

IDO-EVB3506-V1 Linux使用手册



IDO-EVB3506-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	修订内容	修订	审核	日期
V1.0	创建文档	MHK	IDO	2025/04/01
V1.1	更新文档内容	MHK	IDO	2025/04/10
V1.2	更新ble部分	WJY	IDO	2025/04/10

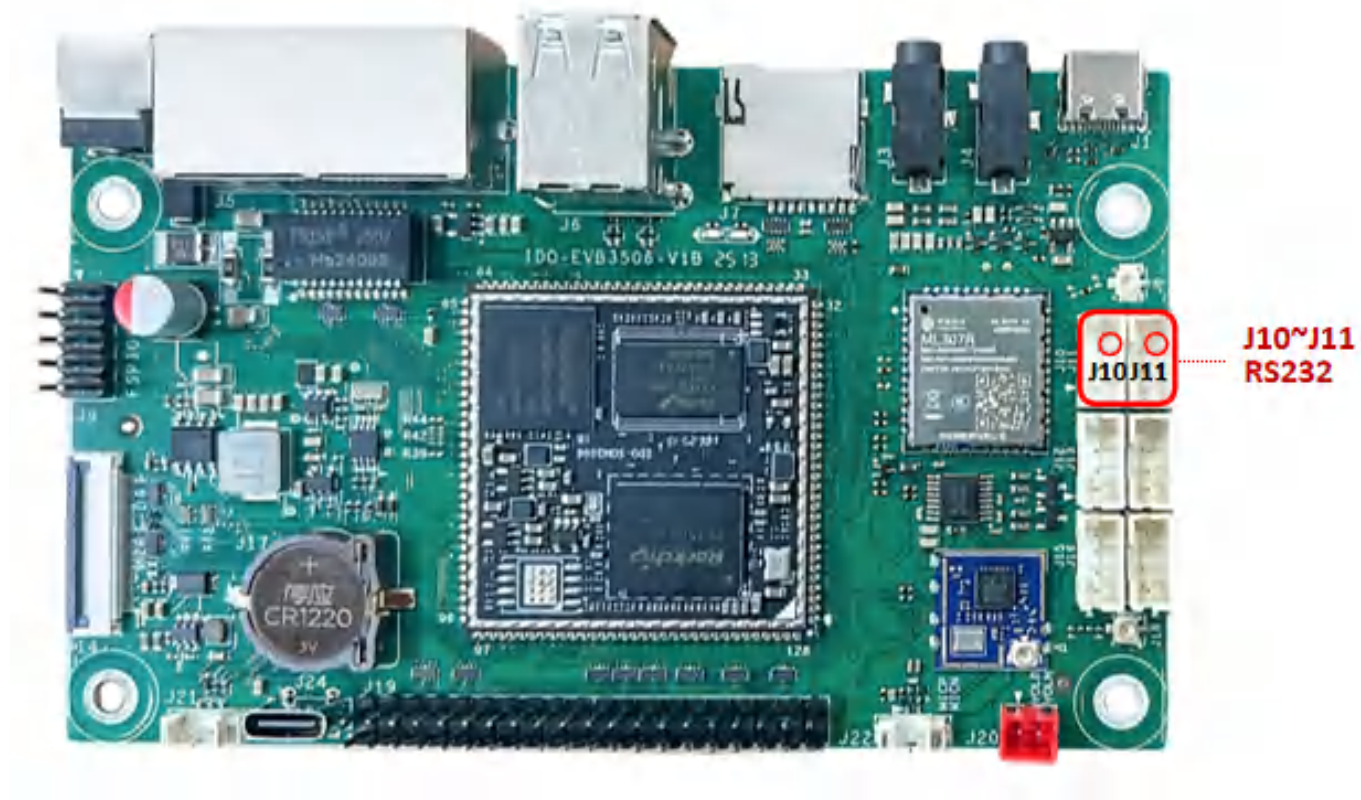
Buildroot使用手册

buildroot系统账号/密码：root/123456

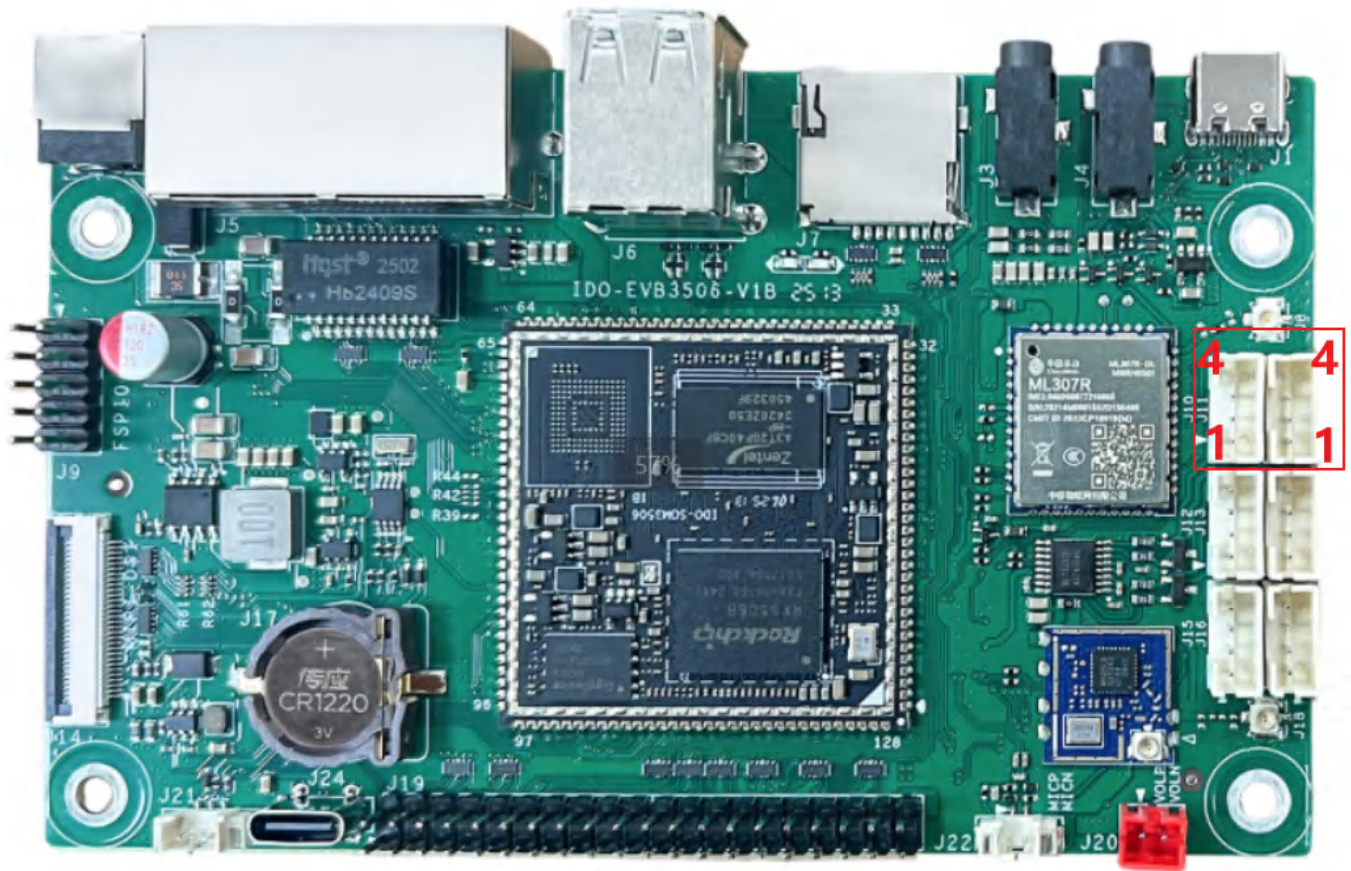
1 UART

主板共配置4路串口（除调试串口），如下所示。

1.1 RS232



序号	接口位置	电平类型	串口设备节点	备注
1	J10	RS232	/dev/ttyS1	
2	J11	RS232	/dev/ttyS2	此功能默认不通(引脚和RGB 屏幕冲突)



引脚定义：

序号	定义	电平/V	说明
1	VCC	5V	5V
2	RXD	/	RXD
3	TXD	/	TXD
4	GND	GND	电源地

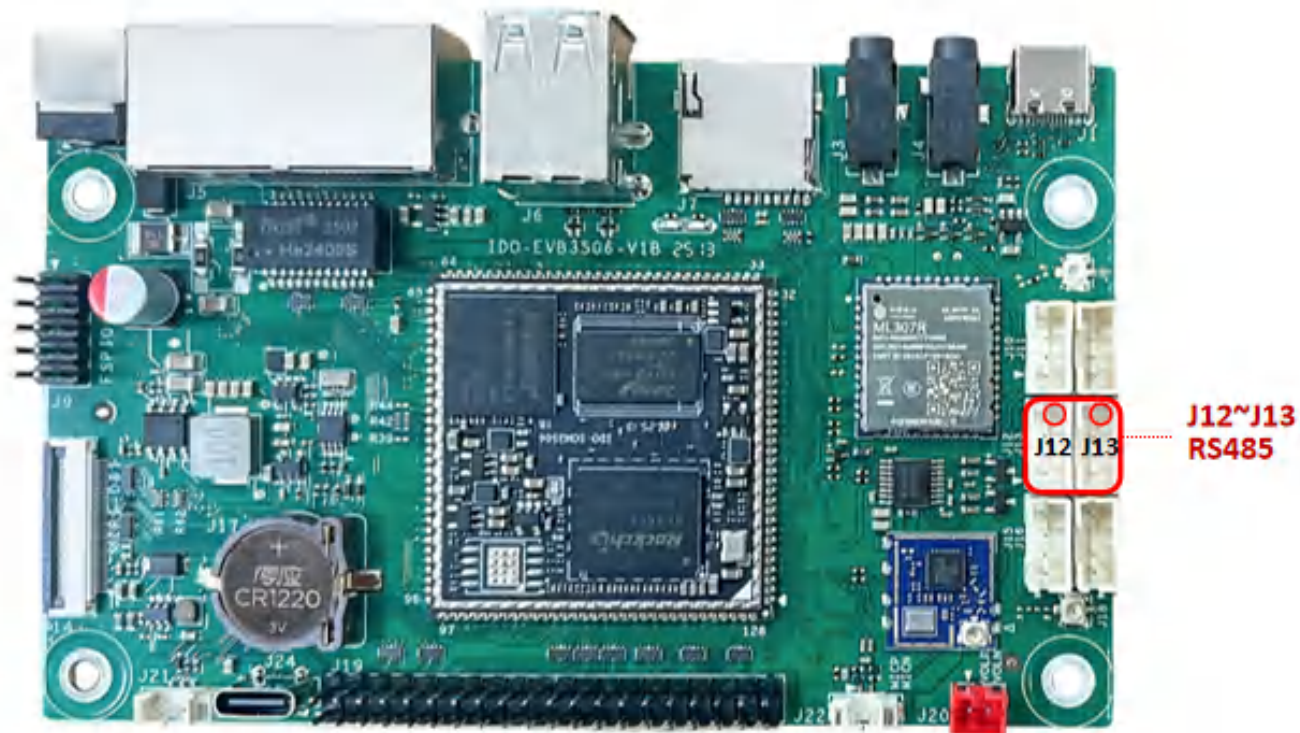
使用microcom可以进行串口收发测试，命令如下：

▼

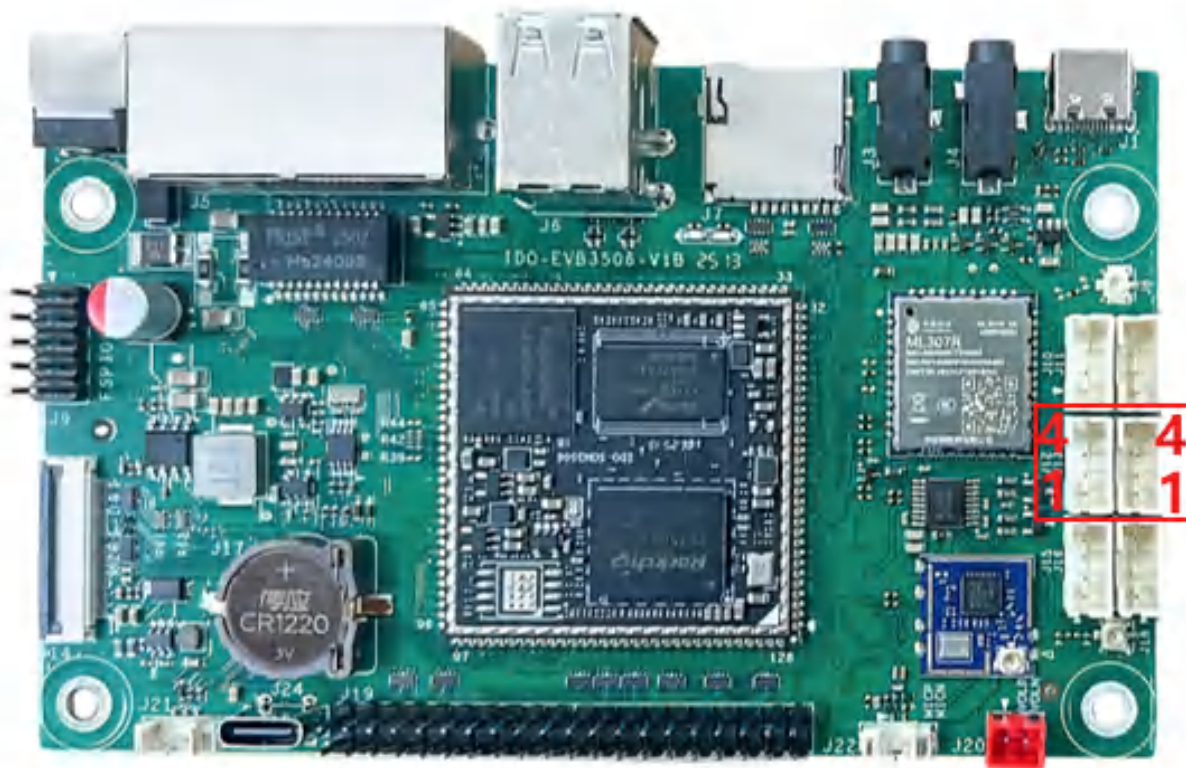
Shell

```
1 root@rk3506-buildroot:/# microcom -s 115200 -p /dev/ttyS1
```

1.2 RS485



序号	接口位置	电平类型	串口设备节点	备注
1	J12	RS485	/dev/ttyS3	/
2	J13	RS485	/dev/ttyS4	此功能默认不通(引脚和RGB 屏幕冲突)



