

Purple Pi OH2 Linux使用手册



Purple Pi OH2 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

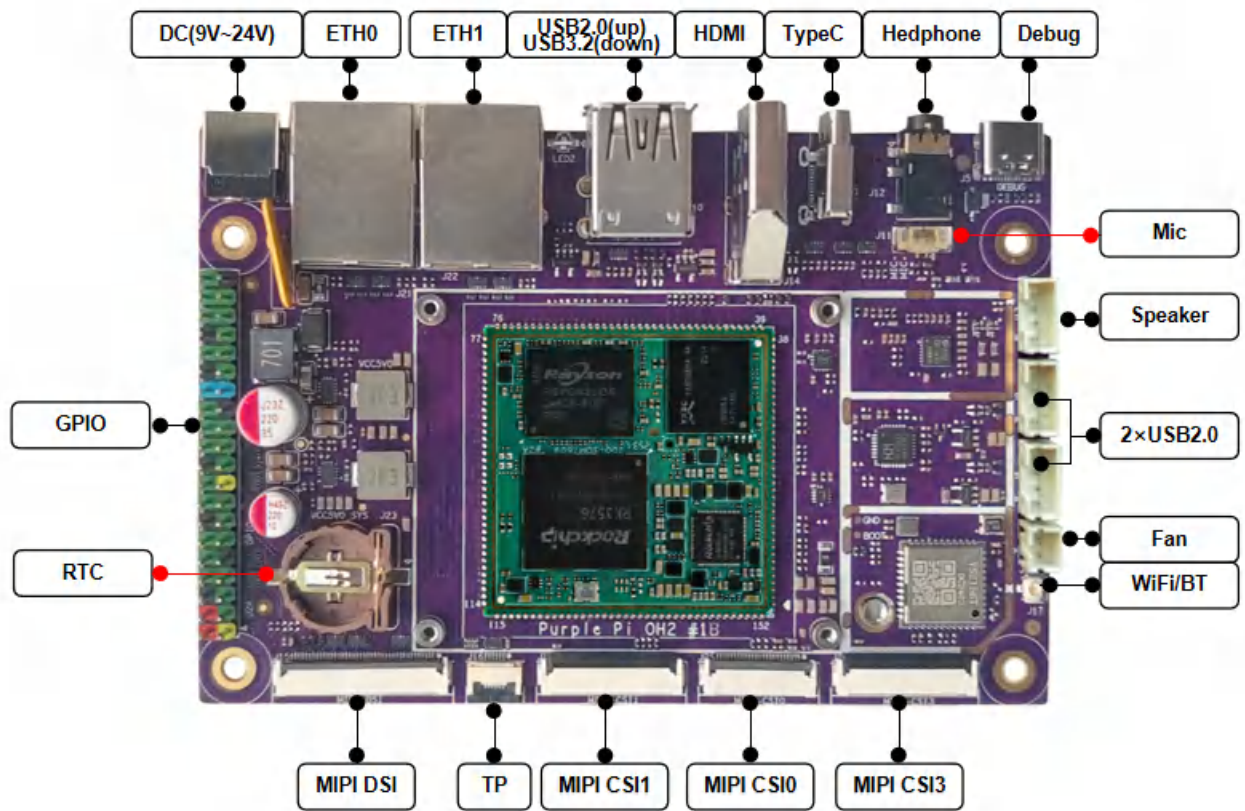
版本	PCBA版本	修订内容	修订	审核	日期
V1.0	V1A	创建文档	TWX		2025/07/18
V1.1	V1A	补充硬件接口标注图片	TWX		2025/07/21

V1.2	V1A	增加扩展接口描述	TWX		2025/07/22
V1.3	V1B	更新摄像头预览指令	TWX		2025/08/25
V1.4	V1B	更新Buildroot和Ubuntu系统	TWX		2025/08/28
V1.5	V1B	修正buildroot系统摄像头预览指令 修正摄像头视频播放方法	TWX		2025/09/05
V1.6	V1B	更新debian系统linaro用户密码	TWX		2025/09/08
V1.7	V1B	增加耳机录音和播放音频方法； 补充ubuntu系统播放音频到hdmi方法； 增加USB摄像头预览方法；	TWX		2025/09/10
V1.8	V1B	修正2路USB3.0为1路USB3.0+1路USB2.0； 增加ADC接口使用描述； 增加三轴传感器使用描述；	TWX		2025/09/16

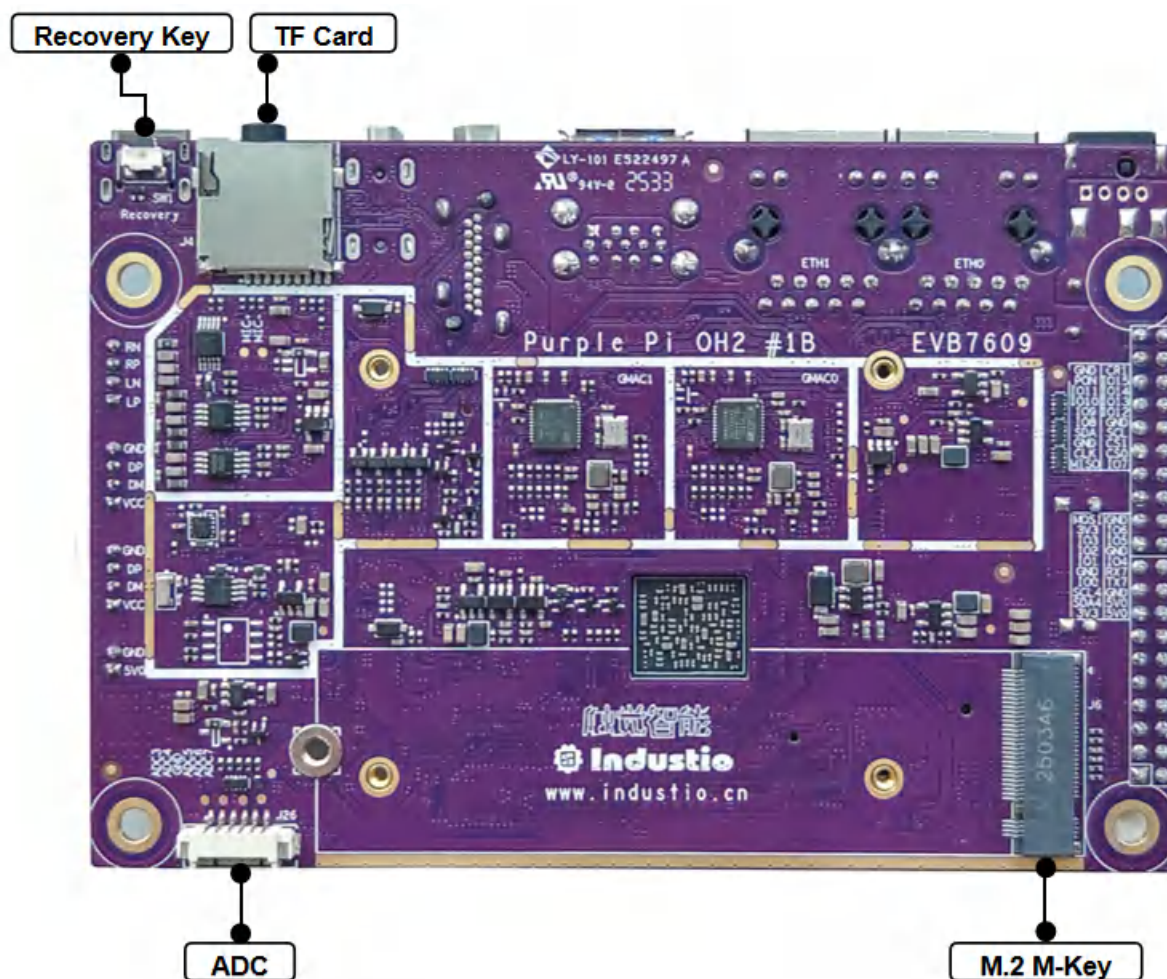
1. 硬件资源概况

1.1. 主板照片

Purple Pi OH2正面接口位号图，如下图所示：



Purple Pi OH2背面接口位号图，如下图所示：



1.2. 硬件资源

序号	名称	描述
1	内核版本	Linux 6.1.99
2	系统版本	Buildroot2024/Debian12/Ubuntu22.04
3	内存	LPDDR4 (4/8G)
4	存储	eMMC5.1 (32/64/128GB)
5	供电	DC接口12V@2A
6	显示	DP ×1 HDMI ×1 MIPI ×1

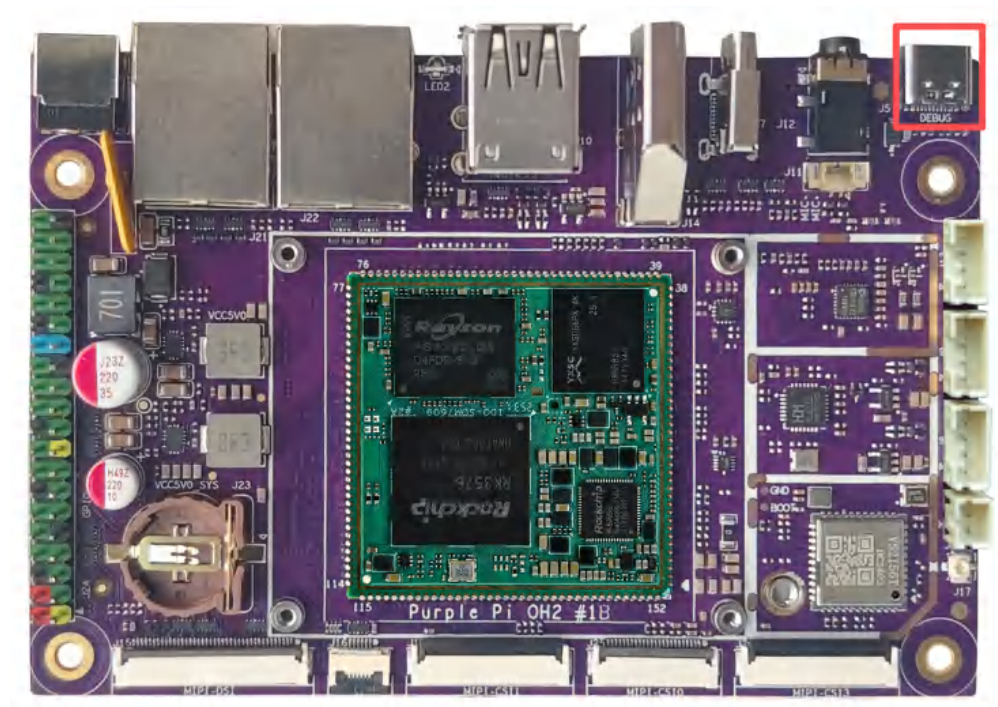
7	触摸	I2C-TP x1
8	USB OTG	USB OTG Type-C
9	USB HOST	USB3.0 HOST(Type-A) x2 USB2.0 HOST(PH2.0) x2
10	PCIe	PCIe2.1 NVME硬盘 x1
11	TF Card	SDIO3.0 TF Card x1
12	以太网	千兆以太网 x2
13	WIFI/BT	WiFi5
14	扬声器	PH2.0-4P(4ohm 3W)
15	耳机	CTIA标准四节耳机座
16	MIC	驻极体麦克风 x1 PDM 阵列麦克风x1
17	Camera	OV13850/IMX415 x 3
18	UART	TTL x3
19	DEBUG UART	TTL
20	CAN	CAN x2
21	RTC	HYM8563 x1
22	系统指示灯	系统指示灯 x1
23	按键	REC x1
24	Fan	5V PH2.02P x1
25	GPIO	GPIO预留 x8
26	ADC	ADC x5
27	SAI	SAI x 2
28	SPI	SPI x 1
29	I2C	I2C x 2

Purple Pi OH2 Debian系统

2. 功能及接口使用方法

2.1. DEBUG UART

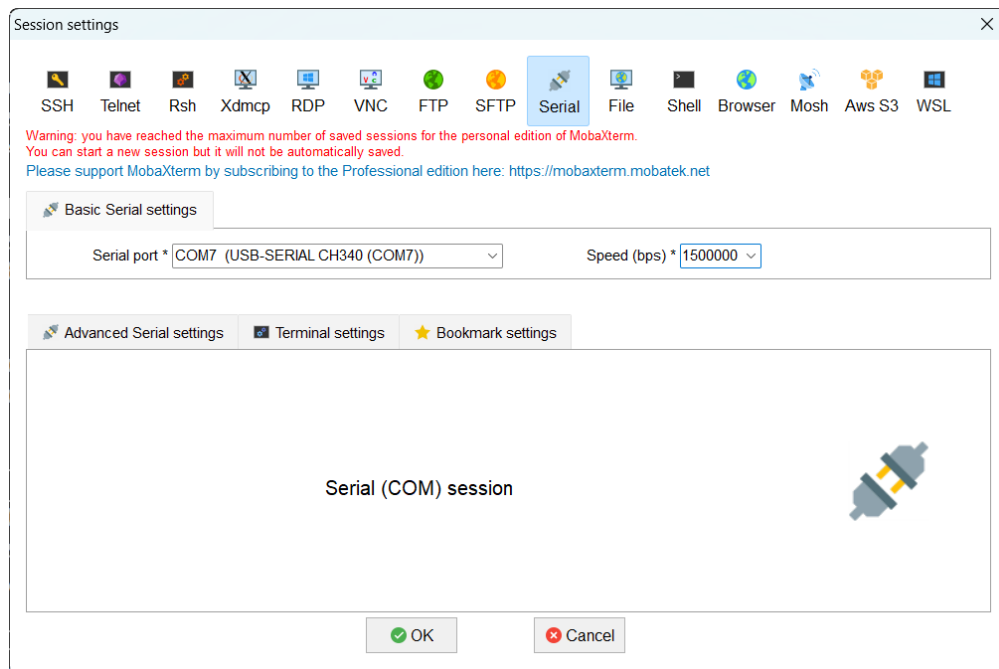
DEBUG UART位置J5，如下图所示：



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



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



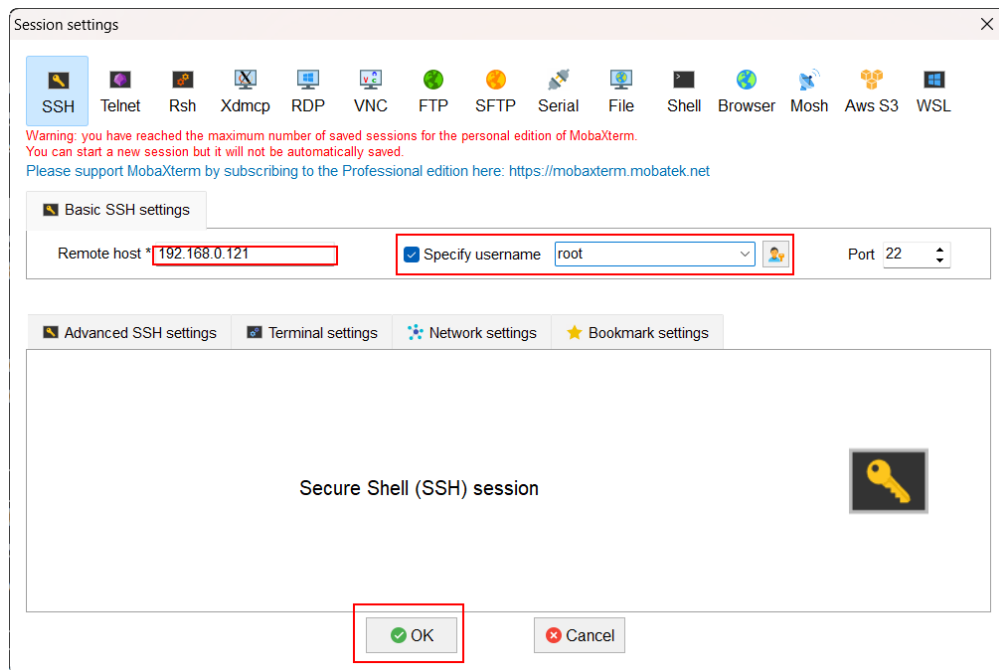
SSH登录

在知道IP的前提下，默认可以通过SSH登录进系统。

SSH登录账号密码：

root/123456

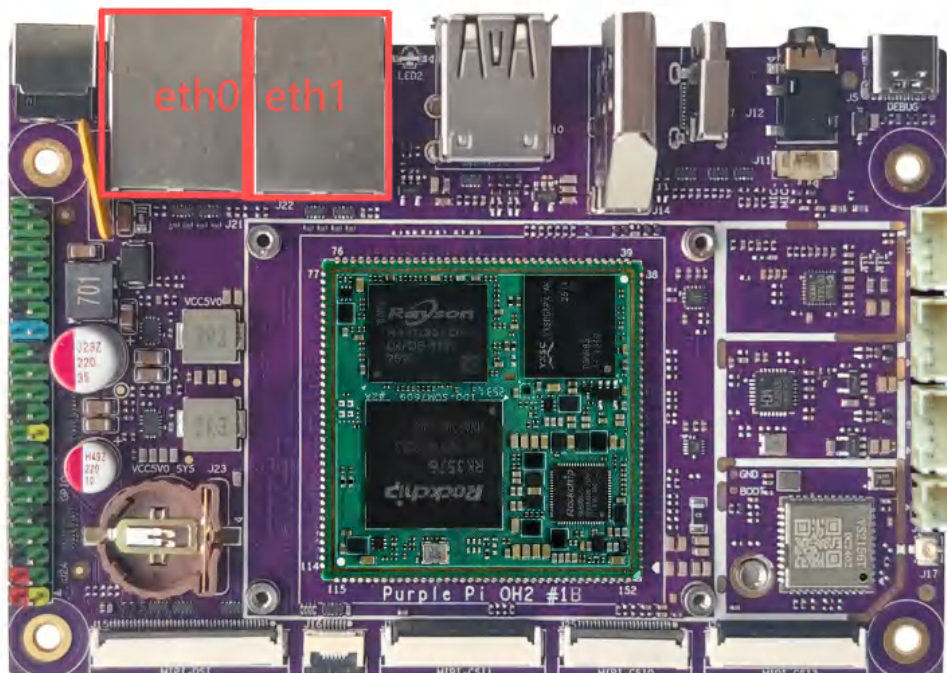
linaro/123456



2.2. 网络

2.2.1. 以太网

主板有两路千兆以太网接口位置J21和J22，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。

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
```

Ethernet静态IP设置:

```

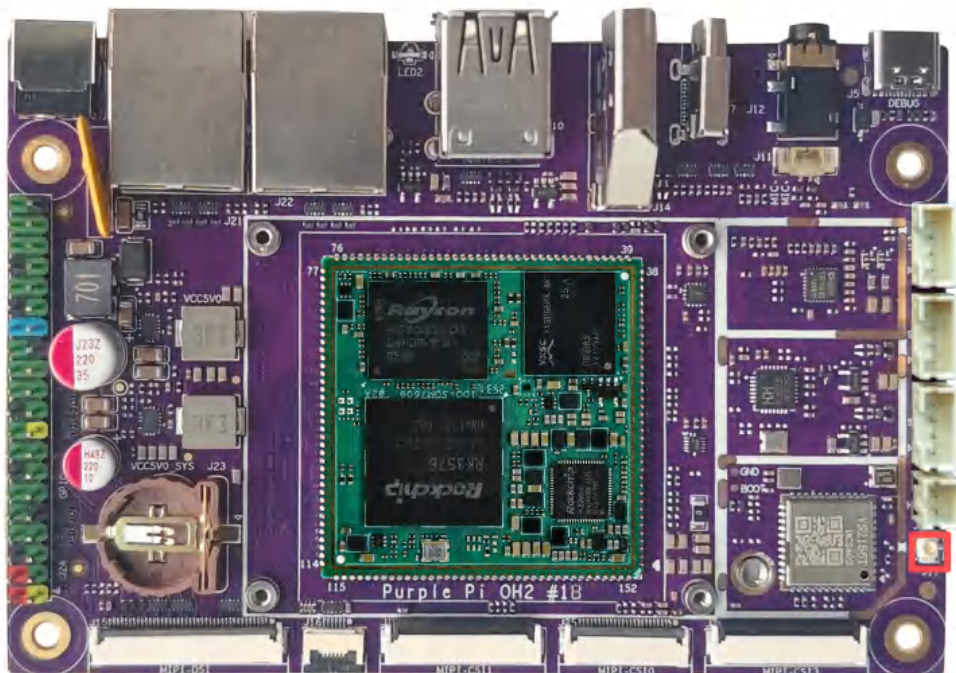
1 root@linaro-alip:/# touch /etc/netplan/01-netcfg.yaml
2 root@linaro-alip:/# vim /etc/netplan/01-netcfg.yaml
3 root@linaro-alip:/# cat /etc/netplan/01-netcfg.yaml
4 network:
5     version: 2
6     ethernet:
7         eth0:
8             dhcp4: no
9             addresses: [192.168.3.121/24]
10            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。

2.2.2. WiFi

使用WiFi/蓝牙时，需要连接天线以获得良好的信号，天线座子位于J17，如下图所示：



命令行可以使用nmcli工具连接wifi热点：

```
▼ Shell |
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：

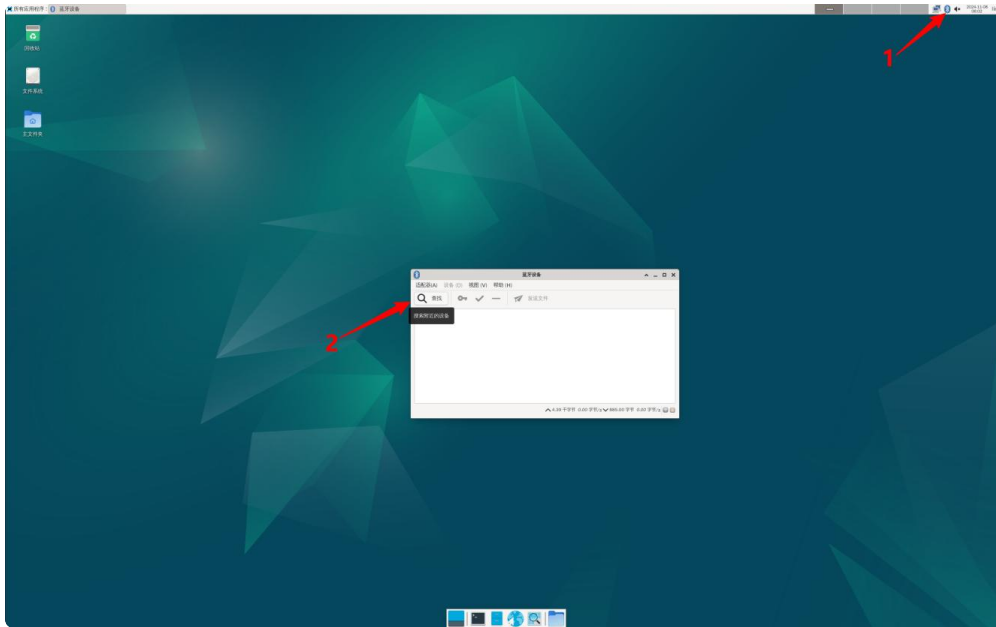
```
1 root@linaro-alip:/# ifconfig wlan0
2 wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
3     inet 192.168.0.118 netmask 255.255.255.0 broadcast 192.168.0.255
4     inet6 fe80::1036:80b4:5082:7440 prefixlen 64 scopeid 0x20<link>
5     ether c0:f5:35:12:ad:a2 txqueuelen 1000 (Ethernet)
6     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
```

2.2.3. Bluetooth

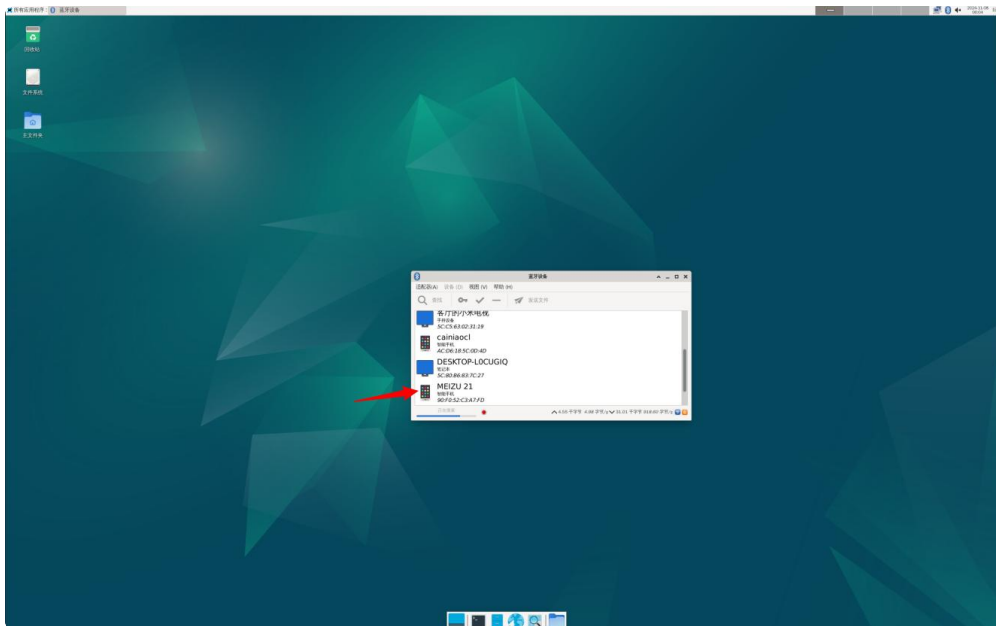
桌面连接：

点击【菜单】->【设置】->【已连接的设备】->【与新设备配对】

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



选择你要连接的蓝牙，点击右键连接即可：



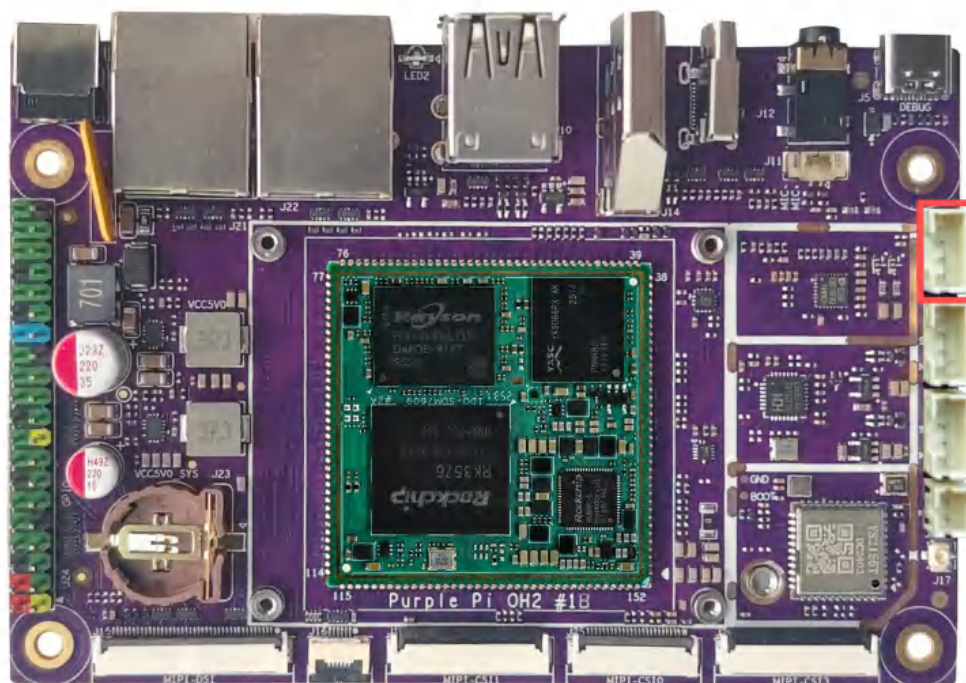
使用命令行操作

```
1 #打开蓝牙
2 root@linaro-alip:/# bluetoothctl power on
3 #扫描蓝牙
4 root@linaro-alip:/# bluetoothctl scan on
5 #查看蓝牙设备
6 root@linaro-alip:/# bluetoothctl devices
7 #信任蓝牙设备
8 root@linaro-alip:/# bluetoothctl trust 7C:C1:80:09:DD:6C
9 #蓝牙配对
10 root@linaro-alip:/# bluetoothctl pair 7C:C1:80:09:DD:6C
11 #连接蓝牙
12 root@linaro-alip:/# bluetoothctl connect 7C:C1:80:09:DD:6C
```

2.3. 声音

2.3.1. 喇叭

喇叭接口为PH2.0-4P连接器J13，如下图所示：

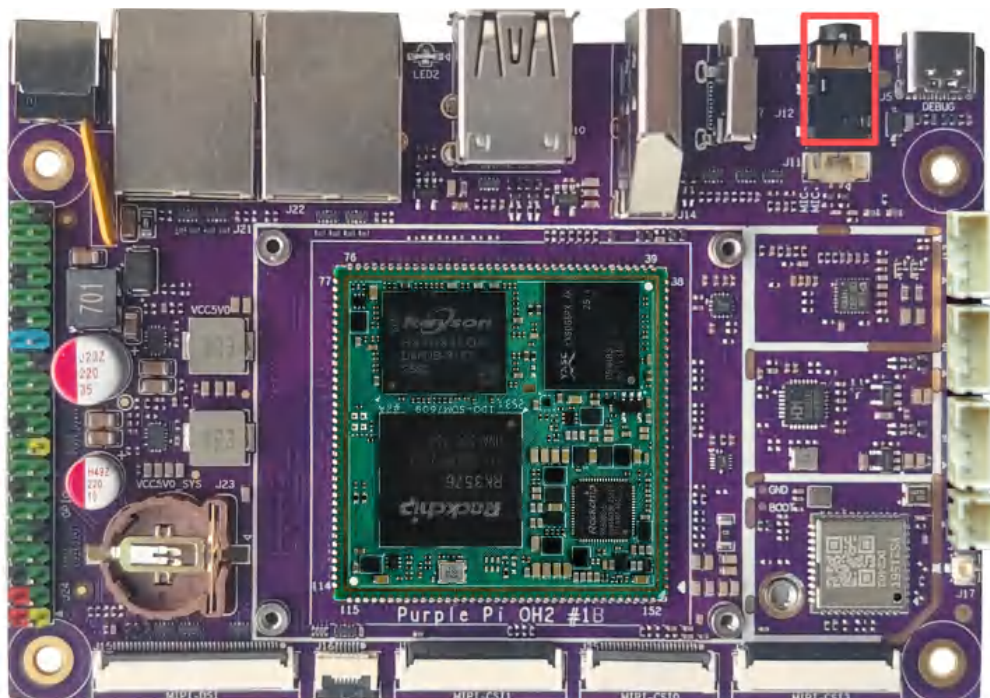


- 支持双声道，每个声道支持4Ω 3W输出

```
1  #查看声卡
2  root@linaro-alip:/# aplay -l
3  **** List of PLAYBACK Hardware Devices ****
4  card 0: rockchipes8388 [rockchip-es8388], device 0: dailink-multicodecs ES
    8323 HiFi-0 [dailink-multicodecs ES8323 HiFi-0]
5      Subdevices: 1/1
6      Subdevice #0: subdevice #0
7  card 1: rockchiphdmi [rockchip-hdmi], device 0: rockchip-hdmi i2s-hifi-0
    [rockchip-hdmi i2s-hifi-0]
8      Subdevices: 1/1
9      Subdevice #0: subdevice #0
10
11
12  #音频播放到声卡
13  root@linaro-alip:/# aplay -D hw:0,0 xihuangni.wav
14  Playing WAVE 'xihuangni.wav' : Signed 16 bit Little Endian, Rate 44100 H
    z, Stereo
15
16  #音频播放到HDMI
17  root@linaro-alip:/# aplay -D hw:1,0 xihuangni.wav
18  Playing WAVE 'xihuangni.wav' : Signed 16 bit Little Endian, Rate 44100 H
    z, Stereo
```

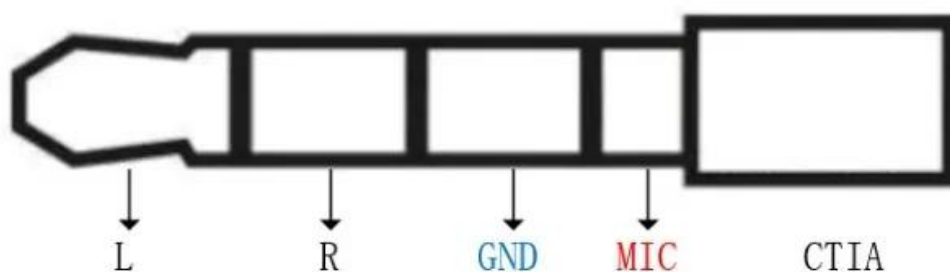
2.3.2. 耳机

主板支持1路3.5mm四节耳机座（CTIA）J12，和1路MX1.25T-5P耳机座子并用，用户可根据需求选用其中1路耳机座子使用，如下图所示：



- 支持耳机检测
- 支持耳机录音

CTIA标准四段式耳机，定义如下：



耳机录音：

```

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

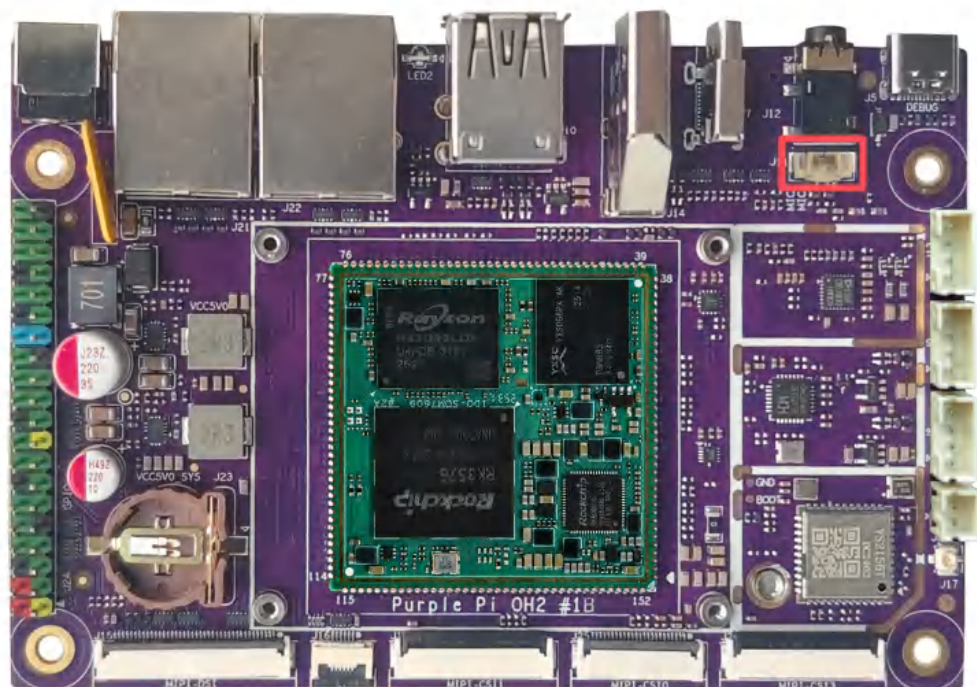
播放音频到耳机：

```

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

2.3.3. Mic

MX1.25T-2P麦克风接口J11，如下图所示：



```
1  #查看录音设备
2  root@linaro-alip:/# arecord -l
3  **** List of CAPTURE Hardware Devices ****
4  card 0: rockchip-es8388 [rockchip-es8388], device 0: dailink-multicodecs ES
   8323 HiFi-0 [dailink-multicodecs ES8323 HiFi-0]
5      Subdevices: 1/1
6      Subdevice #0: subdevice #0
7  card 1: rockchip-hdmi [rockchip-hdmi], device 0: rockchip-hdmi i2s-hifi-0
   [rockchip-hdmi i2s-hifi-0]
8      Subdevices: 1/1
9      Subdevice #0: subdevice #0
10
11 #捕获音量设置(不要设置最大值):
12 root@linaro-alip:~# amixer -c 0 cset numid=50 100
13 numid=50,iface=MIXER,name='Capture Digital Volume'
14     ; type=INTEGER,access=rw---R--,values=2,min=0,max=192,step=0
15     : values=100,100
16     | dBscale-min=-96.00dB,step=0.50dB,mute=1
17
18 root@linaro-alip:~# amixer -c 0 cget numid=52
19 numid=52,iface=MIXER,name='Left Channel Capture Volume'
20     ; type=INTEGER,access=rw---R--,values=1,min=0,max=8,step=0
21     : values=0
22     | dBscale-min=0.00dB,step=3.00dB,mute=0
23
24 root@linaro-alip:~# amixer -c 0 cget numid=53
25 numid=53,iface=MIXER,name='Right Channel Capture Volume'
26     ; type=INTEGER,access=rw---R--,values=1,min=0,max=8,step=0
27     : values=0
28     | dBscale-min=0.00dB,step=3.00dB,mute=0
29
30 自动增益控制(打开):
31 root@linaro-alip:~# amixer -c 0 cset numid=41 3
32 numid=41,iface=MIXER,name='ALC Capture Function'
33     ; type=ENUMERATED,access=rw-----,values=1,items=4
34     ; Item #0 'Off'
35     ; Item #1 'Right'
36     ; Item #2 'Left'
37     ; Item #3 'Stereo'
38     : values=3
39
40 调整目标音量:
41 root@linaro-alip:~# amixer -c 0 cset numid=38 8
42 numid=38,iface=MIXER,name='ALC Capture Target Volume'
43     ; type=INTEGER,access=rw-----,values=1,min=0,max=15,step=0
```

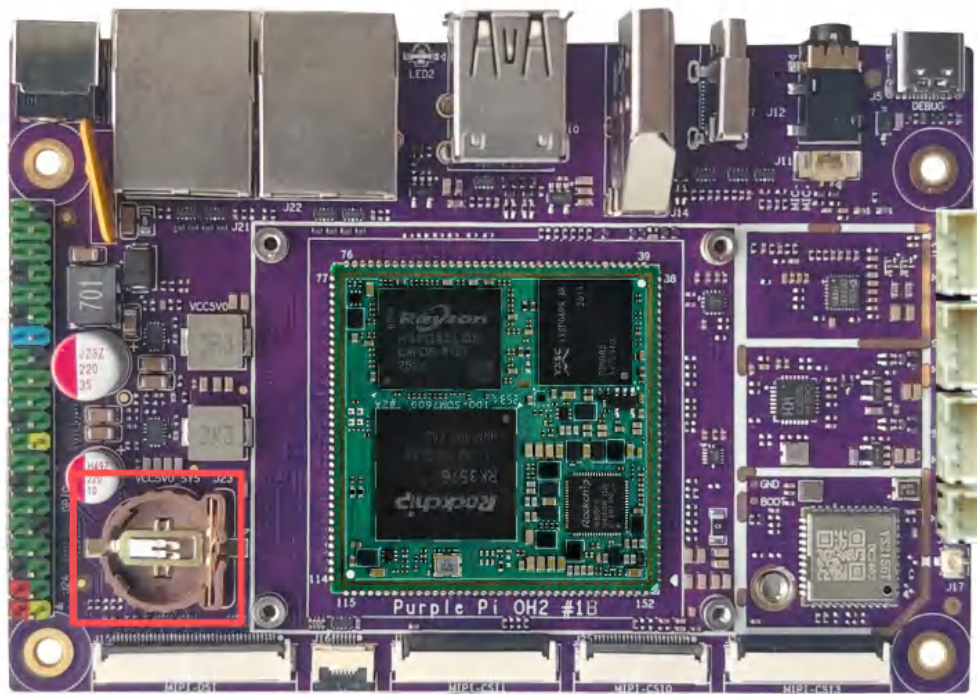
```

