

IDO-EVB1126B-V1 Linux使用手册

1. 硬件资源概况

1.1. 主板照片

1.2. 硬件资源

2. IDO-EVB1126B-V1A-Buildroot系统

2.1. 功能及接口使用方法

2.2. 调试串口

2.3. 按键

2.4. 网络

2.4.1. Ethernet

2.4.2. WIFI

2.4.3. Bluetooth

2.4.3.1. 命令行连接

2.4.4. 4G/5G

2.5. UART

2.6. Audio

2.6.1. Speaker

2.6.2. MIC

2.7. RTC

2.8. USB

2.8.1. OTG

2.8.2. USB3.0

2.8.3. USB2.0-A

2.8.4. USB2.0-PH

2.9. TF Card

2.10. CANFD

2.11. LCD显示

2.11.1. HDMI-TX

2.11.2. MIPI-TX

2.12. TP

2.13. MIPI Camera

2.13.1. IMX415

2.13.1.1. 预览

2.13.1.2. 抓图

2.13.1.3. 抓视频

2.13.2. TP2815

2.13.2.1. 预览

2.14. FAN 风扇

2.15. GPIO

2.16. pwm

2.17. LED

2.18. ADC

2.19. I2C

2.20. SPI NAND

2.21. NPU

2.21.1. yolov5测试:

2.21.1.1. rknn_yolov5_demo:

2.21.1.2. rknn_yolov5_video_demo:

2.21.2. yolo11 测试

3. IDO-EVB1126B-V1A-Debian11系统

3.1. 功能及接口使用方法

3.2. 调试串口

3.3. 按键

3.4. 网络

3.4.1. Ethernet

3.4.2. WIFI

3.4.3. Bluetooth

3.4.3.1. 命令行连接

3.4.4. 4G/5G

3.5. UART

3.6. Audio

3.6.1. Speaker

3.6.2. MIC

3.7. RTC

3.8. USB

3.8.1. OTG

3.8.2. USB3.0

3.8.3. USB2.0-A

3.8.4. USB2.0-PH

3.9. TF Card

3.10. CANFD

3.11. LCD显示

3.11.1. HDMI-TX

3.11.2. MIPI-TX

3.12. TP

3.13. MIPI Camera

3.13.1. IMX415

3.13.1.1. 预览

3.13.1.2. 抓图

3.13.1.3. 抓视频

3.13.2. TP2815 AHD

3.14. FAN 风扇

3.15. GPIO

3.16. pwm

3.17. LED

3.18. ADC

3.19. I2C

3.20. SPI NAND

3.21. NPU

3.21.1. yolov5测试:

3.21.1.1. rknn_yolov5_demo:

3.21.1.2. rknn_yolov5_video_demo:

3.21.2. yolo11 测试



IDO-EVB1126B-V1

Linux使用手册

深圳触觉智能科技有限公司

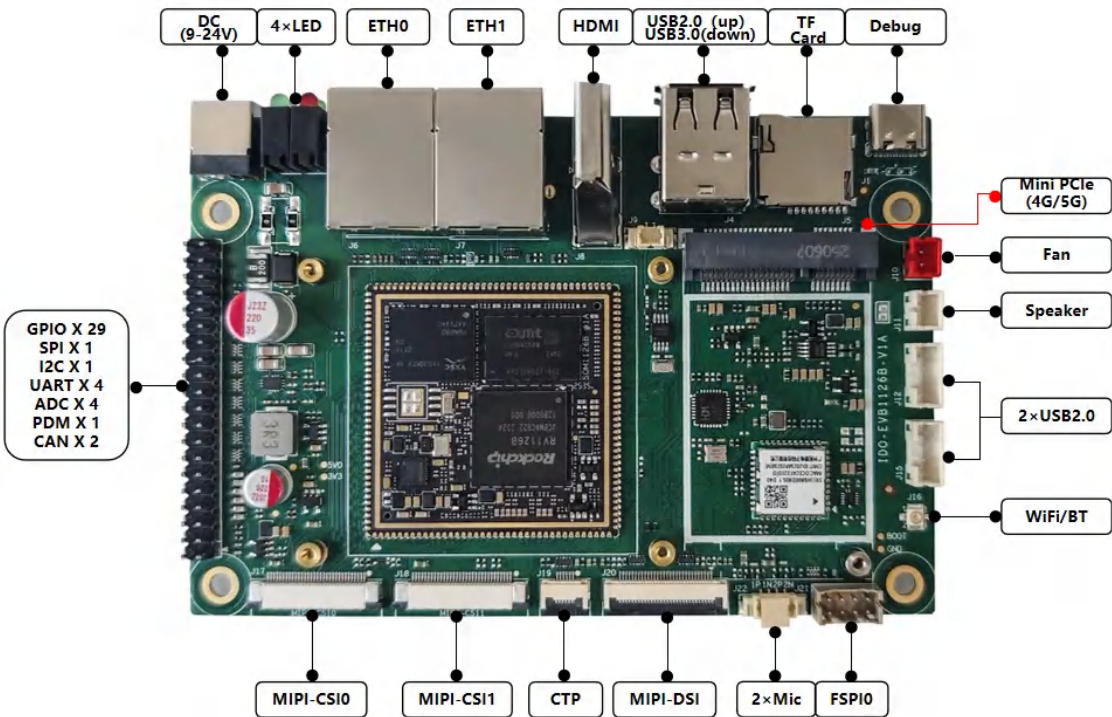
www.industio.cn

文档修订历史

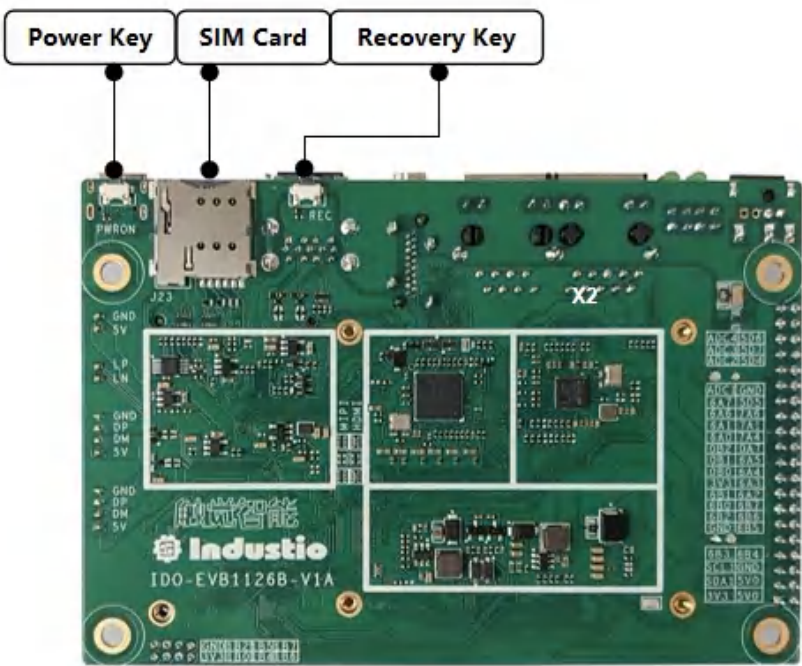
版本	PCBA版本	修订内容	修订	审核	日期
V1.0	V1A	创建文档	LJS	IDO	2025/10/14
V1.1	V1B	以太网调整	LJS	IDO	2025/10/31

1. 硬件资源概况

1.1. 主板照片



IDO-EVB1126B-V1A 正面实物图



IDO-EVB1126B-V1A 背面实物图

1.2. 硬件资源

序号	名称	描述
1	内核版本	Linux 6.1.141
2	系统版本	Buildroot2024/Debian11
3	内存	LPDDR4/4X (1/2/4G)
4	存储	Flash/eMMC (8/16/32GB)
5	供电	DC接口12V@2A
6	显示	HDMI-TX ×1
		MIPI ×1 (二选一)
7	触摸	I2C-TP ×1
8	USB OTG	USB OTG Type-A ×1
9	USB HOST	USB2.0 HOST(Type-A) ×1
		USB2.0 HOST(PH2.0) ×1
10	TF Card	SDIO3.0 TF Card ×1
11	以太网	千兆以太网 ×1
		百兆以太网 × 1
12	WIFI/BT	AIC8800D80 ×1

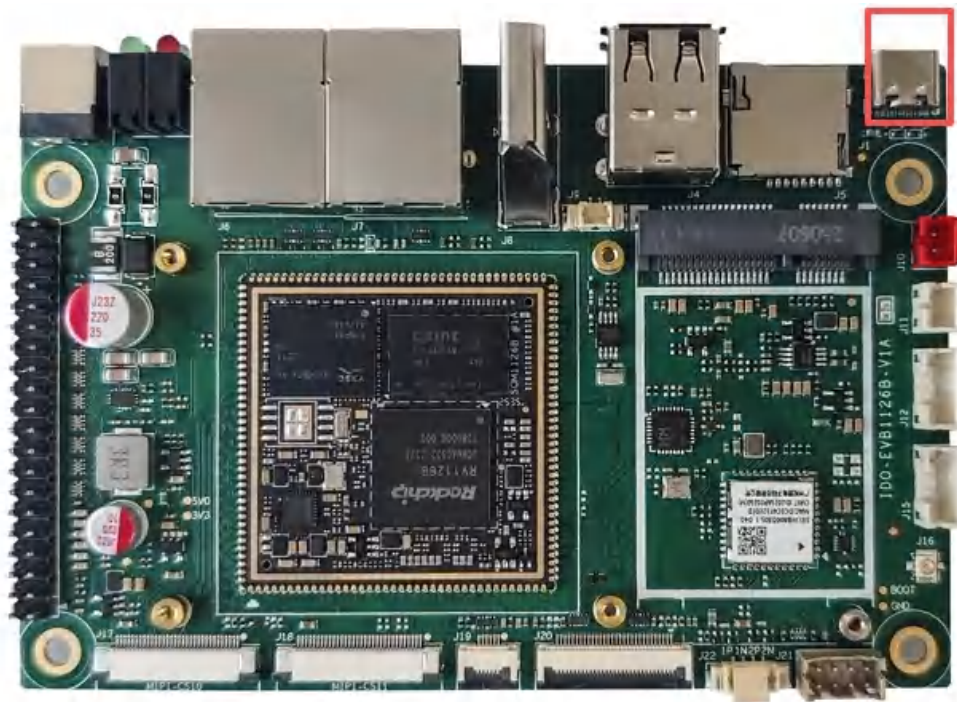
13	扬声器	PH2.0-2P(4Ω5W) ×1
14	MIC	驻极体麦克风 ×1
15	Camera	IMX415 + IMX415
16	UART	TTL x 4 RS485 ×1 （转接板）
17	调试串口	TTL ×1
18	CAN口	CAN ×2
19	RTC	HYM8563 ×1
20	LED	系统指示灯 ×1
21	4G/5G	EC20 4G
22	按键	POWER ×1 REC ×1
23	Fan	5V PH2.0-2P ×1
24	GPIO	GPIO（预留） ×8
25	ADC	ADC（预留） ×4

2. IDO-EVB1126B-V1A-Buildroot系统

2.1. 功能及接口使用方法

2.2. 调试串口

调试串口位置J1，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。



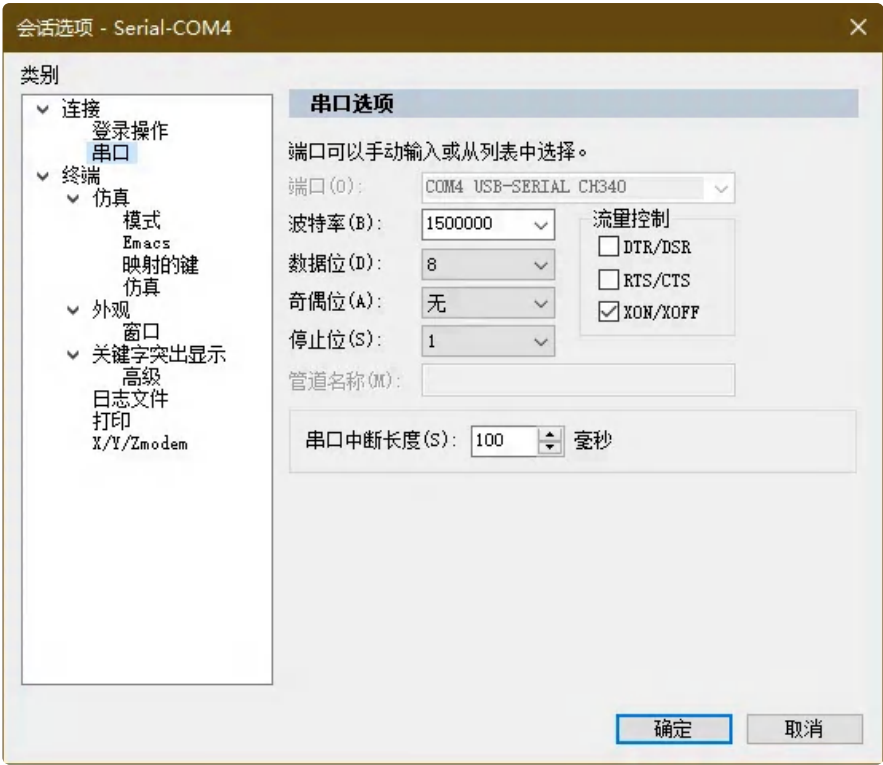
使用调试软件（MobaXterm、putty）等，设置参数如下：

波特率：1500000

数据位：8

奇偶校验位：无

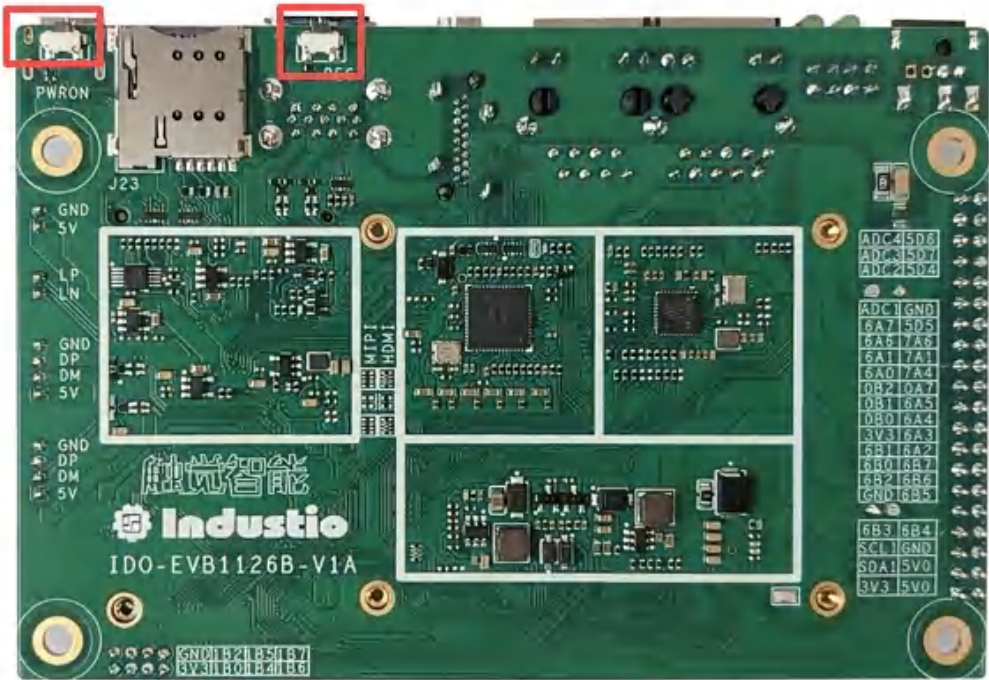
停止位：1



2.3. 按键

主板自带1个Recovery按键REC

Recovery按键，关机状态下，按住Recovery按键再给主板上电，主板会进入Loader模式。



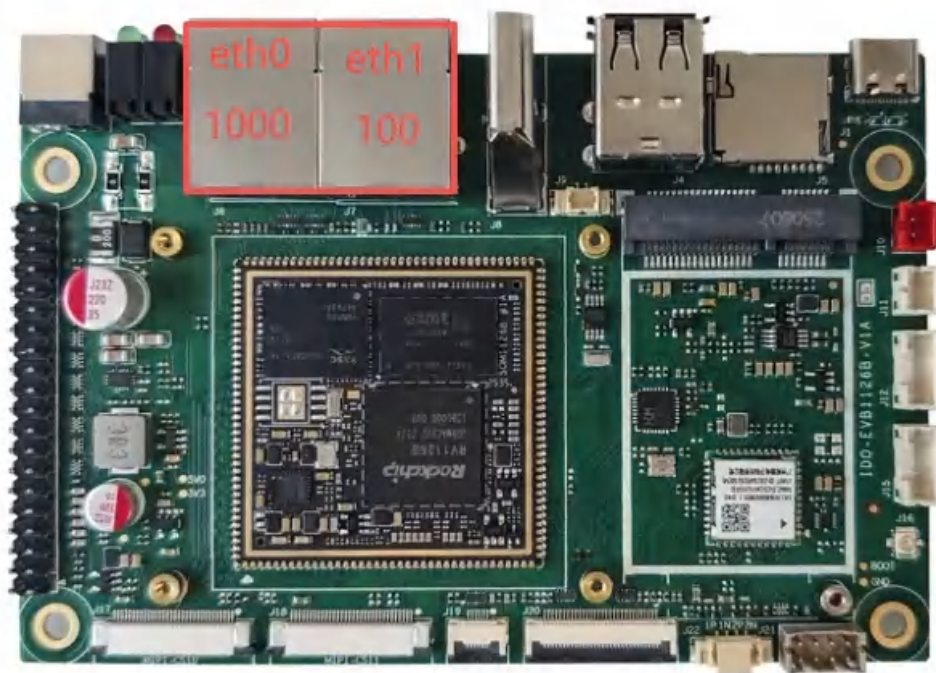
序号	定义	说明
1	PWRON_KEY	电源按键，用于开机/关机；

2	RECOVERY_KEY	烧录按键，用于烧录
---	--------------	-----------

2.4. 网络

2.4.1. Ethernet

主板有 1 路千兆以太网，1路百兆以太网，如下图所示：



```
Bash
1  # TCP
2  #server
3  iperf3 -s -i 1
4  #client
5  iperf3 -c server.ip  # 上行
6  iperf3 -c server.ip -R 下行
7
8  # UDP
9  #server
10 iperf3 -s -i 1
11 #client
12 iperf3 -c server.ip -u -b 1000M  # 上行
13 iperf3 -c server.ip -u -b 1000M -R # 下行
```

可通过ifconfig指令设置临时静态IP，命令如下：

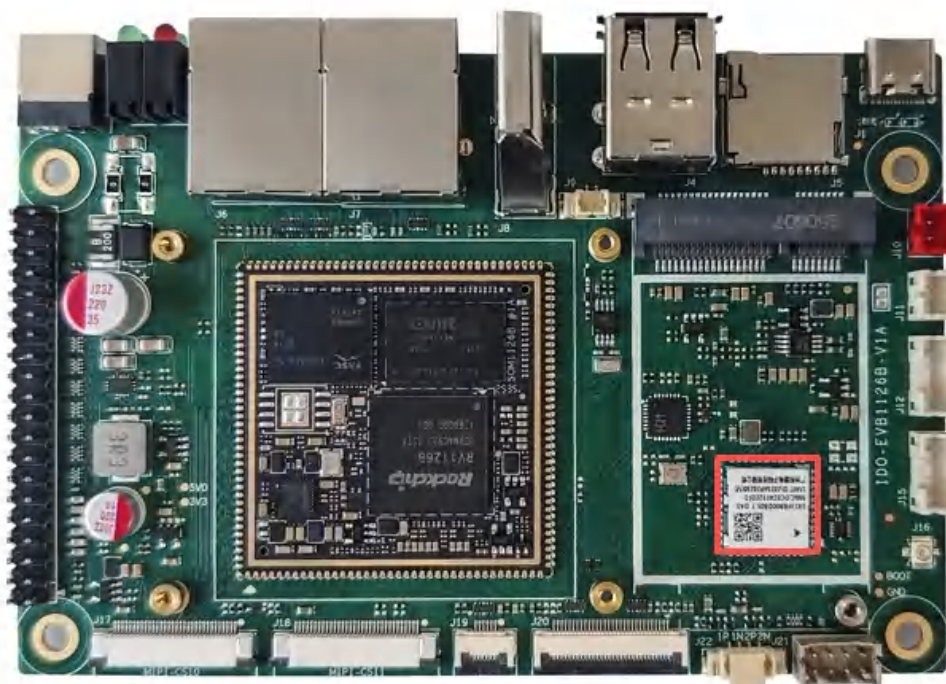
```
Bash
1  $ ifconfig eth0 192.168.1.120
```

如果永久设置编辑文件 /etc/network/interfaces

```
1 root@rv1126b-buildroot:/# vi /etc/network/interfaces
2 root@rv1126b-buildroot:/# cat /etc/network/interfaces
3 # interface file auto-generated by buildroot
4
5 auto lo
6 iface lo inet loopback
7
8 auto eth0
9 iface eth0 inet static
10 address 192.168.3.123
11 netmask 255.255.255.0
12 gateway 192.168.3.1
13
14 source-directory /etc/network/interfaces.d
```

2.4.2. 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J62，如下图所示：



```
1 root@rv1126b-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 }
```

注意:

ssid: 账号
psk: 密码

扫描附近WiFi:

▼

Bash

```
1 iw dev wlan0 scan | grep SSID
```

WiFi连接:

▼

Bash

```
1 wpa_supplicant -Dnl80211 -c /etc/wpa_supplicant.conf -i wlan0 &
```

查看连接结果:

▼

Bash

```
1 wpa_cli -i wlan0 -p /var/run/wpa_supplicant scan
```

2.4.3. Bluetooth

蓝牙节点:

▼

Bash

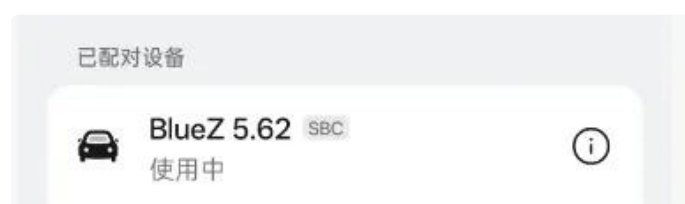
```
1 hciconfig -a
2 hci0: Type: Primary Bus: UART
3 BD Address: 4C:38:00:2B:C1:71 ACL MTU: 1021:8 SCO MTU: 64:1
4 UP RUNNING
5 RX bytes:878 acl:0 sco:0 events:62 errors:0
6 TX bytes:3291 acl:0 sco:0 commands:62 errors:0
7 Features: 0xbf 0xfe 0xcf 0xfe 0xdb 0xff 0x7b 0x87
8 Packet type: DM1 DM3 DM5 DH1 DH3 DH5 HV1 HV2 HV3
9 Link policy: RSWITCH SNIFF
10 Link mode: PERIPHERAL ACCEPT
11 Name: 'BlueZ 5.77'
12 Class: 0x6c0414
13 Service Classes: Rendering, Capturing, Audio, Telephony
14 Device Class: Audio/Video, Loudspeaker
15 HCI Version: 5.0 (0x9) Revision: 0x50
16 LMP Version: 5.0 (0x9) Subversion: 0x6606
17 Manufacturer: Broadcom Corporation (15)
```

2.4.3.1. 命令行连接

```
▼ Bash
1
2  bluetoothctl
3  hci0 new_settings: powered bondable ssp br/edr le secure-conn
4  Agent registered
5  [CHG] Controller 4C:38:00:2B:C1:71 Pairable: yes
6  [bluetooth]#
7  #打开蓝牙
8  [bluetooth]# power on
9  #扫描蓝牙
10 [bluetooth]# scan on
11 # # 开启广播
12 [bluetooth]# discoverable on
13 # 允许配对
14 [bluetooth]# pairable on
15 # 使用 NoInputNoOutput (Just Works)
16 [bluetooth]# agent NoInputNoOutput
17 [bluetooth]# default-agent
18 #查看蓝牙设备
19 [bluetooth]# devices
20 #信任蓝牙设备
21 [bluetooth]# trust 7C:C1:80:09:DD:6C
22 #蓝牙配对
23 [bluetooth]# pair 7C:C1:80:09:DD:6C
24 #连接蓝牙
25 [bluetooth]# connect 7C:C1:80:09:DD:6C
```

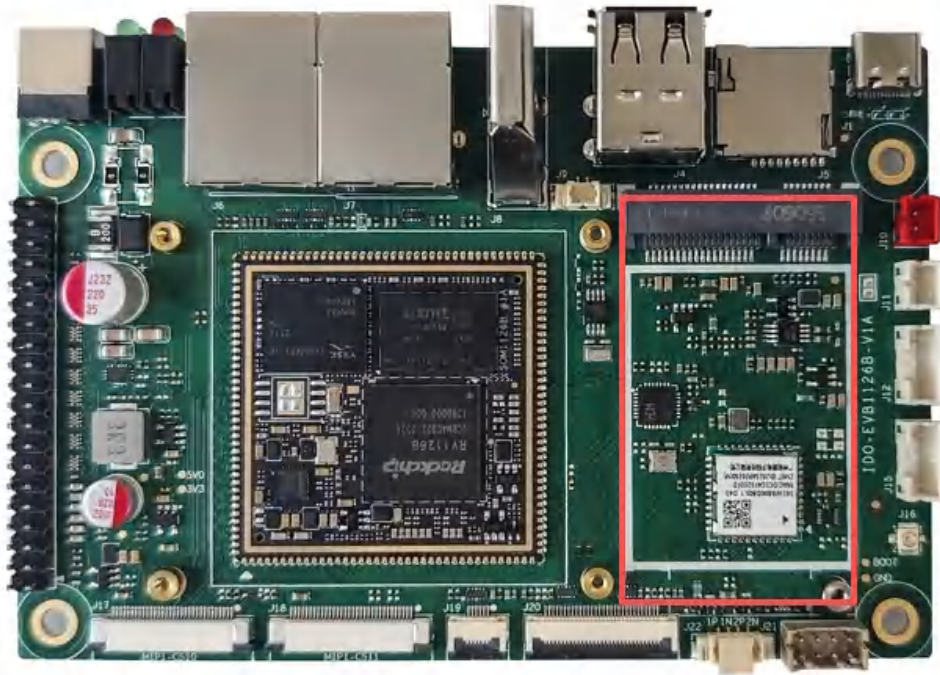
注意：如果出现无法配对的情况，请执行quit退出bluetoothctl，重新启动蓝牙服务，再次执行以上连接命令

```
▼ Bash
1 [bluetooth]# quit
2 root@rv1126b-buildroot:/# /etc/init.d/S40bluetooth restart
```

