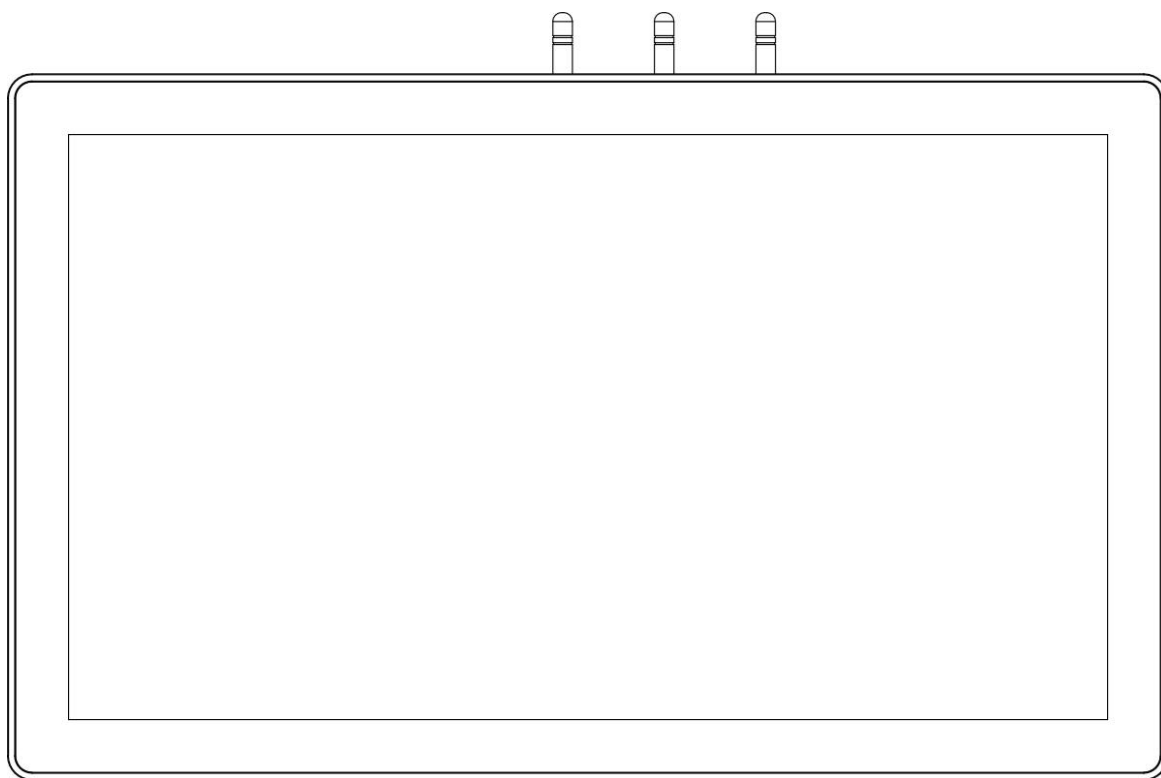




ED-HMI3120-185C

用户手册

by EDA Technology Co., Ltd

built: 2026-09-17

1 硬件手册

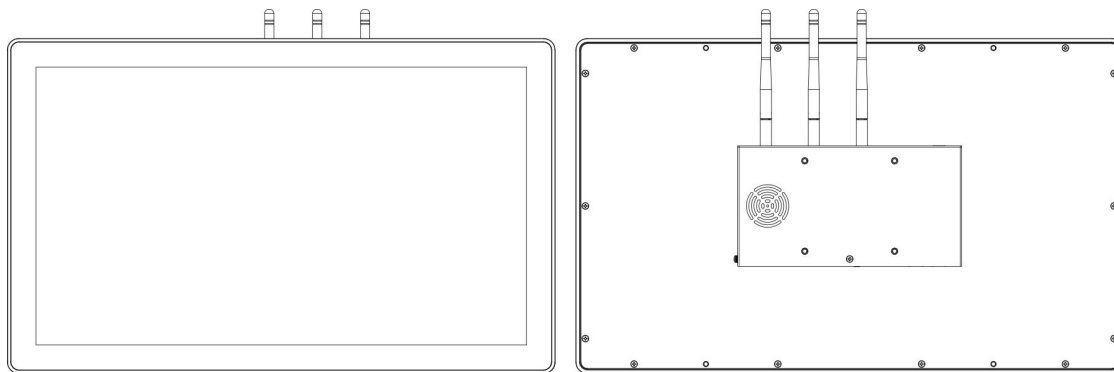
本章介绍产品概述、包装清单、外观、按键、指示灯和接口等。

1.1 产品概述

ED-HMI3120-185C是一款基于Raspberry Pi CM5的18.5英寸工业平板电脑。根据不同的应用场景和用户需求，可选择不同规格的RAM和eMMC的计算机系统。

- RAM可选规格包含2GB、4GB、8GB和16GB。
- eMMC可选规格包含16GB、32GB和64GB。

ED-HMI3120-185C提供HDMI、USB、Ethernet、RS232和RS485等常用的接口，且支持通过Wi-Fi、以太网和4G接入网络；集成超级电容备份电源(选配)、RTC、EEPROM和加密芯片，提升了产品的易用性和可靠性，主要应用于工业控制和物联网领域。



1.2 包装清单

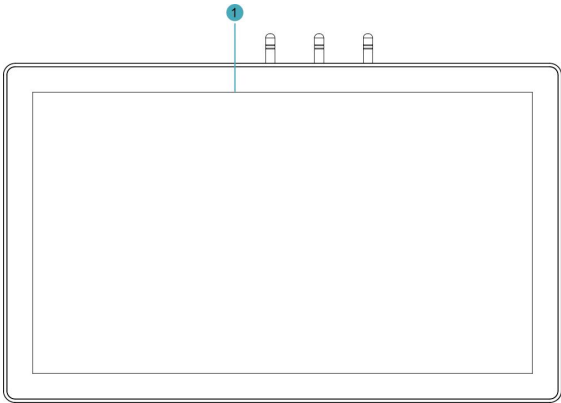
- 1 x ED-HMI3120-185C主机
- 1 x 配件包 (包含4x卡扣、4xM4*10螺钉和4xM4*16螺钉)
- [选配Wi-Fi/BT版本] 1x 2.4GHz/5GHz Wi-Fi/BT天线
- [选配4G版本] 1x 4G/LTE天线
- 1 x 2-Pin带螺丝的凤凰端子
- 2 x 6-Pin的凤凰端子

1.3 产品外观

介绍各面板上接口的功能和定义。

1.3.1 前面板

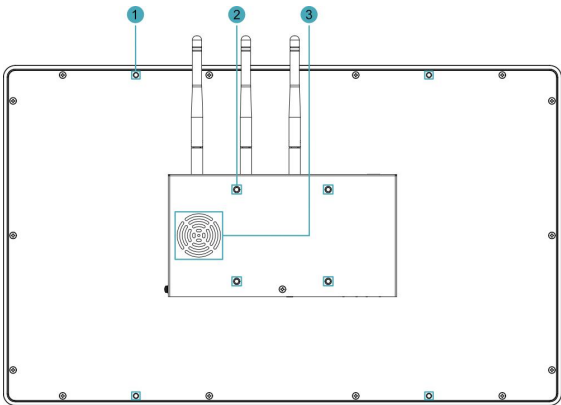
介绍前面板接口类型和定义。



编号	功能定义
1	1 x LCD屏，18.5英寸带触摸屏，分辨率支持1920x1080，多点式电容触摸屏。

1.3.2 后面板

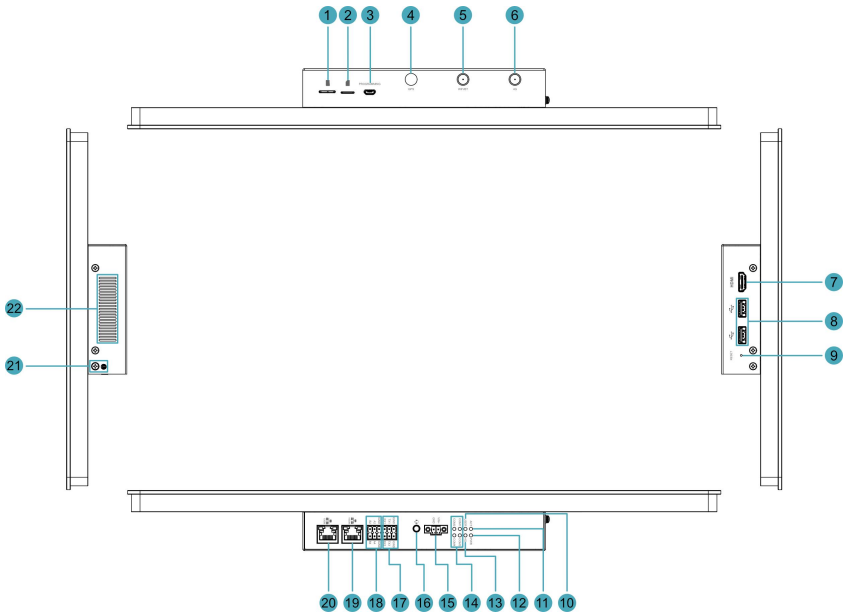
介绍后面板接口类型和定义。



编号	功能定义
1	4 x 卡扣安装孔，用于将卡扣固定在设备上进行嵌入式安装。
2	4 x VESA安装孔，预留进行VESA支架安装。
3	1 x Speaker，内置1个4Ω 5W的喇叭，单通道立体声输出。

1.3.3 侧面板

介绍侧面板接口类型和定义。



编号	功能定义
1	1 x Micro SD卡槽，用于安装Micro SD卡，支持存储用户数据。
2	1 x Nano SIM卡槽，用于安装获取4G信号的Nano SIM卡。
3	1 x Micro USB接口，支持通过此接口对系统进行eMMC烧录。
4	1 x 橡胶塞，此处仅作为预留孔。
5	1 x Wi-Fi/BT天线接口(选配)，SMA接口，连接Wi-Fi/BT天线。
6	1 x 4G天线接口(选配)，SMA接口，连接4G天线。
7	1 x HDMI，Type-A接口，兼容HDMI 2.0，分辨率支持4K 60Hz，支持连接显示器。
8	2 x USB 2.0，Type-A接口，每一路最高支持480Mbps传输速率。
9	1 x 复位按键，隐藏式按键，按下按键可使设备复位。
10	1 x 用户指示灯，绿色，用户可以根据实际需求自定义状态。
11	1 x 系统状态指示灯，绿色，用于查看系统读写数据的状态。
12	1 x 电源指示灯，红色，用于查看设备上电状态。
13	1 x 4G信号指示灯，绿色，用于查看4G信号的状态。
14	4 x 串口指示灯，绿色，用于查看串口的通信状态。
15	1 x DC输入，2-Pin 3.5mm间距带螺丝孔的凤凰端子，支持12V~24V输入。
16	1 x 音频输入/立体声输出，3.5mm音频接口，可作为麦克风输入和立体声输出。 • 当接入耳机时，音频输出切换至耳机； • 当未接入耳机时，音频输出切换至Speaker。
17	2 x RS232接口，6-Pin 3.5mm间距凤凰端子，用于连接第三方控制设备。
18	2 x RS485接口，6-Pin 3.5mm间距凤凰端子，用于连接第三方控制设备。

编号	功能定义
19	1 x 1000M以太网接口，RJ45接口，带有LED灯，10/100/1000M自适应接口，用于接入以太网。
20	1 x 100M以太网接口，RJ45接口，带有LED灯，10/100M自适应接口，用于接入以太网。
21	1 x 接地点，连接外部电源的大地。
22	散热孔，便于外壳内空气流动和散热。

1.4 按键

ED-HMI3120-185C设备包含1个RESET按键，该按键为隐藏式按键，在外壳上的丝印为“RESET”，按下RESET按键使设备复位。

1.5 指示灯

介绍ED-HMI3120-185C设备包含的指示灯的各种状态及含义。

指示灯	状态	描述
PWR	常亮	设备已上电
	闪烁	设备电源异常，立即停止供电
	熄灭	设备未上电
ACT	闪烁	系统启动成功且正在读写数据
	熄灭	设备未上电或未读写数据
USER	常亮	用户自定义
	熄灭	设备未上电或用户未定义，默认状态为熄灭
4G	常亮	拨号成功，连接正常
	熄灭	4G信号未连接或设备未上电
以太网口绿色指示灯（100M）	常亮	数据传输异常
	闪烁	正在传输以太网数据
	熄灭	未接入以太网
以太网口黄色指示灯（100M）	常亮	已正常接入以太网
	闪烁	以太网连接异常
	熄灭	未接入以太网
以太网口绿色指示灯（1000M）	常亮	已正常接入以太网
	闪烁	以太网连接异常

指示灯	状态	描述
以太网口黄色指示灯（1000M）	熄灭	未接入以太网
	常亮	数据传输异常
	闪烁	正在传输以太网数据
	熄灭	未接入以太网
COM1~COM4	常亮/闪烁	正在传输数据
	熄灭	设备未上电或无数据传输

1.6 接口

介绍产品中各接口的定义和功能。

1.6.1 卡槽

ED-HMI3120-185C设备包含1个Micro SD卡槽和1个Nano SIM卡槽。

1.6.1.1 Micro SD 卡槽

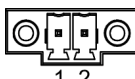
Micro SD卡槽接口丝印为“”，支持安装Micro SD卡，用于存储用户数据。

1.6.1.2 SIM卡槽（选配）

Nano SIM卡槽接口丝印为“”，用于安装获取4G信号的Nano SIM卡。

1.6.2 电源接口

ED-HMI3120-185C设备包含1路电源输入，2-Pin 3.5mm间距的凤凰端子，接口丝印为“VIN+/
GND”，引脚定义如下。

	Pin ID	Pin Name
	1	GND
	2	12V~24V

1.6.3 音频接口

ED-HMI3120-185C设备包含1路音频输入，3.5mm四段式耳机接口，接口丝印为“”，支持OMTP规格立体声输出和单通道麦克风录音。

- 当耳机接入时，音频通道切换至耳机输出；
- 当耳机未接入时，音频通道切换至Speaker输出。

1.6.4 Speaker

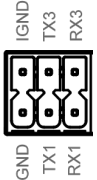
ED-HMI3120-185C设备包含1路功放输出，内置1个4Ω 5W的喇叭，单通道立体声输出。播放音频时，如果耳机接入音频接口，则Speaker无音频输出。

1.6.5 RS232接口

ED-HMI3120-185C设备包含2路RS232接口，6-Pin 3.5mm间距的凤凰端子，单路RS232接口丝印为“IGND/TX/RX”。

引脚定义

端子引脚定义如下：

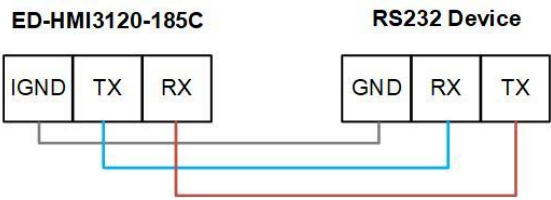
	Pin ID	Pin Name
	1	GND
	2	GND
	3	RS232-1_TX
	4	RS232-3_TX
	5	RS232-1_RX
	6	RS232-3_RX

其中RS232接口对应CM5的管脚名称如下：

Signal	CM5 GPIO Name	CM5 Pin Out
RS232-1_TX	GPIO4	UART3_TXD
RS232-3_TX	GPIO0	UART2_TXD
RS232-1_RX	GPIO5	UART3_RXD
RS232-3_RX	GPIO1	UART2_RXD

连接线缆

RS232串口的接线示意图如下：



1.6.6 RS485接口

ED-HMI3120-185C设备包含2路RS485接口，6-Pin 3.5mm间距的凤凰端子，单路RS485接口丝印为“IGND/A/B”。

引脚定义

端子引脚定义如下：

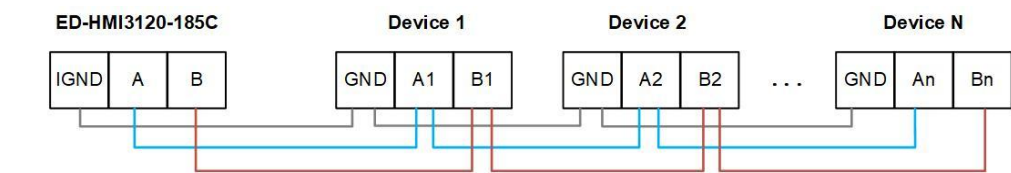
	Pin ID	Pin Name
	1	GND
	2	GND
	3	RS485-2_A
	4	RS485-4_A
	5	RS485-2_B
	6	RS485-4_B

其中RS485接口对应CM5的管脚名称如下：

Signal	CM5 GPIO Name	CM5 Pin Out
RS485-2_A	GPIO12	UART5_TXD
RS485-4_A	GPIO8	UART4_TXD
RS485-2_B	GPIO13	UART5_RXD
RS485-4_B	GPIO9	UART4_RXD

连接线缆

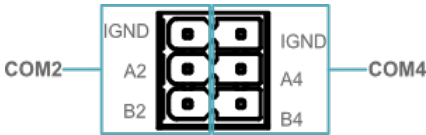
RS485串口的接线示意图如下：



RS485端电阻配置

ED-HMI3120-185C设备包含2路RS485接口，在RS485线路的A和B之间预留120R跨接电阻，插入跳线帽可使能该跨接电阻。默认状态下未连接跳线帽，120R端接电阻功能失效。跨接电阻在PCBA中的位置为J24和J22。

