

IDO-EVB3588-V1 Linux使用手册



IDO-EVB3588-V1 Linux使用手册

深圳触觉智能科技有限公司

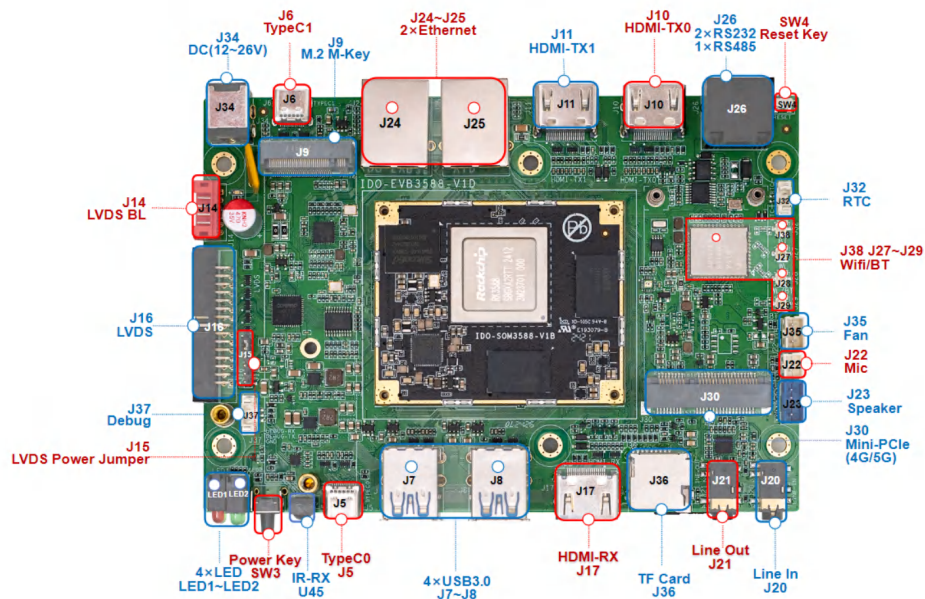
www.industio.cn

文档修订历史

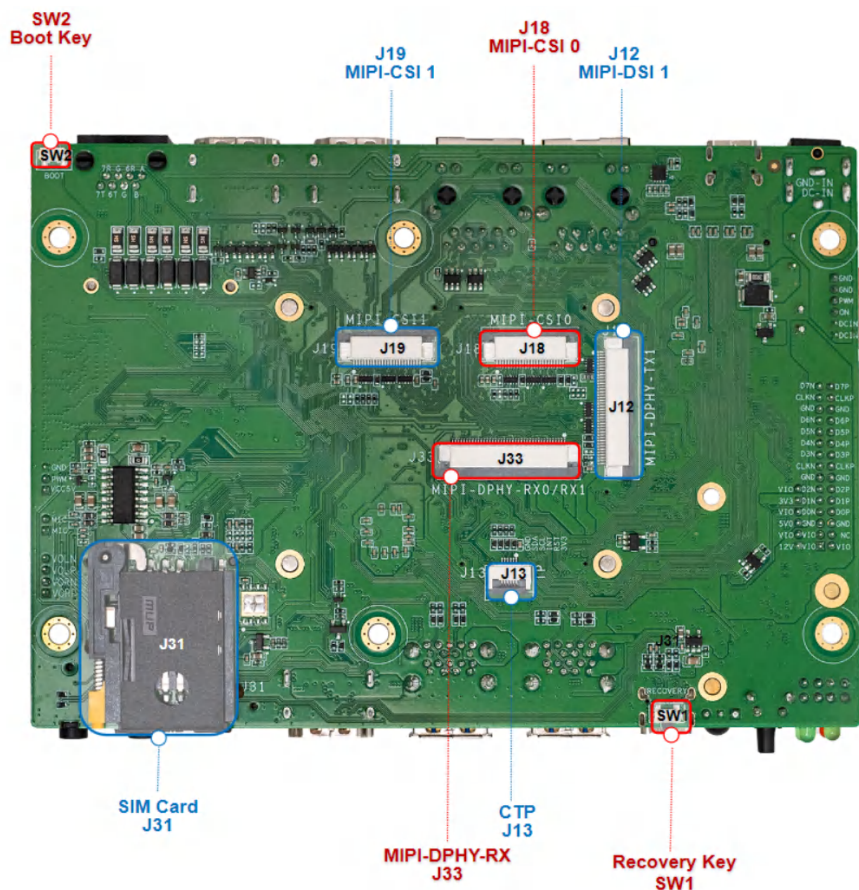
版本	PCBA版本	修订内容	修订	审核	日期
V1.0	V1D	创建文档	MHK	IDO	2025/07/24

1. 硬件资源概况

1.1. 主板照片



IDO-EVB3588-V1D 正面实物图



IDO-EVB3588-V1D 背面实物图

1.2. 硬件资源

序号	名称	描述
1	内核版本	Linux 6.1.118
2	系统版本	Buildroot2024/Debian12/Ubuntu22.04
3	内存	LPDDR4 (8G)
4	存储	eMMC5.1 (64GB)
5	供电	DC接口12V@2A

6	显示	2 × HDMI 2.1 接口, 支持8K@60fps输出 2 × DP接口, 支持8K@30fp输出 1 × MIPI DSI 接口, 支持1080P@60fps输出 1 × Dual LVDS接口, 支持1080P@60fps输出
7	触摸	I2C触摸 (FPC-6P)
8	视频输入	1 × HDMI-IN, 支持 (4K/60fps, HDCP2.3)
9	USB OTG	USB OTG Type-C ×1
10	USB HOST	USB3.0 HOST(Type-A) ×4 USB Type-C ×1
11	TF Card	TF Card ×1
12	M.2	PCIE 3.0 ×1
13	以太网	千兆以太网 ×2
14	WIFI/BT	AP6275S
15	扬声器	PH2.0-4P(4ohm 3W)
16	耳机	CTIA标准四节耳机座
17	音频输入	Line-in (3.5mm) ×1 MIC (PH2.0-2P) ×1
18	Camera	OV13850/IMX415
19	串口	TTL ×1 RS232 ×1 RS485 ×1
20	调试串口	TTL
21	RTC	HYM8563 ×1
22	LED指示灯	系统指示灯 ×1 电源指示灯 ×1 4G/5G指示灯 ×1 用户自定义指示灯 ×1

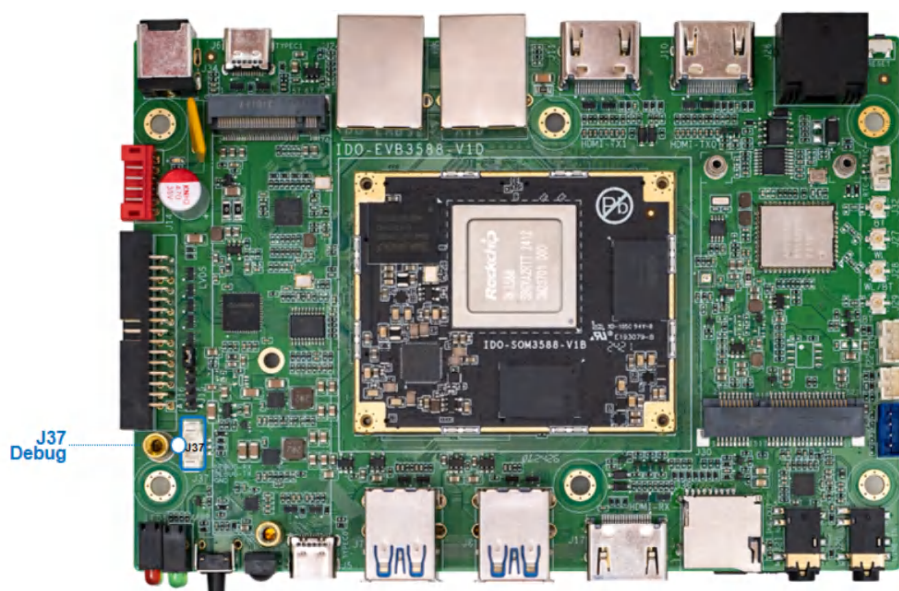
23	红外	板载红外接收头 ×1
24	4G/5G	EC20/EC25 4G模块 RG200U 5G模块
25	按键	Power ×1 Recovery ×1 Boot ×1 Reset ×1

2. IDO-EVB3588-V1D-Debian系统

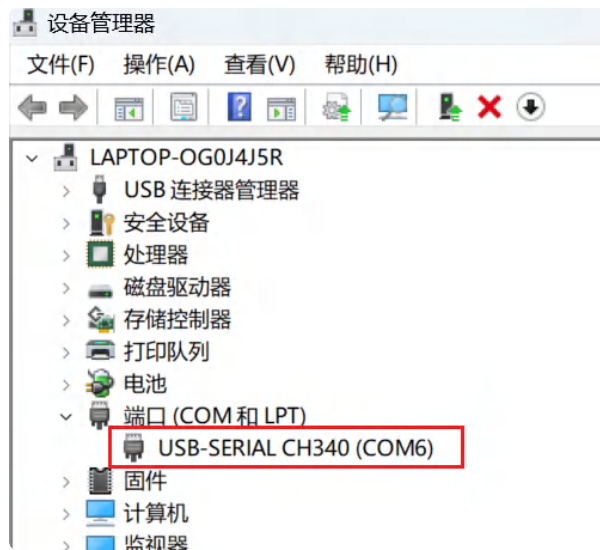
2.1. 功能及接口使用方法

2.1.1. DEBUG UART

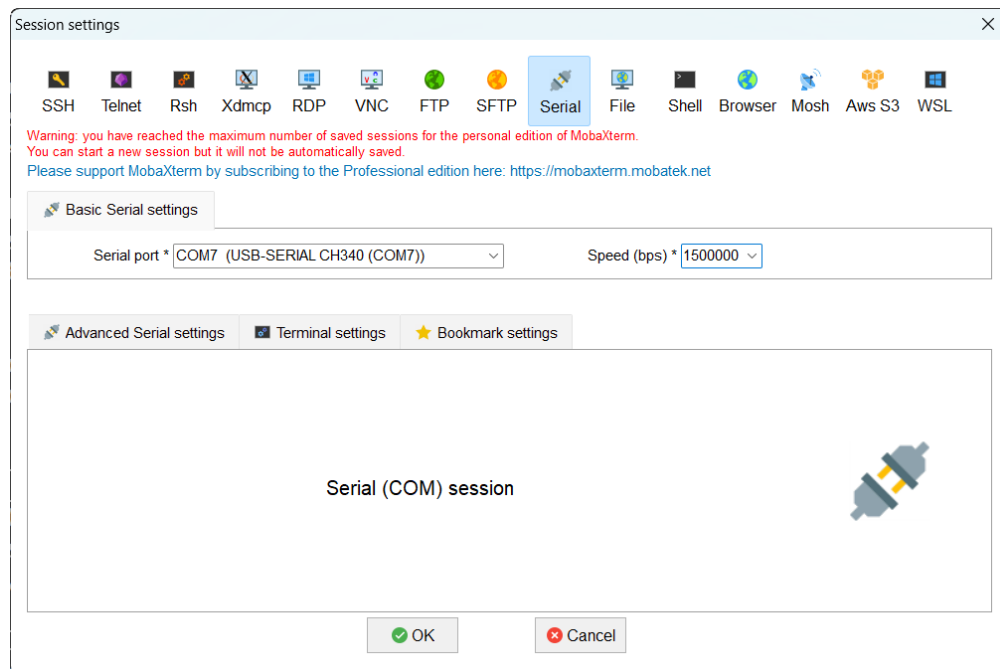
DEBUG UART位置J37，如下图所示：



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。



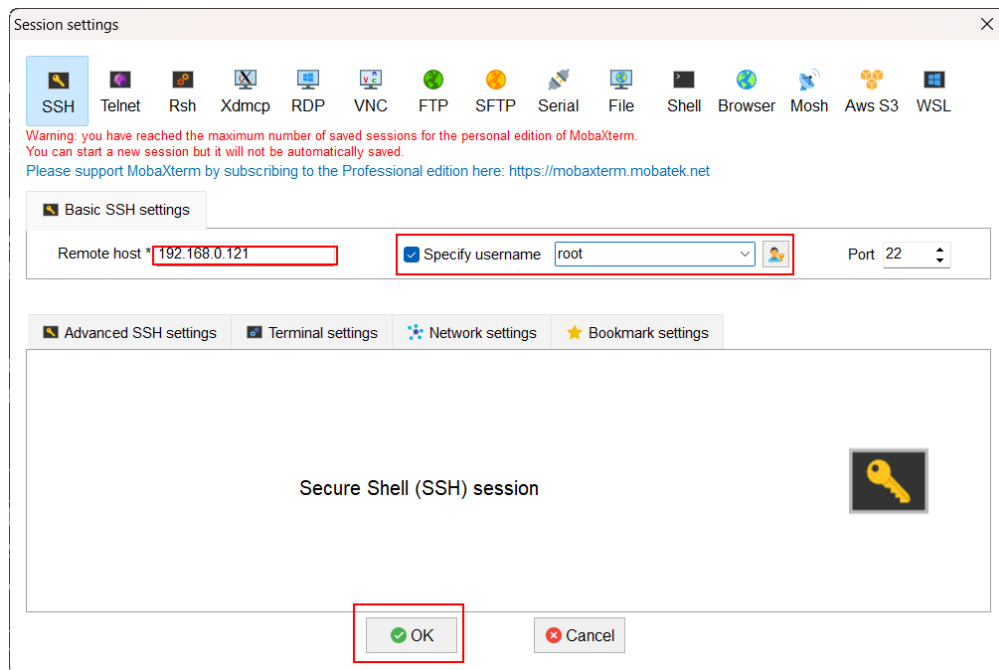
使用调试软件（MobaXterm、putty）等，以MobaXterm为例子，设置参数如下：



2.1.2. SSH登录

在知道IP的前提下，默认可以通过SSH登录进系统。

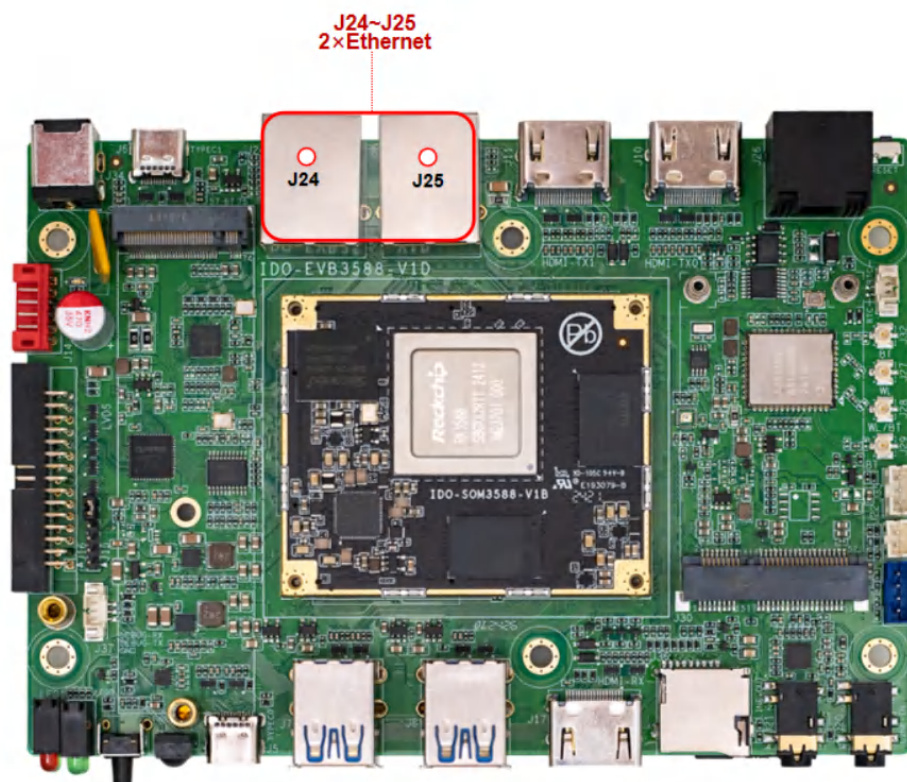
SSH登录账号密码：root/123456



2.1.3. 网络

2.1.3.1. 以太网

主板有两路千兆以太网接口位置J24,J25，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：

2.1.3.2. Ethernet动态IP设置

```
PowerShell |
1  #eth0
2  root@linaro-alip:/# ifconfig eth0
3  eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
4          inet 192.168.0.90 netmask 255.255.255.0 broadcast 192.168.0.255
5          inet6 fe80::1b83:a156:cb16:c6b0 prefixlen 64 scopeid 0x20<link>
6          ether fa:c6:ba:c7:0c:60 txqueuelen 1000 (Ethernet)
7          RX packets 141 bytes 15129 (14.7 KiB)
8          RX errors 0 dropped 4 overruns 0 frame 0
9          TX packets 89 bytes 9306 (9.0 KiB)
10         TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
11         device interrupt 121
12
13  #eth1
14  root@linaro-alip:/# ifconfig eth1
15  eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
16          inet 192.168.0.91 netmask 255.255.255.0 broadcast 192.168.0.255
17          inet6 fe80::8992:70c3:f104:e0be prefixlen 64 scopeid 0x20<link>
18          ether fe:c6:ba:c7:0c:60 txqueuelen 1000 (Ethernet)
19          RX packets 293 bytes 30522 (29.8 KiB)
20          RX errors 0 dropped 6 overruns 0 frame 0
21          TX packets 151 bytes 15448 (15.0 KiB)
22          TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
23          device interrupt 123
```

2.1.3.3. Ethernet静态IP设置

```

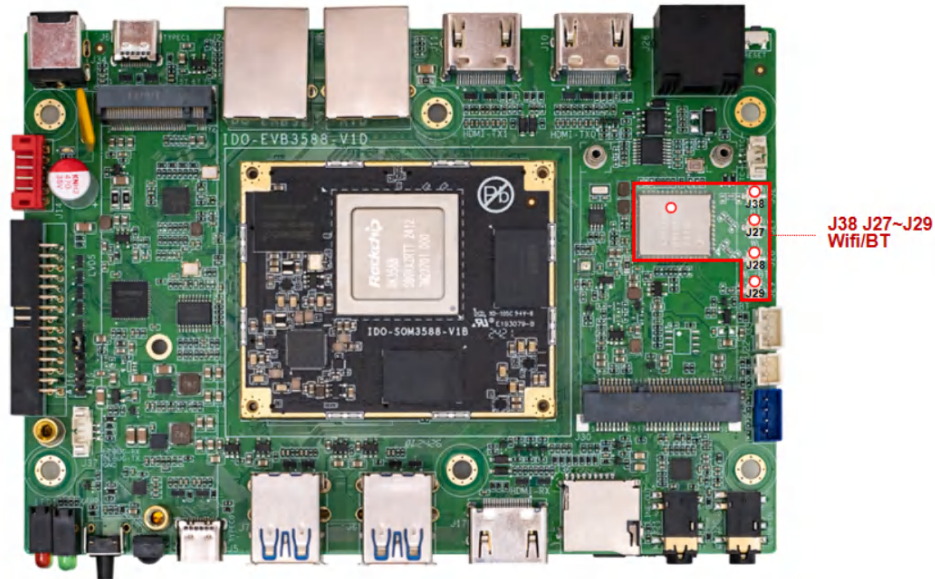
1 root@linaro-alip:/# touch /etc/netplan/01-netcfg.yaml
2 root@linaro-alip:/# vim /etc/netplan/01-netcfg.yaml
3 root@linaro-alip:/# cat /etc/netplan/01-netcfg.yaml
4 network:
5     version: 2
6     ethernet:
7         eth0:
8             dhcp4: no
9             addresses: [192.168.3.121/24]
10            nameservers:
11                addresses: [192.168.3.1, 8.8.8.8, 114.114.1
12                    14.114]
13            routes:
14                - to: 0.0.0.0/0
15                  via: 192.168.3.1
16                  metric: 100
17 #立即生效
18 root@linaro-alip:/# netplan apply
19 Cannot call openvswitch: ovsdb-server.service is not running.
20 root@linaro-alip:/# ifconfig eth0
21 eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
22     inet 192.168.3.121 netmask 255.255.255.0 broadcast 192.168.3.255
23     inet6 fe80::1b83:a156:cb16:c6b0 prefixlen 64 scopeid 0x20<link>
24     inet6 fe80::f8c6:baff:fec7:c60 prefixlen 64 scopeid 0x20<link>
25     ether fa:c6:ba:c7:0c:60 txqueuelen 1000 (Ethernet)
26     RX packets 4203 bytes 417790 (407.9 KiB)
27     RX errors 0 dropped 164 overruns 0 frame 0
28     TX packets 392 bytes 32299 (31.5 KiB)
29     TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
30     device interrupt 121

```

设备断电重启，此静态IP设置仍然生效。

2.1.4. 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：

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

查看wlan0的IP地址，确认连接成功：

```
1 root@linaro-alip:/# ifconfig wlan0
2 wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
3     inet 192.168.0.118 netmask 255.255.255.0 broadcast 192.168.0.255
4     inet6 fe80::1036:80b4:5082:7440 prefixlen 64 scopeid 0x20<link>
5     ether c0:f5:35:12:ad:a2 txqueuelen 1000 (Ethernet)
6     RX packets 170 bytes 22149 (21.6 KiB)
7     RX errors 0 dropped 5 overruns 0 frame 0
8     TX packets 24 bytes 2683 (2.6 KiB)
9     TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
10
11 root@linaro-alip:/# ping www.baidu.com -I wlan0
12 PING www.wshifen.com (103.235.47.188) from 192.168.0.118 wlan0: 56(84) bytes of data.
13 64 bytes from 103.235.47.188 (103.235.47.188): icmp_seq=1 ttl=45 time=274 ms
14 64 bytes from 103.235.47.188 (103.235.47.188): icmp_seq=2 ttl=45 time=299 ms
15 64 bytes from 103.235.47.188 (103.235.47.188): icmp_seq=3 ttl=45 time=320 ms
16 64 bytes from 103.235.47.188 (103.235.47.188): icmp_seq=4 ttl=45 time=241 ms
```

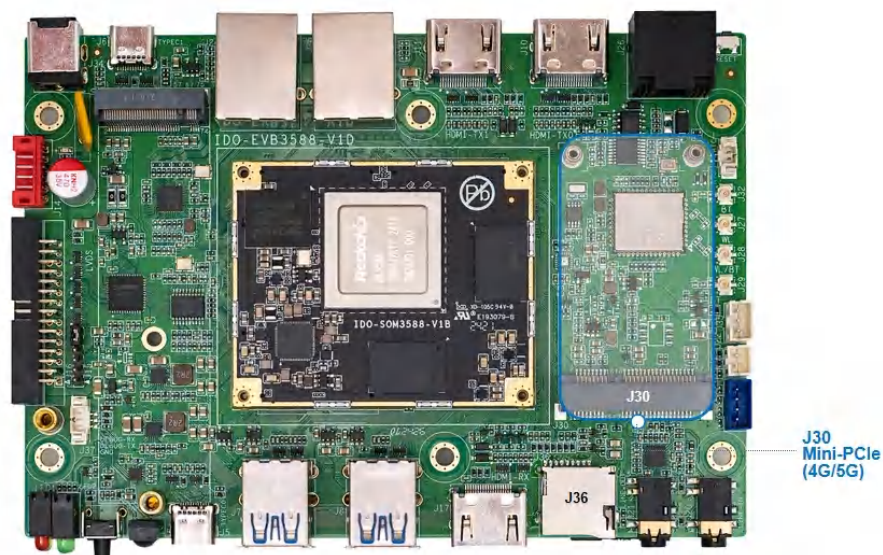
2.1.5. Bluetooth

2.1.5.1. 连接方法

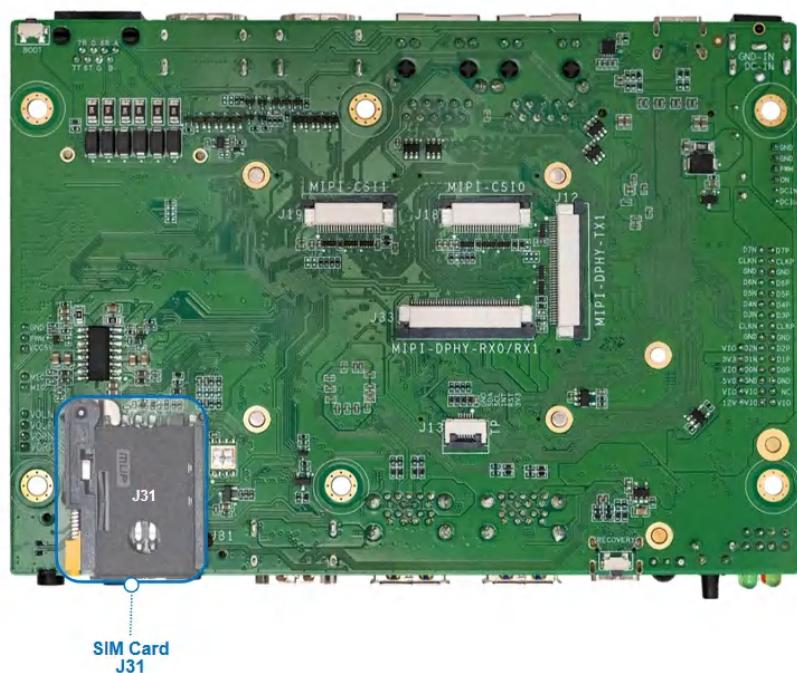
```
1 root@linaro-alip:/# bluetoothctl
2 [bluetooth]# power on
3 #扫描蓝牙
4 [bluetooth]# scan on
5 #查看蓝牙设备
6 [bluetooth]# devices
7 #信任蓝牙设备
8 [bluetooth]# trust AC:C4:BD:6C:F2:10
9 #蓝牙配对
10 [bluetooth]# pair AC:C4:BD:6C:F2:10
11 #连接蓝牙
12 [bluetooth]# connect AC:C4:BD:6C:F2:10
```

2.1.6. 4G/5G

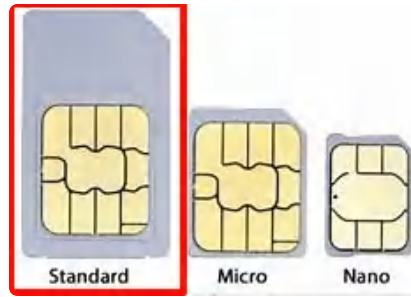
主板内置Mini PCIe 座子扩展 4G/5G模块，4G通信模块适配移远EC20、EC25等通用模组。5G通信模块适配移远RG200U-CN。模块安装位J30，如下图所示：



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J31位于主板背面，如下图所示：



使用Micro尺寸SIM卡，如下图所示：



4G使用 `ifconfig wwan0` 命令可查看到相关网络信息:

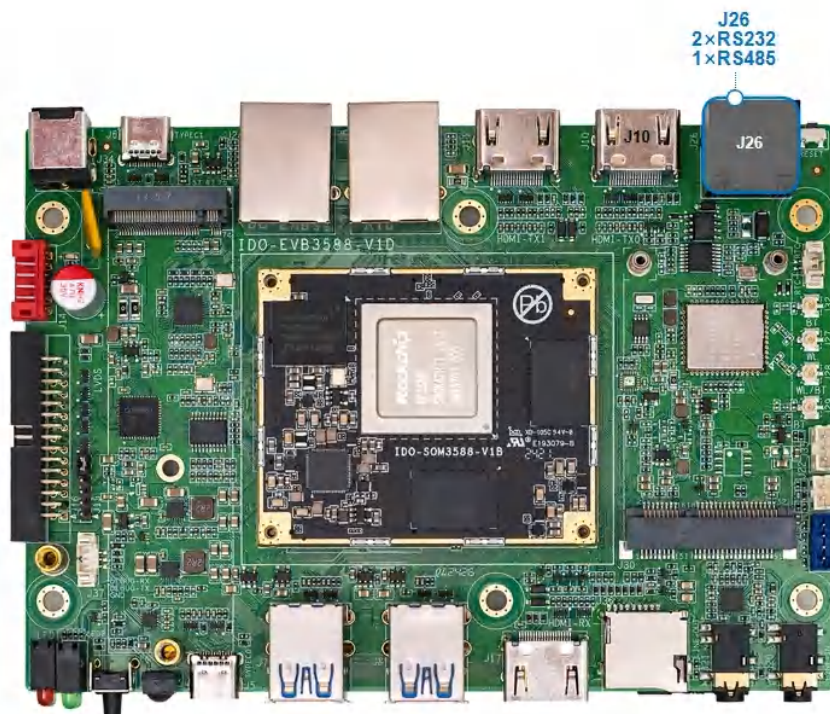
```
1  #拨号
2  root@linaro-alip:~# quectel-CM &
3
4  #查看网络
5  root@linaro-alip:~# ifconfig wwan0
6  wwan0: flags=193<UP,RUNNING,NOARP>  mtu 1500
7      inet 10.101.61.51  netmask 255.255.255.248
8      inet6 fe80::fc:f6ff:fe8d:bab6  prefixlen 64  scopeid 0x20<link>
9      ether 02:fc:f6:8d:ba:b6  txqueuelen 1000  (Ethernet)
10     RX packets 42  bytes 7013 (6.8 KiB)
11     RX errors 0  dropped 0  overruns 0  frame 0
12     TX packets 57  bytes 4608 (4.5 KiB)
13     TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
14
15  #测试通信
16  root@linaro-alip:~# ping www.baidu.com -I wwan0
```

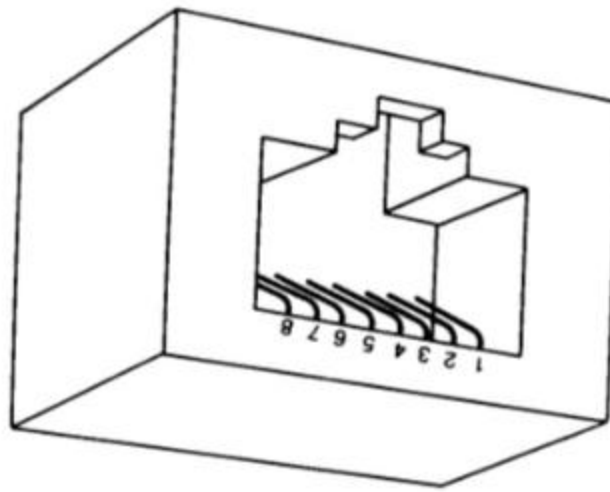
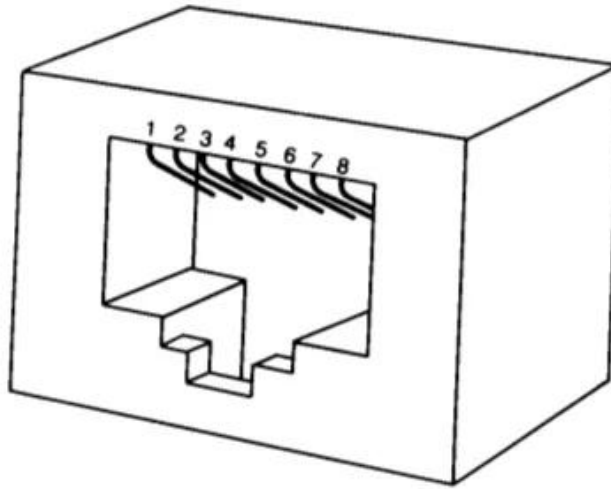
5G使用 `ifconfig usb0` 命令可查看到相关网络信息:

```
1 #拨号
2 root@linaro-alip:~# quectel-CM &
3
4 #查看网络
5 root@linaro-alip:~# ifconfig usb0
6 usb0: flags=193<UP,RUNNING,NOARP>  mtu 1500
7     inet 10.101.61.51  netmask 255.255.255.248
8     inet6 fe80::fc:f6ff:fe8d:bab6  prefixlen 64  scopeid 0x20<link>
9     ether 02:fc:f6:8d:ba:b6  txqueuelen 1000  (Ethernet)
10    RX packets 42  bytes 7013 (6.8 KiB)
11    RX errors 0  dropped 0  overruns 0  frame 0
12    TX packets 57  bytes 4608 (4.5 KiB)
13    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0
14
15 #测试通信
16 root@linaro-alip:~# ping www.baidu.com -I usb0
```

2.1.7. UART

主板一共引出3路串口（不含调试串口）共用J26端口，如下图所示：





UART设备节点列表如下：

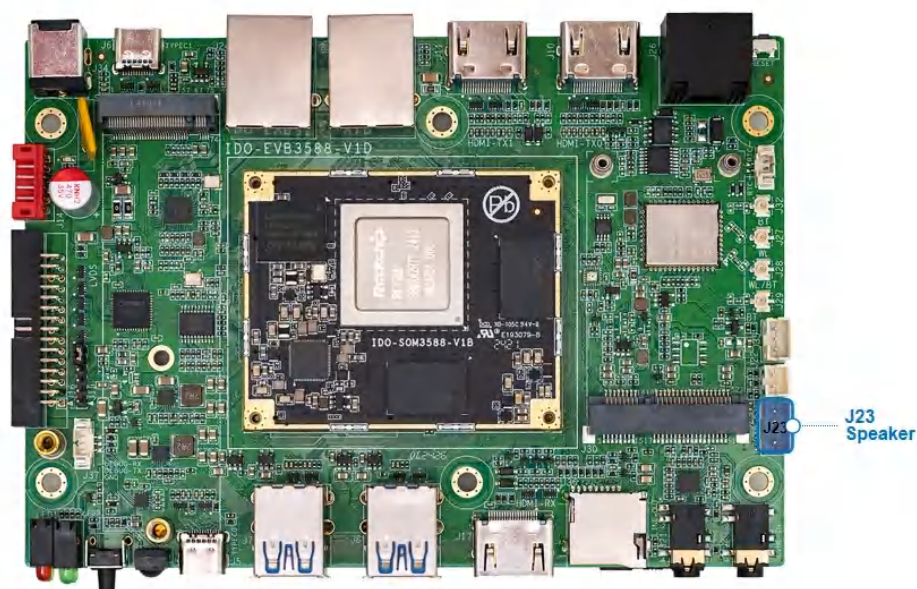
引脚序号	丝印	默认电平类型	功能	设备节点
1	A	RS485	A	/dev/ttyS4
2	B		B	
3	6R	RS232	RX	/dev/ttyS6
4	GND	GND	/	/
5	GND			
6	6T	RS232	TX	/dev/ttyS6
7	7R	RS232	RX	/dev/ttyS7
8	7T		TX	

```
1 root@linaro-alip:~# microcom -p /dev/ttyS4 -s 115200
2
```

2.1.8. 声音

2.1.8.1. 喇叭

喇叭接口为PH2.0-4P连接器J38，如下图所示：

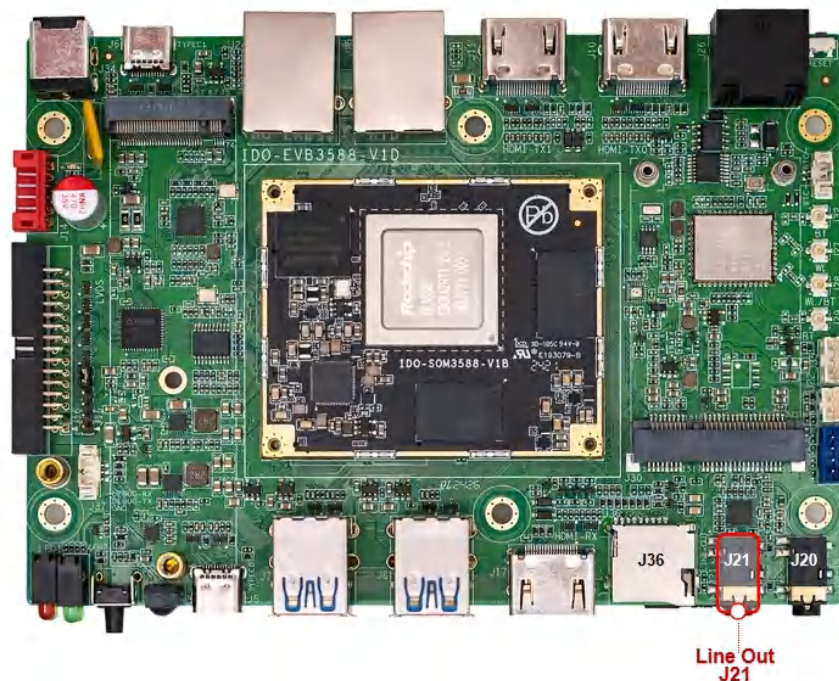


- 支持双声道，每个声道支持4Ω 3W输出

```
1  #查看声卡
2  root@linaro-alip:/# aplay -l
3
4  #音频播放到HDMI0
5  root@linaro-alip:/# aplay -D hw:0,0 xihuangni.wav
6  Playing WAVE 'xihuangni.wav' : Signed 16 bit Little Endian, Rate 44100 H
  z, Stereo
7
8  #音频播放到HDMI1
9  root@linaro-alip:/# aplay -D hw:1,0 xihuangni.wav
10 Playing WAVE 'xihuangni.wav' : Signed 16 bit Little Endian, Rate 44100 H
   z, Stereo
11
12 #音频播放到DP
13 root@linaro-alip:/# aplay -D hw:2,0 xihuangni.wav
14 Playing WAVE 'xihuangni.wav' : Signed 16 bit Little Endian, Rate 44100 H
   z, Stereo
15
16 #音频播放到声卡
17 root@linaro-alip:/# aplay -D hw:3,0 xihuangni.wav
18 Playing WAVE 'xihuangni.wav' : Signed 16 bit Little Endian, Rate 44100 H
   z, Stereo
```

