

全爱科技 AI 开发套件 QA200A2-U1

用户指南

文档版本 01

发布日期 2025-9-30



全爱科技（上海）有限公司

版权所有 全爱科技（上海）有限公司 2025. 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



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：上海市闵行区剑川路920号2栋3层 邮编：200240

网址：www.quanaichina.com

前言

概述

本文档详细介绍了 QA200A2-U1 的产品特点，系统架构，产品规格，接口测试等。使得客户对 QA200A2-U1 有更加深入的了解。

读者对象

本文档主要适用于以下人员

- 用户企业开发人员，售前人员
- AI 开发者与研究人员

文档更新记录

版本	日期	更新记录
0.1	2025-9-30	初版发布

支持版本如下表：

操作系统版本	Ubuntu 22.04 LTS Arm64
固件与驱动版本	7.5.0.2.220/23.0.RC3
CANN 版本	8.0.0
全爱科技 硬件产品：	QA200A2-U1

目 录

- 1 安全 - 5 -
 - 1.1 电气安全 - 5 -
 - 1.2 操作安全 - 5 -
 - 1.3 软件安全 - 6 -
 - 1.4 维护与处置 - 6 -
 - 1.5 其他注意事项 - 6 -
- 2 产品简介 7
 - 2.1 产品概述 7
 - 2.2 产品特点 7
- 3 接口测试 9
 - 3.1 NVME 接口测试 10
 - 3.2 USB 接口测试 10
 - 3.3 HDMI 测试 10
 - 3.4 emmc 测试 11
 - 3.5 网口测试 11
 - 3.6 IIC 测试 12
 - 3.6.1 编译操作 12
 - 3.6.2 调试操作 13
- 4 环境安装 15
 - 4.1 环境要求 15
 - 4.2 软件清单 15
 - 4.3 依赖关系 16
 - 4.4 安装与部署 16
 - 4.4.1 准备工作 16
 - 4.4.2 系统基础环境 17
 - 4.4.3 基础环境补充安装 19

4.4.4 基础环境验证	20
4.5 常用命令	21
4.5.1 命令清单	21
4.5.2 命令操作	21
4.6 常见问题及解决办法	22
5 开发资料导航	28
5.1 资料网站	28
5.2 使用 WinSCP 传输文件	28

1 安全

1.1 电气安全

电源要求

- 必须使用官方提供的 12V/5A 电源适配器，严禁使用不符合规格的电源
- 连接电源前，确保电源电压与当地电网电压一致
- 电源适配器应放置在通风良好的位置，避免覆盖任何物品
- 插拔电源时应握住插头部分，而非电源线

防触电措施

- 操作开发板前确保手部干燥，避免在潮湿环境中使用
- 开发板运行时，请勿触摸电路板上的元器件和接口引脚
- 严禁在开发板通电状态下进行插拔扩展模块等操作
- 如遇液体泼溅到设备上，应立即断开电源并联系技术支持

1.2 操作安全

机械安装

- 安装或拆卸扩展模块时，应先断开电源
- 连接 M 摄像头等接口时，注意对齐接口方向，避免用力过猛损坏针脚
- 开发板应固定在不平穩表面，避免跌落或碰撞
- 请勿在开发板上放置任何重物

环境要求

- 工作环境温度应保持在 0 °C 至 40 °C 之间
- 避免在多尘、潮湿或有腐蚀性气体的环境中使用
- 远离强电磁干扰源，如大功率电机、变压器等
- 确保设备周围有足够空间散热，至少预留 10cm 通风空间

1.3 软件安全

系统安全

- 开发板接入网络时，应配置适当的防火墙规则
- 重要数据应定期备份，防止意外丢失

权限管理

- 避免长期使用 root 权限进行开发工作
- 为不同开发任务创建专用用户账户，并分配适当权限
- 对外提供网络服务时，限制不必要的端口开放

1.4 维护与处置

日常维护

- 清洁设备表面时，应先断开电源，使用干燥柔软的布料擦拭
- 如需清洁接口，可使用专用电子清洁剂和软毛刷
- 长期不使用时，应断开电源并妥善存放

故障处理

- 如发现设备异常发热、有异味或异响，应立即断开电源
- 遇到硬件故障时，请勿自行拆解设备，应联系官方技术支持

废弃处置

- 本设备包含电子元器件，应按照当地环保规定进行废弃处理
- 处置前请删除设备中存储的所有敏感数据

1.5 其他注意事项

- 开发板不具备防水功能，请勿在可能接触液体的环境中使用
- 禁止私自改装设备或更换非官方配件
- 设备运行时，避免堵塞散热孔
- 运输过程中应使用原包装或具备防震保护的包装材料
- 遵循以上安全注意事项可确保设备正常运行，减少故障风险，并保障使用者的人身安全。

2 产品简介

2.1 产品概述

全爱科技 AI 开发套件 QA200A2-U1 用于帮助开发者完成全功能、多形态的 AI 应用开发与设计评估，最大可提供 20TOPS INT8 的计算能力。

全爱科技 AI 开发套件 QA200A2-U1 可以实现语音、图像与视频等多种数据分析与推理计算，可广泛用于智能监控、机器人、无人机、视频服务器等场景。

2.2 产品特点

- 最大可提供 20TOPS INT8 算力。
- 支持多种规格的 H.264、H.265 视频编解码，最大可支持 20 路 1080P 30fps 的视频解码，12 路 1080P 30fps 的视频编码，适用于用户不同的视频处理需求。
- 提供丰富的外设接口，满足多种产品形态开发需求

多元接口配置

- **USB 3.0 ×2 (Type-A)**
可接入 UVC 标准摄像头、深度相机（如 Intel RealSense）或 AI 视频采集棒，实现即插即用。
- **HDMI 1.4a 输出 (1080p@60Hz)**
支持本地显示调试，可直接连接显示器查看 AI 推理结果、视频分析画面或系统状态信息。
可选 LVDS/eDP 接口（通过扩展板）适用于嵌入式设备集成小型液晶屏。
- **千兆以太网 (RJ45)**

支持 TCP/IP、UDP、MQTT 等协议，适用于远程监控、数据回传与边缘协同计算。

- **Wi-Fi 6 + Bluetooth 5.2 (板载)**

支持高速无线连接、低延迟语音传输与蓝牙设备配对（如麦克风阵列、遥控手柄）

丰富软件生态

- 支持主流嵌入式 Linux 操作系统（如 Ubuntu Core、Yocto 定制系统），提供稳定可靠的运行环境。
- 系统经过深度优化，确保低延迟响应与高资源利用率，适用于长时间运行的边缘计算场景。
- 内置高性能 AI 推理引擎，兼容主流深度学习框架模型，包括：

