

NVIDIA 系列产品 快速刷机教程

主题	NVIDIA 系列产品 快速刷机教程
文档号	1.0
创建时间	2021-11-11
最后修改	2023-06-13
版本号	1.0
文件名	NVIDIA 系列产品 快速刷机教程.pdf
文件格式	Portable Document Format

阅前须知

声明

北京合众恒跃科技有限公司保留随时对其产品进行修正、改进和完善的权利，客户在下单前应获取相关信息的最新版本，并验证这些信息是正确的。本文档一切解释权归北京合众恒跃科技有限公司所有。

简介

北京合众恒跃科技有限公司（简称：HZHY）成立于 2011 年，是一家专业提供嵌入式应用解决方案的高新技术企业。本企业位于中关村高科技园区，专注于嵌入式产品研发、生产、销售等工作，并有着多年的嵌入式系统软、硬件技术积累，同时拥有一支技术精湛的嵌入式研发团队。该团队在嵌入式原理设计、信号完整性、电磁兼容性、电源冗余设计、系统热设计、完整的测试体系方面具有丰富的产品研发经验。

合众恒跃以卓越的服务品质、专业安全的技术服务实力，为不同的客户群体提供更好、更优质的嵌入式产品研发及生产服务。我们一直秉持“您身边的定制专家”理念，与诸多客户保持良好的合作关系，并致力于将低功耗、高性能、稳定可靠的嵌入式产品应用于工业控制、智能仪表、智能图像分析等诸多领域。

联系方式

北京合众恒跃科技有限公司

地址：北京市海淀区安宁庄后街南 1 号 A 区一层 1020 号

电话：010-62129511-8001

传真：010-62126811

官网：<http://www.hzhytech.com>

修改记录

版本号	日期	修改人	备注
1.0	2021-11-11	白新乐	初始版本
2.0	2023-02-20	白新乐	增加 AGX ORIN /AGX XAVIER 刷机部分
3.0	2023-06-13	白新乐	增加 ORIN NX/ORIN NANO 刷机部分

目录

阅前须知.....	2
声明	2
简介	2
联系方式	2
修改记录.....	3
目录.....	4
第 1 章 核心模块版本简介	6
1.1 版本说明	6
1.1.1 NX EMMC 版本.....	6
1.1.2 NX TF 卡版本.....	7
1.1.3 Nano EMMC 版本	7
1.1.4 Nano TF 卡 版本	7
1.1.5 TX2-NX EMMC 版本.....	8
1.1.6 AGX Xavier 版本	8
1.1.7 AGX Orin 32G 版本.....	9
1.1.8 Orin NX 版本	10
1.1.9 Orin NANO 版本	11
第 2 章 刷机步骤	12
2.1 刷机准备工作.....	12
2.2 进入刷机模式.....	12
2.2.1 智能盒 REC 按键对应关系.....	12
2.2.2 载板 REC 按键对应关系.....	13
2.2.3 智能盒进入刷机模式	13

2.2.4 裸板进入刷机模式	16
2.3 开始刷机	18
第 3 章 扩展系统空间	21
3.1 将系统扩展至 SSD 硬盘	21
3.1.1 将 NVMe SSD 硬盘分区并格式化	21
3.1.2 将文件系统扩展至 SSD 硬盘	22
3.2 将系统扩展至 TF 卡	23
第 4 章 安装官方组件	25

第1章 核心模块版本简介

1.1 版本说明

Nvidia Jetson 平台提供了多种模块，本公司基于不同的模块，提供不同的镜像包。不同模块的刷机方式也不同，请先对应自己模块的版本，在选择相应的刷机方式。

1.1.1 NX EMMC 版本

此版本对应官方“Jetson Xavier NX 8G（900-83668-0000-000），实物图如图 1.1-1：



图 1.1-1

此版本对应官方“Jetson Xavier NX 16G（900-83668-0030-000），实物图如图 1.1-2：



图 1.1-2

1.1.2 NX TF 卡版本

此版本对应官方“Jetson Xavier NX [developer kit version] (P3668-0000 module)”版本，实物图如

图 1.1-3:



图 1.1-3

1.1.3 Nano EMMC 版本

此版本对应官方“Jetson Nano (900-13448-0020-000)”版本，实物如图 1.1-4:



图 1.1-4

1.1.4 Nano TF 卡 版本

此版本对应官方“Jetson Nano [developer kit version] (P3448-0000 module)”版本，实物如图 1.1-5:



图 1.1-5

1.1.5 TX2-NX EMMC 版本

此版本对应官方“Jetson TX2-NX (900-13636-0010-000)”版本，实物如图 1.1-6：

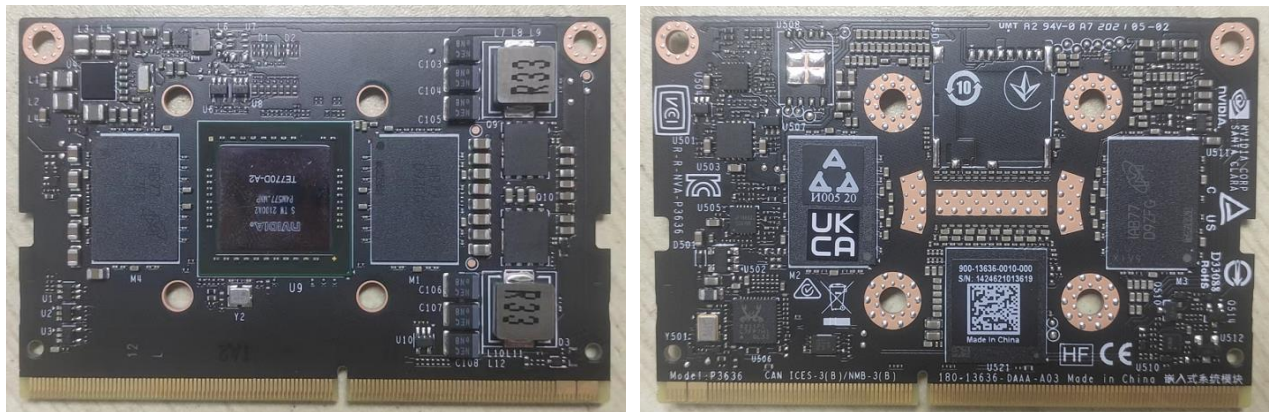


图 1.1-6

1.1.6 AGX Xavier 版本



图 1.1-7

1.1.7 AGX Orin 32G 版本



图 1.1-8

1.1.8 Orin NX 版本

此版本对应官方“Jetson Orin NX 16G（900-13767-0000-000）”版本，实物如图 1.1-9:



