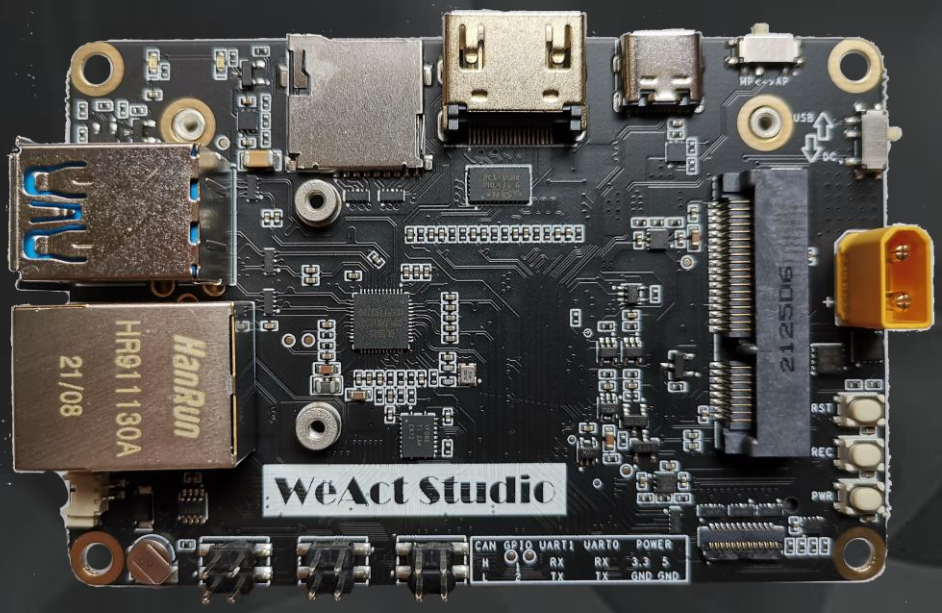


WeAct Studio

NANO&XAVIER TX2 NX CB

Tutorial

WeAct Studio

Catalog

REVISION HISTORY

Draft Date	Revision	Description
2021.9.20	V1.0	1. Init Version.
2021.12.26	V1.1	1. Add system backup and migration to NVME SSD
2022.1.22	V1.2	1. Add system migration to SD Card

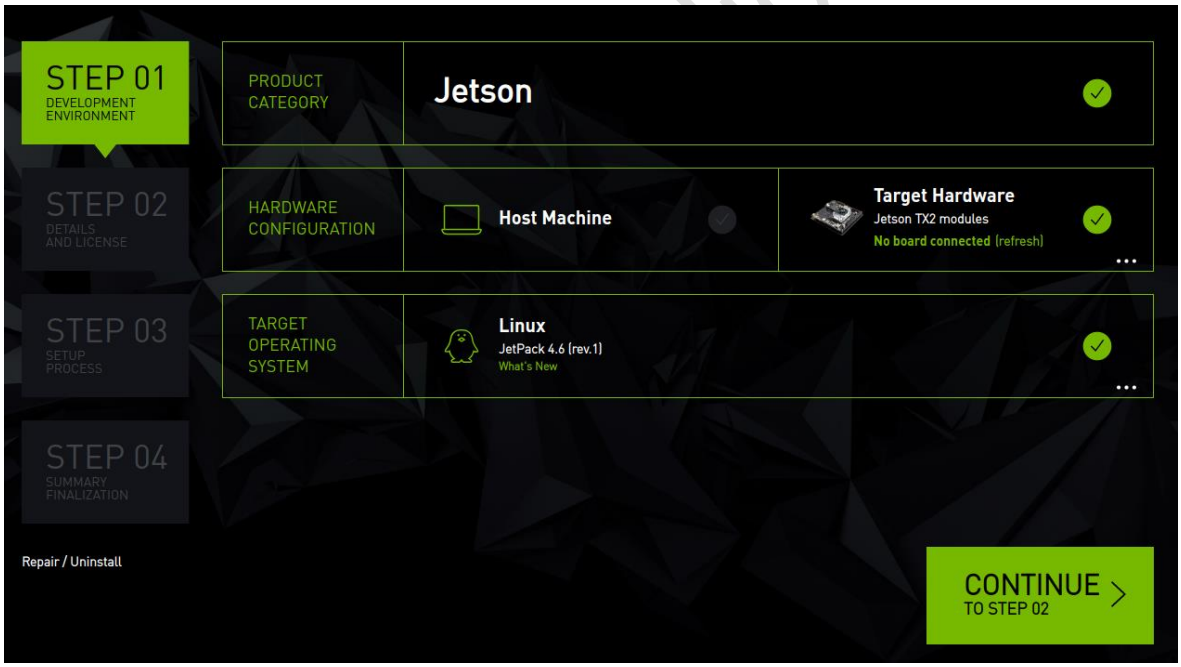
1. BUILD A FLASH ENVIRONMENT

a) First, you need a computer with Ubuntu 16.04 or above as the host to burn nano / NX, or you can install VMware on windows.

b) Download the latest SDK manager from NVIDIA and install it in Ubuntu 18.04 (You need to register an NVIDIA account, which will also be used later)

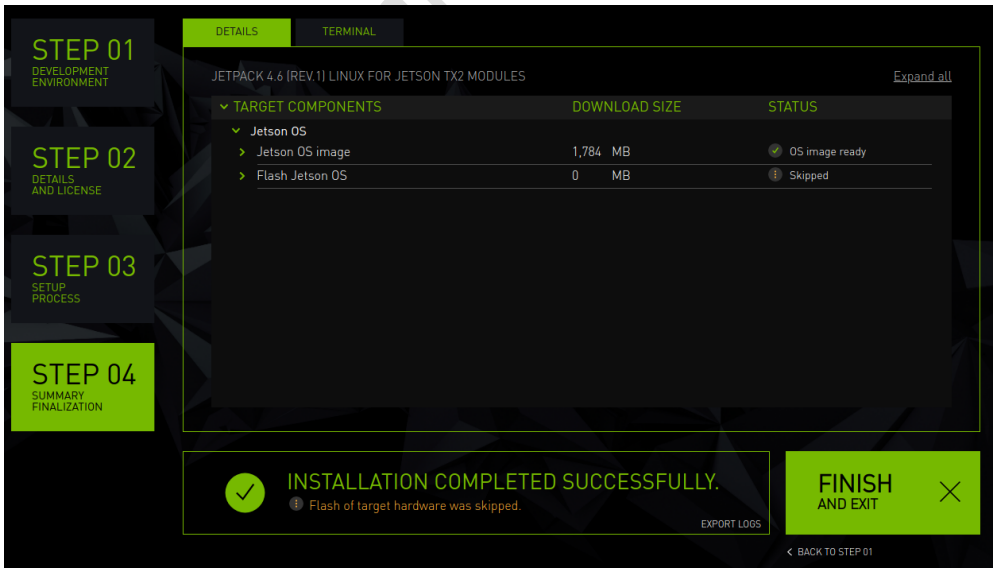
- SDK-Manager Download：https://developer.nvidia.com/nvidia-sdk-manager

c) Select the target hardware and jetpack version required, uncheck the host machine, take tx2nx as an example, and click continue.