TensorFlow Lite

PyTorch (via TorchScript)

ONNX Runtime

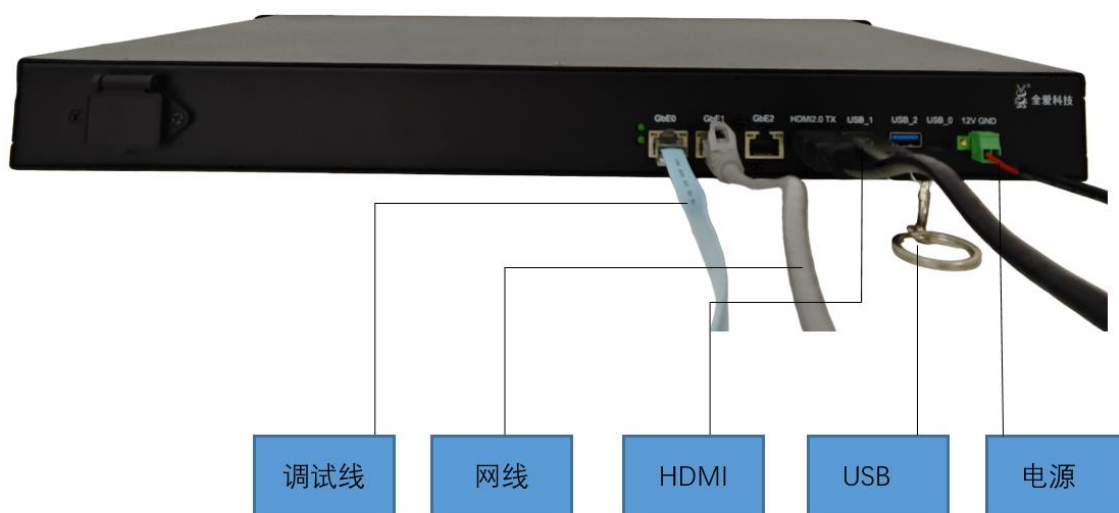
- 支持 C/C++ 和 Python 开发语言，便于快速原型开发与性能优化。
- 配套提供命令行工具、性能监控工具（如算力利用率、功耗、温度监测）及调试接口。

3 接口测试

接口测试说明：进行接口测试需要进入板卡系统，进入系统的操作步骤请参考快速开始文档

注：接口描述详情可参考产品白皮书

个接口连接如图所示：



3.1 NVME 接口测试

进入板卡输入 `fdisk -l`，可查看到一个 nvme 存储盘。

```
(base) root@quanai:~# fdisk -l
Disk /dev/nvme0n1: 465.76 GiB, 500107862016 bytes, 976773168 sectors
Disk model: ZHITAI Ti600 500GB
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/mmcblk1: 57.64 GiB, 61891149824 bytes, 120881152 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: F3BBE73-3366-4BDF-B1E2-9034D7F432AC

Device            Start      End  Sectors  Size Type
/dev/mmcblk1p1     559104    561151    2048    1M Linux filesystem
/dev/mmcblk1p2     561152 120776703 120215552 57.3G Linux filesystem
/dev/mmcblk1p3 120776704 120879103   102400   50M Microsoft basic data

Disk /dev/mmcblk0: 58.25 GiB, 62545461248 bytes, 122159104 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
(base) root@quanai:~#
```

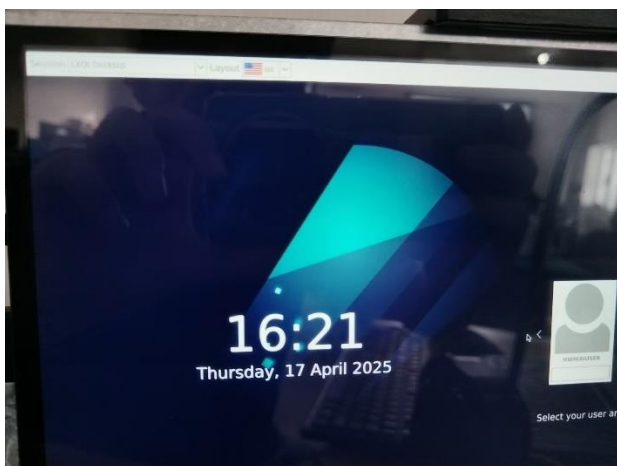
3.2 USB 接口测试

将 U 盘插入开发者套件中，输入 `lsusb` 可以识别 U 盘

```
(base) root@quanai:~# lsusb
Bus 004 Device 002: ID 0bda:0411 Realtek Semiconductor Corp. Hub
Bus 004 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 003 Device 003: ID 0bda:c820 Realtek Semiconductor Corp. 802.11ac NIC
Bus 003 Device 004: ID 17ef:38ac Lenovo
Bus 003 Device 002: ID 0bda:5411 Realtek Semiconductor Corp. RTS5411 Hub
Bus 003 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
(base) root@quanai:~#
```

3.3 HDMI 测试

HDMI 接口接入显示器，启动后会有登录界面显示



3.4 emmc 测试

使用 `fdisk -l` 查看 emmc 设备

```
(base) root@quanai:~# fdisk -l
Disk /dev/nvme0n1: 465.76 GiB, 500107862016 bytes, 976773168 sectors
Disk model: ZHITAI Ti600 500GB
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/mmcblk1: 57.64 GiB, 61891149824 bytes, 120881152 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: F3BBBE73-3366-4BDF-B1E2-9034D7F432AC

Device            Start       End   Sectors  Size Type
/dev/mmcblk1p1     559104     561151     2048    1M Linux filesystem
/dev/mmcblk1p2     561152 120776703 120215552 57.3G Linux filesystem
/dev/mmcblk1p3 120776704 120879103   102400    50M Microsoft basic data

Disk /dev/mmcblk0: 58.25 GiB, 62545461248 bytes, 122159104 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

3.5 网口测试

- 使用网线连接 PC 机网口板卡 eth0, 配置 PC 网络链接 IP: 192.168.2.10
- 使用 `ifconfig eth0 192.168.2.100` 配置网络 IP
- PC 机 Ping 192.168.2.100 是可以 ping 通的
- 同理板卡 ping 192.168.2.10 也可以 ping 通

- 使用网线连接 PC 机网口板卡 eth1, 配置 PC 网络链接 IP: 192. 168. 137. 10
- PC 机 Ping 192. 168. 137. 10 是可以 ping 通的
- 同理板卡 ping 192. 168. 137. 100 也可以 ping 通

3.6 IIC 测试

IIC 源码在 i2c-tools-4.3 文件夹下

3.6.1 编译操作

- 访问如下链接, 下载 i2c-tools-4.3.tar.gz。

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

登录 Arm 架构的 Linux 服务器。

- 执行如下命令, 切换至 root 用户。

