LEX System

2NOR02 Porting Guide

NVIDIA Orin Jetson Linux 36.4.3(JetPack6.2)

LEX SW Team

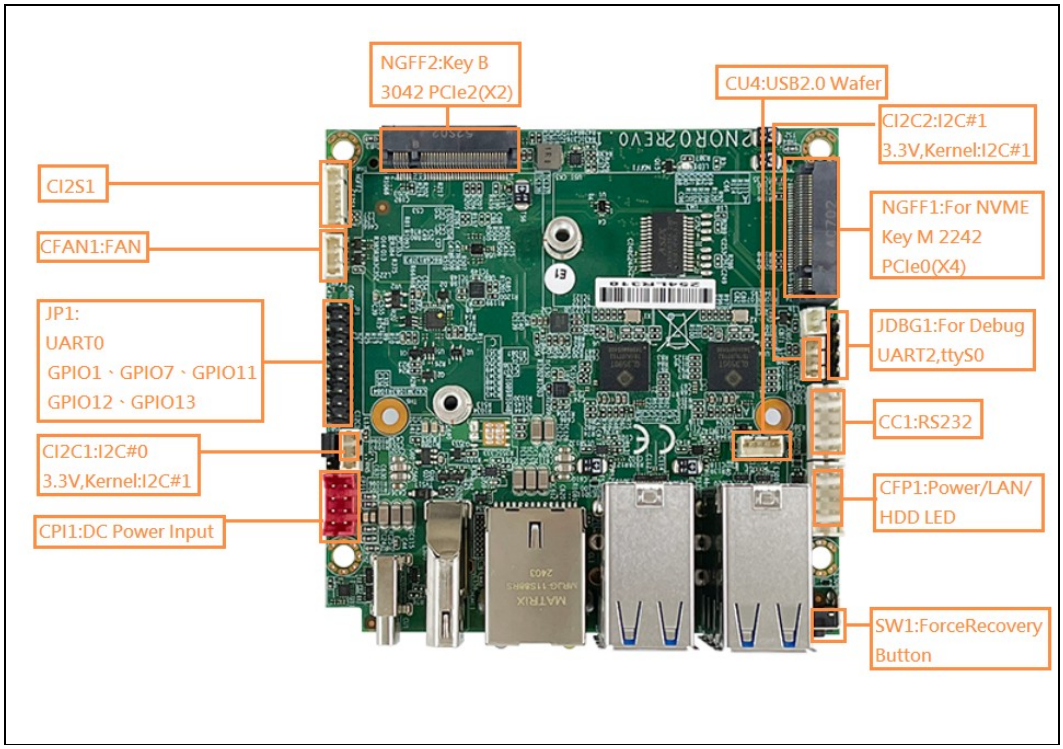
2025/5/12

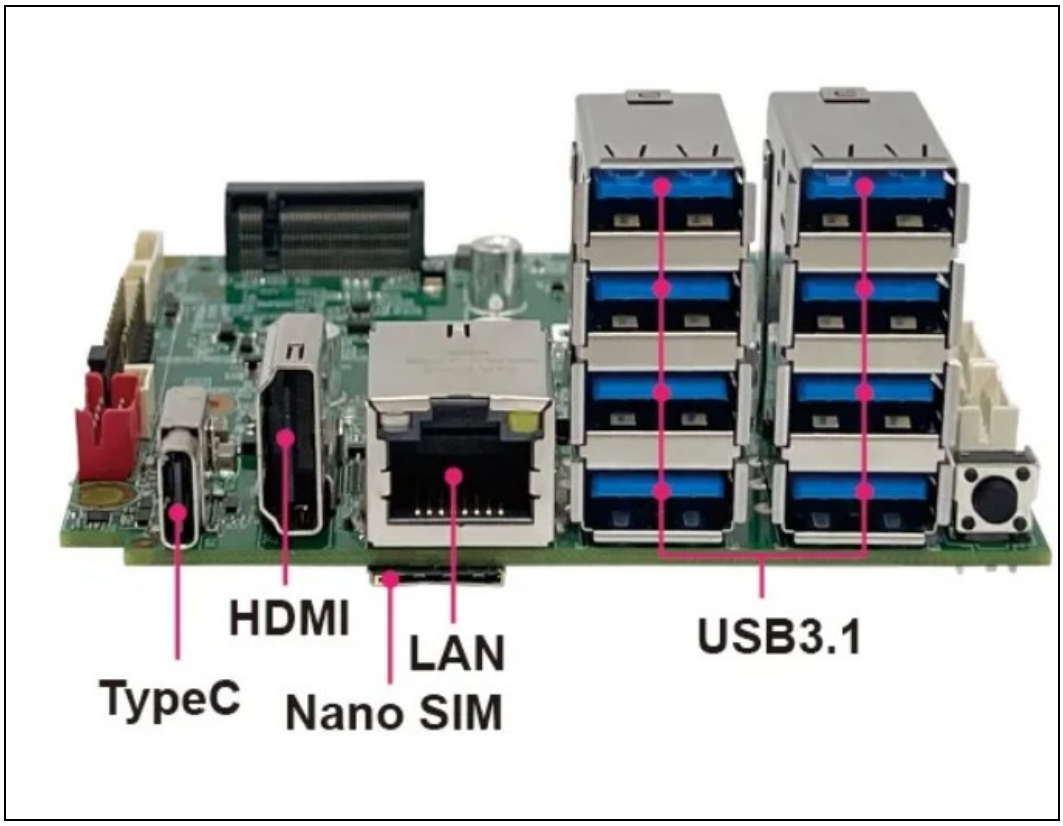
History

Version	Date	Description
1.0	2025/05/12	Create File

Table of Contents

1. Overview



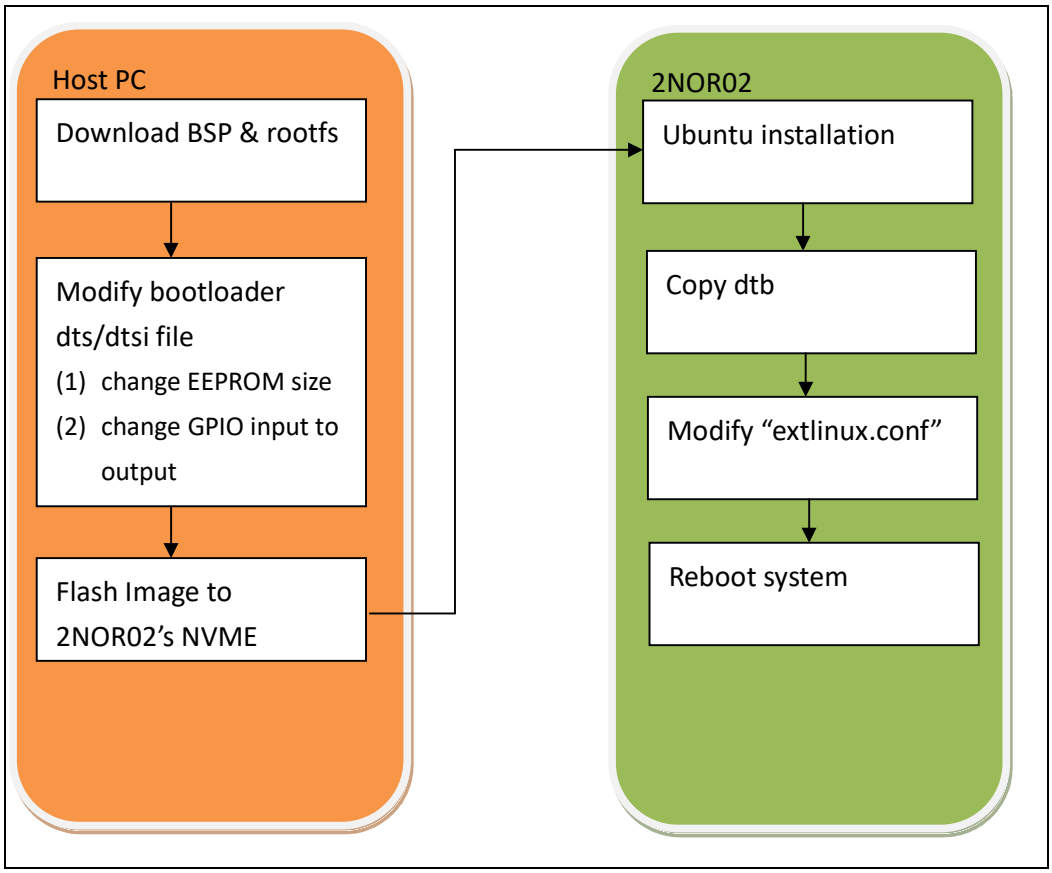