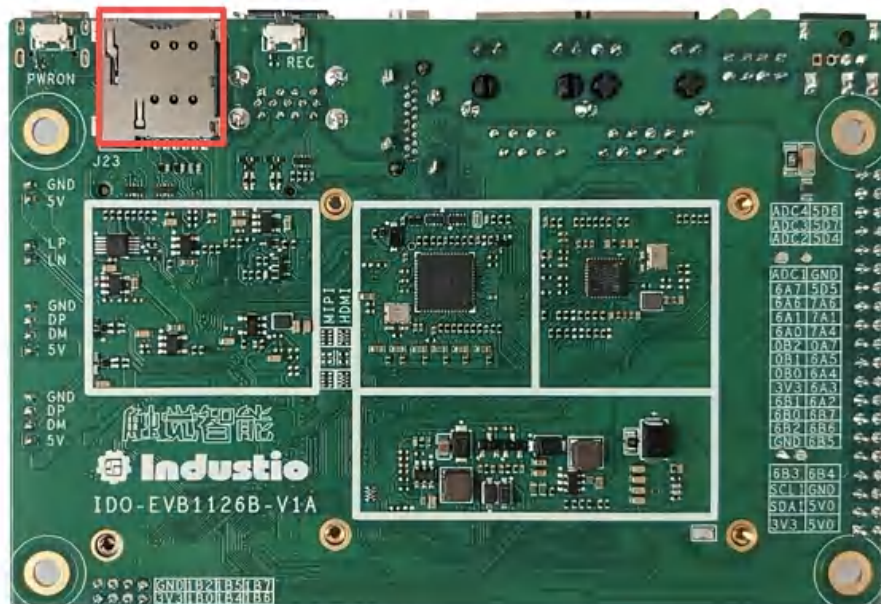


2.4.4. 4G/5G

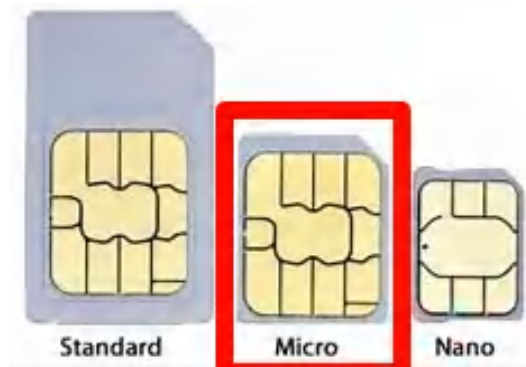
主板内置Mini PCIe 座子扩展 4G/5G模块，4G通信模块适配移远EC20、EC25等通用模组。如下图所示：



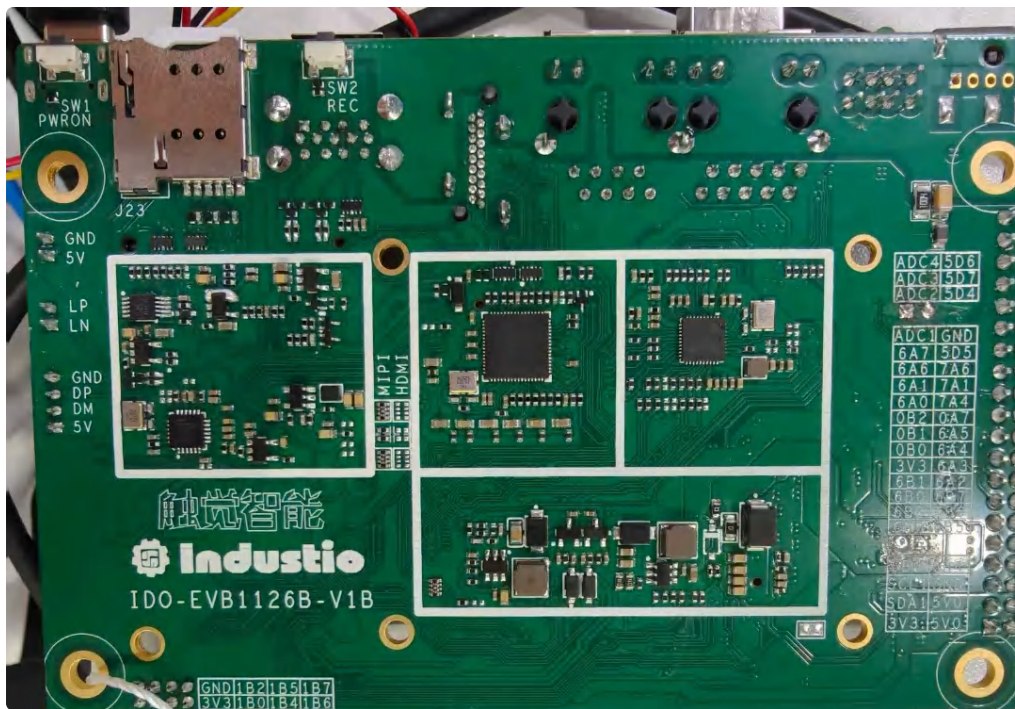
模块需要接上天线以确保获得更好的信号质量，SIM卡安装位J23位于主板背面，如下图所示：



使用Micro尺寸SIM卡，如下图所示：



主板朝上时，SIM卡触点朝上，缺口朝外安装，如下图所示：



拨号工具使用quectel-CM

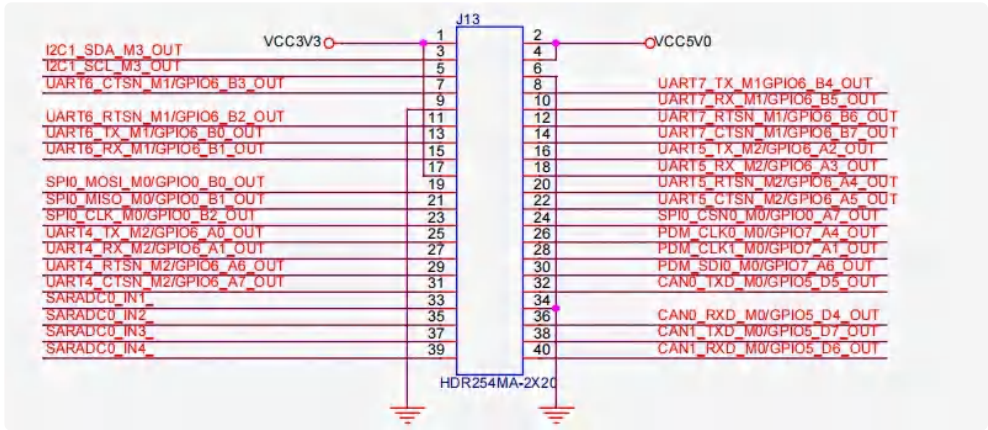
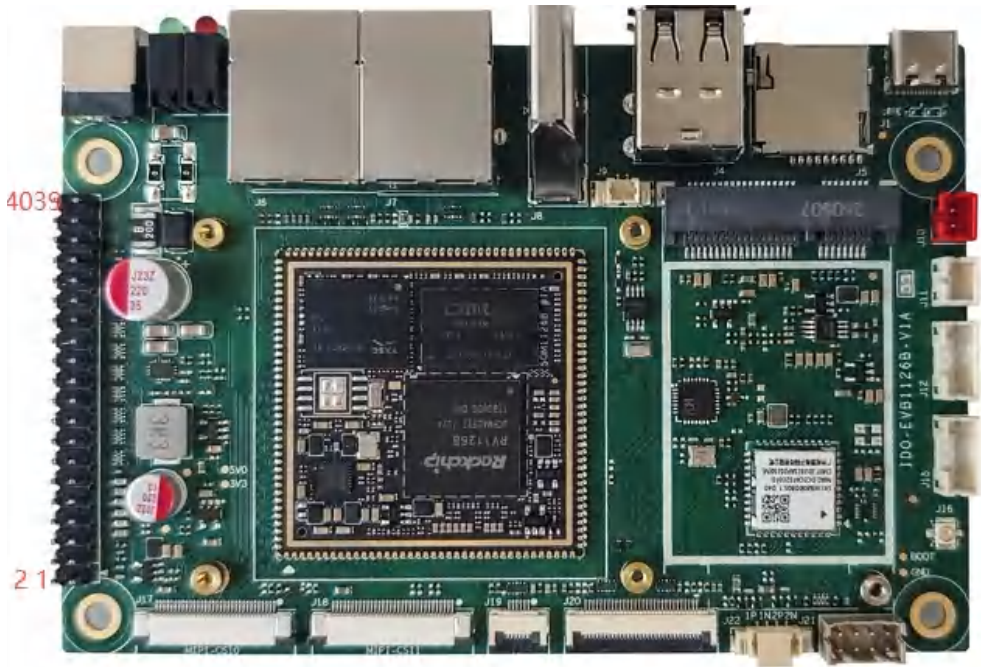
```

1 root@rv1126b-buildroot:/# quectel-CM &
2 wwan0      Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
3
4          inet addr:10.230.31.108  P-t-P:10.230.31.108  Mask:255.255.255.248
5          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
6          RX packets:5 errors:0 dropped:0 overruns:0 frame:0
7          TX packets:29 errors:0 dropped:0 overruns:0 carrier:0
8          collisions:0 txqueuelen:1000
9          RX bytes:1420 (1.3 KiB)  TX bytes:5702 (5.5 KiB)
10
11 root@rv1126b-buildroot:/# ping www.baidu.com -I wwan0
12 PING www.baidu.com (157.148.69.186) from 10.230.31.108 wwan0: 56(84) bytes of data:
13 64 bytes from 157.148.69.186: icmp_seq=1 ttl=52 time=29.2 ms
14 64 bytes from 157.148.69.186: icmp_seq=2 ttl=52 time=37.1 ms
15 64 bytes from 157.148.69.186: icmp_seq=3 ttl=52 time=35.4 ms
16 64 bytes from 157.148.69.186: icmp_seq=4 ttl=52 time=33.6 ms
17 64 bytes from 157.148.69.186: icmp_seq=5 ttl=52 time=31.7 ms
18 64 bytes from 157.148.69.186: icmp_seq=6 ttl=52 time=29.6 ms
19 64 bytes from 157.148.69.186: icmp_seq=7 ttl=52 time=37.7 ms
20 64 bytes from 157.148.69.186: icmp_seq=8 ttl=52 time=36.3 ms
21 ^C
22 --- www.baidu.com ping statistics ---
23 8 packets transmitted, 8 received, 0% packet loss, time 26063ms
    rtt min/avg/max/mdev = 29.170/33.826/37.680/3.133 ms

```

2.5. UART

IDO-EVB1126B-V1主板一共引出4路UART串口（不含调试串口），如下图所示：



UART串口设备节点列表如下：

序号	默认电平类型	设备节点
J13-25,J13-27	TTL	/dev/ttyS4
J3-16,J3-18,J3-20	RS485	/dev/ttyS5
J3-13,J3-15	TTL	/dev/ttyS6
J3-8,J3-10	TTL	/dev/ttyS7

使用microcom进行串口测试：

▼

Bash

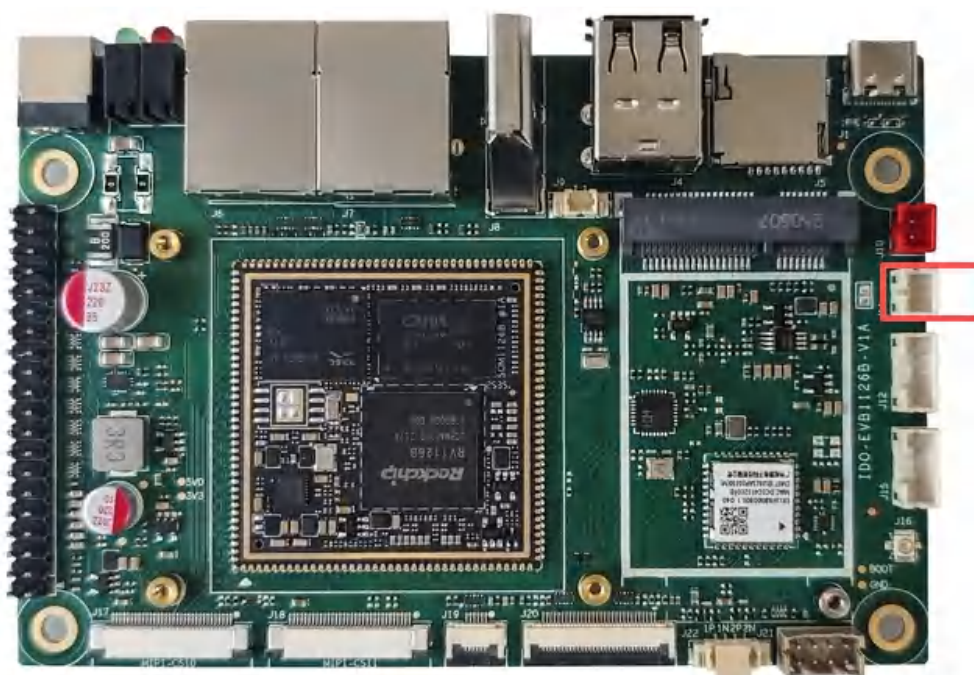
1 root@rv1126b-buildroot:/# microcom /dev/ttyS7 -s 115200

注意：执行 Ctrl x 退出程序； RS485 需要使用转接板测试

2.6. Audio

2.6.1. Speaker

喇叭接口为J11，如下图所示：



声卡节点：

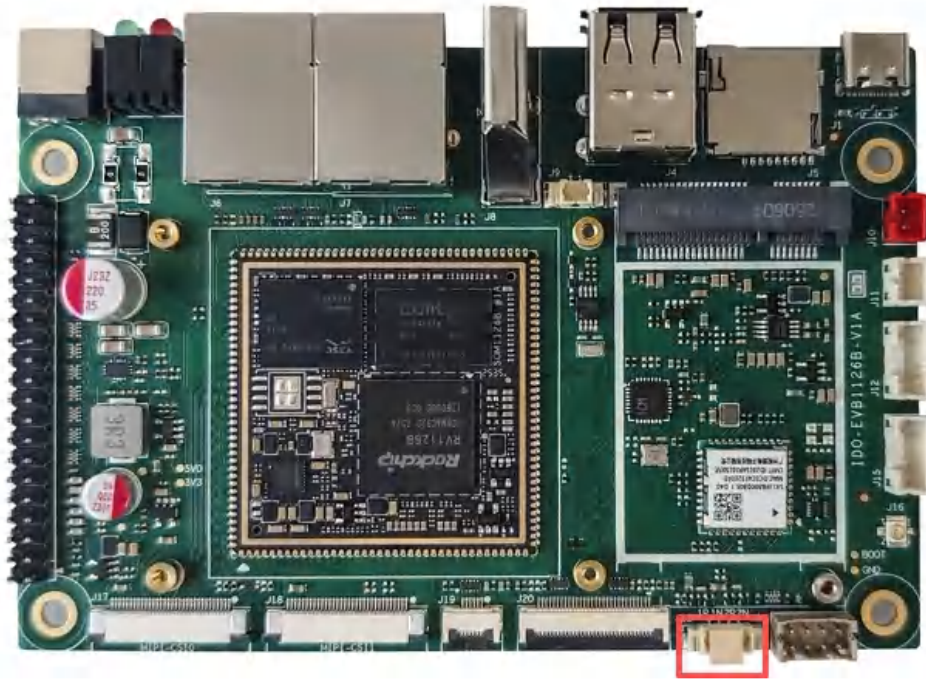
```
▼ Bash
1 root@rv1126b-buildroot:/# aplay -l
2 **** List of PLAYBACK Hardware Devices ****
3 card 0: rockchiprv1126b [rockchip,rv1126b-acodec], device 0: dailink-multicodecs m
  ulticodec-0 [dailink-multicodecs multicodec-0]
4   Subdevices: 1/1
5   Subdevice #0: subdevice #0
```

音频播放：

```
▼ Bash
1 #音频播放到声卡
2 root@rv1126b-buildroot:/# aplay -D hw:0,0 test.wav
```

2.6.2. MIC

MX1.25T-2P麦克风接口J52，如下图所示：



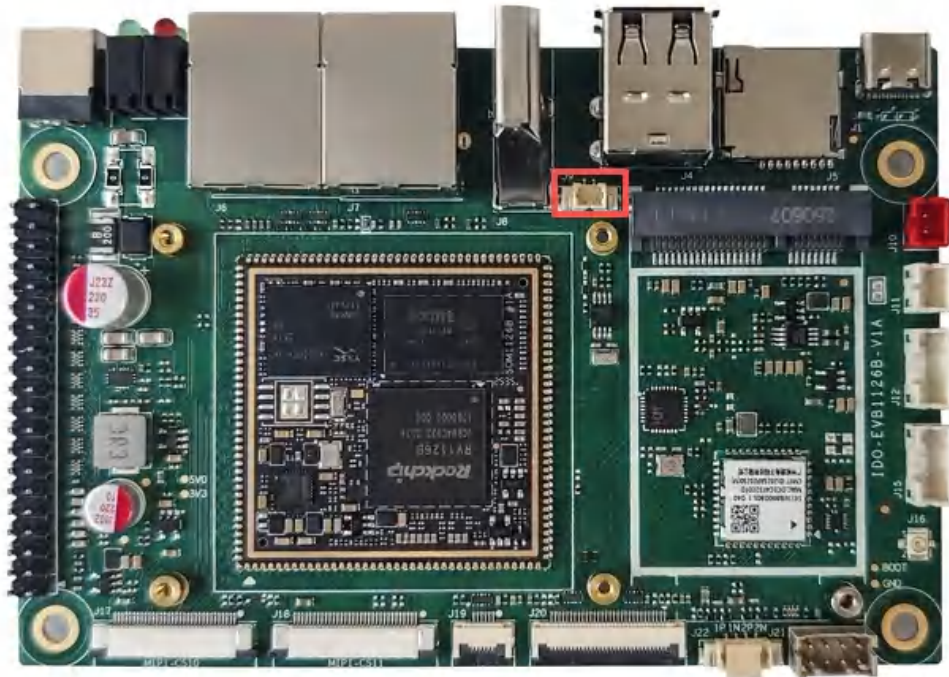
注意：录出来音频文件需要导出到PC 上播放

测试命令：

```
▼ Bash |
1 # 一共两路mic , mic0 录音命令
2 arecord -D hw:0,0 -f S16_LE -r 16000 -c 4 -t wav test_mic0.wav
3 # mic1 录音命令
4 arecord -D hw:0,0 -f S16_LE -r 16000 -c 4 -t wav test_mic1.wav
```

2.7. RTC

MX1.25T-2P RTC电池座J22，如下图所示：



连接3V 纽扣电池，RTC电池参考如下



系统默认使用HYM8563作为系统时钟，设备节点：/dev/rtc0，时间设置方法：

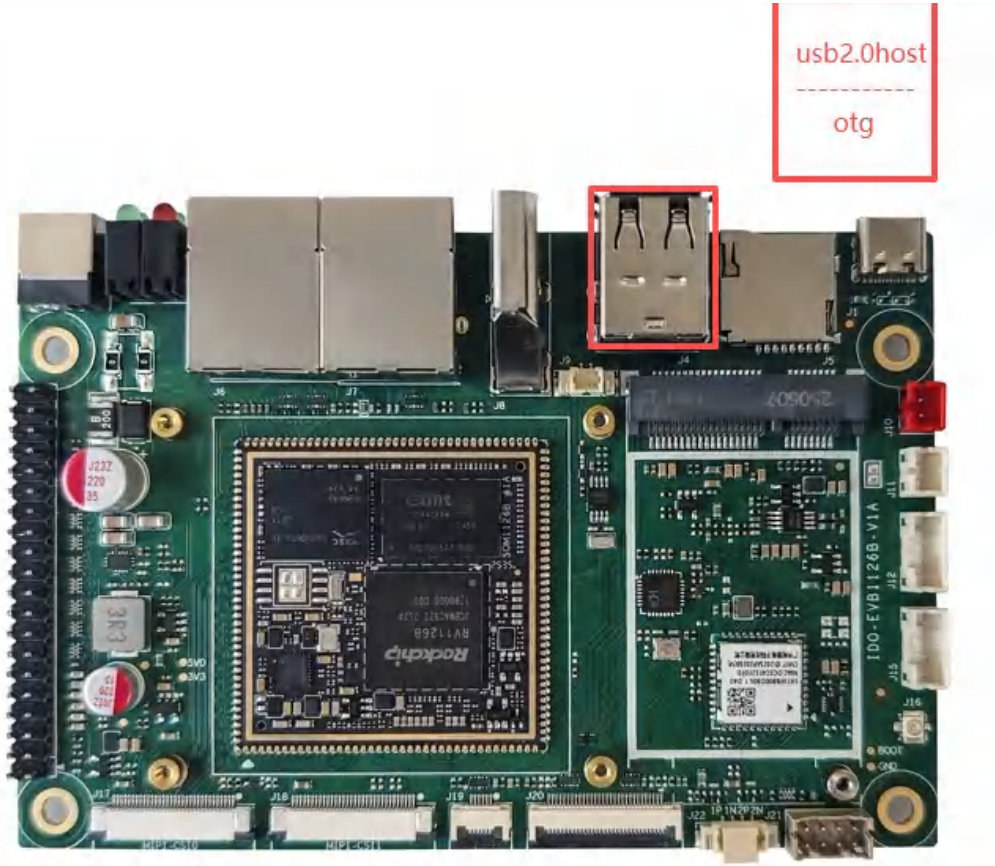
```
▼ Bash
1 #
2 root@rv1126b-buildroot:/# hwclock -w      # 向rtc写时间
3 root@rv1126b-buildroot:/# hwclock -r      # 向rtc读时间
```

2.8. USB

主板支持 1 个USB3-OTG，1 个 USB Type-A ,1 个USB2.0PH2.0-4P 接口，USB对外总供电应小于4A。

2.8.1. OTG

TypeA接口（USB3.0 OTG）J4，如下图所示：



EVB1126B 已实现上电启动自动切换OTG

- 1. 上电启动前接入OTG 通过双公头usb连接 PC ，那么otg为device模式
- 2. 上电启动前接入U盘等usb设备，启动进入系统后otg为host模式

系统下otg切换命令：

命令	模式
gpio-otg HOST	HOST 模式
gpio-otg DEVICE	DEVICE 模式

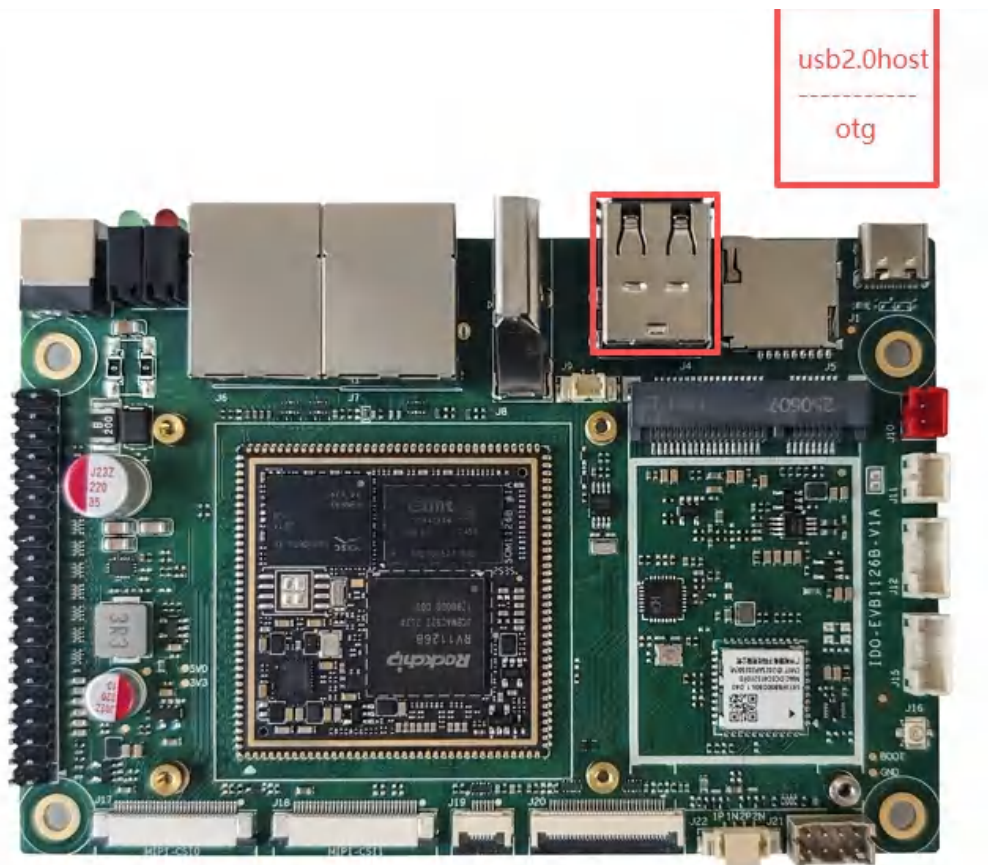
▼

Bash

```
1 # host模式
2 root@rv1126b-buildroot:/# gpio-otg HOST
3 # device模式
4 root@rv1126b-buildroot:/# gpio-otg DEVICE
```

