



ED-IPC1100

用户手册

by EDA Technology Co., Ltd

built: 2025-12-01

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-IPC1100是一款基于Raspberry Pi CM0的工业计算机，默认为512MB RAM，可选配8GB或16GB的eMMC。ED-IPC1100提供HDMI、USB、Ethernet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RTC，主要应用于工业控制和物联网领域。



1.2 包装清单

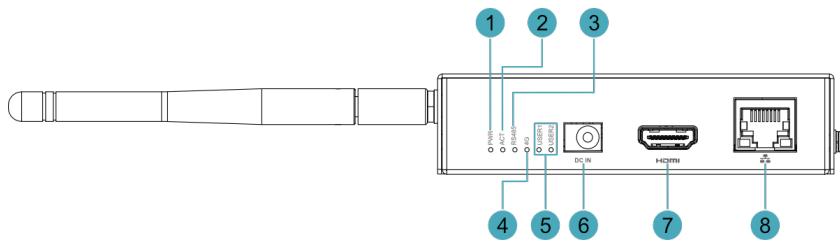
- 1 x ED-IPC1100主机
- 1 x 2.4GHz Wi-Fi/BT天线
- 1 x 4G天线
- 1 x 3-Pin凤凰端子

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

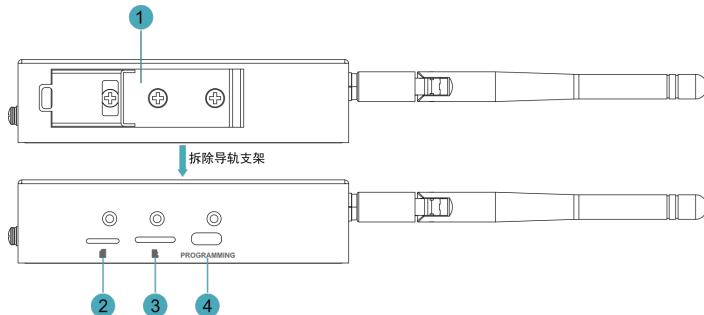
介绍前面板接口类型和定义。



编号	功能定义
1	1 x 电源指示灯 , 红色 , 用于查看设备上电状态。
2	1 x 系统状态指示灯 , 绿色 , 用于查看系统读写数据的状态。
3	1 x RS485指示灯 , 绿色 , 用于查看RS485串口的通信状态。
4	1 x 4G信号指示灯 , 绿色 , 用于查看4G信号的状态。
5	2 x 用户指示灯 , 绿色 , 用户可以根据实际需求自定义状态。
6	1 x DC输入 , DC Jack连接器 , 支持9V~28V输入。
7	1 x HDMI , Type-A接口 , 兼容HDMI 1.3a , 分辨率支持1080p 30Hz , 支持连接显示器。
8	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。

1.3.2 后面板

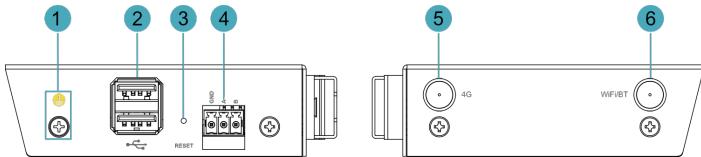
介绍后面板接口类型和定义。



编号	功能定义
1	1 x 导轨支架 , 通过支架将ED-IPC1100主机安装在DIN导轨上。
2	1 x Nano SIM卡槽 , 用于安装获取4G信号的Nano SIM卡。
3	1 x Micro SD卡槽 , 支持安装Micro SD卡 , 用于启动系统。 注 : 仅当ED-IPC1100使用CMOLite时 , 支持从Micro SD卡启动系统。
4	1 x Type-C USB接口 , 支持通过此接口对系统进行eMMC烧录。

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 接地点 , 连接外部电源的大地。
2	2 x USB 2.0 , 双层Type-A接口 , 每一路最高支持480Mbps传输速率。
3	1 x 复位按键 , 隐藏式按键 , 按下按键可重新启动设备。
4	1 x RS485接口 , 3-Pin 3.5mm间距凤凰端子 , 用于连接第三方控制设备。
5	1 x 4G天线接口 , SMA接口 , 连接4G天线。
6	1 x Wi-Fi/BT天线接口 , SMA接口 , 连接Wi-Fi/BT天线

1.4 按键

ED-IPC1100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-IPC1100系列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
USER1和USER2	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	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

指示灯	状态	描述
以太网口黄色指示灯	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
RS485	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-IPC1100系列设备包含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，用于启动系统。

提示

- 仅当ED-IPC1100使用CM0Lite时，支持从Micro SD卡启动系统。
- 当ED-IPC1100使用8GB或16GB eMMC的CM0时，Micro SD卡槽为无效接口。

1.6.1.2 Nano SIM卡槽

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-IPC1100设备包含1路电源输入接口，使用DC Jack连接器，接口丝印为“DC IN”，支持9V~28V的电源输入。

提示

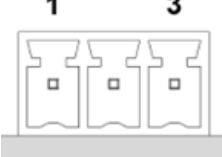
推荐使用12V 2A的电源适配器。

1.6.3 RS485接口

ED-IPC1100设备包含1路RS485接口，使用3-Pin 3.5mm间距的凤凰端子，接口丝印为“GND/A/B”，支持连接第三方控制器。

引脚定义

端子引脚定义如下：

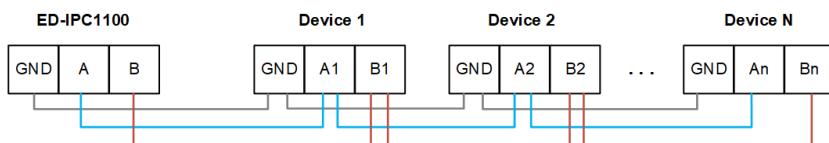
	Pin ID	Pin Name
1		GND
2		RS485-A
3		RS485-B