120R终端电阻与串口的对应关系如下：

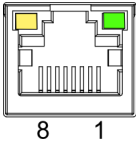
PCBA中的位置	对应的COM口	对应COM的具体位置
J24	COM4	
J22	COM2	

提示

需要打开设备外壳才能查看120R跨接电阻的位置。

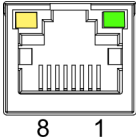
1.6.7 1000M以太网接口

ED-HMI3120-185C设备包含1路自适应10/100/1000M以太网接口，接口丝印为“”。使用RJ45端子，接入以太网时建议采用Cat6及以上规格的网线配合使用。端子对应的引脚定义如下：

	Pin ID	Pin Name
	1	TX1+
	2	TX1-
	3	TX2+
	4	TX2-
	5	TX3+
	6	TX3-
	7	TX4+
	8	TX4-

1.6.8 100M以太网接口

ED-HMI3120-185C 设备包含1路自适应10/100M以太网接口，接口丝印为“”。使用RJ45端子，接入以太网时建议采用Cat6及以上规格的网线配合使用。端子对应的引脚定义如下：

	Pin ID	Pin Name
	1	TX+
	2	TX-
	3	Rx+
	4	-
	5	-
	6	RX-

	7	-
	8	-

1.6.9 HDMI接口

ED-HMI3120-185C 设备包含1路HDMI接口，接口丝印为“HDMI”，标准的Type-A接口。支持连接HDMI显示器，最大支持4Kp60的视频输出。

1.6.10 USB 2.0接口

ED-HMI3120-185C设备包含2路USB 2.0接口，接口丝印为“”，标准的Type-A接口。支持连接标准的USB 2.0外设，最大支持480Mbps的传输速率。

1.6.11 Micro USB接口

ED-HMI3120-185C 设备包含1路Micro USB接口，接口丝印为“PROGRAMMING”，支持通过连接PC对设备的eMMC进行烧录。

1.6.12 天线接口（选配）

ED-HMI3120-185C设备包含2路SMA天线接口，接口丝印分别为“4G”和“WiFi/BT”，分别对应连接4G天线和Wi-Fi/BT天线。

提示

天线接口的数量 and 用户实际选配的型号相关，此处仅以包含2路天线接口为例进行说明。

1.7 超级电容（选配）

ED-HMI3120-185C支持选配超级电容备份电源模块，具备完善的安全关机保护机制。

- 当设备在超级电容已满电的情况下意外断电，系统会自动关闭屏幕背光并降低CPU频率以节能，设备持续处于低功耗运行状态，直至电容电量耗尽后自动安全关机，有效避免运行中断与数据丢失。
- 设备提供用户可自定义的应急脚本，支持预先配置关键数据的保存、备份或其他应急处理流程。在断电瞬间，系统将自动执行预设脚本，确保重要操作得以完成，从容应对突发断电情况。

提示

- 自定义超级电容相关参数的具体操作，请参见[AN18 超级电容使用指南](#)。
- 超级电容需要在设备上电至少五分钟后才能充满电，充满电后才能保证功能的正常使用。

2 安装部件（可选）

本章介绍安装可选部件的具体操作。

2.1 安装天线

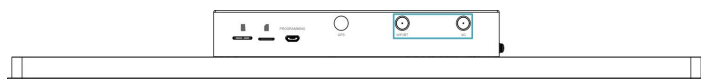
若选购的ED-HMI3120-185C设备包含4G和Wi-Fi功能，在使用设备之前需要先安装天线。

前提条件：

已从包装盒中获取对应的天线，若包含多根天线，通过天线上的标签来区分。

操作步骤：

1. 确定设备侧天线接口的位置，如下图标注位置所示。



2. 对准设备和天线两侧的接口，沿顺时针方向拧紧天线，确保天线不会脱落即可。

2.2 安装Micro SD卡

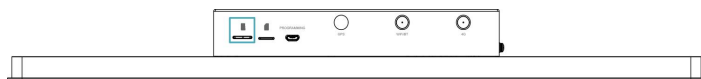
如果在产品使用中需要安装Micro SD卡，则可以参考下文来操作。

前提条件：

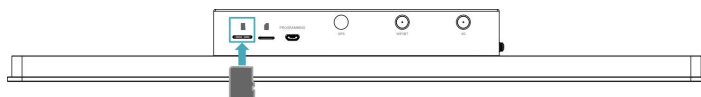
- 已获取待使用的Micro SD卡。

操作步骤：

1. 确定设备侧Micro SD卡槽的位置，如下图标注位置所示。



2. 将Micro SD卡触点面朝下插入对应的卡槽，听到一声响表示安装完成。



2.3 安装Nano SIM卡

若选购的ED-HMI3120-185C设备包含4G功能，则在使用4G功能前需要先安装Nano SIM卡。

注意

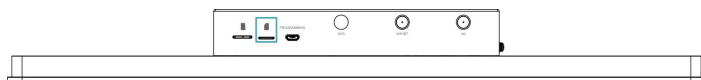
Nano SIM卡不支持带电插拔。

前提条件：

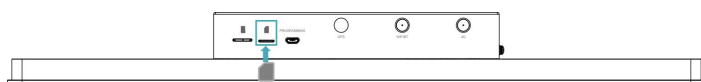
- 已获取待使用的4G Nano SIM卡。
- 已断开设备电源。

操作步骤：

1. 确定设备侧Nano SIM卡槽的位置，如下图标注位置所示。



2. 将Nano SIM卡触点面朝下插入对应的卡槽，听到一声响表示安装完成。



3 安装设备

ED-HMI3120-185C设备支持嵌入式前安装，标准包装中包含嵌入式安装配件包（ED-ACCHMI-Front）。

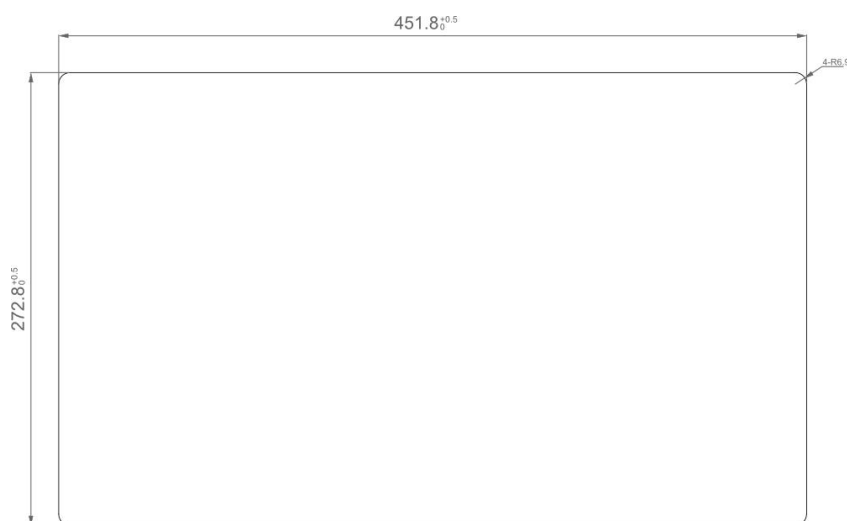
前提条件：

- 已获取ED-ACCHMI-Front配件包（包含4颗M4*10螺钉、4颗M4*16螺钉和4个卡扣）。
- 已准备一把十字螺丝刀。

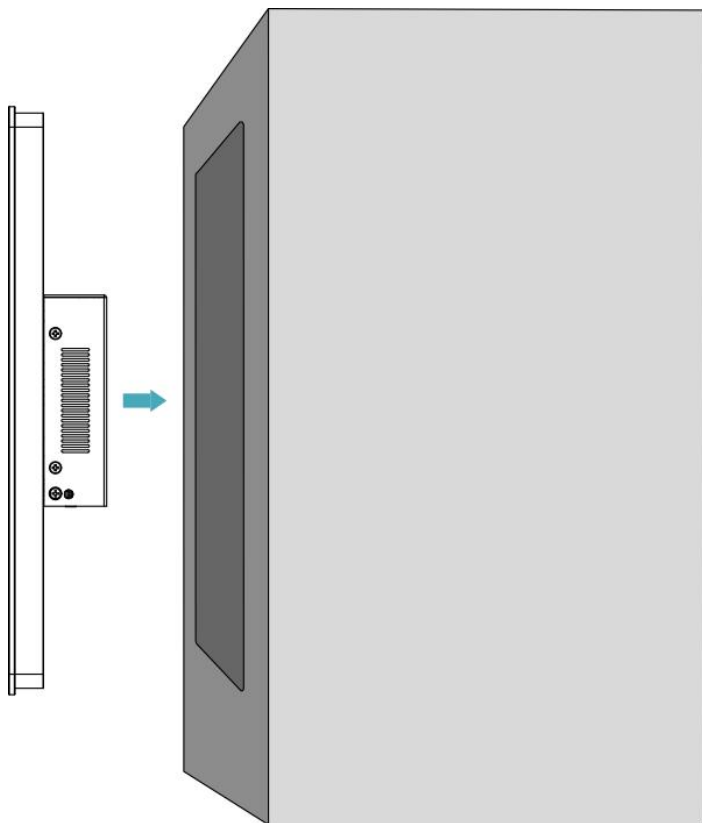
操作步骤：

1. 根据ED-HMI3120-185C的尺寸，确定机柜的开孔尺寸，如下图。

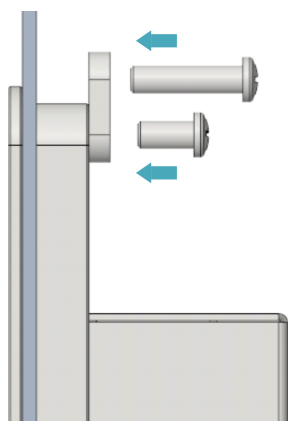
单位：mm



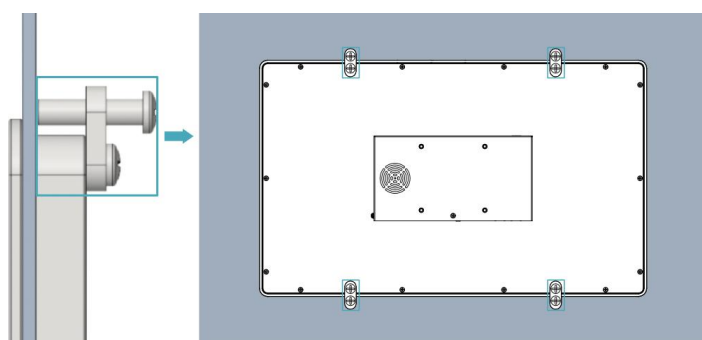
2. 根据步骤1的孔径大小在机柜上钻孔。
3. 将ED-HMI3120-185C从外部嵌入到机柜中。



4. 将卡扣的螺丝孔（不带螺纹的孔）对准设备侧的卡扣安装孔。



5. 将4颗M4*10螺钉穿过卡扣的螺丝孔（不带螺纹的孔）并使用螺丝刀顺时针拧紧，将卡扣固定至设备上；再将4颗M4*16螺钉穿过卡扣的螺丝孔（带螺纹的孔）顶住机柜内侧并使用螺丝刀顺时针拧紧到底。



4 启动设备

本章介绍连接线缆和启动设备的具体操作。

4.1 连接线缆

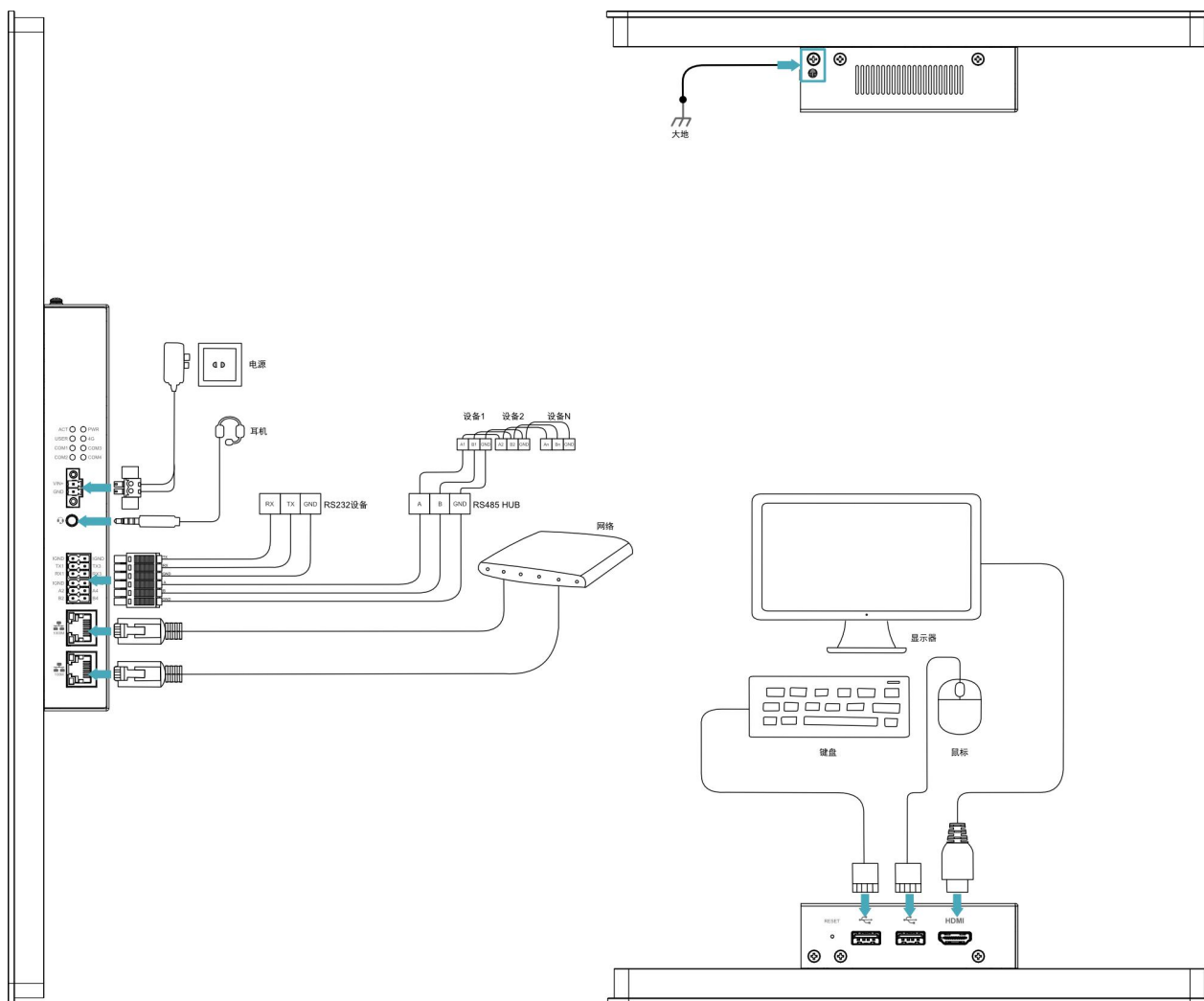
介绍线缆的连接方法。

前提条件：

- 已获取可以正常使用的显示器、鼠标、键盘和电源适配器等配件。
- 已获取可以正常使用的网络。
- 已获取可以正常使用的HDMI线和网线。

连接线缆示意图：

各接口的引脚定义以及连线的具体方法，请参见1.6接口。



4.2 首次启动系统

ED-HMI3120-185C设备无电源开关，接入电源后，系统将会开始启动。

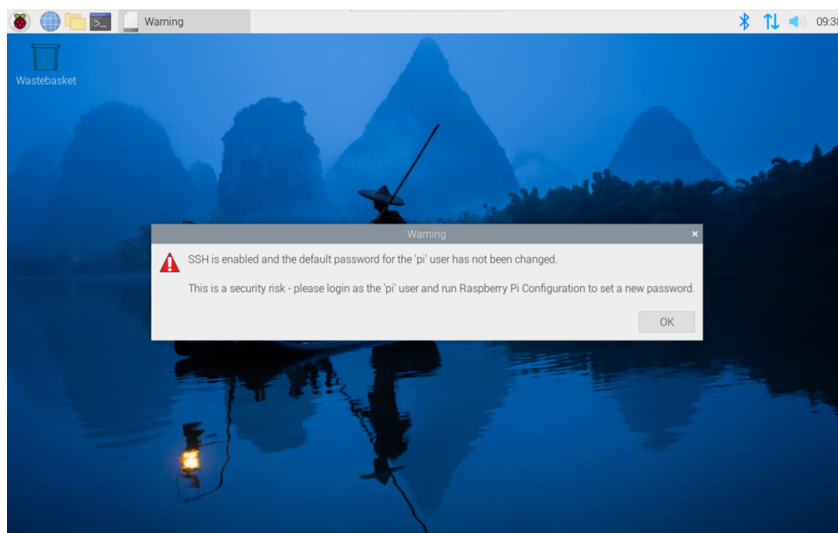
- 红色PWR灯点亮，表示设备已正常供电。
- 绿灯ACT闪烁，表示系统正常启动，然后屏幕的左上角会出现Raspberry Pi 的logo。