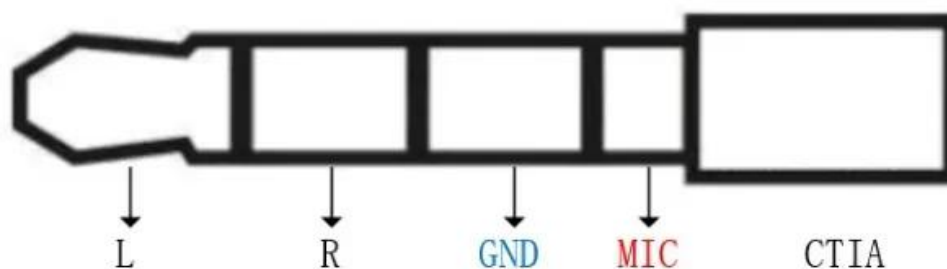
2.1.8.2. 耳机

主板支持1路3.5mm四节耳机座J21，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：



耳机录音：

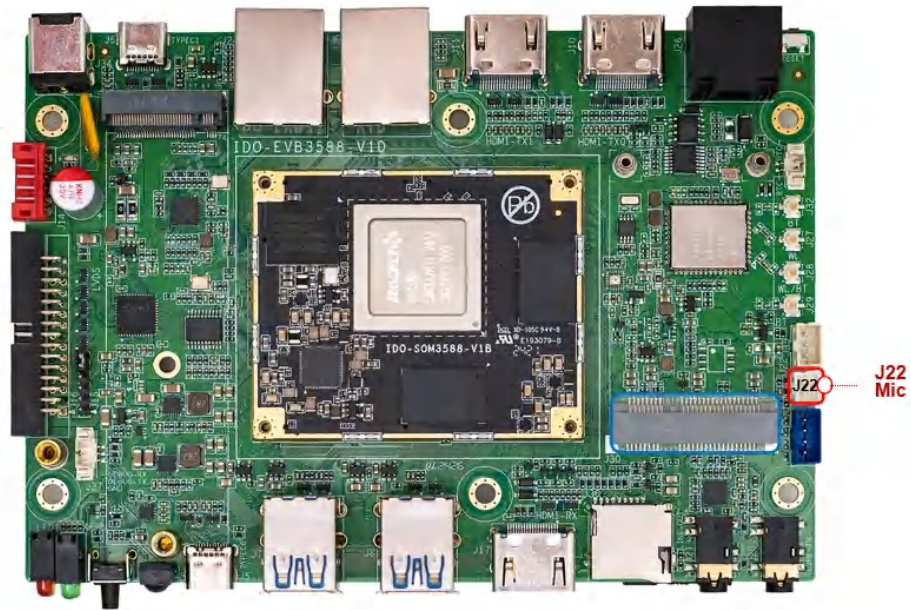
```

1  #耳机录音
2  root@linaro-alip:/# amixer -c 3 cset numid=43 on
3
4  root@linaro-alip:/# arecord -D hw:3,0 -f cd -r 44100 -c 2 test.wav
5  Recording WAVE 'test.wav' : Signed 16 bit Little Endian, Rate 44100 Hz, Stereo

```

2.1.8.3. Mic

MX1.25T-2P麦克风接口J22，如下图所示：



Mic录音：

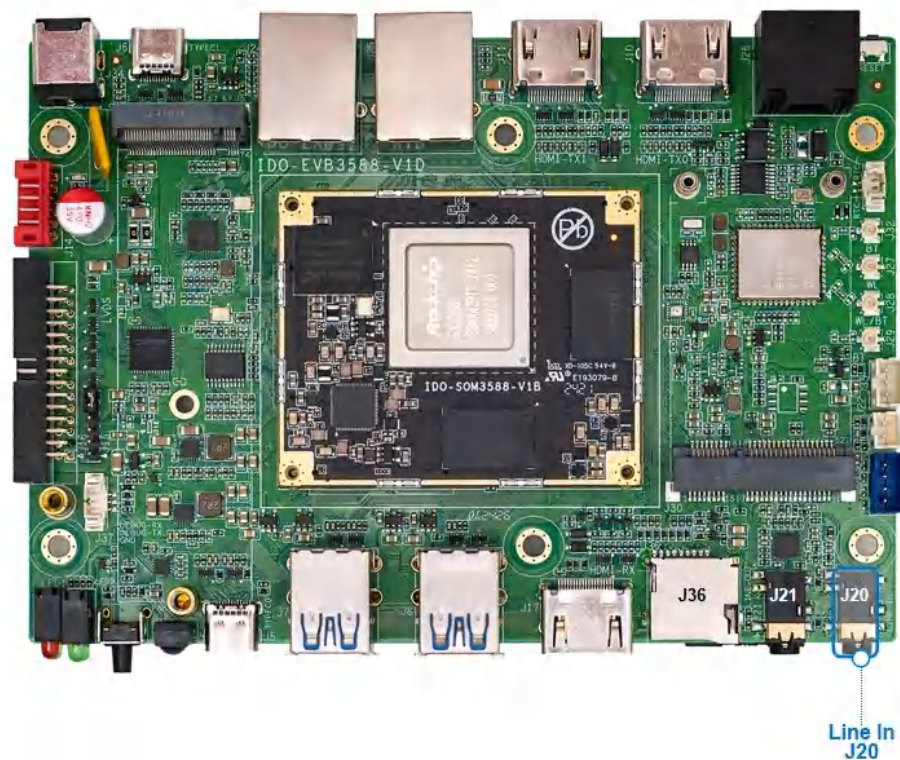


Plain Text |

```
1 #Mic录音
2 root@linaro-alip:/# arecord -D hw:3,0 -f S16_LE -r 44100 -c 2 test.wav
3 Recording WAVE 'test.wav' : Signed 16 bit Little Endian, Rate 44100 Hz, Stereo
```

2.1.8.4. Line-in

主板支持1路3.5mm 三节音频Line-in接口J20，如下图所示：

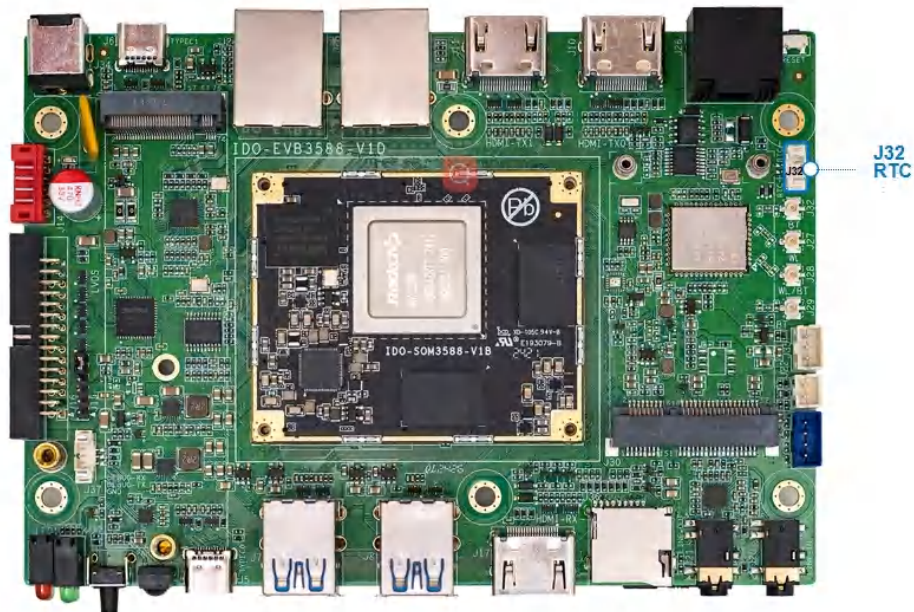


Plain Text

```
1 #切换到line in
2 root@linaro-alip:/# echo -n "mic1" > /sys/class/es8388_mic/mic_channel
3
4 #录音
5 root@linaro-alip:/# arecord -D hw:3,0 -f cd -r 44100 -c 2 test.wav
6 Recording WAVE 'test.wav' : Signed 16 bit Little Endian, Rate 44100 Hz, Stereo
```

2.1.9. 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：

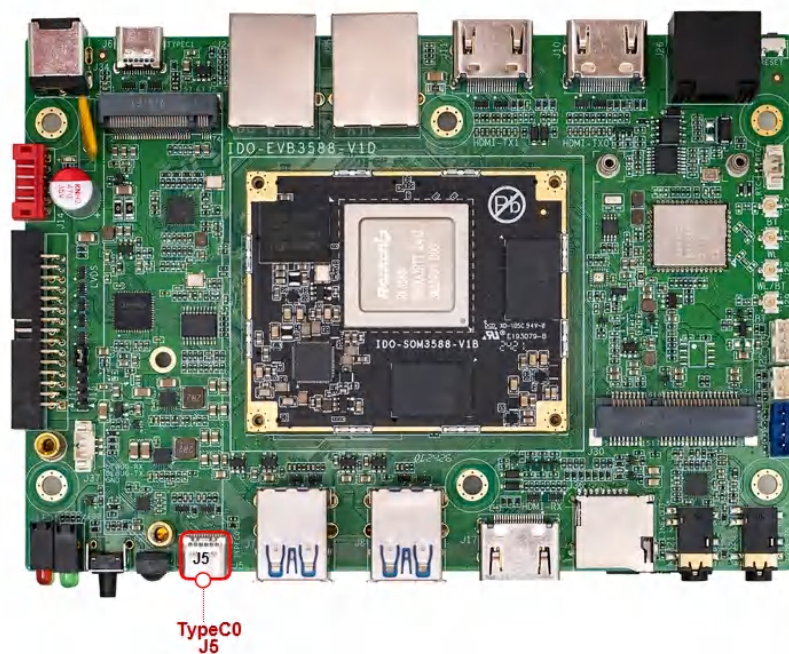
```
1  #设置时间
2  root@linaro-alip:/# date -s "2025-10-10 10:10:10"
3
4  #将rtc时钟调整为与目前的系统时钟一致
5  root@linaro-alip:/# hwclock -w
6
7  #获取硬件rtc当前时间（断电重启读取时间没有太大偏差）
8  root@linaro-alip:/# hwclock -r
9  2025-10-10 10:11:15.381196+08:00
```

2.1.10. USB接口

主板支持2个TypeC接口，支持4个USB3.0-A接口，USB对外总供电应小于3A。

2.1.10.1. TypeC0 接口

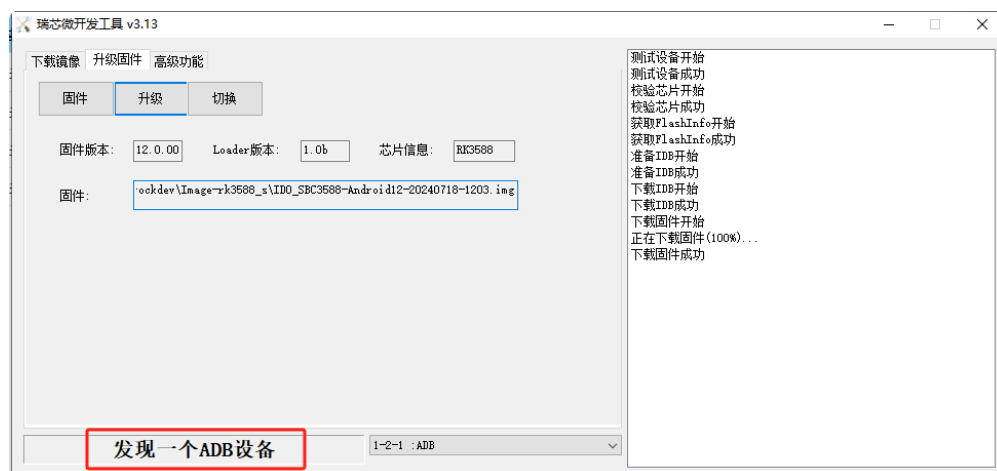
TypeC0接口（USB3 Gen1 OTG）J5，如下图所示：



- 支持Host、Device模式自动切换
- 支持DP显示输出

2.1.10.1.1. 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

2.1.10.1.2. 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

2.1.10.1.3. 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

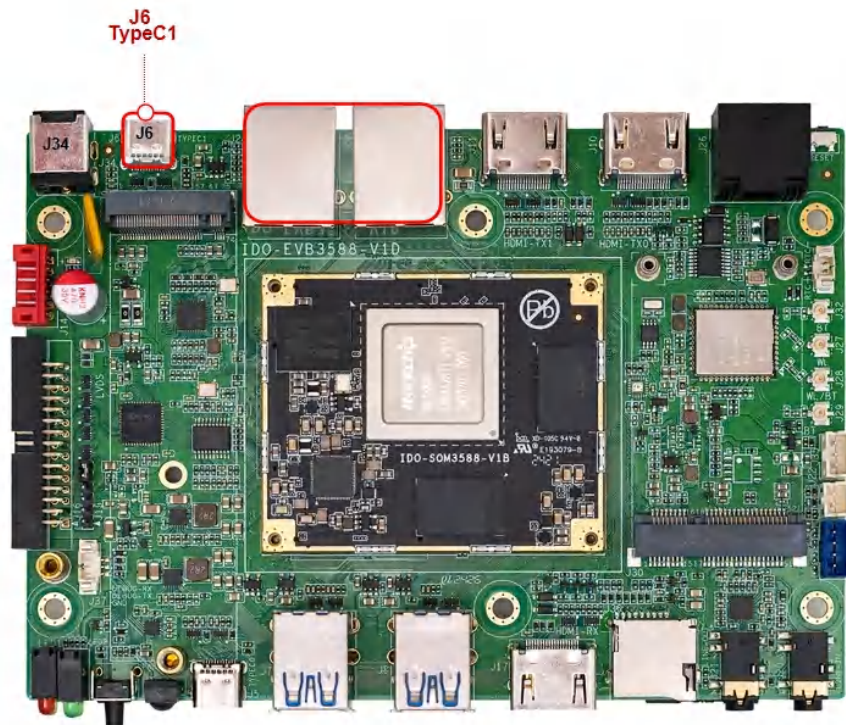


- 主板画面通过TYPE-C DP输出正常，DP声音输出正常

注意：固件默认关闭DP模式

2.1.10.2. TypeC1 接口

TypeC0接口（USB3 Gen1）J6，如下图所示：



- 支持Host模式
- 支持DP显示输出

2.1.10.2.1. 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

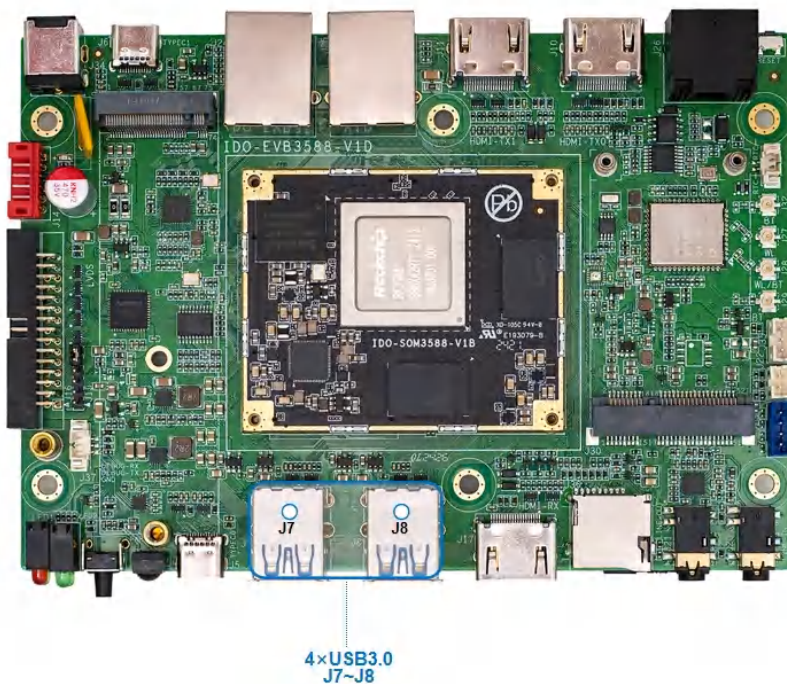
2.1.10.2.2. DP模式

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

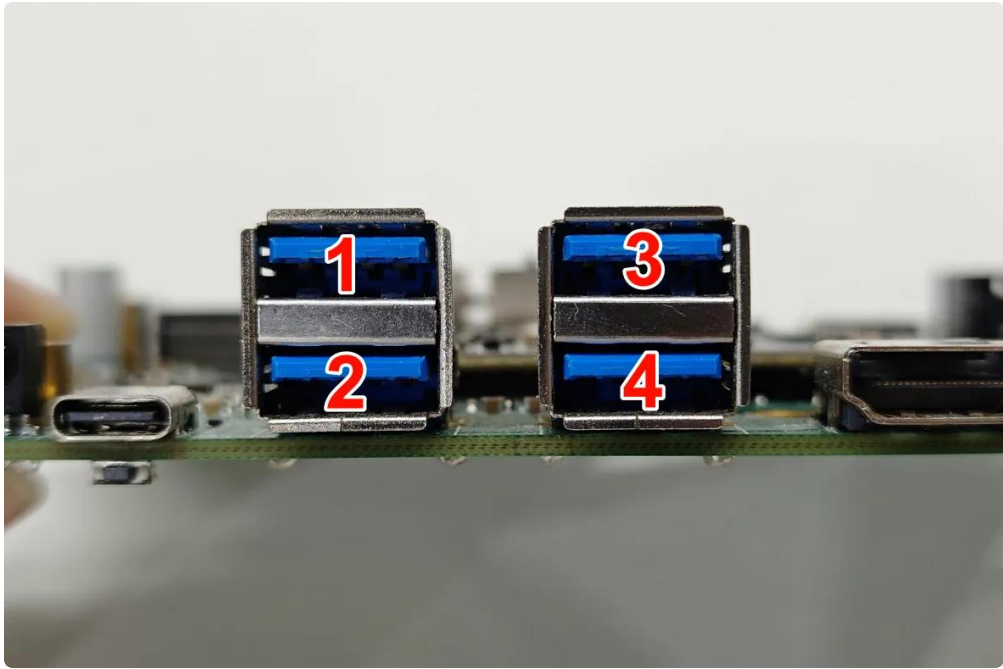


2.1.10.3. USB3.0接口

主板支持4个USB3.0接口，接口为标准的双层A口J7、J8，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：



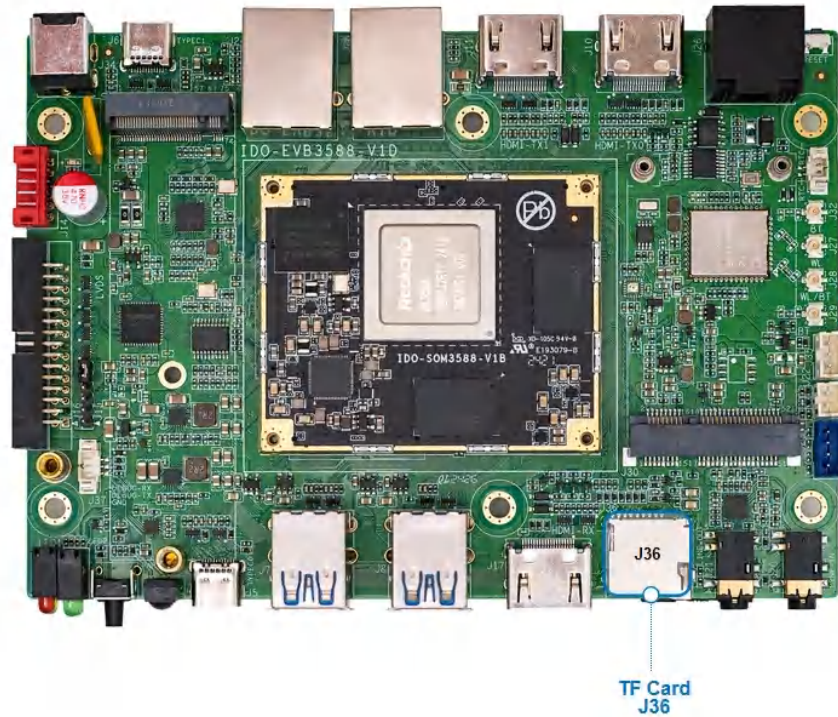
USB电源控制，如下表所示：

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。

2.1.11. TF Card

TF卡座支持SDIO3.0， 支持高速SD卡， 接口位于J36， 如下图所示：



Shell

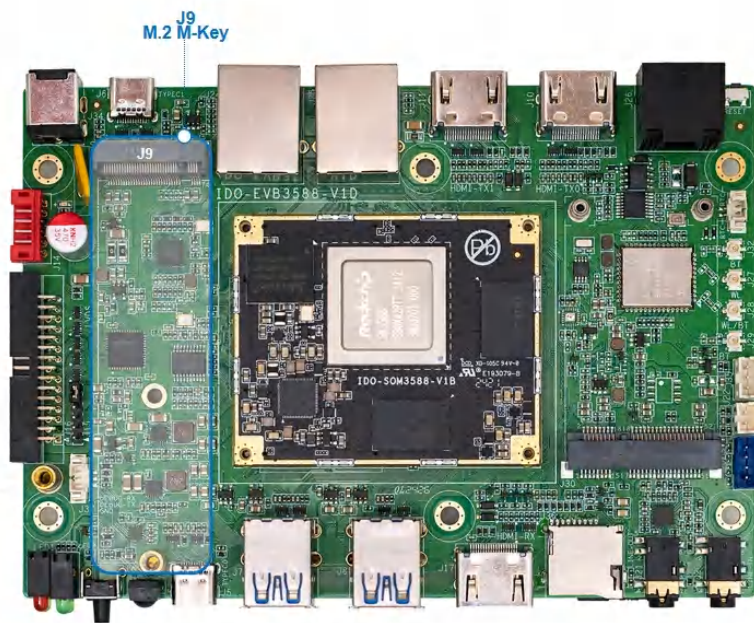
```

1 root@linaro-alip:/# fdisk -l
2 Disk /dev/ram0: 8 MiB, 8388608 bytes, 16384 sectors
3 Units: sectors of 1 * 512 = 512 bytes
4 Sector size (logical/physical): 512 bytes / 4096 bytes
5 I/O size (minimum/optimal): 4096 bytes / 4096 bytes
6 .....
7
8 Device          Boot Start      End  Sectors  Size Id Type
9 /dev/mmcblk1p1    135 3858623 3858489   1.8G  e W95 FAT16 (LBA)
10
11 root@linaro-alip:/# df -h
12 文件系统      大小  已用  可用  已用% 挂载点
13 /dev/root      53G   6.5G   45G   13% /
14 devtmpfs       4.0M   8.0K   4.0M    1% /dev
15 ...
16 /dev/mmcblk1p1 1.9G   768K   1.9G    1% /mnt/sdcard

```

2.1.12. M.2接口

主板上使用标准M.2-M-key M.2接口连接座J9，如下图所示：



支持PCIe2.1通信，适用2280尺寸NVME固态硬盘。

```

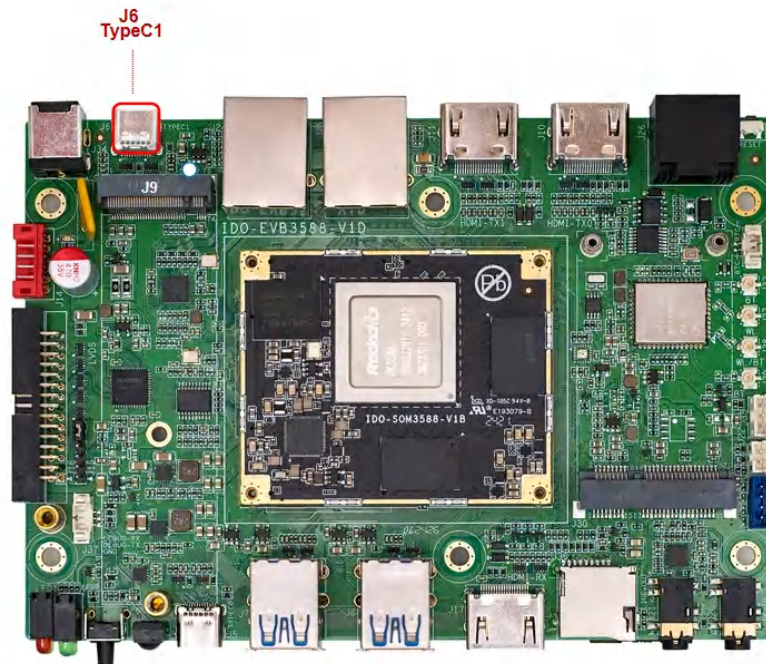
1  root@linaro-alip:/# fdisk -l
2  Disk /dev/ram0: 8 MiB, 8388608 bytes, 16384 sectors
3  Units: sectors of 1 * 512 = 512 bytes
4  Sector size (logical/physical): 512 bytes / 4096 bytes
5  I/O size (minimum/optimal): 4096 bytes / 4096 bytes
6
7
8  Disk /dev/ram1: 8 MiB, 8388608 bytes, 16384 sectors
9  Units: sectors of 1 * 512 = 512 bytes
10 Sector size (logical/physical): 512 bytes / 4096 bytes
11 I/O size (minimum/optimal): 4096 bytes / 4096 bytes
12 ...
13 Disk /dev/nvme0n1: 476.94 GiB, 512110190592 bytes, 1000215216 sectors
14 Disk model: XPG GAMMIX S11L
15 Units: sectors of 1 * 512 = 512 bytes
16 Sector size (logical/physical): 512 bytes / 512 bytes
17 I/O size (minimum/optimal): 512 bytes / 512 bytes
18 Disklabel type: dos
19 Disk identifier: 0x00000000
20
21 Device          Boot Start          End      Sectors   Size Id Type
22 /dev/nvme0n1p1      2048 1000215182 1000213135 476.9G  c W95 FAT32 (LBA)
23
24 root@linaro-alip:/# df -h
25 文件系统      大小  已用  可用  已用% 挂载点
26 /dev/root      53G   6.5G   45G   13% /
27 devtmpfs       4.0M   8.0K   4.0M    1% /dev
28 ...
29 /dev/nvme0n1p1 477G   2.1G  475G    1% /mnt/m2

```

2.1.13. LCD显示

2.1.13.1. DP

DP接口 (TYPE C1) J6, 如下图所示:



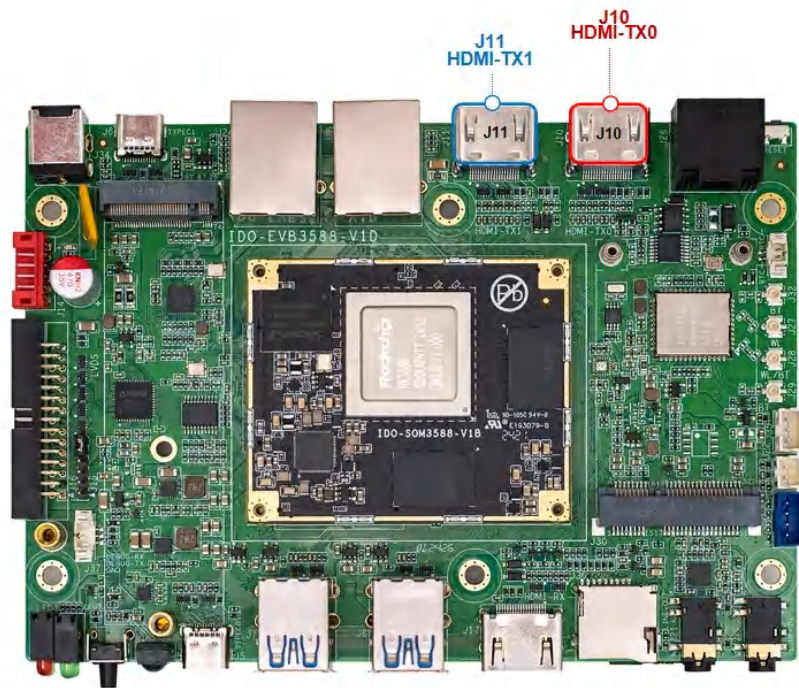
可以使用 USB-C 转 HDMI 高清线连接 HDMI 显示器输出画面，如下图所示：



- 最高支持8K@30fps输出

2.1.13.2. HDMI-TX

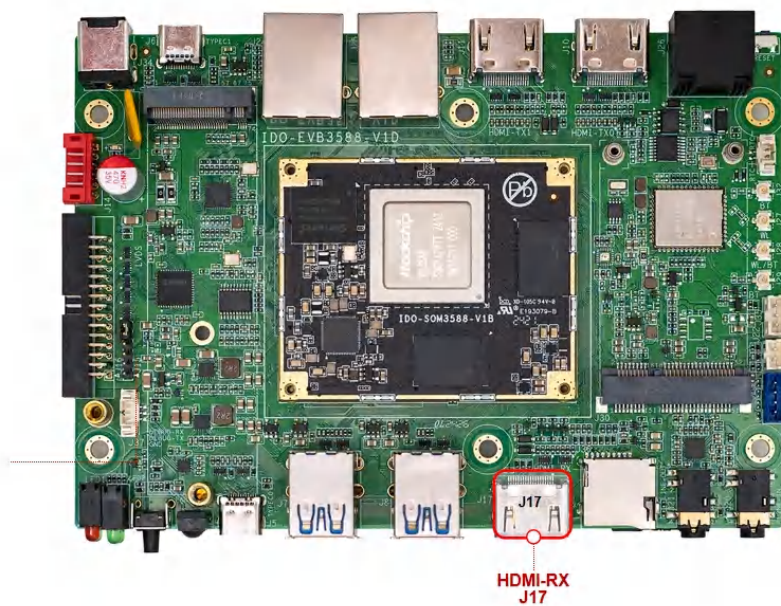
主板支持2路标准HDMI-A接口，J10、J11，如下图所示：



- 最高支持 HDMI2.1 8K@60fps 输出

2.1.13.3. HDMI-RX接口

主板支持1路标准HDMI-A接口的HDMI-RX接口J17，如下图所示：



查看输入hdm信号格式：

```

1 root@linaro-alip:/# v4l2-ctl -d /dev/video40 --query-dv-timings
2   Active width: 3840
3   Active height: 2160
4   Total width: 4400
5   Total height: 2250
6   Frame format: progressive
7   Polarities: -vsync -hsync
8   Pixelclock: 593992000 Hz (60.00 frames per second)
9   Horizontal frontporch: 176
10  Horizontal sync: 88
11  Horizontal backporch: 296
12  Vertical frontporch: 8
13  Vertical sync: 10
14  Vertical backporch: 72
15  Standards:
16  Flags:
17
18 root@linaro-alip:/# v4l2-ctl -d /dev/video40 --get-fmt-video
19 Format Video Capture Multiplanar:
20   Width/Height      : 3840/2160
21   Pixel Format       : 'NV12' (Y/CbCr 4:2:0)
22   Field              : None
23   Number of planes  : 1
24   Flags              : premultiplied-alpha, set-csc, 0x000000fc
25   Colourspace        : Rec. 709
26   Transfer Function  : Unknown (0x000000b8)
27   YCbCr/HSV Encoding: xvYCC 709
28   Quantization       : Limited Range
29   Plane 0           :
30     Bytes per Line   : 3840
31     Size Image       : 12441600

```

2.1.13.3.1. 预览

```

1 root@linaro-alip:/# gst-launch-1.0 v4l2src device=/dev/video40 ! video/x-raw, format=NV12, width=3840, height=2160, framerate=30/1 ! videoconvert ! autovideosink

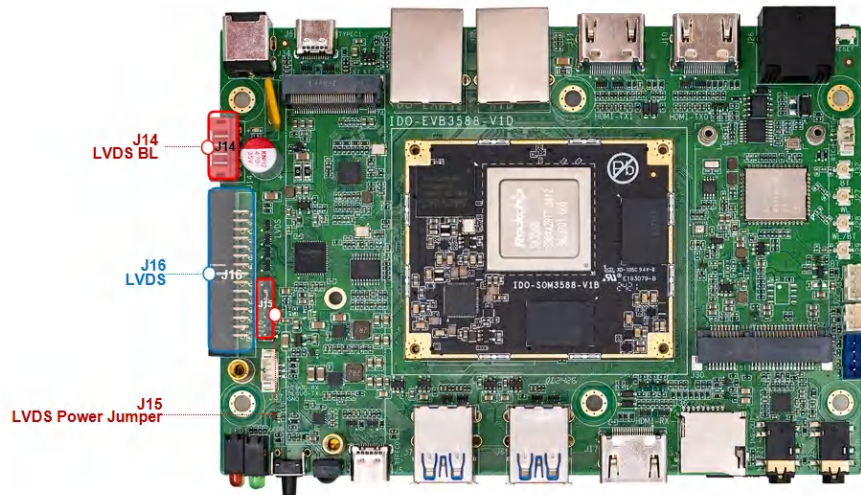
```

- HDMI2.0-RX, 支持4K@30fps

注意：hdmi in预览时需要先确认好分辨率和预览格式。

2.1.13.4. Dual LVDS屏

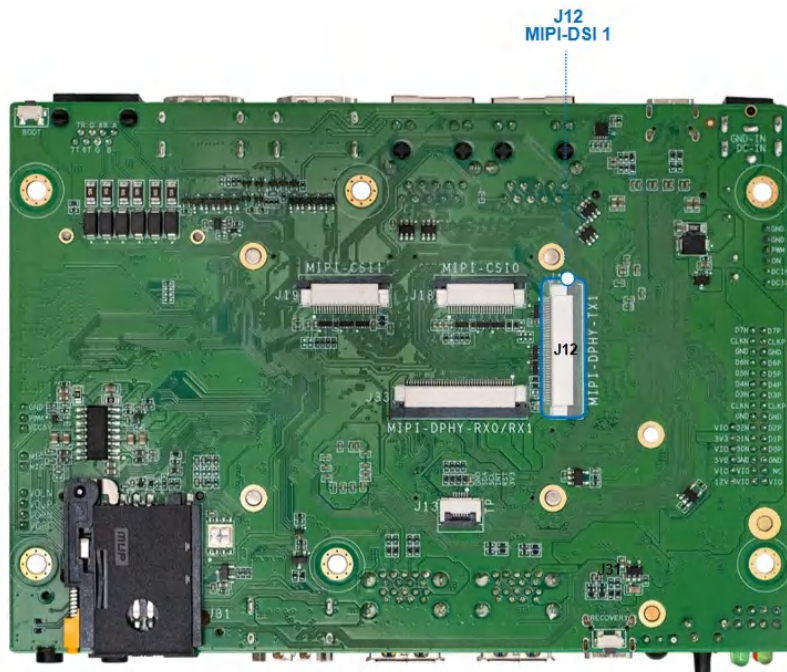
Dual LVDS接口J16，背光接 J14，屏幕供电电压选择接口J18，如下图所示：



J18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供电跳线帽接到3.3V、5V或12V，默认3.3V。