其中RS485接口对应CM0的管脚名称如下：

Signal	CM0 Pin Out
RS485-A	GPIO14
RS485-B	GPIO15

连接线缆

RS485接线示意图如下：



RS485端电阻配置

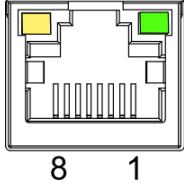
ED-IPC1100设备包含1路RS485接口，在RS485线路的A和B之间预留120R跨接电阻，插入跳线帽可使能该跨接电阻。默认状态下未连接跳线帽，120R端接电阻功能失效。跨接电阻在PCBA中的位置为J10。

提示

需要打开设备外壳才能查看120R跨接电阻的位置。

1.6.4 100M以太网接口

ED-IPC1100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	-

1.6.5 HDMI接口

ED-IPC1100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1080p30的视频输出。

1.6.6 USB 2.0接口

ED-IPC1100设备包含2路USB 2.0接口，接口丝印为“”，标准的双层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。

1.6.7 Type-C USB接口

ED-IPC1100设备包含1路Type-C 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。

1.6.8 天线接口

ED-IPC1100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。

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

ED-IPC1100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。

提示

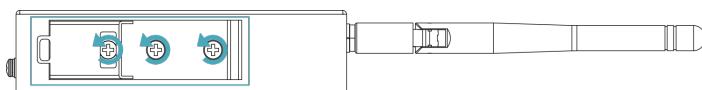
- 仅当ED-IPC1100使用CM0Lite时，支持从Micro SD卡启动系统。
- 当ED-IPC1100使用8GB或16GB eMMC的CM0时，Micro SD卡槽为无效接口。
- Micro SD卡不支持带电插拔。

前提条件：

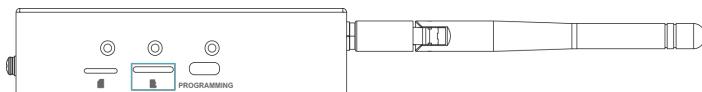
- 已获取待使用的Micro SD卡。
- 已断开设备电源

操作步骤：

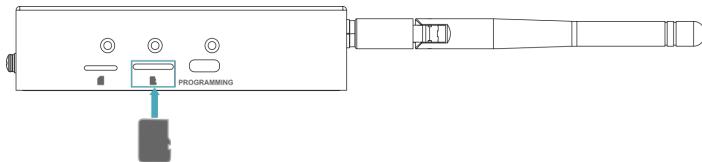
1. 使用十字螺丝刀逆时针拧下DIN导轨支架上的3颗螺钉，将默认的DIN导轨支架拆除。



2. 确定设备侧Micro SD卡槽的位置，如下图标注位置所示。



3. 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。



4. 将导轨支架安装至设备上。

2.3 安装Nano SIM卡

ED-IPC1100设备默认包含4G功能，则在使用4G功能前需要先安装Nano SIM卡。

注意

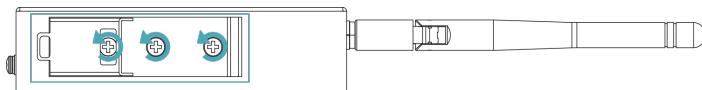
Nano SIM卡不支持带电插拔。

前提条件：

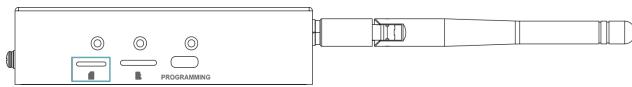
- 已获取待使用的4G Nano SIM卡。
- 已断开设备电源。

操作步骤：

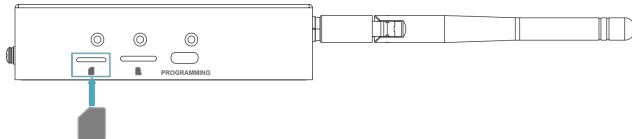
1. 使用十字螺丝刀逆时针拧下DIN导轨支架上的3颗螺钉，将默认的DIN导轨支架拆除。



