

LEX System

# 2NOR03 Porting Guide

NVIDIA Orin Jetson Linux 36.4.3( JetPack6.2 )

LEX SW Team

2025/5/9

## History

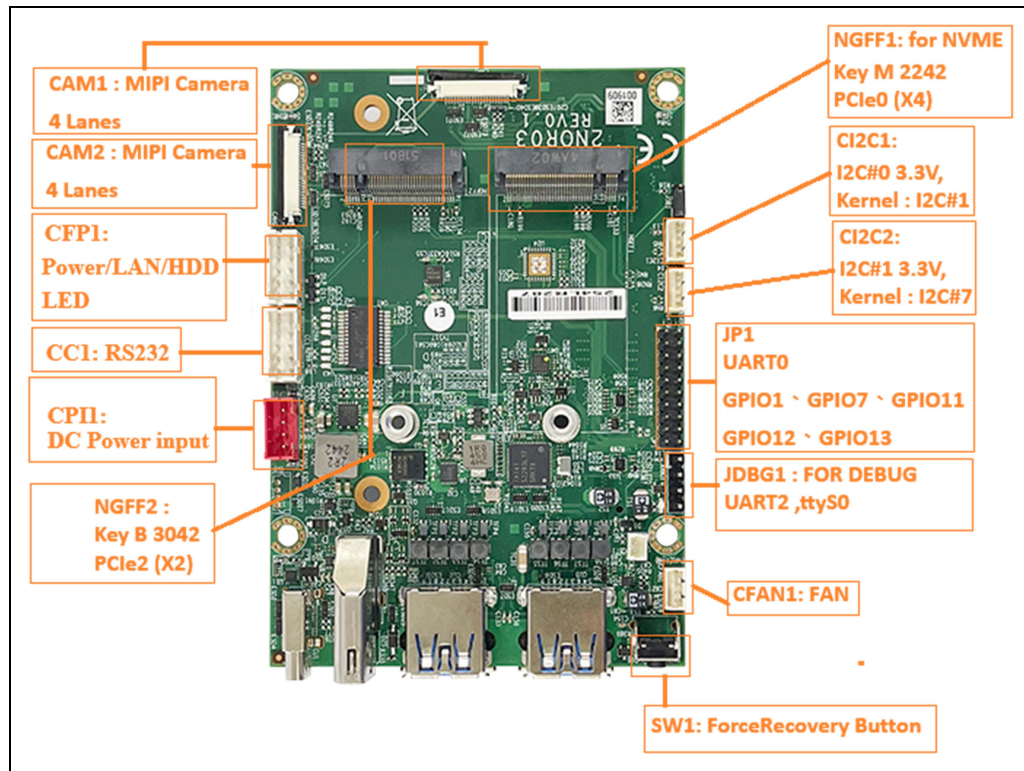
| Version | Date       | Description               |
|---------|------------|---------------------------|
| 1.0     | 2025/05/09 | Create File               |
|         | 2025/05/15 | Add TN MIPI Camera Driver |
|         | 2025/5/19  | Config MAC Address        |

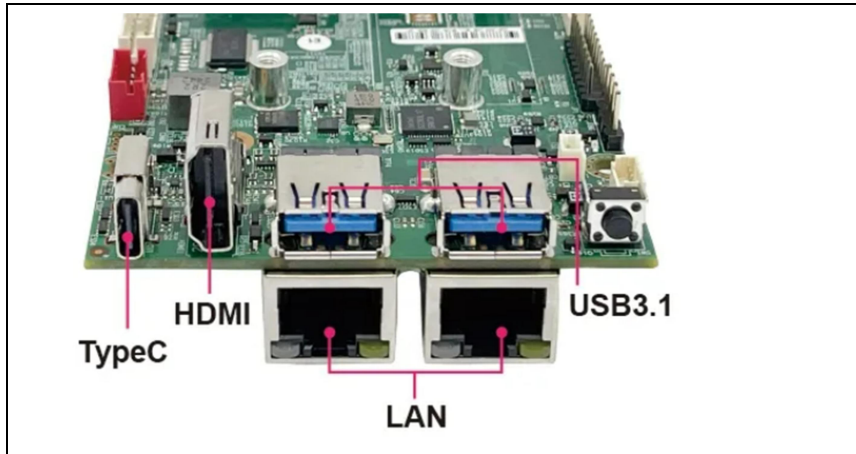
## Table of Contents

|                                                                             |    |
|-----------------------------------------------------------------------------|----|
| 1. Overview .....                                                           | 2  |
| 2. How to Create 2NOR03 OS - without Compile Jetson Linux source code ..... | 3  |
| 2.1. Download Driver Package (BSP) & Sample Root Filesystem (rootfs).....   | 4  |
| (1) Uncompress Driver Package (BSP) .....                                   | 5  |
| (2) Uncompress Sample Root Filesystem (rootfs) .....                        | 5  |
| (3) Execute “apply_binaries.sh” .....                                       | 6  |
| 2.2. Modify bootloader dts/dtsi file .....                                  | 6  |
| (1) Set the EEPROM size to 0.....                                           | 6  |
| (2) Set the GPIO from input to output (depending on customer needs)         | 7  |
| 2.3. Flash Image to 2NOR03’s NVME .....                                     | 9  |
| (1) Switch 2NOR03 into Force Recovery Mode .....                            | 9  |
| (2) Flash Image to NVME .....                                               | 10 |
| 2.4. Setup 2NOR03.....                                                      | 10 |
| (1) Copy 2NOR03’s dtb fils .....                                            | 10 |
| (2) Modify “/boot/extlinux/extlinux.conf” .....                             | 11 |
| (3) Install IGC Driver (I226IT, igc.ko).....                                | 12 |
| (4) Install TEVS Driver(TN MIPI Camera,tevs.ko) .....                       | 13 |
| (5) Config MAC address .....                                                | 13 |
| (6) Reboot system.....                                                      | 13 |
| 3. How to Create 2NOR03 OS - Compile Jetson Linux source code .....         | 13 |
| 3.1. Host computer Install and download necessary files .....               | 15 |
| (1) Install request packages.....                                           | 15 |
| (2) Install GCC toolchain.....                                              | 15 |
| (3) Set environment.....                                                    | 16 |
| 3.2. Download Driver Package (BSP) and Sample Root Filesystem (rootfs)...   | 16 |
| 3.3. Modify bootloader dts/dtsi file .....                                  | 16 |
| (1) Set the EEPROM size to 0.....                                           | 16 |
| (2) Set the GPIO from input to output (depending on customer needs)         |    |

|       |                                          |    |
|-------|------------------------------------------|----|
| 3.4.  | Download Jetson-Linux JP6.2 source ..... | 16 |
| 3.5.  | Enable IGC Driver (CONFIG_IGC=y).....    | 16 |
| 3.6.  | Enable USB3-2 and UART0 .....            | 16 |
| 3.7.  | Enable TPM and UART0.....                | 18 |
| 3.8.  | Modify SPI1/SPI3 dts .....               | 19 |
| 3.9.  | Setting MIPI Camera.....                 | 22 |
| 3.10. | Setting MIPI Camera driver.....          | 22 |
| 3.11. | Compile Image .....                      | 23 |
| 3.12. | Copy Image and dtb .....                 | 23 |
| 3.13. | Flash Image to NVME .....                | 23 |
| 3.14. | Setup 2NOR03.....                        | 23 |
| (1).  | Modify "extlinux.conf" .....             | 23 |
| (2).  | Install TN MIPI Camera driver .....      | 24 |
| (3).  | Config MAC Address .....                 | 24 |
| (4).  | Reboot System.....                       | 24 |