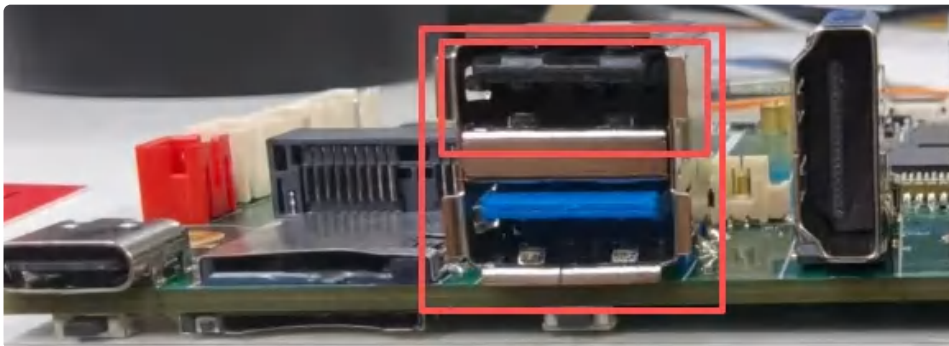
2.8.2. USB3.0

RV1126B 只有一个OTG3.0,



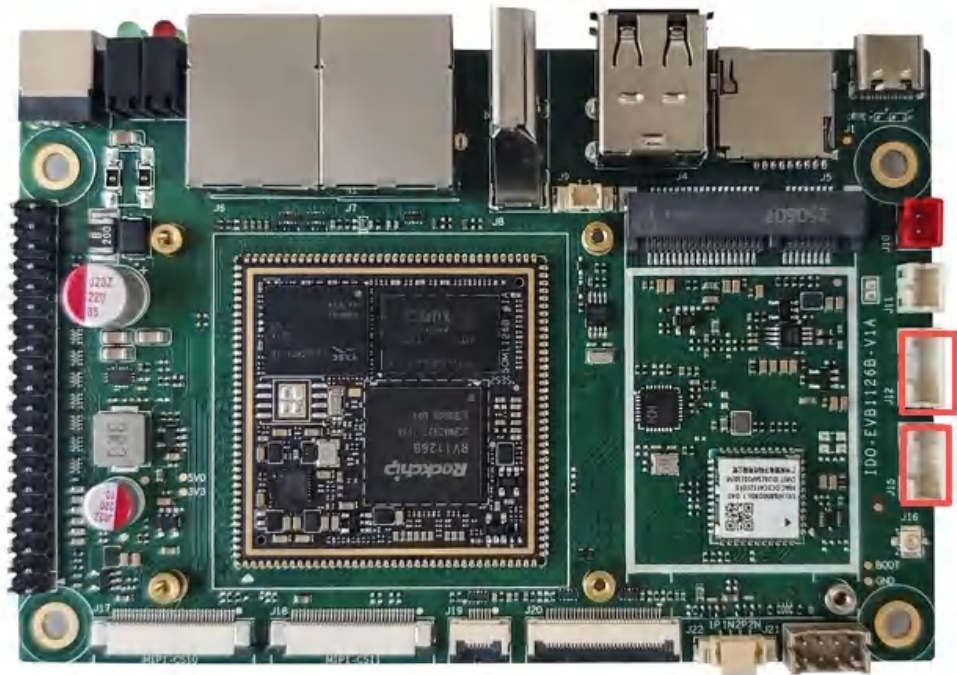
2.8.3. USB2.0-A

主板配置了1路USB2.0-A接口，如下图所示：



2.8.4. USB2.0-PH

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



	J12
控制节点	/sys/class/leds/usb_vbus3_en/brightness



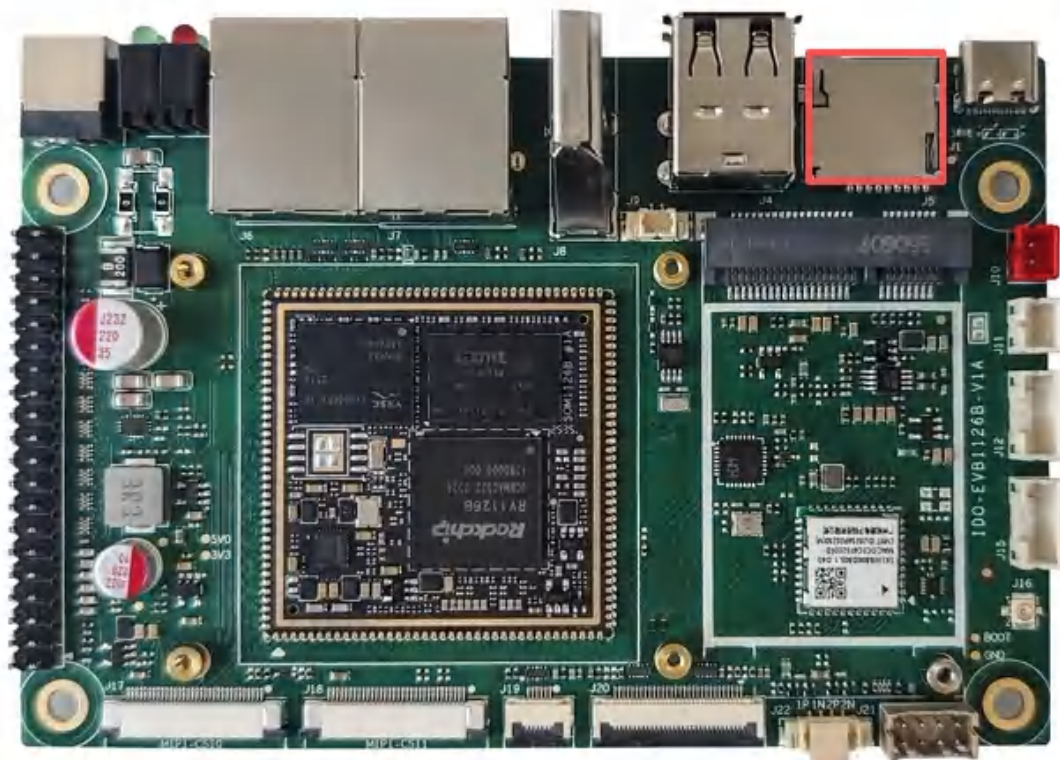
▼

Bash

```
1 echo 1 > /sys/class/leds/usb_vbus3_en/brightness # 打开
2 echo 0 > /sys/class/leds/usb_vbus3_en/brightness # 关闭
```

2.9. TF Card

TF卡座支持SDIO3.0，支持高速SD卡，接口位于主板背面J43，如下图所示：



Plain Text

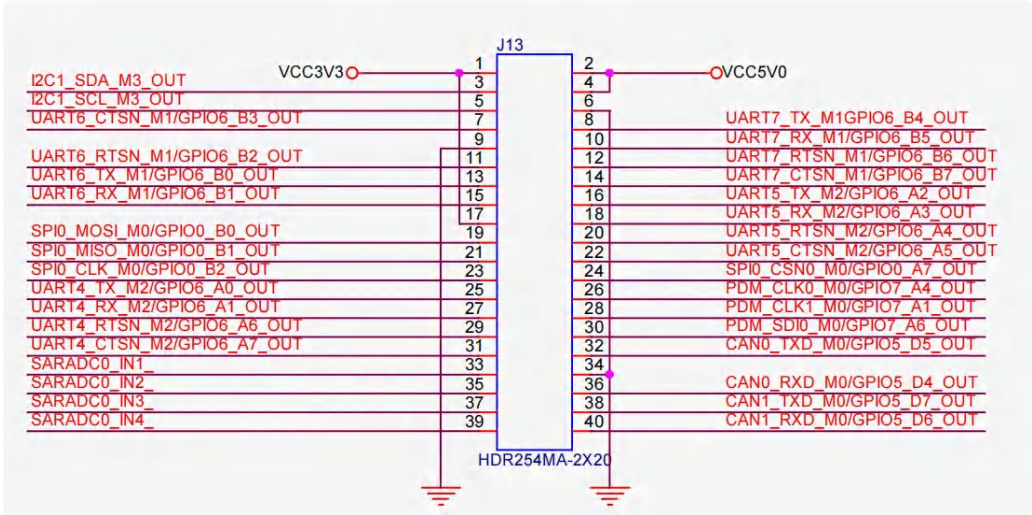
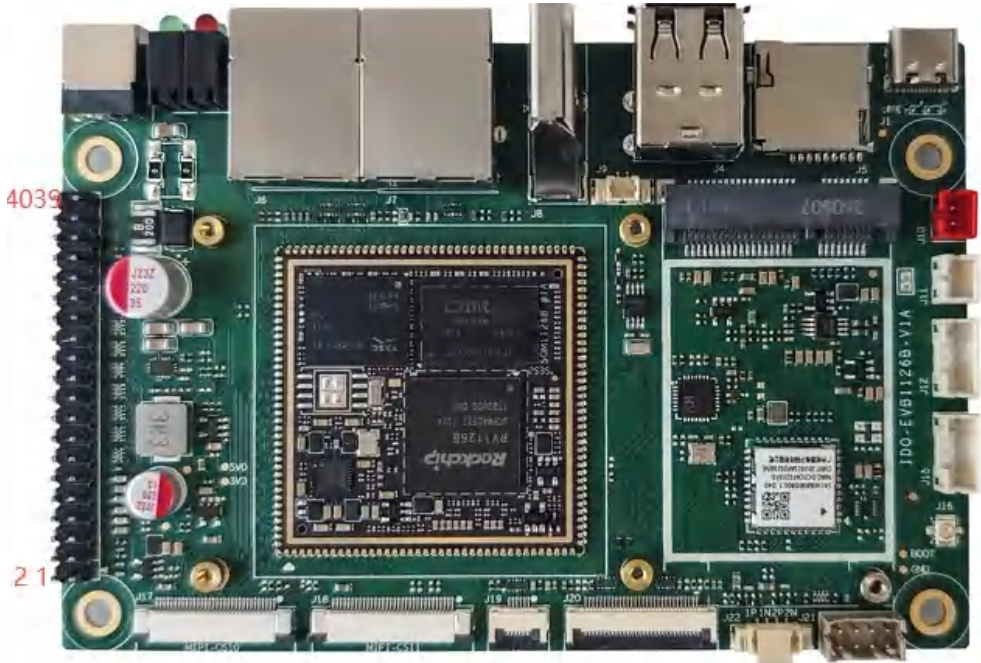
```

1 root@rv1126b-buildroot:/# fdisk -l
2 Found valid GPT with protective MBR; using GPT
3 ...
4 Device      Boot StartCHS      EndCHS          StartLBA        EndLBA        Sectors  Size
   Id Type
5 /dev/mmcblk1p1  0,2,10      239,254,63          135      3858623      3858489 1884M
   e Win95 FAT16 (LBA)
6
7 root@rv1126b-buildroot:/# df -h
8 Filesystem      Size  Used Avail Use% Mounted on
9 /dev/root        56G   994M   52G   2% /
10 devtmpfs        1.9G   8.0K   1.9G   1% /dev
11 tmpfs           2.0G   140K   2.0G   1% /tmp
12 tmpfs           2.0G   468K   2.0G   1% /run
13 tmpfs           2.0G   244K   2.0G   1% /var/log
14 tmpfs           2.0G     0   2.0G   0% /dev/shm
15 /dev/mmcblk0p7  124M   12M  109M  10% /oem
16 /dev/mmcblk0p6  2.0G   616K   2.0G   1% /userdata
17 /dev/mmcblk1p1  32G    2G   30G    1% /mnt/sdcard

```

2.10. CANFD

主板共有2路CAN接口如下图所示:



CAN	接线
can0	J13-32 ,J13-34
can1	J13-38 ,J13-40



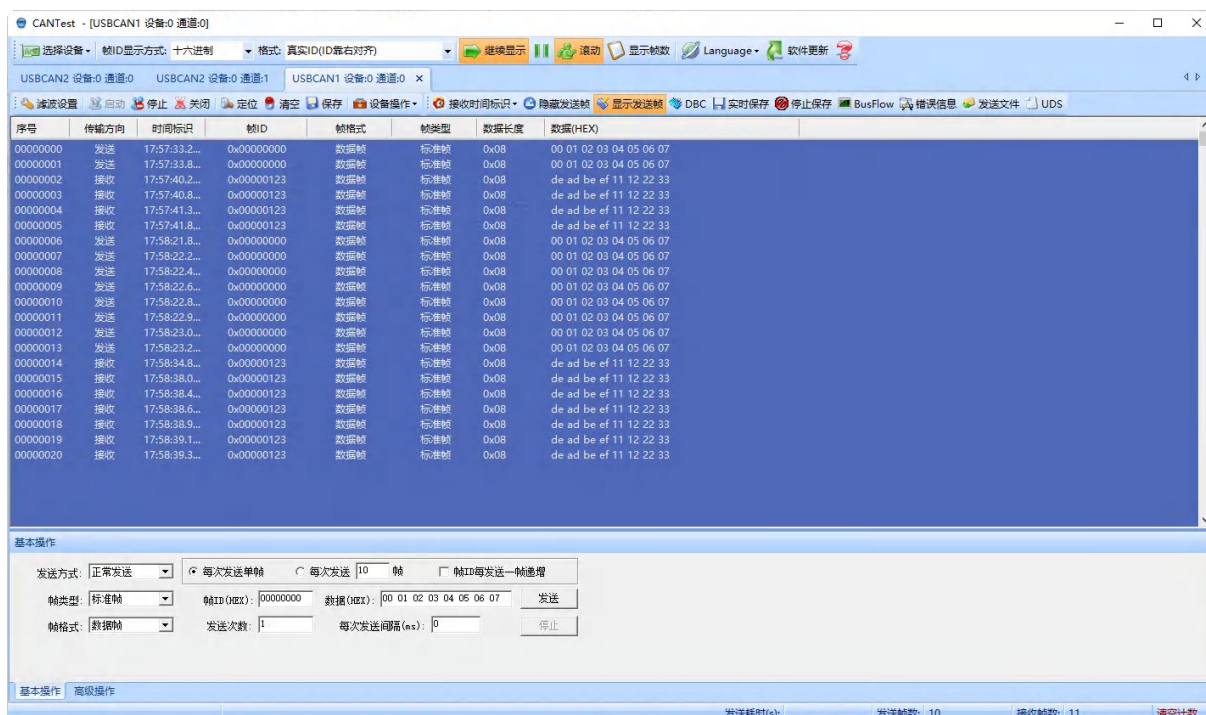
分别对应系统节点名称为 can0、can1。使用can盒测试：

1. 设置CAN参数

```

1  # can0节点发送数据
2  ip link set can0 down
3  ip link set can0 type can bitrate 1000000 dbitrte 1000000 fd on
4  ip link set can0 up
5  cansend can0 123#DEADBEEF
6  cansend can0 123#DEADBEEF11122233
7
8  # can0接收数据
9  ip link set can0 down
10 ip link set can0 type can bitrate 1000000 dbitrte 1000000 fd on
11 ip link set can0 up
12 candump can1
13
14 # can1 接收数据
15 ip link set can1 down
16 ip link set can1 type can bitrate 1000000 dbitrte 1000000 fd on
17 ip link set can1 up

```



```

root@rv1126b-buildroot:/# candump can0
can0 000 [8] 00 01 02 03 04 05 06 07
can0 000 [8] 00 01 02 03 04 05 06 07
can0 000 [8] 00 01 02 03 04 05 06 07
^Croot@rv1126b-buildroot:/# ^C
root@rv1126b-buildroot:/# ^C
root@rv1126b-buildroot:/# ^C
root@rv1126b-buildroot:/# cansend can0 123#DEADBEEF11122233
root@rv1126b-buildroot:/# cansend can0 123#DEADBEEF11122233
root@rv1126b-buildroot:/# cansend can0 123#DEADBEEF11122233
root@rv1126b-buildroot:/# cansend can0 123#DEADBEEF11122233

```

```
root@rv1126b-buildroot:/# candump can1
can1 000 [8] 00 01 02 03 04 05 06 07
can1 000 [8] 00 01 02 03 04 05 06 07
can1 000 [8] 00 01 02 03 04 05 06 07
can1 000 [8] 00 01 02 03 04 05 06 07
can1 000 [8] 00 01 02 03 04 05 06 07
can1 000 [8] 00 01 02 03 04 05 06 07
can1 000 [8] 00 01 02 03 04 05 06 07
can1 000 [8] 00 01 02 03 04 05 06 07
^Croot@rv1126b-buildroot:/# ^C
root@rv1126b-buildroot:/# ^C
root@rv1126b-buildroot:/# ^C
root@rv1126b-buildroot:/# cansend can0 123#DEADBEEF11122233
root@rv1126b-buildroot:/# cansend can1 123#DEADBEEF11122233
root@rv1126b-buildroot:/# cansend can1 123#DEADBEEF11122233
root@rv1126b-buildroot:/# cansend can1 123#DEADBEEF11122233
root@rv1126b-buildroot:/# cansend can1 123#DEADBEEF11122233
root@rv1126b-buildroot:/# cansend can1 123#DEADBEEF11122233
root@rv1126b-buildroot:/# cansend can1 123#DEADBEEF11122233
root@rv1126b-buildroot:/# cansend can1 123#DEADBEEF11122233
root@rv1126b-buildroot:/# cansend can1 123#DEADBEEF11122233
```

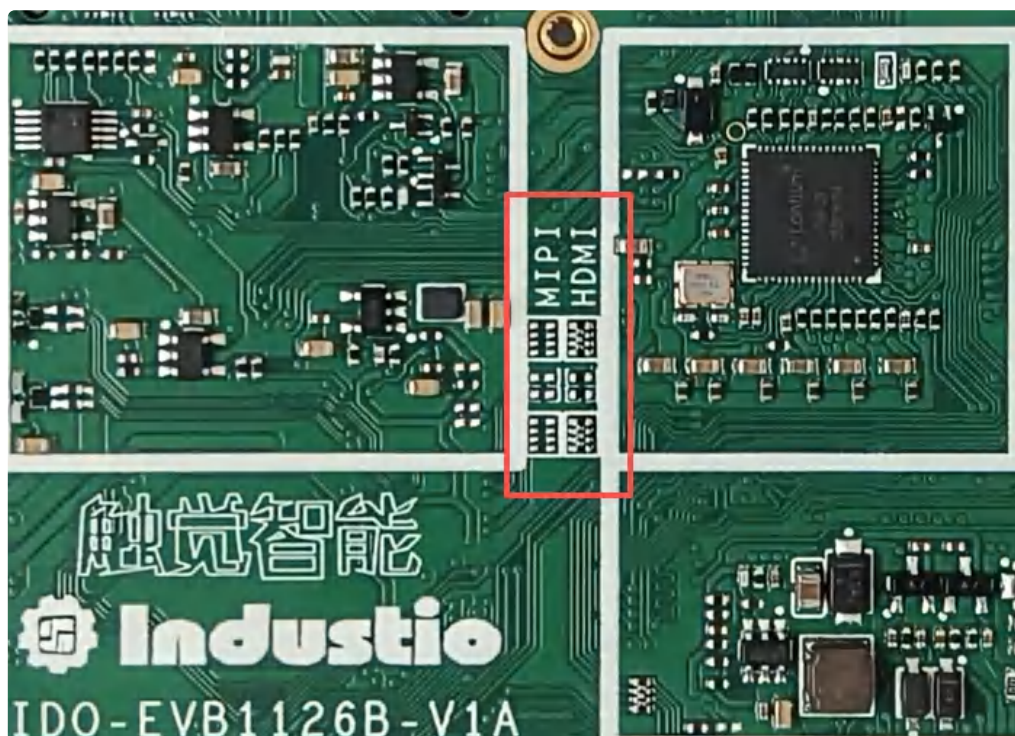
注意： 测试需要使用转接板。

2.11. LCD显示

2.11.1. HDMI-TX

注意： HDMI-TX 是由 MIPI-TX 接口 + 信号转换芯片，二选一，硬件默认配置为 HDMI-TX。

HDMI-TX硬件确认，如下图所示：



2.11.2. MIPI-TX

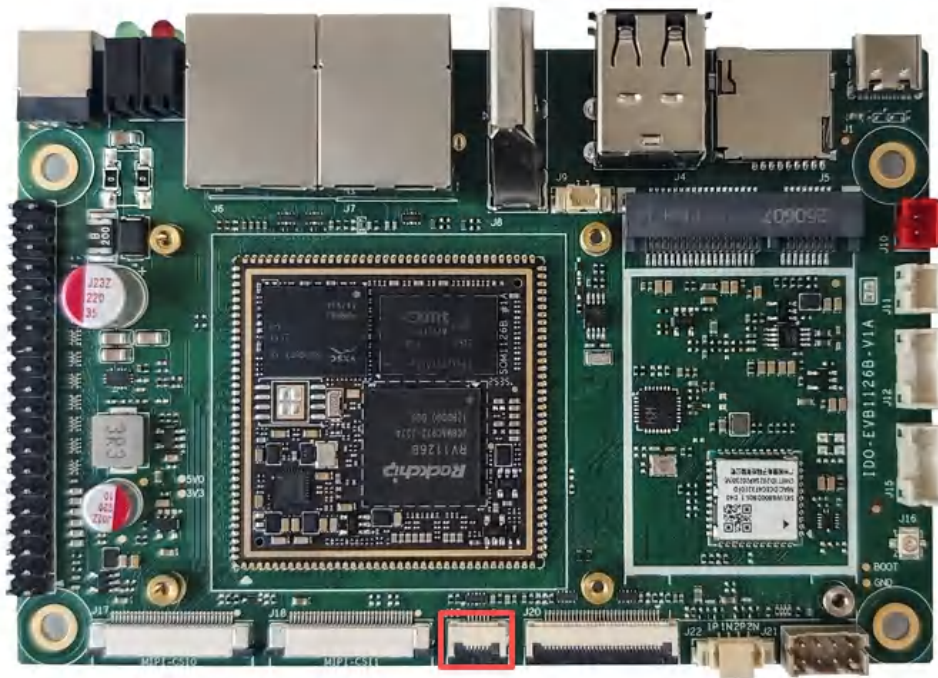
注意： MIPI-TX 与 HDMI 为复用关系，二选一，硬件默认配置为 HDMI。

要使用mipi的话，需要将下图中HDMI 列电阻移动至MIPI 列



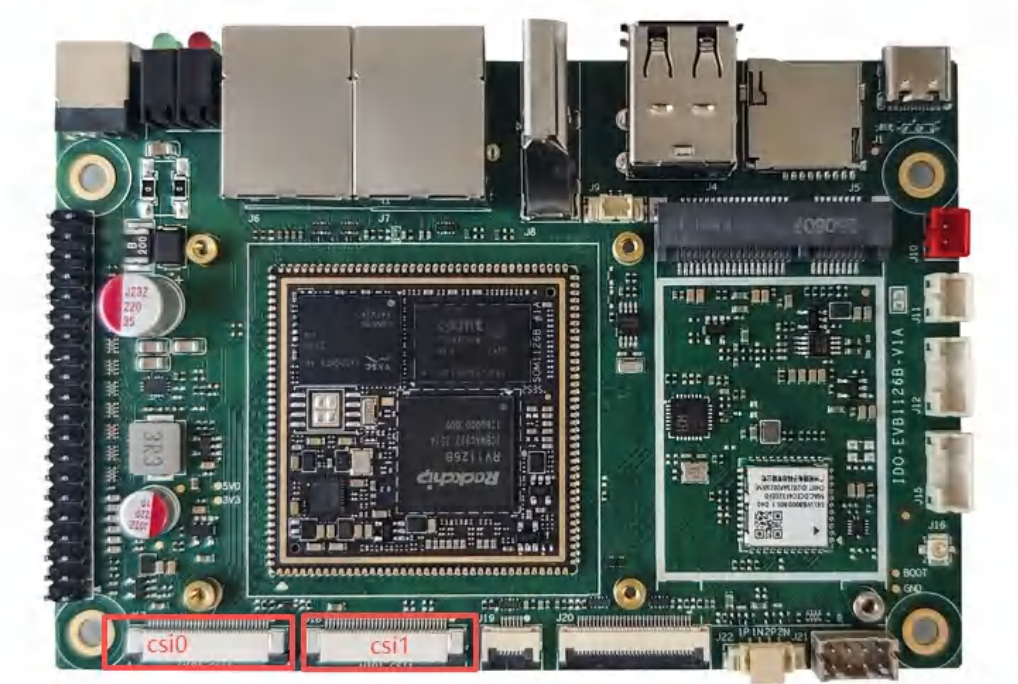
2.12. TP

I2C-TP接口为 6Pin FPC 0.5mm座子 (J13)，如下图所示：



2.13. MIPI Camera

主板内置两路MIPI Camer接口J17和J18，使用FPC05-PUW-32P抽拉式上接座子，如下图所示：



2.13.1. IMX415

J18 默认适配IMX415作为后置摄像头，接法如下图所示：



J17	J18
/dev/video22	/dev/video30

2.13.1.1. 预览

▼

Bash

```
1  gst-launch-1.0 v4l2src device=/dev/video22 ! video/x-raw, format=NV12, width=1920, height=1080, framerate=30/1 ! videoconvert ! autovideosink
2
3  gst-launch-1.0 v4l2src device=/dev/video30 ! video/x-raw, format=NV12, width=1920, height=1080, framerate=30/1 ! videoconvert ! autovideosink
```

2.13.1.2. 抓图

▼

Bash

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

2.13.1.3. 抓视频

▼

Bash

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

2.13.2. TP2815



J17	J18
/dev/video0	/dev/video11
/dev/video1	/dev/video12
/dev/video2	/dev/video13

