IDO-EVB7608-V1 Linux使用手册

IDO-EVB7608-V1

Linux使用手册

深圳触觉智能科技有限公司

www.industio.cn

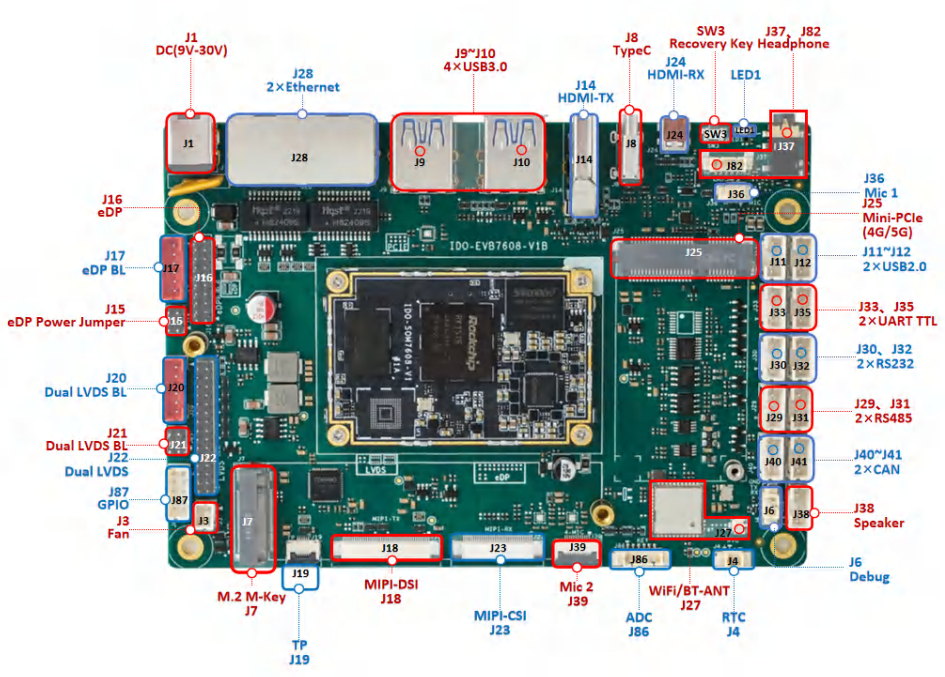
文档修订历史

版本	PCBA版本	修订内容	修订	审核	日期
V1.0	V1B	创建文档	MHK	IDO	2024/11/27

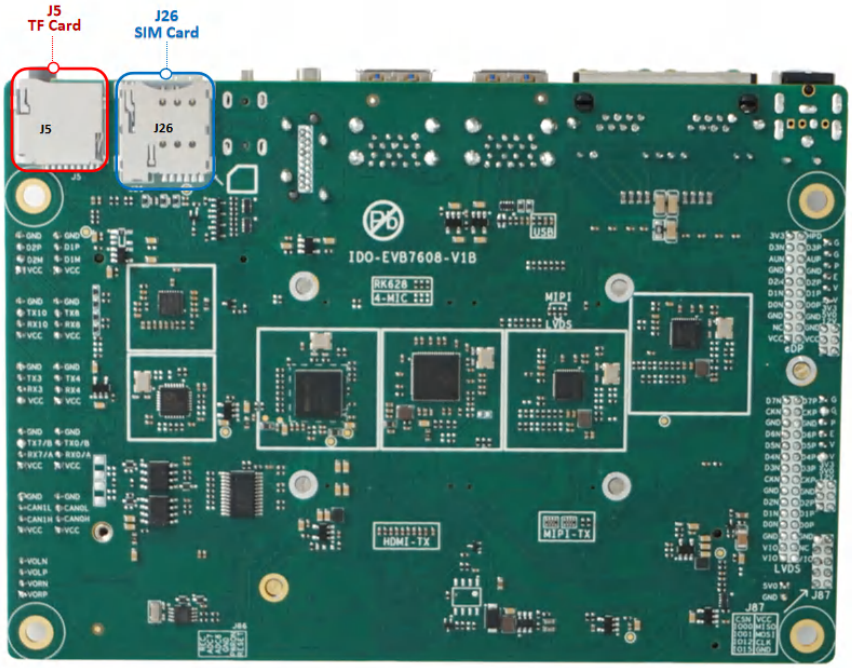
V1.1	V1B	增加ubuntu系统使用描述	TWX	IDO	2025/04/24
V1.2	V1B	修正音频部分声卡编号； 修正QT demo测试方法；	TWX	IDO	2025/05/06
V1.1	V1B	更新摄像头转接板图片	LYT	IDO	2025/10/10

1. 硬件资源概况

1.1. 主板照片



IDO–EVB7608–V1B 正面实物图



IDO-EVB7608-V1B 背面实物图

1.2. 硬件资源

序号	名称	描述
1	内核版本	Linux 6.1.75
2	系统版本	Buildroot2024/Debian12/Ubuntu22.04
3	内存	LPDDR4（4/6G）
4	存储	eMMC5.1（32/64/128GB）
5	供电	DC接口12V@2A
6	显示	DP ×1 HDMI ×1 Dual LVDS ×1 eDP ×1 MIPI ×1
7	触摸	I2C-TP ×1

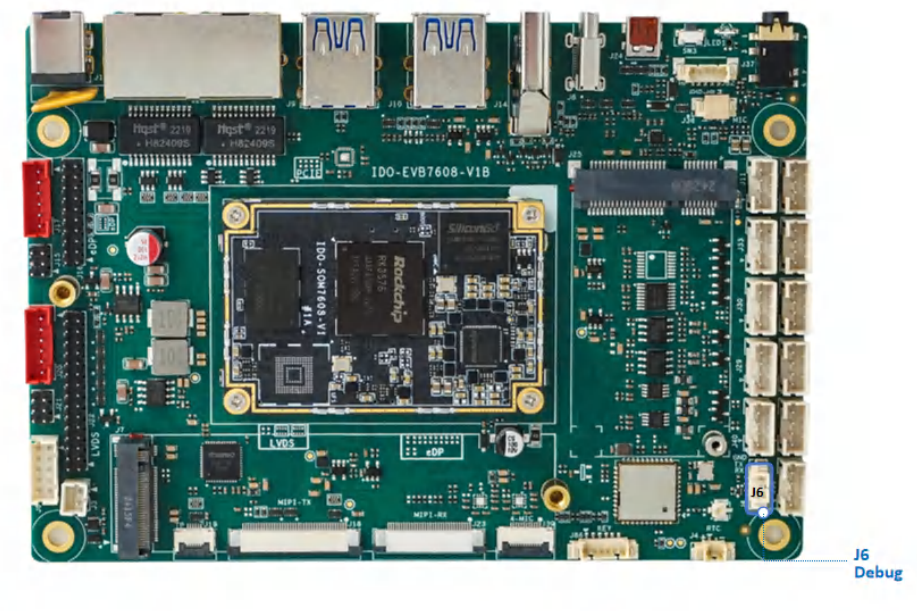
8	HDMI-RX	Micro-HDMI
9	USB OTG	USB OTG Type-C
10	USB HOST	USB3.0 HOST(Type-A) ×4 USB2.0 HOST(PH2.0) ×2
11	PCIe	PCIe2.1 NVME硬盘 ×1
12	TF Card	SDIO3.0 TF Card ×1
13	以太网	千兆以太网 ×2
14	WIFI/BT	AP6256
15	扬声器	PH2.0-4P(4ohm 3W)
16	耳机	CTIA标准四节耳机座
17	MIC	驻极体麦克风 ×1 PDM 阵列麦克风×1
18	Camera	OV13850/IMX415
19	UART	TTL ×2 RS232 ×2 RS485 ×2
20	DEBUG UART	TTL
21	CAN口	CAN ×2
22	RTC	HYM8563 ×1
23	系统指示灯	系统指示灯 ×1
24	4G/5G	EC20 4G/RG200U 5G模块
25	按键	REC ×1
26	Fan	5V PH2.02P ×1
27	GPIO	GPIO预留 ×8
28	ADC	ADC ×2

IDO-EVB7608-V1B-Debian系统

2. 功能及接口使用方法

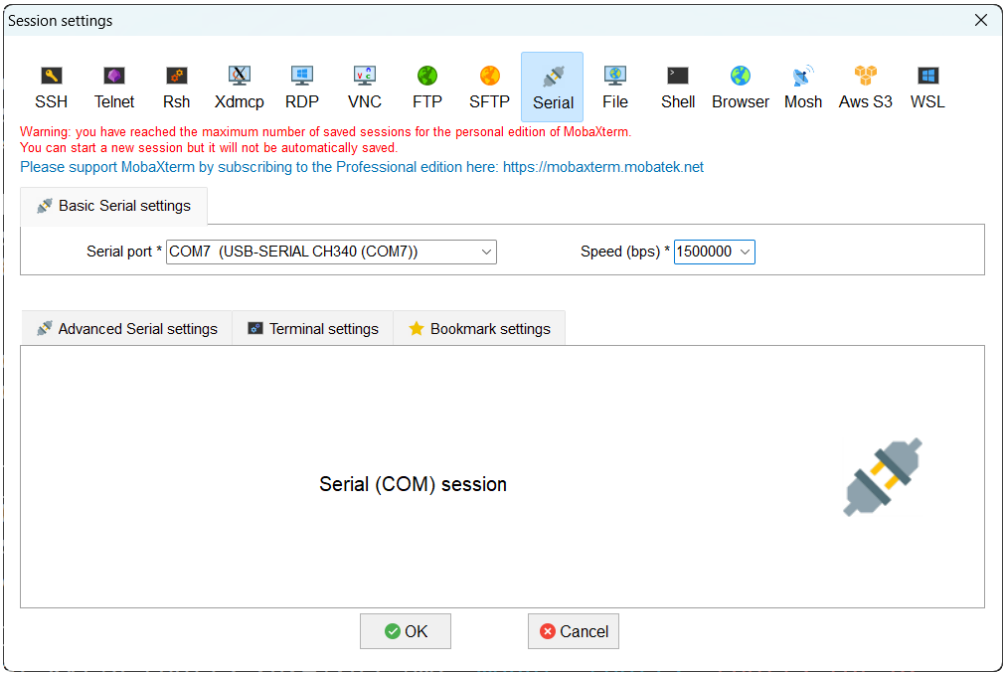
2.1. DEBUG UART

DEBUG UART位置J6，如下图所示：



使用USB Type-C数据线，连接PC端的USB接口。系统会识别到一个“USB-SERIAL CH340”端口设备。

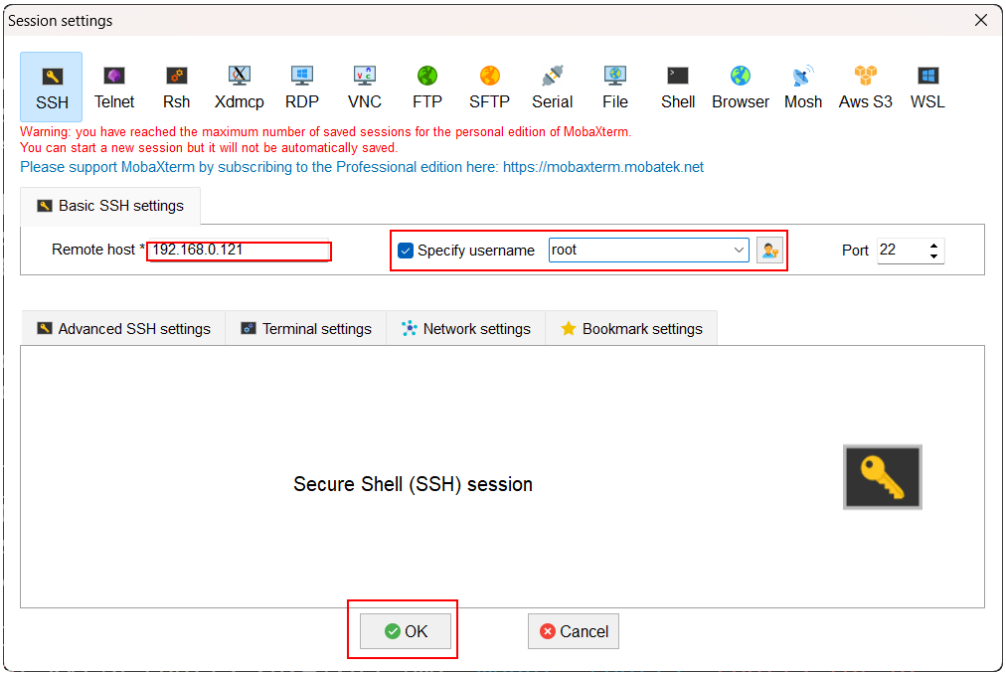
使用调试软件（MobaXterm、putty）等，以MobaXterm为例子，设置参数如下：



SSH登录

在知道IP的前提下，默认可以通过SSH登录进系统。

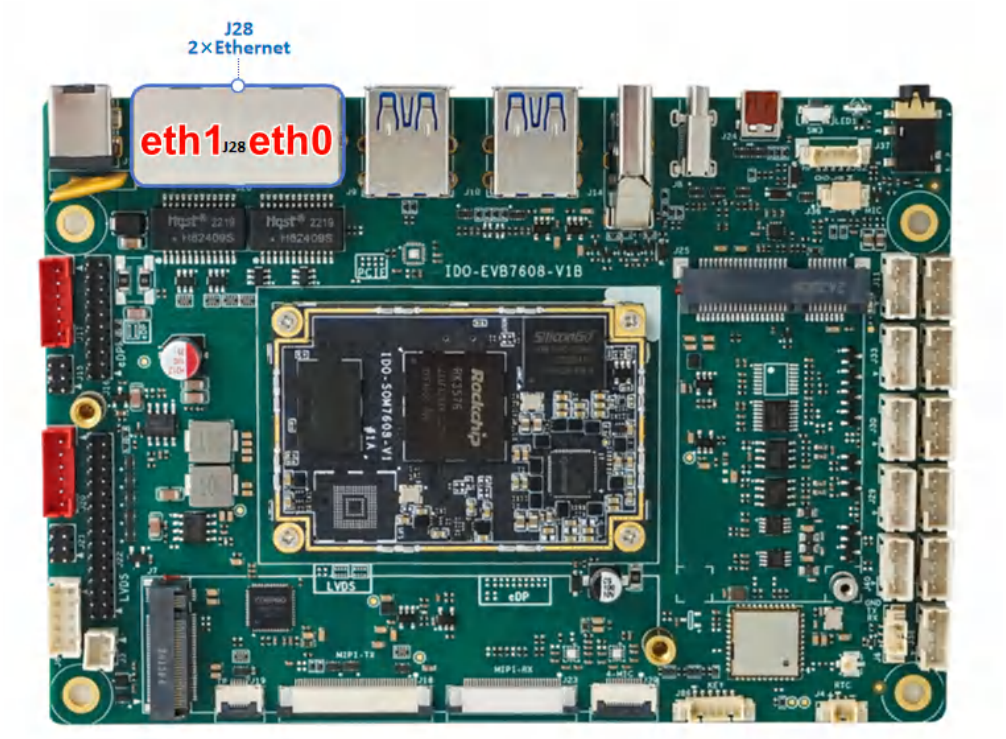
SSH登录账号密码：root/123456



2.2. 网络

2.2.1. 以太网

主板有两路千兆以太网接口位置J28，如下图所示：



设备节点分别为eth0和eth1，以太网接口默认支持DHCP，只需要将以太网接口连接路由器即可为主板动态分配 IP 地址。网络正常连接图标，如下图所示：

Ethernet动态IP设置：

PowerShell

```
#eth0
root@linaro-alip:/# ifconfig eth0
eth0: 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::1b83:a156:cb16:c6b0  prefixlen 64  scopeid 0x20<link>
        ether fa:c6:ba:c7:0c:60  txqueuelen 1000  (Ethernet)
        RX packets 141  bytes 15129 (14.7 KiB)
        RX errors 0  dropped 4  overruns 0  frame 0
        TX packets 89  bytes 9306 (9.0 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 121

#eth1
root@linaro-alip:/# ifconfig eth1
eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.91  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::8992:70c3:f104:e0be  prefixlen 64  scopeid 0x20<link>
        ether fe:c6:ba:c7:0c:60  txqueuelen 1000  (Ethernet)
        RX packets 293  bytes 30522 (29.8 KiB)
        RX errors 0  dropped 6  overruns 0  frame 0
        TX packets 151  bytes 15448 (15.0 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 123
```

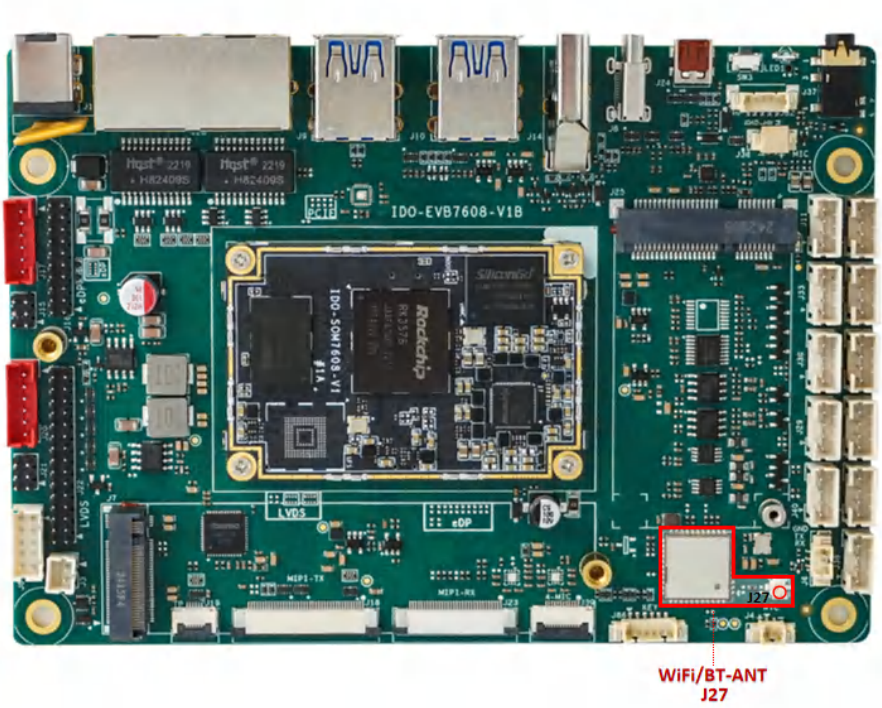
Ethernet静态IP设置：

```PowerShell
root@linaro-alip:/# touch /etc/netplan/01-netcfg.yaml
root@linaro-alip:/# vim /etc/netplan/01-netcfg.yaml
root@linaro-alip:/# cat  /etc/netplan/01-netcfg.yaml
network:
        version: 2
        ethernets:
                eth0:
                        dhcp4: no
                        addresses: [192.168.3.121/24]
                        nameservers:
                                addresses: [192.168.3.1,8.8.8.8, 114.114.114.114]
                        routes:
                                - to: 0.0.0.0/0
                                  via: 192.168.3.1
                                  metric: 100
#立即生效
root@linaro-alip:/# netplan apply
Cannot call openvswitch: ovsdb-server.service is not running.

root@linaro-alip:/# ifconfig eth0
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.3.121  netmask 255.255.255.0  broadcast 192.168.3.255
        inet6 fe80::1b83:a156:cb16:c6b0  prefixlen 64  scopeid 0x20<link>
        inet6 fe80::f8c6:baff:fec7:c60  prefixlen 64  scopeid 0x20<link>
        ether fa:c6:ba:c7:0c:60  txqueuelen 1000  (Ethernet)
        RX packets 4203  bytes 417790 (407.9 KiB)
        RX errors 0  dropped 164  overruns 0  frame 0
        TX packets 392  bytes 32299 (31.5 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 121
```

设备断电重启，此静态IP设置仍然生效。

2.2.2. WiFi

使用WiFi/蓝牙时，需要连接天线以获得良好的信号，WiFi模块和天线座子J27，如下图所示：



命令行可以使用nmcli工具连接wifi热点：

Shell

```
#查看网络接口状态:
root@linaro-alip:~# nmcli device
#使用以下命令查看当前可用的 WiFi 网络:
root@linaro-alip:~# nmcli device wifi list
#连接到 WiFi 网络:
root@linaro-alip:~# nmcli device wifi connect 账号 password 密码
#确认连接状态:
root@linaro-alip:~# nmcli connection show --active
```

查看wlan0的IP地址，确认连接成功：

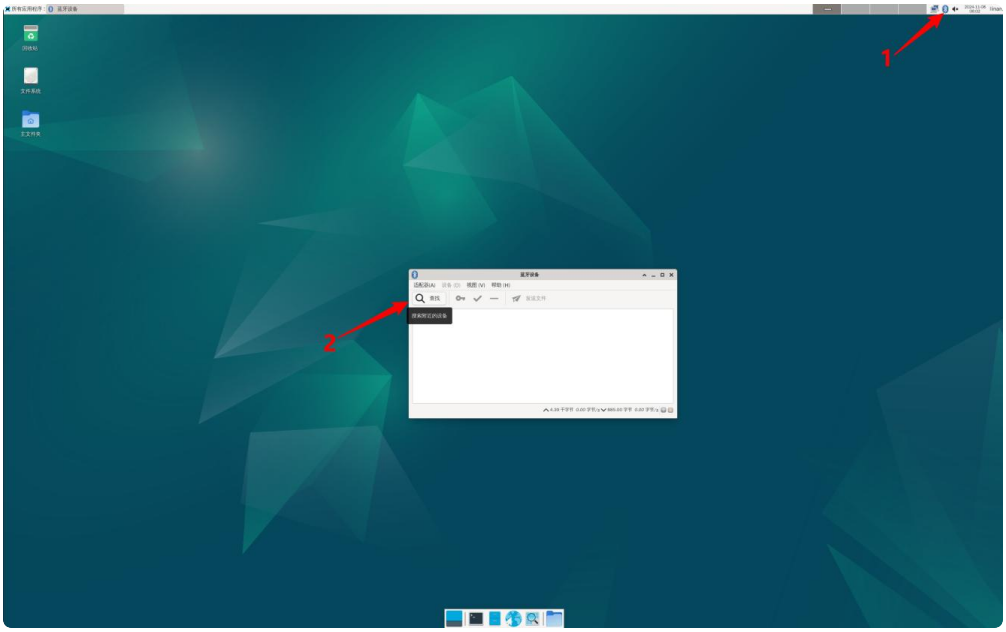
```shell
root@linaro-alip:/# ifconfig wlan0
wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.118  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::1036:80b4:5082:7440  prefixlen 64  scopeid 0x20<link>
        ether c0:f5:35:12:ad:a2  txqueuelen 1000  (Ethernet)
        RX packets 170  bytes 22149 (21.6 KiB)
        RX errors 0  dropped 5  overruns 0  frame 0
        TX packets 24  bytes 2683 (2.6 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@linaro-alip:/# ping www.baidu.com -I wlan0
PING www.wshifen.com (103.235.47.188) from 192.168.0.118 wlan0: 56(84) bytes of data.
64 bytes from 103.235.47.188 (103.235.47.188): icmp_seq=1 ttl=45 time=274 ms
64 bytes from 103.235.47.188 (103.235.47.188): icmp_seq=2 ttl=45 time=299 ms
64 bytes from 103.235.47.188 (103.235.47.188): icmp_seq=3 ttl=45 time=320 ms
64 bytes from 103.235.47.188 (103.235.47.188): icmp_seq=4 ttl=45 time=241 ms
```

2.2.3. Bluetooth

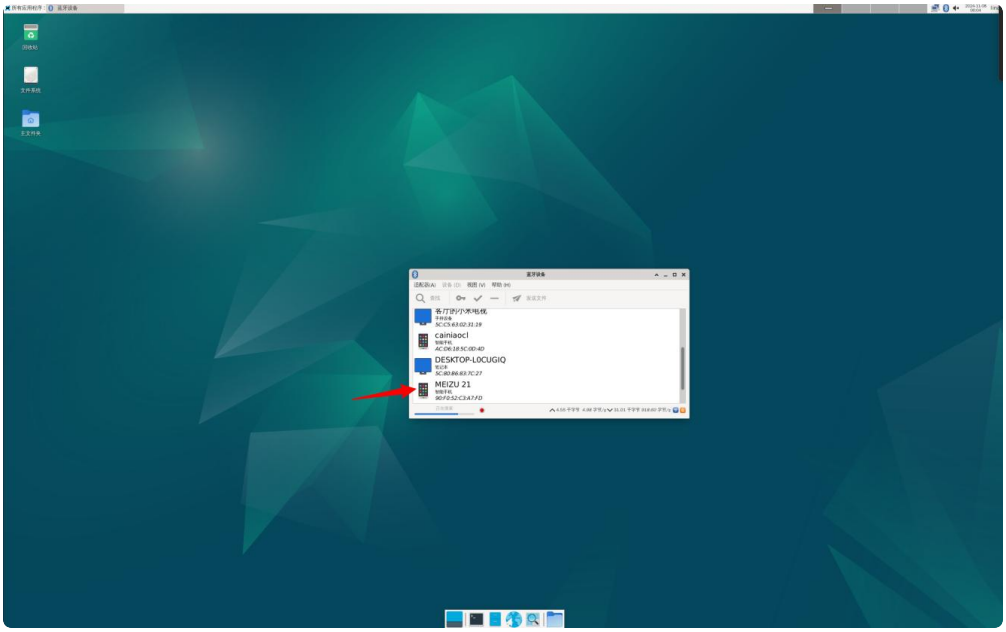
桌面连接：

点击【菜单】->【设置】->【已连接的设备】->【与新设备配对】

即可扫描到附近的蓝牙设备，选择需要连接的设备即可根据配对信息进行连接，如下图所示：



选择你要连接的蓝牙，点击右键连接即可：

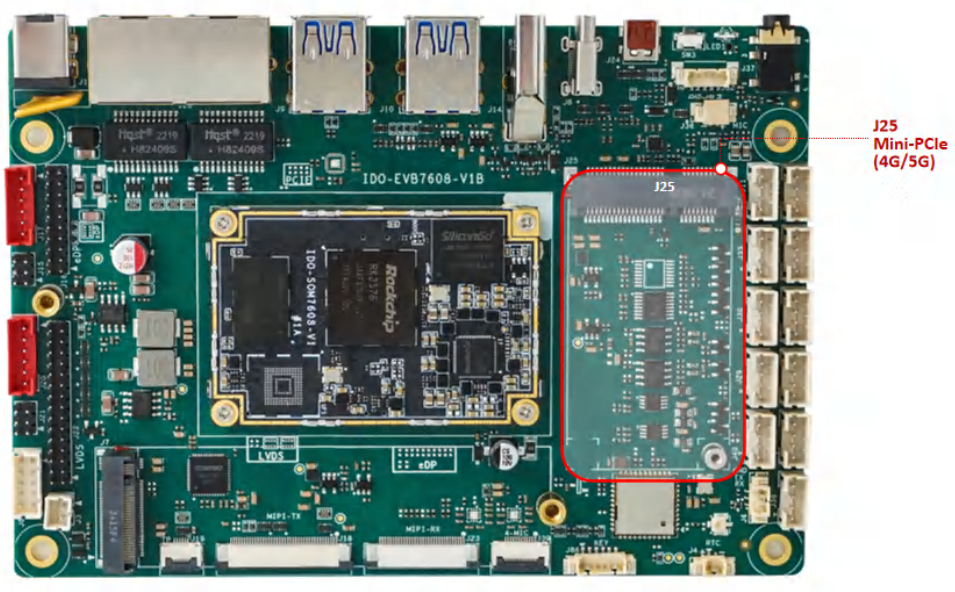


使用命令行操作

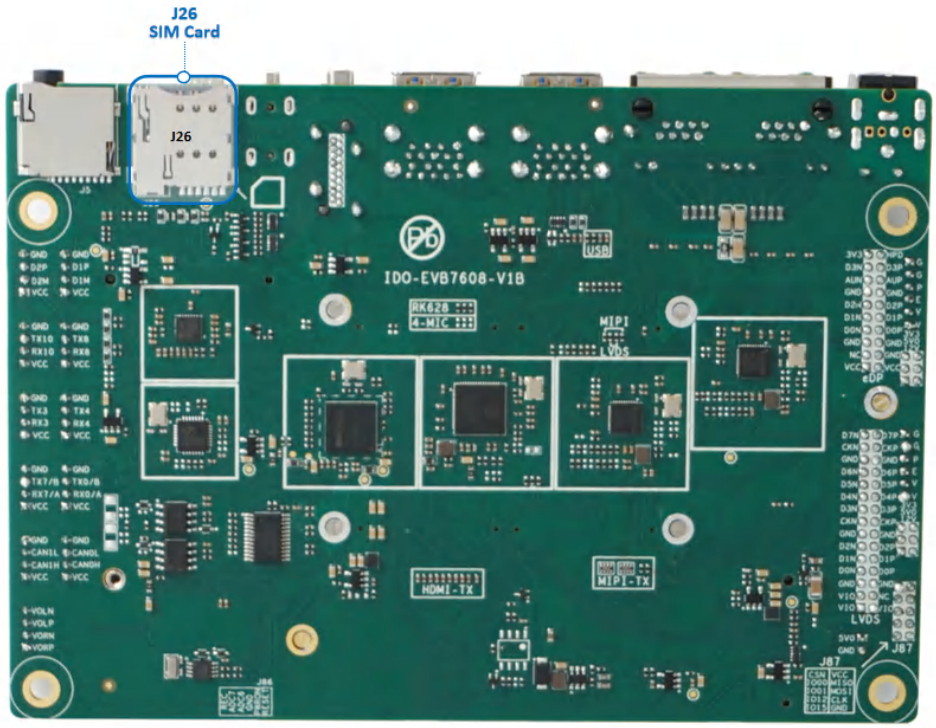
```shell
#打开蓝牙
root@linaro-alip:/# bluetoothctl power on
#扫描蓝牙
root@linaro-alip:/# bluetoothctl scan on
#查看蓝牙设备
root@linaro-alip:/# bluetoothctl devices
#信任蓝牙设备
root@linaro-alip:/# bluetoothctl trust 7C:C1:80:09:DD:6C
#蓝牙配对
root@linaro-alip:/# bluetoothctl pair 7C:C1:80:09:DD:6C
#连接蓝牙
root@linaro-alip:/# bluetoothctl connect 7C:C1:80:09:DD:6C
```

2.2.4. 4G/5G

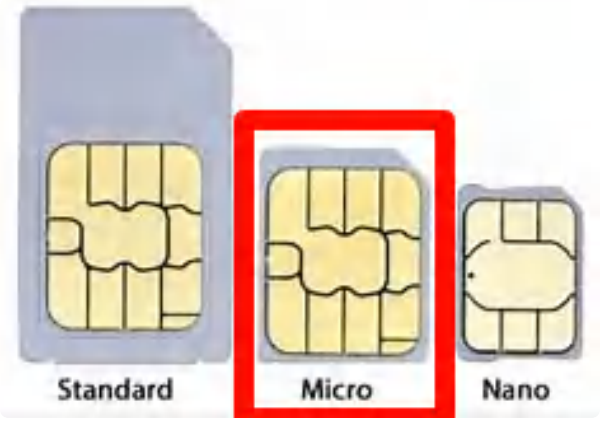
主板内置Mini PCIe 座子扩展 4G/5G模块，4G通信模块适配移远EC20、EC25等通用模组。5G通信模块适配移远RG200U-CN。模块安装位J25，如下图所示：



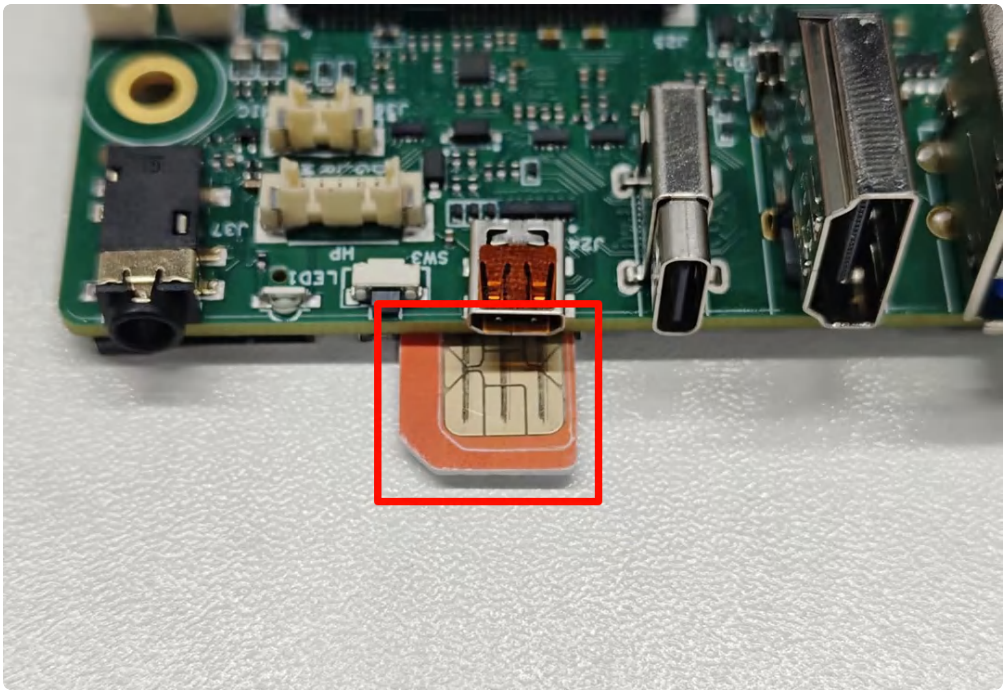
模块需要接上天线以确保获得更好的信号质量，SIM卡安装位J26位于主板背面，如下图所示：



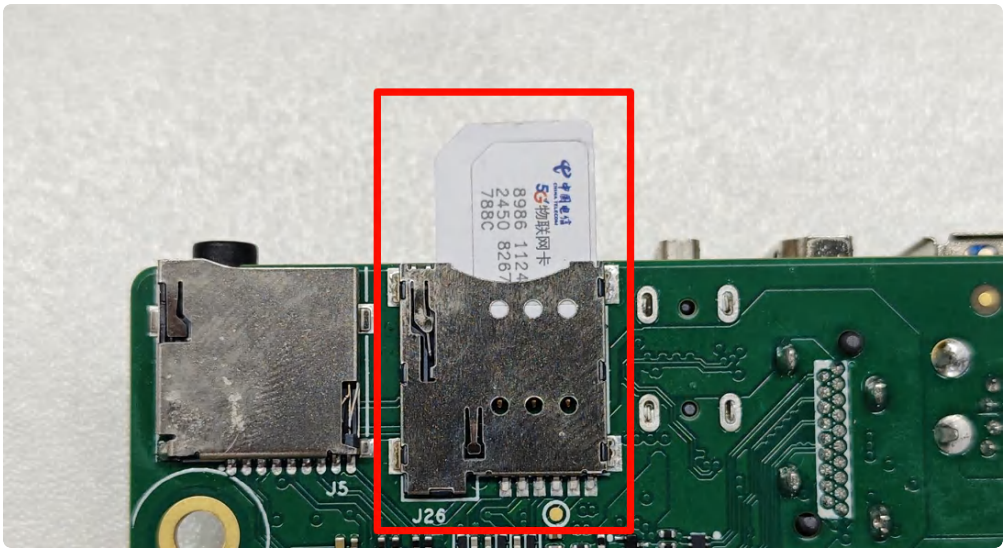
使用Micro尺寸SIM卡，如下图所示：



主板朝上时，SIM卡触点朝上，缺口朝外安装，如下图所示：



主板朝下时，SIM卡触点朝下缺口朝外安装，如下图所示：



4G使用 `ifconfig wwan0` 命令可查看到相关网络信息：

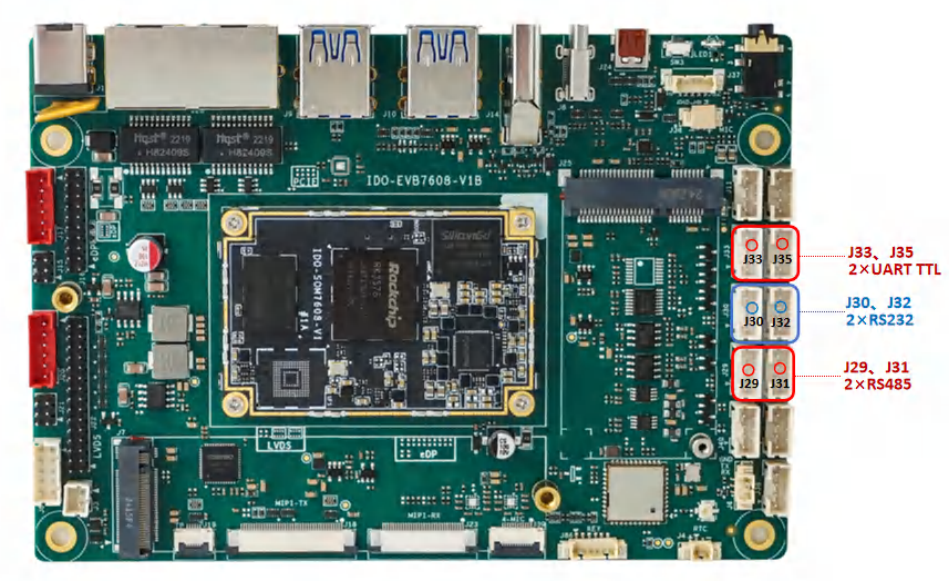
```
#拨号
root@linaro-alip:~# quectel-CM &
#查看网络
root@linaro-alip:~# ifconfig wwan0
wwan0: flags=193<UP,RUNNING,NOARP>  mtu 1500
        inet 10.101.61.51  netmask 255.255.255.248
        inet6 fe80::fc:f6ff:fe8d:bab6  prefixlen 64  scopeid 0x20<link>
        ether 02:fc:f6:8d:ba:b6  txqueuelen 1000  (Ethernet)
        RX packets 42  bytes 7013 (6.8 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 57  bytes 4608 (4.5 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

#测试通信
root@linaro-alip:~# ping www.baidu.com -I wwan0
```

5G使用 `ifconfig usb0` 命令可查看到相关网络信息:

```
#拨号
root@linaro-alip:~# quectel-CM &
#查看网络
root@linaro-alip:~# ifconfig usb0
usb0: flags=193<UP,RUNNING,NOARP>  mtu 1500
        inet 10.101.61.51  netmask 255.255.255.248
        inet6 fe80::fc:f6ff:fe8d:bab6  prefixlen 64  scopeid 0x20<link>
        ether 02:fc:f6:8d:ba:b6  txqueuelen 1000  (Ethernet)
        RX packets 42  bytes 7013 (6.8 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 57  bytes 4608 (4.5 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

#测试通信
root@linaro-alip:~# ping www.baidu.com -I usb0
```

2.3. UART

IDO-EVB7608-V1主板一共引出6路UART（不含DEBUG UART），如下图所示：



UART设备节点列表如下：

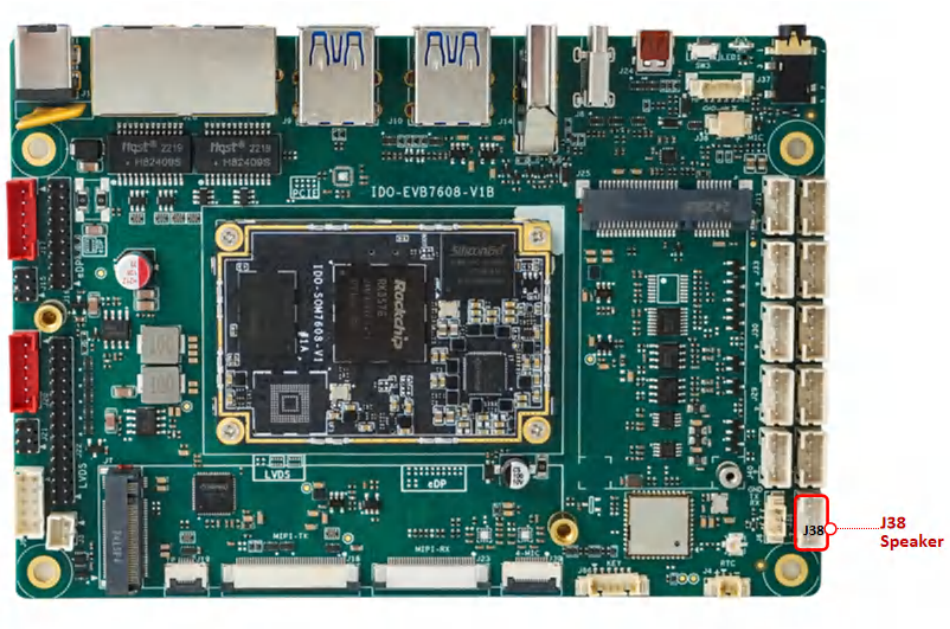
序号	默认电平类型	可改电平类型	设备节点
J33	TTL	可改RS232	/dev/ttyS8
J35	TTL	可改RS232	/dev/ttyS10
J30	RS232	可改TTL	/dev/ttyS4
J32	RS232	可改TTL	/dev/ttyS3
J29	RS485	可改TTL	/dev/ttyS1
J31	RS485	可改TTL	/dev/ttyS7

```
#microcom安装命令
root@linaro-alip:~# apt-get install microcom
```

2.4. 声音

2.4.1. 喇叭

喇叭接口为PH2.0-4P连接器J38，如下图所示：



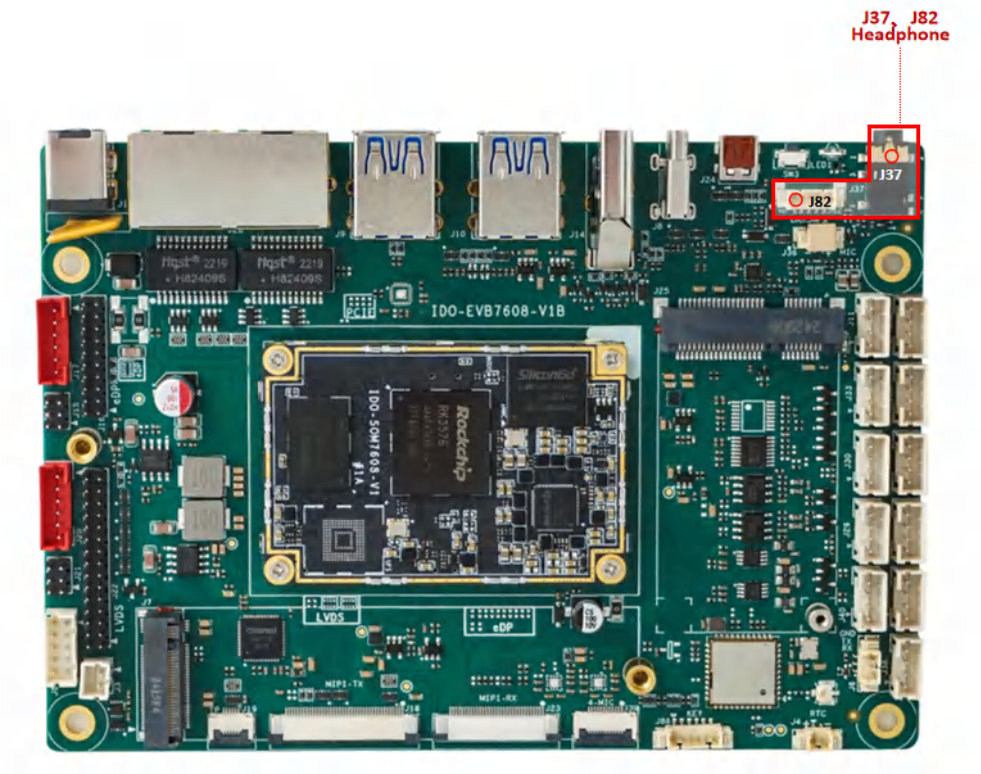
- 支持双声道，每个声道支持4Ω 3W输出

Plain Text

```
#查看声卡
root@linaro-alip:/# aplay -l
**** List of PLAYBACK Hardware Devices ****
card 1: rockchipes8388 [rockchip-es8388], device 0: dailink-multicodecs ES8323 HiFi-0 [dailink-multicodecs ES8323 HiFi-0]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
card 2: rockchiphdmi [rockchip-hdmi], device 0: rockchip-hdmi i2s-hifi-0 [rockchip-hdmi i2s-hifi-0]
  Subdevices: 1/1
  Subdevice #0: subdevice #0



#音频播放到声卡
root@linaro-alip:/# aplay -D hw:1,0 xihuangni.wav
Playing WAVE 'xihuangni.wav' : Signed 16 bit Little Endian, Rate 44100 Hz, Stereo

#音频播放到HDMI
root@linaro-alip:/# aplay -D hw:2,0 xihuangni.wav
Playing WAVE 'xihuangni.wav' : Signed 16 bit Little Endian, Rate 44100 Hz, Stereo
```

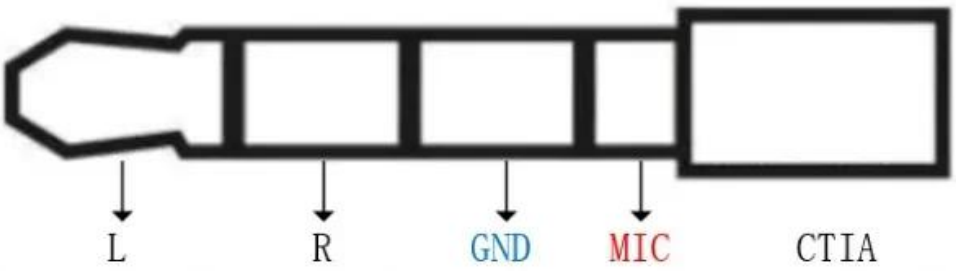
2.4.2. 耳机

主板支持1路3.5mm四节耳机座（CTIA）J37，和1路MX1.25T-5P耳机座子并用，用户可根据需求选用其中1路耳机座子使用 ，如下图所示：



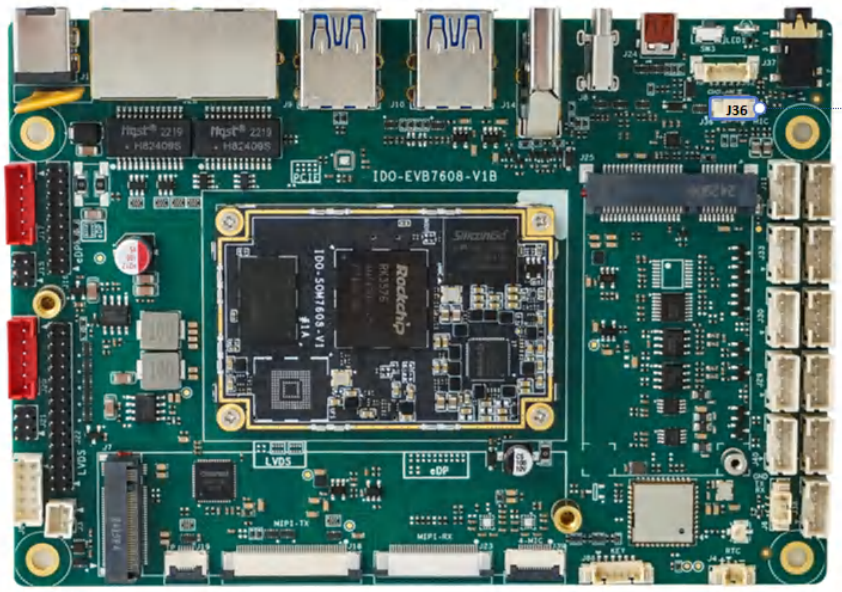
- 支持耳机检测
- 支持耳机录音

CTIA标准四段式耳机，定义如下：



2.4.3. Mic

MX1.25T-2P麦克风接口J36，如下图所示：


J36
Mic 1
J36
IDO-EVB7608-V1B
IDO-SOM7608-V1
LVDS
eDP
MIPI-TX
MIPI-RX

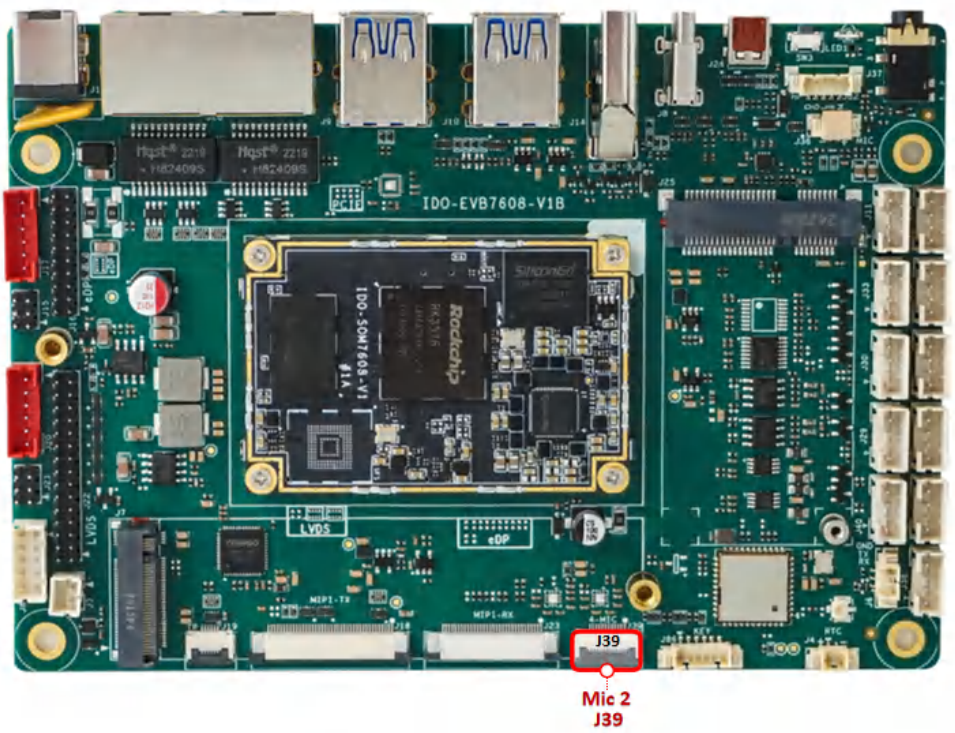
```
#查看录音设备
root@linaro-alip:/# arecord -l
**** List of CAPTURE Hardware Devices ****
card 0: rockchiprk628hd [rockchip,rk628hdmiin], device 0: rockchip,rk628hdmiin i2s-hifi-0 [rockchip,rk628hdmiin i2s-hifi-0]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
card 1: rockchipes8388 [rockchip-es8388], device 0: dailink-multicodecs ES8323 HiFi-0 [dailink-multicodecs ES8323 HiFi-0]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
card 2: rockchiphdmi [rockchip-hdmi], device 0: rockchip-hdmi i2s-hifi-0 [rockchip-hdmi i2s-hifi-0]
  Subdevices: 1/1
  Subdevice #0: subdevice #0

#捕获音量设置(不要设置最大值):
root@linaro-alip:~# amixer -c 1 cset numid=50 100
numid=50,iface=MIXER,name='Capture Digital Volume'
  ; type=INTEGER,access=rw---R--,values=2,min=0,max=192,step=0
  : values=100,100
  | dBscale-min=-96.00dB,step=0.50dB,mute=1

root@linaro-alip:~# amixer -c 1 cget numid=52
numid=52,iface=MIXER,name='Left Channel Capture Volume'
  ; type=INTEGER,access=rw---R--,values=1,min=0,max=8,step=0
  : values=0
  | dBscale-min=0.00dB,step=3.00dB,mute=0

root@linaro-alip:~# amixer -c 1 cget numid=53
numid=53,iface=MIXER,name='Right Channel Capture Volume'
  ; type=INTEGER,access=rw---R--,values=1,min=0,max=8,step=0
  : values=0
  | dBscale-min=0.00dB,step=3.00dB,mute=0

自动增益控制(打开):
root@linaro-alip:~# amixer -c 1 cset numid=41 3
numid=41,iface=MIXER,name='ALC Capture Function'
  ; type=ENUMERATED,access=rw------,values=1,items=4
  ; Item #0 'Off'
  ; Item #1 'Right'
  ; Item #2 'Left'
  ; Item #3 'Stereo'
  : values=3
```

```
调整目标音量：
root@linaro-alip:~# amixer -c 1 cset numid=38 8
numid=38,iface=MIXER,name='ALC Capture Target Volume'
  ; type=INTEGER,access=rw------,values=1,min=0,max=15,step=0
  : values=8
设置噪声门限值：
root@linaro-alip:~# amixer -c 1 cset numid=46 15
numid=46,iface=MIXER,name='ALC Capture NG Threshold'
  ; type=INTEGER,access=rw------,values=1,min=0,max=31,step=0
  : values=15
确保捕获静音未开启：
root@linaro-alip:~# amixer -c 1 cset numid=51 off
numid=51,iface=MIXER,name='Capture Mute'
  ; type=BOOLEAN,access=rw------,values=1
  : values=off

#主mic开关（mic录音需要打开，耳机录音需要关闭）：
root@linaro-alip:~# amixer -c 1 cset  numid=67 on
numid=67,iface=MIXER,name='Main Mic Switch'
  ; type=BOOLEAN,access=rw------,values=1
  : values=on
root@linaro-alip:/# echo -n "mic2" > /sys/class/es8388_mic/mic_channel
root@linaro-alip:/# arecord -D hw:1,0 -f S16_LE -r 16000 -c 2 test_mic2.wa
v

#耳机录音：
root@linaro-alip:/# amixer -c 1 cset  numid=67 off
root@linaro-alip:/# echo -n "mic1" > /sys/class/es8388_mic/mic_channel
root@linaro-alip:/# arecord -D hw:1,0 -f S16_LE -r 16000 -c 2 test_mic1.wa
v