图 1.1-9

此版本对应官方“Jetson Orin NX 8G（900-13767-0010-000）”版本，实物如图 1.1-10：



图 1.1-10

1.1.9 Orin NANO 版本

此版本对应官方“Jetson Orin NANO 8G（900-13767-0030-000）”版本，实物如图 1.1-11：

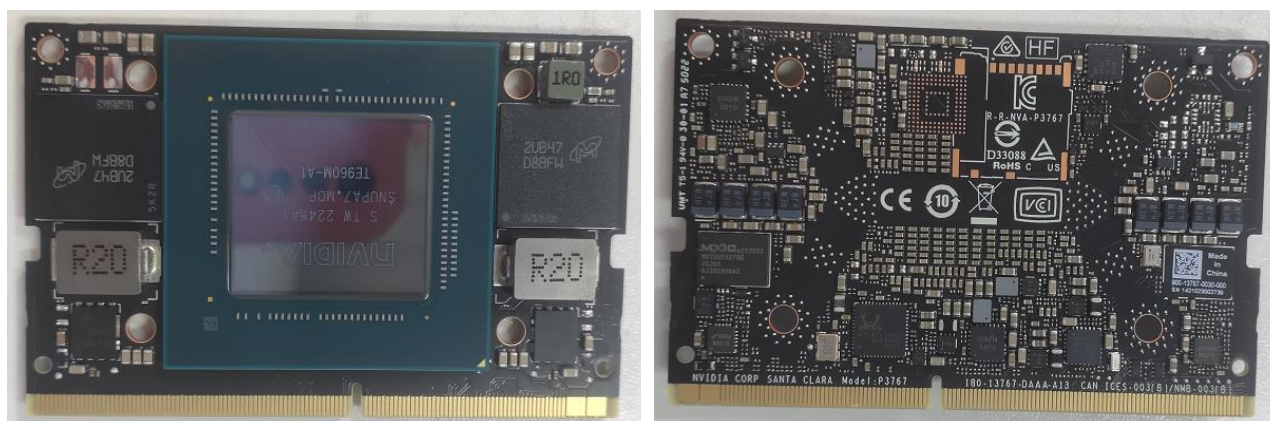


图 1.1-11

此版本对应官方“Jetson Orin NANO 4G（900-13767-0040-000）”版本，实物如图 1.1-12：

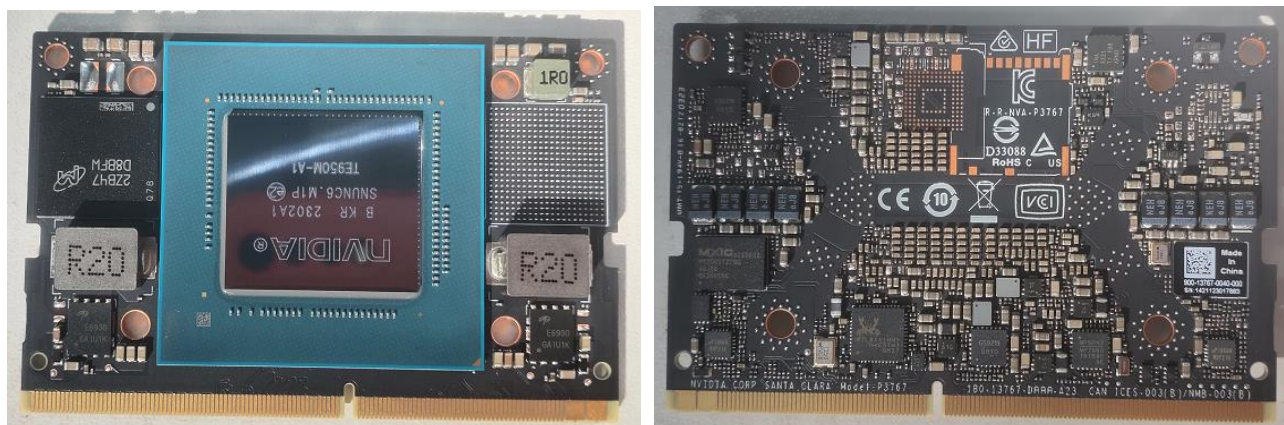


图 1.1-12

第2章 刷机步骤

2.1 刷机准备工作

1) 刷机前，请确认已有如下设备。

- HZHY NVIDIA 系列载板+核心模块壹套（或智能盒）。
- 准备 X86 Ubuntu 18.04 系统（或虚拟机）。 **注意：必须使用 PC 给智能盒刷机，不能使用智能盒给智能盒刷机。Ubuntu 版本必须是 18.04 的。**
- 通用 Type-C 数据线壹条（用作烧写和调试串口）。
- 通用 HDMI 显示器壹台（用作 HDMI 显示输出）。
- 通用 USB 键盘、USB 鼠标各壹个（用作键盘、鼠标输入）。
- 通用 12V~36V 电源适配器壹个（用作供电）。
- 刷机镜像包（根据淘宝提供的网盘链接，下载相应的刷机包）。

备注：刷机镜像在网盘资料“软件设计/烧写镜像”目录下，请根据所需的 Jetpack 版本自行下载。

2.2 进入刷机模式

2.2.1 智能盒 REC 按键对应关系

AI200、AI600、AI300、AI700、AI528 智能盒，均在智能盒表面印有 REC 按键标识，在此不在描述。

AI629 智能盒 REC 按键如图 2.2-1

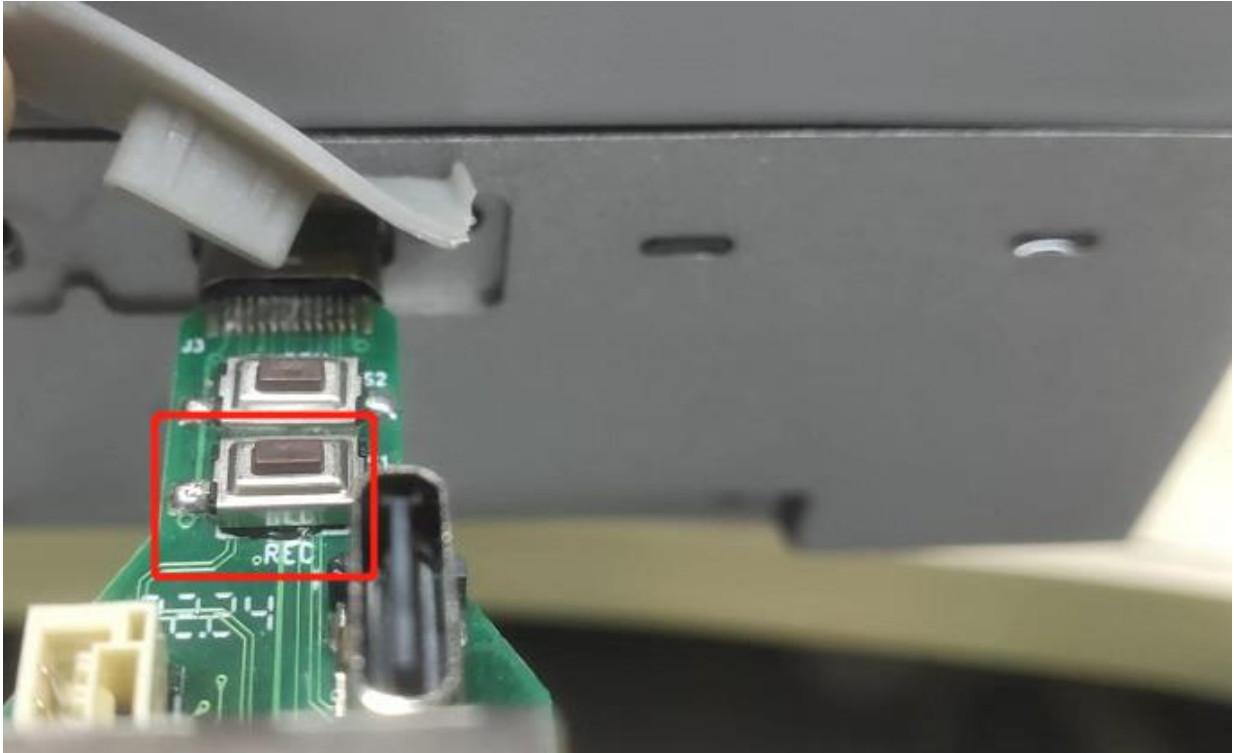


图 2.2-1

2.2.2 载板 REC 按键对应关系

裸板 REC 按键对应表如下：

载板名称	REC 按键
AI200 载板	J24
AI600 载板	J24
AI300 载板	J23
310UAV 载板	J23
311UAV 载板	SW1
701/702 载板	SW3

2.2.3 智能盒进入刷机模式

此处以 AI300 智能盒为例进行说明。

- 1) 用一根 Type-C 数据线连接智能盒 REC 接口与 PC 机，如图 2.2-2



图 2.2-2

- 2) 按住智能盒的 REC 按键，然后对智能盒上电（上电后等待 3-4 秒松开手），如图 2.2-3



图 2.2-3

3) 之后在 ubuntu 系统输入 lsusb 可查看到已识别到 Nvidia 设备，如图 2.2-4

```
hzy@hzy-IdeaCentre-GeekPro-14I08:/workspace/nvidia/nvidia_sdk/JetPack_4.4_Linux_JETSON_XAVIER_NX/Linux_for_Tegra$ lsusb
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 005: ID 067b:2731 Prolific Technology, Inc.
Bus 001 Device 003: ID 093a:2510 Pixart Imaging, Inc. Optical Mouse
Bus 001 Device 002: ID 0bda:c123 Realtek Semiconductor Corp.
Bus 001 Device 004: ID 0e8f:0022 GreenAsia Inc. multimedia keyboard controller
Bus 001 Device 010: ID 0955:7e19 NVidia Corp.
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

图 2.2-4

Bus <bbb> Device <ddd>: ID 0955: <nnnn> Nvidia Corp.

参数说明:

- <bbb> 是任何三位数字
- <ddd> 是任何三位数字
- <nnnn> 是一个四位数的数字，代表 Jetson 模块的类型：
 - 7023 for Jetson AGX Orin (P3701-0000 with 32GB)
 - 7023 for Jetson AGX Orin (P3701-0005 with 64GB)
 - 7223 for Jetson AGX Orin (P3701-0004 with 32GB)
 - 7323 for Jetson Orin NX (P3767-0000 with 16GB)
 - 7423 for Jetson Orin NX (P3767-0001 with 8GB)
 - 7523 for Jetson Orin Nano (P3767-0003 and P3767-0005 with 8GB)
 - 7623 for Jetson Orin Nano (P3767-0004 with 4GB)
 - 7019 for Jetson AGX Xavier (P2888-0001 with 16GB)

- 7019 for Jetson AGX Xavier (P2888-0004 with 32GB)
- 7019 for Jetson AGX Xavier (P2888-0005 with 64GB)
- 7019 for Jetson AGX Xavier Industrial (P2888-0008)
- 7e19 for Jetson Xavier NX (P3668)
- 7f21 for Supplied with Jetson Nano Developer Kit (P3448-0000)
- 7f21 for Production (P3448-0002)
- 7f21 for Supplied with Jetson Nano 2GB Developer Kit (P3448-0003)
- 7c18 for Jetson TX2 NX (P3636-0001)

备注：如果数值为其他表示系统已启动，未能成功进入恢复模式，请断电后按住 **REC** 按键，再次尝试。

2.2.4 裸板进入刷机模式

- 1) 用一根 Type-C 数据线连接裸板 J7 (REC) 接口与 PC 机，如图 2.2-5（此处以 HZHY-301_BV2.0 底板为例）。



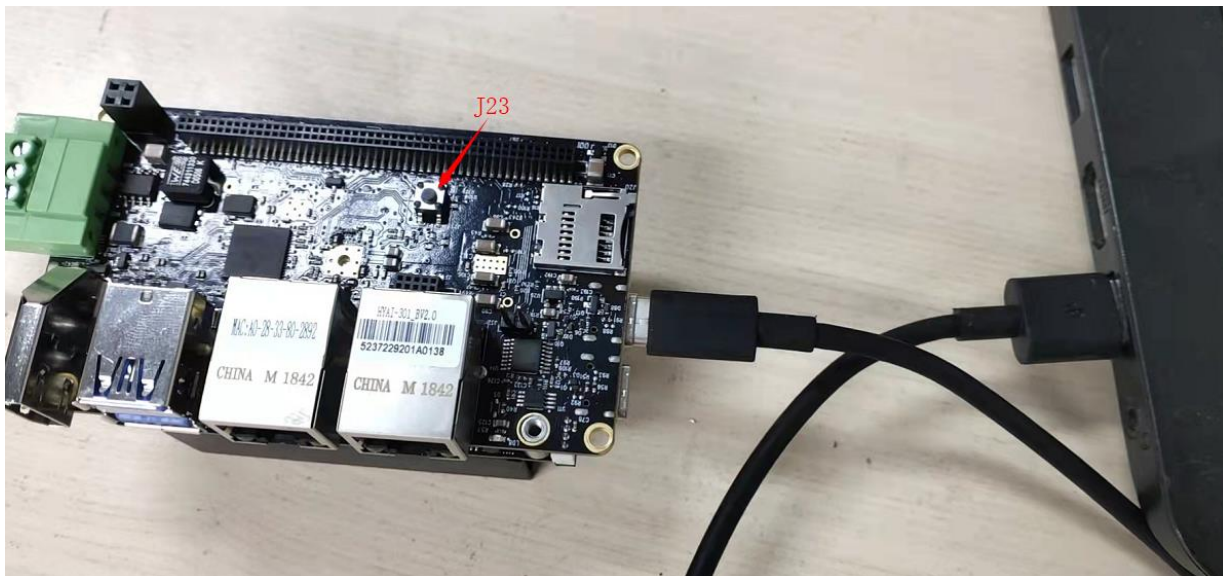


图 2.2-5

- 2) 按住裸板的 REC 按键，然后对裸板上电（上电后等待 3-4 秒松开手），之后在 ubuntu 系统输入 `lsusb` 可查看到已识别到 Nvidia 设备，如图 2.2-6