提示

默认用户名：`pi`；默认密码：`raspberrypi`。

4.2.1 Raspberry Pi OS (Desktop)

如果产品在出厂时安装的是Desktop版系统，则设备启动完成后，直接进入桌面，如下图所示。



4.2.2 Raspberry Pi OS (Lite)

如果产品在出厂时安装的是Lite版系统，则设备启动完成后会使用默认用户名 `pi` 自动登录，默认密码为 `raspberrypi`，下图所示表示系统已正常启动。

```
[ OK ] Started LSB: rng-tools (Debian variant).
[ OK ] Started WPA supplicant.
[ OK ] Started Authorization Manager.
[ OK ] Reached target Network.
[ OK ] Listening on Load/Save RF Kill Switch Status /dev/rfkill Watch.
        Starting Modem Manager...
        Starting /etc/rc.local Compatibility...
        Starting Permit User Sessions...
[ OK ] Finished Remove Stale OnlimeX4 Metadata Check Snapshots.
[ OK ] Started /etc/rc.local Compatibility.
        Starting Load/Save RF Kill Switch Status...
[ OK ] Finished Permit User Sessions.
[ OK ] Started Getty on tty1.
[ OK ] Reached target Login Prompts.
[ OK ] Started Load/Save RF Kill Switch Status.
[ OK ] Started User Login Management.
        Starting Save/Restore Sound Card State...
[ OK ] Finished Save/Restore Sound Card State.
[ OK ] Reached target Sound Card.
[ OK ] Started Modem Manager.
[ OK ] Started LSB: Switch to on=(unless shift key is pressed).

Raspbian GNU/Linux 11 raspberrypi tty1

raspberrypi login: pi
Password:
Linux raspberrypi 6.1.21-v8+ #1642 SMP PREEMPT Mon Apr  3 17:24:16 BST 2023 aarch64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Jul 11 11:15:28 BST 2023 on tty1

Wi-Fi is currently blocked by rfkill.
Use raspi-config to set the country before use.

pi@raspberrypi:~$ ~
```

5 系统配置

本章介绍系统配置的具体操作。

5.1 查找设备IP

部分应用场景下都需要远程登录或者管理设备，故需要获取设备IP。

5.1.1 在Desktop上的Network图标处查看

设备正常启动并已连接显示器后，可以在Network图标处查看当前设备IP地址。

提示

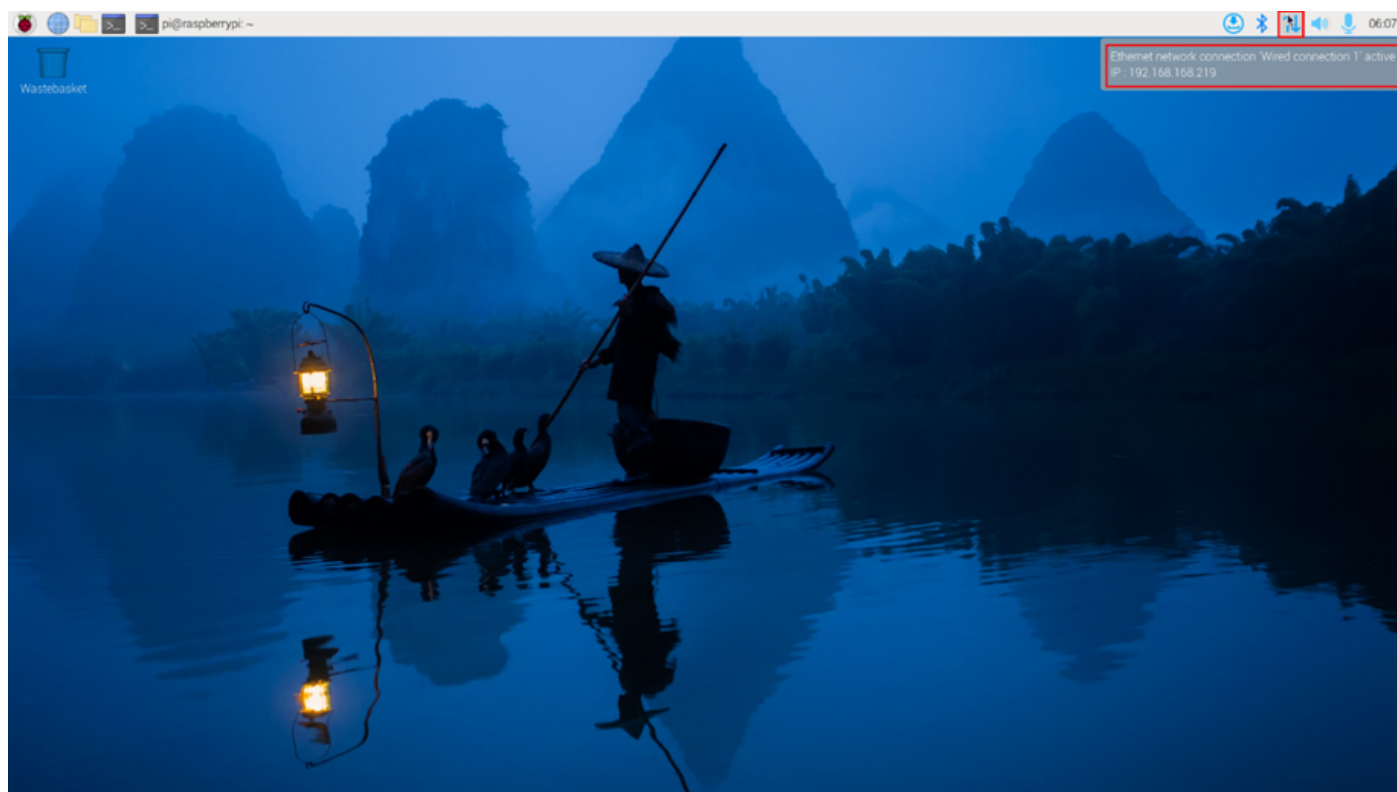
仅Desktop版系统支持。

前提条件：

设备已通过路由器接入网络。

操作步骤：

将鼠标悬浮放置在桌面右上角的图标上，会弹出一个提示，显示当前连接的网络和设备的IP地址。



5.1.2 使用hostname命令查询

设备正常启动后，可以使用hostname命令查看当前设备IP。

前提条件：

设备已通过路由器接入网络。

操作步骤：

在命令窗格执行如下命令，查看设备IP，如下图所示。

```
hostname -I
```

sh

```
pi@raspberrypi:~$ hostname -I  
192.168.168.219
```

5.1.3 使用ifconfig命令查询

设备正常启动后，可以使用ifconfig命令查看当前设备IP。

前提条件：

设备已通过路由器接入网络。

操作步骤：

在命令窗格执行如下命令，查看设备各端口的详细信息，其中eth接口中的inet值即为设备IP，如下图所示。

```
ifconfig
```

sh

```
pi@raspberrypi:~$ ifconfig  
eth0: flags=4096<UP,BROADCAST,MULTICAST> mtu 1500  
    ether d8:3a:dd:04:7e:78 txqueuelen 1000 (Ethernet)  
    RX packets 0 bytes 0 (0.0 B)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 0 bytes 0 (0.0 B)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 192.168.168.206 netmask 255.255.255.0 broadcast 192.168.168.255  
    inet6 fe80::2190:b903:834:c9e8 prefixlen 64 scopeid 0x20<link>  
    ether 00:e0:99:9a:b4:61 txqueuelen 1000 (Ethernet)  
    RX packets 110058 bytes 9062950 (8.6 MiB)  
    RX errors 130 dropped 0 overruns 0 frame 0  
    TX packets 18703 bytes 15467955 (14.7 MiB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

注意

若设备的多个以太网口均接入以太网，则查询到的多个IP均为设备IP地址。

5.1.4 使用Network Manager CLI查询

设备正常启动后，使用nmcli device show命令查看当前设备IP。

前提条件：

设备已通过路由器接入网络。

操作步骤：

在命令窗格执行如下命令，查看设备IP，如下图所示。