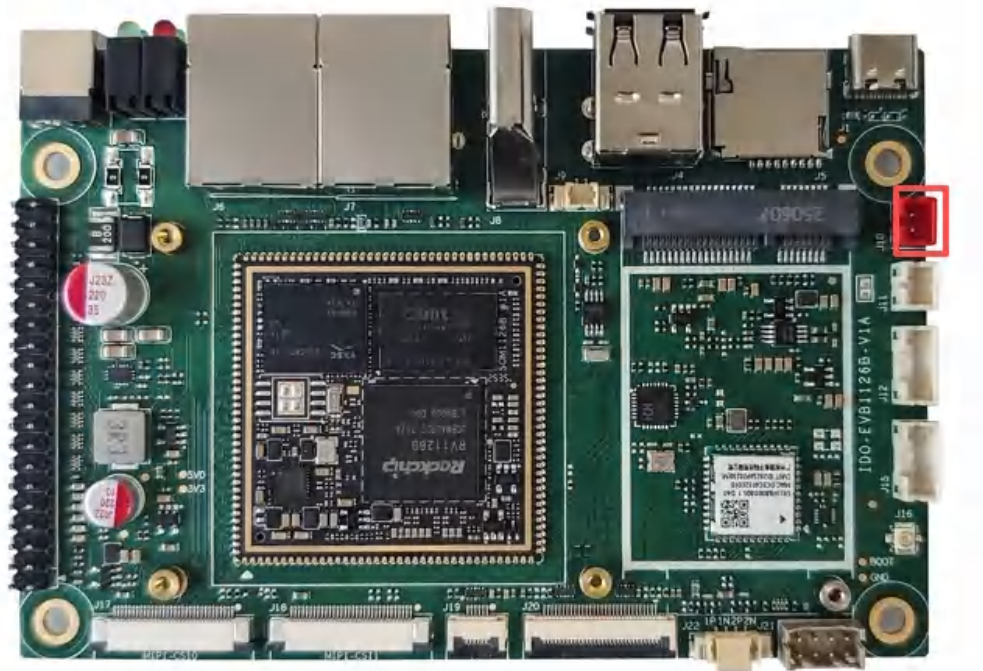
/dev/video3	/dev/video14
-------------	--------------

2.13.2.1. 预览

```
▼ Bash |
1  gst-launch-1.0 v4l2src device=/dev/video0 ! video/x-raw, format=NV12, width=128
   0, height=720, framerate=25/1 ! videoconvert ! autovideosink
2
3
4  gst-launch-1.0 v4l2src device=/dev/video1 ! video/x-raw, format=NV12, width=128
   0, height=720, framerate=25/1 ! videoconvert ! autovideosink
5
6
7      gst-launch-1.0 v4l2src device=/dev/video2 ! video/x-raw, format=NV12, width=1
   280, height=720, framerate=25/1 ! videoconvert ! autovideosink
8
9
10     gst-launch-1.0 v4l2src device=/dev/video3 ! video/x-raw, format=NV12, width=1
   280, height=720, framerate=25/1 ! videoconvert ! autovideosink
11
12
13     gst-launch-1.0 v4l2src device=/dev/video11 ! video/x-raw, format=NV12, width=
   1280, height=720, framerate=25/1 ! videoconvert ! autovideosink
14
15     gst-launch-1.0 v4l2src device=/dev/video12 ! video/x-raw, format=NV12, width=
   1280, height=720, framerate=25/1 ! videoconvert ! autovideosink
16
17
18     gst-launch-1.0 v4l2src device=/dev/video13 ! video/x-raw, format=NV12, width=
   1280, height=720, framerate=25/1 ! videoconvert ! autovideosink
19
20
21     gst-launch-1.0 v4l2src device=/dev/video14 ! video/x-raw, format=NV12, width=
   1280, height=720, framerate=25/1 ! videoconvert ! autovideosink
22
```

2.14. FAN 风扇

PH2 5V FAN风扇接口J3, 如下图所示:

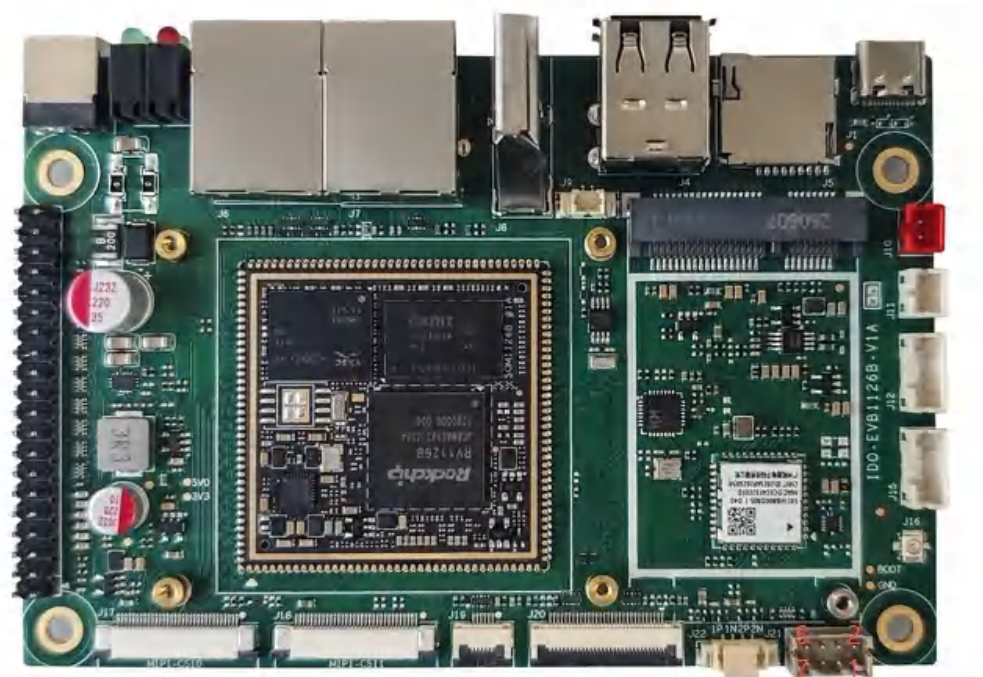


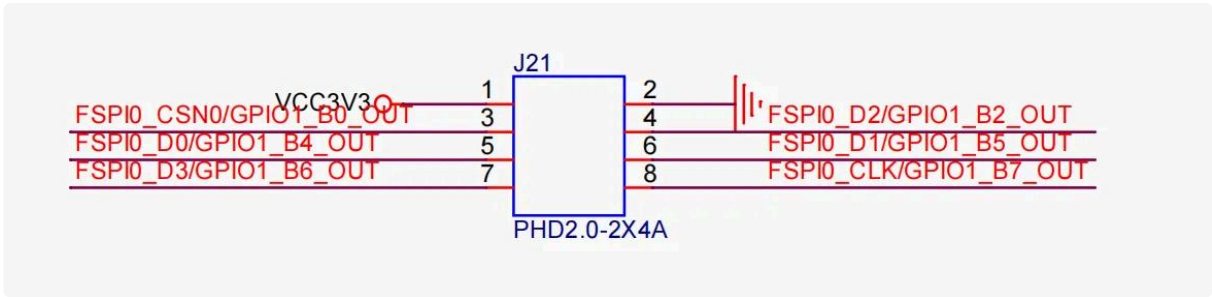
控制功能	控制命令
打开风扇	<code>echo 1 > /sys/class/leds/fan_led/brightness</code>
关闭风扇	<code>echo 0 > /sys/class/leds/fan_led/brightness</code>



2.15. GPIO

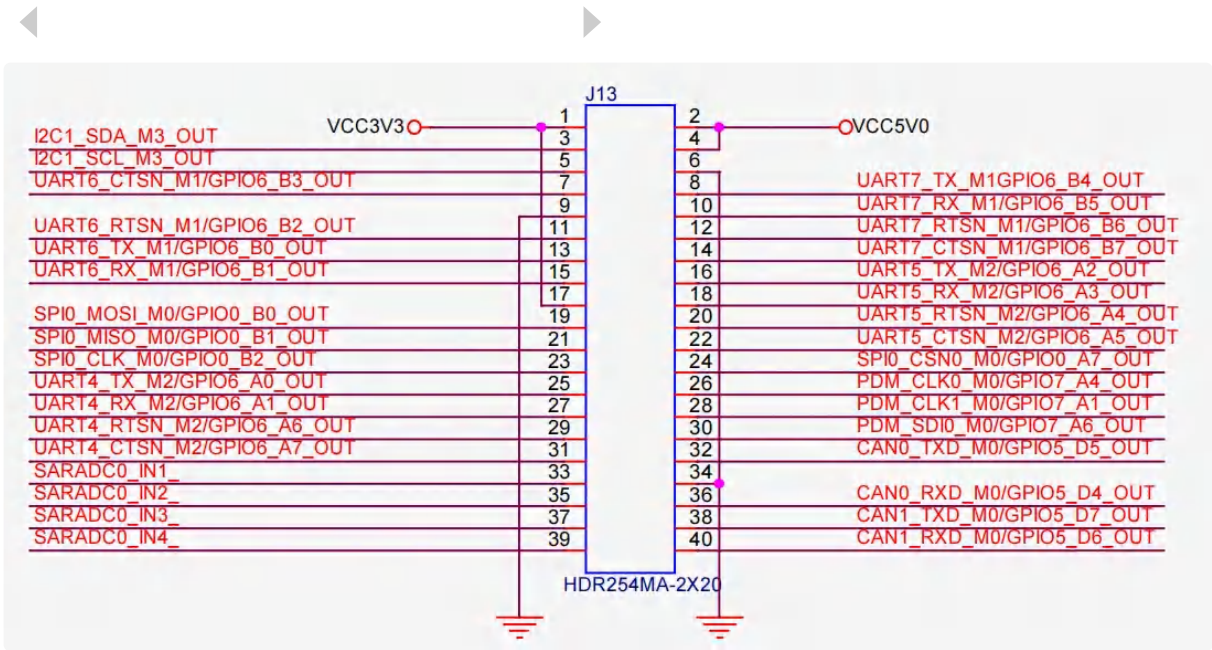
主板预留了8个GPIO接口J3，如下图所示：





GPIO引脚定义描述，如下图所示：

序号	管脚标号	GPIO 序号
1	VCC3V3	/
2	GND	/
3	GPIO1_B0	40
4		42
5	GPIO1_B4	44
6	GPIO1_B5	45
7	GPIO1_B6	46
8	GPIO1_B7	47



序号	管脚标号	GPIO 序号
7	GPIO6_B3	203
11	GPIO6_B2	202
19	GPIO0_B0	8
21		9

23	GPIO0_B2	10
29	GPIO6_A6	198
31	GPIO6_A7	199
12	GPIO6_B6	206
14	GPIO6_B7	207
22	GPIO6_A5	197
24	GPIO0_A7	7
26	GPIO7_A4	228
28	GPIO7_A1	225
30	GPIO7_A6	230

执行以下脚本，使用万用表实际去量电压值，是否是高电平

```

1  #!/bin/bash
2
3  GPIO_NUM=(40 42 44 45 46 47 203 202 8 9 10 198 199 206 207 197 7 228 225 230)
4
5  for gpio in "${GPIO_NUM[@]"; do
6      echo "$gpio" > /sys/class/gpio/export 2>/dev/null
7      if [ $? -eq 0 ]; then
8          echo "GPIO $gpio exported successfully."
9          echo out > /sys/class/gpio/gpio$gpio/direction
10         echo 1 > /sys/class/gpio/gpio$gpio/value
11         echo "GPIO $gpio Out High"
12     else
13         echo "GPIO $gpio already exported or failed to export."
14     fi
15 done

```

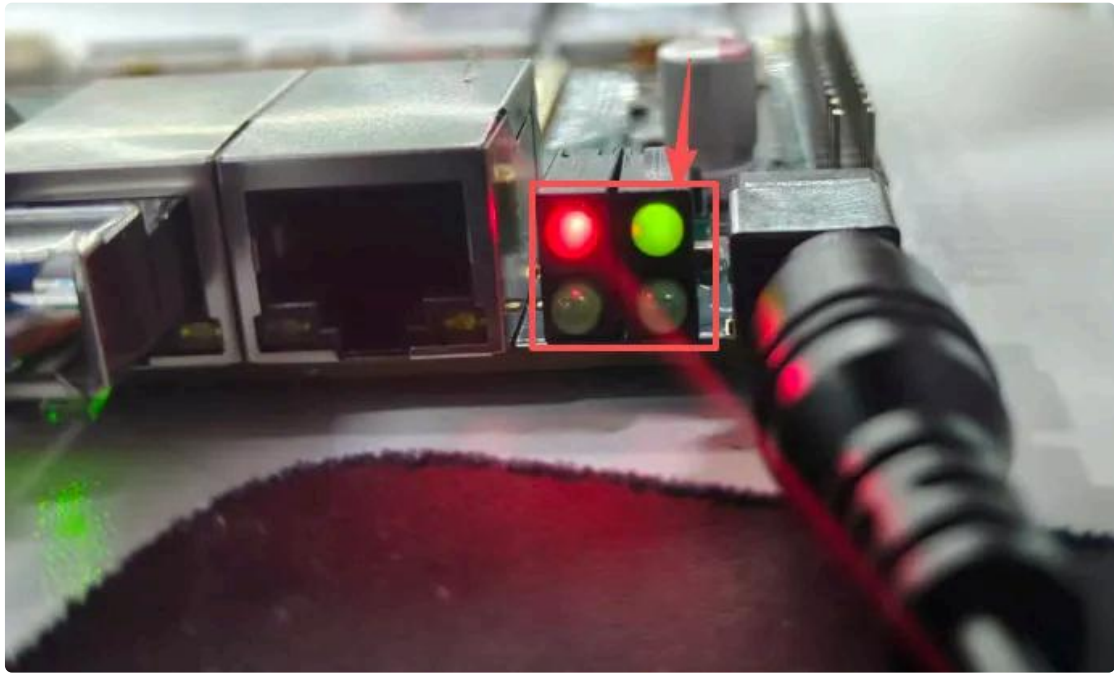
执行以下脚本，使用万用表实际去量电压值，是否是低电平

```

1  #!/bin/bash
2
3  GPIO_NUM=(40 42 44 45 46 47 203 202 8 9 10 198 199 206 207 197 7 228 225 230)
4
5  for gpio in "${GPIO_NUM[@]"; do
6      echo 0 > /sys/class/gpio/gpio$gpio/value
7      echo "GPIO $gpio in "
8
9  done

```

2.16. pwm



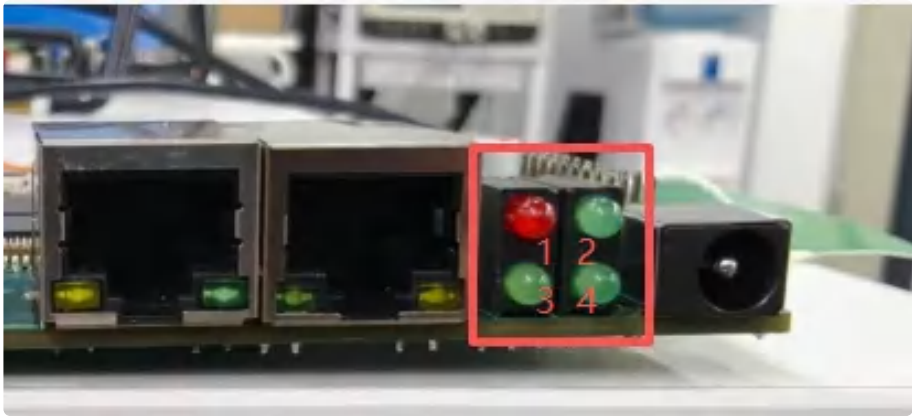
HDMI的 版本镜像pwm节点是 pwmchip1

```
▼ Bash |
1  cd /sys/class/pwm/pwmchip1/
2  echo 0 > export
3  cd pwm0
4  echo 10000 > period
5  echo 5000 > duty_cycle
6  echo normal > polarity
7  echo 1 > enable
```

MIPI 的版本镜像 pwm节点是 pwmchip2

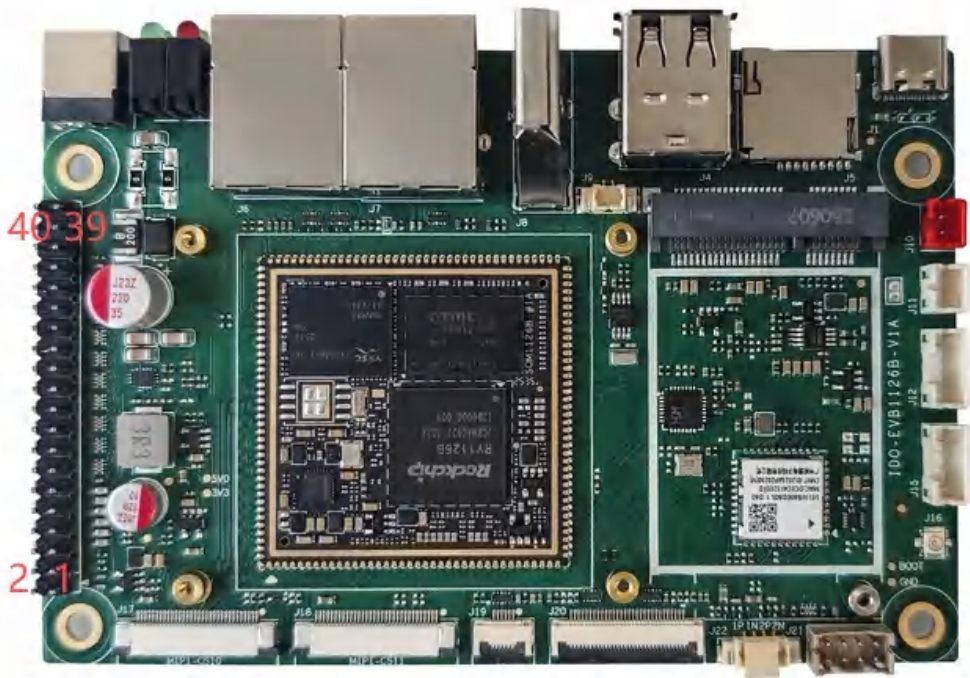
```
▼ Bash |
1  cd /sys/class/pwm/pwmchip2/
2  echo 0 > export
3  cd pwm0
4  echo 10000 > period
5  echo 5000 > duty_cycle
6  echo normal > polarity
7  echo 1 > enable
```

2.17. LED



- 1. 电源指示灯
- 2. PWM-LED 灯
- 3. 系统运行指示灯
- 4. 4G模块指示灯

2.18. ADC



序号	ADC通道	读值
33	SARADC0_IN1_	cat /sys/bus/iio/devices/iio\:device0/in_voltage1_raw
35	SARADC0_IN2_	cat /sys/bus/iio/devices/iio\:device0/in_voltage2_raw
37	SARADC0_IN3_	cat /sys/bus/iio/devices/iio\:device0/in_voltage3_raw
39	SARADC0_IN4_	cat /sys/bus/iio/devices/iio\:device0/in_voltage4_raw

4路ADC精度为13位，参考电压为1.8V。

ADC输入电压计算公式:

$$V=(\text{raw}/8191)*1.8\text{v}$$

2.19. I2C

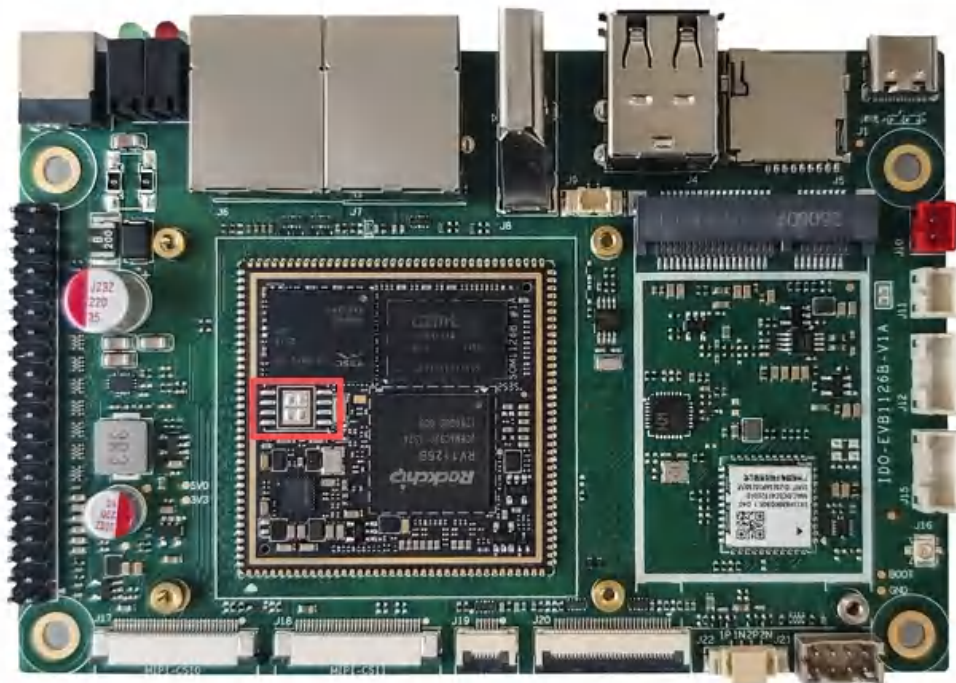
▼

Bash

```
1 root@rv1126b-buildroot:/# i2cdetect -y -r 1
2      0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
3 00:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
4 10:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
5 20:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
6 30:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
7 40:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
8 50:  -- 51  --  --  --  --  --  --  --  --  --  --  --  --  --  --
9 60:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
10 70:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
11 root@rv1126b-buildroot:/#
```

注意: 测试需要使用转接板。

2.20. SPI NAND



Bash

```
1 root@rv1126b-buildroot:/# dmesg | grep spi
2 [ 2.847504] spi-nand spi2.0: Winbond SPI NAND was found.
3 [ 2.847513] spi-nand spi2.0: 512 MiB, block size: 128 KiB, page size: 2048, OOB size: 128
4 [ 3.653787] mtdblock: MTD device 'spi-nand0' is NAND, please consider using UBIFS block devices instead.
5 [ 4.464264] mtdblock: MTD device 'spi-nand0' is NAND, please consider using UBIFS block devices instead.
6 root@rv1126b-buildroot:/#
```

2.21. NPU

2.21.1. yolov5测试:

链接: <https://pan.baidu.com/s/1N-vgqd1eH1Xcd7zxpfhZ4g> 提取码: hnpm

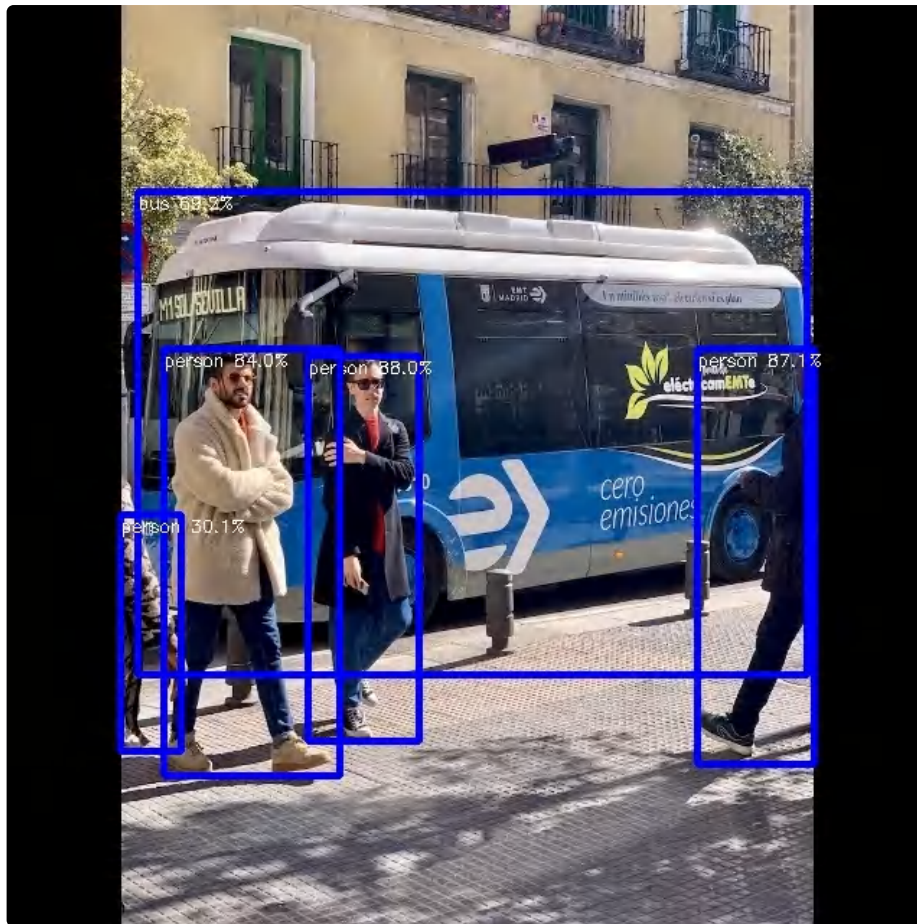
2.21.1.1. rknn_yolov5_demo:

```

1  root@rv1126b-buildroot:/# unzip rknn_yolov5_demo_Linux.zip
2  root@rv1126b-buildroot:/# cd rknn_yolov5_demo_Linux
3  root@rv1126b-buildroot:/# chmod a+x rknn_yolov5_demo
4  root@rv1126b-buildroot:/# ./rknn_yolov5_demo ./model/RV1126B/yolov5s-640-640.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=INT8, 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=INT8, 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_type=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_type=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：



2.21.1.2. rknn_yolov5_video_demo:

▼ Shell

```
1 root@rv1126b-buildroot:/rknn_yolov5_demo_Linux# ./rknn_yolov5_video_demo ./model/R
  V1126B/yolov5s-640-640.rknn ./1280x720.h264 264
2
```

最终生成 out.h264

▼ Shell

