



ED-IPC3400

用户手册

by EDA Technology Co., Ltd

built: 2026-07-13

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-IPC3400是一款基于Raspberry Pi CM5的工业计算机，根据不同的应用场景和用户需求，可选择不同规格的RAM、eMMC和SD卡的计算机系统。

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

ED-IPC3400配备HDMI、USB 2.0/3.0 以及2.5G+1G双网口，支持Wi-Fi与有线网络接入，并集成了RTC、EEPROM与加密芯片，显著提升了系统在工业控制与物联网应用中的易用性与可靠性；同时内置 M.2 M Key 扩展接口，可选配128GB/256GB/512GB SSD或Hailo-8 AI 模块（13 TOPS），可高效支撑机器视觉等高算力工业场景。



1.2 包装清单

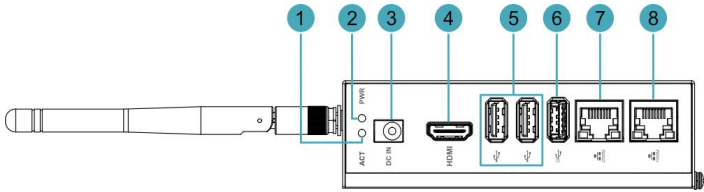
- 1 x ED-IPC3400主机
- [选配Wi-Fi/BT版本] 1 x 2.4GHz/5GHz Wi-Fi/BT天线

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

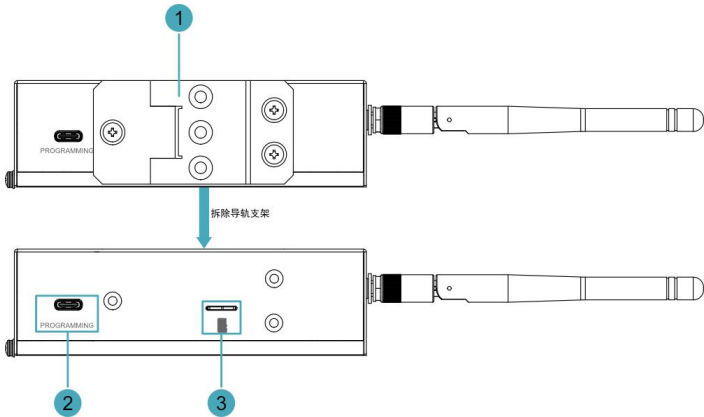
介绍前面板接口类型和定义。



编号	功能定义
1	1 x 系统状态指示灯，绿色，用于查看系统读写数据的状态。
2	1 x 电源指示灯，红色，用于查看设备上电状态。
3	1 x DC输入，DC Jack（5.5x2.5）连接器，支持9V~28V输入。
4	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。
5	2 x USB 2.0，Type-A接口，每一路最高支持480Mbps传输速率。
6	1 x USB 3.0，Type-A接口，最高支持5Gbps传输速率。
7	1 x 2500M以太网接口，RJ45接口，带有LED灯，10/100/1000M/2500M自适应接口，用于接入以太网。
8	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。

1.3.2 后面板

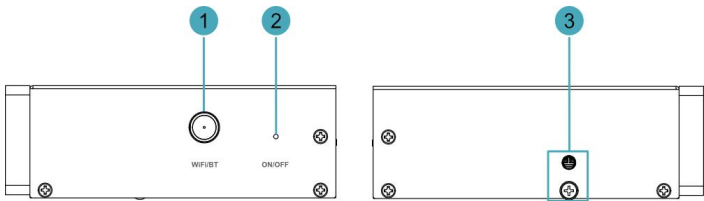
介绍后面板接口类型和定义。



编号	功能定义
1	1 x 导轨支架，通过支架将ED-IPC3400主机安装在导轨上。
2	1 x Type-C USB接口，支持通过此接口对系统进行eMMC烧录。
3	1 x Micro SD卡槽，支持安装Micro SD卡，用于启动系统。 注：仅当使用CM5Lite时，安装的Micro SD卡才有效。

1.3.3 侧面板

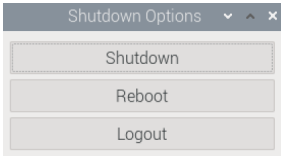
介绍侧面板接口类型和定义。



编号	功能定义
1	1 x Wi-Fi/BT天线接口（选配），SMA接口，连接Wi-Fi/BT天线。
2	1 x ON/OFF按键，隐藏式按键，按下按键打开Desktop系统的关机选项菜单。
3	1 x 接地点，连接外部电源的大地。

1.4 按键

ED-IPC3400设备包含1个ON/OFF按键，该按键为隐藏式按键，在外壳上的丝印为“ON/OFF”。在Desktop系统中短按按键，打开关机选项菜单，可选择关机、重启或注销。



提示

- 在Desktop系统中，快速连续按2次ON/OFF按键可直接关机。
- 连续按2次ON/OFF按键使设备关机后，再短按1次ON/OFF按键可使设备开机。

1.5 指示灯

介绍ED-IPC3400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

指示灯	状态	描述
以太网口黄色指示灯	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
以太网口绿色指示灯	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

1.6 接口

介绍产品中各接口的定义和功能。

1.6.1 Micro SD卡槽

ED-IPC3400设备包含1个Micro SD卡槽，接口丝印为“”，支持安装Micro SD卡，用于启动系统。

提示

仅当使用CM5Lite时，安装的Micro SD卡才有效。

1.6.2 电源接口

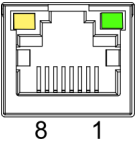
ED-IPC3400设备包含1路电源输入，使用DC Jack（5.5x2.5）连接器，接口丝印为“DC IN”，支持9V~28V的电源输入。

提示

推荐使用12V 4A的电源适配器。

1.6.3 2500M以太网接口

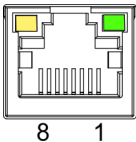
ED-IPC3400设备包含1路自适应10/100/1000M/2500M以太网接口，接口丝印为“ 2500M”。使用RJ45端子，接入以太网时建议采用Cat6及以上规格的网线配合使用。端子对应的引脚定义如下：

