

MIN-EC07B

边缘整机

USER' Manual V10

USER'S MANUAL 用户手册

声明

本使用说明书内容如有变更，恕不另行通知。本公司并不对本使用说明书之适用性，适合做某种特殊用途之使用或其他任何事项做任何明示，或做其他形式之保证或担保。因此本公司将不对手册内容之错误，或因增减展示或以其他方式使用本手册所造成之直接，间接，突发性过、或继发性之损害负任何责任。

版权声明：

版权所有-----深圳智锐通科技有限公司。未经本公司许可或依著作权法之规定许准，不得复制，节录和翻译使用说明书之任何内容。

安全须知

1	产品使用前，务必仔细阅读产品说明书。
2	对未准备安装的板卡，应将其保存在防静电保护袋中。
3	在从包装袋中拿板卡前，应将手先置于接地金属物体上一会儿，以释放身体及手中的静电。
4	在拿板卡时，需佩戴静电保护手套，并且应该养成只触及边缘部份的习惯。
5	主板与电源连接时，请确认电源电压。
6	为避免人本被电击或产品被损坏，在每次对主板、板卡进行拔插或生新配置时须先关闭交流电源或将交流电源线从电源插座中拔掉。
7	在对板卡进行搬动前，先将交流电源线从电源插座中拔掉。
8	当您需连接或拔除任何设备前，须确定所有的电源线事先已被拔掉。
9	为避免频繁开关机对产品造成不必要的损伤,关机后,应至少等待30秒后再开机。
10	设备在使用过程时出现异常情况，请找专业人员处理。

目录

第一章 产品介绍.....	5
1.1 产品规格.....	5
1.2 产品料号.....	7
1.3 产品照片.....	7
1.4 尺寸图.....	9
1.5 接口引脚定义.....	9
第二章 安装说明.....	10
2.1 烧写系统.....	10
2.2 Recovery 模式.....	11
2.3 安装系统镜像.....	11
2.4 切换工作模式.....	12
2.5 Xshell 的使用.....	12
2.6 系统配置.....	15
2.7 查看系统版本.....	16
2.8 制作备份镜像.....	16
2.9 安装 Jtop 工具.....	17
附录.....	18
附一：术语表.....	18
附二：Jetpack 的主要功能.....	18

第一章 产品介绍

1.1 产品规格

Model		MIN-EC07B			
产品类型 Form Factor	产品类型 Form Factor	边缘整机			
模块 Module	型号 Model	MIN-EC07B (Nano 4G)	MIN-EC07B (Nano 8G)	MIN-EC07B (NX 8G)	MIN-EC07B (NX 16G)
	AI Performance	20TOPS	40TOPS	70TOPS	100TOPS
	功耗 TDP	5~10W	7~15W	10~20W	10~25W
INT8	Sparse	20TOPS	40TOPS	70TOPS	100TOPS
	Dense	10TOPS	20TOPS	35TOPS	50TOPS
CPU	架构 Architecture	Arm Cortex A78AE v8.2 64bit	Arm Cortex A78AE v8.2 64bit	Arm Cortex A78AE v8.2 64bit	Arm Cortex A78AE v8.2 64bit
	Core	6Core	6Core	6Core	6Core
	L2	1.5MB	1.5MB	1.5MB	2MB
	L3	4MB	4MB	4MB	4MB
GPU	架构 Architecture	NVIDIA Ampere	NVIDIA Ampere	NVIDIA Ampere	NVIDIA Ampere
	Tensor Cores	16 Tensor Cores	32 Tensor Cores	32 Tensor Cores	32 Tensor Cores
	Cuda	512Core	1024Core	1024Core	1024Core
内存 Memory	容量 Size	4GB	8GB	8GB	16GB
	位宽 Bus Width	64bit	64bit	128bit	128bit
	带宽 Bandwidth	34GB/s	68GB/s	102.4GB/s	102.4GB/s
扩展插槽 Expansion Slot	M.2	1 x M.2 Key M			

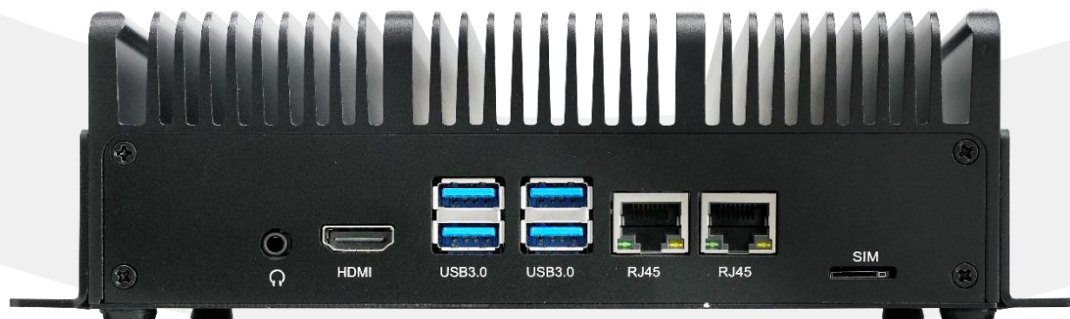
视频 Video	视频编码 Video Encode	1080p30 Supported By 1-2 CPU Cores	1x 4K60(H.265) 3x 4K30(H.265) 6x 1080p60(H.265) 12x 1080p30(H.265) 1x 4K60(H.264) 2x 4K30(H.264) 5x 1080p60(H.264) 11x 1080p30(H.264)
	视频解码 Video Decode	1x 4K60(H.265) 2x 4K30(H.265) 5x 1080p60(H.265) 11x 1080p30(H.265)	1x 8K30(H.265) 2x 4K60(H.265) 4x 4K30(H.265) 9x 1080p60(H.265) 18x 1080p30(H.265) 1x 4K60(H.264) 2x 4K30(H.264) 5x 1080p60(H.264) 11x 1080p30(H.264)
网口 Ethernet	前面板 Front I/O	2 x RJ45 10/100/1000M	
USB	前面板 Front I/O	4 x USB3.0 Type A	
显示 Display	前面板 Front I/O	1 x HDMI	
SIM	前面板 Front I/O	1 x SIM	
音频 Audio	前面板 Front I/O	1 x Audio	
Type C	后面板 Rear I/O	1 x Type C	
按钮 Button	后面板 Rear I/O	1 x Recovery 1 x Reset 1 x Power	
CAN	后面板 Rear I/O	1 x CAN	
串口 COM	后面板 Rear I/O	1 x RS485 1 x RS232	
SPI Bus	后面板 Rear I/O	1 x SPI Bus(3.3V Level)	
电源 Power Requirements	电源瓦数 Consumption	70W	
	电源电压 Input Voltage	12v(7A)~36v(3A) DC Input	

