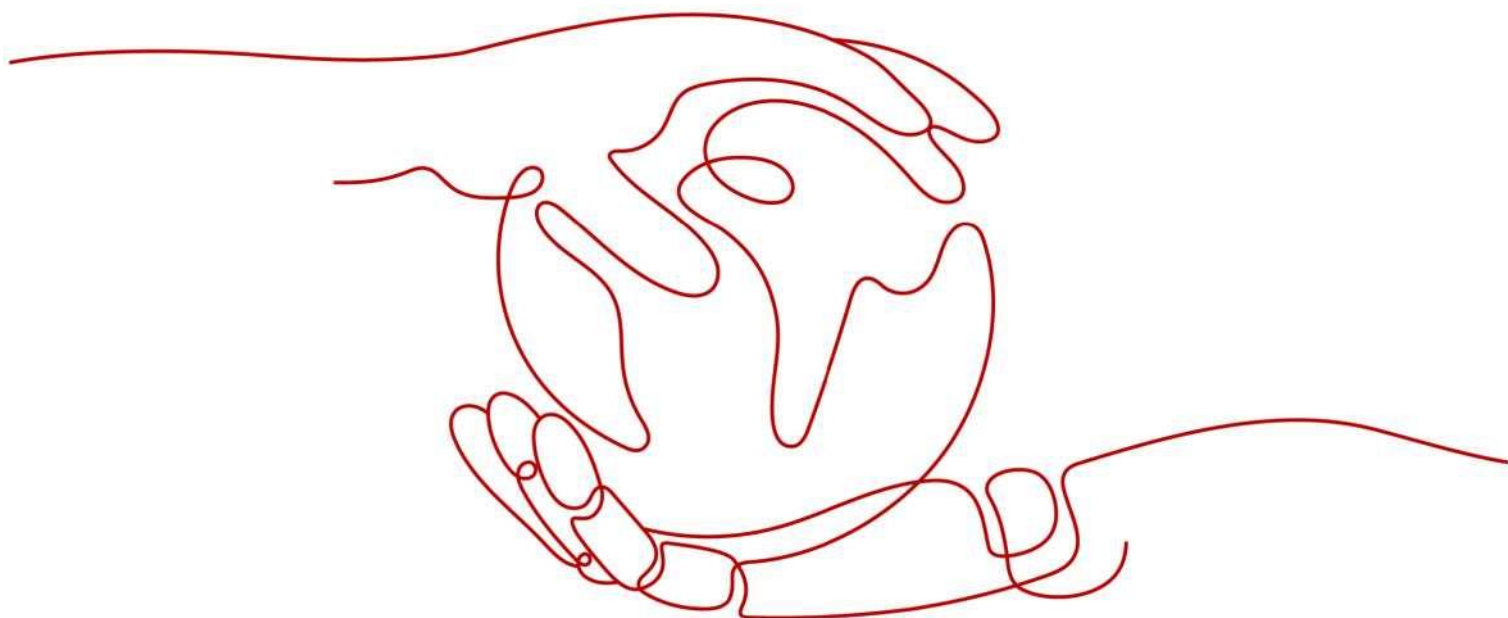


Atlas-PI A2 开发者套件 23.0.RC3

快速开始

文档版本 02

发布日期:2023-12-29



QUANAI

版权所有全爱科技（上海）有限公司 2021。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明



QUANAI 和其他全爱商标均为全爱科技（上海）有限公司的商标。
本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受全爱科技商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，全爱公司对本文档内容不做任何明示或默示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

全爱科技（上海）有限公司

地址：上海市闵行区剑川 930 号 D 栋 3 层邮编：

200240 网址：www.quanaichina.com

电话：021-64025956

目 录

快速开始.....	1
1 准备硬件.....	1
2 准备软件.....	3
3 一键制卡.....	3
4 硬件连接.....	9
5 设置网口 IP 地址	19
6 远程登录开发者套件.....	23
7 接口测试.....	26
8 运行基础样例.....	27

1 准备硬件

运行开发者套件所需的相关硬件，如表 1-1 所示。

表 1-1 相关硬件

硬件	是否需要额外购买	说明
开发者套件	否	开箱后的开发者套件包括套件主板和电源。
SD 卡	是	用于装载镜像运行开发者套件。制卡工具烧录镜像到 SD 卡时会格式化 SD 卡，建议准备一个开发者套件专用的 SD 卡。 SD 卡规格要求：需配套使用 64G 容量及以上的 SD 卡，推荐 128G。
读卡器	是	用于插入 SD 卡烧录镜像。
RJ45 网线	是	用于连接开发者套件和 PC 的以太网口。 开发者套件也支持通过串口或 Type-C 接口实现远程登录，如果现场需要通过串口或 Type-C 接口登录开发者套件，可以参见串口登录或使用 Type-C 接口登录章节。
PC（笔记本或台式）	是	用于安装制卡工具、烧录镜像和远程连接开发者套件。 PC 配置要求如下。 •操作系统：Windows10、Windows11。 说明 当 PC 操作系统为 Windows11 时，制卡工具的配置网络和备份镜像功能不支持使用。 •具备 USB 或 Type-C 接口，且读写功能正常。 •确保 C 盘剩余空间充足（大于 10G），否则将不会对下载文件进行缓存，再次烧录镜像时需要再次从网络下载镜像。