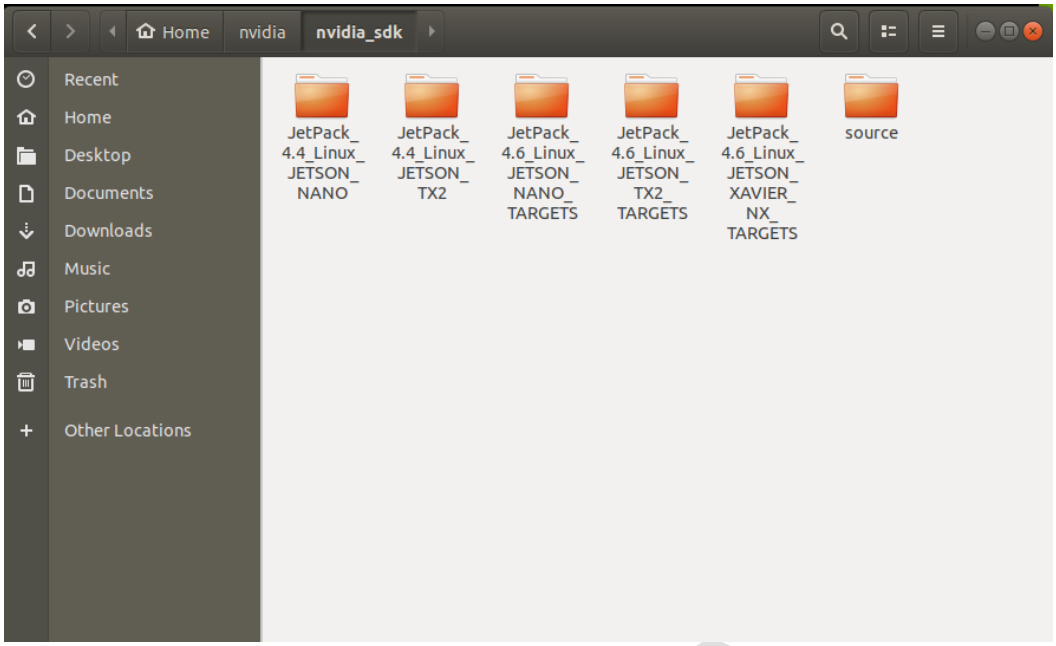


d) Check I accept the terms and conditions of the license agreements, uncheck the Jetson SDK components, and click continue to proceed to the next step.

P. S: please download and install in a smooth network environment. When the download or installation fails, click Retry to continue until all the status is installed and green is displayed. During the installation process, a network burning message will pop up and select skip.



e) After the installation is successful, the required files will be burned with the corresponding version under ~ / NVIDIA / nvidia_sdk /



f) Install Python support through sudo apt get install Python on the terminal for subsequent environment burning.

2. UPDATE DEVICE TREE FOR NANO / NX OR FLASH SYSTEM

P.S: the Weact device tree is different from the official device tree (Other functions are the same). If there is no need, the device tree can not be updated.

！！！ Note that updating the device tree does not affect any system files. Please be assured to update

NVIDIA and WeAct device tree diff

	NVIDIA	WeAct Studio
Nano-SD	Same	Same
Nano-EMMC	Cant use SD	Can use SD
TX2NX	Cant use **SD&UART1**	Can use **SD &UART1**
XavierNX	Cant use **SD**	Can use **SD**

a) Here, take tx2nx as an example, download the corresponding device tree file on the GitHub of Weact studio.
   - Github：https://github.com/WeActTC/Nano_TX2-Xavier_NX-CB

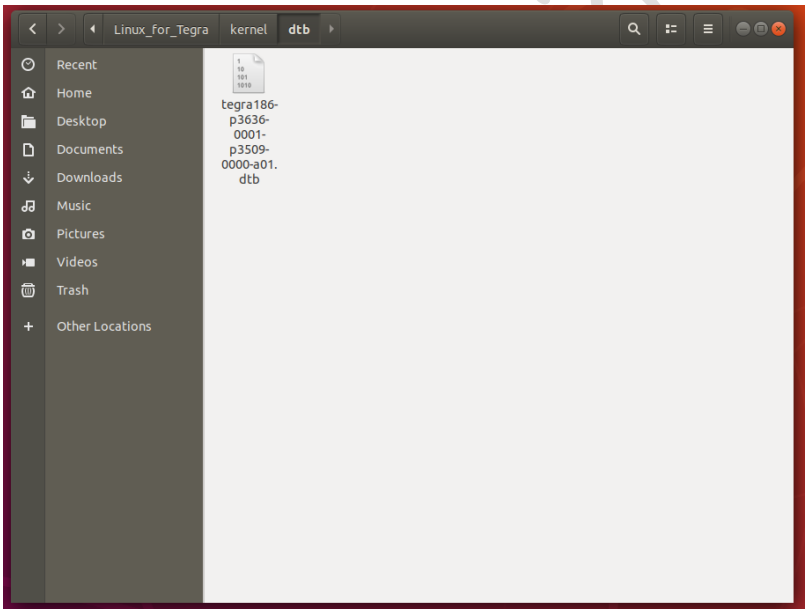
Update path and equipment tree name of each equipment tree