```
su - root
```

- 使用“WinSCP”, 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。详细操作请参见 11.2 使用 WinSCP 传输文件。
- 执行如下命令, 进入源码包所在目录, 例如/opt。

```
cd /opt
```

- 执行以下命令, 解压 i2c-tools-4.3.tar.gz。

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

- 执行以下命令, 进入 i2c-tools-4.3 目录。

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

- 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”。

3.6.2 调试操作

- 1、登录待调试 I2C 的环境。
- 2、执行如下命令，切换至 root 用户。

```
su - root
```

- 3、使用“WinSCP”，将 i2cdetect 与 i2ctransfer 上传至 root 用户属组目录下，例如/opt。
- 4、执行以下命令，添加可执行权限。

```
chmod +x i2ctransfer
```

```
chmod +x i2cdetect
```

- 5、执行以下命令，查看系统的 I2C 总线信息。

```
./i2cdetect -l
```

结果如下所示

```
i2c-0 i2c HiSilicon I2C Controller 300170000.i2c0 I2C adapter
i2c-1 i2c HiSilicon I2C Controller 300180000.i2c1 I2C adapter
i2c-2 i2c HiSilicon I2C Controller 300190000.i2c2 I2C adapter
i2c-3 i2c HiSilicon I2C Controller 3001a0000.i2c3 I2C adapter
i2c-5 i2c HiSilicon I2C Controller c4170000.i2c5 I2C adapter
i2c-6 i2c HiSilicon I2C Controller c4180000.i2c6 I2C adapter
i2c-7 i2c HiSilicon I2C Controller 82060000.i2c7 I2C adapter
i2c-8 i2c HiSilicon I2C Controller 82070000.i2c8 I2C adapter
i2c-9 i2c HiSilicon I2C Controller c4190000.i2c9 I2C adapter
i2c-13 i2c HiSilicon I2C Controller 401040000.i2c13 I2C
adapter
```

- 6、执行以下命令，使用 i2cdetect -y 查询 i2c 上挂载的设备。

```
./i2cdetect -y 5
```

结果如下所示

```
0 1 2 3 4 5 6 7 8 9 a b c d e f
```

```

00:  --  --  --  --  --  --  --  --
10:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
20:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
30:  --  --  UU  --  --  --  --  --  --  --  --  --  --  --  --  --
40:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
50:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
60:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
70:  --  --  --  --  --  --  --  --

```

说明： 当前表明只有 0x32 挂载有设备。

7、执行以下命令，写 I2C。

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

字段说明

8: I2C 设备号。

w3: 表示写 3 个值，前两位为对应 16 位寄存器地址，之后则为要写的值。

0x50: I2C 从设备地址。

0x00 0x80 : 表示往 0x0080 地址写值。

0xdf : 表示给对应寄存器写 0xdf。

8、执行以下命令，读 I2C。