## 1. Overview





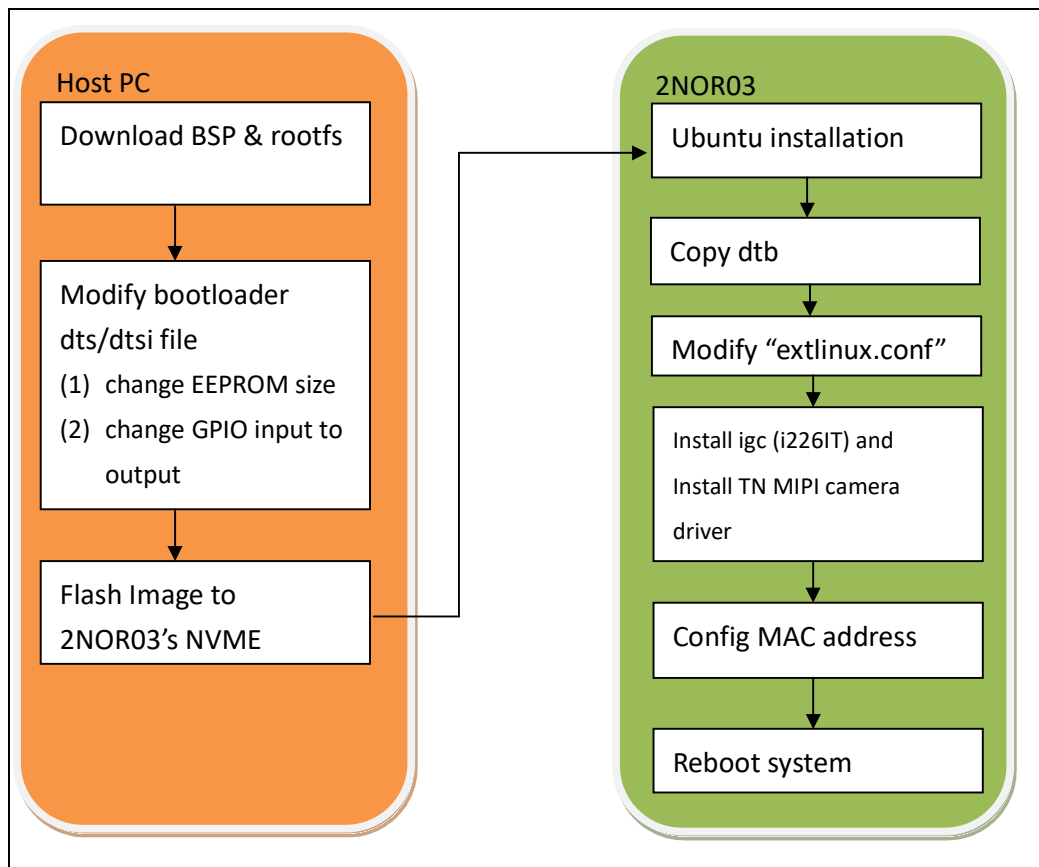
|                     |                                       |
|---------------------|---------------------------------------|
| Jetson Orin Nano 8G | tegra234-p3768-0000+p3767-0003-nv.dtb |
| Jetson NX 8G        | tegra234-p3768-0000+p3767-0001-nv.dtb |
| Jetson NX 16G       | tegra234-p3768-0000+p3767-0000-nv.dtb |

*The NVIDIA Jetson Orin dev kit includes an EEPROM, whereas the 2NOR03 does not. Therefore, please refrain from installing the operating system via the SDK Manager. However, you may install DeepStream through the SDK Manager.*

The following describes method for 2NOR03 on NVIDIA Jetson Orin OS

## 2. How to Create 2NOR03 OS - without Compile Jetson Linux source code

Jetson Linux 36.4.3 is part of JetPack 6.2, it includes Linux Kernel 5.15, an Ubuntu 22.04 based root file system



## 2.1. Download Driver Package (BSP) & Sample Root Filesystem (rootfs)

Download website : <https://developer.nvidia.com/embedded/jetson-linux-r3643>

## Vulkan Support on L4T

> Vulkan 1.3

### Downloads and Links

|         |                                                                                           |
|---------|-------------------------------------------------------------------------------------------|
|         | Jetson Orin Modules and Developer Kit                                                     |
| DRIVERS | <a href="#">Driver Package (BSP)</a> <span style="color: red;">First Download</span>      |
|         | <a href="#">Sample Root Filesystem</a> <span style="color: green;">Second Download</span> |
|         | <a href="#">Jetson Linux API Reference</a>                                                |
| SOURCES | <a href="#">Driver Package (BSP) Sources</a>                                              |
|         | <a href="#">Sample Root Filesystem Sources</a>                                            |
| DOCS    | <a href="#">Jetson AGX Orin Developer Kit User Guide</a>                                  |
|         | <a href="#">Jetson Orin Nano Developer Kit User Guide</a>                                 |
|         | <a href="#">Release Notes</a>                                                             |
|         | <a href="#">Jetson Linux Developer Guide (online version)</a>                             |
|         | <a href="#">Software License Agreement</a>                                                |
|         | <a href="#">Jetson Linux API Reference</a>                                                |
|         | <a href="#">Release sha1sum hashes</a>                                                    |

#### (1) Uncompress Driver Package (BSP)

For example :

```
#tar -jxf Jetson_Linux_R36.4.3_aarch64.tbz2 -C lex/

jinny@jinny-System-Product-Name:/media/jinny/nvidia-hdd/jinny$ mkdir lex
jinny@jinny-System-Product-Name:/media/jinny/nvidia-hdd/jinny$ tar -jxf Jetson_Linux_R36.4.3_aarch64.tbz2 -C lex/
jinny@jinny-System-Product-Name:/media/jinny/nvidia-hdd/jinny$ cd lex
jinny@jinny-System-Product-Name:/media/jinny/nvidia-hdd/jinny/lex$ ls
linux_for_Tegra
jinny@jinny-System-Product-Name:/media/jinny/nvidia-hdd/jinny/lex$
```

#### (2) Uncompress Sample Root Filesystem (rootfs)

For example :

```
# sudo tar -jxf Tegra_Linux_Sample-Root-Filesystem_R36.4.3_aarch64.tbz2 -C
lex/Linux_for_Tegra/rootfs
```

```

jimmy@jimmy-System-Product-Name:/media/jimmy/nvidia-hdd/jimmy$ sudo tar -jxf Tegra_Linux_Sample-Root-Filesystem_R36.4.3_aarch64.tbz2 -C lex/Linux_for_Tegra/rootfs/
jimmy@jimmy-System-Product-Name:/media/jimmy/nvidia-hdd/jimmy$ cd lex/Linux_for_Tegra/rootfs/
jimmy@jimmy-System-Product-Name:/media/jimmy/nvidia-hdd/jimmy/lex/Linux_for_Tegra/rootfs$ ls -al
total 84
drwxr-xr-x 18 root root 4096  4 30 09:51 .
drwxr-xr-x 10 jimmy jimmy 4096  8 09:56 ..
lrwxrwxrwx 1 root root 7 18 2023 bin -> usr/bin
drwxr-xr-x 2 root root 4096 18 2022 boot
drwxr-xr-x 2 root root 4096 18 2023 dev
drwxr-xr-x 137 root root 12288  4 14:31 etc
drwxr-xr-x 2 root root 4096 18 2022 home
lrwxrwxrwx 1 root root 7 18 2023 lib -> usr/lib
drwxr-xr-x 2 root root 4096 18 2023 media
drwxr-xr-x 2 root root 4096 18 2023 mnt
drwxr-xr-x 2 root root 4096 18 2023 opt
drwxr-xr-x 2 root root 4096 18 2022 proc
-rw-rw-r-- 1 jimmy jimmy 62  8 09:42 README.txt
drwx----- 3 root root 4096 22 2023 root
drwxr-xr-x 19 root root 4096 17 2024 run
lrwxrwxrwx 1 root root 8 18 2023/sbin -> usr/sbin
drwxr-xr-x 2 root root 4096  1 2022 snap
drwxr-xr-x 2 root root 4096 18 2023 srv
drwxr-xr-x 2 root root 4096 18 2022 sys
drwxrwxrwt 2 root root 4096  4 14:30 tmp
drwxr-xr-x 11 root root 4096 18 2023 usr
drwxr-xr-x 14 root root 4096 22 2023 var