```
hzhzy@hzhzy-IdeaCentre-GeekPro-14I08:/workspace/nvidia/nvidia_sdk/JetPack_4.4_Linux_JETSON_XAVIER_NX/Linux_for_Tegra$ lsusb
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 005: ID 067b:2731 Prolific Technology, Inc.
Bus 001 Device 003: ID 093a:2510 Pixart Imaging, Inc. Optical Mouse
Bus 001 Device 002: ID 0bda:c123 Realtek Semiconductor Corp.
Bus 001 Device 004: ID 0e8f:0022 GreenAsia Inc. multimedia keyboard controller
Bus 001 Device 010: ID 0955:7e19 NVidia Corp.
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

图 2.2-6

Bus <bbb> Device <ddd>: ID 0955: <nnnn> Nvidia Corp.

参数说明：

- <bbb> 是任何三位数字
- <ddd> 是任何三位数字
- <nnnn> 是一个四位数的数字，代表 Jetson 模块的类型：
 - 7023 for Jetson AGX Orin (P3701-0000 with 32GB)
 - 7023 for Jetson AGX Orin (P3701-0005 with 64GB)
 - 7223 for Jetson AGX Orin (P3701-0004 with 32GB)
 - 7323 for Jetson Orin NX (P3767-0000 with 16GB)
 - 7423 for Jetson Orin NX (P3767-0001 with 8GB)
 - 7523 for Jetson Orin Nano (P3767-0003 and P3767-0005 with 8GB)
 - 7623 for Jetson Orin Nano (P3767-0004 with 4GB)
 - 7019 for Jetson AGX Xavier (P2888-0001 with 16GB)

- 7019 for Jetson AGX Xavier (P2888-0004 with 32GB)
- 7019 for Jetson AGX Xavier (P2888-0005 with 64GB)
- 7019 for Jetson AGX Xavier Industrial (P2888-0008)
- 7e19 for Jetson Xavier NX (P3668)
- 7f21 for Supplied with Jetson Nano Developer Kit (P3448-0000)
- 7f21 for Production (P3448-0002)
- 7f21 for Supplied with Jetson Nano 2GB Developer Kit (P3448-0003)
- 7c18 for Jetson TX2 NX (P3636-0001)

备注：如果数值为其他表示系统已启动，未能成功进入恢复模式，请断电后按住 **REC** 按键，再次尝试。

2.3 开始刷机

- 1) 将提供的镜像从网盘下载下来之后，拷贝到 ubuntu 系统。
- 2) 解压之后进入到“Linux_for_Tegra”目录下（在 jetpack 的一级目录下），如图 2.3-1

注意：

- a) 一定要用命令行解压，否则会刷写失败（命令：**sudo tar -xvf XXX.tar.gz** XXX 代表压缩包的名字）。
- b) 一定要拷贝到 PC 机上面之后在解压，否则会刷写失败。
- c) 一定要拷贝到 PC 机上面进行刷机，否则会刷写失败。
- d) 刷写过程不要碰到 Type-C 的线，不要碰到板子或盒子，否则会刷写失败。

```
hzhzy@hzhzy-IdeaCentre-GeekPro-1410B:/workspace/NVIDIA_Archived/NVIDIA_flash/flashAI_SC_release/flash-AI300serial-nx/JetPack_4.6.2_Linux_JETS
ON_XAVIER_NX_TARGETS-AI300serial-4g-SC/Linux_for_Tegra$ ls
apply_binaries.sh      jetson-xavier-slvs-ec.conf      p2822-0000+p2888-0008.conf
bootloader             kernel                          p2822-0000+p2888-0008-maxn.conf
build_l4t_bup.sh       l4t_generate_soc_bup.sh        p2822-0000+p2888-0008-noecc.conf
clara-agx-xavier-devkit.conf  l4t_sign_image.sh             p2972-0000.conf.common
e3900-0000+p2888-0004-b00.conf  LICENSE.sce_t194              p2972-0000-devkit-maxn.conf
flash.sh               nvautoflash.sh                 p2972-0000-devkit-slvs-ec.conf
jetson-agx-xavier-devkit.conf  nvmassflashgen.sh             p2972.conf
jetson-agx-xavier-ind-noecc.conf  nvmasffusegen.sh             p3449-0000+p3668-0000-qspi-sd.conf
jetson-agx-xavier-industrial.conf  nvskdmanager_flash.sh        p3449-0000+p3668-0001-qspi-emmc.conf
jetson-agx-xavier-industrial-mxn.conf  nv_tegra                     p3509-0000+p3636-0001.conf
jetson-tx2-4GB.conf      nv_tools                       p3509-0000+p3668-0000-qspi.conf
jetson-tx2-as-4GB.conf    odmfuse.func                   p3509-0000+p3668-0000-qspi-sd.conf
jetson-tx2.conf          odmfseread.sh                  p3509-0000+p3668-0001-qspi-emmc.conf
jetson-tx2-devkit-4gb.conf  odmfuse.sh                     p3636.conf.common
jetson-tx2-devkit.conf     p2597-0000+p3310-1000-as-p3489-0888.conf  p3668.conf.common
jetson-tx2-devkit-tx2i.conf  p2597-0000+p3310-1000.conf     pkc
jetson-tx2i.conf          p2597-0000+p3489-0000-ucm1.conf  README_Autoflash.txt
jetson-xavier.conf        p2597-0000+p3489-0000-ucm2.conf  README_Massflash.txt
jetson-xavier-maxn.conf    p2597-0000+p3489-0888.conf      rootfs
jetson-xavier-nx-devkit.conf  p2771-0000.conf.common         source
jetson-xavier-nx-devkit-emmc.conf  p2771-0000-dsi-hdmi-dp.conf    source_sync.sh
jetson-xavier-nx-devkit-qspi.conf  p2822-0000+p2888-0001.conf     TForSSD-kuozhan-AI300serial-jetpack4.6.2-4g-SC.img.raw
jetson-xavier-nx-devkit-tx2-nx.conf  p2822-0000+p2888-0004.conf     tools
hzhzy@hzhzy-IdeaCentre-GeekPro-1410B:/workspace/NVIDIA_Archived/NVIDIA_flash/flashAI_SC_release/flash-AI300serial-nx/JetPack_4.6.2_Linux_JETS
ON_XAVIER_NX_TARGETS-AI300serial-4g-SC/Linux_for_Tegra$
```

图 2.3-1

3) 之后输入下面的命令进行系统的刷写（根据实际情况选择板卡命令）：

NX EMMC 版本（jetpack4.x）：

```
Linux_for_Tegra/$ sudo ./flash.sh -r p3449-0000+p3668-0001-qspi-emmc mmcblk0p1
```