```
./i2ctransfer -y -f 8 w2@0x50 0x00 0x80 r1
```

字段说明

8 : I 2C 的总线号。

w2: 表示写 2 个值，即要读的对 16 位寄存器地址。

0x50: I 2C 从设备地址。

0x00 0x80: 表示读取 0x0080 地址的值。

r1: 表示从寄存器开始读取 1 个字节

4 环境安装

4.1 环境要求

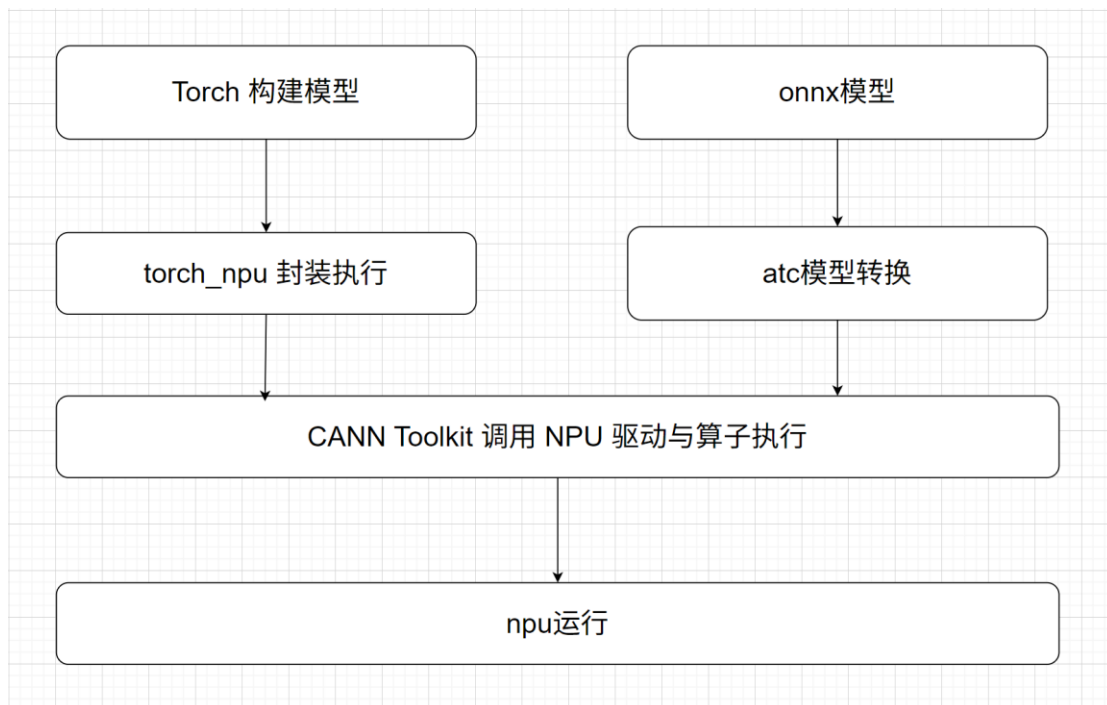
(出厂已安装，如无必要不用重复安装)

- OS 版本：UbuntuARM6422.04
- 驱动：23.0.0
- CANN：8.0.0
- Python 版本：>=3.9
- PyTorch 系列：pytorch-2.1.0 | torch_npu-2.1.0.post3

4.2 软件清单

序号	名字	类别	必装	描述
1	CANN Toolkit	系统	☒	基础开发套件，用于用户开发应用、自定义算子和模型转换等。
2	miniconda3	AI	☐	Python 环境管理工具。
3	python	AI	☐	基于 miniconda3 管理，这里默认自带 python3.9.2
4	torch	AI	☐	目前，最主流的深度学习框架。
5	torch_npu	AI	☐	Torch 的 Ascend 扩展，适配昇腾 NPU，以便使用 Torch 的开发人员可以获得昇腾 AI 处理器的强大计算能力。
6	npu-smi	系统	☒	Ascend NPU 的监控与管理工具，用于查看 NPU 的利用率、内存、功耗、进程等运行状态。
7	atc	系统	☒	模型转换工具，用于将深度学习模型编译为 Ascend NPU 可以直接运行的格式

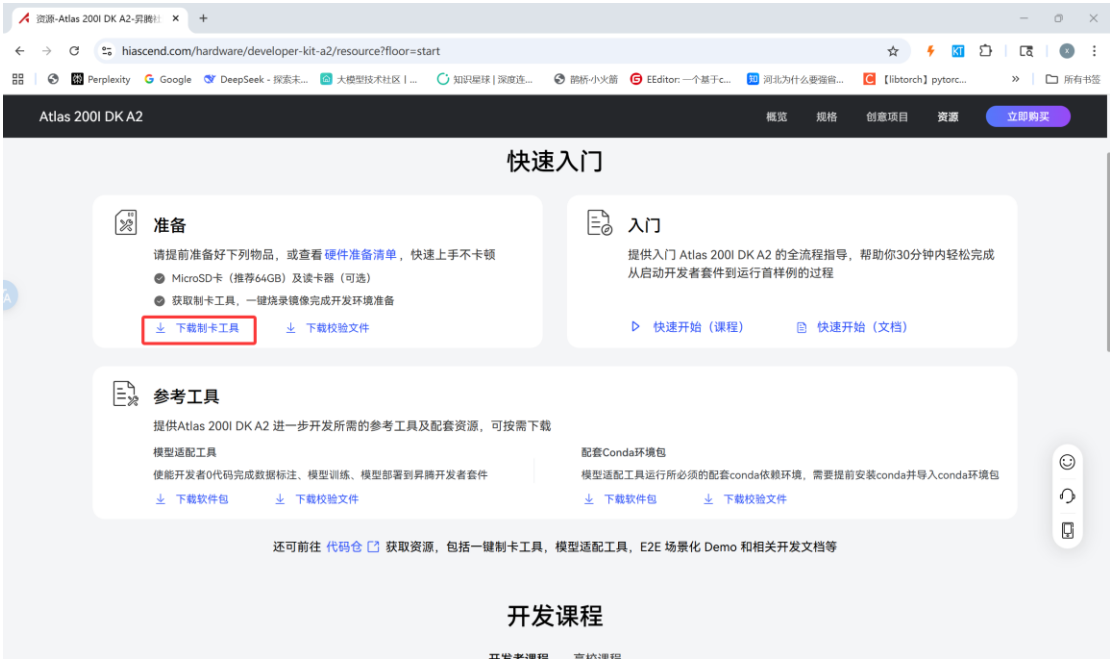
4.3 依赖关系



4.4 安装与部署

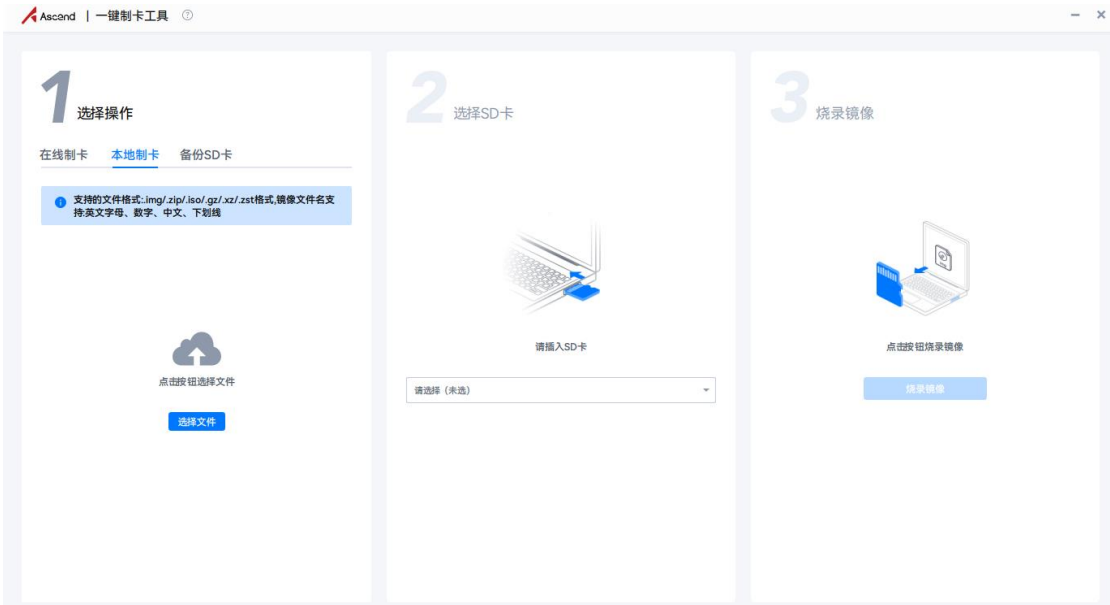
4.4.1 准备工作

安装官方制卡工具。下载网址：<https://www.hiascend.com/hardware/developer-kit-a2/resource?floor=start>



4.4.2 系统基础环境

使用官方制卡工具，烧录镜像。



镜像基础环境

操作系统信息：

```
PRETTY_NAME="Ubuntu 22.04.5 LTS"
NAME="Ubuntu"
VERSION_ID="22.04"
VERSION="22.04.5 LTS (Jammy Jellyfish)"
VERSION_CODENAME=jammy
ID=ubuntu
ID_LIKE=debian
HOME_URL="https://www.ubuntu.com/"
SUPPORT_URL="https://help.ubuntu.com/"
BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/"
PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-policy"
UBUNTU_CODENAME=jammy
```

软件、固件等信息：

```

      NPU ID                : 0
      Product Name          : IT22MMDA
      Model                 : Model=
      Manufacturer          : Huawei
      Serial Number         : 102366342654
      Software Version       : 23.0.0
      Firmware Version      : 7.5.0.2.220
      Board ID              : NA
      PCB ID                : A
      BOM ID                : 1
      Chip Count            : 1
      Faulty Chip Count     : 0