```

### (3) Execute “apply\_binaries.sh”

For example :

```
#sudo ./apply_binaries.sh
```

```

jimmy@jimmy-System-Product-Name:/media/jimmy/nvidia-hdd/jimmy/lex/Linux_for_Tegra$ sudo ./apply_binaries.sh
Using rootfs directory of: /media/jimmy/nvidia-hdd/jimmy/lex/Linux_for_Tegra/rootfs
Installing extlinux.conf into /boot/extlinux in target rootfs
/media/jimmy/nvidia-hdd/jimmy/lex/Linux_for_Tegra/nv_tegra/nv-apply-debs.sh
Root file system directory is /media/jimmy/nvidia-hdd/jimmy/lex/Linux_for_Tegra/rootfs
Copying public debian packages to rootfs
Skipping installation of nvidia-igx-oem-config_36.4.3-20250107174145_arm64.deb ....
Skipping installation of nvidia-igx-systemd-reboot-hooks_36.4.3-20250107174145_arm64.deb ....
Skipping installation of nvidia-l4t-dgpu-apt-source_36.4.3-20250107174145_arm64.deb ....
Skipping installation of nvidia-l4t-dgpu-config_36.4.3-20250107174145_arm64.deb ....
Skipping installation of nvidia-l4t-dgpu-tools_36.4.3-20250107174145_arm64.deb ....
Skipping installation of nvidia-l4t-dgpu-x11_36.4.3-20250107174145_arm64.deb ....
Skipping installation of nvidia-l4t-factory-service_36.4.3-20250107174145_arm64.deb ....
Skipping installation of nvidia-igx-bootloader_36.4.3-20250107174145_arm64.deb ....
Skipping installation of nvidia-l4t-jetson-orin-nano-qspi-updater_36.4.3-20250107174145_arm64.deb ....
Start L4T BSP package installation
QEMU binary is not available, looking for QEMU from host system
Found /usr/bin/qemu-aarch64-static
Installing QEMU binary in rootfs
/media/jimmy/nvidia-hdd/jimmy/lex/Linux_for_Tegra/rootfs /media/jimmy/nvidia-hdd/jimmy/lex/Linux_for_Tegra
Host qemu-aarch64-static version: 4.2.1
Installing BSP Debian packages in /media/jimmy/nvidia-hdd/jimmy/lex/Linux_for_Tegra/rootfs
Selecting previously unselected package nvidia-l4t-core.
(Reading database ... 167181 files and directories currently installed.)
Preparing to unpack .../nvidia-l4t-core_36.4.3-20250107174145_arm64.deb ...
Add config /lib/modprobe.d/aliases.conf
Add config /lib/modprobe.d/fbdev-blacklist.conf
Add config /lib/modprobe.d/systemd.conf
Cleaning up the temporary directory for updating the initrd..
/media/jimmy/nvidia-hdd/jimmy/lex/Linux_for_Tegra
Removing QEMU binary from rootfs
Removing stashed Debian packages from rootfs
L4T BSP package installation completed!
Disabling NetworkManager-wait-online.service
Disable the ondemand service by changing the runlevels to 'K'
Success!
jimmy@jimmy-System-Product-Name:/media/jimmy/nvidia-hdd/jimmy/lex/Linux_for_Tegra$

```

*The basic setup of JP6.2 has been completed. Next, proceed with modifying the functions of the 2NOR03 carrier board.*

## 2.2. Modify bootloader dts/dtsi file

### (1) Set the EEPROM size to 0

*The NVIDIA Jetson Orin dev kit includes an EEPROM, whereas the 2NOR03 does not.*

Modify "bootloader/generic/BCT/tegra234-mb2-bct-misc-p3767-0000.dts"

```
jimmy@jimmy-System-Product-Name:/media/jimmy/nvidia-hdd/jimmy/lex/Linux_for_Tegra$ gedit bootloader/generic/BCT/tegra234-mb2-bct-misc-p3767-0000.dts
jimmy@jimmy-System-Product-Name:/media/jimmy/nvidia-hdd/jimmy/lex/Linux_for_Tegra$

1 /dts-v1/;
2
3 #include "tegra234-mb2-bct-common.dtsi"
4
5 / {
6     mb2-misc {
7         eeprom {
8             cvm_eeprom_i2c_instance = <0>;
9             cvm_eeprom_i2c_slave_address = <0xa0>;
10            cvm_eeprom_read_size = <0x100>;
11            cvb_eeprom_i2c_instance = <0x0>;
12            cvb_eeprom_i2c_slave_address = <0xae>;
13            //cvb_eeprom_read_size = <0x100>;
14            cvb_eeprom_read_size = <0x0>;
15        };
16    };
17 };
```

(2) Set the GPIO from input to output (depending on customer needs)

Modify "bootloader/tegra234-mb1-bct-gpio-p3767-hdmi-a03.dtsi"

```
gpio_main_default: default {
    gpio-input = <
        TEGRA234_MAIN_GPIO(B, 0)
        TEGRA234_MAIN_GPIO(Y, 0)
        TEGRA234_MAIN_GPIO(Y, 1)
        TEGRA234_MAIN_GPIO(Y, 2)
        TEGRA234_MAIN_GPIO(Y, 3)
        TEGRA234_MAIN_GPIO(Y, 4)
        TEGRA234_MAIN_GPIO(Z, 1)
        TEGRA234_MAIN_GPIO(Z, 3)
        TEGRA234_MAIN_GPIO(Z, 4)
        TEGRA234_MAIN_GPIO(Z, 5)
        TEGRA234_MAIN_GPIO(Z, 6)
        TEGRA234_MAIN_GPIO(Z, 7)
        TEGRA234_MAIN_GPIO(P, 6)
        //TEGRA234_MAIN_GPIO(Q, 5)
        //TEGRA234_MAIN_GPIO(Q, 6)
        TEGRA234_MAIN_GPIO(R, 4)
        TEGRA234_MAIN_GPIO(R, 5)
        TEGRA234_MAIN_GPIO(M, 0)
        //TEGRA234_MAIN_GPIO(N, 1)
        TEGRA234_MAIN_GPIO(G, 0)
        //TEGRA234_MAIN_GPIO(G, 6)
        TEGRA234_MAIN_GPIO(G, 7)
        //TEGRA234_MAIN_GPIO(H, 0)
        TEGRA234_MAIN_GPIO(H, 7)
        TEGRA234_MAIN_GPIO(I, 0)
        TEGRA234_MAIN_GPIO(I, 1)
        TEGRA234_MAIN_GPIO(I, 2)
        TEGRA234_MAIN_GPIO(AC, 6)
        TEGRA234_MAIN_GPIO(L, 2)
    >
```

```

gpio-output-high = <
    TEGRA234_MAIN_GPIO(Q, 3)
    TEGRA234_MAIN_GPIO(A, 0)
//lex++
    TEGRA234_MAIN_GPIO(Q, 5)
    TEGRA234_MAIN_GPIO(Q, 6)
    TEGRA234_MAIN_GPIO(N, 1)
    TEGRA234_MAIN_GPIO(G, 6)
    TEGRA234_MAIN_GPIO(H, 0)
//lex++ end
>;