引脚定义：

序号	定义	电平/V	说明
1	VCC	5V	5V
2	RXD	/	RXD
3	TXD	/	TXD
4	GND	GND	电源地

使用microcom可以进行串口收发测试，命令如下：

▼

Shell

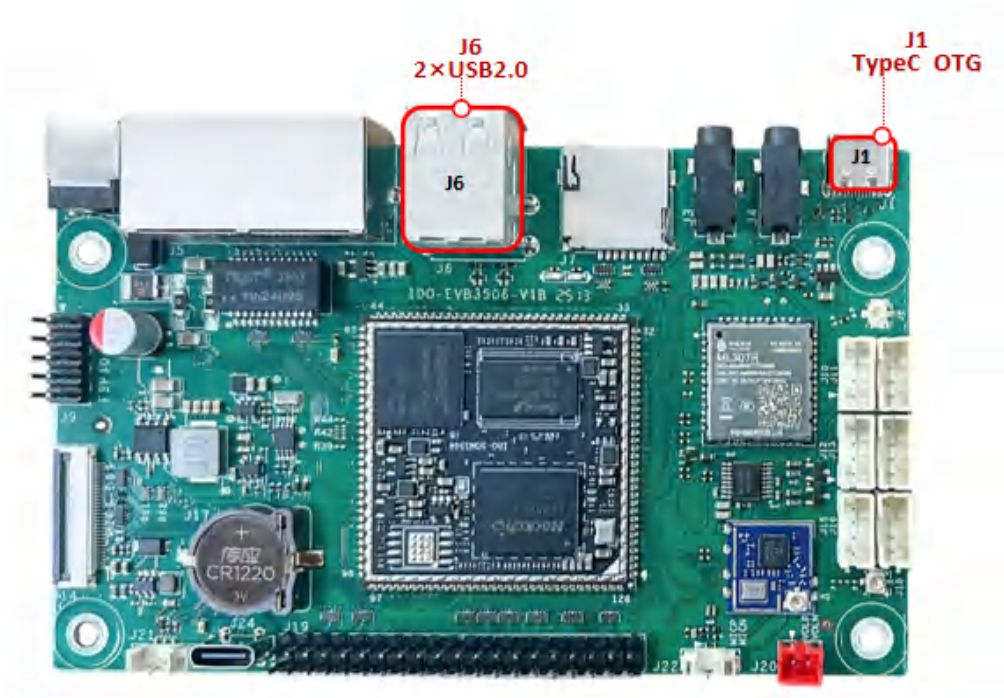
```
1 root@rk3506-buildroot:/# microcom -s 115200 /dev/ttyS3
```

注意：

1. microcom测试完成，按Ctrl+x退出。
2. /dev/ttyS2,/dev/ttyS4 默认不通，此功能需要修改软硬件。

2 USB

主板共配置2路USB座子，如下图所示：



序号	座子位置	类型
USB1	J1	OTG
USB2	J6-上	USB2.0 HOST
USB3	J6-下	USB2.0 HOST

其中USB1接电脑上电时，默认为device模式（可使用adb调试）；接U盘或者其他存储设备上电时，默认为host模式（可读取U盘）。

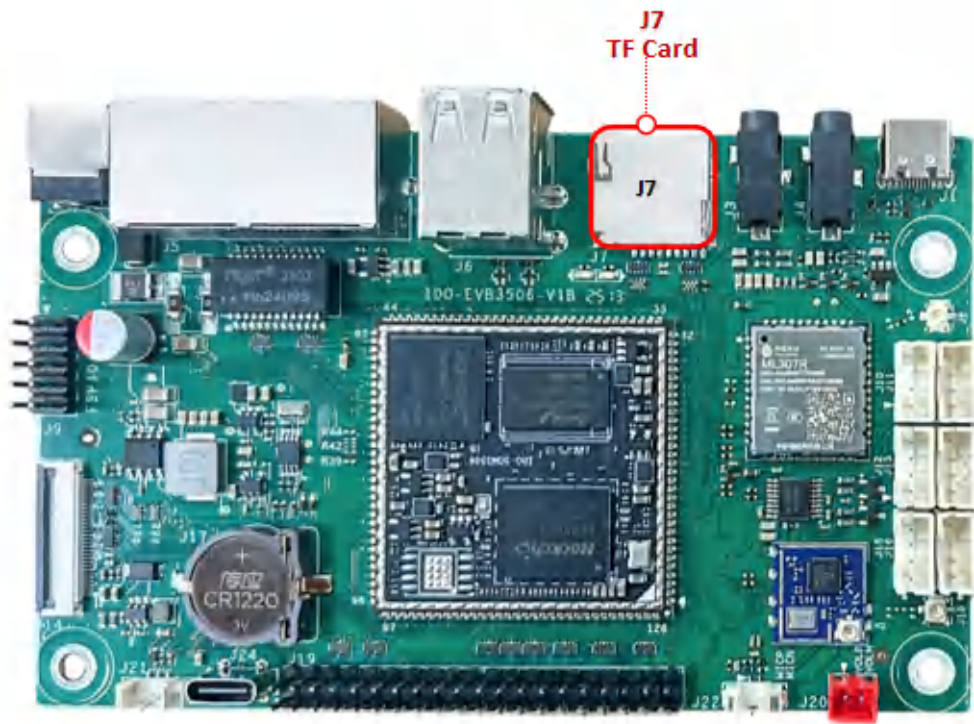
当usb host接上U盘等存储设备时，默认挂载到/mnt/udisk目录，命令如下：

▼Shell

```
1 root@rk3506-buildroot:/# mount
2 ...
3 /dev/sda1 on /mnt/udisk type vfat (rw,nodew,noexec,noatime,nodiratime,fmask
=0022,dmask=0022,codepage=437,iocharset=iso8859-1,shortname=mixed,errors=re
mount-ro)
```

3 Micro SD

主板共配置一路Micro SD接口，如下图所示：



插入SD卡后，默认挂载到/mnt/sdcard目录，如下所示：

▼ Shell

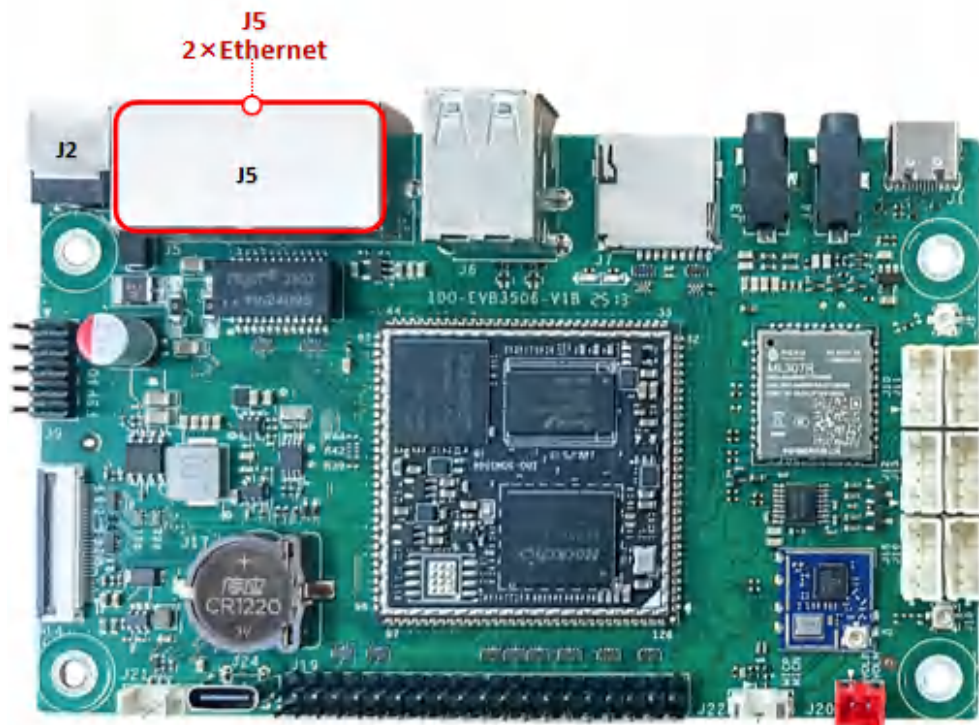
```
1 root@rk3506-buildroot:/# mount
2 ...
3 /dev/mmcblk0p1 on /mnt/sdcard type vfat (rw,nodev,noexec,noatime,nodiratime
,fmask=0022,dmask=0022,codepage=437,iocharset=iso8859-1,shortname=mixed,err
ors=remount-ro)
```

测试项目	要求	测试结果
挂载SD	可以自动挂载fat32格式sd卡	✓
	可以手动挂载NTFS的sd卡	✓

注意： EMMC版本不支持SD卡。

4 Ethernet

主板共配置2路百兆以太网接口，如下图所示：



序号	接口位置	速率	网络节点
1	J5-右	百兆	eth0
1	J5-左	百兆	eth1

系统默认开启DHCP服务，动态获取IP。

▼

Shell

```

1 root@rk3506-buildroot:/# ifconfig eth0
2 eth0      Link encap:Ethernet  HWaddr 7A:1F:F2:40:0F:F0
3           inet addr:192.168.0.12  Bcast:192.168.0.255  Mask:255.255.255.0
4           UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
5           RX packets:25 errors:0 dropped:0 overruns:0 frame:0
6           TX packets:4 errors:0 dropped:0 overruns:0 carrier:0
7           collisions:0 txqueuelen:1000
8           RX bytes:3358 (3.2 KiB)  TX bytes:1368 (1.3 KiB)
9           Interrupt:57 Base address:0xe000

```

可通过ifconfig指令设置临时静态IP，命令如下：

▼

Shell

```

1 root@rk3506-buildroot:/# ifconfig eth0 192.168.3.123

```

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

▼Shell

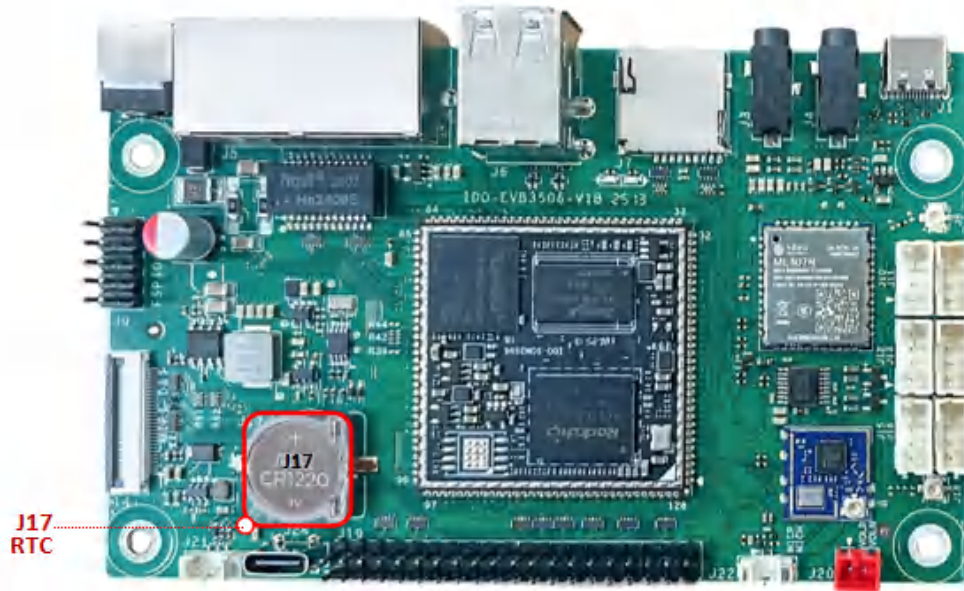
```
1 root@rk3506-buildroot:/# vi /etc/network/interfaces
2 root@rk3506-buildroot:/# cat /etc/network/interfaces
3 # interface file auto-generated by buildroot
4
5 auto lo
6 iface lo inet loopback
7
8 auto eth0
9 iface eth0 inet static
10 address 192.168.3.123
11 netmask 255.255.255.0
12 gateway 192.168.3.1
13
14 source-directory /etc/network/interfaces.d
```

断电重启生效，此静态IP设置仍然生效。

测试项目	要求	测试结果
eth0	识别正常，并成功上网	✓
	自动获取IP地址	✓
	手动修改IP地址	✓
eth1	识别正常，并成功上网	✓
	自动获取IP地址	✓
	手动修改IP地址	✓

5 RTC

主板配置了1路rtc接口，如下图所示：



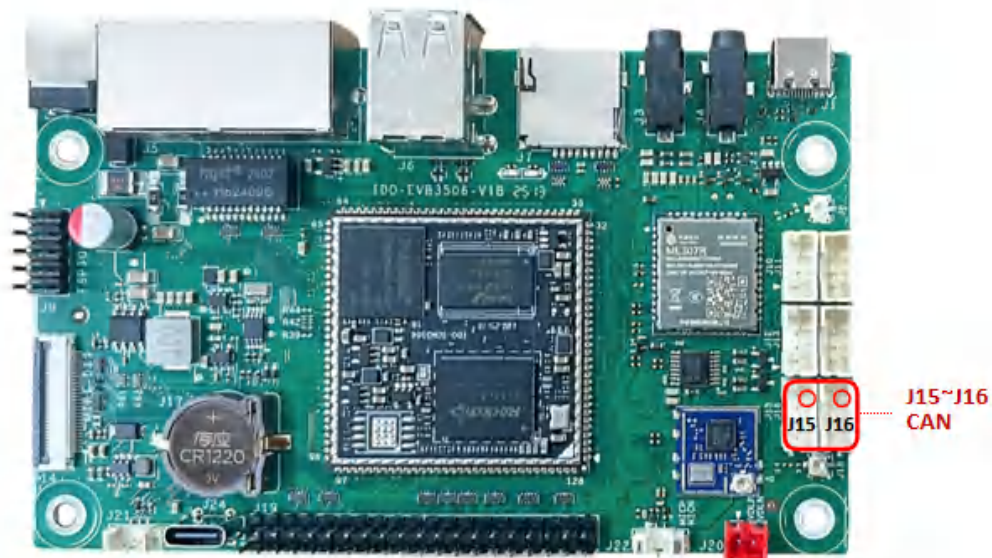
rtc时间设置方法：

Bash

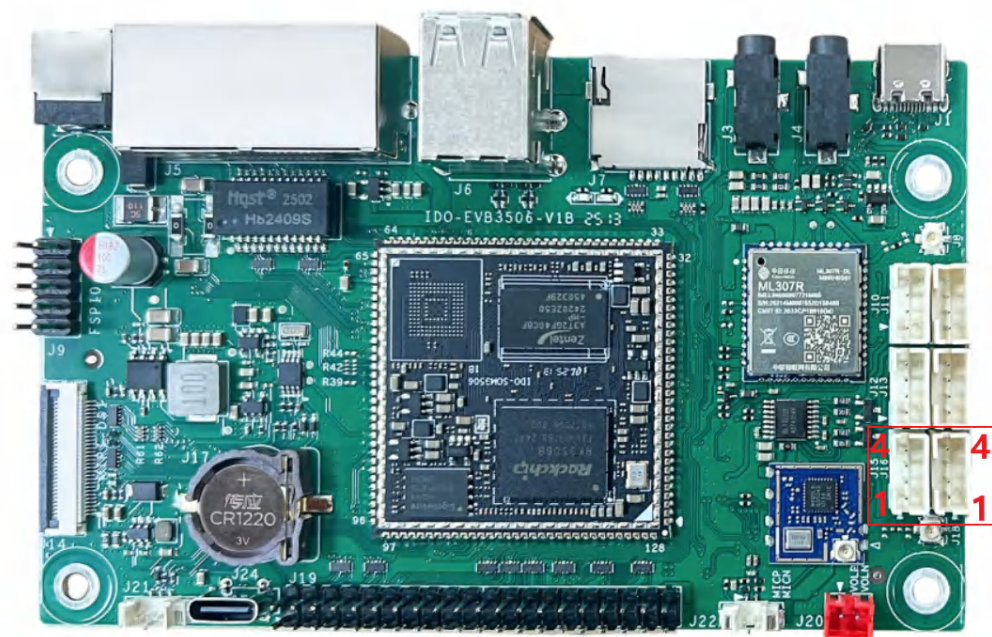
```
1  #同步网络时间
2  root@rk3506-buildroot:/# ntpdate cn.pool.ntp.org
3
4  #设置系统时间
5  root@rk3506-buildroot:/# date -s "2024-10-09 14:02:30"
6
7  #将rtc时间调整为与目前的系统时间一致
8  root@rk3506-buildroot:/# hwclock -w
9
10 #获取硬件rtc当前时间（断电重启读取时间没有太大偏差）
11 root@rk3506-buildroot:/# hwclock -r
12 2024-10-09 14:02:35.945604+00:00
13
14 #将rtc时间设置为系统时间
15 root@rk3506-buildroot:/# hwclock --systohc
```