44      : values=8
45      设置噪声门限值:
46      root@linaro-alip:~# amixer -c 0 cset numid=46 15
47      numid=46,iface=MIXER,name='ALC Capture NG Threshold'
48      ; type=INTEGER,access=rw-----,values=1,min=0,max=31,step=0
49      : values=15
50      确保捕获静音未开启:
51      root@linaro-alip:~# amixer -c 0 cset numid=51 off
52      numid=51,iface=MIXER,name='Capture Mute'
53      ; type=BOOLEAN,access=rw-----,values=1
54      : values=off
55
56      #主mic开关 (mic录音需要打开, 耳机录音需要关闭) :
57      root@linaro-alip:~# amixer -c 0 cset numid=67 on
58      numid=67,iface=MIXER,name='Main Mic Switch'
59      ; type=BOOLEAN,access=rw-----,values=1
60      : values=on
61      root@linaro-alip:/# echo -n "mic2" > /sys/class/es8388_mic/mic_channel
62      root@linaro-alip:/# arecord -D hw:0,0 -f S16_LE -r 16000 -c 2 test_mic2.wa
63      v
64
65      #耳机录音:
66      root@linaro-alip:/# amixer -c 0 cset numid=67 off
67      root@linaro-alip:/# echo -n "mic1" > /sys/class/es8388_mic/mic_channel
        root@linaro-alip:/# arecord -D hw:0,0 -f S16_LE -r 16000 -c 2 test_mic1.wa
        v

```

2.4. RTC

MX1.25T-2P RTC电池座J23, 如下图所示:



使用3V 纽扣电池。

rtc时间设置方法：

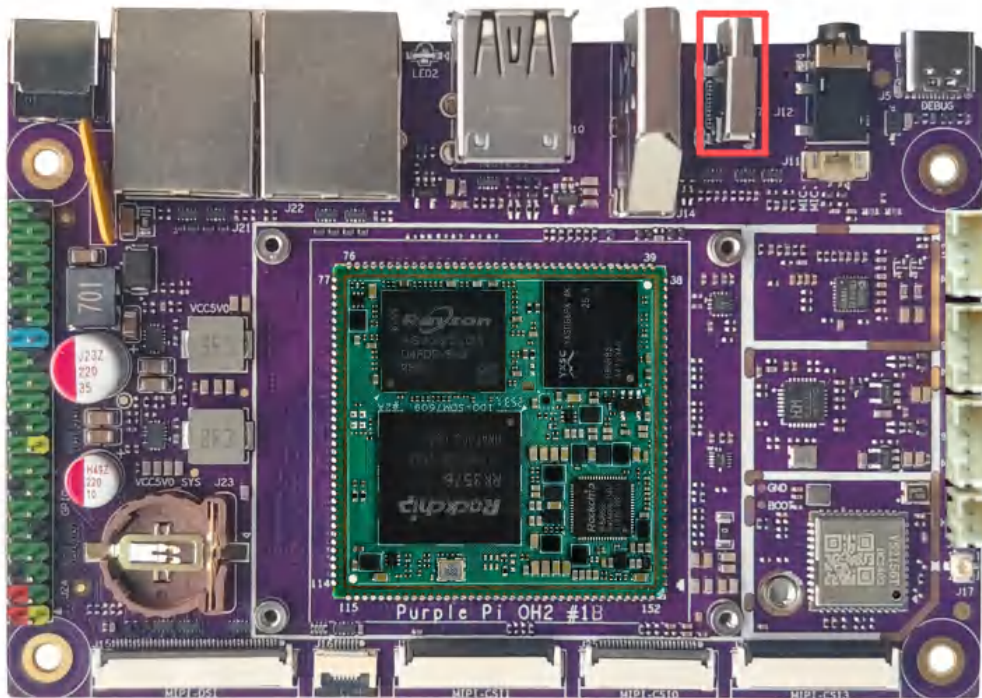
```
1  #设置时间
2  root@linaro-alip:/# date -s "2024-10-09 14:02:30"
3
4  #将rtc时钟调整为与目前的系统时钟一致
5  root@linaro-alip:/# hwclock -w
6
7  #获取硬件rtc当前时间（断电重启读取时间没有太大偏差）
8  root@linaro-alip:/# hwclock -r
9  2024-10-09 14:02:35.945604+00:00
```

2.5. USB接口

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

2.5.1. TypeC接口

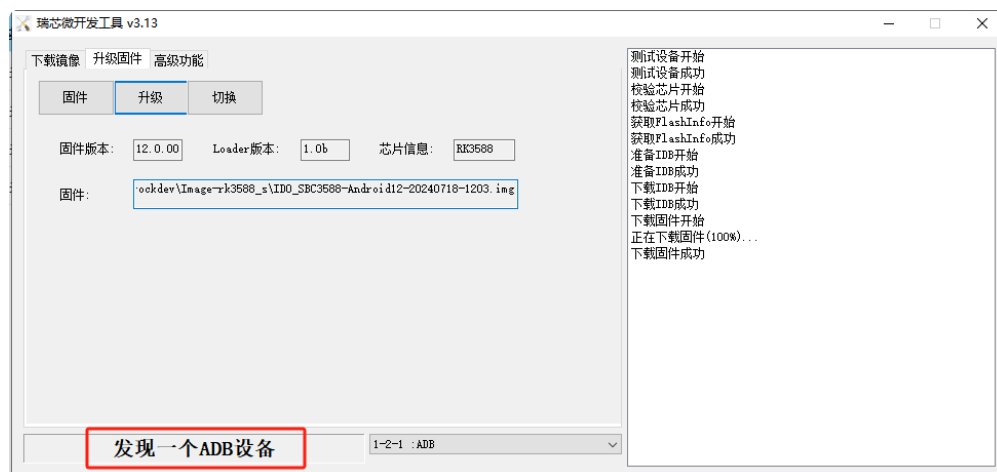
TypeC接口（USB3.2 Gen1 OTG+DP1.4输出）J7，如下图所示：



- 支持Host、Device模式自动切换
- 支持DP显示输出

Device从机模式

使用TypeC数据线连接电脑，烧录工具能发现一个ADB设备，如下图所示：



- 可使用ADB相关开发工具对盒子进行功能调试

Host主机模式

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



- 可识别U盘、键盘、鼠标并正常使用

DP模式

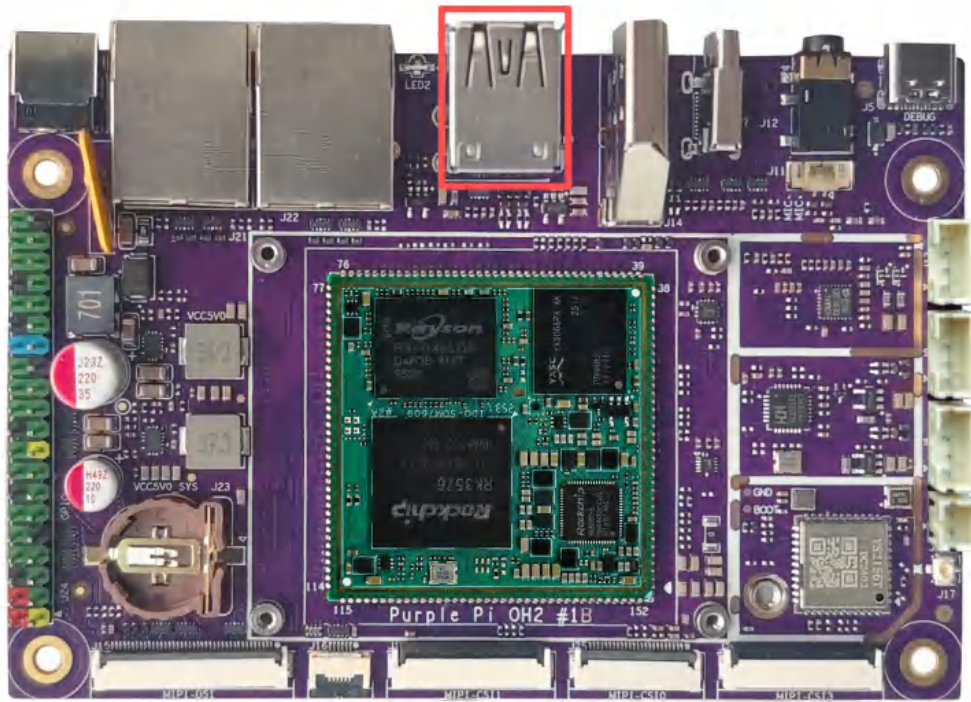
通过TypeC全功能数据线接入DP显示器，或通过TYPE-C to HDMI数据线连接HDMI显示器



- 主板画面通过TYPE-C DP输出正常，DP声音输出正常

2.5.2. USB3.0接口

主板支持1个USB3.0 TYPE-A 接口，位于J10（下），如下图所示：



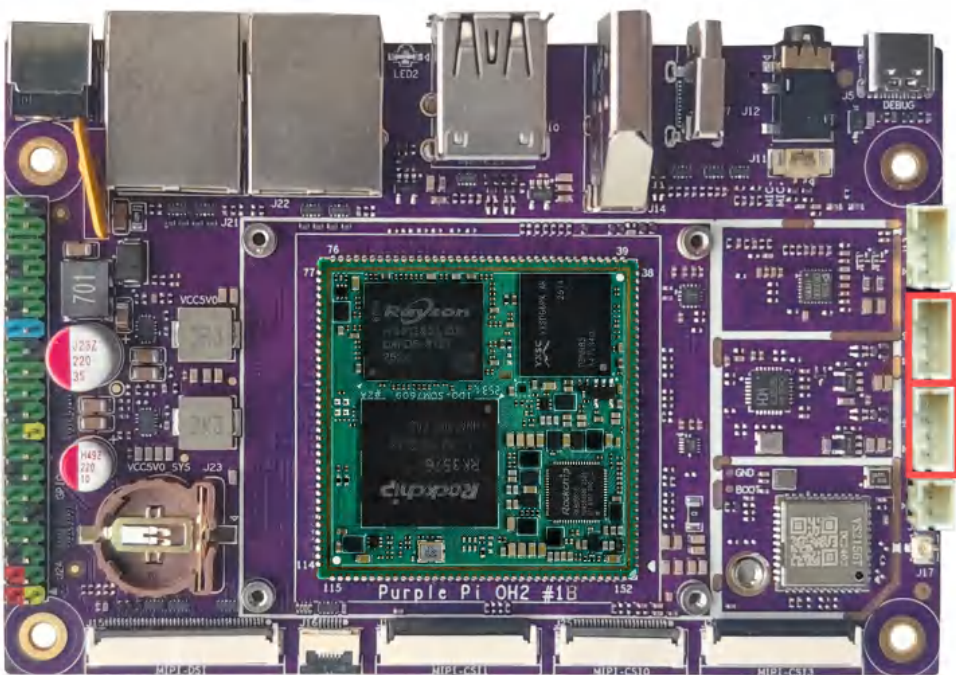
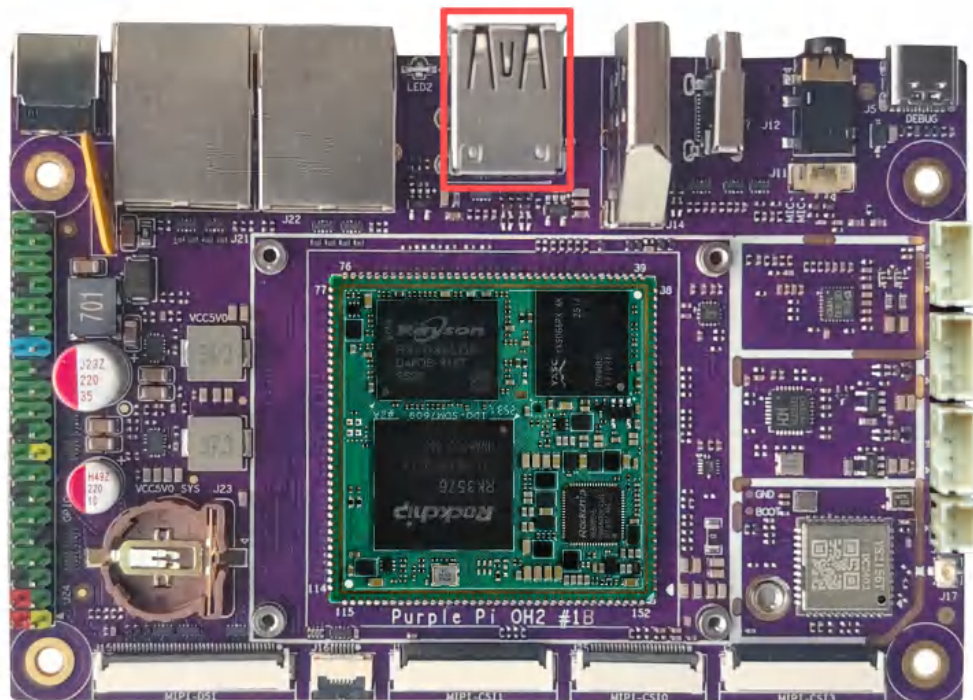
USB母座提供5V@1A供电能力，2个USB端口供电控制是共用的，USB电源控制，如下表所示：

USB端口	默认状态	动作	命令
USB1/2	开启	关闭电源	echo 0 > /sys/class/leds/usb_pwr/brightness
		开启电源	echo 1 > /sys/class/leds/usb_pwr/brightness

供电控制说明：设备节点写"0"关闭电源，写"1"开启电源

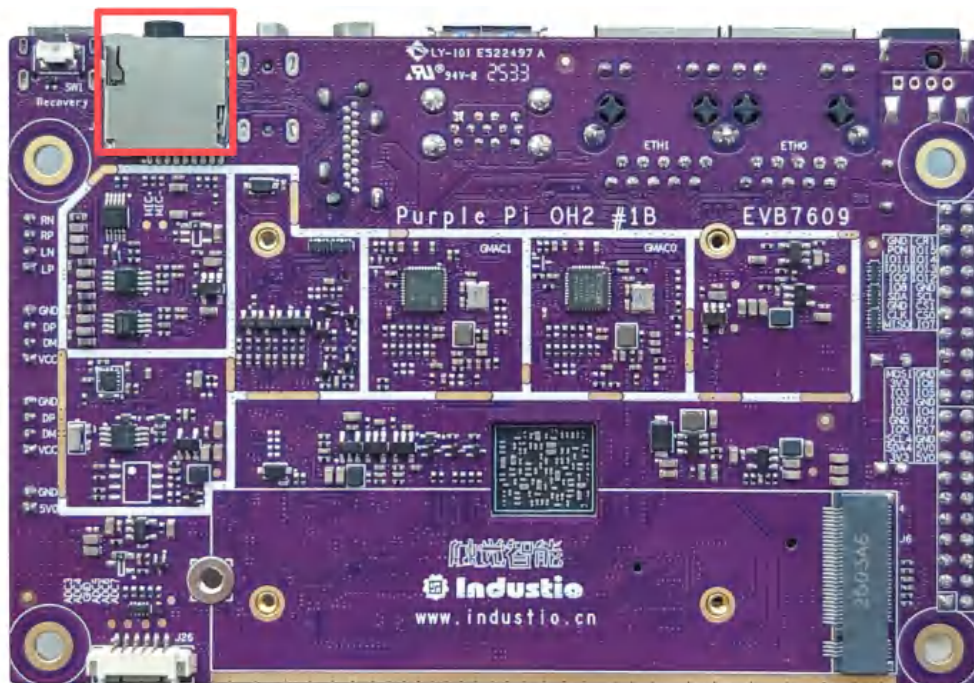
2.5.3. USB2.0接口

主板配置了1路USB2.0 TYPE-A接口，位于J10（上），及2路PH2.0-4P接口，位于J8和J9，USB接口均提供5V@1A的驱动能力，如下图所示：



2.6. TF Card

TF卡座支持SDIO3.0，支持高速SD卡，接口位于主板背面J4，如下图所示：



TF卡安装方向，主板正面朝上时TF卡触点朝上。

Shell

```

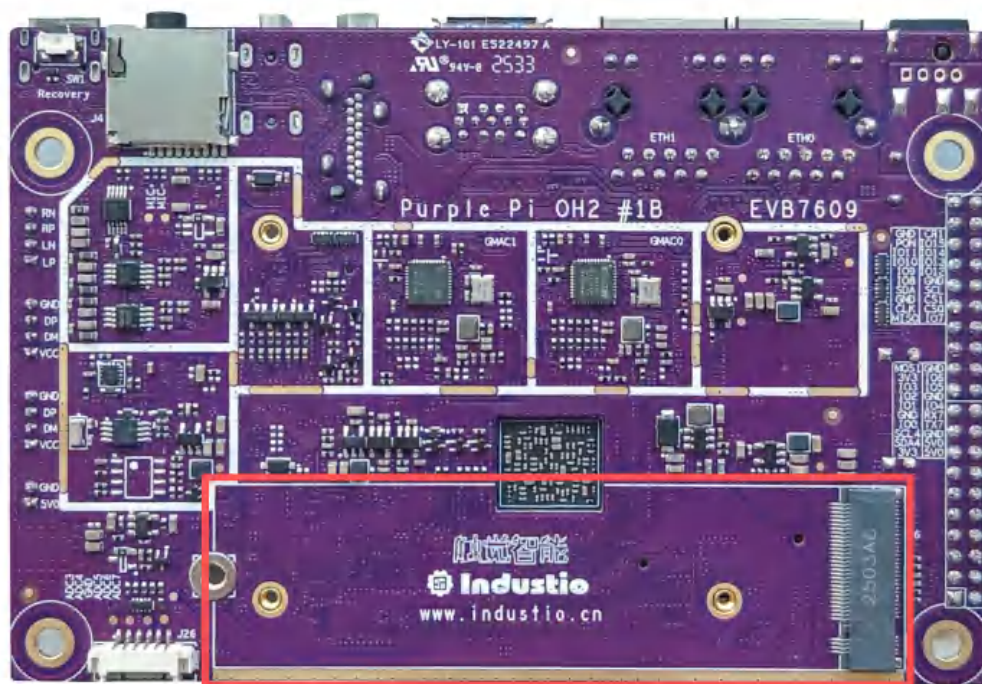
1  root@linaro-alip:/# fdisk -l
2  Disk /dev/ram0: 8 MiB, 8388608 bytes, 16384 sectors
3  Units: sectors of 1 * 512 = 512 bytes
4  Sector size (logical/physical): 512 bytes / 4096 bytes
5  I/O size (minimum/optimal): 4096 bytes / 4096 bytes
6  .....
7
8  Device          Boot Start      End  Sectors  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 tmpfs          2.9G     0   2.9G    0% /dev/shm
16 tmpfs          1.2G   3.6M   1.2G    1% /run
17 tmpfs          5.0M   12K   5.0M    1% /run/lock
18 tmpfs          2.9G   16K   2.9G    1% /tmp
19 /dev/mmcblk2p7 124M   12M  109M   10% /oem
20 /dev/mmcblk2p6 4.0G   292K   4.0G    1% /userdata
21 /dev/mmcblk1p1 1.9G   768K   1.9G    1% /media/linaro/sdcard

```

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

2.7. M.2接口

主板上使用标准M.2-M-key M.2接口连接座，如下图所示：



支持PCIe2.1通信，适用2280尺寸NVME固态硬盘。

```

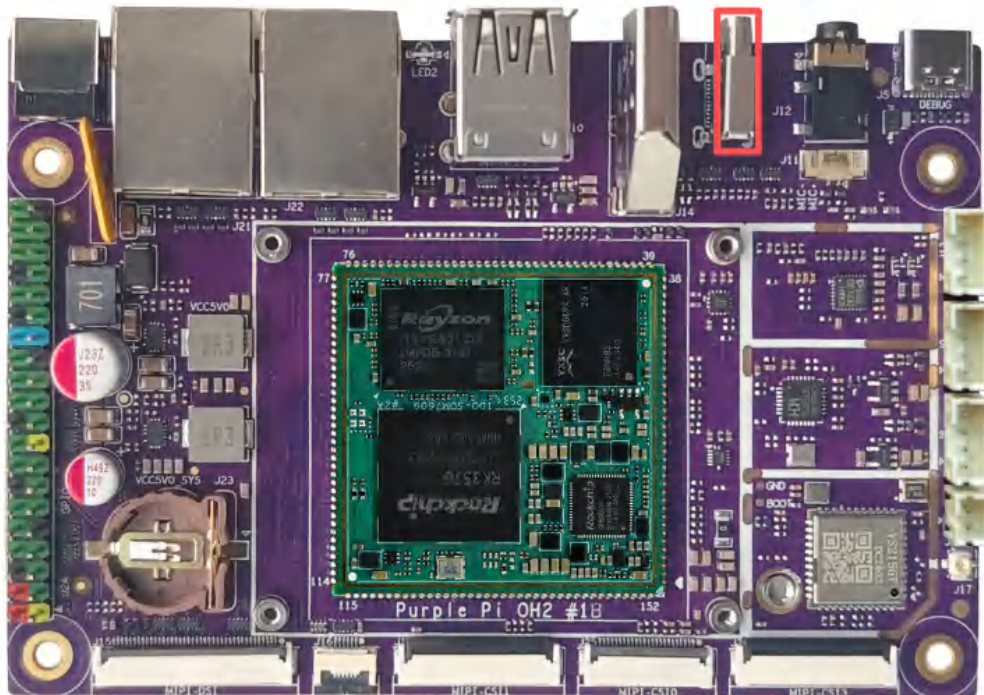
1  root@linaro-alip:/# fdisk -l
2  Disk /dev/ram0: 8 MiB, 8388608 bytes, 16384 sectors
3  Units: sectors of 1 * 512 = 512 bytes
4  Sector size (logical/physical): 512 bytes / 4096 bytes
5  I/O size (minimum/optimal): 4096 bytes / 4096 bytes
6
7
8  Disk /dev/ram1: 8 MiB, 8388608 bytes, 16384 sectors
9  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 tmpfs          2.9G     0   2.9G    0% /dev/shm
29 tmpfs          1.2G   3.6M   1.2G    1% /run
30 tmpfs          5.0M   12K   5.0M    1% /run/lock
31 tmpfs          2.9G   24K   2.9G    1% /tmp
32 /dev/mmcblk2p7 124M   12M  109M   10% /oem
33 /dev/mmcblk2p6 4.0G  292K   4.0G    1% /userdata
34 /dev/nvme0n1p1 477G   2.1G  475G    1% /media/linaro/nvme

```

2.8. LCD显示

2.8.1. DP

DP接口 (USB-C) J7, 如下图所示:



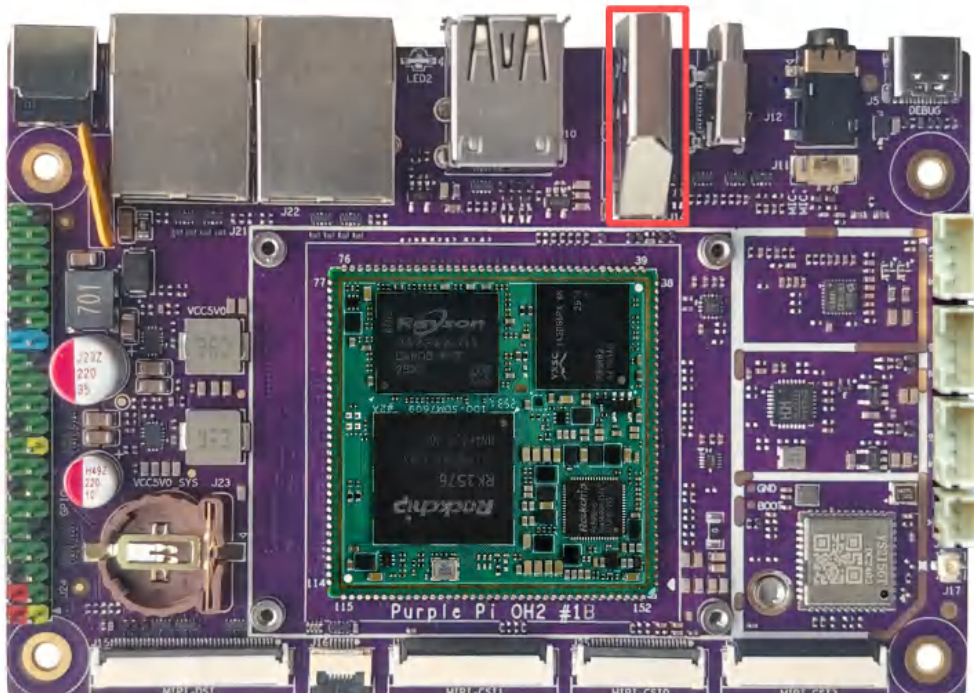
可以使用 USB-C 转 HDMI 高清线连接 HDMI 显示器输出画面，如下图所示：



- 最高支持4K@30fps输出

2.8.2. HDMI-TX

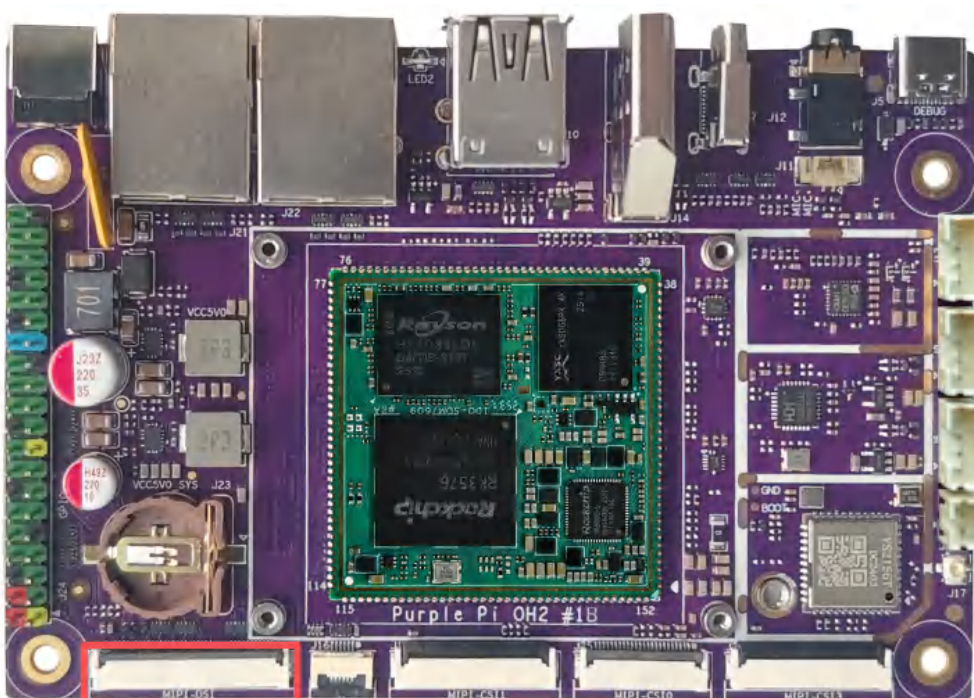
标准HDMI-A接口J14，如下图所示：



- 最高支持 HDMI2.1 8K@60fps 输出

2.8.3. MIPI

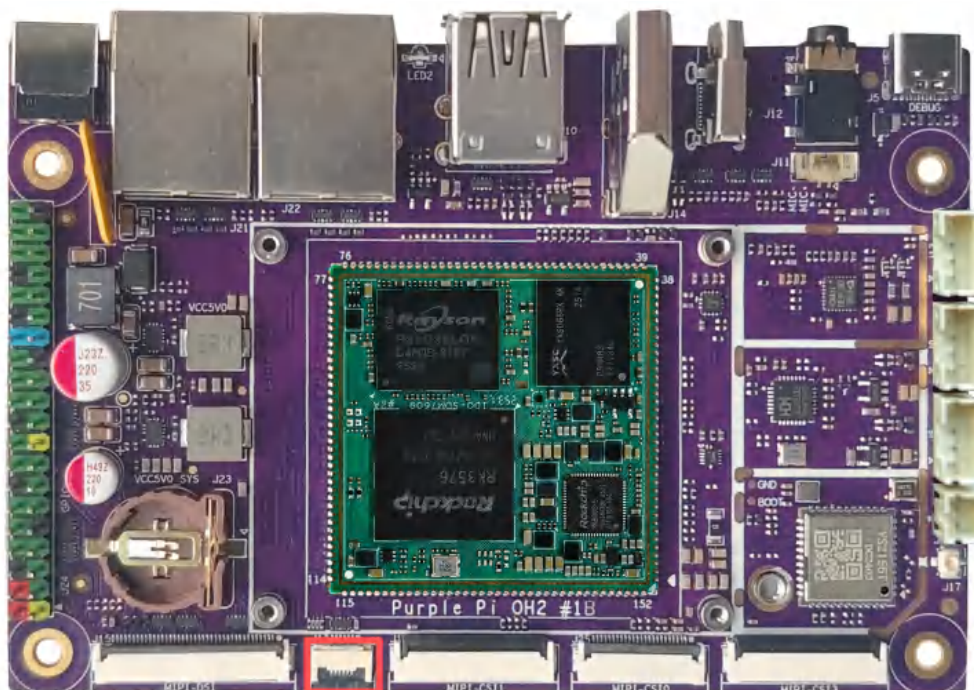
J15 MIPI接口为40Pin FPC 0.5mm 上接，如下图所示：



- MIPI供电为3.3V

2.9. TP接口

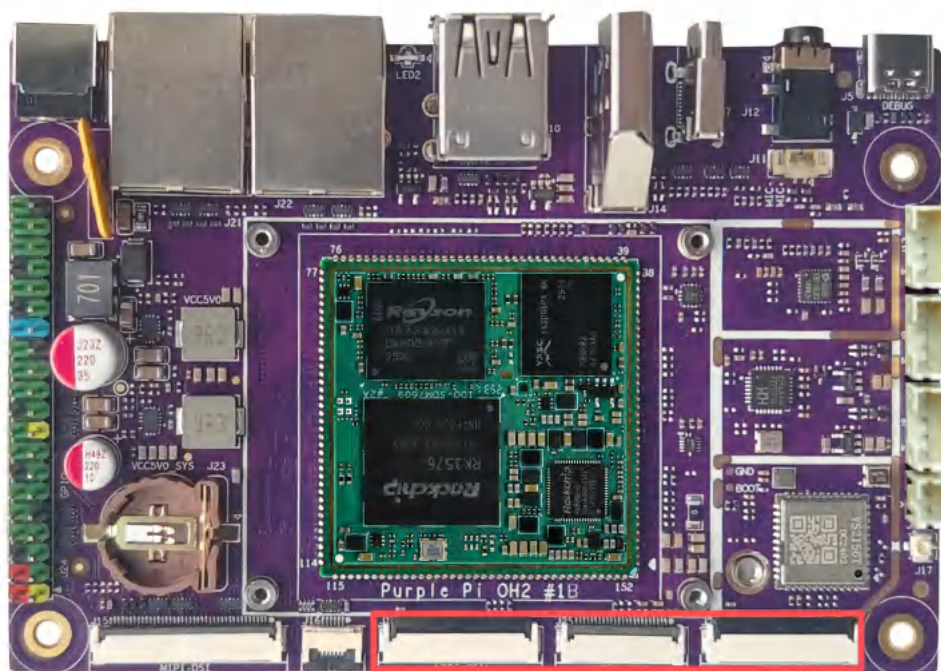
TP接口为(J16) 6Pin FPC 0.5mm座子，如下图所示：



- 触摸 TP 接口接线方法为下接

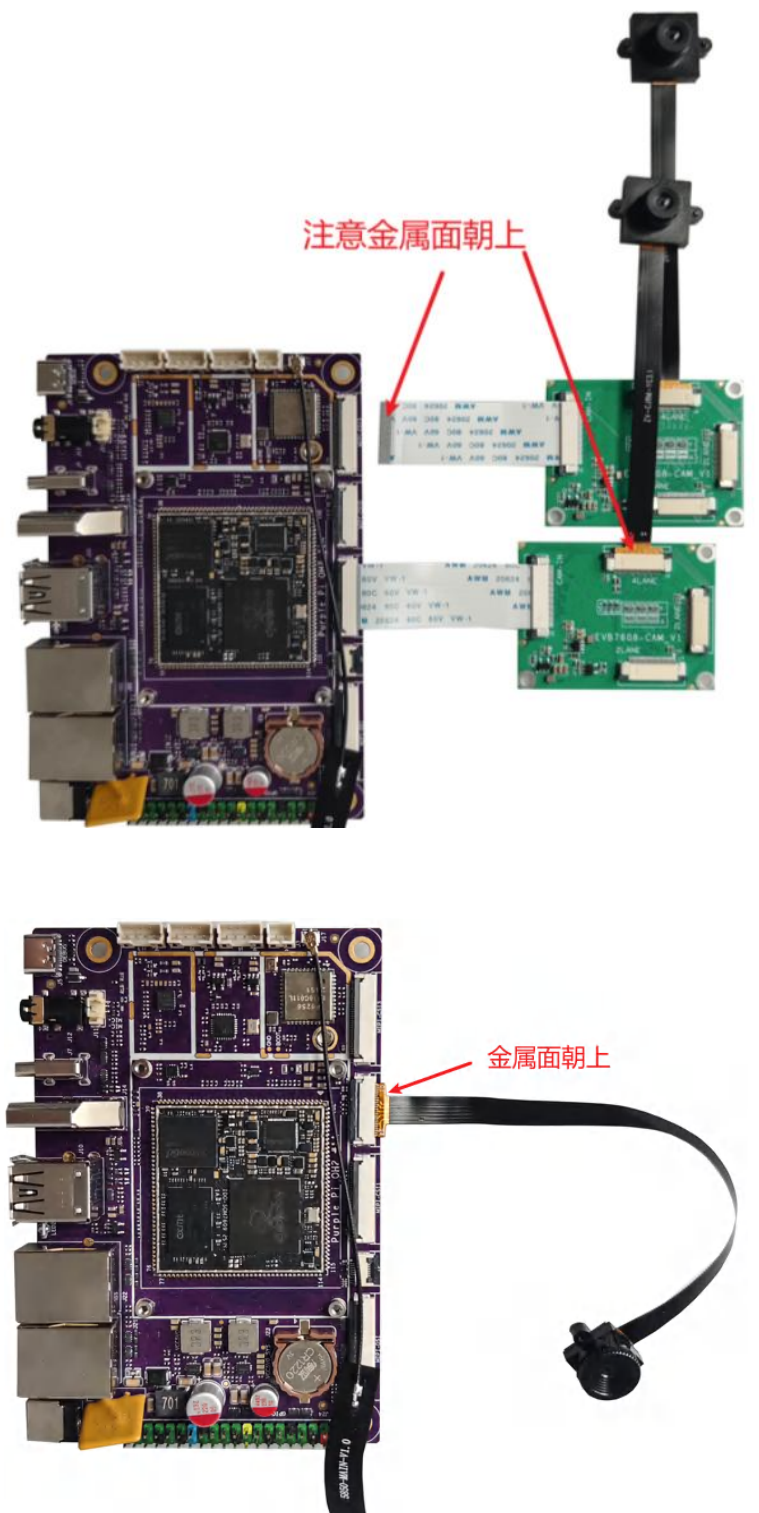
2.10. MIPI Camera

主板共有3路摄像头接口，分别是J19、J20和J25，如下图所示：



J19和J20使用FPC05-PUW-32P抽拉式上接座子，J25使用FPC05-PUW-24P抽拉式上接座子。其中J19和J25默认适配 IMX415摄像头，J20默认适配OV13855摄像头。

接法如下图所示：



2.10.1. 预览命令：

预览J19摄像头：

```

1 root@linaro-alip:/# export DISPLAY=:0
2 root@linaro-alip:/# gst-launch-1.0 v4l2src device=/dev/video42 ! video/x-raw, format=NV12, width=1920, height=1080, framerate=30/1 ! videoconvert ! autovideosink sync=false

```

预览J20摄像头（当前只支持OV13855，画面偏绿色）：

```

1 root@linaro-alip:/# export DISPLAY=:0
2 root@linaro-alip:/# gst-launch-1.0 v4l2src device=/dev/video51 ! video/x-raw, format=NV12, width=1920, height=1080, framerate=30/1 ! videoconvert ! autovideosink sync=false

```

预览J25摄像头：

```

1 root@linaro-alip:/# export DISPLAY=:0
2 root@linaro-alip:/# gst-launch-1.0 v4l2src device=/dev/video33 ! video/x-raw, format=NV12, width=1920, height=1080, framerate=30/1 ! videoconvert ! autovideosink sync=false

```

2.10.2. 抓图：

抓取J19摄像头图片：

```

1 root@linaro-alip:/# v4l2-ctl --verbose -d /dev/video42 --set-fmt-video=width=1920,height=1080,pixelformat='NV12' --stream-mmap=4 --stream-skip=3 --stream-to=./test-1920x1080.yuv --stream-count=1 --stream-poll

```

抓取J20摄像头图片：

```

1 root@linaro-alip:/# v4l2-ctl --verbose -d /dev/video51 --set-fmt-video=width=1920,height=1080,pixelformat='NV12' --stream-mmap=4 --stream-skip=3 --stream-to=./test-1920x1080.yuv --stream-count=1 --stream-poll

```

抓取J25摄像头图片：

```
1 root@linaro-alip:/# v4l2-ctl --verbose -d /dev/video33 --set-fmt-video=width=1920,height=1080,pixelformat='NV12' --stream-mmap=4 --stream-skip=3 --stream-to=./test-1920x1080.yuv --stream-count=1 --stream-poll
```

查看照片:

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

2.10.3. 抓视频:

抓取J19摄像头视频:

```
1 root@linaro-alip:/# v4l2-ctl --stream-mmap=4 -d /dev/video42 --set-fmt-video=width=1920,height=1080,pixelformat='NV12' --stream-to=./output.yuv
```

抓取J20摄像头视频:

```
1 root@linaro-alip:/# v4l2-ctl --stream-mmap=4 -d /dev/video51 --set-fmt-video=width=1920,height=1080,pixelformat='NV12' --stream-to=./output.yuv
```

抓取J25摄像头视频:

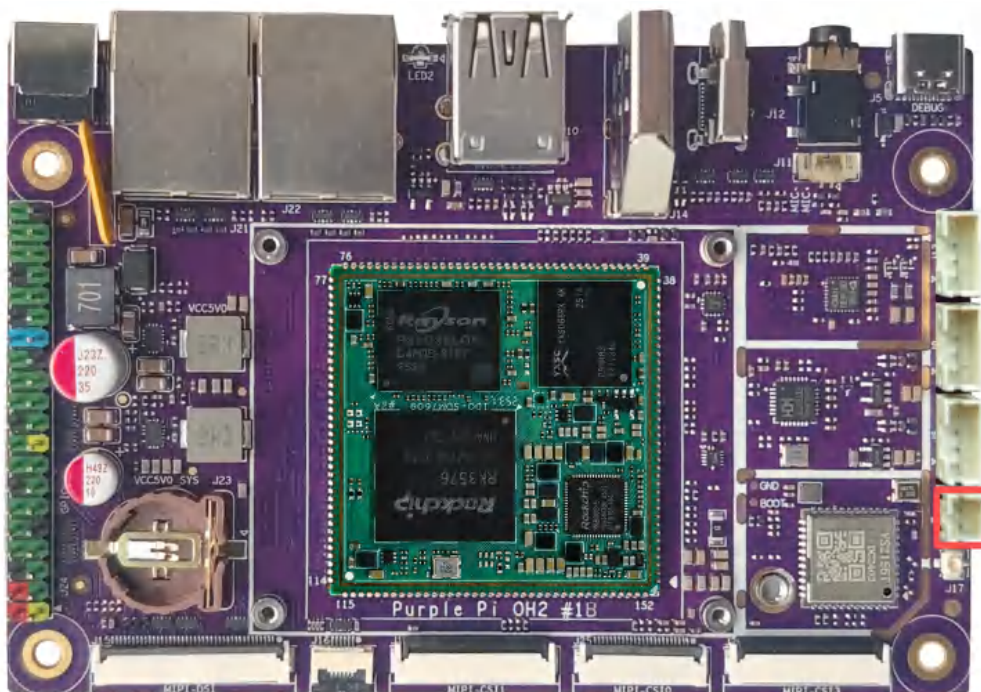
```
1 root@linaro-alip:/# v4l2-ctl --stream-mmap=4 -d /dev/video33 --set-fmt-video=width=1920,height=1080,pixelformat='NV12' --stream-to=./output.yuv
```

播放视频:

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

2.11. Fan风扇

PH2 5V Fan风扇接口J3，如下图所示：



2.12. 三轴传感器

主板配置了1路三轴传感器，传感器型号是SC7A20HTR，在系统中对应的设备节点为/sys/bus/iio/devices/iio:device0。

2.12.1. 设置采样率

默认采样率为1Hz，最高支持400Hz，采样率设置方法如下：

```
1 echo 400 > /sys/bus/iio/devices/iio:device0/sampling_frequency
```

2.12.2. 获取三轴传感器数据

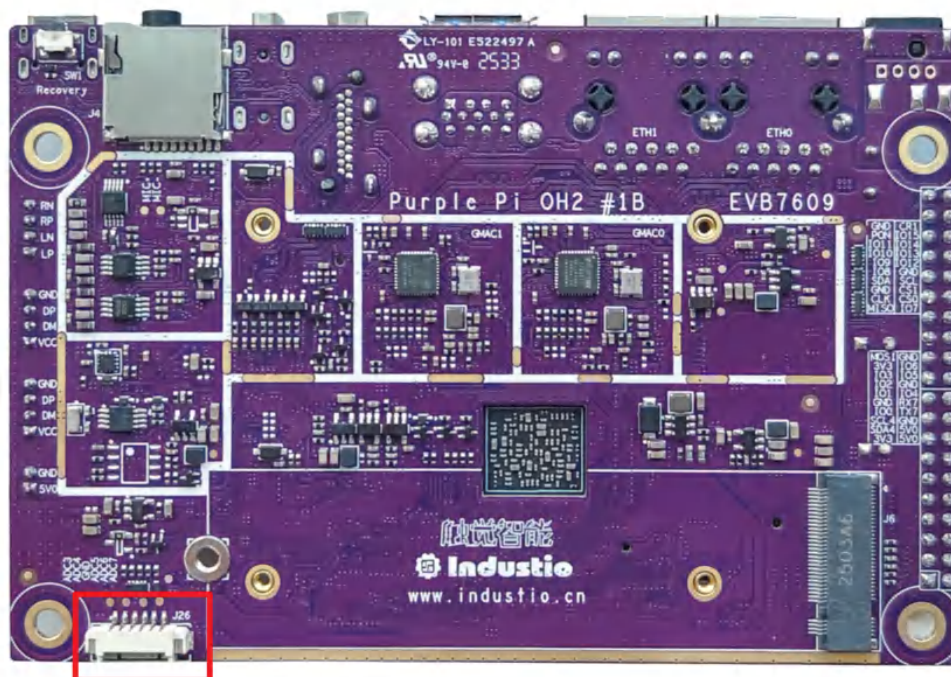
```