```

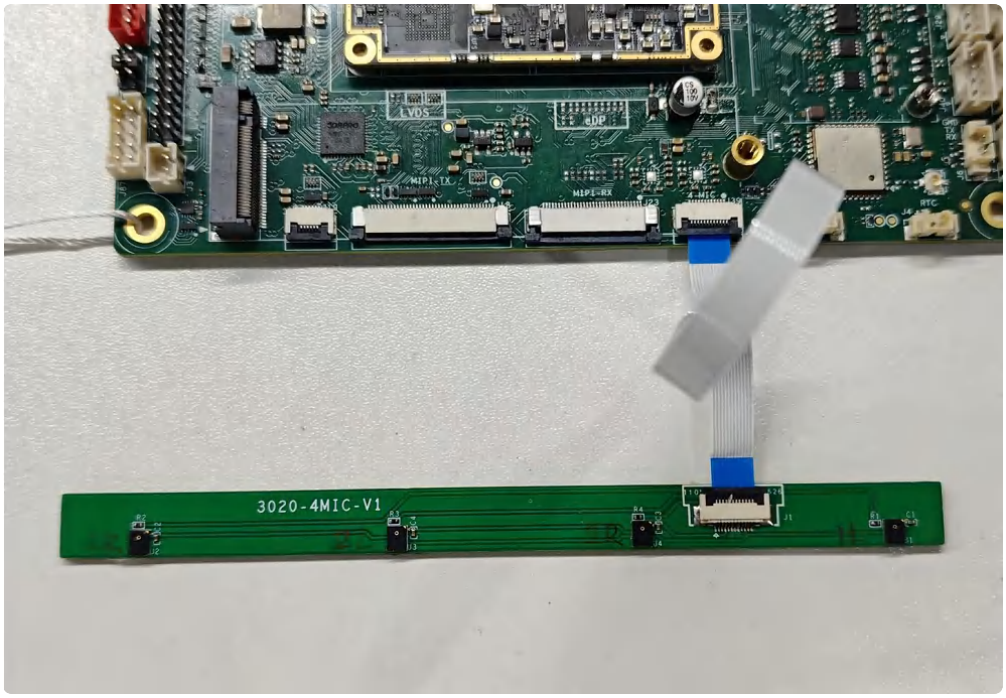
2.4.4. PDM-Mic

注意：PDM-Mic与HDMI-RX功能二选一，软硬件默认配置为HDMI-RX功能。

主板预留 PDM-Mic 阵列接口J39，如下图所示：



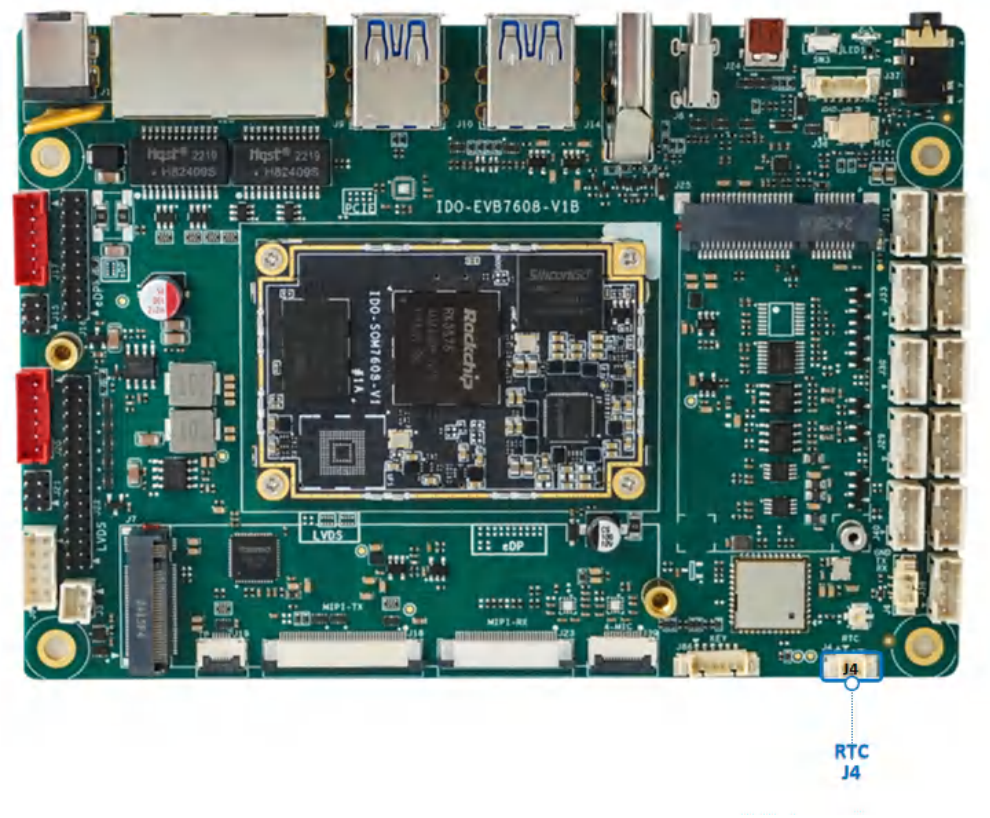
采用FPC05-FDW-12P座子连接Mic阵列，如下图所示：



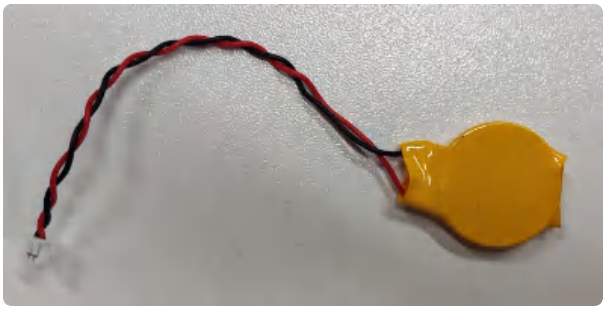
```
#查看录音设备
root@rk3576-buildroot:/# arecord -l
**** List of CAPTURE Hardware Devices ****
card 0: rockchipes8388 [rockchip-es8388], device 0: dailink-multicodecs ES8323 HiFi-0 [dailink-multicodecs ES8323 HiFi-0]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
card 1: rockchippdmmica [rockchip,pdm-mic-array], device 0: 2a6e0000.pdm-dummy_codec dummy_codec-0 [2a6e0000.pdm-dummy_codec dummy_codec-0]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
card 2: rockchiphdmi [rockchip-hdmi], device 0: rockchip-hdmi i2s-hifi-0 [rockchip-hdmi i2s-hifi-0]
  Subdevices: 1/1
  Subdevice #0: subdevice #0

#设置增益
root@rk3576-buildroot:/# amixer -c 1 cset numid=7 100
numid=7,iface=MIXER,name='PDM1 Gain Volume'
  ; type=INTEGER,access=rw---R--,values=1,min=0,max=127,step=0
  : values=100
  | dBscale-min=-65.63dB,step=0.75dB,mute=0

#录音测试
root@rk3576-buildroot:/# arecord -D hw:1,0 -f S16_LE -r 16000 -c 4 pmd_mic.wav
Recording WAVE 'pmd_mic5555.wav' : Signed 16 bit Little Endian, Rate 16000 Hz, Stereo

```

2.5. RTC

MX1.25T-2P RTC电池座J4，如下图所示：



连接3V 纽扣电池，RTC电池参考如下



rtc时间设置方法：

Plain Text

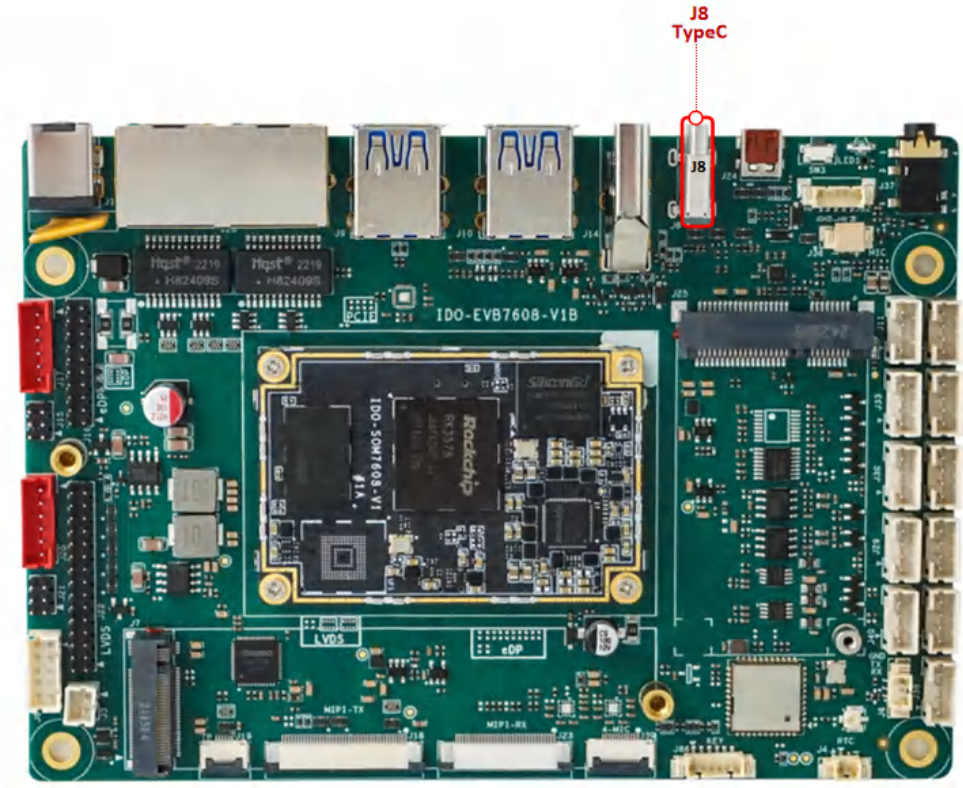
```
#设置时间
root@linaro-alip:/# date -s "2024-10-09 14:02:30"

#将rtc时钟调整为与目前的系统时钟一致
root@linaro-alip:/# hwclock -w

#获取硬件rtc当前时间（断电重启读取时间没有太大偏差）
root@linaro-alip:/# hwclock -r
2024-10-09 14:02:35.945604+00:00
```

2.6. USB接口

主板支持1个TypeC接口（USB3.2 Gen1 OTG+DP1.4），支持4个USB3.0-A接口，2个USB2.0PH2.0-4P 接口，USB对外总供电应小于4A。

2.6.1. TypeC接口

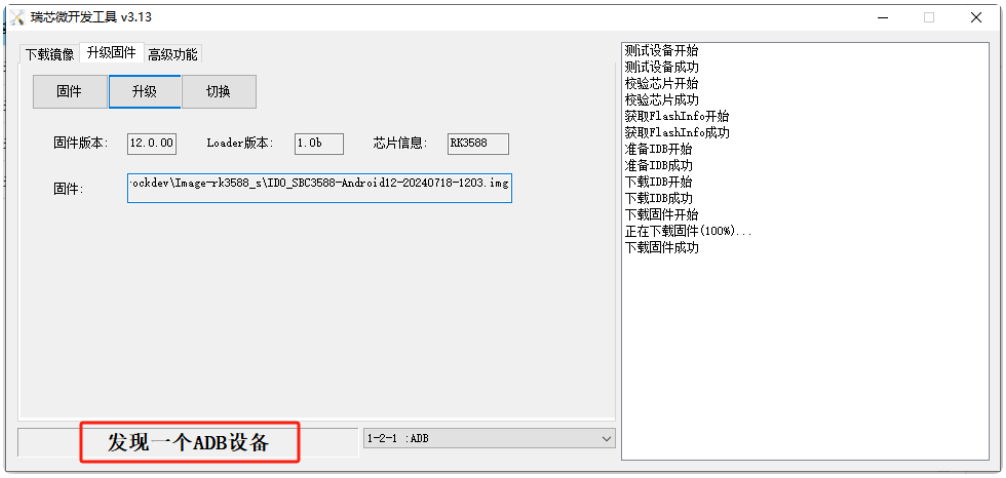
TypeC接口（USB3.2 Gen1 OTG+DP1.4输出）J8，如下图所示：



- 支持Host、Device模式自动切换
- 支持DP显示输出

Device从机模式

使用TypeC数据线连接电脑，烧录工具能发现一个ADB设备，如下图所示：



- 可使用ADB相关开发工具对盒子进行功能调试

Host主机模式

接入TypeC设备，或通过TypeC to USB-A转接头接入USB外设，如下图所示：

- 可识别U盘、键盘、鼠标并正常使用

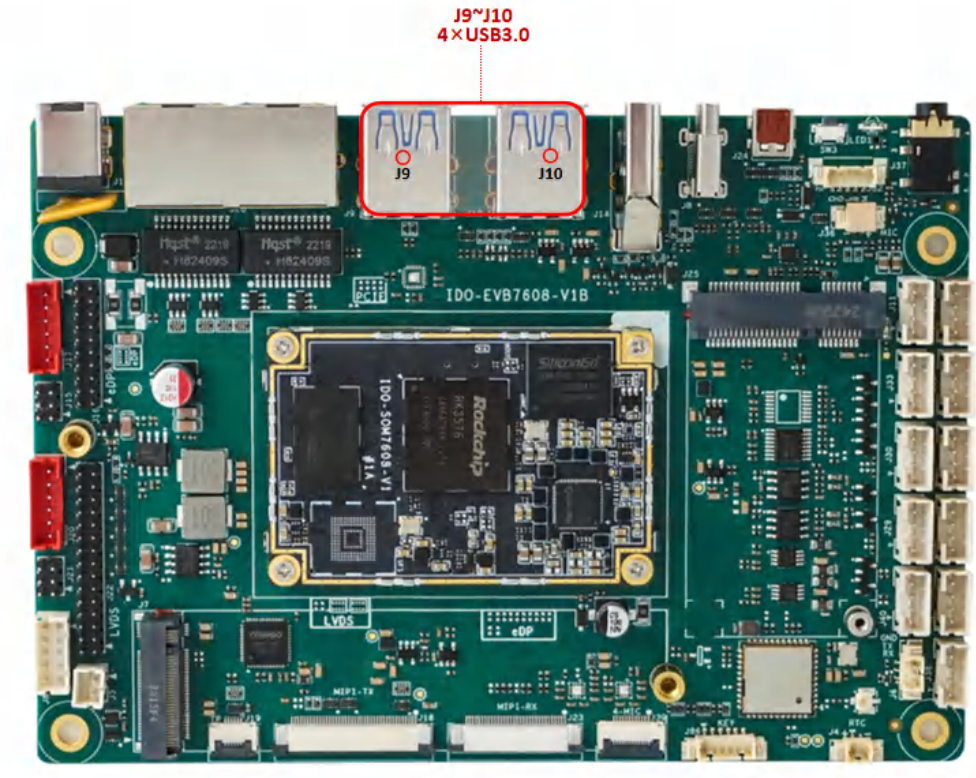
DP模式

通过TypeC全功能数据线接入DP显示器，或通过TYPE-C to HDMI数据线连接HDMI显示器

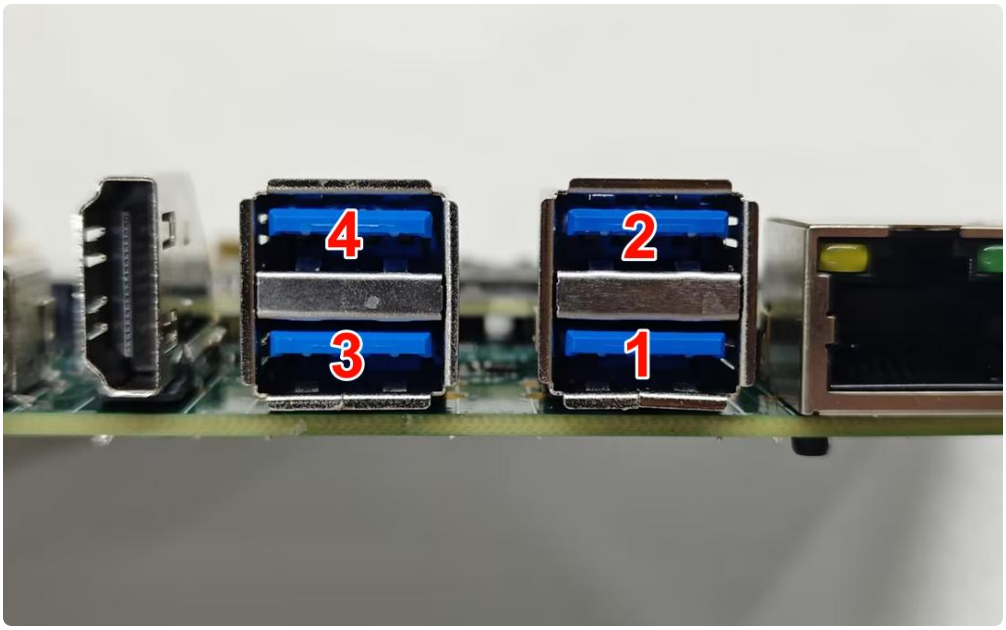
- 主板画面通过TYPE-C DP输出正常，DP声音输出正常

2.6.2. USB3.0接口

主板支持4个USB3.0接口，接口为标准的双层A口J9、J10，如下图所示：



USB母座提供5V@1A供电能力，每个USB端口供电可独立控制，USB序号如下图所示：



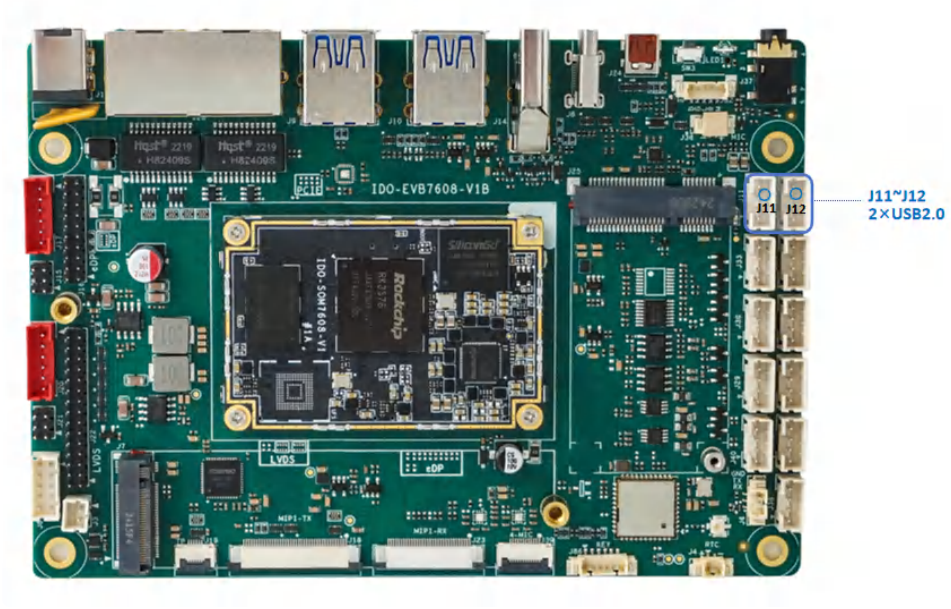
USB电源控制，如下表所示：

USB端口	默认状态	动作	命令
USB1	开启	关闭电源	echo 0 > /sys/class/leds/usb1_pwr/brightness
		开启电源	echo 1 > /sys/class/leds/usb1_pwr/brightness
USB2	开启	关闭电源	echo 0 > /sys/class/leds/usb2_pwr/brightness
		开启电源	echo 1 > /sys/class/leds/usb2_pwr/brightness
USB3	开启	关闭电源	echo 0 > /sys/class/leds/usb3_pwr/brightness
		开启电源	echo 1 > /sys/class/leds/usb3_pwr/brightness
USB4	开启	关闭电源	echo 0 > /sys/class/leds/usb4_pwr/brightness
		开启电源	echo 1 > /sys/class/leds/usb4_pwr/brightness

供电控制说明：设备节点写"0"关闭电源，写"1"开启电源

2.6.3. USB2.0接口

主板配置了2路USB2.0 PH2.0-4P接口，USB接口均提供5V@1A的驱动能力，如下图所示：

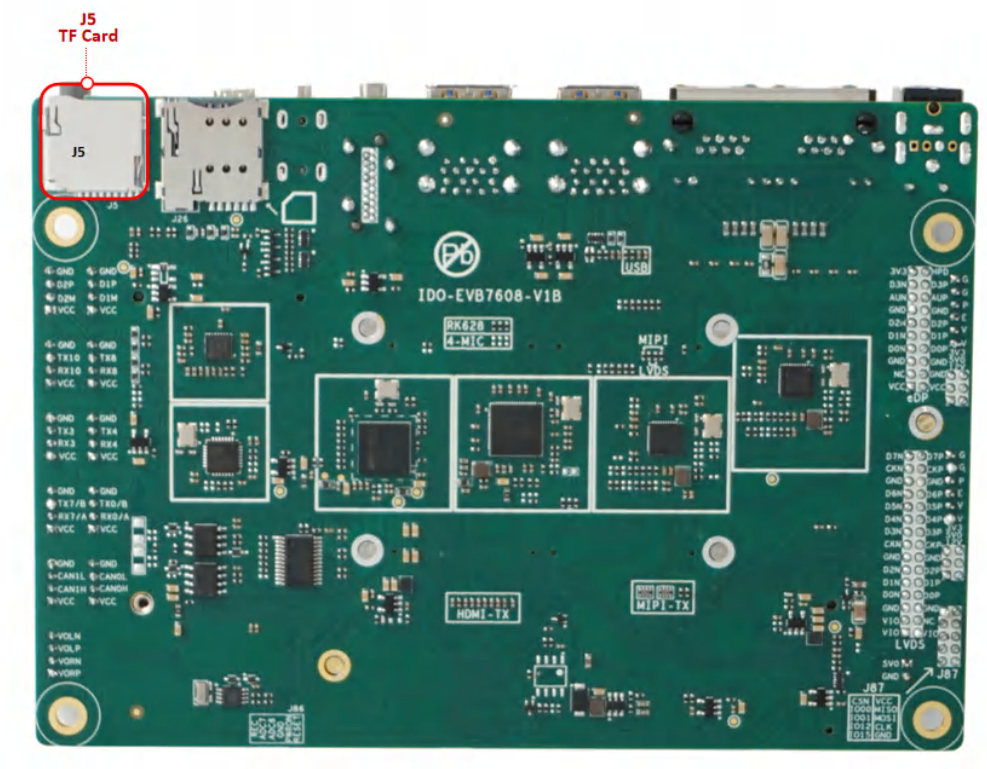


USB端口	默认状态	动作	命令
USB J11	开启	关闭电源	echo 0 > /sys/class/leds/usb5_pwr/brightness
		开启电源	echo 1 > /sys/class/leds/usb5_pwr/brightness
USB J12	开启	关闭电源	echo 0 > /sys/class/leds/usb6_pwr/brightness
		开启电源	echo 1 > /sys/class/leds/usb6_pwr/brightness

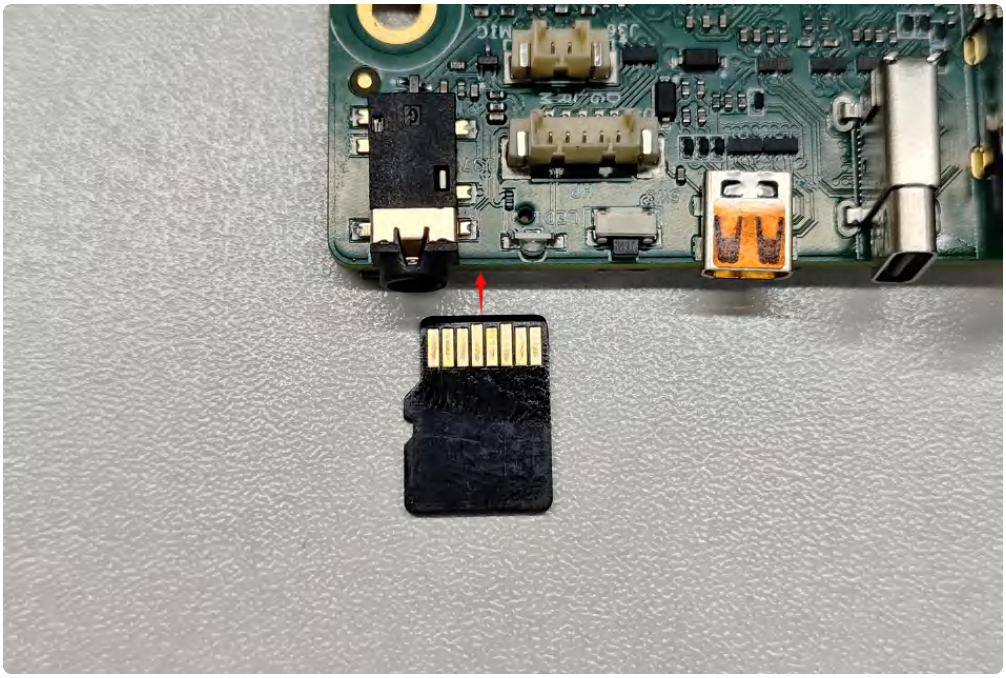
供电控制说明：设备节点写"0"关闭电源，写"1"开启电源

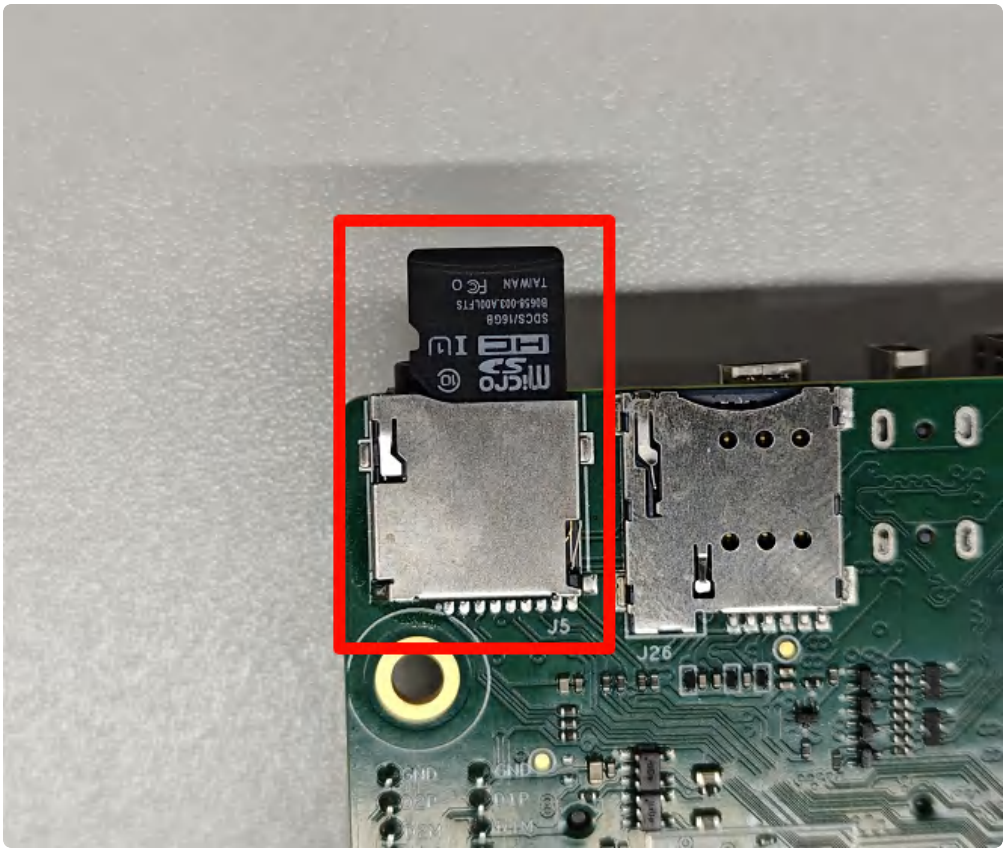
2.7. TF Card

TF卡座支持SDIO3.0， 支持高速SD卡，接口位于主板背面J5，如下图所示：



TF卡安装方向，主板正面朝上时TF卡触点朝上，如下图所示：





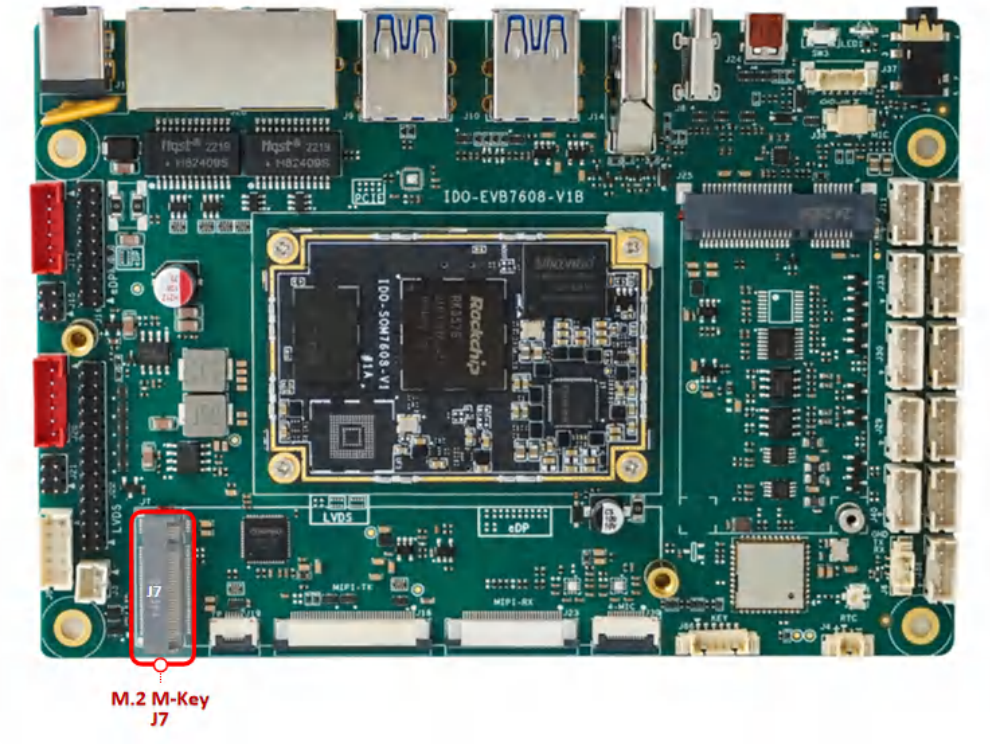
Shell

```
root@linaro-alip:/# fdisk -l
Disk /dev/ram0: 8 MiB, 8388608 bytes, 16384 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes
......

Device         Boot Start     End Sectors  Size Id Type
/dev/mmcblk1p1        135 3858623 3858489  1.8G  e W95 FAT16 (LBA)

root@linaro-alip:/# df -h
文件系统        大小  已用  可用 已用% 挂载点
/dev/root        53G  6.5G   45G   13% /
devtmpfs        4.0M  8.0K  4.0M    1% /dev
tmpfs           2.9G     0  2.9G    0% /dev/shm
tmpfs           1.2G  3.6M  1.2G    1% /run
tmpfs           5.0M   12K  5.0M    1% /run/lock
tmpfs           2.9G   16K  2.9G    1% /tmp
/dev/mmcblk2p7  124M   12M  109M   10% /oem
/dev/mmcblk2p6  4.0G  292K  4.0G    1% /userdata
/dev/mmcblk1p1  1.9G  768K  1.9G    1% /media/linaro/sdcard
```

注意：因为开机过程中由于emmc分区的变化，可能导致tf卡的分区也会变化，导致无法自动挂载，手动挂载即可。

2.8. M.2接口

IDO-EVB7608-V1主板上使用标准M.2-M-key M.2接口连接座，如下图所示：

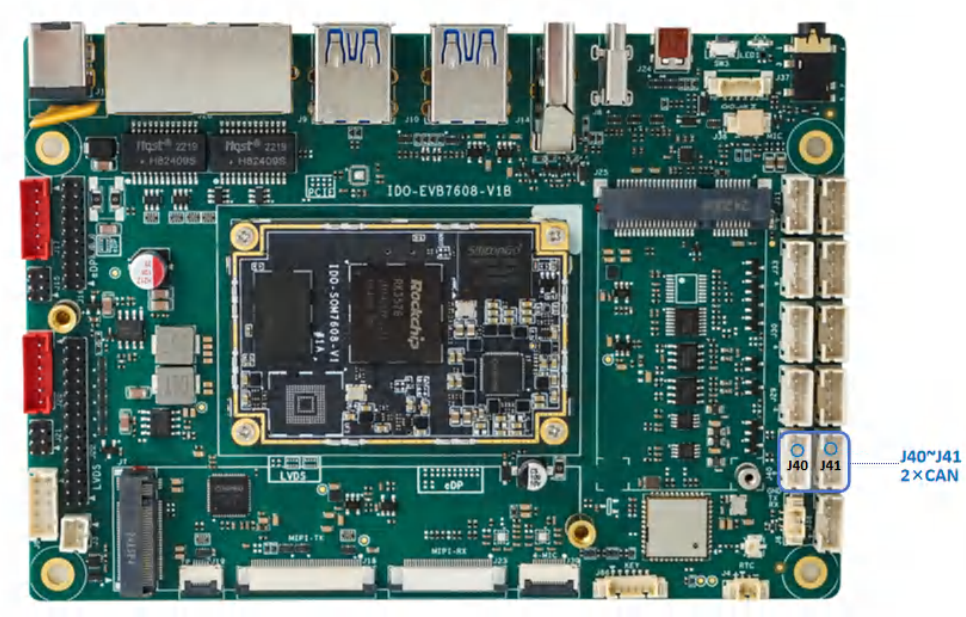


支持PCIe2.1通信，适用2280尺寸NVME固态硬盘。

```shell
root@linaro-alip:/# fdisk -l
Disk /dev/ram0: 8 MiB, 8388608 bytes, 16384 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes


Disk /dev/ram1: 8 MiB, 8388608 bytes, 16384 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes
...
Disk /dev/nvme0n1: 476.94 GiB, 512110190592 bytes, 1000215216 sectors
Disk model: XPG GAMMIX S11L
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x00000000

Device         Boot Start        End    Sectors   Size Id Type
/dev/nvme0n1p1       2048 1000215182 1000213135 476.9G  c W95 FAT32 (LBA)

root@linaro-alip:/# df -h
文件系统        大小  已用  可用 已用% 挂载点
/dev/root        53G  6.5G   45G   13% /
devtmpfs        4.0M  8.0K  4.0M    1% /dev
tmpfs           2.9G     0  2.9G    0% /dev/shm
tmpfs           1.2G  3.6M  1.2G    1% /run
tmpfs           5.0M   12K  5.0M    1% /run/lock
tmpfs           2.9G   24K  2.9G    1% /tmp
/dev/mmcblk2p7  124M   12M  109M   10% /oem
/dev/mmcblk2p6  4.0G  292K  4.0G    1% /userdata
/dev/nvme0n1p1  477G  2.1G  475G    1% /media/linaro/nvme
```

2.9. CAN接口

主板共有2路CAN接口J40、J41，如下图所示:



分别对应系统节点名称为 can0、can1。

1. 设置CAN参数

Shell

```
# 查看can0配置信息，可以看到can节点的波特率、位时序设置等
# ip -d -s -s link show can0

# 配置can之前需要先把can节点关闭
# ip link set can0 down

# 设置can0异常10ms重启
# ip link set can0 type can restart-ms 10

# 设置can0的比特率为1 Mbps/s
# ip link set can0 type can bitrate 1000000 dbitrate 1000000 fd on

# 打开can0节点
# ip link set can0 up
```

2. 测试收发

两台机器通过收发器连接好，发送方的 can_H 与接收方的 can_H连接，发送方的can_L 与接收方的 can_L连接。

can0 Shell

```
# can0节点发送数据
root@linaro-alip:/# ip link set can0 down
root@linaro-alip:/# ip link set can0 type can bitrate 1000000 dbitrate 1000000 fd on
root@linaro-alip:/# ip link set can0 up
root@linaro-alip:/# cansend can0 123#DEADBEEF
root@linaro-alip:/# cansend can0 123#DEADBEEF12345679

# can0接收数据
root@linaro-alip:/# ip link set can0 down
root@linaro-alip:/# ip link set can0 type can bitrate 1000000 dbitrate 1000000 fd on
root@linaro-alip:/# ip link set can0 up
root@linaro-alip:/# candump can0
```

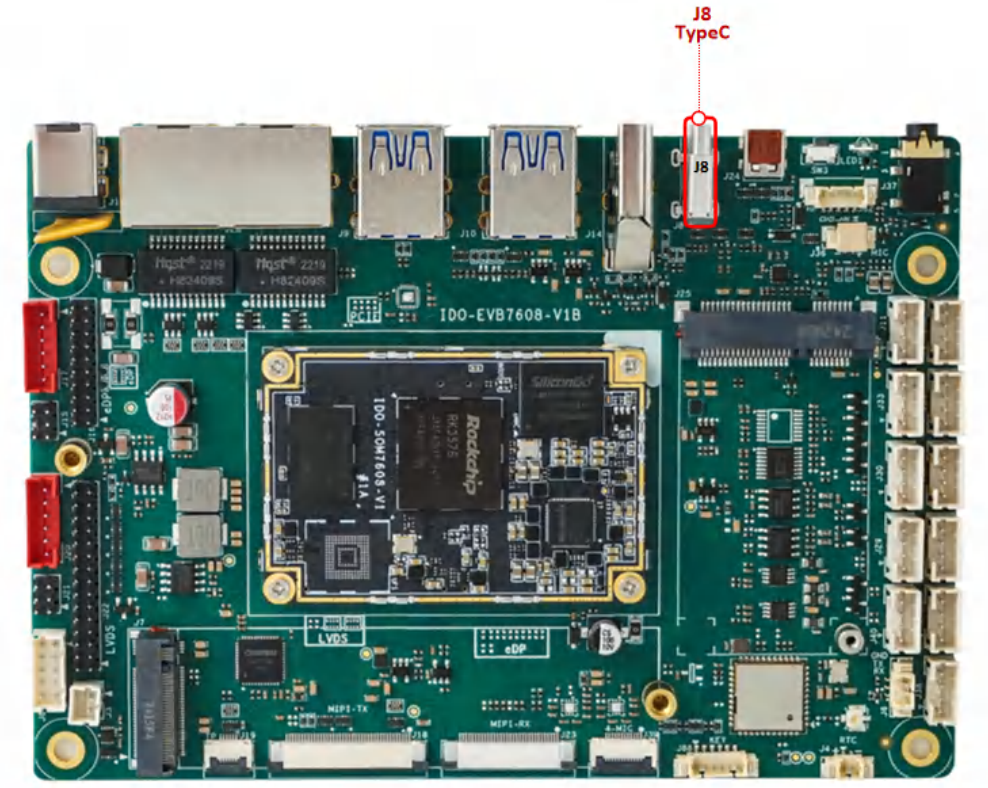
can1 Shell

```
# can1节点发送数据
root@linaro-alip:/# ip link set can1 down
root@linaro-alip:/# ip link set can1 type can bitrate 1000000 dbitrate 1000000 fd on
root@linaro-alip:/# ip link set can1 up
root@linaro-alip:/# cansend can1 123#DEADBEEF
root@linaro-alip:/# cansend can1 123#DEADBEEF11122233

# can1接收数据
root@linaro-alip:/# ip link set can1 down
root@linaro-alip:/# ip link set can1 type can bitrate 1000000 dbitrate 1000000 fd on
root@linaro-alip:/# ip link set can1 up
root@linaro-alip:/# candump can1
```

2.10. LCD显示

2.10.1. DP

DP接口（USB-C）J8，如下图所示：

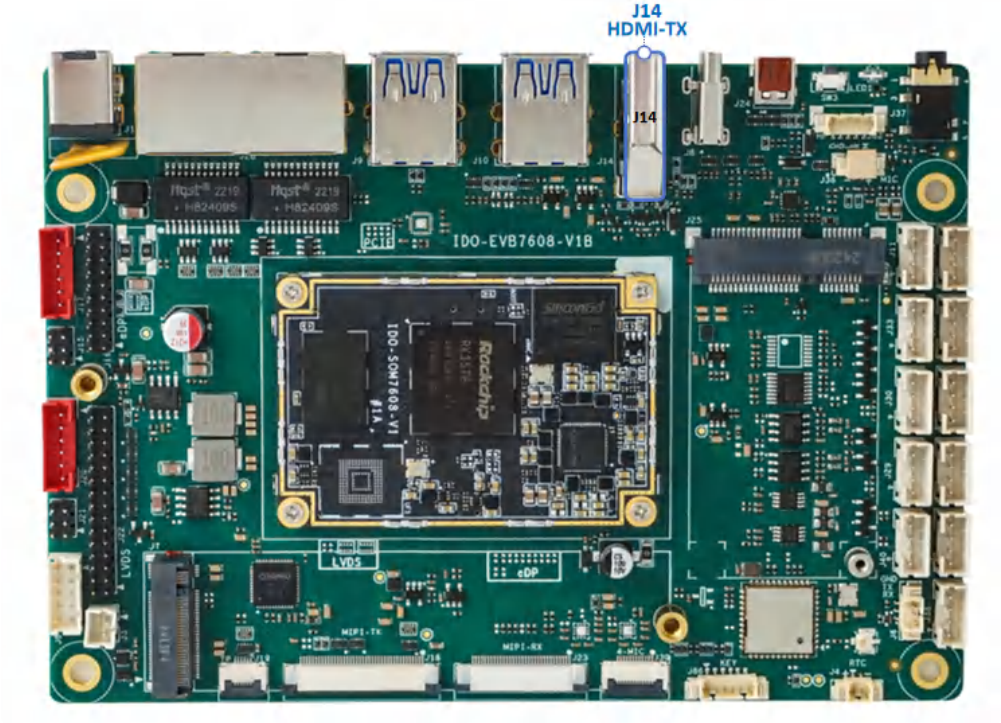


可以使用 USB-C 转 HDMI 高清线连接 HDMI 显示器输出画面，如下图所示：

- 最高支持4K@30fps输出

2.10.2. HDMI-TX

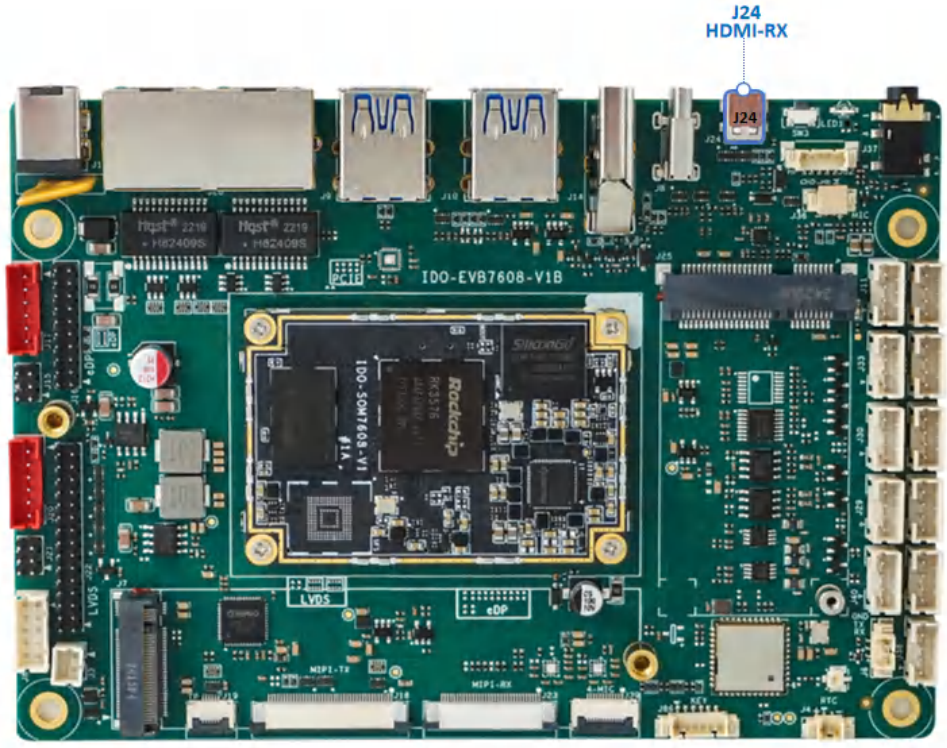
标准HDMI-A接口J14，如下图所示：



- 最高支持 HDMI2.1 8K@60fps 输出

2.10.3. HDMI-RX接口

Micro HDMI接口J24，如下图所示：



预览：

```
root@linaro-alip:/# gst-launch-1.0 v4l2src device=/dev/video0 ! video/x-raw,width=1920,height=1080,framerate=30/1 ! videoconvert ! autovideosink
```

- HDMI2.0-RX，支持4K@30fps

录音：

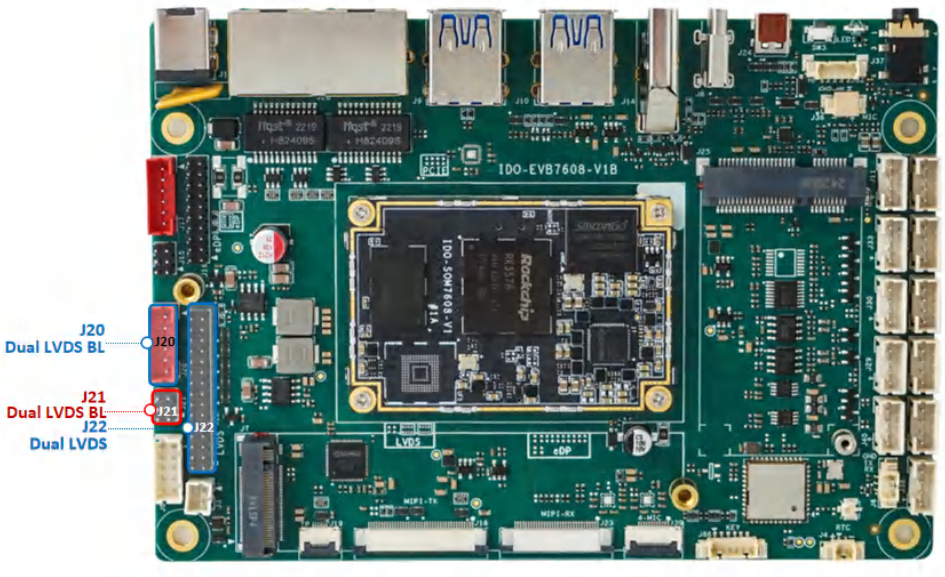
```
#窗口1
root@linaro-alip:/# gst-launch-1.0 v4l2src device=/dev/video0 ! video/x-raw,width=1920,height=1080,framerate=30/1 ! videoconvert ! autovideosink
#窗口2
root@linaro-alip:/# arecord -D hw:0,0 -f S16_LE -r 48000 -c 2 test.wav
```