```
1 # 播放测试
2 gst-play-1.0 out.h264
```



2.21.2. yolo11 测试

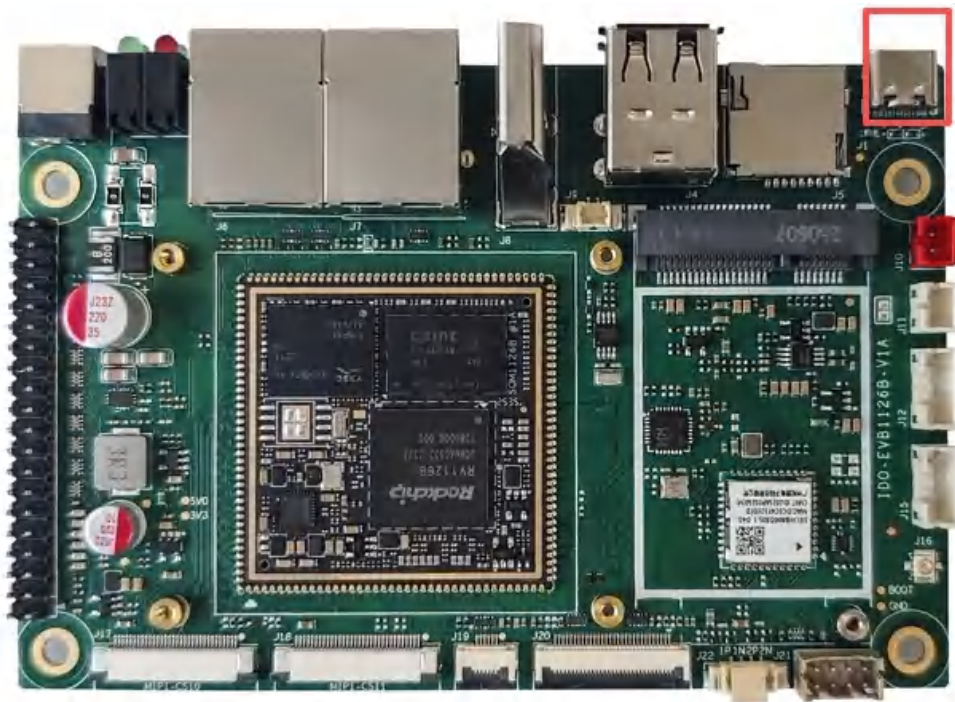
暂时不支持

3. IDO-EVB1126B-V1A-Debian11系统

3.1. 功能及接口使用方法

3.2. 调试串口

调试串口位置J1，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。



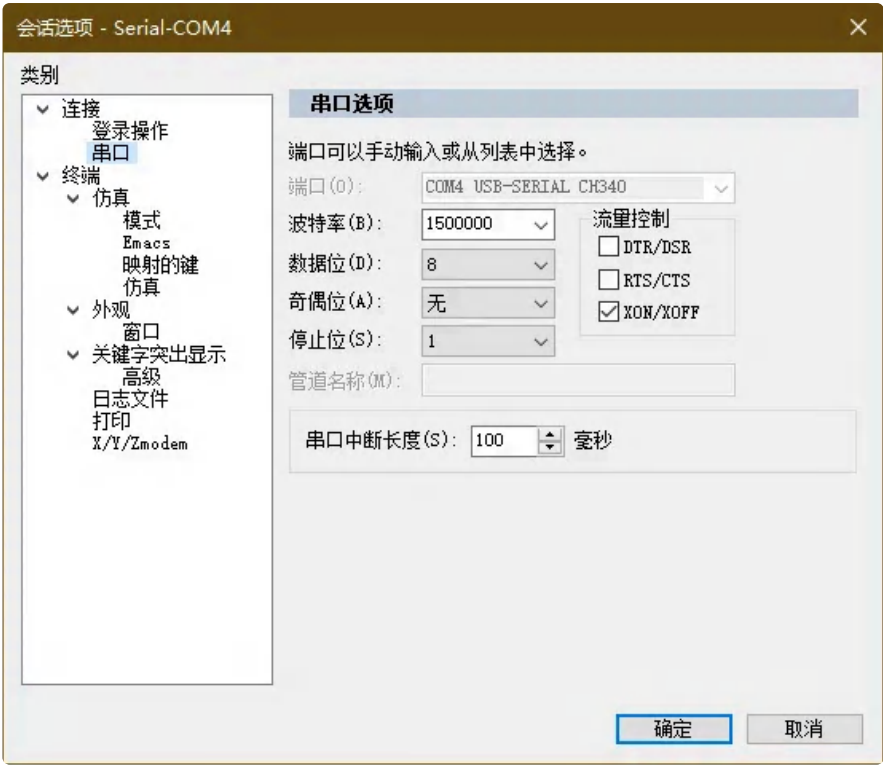
使用调试软件（MobaXterm、putty）等，设置参数如下：

波特率：1500000

数据位：8

奇偶校验位：无

停止位：1

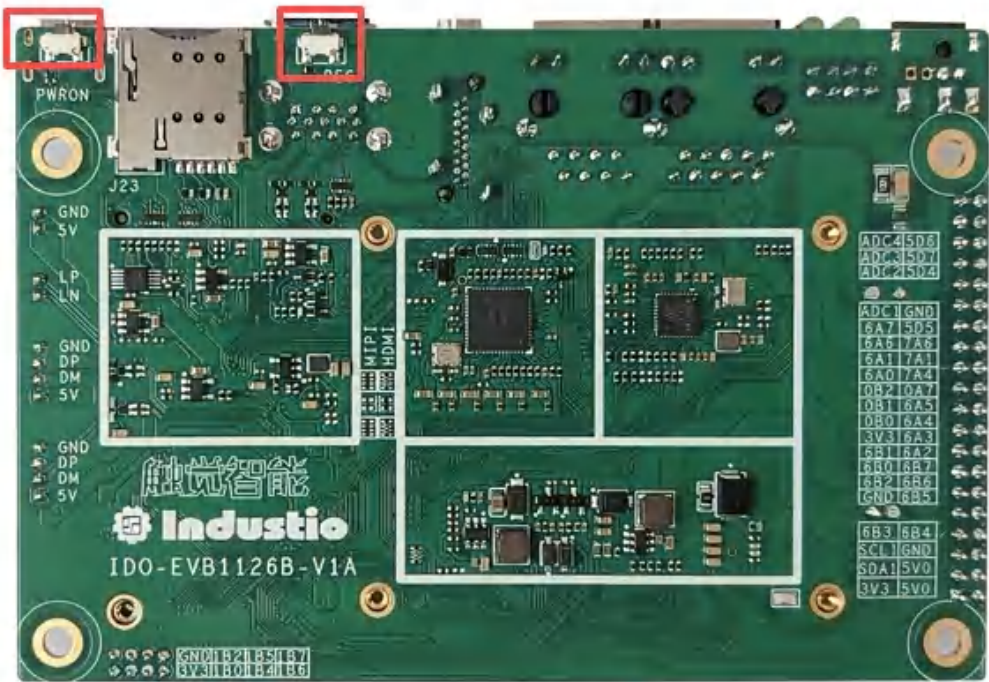


3.3. 按键

主板自带1个Recovery按键REC

Recovery按键，关机状态下，按住Recovery按键再给主板上电，主板会进入Loader模式。

power长按 强制关机，



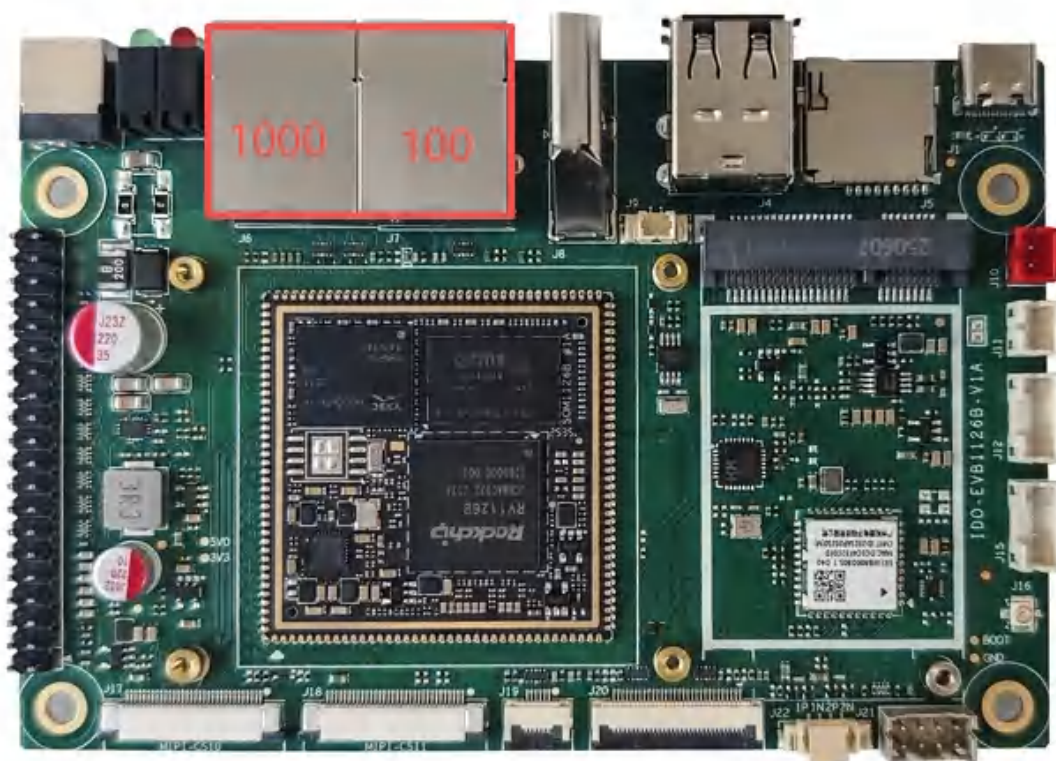
序号	定义	说明
1	PWRON_KEY	电源按键，用于开机/关机；

2	RECOVERY_KEY	烧录按键，用于烧录
---	--------------	-----------

3.4. 网络

3.4.1. Ethernet

主板有 1 路千兆以太网，1路百兆以太网，如下图所示：



▼

Bash

```
1  # TCP
2  #server
3  iperf -s -i 1
4  #client
5  iperf -c server.ip  # 上行
6  iperf -c server.ip -R 下行
7
8  # UDP
9  #server
10 iperf -u -s -i 1
11 #client
12 iperf -u -c server.ip -b 1000M  # 上行
13 iperf -u -c server.ip -b 1000M -R # 下行
```

可通过ifconfig指令设置临时静态IP，命令如下：

```
1 $ ifconfig eth0 192.168.3.123
```

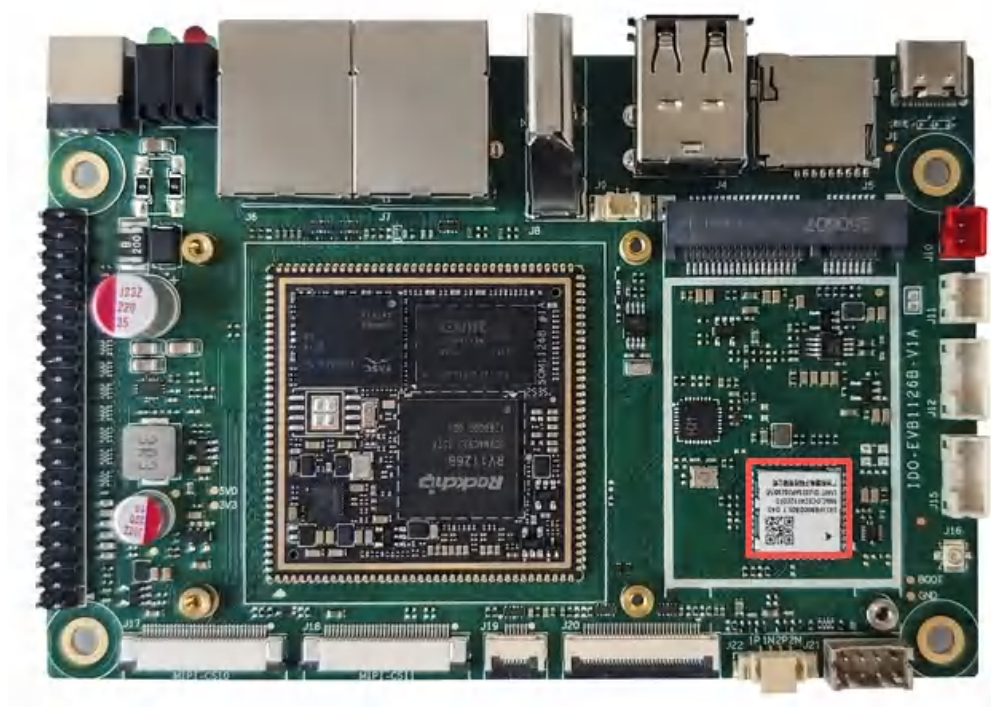
如果能让静态IP永久生效，编辑文件 /etc/network/interfaces

```
1 root@linaro-alip:~# vi /etc/network/interfaces
2 root@linaro-alip:~# cat /etc/network/interfaces
3 # interface file auto-generated by buildroot
4
5 auto lo
6 iface lo inet loopback
7
8 auto eth0
9 iface eth0 inet static
10 address 192.168.3.123
11 netmask 255.255.255.0
12 gateway 192.168.3.1
13
14 source-directory /etc/network/interfaces.d
```

设置后，重启

3.4.2. 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J62，如下图所示：



使用nmcli工具连接wifi热点：



Bash

```
1 root@linaro-alip:~# nmcli dev wifi list #查询wifi列表
2 root@linaro-alip:~# nmcli dev wifi connect "industio" password "123456789" ifname wlan0 #连接热点
```

3.4.3. Bluetooth

蓝牙节点：



Bash

```
1 root@linaro-alip:~# hciconfig -a
2 hci0: Type: Primary Bus: UART
3 BD Address: DC:EC:4F:32:E6:D4 ACL MTU: 1021:9 SCO MTU: 255:4
4 UP RUNNING
5 RX bytes:930 acl:0 sco:0 events:73 errors:0
6 TX bytes:5493 acl:0 sco:0 commands:73 errors:0
7 Features: 0xbf 0x2e 0x4d 0xfe 0xd8 0x3f 0x7b 0x87
8 Packet type: DM1 DM3 DM5 DH1 DH3 DH5 HV1 HV3
9 Link policy: RSWITCH SNIFF
10 Link mode: PERIPHERAL ACCEPT
11 Name: 'linaro-alip'
12 Class: 0x7c0000
13 Service Classes: Rendering, Capturing, Object Transfer, Audio, Telephony
14 Device Class: Miscellaneous,
15 HCI Version: (0xd) Revision: 0xb
16 LMP Version: (0xd) Subversion: 0xb
17 Manufacturer: not assigned (2875)
18
19 root@linaro-alip:~#
```

3.4.3.1. 命令行连接

Bash

```

1  bluetoothctl
2  hci0 new_settings: powered bondable ssp br/edr le secure-conn
3  Agent registered
4  [CHG] Controller 4C:38:00:2B:C1:71 Pairable: yes
5  [bluetooth]#
6  #打开蓝牙
7  [bluetooth]# power on
8  #扫描蓝牙
9  [bluetooth]# scan on
10 # # 开启广播
11 [bluetooth]# discoverable on
12 # 允许配对
13 [bluetooth]# pairable on
14 # 使用 NoInputNoOutput (Just Works)
15 [bluetooth]# agent NoInputNoOutput
16 [bluetooth]# default-agent
17 #查看蓝牙设备
18 [bluetooth]# devices
19 #信任蓝牙设备
20 [bluetooth]# trust 7C:C1:80:09:DD:6C
21 #蓝牙配对
22 [bluetooth]# pair 7C:C1:80:09:DD:6C
23 #连接蓝牙
24 [bluetooth]# connect 7C:C1:80:09:DD:6C

```

已配对设备



linaro-alip aptX HD
使用中



```

[bluetooth]# connect A8:C5:6F:AA:FC:A8
Attempting to connect to A8:C5:6F:AA:FC:A8
[CHG] Device A8:C5:6F:AA:FC:A8 Connected: yes
[NEW] Endpoint /org/bluez/hci0/dev_A8_C5_6F_AA_FC_A8/sep1
[NEW] Endpoint /org/bluez/hci0/dev_A8_C5_6F_AA_FC_A8/sep2
[NEW] Endpoint /org/bluez/hci0/dev_A8_C5_6F_AA_FC_A8/sep3
[NEW] Endpoint /org/bluez/hci0/dev_A8_C5_6F_AA_FC_A8/sep4
[NEW] Endpoint /org/bluez/hci0/dev_A8_C5_6F_AA_FC_A8/sep5
[NEW] Transport /org/bluez/hci0/dev_A8_C5_6F_AA_FC_A8/sep4/fd0
Connection successful
[NEW] Player /org/bluez/hci0/dev_A8_C5_6F_AA_FC_A8/player0 [default]
[CHG] Device A8:C5:6F:AA:FC:A8 ServicesResolved: yes
[CHG] Player /org/bluez/hci0/dev_A8_C5_6F_AA_FC_A8/player0 Type: Audio
[CHG] Player /org/bluez/hci0/dev_A8_C5_6F_AA_FC_A8/player0 Subtype: None
[CHG] Player /org/bluez/hci0/dev_A8_C5_6F_AA_FC_A8/player0 Status: paused
[CHG] Player /org/bluez/hci0/dev_A8_C5_6F_AA_FC_A8/player0 Name: Bluetooth Player
[CHG] Player /org/bluez/hci0/dev_A8_C5_6F_AA_FC_A8/player0 Repeat: off
[CHG] Player /org/bluez/hci0/dev_A8_C5_6F_AA_FC_A8/player0 Shuffle: off
[CHG] Player /org/bluez/hci0/dev_A8_C5_6F_AA_FC_A8/player0 Title: 凌晨的国道上, 有赌徒, 有破产的老板, 有三教九流的大神, 唯独没有
[CHG] Player /org/bluez/hci0/dev_A8_C5_6F_AA_FC_A8/player0 TrackNumber: 0x00000001 (1)
[CHG] Player /org/bluez/hci0/dev_A8_C5_6F_AA_FC_A8/player0 NumberOfTracks: 0x00000001 (1)
[CHG] Player /org/bluez/hci0/dev_A8_C5_6F_AA_FC_A8/player0 Duration: 0x0000491b (18715)
[CHG] Player /org/bluez/hci0/dev_A8_C5_6F_AA_FC_A8/player0 Artist: 摄影师-高原
[CHG] Player /org/bluez/hci0/dev_A8_C5_6F_AA_FC_A8/player0 Position: 0x00000dec (3564)
[CHG] Player /org/bluez/hci0/dev_A8_C5_6F_AA_FC_A8/player0 Position: 0x00000dec (3564)
[CHG] Controller DC:EC:4F:1D:E7:C6 Discoverable: no
[OPPO Find X8]#

```

注意: 如果出现无法配对的情况, 请执行quit退出bluetoothctl, 重新启动蓝牙服务, 再次执行以上连接命令

Bash

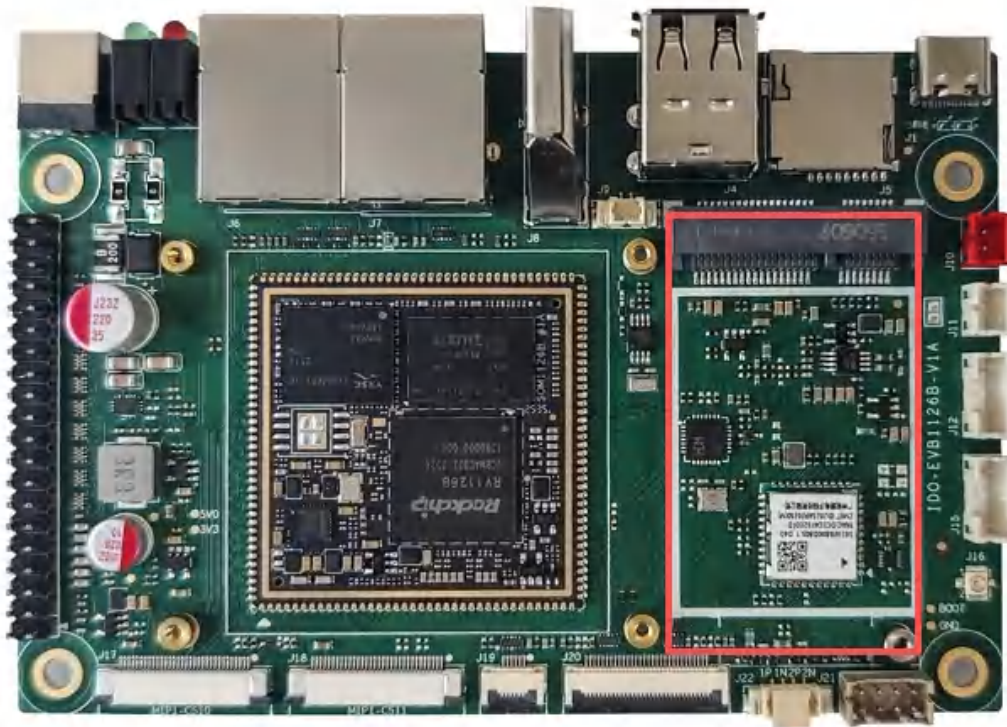
```

1  [bluetooth]# quit
2  root@linaro-alip:/# systemctl restart bluetooth

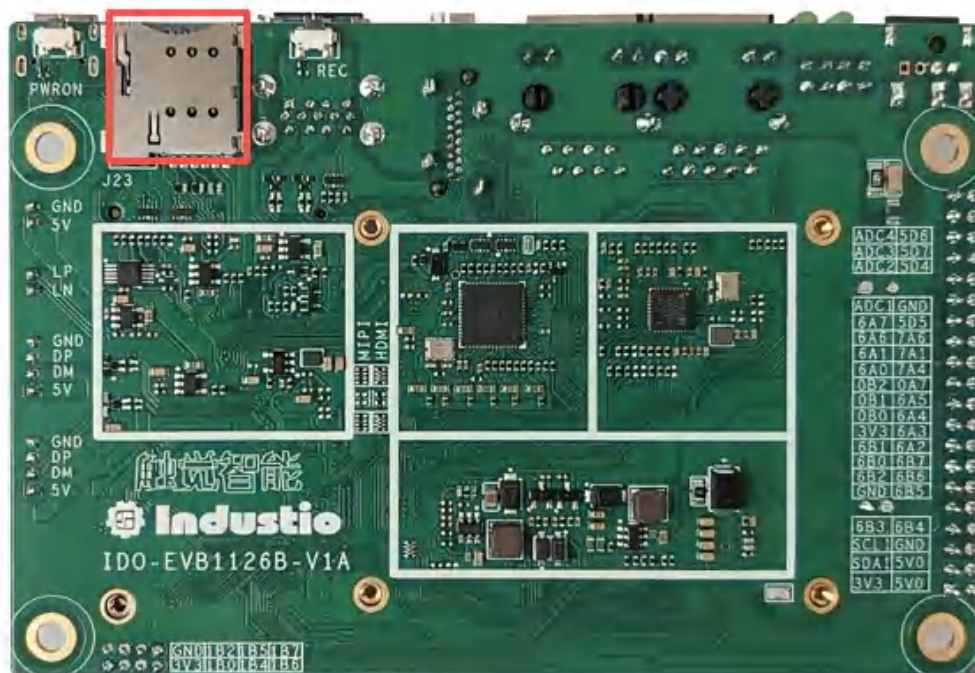
```

3.4.4. 4G/5G

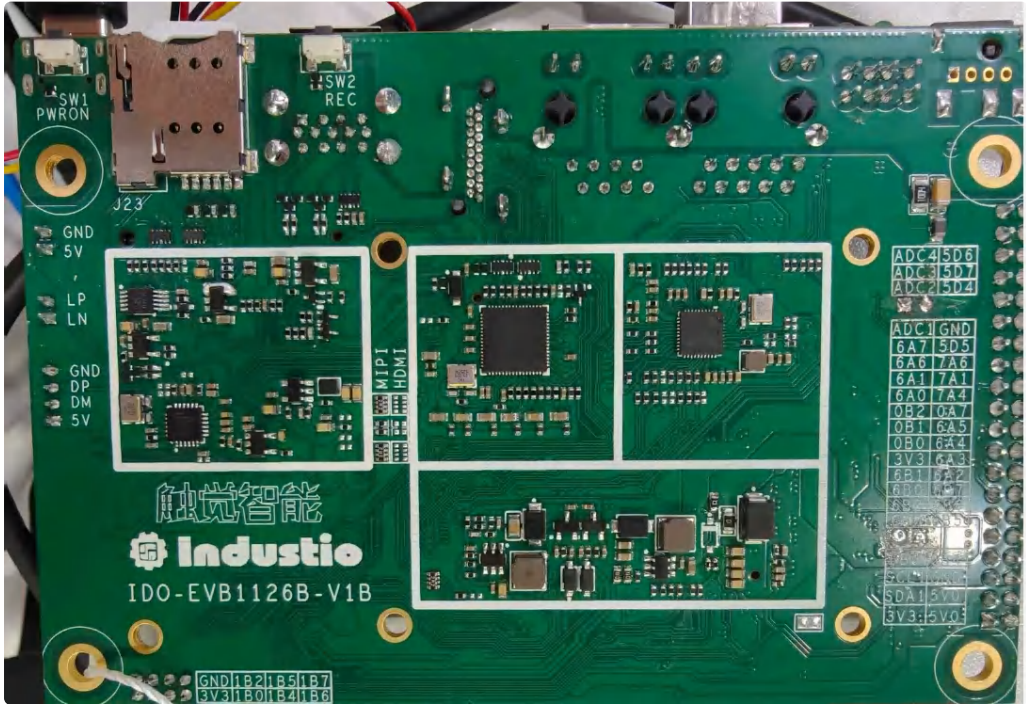
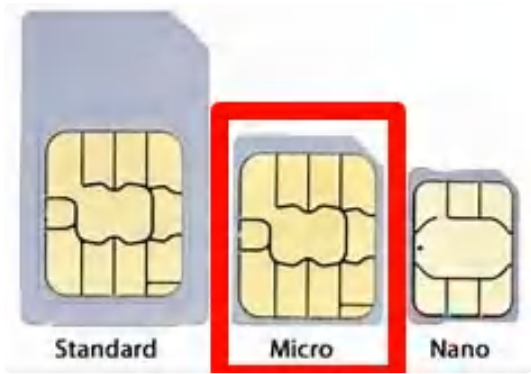
主板内置Mini PCIe 座子扩展 4G/5G模块，4G通信模块适配移远EC20、EC25等通用模组。如下图所示：



模块需要接上天线以确保获得更好的信号质量，SIM卡安装位J23位于主板背面，如下图所示：



使用Micro尺寸SIM卡，如下图所示：



EC20/25	ecm模式	rmnet模式
节点	usb0	wwan0
拨号方式	自动拨号	quectel-CM



ec20 模式切换。 , 切换后重启

Bash

