IDO-EVB5301-V1 Linux使用手册

Industio 触觉智能

IDO-EVB5301-V1

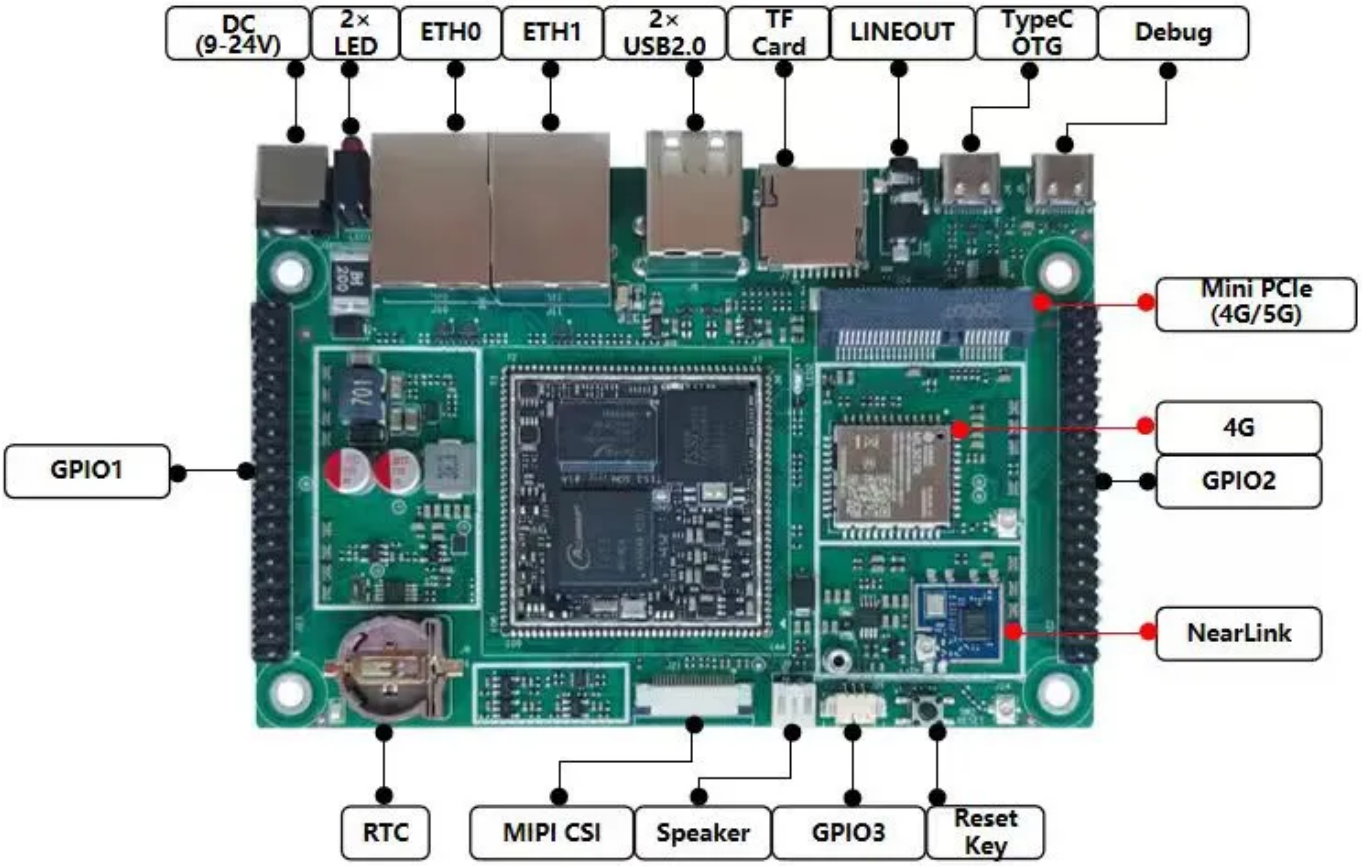
Linux使用手册

深圳触觉智能科技有限公司

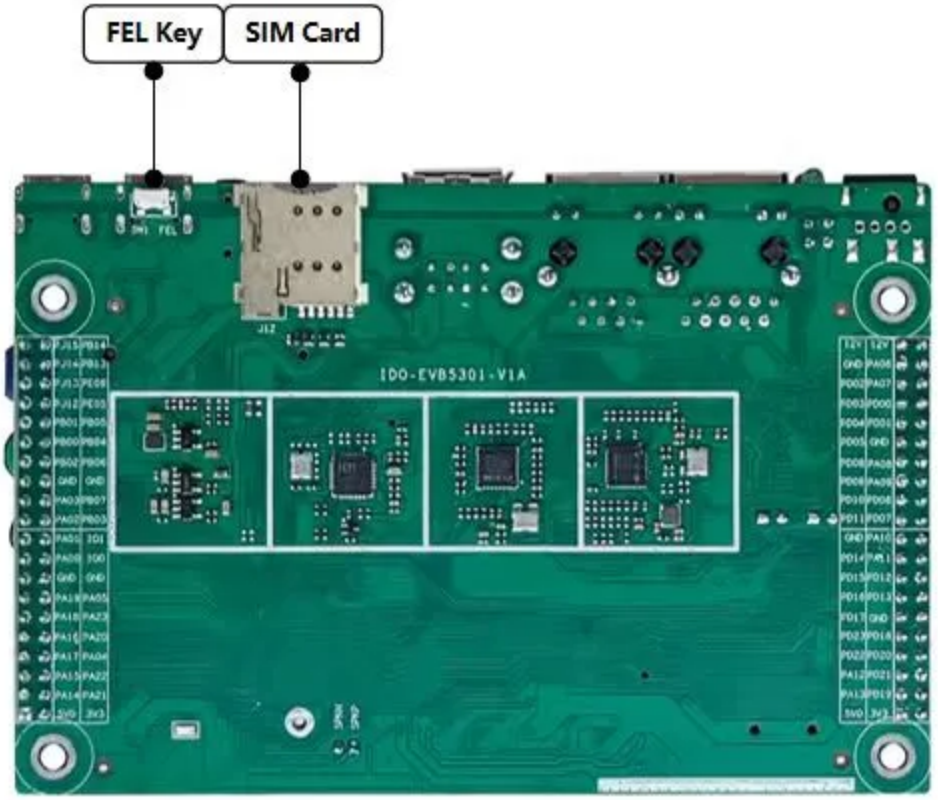
www.industio.cn

文档修订历史

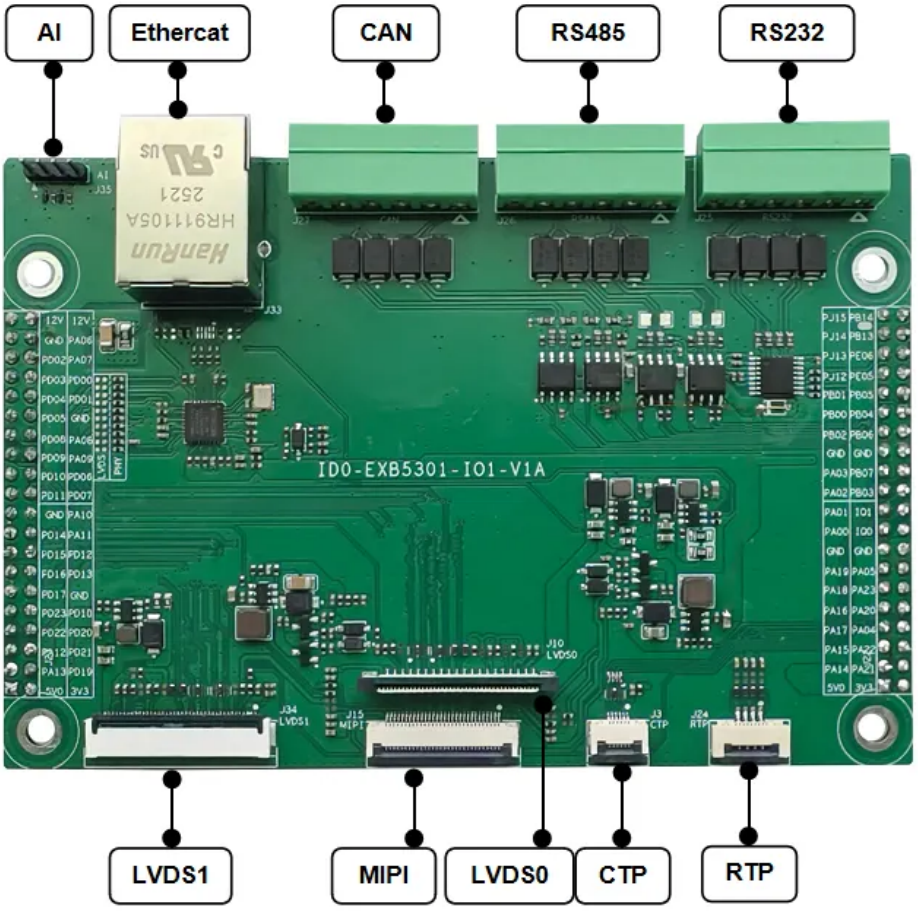
版本	PCBA版本	修订内容	修订	审核	日期
V1.0	V1B	创建文档	LJH	IDO	2025/09/28


DC
(9-24V)
2×
LED
ETH0
ETH1
2×
USB2.0
TF
Card
LINEOUT
TypeC
OTG
Debug
Mini PCIe
(4G/5G)
4G
GPIO2
NearLink
GPIO1
RTC
MIPI CSI
Speaker
GPIO3
Reset
Key

底板（正面）


FEL Key
SIM Card
IDO-EVB5301-V1A

底板（背面）



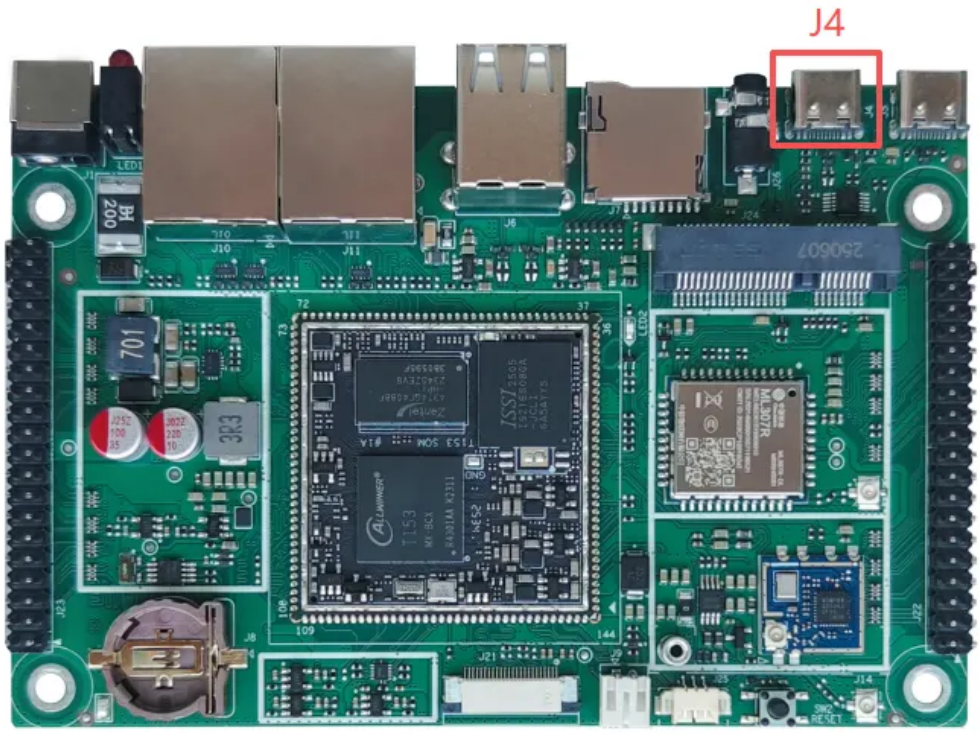
拓展板

Buildroot使用手册

1. 调试接口

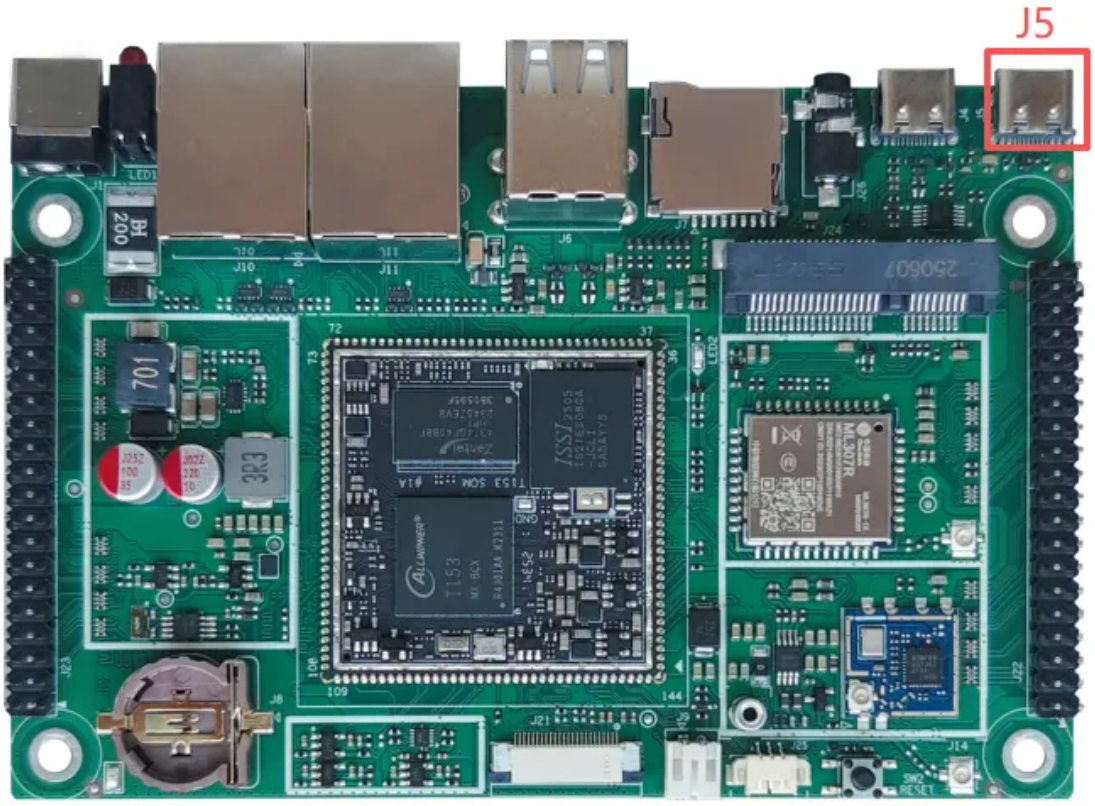
1.1. adb

开发板adb调试口位于主板J4处



1.2. 调试串口

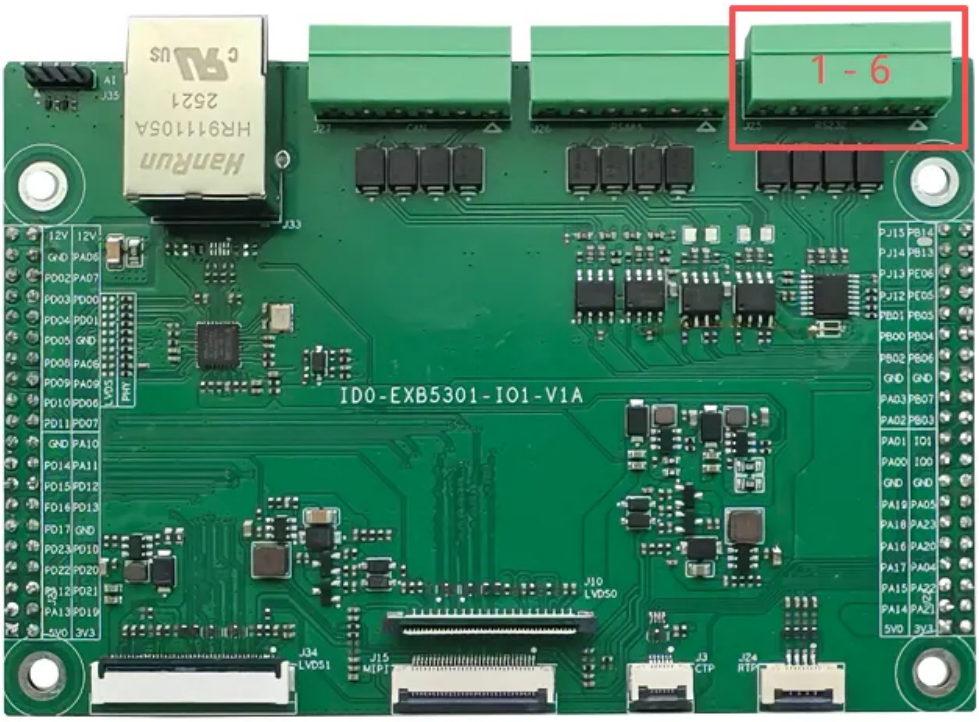
开发板debug串口位于J5处，波特率：115200



2. UART

主板共配置4路串口，包括2路RS232和2路RS485（除调试串口），如下所示。

2.1. RS232



序号	接口位置	电平类型	串口设备节点	备注
1	J25	RS232	/dev/ttyAS5	/
2		RS232	/dev/ttyAS7	/

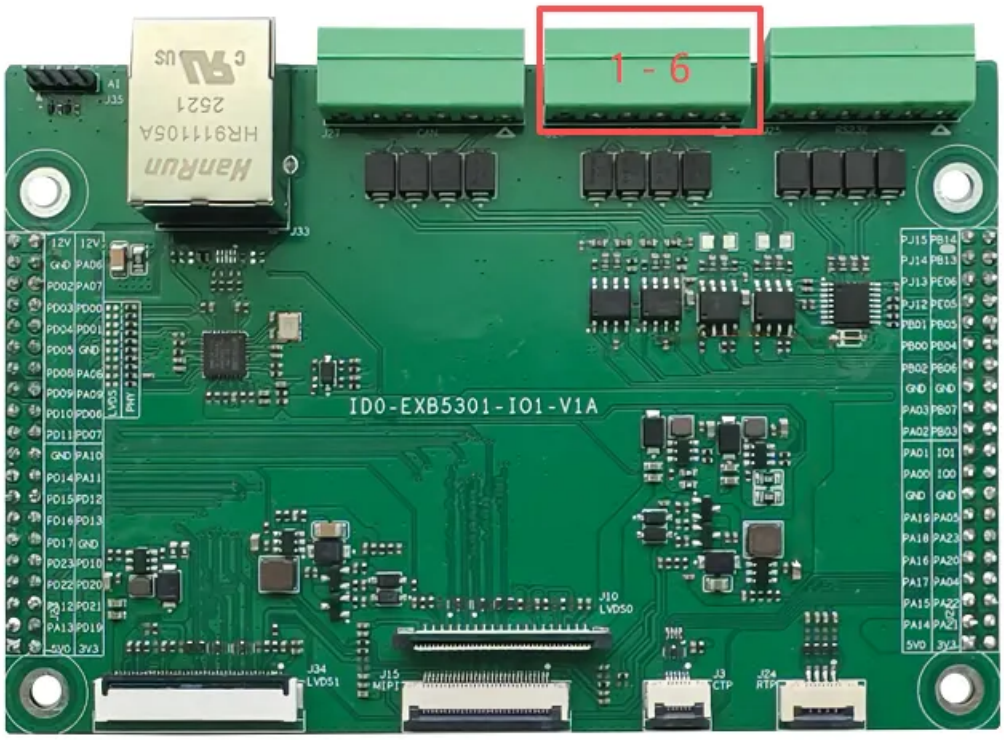
引脚定义：

序号	定义	电平/V	说明
1	GND	GND	电源地
2	TX7	/	RS232_TX
3	RX7	/	RS232_RX
4	TX5	/	RS232_TX
5	RX5	/	RS232_RX
6	5V	5V	电源5V