```
nmcli device show
```

sh

```
pi@raspberrypi:~$ nmcli device show
GENERAL.DEVICE:      eth0
GENERAL.TYPE:        ethernet
GENERAL.HWADDR:      D8:3A:DD:BF:BE:C5
GENERAL.MTU:         1500
GENERAL.STATE:        100 (connected)
GENERAL.CONNECTION:   Wired connection 1
GENERAL.CON-PATH:     /org/freedesktop/NetworkManager/ActiveConnection/3
WIRED-PROPERTIES.CARRIER: on
IP4.ADDRESS[1]:       192.168.168.219/24
IP4.GATEWAY:          192.168.168.1
IP4.ROUTE[1]:         dst = 192.168.168.0/24, nh = 0.0.0.0, mt = 100
IP4.ROUTE[2]:         dst = 0.0.0.0/0, nh = 192.168.168.1, mt = 100
IP4.DNS[1]:           192.168.168.1
IP6.ADDRESS[1]:       fe80::382a:b964:5832:e59a/64
IP6.GATEWAY:          --
IP6.ROUTE[1]:         dst = fe80::/64, nh = ::, mt = 1024
```

5.1.5 登录路由器查询

当设备正常启动后，可以登录路由器查看当前设备IP。

前提条件：

- 设备已通过路由器接入网络。
- 已获取所在网络的路由器的IP和网络密码，IP地址如192.168.X.X。

操作步骤：

1. 打开浏览器，在地址栏中输入设备所在网络的路由器IP：192.168.X.X，按Enter键进入路由器登录界面。
2. 按照界面提示，输入网络密码，进入路由器管理界面。
3. 在管理界面的终端设备中找到设备的IP地址。

5.1.6 使用nmap工具扫描获取

当设备正常启动后，可以使用nmap工具扫描当前网络下的IP来获取设备IP信息，nmap支持Linux、macOS、Windows等多个平台。

前提条件：

- 设备已通过路由器接入网络。
- 已获取所在网络的路由器的IP网段和掩码，例如192.168.X.X/24，其中24为子网掩码。

操作步骤：

例如使用nmap扫描192.168.3.0/24网段，则可以使用如下步骤：

1. 打开nmap工具，扫描192.168.X.X/24网段中的主机。

注意

nmap工具在不同的操作系统中的操作不同，请根据实际界面或命令的提示来操作即可。

2. 根据扫描出的结果，获取设备IP。

5.2 远程登录

远程登录的方法很多，用户可以根据实际的需求进行选择，本文仅介绍SSH和VNC两种远程登录方法。

5.2.1 通过SSH连接到设备

设备正常启动后，可以选择通过SSH远程连接到设备对其进行配置或调试。远程登录的工具由用户自己选择，下文以通过MobaXterm登录为例进行说明。

前提条件：

- 已在PC上安装MobaXterm工具。
- 设备已通过路由器接入网络。
- 已获取设备的IP地址。

操作步骤：

- 1.



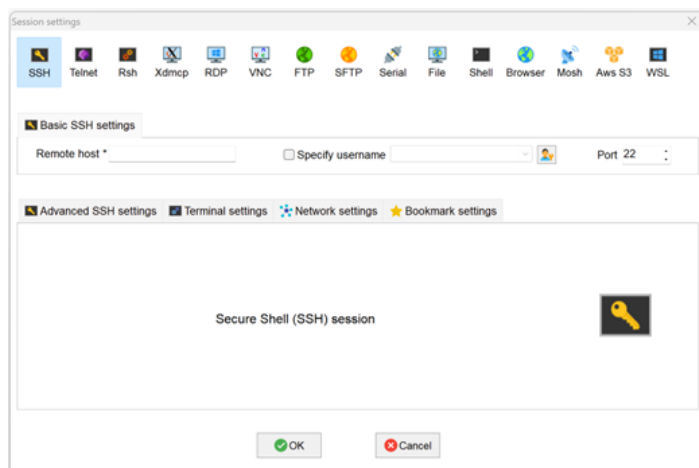
打开MobaXterm，单击 **Session**，打开创建连接的窗口，如下图所示。



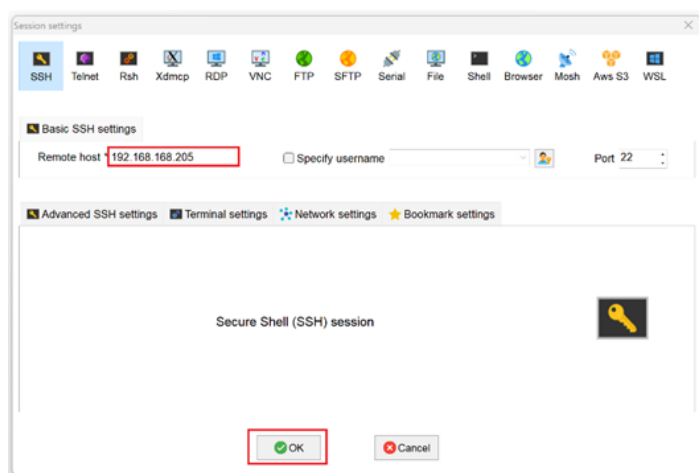
- 2.



单击左上角的 **SSH**，打开SSH连接界面。



3. 输入已获取的设备IP地址后，单击“OK”。

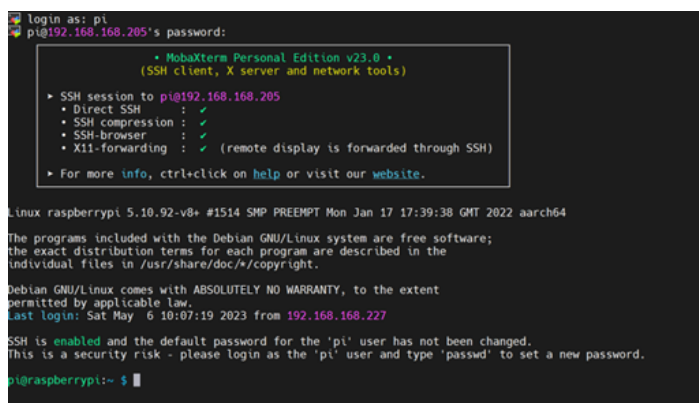


4. 在弹出的提示框中单击“Accept”，进入系统登录界面。

5. 根据提示输入用户名和密码，完成登录后进入系统。

提示

默认用户名为 `pi`，默认密码为 `raspberrypi`。

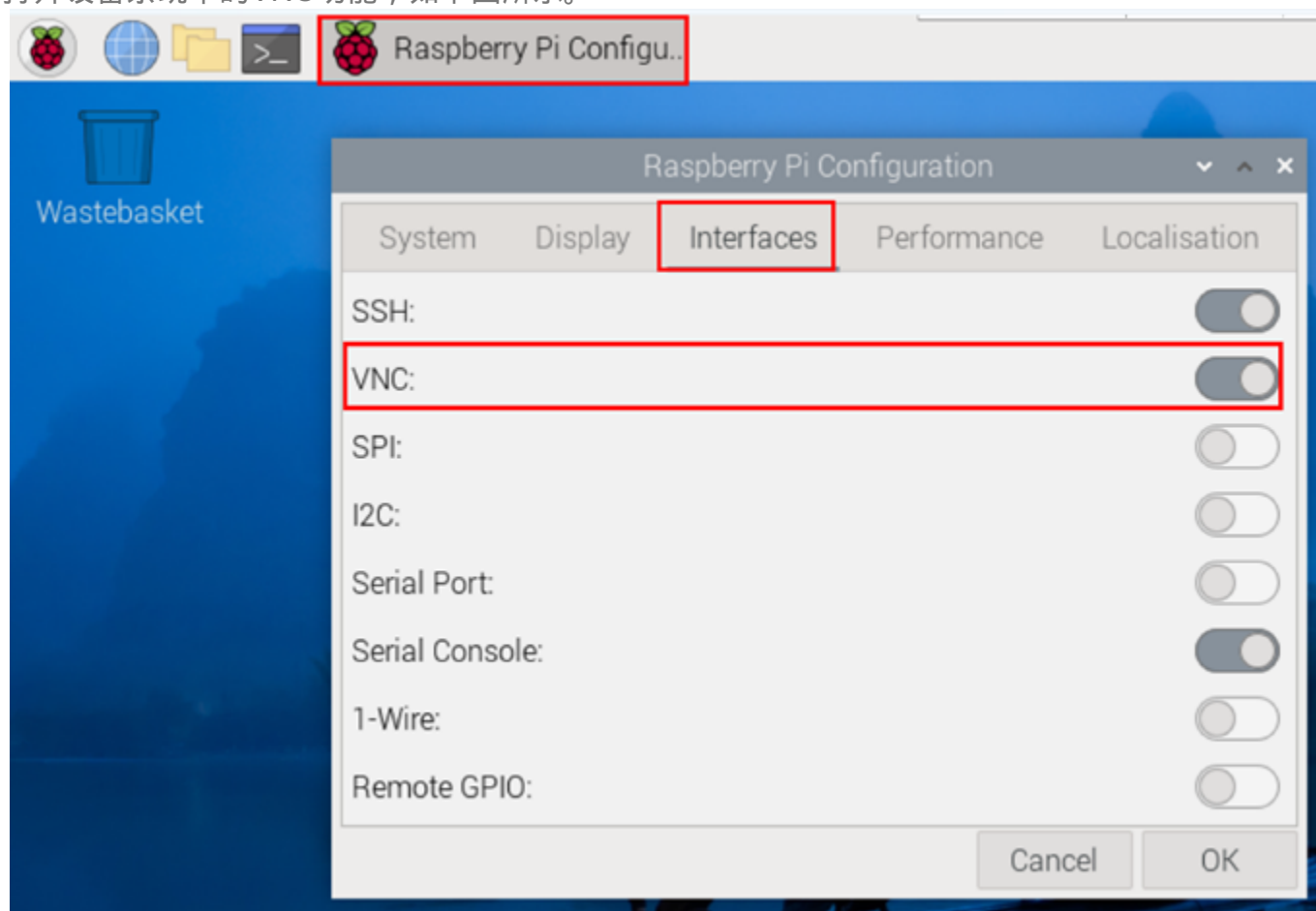


5.2.2 通过VNC连接到设备桌面

设备正常启动后，可以选择通过VNC远程连接到设备对其进行配置或调试。

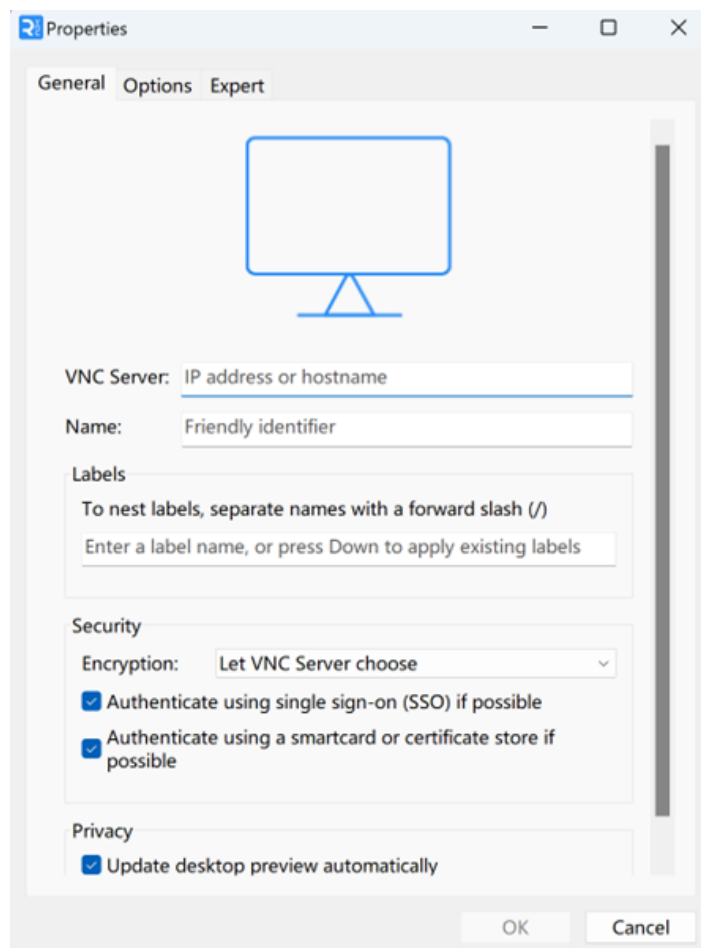
前提条件：

- 已在PC上安装RealVNC Viewer工具。
- 设备已通过路由器接入网络。
- 已获取设备的IP地址。
- 已打开设备系统中的VNC功能，如下图所示。

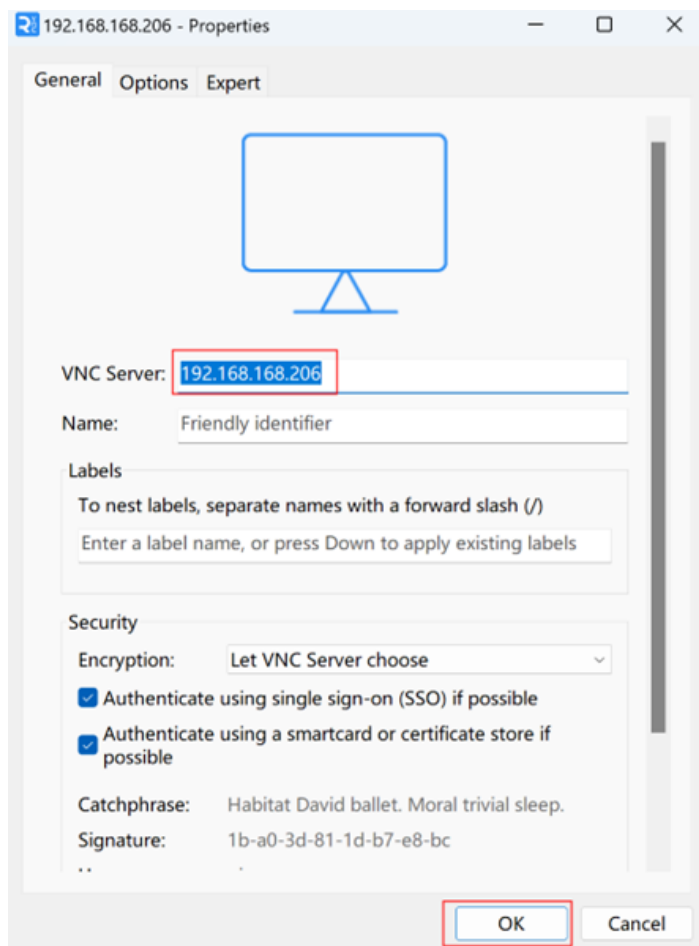


操作步骤：

1. 打开RealVNC Viewer，在菜单栏的“File”中选择“New connection...”，打开创建连接的窗口，如下图所示。



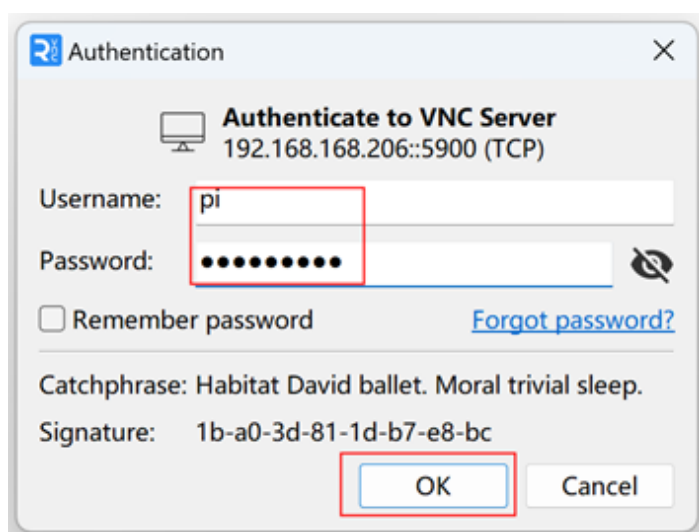
2. 输入已获取的设备IP地址后，单击“OK”。



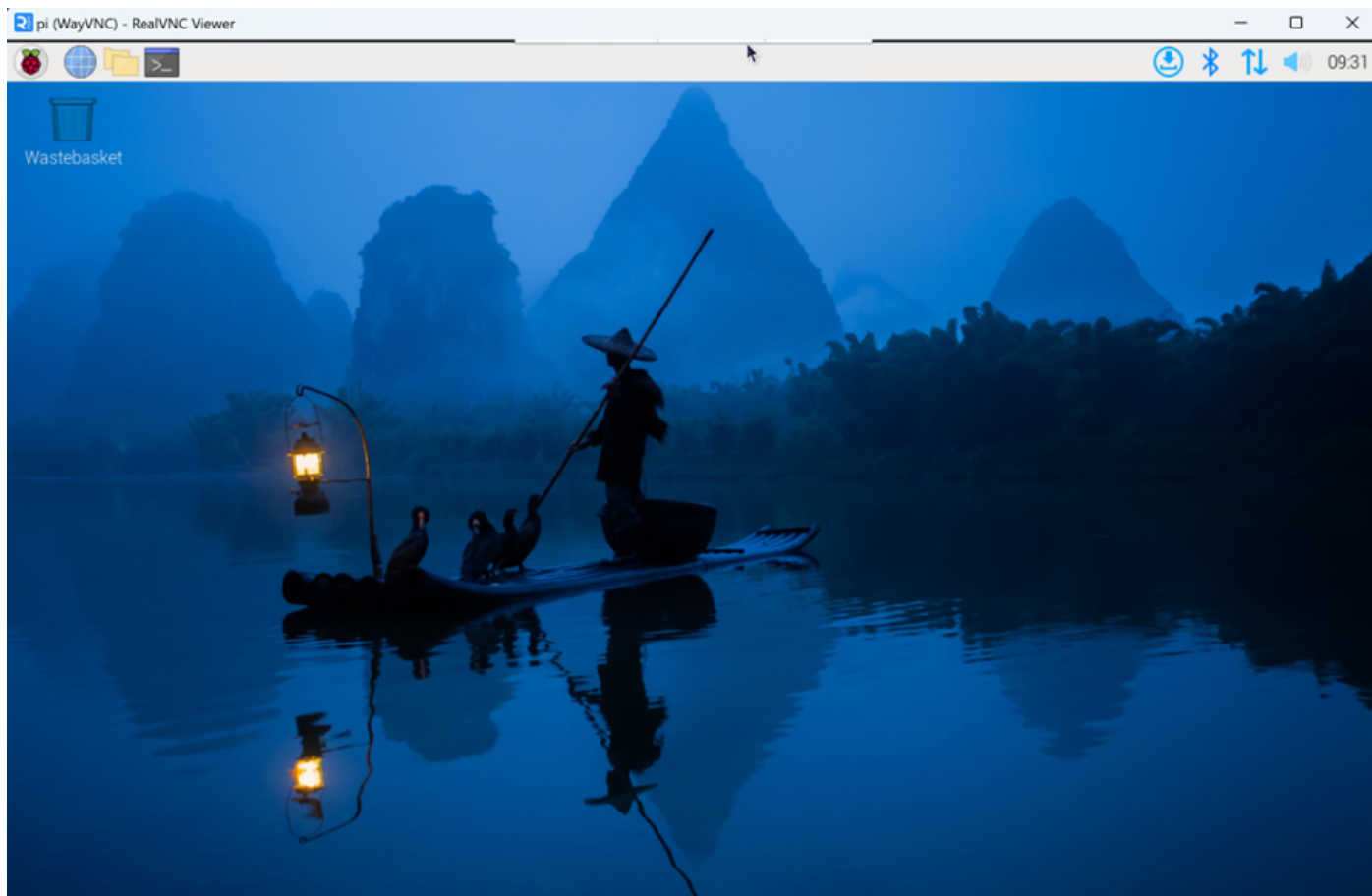
3. 在弹出的Authentication提示框中输入用户名和密码。

提示

默认用户名为 `pi`，默认密码为 `raspberrry`。



4. 选择“OK”，登录后连接到远程桌面。



5.3 配置存储设备

可以将外部硬盘或U盘连接到设备上对应的端口，并挂载文件系统以访问存储在其上的数据。在设备关机时，需要卸载存储设备以便可以将其安全拔出。设备可连接的存储设备包含USB存储设备和SD卡，本章介绍配置存储设备挂载和卸载的具体操作。

5.3.1 挂载存储设备

如果使用的是Lite版本的操作系统（Desktop版本的系统支持自动挂载），则存储设备连接至设备上对应的端口后，需要通过配置将存储设备挂载在特定的文件夹位置，通常挂载在 `/mnt` 文件夹中，例如 `/mnt/mydisk`。

提示

`/mnt` 文件夹必须为空文件夹。

前提条件：

已准备待挂载的存储设备。

操作步骤：

1. 将待挂载的存储设备连接至设备上对应的端口（USB存储设备插入USB端口）。
2. 执行如下命令查看设备上的所有磁盘分区。

```
sudo lsblk -o UUID,NAME,FSTYPE,SIZE,MOUNTPOINT,LABEL,MODEL
```

sh

执行命令后显示信息如下：

```
pi@raspberrypi:~$ sudo lsblk -o UUID,NAME,FSTYPE,SIZE,MOUNTPOINT,LABEL,MODEL
UUID NAME FSTYPE SIZE MOUNTPOINT LABEL MODEL
7C9E-4F13 sda 3.8G 3.8G /media/pi/ADMESY ADMESY Flash_Disk
~sda1 vfat 3.8G /media/pi/ADMESY ADMESY
mmcblk0 7.3G /boot bootfs
C336-AC83 |~mmcblk0p1 vfat 256M /boot bootfs
eaaa4faa-eab6-400c-950f-dc96ae4e0400 ~mmcblk0p2 ext4 7G / rootfs
mmcblk0boot0 4M
mmcblk0boot1 4M
```

- UUID、NAME、FSTYPE、SIZE、MOUNTPOINT、LABEL和MODEL均为需要列举的磁盘参数。
- MOUNTPOINT的类型为“/”和“/boot”。
- LABEL为“ADMESY”即为已插入的USB存储设备，对应的磁盘名称为sda1。
- FSTYPE表示包含的文件系统类型。

如果插入的存储设备的文件系统类型为exFAT，则请依次执行如下命令安装exFAT驱动程序。

```
sudo apt update
sudo apt install exfat-fuse
```

sh

如果插入的存储设备的文件系统类型为NTFS（仅支持读权限），则可以安装ntfs-3g驱动程序来实现写权限。依次执行如下命令安装ntfs-3g驱动程序。

```
sudo apt update
sudo apt install ntfs-3g
```

sh

3. 执行如下命令来获取磁盘分区的位置。

```
sudo blkid
```

sh

执行命令后显示信息如下，其中连接的存储设备的磁盘分区显示为“/dev/sda1”。

```
pi@raspberrypi:~$ sudo blkid
/dev/mmcblk0p1: LABEL="FATBOOT" LABEL="bootfs" UUID="c336-ac83" BLOCK_SIZE="512" TYPE="vfat" PARTUUID="7902e585-01"
/dev/mmcblk0p2: LABEL="rootfs" UUID="eaaa4faa-eab6-400c-950f-dc96ae4e0400" BLOCK_SIZE="4096" TYPE="ext4" PARTUUID="7902e585-02"
/dev/sda1: LABEL="ADMESY" UUID="7C9E-4F13" BLOCK_SIZE="512" TYPE="vfat"
```

4. 创建一个目标文件夹作为存储设备的挂载点。假设挂载点名称为mydisk、待挂载的目录为“/mnt”，则执行的命令如下：

```
sudo mkdir /mnt/mydisk
```

sh

5. 在创建的挂载点挂载存储设备，执行的命令如下：

```
sudo mount /dev/sda1 /mnt/mydisk
```

sh

6. 通过执行如下命令来验证挂载存储设备是否成功。

```
ls /mnt/mydisk
```

sh

- 执行命令后，若显示信息列出了存储设备中的所有文件，则表示挂载成功。
- 执行命令后，若显示信息未列出相关文件内容，则表示挂载失败。

5.3.2 卸载存储设备

当设备关机时，需要手动卸载存储设备以便可以安全地将其拔出。

提示

Lite和Desktop版本系统均需要手动卸载存储设备。

前提条件：

已成功挂载存储设备。

操作步骤：

假如“/mnt”为挂载的目录，“mydisk”为挂载点名称，则可以执行如下命令来完成卸载。

```
sudo umount /mnt/mydisk
```

sh

- 执行命令后，若未显示错误信息，则表示已完成卸载，可以完全拔出存储设备。
- 执行命令后，若显示错误信息，则表示卸载不成功。

5.3.3 设置存储设备自动挂载

如果使用的是Lite版本的操作系统，则可以通过修改fstab设置实现自动挂载。

前提条件：

已将待挂载的存储设备连接至设备上对应的端口。

操作步骤：

1. 执行如下命令查看设备上的所有磁盘分区，获取待挂载的存储设备的文件系统类型，如下图中的“vfat”。

```
sudo lsblk -o UUID,NAME,FSTYPE,SIZE,MOUNTPOINT,LABEL,MODEL
```

sh

```
pi@raspberrypi:~$ sudo lsblk -o UUID,NAME,FSTYPE,SIZE,MOUNTPOINT,LABEL,MODEL
UUID                                NAME      FSTYPE  SIZE MOUNTPOINT  LABEL  MODEL
7C9E-4F13                          sda1      vfat    3.8G /media/pi/ADMESY ADMESY  Flash_Disk
C336-AC83                          mmcblk0p1 vfat    256M /boot        bootfs
eaaa4faa-eab6-400c-950f-dc96ae4e0400 mmcblk0p2 ext4     7G  /            rootfs
                                     mmcblk0boot0
                                     4M
                                     mmcblk0boot1
                                     4M
```

2. 执行如下命令，获取待挂载的存储设备的UUID，例如下图中的“7C9E-4F13”。

```
sudo blkid
```

sh

```
pi@raspberrypi:~$ sudo blkid
/dev/mmcblk0p1: LABEL="FATBOOT" LABEL="bootfs" UUID="C336-AC83" BLOCK_SIZE="512" TYPE="vfat"
PARTUUID="7902e585-01"
/dev/mmcblk0p2: LABEL="rootfs" UUID="eaaa4faa-eab6-400c-950f-dc96ae4e0400" BLOCK_SIZE="4096" TYPE="ext4"
PARTUUID="7902e585-02"
/dev/sda1: LABEL="ADMESY" UUID="7C9E-4F13" BLOCK_SIZE="512" TYPE="vfat"
```

3. 执行如下命令，打开fstab文件。

