

QA 200 RC

环境部署指南

文档版本 01
发布日期 2021-12-27



全爱科技（上海）有限公司

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



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930号D栋3层 邮编：200240网

址：www.quanaichina.com

1 简介

1.1 概述

QA200RC Atlas 200 开发者套件（型号 3000）是以Atlas 200 AI加速模块（型号 3000）为核心的开发者板形态的终端类产品。主要功能是将Atlas 200 AI加速模块（型号3000）的接口对外开放，方便用户快速简捷的使用Atlas 200 AI加速模块（型号3000），可以运用于平安城市、无人机、机器人、视频服务器等众多领域的预研开发。

Atlas 200 AI加速模块（型号 3000）是一款高性能的AI智能计算模块，集成了昇腾310 AI处理器（Ascend 310 AI处理器），可以实现图像、视频等多种数据分析与推理计算，可广泛用于智能监控、机器人、无人机、视频服务器等场景。

说明

昇腾310是一款华为专门为图像识别、视频处理、推理计算及机器学习等领域设计的高性能、低功耗AI芯片。芯片内置2个AI core，可支持128位宽的LPDDR4X，可实现最大22TOPS（INT8）的计算能力。

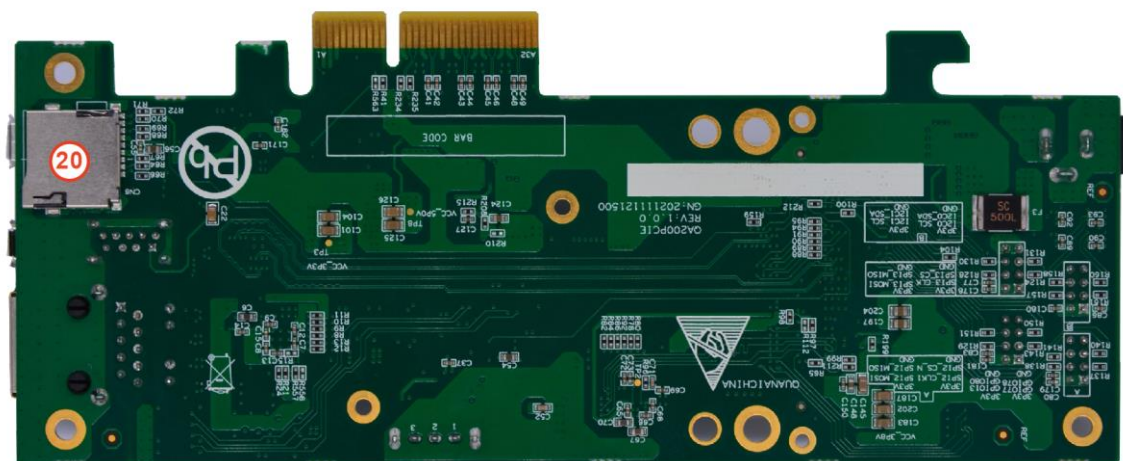
1.2 外观

QA200-RC/EP Atlas 200 开发者套件（型号 3000）外观如图1-1所示。

图 1-1 外观图



A上视图



B下视图

说明

接口对应标识本文中的图仅供参考，具体以实际收货为准

图 1-2 尺寸图（单位：mm）

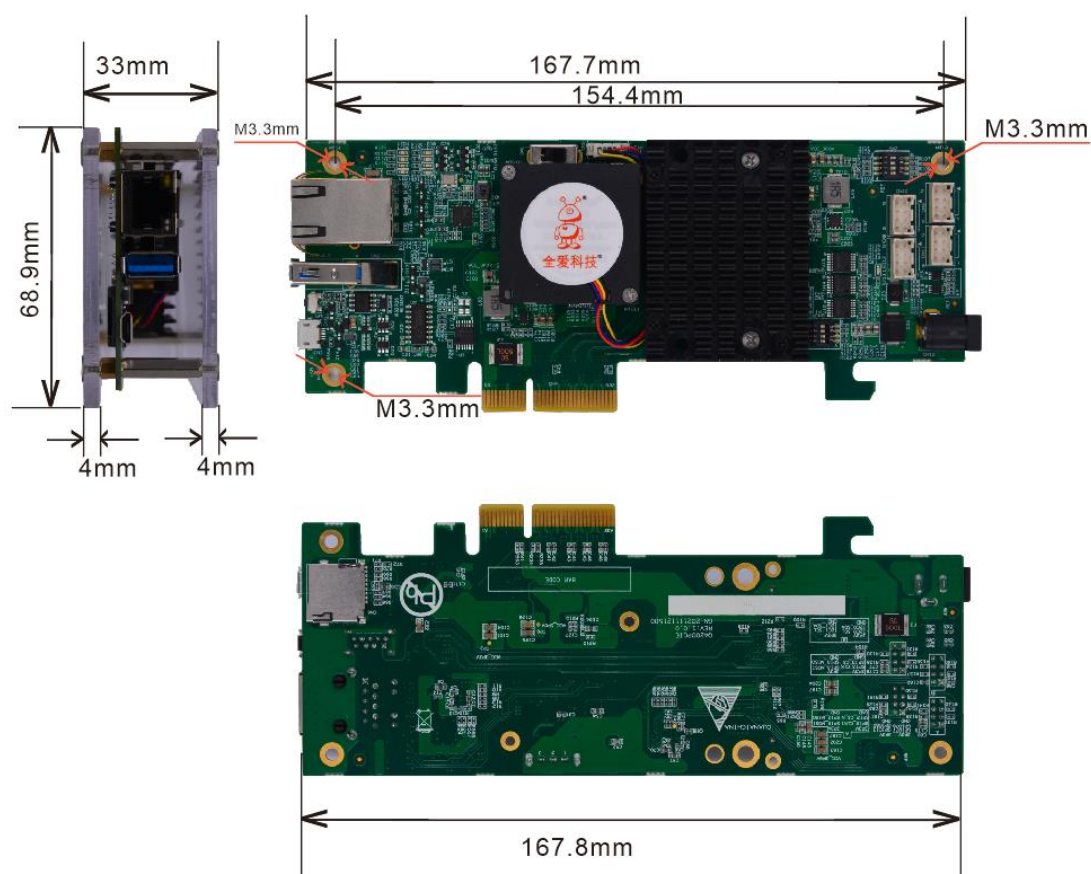


图 1-3 定位孔尺寸图（单位：mm）

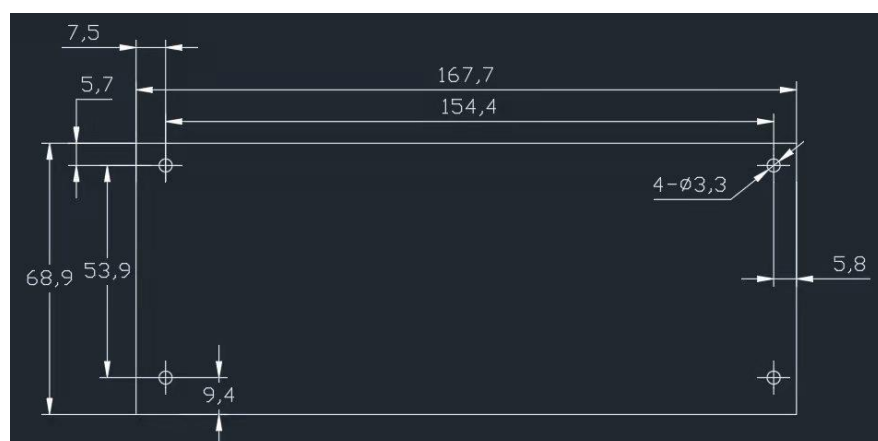
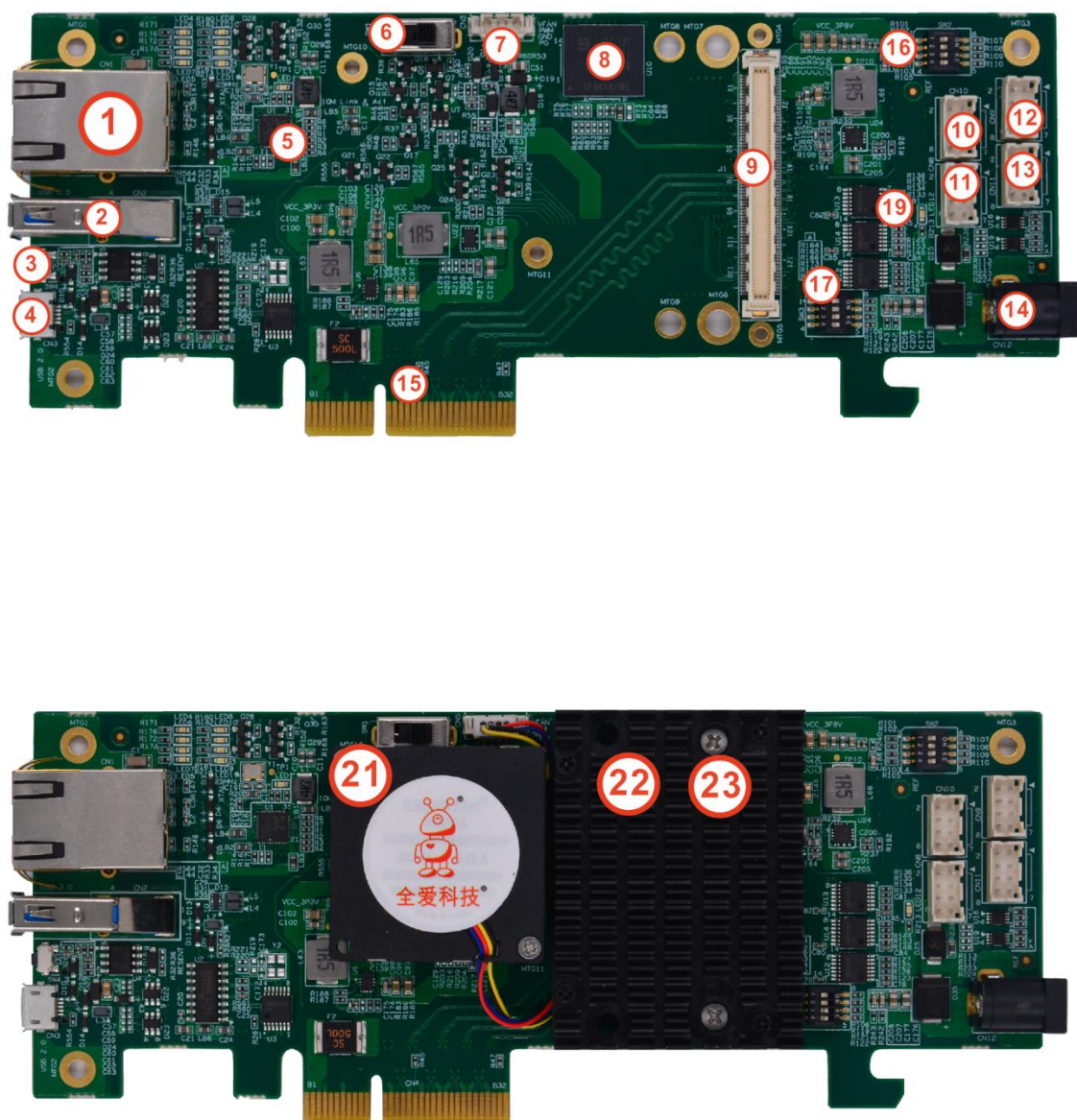


图 1-4 接口说明 QA200-PCIE主板