6 CAN

主板配置了2路can接口，如下图所示：



序号	接口位置	电平类型	串口设备节点	备注
1	J15	CAN	can1	
2	J16	CAN	can0	此功能默认不通(引脚和RGB 屏幕冲突)



引脚定义：

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

查看can节点，以can1为例：

▼

Shell

```

1 root@rk3506-buildroot:/# ifconfig can0
2 can0      Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
3           NOARP  MTU:16  Metric:1
4           RX packets:0 errors:0 dropped:0 overruns:0 frame:0
5           TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
6           collisions:0 txqueuelen:10
7           RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
8           Interrupt:49

```

数据收发，以can1为例：

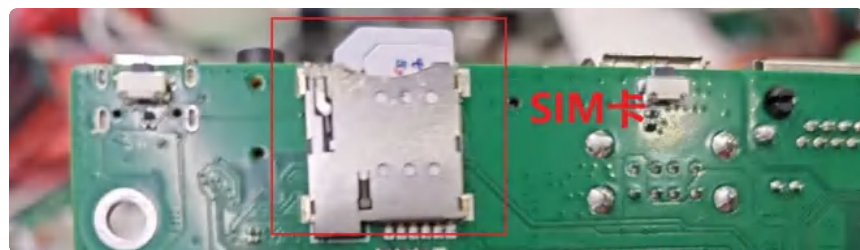
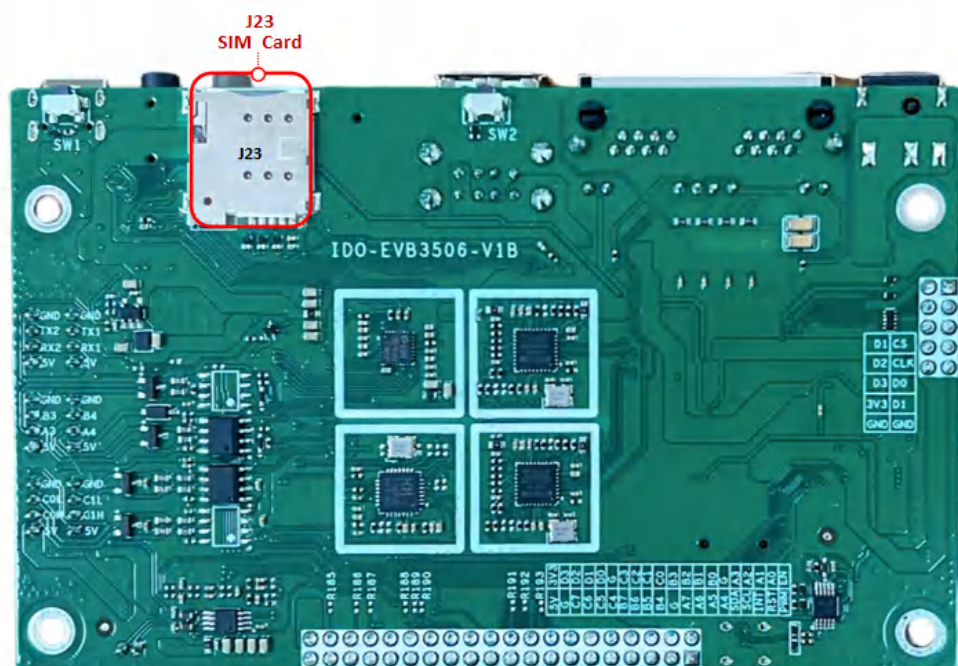
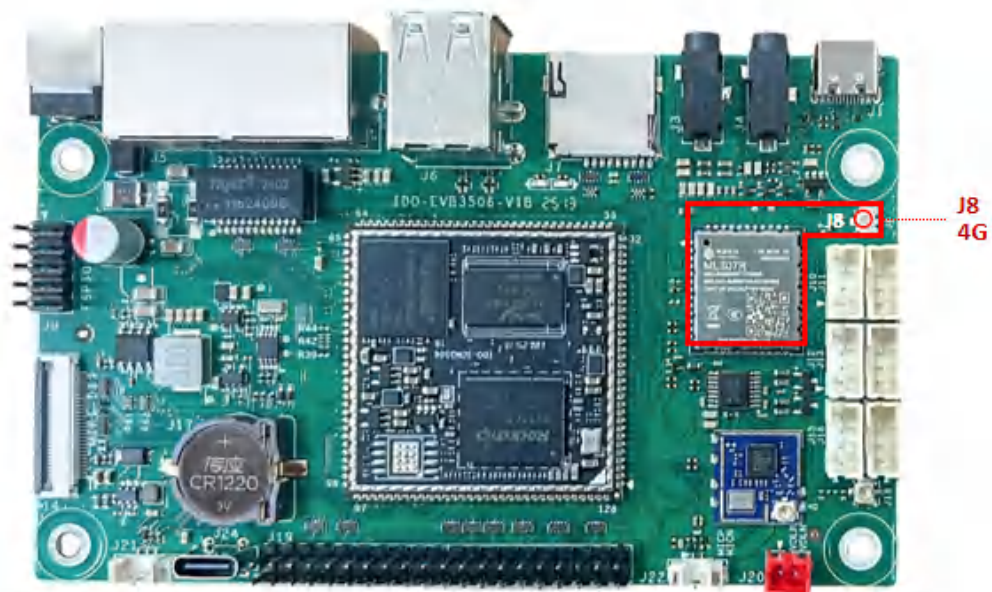
```
1  #检查can设备
2  $ ip link show
3
4  #在收发端关闭can0设备
5  $ ip link set can0 down
6
7  #设置仲裁段1M波特率，数据段1M波特率
8  $ ip link set can0 type can bitrate 1000000 dbitrate 1000000 fd on
9  [ 63.834376] rk3576_canfd ff330000.can can0: bitrate error 0.3%
10 [ 63.834512] rk3576_canfd ff330000.can can0: bitrate error 0.3%
11
12 #在收发端打开can0设备
13 $ ip link set can0 up
14
15 #在接收端执行candump,阻塞等待报文
16 $ candump can0
17
18 #在发送端执行cansend,发送报文
19 $ cansend can0 123#1122334455667788
20
21 #检查是否启用成功
22 $ ip link show can0
```

注意：

- 1.需要can0需要修改硬件。
- 2.内核dts中can0默认关闭，can1默认打开；can1在系统下默认注册为can0。

7 4G模块

主板配置了1路中移4G模块ML307R，如下图所示：



拨号上网：

```

1  cat /dev/ttyUSB2 &
2
3  #设置拨号协议为RNDIS
4  echo -e "AT+MDIALUPCFG="mode",0" > /dev/ttyUSB2
5
6  #设置拨号协议为ECM
7  echo -e "AT+MDIALUPCFG="mode",1" > /dev/ttyUSB2
8
9  #关闭回显
10 echo -e 'ATE0\r\n' > /dev/ttyUSB2
11
12 #查询PDP上下文配置
13 echo -e 'AT+CGDCONT?\r\n' > /dev/ttyUSB2
14
15 #查询驻网状态
16 echo -e 'AT+COPS?\r\n' > /dev/ttyUSB2
17
18 #设置自动拨号命令
19 echo -e 'AT+MDIALUP=1,1\r\n' > /dev/ttyUSB2
20
21 #查询拨号状态
22 echo -e 'AT+MDIALUP?\r\n' > /dev/ttyUSB2

```

查看网络节点：

```

1  root@rk3506-buildroot:/# ifconfig eth2
2  eth2      Link encap:Ethernet  HWaddr AC:0C:29:A3:9B:6D
3           inet addr:192.168.0.100  Bcast:192.168.0.255  Mask:255.255.255.0
4           UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
5           RX packets:4 errors:0 dropped:0 overruns:0 frame:0
6           TX packets:3 errors:0 dropped:0 overruns:0 carrier:0
7           collisions:0 txqueuelen:1000
8           RX bytes:928 (928.0 B)  TX bytes:1026 (1.0 KiB)

```

测试网络

```

1  ping 8.8.8.8 -I eth2

```

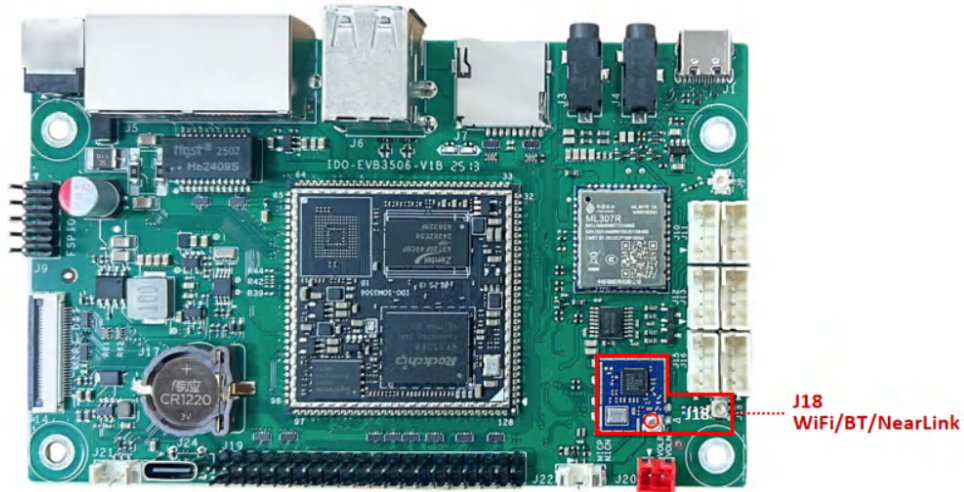
注意：

1.ML307R 4G模块支持RNDIS和ECM模块，两种模式选择其一即可。

2.ML307R 4G模块拨号上网后默认获取局域网内IP地址。

8 星闪模块

主板配置了1路星闪模块，如下图所示：



8.1 SLE

使用两个板子进行对测试。

一台运行：/usr/bin/sle_client_sample。

另一台板子运行：/usr/bin/sle_server_sample。

测试结果如下：

```

uick speed = 707.58 Kbps
t speed = 753.8 Kbps
. speed = 741.92 Kbps
speed = 740.47 Kbps
speed = 747.59 Kbps
mp speed = 666.69 Kbps
eq speed = 672.66 Kbps
eq speed = 710.28 Kbps
speed = 672.31 Kbps
[ 53.189000] [0 96][414]{online_cali_pow_tx_notify::dev
speed = 672.35 Kbps
speed = 689.74 Kbps
speed = 720.58 Kbps
speed = 695.58 Kbps
speed = 677.98 Kbps
speed = 725.32 Kbps
speed = 727.96 Kbps
speed = 639.96 Kbps
speed = 660.83 Kbps
[ 55.349184] [0 96][181][GLA][ALG][TEMP]:cur_state=1
speed = 640.18 Kbps
speed = 677.87 Kbps
speed = 584.6 Kbps
speed = 665.22 Kbps
speed = 677.28 Kbps

```

8.2 BLE

参考IDO-EVB3506-V1 蓝牙开发案例文档。

8.3 WIFI

STA连接方法：

星闪wifi.so加载起来后，会打开Featureid0和wlan0两个网络节点，Featureid0会影响wlan0的ip获取，需要关闭，命令如下。

```

▼ Bash |
1 root@rk3506-buildroot:/# ifconfig Featureid0 down

```

账号密码配置：

```
▼ Bash |
1
2 root@rk3506-buildroot:/# cat /etc/wpa_supplicant.conf
3 ctrl_interface=/var/run/wpa_supplicant
4 ap_scan=1
5 update_config=1
6
7 network={
8     ssid="SSID"           #账号
9     psk="PASSWORD"        #密码
10    key_mgmt=WPA-PSK
11 }
```

ssid: 账号。

psk: 密码。

连接方法:

```
▼ Bash |
1 root@rk3506-buildroot:/# wpa_supplicant -Dnl80211 -c /etc/wpa_supplicant.conf -i wlan0 &
```

获取IP:

```

1 root@rk3506-buildroot:/# udhcpc -i wlan0
2 udhcpc: started, v1.36.1
3 udhcpc: broadcasting discover
4 udhcpc: broadcasting select for 192.168.74.160, server 192.168.74.123
5 udhcpc: lease of 192.168.74.160 obtained from 192.168.74.123, lease time 3
  599
6 deleting routers
7 adding dns 192.168.74.123
8
9 root@rk3506-buildroot:/# ifconfig wlan0
10 wlan0      Link encap:Ethernet  HWaddr 78:22:88:85:63:96
11            inet addr:192.168.74.160  Bcast:192.168.74.255  Mask:255.255.25
  5.0
12            UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
13            RX packets:77 errors:0 dropped:0 overruns:0 frame:0
14            TX packets:94 errors:0 dropped:0 overruns:0 carrier:0
15            collisions:0 txqueuelen:1000
16            RX bytes:6190 (6.0 KiB)  TX bytes:11915 (11.6 KiB)

```

测试:

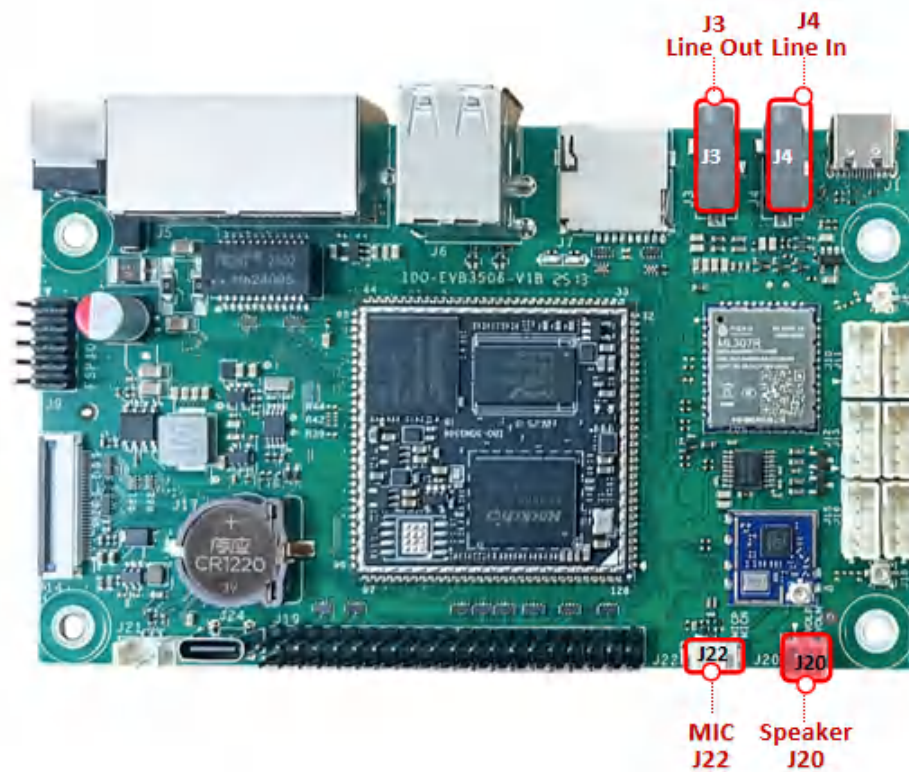
```

1 root@rk3506-buildroot:/# ping www.baidu.com -I wlan0
2 PING www.baidu.com (103.235.46.102): 56 data bytes
3 64 bytes from 103.235.46.102: seq=0 ttl=48 time=266.668 ms
4 64 bytes from 103.235.46.102: seq=1 ttl=48 time=233.898 ms
5 64 bytes from 103.235.46.102: seq=2 ttl=48 time=276.997 ms
6 64 bytes from 103.235.46.102: seq=3 ttl=48 time=229.841 ms

```

9 声卡

主板配置的声卡接口，如下图所示：



序号	接口位置	功能
1	J20	Speaker
2	J3	Line Out
3	J4	Line In
4	J22	MIC

声卡节点

```

1  #声卡节点
2  root@rk3506-buildroot:/# aplay -l
3  **** List of PLAYBACK 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
8  #录音节点
9  root@rk3506-buildroot:/# arecord -l
10 **** List of CAPTURE Hardware Devices ****
11 card 0: rockchip-es8388 [rockchip-es8388], device 0: dailink-multicodecs ES
   8323 HiFi-0 [dailink-multicodecs ES8323 HiFi-0]
12      Subdevices: 1/1
13      Subdevice #0: subdevice #0
14 card 1: rockchip-ahbif [rockchip-ahbif], device 0: ff4a8000.sai-rk3506-hi
   fi rk3506-hifi-0 [ff4a8000.sai-rk3506-hifi rk3506-hifi-0]
15      Subdevices: 1/1
16      Subdevice #0: subdevice #0

```

音频播放

```

1  #音频播放
2  root@rk3506-buildroot:/# aplay /tmp/xihuang
3  Playing WAVE '/tmp/xihuang' : Signed 16 bit Little Endian, Rate 44100 Hz, S
   tereo

```

MIC录音

```

1  #设置音量大小, 0最小, 31最大
2  root@rk3506-buildroot:/# amixer -c 1 cset numid=40 31
3  #录音
4  root@rk3506-buildroot:/# arecord -D hw:1,0 -r 44100 -c 2 -f S16_LE test.wav
5  #播放
6  root@rk3506-buildroot:/# aplay test.wav

```

LINE IN录音

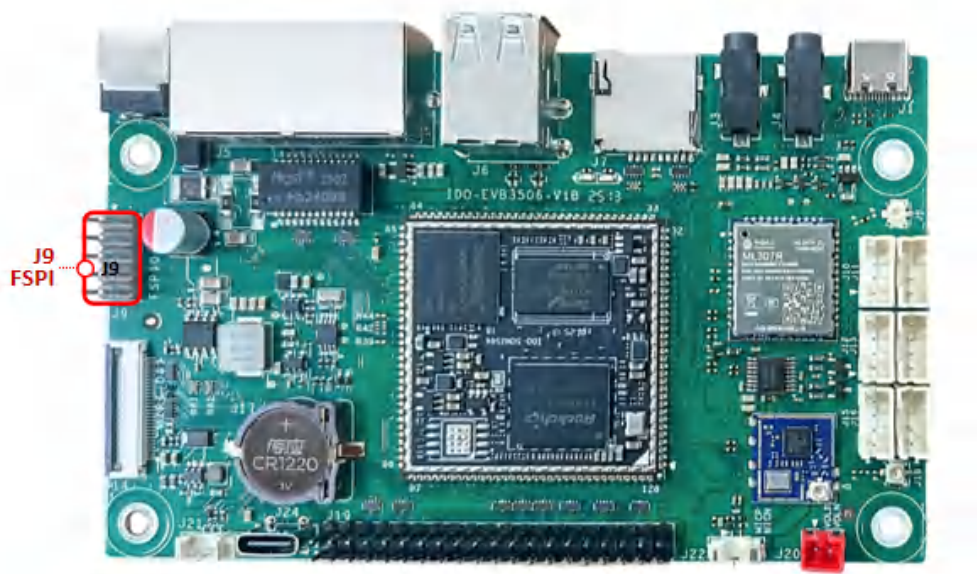
▼

Shell

```
1  #line in录音
2
3  #切换到line in
4  root@rk3506-buildroot:/# echo -n "mic1" > /sys/class/es8388_mic/mic_channel
5
6  root@rk3506-buildroot:/# arecord -D hw:0,0 -f S16_LE -r 44100 -c 1 /tmp/test_mic1.wav
7  Recording WAVE '/tmp/test_mic1.wav' : Signed 16 bit Little Endian, Rate 44100 Hz, Stereo
```

10 扩展接口

主板配置了1路扩展接口，默认做GPIO，如下图所示：



序号	接口位置	GPIO 标号	GPIO 序号
1	J9-1	gpio2_a0	64
2	J9-2	gpio2_a3	67
3	J9-3	gpio2_a1	65

4	J9-4	gpio2_a4	68
5	J9-5	gpio2_a2	66
6	J9-6	gpio2_a5	69

gpio设置：

以gpio2_a3为例：

▼

Shell

```

1  #输出
2  # 设置 gpio2_a3 输出高电平状态
3  gpioset gpiochip2 3=1
4
5  # 设置 gpio2_a3 输出低电平状态
6  gpioset gpiochip2 3=0
7
8  #输入
9  # 获取 gpiochip2 输入电平状态
10 gpioget gpiochip2 3
11

```

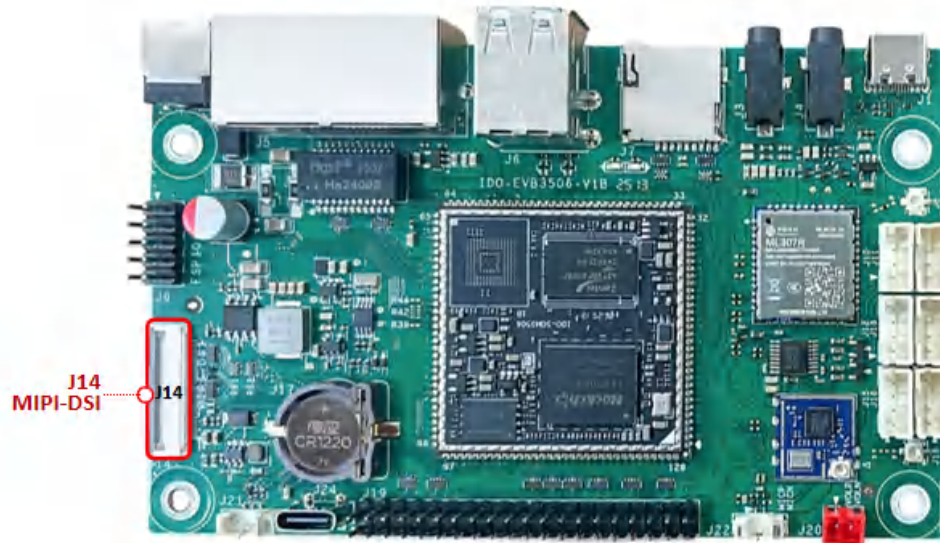
注意：

- 1.nand版本不支持，emmc版本支持。
- 2.emmc版本需要此功能需要修改硬件。

11 显示接口

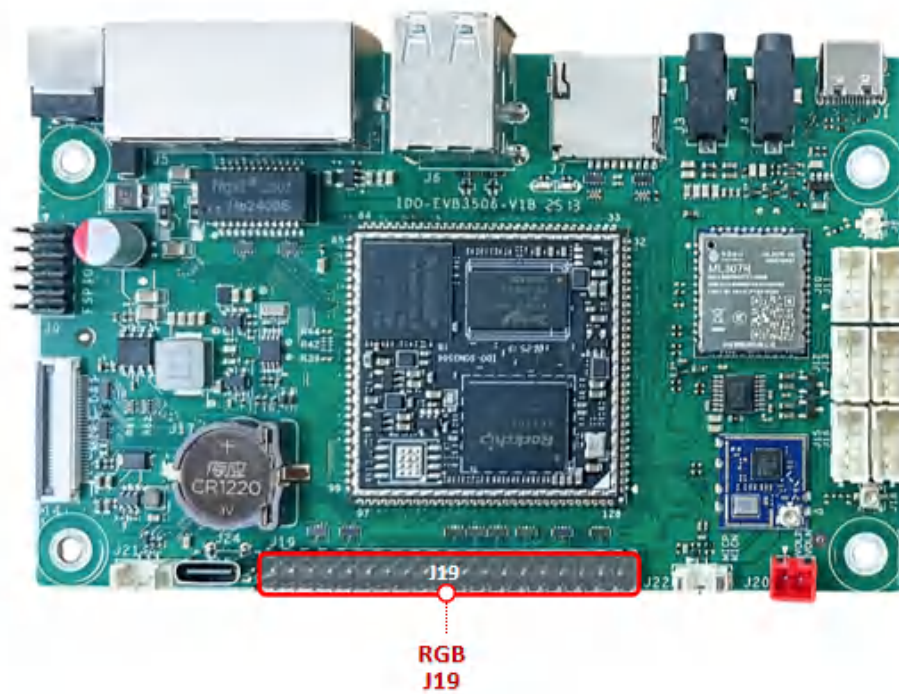
11.1 MIPI

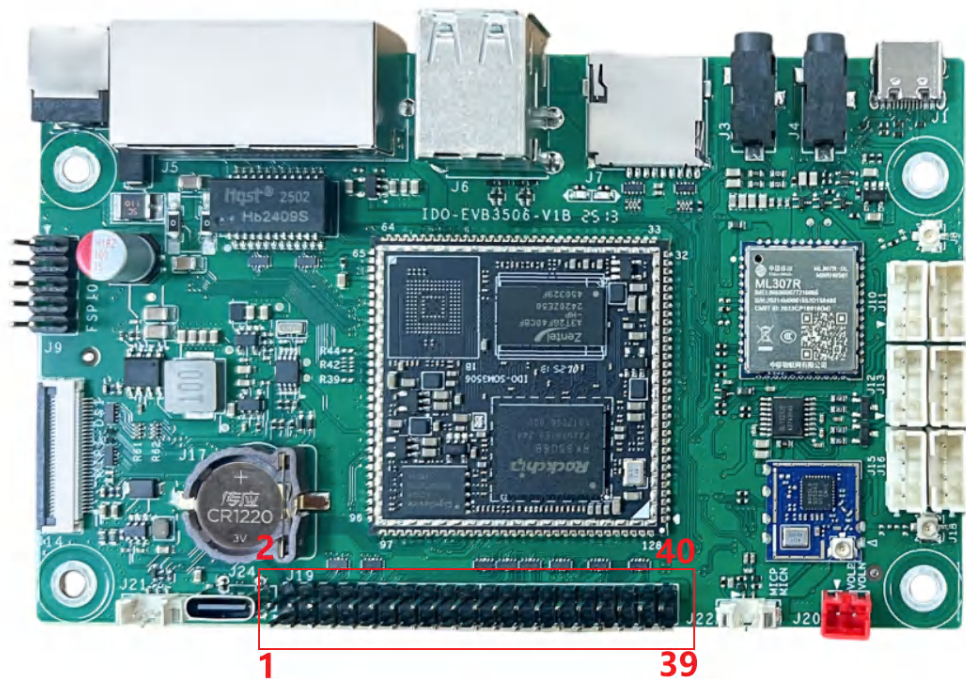
主板配置了1路MIPI接口，如下图所示：



11.2 RGB屏幕

主板配置了1路RGB接口，如下图所示：





引脚定义：

序号	定义	电平/V
1	RGB_PWM	/
2	LCD_PWREN_H_	/
3	TP_RST0	/
4	VO_LCDC_DEN/GPIO1_A0	/
5	TP_INT0	/
6	VO_LCDC_VSYNC/GPIO1_A1	/
7	TP_SCL0	/
8	VO_LCDC_HSYNC/GPIO1_A2	/
9	TP_SDA0	/
10	VO_LCDC_CLK/GPIO1_A3	/
11	VO_LCDC_D23/GPIO1_A4	/
12	GND	GND
13	VO_LCDC_D22/GPIO1_A5	/
14	VO_LCDC_D19/GPIO1_B0	/

15	VO_LCDC_D21/GPIO1_A6	/
16	VO_LCDC_D18/UART2_TX/G PIO1_B1	/
17	VO_LCDC_D20/GPIO1_A7	/
18	VO_LCDC_D17/CAN0_TX/GPI O1_B2	/
19	GND	GND
20	VO_LCDC_D16/CAN0_RX/GP IO1_B3	/
21	VO_LCDC_D15/GPIO1_B4	/
22	VO_LCDC_D11/GPIO1_C0	/
23	VO_LCDC_D14/GPIO1_B5	/
24	VO_LCDC_D10/DSM_AUD_R N/GPIO1_C1	/
25	VO_LCDC_D13/GPIO1_B6	/
26	VO_LCDC_D9/DSM_AUD_RP /GPIO1_C2	/
27	VO_LCDC_D12/GPIO1_B7	/
28	VO_LCDC_D8/UART2_RX/GP IO1_C3	/
29	VO_LCDC_D7/GPIO1_C4	/
30	GND	GND
31	VO_LCDC_D6/GPIO1_C5	/

32	VO_LCDC_D3/GPIO1_D0	/
33	VO_LCDC_D5/GPIO1_C6	/
34	VO_LCDC_D2/UART4_RTSN/ GPIO1_D1	/
35	VO_LCDC_D4/GPIO1_C7	/
36	VO_LCDC_D1/UART4_RX/GPI O1_D2	/
37	GND	GND
38	VO_LCDC_D0/UART4_TX/GP IO1_D3	/
39	VCC5V0	5V
40	VCC3V3	3V3