1 # 获取X轴加速度
2 cat /sys/bus/iio/devices/iio:device0/in_accel_x_raw
3
4 # 获取Y轴加速度
5 cat /sys/bus/iio/devices/iio:device0/in_accel_y_raw
6
7 # 获取Z轴加速度
8 cat /sys/bus/iio/devices/iio:device0/in_accel_z_raw

```

2.13. ADC

主板ADC接口位于背面J26，支持5路ADC输入，精度是12位，最高输入电压1.8V。



5路ADC对应的设备节点如下表：

pin	名称	设备节点
1	ADC3	/sys/bus/iio/devices/iio:device1/in_voltage3_ra
2	ADC4	/sys/bus/iio/devices/iio:device1/in_voltage4_ra
3	GND	—
4	ADC5	/sys/bus/iio/devices/iio:device1/in_voltage5_ra

5	ADC6	/sys/bus/iio/devices/iio:device1/in_voltage6_ra
6	ADC7	/sys/bus/iio/devices/iio:device1/in_voltage7_ra

2.13.1. 获取输入ADC raw值

通过读取设备节点直接获取ADC raw值，如获取ADC3的raw值：

▼

Shell

```

1 root@linaro-alip:~# cat /sys/bus/iio/devices/iio:device1/in_voltage3_raw
2 3770

```

2.13.2. ADC输入电压计算公式

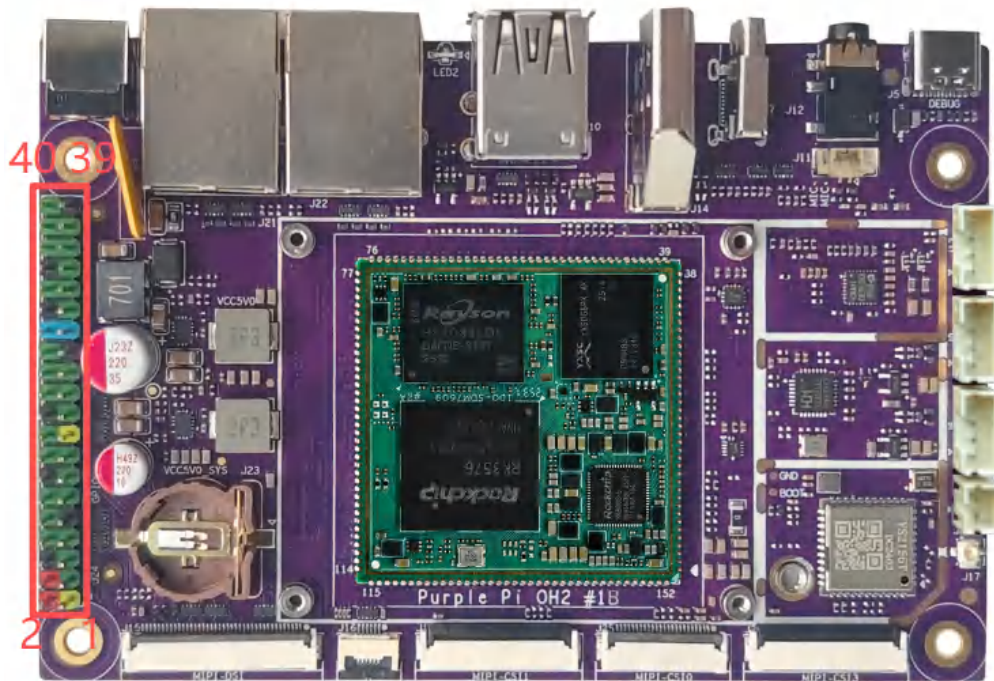
$$V_{in} = (raw/1024) \times 1.8v$$

其中 V_{in} 表示ADC输入电压，raw表示ADC raw值。比如ADC3的raw值为258，则ADC3输入电压为：

$$V_{in} = (3770/1024) \times 1.8v = 1.657v$$

2.14. 扩展接口

主板预留了40pin双排针扩展接口，包括1路UART，1路SPI，2路I2C和16个GPIO。40pin双排针引脚定义如下：



pin	功能	pin	功能
1	3.3V	2	5V
3	I2C4_SDA	4	5V
5	I2C4_SCL	6	GND
7	GPIO76	8	UART7_TX
9	GND	10	UART7_RX
11	GPIO75	12	GPIO77
13	GPIO72	14	GND
15	GPIO73	16	GPIO74
17	3.3V	18	GPIO136
19	SPI0_MOSI	20	GND
21	SPI0_MISO	22	GPIO137
23	SPI0_SCLK	24	SPI0_CS0
25	GND	26	SPI0_CS1
27	I2C7_SDA	28	I2C7_SCL

29	GPIO56	30	GND
31	GPIO57	32	GPIO132
33	GPIO58	34	GND
35	GPIO59	36	GPIO134
37	POWER_KEY	38	GPIO98
39	GND	40	GPIO99

2.14.1. GPIO

主板扩展接口预留了16个GPIO，各个GPIO引脚定义描述，如下图所示：

序号	pin	GPIO 标号	GPIO NUM	LIBGPIO节点
1	7	GPIO2_B4	76	gpiochip2 12
2	11	GPIO2_B3	75	gpiochip2 11
3	12	GPIO2_B5	77	gpiochip2 13
4	13	GPIO2_B0	72	gpiochip2 8
5	15	GPIO2_B1	73	gpiochip2 9
6	16	GPIO2_B2	74	gpiochip2 10
7	18	GPIO4_B0	136	gpiochip4 8
8	22	GPIO4_B1	137	gpiochip4 9
9	29	GPIO1_D0	56	gpiochip1 24
10	31	GPIO1_D1	57	gpiochip1 25
11	32	GPIO4_A4	132	gpiochip4 4
12	33	GPIO1_D2	58	gpiochip1 26
13	35	GPIO1_D3	59	gpiochip1 27
14	36	GPIO4_A6	134	gpiochip4 6
15	38	GPIO3_A2	98	gpiochip3 2

16	40	GPIO3_A3	99	gpiochip3 3
----	----	----------	----	-------------

GPIO控制，根据上表【LIBGPIO节点】进行命令操作

```
libgpiod C |
1 #安装命令
2 apt install -y libgpiod-dev gpiod
3
4 # 设置 GPIO2_B4 输出高电平状态
5 gpioset gpiochip2 12=1
6
7 # 设置 GPIO2_B4 输出低电平状态
8 gpioset gpiochip2 12=0
9
10 # 获取 GPIO2_B4 输入电平状态
11 gpioget gpiochip2 12
```

2.14.2. UART

主板扩展接口预留了1路UART（TTL电平），对应的pin脚和设备节点如下：

pin脚	功能定义	默认电平类型	设备节点
8	UART_TX	TTL	/dev/ttyS7
10	UART_RX		

```
C |
1 #microcom安装命令
2 root@linaro-alip:~# apt-get install microcom
3 # 测试
4 root@linaro-alip:~# microcom -s 115200 -p /dev/ttyS7
```

2.14.3. I2C

主板扩展接口预留2路I2C，对应的pin脚和设备节点如下：

pin脚	功能定义	设备节点
3	I2C4_SDA	/dev/i2c-4

5	I2C4_SCL	/dev/i2c-7
27	I2C7_SDA	
28	I2C7_SCL	

2.14.4. SPI

主板扩展接口预留1路I2C，对应的pin脚和设备节点如下：

pin脚	功能定义	设备节点
19	SPI_MOSI	/dev/spidev0.0
21	SPI_MISO	
23	SPI_SCLK	
24	SPI_CS0	
26	SPI_CS1	/dev/spidev0.1

2.14.5. POWER_KEY

主板扩展接口第37 pin为power_key，对应的设备节点为/dev/input/event0，输入低电平时系统上报键值KEY_POWER，可以通过evtest用具测试：

```

root@linaro-alip:/# evtest
No device specified, trying to scan all of /dev/input/event*
Available devices:
/dev/input/event0:      rk805 pwrkey
/dev/input/event1:      headset-keys
/dev/input/event2:      rockchip-es8388 Headset
/dev/input/event3:      bt-powerkey
/dev/input/event4:      adc-keys
/dev/input/event5:      rockchip-hdmi rockchip-hdmi
Select the device event number [0-5]: 0
Input driver version is 1.0.1
Input device ID: bus 0x19 vendor 0x0 product 0x0 version 0x0
Input device name: "rk805 pwrkey"
Supported events:
  Event type 0 (EV_SYN)
  Event type 1 (EV_KEY)
    Event code 116 (KEY_POWER)
Properties:
Testing ... (interrupt to exit)
Event: time 1724614534.605352, type 1 (EV_KEY), code 116 (KEY_POWER), value 1
Event: time 1724614534.605352, ----- SYN_REPORT -----
Event: time 1724614535.011677, type 1 (EV_KEY), code 116 (KEY_POWER), value 0
Event: time 1724614535.011677, ----- SYN_REPORT -----
Event: time 1724614537.559125, type 1 (EV_KEY), code 116 (KEY_POWER), value 1
Event: time 1724614537.559125, ----- SYN_REPORT -----
Event: time 1724614537.833156, type 1 (EV_KEY), code 116 (KEY_POWER), value 0
Event: time 1724614537.833156, ----- SYN_REPORT -----

```

2.15. GPU

执行glmark2-es2 开启gpu测试窗口:

```

1 root@linaro-alip:/# glmark2-es2
2 arm_release_ver: g13p0-01eac0, rk_so_ver: 10
3 =====
4     glmark2 2023.01
5     =====
6     OpenGL Information
7     GL_VENDOR:      ARM
8     GL_RENDERER:    Mali-G52
9     GL_VERSION:     OpenGL ES 3.2 v1.g13p0-01eac0.0fd2effaec483a5f4c440d2f
fa25eb7a
10     Surface Config: buf=32 r=8 g=8 b=8 a=8 depth=24 stencil=0 samples=0
11     Surface Size:   800x600 windowed
12     =====
13 ▾ [build] use-vbo=false: FPS: 365 FrameTime: 2.740 ms
14 ▾ [build] use-vbo=true: FPS: 418 FrameTime: 2.395 ms
15 ▾ [texture] texture-filter=nearest: FPS: 467 FrameTime: 2.145 ms
16 ▾ [texture] texture-filter=linear: FPS: 461 FrameTime: 2.172 ms
17 ▾ [texture] texture-filter=mipmap: FPS: 462 FrameTime: 2.165 ms
18 ▾ [shading] shading=gouraud: FPS: 382 FrameTime: 2.622 ms
19 ▾ [shading] shading=blinn-phong-inf: FPS: 407 FrameTime: 2.461 ms
20 ▾ [shading] shading=phong: FPS: 390 FrameTime: 2.566 ms
21 ▾ [shading] shading=cel: FPS: 386 FrameTime: 2.594 ms
22 ▾ [bump] bump-render=high-poly: FPS: 258 FrameTime: 3.881 ms
23 ▾ [bump] bump-render=normals: FPS: 474 FrameTime: 2.111 ms
24 ▾ [bump] bump-render=height: FPS: 472 FrameTime: 2.120 ms
25 ▾ [effect2d] kernel=0,1,0;1,-4,1;0,1,0;: FPS: 390 FrameTime: 2.569 ms
26 ▾ [effect2d] kernel=1,1,1,1,1;1,1,1,1,1;1,1,1,1,1;: FPS: 252 FrameTime: 3.97
5 ms
27 ▾ [pulsar] light=false:quads=5:texture=false: FPS: 484 FrameTime: 2.070 ms
28 ▾ [desktop] blur-radius=5:effect=blur:passes=1:separable=true:windows=4: FPS
: 253 FrameTime: 3.956 ms
29 ▾ [desktop] effect=shadow:windows=4: FPS: 375 FrameTime: 2.668 ms
30 ▾ [buffer] columns=200:interleave=false:update-dispersion=0.9:update-fractio
n=0.5:update-method=map: FPS: 124 FrameTime: 8.085 ms
31 ▾ [buffer] columns=200:interleave=false:update-dispersion=0.9:update-fractio
n=0.5:update-method=subdata: FPS: 121 FrameTime: 8.288 ms
32 ▾ [buffer] columns=200:interleave=true:update-dispersion=0.9:update-fraction
=0.5:update-method=map: FPS: 176 FrameTime: 5.688 ms
33 ▾ [ideas] speed=duration: FPS: 220 FrameTime: 4.552 ms
34 ▾ [jellyfish] <default>: FPS: 341 FrameTime: 2.933 ms
35 ▾ [terrain] <default>: FPS: 67 FrameTime: 14.968 ms
36 ▾ [shadow] <default>: FPS: 261 FrameTime: 3.837 ms
37 ▾ [refract] <default>: FPS: 106 FrameTime: 9.462 ms
38 ▾ [conditionals] fragment-steps=0:vertex-steps=0: FPS: 478 FrameTime: 2.093
ms

```

```

39 [conditionals] fragment-steps=5:vertex-steps=0: FPS: 451 FrameTime: 2.218
40 ms
41 [conditionals] fragment-steps=0:vertex-steps=5: FPS: 471 FrameTime: 2.127
42 ms
43 [function] fragment-complexity=low:fragment-steps=5: FPS: 477 FrameTime:
44 2.097 ms
45 [function] fragment-complexity=medium:fragment-steps=5: FPS: 427 FrameTime
46 : 2.343 ms
47 [loop] fragment-loop=false:fragment-steps=5:vertex-steps=5: FPS: 453 Frame
48 Time: 2.209 ms
[loop] fragment-steps=5:fragment-uniform=false:vertex-steps=5: FPS: 469 Fr
ameTime: 2.135 ms
[loop] fragment-steps=5:fragment-uniform=true:vertex-steps=5: FPS: 442 Fra
meTime: 2.267 ms
=====
                                glmark2 Score: 355
=====

```

2.16. NPU

参考《Purple Pi OH2 YOLOv5开发手册》文档。

使用yolov5进行rknn模型转换测试：

2.16.1. 编译rknn_yolov5_demo：

```

1 root@linaro-alip:/# unzip rknn-toolkit2-master.zip
2 root@linaro-alip:/# cd rknn-toolkit2-master/rknpu2/examples/3rdparty/mpp/L
  inux/aarch64
3 root@linaro-alip:/# ln -srf librockchip_mpp.so.0 librockchip_mpp.so
4 root@linaro-alip:/# ln -srf librockchip_mpp.so.0 librockchip_mpp.so.1
5
6 root@linaro-alip:/# cd -
7 root@linaro-alip:/# cd rknn-toolkit2-master/rknpu2/examples/rknn_yolov5_de
  mo
8 root@linaro-alip:/# chmod a+x build-linux.sh
9 root@linaro-alip:/# export GCC_COMPILER=/usr/bin/aarch64-linux-gnu
10 root@linaro-alip:/mnt/rknn-toolkit2-master/rknpu2/examples/rknn_yolov5_dem
  o# ./build-linux.sh -t rk3576 -a aarch64 -b Release
11 ./build-linux.sh -t rk3576 -a aarch64 -b Release
12 /usr/bin/aarch64-linux-gnu
13 =====
14 TARGET_SOC=RK3576
15 TARGET_ARCH=aarch64
16 BUILD_TYPE=Release
17 BUILD_DIR=/mnt/rknn-toolkit2-master/rknpu2/examples/rknn_yolov5_demo/buil
  d/build_RK3576_linux_aarch64_Release
18 CC=/usr/bin/aarch64-linux-gnu-gcc
19 CXX=/usr/bin/aarch64-linux-gnu-g++
20 =====
21 -- Configuring done
22 -- Generating done
23 -- Build files have been written to: /mnt/rknn-toolkit2-master/rknpu2/exam
  ples/rknn_yolov5_demo/build/build_RK3576_linux_aarch64_Release
24 ▾ [ 60%] Built target rknn_yolov5_video_demo
25 ▾ [100%] Built target rknn_yolov5_demo
26 ▾ [ 40%] Built target rknn_yolov5_demo
27 ▾ [100%] Built target rknn_yolov5_video_demo
28 Install the project...
29 -- Install configuration: "Release"
30 -- Up-to-date: /mnt/rknn-toolkit2-master/rknpu2/examples/rknn_yolov5_demo/
  install/rknn_yolov5_demo_Linux/./rknn_yolov5_demo
31 -- Up-to-date: /mnt/rknn-toolkit2-master/rknpu2/examples/rknn_yolov5_demo/
  install/rknn_yolov5_demo_Linux/lib/librknnrt.so
32 -- Up-to-date: /mnt/rknn-toolkit2-master/rknpu2/examples/rknn_yolov5_demo/
  install/rknn_yolov5_demo_Linux/lib/librga.so
33 -- Up-to-date: /mnt/rknn-toolkit2-master/rknpu2/examples/rknn_yolov5_demo/
  install/rknn_yolov5_demo_Linux/./model/RK3576
34 -- Up-to-date: /mnt/rknn-toolkit2-master/rknpu2/examples/rknn_yolov5_demo/
  install/rknn_yolov5_demo_Linux/./model/RK3576/yolov5s-640-640.rknn
35

```

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

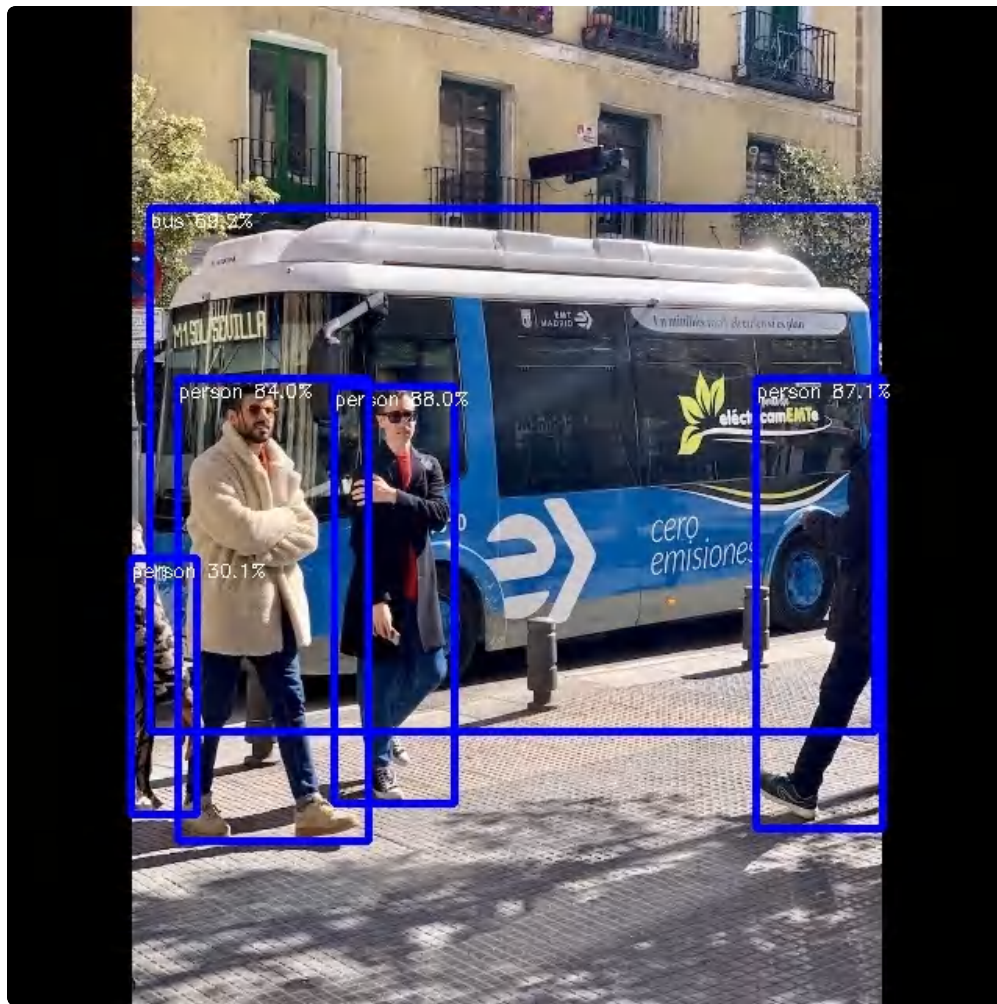
2.16.2. 测试：

```

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

```

执行完后会得到转换后的模型图片，如下所示：

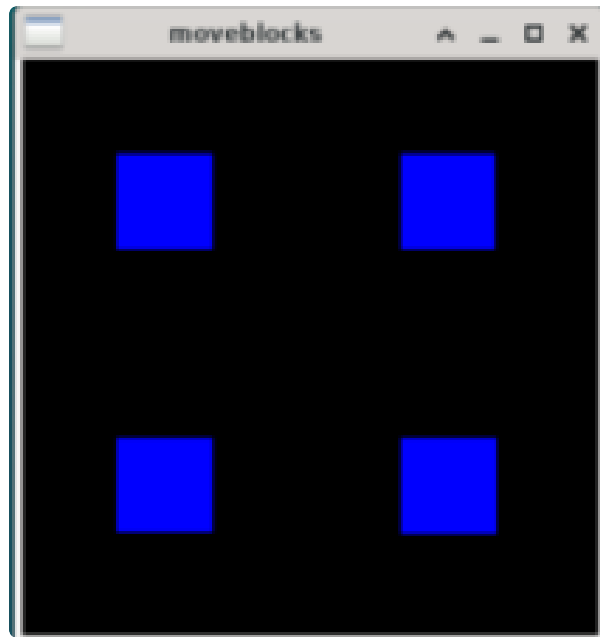


2.17. QT5

支持qt5.15，测试方法如下：

```
▼ Shell |
1 root@linaro-alip:/# source /usr/local/qt5/env.sh
2 root@linaro-alip:/# /usr/local/qt5/example/moveblocks
```

结果如下所示：



2.18. USB 摄像头

2.18.1. 确认USB摄像头设备节点

```
▼ Shell
1 root@linaro-alip:/# grep '' /sys/class/video4linux/video*/name
2 ...
3 /sys/class/video4linux/video73/name:Webcam gadget: UVC Color Camera
4 /sys/class/video4linux/video74/name:Webcam gadget: UVC Color Camera
5 ...
```

从以上信息得知USB摄像头设备节点为/dev/video73和/dev/video74，一般取较小的一个节点即/dev/video73。

2.18.2. 查看USB摄像头支持的图像格式

```

1 root@linaro-alip:/# v4l2-ctl -d /dev/video73 --list-formats-ext
2 ioctl: VIDIOC_ENUM_FMT
3     Type: Video Capture
4
5     [0]: 'MJPG' (Motion-JPEG, compressed)
6         Size: Discrete 1920x1080
7             Interval: Discrete 0.033s (30.000 fps)
8             Interval: Discrete 0.040s (25.000 fps)
9             Interval: Discrete 0.050s (20.000 fps)
10            Interval: Discrete 0.067s (15.000 fps)
11            Interval: Discrete 0.100s (10.000 fps)
12            Interval: Discrete 0.200s (5.000 fps)
13        Size: Discrete 1280x720
14            Interval: Discrete 0.033s (30.000 fps)
15            Interval: Discrete 0.040s (25.000 fps)
16            Interval: Discrete 0.050s (20.000 fps)
17            Interval: Discrete 0.067s (15.000 fps)
18            Interval: Discrete 0.100s (10.000 fps)
19            Interval: Discrete 0.200s (5.000 fps)
20        ...
21     [1]: 'YUYV' (YUYV 4:2:2)
22         Size: Discrete 640x480
23             Interval: Discrete 0.033s (30.000 fps)
24             Interval: Discrete 0.067s (15.000 fps)
25             Interval: Discrete 0.100s (10.000 fps)
26         Size: Discrete 1280x720
27             Interval: Discrete 0.067s (15.000 fps)
28             Interval: Discrete 0.100s (10.000 fps)
29             Interval: Discrete 0.200s (5.000 fps)
30         Size: Discrete 1920x1080
31             Interval: Discrete 0.067s (15.000 fps)
32             Interval: Discrete 0.100s (10.000 fps)
33             Interval: Discrete 0.200s (5.000 fps)
34     [2]: 'H264' (H.264, compressed)
35         Size: Discrete 1920x1080
36             Interval: Discrete 0.033s (30.000 fps)
37             Interval: Discrete 0.040s (25.000 fps)
38             Interval: Discrete 0.050s (20.000 fps)
39             Interval: Discrete 0.067s (15.000 fps)
40             Interval: Discrete 0.100s (10.000 fps)
41             Interval: Discrete 0.200s (5.000 fps)
42         Size: Discrete 1280x720
43             Interval: Discrete 0.033s (30.000 fps)
44             Interval: Discrete 0.040s (25.000 fps)
45             Interval: Discrete 0.050s (20.000 fps)

```

```
46 Interval: Discrete 0.067s (15.000 fps)
47 Interval: Discrete 0.100s (10.000 fps)
48 Interval: Discrete 0.200s (5.000 fps)
49 ...
```

从以上信息得知USB摄像头输出图像支持以下格式：

1. MJPG 1920x1080 30fps
2. MJPG 1280x720 30fps
3. YUYV 640x480 30fps
4. YUYV 1280x720 15fps
5. H264 1920x1080 30fps
6. H264 1280x720 30fps

2.18.3. 预览USB摄像头图像

使用gststreamer预览USB摄像头图像。

2.18.3.1. MJPG格式

由于USB摄像头输出图像是MJPG格式，需指定使用mppjpegdec解码器：

```
1 gst-launch-1.0 v4l2src device=/dev/video73 ! image/jpeg,width=1920,height=1080,framerate=30/1 ! mppjpegdec ! videoconvert ! autovideosink
```

2.18.3.2. YUYV格式

如果USB摄像头输出图像是YUV格式，则不需要指定mppjpegdec解码器：

```
1 gst-launch-1.0 v4l2src device=/dev/video73 ! video/x-raw,format=YUY2, width=640, height=480, framerate=30/1 ! videoconvert ! autovideosink sync=false
```

2.18.3.3. H264格式

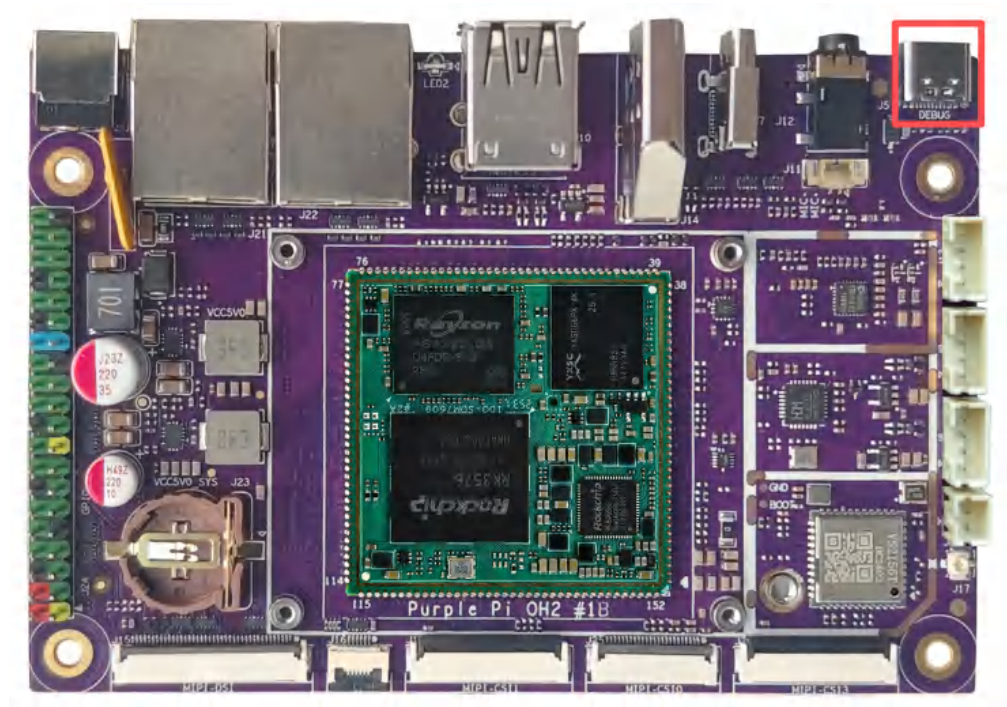
```
1  gst-launch-1.0 v4l2src device=/dev/video73 ! video/x-h264,width=1920,height=1080,framerate=30/1 ! mppvideodec ! videoconvert ! autovideosink
```

Purple Pi OH2 Buildroot系统

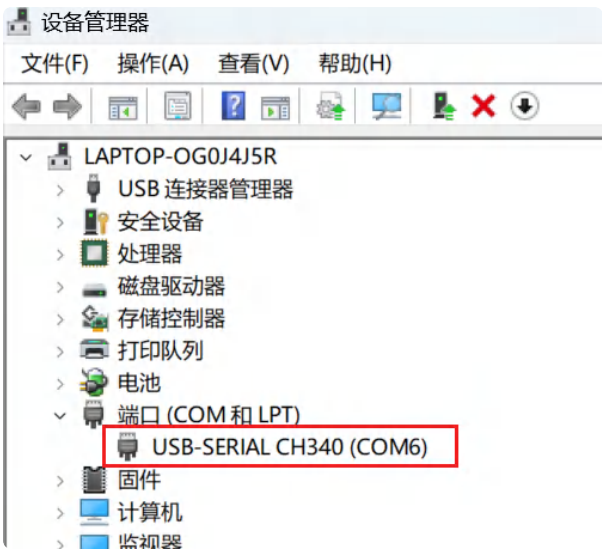
3. 功能及接口使用方法

3.1. DEBUG UART

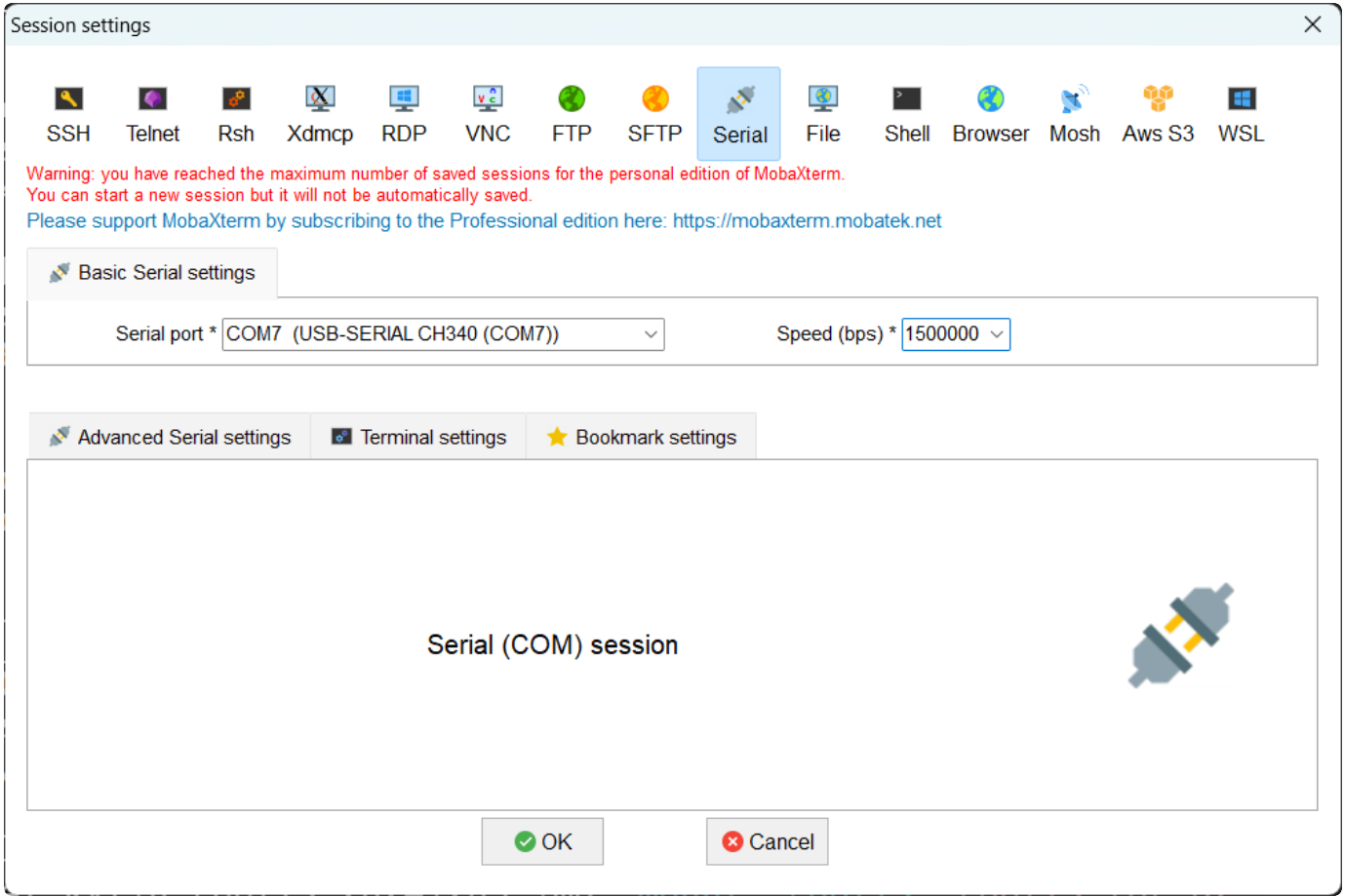
DEBUG UART位置J5，如下图所示：



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



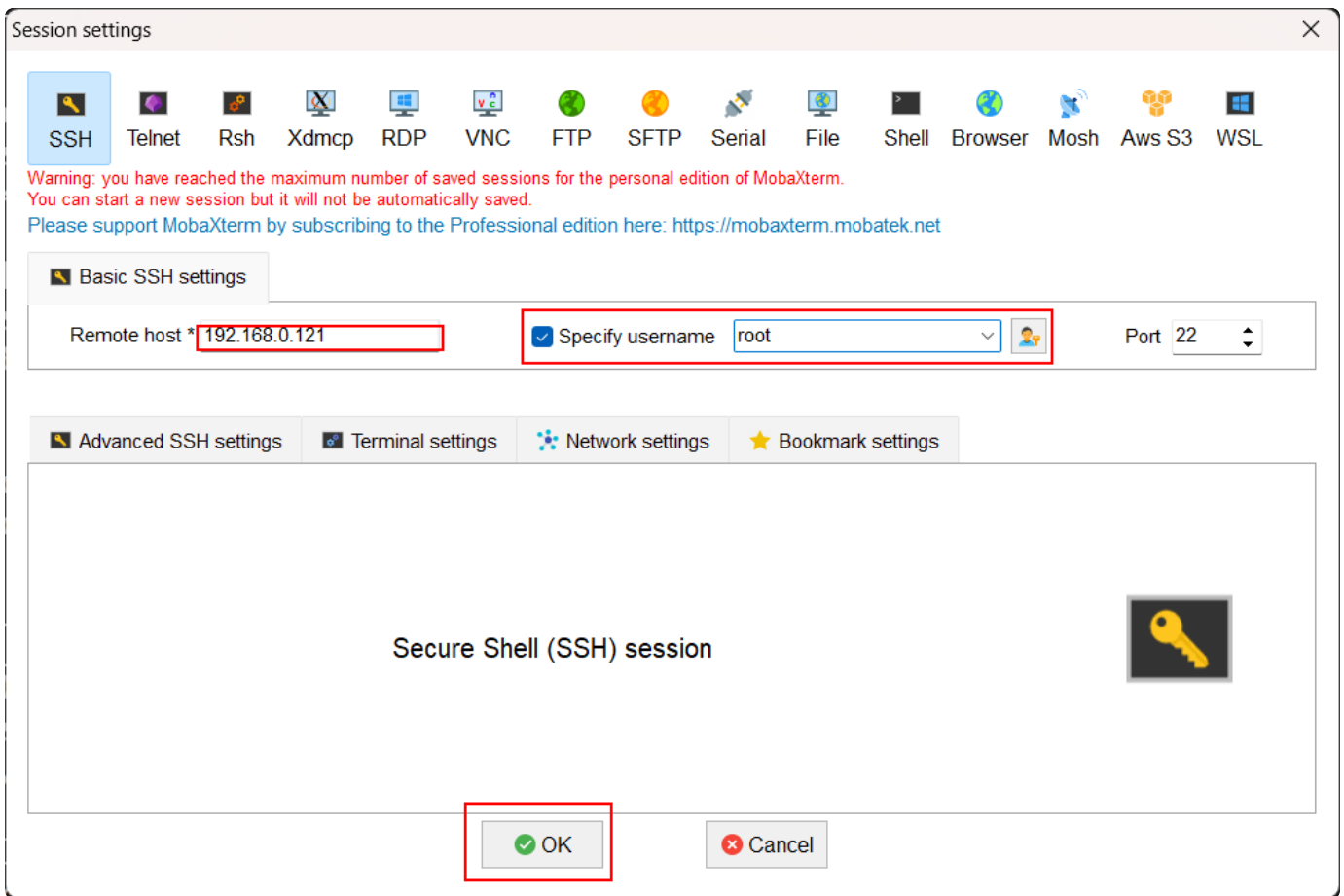
使用调试软件（MobaXterm、putty）等，以MobaXterm为例子，设置参数如下：



SSH登录

在知道IP的前提下，默认可以通过SSH登录进系统。

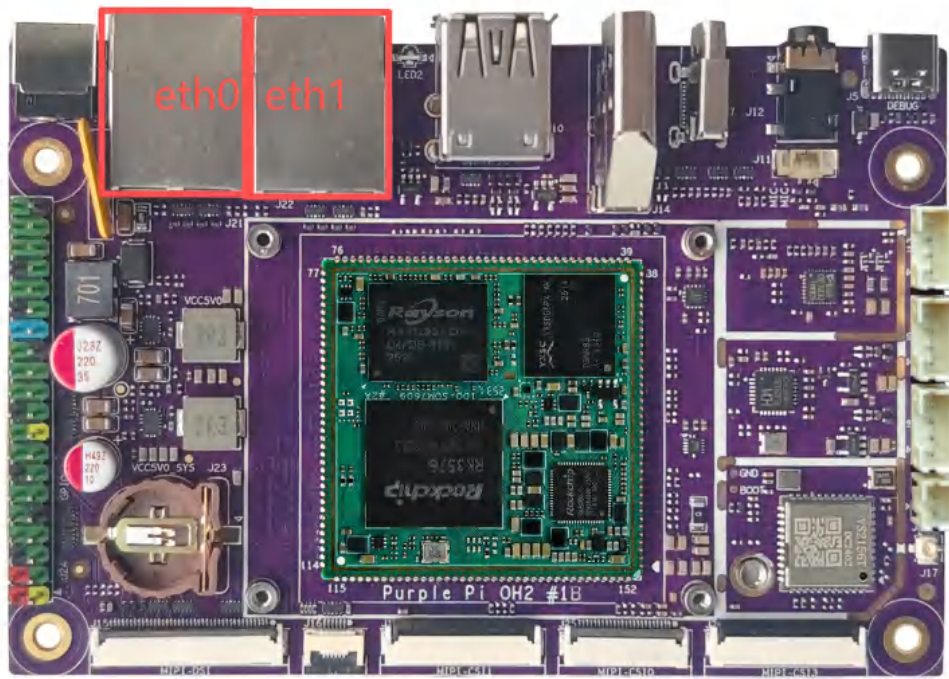
SSH登录账号密码：root/123456



3.2. 网络

3.2.1. 以太网

主板有两路千兆以太网接口位置J28，如下图所示：



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

Ethernet动态IP设置：