2. 确定设备侧Nano SIM卡槽的位置，如下图标标注位置所示。



3. 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。



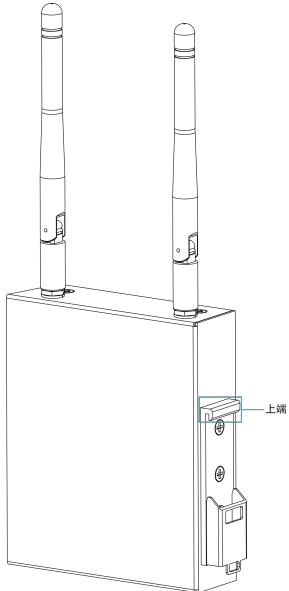
4. 将导轨支架安装至设备上。

3 安装设备

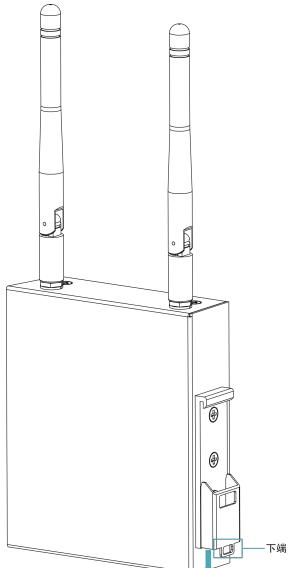
ED-IPC1100支持DIN导轨安装，出厂默认标配并已安装DIN导轨支架。

操作步骤：

1. 将设备的带导轨支架侧对着待安装的轨道，将支架的上端套在DIN导轨的上沿。



2. 用手握住支架的下端，沿着箭头方向向下拉，使支架完全卡入DIN导轨上。



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

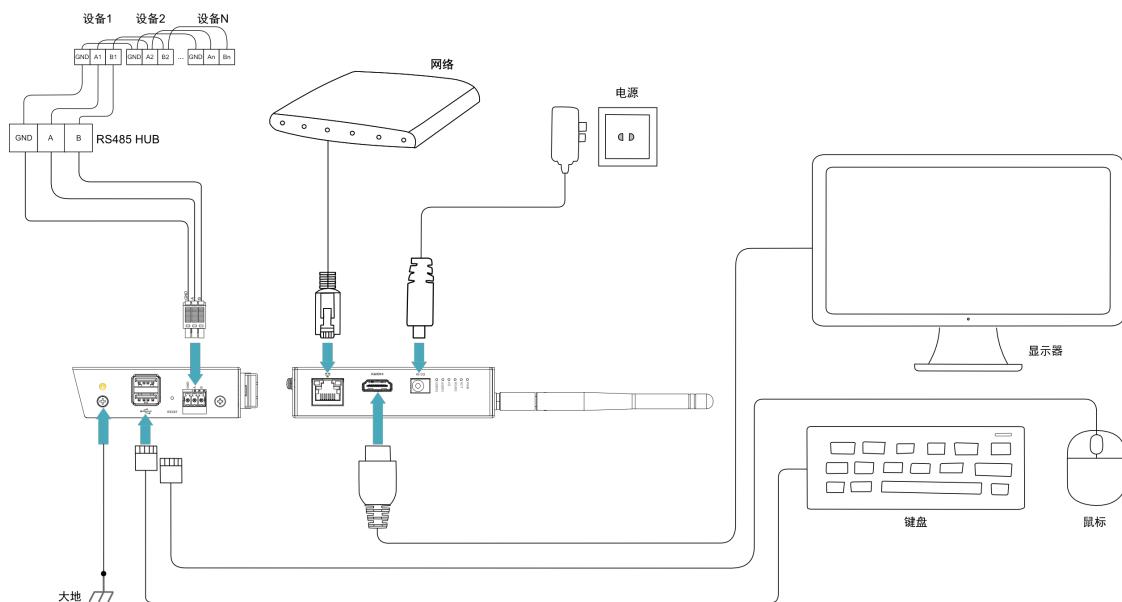
介绍线缆的连接方法。

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-IPC1100系列设备无电源开关，接入电源后，系统将会开始启动。

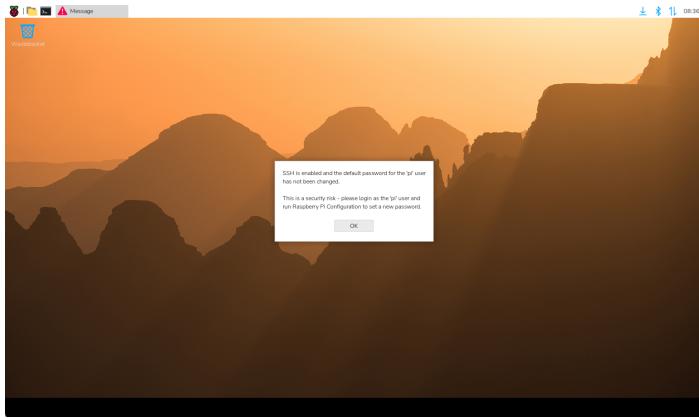
- 红色PWR灯点亮，表示设备已正常供电。
 - 绿灯ACT闪烁，表示系统正常启动，然后屏幕的左上角会出现Raspberry Pi 的logo。

提示

默认用户名 : `pi` ; 默认密码 : `raspberry` 。

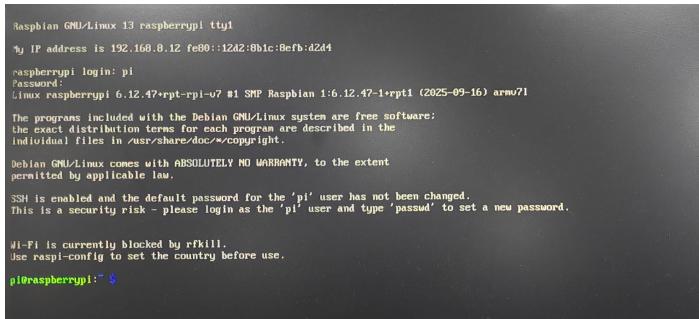
4.2.1 Raspberry Pi OS (Desktop)

如果产品在出厂时安装的是Desktop版系统，则设备启动完成后，直接进入到桌面，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 `raspberry`，下图所示表示系统已正常启动。



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

查找设备IP

5.2 远程登录

远程登录

5.3 配置存储设备

配置存储设备

5.4 配置以太网 IP

配置以太网IP

5.5 配置Wi-Fi

配置Wi-Fi

5.6 配置蓝牙

配置蓝牙

5.7 配置 4G

ED-IPC1100设备出厂默认已开启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。

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：

- ED-IPC1100设备已正常启动。
- 包含4G网络的Nano SIM已正确安装至ED-IPC1100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 (usb0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::732e:31bc:ae26:f477 prefixlen 64 scopeid 0x20<link>
          ether 00:09:a1:b0:52:24 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

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 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

usb0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 10.191.34.79 netmask 255.0.0.0 broadcast 10.255.255.255
        inet6 fe80::c704:00cc:227f:72b6 prefixlen 64 scopeid 0x20<link>
          ether ae:0c:29:a3:9b:6d 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

wlan0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
        ether 8c:1f:64:34:a0:04 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
```

- 若返回信息中usb0接口显示了具体的IP地址，则表示4G已连接。
- 若返回信息中usb0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-ec800m.service

sh

返回信息如下图：

```
pi@raspberrypi: $ sudo systemctl status ed-lte-daemon.service
● ed-lte.daemon.service - EDATEC QM1 Reconnect service
   Loaded: loaded (/lib/systemd/system/ed-lte-daemon.service; enabled; preset: enabled)
   Active: active (running) since Fri 2025-09-19 03:24:01 BST; 42min ago
     Main PID: 541 (ed-lte-tool)
      Tasks: 7 (limit: 178)
        CPU: 2.298s
       CGroup: /system.slice/ed-lte-daemon.service
           └─541 /usr/local/bin/ed-lte-tool --daemon
              ├─2911 quiclet-CM -4