	Pin ID	Pin Name
	1	TX1+
	2	TX1-
	3	TX2+

	4	TX2-
	5	TX3+
	6	TX3-
	7	TX4+
	8	TX4-

1.6.4 1000M以太网接口

ED-IPC3400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-

1.6.5 HDMI接口

ED-IPC3400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。

1.6.6 USB 2.0接口

ED-IPC3400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。

1.6.7 USB 3.0接口

ED-IPC3400设备包含1路USB 3.0接口，接口丝印为“”，标准的Type-A接口。支持连接标准的USB 3.0外设，最大支持5Gbps的传输速率。

1.6.8 Type-C USB接口

ED-IPC3400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。

1.6.9 天线接口（选配）

ED-IPC3400设备包含1路SMA天线接口，接口丝印为“WiFi/BT”，连接Wi-Fi/BT天线。

1.6.10 内部接口

介绍ED-IPC3400设备内部预留的接口，需要打开设备外壳后才能获取，可根据实际需要进行扩展。

1.6.10.1 MIPI接口

ED-IPC3400设备内部包含2路MIPI接口，22-Pin FPC连接器，分别为DPHY0 (MIPI0)和DPHY1 (MIPI1)，均支持扩展连接Raspberry Pi Camera和Raspberry Pi Touch Display 2。

1.6.10.2 40-PIN GPIO接口

ED-IPC3400设备内部包含1个40-PIN GPIO连接器，引脚定义如下：

引脚	定义	引脚	定义
1	3.3V	2	5V
3	GPIO2（默认上拉至3.3V，配置为I2C功能，定义为SDA）	4	5V
5	GPIO3（默认上拉至3.3V，配置为I2C功能，定义为SCL）	6	GND
7	GPIO4	8	GPIO14
9	GND	10	GPIO15
11	GPIO17	12	GPIO18
13	GPIO27	14	GND
15	GPIO22	16	GPIO23
17	3.3V	18	GPIO24
19	GPIO10	20	GND
21	GPIO9	22	GPIO25
23	GPIO11	24	GPIO8
25	GND	26	GPIO7

引脚	定义	引脚	定义
27	ID_SD (已使用)	28	ID_SC (已使用)
29	GPIO5	30	GND
31	GPIO6	32	GPIO12
33	GPIO13	34	GND
35	GPIO19	36	GPIO16 (已使用)
37	GPIO26	38	GPIO20
39	GND	40	GPIO21

1.7 RTC

ED-IPC3400设备内部集成RTC功能，默认提供RTC电池底座，支持安装CR1220电池作为RTC备份电源。

1.8 加密芯片

ED-IPC3400设备内部集成加密芯片（Microchip ATSHA204A），可搭配用户的上层应用，安全存储设备的私钥和数字证书等关键信息，确保关键信息无法被软件读取或导出，从而有效防止设备被复制或者伪造。

1.9 EEPROM

ED-IPC3400设备内部集成EEPROM，具备4K Byte容量的存储，支持用户对容量小的关键数据进行写入和保存。

1.10 Hailo-8（13 TOPS）AI 加速模块（选配）

ED-IPC3400设备支持选配Hailo-8（13 TOPS）AI模块，搭载Raspberry Pi CM5可实现AI边缘运算，具备高算力，适用于机器视觉等工业场景。

2 安装部件（可选）

本章介绍安装天线和Micro SD卡等部件的具体操作。

2.1 安装天线

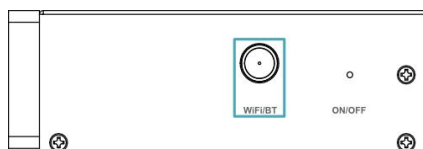
若选购的ED-IPC3400设备包含Wi-Fi功能，在使用设备之前需要先安装天线。

前提条件：

已从包装盒中获取对应的天线。

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，确保不会脱落即可。

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。

注意

Micro SD卡不支持带电插拔。

提示

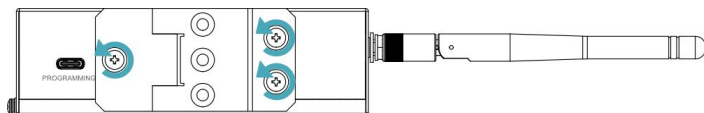
- 仅当ED-IPC3400使用CM5Lite时，支持从Micro SD卡启动系统。
- 当ED-IPC3400使用带eMMC的CM5时，Micro SD卡槽为无效接口。

前提条件：

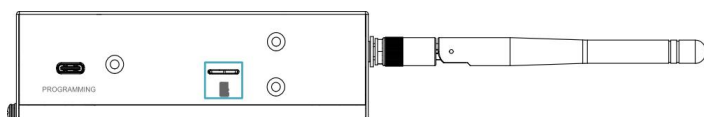
- 已获取待使用的Micro SD卡。
- 已断开设备电源。

操作步骤：

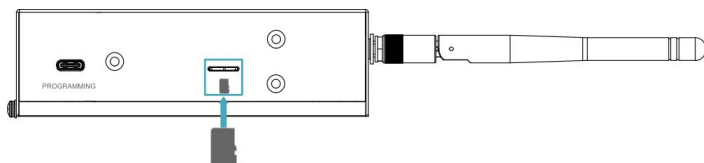
1. 使用十字螺丝刀逆时针拧下DIN导轨支架上的3颗螺钉，将默认DIN导轨支架拆除。



