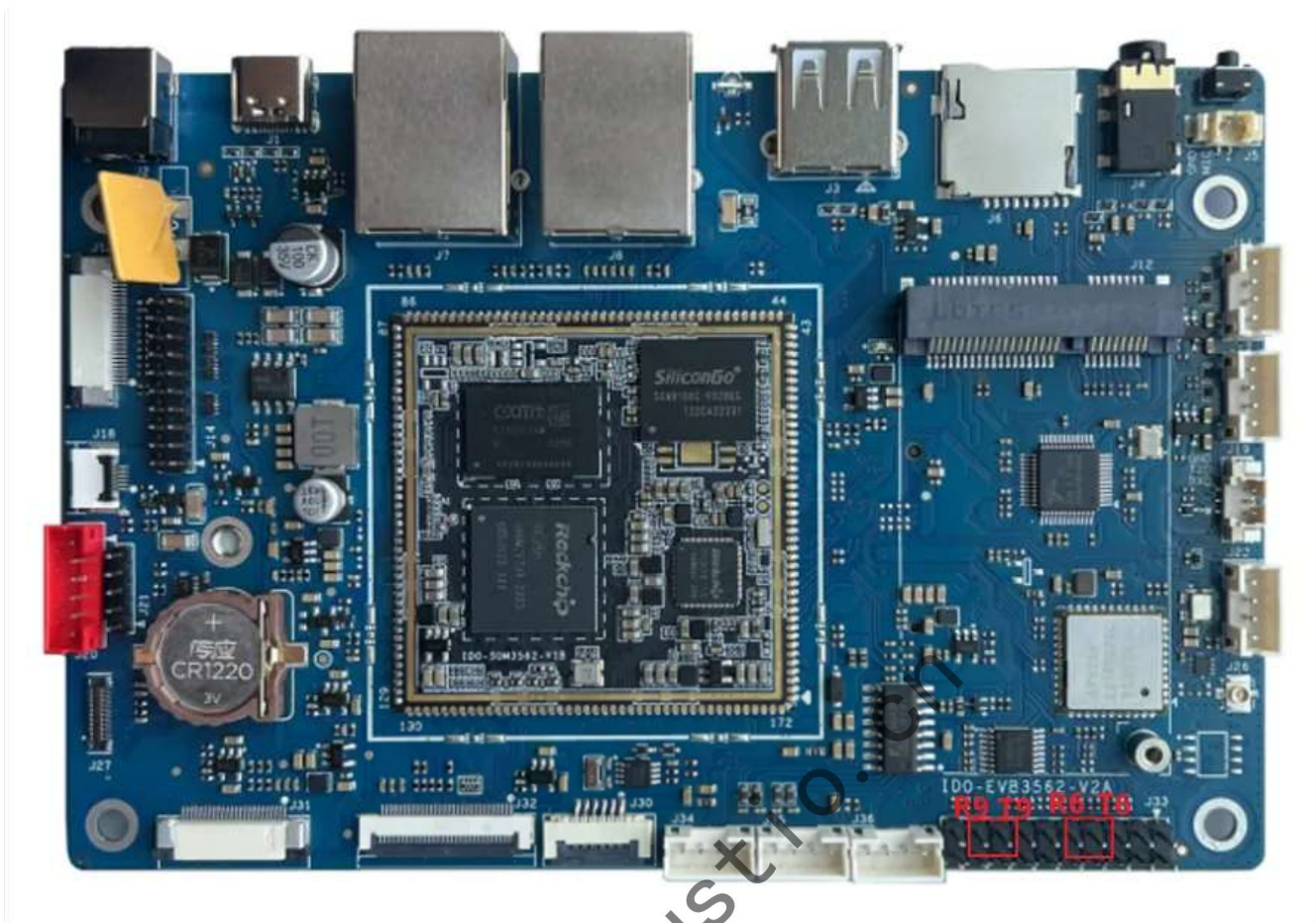
丝印	电平类型	设备节点	说明
J35	RS232	/dev/ttyS8	/
J36	RS232	/dev/ttyS3	/



引脚定义：

序号	定义	电平/V	说明
1	VCC	5V/3.3V	默认5V电源输出，可配置为3.3V
2	RX	/	RS232-接收
3	TX	/	RS232-发送
4	GND	GND	电源地

2.2 TTL



设备节点，如下表所示：

序号	电平类型	设备节点	说明
1	TTL	/dev/ttyS9	/
2	TTL	/dev/ttyS6	默认配置CAN0，此接口无效

使用microcom可以进行串口收发测试，命令如下：

▼Shell

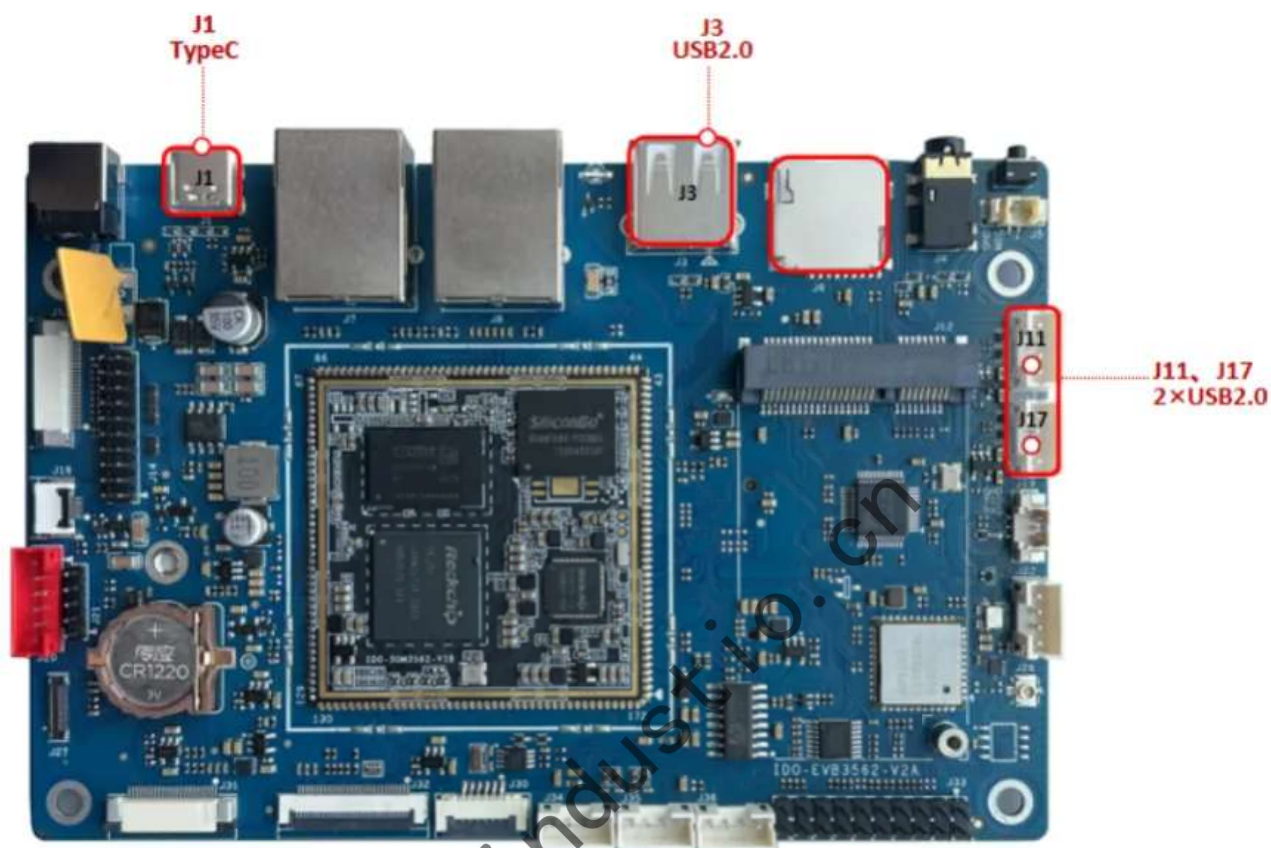
```
1 root@rk3562:~# microcom -s 115200 -p /dev/ttyS3
```