(base) root@qunai-mini:/usr/local/Ascend/ascend-toolkit/latest# cat /usr/local/Ascend/ascend-toolkit/latest/version.cfg
# version: 1.0
runtime_running_version=[7.6.0.1.220:8.0.0]
compiler_running_version=[7.6.0.1.220:8.0.0]
hccl_running_version=[7.6.0.1.220:8.0.0]
opp_running_version=[7.6.0.1.220:8.0.0]
toolkit_running_version=[7.6.0.1.220:8.0.0]
aoe_running_version=[7.6.0.1.220:8.0.0]
ncs_running_version=[7.6.0.1.220:8.0.0]
runtime_upgrade_version=[7.6.0.1.220:8.0.0]
compiler_upgrade_version=[7.6.0.1.220:8.0.0]
hccl_upgrade_version=[7.6.0.1.220:8.0.0]
opp_upgrade_version=[7.6.0.1.220:8.0.0]
toolkit_upgrade_version=[7.6.0.1.220:8.0.0]
aoe_upgrade_version=[7.6.0.1.220:8.0.0]
ncs_upgrade_version=[7.6.0.1.220:8.0.0]
runtime_installed_version=[7.6.0.1.220:8.0.0]
compiler_installed_version=[7.6.0.1.220:8.0.0]
hccl_installed_version=[7.6.0.1.220:8.0.0]
opp_installed_version=[7.6.0.1.220:8.0.0]
toolkit_installed_version=[7.6.0.1.220:8.0.0]
aoe_installed_version=[7.6.0.1.220:8.0.0]
ncs_installed_version=[7.6.0.1.220:8.0.0]
(base) root@qunai-mini:/usr/local/Ascend/ascend-toolkit/latest#
```

CANN版本: 8.0.0

python版本: 3.9.2

源镜像内的apt源,有不少已经404。建议修改替换配置文件
`/etc/apt/sources.list`, 内容如下:

```
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu-ports/ jammy main
restricted universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu-ports/ jammy-updates
main restricted universe multiverse# deb-src
https://mirrors.tuna.tsinghua.edu.cn/ubuntu-ports/ jammy-updates main
restricted universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu-ports/ jammy-backports
main restricted universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu-ports/ jammy-security
main restricted universe multiverse
```

修改后, 执行如下指令更新生效

```
apt update
apt upgrade
```

torch_npu安装

```
#安装 pytorch
pip3 install torch==2.1.0 torchvision==0.16.0
#安装 torch_npu 依赖
pip3 install -i https://pypi.tuna.tsinghua.edu.cn/simple pyyaml
setuptools
#安装 torch_npu
pip3 install torch-npu==2.1.0.post3
```

4.4.3 基础环境补充安装

apt源更新

源镜像内的 apt 源,有不少已经 404。建议修改替换配置文件
`/etc/apt/sources.list`, 内容如下:

```
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu-ports/ jammy main
restricted universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu-ports/ jammy-updates
main restricted universe multiverse# deb-src
```

```
https://mirrors.tuna.tsinghua.edu.cn/ubuntu-ports/ jammy-updates main
restricted universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu-ports/ jammy-backports
main restricted universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu-ports/ jammy-security
main restricted universe multiverse
```

修改后，执行如下指令更新生效

```
apt update
apt upgrade
```

torch_npu安装

```
#安装 pytorch
pip3 install torch==2.1.0 torchvision==0.16.0
#安装 torch_npu 依赖
pip3 install -i https://pypi.tuna.tsinghua.edu.cn/simple pyyaml
setuptools
#安装 torch_npu
pip3 install torch-npu==2.1.0.post3
```

4.4.4 基础环境验证

```
(base) root@davinci-mini:~/scripts# bash ./atlas_env_test.sh
Item |
Status | Results
#####
#####
torch (pip) |
2.1.0 | PASS
torchvision (pip) |
0.16.0 | PASS
torch_npu (pip) |
2.1.0.post3 | PASS
npu-smi info (cmd) |
OK | PASS
atc --help (cmd) |
OK | PASS
python --version (cmd) |
OK | PASS
```

```
python ./test_torch_npu.py (cmd) |
OK | PASS

=====
=====
数据：总计= 7，通过= 7，失败= 0
结论：系统符合性检测通过。
日志：Atlas_Report_20250905_085352.log
```

4.5 常用命令

4.5.1 命令清单

序号	描述	详见
1	npu-smi, npu 的系统管理工具, 可以用于收集设备信息, 查看设备健康状态, 对设备进行配置以及执行固件升级	点击链接
2	atc, CANN 体系下的模型转换工具, 它可以将开源框架的网络模型, 转换为昇腾 AI 处理器支持的.om 格式离线模型。	点击链接

4.5.2 命令操作

npu-smi

```
(base) root@davinci-mini:~# npu-smi info
+-----+
| npu-smi 23.0.rc3 | Version: 23.0.rc3 |
+-----+
| NPU   Name      | Health   | Power(W)  Temp(C)    Hugepages-Usage(page) |
| Chip  Device    | Bus-Id   | AICore(%)  Memory-Usage(MB)      |
+-----+
| 0     310B1     | OK       | 9.1        37          15 / 15              |
| 0     0         | NA       | 0          1584 / 11577         |
+-----+
(base) root@davinci-mini:~#
```

```
(base) root@davinci-mini:~#  
(base) root@davinci-mini:~# npu-smi info -t board -i 0  
      NPU ID                : 0  
      Product Name          : IT22MMDA  
      Model                  : Model=  
      Manufacturer          : Huawei  
      Serial Number         : 102366342654  
      Software Version       : 23.0.rc3  
      Firmware Version       : 7.0.0.5.242  
      Board ID              : NA  
      PCB ID                : A  
      BOM ID                : 1  
      Chip Count            : 1  
      Faulty Chip Count     : 0  
(base) root@davinci-mini:~#
```

atc

```
(base) root@davinci-mini:~/samples/inference/modelInference/sampleYOLOV7/model#  
(base) root@davinci-mini:~/samples/inference/modelInference/sampleYOLOV7/model# atc --model=yolov7x.onnx  
--framework=5 --output=yolov7x --input_shape="images:1,3,640,640" --soc_version=Ascend310 --insert_op_c  
onf=aipp.cfg  
ATC start working now, please wait for a moment.  
...  
ATC run success, welcome to the next use.  
(base) root@davinci-mini:~/samples/inference/modelInference/sampleYOLOV7/model#
```

4.6 常见问题及解决办法

1. 运行 docker 容器，用到-p 网络端口映射的时候会创建默认路由，从而影响正常路由，导致跨网段无法访问；

