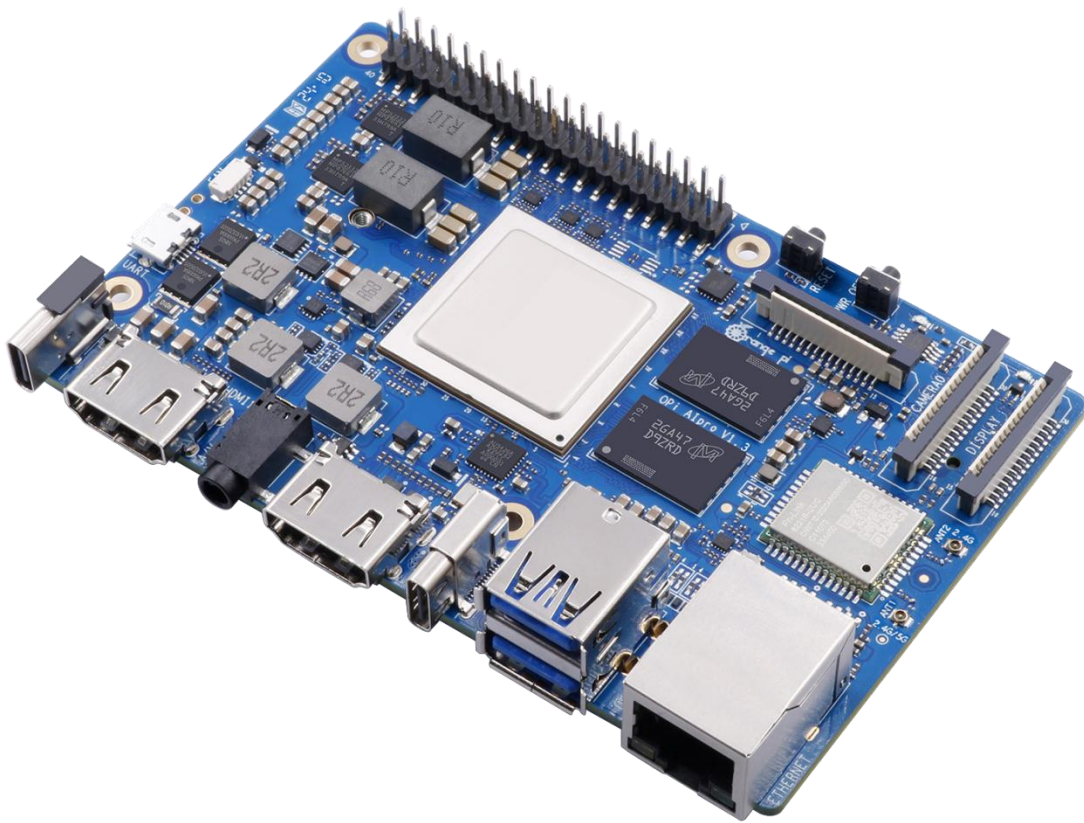


# Orange Pi AI Pro

## 环境搭建指南



# 目录

|                                                |    |
|------------------------------------------------|----|
| 1. 引言 .....                                    | 1  |
| 1.1. Orange Pi AI Pro 产品简介 .....               | 1  |
| 1.2. 开发板的接口详情图 .....                           | 1  |
| 1.3. CANN 简介 .....                             | 2  |
| 1.4. MindSpore 简介 .....                        | 4  |
| 2. 开发板使用介绍 .....                               | 6  |
| 2.1. 准备需要的配件 .....                             | 6  |
| 2.2. 下载开发板系统镜像 .....                           | 10 |
| 2.2.1. 百度网盘下载 .....                            | 10 |
| 2.3. 控制启动设备的两个拨码开关的使用说明 .....                  | 11 |
| 2.4. WIFI 蓝牙天线使用注意事项 .....                     | 11 |
| 2.5. 镜像烧录 .....                                | 13 |
| 2.5.1. 烧写 Linux 镜像到 TF 卡中的方法 .....             | 13 |
| 2.5.2. 烧写 Linux 镜像到 eMMC 中的方法 .....            | 20 |
| 2.5.3. 烧写 Linux 镜像到 NVMe SSD 中的方法 .....        | 25 |
| 2.5.4. 烧写 Linux 镜像到 SATA SSD 中的方法 .....        | 31 |
| 2.6. 板载 LED 灯说明 .....                          | 37 |
| 2.7. 启动开发板的步骤 .....                            | 38 |
| 2.8. 调试串口的使用方法 .....                           | 39 |
| 2.8.1. 通过 Micro USB 接口来使用调试串口的连接说明 .....       | 39 |
| 2.8.2. 通过 40 pin 接口中的 uart0 来使用调试串口的连接说明 ..... | 40 |
| 2.8.3. Ubuntu 平台调试串口的使用方法 .....                | 41 |
| 2.8.4. Windows 平台调试串口的使用方法 .....               | 44 |
| 2.9. Linux 系统功能适配情况 .....                      | 47 |
| 2.10. HDMI 接口的使用说明 .....                       | 48 |
| 2.10.1. HDMI 显示 Linux 桌面的说明 .....              | 48 |
| 2.10.2. 使用 HDMI 接口显示图片和播放音频的方法 .....           | 48 |



|                                                     |    |
|-----------------------------------------------------|----|
| 2. 11. Linux 系统默认登录账号和密码 .....                      | 50 |
| 2. 12. 登录 Linux 系统桌面的方法 .....                       | 50 |
| 2. 13. USB 接口测试 .....                               | 52 |
| 2. 13. 1. Type-C USB 接口使用注意事项 .....                 | 52 |
| 2. 13. 2. 连接 USB 鼠标或键盘测试 .....                      | 52 |
| 2. 13. 3. USB 摄像头测试 .....                           | 53 |
| 2. 13. 4. USB 音频测试 .....                            | 54 |
| 2. 14. 网络连接测试 .....                                 | 56 |
| 2. 14. 1. 以太网口测试 .....                              | 56 |
| 2. 14. 2. WIFI 连接测试 .....                           | 57 |
| 2. 14. 3. 设置静态 IP 地址的方法 .....                       | 63 |
| 2. 15. SSH 远程登录开发板 .....                            | 70 |
| 2. 15. 1. Ubuntu 下 SSH 远程登录开发板 .....                | 71 |
| 2. 15. 2. Windows 下 SSH 远程登录开发板 .....               | 71 |
| 2. 16. 上传文件到开发板 Linux 系统中的方法 .....                  | 73 |
| 2. 16. 1. 在 Ubuntu PC 中上传文件到开发板 Linux 系统中的方法 .....  | 73 |
| 2. 16. 2. 在 Windows PC 中上传文件到开发板 Linux 系统中的方法 ..... | 77 |
| 2. 17. MIPI LCD 屏幕的测试说明 .....                       | 81 |
| 2. 17. 1. 树莓派 5 寸屏幕的组装方法 .....                      | 81 |
| 2. 17. 2. 输出图片到 LCD 屏幕的方法 .....                     | 82 |
| 2. 18. MIPI 摄像头的测试说明 .....                          | 83 |
| 2. 19. 40 Pin 接口引脚功能说明 .....                        | 85 |
| 2. 20. wiringOP 的安装使用方法 .....                       | 86 |
| 2. 20. 1. 安装 wiringOP 的方法 .....                     | 86 |
| 2. 20. 2. 使用 wiringOP 控制 40pin GPIO 的方法 .....       | 88 |
| 2. 21. 散热风扇的使用方法 .....                              | 89 |
| 2. 22. AI CPU 和 control CPU 的设置方法 .....             | 91 |
| 2. 23. 设置 Swap 内存的方法 .....                          | 92 |
| 2. 24. 测试 MindSpore 的方法 .....                       | 93 |
| 2. 25. 使用 ascend 硬件加速的 ffmpeg .....                 | 94 |



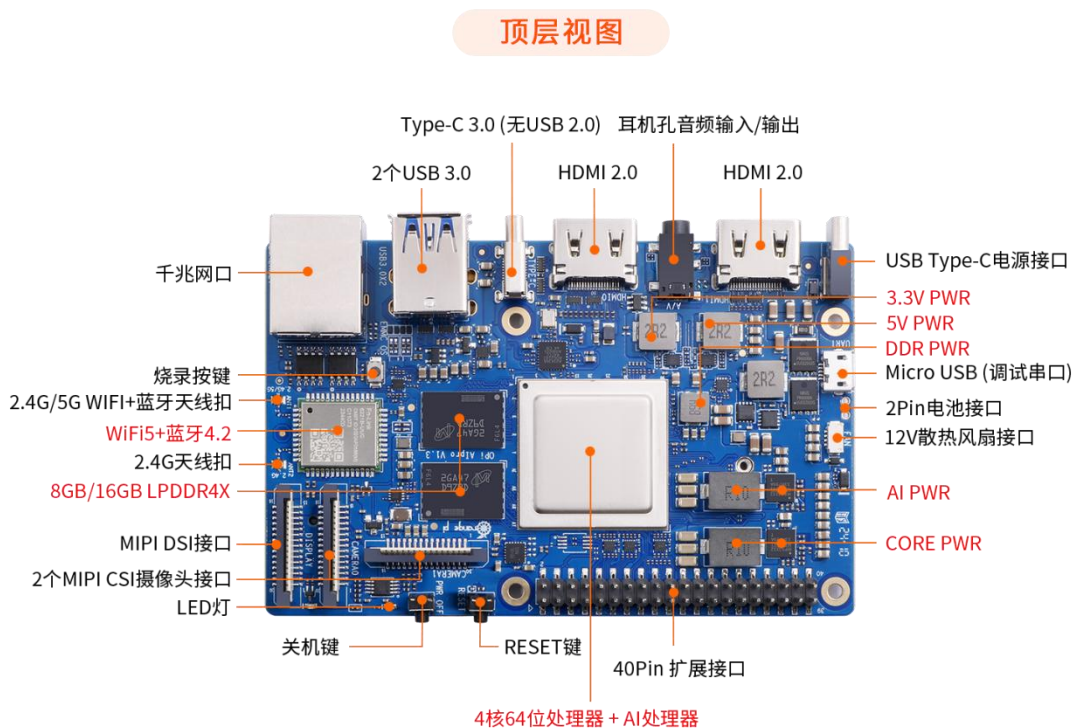
|                                       |    |
|---------------------------------------|----|
| 2. 25. 1. 使用编译好的 <b>deb</b> 软件包 ..... | 94 |
| 2. 25. 2. 从源代码构建 .....                | 95 |
| 2. 26. 关机和重启开发板的方法 .....              | 98 |

# 1. 引言

## 1.1. Orange Pi AI Pro 产品简介

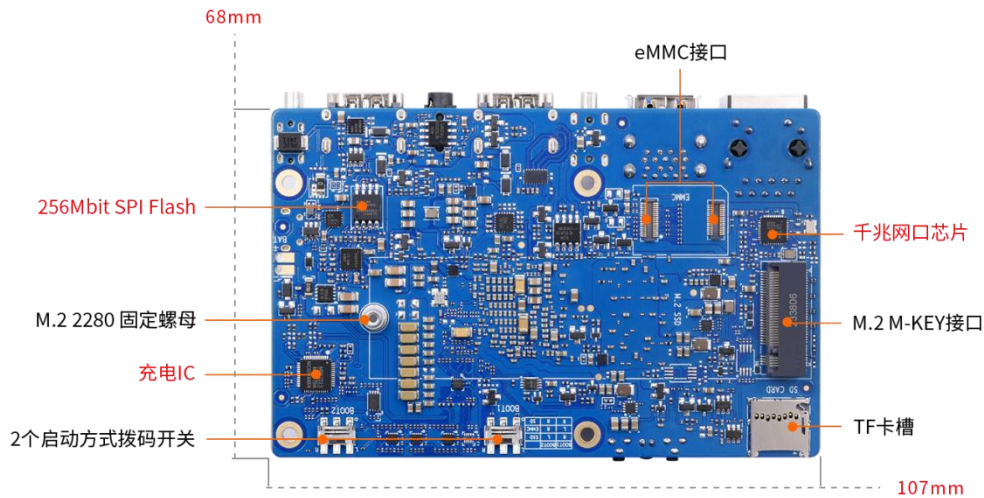
Orange Pi AI Pro 开发板是香橙派联合华为精心打造的高性能 AI 开发板，其搭载了昇腾 AI 处理器，可提供 8TOPS INT8 的计算能力，内存提供了 8GB 和 16GB 两种版本。可以实现图像、视频等多种数据分析与推理计算，可广泛用于教育、机器人、无人机等场景。

## 1.2. 开发板的接口详情图





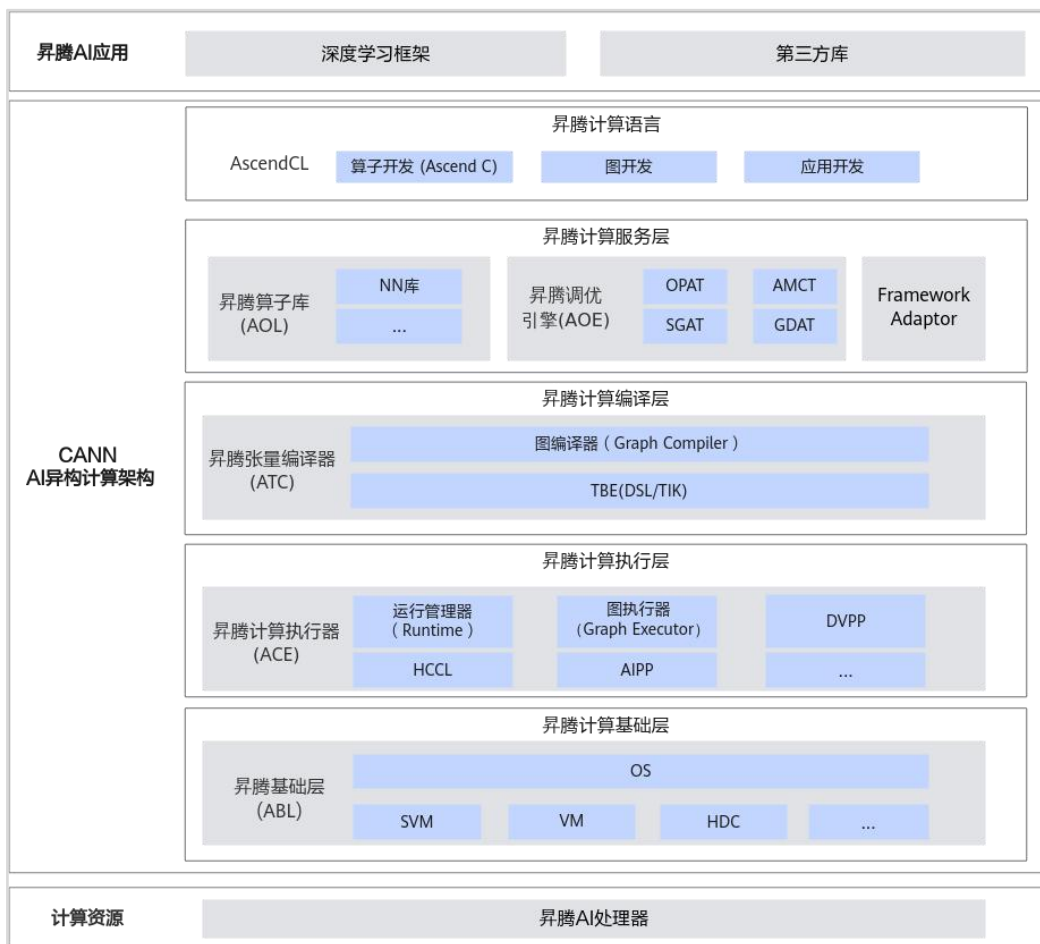
底层视图



### 1.3. CANN 简介

异构计算架构 CANN（Compute Architecture for Neural Networks）是华为针对 AI 场景推出的异构计算架构，向上支持多种 AI 框架，包括 MindSpore、PyTorch、TensorFlow 等，向下服务 AI 处理器与编程，发挥承上启下的关键作用，是提升昇腾 AI 处理器计算效率的关键平台。同时针对多样化应用场景，提供多层次编程接口，支持用户快速构建基于昇腾平台的 AI 应用和业务。

CANN 逻辑架构：



- **昇腾计算语言 (Ascend Computing Language, 简称 AscendCL) :** AscendCL 接口是昇腾计算开放编程框架，是对底层昇腾计算服务接口的封装。它提供设备 (Device) 管理、上下文 (Context) 管理、流 (Stream) 管理、内存管理、模型加载与执行、算子加载与执行、媒体数据处理、图 (Graph) 管理等 API 库，供用户开发人工智能应用。
- **昇腾计算服务层 (Ascend Computing Service Layer) :** 主要提供昇腾算子库 AOL (Ascend Operator Library)，通用神经网络 (Neural Network, NN) 库、线性代数计算库 (Basic Linear Algebra Subprograms, BLAS) 等高性能算子加速计算；昇腾调优引擎 AOE (Ascend Optimization Engine)，通过算子调优 OPAT、子图调优 SGAT、梯度调优 GDAT、模型压缩 AMCT 提升模型端到端运行速度。同时提供 AI 框架适配器 Framework Adaptor 用于兼容 TensorFlow、Py Torch 等主流 AI 框架。
- **昇腾计算编译层 (Ascend Computing Compilation Layer) :** 昇腾计算编译层通过图编译器 (Graph Compiler) 将用户输入中间表达 (Intermediate Representa



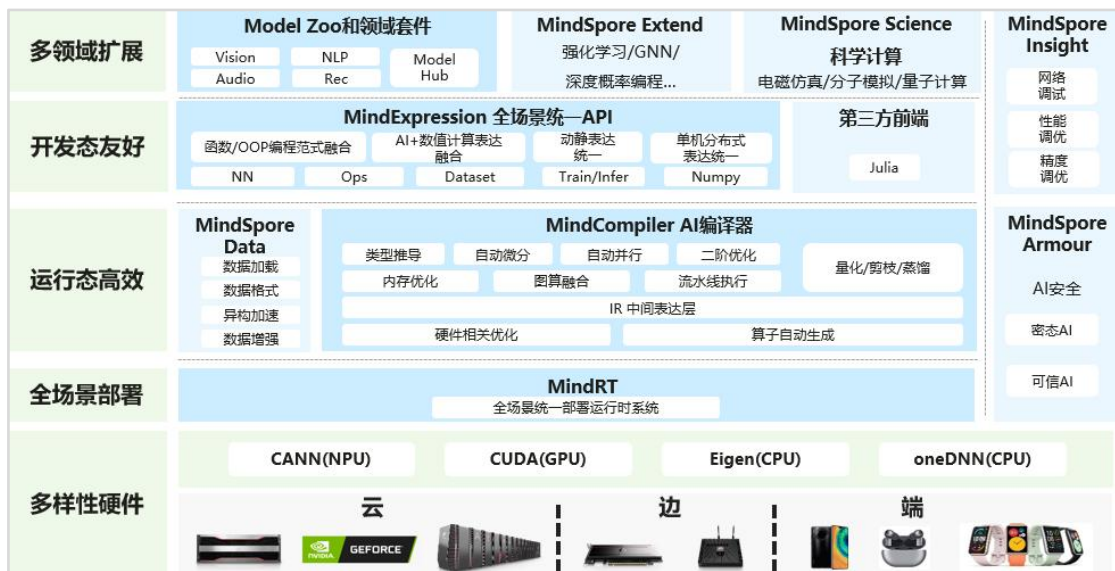
tion, IR) 的计算图编译成昇腾硬件可执行模型; 同时借助张量加速引擎 TBE (Tensor Boost Engine) 的自动调度机制, 高效编译算子。

- **昇腾计算执行层 (Ascend Computing Execution Layer)**: 负责模型和算子的执行, 提供运行时库 (Runtime)、图执行器 (Graph Executor)、数字视觉预处理 (Digital Vision Pre-Processing, DVPP)、人工智能预处理 (Artificial Intelligence Pre-Processing, AIPP)、华为集合通信库 (Huawei Collective Communication Library, HCCL) 等功能单元。
- **昇腾计算基础层 (Ascend Computing Base Layer)**: 主要为其上各层提供基础服务, 如共享虚拟内存 (Shared Virtual Memory, SVM)、设备虚拟化 (Virtual Machine, VM)、主机-设备通信 (Host Device Communication, HDC) 等。

## 1.4. MindSpore 简介

昇思 MindSpore 是一个全场景深度学习框架, 旨在实现易开发、高效执行、全场景统一部署三大目标。

其中, 易开发表现为 API 友好、调试难度低; 高效执行包括计算效率、数据预处理效率和分布式训练效率; 全场景则指框架同时支持云、边缘以及端侧场景。昇思 MindSpore 总体架构如下图所示:



- **ModelZoo (模型库)**: ModelZoo 提供可用的深度学习算法网络, 也欢迎更多开发者贡献新的网络(ModelZoo 地址)。



- **MindSpore Extend（扩展库）**：昇思 MindSpore 的领域扩展库，支持拓展新领域场景，如 GNN/深度概率编程/强化学习等，期待更多开发者来一起贡献和构建。
- **MindSpore Science（科学计算）**：MindScience 是基于昇思 MindSpore 融合架构打造的科学计算行业套件，包含了业界领先的数据集、基础模型、预置高精度模型和前后处理工具，加速了科学行业应用开发。
- **MindExpression（全场景统一 API）**：基于 Python 的前端表达与编程接口，支持两个融合（函数/OOP 编程范式融合、AI+数值计算表达融合）以及两个统一（动静表达统一、单机分布式表达统一）。
- **第三方前端**：支持第三方多语言前端表达，未来计划陆续提供 C/C++ 等第三方前端的对接工作，引入更多的第三方生态。
- **MindSpore Data（数据处理层）**：提供高效的数据处理、常用数据集加载等功能和编程接口，支持用户灵活地定义处理注册和 pipeline 并行优化。
- **MindCompiler（AI 编译器）**：图层的核心编译器，主要基于端云统一的 MindIR 实现三大功能，包括硬件无关的优化（类型推导、自动微分、表达式化简等）、硬件相关优化（自动并行、内存优化、图算融合、流水线执行等）、部署推理相关的优化（量化、剪枝等）。
- **MindRT（全场景运行时）**：昇思 MindSpore 的运行时系统，包含云侧主机侧运行时系统、端侧以及更小 IoT 的轻量化运行时系统。
- **MindSpore Insight（可视化调试调优工具）**：昇思 MindSpore 的可视化调试调优工具，能够可视化地查看训练过程、优化模型性能、调试精度问题、解释推理结果（了解更多）。
- **MindSpore Armour（安全增强库）**：面向企业级运用时，安全与隐私保护相关增强功能，如对抗鲁棒性、模型安全测试、差分隐私训练、隐私泄露风险评估、数据漂移检测等技术（了解更多）。

## 2. 开发板使用介绍

### 2.1. 准备需要的配件

1) TF 卡，最小 32GB 容量的 **class10** 级或以上的高速闪迪卡。强烈推荐使用 64GB 或以上容量的 TF 卡。



2) TF 卡读卡器，用于读写 TF 卡。



3) HDMI 转 HDMI 连接线，用于将开发板连接到 HDMI 显示器或者电视进行显示。



4) Type-C转USB3.0 转接线，用于Type-C接口连接USB3.0 的存储设备。



5) 电源，Type-C 接口的 20V PD-65W 适配器。



6) M.2 M-Key 2280 规格PCIe NVMe SSD。

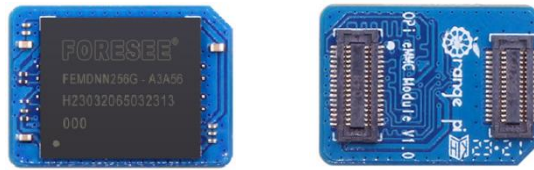
注意，NVMe SSD目前测试了樊想、金士顿和三星的，只有三星的NVMe SSD能稳定运行Linux系统。非三星品牌的NVMe SSD需要等后续软件更新才能正常使用。



7) M.2 M-Key 2280 规格的SATA SSD。



## 8) eMMC模块



## 9) USB接口的鼠标和键盘。



## 10) USB摄像头。



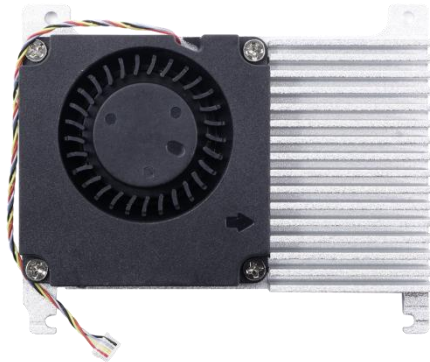
## 11) 金属配套外壳。



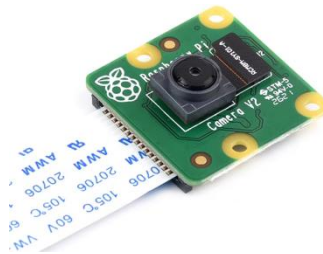
12) 百兆或者千兆网线，用于将开发板连接到因特网。



13) 12V 的散热风扇。



14) 树莓派 IMX219 型号的摄像头。



15) 树莓派 5 寸 MIPI LCD 显示屏。



16) Micro USB 接口的数据线，用于串口调试。



17) 安装有 Ubuntu 22.04 和 Windows 操作系统的 X64 电脑。

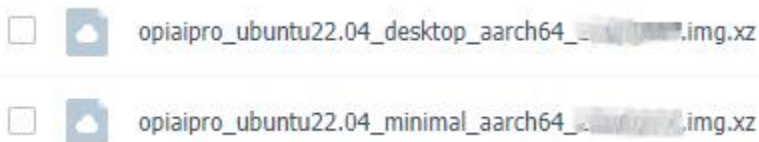
|   |                |                            |
|---|----------------|----------------------------|
| 1 | Ubuntu22.04 PC | 可选，用于编译 Linux 源码           |
| 2 | Windows PC     | 用于烧录 Ubuntu 和 openEuler 镜像 |

## 2.2. 下载开发板系统镜像

注意，Ubuntu 镜像有两个版本，他们的区别是：

1) **minimal** 镜像是一个只有最基础功能的镜像，像 Linux 桌面、CANN 和 AI 示例代码等都没有预装。此镜像只建议想自己从头定制安装 Linux 桌面和 AI 相关软件的开发者使用。

2) **desktop** 镜像预装了 Linux 桌面、CANN、AI 示例代码和一系列测试程序。如果想正常使用开发板的功能，请使用这个镜像。



### 2.2.1. 百度网盘下载

访问开发板资料下载页面，在下方可以看到官方镜像的下载入口：

<http://www.orange-pi.cn/html/hardware/computerAndMicrocontrollers/service-and-support/Orange-Pi-AIpro.html>

官方镜像



ubuntu 镜像

下载



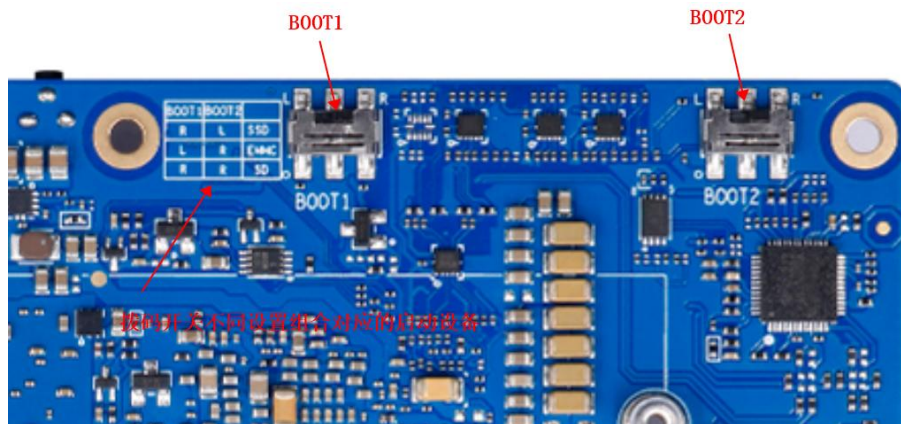
openEuler

openeuler 镜像

下载

## 2.3. 控制启动设备的两个拨码开关的使用说明

开发板支持从 TF 卡、eMMC 和 SSD（支持 NVMe SSD 和 SATA SSD）启动。具体从哪个设备启动是由开发板背面的两个拨码（BOOT1 和 BOOT2）开关来控制的。



BOOT1 和 BOOT2 两个拨码开关都支持左右两种设置状态，所以总共有 4 种设置状态，开发板目前只使用了其中的三种。不同的设置状态对应的启动设备如下表所示：

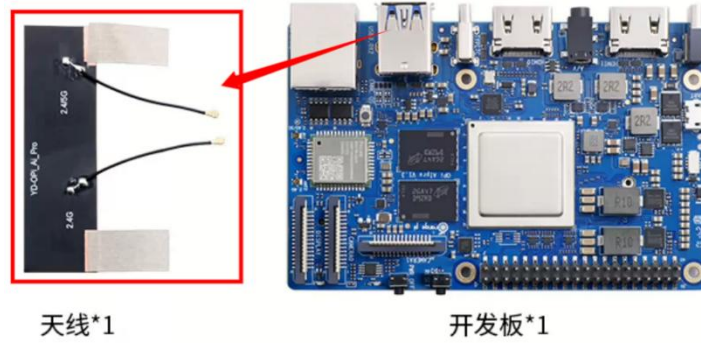
| 拨码开关 BOOT1 | 拨码开关 BOOT2 | 对应的启动设备             |
|------------|------------|---------------------|
| 左          | 左          | 未使用                 |
| 右          | 左          | SATA SSD 和 NVMe SSD |
| 左          | 右          | eMMC                |
| 右          | 右          | TF 卡                |

注意，SATA SSD和NVMe SSD的启动方式对应的拨码开关的设置状态是一样的。这两种启动方式是通过M2\_TYPE引脚的电平来自动区分的。

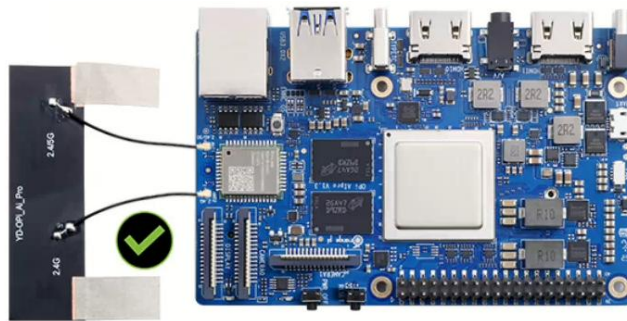
另外请注意，切换拨码开关后必须重新拔插电源上下电才能让新的启动设备选项生效。通过开发板的复位按键来复位系统是不会让拨码开关新设置的配置生效的。

## 2.4. WIFI 蓝牙天线使用注意事项

开发板的 WIFI 蓝牙天线如下图左边所示：

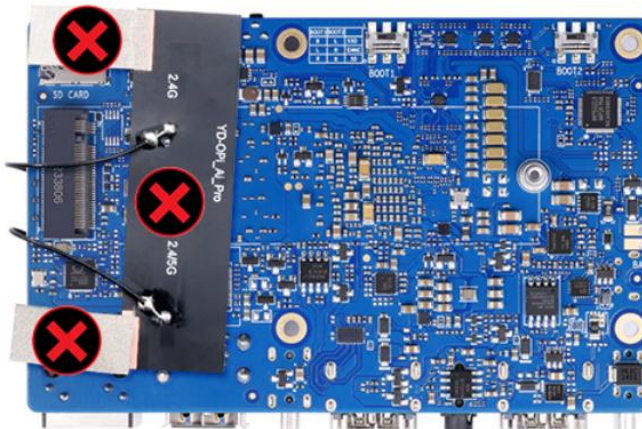


WIFI 蓝牙天线的正确安装方法如下图所示：



※天线正确安装和使用放置示意图。

安装好天线后，请不要像下图所示的一样将天线贴到开发板的背面，同时天线上的导电布也不能挨着开发板，否则可能会烧坏开发板。



## 2.5. 镜像烧录

### 2.5.1. 烧写 Linux 镜像到 TF 卡中的方法

#### 2.5.1.1. 基于 Windows PC 将 Linux 镜像烧写到 TF 卡的方法

注意，这里说的Linux镜像具体指的是按照[下载开发板系统镜像](#)一小节的内容，下载的Ubuntu或者openEuler镜像。

1) 首先准备一张 32GB 或更大容量的 TF 卡（推荐使用 **64GB 或以上容量的 TF 卡**），TF 卡的传输速度必须为 **class10** 级或 **class10** 级以上，建议使用闪迪等品牌的 TF 卡。

2) 然后把 TF 卡插入读卡器，再把读卡器插入电脑。

3) 参考[下载开发板系统镜像](#)一小节的内容，下载想要烧录的 Linux 镜像压缩包。

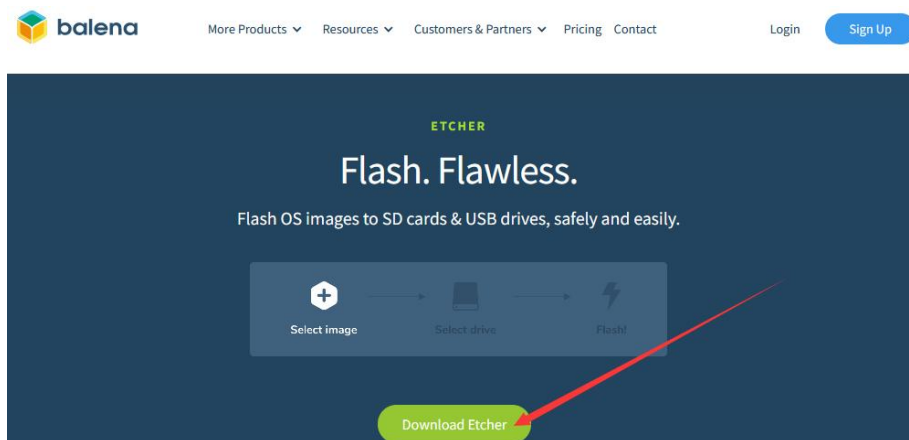
4) 然后下载用于烧录 Linux 镜像的软件——**balenaEtcher**，下载地址为：