Jetson Orin Nano 8G	tegra234-p3768-0000+p3767-0003-nv.dtb
Jetson NX 8G	tegra234-p3768-0000+p3767-0001-nv.dtb
Jetson NX 16G	tegra234-p3768-0000+p3767-0000-nv.dtb

The NVIDIA Jetson Orin dev kit includes an EEPROM, whereas the 2NOR02 does not. Therefore, please refrain from installing the operating system via the SDK Manager. However, you may install DeepStream through the SDK Manager.

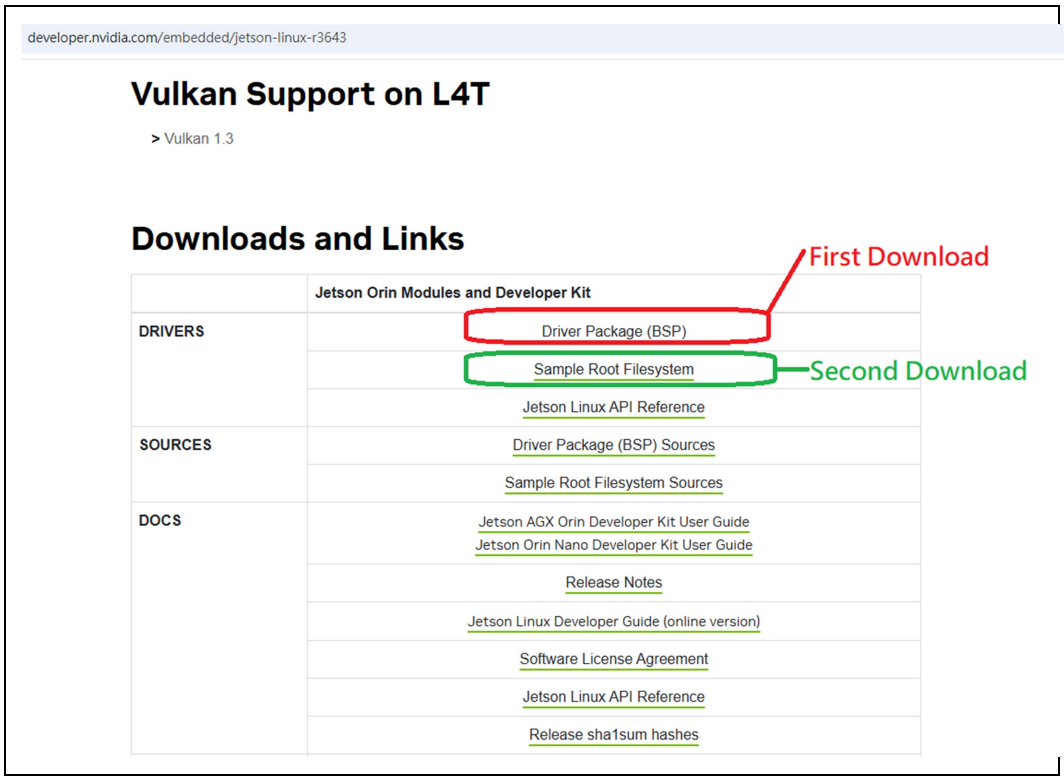
The following describes method for 2NOR02 on NVIDIA Jetson Orin OS

2. How to Create 2NOR02 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



(1) Uncompress Driver Package (BSP)

