# Atlas 200 DK

# 环境部署指南

**文档版本** 01

发布日期 2023-01-19





# 版权所有 © 华为技术有限公司 2023。 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

nuawe和其他华为商标均为华为技术有限公司的商标。 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、服务或特性等应受华为公司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,华为公司对本文档内容不做任何明示或暗示的声明或保证。

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。

# 目录

1 简介	1
2 安装部署流程	2
3 准备配件及 PC 机	4
4 硬件安装	6
4.1 拆卸上盖	6
4.2 安装摄像头(主板为 IT21DMDA)	8
4.3 安装摄像头(主板为 IT21VDMB)	12
5 制作 SD 卡	18
5.1 简介	18
5.2 读卡器场景	18
5.3 无读卡器场景	22
6 配置网络连接	28
7 修改用户密码	36
8 安装 CANN 软件	38
8.1 场景描述	38
8.2 开发环境与运行环境合设(推荐)	39
8.3 开发环境与运行环境分设	40
9 准备编译运行依赖文件	43
10 运行首个应用	45
11 使用 Memory Cgroup 机制实现业务内存资源控制	46
12 常用操作	48
12.1 升级 Atlas 200 DK 固件与驱动	48
12.2 升级 CANN 软件	50
12.3 查看 Atlas 200 DK 固件与驱动版本	51
12.4 上电 Atlas 200 DK 开发者板	52
12.5 下电 Atlas 200 DK 开发者板	53
12.6 通过串口连接 Atlas 200 DK	53
12.7 查看开发者板硬件底板的版本	54
12.8 查看 Atlas 200 AI 加速模块的版本	57

小児の名 1 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日	日 氷
12.9 如何查看摄像头所属 Channel	61
12.10 如何安装 windows 的 USB 网卡驱动	62
12.11 连接 Atlas 200 DK 与用户 PC 机	65
12.12 修改 Atlas 200 DK IP 地址	68
12.13 设置用户有效期	68
12.14 相关工具	69
13 FAQ	73
13.1 SD 卡制作过程中人为拔卡导致系统存在冗余挂载盘	
13.2 SSH 登录时提示无法建立信任关系	73
13.3 开发者板无法正常连接 PC 机如何处理	74
13.4 Driver 升级失败,报"CheckPartitionSpace partition space check failed"	

1 简介

Atlas 200 DK开发者套件 Atlas 200 Developer Kit(以下简称Atlas 200 DK )是以昇腾310 AI处理器为核心的一个开发者板形态产品,为开发者提供一站式开发套件,助力开发者快速进行AI应用程序的开发。

本文主要介绍用户在使用Atlas 200 DK开发、运行AI应用程序前的准备工作,包括硬件的安装,系统启动盘的制作,如何进行Atlas 200 DK与用户PC机的通信连接、如何将Atlas 200 DK接入互联网等。

Atlas 200 DK的系统框图如下所示:

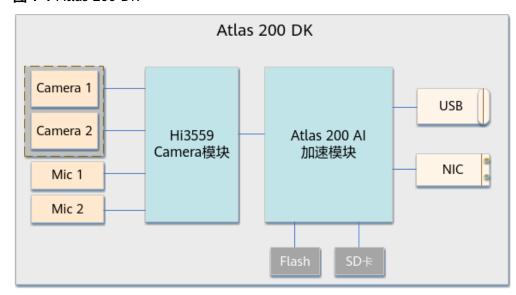


图 1-1 Atlas 200 DK

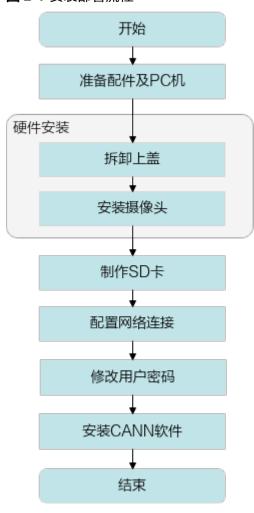
- Atlas 200 AI加速模块是一款高性能的AI智能计算模块,可以实现图像、视频等多种数据分析与推理计算。
- Hi3559 Camera模块为图像/音频接口模块,用于进行图像/音频的采集和处理。
- Atlas 200 DK提供了USB网卡与NIC网卡,开发者可通过USB网卡与用户PC机进行直连通信,可通过NIC网卡将Atlas 200 DK接入互联网。

其中USB网卡的默认IP地址是192.168.1.2, NIC网卡的默认IP地址是192.168.0.2。

# 2 安装部署流程

进行Atlas 200 DK的环境搭建前,请参考图2-1了解环境安装部署的整体流程。

图 2-1 安装部署流程



步骤1 准备Atlas 200 DK正常工作所需配件及一台PC机,其作用及详细要求可参见3 准备配件及PC机。

- 步骤2 进行硬件的安装,包含拆卸上盖及摄像头的安装,请参见4 硬件安装。
- 步骤3 通过制作SD卡的方式安装Atlas 200 DK的操作系统及驱动固件,请参见5 制作SD卡
- 步骤4 配置Atlas 200 DK的网络连接,使得Atlas 200 DK可与用户PC机通信,同时可通过相关配置将Atlas 200 DK接入互联网,请参见6 配置网络连接。
- 步骤5 为提升系统安全性,首次登录Atlas 200 DK请修改操作系统用户密码,请参见7 修改用户密码。
- 步骤6 安装CANN相关软件,部署好开发运行环境,详细请参见8 安装CANN软件。 针对Atlas 200 DK,CANN软件有两种部署场景,如下图所示:

场景一: 开发环境与运行环境合设





场景二: 开发环境与运行环境分设



----结束

# 3 准备配件及 PC 机

介绍使用Atlas 200 DK前需要准备的配件及PC机。

# 配件准备

使用Atlas 200 DK前,需要用户提前自购如表3-1所示配件。

表 3-1 配件信息

配件名称		描述	推荐型号
SD卡		用于制作Atlas 200 DK开发者板启 动系统。	推荐使用经过测试的SD 卡:  E星UHS-I U3 CLASS 10 64G  金士顿UHS-I U1 CLASS 10 64G
制作SD 卡时所 需配	读卡器	使用读卡器制作SD卡的场景,参考 <b>5.2 读卡器场景</b> 。	支持USB3.0协议
件, 器线 料 料 性 是 , <b>一 使 卡</b> <b>一 读</b>	跳线帽杜邦线	通过跳线帽/杜邦线短接开发者板引脚的方式制作SD卡的场景,参考 <b>5.3</b> <b>无读卡器场景</b> 。	跳线帽:间距2.54mm 杜邦线:母对母,间距 2.54mm
Type-C连	妾线	用于与PC机通过USB方式连接,参考6配置网络连接。	支持USB3.0的Type-C连接 线
网线		用于与PC机或路由器通过网线方式 连接,接入互联网,参考6 配置网 络连接。	普通网线,接口类型为 RJ45

配件名称	描述	推荐型号
摄像头	用于与Atlas 200 DK连接获取视 频,参考4.2 安装摄像头(主板为 IT21DMDA)和4.3 安装摄像头 (主板为IT21VDMB)。	推荐使用树莓派摄像头。 型号: RASPBERRY PI V2.1 若Atlas 200 DK主板为 IT21DMDA,使用树莓派摄 像头时需要额外购买黄色的 15 pin树莓派专用排线。
摄像头支架 ( 可选 )	用于固定摄像头,参考 <b>4.2 安装摄像</b> 头(主板为IT21DMDA)和 <b>4.3 安</b> 装摄像头(主板为IT21VDMB)。	树莓派透明摄像头支架
串口线 ( 可 选 )	用于Atlas 200 DK启动灯状态异常或者制卡成功但无法正常与用户PC机通信时,通过串口查看启动日志,参考13.3 开发者板无法正常连接PC机如何处理。	USB转TTL串口线,3.3V接口电平

# 服务器或者 PC 机准备

准备一个操作系统为Ubuntu X86架构的PC机,系统要求如下:

Ubuntu操作系统的版本可以为: 18.04.4、18.04.5,请从Ubuntu官网下载对应版本软件进行安装:

可以下载桌面版"ubuntu-18.04.xx-desktop-**amd64**.iso",或Server版"ubuntu-18.04.xx-server-**amd64**.iso"。

# □ 说明

对于China地区推荐使用华为开源镜像。

- Ubuntu操作系统中需要存在python2.7与python3.x。
- 系统空余空间大于等于20G。
- 系统内存大于等于4G。

# 此PC机的主要作用包含:

- 制作Atlas 200 DK的系统启动盘。
- 连接Atlas 200 DK,以便开发者可以从PC机SSH登录到Atlas 200 DK。
- 将此PC机作为开发环境,用于进行应用程序的开发及编译。

# 4 硬件安装

- 4.1 拆卸上盖
- 4.2 安装摄像头(主板为IT21DMDA)
- 4.3 安装摄像头(主板为IT21VDMB)

# 4.1 拆卸上盖

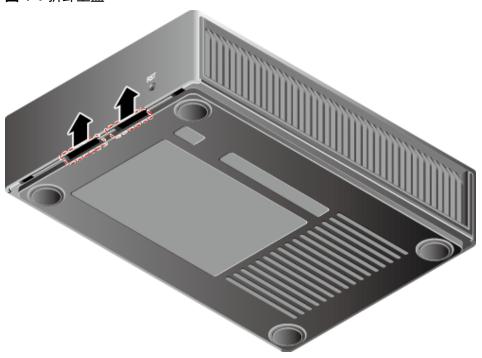
当需要使用摄像头或开发者板引脚时,需要拆卸Atlas 200 DK的上盖,操作步骤如下 所示。

步骤1 查看Atlas 200 DK开发者板上是否有摄像头线缆接出。

- 是,执行步骤3。

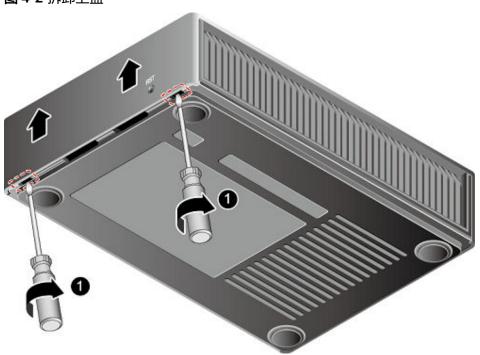
步骤2 当Atlas 200 DK开发者板上没有摄像头线缆接出时,向上按压塑料卡扣,使上盖一端松动,如图4-1所示。