2.1.13.5. MIPI

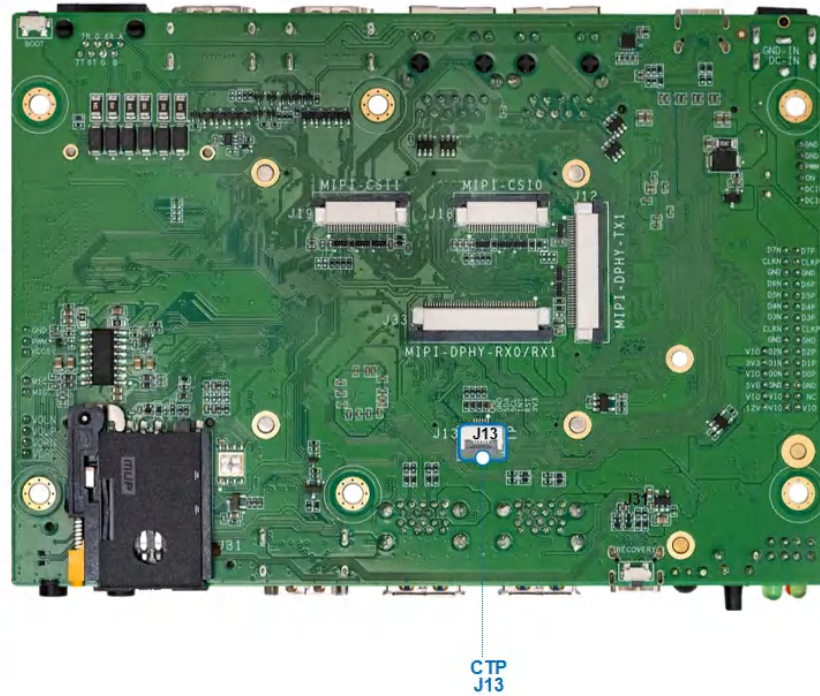
MIPI接口J12 位于主板背面，40Pin FPC 0.5mm 上接座子，如下图所示：



- MIPI供电为3.3V

2.1.14. TP接口

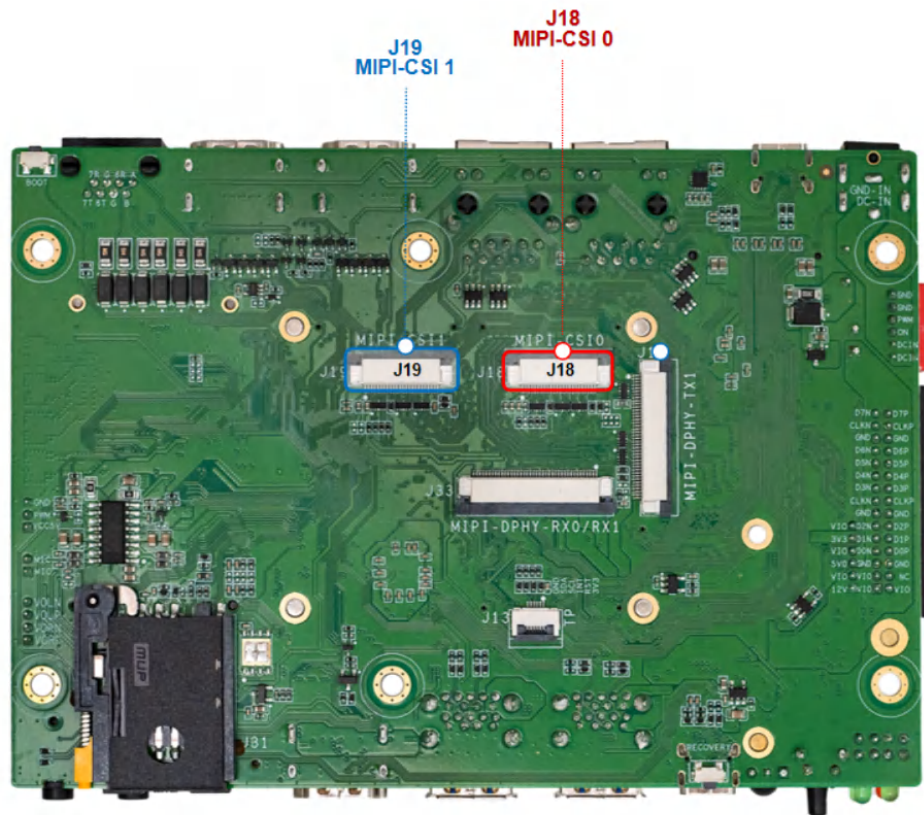
TP接口J13位于主板背面， 6Pin FPC 0.5mm座子， 如下图所示：



- 触摸 TP 接口接线方法为下接

2.1.15. MIPI Camera

MIPI Camer接口J18和J19， 如下图所示：



- MIPI Camera接口接线方法为上接

2.1.15.1. MIPI CSI0

2.1.15.1.1. 预览命令

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

2.1.15.1.2. 抓图

```
1 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
```

2.1.15.1.3. 抓视频

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

2.1.15.2. MIPI CSI1

2.1.15.2.1. 预览命令

```
1 root@linaro-alip:/# export DISPLAY=:0
2 root@linaro-alip:/# gst-launch-1.0 v4l2src device=/dev/video31 ! video/x-raw, format=NV12, width=1920, height=1080, framerate=30/1 ! videoconvert ! autovideosink
```

2.1.15.2.2. 抓图

▼ Plain Text |

```
1 root@linaro-alip:/# v4l2-ctl --verbose -d /dev/video31 --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
```

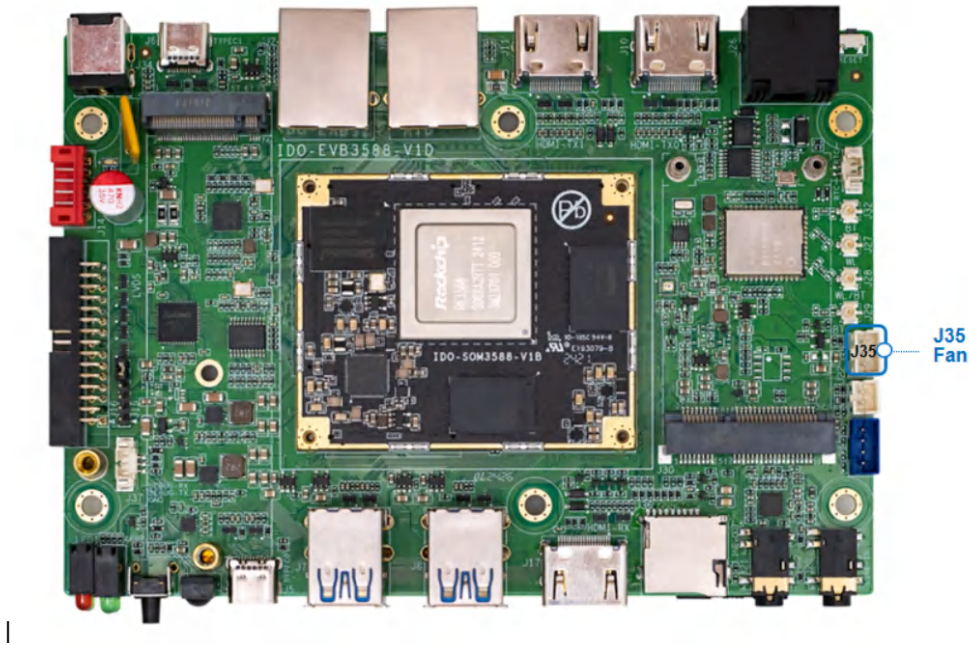
2.1.15.2.3. 抓视频

▼ Plain Text |

```
1 root@linaro-alip:/# v4l2-ctl --stream-mmap=4 -d /dev/video31 --set-fmt-video=width=1920,height=1080,pixelformat='NV12' --stream-to=./output.yuv
```

2.1.16. FAN 风扇

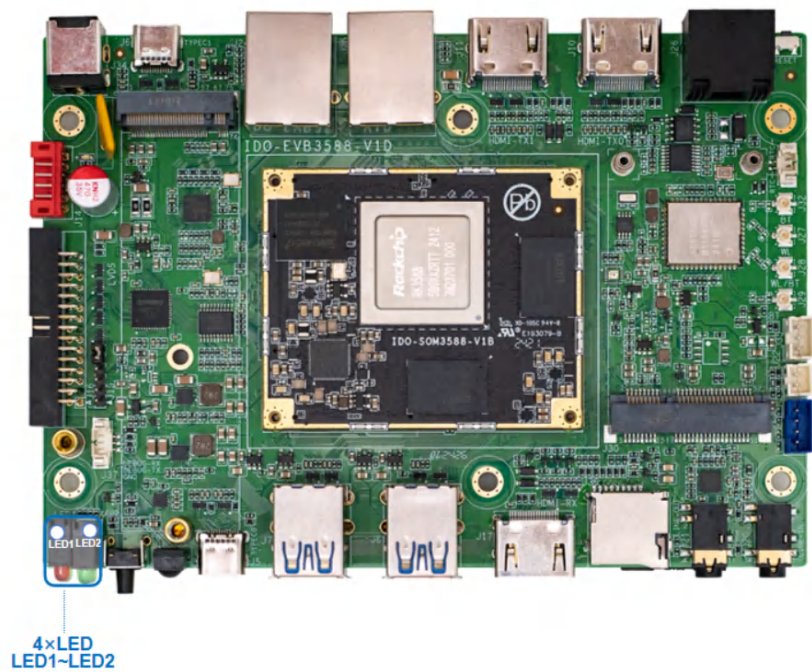
PH2 5V FAN风扇接口J35，如下图所示：



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

2.1.17. 用户LED

主板集成1个用户自定义控制LED2，如下图所示：



其中LED2的上层对应节点为 `/sys/class/leds/led2/brightness`

LED控制方法，如下表所示：

序号	操作	控制命令
LED2	打开	<code>echo 1 > /sys/class/leds/led2/brightness</code>
	关闭	<code>echo 0 > /sys/class/leds/led2/brightness</code>

2.1.18. IR

主板配置了一路红外接口，支持NEC编码遥控器，默认适配的遥控器型号为HTR-A07。



HTR-A07的键值表如下：

编号	按键	键值		编号	按键	键值
1	电源	KEY_POWER		21	1	KEY_1
2	TV	KEY_SCREEN		22	2	KEY_2
3	橙色	KEY_F1		23	3	KEY_3
4	绿色	KEY_F2		24	4	KEY_4
5	黄色	KEY_F3		25	5	KEY_5
6	紫色	KEY_F4		26	6	KEY_6
7	音量+	KEY_VOLUMEUP		27	7	KEY_7
8	音量-	KEY_VOLUMEDOWN		28	8	KEY_8
9	屏显	KEY_DISPLAY_OFF		29	9	KEY_9
10	静音	KEY_MUTE		30	TVNOW	KEY_DOT
11	上一节目	KEY_VIDEO_PREV		31	0	KEY_0
12	下一节目	KEY_VIDEO_NEXT		32	截屏	KEY_PRINT

13	上	KEY_UP		33		
14	左	KEY_LEFT		34		
15	下	KEY_DOWN		35		
16	右	KEY_RIGHT		36		
17	确认	KEY_ENTER		37		
18	返回	KEY_BACK		38		
19	主页	KEY_HOME		39		
20	菜单	KEY_MENU		40		

注：使用以下命令可以从调试串口打印按键的键值。

▼

Bash

```
1 echo 1 > /sys/module/rockchip_pwm_remotectl/parameters/code_print
```

使用evtest工具可以查看按键上报键值：

```
1 root@rk3588:/# evtest
2 No device specified, trying to scan all of /dev/input/event*
3 Available devices:
4 /dev/input/event0:      febd0030.pwm
5 /dev/input/event1:      rockchip-hdmi0 rockchip-hdmi0
6 /dev/input/event2:      rockchip-hdmi1 rockchip-hdmi1
7 /dev/input/event3:      rockchip,dp0 rockchip,dp0
8 /dev/input/event4:      rk805 pwrkey
9 /dev/input/event5:      adc-keys
10 /dev/input/event6:      headset-keys
11 /dev/input/event7:      rockchip-es8388 Headset
12 /dev/input/event8:      USB OPTICAL MOUSE
13 Select the device event number [0-8]: 0
14 Input driver version is 1.0.1
15 Input device ID: bus 0x19 vendor 0x524b product 0x6 version 0x100
16 Input device name: "febd0030.pwm"
17 Supported events:
18   Event type 0 (EV_SYN)
19   Event type 1 (EV_KEY)
20     Event code 2 (KEY_1)
21     Event code 3 (KEY_2)
22     Event code 4 (KEY_3)
23     Event code 5 (KEY_4)
24     Event code 6 (KEY_5)
25     Event code 7 (KEY_6)
26     Event code 8 (KEY_7)
27     Event code 9 (KEY_8)
28     Event code 10 (KEY_9)
29     Event code 11 (KEY_0)
30     Event code 14 (KEY_BACKSPACE)
31     Event code 28 (KEY_ENTER)
32     Event code 52 (KEY_DOT)
33     Event code 61 (KEY_F3)
34     Event code 102 (KEY_HOME)
35     Event code 103 (KEY_UP)
36     Event code 104 (KEY_PAGEUP)
37     Event code 105 (KEY_LEFT)
38     Event code 106 (KEY_RIGHT)
39     Event code 108 (KEY_DOWN)
40     Event code 109 (KEY_PAGEDOWN)
41     Event code 113 (KEY_MUTE)
42     Event code 114 (KEY_VOLUMEDOWN)
43     Event code 115 (KEY_VOLUMEUP)
44     Event code 116 (KEY_POWER)
45     Event code 139 (KEY_MENU)
```

```

46      Event code 143 (KEY_WAKEUP)
47      Event code 158 (KEY_BACK)
48      Event code 183 (KEY_F13)
49      Event code 184 (KEY_F14)
50      Event code 185 (KEY_F15)
51      Event code 186 (KEY_F16)
52      Event code 217 (KEY_SEARCH)
53      Event code 232 (KEY_REPLY)
54      Event code 241 (KEY_VIDEO_NEXT)
55      Event code 242 (KEY_VIDEO_PREV)
56      Event code 245 (KEY_DISPLAY_OFF)
57      Event code 248 (KEY_MICMUTE)
58      Event code 338 (?)
59      Event code 373 (KEY_MODE)
60      Event code 375 (KEY_SCREEN)
61      Event code 388 (KEY_TEXT)
62      Event code 400 (KEY_YELLOW)
63      Event code 401 (KEY_BLUE)
64      Event code 402 (KEY_CHANNELUP)
65  Properties:
66  Testing ... (interrupt to exit)
67  [ 4237.196132] USERCODE=0x1818
68  [ 4237.223070] RMC_GETDATA=ff
69  Event: time 1682532011.417156, type 1 (EV_KEY), code 2 (KEY_1), value 1
70  Event: time 1682532011.417156, ----- SYN_REPORT -----
71  Event: time 1682532011.475441, type 1 (EV_KEY), code 2 (KEY_1), value 0
72  Event: time 1682532011.475441, ----- SYN_REPORT -----
73  [ 4237.866797] USERCODE=0x1818
74  [ 4237.893797] RMC_GETDATA=fe
75  Event: time 1682532012.087855, type 1 (EV_KEY), code 3 (KEY_2), value 1
76  Event: time 1682532012.087855, ----- SYN_REPORT -----
77  Event: time 1682532012.178638, type 1 (EV_KEY), code 3 (KEY_2), value 0
78  Event: time 1682532012.178638, ----- SYN_REPORT -----
79  [ 4238.370395] USERCODE=0x1818
80  [ 4238.397454] RMC_GETDATA=fd
81  Event: time 1682532012.591469, type 1 (EV_KEY), code 4 (KEY_3), value 1
82  Event: time 1682532012.591469, ----- SYN_REPORT -----
83  Event: time 1682532012.682093, type 1 (EV_KEY), code 4 (KEY_3), value 0
84  Event: time 1682532012.682093, ----- SYN_REPORT -----
85  [ 4239.209857] USERCODE=0x1818
86  [ 4239.236861] RMC_GETDATA=fc
87  Event: time 1682532013.430863, type 1 (EV_KEY), code 5 (KEY_4), value 1
88  Event: time 1682532013.430863, ----- SYN_REPORT -----
89  Event: time 1682532013.521980, type 1 (EV_KEY), code 5 (KEY_4), value 0
90  Event: time 1682532013.521980, ----- SYN_REPORT -----
91

```

2.2. GPU

执行glmark2-es2 开启gpu测试窗口：

```

1 root@linaro-alip:/# glmark2-es2
2 arm_release_ver: g24p0-00eac0, rk_so_ver: 6
3 =====
4     glmark2 2023.01
5     =====
6     OpenGL Information
7     GL_VENDOR:      ARM
8     GL_RENDERER:    Mali-G610
9     GL_VERSION:     OpenGL ES 3.2 v1.g24p0-00eac0.6b39a7384565dd0742f84d5a
10    b1cdf9ef
11    Surface Config:  buf=32 r=8 g=8 b=8 a=8 depth=24 stencil=0 samples=0
12    Surface Size:    800x600 windowed
13    =====
14 [build] use-vbo=false: FPS: 1039 FrameTime: 0.963 ms
15 [build] use-vbo=true: FPS: 1149 FrameTime: 0.871 ms
16 [texture] texture-filter=nearest: FPS: 1179 FrameTime: 0.848 ms
17 [texture] texture-filter=linear: FPS: 1149 FrameTime: 0.871 ms
18 [texture] texture-filter=mipmap: FPS: 1173 FrameTime: 0.853 ms
19 [shading] shading=gouraud: FPS: 1060 FrameTime: 0.944 ms
20 [shading] shading=blinn-phong-inf: FPS: 1057 FrameTime: 0.946 ms
21 [shading] shading=phong: FPS: 1032 FrameTime: 0.970 ms
22 [shading] shading=cel: FPS: 1054 FrameTime: 0.949 ms
23 [bump] bump-render=high-poly: FPS: 863 FrameTime: 1.160 ms
24 [bump] bump-render=normals: FPS: 1205 FrameTime: 0.830 ms
25 [bump] bump-render=height: FPS: 1177 FrameTime: 0.850 ms
26 [effect2d] kernel=0,1,0;1,-4,1;0,1,0;: FPS: 1206 FrameTime: 0.830 ms
27 [effect2d] kernel=1,1,1,1,1;1,1,1,1,1;1,1,1,1,1;: FPS: 964 FrameTime: 1.03
28 8 ms
29 [pulsar] light=false:quads=5:texture=false: FPS: 1133 FrameTime: 0.883 ms
30 [desktop] blur-radius=5:effect=blur:passes=1:separable=true:windows=4: FPS
31 : 779 FrameTime: 1.285 ms
32 [desktop] effect=shadow:windows=4: FPS: 970 FrameTime: 1.032 ms
33 [buffer] columns=200:interleave=false:update-dispersion=0.9:update-fractio
34 n=0.5:update-method=map: FPS: 467 FrameTime: 2.142 ms
35 [buffer] columns=200:interleave=false:update-dispersion=0.9:update-fractio
36 n=0.5:update-method=subdata: FPS: 448 FrameTime: 2.236 ms
37 [buffer] columns=200:interleave=true:update-dispersion=0.9:update-fraction
38 =0.5:update-method=map: FPS: 531 FrameTime: 1.885 ms
39 [ideas] speed=duration: FPS: 964 FrameTime: 1.038 ms
40 [jellyfish] <default>: FPS: 1011 FrameTime: 0.990 ms
41 [terrain] <default>: FPS: 263 FrameTime: 3.812 ms
42 [shadow] <default>: FPS: 1012 FrameTime: 0.988 ms
43 [refract] <default>: FPS: 390 FrameTime: 2.567 ms
44 [conditionals] fragment-steps=0:vertex-steps=0: FPS: 1198 FrameTime: 0.835
45 ms

```

```

39 [conditionals] fragment-steps=5:vertex-steps=0: FPS: 1179 FrameTime: 0.849
40 ms
41 [conditionals] fragment-steps=0:vertex-steps=5: FPS: 1144 FrameTime: 0.875
42 ms
43 [function] fragment-complexity=low:fragment-steps=5: FPS: 1126 FrameTime:
44 0.888 ms
45 [function] fragment-complexity=medium:fragment-steps=5: FPS: 1136 FrameTim
46 e: 0.881 ms
47 [loop] fragment-loop=false:fragment-steps=5:vertex-steps=5: FPS: 1173 Fram
48 eTime: 0.853 ms
[loop] fragment-steps=5:fragment-uniform=false:vertex-steps=5: FPS: 1122 F
rameTime: 0.892 ms
[loop] fragment-steps=5:fragment-uniform=true:vertex-steps=5: FPS: 1120 Fr
ameTime: 0.893 ms
=====
                                glmark2 Score: 983
=====

```

2.3. NPU

将rknpu2.tar.gz压缩包拷贝板子上，使用yolov5进行rknn模型转换测试：

2.3.1. 编译rknn_yolov5_demo

```

1 root@linaro-alip:/# tar -xvf rknpu2.tar.gz
2 root@linaro-alip:/# cd rknpu2/examples/rknn_yolov5_demo/
3
4 root@linaro-alip:/rknpu2/examples/rknn_yolov5_demo# chmod a+x build-linux_
RK3588.sh
5 root@linaro-alip:/rknpu2/examples/rknn_yolov5_demo# ./build-linux_RK3588.s
h -t rk3588 -a aarch64 -b Release
6 -- The C compiler identification is GNU 11.4.0
7 -- The CXX compiler identification is GNU 11.4.0
8 -- Detecting C compiler ABI info
9 -- Detecting C compiler ABI info - done
10 -- Check for working C compiler: /usr/bin/aarch64-linux-gnu-gcc - skipped
11 -- Detecting C compile features
12 -- Detecting C compile features - done
13 -- Detecting CXX compiler ABI info
14 -- Detecting CXX compiler ABI info - done
15 -- Check for working CXX compiler: /usr/bin/aarch64-linux-gnu-g++ - skippe
d
16 -- Detecting CXX compile features
17 -- Detecting CXX compile features - done
18 -- Found OpenCV: /rknpu2/examples/3rdparty/opencv/opencv-linux-aarch64 (fo
und version "3.4.5")
19 -- Configuring done
20 -- Generating done
21 -- Build files have been written to: /rknpu2/examples/rknn_yolov5_demo/bui
ld/build_linux_aarch64
22 ▾ [ 10%] Building CXX object CMakeFiles/rknn_yolov5_demo.dir/src/preprocess.
cc.o
23 ▾ [ 30%] Building CXX object CMakeFiles/rknn_yolov5_demo.dir/src/postproces
s.cc.o
24 ▾ [ 30%] Building CXX object CMakeFiles/rknn_yolov5_demo.dir/src/main.cc.o
25 ▾ [ 40%] Building CXX object CMakeFiles/rknn_yolov5_video_demo.dir/src/main_
video.cc.o
26 In file included from /rknpu2/examples/rknn_yolov5_demo/src/main_video.cc:
32:
27 ▾ /rknpu2/examples/rknn_yolov5_demo/Utils/mpp_decoder.h:58:18: warning: conv
verting to non-pointer type 'pthread_t' {aka 'long unsigned int'} from NUL
L [-Wconversion-null]
28     58 |         pthread_t th=NULL;
29         |               ^~~~
30 ▾ [ 50%] Building CXX object CMakeFiles/rknn_yolov5_video_demo.dir/src/postp
rocess.cc.o
31 ▾ [ 60%] Building CXX object CMakeFiles/rknn_yolov5_video_demo.dir/Utils/mpp
_decoder.cpp.o
32

```

```

33 In file included from /rknpu2/examples/rknn_yolov5_demo/utils/mpp_decoder.
    cpp:4:
    /rknpu2/examples/rknn_yolov5_demo/utils/mpp_decoder.h:58:18: warning: conv
34 erting to non-pointer type 'pthread_t' {aka 'long unsigned int'} from NUL
    L [-Wconversion-null]
35     58 |         pthread_t th=NULL;
36         |         ~~~~~
37 [ 70%] Building CXX object CMakeFiles/rknn_yolov5_video_demo.dir/utils/mpp
    _encoder.cpp.o
38 [ 80%] Building CXX object CMakeFiles/rknn_yolov5_video_demo.dir/utils/dra
    wing.cpp.o
39 [ 90%] Linking CXX executable rknn_yolov5_video_demo
40 [ 90%] Built target rknn_yolov5_video_demo
41 [100%] Linking CXX executable rknn_yolov5_demo
42 [100%] Built target rknn_yolov5_demo
43 Consolidate compiler generated dependencies of target rknn_yolov5_demo
44 [ 40%] Built target rknn_yolov5_demo
45 Consolidate compiler generated dependencies of target rknn_yolov5_video_de
    mo
46 [100%] Built target rknn_yolov5_video_demo
47 Install the project...
48 -- Install configuration: ''
49 -- Installing: /rknpu2/examples/rknn_yolov5_demo/install/rknn_yolov5_demo_
    Linux/./rknn_yolov5_demo
50 -- Installing: /rknpu2/examples/rknn_yolov5_demo/install/rknn_yolov5_demo_
    Linux/lib/librknnrt.so
51 -- Installing: /rknpu2/examples/rknn_yolov5_demo/install/rknn_yolov5_demo_
    Linux/lib/librga.so
52 -- Installing: /rknpu2/examples/rknn_yolov5_demo/install/rknn_yolov5_demo_
    Linux/./model
53 -- Installing: /rknpu2/examples/rknn_yolov5_demo/install/rknn_yolov5_demo_
    Linux/./model/RK3588
54 -- Installing: /rknpu2/examples/rknn_yolov5_demo/install/rknn_yolov5_demo_
    Linux/./model/RK3588/yolov5s-640-640.rknn
55 -- Installing: /rknpu2/examples/rknn_yolov5_demo/install/rknn_yolov5_demo_
    Linux/./model/RK3562
56 -- Installing: /rknpu2/examples/rknn_yolov5_demo/install/rknn_yolov5_demo_
    Linux/./model/RK3562/yolov5s-640-640.rknn
57 -- Installing: /rknpu2/examples/rknn_yolov5_demo/install/rknn_yolov5_demo_
    Linux/./model/coco_80_labels_list.txt
58 -- Installing: /rknpu2/examples/rknn_yolov5_demo/install/rknn_yolov5_demo_
    Linux/./model/bus.jpg
59 -- Installing: /rknpu2/examples/rknn_yolov5_demo/install/rknn_yolov5_demo_
    Linux/./model/RK3566_RK3568
60 -- Installing: /rknpu2/examples/rknn_yolov5_demo/install/rknn_yolov5_demo_
    Linux/./model/RK3566_RK3568/yolov5s-640-640.rknn
61 -- Installing: /rknpu2/examples/rknn_yolov5_demo/install/rknn_yolov5_demo_
    Linux/./rknn_yolov5_video_demo

```

```

62  -- Installing: /rknpu2/examples/rknn_yolov5_demo/install/rknn_yolov5_demo_
Linux/lib/librockchip_mpp.so
63  -- Installing: /rknpu2/examples/rknn_yolov5_demo/install/rknn_yolov5_demo_
Linux/lib/libmk_api.so
/rknpu2/examples/rknn_yolov5_demo

```

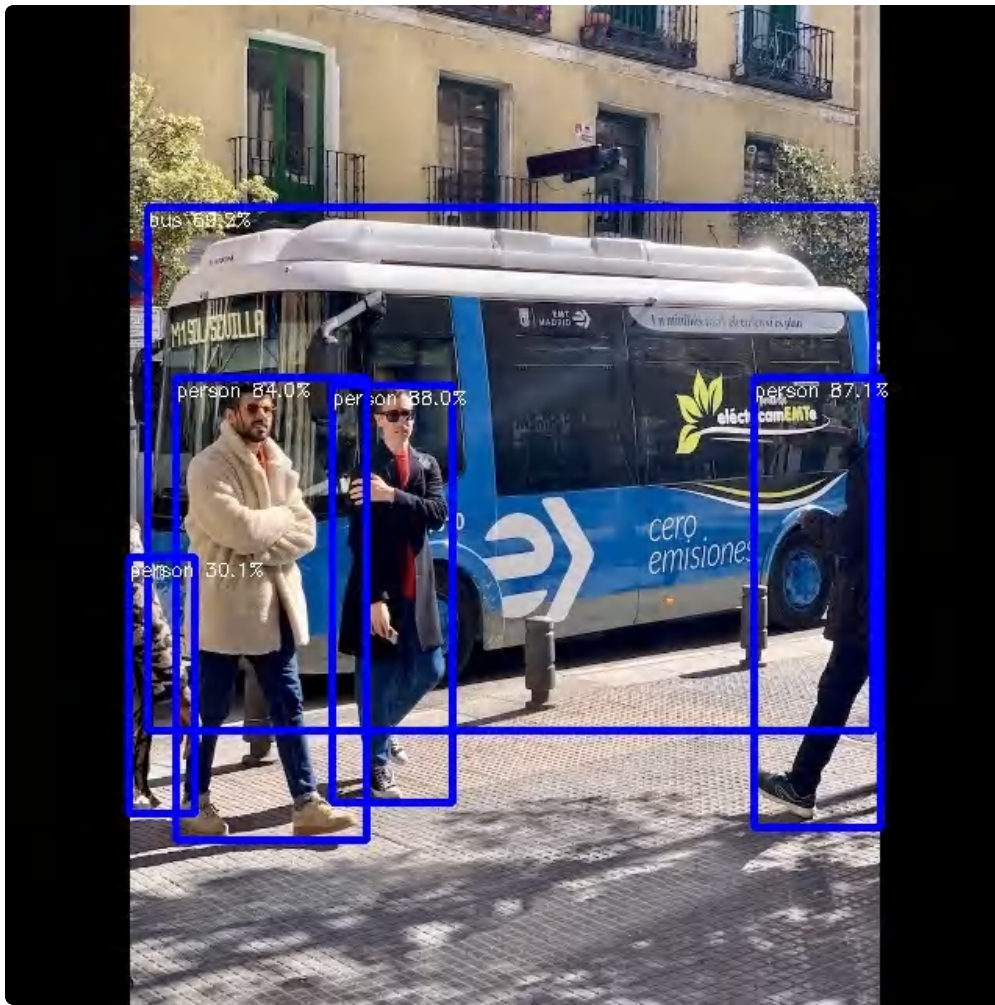
2.3.2. 测试

```

▼ Shell |
1  root@linaro-alip:/rknpu2/examples/rknn_yolov5_demo# cd install/rknn_yolov5_
_demo_Linux/
2  root@linaro-alip:/rknpu2/examples/rknn_yolov5_demo/install/rknn_yolov5_dem
o_Linux# chmod a+x rknn_yolov5_demo
3  root@linaro-alip:/rknpu2/examples/rknn_yolov5_demo/install/rknn_yolov5_dem
o_Linux# ./rknn_yolov5_demo ./model/RK3588/yolov5s-640-640.rknn ./model/bu
s.jpg
4  post process config: box_conf_threshold = 0.25, nms_threshold = 0.45
5  Loading mode...
6  sdk version: 1.5.2 (c6b7b351a@2023-08-23T15:28:22) driver version: 0.9.8
7  model input num: 1, output num: 3
8  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=INT
8, qnt_type=AFFINE, zp=-128, scale=0.003922
9  index=0, name=output, n_dims=4, dims=[1, 255, 80, 80], n_elems=1632000,
size=1632000, w_stride = 0, size_with_stride=1638400, fmt=NCHW, type=INT
8, qnt_type=AFFINE, zp=-128, scale=0.003860
10 index=1, name=283, 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_t
ype=AFFINE, zp=-128, scale=0.003922
11 index=2, name=285, 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_t
ype=AFFINE, zp=-128, scale=0.003915
12 model is NHWC input fmt
13 model input height=640, width=640, channel=3
14 Read ./model/bus.jpg ...
15 img width = 640, img height = 640
16 once run use 29.218000 ms
17 loadLabelName ./model/coco_80_labels_list.txt
18 person @ (209 244 286 506) 0.884139
19 person @ (478 238 559 526) 0.867678
20 person @ (110 238 230 534) 0.824685
21 bus @ (94 129 553 468) 0.705055
22 person @ (79 354 122 516) 0.339254
23 save detect result to ./out.jpg
24 loop count = 10 , average run 20.956600 ms