注意：

- 1.测试完成，按Ctrl+x退出。
- 2./dev/ttyS6 默认不通，此功能需要修改软硬件。

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	J1	OTG
USB2	J3	USB2.0 HOST
USB3	J11	USB2.0 HOST
USB4	J17	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:~# mount
2 /dev/mmcblk0p8 on / type ext4 (rw,relatime)
3 ...
4 /dev/sda1 on /mnt/udisk type vfat (rw,nodev,noexec,noatime,nodiratime,fmask
=0022,dmask=0022,codepage=936,iocharset=utf8,shortname=mixed,errors=remount
-ro)
```

USB电源控制，如下表所示：

USB端口	默认状态	动作	命令
USB2	开启	关闭电源	echo 0 > /sys/class/leds/usb_host2/brightness
		开启电源	echo 1 > /sys/class/leds/usb_host2/brightness
USB3	开启	关闭电源	echo 0 > /sys/class/leds/usb_host3/brightness
		开启电源	echo 1 > /sys/class/leds/usb_host3/brightness
USB4	开启	关闭电源	echo 0 > /sys/class/leds/usb_host4/brightness
		开启电源	echo 1 > /sys/class/leds/usb_host4/brightness

供电控制说明：设备节点写"0"关闭电源，写"1"开启电源。

4 Micro SD

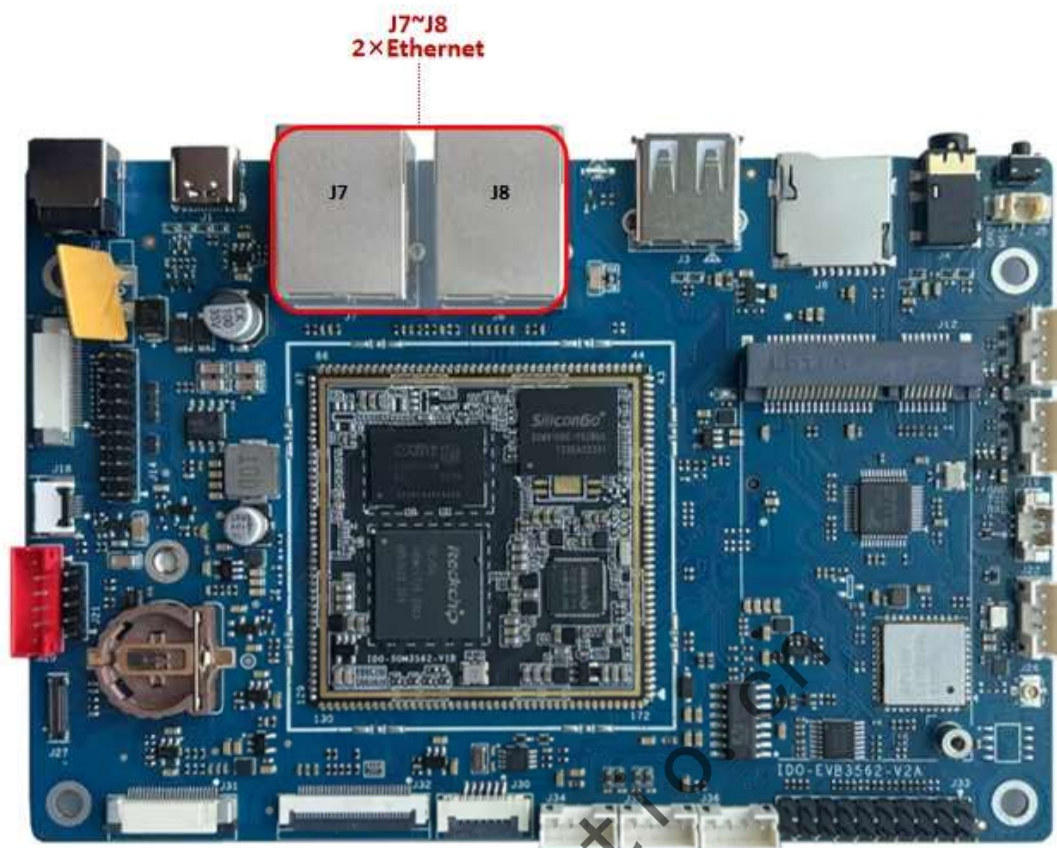
主板共配置一路Micro SD接口，插入SD卡后，默认挂载到/mnt/sdcard目录，如下所示：

```
1 root@rk3562:~# fdisk -l
2 Disk /dev/ram0: 4 MiB, 4194304 bytes, 8192 sectors
3 Units: sectors of 1 * 512 = 512 bytes
4 Sector size (logical/physical): 512 bytes / 4096 bytes
5 I/O size (minimum/optimal): 4096 bytes / 4096 bytes
6 ...
7 Device          Boot Start      End  Sectors  Size Id Type
8 /dev/mmcblk1p1    8192 30449663 30441472 14.5G  c W95 FAT32 (LBA)
9
10 root@rk3562:~# mount
11 /dev/mmcblk0p8 on / type ext4 (rw,relatime)
12 ...
13 /dev/mmcblk1p1 on /mnt/sdcard type vfat (rw,nodev,noexec,noatime,nodiratime,
    fmask=0022,dmask=0022,codepage=936,iocharset=utf8,shortname=mixed,errors
    =remount-ro)
```

5 Ethernet

主板共配置2路千兆网接口，如下图所示：

www.industio.cn



序号	接口位置	速率	网络节点
1	J7	千兆	eth0
2	J8	千兆	eth1

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@rk3562:~# ifconfig eth0 192.168.3.123
```

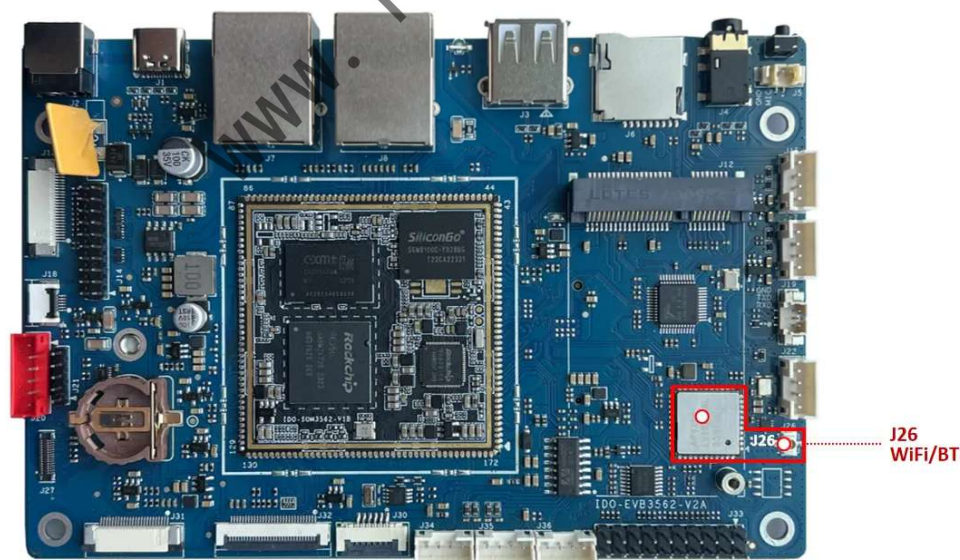
如需设置永久静态IP，则可通过添加/etc/network/interfaces，命令如下：

```
1 root@rk3562:~# cat /etc/network/interfaces
2 # interface file auto-generated by buildroot
3
4 auto lo
5 iface lo inet loopback
6
7 auto eth0
8     iface eth0 inet static
9     address 192.168.3.123
10    netmask 255.255.255.0
11    gateway 192.168.3.1
12    nameserver 192.168.3.1
13
14
15 #立即生效
16 root@rk3562:~# /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，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：

```
1 root@rk3562:~# ifconfig wlan0
2 wlan0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
3         ether c0:f5:35:12:c5:f8 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

命令如下：

www.industio.cn

```
1  #扫描热点
2  root@rk3562:~# iwlist wlan0 scan
3
4  #设置账号密码
5  root@rk3562:~# vim /etc/wpa_supplicant.conf
6  ctrl_interface=/var/run/wpa_supplicant
7  ap_scan=1
8
9  network={
10     ssid="Industio_5.8"
11     psk="industio666"
12     key_mgmt=WPA-PSK
13 }
14
15 #连接WiFi
16 root@rk3562:~# killall wpa_supplicant
17 root@rk3562:~# rm -f /var/run/wpa_supplicant/wlan0
18 root@rk3562:~# wpa_supplicant -Dnl80211 -c /etc/wpa_supplicant.conf -i wlan0 &
19 root@rk3562:~# ifconfig wlan0
20 wlan0      Link encap:Ethernet  HWaddr C0:F5:35:12:C5:F8
21            inet addr:192.168.0.88  Bcast:192.168.0.255  Mask:255.255.255.0
22            inet6 addr: fe80::19f0:b3fd:c566:ca2f/64 Scope:Link
23            UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
24            RX packets:638 errors:0 dropped:1 overruns:0 frame:0
25            TX packets:184 errors:0 dropped:0 overruns:0 carrier:0
26            collisions:0 txqueuelen:1000
27            RX bytes:47450 (46.3 KiB)  TX bytes:21354 (20.8 KiB)
28
29 root@rk3562:~# ping www.baidu.com -I wlan0
30 PING www.wshifen.com (103.235.47.188) from 192.168.0.88 wlan0: 56(84) bytes of data.
31 64 bytes from 103.235.47.188 (103.235.47.188): icmp_seq=2 ttl=48 time=164 ms
32 64 bytes from 103.235.47.188 (103.235.47.188): icmp_seq=3 ttl=48 time=166 ms
33 64 bytes from 103.235.47.188 (103.235.47.188): icmp_seq=4 ttl=48 time=166 ms
```