注意：

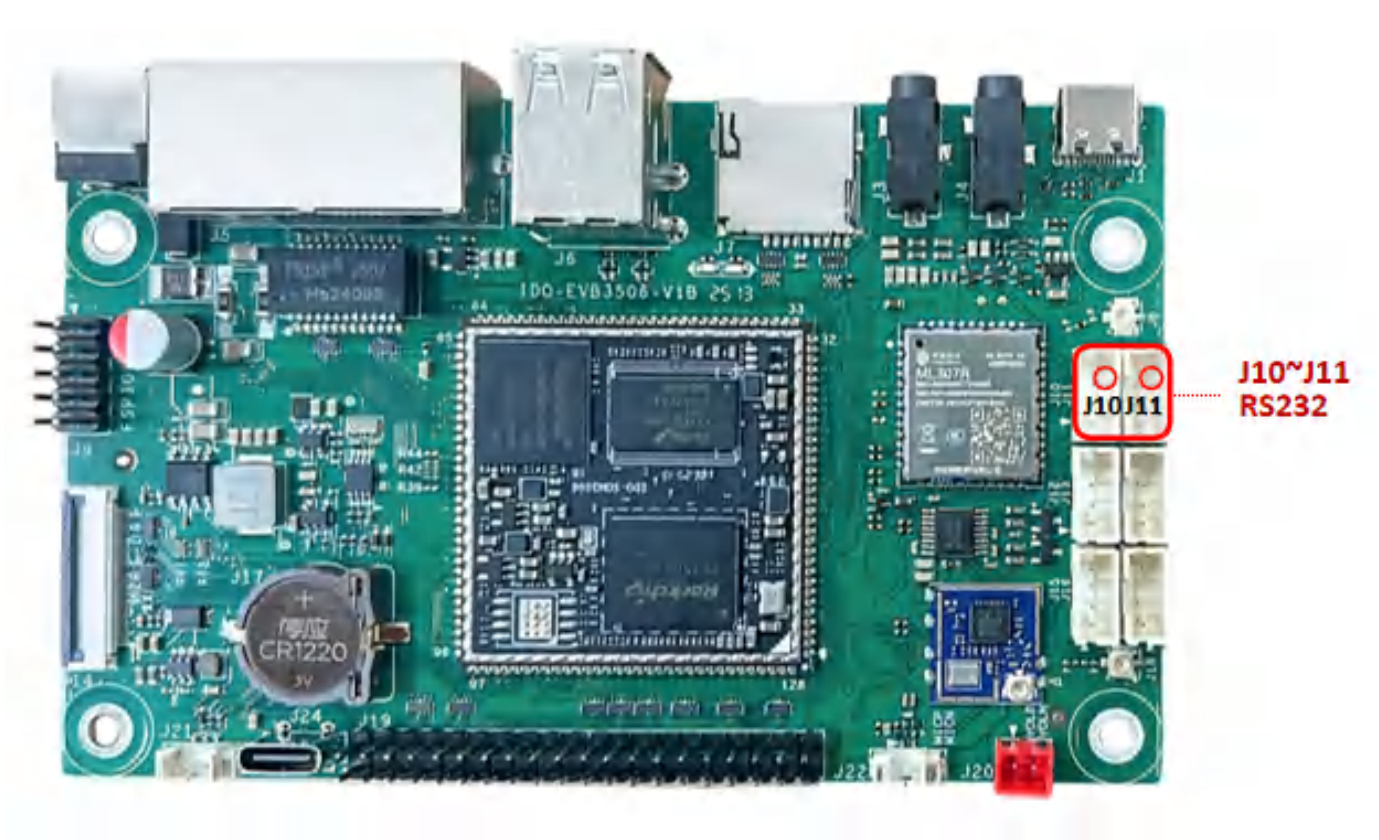
- 1.J19默认配成RGB显示。
- 2.RGB屏幕需要外接转接小板。

Ubuntu使用手册

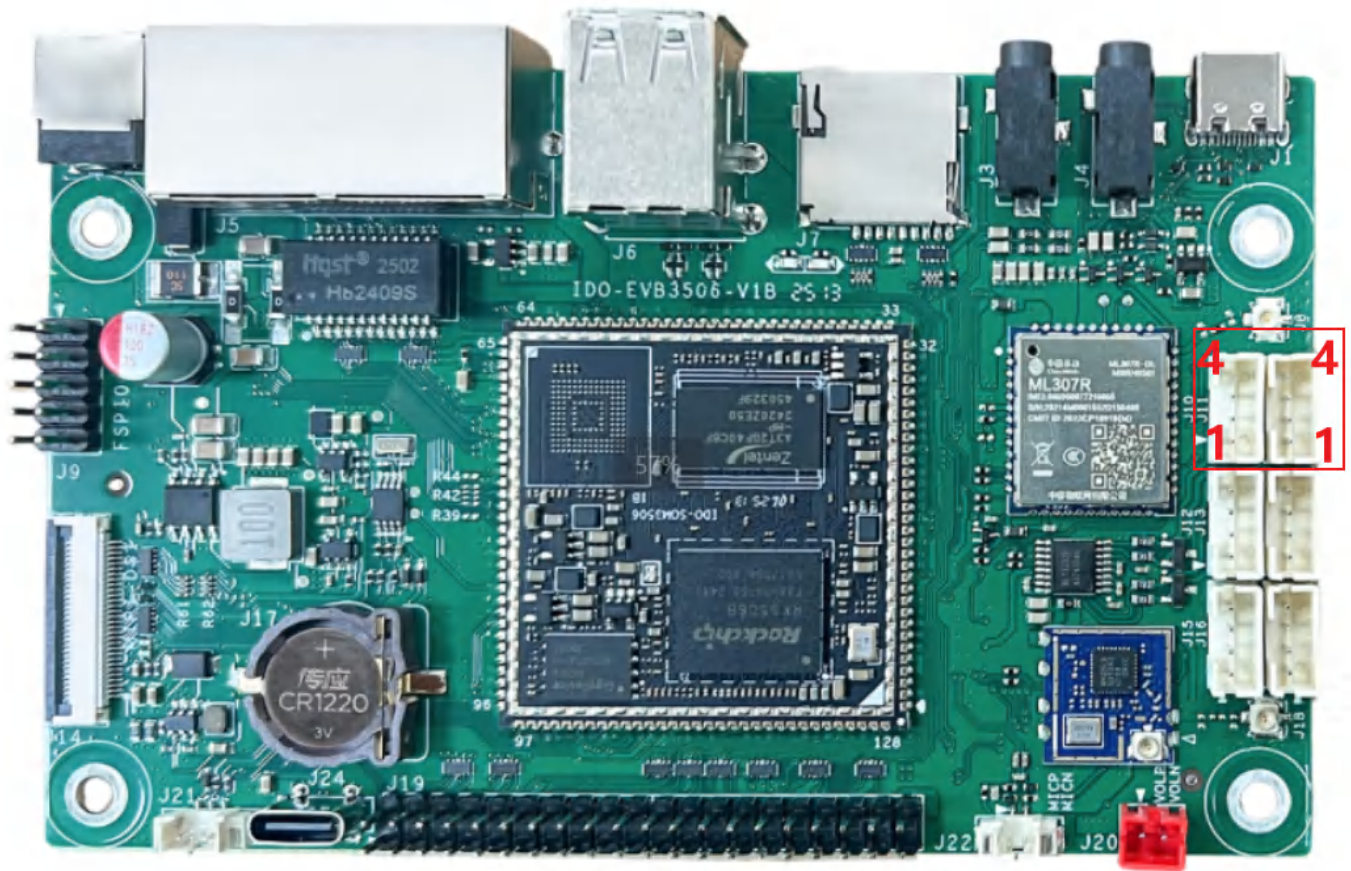
1 UART

主板共配置4路串口（除调试串口），如下所示。

1.1 RS232



序号	接口位置	电平类型	串口设备节点	备注
1	J10	RS232	/dev/ttyS1	
2	J11	RS232	/dev/ttyS2	此功能默认不通(引脚和RGB 屏幕冲突)



引脚定义：

序号	定义	电平/V	说明
1	VCC	5V	5V
2	RXD	/	RXD
3	TXD	/	TXD
4	GND	GND	电源地

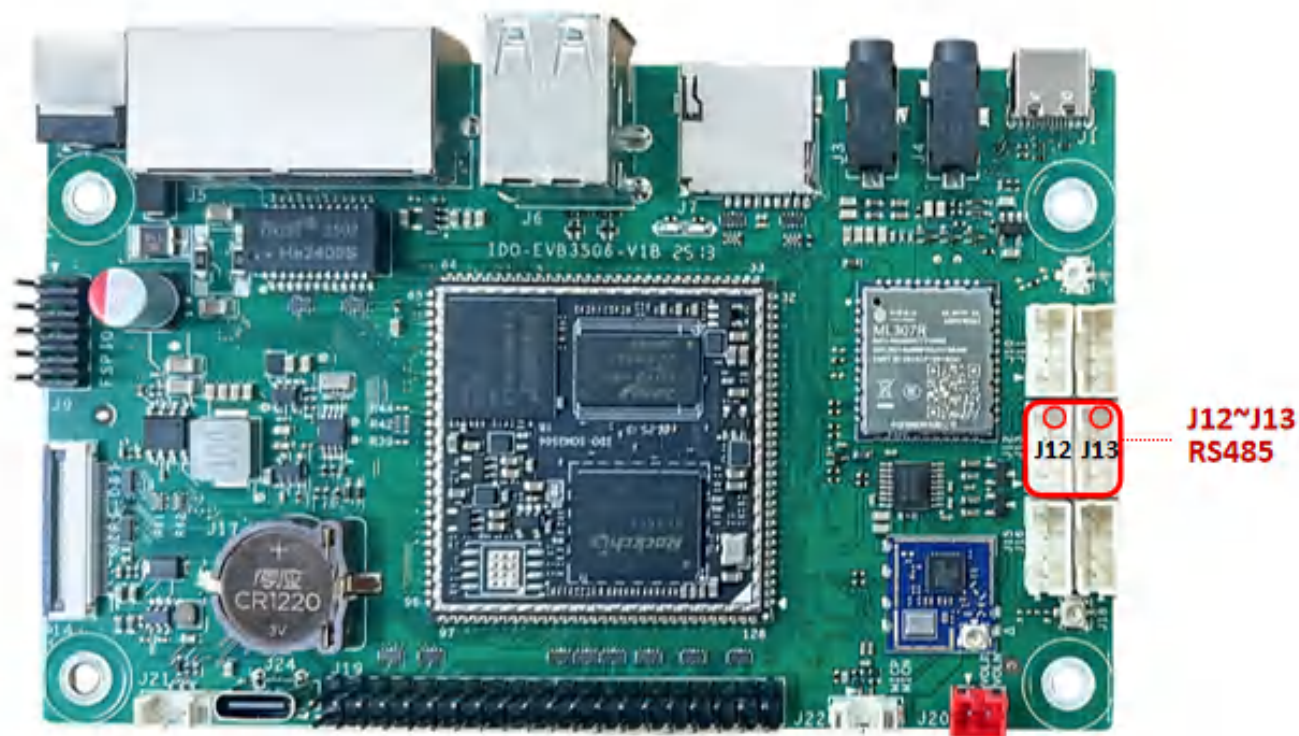
使用microcom可以进行串口收发测试，命令如下：

▼

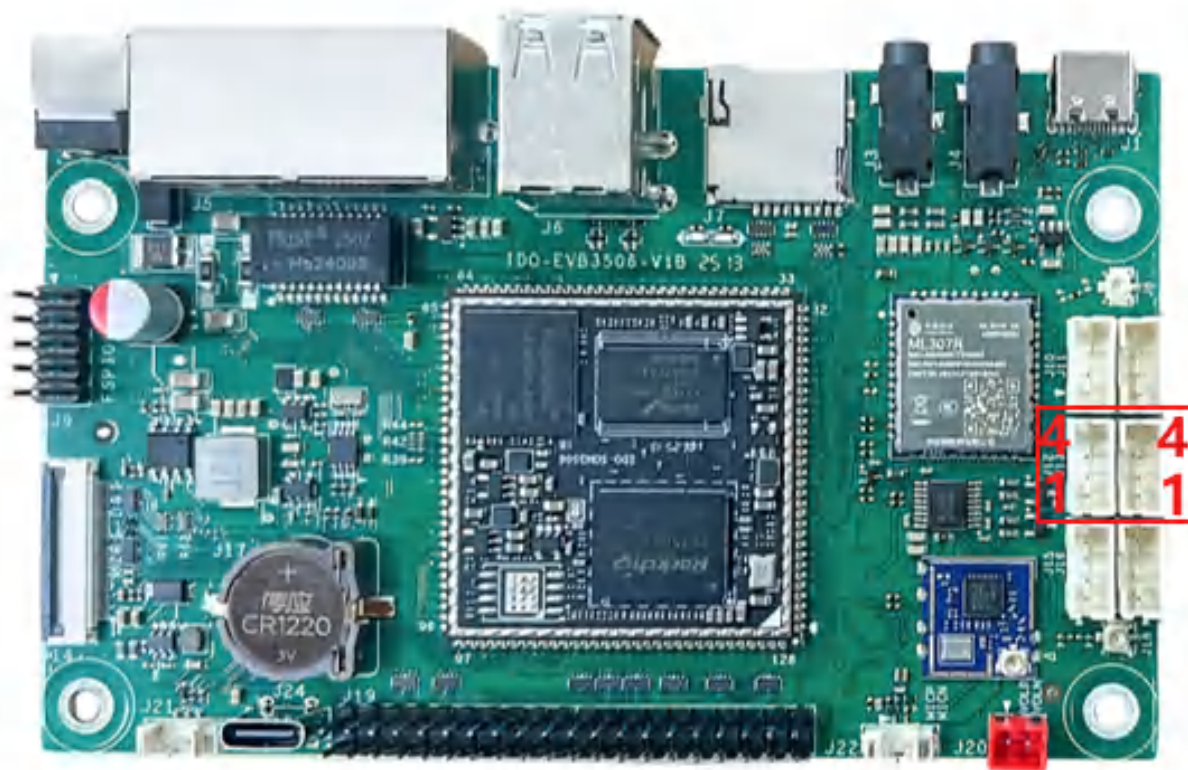
Shell

```
1 root@rk3506-buildroot:/# microcom -s 115200 -p /dev/ttyS1
```

1.2 RS485



序号	接口位置	电平类型	串口设备节点	备注
1	J12	RS485	/dev/ttyS3	
2	J13	RS485	/dev/ttyS4	此功能默认不通(引脚和RGB 屏幕冲突)