```

执行完后会得到转换后的模型图片，如下所示：



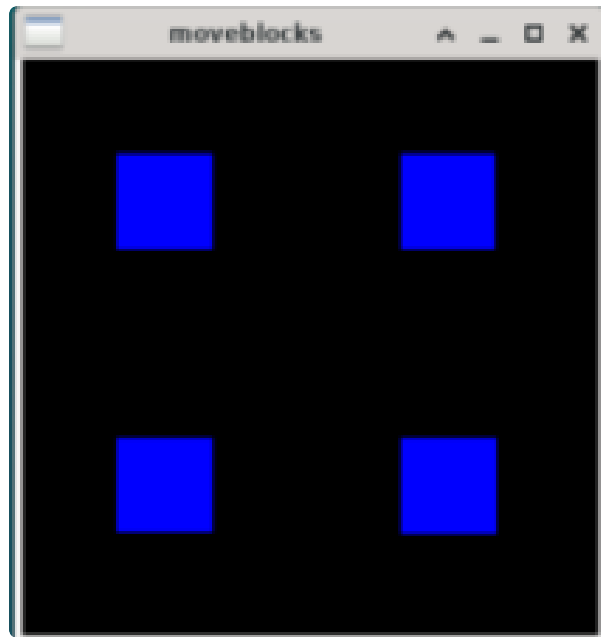
2.4. QT5

支持qt5.15，测试方法如下：

▼ Shell

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

结果如下所示：

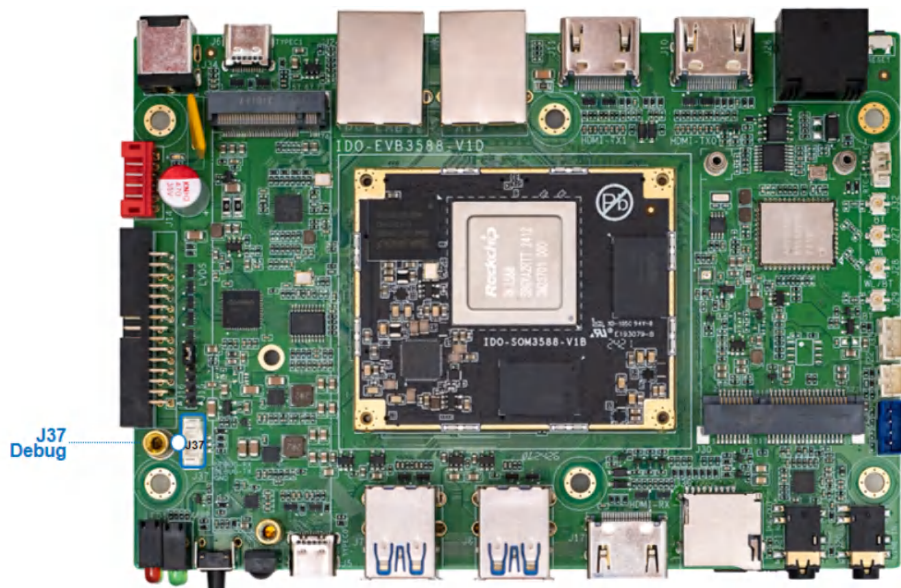


3. IDO-EVB3588-V1D-Ubunut系统

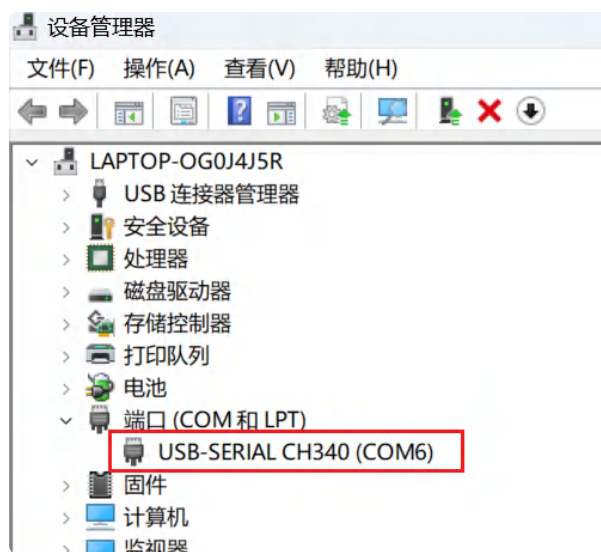
3.1. 功能及接口使用方法

3.1.1. DEBUG UART

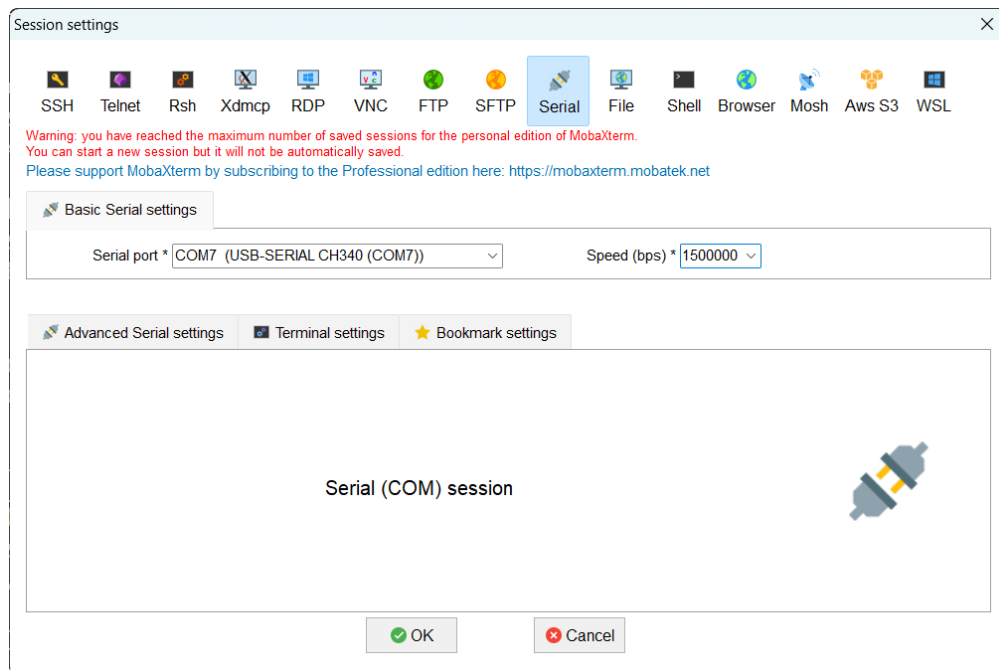
DEBUG UART位置J37，如下图所示：



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：

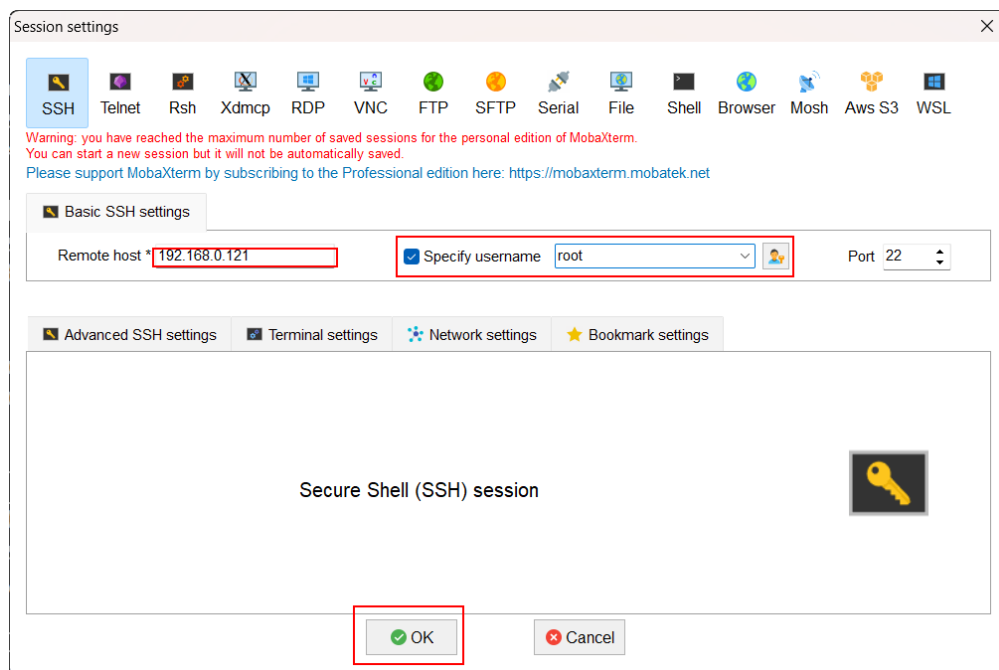


账号密码：industio/123456

3.1.2. SSH登录

在知道IP的前提下，默认可以通过SSH登录进系统。

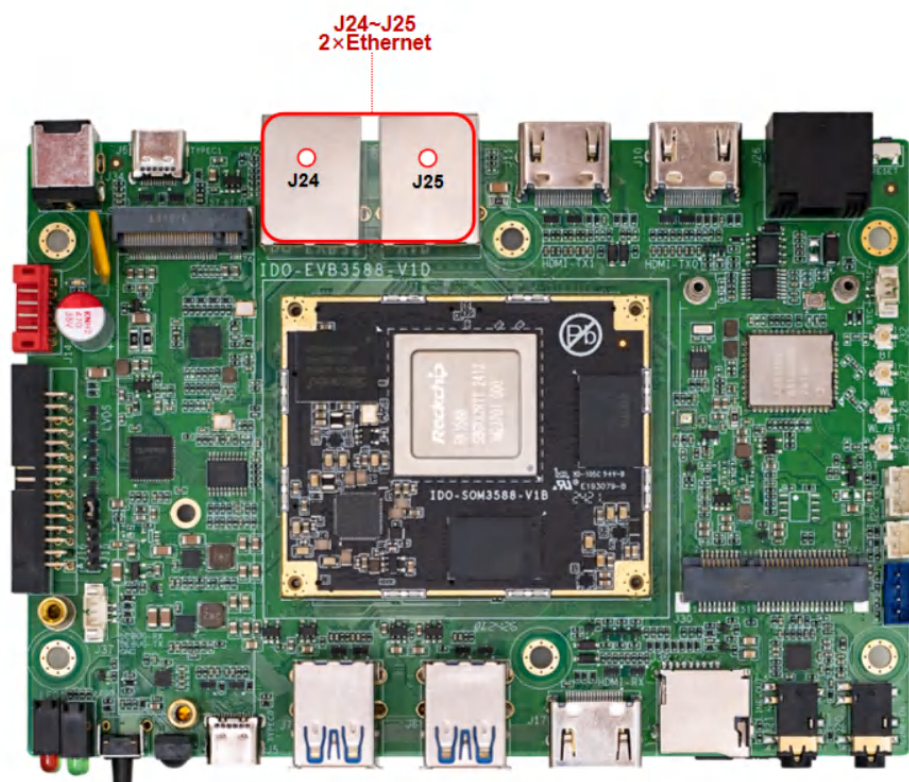
SSH登录账号密码：industio/123456



3.1.3. 网络

3.1.3.1. 以太网

主板有两路千兆以太网接口位置J24,J25，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。

3.1.3.2. Ethernet动态IP设置

```
1 #eth0
2 industio@ido:~$ ifconfig eth0
3 eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
4     inet 192.168.0.90 netmask 255.255.255.0 broadcast 192.168.0.255
5     inet6 fe80::1b83:a156:cb16:c6b0 prefixlen 64 scopeid 0x20<link>
6     ether fa:c6:ba:c7:0c:60 txqueuelen 1000 (Ethernet)
7     RX packets 141 bytes 15129 (14.7 KiB)
8     RX errors 0 dropped 4 overruns 0 frame 0
9     TX packets 89 bytes 9306 (9.0 KiB)
10    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
11    device interrupt 121
12
13 #eth1
14 industio@ido:~$ ifconfig eth1
15 eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
16     inet 192.168.0.91 netmask 255.255.255.0 broadcast 192.168.0.255
17     inet6 fe80::8992:70c3:f104:e0be prefixlen 64 scopeid 0x20<link>
18     ether fe:c6:ba:c7:0c:60 txqueuelen 1000 (Ethernet)
19     RX packets 293 bytes 30522 (29.8 KiB)
20     RX errors 0 dropped 6 overruns 0 frame 0
21     TX packets 151 bytes 15448 (15.0 KiB)
22     TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
23     device interrupt 123
```

3.1.3.3. Ethernet静态IP设置

```

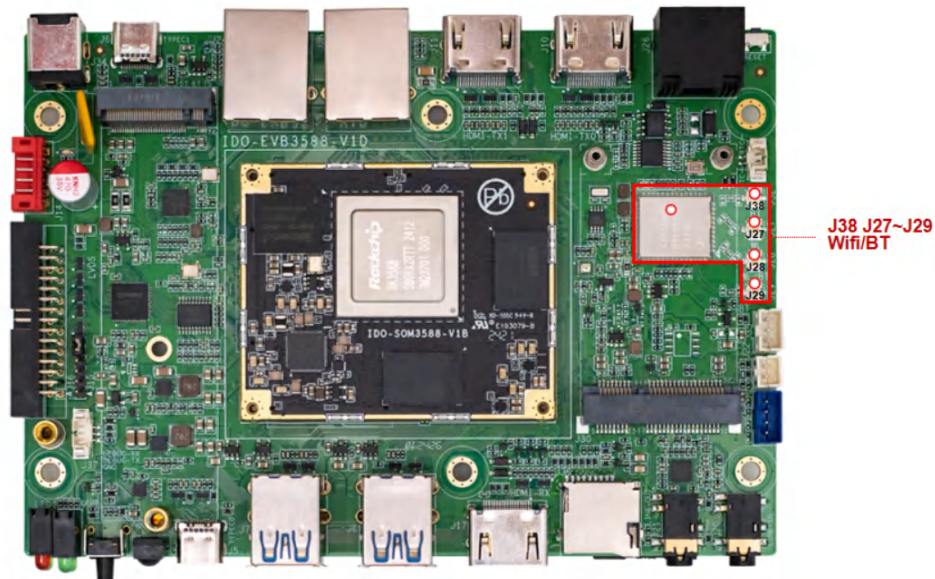
1  industio@ido:~$ sudo touch /etc/netplan/01-netcfg.yaml
2  industio@ido:~$ sudo vim /etc/netplan/01-netcfg.yaml
3  industio@ido:~$ cat /etc/netplan/01-netcfg.yaml
4  network:
5      version: 2
6      ethernet:
7          eth0:
8              dhcp4: no
9              addresses: [192.168.3.121/24]
10             nameservers:
11                 addresses: [192.168.3.1, 8.8.8.8, 114.114.1
12                        14.114]
13             routes:
14                 - to: 0.0.0.0/0
15                   via: 192.168.3.1
16                   metric: 100
17 #立即生效
18 industio@ido:~$ sudo netplan apply
19 Cannot call openvswitch: ovsdb-server.service is not running.
20 industio@ido:~$ ifconfig eth0
21 eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
22     inet 192.168.3.121 netmask 255.255.255.0 broadcast 192.168.3.255
23     inet6 fe80::1b83:a156:cb16:c6b0 prefixlen 64 scopeid 0x20<link>
24     inet6 fe80::f8c6:baff:fec7:c60 prefixlen 64 scopeid 0x20<link>
25     ether fa:c6:ba:c7:0c:60 txqueuelen 1000 (Ethernet)
26     RX packets 4203 bytes 417790 (407.9 KiB)
27     RX errors 0 dropped 164 overruns 0 frame 0
28     TX packets 392 bytes 32299 (31.5 KiB)
29     TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
30     device interrupt 121

```

设备断电重启，此静态IP设置仍然生效。

3.1.4. 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：

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

查看wlan0的IP地址，确认连接成功：

```
1 root@industio:/# ifconfig wlan0
2 wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
3     inet 192.168.0.118 netmask 255.255.255.0 broadcast 192.168.0.255
4     inet6 fe80::1036:80b4:5082:7440 prefixlen 64 scopeid 0x20<link>
5     ether c0:f5:35:12:ad:a2 txqueuelen 1000 (Ethernet)
6     RX packets 170 bytes 22149 (21.6 KiB)
7     RX errors 0 dropped 5 overruns 0 frame 0
8     TX packets 24 bytes 2683 (2.6 KiB)
9     TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
10
11 root@industio:/# ping www.baidu.com
12 PING www.wshifen.com (103.235.47.188) from 192.168.0.118 wlan0: 56(84) bytes of data.
13 64 bytes from 103.235.47.188 (103.235.47.188): icmp_seq=1 ttl=45 time=274 ms
14 64 bytes from 103.235.47.188 (103.235.47.188): icmp_seq=2 ttl=45 time=299 ms
15 64 bytes from 103.235.47.188 (103.235.47.188): icmp_seq=3 ttl=45 time=320 ms
16 64 bytes from 103.235.47.188 (103.235.47.188): icmp_seq=4 ttl=45 time=241 ms
```

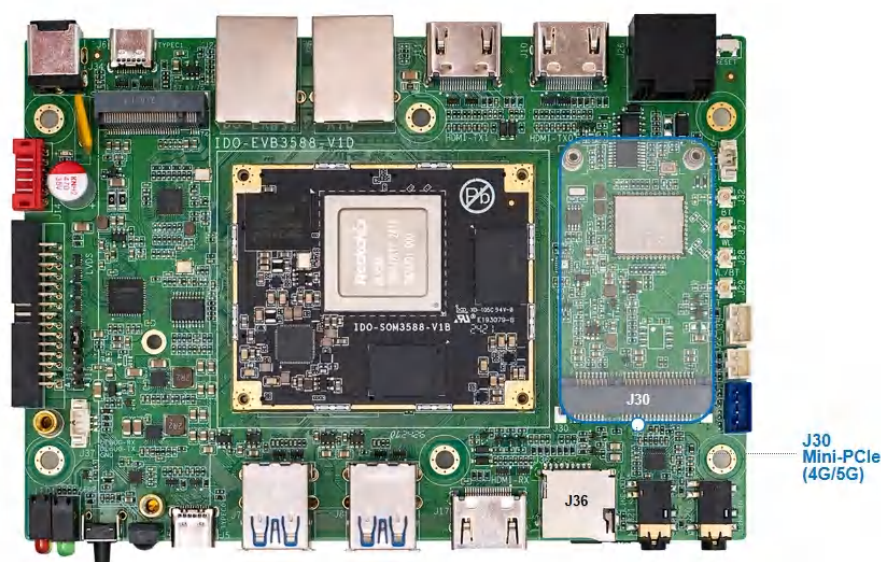
3.1.5. Bluetooth

3.1.5.1. 连接方法

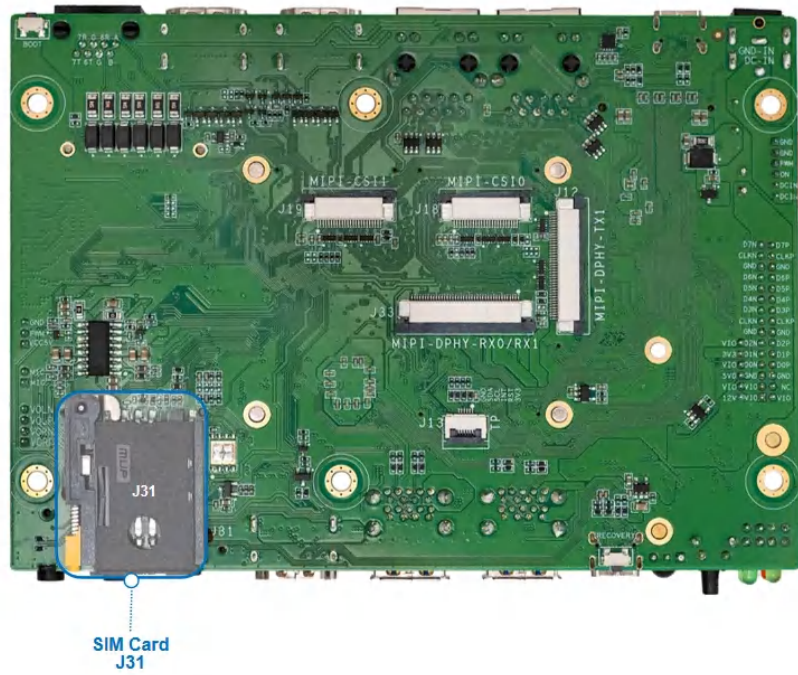
```
1 #打开蓝牙
2 root@industio:/# bluetoothctl
3 ▾ [bluetooth]# power on
4 #扫描蓝牙
5 ▾ [bluetooth]# scan on
6 #查看蓝牙设备
7 ▾ [bluetooth]# devices
8 #信任蓝牙设备
9 ▾ [bluetooth]# trust AC:C4:BD:6C:F2:10
10 #蓝牙配对
11 ▾ [bluetooth]# pair AC:C4:BD:6C:F2:10
12 #连接蓝牙
13 ▾ [bluetooth]# connect AC:C4:BD:6C:F2:10
```

3.1.6. 4G/5G

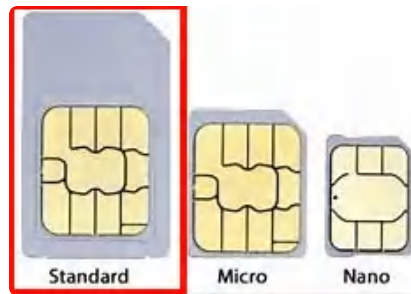
主板内置Mini PCIe 座子扩展 4G/5G模块，4G通信模块适配移远EC20、EC25等通用模组。5G通信模块适配移远RG200U-CN。模块安装位J30，如下图所示：



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J31位于主板背面，如下图所示：



使用Micro尺寸SIM卡，如下图所示：

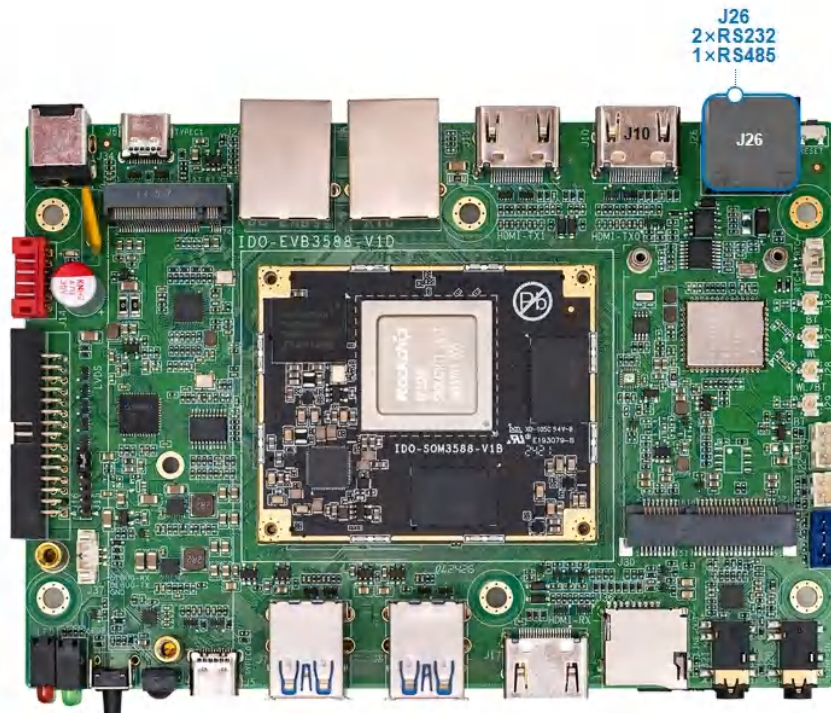


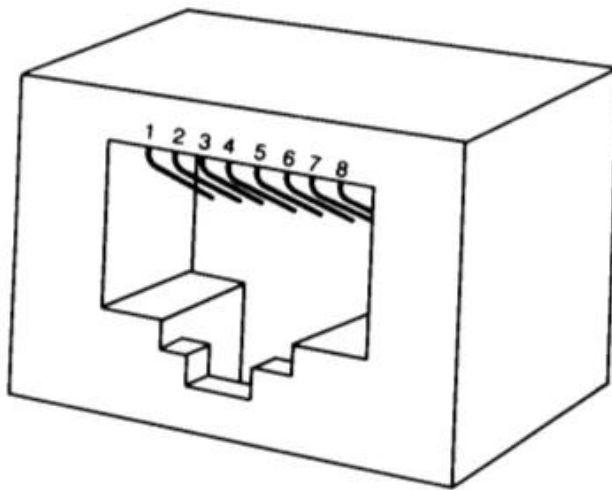
4G/5G:

```
1 #拨号
2 industio@ido:~$ quectel-CM &
3
4 #查看网络
5 industio@ido:~$ ifconfig eth2
6 eth2: flags=193<UP,RUNNING,NOARP>  mtu 1500
7     inet 10.101.61.51  netmask 255.255.255.248
8     inet6 fe80::fc:f6ff:fe8d:bab6  prefixlen 64  scopeid 0x20<link>
9     ether 02:fc:f6:8d:ba:b6  txqueuelen 1000  (Ethernet)
10    RX packets 42  bytes 7013 (6.8 KiB)
11    RX errors 0  dropped 0  overruns 0  frame 0
12    TX packets 57  bytes 4608 (4.5 KiB)
13    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0
14
15 #测试通信
16 industio@ido:~$ ping www.baidu.com
```

3.1.7. UART

主板一共引出3路串口（不含调试串口）共用J26端口，如下图所示：





UART设备节点列表如下：

引脚序号	丝印	默认电平类型	功能	设备节点
1	A	RX485	A	/dev/ttyS4
2	B		B	
3	6R	RS232	RX	/dev/ttyS6
4	GND	GND	/	/
5	GND			
6	6T	RS232	TX	/dev/ttyS6
7	7R	RS232	RX	/dev/ttyS7
8	7T		TX	

▼

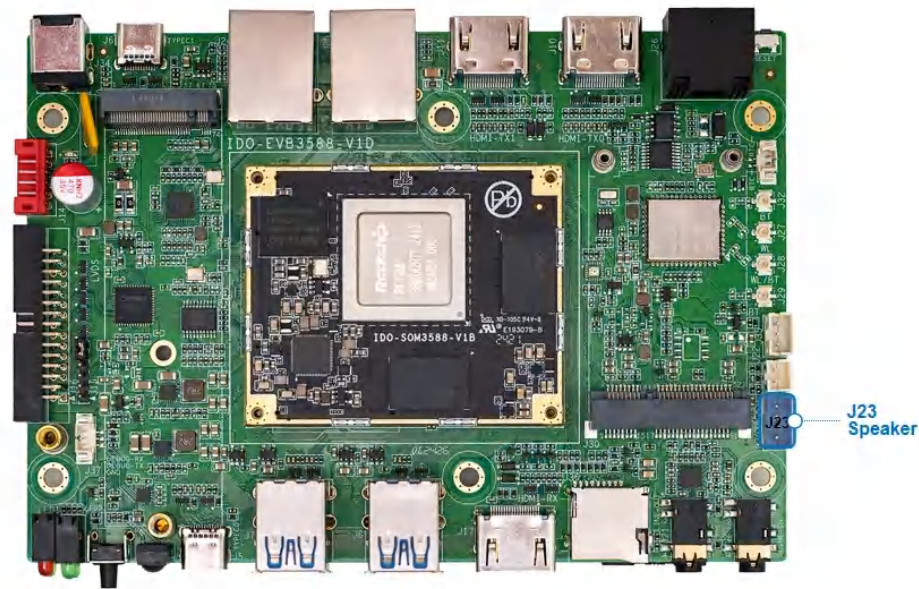
C |

```
1  industio@ido:~$ microcom -p /dev/ttyS4 -s 115200
```

3.1.8. 声音

3.1.8.1. 喇叭

喇叭接口为PH2.0–4P连接器J38，如下图所示：



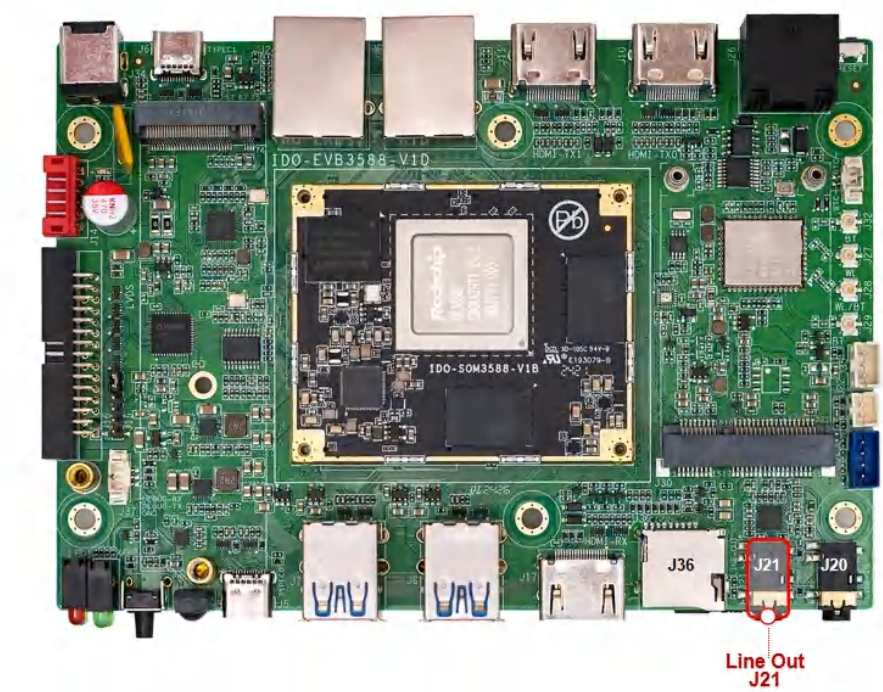
- 支持双声道，每个声道支持4Ω 3W输出

Plain Text

```
1  #查看声卡
2  root@industio:/# aplay -l
3
4  #音频播放到HDMI0
5  root@industio:/# aplay -D hw:0,0 xihuangni.wav
6  Playing WAVE 'xihuangni.wav' : Signed 16 bit Little Endian, Rate 44100 H
  z, Stereo
7
8  #音频播放到HDMI1
9  root@industio:/# aplay -D hw:1,0 xihuangni.wav
10 Playing WAVE 'xihuangni.wav' : Signed 16 bit Little Endian, Rate 44100 H
   z, Stereo
11
12 #音频播放到DP
13 root@industio:/# aplay -D hw:2,0 xihuangni.wav
14 Playing WAVE 'xihuangni.wav' : Signed 16 bit Little Endian, Rate 44100 H
   z, Stereo
15
16 #音频播放到声卡
17 root@industio:/# aplay -D hw:3,0 xihuangni.wav
18 Playing WAVE 'xihuangni.wav' : Signed 16 bit Little Endian, Rate 44100 H
   z, Stereo
```

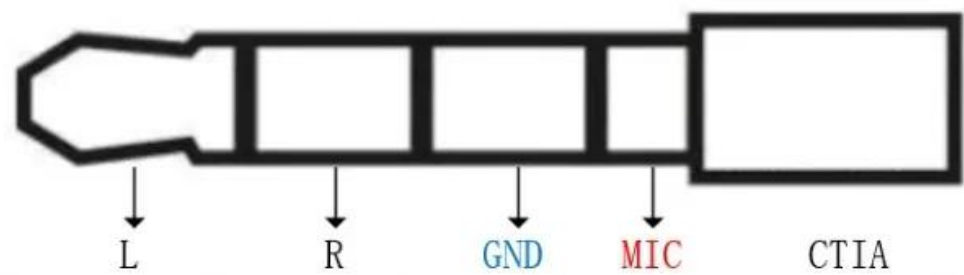
3.1.8.2. 耳机

主板支持1路3.5mm四节耳机座J21，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：



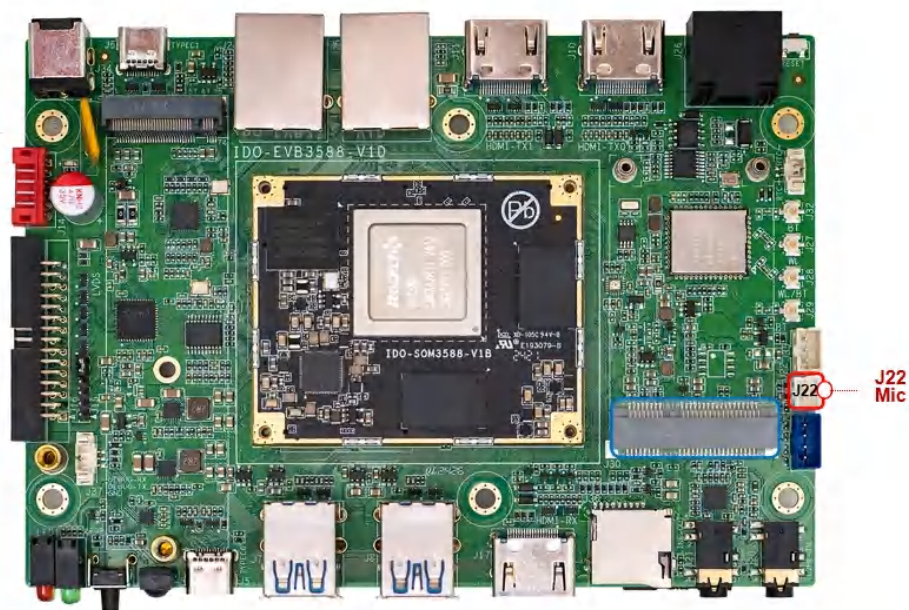
耳机录音：

▼ Plain Text |

```
1 #耳机录音
2 industio@ido:~$ amixer -c 3 cset numid=43 on
3
4 industio@ido:~$ arecord -D hw:3,0 -f cd -r 44100 -c 2 test.wav
5 Recording WAVE 'test.wav' : Signed 16 bit Little Endian, Rate 44100 Hz, Stereo
```

3.1.8.3. Mic

MX1.25T-2P麦克风接口J22，如下图所示：



Mic录音：

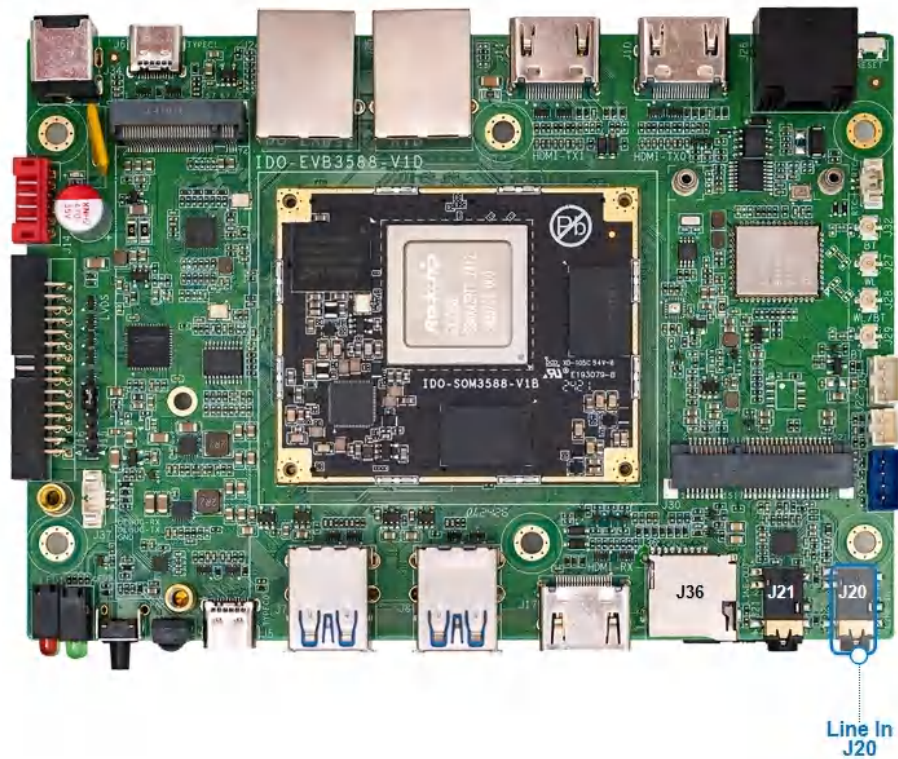


Plain Text

```
1 #Mic录音
2 industio@ido:~$ arecord -D hw:3,0 -f S16_LE -r 44100 -c 2 test.wav
3 Recording WAVE 'test.wav' : Signed 16 bit Little Endian, Rate 44100 Hz, Stereo
```

3.1.8.4. Line-in

主板支持1路3.5mm 三节音频Line-in接口J20，如下图所示：

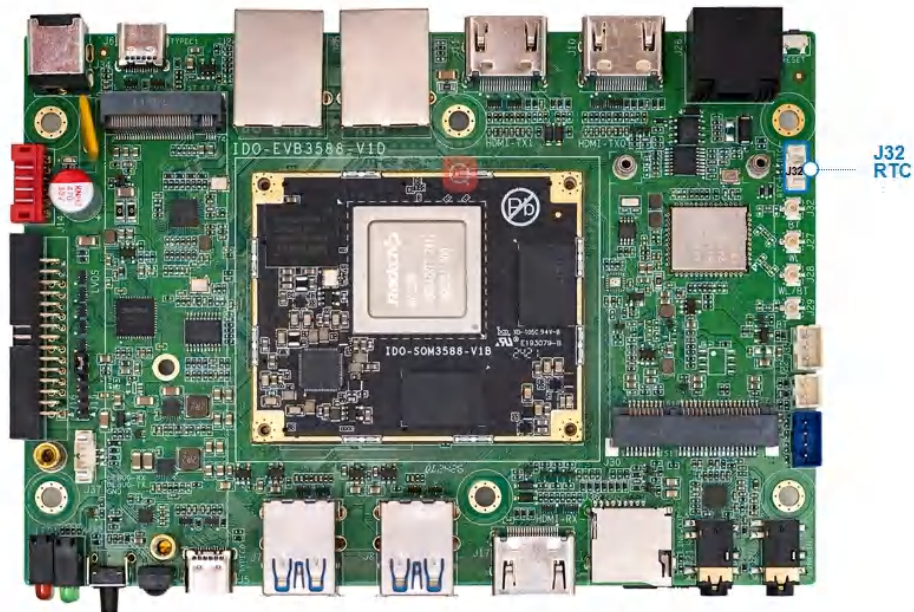


Plain Text |

```
1 #切换到line in
2 root@ido:~$ echo -n "mic1" > /sys/class/es8388_mic/mic_channel
3
4 #录音
5 industio@ido:~$ arecord -D hw:3,0 -f cd -r 44100 -c 2 test.wav
6 Recording WAVE 'test.wav' : Signed 16 bit Little Endian, Rate 44100 Hz, Stereo
```

3.1.9. 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：

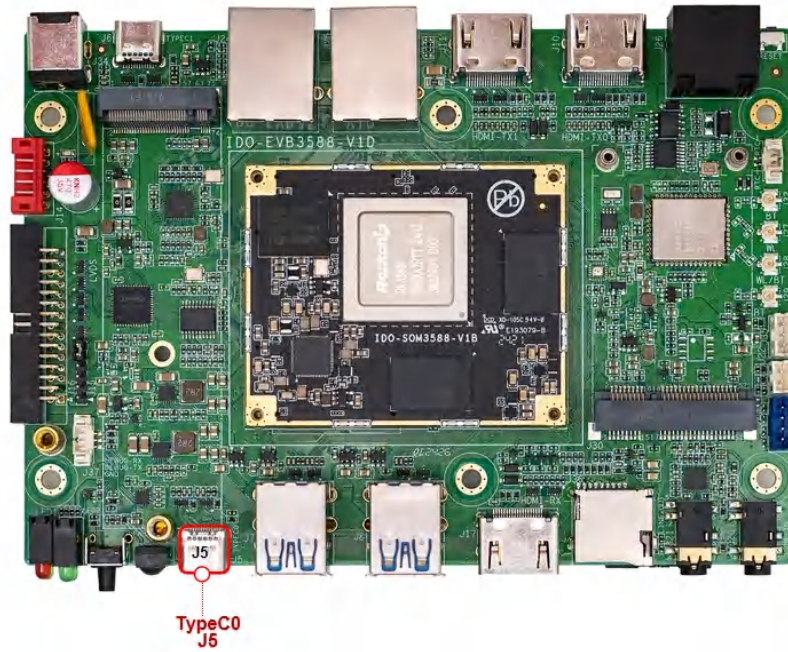
```
1  #设置时间
2  industio@ido:~$ sudo date -s "2024-10-09 14:02:30"
3
4  #将rtc时钟调整为与目前的系统时钟一致
5  industio@ido:~$ sudo hwclock -w
6
7  #获取硬件rtc当前时间（断电重启读取时间没有太大偏差）
8  industio@ido:~$ sudo hwclock -r
9  2024-10-09 14:02:35.945604+00:00
```