其中"Industio_5.8"为热点名称, "industio666"为热点密码。

7 Bluetooth

7.1 开启蓝牙

开启蓝牙，命令如下：

```
▼ Shell
1 root@rk3562:~# hciconfig -a
2 hci0: Type: Primary Bus: UART
3   BD Address: C0:F5:35:12:C5:F9 ACL MTU: 1021:8 SCO MTU: 64:1
4   UP RUNNING
5   RX bytes:885 acl:0 sco:0 events:63 errors:0
6   TX bytes:3296 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: PERIPHERAL ACCEPT
11  Name: 'BCM4345C5'
12  Class: 0x000000
13  Service Classes: Unspecified
14  Device Class: Miscellaneous,
15  HCI Version: 4.2 (0x8) Revision: 0x0
16  LMP Version: 4.2 (0x8) Subversion: 0x6606
17  Manufacturer: Broadcom Corporation (15)
```

7.2 扫描设备

扫描设备，命令如下：

```
▼ Shell
1 root@rk3562:~# bluetoothctl
2 Agent registered
3 root@rk3562:~# power on
4 Changing power on succeeded
5 root@rk3562:~# scan on
```

7.3 连接蓝牙设备

连接蓝牙设备，命令如下：

```
1 [bluetooth]# pair CC:6B:1E:A2:90:68
2 Attempting to pair with CC:6B:1E:A2:90:68
3 hci0 device_flags_changed: CC:6B:1E:A2:90:68 (BR/EDR)
4     supp: 0x00000001 curr: 0x00000000
5 hci0 CC:6B:1E:A2:90:68 type BR/EDR connected eir_len 22
6 [CHG] Device CC:6B:1E:A2:90:68 INFO: 0x0002 (2)
7 [CHG] Device CC:6B:1E:A2:90:68 Connected: yes
8 Request PIN code]#
9 [agent] Enter PIN code: 1234
10 hci0 CC:6B:1E:A2:90:68 auth failed with status 0x0b (Rejected)
11 Failed to pair: org.bluez.Error.AuthenticationRejected
12 [DEL] Device 6C:52:1F:2D:36:40 6C-52-1F-2D-36-40
13 [DEL] Device 5F:09:B6:3E:8D:99 5F-09-B6-3E-8D-99
14 [DEL] Device 59:A9:CA:A2:72:CB 59-A9-CA-A2-72-CB
15 [DEL] Device 43:B2:1B:89:86:19 43-B2-1B-89-86-19
16 [DEL] Device 4B:C8:E7:8B:CC:F9 4B-C8-E7-8B-CC-F9
17 hci0 CC:6B:1E:A2:90:68 type BR/EDR disconnected with reason 2
18
```

电脑提示设备配对：



输入PIN码：

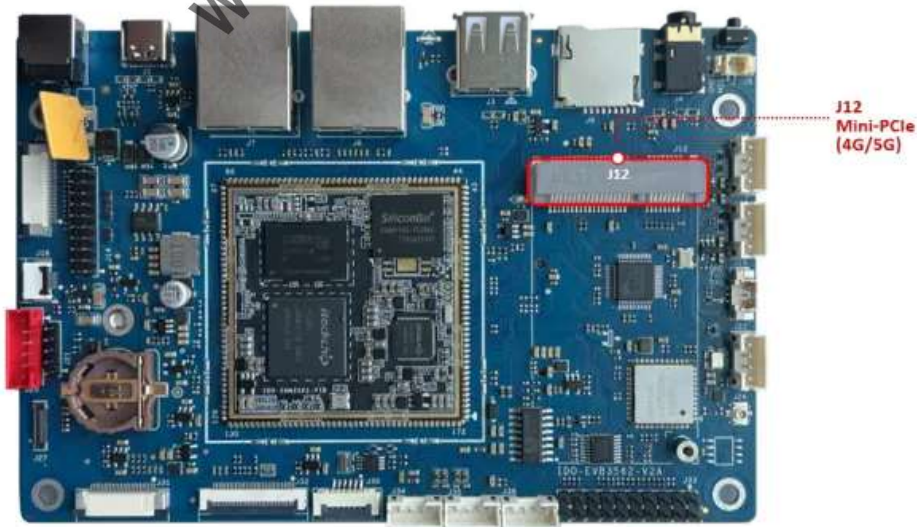


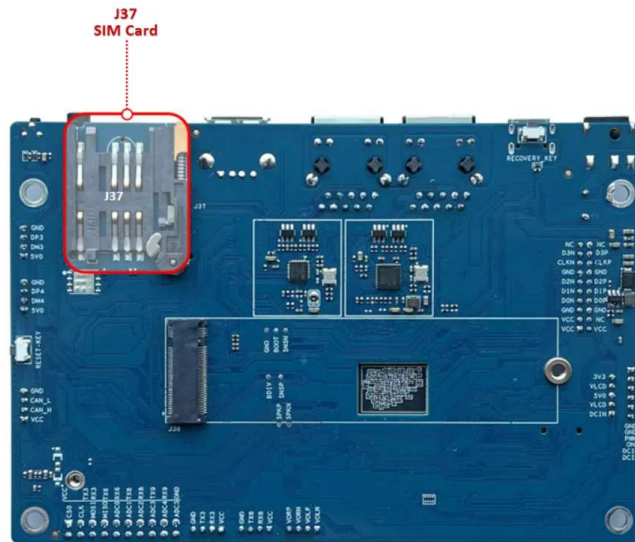
提示连接成功：



8 4G

主板配置了一路4G接口，默认适配EC20模块，如下图所示：





开机自动拨号，拨号成功后，产生wwan0网络节点：

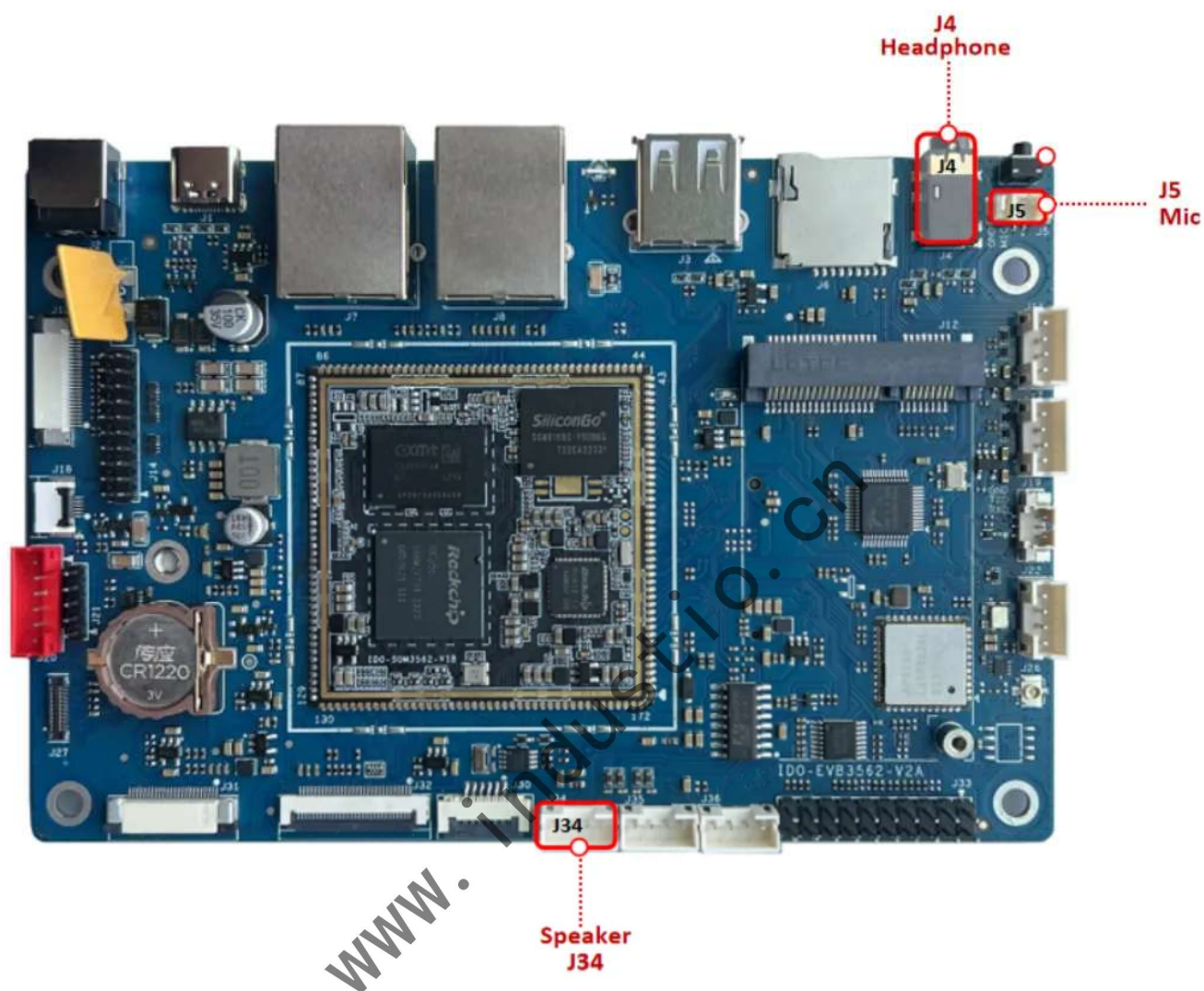
```
root@rk3562:~# ifconfig wwan0
wwan0    Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          inet addr:10.122.45.92  P-t-P:10.122.45.92  Mask:255.255.255.248
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:12 errors:0 dropped:0 overruns:0 frame:0
          TX packets:59 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1396 (1.3 KiB)  TX bytes:8214 (8.0 KiB)
```