```
(base) root@davinci-mini:~# ip route | grep default
default dev veth7cab4ca scope link
default dev veth3c7260b scope link
default dev veth542e871 scope link
default dev veth3309584 scope link
default dev veth4d90b20 scope link
default dev veth565727d scope link
default dev vethbed3353 scope link
default dev vethfde0b83 scope link
default dev vethd8272fe scope link
default dev veth135e28b scope link
default dev veth684d848 scope link
default via 10.242.129.1 dev eth0 proto static
(base) root@davinci-mini:~#
```

每启动一个容器，如用到-p参数的，则会创建默认路由，会替换掉正常的路由，而引发通信异常

```
(base) root@davinci-mini:~# route -n
Kernel IP routing table
Destination      Gateway          Genmask         Flags Metric Ref    Use Iface
0.0.0.0          0.0.0.0         0.0.0.0         U        0      0      0 veth7cab4ca
0.0.0.0          0.0.0.0         0.0.0.0         U        0      0      0 veth3c7260b
0.0.0.0          0.0.0.0         0.0.0.0         U        0      0      0 veth542e871
0.0.0.0          0.0.0.0         0.0.0.0         U        0      0      0 veth3309584
0.0.0.0          0.0.0.0         0.0.0.0         U        0      0      0 veth4d90b20
0.0.0.0          0.0.0.0         0.0.0.0         U        0      0      0 veth565727d
0.0.0.0          0.0.0.0         0.0.0.0         U        0      0      0 vethbed3353
0.0.0.0          0.0.0.0         0.0.0.0         U        0      0      0 vethfde0b83
0.0.0.0          0.0.0.0         0.0.0.0         U        0      0      0 vethd8272fe
0.0.0.0          0.0.0.0         0.0.0.0         U        0      0      0 veth135e28b
0.0.0.0          0.0.0.0         0.0.0.0         U        0      0      0 veth684d848
0.0.0.0          10.242.129.1    0.0.0.0         UG       0      0      0 eth0
10.242.129.0     0.0.0.0         255.255.255.0   U        0      0      0 eth0
169.254.0.0      0.0.0.0         255.255.0.0     U        0      0      0 veth684d848
169.254.0.0      0.0.0.0         255.255.0.0     U        0      0      0 veth135e28b
169.254.0.0      0.0.0.0         255.255.0.0     U        0      0      0 vethd8272fe
169.254.0.0      0.0.0.0         255.255.0.0     U        0      0      0 vethfde0b83
169.254.0.0      0.0.0.0         255.255.0.0     U        0      0      0 vethbed3353
169.254.0.0      0.0.0.0         255.255.0.0     U        0      0      0 veth565727d
169.254.0.0      0.0.0.0         255.255.0.0     U        0      0      0 veth4d90b20
169.254.0.0      0.0.0.0         255.255.0.0     U        0      0      0 veth3309584
169.254.0.0      0.0.0.0         255.255.0.0     U        0      0      0 veth542e871
169.254.0.0      0.0.0.0         255.255.0.0     U        0      0      0 veth3c7260b
169.254.0.0      0.0.0.0         255.255.0.0     U        0      0      0 veth7cab4ca
172.17.0.0       0.0.0.0         255.255.0.0     U        0      0      0 docker0
192.168.1.0      0.0.0.0         255.255.255.0   U        0      0      0 eth0
192.168.1.1      192.168.1.1    255.255.255.255 UGH      0      0      0 eth0
192.168.1.1      0.0.0.0         255.255.255.255 UH       0      0      0 eth0
(base) root@davinci-mini:~#
```

● 使用最新 docker 版本测试

无法稳定复现，有次没有关容器的情况下，直接 reboot，然后再起容器就出现了这种情况，但之后就再也没出现，

● 重启前没有关闭容器导致，系统重启后，容器又在次映射，导致出现两个路由，提供的建议是

- 1、重启前手动关闭容器动
- 2、重启后重启网络
- 3、将 docker 升级为最新版本

2. 重启后多了这两条路由：导致 ping 别的网段 ping 不通 只有手动执行 netplan apply 才能 ping 通

```
(base) root@qunai-mini:~# route -n
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
0.0.0.0 0.0.0.0 0.0.0.0 U 0 0 0 eth1
0.0.0.0 172.30.235.254 0.0.0.0 UG 0 0 0 eth1
169.254.0.0 0.0.0.0 255.255.0.0 U 0 0 0 eth1
172.17.0.0 0.0.0.0 255.255.0.0 U 0 0 0 docker0
172.30.235.0 0.0.0.0 255.255.255.0 U 0 0 0 eth1
(base) root@qunai-mini:~# ping 172.28.0.104
PING 172.28.0.104 (172.28.0.104) 56(84) bytes of data.
From 172.30.235.173 icmp_seq=1 Destination Host Unreachable
From 172.30.235.173 icmp_seq=2 Destination Host Unreachable
From 172.30.235.173 icmp_seq=3 Destination Host Unreachable
From 172.30.235.173 icmp_seq=4 Destination Host Unreachable
From 172.30.235.173 icmp_seq=5 Destination Host Unreachable
From 172.30.235.173 icmp_seq=6 Destination Host Unreachable
^C
--- 172.28.0.104 ping statistics ---
7 packets transmitted, 0 received, +6 errors, 100% packet loss, time 6146ms
pipe 4
(base) root@qunai-mini:~#
```

```
(base) root@qunai-mini:~# netplan apply
(base) root@qunai-mini:~# ping 172.28.0.104
PING 172.28.0.104 (172.28.0.104) 56(84) bytes of data.
^C
--- 172.28.0.104 ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 2050ms