注意：hdmi in录音之前确保另一端的hdmi out已经正常输出音频。

2.10.4. Dual LVDS屏

注意：Dual LVDS接口与MIPI-TX二选一，硬件默认配置为MIPI-TX。

Dual LVDS接口J22，背光接 J20，屏幕供电电压选择接口J21，如下图所示：

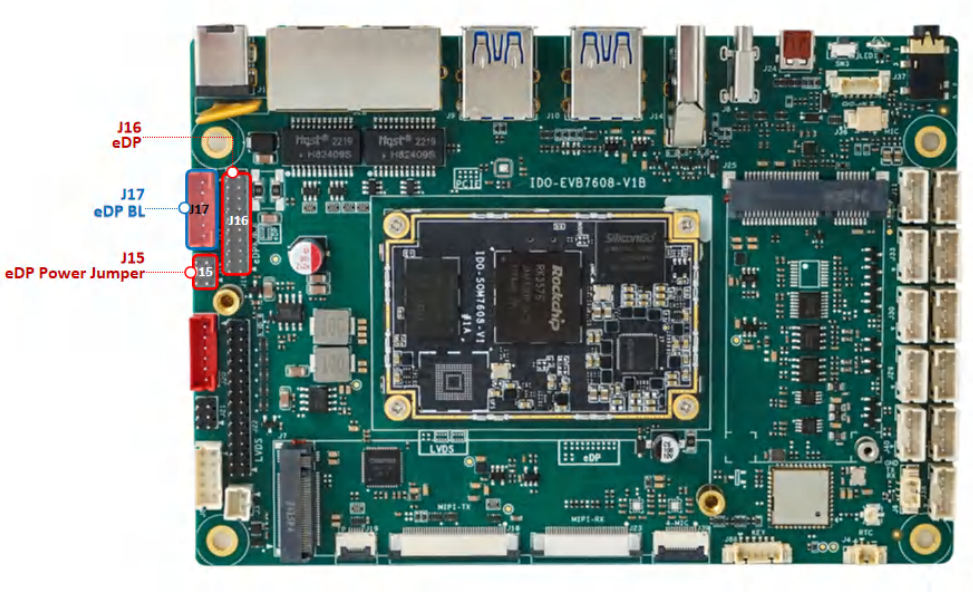


J21屏幕供电电压选择接口，接屏前，需要根据具体的屏幕规格书来确认J21的供电跳线帽接到3.3V、5V或12V，默认3.3V。

2.10.5. eDP

注意：eDP 接口与 HDMI-TX 二选一，硬件默认配置为 HDMI-TX。

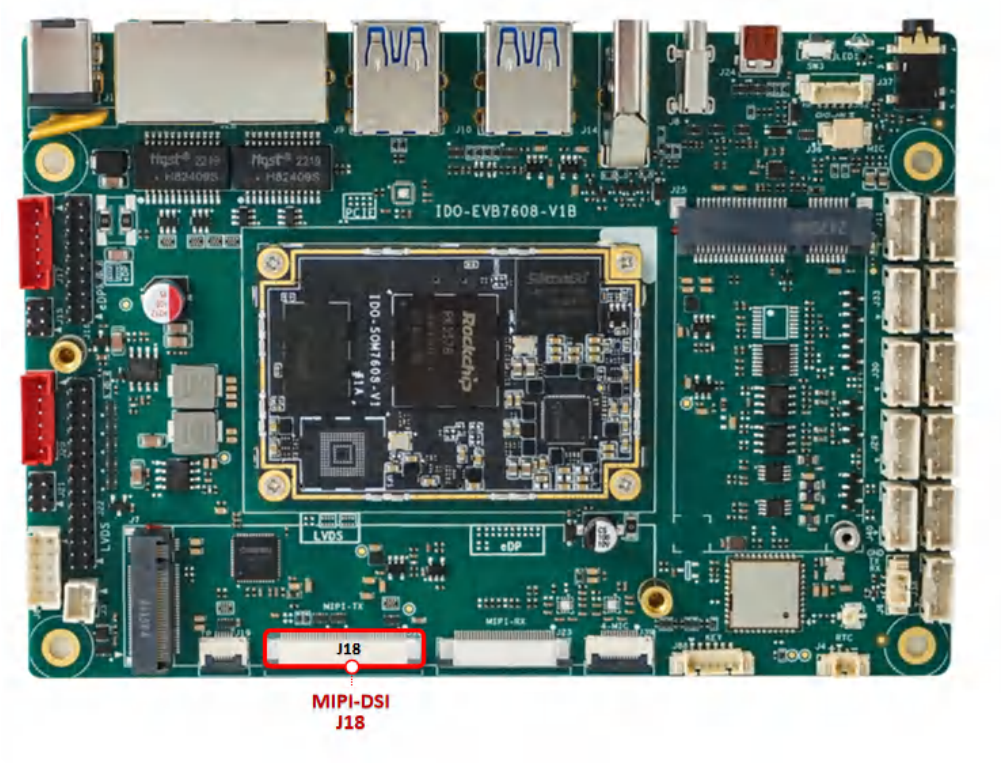
eDP接口J16，eDP背光接 J17，eDP屏幕供电电压选择接口J15，如下图所示：



J15屏幕供电电压选择接口，接屏前，需要根据具体的屏幕规格书来确认J15的供电跳线帽接到3.3V、5V或12V，默认3.3V。

2.10.6. MIPI

J18 MIPI接口为40Pin FPC 0.5mm 上接，如下图所示：

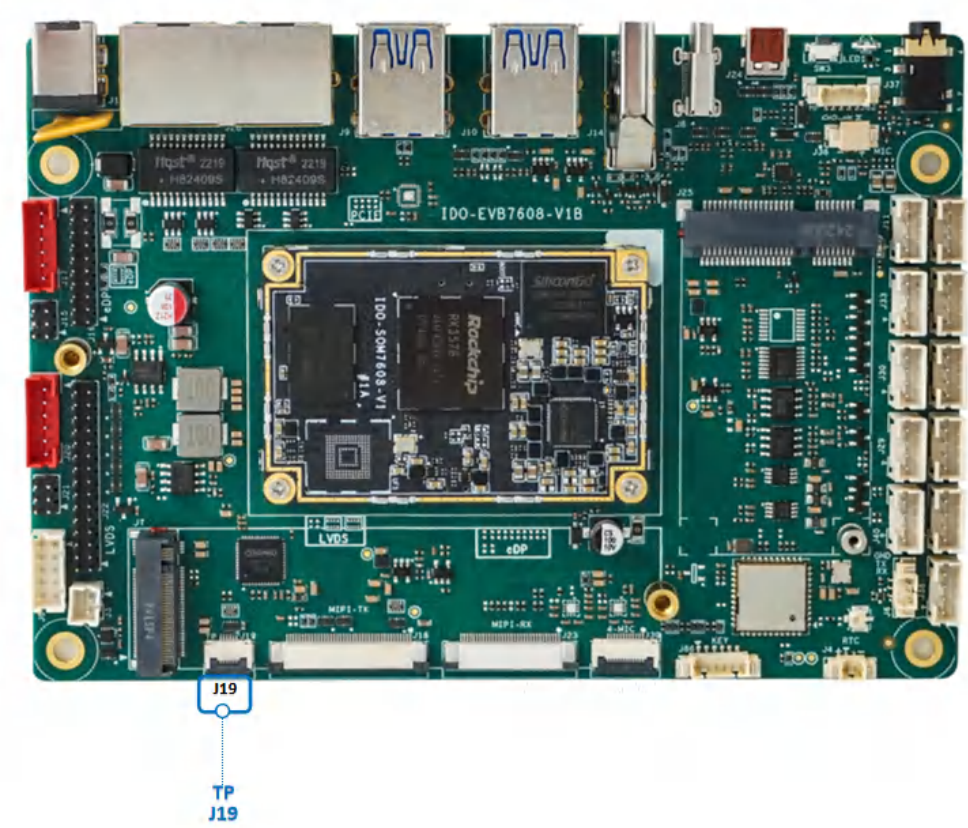


- MIPI供电为3.3V

注意：开发板的MIPI TX和Dual LVDS输出为复用关系，主板默认MIPI TX输出。

2.11. TP接口

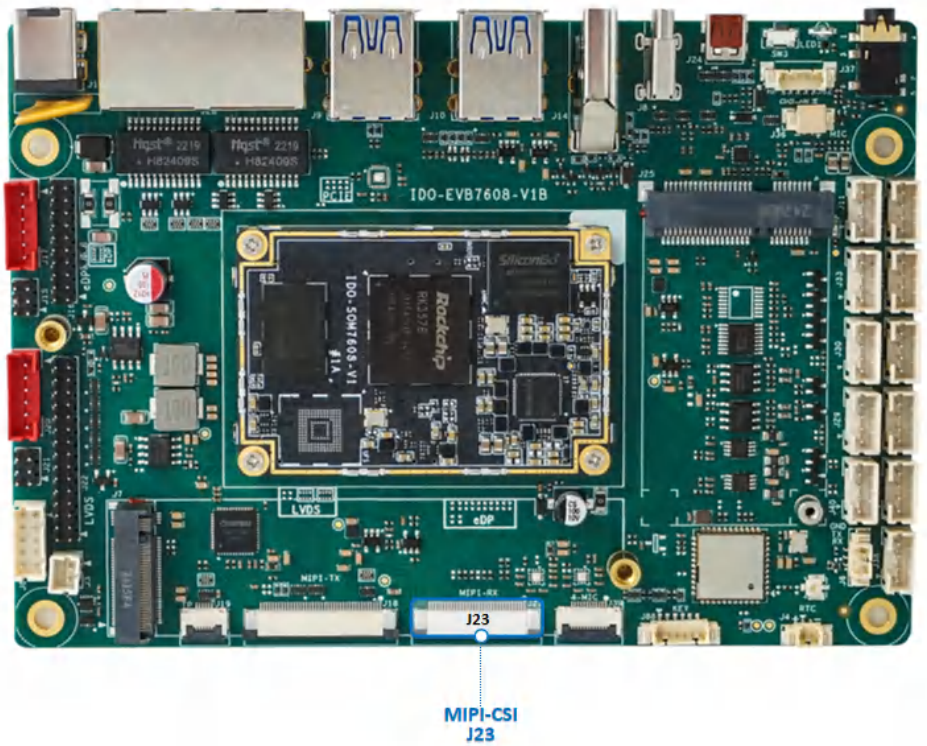
TP接口为(J19) 6Pin FPC 0.5mm座子，如下图所示：



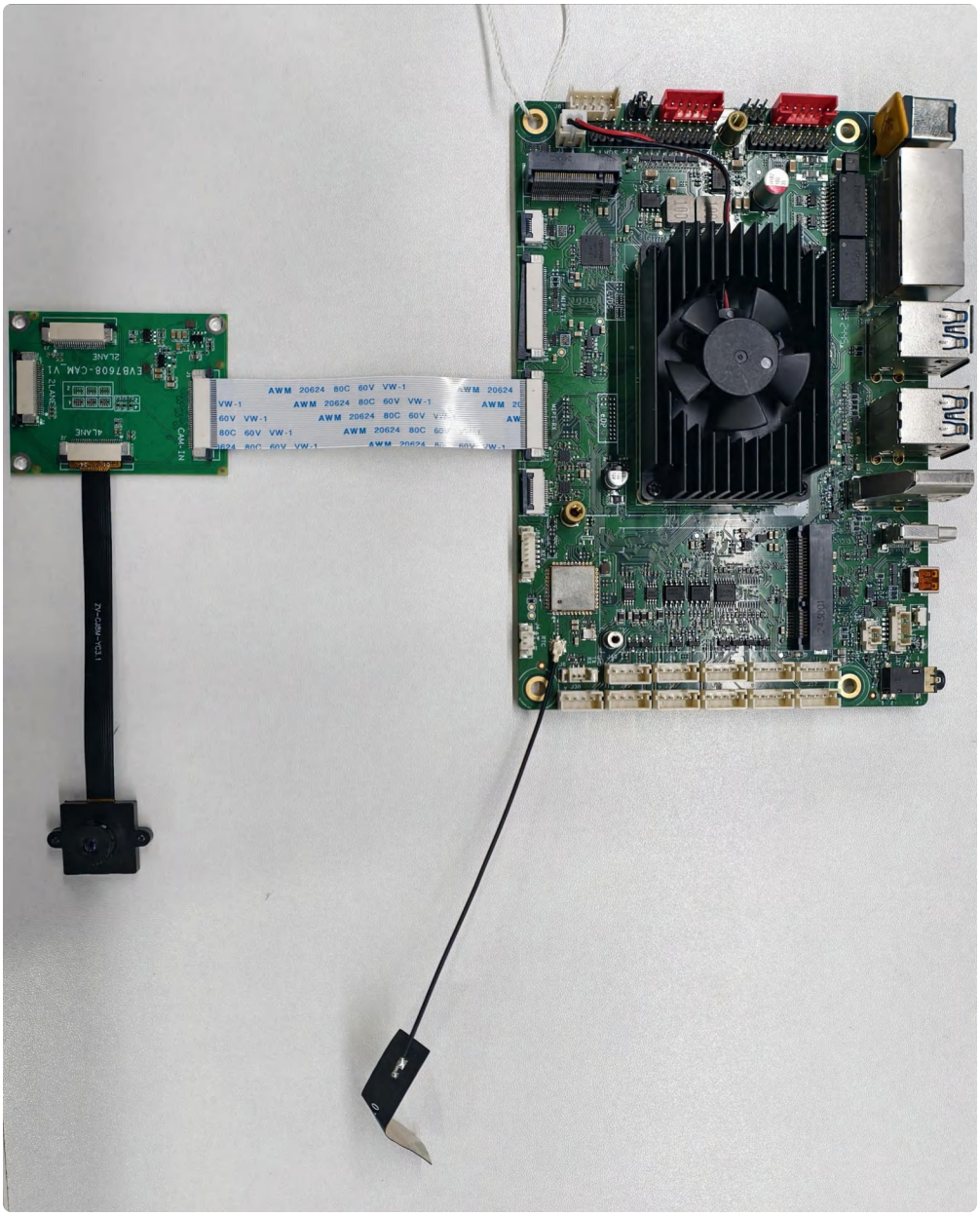
- 触摸 TP 接口接线方法为下接

2.12. MIPI Camera

MIPI Camer接口J23，如下图所示：



使用FPC05-PUW-32P抽拉式上接座子，默认适配IMX415摄像头（可接OV13855），接法如下图所示：



2.12.1. 预览命令：

Plain Text

```
root@linaro-alip:/# export DISPLAY=:0
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
```

2.12.2. 抓图：

```
root@linaro-alip:/# v4l2-ctl --verbose -d /dev/video22 --set-fmt-video=width=1920,height=1080,pixelformat='NV12' --stream-mmap=4 --stream-skip=3  --stream-to=./test-1920x1080-p50.yuv --stream-count=1 --stream-poll
```

查看照片：

```
root@linaro-alip:/# ffplay -f rawvideo -video_size 1920x1080 -pixel_format nv12 ./test-1920x1080-p50.yuv
```

2.12.3. 抓视频：

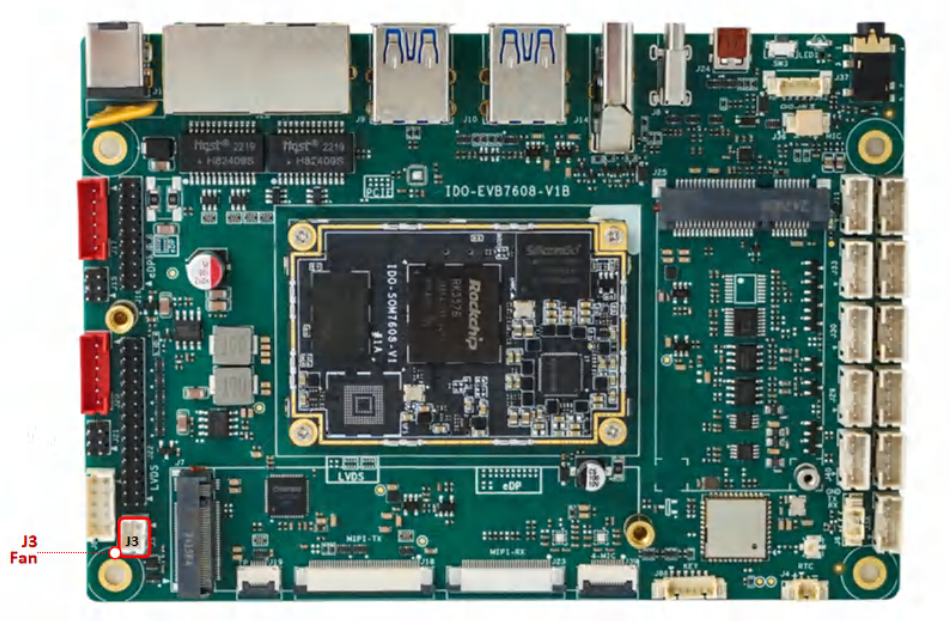
```
root@linaro-alip:/# v4l2-ctl --stream-mmap=4  -d /dev/video22 --set-fmt-video=width=1920,height=1080,pixelformat='NV12' --stream-to=./output.yuv
```

播放视频：

```
root@linaro-alip:/# ffplay -f rawvideo -video_size 1920x1080 -pixel_format nv12 ./output.yuv
```

2.13. FAN 风扇

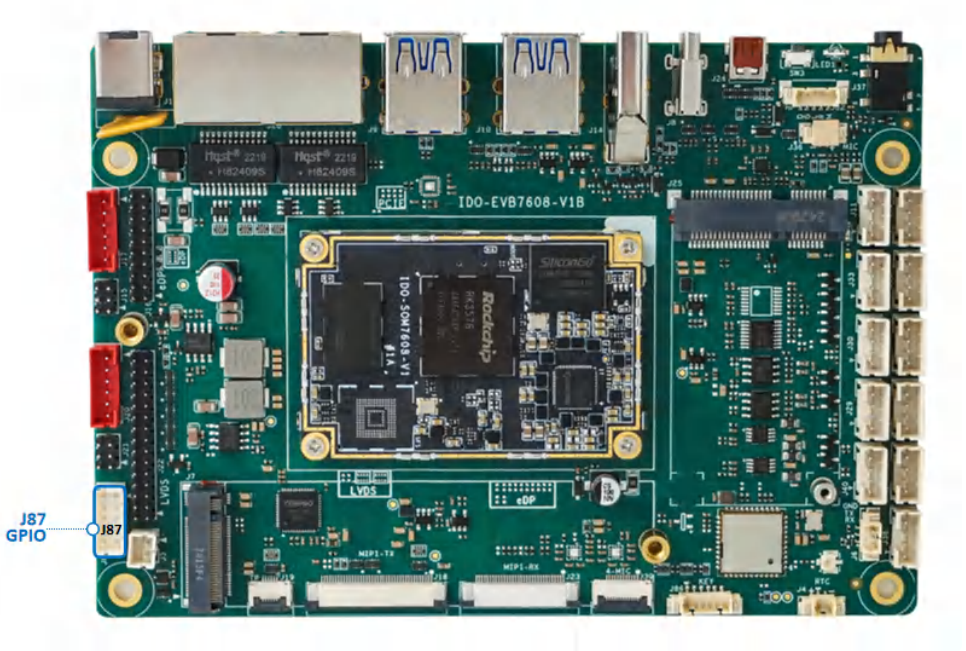
PH2 5V FAN风扇接口J3，如下图所示：



控制功能	控制命令
打开风扇	echo 1 > /sys/class/leds/fan/brightness
关闭风扇	echo 0 > /sys/class/leds/fan/brightness

2.14. GPIO

主板预留了8个GPIO接口J87，如下图所示：



GPIO引脚定义描述，如下图所示：

序号	管脚标号	GPIO 标号	GPIO 序号	LIBGPIOD节点
1	CSN	GPIO3_D0	120	gpiochip3 24
2	VCC	5V/3.3V	/	/
3	IO00	IO0_0	493	gpiochip6 0
4	MISO	GPIO3_C5	117	gpiochip3 21
5	IO01	IO0_1	494	gpiochip6 1
6	MOSI	GPIO3_C6	118	gpiochip3 22
7	IO12	IO1_2	503	gpiochip6 10
8	CLK	GPIO3_C7	119	gpiochip3 23
9	IO15	IO1_5	506	gpiochip6 13
10	GND	地	/	/

GPIO控制，根据上表【LIBGPIOD节点】进行命令操作

libgpiod C

```c
#安装命令
apt install -y libgpiod-dev gpiod

# 设置 GPIO3_D0 输出高电平状态
gpioset gpiochip3 24=1

# 设置 GPIO3_D0 输出低电平状态
gpioset gpiochip3 24=0

# 设置 IO00 输出高电平状态
gpioset gpiochip6 0=1

# 设置 IO00 输出低电平状态
gpioset gpiochip6 0=0


# 获取 GPIO3_D0 输入电平状态
gpioget gpiochip3 24

# 获取 IO00 输入电平状态
gpioget gpiochip6 0
```

其中1、4、6、8脚可配置为SPI接口，对应设备树SPI1_M2

序号	管脚标号	GPIO 标号	GPIO 序号
1	CSN	GPIO1_A3	120
4	MISO	GPIO3_C5	117
6	MOSI	GPIO3_C6	118
8	CLK	GPIO3_C7	119

2.15. GPU

执行glmark2-es2 开启gpu测试窗口：

```
root@linaro-alip:/# glmark2-es2
arm_release_ver: g13p0-01eac0, rk_so_ver: 10
=======================================================
    glmark2 2023.01
=======================================================
    OpenGL Information
    GL_VENDOR:      ARM
    GL_RENDERER:    Mali-G52
    GL_VERSION:     OpenGL ES 3.2 v1.g13p0-01eac0.0fd2effaec483a5f4c440d2ffa25eb7a
    Surface Config: buf=32 r=8 g=8 b=8 a=8 depth=24 stencil=0 samples=0
    Surface Size:   800x600 windowed
=======================================================
[build] use-vbo=false: FPS: 365 FrameTime: 2.740 ms
[build] use-vbo=true: FPS: 418 FrameTime: 2.395 ms
[texture] texture-filter=nearest: FPS: 467 FrameTime: 2.145 ms
[texture] texture-filter=linear: FPS: 461 FrameTime: 2.172 ms
[texture] texture-filter=mipmap: FPS: 462 FrameTime: 2.165 ms
[shading] shading=gouraud: FPS: 382 FrameTime: 2.622 ms
[shading] shading=blinn-phong-inf: FPS: 407 FrameTime: 2.461 ms
[shading] shading=phong: FPS: 390 FrameTime: 2.566 ms
[shading] shading=cel: FPS: 386 FrameTime: 2.594 ms
[bump] bump-render=high-poly: FPS: 258 FrameTime: 3.881 ms
[bump] bump-render=normals: FPS: 474 FrameTime: 2.111 ms
[bump] bump-render=height: FPS: 472 FrameTime: 2.120 ms
[effect2d] kernel=0,1,0;1,-4,1;0,1,0;: FPS: 390 FrameTime: 2.569 ms
[effect2d] kernel=1,1,1,1,1;1,1,1,1,1;1,1,1,1,1;: FPS: 252 FrameTime: 3.975 ms
[pulsar] light=false:quads=5:texture=false: FPS: 484 FrameTime: 2.070 ms
[desktop] blur-radius=5:effect=blur:passes=1:separable=true:windows=4: FPS: 253 FrameTime: 3.956 ms
[desktop] effect=shadow:windows=4: FPS: 375 FrameTime: 2.668 ms
[buffer] columns=200:interleave=false:update-dispersion=0.9:update-fraction=0.5:update-method=map: FPS: 124 FrameTime: 8.085 ms
[buffer] columns=200:interleave=false:update-dispersion=0.9:update-fraction=0.5:update-method=subdata: FPS: 121 FrameTime: 8.288 ms
[buffer] columns=200:interleave=true:update-dispersion=0.9:update-fraction=0.5:update-method=map: FPS: 176 FrameTime: 5.688 ms
[ideas] speed=duration: FPS: 220 FrameTime: 4.552 ms
[jellyfish] <default>: FPS: 341 FrameTime: 2.933 ms
[terrain] <default>: FPS: 67 FrameTime: 14.968 ms
[shadow] <default>: FPS: 261 FrameTime: 3.837 ms
[refract] <default>: FPS: 106 FrameTime: 9.462 ms

```

```
[conditionals] fragment-steps=0:vertex-steps=0: FPS: 478 FrameTime: 2.093 ms
[conditionals] fragment-steps=5:vertex-steps=0: FPS: 451 FrameTime: 2.218 ms
[conditionals] fragment-steps=0:vertex-steps=5: FPS: 471 FrameTime: 2.127 ms
[function] fragment-complexity=low:fragment-steps=5: FPS: 477 FrameTime: 2.097 ms
[function] fragment-complexity=medium:fragment-steps=5: FPS: 427 FrameTime: 2.343 ms
[loop] fragment-loop=false:fragment-steps=5:vertex-steps=5: FPS: 453 FrameTime: 2.209 ms
[loop] fragment-steps=5:fragment-uniform=false:vertex-steps=5: FPS: 469 FrameTime: 2.135 ms
[loop] fragment-steps=5:fragment-uniform=true:vertex-steps=5: FPS: 442 FrameTime: 2.267 ms
=======================================================
                                  glmark2 Score: 355
```

2.16. NPU

使用yolov5进行rknn模型转换测试：

2.16.1. 编译rknn_yolov5_demo：

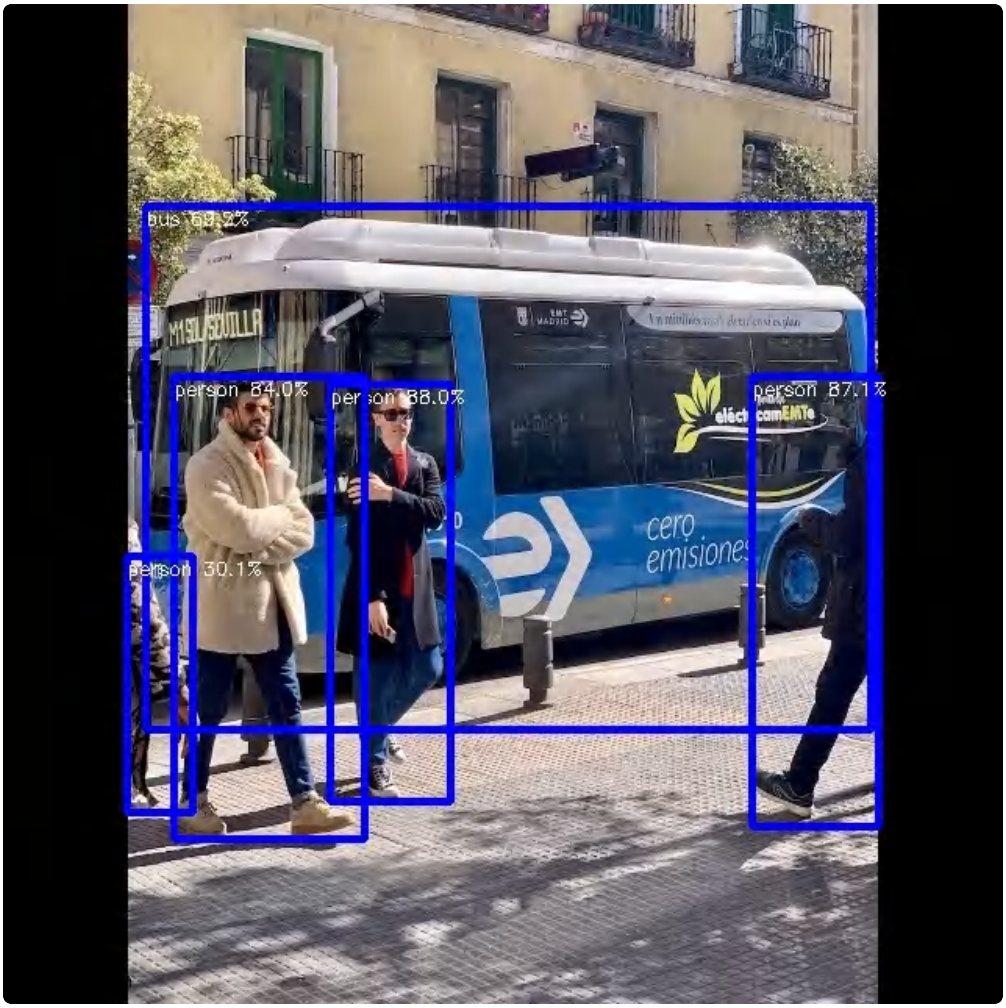
```shell
root@linaro-alip:/# unzip rknn-toolkit2-master.zip
root@linaro-alip:/# cd rknn-toolkit2-master/rknpu2/examples/3rdparty/mpp/Linux/aarch64
root@linaro-alip:/# ln -srf librockchip_mpp.so.0 librockchip_mpp.so
root@linaro-alip:/# ln -srf librockchip_mpp.so.0 librockchip_mpp.so.1

root@linaro-alip:/# cd -
root@linaro-alip:/# cd rknn-toolkit2-master/rknpu2/examples/rknn_yolov5_demo
root@linaro-alip:/# chmod a+x build-linux.sh
root@linaro-alip:/# export GCC_COMPILER=/usr/bin/aarch64-linux-gnu
root@linaro-alip:/mnt/rknn-toolkit2-master/rknpu2/examples/rknn_yolov5_demo# ./build-linux.sh -t rk3576 -a aarch64 -b Release
./build-linux.sh -t rk3576 -a aarch64 -b Release
/usr/bin/aarch64-linux-gnu
==================================
TARGET_SOC=RK3576
TARGET_ARCH=aarch64
BUILD_TYPE=Release
BUILD_DIR=/mnt/rknn-toolkit2-master/rknpu2/examples/rknn_yolov5_demo/build/build_RK3576_linux_aarch64_Release
CC=/usr/bin/aarch64-linux-gnu-gcc
CXX=/usr/bin/aarch64-linux-gnu-g++
==================================
-- Configuring done
-- Generating done
-- Build files have been written to: /mnt/rknn-toolkit2-master/rknpu2/examples/rknn_yolov5_demo/build/build_RK3576_linux_aarch64_Release
[ 60%] Built target rknn_yolov5_video_demo
[100%] Built target rknn_yolov5_demo
[ 40%] Built target rknn_yolov5_demo
[100%] Built target rknn_yolov5_video_demo
Install the project...
-- Install configuration: "Release"
-- Up-to-date: /mnt/rknn-toolkit2-master/rknpu2/examples/rknn_yolov5_demo/install/rknn_yolov5_demo_Linux/./rknn_yolov5_demo
-- Up-to-date: /mnt/rknn-toolkit2-master/rknpu2/examples/rknn_yolov5_demo/install/rknn_yolov5_demo_Linux/lib/librknnrt.so
-- Up-to-date: /mnt/rknn-toolkit2-master/rknpu2/examples/rknn_yolov5_demo/install/rknn_yolov5_demo_Linux/lib/librga.so
-- Up-to-date: /mnt/rknn-toolkit2-master/rknpu2/examples/rknn_yolov5_demo/install/rknn_yolov5_demo_Linux/./model/RK3576
-- Up-to-date: /mnt/rknn-toolkit2-master/rknpu2/examples/rknn_yolov5_demo/install/rknn_yolov5_demo_Linux/./model/RK3576/yolov5s-640-640.rknn
```

```
-- Up-to-date: /mnt/rknn-toolkit2-master/rknpu2/examples/rknn_yolov5_demo/install/rknn_yolov5_demo_Linux/./model/bus.jpg
-- Up-to-date: /mnt/rknn-toolkit2-master/rknpu2/examples/rknn_yolov5_demo/install/rknn_yolov5_demo_Linux/./model/coco_80_labels_list.txt
-- Up-to-date: /mnt/rknn-toolkit2-master/rknpu2/examples/rknn_yolov5_demo/install/rknn_yolov5_demo_Linux/./rknn_yolov5_video_demo
-- Up-to-date: /mnt/rknn-toolkit2-master/rknpu2/examples/rknn_yolov5_demo/install/rknn_yolov5_demo_Linux/lib/librockchip_mpp.so
-- Up-to-date: /mnt/rknn-toolkit2-master/rknpu2/examples/rknn_yolov5_demo/install/rknn_yolov5_demo_Linux/lib/libmk_api.so
```

2.16.2. 测试：

```shell
root@linaro-alip:/# cd install/rknn_yolov5_demo_Linux
root@linaro-alip:/# chmod a+x rknn_yolov5_demo
root@linaro-alip:/mnt/rknn-toolkit2-master/rknpu2/examples/rknn_yolov5_demo/install/rknn_yolov5_demo_Linux# ./rknn_yolov5_demo ./model/RK3576/yolov5s-640-640.rknn ./model/bus.jpg
post process config: box_conf_threshold = 0.25, nms_threshold = 0.45
Loading mode...
sdk version: 2.3.2 (429f97ae6b@2025-04-09T09:09:27) driver version: 0.9.8
model input num: 1, output num: 3
  index=0, name=images, n_dims=4, dims=[1, 640, 640, 3], n_elems=1228800, size=1228800, w_stride = 640, size_with_stride=1228800, fmt=NHWC, type=INT8, qnt_type=AFFINE, zp=-128, scale=0.003922
  index=0, name=output0, n_dims=4, dims=[1, 255, 80, 80], n_elems=1632000, size=1632000, w_stride = 0, size_with_stride=1638400, fmt=NCHW, type=INT8, qnt_type=AFFINE, zp=-128, scale=0.003922
  index=1, name=286, n_dims=4, dims=[1, 255, 40, 40], n_elems=408000, size=408000, w_stride = 0, size_with_stride=491520, fmt=NCHW, type=INT8, qnt_type=AFFINE, zp=-128, scale=0.003922
  index=2, name=288, n_dims=4, dims=[1, 255, 20, 20], n_elems=102000, size=102000, w_stride = 0, size_with_stride=163840, fmt=NCHW, type=INT8, qnt_type=AFFINE, zp=-128, scale=0.003922
model is NHWC input fmt
model input height=640, width=640, channel=3
Read ./model/bus.jpg ...
img width = 640, img height = 640
once run use 41.249000 ms
loadLabelName ./model/coco_80_labels_list.txt
person @ (209 243 286 510) 0.879723
person @ (479 238 560 526) 0.870588
person @ (109 237 232 534) 0.828112
bus @ (93 129 553 464) 0.700761
person @ (79 353 122 517) 0.307297
save detect result to ./out.jpg
loop count = 10 , average run  28.800200 ms
```

执行完后会得到转换后的模型图片，如下所示：



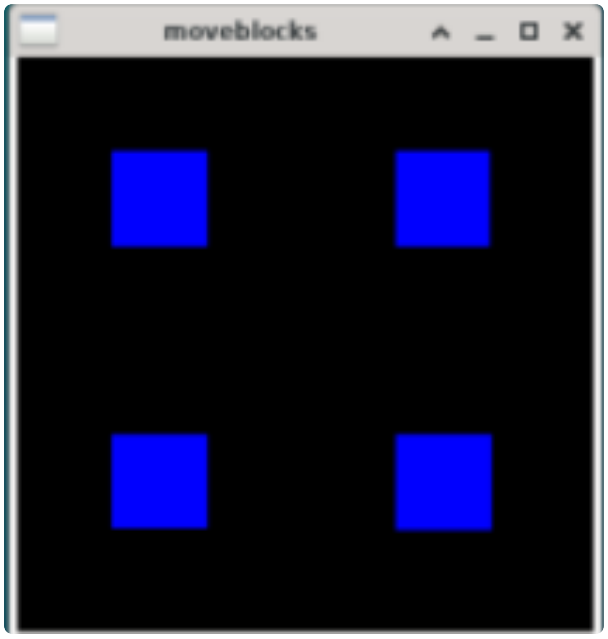
2.17. QT5

支持qt5.15，测试方法如下：

Shell

```
root@linaro-alip:/# source /usr/local/qt5/env.sh
root@linaro-alip:/# chmod a+x ./moveblocks
root@linaro-alip:/# ./moveblocks
```

结果如下所示：

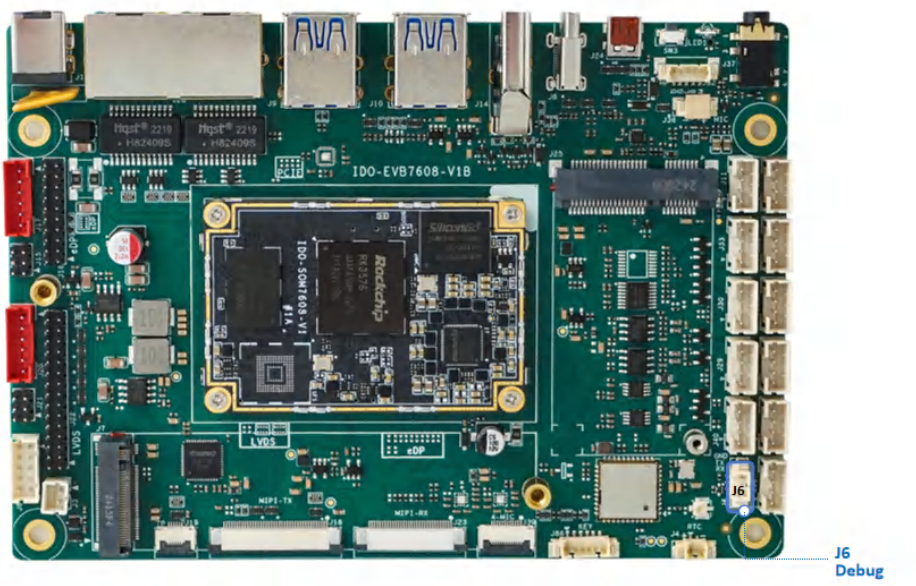

moveblocks

IDO-EVB7608-V1B-Buildroot系统

3. 功能及接口使用方法

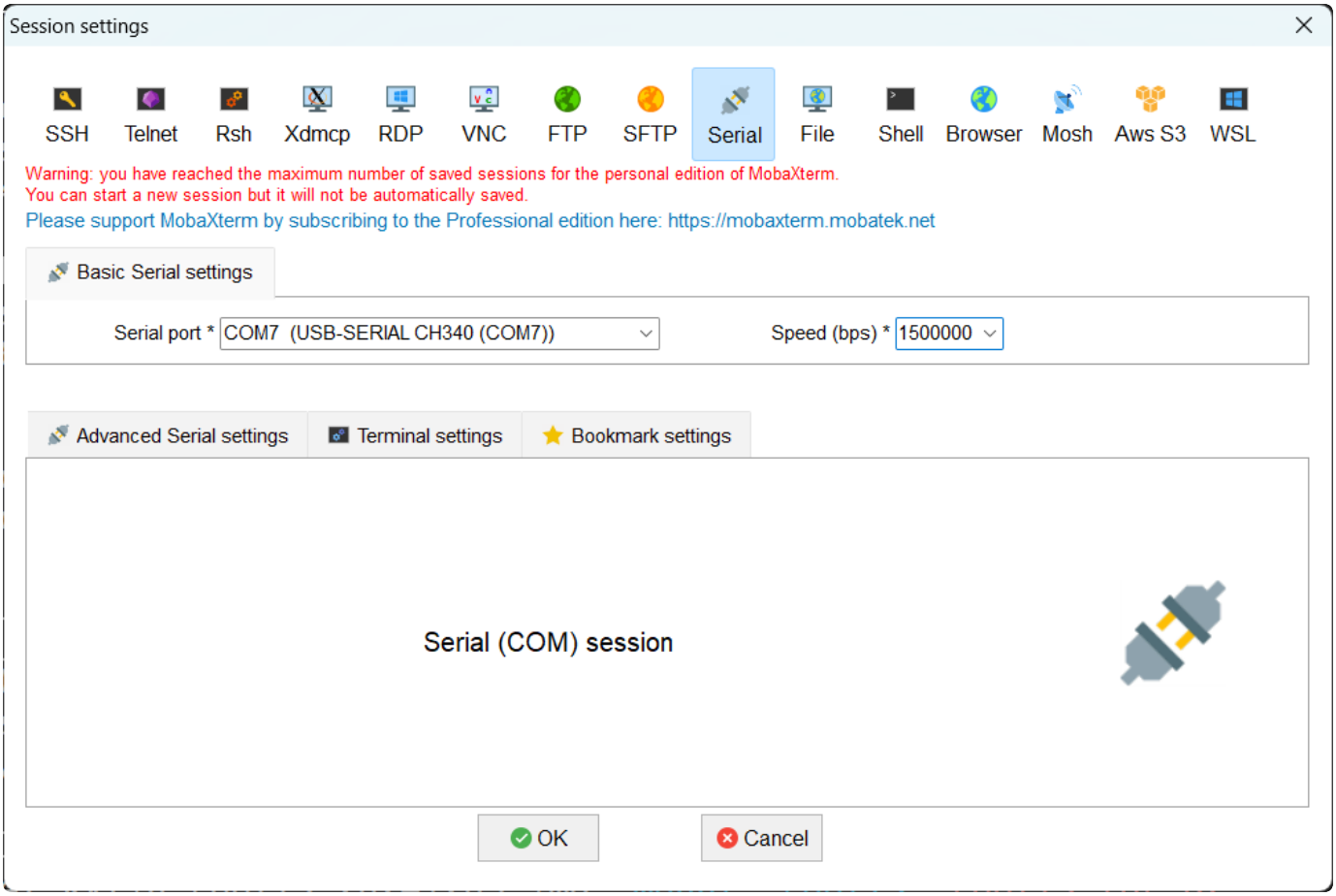
3.1. DEBUG UART

DEBUG UART位置J6，如下图所示：



使用USB Type-C数据线，连接PC端的USB接口。系统会识别到一个“USB-SERIAL CH340”端口设备。

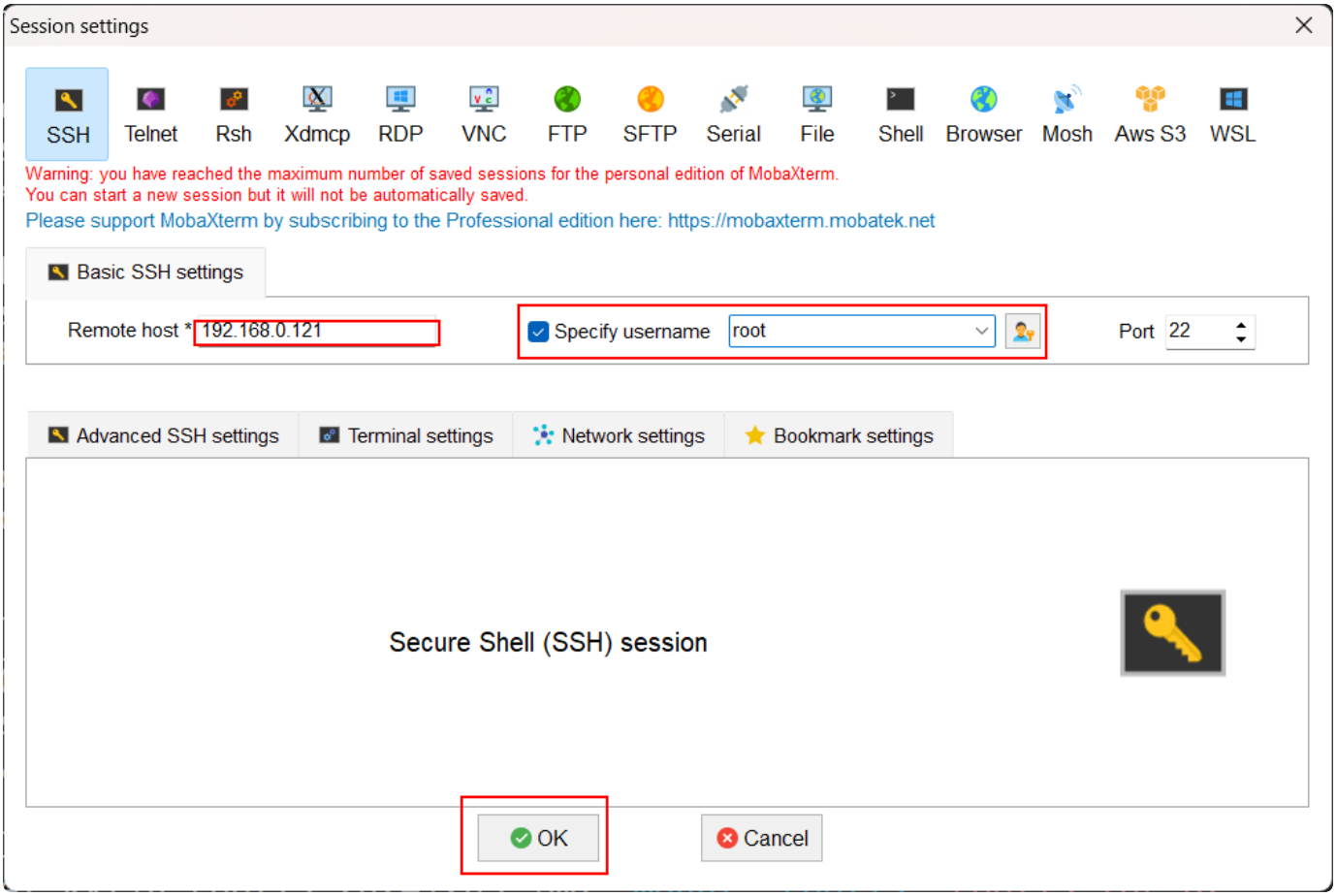
使用调试软件（MobaXterm、putty）等，以MobaXterm为例子，设置参数如下：



SSH登录

在知道IP的前提下，默认可以通过SSH登录进系统。

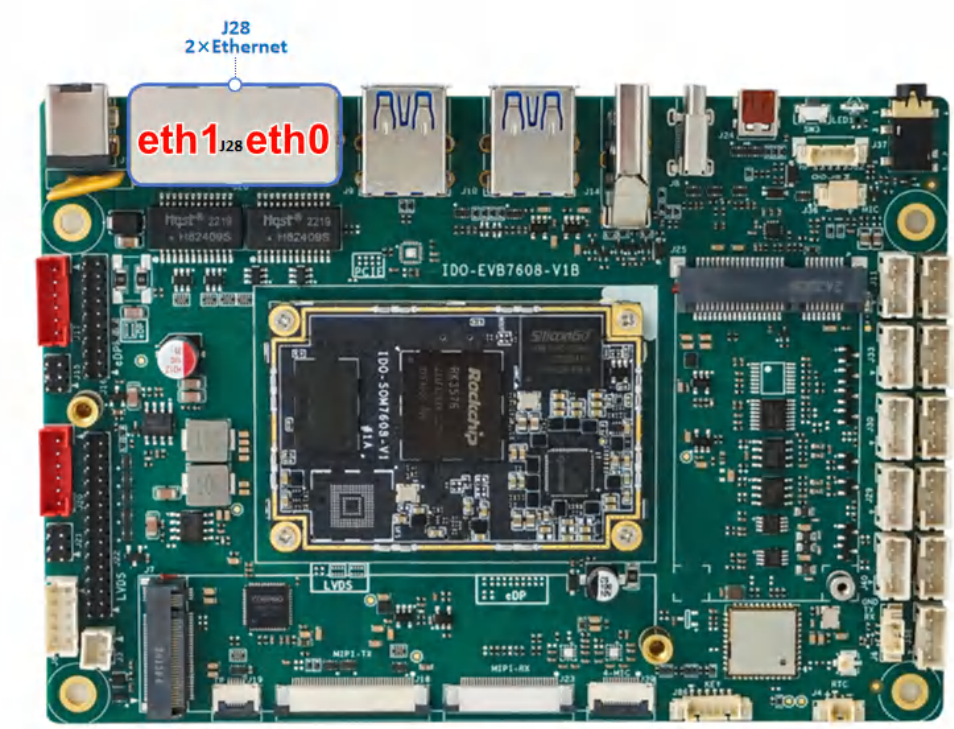
SSH登录账号密码：root/123456



3.2. 网络

3.2.1. 以太网

主板有两路千兆以太网接口位置J28，如下图所示：



设备节点分别为eth0和eth1，以太网接口默认支持DHCP，只需要将以太网接口连接路由器即可为主板动态分配 IP 地址。网络正常连接图标，如下图所示：

Ethernet动态IP设置：

PowerShell

```
#eth0
root@rk3576-buildroot:/# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr CE:40:03:D4:14:E5
          inet addr:192.168.0.121  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::7914:13cf:57f6:547a/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:62198 errors:0 dropped:1579 overruns:0 frame:0
          TX packets:615 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:5921630 (5.6 MiB)  TX bytes:41773 (40.7 KiB)
          Interrupt:68

#eth1
root@rk3576-buildroot:/# ifconfig eth1
eth1      Link encap:Ethernet  HWaddr D2:40:03:D4:14:E5
          inet addr:192.168.0.121  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::7f4b:4008:8cda:3e3f/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:59 errors:0 dropped:1 overruns:0 frame:0
          TX packets:9 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:8690 (8.4 KiB)  TX bytes:1270 (1.2 KiB)
          Interrupt:70
```

Ethernet静态IP设置：

修改/etc/network/interfaces内容如下：

```PowerShell
# interface file auto-generated by buildroot

auto lo
iface lo inet loopback

auto eth1
        iface eth1 inet static
        address 192.168.0.123
        netmask 255.255.255.0
        gateway 192.168.0.1
        nameserver 192.168.0.1

auto eth0
        iface eth0 inet static
        address 192.168.1.123
        netmask 255.255.255.0
        gateway 192.168.1.1
        nameserver 192.168.1.1

source-directory /etc/network/interfaces.d
```

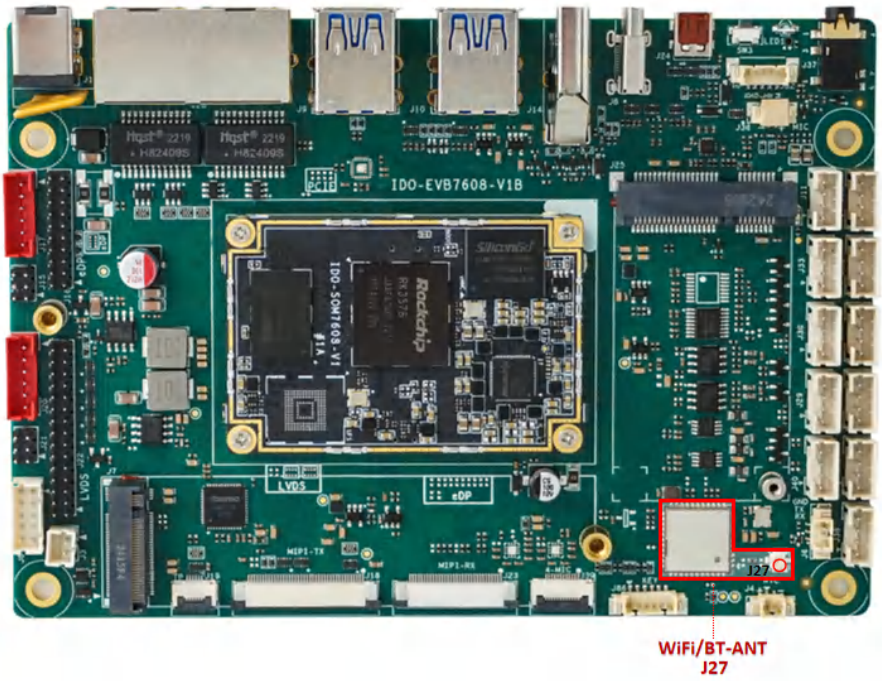
立即生效命令如下：

```Shell
root@rk3576-buildroot:/# /etc/init.d/S40network restart
Stopping network: OK
Starting network: OK
```

