

IDO-IPC3528-V1 Linux软件使用手册

整机信息

1. IDO-IPC3528-V1 Debian系统

1.1. 外围功能接口

1.1.1. 固件烧录

1.1.2. 指示灯

1.1.3. REC按键

1.1.4. 耳机

1.1.5. Mic录音

1.1.6. TF卡槽

1.1.7. HDMI

1.1.8. USB

1.1.9. DIDO

DI的使用

DO的使用

1.1.10. CAN

测试

1.1.11. UART

测试

1.2. 网络测试

1.2.1. Ethernet

ETH0

ETH1

1.2.2. 4G/5G 网络4G/5G 拨号

4G/5G 上网测试

1.2.3 WIFI

连接WiFi热点

WiFi上网测试

1.2.4 Bluetooth

2. IDO-IPC3528-V1 Ubuntu系统

2.1. 外围功能接口

2.1.1. 固件烧录

2.1.2. 指示灯

2.1.3. REC按键

2.1.4. 耳机

[查看声卡设备](#)

[播放音频](#)

2.1.5. Mic录音

2.1.6. TF卡槽

2.1.7. HDMI

2.1.8. USB

2.1.9. DIO

[DI的使用](#)

[DO的使用](#)

2.1.10. CAN

[测试](#)

2.1.11. UART

[测试](#)

2.2. 网络测试

2.2.1. Ethernet

[ETH0](#)

[ETH1](#)

2.2.2 4G/5G 网络

[4G/5G 拨号](#)

[4G/5G 上网测试](#)

2.2.3. WIFI

[连接WiFi热点](#)

2.2.4 Bluetooth

3. IDO-IPC3528-V1 Buildroot系统

3.1. 外围功能接口

3.1.1. 固件烧录

3.1.2. 指示灯

3.1.3. REC按键

3.1.4. 耳机

播放音频

查看声卡设备

播放到耳机

3.1.5. Mic录音

3.1.6. TF卡槽

3.1.7. HDMI

3.1.8. USB

3.1.9. DIO

DI的使用

DO的使用

3.1.10. CAN

测试

3.1.11. UART

测试

3.2. 网络测试

3.2.1. Ethernet

ETH0

ETH1

3.2.2 4G/5G 网络

4G/5G 拨号

4G/5G 上网测试

3.2.3WIFI

连接WiFi热点

3.2.4 Bluetooth

2.4G

IDO-IPC3528-V1

Linux软件使用手册

深圳触觉智能科技有限公司

www.industio.cn

文档修订历史

版本	PCBA版本	修订内容	修订	审核	日期
V1.0	V1D	创建文档	LJH	IDO	2024/11/26

整机信息

基本参数	
PCB版本号	IDO-SBC3528-V1
SOC	RockChip RK3568
内存	LPDDR4, 2GB / 4GB / 8GB
存储	eMMC, 16GB / 32GB / 64GB / 128GB
OS	Ubuntu/Debian/Buildroot
4G/5G	4G: EC20
	5G: RG200U (NC)
WiFi	默认单频2.4G WiFi
BT	BT4.2
Lan	Lan0 (1000M) : YT8531C
	Lan1 (1000M) : YT8531C
Display	HDMI x1: 支持最大分辨率: 4K@60fps
USB	2路 USB3.0 标准USB-A座
	2路 USB2.0 标准USB-A座
TF卡	TF卡座
耳机座	CTIA
按键	REC按键
LED	1 x 电源指示灯
	1 x 系统指示灯
	2 x 用户自定义指示灯
RTC	内置
DC	12V
整机实物图片	



1. IDO-IPC3528-V1 Debian系统

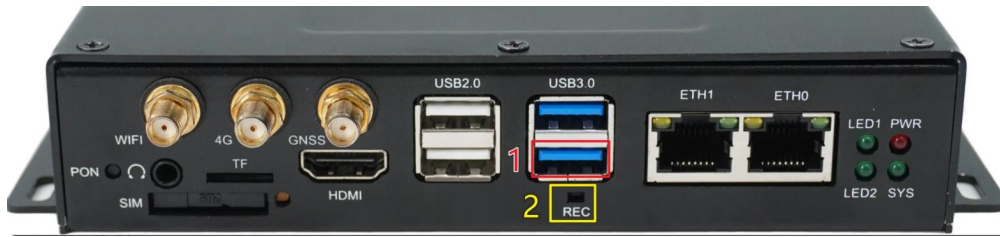
1.1. 外围功能接口

1.1.1. 固件烧录

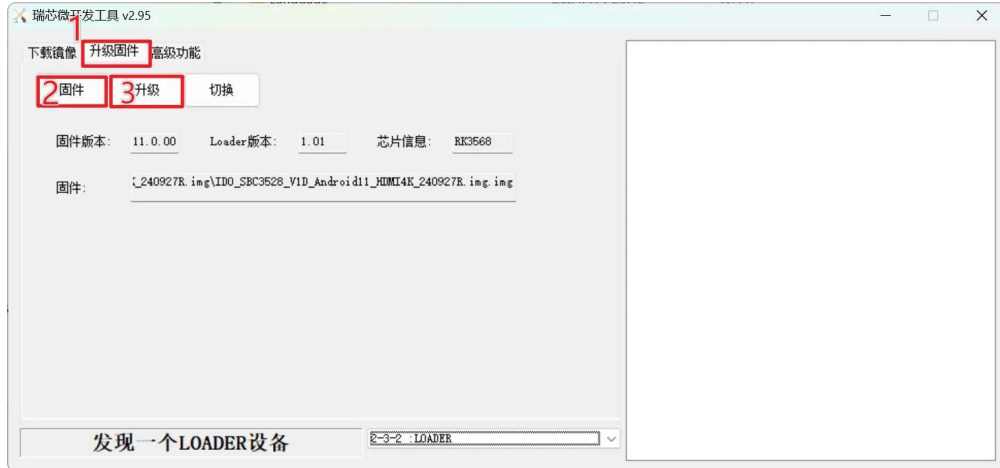
固件地址: https://pan.baidu.com/s/1uZwjJMplS1xBMyd_4_2A9w?pwd=1234

主板下载固件需要将设备切换为 Loader 模式，Loader 模式的操作流程方法如下：

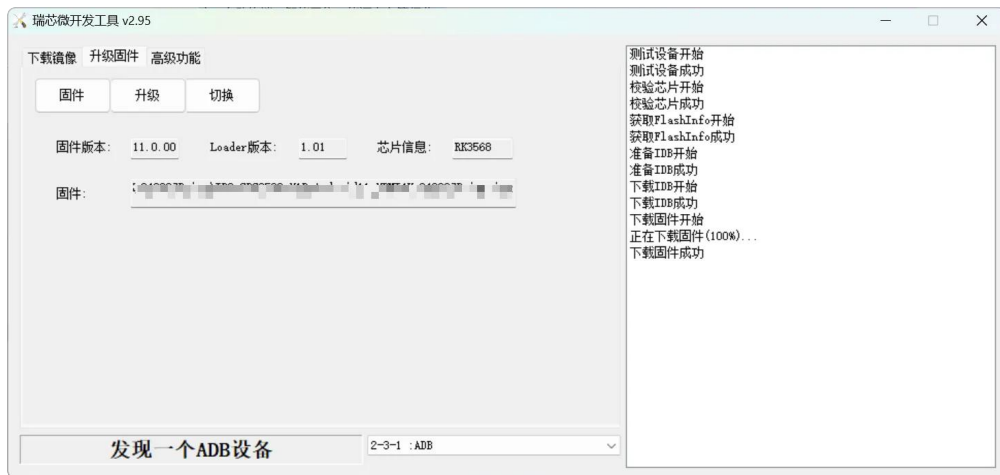
1. 断开整机的所有电源；
2. 使用USB Type A数据线，连接PC端的USB接口（如下图红色框所示）；
3. 按住REC按键并保持（如下图黄色框所示），并保持；
4. 给整机供电；



5. 烧录工具显示“发现一个LOADER设备”后，可松开REC按键；
6. 点击烧录工具【升级固件】界面的【固件】，在系统文件资源管理器中选择烧录固件；



7. 固件加载完成，点击【升级】按键开始烧录程序，烧录完成开发板将会重新启动。



1.1.2. 指示灯

设备上电后，绿色SYS指示灯亮起，正常启动后以1秒2次频率闪动，如下图所示：



- 开机：绿灯闪烁
- 关机：绿灯熄灭
- 异常：绿灯常亮（或灭）

名称	功能	说明
SYS	系统指示灯	上电后系统正常运行时1秒闪烁2次。
PWR	电源指示灯	上电后，常量
LED1	用户自定义LED灯	默认不亮
LED2	用户自定义LED灯	默认不亮

1.1.3. REC按键

【REC】按键，如下图所示：



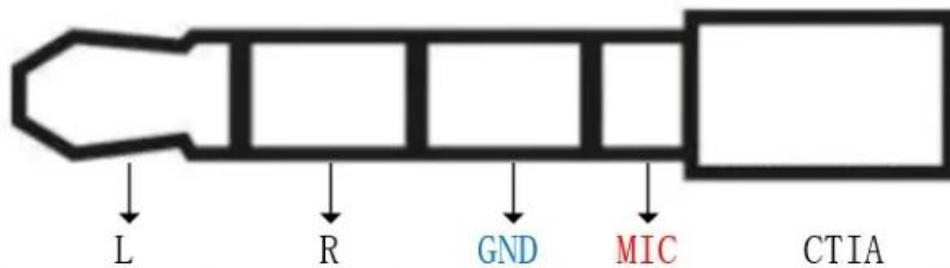
系统正常开机状态下按下【REC】按键，系统启动后为音量加键

1.1.4. 耳机

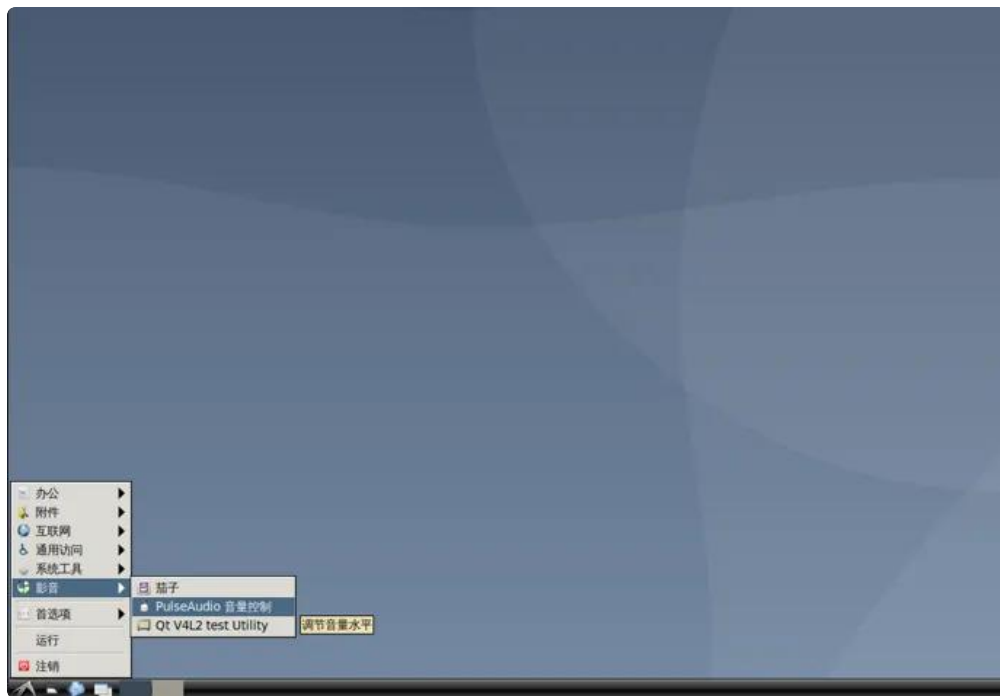
耳机接口，如下图所示：



CTIA标准四段式耳机，定义如下：



桌面左下角打开PulseAudio程序：



弹出的窗口中，在回放页面中设置系统音量：



当需要静音时，点击静音按钮即可：



声音通过3.5mm耳机孔输出到耳机，调节声音大小，音乐清晰无杂音。

测试项目	要求	结果
3.5MM耳机孔	适配3.5耳机，声音清晰无杂音	✓

1.1.5. Mic录音

使用arecord工具可以进行录音测试：

Bash

```
1  linaro@linaro-alip:~$ arecord -D hw:1,0 -r 48000 -c 2 -f S16_LE test.wav
2  Recording WAVE 'test.wav' : Signed 16 bit Little Endian, Rate 48000 Hz, Stereo
3
4  ^CAborted by signal 中断...
5  linaro@linaro-alip:~$
```

录音完后播放测试：

Bash

```
1  linaro@linaro-alip:~$ aplay -D plughw:1,0 ./test.wav
2  Playing WAVE './test.wav' : Signed 16 bit Little Endian, Rate 48000 Hz, Stereo
```