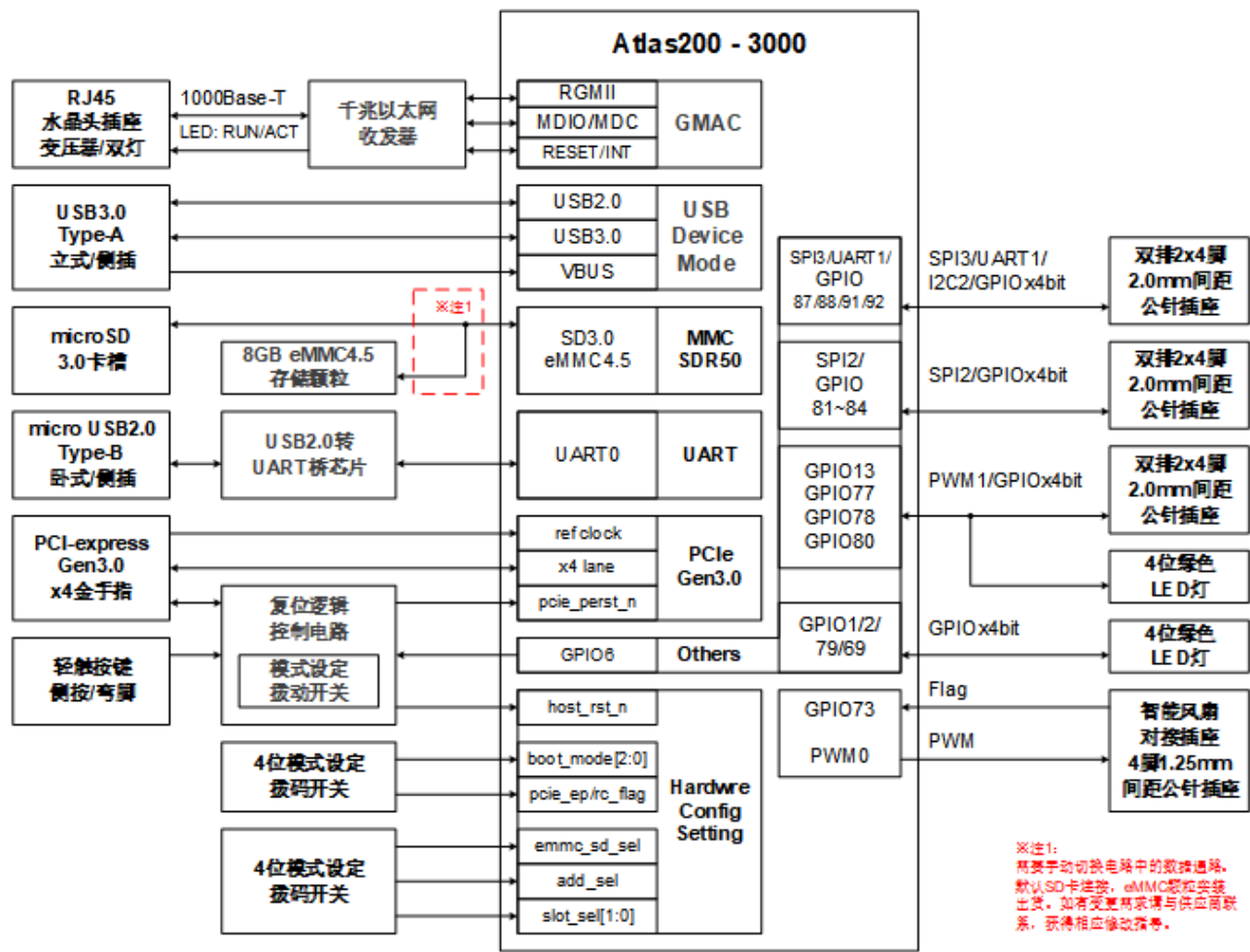


序号	位号	功能说明	原理图
1	CN1	10Base-T、100Base-TX、1000Base-T自适应千兆以太网口，双指示灯。	P02
2	CN2	USB3.0 Type-A插座，仅支持从模式。	P02
3	SW4	轻触开关（无自锁，自动弹起），用于软复位AI处理器。	P03
4	CN3	micro USB2.0 Type-B插座，仅支持从模式，连接AI处理器的调试串口用于AI处理器软件调试使用。	P02
5	U1	自适应千兆网络收发器。	P02
6	SW1	拨动开关，用于设定PCIe的复位模式	P03
7	CN5	智能风扇插座，1.27mm间距公针插座，可接收风扇心跳，并驱动风扇，控制风扇转速，由AI处理器直接控制。默认对接5V风扇。	P03
8	U10	8GB eMMC4.5颗粒对接AI处理器，物理通路和SD卡共用（默认不通）。	P04
9	J1	AI处理器插入卡槽，直接对接Atlas200-3000模块。	P04
10	CN10	多功能I/O扩展插座，2.00mm间距2x4脚公针插座，由AI处理器驱动。	P05
11	CN8	多功能I/O扩展插座，2.00mm间距2x4脚公针插座，由AI处理器驱动。	P05
12	CN9	多功能I/O扩展插座，2.00mm间距2x4脚公针插座，由AI处理器驱动。	P05
13	CN11	多功能I/O扩展插座，2.00mm间距2x4脚公针插座，由AI处理器驱动。	P05
14	CN12	直流电源输入，Jack插座，12V供电输入，用于非PCIe应用场景下，板卡的外部供电输入。	P06
15	CN4	PCI-Epress Gen3.0标准，x4Lane板边金手指。直接对接AI处理器。	P03
16	SW2	AI处理器模式设定用拨码开关。	P04
17	SW3	AI处理器模式设定用拨码开关。	P04
18	LED[11:4]	8位用户自定义功能绿色LED。	P05
19	LED12	电源输入指示灯。	P06
20	CN6	microSD卡槽，支持SD3.0协议标准中SDR50档速度。插-拔卡方式为推-推式，直接对接AI处理器。	P04
21		风扇12	
22		Atlas200 AI加速模块，默认使用垫片	
23		散热片，默认高度为5mm， 可选10mm-15mm高度，15mm高度可不用风扇散热	