测试4G上网功能，命令如下：

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

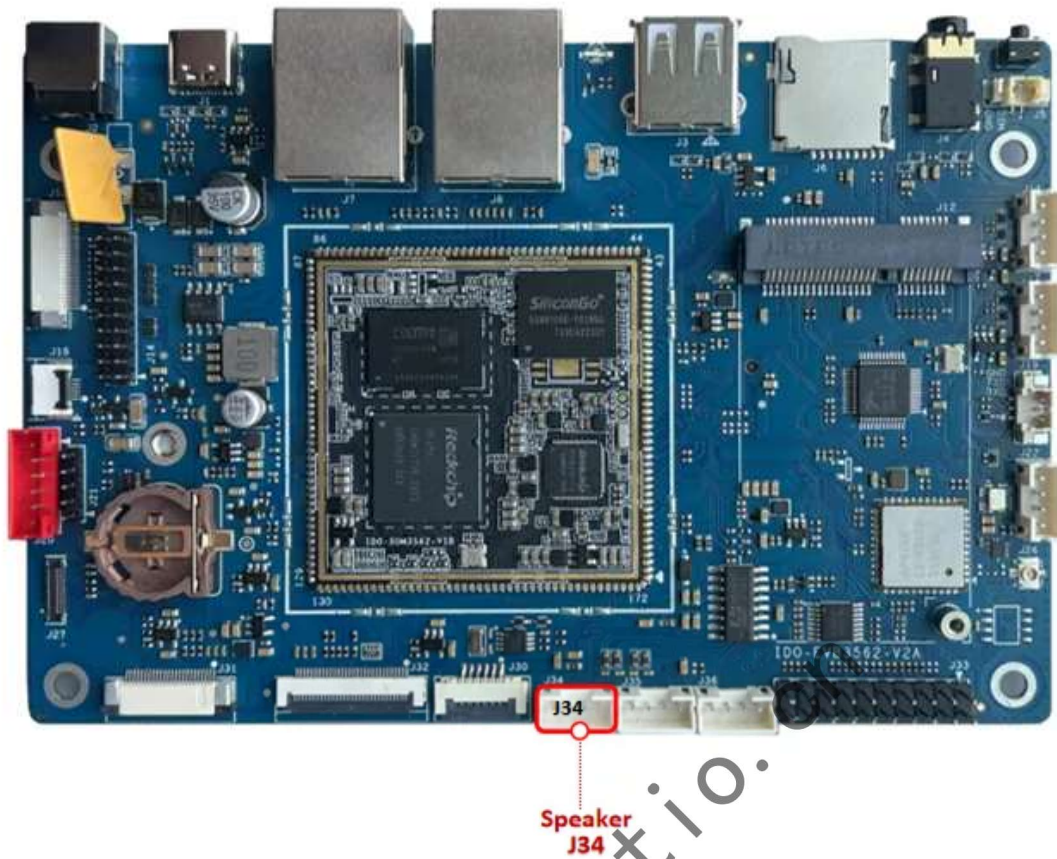
9 音频

主板音频接口如下所示：



9.1 喇叭

双声道Lineout接口位于J34，如下图所示：

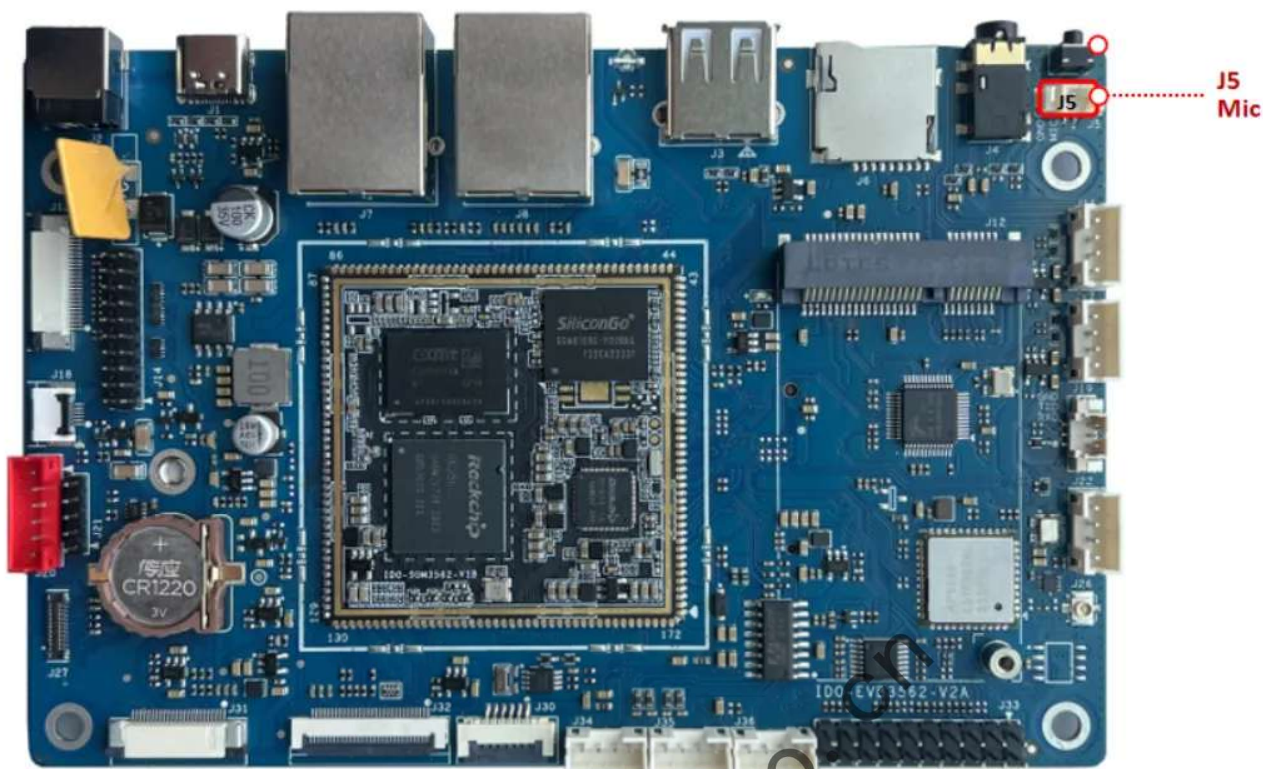


播放音频，命令如下：

```
1 root@rk3562:~# aplay -D hw:1,0 ./xinhuan.wav
```

9.2 MIC

MIC如下图所示：

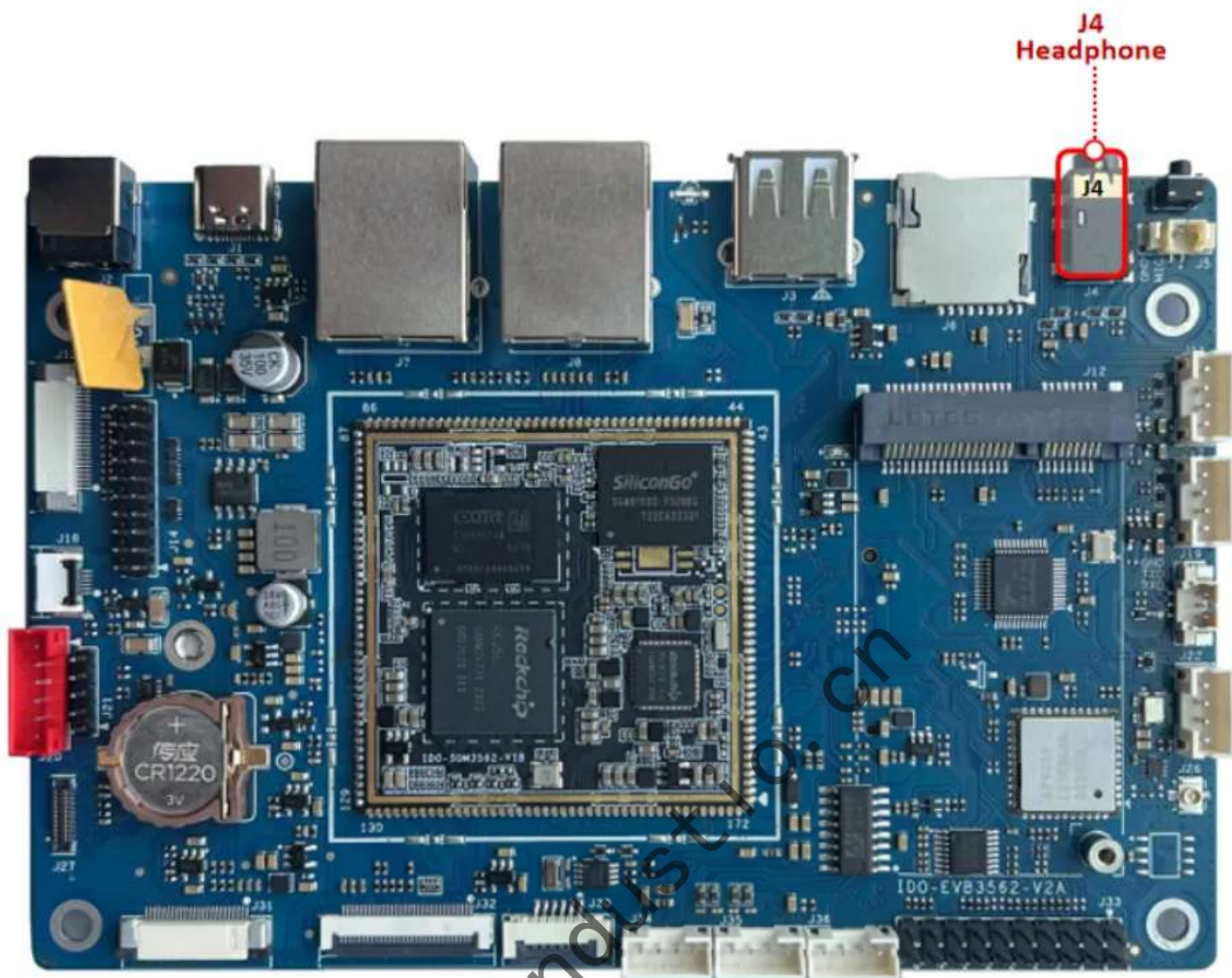


录音，命令如下：

```
1 root@rk3562:~# arecord -D hw:1,0 -r 48000 -c 2 -f S16_LE test.wav
```

9.3 耳机

耳机如下图所示：

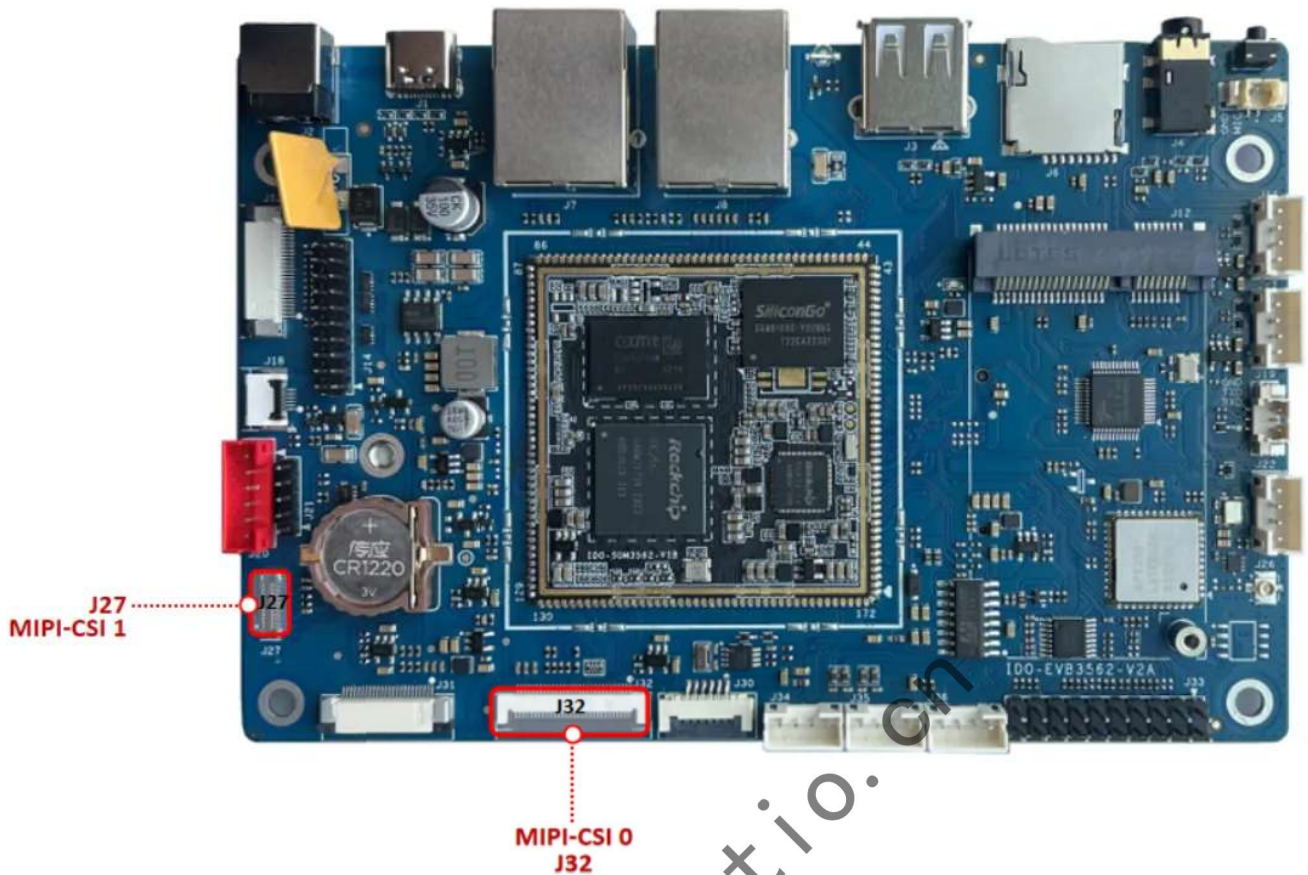


播放音频，命令如下：

```
1 root@rk3562:~# aplay -D hw:1,0 ./xinhuan.wav
```

10 Camera

主板配置2路摄像头，型号为ov13855（J27）和ov13855（J32），如下所示：



10.1 OV13855(J32)

预览

▼ Shell

```

1 root@rk3562:~# export DISPLAY=:0
2 root@rk3562:~# gst-launch-1.0 v4l2src device=/dev/video22 ! video/x-raw, fo
  rmat=NV12, width=1920, height=1080, framerate=30/1 ! videoconvert ! autovid
  eosink
  
```

抓图

▼ Shell

```

1 root@rk3562:~# export DISPLAY=:0
2 root@rk3562:~# v4l2-ctl -d /dev/video22 --set-fmt-video=width=1920,height=1
  080,pixelformat=NV12 --stream-mmap=4 --stream-to=/tmp/output.yuv --stream-c
  ount=1 --stream-skip=30
  
```

10.2 OV13855(J27)

预览

▼ Shell

```
1 root@rk3562:~# export DISPLAY=:0
2 root@rk3562:~# gst-launch-1.0 v4l2src device=/dev/video29 ! video/x-raw, format=NV12, width=1920, height=1080, framerate=30/1 ! videoconvert ! autovideosink
```