(base) root@qunai-mini:~# route -n
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
0.0.0.0 0.0.0.0 0.0.0.0 U 0 0 0 eth1
0.0.0.0 172.30.235.254 0.0.0.0 UG 0 0 0 eth1
172.17.0.0 0.0.0.0 255.255.0.0 U 0 0 0 docker0
172.30.235.0 0.0.0.0 255.255.255.0 U 0 0 0 eth1
(base) root@qunai-mini:~# netplan apply
(base) root@qunai-mini:~# route -n
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
0.0.0.0 172.30.235.254 0.0.0.0 UG 0 0 0 eth1
172.17.0.0 0.0.0.0 255.255.0.0 U 0 0 0 docker0
172.30.235.0 0.0.0.0 255.255.255.0 U 0 0 0 eth1
(base) root@qunai-mini:~# ping 172.28.0.104
PING 172.28.0.104 (172.28.0.104) 56(84) bytes of data.
64 bytes from 172.28.0.104: icmp_seq=1 ttl=63 time=1.27 ms
64 bytes from 172.28.0.104: icmp_seq=2 ttl=63 time=0.294 ms
64 bytes from 172.28.0.104: icmp_seq=3 ttl=63 time=0.301 ms
64 bytes from 172.28.0.104: icmp_seq=4 ttl=63 time=0.294 ms
^C
--- 172.28.0.104 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3041ms
rtt min/avg/max/mdev = 0.294/0.540/1.274/0.424 ms
(base) root@qunai-mini:~#
```

- Netplan 和 NetworkManager 同时启动，导致网络配置混乱，使用 netplan apply 重新配置一下网络就可以了

处理建议为：

修改系统，在系统启动后添加 netplan apply 服务。

3 笔记本已通过 WiFi 连接到 Internet，如何用一根网线让 Atlas200I A2 板卡通过笔记本来上网呢

- 首先将笔记本和板卡 eth0 的以太网口用一根网线接起来（板卡 eth0 设置为 DHCP，笔记本以太网也设置为自动获取）。

当笔记本是 Windows 10 的时候，方法很简单：对任务栏右下角的网络图标单击右键，选择“打开网络和 Internet”设置。

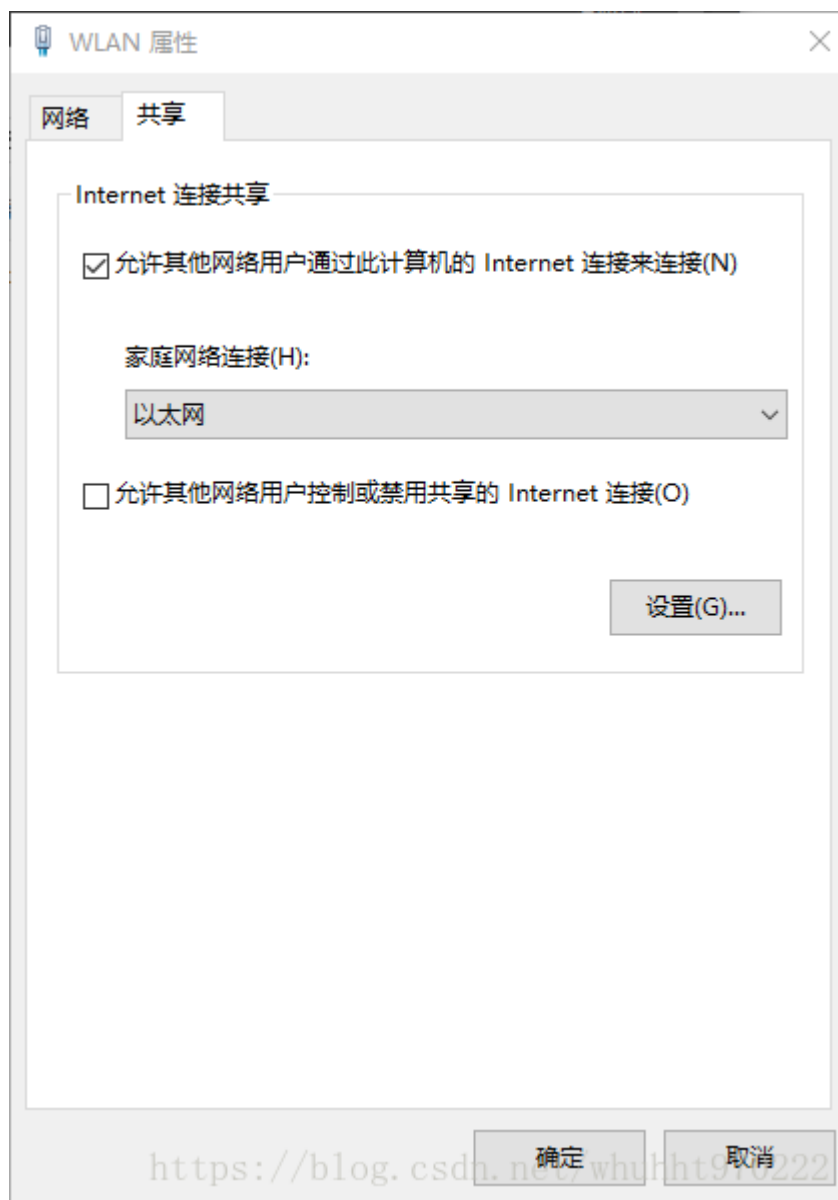
- 在打开的设置窗口中，点击“更改适配器选项”。
