

IDO-SBC3566-V1B-Ubuntu系统使用手册

1、调试

1.1 串口调试

1.2 ADB调试

1.3 ssh调试

2、串口

2.1 测试方法

2.2 RS485开启/关闭方法

3、USB

3.1 OTG模式切换

3.2 电源控制

4、TF CARD

5、以太网

5.1.1 查看以太网IP地址

5.1.1 使用命令查看

5.2 设置以太网临时IP地址

5.2.1 使用命令设置

5.3 设置以太网永久静态IP

5.4 网络唤醒

5.4.1 主板关机

5.4.2 开始唤醒

6、WiFi

6.1 连接热点

6.1.1 方式一

6.1.2 方式二

6.2 查看WiFi的IP地址

7、蓝牙

7.1 扫描设备

7.2 连接蓝牙设备

8、4G

9、音频

9.1 查看声卡设备

9.2 播放音频

9.3 音量的调节

9.4 录音

10、摄像头

10.1 测试

10.1.1 测试摄像头是否存在

10.1.3 抓取视频

11、RTC

11.1 获取RTC时间

11.2 设置RTC时间

11.3 RTC定时开机

12、IR红外功能

13、开机自启动

14、屏幕控制

14.1 背光调节

14.2 屏幕旋转

14.2.1 临时旋转

14.2.2 永久旋转

15、按键

16、ADC

16.1 ADC转换方法

16.2 测试

17、网络优先级设置

17.1 查看路由表

17.2 设置默认路由

17.2.1 设置WiFi为默认路由

17.2.2 设置以太网为默认路由

IDO-SBC3566-V1B

Ubuntu系统使用手册

深圳触觉智能科技有限公司

www.industio.cn

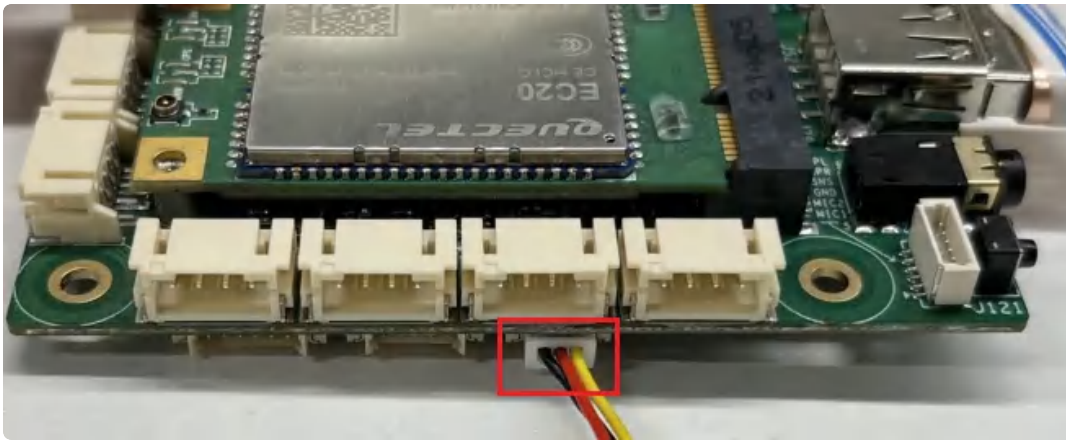
文档修订历史

版本	修订内容	修订	审核	日期
V1.0	创建文档	谭文学		2022/11/13
V1.1	增加串口、USB、TF CARD、以太网的描述；	谭文学		2022/11/15
V1.2	增加WiFi、蓝牙的描述；	谭文学		2022/11/17
V1.3	增加4G、RTC的描述；	谭文学		2022/11/18
V1.4	增加音频、IR红外功能的描述；	谭文学		2022/11/23
V1.5	增加ADC、网络优先级的描述；	谭文学		2022/11/24
V1.6	增加开机自启动、屏幕控制的描述；	谭文学		2022/11/29
V1.7	增加摄像头测试方法；	谭文学		2022/12/6

1、调试

1.1 串口调试

串口调试端口位于J4，通信参数为1500000 8 N 1，电平状态为TTL电平。



串口登录默认以root用户登录，且没有设置密码。

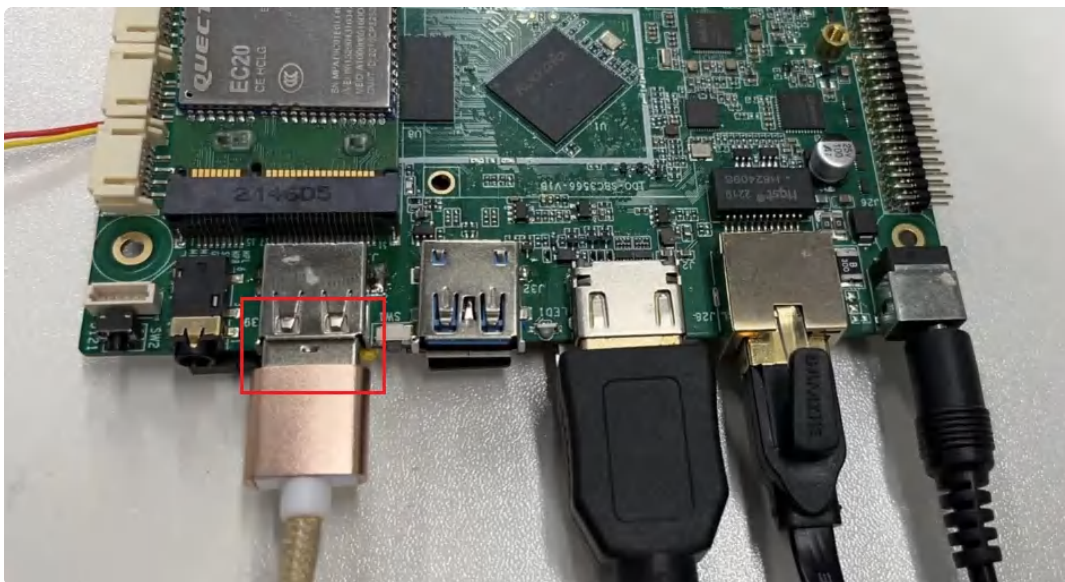
▼

Bash

```
1 root@ido:/# ls
2 app.log data home lost+found opt run srv tmp var
3 bin dev lib media proc sbin sys udisk vendor
4 boot etc log mnt root sdcard system usr
5 root@ido:/#
```

1.2 ADB调试

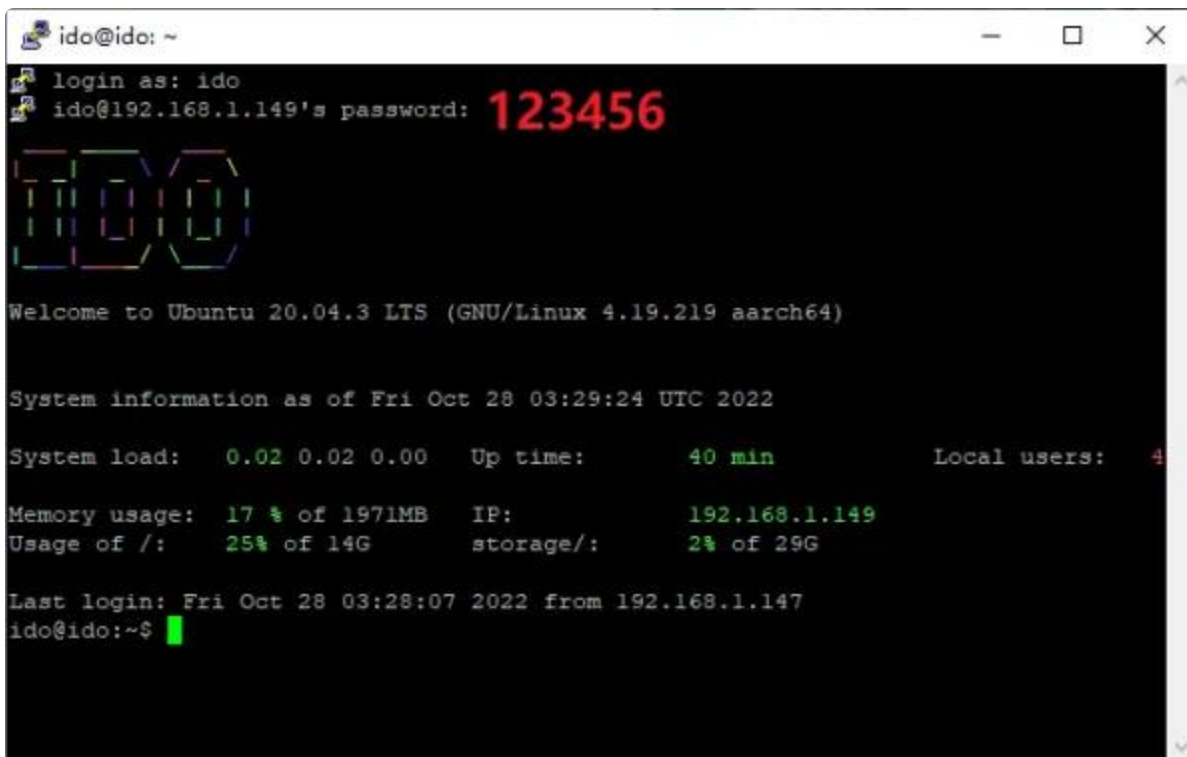
ADB调试端口位于J29，使用双公头USB线，连接开发板和电脑，即可在电脑上使用adb调试。



```
C:\Users\ronnie\Downloads>adb shell
adb server is out of date.  killing...
* daemon started successfully *
root@ido:/# ls
ls
app.log  data  home  lost+found  opt  run  srv  tmp  var
bin      dev  lib  media      proc sbin sys  udisk vendor
boot    etc  log  mnt        root sdcard system usr
root@ido:/#
```

1.3 ssh调试

系统默认ssh账号和密码为 ido @ 123456。



```
ido@ido: ~
login as: ido
ido@192.168.1.149's password: 123456

Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 4.19.219 aarch64)

System information as of Fri Oct 28 03:29:24 UTC 2022

System load:  0.02 0.02 0.00  Up time:    40 min  Local users:  4
Memory usage: 17 % of 1971MB  IP:        192.168.1.149
Usage of /:   25% of 14G      storage/:  2% of 29G

Last login: Fri Oct 28 03:28:07 2022 from 192.168.1.147
ido@ido:~$
```

2、串口

开发板共配置了5路串口，其中4路RS232，1路RS485。

序号	设备节点	默认类型	位置	备注
1	/dev/ttyS0	RS485	J46	可硬件配置为TTL或RS232
2	/dev/ttyS1	RS232	J47	可硬件配置为TTL
3	/dev/ttyS3	RS232	J45	可硬件配置为TTL

4	/dev/ttyS7	RS232	J43	可硬件配置为TTL或RS232
5	/dev/ttyS9	RS232	J44	可硬件配置为TTL或RS232

2.1 测试方法

TTL、RS485和RS232都可以使用microcom工具进行简单的收发测试。

首先安装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。

```
▼ Bash |
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 RS485开启/关闭方法

其中ttyS0、ttyS7和ttyS9可切换为TTL、RS232及RS485，当不是RS485时，需要把rs485_mode禁止：

```
▼ Bash |
1 root@ido:~# echo 0 > /sys/class/ido_ttyS0/rs485_mode
```

同理，当切换至RS485时，重新打开rs485_mode：

```
▼ Bash |
1 root@ido:~# echo 1 > /sys/class/ido_ttyS0/rs485_mode
```

查看当前是否为RS485模式：

```
▼ Bash |
1 root@ido:~# cat /sys/class/ido_ttyS0/rs485_mode
2 disable
```

返回enable说明当前处于RS485模式；返回disable说明当前不是RS485模式。

3、USB

IDO-SBC3566共配置了6个USB接口，分别为USB1-6。其中USB1为OTG（可切换host或device模式），USB2-6为HOST。

编号	类型	位置
usb1	otg	J52
usb2	host-2.0	J32
usb3	host-2.0	J33
usb4	host-2.0	J34
usb5	host-2.0	J35
usb6	host-2.0	J36

3.1 OTG模式切换

USB1默认为device模式，可用于adb调试。如果要切换host模式，执行以下命令

▼

Bash |

```
1 root@ido:~# echo HOST > /dev/otg_mode
```

当要从host切换到device模式，执行以下命令

▼

Bash |

```
1 root@ido:~# echo DEVICE > /dev/otg_mode
```

当USB-HOST插入U盘后，会自动挂载/media/ido/目录下。

▼

Bash |

```
1 root@ido:~# echo HOST > /dev/otg_mode
2 root@ido:~# ls /media/ido/
3 KINGSTON
```

3.2 电源控制

默认所有HOST的电源都是开启的，其中USB2-6我们提供了开启/关闭方法。

编号	电源控制节点
USB2	/sys/class/leds/usb_host1_pwr/brightness
USB3	/sys/class/leds/usb_host2_pwr/brightness
USB4	/sys/class/leds/usb_host3_pwr/brightness
USB5	/sys/class/leds/usb_host4_pwr/brightness
USB6	/sys/class/leds/usb_host5_pwr/brightness

打开USB2的电源：