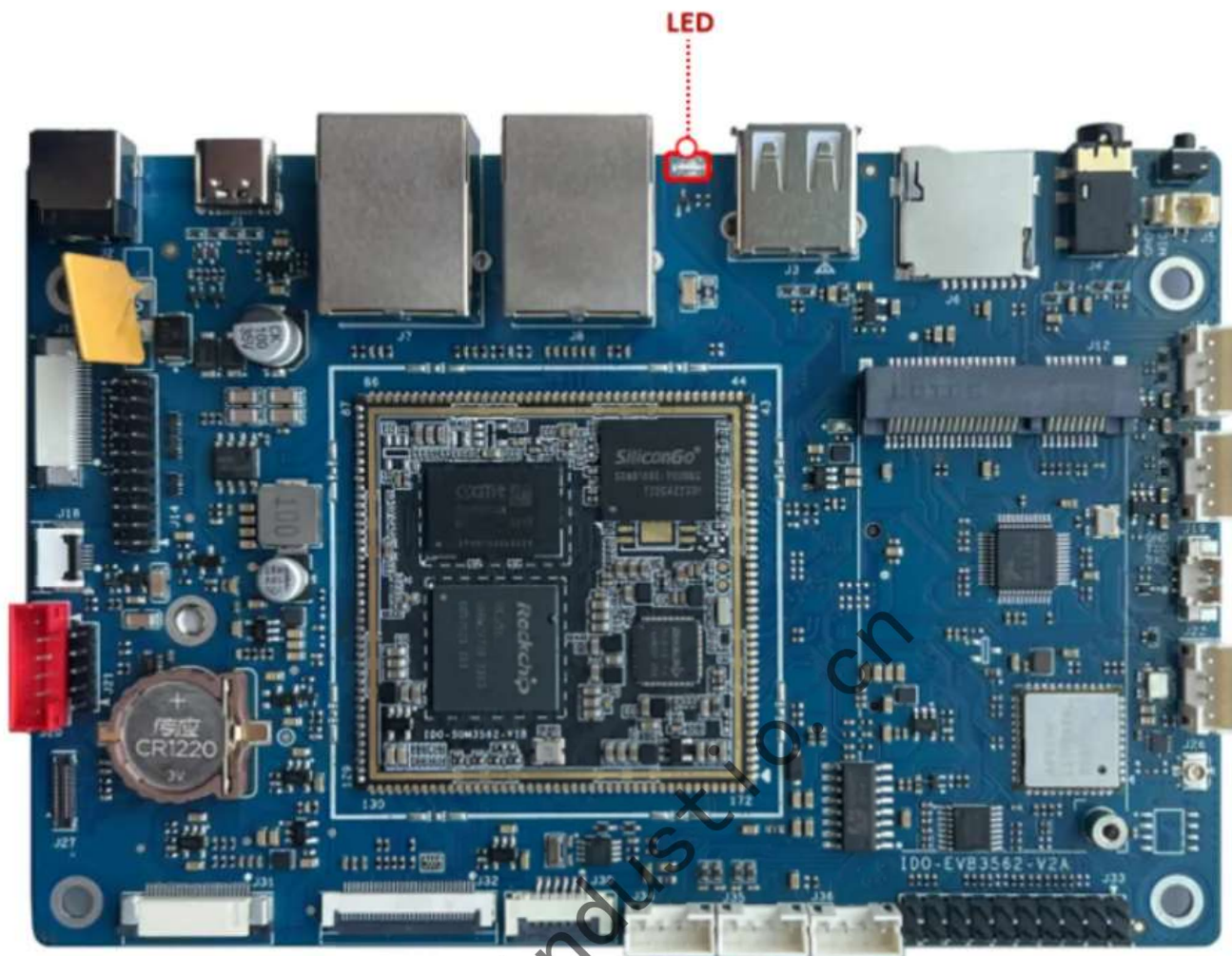
抓图

▼ Shell

```
1 root@rk3562:~# export DISPLAY=:0
2 root@rk3562:~# v4l2-ctl -d /dev/video29 --set-fmt-video=width=1920,height=1080,pixelformat=NV12 --stream-mmap=4 --stream-to=/tmp/output.yuv --stream-count=1 --stream-skip=30
```

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，系统中对应的设备节点为/dev/rtc0，如下图所示：



12.1 读取时间

读取RTC时间，命令如下：

```
1 root@rk3562:~# hwclock -r
2 Fri Aug 4 09:02:38 2017 0.000000 seconds
```

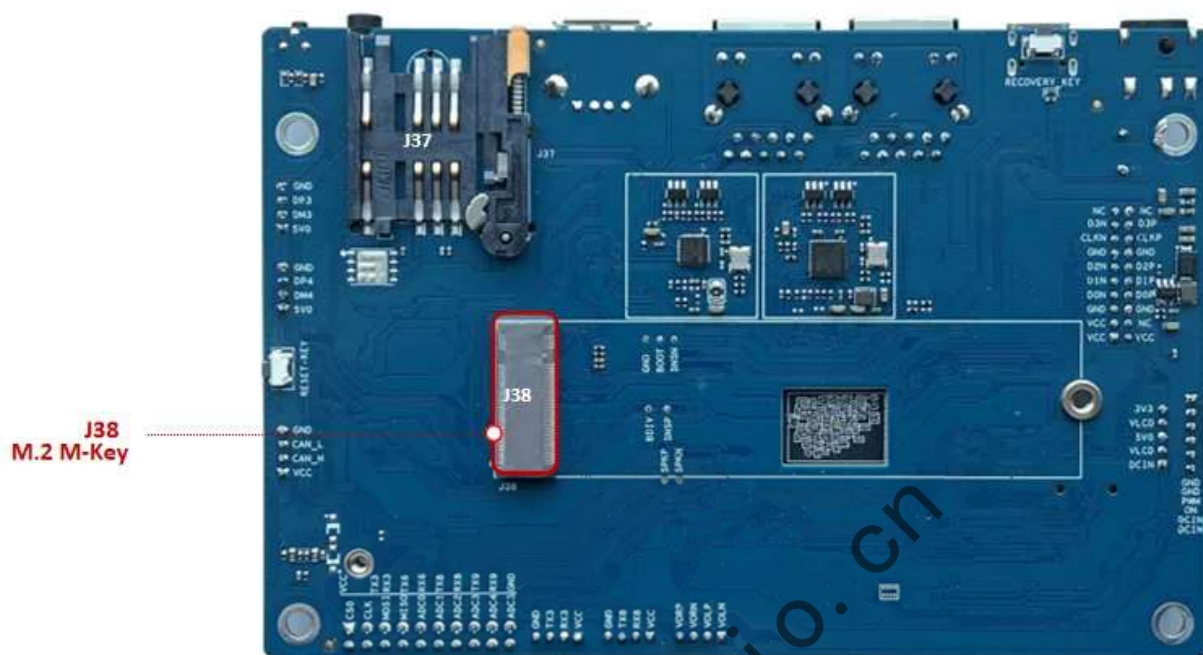
12.2 设置时间

设置RTC时间，命令如下：

```
1 root@rk3562:~# date -s '2024-8-22 15:30:00'
2 Tue Aug 22 15:30:00 UTC 2024
3 root@rk3562:~# hwclock -w
4 root@rk3562:~# hwclock -r
5 Tue Aug 22 15:30:05 2024 0.000000 seconds
```