```
1 echo -e "AT+QCFG=\"usbnet\",1\r\n" > /dev/ttyUSB2 #节点usb0
2 echo -e "AT+QCFG=\"usbnet\",0\r\n" > /dev/ttyUSB2 #节点wwan0
```

测试模块EC20, ecm模式(usb0)

Bash

```
1 echo -e "AT+CGDCONT=1,\"IP\", \"apn\"\r\n" > /dev/ttyUSB2
2 echo -e "AT+QNETDEVTCL=1,1,1\r\n" > /dev/ttyUSB2
```

```
▼ Bash |
1 root@linaro-alip:/# ifconfig usb0
2 usb0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
3     inet 192.168.225.30 netmask 255.255.255.0 broadcast 192.168.225.255
4     inet6 fe80::771f:e144:67dd:d9ad prefixlen 64 scopeid 0x20<link>
5     inet6 240e:47c:3080:e:88ce:6089:1be:5ff4 prefixlen 64 scopeid 0x0<global>
6
7     ether 36:b4:9e:ab:ab:01 txqueuelen 1000 (Ethernet)
8     RX packets 20542 bytes 27780090 (26.4 MiB)
9     RX errors 0 dropped 0 overruns 0 frame 0
10    TX packets 10008 bytes 868483 (848.1 KiB)
11    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
12
13 root@linaro-alip:/# ping baidu.com -I usb0
14 PING baidu.com (39.156.70.37) from 192.168.225.30 usb0: 56(84) bytes of data.
15 64 bytes from 39.156.70.37 (39.156.70.37): icmp_seq=1 ttl=47 time=208 ms
16 64 bytes from 39.156.70.37 (39.156.70.37): icmp_seq=2 ttl=47 time=74.8 ms
17 64 bytes from 39.156.70.37 (39.156.70.37): icmp_seq=3 ttl=47 time=82.8 ms
18 ^C
19 --- baidu.com ping statistics ---
20 3 packets transmitted, 3 received, 0% packet loss, time 2002ms
21 rtt min/avg/max/mdev = 74.768/121.737/207.605/60.807 ms
```

测试模块EC20, qmi(wwan0)

```
▼ Bash |
1 # 模式修改
2 echo -e "AT+QCFG=\"usbnet\",0\r\n" > /dev/ttyUSB2 #节点wwan0
3 # 修改后需要重启系统, 或者重启模块
4 #重启模块命令
5 echo -e "AT+CFUN=1,1\r\n" >/dev/ttyUSB2
```

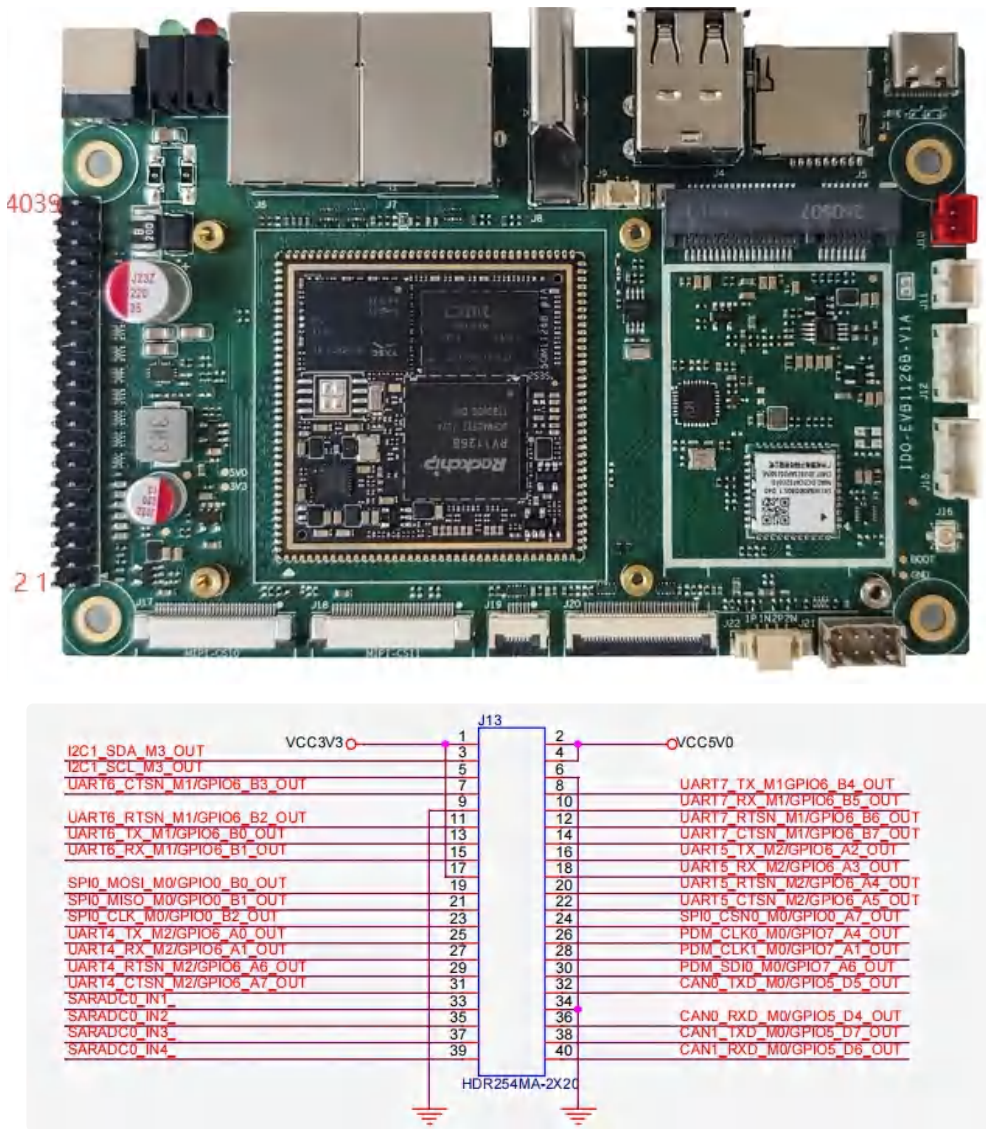
Bash

```
1 root@linaro-alip:/# quectel-CM &
2 [1] 1963
3 root@linaro-alip:/# [10-18_17:16:56:845] QConnectManager_Linux_V1.6.4
4 [10-18_17:16:56:847] Find /sys/bus/usb/devices/1-1.2 idVendor=0x2c7c idProduct=0x
125, bus=0x001, dev=0x005
5 [10-18_17:16:56:848] Auto find qmichannel = /dev/cdc-wdm0
6 [10-18_17:16:56:848] Auto find usbnet_adapter = wwan0
7 [10-18_17:16:56:848] netcard driver = qmi_wwan, driver version = 6.1.141
8 [10-18_17:16:56:848] Modem works in QMI mode
9 [10-18_17:16:56:861] cdc_wdm_fd = 7
10 [10-18_17:16:56:947] Get clientWDS = 5
11 [10-18_17:16:56:979] Get clientDMS = 1
12 [10-18_17:16:57:010] Get clientNAS = 2
13 [10-18_17:16:57:043] Get clientUIM = 1
14 [10-18_17:16:57:075] Get clientWDA = 1
15 [10-18_17:16:57:107] requestBaseBandVersion EC25EUXGAR08A09M1G
16 [10-18_17:16:57:235] requestGetSIMStatus SIMStatus: SIM_READY
17 [10-18_17:16:57:267] requestGetProfile[1] ctlte///0/IPV4V6
18 [10-18_17:16:57:299] requestRegistrationState2 MCC: 460, MNC: 11, PS: Attached, D
ataCap: LTE
19 [10-18_17:16:57:330] requestQueryDataCall IPv4ConnectionStatus: DISCONNECTED
20 [10-18_17:16:57:331] ifconfig wwan0 0.0.0.0
21 [10-18_17:16:57:340] ifconfig wwan0 down
22 [10-18_17:16:57:395] requestSetupDataCall WdsConnectionIPv4Handle: 0x868a7ea0
23 [10-18_17:16:57:524] ifconfig wwan0 up
24 [10-18_17:16:57:533] busybox udhcpc -f -n -q -t 5 -i wwan0
25 udhcpc: started, v1.35.0
26 /etc/resolvconf/update.d/libc: Warning: /etc/resolv.conf is not a symbolic link t
o /run/resolvconf/resolv.conf
27 udhcpc: broadcasting discover
28 udhcpc: broadcasting select for 10.2.29.162, server 10.2.29.161
29 udhcpc: lease of 10.2.29.162 obtained from 10.2.29.161, lease time 7200
30 /etc/resolvconf/update.d/libc: Warning: /etc/resolv.conf is not a symbolic link t
o /run/resolvconf/resolv.conf
31
32
33
34 root@linaro-alip:/# ifconfig wwan0
35 wwan0: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST> mtu 1500
36     inet 10.2.29.162 netmask 255.255.255.252 destination 10.2.29.162
37     inet6 fe80::28fd:1308:143f:2ae9 prefixlen 64 scopeid 0x20<link>
38     unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00 txqueuelen 1000
    (UNSPEC)
39     RX packets 4 bytes 1224 (1.1 KiB)
40     RX errors 0 dropped 0 overruns 0 frame 0
41     TX packets 35 bytes 4960 (4.8 KiB)
42     TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
43
44 root@linaro-alip:/# ping baidu.com -I wwan0
45 PING baidu.com (39.156.70.37) from 10.2.29.162 wwan0: 56(84) bytes of data.
46 64 bytes from 39.156.70.37 (39.156.70.37): icmp_seq=1 ttl=48 time=62.9 ms
47 64 bytes from 39.156.70.37 (39.156.70.37): icmp_seq=2 ttl=48 time=65.8 ms
48 64 bytes from 39.156.70.37 (39.156.70.37): icmp_seq=3 ttl=48 time=58.8 ms
49 64 bytes from 39.156.70.37 (39.156.70.37): icmp_seq=4 ttl=48 time=59.7 ms
50 64 bytes from 39.156.70.37 (39.156.70.37): icmp_seq=5 ttl=48 time=57.9 ms
51 ^C64 bytes from 39.156.70.37: icmp_seq=7 ttl=48 time=58.3 ms
```

```
52
53 --- baidu.com ping statistics ---
54 7 packets transmitted, 6 received, 14.2857% packet loss, time 6003ms
55 rtt min/avg/max/mdev = 57.888/60.576/65.807/2.852 ms
56
```

3.5. UART

IDO-EVB1126B-V1主板一共引出6路UART串口（不含调试串口），如下图所示：



UART串口设备节点列表如下：

序号	默认电平类型	设备节点
J13-25,J13-27	TTL	/dev/ttyS4
J3-16,J3-18,J3-20	RS485	/dev/ttyS5
J3-13,J3-15	TTL	/dev/ttyS6

J3-8,J3-10	TTL	/dev/ttyS7
------------	-----	------------

使用microcom进行串口测试:

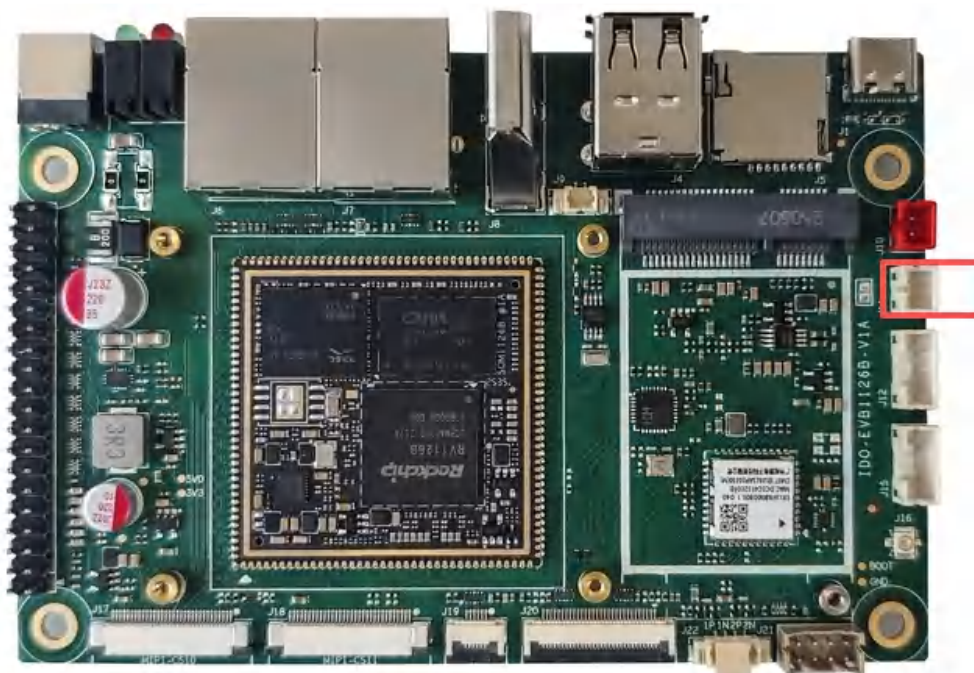
```
▼ Bash
1 root@linaro-alip:/# microcom /dev/ttyS7 -s 115200
2
```

注意: 执行 Ctrl x 退出程序; RS485 需要使用转接板测试

3.6. Audio

3.6.1. Speaker

喇叭接口为PH2.0-P连接器J66, 如下图所示:



声卡节点:

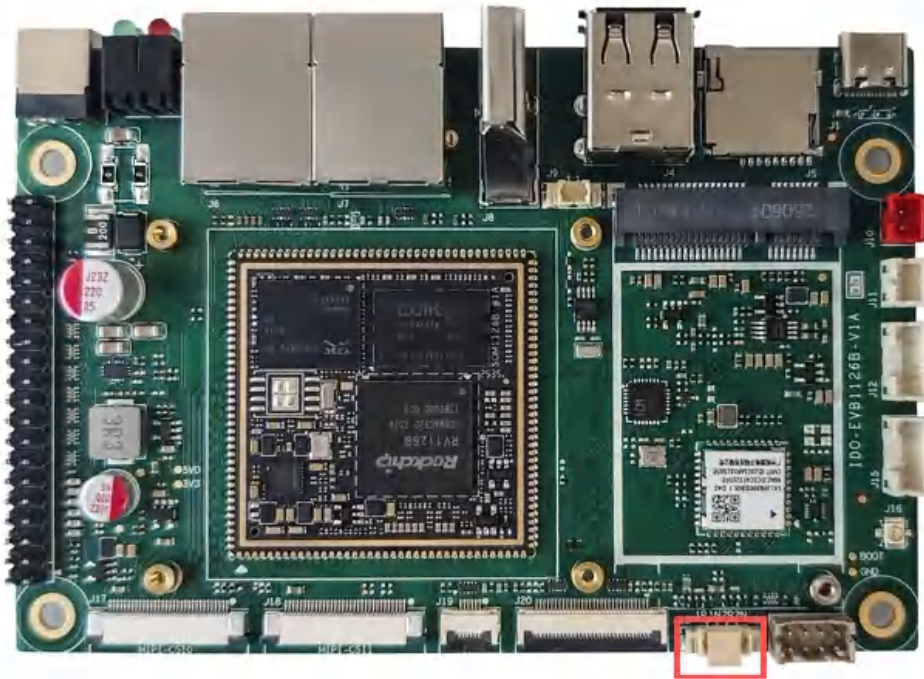
```
▼ Bash
1 root@linaro-alip:/# aplay -l
2 **** List of PLAYBACK Hardware Devices ****
3 card 0: rockchiprv1126b [rockchip,rv1126b-acodec], device 0: dailink-multicodecs m
  ulticodec-0 [dailink-multicodecs multicodec-0]
4   Subdevices: 1/1
5   Subdevice #0: subdevice #0
6 root@linaro-alip:/#
```

音频播放:

▼ Bash

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

3.6.2. MIC



注意：录出来音频文件需要导出到PC 上播放

录音

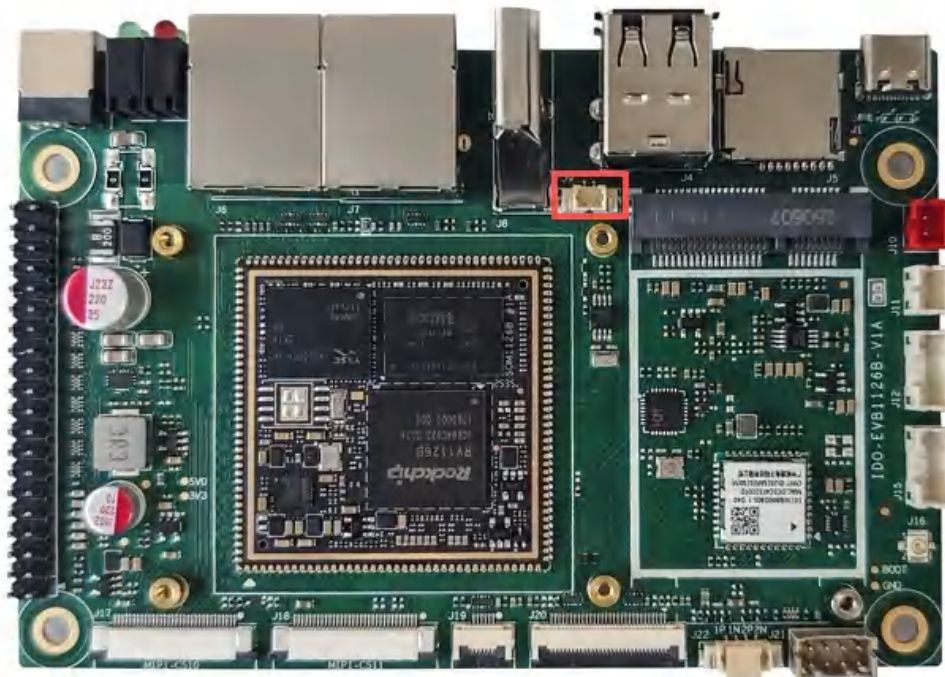
▼ Bash

```
1 # 一共两路mic , mic0 录音命令
2 arecord -D hw:0,0 -f S16_LE -r 16000 -c 4 -t wav test_mic0.wav
3 # mic1 录音命令
4 arecord -D hw:0,0 -f S16_LE -r 16000 -c 4 -t wav test_mic1.wav
```

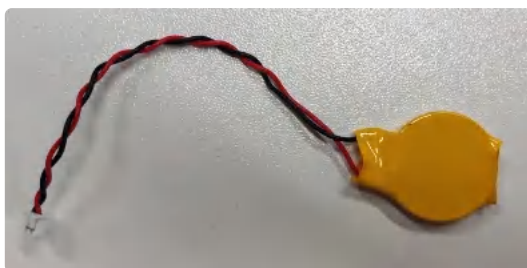
注意：录出来音频文件需要导出到PC 上播放

3.7. RTC

MX1.25T-2P RTC电池座J9，如下图所示：



连接3V 纽扣电池，RTC电池参考如下



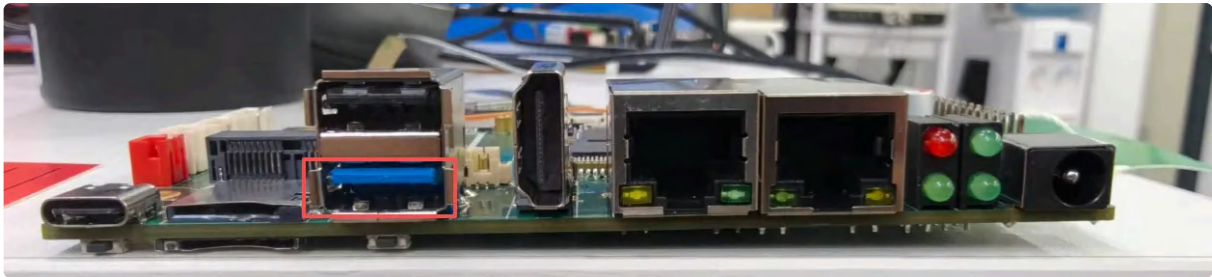
系统默认使用HYM8563作为系统时钟，设备节点：/dev/rtc0，时间设置方法：

```
1 root@linaro-alip:/# hwclock -w    # 向rtc写时间
2 root@linaro-alip:/# hwclock -r    # 向rtc读时间
```

3.8. USB

主板支持1个TypeA接口（USB3.1 Gen1 OTG），支持1个USB3.0-otg接口，1个USB2.0PH2.0-4P 接口，一个USB2.0TypeA USB对外总供电应小于4A。

3.8.1. OTG



EVB1126B 已实现上电启动自动切换OTG

- 1. 上电启动前接入OTG 通过双公头usb连接 PC , 那么otg为device模式
- 2. 上电启动前接入U盘等usb设备, 启动进入系统后otg为host模式

系统下otg模式切换命令:

命令	模式
gpio-otg HOST	HOST 模式
gpio-otg DEVICE	DEVICE 模式

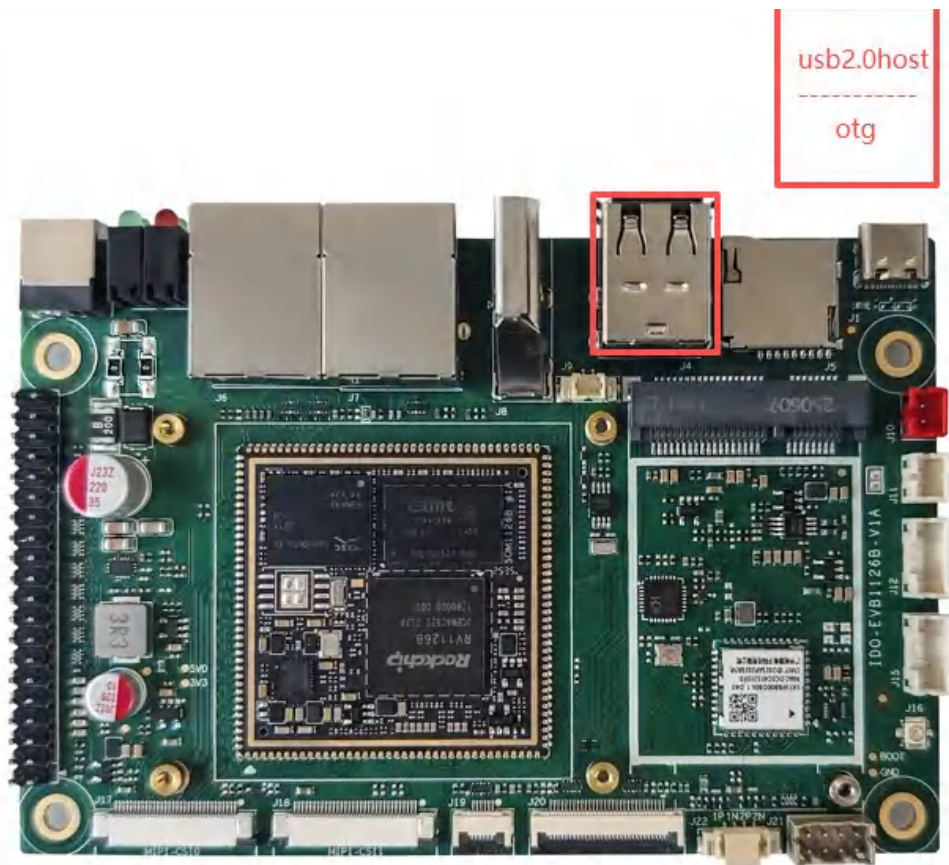
▼

Bash