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。

3 安装设备

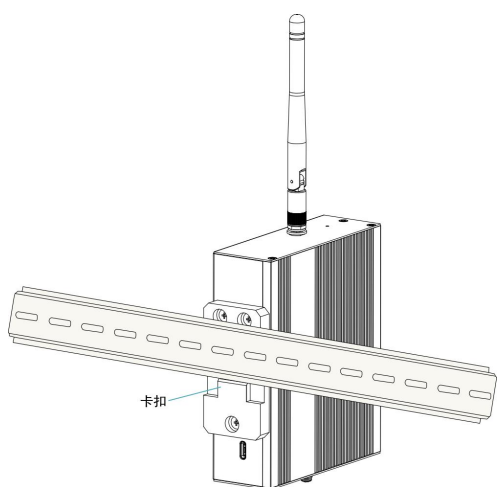
本章介绍安装设备的具体操作。

3.1 导轨安装

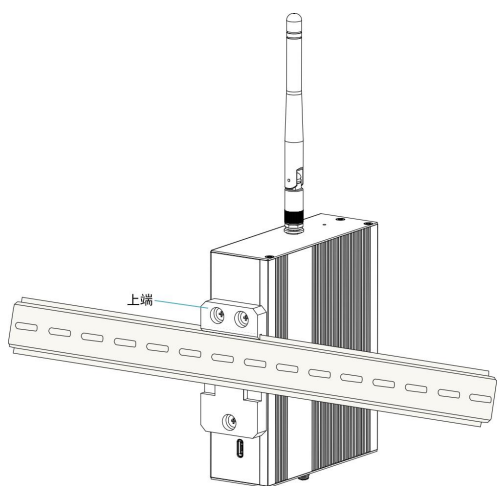
ED-IPC3400设备支持DIN导轨安装，出厂默认标配并已安装DIN导轨支架，下文介绍导轨安装的具体步骤。

操作步骤：

1. 将设备的带导轨支架侧对着待安装的轨道，将支架的可伸缩卡扣从下往上套在DIN导轨的下沿。



2. 用手握住设备由下往上用力按压卡扣，使支架上端完全卡入DIN导轨上沿。



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

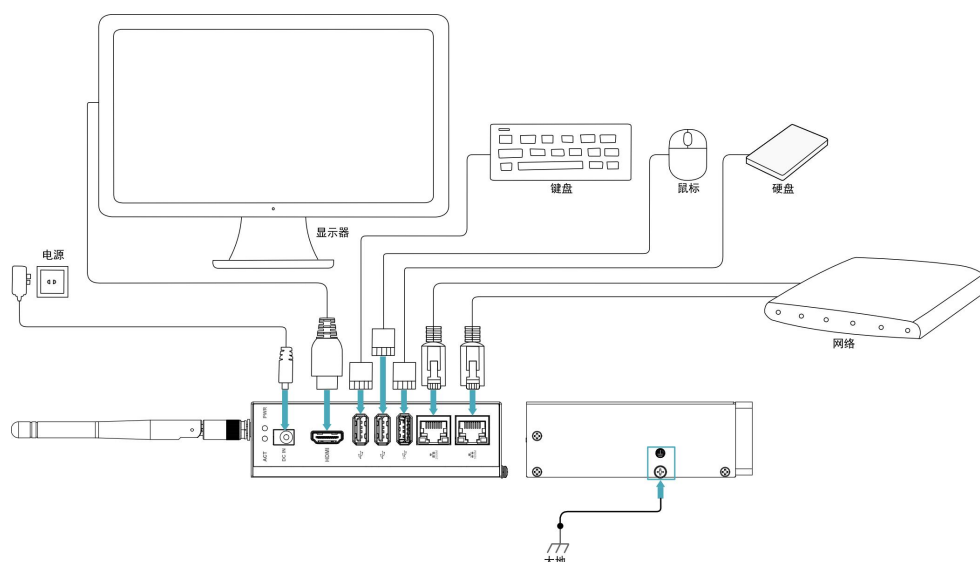
介绍线缆的连接方法。

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-IPC3400设备无电源开关，接入电源后，系统将会开始启动。

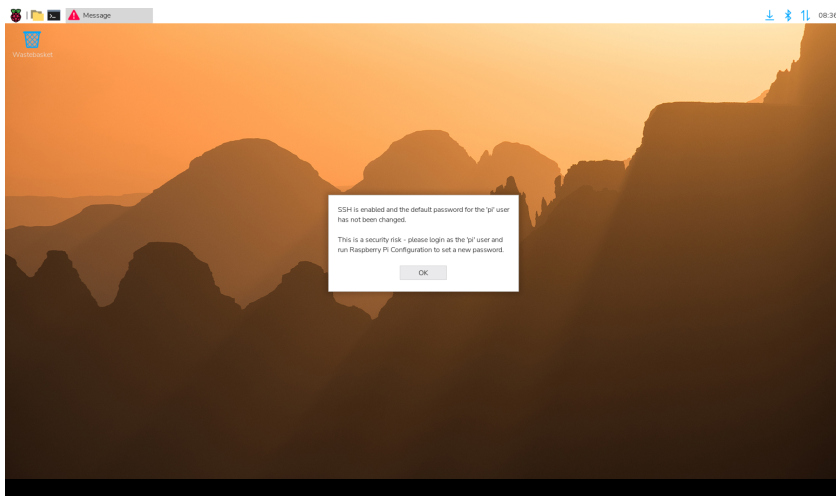
- 红色PWR灯点亮，表示设备已正常供电。
- 绿灯ACT灯闪烁，表示系统正常启动，然后屏幕的左上角会出现Raspberry Pi 的logo。

提示

默认用户名：`pi`；默认密码：`raspberry`。

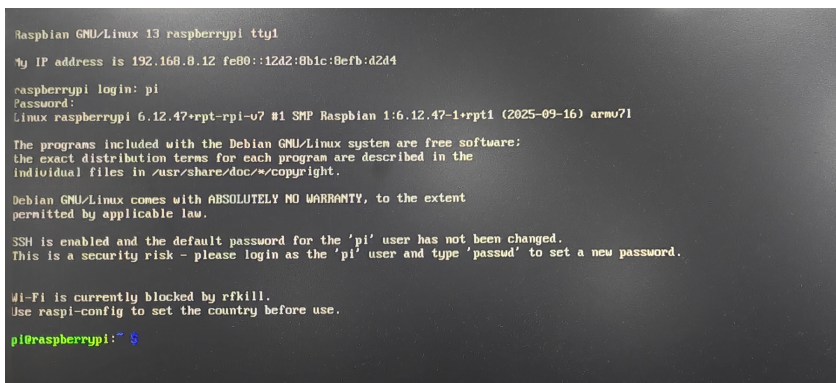
4.2.1 Raspberry Pi OS (Desktop)