13 硬盘

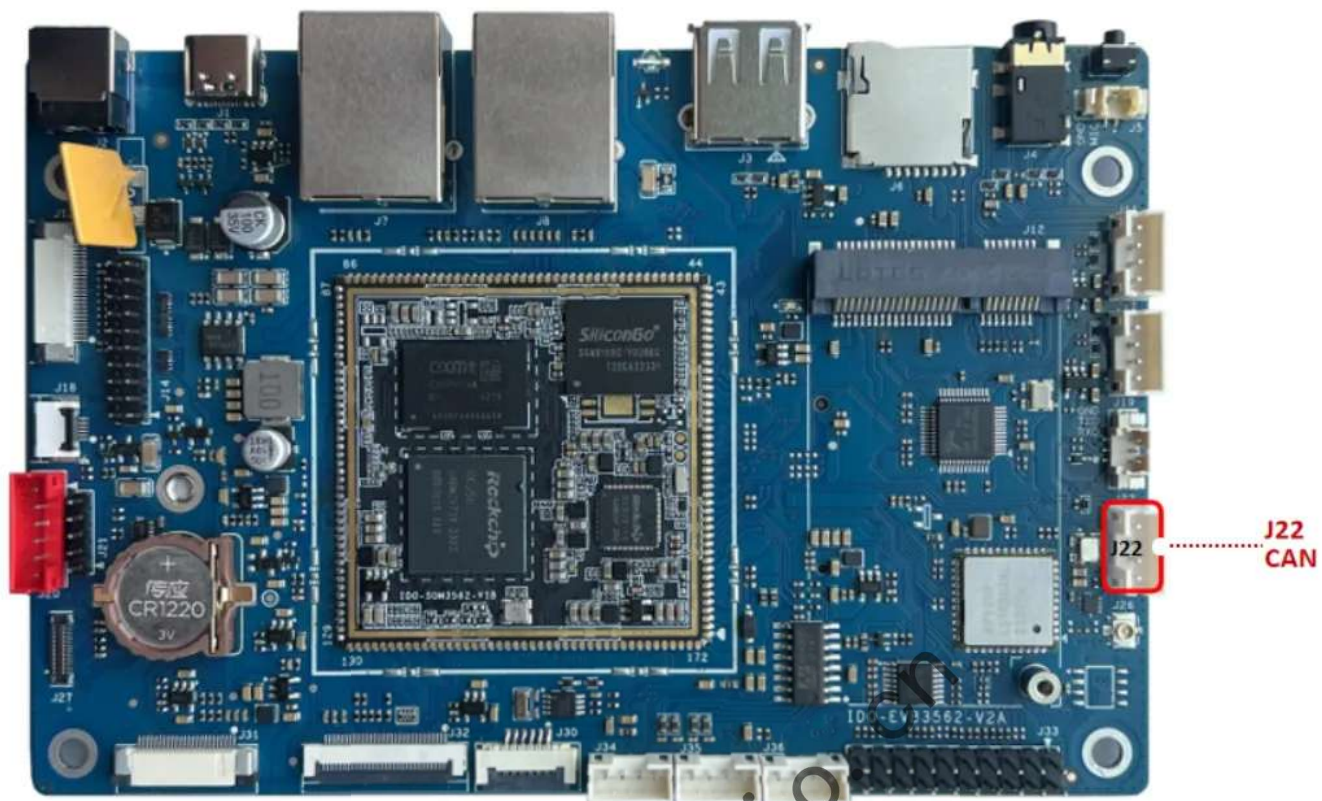
主板配置了一路硬盘接口（M.2），位于主板背面J38，如下图所示：



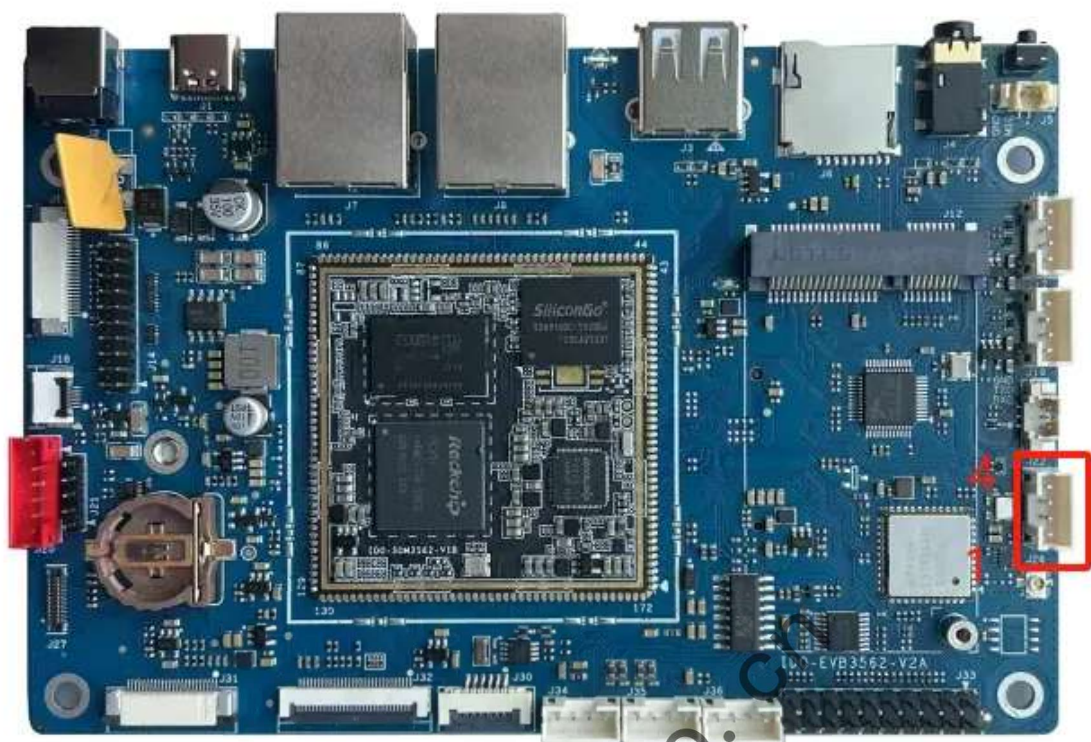
注意：PCIE功能目前和网卡冲突，如果需要此功能需要更改软硬件。

14 CAN

主板配置一路CAN接口，如下图所示：



序号	丝印	设备节点
1	J22	CAN0



引脚定义：

序号	定义	电平/V	说明
1	VCC	5V/3.3V	默认5V供电输出，可配置为3.3V
2	CAN0_H	/	CAN信号
3	CAN0_L	/	
4	GND	GND	电源地

CAN命令行测试命令：

▼

C

```
1  #关闭can0设备
2  ip link set can0 down
3
4  #普通can协议 (1.0)
5  ip link set can0 type can bitrate 125000 triple-sampling on
6
7  #查看can信息, 波特率等等
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 #can格式发送
17 cansend can0 123#1122334455667788
```

15 扩展接口

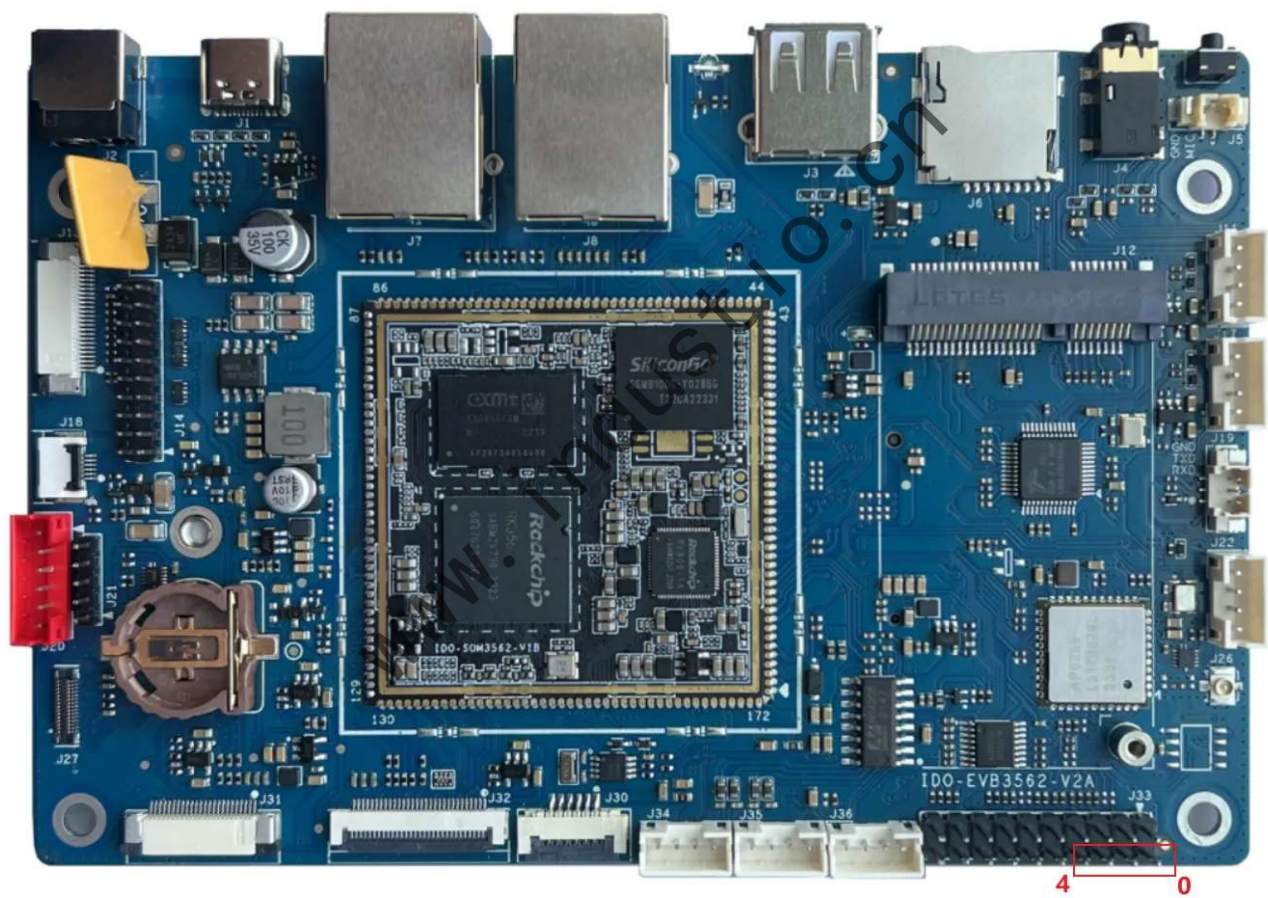
主板配置了一排扩展接口，位于J33，扩展接口的各个引脚功能，如下表所示：

pin	功能	pin	功能
1	5V输出	2	SPI0_CSN0_M0/PWM5_M0
3	UART3_TX	4	SPI0_CLK_M0/PWM0_M0
5	UART3_RX	6	SPI0_MOSI_M0/PWM1_M0
7	UART6_TX	8	SPI0_MISO_M0/PWM2_M0
9	UART6_RX	10	ADC0
11	UART8_TX	12	ADC1
13	UART8_RX	14	ADC2

15	I2C5_SCL_M0/UART 9_TX		16	ADC3
17	I2C5_SDA_M0/UART 9_RX		18	ADC4
19	GND		20	ADC5

15.1 GPIO

主板扩展接口有四路GPIO，如下图所示：



序号	管脚标号	GPIO 标号	LIBGPIO节点
1	SPI0_CSN0_M0/PW M5_M0	GPIO0_C2	gpiochip0 18
2	SPI0_CLK_M0/PWM0 _M0	GPIO0_C3	gpiochip0 19