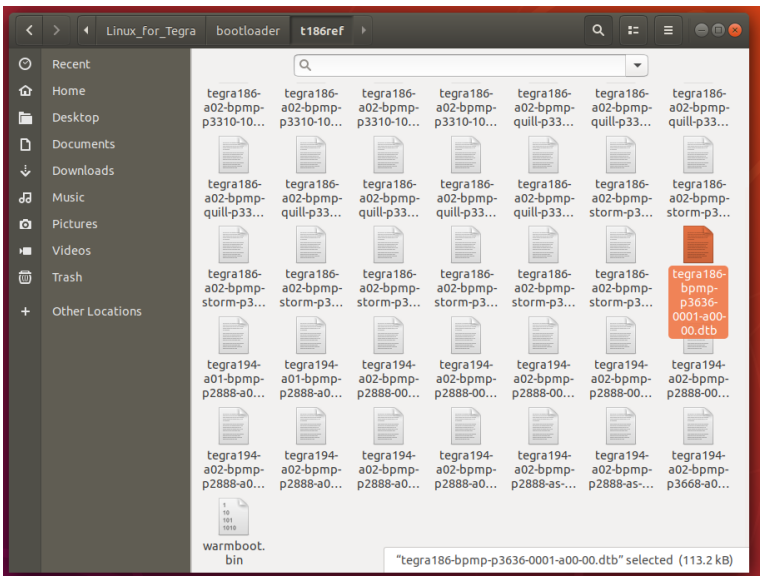
	Linux_for_Tegra/kernel/dtb	**Linux_for_Tegra/bootloader/t186ref(t210f)**
Nano-EMMC	**tegra210-p3448-0002-p3449-0000-b00**	**None**
TX2NX	**tegra186-p3636-0001-p3509-0000-a01**	**tegra186-bpmp-p3636-0001-a00-00**

XavierNX	**tegra194-p3668-all-p3509-0000**	**None**

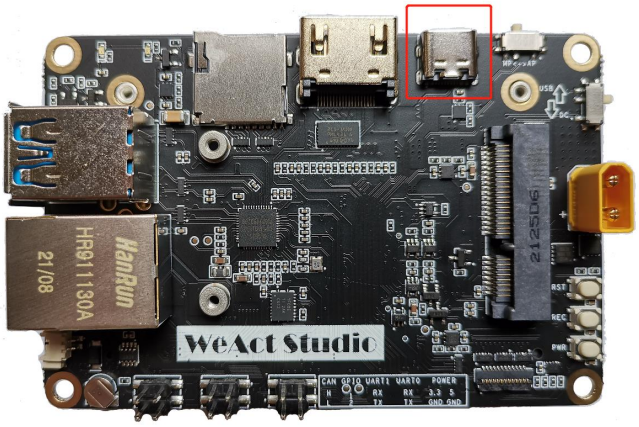
b) Find the corresponding version of the device tree

1. Enter **~/nvidia/nvidia_sdk/JetPack_4.6_Linux_JETSON_TX2_TARGETS/Linux_for_Tegra/kernel/dtb**，copy device-tree **tegra186-p3636-0001-p3509-0000-a01.dtb** to this dir.
2. Enter **~/nvidia/nvidia_sdk/JetPack_4.6_Linux_JETSON_TX2_TARGETS/Linux_for_Tegra/bootloader/t186ref**，Copy device-tree **tegra186-bpmp-p3636-0001-a00-00.dtb** to this dir【**Only TX2NX**】

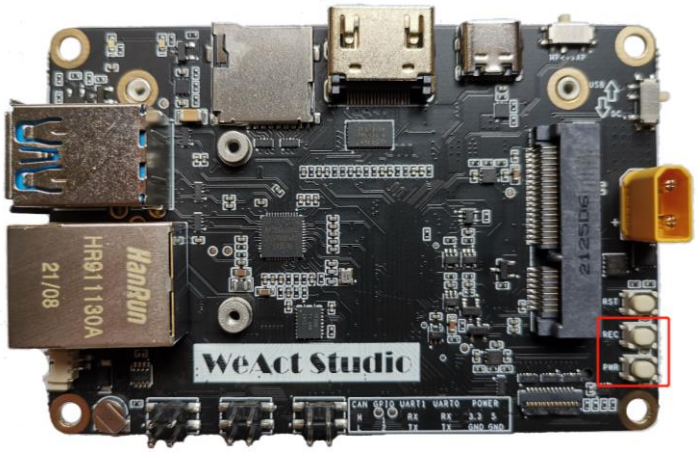




3. Use the USB type-C cable to connect the USB OTG interface on the carrier board.



4. Turn the power on key to MP (Manual power on), press the rec key, then press the PWR key to power on, release the rec key to enter the recovery mode, at this time, NVIDIA USB drive sign will appear in the lower right corner of VMware, or open the terminal and enter lsusb command, NVIDIA Corp will be found.



5. Enter **~/nvidia/nvidia_sdk/JetPack_4.6_Linux_JETSON_TX2_TARGETS/Linux_for_Tegra**, open command
   a) If you have a system, run **sudo ./flash.sh -r -k kernel-dtb jetson-xavier-nx-devkit-tx2-nx mmcblk0p1**, and updated the device-tree.
   b) Or you don’ t have a system, run **sudo ./flash.sh jetson-xavier-nx-devkit-tx2-nx mmcblk0p1,** and flash the system

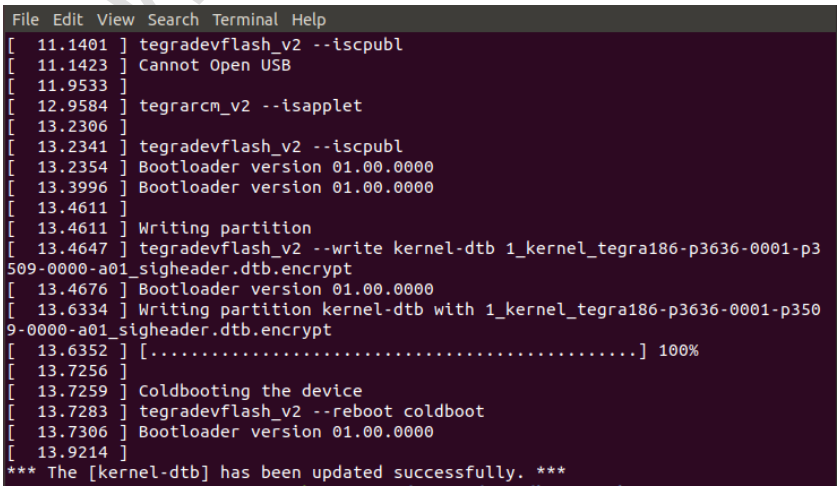
Device tree updated command

Device	Command
Nano-SD	sudo ./flash.sh -r -k DTB jetson-nano-qspi-sd mmcblk0p1
Nano-EMMC	sudo ./flash.sh -r -k DTB jetson-nano-emmc mmcblk0p1
TX2-NX	sudo ./flash.sh -r -k kernel-dtb jetson-xavier-nx-devkit-tx2-nx mmcblk0p1
Xavier-SD	sudo ./flash.sh -r -k kernel-dtb jetson-xavier-nx-devkit-qspi mmcblk0p1
Xavier-EMMC	sudo ./flash.sh -r -k kernel-dtb jetson-xavier-nx-devkit-emmc mmcblk0p1