1.1.6. TF卡槽

TF卡槽，如下图所示：



支持FAT32和NTFS格式分区自动挂载

TF卡正面朝上，触点朝下插入盒子TF卡槽

测试项目	要求	结果
TF卡	识别TF卡正常、读写文件正常	✓

1.1.7. HDMI

HDMI接口，如下图所示：



支持最大分辨率：4K@60fps

测试项目	要求	结果
显示	HDMI画面显示正常	✓
	HDMI音频输出正常	✓
	测试分辨率：4K@60fps	✓
	HDMI热插拔正常	✓

1.1.8. USB

USB接口，如下图所示：



测试项目	要求	结果
USB功能测试	USB1支持host 和device 模式的切换	✓
	USB2需要兼容U盘、键盘、鼠标	✓
	USB3需要兼容U盘、键盘、鼠标	✓
	USB4需要兼容U盘、键盘、鼠标	✓

USB电源开关控制：

USB端口	动作	命令
USB1	切换HOST模式	echo host >/sys/devices/platform/fe8a8000.usb2-phy/otg_mode echo 255 > /sys/class/leds/usb3_otg_pwr/brightness
	切换DEVICE模式	echo peripheral>/sys/devices/platform/fe8a0000.usb2-phy/otg_mode echo 0 > /sys/class/leds/usb3_otg_pwr/brightness
USB2	关闭电源	echo 0 > /sys/class/leds/usb3_host_pwr/brightness
	开启电源	echo 1 > /sys/class/leds/usb3_host_pwr/brightness
USB3	关闭电源	echo 0 > /sys/class/leds/usb2_host_pwr/brightness
	开启电源	echo 1 >/sys/class/leds/usb2_host_pwr/brightness
USB4	关闭电源	echo 0 > /sys/class/leds/usb2_fe2_pwr/brightness
	开启电源	echo 1 > /sys/class/leds/usb2_fe2_pwr/brightness

测试项目	要求	结果
USB开关测试	USB1 OTG测试	✓
	USB2开关测试	✓
	USB3开关测试	✓
	USB4开关测试	✓

1.1.9. DIDO

DIDO接口（与CAN接口复用，二选一），如下图所示：



各个引脚定义如下表所示：

引脚	功能
1	DI1
2	DI2
3	DI_SS
4	DO1
5	DO2
6	GND

DI的使用

通过读取设备节点/dev/ido_di，获取DI1和DI2的输入电平状态。打开/dev/ido_di设备文件，读取3个字节，前2个字节为两路DI的输入结果。

通过代码读取举例：

```

1  #include <stdio.h>
2  #include <sys/types.h>
3  #include <sys/stat.h>
4  #include <fcntl.h>
5  #include <unistd.h>
6  #include <stdlib.h>
7
8  int main(void)
9  {
10     char buf[3] = {0, 0, 0};
11
12     int fd;
13     int ret;
14
15     fd = open("/dev/ido_di", O_RDONLY);
16
17     if (fd < 0){
18         exit(1);
19     }
20
21     ret = read(fd, buf, 3);
22     if (ret != 3){
23         perror("read");
24         close(fd);
25         exit(2);
26     }
27
28     printf("DI1输入%s\n", buf[0] == '1' ? "低电平" : "高电平");
29     printf("DI2输入%s\n", buf[1] == '1' ? "低电平" : "高电平");
30
31     close(fd);
32
33     return 0;
34 }

```

通过命令行来获取举例：

```
root@linaro-alip:/home/linaro# cat /dev/ido_di
00000000000000000000000000000000000000000000
```

返回结果前2个字符与DI1、DI2的输入关系见下表:

结果	DI1输入	DI2输入
00	高	高
10	低	高
01	高	低
11	低	低

DO的使用

通过向/dev/ido_do设备文件写入特定字符串，控制DO1和DO2的输出。

通过代码控制输出举例：

```

1  #include <stdio.h>
2  #include <sys/types.h>
3  #include <sys/stat.h>
4  #include <fcntl.h>
5  #include <unistd.h>
6  #include <stdlib.h>
7
8  int main(void)
9  {
10     int fd;
11     int ret;
12
13     fd = open("/dev/ido_do", O_WRONLY);
14
15     if (fd < 0){
16         exit(1);
17     }
18
19     //D01输出低电平
20     printf("D01 输出低电平\n");
21     ret = write(fd, "1D", 3);
22     if (ret != 3)
23         goto write_err;
24     sleep(3);
25     //D01输出高电平
26     printf("D01 输出高电平\n");
27     ret = write(fd, "1E", 3);
28     if (ret != 3)
29         goto write_err;
30     sleep(3);
31
32     //D02输出低电平
33     printf("D02 输出低电平\n");
34     ret = write(fd, "2D", 3);
35     if (ret != 3)
36         goto write_err;
37     sleep(3);
38
39     //D01输出高电平
40     printf("D02 输出高电平\n");
41     ret = write(fd, "2E", 3);
42     if (ret != 3)
43         goto write_err;
44     sleep(3);
45

```

```

46         close(fd);
47
48         return 0;
49
50     write_err:
51         close(fd);
52         return -1;
53 }

```

通过命令行来设置输出举例：

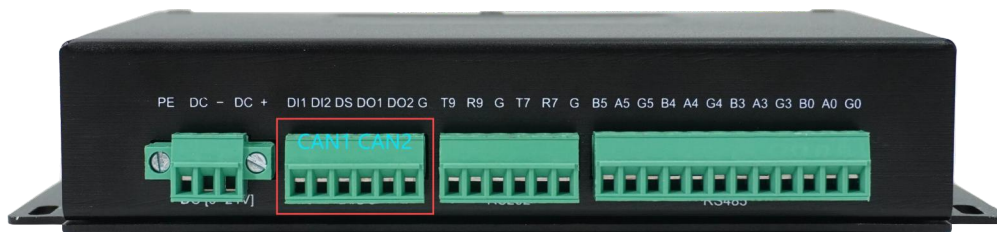
```

1  //D01输出0V
2  echo 1D > /dev/ido_do
3  //D01输出12V
4  echo 1E > /dev/ido_do
5
6  //D02输出0V
7  echo 2D > /dev/ido_do
8  //D02输出12V
9  echo 2E > /dev/ido_do

```

1.1.10. CAN

CAN接口（与DIDO接口复用，二选一），如下图所示：



测试

打开can设备，并设置通信波特率为250Kbps：

▼

Bash

```
1 sudo ip link set can0 down
2 sudo ip link set can0 type can bitrate 250000
3 sudo ip link set can0 up
4
5 sudo ip link set can1 down
6 sudo ip link set can1 type can bitrate 250000
7 sudo ip link set can1 up
```

发送测试:

▼

Bash

```
1 sudo cansend can0 123#DEADBEEF
```

接收测试:

▼

Bash

```
1 sudo candump can0
```

1.1.11. UART

主机共配置6路UART，其中4路RS485，2路RS232，如下:



各个UART对应的设备节点如下表:

编号	设备节点	类型
1	/dev/ttyS0	RS485
2	/dev/ttyS3	RS485
3	/dev/ttyS4	RS485
4	/dev/ttyS5	RS485

5	/dev/ttyS7	RS232
6	/dev/ttyS9	RS232

测试

RS485和RS232都可以使用microcom工具进行简单的收发测试。

需要先安装microcom工具：

```
▼ Bash |
1  linaro@linaro-alip:~$ sudo apt-get update
2  linaro@linaro-alip:~$ sudo apt-get install microcom
```

以测试/dev/ttyS0为例：

```
▼ Bash |
1  linaro@linaro-alip:~$ microcom -s 115200 -p /dev/ttyS0
2  [ 754.636312] of_dma_request_slave_channel: dma-names property of node '/serial@fdd50000' missing or empty
3  [ 754.636443] ttyS0 - failed to request DMA, use interrupt mode
4  connected to /dev/ttyS0
5  Escape character: Ctrl-\
6  Type the escape character to get to the prompt.
```

按下键盘任意键会发送对应的字符，而接收的内容会显示在终端。

按【ctrl】和【\】组合键，然后输入quit退出测试。

```
1  linaro@linaro-alip:~$ ls
2  Desktop
3  linaro@linaro-alip:~$ microcom -s 115200 -p /dev/ttyS0
4  [ 754.636312] of_dma_request_slave_channel: dma-names property of node '/
   serial@fdd50000' missing or empty
5  [ 754.636443] ttyS0 - failed to request DMA, use interrupt mode
6  connected to /dev/ttyS0
7  Escape character: Ctrl-\
8  Type the escape character to get to the prompt.
9
10 Enter command. Try 'help' for a list of builtin commands
11 -> quit
12 exiting
```

1.2. 网络测试

1.2.1. Ethernet

主板有两路千兆以太网接口ETH0和ETH1，如下图所示：



ETH0

查看eth0的ip:

```
1  linaro@linaro-alip:~$ sudo ifconfig eth0
2  eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
3      inet 192.168.1.177  netmask 255.255.255.0  broadcast 192.168.1.255
4      inet6 fe80::4f32:fc2d:eb30:220b  prefixlen 64  scopeid 0x20<link>
5      ether d2:e1:8d:d5:f7:49  txqueuelen 1000  (Ethernet)
6      RX packets 78  bytes 11727 (11.4 KiB)
7      RX errors 0  dropped 0  overruns 0  frame 0
8      TX packets 60  bytes 5822 (5.6 KiB)
9      TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0
10     device interrupt 52
11
```

设置eth0的ip:

```
1  linaro@linaro-alip:~$ sudo ifconfig eth1 192.168.1.123
2  linaro@linaro-alip:~$
3
```

eth0 ping测试:

```
1  linaro@linaro-alip:~$ sudo ping www.baidu.com -I eth0
2  PING www.a.shifen.com (14.215.177.39) from 192.168.1.177 eth1: 56(84) byte
   s of data.
3  64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=1 ttl=54 time=6.75 ms
4  64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=2 ttl=54 time=6.32 ms
5  64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=3 ttl=54 time=7.01 ms
6  64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=4 ttl=54 time=7.47 ms
7  64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=5 ttl=54 time=7.01 ms
```

ETH1

查看eth1的ip:

```

1  linaro@linaro-alip:~$ sudo ifconfig eth1
2  eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
3          inet 192.168.1.177  netmask 255.255.255.0  broadcast 192.168.1.255
4          inet6 fe80::4f32:fc2d:eb30:220b  prefixlen 64  scopeid 0x20<link>
5          ether d2:e1:8d:d5:f7:49  txqueuelen 1000  (Ethernet)
6          RX packets 78  bytes 11727 (11.4 KiB)
7          RX errors 0  dropped 0  overruns 0  frame 0
8          TX packets 60  bytes 5822 (5.6 KiB)
9          TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0
10         device interrupt 52
11

```

设置eth1的ip:

```

1  # 修改eth1 ip
2  linaro@linaro-alip:~$ sudo ifconfig eth1 192.168.1.123
3  # 添加默认网关
4  linaro@linaro-alip:~$ route add default dev eth1
5  linaro@linaro-alip:~$ route add default gw 192.168.1.1
6

```

eth1 ping测试:

```

1  linaro@linaro-alip:~$ sudo ping www.baidu.com -I eth1
2  PING www.a.shifen.com (14.215.177.39) from 192.168.1.177 eth1: 56(84) bytes
   of data.
3  64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=1 ttl=54 time=6.75 ms
4  64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=2 ttl=54 time=6.32 ms
5  64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=3 ttl=54 time=7.01 ms
6  64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=4 ttl=54 time=7.47 ms
7  64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=5 ttl=54 time=7.01 ms

```

1.2.2. 4G/5G 网络4G/5G 拨号

上电前，正确安装模块和SIM卡，上电后，系统会自动进行拨号上网。

拨号成功，则会产生wwan0/usb0网络节点：

```
1  linaro@linaro-alip:~$ sudo ifconfig wwan0
2  wwan0: flags=193<UP,RUNNING,NOARP> mtu 1500
3      inet 10.101.61.51 netmask 255.255.255.248
4      inet6 fe80::fc:f6ff:fe8d:bab6 prefixlen 64 scopeid 0x20<link>
5      ether 02:fc:f6:8d:ba:b6 txqueuelen 1000 (Ethernet)
6      RX packets 42 bytes 7013 (6.8 KiB)
7      RX errors 0 dropped 0 overruns 0 frame 0
8      TX packets 57 bytes 4608 (4.5 KiB)
9      TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
10
11  linaro@linaro-alip:~$
```

4G/5G 上网测试

使用ping命令测试4G/5G上网功能是否正常：

```
1  linaro@linaro-alip:~$ sudo ping www.baidu.com -I wwan0
2  PING www.a.shifen.com (183.232.231.174) from 10.101.61.51 wwan0: 56(84) bytes of data.
3  64 bytes from 183.232.231.174 (183.232.231.174): icmp_seq=1 ttl=55 time=33.3 ms
4  64 bytes from 183.232.231.174 (183.232.231.174): icmp_seq=2 ttl=55 time=48.1 ms
5  64 bytes from 183.232.231.174 (183.232.231.174): icmp_seq=3 ttl=55 time=46.2 ms
6  64 bytes from 183.232.231.174 (183.232.231.174): icmp_seq=4 ttl=55 time=45.2 ms
7  64 bytes from 183.232.231.174 (183.232.231.174): icmp_seq=5 ttl=55 time=42.1 ms
```