```

▼ Bash |
1 root@ido:~# echo 255 > /sys/class/leds/usb_host1_pwr/brightness

```

关闭USB2的电源：

```

▼ Bash |
1 root@ido:~# echo 0 > /sys/class/leds/usb_host1_pwr/brightness

```

USB3-6的电源控制方法类似。

4、TF CARD

SBC3566配置了一个TF CARD接口，当TF CARD接口插入TF卡后，会自动挂载到/media/ido/目录下。

```

▼ Bash |
1 root@ido:~# ls /media/ido/
2 3533-3735
3 root@ido:~#

```

5、以太网

SBC3566配置了一个1000M以太网接口，对应的网络设备节点为eth0。

5.1.1 查看以太网IP地址

5.1.1 使用命令查看

系统默认以太网为动态获取IP，当以太网接口插入网线时，会自动获取IP。

```
1 root@ido:~# ifconfig eth0
2 eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
3     inet 192.168.1.133 netmask 255.255.255.0 broadcast 192.168.1.255
4     inet6 fe80::b7c2:552b:4127:ee1c prefixlen 64 scopeid 0x20<link>
5     ether 82:4c:21:62:f5:35 txqueuelen 1000 (Ethernet)
6     RX packets 13 bytes 2291 (2.2 KB)
7     RX errors 0 dropped 0 overruns 0 frame 0
8     TX packets 28 bytes 4056 (4.0 KB)
9     TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
10    device interrupt 40
11
12 root@ido:~#
```

5.2 设置以太网临时IP地址

5.2.1 使用命令设置

```
1 root@ido:~# ifconfig eth0 192.168.1.123
2 root@ido:~#
```

5.3 设置以太网永久静态IP

新建/etc/netplan/00-installer-config.yaml，然后写入如下内容（注意缩进以Tab为单位）：

```

1  network:
2      version: 2
3      renderer: NetworkManager
4      ethernets:
5          eth0:
6              dhcp4: no
7              dhcp6: no
8              addresses: [192.168.1.10/24]
9              gateway4: 192.168.1.1
10             nameservers:
11                 addresses: [8.8.8.8, 114.114.114.114]

```

然后重启网络

```
1 root@ido:~# netplan apply
```

重启网络后，eth0的ip地址已经变成刚才设置的静态ip。

```

1 root@ido:~# ifconfig eth0
2 eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
3     inet 192.168.1.123 netmask 255.255.255.0 broadcast 192.168.1.255
4     inet6 fe80::b7c2:552b:4127:ee1c prefixlen 64 scopeid 0x20<link>
5     ether 82:4c:21:62:f5:35 txqueuelen 1000 (Ethernet)
6     RX packets 19 bytes 3281 (3.2 KB)
7     RX errors 0 dropped 0 overruns 0 frame 0
8     TX packets 36 bytes 5592 (5.5 KB)
9     TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
10    device interrupt 40

```

设置静态IP后，断电重启设备依旧生效。

5.4 网络唤醒

SBC3566支持网络唤醒开机功能，主板关机后，在局域网中，使用WakeMeOnLan（Linux系统使用wakeonlan）工具可以网络唤醒主板开机。

以下测试在电脑上使用WakeMeOnLan唤醒SBC3566功能。

WakeMeOnLan工具下载地址：

链接: https://pan.baidu.com/s/1UavtDwbFEtzXLMYh_6AzTA?pwd=6bna

提取码: 6bna

5.4.1 主板关机

在关机前, 查看主板以太网的IP地址和MAC:

```
▼ Bash |
1 root@ido:~# ifconfig eth0
2 eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
3     inet 192.168.1.133 netmask 255.255.255.0 broadcast 192.168.1.255
4     inet6 fe80::b7c2:552b:4127:ee1c prefixlen 64 scopeid 0x20<link>
5     ether 82:4c:21:62:f5:35 txqueuelen 1000 (Ethernet)
6     RX packets 19 bytes 3281 (3.2 KB)
7     RX errors 0 dropped 0 overruns 0 frame 0
8     TX packets 36 bytes 5592 (5.5 KB)
9     TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
10    device interrupt 40
```

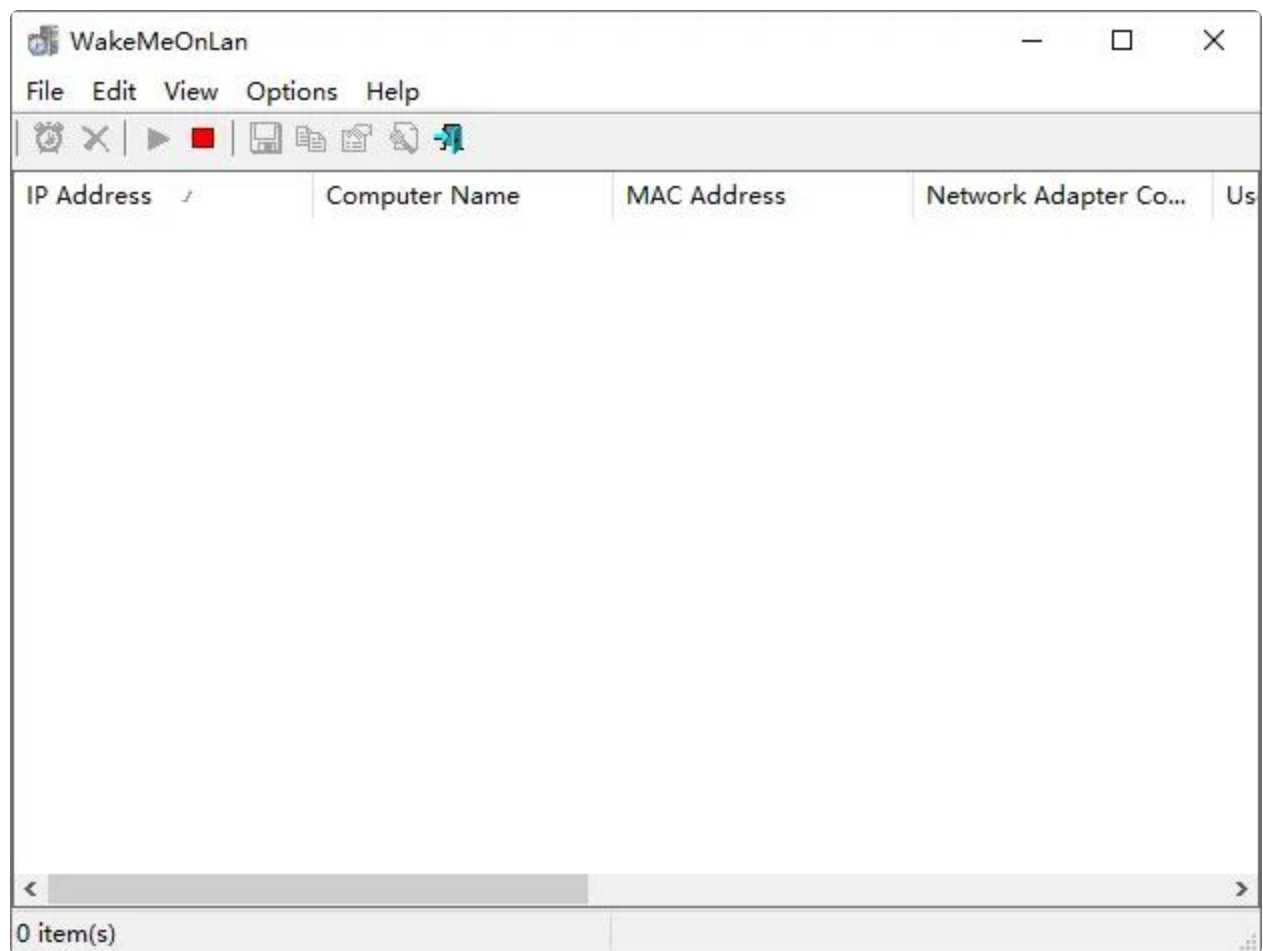
可以看到主板的以太网IP地址为192.168.1.133, MAC为82:4c:21:62:f5:35。

主板上执行poweroff命令进行关机:

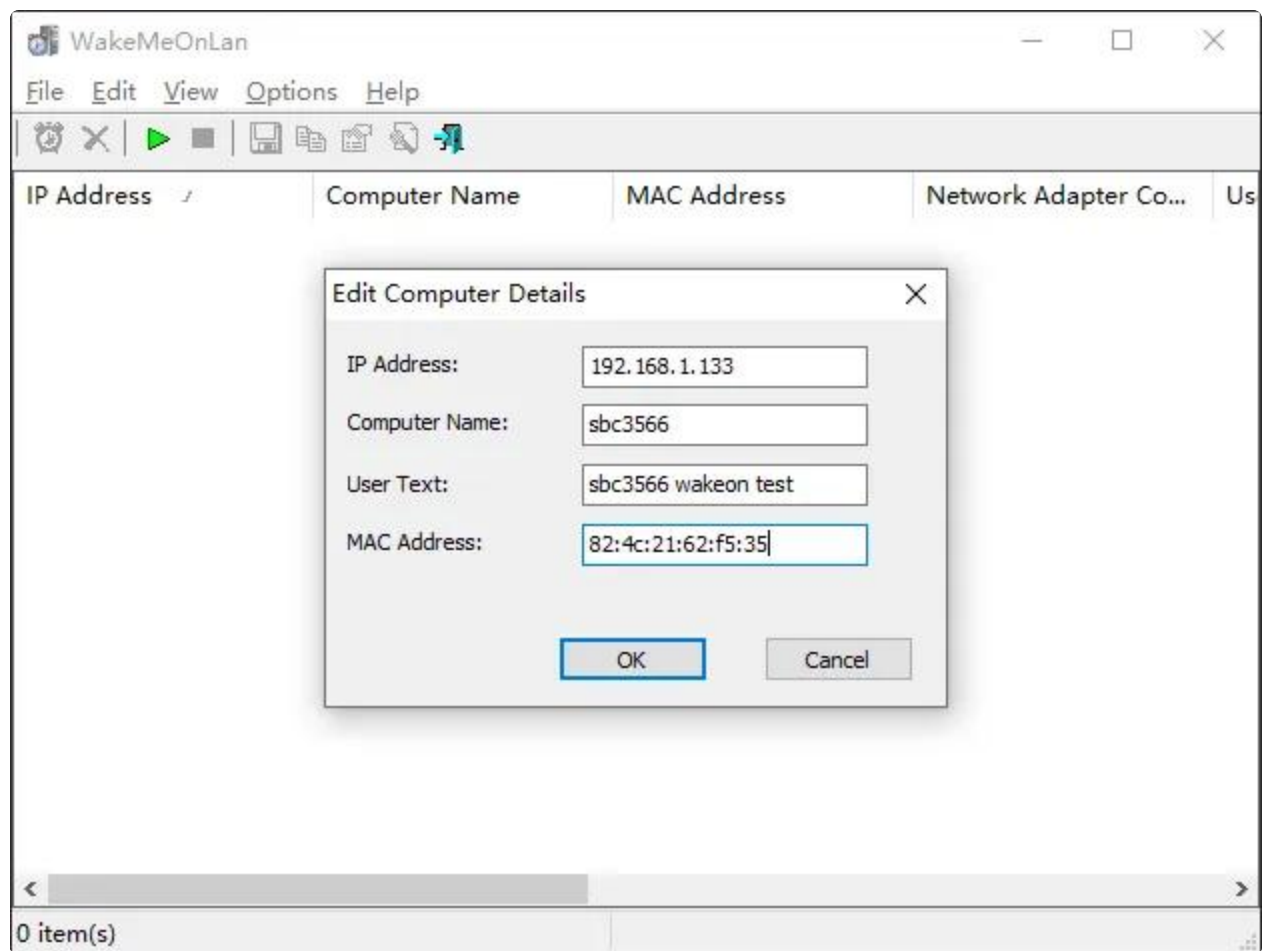
```
1 root@ido:~# poweroff
2 [ OK ] Removed slice system-modprobe.slice.
3 [ OK ] Stopped target Bluetooth.
4 [ OK ] Stopped target Graphical Interface.
5 [ OK ] Stopped target Multi-User System.
6 [ OK ] Stopped target Login Prompts.
7 [ OK ] Stopped target Host and Network Name Lookups.
8 [ OK ] Stopped target Sound Card.
9 ...
10 [ OK ] Stopped Remount Root and Kernel File Systems.
11 [ OK ] Stopped Clean the /media/i...9b58-91bf83198643 mount point.
12 [ OK ] Stopped Clean the /media/i...9450-eee71787380b mount point.
13 [ OK ] Removed slice system-clean\x2dmount\x2dpoint.slice.
14 [ OK ] Reached target Shutdown.