For example：

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

```
jimmy@jimmy-System-Product-Name:/media/jimmy/nvidia-hdd/jimmy$ mkdir lex
jimmy@jimmy-System-Product-Name:/media/jimmy/nvidia-hdd/jimmy$ tar -jxf Jetson_Linux_R36.4.3_aarch64.tbz2 -C lex/
jimmy@jimmy-System-Product-Name:/media/jimmy/nvidia-hdd/jimmy$ cd lex
jimmy@jimmy-System-Product-Name:/media/jimmy/nvidia-hdd/jimmy/lex$ ls
Linux_for_Tegra
jimmy@jimmy-System-Product-Name:/media/jimmy/nvidia-hdd/jimmy/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 四  30 09:51 .
drwxrwxr-x  10 jimmy jimmy  4096 一   8 09:56 ..
lrwxrwxrwx   1 root  root      7 二  18  2023 bin -> usr/bin
drwxr-xr-x   2 root  root   4096 四  18  2022 boot
drwxr-xr-x   2 root  root   4096 二  18  2023 dev
drwxr-xr-x 137 root  root  12288 十二  4 14:31 etc
drwxr-xr-x   2 root  root   4096 四  18  2022 home
lrwxrwxrwx   1 root  root      7 二  18  2023 lib -> usr/lib
drwxr-xr-x   2 root  root   4096 二  18  2023 media
drwxr-xr-x   2 root  root   4096 二  18  2023 mnt
drwxr-xr-x   2 root  root   4096 二  18  2023 opt
drwxr-xr-x   2 root  root   4096 四  18  2022 proc
-rw-rw-r--   1 jimmy jimmy    62 一   8 09:42 README.txt
drwx------   3 root  root   4096 三  22  2023 root
drwxr-xr-x  19 root  root   4096 十  17  2024 run
lrwxrwxrwx   1 root  root      8 二  18  2023 sbin -> usr/sbin
drwxr-xr-x   2 root  root   4096 十二  1  2022 snap
drwxr-xr-x   2 root  root   4096 二  18  2023 srv
drwxr-xr-x   2 root  root   4096 四  18  2022 sys
drwxrwxrwt   2 root  root   4096 十二  4 14:30 tmp
drwxr-xr-x  11 root  root   4096 二  18  2023 usr
drwxr-xr-x  14 root  root   4096 三  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 2NOR02 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 2NOR02 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$
```

```
/dts-v1/;

#include "tegra234-mb2-bct-common.dtsi"

/ {
        mb2-misc {
                eeprom {
                        cvm_eeprom_i2c_instance = <0>;
                        cvm_eeprom_i2c_slave_address = <0xa0>;
                        cvm_eeprom_read_size = <0x100>;
                        cvb_eeprom_i2c_instance = <0x0>;
                        cvb_eeprom_i2c_slave_address = <0xae>;
                        //cvb_eeprom_read_size = <0x100>;
                        cvb_eeprom_read_size = <0x0>;
                };
        };
};
```

(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 2NOR02's NVME

(1) Switch 2NOR02 into Force Recovery Mode

Connect your Linux host computer to the appropriate USB port on your 2NOR02 (USB typeC)

1. Ensure that the 2NOR02 is powered off.
2. Press and hold down the Force Recovery button (SW1).
3. 2NOR02 Power on.
4. Release the Force Recovery button (SW1)

To Determine Whether the 2NOR02 Is in Force Recovery Mode

Open a terminal window on your host computer and enter command" lsusb" .