Sep 19 04:06:00 raspberrypi ed-lte-tool[541]: [2025-09-19 04:06:00][INFO] [cmd] AT< OK
Sep 19 04:06:00 raspberrypi ed-lte-tool[541]: [2025-09-19 04:06:00][INFO] [cmd] AT> AT+QNETDEVCTL?
Sep 19 04:06:00 raspberrypi ed-lte-tool[541]: [2025-09-19 04:06:00][INFO] [cmd] AT+QNETDEVCTL: 1,1,1,
Sep 19 04:06:00 raspberrypi ed-lte-tool[541]: [2025-09-19 04:06:00][INFO] [cmd] AT< OK
Sep 19 04:06:00 raspberrypi ed-lte-tool[541]: [2025-09-19 04:06:00][INFO] [cmd] AT> AT+CGACT?
Sep 19 04:06:00 raspberrypi ed-lte-tool[541]: [2025-09-19 04:06:00][INFO] [cmd] AT+CGACT: 1,1
Sep 19 04:06:00 raspberrypi ed-lte-tool[541]: [2025-09-19 04:06:00][INFO] [cmd] AT< OK
Sep 19 04:06:00 raspberrypi ed-lte-tool[541]: [2025-09-19 04:06:00][INFO] [cmd] AT> AT+CGPADDR=1
Sep 19 04:06:00 raspberrypi ed-lte-tool[541]: [2025-09-19 04:06:00][INFO] [cmd] AT+CGPADDR: 1, "10.191.34.79"
Sep 19 04:06:00 raspberrypi ed-lte-tool[541]: [2025-09-19 04:06:00][INFO] [cmd] AT< OK
```

- 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：

- ED-IPC1100设备已正常启动。
- 包含4G网络的Nano SIM已正确安装至ED-IPC1100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
ed-qml.conf
reset_pin=0
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=usb0
# interval time(Second) and greater than 0, Default 60s
# intervaltime=60
dev=/dev/ttyUSB2
# Use Quectel auto dial: 1~enable(default), 0~disable
quectel_auto_dial=1
```

提示

“Network”中的“ping_server”和“online_script”也支持用户按需配置。

- 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。
- 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 (usb0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::732e:31bc:ae26:f477 prefixlen 64 scopeid 0x20<link>
            ether 00:0e:0a:1b:05:24 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

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 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

usb0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.191.34.79 netmask 255.0.0.0 broadcast 10.255.255.255
        inet6 fe80::c704:40cc:227f:72b6 prefixlen 64 scopeid 0x20<link>
            ether ae:0c:29:a3:9b:6d 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

wlan0: flags=4099<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    ether 8c:1f:64:34:a0:04 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
```

- 若返回信息中usb0接口显示了具体的IP地址，则表示4G已连接。
- 若返回信息中usb0接口未显示具体的IP地址，则表示4G未连接。

- 执行如下命令，查询4G的服务状态。

`sudo systemctl status ed-ec800m.service`

sh

返回信息如下图：

```
pi@raspberrypi:~ $ sudo systemctl status ed-lte-daemon.service
● ed-lte-daemon.service - EDATEC OML Reconnect service
   Loaded: loaded (/lib/systemd/system/ed-lte-daemon.service; enabled; preset: enabled)
   Active: active (running) since Fri 2025-09-19 03:24:01 BST; 42min ago
     Main PID: 541 (ed-lte-tool)
    Tasks: 7 (limit: 178)
   CGroup: /system.slice/ed-lte-daemon.service
           └─ 541 /usr/local/bin/ed-lte-tool --daemon
               ├─2911 quectel-CM -4