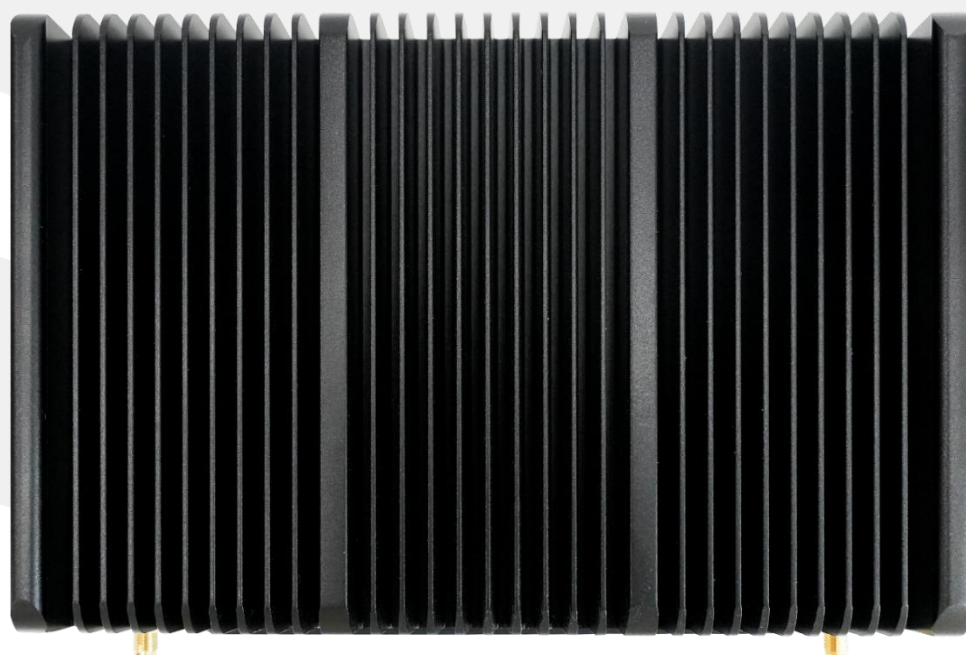
环境 Environment	工作温度 Operating Temperature	-25~60°C
	存储温度 Storage Temperature	-20~60°C
	工作湿度 Operating Humidity	10~90%(non-condensing)
物理特性 Physical	尺寸 Dimensions	224*130*65.5mm
	颜色 Color	Black
	重量 Weight	2KG

1.2 产品料号

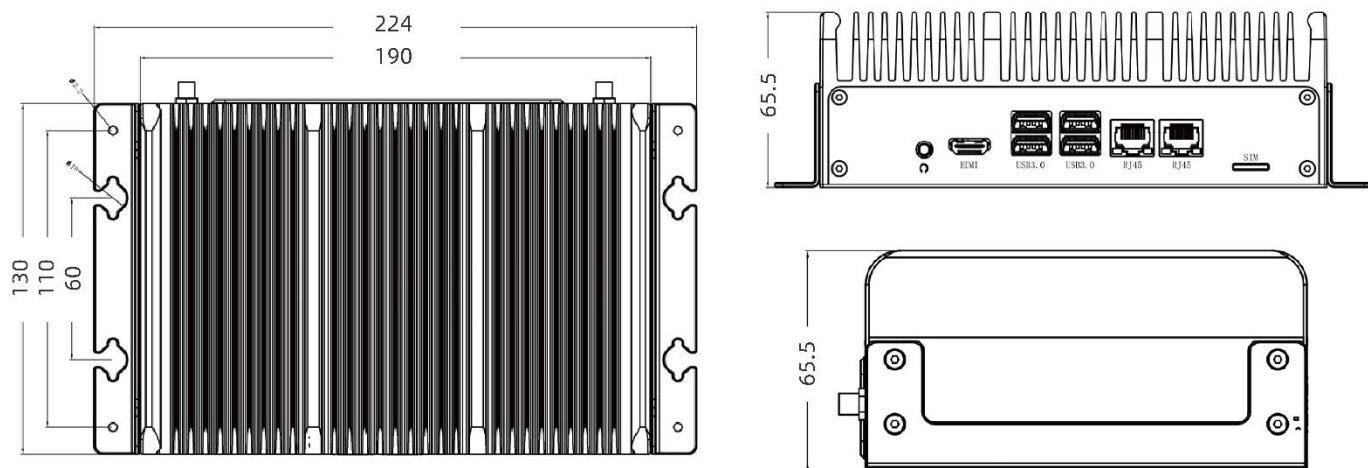
Model	Part Number	Specification
MIN-EC07B(Nano 8G)	8.ZRT.31-0048-00-LFF	MIN-EC07B-Nano 整机 底板 Jetson orin nano 核心板 显存: 8G 存储: 128G 固态 HDMI 显示接口/双千兆网口/电源适配器/包装
MIN-EC07B(NX 8G)	8.ZRT.31-0049-00-LFF	MIN-EC07B-NX 整机 底板 Jetson orin NX 核心板 显存: 8G 存储: 128G 固态 HDMI 显示接口/双千兆网口/电源适配器/包装
MIN-EC07B(NX 16G)	8.ZRT.31-0050-00-LFF	MIN-EC07B-NX-16 整机 底板 Jetson orin NX 核心板 显存: 16G 存储: 512G 固态 HDMI 显示接口/双千兆网口/电源适配器/包装