如图 1-1 所示为开发者套件最简配置的相关硬件。

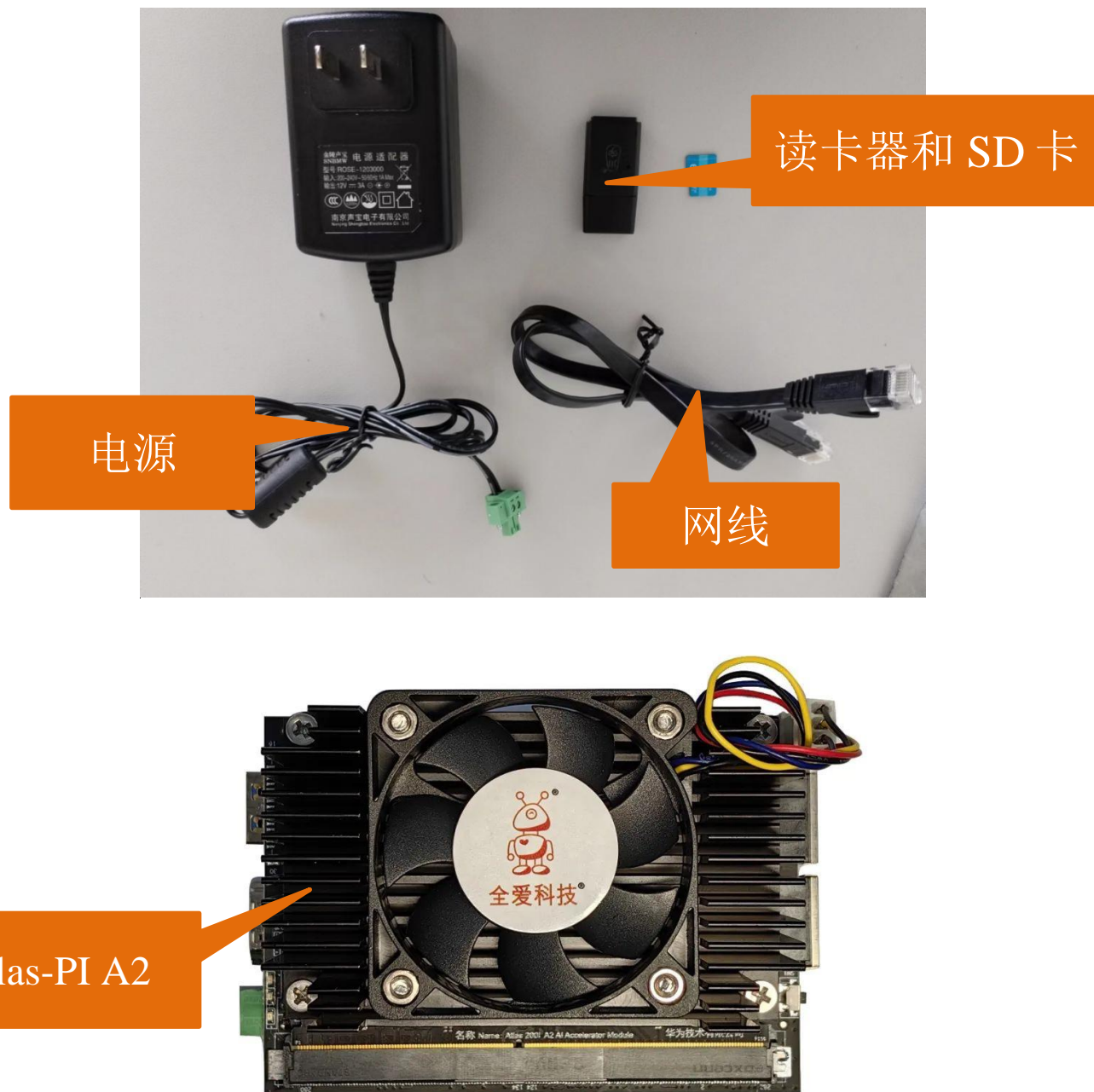


图 1-1 相关硬件

2 准备软件

启动开发者套件前，需要准备如表 2-1 所示软件资源。

表 2-1 相关软件

软件	说明	下载链接
制卡工具	全称 ascend-ai-devkit-imager，用于烧录镜像到 SD 卡。	单击 下载链接 获取制卡工具“Ascend-devkit- image'r_(version)_w\n-x86_64.exe”。
SSH 工具	用户在 PC 端远程登录开发者套件的操作系统进行命令行操作。 本文以 MobaXterm 为例，如果用户已部署 MobaXterm 或其他 SSH 工具，可不用下载。	单击 下载链接 获取 MobaXterm 软件压缩包，解压获得“MobaXterm_Personal_222.exe”。
镜像（可选）	镜像包含 OS、NPU 驱动固件、CANN、代码样例。 制卡过程中，获取镜像有两种方式，选择本地制卡方式时需要提前下载好镜像。 在线制卡：提供 Ubuntu22.04 操作系统的镜像，制卡工具自动通过网络获取镜像烧录到 SD 卡，无需提前下载。镜像版本会迭代更新，用户在烧录镜像时可选择最新版本进行烧录。 本地制卡：提前下载镜像，再使用制卡工具将镜像烧录到 SD 卡。适用于制卡的 PC 无法连接外部网络的场景。	单击 下载链接 下载镜像“ubuntu22.04RC3Image.img.zip”。（注：该镜像使用需经过升级步骤，具体参考 全爱镜像使用 ）

说明

-{version}为工具或镜像版本号，下载链接中制卡工具与镜像为最新版本。

-MobaXterm 为第三方工具，下载链接和版本会跟随第三方的发布定期更新，可以在下载页面下载最新版本。

3 一键制卡

制卡前准备

- 将 Micro SD 卡插入读卡器，并将读卡器插入 PC。
- 如果用户已烧录老版本镜像并在开发者套件运行，SD 卡中已存在有价值的数据或应用文件，现在需要烧录新版本镜像，可以将数据或文件下载到 PC，待烧录新版本镜像的 SD 卡在开发者套件启动运行后，再将数据和文件上传到开发者套件。

下载制卡工具

步骤 1 在 PC 上双击制卡工具安装包“”，系统可能会弹出阻止应用启动的提示，如图 3-1 所示，单击“更多信息”。



图 3-1 阻止应用启动

单击“仍要运行”，在弹出的提示（您要允许来自未知发布者的此应用对你的设备进行更改吗）页面选择“是”。



图 3-2 运行应用

步骤 2 在安装导向界面按照默认配置快速安装工具。

步骤 3 运行一键制卡工具



图 3-3 一键制卡工具界面图

选择和烧录镜像

步骤 1 选择制卡方式。

在线制卡（推荐）：选择 OpenEuler22.03 操作系统的镜像，制卡工具自动通过网络获取镜像烧录到 SD 卡，无需提前下载。镜像版本会迭代更新，用户在烧录镜像时可选择最新版本进行烧录。

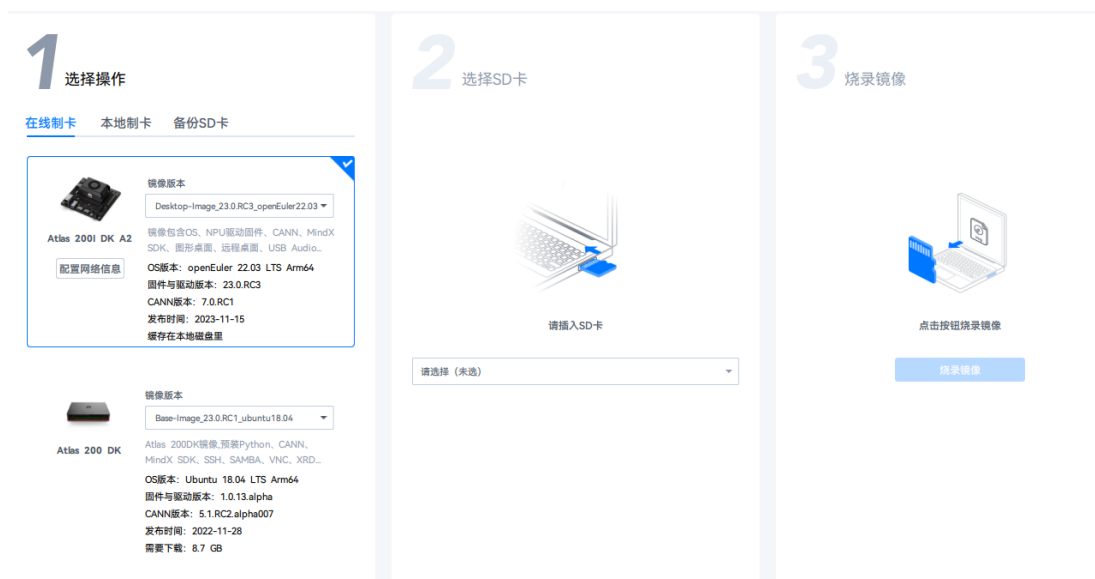


图 3-4 在线制卡界面

说明:

使用在线制卡方式烧录镜像时，需要确保 PC 能持续访问网络，避免以下类似问题导致

网络断开。

- 因 PC 锁屏导致网络断开。
- 杀毒软件禁止联网导致网络断开。
- 网络带宽速率建议使用 300Mbps 以上。
- 制卡工具使用固定的镜像链接，其使用的 https 协议涉及端口号为 443。

步骤 2 选择“在线制卡”方式时，工具支持提前配置开发者套件接口的网络信息，按钮如图 3-5 所示。



图 3-5 选择按钮

（重点阅读）如果不修改网络信息，则镜像烧录完成后，开发者套件 eth1 网口默认静态 IP 地址为 192.168.137.100；eth0 网口为 DHCP 动态模式，未分配 IP 地址；Type-C 接口默认静态 IP 地址为 192.168.0.2。

按照默认配置 eth1 网口和 Type-C 接口可用于固定网络环境，eth0 网口可通过路由器动态分配 IP 用于连接外部网络，建议保留 eth0 网口为 DHCP 动态模式，用于后

续连接路由器。建议保留 eth1 网口 IP 地址的网段为 192.168.137.xxx，后续可以配置通过 PC 共享外部网络给开发者套件。

本文以保留 3 个接口的默认 IP 信息为例进行后续操作，接口 IP 如下图所示：

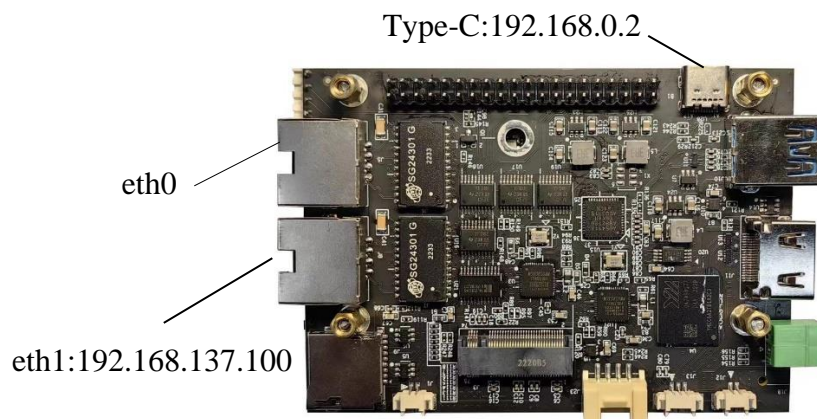


图 3-6 开发者套件各接口默认 IP 地址

本地制卡：本地制卡功能需和备份 SD 卡功能（备份镜像的操作请参见《[备份镜像](#)》）配合使用，将 SD 卡中的镜像备份到 PC，选择“本地制卡”页签，单击“选择文件”，选择镜像烧文件录到新的 SD 卡，如图 3-7 所示。