```
1  # host模式
2  root@linaro-alip:/# gpio-otg HOST
3  # device模式
4  root@linaro-alip:/# gpio-otg DEVICE
```

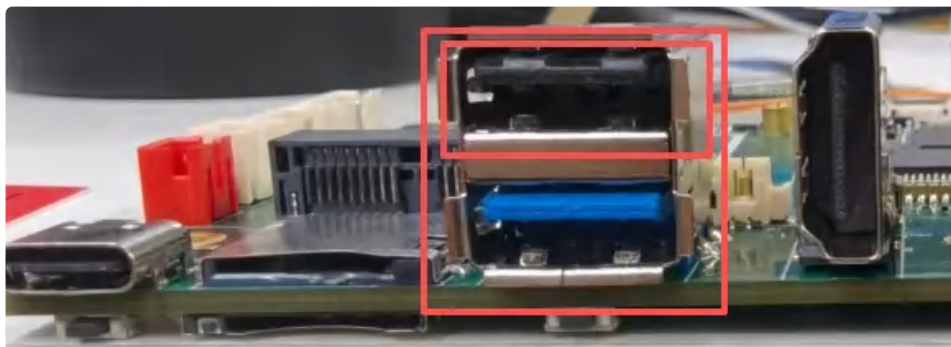
3.8.2. USB3.0

主板支持1个USB3.0-otg接口, 接口为标准的双层A口J4, 如下图所示:



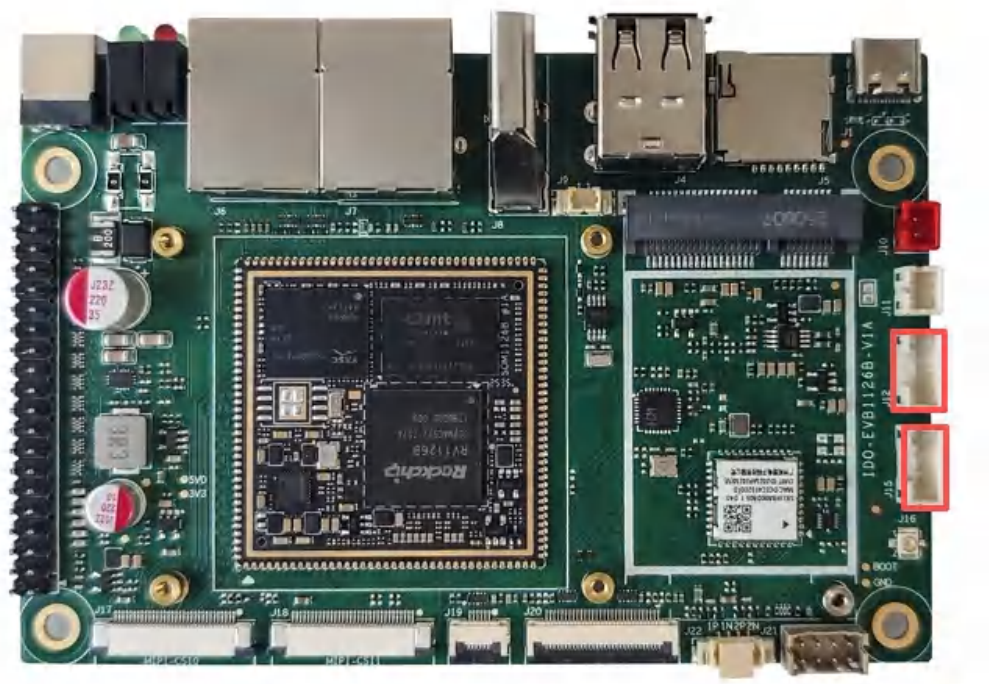
3.8.3. USB2.0-A

主板配置了1路USB2.0-A接口，如下图所示：



3.8.4. USB2.0-PH

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



座子序号	控制节点
J12	/sys/class/leds/usb_vbus3_en/brightness



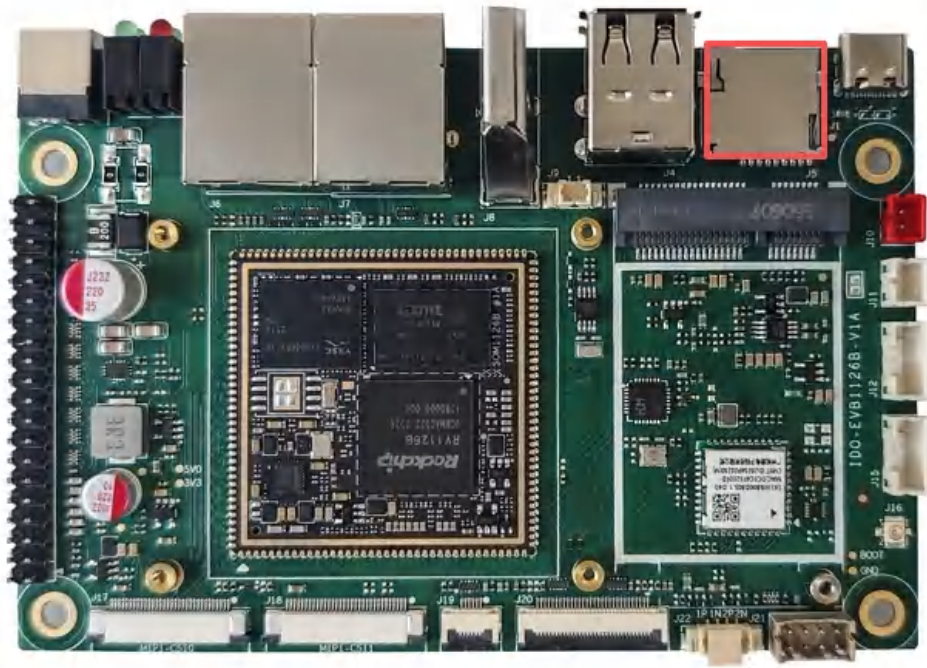
▼

Bash

```
1 echo 1 > /sys/class/leds/usb_vbus3_en/brightness # 打开
2 echo 0 > /sys/class/leds/usb_vbus3_en/brightness # 关闭
```

3.9. TF Card

TF卡座支持SDIO3.0，支持高速SD卡，接口位于主板背面J43，如下图所示：



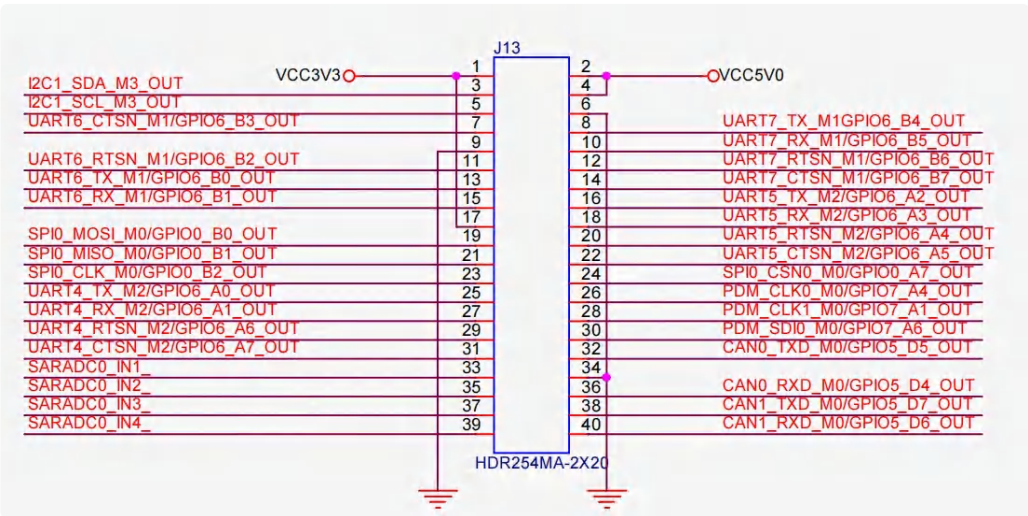
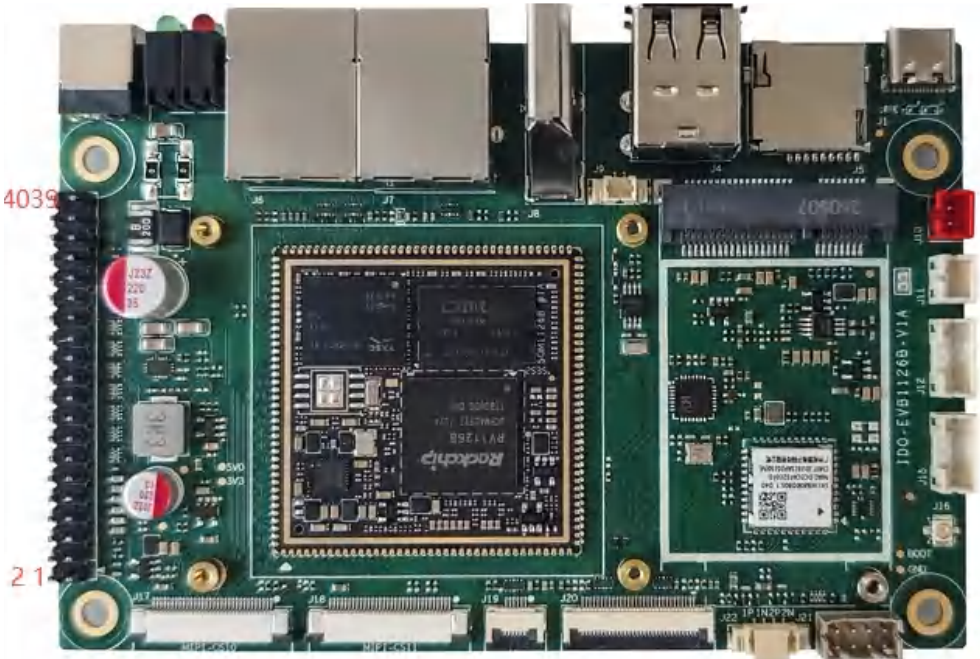
Bash

```
1 root@linaro-alip:/# fdisk -l
2 Disk /dev/ram0: 4 MiB, 4194304 bytes, 8192 sectors
3 Units: sectors of 1 * 512 = 512 bytes
4 Sector size (logical/physical): 512 bytes / 4096 bytes
5 I/O size (minimum/optimal): 4096 bytes / 4096 bytes
6
7
8 Disk /dev/mmcblk0: 116.48 GiB, 125074145280 bytes, 244285440 sectors
9 Units: sectors of 1 * 512 = 512 bytes
10 Sector size (logical/physical): 512 bytes / 512 bytes
11 I/O size (minimum/optimal): 512 bytes / 512 bytes
12 Disklabel type: gpt
13 Disk identifier: DB600000-0000-4377-8000-3FB20000370D
14
15 Device          Start      End  Sectors  Size Type
16 /dev/mmcblk0p1   16384     24575     8192    4M unknown
17 /dev/mmcblk0p2   24576     32767     8192    4M unknown
18 /dev/mmcblk0p3   32768    163839    131072   64M unknown
19 /dev/mmcblk0p4   163840    425983    262144  128M unknown
20 /dev/mmcblk0p5   425984    491519    65536   32M unknown
21 /dev/mmcblk0p6   491520   13074431  12582912 6G unknown
22 /dev/mmcblk0p7  13074432  13336575    262144  128M unknown
23 /dev/mmcblk0p8  13336576  244285406 230948831 110.1G unknown
24
25
26 Disk /dev/mmcblk0boot0: 4 MiB, 4194304 bytes, 8192 sectors
27 Units: sectors of 1 * 512 = 512 bytes
28 Sector size (logical/physical): 512 bytes / 512 bytes
29 I/O size (minimum/optimal): 512 bytes / 512 bytes
30
31
32 Disk /dev/mmcblk0boot1: 4 MiB, 4194304 bytes, 8192 sectors
33 Units: sectors of 1 * 512 = 512 bytes
34 Sector size (logical/physical): 512 bytes / 512 bytes
35 I/O size (minimum/optimal): 512 bytes / 512 bytes
36
37
38 Disk /dev/mmcblk1: 14.56 GiB, 15634268160 bytes, 30535680 sectors
39 Units: sectors of 1 * 512 = 512 bytes
40 Sector size (logical/physical): 512 bytes / 512 bytes
41 I/O size (minimum/optimal): 512 bytes / 512 bytes
42 Disklabel type: dos
43 Disk identifier: 0x00000000
44
45 root@linaro-alip:/# mount /dev/mmcblk1 /mnt
46 root@linaro-alip:/# df
47 Filesystem      1K-blocks    Used Available Use% Mounted on
48 /dev/root        3021512 2575920    271924  91% /
49 devtmpfs         981636      0    981636   0% /dev
50 tmpfs           1009988      0   1009988   0% /dev/shm
51 tmpfs           403996    1636    402360   1% /run
52 tmpfs           5120        8     5112   1% /run/lock
53 tmpfs           201996     24    201972   1% /run/user/1000
54 tmpfs           201996     20    201976   1% /run/user/103
55 /dev/mmcblk1    15267836 2114500  13153336  14% /mnt
56
```

默认挂在路径为 /media/mmcblk*

3.10. CANFD

主板共有2路CAN，如下图所示:



CAN	接线
can0	J13-32 ,J13-34
can1	J13-38 ,J13-40

分别对应系统节点名称为 can0、can1。使用can盒测试:

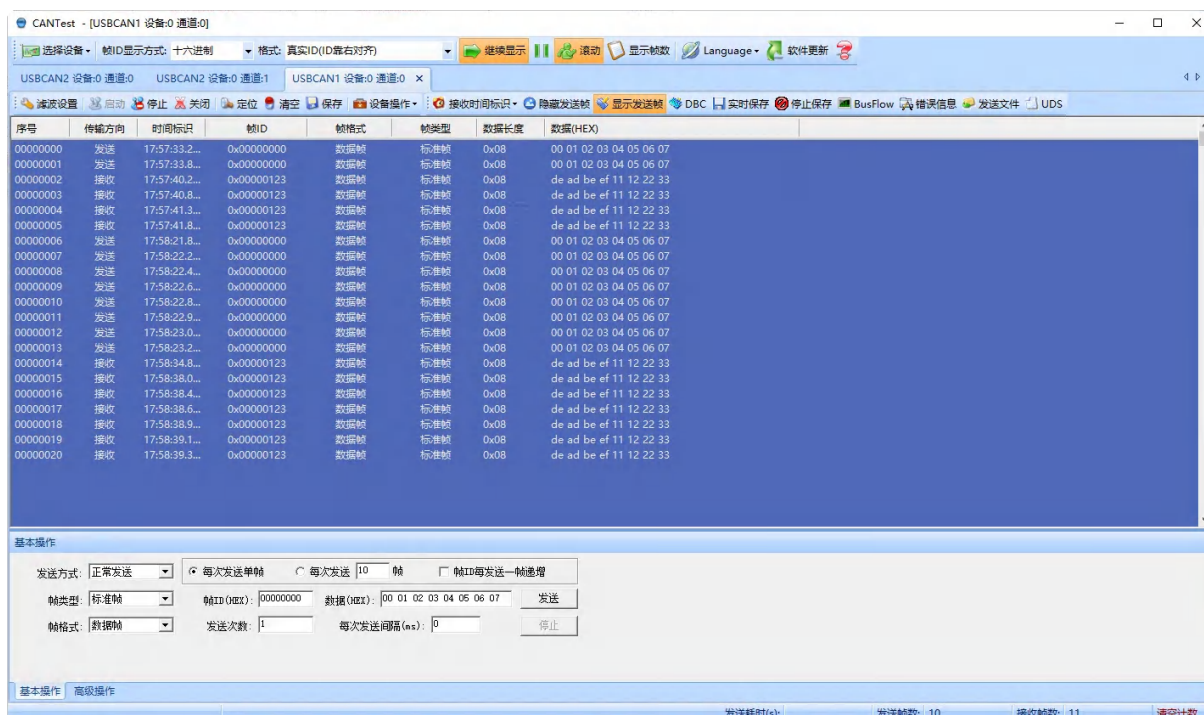
1. 设置CAN参数

Bash

```

1  # can0节点发送数据
2  ip link set can0 down
3  ip link set can0 type can bitrate 1000000 dbitrade 1000000 fd on
4  ip link set can0 up
5  cansend can0 123#DEADBEEF
6  cansend can0 123#DEADBEEF11122233
7
8  # can0接收数据
9  ip link set can0 down
10 ip link set can0 type can bitrate 1000000 dbitrade 1000000 fd on
11 ip link set can0 up
12 candump can1
13
14 # can1 接收数据
15 ip link set can1 down
16 ip link set can1 type can bitrate 1000000 dbitrade 1000000 fd on
17 ip link set can1 up

```



```

root@rv1126b-buildroot:/# candump can0
can0 000 [8] 00 01 02 03 04 05 06 07
can0 000 [8] 00 01 02 03 04 05 06 07
can0 000 [8] 00 01 02 03 04 05 06 07
^Croot@rv1126b-buildroot:/# ^C
root@rv1126b-buildroot:/# ^C
root@rv1126b-buildroot:/# ^C
root@rv1126b-buildroot:/# cansend can0 123#DEADBEEF11122233
root@rv1126b-buildroot:/# cansend can0 123#DEADBEEF11122233
root@rv1126b-buildroot:/# cansend can0 123#DEADBEEF11122233
root@rv1126b-buildroot:/# cansend can0 123#DEADBEEF11122233

```

```
root@rv1126b-buildroot:/# candump can1
can1 000 [8] 00 01 02 03 04 05 06 07
can1 000 [8] 00 01 02 03 04 05 06 07
can1 000 [8] 00 01 02 03 04 05 06 07
can1 000 [8] 00 01 02 03 04 05 06 07
can1 000 [8] 00 01 02 03 04 05 06 07
can1 000 [8] 00 01 02 03 04 05 06 07
can1 000 [8] 00 01 02 03 04 05 06 07
can1 000 [8] 00 01 02 03 04 05 06 07
^Croot@rv1126b-buildroot:/# ^C
root@rv1126b-buildroot:/# ^C
root@rv1126b-buildroot:/# ^C
root@rv1126b-buildroot:/# cansend can0 123#DEADBEEF11122233
root@rv1126b-buildroot:/# cansend can1 123#DEADBEEF11122233
root@rv1126b-buildroot:/# cansend can1 123#DEADBEEF11122233
root@rv1126b-buildroot:/# cansend can1 123#DEADBEEF11122233
root@rv1126b-buildroot:/# cansend can1 123#DEADBEEF11122233
root@rv1126b-buildroot:/# cansend can1 123#DEADBEEF11122233
root@rv1126b-buildroot:/# cansend can1 123#DEADBEEF11122233
root@rv1126b-buildroot:/# cansend can1 123#DEADBEEF11122233
root@rv1126b-buildroot:/# cansend can1 123#DEADBEEF11122233
```

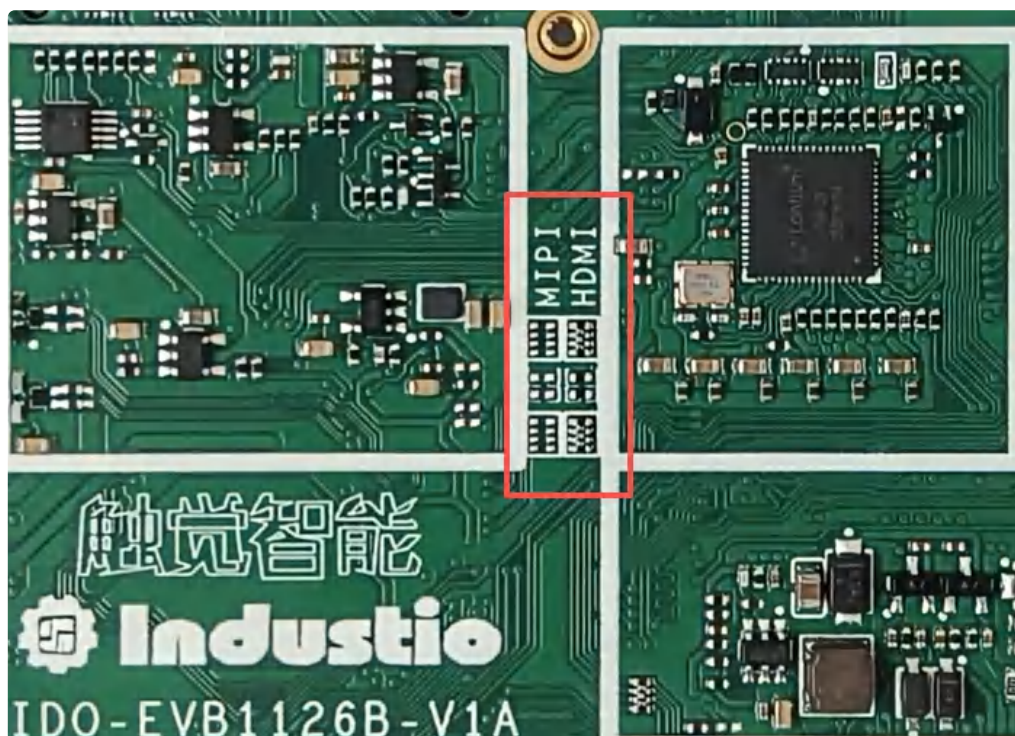
注意： 测试需要使用转接板。

3.11. LCD显示

3.11.1. HDMI-TX

注意： HDMI-TX 是由 MIPI-TX 接口 + 信号转换芯片，二选一，硬件默认配置为 HDMI-TX。

HDMI-TX硬件确认，如下图所示：



3.11.2. MIPI-TX

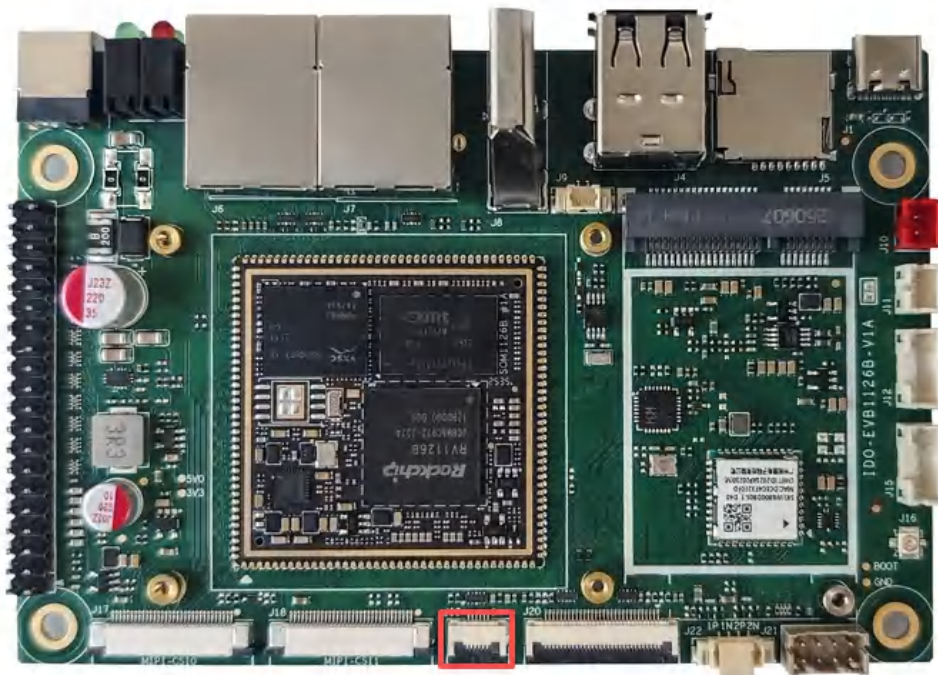
注意： MIPI-TX 与 HDMI 为复用关系，二选一，硬件默认配置为 HDMI。

要使用mipi的话，需要将下图中HDMI 列电阻移动至MIPI 列



3.12. TP

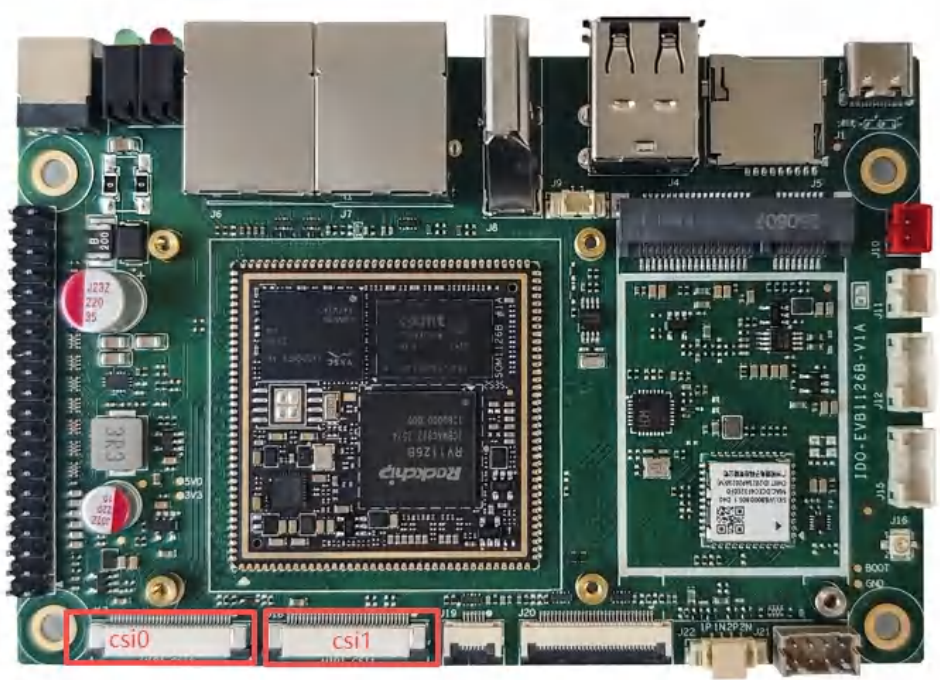
I2C-TP接口为 6Pin FPC 0.5mm座子 (J13)，如下图所示：



- 触摸 TP 接口接线方法为下接。

3.13. MIPI Camera

主板内置两路MIPI Camer接口J17和J18，使用FPC05-PUW-32P抽拉式上接座子，如下图所示：



3.13.1. IMX415

J18 默认适配IMX415作为后置摄像头，接法如下图所示：



J17	J18

/dev/video22	/dev/video30
--------------	--------------



3.13.1.1. 预览

▼

Bash