图 4-1 拆卸上盖



步骤3 当Atlas 200 DK开发者板上有摄像头线缆接出时,将一字螺丝刀插入上盖于底板之间的凹槽,旋转螺丝刀,将上盖的一端从盒体上撬出,如图4-2中①所示。

图 4-2 拆卸上盖



步骤4 向上取出上盖。

Atlas 200 DK有两种型号的主板,分别为IT21DMDA(旧主板)和IT21VDMB(新主板),您可以通过查看产品序号列的前八位来判断主板型号。

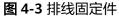
# ----结束

# 4.2 安装摄像头(主板为 IT21DMDA)

# 操作步骤

步骤1 将树莓派摄像头自带的白色排线换成黄色的摄像头排线。

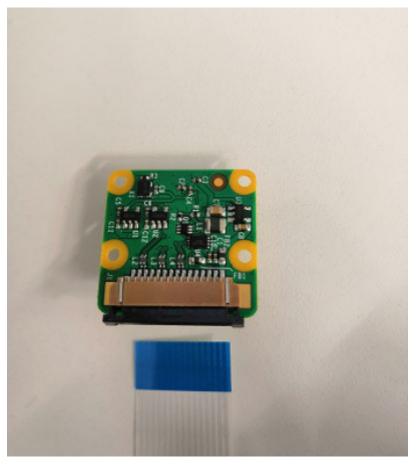
1. 在摄像头一端,拔出摄像头上的黑色排线固定件。如图4-3所示。





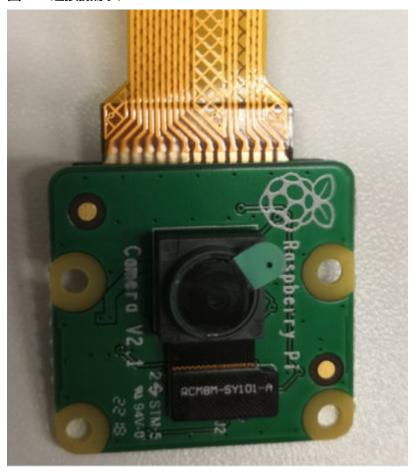
2. 取出白色摄像头排线。如图4-4所示。

图 4-4 白色摄像头排线



3. 将黄色摄像头排线宽的一端金属线方向朝上,水平插入摄像头的排线插槽中,直至不能移动。如<mark>图4-5</mark>所示。

图 4-5 连接摄像头



4. 扣上黑色排线固定件。

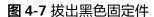
步骤2 将摄像头上的固定胶片套在黄色摄像头排线上。如图4-6所示。

图 4-6 安装固定胶片



步骤3 连接摄像头排线到Atlas 200 DK开发者板。

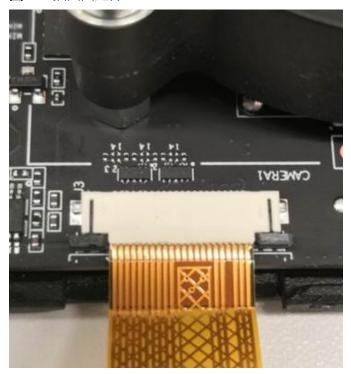
1. 拔出Atlas 200 DK开发者板的摄像头连接器固定件。如图4-7所示。





2. 将黄色摄像头排线窄的一端金属线朝上,水平插入Atlas 200 DK开发者板的摄像 头连接器CAMERAO或者CAMERA1,直至不能移动,并插回固定件。如<mark>图4-8</mark>所 示。

图 4-8 插回固定件



步骤4 参照拆卸前原位置安装Atlas 200 DK开发者板的上盖。

# 步骤5 安装摄像头支架。

- 1. 使用摄像头支架上的夹子夹住固定胶片。
- 2. 安装好摄像头支架,支起摄像头。

# 山 说明

- 使用前,请撕下摄像头上的保护膜。
- 摄像头支架的底座上带有双面胶,可用于固定支架在桌面上,确保摄像头安装稳固。

# ----结束

# 4.3 安装摄像头(主板为 IT21VDMB)

# 操作步骤

步骤1 将树莓派摄像头自带的白色排线换成黑色的摄像头排线。

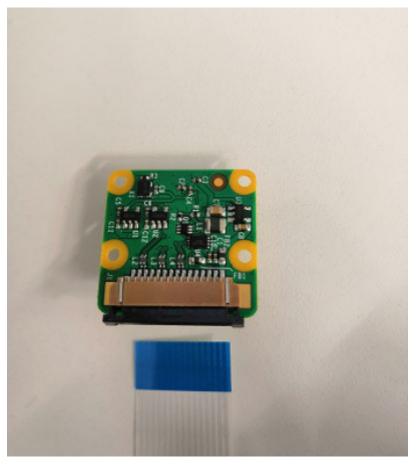
1. 在摄像头一端,拔出摄像头上的黑色排线固定件。如图4-9所示。

图 4-9 拔出排线固定件



2. 取出白色摄像头排线。如图4-10所示。

图 4-10 白色摄像头排线



3. 将黑色摄像头排线丝印显示"TO CAMERA"的一端金属线方向朝上,水平插入摄像头的排线插槽中,直至不能移动。如<mark>图4-11</mark>所示。

图 4-11 连接摄像头



4. 扣上黑色排线固定件。

步骤2 将摄像头上的固定胶片套在黑色摄像头排线上。

步骤3 连接摄像头排线到Atlas 200 DK开发者板。

# 须知

- 连接器固定件的打开范围最大为90度,向上打开连接器固定件时请勿超过90度。
- 请勿反方向打开连接器固定件,否则会导致连接器断裂。

1. 将Atlas 200 DK开发者板的摄像头连接器固定件由里向上打开90°。如图4-12所示。

图 4-12 掰开黑色固定件



2. 将黑色摄像头排线丝印显示"TO MAIN BD"的一端黑色线朝上,水平插入Atlas 200 DK开发者板的摄像头连接器CAMERAO或者CAMERA1,直至不能移动,并扣上固定件。如图4-13所示。

图 4-13 扣好固定件



步骤4 参照拆卸前原位置安装Atlas 200 DK开发者板的上盖。

# 步骤5 安装摄像头支架。

- 1. 使用摄像头支架上的夹子夹住固定胶片。
- 2. 安装好摄像头支架,支起摄像头。

# 🗀 说明

- 使用前,请撕下摄像头上的保护膜。
- 摄像头支架的底座上带有双面胶,可用于固定支架在桌面上,确保摄像头安装稳固。

# ----结束

# 5 制作 SD 卡

- 5.1 简介
- 5.2 读卡器场景
- 5.3 无读卡器场景

# 5.1 简介

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

Atlas 200 DK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Atlas 200 DK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用户名和属组都为HwHiAiUser的默认用户,该用户用来运行应用程序,同时,该用户还有用户态AI驱动的管理权限。首次登录时请参见7 修改用户密码进行密码修改,需要对该用户的登录密码妥善保管。
- 不支持将用户态AI驱动的管理权限切换到其他用户。请不要删除HwHiAiUser 用户,以免导致系统异常。如果系统使用过程中有删除、重新增加 HwHiAiUser用户的操作,需要重新启动设备。

# 5.2 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5-1所示。

表 5-1 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 <b>Ubuntu官网</b> 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"Atlas200 DK开发者套件"。         3. "固件与驱动版本":选择"1.0.13.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Atlas 200 DK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 5-1 SD 制卡回显信息示例

```
Constitution in Profession 1:00-64-PCI-MT:/home/ascend/mksdf python3 make_sd_card.py local /dev/sds
Megain to make so Card
Megain to Megain
Megain to make so Card
Megain to Megain
Megain to make so Card
Megain to Megain
Megain
Megain to Megain
```

# □ 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Atlas 200 DK开发者板卡槽。

**步骤8** 上电Atlas 200 DK开发者板。

# 须知

- Atlas 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Atlas 200 DK开发者板时不能断电,以免对Atlas 200 DK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。

Atlas 200 DK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 Al加速模块 1.0.11及以上 npu-smi 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Atlas 200 DK下电。

步骤2 从Atlas 200 DK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Atlas 200 DK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: Atlas 200 DK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Atlas 200 DK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Atlas 200 DK无法正常启动。

----结束

# 5.3 无读卡器场景

介绍无读卡器场景下,通过跳线帽/杜邦线短接开发者板引脚的方式制作SD卡的操作。

# 硬件准备

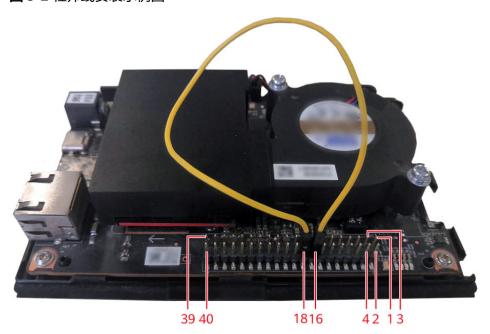
1. 拆卸上盖,请参见**4.1 拆卸上盖**。

2. 请准备跳线帽或者杜邦线,然后参考<mark>图5-2</mark>短接开发者板上的16号与18号两个针脚,使得Atlas 200 DK实现读卡器的功能。

# 须知

- 请在开发者板断电条件下执行此操作。开发者板的下电操作要求请参考12.5 下 电Atlas 200 DK开发者板。
- 请仔细核对针脚位置,若针脚插错将会对Atlas 200 DK开发者板造成严重损坏。
- 面板上用白色字体标出了1、2、40三个针脚的位置,可仔细查看。

# 图 5-2 杜邦线安装示例图



- 3. 将Atlas 200 DK开发者板与用户PC机的USB接口连接。
- 4. 将Atlas 200 DK开发者板上电,详细上电操作请参考**12.4 上电Atlas 200 DK开发 者板**。

# 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5-2所示。

表 5-2 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 <b>Ubuntu官网</b> 下载此软件包。

包信息	包名	说明	下载地址
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"Atlas 200 DK开发者 套件"。3. "固件与驱动版 本":选择 "1.0.13.alpha"。

# 须知

软件下载后请保持原命名。

# 操作步骤

步骤1 请参见硬件准备短接Atlas 200 DK的16号与18号针脚,与用户PC机的USB接口连接, 并上电Atlas 200 DK。

**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qithub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qithub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Atlas 200 DK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5-3所示表示制卡成功。

# 图 5-3 SD 制卡回显信息示例

```
root@ascend.#P.Probesk.600-64-PCI-MT:/home/ascend/mksdf python3 make_sd_card.py local /dev/sda
Spines asks sure you have installed dependency packages:

__astropy of the installed dependency packages:

__astropy of the installed dependency packages:

__astropy of the installed dependency packages:

__astropy installed value-user-static buffet-support gcc-aarch64-linux-gnu g**-aarch64-linux-gnu
Pleass input Y: continue, other to install them?

Pleass input Y: continue, other to install them?

Pleass input Y: continue, other to install them?

Beautiful them as the installed dependency of the installed dependency
```

# □ 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Atlas 200 DK开发者板下电,详细下电操作请参考12.5 下电Atlas 200 DK开发者板。

步骤8 拔掉跳线帽或者杜邦线。

**步骤9** 上电Atlas 200 DK开发者板。

### 须知

- Atlas 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Atlas 200 DK开发者板时不能断电,以免对Atlas 200 DK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。