- 在你连接到 Internet 的无线网卡项目上右键，“属性”。



<https://blog.csdn.net/whuhht970222>

- 在弹出的对话框中，切换到“共享”选项卡。

- 勾选“允许其他网络用户通过此计算机的 Internet 连接来连接”，家庭网络连接处勾选通过网线与板卡连接的网卡，我这里以太网。点击确定。



现在，你应该可以看到无线网卡的状态多了一条“共享的”。



稍等一会，不出意外，你的板卡就可以访问 Internet 了。

在板卡端 `ifconfig` 可以看到 `eth0` 已经自动分配到一个 IP 地址，
`ping www.baidu.com` 可以 ping 通表示网络上网正常。

4. 如何将 QA200A2-U1, 调节启动模式 TF 卡或者 EMMC

- TF 卡模式转换到 EMMC：R281 的电阻拿下来，放到 R10 上，
然后 R16 进行短接
- EMMC 模式转换到 TF 卡：R10 的电阻拿下来，放到 R281 上，
然后 R16 上的连锡断开

5 开发资料导航

5.1 资料网站

- [全爱科技下载中心](#)
- [测试例程下载地址](#)
- 系统镜像：

<https://pan.baidu.com/s/1LJhhGDTUG2RCqGfIKvj-nQ?pwd=p8f9>

提取码: p8f9

5.2 使用 WinSCP 传输文件

操作场景

在 PC 机上使用 WinSCP 工具进行文件传输。

必备事项

前提条件

目的设备已开启 SFTP 服务。

数据

需准备如下数据：

- 待连接服务器的 IP 地址
- 登录待连接服务器的用户名和密码

软件

WinSCP.exe：此工具为第三方软件，请自行准备。

操作步骤

步骤 1 打开“WinSCP”文件夹，双击“WinSCP.exe”。

弹出“WinSCP 登录”对话框，如图 5-1 所示。

 说明 若系统非中文操作系统，可以单击“Languages”进行界面语言的选择。

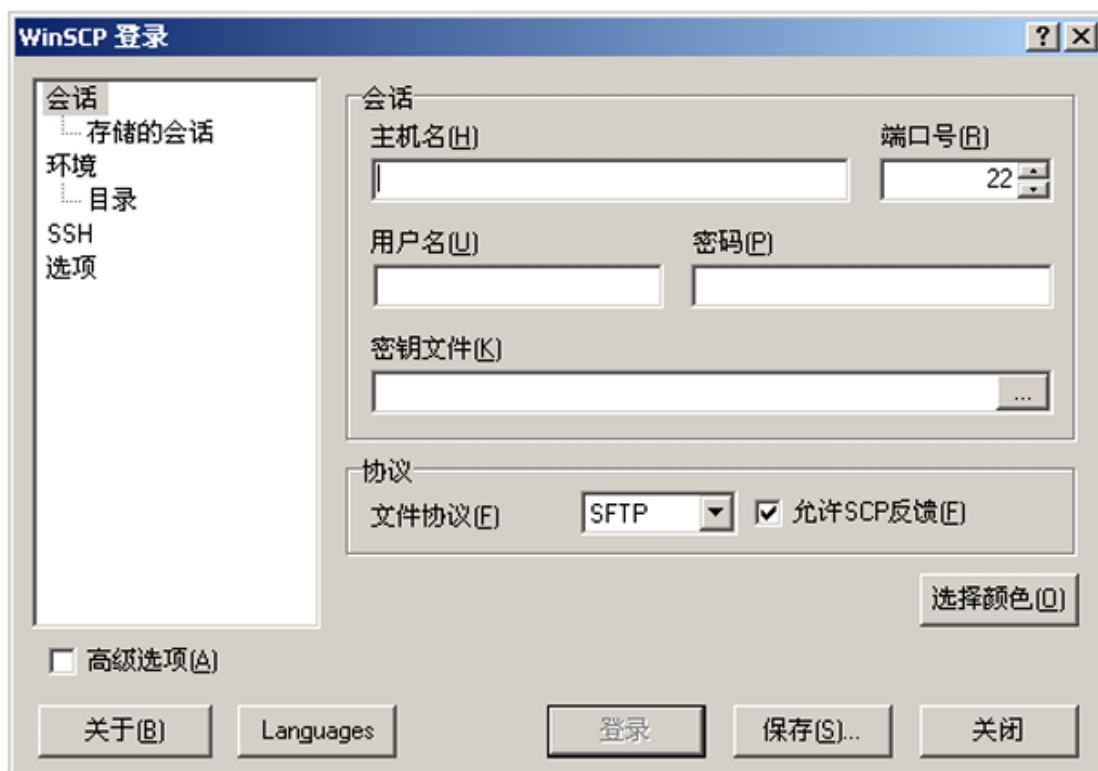


图 5-1

步骤 2 单击“会话”。

步骤 3 设置登录参数。

参数说明如下。

- 主机名 (H)：待连接设备的 IP 地址，例如“192.168.2.10”。
- 端口号 (R)：默认为“22”。
- 用户名 (U)：待连接设备的操作系统用户名，例如“admin123”。
- 密码 (P)：待连接设备的操作系统用户的密码，例如“admin123”。
- 密钥文件 (K)：默认为空，保留默认值。
- 协议：选择默认文件协议“SFTP”，并勾选“允许 SCP 反馈 (F)”。

步骤 4 单击“登录”。

进入“WinSCP”文件传输界面。

说明如果首次登录时没有选择密钥文件，此时会弹出一个警告提示框，询问“是否连接并添加密钥到缓存？”，单击“是（Y）”，进入“WinSCP”文件传输界面。

步骤 5 根据实际需求，在界面左右区的指定目录中进行文件夹的创建、删除和复制等操作

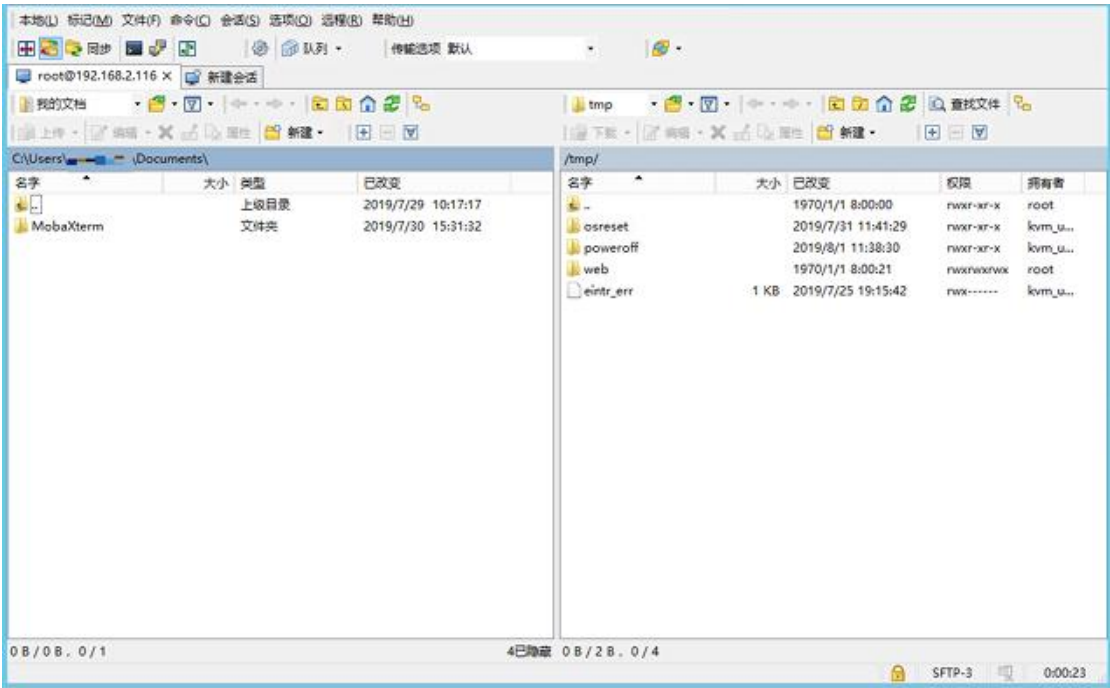


图 5-2WinSCP 界面

说明界面左侧区域为本地 PC 的目录，右侧区域为已连接设备的目录。