System flash command

Device	Command
Nano-SD	sudo ./flash.sh jetson-nano-qspi-sd mmcblk0p1
Nano-EMMC	sudo ./flash.sh jetson-nano-emmc mmcblk0p1
TX2-NX	sudo ./flash.sh jetson-xavier-nx-devkit-tx2-nx mmcblk0p1
Xavier-SD	sudo ./flash.sh jetson-xavier-nx-devkit-qspi mmcblk0p1
Xavier-EMMC	sudo ./flash.sh jetson-xavier-nx-devkit-emmc mmcblk0p1

After updating the device tree, it will be successful! Display, as shown in the following figure.

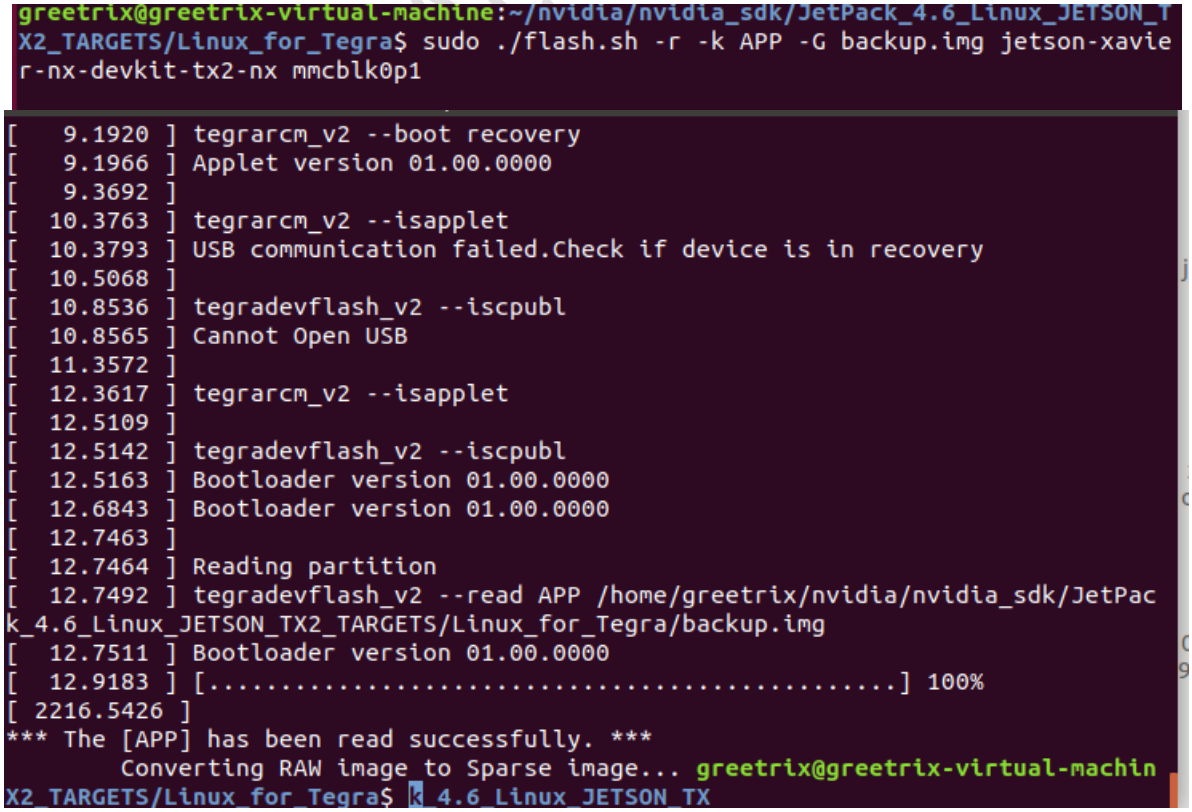


3. SYSTEM BACKUP AND RECOVERY

a) Refer to Chapter 2. No matter backup or image burning, enter the recovery mode. Note that the image is large. Please ensure that Ubuntu has sufficient space (> 40g).

b) Backup: take TX2NX as an example (for other devices, please refer to the previous chapter to modify the Jetson name) to backup the existing environment of the core board. Enter **~/nvidia/nvidia_sdk/JetPack_4.6_Linux_JETSON_TX2_TARGETS/Linux_for_Tegra, open the terminal.**

Use system backup command: **sudo ./flash.sh -r -k APP -G backup.img jetson-xavier-nx-devkit-tx2-nx mmcblk0p1.** Wait for the backup to complete, and there will be backup.img in the directory(it is recommended to copy a copy to another location for backup). At this time, the backup has been successful.



c) **Recovery**: Enter **~/nvidia/nvidia_sdk/JetPack_4.6_Linux_JETSON_TX2_TARGETS/Linux_for_Tegra,** copy the backup.img to **Liunx_for_Tegra/bootloader/** and rename with "**system.img**" , goto **Linux_for_Tegra** and open terminal : **sudo ./flash.sh -r jetson-xavier-nx-devkit-tx2-nx mmcblk0p1**, wait finishing.