Atlas 200 DK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 Al加速模块 1.0.11及以上 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Atlas 200 DK下电。

步骤2 使用跳线帽/杜邦线短接Atlas 200 DK的16与18号针脚,如硬件准备所示,将Atlas 200 DK作为USB设备。

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

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Atlas 200 DK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: Atlas 200 DK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Atlas 200 DK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Atlas 200 DK无法正常启动。

# ----结束

# 6 配置网络连接

# 前提条件

进行网络配置前,需要确保Atlas 200 DK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

Atlas 200 DK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Atlas 200 DK接入互联网。

Atlas 200 DK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Atlas 200 DK连接用户PC机,使得可以在PC机上SSH登录到Atlas 200 DK; 使用网线将Atlas 200 DK接入路由器的网口,通过配置NIC网卡将Atlas 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Atlas 200 DK与PC机连接,通过网络配置使得Atlas 200 DK可以共享PC机网络,从而接入互联网。

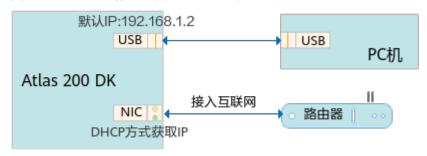
# □ 说明

若您的Atlas 200 DK无条件接入互联网,您可以参见**12.11 连接Atlas 200 DK与用户PC机**,仅实现Atlas 200 DK与PC机之间的通信,需要注意此种场景下,Atlas 200 DK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Atlas 200 DK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Atlas 200 DK; 然后将Atlas 200 DK通过网线接入路由器,并配置Atlas 200 DK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 6-1 通过接入路由器联网的配置示例



步骤1 使用USB连接线将Atlas 200 DK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Atlas 200 DK通信。

Atlas 200 DK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Atlas 200 DK通信。

# 须知

- 若制作SD卡时,您已修改Atlas 200 DK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Atlas 200 DK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qithub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Atlas 200 DK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。
- o 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。
- 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enpOs20fOu4的IP地址是否生效。

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

步骤3 配置Atlas 200 DK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Atlas 200 DK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<mark>密码修改</mark>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Atlas 200 DK开发者板上连接网络了。

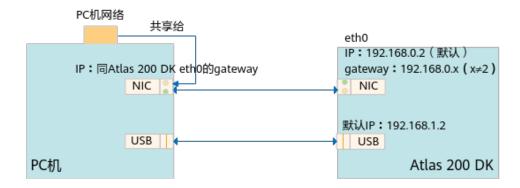
**步骤5** 执行ifconfig命令获取eth0网卡的IP地址,您可以使用此IP地址与PC机通信,当然,您也可以继续使用USB网卡的IP地址与PC机通信。

# ----结束

# 通过网络共享方式接入互联网

此种场景下,Atlas 200 DK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Atlas 200 DK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Atlas 200 DK共享PC机网络,Atlas 200 DK 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Atlas 200 DK的NIC网卡,则Atlas 200 DK即可接入互联网。配置示例如下图所示:

# 图 6-2 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Atlas 200 DK与PC机连接。

PC机中的"网络连接"显示如下图所示:

### 图 6-3 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Atlas 200 DK。
- "Ethernet2(以太网2)":通过网线连接Atlas 200 DK的NIC网卡,后续将使用 此网卡作为网关,用于Atlas 200 DK接入互联网。

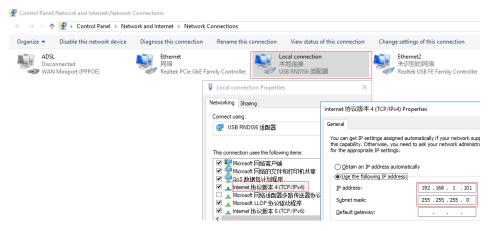
说明:以上名字仅为示例,您可以通过拔插网口的方式查看连接Atlas 200 DK的是哪一个网卡。

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Atlas 200 DK。

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

Atlas 200 DK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 6-4 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Atlas 200 DK:

ssh HwHiAiUser@192.168.1.2

步骤3 配置Atlas 200 DK的eth0网卡的网关。

Atlas 200 DK中eth0网卡的网关与eth0 IP地址在同一网段即可,例如Atlas 200 DK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:** Atlas 200 DK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Atlas 200 DK。

#### 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 6-5 Atlas 200 DK 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。

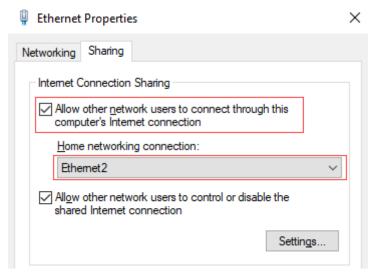
步骤4 在PC侧进行网络配置,将网络共享给Atlas 200 DK。

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Atlas 200 DK的 网卡。

如<mark>图6-3</mark>所示,能够接入互联网的网卡为"Ethernet",连接Atlas 200 DK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。

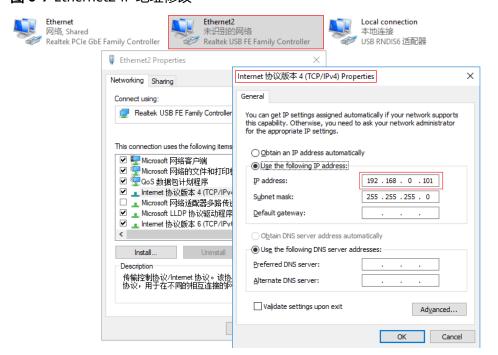
#### 图 6-6 共享网络配置



- 2. 单击"OK",保存配置。
- 3. 修改 "Ethernet2"的IP地址为Atlas 200 DK eth0网关的IP地址(例如,<mark>图6-5</mark>中配置的"192.168.0.101")。

修改方法如图6-7所示,至此,您的Atlas 200 DK已可以接入互联网。

#### 图 6-7 Ethernet2 IP 地址修改



#### □ 说明

通过网络共享方式使得Atlas 200 DK接入网络的原理如下:

- Atlas 200 DK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。
- Atlas 200 DK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

# **了**修改用户密码

#### 密码修改策略

环境搭建完成后,请修改Atlas 200 DK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**"。Atlas 200 DK开发者板与PC机连接成功后需要修改Atlas 200 DK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Atlas 200 DK开发者板。

#### □ 说明

若以SSH方式登录Atlas 200 DK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。

执行passwd命令修改HwHiAiUser密码。
 如图7-1所示。

#### 图 7-1 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:
Enter 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Atlas 200 DK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7-2所示。

#### 图 7-2 修改 root 用户密码

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

# **8** 安装 CANN 软件

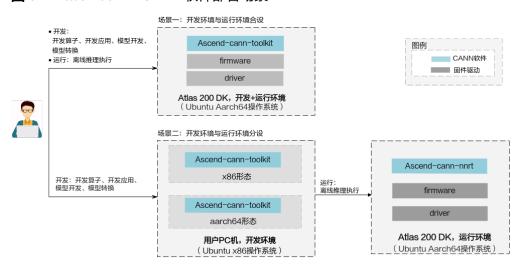
- 8.1 场景描述
- 8.2 开发环境与运行环境合设(推荐)
- 8.3 开发环境与运行环境分设

# 8.1 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Atlas 200 DK, 常见的CANN软件部署场景有下图所示的两种:





● 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,Atlas 200 DK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,Atlas 200 DK需要接入互联网,详情操作方法可参见6 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6配置网络连接将Atlas 200 DK与开发环境所在PC机进行连接。

- 运行环境: Atlas 200 DK。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Atlas 200 DK侧,进行离线推理的执行。

## 8.2 开发环境与运行环境合设(推荐)

此种场景下,Atlas 200 DK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Atlas 200 DK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-canntoolkit\_{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Atlas 200 DK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Atlas 200 DK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

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

# 8.3 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,Atlas 200 DK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 ./Ascend-cann-toolkit {version} linux-aarch64.run --install

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Atlas 200 DK上安装离线推理引擎包。

请以制卡时创建的**HwHiAiUser**用户作为Atlas 200 DK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Atlas 200 DK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 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

# 9 准备编译运行依赖文件

#### 简介

若开发者需要使用Atlas 200 DK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Atlas 200 DK上进行应用程序的开发与编译。

制作SD卡时,已在Atlas 200 DK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Atlas 200 DK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/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拷贝到\${INSTALL\_DIR}/driver目录下。

- cp /usr/lib64/libmedia\_mini.so \${INSTALL\_DIR}/driver/
- cp /usr/lib64/libslog.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/libslog.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./usr/lib64/libmmpa.so \${INSTALL\_DIR}/driver/

sudo scp -r HwHiAiUser@X.X.X./usr/local/Ascend/include/peripheral\_api.h \${INSTALL\_DIR}/driver/

#### ----结束

# 10 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。

#### 山 说明

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

# 1 1 使用 Memory Cgroup 机制实现业务内存资源控制

#### 简介

Atlas 200 DK提供了基于cgroup(Control Groups)机制的内存限制功能,当业务进程的cgroup达到特定的内存限额后(内存限额设置方法请参见**内存限额设置**),内核会尝试回收内存。

如果内存回收失败,业务进程会被挂起,后续有内存后再被唤醒,从而避免内存溢出 (用户也可以参见**内存不足时应用退出设置**,实现内存回收失败时应用程序直接退 出)。

#### 添加业务进程到 Cgroup

用户需要自行在Atlas 200 DK开发板上将业务应用进程添加到业务进程的cgroup中,添加方法如下:

以root用户执行如下命令:

echo ServicePID > /sys/fs/cgroup/memory/usermemory/cgroup.procs

ServicePID: 业务应用程序的进程ID。

#### 内存限额设置

● 内存限额的计算方法如下:

limit = FreeMem - FreeCma - 预留大小

- FreeMem:空闲内存,可通过"cat /proc/meminfo"命令获取 "MemFree"的值,并将其单位转换为字节,即:MemFree\*1024。
- FreeCma:空闲的连续内存,可通过"cat /proc/meminfo"命令获取 "CmaFree"的值,并将其单位转换为字节,即:CmaFree \* 1024。
- 预留大小: 预留给不在user memory cgroup中的服务进程使用,若设置太小可能会导致内存限额(limit)起不到限制作用,推荐设置为:
   200\*1024\*1024 Byte~300\*1024\*1024 Byte。
- 内存限额的设置方法如下:

echo limit > /sys/fs/cgroup/memory/usermemory/memory.limit\_in\_bytes

内存限额设置代码示例:

local free\_mem=`cat /proc/meminfo | grep "MemFree" | awk '{print \$2}'` local free\_cma=`cat /proc/meminfo | grep "CmaFree" | awk '{print \$2}'`

echo  $((free_mem * 1024 - free_cma * 1024 - 300 * 1024 * 1024)) > /sys/fs/cgroup/memory/usermemory/memory.limit_in_bytes$ 