1.3 系统框图

QA200-RC Atlas 200 开发者套件主要包含Atlas 200 AI加速模块（型号3000）、PHY三部分，系统架构如图1-5所示。

图 1-5 系统框图（主板名称为 QA200-PCIE）



3 制作 SD 卡

简介

通过SD卡制作功能可以自行制作QA 200 RC开发者板的系统启动盘，完成QA 200 RC操作系统及驱动固件的安装。

QA 200 RC开发者板的SD卡制作方式有如下两种：

- 如果有读卡器，可以将SD卡放入读卡器，将读卡器与用户PC机的USB接口连接，然后通过制卡脚本进行SD卡的制作。
- 如果没有读卡器，可以将SD卡放入QA 200 RC开发者板的卡槽，通过跳线帽/杜邦线进行开发者板相关针脚连接，使得开发者板实现读卡器的功能，然后将开发者板与用户PC机的USB接口连接，再通过制卡脚本进行SD卡的制作。

须知

SD卡制作过程中会自动创建默认用户HwHiAiUser，该用户用来运行应用程序。
用户HwHiAiUser缺省登录密码为 **“Mind@123”**。

读卡器场景

介绍有读卡器的场景下将读卡器与用户PC机的USB接口连接，通过制卡脚本制作SD卡的操作。

硬件准备

1. 准备一张SD卡，建议容量不低于16G。

须知

制作过程将会对SD卡做格式化处理，请提前备份SD卡中数据。

2. 请将SD卡放入读卡器，并将读卡器与用户PC机的USB接口连接。

软件包准备

获取Ubuntu操作系统镜像包与开发者板驱动包，下载信息如表3-1所示。

表 3-1 软件包信息

包信息	包名	说明	下载地址
Ubuntu 操作系统镜像包	ubuntu-18.04.xx-server-arm64.iso	开发者板操作系统镜像包。 Ubuntu操作系统的版本可以为：18.04.4、18.04.5。 注意：为ARM版本的Server软件包。	请从 Ubuntu官网 下载此软件包。
开发者板驱动包	A200dk-npu-driver-{software version}-ubuntu18.04-aarch64-minirc.tar.gz	开发者板驱动包，包含OS外围软件、AI软件栈、维测相关软件及驱动固件信息。 制作SD卡时，会从驱动包中获取固件信息，不需要单独获取固件包。	获取链接 1. “产品系列”请选择“AI开发者套件”。 2. “产品型号”请选择“QA 200 RC开发者套件”。 3. “固件与驱动版本”：选择“1.0.12.alpha”。

须知

软件下载后请保持原命名。

操作步骤

步骤1 请参见[硬件准备](#)将装入SD卡的读卡器与用户PC机的USB接口连接。

步骤2 在用户PC机中执行如下命令安装qemu-user-static、binfmt-support、yaml、squashfs-tools与交叉编译器。

su - root

执行如下命令更新源：

apt-get update

执行如下命令安装相关python依赖：

```
pip3 install pyyaml
```

```
apt-get install qemu-user-static binfmt-support python3-yaml squashfs-tools  
gcc-aarch64-linux-gnu g++-aarch64-linux-gnu
```

步骤3 在用户PC机中以root用户执行如下命令创建制卡工作目录。

```
mkdir $HOME/mksd
```

制卡目录可任意指定。

步骤4 将**软件包准备**获取的Ubuntu操作系统镜像包、开发者板驱动包上传到制卡工作目录中（例如：“\$HOME/mksd”）。

步骤5 在制卡工作目录下（例如：“\$HOME/mksd”）执行如下命令获取制卡脚本。

- 下载制卡入口脚本 “make_sd_card.py”。

从gitee下载：

```
wget https://gitee.com/ascend/tools/raw/master/makesd/generic_script/  
make_sd_card.py
```

从github下载：

```
wget https://raw.githubusercontent.com/Ascend/tools/master/makesd/  
generic_script/make_sd_card.py
```

- 下载制作SD卡操作系统的脚本 “make_ubuntu_sd.sh”。

从gitee下载：

```
wget https://gitee.com/ascend/tools/raw/master/makesd/generic_script/  
make_ubuntu_sd.sh
```

从github下载：

```
wget https://raw.githubusercontent.com/Ascend/tools/master/makesd/  
generic_script/make_ubuntu_sd.sh
```

说明

您可以通过修改 “make_sd_card.py” 中的如下参数配置QA 200 RC的USB网卡IP与NIC网卡IP。

- “NETWORK_CARD_DEFAULT_IP”：NIC网卡的IP地址，默认值 “192.168.0.2”。
- “USB_CARD_DEFAULT_IP”：USB网卡的IP地址，默认值 “192.168.1.2”。

步骤6 执行制卡脚本。

1. 以root用户执行如下命令查找SD卡所在的USB设备名称。

```
fdisk -l
```

例如，SD卡所在USB设备名称为 “/dev/sda”，可通过插拔SD卡的方式确定设备名称。

2. 运行SD制卡脚本 “make_sd_card.py”。

```
python3 make_sd_card.py local /dev/sda
```

– “local” 表示使用本地方式制作SD卡。

– “/dev/sda” 为SD卡所在的USB设备名称。

如**图5-1**所示表示制卡成功。

图 3-1 SD 制卡回显信息示例

```
root@ascend-HP-ProDesk-600-G4-PC1-MT:/home/ascend/mkcard: python3 make_sd_card.py local /dev/sda
Begin to make SD Card...
Please make sure you have installed dependency packages:
  apt-get install -y qemu-user-static binfmt-support gcc-aarch64-linux-gnu g++-aarch64-linux-gnu
Please input Y to continue, other to install them:Y
Step: Start to make SD Card. It need some time, please wait...
Command:
bash /home/ascend/mkcard/make_ubuntu_sd.sh /dev/sda /home/ascend/mkcard/ubuntu-18.04.4-server-arm64.iso 192.168.0.2 192.168.1.2 > /home/ascend/mkcard/sd_card_making_log/make_ubuntu_sd.log
Make SD Card successfully!
```

说明

如果制卡失败，可以查看当前目录下的sd_card_making_log文件夹下的日志文件进行分析。

步骤7 制卡成功后，将SD卡从读卡器取出并插入QA 200 RC开发者板卡槽。

步骤8 上电QA 200 RC开发者板。

须知

- QA 200 DK制卡后，首次上电启动过程中会进行固件的升级，升级完成后会自动进行重启的操作，重启后再进行其他组件的安装。
- 首次启动QA 200 RC开发者板时不能断电，以免对QA 200 RC开发者板造成损害，再次上电需与上次下电时间保持2S以上的安全时间间隔。

QA 200 RC开发者板的上电操作及上电后指示灯的状态说明，请参见[12.4 上电Atlas 200 DK开发者板](#)。

----结束

后续处理

您可以通过npusmi工具查询、配置芯片相关信息。

npusmi工具存储在“/usr/local/sbin”目录下，工具的使用方法可参见《[Atlas 200 AI加速模块 1.0.11 npusmi 命令参考](#)》