收发测试

```bash
# cat /dev/ttyAS5 &
[1] 453
# echo 123123 > /dev/ttyAS7
123123

# echo 123123 > /dev/ttyAS7
123123

# kill 453
# cat /dev/ttyAS7 &
[2] 454
[1]   Terminated              cat /dev/ttyAS5
# echo 123123 > /dev/ttyAS5
123123

# echo 123123 > /dev/ttyAS5
123123

# echo 123123 > /dev/ttyAS5
123123

#
```

2.2. RS485



序号	接口位置	电平类型	串口设备节点	备注
1	J26	RS485	/dev/ttyAS2	
2		RS485	/dev/ttyAS8	

引脚定义：

序号	定义	电平/V	说明
1	GND	GND	电源地
2	RS485_B8	/	RS485_B
3	RS485_A8	/	RS485_A
4	RS485_B2	/	RS485_B
5	RS485_A2	/	RS485_A

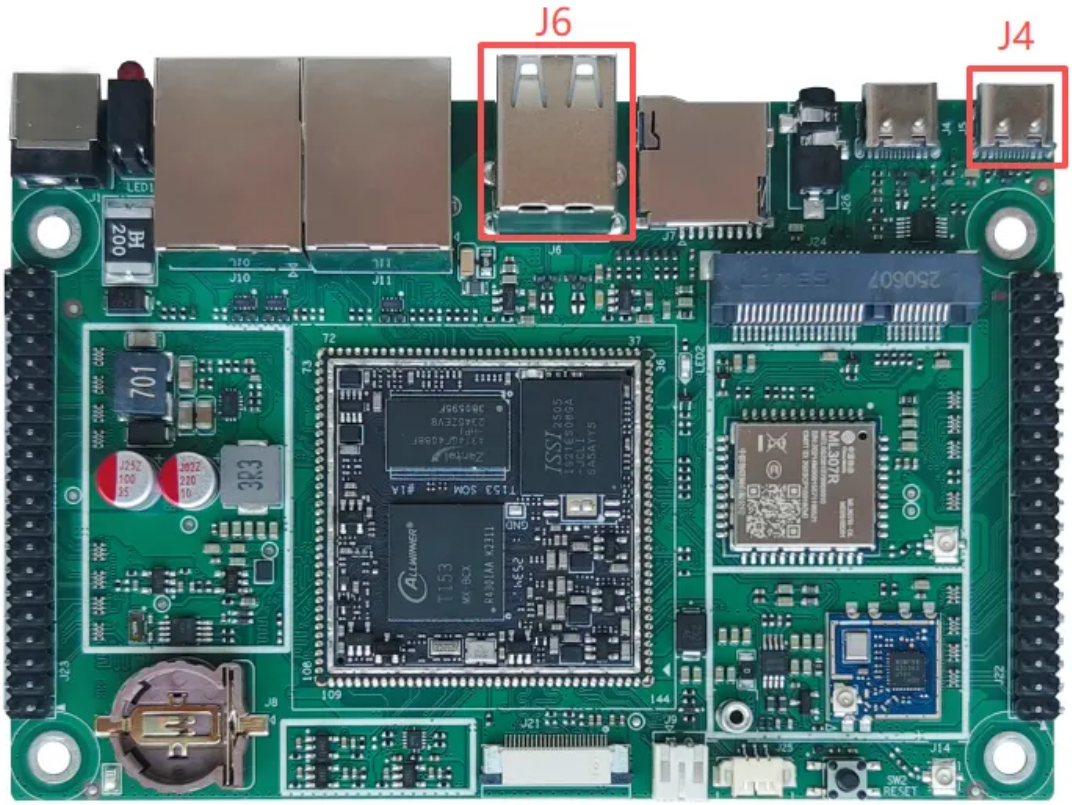
6	5V	5V	电源5V

rs485可以通过内置 `rs485_test` 测试demo进行短接验证，短接 `ttyAS2` 与 `ttyAS8`

```Bash
# rs485_test
usage: ./rs485_test /dev/ttyS2 /dev/ttyAS4
# 若出现 check okay 打印则说明测试验证通过
# rs485_test /dev/ttyAS2 /dev/ttyAS8
 len=1
 len=3
timeout
recv:31
32
33

check okay
# rs485_test /dev/ttyAS8 /dev/ttyAS2
 len=1
 len=3
timeout
recv:31
32
33

check okay

# 若出现以下 check fail 打印，则需检查硬件连接
# rs485_test /dev/ttyAS8 /dev/ttyAS2
timeout
recv:
check fail
#
```

3. 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	J4	OTG
USB2	J6-上	USB2.0 HOST
USB3	J6-下	USB2.0 HOST

接上U盘后，系统出现如下打印

```Bash
[ 2487.378996] usb 1-1.2: new high-speed USB device number 9 using sunxi-ehci
[ 2487.533172] usb 1-1.2: New USB device found, idVendor=0951, idProduct=1666, bcdDevice= 2.00
[ 2487.533186] usb 1-1.2: New USB device strings: Mfr=2, Product=3, SerialNumber=4
[ 2487.533195] usb 1-1.2: Product: DataTraveler 3.0
[ 2487.533203] usb 1-1.2: Manufacturer: Kingston
[ 2487.533210] usb 1-1.2: SerialNumber: E0D55E6C0EC11691B8C1065B
[ 2487.534261] usb-storage 1-1.2:1.0: USB Mass Storage device detected
[ 2487.536615] scsi host0: usb-storage 1-1.2:1.0
```

通过 `fdisk -l` 命令可以查看识别出的usb设备

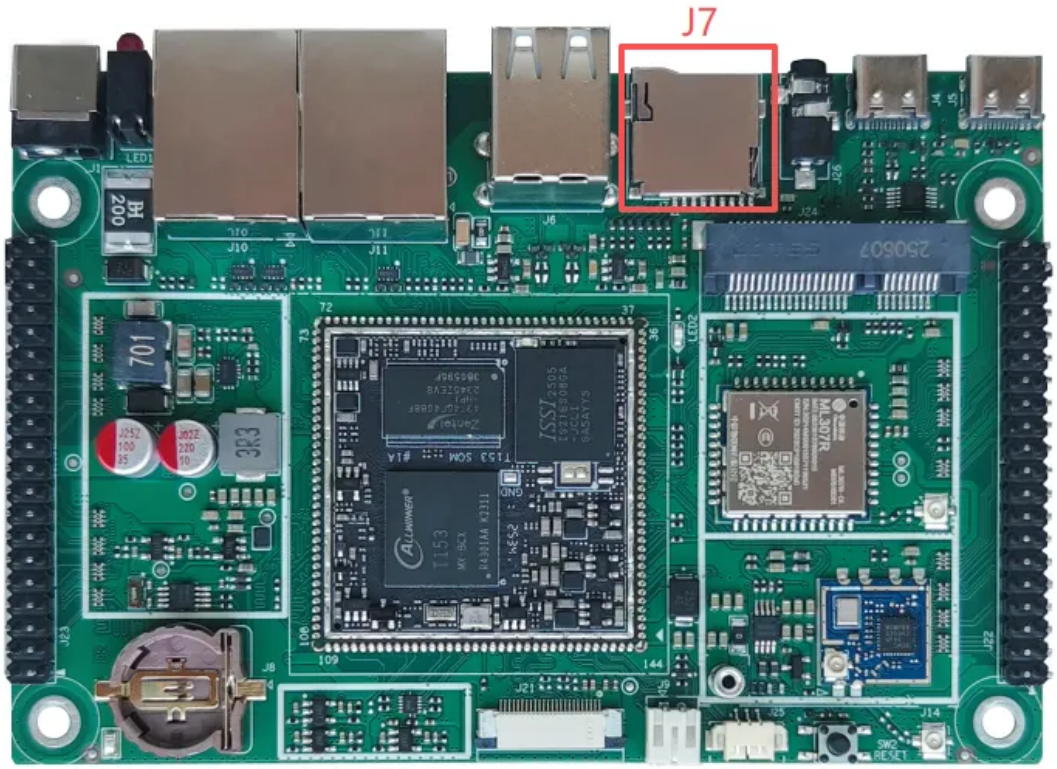
```Bash
# fdisk -l
Found valid GPT with protective MBR; using GPT

……

Disk /dev/sda: 29 GB, 30995907072 bytes, 60538881 sectors
3768 cylinders, 255 heads, 63 sectors/track
Units: sectors of 1 * 512 = 512 bytes

Device  Boot StartCHS    EndCHS        StartLBA     EndLBA    Sectors  Size Id Type
/dev/sda1 *  122,32,39   1023,254,63    1961984   59617280   57655297 27.4G  7 HPFS/NTFS
/dev/sda2    1023,254,63 1023,254,63   59619328   60534783     915456  447M 1b Hidden Win95 FAT32
```

通过 `mount` 手动将U盘挂载至rootfs

```Bash
# mkdir /mnt/usb && mount /dev/sda1 /mnt/usb
```

4. Micro SD

主板共配置一路Micro SD接口，如下图所示：



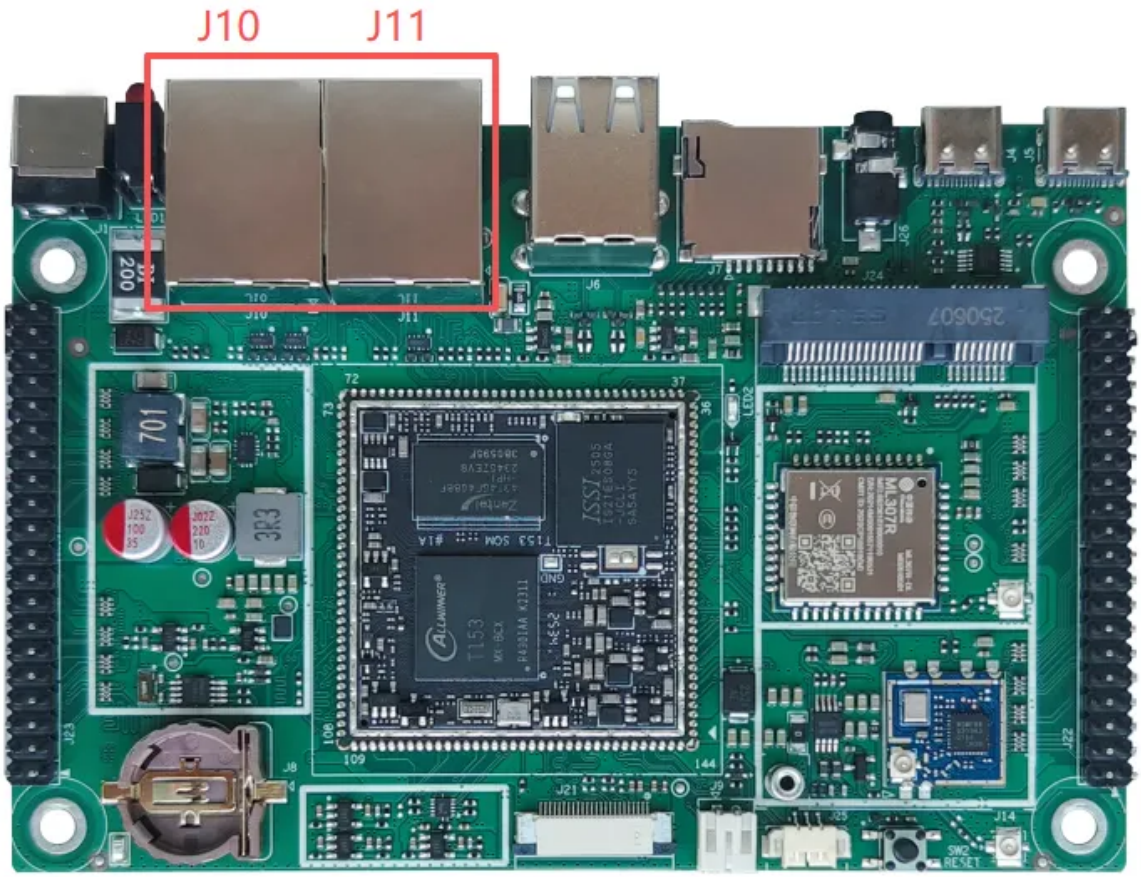
插入SD卡后，默认挂载到/mnt/sdcard目录，如下所示：

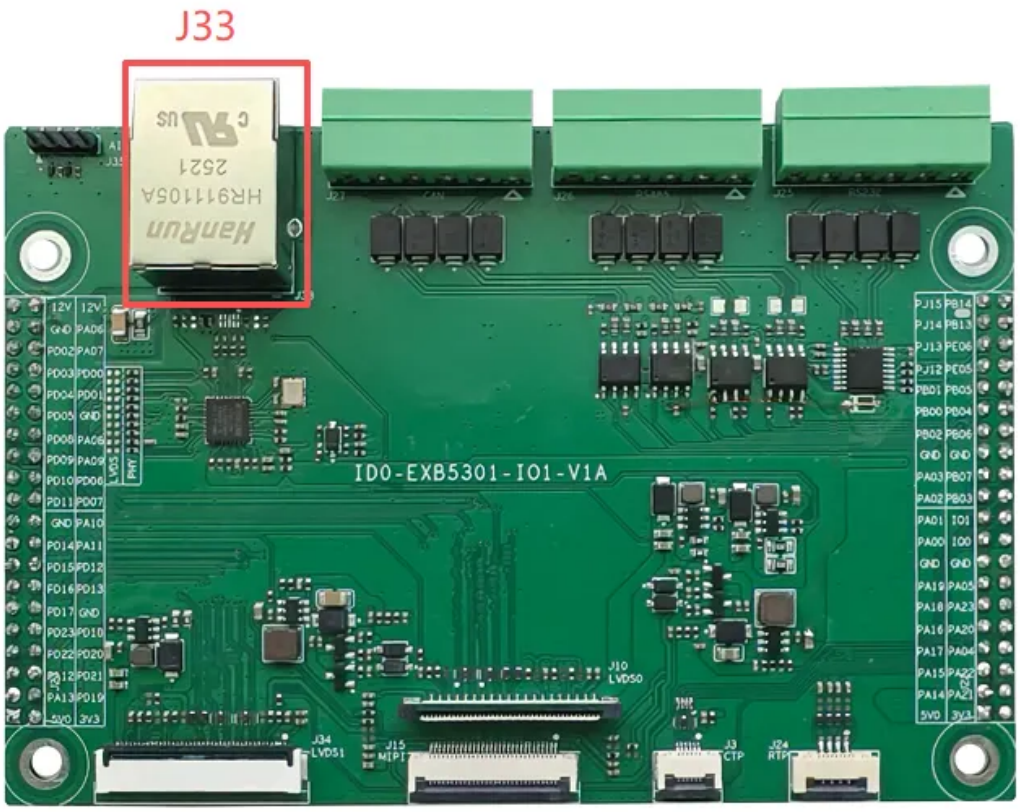
```shell
# mount
...
/dev/mmcblk0p1 on /mnt/sdcard type vfat (rw,nodev,noexec,noatime,nodiratime,fmask=0022,dmask=0022,codepage=437,iocharset=iso8859-1,shortname=mixed,errors=remount-ro)
```