设备断电重启，此静态IP设置仍然生效。

3.2.2. WiFi

使用WiFi/蓝牙时，需要连接天线以获得良好的信号，WiFi模块和天线座子J27，如下图所示：



命令行可以使用nmcli工具连接wifi热点：

添加WiFi账号密码：

Shell

```
root@rk3576-buildroot:/# vim /etc/wpa_supplicant.conf
ctrl_interface=/var/run/wpa_supplicant
ap_scan=1

network={
        ssid="SSID"
        psk="PASSWORD"
        key_mgmt=WPA-PSK
}

#ssid: 账号
#psk: 密码
```

扫描附近WiFi：

Shell

```
root@rk3576-buildroot:/# iw dev wlan0 scan | grep SSID
```

WiFi连接：

```shell
root@rk3576-buildroot:/# wpa_supplicant -Dnl80211 -c /etc/wpa_supplicant.conf -i wlan0 &
```

查看连接结果：

```shell
root@rk3576-buildroot:/# wpa_cli -i wlan0 -p /var/run/wpa_supplicant scan
```

扫描周围热点：

```shell
root@rk3576-buildroot:/# wpa_cli -i wlan0 -p /var/run/wpa_supplicant scan_results
```

查看wlan0的IP地址，确认连接成功：

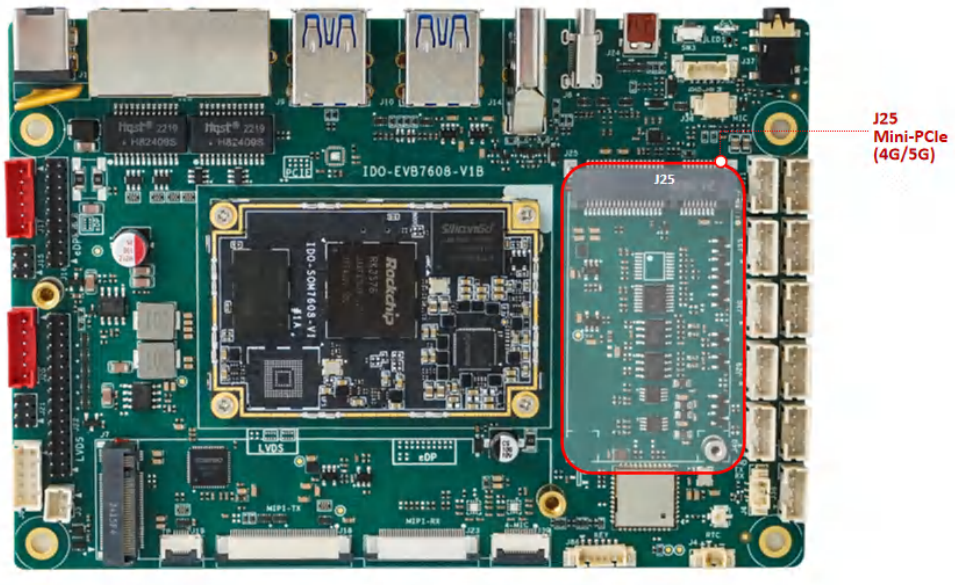
```shell
root@rk3576-buildroot:/# ifconfig wlan0
wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.118  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::1036:80b4:5082:7440  prefixlen 64  scopeid 0x20<link>
        ether c0:f5:35:12:ad:a2  txqueuelen 1000  (Ethernet)
        RX packets 170  bytes 22149 (21.6 KiB)
        RX errors 0  dropped 5  overruns 0  frame 0
        TX packets 24  bytes 2683 (2.6 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@rk3576-buildroot:/# ping www.baidu.com -I wlan0
PING www.wshifen.com (103.235.47.188) from 192.168.0.118 wlan0: 56(84) bytes of data.
64 bytes from 103.235.47.188 (103.235.47.188): icmp_seq=1 ttl=45 time=274 ms
64 bytes from 103.235.47.188 (103.235.47.188): icmp_seq=2 ttl=45 time=299 ms
64 bytes from 103.235.47.188 (103.235.47.188): icmp_seq=3 ttl=45 time=320 ms
64 bytes from 103.235.47.188 (103.235.47.188): icmp_seq=4 ttl=45 time=241 ms
```

3.2.3. Bluetooth

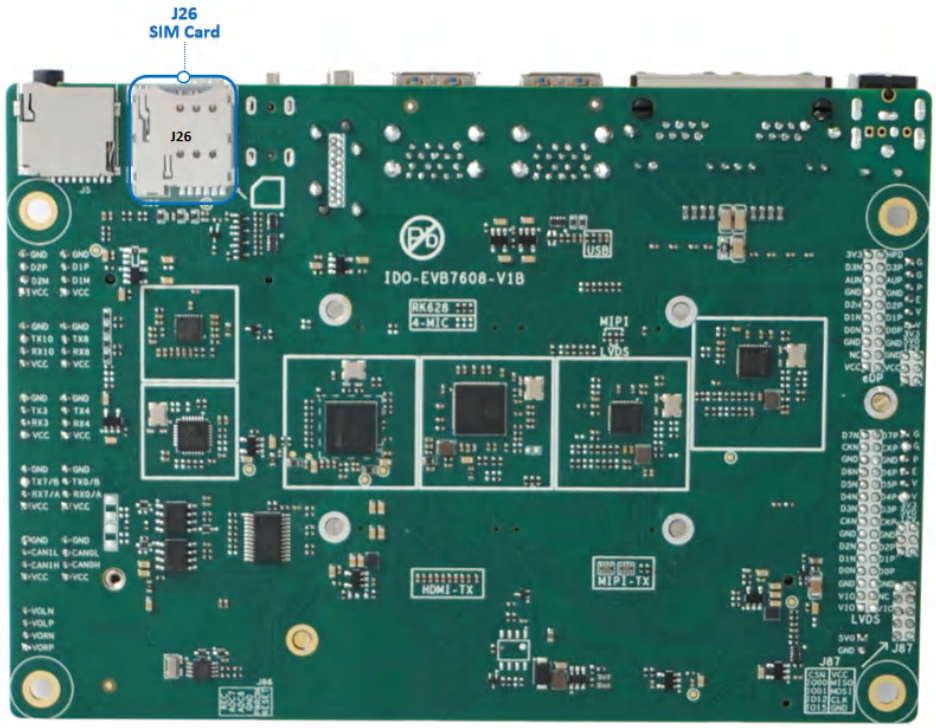
```shell
#打开蓝牙
root@rk3576-buildroot:/# bluetoothctl power on
#扫描蓝牙
root@rk3576-buildroot:/# bluetoothctl scan on
#查看蓝牙设备
root@rk3576-buildroot:/# bluetoothctl devices
#信任蓝牙设备
root@rk3576-buildroot:/# bluetoothctl trust 7C:C1:80:09:DD:6C
#蓝牙配对
root@rk3576-buildroot:/# bluetoothctl pair 7C:C1:80:09:DD:6C
#连接蓝牙
root@rk3576-buildroot:/# bluetoothctl connect 7C:C1:80:09:DD:6C
```

3.2.4. 4G/5G

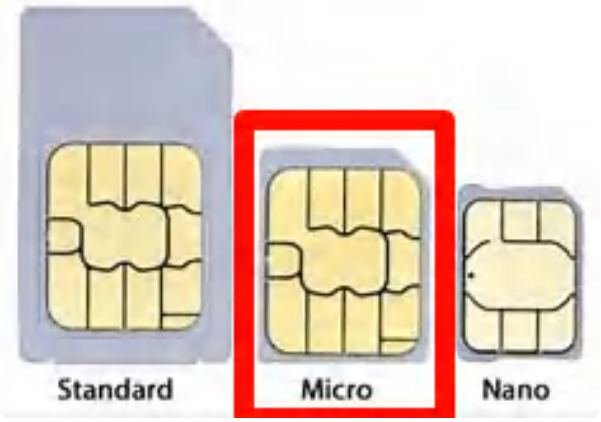
主板内置Mini PCIe 座子扩展 4G/5G模块，4G通信模块适配移远EC20、EC25等通用模组。5G通信模块适配移远RG200U-CN。模块安装位J25，如下图所示：



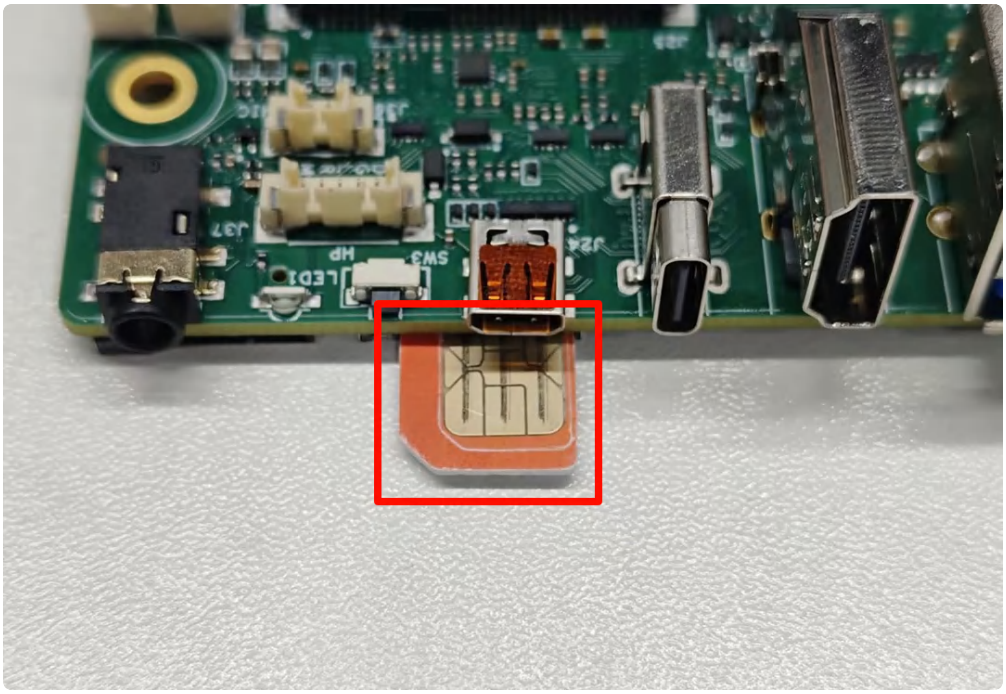
模块需要接上天线以确保获得更好的信号质量，SIM卡安装位J26位于主板背面，如下图所示：



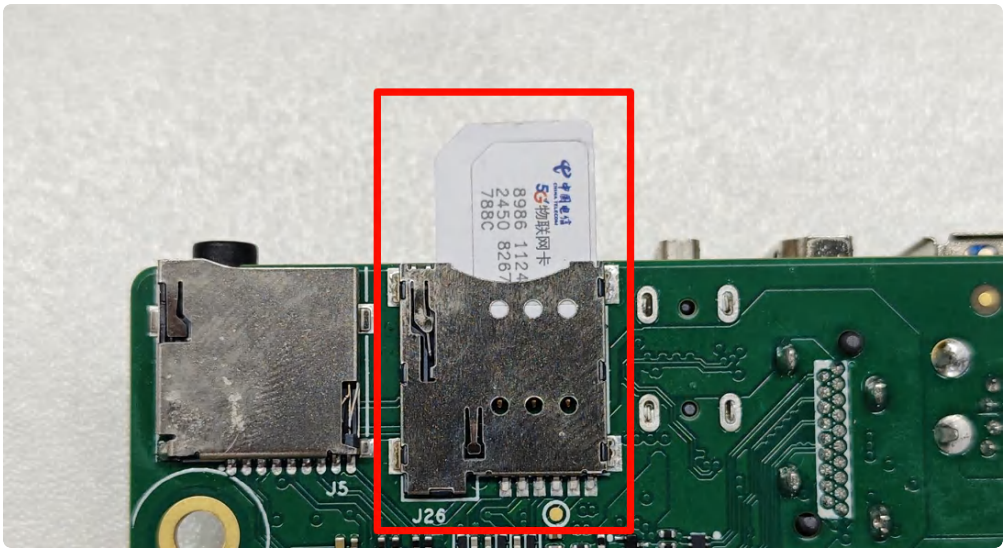
使用Micro尺寸SIM卡，如下图所示：



主板朝上时，SIM卡触点朝上，缺口朝外安装，如下图所示：



主板朝下时，SIM卡触点朝下缺口朝外安装，如下图所示：



4G使用 `ifconfig wwan0` 命令可查看到相关网络信息：

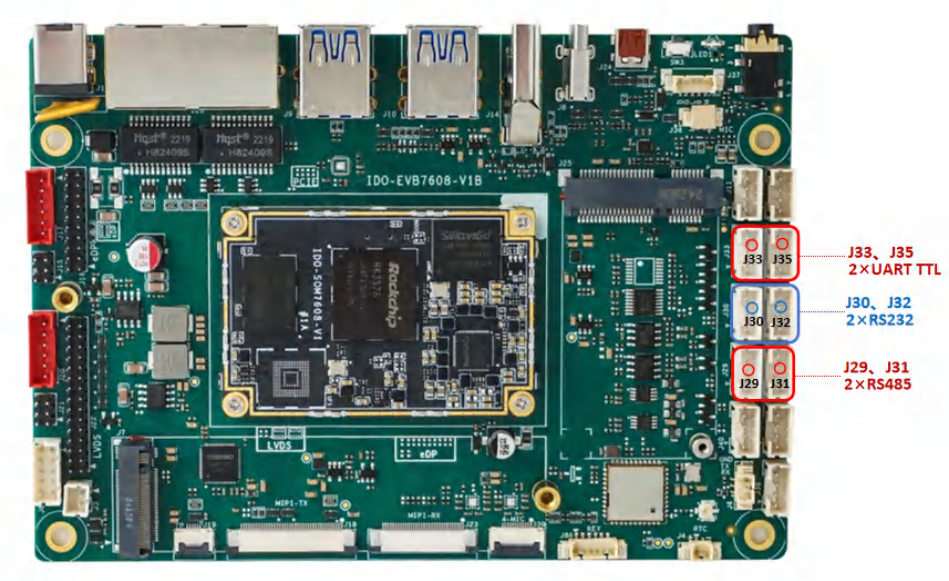
```
#拨号
root@rk3576-buildroot:/# quectel-CM &
#查看网络
root@rk3576-buildroot:/# ifconfig wwan0
wwan0: flags=193<UP,RUNNING,NOARP>  mtu 1500
        inet 10.101.61.51  netmask 255.255.255.248
        inet6 fe80::fc:f6ff:fe8d:bab6  prefixlen 64  scopeid 0x20<link>
        ether 02:fc:f6:8d:ba:b6  txqueuelen 1000  (Ethernet)
        RX packets 42  bytes 7013 (6.8 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 57  bytes 4608 (4.5 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

#测试通信
root@rk3576-buildroot:/# ping www.baidu.com -I wwan0
```

5G使用 `ifconfig usb0` 命令可查看到相关网络信息:

```
#拨号
root@rk3576-buildroot:/# quectel-CM &
#查看网络
root@rk3576-buildroot:/# ifconfig usb0
usb0: flags=193<UP,RUNNING,NOARP>  mtu 1500
        inet 10.101.61.51  netmask 255.255.255.248
        inet6 fe80::fc:f6ff:fe8d:bab6  prefixlen 64  scopeid 0x20<link>
        ether 02:fc:f6:8d:ba:b6  txqueuelen 1000  (Ethernet)
        RX packets 42  bytes 7013 (6.8 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 57  bytes 4608 (4.5 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

#测试通信
root@rk3576-buildroot:/# ping www.baidu.com -I usb0
```

3.3. UART

IDO-EVB7608-V1主板一共引出6路UART（不含DEBUG UART），如下图所示：



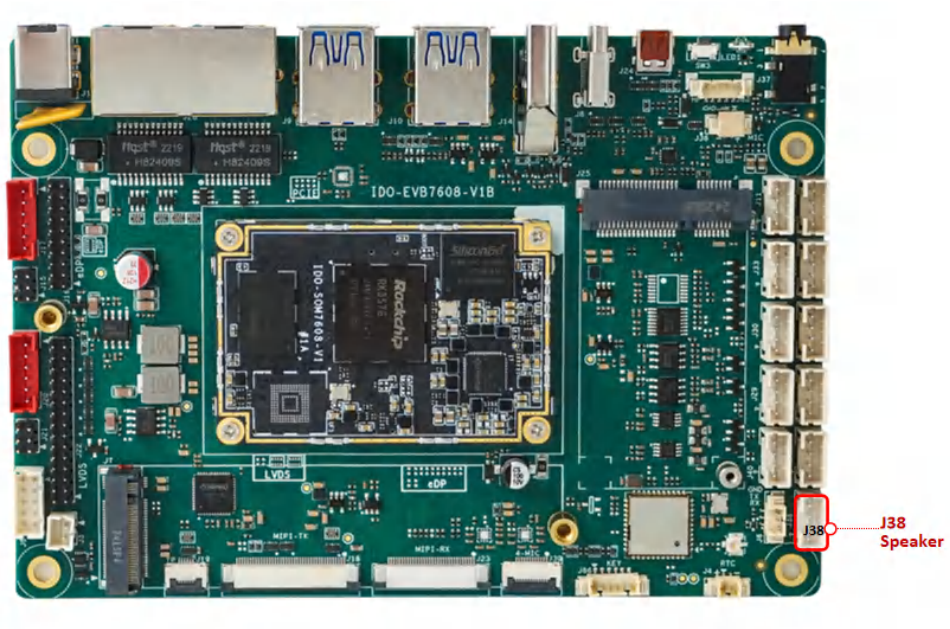
UART设备节点列表如下：

序号	默认电平类型	可改电平类型	设备节点
J33	TTL	可改RS232	/dev/ttyS8
J35	TTL	可改RS232	/dev/ttyS10
J30	RS232	可改TTL	/dev/ttyS4
J32	RS232	可改TTL	/dev/ttyS3
J29	RS485	可改TTL	/dev/ttyS1
J31	RS485	可改TTL	/dev/ttyS7

3.4. 声音

3.4.1. 喇叭

喇叭接口为PH2.0-4P连接器J38，如下图所示：



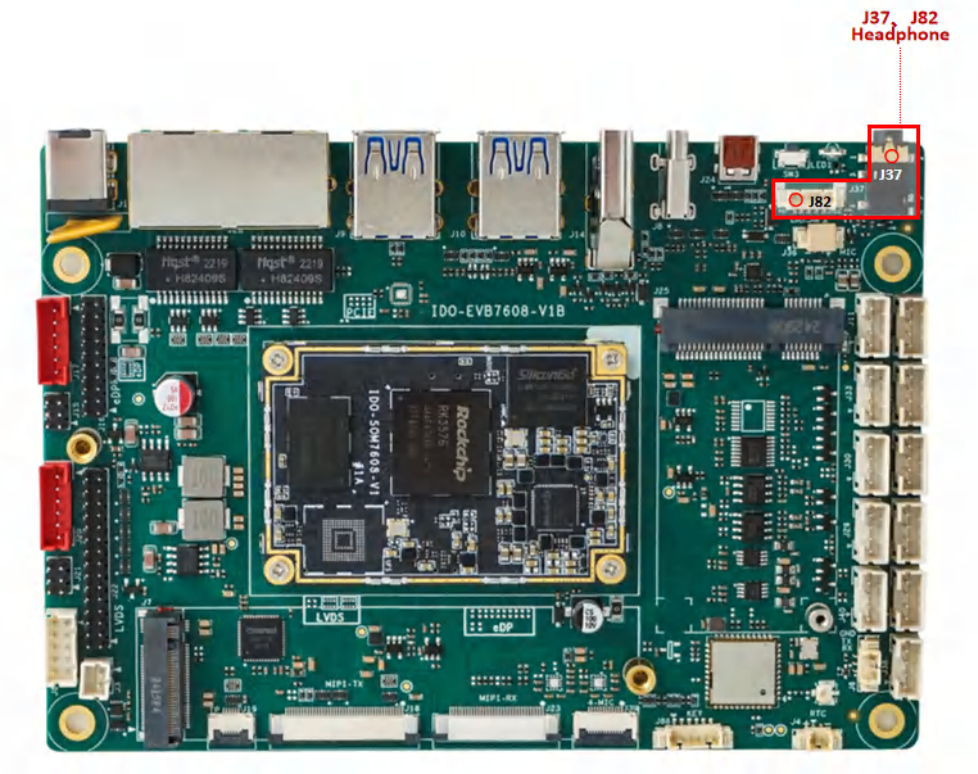
- 支持双声道，每个声道支持4Ω 3W输出

Plain Text

```
#查看声卡
root@rk3576-buildroot:/# aplay -l
**** List of PLAYBACK Hardware Devices ****
card 1: rockchipes8388 [rockchip-es8388], device 0: dailink-multicodecs ES8323 HiFi-0 [dailink-multicodecs ES8323 HiFi-0]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
card 2: rockchiphdmi [rockchip-hdmi], device 0: rockchip-hdmi i2s-hifi-0 [rockchip-hdmi i2s-hifi-0]
  Subdevices: 1/1
  Subdevice #0: subdevice #0

#打开扬声器开关
root@rk3576-buildroot:/# amixer -c 1 cset name='Speaker Switch' on
numid=66,iface=MIXER,name='Speaker Switch'
  ; type=BOOLEAN,access=rw------,values=1
  : values=on
#播放音频
root@linaro-alip:/# aplay -D hw:1,0 xihuangni.wav
Playing WAVE 'xihuangni.wav' : Signed 16 bit Little Endian, Rate 44100 Hz, Stereo
```

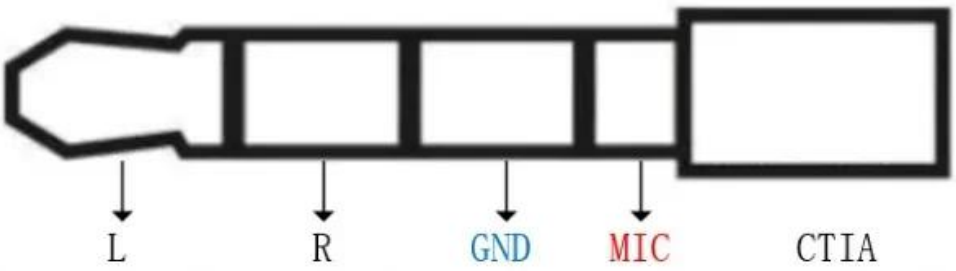
3.4.2. 耳机

主板支持1路3.5mm四节耳机座（CTIA）J37，和1路MX1.25T-5P耳机座子并用，用户可根据需求选用其中1路耳机座子使用 ，如下图所示：



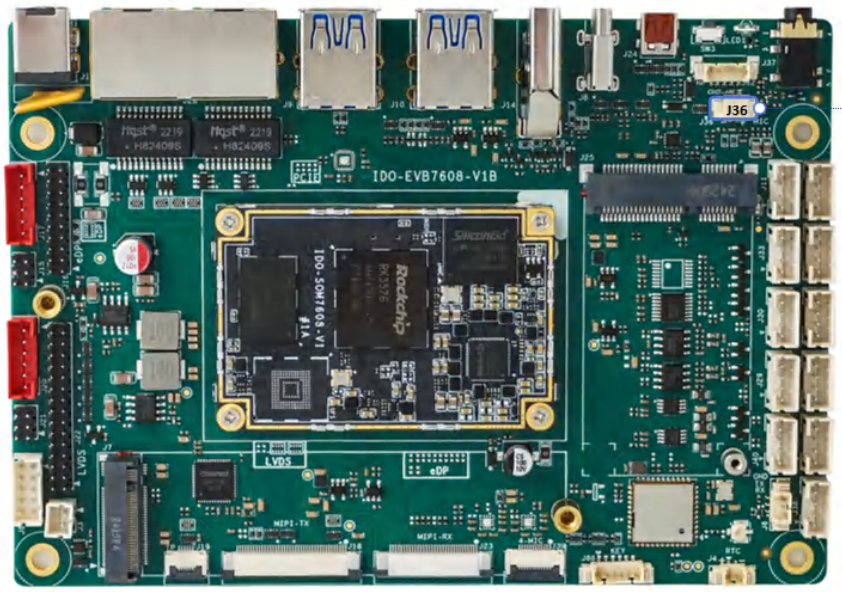
- 支持耳机检测
- 支持耳机录音

CTIA标准四段式耳机，定义如下：



3.4.3. Mic

MX1.25T-2P麦克风接口J36，如下图所示：


J36
Mic 1
J36
IDO-EVB7608-V1B
IDO-SOM7608-V1
LVDS
eDP
MIPI-TX
MIPI-RX

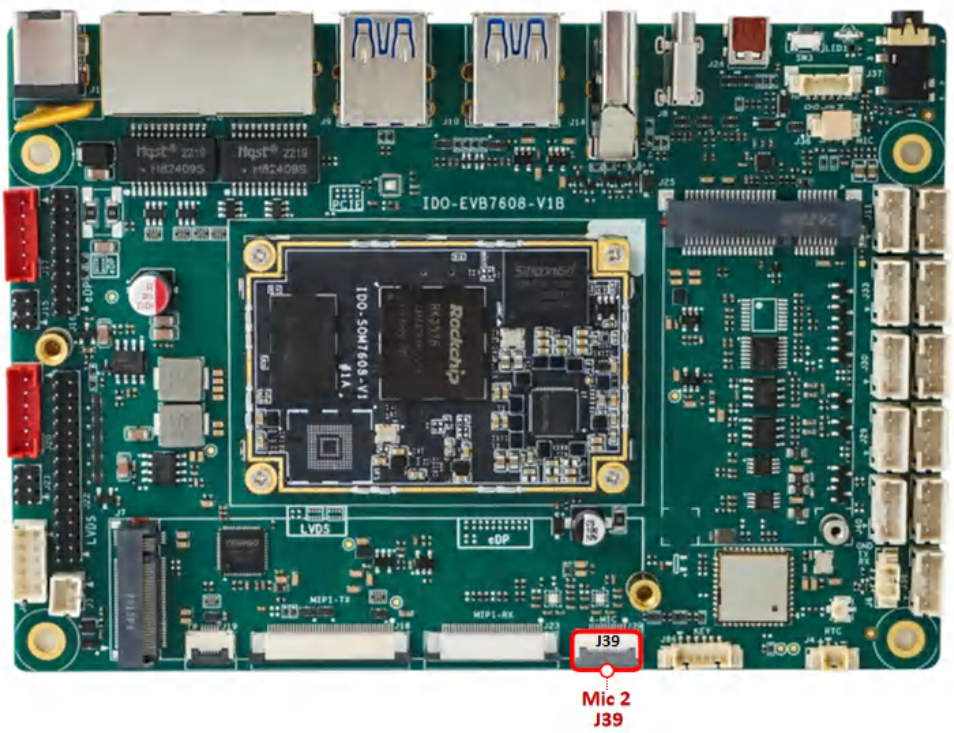
```
#查看录音设备
root@rk3576-buildroot:/# arecord -l
**** List of CAPTURE Hardware Devices ****
card 0: rockchiprk628hd [rockchip,rk628hdmiin], device 0: rockchip,rk628hdmiin i2s-hifi-0 [rockchip,rk628hdmiin i2s-hifi-0]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
card 1: rockchipes8388 [rockchip-es8388], device 0: dailink-multicodecs ES8323 HiFi-0 [dailink-multicodecs ES8323 HiFi-0]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
card 2: rockchiphdmi [rockchip-hdmi], device 0: rockchip-hdmi i2s-hifi-0 [rockchip-hdmi i2s-hifi-0]
  Subdevices: 1/1
  Subdevice #0: subdevice #0



#捕获音量设置(不要设置最大值):
root@rk3576-buildroot:/# amixer -c 1 cset numid=50 100
numid=50,iface=MIXER,name='Capture Digital Volume'
  ; type=INTEGER,access=rw---R--,values=2,min=0,max=192,step=0
  : values=100,100
  | dBscale-min=-96.00dB,step=0.50dB,mute=1

root@rk3576-buildroot:/# amixer -c 1 cget numid=52
numid=52,iface=MIXER,name='Left Channel Capture Volume'
  ; type=INTEGER,access=rw---R--,values=1,min=0,max=8,step=0
  : values=0
  | dBscale-min=0.00dB,step=3.00dB,mute=0

root@rk3576-buildroot:/# amixer -c 1 cget numid=53
numid=53,iface=MIXER,name='Right Channel Capture Volume'
  ; type=INTEGER,access=rw---R--,values=1,min=0,max=8,step=0
  : values=0
  | dBscale-min=0.00dB,step=3.00dB,mute=0

自动增益控制(打开):
root@rk3576-buildroot:/# amixer -c 1 cset numid=41 3
numid=41,iface=MIXER,name='ALC Capture Function'
  ; type=ENUMERATED,access=rw------,values=1,items=4
  ; Item #0 'Off'
  ; Item #1 'Right'
  ; Item #2 'Left'
```

```
  ; Item #3 'Stereo'
  : values=3

调整目标音量:
root@rk3576-buildroot:/# amixer -c 1 cset numid=38 8
numid=38,iface=MIXER,name='ALC Capture Target Volume'
  ; type=INTEGER,access=rw------,values=1,min=0,max=15,step=0
  : values=8
设置噪声门限值:
root@rk3576-buildroot:/# amixer -c 1 cset numid=46 15
numid=46,iface=MIXER,name='ALC Capture NG Threshold'
  ; type=INTEGER,access=rw------,values=1,min=0,max=31,step=0
  : values=15
确保捕获静音未开启:
root@rk3576-buildroot:/# amixer -c 1 cset numid=51 off
numid=51,iface=MIXER,name='Capture Mute'
  ; type=BOOLEAN,access=rw------,values=1
  : values=off

#主mic开关（mic录音需要打开，耳机录音需要关闭）:
root@rk3576-buildroot:/# amixer -c 1 cset  numid=67 on
numid=67,iface=MIXER,name='Main Mic Switch'
  ; type=BOOLEAN,access=rw------,values=1
  : values=on
root@rk3576-buildroot:/# echo -n "mic2" > /sys/class/es8388_mic/mic_channe
l
root@rk3576-buildroot:/# arecord -D hw:1,0 -f S16_LE -r 16000 -c 2 test_mi
c2.wav


#耳机录音:
root@rk3576-buildroot:/# amixer -c 1 cset  numid=67 off
root@rk3576-buildroot:/# echo -n "mic1" > /sys/class/es8388_mic/mic_channe
l
root@rk3576-buildroot:/# arecord -D hw:1,0 -f S16_LE -r 16000 -c 2 test_mi
c1.wav
```

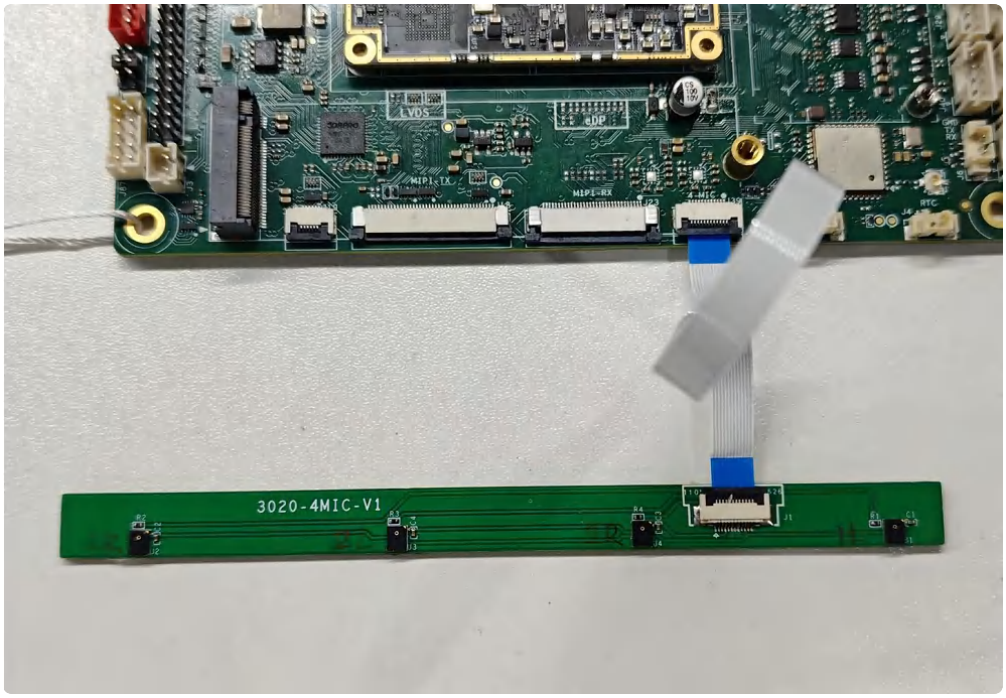
3.4.4. PDM-Mic

注意：PDM-Mic与HDMI-RX功能二选一，软硬件默认配置为HDMI-RX功能。

主板预留 PDM-Mic 阵列接口J39，如下图所示：

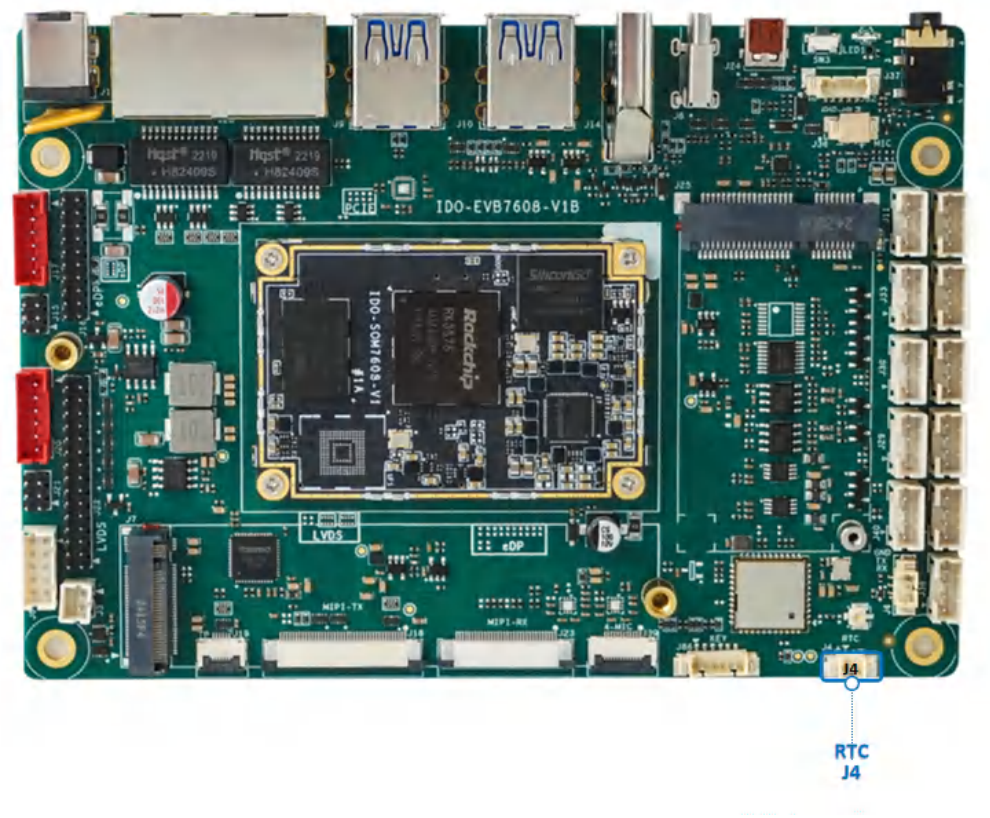


采用FPC05-FDW-12P座子连接Mic阵列，如下图所示：

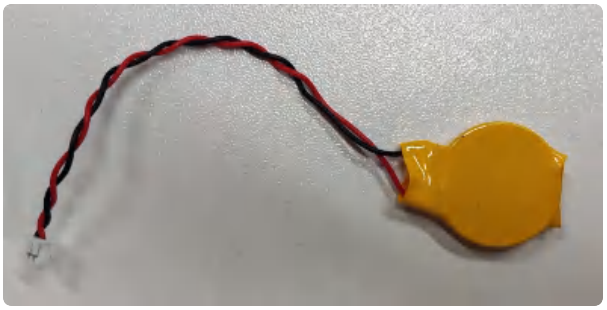


3.5. RTC

MX1.25T-2P RTC电池座J4，如下图所示：



连接3V 纽扣电池，RTC电池参考如下



rtc时间设置方法：

Plain Text

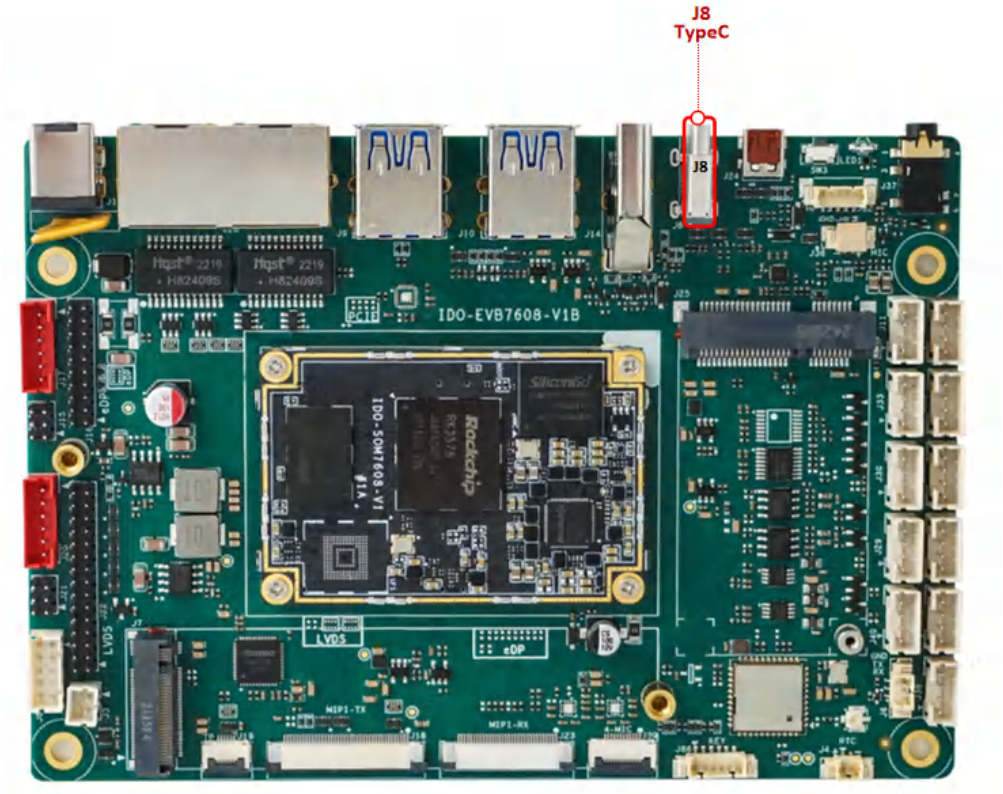
```
#设置时间
root@rk3576-buildroot:/# date -s "2024-10-09 14:02:30"

#将rtc时钟调整为与目前的系统时钟一致
root@rk3576-buildroot:/# hwclock -w

#获取硬件rtc当前时间（断电重启读取时间没有太大偏差）
root@rk3576-buildroot:/# hwclock -r
2024-10-09 14:02:35.945604+00:00
```

3.6. USB接口

主板支持1个TypeC接口（USB3.2 Gen1 OTG+DP1.4），支持4个USB3.0-A接口，2个USB2.0PH2.0-4P 接口，USB对外总供电应小于4A。

3.6.1. TypeC接口

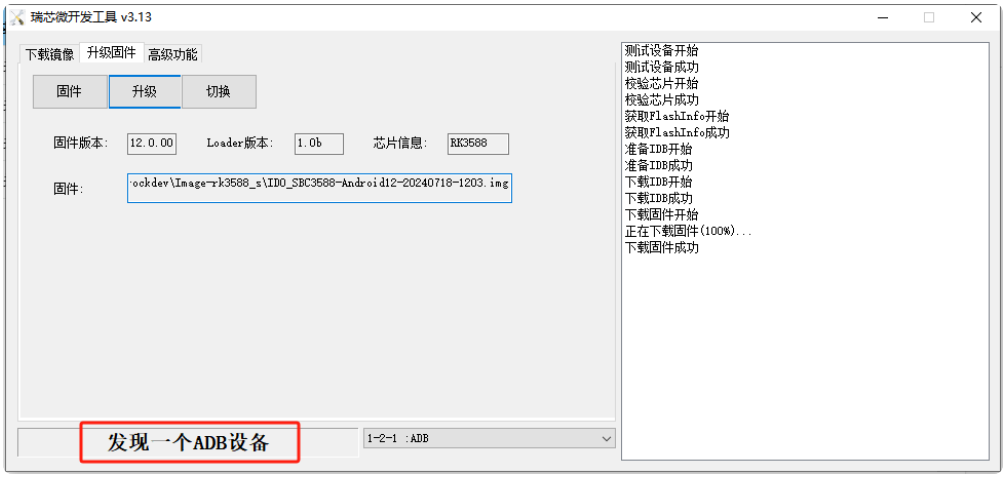
TypeC接口（USB3.2 Gen1 OTG+DP1.4输出）J8，如下图所示：



- 支持Host、Device模式自动切换
- 支持DP显示输出

Device从机模式

使用TypeC数据线连接电脑，烧录工具能发现一个ADB设备，如下图所示：



- 可使用ADB相关开发工具对盒子进行功能调试

Host主机模式

接入TypeC设备，或通过TypeC to USB-A转接头接入USB外设，如下图所示：

- 可识别U盘、键盘、鼠标并正常使用

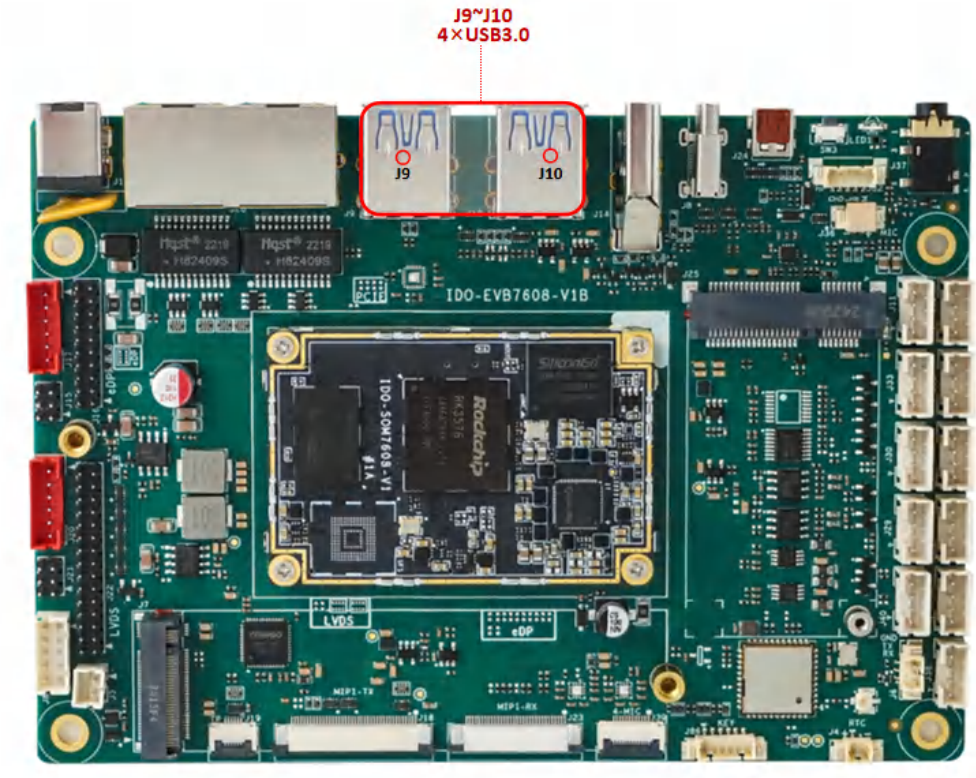
DP模式

通过TypeC全功能数据线接入DP显示器，或通过TYPE-C to HDMI数据线连接HDMI显示器

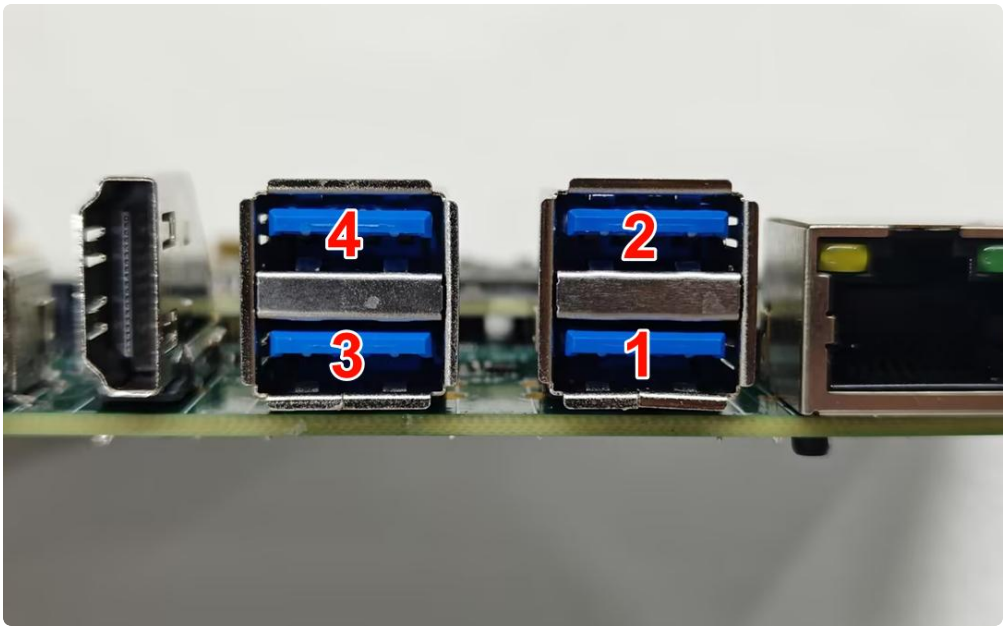
- 主板画面通过TYPE-C DP输出正常，DP声音输出正常

3.6.2. USB3.0接口

主板支持4个USB3.0接口，接口为标准的双层A口J9、J10，如下图所示：



USB母座提供5V@1A供电能力，每个USB端口供电可独立控制，USB序号如下图所示：



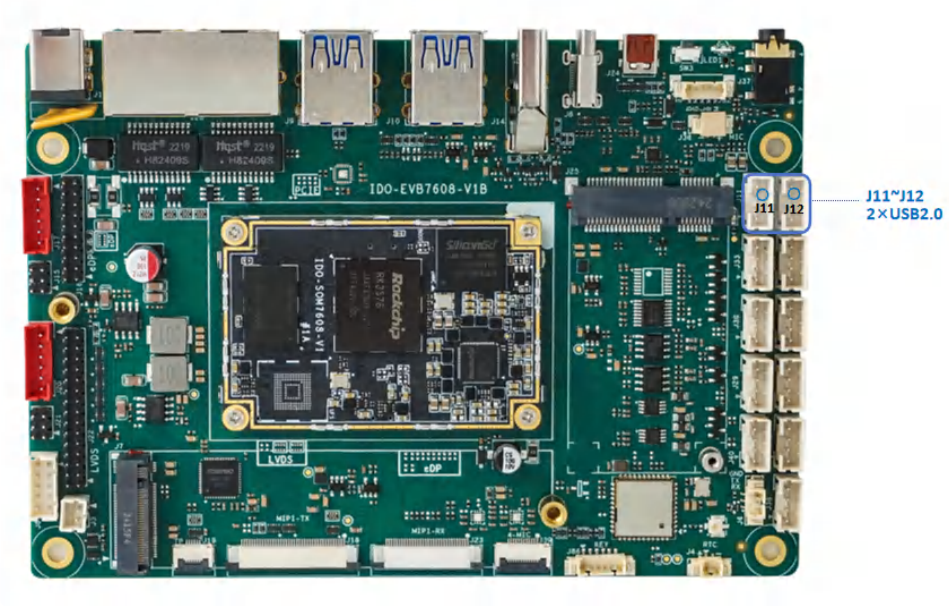
USB电源控制，如下表所示：

USB端口	默认状态	动作	命令
USB1	开启	关闭电源	echo 0 > /sys/class/leds/usb1_pwr/brightness
		开启电源	echo 1 > /sys/class/leds/usb1_pwr/brightness
USB2	开启	关闭电源	echo 0 > /sys/class/leds/usb2_pwr/brightness
		开启电源	echo 1 > /sys/class/leds/usb2_pwr/brightness
USB3	开启	关闭电源	echo 0 > /sys/class/leds/usb3_pwr/brightness
		开启电源	echo 1 > /sys/class/leds/usb3_pwr/brightness
USB4	开启	关闭电源	echo 0 > /sys/class/leds/usb4_pwr/brightness
		开启电源	echo 1 > /sys/class/leds/usb4_pwr/brightness