NX EMMC 版本（jetpack5.x）：

```
Linux_for_Tegra/$ sudo ./flash.sh -r jetson-xavier-nx-devkit-emmc internal
```

NX TF 卡版本：

```
Linux_for_Tegra/$ sudo ./flash.sh -r jetson-xavier-nx-devkit mmcblk0p1
```

Nano EMMC 版本：

```
Linux_for_Tegra/$ sudo ./flash.sh -r jetson-nano-emmc mmcblk0p1
```

Nano TF 卡版本：

使用官方工具刷写 TF 卡，具体参考镜像文件夹下面的《b01 镜像烧写备份》文档。

TX2-NX EMMC 版本：

```
Linux_for_Tegra/$ sudo ./flash.sh -r jetson-xavier-nx-devkit-tx2-nx mmcblk0p1
```

AGX Xavier 版本：

```
Linux_for_Tegra/$ sudo ./flash.sh -r jetson-xavier mmcblk0p1
```

AGX Orin 32G 版本：

```
Linux_for_Tegra/$ sudo ./flash.sh -r jetson-agx-orin-devkit internal
```

ORIN NX 16G/8G 版本：

```
Linux_for_Tegra/$ 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
```

ORIN NANO 8G/4G 版本：

```
Linux_for_Tegra/$ 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) EMMC 存储刷机成功后如图 2.3-2

```

[ 784.5190 ] Writing partition kernel with boot_sigheader.img.encrypt
[ 784.5510 ] [.....] 100%
[ 786.3961 ] Writing partition kernel_b with boot_sigheader.img.encrypt
[ 786.4062 ] [.....] 100%
[ 788.1846 ] Writing partition kernel-dtb with tegra194-p3668-all-p3509-0000_sigheader.dtb.encrypt
[ 788.1951 ] [.....] 100%
[ 788.1997 ] Writing partition kernel-dtb_b with tegra194-p3668-all-p3509-0000_sigheader.dtb.encrypt
[ 788.2242 ] [.....] 100%
[ 788.2585 ] [.....] 100%
[ 788.2661 ] tegradevflash_v2 --write BCT br_bct_BR.bct
[ 788.2682 ] Bootloader version 01.00.0000
[ 788.2711 ] Writing partition BCT with br_bct_BR.bct
[ 788.2722 ] [.....] 100%
[ 788.4233 ] [.....] 100%
[ 788.4534 ] tegradevflash_v2 --write MB1_BCT mb1_cold_boot_bct_MB1_sigheader.bct.encrypt
[ 788.4553 ] Bootloader version 01.00.0000
[ 788.4577 ] Writing partition MB1_BCT with mb1_cold_boot_bct_MB1_sigheader.bct.encrypt
[ 788.4585 ] [.....] 100%
[ 788.6731 ] [.....] 100%
[ 788.6757 ] tegradevflash_v2 --write MB1_BCT_b mb1_cold_boot_bct_MB1_sigheader.bct.encrypt
[ 788.6780 ] Bootloader version 01.00.0000
[ 788.6808 ] Writing partition MB1_BCT_b with mb1_cold_boot_bct_MB1_sigheader.bct.encrypt
[ 788.6829 ] [.....] 100%
[ 788.8849 ] [.....] 100%
[ 788.9857 ] tegradevflash_v2 --write MEM_BCT mem_coldboot_sigheader.bct.encrypt
[ 788.9876 ] Bootloader version 01.00.0000
[ 788.9898 ] Writing partition MEM_BCT with mem_coldboot_sigheader.bct.encrypt
[ 788.9105 ] [.....] 100%
[ 790.1918 ] [.....] 100%
[ 790.1947 ] tegradevflash_v2 --write MEM_BCT_b mem_coldboot_sigheader.bct.encrypt
[ 790.1966 ] Bootloader version 01.00.0000
[ 790.1993 ] Writing partition MEM_BCT_b with mem_coldboot_sigheader.bct.encrypt
[ 790.2011 ] [.....] 100%
[ 791.4598 ] [.....] 100%
[ 791.4600 ] Flashing completed

[ 791.4601 ] Coldbooting the device
[ 791.4866 ] tegrarcm_v2 --ismb2
[ 791.5074 ] [.....] 100%
[ 791.5103 ] tegradevflash_v2 --reboot coldboot
[ 791.5128 ] Bootloader version 01.00.0000
[ 791.5322 ] [.....] 100%
*** The target t186ref has been flashed successfully. ***
Reset the board to boot from internal eMMC.

```

图 2.3-2

5) SSD 存储刷机成功后如图 2.2-3

```

tar: Read checkpoint 450000
tar: Read checkpoint 460000
tar: Read checkpoint 470000
tar: Read checkpoint 480000
tar: Read checkpoint 490000
tar: Read checkpoint 500000
tar: Read checkpoint 510000
tar: Read checkpoint 520000
tar: Read checkpoint 530000
tar: Read checkpoint 540000
tar: Read checkpoint 550000
tar: Read checkpoint 560000
tar: Read checkpoint 570000
tar: Read checkpoint 580000
tar: Read checkpoint 590000
tar: Read checkpoint 600000
tar: Read checkpoint 610000
tar: Read checkpoint 620000
tar: Read checkpoint 630000
tar: Read checkpoint 640000
tar: Read checkpoint 650000
tar: Read checkpoint 660000
tar: Read checkpoint 670000
tar: Read checkpoint 680000
tar: Read checkpoint 690000
tar: Read checkpoint 700000
tar: Read checkpoint 710000
tar: Read checkpoint 720000
tar: Read checkpoint 730000
tar: Read checkpoint 740000
tar: Read checkpoint 750000
tar: Read checkpoint 760000
writing items=16, 9:0:secondary_gpt, 61203267072, 16896, gpt_secondary_9_0.bin, 16896, fixed-<reserved>-0, 555a95db6a2d42b976b0974275f16ac959154b00
[ 580]: l4t_flash_from_kernel: Successfully flash the external device
[ 580]: l4t_flash_from_kernel: Flashing success
[ 580]: l4t_flash_from_kernel: The device size indicated in the partition layout xml is smaller than the actual size. This utility will try to fix the GPT.
Flash is successful
Reboot device
Cleaning up...
Log is saved to Linux for Tegra/initrdlog/flash_1-9.3_0_20230613-145024.log
hzhy@hzy-IdeaCentre-GeekPro-1410B:/workspace/NVIDIA_Sources/Jetson-Orin-Nano/flash-AI300serial-Orin-Nano-Jetpack5.1.1-SC/Linux_for_Tegra$

```

图 2.3-3

第3章 扩展系统空间

由于 Xavier NX 的 EMMC 只有 16G 大小，当我们需要部署一些软件或者安装官方组件的时候，空间会不够，这时候可以选择进行系统扩展，我们的扩展方式通常有两种，一种是 SSD 扩展，一种是 TF 卡扩展，选择不同的扩展，系统运行速度有很大区别。客户根据需求选择相应的扩展方式。

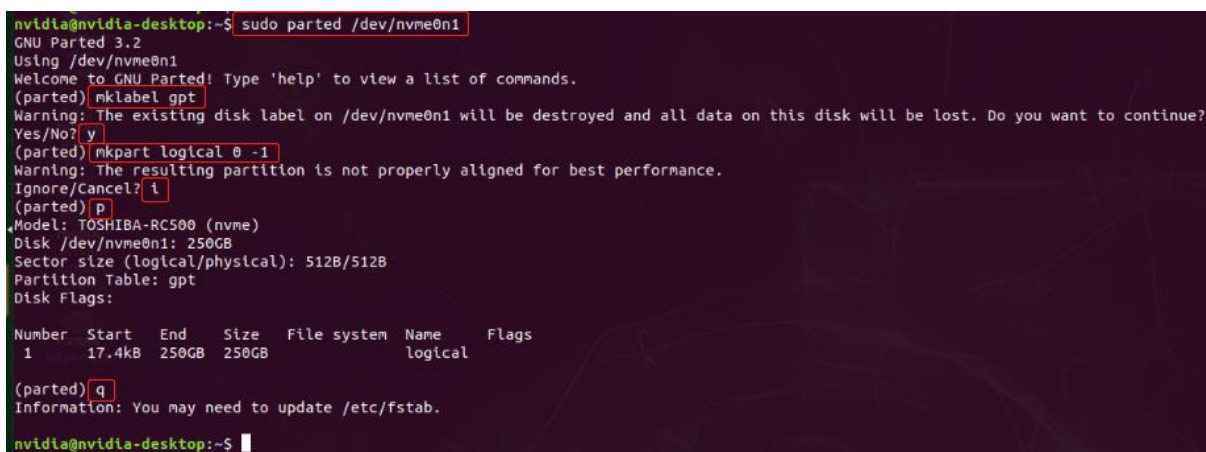
Orin NX/NANO 系列直接将 SSD 作为存储盘，所以不需要进行扩展系统空间的步骤。

3.1 将系统扩展至 SSD 硬盘

3.1.1 将 NVMe SSD 硬盘分区并格式化

- 1) 将 NVMe M.2 M SSD 硬盘分区为 GPT 格式，输入下面的命令（一般新购买的硬盘需要进行分区的操作，如果已经对硬盘进行过分区，则直接执行下面的步骤），如图 3.1-1