图 3-7 本地制卡

步骤 3 选择 SD 卡。

烧录镜像时会自动将 SD 卡格式化，用户需要提前检查 SD 卡是否有数据需要备份。



图 3-8 选择 SD 卡

步骤 4 单击“烧录”按钮，开始烧录，工具会预估完成烧录所需时间。



图 3-9 烧录镜像

在等待烧录时可同步执行 5 设置 PC 网口 IP 地址以缩短启动开发者套件前的准备时间。

步骤 5 烧录成功后，会弹出“烧录成功”提示窗，根据提示单击“继续”，并将 SD 卡从读卡器中取出。

----结束

4 硬件连接

本节主要介绍开发者套件连线和插入 SD 卡的操作步骤。

步骤 1 将 SD 卡插入开发者套件的 SD 插槽，并确保完全推入插槽底部。

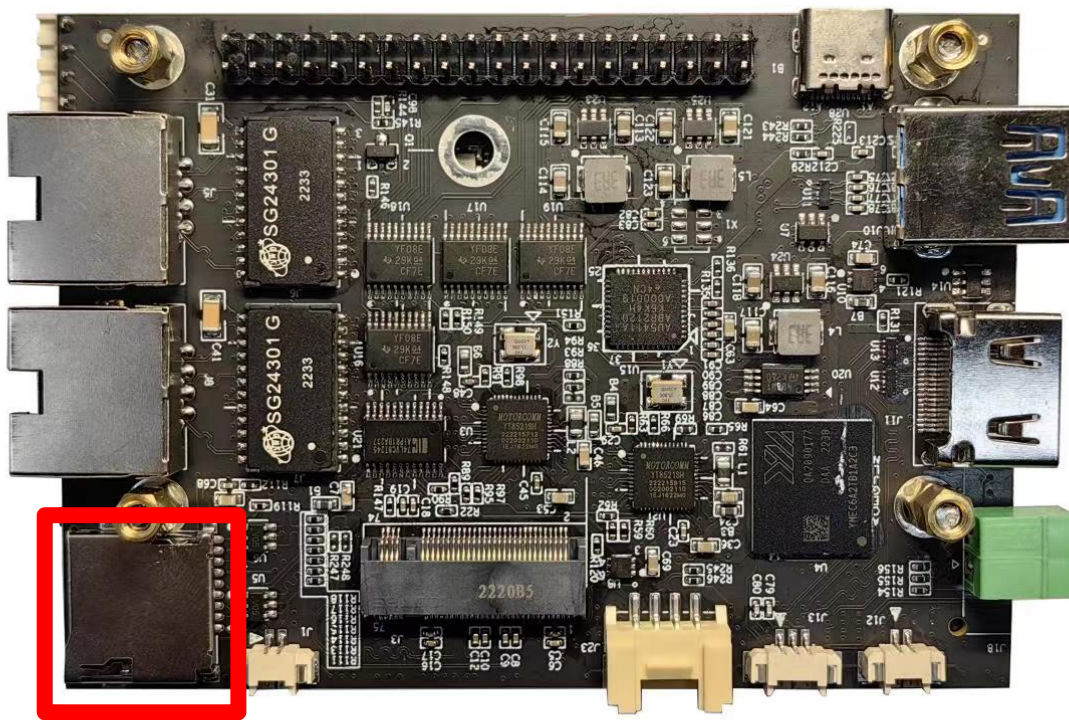


图 4-1 插入 SD 卡

步骤 2 连接方式分为两种，第一种使用开发者套件串口连接 PC 机串口；第二种网线连接开发者套件 eth1 网口和 PC 网口。

表格 1 40Pin IO 接口定义

序号	信号名称	模组管脚定义	序号	信号名称	
1	VCC_3V3D	电源	2	VCC_5V0D	电源
3	I2C7_SCL	Pin_2	4	VCC_5V0D	电源
5	I2C7_SDA	Pin_320	6	DGND	电源
7	I2S0_MCLK	Pin_263	8	UART0_TX	Debug 接口
9	DGND	电源	10	UART0_RX	Debug 接口
11	GPIO17_IO	Pin_147	12	I2S0_BCLK	Pin_261
13	GPIO27_IO	Pin_191	14	DGND	电源
15	GPIO22_IO	Pin_20	16	GPIO23_IO	Pin_18
17	VCC_3V3D	电源	18	GPIO24_IO	Pin_282
19	SPI0_MOSI	Pin_187	20	DGND	电源
21	SPI0_MISO	Pin_185	22	GPIO25_IO	Pin_292
23	SPI0_CLK	Pin_189	24	SPI0_CS0	Pin_193
25	DGND	电源	26	UART2_RTX	Pin_145
27	I2C6_SDA	Pin_4	28	I2C6_SCL	Pin_6
29	I2S1_MCLK	Pin_251	30	DGND	电源
31	UART2_CTX	Pin_143	32	PWM1_IO	Pin_286
33	GPIO13_IO	Pin_314	34	DGND	电源
35	I2S0_LRCK	Pin_259	36	GPIO16_IO	Pin_149
37	GPIO26_IO	Pin_290	38	I2S0_MISO	Pin_255
39	DGND	电源	40	I2S0_MOSI	Pin_257

- 注：1、所有 IO 均为 3.3V，GPIO 均是双向 IO，可设置成输入或输出；
2、所有 IO 均可以作为 GPIO，做 GPIO 时与树莓派接口定义兼容。

使用开发者套件调试 UART 引脚如下：

6: DGND

8: UART0_TX

10: UART0_RX

PC 端使用 USB 转 UART（TTL 电平型号）连接开发者套件串口 IO，接线方式如下：

开发者套件 **6: DGND** <--> USB 转 UART **GND**

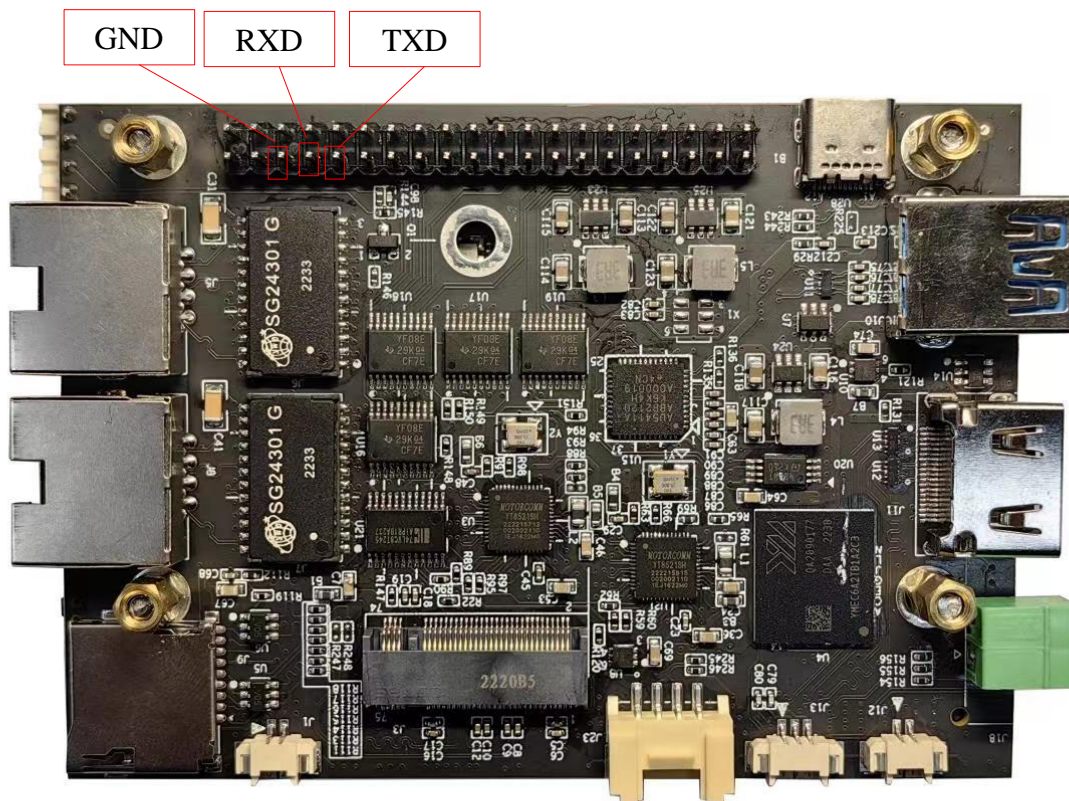
开发者套件 8: **UART0_TXD** <--> USB 转 UART RXD