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。



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 RTC

配置RTC

5.8 配置MIPI接口（可选）

ED-IPC3400设备包含2路MIPI接口（位于设备内部），22-Pin FPC连接器，分别为DPHY0 (MIPI0) 和DPHY1 (MIPI1)，均支持扩展连接Raspberry Pi Camera和Raspberry Pi Touch Display。

提示

此接口位于设备内部，需要打开设备外壳才能使用，仅作为预留接口。

2路MIPI接口支持扩展连接的Raspberry Pi Camera/Touch Display的型号如下：

- Camera :
 - OmniVision OV5647
 - Sony IMX219
 - Sony IMX296
 - Sony IMX477
 - Sony IMX708
- Raspberry Pi Touch Display 2 (5-inch和7-inch)

在ED-IPC3400上使用Raspberry Pi Camera/Touch Display之前，需要先进行相关的配置，添加Camera和Touch Display的参数，再连接Raspberry Pi Camera/Touch Display，下文介绍具体的操作。

前提条件：

ED-IPC3400设备已正常启动。

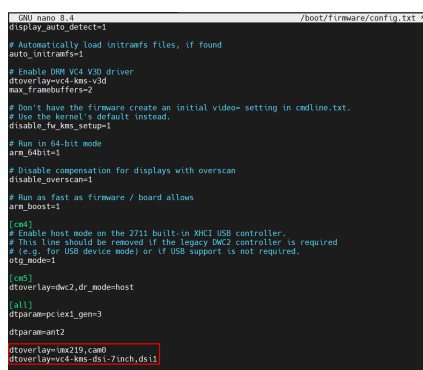
操作步骤：

1. 设备正常启动后，打开命令窗格，执行如下命令，打开 `config.txt` 配置文件。

```
sudo nano /boot/firmware/config.txt
```

sh

2. 在配置文件的末尾添加待使用的Camera和Touch Display的信息。（如下图红框所示的内容，将MIPI0接口配置为Sony IMX219 Camera，将MIPI1配置为7-inch Raspberry Pi Touch Display 2）。



```
GNU nano 2.9.4 /boot/firmware/config.txt
display_auto_detect=1
# Automatically load untrams files, if found
auto_untrams=1
# Enable BCM V4 V3D driver
dtoverlay=vc4-kms-v3d
max_framebuffers=2
# Don't have the firmware create an initial video= setting in cmdline.txt.
# Use the kernel's default instead.
disable_fw_kms_setup=1
# Run in 64-bit mode
arm_64bit=1
# Disable compensation for displays with overscan
disable_overscan=1
# Run as fast as firmware / board allows
arm_boost=1
[cam]
# Enable host mode on the 2711 built-in XHCI USB controller.
# This line should be removed if the legacy DWC2 controller is required
# (e.g. for USB device mode) or if USB support is not required.
usb_mode=1
[cam]
dtoverlay=dwc2,dr_mode=host
[all]
dtparam=pcie1_gen3
dtparam=ant2
dtoverlay=imx219,cam0
dtoverlay=vc4-kms-dsi-7inch-ds1
```

其他配置参数示例：

- 将MIPI0和MIPI1均配置为Sony IMX219 Camera。

```
dtoverlay=imx219,cam0
dtoverlay=imx219,cam1
```

sh

- 将MIPI0和MIPI1均配置为7-inch Raspberry Pi Touch Display 2。

```
dtoverlay=vc4-kms-dsi-ili9881-7inch,dsi0
dtoverlay=vc4-kms-dsi-ili9881-7inch,dsi1
```

sh

- 将MIPI0接口配置为7-inch Raspberry Pi Touch Display 2，将MIPI1配置为Sony IMX219 Camera。

```
dtoverlay=vc4-kms-dsi-ili9881-7inch,dsi0
dtoverlay=imx219,cam1
```

sh

提示

- 若使用的是其他型号的Camera，在配置参数中修改Camera的型号即可，例如：如果使用的是Sony IMX296，则可以将 `imx219` 更改为 `imx296`。
- 若使用的是5-inch的Touch Display 2，在配置参数中修改屏幕尺寸即可，例如：如果使用的是5-inch Raspberry Pi Touch Display 2，则可以将 `vc4-kms-dsi-ili9881-7inch` 更改为 `vc4-kms-dsi-ili9881-5inch`。
- 更多关于Raspberry Pi Camera的配置信息，请参考[Raspberry Pi Camera \(https://www.raspberrypi.com/documentation/accessories/camera.html#about-the-camera-modules\)](https://www.raspberrypi.com/documentation/accessories/camera.html#about-the-camera-modules)。
- 更对关于Raspberry Pi Touch Display 2的配置信息，请参考[Raspberry Pi Touch Display 2 \(https://www.raspberrypi.com/documentation/accessories/touch-display-2.html#raspberry-pi-touch-display-2\)](https://www.raspberrypi.com/documentation/accessories/touch-display-2.html#raspberry-pi-touch-display-2)。

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. 执行如下命令重启设备，使配置生效。

```
sudo reboot
```

sh

5. 断开ED-IPC3400设备的电源。
6. 将Raspberry Pi Camera/Touch Display通过FPC接口安装至ED-IPC3400主板上。

5.9 Hailo-8模块（可选）

ED-IPC3400设备支持选配Hailo-8（13 TOPS）AI模块，我司预安装的操作系统默认已完成Hailo-8（一个固定版本）对应的配置和软件的安装。若用户自行安装标准的Raspberry Pi OS，则可参照下文来进行配置和软件的安装。