1.2.3 WIFI

主机配置了一路2.4G WiFi模块，型号为AW-NM372SM。

系统上电默认会打开WiFi，对应的网络设备节点为wlan0。

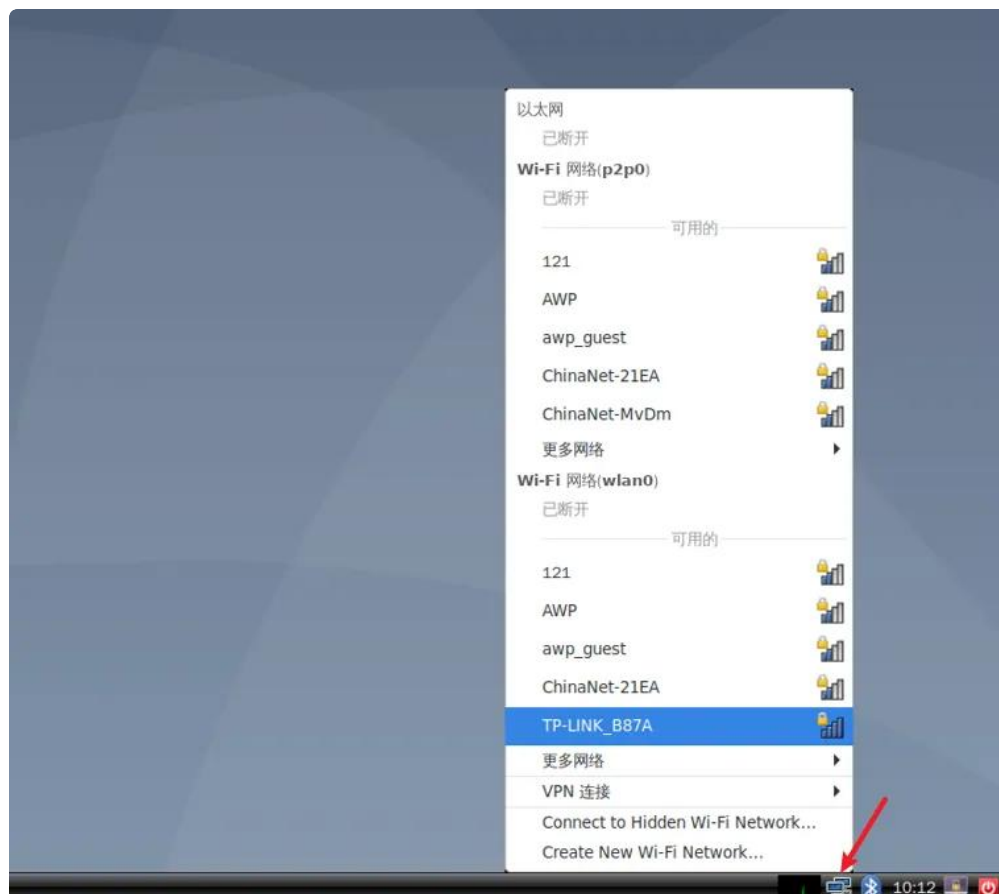
```
1  linaro@linaro-alip:~$ sudo ifconfig wlan0
2  wlan0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
3      ether 2c:d2:6b:10:ea:4d txqueuelen 1000 (Ethernet)
4      RX packets 0 bytes 0 (0.0 B)
5      RX errors 0 dropped 0 overruns 0 frame 0
6      TX packets 0 bytes 0 (0.0 B)
7      TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

连接WiFi热点

连接热点有2种方式，可以在桌面上操作，也可以使用命令行操作。

方式一

点击桌面右下角的网络图标，即可看到WiFi热点列表：

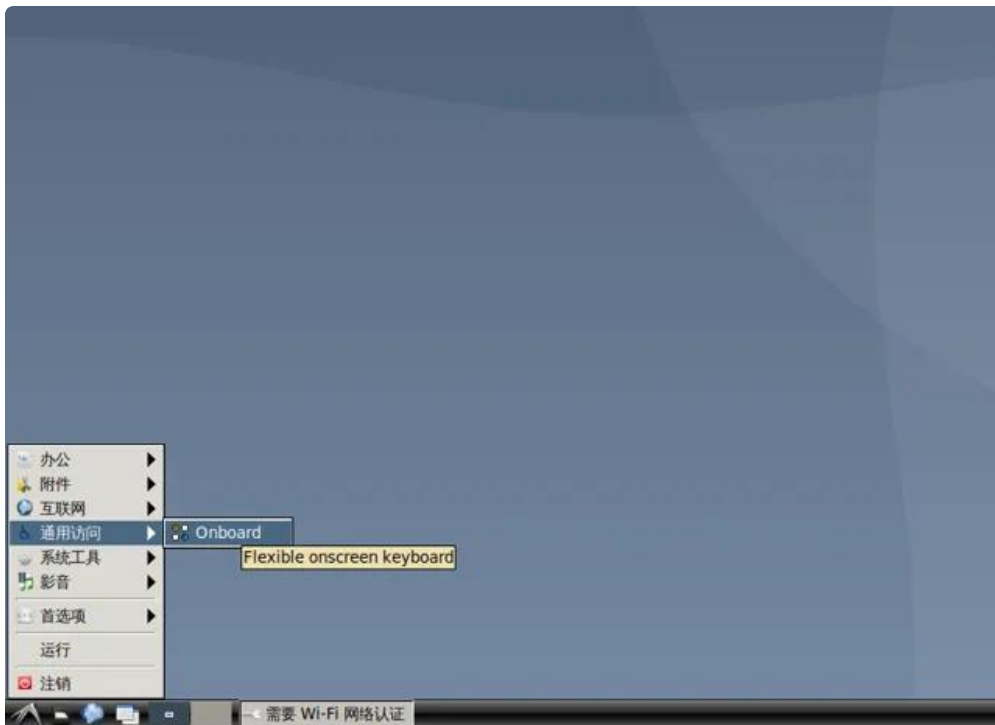


点击要连接的热点，弹出密码输入窗口：



如果有连接键盘，直接输入密码即可；如果没有连接键盘，可以使用系统自带的软键盘。

点击桌面左下角的应用图标，然后依次点击【通用访问】->【Onboard】：



使用软键盘输入密码后，点击【Connect】连接热点：



连接成功后，桌面右下角的网络图标将改变：



方式二

修改/etc/network/interfaces：

```
Bash |
1 # interfaces(5) file used by ifup(8) and ifdown(8)
2 # Include files from /etc/network/interfaces.d:
3 source-directory /etc/network/interfaces.d
4 allow-hotplug wlan0
5 auto wlan0
6 iface wlan0 inet dhcp
7 wpa-conf /etc/wpa.conf
```

新建/etc/wpa.conf：

```
1  linaro@linaro-alip:~$ sudo touch /etc/wpa.conf
2  linaro@linaro-alip:~$ sudo chmod a+w /etc/wpa.conf
```

然后使用wpa_passphrase连接WiFi热点：

```
1  linaro@linaro-alip:~$ sudo killall wpa_supplicant
2  linaro@linaro-alip:~$ sudo wpa_passphrase TP-LINK_B87A 12345678 >> /etc/wpa.conf
3  linaro@linaro-alip:~$ sudo ifup wlan0
```

连接成功后使用ifconfig命令可查看wlan0的IP：