Sep 19 04:06:00 raspberrypi ed-lte-tool[541]: [2025-09-19 04:06:00][INFO] [cmd] AT< OK
Sep 19 04:06:00 raspberrypi ed-lte-tool[541]: [2025-09-19 04:06:00][INFO] [cmd] AT> AT&QNETDEVCTL?
Sep 19 04:06:00 raspberrypi ed-lte-tool[541]: [2025-09-19 04:06:00][INFO] [cmd] AT> AT&QNETDEVCTL: 1,1,1
Sep 19 04:06:00 raspberrypi ed-lte-tool[541]: [2025-09-19 04:06:00][INFO] [cmd] AT< OK
Sep 19 04:06:00 raspberrypi ed-lte-tool[541]: [2025-09-19 04:06:00][INFO] [cmd] AT> AT+CGACT?
Sep 19 04:06:00 raspberrypi ed-lte-tool[541]: [2025-09-19 04:06:00][INFO] [cmd] AT< +CGACT: 1,1
Sep 19 04:06:00 raspberrypi ed-lte-tool[541]: [2025-09-19 04:06:00][INFO] [cmd] AT> OK
Sep 19 04:06:00 raspberrypi ed-lte-tool[541]: [2025-09-19 04:06:00][INFO] [cmd] AT> AT+CGPADDR=1
Sep 19 04:06:00 raspberrypi ed-lte-tool[541]: [2025-09-19 04:06:00][INFO] [cmd] AT< +CGPADDR: 1,"10.191.34.79"
Sep 19 04:06:00 raspberrypi ed-lte-tool[541]: [2025-09-19 04:06:00][INFO] [cmd] AT> OK
```

- 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
ifconfig	<p>查询4G网络是否连接，其中usb0接口表示4G接口，如下图所示。</p> <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 broadcast 192.168.8.255 scopeid 0x20<link> ether 00:0d:9a:1b:05:24 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 lo: flags=256<UP,LOOPBACK,RUNNING> mtu 65536 inet 127.0.0.1 netmask 255.0.0.0 broadcast 127.0.0.1 scopeid 0x10<host> loop txqueuelen 1000 (Local Loopback) 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 usb0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500 inet 10.191.34.79 netmask 255.0.0.0 broadcast 10.255.255.255 broadcast 10.255.255.255 scopeid 0x10<link> ether 08:00:23:9c:90:00 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 wlan0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500 ether 8c:1f:64:34:a0:0e 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</pre> <ul style="list-style-type: none"> 若usb0接口显示了具体的IP地址，则表示4G已连接。 若usb0接口未显示具体的IP地址，则表示4G未连接。
sudo systemctl status ed-ec800m.service	<p>查询4G的服务状态，如下图所示。</p> <pre>pi@raspberrypi:~ \$ sudo systemctl status ed-lte-demon.service ● ed-lte-demon.service - ED-TEC QM Reconnect Service Loaded: loaded /lib/systemd/system/ed-lte-demon.service; enabled; preset: enabled Active: active (running) since Fri 2025-09-19 03:24:01 BST; 42min ago Main PID: 541 (ed-lte-tool) Tasks: 7 (limit: 178) CPU: 0.000 CPU(s) CGroup: /system.slice/ed-lte-demon.service └─ 541 /usr/local/bin/ed-lte-tool --daemon Sep 19 04:06:00 raspberrypi ed-lte-tool[541]: [2025-09-19 04:06:00][INFO] [cmd] AT<- ZX Sep 19 04:06:00 