<https://www.balena.io/etcher/>

5) 如果 balenaEtcher 官网访问比较慢，可以使用下面的链接下载。

<https://obs-9be7.obs.cn-east-2.myhuaweicloud.com/OrangePi/balenaEtcher/balenaEtcher-Setup-1.18.4.exe>

6) 进入 balenaEtcher 下载页面后，点击绿色的下载按钮会跳到软件下载的地方。



7) 然后可以选择下载 Portable 版本的 balenaEtcher，Portable 版本无需安装，双击打开就可以使用。

**DOWNLOAD**

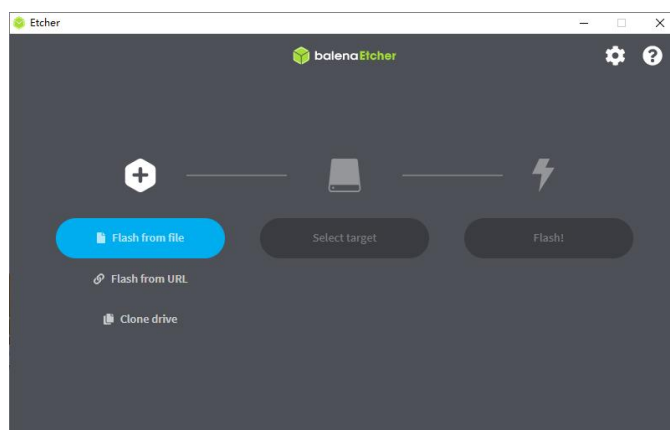
## Download Etcher

| ASSET                                                   | OS      | ARCH    |                          |
|---------------------------------------------------------|---------|---------|--------------------------|
| ETCHER FOR WINDOWS (X86 X64) (INSTALLER)                | WINDOWS | X86 X64 | <a href="#">Download</a> |
| ETCHER FOR WINDOWS (X86 X64) (PORTABLE)                 | WINDOWS | X86 X64 | <a href="#">Download</a> |
| ETCHER FOR WINDOWS (LEGACY 32 BIT) (X86 X64) (PORTABLE) | WINDOWS | X86 X64 | <a href="#">Download</a> |
| ETCHER FOR MACOS                                        | MACOS   | X64     | <a href="#">Download</a> |
| ETCHER FOR LINUX X64 (64-BIT) (APPIMAGE)                | LINUX   | X64     | <a href="#">Download</a> |
| ETCHER FOR LINUX (LEGACY 32 BIT) (APPIMAGE)             | LINUX   | X86     | <a href="#">Download</a> |

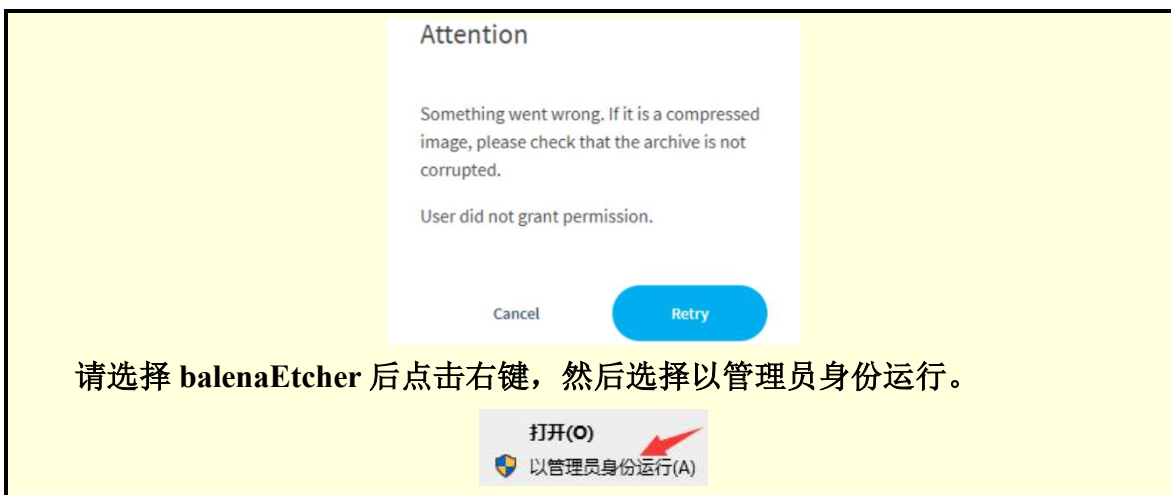
Looking for [Debian \(.deb\) packages](#) or [Red Hat \(.rpm\) packages](#)

OSS hosting by [cloudsmith](#)

8) 打开后的 balenaEtcher 界面如下图所示：

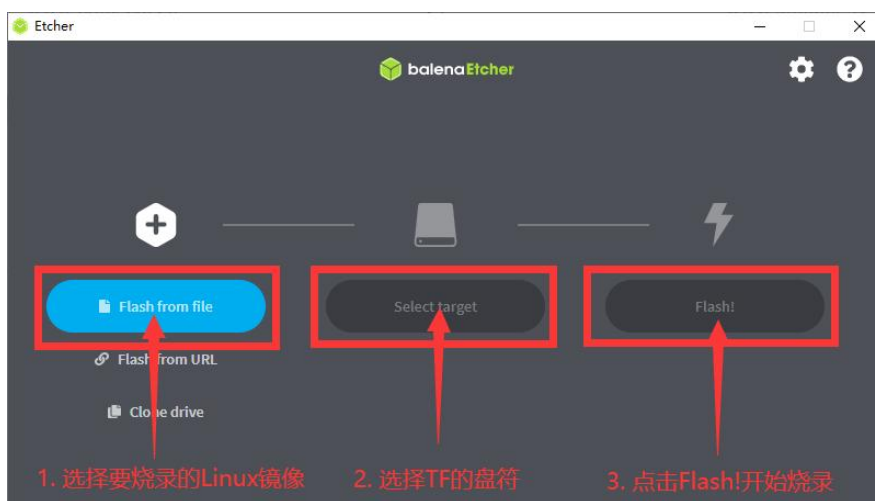


打开 balenaEtcher 时如果提示下面的错误：

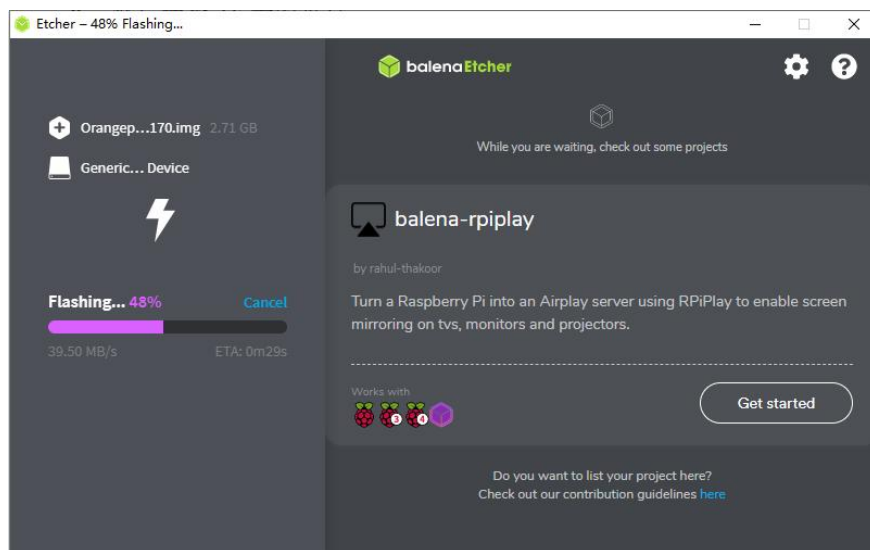


9) 使用 balenaEtcher 烧录 Linux 镜像的具体步骤如下所示：

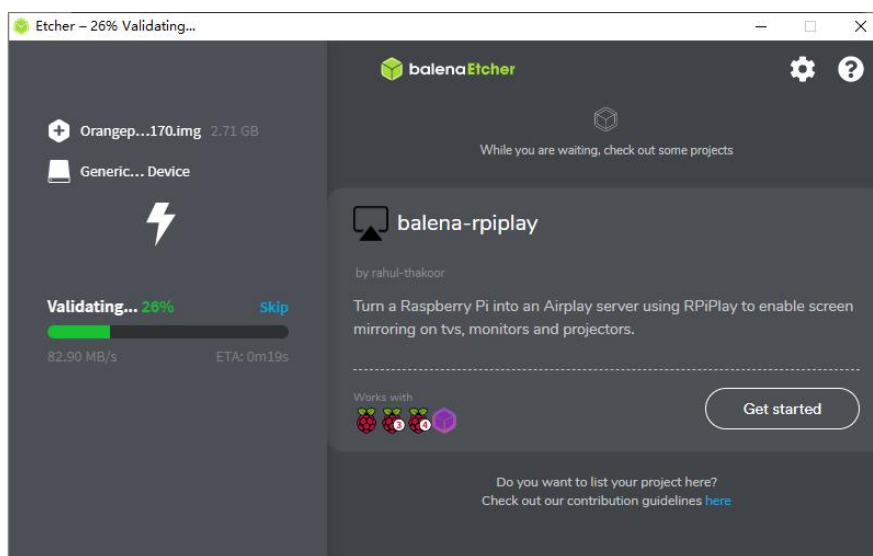
- a. 首先选择要烧录的 Linux 镜像文件的路径。
- b. 然后选择 TF 卡的盘符。
- c. 最后点击 Flash 就会开始烧录 Linux 镜像到 TF 卡中。



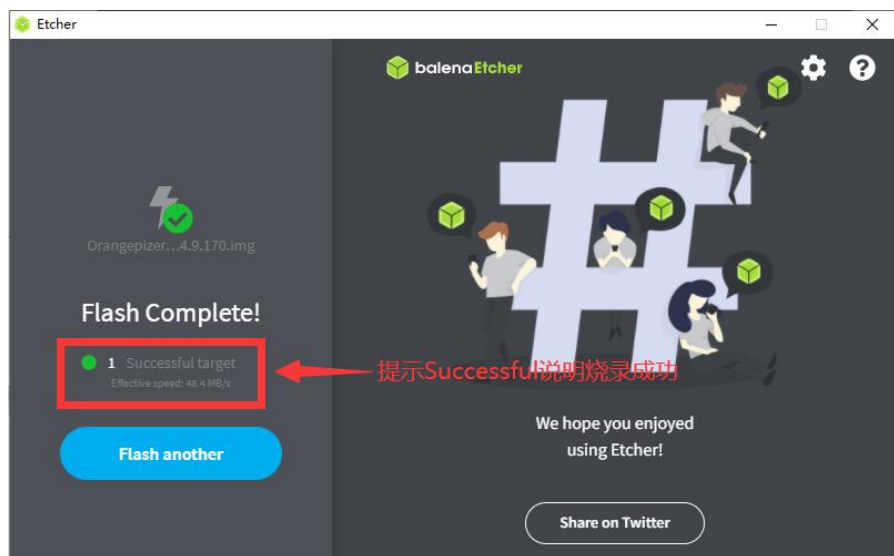
10) balenaEtcher 烧录 Linux 镜像的过程显示的界面如下图所示，另外进度条显示紫色表示正在烧录 Linux 镜像到 TF 卡中。



11) Linux 镜像烧录完后，balenaEtcher 默认还会对烧录到 TF 卡中的镜像进行校验，确保烧录过程没有出问题。如下图所示，显示绿色的进度条就表示镜像已经烧录完成，balenaEtcher 正在对烧录完成的镜像进行校验。



12) 成功烧录完成后 balenaEtcher 的显示界面如下图所示，如果显示绿色的指示图标说明镜像烧录成功，此时就可以退出 balenaEtcher，然后拔出 TF 卡插入到开发板的 TF 卡槽中使用了。



注意，启动系统前请确保拨码开关拨到了TF卡启动的位置了。拨码开关的使用说明请参考[控制启动设备的两个拨码开关的使用说明](#)一小节的说明。

### 2.5.1.2. 基于 Ubuntu PC 将 Linux 镜像烧写到 TF 卡的方法

注意，这里说的Linux镜像具体指的是按照[下载开发板系统镜像](#)一小节的内容，下载的Ubuntu或者openEuler镜像。

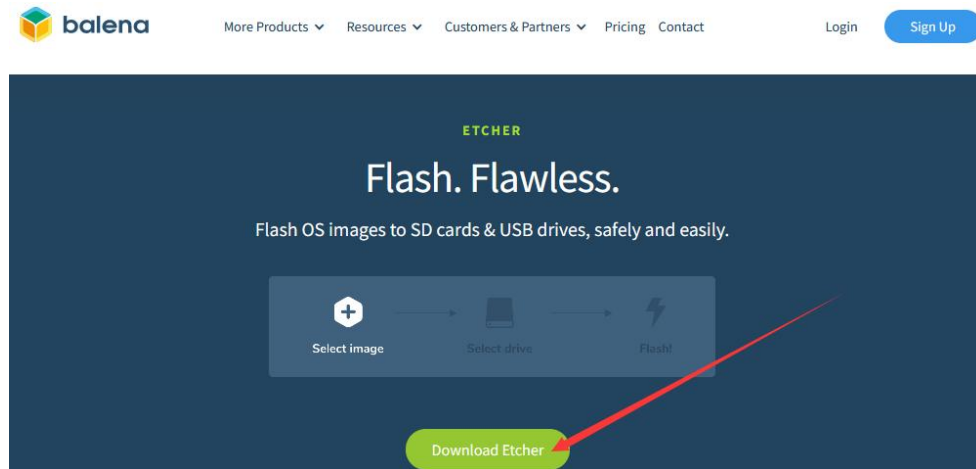
1) 首先准备一张 32GB 或更大容量的 TF 卡（推荐使用 **64GB 或以上容量的 TF 卡**），TF 卡的传输速度必须为 **class10** 级或 **class10** 级以上，建议使用闪迪等品牌的 TF 卡。

2) 然后把 TF 卡插入读卡器，再把读卡器插入电脑。

3) 下载 balenaEtcher 软件，下载地址为：

<https://www.balena.io/etcher/>

4) 进入 balenaEtcher 下载页面后，点击绿色的下载按钮会跳到软件下载的地方。



5) 然后选择下载 Linux 版本的软件即可。

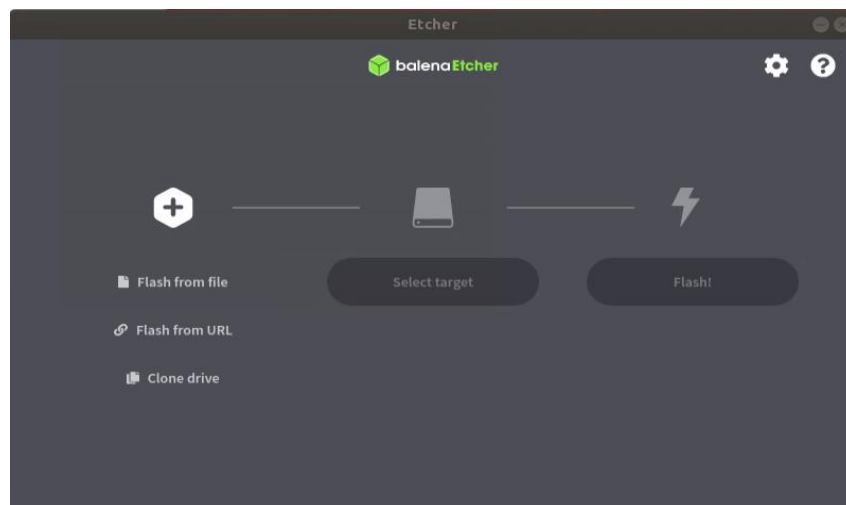
**DOWNLOAD**

## Download Etcher

| ASSET                                                   | OS      | ARCH    |                          |
|---------------------------------------------------------|---------|---------|--------------------------|
| ETCHER FOR WINDOWS (X86 X64) (INSTALLER)                | WINDOWS | X86 X64 | <a href="#">Download</a> |
| ETCHER FOR WINDOWS (X86 X64) (PORTABLE)                 | WINDOWS | X86 X64 | <a href="#">Download</a> |
| ETCHER FOR WINDOWS (LEGACY 32 BIT) (X86 X64) (PORTABLE) | WINDOWS | X86 X64 | <a href="#">Download</a> |
| ETCHER FOR MACOS                                        | MACOS   | X64     | <a href="#">Download</a> |
| ETCHER FOR LINUX X64 (64-BIT) (APPIMAGE)                | LINUX   | X64     | <a href="#">Download</a> |
| ETCHER FOR LINUX (LEGACY 32 BIT) (APPIMAGE)             | LINUX   | X86     | <a href="#">Download</a> |

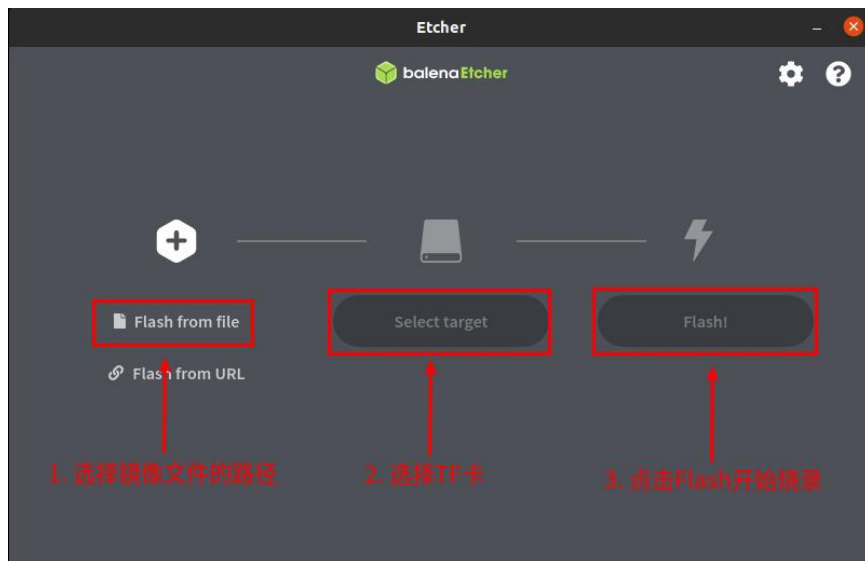
6) 参考[下载开发板系统镜像](#)一小节的内容，下载想要烧录的 Linux 镜像文件压缩包。

7) 然后在 Ubuntu PC 的图形界面双击 **balenaEtcher-x.x.x-x64.AppImage** 即可打开 balenaEtcher，balenaEtcher 打开后的界面显示如下图所示：

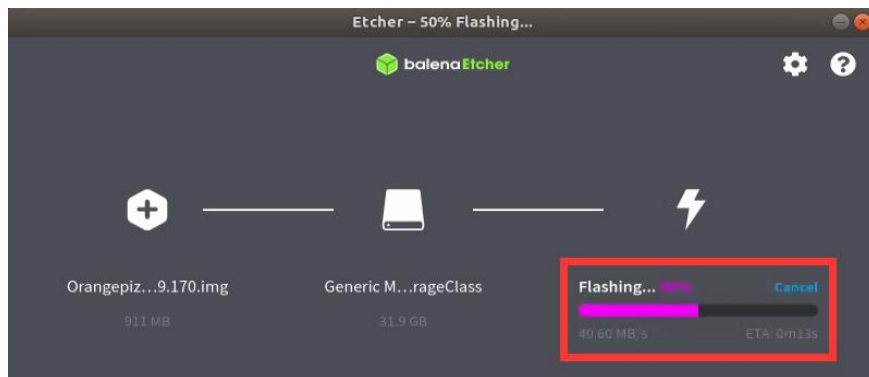


8) 使用 balenaEtcher 烧录 Linux 镜像的具体步骤如下所示：

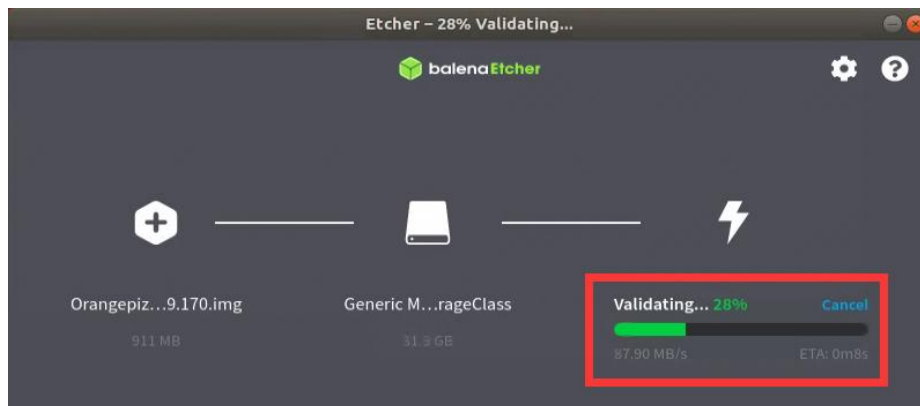
- a. 首先选择要烧录的 Linux 镜像文件的路径。
- b. 然后选择 TF 卡的盘符。
- c. 最后点击 Flash 就会开始烧录 Linux 镜像到 TF 卡中。



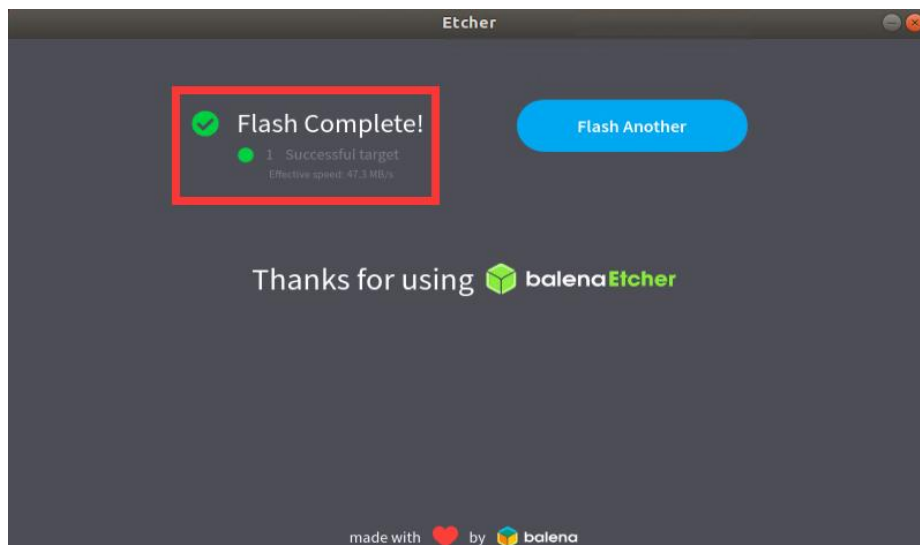
9) balenaEtcher 烧录 Linux 镜像的过程显示的界面如下图所示，另外进度条显示紫色表示正在烧录 Linux 镜像到 TF 卡中。



10) Linux 镜像烧录完后，balenaEtcher 默认还会对烧录到 TF 卡中的镜像进行校验，确保烧录过程没有出问题。如下图所示，显示绿色的进度条就表示镜像已经烧录完成，balenaEtcher 正在对烧录完成的镜像进行校验。



11) 成功烧录完成后 balenaEtcher 的显示界面如下图所示，如果显示绿色的指示图标说明镜像烧录成功，此时就可以退出 balenaEtcher，然后拔出 TF 卡插入到开发板的 TF 卡槽中使用了。



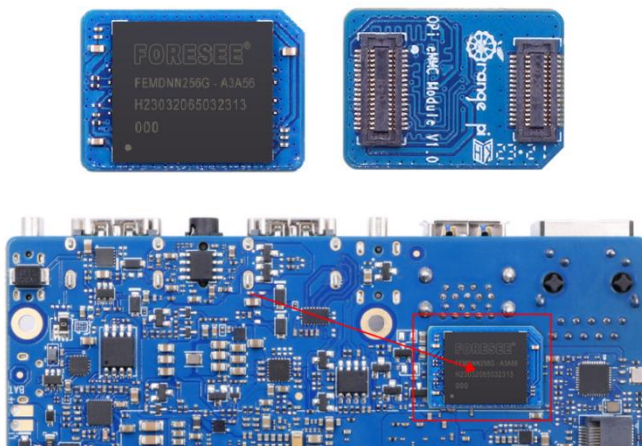
注意，启动系统前请确保拨码开关拨到了TF卡启动的位置了。拨码开关的使用说明请参考[控制启动设备的两个拨码开关的使用说明](#)一小节的说明。

## 2.5.2. 烧写 Linux 镜像到 eMMC 中的方法

注意，这里说的Linux镜像具体指的是按照[下载开发板系统镜像](#)一小节的内容，下载的Ubuntu或者openEuler镜像。

1) 开发板预留了 eMMC 模块的扩展接口，烧录系统到 eMMC 前，需要先购买一个与开发板 eMMC 接口相匹配的 eMMC 模块。然后将 eMMC 模块安装到开发板上。

配套的 eMMC 模块和插入开发板的方法如下所示：



2) 烧录 Linux 镜像到 eMMC 中需要借助 TF 卡来完成，所以首先需要将 Linux 镜像烧录到 TF 卡上，然后使用 TF 卡启动开发板进入 Linux 系统。烧录 Linux 镜像到 TF 卡的方法请见[基于 Windows PC 将 Linux 镜像烧写到 TF 卡的方法](#)和[基于 Ubuntu PC 将 Linux 镜像烧写到 TF 卡的方法](#)两小节的说明。

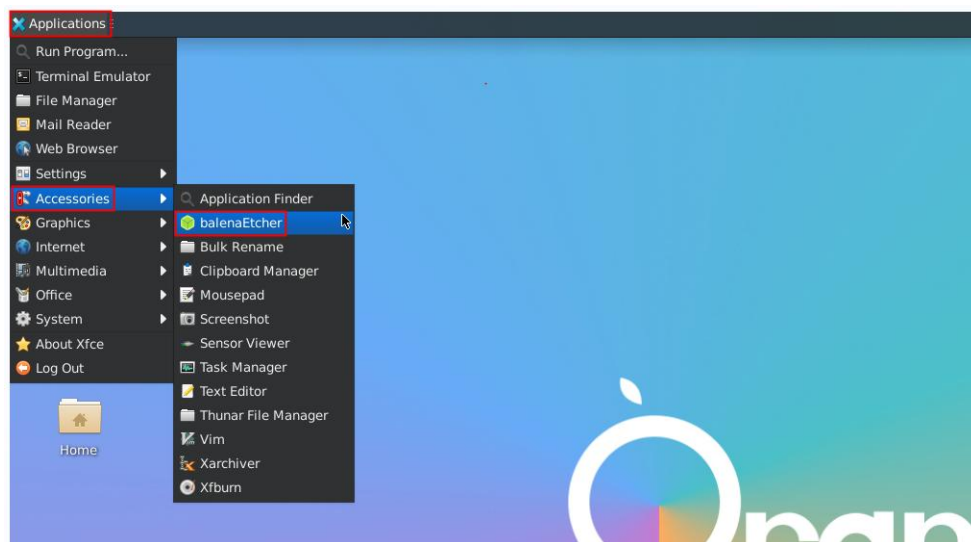
3) 启动进入 TF 卡的 Linux 系统后，请先确认下 eMMC 模块已经被开发板的 Linux 系统正常识别了。如果 eMMC 模块正常识别了的话，在 root 用户下使用 **fdisk -l** 命令就能看到 eMMC 模块的容量信息。

```
(base) root@orangepiaipro:~# fdisk -l
.....
Disk /dev/mmcblk0: 28.91 GiB, 31037849600 bytes, 60620800 sectors
.....
```

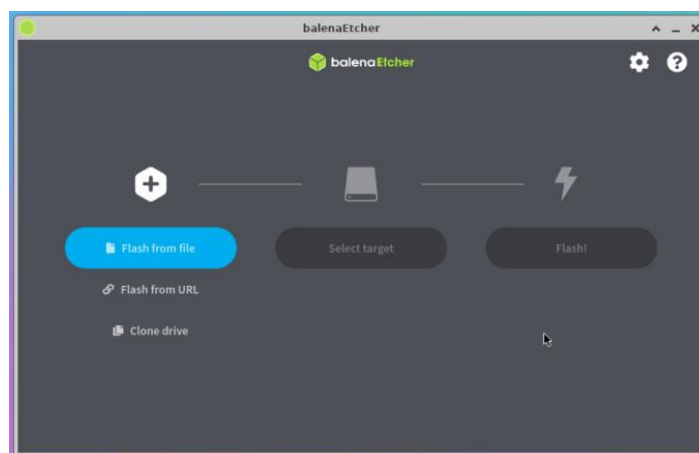
4) 然后将要烧录的 Linux 镜像文件压缩包上传到 TF 卡的 Linux 系统中。

**注意，使用xz格式压缩的Linux镜像压缩包不需要解压，balenaEtcher烧录时会自动解压。**

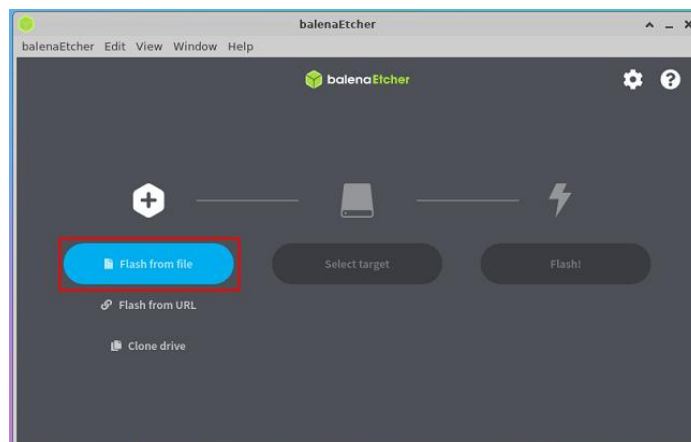
5) 然后就可以开始使用 balenaEtcher 软件烧录镜像到 eMMC 中了。Linux 系统中已经预装了 balenaEtcher，打开方法如下所示：



6) balenaEtcher 打开后的界面如下所示:



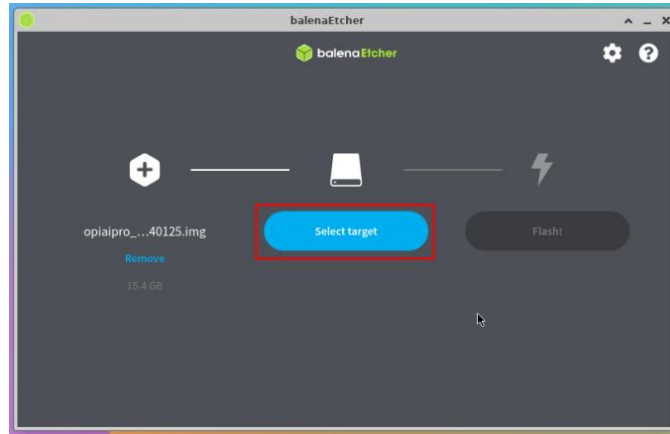
7) 然后点击 **Flash from file** 选择前面上传的想要烧录的镜像文件。