#### 内存不足时应用退出设置

当业务进程的cgroup达到特定的内存限额后,内核会尝试回收内存,如果内存回收失败,业务进程会被挂起。若用户不想一直等待,可执行如下操作,实现内存不足时应用程序自动退出。

#### 步骤1 开启操作系统OOM Killer机制。

登录Atlas 200 DK,以root用户执行如下命令启用enable\_oom\_killer,1表示启用,0表示禁用。

echo 1 > /proc/sys/vm/enable\_oom\_killer

#### 步骤2 开启memory cgroup的OOM控制机制。

登录Atlas 200 DK,以root用户执行如下命令启用memory cgroup的OOM机制,0表示"oom kill enable",1表示"oom kill disable"。

echo 0 > /sys/fs/cgroup/memory/usermemory/memory.oom\_control

#### 须知

memory cgroup的kill enable功能生效的前提是系统enable\_oom\_killer开启。

#### ----结束

# **12** 常用操作

- 12.1 升级Atlas 200 DK固件与驱动
- 12.2 升级CANN软件
- 12.3 查看Atlas 200 DK固件与驱动版本
- 12.4 上电Atlas 200 DK开发者板
- 12.5 下电Atlas 200 DK开发者板
- 12.6 通过串口连接Atlas 200 DK
- 12.7 查看开发者板硬件底板的版本
- 12.8 查看Atlas 200 AI加速模块的版本
- 12.9 如何查看摄像头所属Channel
- 12.10 如何安装windows的USB网卡驱动
- 12.11 连接Atlas 200 DK与用户PC机
- 12.12 修改Atlas 200 DK IP地址
- 12.13 设置用户有效期
- 12.14 相关工具

# 12.1 升级 Atlas 200 DK 固件与驱动

若您需要升级固件或驱动软件,可参见本章节进行升级操作。

#### 须知

若Atlas 200 DK上的固件驱动软件为1.0.7.alpha及之前版本,则不支持升级,您需要参见5 制作SD卡重新制作SD卡。

(1.0.7.alpha版本的固件驱动对应的后台版本号为1.73.xx.xx,后台版本号的查询方法可参见12.3 查看Atlas 200 DK固件与驱动版本)。

#### 获取升级软件包

驱动与固件的软件包信息如下表所示。

表 12-1 软件包信息

组件	包名	说明	升级操作
Driver	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。下载地址: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"Atlas 200 DK开发者套件"。 3. "固件与驱动版本":选择"1.0.13.alpha"。	升级Driver

#### 升级 Driver

步骤1 将Driver软件包A200dk-npu-driver-{software version}-ubuntu18.04-aarch64-minirc.tar.gz拷贝到Atlas 200 DK的"/opt/mini"目录下。

下面以将软件包从用户PC拷贝到Atlas 200 DK为例进行操作介绍:

1. 进入用户PC中存放Driver软件包的目录,并在当前目录下以HwHiAiUser用户ssh登录到Atlas 200 DK,然后切换到root用户。

ssh HwHiAiUser@ 192.168.1.2

su - root

□ 说明

若SSH方式登录Atlas 200 DK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. 进入Atlas 200 DK的/opt/mini目录,并拷贝Driver升级包。

cd /opt/mini

scp *username*@192.168.1.223:/home/ascend/software/A200dk-npu-driver-{software version}-ubuntu18.04-aarch64-minirc.tar.gz .

#### 山 说明

- username为PC机中上传软件包的用户名。
- 192.168.1.223为PC机中与Atlas 200 DK在同一网段的IP地址。
- "/home/ascend/software"为PC机中存放Driver软件包的路径。

步骤2 在/opt/mini目录下执行如下命令,解压缩Driver包,获取Driver包的升级脚本 "minirc\_install\_phase1.sh"。

tar --no-same-owner -zxf A200dk-npu-driver-{software version}-ubuntu18.04-aarch64-minirc.tar.gz --strip-components 2 driver/scripts/minirc\_install\_phase1.sh

执行完成后,解压缩获得的"minirc\_install\_phase1.sh"脚本会替换目录下的原有脚本。

步骤3 执行以下命令进行Driver包升级前准备。

./minirc\_install\_phase1.sh

回显如图12-1所示。

#### 图 12-1 执行升级脚本

```
inflating: /var/mini/install_cache/mini_developerkit/lpm3.img
inflating: /var/mini/install_cache/mini_developerkit/libmatrixdaemon.so
inflating: /var/mini/install_cache/mini_developerkit/libcce_aicpudev_alg.a
inflating: /var/mini/install_cache/mini_developerkit/libdrvupgrade.so
inflating: /var/mini/install_cache/mini_developerkit/libmmpa.so
inflating: /var/mini/install_cache/mini_developerkit/tibd_daemon_client_cert.pem
inflating: /var/mini/install_cache/mini_developerkit/libdrvdevmm.so
inflating: /var/mini/install_cache/mini_developerkit/libdrvdevmm.so
inflating: /var/mini/install_cache/mini_developerkit/libdevmm.so
inflating: /var/mini/install_cache/buildinfo.txt
inflating: /var/mini/install_cache/check_sha.sh
2019-01-16,14:01:1547648536 extrack package succ
writing files to sdcard, please waiting several minutes ...
files check pass!
2019-01-16,14:01:1547648561 the first phase of installation finish, please reboot
root@davinci-mini:/opt/mini#
```

#### □说明

若执行Driver升级脚本时,出现"CheckPartitionSpace partition space check failed"的错误,请参见13.4 Driver升级失败,报"CheckPartitionSpace partition space check failed"进行处理。

步骤4 重启Atlas 200 DK,从而完成Driver包的升级。

reboot

#### 须知

升级过程中请勿将Atlas 200 DK开发者板断电,升级时间15分钟左右。

----结束

### 12.2 升级 CANN 软件

#### 开发环境与运行环境合设场景

此场景下,Atlas 200 DK上安装的是开发套件包"Ascend-cann-toolkit\_{version}\_linux-aarch64.run",可直接使用如下命令进行开发套件包的升级:

- 使用默认路径安装的场景
  - ./Ascend-cann-toolkit\_{version}\_linux-aarch64.run --upgrade
- 使用自定义路径安装的场景
  - ./Ascend-cann-toolkit\_{version}\_linux-aarch64.run --upgrade --install-path=<path>

#### 开发环境与运行环境分设场景

开发环境升级

开发环境上安装的是开发套件包 "Ascend-cann-toolkit\_*{version}\_*linux-x86\_64.run"与 "Ascend-cann-toolkit\_*{version}\_*linux-aarch64.run",可直接使用如下命令进行开发套件包的升级:

- 使用默认路径安装的场景
  - ./Ascend-cann-toolkit\_{version}\_linux-x86\_64.run --upgrade
    ./Ascend-cann-toolkit\_{version}\_linux-aarch64.run --upgrade
- 使用自定义路径安装的场景
  - ./Ascend-cann-toolkit\_{version}\_linux-x86\_64.run --upgrade --install-path=<path>
  - ./Ascend-cann-toolkit\_{version}\_linux-aarch64.run --upgrade --install-path=<path>
- 运行环境升级

运行环境上安装的是离线推理引擎包 "Ascend-cann-nnrt\_*{software version}\_*linux-aarch64.run",可直接使用如下命令进行离线推理引擎包的升级:

- 使用默认路径安装的场景
  - ./Ascend-cann-nnrt\_{software version}\_linux-aarch64.run --upgrade
- 使用自定义路径安装的场景
  - ./Ascend-cann-nnrt\_{software version}\_linux-aarch64.run --upgrade -install-path=<path>

# 12.3 查看 Atlas 200 DK 固件与驱动版本

步骤1 在PC机中以HwHiAiUser用户SSH登录到Atlas 200 DK开发者板。

ssh HwHiAiUser@192.168.1.2

□ 说明

192.168.1.2请替换为开发者板实际IP地址。

步骤2 切换到root用户,执行如下命令查看Driver的版本号。

su - root

cat /var/davinci/driver/version.info

步骤3 执行如下命令查询Firmware的版本号及有效组件版本号。

1. 切换到root用户。

su - root

2. 执行如下命令查询Firmware的版本号。

cd /var/davinci/driver/

./upgrade-tool --device\_index -1 --system\_version

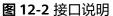
----结束

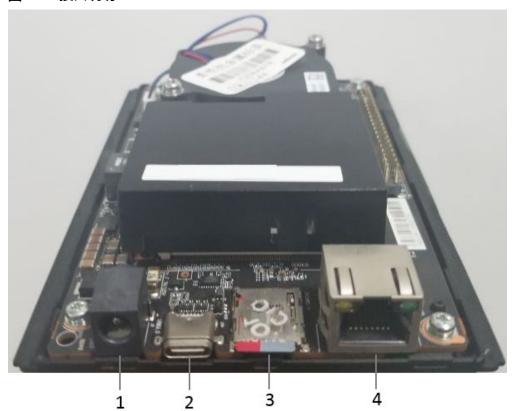
# 12.4 上电 Atlas 200 DK 开发者板

#### 须知

首次启动或者升级Atlas 200 DK开发者板时不能断电,以免对Atlas 200 DK开发者板造成损坏,再次上电需与上次下电时间保持2s以上的安全时间间隔。

步骤1 连接电源线,将电源模块接通外部电源。Atlas 200 DK开发者板的电源接口如<mark>图12-2</mark> 所示。接通电源后,Atlas 200 DK开发者板自动启动。





1	电源接口
2	USB
3	SD卡
4	岡口

步骤2 检查指示灯的状态,确保Atlas 200 DK开发者板上电正常。

指示灯需要在拆卸上盖后才能看到,指示灯的位置及对应的标号如下所示。

----结束

## 12.5 下电 Atlas 200 DK 开发者板

#### 注意事项

请根据步骤2的说明判断是否可以对Atlas 200 DK开发者板进行下电操作。

#### 操作步骤

步骤1 从电源接口处拔掉电源线切断外部电源,将Atlas 200 DK开发者板完全下电。

#### 须知

Atlas 200 DK当前不支持操作系统级别的shutdown命令关机。

----结束

# 12.6 通过串口连接 Atlas 200 DK

#### 通过串口连接 Atlas 200 AI 加速模块

Atlas 200 DK开发者板可以通过如下串口查看Atlas 200 AI加速模块的启动信息。

#### □ 说明

此串口仅用于查看启动信息,启动成功后,Atlas 200 Al加速模块的串口会被禁用,无法登录系统。

通过串口线连接Atlas 200 AI加速模块的示例图如图12-3所示。

图 12-3 Atlas 200 AI 加速模块串口连线示例图



Pin#	NAME		NAME	Pin#
1	3.3V	<b>0</b>	5.0V	2
3	SDA2	<b>o</b> •	5.0V	4
5	SCL2	<b>o</b> •	GND	6
7	GPIO0	• •	(TXD0)	8
9	GND	• •	( RXD0 )	10
11	GPIO1	• •	NC	12
13	NC	. 0	GND	14
15	PWM1	• •	(TXD1)	16
17	3.3V	• •	(RXD1)	18
19	SPI1_MOSI	0 0	GND	20
21	SPI1_MISO	<b>O O</b>	NC	22
23	SPI1_CLK	0 0	SPI1_CS0	24
25	GND	• •	NC	26
27	CANH0	0 0	CANL0	28
29	GPIO3	0 0	GND	30
31	GPIO4	0 0	NC	32
33	GPIO5	• •	GND	34
35	GPIO6	0 0	1.8V	36
37	GPIO7	0 0	Reserved	38
39	GND	0 0	Reserved	40