raspberrypi ed-lte-tool[541]: [2025-09-19 04:06:00][INFO] [cmd] AT> AT&NETDEVCTL? Sep 19 04:06:00 raspberrypi ed-lte-tool[541]: [2025-09-19 04:06:00][INFO] [cmd] AT<- +NETDEVCTL: 1,1,1,1 Sep 19 04:06:00 raspberrypi ed-lte-tool[541]: [2025-09-19 04:06:00][INFO] [cmd] AT> AT&GACT? Sep 19 04:06:00 raspberrypi ed-lte-tool[541]: [2025-09-19 04:06:00][INFO] [cmd] AT<- +GACT: 1,1 Sep 19 04:06:00 raspberrypi ed-lte-tool[541]: [2025-09-19 04:06:00][INFO] [cmd] AT> AT&CGACT? Sep 19 04:06:00 raspberrypi ed-lte-tool[541]: [2025-09-19 04:06:00][INFO] [cmd] AT<- +CGACT: 1,1 Sep 19 04:06:00 raspberrypi ed-lte-tool[541]: [2025-09-19 04:06:00][INFO] [cmd] AT> AT&CGPADDR=1 Sep 19 04:06:00 raspberrypi ed-lte-tool[541]: [2025-09-19 04:06:00][INFO] [cmd] AT<- +CGPADDR: 1,"10.191.34.79" Sep 19 04:06:00 raspberrypi ed-lte-tool[541]: [2025-09-19 04:06:00][INFO] [cmd] AT> AT&CGPADDR=1 Sep 19 04:06:00 raspberrypi ed-lte-tool[541]: [2025-09-19 04:06:00][INFO] [cmd] AT<- +CGPADDR: 1,"10.191.34.79"</pre> <ul style="list-style-type: none"> 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-ec800m.service	使能4G服务
sudo systemctl start ed-ec800m.service	开启4G服务
sudo systemctl stop ed-ec800m.service	关闭4G服务
pinctrl set 0 op dh && sleep 0.3 && pinctrl set 0 op dl	中国版4G模块复位(4G模块的复位引脚为GPIO0)
pinctrl set 0 op dh && sleep 0.1 && pinctrl set 1 op dh && sleep 0.2 && pinctrl set 0 op dl && sleep 0.5 && pinctrl set 1 op dl	欧洲版和美国版4G模块复位(4G模块的复位引脚为GPIO0)

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，可依次使能4G服务和开启4G服务。

5.8 配置 RTC

配置RTC

5.8 配置串口

介绍RS485的配置方法。

5.8.1 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

5.8.2 配置 RS485

ED-IPC1100包含1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
1	RS485	/dev/rs485

提示

RS485通信支持2400~115200的波特率。

前提条件：

已完成ED-IPC1100的RS485端口与外部设备的连接。

操作步骤：

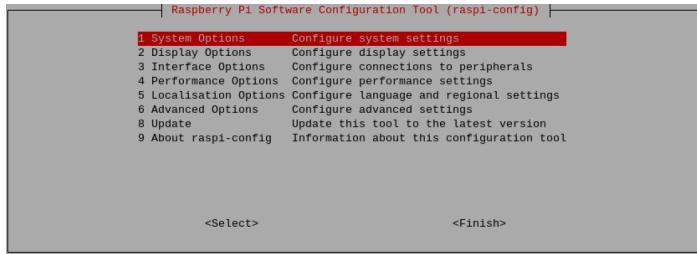
1. 关闭串口登录模式。

提示

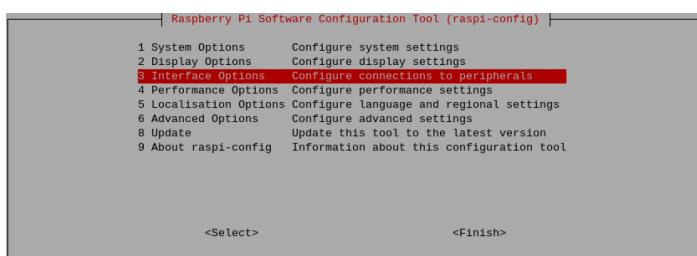
ED-IPC1100设备的RS485端口默认为串口登录模式，在通过串口发送和接收数据前，需要先关闭串口登录模式。

a. 打开命令窗格，执行如下命令，打开配置工具界面。

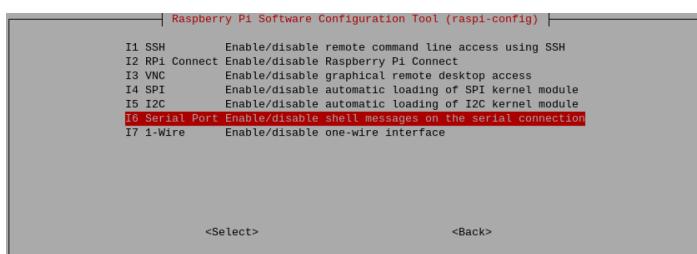
```
sh
sudo raspi-config
```



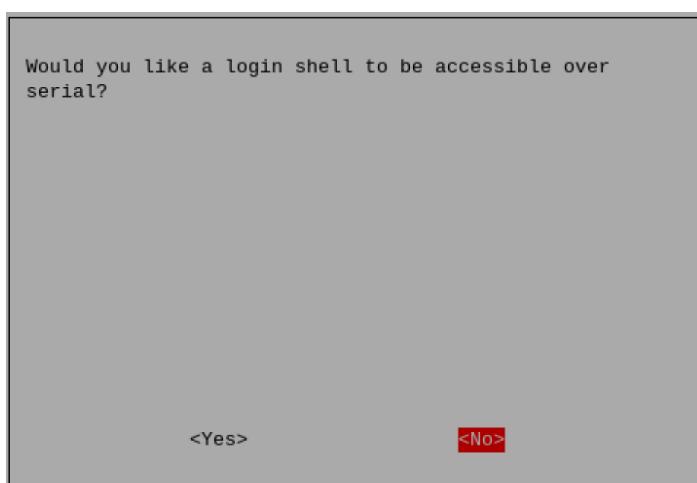
b. 在菜单中选择“Interface Options”，再按“Enter”。



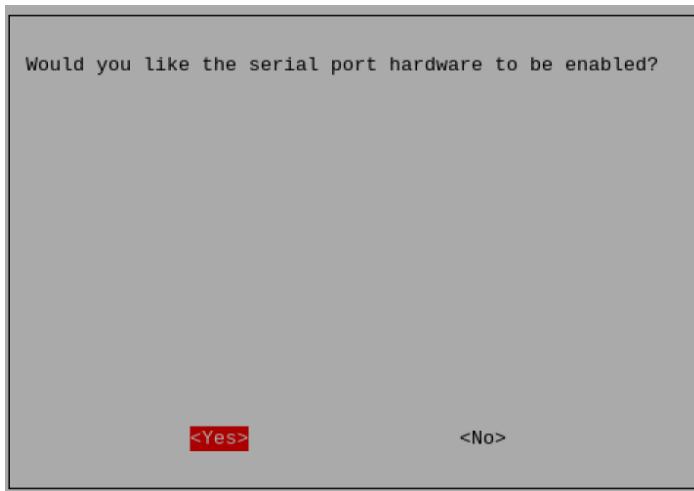
c. 在菜单中选择“Serial Port”，再按“Enter”。



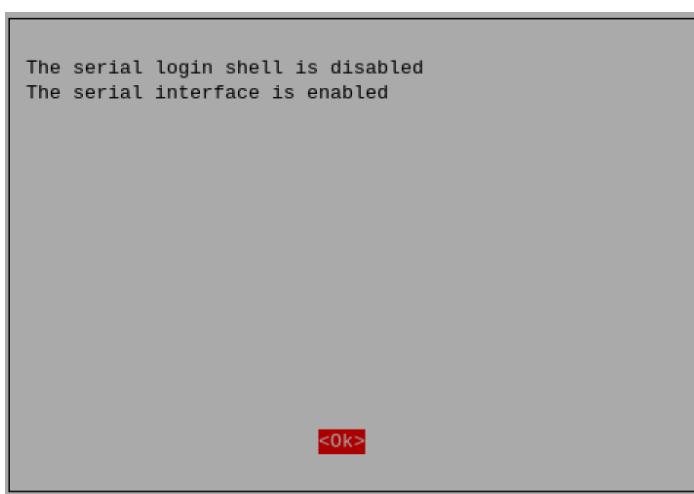
d. 在提示框中选择“No”，再按“Enter”。



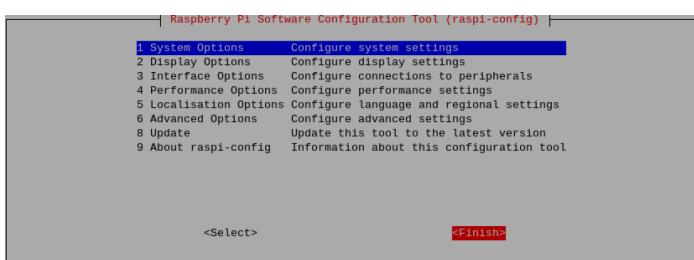
e. 在提示框中选择“Yes”，再按“Enter”。



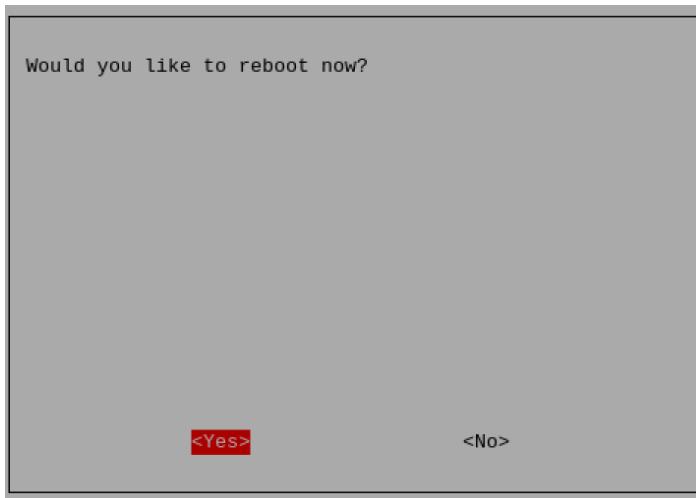
f. 按“Enter”，确认设置。



f. 在配置工具界面选择“Finish”，再按“Enter”。



g. 在提示框中选择“Yes”，再按“Enter”，重启设备。



2. 执行如下命令打开串口rs485，并配置串口波特率为115200。