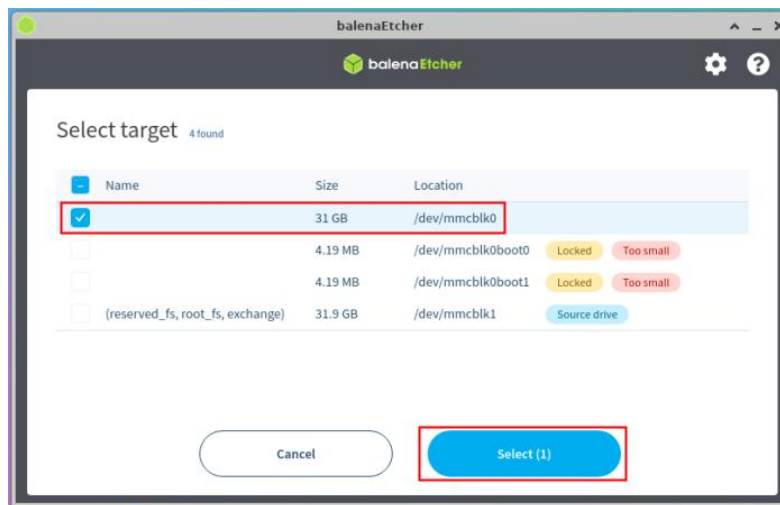


如果打开Linux镜像文件时提示没有权限，请使用 `sudo chmod 777` 镜像文件名 这条命令来给镜像文件添加权限。

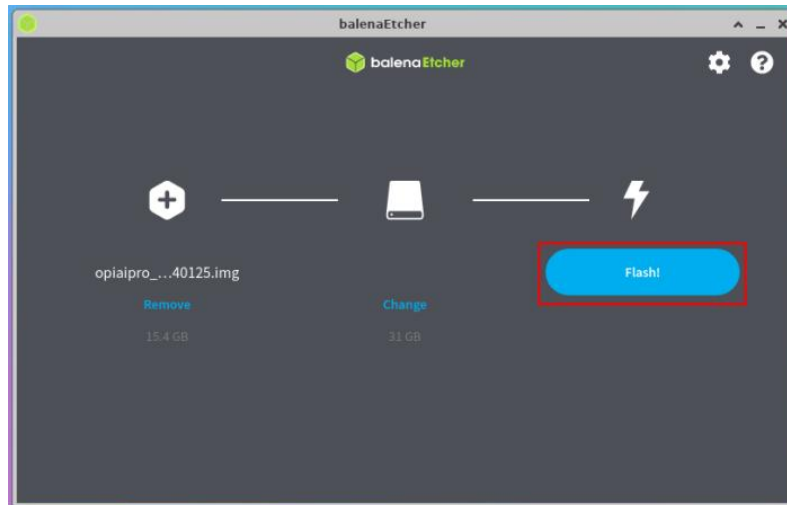
8) 然后点击 **Select target**。



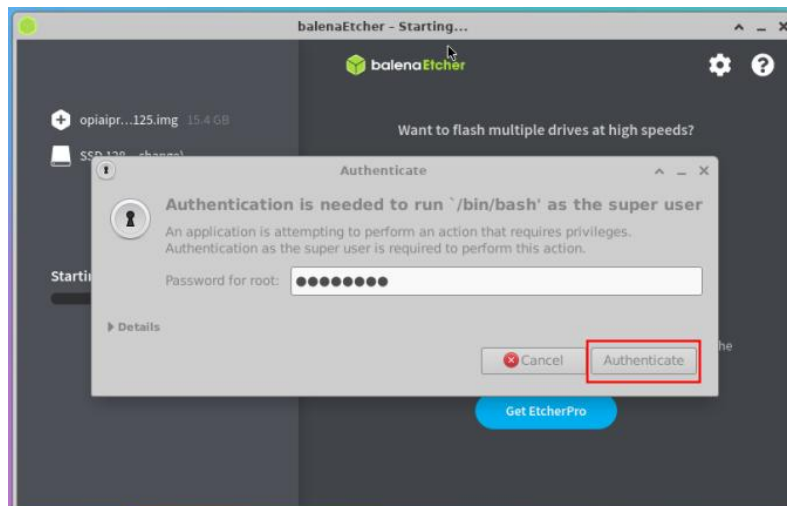
9) 然后选择 eMMC 对应的 `/dev/mmcblk0` 选项，再点击 **Select** 即可。



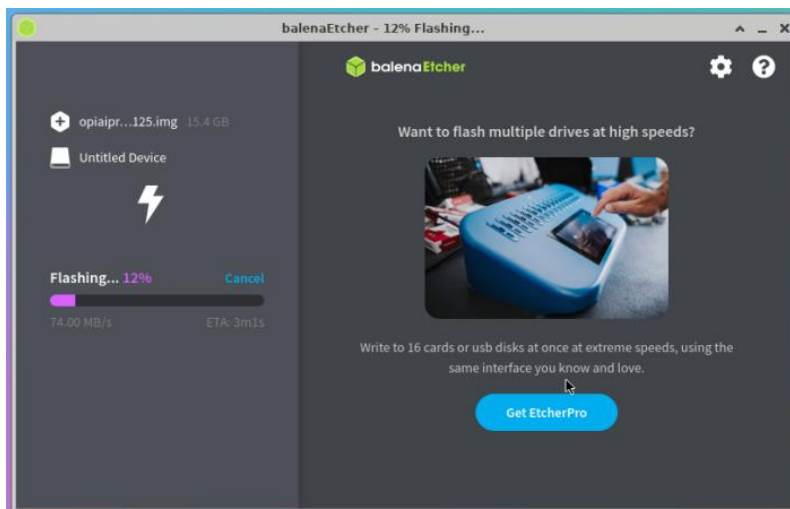
10) 然后点击 **Flash!** 开始烧录。



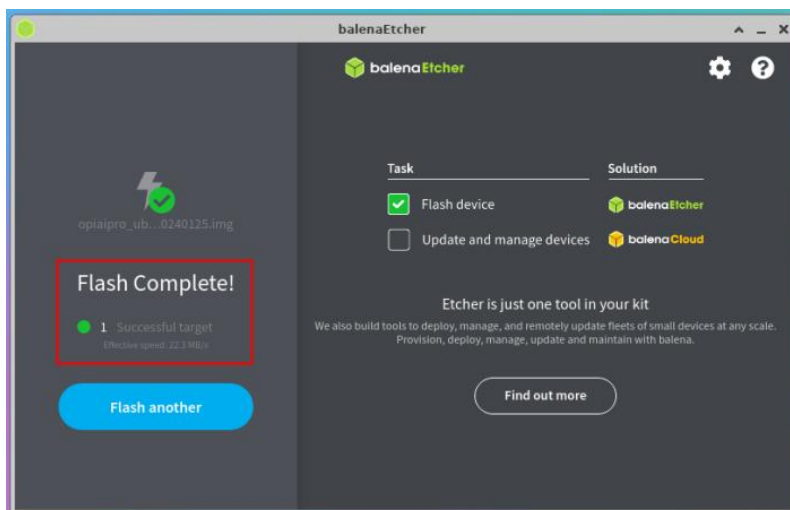
11) 然后输入 Linux 系统的密码: **Mind@123**。



12) 然后就会真正开始烧录 Linux 镜像到 eMMC 中了。



13) Linux 镜像烧录完后的显示如下所示：



14) 此时就可以关闭掉 Linux 系统，**然后拔出 TF 卡**，并断开 Type-C 电源。再将两个拨码开关拨到 eMMC 启动对应的位置，然后重新插入 Type-C 电源就可以启动 eMMC 中的 Linux 系统了。

注意，启动系统前请确保拨码开关拨到eMMC启动的位置了。拨码开关的使用说明请参考[控制启动设备的两个拨码开关的使用说明](#)一小节的说明。

### 2.5.3. 烧写 Linux 镜像到 NVMe SSD 中的方法

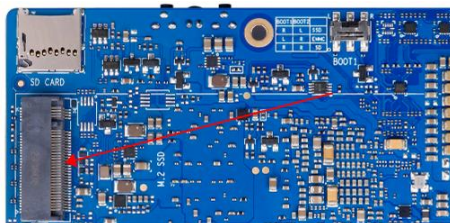
注意，这里说的Linux镜像具体指的是按照[下载开发板系统镜像](#)一小节的内容，下载的Ubuntu或者openEuler镜像。

注意，NVMe SSD目前测试了樊想、金士顿和三星的，只有三星的NVMe SSD能稳定运行Linux系统。非三星品牌的NVMe SSD需要等后续软件更新才能正常使用。

1) 首先需要准备一个 2280 规格 NVMe SSD，开发板 M.2 插槽支持的 PCIe 规格为 PCIe3.0x4。PCIe3.0 和 PCIe4.0 的 NVMe SSD 都是可以用的，只是 PCIe4.0 SSD 速度最高只有 PCIe3.0x4 的速度。2242 等其他规格的 SSD 也都是可以使用的，只是没法固定。



2) 然后把 NVMe SSD 插入开发板的 M.2 接口中，并固定好。



3) 烧录 Linux 镜像到 NVMe SSD 中需要借助 TF 卡来完成，所以首先需要将 Linux 镜像烧录到 TF 卡上，然后使用 TF 卡启动开发板进入 Linux 系统。烧录 Linux 镜像到 TF 卡的方法请见[基于 Windows PC 将 Linux 镜像烧写到 TF 卡的方法](#)和[基于 Ubuntu PC 将 Linux 镜像烧写到 TF 卡的方法](#)两小节的说明。

4) 启动进入 TF 卡的 Linux 系统后，请先确认下 NVMe SSD 已经被开发板的 Linux 系统正常识别了。如果 NVMe SSD 正常识别了的话，使用 `sudo fdisk -l` 命令就能看到 `nvme` 相关的信息。

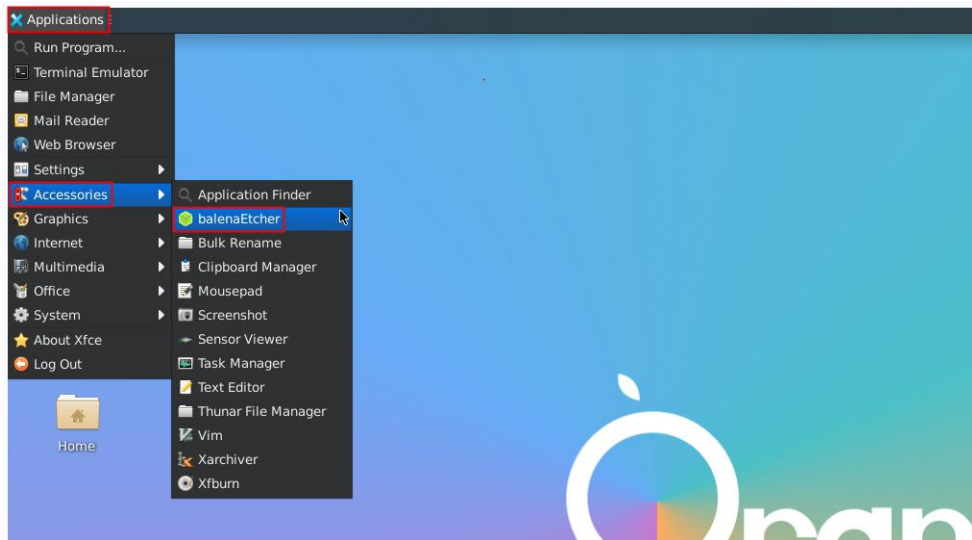
```
(base) HwHiAiUser@orangepiaipro:~$ sudo fdisk -l | grep "nvme0n1"
Disk /dev/nvme0n1: 238.47 GiB, 256060514304 bytes, 500118192 sectors
.....
```

5) 然后将要烧录的 Linux 镜像文件压缩包上传到 TF 卡的 Linux 系统中。

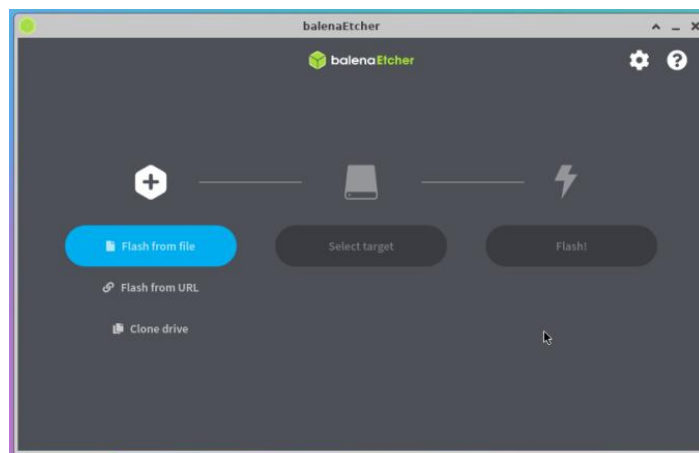
注意，使用xz格式压缩的Linux镜像压缩包不需要解压，balenaEtcher烧录时会自动解压。



6) 然后就可以开始使用 balenaEtcher 软件烧录镜像到 SSD 中了。Linux 系统中已经预装了 balenaEtcher，打开方法如下所示：

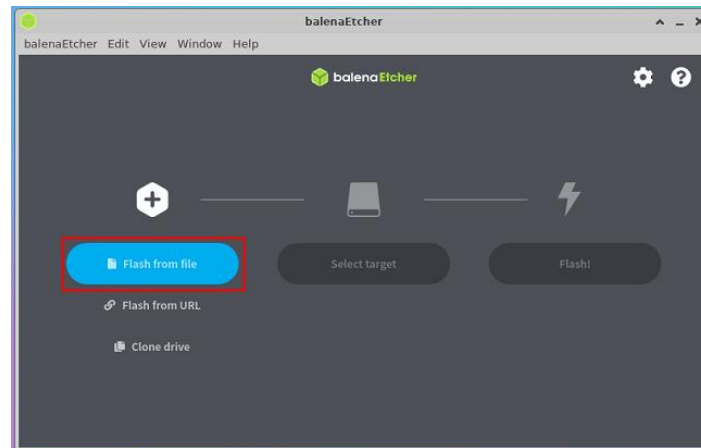


7) balenaEtcher 打开后的界面如下所示：

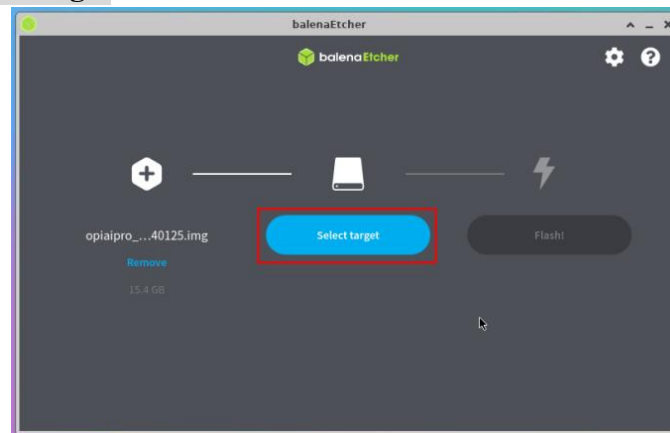


如果打开Linux镜像文件时提示没有权限，请使用 `sudo chmod 777 镜像文件名` 这条命令来给镜像文件添加权限。

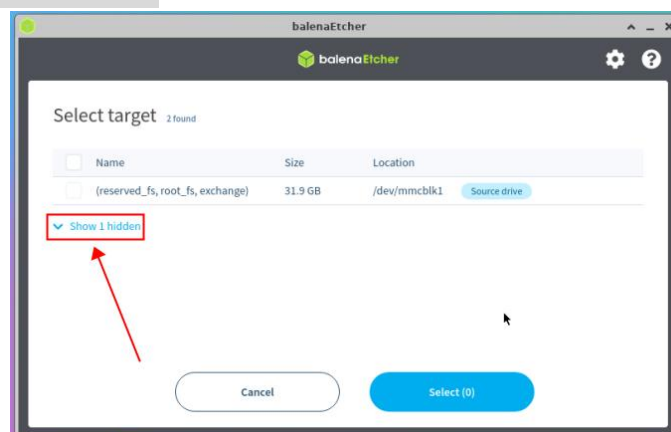
8) 然后点击 **Flash from file** 选择前面上传的想要烧录的镜像文件。



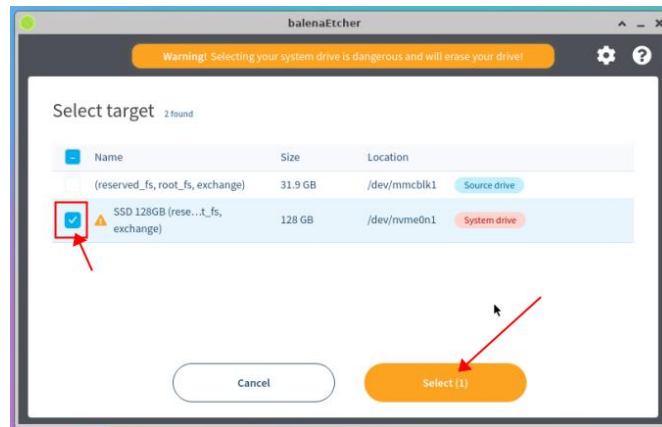
9) 然后点击 **Select target**。



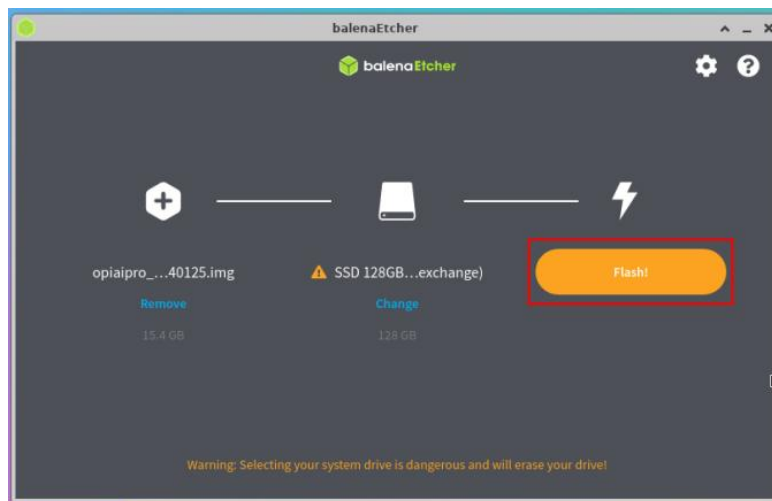
10) 然后点击 **Show 1 hidden**。



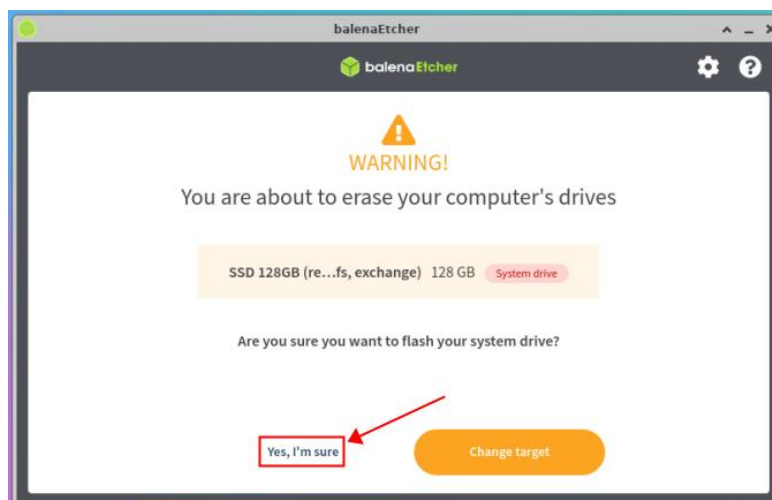
11) 然后选择 SSD 对应的选项，再点击 **Select** 即可。



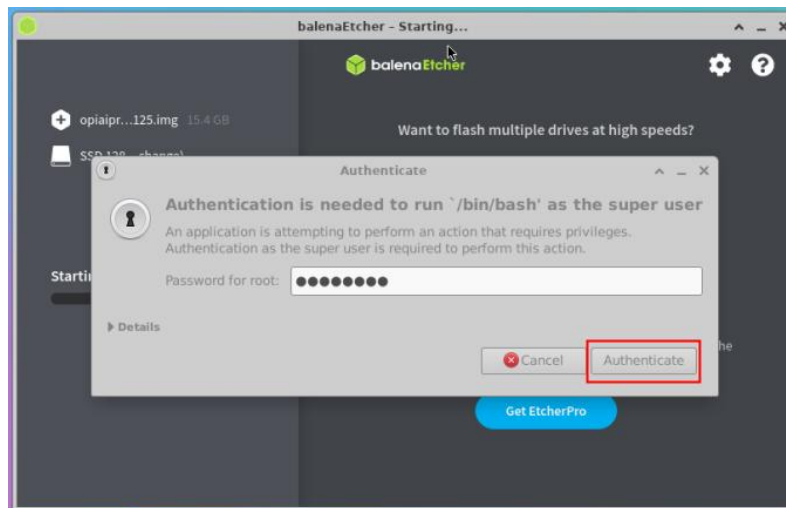
12) 然后点击 **Flash!**开始烧录。



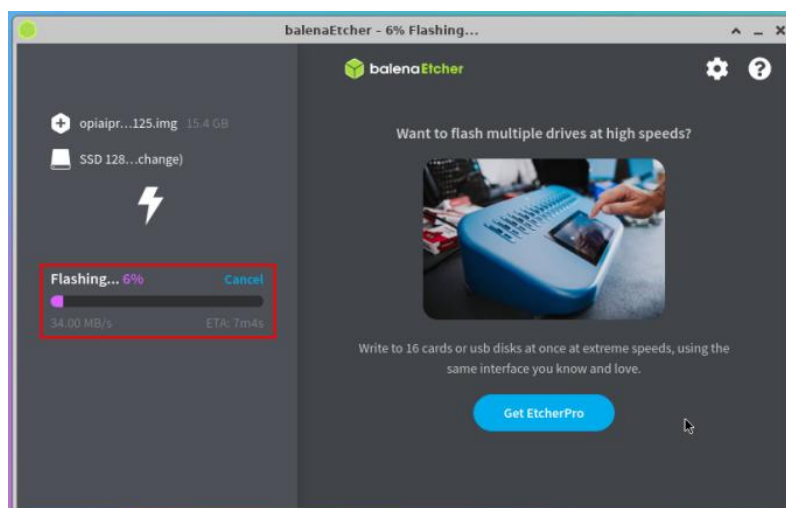
13) 然后选择 **Yes, I'm sure**。



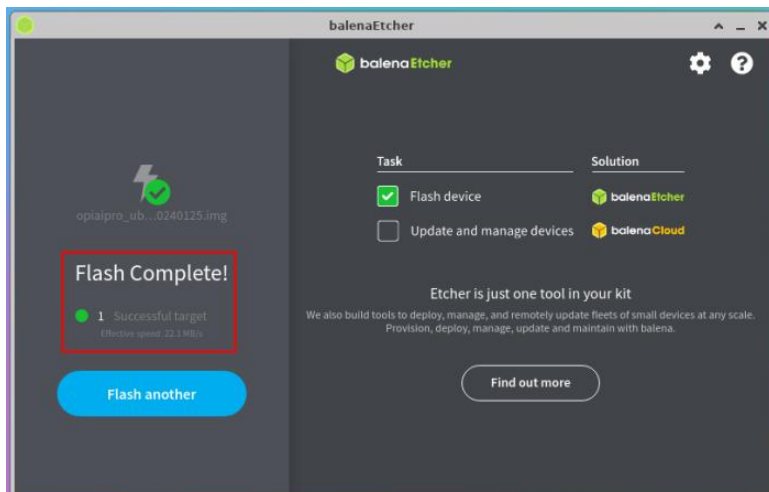
14) 然后输入 Linux 系统的密码: **Mind@123**。



15) 然后就会真正开始烧录 Linux 镜像到 SSD 中了。



16) Linux 镜像烧录完后的显示如下所示:



17) 此时就可以关闭掉 Linux 系统，然后拔出 TF 卡，并断开 Type-C 电源。再将两个拨码开关拨到 SSD 启动对应的位置，然后重新插入 Type-C 电源就可以启动 SSD 中的 Linux 系统了。

注意，启动系统前请确保拨码开关拨到了SSD启动的位置了。拨码开关的使用说明请参考[控制启动设备的两个拨码开关的使用说明](#)一小节的说明。

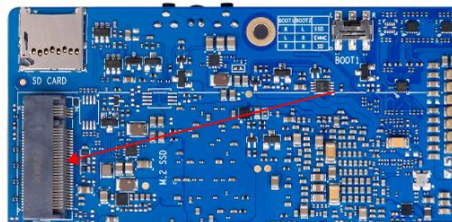
#### 2.5.4. 烧写 Linux 镜像到 SATA SSD 中的方法

注意，这里说的Linux镜像具体指的是按照[下载开发板系统镜像](#)一小节的内容，下载的Ubuntu或者openEuler镜像。

1) 首先需要准备一个 M.2 2280 规格的 SATA SSD。2242 等其他规格的 SSD 也都是可以使用的，只是没法固定。



2) 然后把 SATA SSD 插入开发板的 M.2 接口中，并固定好。



3) 烧录 Linux 镜像到 SATA SSD 中需要借助 TF 卡来完成，所以首先需要将 Linux 镜像烧录到 TF 卡上，然后使用 TF 卡启动开发板进入 Linux 系统。烧录 Linux 镜像到 TF 卡的方法请见[基于 Windows PC 将 Linux 镜像烧写到 TF 卡的方法](#)和[基于 Ubuntu PC 将 Linux 镜像烧写到 TF 卡的方法](#)两小节的说明。

4) 启动进入 TF 卡的 Linux 系统后，首先需要更新下 SATA 对应的 dt.img 文件。步骤如下所示：

- a. 首先进入 `/opt/opi_test/sata` 文件夹。

```
(base) HwHiAiUser@orangepiaipro:~$ cd /opt/opi_test/sata
```

- b. 然后运行下 `update.sh` 脚本来更新 SATA 对应的 dt.img。

```
(base) HwHiAiUser@orangepiaipro:/opt/opi_test/sata$ sudo ./update.sh
```

- c. 运行完 `update.sh` 脚本后会自动重启 Linux 系统让配置生效。

- d. 一切顺利的话，重新进入 TF 卡的 Linux 系统后就能识别到 SATA SSD 了。

```
(base) HwHiAiUser@orangepiaipro:~$ sudo fdisk -l | grep "/dev/sd"
```

```
Disk /dev/sda: 238.47 GiB, 256060514304 bytes, 500118192 sectors
```

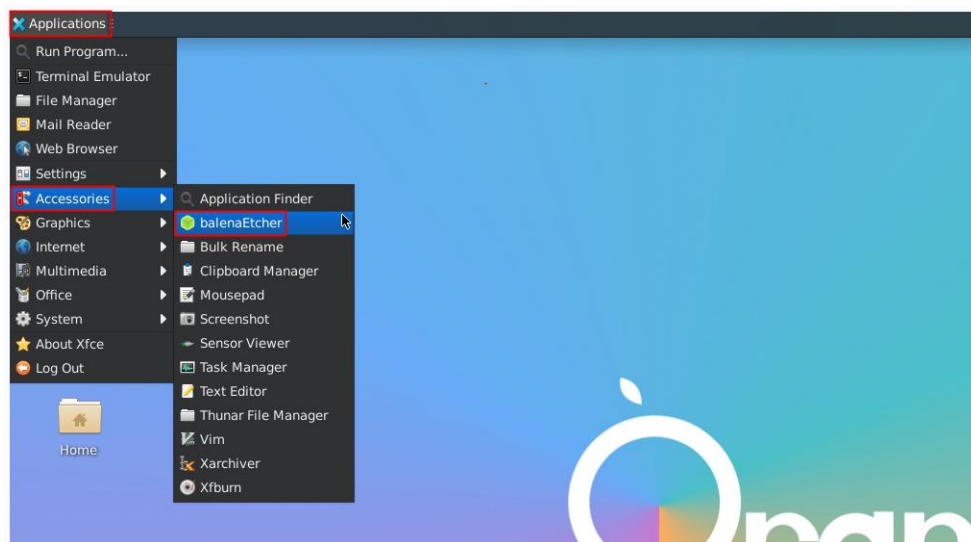
```
.....
```

开发板的 M.2 接口既支持 NVMe SSD 也支持 SATA SSD，只能二选一。Linux 系统中 dt.img 的配置默认打开的是 PCIe 的配置，如果 M.2 接口接的是 SATA SSD，需要手动切换下 SATA 对应的 dt.img 才能正常使用。

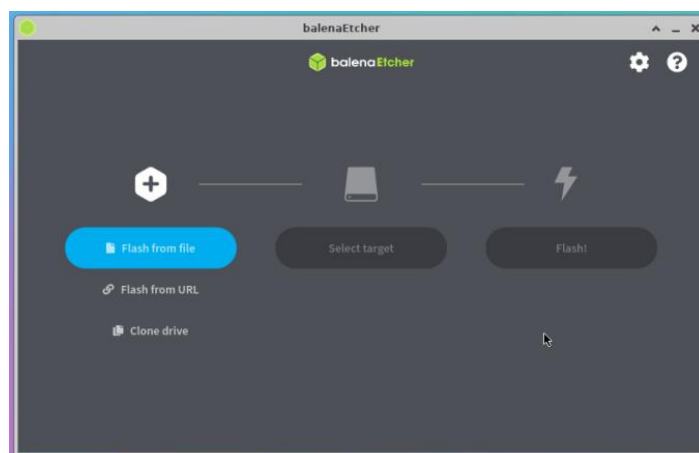
5) 然后将要烧录的 Linux 镜像文件压缩包上传到 TF 卡的 Linux 系统中。

注意，使用 xz 格式压缩的 Linux 镜像压缩包不需要解压，balenaEtcher 烧录时会自动解压。

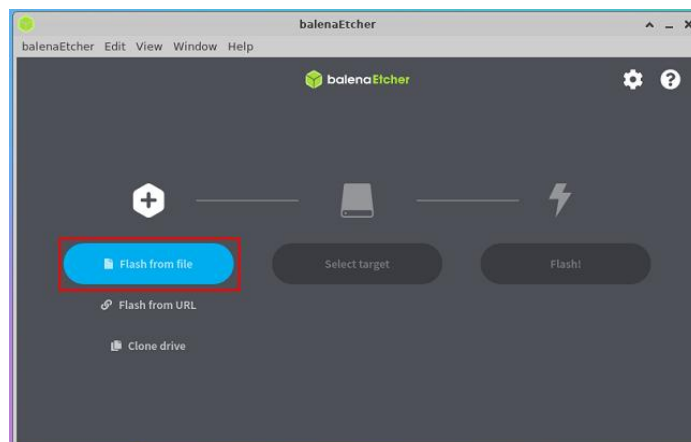
6) 然后就可以开始使用 balenaEtcher 软件烧录镜像到 SSD 中了。Linux 系统中已经预装了 balenaEtcher，打开方法如下所示：



7) balenaEtcher 打开后的界面如下所示:

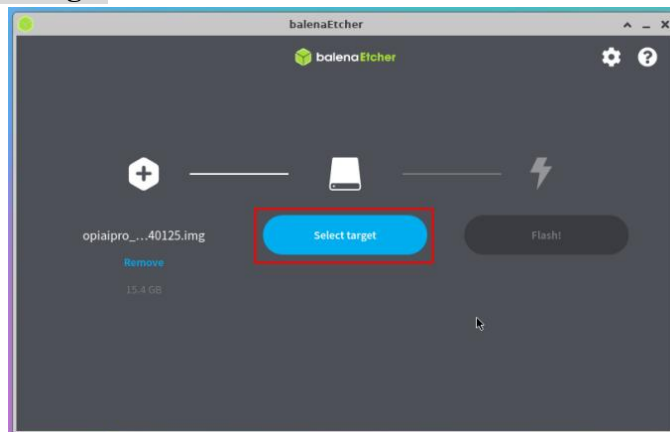


8) 然后点击 **Flash from file** 选择前面上传的想要烧录的镜像文件。

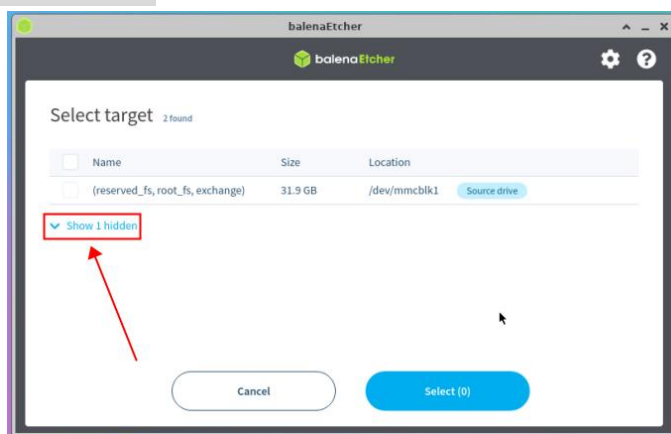


如果打开Linux镜像文件时提示没有权限，请使用 `sudo chmod 777` 镜像文件名 这条命令来给镜像文件添加权限。

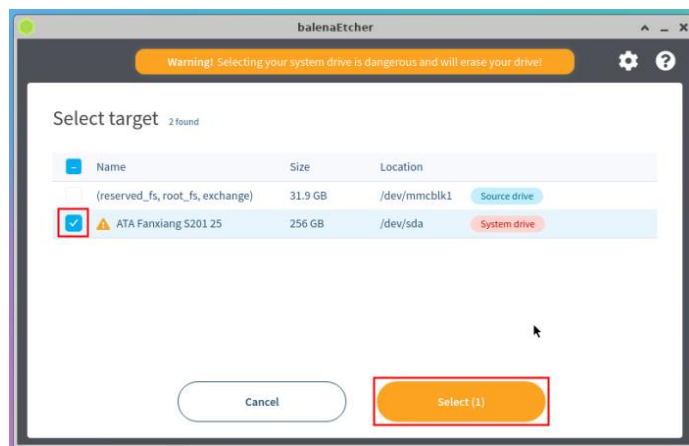
9) 然后点击 **Select target**。



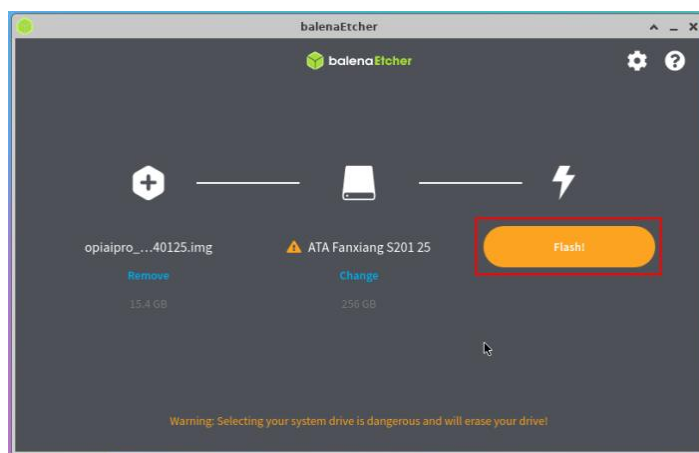
10) 然后点击 **Show 1 hidden**。



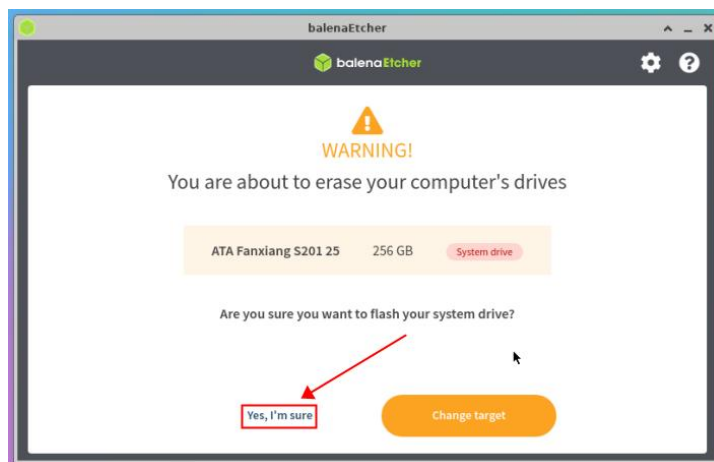
11) 然后选择 SSD 对应的选项，再点击 **Select** 即可。



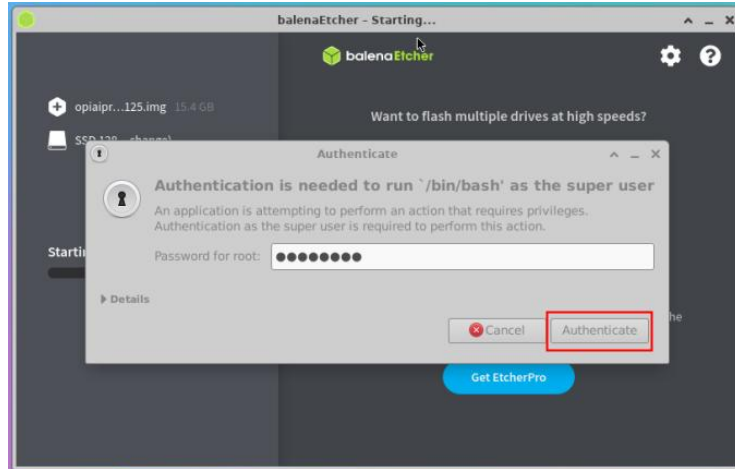
12) 然后点击 **Flash!** 开始烧录。



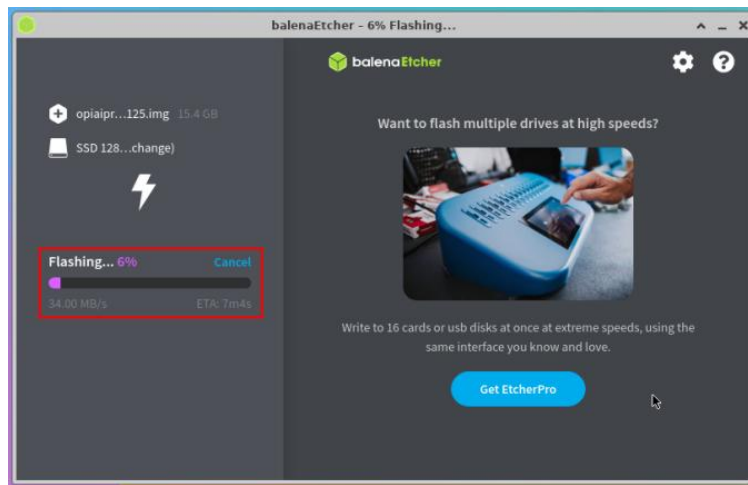
13) 然后选择 **Yes, I'm sure**。



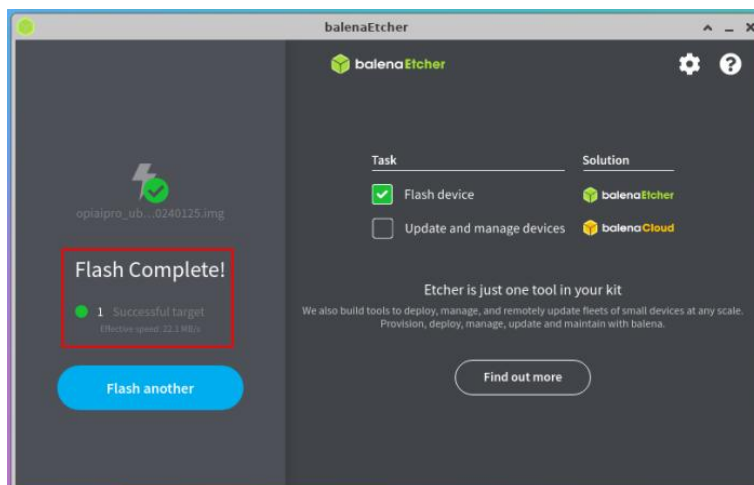
14) 然后输入 Linux 系统的密码: **Mind@123**。



15) 然后就会真正开始烧录 Linux 镜像到 SSD 中了。



16) Linux 镜像烧录完后的显示如下所示：



17) 烧录完成后，还需要将 SATA 版本的 dt.img 烧录到 SATA SSD 中，因为提供的镜像默认打开的都是 PCIe 的配置。具体命令如下所示：

```
sudo dd if=/opt/opi_test/dt_img/dt_drm_sata.img of=/dev/sda count=4096 seek=114688 bs=512
```

注意，上面的命令中，“of=”参数后面的 `/dev/sda` 为 SSD 对应的设备节点名。请根据实际情况进行修改。

18) 此时就可以关闭掉 Linux 系统，然后拔出 TF 卡，并断开 Type-C 电源。再将两个拨码开关拨到 SSD 启动对应的位置，然后重新插入 Type-C 电源就可以启动 SSD 中的 Linux 系统了。

注意，启动系统前请确保拨码开关拨到了 SSD 启动的位置了。拨码开关的使用说明请参考[控制启动设备的两个拨码开关的使用说明](#)一小节的说明。

## 2.6. 板载 LED 灯说明

开发板上有两个绿色的 LED 灯，作用如下所示：

1) 靠近关机按键的绿灯：此绿灯为电源指示灯，由硬件控制其亮灭，软件无法控制。只要开发板接入了 Type-C 电源并上电了，此绿灯就会点亮。

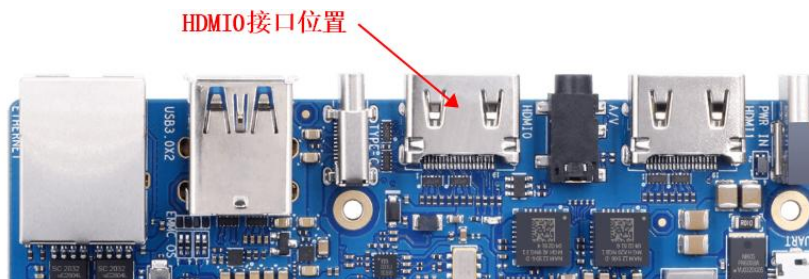


2) MIPI LCD 和 CAMERA0 之间的绿灯：此绿灯由 `GPIO4_19` 控制其亮灭，可以作为 SATA 硬盘的指示灯或者其他需要的用途。目前发布的 Linux 系统默认在 DTS 中将其点亮。当看到此灯点亮后，至少可以说明 Linux 内核已经启动了。

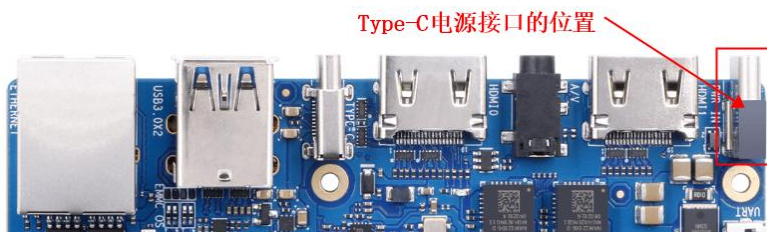


## 2.7. 启动开发板的步骤

- 1) 将烧录好镜像的 TF 卡或者 eMMC 模块或者 SSD 插入开发板对应的插槽中。
- 2) 开发板有两个 HDMI 接口（目前只有 HDMI0 支持显示 Linux 系统的桌面，HDMI1 显示 Linux 系统桌面的功能还需等软件更新），如果想显示 Linux 系统的桌面，可以将开发板的 HDMI0 接口连接到 HDMI 显示器。



- 3) 开发板有 USB 接口，可以接上 USB 鼠标和键盘，来控制开发板。
- 4) 开发板有千兆以太网口，可以插入网线用来上网。
- 5) 然后需要连接一个 20V PD-65W 的 Type C 接口的电源，电源接口的位置如下图所示：

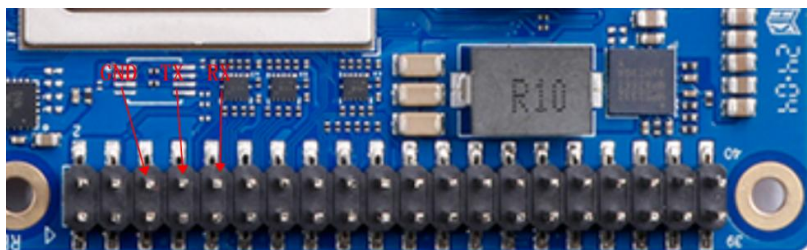


- 6) 然后打开电源适配器的开关，如果一切正常，等待一段时间后，HDMI 显示器就能看到 Linux 系统的登录界面了。
- 7) 如果想通过调试串口查看系统的输出信息，请使用串口线将开发板连接到电脑，串口的连接方法请参看[调试串口的使用方法](#)一节。

## 2.8. 调试串口的使用方法

开发板默认使用 uart0 做为调试串口。需要注意的是，uart0 的 tx 和 rx 引脚同时接到了两个地方，所以有两种使用调试串口的方法：

1) uart0 的 tx 和 rx 引脚接到了 40 pin 扩展接口中的 8 号和 10 号引脚，此种方式需要准备一个 3.3v 的 USB 转 TTL 模块和相应的杜邦线，然后才能正常使用开发板的调试串口功能。



2) uart0 的 tx 和 rx 引脚又接到了开发板的 CH343P 芯片上，再通过 CH343P 芯片引出到 Micro USB 接口上。此种方式只需要一根 Micro USB 接口的数据线将开发板连接到电脑的 USB 接口就可以开始使用开发板的调试串口功能了，无需购买 USB 转 TTL 模块。这种方法是推荐的方法。



另外请注意，上面的两种方法只能二选一，请不要同时使用。

### 2.8.1. 通过 Micro USB 接口来使用调试串口的连接说明

1) 首先需要准备一根 Micro USB 接口的数据线



2) 然后将 Micro USB 接口一端插入开发板的 Micro USB 接口中。



3) 再将数据线的另一端插入电脑的 USB 接口中即可。

## 2.8.2. 通过 40 pin 接口中的 uart0 来使用调试串口的连接说明

1) 首先需要准备一个 3.3v 的 USB 转 TTL 模块，然后将 USB 转 TTL 模块的 USB 接口一端插入到电脑的 USB 接口中。

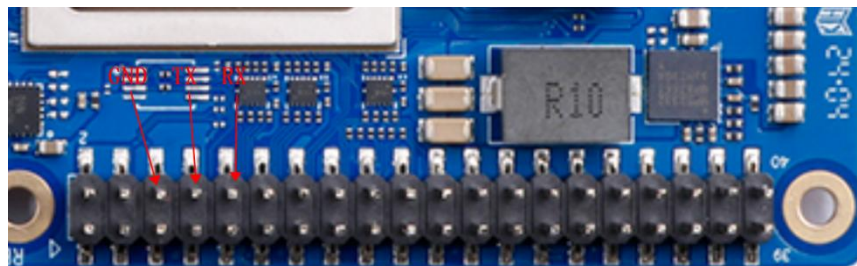
香橙派

USB转TTL模块



USB转TTL模块的3.3V不需要接  
USB转TTL模块的TXD接开发板调试串口的RXD  
USB转TTL模块的RXD接开发板调试串口的TXD  
USB转TTL模块的GND接开发板调试串口的GND  
USB转TTL模块的5V不需要接

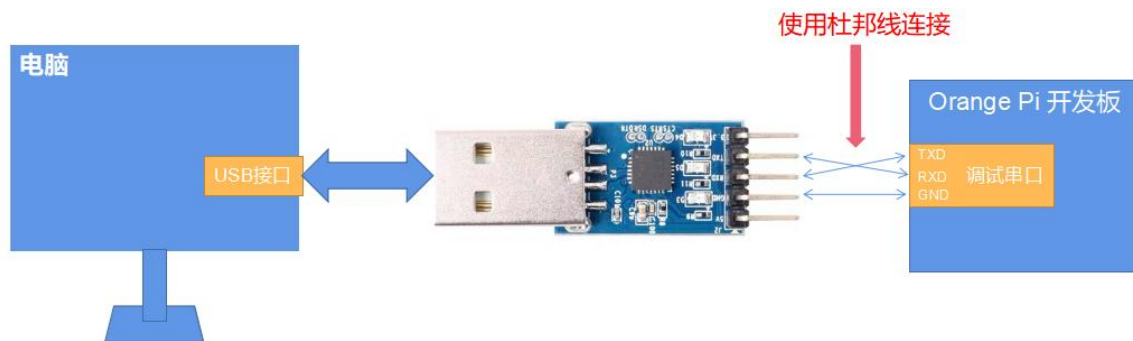
2) 开发板的调试串口 GND、TX 和 RX 引脚的对应关系如下图所示：



3) USB 转 TTL 模块 GND、TX 和 RX 引脚需要通过杜邦线连接到开发板的调试串口上。

- USB 转 TTL 模块的 GND 接到开发板的 GND 上。
- USB 转 TTL 模块的 **RX** 接到开发板的 **TX** 上。
- USB 转 TTL 模块的 **TX** 接到开发板的 **RX** 上。

4) USB 转 TTL 模块连接电脑和开发板的示意图如下所示：



USB转TTL模块连接电脑和 Orange Pi 开发板的示意图

串口的 TX 和 RX 是需要交叉连接的，如果不想仔细区分 TX 和 RX 的顺序，可以把串口的 TX 和 RX 先随便接上，如果测试串口没有输出再交换下 TX 和 RX 的顺序，这样就总有一种顺序是对的。

### 2.8.3. Ubuntu 平台调试串口的使用方法

Linux 下可以使用的串口调试软件有很多，如 **putty**、**minicom** 等，下面演示下 **putty** 的使用方法。

1) 首先请按照[通过 40 pin 接口中的 uart0 来使用调试串口的连接说明](#)或[通过 Micro USB 接口来使用调试串口的连接说明](#)一小节的说明（两种方法请根据自己的情况二选一）将开发板和电脑连接起来，如果串口模块识别正常的话，在 Ubuntu PC 的 **/dev** 下就可以看到对应的设备名，请记住这个设备名，后面设置串口软件时会用到。

a. 如果使用 40 pin 接口中的 uart0 显示的设备名一般为 **/dev/ttyUSB0**。

```
test@test:~$ ls /dev/ttyUSB*
/dev/ttyUSB0
```

b. 如果使用 Micro USB 接口显示的设备名一般为 **/dev/ttyACM0**。

```
test@test:~$ ls /dev/ttyACM*
/dev/ttyACM0
```

2) 然后使用下面的命令在 Ubuntu PC 上安装下 **putty**。

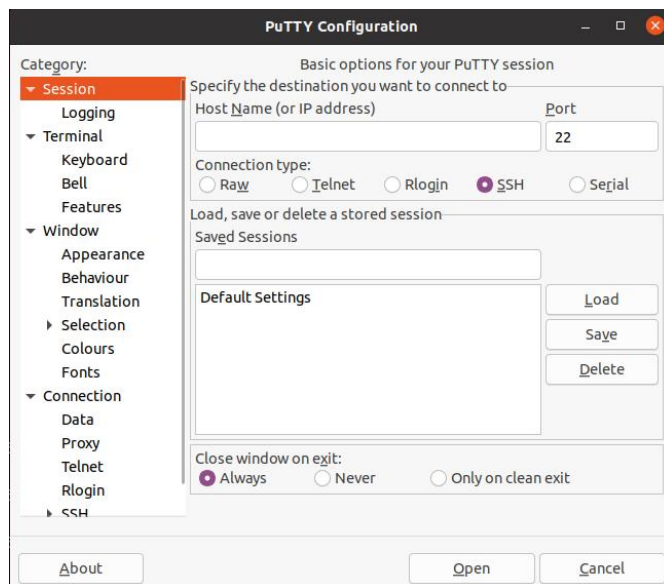
```
test@test:~$ sudo apt-get update
test@test:~$ sudo apt-get install -y putty
```

3) 然后运行 **putty**，记得加 **sudo** 权限。

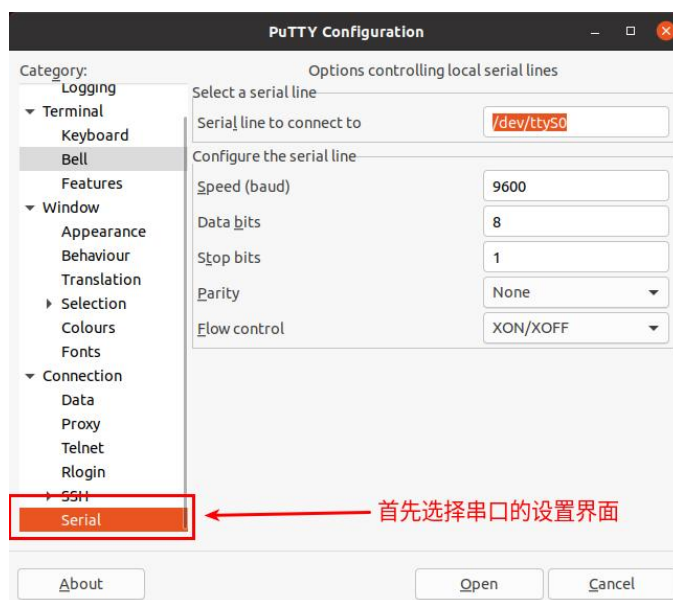
```
test@test:~$ sudo putty
```



4) 执行 `putty` 命令后会弹出下面的界面。

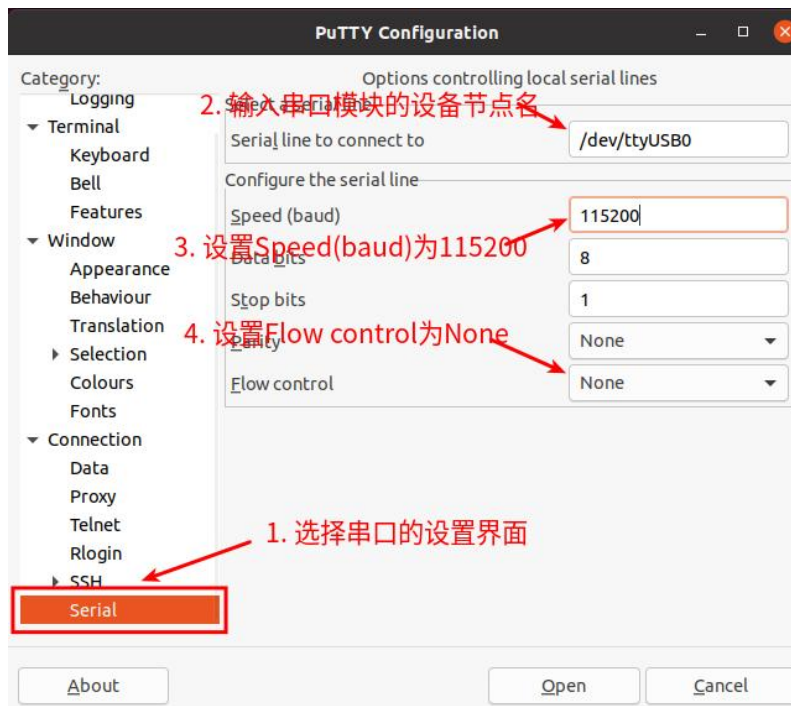


5) 首先选择串口的设置界面。



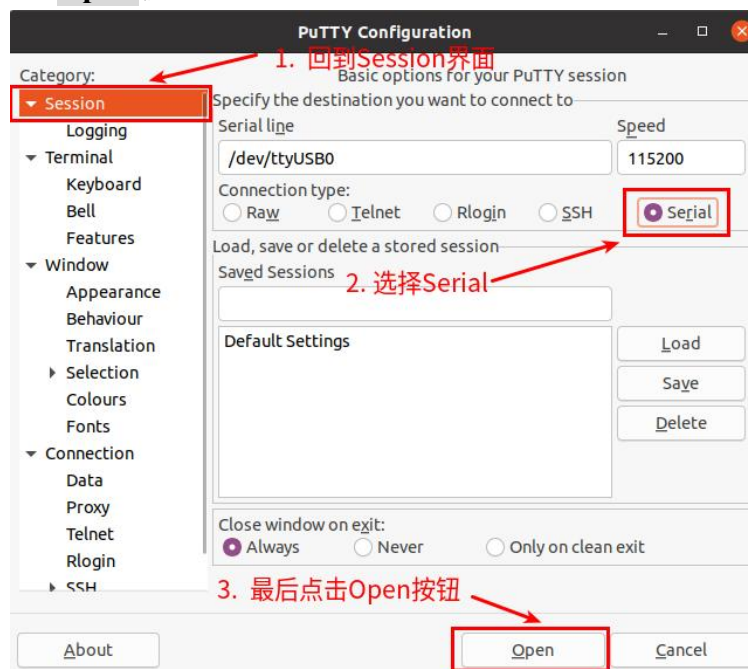
6) 然后设置串口的参数。

- 设置 **Serial line to connect to** 为 `/dev/ttyUSB0` 或者 `/dev/ttyACM0`（请根据实际情况进行修改）。
- 设置 **Speed(baud)** 为 **115200**（串口的波特率）。
- 设置 **Flow control** 为 **None**。



7) 在串口的设置界面设置完后，再回到 Session 界面。

- 首先选择 **Connection type** 为 **Serial**。
- 然后点击 **Open** 按钮连接串口。



8) 然后启动开发板，就能从打开的串口终端中看到系统输出的 Log 信息了。

```

./dev/ttyACMO - PoTTY
2.685907 rsd(hd____ptrval____), sizeof=1000
2.674383 rcs(hd____ptrval____), sizeof=1000
2.674382 rcs(hd____ptrval____), sizeof=1000
2.675602 Run /sbin/init as init process
2.685123 systemd[1]: Seton time before build-time, advancing clock.
2.623661 ip.table: module verification failed: signature and/or required key missing - tainting kernel
2.412474 random: crypto: uninitialized urandom read (4 bytes read)
3.648863 random: cryptdev: uninitialized urandom read (16 bytes read)
3.672000 random: cryptdev: uninitialized urandom read (16 bytes read)
3.745103 fusectl init (v0.7 version 7.33)
3.811202 device-mapper: udevinit: version 1.0.3
3.811202 device-mapper: ioctl: 4.45.0-ioctl (2020-10-01) initialized: dm-devel@redhat.com
3.872602 nme_core: exports duplicate symbol __SCL__to_func_name_sq (owned by kernel)
3.872602 nme_core: exports duplicate symbol __SCL__to_func_name_sq (owned by kernel)
3.838843 random: crng input done
random: SE warning(s) missed due to reinitializing
4.084843 stt: Version 20850205, Fixed bufsize 32768, s/a segs 256
4.118282 usbcore: registered new interface driver usbfs_lsm_generic
4.118282 usbserial: USB Serial support registered for generic
4.145793 usbcore: registered new interface driver pl2303
4.153384 usbserial: USB Serial support registered for pl2303
4.203484 RFS Registered named UNIX socket transport module.
4.212384 RFS Registered udp transport module.
4.220103 RFS Registered tcp transport module.
4.224403 RFS Registered tcp NFSv4.1 backchannel transport module.
4.338863 tun: Universal UNW/FB device driver 1.6
4.339163 NET Registered protocol family 16
4.372807 bridge: filtering via xrp/ip/tPtables is no longer available by default. Update your scripts to load br_netfilter if you need this.
4.401424 bridge: firewalling registered
4.434703 cryptd: max_cpu_mems set to 1000
4.763293 PPP generic driver version 2.4.7
4.763293 PPP HPPC Compression module registered
4.830405 PPP deflate Compression module registered
4.844683 fns BZ2 Compression module registered
4.851403 systemd-journal[183]: Received client request to flush runtime journal.
4.851403 fdinfo: fdinfo[183] was not properly unmounted. Some data may be corrupt. Please run fsck.
4.920206 usbcore: registered new interface driver usb-storage
4.920206 usbcore: registered new interface driver cdc-acm
5.013344 usbcore: registered new interface driver option
5.015577 usbserial: USB Serial support registered for GSM modem (1-port)
5.047100 usbcore: registered new interface driver rcw_sdh
5.112120 mc: Linux media interface: v0.10
5.152968 videodev: Linux video capture interface: v2.00
5.232714 usbcore: registered new interface driver uvcvideo
5.239474 USB Video Class driver (1.1.1)
5.262318 nfpa: driver 2.4.32 [Xig/R/O NMULE].
5.262318 nfpa: the link count 000
NOTICE: Init IPID4, support=03979416
NOTICE: mfm1 P4, modification=0
NOTICE: [RscCbConnHandler][1711] moduleBase ci260000
NOTICE: RscCbConnHandler[181] UE
NOTICE: [hook1] MFM0
NOTICE: F11M1ModuleErrorInfo[1274], moduleID = 0id
NOTICE: [hook1] MFM0
NOTICE: SdsGsl2ChwFp_DeviceInfo[0], SubSpkName=0
NOTICE: MEMOSBLE
NOTICE: MemNotified05
NOTICE: [RscCbConnHandler][183] Handler end
NOTICE: base = 0xc120000
NOTICE: EPR_PSL = 0x242ae2
NOTICE: EPR_FSL = 0x0
NOTICE: EPR_CTRL = 0x015
NOTICE: EPR_CTRLA = 0x0
NOTICE: EPR_STATL = 0xf30f0e
NOTICE: EPR_STATH = 0x0
NOTICE: EPR_ADRPL = 0x1000000
NOTICE: EPR_ADRH = 0x00000004
NOTICE: EPR_MISOL = 0x0
NOTICE: EPR_MISCH = 0x7f
NOTICE: EPR_MISCL = 0xa518005
NOTICE: EPR_MISHL = 0xa00122
NOTICE: #13 test suite
cpu 0 entering scheduler
XXXXXXXXXXXX start success!!!!!!!!!!!!!!
libata 2.04 LTS (aragorn@kernel.org)

```

9) 当看到登录界面时，就可以使用下面的账号和密码来登录 Linux 系统了。

|            |          |
|------------|----------|
| 账号         | 密码       |
| root       | Mind@123 |
| HwHiAiUser | Mind@123 |

#### 2.8.4. Windows 平台调试串口的使用方法

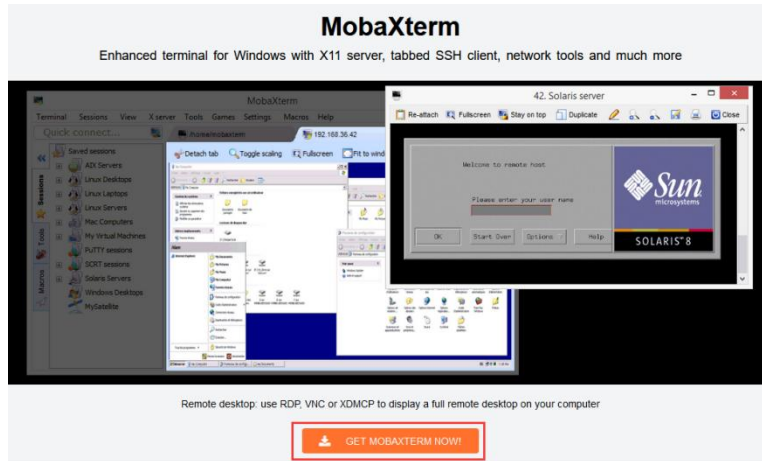
Windows 下可以使用的串口调试软件有很多，如 SecureCRT、MobaXterm 等，下面演示 MobaXterm 的使用方法，这款软件有免费版本，无需购买序列号即可使用。

1) 首先下载 MobaXterm。

a. 下载 MobaXterm 网址如下:

<https://mobaxterm.mobatek.net/>

b. 进入 MobaXterm 下载网页后点击 **GET XOBATERM NOW!**。



c. 然后选择下载 Home 版本。

| Home Edition                                                                                                                                                                                                                                                                                                                                                                                | Professional Edition                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Free</b><br>Full X server and SSH support<br>Remote desktop (RDP, VNC, Xdmcp)<br>Remote terminal (SSH, telnet, rlogin, Mosh)<br>X11-Forwarding<br>Automatic SFTP browser<br>Master password protection<br>Plugins support<br>Portable and installer versions<br>Full documentation<br>Max. 12 sessions<br>Max. 2 SSH tunnels<br>Max. 4 macros<br>Max. 360 seconds for Tftp, Nfs and Cron | <b>\$69 / 49€ per user*</b><br>* Excluding tax. Volume discounts <a href="#">available</a><br><b>Every feature from Home Edition +</b><br>Customize your startup message and logo<br>Modify your profile script<br>Remove unwanted games, screensaver or tools<br>Unlimited number of sessions<br>Unlimited number of tunnels and macros<br>Unlimited run time for network daemons<br>Enhanced security settings<br>12 months updates included<br>Deployment inside company<br>Lifetime right to use |
| <a href="#">Download now</a>                                                                                                                                                                                                                                                                                                                                                                | <a href="#">Subscribe online / Get a quote</a>                                                                                                                                                                                                                                                                                                                                                                                                                                                       |

d. 然后选择 Portable 便携式版本，下载完后无需安装，直接打开就可以使用。



2) 下载完后使用解压缩软件解压下载的压缩包，即可得到 MobaXterm 的可执软件，然后双击打开。

|                             |                 |           |           |
|-----------------------------|-----------------|-----------|-----------|
| MobaXterm_Personal_23.6.exe | 2023/12/21 6:15 | 应用程序      | 16,556 KB |
| CygUtils.plugin             | 2023/12/21 5:08 | PLUGIN 文件 | 17,748 KB |
| CygUtils64.plugin           | 2023/12/21 5:08 | PLUGIN 文件 | 11,723 KB |

3) 打开软件后，设置串口连接的步骤如下：

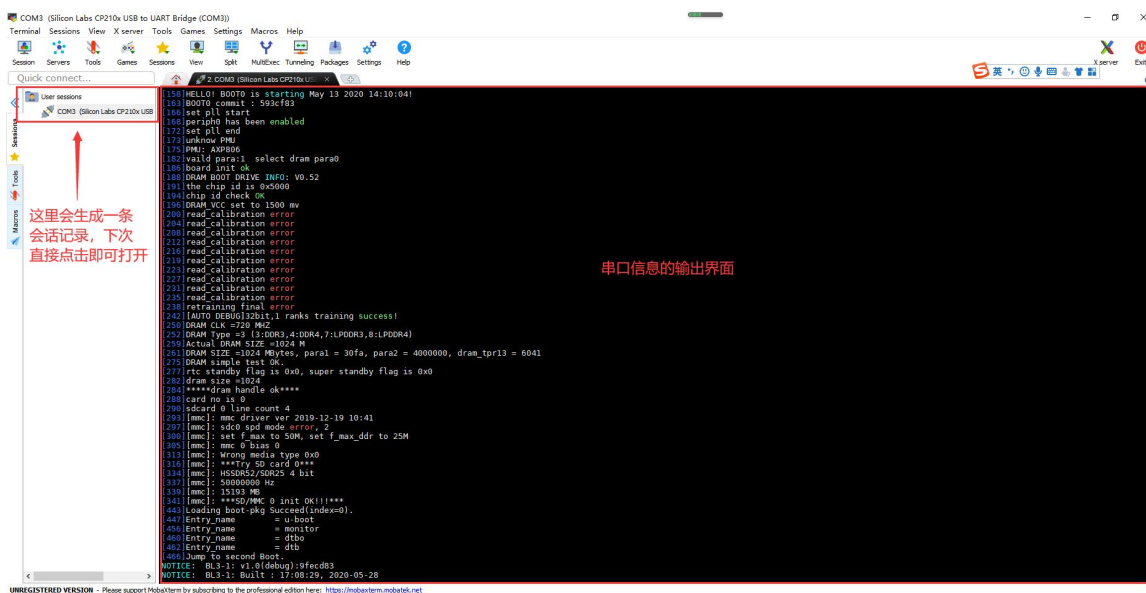
a. 打开会话的设置界面。



- b. 选择串口类型。
- c. 选择串口的端口号（根据实际情况选择对应的端口号），如果看不到端口号，请使用 [360 驱动大师](#) 扫描安装 USB 转 TTL 串口芯片的驱动。
- d. 选择串口的波特率为 115200。
- e. 最后点击 “OK” 按钮完成设置。



4) 点击 “OK” 按钮后会进入下面的界面，此时启动开发板就能看到串口的输出信息了。



5) 当看到登录界面时，就可以使用下面的账号和密码来登录 Linux 系统了。



| 账号         | 密码       |
|------------|----------|
| root       | Mind@123 |
| HwHiAiUser | Mind@123 |