```
1 #eth0
2 root@rk3576-buildroot:/# ifconfig eth0
3 eth0      Link encap:Ethernet  HWaddr CE:40:03:D4:14:E5
4           inet addr:192.168.0.121  Bcast:192.168.0.255  Mask:255.255.255.0
5           inet6 addr: fe80::7914:13cf:57f6:547a/64 Scope:Link
6           UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
7           RX packets:62198 errors:0 dropped:1579 overruns:0 frame:0
8           TX packets:615 errors:0 dropped:0 overruns:0 carrier:0
9           collisions:0 txqueuelen:1000
10          RX bytes:5921630 (5.6 MiB)  TX bytes:41773 (40.7 KiB)
11          Interrupt:68
12
13 #eth1
14 root@rk3576-buildroot:/# ifconfig eth1
15 eth1      Link encap:Ethernet  HWaddr D2:40:03:D4:14:E5
16           inet addr:192.168.0.121  Bcast:192.168.0.255  Mask:255.255.255.0
17           inet6 addr: fe80::7f4b:4008:8cda:3e3f/64 Scope:Link
18           UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
19           RX packets:59 errors:0 dropped:1 overruns:0 frame:0
20           TX packets:9 errors:0 dropped:0 overruns:0 carrier:0
21           collisions:0 txqueuelen:1000
22           RX bytes:8690 (8.4 KiB)  TX bytes:1270 (1.2 KiB)
23           Interrupt:70
```

Ethernet静态IP设置:

修改/etc/network/interfaces内容如下:

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

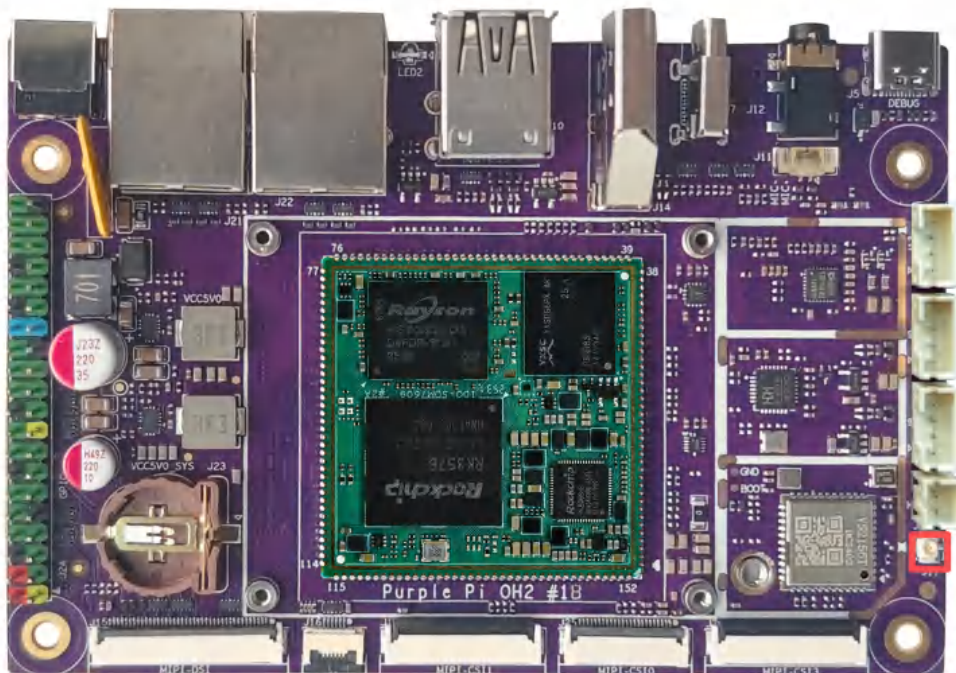
立即生效命令如下：

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

设备断电重启，此静态IP设置仍然生效。

3.2.2. WiFi

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



命令行可以使用nmcli工具连接wifi热点：

添加WiFi账号密码：

```
1 root@rk3576-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
```

WiFi连接：

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

查看连接结果：

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

扫描周围热点:

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

查看wlan0的IP地址, 确认连接成功:

```
1 root@rk3576-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@rk3576-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
```

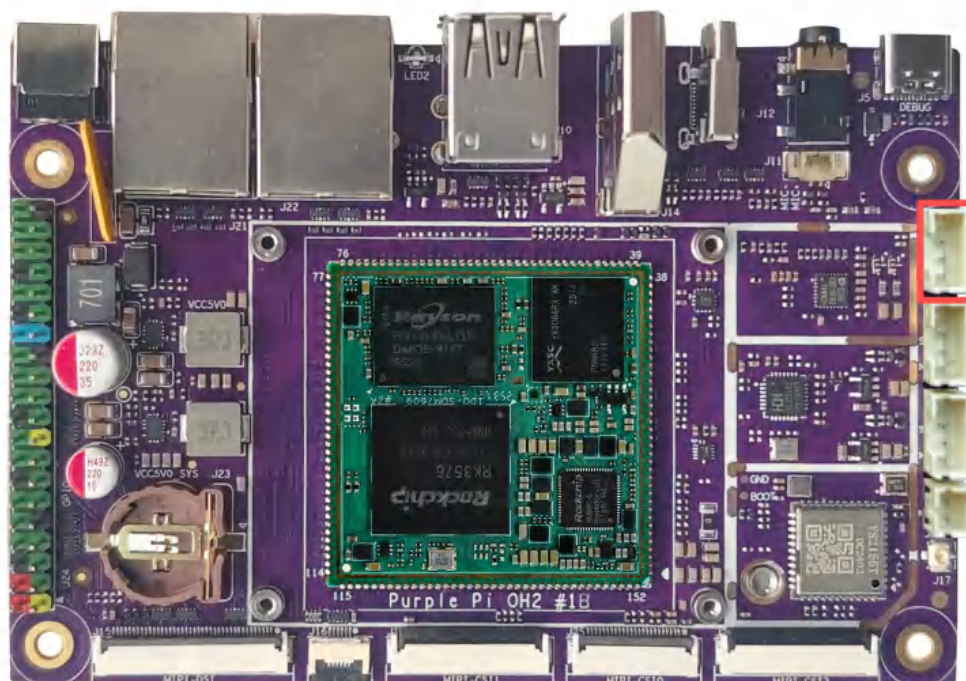
3.2.3. Bluetooth

```
1  #打开蓝牙
2  root@rk3576-buildroot:/# bluetoothctl power on
3  #扫描蓝牙
4  root@rk3576-buildroot:/# bluetoothctl scan on
5  #查看蓝牙设备
6  root@rk3576-buildroot:/# bluetoothctl devices
7  #信任蓝牙设备
8  root@rk3576-buildroot:/# bluetoothctl trust 7C:C1:80:09:DD:6C
9  #蓝牙配对
10 root@rk3576-buildroot:/# bluetoothctl pair 7C:C1:80:09:DD:6C
11 #连接蓝牙
12 root@rk3576-buildroot:/# bluetoothctl connect 7C:C1:80:09:DD:6C
```

3.3. 声音

3.3.1. 喇叭

喇叭接口为PH2.0-4P连接器J13，如下图所示：

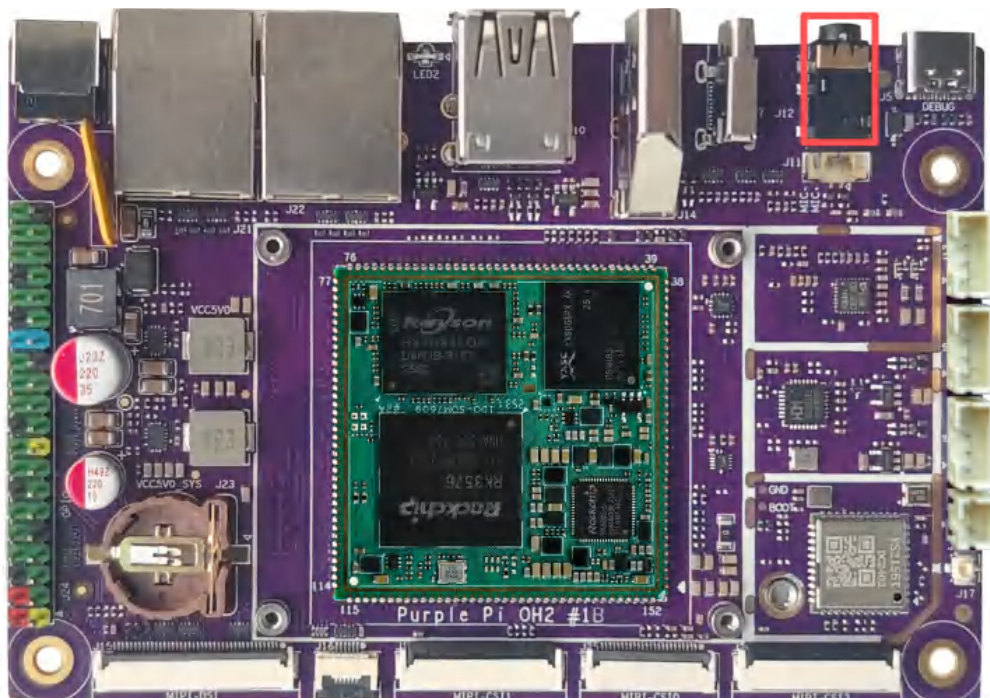


- 支持双声道，每个声道支持4Ω 3W输出

```
1  #查看声卡
2  root@rk3576-buildroot:/# aplay -l
3  **** List of PLAYBACK Hardware Devices ****
4  card 1: rockchipes8388 [rockchip-es8388], device 0: dailink-multicodecs ES
    8323 HiFi-0 [dailink-multicodecs ES8323 HiFi-0]
5      Subdevices: 1/1
6      Subdevice #0: subdevice #0
7  card 2: rockchiphdmi [rockchip-hdmi], device 0: rockchip-hdmi i2s-hifi-0
    [rockchip-hdmi i2s-hifi-0]
8      Subdevices: 1/1
9      Subdevice #0: subdevice #0
10
11 #打开扬声器开关
12 root@rk3576-buildroot:/# amixer -c 0 cset name='Speaker Switch' on
13 numid=66,iface=MIXER,name='Speaker Switch'
14     ; type=BOOLEAN,access=rw-----,values=1
15     : values=on
16 #播放音频
17 root@linaro-alip:/# aplay -D hw:0,0 xihuangni.wav
18 Playing WAVE 'xihuangni.wav' : Signed 16 bit Little Endian, Rate 44100 H
    z, Stereo
```

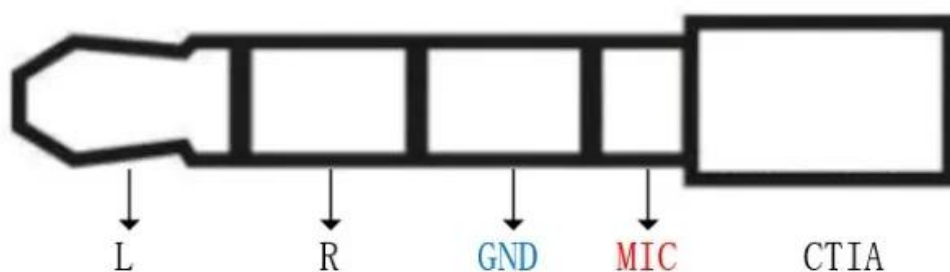
3.3.2. 耳机

主板支持1路3.5mm四节耳机座（CTIA）J12，和1路MX1.25T-5P耳机座子并用，用户可根据需求选用其中1路耳机座子使用，如下图所示：



- 支持耳机检测
- 支持耳机录音

CTIA标准四段式耳机，定义如下：



耳机录音：

▼ Shell |

```

1 root@rk3576-buildroot:/# amixer -c 0 cset numid=67 off
2 root@rk3576-buildroot:/# echo -n "mic1" > /sys/class/es8388_mic/mic_channel
3 root@rk3576-buildroot:/# arecord -D hw:0,0 -f S16_LE -r 16000 -c 2 test_mic
  1.wav
  
```

播放音频到耳机：

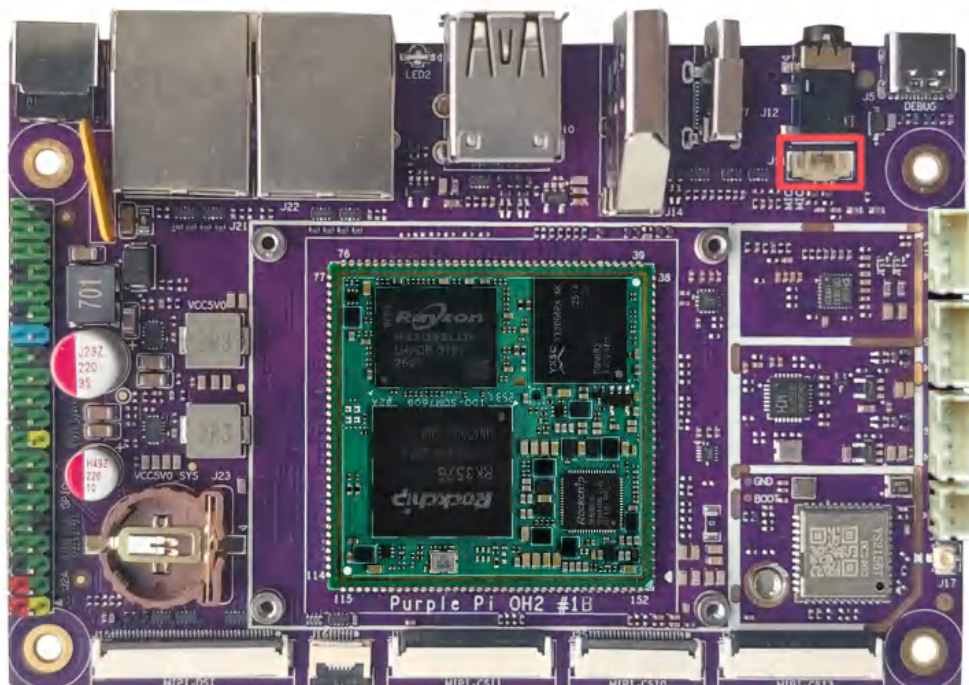
▼ Shell |

```

1 root@rk3576-buildroot:/# amixer -c 0 cset name='Speaker Switch' on
2 root@rk3576-buildroot:/# aplay -D hw:0,0 xihuangni.wav
  
```

3.3.3. Mic

MX1.25T-2P麦克风接口J11，如下图所示：



```
1  #查看录音设备
2  root@rk3576-buildroot:/# arecord -l
3  **** List of CAPTURE Hardware Devices ****
4  card 0: rockchip8388 [rockchip-es8388], device 0: dailink-multicodecs ES
   8323 HiFi-0 [dailink-multicodecs ES8323 HiFi-0]
5      Subdevices: 1/1
6      Subdevice #0: subdevice #0
7  card 1: rockchiphdmi [rockchip-hdmi], device 0: rockchip-hdmi i2s-hifi-0
   [rockchip-hdmi i2s-hifi-0]
8      Subdevices: 1/1
9      Subdevice #0: subdevice #0
10
11
12
13  #捕获音量设置(不要设置最大值):
14  root@rk3576-buildroot:/# amixer -c 0 cset numid=50 100
15  numid=50,iface=MIXER,name='Capture Digital Volume'
16      ; type=INTEGER,access=rw---R--,values=2,min=0,max=192,step=0
17      : values=100,100
18      | dBscale-min=-96.00dB,step=0.50dB,mute=1
19
20  root@rk3576-buildroot:/# amixer -c 0 cget numid=52
21  numid=52,iface=MIXER,name='Left Channel Capture Volume'
22      ; type=INTEGER,access=rw---R--,values=1,min=0,max=8,step=0
23      : values=0
24      | dBscale-min=0.00dB,step=3.00dB,mute=0
25
26  root@rk3576-buildroot:/# amixer -c 0 cget numid=53
27  numid=53,iface=MIXER,name='Right Channel Capture Volume'
28      ; type=INTEGER,access=rw---R--,values=1,min=0,max=8,step=0
29      : values=0
30      | dBscale-min=0.00dB,step=3.00dB,mute=0
31
32  自动增益控制(打开):
33  root@rk3576-buildroot:/# amixer -c 0 cset numid=41 3
34  numid=41,iface=MIXER,name='ALC Capture Function'
35      ; type=ENUMERATED,access=rw-----,values=1,items=4
36      ; Item #0 'Off'
37      ; Item #1 'Right'
38      ; Item #2 'Left'
39      ; Item #3 'Stereo'
40      : values=3
41
42  调整目标音量:
43  root@rk3576-buildroot:/# amixer -c 0 cset numid=38 8
```

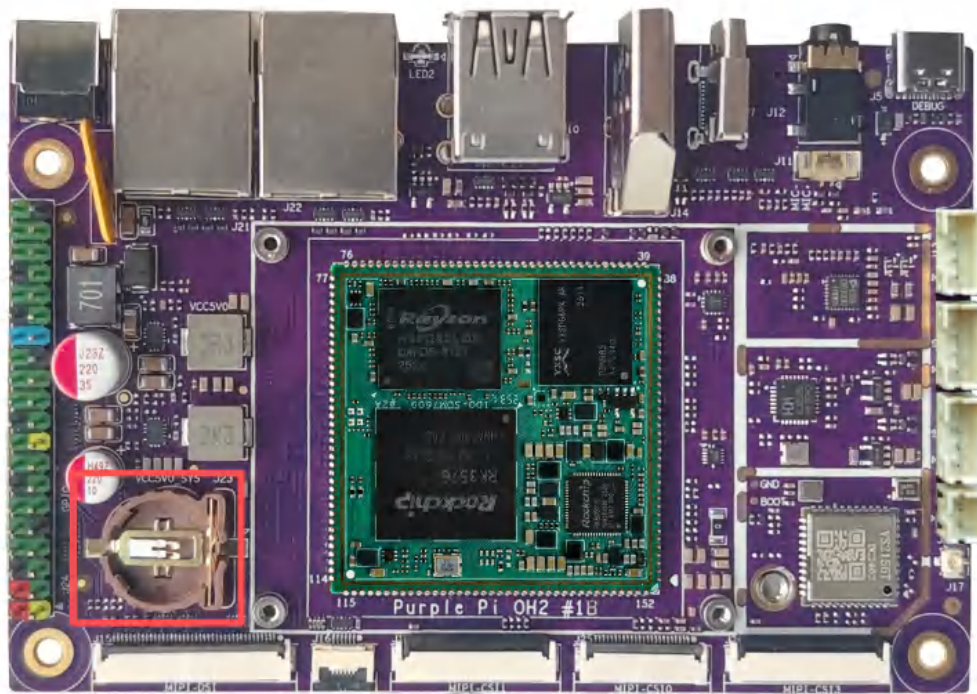
```

44 numid=38,iface=MIXER,name='ALC Capture Target Volume'
45 ; type=INTEGER,access=rw-----,values=1,min=0,max=15,step=0
46 : values=8
47 设置噪声门限值:
48 root@rk3576-buildroot:/# amixer -c 0 cset numid=46 15
49 numid=46,iface=MIXER,name='ALC Capture NG Threshold'
50 ; type=INTEGER,access=rw-----,values=1,min=0,max=31,step=0
51 : values=15
52 确保捕获静音未开启:
53 root@rk3576-buildroot:/# amixer -c 0 cset numid=51 off
54 numid=51,iface=MIXER,name='Capture Mute'
55 ; type=BOOLEAN,access=rw-----,values=1
56 : values=off
57
58 #主mic开关 (mic录音需要打开, 耳机录音需要关闭):
59 root@rk3576-buildroot:/# amixer -c 0 cset numid=67 on
60 numid=67,iface=MIXER,name='Main Mic Switch'
61 ; type=BOOLEAN,access=rw-----,values=1
62 : values=on
63 root@rk3576-buildroot:/# echo -n "mic2" > /sys/class/es8388_mic/mic_channel
64 root@rk3576-buildroot:/# arecord -D hw:0,0 -f S16_LE -r 16000 -c 2 test_mic2.wav
65
66
67 #耳机录音:
68 root@rk3576-buildroot:/# amixer -c 0 cset numid=67 off
69 root@rk3576-buildroot:/# echo -n "mic1" > /sys/class/es8388_mic/mic_channel
70 root@rk3576-buildroot:/# arecord -D hw:0,0 -f S16_LE -r 16000 -c 2 test_mic1.wav

```

3.4. RTC

MX1.25T-2P RTC电池座J23, 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：

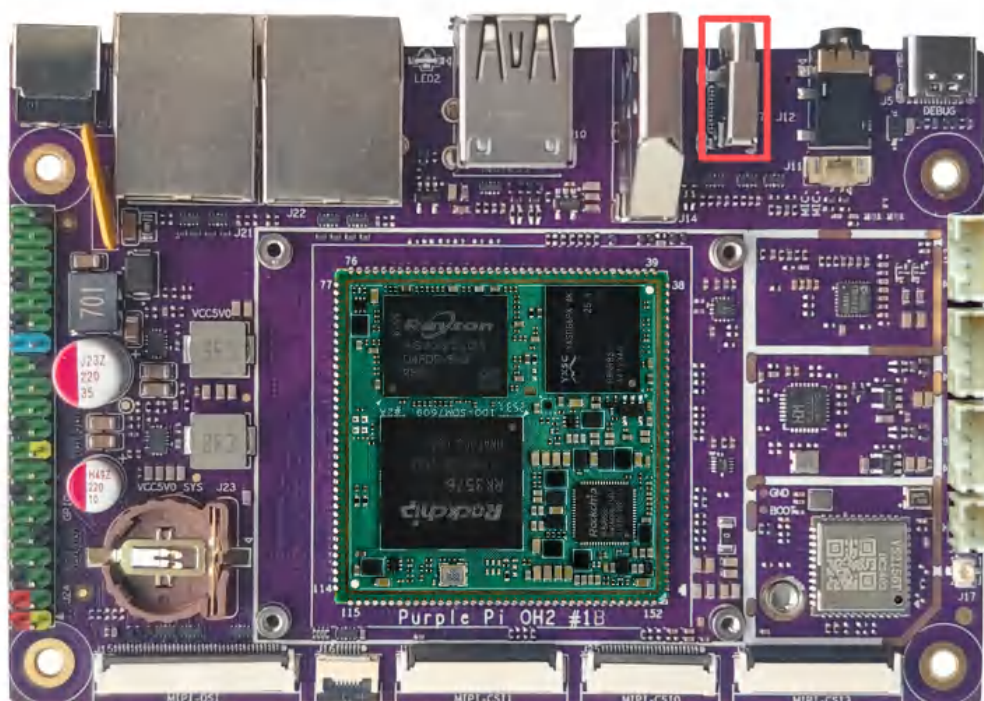
```
1  #设置时间
2  root@rk3576-buildroot:/# date -s "2024-10-09 14:02:30"
3
4  #将rtc时钟调整为与目前的系统时钟一致
5  root@rk3576-buildroot:/# hwclock -w
6
7  #获取硬件rtc当前时间（断电重启读取时间没有太大偏差）
8  root@rk3576-buildroot:/# hwclock -r
9  2024-10-09 14:02:35.945604+00:00
```

3.5. USB接口

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

3.5.1. TypeC接口

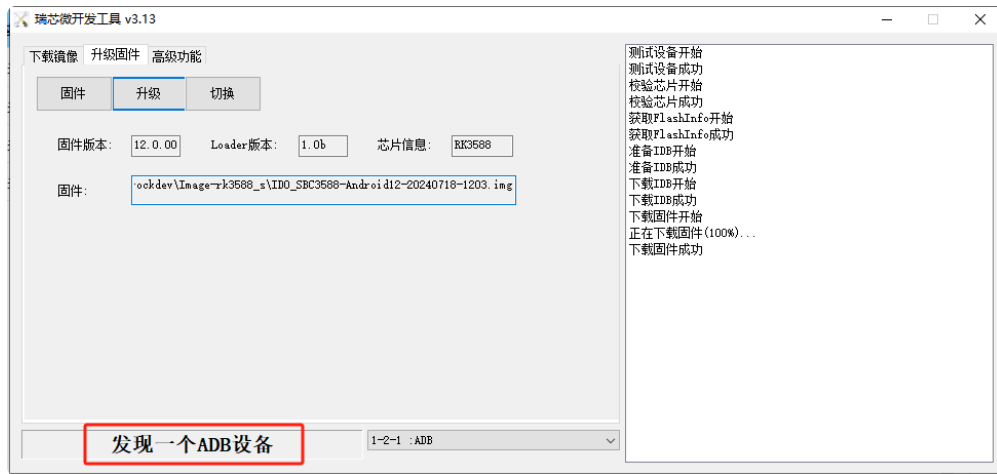
TypeC接口（USB3.2 Gen1 OTG+DP1.4输出）J7，如下图所示：



- 支持Host、Device模式自动切换
- 支持DP显示输出

Device从机模式

使用TypeC数据线连接电脑，烧录工具能发现一个ADB设备，如下图所示：



- 可使用ADB相关开发工具对盒子进行功能调试

Host主机模式

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



- 可识别U盘、键盘、鼠标并正常使用

DP模式

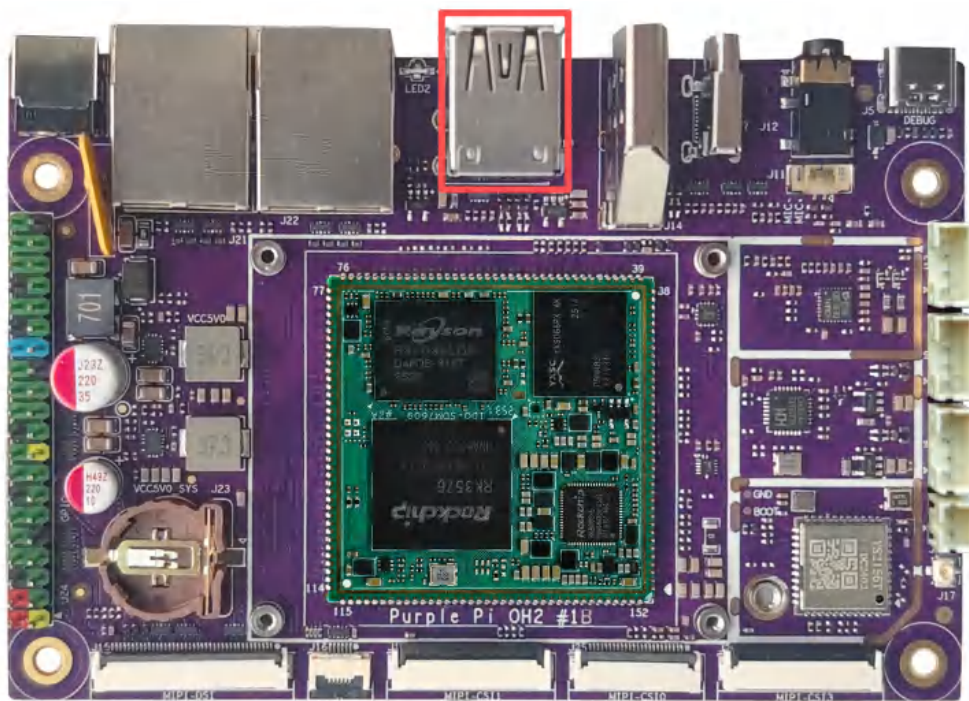
通过TypeC全功能数据线接入DP显示器，或通过TYPE-C to HDMI数据线连接HDMI显示器



- 主板画面通过TYPE-C DP输出正常，DP声音输出正常

3.5.2. USB3.0接口

主板支持1个USB3.0 TYPE-A 接口，位于J10（下），如下图所示：



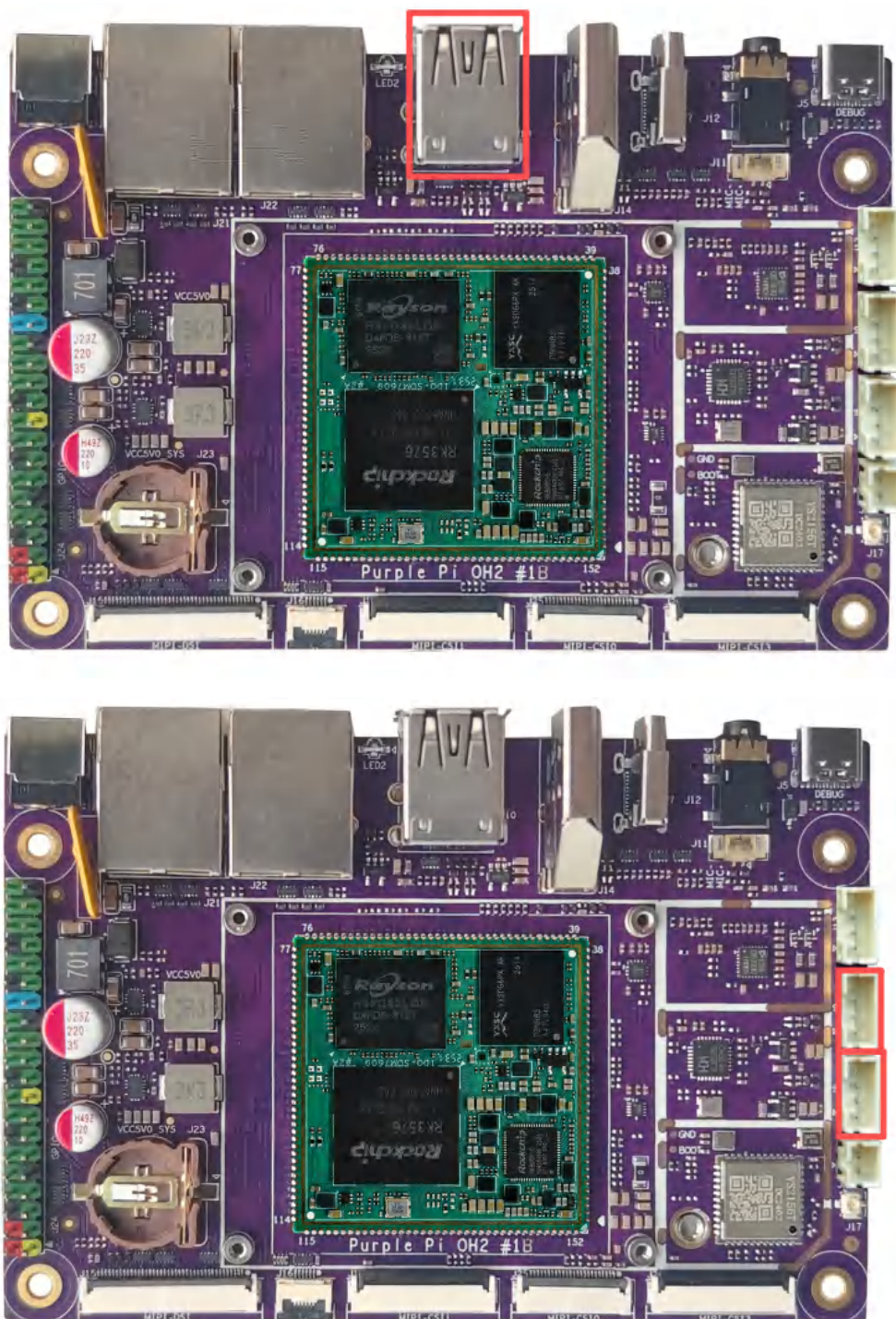
USB母座提供5V@1A供电能力，2个USB端口供电控制是共用的，USB电源控制，如下表所示：

USB端口	默认状态	动作	命令
USB1/2	开启	关闭电源	echo 0 > /sys/class/leds/usb_pwr/brightness
		开启电源	echo 1 > /sys/class/leds/usb_pwr/brightness

供电控制说明：设备节点写"0"关闭电源，写"1"开启电源

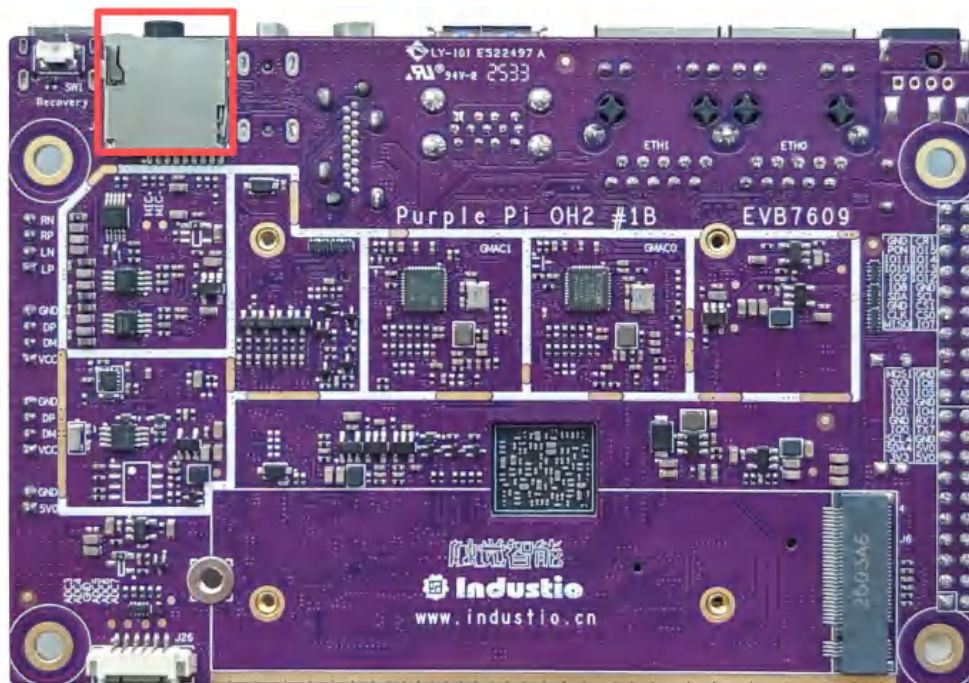
3.5.3. USB2.0接口

主板配置了1路USB2.0 TYPE-A接口，位于J10（上），及2路PH2.0-4P接口，位于J8和J9，USB接口均提供5V@1A的驱动能力，如下图所示：



3.6. TF Card

TF卡座支持SDIO3.0，支持高速SD卡，接口位于主板背面J4，如下图所示：



TF卡安装方向，主板正面朝上时TF卡触点朝上。

Shell

```

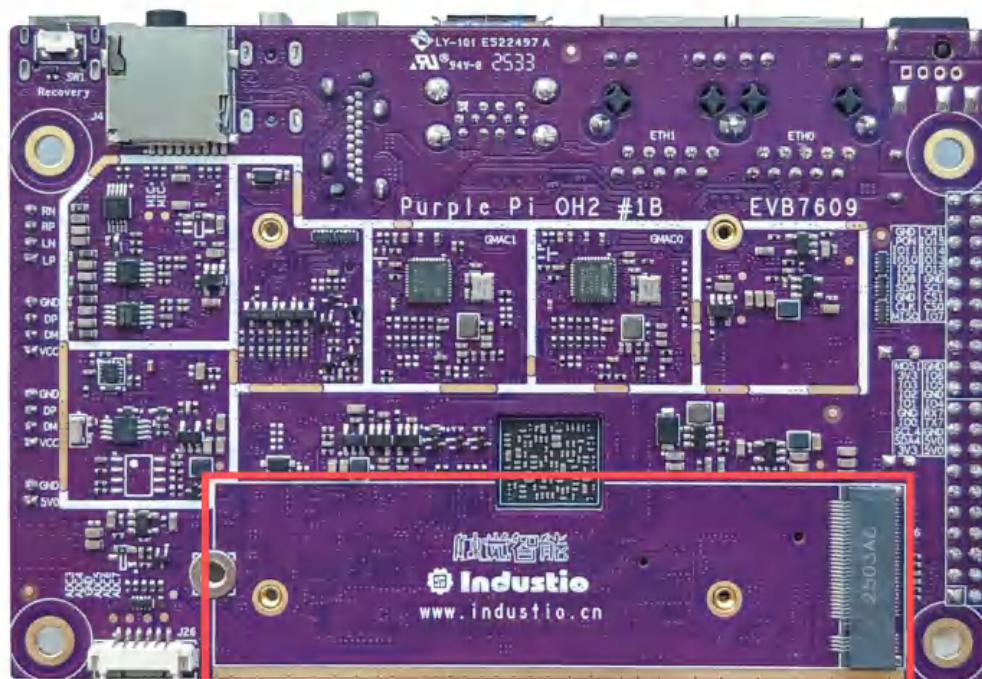
1 root@rk3576-buildroot:/# fdisk -l
2 Found valid GPT with protective MBR; using GPT
3
4 Disk /dev/mmcblk2: 61071360 sectors, 1148M
5 Logical sector size: 512
6 Disk identifier (GUID): 21130000-0000-4444-8000-05e1000030fd
7 Partition table holds up to 128 entries
8 First usable sector is 34, last usable sector is 61071326
9
10 Number  Start (sector)    End (sector)  Size Name
11      1             16384          24575  4096K uboot
12      2             24576          32767  4096K misc
13      3             32768         163839  64.0M boot
14      4             163840         425983  128M recovery
15      5             425984         491519  32.0M backup
16      6             491520         8880127  4096M userdata
17      7             8880128         9142271  128M oem
18      8             9142272         61071295  24.7G rootfs
19 ...
20 Device      Boot StartCHS      EndCHS          StartLBA      EndLBA      Sector
   s  Size Id Type
21 /dev/mmcblk1p1  0,2,10      239,254,63      135      3858623      3858489
    1884M  e Win95 FAT16 (LBA)

```

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

3.7. M.2接口

主板上使用标准M.2-M-key M.2接口连接座，如下图所示：



支持PCIe2.1通信，适用2280尺寸NVME固态硬盘。