```
1  linaro@linaro-alip:~$ sudo ifconfig wlan0
2  wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
3          inet 192.168.1.165  netmask 255.255.255.0  broadcast 192.168.1.255
4          inet6 fe80::868:5528:86fe:c9da  prefixlen 64  scopeid 0x20<link>
5          ether 2c:d2:6b:10:ea:4d  txqueuelen 1000  (Ethernet)
6          RX packets 28  bytes 55800 (54.4 KiB)
7          RX errors 0  dropped 0  overruns 0  frame 0
8          TX packets 40  bytes 64335 (62.8 KiB)
9          TX errors 0  dropped 11 overruns 0  carrier 0  collisions 0
```

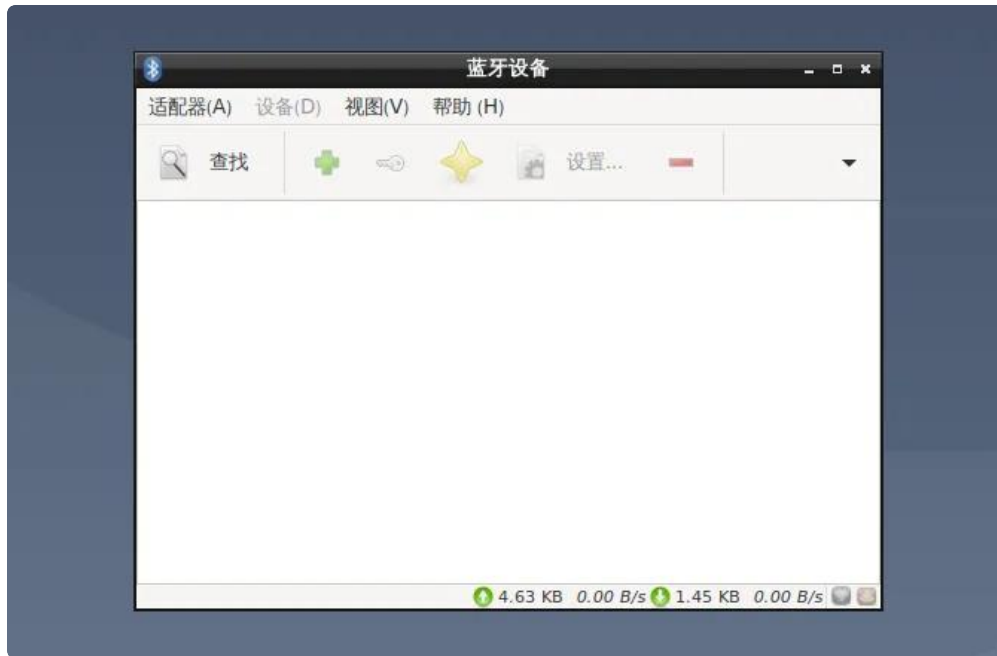
WiFi上网测试

使用ping命令来进行上网测试：

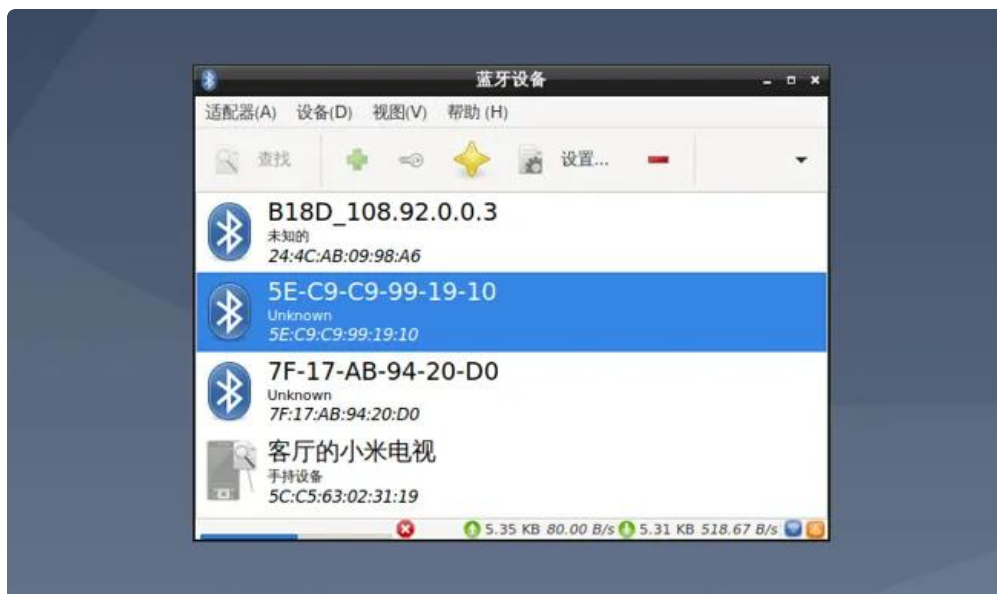
```
1  linaro@linaro-alip:~$ sudo ping www.baidu.com -I wlan0
```

1.2.4 Bluetooth

连接蓝牙设备



将看到附近的蓝牙设备列表：



选中要连接设备，右键->配对，即可连接该设备：



2. IDO-IPC3528-V1 Ubuntu系统

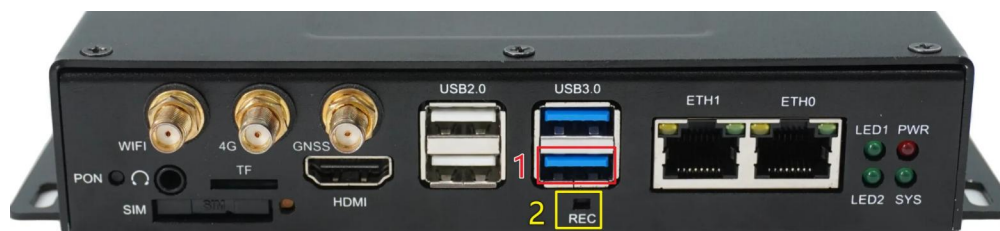
2.1. 外围功能接口

2.1.1. 固件烧录

固件地址: https://pan.baidu.com/s/1uZwjJMplS1xBMyd_4_2A9w?pwd=1234

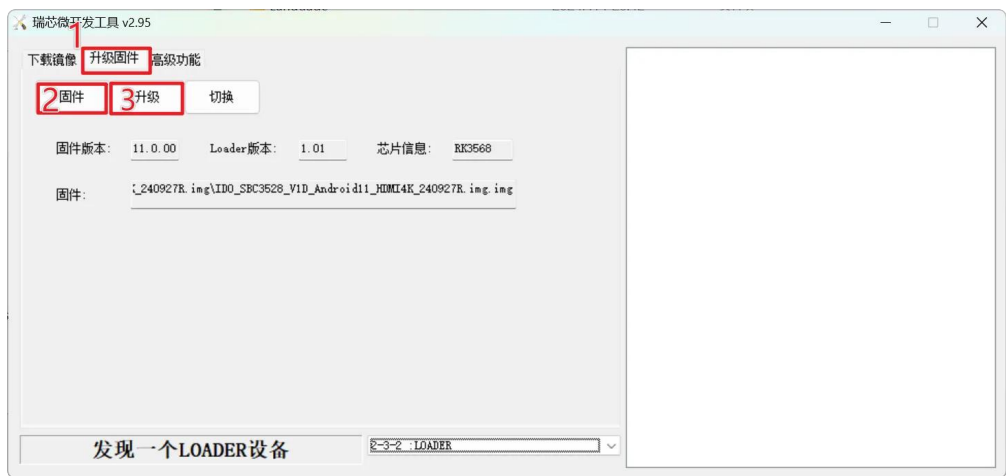
主板下载固件需要将设备切换为 Loader 模式，Loader 模式的操作流程方法如下：

1. 断开整机的所有电源；
2. 使用USB Type A数据线，连接PC端的USB接口（如下图红色框所示）；
3. 按住REC按键并保持（如下图黄色框所示），并保持；
4. 给整机供电；

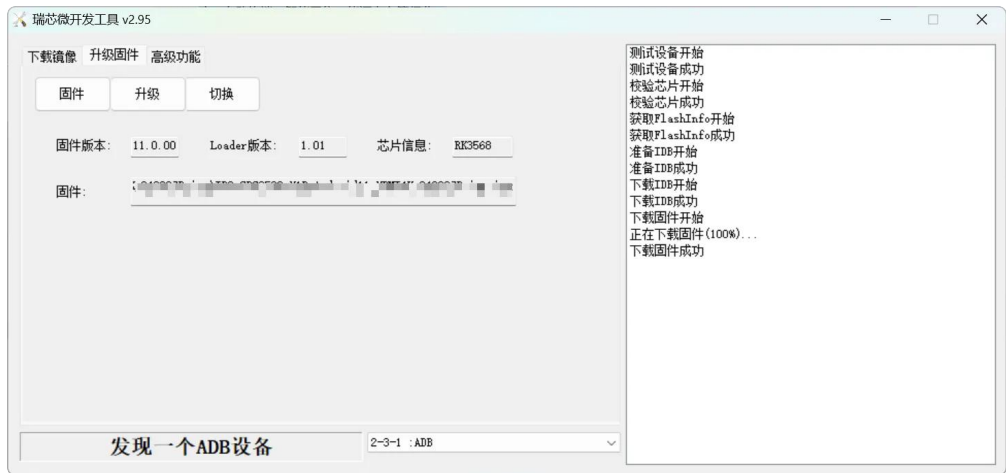


5. 烧录工具显示“发现一个LOADER设备”后，可松开REC按键；

6. 点击烧录工具【升级固件】界面的【固件】，在系统文件资源管理器中选择烧录固件；



7. 固件加载完成，点击【升级】按键开始烧录程序，烧录完成开发板将会重新启动。



2.1.2. 指示灯

设备上电后，绿色SYS指示灯亮起，正常启动后以1秒2次频率闪动，如下图所示：



- 开机：绿灯闪烁
- 关机：绿灯熄灭
- 异常：绿灯常亮（或灭）

名称	功能	说明
SYS	系统指示灯	上电后系统正常运行时1秒闪烁2次。
PWR	电源指示灯	上电后，常量
LED1	用户自定义LED灯	默认不亮
LED2	用户自定义LED灯	默认不亮

2.1.3. REC按键

【REC】按键，如下图所示：



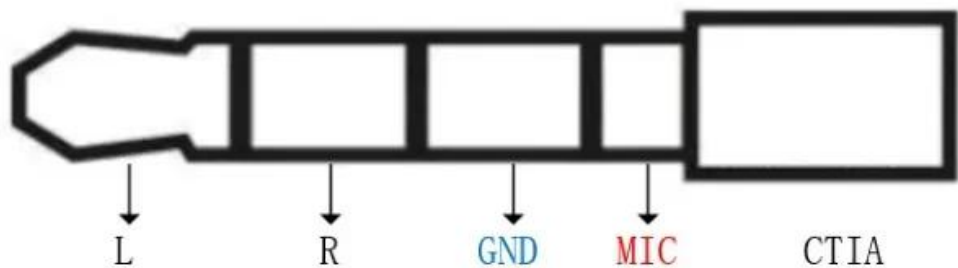
系统正常开机状态下按下【REC】按键，系统启动后为音量加键

2.1.4. 耳机

耳机接口，如下图所示：



CTIA标准四段式耳机，定义如下：



桌面左下角打开PulseAudio程序：

查看声卡设备

```

1  root@ido:/# aplay -l
2  **** List of PLAYBACK Hardware Devices ****
3  card 0: rockchiphdmi [rockchip,hdmi], device 0: rockchip,hdmi i2s-hifi-0 [r
   rockchip,hdmi i2s-hifi-0]
4    Subdevices: 1/1
5    Subdevice #0: subdevice #0
6  card 1: rockchiprk809co [rockchip,rk809-codec], device 0: fe410000.i2s-rk81
   7-hifi rk817-hifi-0 [fe410000.i2s-rk817-hifi rk817-hifi-0]
7    Subdevices: 1/1
8    Subdevice #0: subdevice #0

```

播放音频

插入耳机，执行以下命令。

```

1  aplay -D plughw:1,0 /usr/share/sounds/alsa/Rear_Center.wav

```

声音通过3.5mm耳机孔输出到耳机，调节声音大小，音乐清晰无杂音。

测试项目	要求	结果
3.5MM耳机孔	适配3.5耳机，声音清晰无杂音	✓

2.1.5. Mic录音

使用arecord工具可以进行录音测试：

Bash

```
1 root@ido:~# arecord -D hw:1,0 -r 48000 -c 2 -f S16_LE test.wav
2 Recording WAVE 'test.wav' : Signed 16 bit Little Endian, Rate 48000 Hz, Stereo
3
4 ^CAborted by signal 中断...
```

录音完后播放测试：

Bash

```
1 root@ido:~# aplay -D plughw:1,0 ./test.wav
2 Playing WAVE './test.wav' : Signed 16 bit Little Endian, Rate 48000 Hz, Stereo
```

2.1.6. TF卡槽

TF卡槽，如下图所示：



支持FAT32和NTFS格式分区自动挂载

TF卡正面朝上，触点朝下插入盒子TF卡槽

测试项目	要求	结果
TF卡	识别TF卡正常、读写文件正常	✓

2.1.7. HDMI

HDMI接口，如下图所示：



支持最大分辨率：4K@60fps

测试项目	要求	结果
显示	HDMI画面显示正常	✓
	HDMI音频输出正常	✓
	测试分辨率：4K@60fps	✓
	HDMI热插拔正常	✓

2.1.8. USB

USB接口，如下图所示：



测试项目	要求	结果
USB功能测试	USB1支持host 和device 模式的切换	✓
	USB2需要兼容U盘、键盘、鼠标	✓
	USB3需要兼容U盘、键盘、鼠标	✓
	USB4需要兼容U盘、键盘、鼠标	✓

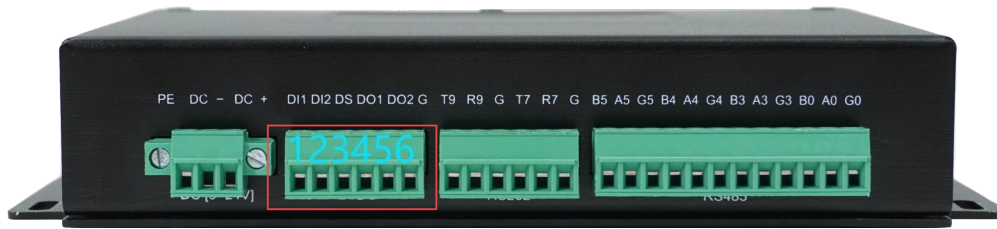
USB电源开关控制：

USB端口	动作	命令
USB1	切换HOST模式	echo host >/sys/devices/platform/fe8a8000.usb2-phy/otg_mode echo 255 > /sys/class/leds/usb3_otg_pwr/brightness
	切换DEVICE模式	echo peripheral>/sys/devices/platform/fe8a0000.usb2-phy/otg_mode echo 0 > /sys/class/leds/usb3_otg_pwr/brightness
USB2	关闭电源	echo 0 > /sys/class/leds/usb3_host_pwr/brightness
	开启电源	echo 1 > /sys/class/leds/usb3_host_pwr/brightness
USB3	关闭电源	echo 0 > /sys/class/leds/usb2_host_pwr/brightness
	开启电源	echo 1 >/sys/class/leds/usb2_host_pwr/brightness
USB4	关闭电源	echo 0 > /sys/class/leds/usb2_fe2_pwr/brightness
	开启电源	echo 1 > /sys/class/leds/usb2_fe2_pwr/brightness

测试项目	要求	结果
USB开关测试	USB1 OTG测试	✓
	USB2开关测试	✓
	USB3开关测试	✓
	USB4开关测试	✓

2.1.9. DIDO

DIDO接口（与CAN接口复用，二选一），如下图所示：



各个引脚定义如下表所示：

引脚	功能
1	DI1
2	DI2
3	DI_SS
4	DO1
5	DO2
6	GND

DI的使用

通过读取设备节点/dev/ido_di，获取DI1和DI2的输入电平状态。打开/dev/ido_di设备文件，读取3个字节，前2个字节为两路DI的输入结果。

通过代码读取举例：

```

1  #include <stdio.h>
2  #include <sys/types.h>
3  #include <sys/stat.h>
4  #include <fcntl.h>
5  #include <unistd.h>
6  #include <stdlib.h>
7
8  int main(void)
9  {
10     char buf[3] = {0, 0, 0};
11
12     int fd;
13     int ret;
14
15     fd = open("/dev/ido_di", O_RDONLY);
16
17     if (fd < 0){
18         exit(1);
19     }
20
21     ret = read(fd, buf, 3);
22     if (ret != 3){
23         perror("read");
24         close(fd);
25         exit(2);
26     }
27
28     printf("DI1输入%s\n", buf[0] == '1' ? "低电平" : "高电平");
29     printf("DI2输入%s\n", buf[1] == '1' ? "低电平" : "高电平");
30
31     close(fd);
32
33     return 0;
34 }

```

通过命令行来获取举例：

[illegible]

返回结果前2个字符与DI1、DI2的输入关系见下表:

结果	DI1输入	DI2输入
00	高	高
10	低	高
01	高	低
11	低	低

DO的使用

通过向/dev/ido_do设备文件写入特定字符串，控制DO1和DO2的输出。

通过代码控制输出举例：

```

1  #include <stdio.h>
2  #include <sys/types.h>
3  #include <sys/stat.h>
4  #include <fcntl.h>
5  #include <unistd.h>
6  #include <stdlib.h>
7
8  int main(void)
9  {
10     int fd;
11     int ret;
12
13     fd = open("/dev/ido_do", O_WRONLY);
14
15     if (fd < 0){
16         exit(1);
17     }
18
19     //D01输出低电平
20     printf("D01 输出低电平\n");
21     ret = write(fd, "1D", 3);
22     if (ret != 3)
23         goto write_err;
24     sleep(3);
25     //D01输出高电平
26     printf("D01 输出高电平\n");
27     ret = write(fd, "1E", 3);
28     if (ret != 3)
29         goto write_err;
30     sleep(3);
31
32     //D02输出低电平
33     printf("D02 输出低电平\n");
34     ret = write(fd, "2D", 3);
35     if (ret != 3)
36         goto write_err;
37     sleep(3);
38
39     //D01输出高电平
40     printf("D02 输出高电平\n");
41     ret = write(fd, "2E", 3);
42     if (ret != 3)
43         goto write_err;
44     sleep(3);
45

```