```

Modify

“bootloader/generic/BCT/tegra234-mb1-bct-pinmux-p3767-hdmi-a03.dtsi”

```

soc_gpio32_pq5 {
    nvidia,pins = "soc_gpio32_pq5";
    nvidia,function = "rsvd0";
    nvidia,pull = <TEGRA_PIN_PULL_UP>;
    /*
    nvidia,tristate = <TEGRA_PIN_ENABLE>;
    nvidia,enable-input = <TEGRA_PIN_ENABLE>;
    */
    nvidia,tristate = <TEGRA_PIN_DISABLE>;
    nvidia,enable-input = <TEGRA_PIN_DISABLE>;
    nvidia,io-high-voltage = <TEGRA_PIN_DISABLE>;
    nvidia,lpdr = <TEGRA_PIN_DISABLE>;
};

soc_gpio33_pq6 {
    nvidia,pins = "soc_gpio33_pq6";
    nvidia,function = "rsvd0";
    nvidia,pull = <TEGRA_PIN_PULL_UP>;
    /*
    nvidia,tristate = <TEGRA_PIN_ENABLE>;
    nvidia,enable-input = <TEGRA_PIN_ENABLE>;
    */
    nvidia,tristate = <TEGRA_PIN_DISABLE>;
    nvidia,enable-input = <TEGRA_PIN_DISABLE>;
    nvidia,io-high-voltage = <TEGRA_PIN_DISABLE>;
    nvidia,lpdr = <TEGRA_PIN_DISABLE>;
};

```

```

soc_gpio39_pn1 {
    nvidia,pins = "soc_gpio39_pn1";
    nvidia,function = "rsvd1";
    /*
    nvidia,pull = <TEGRA_PIN_PULL_DOWN>;
    nvidia,tristate = <TEGRA_PIN_ENABLE>;
    nvidia,enable-input = <TEGRA_PIN_ENABLE>;
    */
    nvidia,pull = <TEGRA_PIN_PULL_UP>;
    nvidia,tristate = <TEGRA_PIN_DISABLE>;
    nvidia,enable-input = <TEGRA_PIN_DISABLE>;
    nvidia,lpdr = <TEGRA_PIN_DISABLE>;
};

soc_gpio19_pg6 {
    nvidia,pins = "soc_gpio19_pg6";
    nvidia,function = "rsvd1";
    /*
    nvidia,pull = <TEGRA_PIN_PULL_DOWN>;
    nvidia,tristate = <TEGRA_PIN_ENABLE>;
    nvidia,enable-input = <TEGRA_PIN_ENABLE>;
    */
    nvidia,pull = <TEGRA_PIN_PULL_UP>;
    nvidia,tristate = <TEGRA_PIN_DISABLE>;
    nvidia,enable-input = <TEGRA_PIN_DISABLE>;
    nvidia,lpdr = <TEGRA_PIN_DISABLE>;
};

soc_gpio21_ph0 {
    nvidia,pins = "soc_gpio21_ph0";
    nvidia,function = "rsvd0";
    /*
    nvidia,pull = <TEGRA_PIN_PULL_DOWN>;
    nvidia,tristate = <TEGRA_PIN_ENABLE>;
    nvidia,enable-input = <TEGRA_PIN_ENABLE>;
    */
    nvidia,pull = <TEGRA_PIN_PULL_UP>;
    nvidia,tristate = <TEGRA_PIN_DISABLE>;
    nvidia,enable-input = <TEGRA_PIN_DISABLE>;
    nvidia,lpdr = <TEGRA_PIN_DISABLE>;
};

```

### 2.3. Flash Image to 2NOR03's NVME

#### (1) Switch 2NOR03 into Force Recovery Mode

Connect your Linux host computer to the appropriate USB port on your 2NOR03 (USB typeC)

1. Ensure that the 2NOR03 is powered off.
2. Press and hold down the Force Recovery button (SW1).
3. 2NOR03 Power on.
4. Release the Force Recovery button (SW1)

***To Determine Whether the 2NOR03 Is in Force Recovery Mode***

***The Jetson module is in Force Recovery Mode if you see the message:***

***For example: Orin NX 16GB***

Bus 001 Device 058: ID 0955:7423 NVIDIA Corp

## (2) Flash Image to NVME

For example :

```
#sudo ./tools/kernel_flash/l4t_initrd_flash.sh --external-device nvme0n1p1 -c
tools/kernel_flash/flash_l4t_t234_nvme.xml -p " -c
bootloader/generic/cfg/flash_t234_qspi.xml" --showlogs --network usb0
p3509-a02-p3767-0000 internal
```

```
p -> c bootloader/generic/cfg/flash_t234_gspi.xml --showlogs --network usb0 p3589-a0-p3767-0000 Internal
Please install the Secureboot package to use Tegra flash for fused boot
Entry added by NVIDIA Unified Flash tool
/media/jlmy/nvidia-hdd/jlmy/lex/Linux_for_Tegra/tools/kernel_flash/tmp_127.0.0.1(rw,nohide,inscure,no_subtree_check,async,no_root_squash)
root@another-replica:~# already running. Aborting
Report list for localhost:
/media/jlmy/nvidia-hdd/jlmy/lex/Linux_for_Tegra/tools/kernel_flash/tmp_127.0.0.1
=====
generic/cfg/flash_t234_gspi.xml --showlogs --network usb0 p3589-a0-p3767-0000 Internal
=====
Step 1: Generate Flash packages *
=====
Create folder to store images to flash
Generate Image for internal storage devices
Generate images to be flashed
ADDITIONAL_OTB_OVERLAY="" /media/jlmy/nvidia-hdd/jlmy/lex/Linux_for_Tegra/Flash.sh --no-flash --sign -c bootloader/generic/cfg/flash_t234_gspi.xml p3589-a0-p3767-0000 Internal
=====
# L4T BSP Information:
# R3x : REV15TON: 4.3
# User release: 0.0
=====
cd $(ls -ld /dev/disk/by-id/*nvme*)/..
cpio is hdparm:sdac9f303d35fc0000015f0bc0e
Copying device.config (/media/jlmy/nvidia-hdd/jlmy/lex/Linux_for_Tegra/bootloader/generic/BCT/t2gta234-br-bct-device-p3767-0000.dts)... done.
Copying nvsic.config (/media/jlmy/nvidia-hdd/jlmy/lex/Linux_for_Tegra/bootloader/generic/BCT/t2gta234-br-bct-nvsc-p3767-0000.dts)... done.
Copying fuses.fuse_parms (/media/jlmy/nvidia-hdd/jlmy/lex/Linux_for_Tegra/bootloader/generic/BCT/t2gta234-br-bct-diag-boot-dts)... done
Copying mnratcatch.config (/media/jlmy/nvidia-hdd/jlmy/lex/Linux_for_Tegra/bootloader/generic/BCT/t2gta234-br-bct-ratchet-p3767-0000.dts)... done.
=====
nvidia/custinfo/jlmy/nvidia-hdd/jlmy/lex/Linux_for_Tegra/bootloader/fuse_t234.xml reused.
./tegraflash.py -chip bx21m --aplet "/media/jlmy/nvidia-hdd/jlmy/lex/Linux_for_Tegra/bootloader/mbl_1234prod.bin" --skipuid -cfg readinfo_t234_nml_prod.xml --dev_params tegra234-br-bct-diag-boot-d
-t234-devconfig-tegrat234-mbt-device-p3767-0000.dts --nvsic config t2gta234-mbt-bct-nvsc-p3767-0000.dts --bins "mb2_applet appler_t234-bits" --mnratcatcher config tegrat234-mbt-bct-ratchet-p3767-0000.d
t234-bits t234-bin fuse_t234.xml dump epronn cvm bin; dump try.custinfo custinfo.out.bin; reboot recovery"
welcome to Tegra Flash
Version 1.0.0
Setting /mnt/internal/qspi_bootload_ver.txt to (609 bytes) into /dev/ndvdms:6677792
Copied 169 bytes from /mnt/internal/qspi_bootload_ver.txt to address 0xb0f60000 in flash
Writing gpt_secondary_3_0.bin (partition: secondary gpt) into /dev/ndvd
Last checksum matched for /mnt/internal/qspi_bootload_ver_3_0.bin
Writing /mnt/internal/qpi_bootload_ver_3_0.bin (16896 bytes) into /dev/ndvd:6709398
Copied 16896 bytes from /mnt/internal/qpi_bootload_ver_3_0.bin to address 0xb0f60000 in flash
2251: ldt:flash_from_kernel: Successfully flash the qspi
2251: ldt:flash_from_kernel: Flushing success.
2251: ldt: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
Press any key to continue booting up
Log saved to Linux_for_Tegra/intelqpio/Flash-1-0-20258039-103839_Log
```