```
sudo parted /dev/nvme0n1    #进入 parted
mklabel gpt                #将磁盘设置为 gpt 格式，
mkpart logical 0 -1        #将磁盘所有的容量设置为 GPT 格式
p                            #查看设置结果
```



```
nvidia@nvidia-desktop:~$ sudo parted /dev/nvme0n1
GNU Parted 3.2
Using /dev/nvme0n1
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) mklabel gpt
Warning: The existing disk label on /dev/nvme0n1 will be destroyed and all data on this disk will be lost. Do you want to continue?
Yes/No? y
(parted) mkpart logical 0 -1
Warning: The resulting partition is not properly aligned for best performance.
Ignore/Cancel? i
(parted) p
Model: TOSHIBA-RC500 (nvme)
Disk /dev/nvme0n1: 250GB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number  Start   End     Size    File system  Name      Flags
 1      17.4kB  250GB   250GB                   logical

(parted) q
Information: You may need to update /etc/fstab.
nvidia@nvidia-desktop:~$
```

图 3.1-1

- 2) 输入下面的命令格式化分区，如图 3.1-2

```
sudo mkfs.ext4 /dev/nvme0n1p1
```

```
nvidia@nvidia-desktop:~$ sudo mkfs.ext4 /dev/nvme0n1p1
mke2fs 1.44.1 (24-Mar-2018)
Found a gpt partition table in /dev/nvme0n1p1
Proceed anyway? (y,N) y
Creating filesystem with 61049397 4k blocks and 15269888 inodes
Filesystem UUID: 4ec3adb-25ea-4bc4-a40f-3775dd8488c9
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
    4096000, 7962624, 11239424, 20480000, 23887872

Allocating group tables: done
Writing inode tables: done
Creating journal (262144 blocks): done
Writing superblocks and filesystem accounting information: done

nvidia@nvidia-desktop:~$
```

图 3.1-2

3.1.2 将文件系统扩展至 SSD 硬盘

- 1) 输入下面的命令，进入到/hzhy/rootOnNVMe 目录下，如图 3.1-3。

```
cd /hzhy/rootOnNVMe
```

```
nvidia@nvidia-desktop:~$ cd /hzhy/rootOnNVMe/
nvidia@nvidia-desktop:/hzhy/rootOnNVMe$ ls
copy-rootfs-ssd.sh  data  LICENSE  README.md  setup-service.sh
```

图 3.1-3

- 2) 将 eMMC 的 rootfs 复制到 SSD 硬盘，执行下面的命令如图 3.1-4，复制的过程比较长，请耐心等待。

```
sudo ./copy-rootfs-ssd.sh
```

```
nvidia@nvidia-desktop:/hzhy/rootOnNVMe$ sudo ./copy-rootfs-ssd.sh
138,283,092 33% 42.17MB/s 0:00:03 (xfr#2936, lr-chk=1297/5833)
```

图 3.1-4

- 3) 复制完成后，设置启动方式，执行下面的命令，执行命令后在弹出提示后输入系统密码“nvidia”，设置成功后如图 3.1-5，之后输入 sync 同步上述操作

```
sudo ./setup-service.sh
```

```
nvidia@nvidia-desktop:/hzhy/rootOnNVMe$ sudo ./copy-rootfs-ssd.sh
12,951,718,150 96% 62.09MB/s 0:03:18 (xfr#130041, to-chk=0/183427)
nvidia@nvidia-desktop:/hzhy/rootOnNVMe$ sudo ./setup-service.sh
Service to set the rootfs to the SSD installed.
Make sure that you have copied the rootfs to SSD.
Reboot for changes to take effect.
nvidia@nvidia-desktop:/hzhy/rootOnNVMe$ sync
nvidia@nvidia-desktop:/hzhy/rootOnNVMe$
```

图 3.1-5

- 4) 重新启动板卡，之后输入 `df -h` 命令查看系统已经从 SSD 启动，如图 3.1-6

```
nvidia@nvidia-desktop:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
none            3.5G   0    3.5G   0% /dev
tmpfs           3.8G  4.0K  3.8G   1% /dev/shm
tmpfs           3.8G  21M  3.8G   1% /run
tmpfs           5.0M  4.0K  5.0M   1% /run/lock
tmpfs           3.8G   0    3.8G   0% /sys/fs/cgroup
/dev/nvme0n1p1  229G  13G  205G   6% /
tmpfs           777M  12K  777M   1% /run/user/120
/dev/sda1       15G   1.9G   14G  13% /media/2A9E-2030
tmpfs           777M  100K  777M   1% /run/user/1000
/dev/mmcblk0p1  14G   13G   994M  93% /media/nvidia/6dfb5d79-99d6-4106-b949-fec1ac7ae6c8
nvidia@nvidia-desktop:~$
```

图 3.1-6

3.2 将系统扩展至 TF 卡

- 1) 输入下面的命令格式化 TF 卡分区（如果 TF 卡已经被挂载了，则需要先卸载 TF 卡），如图 3.2-1

```
df -h
```

```
sudo umount /dev/mmcblk1p1
```

```
sudo mkfs.ext4 /dev/mmcblk1p1
```

```
nvidia@nvidia-desktop:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/mmcblk0p1  14G   13G  528M  97% /
none            3.5G   0    3.5G   0% /dev
tmpfs           3.8G  4.0K  3.8G   1% /dev/shm
tmpfs           3.8G  21M  3.8G   1% /run
tmpfs           5.0M  4.0K  5.0M   1% /run/lock
tmpfs           3.8G   0    3.8G   0% /sys/fs/cgroup
tmpfs           777M  12K  777M   1% /run/user/120
tmpfs           777M  124K  777M   1% /run/user/1000
/dev/mmcblk1p1  59G   13G   43G  23% /media/nvidia/307df0a8-4908-455a-a840-47c0a025e4c8
nvidia@nvidia-desktop:~$ sudo umount /dev/mmcblk1p1
[sudo] password for nvidia:
nvidia@nvidia-desktop:~$ sudo mkfs.ext4 /dev/mmcblk1p1
mke2fs 1.44.1 (24-Mar-2018)
/dev/mmcblk1p1 contains a ext4 file system
    last mounted on / on Tue Nov  3 11:13:12 2020
Proceed anyway? (y,N) y
Discarding device blocks: done
Creating filesystem with 15587840 4k blocks and 3899392 inodes
Filesystem UUID: 63dbbad5-f3c8-45dd-84b1-8688c8feded6
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
    4096000, 7962624, 11239424

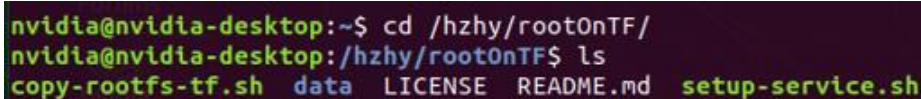
Allocating group tables: done
Writing inode tables: done
Creating journal (65536 blocks): done
Writing superblocks and filesystem accounting information: done

nvidia@nvidia-desktop:~$
```

图 3.2-1

- 2) 输入下面的命令，进入到/hzhy/rootOnTF 目录下，如图 3.2-2。

```
cd /hzhy/rootOnTF
```

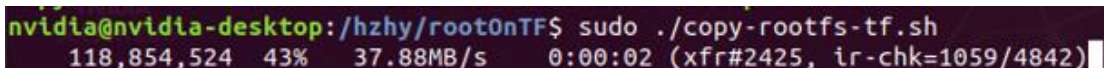


```
nvidia@nvidia-desktop:~$ cd /hzhy/rootOnTF/  
nvidia@nvidia-desktop:/hzhy/rootOnTF$ ls  
copy-rootfs-tf.sh  data  LICENSE  README.md  setup-service.sh
```

图 3.2-2

- 3) 将 eMMC 的 rootfs 复制到 TF 卡，执行下面的命令如图 3.2-3，复制的过程比较长，请耐心等待。