1.3 产品照片





1.4 尺寸图



1.5 接口引脚定义

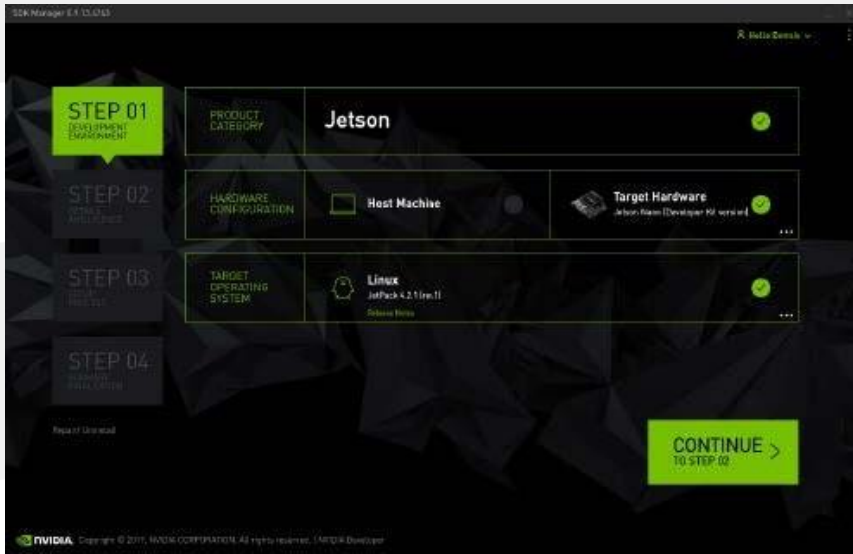
Pin	Signal Name	Pin	Signal Name	Pin	Signal Name	Pin	Signal Name
1	CAN_H			2	CAN_L		
3	RS232_RX			4	RS232_TX		
5	RS485_B			6	RS485_A		
7	I2C1_SCL(3.3V)	189	GEN2_I2C_SCL	8	I2C1_SDA(3.3V)	191	GEN2_I2C_SDA
9	3.3V			10	GND		
11	UART2_RXD_3V3	238	UART1_RXD	12	UART2_TXD_3V3	236	UART2_TXD
13	SPI0_SCK_3V3	91	SPI1_SCK	14	SPI0_MOSI_3V3	89	SPI1_MOSI
15	SPI0_CS0_3V3	95	SPI1_CS0	16	SPI0_MISO_3V3	93	SPI1_MISO
17	GPIO2_3V3	124	GPIO_PH6	18	SPI0_SC1_3V3	97	SPI1_CS1
19	5V			20	GND		

注：

- 1、UART2 经过载板电平转换电路转换为 3.3V 逻辑电平。
- 2、SPI0 经过载板电平转换电路转换为 3.3V 逻辑电平。
- 3、GPIO2_3V3 是 3.3V 逻辑电平。

第二章 安装说明

2.1 烧写系统

步骤	说明	图示
1	将系统镜像包下载到 Ubuntu18.04 系统的 PC 主机上。	
2	用 USB 线将 Ubuntu18.04 系统的 PC 电脑的 USB Type-A 与 EC07B 的 Type-C 相连。	
3	给 EC07B 上电, 并进入 Recovery 模式。	
4	PC 电脑打开 Nvidia-SDK-Manager, 在界面中选择 Jetson Orin NX /Orin Nano 下载 Jetpack5.xx 系统镜像包和开发工具。或者从 https://developer.nvidia.com/embedded/downloads 下载最新的 Jetson Linux 发布包和 Jetson 开发工具包的示例文件系统。(Jetson Linux Driver Package (L4T))	
5	解压下载下来的镜像包, 并进入 Linux for Tegra(L4T)目录。	<pre> 1 tar xf \${L4T_RELEASE_PACKAGE} 2 cd Linux_for_tegra/rootfs 3 sudo tar xpf \${SAMPLE_FS_PACKAGE} 4 cd .. 5 sudo ./tools/l4t_flash_prerequisites.sh 6 sudo ./apply_binaries.sh 7 #替换驱动包 8 sudo cp -r 607_xxx/Linux_for_tegra /Linux_for_tegra </pre>
6	进入 <i>Linux_for_tegra</i> 目录使用刷机命令 <i>flash to Nvme</i> 。	<pre> 1 #flash to nvme 2 #orin nano 3 sudo ./tools/kernel_flash/l4t_initrd_flash.sh --external-device nvme0n1p1 -c tools/kernel_flash/flash_l4t_external.xml -p "-c bootloader/t186ref/cfg/flash_t234_qspi.xml" --showlogs --network usb0 jetson-orin-nano-devkit internal 4 #orin nx 5 sudo ./tools/kernel_flash/l4t_initrd_flash.sh --external-device nvme0n1p1 -c tools/kernel_flash/flash_l4t_external.xml -p "-c bootloader/t186ref/cfg/flash_t234_qspi.xml" --showlogs --network usb0 p3509-a02+p3767-0000 internal </pre>

7	进入 <i>Linux_for_tegra</i> 目录使用刷机命令 <i>flash to USB</i> 。	<pre> 1 #flash to USB: 2 #orin nano 3 sudo ./tools/kernel_flash/l4t_initrd_flash.sh --external-device sda1 -c tools/kernel_flash/flash_l4t_external.xml -p "-c bootloader/t186ref/cfg/flash_t234_qspi.xml" --showlogs --network usb0 jetson- orin-nano-devkit internal 4 #orin nx 5 sudo ./tools/kernel_flash/l4t_initrd_flash.sh --external-device sda1 -c tools/kernel_flash/flash_l4t_external.xml -p "-c bootloader/t186ref/cfg/flash_t234_qspi.xml" --showlogs --network usb0 p3509- a02+p3767-0000 internal </pre>
8	进入 <i>Linux_for_tegra</i> 目录使用刷机命令 <i>flash to SD</i> 。	<pre> 1 #flash to SD 2 #orin nano 3 sudo ./tools/kernel_flash/l4t_initrd_flash.sh --external-device mmcblk1p1 -c tools/kernel_flash/flash_l4t_external.xml -p "-c bootloader/t186ref/cfg/flash_t234_qspi.xml" --showlogs --network usb0 jetson- orin-nano-devkit internal </pre>

2.2 Recovery 模式

可以通过 USB 进行系统更新，更新需要进入 USB Recovery 模式。USB Recovery 模式下可以进行文件系统更新，内核更新，boot loader 更新，BCT 更新等操作。进入 Recovery 模式的步骤如下：

步骤	说明
1	关闭系统电源，请确保使电源关闭而不是进入待机状态。
2	使用 USB Type-C 到 USB Type A 的链接线链接载板和主机。
3	对设备上电。
4	保持按下 Recovery 按键的同时按下并释放 Reset 按键，等待 2 秒释放 Recovery 按键。