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	✅

5. Ethernet

主板共配置1路千兆以太网接口和2路百兆以太网接口，如下图所示：


J10
J11



序号	接口位置	速率	网络节点
1	J11	百兆	eth0
2	J10	千兆	eth1
3	J33	百兆	eth2

系统默认开启DHCP服务，动态获取IP。

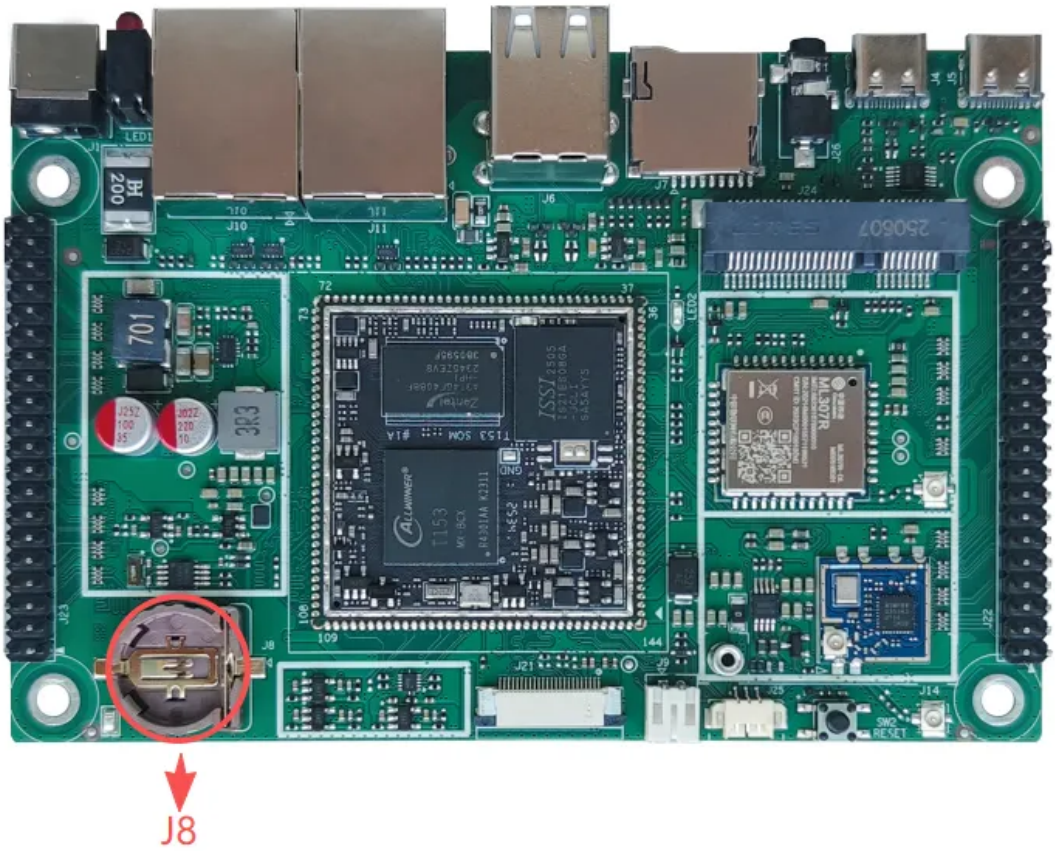
```shell
# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 7A:1F:F2:40:0F:F0
          inet addr:192.168.0.12  Bcast:192.168.0.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:25 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4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

可通过ifconfig指令设置临时静态IP，命令如下：

```shell
# ifconfig eth0 192.168.3.123
```

6. RTC

主板配置了1路rtc接口，如下图所示：



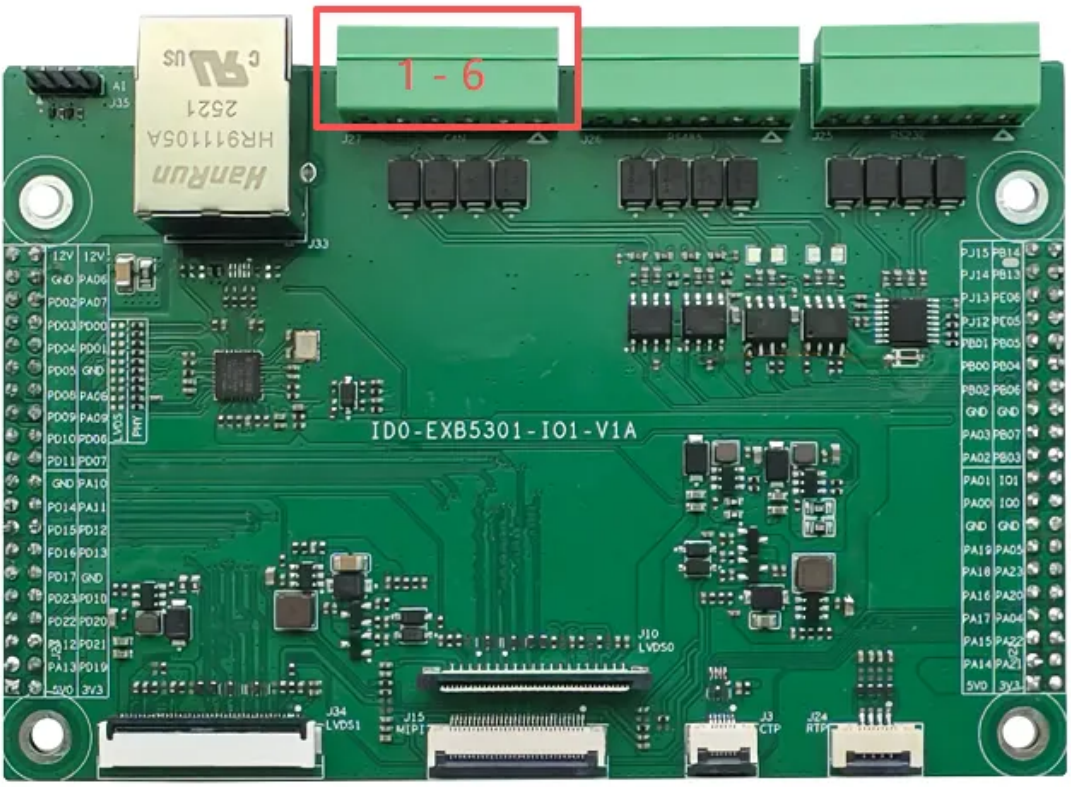
rtc时间设置方法：

```Bash
#设置系统时间
$ date -s "2024-10-09 14:02:30"

#将rtc时间调整为与目前的系统时间一致
$ hwclock -w -f /dev/rtc1

#获取硬件rtc当前时间（断电重启读取时间没有太大偏差）
$ hwclock -r -f /dev/rtc1
2024-10-09 14:02:35.945604+00:00
```

//待补充 rtc0、定时开机

7. 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	J27	CAN	can1	/
2		CAN	can0	/

引脚定义：

序号	定义	电平/V	说明
1	GND	GND	电源地
2	CAN0_L	/	CAN_L
3	CAN0_H	/	CAN_H
4	CAN1_L	/	CAN_L
5	CAN1_H	/	CAN_H
6	5V	5V	电源5V

查看can节点，以can1为例：

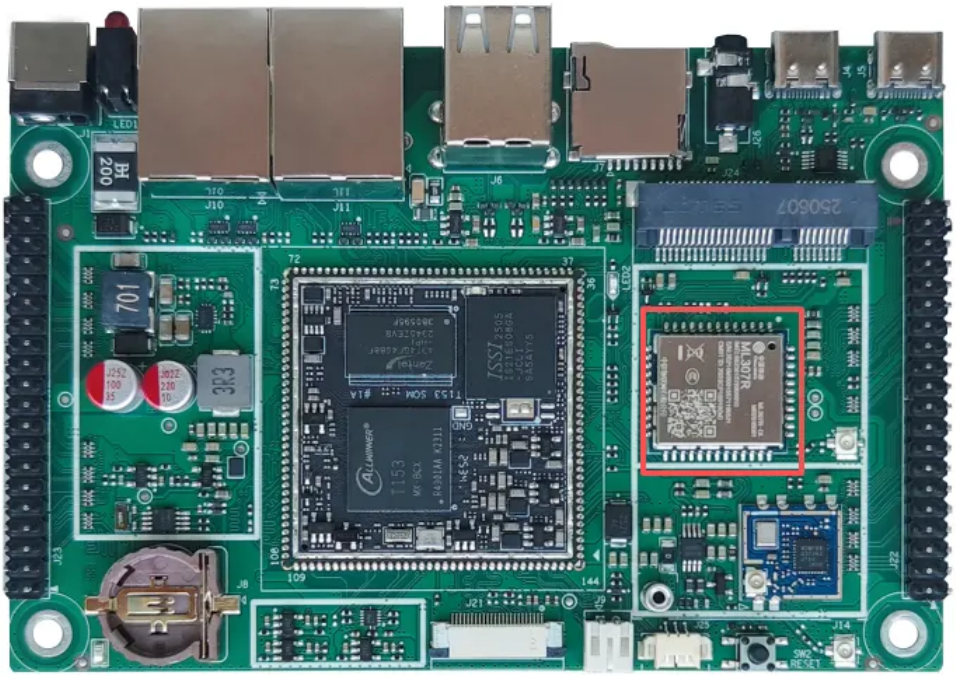
```shell
# ifconfig can0
can0      Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          NOARP  MTU:16  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:10
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
          Interrupt:49
```

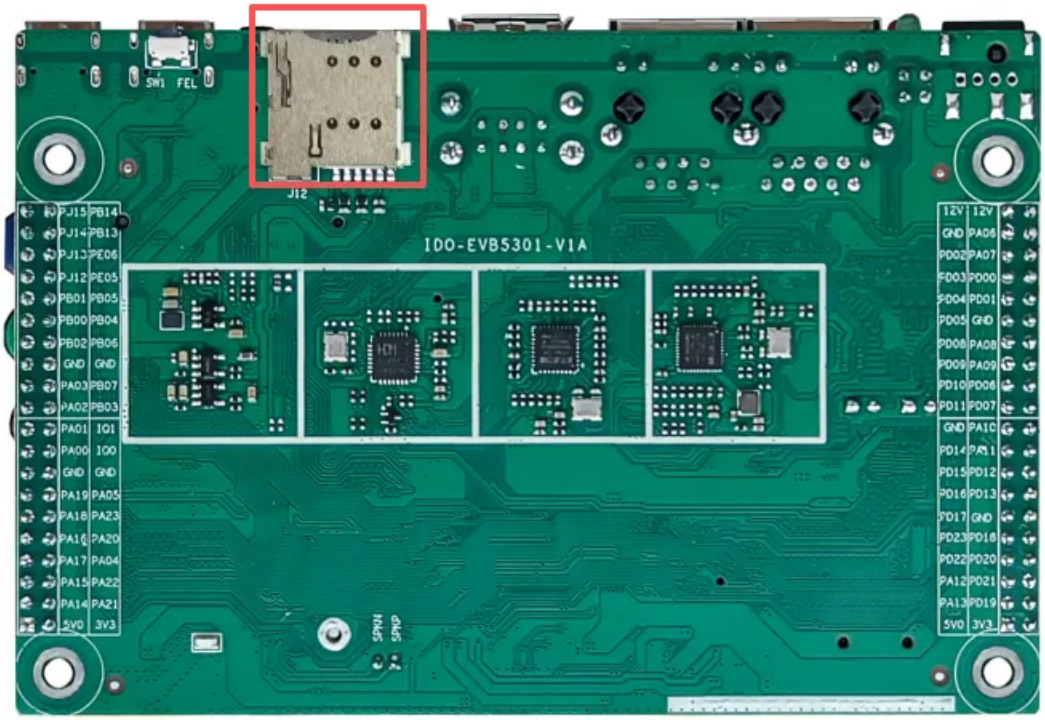
数据收发，以can1为例：

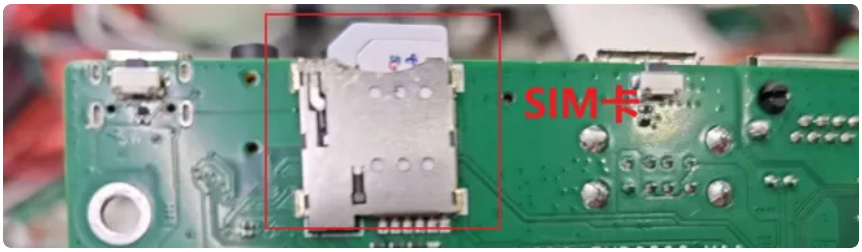
```shell
#检查can设备
$ ip link show

#在收发端关闭can0设备
$ ip link set can0 down

#设置仲裁段1M波特率，数据段1M波特率
$ ip link set can0 type can bitrate 1000000 loopback off
[ 8846.889995] sunxi-can 453c800.can0: PM runtime resume.
[ 8846.895843] sunxi-can 453c800.can0: PM runtime suspend.

#在收发端打开can0设备
$ ip link set can0 up

#在接收端执行candump,阻塞等待报文
$ candump can0

#在发送端执行cansend，发送报文
$ cansend can0 123#1122334455667788

#检查是否启用成功
$ ip link show can0
```

8. 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：

```shell
cat /dev/ttyUSB2 &

#设置拨号协议为RNDIS
echo -e "AT+MDIALUPCFG="mode",0" > /dev/ttyUSB2

#设置拨号协议为ECM
echo -e "AT+MDIALUPCFG="mode",1" > /dev/ttyUSB2

#关闭回显
echo -e 'ATE0\r\n' > /dev/ttyUSB2

#查询PDP上下文配置
echo -e 'AT+CGDCONT?\r\n' > /dev/ttyUSB2

#查询驻网状态
echo -e 'AT+COPS?\r\n' > /dev/ttyUSB2

#设置自动拨号命令
echo -e 'AT+MDIALUP=1,1\r\n' > /dev/ttyUSB2

#查询拨号状态
echo -e 'AT+MDIALUP?\r\n' > /dev/ttyUSB2
```

查看网络节点：

```shell
# 如果是mipi屏幕的固件，4g节点则为eth1
$ ifconfig eth2
eth2      Link encap:Ethernet  HWaddr AC:0C:29:A3:9B:6D
          inet addr:192.168.0.100  Bcast:192.168.0.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:4 errors:0 dropped:0 overruns:0 frame:0
          TX packets:3 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:928 (928.0 B)  TX bytes:1026 (1.0 KiB)
```

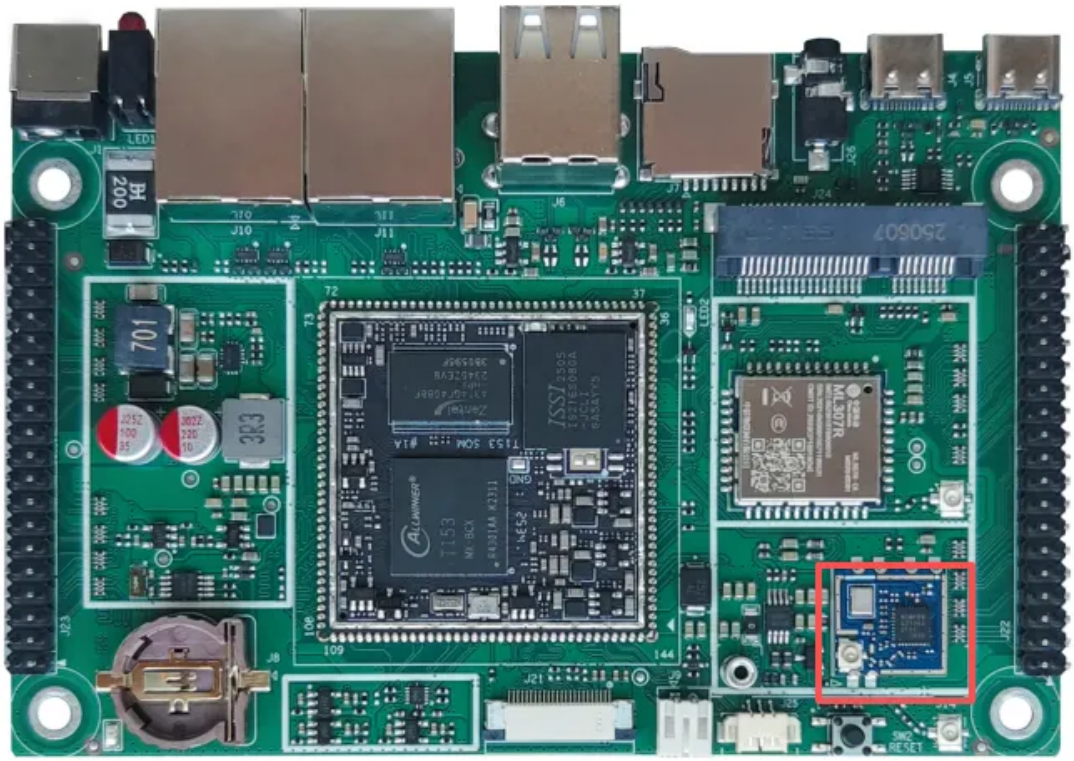
注意：

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。

9. 星闪模块

主板配置了1路星闪模块，如下图所示：



9.1. 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
# ifconfig Featureid0 down
```