## 2.9. Linux 系统功能适配情况

| 功能                       | 是否能测试 | Linux 内核驱动 |
|--------------------------|-------|------------|
| HDMI0 1080p 显示           | OK    | OK         |
| HDMI0 4K 显示              | NO    | NO         |
| HDMI0 音频                 | OK    | NO         |
| HDMI1 显示                 | OK    | NO         |
| HDMI1 音频                 | OK    | NO         |
| 耳机播放                     | OK    | NO         |
| 耳机 MIC 录音                | OK    | NO         |
| Type-C USB3.0 (无 USB2.0) | OK    | OK         |
| USB3.0 Host x 2          | OK    | OK         |
| 千兆网口                     | OK    | OK         |
| 千兆网口灯                    | OK    | OK         |
| WIFI                     | OK    | OK         |
| 蓝牙                       | OK    | OK         |
| Micro USB 调试串口           | OK    | OK         |
| 复位按键                     | OK    | OK         |
| 关机按键 (无开机功能)             | OK    | OK         |
| 烧录按键                     | OK    | OK         |
| MIPI 摄像头 0               | OK    | NO         |
| MIPI 摄像头 1               | OK    | NO         |
| MIPI LCD 显示              | OK    | NO         |
| 电源指示灯                    | OK    | OK         |
| 软件可控的 LED 灯              | OK    | OK         |
| 风扇接口                     | OK    | OK         |
| 电池接口                     | OK    | OK         |
| TF 卡启动                   | OK    | OK         |
| TF 卡启动识别 eMMC            | OK    | OK         |
| TF 卡启动识别 NVMe SSD        | OK    | OK         |



|                   |    |    |
|-------------------|----|----|
| TF 卡启动识别 SATA SSD | OK | OK |
| eMMC 启动           | OK | OK |
| SATA SSD 启动       | OK | OK |
| NVMe SSD 启动       | OK | OK |
| 2 个拨码开关           | OK | OK |
| 40 pin-GPIO       | OK | OK |
| 40 pin-UART       | OK | OK |
| 40 pin-SPI        | OK | OK |
| 40 pin-I2C        | OK | OK |
| 40 pin-PWM        | OK | NO |

## 2. 10. HDMI 接口的使用说明

### 2. 10. 1. HDMI 显示 Linux 桌面的说明

开发板有两个 HDMI2.0 接口，目前只有 HDMI0 支持显示 Linux 系统的桌面，HDMI0 显示 Linux 系统桌面的详细说明请查看[登录 Linux 系统桌面的方法](#)一小节的说明。

### 2. 10. 2. 使用 HDMI 接口显示图片和播放音频的方法

当 Linux 系统的桌面系统关闭时，HDMI0 和 HDMI1 还可以用于 NVR 二次开发场景输出图片。

测试 HDMI0 输出一张图片的方法如下所示：

1) 首先连接 HDMI0 接口到 HDMI 显示器。

2) 然后切换到 root 用户，并进入 HDMI0 测试程序所在路径。

```
(base) HwHiAiUser@orangepiaipro:~$ sudo -i
(base) root@orangepiaipro:~# cd /opt/opi_test/hdmi0_pic
(base) root@orangepiaipro:/opt/opi_test/hdmi0_pic# ls
sample_hdmi test.sh update_dt.sh ut_1920x1080_nv12.yuv
```

3) 然后运行 **update\_dt.sh** 脚本更新下 dt.img（关闭 drm 的配置，打开 vdp 的配置）。**update\_dt.sh** 脚本运行完后会自动重启 Linux 系统。

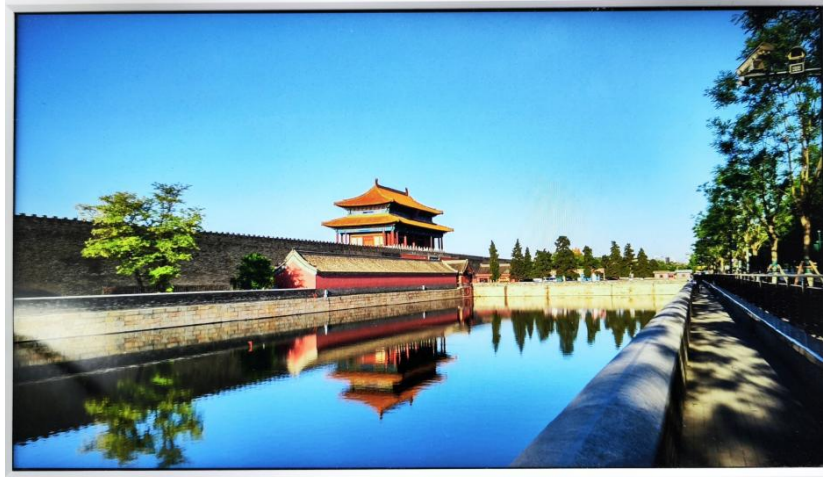
```
(base) root@orangepiaipro:/opt/opi_test/hdmi0_pic# ./update_dt.sh
```



4) 重启后再次进入 HDMI0 测试程序所在的路径，然后运行 **test.sh** 脚本就会播放一张图片到 HDMI 显示器（默认显示 10 秒），并且同时会播放一段音频到 HDMI 显示器，如果 HDMI 显示器支持播放音频的话，还能听到声音。

```
(base) root@orangepiaipro:/opt/opi_test/hdmi0_pic# ./test.sh
```

5) HDMI 显示的图片如下所示：



测试 HDMI1 输出一张图片的方法和 HDMI0 是一样的，只是测试程序的路径为：

```
/opt/opi_test/hdmi1_pic
```

测试完 HDMI 播放图片后，再切换回 HDMI0 显示 Linux 系统桌面的方法如下所示：

1) 首先切换到 root 用户，并进入 **/opt/opi\_test/hdmi\_desktop** 目录。

```
(base) HwHiAiUser@orangepiaipro:~$ sudo -i
(base) root@orangepiaipro:~# cd /opt/opi_test/hdmi_desktop
(base) root@orangepiaipro:/opt/opi_test/hdmi_desktop# ls
update_dt.sh
```

2) 然后运行 **update\_dt.sh** 脚本更新下 dt.img（关闭 vdp 的配置，打开 drm 的配置）。**update\_dt.sh** 脚本运行完后会自动重启 Linux 系统。

```
(base) root@orangepiaipro:/opt/opi_test/hdmi_desktop# ./update_dt.sh
```

3) 重启后就能看到 HDMI 显示器会显示 Linux 系统的桌面了。

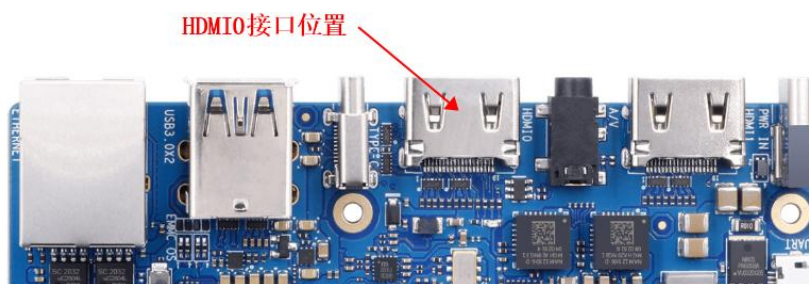
## 2.11. Linux 系统默认登录账号和密码

| 账号         | 密码       |
|------------|----------|
| root       | Mind@123 |
| HwHiAiUser | Mind@123 |

当输入密码提示错误，或者 ssh 连接有问题，请注意，只要使用的是 Orange Pi 提供的 Linux 镜像，**就请不要怀疑上面的密码不对**，而是要找其他的原因。

## 2.12. 登录 Linux 系统桌面的方法

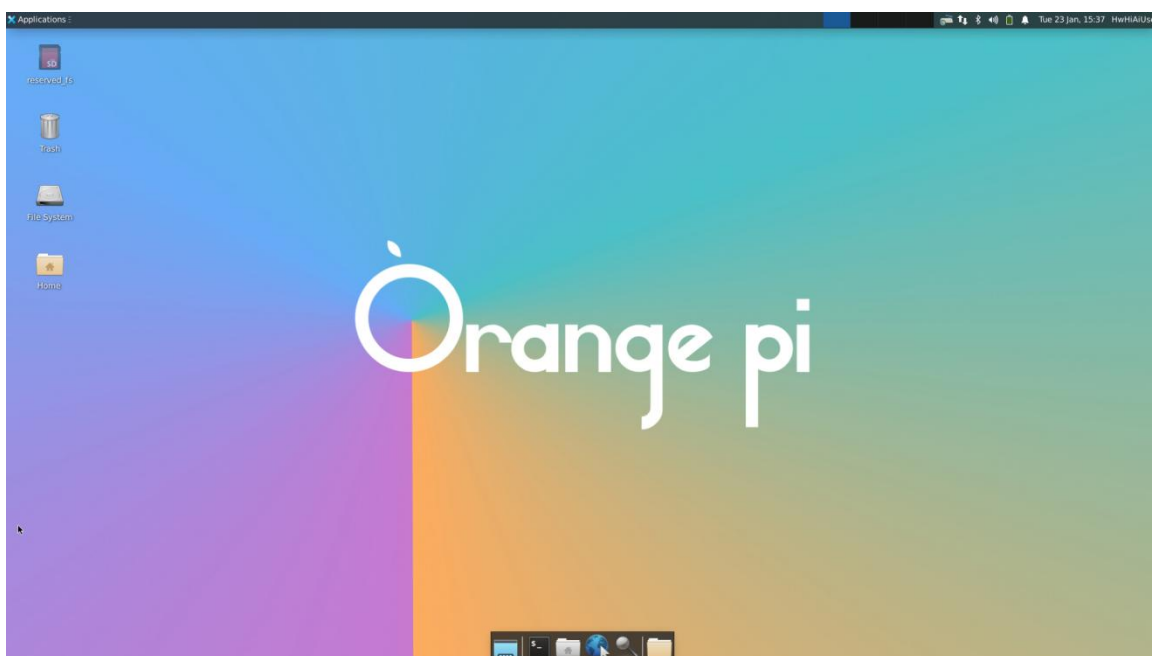
开发板有两个 HDMI 接口，目前只有 HDMI0 支持显示 Linux 系统的桌面，HDMI1 还需等软件更新。如果想显示 Linux 系统的桌面，请将开发板的 HDMI0 接口连接到 HDMI 显示器。



开发板上电开机后，需要等待一段时间，HDMI 显示器才会显示 Linux 系统的登录界面，登录界面如下图所示：



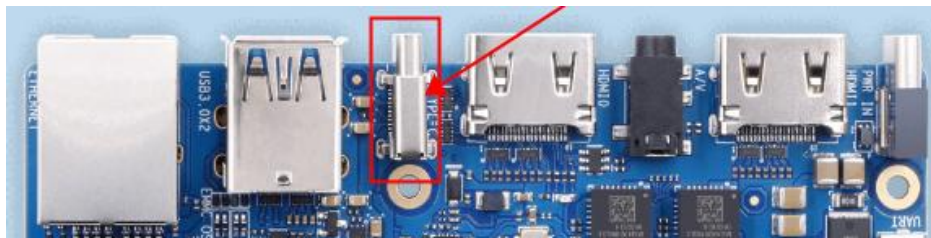
Linux 桌面系统的默认登录用户为 **HwHiAiUser**，登录密码为 **Mind@123**。目前没有打开 root 用户登录的通道。成功登录后显示的 Linux 系统桌面如下图所示：



## 2.13. USB 接口测试

### 2.13.1. Type-C USB 接口使用注意事项

开发板有一个 Type-C USB 接口，其所在位置如下图所示：



需要注意的是，Type-C USB 接口只有 USB3.0 的功能，没有 USB2.0 和 USB1.1 的功能（Soc 没有多余的 D+和 D-信号线可以用），所以无法接 USB 鼠标和键盘以及 USB2.0 的存储设备。测试此接口时，可以使用下图所示的 Type-C 转 USB3.0 转接线，来连接一个 USB3.0 的存储设备。



### 2.13.2. 连接 USB 鼠标或键盘测试

开发板有两个 USB3.0 HOST 接口可以接鼠标和键盘，其所在位置如下所示。如果 USB 接口不够用的话，还可以通过 USB Hub 来扩展 USB 接口的数量。



将 USB 接口的鼠标和键盘插入开发板的 USB 3.0 接口中，然后连接开发板到的 HDMI0 接口到 HDMI 显示器，等 HDMI 显示器显示 Linux 系统的桌面后就可以使用鼠标键盘来操作 Linux 系统了。



### 2.13.3. USB 摄像头测试

1) 首先将 USB 摄像头插入到开发板的 USB3.0 HOST 接口中。



2) 然后通过 **v4l2-ctl** 命令就可以看到 USB 摄像头的设备节点信息为 **/dev/video0**

```
(base) HwHiAiUser@orangepiaipro:~$ sudo apt-get update
(base) HwHiAiUser@orangepiaipro:~$ sudo apt-get install -y v4l-utils
(base) HwHiAiUser@orangepiaipro:~$ sudo v4l2-ctl --list-devices
Q8 HD Webcam: Q8 HD Webcam (usb-xhci-hcd.3.auto-1):
    /dev/video0
    /dev/video1  #这个是用来采集 metadata 的，先忽略。
    /dev/media0
```

注意 **v4l2** 中的 **l** 是小写字母 **l**，不是数字 **1**。

另外 **video** 的序号不一定是 **video0**，请以实际看到的为准。

3) 使用 **fswebcam** 测试 USB 摄像头

a. 安装 **fswebcam**

```
(base) HwHiAiUser@orangepiaipro:~$ sudo apt-get update
(base) HwHiAiUser@orangepiaipro:~$ sudo apt-get install -y fswebcam
```

b. 安装完 **fswebcam** 后可以使用下面的命令来拍照

- a) **-d** 选项用于指定 USB 摄像头的设备节点
- b) **--no-banner** 用于去除照片的水印
- c) **-r** 选项用于指定照片的分辨率
- d) **-S** 选项用于设置于跳过前面的帧数
- e) **./image.jpg** 用于设置生成的照片的名字和路径

```
(base) HwHiAiUser@orangepiaipro:~$ sudo fswebcam -d /dev/video0 --no-banner -r 1280x720 -S 5 ./image.jpg
```

c. 然后就可以通过 HDMI 显示器在 Linux 桌面直接打开查看拍摄的图片。

4) 使用内置的 USB Camera 样例代码测试 USB 摄像头。

a. 首先进入 USB Camera 样例代码的路径。

```
(base) HwHiAiUser@orangepiaipro:~$ sudo -i
(base) root@orangepiaipro:~# cd /opt/opi_test/USBCamera
(base) root@orangepiaipro:/opt/opi_test/USBCamera# ls
main main.cpp readme.md
```

b. 然后运行下面的命令就可以使用 USB 摄像头拍一张照片：

```
(base) root@orangepiaipro:/opt/opi_test/USBCamera# ./main /dev/video0
```

c. 运行成功后，在 USB Camera 样例目录下会生成一个 yuv422 格式、1280\*720 分辨率的 **out.yuv** 文件。

```
(base) root@orangepiaipro:/opt/opi_test/USBCamera# ls
main main.cpp out.yuv readme.md
```

d. 然后在 Linux 桌面中使用下面的命令可以查看 **out.yuv** 文件的内容。

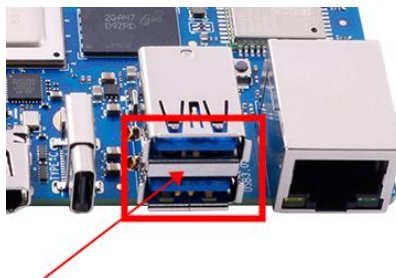
```
(base) root@orangepiaipro:/opt/opi_test/USBCamera# ffplay -pix_fmt yuv422 -video_size 1280*720 out.yuv
```

## 2.13.4. USB 音频测试

1) 首先需要准备一个带录音功能的 USB 接口的耳机。



2) 然后将 USB 接口的耳机插入开发的 USB 接口中。





3) 然后使用 **arecord -l** 命令查看下录音设备的编号，如下面的输出所示，其中 **card 0** 中的 **0** 表示录音设备编号为 **0**。

```
(base) HwHiAiUser@orangepiaipro:~$ arecord -l
**** List of CAPTURE Hardware Devices ****
card 0: Audio [AB13X USB Audio], device 0: USB Audio [USB Audio]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
```

4) 然后进入 USB 音频测试代码的路径中。

```
(base) HwHiAiUser@orangepiaipro:~$ sudo -i
(base) root@orangepiaipro:~# cd /opt/opi_test/USBAudio
(base) root@orangepiaipro:/opt/opi_test/USBAudio# ls
Readme.md main main.c
```

5) 然后使用下面的命令可以使用 USB 音频设备录制一段音频。其中 **0** 表示录音设备编号，需根据实际情况填写。

```
(base) root@orangepiaipro:/opt/opi_test/USBAudio# ./main plughw:0
```

6) 录制结束后，在终端界面输入 **over** 即可退出录制。

```
(base) root@orangepiaipro:/opt/opi_test/USBAudio# ./main plughw:0
Start record!
over      #输入 over 结束录制音频。
(base) root@orangepiaipro:/opt/opi_test/USBAudio#
```

7) 录音成功后，在 USBAudio 样例目录下会生成音频文件 **audio.pcm**。

```
(base) root@orangepiaipro:/opt/opi_test/USBAudio# ls *.pcm
audio.pcm
```

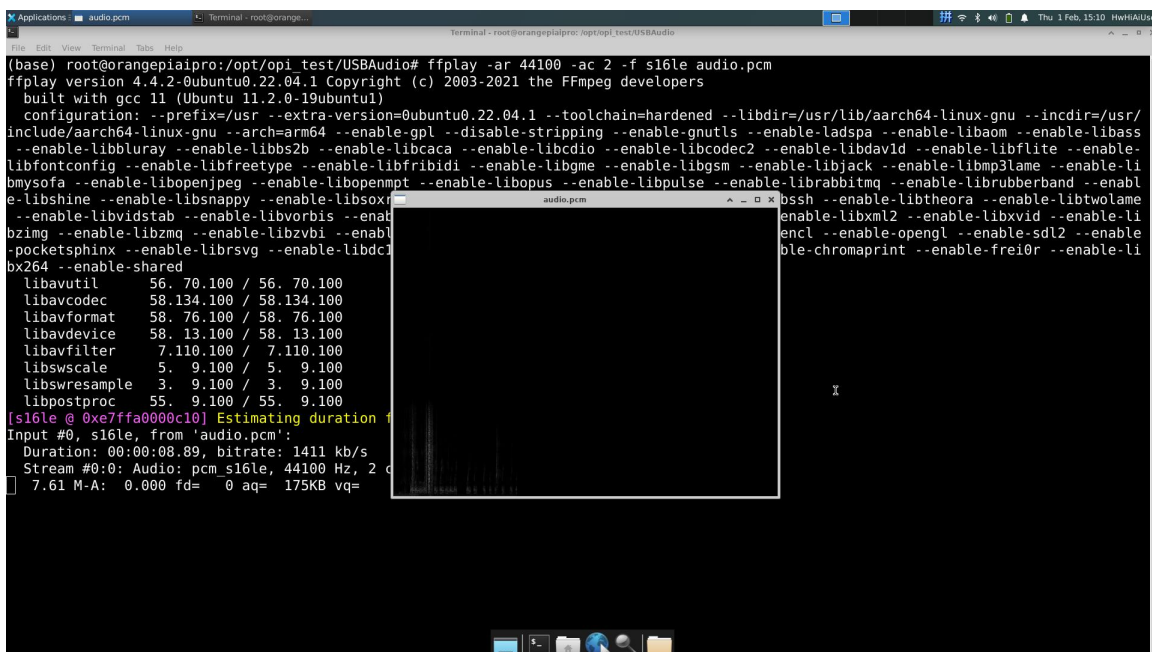
8) 然后确保使用 **aplay -l** 命令能看到 USB 的播音设备。

```
(base) root@orangepiaipro:/opt/opi_test/USBAudio# aplay -l
**** List of PLAYBACK Hardware Devices ****
card 0: Audio [AB13X USB Audio], device 0: USB Audio [USB Audio]
  Subdevices: 0/1
  Subdevice #0: subdevice #0
```



9) 然后在 Linux 系统桌面中，使用下面的命令可以将录制的音频播放到 USB 耳机。

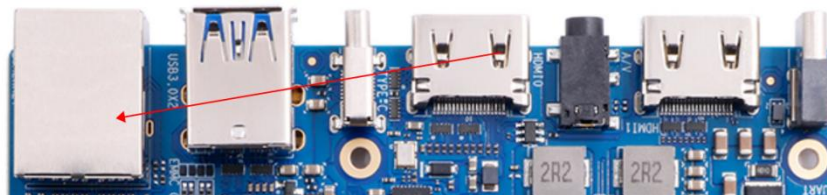
```
(base) root@orangepiaipro:/opt/opi_test/USBAudio# ffplay -ar 44100 -ac 2 -f s16le audio.pcm
```



## 2. 14. 网络连接测试

### 2. 14. 1. 以太网口测试

1) 首先将网线的一端插入开发板的以太网接口，网线的另一端接入交换机或者路由器，并确保网络是畅通的。



2) 系统启动后会通过 **DHCP** 自动给以太网口分配 IP 地址。

3) 在开发板的 Linux 系统中查看 IP 地址的命令如下所示：

```
(base) HwHiAiUser@orangepiaipro:~$ ip a s eth0
```

```
3: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP
group default qlen 1000
```



```
link/ether 2c:52:af:89:11:11 brd ff:ff:ff:ff:ff:ff
inet 192.168.2.100/24 brd 192.168.2.255 scope global dynamic noprefixroute eth0
    valid_lft 42077sec preferred_lft 42077sec
inet6 fe80::913d:474c:4834:a1a4/64 scope link noprefixroute
    valid_lft forever preferred_lft forever
```

4) 测试网络连通性的命令如下所示，如果能 ping 通百度说明开发板的网络连接正常，**ping** 命令可以通过 **Ctrl+C** 快捷键来中断运行。

```
(base) HwHiAiUser@orangepiaipro:~$ ping www.baidu.com -I eth0
PING www.a.shifen.com (183.2.172.185) from 192.168.2.100 eth0: 56(84) bytes of data.
64 bytes from 183.2.172.185 (183.2.172.185): icmp_seq=1 ttl=52 time=10.0 ms
64 bytes from 183.2.172.185 (183.2.172.185): icmp_seq=2 ttl=52 time=9.77 ms
64 bytes from 183.2.172.185 (183.2.172.185): icmp_seq=3 ttl=52 time=9.94 ms
64 bytes from 183.2.172.185 (183.2.172.185): icmp_seq=4 ttl=52 time=9.94 ms
^C
--- www.a.shifen.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 9.770/9.931/10.065/0.105 ms
```

## 2. 14. 2. WIFI 连接测试

请不要通过修改/etc/network/interfaces 配置文件的方式来连接 WIFI，通过这种方式连接 WIFI 网络使用会有问题。

### 2. 14. 2. 1. 通过 nmcli 命令连接 WIFI 的方法

1) 先登录 Linux 系统，有下面三种方式：

- a. 如果开发板连接了网线，可以通过 [ssh 远程登录 Linux 系统](#)。
- a. 如果开发板连接好了调试串口，可以使用串口终端登录 Linux 系统。
- b. 如果连接了开发板到 HDMI 显示器，可以通过 HDMI 显示的终端登录到 Linux 系统。

2) 然后使用 **nmcli dev wifi** 命令扫描周围的 WIFI 热点。

```
(base) HwHiAiUser@orangepiaipro:~$ nmcli dev wifi
```

3) 然后使用 **nmcli** 命令连接扫描到的 WIFI 热点，其中：



- a. **wifi\_name** 需要换成想连接的 WIFI 热点的名字。
- b. **wifi\_passwd** 需要换成想连接的 WIFI 热点的密码。

```
(base) HwHiAiUser@orangepiaipro:~$ sudo nmcli dev wifi connect wifi_name password wifi_passwd  
Device 'wlan0' successfully activated with 'cf937f88-ca1e-4411-bb50-61f402eef293'.
```

4) 通过 **ip addr show wlan0** 命令可以查看 wifi 的 IP 地址。

```
(base) HwHiAiUser@orangepiaipro:~$ ip a s wlan0  
4: wlan0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP  
group default qlen 1000  
    link/ether 54:f2:9f:7b:ba:36 brd ff:ff:ff:ff:ff:ff  
    inet 10.31.2.93/16 brd 10.31.255.255 scope global dynamic noprefixroute wlan0  
        valid_lft 43191sec preferred_lft 43191sec  
    inet6 fe80::5297:7036:a33c:bb93/64 scope link noprefixroute  
        valid_lft forever preferred_lft forever
```

5) 使用 **ping** 命令可以测试 wifi 网络的连通性，**ping** 命令可以通过 **Ctrl+C** 快捷键来中断运行。

```
(base) HwHiAiUser@orangepiaipro:~$ ping www.orangepi.org -I wlan0  
PING www.orangepi.org (123.57.147.237) from 10.31.2.93 wlan0: 56(84) bytes of data.  
64 bytes from 123.57.147.237 (123.57.147.237): icmp_seq=1 ttl=53 time=47.1 ms  
64 bytes from 123.57.147.237 (123.57.147.237): icmp_seq=2 ttl=53 time=44.3 ms  
64 bytes from 123.57.147.237 (123.57.147.237): icmp_seq=3 ttl=53 time=45.0 ms  
64 bytes from 123.57.147.237 (123.57.147.237): icmp_seq=4 ttl=53 time=71.0 ms  
^C  
--- www.orangepi.org ping statistics ---  
4 packets transmitted, 4 received, 0% packet loss, time 3002ms  
rtt min/avg/max/mdev = 44.377/51.902/71.082/11.119 ms
```

## 2.14.2.2. 通过 nmtui 图形化方式连接 WIFI 的方法

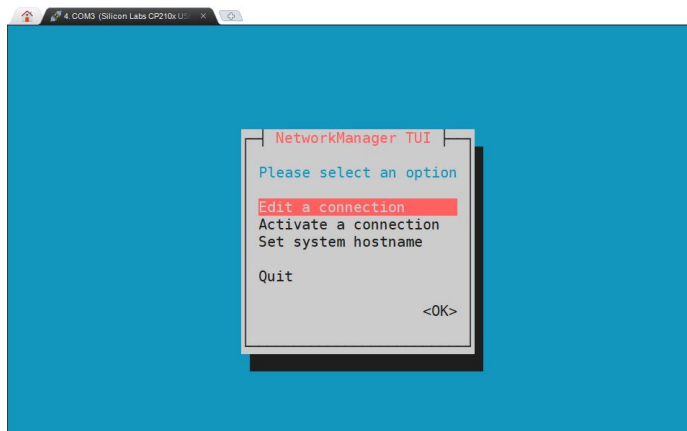
1) 先登录 Linux 系统，有下面三种方式：

- a. 如果开发板连接了网线，可以通过 [ssh 远程登录 Linux 系统](#)。
- b. 如果开发板连接好了调试串口，可以使用串口终端登录 Linux 系统。
- c. 如果连接了开发板到 HDMI 显示器，可以通过 HDMI 显示的终端登录到 Linux 系统。

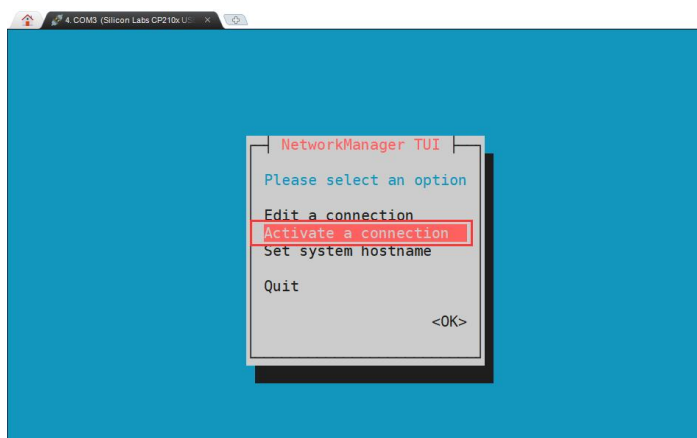
2) 然后在命令行中输入 `nmtui` 命令打开 wifi 连接的界面。

```
(base) HwHiAiUser@orangepiaipro:~$ sudo nmtui
```