前提条件：

ED-IPC3400设备已正常启动。

操作步骤：

1. 修改 `config.txt` 配置文件，使能PCIe Gen3.0接口。

a. 设备正常启动后，打开命令窗格，执行如下命令，打开 `config.txt` 配置文件。

```
sudo nano /boot/firmware/config.txt
```

sh

b. 在配置文件中添加如下内容。

```
dtparam=pciex1_gen=3
```

sh

```
# Disable compensation for displays with overscan
disable_overscan=1

# Run as fast as firmware / board allows
arm_boost=1

[cm4]
# Enable host mode on the 2711 built-in XHCI USB controller.
# This line should be removed if the legacy DWC2 controller is required
# (e.g. for USB device mode) or if USB support is not required.
otg_mode=1

[cm5]
dtoverlay=dwc2,dr_mode=host

[all]
dtparam=pciex1_gen=3
```

c. 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。

d. 执行如下命令重启设备，使配置生效。

```
sudo reboot
```

sh

2. 依次执行如下命令，安装Hailo软件。

```
sudo apt install dkms
sudo apt install hailo-all
```

sh

3. 执行如下命令，查看已安装的Hailo软件的详细信息。

```
dpkg -l | grep hailo
```

sh

4. 更多关于Hailo模块的具体配置，请查看Raspberry Pi AI Software文档 (<https://www.raspberrypi.com/documentation/computers/ai.html>)。

提示

我司预安装的操作系统中内置的Hailo软件的版本信息如下：

```
pi@ipc3400:~$ dpkg -l | grep hailo
ii hailo-all          5.1.1      all      Hailo-8 support (metapackage)
ii hailo-models       1.0.0-2   all      AI models for Hailo modules
ii hailo-tappas-core  5.1.0     arm64    Core components of the Hailo TAPPAS platform
ii hailort            4.23.0    arm64    AI acceleration runtime library for Hailo hardware
ii hailort-pcie-driver 4.23.0    all      Hailo PCIe driver and firmware for AI hardware accelerators
ii python3-hailo-tappas 5.1.0     arm64    Python binding for tappas
ii python3-hailort    4.23.0-1  arm64    HailoRT Python API, which wraps the runtime library
ii rpicam-apps-hailo-postprocess 1.12.0-1 arm64    Raspberry Pi AI HAT+ post-processing plugin for rpicam-apps
```

5.10 配置加密芯片

ED-IPC3400设备内部集成加密芯片（Microchip ATSHA204A），加密芯片的具体使用和用户实际的上层应用有关；因为每个客户的需求不同，故下文仅介绍加密芯片对应的加密库源码下载/编译和加密芯片的测试示例。

前提条件：

ED-IPC3400设备已正常启动。

操作步骤：

1. 修改 `config.txt` 配置文件，使能I2C和加密芯片。

a. 设备正常启动后，打开命令窗格，执行如下命令，打开 `config.txt` 配置文件。

```
sudo nano /boot/firmware/config.txt
```

sh

b. 在配置文件中添加如下内容。

```
dtparam=i2c_arm=on
dtparam=ant2
```

sh

```
GNU nano 8.4 /boot/firmware/config.txt *
# For more options and information see
# http://rptl.io/configtxt
# Some settings may impact device functionality. See link above for details
# Uncomment some or all of these to enable the optional hardware interfaces
dtparam=i2c_arm=on
dtparam=ant2
#dtparam=vs=on
#dtparam=spi=on
#
# Enable audio (loads snd_bcm2835)
dtparam=audio=on
```

c. 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。

d. 执行如下命令重启设备，使配置生效。

```
sudo reboot
```

sh

2. （可选）在GitHub上获取CryptoAuthLib v3.7.0版本的加密库源码，路径为：[CryptoAuthLib v3.7.0 \(https://github.com/MicrochipTech/cryptoauthlib/tree/v3.7.0\)](https://github.com/MicrochipTech/cryptoauthlib/tree/v3.7.0)。

3. （可选）在ED-IPC3400设备上对下载的加密库源码进行编译。

```
mkdir build && cd build
cmake .. -DATCA_HAL_I2C=ON -DATCA_BUILD_SHARED_LIBS=on -DATCA_USE_ATCAB_FUNCTIONS=ON
make -j4
sudo make install
```

sh

4. 安装Python，使Python能导入cryptoauthlib。

```
cd ~/cryptoauthlib-3.7.0/python
sudo cp -r cryptoauthlib/ $(python3 -c "import site; print(site.getsitepackages()[0])")
```

sh

5. 通过示例代码来测试加密芯片是否可以正常使用。

```
#!/usr/bin/env python3
"""
CryptoAuthLib 3.7.0 + ATSHA204A @ i2c-1 0x64
"""
import ctypes

from cryptoauthlib.library import load_cryptoauthlib, AtcaReference
load_cryptoauthlib()

from cryptoauthlib.atcab import (
    atcab_init, atcab_release,
    atcab_info, atcab_read_serial_number,
    atcab_random, atcab_is_locked
)
# from cryptoauthlib.iface import cfg_ateccx08a_i2c_default
from cryptoauthlib.iface import cfg_atsha20xa_i2c_default

cfg = cfg_atsha20xa_i2c_default()

cfg.cfg.atcai2c.u.slave_address = 0x64 << 1
cfg.cfg.atcai2c.bus = 1
print(f"slave_address = 0x{cfg.cfg.atcai2c.u.slave_address:02X}")

st = atcab_init(cfg)
if st != 0:
    print(f" atcab_init failed: {st}")
    exit(1)

print("ATSHA204A@0x64 initialized ok.")

rev = bytearray(4)
atcab_info(rev)
print(f"Rev : {rev.hex()}")

sn = bytearray(9)
```

sh

```

atcab_read_serial_number(sn)
print(f"SN : {sn.hex()}")

rnd = bytearray(32)
atcab_random(rnd)
print(f"RND : {rnd[:8].hex()}...")

is_locked_cfg = AtcaReference(ctypes.c_bool(False))

status1 = atcab_is_locked(0x00, is_locked_cfg)
if status1 == 0:
    cfg_lock = is_locked_cfg.value
    print(f"Config loaded: {cfg_lock}")
else:
    print(f"Config lock status read failed: {status1}")

atcab_release()
print("Test completed.")