```
picocom -b 115200 /dev/rs485
```

3. 按需输入命令来控制外部设备。

5.9 配置 USER 指示灯

ED-IPC1100包含2个USER接口，分别为USER1和USER2，均通过GPIO来控制，其对应的GPIO引脚如下表：

USER指示灯	对应的GPIO
USER1	GPIO5
USER2	GPIO6

- 点亮USER1指示灯

```
pinctrl set 5 op dl
```

- 点亮USER2指示灯

```
pinctrl set 6 op dl
```

- 熄灭USER1指示灯

```
pinctrl set 5 op dh
```

sh

- 熄灭USER2指示灯

```
pinctrl set 6 op dh
```

sh

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。

下文介绍镜像下载、eMMC烧录和安装Firmware包的具体操作。

6.1 镜像下载

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-trixie (Debian 13)	https://downloads.raspberrypi.com/raspios_arm64/images/raspios_arm64-2025-10-02/2025-10-01-raspios-trixie-arm64.img.xz (https://downloads.raspberrypi.com/raspios_arm64/images/raspios_arm64-2025-10-02/2025-10-01-raspios-trixie-arm64.img.xz)
Raspberry Pi OS(Lite) 64-bit-trixie (Debian 13)	https://downloads.raspberrypi.com/raspios_lite_arm64/images/raspios_lite_arm64-2025-10-02/2025-10-01-raspios-trixie-arm64-lite.img.xz (https://downloads.raspberrypi.com/raspios_lite_arm64/images/raspios_lite_arm64-2025-10-02/2025-10-01-raspios-trixie-arm64-lite.img.xz)
Raspberry Pi OS/Desktop) 32-bit-trixie (Debian 13)	https://downloads.raspberrypi.com/raspios_armhf/images/raspios_armhf-2025-10-02/2025-10-01-raspios-trixie-armhf.img.xz (https://downloads.raspberrypi.com/raspios_armhf/images/raspios_armhf-2025-10-02/2025-10-01-raspios-trixie-armhf.img.xz)
Raspberry Pi OS(Lite) 32-bit-trixie (Debian 13)	https://downloads.raspberrypi.com/raspios_lite_armhf/images/raspios_lite_armhf-2025-10-02/2025-10-01-raspios-trixie-armhf-lite.img.xz (https://downloads.raspberrypi.com/raspios_lite_armhf/images/raspios_lite_armhf-2025-10-02/2025-10-01-raspios-trixie-armhf-lite.img.xz)

提示

推荐使用Lite版本的镜像，最新版本的镜像请参考Raspberry Pi官网：[Raspberry Pi OS](https://www.raspberrypi.com/software/operating-systems/) (<https://www.raspberrypi.com/software/operating-systems/>)。

6.2 eMMC烧录

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)

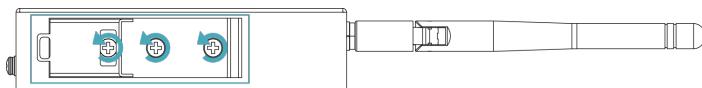
前提条件：

- 已完成烧录工具的下载，并安装至电脑。
- 已准备一根Type-C USB转USB-A线（USB烧录线）。
- 已获取待烧录的镜像文件。

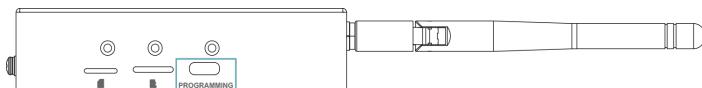
操作步骤：

操作步骤以Windows系统为例进行说明。

1. 使用十字螺丝刀逆时针拧下DIN导轨支架上的3颗螺钉，将默认的DIN导轨支架拆除。



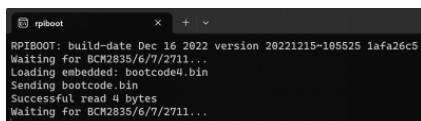
2. 确定设备侧Type-C USB接口的位置，如下图标标注位置所示。



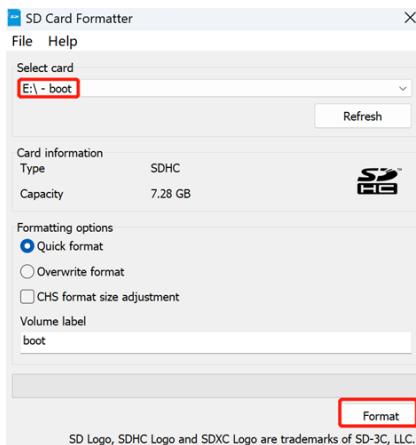
3. 连接好电源线和USB烧录线（Type-C USB转USB-A）。

- 连接USB烧录线：一端连接设备侧的Type-C 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 Jack端子，另一端连接外部电源。

4. 断开ED-IPC1100电源，再重新上电。
5. 打开已安装的rpiboot工具，自动进行盘符化。



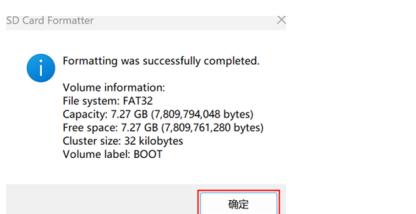
6. 待盘符化完成后，电脑右下角会弹出盘符。
7. 打开 **SD Card Formatter**，选择被格式化的盘符，单击右下方“Format”进行格式化。



8. 在弹出的提示框中，单击“是”。

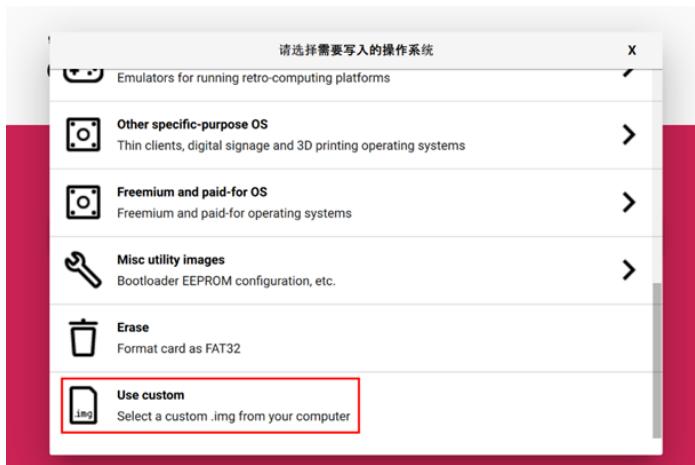


9. 格式化完成后，在提示框中单击“确定”。



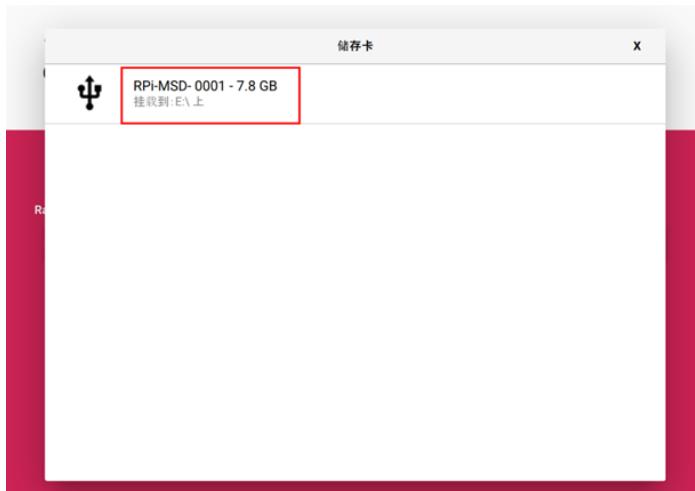
10. 关闭 SD Card Formatter。

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

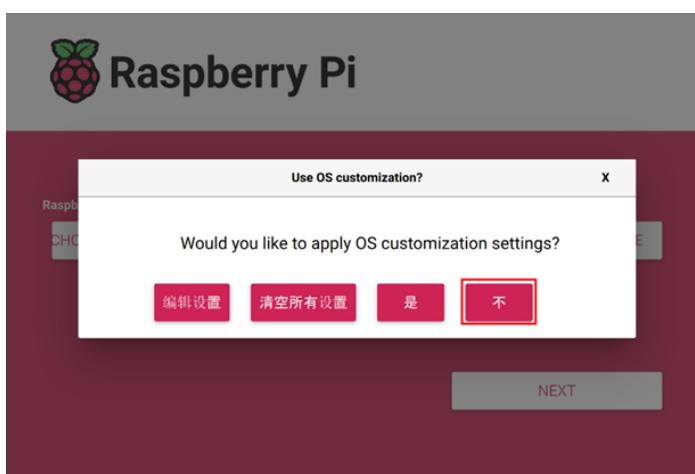


12. 根据提示，在自定义路径下选择已获取的镜像文件，并返回至烧录主界面。

13. 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。



14. 单击“NEXT”，在弹出的“Use OS customization？”提示框中选择“不”。



15. 在弹出的“警告”提示框中选择“确认”，开始写入镜像。



16. 待镜像写入完成后，会进行文件的验证。



17. 验证完成后，弹出“烧录成功”提示框，单击“继续”完成烧录。



18. 关闭 `Raspberry Pi Imager`，取下USB连接线，重新给设备上电。

6.3 安装Firmware包

在ED-IPC1100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 13 (trixie) Lite版为例进行说明。

前提条件：

- 已完成Raspberry Pi标准的trixie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 ipc1100
```