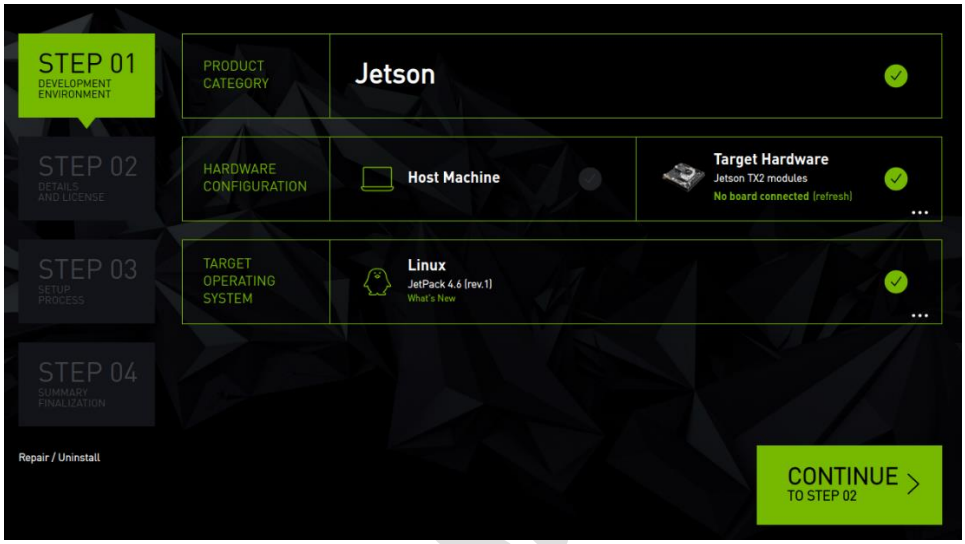
```
[  18.0000 ] Writing partition spe-fw_b with spe_sigheader.bin.encrypt
[  18.0298 ] [.............................................] 100%
[  18.0790 ] Writing partition mb2 with nvtboot_sigheader.bin.encrypt
[  18.1057 ] [.............................................] 100%
[  18.1596 ] Writing partition mb2_b with nvtboot_sigheader.bin.encrypt
[  18.1895 ] [.............................................] 100%
[  18.2416 ] Writing partition mts-preboot with preboot_d15_prod_cr_sigheader.bi
n.encrypt
[  18.2710 ] [.............................................] 100%
[  18.6760 ] Writing partition mts-preboot_b with preboot_d15_prod_cr_sigheader.
bin.encrypt
[  18.7053 ] [.............................................] 100%
[  18.7467 ] Writing partition SMD with slot_metadata.bin
[  18.7744 ] [.............................................] 100%
[  18.9037 ] Writing partition SMD_b with slot_metadata.bin
[  18.9302 ] [.............................................] 100%
[  18.9658 ] Writing partition VER_b with emmc_bootblob_ver.txt
[  18.9922 ] [.............................................] 100%
[  19.0322 ] Writing partition VER with emmc_bootblob_ver.txt
[  19.0592 ] [.............................................] 100%
[  19.0966 ] Writing partition master_boot_record with mbr_1_3.bin
[  19.1194 ] [.............................................] 100%
[  19.1525 ] Writing partition APP with system.img
[  19.1800 ] [........                                     ] 016%
```

```
t.encrypt
[ 1888.6372 ] Bootloader version 01.00.0000
[ 1888.8013 ] Writing partition MB1_BCT with mb1_cold_boot_bct_MB1_sigheader.b
.encrypt
[ 1888.8019 ] [.............................................] 100%
[ 1888.8706 ]
[ 1888.8837 ] tegradevflash_v2 --write MB1_BCT_b mb1_cold_boot_bct_MB1_sigheade
.bct.encrypt
[ 1888.8849 ] Bootloader version 01.00.0000
[ 1889.0452 ] Writing partition MB1_BCT_b with mb1_cold_boot_bct_MB1_sigheader
t.encrypt
[ 1889.0468 ] [.............................................] 100%
[ 1889.1180 ]
[ 1889.1181 ] Flashing completed

[ 1889.1181 ] Coldbooting the device
[ 1889.1436 ] tegradevflash_v2 --reboot coldboot
[ 1889.1449 ] Bootloader version 01.00.0000
[ 1889.3379 ]
*** The target t186ref has been flashed successfully. ***
Reset the board to boot from internal eMMC.
```

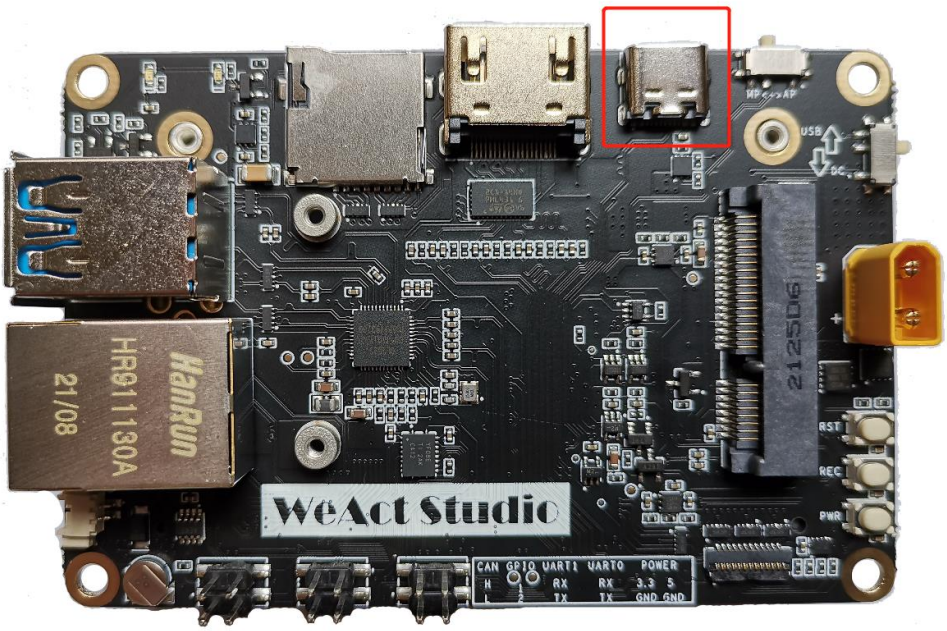
4. INSTALLING NVIDIA SDK

d) Select the target hardware and jetpack version required, uncheck the host machine, take tx2nx as an example, and click continue.