```

46         close(fd);
47
48         return 0;
49
50     write_err:
51         close(fd);
52         return -1;
53 }

```

通过命令行来设置输出举例：

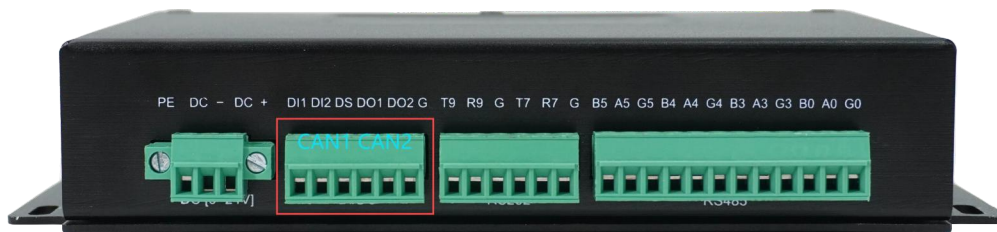
```

1  //D01输出0V
2  echo 1D > /dev/ido_do
3  //D01输出12V
4  echo 1E > /dev/ido_do
5
6  //D02输出0V
7  echo 2D > /dev/ido_do
8  //D02输出12V
9  echo 2E > /dev/ido_do

```

2.1.10. CAN

CAN接口（与DIDO接口复用，二选一），如下图所示：



测试

打开can设备，并设置通信波特率为250Kbps：

▼

Bash

```
1 sudo ip link set can0 down
2 sudo ip link set can0 type can bitrate 250000
3 sudo ip link set can0 up
4
5 sudo ip link set can1 down
6 sudo ip link set can1 type can bitrate 250000
7 sudo ip link set can1 up
```

发送测试:

▼

Bash

```
1 sudo cansend can0 123#DEADBEEF
```

接收测试:

▼

Bash

```
1 sudo candump can0
```

2.1.11. UART

主机共配置6路UART，其中4路RS485，2路RS232，如下:



各个UART对应的设备节点如下表:

编号	设备节点	类型
1	/dev/ttyS0	RS485
2	/dev/ttyS3	RS485
3	/dev/ttyS4	RS485
4	/dev/ttyS5	RS485

5	/dev/ttyS7	RS232
6	/dev/ttyS9	RS232

测试

RS485和RS232都可以使用microcom工具进行简单的收发测试。

需要先安装microcom工具：

```
▼ Bash |
1 root@ido:/# apt-get update
2 root@ido:/# apt-get install microcom
```

以测试/dev/ttyS0为例：

```
▼ Bash |
1 root@ido:/# microcom -s 115200 -p /dev/ttyS0
2 [ 754.636312] of_dma_request_slave_channel: dma-names property of node '/serial@fdd50000' missing or empty
3 [ 754.636443] ttyS0 - failed to request DMA, use interrupt mode
4 connected to /dev/ttyS0
5 Escape character: Ctrl-\
6 Type the escape character to get to the prompt.
```

按下键盘任意键会发送对应的字符，而接收的内容会显示在终端。

按【ctrl】和【\】组合键，然后输入quit退出测试。

```
1 root@ido:/# ls
2 Desktop
3 root@ido:/# microcom -s 115200 -p /dev/ttyS0
4 [ 754.636312] of_dma_request_slave_channel: dma-names property of node '/
serial@fdd50000' missing or empty
5 [ 754.636443] ttyS0 - failed to request DMA, use interrupt mode
6 connected to /dev/ttyS0
7 Escape character: Ctrl-\
8 Type the escape character to get to the prompt.
9
10 Enter command. Try 'help' for a list of builtin commands
11 -> quit
12 exiting
```

2.2. 网络测试

2.2.1. Ethernet

主板有两路千兆以太网接口ETH0和ETH1，如下图所示：



ETH0

查看eth0的ip:

```
1 root@ido:~$ sudo ifconfig eth0
2 eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
3     inet 192.168.1.177 netmask 255.255.255.0 broadcast 192.168.1.255
4     inet6 fe80::4f32:fc2d:eb30:220b prefixlen 64 scopeid 0x20<link>
5     ether d2:e1:8d:d5:f7:49 txqueuelen 1000 (Ethernet)
6     RX packets 78 bytes 11727 (11.4 KiB)
7     RX errors 0 dropped 0 overruns 0 frame 0
8     TX packets 60 bytes 5822 (5.6 KiB)
9     TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
10    device interrupt 52
11
```

设置eth0的ip:

```
1 root@ido:~$ sudo ifconfig eth1 192.168.1.123
2 root@ido:~$
3
```

eth0 ping测试:

```
1 root@ido:~$ sudo ping www.baidu.com -I eth0
2 PING www.a.shifen.com (14.215.177.39) from 192.168.1.177 eth1: 56(84) bytes
  of data.
3 64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=1 ttl=54 time=6.75 ms
4 64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=2 ttl=54 time=6.32 ms
5 64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=3 ttl=54 time=7.01 ms
6 64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=4 ttl=54 time=7.47 ms
7 64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=5 ttl=54 time=7.01 ms
```

ETH1

查看eth1的ip:

```

1 root@ido:~$ sudo ifconfig eth1
2 eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
3     inet 192.168.1.177 netmask 255.255.255.0 broadcast 192.168.1.255
4     inet6 fe80::4f32:fc2d:eb30:220b prefixlen 64 scopeid 0x20<link>
5     ether d2:e1:8d:d5:f7:49 txqueuelen 1000 (Ethernet)
6     RX packets 78 bytes 11727 (11.4 KiB)
7     RX errors 0 dropped 0 overruns 0 frame 0
8     TX packets 60 bytes 5822 (5.6 KiB)
9     TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
10    device interrupt 52
11

```

设置eth1的ip:

```

1 root@ido:~$ sudo ifconfig eth1 192.168.1.123
2 root@ido:~$

```

eth1 ping测试:

```

1 root@ido:~$ sudo ping www.baidu.com -I eth1
2 PING www.a.shifen.com (14.215.177.39) from 192.168.1.177 eth1: 56(84) bytes
  of data.
3 64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=1 ttl=54 time=6.75 ms
4 64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=2 ttl=54 time=6.32 ms
5 64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=3 ttl=54 time=7.01 ms
6 64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=4 ttl=54 time=7.47 ms
7 64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=5 ttl=54 time=7.01 ms

```

2.2.2 4G/5G 网络

4G/5G 拨号

正确按照4G模块、SIM卡和4G天线后，使用quectel-CM工具拨号，命令如下：

```

1 root@ido:~# /usr/bin/quectel-CM &

```

拨号成功，则会产生wwan0/usb0网络节点：

```
1 root@ido:~$ sudo ifconfig wwan0
2 wwan0: flags=193<UP,RUNNING,NOARP> mtu 1500
3     inet 10.101.61.51 netmask 255.255.255.248
4     inet6 fe80::fc:f6ff:fe8d:bab6 prefixlen 64 scopeid 0x20<link>
5     ether 02:fc:f6:8d:ba:b6 txqueuelen 1000 (Ethernet)
6     RX packets 42 bytes 7013 (6.8 KiB)
7     RX errors 0 dropped 0 overruns 0 frame 0
8     TX packets 57 bytes 4608 (4.5 KiB)
9     TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
10
11 root@ido:~$
```

4G/5G 上网测试

使用ping命令测试4G/5G上网功能是否正常：

```
1 root@ido:~$ sudo ping www.baidu.com -I wwan0
2 PING www.a.shifen.com (183.232.231.174) from 10.101.61.51 wwan0: 56(84) bytes of data.
3 64 bytes from 183.232.231.174 (183.232.231.174): icmp_seq=1 ttl=55 time=33.3 ms
4 64 bytes from 183.232.231.174 (183.232.231.174): icmp_seq=2 ttl=55 time=48.1 ms
5 64 bytes from 183.232.231.174 (183.232.231.174): icmp_seq=3 ttl=55 time=46.2 ms
6 64 bytes from 183.232.231.174 (183.232.231.174): icmp_seq=4 ttl=55 time=45.2 ms
7 64 bytes from 183.232.231.174 (183.232.231.174): icmp_seq=5 ttl=55 time=42.1 ms
```

2.2.3. WIFI

主机配置了一路2.4G WiFi模块，型号为AW-NM372SM。

系统上电默认会打开WiFi，对应的网络设备节点为wlan0。

```
1 root@ido:~$ sudo ifconfig wlan0
2 wlan0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
3         ether 2c:d2:6b:10:ea:4d txqueuelen 1000 (Ethernet)
4         RX packets 0 bytes 0 (0.0 B)
5         RX errors 0 dropped 0 overruns 0 frame 0
6         TX packets 0 bytes 0 (0.0 B)
7         TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

连接WiFi热点

连接热点有2种方式，可以在桌面上操作，也可以使用命令行操作。

方式一

使用wpa_cli工具连接，命令如下：

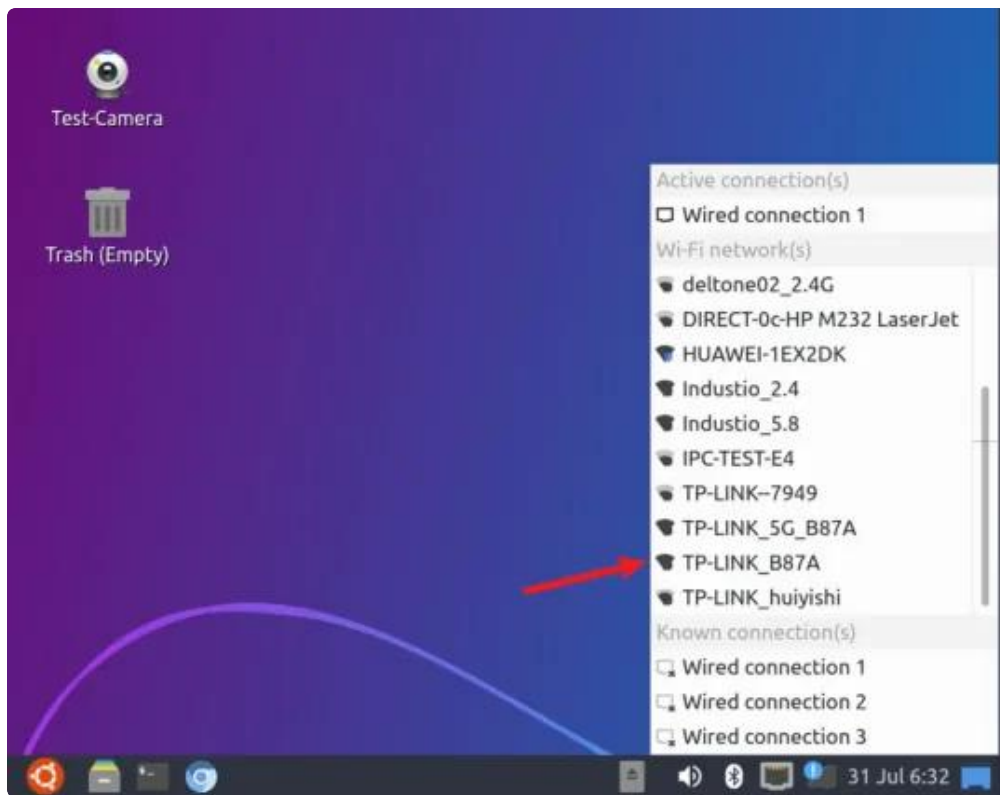
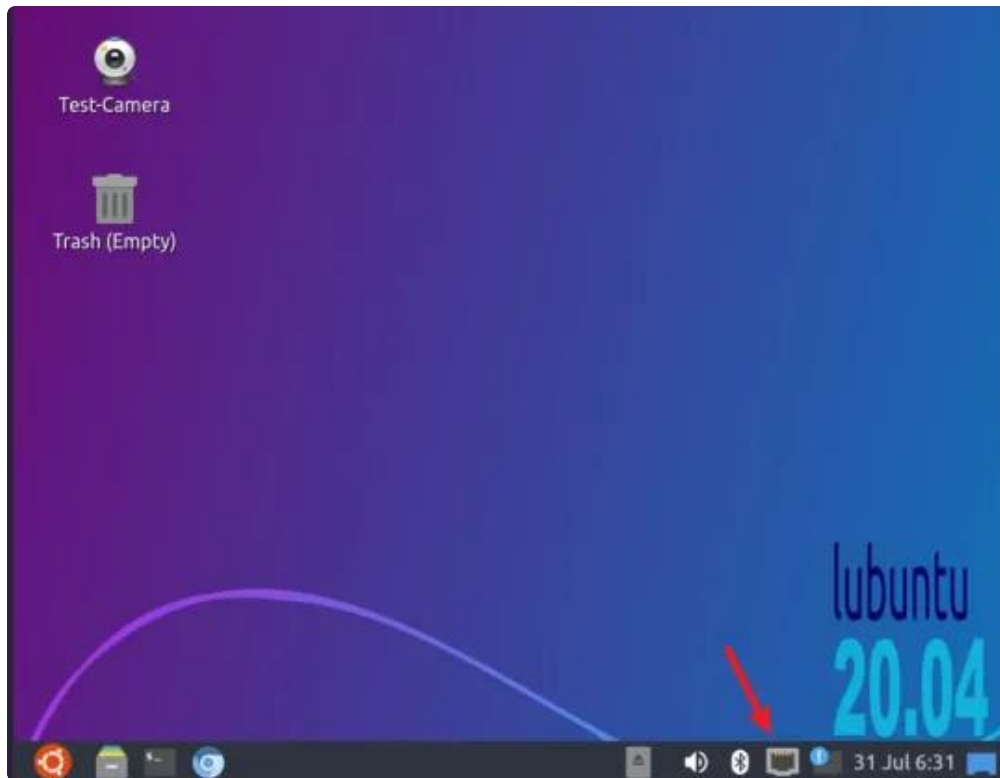
```
1 root@ido:~# wpa_cli -i wlan0 add_network
2 root@ido:~# wpa_cli -i wlan0 set_network 0 ssid '"TP-LINK_B87A"'
3 root@ido:~# wpa_cli -i wlan0 set_network 0 psk '"12345678"'
4 root@ido:~# wpa_cli -i wlan0 enable_network 0
5 root@ido:~# dhclient -i wlan0
```