*After the flashing process is complete, the NVIDIA Orin will reboot and display the Ubuntu installation screen.*

## 2.4. Setup 2NOR03

After completing the Ubuntu installation, proceed with configuring the system settings

(1) Copy 2NOR03's dtb files

copy files to /boot

|                     |                                       |
|---------------------|---------------------------------------|
| Jetson Orin Nano 8G | tegra234-p3768-0000+p3767-0003-nv.dtb |
| Jetson NX 8G        | tegra234-p3768-0000+p3767-0001-nv.dtb |
| Jetson NX 16G       | tegra234-p3768-0000+p3767-0000-nv.dtb |

|                                |                                                   |
|--------------------------------|---------------------------------------------------|
| OVERLAYS , for SPI1/SPI3       | tegra234-p3767-0000+p3509-a02-hdr40.dtbo          |
| OVERLAYS , for TVN MIPI-Camera | tegra234-p3767-camera-0044-tevs-dual-overlay.dtbo |

## (2) Modify “/boot/extlinux/extlinux.conf”

Create new LABEL “lex” and set to default

DEFAULT lex

LABEL lex

MENU LABEL lex kernel

LINUX /boot/Image

INITRD /boot/initrd

FDT /boot/tegra234-p3768-0000+p3767-0000-nv.dtb

OVERLAYS

/boot/tegra234-p3767-camera-0044-tevs-dual-overlay.dtbo,/boot/tegra234-p3767-000+p3509-a02-hdr40.dtbo

APPEND \${cbootargs} root=PARTUUID=a7be4080-ee56-482a-b639-1514ef9ea005  
rw rootwait rootfstype=ext4 mminit\_loglevel=4 console=ttyTCU0,115200  
firmware\_class.path=/etc/firmware fbcon=map:0 nospectre\_bhb video=efifb:off  
console=tty0

LABEL lex

MENU LABEL lex kernel

LINUX /boot/Image

FDT /boot/tegra234-p3768-0000+p3767-0000-nv.dtb **TN MIPI Camera** **SPI1/SPI3**

OVERLAYS /boot/tegra234-p3767-camera-0044-tevs-dual-overlay.dtbo /boot/tegra234-p3767-0000+p3509-a02-hdr40.dtbo

INITRD /boot/initrd

APPEND \${cbootargs} root=PARTUUID=1f7c61c3-ea0c-4493-ad8e-a2f4cbec9731 rw rootwait  
rootfstype=ext4 mminit\_loglevel=4 console=ttyTCU0,115200 firmware\_class.path=/etc/firmware  
fbcon=map:0 nospectre\_bhb video=efifb:off console=tty0

```

1 TIMEOUT 30
2 DEFAULT lex
3
4 MENU TITLE L4T boot options
5
6 LABEL primary
7     MENU LABEL primary kernel
8     LINUX /boot/Image
9     INITRD /boot/initrd
10    APPEND ${cbootargs} root=PARTUUID=1f7c61c3-ea0c-4493-ad8e-a2f4cbec9731 rw rootwait
    rootfstype=ext4 mminit_loglevel=4 console=ttyTCU0,115200 firmware_class.path=/etc/firmware
    fbcon=map:0 nospectre_bhb video=efifb:off console=tty0
11
12 # When testing a custom kernel, it is recommended that you create a backup of
13 # the original kernel and add a new entry to this file so that the device can
14 # fallback to the original kernel. To do this:
15 #
16 # 1, Make a backup of the original kernel
17 #     sudo cp /boot/Image /boot/Image.backup
18 #
19 # 2, Copy your custom kernel into /boot/Image
20 #
21 # 3, Uncomment below menu setting lines for the original kernel
22 #
23 # 4, Reboot
24
25 # LABEL backup
26 #     MENU LABEL backup kernel
27 #     LINUX /boot/Image.backup
28 #     INITRD /boot/initrd
29 #     APPEND ${cbootargs}
30
31
32
33
34 LABEL lex
35     MENU LABEL lex kernel
36     LINUX /boot/Image
37     FDT /boot/tegra234-p3768-0000+p3767-0000-nv.dtb
38     OVERLAYS /boot/tegra234-p3767-camera-0044-tevs-dual-overlay.dtbo,/boot/tegra234-
    p3767-0000+p3509-a02-hdr40.dtbo
39     INITRD /boot/initrd
40     APPEND ${cbootargs} root=PARTUUID=1f7c61c3-ea0c-4493-ad8e-a2f4cbec9731 rw rootwait
    rootfstype=ext4 mminit_loglevel=4 console=ttyTCU0,115200 firmware_class.path=/etc/firmware
    fbcon=map:0 nospectre_bhb video=efifb:off console=tty0

```

### (3) Install IGC Driver (I226IT, igc.ko)

```

$ sudo cp -v igc.ko /lib/modules/5.15.148-tegra/kernel/drivers/net/ethernet/intel/
$ sudo insmod
/lib/modules/5.15.148-tegra/kernel/drivers/net/ethernet/intel/igc.ko
$ sudo depmod -a