异常处理

若上电后，开发者板无法正常启动（指示灯状态异常），可通过如下步骤查看相关日志：

步骤1 将QA 200 RC下电。

步骤2 从QA 200 RC中拔出SD卡，将SD卡插入读卡器，并将读卡器与用户PC机的USB接口连接。

步骤3 以root用户执行如下命令查找SD卡所在的USB设备及其分区信息。

fdisk -l

显示信息如下所示：

```
Disk /dev/sda: 29.7 GiB, 31914983424 bytes, 62333952 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: dos
Disk identifier: 0x00000000
```

Device	Boot Start	End Sectors	Size	Id	Type
/dev/sda1	2048	10487807	10485760	5G 83	Linux
/dev/sda2	10487808	12584959	2097152	1G 83	Linux
/dev/sda3	12584960	62333951	49748992	23.7G 83	Linux

步骤4 将SD卡的第一个分区挂载到用户PC机。

1. 使用root用户，创建一个空目录。

例如：

```
mkdir -p /home/sdinfo
```

2. 将/dev/sda1挂载到/home/sdinfo目录。

```
mount /dev/sda1 /home/sdinfo
```

步骤5 进入/home/sdinfo目录，即进入QA 200 RC的文件系统，可在此目录的var/log/ascend_seclog路径下查看相关日志文件。

```
cd /home/sdinfo
```

```
cd var/log/ascend_seclog/
```

可查看此路径的日志文件，日志文件描述如下所示：

- operation.log: 操作日志，记录安装或升级等事件的结果信息，通过此日志可看出安装升级的最终结果是成功还是失败。
日志格式：事件类型+事件级别+用户ID+日期+发起端地址+访问文件名+操作指令+执行结果
- ascend_install.log: 安装/升级详细日志，记录安装/升级的详细运维脚本日志，可通过此日志查看安装/升级的详细操作及状态信息。
日志格式：组件+日期+日志级别+日志内容
- ascend_run_servers.log: QA 200 RC启动信息。

须知

- 若开发者无法解决问题，在[Ascend论坛](#)上发求助帖，并将以上日志信息作为帖子的附件上传，将会有华为工程师为您解答。
- 若QA 200 RC启动三次及以上后均失败，解决问题后，需要删除var/log/ascend_seclog/目录下的boot_fail_count文件，否则QA 200 RC无法正常启动。

----结束

后续处理

您可以通过npu-smi工具查询、配置芯片相关信息。

npu-smi工具存储在“/usr/local/sbin”目录下，工具的使用方法可参见《[Atlas 200 AI加速模块 1.0.11 npu-smi 命令参考](#)》

异常处理

若上电后，开发者板无法正常启动（例如，指示灯状态异常），可通过如下步骤查看相关日志：

步骤1 将QA 200 RC下电。

步骤2 使用跳线帽/杜邦线短接QA 200 RC的16与18号针脚，如[硬件准备](#)所示，将QA 200 RC 作为USB设备。

步骤3 将QA 200 RC通过USB端口与用户PC机链接，并上电QA 200 RC。

步骤4 以root用户执行如下命令查找SD卡所在的USB设备及其分区信息。

fdisk -l

显示信息如下所示：

```
Disk /dev/sda: 29.7 GiB, 31914983424 bytes, 62333952 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: dos
Disk identifier: 0x00000000

Device     Boot Start      End Sectors Size Id Type
/dev/sda1             2048 10487807 10485760 5G 83 Linux
/dev/sda2    10487808 12584959 2097152 1G 83 Linux
/dev/sda3    12584960 62333951 49748992 23.7G 83 Linux
```

步骤5 将SD卡的第一个分区挂载到用户PC机。

1. 使用root用户，创建一个空目录。

例如：

mkdir -p /home/sdinfo

2. 将/dev/sda1挂载到/home/sdinfo目录。

mount /dev/sda1 /home/sdinfo

步骤6 进入/home/sdinfo目录，即进入QA 200 RC的文件系统，可在此目录的var/log/ascend_seclog路径下查看相关日志文件。

```
cd /home/sdinfo
```

```
cd var/log/ascend_seclog/
```

可查看此路径的日志文件，日志文件描述如下所示：

- operation.log: 操作日志，记录安装或升级等事件的结果信息，通过此日志可看出安装升级的最终结果是成功还是失败。
日志格式：事件类型+事件级别+用户ID+日期+发起端地址+访问文件名+操作指令+执行结果
- ascend_install.log: 安装/升级详细日志，记录安装/升级的详细运维脚本日志，可通过此日志查看安装/升级的详细操作及状态信息。
日志格式：组件+日期+日志级别+日志内容
- ascend_run_servers.log: QA 200 RC启动信息。

须知

- 若开发者无法解决问题，在[Ascend论坛](#)上发求助帖，并将以上日志信息作为帖子的附件上传，将会有华为工程师为您解答。
- 若QA 200 RC启动三次及以上后均失败，解决问题后，需要删除var/log/ascend_seclog/目录下的boot_fail_count文件，否则QA 200 RC无法正常启动。

-----结束

4 配置网络连接

前提条件

进行网络配置前，需要确保QA 200 RC已完成操作系统及驱动的安装（通过制作SD卡的方式进行安装），且已正常启动。

场景描述

QA 200 RC提供了USB网卡和NIC网卡，其中USB网卡的默认IP地址为“192.168.1.2”，NIC网卡的默认IP地址为“192.168.0.2”，用户可以通过这两个网卡实现与PC机的通信，并把QA 200 RC接入互联网。

QA 200 RC的网络连接配置有以下两种方式，您可以自行选择方便的联网方式。

- **通过NIC接口连接路由器**

使用USB方式将QA 200 RC连接用户PC机，使得可以在PC机上SSH登录到QA 200 DK；使用网线将QA 200 DK接入路由器的网口，通过配置NIC网卡将QA 200 DK接入互联网。

- **通过网络共享方式接入互联网**

分别使用USB连接线及网线将QA 200 RC与PC机连接，通过网络配置使得QA 200 DK可以共享PC机网络，从而接入互联网。

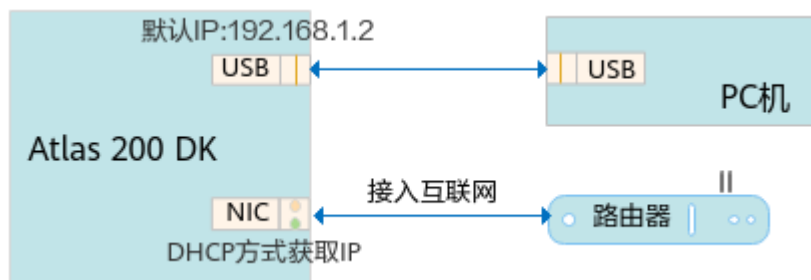
说明

若您的QA 200 RC无条件接入互联网，您可以参见 [12.11 连接QA 200 RC与用户PC机](#)，仅实现QA 200 RC与PC机之间的通信，需要注意此种场景下，QA 200 RC仅适合作为运行环境，应用的开发及编译需要在开发环境（用户PC机）上进行。

通过 NIC 接口连接路由器

此种场景下，首先使用USB连线将QA 200 RC与PC机进行连接，将二者的USB网卡IP地址配置在同一网段，从而可以在PC机上SSH登录到QA 200 RC；然后将Atlas 200 DK通过网线接入路由器，并配置QA 200 RC的IP地址获取方式为DHCP，使其可以接入互联网，配置示例如下图所示：

图 4-1 通过接入路由器联网的配置示例



步骤1 使用USB连接线将QA 200 RC与PC机连接，并参考本步骤配置PC机的USB网卡的IP地址，使其可与QA 200 RC通信。

QA 200 RC的USB网卡的默认IP地址为192.168.1.2，所以需要修改PC机的USB网卡的IP地址为192.168.1.x（x取值范围为0~1，3~254），使PC机可以和QA 200 RC通信。

须知

- 若制作SD卡时，您已修改QA 200 RC开发者板的USB网卡地址与PC机的USB网卡的IP地址在同一网段，则本步骤可直接跳过。
- 若您的Ubuntu操作系统是通过Windows主机上的虚拟机进行安装的场景，需要参考 [12.10 如何安装windows的USB网卡驱动](#)在Windows主机上安装USB网卡驱动，否则虚拟机上的Ubuntu系统无法识别QA 200 RC的USB网卡。