```
sudo nano /etc/fstab
```

sh

4. 将如下内容添加至fstab文件中。

```
UUID=7C9E-4F13 /mnt/mydisk vfat defaults,auto,users,rw,nofail 0 0
```

sh

- UUID的值为上步骤2中查询到的值
- `/mnt` 为待挂载的目录，`mydisk` 为挂载点名称
- vfat为步骤1中查询的文件系统类型。
- 如果文件系统类型为FAT或NTFS，则添加的内容为“`UUID=7C9E-4F13 /mnt/mydisk vfat defaults,auto,users,rw,nofail umask = 000 0 0`”，将允许所有用户对存储设备上的每个文件进行“读/写”访问。

提示

更多fstab命令的信息，可以通过执行 `man fstab` 命令来查看。

5. 使用 `Ctrl+X` 保存文件，退出编辑模式。

5.4 配置以太网 IP

出厂默认为自动获取IP地址，如果需要重新配置IP，可通过NetworkManager和dhcpcd两种工具来配置。

5.4.1 使用NetworkManager工具配置

出厂默认的Desktop和Lite操作系统均已使能NetworkManager，可直接使用NetworkManager进行配置。

5.4.1.1 Raspberry Pi OS (Desktop)

在Desktop版本的操作系统中，建议使用图形化的NetworkManager工具来配置IP。

提示

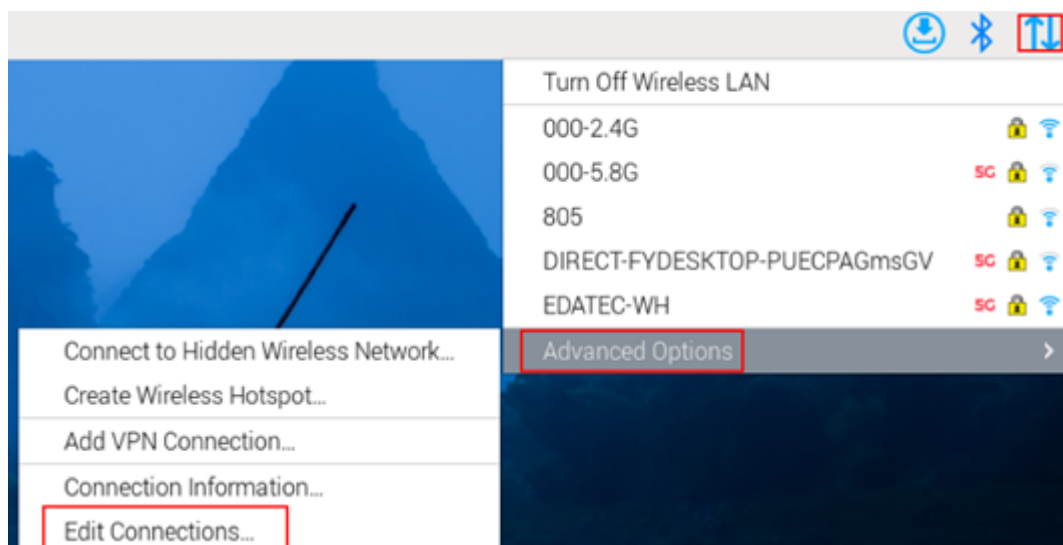
Desktop版本的操作系统已默认安装NetworkManager图形化工具。

前提条件：

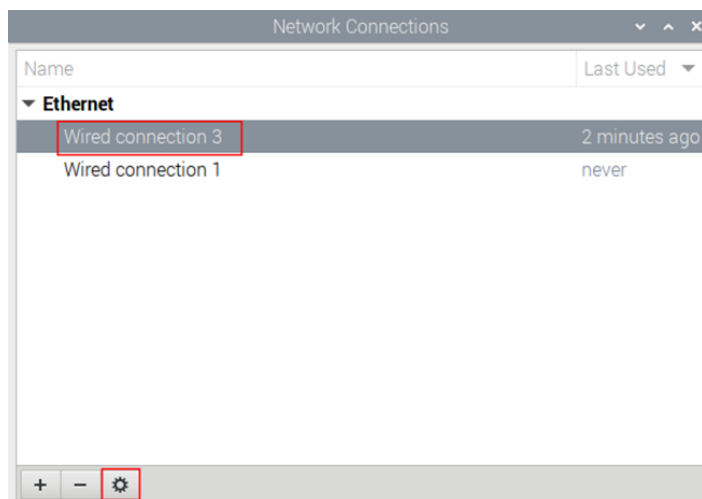
已使能Wi-Fi功能。

操作步骤：

1. 左键单击桌面右上角的  图标，在菜单中选择“Advanced Options→Edit Connections”。

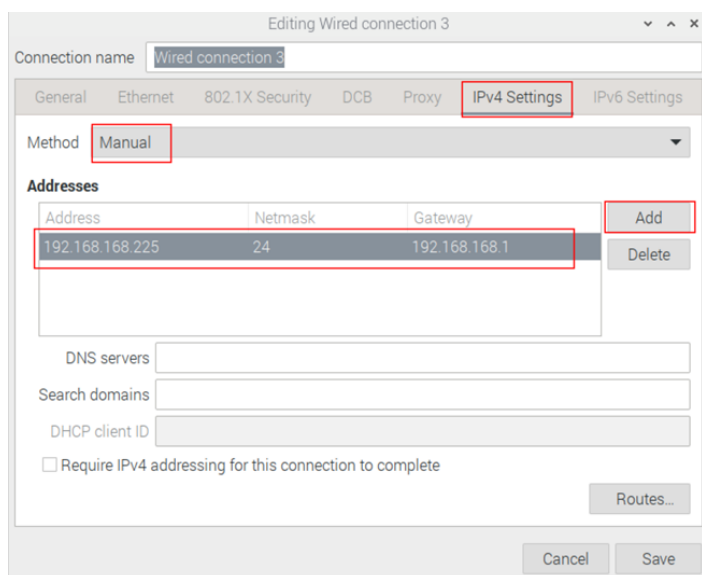


2. 在弹出的Network Connections窗格中选中要修改的连接名称，再单击下方的设置按钮。

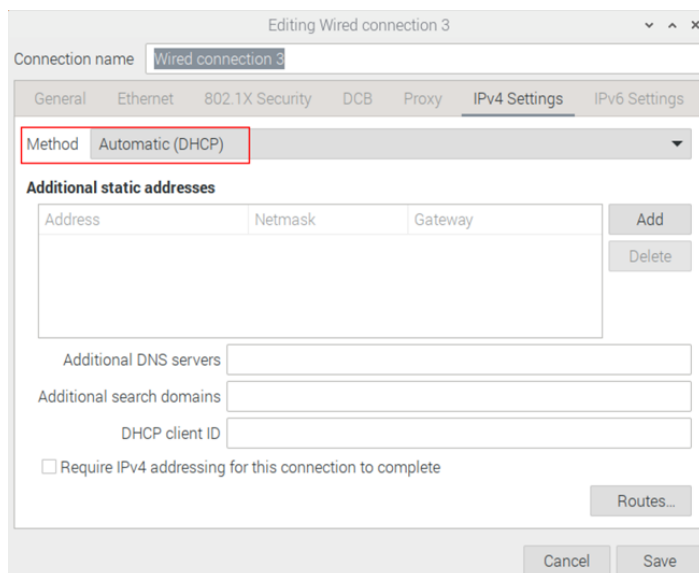


3. 在弹出的Editing Wired connection中选择IPv4 Settings配置页，按需设置IP地址。

- 如果要将IP设置为静态IP，则设置Method为Manual，在Addresses中增加一个条目并输入对应的IP地址信息。



- 如果要将IP设置为自动获取，则仅需要将Method设置为Automatic(DHCP)。



4. 单击 `save` 返回至Network Connections，关闭页面。
5. 在命令窗格中执行 `sudo reboot` 命令，重启设备。

5.4.1.2 Raspberry Pi OS (Lite)

在Lite版本的操作系统中，支持通过命令行来配置IP。

前提条件：

已使能NetworkManager。

操作步骤：

设置静态IP地址

1. 获取分配的IP地址、子网掩码和网关地址，例如IP地址为192.168.1.101/24，网关IP为192.168.1.1。
2. 获取待修改的连接的名称，例如e167c45f-efed-3f8d-89a5-f2430f92fae8，可在命令窗格中执行如下命令查询连接名称。

```
nmcli c
```

sh

```
pi@raspberrypi:~$ nmcli c
NAME                UUID                                TYPE      DEVICE
Wired connection 1  e167c45f-efed-3f8d-89a5-f2430f92fae8 ethernet  eth0
EDATEC-WH           0e6ae3ef-d53d-447d-9da7-79f72293c3f3 wifi      wlan0
Wired connection 2  2699e0b9-277b-36d4-b145-8bd29ad924c2 ethernet  --
Wired connection 3  c0d88cab-714c-3dd1-acd4-595787994af4 ethernet  --
```

3. 在命令窗格执行如下命令，将对应连接的IP地址设置为已获取的IP地址。

```
sudo nmcli connection modify e167c45f-efed-3f8d-89a5-f2430f92fae8 ipv4.addresses 192.168.1.101
```

sh

4. 执行如下命令，将网关IP设置为已获取的网关IP。

```
sh
sudo nmcli connection modify e167c45f-efed-3f8d-89a5-f2430f92fae8 ipv4.gateway 192.168.1.1
```

设置自动获取IP

1. 获取待修改的连接的名称，例如e167c45f-efed-3f8d-89a5-f2430f92fae8，可在命令窗格中执行如下命令查询连接名称。

```
nmcli c
```

```
pi@raspberrypi:~$ nmcli c
NAME                                UUID                                TYPE      DEVICE
Wired connection 1                  e167c45f-efed-3f8d-89a5-f2430f92fae8 ethernet  eth0
EDATEC-WH                           0e6ae3ef-d53d-447d-9da7-79f72293c3f3 wifi      wlan0
Wired connection 2                  2699e0b9-277b-36d4-b145-8bd29ad924c2 ethernet  --
Wired connection 3                  c0d88cab-714c-3dd1-acd4-595787994af4 ethernet  --
```

2. 执行如下命令，将对应连接的IP地址的方式设置自动获取IP。

```
sh
sudo nmcli connection modify e167c45f-efed-3f8d-89a5-f2430f92fae8 ipv4.method auto
```

5.4.2 使用dhcpcd工具配置

由于出厂默认的Desktop和Lite操作系统均已使能NetworkManager，如果需要切换至使用dhcpcd工具来配置，则需要先停止和禁用NetworkManager服务并启用dhcpcd服务后，再进行配置。

操作步骤：

1. 在命令窗格中执行如下命令，停止NetworkManager服务。

```
sh
sudo systemctl stop NetworkManager
```

2. 执行如下命令，禁用NetworkManager服务。

```
sh
sudo systemctl disable NetworkManager
```

3. 执行如下命令，使能dhcpcd服务。

```
sh
sudo systemctl enable dhcpcd`
```

4. 执行如下命令，重启设备。

```
sudo reboot
```

sh

5. 执行如下命令，打开 `/etc/dhcpd.conf` 文件。

```
sudo nano /etc/dhcpd.conf
```

sh

6. 在 `/etc/dhcpd.conf` 文件的末尾添加如下内容。

```
interface eth0
static ip_address=192.168.168.210/24
static routers=192.168.168.1
static domain_name_servers=192.168.168.1 8.8.8.8 fd51:42f8:caae:d92e::1
```

sh

- `eth0` 表示待配置IP的以太网口；
- 192.168.0.10/24表示待配置的IP地址和子网掩码；
- 192.168.0.1表示待配置的网关IP；
- 8.8.8.8表示DNS服务器地址，根据实际配置；
- fd51:42f8:caae:d92e::1表示IPV6地址，根据实际配置；

```
pi@raspberrypi: ~
File Edit Tabs Help
GNU nano 5.4 /etc/dhcpd.conf *
#static ip6_address=fd51:42f8:caae:d92e::ff/64
#static routers=192.168.0.1
#static domain_name_servers=192.168.0.1 8.8.8.8 fd51:42f8:caae:d92e::1

# It is possible to fall back to a static IP if DHCP fails:
# define static profile
#profile static_eth0
#static ip_address=192.168.1.23/24
#static routers=192.168.1.1
#static domain_name_servers=192.168.1.1

# fallback to static profile on eth0
#interface eth0
#fallback static_eth0

interface eth0
static ip_address=192.168.168.210/24
static routers=192.168.168.1
static domain_name_servers=192.168.1 8.8.8.8 fd51:42f8:caae:d92e::1
```

7. 使用 `Ctrl+S` 保存文件，再执行 `Ctrl+X` 退出编辑模式。

8. 执行如下命令，重启设备。

```
sudo reboot
```

sh

5.5 配置Wi-Fi (可选)

用户可选配带Wi-Fi版本的设备，在使用Wi-Fi之前需要先进行相关配置。

Wi-Fi功能默认是使能的，且默认配置的国家区域为“CN China”，在使用时直接连接Wi-Fi即可。

5.5.1 使用NetworkManager工具配置Wi-Fi连接

出厂默认的Desktop和Lite操作系统均已使能NetworkManager，可直接使用NetworkManager进行配置。

5.5.1.1 Raspberry Pi OS (Desktop)

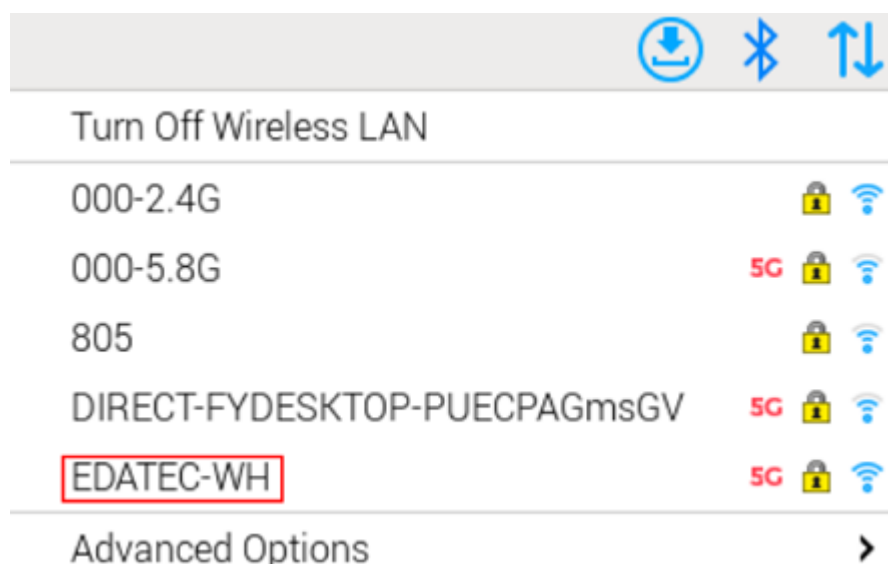
在Desktop版本的操作系统中，可以通过桌面图标来连接Wi-Fi。

前提条件：

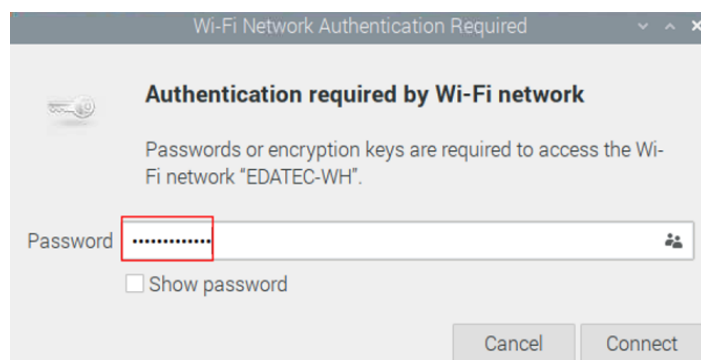
已使能Wi-Fi功能。

操作步骤：

1. 左键单击桌面右上角的  图标，在弹出的Wi-Fi列表中选择待连接的Wi-Fi并单击。



2. 在弹出的Wi-Fi Network Authentication Required窗格的Password右侧输入Wi-Fi密码。



3. 单击“Connect”连接网络，连接完成后，可在桌面右上角单击图标查看Wi-Fi信息。

5.5.1.2 Raspberry Pi OS (Lite)

在Lite版本的操作系统中，支持通过命令行来连接Wi-Fi。

前提条件：

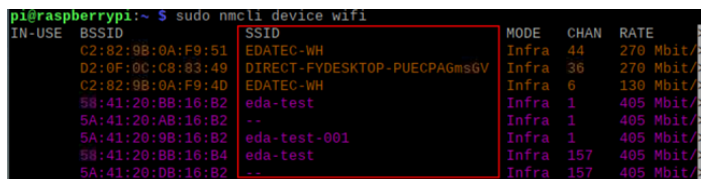
- 已使能Wi-Fi功能。
- 已准备可连接的Wi-Fi名和密码，例如Wi-Fi名为SSID，密码为password。

操作步骤：

1. 打开命令终端窗格执行如下命令，扫描可连接的Wi-Fi列表。

```
sudo nmcli device wifi
```

sh



IN-USE	BSSID	SSID	MODE	CHAN	RATE
	C2:82:9B:0A:F9:51	EDATEC-WH	Infra	44	270 Mbit/s
	D2:0F:0C:C8:83:49	DIRECT-FYDESKTOP-PUECPAGmsGV	Infra	36	270 Mbit/s
	C2:82:9B:0A:F9:4D	EDATEC-WH	Infra	6	130 Mbit/s
	58:41:20:BB:16:B2	eda-test	Infra	1	405 Mbit/s
	5A:41:20:AB:16:B2	--	Infra	1	405 Mbit/s
	5A:41:20:9B:16:B2	eda-test-001	Infra	1	405 Mbit/s
	58:41:20:BB:16:B4	eda-test	Infra	157	405 Mbit/s
	5A:41:20:DB:16:B2	--	Infra	157	405 Mbit/s

2. 执行如下命令，连接待接入的Wi-Fi。