```

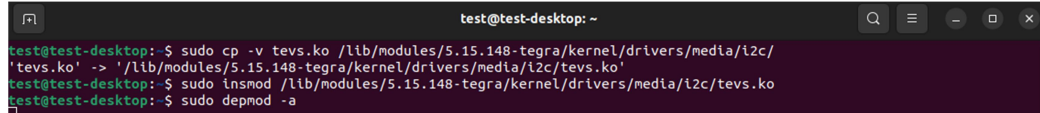
```

test@test-desktop:~$ ls
Desktop  Downloads  jp6.2-2nor03-dtb  Music  Public  Videos
Documents  igc.ko  jp6.2-2nor03-dtb.tar.gz  Pictures  Templates
test@test-desktop:~$ sudo cp -v igc.ko /lib/modules/5.15.148-tegra/kernel/drivers/net/ethernet/intel/
[sudo] password for test:
'igc.ko' -> '/lib/modules/5.15.148-tegra/kernel/drivers/net/ethernet/intel/igc.ko'
test@test-desktop:~$ sudo insmod /lib/modules/5.15.148-tegra/kernel/drivers/net/ethernet/intel/igc.ko
test@test-desktop:~$ sudo depmod -a
test@test-desktop:~$

```

#### (4) Install TEVS Driver(TN MIPI Camera,tevs.ko)

```
#sudo cp -v tevs.ko /lib/modules/5.15.148-tegra/kernel/drivers/media/i2c/
#sudo insmod /lib/modules/5.15.148-tegra/kernel/drivers/media/i2c/tevs.ko
#sudo depmod -a
```

A terminal window titled 'test@test-desktop: ~' showing the execution of three commands: 'sudo cp -v tevs.ko /lib/modules/5.15.148-tegra/kernel/drivers/media/i2c/tevs.ko', 'sudo insmod /lib/modules/5.15.148-tegra/kernel/drivers/media/i2c/tevs.ko', and 'sudo depmod -a'. The output shows the file being copied and the module being loaded successfully.

```
test@test-desktop: ~
test@test-desktop: $ sudo cp -v tevs.ko /lib/modules/5.15.148-tegra/kernel/drivers/media/i2c/
'tevs.ko' -> '/lib/modules/5.15.148-tegra/kernel/drivers/media/i2c/tevs.ko'
test@test-desktop: $ sudo insmod /lib/modules/5.15.148-tegra/kernel/drivers/media/i2c/tevs.ko
test@test-desktop: $ sudo depmod -a
```

#### (5) Config MAC address

Create LAN MAC rules(70-xxxxxxx.rules)

```
#sudo gedit /etc/udev/rules.d/70-persistent-net-rename.rules
```

```
ACTION=="add", SUBSYSTEM=="net", ATTR{address}=="00:a0:c9:00:00:00",
NAME="eth0", RUN+="/sbin/ip link set dev $name address 4c:01:02:03:04:00"
```

Modify MAC Address then save

```
ACTION=="add", SUBSYSTEM=="net", ATTR{address}=="00:a0:c9:00:00:00",
NAME="eth0", RUN+="/sbin/ip link set dev $name address 4c:01:02:03:04:00"
change MAC address
```

Start rules

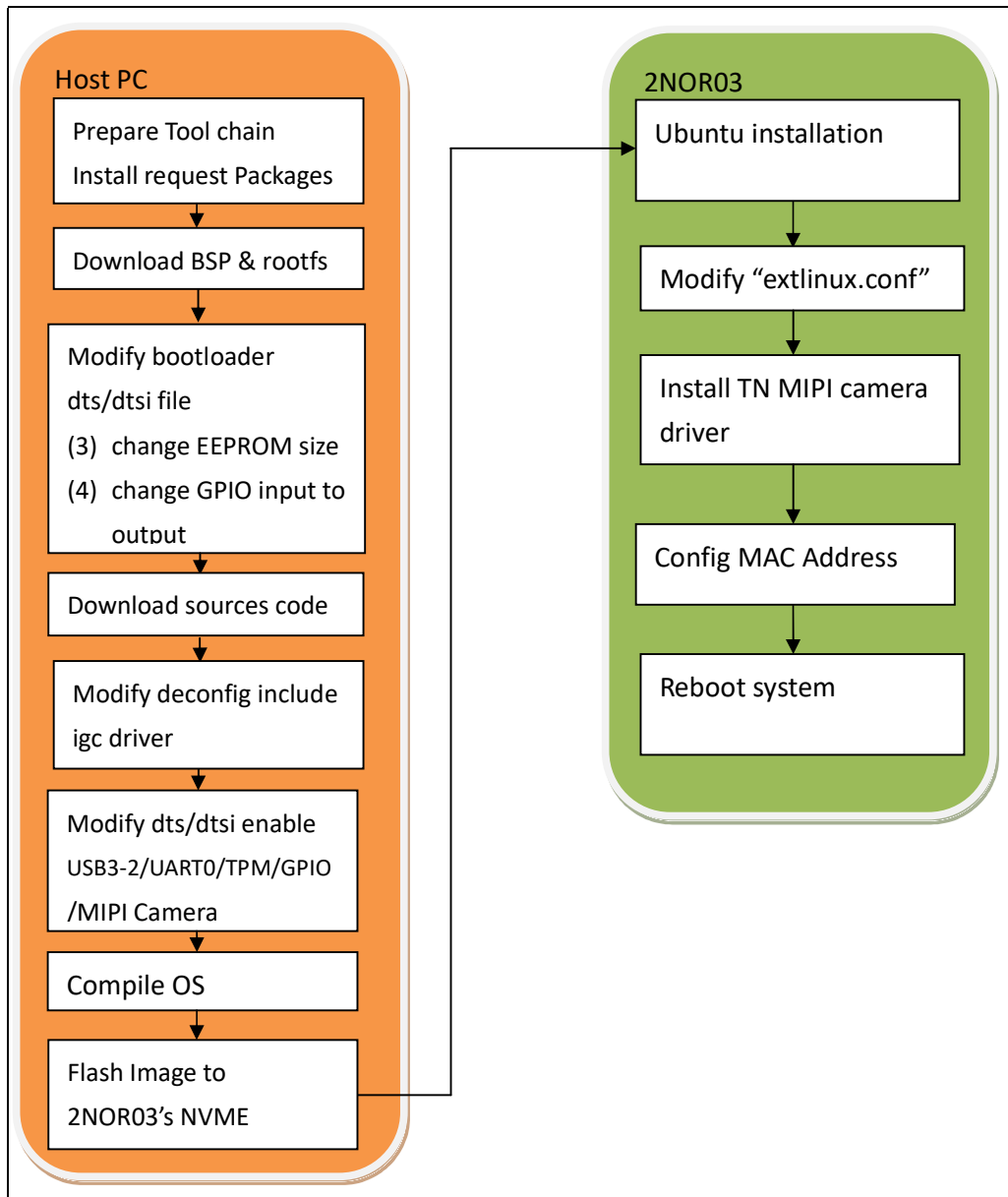
```
#sudo udevadm control --reload-rules
```

```
# sudo udevadm trigger --subsystem-match=net --action=add
```

#### (6) Reboot system

### 3. How to Create 2NOR03 OS - Compile Jetson Linux source code

#### ■ Overview



|                               | 2NOR03                 |
|-------------------------------|------------------------|
| DP1_TX                        | HDMI                   |
| CSI0 (2lenes) + CSI1 (2lenes) | 4 lenes                |
| CSI2 (2lenes) + CSI3 (2lenes) | 4 lenes                |
| GBE_MDI                       | RJ45                   |
| PCIE0 (X4)                    | NGFF1 (NVME)           |
| PCIE1 (X1)                    | I226IT                 |
| PCIE2_0 (X2)                  | NGFF2 (M.2 Key B)      |
| SPI1 1.8V                     | TPM (option)           |
| UART0                         | Pin Header (JP1), 3.3v |

|               |                                     |
|---------------|-------------------------------------|
| UART1         | RS232(default) , RS422(bon control) |
| UART2 (debug) | Pin Header, 3.3v                    |
| I2C0 3.3V     | Type C and Wafer, 3.3v              |
| I2C1 3.3V     | Wafer ,3.3v                         |
| USBSS0 + USB1 | USB TypeA (CU3)                     |
| USBSS1 + USB0 | USB TypeC (CU1)                     |
| USBSS2 + USB2 | USB TypeA (CU2)                     |
| FAN PWM       | Wafer                               |
| GPIO01        | Pin Header (JP1)                    |
| GPIO07        | Pin Header (JP1)                    |
| GPIO11        | Pin Header (JP1)                    |
| GPIO12        | Pin Header (JP1)                    |
| GPIO13        | Pin Header (JP1)                    |