15 [ OK ] Reached target Final Step.
16 [ OK ] Finished Power-Off.
17 [ OK ] Reached target Power-Off.
18 [ 37.461906] Local MAC: 82:4c:21:62:f5:35
19 [ 37.465031] not set rk809 rtc alarm!
20 [ 37.469041] reboot: Power down
```

5.4.2 开始唤醒

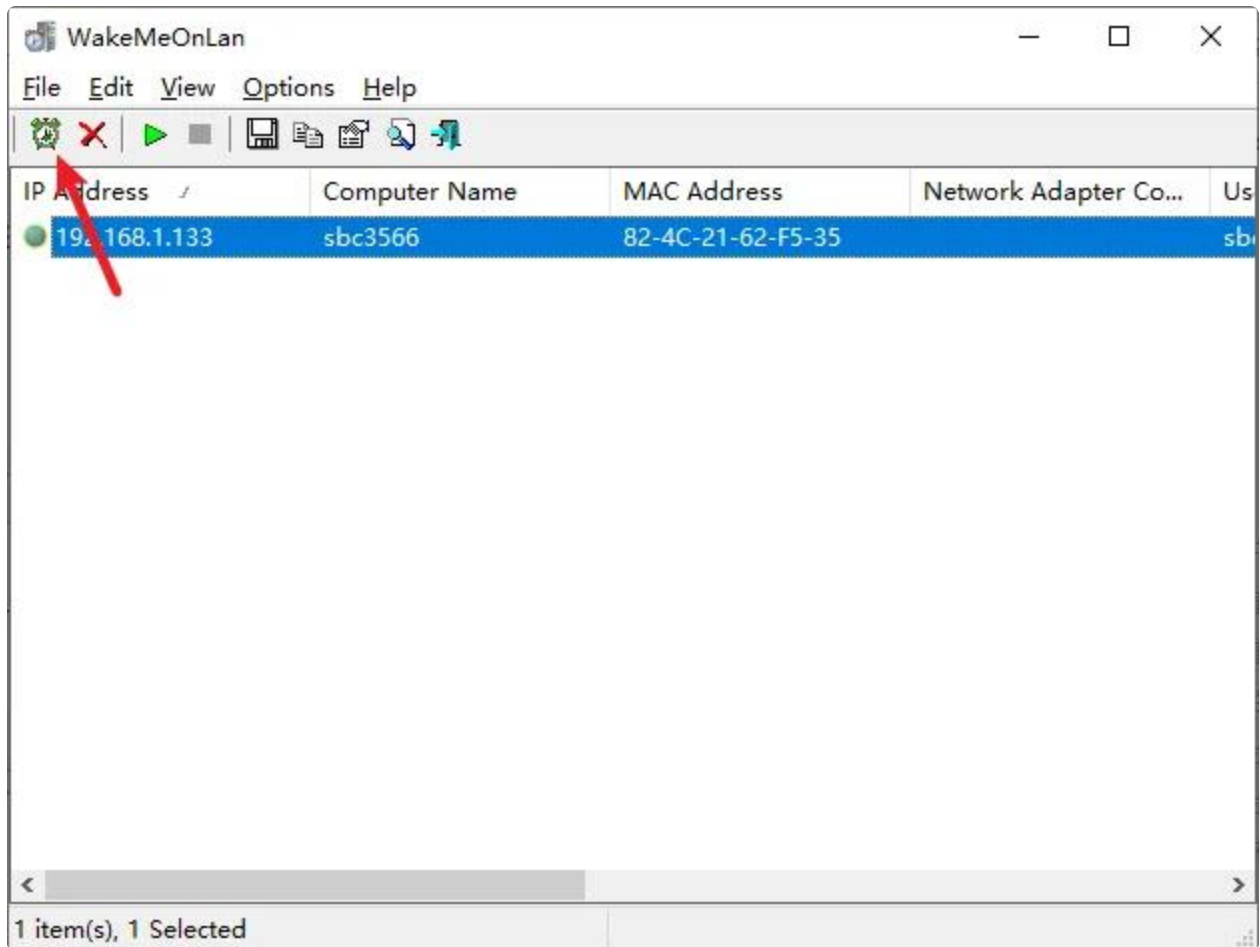
需要确保电脑与主板位于同一局域网内，电脑运行WakeMeOnLan.exe：



File->Add New Computer, 填写主板的IP和MAC:



然后点击唤醒按钮：



主板将被唤醒开机。

6、WiFi

系统上电默认会打开WiFi，对应的网络设备节点为wlan0。

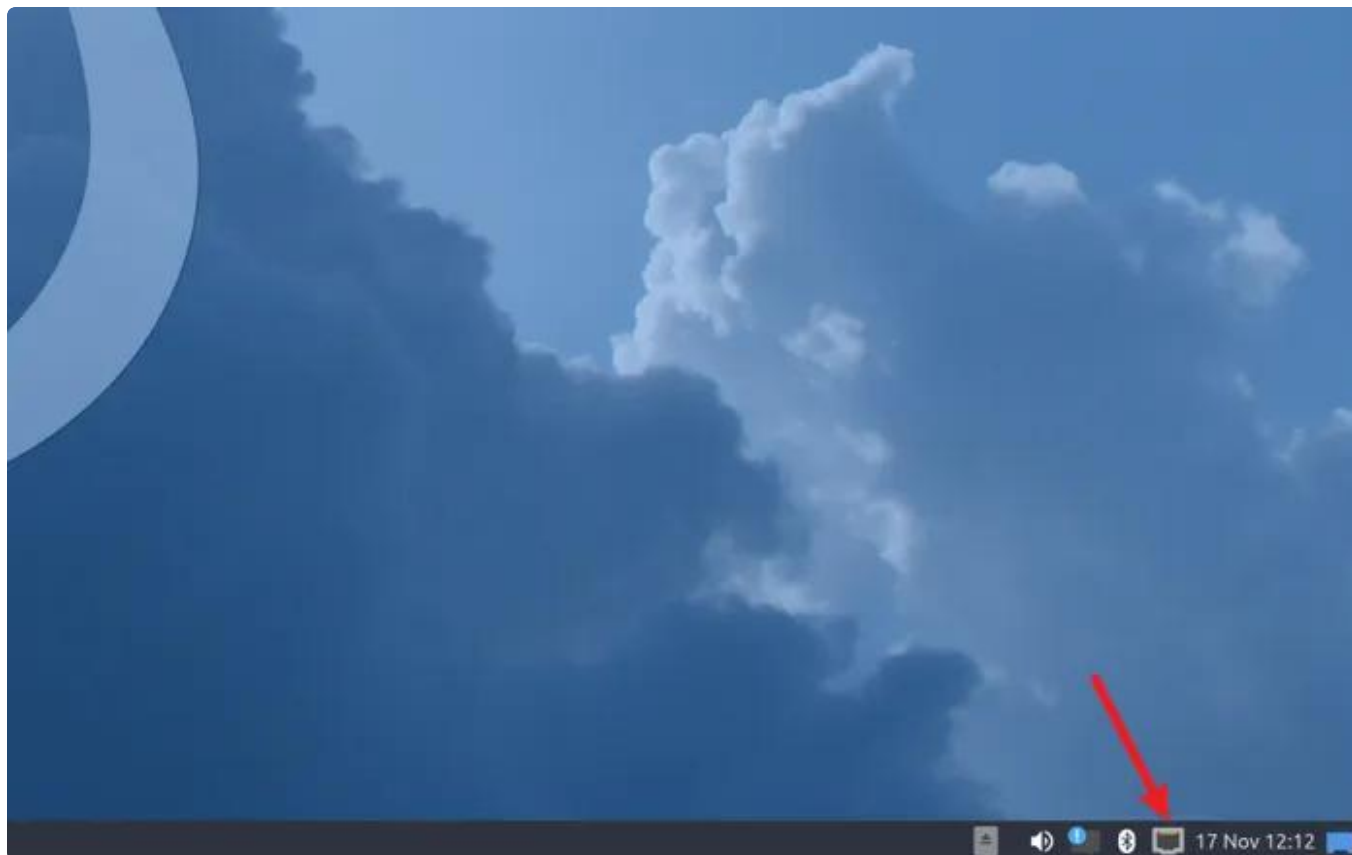
```
1 root@ido:~# ifconfig wlan0
2 wlan0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
3         ether 2c:d2:6b:10:ea:4d txqueuelen 1000 (Ethernet)
4         RX packets 0 bytes 0 (0.0 B)
5         RX errors 0 dropped 0 overruns 0 frame 0
6         TX packets 0 bytes 0 (0.0 B)
7         TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
8
9 root@ido:~#
```

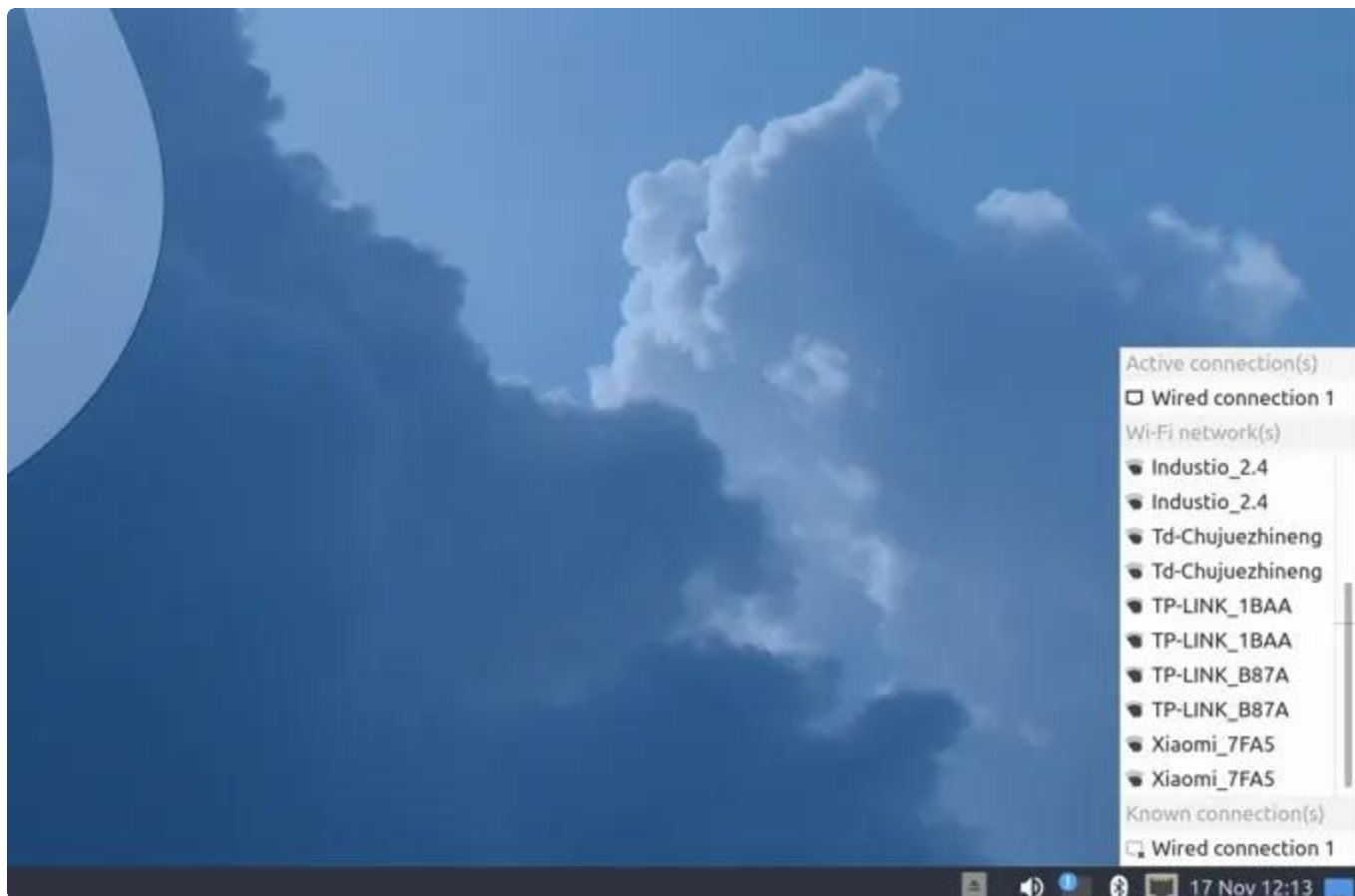
6.1 连接热点

连接热点可以在桌面上操作，也可以使用命令行操作。

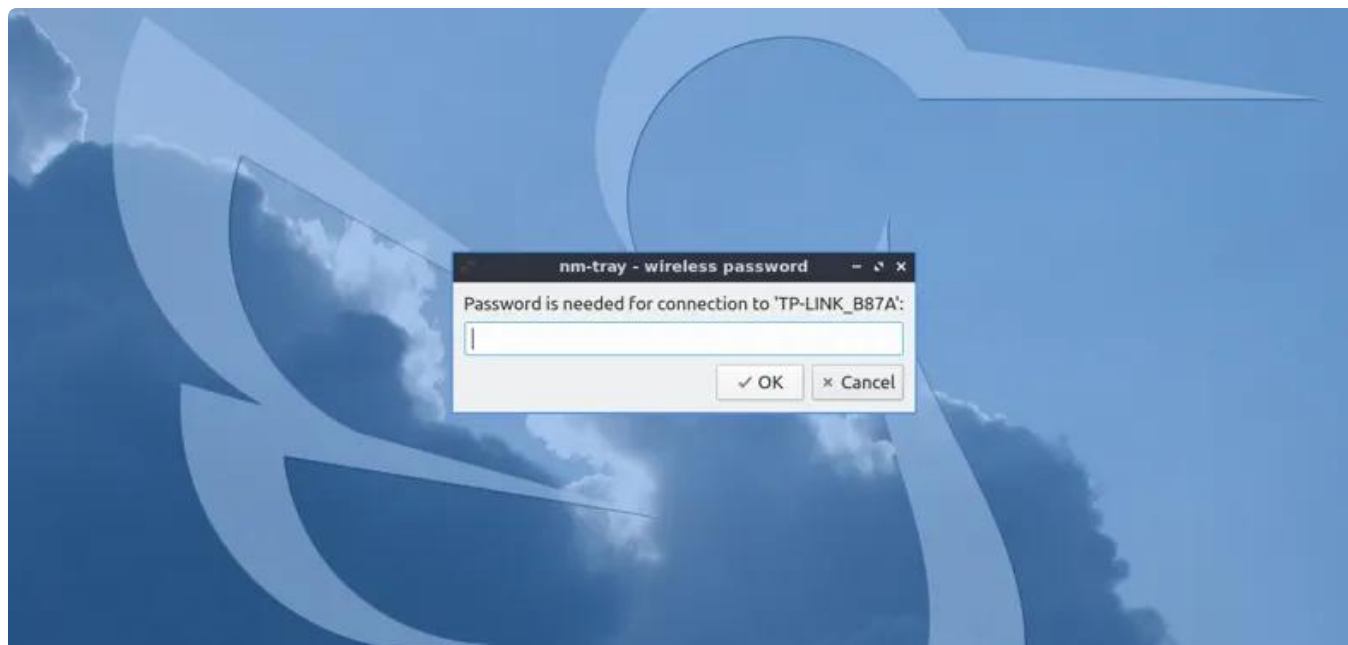
6.1.1 方式一

鼠标左键点击桌面右下角的网络图标，即可看到WiFi热点列表：



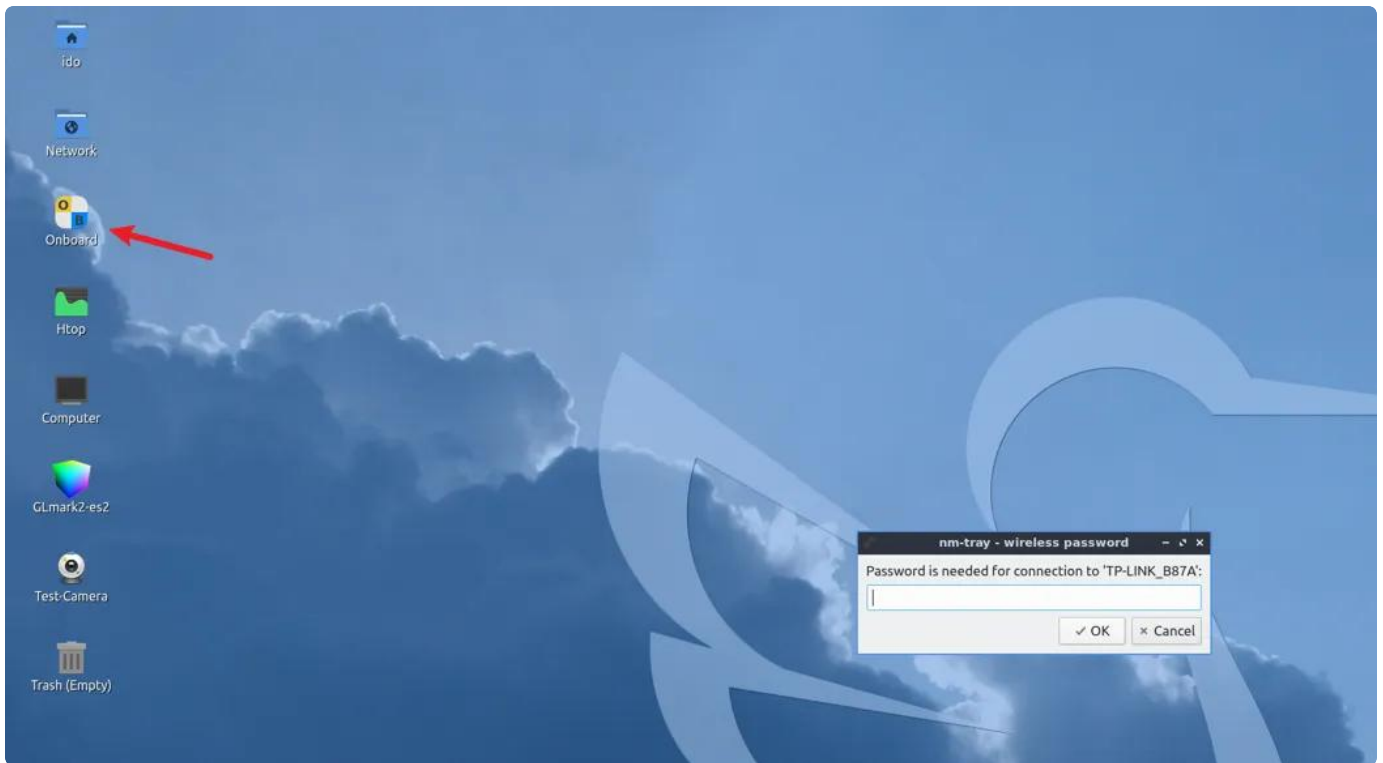


点击要连接的热点，弹出密码输入窗口：

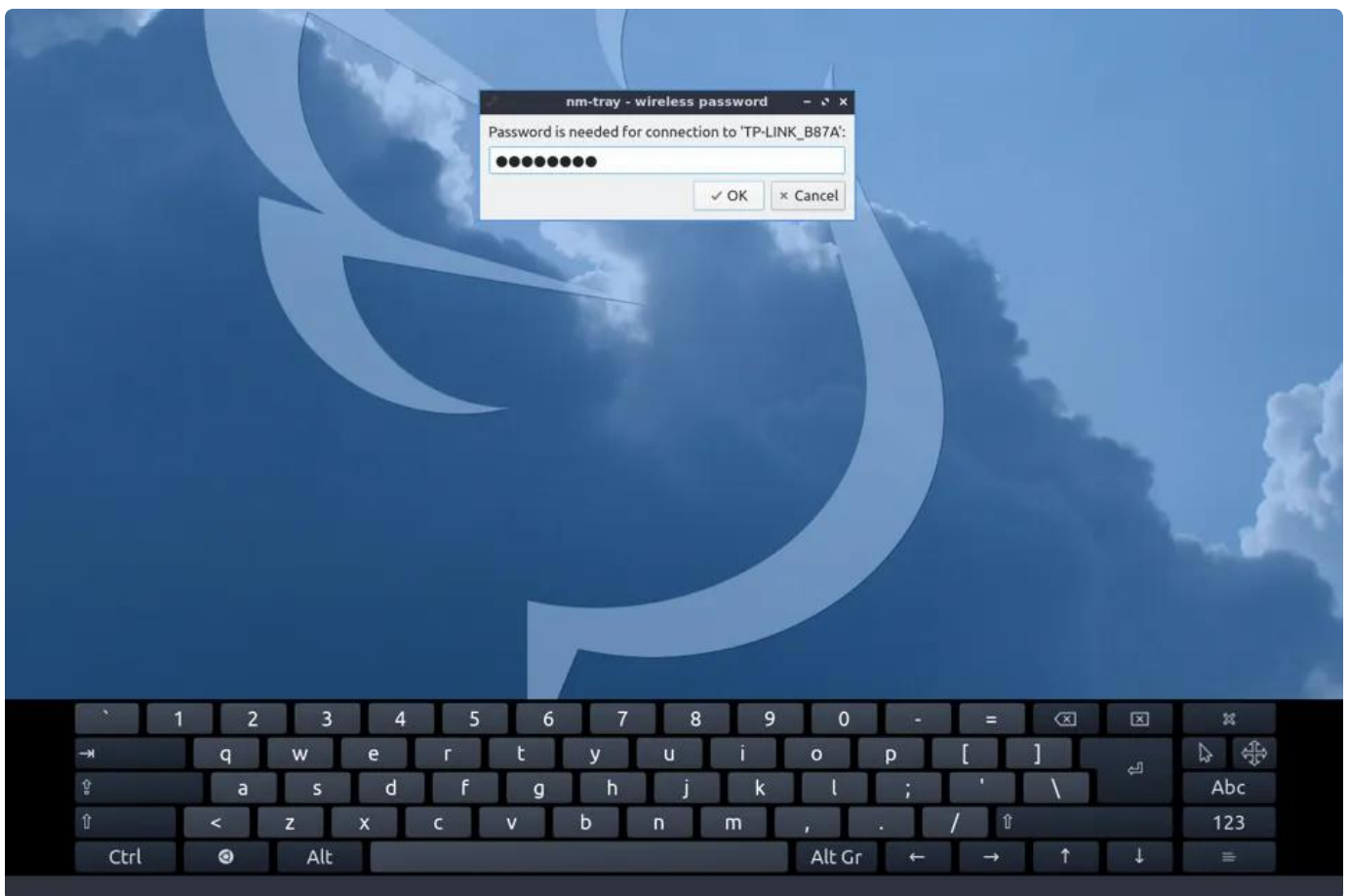


如果有连接键盘，直接输入密码即可；如果没有连接键盘，可以使用系统自带的软键盘。

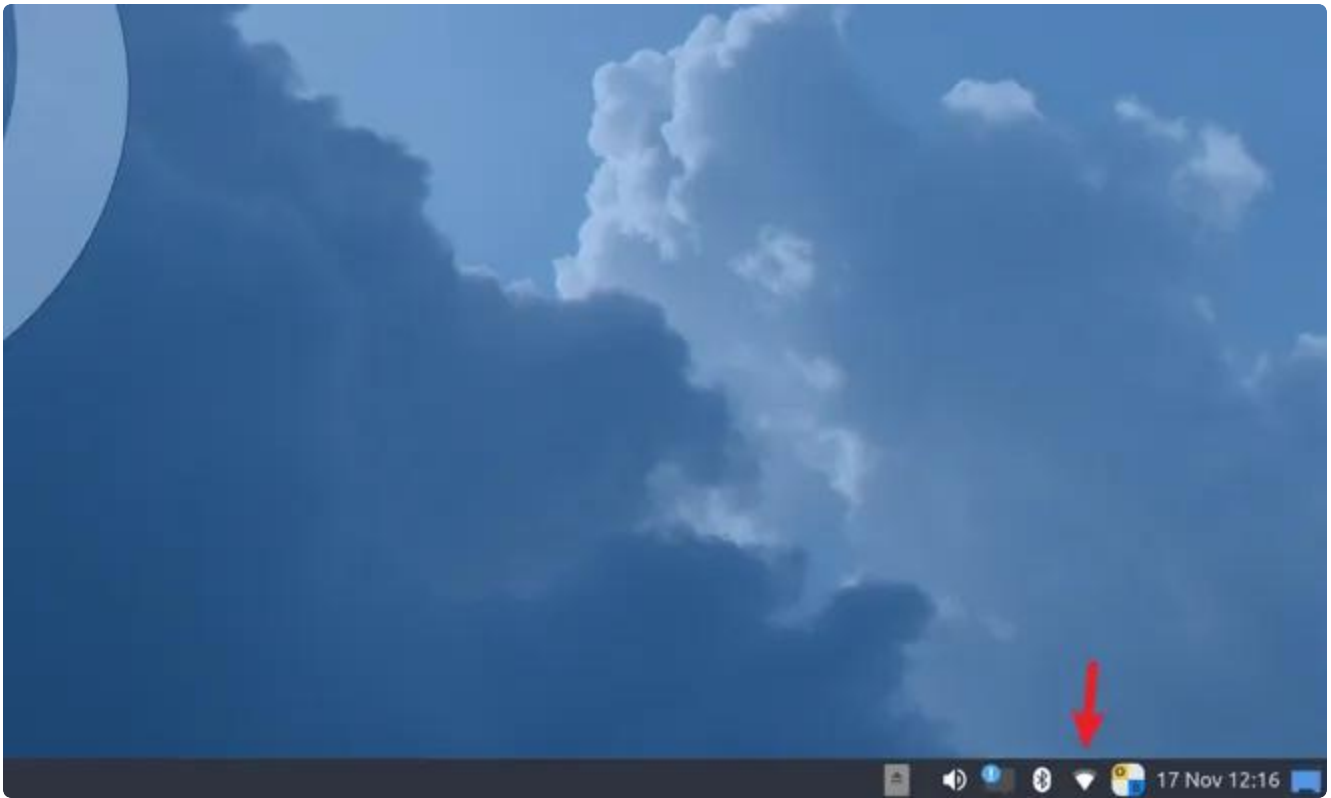
点击桌面的Onboard图标，即可打开系统自带的软键盘



使用软键盘输入密码后，点击【OK】连接热点



连接成功后，桌面右下角的网络图标将改变：



6.1.2 方式二

新建/etc/netplan/01-wifi-sta.yaml，并按照以下格式填写内容：