```
sudo nmcli device wifi connect SSID password password
```

sh

- 其中 `SSID` 为待连接的Wi-Fi名称，第二个 `password` 为待连接的Wi-Fi密码。

3. 执行如下命令，设置Wi-Fi自动连接。

```
sudo nmcli connection modify SSID connection.autoconnect yes
```

sh

- 其中 `SSID` 为待连接的Wi-Fi名称。

5.5.2 使用 `raspi-config` 配置Wi-Fi连接

在Desktop和Lite版本的操作系统中，均可以通过 `raspi-config` 来连接Wi-Fi。

前提条件：

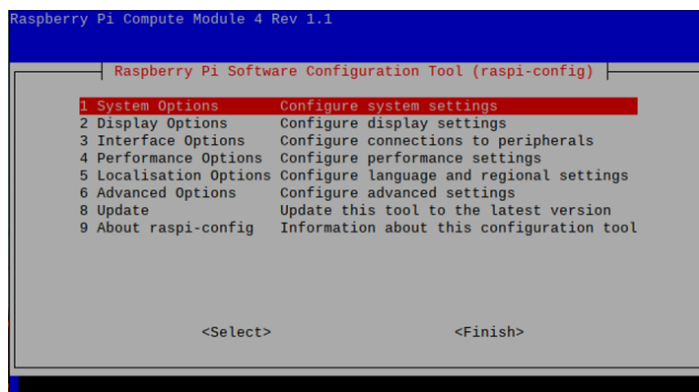
- 已使能Wi-Fi功能。
- 已准备可连接的Wi-Fi名和密码，例如Wi-Fi名为EDATEC-WH，密码为password。

操作步骤：

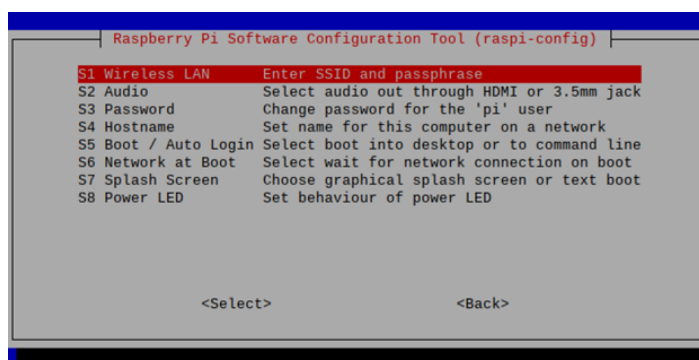
1. 打开命令终端窗格执行如下命令，打开Raspberry Pi Software Configuration Tool (raspi-config) 界面。

```
sudo raspi-config
```

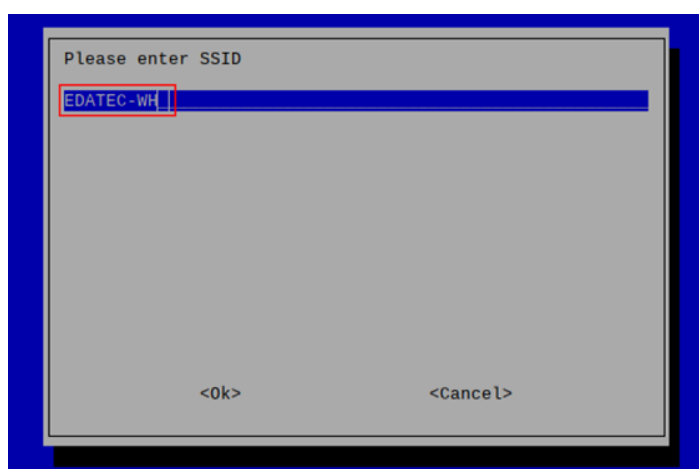
sh



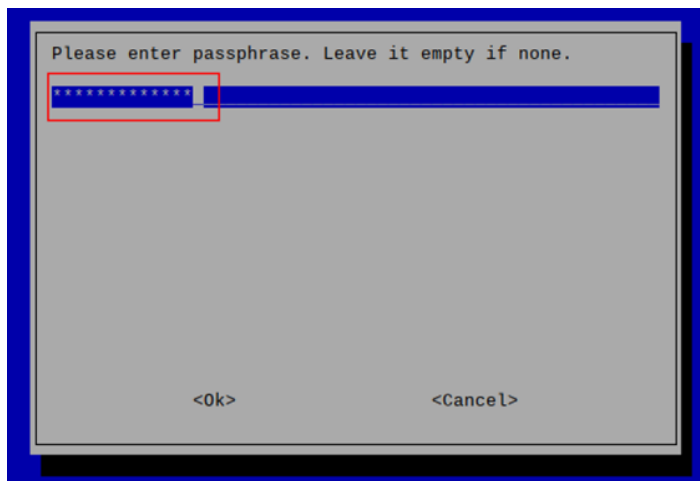
2. 选择“1 System Options”并按Enter，在打开的界面中选择“S1 Wireless LAN”。



3. 按Enter，在打开的Please enter SSID界面中输入Wi-Fi名称。



4. 按Enter，在打开的Please enter passphrase界面中输入Wi-Fi密码（若无密码，则无需输入）。



5. 按Enter连接Wi-Fi，待Wi-Fi连接成功后，选择“Finish”并按Enter完成设置，返回至命令行窗口。

5.6 配置蓝牙 (可选)

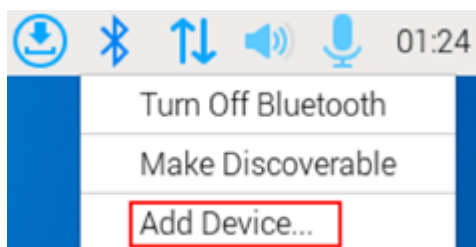
用户可选配带蓝牙版本的设备，且蓝牙功能默认是使能的，在使用蓝牙之前需要先进行相关配置。

5.6.1 Raspberry Pi OS (Desktop)

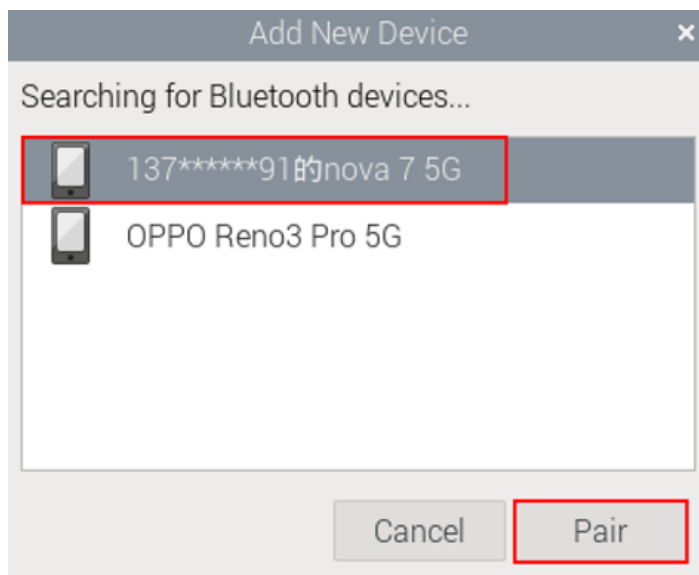
在Desktop版本的操作系统中，可以通过桌面图标来配置蓝牙。

操作步骤：

1. 左键单击桌面右上角的  图标，在弹出的菜单中选择“Add Device”。



2. 在弹出的Add New Device窗格中查看扫描到的蓝牙设备，并选择需要连接的蓝牙设备，单击“Pair”进行配对。



3. 在弹出的提示框中选择“OK”，确认配对请求。

Please confirm that '137*****91的nova 7 5G' is showing the code '645999' to pair

Cancel

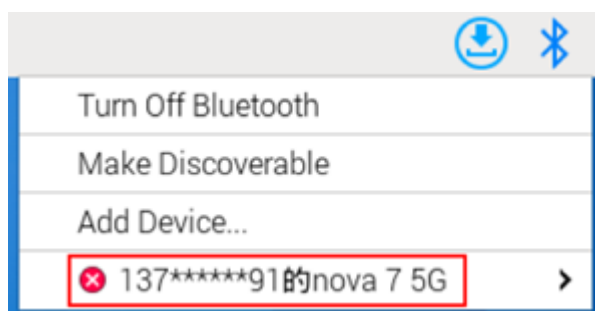
OK

提示

连接的蓝牙设备端也需要确认配对请求，否则会配对失败。

4. 配对成功后，在弹出的提示框中单击“OK”关闭页面。

5. 左键单击桌面右上角的  图标，查看蓝牙设备已连接。



5.6.2 Raspberry Pi OS (Lite)

在Lite版本的操作系统中，可以通过命令行来配置蓝牙。

5.6.2.1 基本配置命令

命令	功能说明
bluetoothctl scan on	开启蓝牙扫描
bluetoothctl scan off	关闭蓝牙扫描

命令	功能说明
bluetoothctl discoverable on	开启蓝牙发现（可以让对方发现）
bluetoothctl discoverable off	关闭蓝牙发现
bluetoothctl trust device_MAC	信任设备
bluetoothctl connect device_MAC	连接设备
bluetoothctl disconnect device_MAC	与设备断开连接

5.6.2.2 配置示例

通过配置示例来介绍配置蓝牙的具体操作。

前提条件：

已开启待配对的蓝牙并确定蓝牙名称。

操作步骤：

1. 进入bluetootch视图。

```
sudo bluetoothctl
```

sh

2. 使能bluetooth。

```
power on
```

sh

3. 扫描蓝牙设备。

```
scan on
```

sh

回显信息：

```
Discovery started
```

```
[CHG] Controller B8:27:EB:85:04:8B Discovering: yes
[NEW] Device 4A:39:CF:30:B3:11 4A-39-CF-30-B3-11
```

4. 查找开启的蓝牙设备名称。

```
devices
```

sh

回显信息： Device 6A:7F:60:69:8B:79 6A-7F-60-69-8B-79 Device 67:64:5A:A3:2C:A2 67-64-5A-A3-2C-A2 Device 56:6A:59:B0:1C:D1 Lefun Device 34:12:F9:91:FF:68 test

5. 配对目标设备。

```
pair 34:12:F9:91:FF:68
```

sh

- 34:12:F9:91:FF:68 为目标设备的 device_MAC 。

回显信息：

```
Attempting to pair with 34:12:F9:91:FF:68
[CHG] Device 34:12:F9:91:FF:68 ServicesResolved: yes
[CHG] Device 34:12:F9:91:FF:68 Paired: yes
Pairing successful
```

提示

连接的蓝牙设备端也需要确认配对请求，否则会配对失败。

6. 添加为信任设备。

```
trust 34:12:F9:91:FF:68
```

sh

34:12:F9:91:FF:68 为目标设备的 device_MAC 。

回显信息：

```
[CHG] Device 34:12:F9:91:FF:68 Trusted: yes
Changing 34:12:F9:91:FF:68 trust succeeded
```

5.7 配置 4G (可选)

用户可选配带4G版本的设备，在使用4G网络之前需要先进行相关配置。若选购的设备包含4G功能，则出厂默认已开启4G功能，用户插入SIM卡之后，再给设备上电，4G网络会自动连接。

我司出厂的设备默认已安装4G拨号重连软件包(ed-qmi-tool)，建议用户采用此工具来连接4G网络。下文介绍基于已安装的ed-qmi-tool工具来配置和连接4G网络的具体操作。

提示

若用户有其他需求，希望通过Raspberry Pi OS中的Network Manager工具来配置网络，具体的操作请参见[配置4G](#)。

5.7.1 无需配置APN的场景

若用户的4G网络无需配置APN，则可以参照如下步骤进行查询4G网络状态。

前提条件：

- 设备已正常启动。
- 包含4G网络的Nano SIM已正确安装至设备的SIM卡槽。

注意

SIM卡不支持带电插拔。

操作步骤：

1. 打开命令窗格，执行如下命令，查询4G网络是否连接。

```
ifconfig
```

```
sh
```

返回信息如下图（wwan0接口表示4G接口）：

```
pi@raspberrypi:~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.8.20 netmask 255.255.255.0 broadcast 192.168.8.255
    inet6 fe80::580c:5b68:eafe:6e5a prefixlen 64 scopeid 0x20<link>
    ether 2c:cf:67:fa:02:1e txqueuelen 1000 (Ethernet)
    RX packets 21347 bytes 7256770 (6.9 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 23885 bytes 2757675 (2.6 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 224 bytes 25909 (25.3 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 224 bytes 25909 (25.3 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.2.181 netmask 255.255.255.0 broadcast 192.168.2.255
    inet6 fe80::78a4:9af5:b3b9:d30e prefixlen 64 scopeid 0x20<link>
    ether 2c:cf:67:fa:02:1f txqueuelen 1000 (Ethernet)
    RX packets 13273 bytes 1683644 (1.6 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 493 bytes 59187 (57.7 KiB)
    TX errors 0 dropped 21 overruns 0 carrier 0 collisions 0

wwan0: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST> mtu 1500
    inet 10.11.0.198 netmask 255.255.255.252 destination 10.11.0.198
    inet6 fe80::5633:bcad:52b3:3f2f prefixlen 64 scopeid 0x20<link>
    unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00 txqueuelen 1000 (UNSPEC)
    RX packets 3 bytes 696 (696.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 7 bytes 932 (932.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

- 若返回信息中wwan0接口显示了具体的IP地址，则表示4G已连接。
- 若返回信息中wwan0接口未显示具体的IP地址，则表示4G未连接。

2. 执行如下命令，查询4G的服务状态。

```
sudo systemctl status ed-lte-daemon.service
```

返回信息如下图：

```
pi@raspberrypi:~$ sudo systemctl status ed-lte-daemon.service
● ed-lte-daemon.service - EDATEC QML Reconnect service
   Loaded: loaded (/usr/lib/systemd/system/ed-lte-daemon.service; enabled; preset: enabled)
   Active: active (running) since Thu 2025-12-25 11:25:26 CST; 14min ago
     Invocation: d4359695fd344a5ead9e78d02a33e23c
       Main PID: 152218 (ed-lte-tool)
         Tasks: 6 (limit: 1571)
            CPU: 1min 26.950s
       CGroup: /system.slice/ed-lte-daemon.service
               └─152218 /usr/local/bin/ed-lte-tool --daemon
                 └─152364 quctel-CM -4

Dec 25 11:39:50 raspberrypi ed-lte-tool[152218]: [2025-12-25 11:39:50][INFO] [cmd] running: ed-net-priority, Some(["wlan0", "400"])
Dec 25 11:39:50 raspberrypi ed-lte-tool[152218]: [2025-12-25 11:39:50][INFO] Successfully set metric 400 for wlan0
Dec 25 11:39:50 raspberrypi ed-lte-tool[152218]: [2025-12-25 11:39:50][INFO] [cmd] running: ed-net-priority, Some(["wwan0", "700"])
Dec 25 11:39:50 raspberrypi ed-lte-tool[152218]: [2025-12-25 11:39:50][INFO] Successfully set metric 700 for wwan0
Dec 25 11:39:52 raspberrypi ed-lte-tool[152218]: [2025-12-25 11:39:52][INFO] [cmd] running: ed-net-priority, Some(["eth0", "100"])
Dec 25 11:39:52 raspberrypi ed-lte-tool[152218]: [2025-12-25 11:39:52][INFO] Successfully set metric 100 for eth0
Dec 25 11:39:52 raspberrypi ed-lte-tool[152218]: [2025-12-25 11:39:52][INFO] [cmd] running: ed-net-priority, Some(["wlan0", "400"])
Dec 25 11:39:52 raspberrypi ed-lte-tool[152218]: [2025-12-25 11:39:52][INFO] Successfully set metric 400 for wlan0
Dec 25 11:39:52 raspberrypi ed-lte-tool[152218]: [2025-12-25 11:39:52][INFO] [cmd] running: ed-net-priority, Some(["wwan0", "700"])
Dec 25 11:39:52 raspberrypi ed-lte-tool[152218]: [2025-12-25 11:39:52][INFO] Successfully set metric 700 for wwan0
```

- 若返回信息中标注位置的信息为“Active：active (running)”，则表示4G状态正常。
- 若返回信息中标注位置的信息为“Active：inactive (dead)”，则表示4G状态异常。

5.7.2 需配置APN的场景

若用户的4G网络需要配置APN，则可以参照如下步骤进行配置。

前提条件：

- 设备已正常启动。
- 包含4G网络的Nano SIM已正确安装至设备的SIM卡槽。
- 已获取APN的名称、用户名和密码，下文以如下的信息举例。
 - APN名称：APN1
 - 用户名：admin
 - 密码：admin