### 3.1. Host computer Install and download necessary files

*The OS version use by LEX is Ubuntu 22.04.6 LTS*

#### (1) Install request packages

```
#sudo apt install wget lbzip2 build-essential bc zip libgmp-dev libmpfr-dev
libmpc-dev vim-common flex bison libssl-dev -y
#sudo apt-get install nfs-kernel-server zstd abootimg sshpass qemu-user-static sudo
apt-get install libxml2-utils -y
```

#### (2) Install GCC toolchain

<https://developer.nvidia.com/embedded/jetson-linux-r3643>

Download "Bootlin Toolchain gcc 11.3"

|       |                                             |
|-------|---------------------------------------------|
| TOOLS | WebRTC                                      |
|       | <b>Bootlin</b> Toolchain gcc 11.3           |
|       | <b>Bootlin</b> Toolchain Sources, 2022.08-1 |

```
# mkdir $HOME/l4t-gcc
# cp -v aarch64--glibc--stable-2022.08-1.tar.bz2 $HOME/l4t-gcc
# cd $HOME/l4t-gcc
# tar xf aarch64--glibc--stable-2022.08-1.tar.bz2
```

### (3) Set environment

```
# export
CROSS_COMPILE=$HOME/l4t-gcc/aarch64--glibc--stable-2022.08-1/bin/aarch64-buildroot-li
nux-gnu-
# export CROSS_COMPILE_AARCH64_PATH=$HOME/l4t-gcc
# export
CROSS_COMPILE_AARCH64=$HOME/l4t-gcc/aarch64--glibc--stable-2022.08-1/bin/aarch64-
buildroot-linux-gnu-
# export LOCALVERSION=-tegra
```

### 3.2. Download Driver Package (BSP) and Sample Root Filesystem (rootfs)

refer to "2.1. Download Driver Package(BSP) & Sample Root Filesystem (rootfs)"

### 3.3. Modify bootloader dts/dtsi file

refer to "2.2 Integrate modifications from the 2NOR03 into the bootloader"

#### (1) Set the EEPROM size to 0

#### (2) Set the GPIO from input to output (depending on customer needs)

### 3.4. Download Jetson-Linux JP6.2 source

Downlaod source

```
cd $HOME/Linux_for_Tegra/source
```

Modify File "source\_sync.sh"

```
-- DownloadAndSync "$WHAT" "${LDK_DIR}/${WHAT}" "git://$REPO" "${TAG}"
"${OPT}"
++ DownloadAndSync "$WHAT" "${LDK_DIR}/${WHAT}" "https://$REPO" "${TAG}"
"${OPT}"
```

Start download source

```
# ./source_sync.sh -k -t jetson_36.4.3
```

### 3.5. Enable IGC Driver (CONFIG\_IGC=y)

```
#cd $HOME/Linux_for_Tegra/source/kernel/kernel-jammy-src/arch/arm64/configs/

# gedit defconfig
CONFIG_IGC=y
```

### 3.6. Enable USB3-2 and UART0

Modify "tegra234-p3768-0000.dtsi"

```
# gedit
$HOME/Linux_for_Tegra/source/hardware/nvidia/t23x/nv-public/tegra234-p3768-0000.dts
i
/{
    aliases {
-        //serial0 = &tcu;
+        serial0 = "/bus@0/serial@3110000";
    };

    padctl@3520000 {
        status = "okay";
        usb3 {
            lanes {
                //lex++
+                usb3-2 {
+                    nvidia,function = "xusb";
+                    status = "okay";
+                };
            };
        };
        ports {
            //lex++
+            usb3-2 {
+                nvidia,usb2-companion = <2>;
+                status = "okay";
+            };
        };
    };

    usb@3610000 {
        status = "okay";

        phys = <&{/bus@0/padctl@3520000/pads/usb2/lanes/usb2-0}>,
              <&{/bus@0/padctl@3520000/pads/usb2/lanes/usb2-1}>,
              <&{/bus@0/padctl@3520000/pads/usb2/lanes/usb2-2}>,
              <&{/bus@0/padctl@3520000/pads/usb3/lanes/usb3-0}>,
```

```

        <&{/bus@0/padctl@3520000/pads/usb3/lanes/usb3-1}>,
+        <&{/bus@0/padctl@3520000/pads/usb3/lanes/usb3-2}>;
//lex++ usb3-2
        phy-names = "usb2-0", "usb2-1", "usb2-2", "usb3-0",
+        "usb3-1", "usb3-2";    //lex++ usb3-2
    };
};

```

### 3.7. Enable TPM and UART0

Modify

“\$HOME/Linux\_for\_Tegra/source/hardware/nvidia/t23x/nv-public/nv-platform/tegra234-p3768-0000+p3767-xxxx-nv-common.dtsi”

```

/{
+
+    //lex++
+    serial@3110000 {
+        compatible = "nvidia,tegra194-hsuart";
+        reset-names = "serial";
+        status = "okay";
+    };
+
+    spi@3230000{
+        status = "okay";
+    //lex--
+    spi@0 {
+        compatible = "tegra-spidev";
+        reg = <0x0>;
+        spi-max-frequency = <50000000>;
+        controller-data {
+            nvidia,enable-hw-based-cs;
+            nvidia,rx-clk-tap-delay = <0x10>;
+            nvidia,tx-clk-tap-delay = <0x0>;
+        };
+    };
+    spi@1 {
+        compatible = "tegra-spidev";

```

```

-         reg = <0x1>;
-         spi-max-frequency = <50000000>;
-         controller-data {
-             nvidia,enable-hw-based-cs;
-             nvidia,rx-clk-tap-delay = <0x10>;
-             nvidia,tx-clk-tap-delay = <0x0>;
-         };
-     };
//lex ++
+         nvidia,clock-always-on;

+     slb9670: slb9670@0 {
+         compatible = "infineon,slb9670";
+         reg = <0x1>;
+         spi-max-frequency = <33000000>;
+         status = "okay";
+         controller-data {
+             nvidia,enable-hw-based-cs;
+             nvidia,rx-clk-tap-delay = <0x10>;
+         };
+     };
};

```

### 3.8. Modify SPI1/SPI3 dts

```

#gedit
/Linux_for_Tegra/source/hardware/nvidia/t23x/nv-public/overlay/tegra234-p3767-0
000-common-hdr40.dtsi
/{
    /* lex-- mark
        hdr40-pin8 {
            nvidia,pins = "uart1_tx_pr2";
        };
        hdr40-pin10 {
            nvidia,pins = "uart1_rx_pr3";
        };
        hdr40-pin12 {
            nvidia,pins = "soc_gpio41_ph7";