修改PC机的USB网卡IP有通过脚本配置和手工配置两种方式，您可以自行选择配置方式：

- 通过脚本配置
 - a. 执行如下命令下载configure_usb_ethernet.sh脚本，并将其上传到PC机的任一目录，例如/home/ascend/config_usb_ip/。
从gitee下载：
wget https://gitee.com/ascend/tools/raw/master/configure_usb_ethernet/for_20.1/configure_usb_ethernet.sh
从github下载：
wget https://raw.githubusercontent.com/Huawei-Ascend/tools/master/configure_usb_ethernet/for_20.1/configure_usb_ethernet.sh

须知

通过脚本配置仅针对首次配置USB网卡对应IP地址的场景。USB网卡IP已经配置，若需要修改其IP地址，请参考 [手工配置](#)修改USB网卡的IP地址。

- b. 以root用户进入PC机中USB网卡IP配置脚本所在目录，例如/home/ascend/config_usb_ip。
- c. 执行如下命令进行USB网卡IP地址的配置。
bash configure_usb_ethernet.sh -s ip_address

以指定的IP地址配置PC机中USB网卡的IP地址，如果直接执行**bash configure_usb_ethernet.sh**，则以默认IP地址“192.168.1.166”进行配置。

- 如果存在多个USB网卡，则首先执行**ifconfig**命令查询USB网卡名称（若系统中有多个USB网卡，可以通过拔插开发者板进行判定，PC机会将QA 200 RC开发者板识别为虚拟USB网卡），再执行如下命令配置指定网卡的IP地址。

```
bash configure_usb_ethernet.sh -s usb_nic_name ip_address
```

- **usb_nic_name**: USB网卡名称。

- **ip_address**: 配置的IP地址。

命令示例，配置PC机的USB网卡IP为192.168.1.223:

```
bash configure_usb_ethernet.sh -s enp0s20f0u8 192.168.1.223
```

配置完成后，可执行**ifconfig**命令查看IP是否生效。

- 手工配置

- a. 以普通用户登录PC机，执行如下命令切换到root用户。

```
su - root
```

- b. 获取USB网卡名

```
ifconfig -a
```

若系统中有多个USB网卡，可以通过拔插开发者板进行判定。

- c. 在“/etc/netplan”目录下后缀为“.yaml”的文件中添加USB网卡的静态IP。

若“/etc/netplan”目录中不存在后缀为“.yaml”的文件，则需要自行创建，此处以文件名为“01-netcfg.yaml”为例进行配置描述。

执行如下命令打开网络配置文件：

```
vi /etc/netplan/01-netcfg.yaml
```

在ethernets层下增加USB网卡网络配置，例如USB网卡名为enp0s20f0u4，配置静态IP为192.168.1.223，配置方法如下：

```
ethernets:
  ...
  enp0s20f0u4:
    dhcp4: no
    addresses: [192.168.1.223/24]
    gateway4: 192.168.0.1
    nameservers:
      addresses: [255.255.0.0]
```

输入:**wq**保存退出。

- d. 执行如下命令重启网络服务。

```
netplan apply
```

重启后执行**ifconfig**命令，可查看USB网卡enp0s20f0u4的IP地址是否生效。

步骤2 将QA 200 RC通过网线接入路由器，并开启路由器的DHCP功能。

步骤3 配置QA 200 RC的NIC网卡IP地址获取方式为DHCP。

1. 在PC机中，以HwHiAiUser用户SSH登录到QA 200 RC开发者板。以默认USB网卡的IP地址192.168.1.2为例，执行如下命令登录：**ssh HwHiAiUser@192.168.1.2**
用户HwHiAiUser缺省登录密码为“Mind@123”。
2. 修改登录密码。
首次登录时，系统将提示密码已过期：


```
WARNING:Your password has expired.
```

此时，必须修改密码并重新登录。请参见[HwHiAiUser用户密码修改](#)进行密码修改，修改完成后，系统会强制退出，并出现如下提示：

```
passwd: password updated successfully
Connection to 192.168.1.2 closed.
```

请使用修改后的密码，执行如下命令重新登录：

```
ssh HwHiAiUser@192.168.1.2
```

3. 登录后，执行如下命令，切换到root用户：

```
su - root
```

此时系统会强制用户更改密码，请参见[root用户密码修改](#)进行密码修改。

4. 执行如下命令，打开网络配置文件。

```
vi /etc/netplan/01-netcfg.yaml
```

5. 修改eth0网卡的IP地址获取方式为DHCP。

把eth0的配置修改为如下内容：

```
eth0:
  dhcp4: true
  addresses: []
  optional: true
```

6. 保存退出。

```
:wq
```

步骤4 执行如下命令重启网络服务。

```
netplan apply
```

至此，您可以在QA 200 RC开发者板上访问网络了。

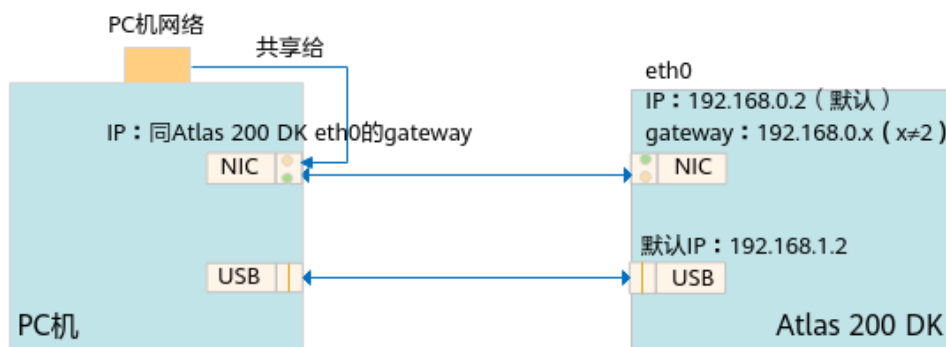
步骤5 执行ifconfig命令获取eth0网卡的IP地址，您可以使用此IP地址与PC机通信，当然，您也可以继续使用USB网卡的IP地址与PC机通信。

-----结束

通过网络共享方式接入互联网

此种场景下，QA 200 RC与PC机分别通过USB连线与网线进行连接，其中USB方式的连接，用于将PC机与QA 200 RC进行通信，从而可以从PC机上SSH登录到Atlas 200 DK；网线连接用于QA 200 RC共享PC机网络，QA 200 RC eth0的网关地址需要与PC机中与Atlas 200 DK相连的NIC网卡的IP地址保持一致，将PC机上网络共享给连接QA 200 RC的NIC网卡，则QA 200 RC即可接入互联网。配置示例如下图所示：图

4-2 共享 PC 机网络方式配置示例



步骤1 分别使用USB连接线及网线将QA 200 RC与PC机连接。

PC机中的“网络连接”显示如下图所示：

图 4-3 网络连接图示



- “Local connection（本地连接）”：USB虚拟网卡，此连接为基本连接，用于从PC侧SSH登录到QA 200 RC。
- “Ethernet2（以太网2）”：通过网线连接QA 200 RC的NIC网卡，后续将使用此网卡作为网关，用于QA 200 RC接入互联网。

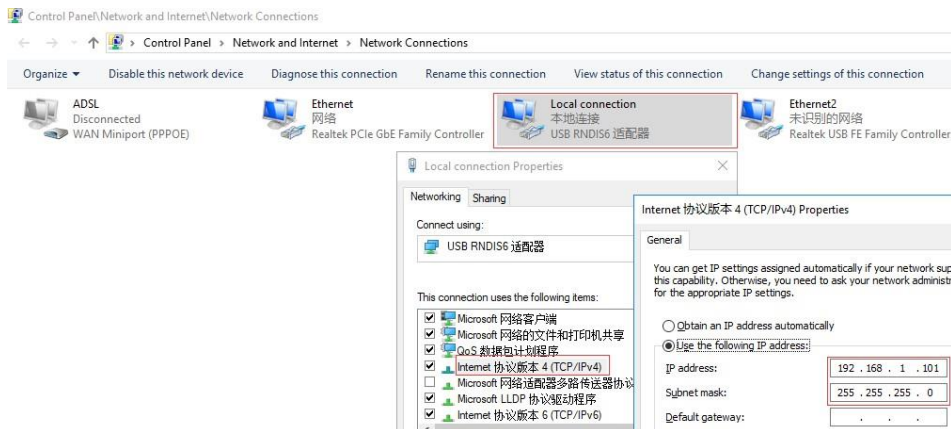
说明：以上名字仅为示例，您可以通过拔插网口的方式查看连接QA 200 RC的是哪一个网卡。