```

提示

- CryptoAuthLib v3.7.0对ATSHA204A的支持较稳定，且功能完整，故建议使用CryptoAuthLib v3.7.0版本的加密库源码。
- 若使用的是我司预安装的操作系统，则默认已安装CryptoAuthLib v3.7.0版本的加密库源码并完成源码编译，故可跳过步骤1和步骤2。
- 若用户自行安装标准的Raspberry Pi OS，则需要执行步骤1~步骤4。

5.11 配置EEPROM

ED-IPC3400设备内部集成EEPROM，具备4K Byte容量的存储，支持用户对容量小的关键数据进行写入和保存。下文介绍具体的写入和读取数据的操作。

前提条件：

ED-IPC3400设备已正常启动。

操作步骤：

1. 打开命令终端，执行如下命令，创建一个自定义名称的 `.txt` 文件。

```
nano edatec_hat.txt
```

sh

- `edatec_hat` 表示自定义的文件名。
2. 在 `edatec_hat.txt` 文件中添加如下需要写入EEPROM的内容（此处仅为示例，用户按需添加内容即可）。

```
cat > myhat.txt << 'EOF'
product_uuid 00000000-0000-0000-0000-000000000000
product_id 0x0001
product_ver 0x0001
vendor "My Company"
product "My HAT"
current_supply 0
dt_blob "my-overlay"
EOF
```

sh

提示

写入EEPROM的内容有格式限制，详细的规格请参考[EEPROM配置文件说明 \(https://github.com/raspberrypi/Utils/blob/master/eeptools/EEPROM_settings.txt\)](https://github.com/raspberrypi/Utils/blob/master/eeptools/EEPROM_settings.txt)。

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，将创建的 `.txt` 文件转换为EEPROM可以识别的 `.eep` 文件。

```
eeptool edatec_hat.txt edatec_hat.eep
```

sh

5. 执行如下命令，将 `edatec_hat.eep` 写入EEPROM。

```
sudo eepflash.sh -w -f=edatec_hat.eep -t=24c32 -d=1
```

sh

- `-w`：表示为写入模式；
- `-f=edatec_hat.eep`：表示待写入的文件；
- `-t=24c32`：表示EEPROM芯片的型号为AT24C32；
- `-d=1`：表示I2C总线编号

6. 执行如下命令，读取EEPROM中的数据，并保存为名为 `readback.bin` 的二进制文件。

```
sudo eepflash.sh -r -f readback.bin -t=24c32 -d=1
```

sh

- `-r`：表示为读取模式；
- `-t=24c32`：表示EEPROM芯片的型号为AT24C32；
- `-d=1`：表示I2C总线编号

7. 执行如下命令，解析读取的 `readback.bin` 的二进制文件。

```
hexdump -C readback.bin | head -20
```

sh

- 20 : 表示显示20行的信息。

提示

EEPROM的读写次数存在寿命限制，请勿频繁操作，以免影响其使用寿命。

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，则需要重新下载系统镜像并进行烧录。下文介绍下载镜像和烧录eMMC/SSD/SD卡的具体操作。

6.1 下载镜像

我司提供2种镜像供用户下载，下载路径如下表：

OS	下载路径
Raspberry Pi OS(Desktop) 64-bit-trixie (Debian 13)，已安装Hailo软件和加密芯片库	2026-06-18_IPC3400_hailo_raspbios_trixie_arm64 (https://1826505135.cdn.123clouddisk.com/1826505135/37871790)
Raspberry Pi OS(Desktop) 64-bit-trixie (Debian 13)，已安装加密芯片库	2026-06-18_IPC3400_raspbios_trixie_arm64 (https://1826505135.cdn.123clouddisk.com/1826505135/38096526)

6.2 烧录eMMC/SSD

提示

烧录SSD的操作步骤与烧录eMMC的操作步骤相同。

若选购的ED-IPC3400设备包含eMMC或者SSD，则支持从eMMC或者SSD启动系统，可参考下文进行烧录。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)

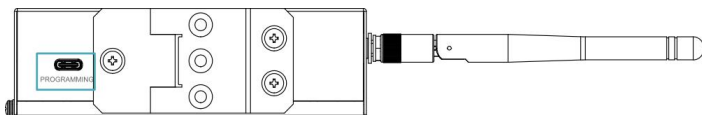
前提条件：

- 已完成烧录工具的下载，并安装至Windows电脑。
- 已准备一根Type-C USB转USB Type-A的连接线（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. 确定设备侧Type-C USB接口的位置，如下图标注位置所示。



2. 连接好电源线和USB烧录线（Type-C USB转USB Type-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。

3. 断开ED-IPC3400的电源，再重新上电。

4. 打开已安装的 `rpi-mass-storage-gadget64.bat` 工具(如下图红框所示)，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

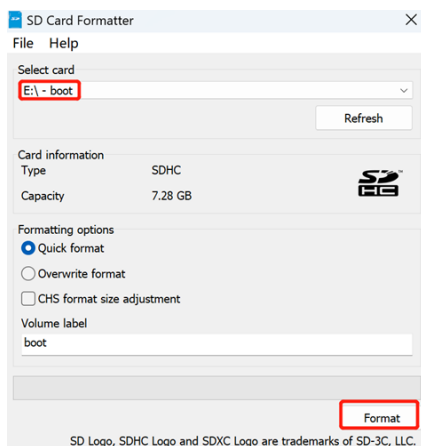
```

C:\WINDOWS\system32\cmd
USB mass storage gadget for Raspberry Pi 5
RPIBOOT: build-date Nov 13 2024 version 20240422-085300 e3e0fa29
Loading: mass-storage-gadget64/bootfiles.bin
Using mass-storage-gadget64/bootfiles.bin
Waiting for BCM2835/6/7/2711/2712...
Loading bootcode.bin
Successful read 4 bytes
Waiting for BCM2835/6/7/2711/2712...
Second stage boot server
File read: mcb.bin
File read: memsys00.bin
File read: memsys01.bin
File read: memsys02.bin
File read: memsys03.bin
File read: bootmain
Loading: mass-storage-gadget64/config.txt
File read: config.txt
Loading: mass-storage-gadget64/boot.img
File read: boot.img
Second stage boot server done

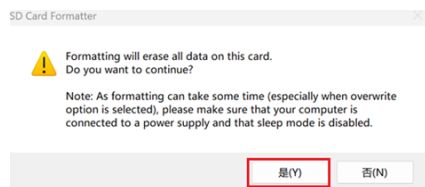
Raspberry Pi Mass Storage Gadget started
EMMC/MMIO devices should be visible in the Raspberry Pi Imager in a few seconds.
For debug, you can login to the device using the USB serial gadget - see COM ports in Device Manager.
Press a key to close this window.
  