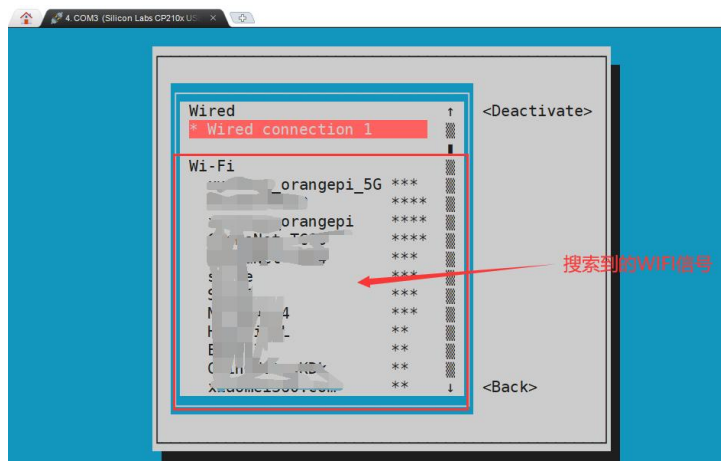
3) 输入 `nmtui` 命令打开的界面如下所示：



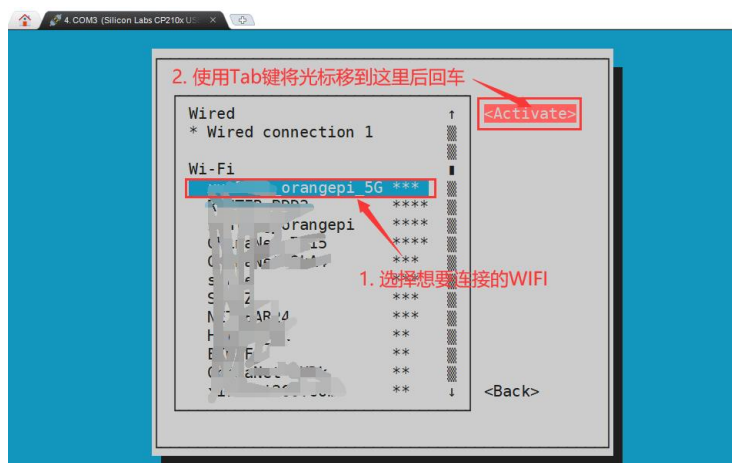
4) 选择 **Activate a connect** 后回车。



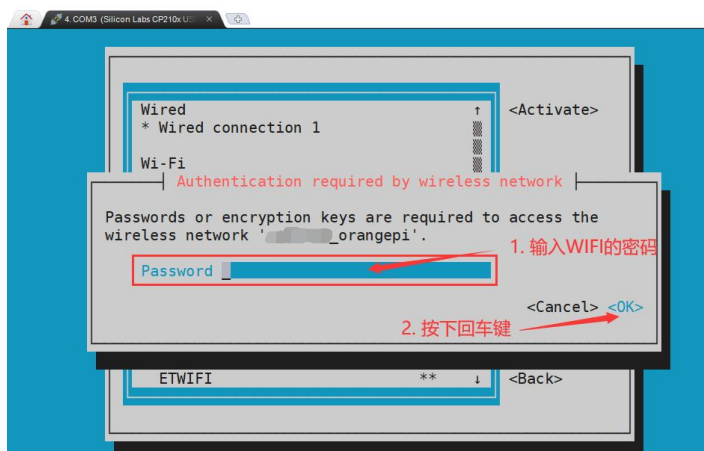
5) 然后就能看到所有搜索到的 WIFI 热点。



6) 选择想要连接的 WIFI 热点后再使用 Tab 键将光标定位到 **Activate** 后回车。

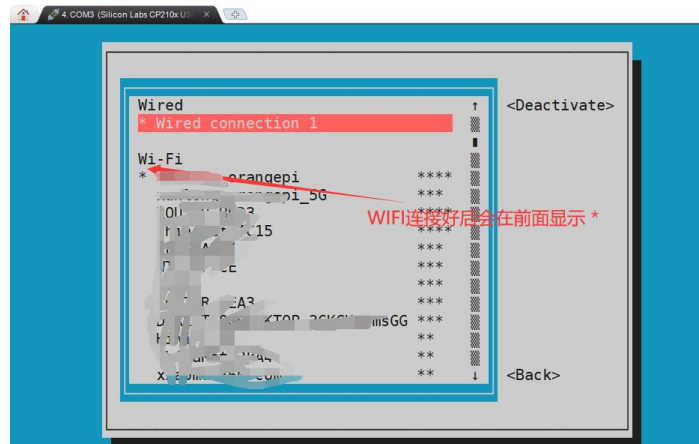


7) 然后会弹出输入密码的对话框，在 **Pssword** 内输入对应的密码然后回车就会开始连接 WIFI。





8) WIFI 连接成功后会在已连接的 WIFI 名称前显示一个 “\*”。



9) 通过 `ip a s wlan0` 命令可以查看 wifi 的 IP 地址。

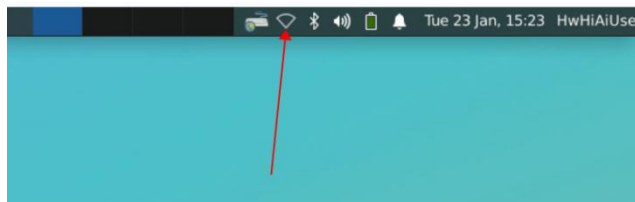
```
(base) HwHiAiUser@orangepiaipro:~$ ip a s wlan0
4: wlan0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP
group default qlen 1000
    link/ether 54:f2:9f:7b:ba:36 brd ff:ff:ff:ff:ff:ff
    inet 10.31.2.93/16 brd 10.31.255.255 scope global dynamic noprefixroute wlan0
        valid_lft 43003sec preferred_lft 43003sec
    inet6 fe80::5297:7036:a33c:bb93/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

10) 使用 `ping` 命令可以测试 wifi 网络的连通性，`ping` 命令可以通过 `Ctrl+C` 快捷键来中断运行。

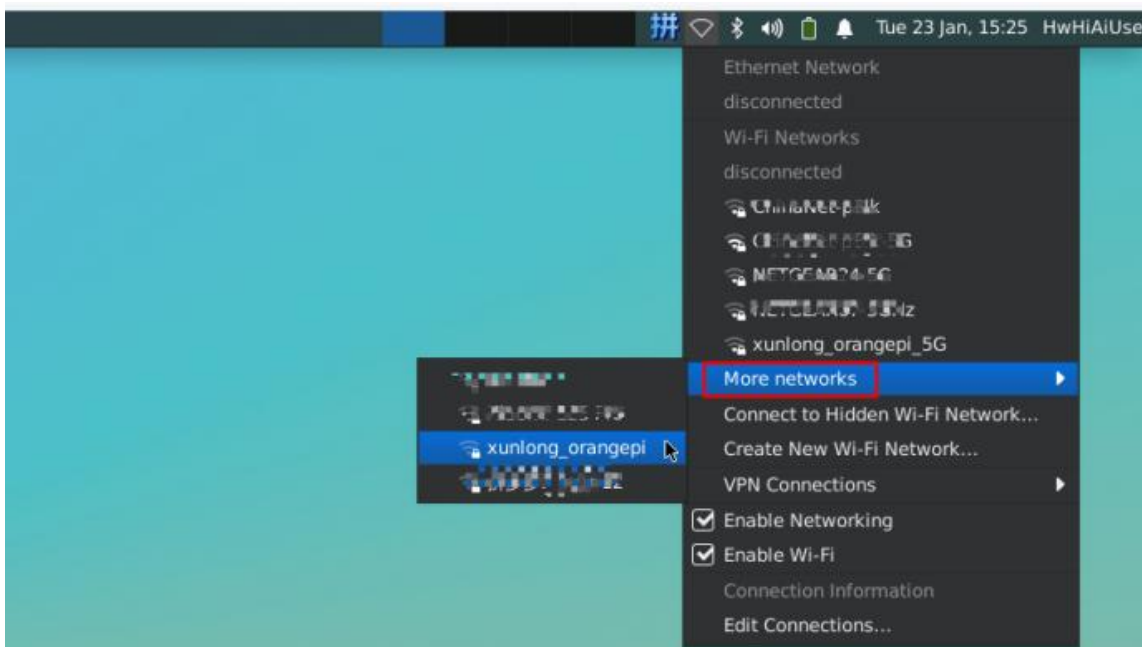
```
(base) HwHiAiUser@orangepiaipro:~$ ping www.orangeapi.org -I wlan0
PING www.orangeapi.org (123.57.147.237) from 10.31.2.93 wlan0: 56(84) bytes of data.
64 bytes from 123.57.147.237 (123.57.147.237): icmp_seq=1 ttl=53 time=47.1 ms
64 bytes from 123.57.147.237 (123.57.147.237): icmp_seq=2 ttl=53 time=44.3 ms
64 bytes from 123.57.147.237 (123.57.147.237): icmp_seq=3 ttl=53 time=45.0 ms
64 bytes from 123.57.147.237 (123.57.147.237): icmp_seq=4 ttl=53 time=71.0 ms
^C
--- www.orangeapi.org ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3002ms
rtt min/avg/max/mdev = 44.377/51.902/71.082/11.119 ms
```

## 2.14.2.3. 桌面版镜像的测试方法

1) 首先点击桌面右上角的网络配置图标。



2) 在弹出的下拉框中点击 **More networks** 可以看到所有扫描到的 WIFI 热点，然后选择想要连接的 WIFI 热点。

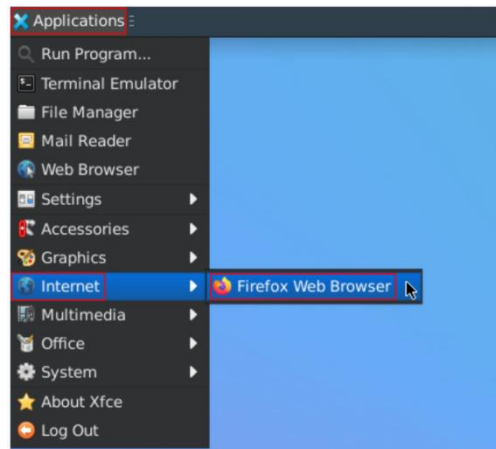


3) 然后输入 WIFI 热点的密码，再点击 **Connect** 就会开始连接 WIFI。





4) 连接好 WIFI 后，可以打开浏览器查看是否能上网，浏览器的入口如下图所示：



5) 打开浏览器后如果能打开其他网页说明 WIFI 连接正常。



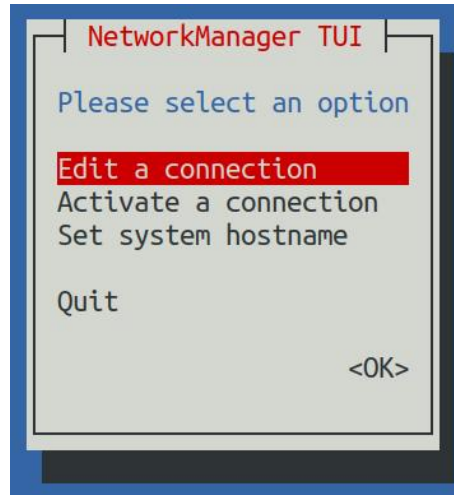
## 2. 14. 3. 设置静态 IP 地址的方法

### 2. 14. 3. 1. 使用 nmtui 命令来设置静态 IP 地址

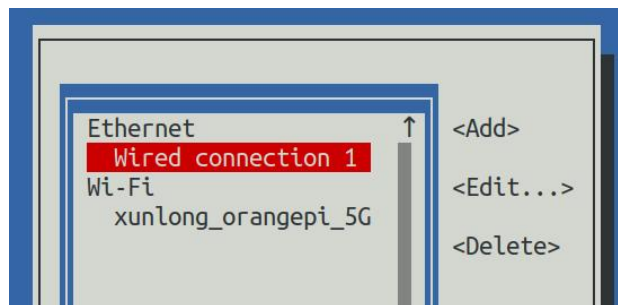
1) 首先运行 **nmtui** 命令。

```
(base) HwHiAiUser@orangepiaipro:~$ sudo nmtui
```

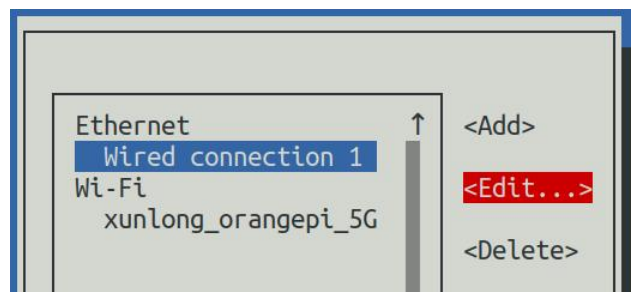
2) 然后选择 **Edit a connection** 并按下回车键。



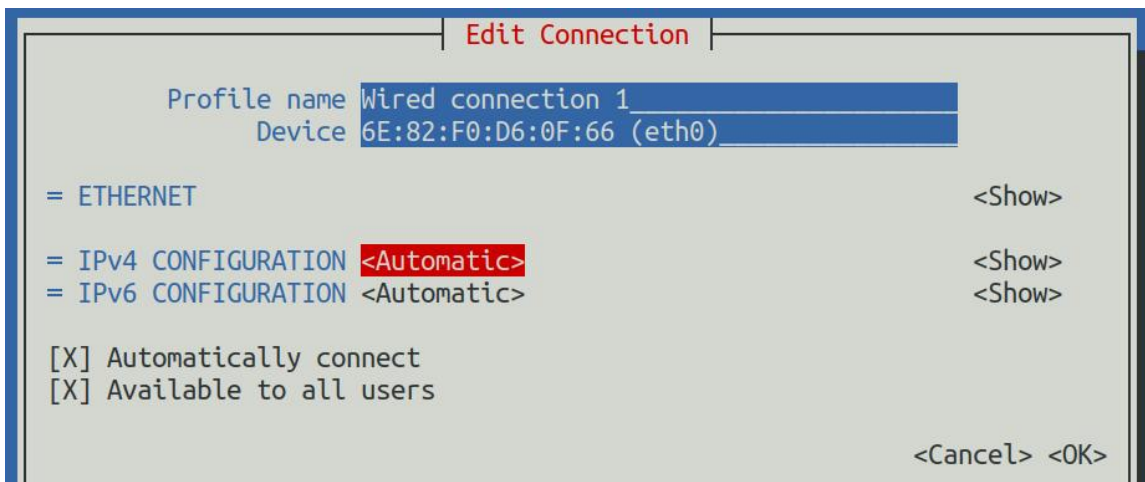
3) 然后选择需要设置静态 IP 地址的网络接口，比如设置 **Ethernet** 接口的静态 IP 地址选择 **Wired connection 1** 就可以了。



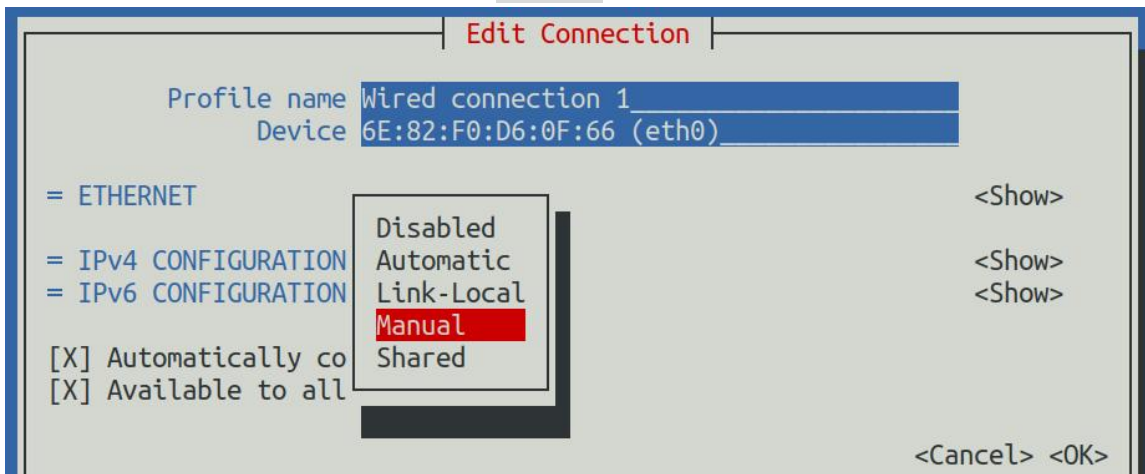
4) 然后通过 **Tab** 键选择 **Edit** 并按下回车键。



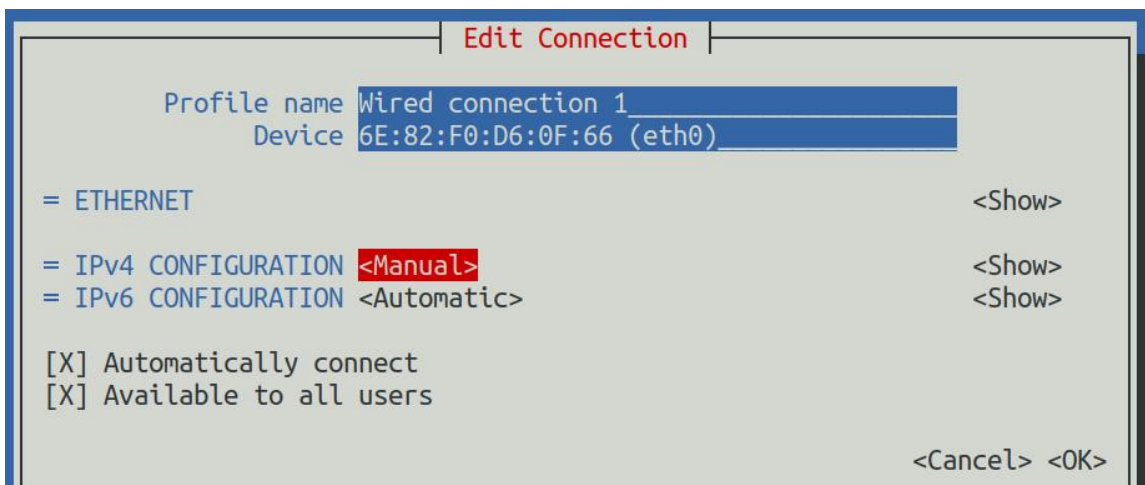
5) 然后通过 **Tab** 键将光标移动到下图所示的 **<Automatic>** 位置进行 IPv4 的配置。



6) 然后回车，通过上下方向键选择 **Manual**，然后回车确定。

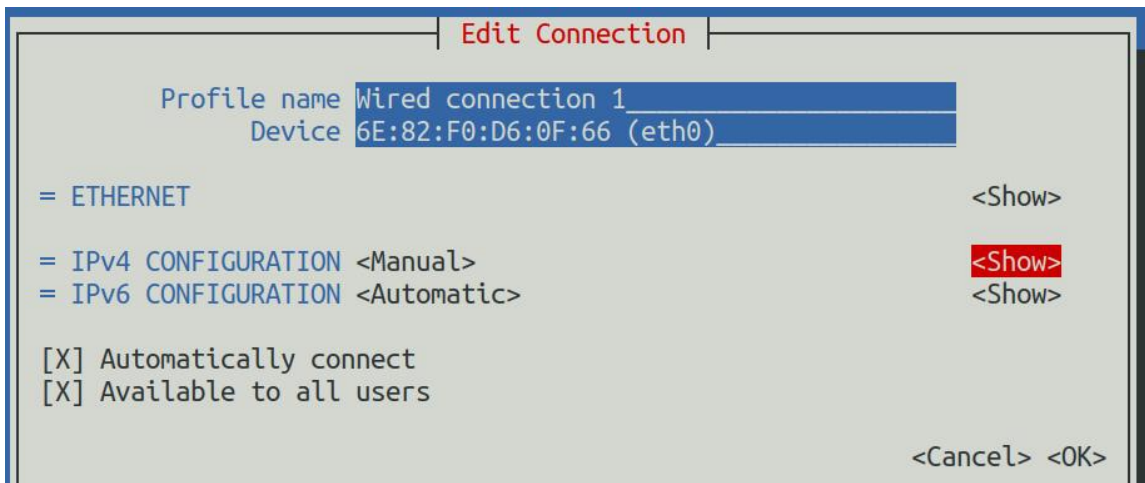


7) 选择完后的显示如下图所示：

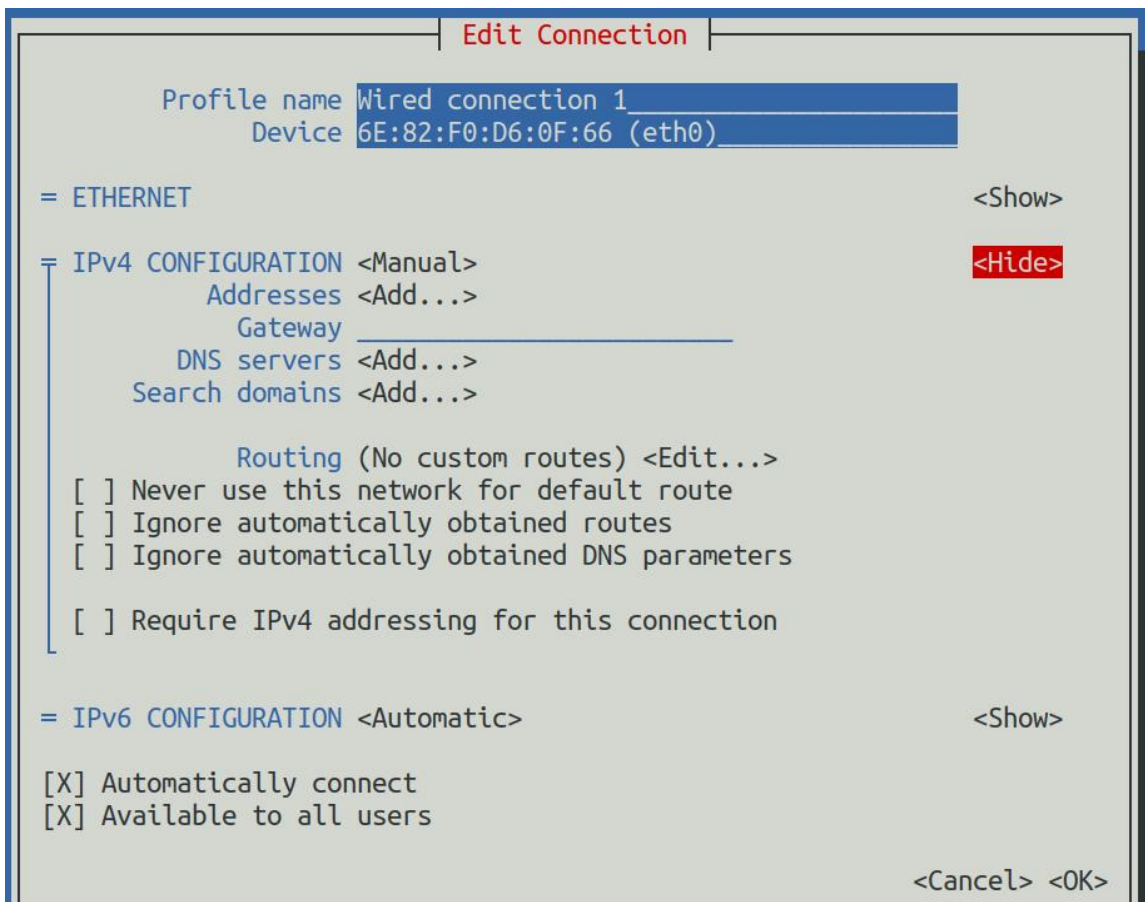




8) 然后通过 Tab 键将光标移动到<Show>。



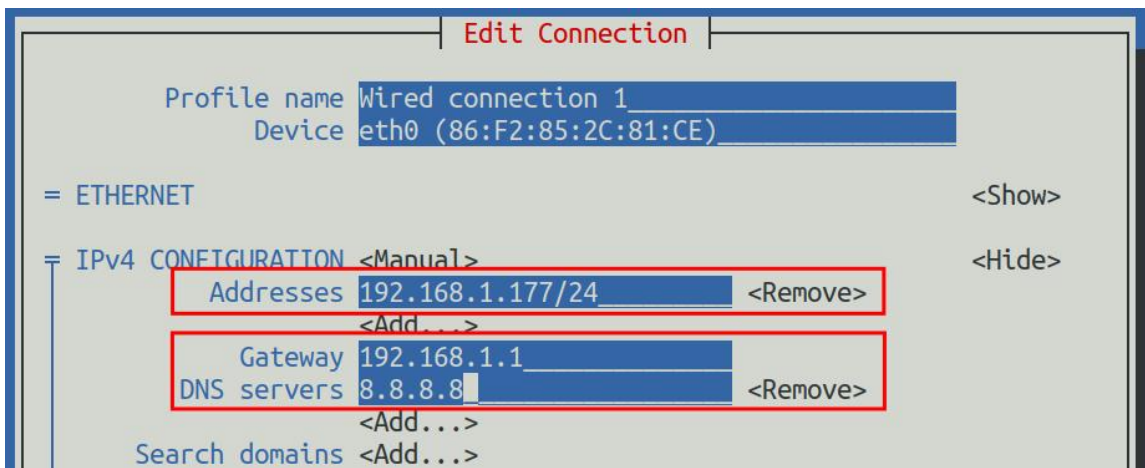
9) 然后回车，回车后会弹出下面的设置界面。



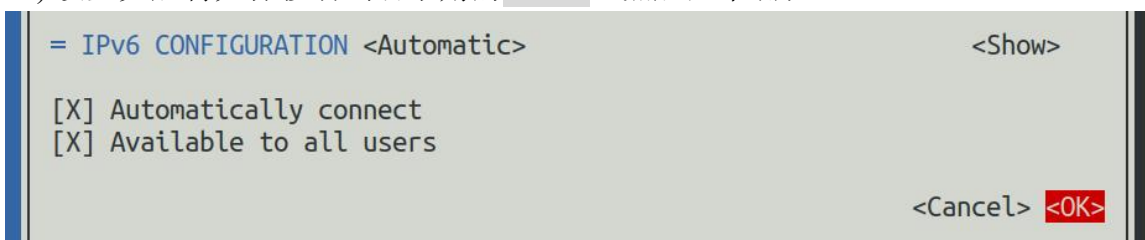
10) 然后就可以在下图所示的位置设置 IP 地址(Addresses)、网关(Gateway)和 DNS



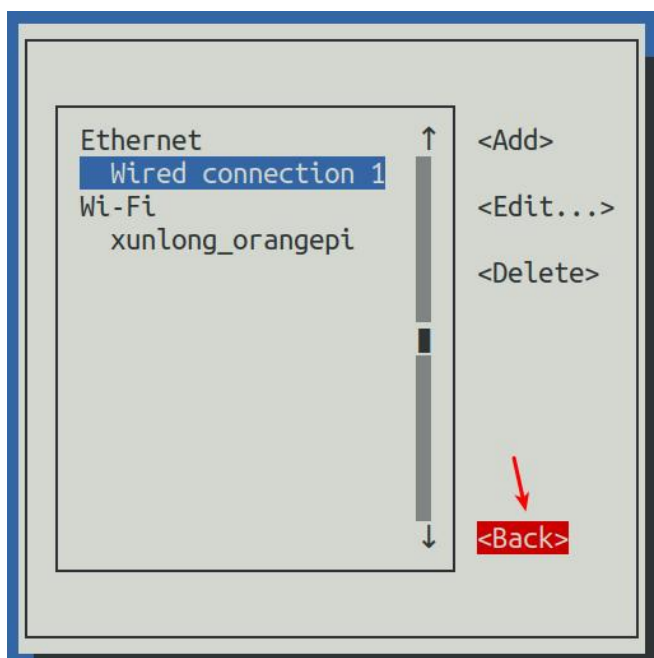
服务器的地址（里面还有很多其他设置选项，请自行探索），**请根据自己的具体需求来设置，下图中设置的值只是一个示例。**



11) 设置完后将光标移动到右下角的<OK>，然后回车确认。

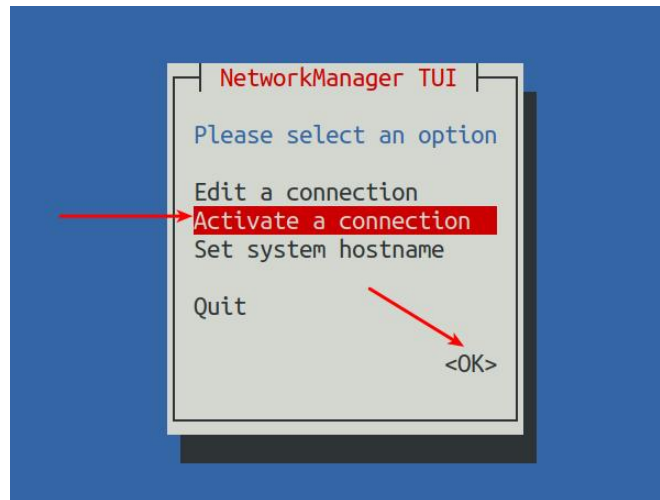


12) 然后点击<Back>回退到上一级选择界面。

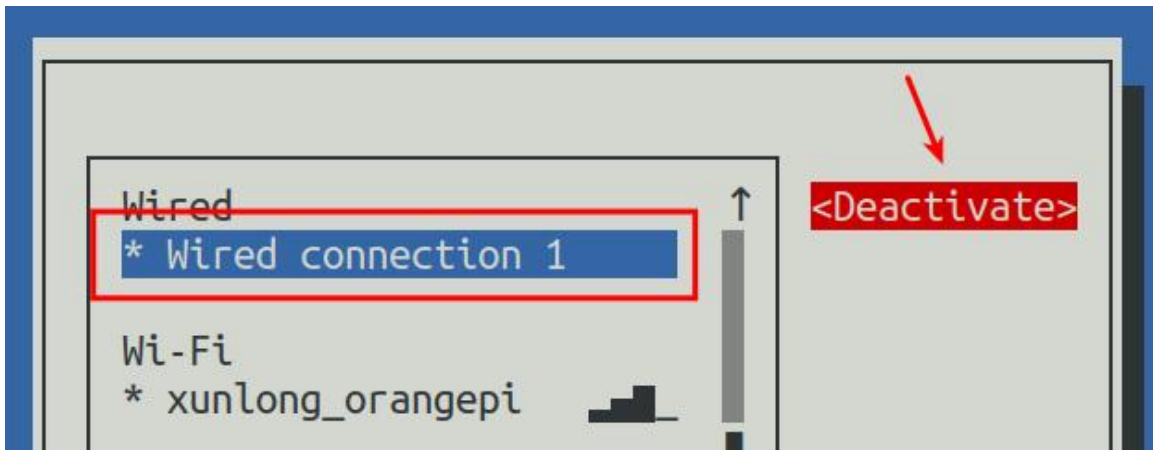




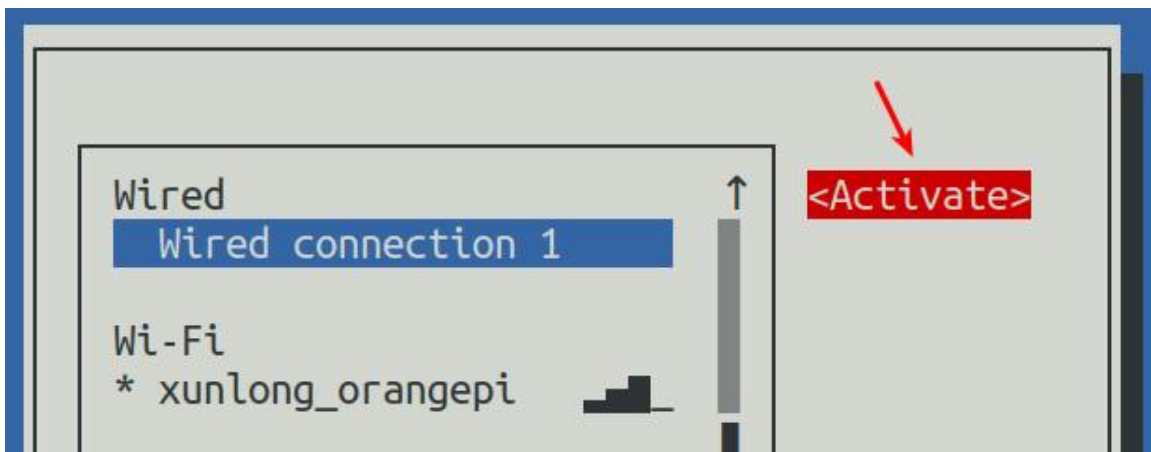
13) 然后选择 **Activate a connection**，再将光标移动到 **<OK>**，最后点击回车。



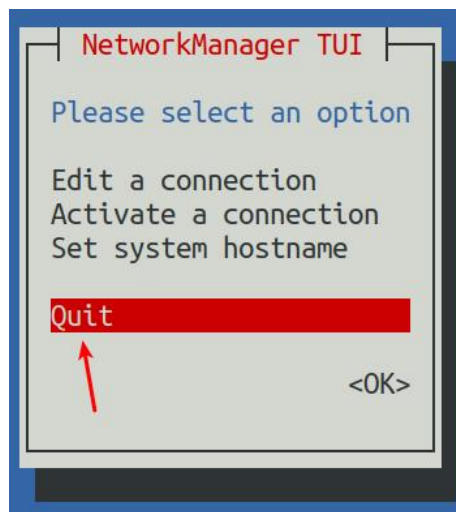
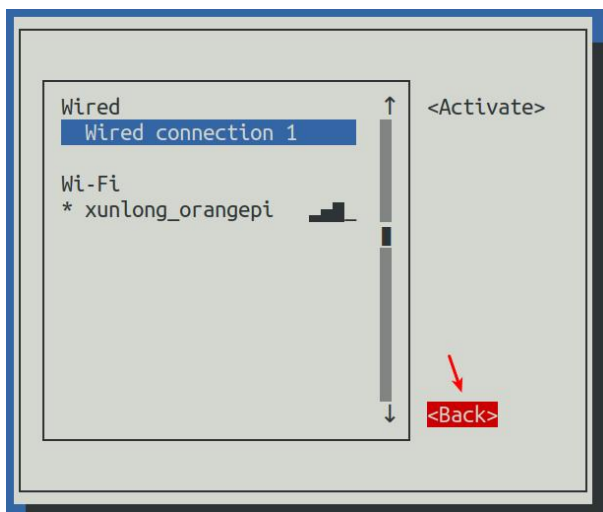
14) 然后选择需要设置的网络接口，比如 **Wired connection 1**，然后将光标移动到 **<Deactivate>**，再按下回车键禁用 **Wired connection 1**。



15) 然后请不要移动光标，再按下回车键重新使能 **Wired connection 1**，这样前面设置的静态 IP 地址就会生效了。



16) 然后通过 **<Back>** 和 **Quit** 按钮就可以退出 nmtui。



17) 然后通过 **ip a s eth0** 就能看到网口的 IP 地址已经变成前面设置的静态 IP 地址了。

```
(base) HwHiAiUser@orangepiaipro:~$ ip a s eth0
```

18) 然后就可以测试网络的连通性来检查 IP 地址是否配置 OK 了，**ping** 命令可以通过 **Ctrl+C** 快捷键来中断运行。

```
(base) HwHiAiUser@orangepiaipro:~$ ping 192.168.x.xxx -I eth0
```

## 2. 14. 3. 2. 使用 nmcli 命令来设置静态 IP 地址

1) 如果要设置网口的静态 IP 地址，请先将网线插入开发板，如果需要设置 WIFI



的静态 IP 地址，请先连接好 WIFI，然后再开始设置静态 IP 地址。

2) 然后通过 **nmcli con show** 命令可以查看网络设备的名字，如下所示：

- a. **orangepi** 为 WIFI 网络接口的名字（名字不一定相同）。
- b. **Wired connection 1** 为以太网接口的名字。

```
(base) HwHiAiUser@orangepiaipro:~$ nmcli con show
NAME                UUID                                TYPE    DEVICE
orangepi            cfc4f922-ae48-46f1-84e1-2f19e9ec5e2a  wifi    wlan0
Wired connection 1  9db058b7-7701-37b8-9411-efc2ae8bfa30  ethernet eth0
```

3) 然后输入下面的命令，其中：

- a. **"Wired connection 1"** 表示设置以太网口的静态 IP 地址，如果需要设置 WIFI 的静态 IP 地址，请修改为 WIFI 网络接口对应的名字（通过 **nmcli con show** 命令可以获取到）。
- b. **ipv4.addresses** 后面是要设置的静态 IP 地址，可以修改为自己想要设置的值。
- c. **ipv4.gateway** 表示网关的地址。

```
(base) HwHiAiUser@orangepiaipro:~$ sudo nmcli con mod "Wired connection 1" \
ipv4.addresses "192.168.1.110" \
ipv4.gateway "192.168.1.1" \
ipv4.dns "8.8.8.8" \
ipv4.method "manual"
```

4) 然后重启 Linux 系统。

```
(base) HwHiAiUser@orangepiaipro:~$ sudo reboot
```

5) 然后重新进入 Linux 系统使用 **ip addr show eth0** 命令就可以看到 IP 地址已经设置为想要的值了。

```
(base) HwHiAiUser@orangepiaipro:~$ ip addr show eth0
```

## 2. 15. SSH 远程登录开发板

Linux 系统默认都开启了 ssh 远程登录，并且允许 root 用户登录系统。ssh 登录前首先需要确保以太网或者 wifi 网络已连接，然后使用 **ip addr** 命令或者通过查看路由器的方式获取开发板的 IP 地址。



## 2. 15. 1. Ubuntu 下 SSH 远程登录开发板

1) 获取开发板的 IP 地址。

2) 然后就可以通过 ssh 命令远程登录 Linux 系统。

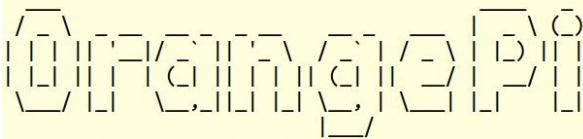
```
test@test:~$ ssh root@192.168.2.xxx      (需要替换为开发板的 IP 地址，不要照抄)
root@192.168.2.xx's password:           (在这里输入密码，默认密码为 Mind@123)
```

注意，输入密码的时候，**屏幕上是不会显示输入的密码的具体内容的**，请不要以为是有什么故障，输入完后直接回车即可。

如果提示拒绝连接，只要使用的是 Orange Pi 提供的镜像，**就请不要怀疑 Mind@123 这个密码是不是不对**，而是要找其他原因。

3) 成功登录系统后的显示如下图所示：