步骤2 在PC机上配置USB虚拟网卡，使其可以通过SSH方式登录QA 200 RC。

配置USB虚拟网卡的IP地址，IP地址需要与QA 200 RC的USB网卡的默认IP地址在同一网段。

QA 200 RC的USB网卡默认IP为“192.168.1.2”，例如配置PC机的USB虚拟网卡的IP地址为“192.168.1.101”，如下图所示：

图 4-4 修改 PC 机侧 USB 虚拟网卡 IP 地址示例



配置完成后，执行如下命令测试下是否能够登录到QA 200 RC：

```
ssh HwHiAiUser@192.168.1.2
```

步骤3 配置QA 200 RC的eth0网卡的网关。

QA 200 RC中eth0网卡的网关与eth0 IP地址在同一网段即可，例如QA 200 RC的eth0的默认IP地址为“192.168.0.2”，则此处可以配置eth0的网关为“192.168.0.101”。

说明：QA 200 RC通过网络共享接入网络的方式，数据走eth0，然后通过网关访问网络，所以需要确保eth0的IP地址能够访问网关。

配置方式如下：

1. 在PC机上以SSH方式登录QA 200 RC。

ssh HwHiAiUser@192.168.1.2

2. 切换到root用户。

su - root

3. 配置eth0的网关地址。

vi /etc/netplan/01-netcfg.yaml

配置示例如下图所示：

图 4-5 QA 200 RC eth0 网关配置

```
network:
  version: 2
  renderer: networkd
  ethernets:
    eth0:
      dhcp4: no
      addresses: [192.168.0.2/24]
      gateway4: 192.168.0.101
      nameservers:
        addresses: [8.8.8.8]
        addresses: [114.114.114.114]

    usb0:
      dhcp4: no
      addresses: [192.168.1.2/24]
      #gateway4: 192.168.1.1
      nameservers:
        addresses: [114.114.114.114]
```

保存退出。

注意：域名服务器可以配置为公共DNS “114.114.114.114” 或者 “8.8.8.8” ，
请根据实际环境进行选择配置。

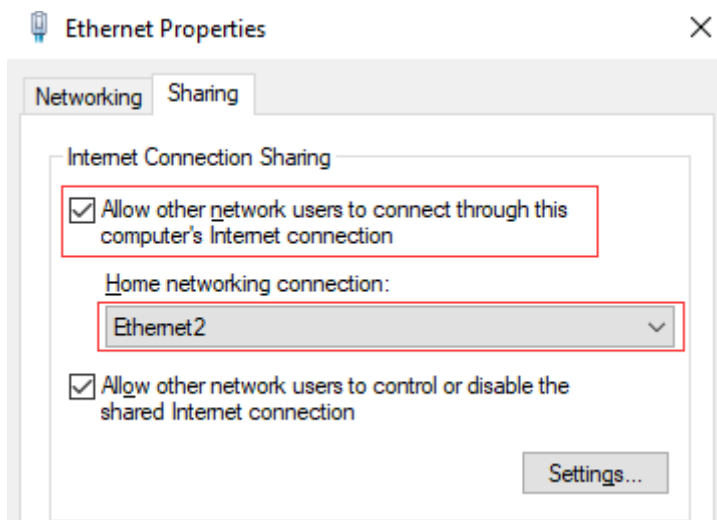
步骤4 在PC侧进行网络配置，将网络共享给QA 200 RC。

1. 将PC机上能够接入互联网的网卡进行网络共享配置，共享给连接QA 200 RC的网卡。

如图4-3所示，能够接入互联网的网卡为“Ethernet”，连接QA 200 RC的网卡为“Ethernet2”。

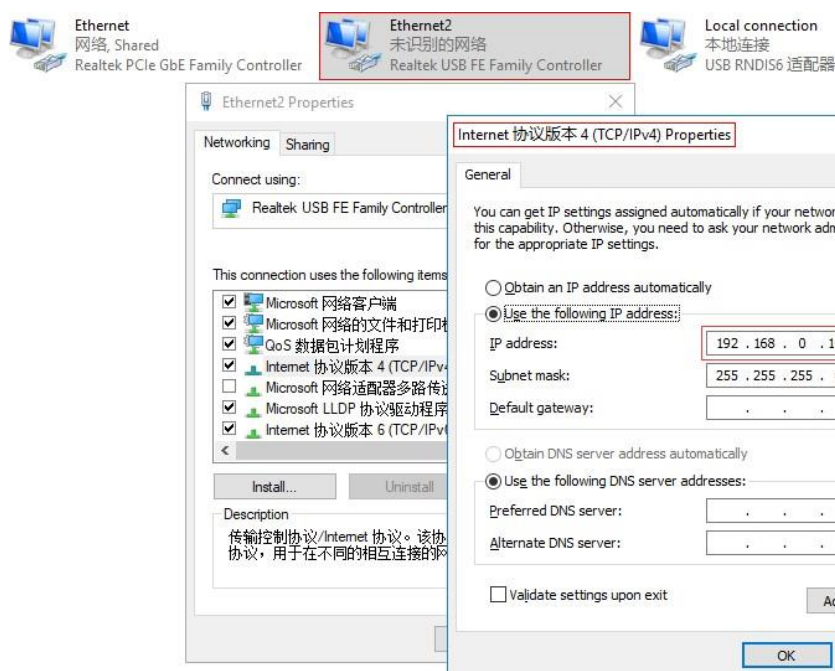
右键单击网卡“Ethernet”，然后选择“Properties > Sharing”，按照下图所示进行配置。

图 4-6 共享网络配置



2. 单击“OK”，保存配置。
 3. 修改“Ethernet2”的IP地址为QA 200 RC eth0网关的IP地址（例如，图4-5中配置的“192.168.0.101”）。
- 修改方法如图4-7所示，至此，您的QA 200 RC已可以接入互联网。图

4-7 Ethernet2 IP 地址修改



说明

通过网络共享方式使得QA 200 RC接入网络的原理如下：

- QA 200 RC中eth0的IP地址“192.168.0.2”与网关“192.168.0.101”是互通的。
- PC机上可以接入互联网的“Ethernet”网卡将网络共享给了网关“192.168.0.101”，则该网关可以接入互联网。
- QA 200 RC需要联网时，通过域名解析，会自动选择请求的出口到网关，便可以接入互联网了。

----结束

5 修改用户密码

密码修改策略

环境搭建完成后，请修改QA 200 RC操作系统中用户的密码，以提升系统安全性。操作系统用户的密码复杂度建议满足如下要求：

- 至少包含如下两种字符的组合：
 - 至少一个小写字母。
 - 至少一个大写字母。
 - 至少一个数字。
 - 至少一个特殊字符：`~!@#\$%^&*()-_+=\|[];:~",<.>/?和空格。
- 长度至少8个字符。
- 不能与帐号或者帐号的倒写相同。

HwHiAiUser 用户密码修改

HwHiAiUser用户为制作SD卡时创建的默认用户，此用户的默认密码是“**Mind@123**”。QA 200 RC开发者板与PC机连接成功后需要修改QA 200 RC开发者板中HwHiAiUser用户的初始密码。

1. 在PC机中以HwHiAiUser用户SSH登录到QA 200 RC开发者板。

说明

若以SSH方式登录QA 200 RC开发者板时无法建立信任关系，请参考[13.2 SSH登录时提示无法建立信任关系](#)。

2. 执行**passwd**命令修改HwHiAiUser密码。
如[图5-1](#)所示。

图 5-1 修改 HwHiAiUser 用户密码

```
ascend@ascend-HP-ProDesk-600-G4-PCI-MT:~/tools/bin$ ssh HwHiAiUser@192.168.1.2
HwHiAiUser@192.168.1.2's password:
Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.1.46+ aarch64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage
Last login: Tue Jan 1 00:39:34 2019 from 192.168.1.223
$ passwd
Changing password for HwHiAiUser.
(current) UNIX password:
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
```

root 用户密码修改

1. 在PC机中以HwHiAiUser用户SSH登录QA 200 RC开发者板。

📖 说明

用户HwHiAiUser缺省登录密码为Mind@123。

2. 执行如下命令切换到root用户。

```
su - root
```

📖 说明

root用户的缺省登录密码为Mind@123。

3. 执行passwd命令修改root用户的密码，如图5-2所示。

图 5-2 修改 root 用户密码

```
root@davinci-mini:~# passwd
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
```