账号密码配置：

```Bash
# cat /etc/wpa_supplicant.conf
ctrl_interface=/var/run/wpa_supplicant
ap_scan=1
update_config=1

network={
  ssid="SSID"                                  #账号
  psk="PASSWOWD"                               #密码
  key_mgmt=WPA-PSK
}
```

ssid：账号。

psk：密码。

若需要连接WPA3加密的WiFi，可以参考如下配置

```Bash
ctrl_interface=/var/run/wpa_supplicant
ap_scan=1
update_config=1

network={
        ssid="CS_2.4G_WPA3"
        sae_password="CS123123"
        pairwise=CCMP TKIP
        group=CCMP TKIP
        key_mgmt=SAE WPA-PSK
        scan_ssid=1
        ieee80211w=2
}
```

连接方法：

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

获取IP：

```Bash
# dhclient wlan0
udhcpc: started, v1.36.1
udhcpc: broadcasting discover
udhcpc: broadcasting select for 192.168.74.160, server 192.168.74.123
udhcpc: lease of 192.168.74.160 obtained from 192.168.74.123, lease time 3599
deleting routers
adding dns 192.168.74.123

# ifconfig wlan0
wlan0     Link encap:Ethernet  HWaddr 78:22:88:85:63:96
          inet addr:192.168.74.160  Bcast:192.168.74.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:77 errors:0 dropped:0 overruns:0 frame:0
          TX packets:94 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:6190 (6.0 KiB)  TX bytes:11915 (11.6 KiB)
```

测试：

```Bash
# ping www.baidu.com -I wlan0
PING www.baidu.com (103.235.46.102): 56 data bytes
64 bytes from 103.235.46.102: seq=0 ttl=48 time=266.668 ms
64 bytes from 103.235.46.102: seq=1 ttl=48 time=233.898 ms
64 bytes from 103.235.46.102: seq=2 ttl=48 time=276.997 ms
64 bytes from 103.235.46.102: seq=3 ttl=48 time=229.841 ms
```

9.2. Bluetooth

打开蓝牙设备：

```Shell
$ hciconfig hci0 up
$ bluetoothd  &
```

扫描蓝牙设备：

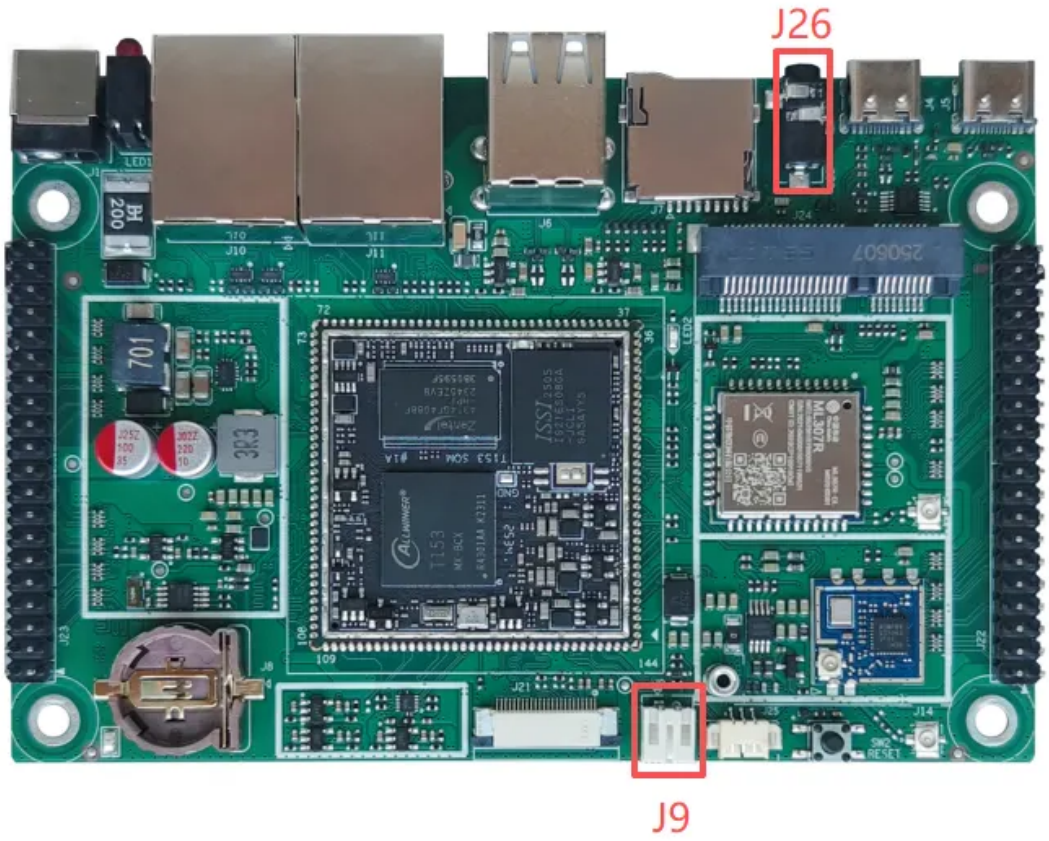
```Shell
$ hciconfig hci0 iscan

$ bluetoothctl
Agent registered
[CHG] Controller 10:BB:F3:56:44:55 Pairable: yes
[bluetooth]# scan on
Discovery started
[CHG] Controller 10:BB:F3:56:44:55 Discovering: yes
[CHG] Device 24:4C:AB:09:98:A6 RSSI: -92
...
[NEW] Device 7C:C1:80:09:DD:6C AirPods - Find My
...

```

通过mac配对蓝牙设备：

```Shell
[bluetooth]# trust 7C:C1:80:09:DD:6C
[CHG] Device 7C:C1:80:09:DD:6C Trusted: yes
Changing 7C:C1:80:09:DD:6C trust succeeded
[bluetooth]# pair 7C:C1:80:09:DD:6C
Attempting to pair with 7C:C1:80:09:DD:6C
[CHG] Device 7C:C1:80:09:DD:6C Connected: yes
[CHG] Device 7C:C1:80:09:DD:6C Name: AirPods
[CHG] Device 7C:C1:80:09:DD:6C Alias: AirPods
[CHG] Device 7C:C1:80:09:DD:6C Modalias: bluetooth:v004Cp2013dB087
[CHG] Device 7C:C1:80:09:DD:6C UUIDs: 00001000-0000-1000-8000-00805f9b34fb
[CHG] Device 7C:C1:80:09:DD:6C UUIDs: 0000110b-0000-1000-8000-00805f9b34fb
[CHG] Device 7C:C1:80:09:DD:6C UUIDs: 0000110c-0000-1000-8000-00805f9b34fb
[CHG] Device 7C:C1:80:09:DD:6C UUIDs: 0000110e-0000-1000-8000-00805f9b34fb
[CHG] Device 7C:C1:80:09:DD:6C UUIDs: 0000111e-0000-1000-8000-00805f9b34fb
[CHG] Device 7C:C1:80:09:DD:6C UUIDs: 00001200-0000-1000-8000-00805f9b34fb
[CHG] Device 7C:C1:80:09:DD:6C UUIDs: 74ec2172-0bad-4d01-8f77-997b2be0722a
[CHG] Device 7C:C1:80:09:DD:6C ServicesResolved: yes
[CHG] Device 7C:C1:80:09:DD:6C Paired: yes
Pairing successful
[AirPods - Find My]# exit
```

10. 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	J9	Speaker
2	J26	Line Out

声卡节点

```Shell
#声卡节点
# aplay -l
**** List of PLAYBACK Hardware Devices ****
card 0: audiocodec [audiocodec], device 0: sunxi-snd-plat-aaudio-sunxi-snd-codec 2030000.codec-0 []
  Subdevices: 1/1
  Subdevice #0: subdevice #0


```

声卡开关

```Shell
#声卡控制节点
# amixer -c 0 controls
numid=2,iface=MIXER,name='DAC DRC Mode'
numid=3,iface=MIXER,name='DAC HPF Mode'
numid=6,iface=MIXER,name='DAC Output Select'
numid=4,iface=MIXER,name='DAC Volume'
numid=5,iface=MIXER,name='DACL Volume'
numid=7,iface=MIXER,name='LINEOUT Gain'
numid=8,iface=MIXER,name='LINEOUTL Switch'
numid=9,iface=MIXER,name='SPK Switch'
numid=1,iface=MIXER,name='tx hub mode'

#打开耳机开关
# amixer -c 0 cset numid=8,iface=MIXER,name='LINEOUTL Switch' on
numid=8,iface=MIXER,name='LINEOUTL Switch'
  ; type=BOOLEAN,access=rw------,values=1
  : values=on

#打开功放开关
# amixer -c 0 cset numid=9,iface=MIXER,name='SPK Switch' on
numid=9,iface=MIXER,name='SPK Switch'
  ; type=BOOLEAN,access=rw------,values=1
  : values=on
#

```

注：由于喇叭是由lineout音频信号经过功放转换而来，所以在测试喇叭音频时，需先打开耳机开关。

音频播放

```shell
#音频播放
# aplay -Dhw:0,0 ./test_mono.wav
Playing WAVE './test_mono.wav' : Signed 16 bit Little Endian, Rate 44100 Hz, Mono
```

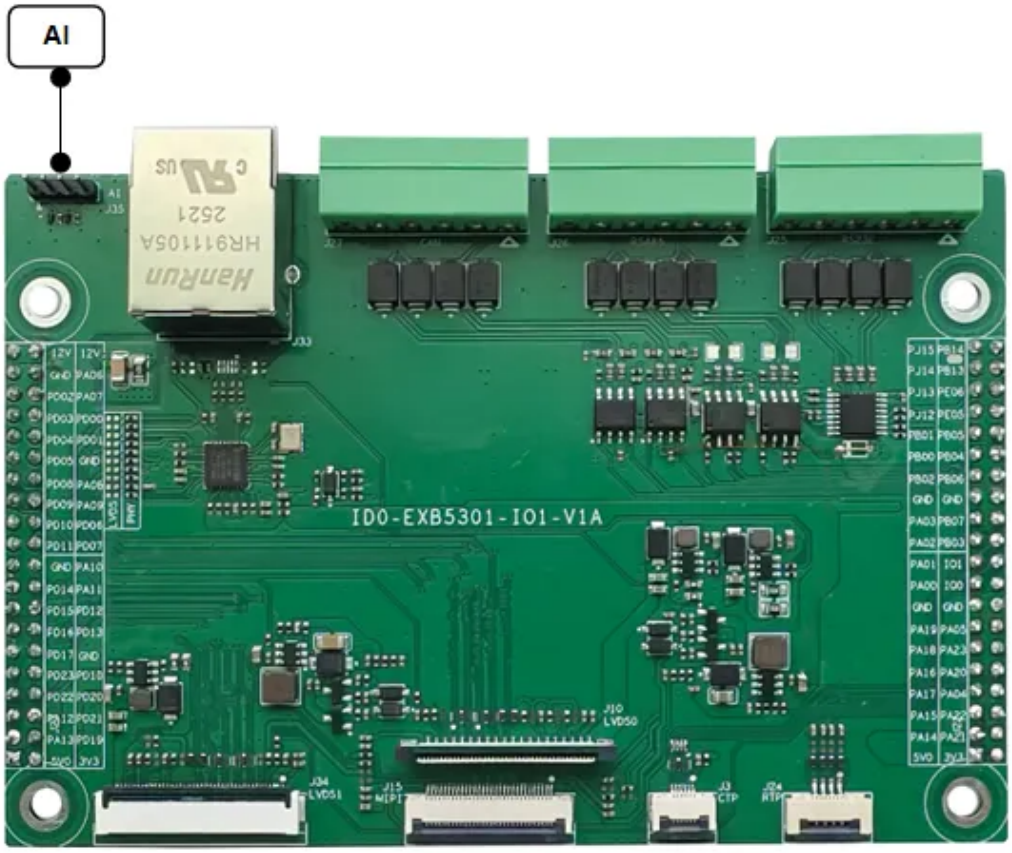
注：由于T153默认声卡只支持单声道音频，播放音频若出现 `aplay:set_params:1358:channels count non available` 等相关报错信息，可通过 `ffmpeg` 将其转换成默认支持的音频。

```shell
$ ffmpeg -i ./8k16bpsStereo.wav -ac 1 8k16bpsStereo_mono.wav
```

`8k16bpsStereo.wav` 为音频源文件， `8k16bpsStereo_mono.wav` 为转换后输出的音频文件（开发板系统默认不带ffmpeg工具，需在其他平台上转换）

11. ADC

拓展板带有两路adc捕获，位于拓展板J35处，如下图



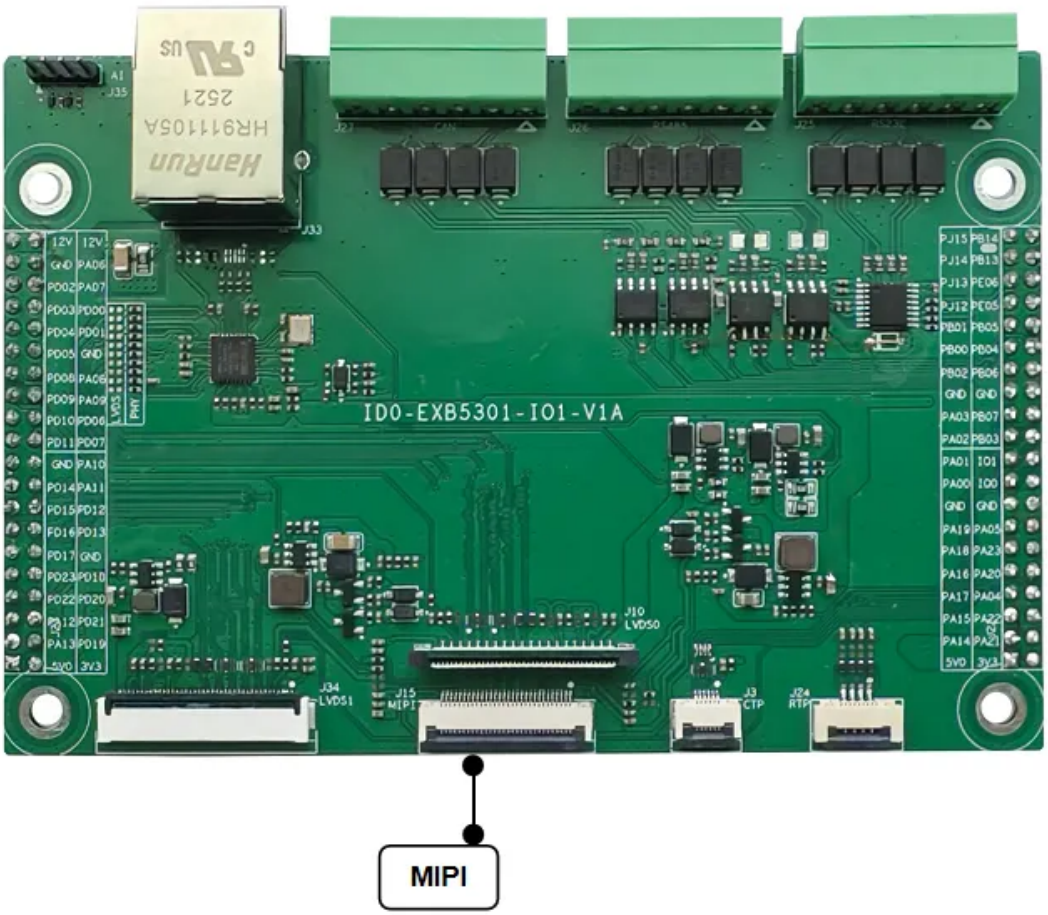
四路引脚从左往右依次为 AI1+，AI1−，AI2+，AI2−

AI捕获（单位为mv） Shell

```
$ cd /sys/class/gpadc/gpadc_chip0
$ echo gpadc6,1 > status
$ echo gpadc7,1 > status

# 捕获AI1的值
$ echo 6 > data ;cat data
gpadc0-channel6 voltage data is 1792
# 捕获AI2的值
$ echo 7 > data ;cat data
gpadc0-channel7 voltage data is 1791
```