```
1  network:
2      wifis:
3          wlan0:
4              dhcp4: true
5              access-points:
6                  "TP-LINK_B87A":
7                      password: "12345678"
8      version: 2
```

其中TP-LINK_B87A为要连接的热点名称，12345678为连接密码。

修改成功后执行以下命令进行连接：

```
1  root@ido:~# killall wpa_supplicant
2  root@ido:~# netplan apply
```

等待几秒钟后，将成功连接WiFi热点。

6.2 查看WiFi的IP地址

使用ifconfig命令可查看连接热点后获取的IP

```
▼ Bash |
1 root@ido:~# 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::984a:9a2f:77b4:e899 prefixlen 64 scopeid 0x20<link>
5     ether 2c:d2:6b:10:ea:4d txqueuelen 1000 (Ethernet)
6     RX packets 83 bytes 10479 (10.4 KB)
7     RX errors 0 dropped 0 overruns 0 frame 0
8     TX packets 35 bytes 5285 (5.2 KB)
9     TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
10
11 root@ido:~#
12
```

测试wifi的网络

```
▼ Bash |
1 root@ido:~# ping www.baidu.com -I wlan0
2 PING www.a.shifen.com (14.215.177.39) from 192.168.1.117 p2p0: 56(84) bytes
  s of data.
3 64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=1 ttl=54 time=17.8 ms
4 64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=2 ttl=54 time=9.30 ms
5 64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=3 ttl=54 time=25.7 ms
6 64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=4 ttl=54 time=42.1 ms
7 64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=5 ttl=54 time=13.1 ms
8 64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=6 ttl=54 time=39.8 ms
9 64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=7 ttl=54 time=28.1 ms
```

7、蓝牙

系统开机默认打开蓝牙，对应的网络节点为hci0。

```
1 root@ido:~# hciconfig
2 hci0:   Type: Primary   Bus: UART
3         BD Address: 70:D5:2B:5B:22:22   ACL MTU: 1021:8   SCO MTU: 255:12
4         UP RUNNING
5         RX bytes:1665 acl:0 sco:0 events:57 errors:0
6         TX bytes:6311 acl:0 sco:0 commands:57 errors:0
7
8 root@ido:~#
```

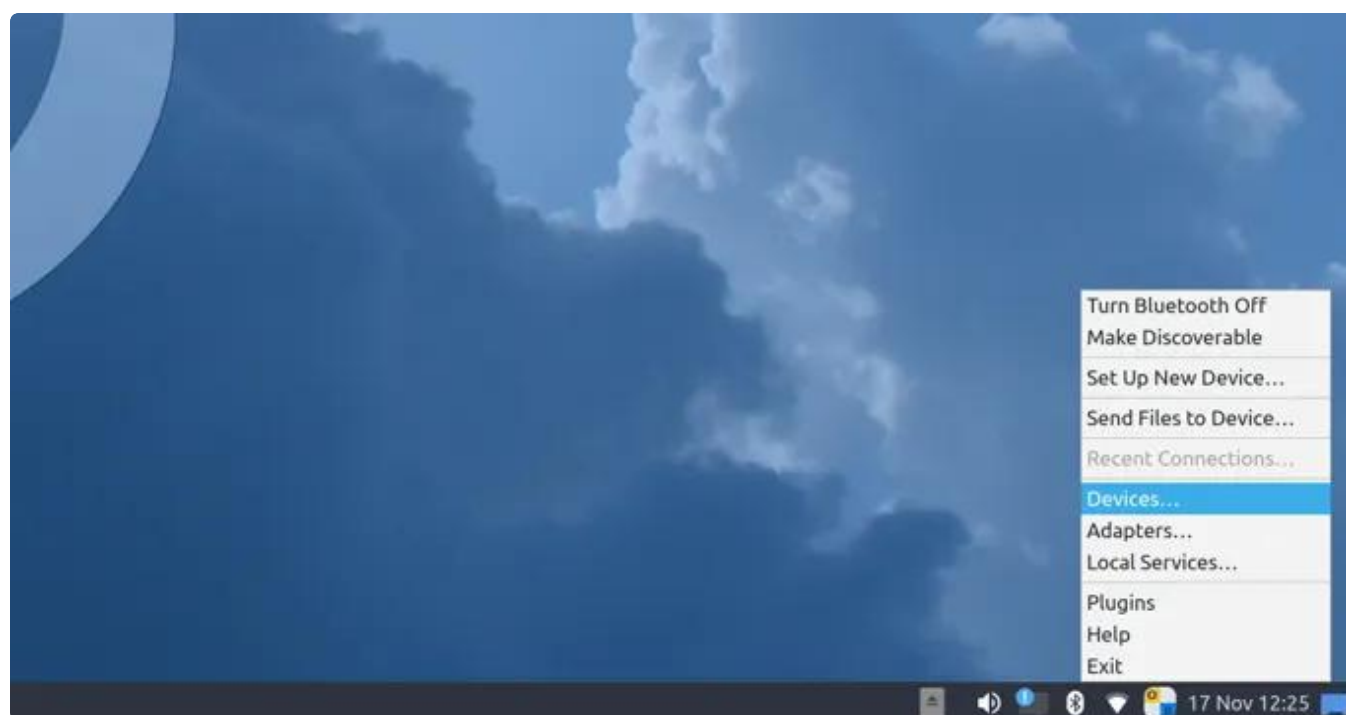
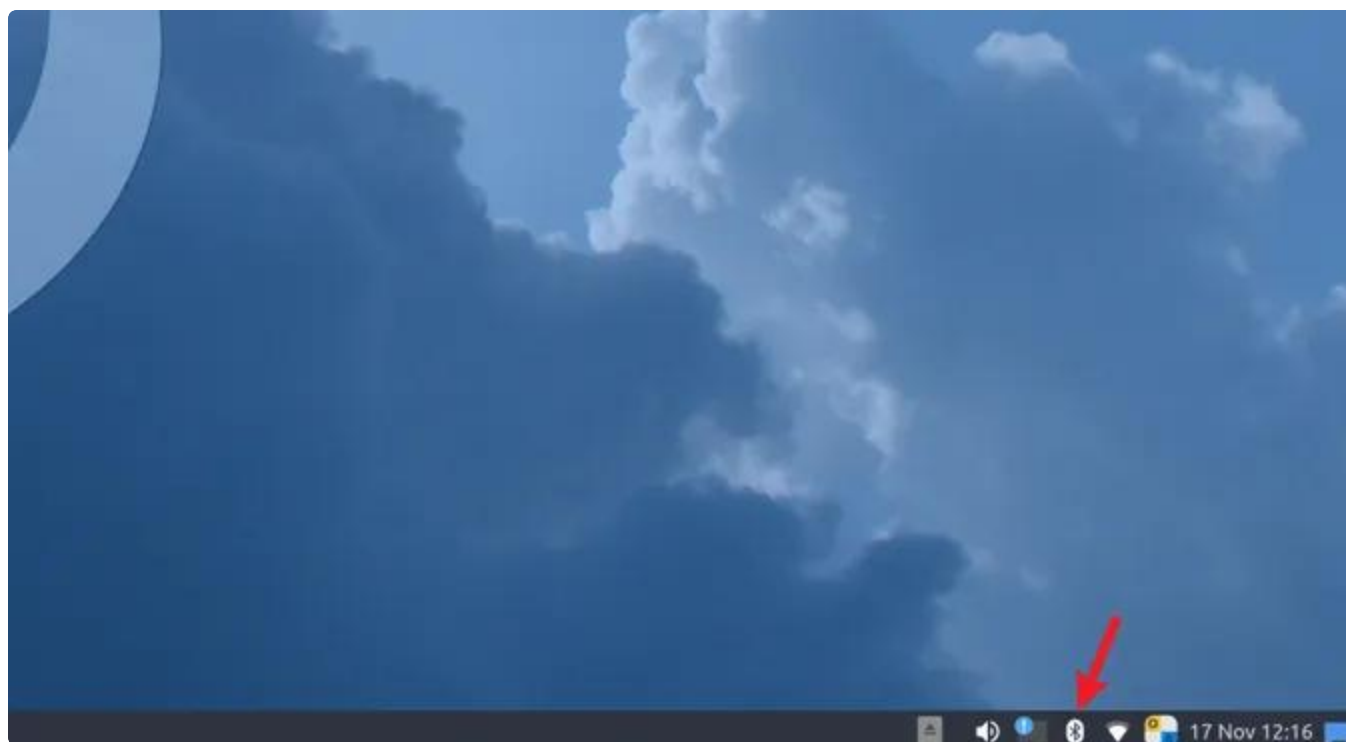
7.1 扫描设备

使用hcitool scan或hcitool lescan对附近的设备进行扫描：

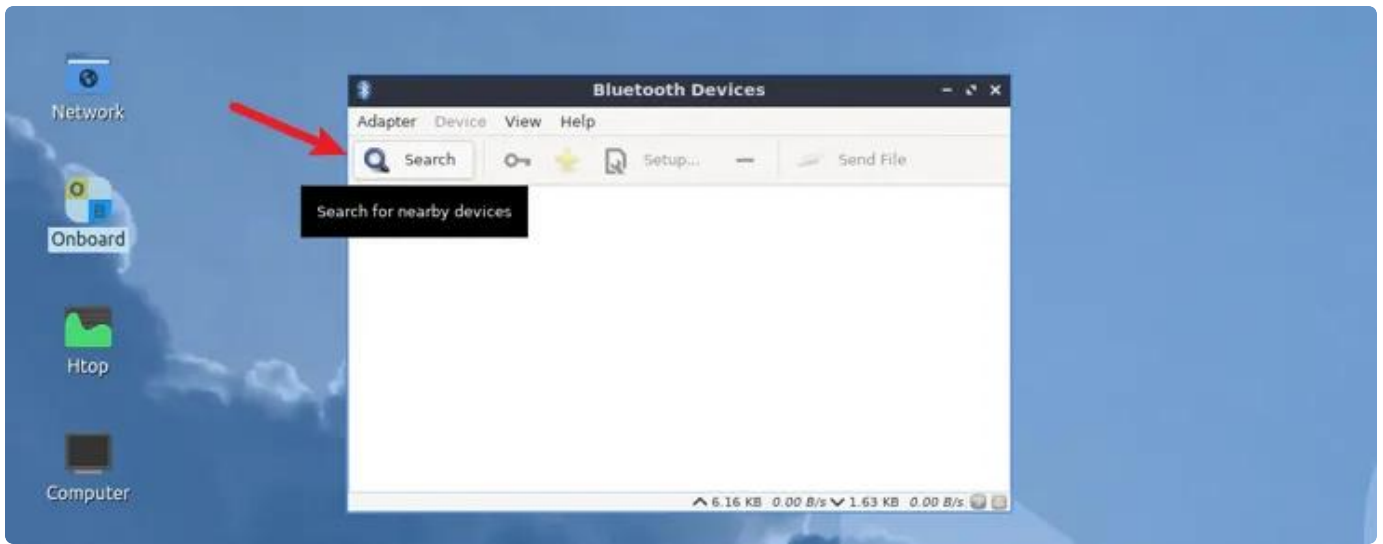
```
1 root@ido:~# hcitool scan
2 Scanning ...
3     94:87:E0:9D:14:12      seeyou
4 root@ido:~# hcitool lescan
5 LE Scan ...
6 48:AA:99:9F:BE:D8 (unknown)
7 24:CF:91:F8:E8:1B (unknown)
8 C2:E0:B3:FE:8A:E6 (unknown)
9 50:94:0D:43:77:86 (unknown)
10 72:6B:43:77:F4:27 (unknown)
11 5E:C3:95:C1:D2:F1 (unknown)
12 1A:1B:A6:E9:55:90 (unknown)
```

7.2 连接蓝牙设备

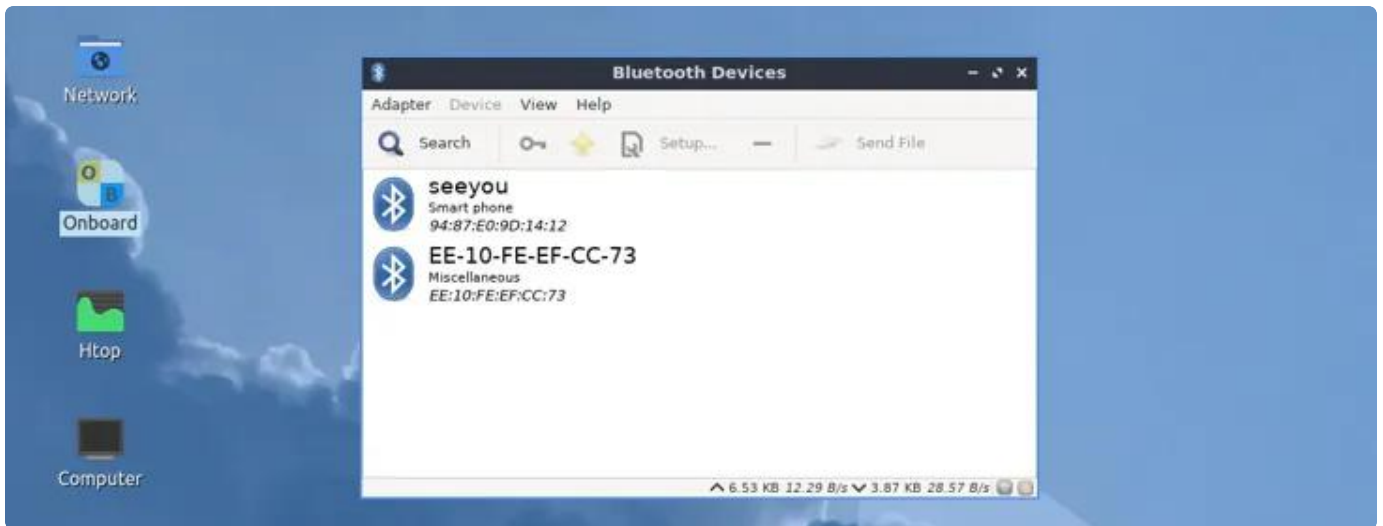
将鼠标放到桌面右下角蓝牙图标，右键->Devices



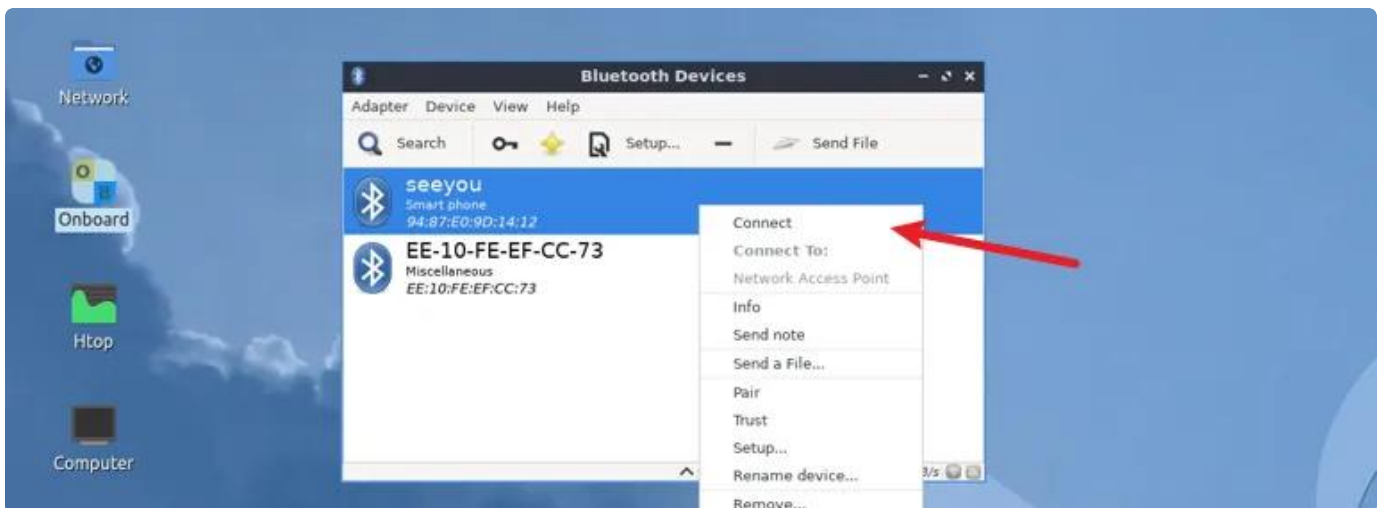
在弹出的窗口中，点击【Search】：



将看到附近的蓝牙设备列表：



选中要连接设备，右键->Connect，即可连接该设备：



8、4G

主板默认适配EC20模块（4G）和 RG200U-CN（5G），上电前，正确按照模块和SIM卡，上电后，系统会自动进行拨号上网。

序号	模块名称	说明
1	EC20	4G LTE
2	RG200U-CN	支持 5G NSA 和 SA 模式，支持 TDD 和 FDD 两种模式，向下兼容 4G/3G。

拨号成功会产生wwan0网络节点：