供电控制说明：设备节点写"0"关闭电源，写"1"开启电源

3.6.3. USB2.0接口

主板配置了2路USB2.0 PH2.0-4P接口，USB接口均提供5V@1A的驱动能力，如下图所示：

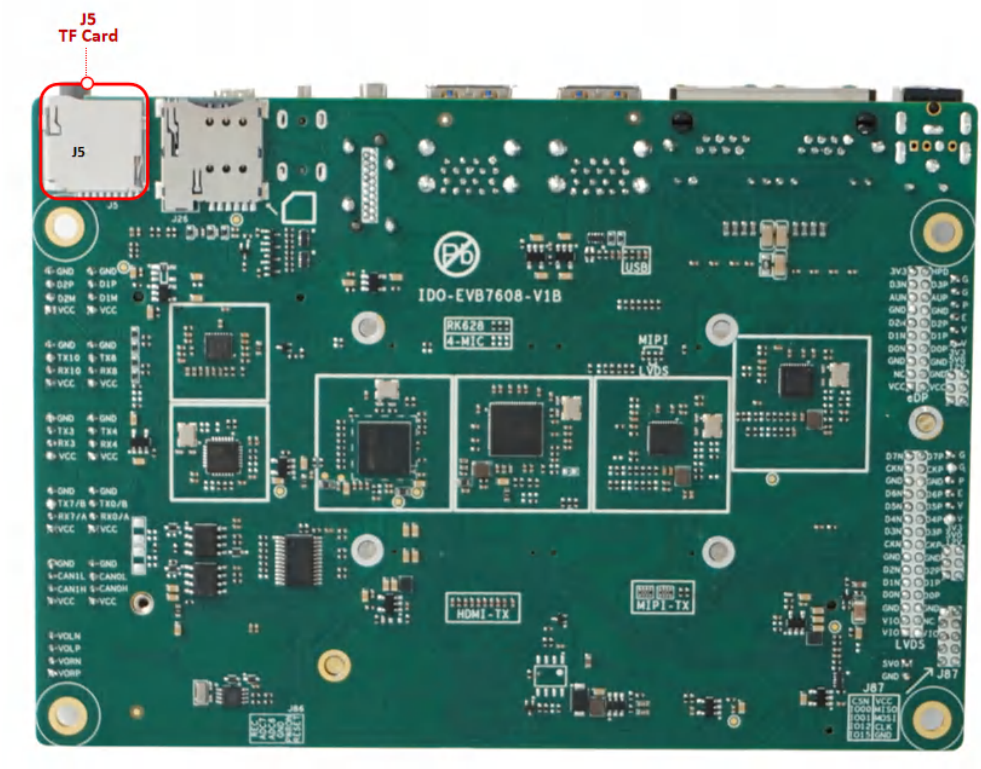


USB端口	默认状态	动作	命令
USB J11	开启	关闭电源	echo 0 > /sys/class/leds/usb5_pwr/brightness
		开启电源	echo 1 > /sys/class/leds/usb5_pwr/brightness
USB J12	开启	关闭电源	echo 0 > /sys/class/leds/usb6_pwr/brightness
		开启电源	echo 1 > /sys/class/leds/usb6_pwr/brightness

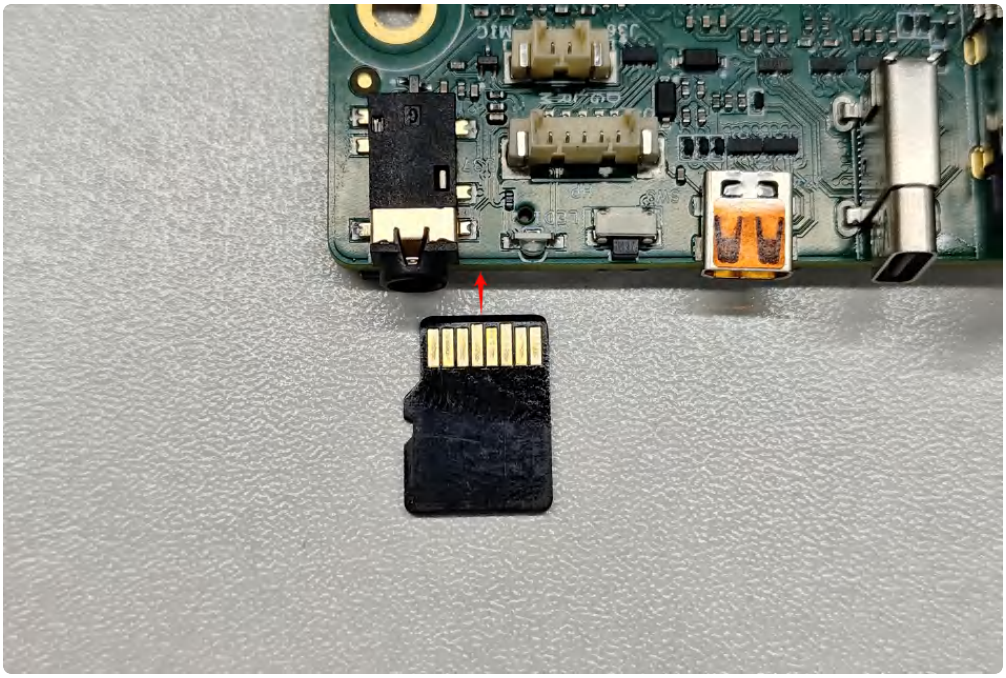
供电控制说明：设备节点写"0"关闭电源，写"1"开启电源

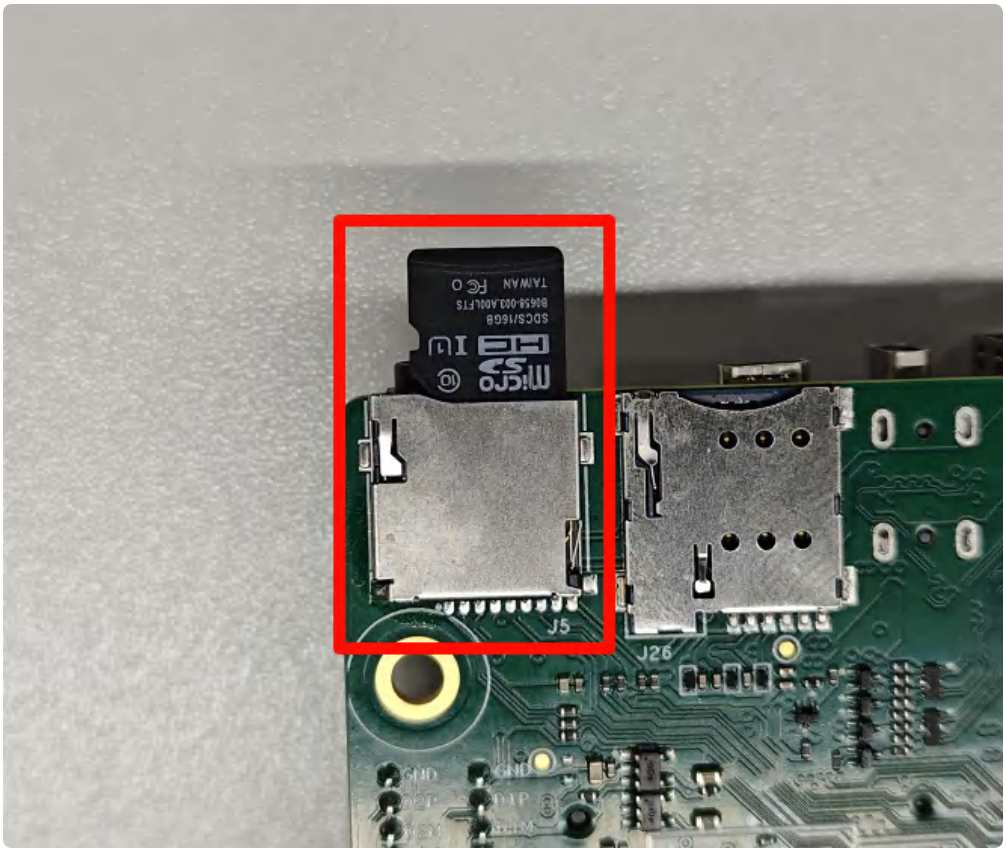
3.7. TF Card

TF卡座支持SDIO3.0， 支持高速SD卡，接口位于主板背面J5，如下图所示：



TF卡安装方向，主板正面朝上时TF卡触点朝上，如下图所示：



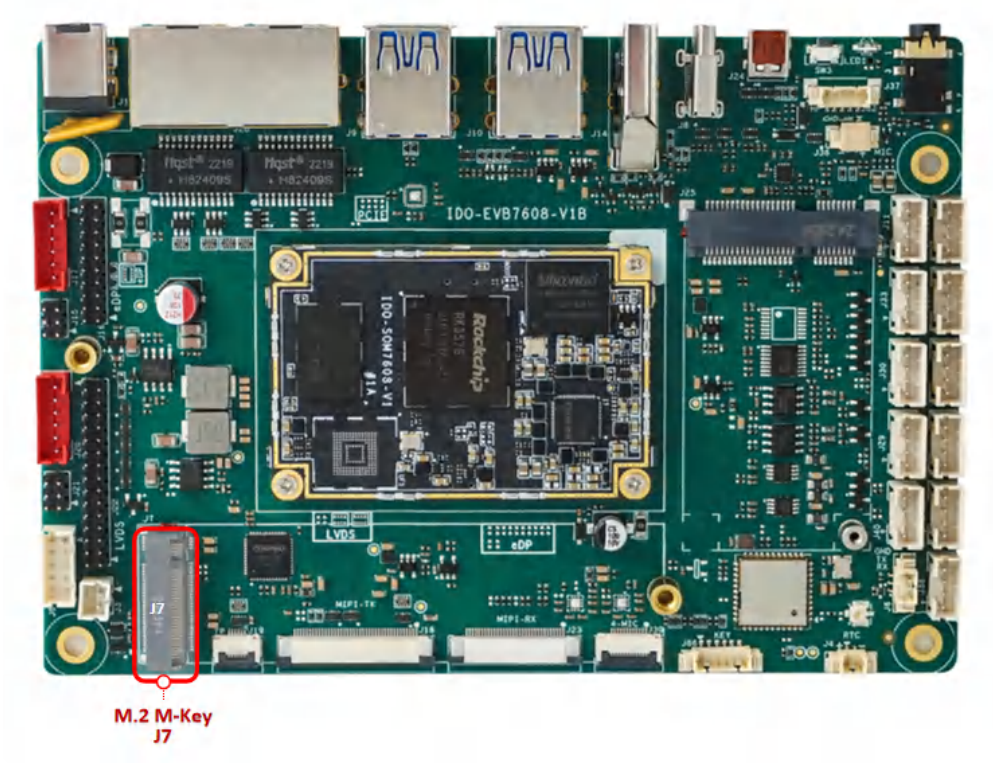

J5
J26

```shell
root@rk3576-buildroot:/# fdisk -l
Found valid GPT with protective MBR; using GPT

Disk /dev/mmcblk2: 61071360 sectors, 1148M
Logical sector size: 512
Disk identifier (GUID): 21130000-0000-4444-8000-05e1000030fd
Partition table holds up to 128 entries
First usable sector is 34, last usable sector is 61071326

Number  Start (sector)    End (sector)  Size Name
     1           16384           24575 4096K uboot
     2           24576           32767 4096K misc
     3           32768          163839 64.0M boot
     4          163840          425983  128M recovery
     5          425984          491519 32.0M backup
     6          491520         8880127 4096M userdata
     7         8880128         9142271  128M oem
     8         9142272        61071295 24.7G rootfs
...
Device       Boot StartCHS    EndCHS        StartLBA     EndLBA    Sectors  Size Id Type
/dev/mmcblk1p1    0,2,10      239,254,63         135     3858623    3858849 1884M  e Win95 FAT16 (LBA)
```

注意：因为开机过程中由于emmc分区的变化，可能导致tf卡的分区也会变化，导致无法自动挂载，手动挂载即可。

3.8. M.2接口

IDO-EVB7608-V1主板上使用标准M.2-M-key M.2接口连接座，如下图所示：



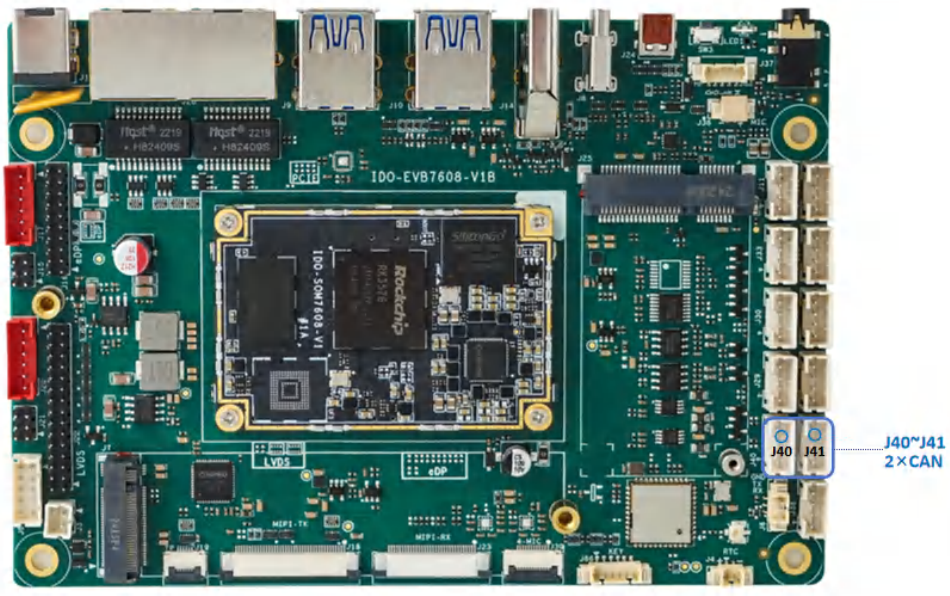
支持PCIe2.1通信，适用2280尺寸NVME固态硬盘。

Shell

```
root@rk3576-buildroot:/# fdisk -l
Disk /dev/ram0: 8 MiB, 8388608 bytes, 16384 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes


Disk /dev/ram1: 8 MiB, 8388608 bytes, 16384 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes
...
Disk /dev/nvme0n1: 476.94 GiB, 512110190592 bytes, 1000215216 sectors
Disk model: XPG GAMMIX S11L
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x00000000

Device         Boot Start        End    Sectors   Size Id Type
/dev/nvme0n1p1       2048 1000215182 1000213135 476.9G  c W95 FAT32 (LBA)
```

3.9. CAN接口

主板共有2路CAN接口J40、J41，如下图所示:



分别对应系统节点名称为 can0、can1。

1. 设置CAN参数

Shell

```
# 查看can0配置信息，可以看到can节点的波特率、位时序设置等
# ip -d -s -s link show can0

# 配置can之前需要先把can节点关闭
# ip link set can0 down

# 设置can0异常10ms重启
# ip link set can0 type can restart-ms 10

# 设置can0的比特率为1 Mbps/s
# ip link set can0 type can bitrate 1000000 dbitrate 1000000 fd on

# 打开can0节点
# ip link set can0 up
```

2. 测试收发

两台机器通过收发器连接好，发送方的 can_H 与接收方的 can_H连接，发送方的can_L 与接收方的 can_L连接。

can0 Shell

```
# can0节点发送数据
root@rk3576-buildroot:/# ip link set can0 down
root@rk3576-buildroot:/# ip link set can0 type can bitrate 1000000 dbitrate 1000000 fd on
root@rk3576-buildroot:/# ip link set can0 up
root@rk3576-buildroot:/# cansend can0 123#DEADBEEF
root@rk3576-buildroot:/# cansend can0 123#DEADBEEF12345679

# can0接收数据
root@rk3576-buildroot:/# ip link set can0 down
root@rk3576-buildroot:/# ip link set can0 type can bitrate 1000000 dbitrate 1000000 fd on
root@rk3576-buildroot:/# ip link set can0 up
root@rk3576-buildroot:/# candump can0
```

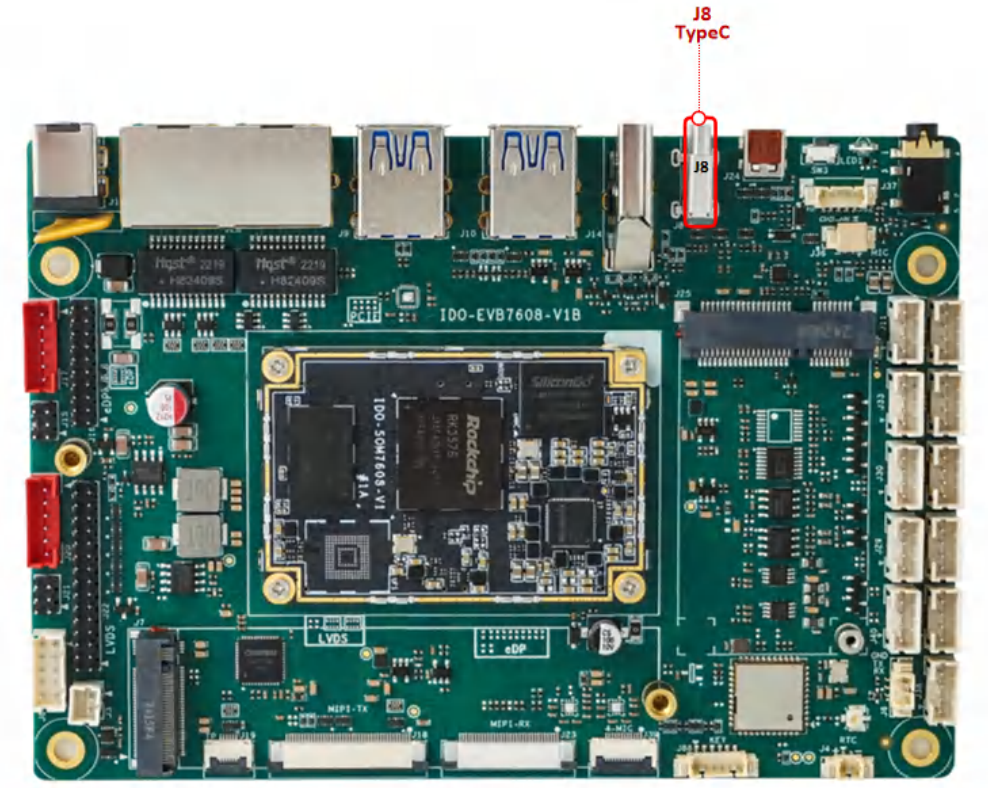
can1 Shell

```
# can1节点发送数据
root@rk3576-buildroot:/# ip link set can1 down
root@rk3576-buildroot:/# ip link set can1 type can bitrate 1000000 dbitrate 1000000 fd on
root@rk3576-buildroot:/# ip link set can1 up
root@rk3576-buildroot:/# cansend can1 123#DEADBEEF
root@rk3576-buildroot:/# cansend can1 123#DEADBEEF11122233

# can1接收数据
root@rk3576-buildroot:/# ip link set can1 down
root@rk3576-buildroot:/# ip link set can1 type can bitrate 1000000 dbitrate 1000000 fd on
root@rk3576-buildroot:/# ip link set can1 up
root@rk3576-buildroot:/# candump can1
```

3.10. LCD显示

3.10.1. DP

DP接口（USB-C）J8，如下图所示：

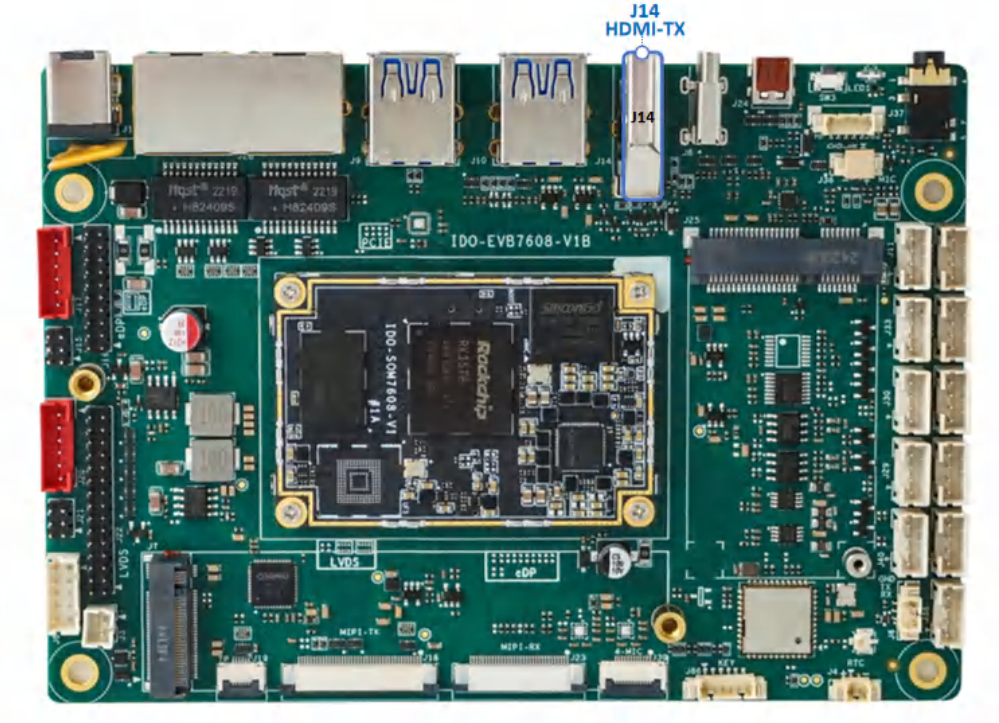


可以使用 USB-C 转 HDMI 高清线连接 HDMI 显示器输出画面，如下图所示：

- 最高支持4K@30fps输出

3.10.2. HDMI-TX

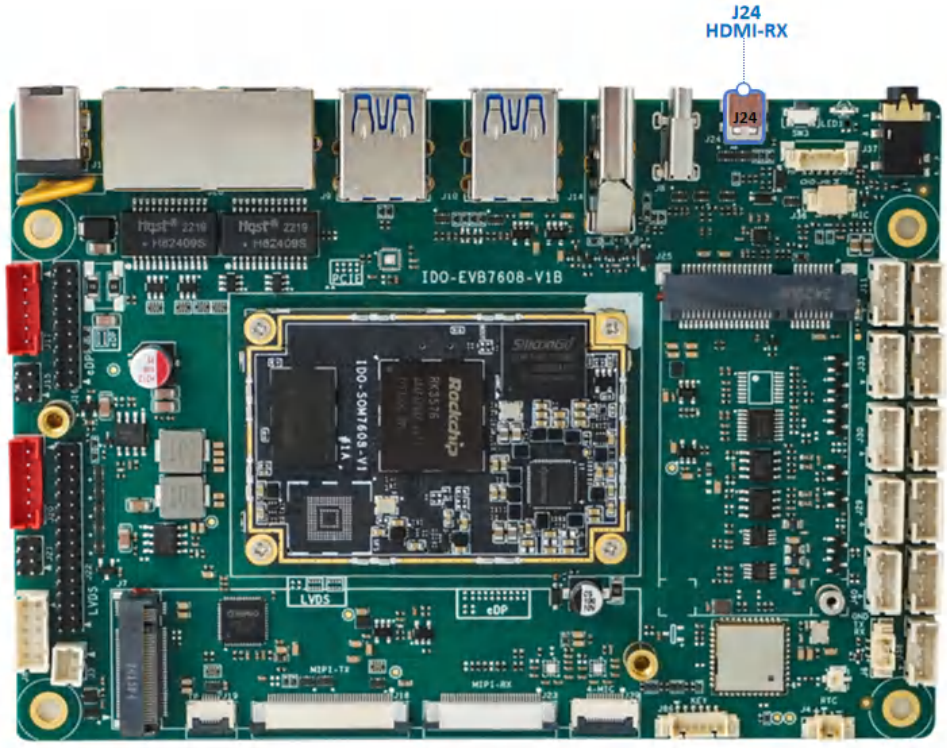
标准HDMI-A接口J14，如下图所示：



- 最高支持 HDMI2.1 8K@60fps 输出

3.10.3. HDMI-RX接口

Micro HDMI接口J24，如下图所示：



预览：

```
root@rk3576-buildroot:/# gst-launch-1.0 v4l2src device=/dev/video0 ! video/x-raw,width=1920,height=1080,framerate=30/1 ! videoconvert ! autovideosink
```

- HDMI2.0-RX，支持4K@30fps

录音：

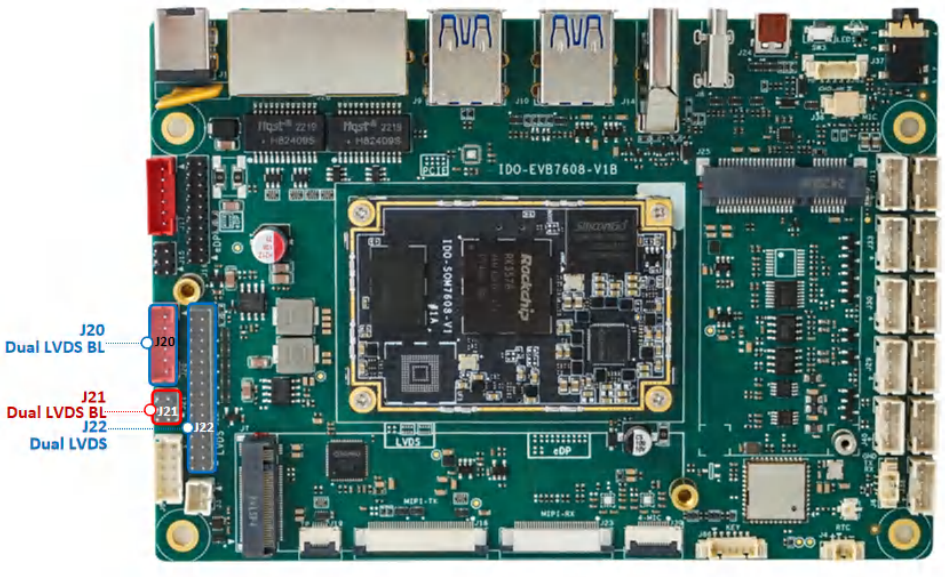
```
#窗口1
root@rk3576-buildroot:/# gst-launch-1.0 v4l2src device=/dev/video0 ! video/x-raw,width=1920,height=1080,framerate=30/1 ! videoconvert ! autovideosink
#窗口2
root@rk3576-buildroot:/# arecord -D hw:0,0 -f S16_LE -r 48000 -c 2 test.wav
```

注意：hdmi in录音之前确保另一端的hdmi out已经正常输出音频。

3.10.4. Dual LVDS屏

注意：Dual LVDS接口与MIPI-TX二选一，硬件默认配置为MIPI-TX。

Dual LVDS接口J22，背光接 J20，屏幕供电电压选择接口J21，如下图所示：

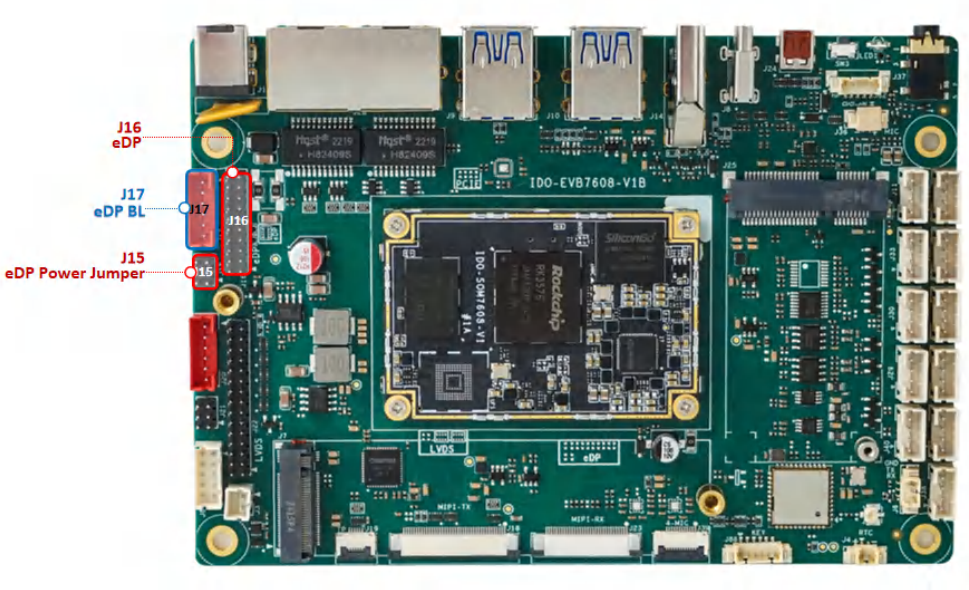


J21屏幕供电电压选择接口，接屏前，需要根据具体的屏幕规格书来确认J21的供电跳线帽接到3.3V、5V或12V，默认3.3V。

3.10.5. eDP

注意：eDP 接口与 HDMI-TX 二选一，硬件默认配置为 HDMI-TX。

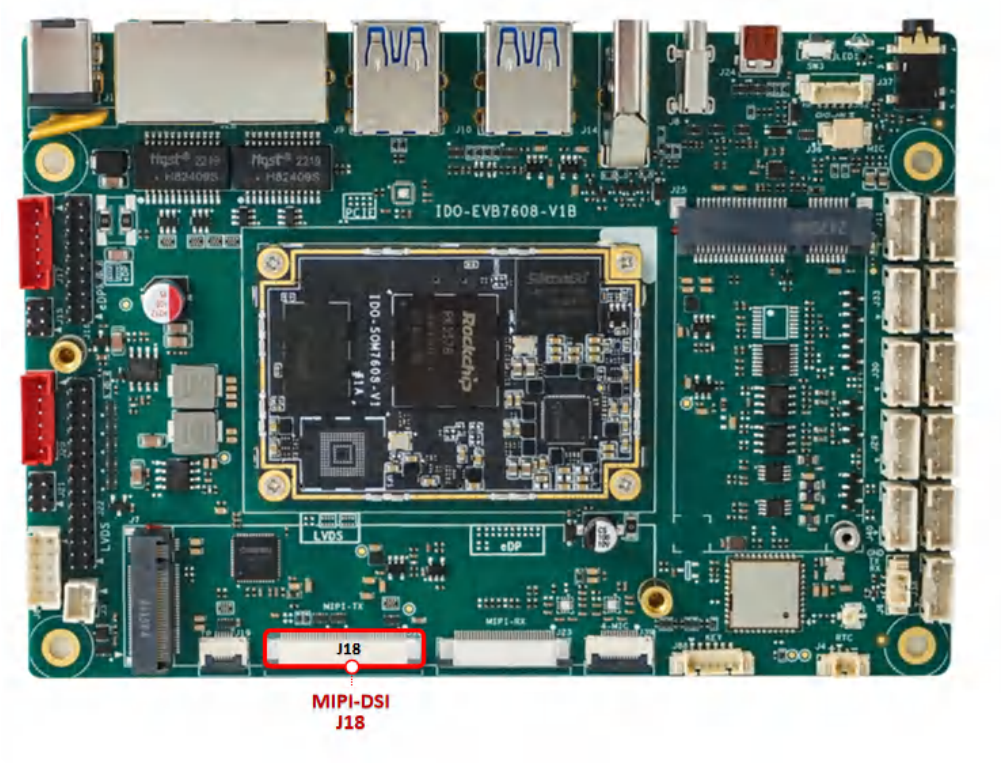
eDP接口J16，eDP背光接 J17，eDP屏幕供电电压选择接口J15，如下图所示：



J15屏幕供电电压选择接口，接屏前，需要根据具体的屏幕规格书来确认J15的供电跳线帽接到3.3V、5V或12V，默认3.3V。

3.10.6. MIPI

J18 MIPI接口为40Pin FPC 0.5mm 上接，如下图所示：

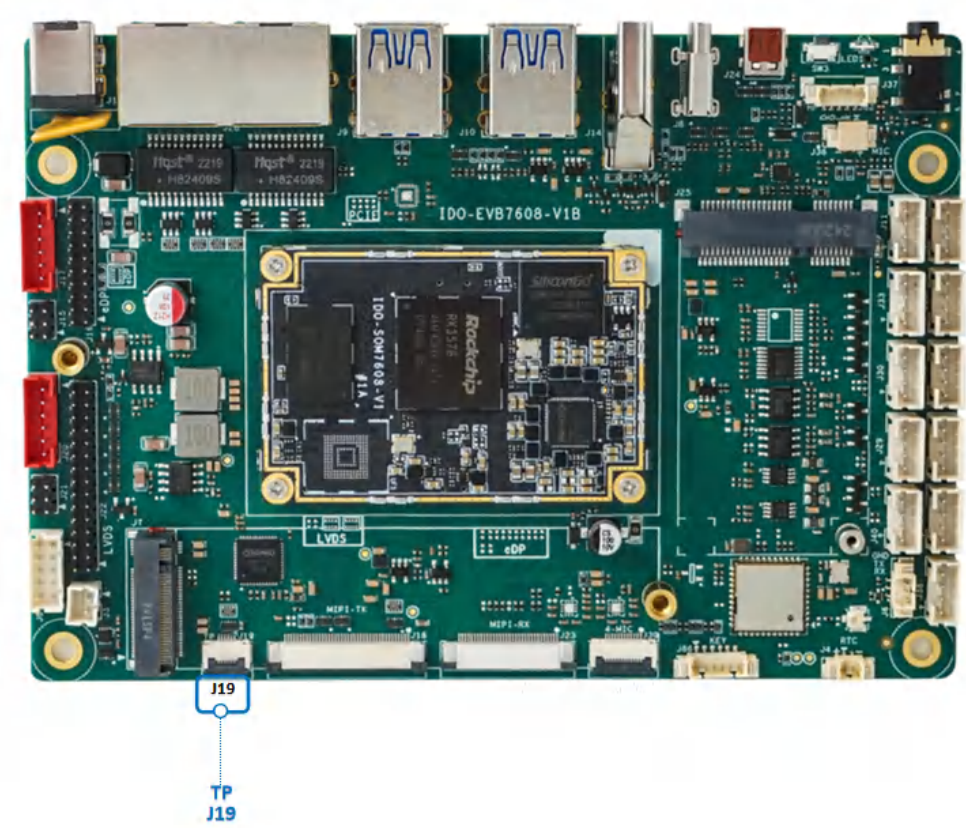


- MIPI供电为3.3V

注意：开发板的MIPI TX和Dual LVDS输出为复用关系，主板默认MIPI TX输出。

3.11. TP接口

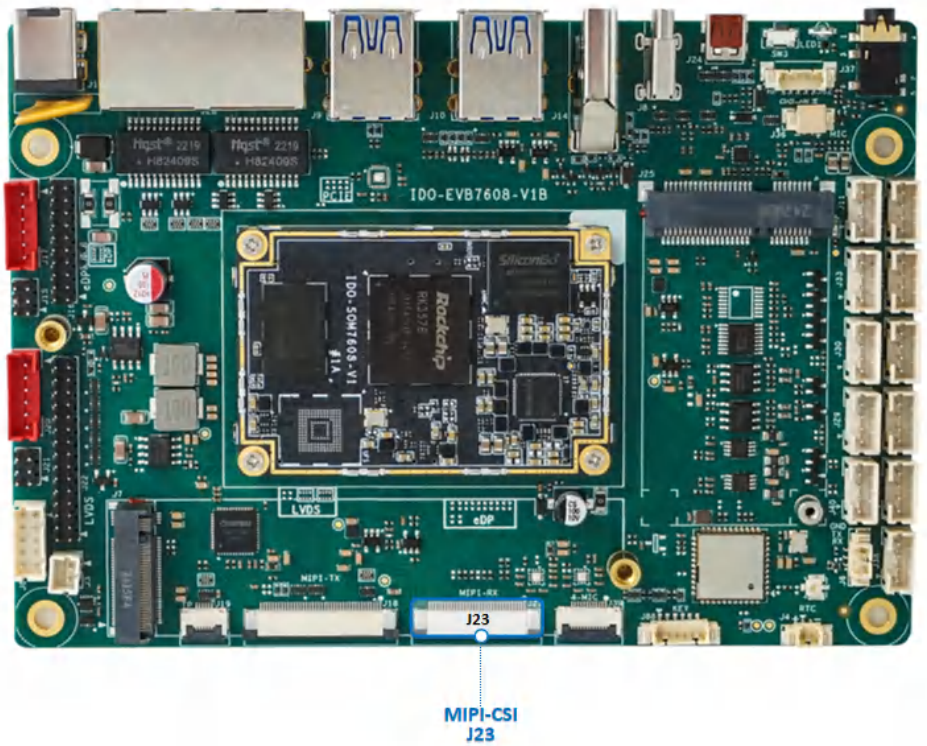
TP接口为(J19) 6Pin FPC 0.5mm座子，如下图所示：



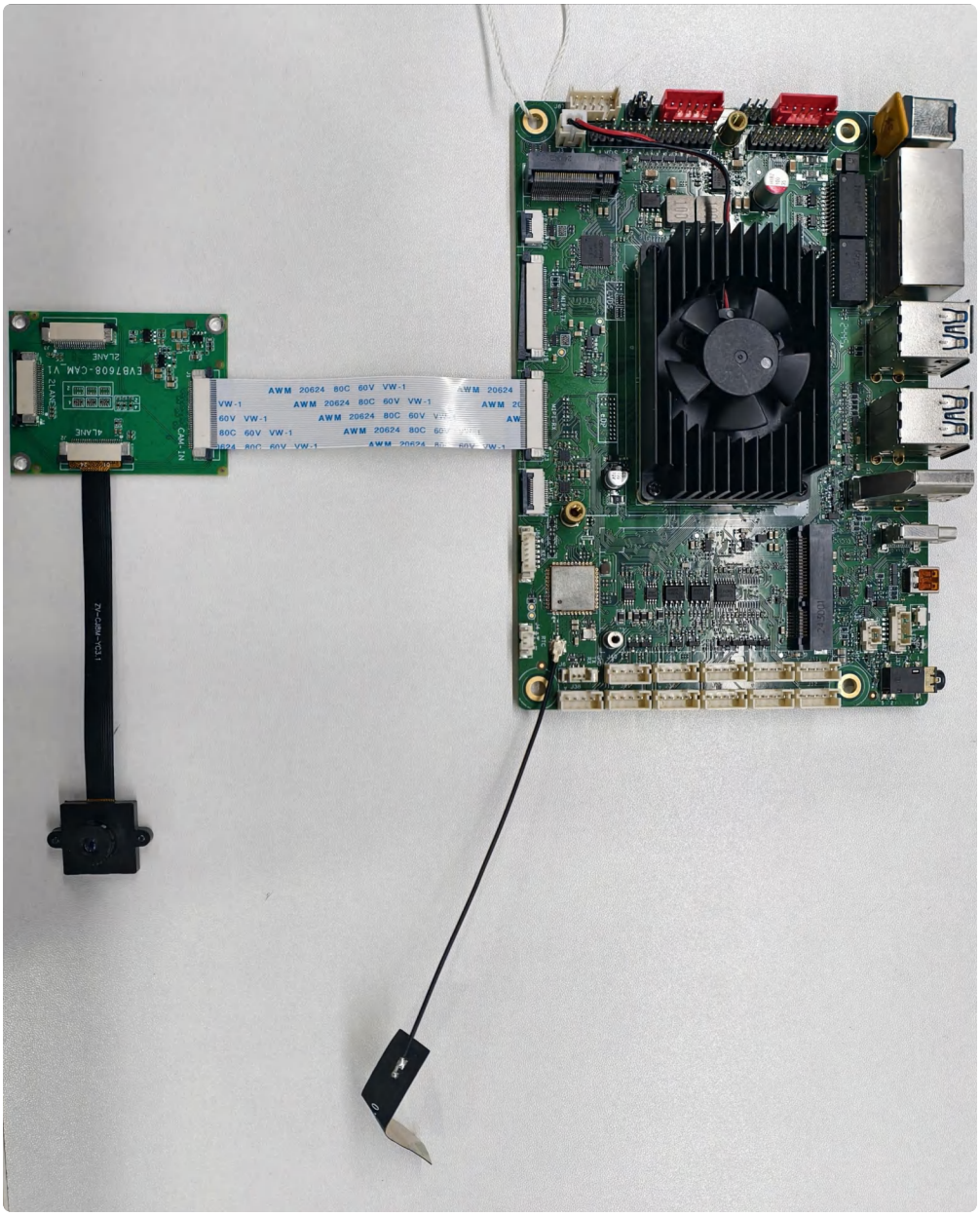
- 触摸 TP 接口接线方法为下接

3.12. MIPI Camera

MIPI Camer接口J23，如下图所示：



使用FPC05-PUW-32P抽拉式上接座子，默认适配IMX415摄像头（可接OV13855），接法如下图所示：



3.12.1. 预览命令：

```
1  root@rk3576-buildroot:/# export DISPLAY=:0
2  root@rk3576-buildroot:/# gst-launch-1.0 v4l2src device=/dev/video22 ! video/x-raw, format=NV12, width=1920, height=1080, framerate=30/1 ! videoconvert ! autovideosink
```

3.12.2. 抓图：

```
root@rk3576-buildroot:/# v4l2-ctl --verbose -d /dev/video22 --set-fmt-video=width=1920,height=1080,pixelformat='NV12' --stream-mmap=4 --stream-skip=3 --stream-to=./test-1920x1080-p50.yuv --stream-count=1 --stream-poll
```

查看照片：

```
root@rk3576-buildroot:/# ffplay -f rawvideo -video_size 1920x1080 -pixel_format nv12 ./test-1920x1080-p50.yuv
```

3.12.3. 抓视频：

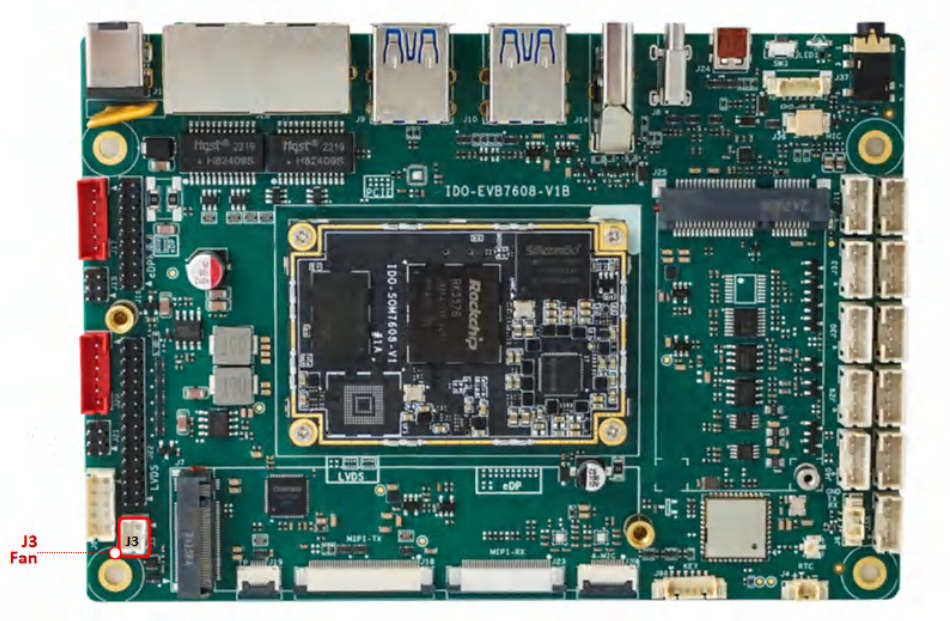
```
root@rk3576-buildroot:/# v4l2-ctl --stream-mmap=4  -d /dev/video22 --set-fmt-video=width=1920,height=1080,pixelformat='NV12' --stream-to=./output.yuv
```

播放视频：

```
root@rk3576-buildroot:/# ffplay -f rawvideo -video_size 1920x1080 -pixel_format nv12 ./output.yuv
```

3.13. FAN 风扇

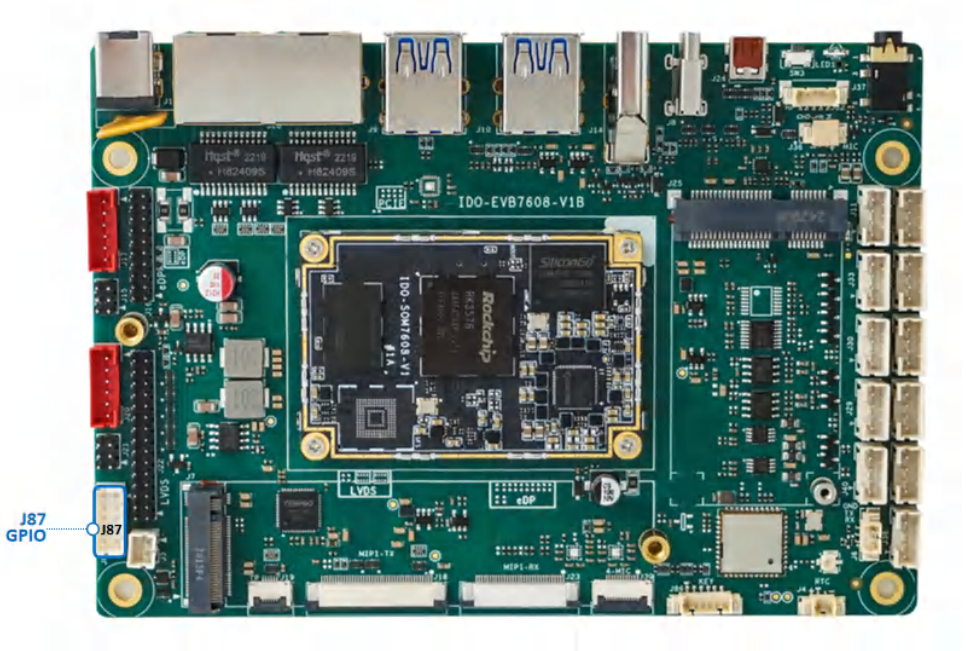
PH2 5V FAN风扇接口J3，如下图所示：



控制功能	控制命令
打开风扇	echo 1 > /sys/class/leds/fan/brightness
关闭风扇	echo 0 > /sys/class/leds/fan/brightness

3.14. GPIO

主板预留了8个GPIO接口J87，如下图所示：



GPIO引脚定义描述，如下图所示：

序号	管脚标号	GPIO 标号	GPIO 序号	LIBGPIOD节点
1	CSN	GPIO3_D0	120	gpiochip3 24
2	VCC	5V/3.3V	/	/
3	IO00	IO0_0	493	gpiochip6 0
4	MISO	GPIO3_C5	117	gpiochip3 21
5	IO01	IO0_1	494	gpiochip6 1
6	MOSI	GPIO3_C6	118	gpiochip3 22
7	IO12	IO1_2	503	gpiochip6 10
8	CLK	GPIO3_C7	119	gpiochip3 23
9	IO15	IO1_5	506	gpiochip6 13
10	GND	地	/	/

GPIO控制，根据上表【LIBGPIOD节点】进行命令操作

libgpiod C

```
# 设置 GPIO3_D0 输出高电平状态
gpioset gpiochip3 24=1

# 设置 GPIO3_D0 输出低电平状态
gpioset gpiochip3 24=0

# 设置 IO00 输出高电平状态
gpioset gpiochip6 0=1

# 设置 IO00 输出低电平状态
gpioset gpiochip6 0=0

# 获取 GPIO3_D0 输入电平状态
gpioget gpiochip3 24

# 获取 IO00 输入电平状态
gpioget gpiochip6 0
```

其中1、4、6、8脚可配置为SPI接口，对应设备树SPI1_M2

序号	管脚标号	GPIO 标号	GPIO 序号

1	CSN	GPIO1_A3	120
4	MISO	GPIO3_C5	117
6	MOSI	GPIO3_C6	118
8	CLK	GPIO3_C7	119

3.15. GPU

执行glmark2-es2-wayland 开启gpu测试窗口：

```
root@rk3576-buildroot:/# glmark2-es2-wayland
arm_release_ver: g24p0-00eac0, rk_so_ver: 7
=======================================================
    glmark2 2023.01
=======================================================
    OpenGL Information
    GL_VENDOR:      ARM
    GL_RENDERER:    Mali-G52
    GL_VERSION:     OpenGL ES 3.2 v1.g24p0-00eac0.7624deb1338aa462bb011099003f89a4
    Surface Config: buf=32 r=8 g=8 b=8 a=8 depth=24 stencil=0 samples=0
    Surface Size:   800x600 windowed
=======================================================
[build] use-vbo=false: FPS: 1473 FrameTime: 0.679 ms
[build] use-vbo=true: FPS: 1399 FrameTime: 0.715 ms
[texture] texture-filter=nearest: FPS: 2044 FrameTime: 0.489 ms
[texture] texture-filter=linear: FPS: 2110 FrameTime: 0.474 ms
[texture] texture-filter=mipmap: FPS: 2101 FrameTime: 0.476 ms
[shading] shading=gouraud: FPS: 1191 FrameTime: 0.840 ms
[shading] shading=blinn-phong-inf: FPS: 1321 FrameTime: 0.757 ms
[shading] shading=phong: FPS: 1248 FrameTime: 0.802 ms
[shading] shading=cel: FPS: 1191 FrameTime: 0.840 ms
[bump] bump-render=high-poly: FPS: 652 FrameTime: 1.535 ms
[bump] bump-render=normals: FPS: 1924 FrameTime: 0.520 ms
[bump] bump-render=height: FPS: 1531 FrameTime: 0.653 ms
[effect2d] kernel=0,1,0;1,-4,1;0,1,0;: FPS: 1111 FrameTime: 0.901 ms
[effect2d] kernel=1,1,1,1,1;1,1,1,1,1;1,1,1,1,1;: FPS: 518 FrameTime: 1.931 ms
[pulsar] light=false:quads=5:texture=false: FPS: 1347 FrameTime: 0.743 ms
[desktop] blur-radius=5:effect=blur:passes=1:separable=true:windows=4: FPS: 723 FrameTime: 1.383 ms
[desktop] effect=shadow:windows=4: FPS: 1458 FrameTime: 0.686 ms
[buffer] columns=200:interleave=false:update-dispersion=0.9:update-fraction=0.5:update-method=map: FPS: 225 FrameTime: 4.453 ms
[buffer] columns=200:interleave=false:update-dispersion=0.9:update-fraction=0.5:update-method=subdata: FPS: 203 FrameTime: 4.933 ms
[buffer] columns=200:interleave=true:update-dispersion=0.9:update-fraction=0.5:update-method=map: FPS: 317 FrameTime: 3.157 ms
[ideas] speed=duration: FPS: 782 FrameTime: 1.280 ms
[jellyfish] <default>: FPS: 978 FrameTime: 1.023 ms
[terrain] <default>: FPS: 94 FrameTime: 10.683 ms
[shadow] <default>: FPS: 647 FrameTime: 1.547 ms
[refract] <default>: FPS: 181 FrameTime: 5.548 ms