开发者套件 10: **UART0_RXD** <--> USB 转 UART TXD

USB 转 UART 驱动根据型号厂家会提供驱动，或者使用驱动精灵安装对应驱动。



图 4-2 USB 转串口



驱动安装完成后设备管理器会有串口设备出现，如下图所示：

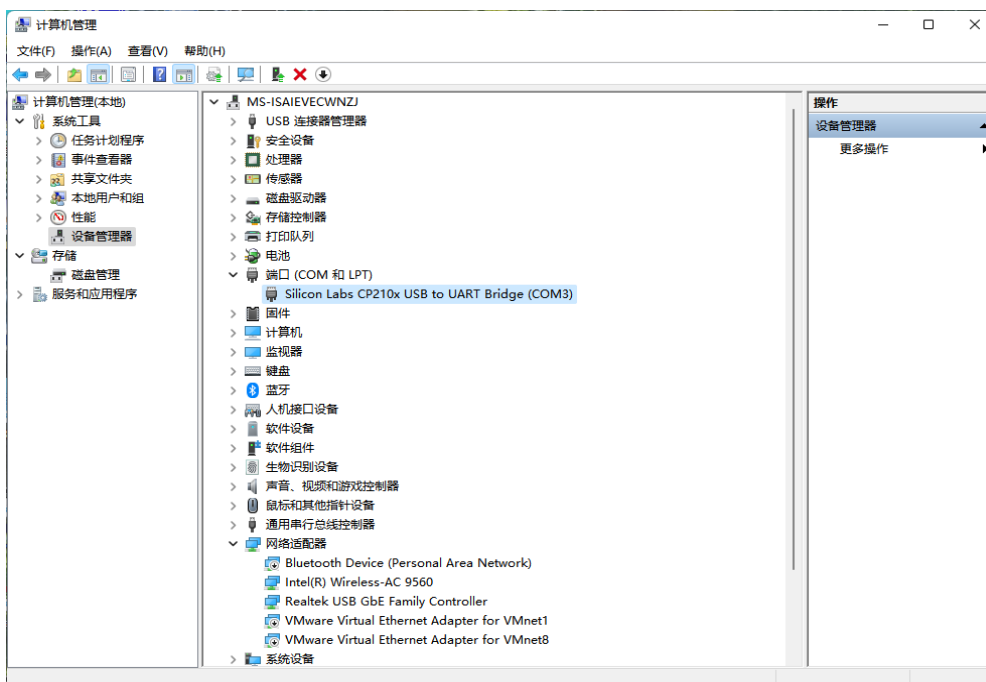


图 4-3 USB 转串口驱动示意

打开 MobaXterm 软件，点击 New session

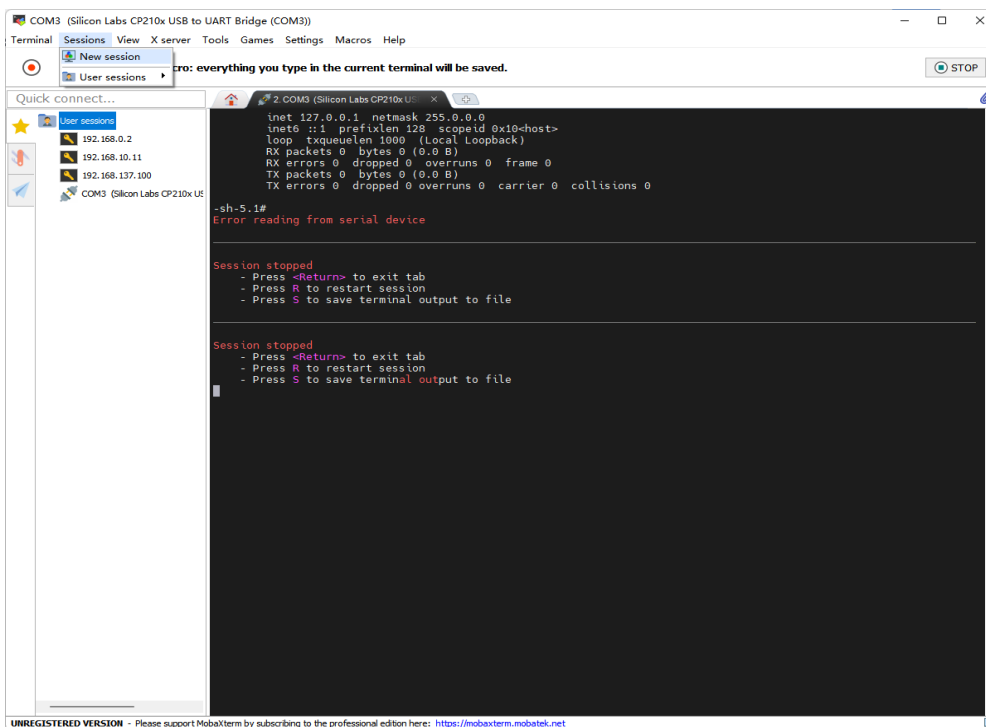


图 4-4

点击 Serial

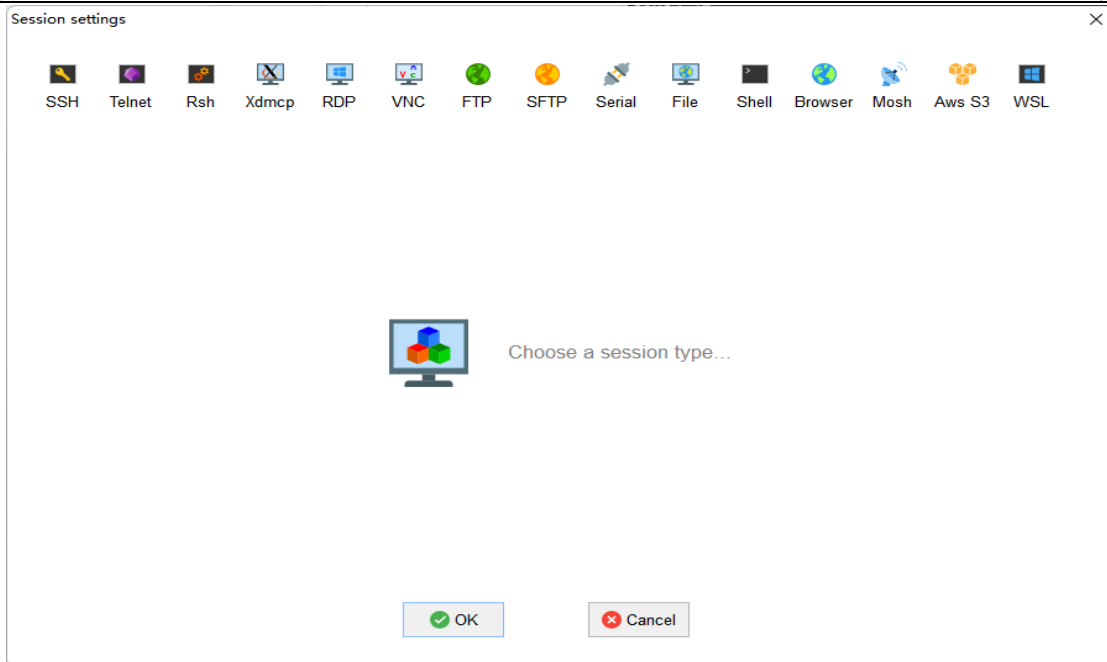


图 4-5

根据串口号创建调试连接

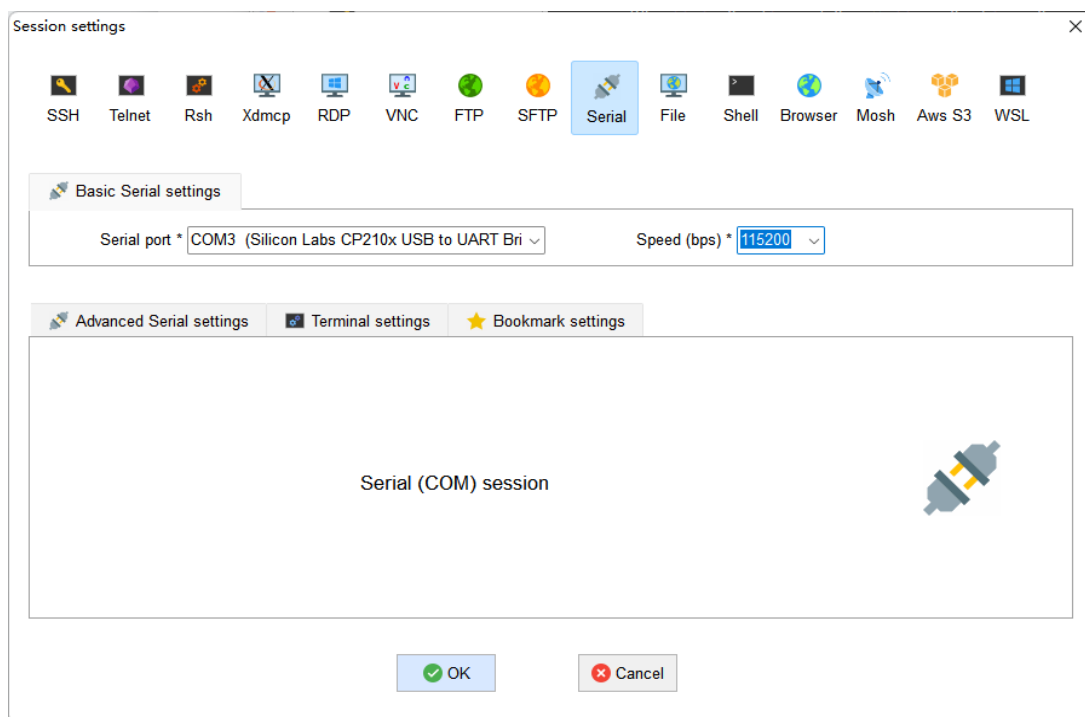


图 4-6

波特率选择 115200，点击 ok。

步骤 3 将电源线插头插入插座，接通开发者套件电源启动开发者套件。

- 串口有启动 log 显示，表示连线正常，如图 4-7 所示。

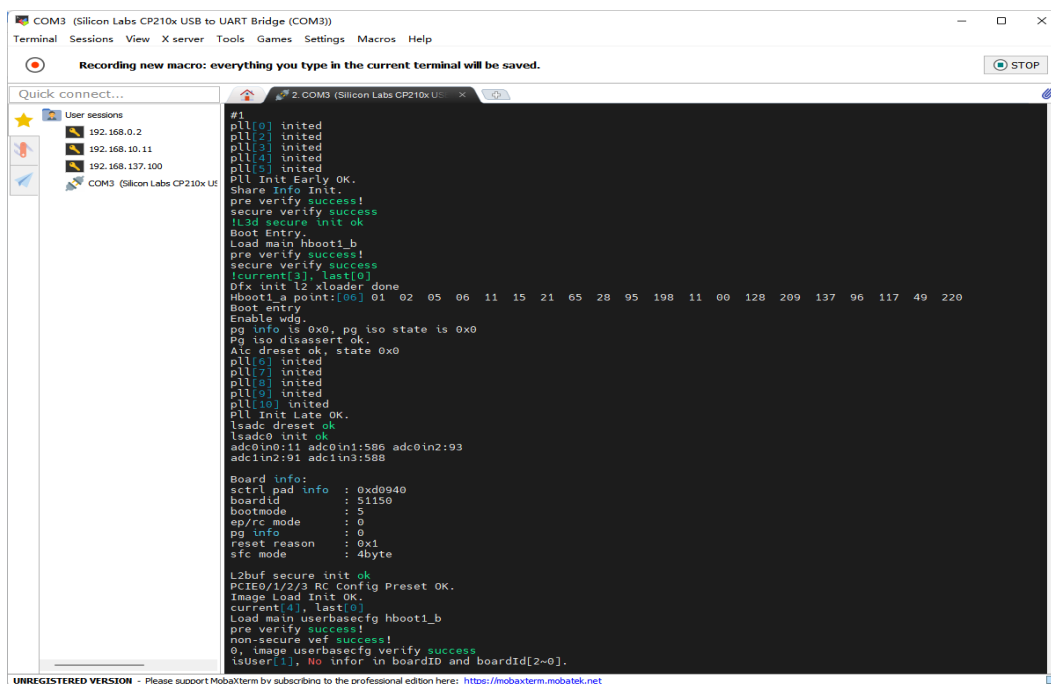


图 4-7 串口连接状态

- 网口指示灯绿灯闪烁，表示连线正常，如图 4-8 所示。