```
sudo ./copy-rootfs-tf.sh
```

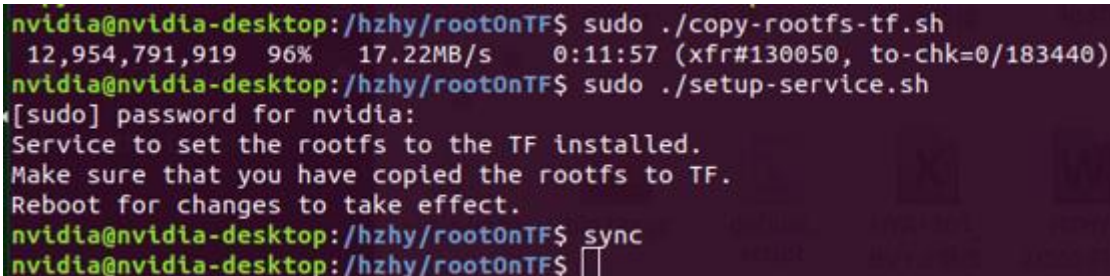


```
nvidia@nvidia-desktop:/hzhy/rootOnTF$ sudo ./copy-rootfs-tf.sh  
118,854,524 43% 37.88MB/s 0:00:02 (xfr#2425, ir-chk=1059/4842)
```

图 3.2-3

- 4) 复制完成后，设置启动方式，执行下面的命令，在弹出提示后输入系统密码“nvidia”，设置成功如图 3.2-4，之后输入 sync 同步上述操作

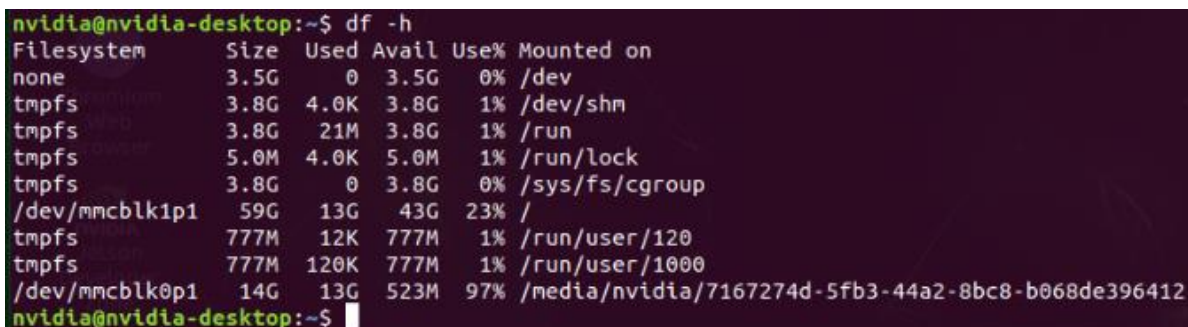
```
sudo ./setup-service.sh
```



```
nvidia@nvidia-desktop:/hzhy/rootOnTF$ sudo ./copy-rootfs-tf.sh  
12,954,791,919 96% 17.22MB/s 0:11:57 (xfr#130050, to-chk=0/183440)  
nvidia@nvidia-desktop:/hzhy/rootOnTF$ sudo ./setup-service.sh  
[sudo] password for nvidia:  
Service to set the rootfs to the TF installed.  
Make sure that you have copied the rootfs to TF.  
Reboot for changes to take effect.  
nvidia@nvidia-desktop:/hzhy/rootOnTF$ sync  
nvidia@nvidia-desktop:/hzhy/rootOnTF$
```

图 3.2-4

- 5) 重新启动板卡，之后输入 df -h 命令查看系统已经从 TF 卡启动，如图 3.2-5



```
nvidia@nvidia-desktop:~$ df -h  
Filesystem      Size  Used Avail Use% Mounted on  
none            3.5G   0    3.5G   0% /dev  
tmpfs            3.8G  4.0K   3.8G   1% /dev/shm  
tmpfs            3.8G  21M   3.8G   1% /run  
tmpfs            5.0M  4.0K   5.0M   1% /run/lock  
tmpfs            3.8G   0    3.8G   0% /sys/fs/cgroup  
/dev/mmcblk1p1   59G   13G   43G   23% /  
tmpfs            777M  12K   777M   1% /run/user/120  
tmpfs            777M  120K   777M   1% /run/user/1000  
/dev/mmcblk0p1   14G   13G   523M  97% /media/nvidia/7167274d-5fb3-44a2-8bc8-b068de396412  
nvidia@nvidia-desktop:~$
```

图 3.2-5

第4章 安装官方组件

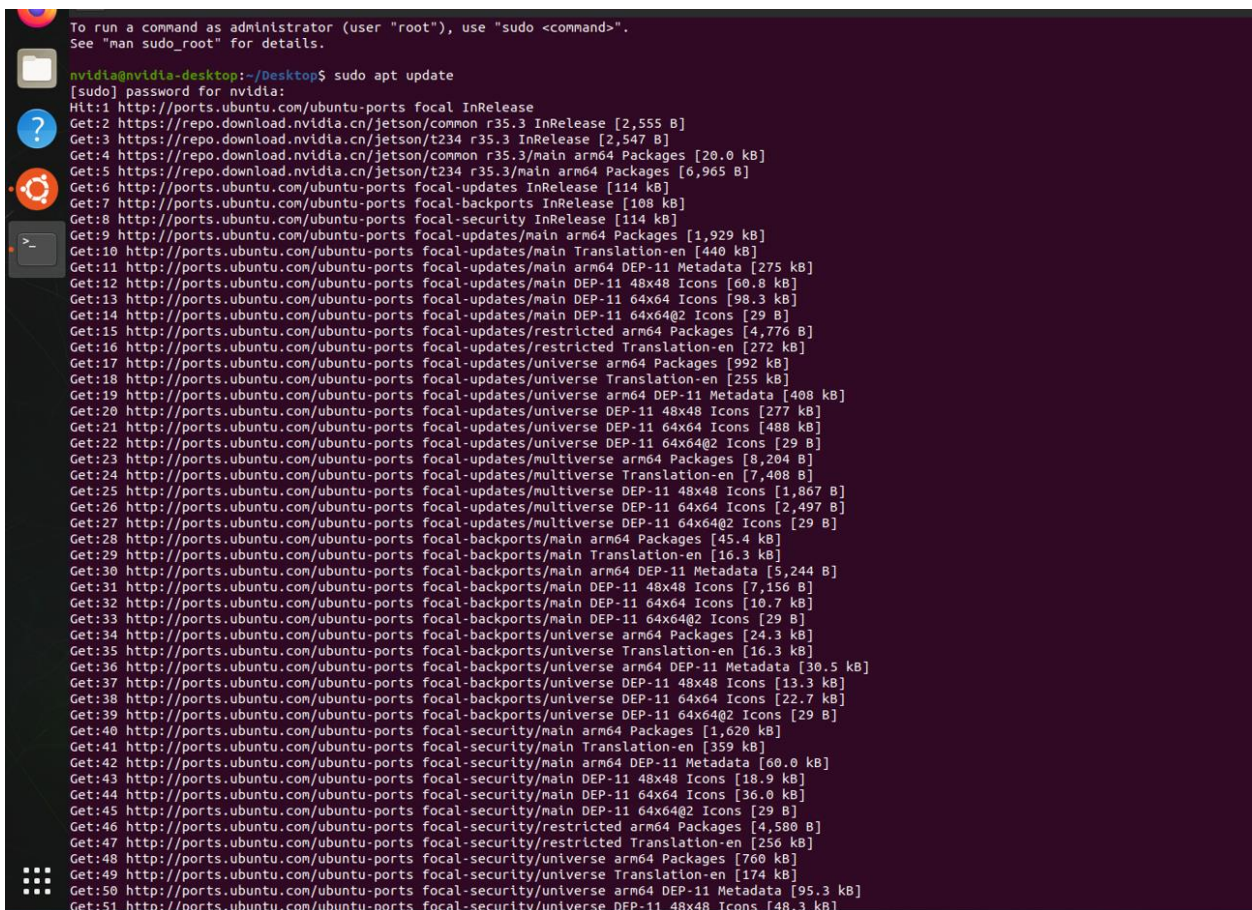
正确刷机系统和扩展系统后，可以从互联网上安装与 L4T 版本相对应的 Jetpack 组件。

注意：Xavier NX 如果刷的是 Jetpack 4.6 或者 Jetpack4.4 版本，EMMC 已经集成组件，无需进行安装组件的操作。

1) 在系统界面打开终端，输入下面的命令，