```
▼ Bash |
1 root@ido:~# ifconfig wwan0
2 wwan0: flags=193<UP,RUNNING,NOARP> mtu 1500
3     inet 10.216.81.159 netmask 255.255.255.192
4     inet6 fe80::2ce5:13ff:fe65:7dd prefixlen 64 scopeid 0x20<link>
5     ether 2e:e5:13:65:07:dd txqueuelen 1000 (Ethernet)
6     RX packets 26 bytes 4777 (4.7 KB)
7     RX errors 0 dropped 0 overruns 0 frame 0
8     TX packets 118 bytes 8278 (8.2 KB)
9     TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
10
11 root@ido:~#
```

使用ping命令测试4G/5G上网功能是否正常：

```
1 root@ido:~# ping www.baidu.com -I wwan0
2 PING www.a.shifen.com (14.215.177.38) from 10.216.81.159 wwan0: 56(84) bytes of data.
3 64 bytes from 14.215.177.38 (14.215.177.38): icmp_seq=1 ttl=49 time=52.2 ms
4 64 bytes from 14.215.177.38 (14.215.177.38): icmp_seq=2 ttl=49 time=52.7 ms
5 64 bytes from 14.215.177.38 (14.215.177.38): icmp_seq=3 ttl=49 time=54.8 ms
6 64 bytes from 14.215.177.38 (14.215.177.38): icmp_seq=4 ttl=49 time=63.2 ms
7 64 bytes from 14.215.177.38 (14.215.177.38): icmp_seq=5 ttl=49 time=50.3 ms
8 64 bytes from 14.215.177.38 (14.215.177.38): icmp_seq=6 ttl=49 time=50.0 ms
9 64 bytes from 14.215.177.38 (14.215.177.38): icmp_seq=7 ttl=49 time=68.3 ms
10 64 bytes from 14.215.177.38 (14.215.177.38): icmp_seq=8 ttl=49 time=61.2 ms
11 64 bytes from 14.215.177.38 (14.215.177.38): icmp_seq=9 ttl=49 time=65.1 ms
```

9、音频

9.1 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 [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-rk817-hifi rk817-hifi-0 [fe410000.i2s-rk817-hifi rk817-hifi-0]
7   Subdevices: 1/1
8   Subdevice #0: subdevice #0
9 root@ido:~#
```

9.2 播放音频

播放到HDMI:

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

播放到Lineout:

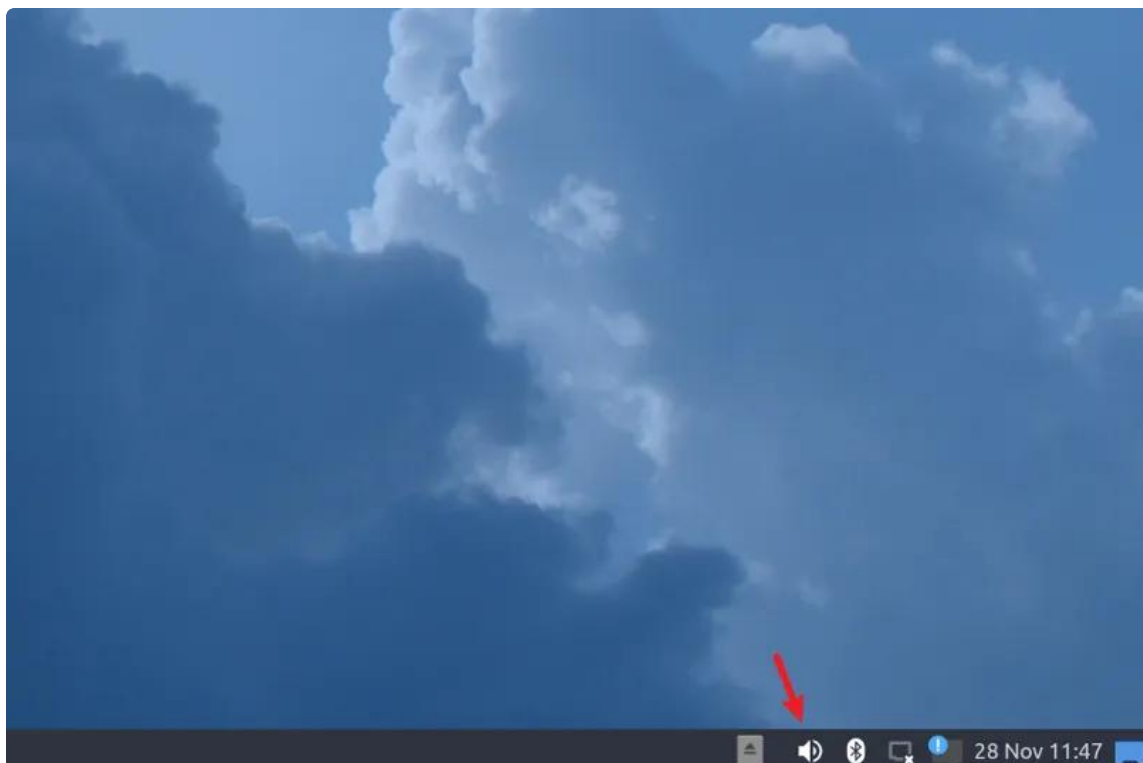
```
1 aplay -D plughw:1,0 /usr/share/sounds/alsa/Rear_Center.wav
```

播放到耳机（需要插入耳机）：

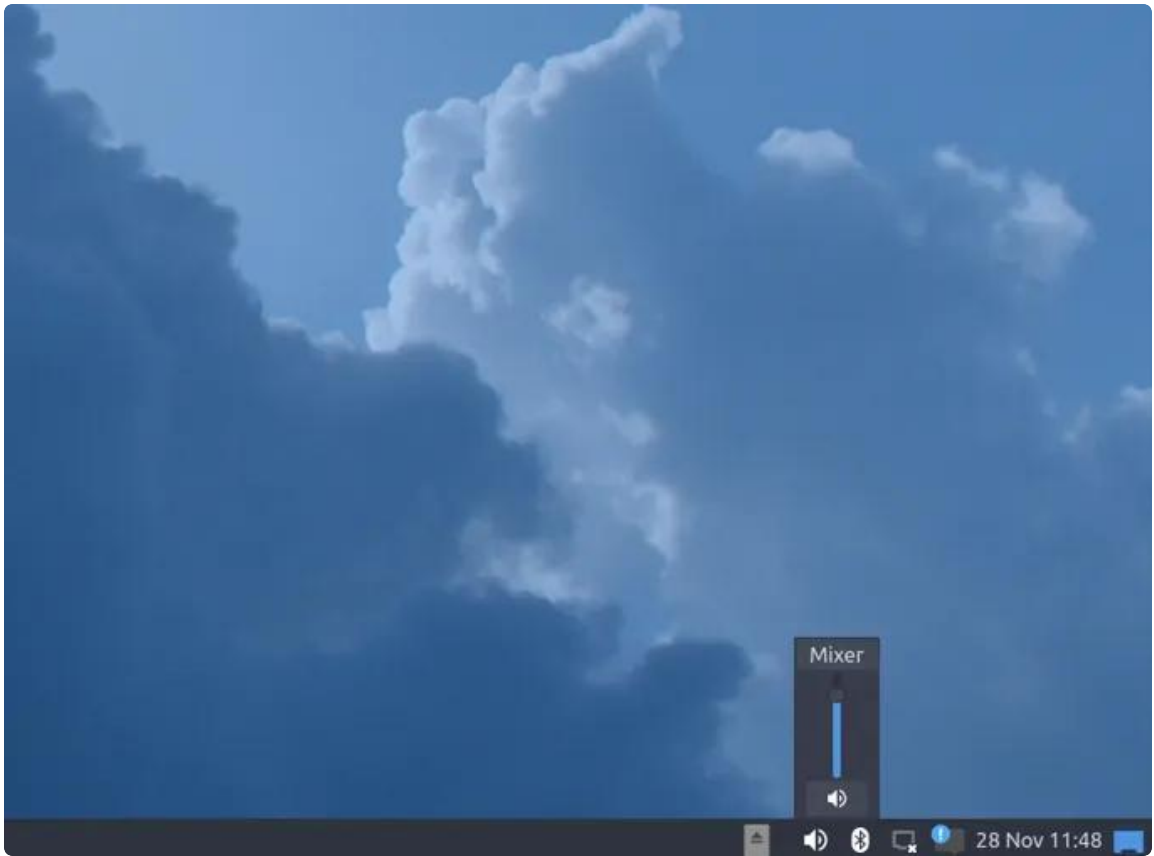
```
1 aplay -D plughw:1,0 /usr/share/sounds/alsa/Rear_Center.wav
```