```
1  gst-launch-1.0 v4l2src device=/dev/video22 ! video/x-raw, format=NV12, width=1920, height=1080, framerate=30/1 ! videoconvert ! autovideosink
2
3  gst-launch-1.0 v4l2src device=/dev/video30 ! video/x-raw, format=NV12, width=1920, height=1080, framerate=30/1 ! videoconvert ! autovideosink
```

3.13.1.2. 抓图

▼

Bash

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

3.13.1.3. 抓视频

▼

Bash

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

3.13.2. TP2815 AHD



J17	J18
/dev/video0	/dev/video11

/dev/video1	/dev/video12
/dev/video2	/dev/video13
/dev/video3	/dev/video14

预览

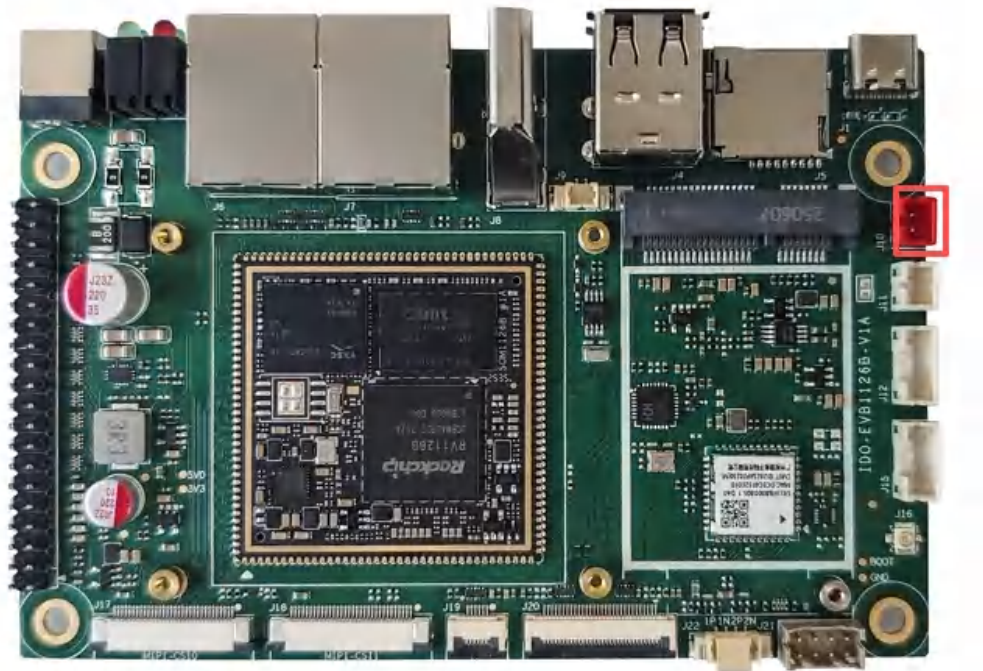
▼

Bash

```
1  gst-launch-1.0 v4l2src device=/dev/video0 ! video/x-raw, format=NV12, width=128
0, height=720, framerate=25/1 ! videoconvert ! autovideosink
2
3
4  gst-launch-1.0 v4l2src device=/dev/video1 ! video/x-raw, format=NV12, width=128
0, height=720, framerate=25/1 ! videoconvert ! autovideosink
5
6
7      gst-launch-1.0 v4l2src device=/dev/video2 ! video/x-raw, format=NV12, width=1
280, height=720, framerate=25/1 ! videoconvert ! autovideosink
8
9
10     gst-launch-1.0 v4l2src device=/dev/video3 ! video/x-raw, format=NV12, width=1
280, height=720, framerate=25/1 ! videoconvert ! autovideosink
11
12
13     gst-launch-1.0 v4l2src device=/dev/video11 ! video/x-raw, format=NV12, width=
1280, height=720, framerate=25/1 ! videoconvert ! autovideosink
14
15     gst-launch-1.0 v4l2src device=/dev/video12 ! video/x-raw, format=NV12, width=
1280, height=720, framerate=25/1 ! videoconvert ! autovideosink
16
17
18     gst-launch-1.0 v4l2src device=/dev/video13 ! video/x-raw, format=NV12, width=
1280, height=720, framerate=25/1 ! videoconvert ! autovideosink
19
20
21     gst-launch-1.0 v4l2src device=/dev/video14 ! video/x-raw, format=NV12, width=
1280, height=720, framerate=25/1 ! videoconvert ! autovideosink
22
```

3.14. FAN 风扇

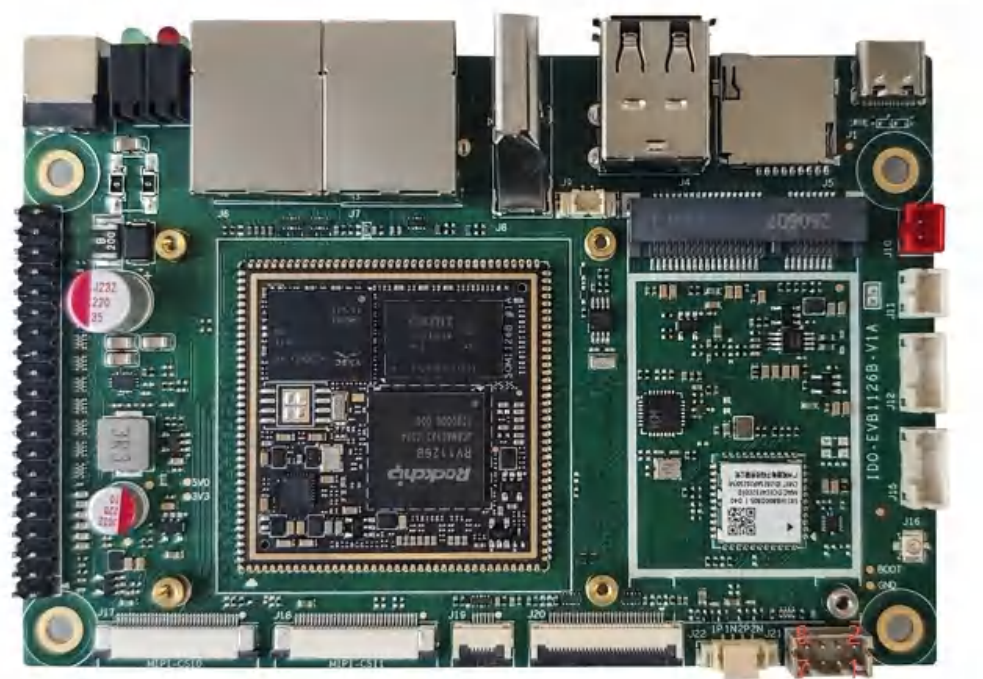
PH2 5V FAN风扇接口J3，如下图所示：

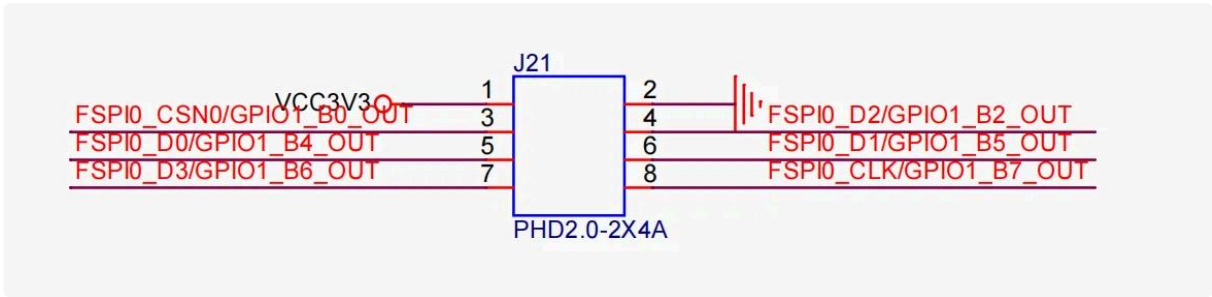


控制功能	控制命令
打开风扇	<code>echo 1 > /sys/class/leds/fan_led/brightness</code>
关闭风扇	<code>echo 0 > /sys/class/leds/fan_led/brightness</code>

3.15. GPIO

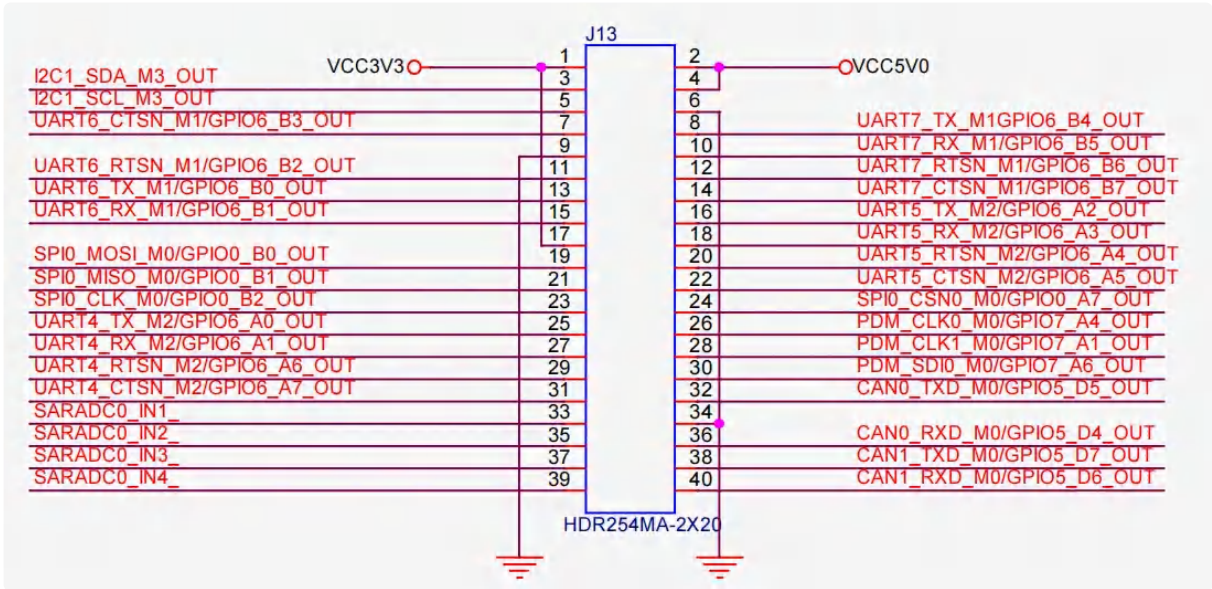
主板预留了8个GPIO接口J3，如下图所示：





GPIO引脚定义描述，如下图所示：

序号	管脚标号	GPIO 序号
1	VCC3V3	/
2	GND	/
3	GPIO1_B0	40
4		42
5	GPIO1_B4	44
6	GPIO1_B5	45
7	GPIO1_B6	46
8	GPIO1_B7	47



序号	管脚标号	GPIO 序号
7	GPIO6_B3	203
11	GPIO6_B2	202
19	GPIO0_B0	8

21	GPIO0_B1	9
23	GPIO0_B2	10
29	GPIO6_A6	198
31	GPIO6_A7	199
12	GPIO6_B6	206
14	GPIO6_B7	207
22	GPIO6_A5	197
24	GPIO0_A7	7
26	GPIO7_A4	228
28	GPIO7_A1	225
30	GPIO7_A6	230

执行以下脚本，使用万用表实际去量电压值，是否是高电平

```

test.sh
1  #!/bin/bash
2
3  GPIO_NUM=(40 42 44 45 46 47 203 202 8 9 10 198 199 206 207 197 7 228 225 230)
4
5  for gpio in "${GPIO_NUM[@]}"; do
6      echo "$gpio" > /sys/class/gpio/export 2>/dev/null
7      if [ $? -eq 0 ]; then
8          echo "GPIO $gpio exported successfully."
9          echo out > /sys/class/gpio/gpio$gpio/direction
10         echo 1 > /sys/class/gpio/gpio$gpio/value
11         echo "GPIO $gpio Out High"
12     else
13         echo "GPIO $gpio already exported or failed to export."
14     fi
15     sleep 0.1
16 done

```

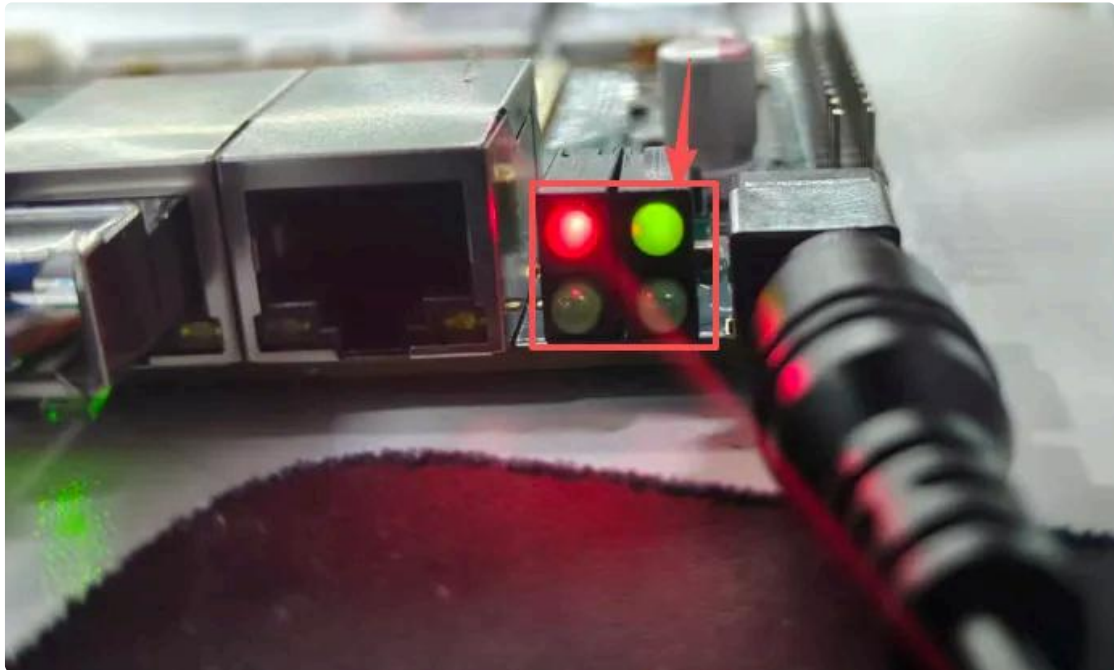
执行以下脚本，使用万用表实际去量电压值，是否是低电平

```

test.sh
1  #!/bin/bash
2
3  GPIO_NUM=(40 42 44 45 46 47 203 202 8 9 10 198 199 206 207 197 7 228 225 230)
4
5  for gpio in "${GPIO_NUM[@]}"; do
6      echo 0 > /sys/class/gpio/gpio$gpio/value
7      echo "GPIO $gpio Out Low"
8
9  done

```

3.16. pwm



pwm灯测试:

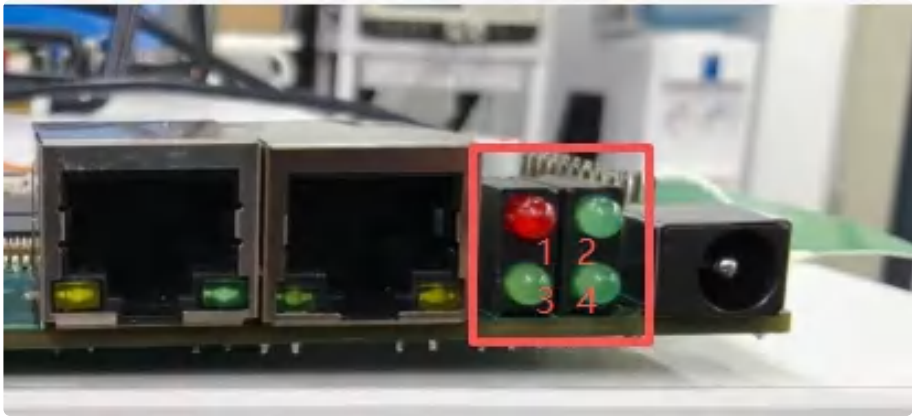
HDMI的 版本镜像pwm节点是 pwmchip1

```
▼ Bash |
1  cd /sys/class/pwm/pwmchip1/
2  echo 0 > export
3  cd pwm0
4  echo 10000 > period
5  echo 5000 > duty_cycle
6  echo normal > polarity
7  echo 1 > enable
```

MIPI 的版本镜像 pwm节点是 pwmchip2

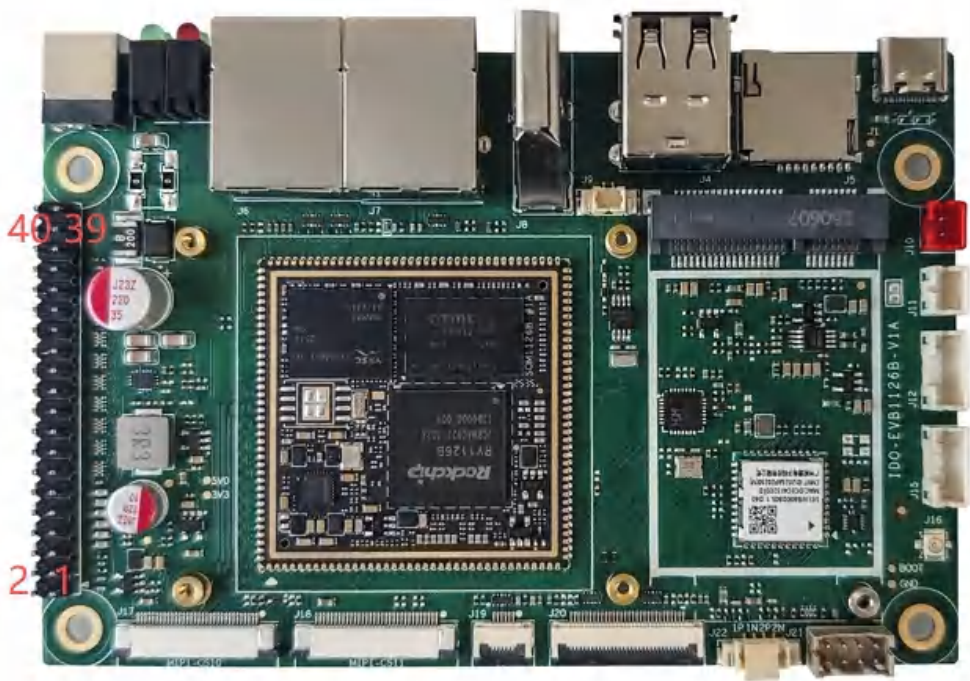
```
▼ Bash |
1  cd /sys/class/pwm/pwmchip2/
2  echo 0 > export
3  cd pwm0
4  echo 10000 > period
5  echo 5000 > duty_cycle
6  echo normal > polarity
7  echo 1 > enable
```

3.17. LED



- 1. 电源指示灯
- 2. PWM-LED 灯
- 3. 系统运行指示灯
- 4. 4G模块指示灯

3.18. ADC



序号	ADC通道	读值
33	SARADC0_IN1_	cat /sys/bus/iio/devices/iio\:device0/in_voltage1_raw
35	SARADC0_IN2_	cat /sys/bus/iio/devices/iio\:device0/in_voltage2_raw
37	SARADC0_IN3_	cat /sys/bus/iio/devices/iio\:device0/in_voltage3_raw
39	SARADC0_IN4_	cat /sys/bus/iio/devices/iio\:device0/in_voltage4_raw

4路ADC精度为13位，参考电压为1.8V。

ADC输入电压计算公式：

$$V=(\text{raw}/8191)*1.8\text{v}$$

3.19. I2C

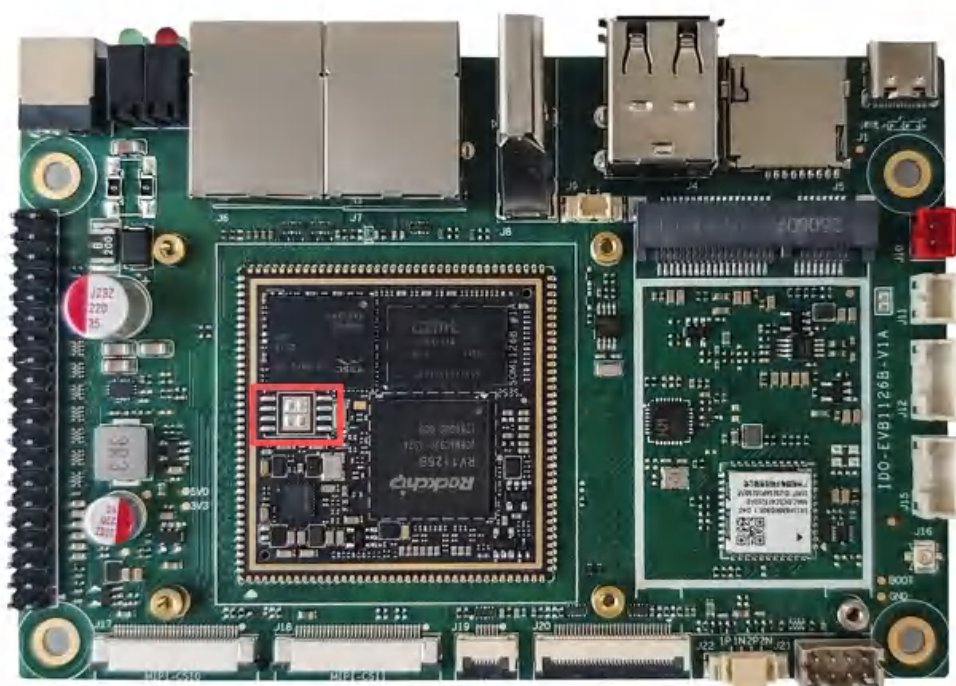
```

1 root@linaro-alip:/# i2cdetect -y -r 1
2      0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
3 00:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
4 10:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
5 20:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
6 30:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
7 40:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
8 50:  -- 51 --  --  --  --  --  --  --  --  --  --  --  --  --  --
9 60:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
10 70:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
11 root@linaro-alip:/#

```

注意： 测试需要使用转接板。

3.20. SPI NAND



```

1 root@rv1126b-buildroot:/# dmesg | grep spi
2 [ 2.847504] spi-nand spi2.0: Winbond SPI NAND was found.
3 [ 2.847513] spi-nand spi2.0: 512 MiB, block size: 128 KiB, page size: 2048, 00
  B size: 128
4 [ 3.653787] mtdblock: MTD device 'spi-nand0' is NAND, please consider using UB
  I block devices instead.
5 [ 4.464264] mtdblock: MTD device 'spi-nand0' is NAND, please consider using UB
  I block devices instead.
6 root@rv1126b-buildroot:/#

```

3.21. NPU

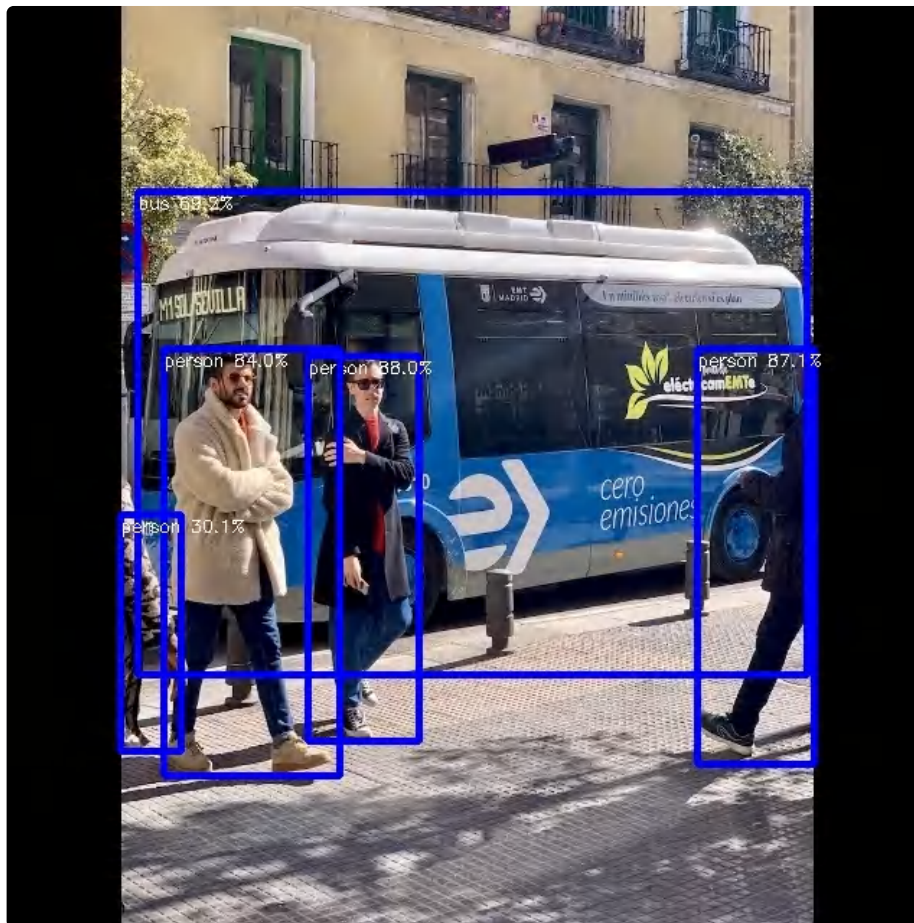
3.21.1. yolov5测试:

3.21.1.1. rknn_yolov5_demo:

通过网盘分享的文件: rknn_yolov5_demo_Linux.zip

链接: <https://pan.baidu.com/s/1N-vgqd1eH1Xcd7zxpfhZ4g> 提取码: hnpm

执行完后会得到转换后的模型图片, 如下所示:



3.21.1.2. rknn_yolov5_video_demo:

```
▼ Shell
1 root@rv1126b-buildroot:/rknn_yolov5_demo_Linux# ./rknn_yolov5_video_demo ./model/R
  V1126B/yolov5s-640-640.rknn ./1280x720.h264 264
2
```

最终生成 out.h264

Shell

```
1 # 播放测试
2 gst-play-1.0 out.h264
```



3.21.2. yolo11 测试

测试 摄像头+yolo11

yolo11_videocapture_demo , yolo11_videocapture_demo_zero_copy

摄像头节点 22, 30

Bash

```
1 # 22 为 /dev/video22 也可改成 30
2 yolo11_videocapture_demo /vendor/etc/firmware/yolo11n.rknn 22
3 # 或者
4 yolo11_videocapture_demo_zero_copy /vendor/etc/firmware/yolo11n.rknn 22
```