```

5. 待盘符化完成后，电脑右下角会弹出盘符。

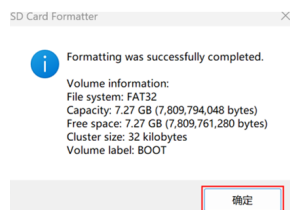
6. 打开 `SD Card Formatter`，选择被格式化的盘符，单击右下方“Format”进行格式化。



7. 在弹出的提示框中，单击“是”。

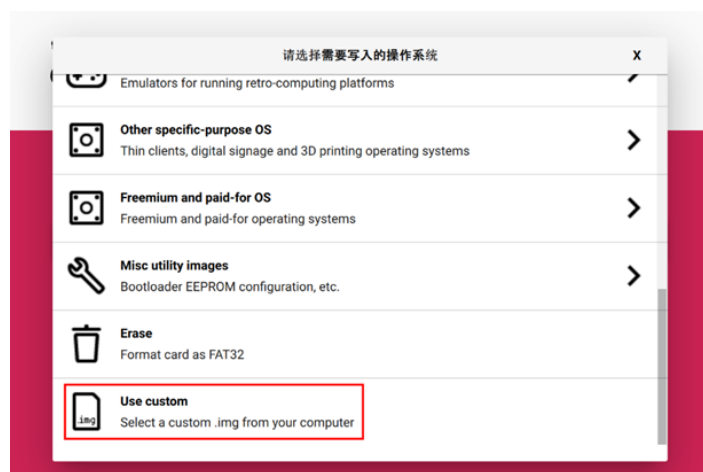


8. 格式化完成后，在提示框中单击“确定”。



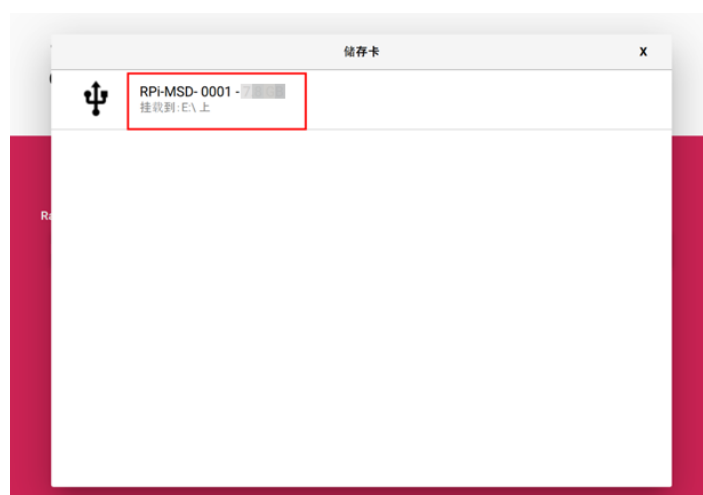
9. 关闭 SD Card Formatter。

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

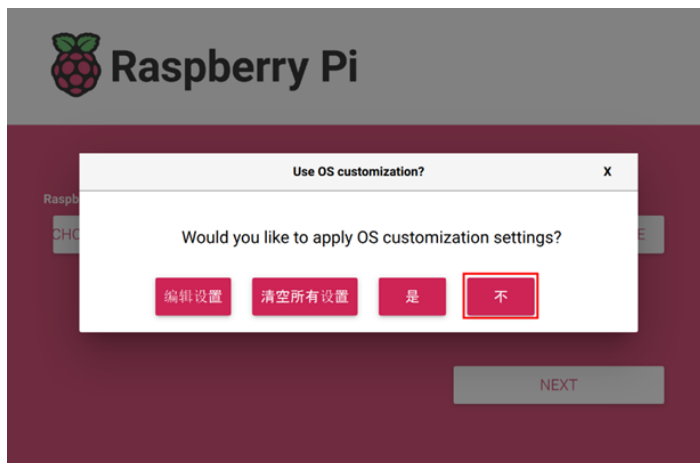


11. 根据提示，在自定义路径下选择已获取的镜像文件，并返回至烧录主界面。

12. 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。



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



14. 在弹出的“警告”提示框中选择“是”，开始写入镜像。



15. 待镜像写入完成后，会进行文件的验证。



16. 验证完成后，弹出“烧录成功”提示框，单击“继续”完成烧录。

17. 关闭 `Raspberry Pi Imager`，取下USB烧录线，重新给设备上电。

6.3 烧录SD卡

若选购的ED-IPC3400设备不包含eMMC仅包含SD卡，则支持从SD卡启动系统，可参考下文进行烧录。

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/>)