3.1.10. USB接口

主板支持2个TypeC接口，支持4个USB3.0-A接口，USB对外总供电应小于3A。

3.1.10.1. TypeC0 接口

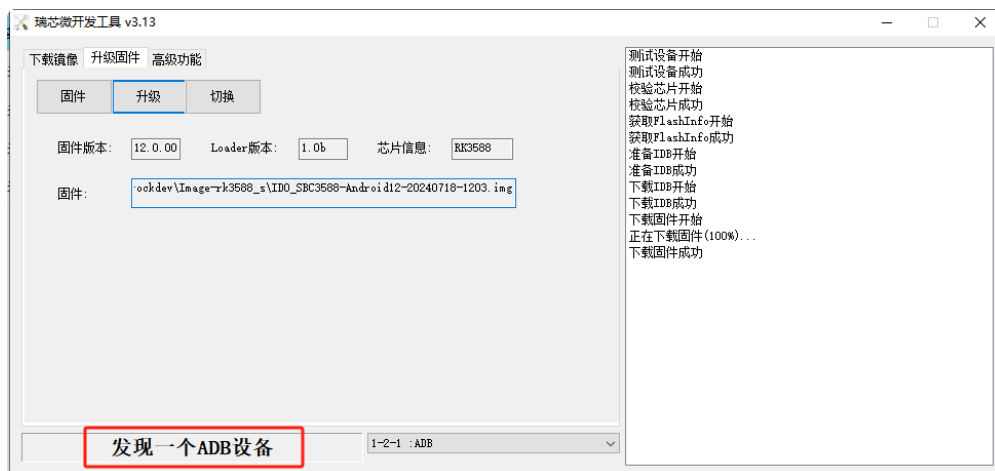
TypeC0接口（USB3 Gen1 OTG）J5，如下图所示：



- 支持Host、Device模式自动切换
- 支持DP显示输出

3.1.10.1.1. 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

3.1.10.1.2. 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

3.1.10.1.3. 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

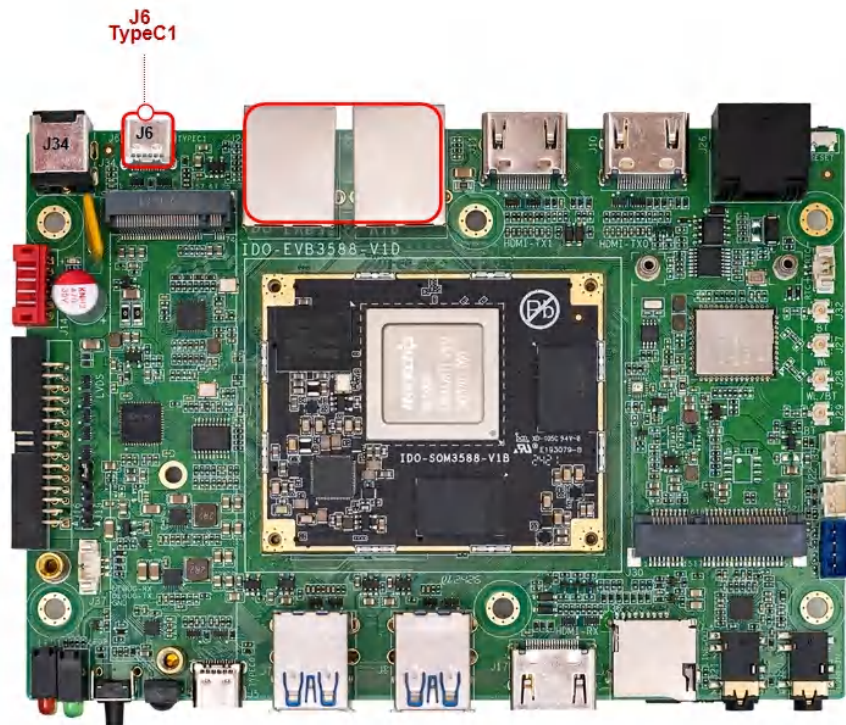


- 主板画面通过TYPE-C DP输出正常，DP声音输出正常

注意：固件默认关闭DP模式

3.1.10.2. TypeC1 接口

TypeC0接口（USB3 Gen1）J6，如下图所示：



- 支持Host模式
- 支持DP显示输出

3.1.10.2.1. 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

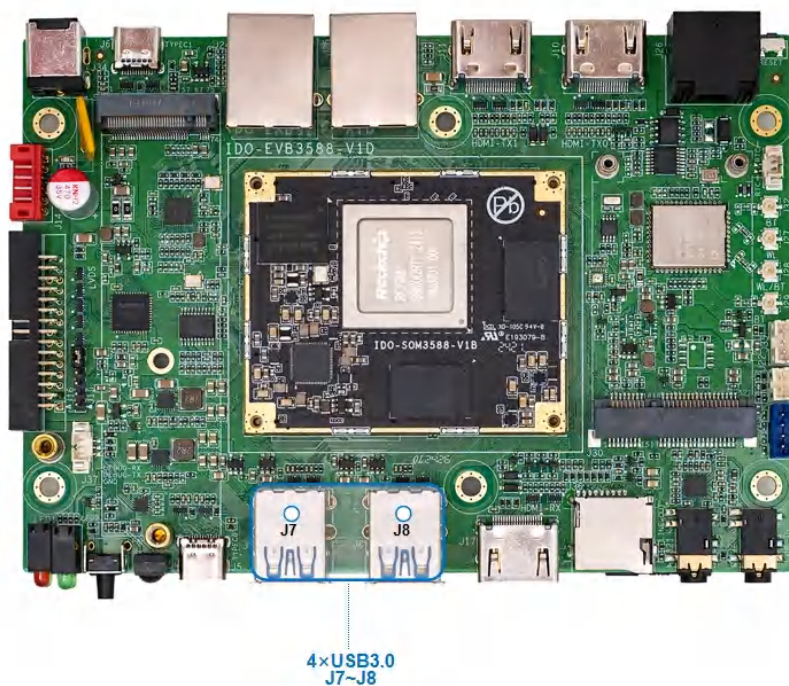
3.1.10.2.2. DP模式

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

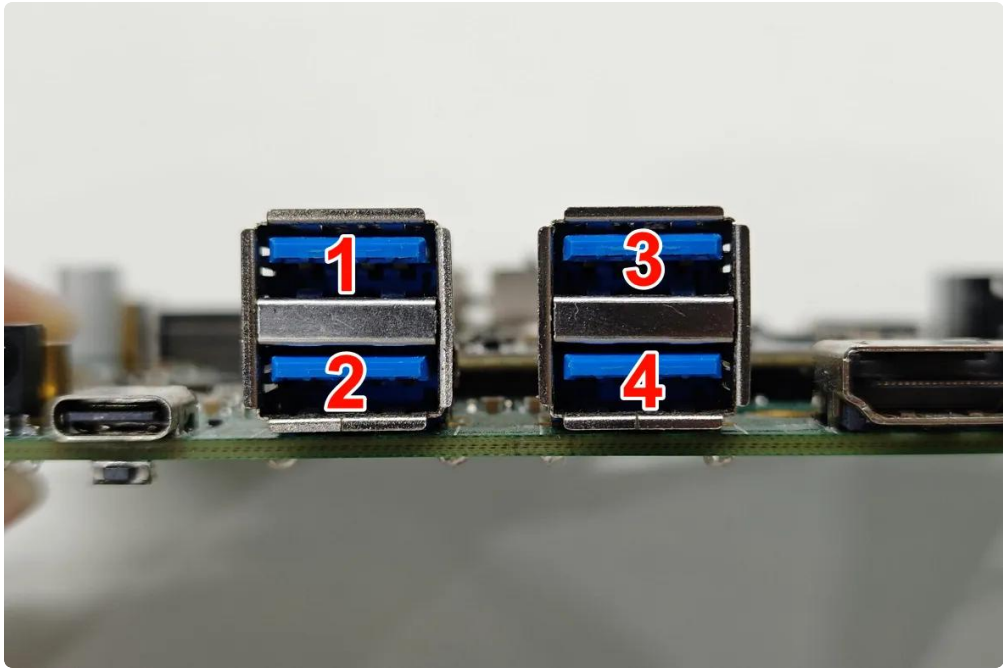


3.1.10.3. USB3.0接口

主板支持4个USB3.0接口，接口为标准的双层A口J7、J8，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：



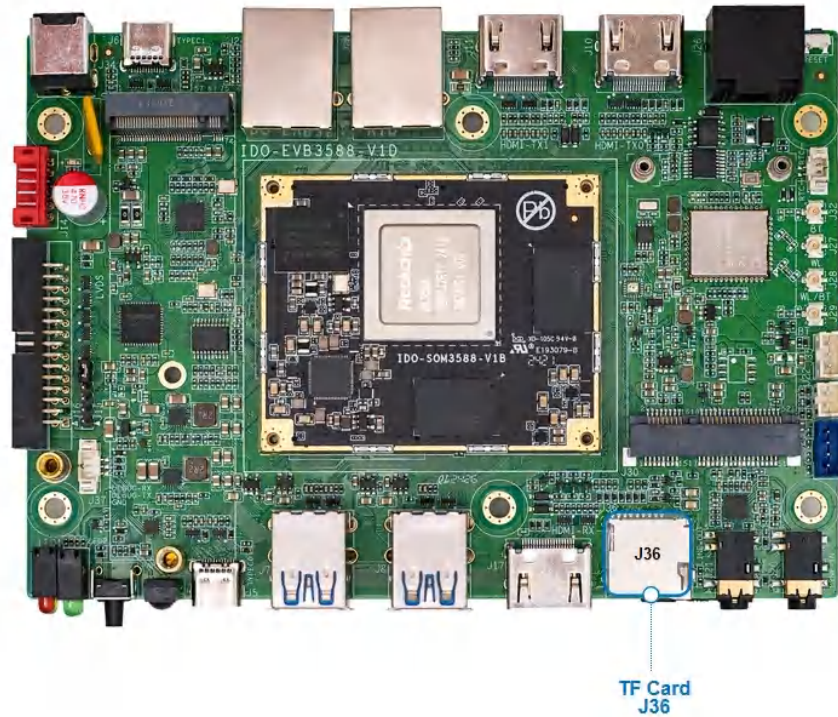
USB电源控制，如下表所示：

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。

3.1.11. TF Card

TF卡座支持SDIO3.0，支持高速SD卡，接口位于J36，如下图所示：



Shell

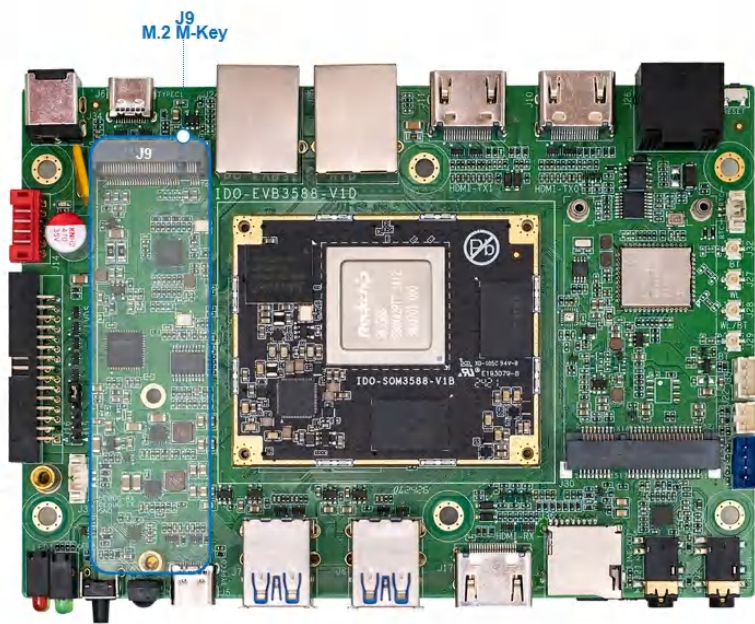
```

1  industio@ido:~$ sudo fdisk -l
2  Disk /dev/ram0: 8 MiB, 8388608 bytes, 16384 sectors
3  Units: sectors of 1 * 512 = 512 bytes
4  Sector size (logical/physical): 512 bytes / 4096 bytes
5  I/O size (minimum/optimal): 4096 bytes / 4096 bytes
6  .....
7
8  设备          启动  起点      末尾      扇区  大小  Id  类型
9  /dev/mmcblk1p1      16 60579839 60579824 28.9G  7  HPFS/NTFS/exFAT
10
11 industio@ido:~$ df -h
12 文件系统      大小  已用  可用  已用% 挂载点
13 /dev/root      56G   8.1G   48G   15% /
14 tmpfs          3.9G    0   3.9G    0% /dev/shm
15 tmpfs          1.6G   4.5M   1.6G    1% /run
16 tmpfs          5.0M   4.0K   5.0M    1% /run/lock
17 tmpfs          3.9G   8.0K   3.9G    1% /tmp
18 /dev/mmcblk0p7  2.0G  292K   2.0G    1% /userdata
19 /dev/mmcblk0p6 124M   12M  109M   10% /oem
20 tmpfs          792M   88K   792M    1% /run/user/1000
21 /dev/mmcblk1p1  29G   64M   29G    1% /media/industio/DECABB50CABB23A5

```

3.1.12. M.2接口

主板上使用标准M.2-M-key M.2接口连接座J9，如下图所示：



支持PCIe2.1通信，适用2280尺寸NVME固态硬盘。

```

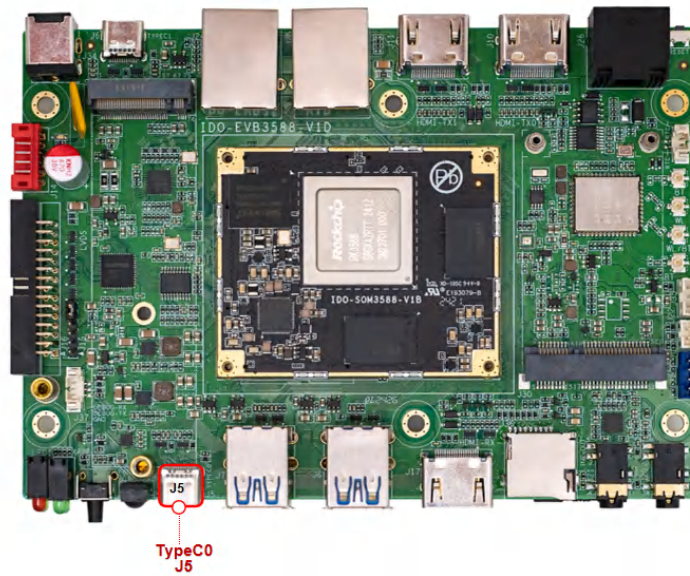
1  industio@ido:~$ sudo fdisk -l
2  Disk /dev/ram0: 8 MiB, 8388608 bytes, 16384 sectors
3  Units: sectors of 1 * 512 = 512 bytes
4  Sector size (logical/physical): 512 bytes / 4096 bytes
5  I/O size (minimum/optimal): 4096 bytes / 4096 bytes
6
7  Disk /dev/ram1: 8 MiB, 8388608 bytes, 16384 sectors
8  Units: sectors of 1 * 512 = 512 bytes
9  Sector size (logical/physical): 512 bytes / 4096 bytes
10 I/O size (minimum/optimal): 4096 bytes / 4096 bytes
11 ...
12 Disk /dev/nvme0n1: 476.94 GiB, 512110190592 bytes, 1000215216 sectors
13 Disk model: XPG GAMMIX S11L
14 Units: sectors of 1 * 512 = 512 bytes
15 Sector size (logical/physical): 512 bytes / 512 bytes
16 I/O size (minimum/optimal): 512 bytes / 512 bytes
17 Disklabel type: dos
18 Disk identifier: 0x00000000
19
20 Device          Boot Start          End      Sectors   Size Id Type
21 /dev/nvme0n1p1      2048 1000215182 1000213135 476.9G  c W95 FAT32 (LBA)
22
23 industio@ido:~$ df -h
24 文件系统      大小  已用  可用  已用% 挂载点
25 /dev/root      56G   8.1G   48G   15% /
26 tmpfs          3.9G     0   3.9G    0% /dev/shm
27 tmpfs          1.6G   4.5M   1.6G    1% /run
28 tmpfs          5.0M   4.0K   5.0M    1% /run/lock
29 tmpfs          3.9G   8.0K   3.9G    1% /tmp
30 /dev/mmcblk0p6  124M   12M   109M   10% /oem
31 /dev/mmcblk0p7   2.0G  292K   2.0G    1% /userdata
32 /dev/nvme0n1p1 233G   14G  219G    6% /media/industio/m2
33 tmpfs          792M   68K   792M    1% /run/user/1000

```

3.1.13. LCD显示

3.1.13.1. DP

DP接口 (TYPE C0) J5, 如下图所示:



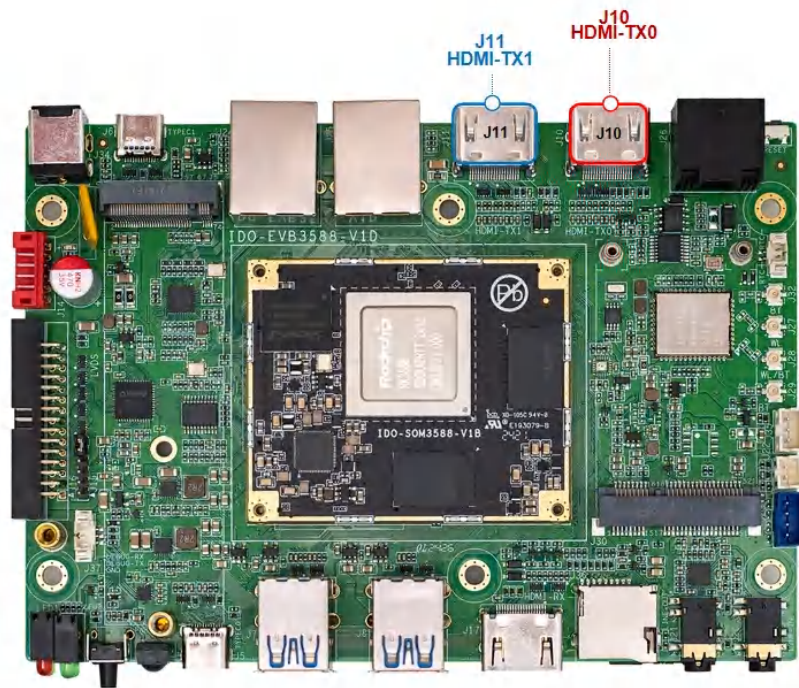
可以使用 USB-C 转 HDMI 高清线连接 HDMI 显示器输出画面，如下图所示：



- 最高支持8K@30fps输出

3.1.13.2. HDMI-TX

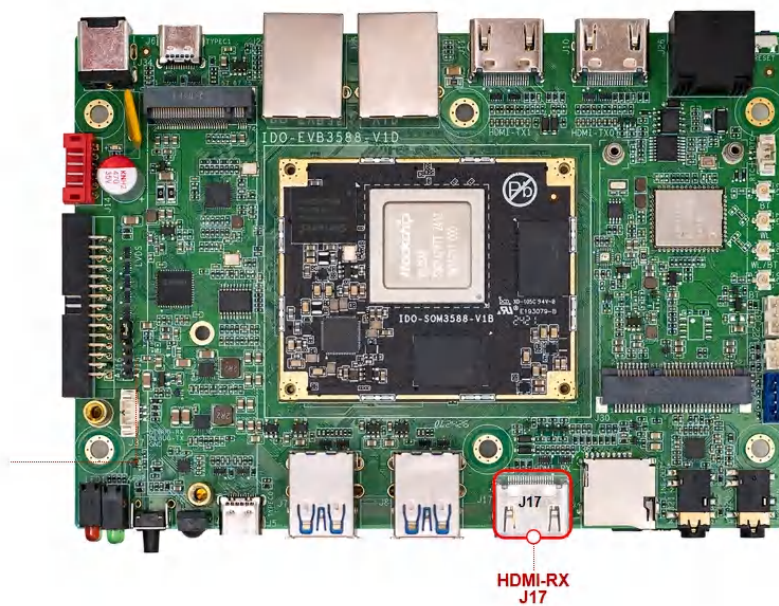
主板支持2路标准HDMI-A接口，J10、J11，如下图所示：



- 最高支持 HDMI2.1 8K@60fps 输出

3.1.13.3. HDMI-RX接口

主板支持1路标准HDMI-A接口的HDMI-RX接口J17，如下图所示：



查看输入hdm信号格式：

```

1  industio@ido:~$ v4l2-ctl -d /dev/video40 --query-dv-timings
2      Active width: 3840
3      Active height: 2160
4      Total width: 4400
5      Total height: 2250
6      Frame format: progressive
7      Polarities: -vsync -hsync
8      Pixelclock: 593992000 Hz (60.00 frames per second)
9      Horizontal frontporch: 176
10     Horizontal sync: 88
11     Horizontal backporch: 296
12     Vertical frontporch: 8
13     Vertical sync: 10
14     Vertical backporch: 72
15     Standards:
16     Flags:
17
18  industio@ido:~$ v4l2-ctl -d /dev/video40 --get-fmt-video
19  Format Video Capture Multiplanar:
20     Width/Height      : 3840/2160
21     Pixel Format       : 'NV12' (Y/CbCr 4:2:0)
22     Field             : None
23     Number of planes  : 1
24     Flags              : premultiplied-alpha, set-csc, 0x000000fc
25     Colourspace       : Rec. 709
26     Transfer Function : Unknown (0x000000b8)
27     YCbCr/HSV Encoding: xvYCC 709
28     Quantization      : Limited Range
29     Plane 0           :
30         Bytes per Line : 3840
31         Size Image     : 12441600

```

3.1.13.3.1. 预览

```

1  industio@ido:~$ gst-launch-1.0 v4l2src device=/dev/video40 ! video/x-raw, format=NV12, width=3840, height=2160, framerate=30/1 ! videoconvert ! autovideosink

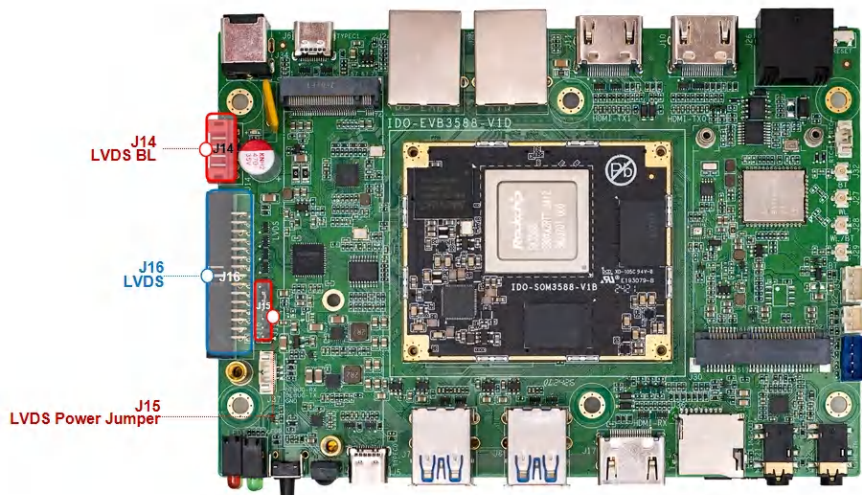
```

- HDMI2.0-RX, 支持4K@30fps

注意：hdmi in预览时需要先确认好分辨率和预览格式。

3.1.13.4. Dual LVDS屏

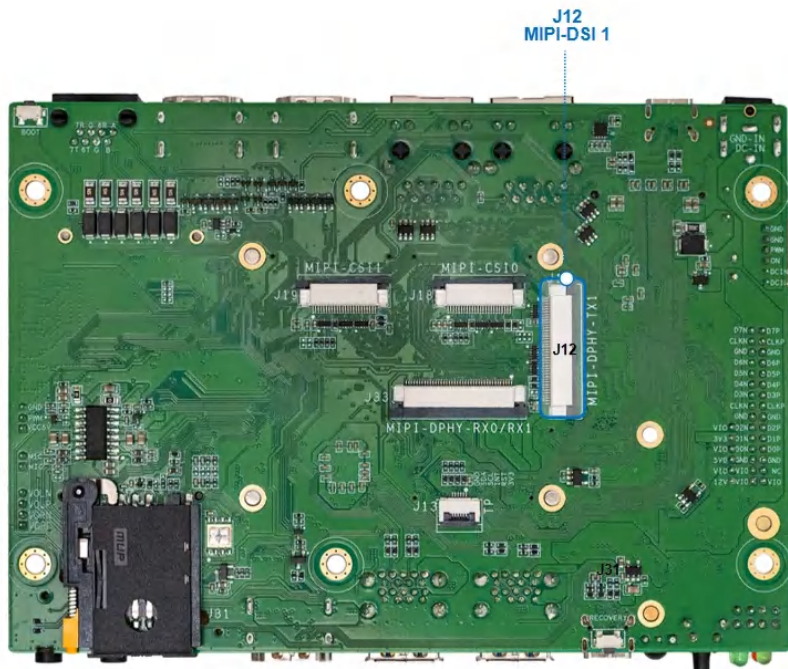
Dual LVDS接口J16，背光接 J14，屏幕供电电压选择接口J18，如下图所示：



J18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供电跳线帽接到3.3V、5V或12V，默认3.3V。

3.1.13.5. MIPI

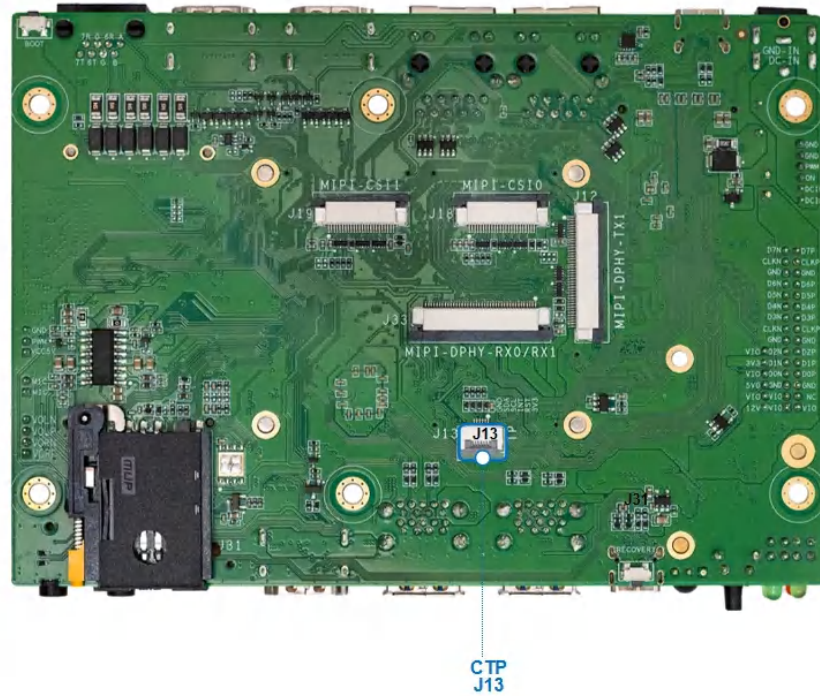
MIPI接口J12 位于主板背面，40Pin FPC 0.5mm 上接座子，如下图所示：



- MIPI供电为3.3V

3.1.14. TP接口

TP接口J13位于主板背面， 6Pin FPC 0.5mm座子， 如下图所示：



- 触摸 TP 接口接线方法为下接

3.1.15. MIPI Camera

MIPI Camer接口J18和J19， 如下图所示：


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

3.1.15.1.2. 抓图

```
1  industio@ido:~$ 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
```

3.1.15.1.3. 抓视频

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

3.1.15.2. MIPI CSI

3.1.15.2.1. 预览命令

```
1  industio@ido:~$ gst-launch-1.0 v4l2src device=/dev/video31 ! video/x-raw, format=NV12, width=1920, height=1080, framerate=30/1 ! videoconvert ! autovideosink
```

3.1.15.2.2. 抓图

```
1  industio@ido:~$ v4l2-ctl --verbose -d /dev/video31 --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
```

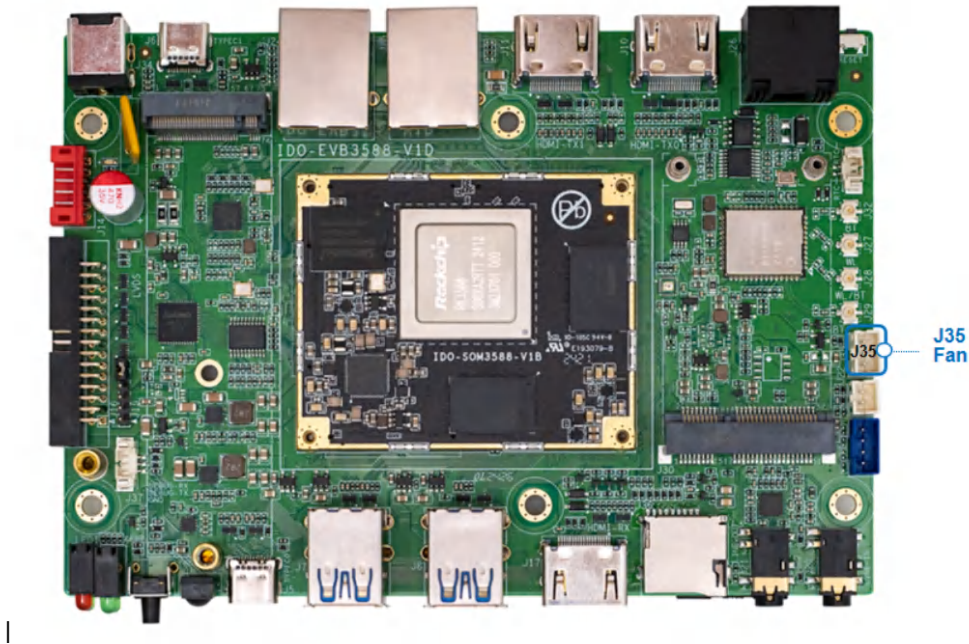
3.1.15.2.3. 抓视频

▼ Plain Text

```
1  industio@ido:~$ v4l2-ctl --stream-mmap=4 -d /dev/video31 --set-fmt-video=w
    idth=1920,height=1080,pixelformat='NV12' --stream-to=./output.yuv
```

3.1.16. FAN 风扇

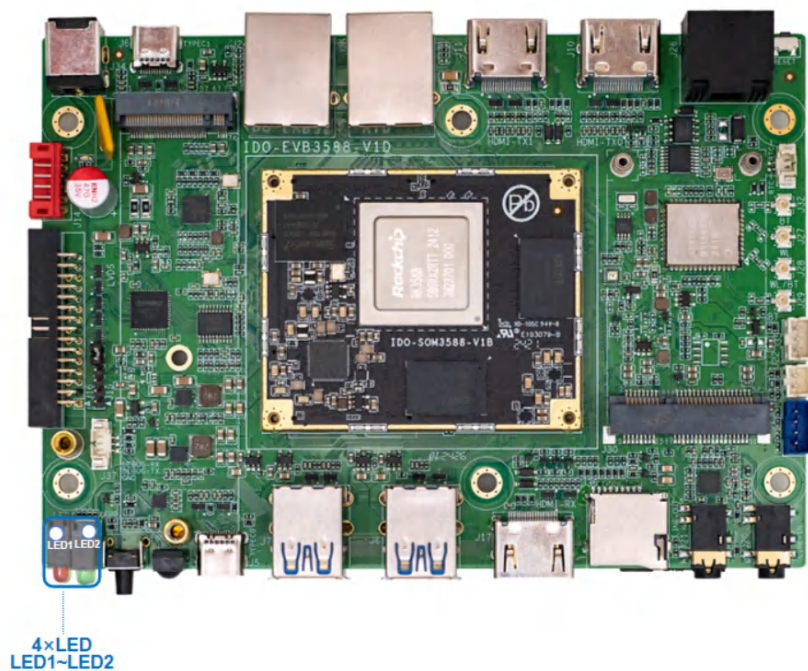
PH2 5V FAN风扇接口J35，如下图所示：



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

3.1.17. 用户LED

主板集成1个用户自定义控制LED2，如下图所示：



其中LED2的上层对应节点为 `/sys/class/leds/led2/brightness`

LED控制方法，如下表所示：

序号	操作	控制命令
LED2	打开	<code>echo 1 > /sys/class/leds/led2/brightness</code>
	关闭	<code>echo 0 > /sys/class/leds/led2/brightness</code>

3.1.18. IR

主板配置了一路红外接口，支持NEC编码遥控器，默认适配的遥控器型号为HTR-A07。



HTR-A07的键值表如下：

编号	按键	键值		编号	按键	键值
1	电源	KEY_POWER		21	1	KEY_1
2	TV	KEY_SCREEN		22	2	KEY_2
3	橙色	KEY_F1		23	3	KEY_3
4	绿色	KEY_F2		24	4	KEY_4
5	黄色	KEY_F3		25	5	KEY_5
6	紫色	KEY_F4		26	6	KEY_6
7	音量+	KEY_VOLUMEUP		27	7	KEY_7
8	音量-	KEY_VOLUMEDOWN		28	8	KEY_8
9	屏显	KEY_DISPLAY_OFF		29	9	KEY_9
10	静音	KEY_MUTE		30	TVNOW	KEY_DOT
11	上一节目	KEY_VIDEO_PREV		31	0	KEY_0
12	下一节目	KEY_VIDEO_NEXT		32	截屏	KEY_PRINT

13	上	KEY_UP		33		
14	左	KEY_LEFT		34		
15	下	KEY_DOWN		35		
16	右	KEY_RIGHT		36		
17	确认	KEY_ENTER		37		
18	返回	KEY_BACK		38		
19	主页	KEY_HOME		39		
20	菜单	KEY_MENU		40		

注：使用以下命令可以从调试串口打印按键的键值。

▼

Bash

```
1 echo 1 > /sys/module/rockchip_pwm_remotectl/parameters/code_print
```

使用evtest工具可以查看按键上报键值：

```
1  industio@ido:~$ evtest
2  No device specified, trying to scan all of /dev/input/event*
3  Available devices:
4  /dev/input/event0:      febd0030.pwm
5  /dev/input/event1:      rockchip-hdmi0 rockchip-hdmi0
6  /dev/input/event2:      rockchip-hdmi1 rockchip-hdmi1
7  /dev/input/event3:      rockchip,dp0 rockchip,dp0
8  /dev/input/event4:      rk805 pwrkey
9  /dev/input/event5:      adc-keys
10 /dev/input/event6:      headset-keys
11 /dev/input/event7:      rockchip-es8388 Headset
12 /dev/input/event8:      USB OPTICAL MOUSE
13 Select the device event number [0-8]: 0
14 Input driver version is 1.0.1
15 Input device ID: bus 0x19 vendor 0x524b product 0x6 version 0x100
16 Input device name: "febd0030.pwm"
17 Supported events:
18   Event type 0 (EV_SYN)
19   Event type 1 (EV_KEY)
20     Event code 2 (KEY_1)
21     Event code 3 (KEY_2)
22     Event code 4 (KEY_3)
23     Event code 5 (KEY_4)
24     Event code 6 (KEY_5)
25     Event code 7 (KEY_6)
26     Event code 8 (KEY_7)
27     Event code 9 (KEY_8)
28     Event code 10 (KEY_9)
29     Event code 11 (KEY_0)
30     Event code 14 (KEY_BACKSPACE)
31     Event code 28 (KEY_ENTER)
32     Event code 52 (KEY_DOT)
33     Event code 61 (KEY_F3)
34     Event code 102 (KEY_HOME)
35     Event code 103 (KEY_UP)
36     Event code 104 (KEY_PAGEUP)
37     Event code 105 (KEY_LEFT)
38     Event code 106 (KEY_RIGHT)
39     Event code 108 (KEY_DOWN)
40     Event code 109 (KEY_PAGEDOWN)
41     Event code 113 (KEY_MUTE)
42     Event code 114 (KEY_VOLUMEDOWN)
43     Event code 115 (KEY_VOLUMEUP)
44     Event code 116 (KEY_POWER)
45     Event code 139 (KEY_MENU)
```

```

46      Event code 143 (KEY_WAKEUP)
47      Event code 158 (KEY_BACK)
48      Event code 183 (KEY_F13)
49      Event code 184 (KEY_F14)
50      Event code 185 (KEY_F15)
51      Event code 186 (KEY_F16)
52      Event code 217 (KEY_SEARCH)
53      Event code 232 (KEY_REPLY)
54      Event code 241 (KEY_VIDEO_NEXT)
55      Event code 242 (KEY_VIDEO_PREV)
56      Event code 245 (KEY_DISPLAY_OFF)
57      Event code 248 (KEY_MICMUTE)
58      Event code 338 (?)
59      Event code 373 (KEY_MODE)
60      Event code 375 (KEY_SCREEN)
61      Event code 388 (KEY_TEXT)
62      Event code 400 (KEY_YELLOW)
63      Event code 401 (KEY_BLUE)
64      Event code 402 (KEY_CHANNELUP)
65
66  Properties:
67  Testing ... (interrupt to exit)
68  [ 4237.196132] USERCODE=0x1818
69  [ 4237.223070] RMC_GETDATA=ff
70  Event: time 1682532011.417156, type 1 (EV_KEY), code 2 (KEY_1), value 1
71  Event: time 1682532011.417156, ----- SYN_REPORT -----
72  Event: time 1682532011.475441, type 1 (EV_KEY), code 2 (KEY_1), value 0
73  Event: time 1682532011.475441, ----- SYN_REPORT -----
74  [ 4237.866797] USERCODE=0x1818
75  [ 4237.893797] RMC_GETDATA=fe
76  Event: time 1682532012.087855, type 1 (EV_KEY), code 3 (KEY_2), value 1
77  Event: time 1682532012.087855, ----- SYN_REPORT -----
78  Event: time 1682532012.178638, type 1 (EV_KEY), code 3 (KEY_2), value 0
79  Event: time 1682532012.178638, ----- SYN_REPORT -----
80  [ 4238.370395] USERCODE=0x1818
81  [ 4238.397454] RMC_GETDATA=fd
82  Event: time 1682532012.591469, type 1 (EV_KEY), code 4 (KEY_3), value 1
83  Event: time 1682532012.591469, ----- SYN_REPORT -----
84  Event: time 1682532012.682093, type 1 (EV_KEY), code 4 (KEY_3), value 0
85  Event: time 1682532012.682093, ----- SYN_REPORT -----
86  [ 4239.209857] USERCODE=0x1818
87  [ 4239.236861] RMC_GETDATA=fc
88  Event: time 1682532013.430863, type 1 (EV_KEY), code 5 (KEY_4), value 1
89  Event: time 1682532013.430863, ----- SYN_REPORT -----
90  Event: time 1682532013.521980, type 1 (EV_KEY), code 5 (KEY_4), value 0
91  Event: time 1682532013.521980, ----- SYN_REPORT -----