- 开发者套件首次上电后等待约 3 分钟才能远程登录，期间请不要断开电源；后续再次上电启动则只需要等待约 1 分钟就能远程登录。

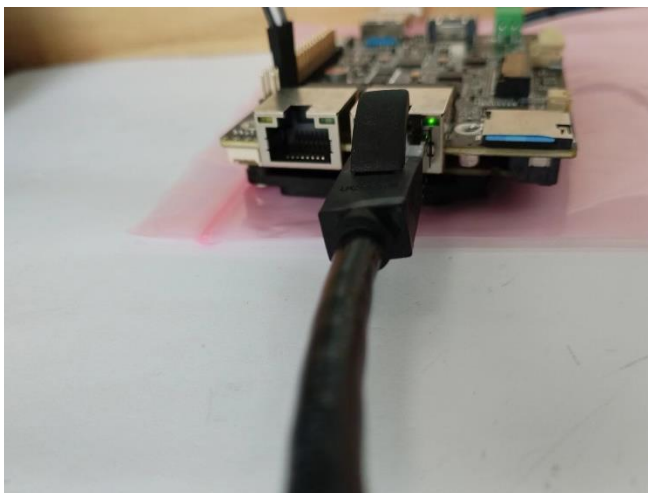


图 4-8 开发者套件连接正常状态

- 步骤 4（可选）** 通过 Type-c 接口连接开发者套件，镜像烧录成功后，开发者套件 Type-C 接口默认 IP 地址为 192.168.0.2，因此远程登录开发者套件前，需要

把 PC 的 USB Type-C 接口和开发者套件的 Type-C 接口地址设置为同一个网段。请提前规划好可用的 IP 地址（本文以 192.168.0.101 为例）。

安装 windows 的 USB 网卡驱动

步骤 1 在“计算机管理”操作界面中选择“设备管理器 > 其他设备”，如下图所示，RNDIS 为未识别状态。



图 4-9 设备管理器

步骤 2 右键单击“RNDIS”，选择“更新驱动程序(P)”。

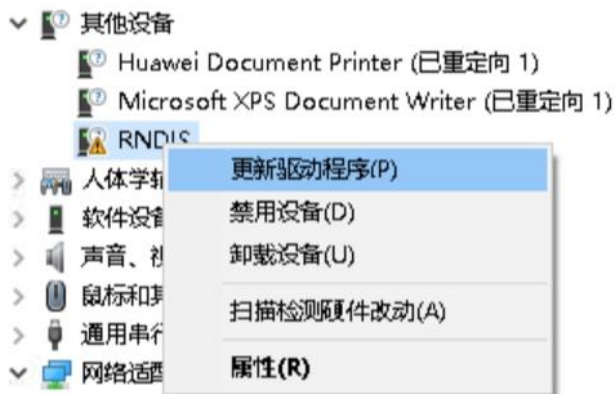


图 4-10 更新 RNDIS

步骤 3 在弹出的“更新驱动程序 - RNDIS 窗口”中选择“浏览我的计算机以查找驱动程序软件 (R)”，然后选择“让我从计算机上的可用驱动程序列表中选择 (L)”，单击“下一步(N)”。