引脚定义：

序号	定义	电平/V	说明
1	VCC	5V	5V
2	RXD	/	RXD
3	TXD	/	TXD
4	GND	GND	电源地

使用microcom可以进行串口收发测试，命令如下：

▼

Shell

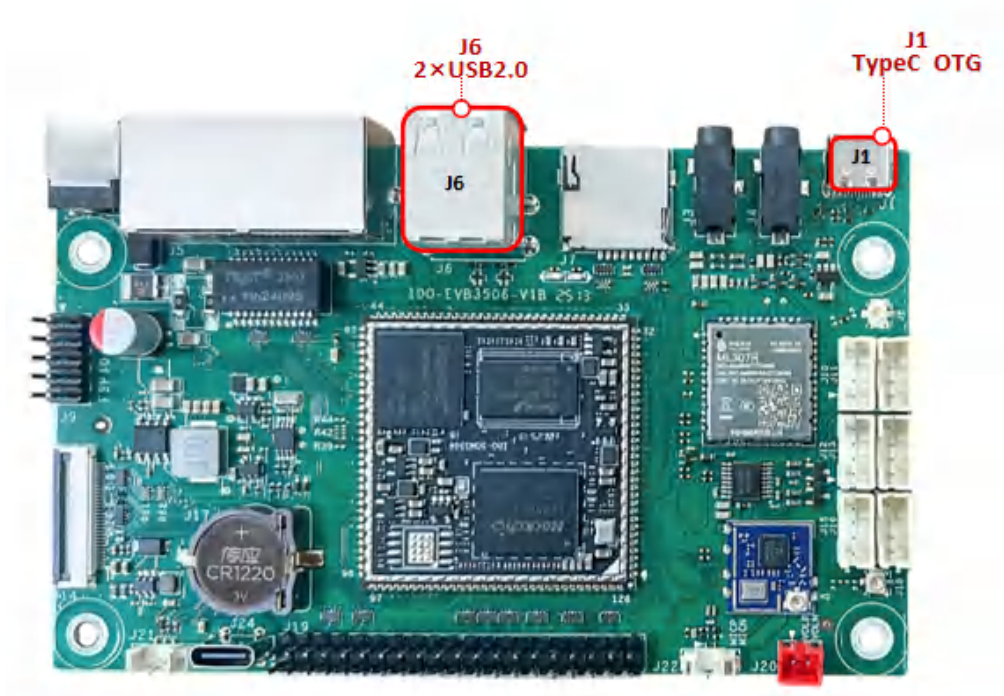
```
1 root@rk3506-buildroot:/# microcom -s 115200 -p /dev/ttyS3
```

注意：

1. microcom测试完成，输入Ctrl-\ 退出。
2. /dev/ttyS2,/dev/ttyS4 默认不通，此功能需要修改软硬件。

2 USB

主板共配置2路USB LVGL，如下图所示：



序号	座子位置	类型
USB1	J1	OTG
USB2	J6-上	USB2.0 HOST
USB3	J6-下	USB2.0 HOST

其中USB1接电脑上电时，默认为device模式（可使用adb调试）；接U盘或者其他存储设备上电时，默认为host模式（可读取U盘）。

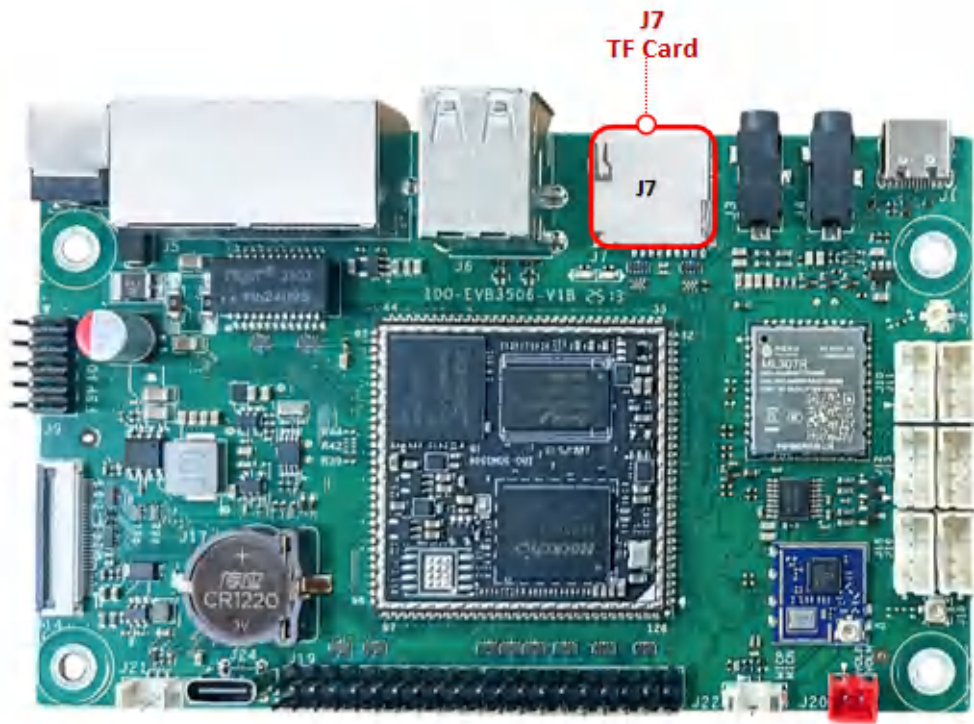
当usb host接上U盘等存储设备时，默认挂载到/mnt/udisk目录，命令如下：

▼Shell

```
1 root@ido:~# mount
2 ...
3 /dev/sda1 on /home/industio/ type vfat (rw,nodev,noexec,noatime,nodiratime,
  fmask=0022,dmask=0022,codepage=437,icharset=iso8859-1,shortname=mixed,erro
  rs=remount-ro)
```

3 Micro SD

主板共配置一路Micro SD接口，如下图所示：



插入SD卡后，默认挂载到/mnt/sdcard目录，如下所示：

▼ Shell

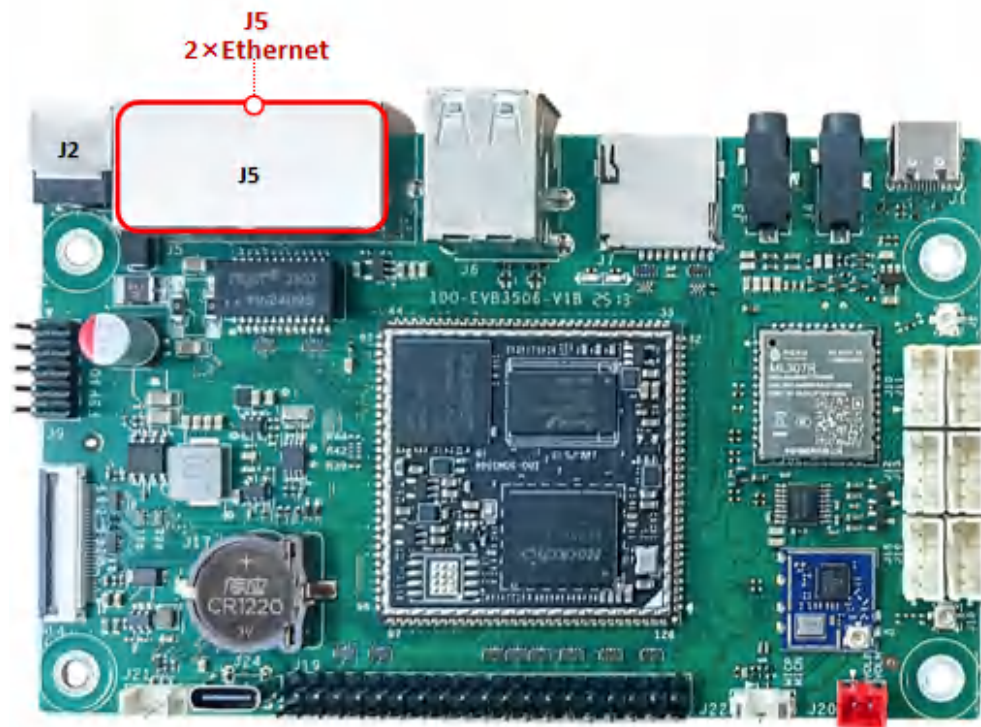
```
1 root@ido:~# mount
2 ...
3 /dev/mmcblk0p1 on /mnt/sdcard type vfat (rw,nodev,noexec,noatime,nodiratime
,fmask=0022,dmask=0022,codepage=437,iocharset=iso8859-1,shortname=mixed,err
ors=remount-ro)
```

测试项目	要求	测试结果
挂载SD	可以自动挂载fat32格式sd卡	✓
	可以手动挂载NTFS的sd卡	✓

注意： EMMC版本不支持SD卡。

4 Ethernet

主板共配置2路百兆以太网接口，如下图所示：



序号	接口位置	速率	网络节点
1	J5-右	百兆	eth0
1	J5-左	百兆	eth1

系统默认开启DHCP服务，动态获取IP。

▼Shell

```
1 root@ido:~# ifconfig eth0
2 eth0      Link encap:Ethernet  HWaddr 7A:1F:F2:40:0F:F0
3           inet addr:192.168.0.12  Bcast:192.168.0.255  Mask:255.255.255.0
4           UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
5           RX packets:25 errors:0 dropped:0 overruns:0 frame:0
6           TX packets:4 errors:0 dropped:0 overruns:0 carrier:0
7           collisions:0 txqueuelen:1000
8           RX bytes:3358 (3.2 KiB)  TX bytes:1368 (1.3 KiB)
9           Interrupt:57 Base address:0xe000
```