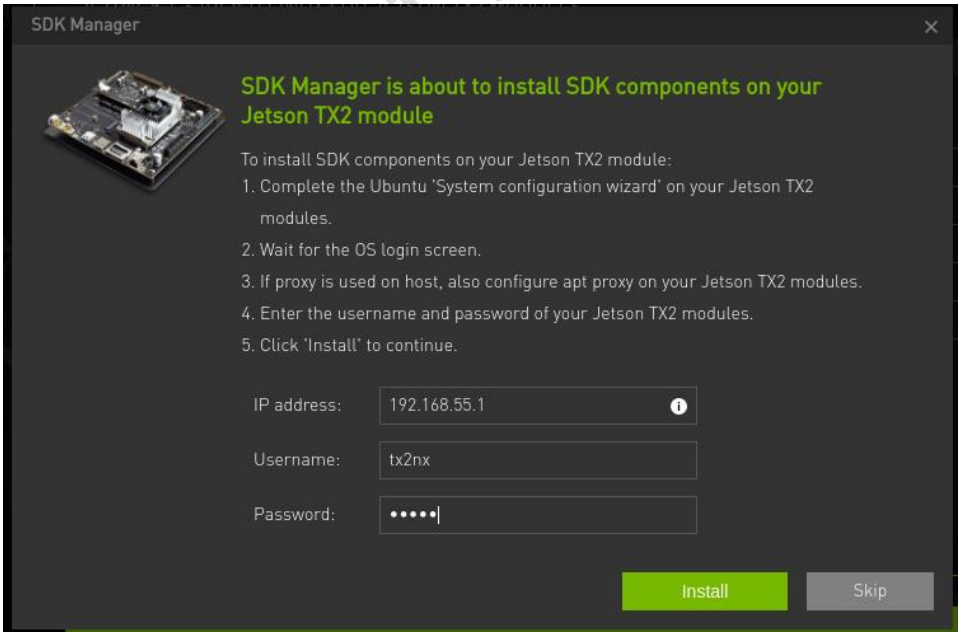
e) Check the required SDK components, check I accept the terms and conditions of the license agreements, and click continue to proceed to the next step.

f) Use the USB type-C cable to connect the USB OTG interface on the carrier board.



g) Turn the power on key to MP (manual power on) and press PWR key to power on. At this time, NVIDIA USB drive sign will appear in the lower right corner of VMware, or open the terminal and enter lsusb command to find NVIDIA Corp.

h) Enter the tx2nx account and password. Please keep the tx2nx terminal connected



i) Wait for the installation to complete.

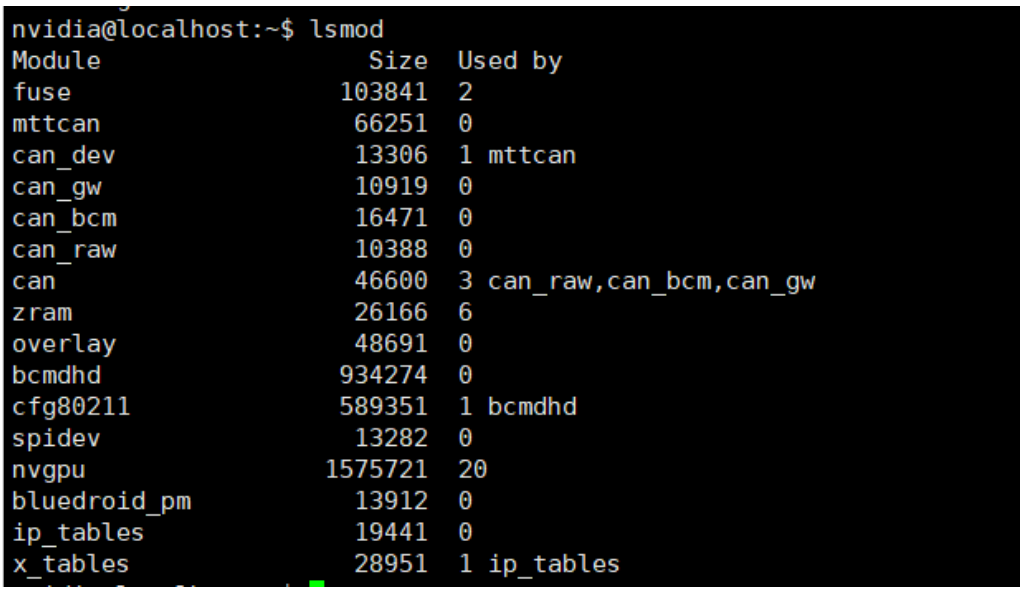
5. COMMUNICATION USING CAN

j) Two CAN controllers (CAN 0 / CAN 1) are integrated on TX2 NX / xaviernx. In addition, a CAN transceiver (CAN 0) is designed on the carrier board of Weact studio, which can be directly attached to the CAN physical bus.

k) TX2 NX / xaviernx has its own CANbus driver and is integrated into the image. It already supports CANbus without further processing. We need to install CANbus module. (enter the following command in the terminal or put it into rc.local to start the self startup)

```
modprobe can
modprobe can-raw
modprobe can-bcm
modprobe can-gw
modprobe can_dev
modprobe mttcan
```

l) Check whether the installation is successful through lsmod.



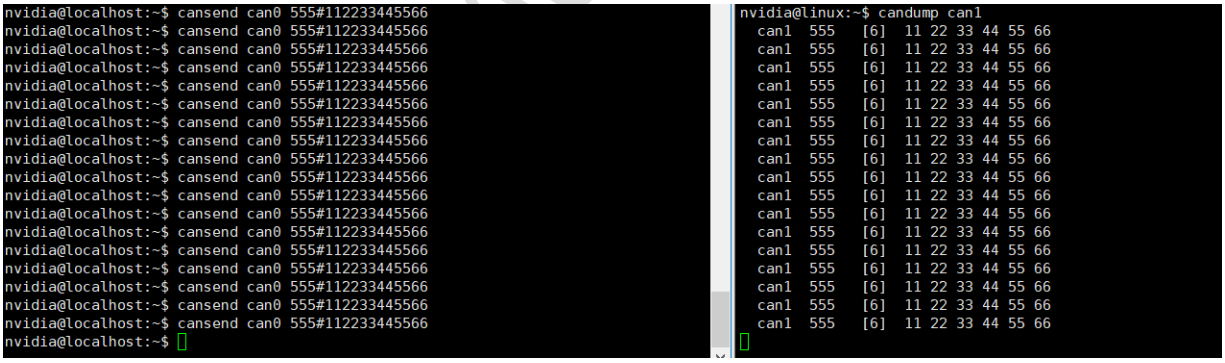
m) Configure CANbus attribute, which is similar to the baud rate setting of serial port

sudo ip link set can0 type can bitrate 500000
sudo ip link set up can0

n) Check whether the configuration is successful through ifconfig.

```
nvidia@localhost:~$ ifconfig
can0: flags=193<UP,RUNNING,NOARP>  mtu 16
        unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen 10  (UNSPEC)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 131
```

o) Can0 (Can1) via cansend at a terminal ××× Command to send data, and the other terminal completes the actual signal transceiver test through candump Can1 (can0).