6 安装 CANN 软件

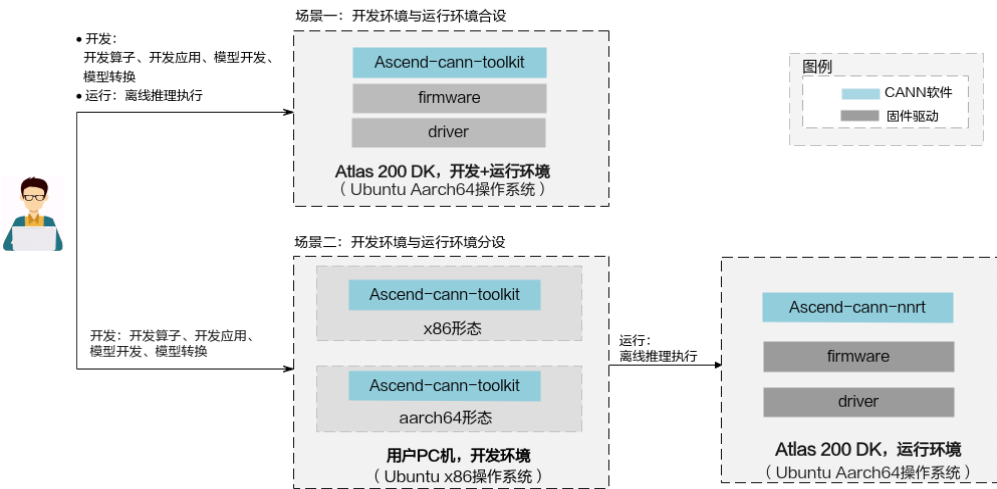
- 8.1 场景描述
- 8.2 开发环境与运行环境合设（推荐）
- 8.3 开发环境与运行环境分设

场景描述

Ascend CANN（Compute Architecture for Neural Networks）解决方案是面向开发者的AI计算平台，包含计算资源、运行框架以及相关配套工具等，让开发者可以便捷高效的编写在特定硬件设备上运行的人工智能应用程序。

针对QA 200 RC，常见的CANN软件部署场景有下图所示的两种：

图 6-1 QA 200 RC CANN 软件部署场景



- 场景一，开发环境与运行环境合设（推荐使用此方式）：
此种场景下，QA200 RC既作为开发环境，又作为运行环境。可进行算子开发、应

用开发、模型开发及模型转换等功能的开发、编译，以及离线推理应用程序的执行。

此种场景下，QA 200 RC需要接入互联网，详情操作方法可参见[4 配置网络连接](#)。

- 场景二，开发环境与运行环境分设：
 - 开发环境：安装了Ubuntu x86操作系统的用户PC机。进行算子开发、应用开发、模型开发及模型转换等功能的开发及编译。
开发者可以直接使用制作SD卡的用户PC机作为开发环境，若使用的开发环境不是制作SD卡的PC机，进行CANN软件的安装前，需要参见[4 配置网络连接](#)将QA 200 RC与开发环境所在PC机进行连接。
 - 运行环境：QA 200 RC。将开发环境编译得到的应用程序可执行文件及依赖文件上传到QA 200 RC侧，进行离线推理的执行。

开发环境与运行环境合设（推荐）

此种场景下，QA 200 RC既作为开发环境，又作为运行环境。下面介绍此种场景下的CANN软件安装流程。

须知

此种场景下QA 200 RC需要接入互联网，接入方法可参见[6 配置网络连接](#)。

步骤1 准备软件包。

请参见[下载链接](#)下载配套驱动版本的开发套件包“Ascend-cann-toolkit_{version}_linux-aarch64.run”。

驱动与CANN版本的配套关系请参见[版本配套说明（Gitee）](#)或[版本配套说明（Github）](#)。

步骤2 准备参考资料。

请从[开发者文档](#)，获取配套版本的《CANN软件安装指南》。

步骤3 准备安装及运行用户。

制作SD卡时，已经在QA 200 RC上创建了用于运行应用程序的HwHiAiUser用户，请直接使用此用户作为CANN软件的安装运行用户。

步骤4 安装OS依赖。

请参见《CANN软件安装指南》的“安装开发环境 > 安装OS依赖 > 安装步骤（Ubuntu 18.04）”进行操作系统源的配置、依赖的安装。

步骤5 安装开发套件包。

1. 以HwHiAiUser用户将开发套件包上传到QA 200 RC任意目录。
2. 执行如下命令为安装包增加可执行权限。
chmod +x *.run
3. 执行如下校验安装包的一致性和完整性。
./Ascend-cann-toolkit_{version}_linux-aarch64.run --check
4. 执行如下命令进行Toolkit软件包的安装。
./Ascend-cann-toolkit_{version}_linux-aarch64.run --install --chip=Ascend310-minirc

`--chip=Ascend310-minirc`（可选）：指定芯片型号为Ascend310 Soc芯片（RC模式启动，作为主控CPU）。

配置了此参数，则会只部署Ascend310RC形态的AI CPU软件包。

以上命令为使用默认路径进行安装的示例，默认安装路径为“\$HOME/Ascend”。

步骤6 配置环境变量。

CANN软件提供进程级环境变量设置脚本，供用户在进程中引用，以自动完成环境变量设置。用户进程结束后自动失效。示例如下（以HwHiAiUser用户默认安装路径为例）：

```
./home/HwHiAiUser/Ascend/ascend-toolkit/set_env.sh
```

用户也可以通过修改`~/.bashrc`文件方式设置永久环境变量，操作如下：

1. 以运行用户在任意目录下执行`vi ~/.bashrc`命令，打开`.bashrc`文件，在文件最后一行后面添加上述内容。
2. 执行`:wq!`命令保存文件并退出。
3. 执行`source ~/.bashrc`命令使其立即生效。

-----结束

开发环境与运行环境分设

简介

此种场景下，用户PC机（Ubuntu x86操作系统）作为开发环境，进行程序的开发、编译，QA 200 RC作为运行环境。

下面分别介绍开发环境与运行环境中CANN软件安装流程。

开发环境中 CANN 软件安装

步骤1 准备软件包。

请参见[下载链接](#)下载配套驱动版本的开发套件包“Ascend-cann-toolkit_{version}_linux-x86_64.run”与“Ascend-cann-toolkit_{version}_linux-aarch64.run”。

驱动与CANN版本的配套关系请参见[版本配套说明（Gitee）](#)或[版本配套说明（Github）](#)。

步骤2 准备参考资料。

请从[开发者文档](#)，获取配套版本的《CANN软件安装指南》。

步骤3 准备安装及运行用户。

请参见《CANN软件安装指南》的“安装开发环境 > 准备安装及运行用户”准备需要进行CANN软件安装的用户。

步骤4 安装OS依赖。

请参见《CANN软件安装指南》的“安装开发环境 > 安装OS依赖 > 安装步骤（Ubuntu 18.04）”进行操作系统源的配置、依赖的安装。

步骤5 安装开发套件包。

1. 以安装用户将开发套件包上传到开发环境任意路径，并进入套件包所在路径。
2. 执行如下命令为安装包增加可执行权限。

```
chmod +x *.run
```

3. 执行如下校验安装包的一致性和完整性。

```
./Ascend-cann-toolkit_{version}_linux-x86_64.run --check  
./Ascend-cann-toolkit_{version}_linux-aarch64.run --check
```

4. 执行以下命令安装软件。

```
./Ascend-cann-toolkit_{version}_linux-x86_64.run --install --chip=Ascend310-minirc  
./Ascend-cann-toolkit_{version}_linux-aarch64.run --install --chip=Ascend310-minirc
```

--chip=Ascend310-minirc（可选）：指定芯片型号为Ascend310 Soc芯片（RC模式启动，作为主控CPU）。配置了此参数，则会只部署Ascend310RC形态的AI CPU软件包。

以上命令示例使用默认路径进行安装，若安装用户为root用户，则默认路径为“/usr/local/Ascend”；若安装用户为非root用户，则默认路径为“\$HOME/Ascend”。