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
```

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

2.4. Setup 2NOR02

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

(1) Copy 2NOR02's dtb fils

copy files to /boot

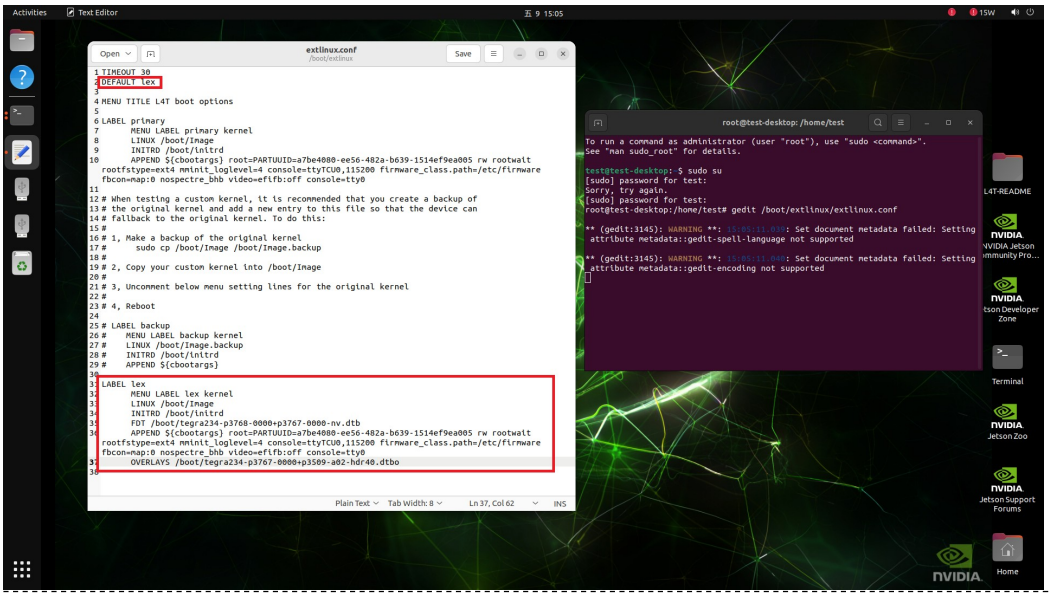
Jetson Orin Nano 8G	tegra234-p3768-0000+p3767-0003-nv.dtb
Jetson NX 8G	tegra234-p3768-0000+p3767-0001-nv.dtb
Jetson NX 16G	tegra234-p3768-0000+p3767-0000-nv.dtb

OVERLAYS , for SPI1/SPI3	tegra234-p3767-0000+p3509-a02-hdr40.dtbo

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

Create new LABEL “lex” and set to default

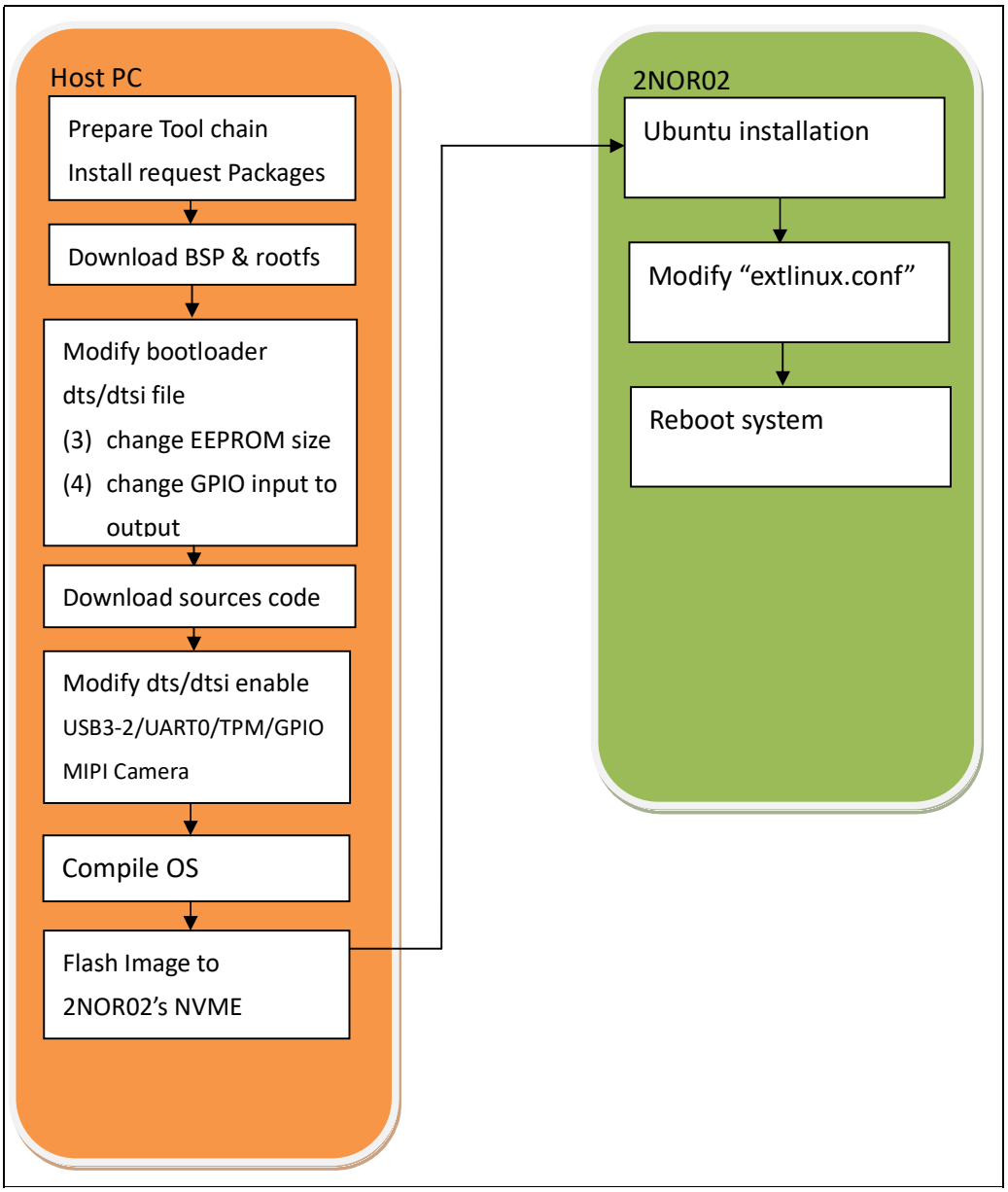
```
DEFAULT lex