6. GPIO USING ON SHELL

a) Nano / TX2 NX / Xavier NX can directly control GPIO input and output through shell commands.

	GPIO1	GPIO2
Nano	194	38
TX2-NX	338	269
Xavier-NX	196	105

b) Take tx2-nx gpio1 as an example
   - Activate IO first : sudo echo 338 > /sys/class/gpio/export
   - Set IO direction: echo out > /sys/class/gpio/gpio338/direction
   - Set output level : echo 1 > /sys/class/gpio/gpio338/value

7. SYSTEM MIGRATION TO NVME SSD

- WeAct-Nano&Xavier-TX2_NX-Cb with WeAct-MiniPCIE2M, support 2242/2230 NVME SSD, speed up 300M/s.

```
tx2nx@tx2nx:/mnt/ssd$ dd if=/dev/zero of=./largefile bs=1M count=1024
dd: failed to open './largefile': Permission denied
tx2nx@tx2nx:/mnt/ssd$ sudo dd if=/dev/zero of=./largefile bs=1M count=1024
1024+0 records in
1024+0 records out
1073741824 bytes (1.1 GB, 1.0 GiB) copied, 3.14822 s, 341 MB/s
tx2nx@tx2nx:/mnt/ssd$ sudo sh -c "sync && echo 3 > /proc/sys/vm/drop_caches"
tx2nx@tx2nx:/mnt/ssd$ dd if=./largefile of=/dev/null bs=4k
262144+0 records in
262144+0 records out
1073741824 bytes (1.1 GB, 1.0 GiB) copied, 3.08641 s, 348 MB/s
tx2nx@tx2nx:/mnt/ssd$
```

- **NVME SSD Configuration**:

- 1. Before configuration, ensure that the system can recognize the nvme SSD. The terminal command is **sudo fdisk – lu**

```
Disk /dev/nvme0n1: 119.2 GiB, 128035676160 bytes, 250069680 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
```

- 2. Set the NVME with GPT format:

  i. Command: **sudo parted /dev/nvme0n1**, enter the parted

```
tx2nx@tx2nx:~$ sudo parted /dev/nvme0n1
[sudo] password for tx2nx:
GNU Parted 3.2
Using /dev/nvme0n1
Welcome to GNU Parted! Type 'help' to view a list of commands.

(parted)
```

  ii. Command: **mklabel gpt**, make the label with **GPT**

```
(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
Yes/No? Yes
```

  iii. Command: **mkpart logical 0 -1**, make the part with **GPT**

```
(parted) mkpart logic 0 -1
Warning: The resulting partition is not properly aligned for best performance.
Ignore/Cancel? Ignore
```

iv. Command: **print**, see the result

```
(parted) print
Model: KBG40ZNS128G NVMe TOSHIBA 128GB (nvme)
Disk /dev/nvme0n1: 128GB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number  Start   End    Size   File system  Name   Flags
 1      17.4kB  128GB  128GB               logic
```

v. Command: **quit**

vi. Command: **sudo fdisk /dev/nvme0n1**

```
(parted) quit
Information: You may need to update /etc/fstab.

tx2nx@tx2nx:~$ sudo fdisk /dev/nvme0n1

Welcome to fdisk (util-linux 2.31.1).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.


Command (m for help):
```

vii. Command: **N**, add new part

```
Command (m for help): n
Partition number (2-128, default 2): 2
First sector (250067728-250069646, default 250068992):
Last sector, +sectors or +size{K,M,G,T,P} (250068992-250069646, default 250069646):

Created a new partition 2 of type 'Linux filesystem' and of size 327.5 KiB.
```

viii. Command: **P**, see the result

```
Device             Start       End   Sectors   Size Type
/dev/nvme0n1p1        34 250067727 250067694 119.2G Linux filesystem
/dev/nvme0n1p2 250068992 250069646       655 327.5K Linux filesystem
```

ix. Command: **quit**

x. Command: **sudo mke2fs -t ext4 /dev/nvme0n1p1**, format the part

```
tx2nx@tx2nx:~$ sudo mke2fs -t ext4 /dev/nvme0n1p1
mke2fs 1.44.1 (24-Mar-2018)
/dev/nvme0n1p1 contains a ext4 file system
        last mounted on / on Sun Dec 26 11:04:07 2021
Proceed anyway? (y,N)
```

xi. Command: **sudo mount /dev/nvme0n1p1/mnt**, if success, configurate NVME ok.

```
tx2nx@tx2nx:~$ sudo mount /dev/nvme0n1p1 /mnt
tx2nx@tx2nx:~$
```

➢ NVIDIA Jetson system migration (!!! Please refer to chapter 3 for system backup before migration):

✓ **Taking tx2nx as an example, other devices can replace the device name in the middle of the command, and the device name can refer to the above command**

➢ 1. Command: **git clone https://github.com/jetsonhacks/rootOnNVMe** or https://github.com/jetsonhacks/rootOnNVMe, download the script

➢ 2. Enter the **rootOnNVME** document, Command: **./copy-rootfs-ssd.sh**, copy the system file to NVME.

```
tx2nx@tx2nx:/home/script/rootOnNVMe-master$ ./copy-rootfs-ssd.sh
mount: /mnt: /dev/nvme0n1p1 already mounted on /mnt.
     17,380,838   0%    2.40MB/s    0:00:06 (xfr#39, ir-chk=1015/44887)
```

➢ 3. Command: **./setup-service.sh**, and configurate the booting

```
tx2nx@tx2nx:/home/script/rootOnNVMe-master$ ./setup-service.sh
==== AUTHENTICATING FOR org.freedesktop.systemd1.reload-daemon ===
Authentication is required to reload the systemd state.
Authenticating as: tx2nx,,, (tx2nx)
Password: Failed to reload daemon: Method call timed out
polkit-agent-helper-1: pam_authenticate failed: Authentication failure
Created symlink /etc/systemd/system/default.target.wants/setssdroot.service → /etc/systemd/system/setssdroot.service.
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.
```

➢ 4. Refer to chapter 2, make the device in recovery mode.