```

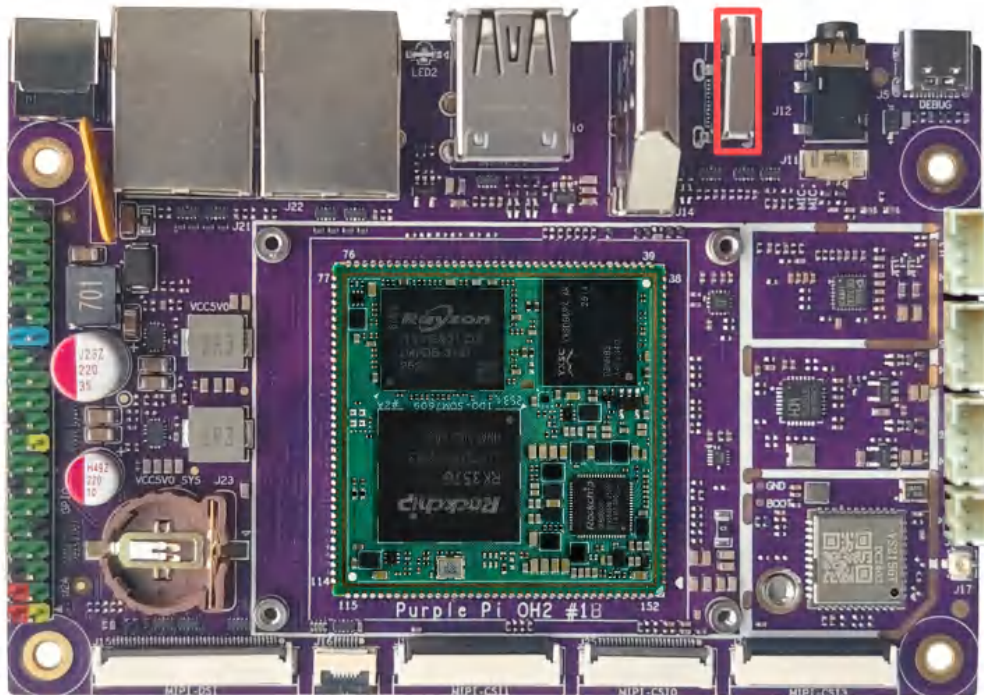
1  root@rk3576-buildroot:/# fdisk -l
2  Disk /dev/ram0: 8 MiB, 8388608 bytes, 16384 sectors
3  Units: sectors of 1 * 512 = 512 bytes
4  Sector size (logical/physical): 512 bytes / 4096 bytes
5  I/O size (minimum/optimal): 4096 bytes / 4096 bytes
6
7
8  Disk /dev/ram1: 8 MiB, 8388608 bytes, 16384 sectors
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)

```

3.8. LCD显示

3.8.1. DP

DP接口 (USB-C) J7, 如下图所示:



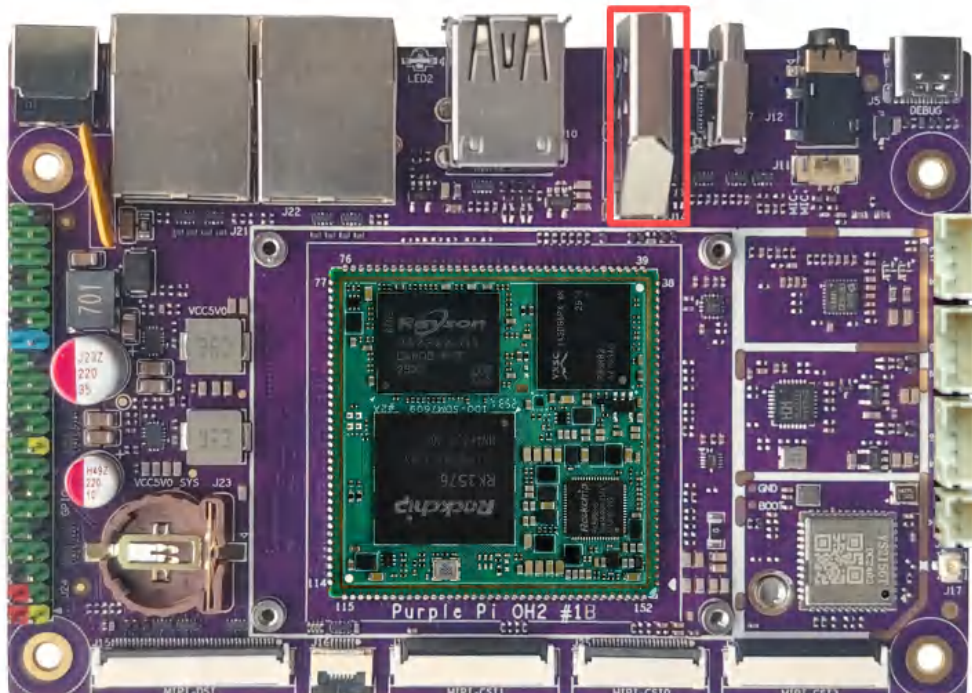
可以使用 USB-C 转 HDMI 高清线连接 HDMI 显示器输出画面，如下图所示：



- 最高支持4K@30fps输出

3.8.2. HDMI-TX

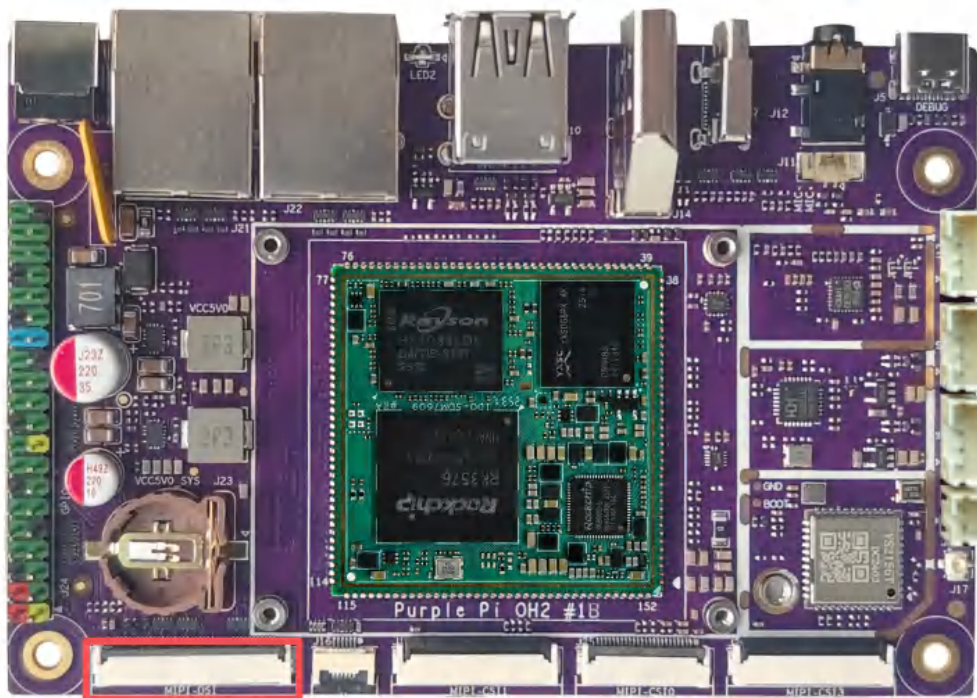
标准HDMI-A接口J14，如下图所示：



- 最高支持 HDMI2.1 8K@60fps 输出

3.8.3. MIPI

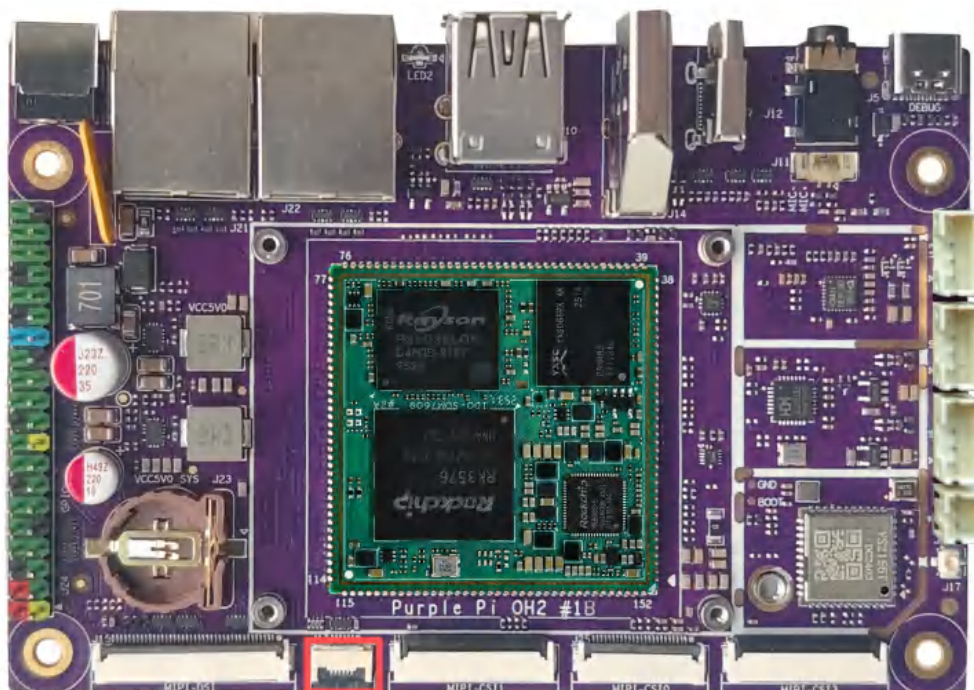
J15 MIPI接口为40Pin FPC 0.5mm 上接，如下图所示：



- MIPI供电为3.3V

3.9. TP接口

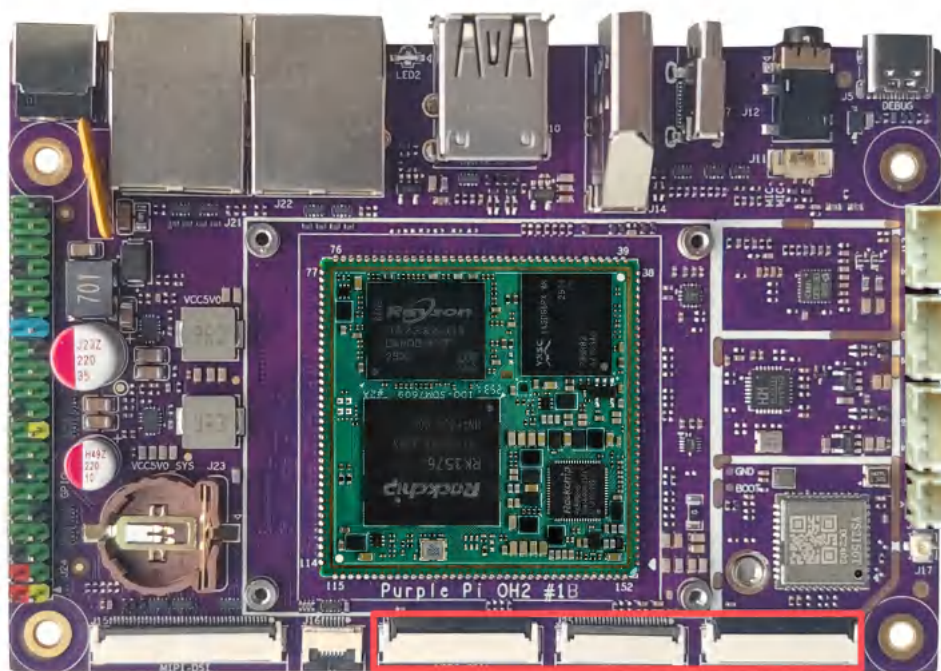
TP接口为(J16) 6Pin FPC 0.5mm座子，如下图所示：



- 触摸 TP 接口接线方法为下接

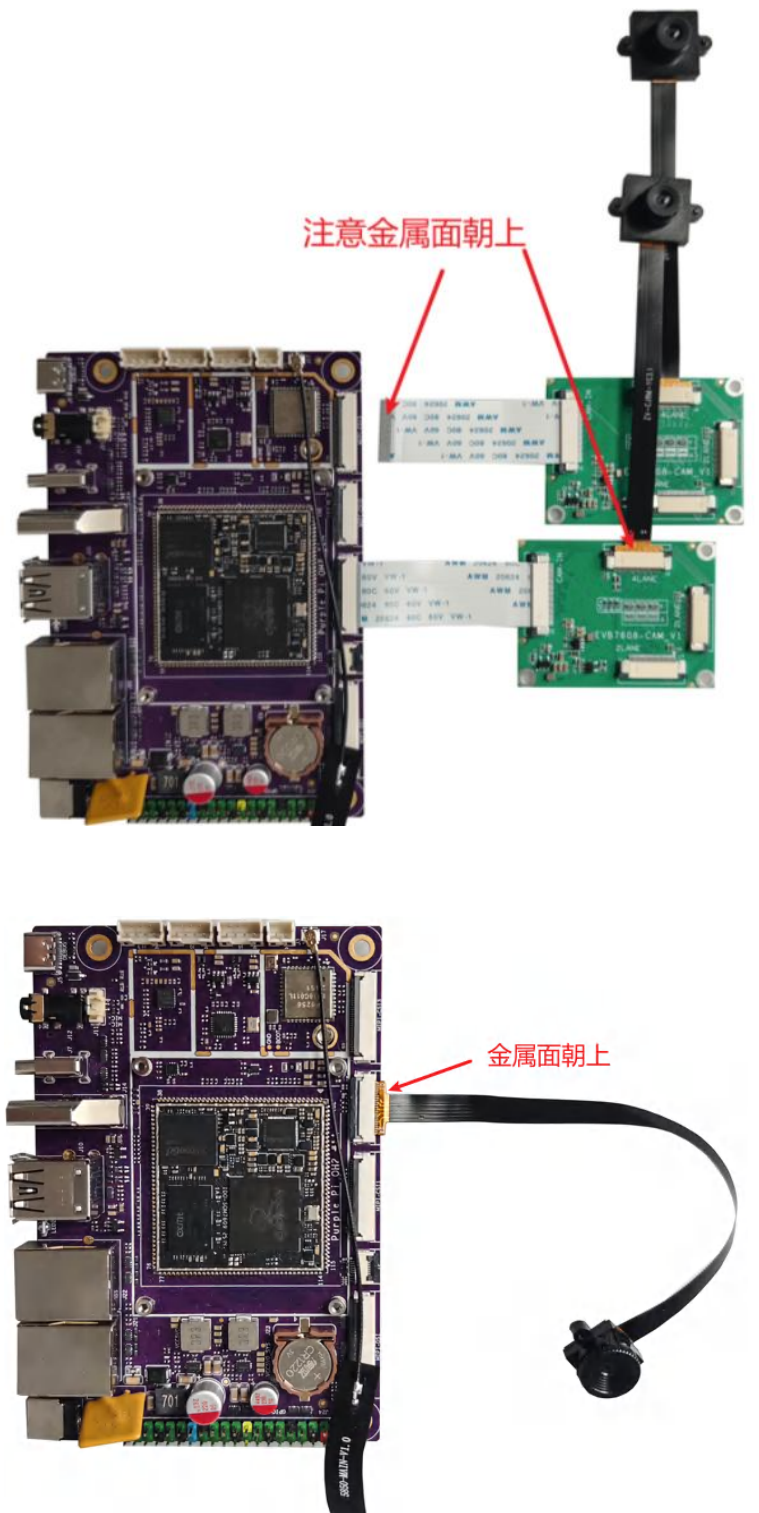
3.10. MIPI Camera

主板共有3路摄像头接口，分别是J19、J20和J25，如下图所示：



J19和J20使用FPC05-PUW-32P抽拉式上接座子，J25使用FPC05-PUW-24P抽拉式上接座子。其中J19和J25默认适配IMX415摄像头，J20默认适配OV13855摄像头。

接法如下图所示：



3.10.1. 预览命令：

预览J19摄像头：

```

1 root@rk3576-buildroot:/# export DISPLAY=:0
2 root@rk3576-buildroot:/# gst-launch-1.0 v4l2src device=/dev/video42 ! vide
o/x-raw, format=NV12, width=1920, height=1080, framerate=30/1 ! videoconver
t ! autovideosink sync=false

```

预览J20摄像头:

```

1 root@rk3576-buildroot:/# export DISPLAY=:0
2 root@rk3576-buildroot:/# gst-launch-1.0 v4l2src device=/dev/video51 ! vide
o/x-raw, format=NV12, width=1920, height=1080, framerate=30/1 ! videoconver
t ! autovideosink sync=false

```

预览J25摄像头:

```

1 root@rk3576-buildroot:/# export DISPLAY=:0
2 root@rk3576-buildroot:/# gst-launch-1.0 v4l2src device=/dev/video33 ! vide
o/x-raw, format=NV12, width=1920, height=1080, framerate=30/1 ! videoconver
t ! autovideosink sync=false

```

3.10.2. 抓图:

抓取J19摄像头图片:

```

1 root@rk3576-buildroot:/# v4l2-ctl --verbose -d /dev/video42 --set-fmt-video
=width=1920,height=1080,pixelformat='NV12' --stream-mmap=4 --stream-skip=3
--stream-to=./test-1920x1080.yuv --stream-count=1 --stream-poll

```

抓取J20摄像头图片:

```

1 root@rk3576-buildroot:/# v4l2-ctl --verbose -d /dev/video51 --set-fmt-video
=width=1920,height=1080,pixelformat='NV12' --stream-mmap=4 --stream-skip=3
--stream-to=./test-1920x1080.yuv --stream-count=1 --stream-poll

```

抓取J25摄像头图片:

▼ Plain Text |

```
1 root@rk3576-buildroot:/# v4l2-ctl --verbose -d /dev/video33 --set-fmt-video=width=1920,height=1080,pixelformat='NV12' --stream-mmap=4 --stream-skip=3 --stream-to=./test-1920x1080.yuv --stream-count=1 --stream-poll
```

查看照片：

▼ Plain Text |

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

3.10.3. 抓视频：

抓取J19摄像头视频：

▼ Plain Text |

```
1 root@rk3576-buildroot:/# v4l2-ctl --stream-mmap=4 -d /dev/video42 --set-fmt-video=width=1920,height=1080,pixelformat='NV12' --stream-to=./output.yuv
```

抓取J20摄像头视频：

▼ Plain Text |

```
1 root@rk3576-buildroot:/# v4l2-ctl --stream-mmap=4 -d /dev/video51 --set-fmt-video=width=1920,height=1080,pixelformat='NV12' --stream-to=./output.yuv
```

抓取J25摄像头视频：

▼ Plain Text |

```
1 root@rk3576-buildroot:/# v4l2-ctl --stream-mmap=4 -d /dev/video33 --set-fmt-video=width=1920,height=1080,pixelformat='NV12' --stream-to=./output.yuv
```

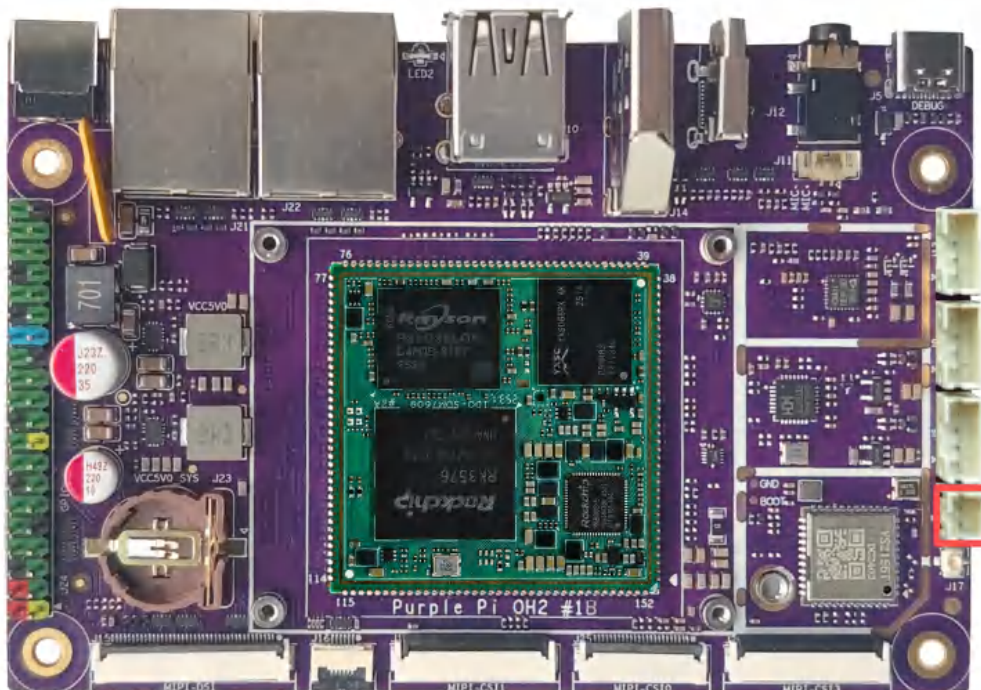
播放视频：

▼ Plain Text |

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

3.11. Fan 风扇

PH2 5V Fan风扇接口J3，如下图所示：



3.12. 三轴传感器

主板配置了1路三轴传感器，传感器型号是SC7A20HTR，在系统中对应的设备节点为/sys/bus/iio/devices/iio:device0。

3.12.1. 设置采样率

默认采样率为1Hz，最高支持400Hz，采样率设置方法如下：

```
1 echo 400 > /sys/bus/iio/devices/iio:device0/sampling_frequency
```

3.12.2. 获取三轴传感器数据

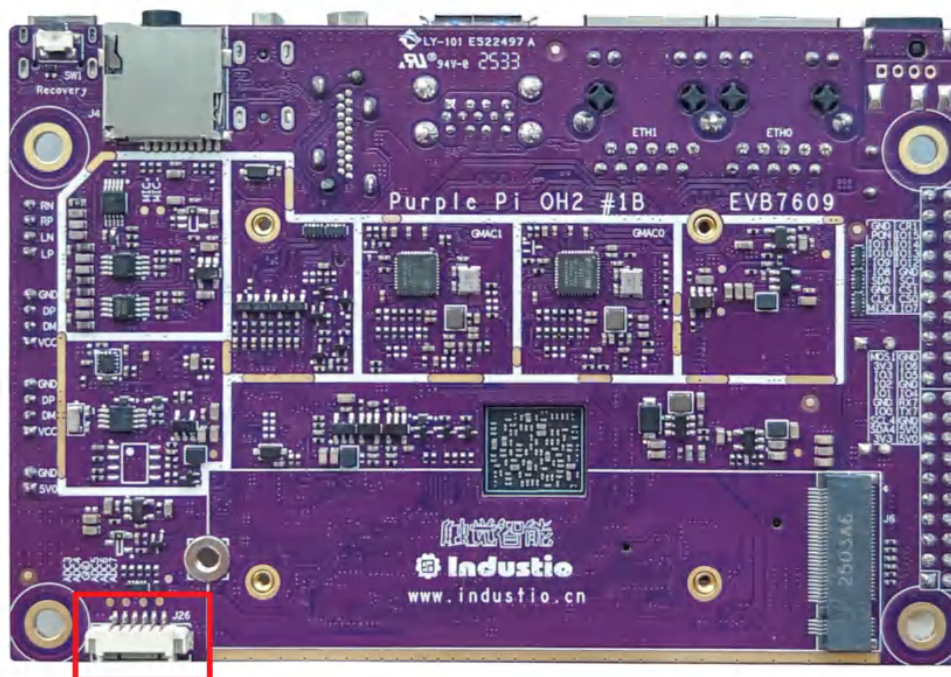
```

1 # 获取X轴加速度
2 cat /sys/bus/iio/devices/iio:device0/in_accel_x_raw
3
4 # 获取Y轴加速度
5 cat /sys/bus/iio/devices/iio:device0/in_accel_y_raw
6
7 # 获取Z轴加速度
8 cat /sys/bus/iio/devices/iio:device0/in_accel_z_raw

```

3.13. ADC

主板ADC接口位于背面J26，支持5路ADC输入，精度是12位，最高输入电压1.8V。



5路ADC对应的设备节点如下表：

pin	名称	设备节点
1	ADC3	/sys/bus/iio/devices/iio:device1/in_voltage3_raw
2	ADC4	/sys/bus/iio/devices/iio:device1/in_voltage4_raw
3	GND	—
4	ADC5	/sys/bus/iio/devices/iio:device1/in_voltage5_raw

5	ADC6	/sys/bus/iio/devices/iio:device1/in_voltage6_raw
6	ADC7	/sys/bus/iio/devices/iio:device1/in_voltage7_raw

3.13.1. 获取输入ADC raw值

通过读取设备节点直接获取ADC raw值，如获取ADC3的raw值：

▼

Shell

```

1 root@rk3576-buildroot:/# cat /sys/bus/iio/devices/iio:device1/in_voltage3_raw
2 3770

```

3.13.2. ADC输入电压计算公式

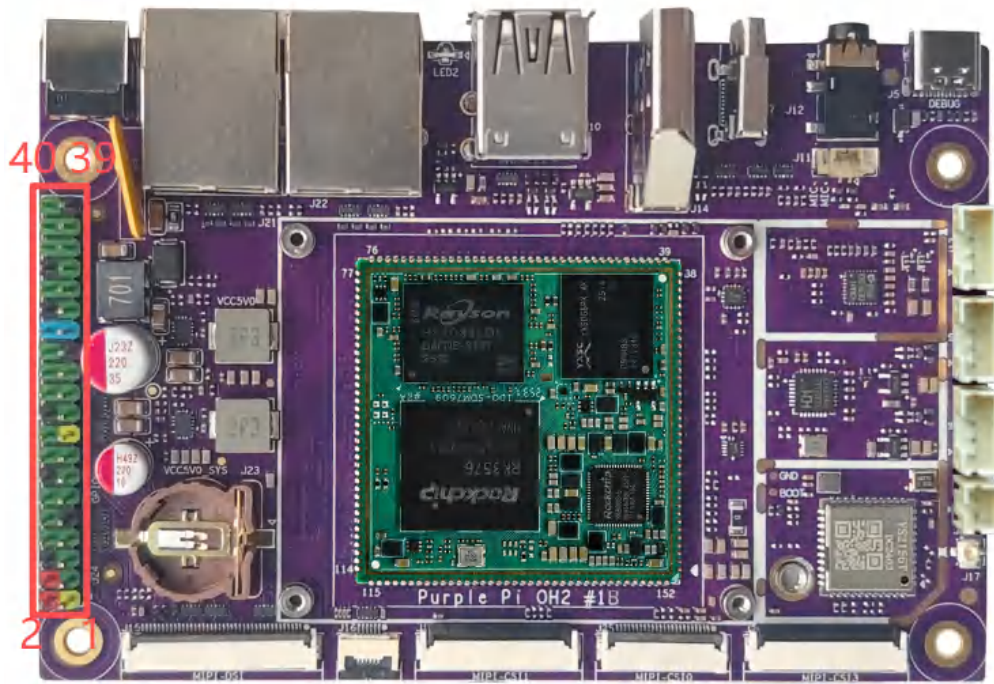
$$V_{in} = (raw/1024) \times 1.8v$$

其中 V_{in} 表示ADC输入电压，raw表示ADC raw值。比如ADC3的raw值为258，则ADC3输入电压为：

$$V_{in} = (3770/4096) \times 1.8v = 1.657v$$

3.14. 扩展接口

主板预留了40pin双排针扩展接口，包括1路UART，1路SPI，2路I2C和16个GPIO。40pin双排针引脚定义如下：



pin	功能	pin	功能
1	3.3V	2	5V
3	I2C4_SDA	4	5V
5	I2C4_SCL	6	GND
7	GPIO76	8	UART7_TX
9	GND	10	UART7_RX
11	GPIO75	12	GPIO77
13	GPIO72	14	GND
15	GPIO73	16	GPIO74
17	3.3V	18	GPIO136
19	SPI0_MOSI	20	GND
21	SPI0_MISO	22	GPIO137
23	SPI0_SCLK	24	SPI0_CS0
25	GND	26	SPI0_CS1
27	I2C7_SDA	28	I2C7_SCL

29	GPIO56	30	GND
31	GPIO57	32	GPIO132
33	GPIO58	34	GND
35	GPIO59	36	GPIO134
37	POWER_KEY	38	GPIO98
39	GND	40	GPIO99

3.14.1. GPIO

主板扩展接口预留了16个GPIO，各个GPIO引脚定义描述，如下图所示：

序号	pin	GPIO 标号	GPIO NUM	LIBGPIO节点
1	7	GPIO2_B4	76	gpiochip2 12
2	11	GPIO2_B3	75	gpiochip2 11
3	12	GPIO2_B5	77	gpiochip2 13
4	13	GPIO2_B0	72	gpiochip2 8
5	15	GPIO2_B1	73	gpiochip2 9
6	16	GPIO2_B2	74	gpiochip2 10
7	18	GPIO4_B0	136	gpiochip4 8
8	22	GPIO4_B1	137	gpiochip4 9
9	29	GPIO1_D0	56	gpiochip1 24
10	31	GPIO1_D1	57	gpiochip1 25
11	32	GPIO4_A4	132	gpiochip4 4
12	33	GPIO1_D2	58	gpiochip1 26
13	35	GPIO1_D3	59	gpiochip1 27
14	36	GPIO4_A6	134	gpiochip4 6
15	38	GPIO3_A2	98	gpiochip3 2

16	40	GPIO3_A3	99	gpiochip3 3
----	----	----------	----	-------------

GPIO控制，根据上表【LIBGPIO节点】进行命令操作

libgpiod
C

```

1  #安装命令
2  apt install -y libgpiod-dev gpiod
3
4  # 设置 GPIO2_B4 输出高电平状态
5  gpioset gpiochip2 12=1
6
7  # 设置 GPIO2_B4 输出低电平状态
8  gpioset gpiochip2 12=0
9
10 # 获取 GPIO2_B4 输入电平状态
11 gpioget gpiochip2 12

```

3.14.2. UART

主板扩展接口预留了1路UART（TTL电平），对应的pin脚和设备节点如下：

pin脚	功能定义	默认电平类型	设备节点
8	UART_TX	TTL	/dev/ttyS7
10	UART_RX		

C

```

1  #microcom安装命令
2  root@rk3576-buildroot:/# apt-get install microcom
3  # 测试
4  root@rk3576-buildroot:/# microcom -s 115200 -p /dev/ttyS7

```

3.14.3. I2C

主板扩展接口预留2路I2C，对应的pin脚和设备节点如下：

pin脚	功能定义	设备节点
3	I2C4_SDA	/dev/i2c-4

5	I2C4_SCL	/dev/i2c-7
27	I2C7_SDA	
28	I2C7_SCL	

3.14.4. SPI

主板扩展接口预留1路I2C，对应的pin脚和设备节点如下：

pin脚	功能定义	设备节点
19	SPI_MOSI	/dev/spidev0.0
21	SPI_MISO	
23	SPI_SCLK	
24	SPI_CS0	
26	SPI_CS1	/dev/spidev0.1

3.14.5. POWER_KEY

主板扩展接口第37 pin为power_key，对应的设备节点为/dev/input/event0，输入低电平时系统上报键值KEY_POWER，可以通过evtest用具测试：

```

root@rk3576-buildroot:/# evtest
No device specified, trying to scan all of /dev/input/event*
Available devices:
/dev/input/event0:      rk805 pwrkey
/dev/input/event1:      headset-keys
/dev/input/event2:      rockchip-es8388 Headset
/dev/input/event3:      bt-powerkey
/dev/input/event4:      adc-keys
/dev/input/event5:      rockchip-hdmi rockchip-hdmi
Select the device event number [0-5]: 0
Input driver version is 1.0.1
Input device ID: bus 0x19 vendor 0x0 product 0x0 version 0x0
Input device name: "rk805 pwrkey"
Supported events:
  Event type 0 (EV_SYN)
  Event type 1 (EV_KEY)
    Event code 116 (KEY_POWER)
Properties:
Testing ... (interrupt to exit)
Event: time 1756372078.207914, type 1 (EV_KEY), code 116 (KEY_POWER), value 1
Event: time 1756372078.207914, ----- SYN_REPORT -----
Event: time 1756372078.474229, type 1 (EV_KEY), code 116 (KEY_POWER), value 0
Event: time 1756372078.474229, ----- SYN_REPORT -----

```

3.15. GPU

执行glmark2-es2-wayland 开启gpu测试窗口：

```

1 root@rk3576-buildroot:/# glmark2-es2-wayland
2 arm_release_ver: g24p0-00eac0, rk_so_ver: 7
3 =====
4     glmark2 2023.01
5     =====
6     OpenGL Information
7     GL_VENDOR:      ARM
8     GL_RENDERER:    Mali-G52
9     GL_VERSION:     OpenGL ES 3.2 v1.g24p0-00eac0.7624deb1338aa462bb011099
10    003f89a4
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: 1473 FrameTime: 0.679 ms
15 [build] use-vbo=true: FPS: 1399 FrameTime: 0.715 ms
16 [texture] texture-filter=nearest: FPS: 2044 FrameTime: 0.489 ms
17 [texture] texture-filter=linear: FPS: 2110 FrameTime: 0.474 ms
18 [texture] texture-filter=mipmap: FPS: 2101 FrameTime: 0.476 ms
19 [shading] shading=gouraud: FPS: 1191 FrameTime: 0.840 ms
20 [shading] shading=blinn-phong-inf: FPS: 1321 FrameTime: 0.757 ms
21 [shading] shading=phong: FPS: 1248 FrameTime: 0.802 ms
22 [shading] shading=cel: FPS: 1191 FrameTime: 0.840 ms
23 [bump] bump-render=high-poly: FPS: 652 FrameTime: 1.535 ms
24 [bump] bump-render=normals: FPS: 1924 FrameTime: 0.520 ms
25 [bump] bump-render=height: FPS: 1531 FrameTime: 0.653 ms
26 [effect2d] kernel=0,1,0;1,-4,1;0,1,0;: FPS: 1111 FrameTime: 0.901 ms
27 [effect2d] kernel=1,1,1,1,1;1,1,1,1,1;1,1,1,1,1;: FPS: 518 FrameTime: 1.93
28 1 ms
29 [pulsar] light=false:quads=5:texture=false: FPS: 1347 FrameTime: 0.743 ms
30 [desktop] blur-radius=5:effect=blur:passes=1:separable=true:windows=4: FPS
31 : 723 FrameTime: 1.383 ms
32 [desktop] effect=shadow:windows=4: FPS: 1458 FrameTime: 0.686 ms
33 [buffer] columns=200:interleave=false:update-dispersion=0.9:update-fractio
34 n=0.5:update-method=map: FPS: 225 FrameTime: 4.453 ms
35 [buffer] columns=200:interleave=false:update-dispersion=0.9:update-fractio
36 n=0.5:update-method=subdata: FPS: 203 FrameTime: 4.933 ms
37 [buffer] columns=200:interleave=true:update-dispersion=0.9:update-fraction
38 =0.5:update-method=map: FPS: 317 FrameTime: 3.157 ms
39 [ideas] speed=duration: FPS: 782 FrameTime: 1.280 ms
40 [jellyfish] <default>: FPS: 978 FrameTime: 1.023 ms
41 [terrain] <default>: FPS: 94 FrameTime: 10.683 ms
42 [shadow] <default>: FPS: 647 FrameTime: 1.547 ms
43 [refract] <default>: FPS: 181 FrameTime: 5.548 ms
44 [conditionals] fragment-steps=0:vertex-steps=0: FPS: 1276 FrameTime: 0.784
45 ms

```

```

39 [conditionals] fragment-steps=5:vertex-steps=0: FPS: 1156 FrameTime: 0.865
40 ms
41 [conditionals] fragment-steps=0:vertex-steps=5: FPS: 1239 FrameTime: 0.808
42 ms
43 [function] fragment-complexity=low:fragment-steps=5: FPS: 1218 FrameTime:
44 0.821 ms
45 [function] fragment-complexity=medium:fragment-steps=5: FPS: 1153 FrameTim
46 e: 0.867 ms
47 [loop] fragment-loop=false:fragment-steps=5:vertex-steps=5: FPS: 1369 Fram
48 eTime: 0.731 ms
[loop] fragment-steps=5:fragment-uniform=false:vertex-steps=5: FPS: 1344 F
rameTime: 0.744 ms
[loop] fragment-steps=5:fragment-uniform=true:vertex-steps=5: FPS: 1333 Fr
ameTime: 0.751 ms
=====
                                glmark2 Score: 1115
=====

```

3.16. NPU

参考《Purple Pi OH2 YOLOv5开发手册》文档。

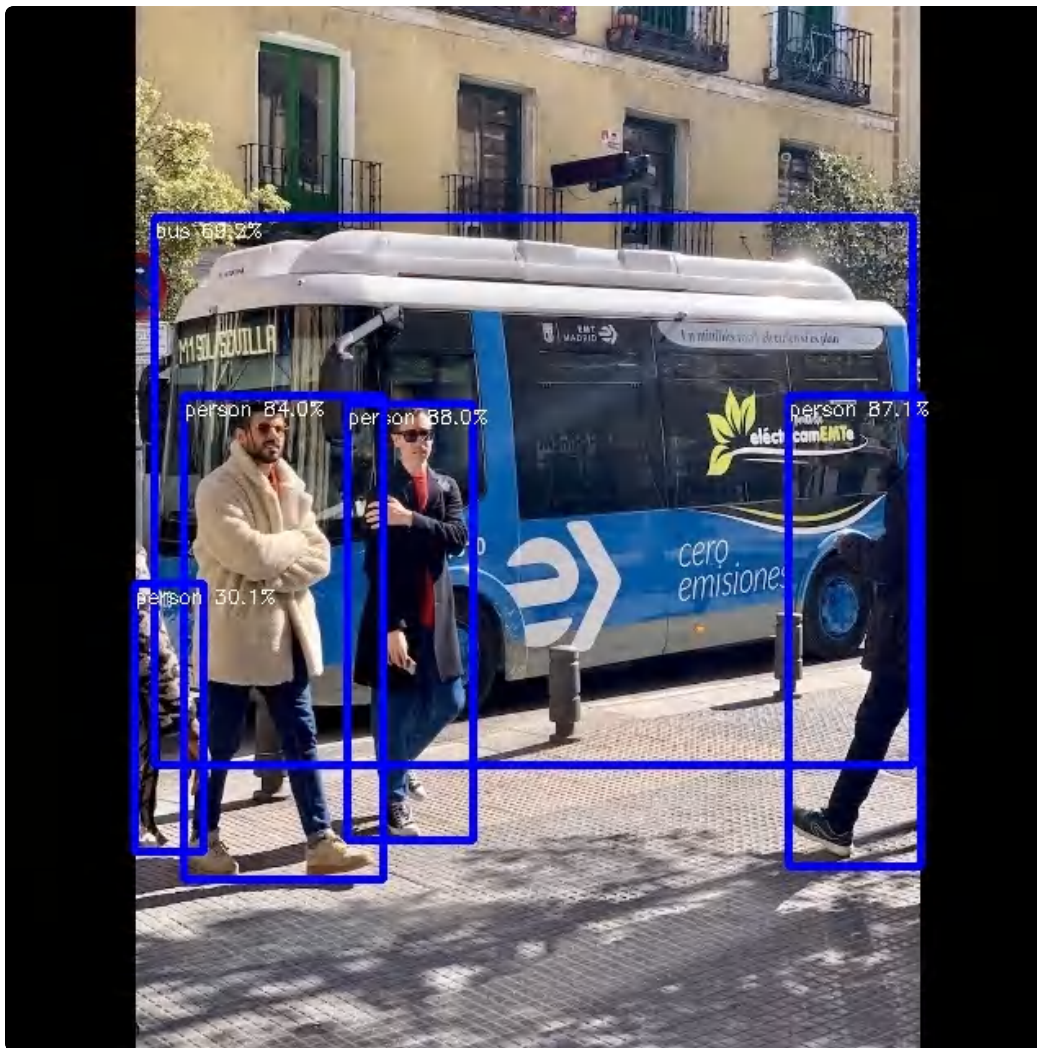
使用yolov5进行模型转换测试：

```

1 root@rk3576-buildroot:/# tar -xvf rknn_yolov5_demo_Linux.tar.gz
2 root@rk3576-buildroot:/# cd rknn_yolov5_demo_Linux
3 root@rk3576-buildroot:/# chmod a+x rknn_yolov5_demo
4 root@rk3576-buildroot:/# ./rknn_yolov5_demo ./model/RK3576/yolov5s-640-64
0.rknn ./model/bus.jpg
5 post process config: box_conf_threshold = 0.25, nms_threshold = 0.45
6 Loading mode...
7 sdk version: 2.3.2 (429f97ae6b@2025-04-09T09:09:27) driver version: 0.9.8
8 model input num: 1, output num: 3
9 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
10 index=0, name=output0, n_dims=4, dims=[1, 255, 80, 80], n_elems=1632000
, size=1632000, w_stride = 0, size_with_stride=1638400, fmt=NCHW, type=INT
8, qnt_type=AFFINE, zp=-128, scale=0.003922
11 index=1, name=286, n_dims=4, dims=[1, 255, 40, 40], n_elems=408000, size
=408000, w_stride = 0, size_with_stride=491520, fmt=NCHW, type=INT8, qnt_t
ype=AFFINE, zp=-128, scale=0.003922
12 index=2, name=288, n_dims=4, dims=[1, 255, 20, 20], n_elems=102000, size
=102000, w_stride = 0, size_with_stride=163840, fmt=NCHW, type=INT8, qnt_t
ype=AFFINE, zp=-128, scale=0.003922
13 model is NHWC input fmt
14 model input height=640, width=640, channel=3
15 Read ./model/bus.jpg ...
16 img width = 640, img height = 640
17 once run use 37.585000 ms
18 loadLabelName ./model/coco_80_labels_list.txt
19 person @ (209 243 286 510) 0.879723
20 person @ (479 238 560 526) 0.870588
21 person @ (109 237 232 534) 0.828112
22 bus @ (93 129 553 464) 0.700761
23 person @ (79 353 122 517) 0.307297
24 save detect result to ./out.jpg
25 loop count = 10 , average run 33.047100 ms

```

执行完后会得到转换后的模型图片，如下所示：



3.17. QT5

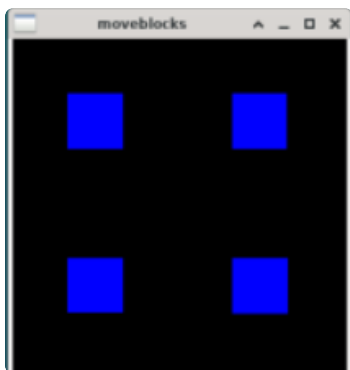
QT应用程序moveblocks编译参考《Purple Pi OH2 QT应用开发手册》

支持qt5.15，测试方法如下：

▼ Shell

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

结果如下所示：



3.18. USB 摄像头

3.18.1. 确认USB摄像头设备节点

```
▼ Shell |
1 root@rk3576-buildroot:/# grep ' ' /sys/class/video4linux/video*/name
2 ...
3 /sys/class/video4linux/video73/name:Webcam gadget: UVC Color Camera
4 /sys/class/video4linux/video74/name:Webcam gadget: UVC Color Camera
5 ...
```

从以上信息得知USB摄像头设备节点为/dev/video73和/dev/video74，一般取较小的一个节点即/dev/video73。

3.18.2. 查看USB摄像头支持的图像格式