Atlas 200 AI加速模块串口: 串口接线按图12-3颜色标示对应接入。

串口线要求: USB转串口线(3.3V)。

#### 通过串口连接 3559 模块

Atlas 200 DK开发者板提供连接3559模块的串口,串口连线示例图如图12-4所示。

#### 山 说明

此串口仅用于查看启动信息,启动成功后,3559模块的串口会被禁用,无法登录系统。

图 12-4 3559 串口连线示例图



Pin#	NAME		NAME	Pin#
1	3.3V		5.0V	2
3	SDA2	<b>o</b> •	5.0V	4
5	SCL2	<b>O</b> O	GND	6
7	GPIO0	• •	( TXD0 )	8
9	GND	0 0	( RXD0 )	10
11	GPIO1	• •	NC	12
13	NC	. 0	GND	14
15	PWM1	• •	(TXD1)	16
17	3.3V	• •	(RXD1)	18
19	SPI1_MOSI	0 0	GND	20
21	SPI1_MISO	•	NC	22
23	SPI1_CLK	0 0	SPI1_CS0	24
25	GND	•	NC	26
27	CANH0	0 0	CANL0	28
29	GPIO3	0 0	GND	30
31	GPIO4	0 0	NC	32
33	GPIO5	• •	GND	34
35	GPIO6	0 0	1.8V	36
37	GPIO7	• •	Reserved	38
39	GND	0 0	Reserved	40

3559串口(Hi3559): 串口接线按<mark>图12-4</mark>颜色标示对应接入。

串口线要求: USB转串口线(3.3V)。

# 12.7 查看开发者板硬件底板的版本

开发者板的硬件底板版本号可以通过查看开发者底板的PCB的版本号获取。

步骤1 新建查询开发者板版本号的代码文件。

以普通用户在用户PC机任一目录新建i2c\_tool\_atlas200dk.c文件。

#### touch i2c\_tool\_atlas200dk.c

将以下代码拷贝到i2c\_tool\_atlas200dk.c文件中。

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/ioctl.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <sys/select.h>
#include <sys/time.h>

```
#include <errno.h>
#include <string.h>
#define I2C0_DEV_NAME "/dev/i2c-0"
#define I2C1_DEV_NAME "/dev/i2c-1"
#define I2C2_DEV_NAME "/dev/i2c-2"
#define I2C3_DEV_NAME "/dev/i2c-3"
#define I2C_RETRIES 0x0701
#define I2C_TIMEOUT 0x0702
#define I2C_SLAVE 0x21
#define I2C_RDWR
                       0x0707
#define I2C_BUS_MODE 0x0780
#define I2C_M_RD 0x01
#define PCB_ID_VER_A 0x1
#define PCB_ID_VER_B 0x2
#define PCB_ID_VER_C 0x3
#define PCB_ID_VER_D 0x4
#define I2C_SLAVE_PCA9555_BOARDINFO (0x20)
#define BOARD_ID_DEVELOP_C (0xCE)
#define DEVELOP_A_BOM_PCB_MASK (0xF)
#define DEVELOP_C_BOM_PCB_MASK (0x7)
typedef unsigned char uint8;
typedef unsigned short uint16;
struct i2c_msg
  uint16 addr; /* slave address */
  uint16 flags;
  uint16 len;
  uint8 *buf; /*message data pointer*/
};
struct i2c_rdwr_ioctl_data
  struct i2c_msg *msgs; /*i2c_msg[] pointer*/
  int nmsgs; /*i2c_msg Nums*/
static uint8 i2c_init(char *i2cdev_name);
static uint8 i2c_read(uint8 slave, unsigned char reg,unsigned char *buf);
int fd = 0;
static uint8 i2c_read(unsigned char slave, unsigned char reg,unsigned char *buf)
  struct i2c_rdwr_ioctl_data ssm_msg;
  unsigned char regs[2] = {0};
  regs[0] = reg;
  regs[1] = req;
  ssm_msg.nmsgs=2;
  ssm_msg.msgs=(struct i2c_msg*)malloc(ssm_msg.nmsgs*sizeof(struct i2c_msg));
  if(!ssm_msg.msgs)
     printf("Memory alloc error!\n");
     return -1:
  (ssm_msg.msgs[0]).flags=0;
  (ssm_msg.msgs[0]).addr=slave;
  (ssm_msg.msgs[0]).buf= regs;
  (ssm_msg.msgs[0]).len=1;
  (ssm_msg.msgs[1]).flags=I2C_M_RD;
  (ssm_msg.msgs[1]).addr=slave;
   (ssm_msg.msgs[1]).buf=buf;
  (ssm_msg.msgs[1]).len=2;
  ret=ioctl(fd, I2C_RDWR, &ssm_msg);
  if(ret<0)
     printf("read data error,ret=%#x, errorno=%#x, %s!\n",ret, errno, strerror(errno));
```

```
free(ssm_msg.msgs);
     return -1;
  free(ssm_msg.msgs);
  return 0;
static uint8 i2c_init(char *i2cdev_name)
  fd = open(i2cdev_name, O_RDWR);
  if(fd < 0)
     printf("Can't open %s!\n", i2cdev_name);
     return -1;
  }
  if(ioctl(fd, I2C_RETRIES, 1)<0)
     printf("set i2c retry fail!\n");
     return -1;
  if(ioctl(fd, I2C_TIMEOUT, 1)<0)
     printf("set i2c timeout fail!\n");
  }
  return 0;
int main(int argc, char *argv[])
  char *dev_name = I2C0_DEV_NAME;
  uint8 board_id;
  uint8 pcb_id;
  uint8 buff[2] = \{0\};
  uint8 ret;
  if (i2c_init(dev_name))
  {
     printf("i2c init fail!\n");
     close(fd);
     return -1;
  usleep(1000*100);
  ret = i2c_read(I2C_SLAVE_PCA9555_BOARDINFO, 0x0, buff);
  if (ret != 0)
     printf("read %s %#x fail, ret %d\n", dev_name, I2C_SLAVE_PCA9555_BOARDINFO, ret);
  close(fd);
  board_id = buff[0];
  if (board_id == BOARD_ID_DEVELOP_C)
     pcb_id = (buff[1]>>3)&DEVELOP_C_BOM_PCB_MASK;
  else
  {
     pcb_id = (buff[1]>>4)&DEVELOP_A_BOM_PCB_MASK;
  // show PCB ID;
  switch (pcb_id)
  case PCB_ID_VER_A:
```

```
printf("PCB version is: Ver.A !\n");
    break;
case PCB_ID_VER_B:
    printf("PCB version is: Ver.B !\n");
    break;
case PCB_ID_VER_C:
    printf("PCB version is: Ver.C !\n");
    break;
case PCB_ID_VER_D:
    printf("PCB version is: Ver.D !\n");
    break;
default:
    break;
}
return 0;
}
```

步骤2 编译生成获取PCB版本号的可执行文件。

执行如下命令将i2c\_tool\_atlas200dk.c文件编译生成在Atlas 200 DK中可执行的文件。 **aarch64-linux-gnu-gcc** *i2c\_tool\_atlas200dk.c* **-o** *atlas200dk\_version\_tool atlas200dk\_version\_tool*为生成的可执行文件的名称。

步骤3 将步骤2生成的可执行文件上传至Atlas 200 DK开发者板。

例如上传到开发者板的HwHiAiUser用户的家目录下。

scp atlas200dk\_version\_tool HwHiAiUser@192.168.1.2:/home/HwHiAiUser

步骤4 以HwHiAiUser用户ssh登录到开发者板,并执行查询开发者板PCB版本号的操作。

ssh HwHiAiUser@192.168.1.2

切换到root用户,并执行查询脚本。

su - root

./atlas200dk\_version\_tool

如下所示,表示当前开发者板为VB版本。

```
root@davinci-mini:/home/HwHiAiUser# ./atlas200dk_version_tool PCB version is: Ver.B !
```

----结束

# 12.8 查看 Atlas 200 AI 加速模块的版本

Atlas 200 AI加速模块的版本号可以通过boardid进行判断,也可以通过Atlas 200 AI加速模块的PCB版本号进行判断,下面分别介绍两种查询方式。

#### 方式一(通过命令查看)

步骤1 以HwHiAiUser用户SSH登录到Atlas 200 DK开发者板。

步骤2 执行如下命令查看Atlas 200 AI加速模块的版本。

cat /proc/cmdline

console=ttyAMA0,115200 root=/dev/mmcblk1p1 rw rootdelay=1 syslog no\_console\_suspend earlycon=pl011,mmio32,0x10cf80000 initrd=0x880004000,200M cma=256M@0x1FC00000 log\_redirect=0x1fc000@0x6fe04000 default\_hugepagesz=2M reboot\_reason=AP\_S\_COLDBOOT

通过boardid的值判断Atlas 200 AI加速模块的版本:

- boardid=000,表示Atlas 200 AI加速模块的版本为VC及VC以前的版本。
- boardid=004,表示Atlas 200 AI加速模块的版本为VD版本。