步骤 4 在“常见硬件类型”列表中选择“网络适配器”，单击“下一步(N)”。



图 4-11 选择网络适配器

步骤 5 在“选择要为此硬件安装的设备驱动程序”界面中选择“Microsoft”厂商的“USB RNDIS6 适配器”。



图 4-12 选择驱动程序

步骤 6 单击“下一步”，在弹出的“更新驱动程序警告”窗口选择“是”。

步骤 7 返回“设备管理器 > 网络适配器”，可看到已经正常显示了 USB RNDIS6 适配器的驱动。



图 4-13 RNDIS 驱动正常显示

全爱镜像使用

SD 卡镜像安装

使用官方镜像启动

安装 I2C，进行底板加密

步骤 1 访问如下链接，下载 i2c-tools-4.3.tar.gz。

```
https://mirrors.edge.kernel.org/pub/software/utils/i2c-tools/
```

步骤 2 登录 Arm 架构的 Linux 服务器。

步骤 3 执行如下命令，切换至 root 用户。

```
su - root
```

步骤 4 将软件包“i2c-tools-4.3.tar.gz”上传至 Linux 系统 root 用户属组目录下，例如 /opt。

步骤 5 执行如下命令，进入源码包所在目录，例如/opt。

```
cd /opt
```

步骤 6 执行以下命令，解压 i2c-tools-4.3.tar.gz。

```
tar -zxvf i2c-tools-4.3.tar.gz
```

步骤 7 执行以下命令，进入 i2c-tools-4.3 目录。

```
cd i2c-tools-4.3
```

步骤 8 执行以下命令进行编译。

```
make USE_STATIC_LIB=1
```

编译成功后，在当前目录的子目录“tools”下会生成对应的可执行文件“i2cdetect”与“i2ctransfer”。

调试操作

步骤 1 登录待调试 I2C 的环境。

步骤 2 执行如下命令，切换至 root 用户。

```
su - root
```

步骤 3 将 i2cdetect 与 i2ctransfer 上传至 root 用户属组目录下，例如/opt。

步骤 4 执行以下命令，添加可执行权限。

```
chmod +x i2ctransfer  
chmod +x i2cdetect
```

步骤 5 执行以下命令，写 I2C。

```
./i2ctransfer -y -f 8 w3@0x50 0x00 0x80 0xdf
```

升级 RC3 配套 userBaseConfig.bin

下载链接：[userBaseConfig.bin](#)

1、制作官方镜像 SD 卡，制卡过程参考 [3 一键制卡](#)

使用官方镜像启动核心板

2、升级弹性配置

使用 root 用户登录设备，密码：Mind@123

将华为数字签名后生成 userBaseConfig.bin 文件，拷贝到/run 文件夹，进入 run 文件夹，然后执行

```
/var/davinci/driver/upgrade-tool --device_index -1 --component Usr_Base_Config  
--path userBaseConfig.bin
```

出现如下回显，表示升级 userBaseConfig.bin 文件成功。

```
{"device": 0, "succeed"}
```

制作修改的 RC3 镜像 SD 卡，制卡过程参考 [3 一键制卡](#)

----结束

5 设置网口 IP 地址

镜像烧录完成后，开发者套件 eth1 网口默认 IP 地址为 192.168.137.100，因此远程登录开发者套件前，需要把 PC 的网口和开发者套件网口 IP 地址设置为同一个网段。请提前规划好可用的 IP 地址（本文以 192.168.137.101 为例）。

设置 PC 网口 IP 地址

本步骤以 Windows10 系统为例。

步骤 1 在 PC 上打开“控制面板”，选择“网络和 Internet>网络和共享中心”，单击“更改适配器设置”。



图 5-1 更改适配器设置

步骤 2 鼠标右键单击 PC 网口（连接开发者套件的 eth1 网口）对应的图标（图标一般为“以太网 x”或“本地连接 x”，x 为数字，以现场实际 PC 图标显示的数字为准），单击“属性”。

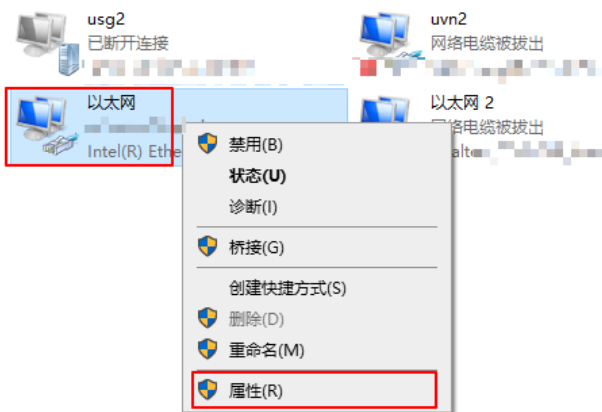


图 5-2 设置 PC 网口 IP 地址界面图

步骤 3 选择“Internet 协议版本 4(TCP/IPv4)”

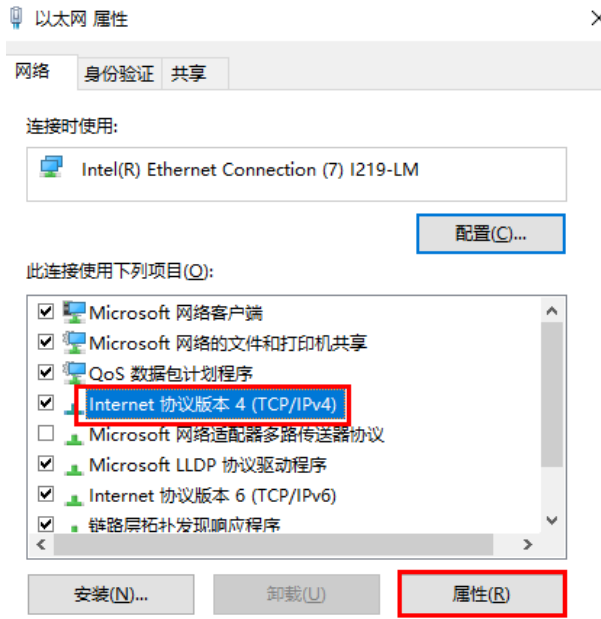


图 5-3 设置 PC 网口 IP 地址界面图

步骤 4 勾选“使用下面的 IP 地址”选项，填写 IP 地址（本示例以 192.168.137.101 为例）、子网掩码默认为 255.255.255.0，默认网关与 DNS 服务器地址为空，单击“确认”保存。



图 5-4 设置 PC 网口 IP 地址界面图

步骤 5 使用快捷键“Win+R”，在运行窗口输入 cmd 进入命令行窗口。输入 ipconfig 命令查询 PC 网口 IP 地址是否修改成功。

```
C:\Users\ATLAS-PIA2>ipconfig
```

其他适配器 本地连接:

连接特定的 DNS 后缀.....:

本地链接 IPv6 地址.....: fe80::59a:842c:c055:60a9%37

IPv4 地址.....: 192.168.137.101

子网掩码.....: 255.255.255.0

默认网关.....

