THSCJ101 V4L2 Driver Integration Manual

Table of contents

1. Overview

This document explains how to integrate the THSCJ101 V4L2 driver into the Jetson Linux and create the Linux binary. There are the following three steps to integrate the V4L2 driver and create the Linux binary.

Step 1. Flash the Linux for Jetson Orin NX from NVIDIA

This section explains the way to flash the Linux for Jetson Orin NX that NVIDIA released. You can confirm that your Jetson Orin NX carrier board (JNX42) is ready to develop.

We use the following JNX42 from Auvidea as the Jetson Orin NX carrier board in this document.
https://auvidea.eu/product/jnx42-70784/

Step 2. Create the Linux Development Environment

This section explains the way to create the Linux development environment. You will create the development environment from the source codes of Linux released from Auvidea and NVIDIA. Then, you will build the Linux binaries for Jetson Orin NX in this environment. The development environment to create your custom Linux binary is ready.

Step 3. Integrate THSCJ101 V4L2 Driver to the Jetson Linux

This section explains the way to add the THSCJ101 V4L2 driver to the Linux development environment. You can get the THSCJ101 V4L2 driver source code as the patch file from THine Solutions, Inc. and the latest THSCJ101 firmware binary required to use the firmware update V4L2 command. You will build the Linux binaries with the files get from THine on the environment created in step 2. You are ready to connect THSCJ101 to the Jetson Orin NX carrier board.

You are ready to develop Jetson Linux with THSCJ101 for your camera system after you complete these three steps above.

2. Flash the Linux for Jetson Orin NX from NVIDIA

- On the Linux PC

2.1 Download the JetPack 5.1 from the NVIDIA and extract the downloaded files.

1) Get the following files from the NVIDEA web page, https://developer.nvidia.com/embedded/jetpack-archive.
   - ✓ Jetson_Linux_R35.2.1_aarch64.tbz2
   - ✓ Tegra_Linux_Sample-Root-Filesystem_R35.2.1_aarch64.tbz2
2) Extract the files with following commands.

```
$ cd <working directory>
$ cp <download_directory>Jetson_Linux_R35.2.1_aarch64.tbz2 .
$ tar xf Jetson_Linux_R35.2.1_aarch64.tbz2
$ cp <download_directory>Tegra_Linux_Sample-Root-Filesystem_R35.2.1_aarch64.tbz2 .
$ cd <working directory>/Linux_for_Tegra/rootfs/
$ sudo tar xpf ../../Tegra_Linux_Sample-Root-Filesystem_R35.2.1_aarch64.tbz2
$ cd ../../Linux_for_Tegra
```

2.2 Prepare prerequired files.

1) Install the flashing prerequisites the following command.

```
$ sudo ./tools/l4t_flash_prerequisites.sh
```

2) Apply the binaries with the following command.

```
$ sudo ./apply_binaries.sh
```

3) Adjust the EEPROM size by changing the "cvb_eeprom_read_size = <0x100>" in "Linux_for_Tegra/bootloader/t186ref/BCT/tegra234-mb2-bct-misc-p3767-0000.dts" to "cvb_eeprom_read_size = <0x0>".

2.3 Connect the Jetson Orin NX carrier board and Linux PC.

1) Connect the Jetson Orin NX carrier board to USB3.0 port of host Linux PC by USB2.0 Cable.
2) Connect keyboard and mouse to the Jetson Orin NX carrier board.
3) Plug the AC adaptor to power on the Jetson Orin NX carrier board.
4) Check if the Linux PC recognizes the Jetson Orin NX carrier board with following command.

```
$ lsusb
```

The Linux PC recognizes the Jetson Orin NX carrier board if the command response shows the entry with "Nvidia Corp.".

2.4 Flash the Linux binary to the Jetson Orin NX carrier board.

1) Flash the Linux binary with the following command.

```
$ 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
```

The Jetson Orin NX reboots after the flashing process completes.

- On the Jetson Orin NX carrier board

2.5 Check if the Jetson Orin NX carrier board displays the Ubuntu desktop on the monitor.

1) Connect the HDMI monitor to the Jetson Orin NX carrier board.
2) Check if the Jetson Orin NX carrier board displays the Ubuntu desktop on the monitor. You succeeded create the Linux development environment if the monitor connected with the Jetson Orin NX carrier board displays the Ubuntu desktop.