```

```

        nvidia,function = "i2s2";
        nvidia,pin-label = "i2s2_sclk";
        nvidia,tristate = <TEGRA_PIN_DISABLE>;
        nvidia,enable-input = <TEGRA_PIN_ENABLE>;
    };
    hdr40-pin15 {
        nvidia,pins = "soc_gpio39_pn1";
        nvidia,function = "gp";
        nvidia,pin-group = "pwm1";
        nvidia,tristate = <TEGRA_PIN_DISABLE>;
        nvidia,enable-input = <TEGRA_PIN_DISABLE>;
    };
    hdr40-pin29 {
        nvidia,pins = "soc_gpio32_pq5";
        nvidia,function = "extperiph3";
        nvidia,pin-group = "extperiph3_clk";
        nvidia,pull = <TEGRA_PIN_PULL_NONE>;
        nvidia,tristate = <TEGRA_PIN_DISABLE>;
        nvidia,enable-input = <TEGRA_PIN_DISABLE>;
        nvidia,io-high-voltage = <TEGRA_PIN_DISABLE>;
        nvidia,lpdr = <TEGRA_PIN_DISABLE>;
    };
    hdr40-pin31 {
        nvidia,pins = "soc_gpio33_pq6";
        nvidia,function = "extperiph4";
        nvidia,pin-group = "extperiph4_clk";
        nvidia,pull = <TEGRA_PIN_PULL_NONE>;
        nvidia,tristate = <TEGRA_PIN_DISABLE>;
        nvidia,enable-input = <TEGRA_PIN_DISABLE>;
        nvidia,io-high-voltage = <TEGRA_PIN_DISABLE>;
        nvidia,lpdr = <TEGRA_PIN_DISABLE>;
    };
    hdr40-pin32 {
        nvidia,pins = "soc_gpio19_pg6";
        nvidia,function = "gp";
        nvidia,pin-group = "pwm7";
        nvidia,tristate = <TEGRA_PIN_DISABLE>;
        nvidia,enable-input = <TEGRA_PIN_DISABLE>;
    };

```

```

    };
    hdr40-pin33 {
        nvidia,pins = "soc_gpio21_ph0";
        nvidia,function = "gp";
        nvidia,pin-group = "pwm5";
        nvidia,tristate = <TEGRA_PIN_DISABLE>;
        nvidia,enable-input = <TEGRA_PIN_DISABLE>;
    };
    hdr40-pin35 {
        nvidia,pins = "soc_gpio44_pi2";
        nvidia,function = "i2s2";
        nvidia,pin-label = "i2s2_fs";
        nvidia,tristate = <TEGRA_PIN_DISABLE>;
        nvidia,enable-input = <TEGRA_PIN_ENABLE>;
    };
    hdr40-pin38 {
        nvidia,pins = "soc_gpio43_pi1";
        nvidia,function = "i2s2";
        nvidia,pin-label = "i2s2_din";
        nvidia,tristate = <TEGRA_PIN_ENABLE>;
        nvidia,enable-input = <TEGRA_PIN_ENABLE>;
    };
    hdr40-pin40 {
        nvidia,pins = "soc_gpio42_pi0";
        nvidia,function = "i2s2";
        nvidia,pin-label = "i2s2_dout";
        nvidia,tristate = <TEGRA_PIN_DISABLE>;
        nvidia,enable-input = <TEGRA_PIN_DISABLE>;
    };
    */
fragment@1 {
    __overlay__ {
        jetson_io_pinmux_aon: exp-header-pinmux {
            /* lex-- mark
                hdr40-pin3 {
                    nvidia,pins = "gen8_i2c_sda_pdd2";
                    nvidia,pin-label = "i2c8";
                };
            */

```

```

        hdr40-pin5 {
            nvidia,pins = "gen8_i2c_scl_pdd1";
            nvidia,pin-label = "i2c8";
        };
        hdr40-pin27 {
            nvidia,pins = "gen2_i2c_sda_pdd0";
        };
        hdr40-pin28 {
            nvidia,pins = "gen2_i2c_scl_pcc7";
        };
        */
};
};
};
};

```

### 3.9. Setting MIPI Camera

```

#cp -v tegra234-p3767-camera-0044-tevs-dual-overlay.dts
$HOME/Linux_for_Tegra/souce/hardware/nvidia/t23x/nv-public/overlay

#gedit
$HOME/Linux_for_Tegra/souce/hardware/nvidia/t23x/nv-public/overlay/Makefile

+ dtbo-y += tegra234-p3767-camera-0044-tevs-dual-overlay.dtbo

```

```

jimmy@jimmy-System-Product-Name:/media/jimmy/nvidia-hdd/jimmy$ gedit /media/jimmy/nvidia-hdd/jimmy/customer/Linux_for_Te
gra/source/hardware/nvidia/t23x/nv-public/overlay/Makefile

dtbo-y += tegra234-aie900a-camera-vls-gm2-fsync-overlay.dtbo
dtbo-y += tegra234-eac5000-camera-vls-gm2-overlay.dtbo
dtbo-y += tegra234-eac5000-camera-vls-gm2-fsync-overlay.dtbo
dtbo-y += tegra234-p3767-camera-0044-tevs-dual-overlay.dtbo
dtbo-y += tegra234-p3767-camera-0024-tevs-dual-overlay.dtbo

```

### 3.10. Setting MIPI Camera driver

```

#copy -av technexion/ $HOME/Linux_for_Tegra/source/nvidia-oot/drivers/media/i2c
$ cp -av technexion/ /media/jimmy/nvidia-hdd/jimmy/customer/Linux_for_Tegra/source/nvidia-oot/drivers/media/i2c

#gedit $HOME/Linux_for_Tegra/souce/nvidia-oot/drivers/media/i2c/Makefile

```

```
+ obj-m += technexion/
```

```
jimmy@jimmy-System-Product-Name:/media/jimmy/nvidia-hdd/jimmy/customer/Linux_for_Tegra/source/nvidia-oot/drivers/media/i2c$ gedit Makefile
```

```
obj-m += pca9570.o  
obj-m += virtual_i2c_mux.o  
endif  
obj-m += technexion/
```

### 3.11. Compile Image

```
#cd $HOME/Linux_for_Tegra/source/  
#./nvbuild.sh -o $HOME/Linux_for_Tegra/images
```

### 3.12. Copy Image and dtb

#### Copy Image

```
#cd $HOME/Linux_for_Tegra/source/  
#cp -v images/kernel/kernel-jammy-src/arch/arm64/boot/Image  
$HOME/Linux_for_Tegra/kernel
```

#### Copy dtb

```
#cp -v images/kernel-devicetree/generic-dts/dtbs/*.* $HOME/Linux_for_Tegra/kernel/dtb/  
#sudo cp -v images/kernel-devicetree/generic-dts/dtbs/*.*  
$HOME/Linux_for_Tegra/rootfs/boot
```

#### Copy MIPI driver

```
#cd  
$HOME/Linux_for_Tegra/source/images/nvidia-oot/drivers/media/i2c/technexion  
#sudo cp -v tevs.ko $HOME
```

### 3.13. Flash Image to NVME

refer to “2.3. Flash Image to 2NOR03’s NVME”

### 3.14. Setup 2NOR03

After completing the Ubuntu installation, proceed with configuring the system settings

#### (1). Modify “extlinux.conf”

refer to “2.4. Setup 2NOR03

(2) Modify "extlinux.conf"

(2).Install TN MIPI Camera driver

refer to "2.4. Setup 2NOR03

(4) Install MIPI Camera driver(tevs.ko)

(3).Config MAC Address

refer to "2.4. Setup 2NOR03

(5) Config MAC Address

(4).Reboot System

refer to "2.4. Setup 2NOR03

(6) Reboot system