开发者也可以通过--install-path=<path>指定安装路径，安装支持的其他详细参数说明可参见《CANN软件安装指南》的“参考信息 > 参数说明”。

步骤6 配置交叉编译环境。

分设场景下，开发环境架构为“x86_64”，运行环境架构为“aarch64”，所以在开发环境中安装交叉编译器进行应用程序的交叉编译。

请使用CANN软件包安装用户，在开发环境执行[aarch64-linux-gnu-g++ --version](#)命令检查交叉编译器是否安装，若已经安装则可以忽略；否则请参考以下命令进行安装（以下命令仅为示例，请用户根据实际情况替换）：

```
sudo apt-get install g++-aarch64-linux-gnu
```

步骤7 配置环境变量。

CANN软件提供进程级环境变量设置脚本，供用户在进程中引用，以自动完成环境变量设置。用户进程结束后自动失效。示例如下（以HwHiAiUser用户默认安装路径为例）：

```
./home/HwHiAiUser/Ascend/ascend-toolkit/set_env.sh
```

用户也可以通过修改~/.bashrc文件方式设置永久环境变量，操作如下：

1. 以运行用户在任意目录下执行[vi ~/.bashrc](#)命令，打开.bashrc文件，在文件最后一行后面添加上述内容。
2. 执行:wq!命令保存文件并退出。
3. 执行source ~/.bashrc命令使其立即生效。

----结束

运行环境中 CANN 软件安装

步骤1 准备软件包。

请参见[下载链接](#)下载配套驱动版本的离线推理引擎包“Ascend-cann-nnrt_{software version}_linux-aarch64.run”。

驱动与CANN版本的配套关系请参见[版本配套说明（Gitee）](#)或[版本配套说明（Github）](#)。

步骤2 在QA 200 RC上安装离线推理引擎包。

请以制卡时创建的**HwHiAiUser**用户作为QA 200 RC上CANN软件的安装用户，安装步骤请参见《CANN软件安装指南》的“安装运行环境（nnrt软件，在物理机/虚拟机安装）> 安装离线推理引擎包”。

1. 以HwHiAiUser用户将离线推理引擎包上传到QA 200 RC任意目录。
2. 执行如下命令为安装包增加可执行权限。

```
chmod +x *.run
```

其中*.run表示软件包名，请根据实际包名进行替换。

3. 执行如下校验安装包的一致性和完整性。

```
./Ascend-cann-nnrt_{software version}_linux-aarch64.run --check
```

4. 执行如下命令进行离线推理引擎包的安装。

```
./Ascend-cann-nnrt_{software version}_linux-aarch64.run --install --  
chip=Ascend310-minirc
```

--chip=Ascend310-minirc（可选）：指定芯片型号为Ascend310 Soc芯片（RC模式启动，作为主控CPU）。

配置了此参数，则会只部署Ascend310RC形态的AI CPU软件包。

以上命令为使用默认路径进行安装的示例，默认安装路径为“\$HOME/Ascend”，安装后的CANN软件存储在“\$HOME/Ascend/nnrt/latest”中。

步骤3 配置环境变量。

CANN软件提供进程级环境变量设置脚本，供用户在进程中引用，以自动完成环境变量设置。用户进程结束后自动失效。示例如下（以HwHiAiUser用户默认安装路径为例）：

```
./home/HwHiAiUser/Ascend/nnrt/set_env.sh
```

用户也可以通过修改~/.bashrc文件方式设置永久环境变量，操作如下：

1. 以运行用户在任意目录下执行**vi ~/.bashrc**命令，打开.bashrc文件，在文件最后一行后面添加上述内容。
2. 执行**:wq!**命令保存文件并退出。
3. 执行**source ~/.bashrc**命令使其立即生效。

----结束

7 准备编译运行依赖文件

简介

若开发者需要使用QA 200 RC外接的摄像头进行数据采集，并将采集的数据作为应用的数据源，则应用程序开发编译时需要依赖媒体控制模块相关的头文件与库文件，如下所示：

- `peripheral_api.h`: 媒体控制模块的头文件，详细的接口使用说明可参见《[Atlas 200 DK Media API参考](#)》。

须知

- Media的API是基于C语言的接口。
- 若使用的RASPBerry PI V2.1型号的摄像头，摄像头支持设置的帧率范围为：1~20fps。
- 若使用的RASPBerry PI V1.3型号的摄像头，摄像头支持设置的帧率范围为：1~15fps。
- `libmedia_mini.so`: 媒体控制模块的库文件。

在进行媒体应用程序的编译时，`libmedia_mini.so`库文件还依赖如下日志、安全、文件操作等功能的公共库文件：

`libslog.so`，`libc_sec.so`，`libmmpa.so`

为了开发者能够方便的运行gitee上的应用样例，建议开发者按照如下参考，将相关头文件与库文件拷贝到推荐路径下。

开发环境与运行环境合设场景

此种场景下，直接在QA 200 RC上进行应用程序的开发与编译。

制作SD卡时，已在QA 200 RC上部署好了媒体相关应用程序编译依赖的库文件与头文件，为了能直接运行gitee上相关样例，建议按照以下步骤将相关文件拷贝到QA 200 RC上的指定路径下。

步骤1 设置环境变量“`INSTALL_DIR`”，指定CANN软件安装后文件存储路径。

例如CANN软件使用默认路径进行安装，则安装后文件存储路径为“`${HOME}/Ascend/nnrt/latest`”，环境变量设置示例如下。

```
export INSTALL_DIR=${HOME}/Ascend/nnrt/latest
```

步骤2 在\${INSTALL_DIR}路径下创建“driver”目录，用于存储媒体应用程序编译依赖的相关文件。

```
mkdir -p ${INSTALL_DIR}/driver
```

步骤3 将媒体应用程序编译依赖的相关文件拷贝到\${INSTALL_DIR}/driver目录下。

```
cp /usr/lib64/libmedia_mini.so ${INSTALL_DIR}/driver/  
cp /usr/lib64/libsllog.so ${INSTALL_DIR}/driver/  
cp /usr/lib64/libc_sec.so ${INSTALL_DIR}/driver/  
cp /usr/lib64/libmmpa.so ${INSTALL_DIR}/driver/  
cp /usr/local/Ascend/include/peripheral_api.h ${INSTALL_DIR}/driver/
```

----结束

开发环境与运行环境分设场景

此种场景下，开发者使用安装了Ubuntu x86操作系统的PC机作为开发环境，需要参考如下步骤在开发环境上手工部署媒体应用程序编译依赖的头文件与库文件。

步骤1 在开发环境中设置环境变量“INSTALL_DIR”，指定CANN软件安装后文件存储路径。

例如CANN软件使用默认路径安装，则安装后文件存储路径为“\${HOME}/Ascend/ascend-toolkit/latest”，环境变量设置示例如下。

```
export INSTALL_DIR=${HOME}/Ascend/ascend-toolkit/latest
```

步骤2 在\${INSTALL_DIR}路径下创建“driver”目录，用于存储媒体应用程序编译依赖的相关文件。

```
mkdir -p ${INSTALL_DIR}/driver
```

步骤3 将媒体应用程序编译依赖的相关文件从板端环境中拷贝到开发环境的\${INSTALL_DIR}/driver目录下。

```
sudo scp -r HwHiAiUser@X.X.X.X:/usr/lib64/libmedia_mini.so ${INSTALL_DIR}/driver/  
sudo scp -r HwHiAiUser@X.X.X.X:/usr/lib64/libsllog.so ${INSTALL_DIR}/driver/  
sudo scp -r HwHiAiUser@X.X.X.X:/usr/lib64/libc_sec.so ${INSTALL_DIR}/driver/  
sudo scp -r HwHiAiUser@X.X.X.X:/usr/lib64/libmmpa.so ${INSTALL_DIR}/driver/  
sudo scp -r HwHiAiUser@X.X.X.X:/usr/local/Ascend/include/peripheral_api.h ${INSTALL_DIR}/driver/
```

----结束

8 运行首个应用

运行环境与开发环境部署完成后，您可以快速运行首个AI应用程序验证环境部署的正确性。

- 若您基于后台命令行进行开发，您可以参见配套CANN版本的应用开发资料中的“AscendCL样例使用指导>基于Caffe ResNet-50网络实现图片分类（同步推理）”章节，快速进行样例的编译运行。

说明

关于QA 200 RC的更多应用案例，可参见<https://gitee.com/ascend/samples>或<https://github.com/Ascend/samples/tree/master>。