2.6 Set Ubuntu settings on Ubuntu desktop and then power off.

3. Create the Linux Development Environment

- On the Linux PC

3.1 Download the following file and extract.

1) Download the following file by selecting JNXxx of firmware for Jetpack 5.1 under "Firmware for Jetson Orin Nano and Orin NX based carrier boards" from the Auvidea web page, https://auvidea.eu/firmware/.
   - ✓ Jetpack_5_1_Orin_NX_JNX42_rev1.tar.xz
2) Extract Jetpack_5_1_Orin_NX_JNX42_rev1.tar.xz with the following commands.

```
$ cd  <working directory>
$ mkdir jnx42_src
$ cd jnx42_src
$ cp <download_directory>Jetpack_5_1_Orin_NX_JNX42_rev1.tar.xz .
$ tar -xf  Jetpack_5_1_Orin_NX_JNX42_rev1.tar.xz
```

You will get following files.

- ✓ How_to_flash_Orin NX_NANO.txt
- ✓ kernel_out.tar.xz
- ✓ kernel_src.tar.xz

3.2 Extract and copy the files into build directory.

1) Extract kernel_src.tar.xz with the following command.

```
$ tar -xf kernel_src.tar.xz
```

2) Copy extracted kernel_src directory as sources directory into Linux_for_Tegra directory made at the 2) of 2.1 with the following command.

```
$ cp -r kernel_src <working directory>/sources
```

3.3 Set environment variables.

1) Set environment variables with the following commands. The commands assume that you will install gcc in the directory, $HOME/jetson/l4t-gcc in the next 3.4.

```
$ export DEVDIR=<working directory>/Linux_for_Tegra
$ mkdir -p $DEVDIR/images/dtb
$ mkdir -p $DEVDIR/images/modules
$ export TOOLCHAIN_SRC=bootlin-toolchain-gcc-93
$ export TOOLCHAIN_DIR=gcc-9.3-glibc-2.31
$ export KERNEL_DIR=$DEVDIR/sources/kernel/kernel-5.10
$ export CROSS_COMPILE=$HOME/jetson/l4t-gcc/bin/aarch64-buildroot-linux-gnu-
$ export TEGRA_KERNEL_OUT=$DEVDIR/images
$ export KERNEL_MODULES_OUT=$DEVDIR/images/modules
$ export ARCH=arm64
```

3.4 Install gcc.

1) Install gcc with the following commands.

```
$ mkdir $HOME/jetson
$ cd $HOME/jetson
$ mkdir -p $HOME/jetson/l4t-gcc
$ cd $HOME/jetson/l4t-gccexport DEVDIR=<working directory>/Linux_for_Tegra
```

Create the following shell script and execute the shell script.

```
if ! test -e ${TOOLCHAIN_SRC}.tar.gz;then
wget -O ${TOOLCHAIN_SRC}.tar.gz https://developer.nvidia.com/embedded/jetson-linux/bootlin-toolchain-gcc-93
tar -xf ${TOOLCHAIN_SRC}.tar.gz
fi
```

3.5 Build Kernel and device tree.

1) Build kernel with the following commands.

```
$ cd $KERNEL_DIR
$ make O=$TEGRA_KERNEL_OUT CROSS_COMPILE=${CROSS_COMPILE} tegra_defconfig
$ make O=$TEGRA_KERNEL_OUT CROSS_COMPILE=${CROSS_COMPILE} -j $(nproc) Image
```

2) Build device tree with the following command.

```
$ make O=$TEGRA_KERNEL_OUT CROSS_COMPILE=${CROSS_COMPILE} -j $(nproc) dtbs
```

3) Build module with the following commands.

```
$ make O=$TEGRA_KERNEL_OUT CROSS_COMPILE=${CROSS_COMPILE} -j $(nproc) modules
$ make ARCH=arm64 O=$TEGRA_KERNEL_OUT LOCALVERSION=-tegra
INSTALL_MOD_PATH=$KERNEL_MODULES_OUT -j4 modules_install
```

3.6 Download the display kernel source code from NVIDA and build display kernel module.

1) Download the following file from the NVIDEA web page, https://developer.nvidia.com/embedded/jetson-linux-r3521.
You can download the file by clicking "Driver Package (BSP) Sources" on the above web page.