#### ----结束

#### 方式二(通过查看 Atlas 200 AI 加速模块的 PCB 版本号)

步骤1 新建查询开发者板版本号的代码文件。

以普通用户在用户PC任一目录新建i2c\_tool\_mini.c文件。

#### touch i2c\_tool\_mini.c

将以下代码拷贝到i2c\_tool\_mini.c文件中。

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/ioctl.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <sys/select.h>
#include <sys/time.h>
#include <errno.h>
#include <string.h>
#define I2C0_DEV_NAME "/dev/i2c-0"
#define I2C1_DEV_NAME "/dev/i2c-1"
#define I2C2_DEV_NAME "/dev/i2c-2"
#define I2C3_DEV_NAME "/dev/i2c-3"
#define I2C_RETRIES 0x0701
#define I2C_TIMEOUT 0x0702
                      0x21
#define I2C_SLAVE
#define I2C_RDWR 0x0707
#define I2C_BUS_MODE 0x0780
#define I2C_M_RD
                      0x01
#define PCB_ID_VER_A 0x10
#define PCB_ID_VER_B 0x20
#define PCB_ID_VER_C 0x30
#define PCB_ID_VER_D 0x40
typedef unsigned char uint8;
typedef unsigned short uint16;
struct i2c_msg
  uint16 addr; /* slave address */
  uint16 flags;
  uint16 len;
  uint8 *buf; /*message data pointer*/
struct i2c_rdwr_ioctl_data
  struct i2c_msg *msgs; /*i2c_msg[] pointer*/
  int nmsgs; /*i2c_msg Nums*/
static uint8 i2c_init(char *i2cdev_name);
static uint8 i2c_write(uint8 slave, unsigned char reg, unsigned char value);
static uint8 i2c_read(uint8 slave, unsigned char reg,unsigned char *buf);
int fd = 0;
static uint8 i2c_write(uint8 slave, unsigned char reg, unsigned char value)
  int ret:
  struct i2c_rdwr_ioctl_data ssm_msg;
```

```
unsigned char buf[2]={0};
  ssm_msg.nmsgs=1;
  ssm_msg.msgs=(struct i2c_msg*)malloc(ssm_msg.nmsgs*sizeof(struct i2c_msg));
  if(!ssm_msg.msgs)
     printf("Memory alloc error!\n");
     return -1;
  }
  buf[0] = reg;
  buf[1] = value;
  (ssm_msg.msgs[0]).flags=0;
  (ssm_msg.msgs[0]).addr=(uint16)slave;
  (ssm_msg.msgs[0]).buf=buf;
  (ssm_msg.msgs[0]).len=2;
  ret=ioctl(fd, I2C_RDWR, &ssm_msg);
  if(ret<0)
  {
     printf("write error, ret=%#x, errorno=%#x, %s!\n",ret, errno, strerror(errno));
     free(ssm_msg.msgs);
     return -1;
  free(ssm_msg.msgs);
  return 0:
static uint8 i2c_read(unsigned char slave, unsigned char reg,unsigned char *buf)
  int ret;
  struct i2c_rdwr_ioctl_data ssm_msg;
  unsigned char regs[2] = {0};
  regs[0] = reg;
  regs[1] = reg;
  ssm_msg.nmsgs=2;
  ssm_msg.msgs=(struct i2c_msg*)malloc(ssm_msg.nmsgs*sizeof(struct i2c_msg));
  if(!ssm_msg.msgs)
     printf("Memory alloc error!\n");
     return -1;
  (ssm_msg.msgs[0]).flags=0;
  (ssm_msg.msgs[0]).addr=slave;
  (ssm_msg.msgs[0]).buf= regs;
  (ssm_msg.msgs[0]).len=1;
  (ssm_msg.msgs[1]).flags=I2C_M_RD;
  (ssm_msg.msgs[1]).addr=slave;
  (ssm_msg.msgs[1]).buf=buf;
  (ssm_msg.msgs[1]).len=1;
  ret=ioctl(fd, I2C_RDWR, &ssm_msg);
  if(ret<0)
  {
     printf("read data error,ret=%#x, errorno=%#x, %s!\n",ret, errno, strerror(errno));
     free(ssm_msg.msgs);
     return -1;
  }
  free(ssm_msg.msgs);
  return 0;
static uint8 i2c_init(char *i2cdev_name)
  fd = open(i2cdev_name, O_RDWR);
```

```
if(fd < 0)
  {
     printf("Can't open %s!\n", i2cdev_name);
     return -1;
  if(ioctl(fd, I2C_RETRIES, 1)<0)
     printf("set i2c retry fail!\n");
     return -1;
  if(ioctl(fd, I2C_TIMEOUT, 1)<0)
     printf("set i2c timeout fail!\n");
     return -1;
  return 0;
int main(int argc, char *argv[])
  char *dev_name = I2C0_DEV_NAME;
  uint8 slave;
  uint8 reg;
  uint8 data;
  int ret;
  if (i2c_init(dev_name))
     printf("i2c init fail!\n");
     close(fd);
     return -1;
  usleep(1000*100);
  // Read PCB ID
  slave = I2C_SLAVE;
  reg = 0x07;
  data = 0x5A;
  ret = i2c_read(slave, reg, &data);
  if (ret != 0)
     printf("read %s %#x %#x to %#x fail!\n", dev_name, slave, data, reg);
  slave = I2C_SLAVE;
  reg = 0x07;
  data = data | 0xF0;
  ret = i2c_write(slave, reg, data);
  if (ret != 0)
     printf("write %s %#x %#x to %#x fail!\n", dev_name, slave, data, reg);
  }
  slave = I2C_SLAVE;
  reg = 0x01;
  data = 0x5A;
  ret = i2c_read(slave, reg, &data);
  if (ret != 0)
  {
     printf("read %s %#x %#x to %#x fail!\n", dev_name, slave, data, reg);
  close(fd);
  // show PCB ID;
```

```
switch (data & 0xF0)
case PCB_ID_VER_A:
  printf("PCB version is: Ver.A !\n");
  break;
case PCB_ID_VER_B:
  printf("PCB version is: Ver.B !\n");
  break;
case PCB ID VER C:
  printf("PCB version is: Ver.C !\n");
  break;
case PCB_ID_VER_D:
  printf("PCB version is: Ver.D !\n");
  break;
default:
  break;
return 0;
```

步骤2 编译生成获取PCB版本号的可执行文件。

执行如下命令将i2c\_tool\_mini.c文件编译生成在开发者板中可执行的文件。

aarch64-linux-gnu-gcc i2c\_tool\_mini.c -o mini\_version\_tool

mini\_version\_tool为生成的可执行文件的名称。

步骤3 将步骤2生成的可执行文件上传至开发者板。

例如上传到开发者板的HwHiAiUser用户的家目录下。

scp mini\_version\_tool HwHiAiUser@192.168.1.2:/home/HwHiAiUser

步骤4 以HwHiAiUser用户ssh登录到开发者板,并执行查询开发者板PCB版本号的操作。

ssh HwHiAiUser@192.168.1.2

切换到root用户,并执行查询脚本。

su - root

./mini\_version\_tool

如下所示,表示当前AI加速模块的版本为VC版本。