12. 显示接口

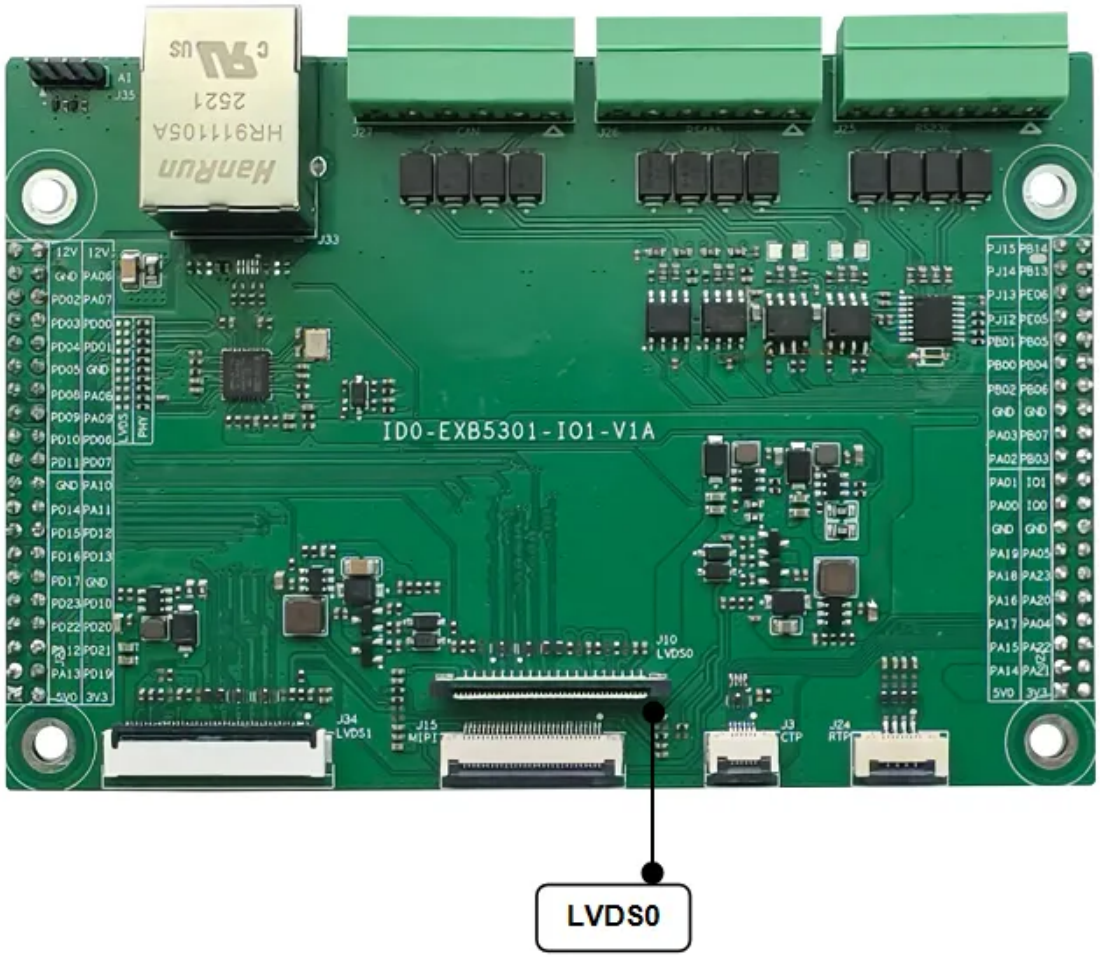
12.1. MIPI



分辨率：1024x600

默认屏幕是否支持触摸：是

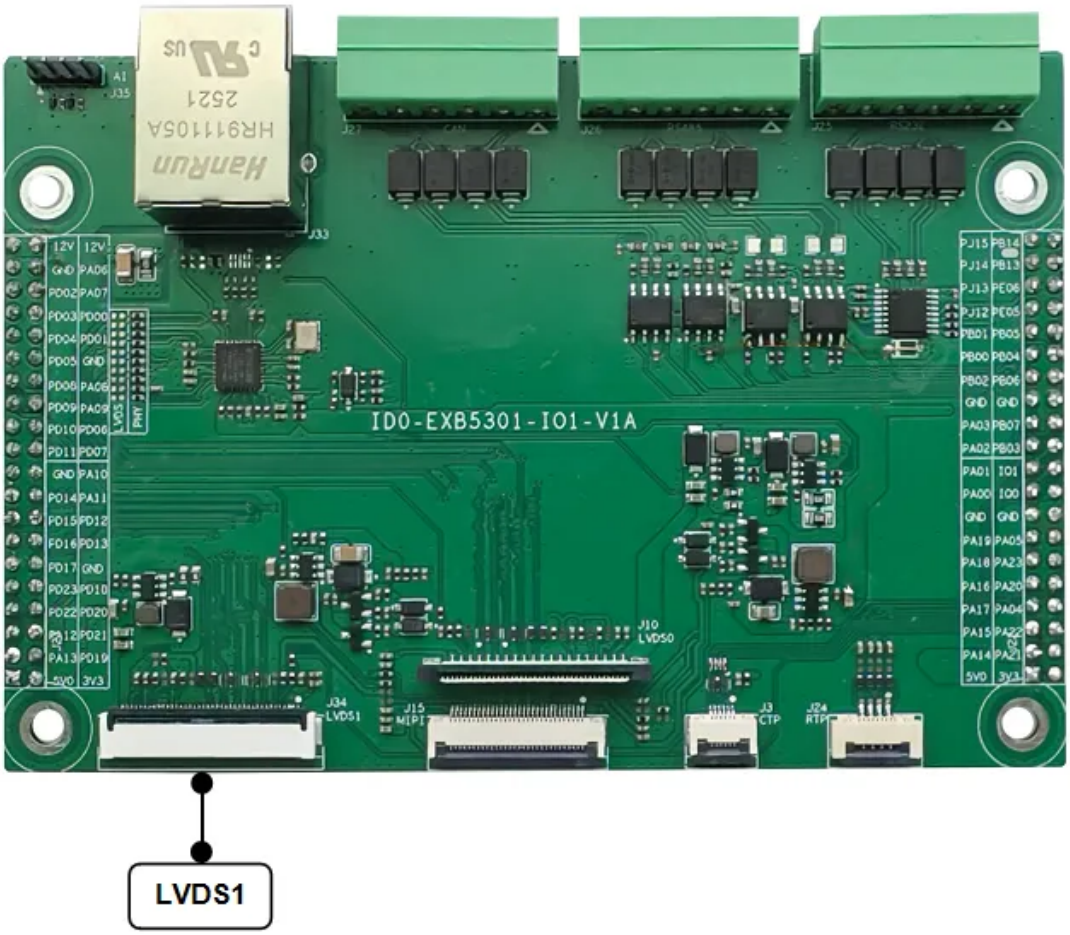
12.2. LVDS0屏幕



分辨率：1024x600

默认屏幕是否支持触摸：否

12.3. LVDS1屏幕



分辨率：1024x600

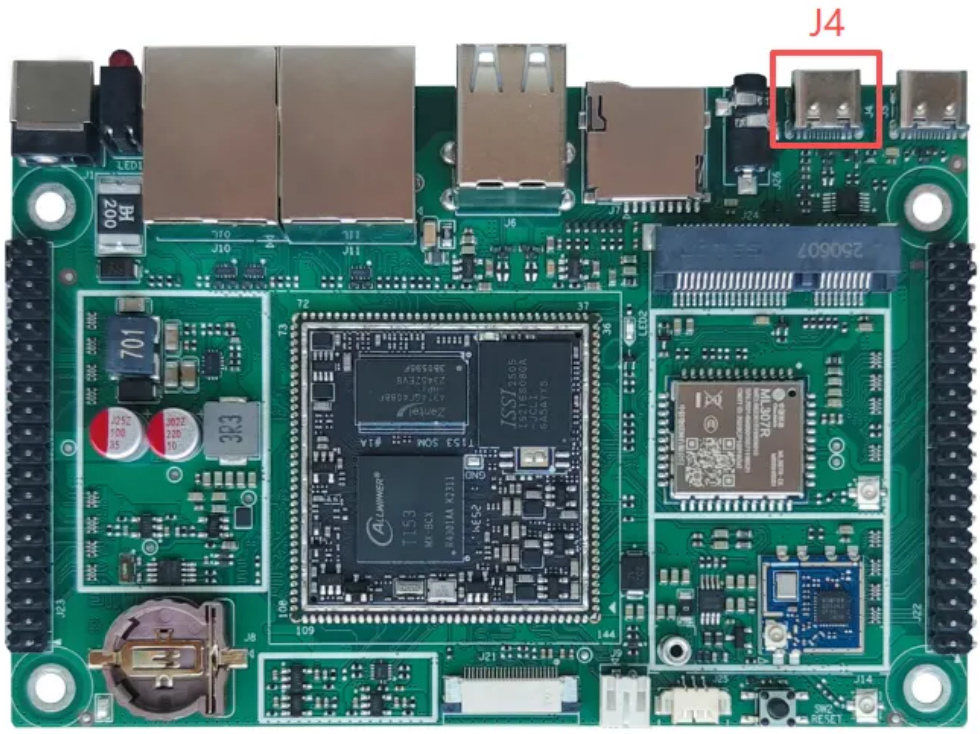
默认屏幕是否支持触摸：否

Ubuntu使用手册

1. 调试接口

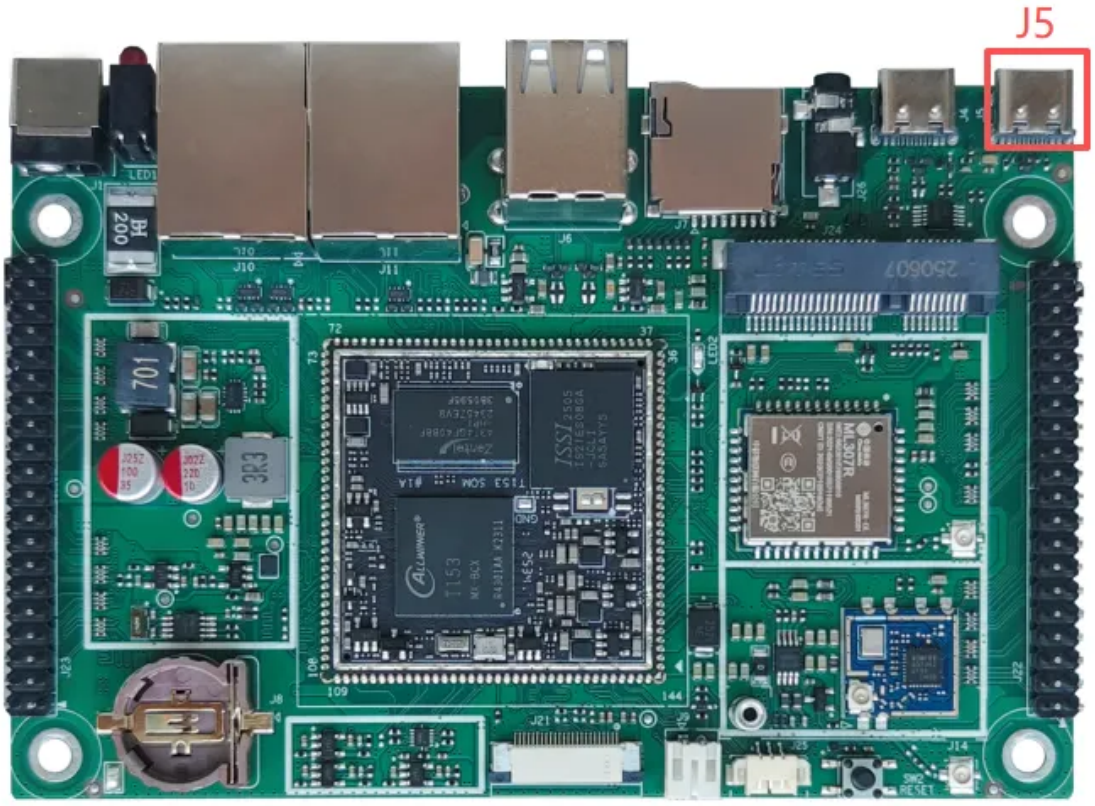
1.1. adb

开发板adb调试口位于主板J4处



1.2. 调试串口

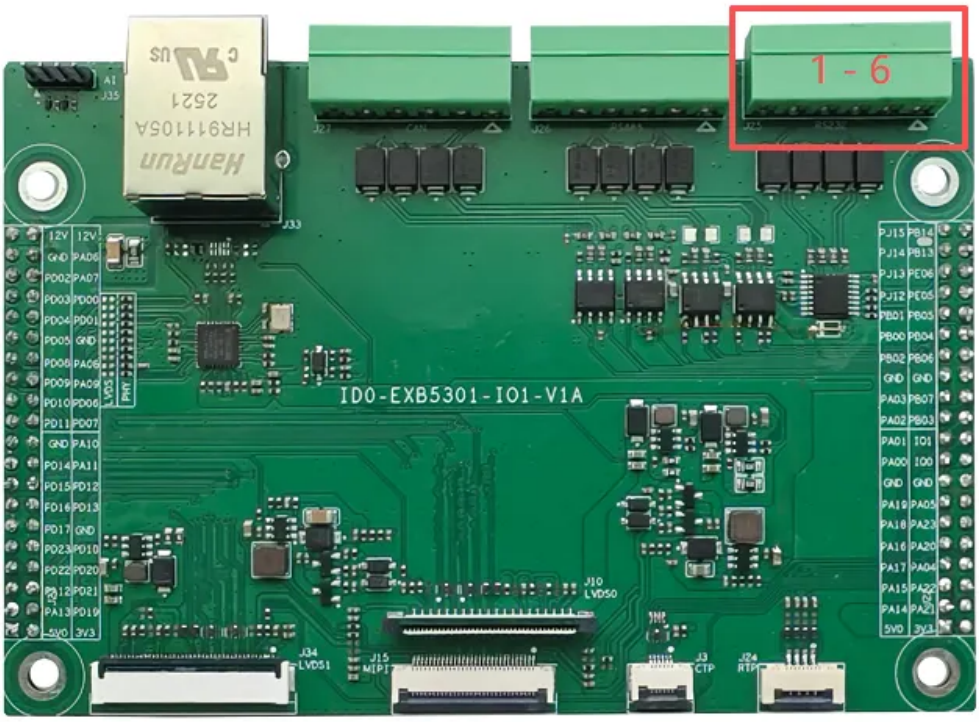
开发板debug串口位于J5处，波特率：115200



2. UART

主板共配置4路串口，包括2路RS232和2路RS485（除调试串口），如下所示。

2.1. RS232



序号	接口位置	电平类型	串口设备节点	备注
1	J25	RS232	/dev/ttyAS5	/
2		RS232	/dev/ttyAS7	/

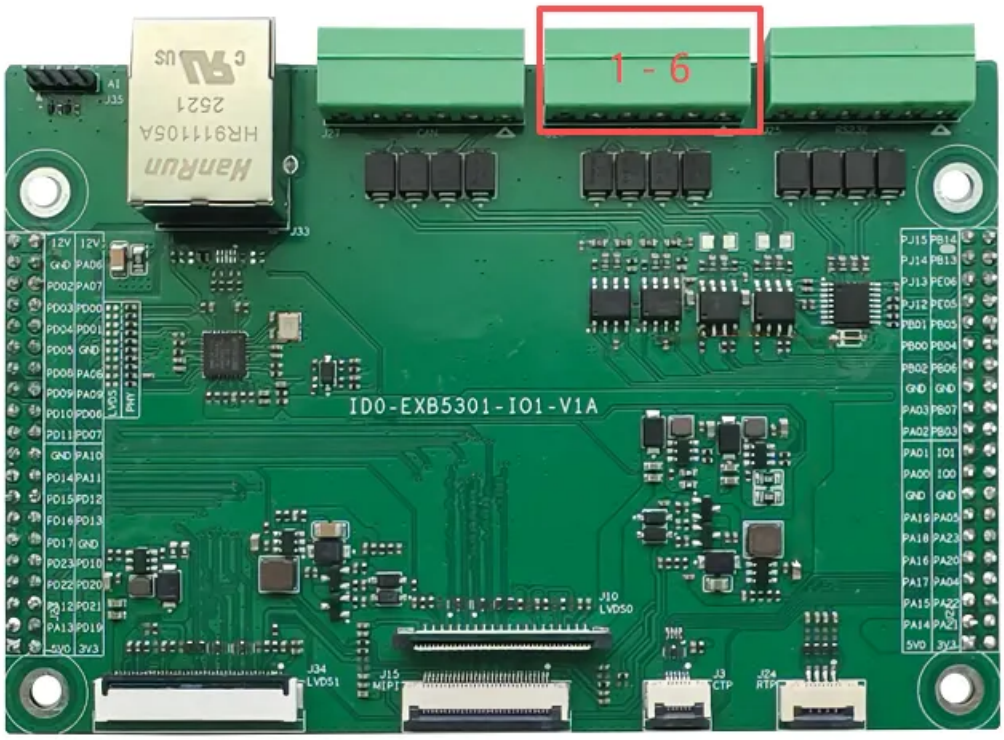
引脚定义：

序号	定义	电平/V	说明
1	GND	GND	电源地
2	TX7	/	RS232_TX
3	RX7	/	RS232_RX
4	TX5	/	RS232_TX
5	RX5	/	RS232_RX
6	5V	5V	电源5V

收发测试

```Bash
# cat /dev/ttyAS5 &
[1] 453
# echo 123123 > /dev/ttyAS7
123123

# echo 123123 > /dev/ttyAS7
123123

# kill 453
# cat /dev/ttyAS7 &
[2] 454
[1]   Terminated              cat /dev/ttyAS5
# echo 123123 > /dev/ttyAS5
123123

# echo 123123 > /dev/ttyAS5
123123

# echo 123123 > /dev/ttyAS5
123123

#
```