```

1 root@rk3576-buildroot:/# v4l2-ctl -d /dev/video73 --list-formats-ext
2 ioctl: VIDIOC_ENUM_FMT
3     Type: Video Capture
4
5     [0]: 'MJPG' (Motion-JPEG, compressed)
6         Size: Discrete 1920x1080
7             Interval: Discrete 0.033s (30.000 fps)
8             Interval: Discrete 0.040s (25.000 fps)
9             Interval: Discrete 0.050s (20.000 fps)
10            Interval: Discrete 0.067s (15.000 fps)
11            Interval: Discrete 0.100s (10.000 fps)
12            Interval: Discrete 0.200s (5.000 fps)
13        Size: Discrete 1280x720
14            Interval: Discrete 0.033s (30.000 fps)
15            Interval: Discrete 0.040s (25.000 fps)
16            Interval: Discrete 0.050s (20.000 fps)
17            Interval: Discrete 0.067s (15.000 fps)
18            Interval: Discrete 0.100s (10.000 fps)
19            Interval: Discrete 0.200s (5.000 fps)
20        ...
21     [1]: 'YUYV' (YUYV 4:2:2)
22         Size: Discrete 640x480
23             Interval: Discrete 0.033s (30.000 fps)
24             Interval: Discrete 0.067s (15.000 fps)
25             Interval: Discrete 0.100s (10.000 fps)
26         Size: Discrete 1280x720
27             Interval: Discrete 0.067s (15.000 fps)
28             Interval: Discrete 0.100s (10.000 fps)
29             Interval: Discrete 0.200s (5.000 fps)
30         Size: Discrete 1920x1080
31             Interval: Discrete 0.067s (15.000 fps)
32             Interval: Discrete 0.100s (10.000 fps)
33             Interval: Discrete 0.200s (5.000 fps)
34     [2]: 'H264' (H.264, compressed)
35         Size: Discrete 1920x1080
36             Interval: Discrete 0.033s (30.000 fps)
37             Interval: Discrete 0.040s (25.000 fps)
38             Interval: Discrete 0.050s (20.000 fps)
39             Interval: Discrete 0.067s (15.000 fps)
40             Interval: Discrete 0.100s (10.000 fps)
41             Interval: Discrete 0.200s (5.000 fps)
42         Size: Discrete 1280x720
43             Interval: Discrete 0.033s (30.000 fps)
44             Interval: Discrete 0.040s (25.000 fps)
45             Interval: Discrete 0.050s (20.000 fps)

```

```
46 Interval: Discrete 0.067s (15.000 fps)
47 Interval: Discrete 0.100s (10.000 fps)
48 Interval: Discrete 0.200s (5.000 fps)
49 ...
```

从以上信息得知USB摄像头输出图像支持以下格式：

1. MJPG 1920x1080 30fps
2. MJPG 1280x720 30fps
3. YUYV 640x480 30fps
4. YUYV 1280x720 15fps
5. H264 1920x1080 30fps
6. H264 1280x720 30fps

3.18.3. 预览USB摄像头图像

使用gststreamer预览USB摄像头图像。

3.18.3.1. MJPG格式

由于USB摄像头输出图像是MJPG格式，需指定使用mppjpegdec解码器：

```
▼ Shell |
1  gst-launch-1.0 v4l2src device=/dev/video73 ! image/jpeg,width=1920,height=1080,framerate=30/1 ! mppjpegdec ! videoconvert ! autovideosink
```

3.18.3.2. YUYV格式

如果USB摄像头输出图像是YUV格式，则不需要指定mppjpegdec解码器：

```
▼ Shell |
1  gst-launch-1.0 v4l2src device=/dev/video73 ! video/x-raw,format=YUY2, width=640, height=480, framerate=30/1 ! videoconvert ! autovideosink sync=false
```

3.18.3.3. H264格式

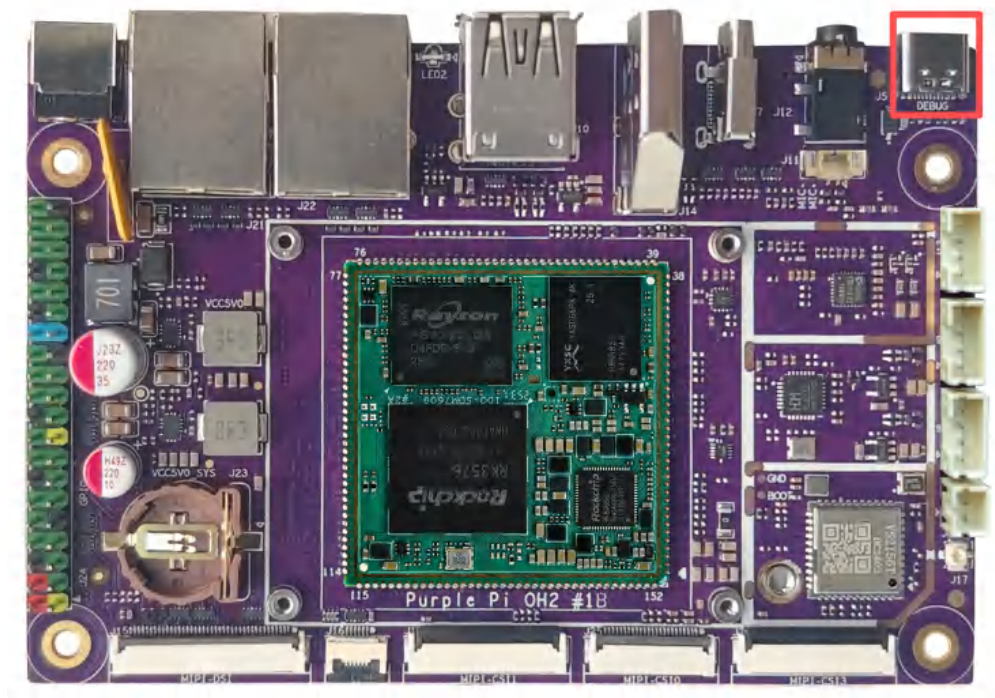
```
1  gst-launch-1.0 v4l2src device=/dev/video73 ! video/x-h264,width=1920,height=1080,framerate=30/1 ! mppvideodec ! videoconvert ! autovideosink
```

Purple Pi OH2 Ubuntu系统

4. 功能及接口使用方法

4.1. DEBUG UART

DEBUG UART位置J5，如下图所示：

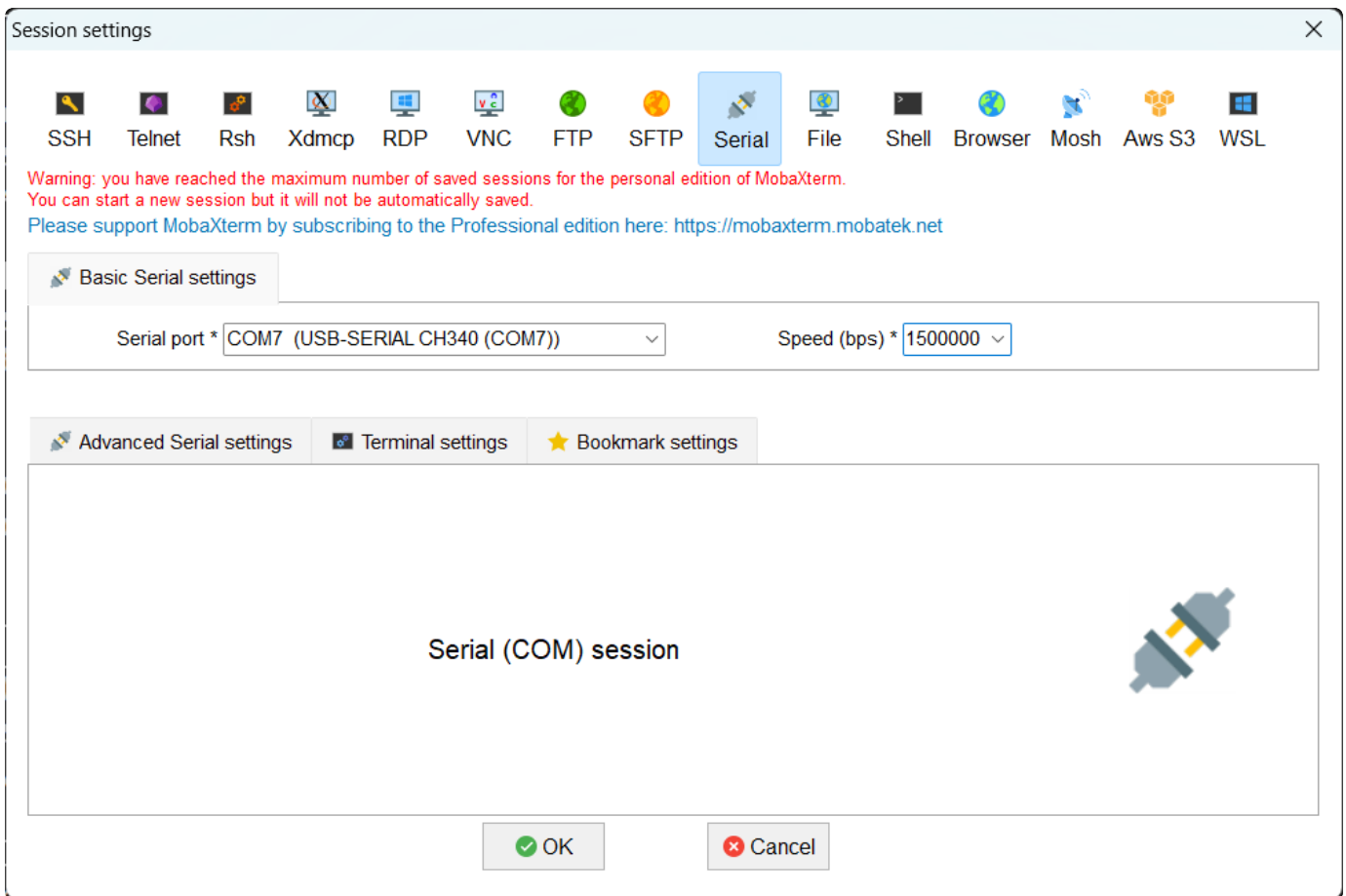


使用USB Type-C数据线，连接PC端的USB接口。系统会识别到一个“USB-SERIAL CH340”端口设备。



串口登录账号密码：industio/123456

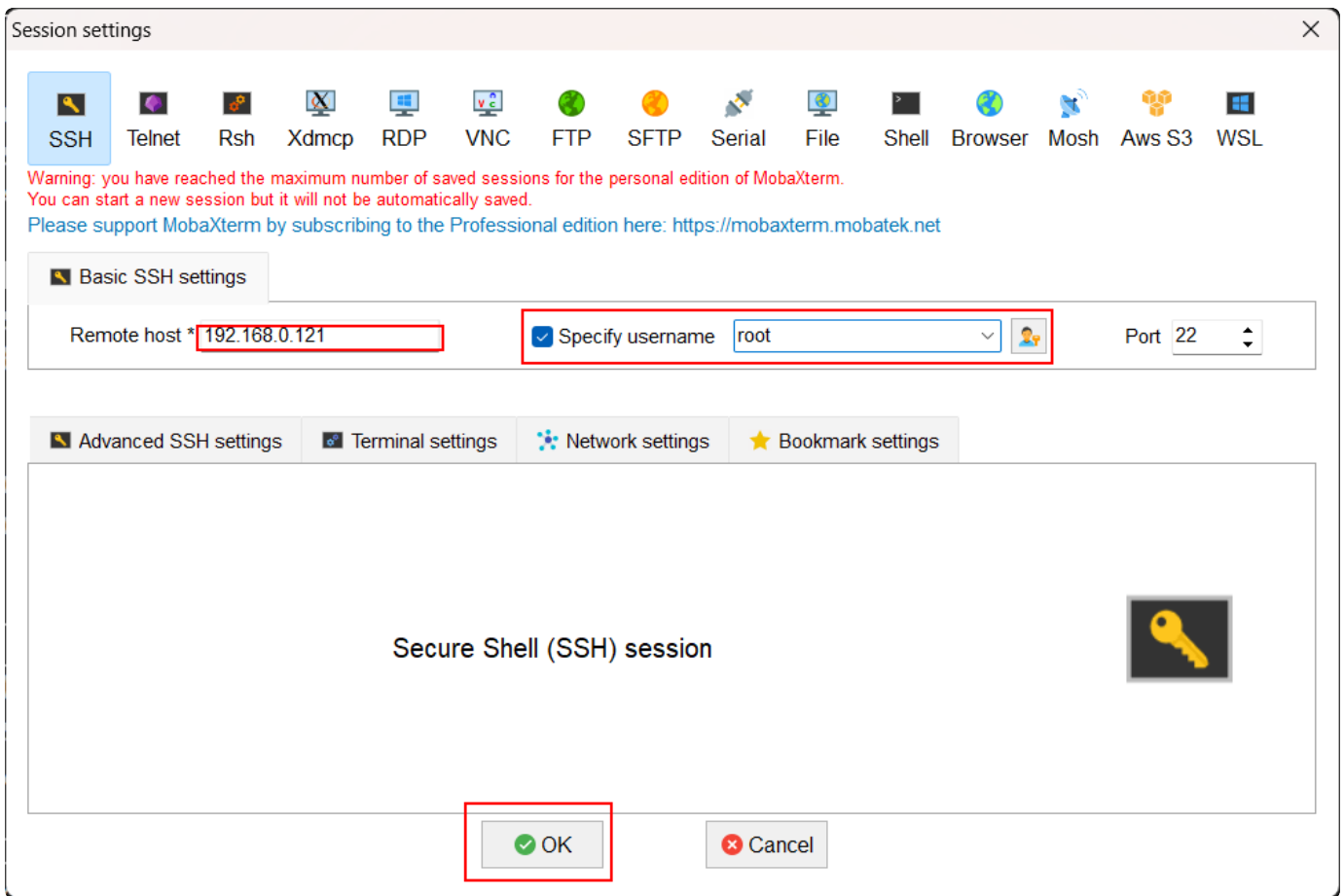
使用调试软件（MobaXterm、putty）等，以MobaXterm为例子，设置参数如下：



SSH登录

在知道IP的前提下，默认可以通过SSH登录进系统。

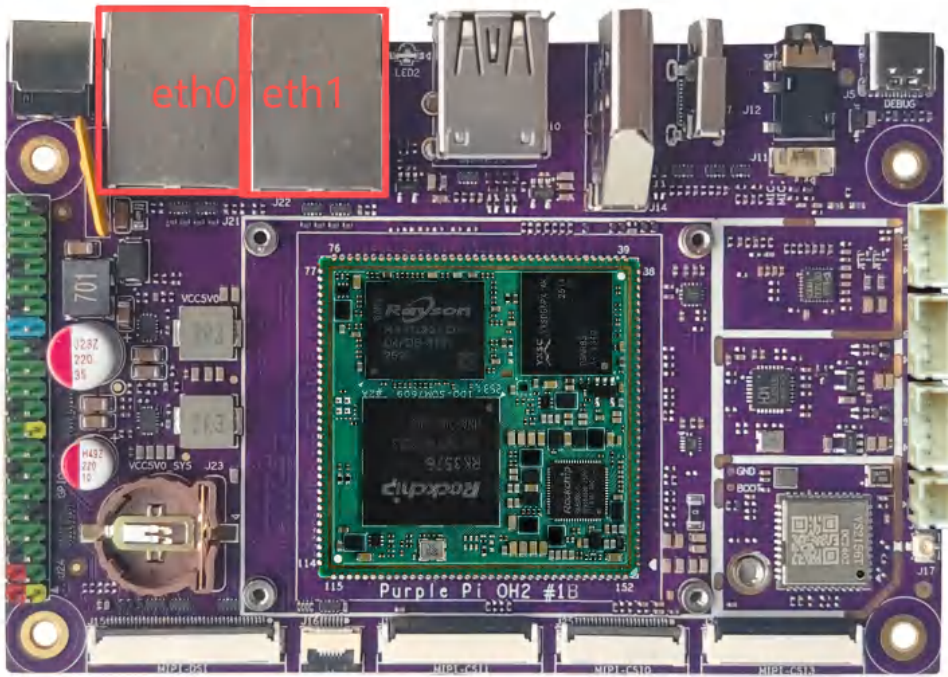
SSH登录账号密码：industio/123456



4.2. 网络

4.2.1. 以太网

主板有两路千兆以太网接口位置J21和J22，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。

Ethernet动态IP设置:

```
1 #eth0
2 industio@industio:~$ ifconfig eth0
3 eth0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
4     ether 2a:6b:1b:4f:e7:bf txqueuelen 1000 (Ethernet)
5     RX packets 0 bytes 0 (0.0 B)
6     RX errors 0 dropped 0 overruns 0 frame 0
7     TX packets 0 bytes 0 (0.0 B)
8     TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
9     device interrupt 67
10
11
12 #eth1
13 industio@industio:~$ ifconfig eth1
14 eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
15     inet 192.168.1.116 netmask 255.255.255.0 broadcast 192.168.1.255
16     inet6 fe80::4281:69a2:6758:f125 prefixlen 64 scopeid 0x20<link>
17     ether 2e:6b:1b:4f:e7:bf txqueuelen 1000 (Ethernet)
18     RX packets 643 bytes 117640 (117.6 KB)
19     RX errors 0 dropped 0 overruns 0 frame 0
20     TX packets 419 bytes 49002 (49.0 KB)
21     TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
22     device interrupt 69
```

Ethernet静态IP设置:

```

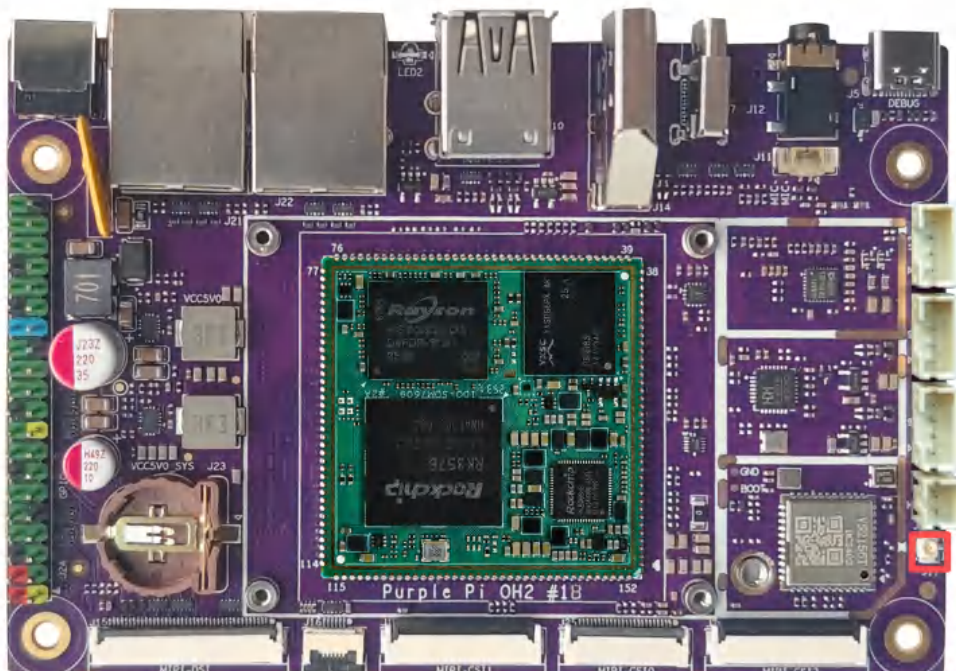
1 root@industio:~# touch /etc/netplan/01-netcfg.yaml
2 root@industio:~# vim /etc/netplan/01-netcfg.yaml
3 root@industio:~# cat /etc/netplan/01-netcfg.yaml
4 network:
5     version: 2
6     ethernet:
7         eth0:
8             dhcp4: no
9             addresses: [192.168.3.121/24]
10            nameservers:
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@industio:~# netplan apply
19 Cannot call openvswitch: ovsdb-server.service is not running.
20
21 industio@industio:~$ ifconfig eth0
22 eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
23     inet 192.168.3.121 netmask 255.255.255.0 broadcast 192.168.3.255
24     inet6 fe80::1b83:a156:cb16:c6b0 prefixlen 64 scopeid 0x20<link>
25     inet6 fe80::f8c6:baff:fec7:c60 prefixlen 64 scopeid 0x20<link>
26     ether 2a:6b:1b:4f:e7:bf txqueuelen 1000 (Ethernet)
27     RX packets 4203 bytes 417790 (407.9 KiB)
28     RX errors 0 dropped 164 overruns 0 frame 0
29     TX packets 392 bytes 32299 (31.5 KiB)
30     TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
31     device interrupt 121

```

设备断电重启，此静态IP设置仍然生效。

4.2.2. WiFi

使用WiFi/蓝牙时，需要连接天线以获得良好的信号，WiFi模块和天线座子J17，如下图所示：



连接WiFi热点

命令行可以使用nmcli工具连接wifi热点：

Shell |

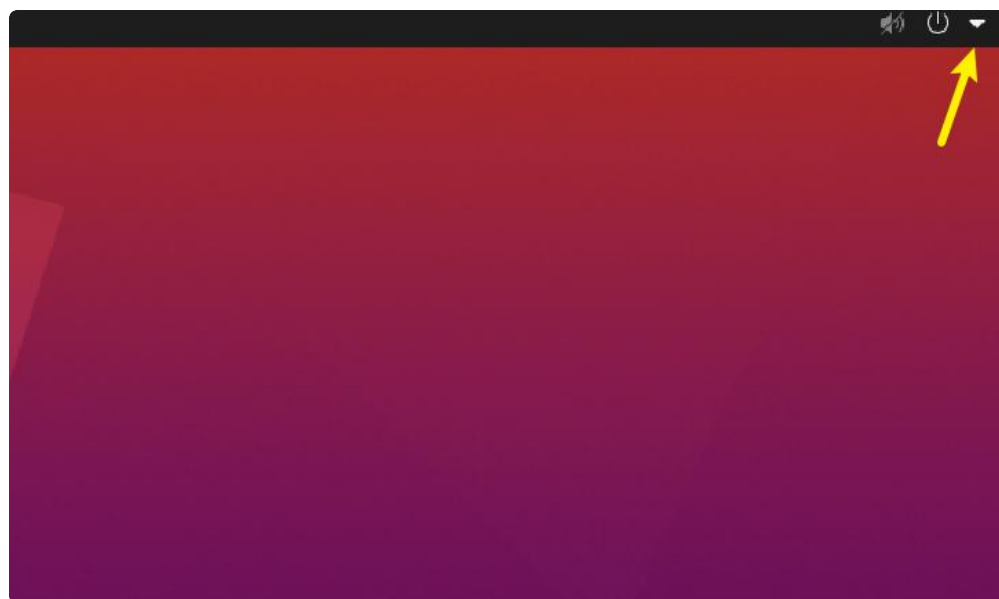
```
1 #查看网络接口状态：
2 root@industio:~# nmcli device
3 #使用以下命令查看当前可用的 WiFi 网络：
4 root@industio:~# nmcli device wifi list
5 #连接到 WiFi 网络：
6 root@industio:~# nmcli device wifi connect 账号 password 密码
7 #确认连接状态：
8 root@industio:~# nmcli connection show --active
```

查看wlan0的IP地址，确认连接成功：

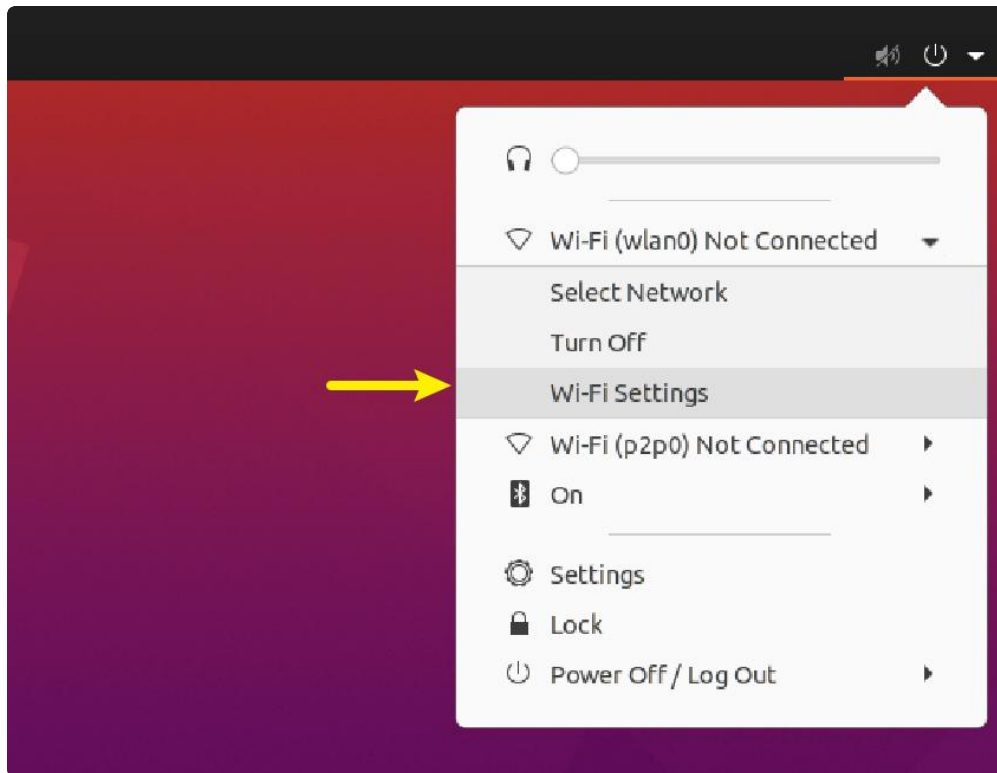
```
▼ Shell |
1 root@industio:~# ifconfig wlan0
2 wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
3     inet 192.168.1.120 netmask 255.255.255.0 broadcast 192.168.1.255
4     inet6 fe80::e6ce:be28:4c94:577e prefixlen 64 scopeid 0x20<link>
5     ether c0:f5:35:12:e6:34 txqueuelen 1000 (Ethernet)
6     RX packets 18 bytes 2238 (2.2 KB)
7     RX errors 0 dropped 0 overruns 0 frame 0
8     TX packets 52 bytes 7844 (7.8 KB)
9     TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
10
11 root@industio:~# ping www.baidu.com
12 PING www.a.shifen.com (183.2.172.17): 56 data bytes
13 64 bytes from 183.2.172.17: icmp_seq=0 ttl=52 time=8.007 ms
14 64 bytes from 183.2.172.17: icmp_seq=1 ttl=52 time=8.049 ms
15 64 bytes from 183.2.172.17: icmp_seq=2 ttl=52 time=7.901 ms
```

也可以在桌面上连接：

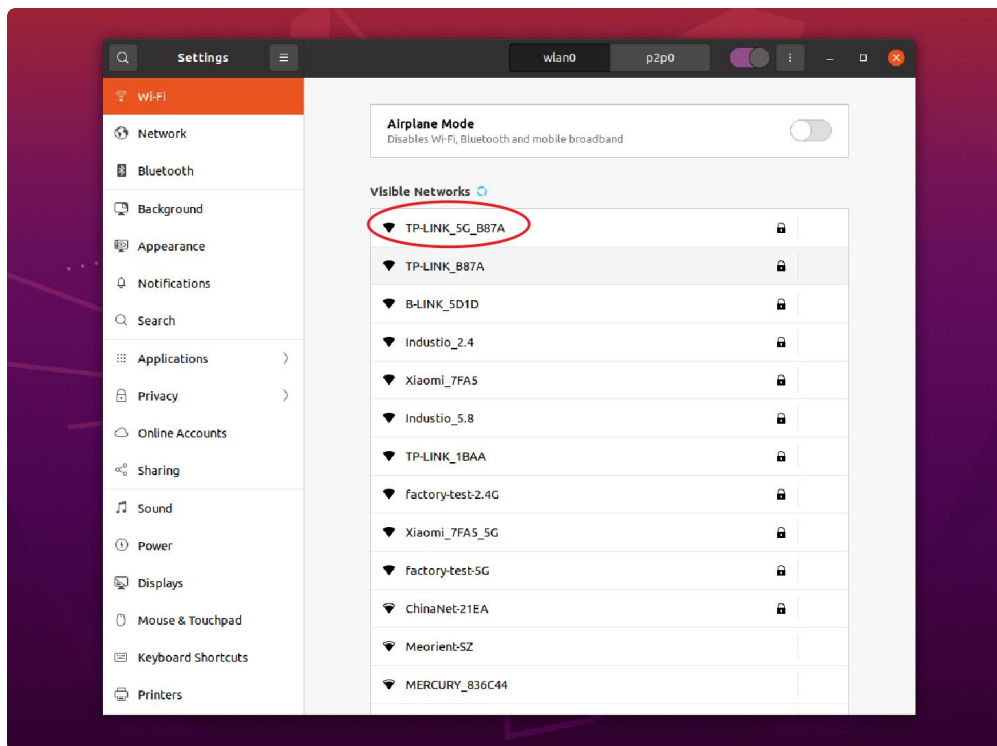
点击桌面右上角的【下拉】选项按钮，如下图所示：



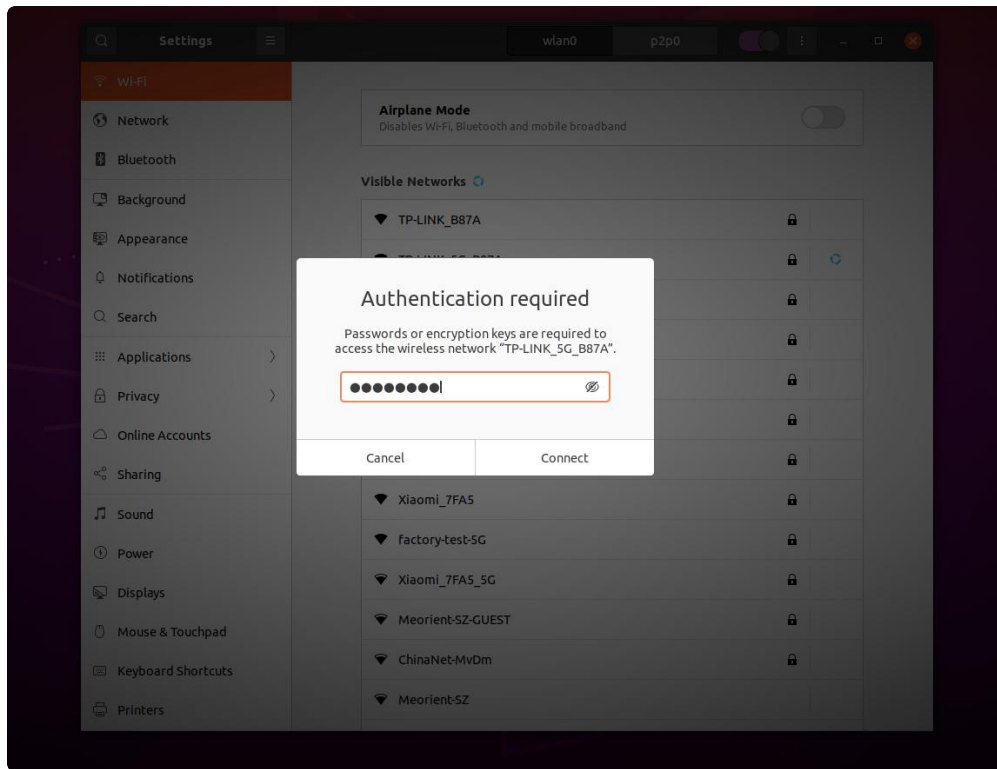
弹出的列表中点击【Wi-Fi Settings】，如下图所示：



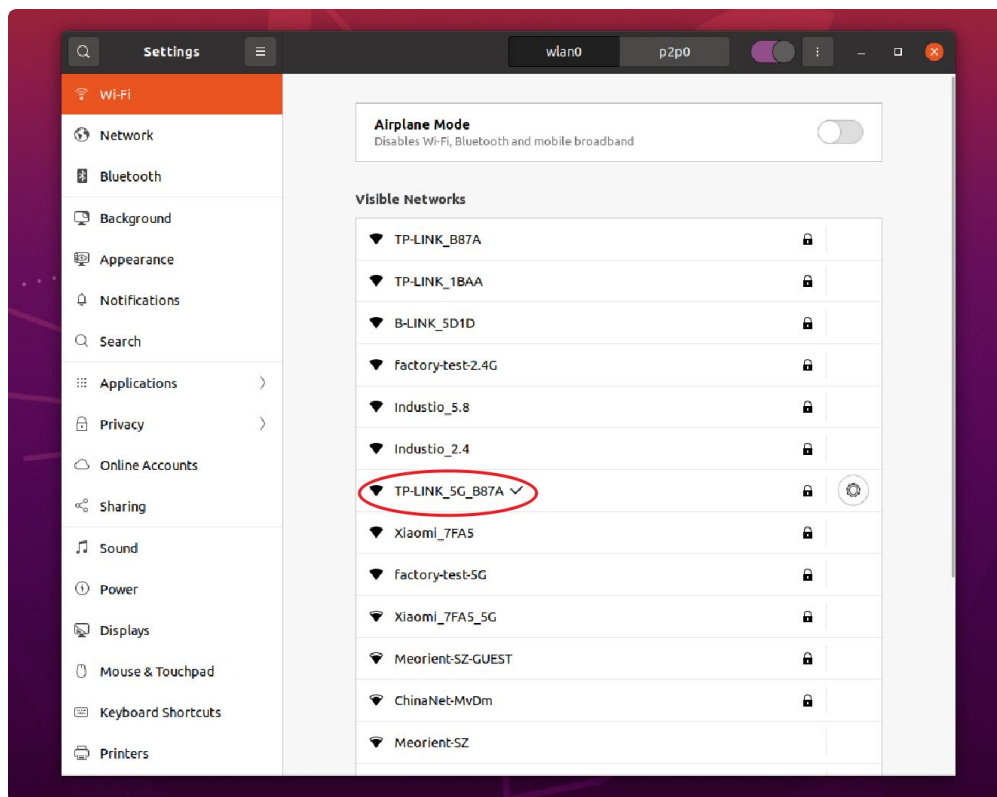
弹出WiFi热点列表，点击要连接的热点名称，如下图所示：



弹出密码输入框，使用键盘输入密码，如下图所示：



如果热点名称后面有"√"标记，表示连接成功，如下图所示：



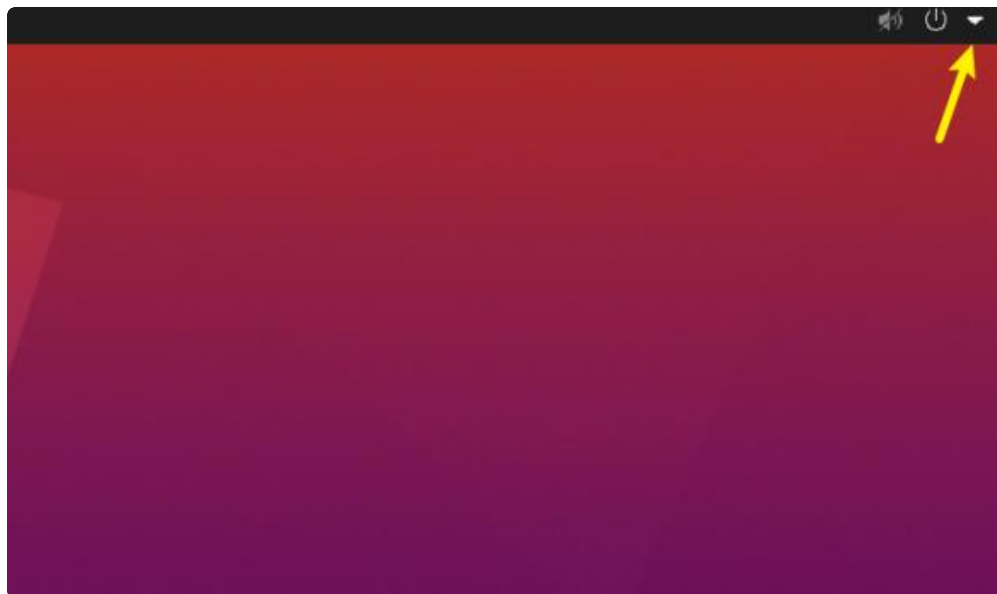
通过ifconfig 命令查看wlan0的IP地址确认，命令如下：

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

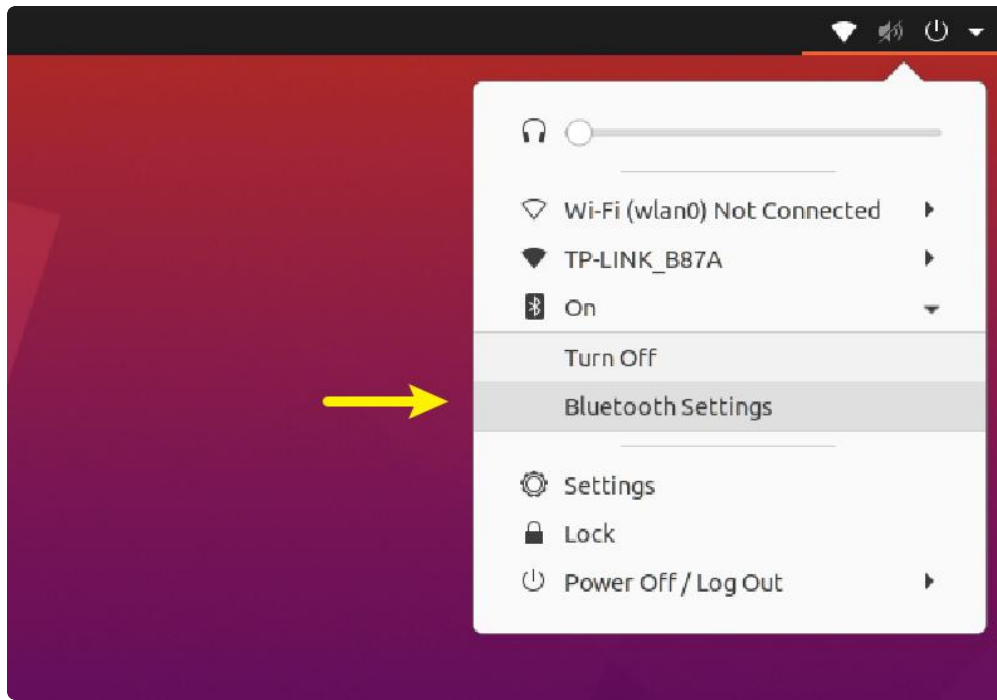
4.2.3. Bluetooth

桌面连接：

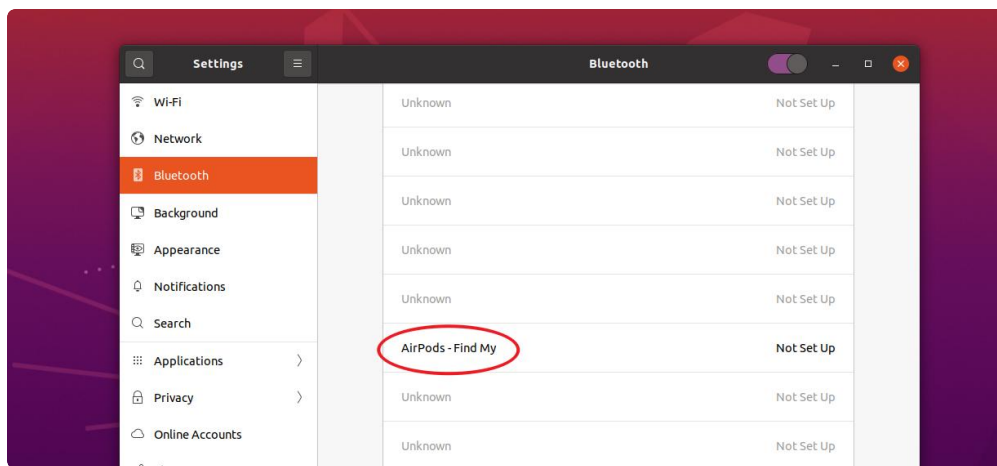
点击桌面右上角的【下拉选项】，如下图所示：



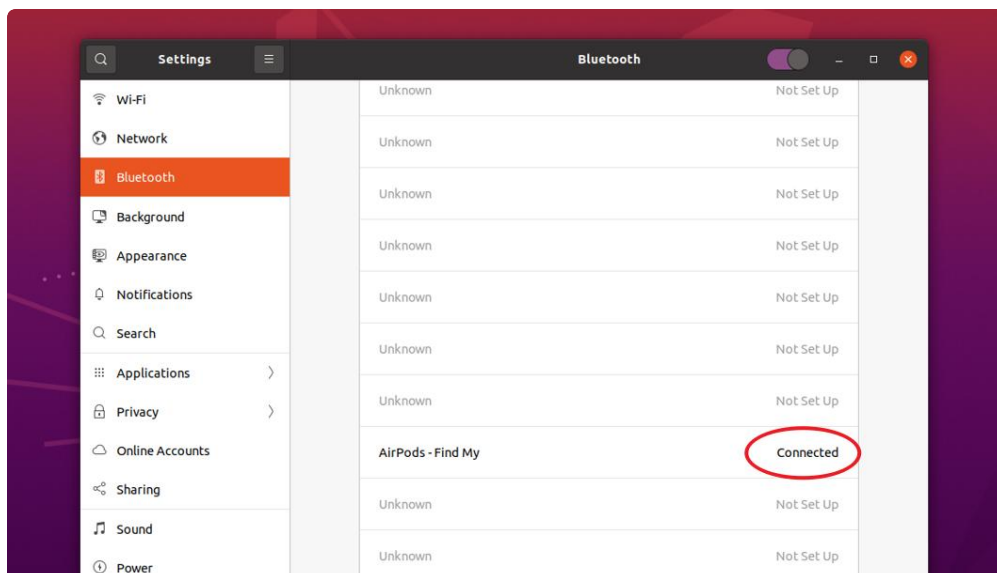
弹出的列表中点击【蓝牙】，继续点击【Bluetooth Setting】，如下图所示：



弹出蓝牙扫描列表，点击要连接的蓝牙设备名称，连接蓝牙设备，如下图所示：



设备名称后面提示"Connected", 表示该设备已连接成功，如下图所示：



使用命令行操作

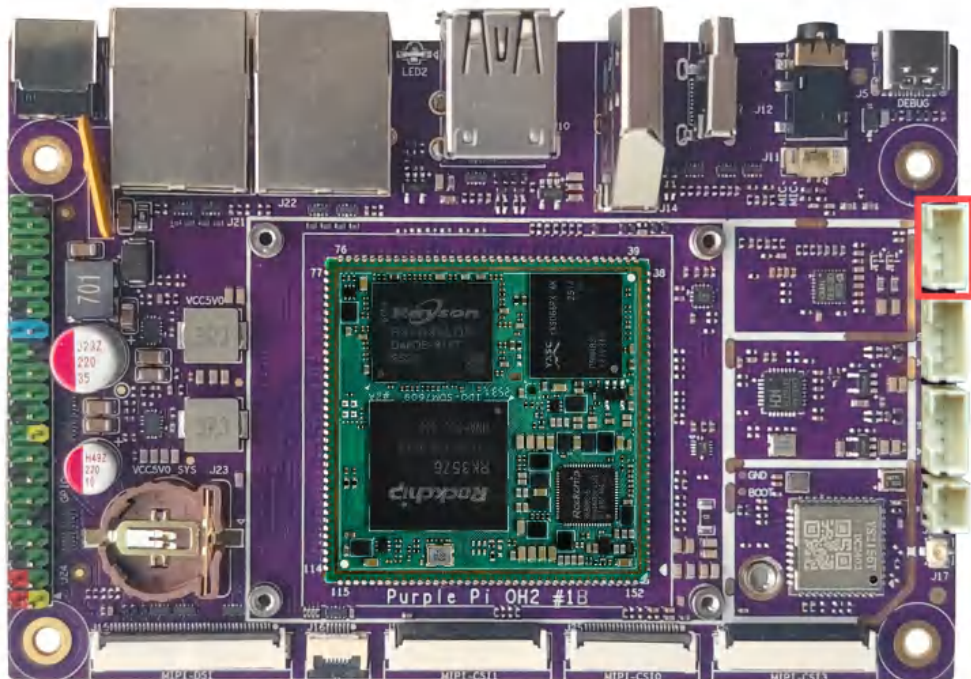
Shell