其中"TP-LINK_B87A"为热点名称，"12345678"为热点密码。

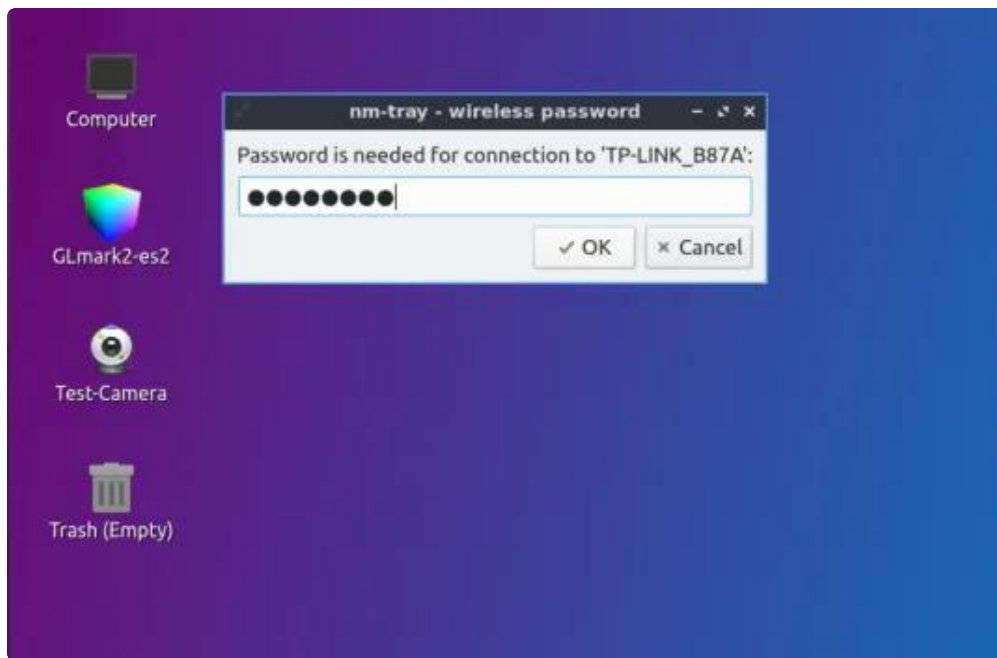
方式二

桌面连接

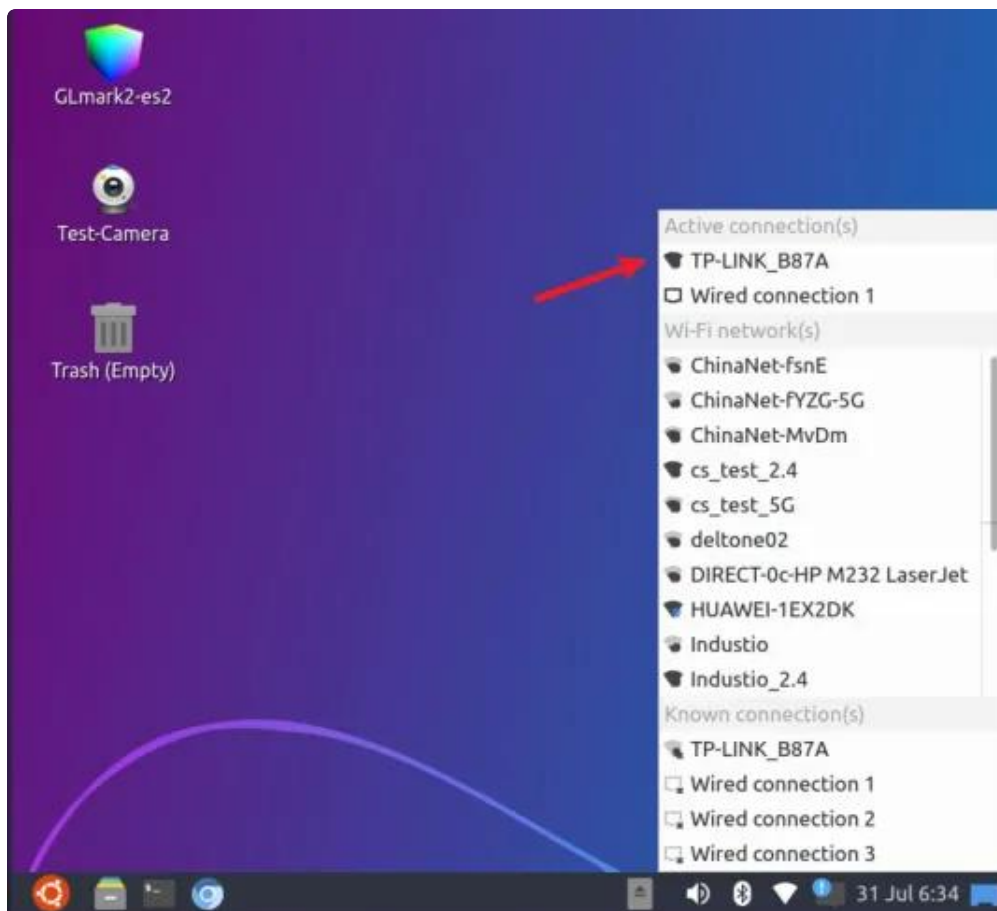
点击桌面右上角的网络按钮，弹出的列表中选择要连接的热点，如下图所示：



弹出密码输入框，使用键盘输入密码（如果没有接键盘，可以使用软键盘Onboard），如下图所示：



输入密码完成后，点击【OK】按钮，即可连接WiFi热点。



```
1 root@ido:~# ifconfig wlan0
2 wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
3     inet 192.168.1.127 netmask 255.255.255.0 broadcast 192.168.1.255
4     inet6 fe80::5ac:29b:8a95:2bf0 prefixlen 64 scopeid 0x20<link>
5     ether 50:41:1c:b0:13:e2 txqueuelen 1000 (Ethernet)
6     RX packets 1399 bytes 824772 (824.7 KB)
7     RX errors 0 dropped 0 overruns 0 frame 0
8     TX packets 977 bytes 128063 (128.0 KB)
9     TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

WiFi上网测试

使用ping命令来进行上网测试：