```
root@davinci-mini:/home/HwHiAiUser# ./mini_version_tool PCB version is: Ver.C !
```

----结束

# 12.9 如何查看摄像头所属 Channel

Atlas 200 DK有两个MIPI-CSI接口,可外接两个摄像头。

可通过查看开发者板面板判断当前所使用的摄像头通道,如图12-5所示。



图 12-5 查看摄像头所属通道

- "CAMERAO"对应的摄像头通道为"Channel-1"
- "CAMERA1"对应的摄像头通道为"Channel-2"。

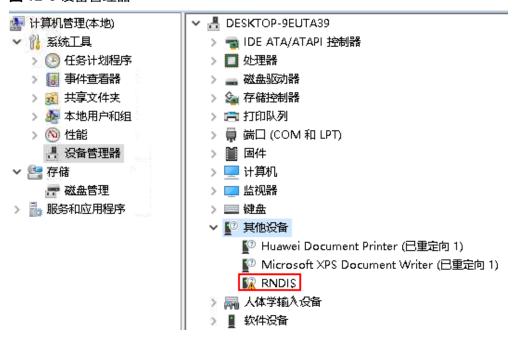
# 12.10 如何安装 windows 的 USB 网卡驱动

Ubuntu操作系统为通过Windows操作系统上的虚拟机安装的场景下,需要在Windows操作系统上安装USB网卡驱动,即RNDIS(Remote Network Driver Interface Specification)驱动,否则Atlas 200 DK通过USB连接Ubuntu所在的Windows主机时,在Ubuntu操作系统中无法识别Atlas 200 DK的USB虚拟网卡。

假设您已经将Atlas 200 DK通过USB连接线连接到了Ubuntu所在的Windows主机,下面以在Windows 10操作系统上安装RNDIS驱动为例进行介绍。

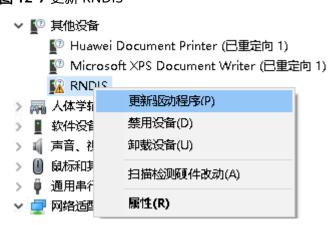
**步骤1** 在"计算机管理"操作界面中选择"设备管理器 >其他设备",如下图所示,RNDIS为未识别状态。

#### 图 12-6 设备管理器



步骤2 右键单击 "RNDIS",选择"更新驱动程序(P)"。

#### 图 12-7 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取(L)",单击"下一步(N)"。

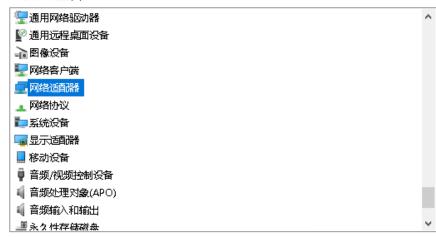
步骤4 在"常见硬件类型"列表中选择"网络适配器",单击"下一步(N)"。

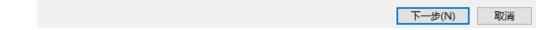
#### 图 12-8 选择网络适配器

← ■ 更新驱动程序 - RNDIS

#### 从以下列表选择设备的类型。

#### 常见硬件类型(H):





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 12-9 选择驱动程序

← ■ 更新驱动程序 - RNDIS

#### 选择要为此硬件安装的设备驱动程序



请选定硬件没备的厂商和型号,然后单击"下一步"。如果手头有包含要安装的驱动程序的磁盘,请单击"从磁盘安装"。



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 12-10 RNDIS 驱动正常显示



----结束

# 12.11 连接 Atlas 200 DK 与用户 PC 机

6 配置网络连接章节已介绍了如何实现Atlas 200 DK与PC机进行通信,并将Atlas 200 DK接入互联网的配置方式,若您的Atlas 200 DK无条件接入互联网,您可以参见本章节进行连接,仅实现Atlas 200 DK与用户PC机的通信。需要注意,此种场景下,Atlas 200 DK仅适合作为运行环境。

Atlas 200 DK提供了USB网卡与NIC网卡,您可以通过USB接口直连PC机或者NIC接口直连PC机的方式,实现Atlas 200 DK与PC机的通信。下面分别介绍两种方法的操作步骤。

#### 通过 USB 接口直连 PC 机

Atlas 200 DK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虚拟 网卡的IP地址为192.168.1.x(x取值范围为0~1,3~254),使Atlas 200 DK与PC机可以通信。

#### 须知

- 若制作SD卡时,您已修改Atlas 200 DK开发者板的USB网卡地址与PC机的虚拟USB 网卡的IP地址在同一网段,则Atlas 200 DK与PC机可直接通过USB网卡的IP地址进行通信,本操作无需再做。
- 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Atlas 200 DK的USB网卡。

下面提供通过脚本配置PC机的USB虚拟网卡的IP地址与手工配置PC机的USB虚拟网卡IP地址的两种方法。

假设您已经通过USB连接线将Atlas 200 DK与PC机的USB接口进行连接,操作步骤如下:

- 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 **于工配置**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Atlas 200 DK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

- o usb nic name: USB网卡名称。
- o 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enpOs20f0u4的IP地址是否生效。

#### 通过 NIC 接口直连 PC 机

Atlas 200 DK NIC网卡的默认地址为192.168.0.2,子网掩码24位,所以需要修改用户PC机的IP地址为192.168.0.x(x取值范围为0~1,3~254),使Atlas 200 DK与用户PC机可以通信。

#### 须知

- 若制作SD卡时,您已修改Atlas 200 DK开发者板NIC网卡的IP地址与用户PC机的NIC网卡的IP地址在同一网段,则Atlas 200 DK与PC机可直接通过NIC网卡的IP地址进行通信,本操作无需再做。
- Atlas 200 DK网口接入网线后,黄色的ACT灯闪烁,代表有数据传输。当是干兆以太网时,绿色的LINK灯会亮,当是百兆/10兆的以太网时,LINK灯正常不会亮。

#### 配置方法如下:

- 1. 以普通用户登录用户PC机,执行如下命令切换到root用户。su-root
- 2. 配置一个虚拟网卡IP,用于与Atlas 200 DK进行通信。

例如,配置ethO:1的静态IP地址。

ifconfig eth0:1 192.168.0.223 netmask 255.255.0.0 up

## 12.12 修改 Atlas 200 DK IP 地址

环境搭建完成后,若您需要修改Atlas 200 DK的IP地址,请参考以下步骤操作。

- 1. 以HwHiAiUser用户ssh登录Atlas 200 DK开发者板。
- 2. 在Atlas 200 DK中,切换到root用户修改Atlas 200 DK中的网络配置文件。
  - a. 修改/etc/netplan/01-netcfg.yaml配置文件中的address及gateway4,然后保存退出。
    - 若Atlas 200 DK通过USB端口连接用户PC机,可修改**usb0**网卡的IP地址。
    - 若Altas 200 DK通过网线连接用户PC机,可修改eth0网卡的IP地址。
  - b. 执行netplan apply命令重启网络服务。

# 12.13 设置用户有效期

为保证用户的安全性,应设置用户的有效期,使用系统命令chage来设置用户的有效期。

#### 命令为:

chage [-m mindays] [-M maxdays] [-d lastday] [-I inactive] [-E expiredate] [-W warndays] user

相关参数请参见表12-2。

#### 表 12-2 设置用户有效期

参数	参数说明
-m	两次修改密码的最小间隔天数。设置为"0"表示任何时候都可以 更改口令。
-M	口令保持有效的最大天数,为相对于上一次修改密码或者创建账号的天数。 设置为"-1"表示可删除这项口令的检测。设置为"-99999"表示无限期。
-d	上一次更改的日期。
-I	停滞时期。过期指定天数后,设定密码为失效状态。
-Е	用户到期的日期。超过该日期,此用户将不可用。
-W	用户口令到期前,提前收到警告信息的天数。
-l	列出当前的设置。由非特权用户来确定口令或帐户何时过期。

#### 山 说明

- 表12-2只列举出常用的参数,用户可通过chage --help命令查询详细的参数说明。
- 日期格式为YYYY-MM-DD,如chage -E 2020-12-01 *test*表示用户*test*的口令在2020年12月1日过期。
- User必须填写,填写时请替换为具体用户,默认为root用户。

#### 举例说明:

● 修改用户HwHiAiUser的有效期为2020年12月01日,用户HwHiAiUser的口令在2020年12月1日过期

chage -E 2020-12-01 HwHiAiUser

● 修改用户HwHiAiUser的有效期为90天,创建HwHiAiUser用户的天数或相对上一次修改密码的天数 chage -M 90 HwHiAiUser

## 12.14 相关工具

步骤1 以HwHiAiUser用户登录Atlas 200 DK。

步骤2 切换到root用户,执行表12-3中的命令调用工具。

表 12-3 相关工具

名称	工具所在路径	作用	执行命令
upgrade-tool	/var/davinci/ driver	查看固件版 本、升级单个 或多个固件 等,只允许 root用户执 行。	请切换到/var/davinci/driver路 径执行如下命令: ● 列举所有Device: /var/davinci/driver/ upgrade-tool mini_devices
			<ul> <li>获取指定设备的版本:         /var/davinci/driver/         upgrade-tool         device_index <dev_id>         system_version</dev_id></li> </ul>
			<ul> <li>获取指定设备的组件信息:         /var/davinci/driver/         upgrade-tool         device_index <dev_id>         components</dev_id></li> </ul>
			<ul> <li>查询某个设备中某一个组件 对应的版本: /var/davinci/driver/ upgrade-tool device_index <dev_id> component <type> version</type></dev_id></li> </ul>
			● 查询设备状态: /var/davinci/driver/ upgrade-tool device_index <dev_id> status</dev_id>
			<ul> <li>查询设备是否是物理机:         /var/davinci/driver/         upgrade-tool         device_index <dev_id>         phymachflag         说明         当前版本仅支持物理机升级固件包。</dev_id></li> </ul>
			<ul> <li>升级指定设备的固件包:         /var/davinci/driver/         upgrade-tool         device_index <dev_id>         component <type>path         <firmware_path></firmware_path></type></dev_id></li> </ul>
			● 异步升级指定设备的固件 包: /var/davinci/driver/ upgrade-tool

名称	工具所在路径	作用	执行命令
名称	工具所在路径	作用	## device_index <dev_id> component <type> asyncpath <firmware_path>async 必须紧跟在 component <type>后面。 如上命令中涉及的各参数含义如下。  •mini_devices: 所有设备列表  •device_index: 设备编号。取值可以是【0~63】和-1,【0~63】表示对应编号的设备。-1表示所有设备。  •system_version: 系统版本。  •components: 列举所有有效的组件。  •component: 指定具体的组件。升级单个组件时,需要指定组件名称(当前aicpu仅支持单个升级)。升级所有组件时,需要输入-1或-9,升级所有组件并重置密码时,需要输入-9。</type></firmware_path></type></dev_id>
			<ul> <li>version:组件的版本。</li> <li>status:设备状态。</li> <li>path:固件包相对路径。 升级单个固件(例如固件名称为nve.bin),则为:path/image/nve.bin 当前版本该工具不支持一次升级全部固件。</li> </ul>
			<ul> <li>hot_reset: 热复位 device。</li> <li>phymachflag: 查询设备 是否是物理机。针对Atlas 200 DK,此参数无需关注。</li> <li>async: 支持异步升级,即 device侧收到host侧发送的 升级请求后,返回请求成功响应到host侧(具体固件是 否升级成功,需要通过"</li> </ul>
			status"参数去查询)。 ●help: 查看帮助信息。

名称	工具所在路径	作用	执行命令
ascend_310_c heck.bin	/var/	安装driver的 时候会调用该 文件,作用 是:获取时间 戳新的crl文件 去校验要安装 的driver软件 包中的image 签名。	Driver包安装时,会自动调用此 文件进行合法性校验,不存在 用户单独调用此工具的场景。

----结束

# **13** FAQ

- 13.1 SD卡制作过程中人为拔卡导致系统存在冗余挂载盘
- 13.2 SSH登录时提示无法建立信任关系
- 13.3 开发者板无法正常连接PC机如何处理
- 13.4 Driver升级失败,报"CheckPartitionSpace partition space check failed"

## 13.1 SD 卡制作过程中人为拔卡导致系统存在冗余挂载盘

SD卡制作过程中人为拔卡导致系统出现冗余临时挂载盘,可使用如下步骤卸除。

步骤1 使用普通用户登录用户PC机,并执行su - root命令切换到root用户。

步骤2 输入命令df-h, 查看到/dev/loop0临时挂载盘。

root@kickseed:~# df -h

Filesystem Size Used Avail Use% Mounted on

/dev/loop0 745M 745M 0 100% /home/ubuntu/studio/scripts/180919002200 /dev/sdc1 118G 60M 112G 1% /home/ubuntu/studio/scripts/sd\_mount\_dir

步骤3 使用umount命令卸除挂载盘,命令中的/dev/loop0、/dev/sdc1设备请用户根据步骤2 实际查询结果调整。

root@kickseed:~# umount /dev/loop0
root@kickseed:~# umount /dev/sdc1

若命令提示target is busy,请尝试重启用户PC机,再重新步骤1至步骤3。

----结束

## 13.2 SSH 登录时提示无法建立信任关系

#### 问题描述

在PC机中以SSH方式登录到Atlas 200 DK时,提示无信任关系。

在PC机中执行如下命令重新建立信任关系:

ssh-keygen -f "\$HOME/.ssh/known\_hosts" -R 192.168.1.2

其中192.168.1.2为Atlas 200 DK开发者板的IP地址。

#### 报如下错误:

ECDSA host key for 192.168.1.2 has changed and you have requested strict checking.

#### 解决方法

出现此错误是本地保存的SSH信息已失效导致,所以需要清空当前保存的SSH信息,然后重新建立连接。

步骤1 清空PC机中当前用户连接192.168.1.2主机的公钥信息。

ssh-keygen -R 192.168.1.2

步骤2 重新以SSH方式连接Atlas 200 DK开发者板。

ssh HwHiAiUser@192.168.1.2

当提示如下信息时,输入yes重新建立SSH连接。

The authenticity of host '192.168.1.2' can't be established. ECDSA key fingerprint is 53:b9:f9:30:67:ec:34:88:e8:bc:2a:a4:6f:3e:97:95. Are you sure you want to continue connecting (yes/no)?

----结束

## 13.3 开发者板无法正常连接 PC 机如何处理

#### 问题描述

有以下几种现象:

- 将制作好的SD卡插入开发者板并上电后,开发者板LED1与LED2灯状态信息异常。
- 将制作好的SD卡插入开发者板,并通过USB方式连接PC机,上电、开发者板启动 完成后,PC机中无USB虚拟网卡信息。
- 将制作好的SD卡插入开发者板,并通过NIC方式连接PC机,上电、开发者板启动完成,配置PC机中NIC网卡信息后,PC机无法与开发者板通信。

#### 解决思路

按照如图13-1所示流程进行排查。

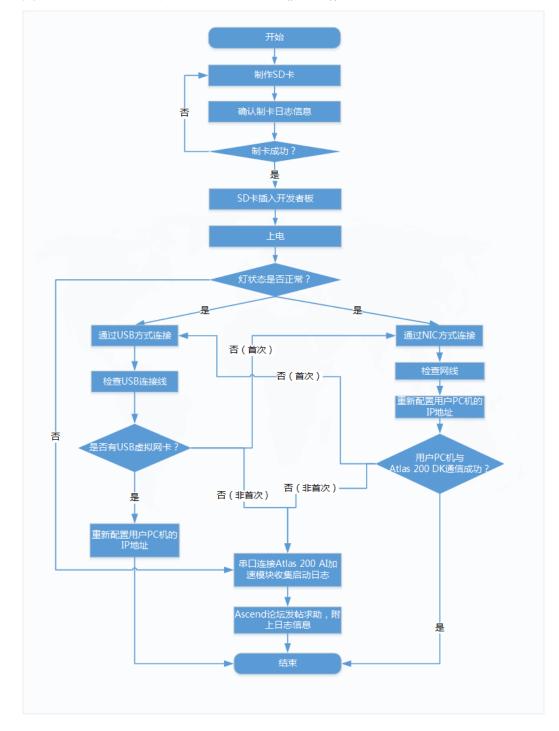


图 13-1 Atlas 200 DK 无法连接 PC 机问题排查思路

### 解决方法

步骤1 确认SD卡制作正确并成功。

请查看制卡脚本所在目录下的sd\_card\_making\_log查看是否制卡成功,若制卡失败,请参考**5 制作SD卡**重新制作SD卡。

步骤2 将成功制作的SD卡插入Atlas 200 DK开发者板,并上电。

- 若Atlas 200 DK开发者板的LED1与LED2状态正常,即启动成功后,LDE1与LDE2 同时处于亮的状态,则执行步骤3。
- 若Atlas 200 DK开发者板的LED1与LED2状态异常,即启动很长一段时间后(15分钟以后),LED1与LED2不同时为亮的状态,首先请参考异常处理查看系统软件安装日志及Atlas 200 DK启动日志。
   若仍无法解决,则执行步骤4。

#### 步骤3 连接Atlas 200 DK开发者板与PC机。

- 若通过USB方式连接开发者板,但PC机中不显示虚拟USB网卡。
   首先检查USB网线,确保USB网线两端口连接正常。
   若PC机中仍然不显示USB虚拟网卡,请尝试使用NIC方式连接。
- 若通过NIC方式连接开发者板,配置好PC机NIC网卡的IP信息后,PC机无法与开发者板通信。

首先检查网线,确保网线两端口正常,然后重新配置PC机的NIC网卡IP地址。若PC机仍然无法与开发者板通信,请尝试使用USB方式连接开发者板。

若USB方式与NIC方式,都无法正常连接PC机与开发者板,请执行步骤步骤4。

步骤4 参考12.6 通过串口连接Atlas 200 DK将Atlas 200 DK开发者板的Atlas 200 AI加速模块通过串口线与PC机相连。

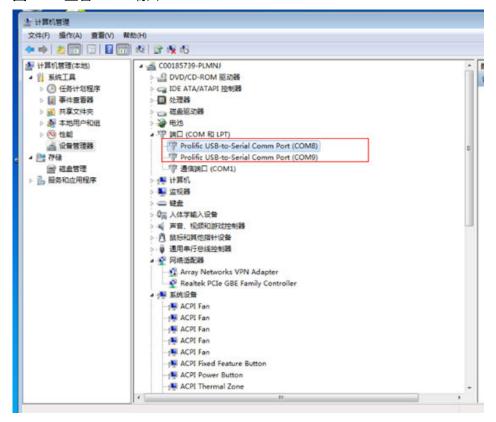
步骤5 在PC机中安装网络调试工具与USB转串口驱动软件。

- 网络调试工具推荐使用IPOP工具。
- USB转串口驱动软件请使用PL2303驱动软件。

步骤6 打开网络工程调试工具,以IPOP工具为例,进入串口窗口。

- 1. 单击"终端工具"页签。
- 2. 在菜单栏选择 ,进入"设置"窗口。
- 3. 进行连接配置。
  - 连接名称: 自定义连接名称。
  - 类型:选择COMX,可以通过计算机的设备管理器查看可用的COM端口,拔插用户PC机上的串口连接线,判断Atlas 200 DK使用的哪个COM口,如图 13-2所示。

#### 图 13-2 查看 COM 端口



- 设置波特率为115200。
- 4. 单击确定。
- 步骤7 上电Atlas 200 DK开发者板,在IPOP的COM连接窗口中查看Atlas 200 DK启动信息。由于启动日志较多,单击菜单栏中的 ❷ 将启动日志保存到IPOP工具的安装目录中,当此按钮变为 ❷ ,IPOP工具底部会出现文件已保存的提示信息,可以根据提示信息在IPOP安装目录中获取以当前时间命名的日志文件。
- 步骤8 在Ascend论坛上发求助帖,并将启动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。

----结束

## 13.4 Driver 升级失败,报"CheckPartitionSpace partition space check failed"

#### 问题描述

执行Atlas 200 DK上的Driver升级脚本"./minirc\_install\_phase1.sh"时,报如下图所示错误:

#### 图 13-3 Driver 升级错误