可通过ifconfig指令设置临时静态IP，命令如下：

▼Shell

```
1 root@ido:~# ifconfig eth0 192.168.3.123
```

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

▼Shell

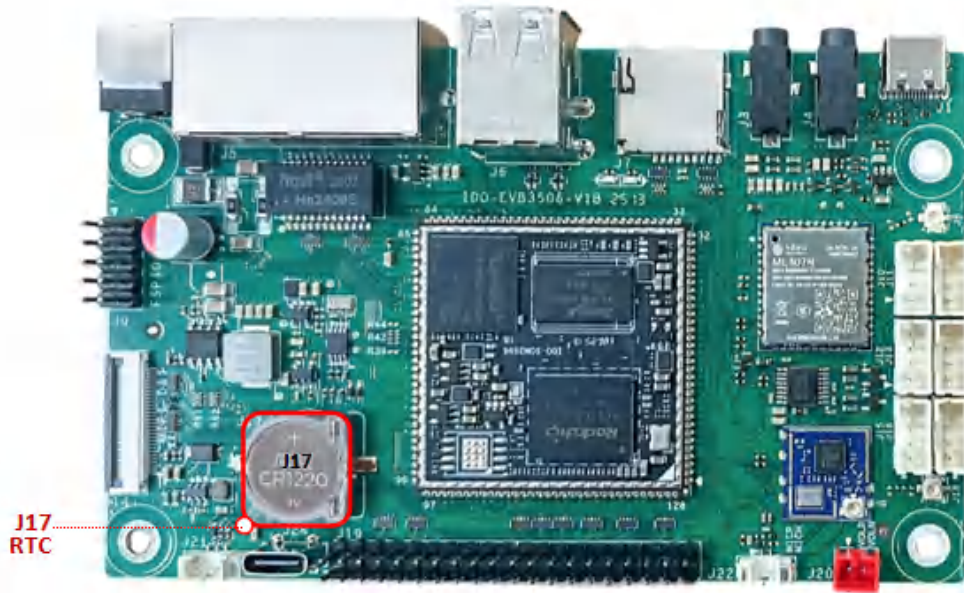
```
1 root@ido:~# vi /etc/network/interfaces
2 root@ido:~# cat /etc/network/interfaces
3 # interface file auto-generated by buildroot
4
5 auto lo
6 iface lo inet loopback
7
8 auto eth0
9 iface eth0 inet static
10 address 192.168.3.123
11 netmask 255.255.255.0
12 gateway 192.168.3.1
13
14 source-directory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

断电重启生效，此静态IP设置仍然生效。

测试项目	要求	测试结果
eth0	识别正常，并成功上网	✓
	自动获取IP地址	✓
	手动修改IP地址	✓
eth1	识别正常，并成功上网	✓
	自动获取IP地址	✓
	手动修改IP地址	✓

5 RTC

主板配置了1路rtc接口，如下图所示：



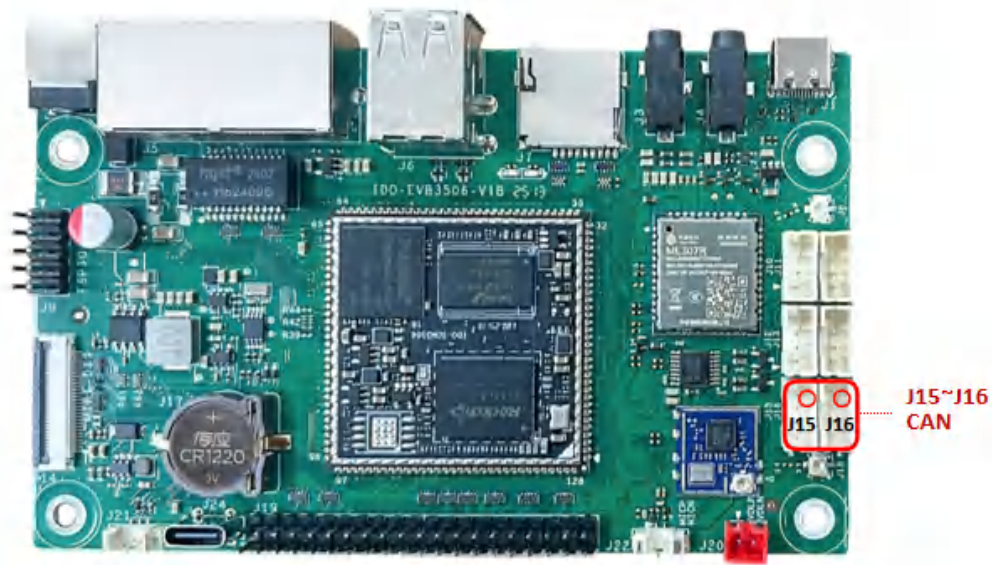
rtc时间设置方法：

Bash

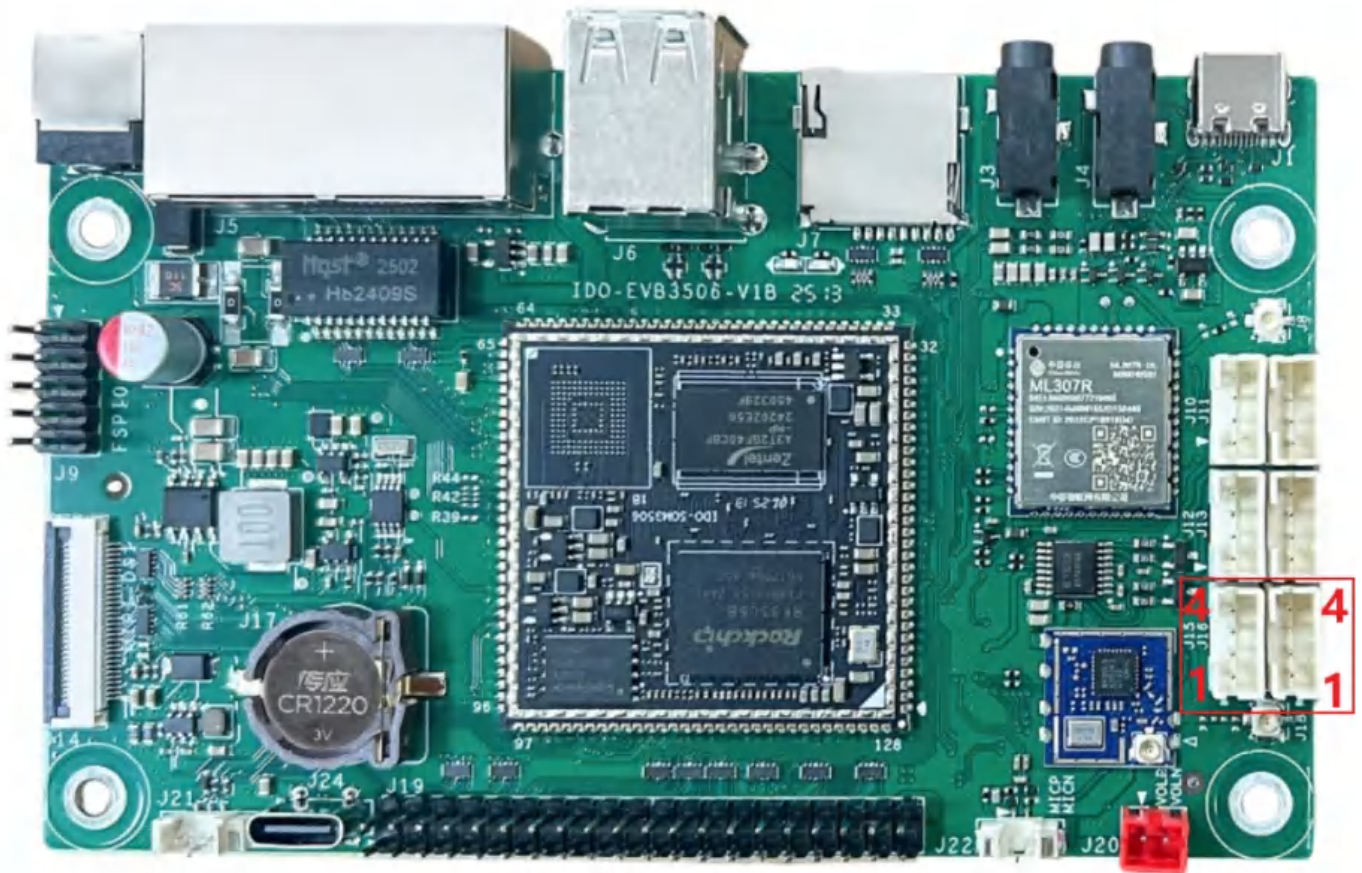
```
1  #同步网络时间
2  root@ido:~# ntpdate cn.pool.ntp.org
3
4  #设置系统时间
5  root@ido:~# date -s "2024-10-09 14:02:30"
6
7  #将rtc时间调整为与目前的系统时间一致
8  root@ido:~# hwclock -w
9
10 #获取硬件rtc当前时间（断电重启读取时间没有太大偏差）
11 root@ido:~# hwclock -r
12 2024-10-09 14:02:35.945604+00:00
13
14 #将rtc时间设置为系统时间
15 root@ido:~# hwclock --systohc
```

6 CAN

主板配置了2路can接口，如下图所示：



序号	接口位置	电平类型	串口设备节点	备注
1	J15	CAN	can1	
2	J16	CAN	can0	此功能默认不通(引脚和RGB 屏幕冲突)



引脚定义：

序号	定义	电平/V	说明
1	VCC	5V	CAN信号
2	CAN_H	/	
3	CAN_L	/	
4	GND	GND	电源地

查看can节点，以can1为例：

8

数据收发，以can1为例：

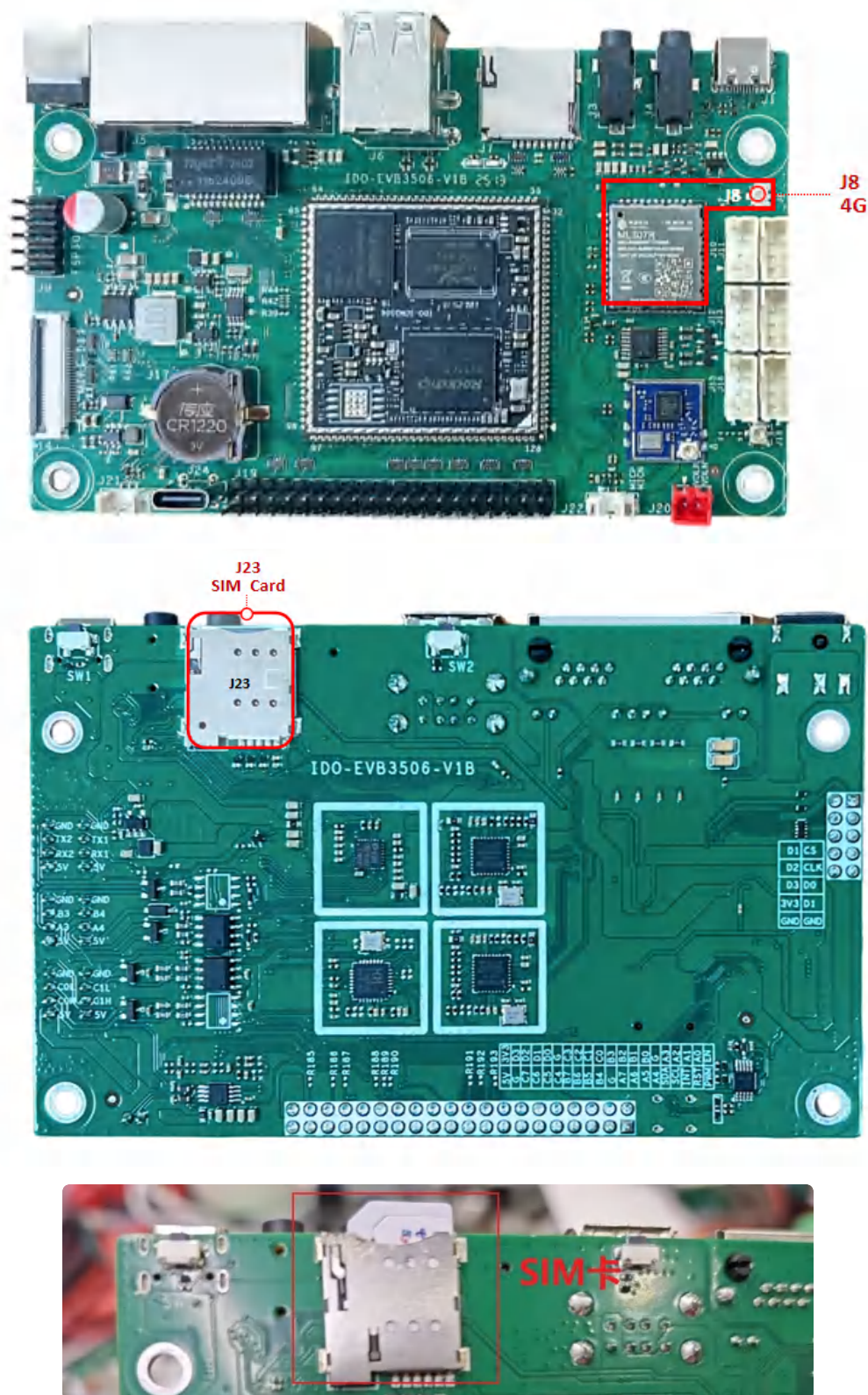
2

注意：

- 1.需要can0需要修改硬件。
- 2.内核dts中can0默认关闭，can1默认打开；can1在系统下默认注册为can0。

7 4G模块

主板配置了1路中移4G模块ML307R，如下图所示：



拨号上网：

```
1 cat /dev/ttyUSB2 &
2
3 #设置拨号协议为RNDIS
4 echo -e "AT+MDIALUPCFG="mode",0" > /dev/ttyUSB2
5
6 #设置拨号协议为ECM
7 echo -e "AT+MDIALUPCFG="mode",1" > /dev/ttyUSB2
8
9 #关闭回显
10 echo -e 'ATE0\r\n' > /dev/ttyUSB2
11
12 #查询PDP上下文配置
13 echo -e 'AT+CGDCONT?\r\n' > /dev/ttyUSB2
14
15 #查询驻网状态
16 echo -e 'AT+COPS?\r\n' > /dev/ttyUSB2
17
18 #设置自动拨号命令
19 echo -e 'AT+MDIALUP=1,1\r\n' > /dev/ttyUSB2
20
21 #查询拨号状态
22 echo -e 'AT+MDIALUP?\r\n' > /dev/ttyUSB2
```

查看网络节点：

```
1 root@ido:~# ifconfig eth2
2 eth2      Link encap:Ethernet  HWaddr AC:0C:29:A3:9B:6D
3           inet addr:192.168.0.100  Bcast:192.168.0.255  Mask:255.255.255.0
4           UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
5           RX packets:4 errors:0 dropped:0 overruns:0 frame:0
6           TX packets:3 errors:0 dropped:0 overruns:0 carrier:0
7           collisions:0 txqueuelen:1000
8           RX bytes:928 (928.0 B)  TX bytes:1026 (1.0 KiB)
```

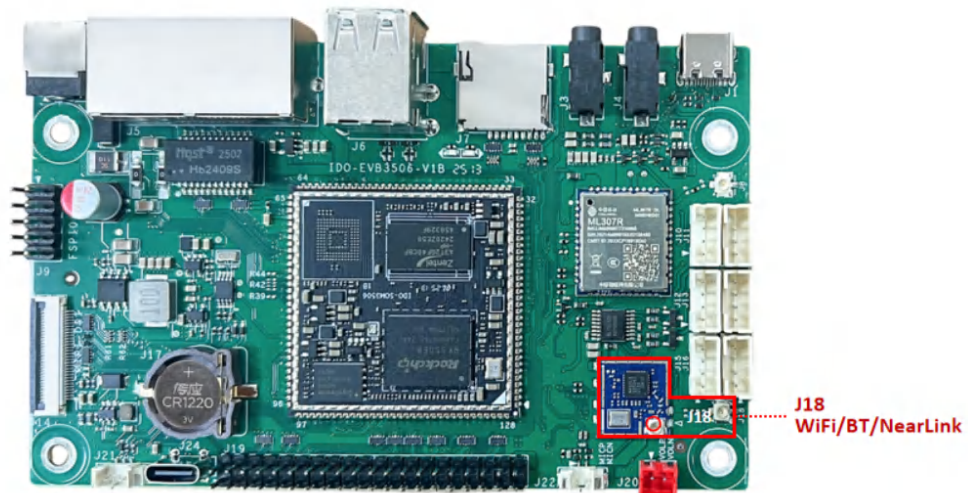
注意：

1.ML307R 4G模块支持RNDIS和ECM模块，两种模式选择其一即可。

2.ML307R 4G模块拨号上网后默认获取局域网内IP地址。

8 星闪模块

主板配置了1路星闪模块，如下图所示：



8.1 SLE

使用两个板子进行对测试。

一台运行：/usr/bin/sle_client_sample。

另一台板子运行：/usr/bin/sle_server_sample。

测试结果如下：

```

uick speed = 707.58 Kbps
t speed = 753.8 Kbps
. speed = 741.92 Kbps
speed = 740.47 Kbps
speed = 747.59 Kbps
mp speed = 666.69 Kbps
eq speed = 672.66 Kbps
eq speed = 710.28 Kbps
speed = 672.31 Kbps
[ 53.189000] [0 96][414]{online_cali_pow_tx_notify::dev
speed = 672.35 Kbps
speed = 689.74 Kbps
speed = 720.58 Kbps
speed = 695.58 Kbps
speed = 677.98 Kbps
speed = 725.32 Kbps
speed = 727.96 Kbps
speed = 639.96 Kbps
speed = 660.83 Kbps
[ 55.349184] [0 96][181][GLA][ALG][TEMP]:cur_state=1
speed = 640.18 Kbps
speed = 677.87 Kbps
speed = 584.6 Kbps
speed = 665.22 Kbps
speed = 677.28 Kbps

```

8.2 BLE

参考BlueZ中的ble部分。

8.3 WIFI

账号密码配置：

```

▼ Bash
1 root@ido:~# cat /etc/wpa_supplicant.conf
2 ctrl_interface=/var/run/wpa_supplicant
3 ap_scan=1
4 update_config=1
5
6 network={
7     ssid="SSID" #账号
8     psk="PASSWOWD" #密码
9     key_mgmt=WPA-PSK
10 }

```

ssid: 账号。

psk: 密码。

连接方法:

```
▼ Bash |
1  #杀掉系统的 wpa_supplicant
2  root@ido:~# pkill wpa_supplicant
3  #删除残留控制接口
4  root@ido:~# rm -rf /var/run/wpa_supplicant/wlan0
5  #连接
6  root@ido:~# wpa_supplicant -Dnl80211 -c /etc/wpa_supplicant.conf -i wlan0 &
```

获取IP:

```
▼ Bash |
1  root@ido:~# udhcpc -i wlan0
2  udhcpc: started, v1.36.1
3  udhcpc: broadcasting discover
4  udhcpc: broadcasting select for 192.168.74.160, server 192.168.74.123
5  udhcpc: lease of 192.168.74.160 obtained from 192.168.74.123, lease time 3
    599
6  deleting routers
7  adding dns 192.168.74.123
8
9  root@ido:~# ifconfig wlan0
10 wlan0      Link encap:Ethernet  HWaddr 78:22:88:85:63:96
11           inet addr:192.168.74.160  Bcast:192.168.74.255  Mask:255.255.25
    5.0
12           UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
13           RX packets:77 errors:0 dropped:0 overruns:0 frame:0
14           TX packets:94 errors:0 dropped:0 overruns:0 carrier:0
15           collisions:0 txqueuelen:1000
16           RX bytes:6190 (6.0 KiB)  TX bytes:11915 (11.6 KiB)
```

测试:

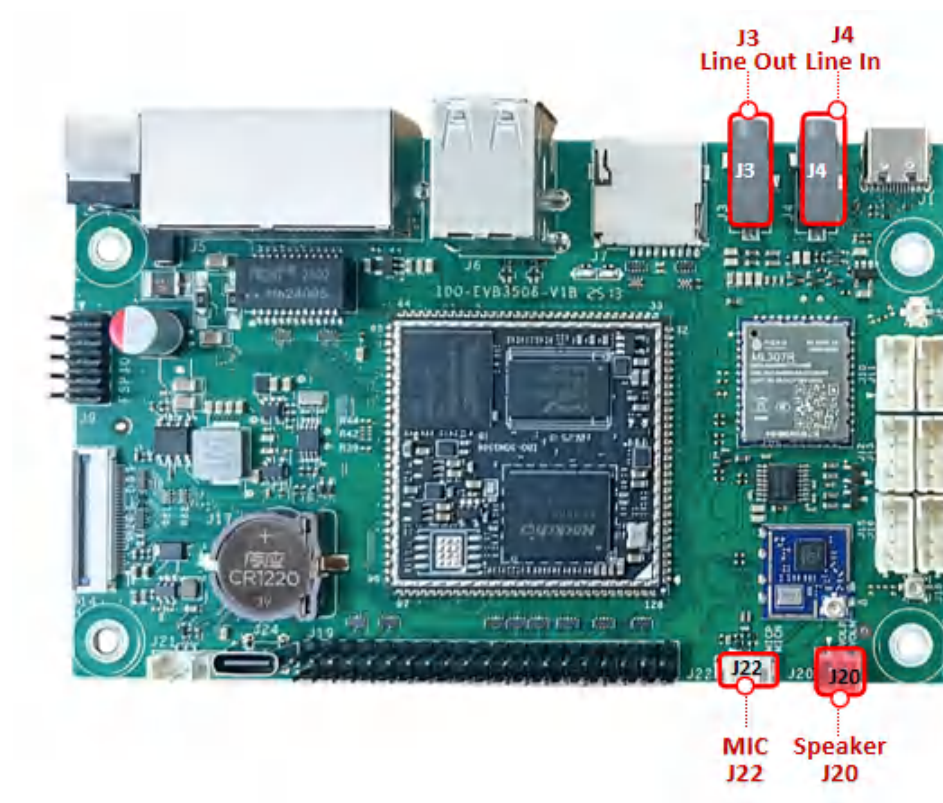
```