```
1 root@ido:~$ sudo ping www.baidu.com -I wlan0
```

2.2.4 Bluetooth

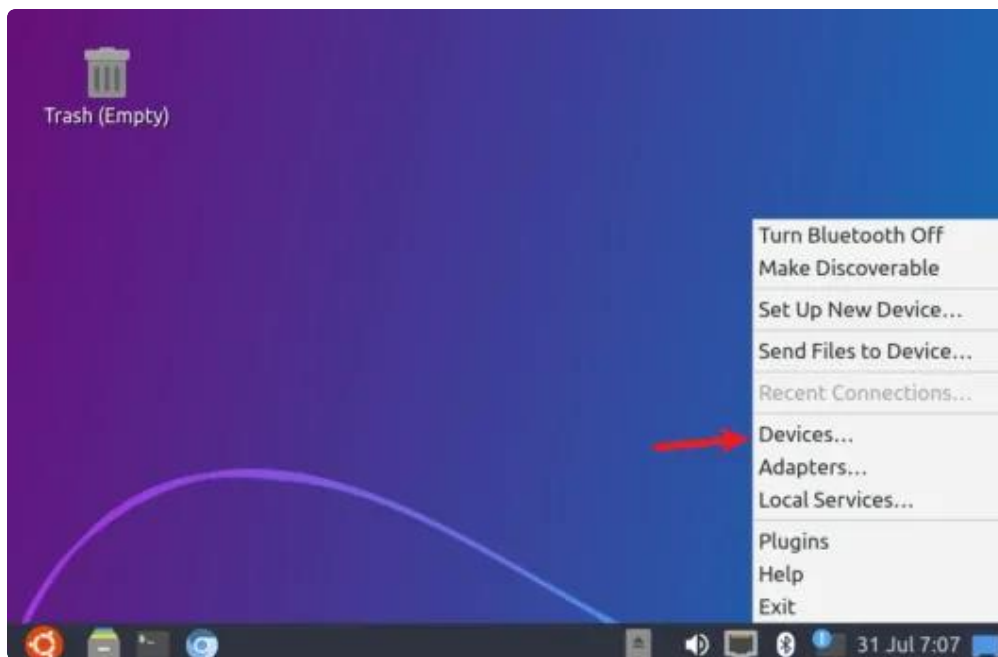
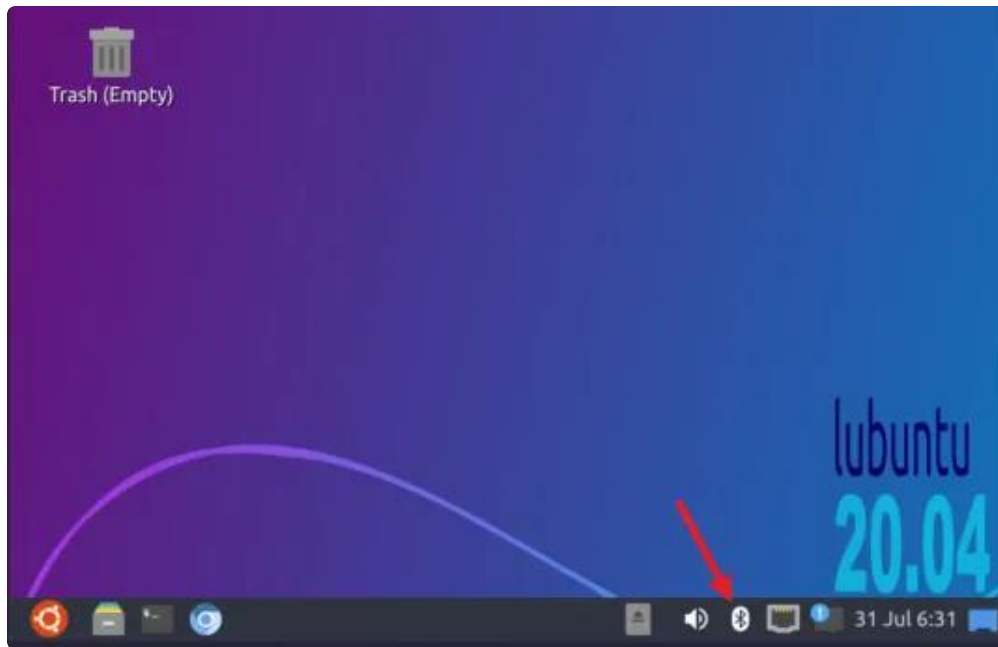
连接蓝牙设备

使用命令行连接蓝牙设备，命令如下：

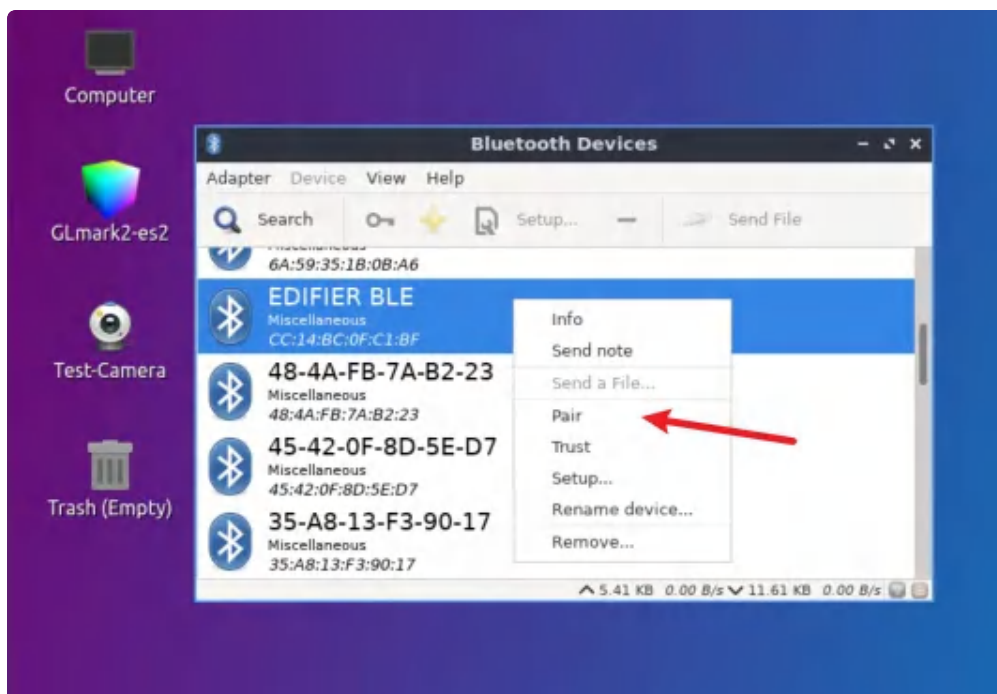
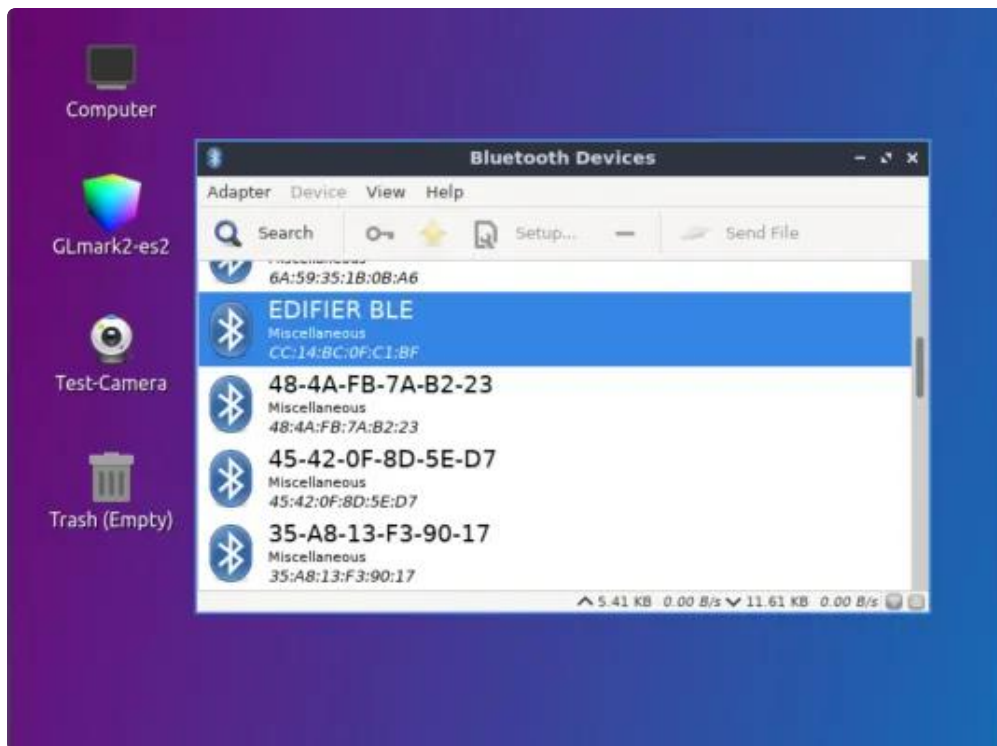
```
1 root@ido:~# bluetoothctl
2 [bluetooth]# scan on
3 ...
4 [bluetooth]# trust 7C:C1:80:09:DD:6C
5 [bluetooth]# pair 7C:C1:80:09:DD:6C
6 [cainiaocl]# exit
```

也可在桌面上连接蓝牙设备，步骤如下：

在桌面右下角蓝牙图标右键->Devices：



选中要连接的蓝牙设备，右键->pair:



3. IDO-IPC3528-V1 Buildroot系统

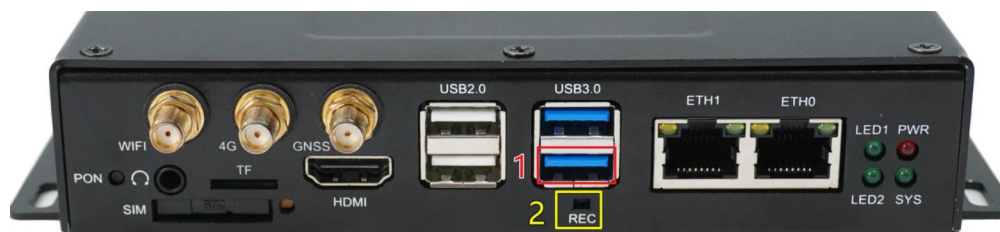
3.1. 外围功能接口

3.1.1. 固件烧录

固件地址: https://pan.baidu.com/s/1uZwjJMplS1xBMyd_4_2A9w?pwd=1234

主板下载固件需要将设备切换为 Loader 模式，Loader 模式的操作流程方法如下：

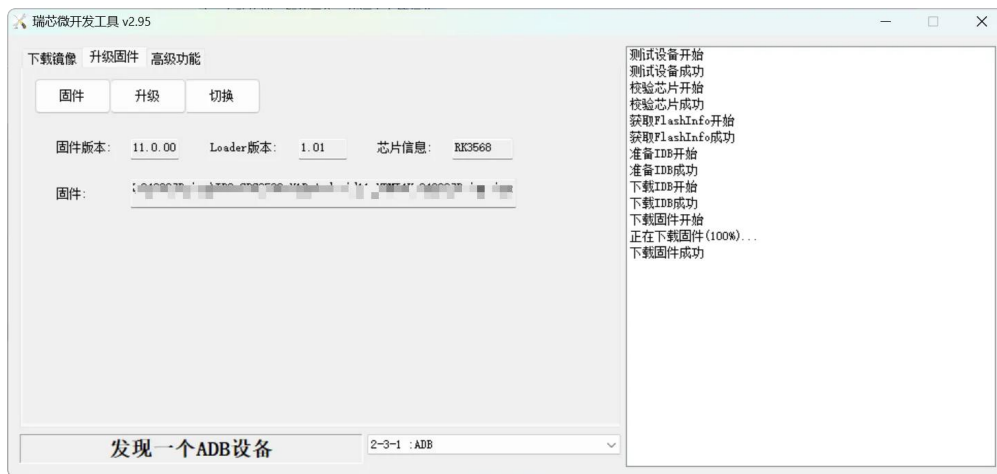
1. 断开整机的所有电源；
2. 使用USB Type A数据线，连接PC端的USB接口（如下图红色框所示）；
3. 按住REC按键并保持（如下图黄色框所示），并保持；
4. 给整机供电；



5. 烧录工具显示“发现一个LOADER设备”后，可松开REC按键；
6. 点击烧录工具【升级固件】界面的【固件】，在系统文件资源管理器中选择烧录固件；



7. 固件加载完成，点击【升级】按键开始烧录程序，烧录完成开发板将会重新启动。



3.1.2. 指示灯

设备上电后，绿色SYS指示灯亮起，正常启动后以1秒2次频率闪动，如下图所示：



- 开机：绿灯闪烁
- 关机：绿灯熄灭
- 异常：绿灯常亮（或灭）

名称	功能	说明
SYS	系统指示灯	上电后系统正常运行时1秒闪烁2次。
PWR	电源指示灯	上电后，常亮
LED1	用户自定义LED灯	默认不亮
LED2	用户自定义LED灯	默认不亮

3.1.3. REC按键

【REC】按键，如下图所示：



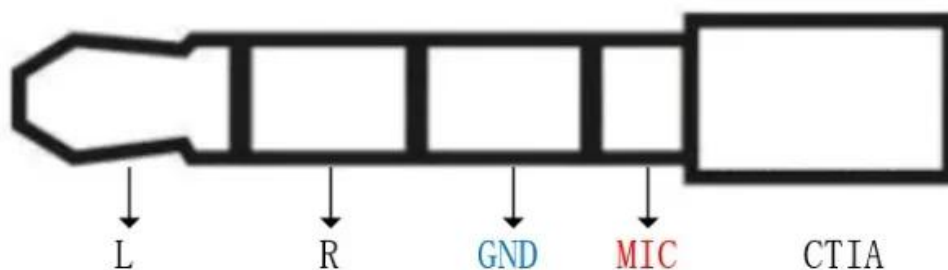
系统正常开机状态下按下【REC】按键，系统启动后为音量加键

3.1.4. 耳机

耳机接口，如下图所示：



CTIA标准四段式耳机，定义如下：



播放音频

查看声卡设备

Bash

```
1 [root@RK356X:~]# aplay -l
2 **** List of PLAYBACK Hardware Devices ****
3 card 0: rockchipdmi [rockchip,dmi], device 0: fe400000.i2s-i2s-hifi i2s-hifi-0 [fe400000.i2s-i2s-hifi i2s-hifi-0]
4   Subdevices: 1/1
5   Subdevice #0: subdevice #0
6 card 1: rockchiprk809co [rockchip,rk809-codec], device 0: fe410000.i2s-rk817-hifi rk817-hifi-0 [fe410000.i2s-rk817-hifi rk817-hifi-0]
7   Subdevices: 1/1
8   Subdevice #0: subdevice #0
```

播放到耳机

设置只输出到耳机：

Bash

```
1 amixer -c 1 cset numid=2,iface=MIXER,name='Playback Path' 'HP'
```

插入耳机，执行以下命令播放音频到耳机：

Bash

```
1 aplay -D plughw:1,0 /etc/bsa_file/8k16bpsStereo.wav
```

声音通过3.5mm耳机孔输出到耳机，调节声音大小，音乐清晰无杂音。

测试项目	要求	结果
3.5MM耳机孔	适配3.5耳机，声音清晰无杂音	✓

3.1.5. Mic录音

打开录音开关：

Bash

```
1 amixer -c 1 cset numid=3,iface=MIXER,name='Capture MIC Path' 'Main Mic'
```

使用arecord工具可以进行录音测试：

Bash

```
1 [root@RK356X:/]# arecord -D hw:1,0 -r 48000 -c 2 -f S16_LE test.wav
2 Recording WAVE 'test.wav' : Signed 16 bit Little Endian, Rate 48000 Hz, Stereo
3
4 ^CAborted by signal 中断...
```

录音完后播放测试：

Bash

```
1 [root@RK356X:/]# aplay -D plughw:1,0 ./test.wav
2 Playing WAVE './test.wav' : Signed 16 bit Little Endian, Rate 48000 Hz, Stereo
```

3.1.6. TF卡槽

TF卡槽，如下图所示：



支持FAT32和NTFS格式分区自动挂载

TF卡正面朝上，触点朝下插入盒子TF卡槽

测试项目	要求	结果
TF卡	识别TF卡正常、读写文件正常	✓

3.1.7. HDMI

HDMI接口，如下图所示：

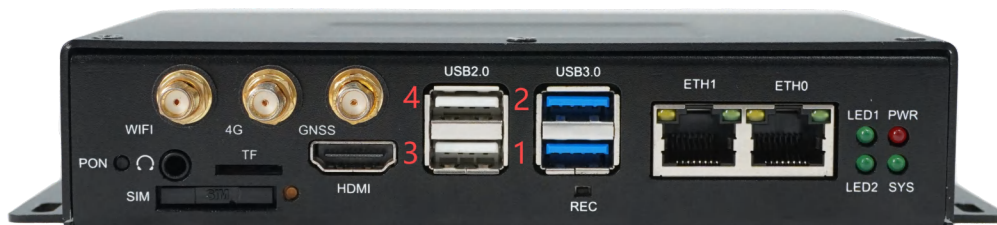


支持最大分辨率：4K@60fps

测试项目	要求	结果
显示	HDMI画面显示正常	✓
	HDMI音频输出正常	✓
	测试分辨率：4K@60fps	✓
	HDMI热插拔正常	✓

3.1.8. USB

USB接口，如下图所示：



测试项目	要求	结果
USB功能测试	USB1支持host 和device 模式的切换	✓
	USB2需要兼容U盘、键盘、鼠标	✓
	USB3需要兼容U盘、键盘、鼠标	✓

	USB4需要兼容U盘、键盘、鼠标	✓
--	------------------	---

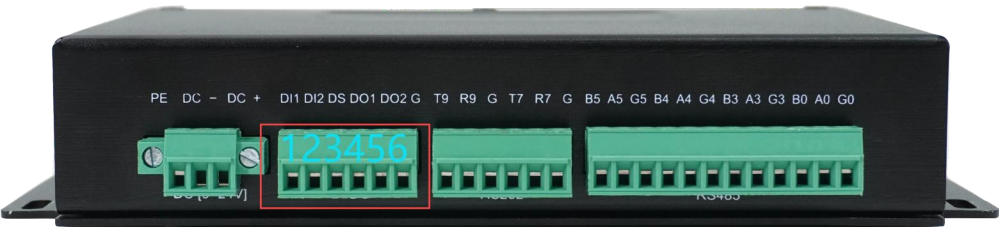
USB电源开关控制：

USB端口	动作	命令
USB1	切换HOST模式	echo host >/sys/devices/platform/fe8a8000.usb2-phy/otg_mode echo 255 > /sys/class/leds/usb3_otg_pwr/brightness
	切换DEVICE模式	echo peripheral>/sys/devices/platform/fe8a0000.usb2-phy/otg_mode echo 0 > /sys/class/leds/usb3_otg_pwr/brightness
USB2	关闭电源	echo 0 > /sys/class/leds/usb3_host_pwr/brightness
	开启电源	echo 1 > /sys/class/leds/usb3_host_pwr/brightness
USB3	关闭电源	echo 0 > /sys/class/leds/usb2_host_pwr/brightness
	开启电源	echo 1 >/sys/class/leds/usb2_host_pwr/brightness
USB4	关闭电源	echo 0 > /sys/class/leds/usb2_fe2_pwr/brightness
	开启电源	echo 1 > /sys/class/leds/usb2_fe2_pwr/brightness

测试项目	要求	结果
USB开关测试	USB1 OTG测试	✓
	USB2开关测试	✓
	USB3开关测试	✓
	USB4开关测试	✓

3.1.9. DIDO

DIDO接口（与CAN接口复用，二选一），如下图所示：



各个引脚定义如下表所示：

引脚	功能
1	DI1
2	DI2
3	DI_SS
4	DO1
5	DO2
6	GND

DI的使用

通过读取设备节点/dev/ido_di，获取DI1和DI2的输入电平状态。打开/dev/ido_di设备文件，读取3个字节，前2个字节为两路DI的输入结果。

通过代码读取举例：