3	SPI0_MOSI_M0/PWM 1_M0	GPIO0_C4	gpiochip0 20
4	SPI0_MISO_M0/PWM 2_M0	GPIO0_C5	gpiochip0 21

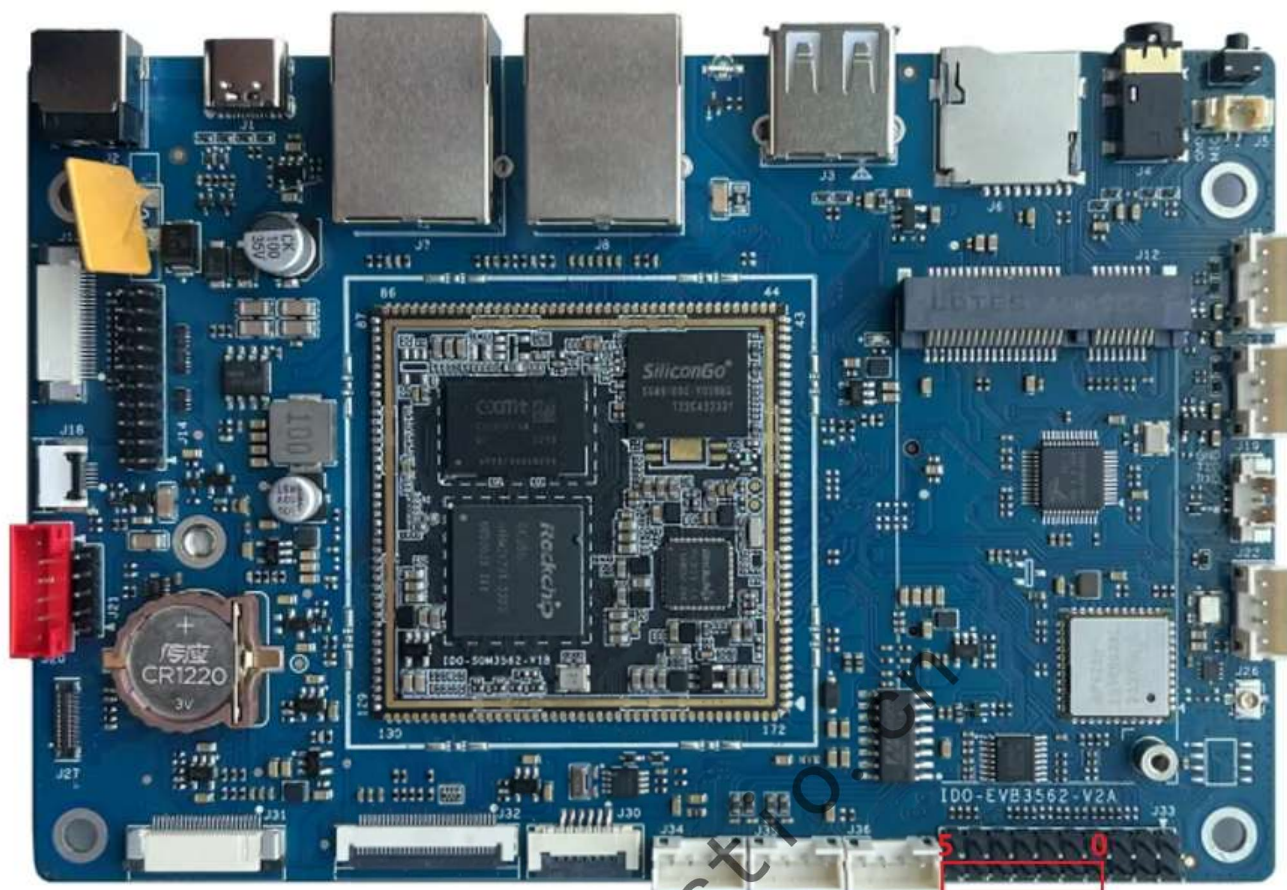
GPIO控制，根据上表【LIBGPIO节点】进行命令操作，以GPIO0_C2为例：

▼Shell

```
1  #输出：
2  #1为输出高电平(3.3V)，0为输出低电平(0V)。
3  root@rk3562:~# gpioset gpiochip0 18=1
4
5  root@rk3562:~# gpioset gpiochip0 18=0
6
7  #输入：
8  #0为外部输入低电平，1为外部输入高电平。
9  root@rk3562:~# gpioget gpiochip0 18
10 0
11 root@rk3562:~# gpioget gpiochip0 18
12 1
```

15.2 ADC

主板扩展接口有六路ADC，如下图所示：



扩展接口配置了6路ADC输入（参考电压为1.8v，支持10bit ADC采样）如下表所示：

名称	设备节点
ADCIN0	/sys/bus/iio/devices/iio:device1/in_voltage0_raw
ADCIN1	/sys/bus/iio/devices/iio:device1/in_voltage1_raw
ADCIN2	/sys/bus/iio/devices/iio:device1/in_voltage2_raw
ADCIN3	/sys/bus/iio/devices/iio:device1/in_voltage3_raw
ADCIN4	/sys/bus/iio/devices/iio:device1/in_voltage4_raw
ADCIN5	/sys/bus/iio/devices/iio:device1/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ADCIN5为例：

硬件连接：ADC5外接0V。

1.读取in_voltage5_raw的值，命令如下：

▼Shell

```
1 root@rk3562:~# cat /sys/bus/iio/devices/iio:device1/in_voltage5_raw
2 0
```

2.计算ADC5的输入电压V

$$V = (\text{voltage_raw}/1024)*1.8V = (0/1024)*1.8V = 0V$$

硬件连接：ADC5外接1.8V。

1.读取in_voltage5_raw的值，命令如下：

▼Shell

```
1 root@rk3562:~# cat /sys/bus/iio/devices/iio:device1/in_voltage5_raw
2 1020
```

2.计算ADC5的输入电压：

$$V = (\text{voltage_raw}/1024)*1.8V = (1020/1024)*1.8V = 1.79V$$

注意：ADC脚默认悬空状态，需要外部提供0~1.8v电压。

16 系统使用说明

16.1 QT5

16.1.1 3d测试

▼Bash

```
1 root@rk3562:/# chmod a+x ./simple-cpp
2 root@rk3562:/# ./simple-cpp
3 QStandardPaths: runtime directory '/var/run' is not a directory, but a symbolic link to a directory permissions 0755 owned by UID 0 GID 0
4 arm_release_ver: g24p0-00eac0, rk_so_ver: 7
```

如图所示：



16.2 glmark2

```

1  #设置cpu性能模式
2  root@rk3562:/# echo performance > /sys/devices/system/cpu/cpufreq/policy0/
   scaling_governor
3
4  #测试系统性能
5  root@rk3562:/# glmark2-es2-wayland
6  arm_release_ver: g24p0-00eac0, rk_so_ver: 7
7  =====
8      glmark2 2023.01
9  =====
10     OpenGL Information
11     GL_VENDOR:      ARM
12     GL_RENDERER:    Mali-G52
13     GL_VERSION:     OpenGL ES 3.2 v1.g24p0-00eac0.7624deb1338aa462bb011099
   003f89a4
14     Surface Config: buf=32 r=8 g=8 b=8 a=8 depth=24 stencil=0 samples=0
15     Surface Size:   800x600 windowed
16     =====
17  [build] use-vbo=false: FPS: 1002 FrameTime: 0.998 ms
18  [build] use-vbo=true: FPS: 1354 FrameTime: 0.739 ms
19  [texture] texture-filter=nearest: FPS: 1767 FrameTime: 0.566 ms
20  [texture] texture-filter=linear: FPS: 1750 FrameTime: 0.572 ms
21  [texture] texture-filter=mipmap: FPS: 1766 FrameTime: 0.567 ms
22  [shading] shading=gouraud: FPS: 1008 FrameTime: 0.992 ms
23  [shading] shading=blinn-phong-inf: FPS: 1238 FrameTime: 0.808 ms
24  [shading] shading=phong: FPS: 1044 FrameTime: 0.959 ms
25  [shading] shading=cel: FPS: 1021 FrameTime: 0.980 ms
26  [bump] bump-render=high-poly: FPS: 404 FrameTime: 2.479 ms
27  [bump] bump-render=normals: FPS: 1792 FrameTime: 0.558 ms
28  [bump] bump-render=height: FPS: 1717 FrameTime: 0.582 ms
29  [effect2d] kernel=0,1,0;1,-4,1;0,1,0,: FPS: 701 FrameTime: 1.428 ms
30  [effect2d] kernel=1,1,1,1,1;1,1,1,1,1;1,1,1,1,1,: FPS: 237 FrameTime: 4.22
   8 ms
31  [pulsar] light=false:quads=5:texture=false: FPS: 1680 FrameTime: 0.596 ms
32  [desktop] blur-radius=5:effect=blur:passes=1:separable=true:windows=4: FP
   S: 280 FrameTime: 3.581 ms
33  [desktop] effect=shadow:windows=4: FPS: 958 FrameTime: 1.044 ms
34  [buffer] columns=200:interleave=false:update-dispersion=0.9:update-fractio
   n=0.5:update-method=map: FPS: 180 FrameTime: 5.557 ms
35  [buffer] columns=200:interleave=false:update-dispersion=0.9:update-fractio
   n=0.5:update-method=subdata: FPS: 179 FrameTime: 5.610 ms
36  [buffer] columns=200:interleave=true:update-dispersion=0.9:update-fraction
   =0.5:update-method=map: FPS: 268 FrameTime: 3.740 ms
37  [ideas] speed=duration: FPS: 496 FrameTime: 2.017 ms
38  [jellyfish] <default>: FPS: 630 FrameTime: 1.590 ms

```

```

39 [terrain] <default>: FPS: 44 FrameTime: 22.775 ms
40 [shadow] <default>: FPS: 424 FrameTime: 2.361 ms
41 [refract] <default>: FPS: 91 FrameTime: 11.039 ms
42 [conditionals] fragment-steps=0:vertex-steps=0: FPS: 1661 FrameTime: 0.60
43 2 ms
44 [conditionals] fragment-steps=5:vertex-steps=0: FPS: 1257 FrameTime: 0.79
45 6 ms
46 [conditionals] fragment-steps=0:vertex-steps=5: FPS: 1657 FrameTime: 0.60
47 4 ms
48 [function] fragment-complexity=low:fragment-steps=5: FPS: 1430 FrameTime:
49 0.700 ms
50 [function] fragment-complexity=medium:fragment-steps=5: FPS: 1079 FrameTim
51 e: 0.927 ms
52 [loop] fragment-loop=false:fragment-steps=5:vertex-steps=5: FPS: 1420 Fram
eTime: 0.704 ms
[loop] fragment-steps=5:fragment-uniform=false:vertex-steps=5: FPS: 1431 F
rameTime: 0.699 ms
[loop] fragment-steps=5:fragment-uniform=true:vertex-steps=5: FPS: 1252 Fr
ameTime: 0.799 ms
=====
glmark2 Score: 1005
=====

```

如图所示：