注意

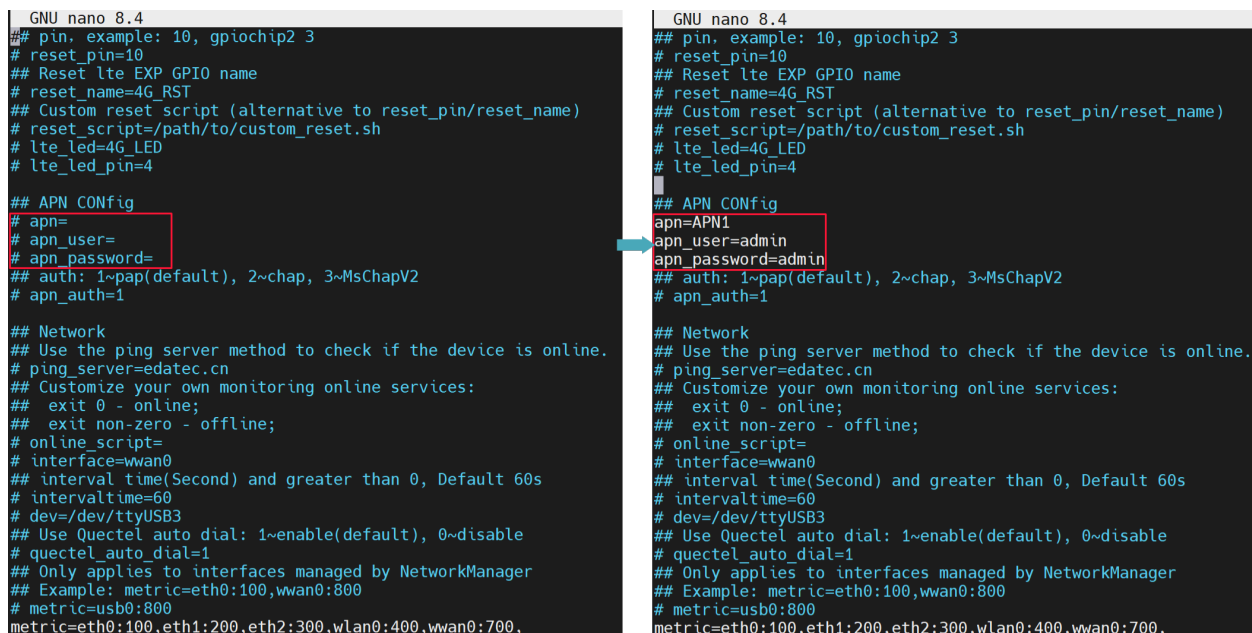
SIM卡不支持带电插拔。

操作步骤：

1. 打开命令窗格，依次执行如下命令，打开 `ed-qml.conf` 配置文件；

```
cd /etc/
sudo nano ed-qml.conf
```

2. 按需设置“APN CONFIG”中的“apn”、“apn_user”和“apn_password”。



```

GNU nano 8.4
## pin. example: 10, gpiochip2 3
# reset_pin=10
## Reset lte EXP GPIO name
# reset_name=4G_RST
## Custom reset script (alternative to reset_pin/reset_name)
# reset_script=/path/to/custom_reset.sh
# lte_led=4G_LED
# lte_led_pin=4

## APN Config
# apn=
# apn_user=
# apn_password=
## auth: 1~pap(default), 2~chap, 3~MsChapV2
# apn_auth=1

## Network
## Use the ping server method to check if the device is online.
# ping_server=edatec.cn
## Customize your own monitoring online services:
## exit 0 - online;
## exit non-zero - offline;
# online_script=
# interface=wwan0
## interval time(Second) and greater than 0, Default 60s
# intervaltime=60
# dev=/dev/ttyUSB3
## Use Quectel auto dial: 1~enable(default), 0~disable
# quectel_auto_dial=1
## Only applies to interfaces managed by NetworkManager
## Example: metric=eth0:100,wwan0:800
# metric=usb0:800
metric=eth0:100,eth1:200,eth2:300,wlan0:400,wwan0:700,

GNU nano 8.4
## pin. example: 10, gpiochip2 3
# reset_pin=10
## Reset lte EXP GPIO name
# reset_name=4G_RST
## Custom reset script (alternative to reset_pin/reset_name)
# reset_script=/path/to/custom_reset.sh
# lte_led=4G_LED
# lte_led_pin=4

## APN Config
apn=APN1
apn_user=admin
apn_password=admin
## auth: 1~pap(default), 2~chap, 3~MsChapV2
# apn_auth=1

## Network
## Use the ping server method to check if the device is online.
# ping_server=edatec.cn
## Customize your own monitoring online services:
## exit 0 - online;
## exit non-zero - offline;
# online_script=
# interface=wwan0
## interval time(Second) and greater than 0, Default 60s
# intervaltime=60
# dev=/dev/ttyUSB3
## Use Quectel auto dial: 1~enable(default), 0~disable
# quectel_auto_dial=1
## Only applies to interfaces managed by NetworkManager
## Example: metric=eth0:100,wwan0:800
# metric=usb0:800
metric=eth0:100,eth1:200,eth2:300,wlan0:400,wwan0:700,

```

提示

“4G reset”、“Network”和“intervaltime”等参数均支持用户按需配置。

3. 输入 `ctrl+o` 保存文件，再按 `Enter`，最后输入 `ctrl+x` 退出文件编辑模式。
4. 执行如下命令，重启4G网络服务，使配置生效。

```
sudo systemctl restart ed-lte-daemon.service
```

sh

5. 执行如下命令，查询4G网络是否连接。

```
ifconfig
```

sh

返回信息如下图（wwan0接口表示4G接口）：

```

pi@raspberrypi:~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.8.20 netmask 255.255.255.0 broadcast 192.168.8.255
    inet6 fe80::580c:5b68:eafe:6e5a prefixlen 64 scopeid 0x20<link>
    ether 2c:cf:67:fa:02:1e txqueuelen 1000 (Ethernet)
    RX packets 21347 bytes 7256770 (6.9 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 23885 bytes 2757675 (2.6 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 224 bytes 25909 (25.3 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 224 bytes 25909 (25.3 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.2.181 netmask 255.255.255.0 broadcast 192.168.2.255
    inet6 fe80::78a4:9af5:b3b9:d30e prefixlen 64 scopeid 0x20<link>
    ether 2c:cf:67:fa:02:1f txqueuelen 1000 (Ethernet)
    RX packets 13273 bytes 1683644 (1.6 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 493 bytes 59187 (57.7 KiB)
    TX errors 0 dropped 21 overruns 0 carrier 0 collisions 0

wwan0: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST> mtu 1500
    inet 10.11.0.198 netmask 255.255.255.252 destination 10.11.0.198
    inet6 fe80::5633:bcad:52b3:3f2f prefixlen 64 scopeid 0x20<link>
    unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00 txqueuelen 1000 (UNSPEC)
    RX packets 3 bytes 696 (696.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 7 bytes 932 (932.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

- 若返回信息中wwan0接口显示了具体的IP地址，则表示4G已连接。
- 若返回信息中wwan0接口未显示具体的IP地址，则表示4G未连接。

6. 执行如下命令，查询4G的服务状态。

```
sudo systemctl status ed-lte-daemon.service
```

sh

返回信息如下图：

```

pi@raspberrypi:~$ sudo systemctl status ed-lte-daemon.service
● ed-lte-daemon.service - EDATEC QML Reconnect service
   Loaded: loaded (/usr/lib/systemd/system/ed-lte-daemon.service; enabled; preset: enabled)
   Active: active (running) since Thu 2025-12-25 11:25:26 CST; 14min ago
  Invocation: d4359b95fd344a5ead9e78d02a33e23c
    Main PID: 152218 (ed-lte-tool)
      Tasks: 6 (limit: 1571)
         CPU: 1min 26.950s
   CGroup: /system.slice/ed-lte-daemon.service
           └─152218 /usr/local/bin/ed-lte-tool --daemon
             └─152364 quectel-CM -4

Dec 25 11:39:50 raspberrypi ed-lte-tool[152218]: [2025-12-25 11:39:50][INFO] [cmd] running: ed-net-priority, Some(["wlan0", "400"])
Dec 25 11:39:50 raspberrypi ed-lte-tool[152218]: [2025-12-25 11:39:50][INFO] Successfully set metric 400 for wlan0
Dec 25 11:39:50 raspberrypi ed-lte-tool[152218]: [2025-12-25 11:39:50][INFO] [cmd] running: ed-net-priority, Some(["wwan0", "700"])
Dec 25 11:39:50 raspberrypi ed-lte-tool[152218]: [2025-12-25 11:39:50][INFO] Successfully set metric 700 for wwan0
Dec 25 11:39:52 raspberrypi ed-lte-tool[152218]: [2025-12-25 11:39:52][INFO] [cmd] running: ed-net-priority, Some(["eth0", "100"])
Dec 25 11:39:52 raspberrypi ed-lte-tool[152218]: [2025-12-25 11:39:52][INFO] Successfully set metric 100 for eth0
Dec 25 11:39:52 raspberrypi ed-lte-tool[152218]: [2025-12-25 11:39:52][INFO] [cmd] running: ed-net-priority, Some(["wlan0", "400"])
Dec 25 11:39:52 raspberrypi ed-lte-tool[152218]: [2025-12-25 11:39:52][INFO] Successfully set metric 400 for wlan0
Dec 25 11:39:52 raspberrypi ed-lte-tool[152218]: [2025-12-25 11:39:52][INFO] [cmd] running: ed-net-priority, Some(["wwan0", "700"])
Dec 25 11:39:52 raspberrypi ed-lte-tool[152218]: [2025-12-25 11:39:52][INFO] Successfully set metric 700 for wwan0

```

- 若返回信息中标注位置的信息为“Active：active (running)”，则表示4G状态正常。
- 若返回信息中标注位置的信息为“Active：inactive (dead)”，则表示4G状态异常。

5.7.3 基本配置命令

若遇到4G网络无法连接的情况，可使用如下命令进行查询和配置。

命令	说明
sudo ed-lte-tool --daemon	启动4G监测程序并自动连接4G网络

命令	说明
ifconfig	查询4G网络是否连接，其中wwan0接口表示4G接口，如下图所示。 <pre>pi@raspberrypi:~\$ ifconfig eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500 inet 192.168.8.20 netmask 255.255.255.0 broadcast 192.168.8.255 inet6 fe80::580c:5b58:eafe:6e5a prefixlen 64 scopeid 0x20<link> ether 2c:c:f5:97:fa:62:14 txqueuelen 1000 (Ethernet) RX packets 21347 bytes 7256770 (6.9 MiB) RX errors 0 dropped 0 overruns 0 frame 0 TX packets 23885 bytes 2757476 (2.6 MiB) TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0 lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536 inet 127.0.0.1 netmask 255.0.0.0 inet6 ::1 prefixlen 128 scopeid 0x10<host> loop txqueuelen 1000 (Local Loopback) RX packets 224 bytes 25909 (25.3 KiB) RX errors 0 dropped 0 overruns 0 frame 0 TX packets 224 bytes 25909 (25.3 KiB) TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0 wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500 inet 192.168.2.181 netmask 255.255.255.0 broadcast 192.168.2.255 inet6 fe80::76a4:9af5:b3b9:d30e prefixlen 64 scopeid 0x20<link> ether 2c:c:f5:97:fa:62:14 txqueuelen 1000 (Ethernet) RX packets 13273 bytes 1683644 (1.6 MiB) RX errors 0 dropped 0 overruns 0 frame 0 TX packets 493 bytes 59187 (57.7 KiB) TX errors 0 dropped 21 overruns 0 carrier 0 collisions 0 wwan0: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST> mtu 1500 inet 10.11.0.198 netmask 255.255.255.252 destination 10.11.0.198 inet6 fe80::68b3:bca0:52b3:3f2f prefixlen 64 scopeid 0x20<link> unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00 txqueuelen 1000 (UNSPEC) RX packets 3 bytes 696 (696.0 B) RX errors 0 dropped 0 overruns 0 frame 0 TX packets 7 bytes 932 (932.0 B) TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0</pre>
	• 若wwan0接口显示了具体的IP地址，则表示4G已连接。 • 若wwan0接口未显示具体的IP地址，则表示4G未连接。
sudo systemctl status ed-lte-daemon.service	查询4G的服务状态，如下图所示。 <pre>pi@raspberrypi:~\$ sudo systemctl status ed-lte-daemon.service ● ed-lte-daemon.service - EDATEC QML Reconnect service Loaded: loaded (/usr/lib/systemd/system/ed-lte-daemon.service; enabled; preset: enabled) Active: active (running) since Thu 2025-12-25 11:25:26 CST; 14min ago Invocation: 04359b95fd344a5ead9e78d02a33e23c Main PID: 152218 (ed-lte-tool) Tasks: 6 (limit: 1571) CPU: 1min 26.950s CGroup: /system.slice/ed-lte-daemon.service └─152218 /usr/local/bin/ed-lte-tool --daemon └─152364 quectel-CM-4 Dec 25 11:39:50 raspberrypi ed-lte-tool[152218]: [2025-12-25 11:39:50][INFO] [cmd] running: ed-net-priority, Some(["wlan0", "400"]) Dec 25 11:39:50 raspberrypi ed-lte-tool[152218]: [2025-12-25 11:39:50][INFO] Successfully set metric 400 for wlan0 Dec 25 11:39:50 raspberrypi ed-lte-tool[152218]: [2025-12-25 11:39:50][INFO] [cmd] running: ed-net-priority, Some(["wwan0", "700"]) Dec 25 11:39:50 raspberrypi ed-lte-tool[152218]: [2025-12-25 11:39:50][INFO] Successfully set metric 700 for wwan0 Dec 25 11:39:52 raspberrypi ed-lte-tool[152218]: [2025-12-25 11:39:52][INFO] [cmd] running: ed-net-priority, Some(["eth0", "100"]) Dec 25 11:39:52 raspberrypi ed-lte-tool[152218]: [2025-12-25 11:39:52][INFO] Successfully set metric 100 for eth0 Dec 25 11:39:52 raspberrypi ed-lte-tool[152218]: [2025-12-25 11:39:52][INFO] [cmd] running: ed-net-priority, Some(["wlan0", "400"]) Dec 25 11:39:52 raspberrypi ed-lte-tool[152218]: [2025-12-25 11:39:52][INFO] Successfully set metric 400 for wlan0 Dec 25 11:39:52 raspberrypi ed-lte-tool[152218]: [2025-12-25 11:39:52][INFO] [cmd] running: ed-net-priority, Some(["wwan0", "700"]) Dec 25 11:39:52 raspberrypi ed-lte-tool[152218]: [2025-12-25 11:39:52][INFO] Successfully set metric 700 for wwan0</pre>
	• 若显示标注位置的信息为“Active：active (running)”，则表示4G状态正常。 • 若显示标注位置的信息为“Active：inactive (dead)”，则表示4G状态异常。
sudo systemctl enable ed-lte-daemon.service	使能4G服务
sudo systemctl start ed-lte-daemon.service	开启4G服务
sudo systemctl stop ed-lte-daemon.service	关闭4G服务
sudo ed-lte-tool -m	查看4G模块的信息
sudo ed-lte-tool -s	查看4G信号强度
sudo ed-lte -c	拨号上网
sudo ed-lte -d	断开网络连接，断开后不支持自动重连
	进入/var/log/ed-qmi/目录下，查看日志文件，其中xxxx-xx-xx为年-月-日，格式为2025-06-18

命令	说明
cd /var/log/ed-qmi/ sudo nano xxxx-xx-xx.log	
journalctl -u ed-lte-daemon.service	查看4G网络的实时日志

提示

在4G端口连接正常的情况下，若查询到4G的服务状态异常，可依次执行如下命令，使能4G服务和开启4G服务。

```
sudo systemctl enable ed-lte-daemon.service
sudo systemctl start ed-lte-daemon.service
```

sh

5.8 配置 Buzzer

设备包含1个蜂鸣器，支持手动配置打开和关闭蜂鸣器。

1. 依次执行如下命令，检测和安装gpio工具。

```
sudo apt update
sudo apt install gpio
```

sh

2. 执行如下命令，打开和关闭蜂鸣器。

• 打开蜂鸣器：

```
gpiofind BUZZER_EN | awk '{print substr($0,9)}' | xargs -i bash -c "gpioset {}=1"
```

sh

1 表示高电平

• 关闭蜂鸣器：

```
gpiofind BUZZER_EN | awk '{print substr($0,9)}' | xargs -i bash -c "gpioset {}=0"
```

sh

0 表示低电平

5.9 配置 RTC

设备包含1个集成的RTC，出厂默认为自动同步读取系统时间，可以手动读取RTC时间和将系统时间写入RTC。

- 执行如下命令，手动读取RTC时间。

```
sudo hwclock -r
```

sh

- 执行如下命令，将系统时间写入RTC。

```
sudo hwclock -w
```

sh

5.10 配置串口

介绍RS232和RS485的配置方法。

5.10.1 安装picocom工具

在Linux环境下，可以通过picocom工具对串口RS232和RS485进行调试。

执行如下命令，安装picocom工具。

```
sudo apt-get install picocom
```

sh

5.10.2 配置 RS232

ED-HMI3120-185C包含2路RS232接口，其对应的COM口和设备文件，具体如下表：

RS232口数量	对应的COM口	对应设备文件
2	COM1, COM3	/dev/com1, /dev/com3

前提条件：

已完成ED-HMI3120-185C的RS232端口与外部设备的连接。

操作步骤：

1. 执行如下命令打开串口com1，并配置串口波特率为115200。

```
picocom -b 115200 /dev/com1
```

sh

2. 按需输入命令来控制外部设备。

5.10.3 配置 RS485

ED-HMI3120-185C包含2路RS485接口，其对应的COM口和设备文件，具体如下表：

RS485口数量	对应的COM口	对应设备文件
2	COM2, COM4	/dev/com2, /dev/com4

前提条件：

已完成ED-HMI3120-185C的RS485端口与外部设备的连接。

操作步骤：

1. 执行如下命令打开串口com4，并配置串口波特率为115200。