步骤 6（可选）设置开发者套件 ubuntu 镜像网口 IP 地址

- 系统启动后输入账户: root 登录密码: Mind@123
- 输入 sudo su 命令, 输入密码: Mind@123, 进入 root 模式
- 输入命令 vi /etc/netplan/01-netcfg.yaml
- 修改网络配置文件, 如下图, 给网络 0、网络 1 和 usb-typec 配置静态 IP

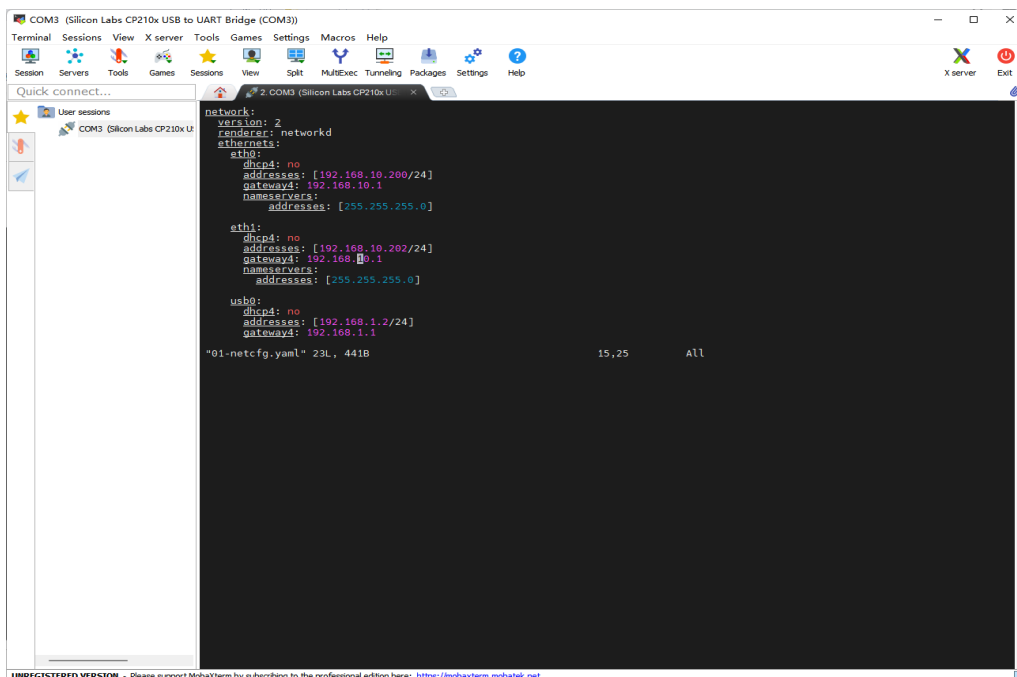


图 5-5 修改网络配置文件

注: vim 操作命令, 输入 i, 进入插入编辑模式
修改完成后按 Esc 退出编辑模式
输入: wq! 保存退出

- e) 输入命令 netplan apply 重启网络
- f) 输入命令 ifconfig 查看 IP 地址已配置成功 (要插网线)

----结束

6 远程登录开发者套件

修改 PC 网口 IP 地址后，通过 SSH 方式登录开发者套件。

通过 SSH 登录开发者套件

步骤 1 双击“MobaXterm_Personal_22.2.exe”启动 SSH 登录工具。

步骤 2 点击左上方的"Session"进入界面。

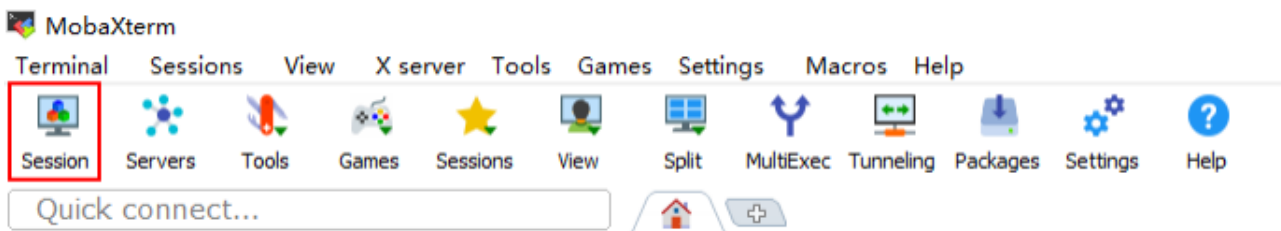


图 6-1 通过 SSH 登录开发者套件界面图

步骤 3 点击左上方的“SSH”进入 SSH 连接配置界面。



图 6-2 通过 SSH 登录开发者套件界面图

步骤 4 按照下图填写开发者套件 eth1 网口 IP 地址（以默认 192.168.137.100 为例）和 root 用户。



图 6-3 通过 SSH 登录开发者套件界面图

步骤 5 点击“OK”按钮，进入远程登录界面，输入 root 用户密码（默认为 Mind@123）登录开发者套件。

SSH 工具界面会出现保存密码提示，可以点击“No”，不保存密码直接登录开发者套件。

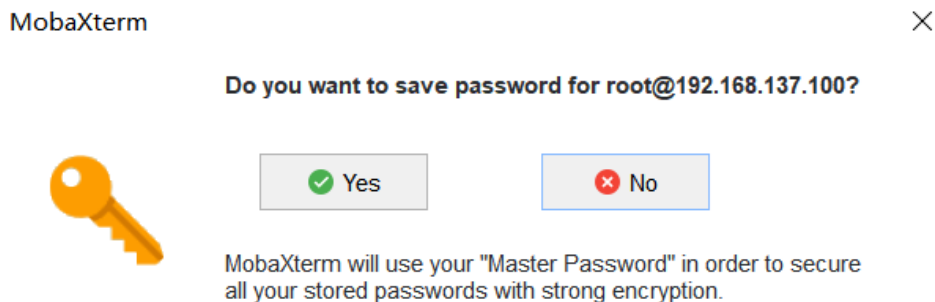


图 6-4 密码保存提示

如果点击"Yes”后进入 MASTERPASSWORD 设置界面，该密码用于找回保存的登录密码，请妥善保管。

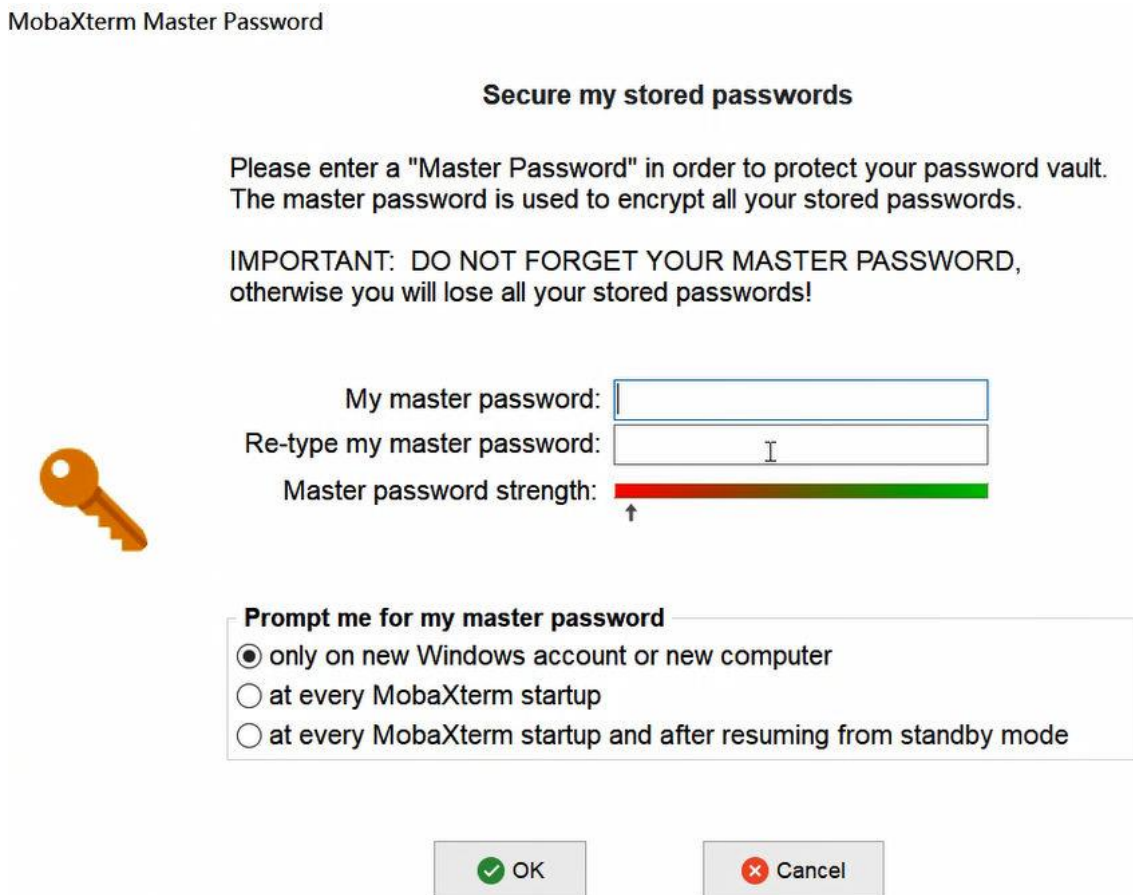


图 6-5 MobaXterm 管理密码界面

远程登录开发者套件成功界面如图 6-6 所示。

```
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.10.0+ aarch64)

* Documentation: https://help.ubuntu.com
* Management:   https://landscape.canonical.com
* Support:      https://ubuntu.com/advantage

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.
Last login: Fri Sep 9 19:26:58 2022 from 192.168.137.99
(base) root@davinci-mini:~#
```