注：在进入 USB Recovery 模式下，系统不会启动，串口不会有调试信息输出。

2.3 安装系统镜像

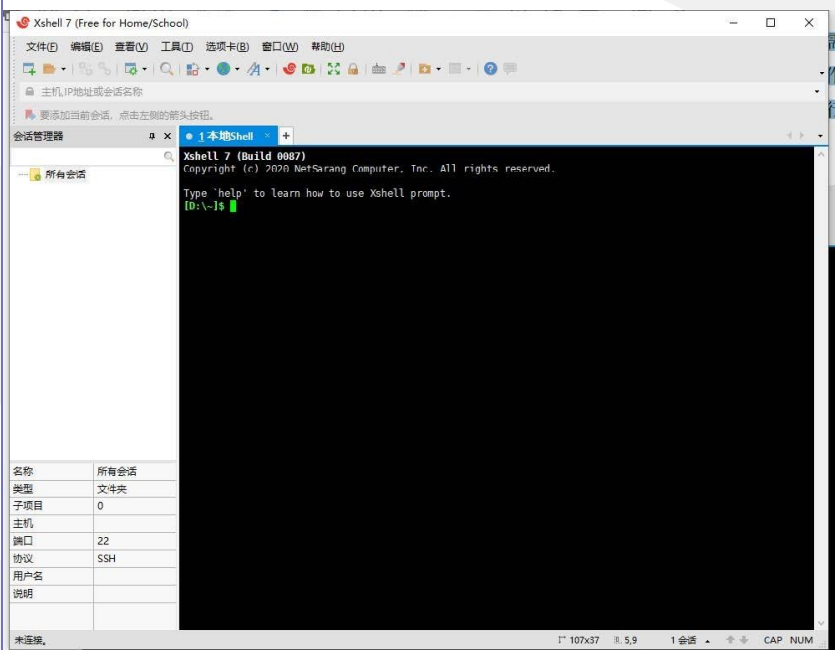
步骤	说明	图示
1	将 Ubuntu18.04 主机的 USB Type-A 与 MIN-EC07B 的 Type-c 相连。	<pre> 1 cd Linux_for_tegra 2 #flash to nvme 3 #orin nano 4 sudo ./tools/kernel_flash/l4t_initrd_flash.sh --external-device nvme0n1p1 -c tools/kernel_flash/flash_l4t_external.xml -p "-c bootloader/t186ref/cfg/flash_t234_qspi.xml" --showlogs --network usb0 jetson- orin-nano-devkit internal 5 #orin nx 6 sudo ./tools/kernel_flash/l4t_initrd_flash.sh --external-device nvme0n1p1 -c tools/kernel_flash/flash_l4t_external.xml -p "-c bootloader/t186ref/cfg/flash_t234_qspi.xml" --showlogs --network usb0 p3509- a02+p3767-0000 internal </pre>
2	给 MIN-EC07B 上电，并进入 Recovery 模式。	
3	PC 主机进入 L4T 目录，执行刷机指令。	
4	刷机完成后，给 MIN-EC07B 重新上电登录系统。	

2.4 切换工作模式

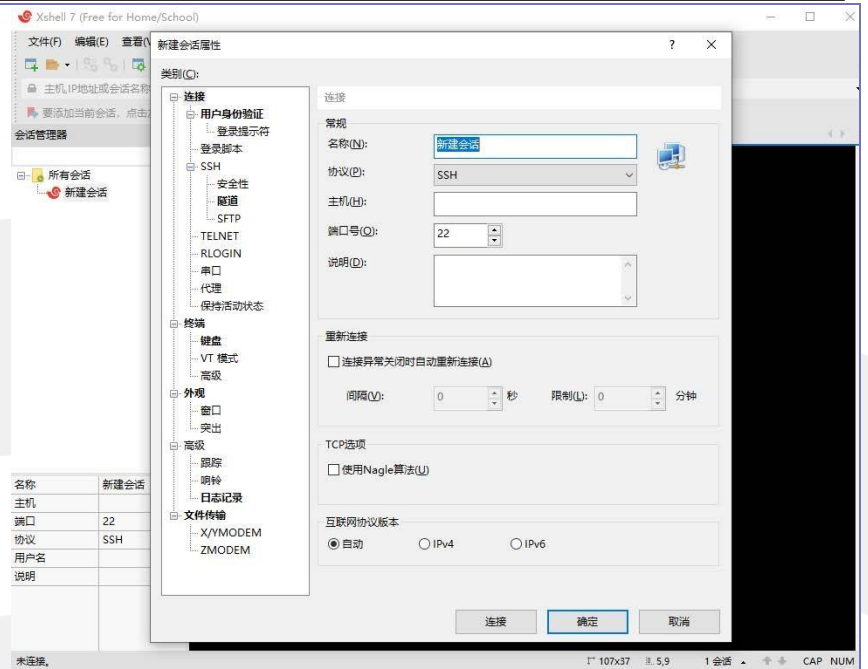
步骤	说明	图示
1	登录系统后，可在系统界面右上角点击操作修改。	
2	或者，在终端输入命令进行切换。	<pre> 1 # 切换到模式0,可用-q参数查看当前模式 Switch to mode 0, use the -q parameter to view the current mode 2 sudo nvpmode1 -m 0 </pre>

2.5 Xshell 的使用

Xshell 是一个强大的安全终端模拟软件，它支持 SSH1, SSH2 以及 Microsoft Windows 平台的 TELNET 协议。Xshell 通过互联网到远程主机的安全连接以及它创新性的设计和特色帮助用户在复杂的网络环境中享受他们的工作。Xshell 可以在 Windows 界面下用来访问远端不同系统下的服务器，从而比较好的达到远程控制终端的目的。Xshell 不是必需品，但是它能更好的辅助我们使用设备，可以将 Windows 系统和 Ubuntu 系统连接起来，在 Windows 系统下，去操作 Linux 系统。安装 Xshell 可以在网上搜索下载安装即可。(当产品无法进入桌面系统的情况时，也可以通过 Xshell 来进行远程控制，修改配置上的错误)

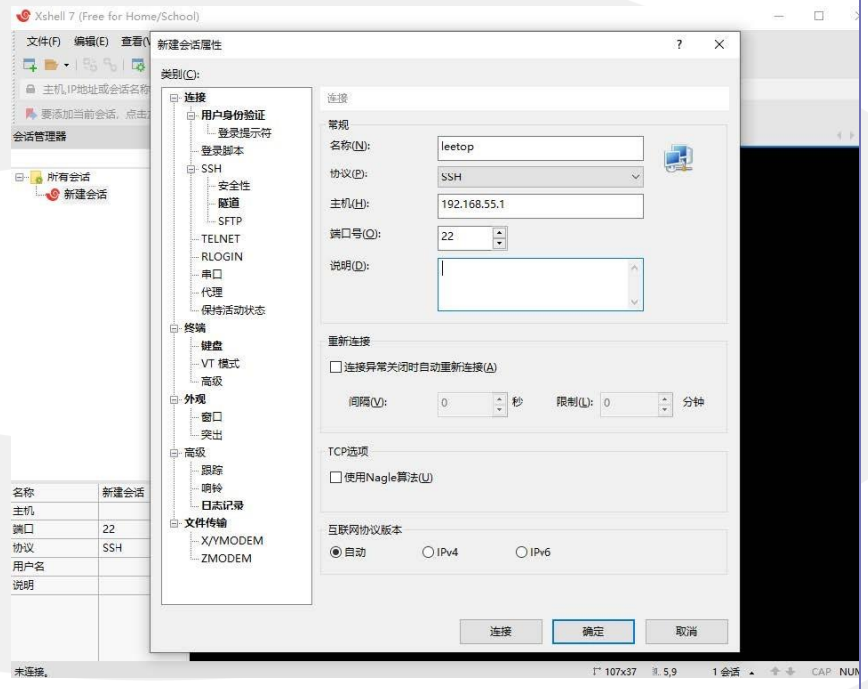
步骤	说明	图示																
1	打开 Xshell。	 <table border="1" data-bbox="646 1534 821 1713"> <thead> <tr> <th>名称</th> <th>所有会话</th> </tr> </thead> <tbody> <tr> <td>类型</td> <td>文件夹</td> </tr> <tr> <td>子项目</td> <td>0</td> </tr> <tr> <td>主机</td> <td></td> </tr> <tr> <td>端口</td> <td>22</td> </tr> <tr> <td>协议</td> <td>SSH</td> </tr> <tr> <td>用户名</td> <td></td> </tr> <tr> <td>说明</td> <td></td> </tr> </tbody> </table>	名称	所有会话	类型	文件夹	子项目	0	主机		端口	22	协议	SSH	用户名		说明	
名称	所有会话																	
类型	文件夹																	
子项目	0																	
主机																		
端口	22																	
协议	SSH																	
用户名																		
说明																		