```
sudo apt update
```

```
sudo apt install nvidia-jetpack
```



```
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

nvidia@nvidia-desktop: ~/Desktop$ sudo apt update
[sudo] password for nvidia:
Hit:1 http://ports.ubuntu.com/ubuntu-ports focal InRelease
Get:2 https://repo.download.nvidia.cn/jetson/common r35.3 InRelease [2,555 B]
Get:3 https://repo.download.nvidia.cn/jetson/t234 r35.3 InRelease [2,547 B]
Get:4 https://repo.download.nvidia.cn/jetson/common r35.3/main arm64 Packages [20.0 kB]
Get:5 https://repo.download.nvidia.cn/jetson/t234 r35.3/main arm64 Packages [6,965 B]
Get:6 http://ports.ubuntu.com/ubuntu-ports focal-updates InRelease [114 kB]
Get:7 http://ports.ubuntu.com/ubuntu-ports focal-backports InRelease [108 kB]
Get:8 http://ports.ubuntu.com/ubuntu-ports focal-security InRelease [114 kB]
Get:9 http://ports.ubuntu.com/ubuntu-ports focal-updates/main arm64 Packages [1,929 kB]
Get:10 http://ports.ubuntu.com/ubuntu-ports focal-updates/main Translation-en [440 kB]
Get:11 http://ports.ubuntu.com/ubuntu-ports focal-updates/main DEP-11 Metadata [275 kB]
Get:12 http://ports.ubuntu.com/ubuntu-ports focal-updates/main DEP-11 48x48 Icons [60.8 kB]
Get:13 http://ports.ubuntu.com/ubuntu-ports focal-updates/main DEP-11 64x64 Icons [98.3 kB]
Get:14 http://ports.ubuntu.com/ubuntu-ports focal-updates/main DEP-11 64x64@2 Icons [29 B]
Get:15 http://ports.ubuntu.com/ubuntu-ports focal-updates/restricted arm64 Packages [4,776 B]
Get:16 http://ports.ubuntu.com/ubuntu-ports focal-updates/restricted Translation-en [272 kB]
Get:17 http://ports.ubuntu.com/ubuntu-ports focal-updates/universe arm64 Packages [992 kB]
Get:18 http://ports.ubuntu.com/ubuntu-ports focal-updates/universe Translation-en [255 kB]
Get:19 http://ports.ubuntu.com/ubuntu-ports focal-updates/universe arm64 DEP-11 Metadata [408 kB]
Get:20 http://ports.ubuntu.com/ubuntu-ports focal-updates/universe DEP-11 48x48 Icons [277 kB]
Get:21 http://ports.ubuntu.com/ubuntu-ports focal-updates/universe DEP-11 64x64 Icons [488 kB]
Get:22 http://ports.ubuntu.com/ubuntu-ports focal-updates/universe DEP-11 64x64@2 Icons [29 B]
Get:23 http://ports.ubuntu.com/ubuntu-ports focal-updates/multiverse arm64 Packages [8,204 B]
Get:24 http://ports.ubuntu.com/ubuntu-ports focal-updates/multiverse Translation-en [7,408 B]
Get:25 http://ports.ubuntu.com/ubuntu-ports focal-updates/multiverse DEP-11 48x48 Icons [1,867 B]
Get:26 http://ports.ubuntu.com/ubuntu-ports focal-updates/multiverse DEP-11 64x64 Icons [2,497 B]
Get:27 http://ports.ubuntu.com/ubuntu-ports focal-updates/multiverse DEP-11 64x64@2 Icons [29 B]
Get:28 http://ports.ubuntu.com/ubuntu-ports focal-backports/main arm64 Packages [45.4 kB]
Get:29 http://ports.ubuntu.com/ubuntu-ports focal-backports/main Translation-en [16.3 kB]
Get:30 http://ports.ubuntu.com/ubuntu-ports focal-backports/main arm64 DEP-11 Metadata [5,244 B]
Get:31 http://ports.ubuntu.com/ubuntu-ports focal-backports/main DEP-11 48x48 Icons [7,156 B]
Get:32 http://ports.ubuntu.com/ubuntu-ports focal-backports/main DEP-11 64x64 Icons [10.7 kB]
Get:33 http://ports.ubuntu.com/ubuntu-ports focal-backports/main DEP-11 64x64@2 Icons [29 B]
Get:34 http://ports.ubuntu.com/ubuntu-ports focal-backports/universe arm64 Packages [24.3 kB]
Get:35 http://ports.ubuntu.com/ubuntu-ports focal-backports/universe Translation-en [16.3 kB]
Get:36 http://ports.ubuntu.com/ubuntu-ports focal-backports/universe arm64 DEP-11 Metadata [30.5 kB]
Get:37 http://ports.ubuntu.com/ubuntu-ports focal-backports/universe DEP-11 48x48 Icons [13.3 kB]
Get:38 http://ports.ubuntu.com/ubuntu-ports focal-backports/universe DEP-11 64x64 Icons [22.7 kB]
Get:39 http://ports.ubuntu.com/ubuntu-ports focal-backports/universe DEP-11 64x64@2 Icons [29 B]
Get:40 http://ports.ubuntu.com/ubuntu-ports focal-security/main arm64 Packages [1,620 kB]
Get:41 http://ports.ubuntu.com/ubuntu-ports focal-security/main Translation-en [359 kB]
Get:42 http://ports.ubuntu.com/ubuntu-ports focal-security/main arm64 DEP-11 Metadata [60.0 kB]
Get:43 http://ports.ubuntu.com/ubuntu-ports focal-security/main DEP-11 48x48 Icons [18.9 kB]
Get:44 http://ports.ubuntu.com/ubuntu-ports focal-security/main DEP-11 64x64 Icons [36.0 kB]
Get:45 http://ports.ubuntu.com/ubuntu-ports focal-security/main DEP-11 64x64@2 Icons [29 B]
Get:46 http://ports.ubuntu.com/ubuntu-ports focal-security/restricted arm64 Packages [4,580 B]
Get:47 http://ports.ubuntu.com/ubuntu-ports focal-security/restricted Translation-en [256 kB]
Get:48 http://ports.ubuntu.com/ubuntu-ports focal-security/universe arm64 Packages [760 kB]
Get:49 http://ports.ubuntu.com/ubuntu-ports focal-security/universe Translation-en [174 kB]
Get:50 http://ports.ubuntu.com/ubuntu-ports focal-security/universe arm64 DEP-11 Metadata [95.3 kB]
Get:51 http://ports.ubuntu.com/ubuntu-ports focal-security/universe DEP-11 48x48 Icons [48.3 kB]
```

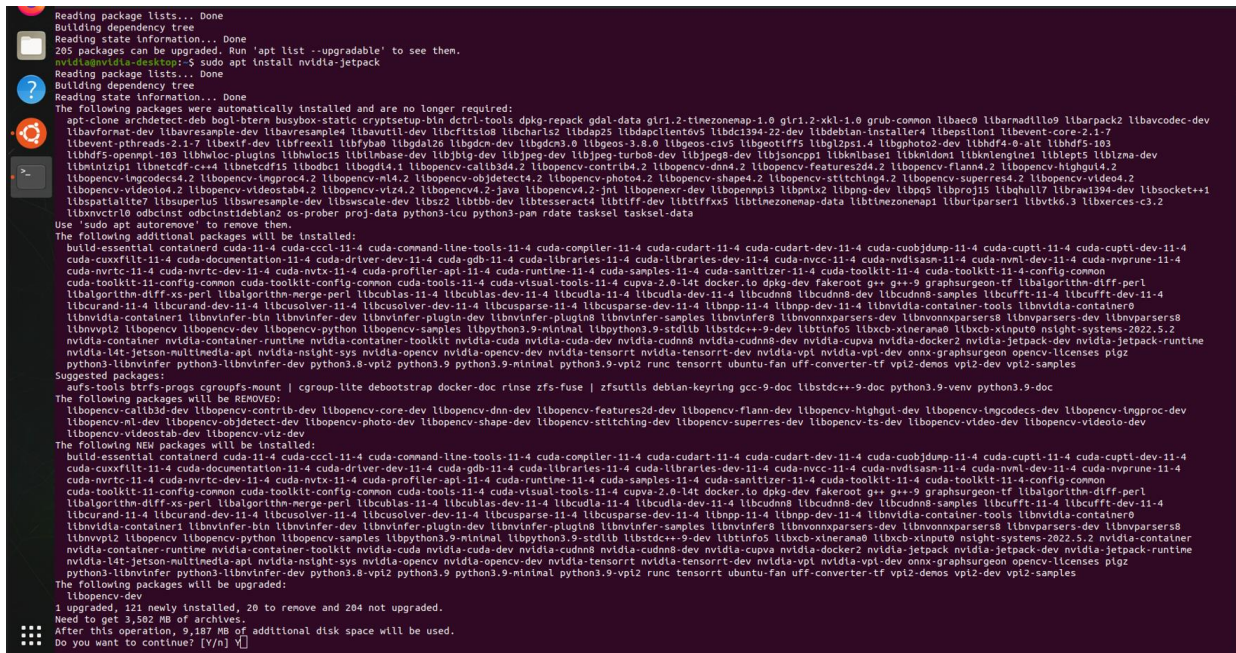
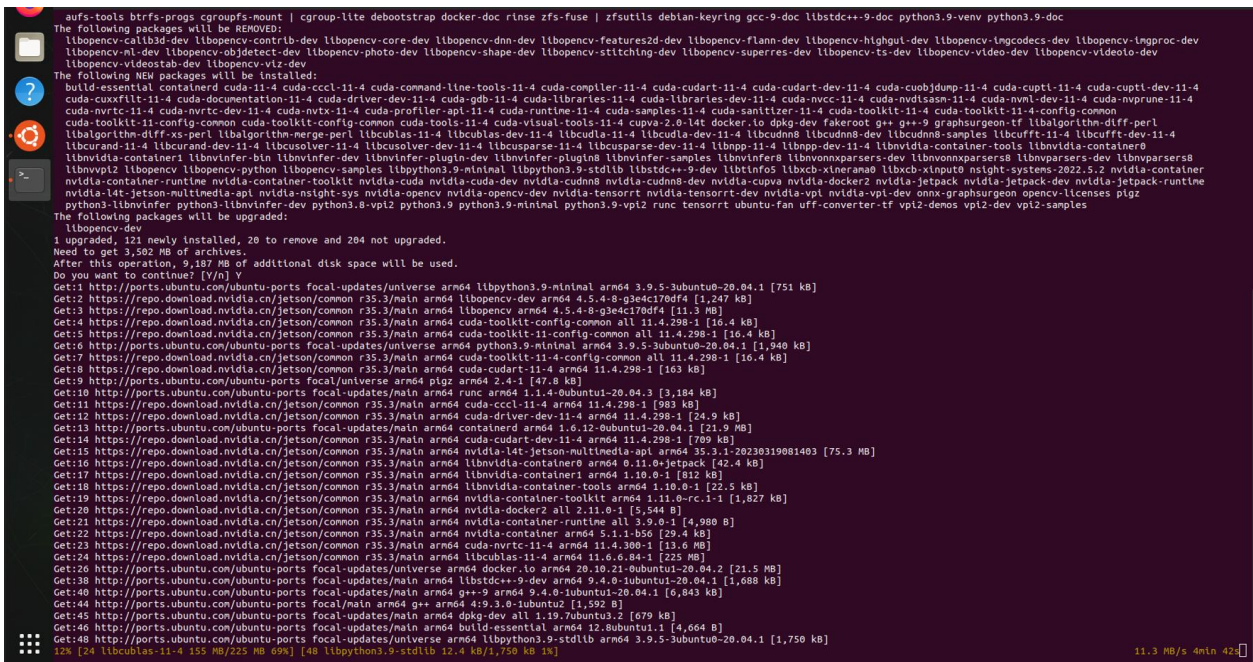



图 3.2-1

2) 正在安装过程中，截图如下