图 6-6 远程登录开发者套件成功界面图

远程登录开发者套件后，此时开发者套件只是和 PC 之间连通，如果需要实现开发者套件和外部网络联网，可参见配置以太网口和 Type-C 网络或者配置路由器直连网络。

----结束

7 接口测试

Ubuntu22.04 镜像接口测试

步骤 1 插入 u 盘，输入 `lsusb` 命令可以看到所有挂载的 usb 设置，如有 u 盘表示 usb 接口正常

步骤 2 输入 `lsusb -vvv` 可以查看 USB 接口的参数，`bcdusb` 参数表示 usb 协议的版本，正常 u 盘接口显示 3.1 版本

步骤 3 断电插入 nvme 盘，系统启动后输入 `fdisk -l`，信息中出现 `/dev/nvme0n1`，表示 nvme 已被识别

----结束

8 运行基础样例

为方便新手开发者进行应用开发和程序运行，镜像中已包含 jupyterlab 软件（可视化代码演示、数据分析工具）可为用户提供一个图形化操作的界面。

登录 jupyterlab

步骤 1 以 root 用户登录开发者套件，进入“notebooks”目录，命令如下：

```
cd/home/HwHiAiUser/samples/notebooks
```

步骤 2(可选) jupyterlab 启动脚本中默认配置为开发者套件 eth1 网口默认 IP 地址，如果启动 jupyterlab 时，PC 连接开发者套件 eth1 网口，且使用默认 IP 地址 (192.168.137.100)，可以跳过本步骤。

如果制卡时修改了 eth1 网口默认 IP 地址或者使用 eth0 网口、Type-C 接口，则需要修改 jupyterlab 启动脚本中的 IP 地址。

执行 `vistart_notebook.sh` 命令修改启动 IP 地址。在键盘按 I 键进入编辑模式，将脚本中以下加粗的 IP 地址修改为现场实际 IP 地址。

修改完成后，在键盘按 Esc 键退出编辑模式，输入:wq 保存文件。

```
jupyterlab--ip192.168.137.100--allow-root
```

步骤 3 执行 `./start_notebook.sh` 命令启动 jupyterlab。

系统回显类似以下信息表示 jupyterlab 已正常运行：

```
[W 19:18:32.954 NotebookApp] Loading JupyterLab as a classic notebook (v6) extension.
```

```
[W 2022-09-09 19:18:32.966 LabApp] 'ip' has moved from NotebookApp to
ServerApp. This □Config will be
passed to ServerApp. Be sure to update your □config before our next release.
[W 2022-09-09 19:18:32.966 LabApp] 'allow_root' has moved from NotebookApp to
ServerApp. This □Config
will be passed to ServerApp. Be sure to update your □Config before our next
release.
[W 2022-09-09 19:18:32.966 LabApp] 'allow_root' has moved from NotebookApp to
ServerApp. This □config
will be passed to ServerApp. Be sure to update your □Config before our next
release.
[I 2022-09-09 19:18:32.976 LabApp] JupyterLab extension loaded from
/usr/local/miniconda3/lib/python3.7/
site-packages/jupyterlab
[I 2022-09-09 19:18:32.976 LabApp] JupyterLab application directory is
/usr/local/miniconda3/share/jupyter/
lab
[I 19:18:32.990 NotebookApp] Serving notebooks from local directory:
/home/HwHiAiUser/samples/
notebooks
[I 19:18:32.990 NotebookApp] Jupyter Notebook 6.5.2 is running at:
[I 19:18:32.990 NotebookApp] http://192.168.137.100:8888/?
token=67cb1b9b39e907bb2150c010a0fdf2a572bf56e6be176eed
[I 19:18:32.990 NotebookApp] or http://127.0.0.1:8888/?
token=67cb1b9b39e907bb2150c010a0fdf2a572bf56e6be176eed
```

步骤 4 在 PC 打开浏览器，复制步骤 3 回显中加粗的网址链接（包含 token）到浏览器或在回显信息中按住键盘“Ctrl”键并使用鼠标左键单击网址链接，进 Ajupyterlab 界面，即可运行开发者套件预置的 Python 推理样例。

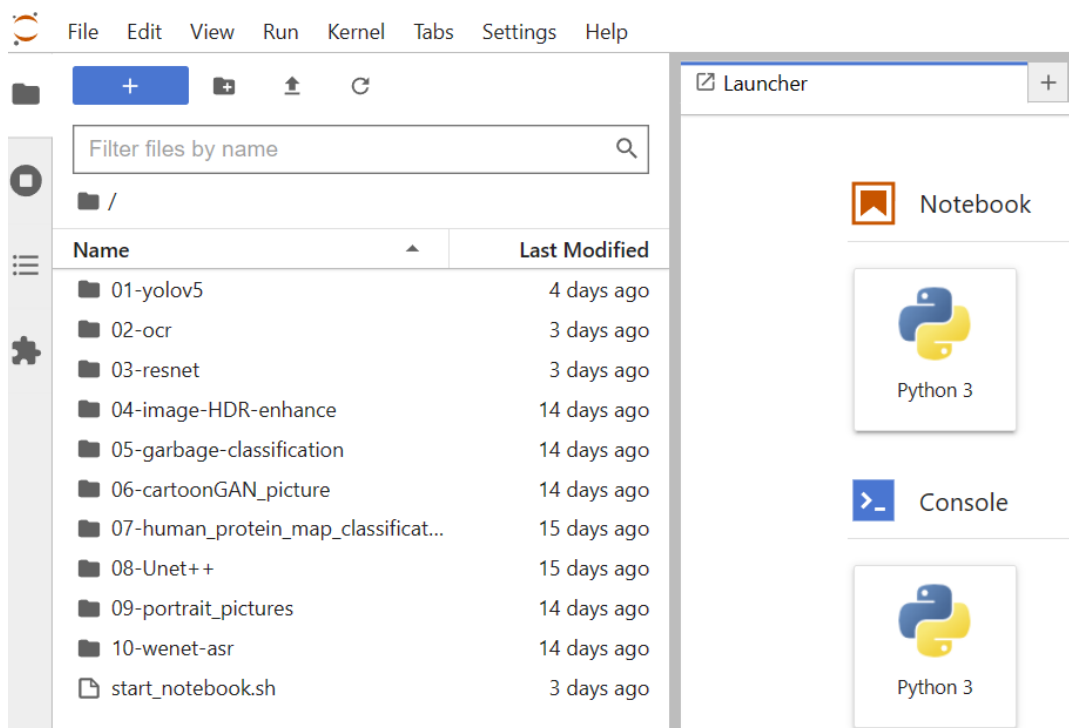


图 8-1 样例目录

运行完成后大约等待 5 到 10 秒直到模型推理完成，查看推理结果，如下图 8-2 所示。

图 8-2 推理结果

```
In [5]: net = Net(device, model_path, idx2label_list)
images_list = [os.path.join(images_path, img)
               for img in os.listdir(images_path)
               if os.path.splitext(img)[1] in IMG_EXT]

for image in images_list:
    print('images: {}'.format(image))
    img = transfer_pic(image)
    pred_dict = net.run([img])
    display_image(image, pred_dict)

print("****run finish****")
net.release_resource()

init resource stage:
model_id:1
init resource success
images:./data/dog2_1024_683.jpg
data interaction from host to device
data interaction from host to device success
execute stage:
execute stage success
data interaction from device to host
data interaction from device to host success
===== top5 inference results: =====
Standard Poodle: 0.935546875
Miniature Poodle: 0.041107177734375
Toy Poodle: 0.0191192626953125
Cocker Spaniels: 0.0028858184814453125
```



```
****run finish****
Releasing resources stage:
Resources released successfully.
```

----结束