2 新建。

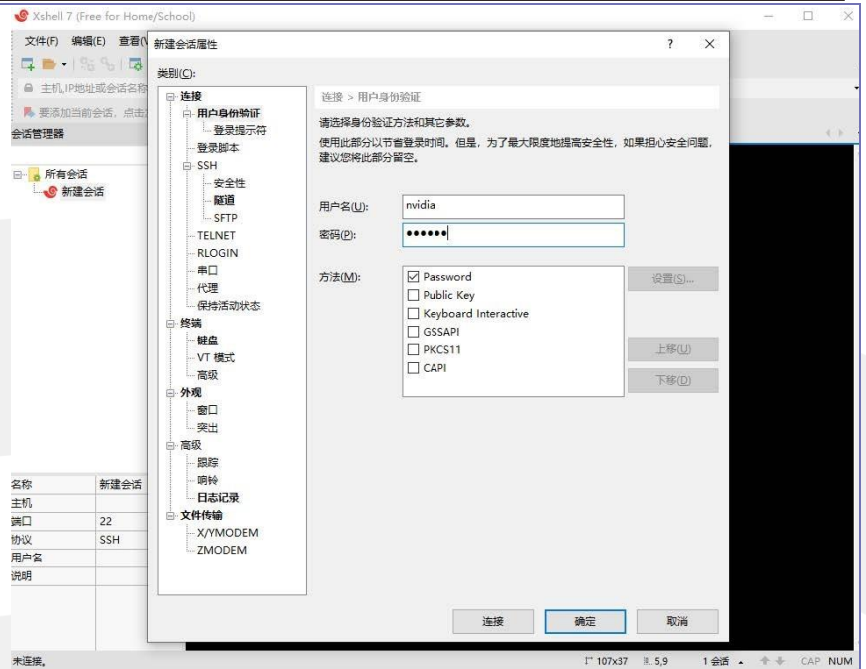


3

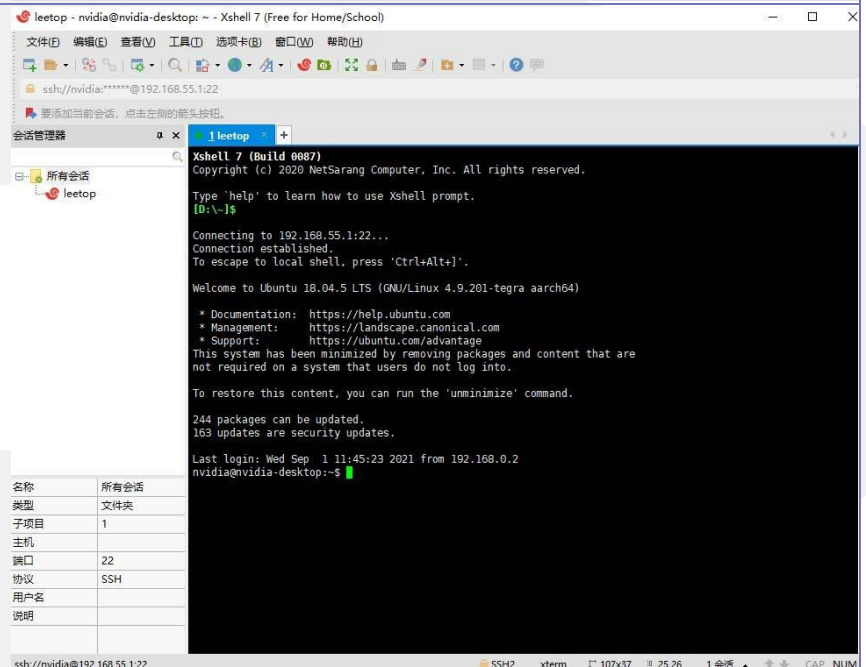
填写名称以及主机 IP
(正常情况下可以通过网络 IP 来连接, 若不知道 IP 的情况下, 可以通过 USB 数据线连接电脑和设备 OTG 口, 填写固定 IP 来进行连接。