```

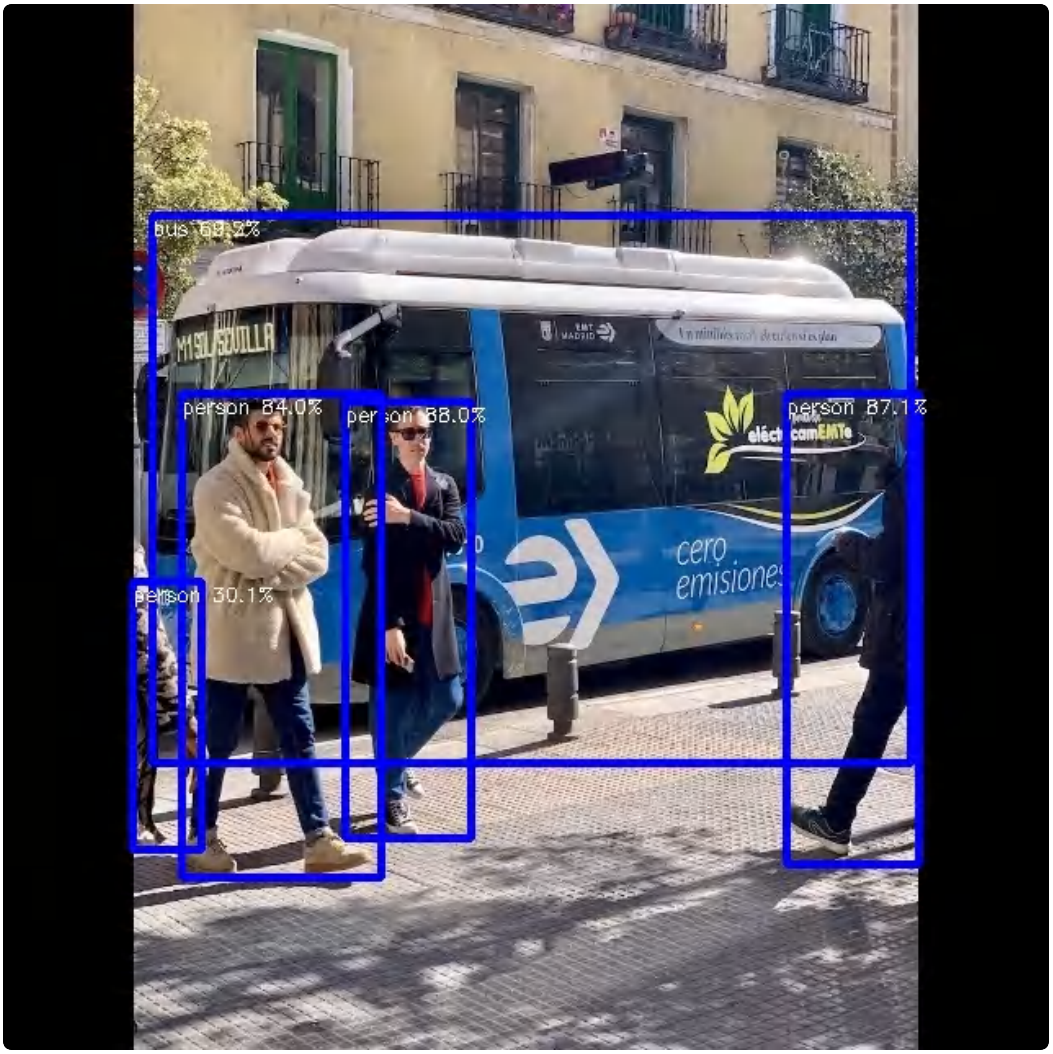
```
[conditionals] fragment-steps=0:vertex-steps=0: FPS: 1276 FrameTime: 0.784 ms
[conditionals] fragment-steps=5:vertex-steps=0: FPS: 1156 FrameTime: 0.865 ms
[conditionals] fragment-steps=0:vertex-steps=5: FPS: 1239 FrameTime: 0.808 ms
[function] fragment-complexity=low:fragment-steps=5: FPS: 1218 FrameTime: 0.821 ms
[function] fragment-complexity=medium:fragment-steps=5: FPS: 1153 FrameTime: 0.867 ms
[loop] fragment-loop=false:fragment-steps=5:vertex-steps=5: FPS: 1369 FrameTime: 0.731 ms
[loop] fragment-steps=5:fragment-uniform=false:vertex-steps=5: FPS: 1344 FrameTime: 0.744 ms
[loop] fragment-steps=5:fragment-uniform=true:vertex-steps=5: FPS: 1333 FrameTime: 0.751 ms
=======================================================
                                  glmark2 Score: 1115
```

3.16. NPU

使用yolov5进行模型转换测试：

```shell
root@rk3576-buildroot:/# tar -xvf rknn_yolov5_demo_Linux.tar.gz
root@rk3576-buildroot:/# cd rknn_yolov5_demo_Linux
root@rk3576-buildroot:/# chmod a+x rknn_yolov5_demo
root@rk3576-buildroot:/# ./rknn_yolov5_demo ./model/RK3576/yolov5s-640-640.rknn ./model/bus.jpg
post process config: box_conf_threshold = 0.25, nms_threshold = 0.45
Loading mode...
sdk version: 2.3.2 (429f97ae6b@2025-04-09T09:09:27) driver version: 0.9.8
model input num: 1, output num: 3
  index=0, name=images, n_dims=4, dims=[1, 640, 640, 3], n_elems=1228800, size=1228800, w_stride = 640, size_with_stride=1228800, fmt=NHWC, type=INT8, qnt_type=AFFINE, zp=-128, scale=0.003922
  index=0, name=output0, n_dims=4, dims=[1, 255, 80, 80], n_elems=1632000, size=1632000, w_stride = 0, size_with_stride=1638400, fmt=NCHW, type=INT8, qnt_type=AFFINE, zp=-128, scale=0.003922
  index=1, name=286, n_dims=4, dims=[1, 255, 40, 40], n_elems=408000, size=408000, w_stride = 0, size_with_stride=491520, fmt=NCHW, type=INT8, qnt_type=AFFINE, zp=-128, scale=0.003922
  index=2, name=288, n_dims=4, dims=[1, 255, 20, 20], n_elems=102000, size=102000, w_stride = 0, size_with_stride=163840, fmt=NCHW, type=INT8, qnt_type=AFFINE, zp=-128, scale=0.003922
model is NHWC input fmt
model input height=640, width=640, channel=3
Read ./model/bus.jpg ...
img width = 640, img height = 640
once run use 37.585000 ms
loadLabelName ./model/coco_80_labels_list.txt
person @ (209 243 286 510) 0.879723
person @ (479 238 560 526) 0.870588
person @ (109 237 232 534) 0.828112
bus @ (93 129 553 464) 0.700761
person @ (79 353 122 517) 0.307297
save detect result to ./out.jpg
loop count = 10 , average run  33.047100 ms
```

执行完后会得到转换后的模型图片，如下所示：



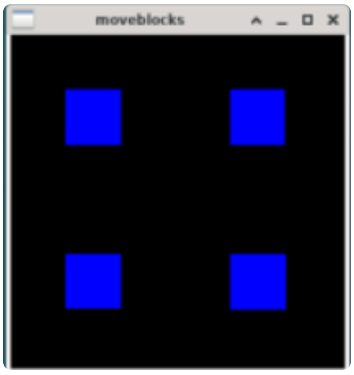
3.17. QT5

支持qt5.15，测试方法如下：

Shell

```
root@rk3576-buildroot:/# chmod a+x ./moveblocks
root@rk3576-buildroot:/# ./moveblocks
```

结果如下所示：

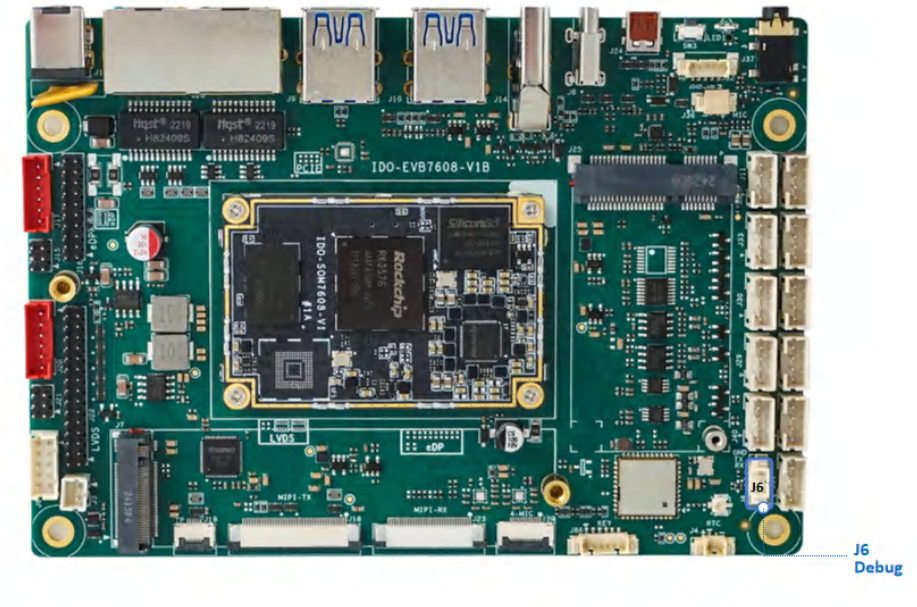


IDO-EVB7608-V1B-Ubuntu系统

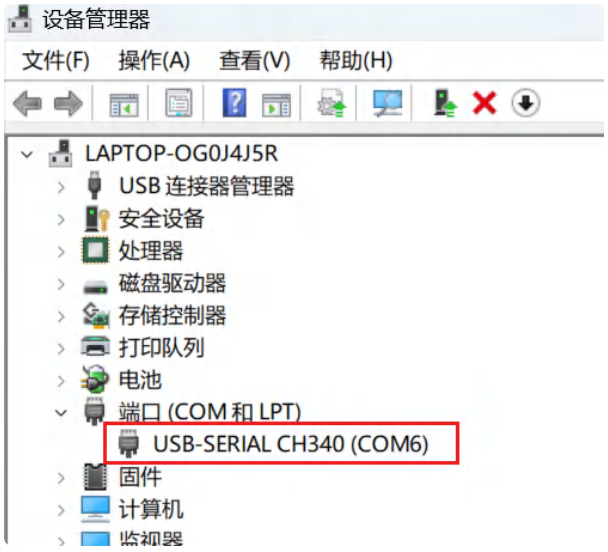
4. 功能及接口使用方法

4.1. DEBUG UART

DEBUG UART位置J6，如下图所示：

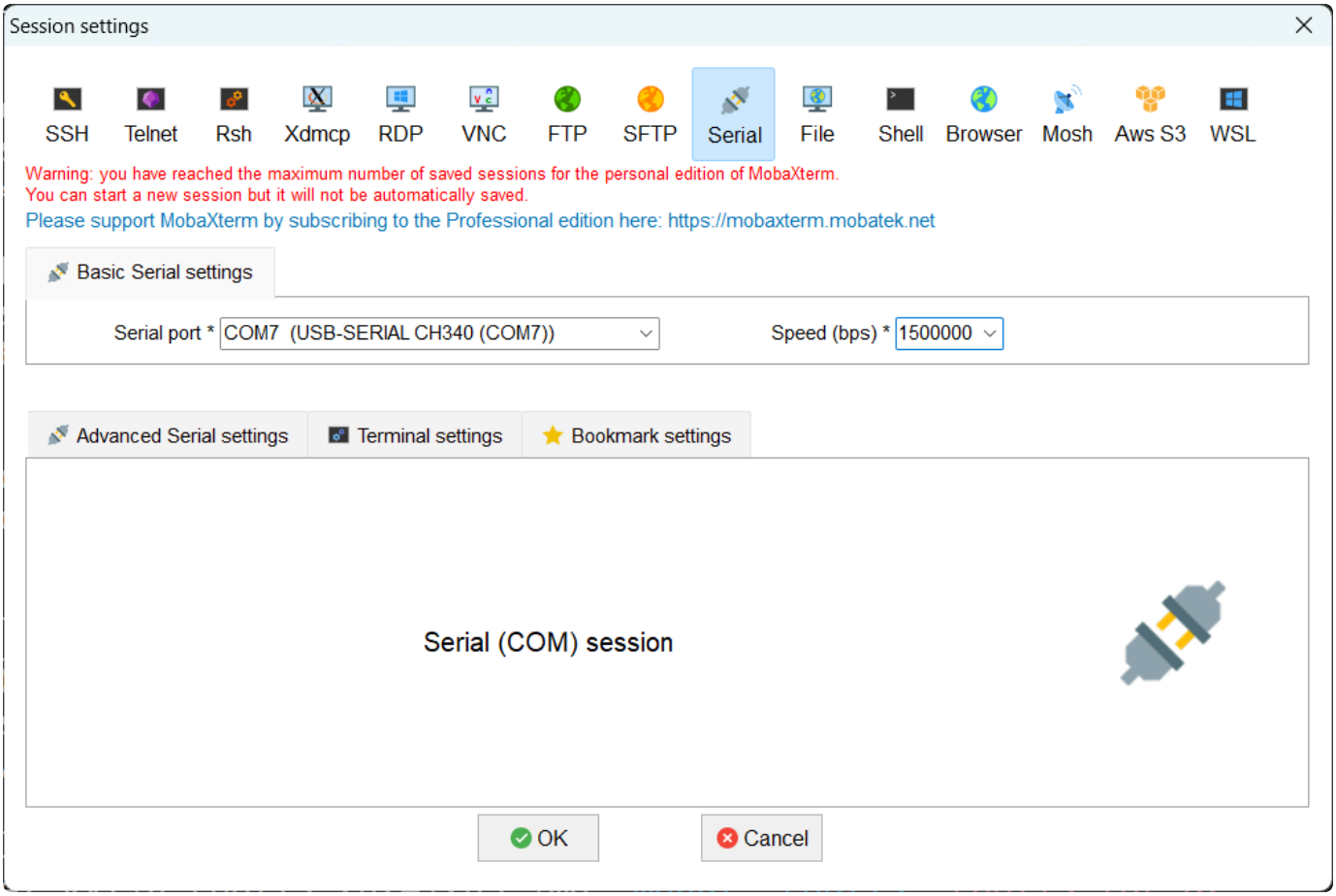


使用USB Type-C数据线，连接PC端的USB接口。系统会识别到一个“USB-SERIAL CH340”端口设备。



串口登录账号密码：industio/123456

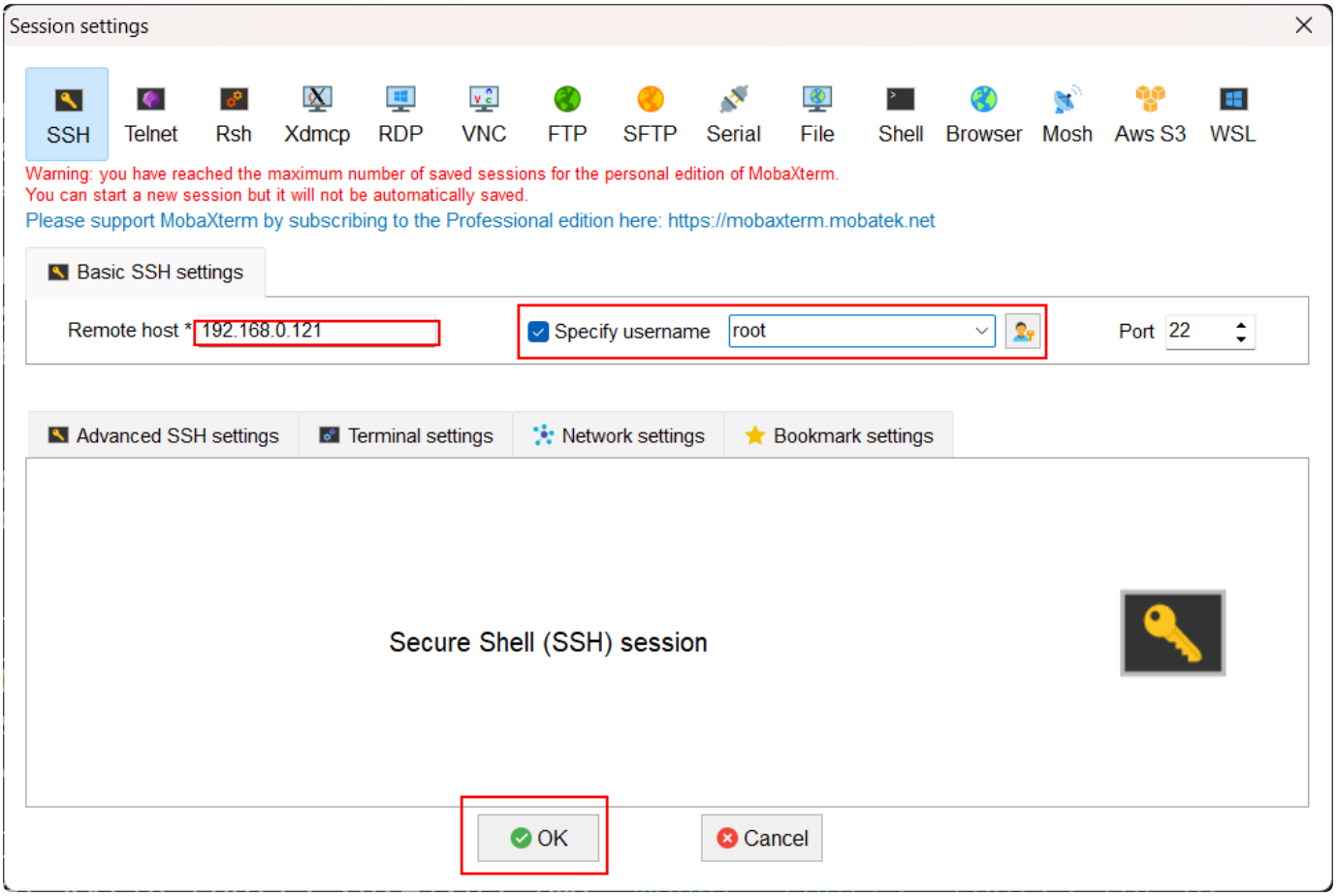
使用调试软件（MobaXterm、putty）等，以MobaXterm为例子，设置参数如下：



SSH登录

在知道IP的前提下，默认可以通过SSH登录进系统。

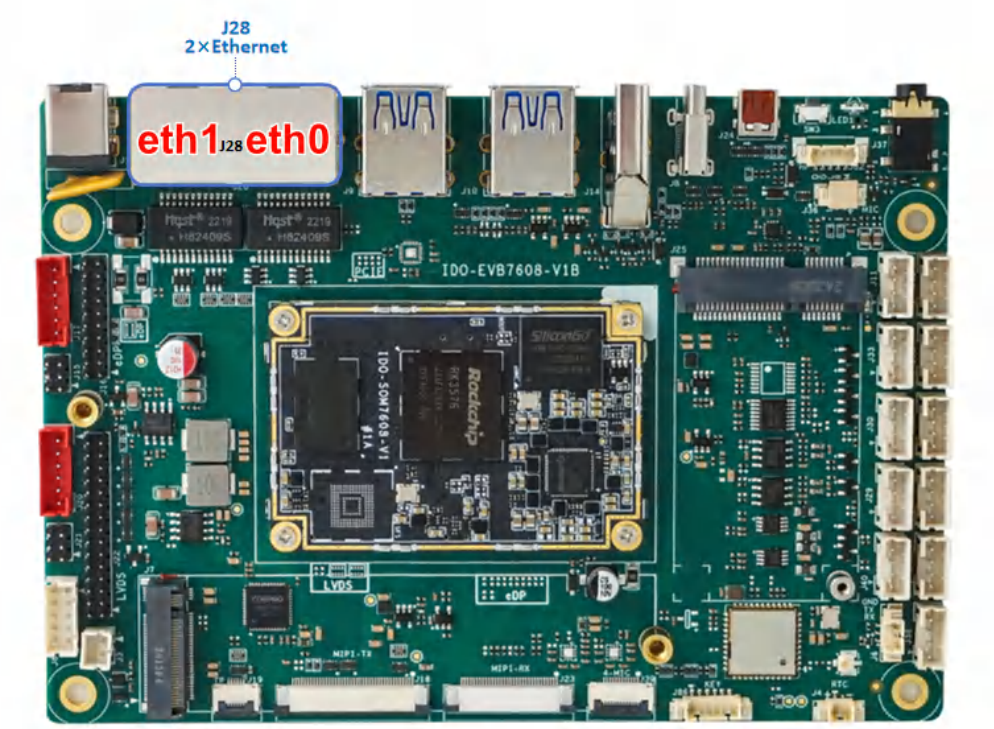
SSH登录账号密码：industio/123456



4.2. 网络

4.2.1. 以太网

主板有两路千兆以太网接口位置J28，如下图所示：



设备节点分别为eth0和eth1，以太网接口默认支持DHCP，只需要将以太网接口连接路由器即可为主板动态分配 IP 地址。网络正常连接图标，如下图所示：

Ethernet动态IP设置：

PowerShell

```
#eth0
industio@industio:~$ ifconfig eth0
eth0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether 2a:6b:1b:4f:e7:bf  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
        device interrupt 67


#eth1
industio@industio:~$ ifconfig eth1
eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.116  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::4281:69a2:6758:f125  prefixlen 64  scopeid 0x20<link>
        ether 2e:6b:1b:4f:e7:bf  txqueuelen 1000  (Ethernet)
        RX packets 643  bytes 117640 (117.6 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 419  bytes 49002 (49.0 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 69
```

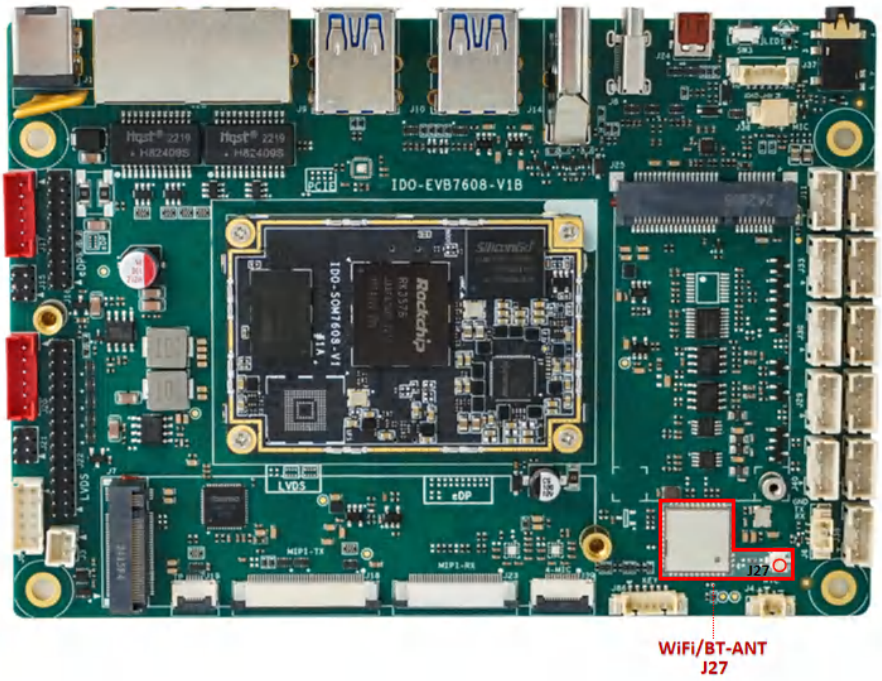
Ethernet静态IP设置：

```PowerShell
root@industio:~# touch /etc/netplan/01-netcfg.yaml
root@industio:~# vim /etc/netplan/01-netcfg.yaml
root@industio:~# cat  /etc/netplan/01-netcfg.yaml
network:
        version: 2
        ethernets:
                eth0:
                        dhcp4: no
                        addresses: [192.168.3.121/24]
                        nameservers:
                                addresses: [192.168.3.1,8.8.8.8, 114.114.114.114]
                        routes:
                                - to: 0.0.0.0/0
                                  via: 192.168.3.1
                                  metric: 100
#立即生效
root@industio:~# netplan apply
Cannot call openvswitch: ovsdb-server.service is not running.

industio@industio:~$ ifconfig eth0
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.3.121  netmask 255.255.255.0  broadcast 192.168.3.255
        inet6 fe80::1b83:a156:cb16:c6b0  prefixlen 64  scopeid 0x20<link>
        inet6 fe80::f8c6:baff:fec7:c60  prefixlen 64  scopeid 0x20<link>
        ether 2a:6b:1b:4f:e7:bf  txqueuelen 1000  (Ethernet)
        RX packets 4203  bytes 417790 (407.9 KiB)
        RX errors 0  dropped 164  overruns 0  frame 0
        TX packets 392  bytes 32299 (31.5 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 121
```

设备断电重启，此静态IP设置仍然生效。

4.2.2. WiFi

使用WiFi/蓝牙时，需要连接天线以获得良好的信号，WiFi模块和天线座子J27，如下图所示：



连接WiFi热点

命令行可以使用nmcli工具连接wifi热点：

Shell

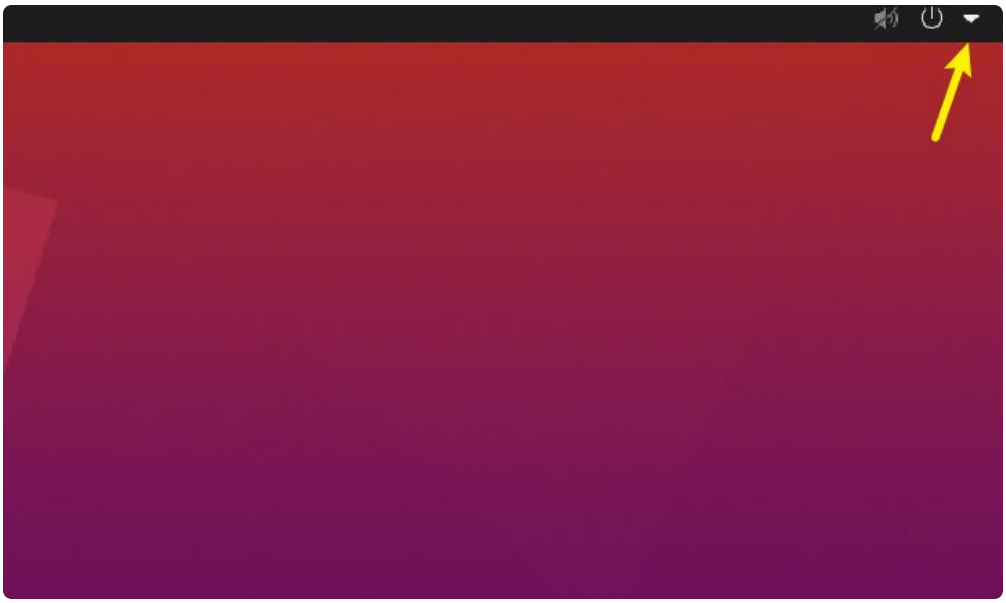
```
#查看网络接口状态:
root@industio:~# nmcli device
#使用以下命令查看当前可用的 WiFi 网络:
root@industio:~# nmcli device wifi list
#连接到 WiFi 网络:
root@industio:~# nmcli device wifi connect 账号 password 密码
#确认连接状态:
root@industio:~# nmcli connection show --active
```

查看wlan0的IP地址，确认连接成功：

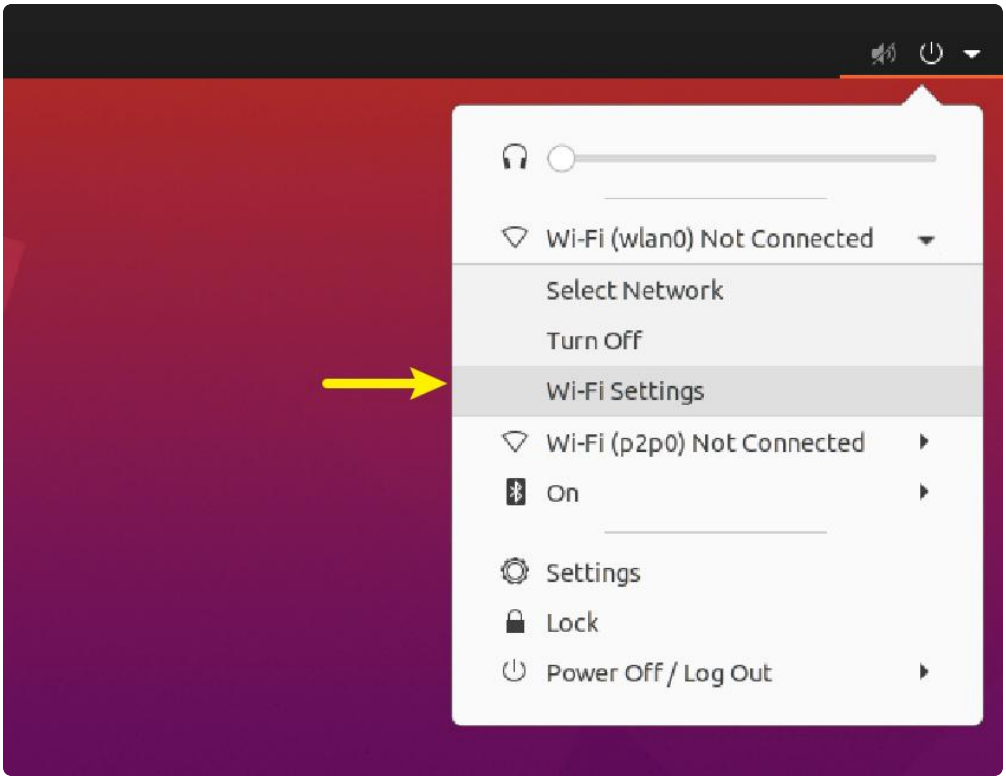
```shell
root@industio:~# ifconfig wlan0
wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.120  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::e6ce:be28:4c94:577e  prefixlen 64  scopeid 0x20<link>
        ether c0:f5:35:12:e6:34  txqueuelen 1000  (Ethernet)
        RX packets 18  bytes 2238 (2.2 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 52  bytes 7844 (7.8 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@industio:~# ping www.baidu.com
PING www.a.shifen.com (183.2.172.17): 56 data bytes
64 bytes from 183.2.172.17: icmp_seq=0 ttl=52 time=8.007 ms
64 bytes from 183.2.172.17: icmp_seq=1 ttl=52 time=8.049 ms
64 bytes from 183.2.172.17: icmp_seq=2 ttl=52 time=7.901 ms
```

也可以在桌面上连接：

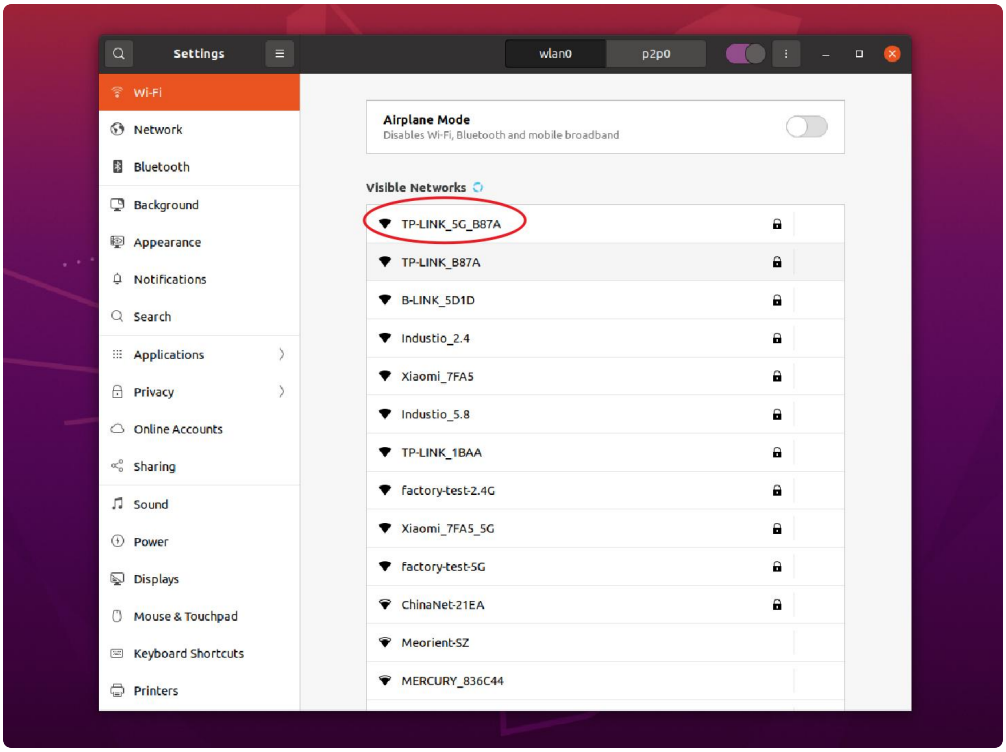
点击桌面右上角的【下拉】选项按钮，如下图所示：



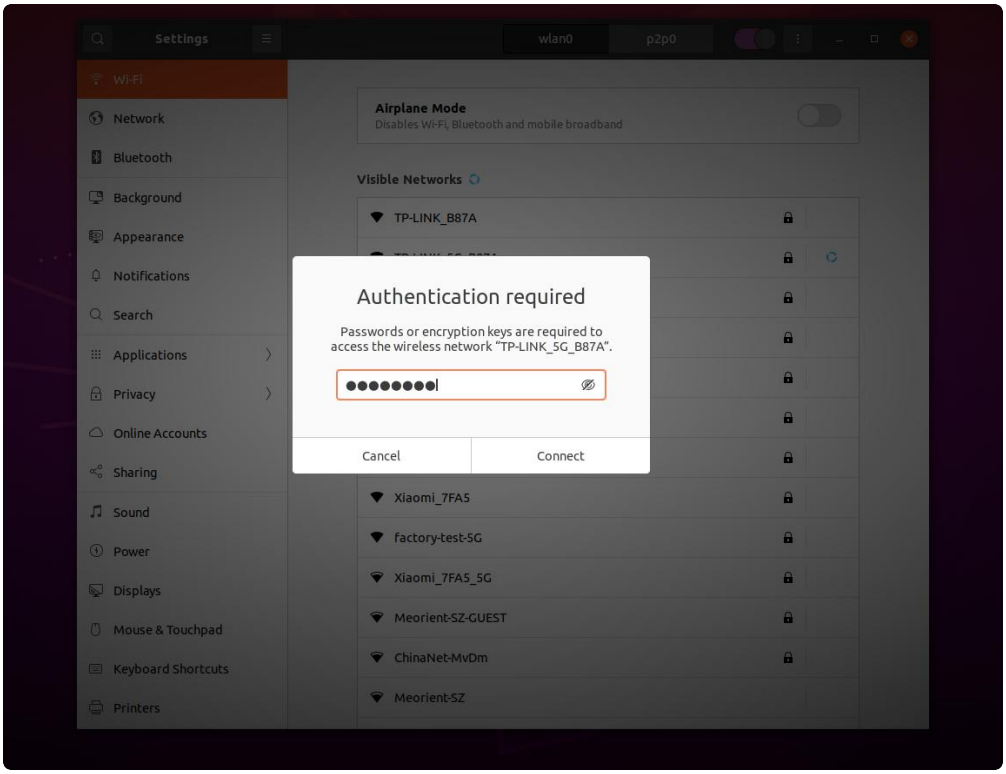
弹出的列表中点击【Wi-Fi Settings】，如下图所示：



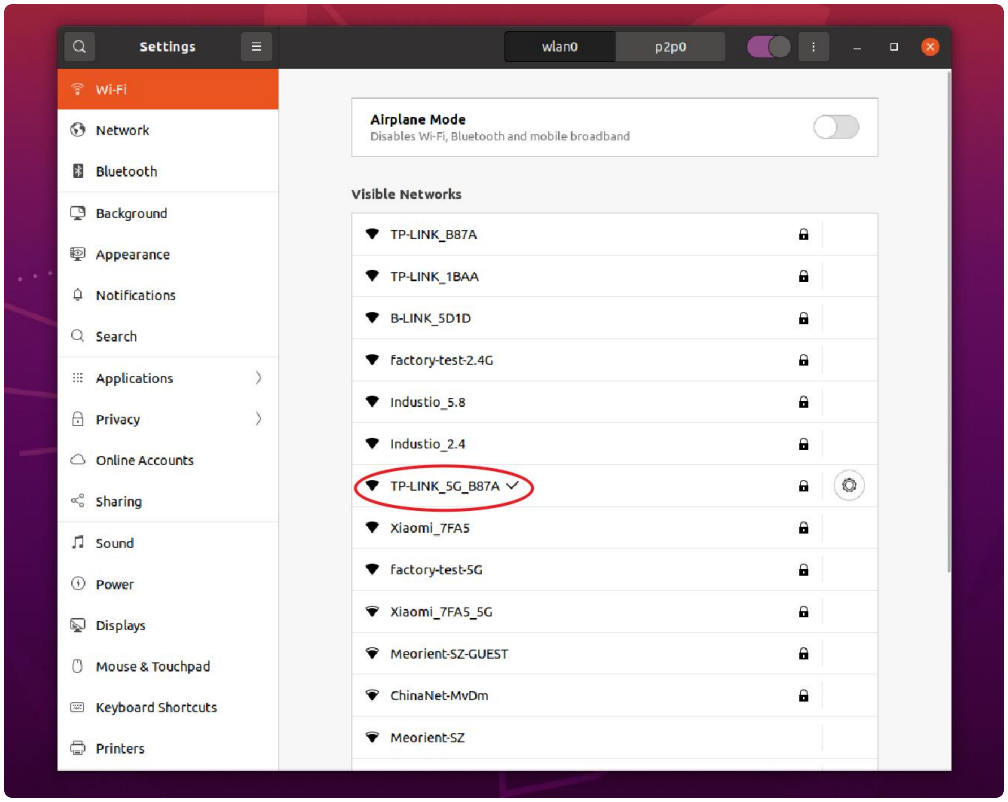
弹出WiFi热点列表，点击要连接的热点名称，如下图所示：



弹出密码输入框，使用键盘输入密码，如下图所示：



如果热点名称后面有"√"标记，表示连接成功，如下图所示：



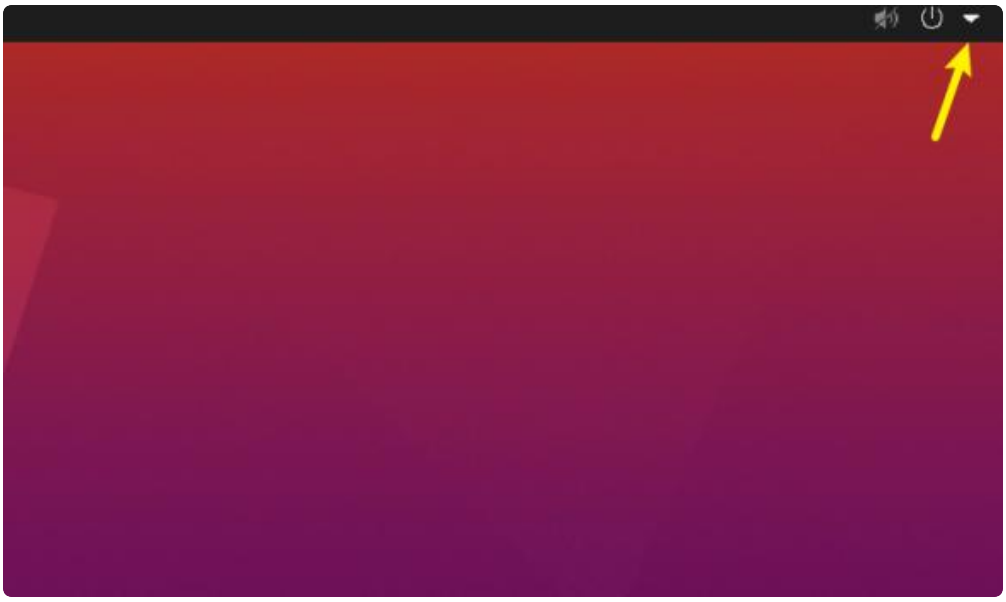
通过ifconfig 命令查看wlan0的IP地址确认，命令如下：

```shell
root@industio:~# ifconfig wlan0
wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.120  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::e6ce:be28:4c94:577e  prefixlen 64  scopeid 0x20<link>
        ether c0:f5:35:12:e6:34  txqueuelen 1000  (Ethernet)
        RX packets 141  bytes 40786 (40.7 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 114  bytes 30656 (30.6 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

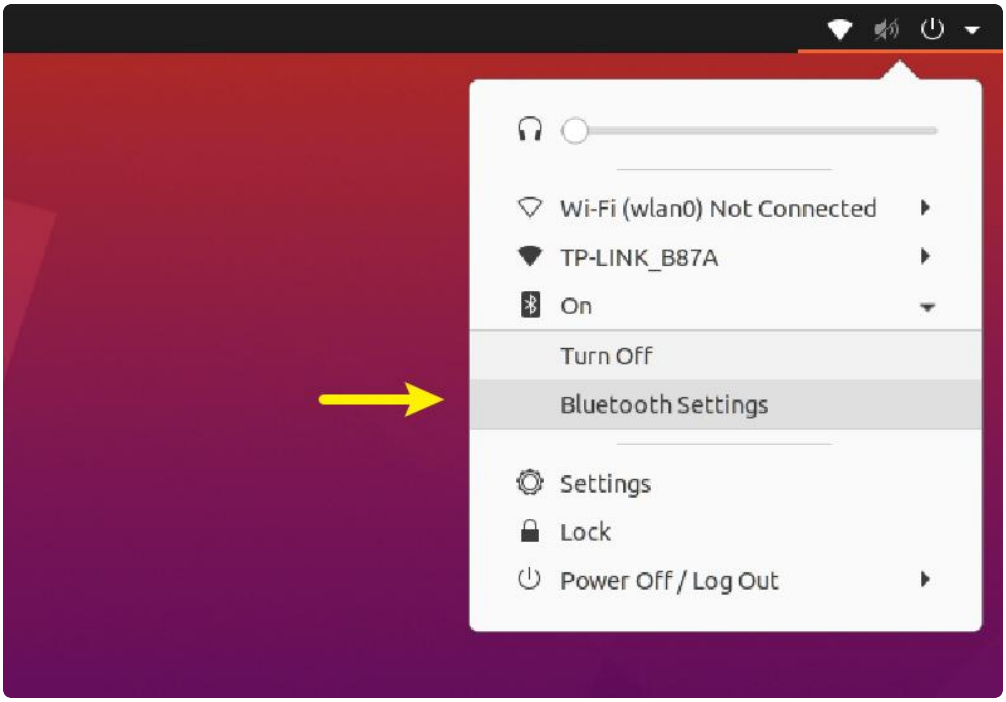
4.2.3. Bluetooth

桌面连接：

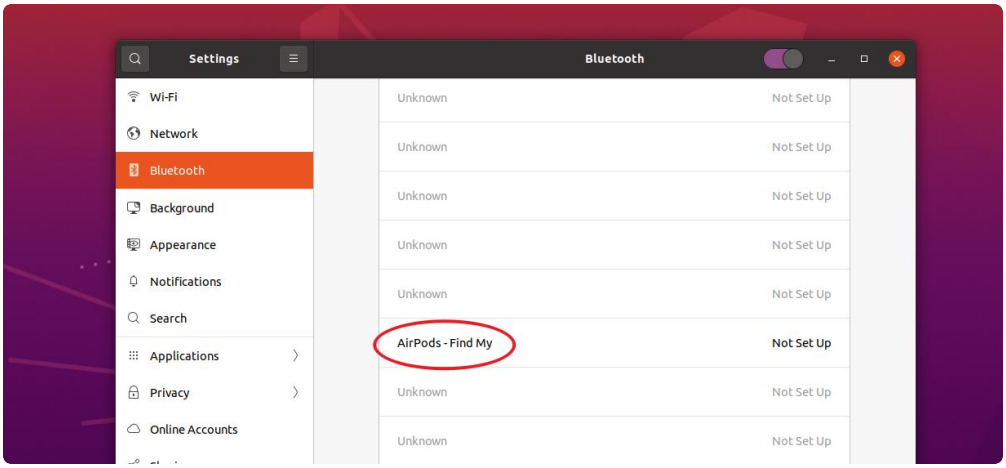
点击桌面右上角的【下拉选项】，如下图所示：



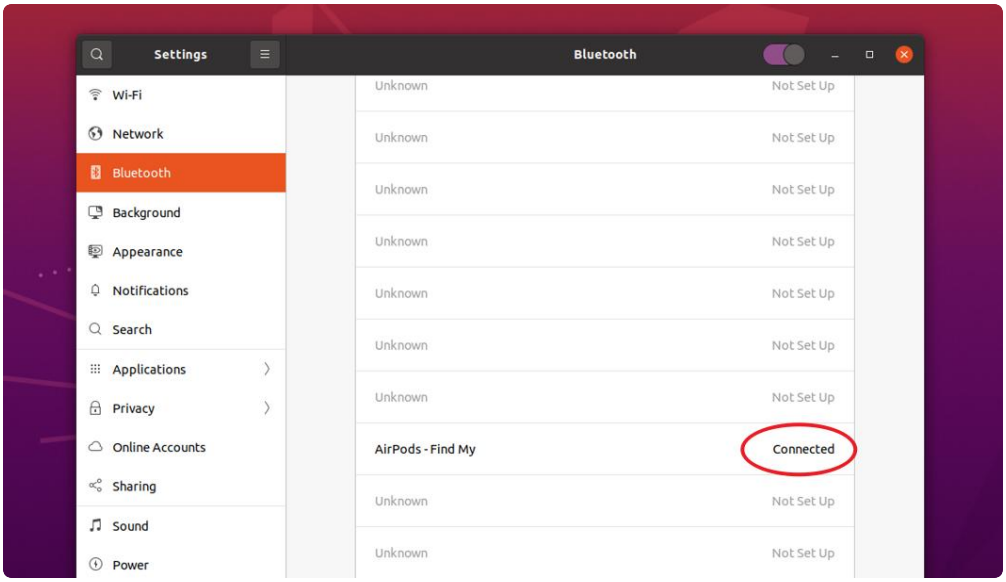
弹出的列表中点击【蓝牙】，继续点击【Bluetooth Setting】，如下图所示：



弹出蓝牙扫描列表，点击要连接的蓝牙设备名称，连接蓝牙设备，如下图所示：



设备名称后面提示"Connected"，表示该设备已连接成功，如下图所示：

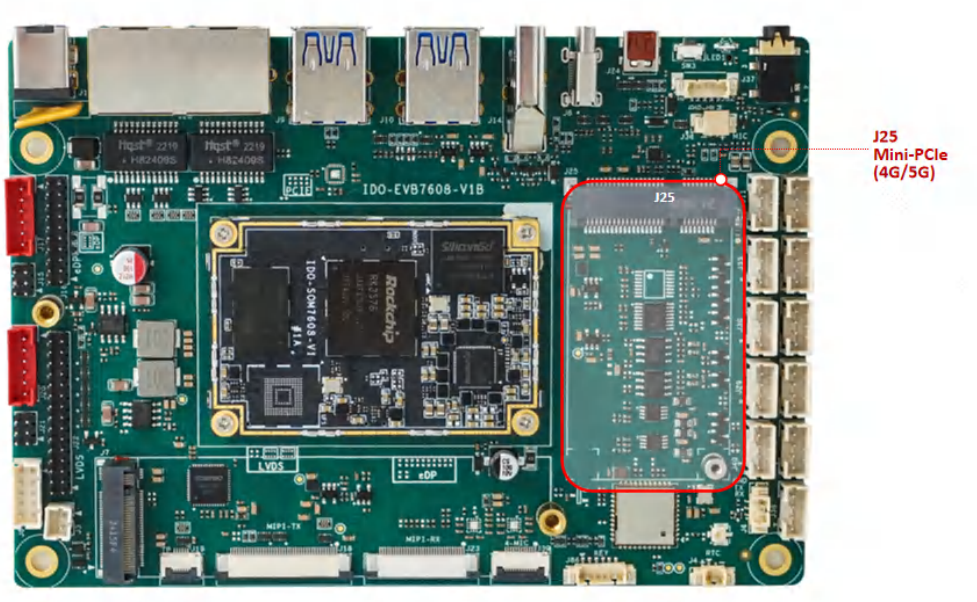


使用命令行操作

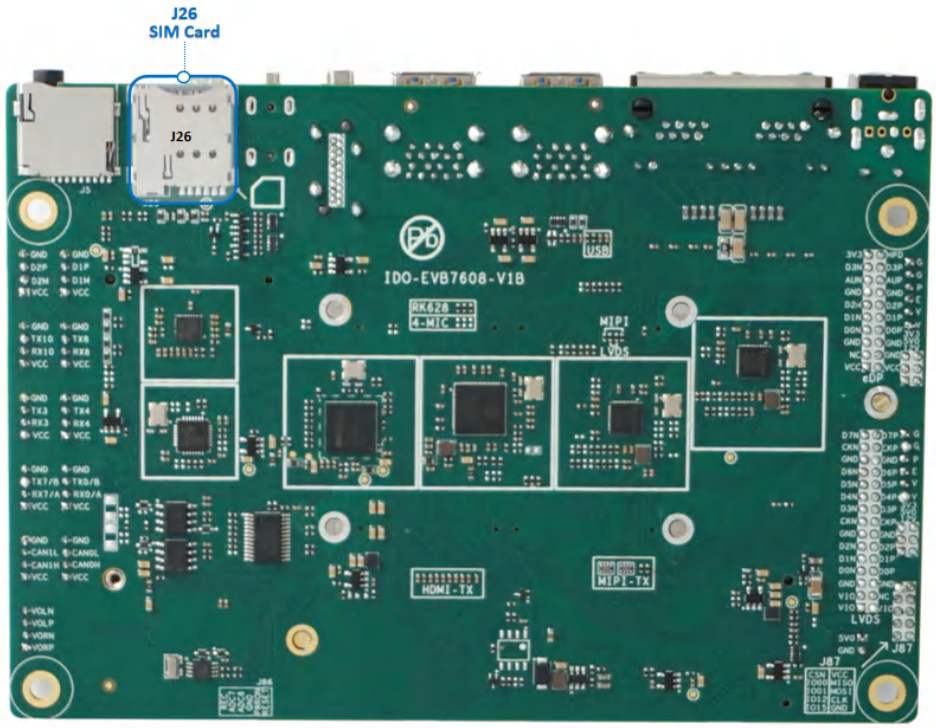
```shell
#打开蓝牙
root@industio:~# bluetoothctl power on
#扫描蓝牙
root@industio:~# bluetoothctl scan on
#查看蓝牙设备
root@industio:~# bluetoothctl devices
#信任蓝牙设备
root@industio:~# bluetoothctl trust 7C:C1:80:09:DD:6C
#蓝牙配对
root@industio:~# bluetoothctl pair 7C:C1:80:09:DD:6C
#连接蓝牙
root@industio:~# bluetoothctl connect 7C:C1:80:09:DD:6C
```

4.2.4. 4G/5G

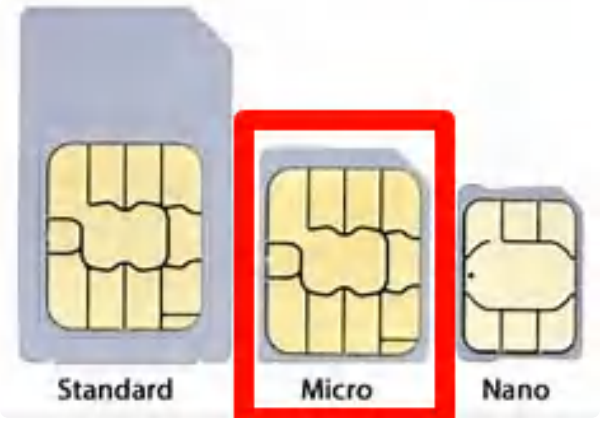
主板内置Mini PCIe 座子扩展 4G/5G模块，4G通信模块适配移远EC20、EC25等通用模组。5G通信模块适配移远RG200U-CN。模块安装位J25，如下图所示：



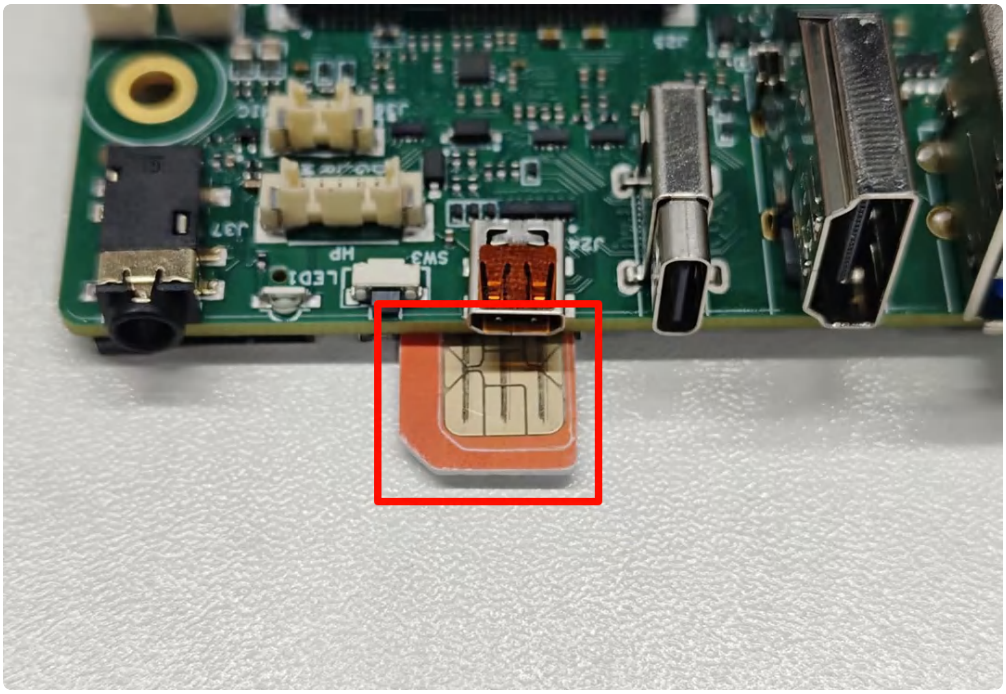
模块需要接上天线以确保获得更好的信号质量，SIM卡安装位J26位于主板背面，如下图所示：



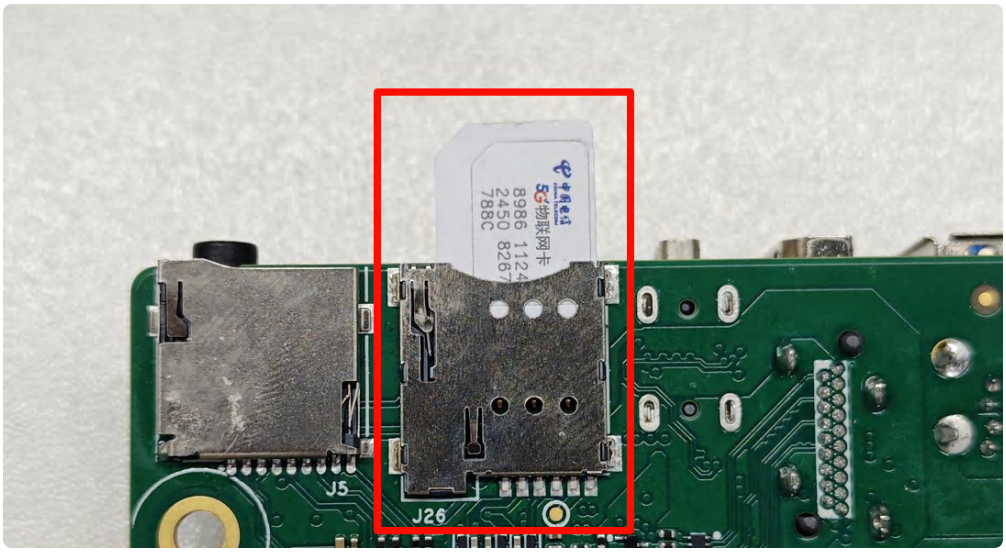
使用Micro尺寸SIM卡，如下图所示：



主板朝上时，SIM卡触点朝上，缺口朝外安装，如下图所示：



主板朝下时，SIM卡触点朝下缺口朝外安装，如下图所示：



4G使用 `ifconfig wwan0` 命令可查看到相关网络信息：

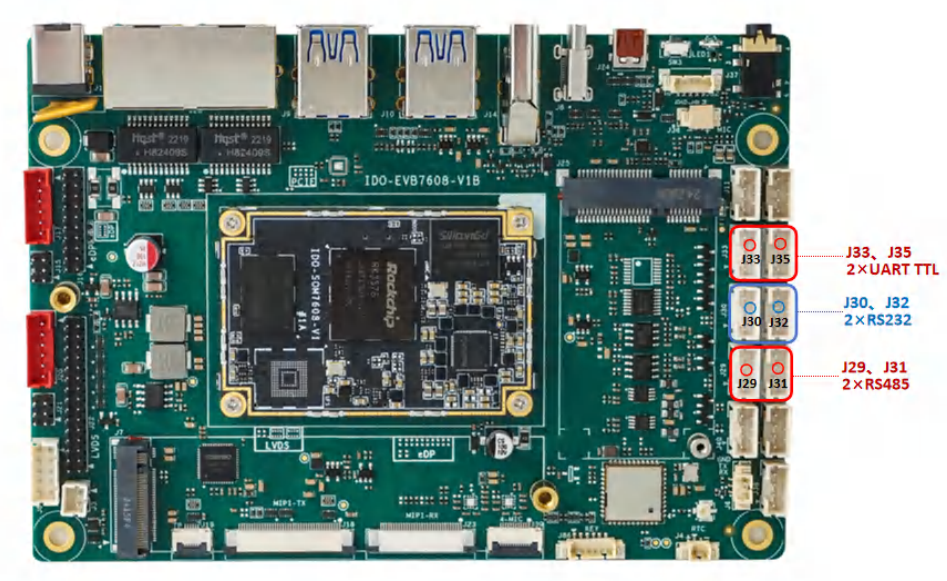
```
#拨号
root@industio:~# quectel-CM &
#查看网络
root@industio:~# ifconfig wwan0
wwan0: flags=193<UP,RUNNING,NOARP>  mtu 1500
        inet 10.101.61.51  netmask 255.255.255.248
        inet6 fe80::fc:f6ff:fe8d:bab6  prefixlen 64  scopeid 0x20<link>
        ether 02:fc:f6:8d:ba:b6  txqueuelen 1000  (Ethernet)
        RX packets 42  bytes 7013 (6.8 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 57  bytes 4608 (4.5 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