LABEL lex
        MENU LABEL lex kernel
        LINUX /boot/Image
        INITRD /boot/initrd
         FDT /boot/tegra234-p3768-0000+p3767-0000-nv.dtb
         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
         OVERLAYS /boot/tegra234-p3767-0000+p3509-a02-hdr40.dtbo
```



(3) Reboot system

3. How to Create 2NOR02 OS - Compile Jetson Linux source code

■ Overview



	2NOR02
DP1_TX	HDMI
GBE_MDI	RJ45
PCIE0 (X4)	NGFF1 (NVME)
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 (CU2) 4port USB3
USBSS1 + USB0	USB TypeC (CU1)
USBSS2 + USB2	USB TypeA (CU3) 4port USB3
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

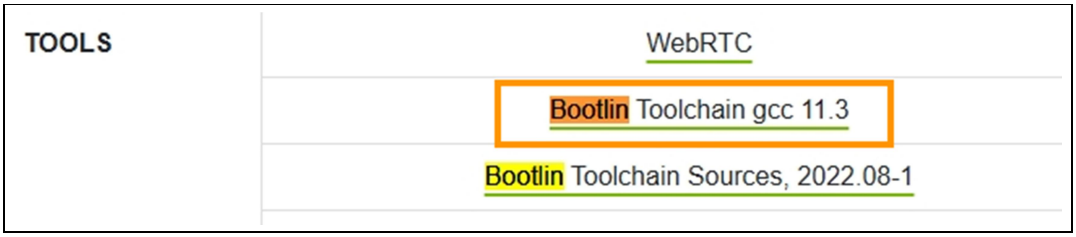
(1) Install request packages

```
#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”




```
# 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=$HOME/l4t-gcc/aarch64--glibc--stable-2022.08-1/bin/aarch64-buildroot-linux-gnu-
# export CROSS_COMPILE_AARCH64_PATH=$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 2NOR02 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 USB3-2 and UART0

Modify “tegra234-p3768-0000.dtsi”

```
# gedit $HOME/Linux_for_Tegra/source/hardware/nvidia/t23x/nv-public/tegra234-p3768-0000.dtsi
        /{
                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
            };
```

```
};
```

3.6. 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>;
+                               };
+                       };
               };
};
```

3.7. Compile Image

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

3.8. 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
```

3.9. Flash Image to NVME

refer to “2.3. Flash Image to 2NOR02’s NVME”

3.10. Setup 2NOR02

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

(1). Modify “extlinux.conf”

refer to “2.4. Setup 2NOR02

(2) Modify “extlinux.conf”

(2). Reboot System