9.3 音量的调节

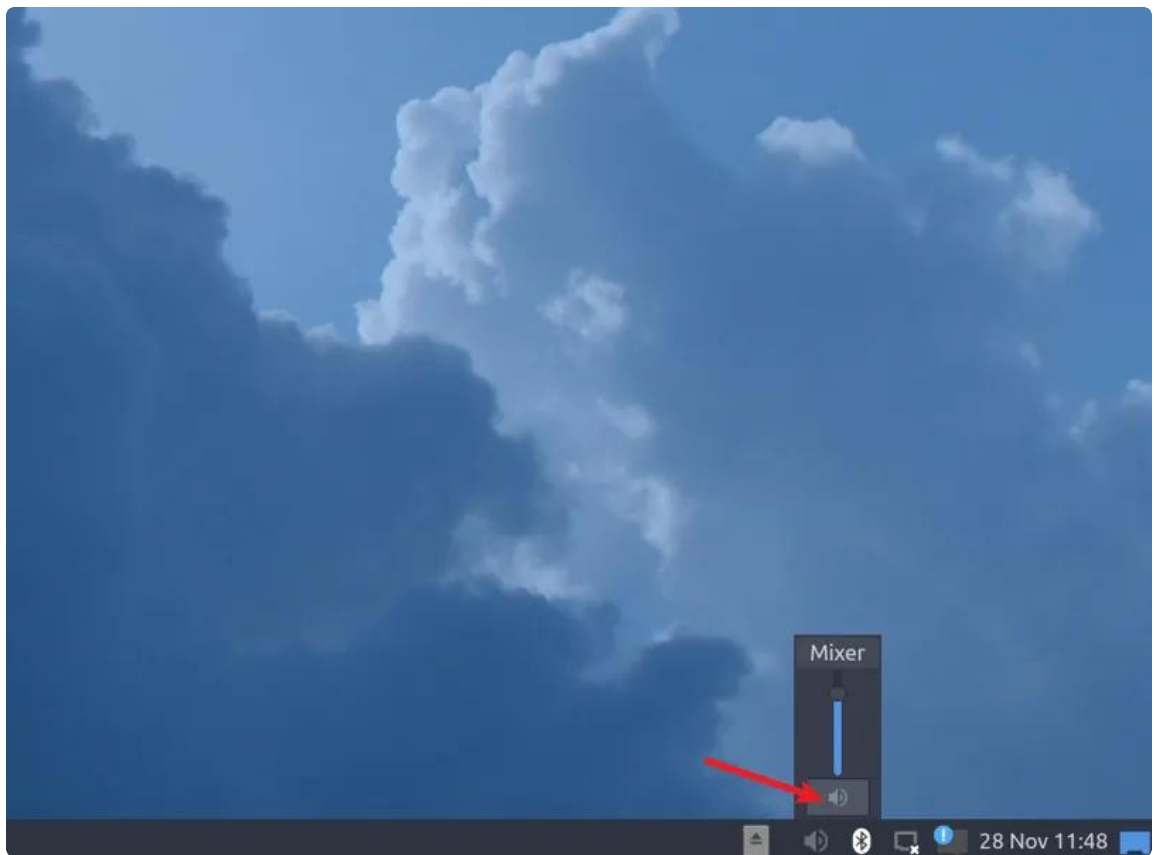
鼠标点击桌面右下角的音量图标：



然后滑动鼠标进行音量调节：



当需要静音是，点击静音按钮：



9.4 录音

将麦克风连接到J121。

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 ^CAborted by signal Interrupt...
4 root@ido:~#
```

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
3 root@ido:~#
```

10、摄像头

开发板默认适配OV5648摄像头。

10.1 测试

10.1.1 测试摄像头是否存在

```
1 root@ido:~# media-ctl -p -d /dev/media0
2 ...
3 - entity 67: rockchip-csi2-dphy0 (2 pads, 2 links)
4     type V4L2 subdev subtype Unknown flags 0
5     device node name /dev/v4l-subdev2
6     pad0: Sink
7         [fmt:SBGGR10_1X10/2592x1944@10000/150000 field:none]
8         <- "m00_b_ov5648 2-0036":0 [ENABLED]
9     pad1: Source
10        [fmt:SBGGR10_1X10/2592x1944@10000/150000 field:none]
11        -> "rkisp-csi-subdev":0 [ENABLED]
12
13 - entity 70: m00_b_ov5648 2-0036 (1 pad, 1 link)
14     type V4L2 subdev subtype Sensor flags 0
15     device node name /dev/v4l-subdev3
16     pad0: Source
17         [fmt:SBGGR10_1X10/2592x1944@10000/150000 field:none]
18         -> "rockchip-csi2-dphy0":0 [ENABLED]
19 root@ido:~#
```

结果显示m00_b_ov5648，说明摄像头存在，最高分辨率支持2592x1944。

10.1.3 抓取视频

使用v4l2-ctl工具可以抓取摄像头的视频数据流。

```

1 root@ido:~# v4l2-ctl --verbose -d /dev/video0 --set-fmt-video=width=1920,height=1080,pixelformat='NV12' --stream-mmap=4 --set-selection=target=crop,flags=0,top=0,left=0,width=1920,height=1080 --stream-to=./out.yuv
2 VIDIOC_QUERYCAP: ok
3 VIDIOC_G_FMT: ok
4 VIDIOC_S_FMT: ok
5 Format Video Capture Multiplanar:
6     Width/Height      : 1920/1080
7     Pixel Format       : 'NV12' (Y/CbCr 4:2:0)
8     Field              : None
9     Number of planes   : 1
10    Flags              :
11    Colorspace          : Default
12    Transfer Function   : Default
13    YCbCr/HSV Encoding : Default
14    Quantization        : Full Range
15    Plane 0             :
16        Bytes per Line : 1920
17        Size Image     : 3110400
18 VIDIOC_G_SELECTION: ok
19 VIDIOC_S_SELECTION: ok
20     VIDIOC_REQBUFS returned 0 (Success)
21     VIDIOC_QUERYBUF returned 0 (Success)
22     VIDIOC_QUERYBUF returned 0 (Success)
23     VIDIOC_QUERYBUF returned 0 (Success)
24     VIDIOC_QUERYBUF returned 0 (Success)
25     VIDIOC_QBUF returned 0 (Success)
26     VIDIOC_QBUF returned 0 (Success)
27     VIDIOC_QBUF returned 0 (Success)
28     VIDIOC_QBUF returned 0 (Success)
29     VIDIOC_STREAMON returned 0 (Success)
30 cap dqbuf: 0 seq:      1 bytesused: 3110400 ts: 1384.549991 (ts-monotonic, ts-src-eof)
31 cap dqbuf: 1 seq:      2 bytesused: 3110400 ts: 1384.616490 delta: 66.499 ms (ts-monotonic, ts-src-eof)
32 cap dqbuf: 2 seq:      3 bytesused: 3110400 ts: 1384.682975 delta: 66.485 ms (ts-monotonic, ts-src-eof)
33 cap dqbuf: 3 seq:      4 bytesused: 3110400 ts: 1384.749486 delta: 66.511 ms (ts-monotonic, ts-src-eof)
34 cap dqbuf: 0 seq:      5 bytesused: 3110400 ts: 1384.816022 delta: 66.536 ms fps: 15.04 (ts-monotonic, ts-src-eof)
35 cap dqbuf: 1 seq:      6 bytesused: 3110400 ts: 1384.882509 delta: 66.487 ms fps: 15.04 (ts-monotonic, ts-src-eof)
36 cap dqbuf: 2 seq:      7 bytesused: 3110400 ts: 1384.949025 delta: 66.516 ms fps: 15.04 (ts-monotonic, ts-src-eof)

```



```

37 cap dqbuf: 3 seq:      8 bytesused: 3110400 ts: 1385.015545 delta: 66.520
   ms fps: 15.04 (ts-monotonic, ts-src-eof)
38 cap dqbuf: 0 seq:      9 bytesused: 3110400 ts: 1385.082051 delta: 66.506
   ms fps: 15.04 (ts-monotonic, ts-src-eof)
39 cap dqbuf: 1 seq:     10 bytesused: 3110400 ts: 1385.148567 delta: 66.516
   ms fps: 15.04 (ts-monotonic, ts-src-eof)
40 cap dqbuf: 2 seq:     11 bytesused: 3110400 ts: 1385.215079 delta: 66.512
   ms fps: 15.04 (ts-monotonic, ts-src-eof)
41 cap dqbuf: 3 seq:     12 bytesused: 3110400 ts: 1385.281594 delta: 66.515
   ms fps: 15.04 (ts-monotonic, ts-src-eof)
42 cap dqbuf: 0 seq:     13 bytesused: 3110400 ts: 1385.348115 delta: 66.521
   ms fps: 15.04 (ts-monotonic, ts-src-eof)
43 cap dqbuf: 1 seq:     14 bytesused: 3110400 ts: 1385.414669 delta: 66.554
   ms fps: 15.03 (ts-monotonic, ts-src-eof)
44 cap dqbuf: 2 seq:     15 bytesused: 3110400 ts: 1385.481133 delta: 66.464
   ms fps: 15.04 (ts-monotonic, ts-src-eof)
45 cap dqbuf: 3 seq:     16 bytesused: 3110400 ts: 1385.547656 delta: 66.523
   ms fps: 15.04 (ts-monotonic, ts-src-eof)
46 cap dqbuf: 0 seq:     17 bytesused: 3110400 ts: 1385.614172 delta: 66.516
   ms fps: 15.04 (ts-monotonic, ts-src-eof)
47 cap dqbuf: 1 seq:     18 bytesused: 3110400 ts: 1385.680680 delta: 66.508
   ms fps: 15.04 (ts-monotonic, ts-src-eof)
48 cap dqbuf: 2 seq:     19 bytesused: 3110400 ts: 1385.747241 delta: 66.561
   ms fps: 15.03 (ts-monotonic, ts-src-eof)
49 cap dqbuf: 3 seq:     20 bytesused: 3110400 ts: 1385.813714 delta: 66.473
   ms fps: 15.03 (ts-monotonic, ts-src-eof)
50 ^C