```

3.2. GPU

3.2.1. glmark2

```

1  industio@ido:~$ glmark2
2  =====
3  glmark2 2021.02
4  =====
5  OpenGL Information
6  GL_VENDOR:      Mesa
7  GL_RENDERER:    Mali-G610 (Panfrost)
8  GL_VERSION:     3.1 Mesa 25.0.7 (git-de240b8d61)
9  =====
10 ▾ [build] use-vbo=false: FPS: 1214 FrameTime: 0.824 ms
11 ▾ [build] use-vbo=true: FPS: 1214 FrameTime: 0.824 ms
12 ▾ [texture] texture-filter=nearest: FPS: 1393 FrameTime: 0.718 ms
13 ▾ [texture] texture-filter=linear: FPS: 1208 FrameTime: 0.828 ms
14 ▾ [texture] texture-filter=mipmap: FPS: 1209 FrameTime: 0.827 ms
15 ▾ [shading] shading=gouraud: FPS: 1097 FrameTime: 0.912 ms
16 ▾ [shading] shading=blinn-phong-inf: FPS: 1234 FrameTime: 0.810 ms
17 ▾ [shading] shading=phong: FPS: 1046 FrameTime: 0.956 ms
18 ▾ [shading] shading=cel: FPS: 1188 FrameTime: 0.842 ms
19 ▾ [bump] bump-render=high-poly: FPS: 896 FrameTime: 1.116 ms
20 ▾ [bump] bump-render=normals: FPS: 1486 FrameTime: 0.673 ms
21 ▾ [bump] bump-render=height: FPS: 1353 FrameTime: 0.739 ms
22 ▾ [effect2d] kernel=0,1,0;1,-4,1;0,1,0;: FPS: 996 FrameTime: 1.004 ms
23 ▾ [effect2d] kernel=1,1,1,1,1;1,1,1,1,1;1,1,1,1,1;: FPS: 659 FrameTime: 1.51
    7 ms
24 ▾ [pulsar] light=false:quads=5:texture=false: FPS: 1224 FrameTime: 0.817 ms
25 ▾ [desktop] blur-radius=5:effect=blur:passes=1:separable=true:windows=4: FPS
    : 330 FrameTime: 3.030 ms
26 ▾ [desktop] effect=shadow:windows=4: FPS: 1071 FrameTime: 0.934 ms
27 ▾ [buffer] columns=200:interleave=false:update-dispersion=0.9:update-fractio
    n=0.5:update-method=map: FPS: 452 FrameTime: 2.212 ms
28 ▾ [buffer] columns=200:interleave=false:update-dispersion=0.9:update-fractio
    n=0.5:update-method=subdata: FPS: 474 FrameTime: 2.110 ms
29 ▾ [buffer] columns=200:interleave=true:update-dispersion=0.9:update-fraction
    =0.5:update-method=map: FPS: 624 FrameTime: 1.603 ms
30 ▾ [ideas] speed=duration: FPS: 1058 FrameTime: 0.945 ms
31 ▾ [jellyfish] <default>: FPS: 1088 FrameTime: 0.919 ms
32 ▾ [terrain] <default>: FPS: 100 FrameTime: 10.000 ms
33 ▾ [shadow] <default>: FPS: 925 FrameTime: 1.081 ms
34 ▾ [refract] <default>: FPS: 304 FrameTime: 3.289 ms
35 ▾ [conditionals] fragment-steps=0:vertex-steps=0: FPS: 1350 FrameTime: 0.741
    ms
36 ▾ [conditionals] fragment-steps=5:vertex-steps=0: FPS: 1250 FrameTime: 0.800
    ms
37 ▾ [conditionals] fragment-steps=0:vertex-steps=5: FPS: 1346 FrameTime: 0.743
    ms

```

```

38 [function] fragment-complexity=low:fragment-steps=5: FPS: 1173 FrameTime:
39 0.853 ms
40 [function] fragment-complexity=medium:fragment-steps=5: FPS: 1301 FrameTim
41 e: 0.769 ms
42 [loop] fragment-loop=false:fragment-steps=5:vertex-steps=5: FPS: 1158 Fram
43 eTime: 0.864 ms
44 [loop] fragment-steps=5:fragment-uniform=false:vertex-steps=5: FPS: 1340 F
45 rameTime: 0.746 ms
[loop] fragment-steps=5:fragment-uniform=true:vertex-steps=5: FPS: 1299 Fr
ameTime: 0.770 ms
=====
glmark2 Score: 1032
=====

```

3.2.2. glmark2-es2

```

1  industio@ido:~$ glmark2-es2
2  =====
3  glmark2 2021.02
4  =====
5  OpenGL Information
6  GL_VENDOR:      Mesa
7  GL_RENDERER:    Mali-G610 (Panfrost)
8  GL_VERSION:     OpenGL ES 3.1 Mesa 25.0.7 (git-de240b8d61)
9  =====
10 ▾ [build] use-vbo=false: FPS: 1152 FrameTime: 0.868 ms
11 ▾ [build] use-vbo=true: FPS: 1241 FrameTime: 0.806 ms
12 ▾ [texture] texture-filter=nearest: FPS: 1481 FrameTime: 0.675 ms
13 ▾ [texture] texture-filter=linear: FPS: 1500 FrameTime: 0.667 ms
14 ▾ [texture] texture-filter=mipmap: FPS: 1506 FrameTime: 0.664 ms
15 ▾ [shading] shading=gouraud: FPS: 1127 FrameTime: 0.887 ms
16 ▾ [shading] shading=blinn-phong-inf: FPS: 1203 FrameTime: 0.831 ms
17 ▾ [shading] shading=phong: FPS: 1249 FrameTime: 0.801 ms
18 ▾ [shading] shading=cel: FPS: 1110 FrameTime: 0.901 ms
19 ▾ [bump] bump-render=high-poly: FPS: 951 FrameTime: 1.052 ms
20 ▾ [bump] bump-render=normals: FPS: 1569 FrameTime: 0.637 ms
21 ▾ [bump] bump-render=height: FPS: 1581 FrameTime: 0.633 ms
22 ▾ [effect2d] kernel=0,1,0;1,-4,1;0,1,0;: FPS: 1284 FrameTime: 0.779 ms
23 ▾ [effect2d] kernel=1,1,1,1,1;1,1,1,1,1;1,1,1,1,1;: FPS: 980 FrameTime: 1.02
    0 ms
24 ▾ [pulsar] light=false:quads=5:texture=false: FPS: 1487 FrameTime: 0.672 ms
25 ▾ [desktop] blur-radius=5:effect=blur:passes=1:separable=true:windows=4: FPS
    : 273 FrameTime: 3.663 ms
26 ▾ [desktop] effect=shadow:windows=4: FPS: 1091 FrameTime: 0.917 ms
27 ▾ [buffer] columns=200:interleave=false:update-dispersion=0.9:update-fractio
    n=0.5:update-method=map: FPS: 527 FrameTime: 1.898 ms
28 ▾ [buffer] columns=200:interleave=false:update-dispersion=0.9:update-fractio
    n=0.5:update-method=subdata: FPS: 536 FrameTime: 1.866 ms
29 ▾ [buffer] columns=200:interleave=true:update-dispersion=0.9:update-fraction
    =0.5:update-method=map: FPS: 592 FrameTime: 1.689 ms
30 ▾ [ideas] speed=duration: FPS: 825 FrameTime: 1.212 ms
31 ▾ [jellyfish] <default>: FPS: 1107 FrameTime: 0.903 ms
32 ▾ [terrain] <default>: FPS: 88 FrameTime: 11.364 ms
33 ▾ [shadow] <default>: FPS: 869 FrameTime: 1.151 ms
34 ▾ [refract] <default>: FPS: 331 FrameTime: 3.021 ms
35 ▾ [conditionals] fragment-steps=0:vertex-steps=0: FPS: 1357 FrameTime: 0.737
    ms
36 ▾ [conditionals] fragment-steps=5:vertex-steps=0: FPS: 1382 FrameTime: 0.724
    ms
37 ▾ [conditionals] fragment-steps=0:vertex-steps=5: FPS: 1459 FrameTime: 0.685
    ms

```

```

38 [function] fragment-complexity=low:fragment-steps=5: FPS: 1448 FrameTime:
39 0.691 ms
40 [function] fragment-complexity=medium:fragment-steps=5: FPS: 1218 FrameTim
41 e: 0.821 ms
42 [loop] fragment-loop=false:fragment-steps=5:vertex-steps=5: FPS: 1362 Fram
43 eTime: 0.734 ms
44 [loop] fragment-steps=5:fragment-uniform=false:vertex-steps=5: FPS: 1450 F
45 rameTime: 0.690 ms
[loop] fragment-steps=5:fragment-uniform=true:vertex-steps=5: FPS: 1303 Fr
ameTime: 0.767 ms
=====
                                glmark2 Score: 1110
=====

```

3.3. NPU

将rknpu2.tar.gz压缩包拷贝板子上，使用yolov5进行rknn模型转换测试：

3.3.1. 编译rknn_yolov5_demo

```

▼ Shell
1 industio@ido:~$ export GCC_COMPILER=/usr/bin/aarch64-linux-gnu
2 industio@ido:~$ tar -xvf rknn-toolkit2-master.tar.gz
3 industio@ido:~$ cd rknn-toolkit2-master/rknpu2/examples/rknn_yolov5_demo
4 industio@ido:~/rknn-toolkit2-master/rknpu2/examples/rknn_yolov5_demo$ chmod
  a+x build-linux_RK3588.sh
5 industio@ido:~/rknn-toolkit2-master/rknpu2/examples/rknn_yolov5_demo$ ./bui
  ld-linux.sh -t rk3588 -a aarch64 -b Release

```

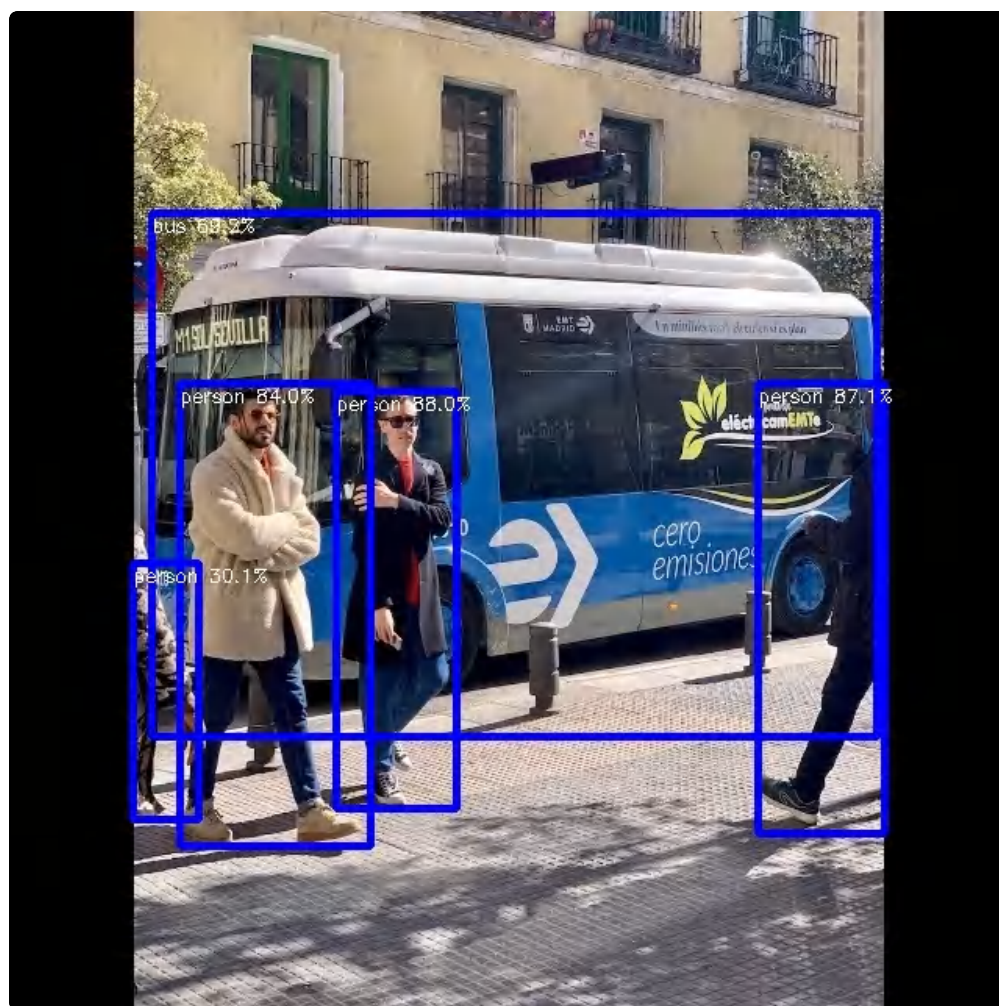
3.3.2. 测试

```

▼ Shell
1 industio@ido:~/rknn-toolkit2-master/rknpu2/examples/rknn_yolov5_demo$ cd in
  stall/rknn_yolov5_demo_Linux/
2 industio@ido:~/rknn-toolkit2-master/rknpu2/examples/rknn_yolov5_demo/instal
  l/rknn_yolov5_demo_Linux$ ./rknn_yolov5_demo ./model/RK3588/yolov5s-640-64
  0.rknn ./model/bus.jpg
3

```

执行完后会得到转换后的模型图片，如下所示：



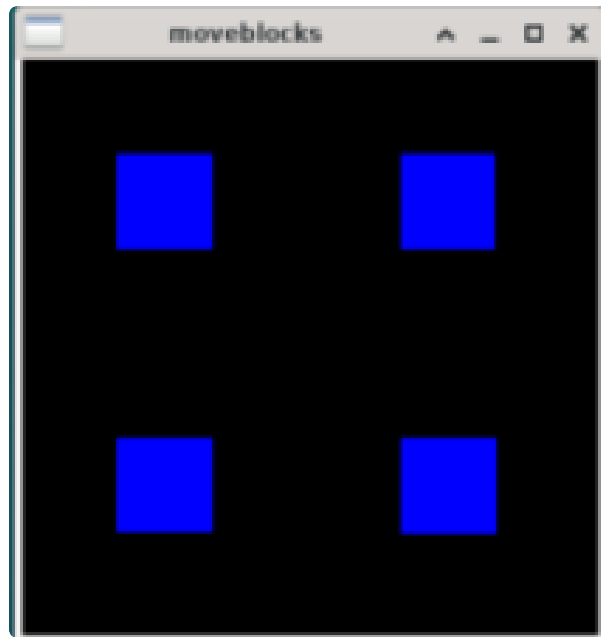
3.4. QT5

支持qt5.15，测试方法如下：

▼ Shell

```
1  industio@ido:~$ chmod a+x ./moveblocks
2  industio@ido:~$ ./moveblocks
```

结果如下所示：

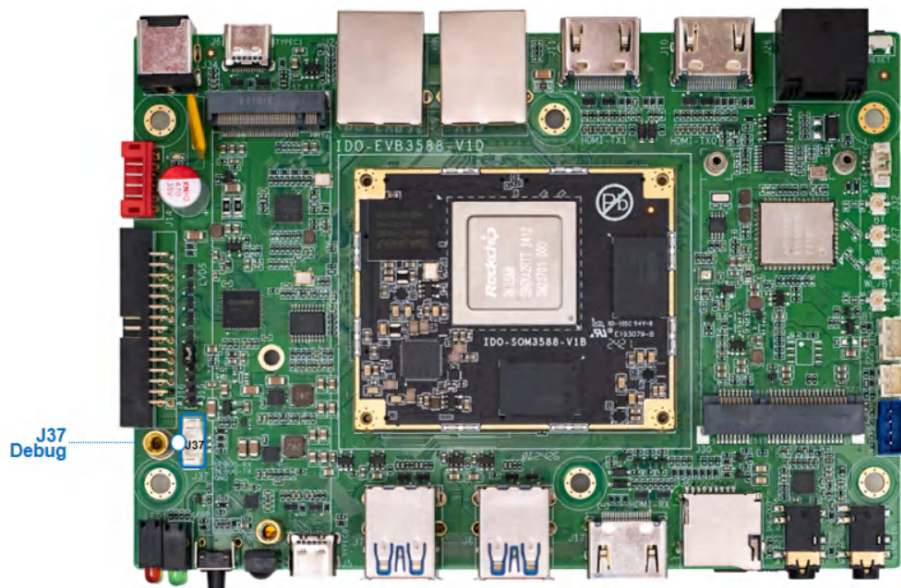


4. IDO-EVB3588-V1D-Buildroot系统

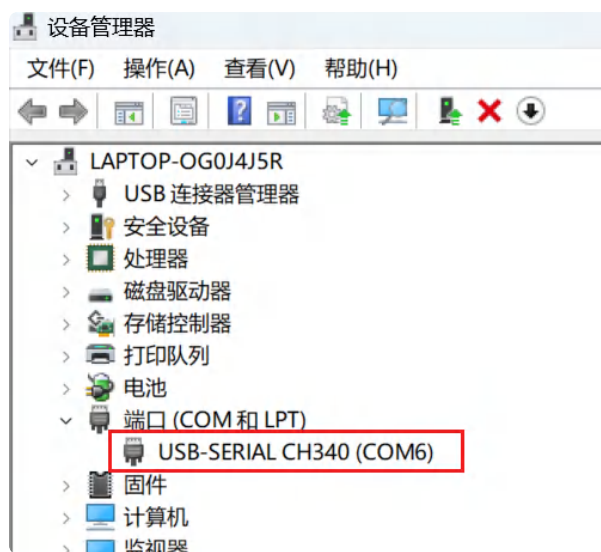
4.1. 功能及接口使用方法

4.1.1. DEBUG UART

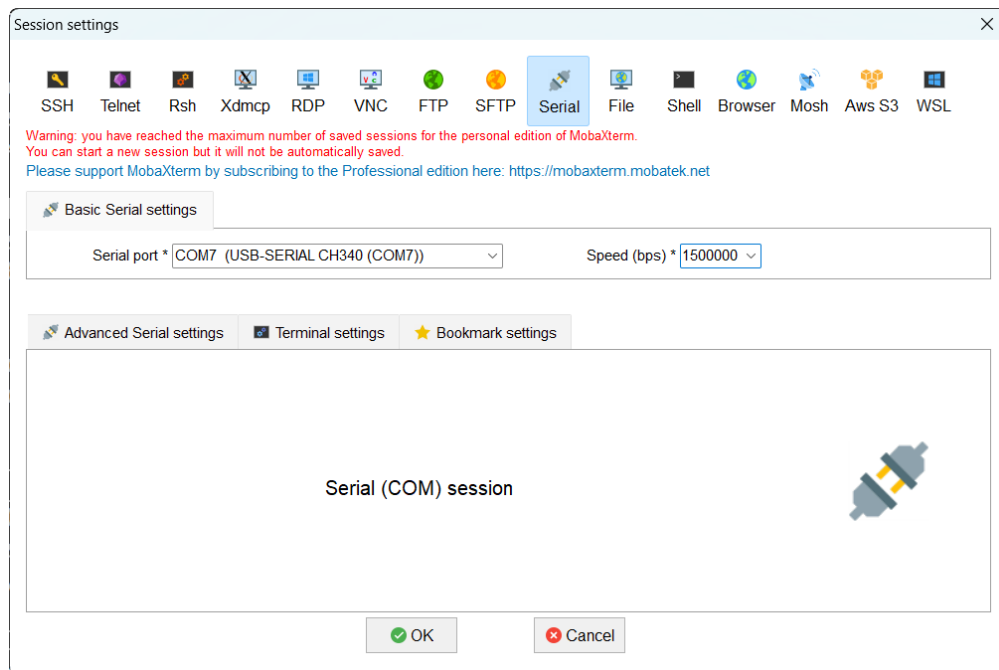
DEBUG UART位置J37，如下图所示：



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。



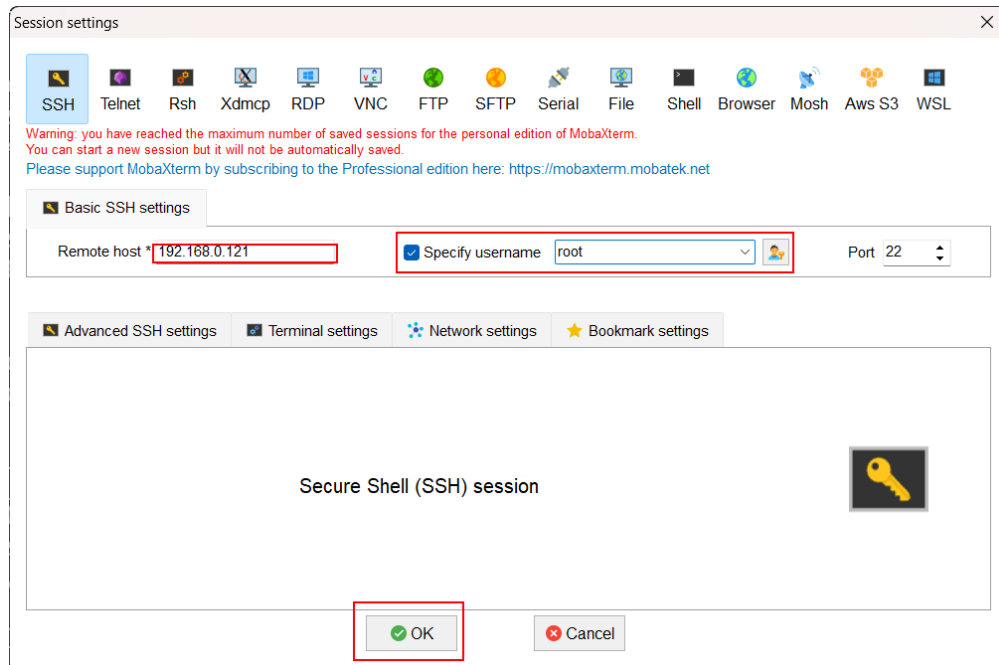
使用调试软件（MobaXterm、putty）等，以MobaXterm为例子，设置参数如下：



4.1.2. SSH登录

在知道IP的前提下，默认可以通过SSH登录进系统。

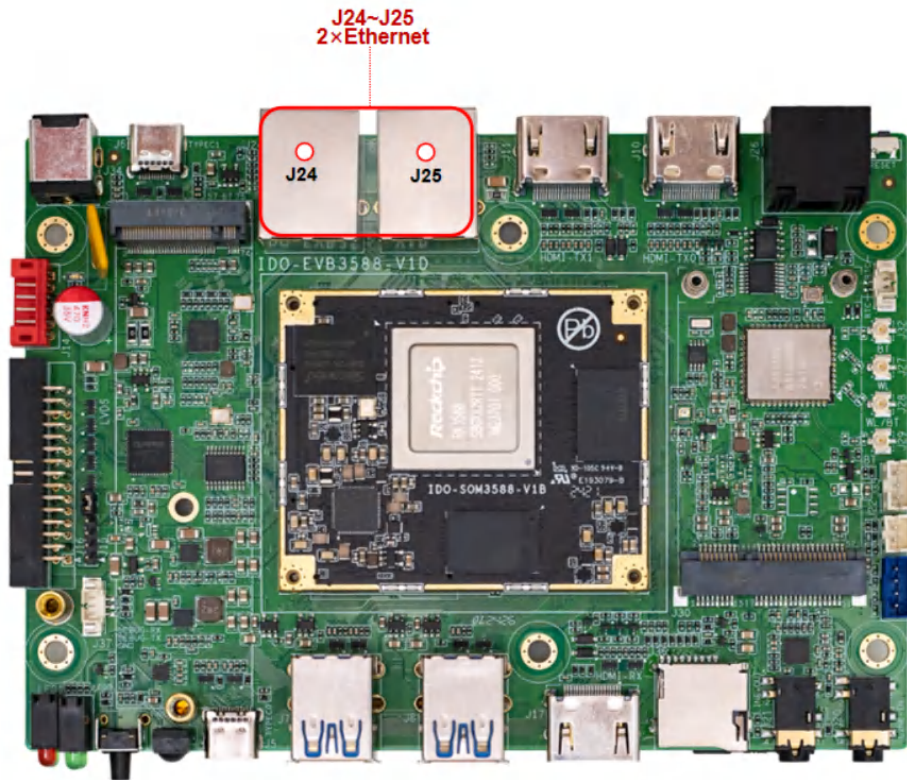
SSH登录账号密码：root/123456



4.1.3. 网络

4.1.3.1. 以太网

主板有两路千兆以太网接口位置J24,J25，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：

4.1.3.2. Ethernet动态IP设置

```
1 #eth0
2 root@rk3588-buildroot:/# ifconfig eth0
3 eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
4     inet 192.168.0.90 netmask 255.255.255.0 broadcast 192.168.0.255
5     inet6 fe80::1b83:a156:cb16:c6b0 prefixlen 64 scopeid 0x20<link>
6     ether fa:c6:ba:c7:0c:60 txqueuelen 1000 (Ethernet)
7     RX packets 141 bytes 15129 (14.7 KiB)
8     RX errors 0 dropped 4 overruns 0 frame 0
9     TX packets 89 bytes 9306 (9.0 KiB)
10    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
11    device interrupt 121
12
13 #eth1
14 root@rk3588-buildroot:/# ifconfig eth1
15 eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
16     inet 192.168.0.91 netmask 255.255.255.0 broadcast 192.168.0.255
17     inet6 fe80::8992:70c3:f104:e0be prefixlen 64 scopeid 0x20<link>
18     ether fe:c6:ba:c7:0c:60 txqueuelen 1000 (Ethernet)
19     RX packets 293 bytes 30522 (29.8 KiB)
20     RX errors 0 dropped 6 overruns 0 frame 0
21     TX packets 151 bytes 15448 (15.0 KiB)
22     TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
23     device interrupt 123
```

4.1.3.3. Ethernet静态IP设置

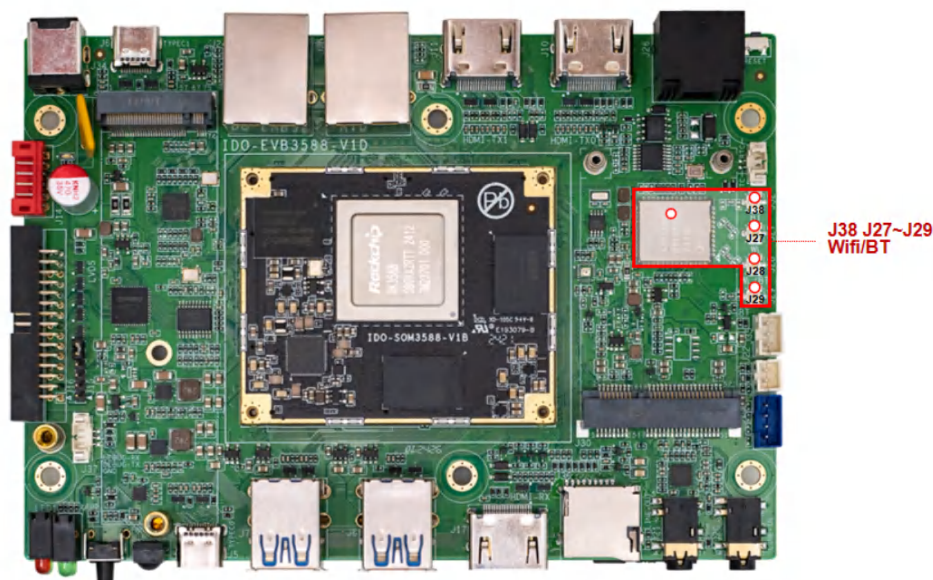
```
1 root@rk3588-buildroot:/# vim /etc/network/interfaces
2 # interface file auto-generated by buildroot
3
4 auto lo
5 iface lo inet loopback
6
7
8 auto eth1
9     iface eth1 inet static
10     address 192.168.2.123
11     netmask 255.255.255.0
12     gateway 192.168.2.1
13     nameserver 192.168.2.1
14
15 auto eth0
16     iface eth0 inet static
17     address 192.168.1.123
18     netmask 255.255.255.0
19     gateway 192.168.1.1
20     nameserver 192.168.1.1
21
22 source-directory /etc/network/interfaces.d
23
```

重启生效

```
1
2 root@rk3588-buildroot:/# ifconfig eth0
3 eth0      Link encap:Ethernet  HWaddr 0E:DC:F0:57:6F:9F
4           inet addr:192.168.1.123  Bcast:0.0.0.0  Mask:255.255.255.0
5           inet6 addr: fe80::fae:beb3:d0bf:316a/64 Scope:Link
6           UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
7           RX packets:897 errors:0 dropped:0 overruns:0 frame:0
8           TX packets:47 errors:0 dropped:0 overruns:0 carrier:0
9           collisions:0 txqueuelen:1000
10          RX bytes:104459 (102.0 KiB)  TX bytes:3694 (3.6 KiB)
11          Interrupt:95
12
13 root@rk3588-buildroot:/#
14 root@rk3588-buildroot:/# ifconfig eth1
15 eth1      Link encap:Ethernet  HWaddr FE:B7:D4:43:6E:4B
16           inet addr:192.168.2.123  Bcast:0.0.0.0  Mask:255.255.255.0
17           UP BROADCAST MULTICAST  MTU:1500  Metric:1
18           RX packets:0 errors:0 dropped:0 overruns:0 frame:0
19           TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
20           collisions:0 txqueuelen:1000
21           RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
22          Interrupt:172 Base address:0x5000
```

4.1.4. 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：



添加WiFi账号密码:

▼ Shell |

```
1 root@rk3588-buildroot:/# vim /etc/wpa_supplicant.conf
2 ctrl_interface=/var/run/wpa_supplicant
3 ap_scan=1
4
5 network={
6     ssid="SSID"
7     psk="PASSWORD"
8     key_mgmt=WPA-PSK
9 }
10
11 #ssid: 账号
12 #psk: 密码
```

扫描附近WiFi:

▼ Shell |

```
1 root@rk3588-buildroot:/# iw dev wlan0 scan | grep SSID
```

WiFi连接:

▼ Shell |

```
1 root@rk3588-buildroot:/# wpa_supplicant -Dnl80211 -c /etc/wpa_supplicant.conf -i wlan0 &
```

查看wlan0的IP地址, 确认连接成功:

```
1 root@rk3588-buildroot:/# ifconfig wlan0
2 wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
3     inet 192.168.0.118 netmask 255.255.255.0 broadcast 192.168.0.255
4     inet6 fe80::1036:80b4:5082:7440 prefixlen 64 scopeid 0x20<link>
5     ether c0:f5:35:12:ad:a2 txqueuelen 1000 (Ethernet)
6     RX packets 170 bytes 22149 (21.6 KiB)
7     RX errors 0 dropped 5 overruns 0 frame 0
8     TX packets 24 bytes 2683 (2.6 KiB)
9     TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
10
11 root@rk3588-buildroot:/# ping www.baidu.com -I wlan0
12 PING www.wshifen.com (103.235.47.188) from 192.168.0.118 wlan0: 56(84) bytes of data.
13 64 bytes from 103.235.47.188 (103.235.47.188): icmp_seq=1 ttl=45 time=274 ms
14 64 bytes from 103.235.47.188 (103.235.47.188): icmp_seq=2 ttl=45 time=299 ms
15 64 bytes from 103.235.47.188 (103.235.47.188): icmp_seq=3 ttl=45 time=320 ms
16 64 bytes from 103.235.47.188 (103.235.47.188): icmp_seq=4 ttl=45 time=241 ms
```

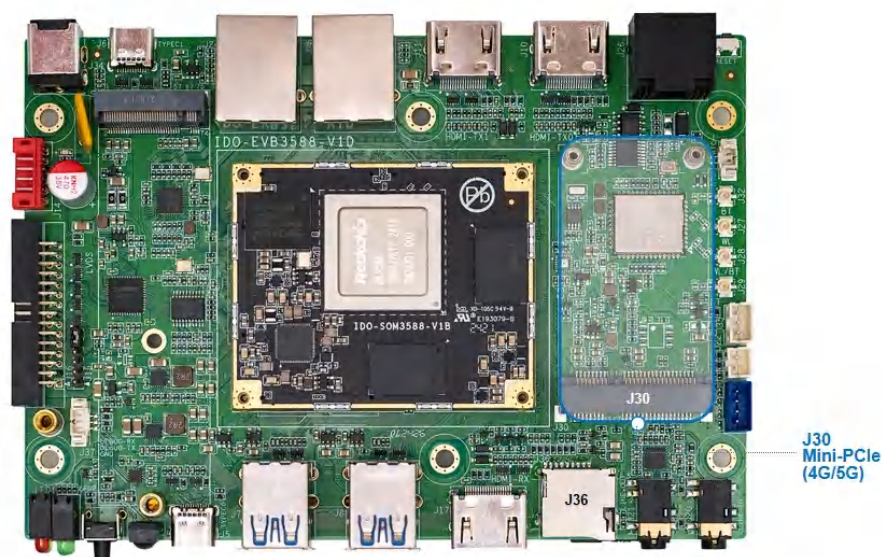
4.1.5. Bluetooth

4.1.5.1. 连接方法

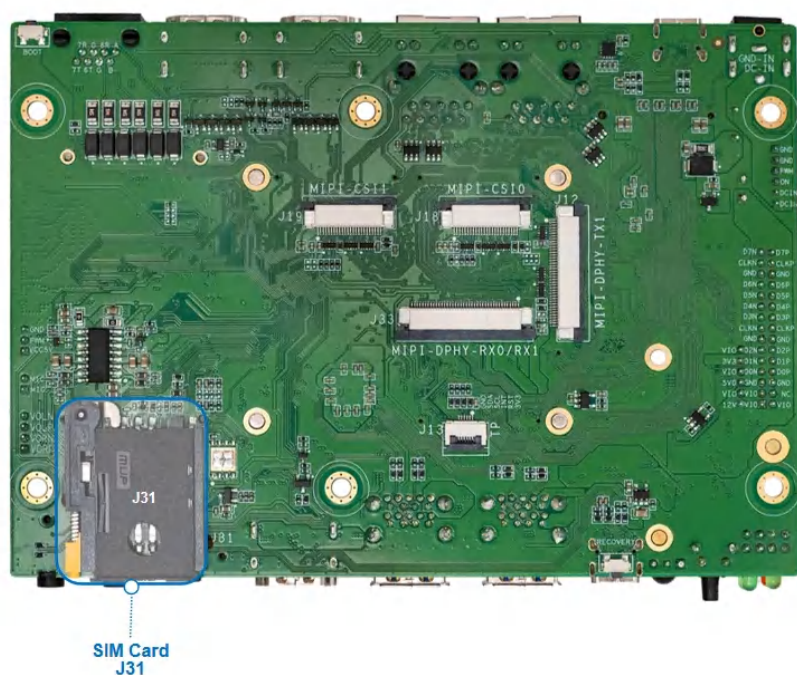
```
1 #打开蓝牙
2 root@rk3588-buildroot:/# bluetoothctl
3 [bluetooth]# power on
4 #扫描蓝牙
5 [bluetooth]# scan on
6 #查看蓝牙设备
7 [bluetooth]# devices
8 #信任蓝牙设备
9 [bluetooth]# trust AC:C4:BD:6C:F2:10
10 #蓝牙配对
11 [bluetooth]# pair AC:C4:BD:6C:F2:10
12 #连接蓝牙
13 [bluetooth]# connect AC:C4:BD:6C:F2:10
```