```
1  #打开蓝牙
2  root@industio:~# bluetoothctl power on
3  #扫描蓝牙
4  root@industio:~# bluetoothctl scan on
5  #查看蓝牙设备
6  root@industio:~# bluetoothctl devices
7  #信任蓝牙设备
8  root@industio:~# bluetoothctl trust 7C:C1:80:09:DD:6C
9  #蓝牙配对
10 root@industio:~# bluetoothctl pair 7C:C1:80:09:DD:6C
11 #连接蓝牙
12 root@industio:~# bluetoothctl connect 7C:C1:80:09:DD:6C
```

4.3. 声音

4.3.1. 喇叭

喇叭接口为PH2.0-4P连接器J13，如下图所示：

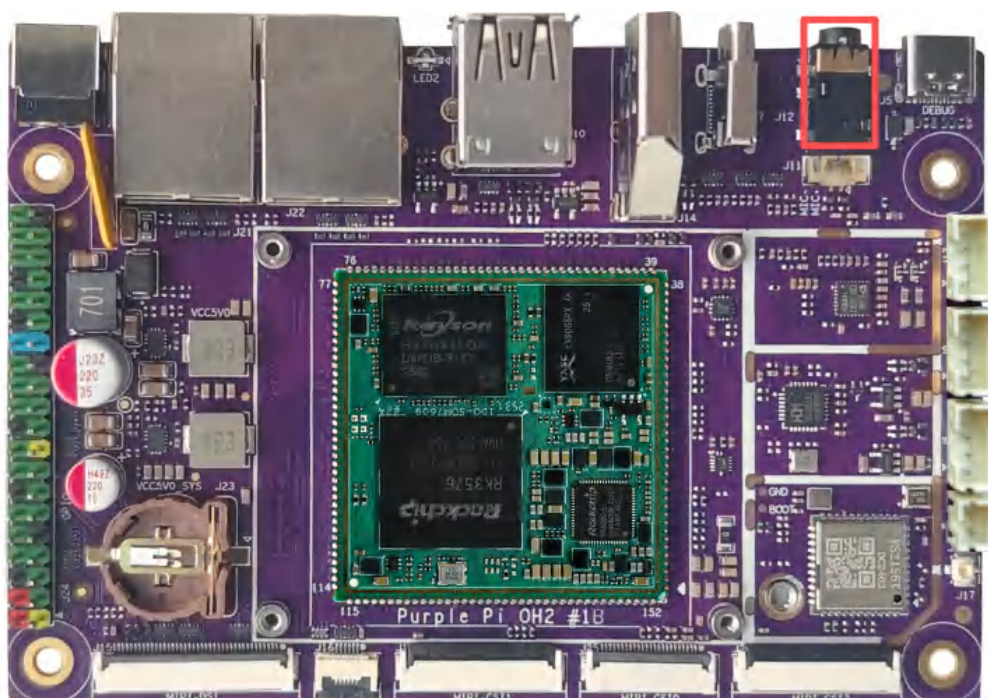


- 支持双声道，每个声道支持4Ω 3W输出

```
1 #查看声卡
2 root@industio:~# aplay -l
3 **** List of PLAYBACK Hardware Devices ****
4 card 1: rockchipes8388 [rockchip-es8388], device 0: dailink-multicodecs ES
   8323 HiFi-0 [dailink-multicodecs ES8323 HiFi-0]
5   Subdevices: 1/1
6   Subdevice #0: subdevice #0
7 card 2: rockchiphdmi [rockchip-hdmi], device 0: rockchip-hdmi i2s-hifi-0
   [rockchip-hdmi i2s-hifi-0]
8   Subdevices: 1/1
9   Subdevice #0: subdevice #0
10
11 #音频播放到声卡
12 root@industio:~# aplay -D hw:0,0 xihuangni.wav
13 Playing WAVE 'xihuangni.wav' : Signed 16 bit Little Endian, Rate 44100 H
   z, Stereo
14 #音频播放到HDMI
15 root@industio:~# aplay -D hw:1,0 xihuangni.wav
16 Playing WAVE 'xihuangni.wav' : Signed 16 bit Little Endian, Rate 44100 H
   z, Stereo
```

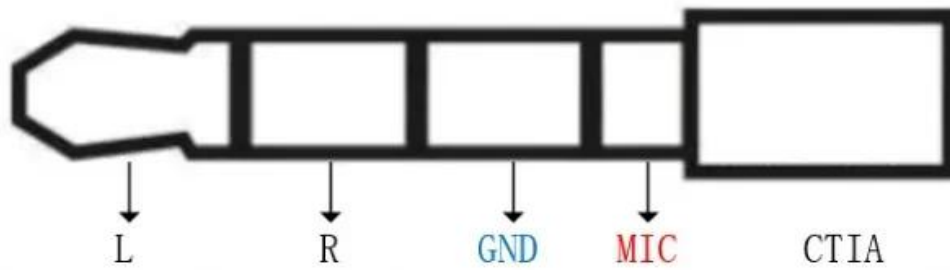
4.3.2. 耳机

主板支持1路3.5mm四节耳机座（CTIA）J12，和1路MX1.25T-5P耳机座子并用，用户可根据需求选用其中1路耳机座子使用，如下图所示：



- 支持耳机检测
- 支持耳机录音

CTIA标准四段式耳机，定义如下：



耳机录音：

```

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

```

播放音频到耳机：

```

1 root@industio:~# amixer -c 0 cset name='Speaker Switch' on
2 root@industio:~# aplay -D hw:0,0 xihuangni.wav

```

4.3.3. Mic

MX1.25T-2P麦克风接口J11，如下图所示：


```
1  #查看录音设备
2  root@industio:~# arecord -l
3  **** List of CAPTURE Hardware Devices ****
4  card 0: rockchipes8388 [rockchip-es8388], device 0: dailink-multicodecs ES
   8323 HiFi-0 [dailink-multicodecs ES8323 HiFi-0]
5      Subdevices: 1/1
6      Subdevice #0: subdevice #0
7  card 1: rockchiphdmi [rockchip-hdmi], device 0: rockchip-hdmi i2s-hifi-0
   [rockchip-hdmi i2s-hifi-0]
8      Subdevices: 1/1
9      Subdevice #0: subdevice #0
10
11
12  #捕获音量设置(不要设置最大值):
13  root@industio:~# amixer -c 0 cset numid=50 100
14  numid=50,iface=MIXER,name='Capture Digital Volume'
15      ; type=INTEGER,access=rw---R--,values=2,min=0,max=192,step=0
16      : values=100,100
17      | dBscale-min=-96.00dB,step=0.50dB,mute=1
18
19  root@industio:~# amixer -c 0 cget numid=52
20  numid=52,iface=MIXER,name='Left Channel Capture Volume'
21      ; type=INTEGER,access=rw---R--,values=1,min=0,max=8,step=0
22      : values=0
23      | dBscale-min=0.00dB,step=3.00dB,mute=0
24
25  root@industio:~# amixer -c 0 cget numid=53
26  numid=53,iface=MIXER,name='Right Channel Capture Volume'
27      ; type=INTEGER,access=rw---R--,values=1,min=0,max=8,step=0
28      : values=0
29      | dBscale-min=0.00dB,step=3.00dB,mute=0
30
31  自动增益控制(打开):
32  root@industio:~# amixer -c 0 cset numid=41 3
33  numid=41,iface=MIXER,name='ALC Capture Function'
34      ; type=ENUMERATED,access=rw-----,values=1,items=4
35      ; Item #0 'Off'
36      ; Item #1 'Right'
37      ; Item #2 'Left'
38      ; Item #3 'Stereo'
39      : values=3
40
41  调整目标音量:
42  root@industio:~# amixer -c 0 cset numid=38 8
43  numid=38,iface=MIXER,name='ALC Capture Target Volume'
```

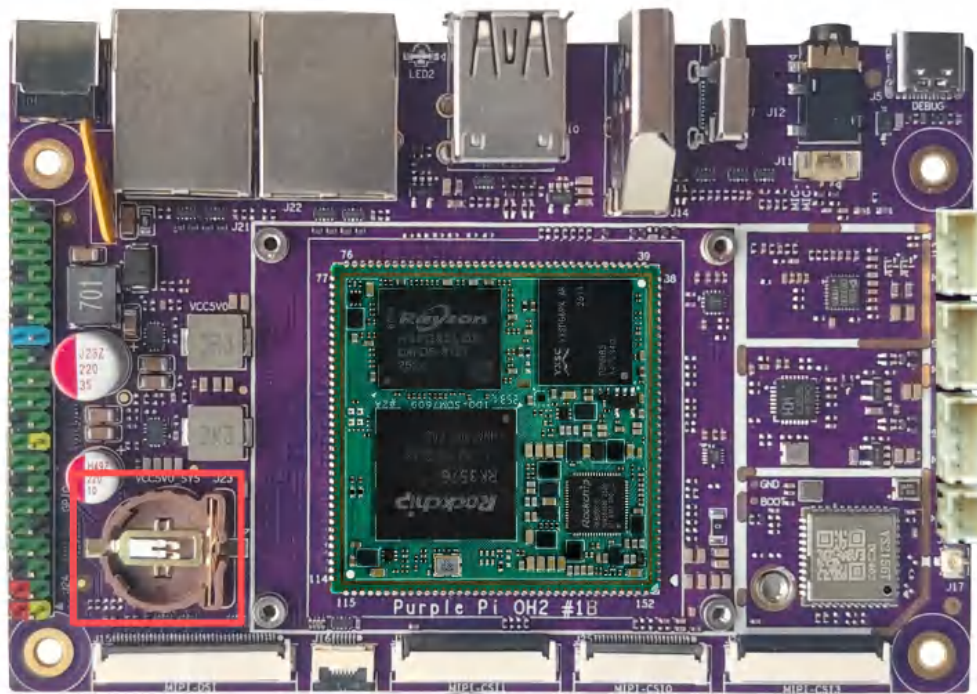
```

44     ; type=INTEGER,access=rw-----,values=1,min=0,max=15,step=0
45     : values=8
46 设置噪声门限值:
47  root@industio:~# amixer -c 0 cset numid=46 15
48  numid=46,iface=MIXER,name='ALC Capture NG Threshold'
49     ; type=INTEGER,access=rw-----,values=1,min=0,max=31,step=0
50     : values=15
51 确保捕获静音未开启:
52  root@industio:~# amixer -c 0 cset numid=51 off
53  numid=51,iface=MIXER,name='Capture Mute'
54     ; type=BOOLEAN,access=rw-----,values=1
55     : values=off
56
57  #主mic开关 (mic录音需要打开, 耳机录音需要关闭) :
58  root@industio:~# amixer -c 0 cset numid=67 on
59  numid=67,iface=MIXER,name='Main Mic Switch'
60     ; type=BOOLEAN,access=rw-----,values=1
61     : values=on
62  root@industio:~# echo -n "mic2" > /sys/class/es8388_mic/mic_channel
63  root@industio:~# arecord -D hw:0,0 -f S16_LE -r 16000 -c 2 test_mic2.wav
64
65  #耳机录音:
66  root@industio:~# amixer -c 0 cset numid=67 off
67  root@industio:~# echo -n "mic1" > /sys/class/es8388_mic/mic_channel
68  root@industio:~# arecord -D hw:0,0 -f S16_LE -r 16000 -c 2 test_mic1.wav

```

4.4. RTC

MX1.25T-2P RTC电池座J23, 如下图所示:



连接3V 纽扣电池。

rtc时间设置方法：

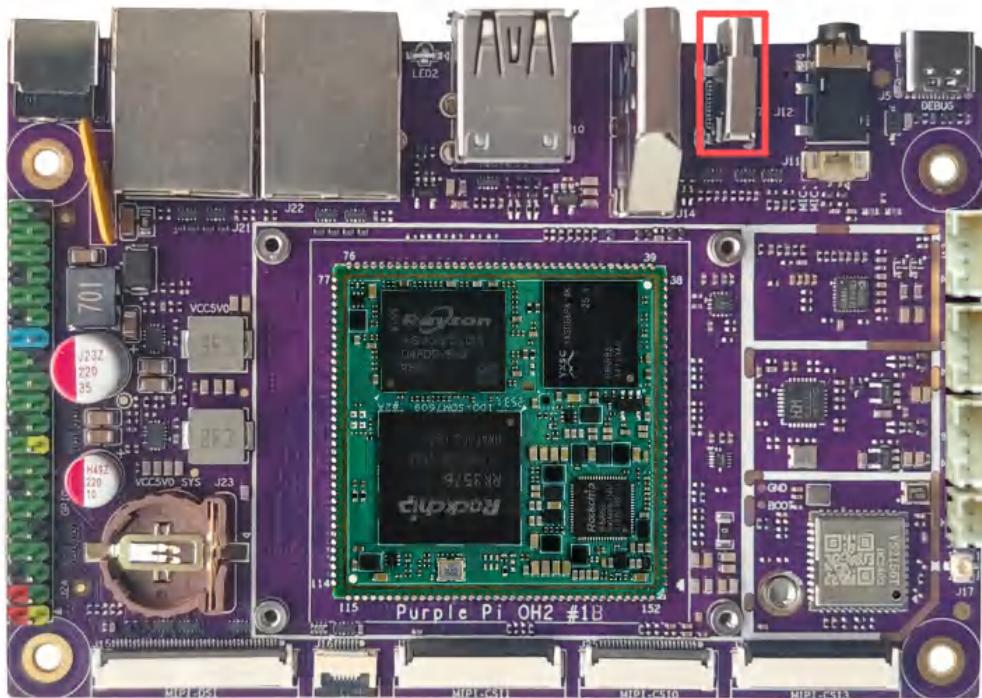
```
1  #设置时间
2  root@industio:~# date -s "2024-10-09 14:02:30"
3
4  #将rtc时钟调整为与目前的系统时钟一致
5  root@industio:~# hwclock -w
6
7  #获取硬件rtc当前时间（断电重启读取时间没有太大偏差）
8  root@industio:~# hwclock -r
9  2024-10-09 14:02:35.945604+00:00
```

4.5. USB接口

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

4.5.1. TypeC接口

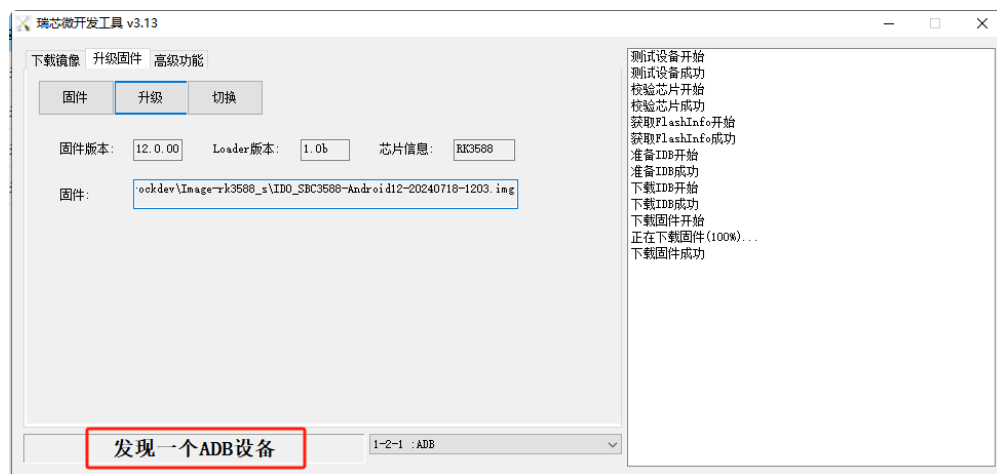
TypeC接口（USB3.2 Gen1 OTG+DP1.4输出）J7，如下图所示：



- 支持Host、Device模式自动切换
- 支持DP显示输出

Device从机模式

使用TypeC数据线连接电脑，烧录工具能发现一个ADB设备，如下图所示：



- 可使用ADB相关开发工具对盒子进行功能调试

Host主机模式

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



- 可识别U盘、键盘、鼠标并正常使用

DP模式

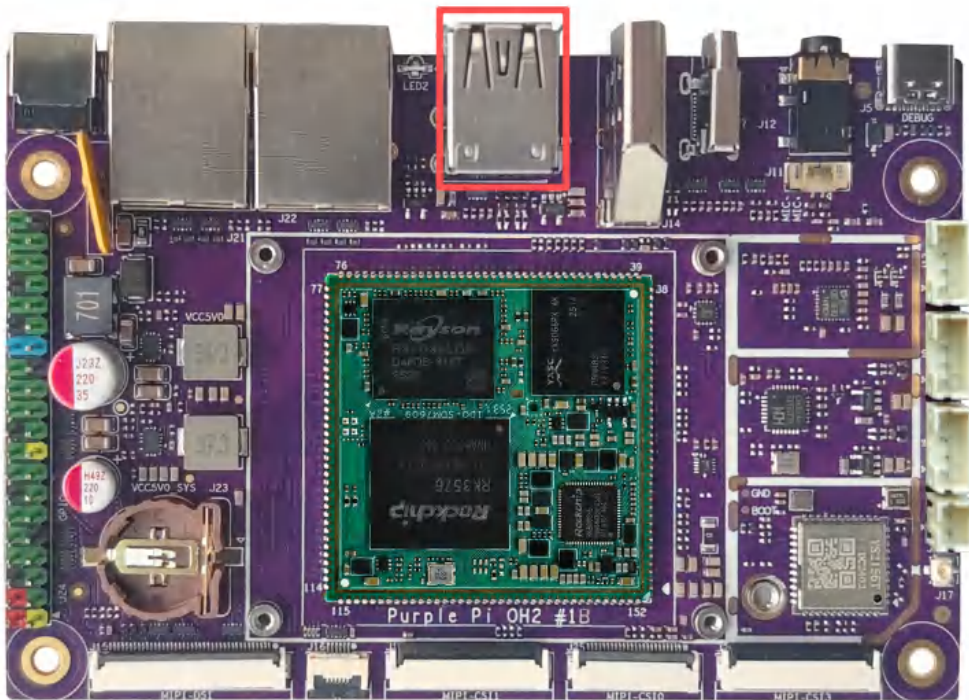
通过TypeC全功能数据线接入DP显示器，或通过TYPE-C to HDMI数据线连接HDMI显示器



- 主板画面通过TYPE-C DP输出正常，DP声音输出正常

4.5.2. USB3.0接口

主板支持1个USB3.0 TYPE-A 接口，位于J10（下），如下图所示：



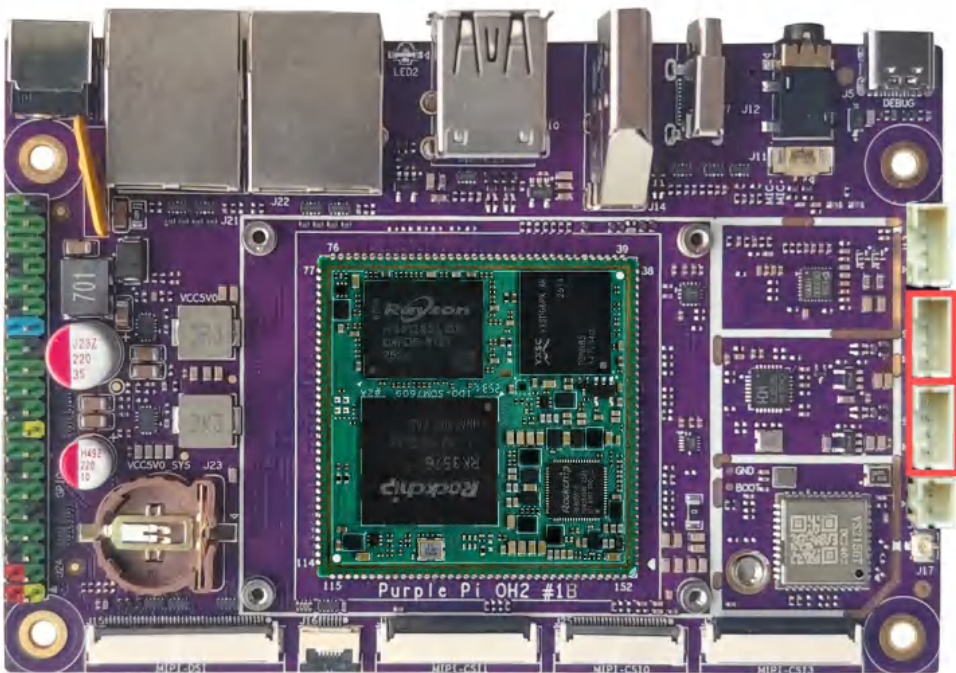
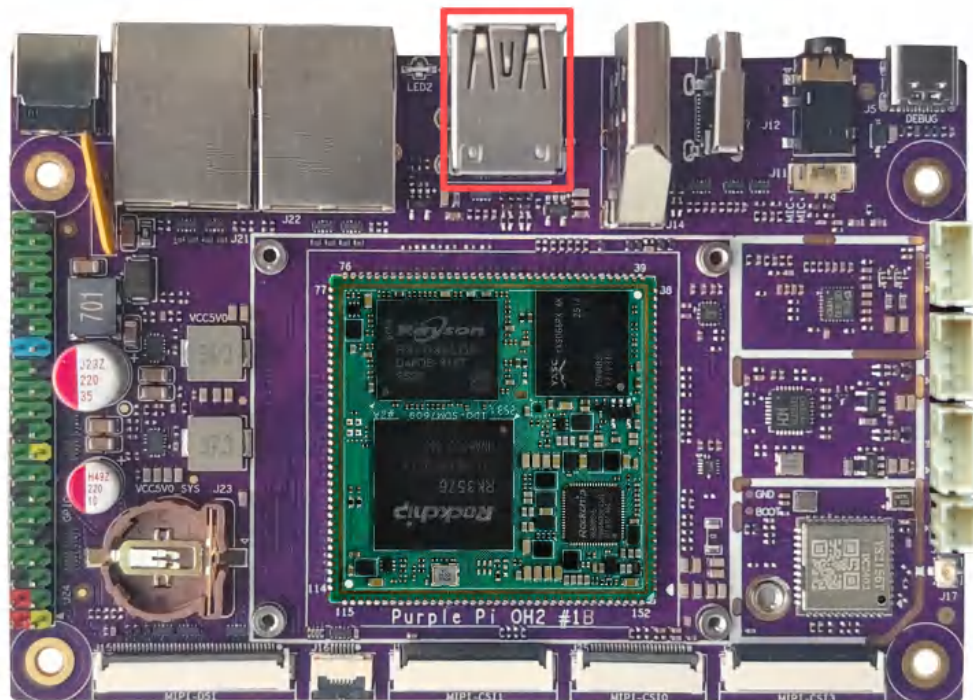
USB母座提供5V@1A供电能力，2个USB端口供电控制是共用的，USB电源控制，如下表所示：

USB端口	默认状态	动作	命令
USB1/2	开启	关闭电源	echo 0 > /sys/class/leds/usb_pwr/brightness
		开启电源	echo 1 > /sys/class/leds/usb_pwr/brightness

供电控制说明：设备节点写"0"关闭电源，写"1"开启电源

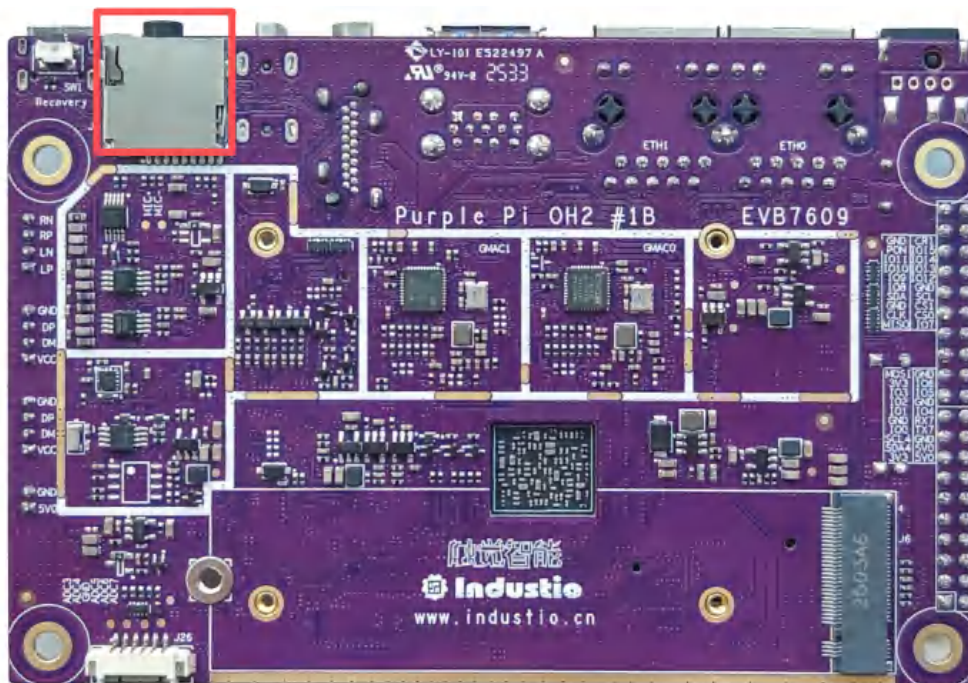
4.5.3. USB2.0接口

主板配置了1路USB2.0 TYPE-A接口，位于J10（上），及2路PH2.0-4P接口，位于J8和J9，USB接口均提供5V@1A的驱动能力，如下图所示：



4.6. TF Card

TF卡座支持SDIO3.0，支持高速SD卡，接口位于主板背面J4，如下图所示：



TF卡安装方向，主板正面朝上时TF卡触点朝上。

Shell

```

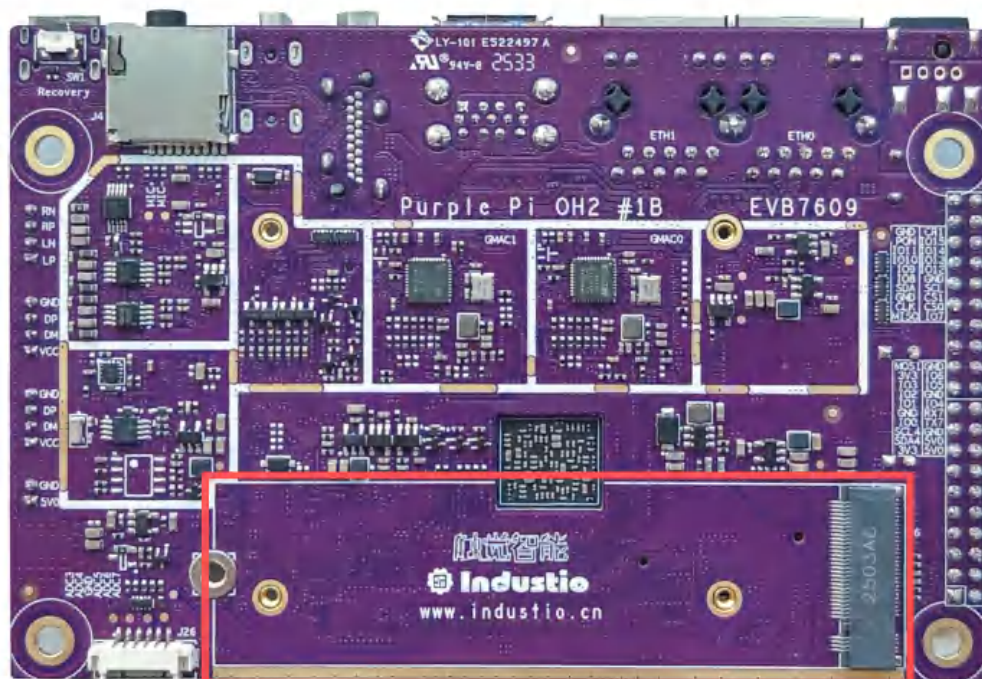
1  root@industio:~# fdisk -l
2  Found valid GPT with protective MBR; using GPT
3
4  Disk /dev/mmcblk2: 61071360 sectors, 1148M
5  Logical sector size: 512
6  Disk identifier (GUID): 21130000-0000-4444-8000-05e1000030fd
7  Partition table holds up to 128 entries
8  First usable sector is 34, last usable sector is 61071326
9
10 Number  Start (sector)    End (sector)  Size Name
11     1           16384           24575  4096K uboot
12     2           24576           32767  4096K misc
13     3           32768          163839  64.0M boot
14     4          163840          425983  128M recovery
15     5          425984          491519  32.0M backup
16     6          491520          8880127  4096M userdata
17     7          8880128          9142271  128M oem
18     8          9142272          61071295  24.7G rootfs
19  ...
20 Device      Boot StartCHS      EndCHS          StartLBA      EndLBA      Sector
   s  Size Id Type
21 /dev/mmcblk1p1  0,2,10      239,254,63      135      3858623      3858489
    1884M  e Win95 FAT16 (LBA)

```

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

4.7. M.2接口

主板上使用标准M.2-M-key M.2接口连接座，如下图所示：



支持PCIe2.1通信，适用2280尺寸NVME固态硬盘。

```

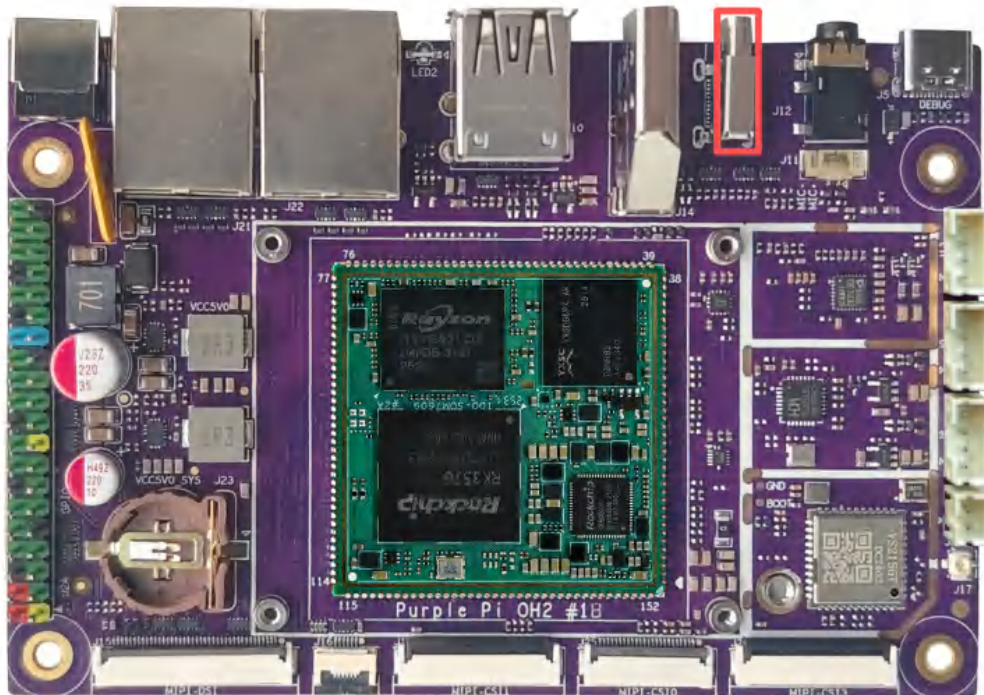
1  root@industio:~# fdisk -l
2  Disk /dev/ram0: 8 MiB, 8388608 bytes, 16384 sectors
3  Units: sectors of 1 * 512 = 512 bytes
4  Sector size (logical/physical): 512 bytes / 4096 bytes
5  I/O size (minimum/optimal): 4096 bytes / 4096 bytes
6
7
8  Disk /dev/ram1: 8 MiB, 8388608 bytes, 16384 sectors
9  Units: 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)

```

4.8. LCD显示

4.8.1. DP

DP接口（USB-C）J7，如下图所示：



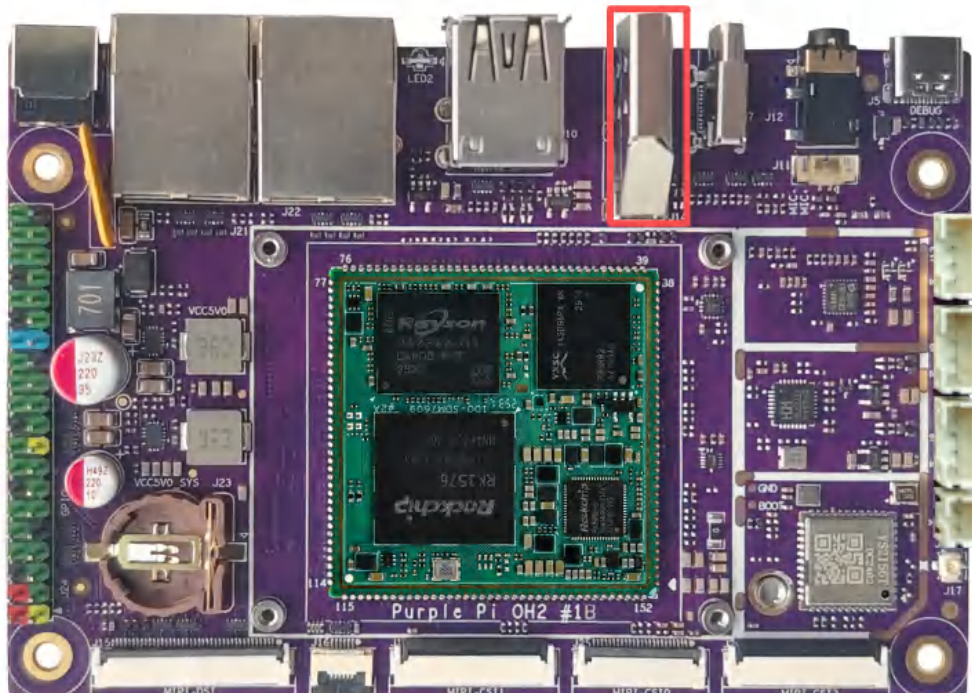
可以使用 USB-C 转 HDMI 高清线连接 HDMI 显示器输出画面，如下图所示：



- 最高支持4K@30fps输出

4.8.2. HDMI-TX

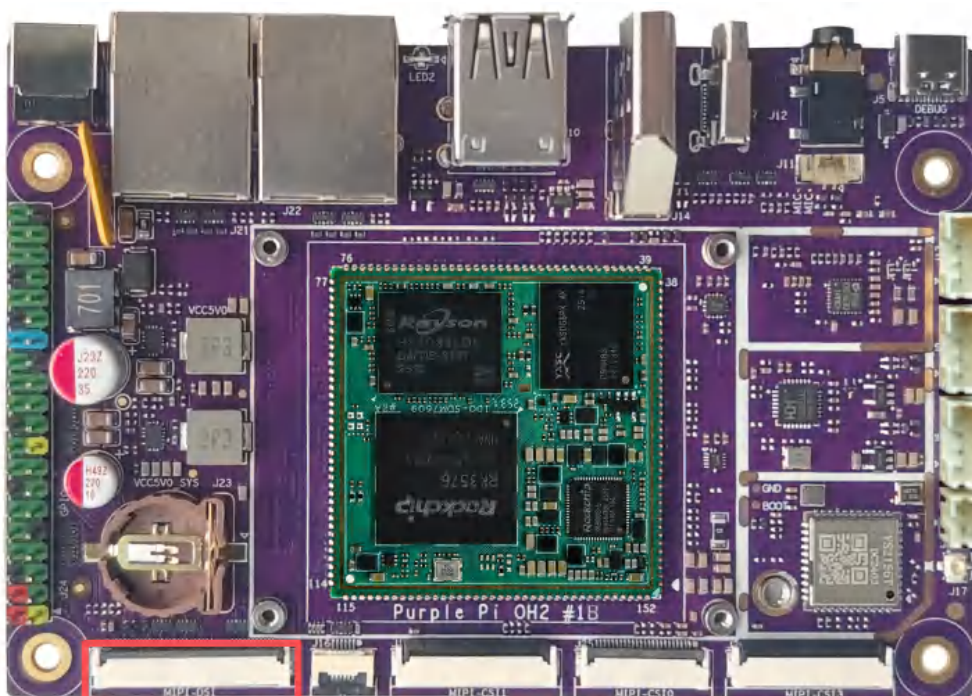
标准HDMI-A接口J14，如下图所示：



- 最高支持 HDMI2.1 8K@60fps 输出

4.8.3. MIPI

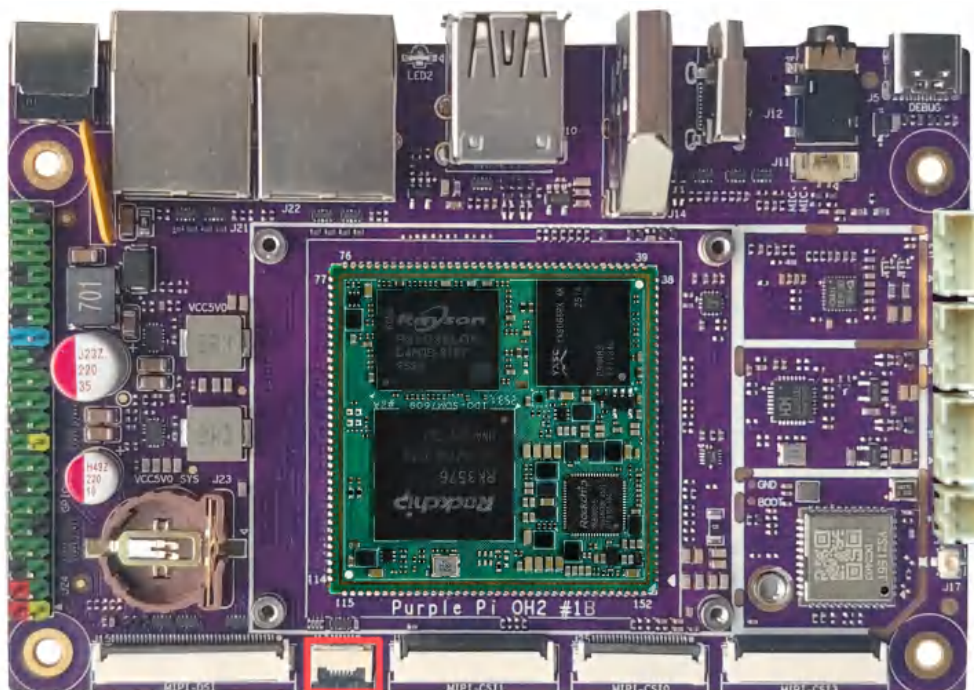
J15 MIPI接口为40Pin FPC 0.5mm 上接，如下图所示：



- MIPI供电为3.3V

4.9. TP接口

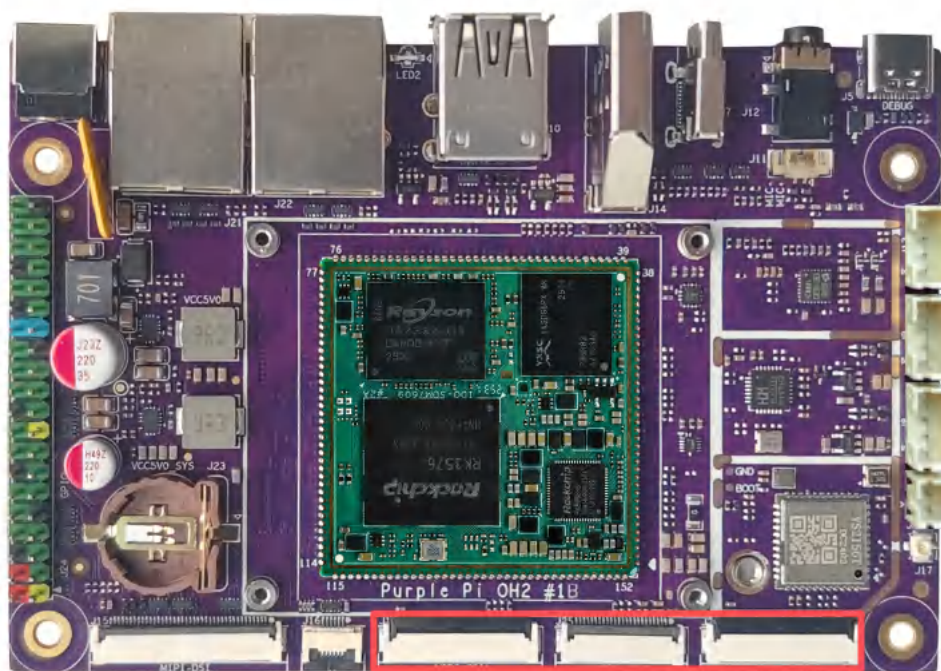
TP接口为(J16) 6Pin FPC 0.5mm座子，如下图所示：



- 触摸 TP 接口接线方法为下接

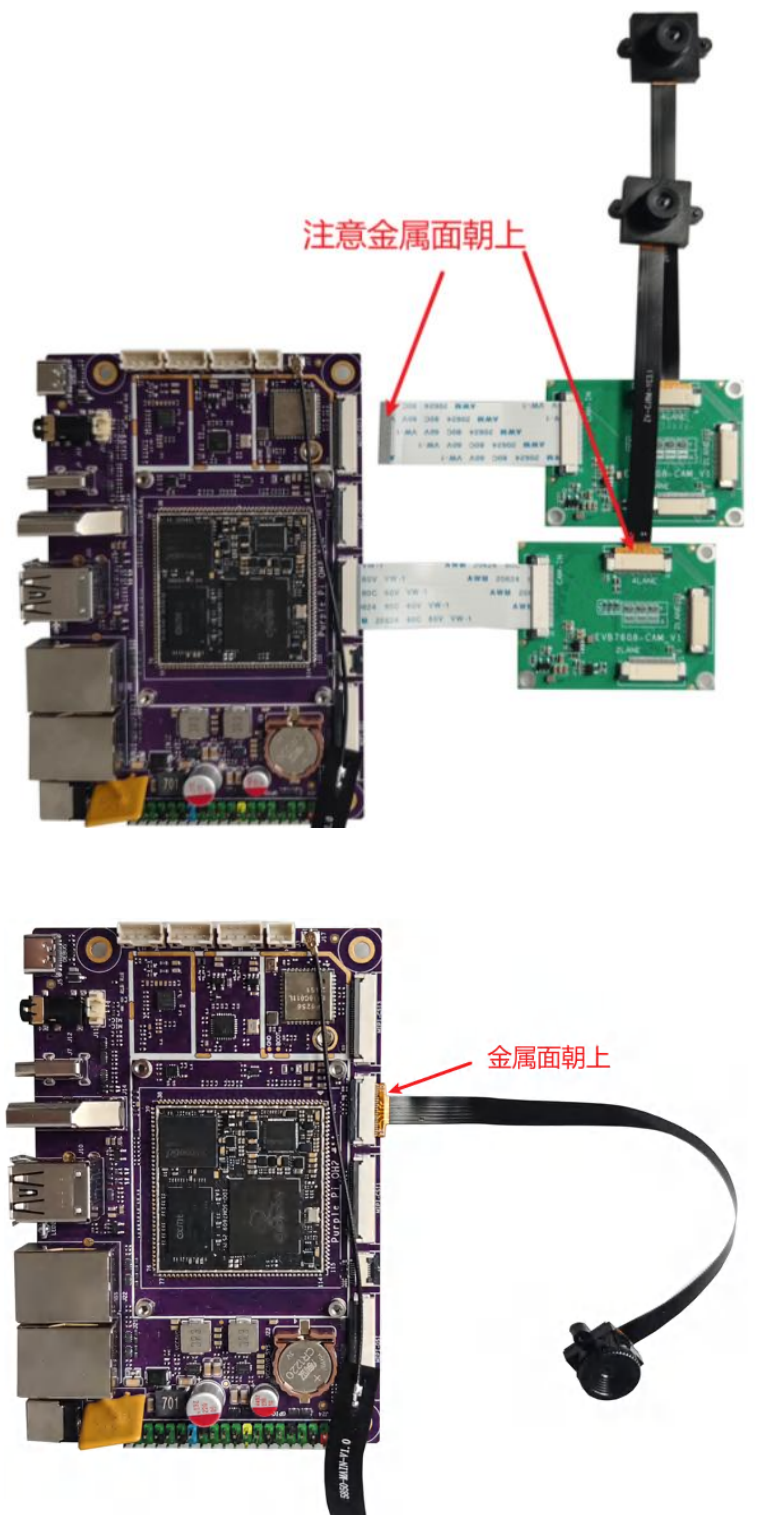
4.10. MIPI Camera

主板共有3路摄像头接口，分别是J19、J20和J25，如下图所示：



J19和J20使用FPC05-PUW-32P抽拉式上接座子，J25使用FPC05-PUW-24P抽拉式上接座子。其中J19和J25默认适配IMX415摄像头，J20默认适配OV13855摄像头。

接法如下图所示：



4.10.1. 预览命令：

预览J19摄像头：