2.2. RS485



序号	接口位置	电平类型	串口设备节点	备注
1	J26	RS485	/dev/ttyAS2	
2		RS485	/dev/ttyAS8	

引脚定义：

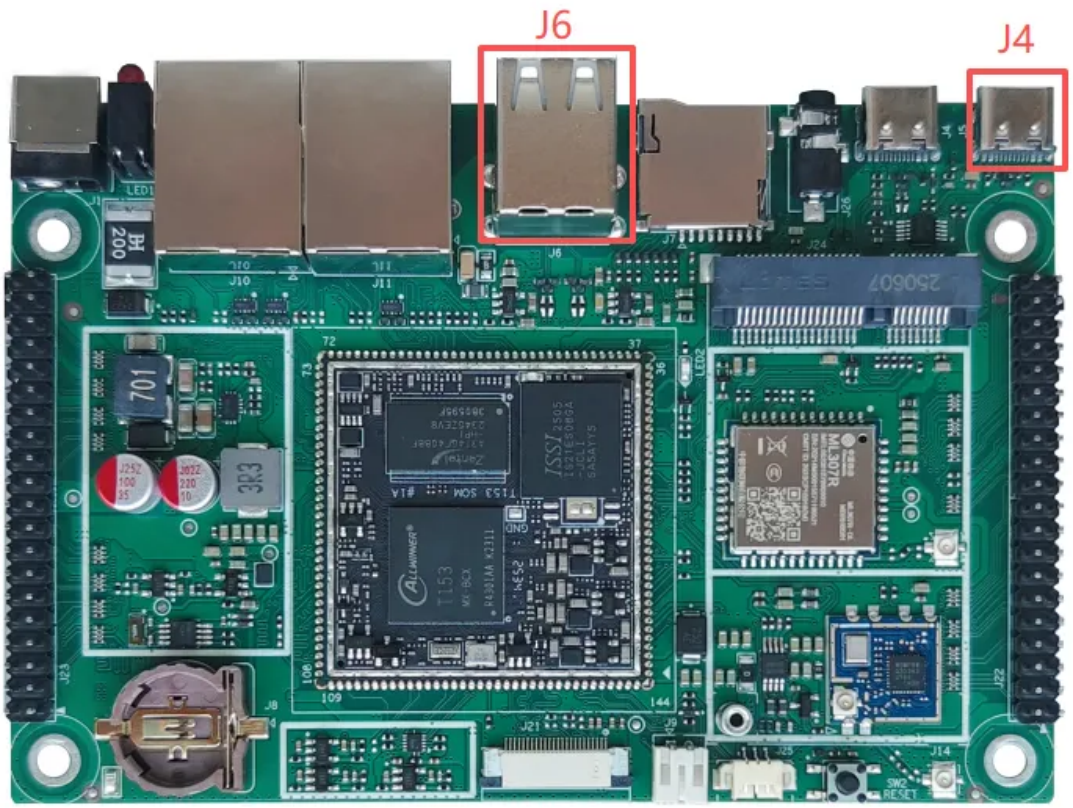
序号	定义	电平/V	说明
1	GND	GND	电源地
2	RS485_B8	/	RS485_B
3	RS485_A8	/	RS485_A
4	RS485_B2	/	RS485_B
5	RS485_A2	/	RS485_A

6	5V	5V	电源5V

rs485可以通过《IDO-EVB5301-V1 串口开发手册》进行验证

3. 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	J4	OTG
USB2	J6-上	USB2.0 HOST
USB3	J6-下	USB2.0 HOST

接上U盘后，系统出现如下打印

```Bash
[ 2487.378996] usb 1-1.2: new high-speed USB device number 9 using sunxi-ehci
[ 2487.533172] usb 1-1.2: New USB device found, idVendor=0951, idProduct=1666, bcdDevice= 2.00
[ 2487.533186] usb 1-1.2: New USB device strings: Mfr=2, Product=3, SerialNumber=4
[ 2487.533195] usb 1-1.2: Product: DataTraveler 3.0
[ 2487.533203] usb 1-1.2: Manufacturer: Kingston
[ 2487.533210] usb 1-1.2: SerialNumber: E0D55E6C0EC11691B8C1065B
[ 2487.534261] usb-storage 1-1.2:1.0: USB Mass Storage device detected
[ 2487.536615] scsi host0: usb-storage 1-1.2:1.0
```

通过 `fdisk -l` 命令可以查看识别出的usb设备

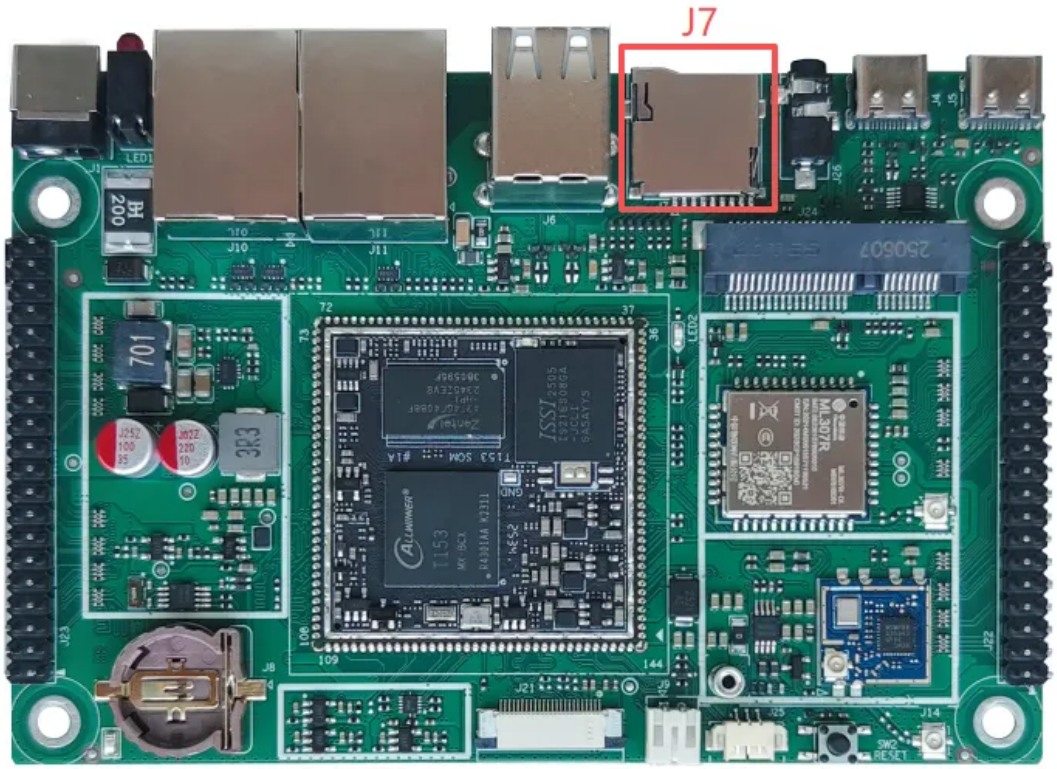
```Bash
# fdisk -l
Found valid GPT with protective MBR; using GPT

……

Disk /dev/sda: 29 GB, 30995907072 bytes, 60538881 sectors
3768 cylinders, 255 heads, 63 sectors/track
Units: sectors of 1 * 512 = 512 bytes

Device  Boot StartCHS    EndCHS        StartLBA     EndLBA    Sectors  Size Id Type
/dev/sda1 *  122,32,39   1023,254,63    1961984   59617280   57655297 27.4G  7 HPFS/NTFS
/dev/sda2    1023,254,63 1023,254,63   59619328   60534783     915456  447M 1b Hidden Win95 FAT32
```

通过 `mount` 手动将U盘挂载至rootfs

```Bash
# mkdir /mnt/usb && mount /dev/sda1 /mnt/usb
```

4. Micro SD

主板共配置一路Micro SD接口，如下图所示：



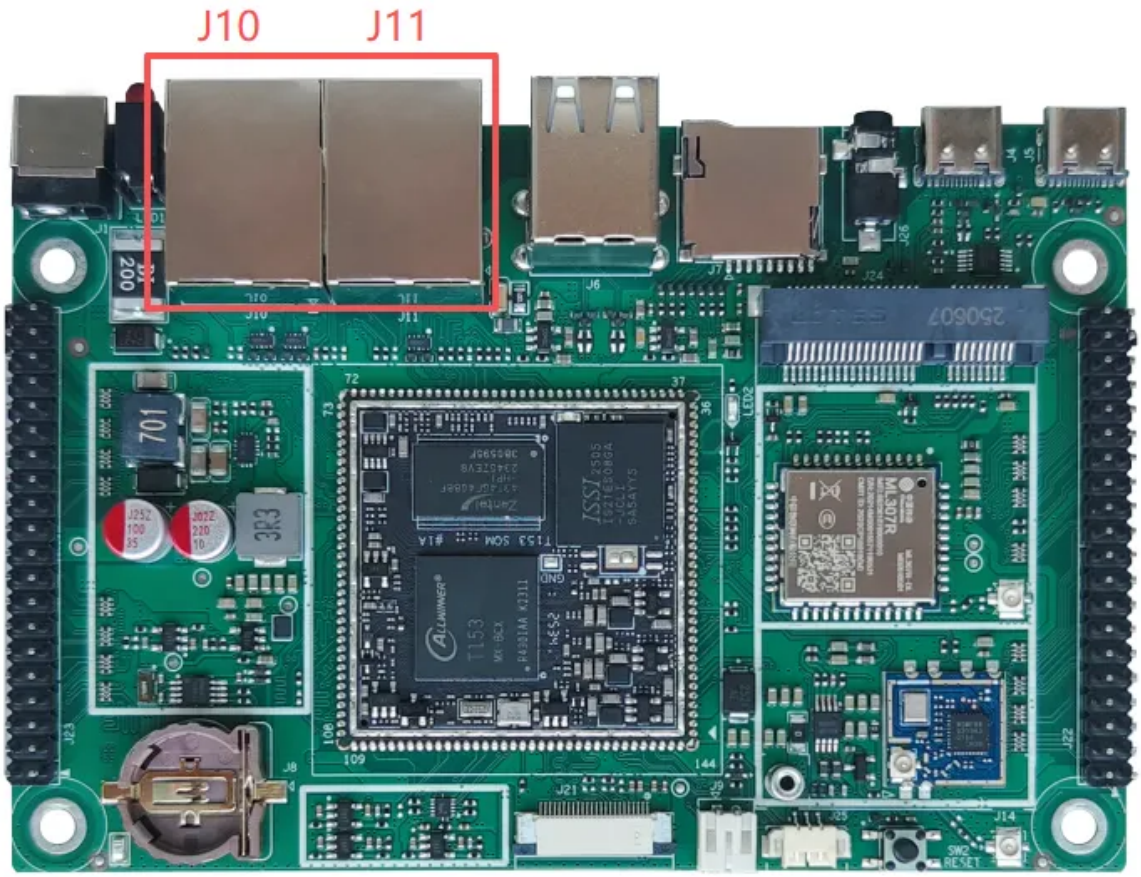
插入SD卡后，默认挂载到/mnt/sdcard目录，如下所示：

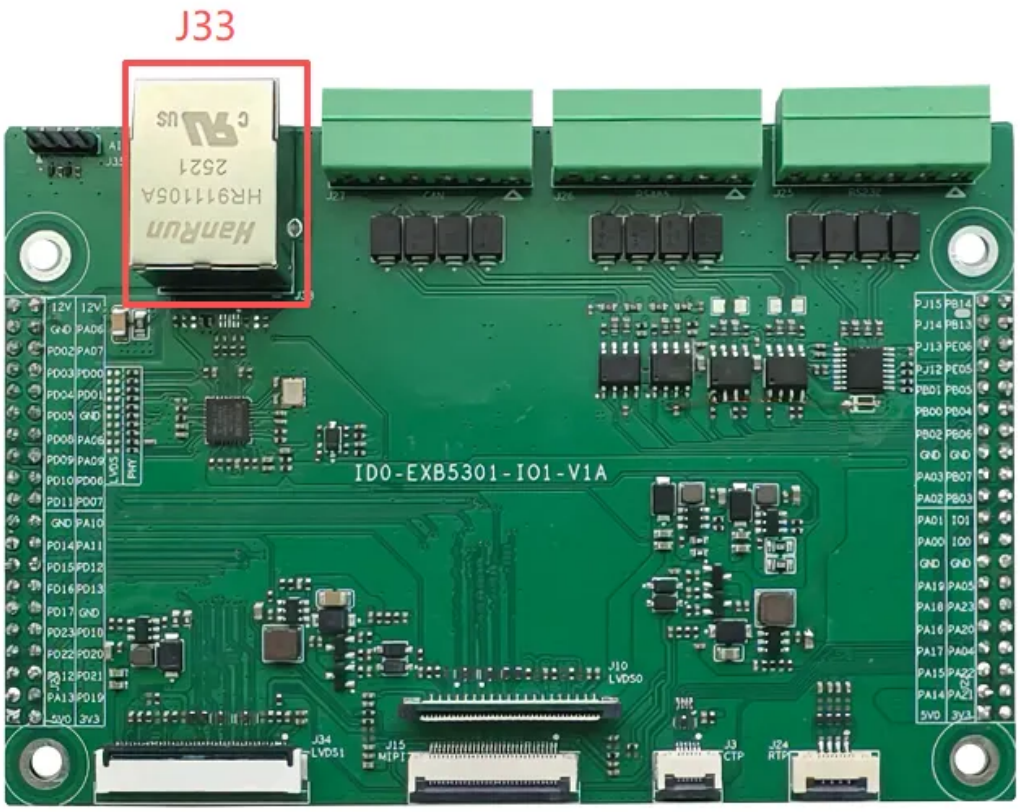
```shell
# mount
...
/dev/mmcblk0p1 on /mnt/sdcard type vfat (rw,nodev,noexec,noatime,nodiratime,fmask=0022,dmask=0022,codepage=437,iocharset=iso8859-1,shortname=mixed,errors=remount-ro)
```

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	✅

5. Ethernet

主板共配置1路千兆以太网接口和2路百兆以太网接口，如下图所示：


J10
J11



序号	接口位置	速率	网络节点
1	J11	百兆	eth0
2	J10	千兆	eth1
3	J33	百兆	eth2

系统默认开启DHCP服务，动态获取IP。

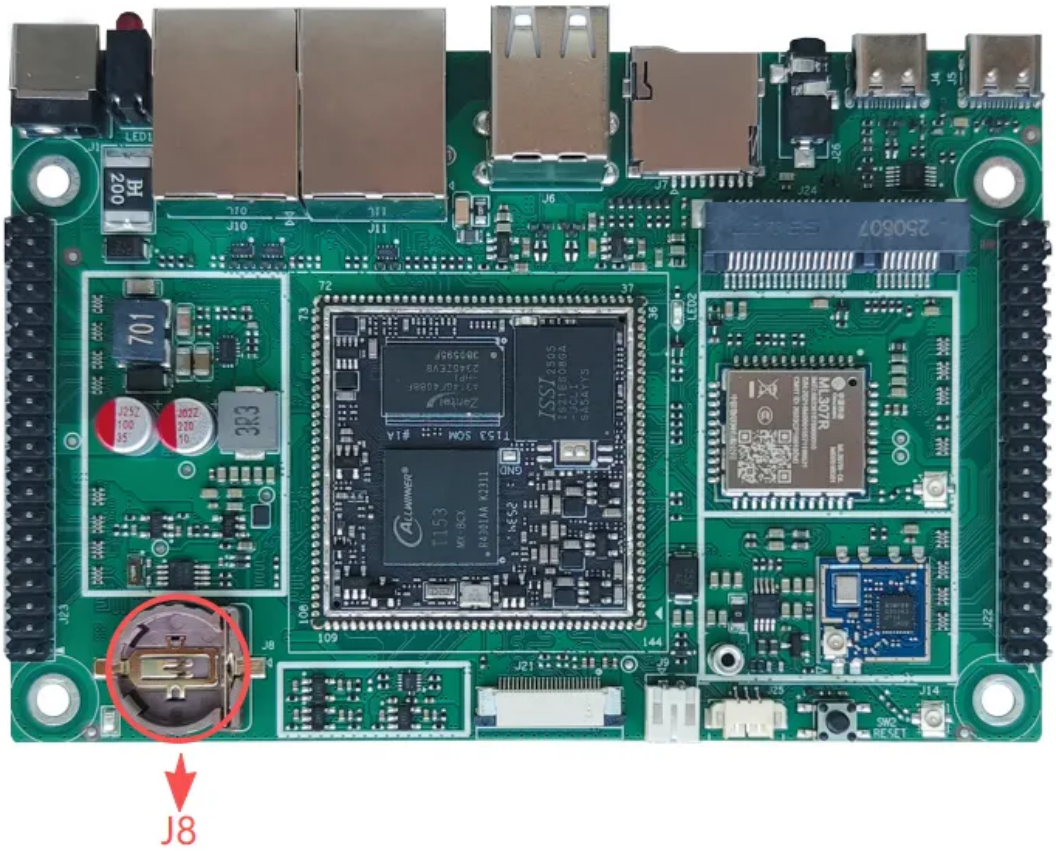
```shell
# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 7A:1F:F2:40:0F:F0
          inet addr:192.168.0.12  Bcast:192.168.0.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:25 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4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

可通过ifconfig指令设置临时静态IP，命令如下：

```shell
# ifconfig eth0 192.168.3.123
```

6. RTC

主板配置了1路rtc接口，如下图所示：



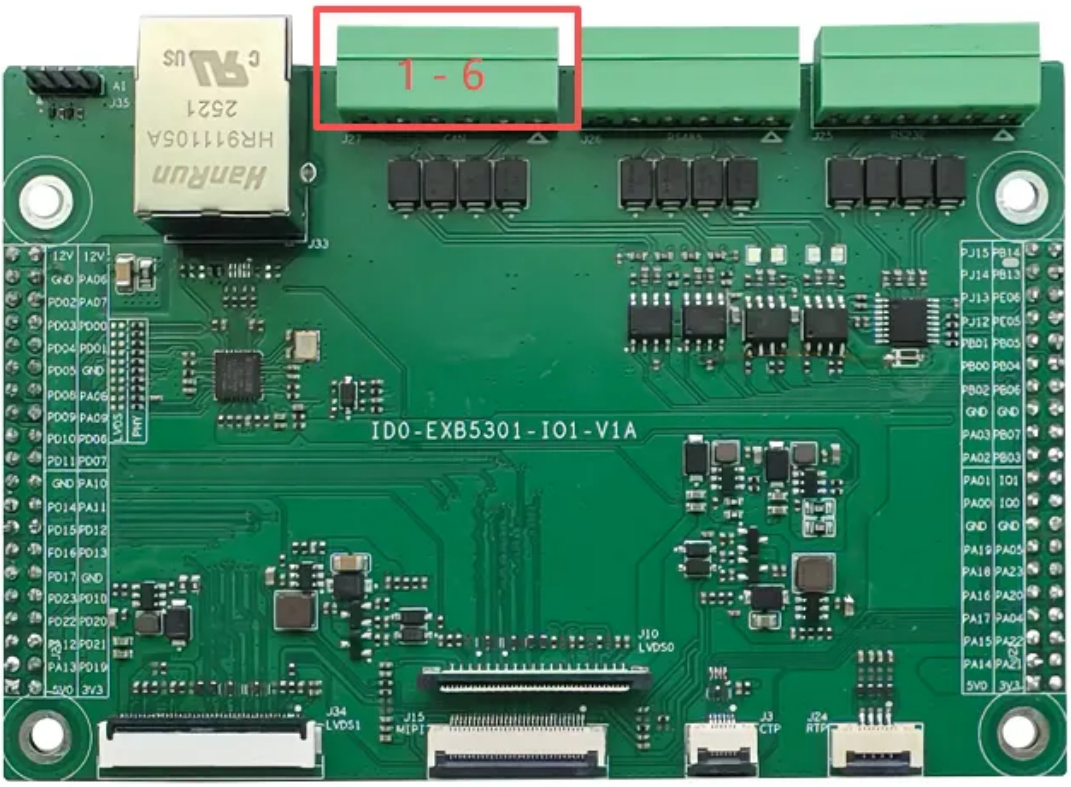
rtc时间设置方法：