```
picocom -b 115200 /dev/com4
```

sh

2. 按需输入命令来控制外部设备。

5.11 配置音频

设备包含1路音频（支持接入MIC）和1路扩展的speaker输出，Master音量、MIC音量和Speaker音量均支持调节，且支持对MIC进行录音。

5.11.1 调节音量

支持手动调节MIC、Speaker和Master音量。

- 如果使用的是Desktop版本的系统，可直接通过桌面图标来调节MIC和Master的音量。

操作步骤：

1. 单击桌面右上角的  或 ，打开音量调节柱。

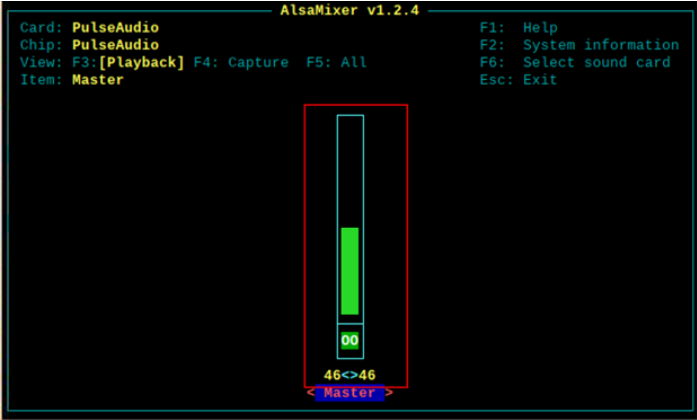


2. 上下拖动音量调节柱的按钮调节音量的大小；勾选或去勾选Mute复选框将音频静音和取消静音。
- Desktop和Lite版本的操作系统均支持通过命令行打开音量调节界面来调节音量。

操作步骤：

1. 在命令窗格执行如下命令，打开音量调节界面。

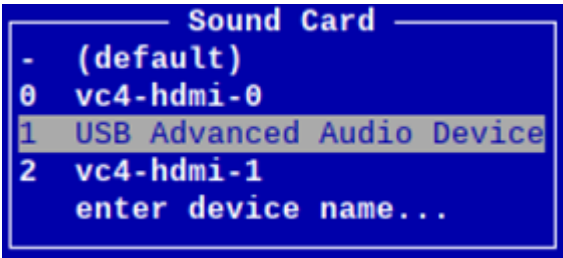
```
alsamixersh
```



2. 在打开的界面中查看当前Master的音量值，可通过键盘上的↑键和↓键来调节音量大小，按M键可以对音频进行静音和解除静音。

键盘按键	功能
↑	音量+
↓	音量-
M	静音或解除静音

3. 按F6打开“Sound Card”界面，选择“1 USB Advanced Audio Device”。



4. 按Enter，打开MIC和Speaker的音量界面，可通过键盘上的↑键和↓键来分别调节MIC和Speaker音量大小，按M键可以对MIC/Speaker进行静音和解除静音。

键盘按键	功能
↑	音量+
↓	音量-
M	静音或解除静音

5.11.2 配置录音

支持对MIC输入的音频进行录音。

前提条件：

音频接口已接入MIC。

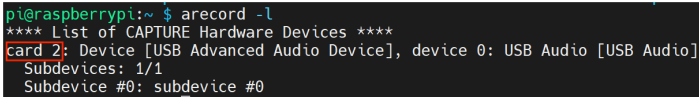
操作步骤：

1. 在命令窗格执行如下命令，打开音量调节界面。

```
alsamixer
```

sh

2. 按F6打开“Sound Card”界面，选择“1 USB Advanced Audio Device”，按Enter，打开MIC和Speaker的音量调节界面，确保MIC未静音。
3. 按Esc退出至命令窗格, 执行如下命令，查询表示录音的声卡设备的编号。



4. 获取编号后执行如下命令，开始录制名为audio1的音频，如下图所示。

```
arecord -fcd -Dhw:2 -c 1 --vumeter=mono audio1
```

sh

```
pi@raspberrypi:~$ arecord -fcd -Dhw:1 -c 1 --vumeter=mono audio1 | aplay -fcd -Dhw:1
Recording WAVE 'audio1': Signed 16 bit Little Endian, Rate 44100 Hz, Mono
##+ | 03%
```

参数	说明
cd	表示音频格式，支持设置为cd、cdr和dat等格式，按需设置。
audio1	表示录制的文件名，用户自定义即可。

5. 按Ctrl+C，关闭录音。
6. 执行如下命令，获取录音文件的保存路径。

```
pwd
```

sh

```
pi@raspberrypi:~ $ pwd
/home/pi
```

7. 在录音文件所在的目录下，执行如下命令播放音频。

```
aplay audio1
```

sh

```
pi@raspberrypi:~ $ aplay audio1
Playing WAVE 'audio1' : Signed 16 bit Little Endian, Rate 44100 Hz, Mono
```

5.12 配置 USER 指示灯

设备包含1个USER指示灯，用户可以根据实际需求自定义状态。

1. 依次执行如下命令，检测和安装gpio工具。

```
sudo apt update
sudo apt install gpio
```

sh

2. 执行如下命令，点亮和熄灭USER指示灯。

- 点亮USER指示灯

```
gpiofind USER_LED | awk '{print substr($0,9)}' | xargs -i bash -c "gpioset {}=0"
```

sh

0 表示管脚为高电平

- 熄灭USER指示灯

```
gpiofind USER_LED | awk '{print substr($0,9)}' | xargs -i bash -c "gpioset {}=1"
```

sh

1 表示管脚为低电平

6 安装操作系统（可选）

设备出厂时，默认带有操作系统。如果在使用过程中操作系统被损坏或者用户需要更换操作系统，则需要重新下载合适的系统镜像并进行烧录。我司支持通过先安装标准Raspberry Pi OS，再安装Firmware包，来实现操作系统的安装。

下文介绍下载镜像、烧录eMMC和安装Firmware包的具体操作。

6.1 下载镜像

可根据实际的需要下载对应的Raspberry Pi官方系统镜像，下载路径如下表：

OS	下载路径
Raspberry Pi OS (Desktop) 64-bit-bookworm (Debian 12)	https://downloads.raspberrypi.com/raspios_arm64/images/raspios_arm64-2024-11-19/2024-11-19-raspios-bookworm-arm64.img.xz (https://downloads.raspberrypi.com/raspios_arm64/images/raspios_arm64-2024-11-19/2024-11-19-raspios-bookworm-arm64.img.xz)
Raspberry Pi OS (Lite) 64-bit-bookworm (Debian 12)	https://downloads.raspberrypi.com/raspios_lite_arm64/images/raspios_lite_arm64-2024-11-19/2024-11-19-raspios-bookworm-arm64-lite.img.xz (https://downloads.raspberrypi.com/raspios_lite_arm64/images/raspios_lite_arm64-2024-11-19/2024-11-19-raspios-bookworm-arm64-lite.img.xz)

提示

我司工程师正在适配开发Raspberry Pi OS-trixie (Debian 13)的Firmware包，故暂时不支持Raspberry Pi OS-trixie (Debian 13)。建议使用Raspberry Pi OS 64-bit-bookworm (Debian 12)版本的操作系统。

6.2 烧录eMMC

建议使用Raspberry Pi官方烧录工具，下载路径如下：

- Raspberry Pi Imager : https://downloads.raspberrypi.org/imager/imager_latest.exe (https://downloads.raspberrypi.org/imager/imager_latest.exe)
- SD Card Formatter : <https://www.sdcardformatter.com/download/> (<https://www.sdcardformatter.com/download/>)
- Rpiboot : https://github.com/raspberrypi/usbboot/raw/master/win32/rpiboot_setup.exe (https://github.com/raspberrypi/usbboot/raw/master/win32/rpiboot_setup.exe)

前提条件：

- 已获取1台Windows PC，并完成烧录工具的下载和安装。

- 已准备一根Micro USB转USB-A连接线（USB烧录线）。

操作步骤：

提示

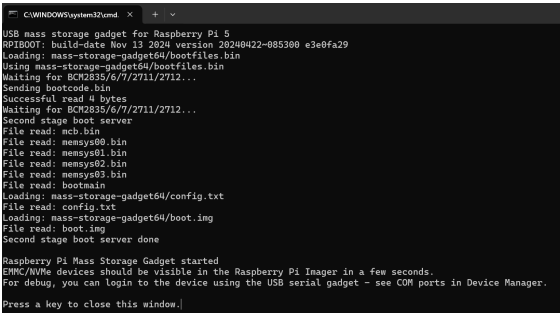
操作步骤以Windows系统为例进行说明。

1. 连接电源线和USB烧录线（Micro-USB转USB-A）。
 - 连接USB烧录线：一端连接设备侧的Micro USB接口，另一端连接PC上的USB接口。
 - 连接电源线：一端连接设备侧的DC 2-Pin凤凰端子，另一端连接外部电源。
2. 断开ED-HMI3120-185C电源，再重新上电。
3. 打开已安装的 `rpi-mass-storage-gadget64.bat` 工具（如下图红框内），自动进行盘符化。

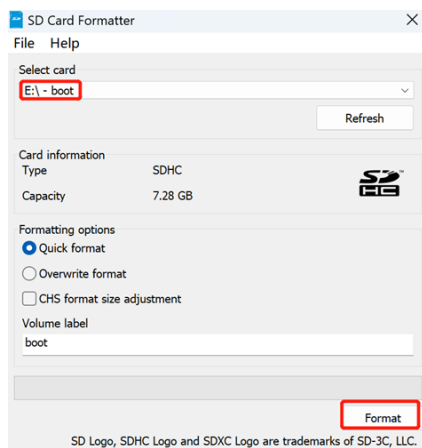
提示

`rpi-mass-storage-gadget64.bat` 工具位于Rpiboot的安装目录下。

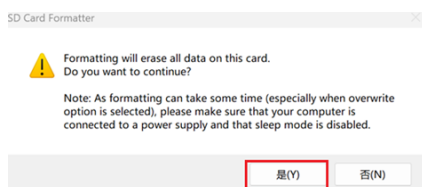
	cygwin1.dll	2024/9/26 23:06
	rpiboot.exe	2024/11/13 17:49
	rpi-mass-storage-gadget64.bat	2024/11/23 2:05
	Uninstall.exe	2025/2/10 19:01



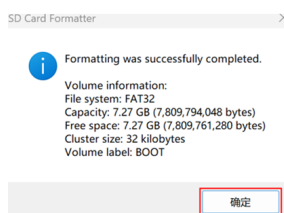
4. 待盘符化完成后，关闭 `rpi-mass-storage-gadget64.bat` 工具，电脑右下角会弹出盘符。
5. 打开 `SD Card Formatter`，选择被格式化的盘符，单击右下方“Format”进行格式化。



6. 在弹出的提示框中，单击“是”。

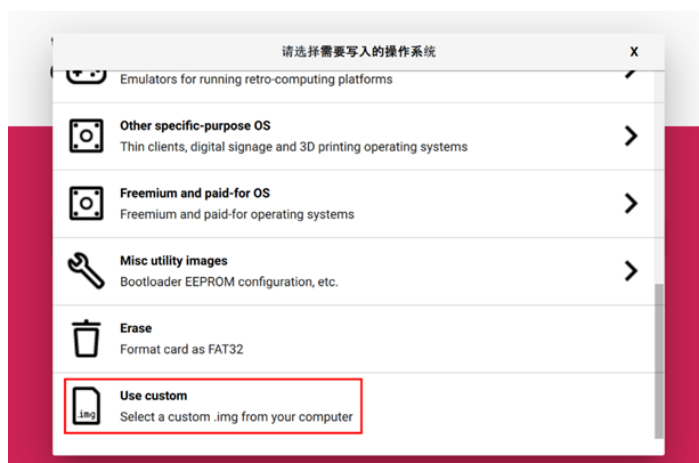


7. 格式化完成后，在提示框中单击“确定”。



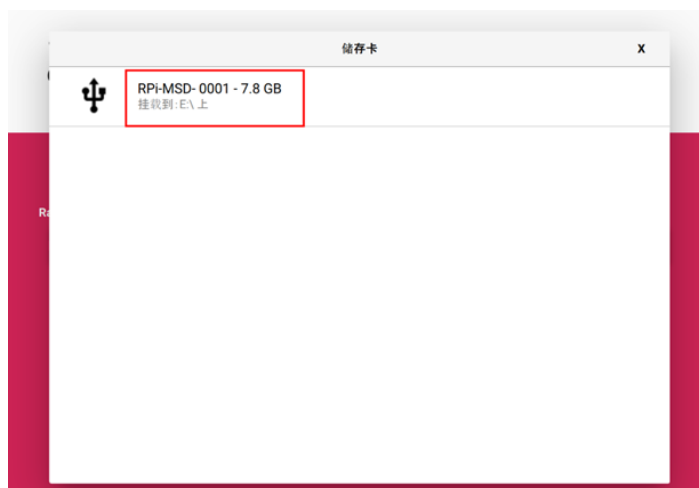
8. 关闭 SD Card Formatter。

9. 打开 Raspberry Pi Imager，单击“选择操作系统”，在弹出的窗格中选择“Use custom”。

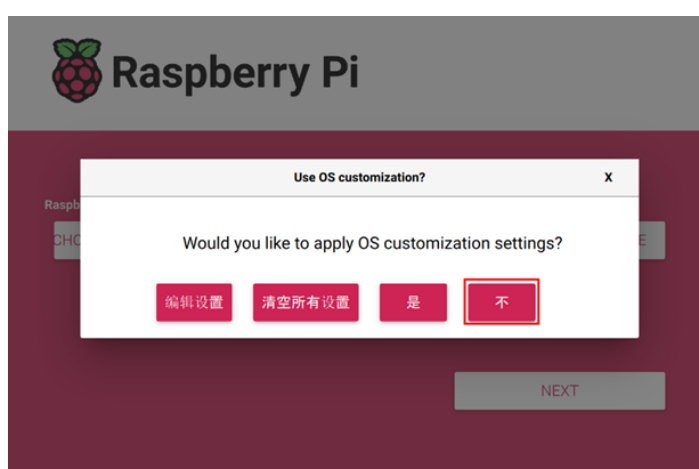


10. 根据提示，在自定义路径下选择已获取的镜像文件，并返回至烧录主界面。

11. 单击“选择SD卡”，在“存储卡”界面选择默认的SD卡，并返回至烧录主界面。



12. 单击“NEXT”，在弹出的“Use OS customization？”提示框中选择“不”，开始写入镜像。



13. 在弹出的“警告”提示框中选择“是”，开始写入镜像。



14. 待镜像写入完成后，会进行文件的验证。



15. 验证完成后，弹出“烧录成功”提示框，单击“继续”完成烧录。
16. 关闭 `Raspberry Pi Imager`，取下USB烧录线，重新给设备上电。

6.3 安装Firmware包

在ED-HMI3120-185C上烧录标准的Raspberry Pi OS后。需要通过添加edatec apt源和安装Firmware包来配置系统，使系统能够正常使用, 下文以Debian 12 (bookworm) 桌面版为例进行说明。

提示

我司工程师正在适配开发Raspberry Pi OS-trixie (Debian 13)的Firmware包，故暂时不支持Raspberry Pi OS-trixie (Debian 13)。建议使用Raspberry Pi OS 64-bit-bookworm (Debian 12)版本的操作系统。

前提条件：

- 已完成Raspberry Pi标准的bookworm镜像的烧录。
- 设备已正常启动，且已完成相关的启动配置。

操作步骤：

1. 设备正常启动后，在命令窗格依次执行如下命令，添加edatec apt源和安装Firmware包。

```
curl -s https://apt.edatec.cn/bsp/ed-install.sh | sudo bash -s hmi3120_185c
```

sh

2. 安装完成后，设备自动重启。
3. 执行如下命令，检查Firmware包是否安装成功。

```
dpkg -l | grep ed-
```

sh

下图中的结果表示Firmware包已安装成功。

```

pi@raspberrypi:~$ dpkg -l | grep ed-
ii  ed-base-bsp-v8                2:1.20250507.2      all      EDATec BSP for Raspberry Pi v8
ii  ed-ddcci-mib-tool             1.20250604.1        all      EDATec MIB Monitor ddcci tool
ii  ed-ipc3110-firmware           1.20241204.1        all      Firmware of EDATec Software Package
ii  ed-linux-image-base           1.20250102.1        all      EDATec linux base
ii  ed-lvd                         1.20250903.1        all      LVD detect service
ii  ed-qmi-tool                   1.20250913.2        all      EDATec Quectel QMI Software Package
ii  ed-reboot                     1.20250704.1        all      Add set-timeout-override.service to adjust systemd timeo
uts.
ii  ed-rtc                         1.20250620.1        all      RTC auto load and sync service for EDATec products power
ed by Raspberry Pi
ii  ed-touchscreen-test           1.20250731.1        all      EDATec touch screen test
ii  ed-usb-tools                   1.20250912          all      detect and auto reset usb
ii  libparted-fs-resize0:arm64     3.5-3               arm64    disk partition manipulator - shared FS resizing library
ii  libshine3:arm64               3.1.1-2             arm64    Fixed-point MP3 encoding library - runtime files
ii  shared-mime-info               2.2-1               arm64    FreeDesktop.org shared MIME database and spec
ii  usr-is-merged                 37~deb12u1          all      Transitional package to assert a merged-/usr system

```

提示

如果安装了错误的Firmware包，可以执行 `sudo apt-get --purge remove package` 进行删除，其中 `package` 为包的名字。