```
orangeipi@orangeipi-M600:~$ ssh root@192.168.2.100
root@192.168.2.100's password:
```



Welcome to Orange Pi Ai Pro

This system is based on Ubuntu 22.04 LTS (GNU/Linux 5.10.0+ aarch64)

This system is only applicable to individual developers and cannot be used for commercial purposes.

By using this system, you have agreed to the Huawei Software License Agreement.

Please refer to the agreement for details on <https://www.hiascend.com/software/protocol>

Last login: Fri Jan 26 16:55:15 2024 from 192.168.2.220

-bash: warning: setlocale: LC\_ALL: cannot change locale (en\_US.UTF-8)

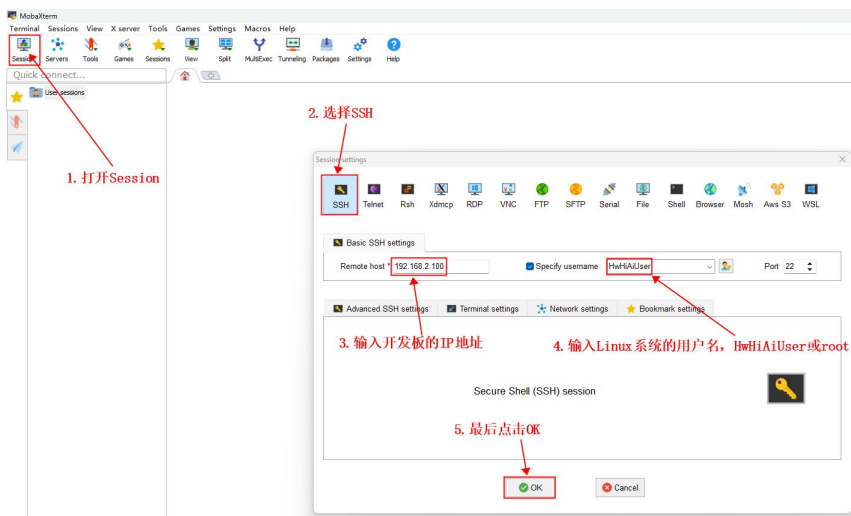
(base) root@orangepiaipro:~#

## 2. 15. 2. Windows 下 SSH 远程登录开发板

1) 首先获取开发板的 IP 地址。

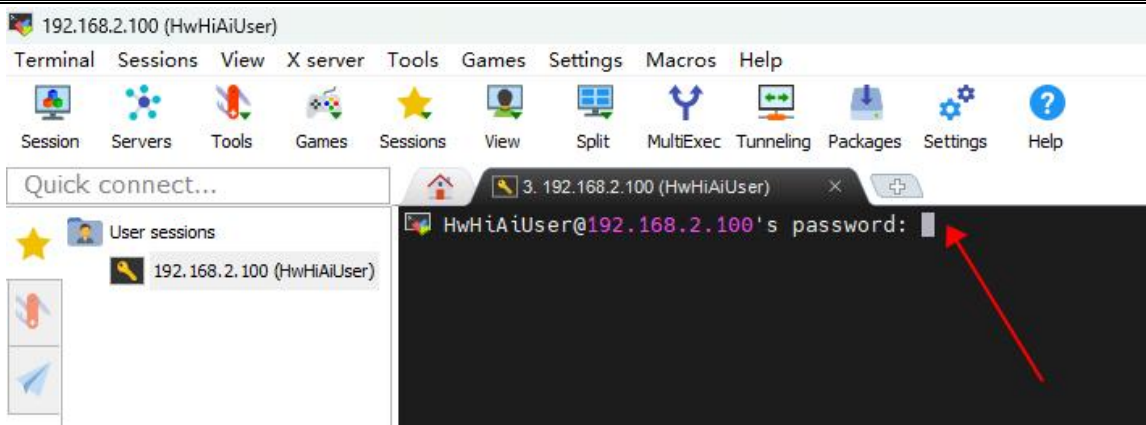
2) 在 Windows 下可以使用 MobaXterm 远程登录开发板，首先新建一个 ssh 会话。

- a. 打开 **Session**。
- b. 然后在 **Session Setting** 中选择 **SSH**。
- c. 然后在 **Remote host** 中输入开发板的 IP 地址。
- d. 然后在 **Specify username** 中输入 Linux 系统的用户名 **root** 或 **HwHiAiUser**。
- e. 最后点击 **OK** 即可。

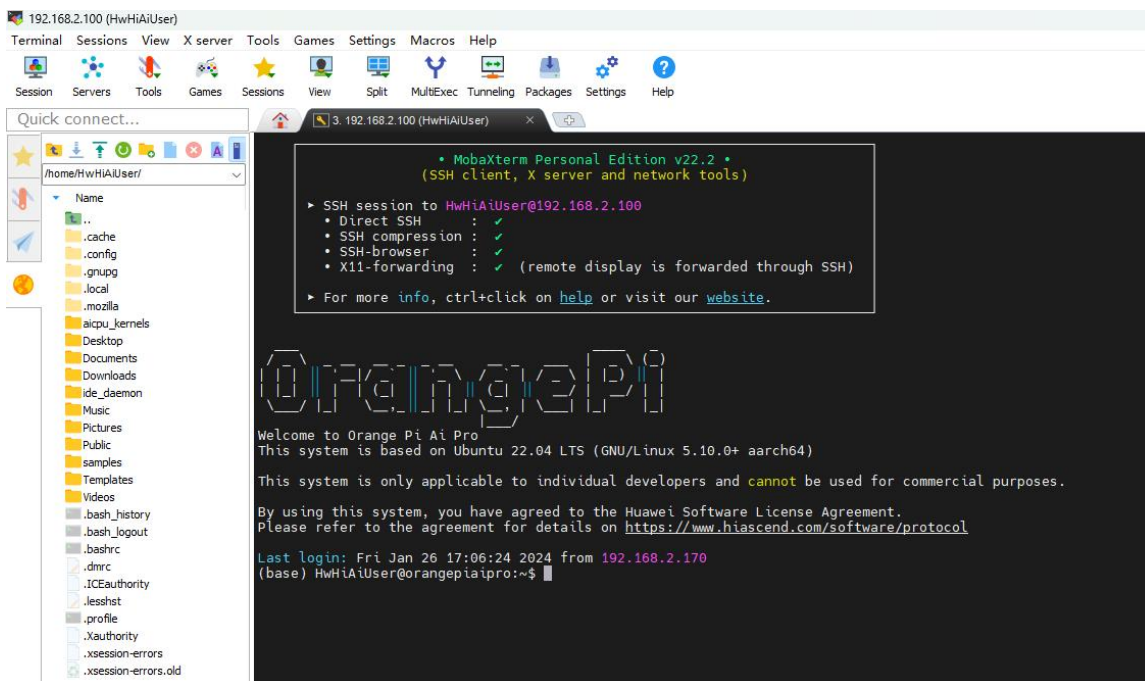


3) 然后会提示输入密码，默认 **root** 和 **HwHiAiUser** 用户的密码都为 **Mind@123**。

注意，输入密码的时候，**屏幕上是不会显示输入的密码的具体内容的**，请不要以为是有什么故障，输入完后直接回车即可。



4) 成功登录系统后的显示如下图所示



## 2. 16. 上传文件到开发板 Linux 系统中的方法

### 2. 16. 1. 在 Ubuntu PC 中上传文件到开发板 Linux 系统中的方法

#### 2. 16. 1. 1. 使用 scp 命令上传文件的方法

1) 使用 scp 命令可以在 Ubuntu PC 中上传文件到开发板的 Linux 系统中，具体命令如下所示：

- file\_path:** 需要替换为要上传文件的路径。
- root:** 为开发板 Linux 系统的用户名，也可以替换成其他的，比如 **HwHiAiUser**。
- 192.168.xx.xx:** 为开发板的 IP 地址，请根据实际情况进行修改。
- /root:** 开发板 Linux 系统中的路径，也可以修改为其他的路径。

```
test@test:~$ scp file_path root@192.168.xx.xx:/root/
```

2) 如果要上传文件夹，需要加上 **-r** 参数。

```
test@test:~$ scp -r dir_path root@192.168.xx.xx:/root/
```



3) scp 还有更多的用法, 请使用下面的命令查看 man 手册。

```
test@test:~$ man scp
```

## 2. 16. 1. 2. 使用 filezilla 上传文件的方法

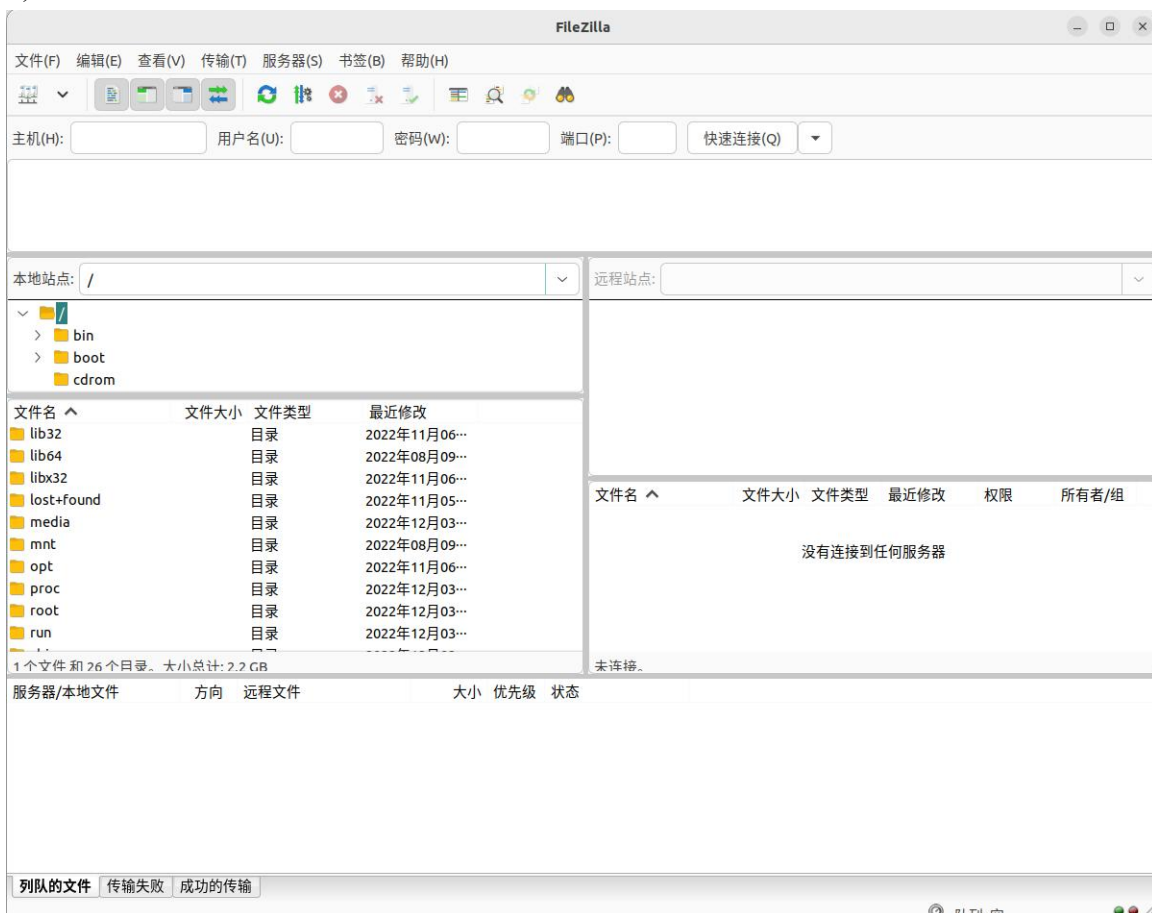
1) 首先在 Ubuntu PC 中安装 filezilla。

```
test@test:~$ sudo apt-get install -y filezilla
```

2) 然后使用下面的命令打开 filezilla。

```
test@test:~$ filezilla
```

3) filezilla 打开后的界面如下所示, 此时右边远程站点下面显示的是空的。



4) 连接开发板的方法如下图所示:



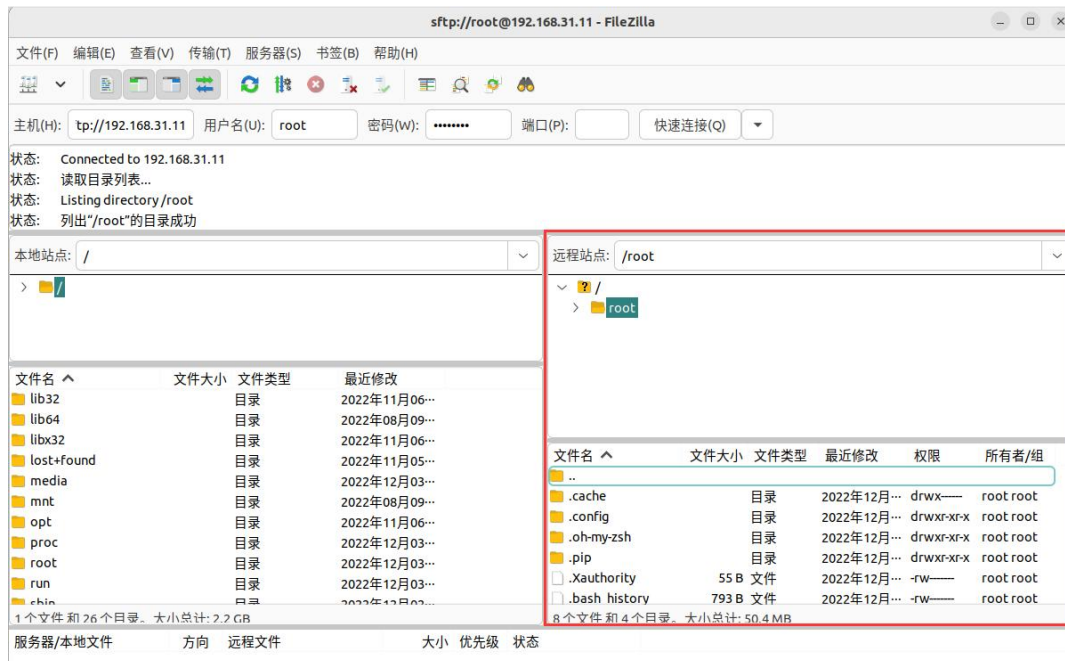
5) 然后选择**保存密码**，再点击**确定**。



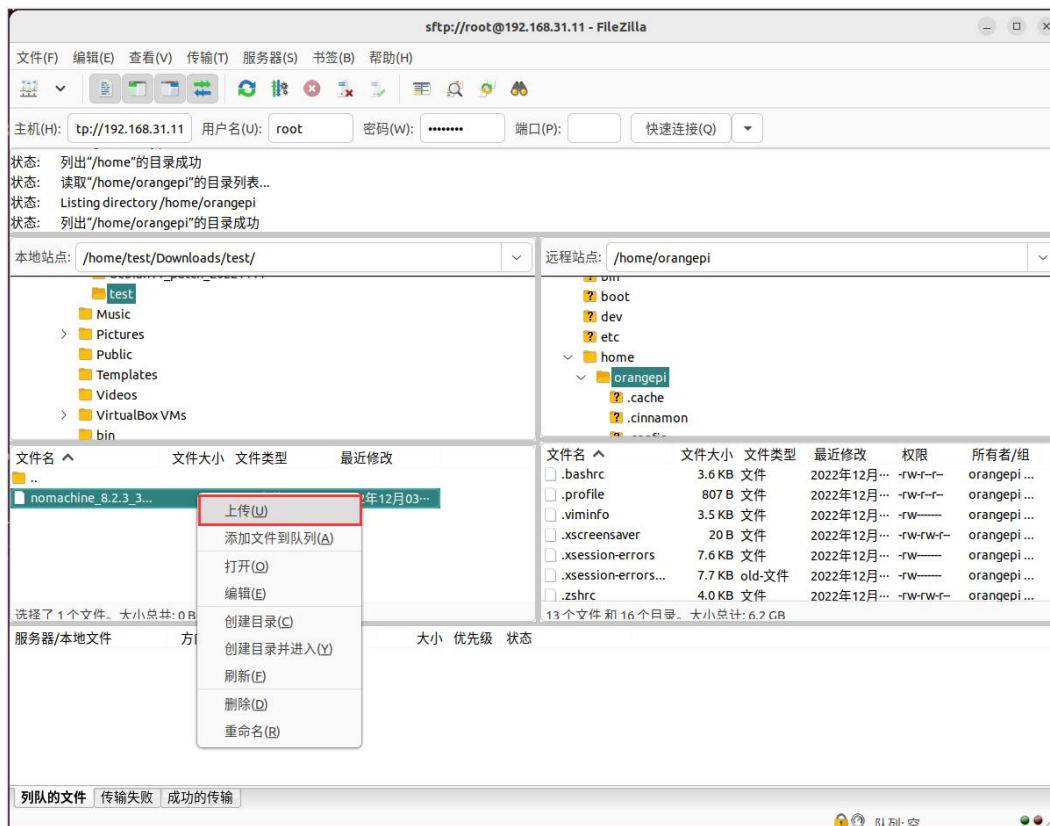
6) 然后选择**总是信任该主机**，再点击**确定**。



7) 连接成功后在 filezilla 软件的右边就可以看到开发板 linux 文件系统的目录结构了。



8) 然后在 filezilla 软件的右边选择要上传到开发板中的路径，再在 filezilla 软件的左边选中 Ubuntu PC 中要上传的文件，再点击鼠标右键，再点击上传选项就会开始上传文件到开发板中了。





9) 上传完成后就可以去开发板 Linux 系统中的对应路径中查看上传的文件了。

10) 上传文件夹的方法和上传文件的方法是一样的，这里就不再赘述了。

## 2. 16. 2. 在 Windows PC 中上传文件到开发板 Linux 系统中的方法

### 2. 16. 2. 1. 使用 filezilla 上传文件的方法

1) 首先下载 filezilla 软件 Windows 版本的安装文件，下载链接如下所示：

<https://filezilla-project.org/download.php?type=client>



**Please select your edition of FileZilla Client**

|                                     | FileZilla | FileZilla with manual | FileZilla Pro | FileZilla Pro + CLI |
|-------------------------------------|-----------|-----------------------|---------------|---------------------|
| Standard FTP                        | Yes       | Yes                   | Yes           | Yes                 |
| FTP over TLS                        | Yes       | Yes                   | Yes           | Yes                 |
| SFTP                                | Yes       | Yes                   | Yes           | Yes                 |
| Comprehensive PDF manual            | -         | Yes                   | Yes           | Yes                 |
| Amazon S3                           | -         | -                     | Yes           | Yes                 |
| Backblaze B2                        | -         | -                     | Yes           | Yes                 |
| Dropbox                             | -         | -                     | Yes           | Yes                 |
| Microsoft OneDrive                  | -         | -                     | Yes           | Yes                 |
| Google Drive                        | -         | -                     | Yes           | Yes                 |
| Google Cloud Storage                | -         | -                     | Yes           | Yes                 |
| Microsoft Azure Blob + File Storage | -         | -                     | Yes           | Yes                 |
| WebDAV                              | -         | -                     | Yes           | Yes                 |
| OpenStack Swift                     | -         | -                     | Yes           | Yes                 |
| Box                                 | -         | -                     | Yes           | Yes                 |
| Site Manager synchronization        | -         | -                     | Yes           | Yes                 |
| Command-line interface              | -         | -                     | -             | Yes                 |
| Batch transfers                     | -         | -                     | -             | Yes                 |

然后选择这里下载

**Download** **Select** **Select** **Select**

2) 下载的安装包如下所示，然后双击直接安装即可。

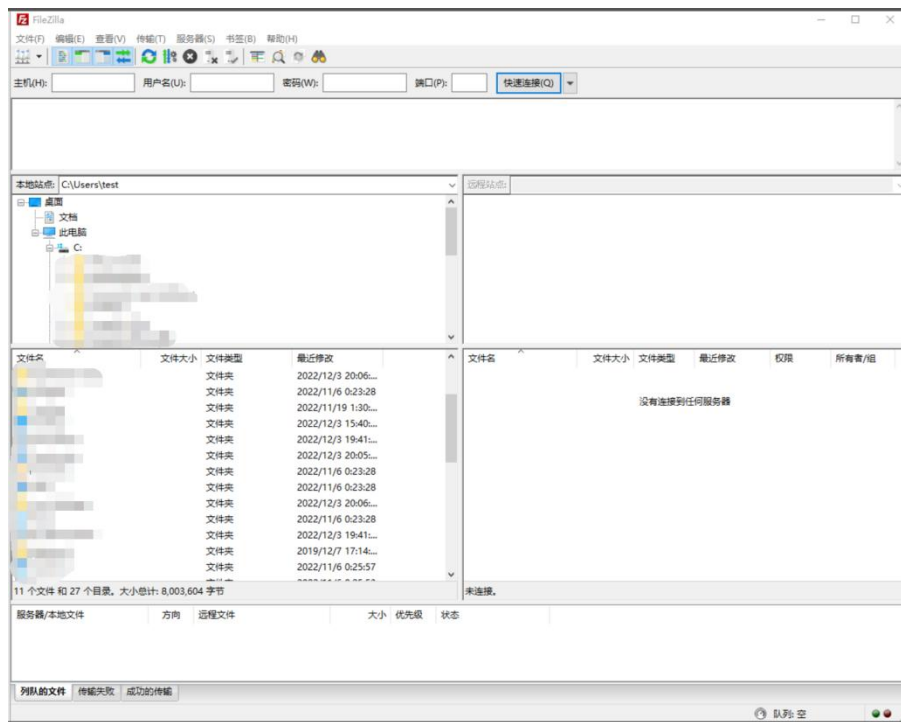
**FileZilla\_Server\_x.x.x\_win64-setup.exe**



安装过程中，下面的安装界面请选择 **Decline**，然后再选择 **Next>**。



3) filezilla 打开后的界面如下所示，此时右边远程站点下面显示的是空的。



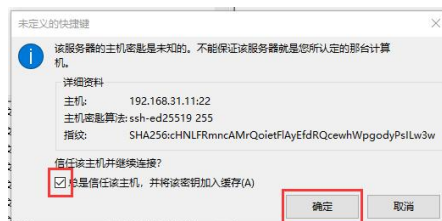
4) 连接开发板的方法如下图所示：



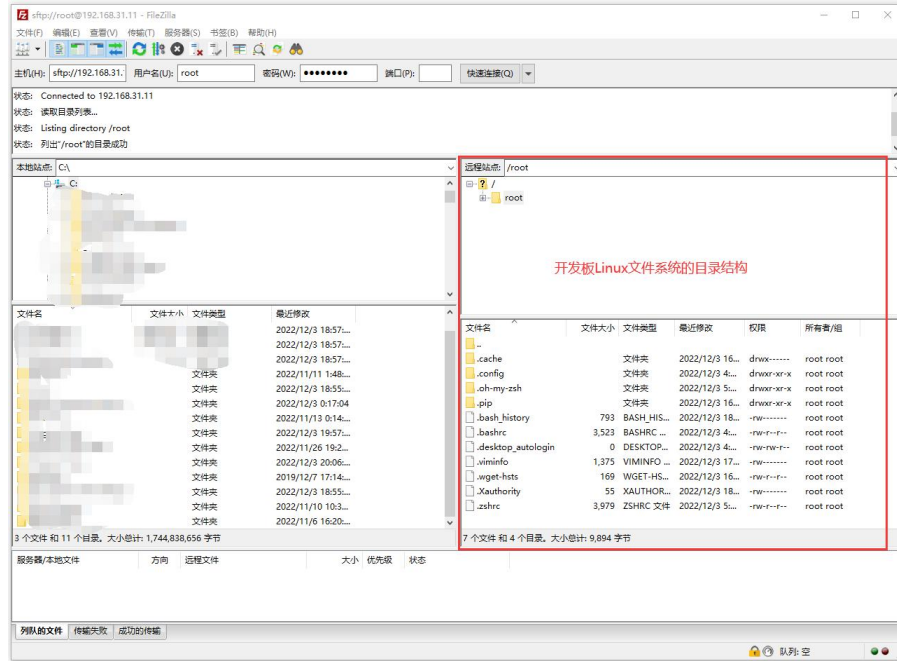
5) 然后选择**保存密码**，再点击**确定**。



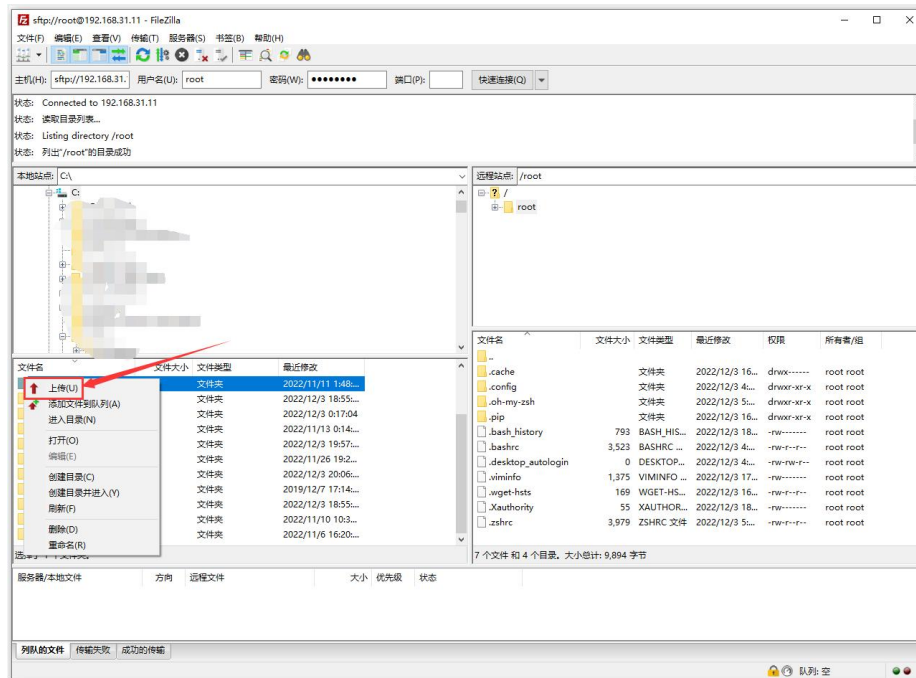
6) 然后选择**总是信任该主机**，再点击**确定**。



7) 连接成功后在 filezilla 软件的右边就可以看到开发板 linux 文件系统的目录结构了。



8) 然后在 filezilla 软件的右边选择要上传到开发板中的路径，再在 filezilla 软件的左边选中 Windows PC 中要上传的文件，再点击鼠标右键，再点击上传选项就会开始上传文件到开发板中了。



9) 上传完成后就可以去开发板 Linux 系统中的对应路径中查看上传的文件了。

10) 上传文件夹的方法和上传文件的方法是一样的，这里就不再赘述了。

## 2. 17. MIPI LCD 屏幕的测试说明

### 2. 17. 1. 树莓派 5 寸屏幕的组装方法

1) 首先准备需要的配件。

a. 树莓派 5 寸 MIPI LCD 显示屏。



b. 15pin MIPI 排线。

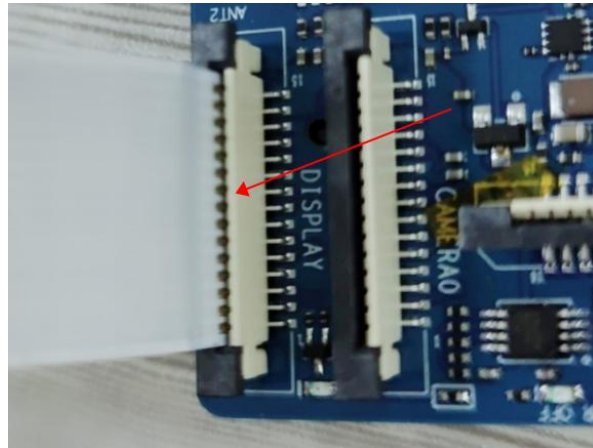


2) 然后将 15pin MIPI 排线按照下图所示的方式连接到树莓派 5 寸屏幕上（注意绝缘面的朝向）。





3) 最后将 LCD 屏幕连接到开发板的 MIPI LCD 接口上。



## 2. 17. 2. 输出图片到 LCD 屏幕的方法

使用 MIPI LCD 屏幕显示 Linux 系统桌面的功能还在开发中。目前 MIPI LCD 屏幕只能用样例程序显示一张图片，步骤如下所示：

1) 首先连接好树莓派 5 寸 LCD 屏幕，具体方法情况参考[树莓派 5 寸屏幕的组装方法](#)一小节的说明。

2) 然后进入测试程序所在的路径。

```
(base) HwHiAiUser@orangepiaipro:~$ sudo -i
(base) root@orangepiaipro:~# cd /opt/opi_test/lcd
(base) root@orangepiaipro:/opt/opi_test/lcd# ls
raspberry_demo test.sh update_dt.sh ut_800x480_nv12.yuv
```

3) 然后运行 **update\_dt.sh** 脚本更新下 dt.img（关闭 drm 的配置，打开 vdp 的配置）。**update\_dt.sh** 脚本运行完后会自动重启 Linux 系统。

```
(base) root@orangepiaipro:/opt/opi_test/lcd# ./update_dt.sh
```

4) 重启后再次进入 LCD 测试程序所在的路径，然后运行 **test.sh** 脚本就会播放一张图片到 LCD 屏幕。

```
(base) root@orangepiaipro:/opt/opi_test/lcd# ./test.sh
```



5) 使用 **Ctrl+C** 快捷键可以停止显示图片。

6) 测试完 LCD 播放图片后，再切换回 HDMI0 显示 Linux 系统桌面的方法如下所示：

a. 首先切换到 root 用户，并进入 **/opt/opi\_test/hdmi\_desktop** 目录。

```
(base) root@orangepiaipro:~# cd /opt/opi_test/hdmi_desktop  
(base) root@orangepiaipro:/opt/opi_test/hdmi_desktop# ls  
update_dt.sh
```

b. 然后运行 **update\_dt.sh** 脚本更新下 dt.img（关闭 vdp 的配置，打开 drm 的配置）。**update\_dt.sh** 脚本运行完后会自动重启 Linux 系统。

```
(base) root@orangepiaipro:/opt/opi_test/hdmi_desktop# ./update_dt.sh
```

c. 重启后就能看到 HDMI 显示器会显示 Linux 系统的桌面了。

## 2. 18. MIPI 摄像头的测试说明

目前 MIPI 摄像头测试只能用样例程序拍一张图片，步骤如下所示：

1) 首先需要准备一个树莓派 IMX219 摄像头。



2) 然后将摄像头连接到开发板的 MIPI 摄像头接口中。测试一个摄像头拍照时，请使用 CAMERA0 接口。



3) 然后登录 Linux 系统的桌面，打开一个终端，再进入测试程序所在的路径。

```
(base) HwHiAiUser@orangepiaipro:~$ sudo -i
(base) root@orangepiaipro:~# cd /opt/opi_test/camera
(base) root@orangepiaipro:/opt/opi_test/camera# ls
sample_hdmi test_one.sh test_two.sh update_dt.sh vi_l1_sample
```

4) 然后运行下面的命令就会使用 CAMERA0 拍摄一张照片。

```
(base) root@orangepiaipro:/opt/opi_test/camera# ./vi_l1_sample 1 1 1
```

5) 拍摄好的图片如下所示：

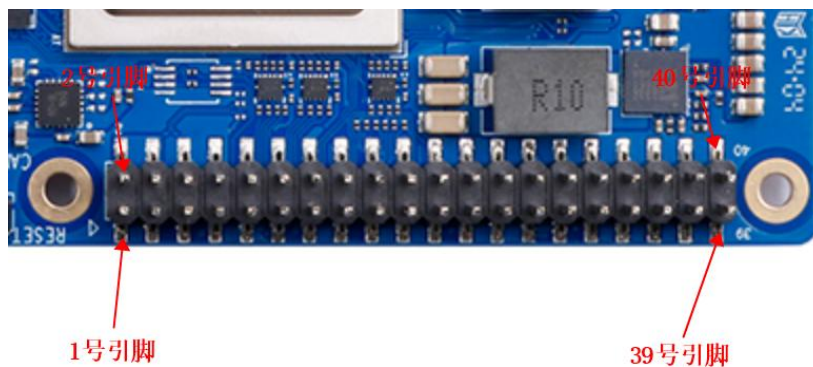
```
(base) root@orangepiaipro:/opt/opi_test/camera# ls *.yuv
vi_pipe0_chn0_w1920_h1080.yuv
```

6) 然后在 Linux 桌面中使用 ffplay 命令可以查看下拍摄的图片。