```Bash
#设置系统时间
$ date -s "2024-10-09 14:02:30"

#将rtc时间调整为与目前的系统时间一致
$ hwclock -w -f /dev/rtc1

#获取硬件rtc当前时间（断电重启读取时间没有太大偏差）
$ hwclock -r -f /dev/rtc1
2024-10-09 14:02:35.945604+00:00
```

//待补充 rtc0、定时开机

7. 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	J27	CAN	can1	/
2		CAN	can0	/

引脚定义：

序号	定义	电平/V	说明
1	GND	GND	电源地
2	CAN0_L	/	CAN_L
3	CAN0_H	/	CAN_H
4	CAN1_L	/	CAN_L
5	CAN1_H	/	CAN_H
6	5V	5V	电源5V

查看can节点，以can1为例：

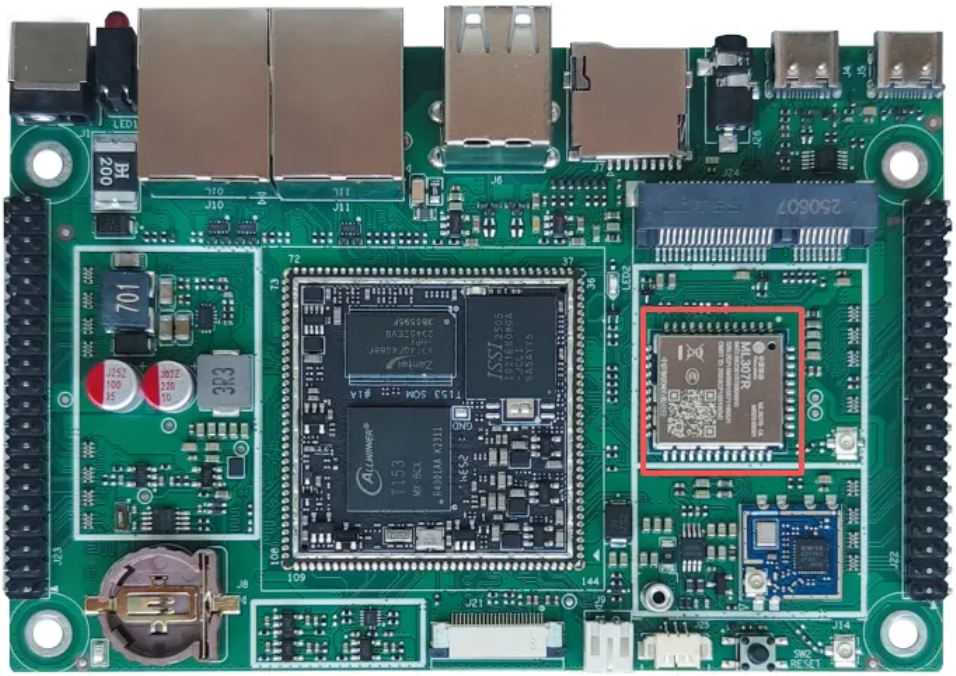
```shell
# ifconfig can0
can0      Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          NOARP  MTU:16  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:10
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
          Interrupt:49
```

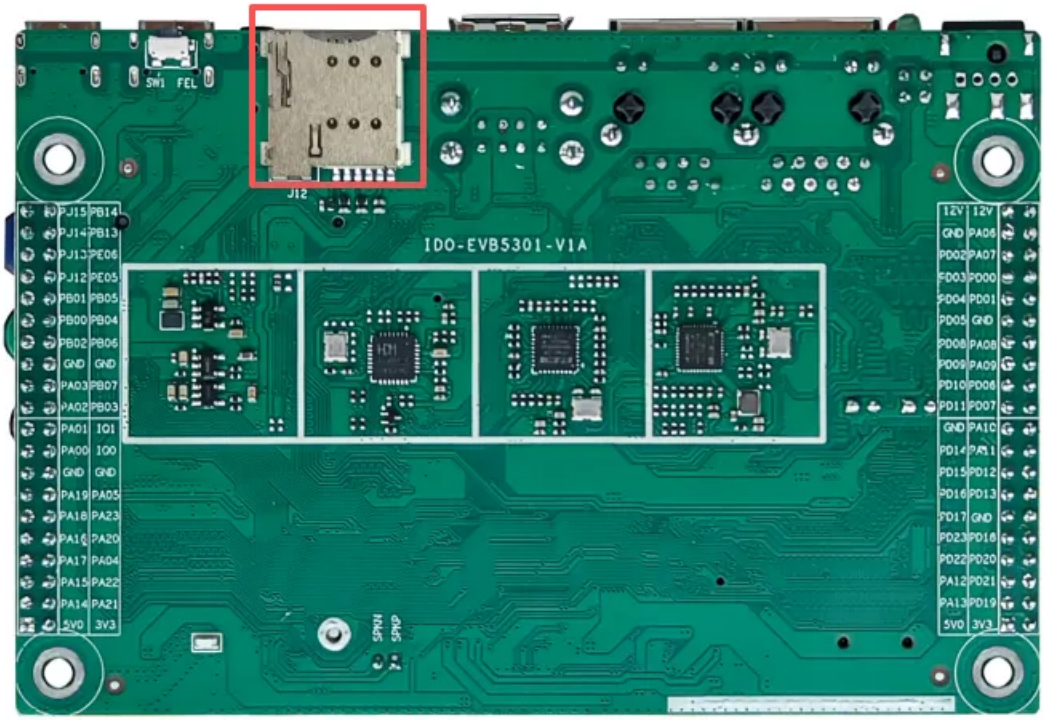
数据收发，以can1为例：

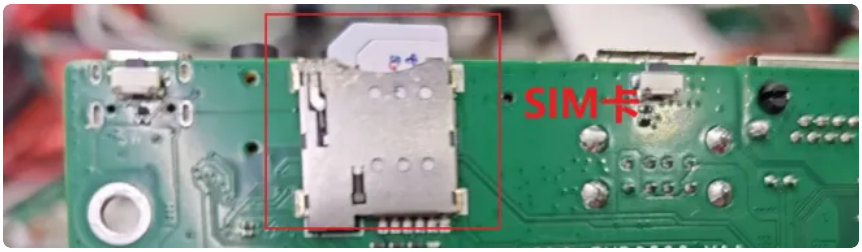
```shell
#检查can设备
$ ip link show

#在收发端关闭can0设备
$ ip link set can0 down

#设置仲裁段1M波特率，数据段1M波特率
$ ip link set can0 type can bitrate 1000000 loopback off
[ 8846.889995] sunxi-can 453c800.can0: PM runtime resume.
[ 8846.895843] sunxi-can 453c800.can0: PM runtime suspend.

#在收发端打开can0设备
$ ip link set can0 up

#在接收端执行candump,阻塞等待报文
$ candump can0

#在发送端执行cansend，发送报文
$ cansend can0 123#1122334455667788

#检查是否启用成功
$ ip link show can0
```

8. 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：

```shell
cat /dev/ttyUSB2 &

#设置拨号协议为RNDIS
echo -e "AT+MDIALUPCFG="mode",0" > /dev/ttyUSB2

#设置拨号协议为ECM
echo -e "AT+MDIALUPCFG="mode",1" > /dev/ttyUSB2

#关闭回显
echo -e 'ATE0\r\n' > /dev/ttyUSB2

#查询PDP上下文配置
echo -e 'AT+CGDCONT?\r\n' > /dev/ttyUSB2

#查询驻网状态
echo -e 'AT+COPS?\r\n' > /dev/ttyUSB2

#设置自动拨号命令
echo -e 'AT+MDIALUP=1,1\r\n' > /dev/ttyUSB2

#查询拨号状态
echo -e 'AT+MDIALUP?\r\n' > /dev/ttyUSB2
```

查看网络节点：

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

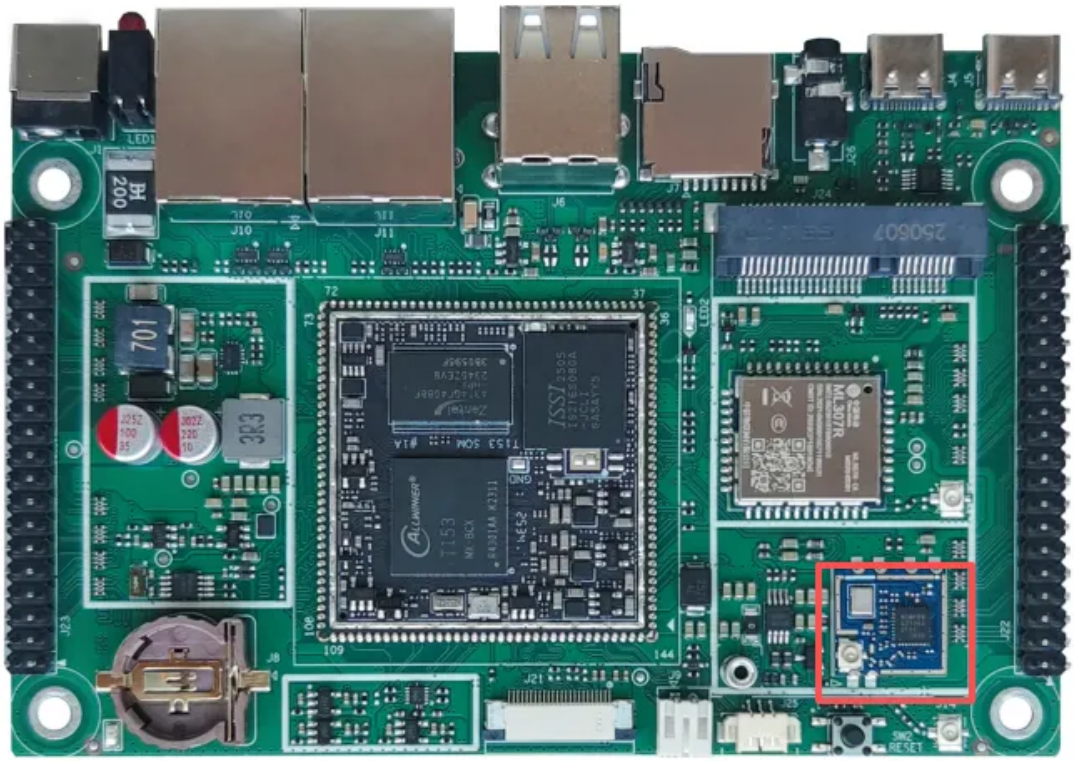
注意：

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。

9. 星闪模块

主板配置了1路星闪模块，如下图所示：



9.1. 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

```
# ifconfig Featureid0 down
```

账号密码配置：

```Bash
# cat /etc/wpa_supplicant.conf
ctrl_interface=/var/run/wpa_supplicant
ap_scan=1
update_config=1

network={
  ssid="SSID"                                  #账号
  psk="PASSWOWD"                               #密码
  key_mgmt=WPA-PSK
}
```

ssid：账号。

psk：密码。

若需要连接WPA3加密的WiFi，可以参考如下配置

```Bash
ctrl_interface=/var/run/wpa_supplicant
ap_scan=1
update_config=1

network={
        ssid="CS_2.4G_WPA3"
        sae_password="CS123123"
        pairwise=CCMP TKIP
        group=CCMP TKIP
        key_mgmt=SAE WPA-PSK
        scan_ssid=1
        ieee80211w=2
}
```

连接方法：

```Bash
# killall wpa_supplicant
# wpa_supplicant -Dnl80211 -c /etc/wpa_supplicant.conf -i wlan0 &
```

获取IP：

```Bash
# dhclient wlan0
udhcpc: started, v1.36.1
udhcpc: broadcasting discover
udhcpc: broadcasting select for 192.168.74.160, server 192.168.74.123
udhcpc: lease of 192.168.74.160 obtained from 192.168.74.123, lease time 3599
deleting routers
adding dns 192.168.74.123

# ifconfig wlan0
wlan0     Link encap:Ethernet  HWaddr 78:22:88:85:63:96
          inet addr:192.168.74.160  Bcast:192.168.74.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:77 errors:0 dropped:0 overruns:0 frame:0
          TX packets:94 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:6190 (6.0 KiB)  TX bytes:11915 (11.6 KiB)
```

测试：

```Bash
# ping www.baidu.com -I wlan0
PING www.baidu.com (103.235.46.102): 56 data bytes
64 bytes from 103.235.46.102: seq=0 ttl=48 time=266.668 ms
64 bytes from 103.235.46.102: seq=1 ttl=48 time=233.898 ms
64 bytes from 103.235.46.102: seq=2 ttl=48 time=276.997 ms
64 bytes from 103.235.46.102: seq=3 ttl=48 time=229.841 ms
```

9.2. Bluetooth

打开蓝牙设备：

```shell
$ hciconfig hci0 up
$ bluetoothd  &
```

扫描蓝牙设备：

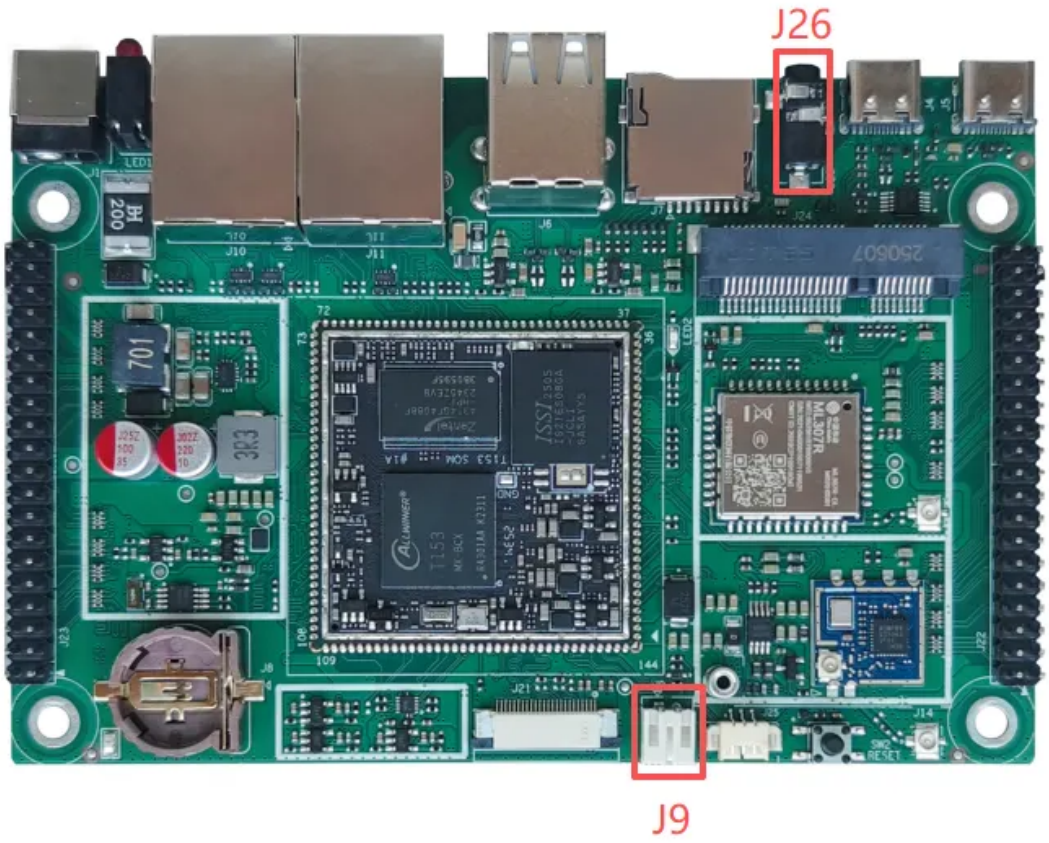
```shell
$ hciconfig hci0 iscan

$ bluetoothctl
Agent registered
[CHG] Controller 10:BB:F3:56:44:55 Pairable: yes
[bluetooth]# scan on
Discovery started
[CHG] Controller 10:BB:F3:56:44:55 Discovering: yes
[CHG] Device 24:4C:AB:09:98:A6 RSSI: -92
...
[NEW] Device 7C:C1:80:09:DD:6C AirPods - Find My
...

```

通过mac配对蓝牙设备：