➢ 5. (In Ubuntu, refer to chapter 2): **Enter the ~/nvidia/nvidia_sdk/JetPack_4.6_Linux_JETSON_TX2_TARGETS/Linux_for_Tegra**, command: **sudo ./flash.sh jetson-xavier-nx-devkit-tx2-nx nvme0n1p1** and updated the EMMC boot

```
greetrix@greetrix-virtual-machine:~/nvidia/nvidia_sdk/JetPack_4.6_Linux_JETSON_TX2_TARGETS/Linux_for_Tegra$ sudo ./flash.sh jetson-xavier-nx-devkit-tx2-nx nvme0n1p1
```

```
[   37.3739 ] Coldbooting the device
[   37.3775 ] tegradevflash_v2 --reboot coldboot
[   37.3788 ] Bootloader version 01.00.0000
[   37.5711 ]
*** The target t186ref has been flashed successfully. ***
Make the target filesystem available to the device and reset the board to boot f
rom external nvme0n1p1.
```

- 6. Reboot TX2NX, command: **df -l**, now the system is in NVME SSD and migration successful.

```
tx2nx@tx2nx:~$ df -l
Filesystem     1K-blocks     Used Available Use% Mounted on
/dev/nvme0n1p1 122547172 11949920 104329176  11% /
none             1578060        0   1578060   0% /dev
tmpfs            1962748       52   1962696   1% /dev/shm
tmpfs            1962748    20764   1941984   2% /run
tmpfs               5120        4      5116   1% /run/lock
tmpfs            1962748        0   1962748   0% /sys/fs/cgroup
tmpfs             392548       12    392536   1% /run/user/120
tmpfs             392548        0    392548   0% /run/user/1000
```

8. SYSTEM MIGRATION TO SD CARD

a) SD configuration:

1. Make sure the system can recognize the SD card, command: **sudo fdisk -lu**

2. Set the SD with **GPT** format:

   i. Command: **sudo fdisk /dev/mmcblk1**, enter the SD configuration.

```
tx2nx@tx2nx:~/Desktop$ sudo fdisk /dev/mmcblk1

Welcome to fdisk (util-linux 2.31.1).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.
```

   ii. Command: **g**, new a GPT disklabel

```
Command (m for help): g
Created a new GPT disklabel (GUID: E39DF30E-48FE-B041-A6FA-5EFAEC223CEA).
```

   iii. Command: **n**, new partition

```
Command (m for help): n
Partition number (1-128, default 1):
First sector (2048-124735454, default 2048):
Last sector, +sectors or +size{K,M,G,T,P} (2048-124735454, default 124735454):
```

   iv. Command: **w**, save the configuration

```
Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.
```

   v. Command: **sudo mke2fs -t ext4 /dev/mmcblk1p1**, format the partition

```
tx2nx@tx2nx:~/Desktop$ sudo mke2fs -t ext4 /dev/mmcblk1p1
mke2fs 1.44.1 (24-Mar-2018)
Discarding device blocks: done
Creating filesystem with 15591675 4k blocks and 3899392 inodes
Filesystem UUID: 3c6ce310-2a86-4385-8dd0-86ab0089767e
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
```

   vi. Command: **sudo mount /dev/mmcblk1p1/mnt**, if success,

configurate the SD Card successfully.

```
tx2nx@tx2nx:~/Desktop$ sudo mount /dev/mmcblk1p1 /mnt
```

b) NVIDIA Jetson system migration (!!! Please refer to chapter 3 for system backup before migration):

- ✓ **Taking tx2nx as an example, other devices can replace the device name in the middle of the command, and the device name can refer to the above command**

i. Command: **git clone** https://github.com/jetsonhacks/rootOnNVMe or https://github.com/jetsonhacks/rootOnNVMe , download the script

ii. Modify **copy-rootfs-ssd.sh**, notes the mount

```
#!/bin/bash
# Mount the SSD as /mnt
# sudo mount /dev/nvme0n1p1 /mnt
# Copy over the rootfs from the SD card to the SSD
sudo rsync -axHAWX --numeric-ids --info=progress2 --exclude={"/dev/","/proc/","/sys/","/tmp/"
,"/run/","/mnt/","/media/*","/lost+found"} / /mnt
# We want to keep the SSD mounted for further operations
# So we do not unmount the SSD
```

iii. Enter the **rootOnNVMe**, Command: **./copy-rootfs-ssd.sh**, copy the system file to SD Card

```
tx2nx@tx2nx:~/Desktop/rootOnNVMe$ sudo ./copy-rootfs-ssd.sh
  1,149,753,593  71%   23.18MB/s    0:00:19  (xfr#6703, ir-chk=2715/12064)
```

iv. Refer to chapter 2, make the device in **recovery mode.**

v. (**In Ubuntu, refer to chapter 2**): Enter the **~/nvidia/nvidia_sdk/JetPack_4.6_Linux_JETSON_TX2_TARGETS/Linux_for_Tegra**, command: **sudo ./flash.sh jetson-xavier-nx-devkit-tx2-nx mmcblk1p1** and updated the EMMC boot

```
greetrix@greetrix-virtual-machine:~/nvidia/nvidia_sdk/JetPack_4.6_Linux_JETSON_TX2_TARGETS/Linux_for_Tegra$ sudo ./flash.sh jetson-xavier-nx-devkit-tx2-nx mmcblk1p1
```

```
[   30.4511 ] Coldbooting the device
[   30.4521 ] tegradevflash_v2 --reboot coldboot
[   30.4531 ] Bootloader version 01.00.0000
[   30.6253 ]
*** The target t186ref has been flashed successfully. ***
Make the target filesystem available to the device and reset the board to boot f
rom external mmcblk1p1.
```

vi. Reboot TX2NX, command: **df -l**, the system in SD Card

```
tx2nx@tx2nx:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/mmcblk1p1   59G   12G   44G  21% /
devtmpfs        1.6G     0  1.6G   0% /dev
tmpfs           1.9G   52K  1.9G   1% /dev/shm
tmpfs           1.9G   21M  1.9G   2% /run
tmpfs           5.0M  4.0K  5.0M   1% /run/lock
tmpfs           1.9G     0  1.9G   0% /sys/fs/cgroup
tmpfs           384M   12K  384M   1% /run/user/120
tmpfs           384M     0  384M   0% /run/user/1000
```

CONTACT WITH US

- Github：https://github.com/WeActTC
- Gitee：https://gitee.com/WeAct-TC
- Site：https://www.weact-tc.cn/
- Aliexpress: https://www.aliexpress.com/item/1005003334440054.html?spm=5261.ProductManageOnline.0.0.48104edfJwGktm