#测试通信
root@industio:~# ping www.baidu.com -I wwan0
```

5G使用 `ifconfig usb0` 命令可查看到相关网络信息:

```
#拨号
root@industio:~# quectel-CM &
#查看网络
root@industio:~# ifconfig usb0
usb0: flags=193<UP,RUNNING,NOARP>  mtu 1500
        inet 10.101.61.51  netmask 255.255.255.248
        inet6 fe80::fc:f6ff:fe8d:bab6  prefixlen 64  scopeid 0x20<link>
        ether 02:fc:f6:8d:ba:b6  txqueuelen 1000  (Ethernet)
        RX packets 42  bytes 7013 (6.8 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 57  bytes 4608 (4.5 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

#测试通信
root@industio:~# ping www.baidu.com -I usb0
```

4.3. UART

IDO-EVB7608-V1主板一共引出6路UART（不含DEBUG UART），如下图所示：



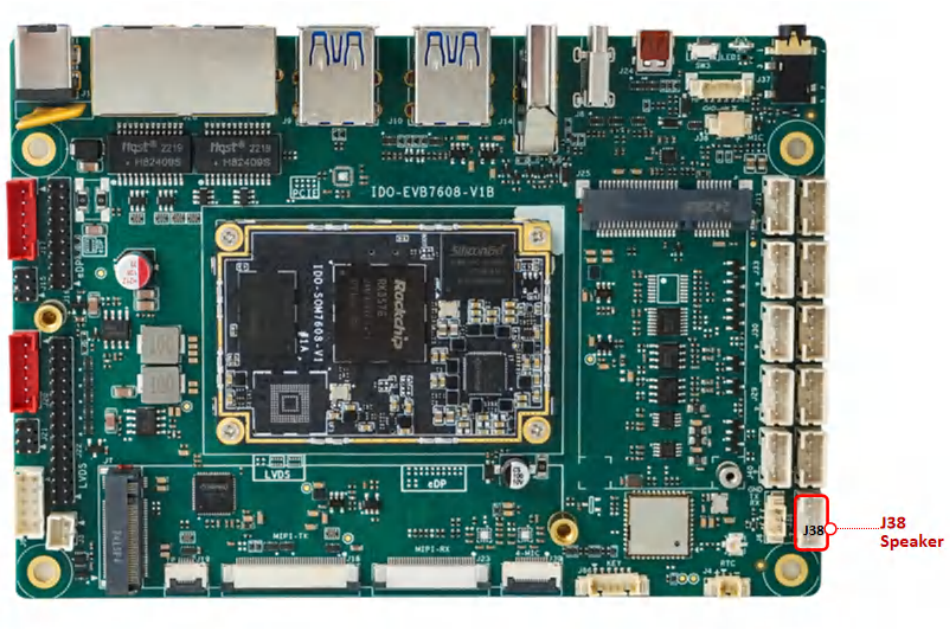
UART设备节点列表如下：

序号	默认电平类型	可改电平类型	设备节点
J33	TTL	可改RS232	/dev/ttyS8
J35	TTL	可改RS232	/dev/ttyS10
J30	RS232	可改TTL	/dev/ttyS4
J32	RS232	可改TTL	/dev/ttyS3
J29	RS485	可改TTL	/dev/ttyS1
J31	RS485	可改TTL	/dev/ttyS7

4.4. 声音

4.4.1. 喇叭

喇叭接口为PH2.0-4P连接器J38，如下图所示：

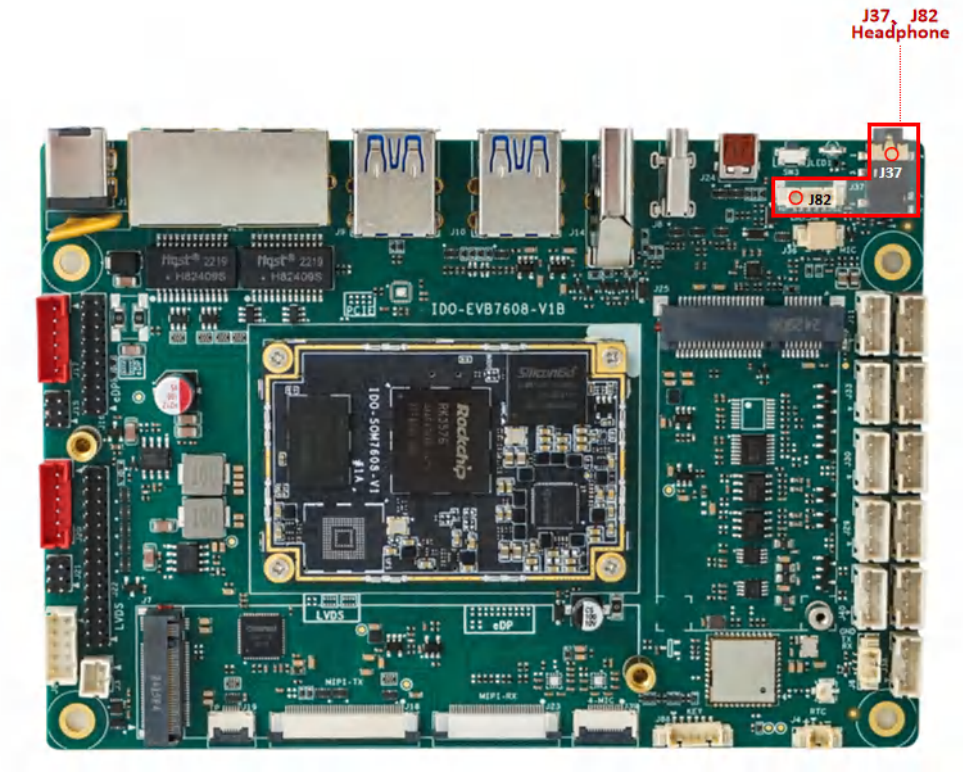


- 支持双声道，每个声道支持4Ω 3W输出

```
#查看声卡
root@industio:~# aplay  -l
**** List of PLAYBACK Hardware Devices ****
card 1: rockchipes8388 [rockchip-es8388], device 0: dailink-multicodecs ES8323 HiFi-0 [dailink-multicodecs ES8323 HiFi-0]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
card 2: rockchiphdmi [rockchip-hdmi], device 0: rockchip-hdmi i2s-hifi-0 [rockchip-hdmi i2s-hifi-0]
  Subdevices: 1/1
  Subdevice #0: subdevice #0

#音频播放到声卡
root@industio:~# aplay -D hw:1,0 xihuangni.wav
Playing WAVE 'xihuangni.wav' : Signed 16 bit Little Endian, Rate 44100 Hz, Stereo

```

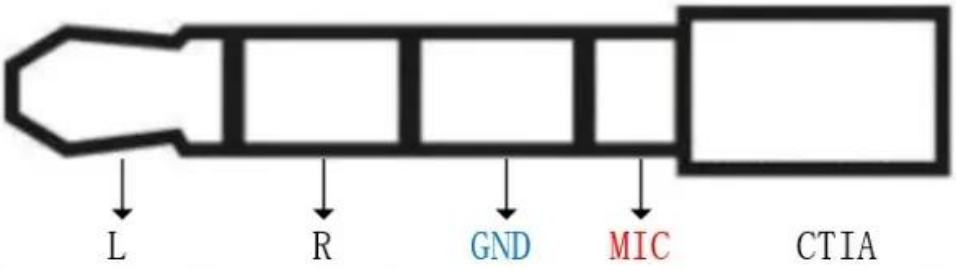
4.4.2. 耳机

主板支持1路3.5mm四节耳机座（CTIA）J37，和1路MX1.25T-5P耳机座子并用，用户可根据需求选用其中1路耳机座子使用 ，如下图所示：



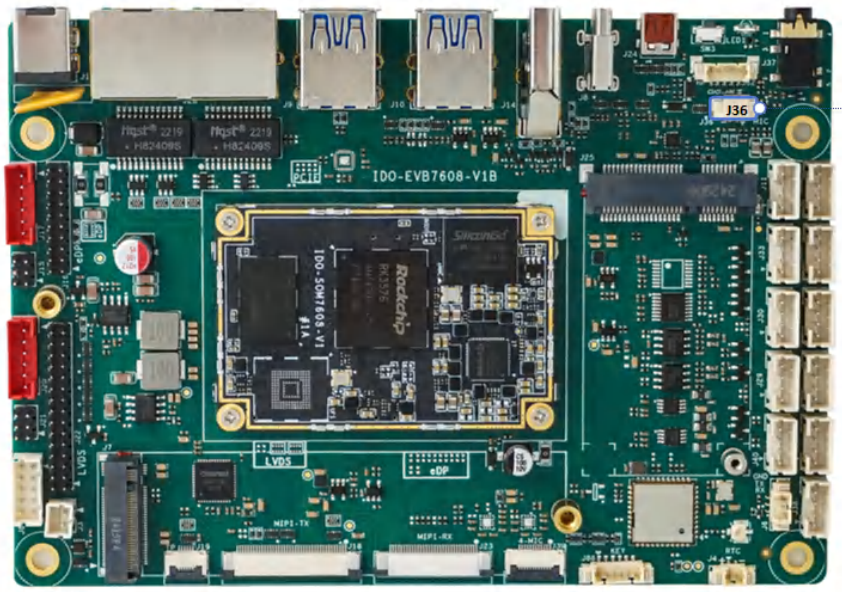
- 支持耳机检测
- 支持耳机录音

CTIA标准四段式耳机，定义如下：



4.4.3. Mic

MX1.25T–2P麦克风接口J36，如下图所示：


J36
Mic 1
J36
IDO-EVB7608-V1B
IDO-SOM7608-V1
LVDS
eDP
MIPI-TX
MIPI-RX

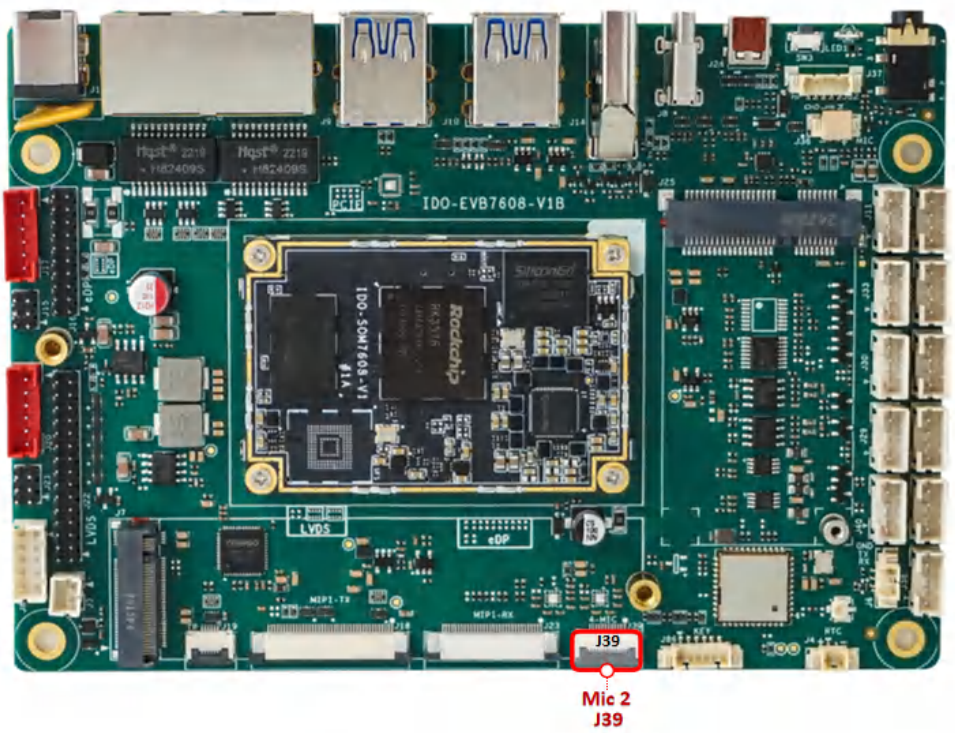
```
#查看录音设备
root@industio:~# arecord -l
**** List of CAPTURE Hardware Devices ****
card 0: rockchiprk628hd [rockchip,rk628hdmiin], device 0: rockchip,rk628hdmiin i2s-hifi-0 [rockchip,rk628hdmiin i2s-hifi-0]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
card 1: rockchipes8388 [rockchip-es8388], device 0: dailink-multicodecs ES8323 HiFi-0 [dailink-multicodecs ES8323 HiFi-0]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
card 2: rockchiphdmi [rockchip-hdmi], device 0: rockchip-hdmi i2s-hifi-0 [rockchip-hdmi i2s-hifi-0]
  Subdevices: 1/1
  Subdevice #0: subdevice #0


#捕获音量设置(不要设置最大值):
root@industio:~# amixer -c 1 cset numid=50 100
numid=50,iface=MIXER,name='Capture Digital Volume'
  ; type=INTEGER,access=rw---R--,values=2,min=0,max=192,step=0
  : values=100,100
  | dBscale-min=-96.00dB,step=0.50dB,mute=1

root@industio:~# amixer -c 1 cget numid=52
numid=52,iface=MIXER,name='Left Channel Capture Volume'
  ; type=INTEGER,access=rw---R--,values=1,min=0,max=8,step=0
  : values=0
  | dBscale-min=0.00dB,step=3.00dB,mute=0

root@industio:~# amixer -c 1 cget numid=53
numid=53,iface=MIXER,name='Right Channel Capture Volume'
  ; type=INTEGER,access=rw---R--,values=1,min=0,max=8,step=0
  : values=0
  | dBscale-min=0.00dB,step=3.00dB,mute=0

自动增益控制(打开):
root@industio:~# amixer -c 1 cset numid=41 3
numid=41,iface=MIXER,name='ALC Capture Function'
  ; type=ENUMERATED,access=rw------,values=1,items=4
  ; Item #0 'Off'
  ; Item #1 'Right'
  ; Item #2 'Left'
  ; Item #3 'Stereo'
```

```
  : values=3

调整目标音量:
root@industio:~# amixer -c 1 cset numid=38 8
numid=38,iface=MIXER,name='ALC Capture Target Volume'
  ; type=INTEGER,access=rw------,values=1,min=0,max=15,step=0
  : values=8
设置噪声门限值:
root@industio:~# amixer -c 1 cset numid=46 15
numid=46,iface=MIXER,name='ALC Capture NG Threshold'
  ; type=INTEGER,access=rw------,values=1,min=0,max=31,step=0
  : values=15
确保捕获静音未开启:
root@industio:~# amixer -c 1 cset numid=51 off
numid=51,iface=MIXER,name='Capture Mute'
  ; type=BOOLEAN,access=rw------,values=1
  : values=off

#主mic开关（mic录音需要打开，耳机录音需要关闭）:
root@industio:~# amixer -c 1 cset  numid=67 on
numid=67,iface=MIXER,name='Main Mic Switch'
  ; type=BOOLEAN,access=rw------,values=1
  : values=on
root@industio:~# echo -n "mic2" > /sys/class/es8388_mic/mic_channel
root@industio:~# arecord -D hw:1,0 -f S16_LE -r 16000 -c 2 test_mic2.wav

#耳机录音:
root@industio:~# amixer -c 1 cset  numid=67 off
root@industio:~# echo -n "mic1" > /sys/class/es8388_mic/mic_channel
root@industio:~# arecord -D hw:1,0 -f S16_LE -r 16000 -c 2 test_mic1.wav
```

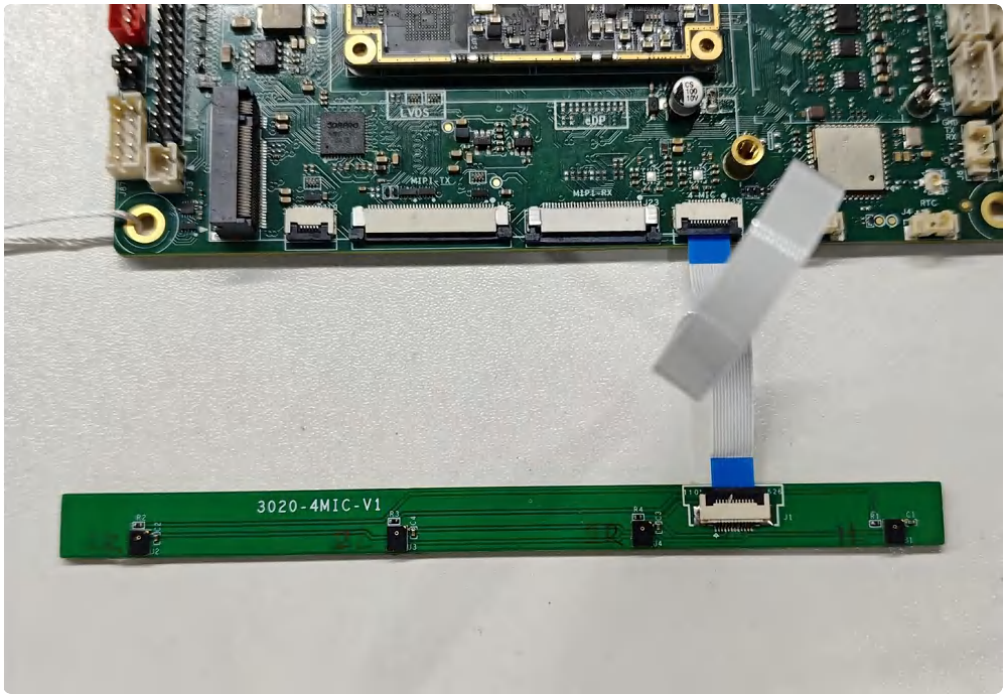
4.4.4. PDM-Mic

注意：PDM-Mic与HDMI-RX功能二选一，软硬件默认配置为HDMI-RX功能。

主板预留 PDM-Mic 阵列接口J39，如下图所示：

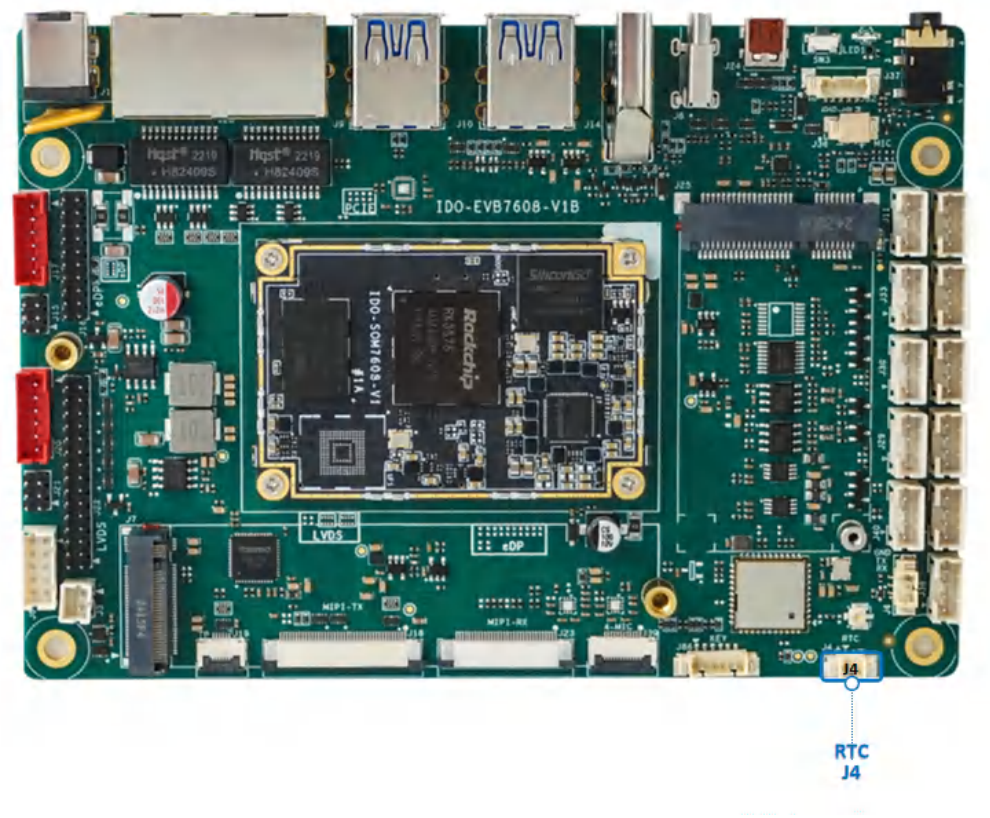


采用FPC05-FDW-12P座子连接Mic阵列，如下图所示：

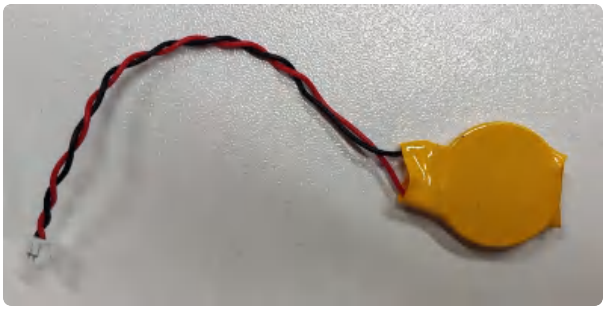


4.5. RTC

MX1.25T-2P RTC电池座J4，如下图所示：



连接3V 纽扣电池，RTC电池参考如下



rtc时间设置方法：

Plain Text

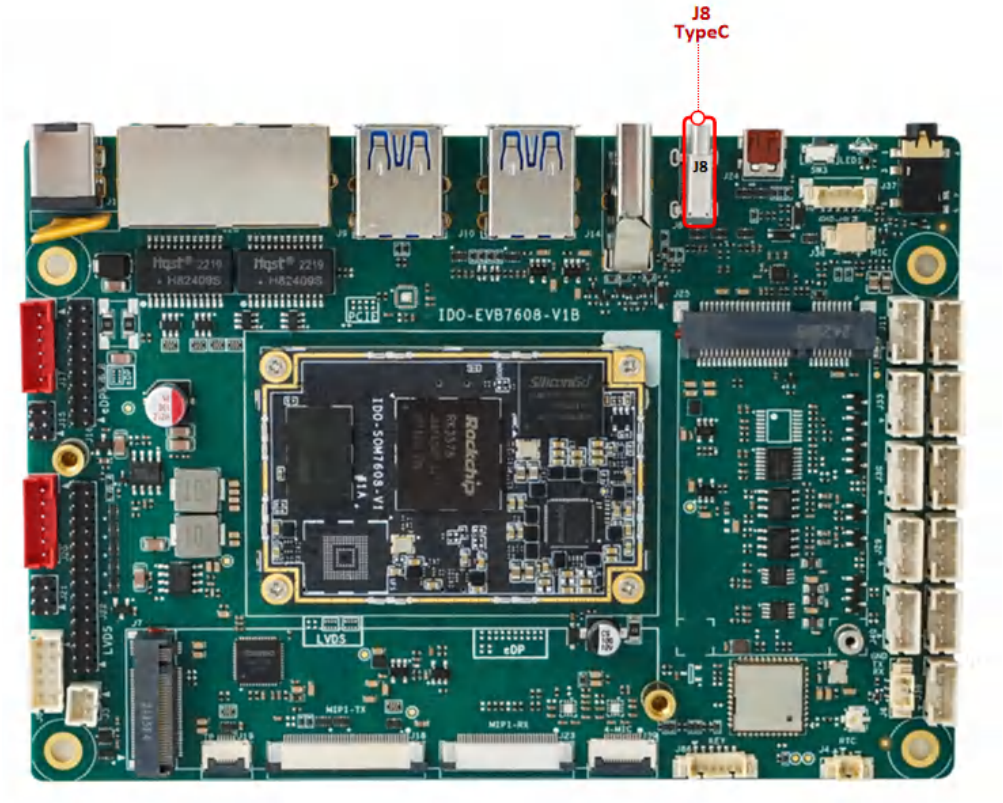
```
#设置时间
root@industio:~# date -s "2024-10-09 14:02:30"

#将rtc时钟调整为与目前的系统时钟一致
root@industio:~# hwclock -w

#获取硬件rtc当前时间（断电重启读取时间没有太大偏差）
root@industio:~# hwclock -r
2024-10-09 14:02:35.945604+00:00
```

4.6. USB接口

主板支持1个TypeC接口（USB3.2 Gen1 OTG+DP1.4），支持4个USB3.0-A接口，2个USB2.0PH2.0-4P 接口，USB对外总供电应小于4A。

4.6.1. TypeC接口

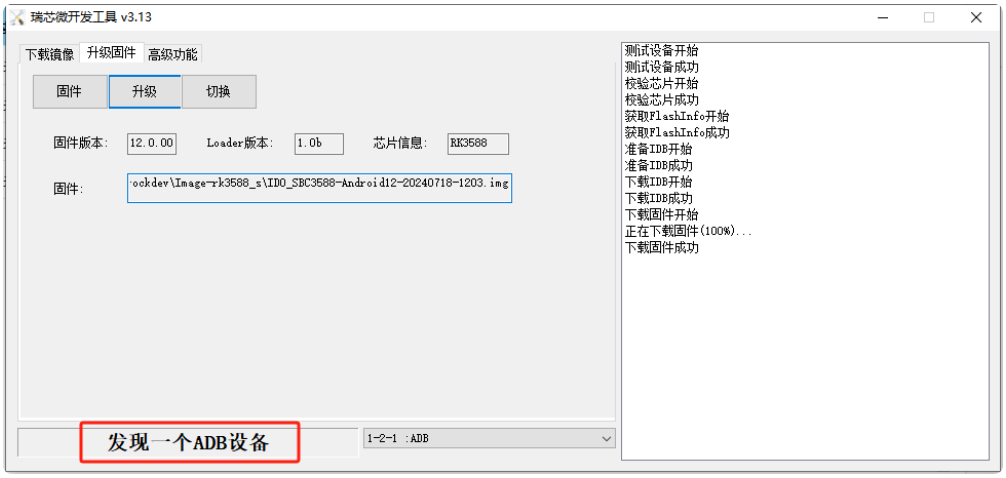
TypeC接口（USB3.2 Gen1 OTG+DP1.4输出）J8，如下图所示：



- 支持Host、Device模式自动切换
- 支持DP显示输出

Device从机模式

使用TypeC数据线连接电脑，烧录工具能发现一个ADB设备，如下图所示：



- 可使用ADB相关开发工具对盒子进行功能调试

Host主机模式

接入TypeC设备，或通过TypeC to USB-A转接头接入USB外设，如下图所示：

- 可识别U盘、键盘、鼠标并正常使用

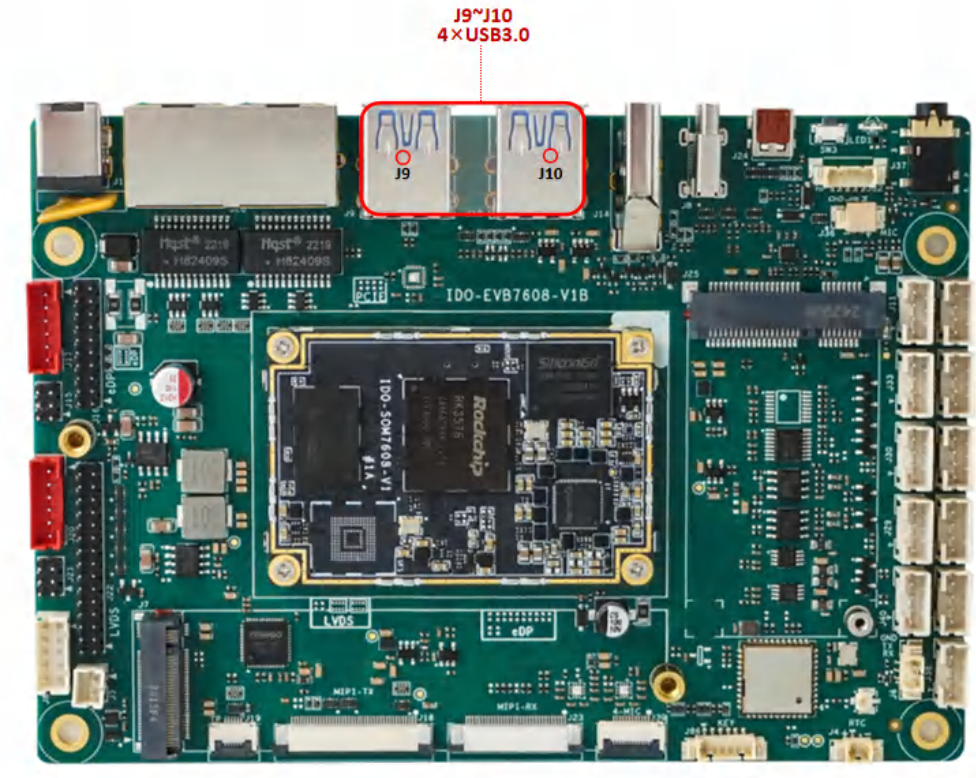
DP模式

通过TypeC全功能数据线接入DP显示器，或通过TYPE-C to HDMI数据线连接HDMI显示器

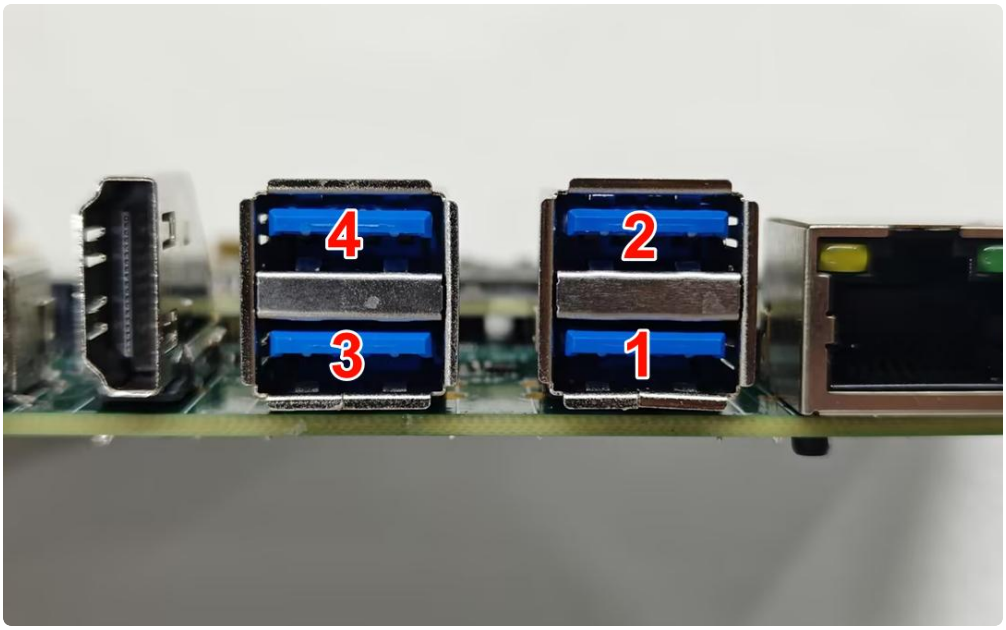
- 主板画面通过TYPE-C DP输出正常，DP声音输出正常

4.6.2. USB3.0接口

主板支持4个USB3.0接口，接口为标准的双层A口J9、J10，如下图所示：



USB母座提供5V@1A供电能力，每个USB端口供电可独立控制，USB序号如下图所示：



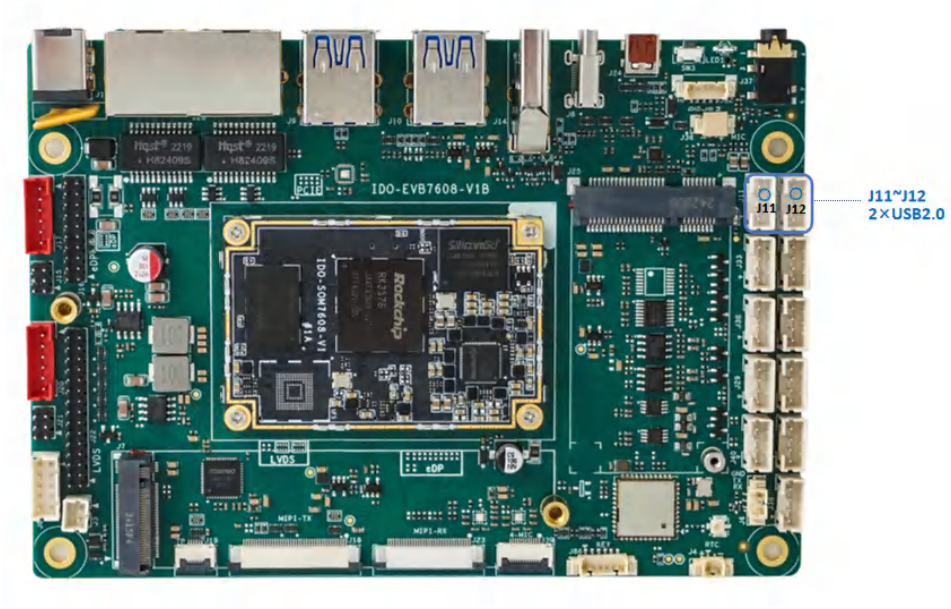
USB电源控制，如下表所示：

USB端口	默认状态	动作	命令
USB1	开启	关闭电源	echo 0 > /sys/class/leds/usb1_pwr/brightness
		开启电源	echo 1 > /sys/class/leds/usb1_pwr/brightness
USB2	开启	关闭电源	echo 0 > /sys/class/leds/usb2_pwr/brightness
		开启电源	echo 1 > /sys/class/leds/usb2_pwr/brightness
USB3	开启	关闭电源	echo 0 > /sys/class/leds/usb3_pwr/brightness
		开启电源	echo 1 > /sys/class/leds/usb3_pwr/brightness
USB4	开启	关闭电源	echo 0 > /sys/class/leds/usb4_pwr/brightness
		开启电源	echo 1 > /sys/class/leds/usb4_pwr/brightness

供电控制说明：设备节点写"0"关闭电源，写"1"开启电源

4.6.3. USB2.0接口

主板配置了2路USB2.0 PH2.0-4P接口，USB接口均提供5V@1A的驱动能力，如下图所示：

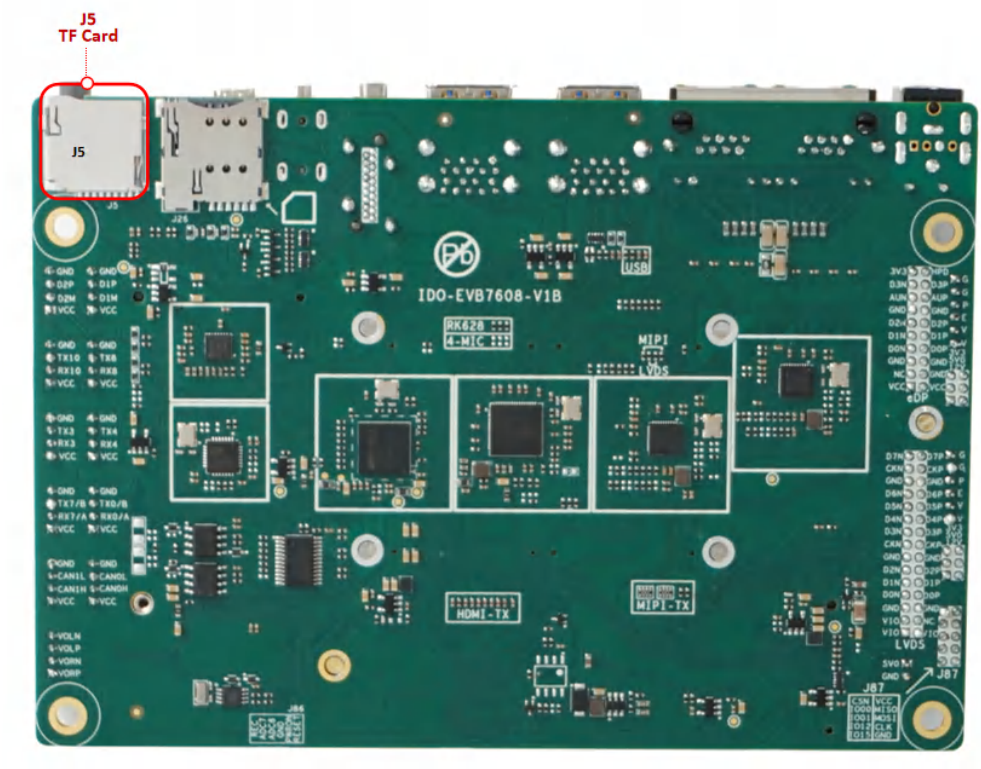


USB端口	默认状态	动作	命令
USB J11	开启	关闭电源	echo 0 > /sys/class/leds/usb5_pwr/brightness
		开启电源	echo 1 > /sys/class/leds/usb5_pwr/brightness
USB J12	开启	关闭电源	echo 0 > /sys/class/leds/usb6_pwr/brightness
		开启电源	echo 1 > /sys/class/leds/usb6_pwr/brightness

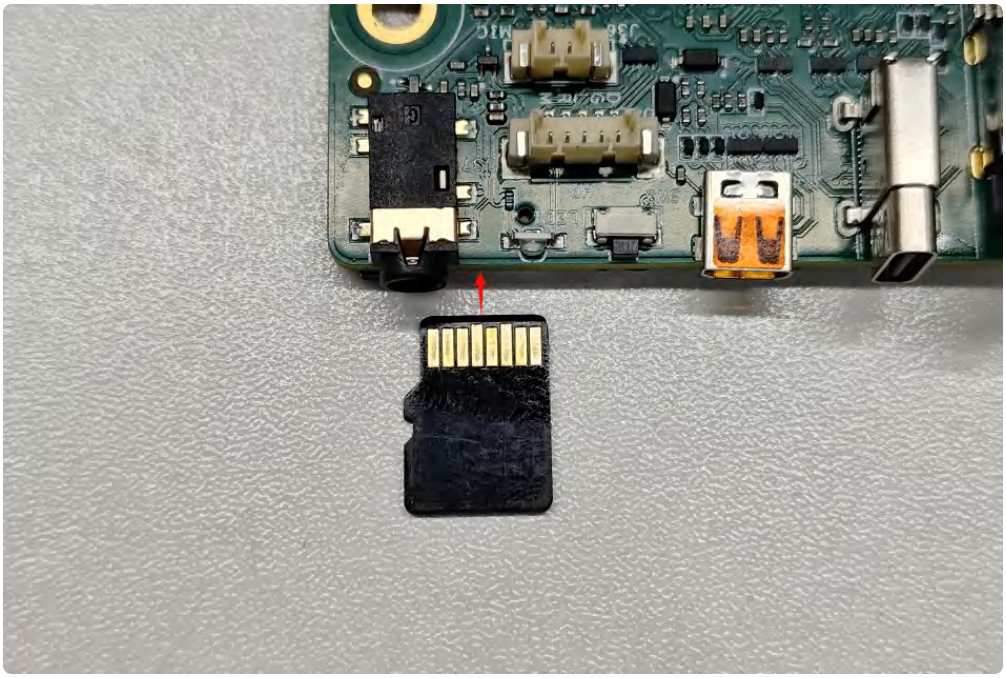
供电控制说明：设备节点写"0"关闭电源，写"1"开启电源

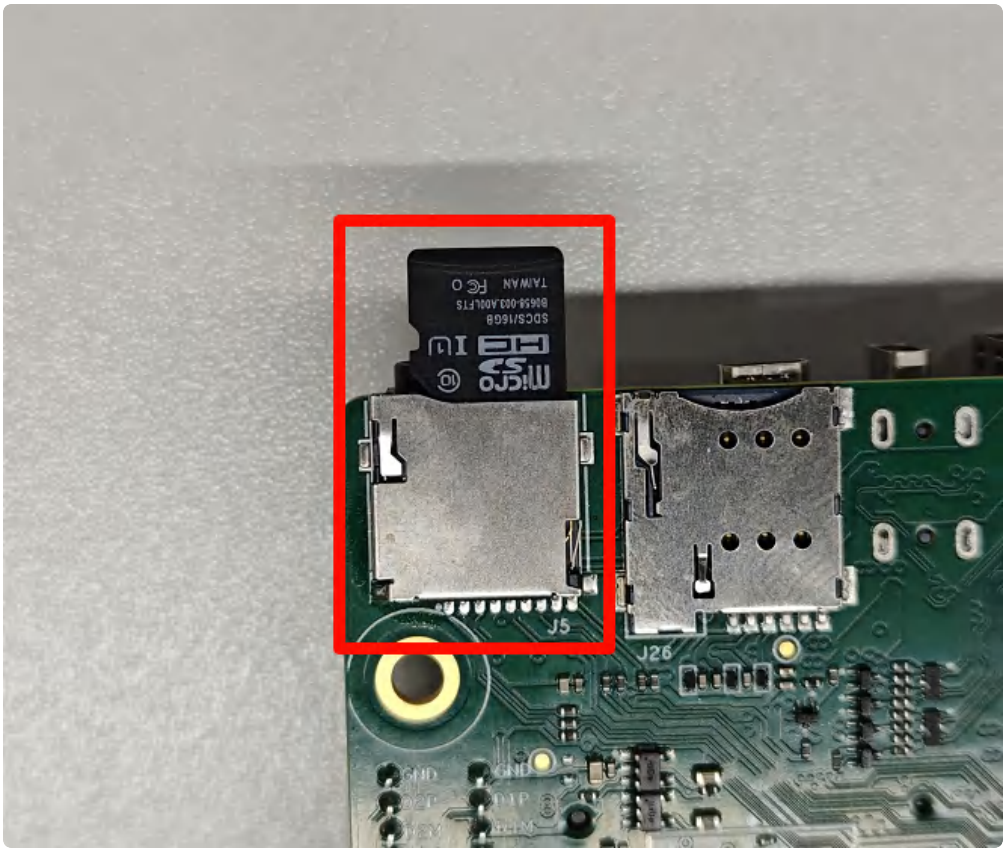
4.7. TF Card

TF卡座支持SDIO3.0， 支持高速SD卡，接口位于主板背面J5，如下图所示：



TF卡安装方向，主板正面朝上时TF卡触点朝上，如下图所示：



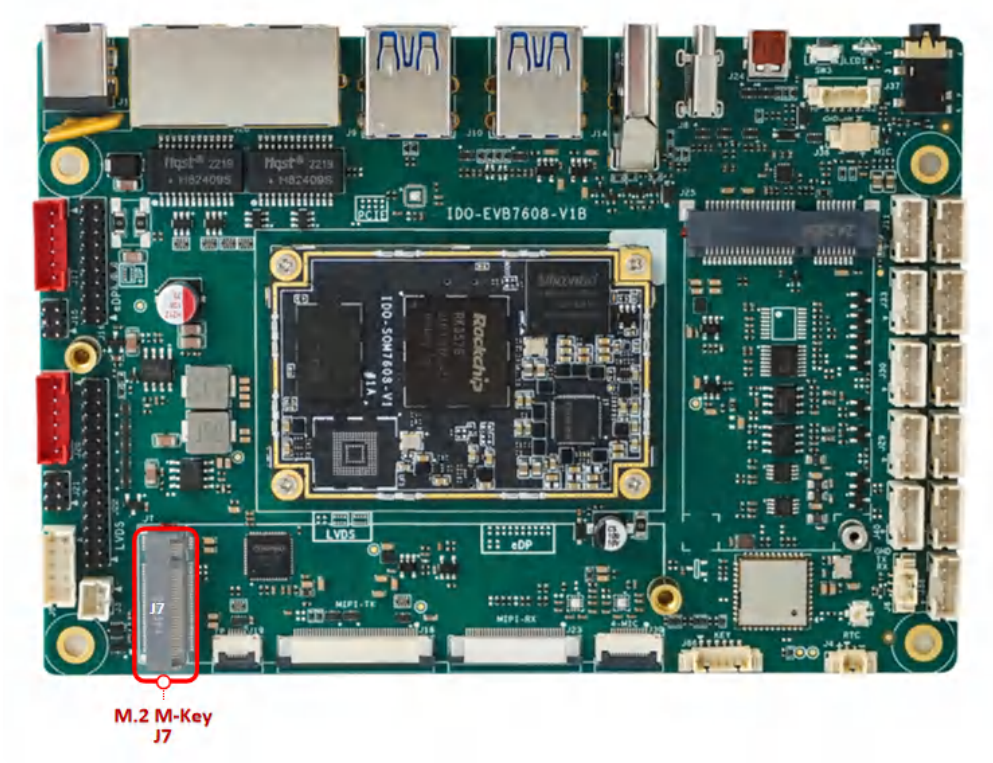

J5
J26