```Shell
[bluetooth]# trust 7C:C1:80:09:DD:6C
[CHG] Device 7C:C1:80:09:DD:6C Trusted: yes
Changing 7C:C1:80:09:DD:6C trust succeeded
[bluetooth]# pair 7C:C1:80:09:DD:6C
Attempting to pair with 7C:C1:80:09:DD:6C
[CHG] Device 7C:C1:80:09:DD:6C Connected: yes
[CHG] Device 7C:C1:80:09:DD:6C Name: AirPods
[CHG] Device 7C:C1:80:09:DD:6C Alias: AirPods
[CHG] Device 7C:C1:80:09:DD:6C Modalias: bluetooth:v004Cp2013dB087
[CHG] Device 7C:C1:80:09:DD:6C UUIDs: 00001000-0000-1000-8000-00805f9b34fb
[CHG] Device 7C:C1:80:09:DD:6C UUIDs: 0000110b-0000-1000-8000-00805f9b34fb
[CHG] Device 7C:C1:80:09:DD:6C UUIDs: 0000110c-0000-1000-8000-00805f9b34fb
[CHG] Device 7C:C1:80:09:DD:6C UUIDs: 0000110e-0000-1000-8000-00805f9b34fb
[CHG] Device 7C:C1:80:09:DD:6C UUIDs: 0000111e-0000-1000-8000-00805f9b34fb
[CHG] Device 7C:C1:80:09:DD:6C UUIDs: 00001200-0000-1000-8000-00805f9b34fb
[CHG] Device 7C:C1:80:09:DD:6C UUIDs: 74ec2172-0bad-4d01-8f77-997b2be0722a
[CHG] Device 7C:C1:80:09:DD:6C ServicesResolved: yes
[CHG] Device 7C:C1:80:09:DD:6C Paired: yes
Pairing successful
[AirPods - Find My]# exit
```

10. 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	J9	Speaker
2	J26	Line Out

10.1. 声卡节点

```shell
#声卡节点
# aplay -l
**** List of PLAYBACK Hardware Devices ****
card 0: audiocodec [audiocodec], device 0: sunxi-snd-plat-aaudio-sunxi-snd-codec 2030000.codec-0 []
  Subdevices: 1/1
  Subdevice #0: subdevice #0


```

10.2. 声卡开关

```shell
#声卡控制节点
# amixer -c 0 controls
numid=2,iface=MIXER,name='DAC DRC Mode'
numid=3,iface=MIXER,name='DAC HPF Mode'
numid=6,iface=MIXER,name='DAC Output Select'
numid=4,iface=MIXER,name='DAC Volume'
numid=5,iface=MIXER,name='DACL Volume'
numid=7,iface=MIXER,name='LINEOUT Gain'
numid=8,iface=MIXER,name='LINEOUTL Switch'
numid=9,iface=MIXER,name='SPK Switch'
numid=1,iface=MIXER,name='tx hub mode'

#打开耳机开关
# amixer -c 0 cset numid=8,iface=MIXER,name='LINEOUTL Switch' on
numid=8,iface=MIXER,name='LINEOUTL Switch'
  ; type=BOOLEAN,access=rw------,values=1
  : values=on

#打开功放开关
# amixer -c 0 cset numid=9,iface=MIXER,name='SPK Switch' on
numid=9,iface=MIXER,name='SPK Switch'
  ; type=BOOLEAN,access=rw------,values=1
  : values=on
#

```

注：由于喇叭是由lineout音频信号经过功放转换而来，所以在测试喇叭音频时，需先打开耳机开关。

10.3. 音频播放

```shell
#音频播放
# aplay -Dhw:0,0 ./test_mono.wav
Playing WAVE './test_mono.wav' : Signed 16 bit Little Endian, Rate 44100 Hz, Mono
```

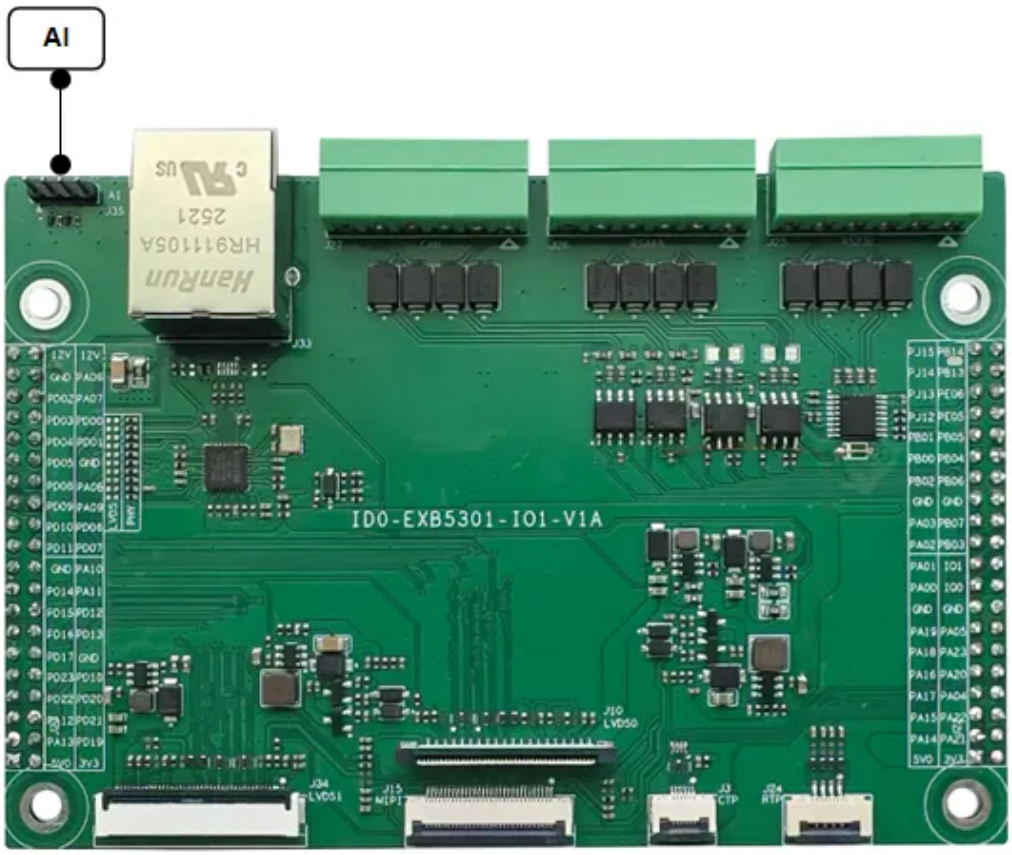
注：由于T153默认声卡只支持单声道音频，播放音频若出现 `aplay:set_params:1358:channels count non available` 等相关报错信息，可通过 `ffmpeg` 将其转换成默认支持的音频。

```shell
$ ffmpeg -i ./8k16bpsStereo.wav -ac 1 8k16bpsStereo_mono.wav
```

`8k16bpsStereo.wav` 为音频源文件， `8k16bpsStereo_mono.wav` 为转换后输出的音频文件（开发板系统默认不带ffmpeg工具，需在其他平台上转换）

11. ADC

拓展板带有两路adc捕获，位于拓展板J35处，如下图



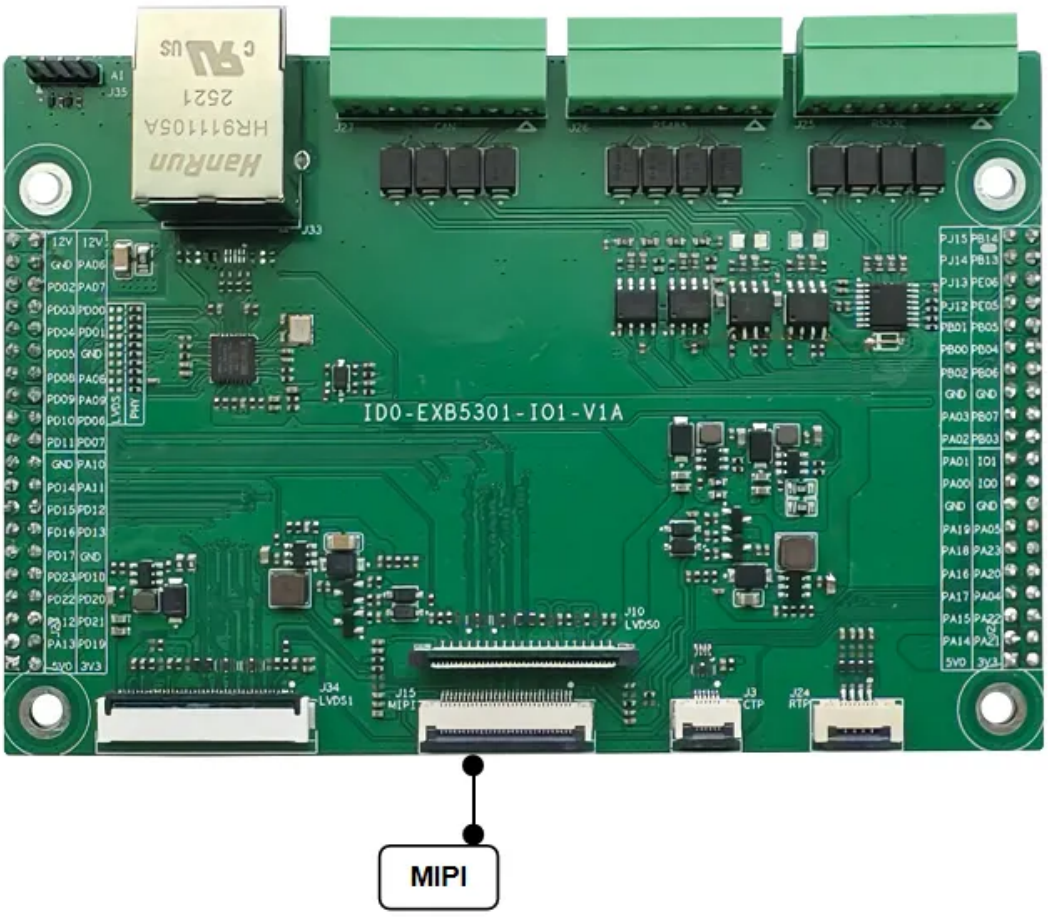
四路引脚从左往右依次为 AI1+，AI1−，AI2+，AI2−

AI捕获（单位为mv） Shell

```
$ cd /sys/class/gpadc/gpadc_chip0
$ echo gpadc6,1 > status
$ echo gpadc7,1 > status

# 捕获AI1的值
$ echo 6 > data ;cat data
gpadc0-channel6 voltage data is 1792
# 捕获AI2的值
$ echo 7 > data ;cat data
gpadc0-channel7 voltage data is 1791
```

12. 显示接口

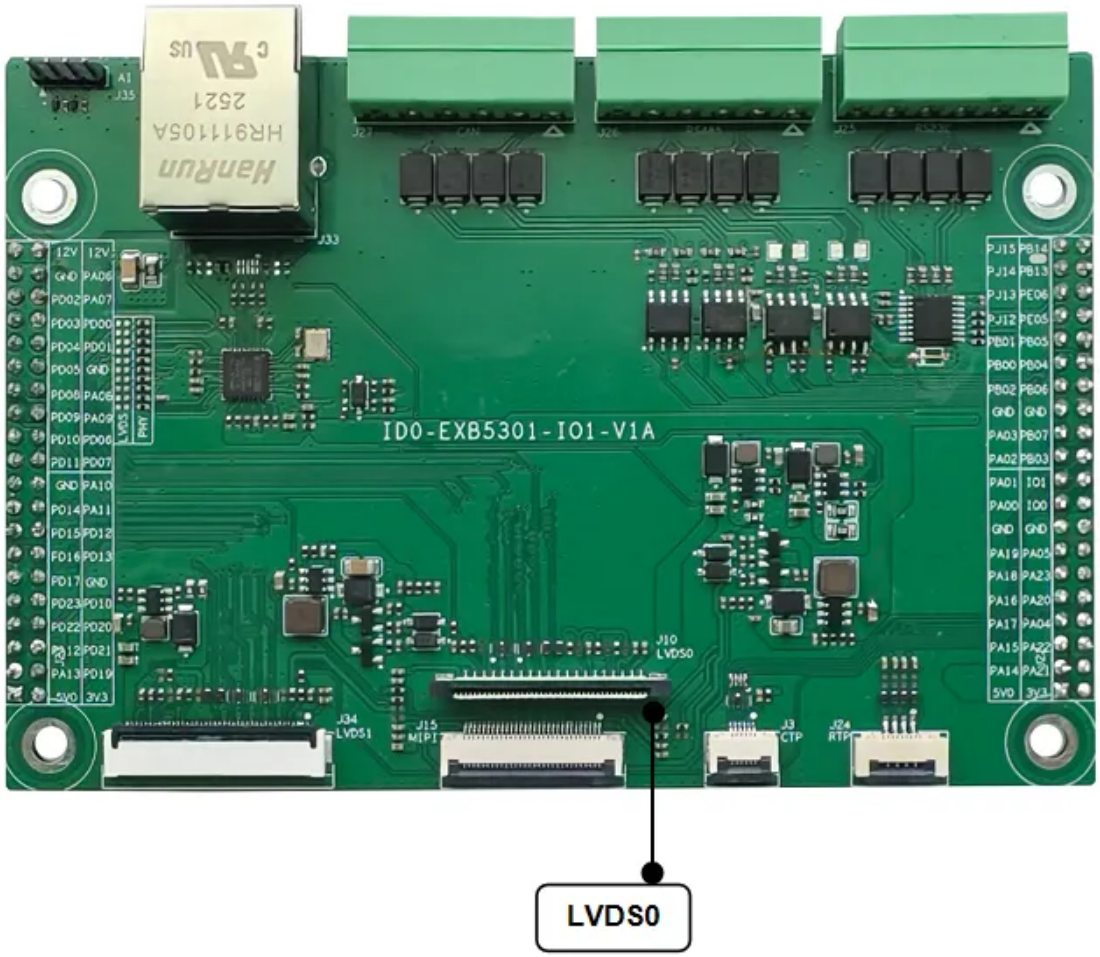
12.1. MIPI



分辨率：1024x600

默认屏幕是否支持触摸：是

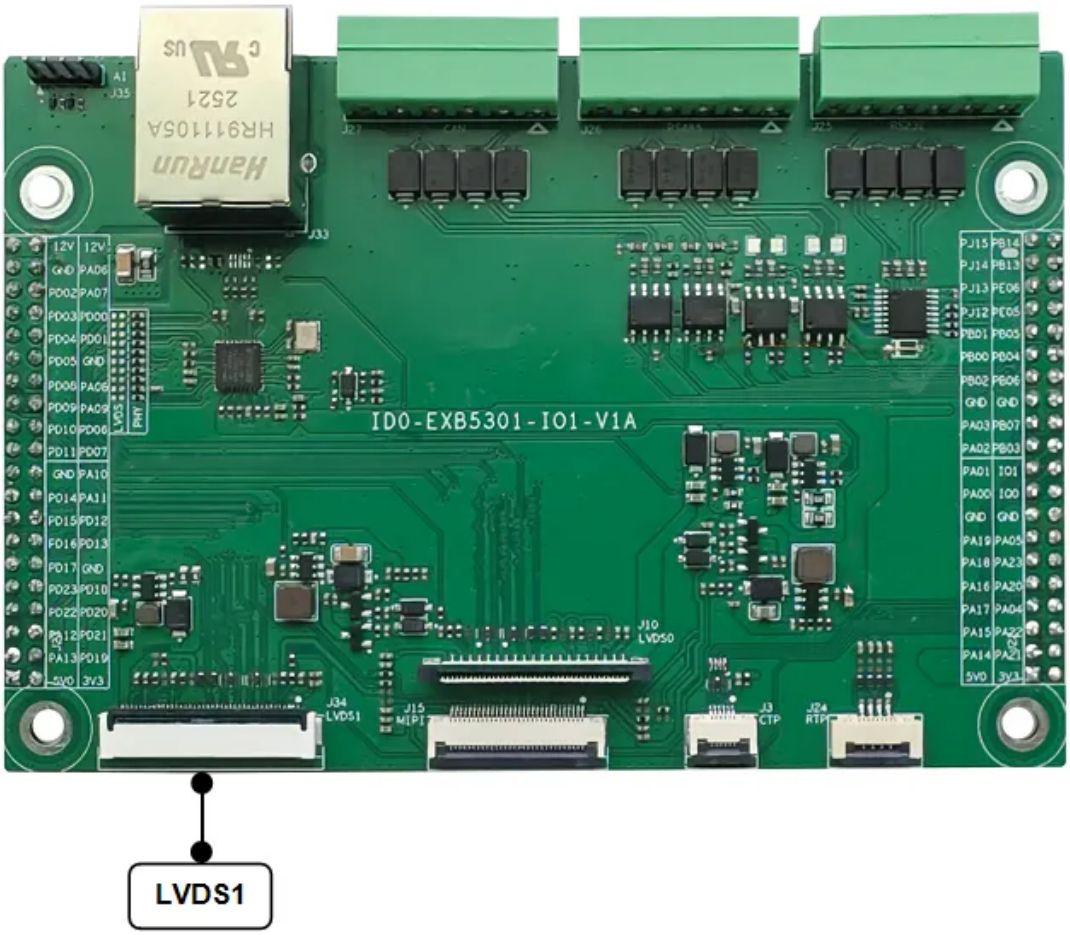
12.2. LVDS0屏幕



分辨率：1024x600

默认屏幕是否支持触摸：否

12.3. LVDS1屏幕



分辨率：1024x600

默认屏幕是否支持触摸：否