```

1 root@industio:~# export DISPLAY=:0
2 root@industio:~# gst-launch-1.0 v4l2src device=/dev/video42 ! video/x-raw,
  format=NV12, width=1920, height=1080, framerate=30/1 ! videoconvert ! autov
  ideosink sync=false

```

预览J20摄像头：

```

1 root@industio:~# export DISPLAY=:0
2 root@industio:~# gst-launch-1.0 v4l2src device=/dev/video51 ! video/x-raw,
  format=NV12, width=1920, height=1080, framerate=30/1 ! videoconvert ! autov
  ideosink sync=false

```

预览J25摄像头：

```

1 root@industio:~# export DISPLAY=:0
2 root@industio:~# gst-launch-1.0 v4l2src device=/dev/video33 ! video/x-raw,
  format=NV12, width=1920, height=1080, framerate=30/1 ! videoconvert ! autov
  ideosink sync=false

```

4.10.2. 抓图：

抓取J19摄像头图片：

```

1 root@industio:~# v4l2-ctl --verbose -d /dev/video42 --set-fmt-video=width=1
  920,height=1080,pixelformat='NV12' --stream-mmap=4 --stream-skip=3 --strea
  m-to=./test-1920x1080.yuv --stream-count=1 --stream-poll

```

抓取J20摄像头图片：

```

1 root@industio:~# v4l2-ctl --verbose -d /dev/video51 --set-fmt-video=width=1
  920,height=1080,pixelformat='NV12' --stream-mmap=4 --stream-skip=3 --strea
  m-to=./test-1920x1080.yuv --stream-count=1 --stream-poll

```

抓取J25摄像头图片：

▼ Plain Text |

```
1 root@industio:~# v4l2-ctl --verbose -d /dev/video33 --set-fmt-video=width=1920,height=1080,pixelformat='NV12' --stream-mmap=4 --stream-skip=3 --stream-to=./test-1920x1080.yuv --stream-count=1 --stream-poll
```

查看照片：

▼ Plain Text |

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

4.10.3. 抓视频：

抓取J19摄像头视频：

▼ Plain Text |

```
1 root@industio:~# v4l2-ctl --stream-mmap=4 -d /dev/video42 --set-fmt-video=width=1920,height=1080,pixelformat='NV12' --stream-to=./output.yuv
```

抓取J20摄像头视频：

▼ Plain Text |

```
1 root@industio:~# v4l2-ctl --stream-mmap=4 -d /dev/video51 --set-fmt-video=width=1920,height=1080,pixelformat='NV12' --stream-to=./output.yuv
```

抓取J25摄像头视频：

▼ Plain Text |

```
1 root@industio:~# v4l2-ctl --stream-mmap=4 -d /dev/video33 --set-fmt-video=width=1920,height=1080,pixelformat='NV12' --stream-to=./output.yuv
```

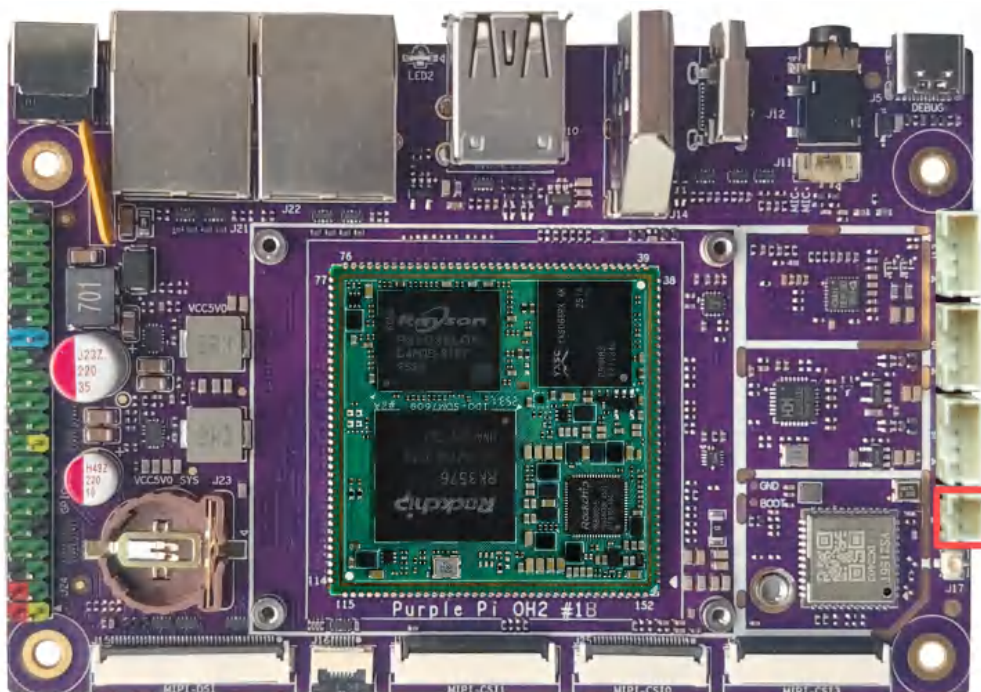
播放视频：

▼ Plain Text |

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

4.11. Fan 风扇

PH2 5V Fan风扇接口J3，如下图所示：



4.12. 三轴传感器

主板配置了1路三轴传感器，传感器型号是SC7A20HTR，在系统中对应的设备节点为/sys/bus/iio/devices/iio:device0。

4.12.1. 设置采样率

默认采样率为1Hz，最高支持400Hz，采样率设置方法如下：

```
1 echo 400 > /sys/bus/iio/devices/iio:device0/sampling_frequency
```

4.12.2. 获取三轴传感器数据

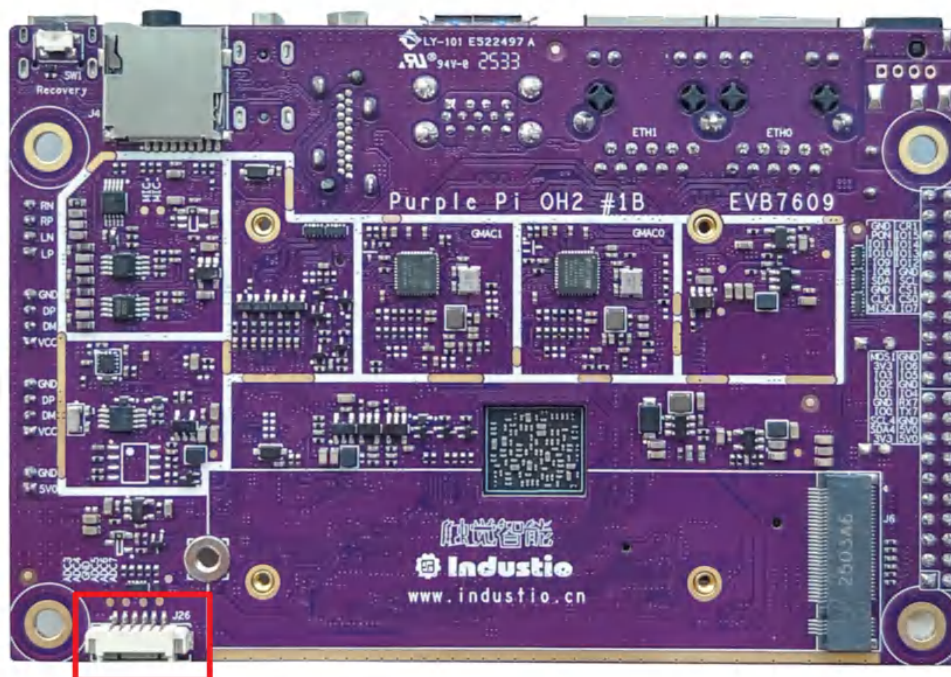
```

1 # 获取X轴加速度
2 cat /sys/bus/iio/devices/iio:device0/in_accel_x_raw
3
4 # 获取Y轴加速度
5 cat /sys/bus/iio/devices/iio:device0/in_accel_y_raw
6
7 # 获取Z轴加速度
8 cat /sys/bus/iio/devices/iio:device0/in_accel_z_raw

```

4.13. ADC

主板ADC接口位于背面J26，支持5路ADC输入，精度是12位，最高输入电压1.8V。



5路ADC对应的设备节点如下表：

pin	名称	设备节点
1	ADC3	/sys/bus/iio/devices/iio:device1/in_voltage3_raw
2	ADC4	/sys/bus/iio/devices/iio:device1/in_voltage4_raw
3	GND	—
4	ADC5	/sys/bus/iio/devices/iio:device1/in_voltage5_raw

5	ADC6	/sys/bus/iio/devices/iio:device1/in_voltage6_ra
6	ADC7	/sys/bus/iio/devices/iio:device1/in_voltage7_ra

4.13.1. 获取输入ADC raw值

通过读取设备节点直接获取ADC raw值，如获取ADC3的raw值：

▼

Shell

```

1 root@industio:/# cat /sys/bus/iio/devices/iio:device1/in_voltage3_raw
2 3770

```

4.13.2. ADC输入电压计算公式

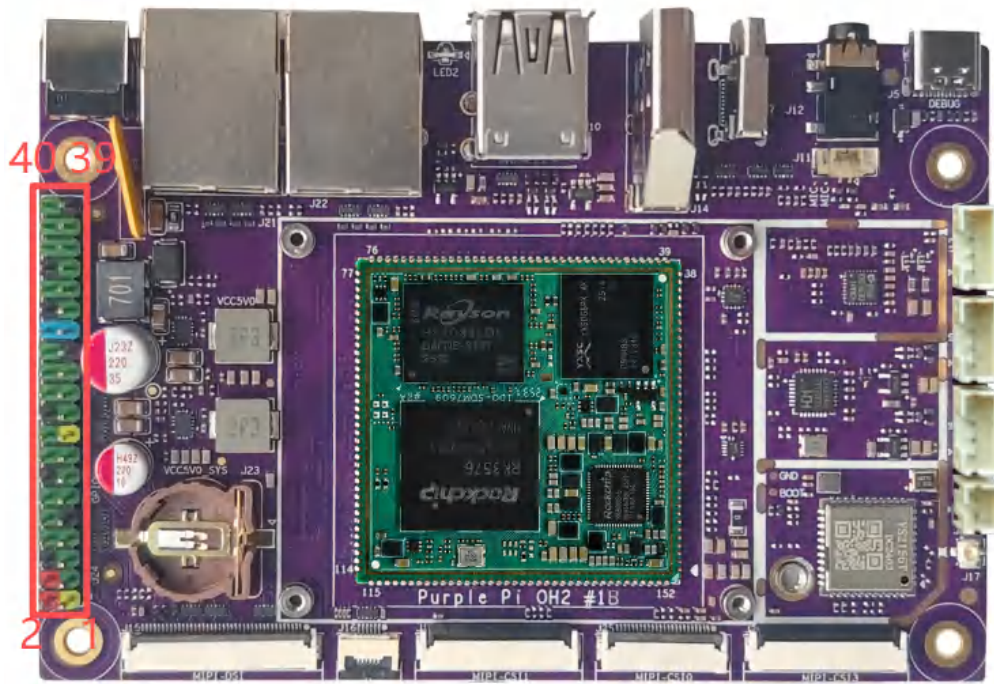
$$V_{in} = (raw/4096) \times 1.8v$$

其中 V_{in} 表示ADC输入电压，raw表示ADC raw值。比如ADC3的raw值为258，则ADC3输入电压为：

$$V_{in} = (3770/4096) \times 1.8v = 1.657v$$

4.14. 扩展接口

主板预留了40pin双排针扩展接口，包括1路UART，1路SPI，2路I2C和16个GPIO。40pin双排针引脚定义如下：



pin	功能	pin	功能
1	3.3V	2	5V
3	I2C4_SDA	4	5V
5	I2C4_SCL	6	GND
7	GPIO76	8	UART7_TX
9	GND	10	UART7_RX
11	GPIO75	12	GPIO77
13	GPIO72	14	GND
15	GPIO73	16	GPIO74
17	3.3V	18	GPIO136
19	SPI0_MOSI	20	GND
21	SPI0_MISO	22	GPIO137
23	SPI0_SCLK	24	SPI0_CS0
25	GND	26	SPI0_CS1
27	I2C7_SDA	28	I2C7_SCL

29	GPIO56	30	GND
31	GPIO57	32	GPIO132
33	GPIO58	34	GND
35	GPIO59	36	GPIO134
37	POWER_KEY	38	GPIO98
39	GND	40	GPIO99

4.14.1. GPIO

主板扩展接口预留了16个GPIO，各个GPIO引脚定义描述，如下图所示：

序号	pin	GPIO 标号	GPIO NUM	LIBGPIO节点
1	7	GPIO2_B4	76	gpiochip2 12
2	11	GPIO2_B3	75	gpiochip2 11
3	12	GPIO2_B5	77	gpiochip2 13
4	13	GPIO2_B0	72	gpiochip2 8
5	15	GPIO2_B1	73	gpiochip2 9
6	16	GPIO2_B2	74	gpiochip2 10
7	18	GPIO4_B0	136	gpiochip4 8
8	22	GPIO4_B1	137	gpiochip4 9
9	29	GPIO1_D0	56	gpiochip1 24
10	31	GPIO1_D1	57	gpiochip1 25
11	32	GPIO4_A4	132	gpiochip4 4
12	33	GPIO1_D2	58	gpiochip1 26
13	35	GPIO1_D3	59	gpiochip1 27
14	36	GPIO4_A6	134	gpiochip4 6
15	38	GPIO3_A2	98	gpiochip3 2

16	40	GPIO3_A3	99	gpiochip3 3
----	----	----------	----	-------------

GPIO控制，根据上表【LIBGPIO节点】进行命令操作

libgpiodC

```
1 #安装命令
2 apt install -y libgpiod-dev gpiod
3
4 # 设置 GPIO2_B4 输出高电平状态
5 gpioset gpiochip2 12=1
6
7 # 设置 GPIO2_B4 输出低电平状态
8 gpioset gpiochip2 12=0
9
10 # 获取 GPIO2_B4 输入电平状态
11 gpioget gpiochip2 12
```

4.14.2. UART

主板扩展接口预留了1路UART（TTL电平），对应的pin脚和设备节点如下：

pin脚	功能定义	默认电平类型	设备节点
8	UART_TX	TTL	/dev/ttyS7
10	UART_RX		

Plain Text

```
1 #microcom安装命令
2 root@linaro-alip:~# apt-get install microcom
3 # 测试
4 root@linaro-alip:~# microcom -s 115200 -p /dev/ttyS7
```

4.14.3. I2C

主板扩展接口预留2路I2C，对应的pin脚和设备节点如下：

pin脚	功能定义	设备节点
3	I2C4_SDA	/dev/i2c-4

5	I2C4_SCL	/dev/i2c-7
27	I2C7_SDA	
28	I2C7_SCL	

4.14.4. SPI

主板扩展接口预留1路I2C，对应的pin脚和设备节点如下：

pin脚	功能定义	设备节点
19	SPI_MOSI	/dev/spidev0.0
21	SPI_MISO	
23	SPI_SCLK	
24	SPI_CS0	
26	SPI_CS1	/dev/spidev0.1

4.14.5. POWER_KEY

主板扩展接口第37 pin为power_key，对应的设备节点为/dev/input/event0，输入低电平时系统上报键值KEY_POWER，可以通过evtest用具测试：

```

# evtest
No device specified, trying to scan all of /dev/input/event*
Available devices:
/dev/input/event0:      rk805 pwrkey
/dev/input/event1:      headset-keys
/dev/input/event2:      rockchip-es8388 Headset
/dev/input/event3:      bt-powerkey
/dev/input/event4:      adc-keys
/dev/input/event5:      rockchip-hdmi rockchip-hdmi
Select the device event number [0-5]: 0
Input driver version is 1.0.1
Input device ID: bus 0x19 vendor 0x0 product 0x0 version 0x0
Input device name: "rk805 pwrkey"
Supported events:
  Event type 0 (EV_SYN)
  Event type 1 (EV_KEY)
    Event code 116 (KEY_POWER)
Properties:
Testing ... (interrupt to exit)
Event: time 1756374595.073583, type 1 (EV_KEY), code 116 (KEY_POWER), value 1
Event: time 1756374595.073583, ----- SYN_REPORT -----
Event: time 1756374595.200850, type 1 (EV_KEY), code 116 (KEY_POWER), value 0
Event: time 1756374595.200850, ----- SYN_REPORT -----

```

4.15. GPU

执行glmark2-es2 开启gpu测试窗口:

```

1 root@industio:/# glmark2-es2
2 arm_release_ver: g13p0-01eac0, rk_so_ver: 11
3 =====
4     glmark2 2023.01
5     =====
6     OpenGL Information
7     GL_VENDOR:      ARM
8     GL_RENDERER:    Mali-G52
9     GL_VERSION:     OpenGL ES 3.2 v1.g13p0-01eac0.e22bb8d1d7959f189f908963
10    172c514e
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: 227 FrameTime: 4.422 ms
15 ▾ [build] use-vbo=true: FPS: 222 FrameTime: 4.520 ms
16 ▾ [texture] texture-filter=nearest: FPS: 276 FrameTime: 3.631 ms
17 ▾ [texture] texture-filter=linear: FPS: 218 FrameTime: 4.599 ms
18 ▾ [texture] texture-filter=mipmap: FPS: 213 FrameTime: 4.710 ms
19 ▾ [shading] shading=gouraud: FPS: 169 FrameTime: 5.942 ms
20 ▾ [shading] shading=blinn-phong-inf: FPS: 188 FrameTime: 5.336 ms
21 ▾ [shading] shading=phong: FPS: 177 FrameTime: 5.667 ms
22 ▾ [shading] shading=cel: FPS: 168 FrameTime: 5.955 ms
23 ▾ [bump] bump-render=high-poly: FPS: 161 FrameTime: 6.247 ms
24 ▾ [bump] bump-render=normals: FPS: 198 FrameTime: 5.054 ms
25 ▾ [bump] bump-render=height: FPS: 189 FrameTime: 5.301 ms
26 ▾ [effect2d] kernel=0,1,0;1,-4,1;0,1,0;: FPS: 158 FrameTime: 6.354 ms
27 ▾ [effect2d] kernel=1,1,1,1,1;1,1,1,1,1;1,1,1,1,1;: FPS: 118 FrameTime: 8.54
28    4 ms
29 ▾ [pulsar] light=false:quads=5:texture=false: FPS: 186 FrameTime: 5.387 ms
30 ▾ [desktop] blur-radius=5:effect=blur:passes=1:separable=true:windows=4: FPS
31    : 118 FrameTime: 8.491 ms
32 ▾ [desktop] effect=shadow:windows=4: FPS: 148 FrameTime: 6.794 ms
33 ▾ [buffer] columns=200:interleave=false:update-dispersion=0.9:update-fractio
34    n=0.5:update-method=map: FPS: 82 FrameTime: 12.242 ms
35 ▾ [buffer] columns=200:interleave=false:update-dispersion=0.9:update-fractio
36    n=0.5:update-method=subdata: FPS: 83 FrameTime: 12.054 ms
37 ▾ [buffer] columns=200:interleave=true:update-dispersion=0.9:update-fraction
38    =0.5:update-method=map: FPS: 98 FrameTime: 10.270 ms
39 ▾ [ideas] speed=duration: FPS: 139 FrameTime: 7.215 ms
40 ▾ [jellyfish] <default>: FPS: 227 FrameTime: 4.406 ms
41 ▾ [terrain] <default>: FPS: 57 FrameTime: 17.605 ms
42 ▾ [shadow] <default>: FPS: 168 FrameTime: 5.974 ms
43 ▾ [refract] <default>: FPS: 83 FrameTime: 12.066 ms
44 ▾ [conditionals] fragment-steps=0:vertex-steps=0: FPS: 301 FrameTime: 3.325
45    ms

```

```

39 [conditionals] fragment-steps=5:vertex-steps=0: FPS: 286 FrameTime: 3.501
40 ms
41 [conditionals] fragment-steps=0:vertex-steps=5: FPS: 278 FrameTime: 3.602
42 ms
43 [function] fragment-complexity=low:fragment-steps=5: FPS: 265 FrameTime:
44 3.782 ms
45 [function] fragment-complexity=medium:fragment-steps=5: FPS: 276 FrameTime
46 : 3.629 ms
47 [loop] fragment-loop=false:fragment-steps=5:vertex-steps=5: FPS: 289 Frame
48 Time: 3.470 ms
[loop] fragment-steps=5:fragment-uniform=false:vertex-steps=5: FPS: 290 Fr
ameTime: 3.449 ms
[loop] fragment-steps=5:fragment-uniform=true:vertex-steps=5: FPS: 287 Fra
meTime: 3.490 ms
=====
glmark2 Score: 191
=====

```

4.16. NPU

参考《Purple Pi OH2 YOLOv5开发手册》文档。

使用yolov5进行rknn模型转换测试：

4.16.1. 编译rknn_yolov5_demo：

```

1 root@industio:/# unzip rknn-toolkit2-master.zip
2 root@industio:/# cd rknn-toolkit2-master/rknpu2/examples/3rdparty/mpp/Linu
x/aarch64
3 root@industio:/# ln -srf librockchip_mpp.so.0 librockchip_mpp.so
4 root@industio:/# ln -srf librockchip_mpp.so.0 librockchip_mpp.so.1
5
6 root@industio:/# cd -
7 root@industio:/# cd rknn-toolkit2-master/rknpu2/examples/rknn_yolov5_demo
8 root@industio:/# chmod a+x build-linux.sh
9 root@industio:/# export GCC_COMPILER=/usr/bin/aarch64-linux-gnu
10 root@linaro-alip:/mnt/rknn-toolkit2-master/rknpu2/examples/rknn_yolov5_dem
o# ./build-linux.sh -t rk3576 -a aarch64 -b Release
11 ./build-linux.sh -t rk3576 -a aarch64 -b Release
12 /usr/bin/aarch64-linux-gnu
13 =====
14 TARGET_SOC=RK3576
15 TARGET_ARCH=aarch64
16 BUILD_TYPE=Release
17 BUILD_DIR=/mnt/rknn-toolkit2-master/rknpu2/examples/rknn_yolov5_demo/buil
d/build_RK3576_linux_aarch64_Release
18 CC=/usr/bin/aarch64-linux-gnu-gcc
19 CXX=/usr/bin/aarch64-linux-gnu-g++
20 =====
21 -- Configuring done
22 -- Generating done
23 -- Build files have been written to: /mnt/rknn-toolkit2-master/rknpu2/exam
ples/rknn_yolov5_demo/build/build_RK3576_linux_aarch64_Release
24 ▾ [ 60%] Built target rknn_yolov5_video_demo
25 ▾ [100%] Built target rknn_yolov5_demo
26 ▾ [ 40%] Built target rknn_yolov5_demo
27 ▾ [100%] Built target rknn_yolov5_video_demo
28 Install the project...
29 -- Install configuration: "Release"
30 -- Up-to-date: /mnt/rknn-toolkit2-master/rknpu2/examples/rknn_yolov5_demo/
install/rknn_yolov5_demo_Linux/./rknn_yolov5_demo
31 -- Up-to-date: /mnt/rknn-toolkit2-master/rknpu2/examples/rknn_yolov5_demo/
install/rknn_yolov5_demo_Linux/lib/librknnrt.so
32 -- Up-to-date: /mnt/rknn-toolkit2-master/rknpu2/examples/rknn_yolov5_demo/
install/rknn_yolov5_demo_Linux/lib/librga.so
33 -- Up-to-date: /mnt/rknn-toolkit2-master/rknpu2/examples/rknn_yolov5_demo/
install/rknn_yolov5_demo_Linux/./model/RK3576
34 -- Up-to-date: /mnt/rknn-toolkit2-master/rknpu2/examples/rknn_yolov5_demo/
install/rknn_yolov5_demo_Linux/./model/RK3576/yolov5s-640-640.rknn
35 -- Up-to-date: /mnt/rknn-toolkit2-master/rknpu2/examples/rknn_yolov5_demo/
install/rknn_yolov5_demo_Linux/./model/bus.jpg

```

```
36  -- Up-to-date: /mnt/rknn-toolkit2-master/rknpu2/examples/rknn_yolov5_demo/  
install/rknn_yolov5_demo_Linux/./model/coco_80_labels_list.txt  
37  -- Up-to-date: /mnt/rknn-toolkit2-master/rknpu2/examples/rknn_yolov5_demo/  
install/rknn_yolov5_demo_Linux/./rknn_yolov5_video_demo  
38  -- Up-to-date: /mnt/rknn-toolkit2-master/rknpu2/examples/rknn_yolov5_demo/  
install/rknn_yolov5_demo_Linux/lib/librockchip_mpp.so  
39  -- Up-to-date: /mnt/rknn-toolkit2-master/rknpu2/examples/rknn_yolov5_demo/  
install/rknn_yolov5_demo_Linux/lib/libmk_api.so  
40
```

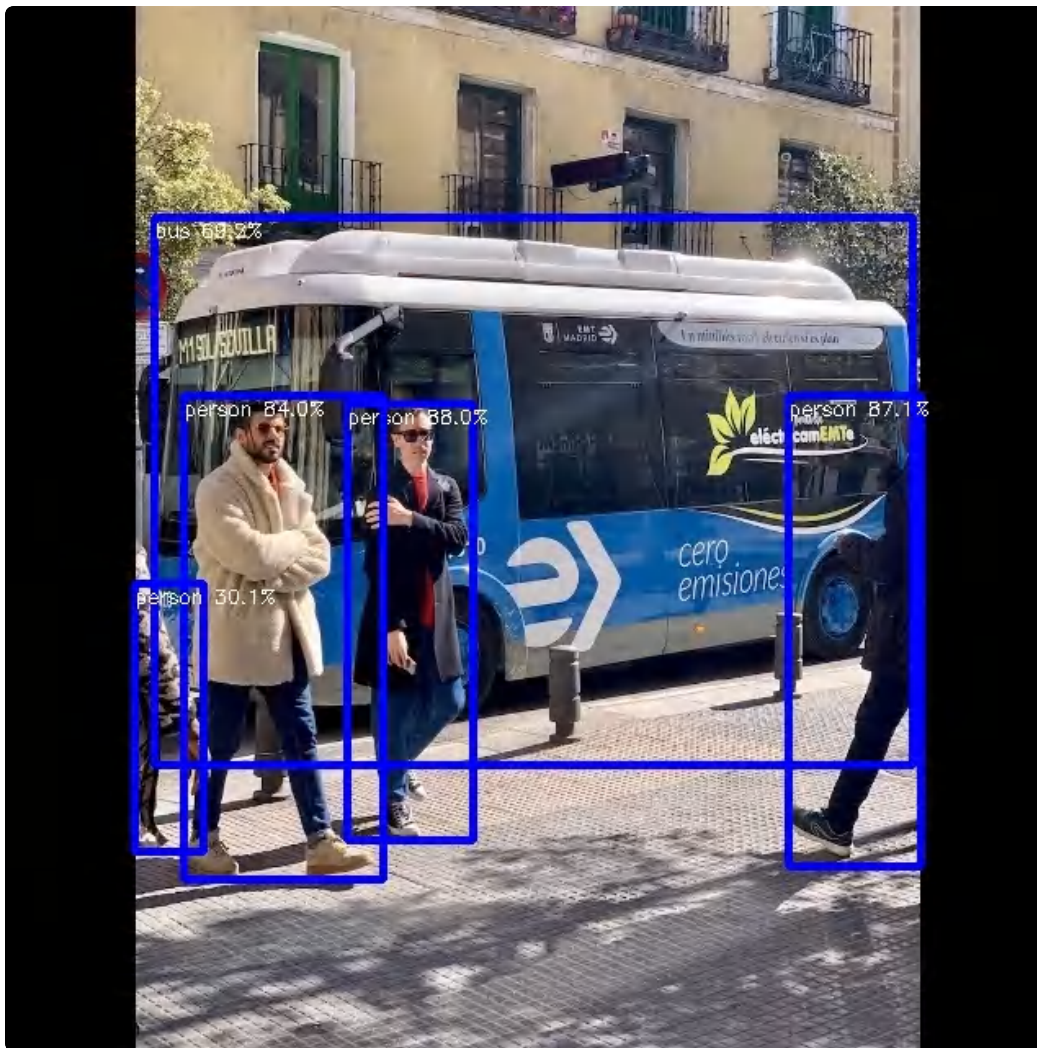
4.16.2. 测试：

```

1 root@industio:/# cd install/rknn_yolov5_demo_Linux
2 root@industio:/# chmod a+x rknn_yolov5_demo
3 root@industio:/# ./rknn_yolov5_demo ./model/RK3576/yolov5s-640-640.rknn ./
  model/bus.jpg
4 post process config: box_conf_threshold = 0.25, nms_threshold = 0.45
5 Loading mode...
6 sdk version: 2.3.2 (429f97ae6b@2025-04-09T09:09:27) 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=output0, 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=0.003922
10 index=1, name=286, n_dims=4, dims=[1, 255, 40, 40], n_elems=408000, size
  =408000, w_stride = 0, size_with_stride=491520, fmt=NCHW, type=INT8, qnt_t
  ype=AFFINE, zp=-128, scale=0.003922
11 index=2, name=288, n_dims=4, dims=[1, 255, 20, 20], n_elems=102000, size
  =102000, w_stride = 0, size_with_stride=163840, fmt=NCHW, type=INT8, qnt_t
  ype=AFFINE, zp=-128, scale=0.003922
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 41.249000 ms
17 loadLabelName ./model/coco_80_labels_list.txt
18 person @ (209 243 286 510) 0.879723
19 person @ (479 238 560 526) 0.870588
20 person @ (109 237 232 534) 0.828112
21 bus @ (93 129 553 464) 0.700761
22 person @ (79 353 122 517) 0.307297
23 save detect result to ./out.jpg
24 loop count = 10 , average run 28.800200 ms

```

执行完后会得到转换后的模型图片，如下所示：



4.17. QT

运行QT动画测试程序：



Plain Text |

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

运行QT虚拟键盘测试程序：



Plain Text |

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

运行QT Webengine测试程序：

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



4.18. USB 摄像头

4.18.1. 确认USB摄像头设备节点

```
1 industio@industio:~$ grep ' /sys/class/video4linux/video*/name
2 ...
3 /sys/class/video4linux/video73/name:Webcam gadget: UVC Color Camera
4 /sys/class/video4linux/video74/name:Webcam gadget: UVC Color Camera
5 ...
```

从以上信息得知USB摄像头设备节点为/dev/video73和/dev/video74, 一般取较小的一个节点即/dev/video73。

4.18.2. 查看USB摄像头支持的图像格式

```

1  industio@industio:~$ v4l2-ctl -d /dev/video73 --list-formats-ext
2  ioctl: VIDIOC_ENUM_FMT
3      Type: Video Capture
4
5  [0]: 'MJPG' (Motion-JPEG, compressed)
6      Size: Discrete 1920x1080
7          Interval: Discrete 0.033s (30.000 fps)
8          Interval: Discrete 0.040s (25.000 fps)
9          Interval: Discrete 0.050s (20.000 fps)
10         Interval: Discrete 0.067s (15.000 fps)
11         Interval: Discrete 0.100s (10.000 fps)
12         Interval: Discrete 0.200s (5.000 fps)
13     Size: Discrete 1280x720
14         Interval: Discrete 0.033s (30.000 fps)
15         Interval: Discrete 0.040s (25.000 fps)
16         Interval: Discrete 0.050s (20.000 fps)
17         Interval: Discrete 0.067s (15.000 fps)
18         Interval: Discrete 0.100s (10.000 fps)
19         Interval: Discrete 0.200s (5.000 fps)
20     ...
21 [1]: 'YUYV' (YUYV 4:2:2)
22     Size: Discrete 640x480
23         Interval: Discrete 0.033s (30.000 fps)
24         Interval: Discrete 0.067s (15.000 fps)
25         Interval: Discrete 0.100s (10.000 fps)
26     Size: Discrete 1280x720
27         Interval: Discrete 0.067s (15.000 fps)
28         Interval: Discrete 0.100s (10.000 fps)
29         Interval: Discrete 0.200s (5.000 fps)
30     Size: Discrete 1920x1080
31         Interval: Discrete 0.067s (15.000 fps)
32         Interval: Discrete 0.100s (10.000 fps)
33         Interval: Discrete 0.200s (5.000 fps)
34 [2]: 'H264' (H.264, compressed)
35     Size: Discrete 1920x1080
36         Interval: Discrete 0.033s (30.000 fps)
37         Interval: Discrete 0.040s (25.000 fps)
38         Interval: Discrete 0.050s (20.000 fps)
39         Interval: Discrete 0.067s (15.000 fps)
40         Interval: Discrete 0.100s (10.000 fps)
41         Interval: Discrete 0.200s (5.000 fps)
42     Size: Discrete 1280x720
43         Interval: Discrete 0.033s (30.000 fps)
44         Interval: Discrete 0.040s (25.000 fps)
45         Interval: Discrete 0.050s (20.000 fps)

```

```
46                                     Interval: Discrete 0.067s (15.000 fps)
47                                     Interval: Discrete 0.100s (10.000 fps)
48                                     Interval: Discrete 0.200s (5.000 fps)
49                                     ...
```

从以上信息得知USB摄像头输出图像支持以下格式：

1. MJPG 1920x1080 30fps
2. MJPG 1280x720 30fps
3. YUYV 640x480 30fps
4. YUYV 1280x720 15fps
5. H264 1920x1080 30fps
6. H264 1280x720 30fps

4.18.3. 预览USB摄像头图像

使用gststreamer预览USB摄像头图像。

4.18.3.1. MJPG格式

由于USB摄像头输出图像是MJPG格式，需指定使用mppjpegdec解码器：

```
1  gst-launch-1.0 v4l2src device=/dev/video73 ! image/jpeg,width=1920,height=1080,framerate=30/1 ! mppjpegdec ! videoconvert ! autovideosink
```

4.18.3.2. YUYV格式

如果USB摄像头输出图像是YUV格式，则不需要指定mppjpegdec解码器：

```
1  gst-launch-1.0 v4l2src device=/dev/video73 ! video/x-raw,format=YUY2, width=640, height=480, framerate=30/1 ! videoconvert ! autovideosink sync=false
```

4.18.3.3. H264格式

```
1  gst-launch-1.0 v4l2src device=/dev/video73 ! video/x-h264,width=1920,height=1080,framerate=30/1 ! mppvideodec ! videoconvert ! autovideosink
```