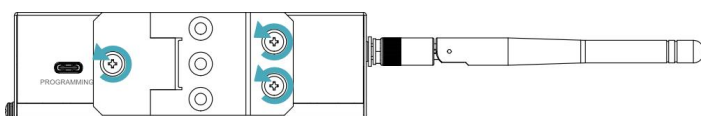
前提条件：

- 已完成烧录工具的下载，并安装至Windows电脑。
- 已获取待烧录的镜像文件。
- 已准备一个SD卡读卡器。
- 已断开设备电源。

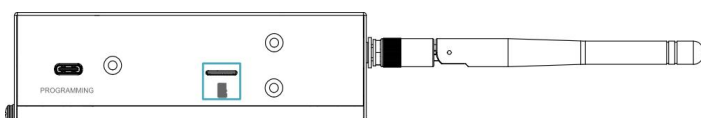
操作步骤：

操作步骤以Windows系统为例进行说明。

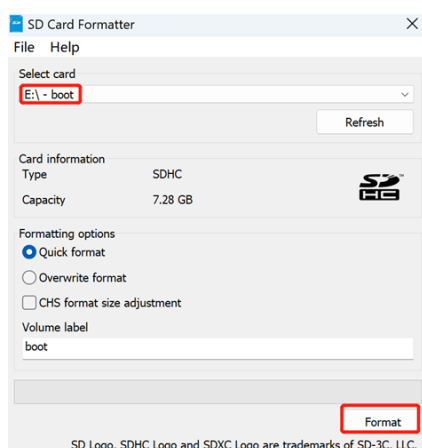
1. 使用十字螺丝刀逆时针拧下导轨支架上的3颗螺钉（下图箭头位置），将默认的导轨支架拆除。



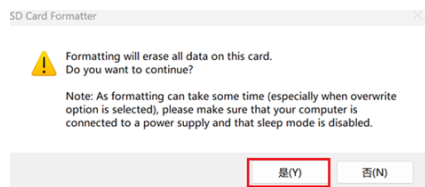
2. 确定Micro SD卡的位置，如下图标注位置所示。



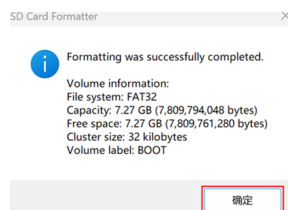
3. 用手将Micro SD卡向卡槽里按下使其弹出，再拔出Micro SD卡。。
4. 打开 **SD Card Formatter**，选择被格式化的盘符，单击右下方“Format”进行格式化。



5. 在弹出的提示框中，单击“是”。

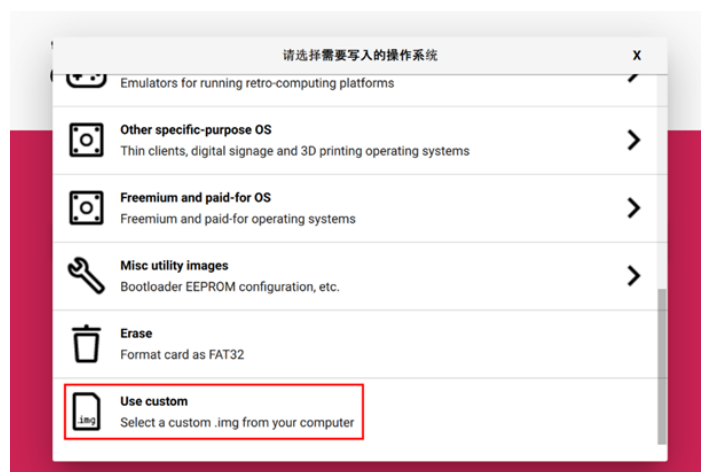


6. 格式化完成后，在提示框中单击“确定”。



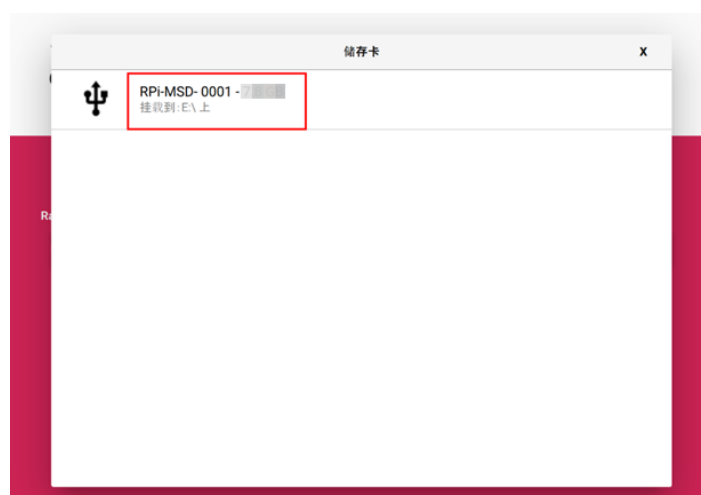
7. 关闭 SD Card Formatter。

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

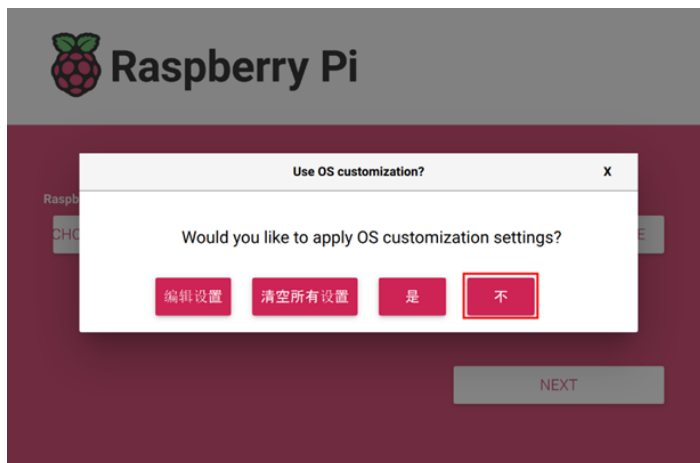


9. 根据提示，在自定义路径下选择已获取的镜像文件，并返回至烧录主界面。

10. 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。



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



12. 在弹出的“警告”提示框中选择“是”，开始写入镜像。



13. 待镜像写入完成后，会进行文件的验证。



14. 验证完成后，弹出“烧录成功”提示框，单击“继续”完成烧录。

15. 关闭 **Raspberry Pi Imager**，取下读卡器和Micro SD卡。

16. 将Micro SD卡重新插入设备中，再给设备上电。