sh

```

pi@raspberrypi: ~ $ curl -s https://apt.edatec.cn/ed/install.sh | sudo bash -s > spc1100
  % Total    % Received % Xferd  Average Speed   Time   Time  Current
  100  406  100  406    0     0  Load  Upload Total  Spent   Left  Speed
--:--:-- 2025-09-28 11:25:24-- https://apt.edatec.cn/ed/install.sh
Resolving apt.edatec.cn (apt.edatec.cn) ... 47.242.199.148
Connecting to apt.edatec.cn (apt.edatec.cn) 47.242.199.148:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 36089 (35K) [image/png]
Saving to: '/tmp/eda-common/eda/splash.png'

/tmp/eda-common/eda/splash.png 100%[=====] 35.17K --KB/s in 0.02s
2025-09-28 11:25:25 (1.40 MB/s) - '/tmp/eda-common/eda/splash.png' saved [36089/36089]

--:--:-- 2025-09-28 11:25:25-- https://apt.edatec.cn/pubkey.gpg
Resolving apt.edatec.cn (apt.edatec.cn) ... 47.242.199.148
Connecting to apt.edatec.cn (apt.edatec.cn) 47.242.199.148:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1635 (1.6K) [application/octet-stream]
Saving to: '/tmp/eda-common/eda/edatec.gpg'

/tmp/eda-common/eda/edatec.gpg 100%[=====] 1.60K --KB/s in 0s
2025-09-28 11:25:25 (9.99 MB/s) - '/tmp/eda-common/eda/edatec.gpg' saved [1635/1635]

deb https://apt.edatec.cn/raspbian stable main
deb https://archive.raspberrypi.com/raspbian stable InRelease
Hit:1 https://apt.edatec.cn (apt.edatec.cn) 47.242.199.148 InRelease
Hit:2 https://apt.edatec.cn (apt.edatec.cn) 47.242.199.148 InRelease
Hit:3 http://deb.debian.org/debian-security bookworm-security InRelease
Hit:4 http://deb.debian.org/debian-security bookworm-updates InRelease
Hit:5 http://archive.raspberrypi.com/debian bookworm InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ii  packages can be upgraded. Run 'apt list --upgradable' to see them.

Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ed-sbc1100-firmware is already the newest version (1.20250913.1).
ed-qmi-tool is already the newest version (1.20250913.2).
ed-usb-tools is already the newest version (1.20250912).
ed-lvd is already the newest version (1.20250912).
ed-lvd-19 is already the newest version (1.20250903.1).

```

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-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
ttmeouts.
ii  ed-rtc                           1.20250620.1          all      RTC auto load and sync service for EDATEC products
powered by Raspberry Pi
ii  ed-sbc1100-firmware               1.20250913.1          all      Firmware of EDATEC Software Package
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 lib
rary
ii  libshine32: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
pi@raspberrypi: ~ $ 

```

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。