4.1.6. 4G/5G

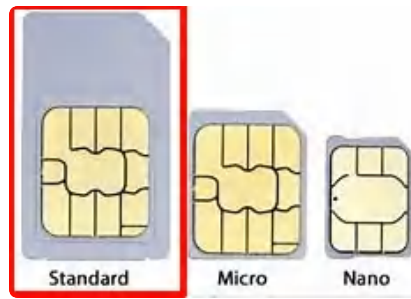
主板内置Mini PCIe 座子扩展 4G/5G模块，4G通信模块适配移远EC20、EC25等通用模组。5G通信模块适配移远RG200U-CN。模块安装位J30，如下图所示：



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J31位于主板背面，如下图所示：



使用Micro尺寸SIM卡，如下图所示：



4G使用 `ifconfig wwan0` 命令可查看到相关网络信息:

Plain Text

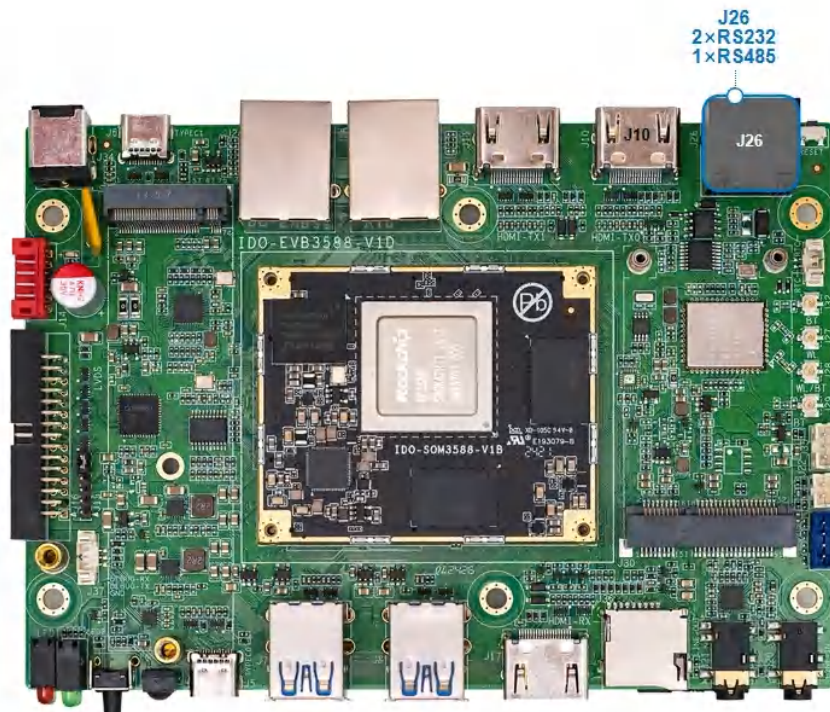
```
1  #拨号
2  root@rk3588-buildroot:/# quectel-CM &
3
4  #查看网络
5  root@rk3588-buildroot:/# ifconfig wwan0
6  wwan0: flags=193<UP,RUNNING,NOARP> mtu 1500
7      inet 10.101.61.51 netmask 255.255.255.248
8      inet6 fe80::fc:f6ff:fe8d:bab6 prefixlen 64 scopeid 0x20<link>
9      ether 02:fc:f6:8d:ba:b6 txqueuelen 1000 (Ethernet)
10     RX packets 42 bytes 7013 (6.8 KiB)
11     RX errors 0 dropped 0 overruns 0 frame 0
12     TX packets 57 bytes 4608 (4.5 KiB)
13     TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
14
15  #测试通信
16  root@rk3588-buildroot:/# ping www.baidu.com -I wwan0
```

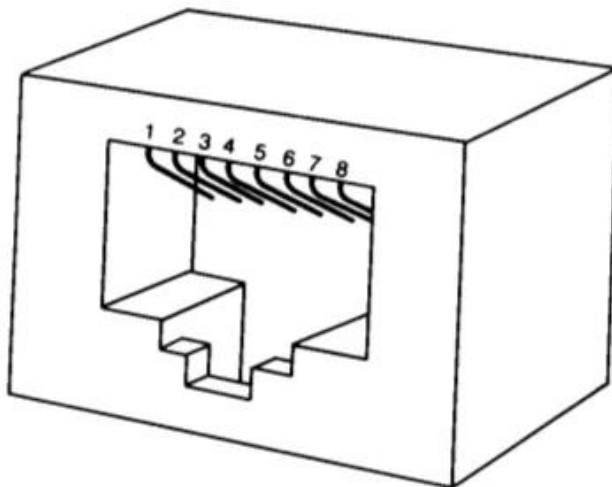
5G使用 `ifconfig usb0` 命令可查看到相关网络信息:

```
1 #拨号
2 root@rk3588-buildroot:/# quectel-CM &
3
4 #查看网络
5 root@rk3588-buildroot:/# ifconfig usb0
6 usb0: flags=193<UP,RUNNING,NOARP>  mtu 1500
7     inet 10.101.61.51  netmask 255.255.255.248
8     inet6 fe80::fc:f6ff:fe8d:bab6  prefixlen 64  scopeid 0x20<link>
9     ether 02:fc:f6:8d:ba:b6  txqueuelen 1000  (Ethernet)
10    RX packets 42  bytes 7013 (6.8 KiB)
11    RX errors 0  dropped 0  overruns 0  frame 0
12    TX packets 57  bytes 4608 (4.5 KiB)
13    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0
14
15 #测试通信
16 root@rk3588-buildroot:/# ping www.baidu.com -I usb0
```

4.1.7. UART

主板一共引出3路串口（不含调试串口）共用J26端口，如下图所示：





UART设备节点列表如下：

引脚序号	丝印	默认电平类型	功能	设备节点
1	A	RS485	A	/dev/ttyS4
2	B		B	
3	6R	RS232	RX	/dev/ttyS6
4	GND	GND	/	/
5	GND			
6	6T	RS232	TX	/dev/ttyS6
7	7R	RS232	RX	/dev/ttyS7
8	7T		TX	

▼

C |

```

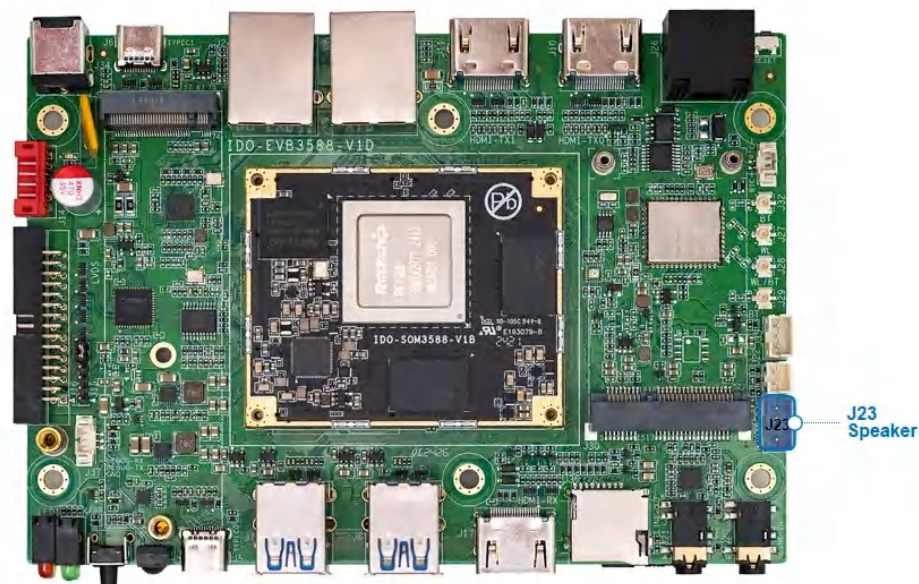
1  root@rk3588-buildroot:/# microcom /dev/ttyS4 -s 115200
2

```

4.1.8. 声音

4.1.8.1. 喇叭

喇叭接口为PH2.0–4P连接器J38，如下图所示：



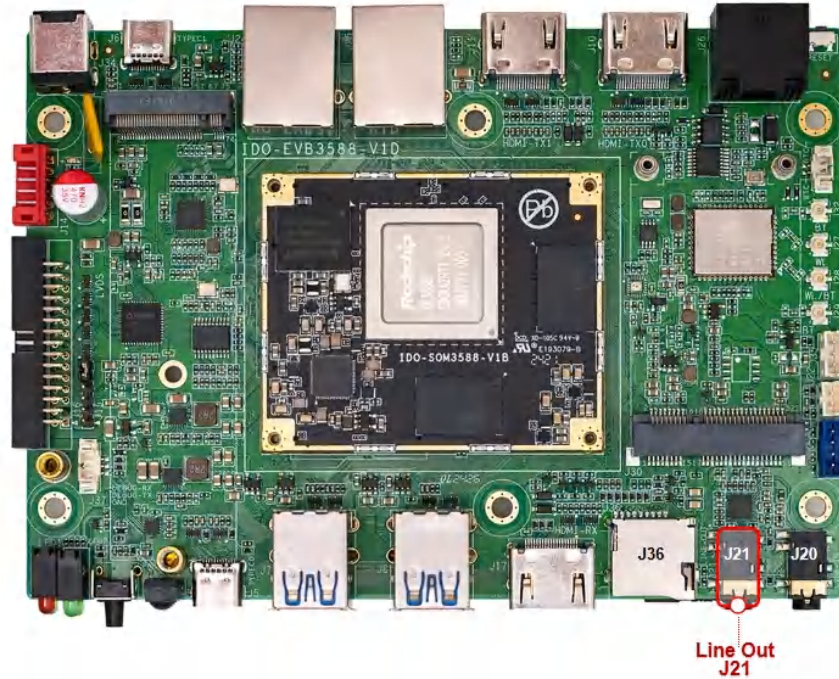
- 支持双声道，每个声道支持4Ω 3W输出

Plain Text

```
1 #查看声卡
2 root@rk3588-buildroot:/# aplay -l
3
4 #音频播放到HDMI0
5 root@rk3588-buildroot:/# aplay -D hw:0,0 xihuangni.wav
6 Playing WAVE 'xihuangni.wav' : Signed 16 bit Little Endian, Rate 44100 H
  z, Stereo
7
8 #音频播放到HDMI1
9 root@rk3588-buildroot:/# aplay -D hw:1,0 xihuangni.wav
10 Playing WAVE 'xihuangni.wav' : Signed 16 bit Little Endian, Rate 44100 H
  z, Stereo
11
12 #音频播放到DP
13 root@rk3588-buildroot:/# aplay -D hw:2,0 xihuangni.wav
14 Playing WAVE 'xihuangni.wav' : Signed 16 bit Little Endian, Rate 44100 H
  z, Stereo
15
16 #音频播放到声卡
17 root@rk3588-buildroot:/# amixer -c 3 cset numid=41 on
18
19 root@rk3588-buildroot:/# aplay -D hw:3,0 xihuangni.wav
20 Playing WAVE 'xihuangni.wav' : Signed 16 bit Little Endian, Rate 44100 H
  z, Stereo
```

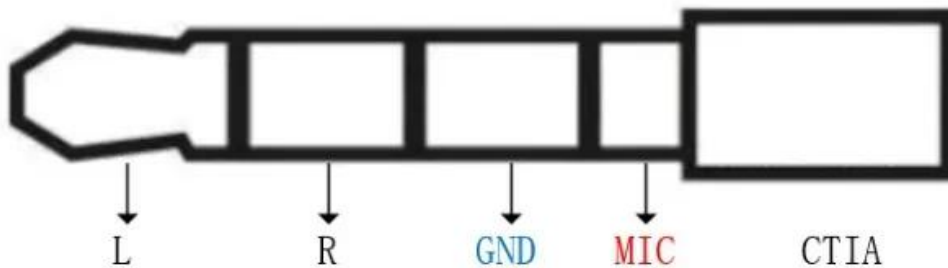
4.1.8.2. 耳机

主板支持1路3.5mm四节耳机座J21，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：

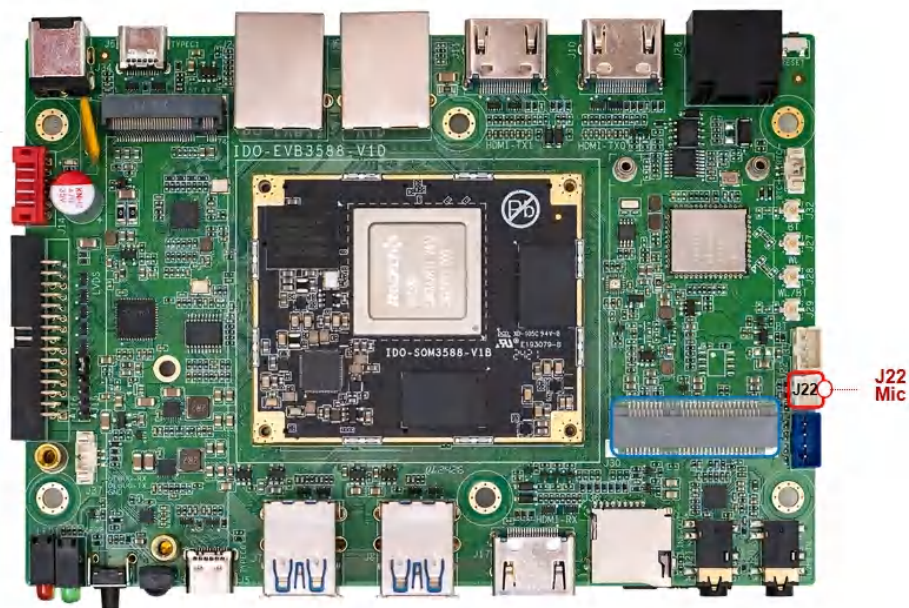


耳机录音：

```
1 #耳机录音
2 root@rk3588-buildroot:/# amixer -c 3 cset numid=43 on
3
4 root@rk3588-buildroot:/# arecord -D hw:3,0 -f cd -r 44100 -c 2 test.wav
5 Recording WAVE 'test.wav' : Signed 16 bit Little Endian, Rate 44100 Hz, Stereo
```

4.1.8.3. Mic

MX1.25T-2P麦克风接口J22，如下图所示：

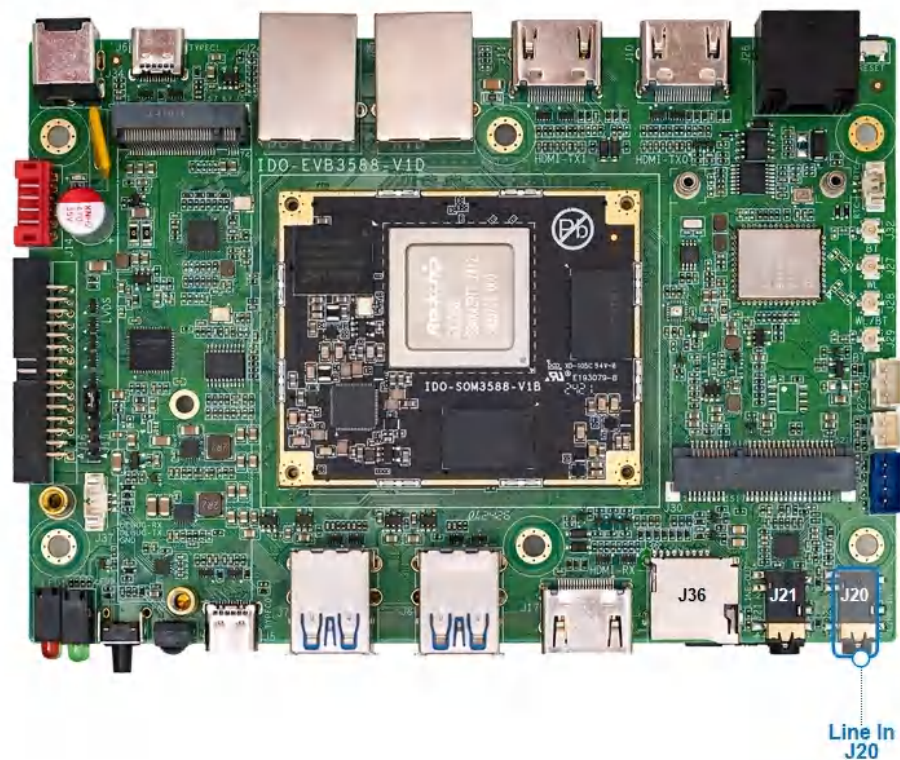


Mic录音：

```
1 #Mic录音
2 root@rk3588-buildroot:/# arecord -D hw:3,0 -f S16_LE -r 44100 -c 2 test.wav
3 Recording WAVE 'test.wav' : Signed 16 bit Little Endian, Rate 44100 Hz, Stereo
```

4.1.8.4. Line-in

主板支持1路3.5mm 三节音频Line-in接口J20，如下图所示：

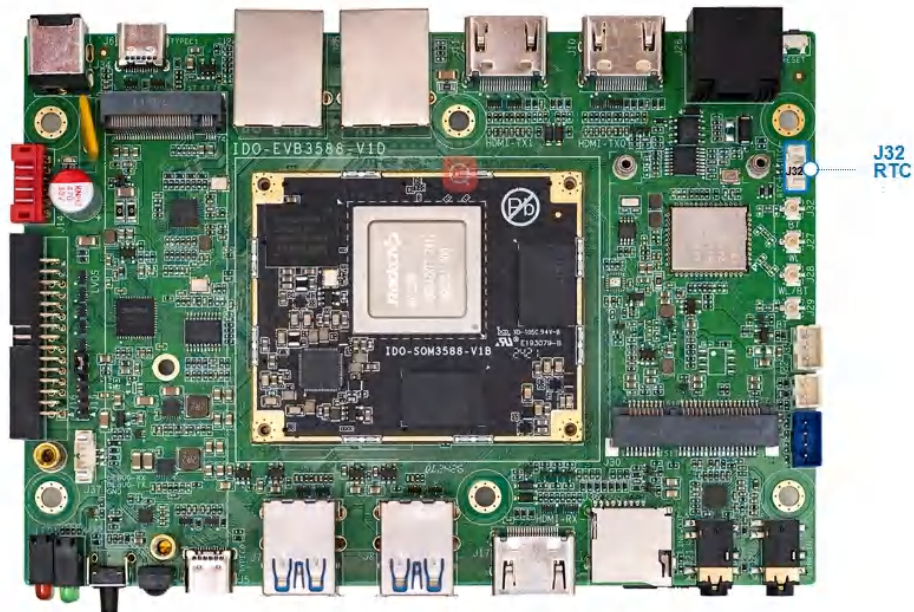


Plain Text

```
1 #切换到line in
2 root@rk3588-buildroot:/# echo -n "mic1" > /sys/class/es8388_mic/mic_channel
3
4 #录音
5 root@rk3588-buildroot:/# arecord -D hw:3,0 -f cd -r 44100 -c 2 test.wav
6 Recording WAVE 'test.wav' : Signed 16 bit Little Endian, Rate 44100 Hz, Stereo
```

4.1.9. 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：

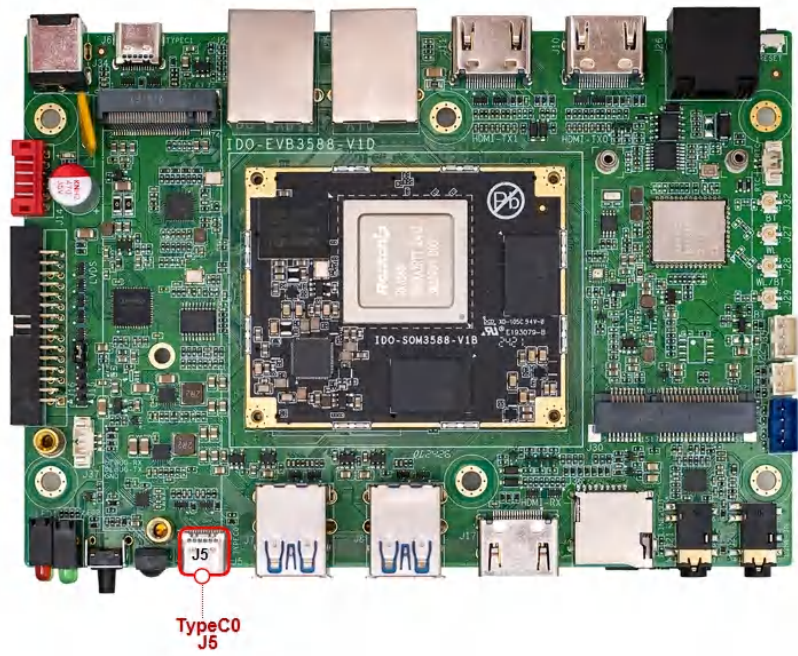
```
1  #设置时间
2  root@rk3588-buildroot:/# date -s "2025-10-10 10:10:10"
3
4  #将rtc时钟调整为与目前的系统时间一致
5  root@rk3588-buildroot:/# hwclock -w
6
7  #获取硬件rtc当前时间（断电重启读取时间没有太大偏差）
8  root@rk3588-buildroot:/# hwclock -r
9  Fri Oct 10 10:10:42 2025  0.000000 seconds
```

4.1.10. USB接口

主板支持2个TypeC接口，支持4个USB3.0-A接口，USB对外总供电应小于3A。

4.1.10.1. TypeC0 接口

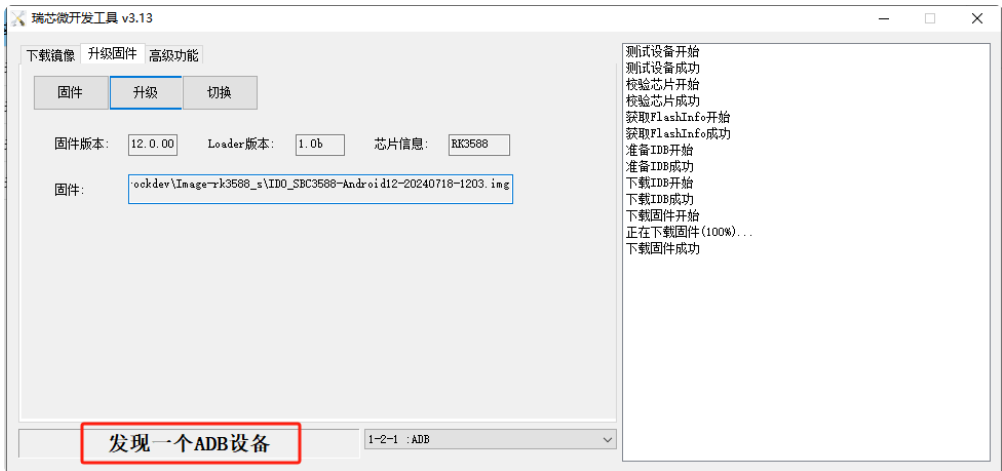
TypeC0接口（USB3 Gen1 OTG）J5，如下图所示：



- 支持Host、Device模式自动切换
- 支持DP显示输出

4.1.10.1.1. 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

4.1.10.1.2. 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

4.1.10.1.3. 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

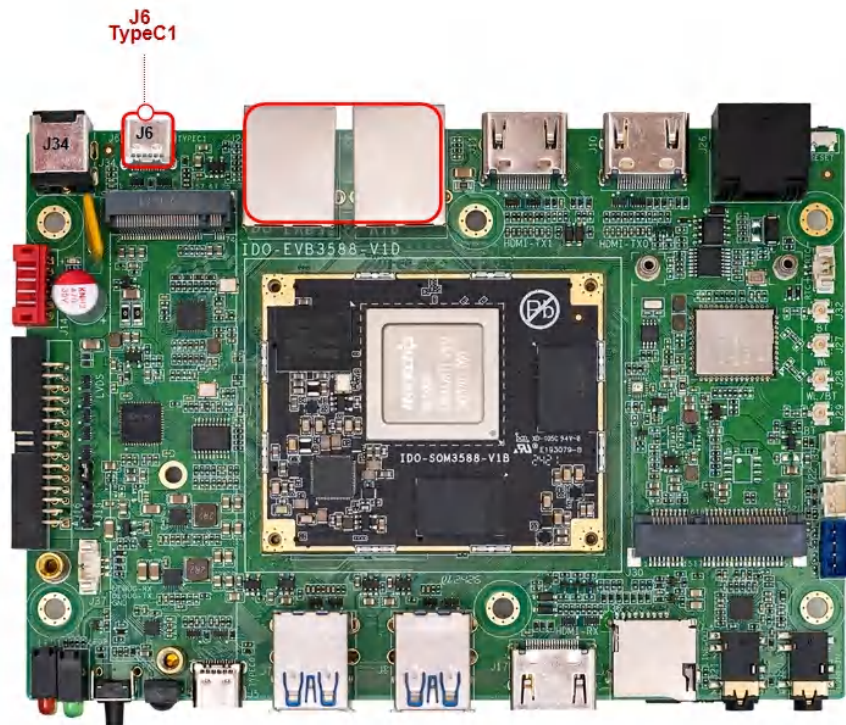


- 主板画面通过TYPE-C DP输出正常，DP声音输出正常

注意：固件默认关闭DP模式

4.1.10.2. TypeC1 接口

TypeC0接口（USB3 Gen1）J6，如下图所示：



- 支持Host模式
- 支持DP显示输出

4.1.10.2.1. 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

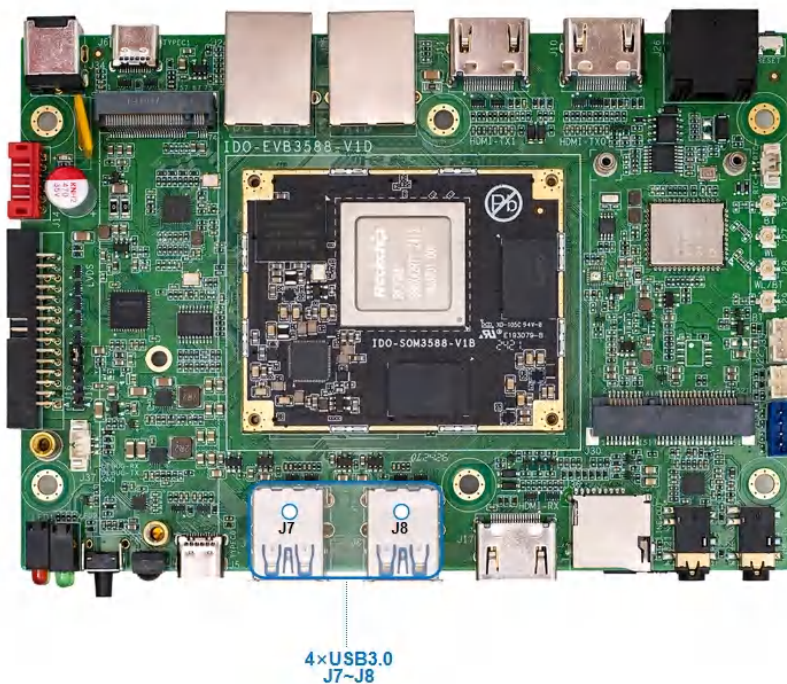
4.1.10.2.2. DP模式

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

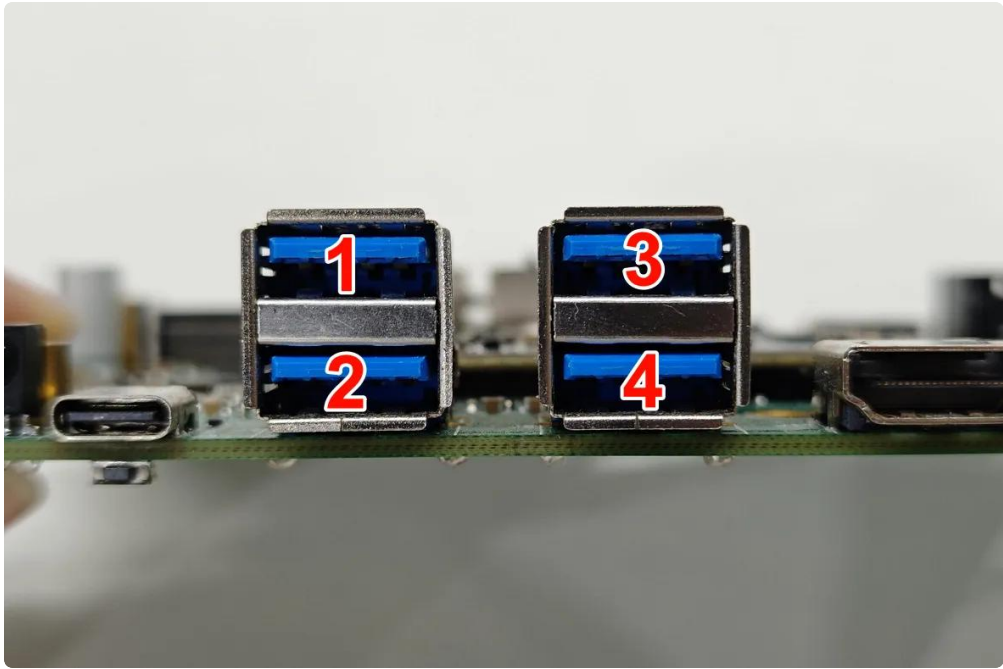


4.1.10.3. USB3.0接口

主板支持4个USB3.0接口，接口为标准的双层A口J7、J8，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：



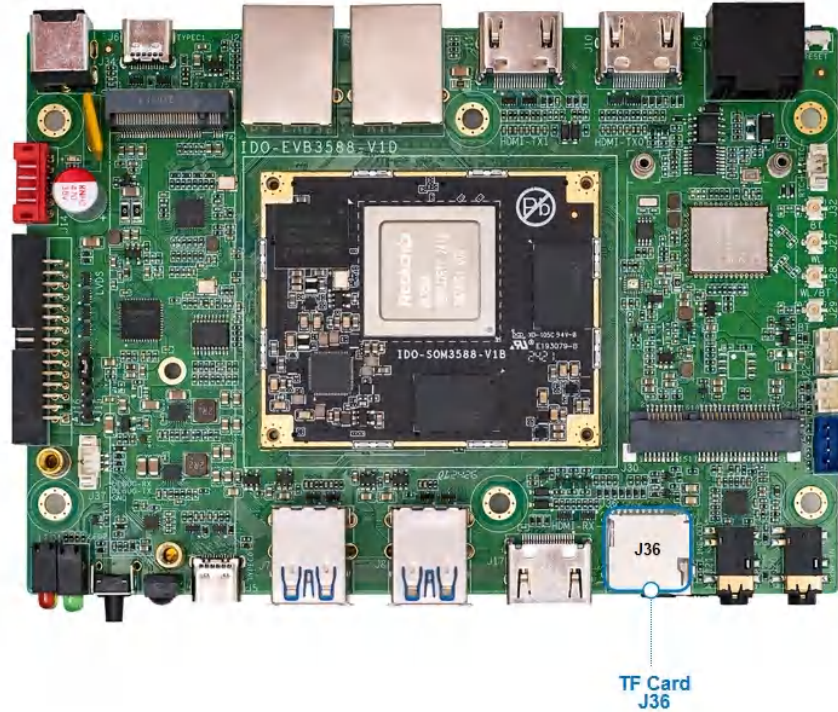
USB电源控制，如下表所示：

USB端口	默认状态	动作	命令
USB1	开启	关闭电源	echo 0 > /sys/class/leds/usb_host1_pwr/brightness
		开启电源	echo 1 > /sys/class/leds/usb_host1_pwr/brightness
USB2	开启	关闭电源	echo 0 > /sys/class/leds/usb_host2_pwr/brightness
		开启电源	echo 1 > /sys/class/leds/usb_host2_pwr/brightness
USB3	开启	关闭电源	echo 0 > /sys/class/leds/usb_host3_pwr/brightness
		开启电源	echo 1 > /sys/class/leds/usb_host3_pwr/brightness
USB4	开启	关闭电源	echo 0 > /sys/class/leds/usb_host4_pwr/brightness
		开启电源	echo 1 > /sys/class/leds/usb_host4_pwr/brightness

供电控制说明：设备节点写"0"关闭电源，写"1"开启电源。

4.1.11. TF Card

TF卡座支持SDIO3.0，支持高速SD卡，接口位于J36，如下图所示：

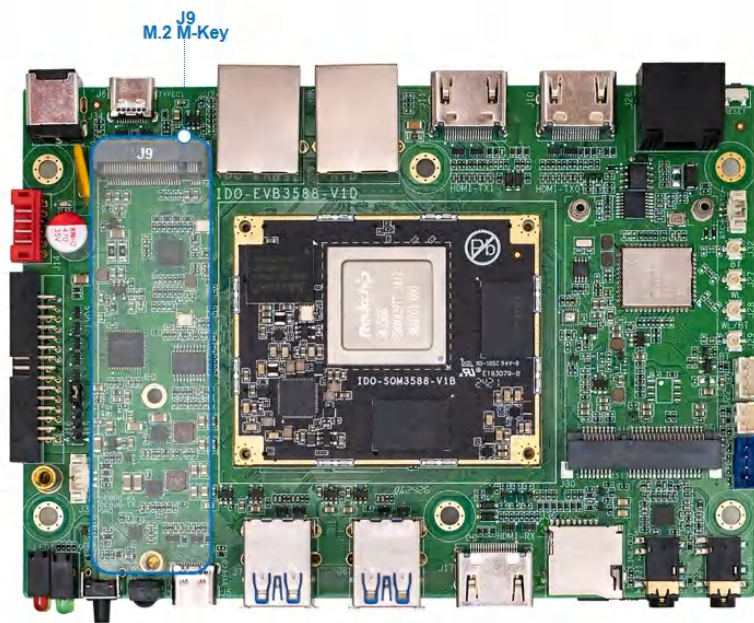


Shell

```
1 root@rk3588-buildroot:/# fdisk -l
2 Disk /dev/ram0: 8 MiB, 8388608 bytes, 16384 sectors
3 Units: sectors of 1 * 512 = 512 bytes
4 Sector size (logical/physical): 512 bytes / 4096 bytes
5 I/O size (minimum/optimal): 4096 bytes / 4096 bytes
6 .....
7
8 Device          Boot Start      End Sectors  Size Id Type
9 /dev/mmcblk1p1    135 3858623 3858489   1.8G  e W95 FAT16 (LBA)
10
11 root@rk3588-buildroot:/# df -h
12 文件系统      大小  已用  可用 已用% 挂载点
13 /dev/root      53G   6.5G   45G   13% /
14 devtmpfs       4.0M   8.0K   4.0M    1% /dev
15 ...
16 /dev/mmcblk1p1  1.9G   768K   1.9G    1% /mnt/sdcard
```

4.1.12. M.2接口

主板上使用标准M.2-M-key M.2接口连接座J9，如下图所示：



支持PCIe2.1通信，适用2280尺寸NVME固态硬盘。

```

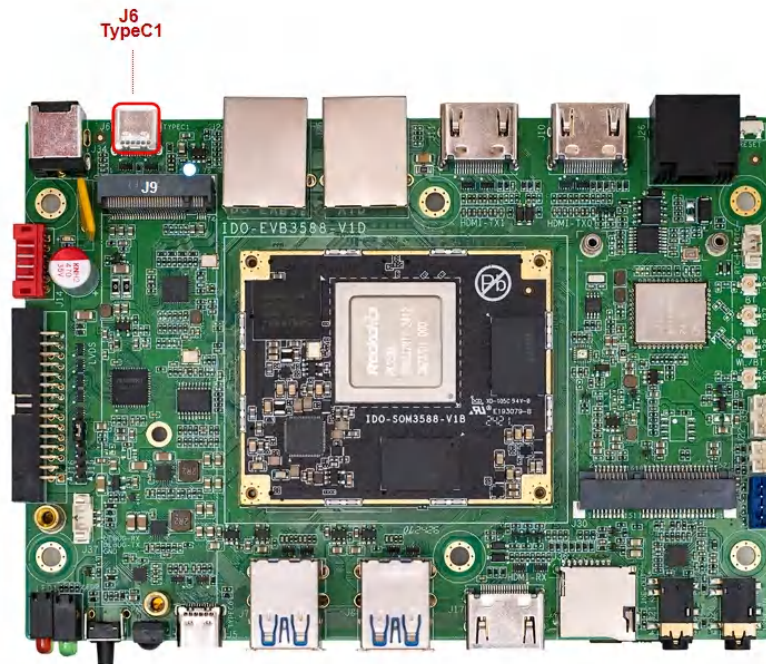
1  root@rk3588-buildroot:/# fdisk -l
2  Disk /dev/ram0: 8 MiB, 8388608 bytes, 16384 sectors
3  Units: sectors of 1 * 512 = 512 bytes
4  Sector size (logical/physical): 512 bytes / 4096 bytes
5  I/O size (minimum/optimal): 4096 bytes / 4096 bytes
6
7
8  Disk /dev/ram1: 8 MiB, 8388608 bytes, 16384 sectors
9  Units: sectors of 1 * 512 = 512 bytes
10 Sector size (logical/physical): 512 bytes / 4096 bytes
11 I/O size (minimum/optimal): 4096 bytes / 4096 bytes
12 ...
13 Disk /dev/nvme0n1: 476.94 GiB, 512110190592 bytes, 1000215216 sectors
14 Disk model: XPG GAMMIX S11L
15 Units: sectors of 1 * 512 = 512 bytes
16 Sector size (logical/physical): 512 bytes / 512 bytes
17 I/O size (minimum/optimal): 512 bytes / 512 bytes
18 Disklabel type: dos
19 Disk identifier: 0x00000000
20
21 Device          Boot Start          End      Sectors   Size Id Type
22 /dev/nvme0n1p1    2048 1000215182 1000213135 476.9G  c W95 FAT32 (LBA)
23
24 root@rk3588-buildroot:/# df -h
25 文件系统      大小  已用  可用  已用% 挂载点
26 /dev/root      53G   6.5G   45G   13% /
27 devtmpfs       4.0M   8.0K   4.0M    1% /dev
28 ...
29 /dev/nvme0n1p1 477G   2.1G  475G    1% /mnt/m2