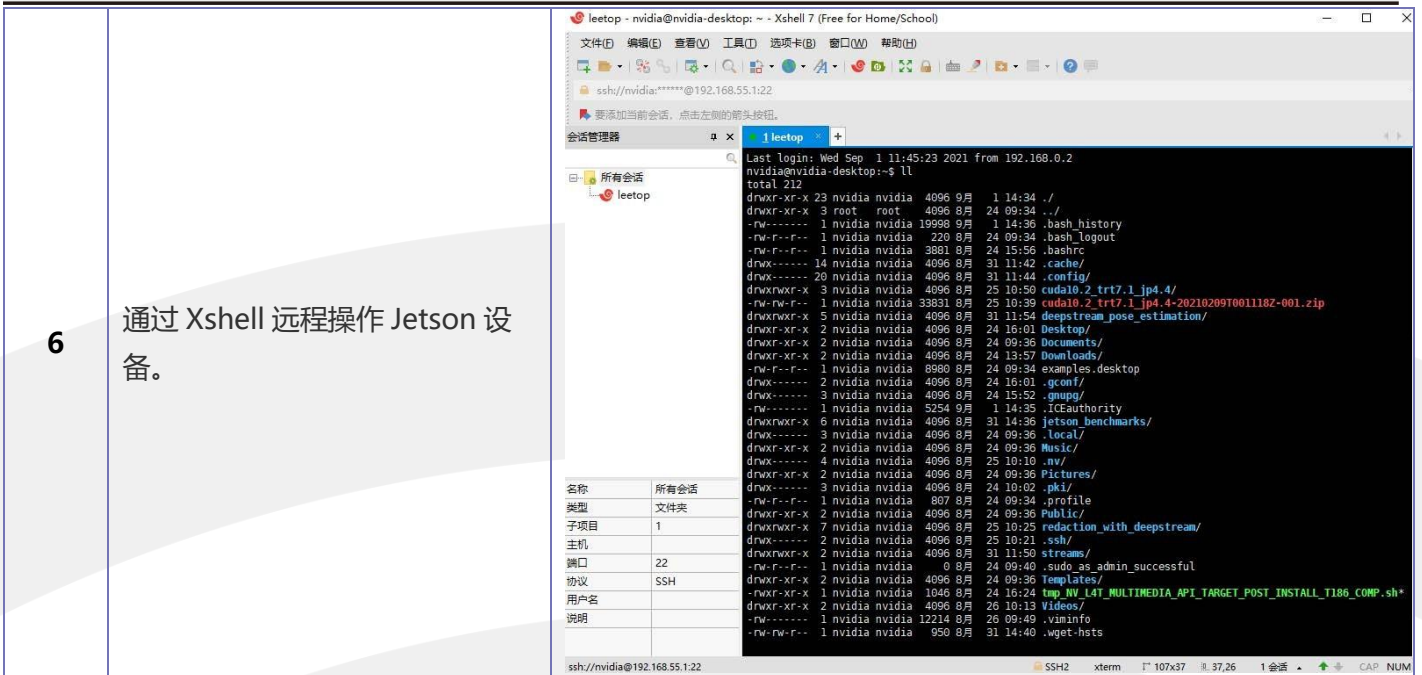


4 输入用户和密码。



5 点击连接进入命令行界面。





2.6 系统配置

2.6.1 NVIDIA Linux For Tegra (L4T)

载板支持原生 NVIDIA Linux For Tegra (L4T) Builds。HDMI、千兆以太网、USB3.0、USB OTG、串口、GPIO、I2C 总线均可得到支持，无须更改可直接运行在 MIN-EC07B 上。

详细说明和工具下载链接：

<https://developer.nvidia.com/embedded/jetson-linux-r3521>

<http://s://developer.nvidia.com/embedded/jetson-linux-r3531>

NVIDIA Jetpack for L4T

Jetpack 使一个 NVIDIA 发布的软件包，包含了使用 MIN-EC07B 进行 Orin NX / Orin Nano 开发所需要的各种软件工具，包含主机端和目标机端各种工具，包括 OS 镜像文件，中间件，示例程序，文档等内容。最新发布的 JetPack 运行在 Ubuntu 20.04 Linux 64 位主机上。

可从下面链接进行下载：<https://developer.nvidia.com/embedded/jetpack>

默认配置系统

MIN-EC07B 采用 Ubuntu 20.04 系统，默认用户名：nvidia 密码：nvidia

开发资料及论坛

L4T 开发资料：<https://developer.nvidia.com/embedded/linux-tegra>

开发者论坛：<https://forums.developer.nvidia.com/>

2.7 查看系统版本

查看安装的系统包版本。

```
1 | cat /etc/nv_tegra_release
```

查看安装的 JetPack 信息。

```
1 | sudo apt show nvidia-jetpack
```

2.8 制作备份镜像

制作备份镜像需在命令行刷机的环境下进行，只备份 system.img 文件。

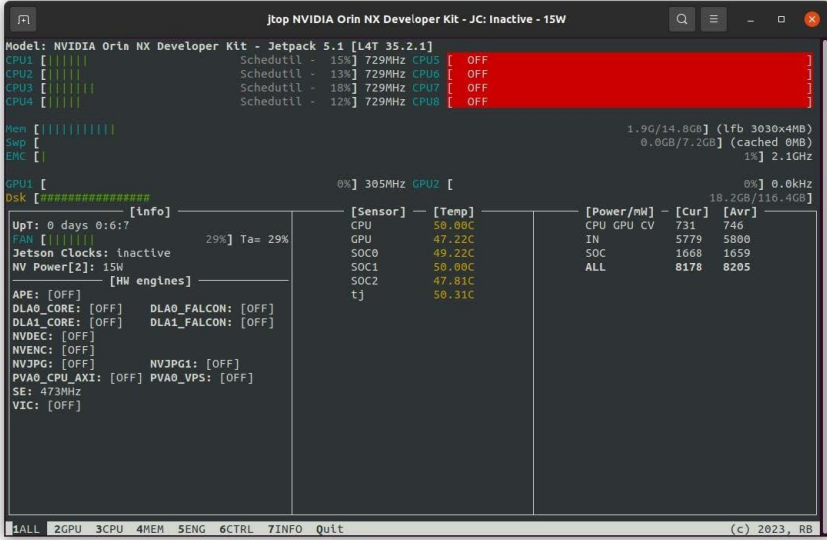
步骤	说明	图示
1	用 USB 线将 Ubuntu18.04 系统的 PC 电脑的 USB Type-A 与 MIN-EC07B 的 Type-C 相连。	<pre> 1 cd Linux_for_Tegra/tools/backup_restore 2 # 将以下文件中mmcblk0替换为nvme0n1 Replace mmcblk0 with nvme0n1 in the following files. 3 sudo sed -i 's/mmcblk0/nvme0n1/g' l4t_backup_restore.sh 4 sudo sed -i 's/mmcblk0/nvme0n1/g' l4t_backup_restore.func 5 sudo sed -i 's/mmcblk0/nvme0n1/g' nvbackup_partitions.sh 6 sudo sed -i 's/mmcblk0/nvme0n1/g' nvrestore_partitions.sh 7 cd ../../ 8 #须暂时禁用新外部存储设备的自动挂载备份恢复。Automount of new external storage devices must be temporarily disabled during backing up and restoring 9 systemctl stop udisks2.service 10 sudo tools/l4t_flash_prerequisites.sh # For Debian-based Linux 11 sudo service nfs-kernel-server start 12 sudo ./tools/backup_restore/l4t_backup_restore.sh -b <board-name> 13 #备份镜像，备份成功后在Linux_for_Tegra/tools/backup_restore下生成image文件。 Backup image, after the backup is successful, an image file will be generated under Linux_for_Tegra/tools/backup_restore. 14 sudo ./tools/backup_restore/l4t_backup_restore.sh -r <board-name> 15 #备份镜像恢复。Backup image restoration.</pre>
2	给 MIN-EC07B 上电，并进入 Recovery 模式。	
3	进入 <i>Linux_for_tegra</i> 目录，参照 backup_restore 中 README_backup_restore.txt 进行备份。	
4	备份 Jetson Orin Nano/NX 系统的指令。	
5	使用备份的镜像进行刷机。能正常刷机使用表示制作的备份镜像可用。	

```

1 | sudo ./tools/backup_restore/l4t_backup_restore.sh -r <board-name>
2 | #备份镜像恢复。Backup image restoration.
```