```

1  #include <stdio.h>
2  #include <sys/types.h>
3  #include <sys/stat.h>
4  #include <fcntl.h>
5  #include <unistd.h>
6  #include <stdlib.h>
7
8  int main(void)
9  {
10     char buf[3] = {0, 0, 0};
11
12     int fd;
13     int ret;
14
15     fd = open("/dev/ido_di", O_RDONLY);
16
17     if (fd < 0){
18         exit(1);
19     }
20
21     ret = read(fd, buf, 3);
22     if (ret != 3){
23         perror("read");
24         close(fd);
25         exit(2);
26     }
27
28     printf("DI1输入%s\n", buf[0] == '1' ? "低电平" : "高电平");
29     printf("DI2输入%s\n", buf[1] == '1' ? "低电平" : "高电平");
30
31     close(fd);
32
33     return 0;
34 }

```

通过命令行来获取举例：

[illegible]

返回结果前2个字符与DI1、DI2的输入关系见下表:

结果	DI1输入	DI2输入
00	高	高
10	低	高
01	高	低
11	低	低

DO的使用

通过向/dev/ido_do设备文件写入特定字符串，控制DO1和DO2的输出。

通过代码控制输出举例：

```

1  #include <stdio.h>
2  #include <sys/types.h>
3  #include <sys/stat.h>
4  #include <fcntl.h>
5  #include <unistd.h>
6  #include <stdlib.h>
7
8  int main(void)
9  {
10     int fd;
11     int ret;
12
13     fd = open("/dev/ido_do", O_WRONLY);
14
15     if (fd < 0){
16         exit(1);
17     }
18
19     //D01输出低电平
20     printf("D01 输出低电平\n");
21     ret = write(fd, "1D", 3);
22     if (ret != 3)
23         goto write_err;
24     sleep(3);
25     //D01输出高电平
26     printf("D01 输出高电平\n");
27     ret = write(fd, "1E", 3);
28     if (ret != 3)
29         goto write_err;
30     sleep(3);
31
32     //D02输出低电平
33     printf("D02 输出低电平\n");
34     ret = write(fd, "2D", 3);
35     if (ret != 3)
36         goto write_err;
37     sleep(3);
38
39     //D01输出高电平
40     printf("D02 输出高电平\n");
41     ret = write(fd, "2E", 3);
42     if (ret != 3)
43         goto write_err;
44     sleep(3);
45

```

```

46         close(fd);
47
48         return 0;
49
50     write_err:
51         close(fd);
52         return -1;
53 }

```

通过命令行来设置输出举例：

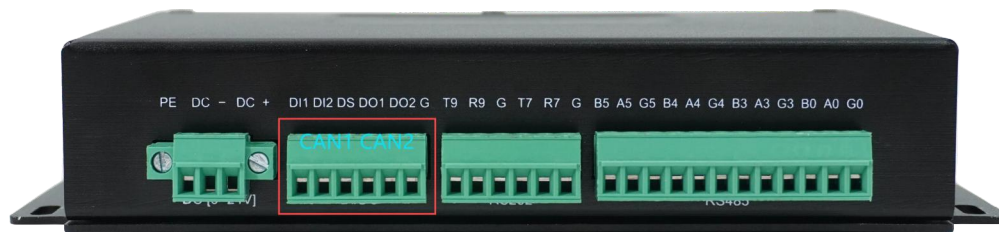
```

1  //D01输出0V
2  echo 1D > /dev/ido_do
3  //D01输出12V
4  echo 1E > /dev/ido_do
5
6  //D02输出0V
7  echo 2D > /dev/ido_do
8  //D02输出12V
9  echo 2E > /dev/ido_do

```

3.1.10. CAN

CAN接口（与DIDO接口复用，二选一），如下图所示：



测试

打开can设备，并设置通信波特率为250Kbps：

▼

Bash

```
1 sudo ip link set can0 down
2 sudo ip link set can0 type can bitrate 250000
3 sudo ip link set can0 up
4
5 sudo ip link set can1 down
6 sudo ip link set can1 type can bitrate 250000
7 sudo ip link set can1 up
```

发送测试:

▼

Bash

```
1 sudo cansend can0 123#DEADBEEF
```

接收测试:

▼

Bash

```
1 sudo candump can0
```

3.1.11. UART

主机共配置6路UART，其中4路RS485，2路RS232，如下:



各个UART对应的设备节点如下表:

编号	设备节点	类型
1	/dev/ttyS0	RS485
2	/dev/ttyS3	RS485
3	/dev/ttyS4	RS485
4	/dev/ttyS5	RS485

5	/dev/ttyS7	RS232
6	/dev/ttyS9	RS232

测试

RS485和RS232都可以使用microcom工具进行简单的收发测试。

以测试/dev/ttyS0为例：

```

1 [root@RK356X:/]# microcom -s 115200 /dev/ttyS0

```

按下键盘任意键会发送对应的字符，而接收的内容会显示在终端。

按【ctrl】和【x】组合键，然后输入quit退出测试。

```

1 [root@RK356X:/]# microcom -s 115200 -p /dev/ttyS0
2 [root@RK356X:/]#

```

3.2. 网络测试

3.2.1. Ethernet

主板有两路千兆以太网接口ETH0和ETH1，如下图所示：



ETH0

查看eth0的ip:

```
▼ Bash |
1 [root@RK356X:~]# ifconfig eth0
2 eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
3     inet 192.168.1.177 netmask 255.255.255.0 broadcast 192.168.1.255
4     inet6 fe80::4f32:fc2d:eb30:220b prefixlen 64 scopeid 0x20<link>
5     ether d2:e1:8d:d5:f7:49 txqueuelen 1000 (Ethernet)
6     RX packets 78 bytes 11727 (11.4 KiB)
7     RX errors 0 dropped 0 overruns 0 frame 0
8     TX packets 60 bytes 5822 (5.6 KiB)
9     TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
10    device interrupt 52
11
```

设置eth0的ip:

```
▼ Bash |
1 [root@RK356X:~]# ifconfig eth0 192.168.1.123
```

eth0 ping测试:

```
▼ Bash |
1 [root@RK356X:~]# ping www.baidu.com -I eth0
2 PING www.a.shifen.com (14.215.177.39) from 192.168.1.177 eth1: 56(84) byte
  s of data.
3 64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=1 ttl=54 time=6.75 ms
4 64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=2 ttl=54 time=6.32 ms
5 64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=3 ttl=54 time=7.01 ms
6 64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=4 ttl=54 time=7.47 ms
7 64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=5 ttl=54 time=7.01 ms
```

ETH1

查看eth1的ip:

```
▼ Bash |
1 [root@RK356X:~]# ifconfig eth1
2 eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
3     inet 192.168.1.177 netmask 255.255.255.0 broadcast 192.168.1.255
4     inet6 fe80::4f32:fc2d:eb30:220b prefixlen 64 scopeid 0x20<link>
5     ether d2:e1:8d:d5:f7:49 txqueuelen 1000 (Ethernet)
6     RX packets 78 bytes 11727 (11.4 KiB)
7     RX errors 0 dropped 0 overruns 0 frame 0
8     TX packets 60 bytes 5822 (5.6 KiB)
9     TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
10    device interrupt 52
11
```

设置eth1的ip:

```
▼ Bash |
1 [root@RK356X:~]# ifconfig eth1 192.168.1.123
```

eth1 ping测试:

```
▼ Bash |
1 [root@RK356X:~]# ping www.baidu.com -I eth1
2 PING www.a.shifen.com (14.215.177.39) from 192.168.1.177 eth1: 56(84) byte
  s of data.
3 64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=1 ttl=54 time=6.75 ms
4 64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=2 ttl=54 time=6.32 ms
5 64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=3 ttl=54 time=7.01 ms
6 64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=4 ttl=54 time=7.47 ms
7 64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=5 ttl=54 time=7.01 ms
```

3.2.2 4G/5G 网络

4G/5G 拨号

正确按照4G模块、SIM卡和4G天线后,使用quectel-CM工具拨号, 命令如下:

```
▼ Bash |
1 [root@RK356X:/]# /usr/bin/quectel-CM &
```

拨号成功，则会产生wwan0/usb0网络节点：

```
1 [root@RK356X:/]# ifconfig wwan0
2 wwan0      Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
3            inet addr:10.122.45.92  P-t-P:10.122.45.92  Mask:255.255.255.248
4            UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
5            RX packets:12 errors:0 dropped:0 overruns:0 frame:0
6            TX packets:59 errors:0 dropped:0 overruns:0 carrier:0
7            collisions:0 txqueuelen:1000
8            RX bytes:1396 (1.3 KiB)  TX bytes:8214 (8.0 KiB)
```

4G/5G 上网测试

使用ping命令测试4G/5G上网功能是否正常：

```
1 [root@RK356X:/]# ping www.baidu.com -I wwan0
2 PING www.a.shifen.com (14.119.104.189) from 10.122.45.92 wwan0: 56(84) bytes of data.
3 64 bytes from 14.119.104.189 (14.119.104.189): icmp_seq=1 ttl=52 time=95.5 ms
4 64 bytes from 14.119.104.189 (14.119.104.189): icmp_seq=2 ttl=52 time=99.3 ms
5 64 bytes from 14.119.104.189 (14.119.104.189): icmp_seq=3 ttl=52 time=97.8 mss
```

3.2.3WIFI

主机配置了一路2.4G WiFi模块，型号为AW-NM372SM。

系统上电默认会打开WiFi，对应的网络设备节点为wlan0。

```

1 root@ido:~$ sudo ifconfig wlan0
2 wlan0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
3         ether 2c:d2:6b:10:ea:4d txqueuelen 1000 (Ethernet)
4         RX packets 0 bytes 0 (0.0 B)
5         RX errors 0 dropped 0 overruns 0 frame 0
6         TX packets 0 bytes 0 (0.0 B)
7         TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

连接WiFi热点

使用脚本wifi_start.sh连接WiFi热点：

```

1 [root@RK356X:/]# wifi_start.sh "TP-LINK_B87A" "12345678"

```

其中"TP-LINK_B87A"为热点名称，"12345678"为热点密码。

```

1 [root@RK356X:/]# ifconfig wlan0
2 wlan0      Link encap:Ethernet HWaddr A8:41:F4:4F:B1:6D
3             inet addr:192.168.1.178 Bcast:192.168.1.255 Mask:255.255.255.0
4             inet6 addr: fe80::af5b:f8e4:625e:91f9/64 Scope:Link
5             UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
6             RX packets:5 errors:0 dropped:27 overruns:0 frame:0
7             TX packets:38 errors:0 dropped:0 overruns:0 carrier:0
8             collisions:0 txqueuelen:1000
9             RX bytes:1010 (1010.0 B) TX bytes:5221 (5.0 KiB)

```

3.2.4 Bluetooth

连接蓝牙设备

使用命令行连接蓝牙设备，命令如下：

```
1 [root@RK356X:/]# bluetoothctl
2 [bluetooth]# scan on
3 ...
4 [bluetooth]# trust 7C:C1:80:09:DD:6C
5 [bluetooth]# pair 7C:C1:80:09:DD:6C
6 [cainiaocl]# exit
```