```
(base) root@orangepiaipro:/opt/opi_test/camera# ffplay -pix_fmt yuv420p -video_size
1920*1080 ./vi_pipe0_chn0_w1920_h1080.yuv
```

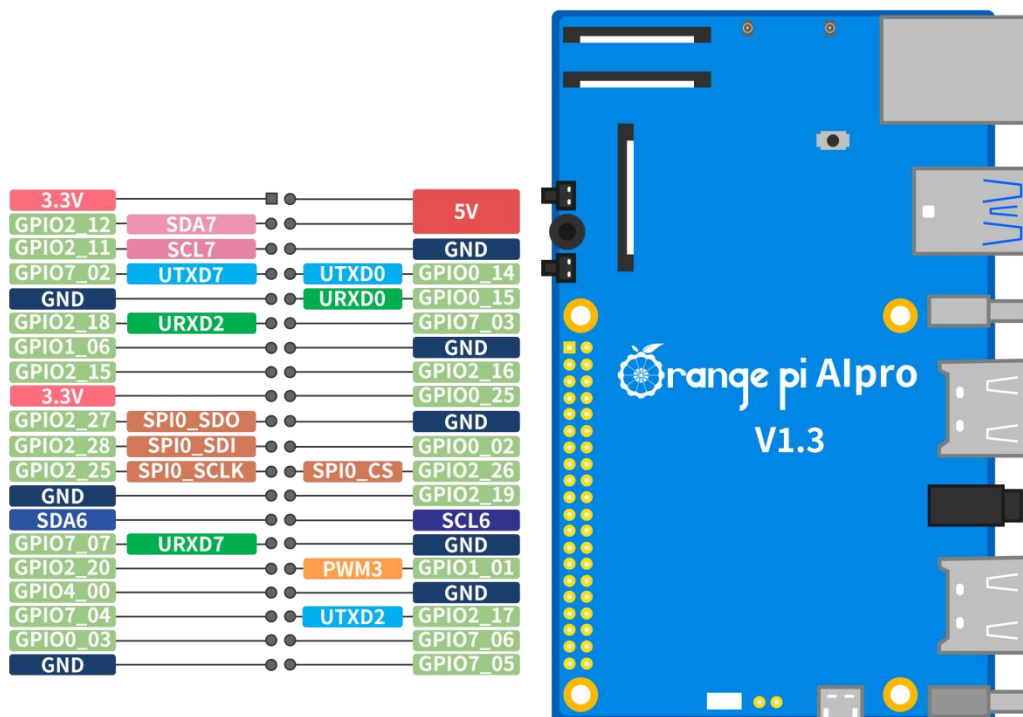
## 2. 19. 40 Pin 接口引脚功能说明

开发板的 40 pin 接口引脚的顺序如下图所示：



开发板 40 pin 接口引脚的功能如下表所示：

| GPIO序号 | GPIO     | 功能        | 引脚 | 引脚 | 功能      | GPIO     | GPIO序号 |
|--------|----------|-----------|----|----|---------|----------|--------|
|        |          | 3.3V      | 1  | 2  | 5V      |          |        |
| 76     | GPIO2_12 | SDA7      | 3  | 4  | 5V      |          |        |
| 75     | GPIO2_11 | SCL7      | 5  | 6  | GND     |          |        |
| 226    | GPIO7_02 | UTXD7     | 7  | 8  | UTXD0   | GPIO0_14 | 14     |
|        |          | GND       | 9  | 10 | URXD0   | GPIO0_15 | 15     |
| 82     | GPIO2_18 | URXD2     | 11 | 12 |         | GPIO7_03 | 227    |
| 38     | GPIO1_06 |           | 13 | 14 | GND     |          |        |
| 79     | GPIO2_15 |           | 15 | 16 |         | GPIO2_16 | 80     |
|        |          | 3.3V      | 17 | 18 |         | GPIO0_25 | 25     |
| 91     | GPIO2_27 | SPI0_SDO  | 19 | 20 | GND     |          |        |
| 92     | GPIO2_28 | SPI0_SDI  | 21 | 22 |         | GPIO0_02 | 2      |
| 89     | GPIO2_25 | SPI0_SCLK | 23 | 24 | SPI0_CS | GPIO2_26 | 90     |
|        |          | GND       | 25 | 26 |         | GPIO2_19 | 83     |
|        |          | SDA6      | 27 | 28 | SCL6    |          |        |
| 231    | GPIO7_07 | URXD7     | 29 | 30 | GND     |          |        |
| 84     | GPIO2_20 |           | 31 | 32 | PWM3    | GPIO1_01 | 33     |
| 128    | GPIO4_00 |           | 33 | 34 | GND     |          |        |
| 228    | GPIO7_04 |           | 35 | 36 | UTXD2   | GPIO2_17 | 81     |
| 3      | GPIO0_03 |           | 37 | 38 |         | GPIO7_06 | 230    |
|        |          | GND       | 39 | 40 |         | GPIO7_05 | 229    |



40 pin 接口使用注意事项如下所示：

1) 40 pin 接口中总共有 **26** 个 GPIO 口，但 8 号和 10 号引脚默认是用于调试串口功能的，并且这两个引脚和 Micro USB 调试串口是连接在一起的，所以这两个引脚请不要设置为 GPIO 等功能。

2) 所有的 GPIO 口的电压都是 **3.3v**。

3) 40 pin 接口中 27 号和 28 号引脚只有 I2C 的功能，没有 GPIO 等其他复用功能，另外这两个引脚的电压默认都为 **1.8v**。

## 2. 20. wiringOP 的安装使用方法

### 2. 20. 1. 安装 wiringOP 的方法

1) 安装 wiringOP 前，请先确保 Linux 系统中存在 `/etc/orangepi-release` 这个配置文件，里面的内容为：**BOARD=orangepiaipro**。

```
(base) HwHiAiUser@orangepi:~$ cat /etc/orangepi-release
BOARD=orangepiaipro
```

2) 如果 Linux 中没有 `/etc/orangepi-release` 这个配置文件，可以使用下面的命令创建



一个。

```
(base) HwHiAiUser@orangepi:~$ echo "BOARD=orangepiaipro" | sudo tee /etc/orangepi-release
```

3) 下载 wiringOP 的代码。

```
(base) HwHiAiUser@orangepi:~$ sudo apt-get update
```

```
(base) HwHiAiUser@orangepi:~$ sudo apt-get install -y git
```

```
(base) HwHiAiUser@orangepi:~$ git clone https://github.com/orangepi-xunlong/wiringOP.git -b next
```

**注意，源码需要下载 wiringOP next 分支的代码，请别漏了 -b next 这个参数。**

4) 然后编译安装 wiringOP。

```
(base) HwHiAiUser@orangepi:~$ sudo apt-get install -y gcc make build-essential
```

```
(base) HwHiAiUser@orangepi:~$ cd wiringOP
```

```
(base) HwHiAiUser@orangepi:~/wiringOP$ sudo ./build clean
```

```
(base) HwHiAiUser@orangepi:~/wiringOP$ sudo ./build
```

5) 测试 gpio readall 命令的输出如下：

```
(base) HwHiAiUser@orangepi:~$ gpio readall
```

| AI PRO |     |          |      |   |          |    |      |          |     |      |  |
|--------|-----|----------|------|---|----------|----|------|----------|-----|------|--|
| GPIO   | wPi | Name     | Mode | V | Physical | V  | Mode | Name     | wPi | GPIO |  |
|        |     | 3.3V     |      |   | 1        | 2  |      | 5V       |     |      |  |
| 76     | 0   | SDA7     | OFF  | 0 | 3        | 4  |      | 5V       |     |      |  |
| 75     | 1   | SCL7     | OFF  | 0 | 5        | 6  |      | GND      |     |      |  |
| 226    | 2   | GPIO7_02 | OFF  | 0 | 7        | 8  | 0    | UTXD0    | 3   | 14   |  |
|        |     | GND      |      |   | 9        | 10 | 0    | URXD0    | 4   | 15   |  |
| 82     | 5   | GPIO2_18 | OFF  | 0 | 11       | 12 | 0    | GPIO7_03 | 6   | 227  |  |
| 38     | 7   | GPIO1_06 | IN   | 1 | 13       | 14 |      | GND      |     |      |  |
| 79     | 8   | GPIO2_15 | IN   | 1 | 15       | 16 | 1    | GPIO2_16 | 9   | 80   |  |
|        |     | 3.3V     |      |   | 17       | 18 | 1    | GPIO0_25 | 10  | 25   |  |
| 91     | 11  | SPI0_SD0 | OFF  | 0 | 19       | 20 |      | GND      |     |      |  |
| 92     | 12  | SPI0_SDI | OFF  | 0 | 21       | 22 | 1    | GPIO0_02 | 13  | 2    |  |
| 89     | 14  | SPI0_CLK | OFF  | 0 | 23       | 24 | 0    | SPI0_CS  | 15  | 90   |  |
|        |     | GND      |      |   | 25       | 26 | 1    | GPIO2_19 | 16  | 83   |  |
|        |     | SDA6     |      |   | 27       | 28 |      | SCL6     |     |      |  |
| 231    | 17  | URXD7    | OFF  | 0 | 29       | 30 |      | GND      |     |      |  |
| 84     | 18  | GPIO2_20 | IN   | 1 | 31       | 32 | 1    | PWM3     | 19  | 33   |  |
| 128    | 20  | GPIO4_00 | IN   | 1 | 33       | 34 |      | GND      |     |      |  |
| 228    | 21  | GPIO7_04 | OFF  | 0 | 35       | 36 | 0    | GPIO2_17 | 22  | 81   |  |
| 3      | 23  | GPIO0_03 | IN   | 1 | 37       | 38 | 1    | GPIO7_06 | 24  | 230  |  |
|        |     | GND      |      |   | 39       | 40 | 0    | GPIO7_05 | 25  | 229  |  |



## 2. 20. 2. 使用 wiringOP 控制 40pin GPIO 的方法

1) 下面以 7 号引脚——对应 GPIO 为 GPIO7\_02——对应 wPi 序号为 2——为例演示如何设置 GPIO 口的方向和高低电平。

```
(base) HwHiAiUser@orangePi:~/wiringOP$ gpio readall
```

| GPIO | wPi | Name     | Mode | V | Physical | V  | Mode | Name     | wPi | GPIO |
|------|-----|----------|------|---|----------|----|------|----------|-----|------|
|      |     | 3.3V     |      |   | 1        | 2  |      | 5V       |     |      |
| 76   | 0   | SDA7     | OFF  | 0 | 3        | 4  |      | 5V       |     |      |
| 75   | 1   | SCL7     | OFF  | 0 | 5        | 6  |      | GND      |     |      |
| 226  | 2   | GPIO7_02 | OFF  | 0 | 7        | 8  | 0    | UTXD0    | 3   | 14   |
|      |     | GND      |      |   | 9        | 10 | 0    | URXD0    | 4   | 15   |
| 82   | 5   | GPIO2_18 | OFF  | 0 | 11       | 12 | 0    | GPIO7_03 | 6   | 227  |
| 38   | 7   | GPIO1_06 | IN   | 1 | 13       | 14 |      | GND      |     |      |
| 79   | 8   | GPIO2_15 | IN   | 1 | 15       | 16 | 1    | GPIO2_16 | 9   | 80   |

2) 首先设置 GPIO 口为输出模式，其中第三个参数需要输入引脚对应的 wPi 的序号。

```
(base) HwHiAiUser@orangePi:~$ gpio mode 2 out
```

3) 然后使用下面的命令可以查看下 GPIO 当前的模式，可以看到当前的模式为输出——**OUT**。命令中第二个参数需要输入引脚对应的 wPi 的序号。

```
(base) HwHiAiUser@orangePi:~$ gpio qmode 2
OUT
```

4) 然后就可以开始设置引脚输出高低电平了。比如使用下面的命令可以设置 GPIO 口输出低电平，设置完后可以使用万用表测量下引脚的电压的数值，如果为 0v，说明设置低电平成功。

```
(base) HwHiAiUser@orangePi:~$ gpio write 2 0
```

5) 除了使用万用表测量引脚的电压外，还可以使用 **gpio read** 命令查看引脚的高低电平状态。当前命令输出为 0，说明引脚目前为低电平。

```
(base) HwHiAiUser@orangePi:~$ gpio read 2
0
```

6) 然后可以设置 GPIO 口输出高电平，设置完后可以使用万用表测量引脚的电压的数值，如果为 3.3v，说明设置高电平成功。

```
(base) HwHiAiUser@orangePi:~$ gpio write 2 1
```



7) 除了使用万用表测量引脚的电压外，还可以使用 **gpio read** 命令查看引脚的高低电平状态。当前命令输出为 1，说明引脚目前为高电平。

```
(base) HwHiAiUser@orangePi:~$ gpio read 2
1
```

8) 另外使用 **gpio readall** 命令也可以查看 GPIO 当前的设置情况。

```
(base) HwHiAiUser@orangePi:~$ gpio readall
```

| GPIO | wPi | Name     | Mode | V | AI | PRO | Physical | V   | Mode | Name     | wPi | GPIO |
|------|-----|----------|------|---|----|-----|----------|-----|------|----------|-----|------|
|      |     | 3.3V     |      |   | 1  | 2   |          |     |      | 5V       |     |      |
| 76   | 0   | SDA7     | OFF  | 0 | 3  | 4   |          |     |      | 5V       |     |      |
| 75   | 1   | SCL7     | OFF  | 0 | 5  | 6   |          |     |      | GND      |     |      |
| 226  | 2   | GPIO7_02 | OUT  | 1 | 7  | 8   | 0        | OFF |      | UTXD0    | 3   | 14   |
|      |     | GND      |      |   | 9  | 10  | 0        | OFF |      | URXD0    | 4   | 15   |
| 82   | 5   | GPIO2_18 | OFF  | 0 | 11 | 12  | 0        | OFF |      | GPIO7_03 | 6   | 227  |
| 38   | 7   | GPIO1_06 | IN   | 1 | 13 | 14  |          |     |      | GND      |     |      |
| 79   | 8   | GPIO2_15 | IN   | 1 | 15 | 16  | 1        | IN  |      | GPIO2_16 | 9   | 80   |

9) 使用下面的命令可以将 GPIO 口设置为输入模式，其中第三个参数需要输入引脚对应的 wPi 的序号。

```
(base) HwHiAiUser@orangePi:~$ gpio mode 2 in
```

10) 将 GPIO 口设置为输入模式后，当将 GPIO 口用杜邦线连接到 GND 引脚后，使用 **gpio read** 命令可以看到 GPIO 口的输入值会为 0。

```
(base) HwHiAiUser@orangePi:~$ gpio read 2
0
```

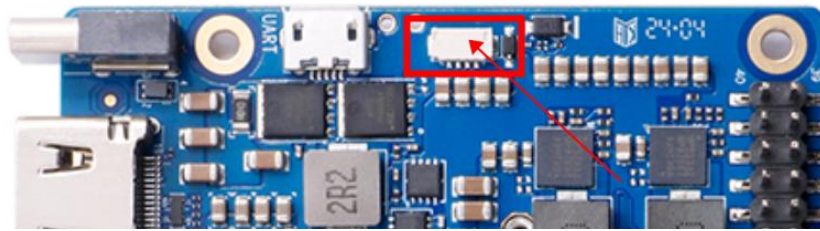
11) 将 GPIO 口设置为输入模式后，当将 GPIO 口用杜邦线连接到 3.3v 引脚后，使用 **gpio read** 命令可以看到 GPIO 口的输入值会为 1。

```
(base) HwHiAiUser@orangePi:~$ gpio read 2
1
```

12) 其他引脚的设置方法类似，只需修改 wPi 的序号为引脚对应的序号即可。

## 2.21. 散热风扇的使用方法

开发板散热风扇的接口所在的位置如下所示：



开发板使用的散热风扇为 12V 的，接口为 4pin，0.8mm 间距规格。可以通过 PWM 来控制风扇的转速。

使用 `npu-smi` 命令可以查询和控制 PWM 风扇，详细用法如下所示：

1) 查询风扇模式的命令如下所示：

| <b>(base) HwHiAiUser@orangepiaipro:~\$ sudo npu-smi info -t pwm-mode</b> |                                                                                                                   |
|--------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------|
| <b>pwm-mode</b>                                                          | <b>: auto</b>                                                                                                     |
| 字段                                                                       | 说明                                                                                                                |
| pwm-mode                                                                 | 风扇模式。<br>有如下两种模式： <ul style="list-style-type: none"> <li>• manual: 手动模式</li> <li>• auto: 自动模式</li> </ul> 默认为自动模式。 |

2) 查询风扇调速比的命令如下所示：

|                                                                                |             |
|--------------------------------------------------------------------------------|-------------|
| <b>(base) HwHiAiUser@orangepiaipro:~\$ sudo npu-smi info -t pwm-duty-ratio</b> |             |
| <b>pwm-duty-ratio(%)</b>                                                       | <b>: 15</b> |

3) 设置风扇模式为手动模式的命令如下所示：

| <b>(base) HwHiAiUser@orangepiaipro:~\$ sudo npu-smi set -t pwm-mode -d 0</b>   |  |
|--------------------------------------------------------------------------------|--|
| 描述                                                                             |  |
| 风扇使能模式。分为手动模式、自动模式。默认为自动模式。                                                    |  |
| <ul style="list-style-type: none"> <li>• 0: 手动模式</li> <li>• 1: 自动模式</li> </ul> |  |



4) 将风扇设置为手动模式后就可以通过下面的命令来设置风扇的调速比了。比如下面的命令会将风扇调速比设置为 100，设置完后风扇会用最大的转速运行。

```
(base) HwHiAiUser@orangepiaipro:~$ sudo npu-smi set -t pwm-duty-ratio -d 100
```

| 描述            |
|---------------|
| 风扇调速比         |
| 取值范围: [0-100] |

## 2.22. AI CPU 和 control CPU 的设置方法

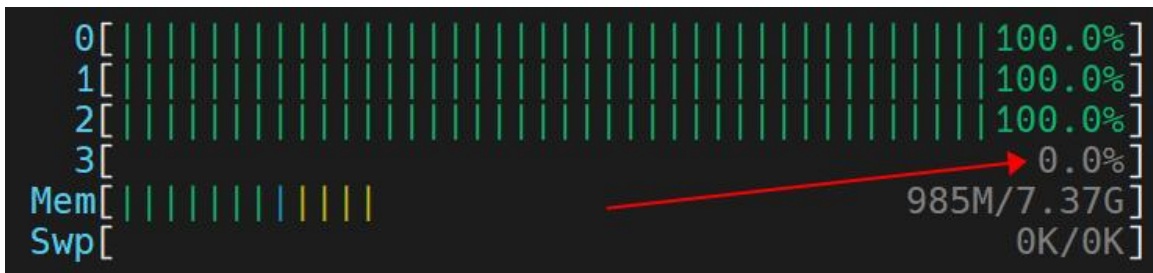
开发板使用的昇腾 SOC 总共有 4 个 CPU，这 4 个 CPU 既可以设置为 control CPU，也可以设置为 AI CPU。默认情况下，control CPU 和 AI CPU 的分配数量为 3:1。使用 **npu-smi info** 命令可以查看下 control CPU 和 AI CPU 的分配数量。

```
(base) HwHiAiUser@orangepiaipro:~$ npu-smi info -t cpu-num-cfg -i 0 -c 0
```

```
Current AI CPU number      : 1
Current control CPU number : 3
Current data CPU number    : 0
```

```
(base) HwHiAiUser@orangepiaipro:~$
```

当 Linux 系统跑满后，使用 **htop** 命令会看到有一个 CPU 的占用率始终接近 0，请注意，这是正常的。因为这个 CPU 默认用于 AI CPU。



如果当前环境模型中无 AI CPU 算子，且运行业务时查询 AI CPU 占用率持续为 0，则可以将 AI CPU 的数量配置为 0。查询 AI CPU 占用率的命令如下所示：

```
(base) HwHiAiUser@orangepiaipro:~$ npu-smi info -t usages -i 0 -c 0
```

```
Memory Capacity(MB)      : 7545
Memory Usage Rate(%)     : 20
Hugepages Total(page)    : 15
Hugepages Usage Rate(%)  : 100
```



```
Aicore Usage Rate(%)      : 0
Aicpu Usage Rate(%)       : 0
Ctrlcpu Usage Rate(%)     : 1
Memory Bandwidth Usage Rate(%) : 1
```

如果不需要使用 AI CPU，使用下面的命令可以将 4 个 CPU 都设置为 control CPU。设置完后需要重启系统让配置生效。

```
(base) HwHiAiUser@orangepiaipro:~$ sudo npu-smi set -t cpu-num-cfg -i 0 -c 0 -v 0:4:0
Status      : OK
Message : The cpu-num-cfg of the chip is set successfully. Reset system for the configuration to take effect.
```

当 4 个 CPU 都设置为 control CPU 后，再运行任务让所有 CPU 跑满，使用 htop 命令就能看到 4 个 CPU 的占用率都能达到 100%了。

```
0[|||||||||||||||||||||||||||||||||||||100.0%]
1[|||||||||||||||||||||||||||||||||||||100.0%]
2[|||||||||||||||||||||||||||||||||||||100.0%]
3[|||||||||||||||||||||||||||||||||||||100.0%]
Mem[|||||||||||||||||913M/7.37G]
Swp[|||||0K/0K]
```

## 2.23. 设置 Swap 内存的方法

虽然开发板有 8GB 或 16GB 的大内存，但有些应用需要的内存大于 8GB 或 16GB，此时我们可以通过 Swap 内存来扩展系统能使用的最大内存容量。方法如下所示：

1) 首先创建一个 swap 文件，下面的命令会创建一个 16GB 大小的 swap 文件，容量大小请根据自己的需求进行修改。

```
(base) HwHiAiUser@orangepiaipro:~$ sudo fallocate -l 16G /swapfile
```

如果已经启用了Swap分区再运行fallocate 命令会报下面的错误：

```
fallocate: fallocate failed: Text file busy
```

需要先运行sudo swapoff /swapfile命令关闭系统上的swap分区才行。

注意，添加Swap内存前，请确保TF卡、eMMC或者SSD的剩余空间大于需要添加的Swap内存容量。



2) 然后修改文件权限，确保只有 root 用户可以读写。

```
(base) HwHiAiUser@orangepiaipro:~$ sudo chmod 600 /swapfile
```

3) 然后把这个文件设置成 swap 空间。

```
(base) HwHiAiUser@orangepiaipro:~$ sudo mkswap /swapfile
```

4) 然后启用 swap。

```
(base) HwHiAiUser@orangepiaipro:~$ sudo swapon /swapfile
```

5) 完成以上步骤后，通过下面的命令可以检查 swap 内存是否已经添加成功。

```
(base) HwHiAiUser@orangepiaipro:~$ free -h
```

|       | total       | used  | free  | shared | buff/cache | available |
|-------|-------------|-------|-------|--------|------------|-----------|
| Mem:  | 7.4Gi       | 1.1Gi | 5.5Gi | 27Mi   | 835Mi      | 6.1Gi     |
| Swap: | <b>15Gi</b> | 0B    | 15Gi  |        |            |           |

6) 如果需要 swap 设置在重启之后依然有效，请运行下面命令将对应的配置添加到 `/etc/fstab` 文件中。

```
(base) HwHiAiUser@orangepiaipro:~$ echo '/swapfile none swap sw 0 0' | sudo tee -a /etc/fstab
```

## 2.24. 测试 MindSpore 的方法

开发板的 Ubuntu 系统中已经预装了 MindSpore，使用下面的方法可以测试下 MindSpore 是否能正常使用。

1) 首先请根据[设置 Swap 内存的方法](#)一小节的说明给系统额外添加 16GB 的 Swap 内存。

2) 然后在 **HwHiAiUser** 用户中执行下面的命令：

```
(base) HwHiAiUser@orangepiaipro:~$ python -c \
"import mindspore;mindspore.set_context(device_target='Ascend');mindspore.run_check()"
```

3) 等待一段时间后，如果能看到下面的输出就说明 MindSpore 能正常使用。

```
MindSpore version: 2.x.xx.2024xxxx
The result of multiplication calculation is correct, MindSpore has been installed on platform
[Ascend] successfully!
```

4) 更新 MindSpore 版本的命令如下所示：

```
(base) HwHiAiUser@orangepiaipro:~$ pip install --upgrade mindspore
```

## 2.25. 使用 ascend 硬件加速的 ffmpeg

FFmpeg 是一个开源项目，它提供了一套用于处理音频和视频内容的库和程序，可以实现音视频的录制、转换、流处理等功能。FFmpeg 支持多种媒体格式，几乎可以处理所有常见的音视频数据。

我们通过使用由 DVPP 接口实现的 FFmpeg 硬件编解码器可以大幅提高视频编码和解码的速度。通过将视频处理任务卸载到 Ascend 芯片上的专用硬件，可以显著减少 CPU 的负载，加速视频、图片数据的处理过程。

### 2.25.1. 使用编译好的 deb 软件包

1) 支持 Ascend 硬件编解码器的 deb 格式的软件包可以从开发板的资料下载页面下载到。步骤为：

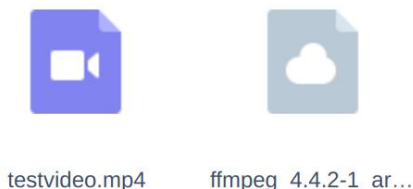
a. 打开下面的链接：

<http://www.orangepi.cn/html/hardWare/computerAndMicrocontrollers/service-and-support/Orange-Pi-AIpro.html>

b. 然后选择官方工具。



c. 然后下载 FFmpeg\_ascend 文件夹中的 **ffmpeg\_4.4.2-1\_arm64.deb** 软件包和测试视频。



2) 然后参考手册[上传文件到开发板 Linux 系统中的方法](#)将其上传到开发板的以下目录。

/home/HwHiAiUser/



3) 首先在”/home/HwHiAiUser/.bashrc”文件的最后添加下面两个环境变量，并使其生效。

```
HwHiAiUser@orangepi:~$ vim .bashrc
source /usr/local/Ascend/ascend-toolkit/set_env.sh
export LD_LIBRARY_PATH=/usr/lib:/usr/local/Ascend/ascend-toolkit/latest/acclib/lib64:
$LD_LIBRARY_PATH
HwHiAiUser@orangepi:~$ source .bashrc
```

4) 在开发板上执行以下命令安装软件包。

```
HwHiAiUser@orangepi:~$ ls
ffmpeg_4.4.2-1_arm64.deb
HwHiAiUser@orangepi:~$ sudo dpkg -i ffmpeg_4.4.2-1_arm64.deb
(Reading database ... 129897 files and directories currently installed.)
Preparing to unpack ffmpeg_4.4.2-1_arm64.deb ...
Unpacking ffmpeg (4.4.2-1) over (4.4.2-1) ...
Setting up ffmpeg (4.4.2-1) ...
Processing triggers for man-db (2.10.2-1) ...
```

5) 验证安装。依次输入以下 3 条命令，如果在输出的最后存在下面列出的内容，则说明编解码器安装成功。

```
HwHiAiUser@orangepi:~$ ffmpeg -hwaccels | grep ascend
ascend
HwHiAiUser@orangepi:~$ ffmpeg -encoders | grep ascend
V..... h264_ascend      Ascend HiMpi H264 encoder (codec h264)
V..... h265_ascend      Ascend HiMpi H265 encoder (codec hevc)
HwHiAiUser@orangepi:~$ ffmpeg -decoders | grep ascend
V..... h264_ascend      Ascend HiMpi H264 decoder (codec h264)
V..... h265_ascend      Ascend HiMpi H265 decoder (codec hevc)
```

## 2. 25. 2. 从源代码构建

1) 编解码器补丁包可以从开发板的资料下载页面下载到。步骤为：

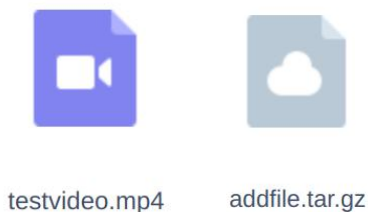
a. 打开下面的链接：

<http://www.orangepi.cn/html/hardWare/computerAndMicrocontrollers/service-and-support/Orange-Pi-AIpro.html>

b. 然后选择官方工具。



c. 然后下载 FFmpeg\_ascend 文件夹中的 **addfile.tar.gz** 和测试视频。



2) 然后参考手册[上传文件到开发板 Linux 系统中的方法](#)将其上传到开发板的以下目录。

```
/home/HwHiAiUser/
```

3) 接下来下载 ffmpeg 源码。首先打开 `/etc/apt/sources.list` 配置文件，然后取消掉 **deb-src** 前面的注释，再使用 **apt update** 命令更新软件包列表的缓存，再使用 **apt source ffmpeg** 命令下载 ffmpeg 的源码。

```
HwHiAiUser@orangePi:~$ sudo vim /etc/apt/sources.list
deb http://repo.huaweicloud.com/ubuntu-ports/ jammy main restricted universe multivers
e
deb http://repo.huaweicloud.com/ubuntu-ports/ jammy-security main restricted universe
multiverse
deb http://repo.huaweicloud.com/ubuntu-ports/ jammy-updates main restricted universe
multiverse
deb http://repo.huaweicloud.com/ubuntu-ports/ jammy-proposed main restricted universe
multiverse
deb-src http://repo.huaweicloud.com/ubuntu-ports/ jammy main restricted universe multi
verse
deb-src http://repo.huaweicloud.com/ubuntu-ports/ jammy-security main restricted univer
```



```
se multiverse
deb-src http://repo.huaweicloud.com/ubuntu-ports/ jammy-updates main restricted univer
se multiverse
deb-src http://repo.huaweicloud.com/ubuntu-ports/ jammy-proposed main restricted univ
erse multiverse
HwHiAiUser@orangePi:~$ sudo apt update
HwHiAiUser@orangePi:~$ sudo apt-get install dpkg-dev
HwHiAiUser@orangePi:~$ apt source ffmpeg
```

4) 运行 **apt source ffmpeg** 命令后，当前目录会多出下面所示的文件和文件夹，其中 **ffmpeg-4.4.2** 文件夹为 ffmpeg 源码的目录。

```
HwHiAiUser@orangePi:~$ ls | grep ffmpeg
ffmpeg-4.4.2
ffmpeg_4.4.2-0ubuntu0.22.04.1.debian.tar.xz
ffmpeg_4.4.2-0ubuntu0.22.04.1.dsc
ffmpeg_4.4.2.ascend-1_arm64.deb
ffmpeg_4.4.2.orig.tar.xz
ffmpeg_4.4.2.orig.tar.xz.asc
```

5) 然后将 ffmpeg dvpp 的补丁文件解压并复制到 ffmpeg 源码的文件夹中。

```
HwHiAiUser@orangePi:~$ tar xzf addfile.tar.gz
HwHiAiUser@orangePi:~$ cp -r addfile/* ./ffmpeg-4.4.2
```

6) 然后在 `"/home/HwHiAiUser/.bashrc"` 文件的最后添加下面两个环境变量，并使其生效。

```
HwHiAiUser@orangePi:~$ vim .bashrc
source /usr/local/Ascend/ascend-toolkit/set_env.sh
export LD_LIBRARY_PATH=/usr/lib:/usr/local/Ascend/ascend-toolkit/latest/acclib/lib64:
$LD_LIBRARY_PATH
HwHiAiUser@orangePi:~$ source .bashrc
```

7) 然后就可以开始编译安装带有 dvpp 功能的 ffmpeg 包。具体命令如下所示：

```
HwHiAiUser@orangePi:~$ cd ffmpeg-4.4.2
HwHiAiUser@orangePi:~/ffmpeg-4.4.2$ ./configure \
--prefix=/usr \
```



```
--enable-shared \
--extra-cflags="-I${ASCEND_HOME_PATH}/acllib/include" \
--extra-ldflags="-L${ASCEND_HOME_PATH}/aarch64-linux/lib64" \
--extra-libs="-lacl_dvpp_mpi -lascendcl" \
--enable-ascend
```

```
HwHiAiUser@orangePi:~/ffmpeg-4.4.2$ make -j4
```

```
HwHiAiUser@orangePi:~/ffmpeg-4.4.2$ sudo make install
```

注意，这里./configure后所列出的配置是最简必要配置，如果需要添加其他配置，请自行按照格式添加，如--enable-libx264。

6) 验证安装。依次输入以下 3 条命令，如果在输出的最后几行存在下面列出的内容，则说明编解码器安装成功。

```
HwHiAiUser@orangePi:~$ ffmpeg -hwaccels | grep ascend
```

```
ascend
```

```
HwHiAiUser@orangePi:~$ ffmpeg -encoders | grep ascend
```

```
V..... h264_ascend      Ascend HiMpi H264 encoder (codec h264)
```

```
V..... h265_ascend      Ascend HiMpi H265 encoder (codec hevc)
```

```
HwHiAiUser@orangePi:~$ ffmpeg -decoders | grep ascend
```

```
V..... h264_ascend      Ascend HiMpi H264 decoder (codec h264)
```

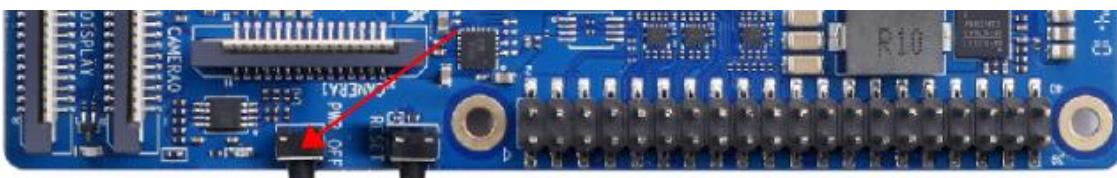
```
V..... h265_ascend      Ascend HiMpi H265 decoder (codec hevc)
```

## 2. 26. 关机和重启开发板的方法

1) 在 Linux 系统运行的过程中，如果直接拔掉电源断电，可能会导致文件系统丢失某些数据，建议断电前先使用 **poweroff** 命令关闭开发板的 Linux 系统，然后再拔掉电源。

```
(base) HwHiAiUser@orangePiapro:~$ sudo poweroff
```

2) 除了 **poweroff** 命令可以关闭 Linux 系统外，还可以使用开发板上的关机按键来关闭开发板的 Linux 系统，然后再拔掉电源。请注意，关机按键是没有开机功能的。





3) 使用 **reboot** 命令即可重启开发板中的 Linux 系统。

```
(base) HwHiAiUser@orangepiaipro:~$ sudo reboot
```