2.9 安装 Jtop 工具

Jtop 是一个 Jetson 的系统监视实用程序，可在终端上运行，可实时查看和控制 NVIDIA Jetson 的状态。

步骤	说明	图示
1	安装 pip3 工具。	1 <code>sudo apt-get install python3-pip</code>
2	用 pip3 安装 jtop 包。	1 <code>sudo -H pip3 install -U jetson-stats</code>
3	重启后可运行 jtop。	1 <code>jtop</code>
4	运行后如图所示	

附录

附一：术语表

JetPack

NVIDIA JetPack SDK 是构建 AI 应用程序的最全面的解决方案。它捆绑了 Jetson 平台软件，包括 TensorRT、cuDNN、CUDA Toolkit、VisionWorks、GStreamer 和 OpenCV，所有这些软件都构建在带有 LTS Linux 内核的 L4T 之上。

L4T

英伟达 L4T 为 Jetson 平台提供了 Linux 内核、引导程序、驱动程序、闪光实用程序、样本文件系统等。你可以对 L4T 软件进行定制，以适应你的项目的需要。通过遵循平台适应性和带入指南，你可以优化使用完整的 Jetson 产品功能集。遵循以下链接，了解最新的软件库、框架和源代码包的详细信息。

<https://docs.nvidia.com/jetson/archives/r35.2.1/DeveloperGuide/index.html>

DeepStream SDK on Jetson

NVIDIA 的 DeepStream SDK 为基于 AI 的多传感器处理、视频和图像理解提供了完整的流分析工具包。DeepStream 是 NVIDIA Metropolis 的一个组成部分，该平台用于构建端到端服务和将像素和传感器数据转化为可操作见解的解决方案。

Isaac SDK

NVIDIA Isaac SDK 使开发人员可以轻松创建和部署 AI 驱动的机器人。SDK 包括 Isaac Engine（应用程序框架）、Isaac GEM（具有高性能机器人算法的包）、Isaac Apps（参考应用程序）和 Isaac Sim for Navigation（一个强大的模拟平台）。这些工具和 API 可以更轻松地将用于感知和导航的人工智能（AI）添加到机器人中，从而加速机器人的开发。

附二：Jetpack 的主要功能

功能	说明
TensorRT	TensorRT 是一种用于图像分类、分割和对象检测神经网络的高性能深度学习推理运行。TensorRT 基于 NVIDIA 的并行编程模型 CUDA 构建，使您能够优化所有深度学习框架的推理。它包括一个深度学习推理优化器和运行时，可为深度学习推理应用程序提供低延迟和高吞吐量。JetPack 5.1.1 includes TensorRT 8.5.2。
cuDNN	CUDA 深度神经网络库为深度学习框架提供高性能原语。它为标准例程提供高度调整的实现，例如前向和反向卷积、池化、归一化和激活层。JetPack 5.1.1 includes cuDNN 8.6.0。
CUDA	CUDA 工具包为构建 GPU 加速应用程序的 C 和 C++ 开发人员提供了一个全面的开发环境。该工具包包括用于 NVIDIA GPU 的编译器、数学库以及用于调试和优化应用程序性能的工具。

	<p>JetPack 5.1.1 includes CUDA 11.4.19 从 JetPack 5.0.2 开始，从 CUDA 11.8 升级到最新和最好的 CUDA 版本，而无需更新 Jetson Linux 的其他 JetPack 组件。请参阅 CUDA 文档中的说明，了解如何在 JetPack 上获取最新的 CUDA。</p>
<p>Multimedia</p> <p>API</p>	<p>Jetson 多媒体 API 包为灵活的应用程序开发提供低级 API。</p> <p>相机应用程序 API: libargus 为相机应用程序提供低级帧同步 API，具有每帧相机参数控制、多个（包括同步）相机支持和 EGL 流输出。需要 ISP 的 RAW 输出 CSI 相机可以与 libargus 或 GStreamer 插件一起使用。在任何一种情况下，都会使用 V4L2 媒体控制器传感器驱动程序 API。</p> <p>传感器驱动程序 API: V4L2 API 支持视频解码、编码、格式转换和缩放功能。用于编码的 V4L2 开放了许多功能，如比特率控制、质量预设、低延迟编码、时间权衡、运动向量图等。</p> <p>JetPack 5.1.1 相机亮点包括：支持 Orin 上的多点镜头阴影校正 (LSC)。增强了 Argus SyncStereo 应用程序的弹性，以保持立体相机对之间的同步。</p> <p>JetPack 5.1.1 多媒体亮点包括：支持 AV1 编码中的动态帧率用于演示 CPU 内核上的软件编码的新 argus_camera_sw_encode 示例使用 CPU 内核上的软件编码选项更新了 nvgstcapture-1.0。</p>
<p>Computer</p> <p>Vision</p>	<p>VPI (视觉编程接口) 是一个软件库，它提供在 Jetson 上的多个硬件加速器上实现的计算机视觉/图像处理算法，例如 PVA (可编程视觉加速器)、GPU、NVDEC (NVIDIA 解码器)、NVENC (NVIDIA 编码器)、VIC (视频图像合成器) 等。</p> <p>OpenCV 是一个用于计算机视觉、图像处理和机器学习的开源库。</p> <p>JetPack 5.1.1 包含对 VPI 2.2 的小更新，并修复了错误 JetPack 5.1.1 包含 OpenCV 4.5.4。</p>
<p>Graphics</p>	<p>JetPack 5.1.1 包括以下图形库: Vulkan® 1.3 (包括 Roadmap 2022 Profile)。Vulkan 1.3 公告 Vulkan® SC 1.0 Vulkan SC 是一个低级的、确定性的、健壮的 API，它基于 Vulkan 1.2。此 API 可实现最先进的 GPU 加速图形和计算，这些图形和计算可部署在安全关键型系统中，并经认证符合行业功能安全标准。有关详细信息，请参阅 https://www.khronos.org/vulkansc/。</p> <p>Vulkan SC 对于实时非安全关键型嵌入式应用程序也具有无可估量的价值。Vulkan SC 通过尽可能多地将运行时应用程序环境的准备转移到离线或应用程序设置中来提高确定性并减小应用程序大小。这包括定义 GPU 如何处理数据的图形管道的离线编译，以及静态内存分配，它们共同实现了可以严格指定和测试的详细 GPU 控制。</p> <p>Vulkan SC1.0 从 Vulkan 1.2 演变而来，包括：删除安全关键市场不需要的运行时功能，更新设计以提供可预测的执行时间和结果，以及消除其操作中潜在歧义的说明。有关详细信息，请参阅 https://www.khronos.org/blog/vulkan-sc-overview</p> <p>注意：Jetson 对 Vulkan SC 的支持未经安全认证。OpenWF™ 显示 1.0 OpenWF Display 是一种 Khronos API，用于与 Jetson 上的本机显示驱动程序进行低开销交互，并允许与 Vulkan SC 交互以显示图像。</p> <p>注意：Jetson 对 OpenWF Display 的支持未经安全认证。</p>
<p>Developer</p> <p>Tools</p>	<p>CUDA 工具包为使用 CUDA 库构建高性能 GPU 加速应用程序的 C 和 C++ 开发人员提供了一个全面的开发环境。该工具包包括 Nsight Visual Studio Code Edition、Nsight Eclipse 插件、包括 Nsight Compute 在内的调试和分析工具，以及用于交叉编译应用程序的工具链 NVIDIA Nsight Systems 是一种低开销的系统级分析工具，可为开发人员提供分析和优化软件性能所需</p>