```
| Continue of the continue of
```

#### 可能原因

Atlas 200 DK的SD卡上未预留512MB的未分配空间用于驱动升级,可能是使用了旧的制卡脚本(最新的制卡脚本已修复该问题)。

可通过如下方法查看Atlas 200 DK上是否预留了未分配空间:

在Atlas 200 DK上以root用户执行如下命令:

#### fdisk -l

若回显如下图所示,可用扇区个数62333952,已分配扇区个数61285375,则未分配空间大小=(62333952-61285375)\*512/1024/1024=512MB)。

此SD卡已预留足够未分配空间,满足正常升级要求。

```
root@davinci-mini:/home/HwHiAiUser# fdisk -l
Disk /dev/mmcblk1: 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: 0xaf954ed3
Device
                     Start
                                 End Sectors Size Id Type
             Boot
/dev/mmcblk1p1
                      2048 10487807 10485760
                                                5G 83 Linux
/dev/mmcblk1p2
                   10487808 12584959 2097152
                                                1G 83 Linux
                   12584960 61285375 48700416 23.2G 83 Linux
/dev/mmcblk1p3
```

● 若回显如下图所示,可用扇区个数是6233952,已分配的扇区个数是6233951, 此SD卡未预留足够未分配空间,无法正常升级。

```
root@davinci-mini:/home/HwHiAiUser# fdisk -l
Disk /dev/mmcblk1: 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: 0xaf954ed3
Device
               Boot
                       Start
                                  End Sectors
                                                Size Id Type
/dev/mmcblk1p1
                                                  5G 83 Linux
                        2048 10487807 10485760
/dev/mmcblk1p2
                    10487808 12584959 2097152
                                                  1G 83 Linux
                    12584960 62333951 48700416 23.2G 83 Linux
/dev/mmcblk1p3
```

#### 解决方法

步骤1 将SD卡通过USB方式连接用户PC机。

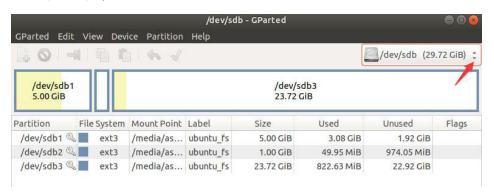
- 存在读卡器的场景,将SD卡插入读卡器,并将读卡器与用户PC机的USB口相连。
- 无读卡器场景,参见硬件准备使用跳线帽或者杜邦线将Atlas 200 DK的16号与18号针脚短接(下电状态下操作),将Atlas 200 DK与用户PC机的USB口连接,然后上电Atlas 200 DK。
- 步骤2 以root用户登录用户PC机
- 步骤3 执行如下命令安装磁盘分区工具Gparted。

#### apt-get install gparted

步骤4 执行如下命令,打开Gparted工具。

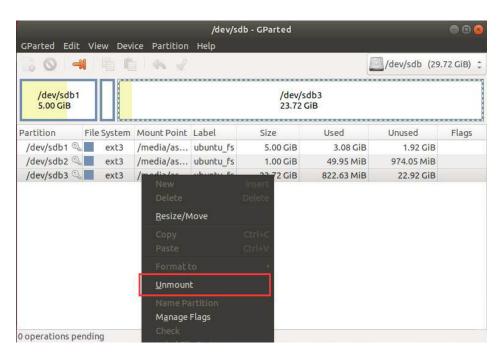
#### gparted

**步骤5** 切换到正确的设备(即SD卡所对应设备,可根据SD卡大小进行判断,下图所示为32G的SD卡的示例)。



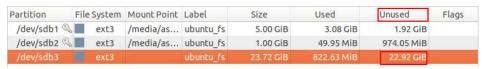
#### 步骤6 卸载分区。

右键单击"/dev/sdb3",选择"Unmount",卸载此分区。

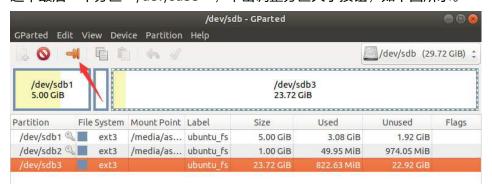


#### 步骤7 调整分区大小。

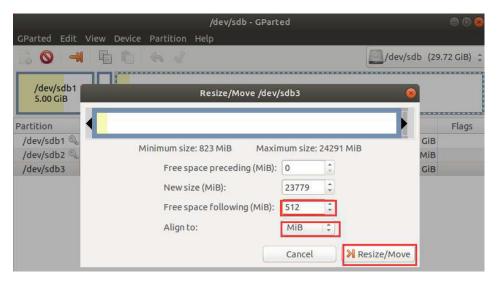
1. 请确保最后一个分区的"Unused"空间大于512MB。若剩余空间不够,请在Atlas 200 DK中将"/home/HwHiAiUser"目录下不需要的文件删除,预留出512MB,以免调整分区大小时丢失重要数据。



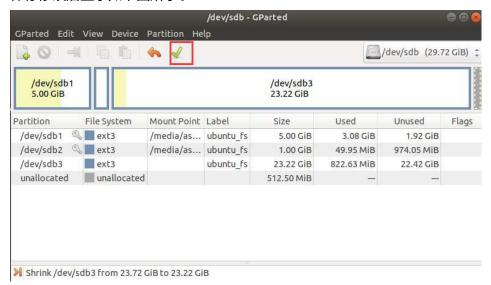
2. 选中最后一个分区"/dev/sdb3",单击调整分区大小按钮,如下图所示。



3. 在"Free space following(MiB)"中输入"512","Align to"选择"MIB",然后单击"Resize/Move"。 如下图所示。



4. 保存修改后显示如下图所示。



#### 步骤8 此时您已完成SD卡的分区的修改。

- 使用读卡器场景,请将SD卡插入Atlas 200 DK,上电后重新执行Atlas 200 DK的升级操作。
- 无读卡器场景,请下电Atlas 200 DK后,拔掉跳线帽/杜邦线,然后上电,重新执行Atlas 200 DK的升级操作。

#### ----结束