```

Removing libopencv-superres-dev:arm64 (4.2.0+dfsg-5) ...
dpkg: libopencv-videotest-dev:arm64: dependency problems, but removing anyway as you requested:
libopencv-dev depends on libopencv-videotest-dev (= 4.2.0+dfsg-5); however:
Package libopencv-videotest-dev:arm64 is to be removed.
libopencv-contrib-dev:arm64 depends on libopencv-videotest-dev (= 4.2.0+dfsg-5); however:
Package libopencv-videotest-dev:arm64 is to be removed.

Removing libopencv-videotest-dev:arm64 (4.2.0+dfsg-5) ...
dpkg: libopencv-ngcodecs-dev:arm64: dependency problems, but removing anyway as you requested:
libopencv-dev depends on libopencv-ngcodecs-dev (= 4.2.0+dfsg-5); however:
Package libopencv-ngcodecs-dev:arm64 is to be removed.
libopencv-contrib-dev:arm64 depends on libopencv-ngcodecs-dev (= 4.2.0+dfsg-5); however:
Package libopencv-ngcodecs-dev:arm64 is to be removed.

Removing libopencv-ngcodecs-dev:arm64 (4.2.0+dfsg-5) ...
dpkg: libopencv-shape-dev:arm64: dependency problems, but removing anyway as you requested:
libopencv-dev depends on libopencv-shape-dev (= 4.2.0+dfsg-5); however:
Package libopencv-shape-dev:arm64 is to be removed.
libopencv-contrib-dev:arm64 depends on libopencv-shape-dev (= 4.2.0+dfsg-5); however:
Package libopencv-shape-dev:arm64 is to be removed.

Removing libopencv-shape-dev:arm64 (4.2.0+dfsg-5) ...
(Reading database ... 145808 files and directories currently installed.)
Preparing to unpack .../libopencv-dev_4.5.4-8-g3e4c170df4_arm64.deb ...
Unpacking libopencv-dev (4.5.4-8-g3e4c170df4) over (4.2.0+dfsg-5) ...
(Reading database ... 146072 files and directories currently installed.)
Removing libopencv-contrib-dev:arm64 (4.2.0+dfsg-5) ...
Removing libopencv-viz-dev:arm64 (4.2.0+dfsg-5) ...
Selecting previously unselected package libopencv.
(Reading database ... 146002 files and directories currently installed.)
Preparing to unpack .../000-libopencv_4.5.4-8-g3e4c170df4_arm64.deb ...
Unpacking libopencv (4.5.4-8-g3e4c170df4) ...
Selecting previously unselected package cuda-toolkit-config-common.
Preparing to unpack .../001-cuda-toolkit-config-common_11.4.298-1_all.deb ...
Unpacking cuda-toolkit-config-common (11.4.298-1) ...
Selecting previously unselected package cuda-toolkit-11-config-common.
Preparing to unpack .../002-cuda-toolkit-11-config-common_11.4.298-1_all.deb ...
Unpacking cuda-toolkit-11-config-common (11.4.298-1) ...
Selecting previously unselected package cuda-toolkit-11-4-config-common.
Preparing to unpack .../003-cuda-toolkit-11-4-config-common_11.4.298-1_all.deb ...
Unpacking cuda-toolkit-11-4-config-common (11.4.298-1) ...
Selecting previously unselected package cuda-cudart-11-4.
Preparing to unpack .../004-cuda-cudart-11-4_11.4.298-1_arm64.deb ...
Unpacking cuda-cudart-11-4 (11.4.298-1) ...
Selecting previously unselected package cuda-cccl-11-4.
Preparing to unpack .../005-cuda-cccl-11-4_11.4.298-1_arm64.deb ...
Unpacking cuda-cccl-11-4 (11.4.298-1) ...
Selecting previously unselected package cuda-driver-dev-11-4.
Preparing to unpack .../006-cuda-driver-dev-11-4_11.4.298-1_arm64.deb ...
Unpacking cuda-driver-dev-11-4 (11.4.298-1) ...
Selecting previously unselected package cuda-cudart-dev-11-4.
Preparing to unpack .../007-cuda-cudart-dev-11-4_11.4.298-1_arm64.deb ...
Unpacking cuda-cudart-dev-11-4 (11.4.298-1) ...
Progress: [ 10%] [#####]

```

图 3.2-2

3) 安装时间比较长，一般由网速决定，安装成功如下图

```

Setting up libcuFFT-dev-11-4 (10.6.0.202-1) ...
Setting up nvidia-ocpvc-dev (5.1.1-b50) ...
Setting up nvidia-container-runtime (3.9.0-1) ...
Setting up libcusparse-dev-11-4 (11.6.0.299-1) ...
Setting up nvidia-container (5.1.1-b50) ...
Setting up libcurand-dev-11-4 (10.2.5.297-1) ...
Setting up cuda-runtime-11-4 (11.4.19-1) ...
Setting up cuda-cudart-dev-11-4 (11.4.298-1) ...
Setting up libnvonnxparsers8 (8.5.2-1-cuda11.4) ...
Setting up libnvvp12 (2.2.7) ...
Setting up vpl2-demos (2.2.7) ...
Setting up libnvInfer-dev (8.5.2-1-cuda11.4) ...
Setting up nvidia-cuda (5.1.1-b50) ...
Setting up cuda-nvcc-11-4 (11.4.315-1) ...
Setting up libnvonnxparsers-dev (8.5.2-1-cuda11.4) ...
Setting up python3-9-vpl2 (2.2.7) ...
Setting up libnvvparsers-dev (8.5.2-1-cuda11.4) ...
Setting up cuda-compiler-11-4 (11.4.19-1) ...
Setting up cuda-libraries-dev-11-4 (11.4.19-1) ...
Setting up cuda-cupti-11-4 (11.4.298-1) ...
Setting up python3-libnvInfer (8.5.2-1-cuda11.4) ...
Setting up vpl2-dev (2.2.7) ...
update-alternatives: using /opt/nvidia/vpl2/include/vpl to provide /usr/include/vpl (vpl-dev) in auto mode
Setting up libnvInfer-plugin-dev (8.5.2-1-cuda11.4) ...
Setting up libnvInfer-bin (8.5.2-1-cuda11.4) ...
Setting up libnvInfer-samples (8.5.2-1-cuda11.4) ...
Setting up tensorrt (8.5.2-1-cuda11.4) ...
Setting up cuda-cupti-dev-11-4 (11.4.298-1) ...
Setting up nvidia-l4t-jetson-multimedia-api (35.3.1-20230319081403) ...
Setting up python3-8-vpl2 (2.2.7) ...
Setting up cuda-samples-11-4 (11.4.300-1) ...
Setting up vpl2-samples (2.2.7) ...
Setting up cuda-visual-tools-11-4 (11.4.19-1) ...
Setting up nvidia-tensorrt (5.1.1-b50) ...
Setting up python3-libnvInfer-dev (8.5.2-1-cuda11.4) ...
Setting up nvidia-vpl (5.1.1-b50) ...
Setting up cuda-command-line-tools-11-4 (11.4.19-1) ...
Setting up cuda-tools-11-4 (11.4.19-1) ...
Setting up nvidia-jetpack-runtime (5.1.1-b50) ...
Setting up nvidia-tensorrt-dev (5.1.1-b50) ...
Setting up cuda-toolkit-11-4 (11.4.19-1) ...
Setting up nvidia-vpl-dev (5.1.1-b50) ...
Setting up cuda-11-4 (11.4.19-1) ...
Setting up nvidia-cuda-dev (5.1.1-b50) ...
Setting up nvidia-jetpack-dev (5.1.1-b50) ...
Setting up nvidia-jetpack (5.1.1-b50) ...
Processing triggers for mime-support (3.6.0-1ubuntu1) ...
Processing triggers for gnome-menus (3.36.0-1ubuntu1) ...
Processing triggers for libc-bin (2.31-0ubuntu9) ...
Processing triggers for systemd (245.4-4ubuntu3.19) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for desktop-file-utils (0.24-1ubuntu3) ...
nvidia@nvidia-desktop:~$

```

图 3.2-3