	<p>的见解。NVIDIA Nsight Graphics 是一个独立的应用程序，用于调试和分析图形应用程序。NVIDIA Nsight Deep Learning Designer 是一个集成开发环境，可帮助开发人员高效地设计和开发用于应用内推理的深度神经网络。Jetson Orin 模块支持 Nsight System、Nsight Graphics 和 Nsight Compute，以协助开发自主机器。</p> <p>JetPack 5.1.1 包括 NVIDIA Nsight Systems v2022.5。JetPack 5.1.1 包含 NVIDIA Nsight Graphics 2022.6。JetPack 5.1.1 包含 NVIDIA Nsight Deep Learning Designer 2022.2。</p>
Supported SDKs and Tools	<p>NVIDIA DeepStream SDK 是一个完整的分析工具包，用于基于 AI 的多传感器处理以及视频和音频理解。DeepStream 6.2 版本支持 JetPack 5.1.1 NVIDIA Triton™推理服务器可大规模简化 AI 模型的部署。Triton 推理服务器是开源的，支持在 Jetson 上部署来自 NVIDIA TensorRT、TensorFlow 和 ONNX Runtime 的经过训练的 AI 模型。在 Jetson 上，Triton 推理服务器作为共享库提供，用于与 C API 直接集成。PowerEstimator 是一个网络应用程序，可简化自定义电源模式配置文件的创建并估算 Jetson 模块功耗。JetPack 5.1.1 支持 Jetson AGX Orin 和 Jetson Xavier NX 模块的 PowerEstimator NVIDIA Isaac™ ROS 是硬件加速包的集合，使 ROS 开发人员可以更轻松地在包括 NVIDIA Jetson 在内的 NVIDIA 硬件上构建高性能解决方案。Isaac ROS DP3 发布支持 JetPack 5.1.1。</p>
Cloud Native	<p>Jetson 将 Cloud-Native 带到了边缘，并支持容器和容器编排等技术。NVIDIA JetPack 包括带有 Docker 集成的 NVIDIA Container Runtime，可在 Jetson 平台上启用 GPU 加速的容器化应用程序。NVIDIA 在 NVIDIA NGC 上为 Jetson 托管了多个容器镜像。有些适用于带有示例和文档的软件开发，有些适用于生产软件部署，仅包含运行时组件。在 Jetson 上的 Cloud-Native 页面上找到更多信息和所有容器镜像的列表。</p>
Security	<p>NVIDIA Jetson 模块包括各种安全功能，包括硬件信任根、安全启动、硬件加密加速、可信执行环境、磁盘和内存加密、物理攻击保护等。跳转到 Jetson Linux 开发人员指南的安全部分，了解安全功能。</p>

Sample Applications

JetPack 包括几个演示 JetPack 组件使用的示例。这些存储在参考文件系统中，可以在开发人员工具包上进行编译。

JetPack component	Sample locations on reference filesystem
TensorRT	/usr/src/tensorrt/samples/
cuDNN	/usr/src/cudnn_samples_/
CUDA	/usr/local/cuda-/samples/
Multimedia API	/usr/src/tegra_multimedia_api/
VisionWorks	/usr/share/visionworks/sources/samples/ /usr/share/visionworks-tracking/sources/samples/ /usr/share/visionworks-sfm/sources/samples/

OpenCV	/usr/share/OpenCV/samples/
VPI	/opt/nvidia/vpi/vpi-/samples

开发者工具

JetPack 包括以下开发工具。有些直接在 Jetson 系统上使用，有些则在连接到 Jetson 系统的 Linux 主机上运行。

用于应用程序开发和调试的工具(支持所有 Jetson 产品):

- NSight Eclipse 版用于开发 GPU 加速的应用程序。在 Linux 主机上运行。
- 用于应用程序调试的 CUDA-GDB。在 Jetson 系统或 Linux 主机上运行。
- CUDA-MEMCHECK 用于调试应用程序的内存错误。在 Jetson 系统上运行。

用于应用分析和优化的工具:

- NSight Systems 用于应用多核 CPU 分析。在 Linux 主机上运行。通过识别代码的缓慢部分，帮助你提高应用程序的性能。支持所有 Jetson 产品。
- NVIDIA® Nsight™ 计算内核剖析器。一个针对 CUDA 应用程序的互动式剖析工具。它通过一个用户界面和命令行工具提供详细的性能指标和 API 调试。
- NSight Graphics 用于图形应用程序的调试和分析。一个控制台级别的工具，用于调试和优化 OpenGL 和 OpenGL ES 程序。在 Linux 主机上运行。支持所有 Jetson 产品。

深圳智锐通科技有限公司
Shenzhen Zrt Co., Ltd.



智锐通公众号

&



智锐通抖音号

- 集团总部：深圳市宝安区碧桂园凤凰智谷A栋21楼
- 北京分公司：北京市昌平区科兴西路106号院2号楼5层
- 南京分公司：南京市江宁区万科都荟天地B2栋7楼
- 苏州分公司：苏州市虎丘区港龙城市广场4栋13楼
- 成都分公司：成都市武侯区世纪百合尚寓3栋10楼
- 400-838-6869