```

4.1.13. LCD显示

4.1.13.1. DP

DP接口 (TYPE C1) J6, 如下图所示:



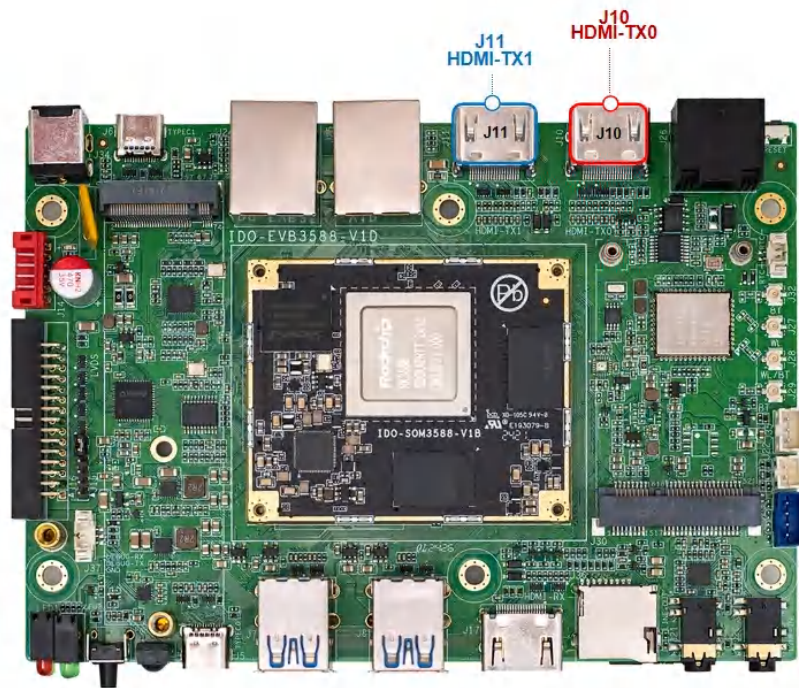
可以使用 USB-C 转 HDMI 高清线连接 HDMI 显示器输出画面，如下图所示：



- 最高支持8K@30fps输出

4.1.13.2. HDMI-TX

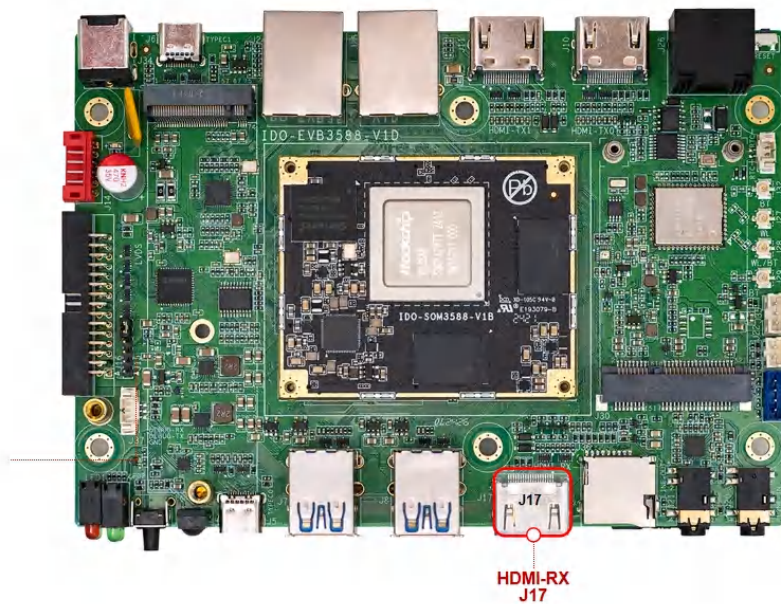
主板支持2路标准HDMI-A接口，J10、J11，如下图所示：



- 最高支持 HDMI2.1 8K@60fps 输出

4.1.13.3. HDMI-RX接口

主板支持1路标准HDMI-A接口的HDMI-RX接口J17，如下图所示：



查看输入hdm信号格式：

```

1 root@rk3588-buildroot:/# v4l2-ctl -d /dev/video40 --query-dv-timings
2   Active width: 3840
3   Active height: 2160
4   Total width: 4400
5   Total height: 2250
6   Frame format: progressive
7   Polarities: -vsync -hsync
8   Pixelclock: 593992000 Hz (60.00 frames per second)
9   Horizontal frontporch: 176
10  Horizontal sync: 88
11  Horizontal backporch: 296
12  Vertical frontporch: 8
13  Vertical sync: 10
14  Vertical backporch: 72
15  Standards:
16  Flags:
17
18 root@rk3588-buildroot:/# v4l2-ctl -d /dev/video40 --get-fmt-video
19 Format Video Capture Multiplanar:
20   Width/Height      : 3840/2160
21   Pixel Format       : 'NV12' (Y/CbCr 4:2:0)
22   Field              : None
23   Number of planes  : 1
24   Flags              : premultiplied-alpha, set-csc, 0x000000fc
25   Colourspace        : Rec. 709
26   Transfer Function  : Unknown (0x000000b8)
27   YCbCr/HSV Encoding: xvYCC 709
28   Quantization       : Limited Range
29   Plane 0            :
30     Bytes per Line   : 3840
31     Size Image       : 12441600

```

4.1.13.3.1. 预览

```

1 root@rk3588-buildroot:/# gst-launch-1.0 v4l2src device=/dev/video40 ! vide
o/x-raw, format=NV12, width=3840, height=2160, framerate=30/1 ! videoconver
t ! autovideosink

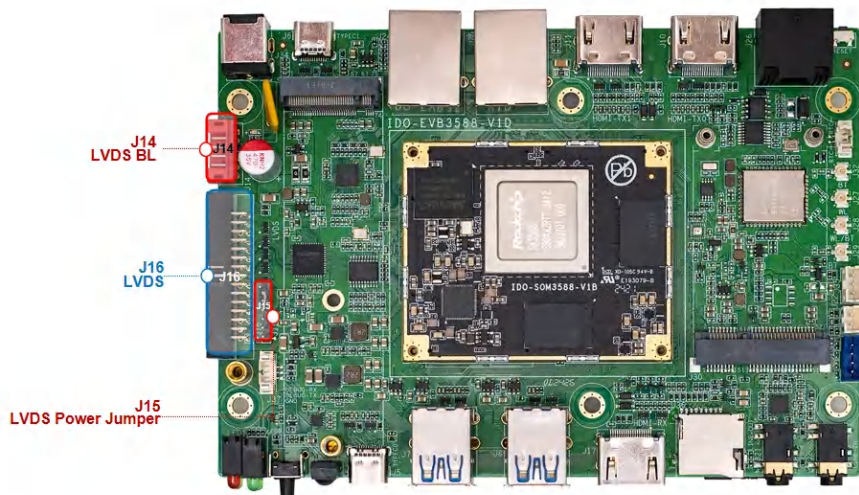
```

- HDMI2.0-RX, 支持4K@30fps

注意：hdmi in预览时需要先确认好分辨率和预览格式。

4.1.13.4. Dual LVDS屏

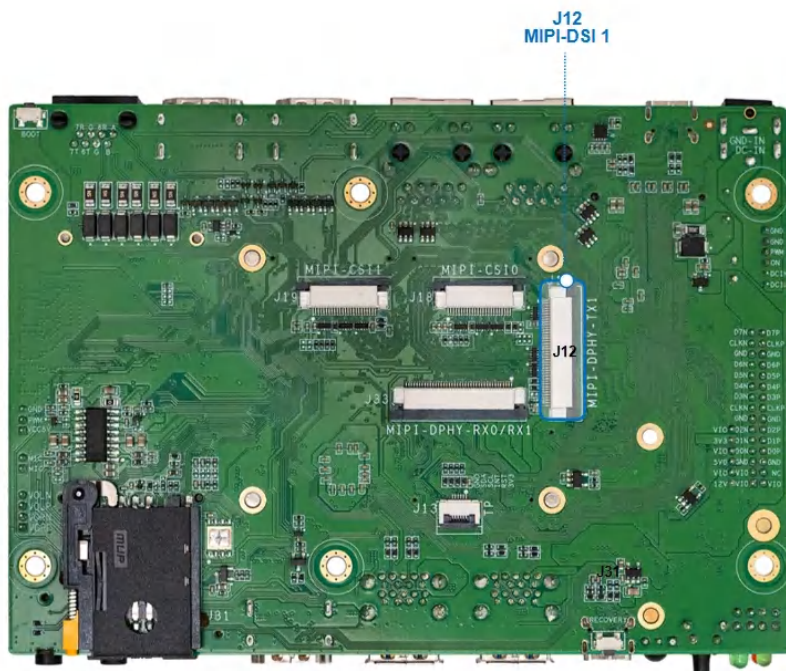
Dual LVDS接口J16，背光接 J14，屏幕供电电压选择接口J18，如下图所示：



J18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供电跳线帽接到3.3V、5V或12V，默认3.3V。

4.1.13.5. MIPI

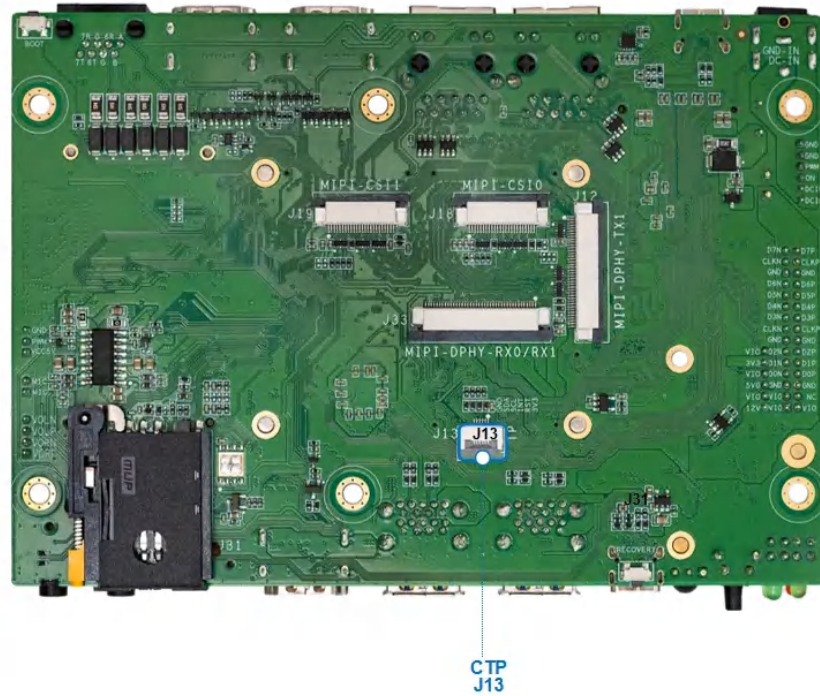
MIPI接口J12 位于主板背面，40Pin FPC 0.5mm 上接座子，如下图所示：



- MIPI供电为3.3V

4.1.14. TP接口

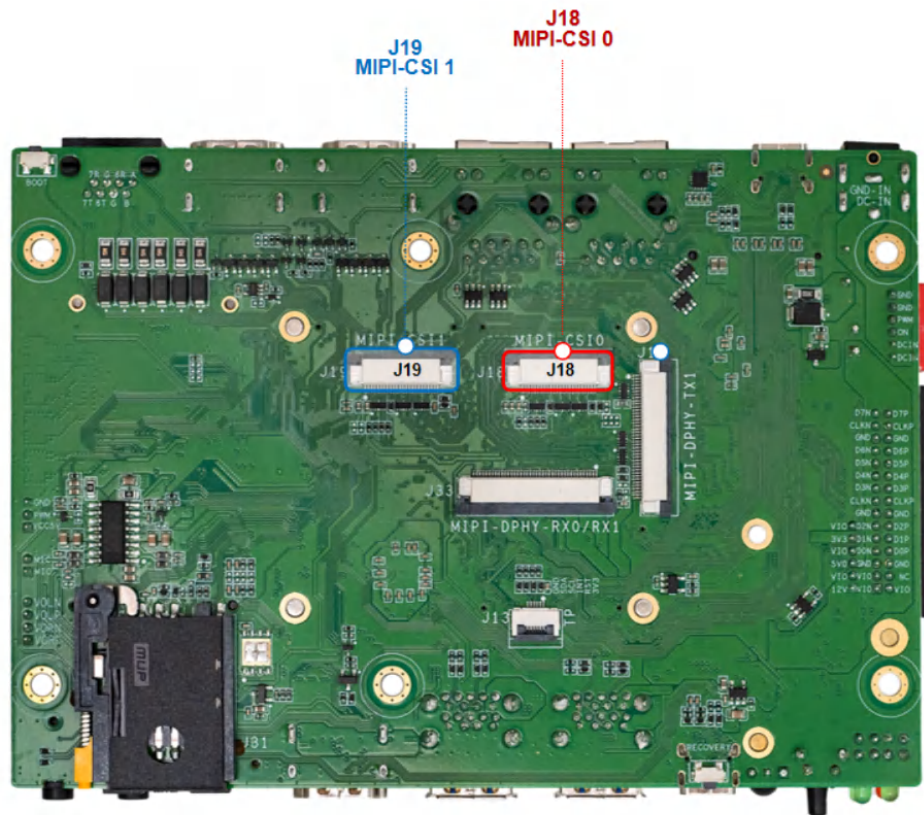
TP接口J13位于主板背面， 6Pin FPC 0.5mm座子， 如下图所示：



- 触摸 TP 接口接线方法为下接

4.1.15. MIPI Camera

MIPI Camer接口J18和J19， 如下图所示：



- MIPI Camera接口接线方法为上接

4.1.15.1. MIPI CSI0

4.1.15.1.1. 预览命令

```
1 root@rk3588-buildroot:/# export DISPLAY=:0
2 root@rk3588-buildroot:/# gst-launch-1.0 v4l2src device=/dev/video22 ! vide
o/x-raw, format=NV12, width=1920, height=1080, framerate=30/1 ! videoconver
t ! autovideosink
```

4.1.15.1.2. 抓图

```
1 root@rk3588-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
```

4.1.15.1.3. 抓视频

```
1 root@rk3588-buildroot:/# v4l2-ctl --stream-mmap=4 -d /dev/video22 --set-fm
t-video=width=1920,height=1080,pixelformat='NV12' --stream-to=./output.yuv
```

4.1.15.2. MIPI CSI1

4.1.15.2.1. 预览命令

```
1 root@rk3588-buildroot:/# export DISPLAY=:0
2 root@rk3588-buildroot:/# gst-launch-1.0 v4l2src device=/dev/video31 ! vide
o/x-raw, format=NV12, width=1920, height=1080, framerate=30/1 ! videoconver
t ! autovideosink
```

4.1.15.2.2. 抓图

▼ Plain Text |

```
1 root@rk3588-buildroot:/# v4l2-ctl --verbose -d /dev/video31 --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
```

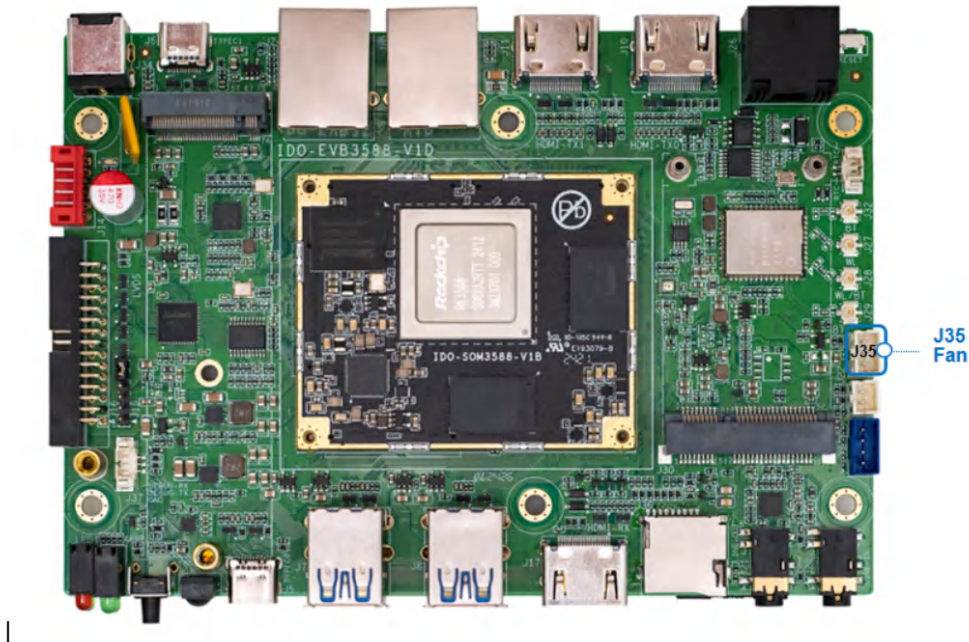
4.1.15.2.3. 抓视频

▼ Plain Text |

```
1 root@rk3588-buildroot:/# v4l2-ctl --stream-mmap=4 -d /dev/video31 --set-fmt-video=width=1920,height=1080,pixelformat='NV12' --stream-to=./output.yuv
```

4.1.16. FAN 风扇

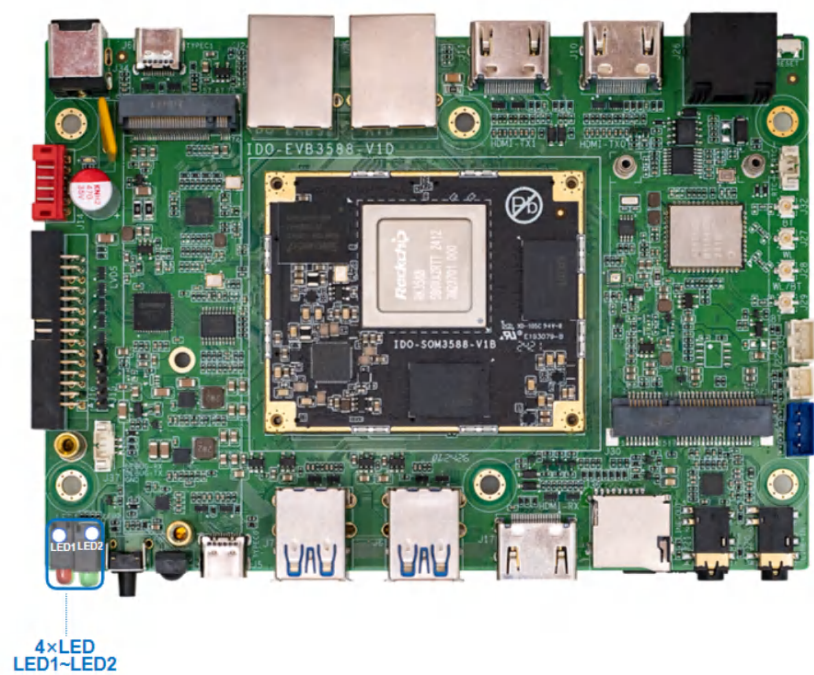
PH2 5V FAN风扇接口J35，如下图所示：



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

4.1.17. 用户LED

主板集成1个用户自定义控制LED2，如下图所示：



其中LED2的上层对应节点为 `/sys/class/leds/led2/brightness`

LED控制方法，如下表所示：

序号	操作	控制命令
LED2	打开	<code>echo 1 > /sys/class/leds/led2/brightness</code>
	关闭	<code>echo 0 > /sys/class/leds/led2/brightness</code>

4.1.18. IR

主板配置了一路红外接口，支持NEC编码遥控器，默认适配的遥控器型号为HTR-A07。



HTR-A07的键值表如下：

编号	按键	键值		编号	按键	键值
1	电源	KEY_POWER		21	1	KEY_1
2	TV	KEY_SCREEN		22	2	KEY_2
3	橙色	KEY_F1		23	3	KEY_3
4	绿色	KEY_F2		24	4	KEY_4
5	黄色	KEY_F3		25	5	KEY_5
6	紫色	KEY_F4		26	6	KEY_6
7	音量+	KEY_VOLUMEUP		27	7	KEY_7
8	音量-	KEY_VOLUMEDOWN		28	8	KEY_8
9	屏显	KEY_DISPLAY_OFF		29	9	KEY_9
10	静音	KEY_MUTE		30	TVNOW	KEY_DOT
11	上一节目	KEY_VIDEO_PREV		31	0	KEY_0
12	下一节目	KEY_VIDEO_NEXT		32	截屏	KEY_PRINT

13	上	KEY_UP		33		
14	左	KEY_LEFT		34		
15	下	KEY_DOWN		35		
16	右	KEY_RIGHT		36		
17	确认	KEY_ENTER		37		
18	返回	KEY_BACK		38		
19	主页	KEY_HOME		39		
20	菜单	KEY_MENU		40		

注：使用以下命令可以从调试串口打印按键的键值。

▼

Bash

```
1 echo 1 > /sys/module/rockchip_pwm_remotectl/parameters/code_print
```

使用evtest工具可以查看按键上报键值：

```
1 root@rk3588:/# evtest
2 No device specified, trying to scan all of /dev/input/event*
3 Available devices:
4 /dev/input/event0:      febd0030.pwm
5 /dev/input/event1:      rockchip-hdmi0 rockchip-hdmi0
6 /dev/input/event2:      rockchip-hdmi1 rockchip-hdmi1
7 /dev/input/event3:      rockchip,dp0 rockchip,dp0
8 /dev/input/event4:      rk805 pwrkey
9 /dev/input/event5:      adc-keys
10 /dev/input/event6:      headset-keys
11 /dev/input/event7:      rockchip-es8388 Headset
12 /dev/input/event8:      USB OPTICAL MOUSE
13 Select the device event number [0-8]: 0
14 Input driver version is 1.0.1
15 Input device ID: bus 0x19 vendor 0x524b product 0x6 version 0x100
16 Input device name: "febd0030.pwm"
17 Supported events:
18   Event type 0 (EV_SYN)
19   Event type 1 (EV_KEY)
20     Event code 2 (KEY_1)
21     Event code 3 (KEY_2)
22     Event code 4 (KEY_3)
23     Event code 5 (KEY_4)
24     Event code 6 (KEY_5)
25     Event code 7 (KEY_6)
26     Event code 8 (KEY_7)
27     Event code 9 (KEY_8)
28     Event code 10 (KEY_9)
29     Event code 11 (KEY_0)
30     Event code 14 (KEY_BACKSPACE)
31     Event code 28 (KEY_ENTER)
32     Event code 52 (KEY_DOT)
33     Event code 61 (KEY_F3)
34     Event code 102 (KEY_HOME)
35     Event code 103 (KEY_UP)
36     Event code 104 (KEY_PAGEUP)
37     Event code 105 (KEY_LEFT)
38     Event code 106 (KEY_RIGHT)
39     Event code 108 (KEY_DOWN)
40     Event code 109 (KEY_PAGEDOWN)
41     Event code 113 (KEY_MUTE)
42     Event code 114 (KEY_VOLUMEDOWN)
43     Event code 115 (KEY_VOLUMEUP)
44     Event code 116 (KEY_POWER)
45     Event code 139 (KEY_MENU)
```

```

46      Event code 143 (KEY_WAKEUP)
47      Event code 158 (KEY_BACK)
48      Event code 183 (KEY_F13)
49      Event code 184 (KEY_F14)
50      Event code 185 (KEY_F15)
51      Event code 186 (KEY_F16)
52      Event code 217 (KEY_SEARCH)
53      Event code 232 (KEY_REPLY)
54      Event code 241 (KEY_VIDEO_NEXT)
55      Event code 242 (KEY_VIDEO_PREV)
56      Event code 245 (KEY_DISPLAY_OFF)
57      Event code 248 (KEY_MICMUTE)
58      Event code 338 (?)
59      Event code 373 (KEY_MODE)
60      Event code 375 (KEY_SCREEN)
61      Event code 388 (KEY_TEXT)
62      Event code 400 (KEY_YELLOW)
63      Event code 401 (KEY_BLUE)
64      Event code 402 (KEY_CHANNELUP)
65
66  Properties:
67  Testing ... (interrupt to exit)
68  [ 4237.196132] USERCODE=0x1818
69  [ 4237.223070] RMC_GETDATA=ff
70  Event: time 1682532011.417156, type 1 (EV_KEY), code 2 (KEY_1), value 1
71  Event: time 1682532011.417156, ----- SYN_REPORT -----
72  Event: time 1682532011.475441, type 1 (EV_KEY), code 2 (KEY_1), value 0
73  Event: time 1682532011.475441, ----- SYN_REPORT -----
74  [ 4237.866797] USERCODE=0x1818
75  [ 4237.893797] RMC_GETDATA=fe
76  Event: time 1682532012.087855, type 1 (EV_KEY), code 3 (KEY_2), value 1
77  Event: time 1682532012.087855, ----- SYN_REPORT -----
78  Event: time 1682532012.178638, type 1 (EV_KEY), code 3 (KEY_2), value 0
79  Event: time 1682532012.178638, ----- SYN_REPORT -----
80  [ 4238.370395] USERCODE=0x1818
81  [ 4238.397454] RMC_GETDATA=fd
82  Event: time 1682532012.591469, type 1 (EV_KEY), code 4 (KEY_3), value 1
83  Event: time 1682532012.591469, ----- SYN_REPORT -----
84  Event: time 1682532012.682093, type 1 (EV_KEY), code 4 (KEY_3), value 0
85  Event: time 1682532012.682093, ----- SYN_REPORT -----
86  [ 4239.209857] USERCODE=0x1818
87  [ 4239.236861] RMC_GETDATA=fc
88  Event: time 1682532013.430863, type 1 (EV_KEY), code 5 (KEY_4), value 1
89  Event: time 1682532013.430863, ----- SYN_REPORT -----
90  Event: time 1682532013.521980, type 1 (EV_KEY), code 5 (KEY_4), value 0
91  Event: time 1682532013.521980, ----- SYN_REPORT -----

```

4.2. GPU

执行`glmark2-es2-wayland` 开启gpu测试窗口：

```

1 root@rk3588-buildroot:/# glmark2-es2-wayland
2 arm_release_ver: g24p0-00eac0, rk_so_ver: 8
3 =====
4     glmark2 2023.01
5     =====
6     OpenGL Information
7     GL_VENDOR:      ARM
8     GL_RENDERER:    Mali-G610
9     GL_VERSION:     OpenGL ES 3.2 v1.g24p0-00eac0.4388bae05e8d93b62450e275
10    21ab98fa
11    Surface Config: buf=32 r=8 g=8 b=8 a=8 depth=24 stencil=0 samples=0
12    Surface Size:   800x600 windowed
13    =====
14 [build] use-vbo=false:[10:13:41.753] seeing the first app
15 FPS: 3216 FrameTime: 0.311 ms
16 [build] use-vbo=true: FPS: 3256 FrameTime: 0.307 ms
17 [texture] texture-filter=nearest: FPS: 3808 FrameTime: 0.263 ms
18 [texture] texture-filter=linear: FPS: 3142 FrameTime: 0.318 ms
19 [texture] texture-filter=mipmap: FPS: 3990 FrameTime: 0.251 ms
20 [shading] shading=gouraud: FPS: 3889 FrameTime: 0.257 ms
21 [shading] shading=blinn-phong-inf: FPS: 3476 FrameTime: 0.288 ms
22 [shading] shading=phong: FPS: 3221 FrameTime: 0.311 ms
23 [shading] shading=cel: FPS: 3095 FrameTime: 0.323 ms
24 [bump] bump-render=high-poly: FPS: 2364 FrameTime: 0.423 ms
25 [bump] bump-render=normals: FPS: 3843 FrameTime: 0.260 ms
26 [bump] bump-render=height: FPS: 3979 FrameTime: 0.251 ms
27 [effect2d] kernel=0,1,0;1,-4,1;0,1,0;: FPS: 2928 FrameTime: 0.342 ms
28 [effect2d] kernel=1,1,1,1;1,1,1,1;1,1,1,1;: FPS: 2314 FrameTime: 0.4
32 ms
29 [pulsar] light=false:quads=5:texture=false: FPS: 3507 FrameTime: 0.285 ms
30 [desktop] blur-radius=5:effect=blur:passes=1:separable=true:windows=4: FPS
31 : 1190 FrameTime: 0.841 ms
32 [desktop] effect=shadow:windows=4: FPS: 2114 FrameTime: 0.473 ms
33 [buffer] columns=200:interleave=false:update-dispersion=0.9:update-fractio
34 n=0.5:update-method=map: FPS: 265 FrameTime: 3.782 ms
35 [buffer] columns=200:interleave=false:update-dispersion=0.9:update-fractio
36 n=0.5:update-method=subdata: FPS: 261 FrameTime: 3.833 ms
37 [buffer] columns=200:interleave=true:update-dispersion=0.9:update-fraction
38 =0.5:update-method=map: FPS: 401 FrameTime: 2.497 ms
39 [ideas] speed=duration: FPS: 1562 FrameTime: 0.640 ms
40 [jellyfish] <default>: FPS: 2834 FrameTime: 0.353 ms
41 [terrain] <default>: FPS: 358 FrameTime: 2.800 ms
42 [shadow] <default>: FPS: 3222 FrameTime: 0.310 ms
43 [refract] <default>:[ 439.446232] mali fb000000.gpu: GPU activity takes l
44 onger than time interval: 119999830 ns > 103349167 ns

```

```

39 FPS: 595 FrameTime: 1.682 ms
40 [conditionals] fragment-steps=0:vertex-steps=0: FPS: 3070 FrameTime: 0.326
    ms
41 [conditionals] fragment-steps=5:vertex-steps=0: FPS: 3068 FrameTime: 0.326
    ms
42 [conditionals] fragment-steps=0:vertex-steps=5: FPS: 3053 FrameTime: 0.328
    ms
43 [function] fragment-complexity=low:fragment-steps=5: FPS: 3447 FrameTime:
44 0.290 ms
    [function] fragment-complexity=medium:fragment-steps=5: FPS: 3694 FrameTim
45 e: 0.271 ms
    [loop] fragment-loop=false:fragment-steps=5:vertex-steps=5: FPS: 3075 Fram
46 eTime: 0.325 ms
    [loop] fragment-steps=5:fragment-uniform=false:vertex-steps=5: FPS: 3042 F
47 rameTime: 0.329 ms
    [loop] fragment-steps=5:fragment-uniform=true:vertex-steps=5: FPS: 3069 Fr
    ameTime: 0.326 ms
48
49 =====
                    glmark2 Score: 2676
50 =====

```

4.3. NPU

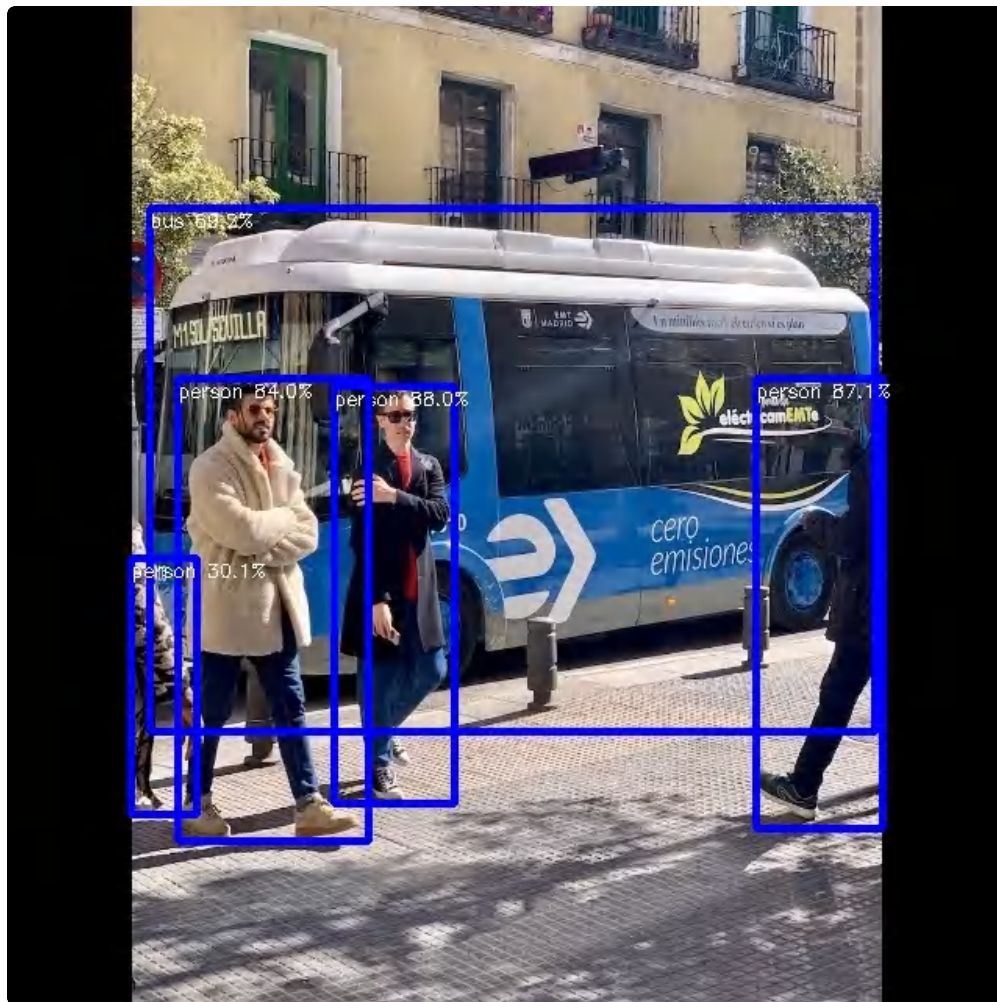
使用yolov5进行rknn模型转换测试，如下所示：

```

1 root@rk3588-buildroot:/# tar -xvf rknn_yolov5_demo_Linux.tar.gz
2
3 root@rk3588-buildroot:/# cd rknn_yolov5_demo_Linux
4
5 root@rk3588-buildroot:/rknn_yolov5_demo_Linux# chmod a+x rknn_yolov5_demo
6
7 root@rk3588-buildroot:/rknn_yolov5_demo_Linux# ./rknn_yolov5_demo ./model/
RK3588/yolov5s-640-640.rknn ./model/bus.jpg
8 post process config: box_conf_threshold = 0.25, nms_threshold = 0.45
9 Loading mode...
10 sdk version: 2.3.0 (c949ad889d@2024-11-07T11:35:33) driver version: 0.9.8
11 model input num: 1, output num: 3
12 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=INT
8, qnt_type=AFFINE, zp=-128, scale=0.003922
13 index=0, name=output, n_dims=4, dims=[1, 255, 80, 80], n_elems=1632000,
size=1632000, w_stride = 0, size_with_stride=1638400, fmt=NCHW, type=INT
8, qnt_type=AFFINE, zp=-128, scale=0.003860
14 index=1, name=283, 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_t
ype=AFFINE, zp=-128, scale=0.003922
15 index=2, name=285, 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_t
ype=AFFINE, zp=-128, scale=0.003915
16 model is NHWC input fmt
17 model input height=640, width=640, channel=3
18 Read ./model/bus.jpg ...
19 img width = 640, img height = 640
20 once run use 33.754000 ms
21 loadLabelName ./model/coco_80_labels_list.txt
22 person @ (209 244 286 506) 0.884139
23 person @ (478 238 559 526) 0.867678
24 person @ (110 238 230 534) 0.824685
25 bus @ (94 129 553 468) 0.705055
26 person @ (79 354 122 516) 0.339254
27 save detect result to ./out.jpg
28 loop count = 10 , average run 33.429000 ms

```

执行完后会得到转换后的模型图片，如下所示：



4.4. QT5

支持qt5.15，测试方法如下：

```
▼ Shell |
1 root@rk3588-buildroot:/# chmod a+x ./moveblocks
2 root@rk3588-buildroot:/# ./moveblocks
```

结果如下所示：