```shell
root@industio:~# fdisk -l
Found valid GPT with protective MBR; using GPT

Disk /dev/mmcblk2: 61071360 sectors, 1148M
Logical sector size: 512
Disk identifier (GUID): 21130000-0000-4444-8000-05e1000030fd
Partition table holds up to 128 entries
First usable sector is 34, last usable sector is 61071326

Number  Start (sector)    End (sector)  Size Name
     1           16384           24575 4096K uboot
     2           24576           32767 4096K misc
     3           32768          163839 64.0M boot
     4          163840          425983  128M recovery
     5          425984          491519 32.0M backup
     6          491520         8880127 4096M userdata
     7         8880128         9142271  128M oem
     8         9142272        61071295 24.7G rootfs
...
Device       Boot StartCHS    EndCHS        StartLBA     EndLBA    Sectors  Size Id Type
/dev/mmcblk1p1    0,2,10      239,254,63         135    3858623    3858849 1884M  e Win95 FAT16 (LBA)
```

注意：因为开机过程中由于emmc分区的变化，可能导致tf卡的分区也会变化，导致无法自动挂载，手动挂载即可。

4.8. M.2接口

IDO-EVB7608-V1主板上使用标准M.2-M-key M.2接口连接座，如下图所示：



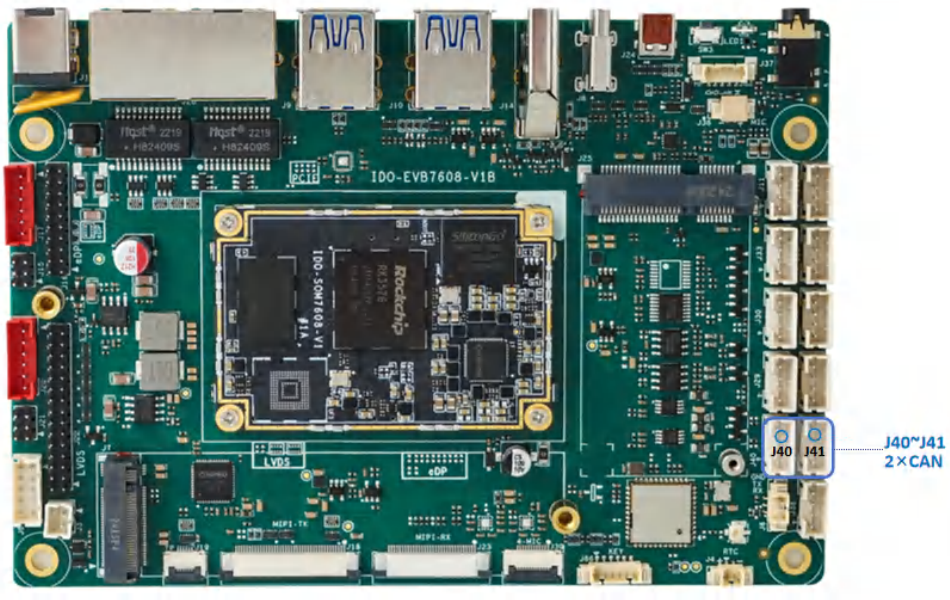
支持PCIe2.1通信，适用2280尺寸NVME固态硬盘。

Shell

```
root@industio:~# fdisk -l
Disk /dev/ram0: 8 MiB, 8388608 bytes, 16384 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes


Disk /dev/ram1: 8 MiB, 8388608 bytes, 16384 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes
...
Disk /dev/nvme0n1: 476.94 GiB, 512110190592 bytes, 1000215216 sectors
Disk model: XPG GAMMIX S11L
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x00000000

Device         Boot Start        End    Sectors   Size Id Type
/dev/nvme0n1p1       2048 1000215182 1000213135 476.9G  c W95 FAT32 (LBA)
```

4.9. CAN接口

主板共有2路CAN接口J40、J41，如下图所示:



分别对应系统节点名称为 can0、can1。

1. 设置CAN参数

Shell

```
# 查看can0配置信息，可以看到can节点的波特率、位时序设置等
# ip -d -s -s link show can0

# 配置can之前需要先把can节点关闭
# ip link set can0 down

# 设置can0异常10ms重启
# ip link set can0 type can restart-ms 10

# 设置can0的比特率为1 Mbps/s
# ip link set can0 type can bitrate 1000000 dbitrate 1000000 fd on

# 打开can0节点
# ip link set can0 up
```

2. 测试收发

两台机器通过收发器连接好，发送方的 can_H 与接收方的 can_H连接，发送方的can_L 与接收方的 can_L连接。

can0 Shell

```
# can0节点发送数据
root@industio:~# ip link set can0 down
root@industio:~# ip link set can0 type can bitrate 1000000 dbitrate 1000000 fd on
root@industio:~# ip link set can0 up
root@industio:~# cansend can0 123#DEADBEEF
root@industio:~# cansend can0 123#DEADBEEF12345679

# can0接收数据
root@industio:~# ip link set can0 down
root@industio:~# ip link set can0 type can bitrate 1000000 dbitrate 1000000 fd on
root@industio:~# ip link set can0 up
root@industio:~# candump can0
```

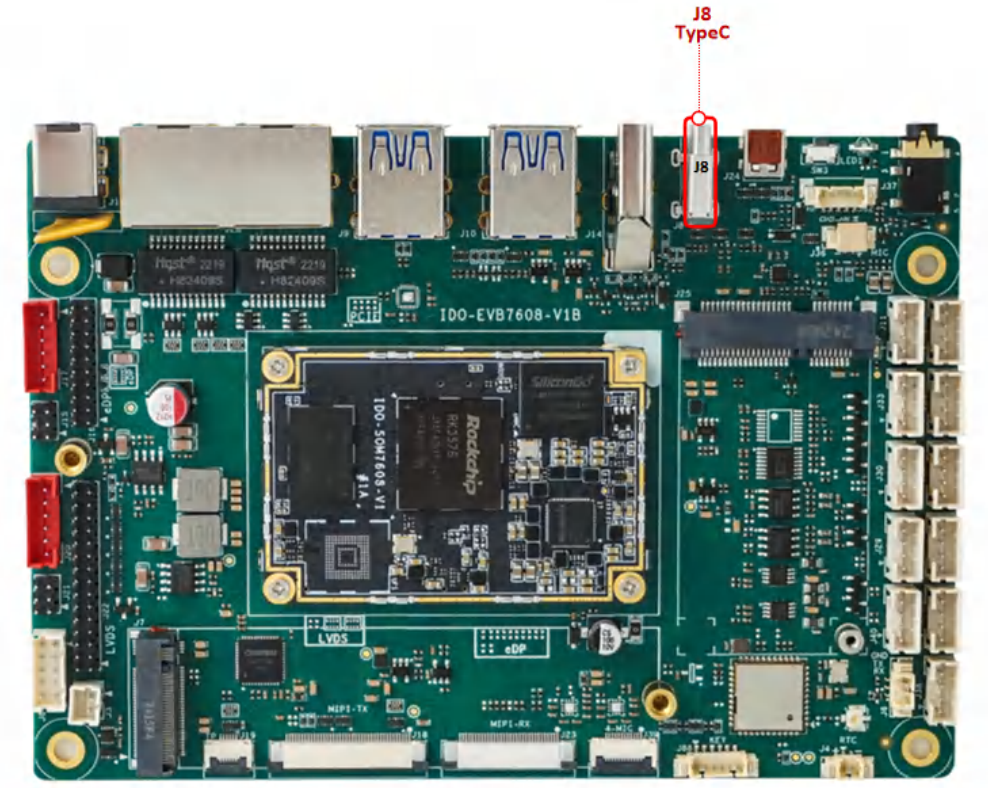
can1 Shell

```
# can1节点发送数据
root@industio:~# ip link set can1 down
root@industio:~# ip link set can1 type can bitrate 1000000 dbitrate 1000000 fd on
root@industio:~# ip link set can1 up
root@industio:~# cansend can1 123#DEADBEEF
root@industio:~# cansend can1 123#DEADBEEF11122233

# can1接收数据
root@industio:~# ip link set can1 down
root@industio:~# ip link set can1 type can bitrate 1000000 dbitrate 1000000 fd on
root@industio:~# ip link set can1 up
root@industio:~# candump can1
```

4.10. LCD显示

4.10.1. DP

DP接口（USB-C）J8，如下图所示：

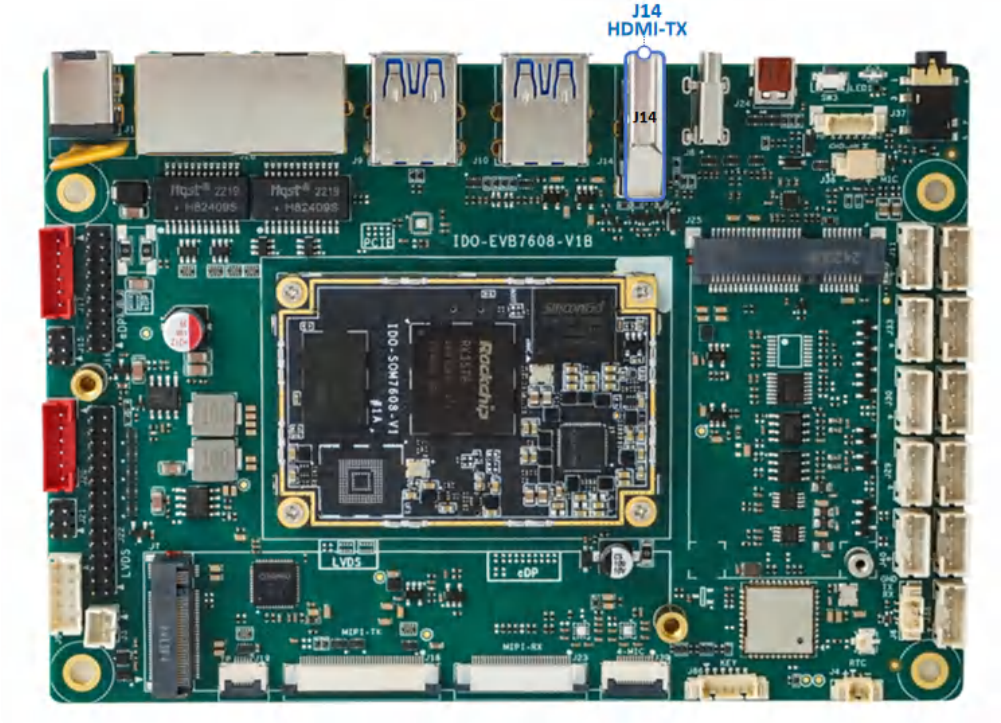


可以使用 USB-C 转 HDMI 高清线连接 HDMI 显示器输出画面，如下图所示：

- 最高支持4K@30fps输出

4.10.2. HDMI-TX

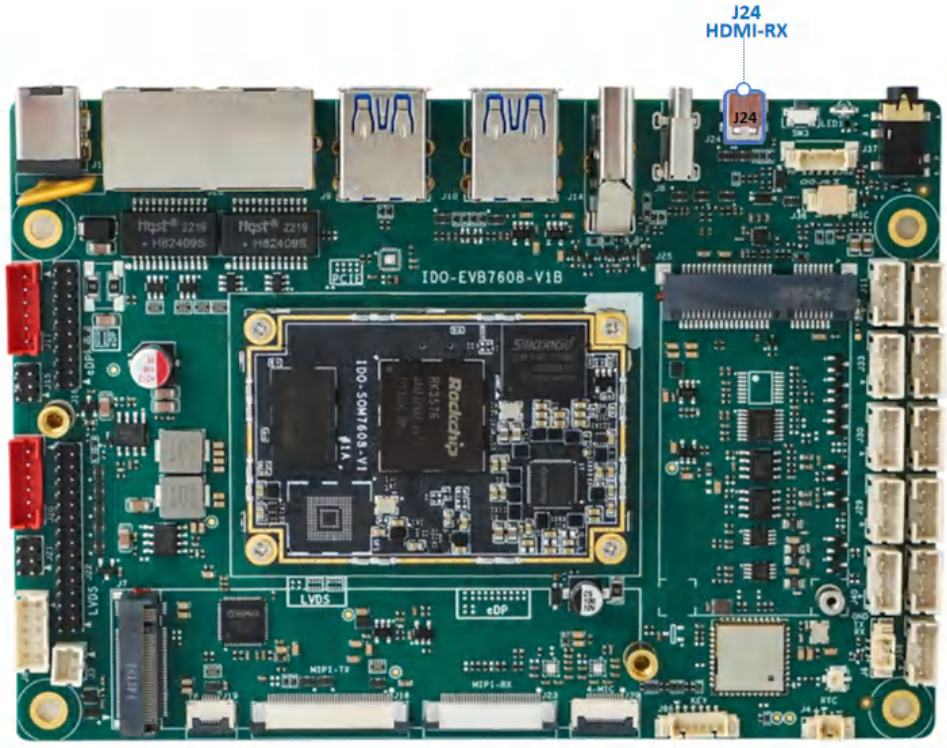
标准HDMI-A接口J14，如下图所示：



- 最高支持 HDMI2.1 8K@60fps 输出

4.10.3. HDMI-RX接口

Micro HDMI接口J24，如下图所示：



预览：

```
root@industio:~# gst-launch-1.0 v4l2src device=/dev/video0 ! video/x-raw,width=1920,height=1080,framerate=30/1 ! videoconvert ! autovideosink
```

- HDMI2.0-RX，支持4K@30fps

录音：

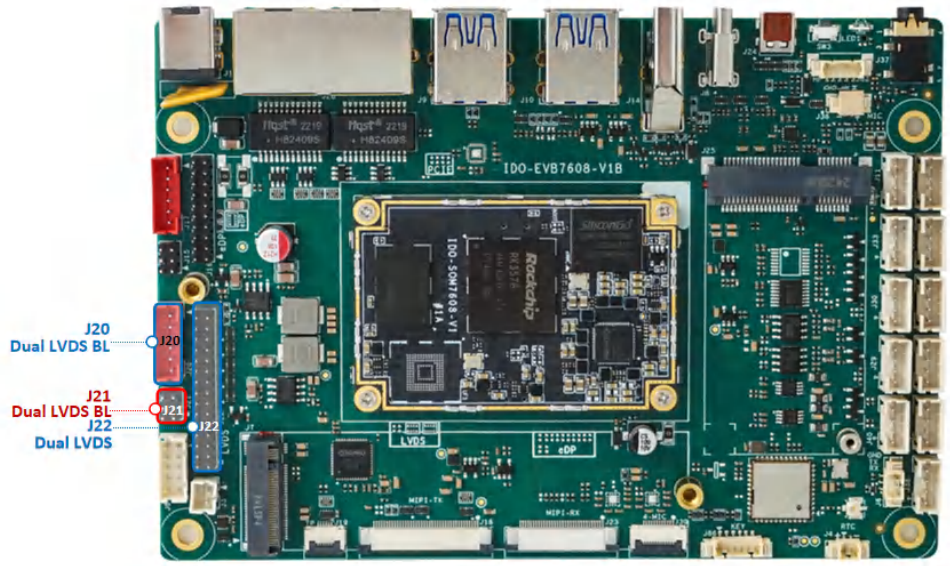
```
#窗口1
root@industio:~# gst-launch-1.0 v4l2src device=/dev/video0 ! video/x-raw,width=1920,height=1080,framerate=30/1 ! videoconvert ! autovideosink

#窗口2
root@industio:/# amixer -c 0 cset numid=35 500
numid=35,iface=PCM,name='SAI4 PCM Read Wait Time MS'
  ; type=INTEGER,access=rw------,values=1,min=0,max=10000,step=1
  : values=500

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

注意：hdmi in录音之前确保另一端的hdmi out已经正常输出音频。

4.10.4. Dual LVDS屏

注意：Dual LVDS接口与MIPI-TX二选一，硬件默认配置为MIPI-TX。

Dual LVDS接口J22，背光接 J20，屏幕供电电压选择接口J21，如下图所示：

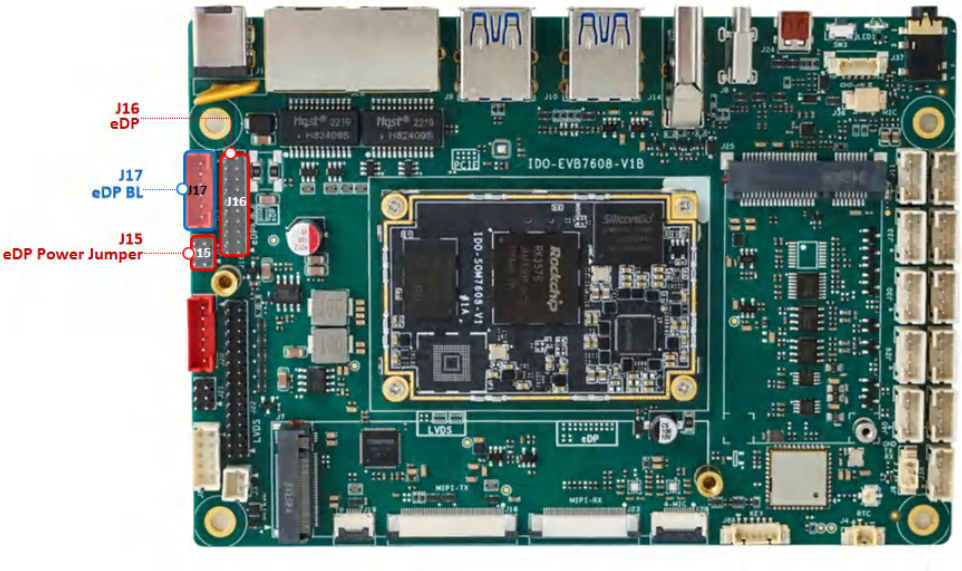


J21屏幕供电电压选择接口，接屏前，需要根据具体的屏幕规格书来确认J21的供电跳线帽接到3.3V、5V或12V，默认3.3V。

4.10.5. eDP

注意：eDP 接口与 HDMI-TX 二选一，硬件默认配置为 HDMI-TX。

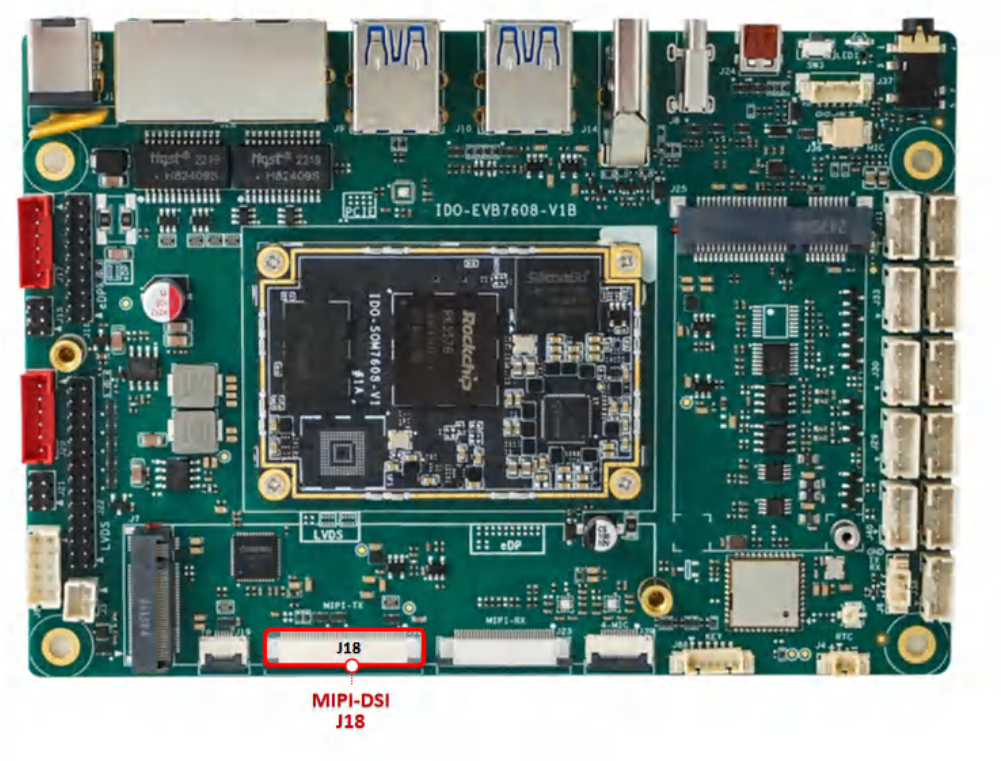
eDP接口J16，eDP背光接 J17，eDP屏幕供电电压选择接口J15，如下图所示：



J15屏幕供电电压选择接口，接屏前，需要根据具体的屏幕规格书来确认J15的供电跳线帽接到3.3V、5V或12V，默认3.3V。

4.10.6. MIPI

J18 MIPI接口为40Pin FPC 0.5mm 上接，如下图所示：

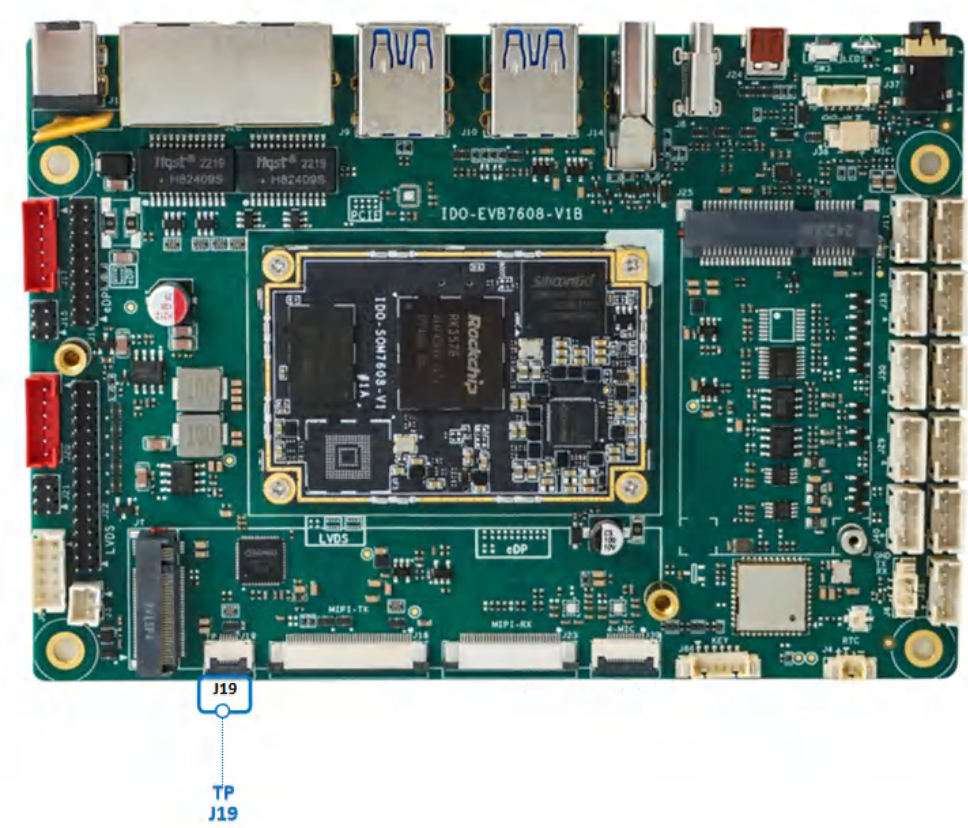


- MIPI供电为3.3V

注意：开发板的MIPI TX和Dual LVDS输出为复用关系，主板默认MIPI TX输出。

4.11. TP接口

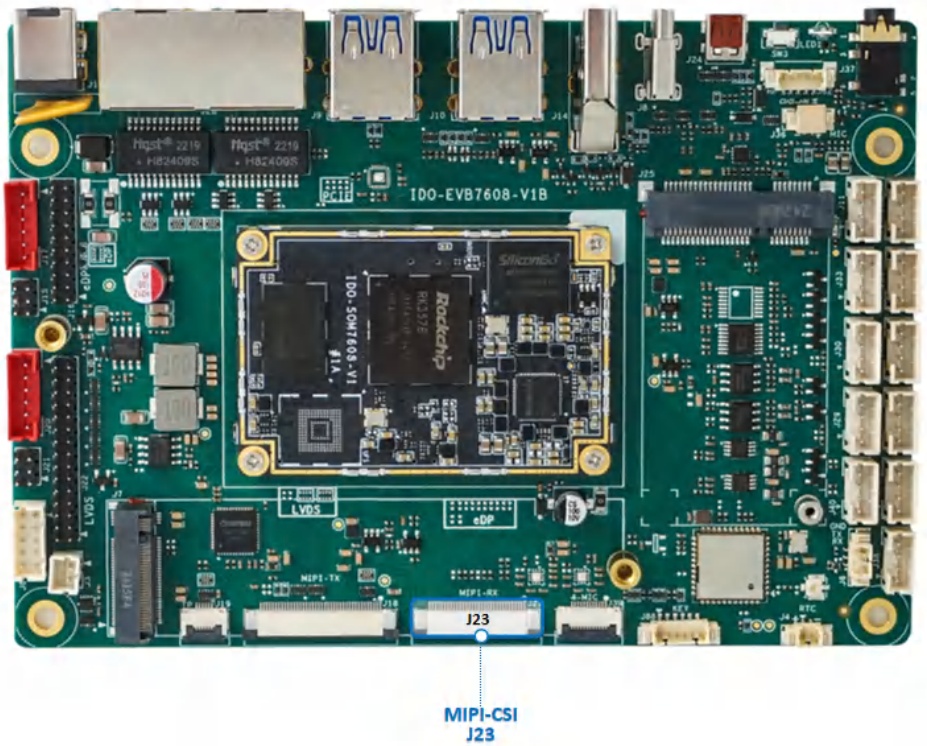
TP接口为(J19) 6Pin FPC 0.5mm座子，如下图所示：



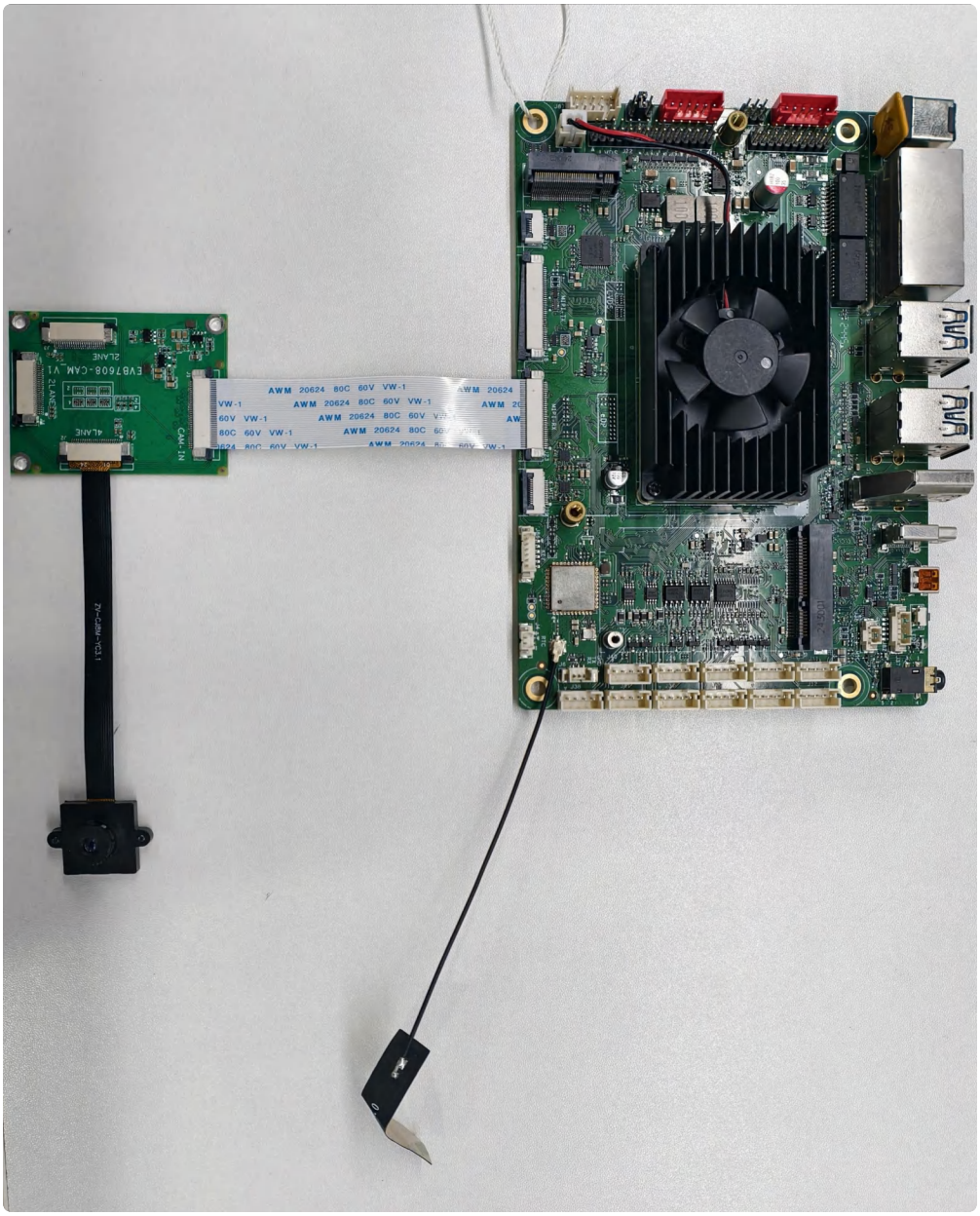
- 触摸 TP 接口接线方法为下接

4.12. MIPI Camera

MIPI Camer接口J23，如下图所示：



使用FPC05-PUW-32P抽拉式上接座子，默认适配IMX415摄像头（可接OV13855），接法如下图所示：



4.12.1. 预览命令：

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

4.12.2. 抓图：

```
root@industio:~# v4l2-ctl --verbose -d /dev/video22 --set-fmt-video=width=1920,height=1080,pixelformat='NV12' --stream-mmap=4 --stream-skip=3  --stream-to=./test-1920x1080-p50.yuv --stream-count=1 --stream-poll
```

查看照片：

```
root@industio:~# ffplay -f rawvideo -video_size 1920x1080 -pixel_format nv12 ./test-1920x1080-p50.yuv
```

4.12.3. 抓视频：

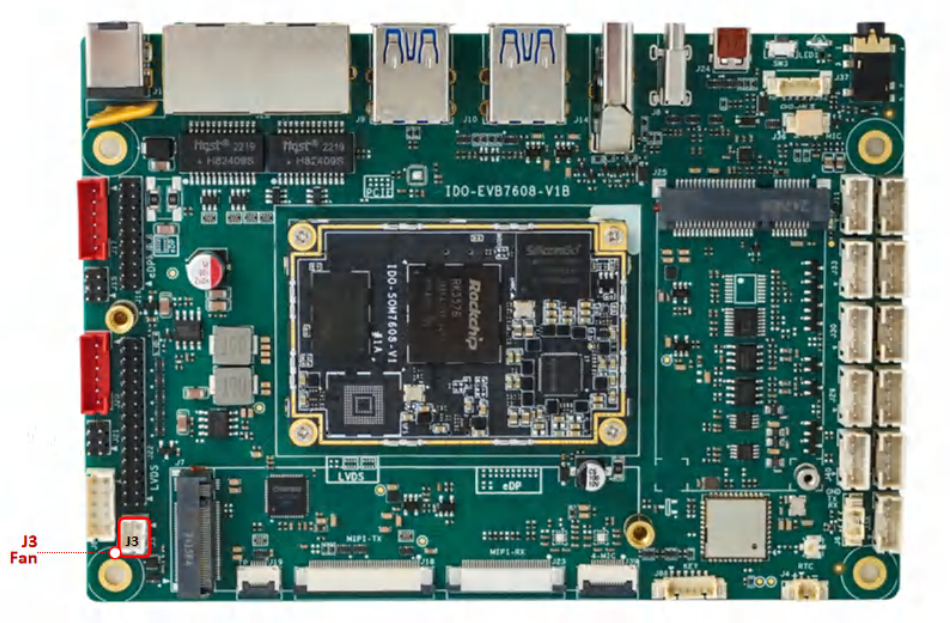
```
root@industio:~# v4l2-ctl --stream-mmap=4  -d /dev/video22 --set-fmt-video=width=1920,height=1080,pixelformat='NV12' --stream-to=./output.yuv
```

播放视频：

```
root@industio:~# ffplay -f rawvideo -video_size 1920x1080 -pixel_format nv12 ./output.yuv
```

4.13. FAN 风扇

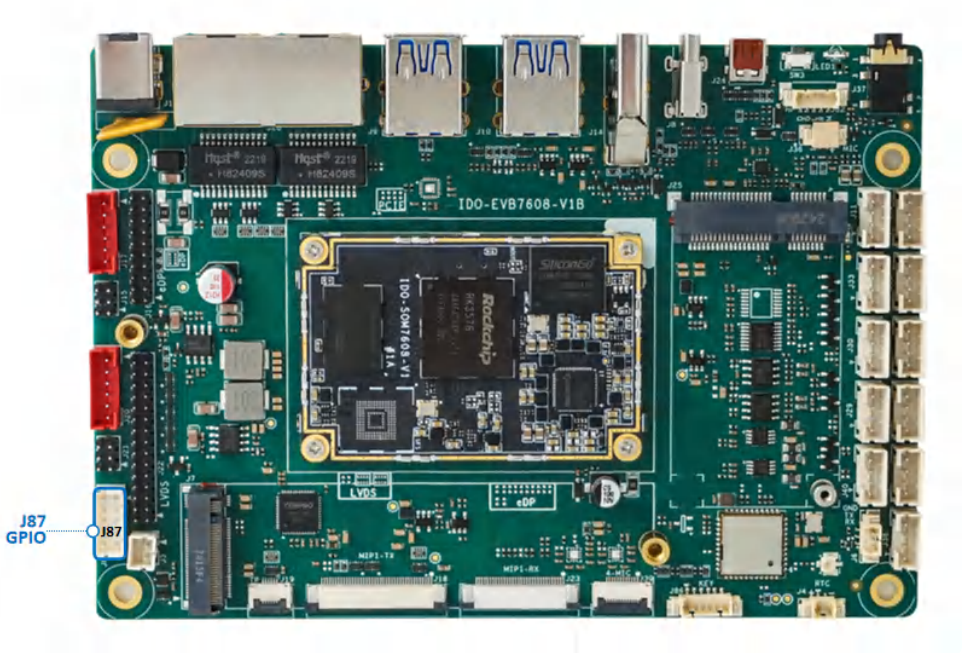
PH2 5V FAN风扇接口J3，如下图所示：



控制功能	控制命令
打开风扇	echo 1 > /sys/class/leds/fan/brightness
关闭风扇	echo 0 > /sys/class/leds/fan/brightness

4.14. GPIO

主板预留了8个GPIO接口J87，如下图所示：



GPIO引脚定义描述，如下图所示：

序号	管脚标号	GPIO 标号	GPIO 序号	LIBGPIOD节点
1	CSN	GPIO3_D0	120	gpiochip3 24
2	VCC	5V/3.3V	/	/
3	IO00	IO0_0	493	gpiochip6 0
4	MISO	GPIO3_C5	117	gpiochip3 21
5	IO01	IO0_1	494	gpiochip6 1
6	MOSI	GPIO3_C6	118	gpiochip3 22
7	IO12	IO1_2	503	gpiochip6 10
8	CLK	GPIO3_C7	119	gpiochip3 23
9	IO15	IO1_5	506	gpiochip6 13
10	GND	地	/	/

GPIO控制，根据上表【LIBGPIOD节点】进行命令操作

libgpiod C

```
# 设置 GPIO3_D0 输出高电平状态
gpioset gpiochip3 24=1

# 设置 GPIO3_D0 输出低电平状态
gpioset gpiochip3 24=0

# 设置 IO00 输出高电平状态
gpioset gpiochip6 0=1

# 设置 IO00 输出低电平状态
gpioset gpiochip6 0=0

# 获取 GPIO3_D0 输入电平状态
gpioget gpiochip3 24

# 获取 IO00 输入电平状态
gpioget gpiochip6 0
```

其中1、4、6、8脚可配置为SPI接口，对应设备树SPI1_M2

序号	管脚标号	GPIO 标号	GPIO 序号

1	CSN	GPIO1_A3	120
4	MISO	GPIO3_C5	117
6	MOSI	GPIO3_C6	118
8	CLK	GPIO3_C7	119

4.15. GPU

执行glmark2-es2 开启gpu测试窗口：

```
root@industio:/# glmark2-es2
arm_release_ver: g13p0-01eac0, rk_so_ver: 11
=======================================================
    glmark2 2023.01
=======================================================
    OpenGL Information
    GL_VENDOR:      ARM
    GL_RENDERER:    Mali-G52
    GL_VERSION:     OpenGL ES 3.2 v1.g13p0-01eac0.e22bb8d1d7959f189f908963172c514e
    Surface Config: buf=32 r=8 g=8 b=8 a=8 depth=24 stencil=0 samples=0
    Surface Size:   800x600 windowed
=======================================================
[build] use-vbo=false: FPS: 227 FrameTime: 4.422 ms
[build] use-vbo=true: FPS: 222 FrameTime: 4.520 ms
[texture] texture-filter=nearest: FPS: 276 FrameTime: 3.631 ms
[texture] texture-filter=linear: FPS: 218 FrameTime: 4.599 ms
[texture] texture-filter=mipmap: FPS: 213 FrameTime: 4.710 ms
[shading] shading=gouraud: FPS: 169 FrameTime: 5.942 ms
[shading] shading=blinn-phong-inf: FPS: 188 FrameTime: 5.336 ms
[shading] shading=phong: FPS: 177 FrameTime: 5.667 ms
[shading] shading=cel: FPS: 168 FrameTime: 5.955 ms
[bump] bump-render=high-poly: FPS: 161 FrameTime: 6.247 ms
[bump] bump-render=normals: FPS: 198 FrameTime: 5.054 ms
[bump] bump-render=height: FPS: 189 FrameTime: 5.301 ms
[effect2d] kernel=0,1,0;1,-4,1;0,1,0;: FPS: 158 FrameTime: 6.354 ms
[effect2d] kernel=1,1,1,1,1;1,1,1,1,1;1,1,1,1,1;: FPS: 118 FrameTime: 8.544 ms
[pulsar] light=false:quads=5:texture=false: FPS: 186 FrameTime: 5.387 ms
[desktop] blur-radius=5:effect=blur:passes=1:separable=true:windows=4: FPS: 118 FrameTime: 8.491 ms
[desktop] effect=shadow:windows=4: FPS: 148 FrameTime: 6.794 ms
[buffer] columns=200:interleave=false:update-dispersion=0.9:update-fraction=0.5:update-method=map: FPS: 82 FrameTime: 12.242 ms
[buffer] columns=200:interleave=false:update-dispersion=0.9:update-fraction=0.5:update-method=subdata: FPS: 83 FrameTime: 12.054 ms
[buffer] columns=200:interleave=true:update-dispersion=0.9:update-fraction=0.5:update-method=map: FPS: 98 FrameTime: 10.270 ms
[ideas] speed=duration: FPS: 139 FrameTime: 7.215 ms
[jellyfish] <default>: FPS: 227 FrameTime: 4.406 ms
[terrain] <default>: FPS: 57 FrameTime: 17.605 ms
[shadow] <default>: FPS: 168 FrameTime: 5.974 ms
[refract] <default>: FPS: 83 FrameTime: 12.066 ms

```

```
[conditionals] fragment-steps=0:vertex-steps=0: FPS: 301 FrameTime: 3.325 ms
[conditionals] fragment-steps=5:vertex-steps=0: FPS: 286 FrameTime: 3.501 ms
[conditionals] fragment-steps=0:vertex-steps=5: FPS: 278 FrameTime: 3.602 ms
[function] fragment-complexity=low:fragment-steps=5: FPS: 265 FrameTime: 3.782 ms
[function] fragment-complexity=medium:fragment-steps=5: FPS: 276 FrameTime: 3.629 ms
[loop] fragment-loop=false:fragment-steps=5:vertex-steps=5: FPS: 289 FrameTime: 3.470 ms
[loop] fragment-steps=5:fragment-uniform=false:vertex-steps=5: FPS: 290 FrameTime: 3.449 ms
[loop] fragment-steps=5:fragment-uniform=true:vertex-steps=5: FPS: 287 FrameTime: 3.490 ms
=======================================================
                                  glmark2 Score: 191
```

4.16. NPU

使用yolov5进行rknn模型转换测试：

4.16.1. 编译rknn_yolov5_demo：

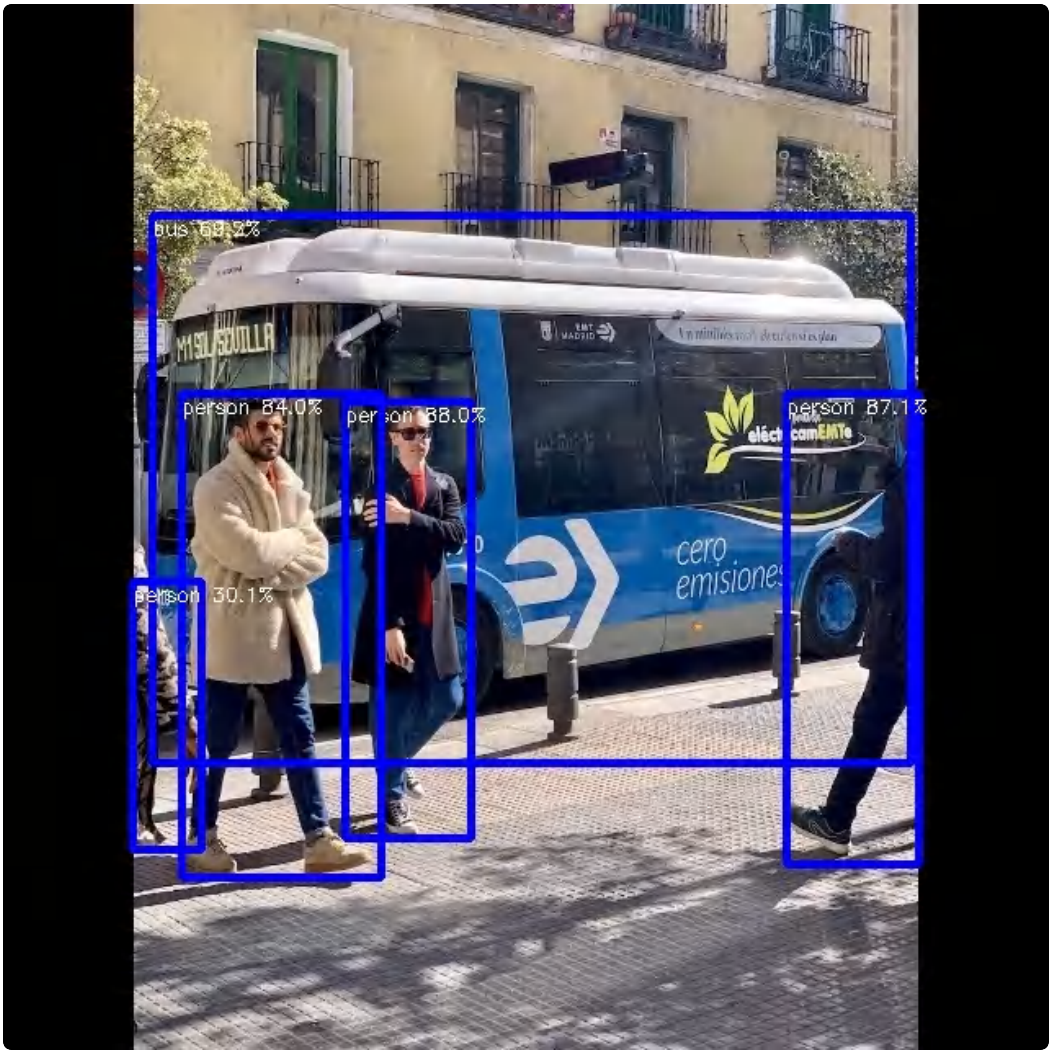
```shell
root@industio:/# unzip rknn-toolkit2-master.zip
root@industio:/# cd rknn-toolkit2-master/rknpu2/examples/3rdparty/mpp/Linux/aarch64
root@industio:/# ln -srf librockchip_mpp.so.0 librockchip_mpp.so
root@industio:/# ln -srf librockchip_mpp.so.0 librockchip_mpp.so.1

root@industio:/# cd -
root@industio:/# cd rknn-toolkit2-master/rknpu2/examples/rknn_yolov5_demo
root@industio:/# chmod a+x build-linux.sh
root@industio:/# export GCC_COMPILER=/usr/bin/aarch64-linux-gnu
root@linaro-alip:/mnt/rknn-toolkit2-master/rknpu2/examples/rknn_yolov5_demo# ./build-linux.sh -t rk3576 -a aarch64 -b Release
./build-linux.sh -t rk3576 -a aarch64 -b Release
/usr/bin/aarch64-linux-gnu
==================================
TARGET_SOC=RK3576
TARGET_ARCH=aarch64
BUILD_TYPE=Release
BUILD_DIR=/mnt/rknn-toolkit2-master/rknpu2/examples/rknn_yolov5_demo/build/build_RK3576_linux_aarch64_Release
CC=/usr/bin/aarch64-linux-gnu-gcc
CXX=/usr/bin/aarch64-linux-gnu-g++
==================================
-- Configuring done
-- Generating done
-- Build files have been written to: /mnt/rknn-toolkit2-master/rknpu2/examples/rknn_yolov5_demo/build/build_RK3576_linux_aarch64_Release
[ 60%] Built target rknn_yolov5_video_demo
[100%] Built target rknn_yolov5_demo
[ 40%] Built target rknn_yolov5_demo
[100%] Built target rknn_yolov5_video_demo
Install the project...
-- Install configuration: "Release"
-- Up-to-date: /mnt/rknn-toolkit2-master/rknpu2/examples/rknn_yolov5_demo/install/rknn_yolov5_demo_Linux/./rknn_yolov5_demo
-- Up-to-date: /mnt/rknn-toolkit2-master/rknpu2/examples/rknn_yolov5_demo/install/rknn_yolov5_demo_Linux/lib/librknnrt.so
-- Up-to-date: /mnt/rknn-toolkit2-master/rknpu2/examples/rknn_yolov5_demo/install/rknn_yolov5_demo_Linux/lib/librga.so
-- Up-to-date: /mnt/rknn-toolkit2-master/rknpu2/examples/rknn_yolov5_demo/install/rknn_yolov5_demo_Linux/./model/RK3576
-- Up-to-date: /mnt/rknn-toolkit2-master/rknpu2/examples/rknn_yolov5_demo/install/rknn_yolov5_demo_Linux/./model/RK3576/yolov5s-640-640.rknn

```

```
-- Up-to-date: /mnt/rknn-toolkit2-master/rknpu2/examples/rknn_yolov5_demo/install/rknn_yolov5_demo_Linux/./model/bus.jpg
-- Up-to-date: /mnt/rknn-toolkit2-master/rknpu2/examples/rknn_yolov5_demo/install/rknn_yolov5_demo_Linux/./model/coco_80_labels_list.txt
-- Up-to-date: /mnt/rknn-toolkit2-master/rknpu2/examples/rknn_yolov5_demo/install/rknn_yolov5_demo_Linux/./rknn_yolov5_video_demo
-- Up-to-date: /mnt/rknn-toolkit2-master/rknpu2/examples/rknn_yolov5_demo/install/rknn_yolov5_demo_Linux/lib/librockchip_mpp.so
-- Up-to-date: /mnt/rknn-toolkit2-master/rknpu2/examples/rknn_yolov5_demo/install/rknn_yolov5_demo_Linux/lib/libmk_api.so
```

4.16.2. 测试：

```shell
root@industio:/# cd install/rknn_yolov5_demo_Linux
root@industio:/# chmod a+x rknn_yolov5_demo
root@industio:/# ./rknn_yolov5_demo ./model/RK3576/yolov5s-640-640.rknn ./model/bus.jpg
post process config: box_conf_threshold = 0.25, nms_threshold = 0.45
Loading mode...
sdk version: 2.3.2 (429f97ae6b@2025-04-09T09:09:27) driver version: 0.9.8
model input num: 1, output num: 3
  index=0, name=images, n_dims=4, dims=[1, 640, 640, 3], n_elems=1228800, size=1228800, w_stride = 640, size_with_stride=1228800, fmt=NHWC, type=INT8, qnt_type=AFFINE, zp=-128, scale=0.003922
  index=0, name=output0, n_dims=4, dims=[1, 255, 80, 80], n_elems=1632000, size=1632000, w_stride = 0, size_with_stride=1638400, fmt=NCHW, type=INT8, qnt_type=AFFINE, zp=-128, scale=0.003922
  index=1, name=286, n_dims=4, dims=[1, 255, 40, 40], n_elems=408000, size=408000, w_stride = 0, size_with_stride=491520, fmt=NCHW, type=INT8, qnt_type=AFFINE, zp=-128, scale=0.003922
  index=2, name=288, n_dims=4, dims=[1, 255, 20, 20], n_elems=102000, size=102000, w_stride = 0, size_with_stride=163840, fmt=NCHW, type=INT8, qnt_type=AFFINE, zp=-128, scale=0.003922
model is NHWC input fmt
model input height=640, width=640, channel=3
Read ./model/bus.jpg ...
img width = 640, img height = 640
once run use 41.249000 ms
loadLabelName ./model/coco_80_labels_list.txt
person @ (209 243 286 510) 0.879723
person @ (479 238 560 526) 0.870588
person @ (109 237 232 534) 0.828112
bus @ (93 129 553 464) 0.700761
person @ (79 353 122 517) 0.307297
save detect result to ./out.jpg
loop count = 10 , average run  28.800200 ms
```

执行完后会得到转换后的模型图片，如下所示：



4.17. QT

运行QT动画测试程序：

```
source /usr/local/qt5/env.sh
/usr/local/qt5/examples/widgets/animation/moveblocks/moveblocks
```

运行QT虚拟键盘测试程序：

```
source /usr/local/qt5/env.sh
/usr/local/qt5/examples/virtualkeyboard/basic/basic
```

运行QT Webengine测试程序：

```
source /usr/local/qt5/env.sh
/usr/local/qt5/examples/webenginewidgets/simplebrowser/simplebrowser www.baidu.com
```