1 root@ido:~# ping www.baidu.com -I wlan0
2 PING www.baidu.com (103.235.46.102): 56 data bytes
3 64 bytes from 103.235.46.102: seq=0 ttl=48 time=266.668 ms
4 64 bytes from 103.235.46.102: seq=1 ttl=48 time=233.898 ms
5 64 bytes from 103.235.46.102: seq=2 ttl=48 time=276.997 ms
6 64 bytes from 103.235.46.102: seq=3 ttl=48 time=229.841 ms

```

9 声卡

主板配置的声卡接口，如下图所示：



序号	接口位置	功能
1	J20	Speaker
2	J3	Line Out
3	J4	Line lut
4	J22	MIC

声卡节点

```
▼ Shell |
1  #声卡节点
2  root@ido:~# aplay -l
3  **** List of PLAYBACK 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
8  #录音节点
9  root@ido:~# arecord -l
10 **** List of CAPTURE Hardware Devices ****
11 card 0: rockchip-es8388 [rockchip-es8388], device 0: dailink-multicodecs ES
   8323 HiFi-0 [dailink-multicodecs ES8323 HiFi-0]
12     Subdevices: 1/1
13     Subdevice #0: subdevice #0
14 card 1: rockchip-acoec [rockchip,acoec], device 0: ff4a8000.sai-rk3506-hi
   fi rk3506-hifi-0 [ff4a8000.sai-rk3506-hifi rk3506-hifi-0]
15     Subdevices: 1/1
16     Subdevice #0: subdevice #0
```

音频播放

```
▼ Shell |
1  #音频播放
2  root@ido:~# amixer -c 0 cset numid=56 192
3  numid=56,iface=MIXER,name='PCM Volume'
4      ; type=INTEGER,access=rw---R--,values=2,min=0,max=192,step=0
5      : values=192,192
6      | dBscale-min=-96.00dB,step=0.50dB,mute=1
7
8  root@ido:~# aplay /tmp/xihuang
9  Playing WAVE '/tmp/xihuang' : Signed 16 bit Little Endian, Rate 44100 Hz, S
   tereo
```

MIC录音

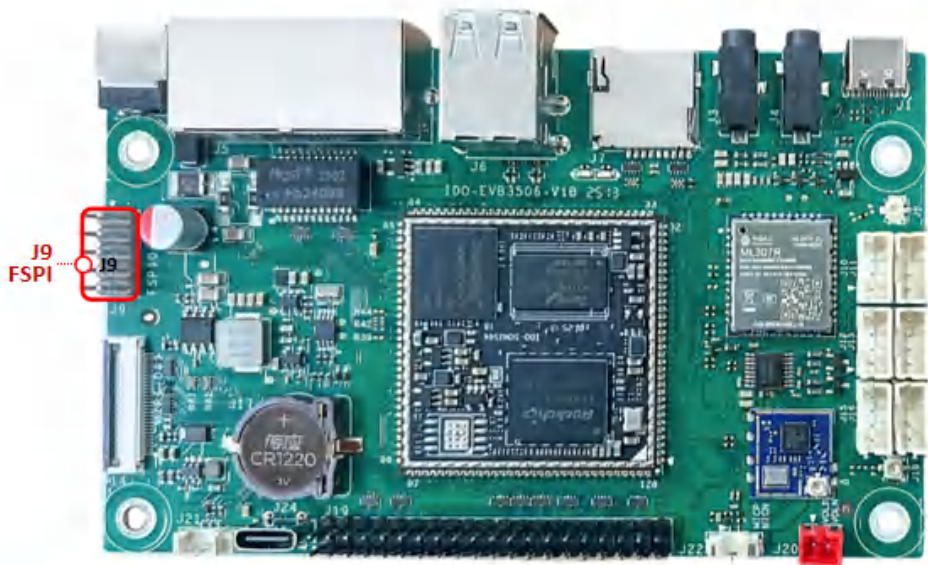
```
1 #设置音量大小, 0最小, 31最大
2 root@ido:~# amixer -c 1 cset numid=40 31
3 #录音
4 root@ido:~# arecord -D hw:1,0 -r 44100 -c 2 -f S16_LE test.wav
5 #播放
6 root@ido:~# aplay test.wav
```

LINE IN录音

```
1 #line in录音
2 #切换到line in
3 root@ido:~# echo -n "mic1" > /sys/class/es8388_mic/mic_channel
4
5 root@ido:~# arecord -D hw:0,0 -f S16_LE -r 44100 -c 2 /tmp/test_mic1.wav
6 Recording WAVE '/tmp/test_mic1.wav' : Signed 16 bit Little Endian, Rate 44100 Hz, Stereo
```

10 扩展接口

主板配置了1路扩展接口，默认做GPIO，如下图所示：



序号	接口位置	GPIO 标号	GPIO 序号
----	------	---------	---------

1	J9-1	gpio2_a0	64
2	J9-2	gpio2_a3	67
3	J9-3	gpio2_a1	65
4	J9-4	gpio2_a4	68
5	J9-5	gpio2_a2	66
6	J9-6	gpio2_a5	69

gpio设置：

以gpio2_a3为例：

▼

Shell

```

1  #输出
2  # 设置 gpio2_a3 输出高电平状态
3  gpioset gpiochip2 3=1
4
5  # 设置 gpio2_a3 输出低电平状态
6  gpioset gpiochip2 3=0
7
8  #输入
9  # 获取 gpiochip2 输入电平状态
10 gpioget gpiochip2 3
11

```

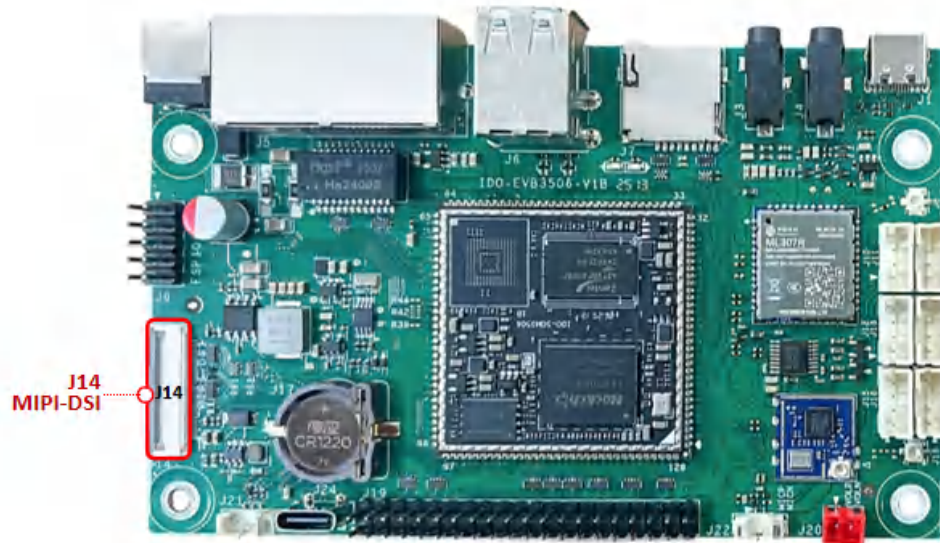
注意：

- 1.nand版本不支持，emmc版本支持。
- 2.emmc版本需要此功能需要修改硬件。

11 显示接口

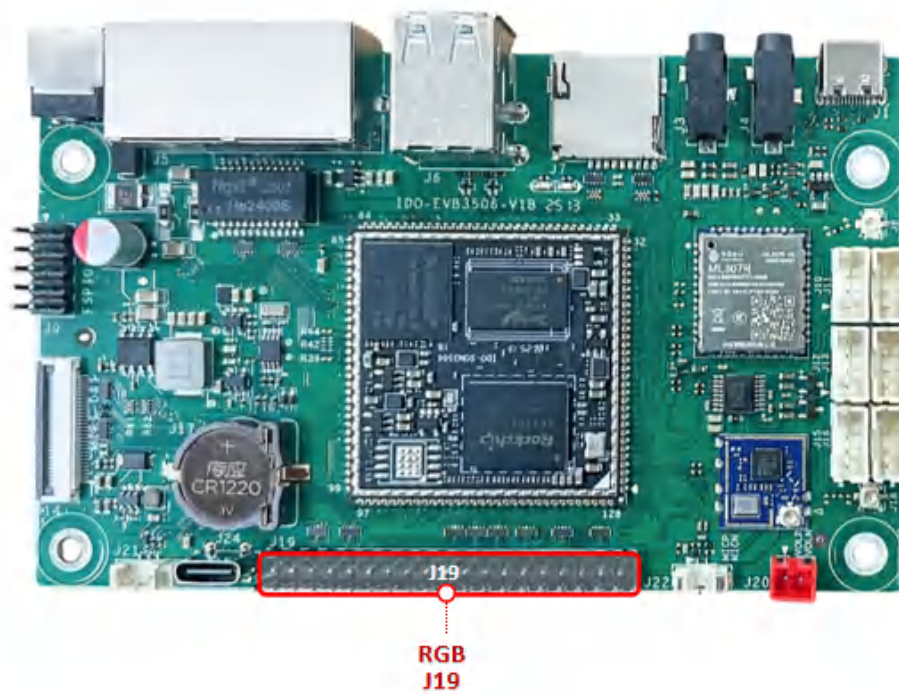
11.1 MIPI

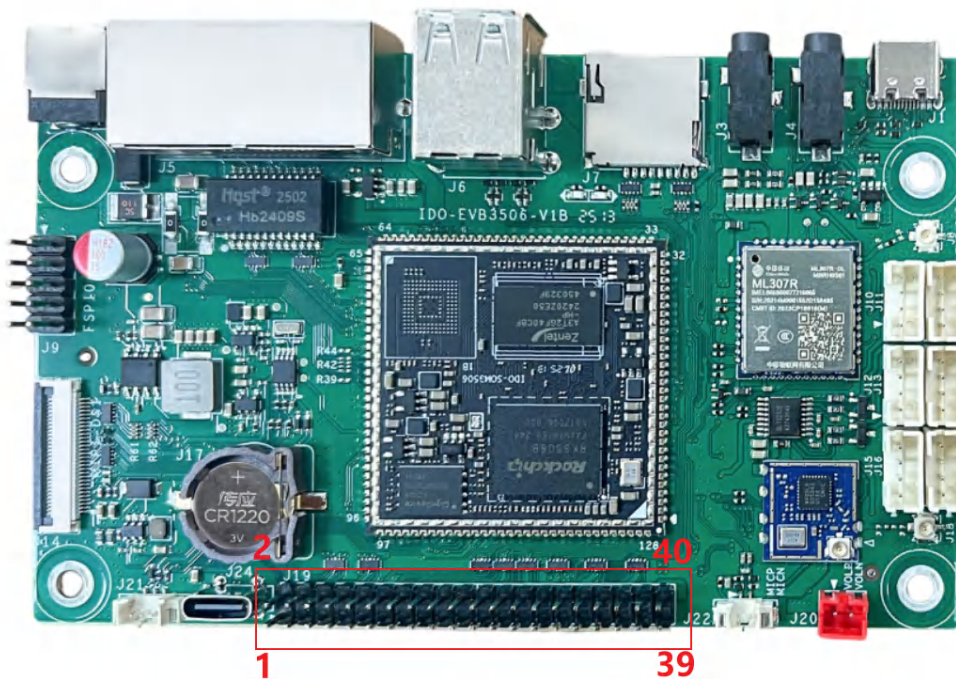
主板配置了1路MIPI接口，如下图所示：



11.2 RGB屏幕

主板配置了1路RGB接口，如下图所示：





引脚定义:

序号	定义	电平/V
1	RGB_PWM	/
2	LCD_PWREN_H_	/
3	TP_RST0	/
4	VO_LCDC_DEN/GPIO1_A0	/
5	TP_INT0	/
6	VO_LCDC_VSYNC/GPIO1_A1	/
7	TP_SCL0	/
8	VO_LCDC_HSYNC/GPIO1_A2	/
9	TP_SDA0	/
10	VO_LCDC_CLK/GPIO1_A3	/
11	VO_LCDC_D23/GPIO1_A4	/
12	GND	GND
13	VO_LCDC_D22/GPIO1_A5	/
14	VO_LCDC_D19/GPIO1_B0	/
15	VO_LCDC_D21/GPIO1_A6	/

16	VO_LCDC_D18/UART2_TX/G PIO1_B1	/
17	VO_LCDC_D20/GPIO1_A7	/
18	VO_LCDC_D17/CAN0_TX/GPI O1_B2	/
19	GND	GND
20	VO_LCDC_D16/CAN0_RX/GP IO1_B3	/
21	VO_LCDC_D15/GPIO1_B4	/
22	VO_LCDC_D11/GPIO1_C0	/
23	VO_LCDC_D14/GPIO1_B5	/
24	VO_LCDC_D10/DSM_AUD_R N/GPIO1_C1	/
25	VO_LCDC_D13/GPIO1_B6	/
26	VO_LCDC_D9/DSM_AUD_RP /GPIO1_C2	/
27	VO_LCDC_D12/GPIO1_B7	/
28	VO_LCDC_D8/UART2_RX/GP IO1_C3	/
29	VO_LCDC_D7/GPIO1_C4	/
30	GND	GND
31	VO_LCDC_D6/GPIO1_C5	/
32	VO_LCDC_D3/GPIO1_D0	/

33	VO_LCDC_D5/GPIO1_C6	/
34	VO_LCDC_D2/UART4_RTSN/ GPIO1_D1	/
35	VO_LCDC_D4/GPIO1_C7	/
36	VO_LCDC_D1/UART4_RX/GPI O1_D2	/
37	GND	GND
38	VO_LCDC_D0/UART4_TX/GP IO1_D3	/
39	VCC5V0	5V
40	VCC3V3	3V3

注意：

1.J19默认配成RGB显示。

2.RGB屏幕需要外接转接小板。