```

按Ctrl-C停止抓取，视频流保存到文件out.yuv。

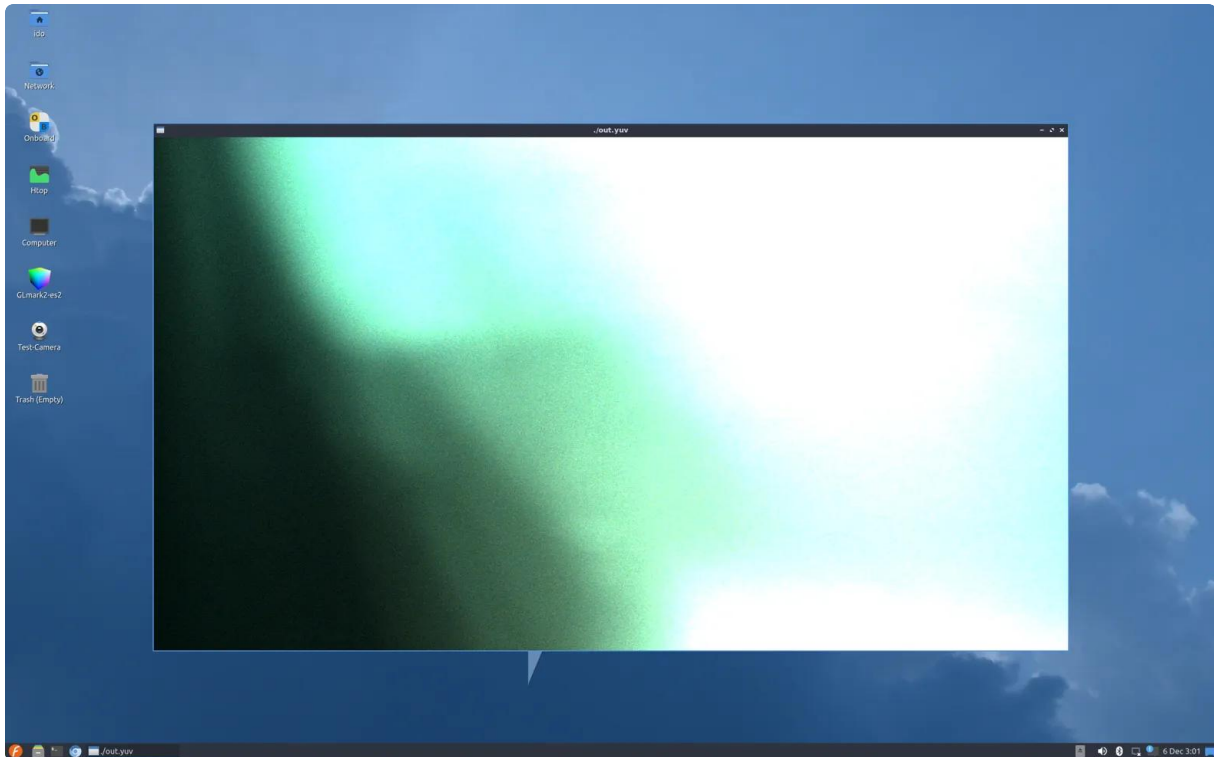
使用ffplay工具播放抓取的视频流：

```

1  root@ido:~# ffplay -f rawvideo -video_size 1920x1080 -pix_fmt nv12 ./out.yuv
2  ffplay version 4.2.4-1ubuntu1.0firefly1 Copyright (c) 2003-2020 the FFmpeg
   developers
3  built with gcc 9 (Ubuntu 9.3.0-17ubuntu1~20.04)
4  configuration: --prefix=/usr --extra-version=1ubuntu1.0firefly1 --toolchain=hardened --libdir=/usr/lib/aarch64-linux-gnu --incdir=/usr/include/aarch64-linux-gnu --arch=arm64 --enable-gpl --disable-stripping --enable-avresample --disable-filter=resample --enable-avisynth --enable-gnutls --enable-ladspa --enable-libaom --enable-libass --enable-libbluray --enable-libbs2b --enable-libcaca --enable-libcdio --enable-libcodec2 --enable-libflite --enable-libfontconfig --enable-libfreetype --enable-libfribidi --enable-libgme --enable-libgsm --enable-libjack --enable-libmp3lame --enable-libmysofa --enable-libopenjpeg --enable-libopenmpt --enable-libopus --enable-libpulse --enable-librsvg --enable-librubberband --enable-libshine --enable-libsnappy --enable-libsoxr --enable-lsbspeex --enable-libssh --enable-libtheora --enable-libtwolame --enable-libvidstab --enable-libvorbis --enable-libvpx --enable-libwavpack --enable-libwebp --enable-libx265 --enable-libx264 --enable-libxvid --enable-libzmq --enable-libzvbi --enable-lv2 --enable-omx --enable-opengl --enable-opencore --enable-openssl --enable-sdl2 --enable-libdc1394 --enable-libdrm --enable-libiec61883 --enable-chromaprint --enable-frei0r --enable-libx264 --enable-libdrm --enable-librga --enable-rkmp --enable-version3 --disable-libopenh264 --disable-vaapi --disable-vdpau --disable-decoder=h264_v4l2m2m --disable-decoder=vp8_v4l2m2m --disable-decoder=mpeg2_v4l2m2m --disable-decoder=mpeg4_v4l2m2m --disable-muxer='ac3,eac3,mlp,truehd' --disable-encoder='ac3_fixed,ac3,mlp,spdif,truehd' --disable-demuxer='ac3,eac3,mlp,truehd,dts,dtsd' --disable-parser='aac,ac3,mlp' --disable-decoder='ac3,eac3,mlp,dolby_e' --enable-shared --disable-doc
5  libavutil      56. 31.100 / 56. 31.100
6  libavcodec     58. 54.100 / 58. 54.100
7  libavformat    58. 29.100 / 58. 29.100
8  libavdevice    58.  8.100 / 58.  8.100
9  libavfilter     7. 57.100 /  7. 57.100
10 libavresample   4.  0.  0 /  4.  0.  0
11 libswscale      5.  5.100 /  5.  5.100
12 libswresample   3.  5.100 /  3.  5.100
13 libpostproc    55.  5.100 / 55.  5.100
14 Option -pix_fmt is deprecated, use -pixel_format.
15 libGL error: failed to create dri screen
16 libGL error: failed to load driver: rockchip
17 libGL error: failed to create dri screen
18 libGL error: failed to load driver: rockchip
19 [rawvideo @ 0x7f3c000ba0] Estimating duration from bitrate, this may be inaccurate
20 Input #0, rawvideo, from './out.yuv':

```

```
21 Duration: 00:00:04.00, start: 0.000000, bitrate: 622075 kb/s
22 Stream #0:0: Video: rawvideo (NV12 / 0x3231564E), nv12, 1920x1080, 622
23 080 kb/s, 25 tbr, 25 tbn, 25 tbc
```



11、RTC

SBC3566包含2个RTC，其中/dev/rtc0为外部RTC（HYM8563），/dev/rtc1为CPU内部的RTC（RK808）。系统默认使用rtc0的时间。

11.1 获取RTC时间

```
▼ Bash |
1 root@ido:~# hwclock
2 2022-11-10 02:16:23.617474+00:00
```

11.2 设置RTC时间

```

1 root@ido:~# hwclock
2 2022-11-18 08:30:40.778874+00:00
3 root@ido:~# date -s '2022-11-10 10:17:00'
4 Thu Nov 10 10:17:00 UTC 2022
5 root@ido:~# hwclock -w
6 root@ido:~# hwclock
7 2022-11-18 08:31:06.829691+00:00
8 root@ido:~#

```

11.3 RTC定时开机

CPU内部的rtc1支持支持定时开机功能。

以下测试关机60秒定时开机：

```

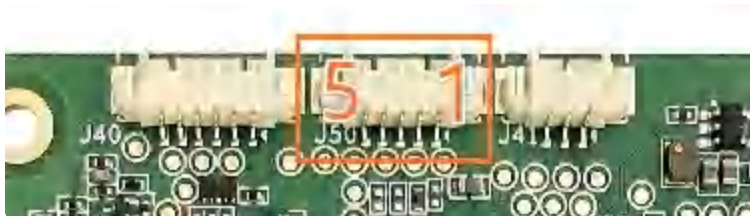
1 root@ido:~# echo +60 > /sys/class/rtc/rtc1/wakealarm
2 root@ido:~# poweroff
3     Stopping Session c1 of user ido.
4     Stopping Session c2 of user root.
5 [ OK ] Removed slice system-modprobe.slice.
6 [ OK ] Stopped target Bluetooth.
7 ...
8 [ OK ] Reached target Shutdown.
9 [ OK ] Reached target Final Step.
10 [ OK ] Finished Power-Off.
11 [ OK ] Reached target Power-Off.
12 [ 1195.829009] Local MAC: 82:4c:21:62:f5:35
13 [ 1195.834186] set rk809 rtc alarm!
14 [ 1195.837515] reboot: Power down
15

```

等待60秒后，主板会开机。

12、IR红外功能

SBC3566支持一路IR红外接口，位于J50。



序号	定义	电平	说明
1	VDD5V0_SYS	5V	供电输出5V
2	GND	GND	电源地
3	IR_INTER	/	红外信号输入
4	IR_LED1	2.3V	LED控制信号输出
5	IR_LED2	2.3V	LED控制信号输出

系统默认适配的红外遥控器型号为HTR-A07:



13、开机自启动

默认系统开机运行/etc/rc.local脚本，将要开机执行的程序放到该脚本中即可。

以设置开机运行/usr/sbin/app为例。

编辑/etc/rc.local（如果不存在则创建，且赋予可执行权限）：

```
1  #!/bin/sh
2  export PATH="${PATH:+$PATH:}/usr/sbin:/sbin"
3
4  case "$1" in
5      start)
6          /usr/sbin/app
7      ;;
8  esac
9
10 exit 0
```

这样每次开机后，/usr/sbin/app就会执行。

14、屏幕控制

14.1 背光调节

通过修改/sys/class/backlight/backlight/brightness的值，实现背光的调节，范围取0–255，值越大，亮度越高。

设置亮度为100：

```
1  root@ido:~# echo 100 > /sys/class/backlight/backlight/brightness
2  root@ido:~#
```

14.2 屏幕旋转

使用xrandr工具可以实现屏幕的旋转。

14.2.1 临时旋转

系统启动后，执行xrandr -o normal,inverted,left,right，可以实现临时旋转屏幕方向，其中normal表示顺时针旋转0度，inverted表示顺时针旋转180度，left表示顺时针旋转270度，right表示顺时针旋转90度。

▼

Bash

```
1 root@ido:~# xrandr -o inverted
```

14.2.2 永久旋转

修改/etc/default/xrandr启动文件，可以实现永久旋转。
以旋转180度为例：

▼

Bash

```
1 root@ido:~# cat /etc/default/xrandr
2 #!/bin/sh
3     /usr/bin/xrandr -o inverted
4
5 root@ido:~#
```

这样修改后，每次重启设备，桌面将旋转180度。

15、按键

SBC3566配置了2个ADC按键，对应的设备节点为/dev/input/event3。

编号	功能	上报键值
1	烧录按键、音量加键	KEY_VOLUMEUP
2	电源键	KEY_POWER

16、ADC

SBC3566配置了2路ADC，位于J40的第4，5引脚，分别记作ADC1和ADC2。精度为10位。

16.1 ADC转换方法

$$V = (raw/1024)*1.8v$$

其中raw为对应设备节点读取的值，范围为0-1023。

编号	设备节点
ADC1	/sys/bus/iio/devices/iio:device0/in_voltage3_raw
ADC2	/sys/bus/iio/devices/iio:device0/in_voltage2_raw

16.2 测试

以测试ADC1为例，ADC2测试方法类似。

▼

Bash

```
1 root@ido:~# cat /sys/bus/iio/devices/iio:device0/in_voltage3_raw
2 997
```

设备节点读取的raw值为997，代入到公式计算：

$$V=(997/1024)*1.8v=1.75v$$

即ADC1输入的电压为1.75v。

17、网络优先级设置

SBC3566支持以太网、WiFi和4G/5G三种网络，通过路由表来设置它们的网络优先级。

17.1 查看路由表

▼

Bash

```
1 root@ido:~# route
2 Kernel IP routing table
3 Destination      Gateway           Genmask          Flags Metric Ref    Use Ifa
4 default          _gateway         0.0.0.0          UG    100   0      0 eth
5 default          _gateway         0.0.0.0          UG    600   0      0 wla
6 192.168.1.0      0.0.0.0          255.255.255.0    U     100   0      0 eth
7 192.168.1.0      0.0.0.0          255.255.255.0    U     600   0      0 wla
```


17.2 设置默认路由

17.2.1 设置WiFi为默认路由

当前以太网和WiFi同时使用，设置WiFi优先：

▼

Bash

```
1 root@ido:~# route
2 Kernel IP routing table
3 Destination      Gateway            Genmask           Flags Metric Ref    Use If
4 default          _gateway          0.0.0.0           UG      100    0      0 et
5 default          _gateway          0.0.0.0           UG      600    0      0 wl
6 192.168.1.0      0.0.0.0           255.255.255.0    U       100    0      0 et
7 192.168.1.0      0.0.0.0           255.255.255.0    U       600    0      0 wl
8 root@ido:~# route del default dev eth0
9 root@ido:~# route
10 Kernel IP routing table
11 Destination      Gateway            Genmask           Flags Metric Ref    Use If
12 default          _gateway          0.0.0.0           UG      600    0      0 wl
13 192.168.1.0      0.0.0.0           255.255.255.0    U       100    0      0 et
14 192.168.1.0      0.0.0.0           255.255.255.0    U       600    0      0 wl
```

这样默认路由就是wlan0了，即优先使用WiFi进行数据通信。

17.2.2 设置以太网为默认路由

当前以太网和WiFi同时使用，且WiFi优先：

▼

Bash |

```
1 root@ido:~# route
2 Kernel IP routing table
3 Destination      Gateway            Genmask           Flags Metric Ref    Use Ifa
4 default          _gateway          0.0.0.0           UG        600    0      0 wlan0
5 192.168.1.0      0.0.0.0           255.255.255.0    U         100    0      0 eth0
6 192.168.1.0      0.0.0.0           255.255.255.0    U         600    0      0 wlan0
7 root@ido:~#
```

设置为以太网优先：

▼

Bash |

```
1 root@ido:~# route del default dev wlan0
2 root@ido:~# route add default dev eth0
3 root@ido:~# route add default gw 192.168.1.1
4 root@ido:~# route
5 Kernel IP routing table
6 Destination      Gateway            Genmask           Flags Metric Ref    Use If
7 default          0.0.0.0           0.0.0.0           U         0      0      0 eth
8 192.168.1.0      0.0.0.0           255.255.255.0    U         100    0      0 et
9 192.168.1.0      0.0.0.0           255.255.255.0    U         600    0      0 wl
10 root@ido:~#
```

其他情况按照类似的方法进行处理即可。