✓ public_sources.tbz2

2) Extract the files with the following commands.

```
$ mkdir <display_kernel_working directory>
$ cd <display_kernel_working directory>
$ cp <download_directory>public_sources.tbz2 .
$ tar -xjf public_sources.tbz2

$ cd Linux_for_Tegra/source/public
$ tar -xjf kernel_src.tbz2

$ tar xjf  nvidia_kernel_display_driver_source.tbz2
$ cd NVIDIA-kernel-module-source-TempVersion
```

2) Set environment variables with the following commands.

```
$ export CROSS_COMPILE_AARCH64=$HOME/jetson/l4t-gcc/bin/aarch64-buildroot-linux-gnu-
$ export LOCALVERSION="-tegra"
```

3) Build the display kernel module with the following commands.

```
$ make \
    modules \
    SYSSRC=$DEVDIR/sources/kernel/kernel-5.10 \
    SYSOUT=$DEVDIR/images \
    CC=${CROSS_COMPILE_AARCH64}gcc \
    LD=${CROSS_COMPILE_AARCH64}ld.bfd \
    AR=${CROSS_COMPILE_AARCH64}ar \
    CXX=${CROSS_COMPILE_AARCH64}g++ \
    OBJCOPY=${CROSS_COMPILE_AARCH64}objcopy \
    TARGET_ARCH=aarch64 \
    ARCH=arm64
```

Check if there are following display kernel module files in the directory, “<display_kernel_working directory>/Linux_for_Tegra/source/public/NVIDIA-kernel-module-source-TempVersion/kernel-open/”.

✓ nvidia-drm.ko
✓ nvidia-modeset.ko
✓ nvidia.ko

4) Reset environment variables with the following commands.

```
export -n CROSS_COMPILE_AARCH64
export -n LOCALVERSION
```

3.7 Copy built files to the Jetson Orin NX carrier board.

1) Insert an Ethernet cable into the Jetson Orin NX carrier board to connect local network where the Linux PC exists.
2) Copy the kernel files to the Jetson Orin NX carrier board with the following command. The following commands assume that the IP address of the carrier board is 10. 1. 108. 50 and the login user name is "user".
   ```
   $ scp $TEGRA_KERNEL_OUT/arch/arm64/boot/Image user@10.1.108.50:~
   ```
3) Copy the device tree file to the Jetson Orin NX carrier board with the following command.
   ```
   $ scp $TEGRA_KERNEL_OUT/arch/arm64/boot/dts/nvidia/tegra234-p3767-0000-p3509-a02.dtb
   user@10.1.108.50:~
   ```
4) Copy the kernel module tree files to the Jetson Orin NX carrier board with the following command.
   ```
   $ cd $DEVDIR/images/modules/lib/modules/5.10.104/
   $ rsync --progress -avhe ssh ../5.10.104  user@10.1.108.50:~
   ```
5) Copy the display module files to the Jetson Orin NX carrier board with the following command.
   ```
   $ scp <display_kernel_working directory>/Linux_for_Tegra/source/public/NVIDIA-kernel-module-source-
   TempVersion/kernel-open/nvidia*.ko user@10.1.108.50:~
   ```

■ On the remote terminal of the PC connected with the Jetson Orin NX carrier board

3.8 Log in to the carrier board from the PC with the remote terminal.

1) Connect the PC with the carrier board via USB2 micro B cable.
2) Log in to the carrier board from the PC with the remote terminal such as PuTTY or Tera Term with the following setting.
   - Connection type: Serial
   - Port: Select the appropriate port connected to the carrier board.
   - Speed: 115200
   - Data: 8bit
   - Parity: none
   - Stop bits: 1bit
   - Flow control: none

3.9 Place the copied files and update the modules.

1) Move the kernel files on the Linux of the Jetson Orin NX board with the following command.
   ```
   $ sudo mv ~/Image /boot/
   ```
2) Move the device tree files on the Linux of the Jetson Orin NX board with the following command.
   ```
   $ sudo mv ~/tegra234-p3767-0000-p3509-a02.dtb /boot/dtb/kernel_tegra234-p3767-0000-p3509-a02.dtb
   ```
3) Move the kernel module files on the Linux of the Jetson Orin NX board with the following command
   ```
   $ sudo rm -r /lib/modules/5.10.104 (if exist)

   $ sudo mv ~/5.10.104/ /lib/modules/
   $ sudo ln -s /lib/modules/5.10.104 /lib/modules/5.10.104-tegra  (only once after Linux binary was flashed to
   the Jetson Orin NX board)
   ```
4) Move the display module files on the Linux of the Jetson Orin NX board with the following command.
   ```
   $ sudo mkdir /lib/modules/5.10.104/extra
   ```

```
$ sudo mkdir /lib/modules/5.10.104/extra/opensrc-disp

$ sudo cp ./nvidia*.ko /lib/modules/5.10.104/extra/opensrc-disp
```

5) Reboot the Jetson Orin NX carrier board with the following command.

```
$ sudo reboot
```

Wait until the reboot finishes. The display connected to the carrier board will be blank/black after the boot screen.

6) Update the modules with the following commands.

```
$ sudo depmod -a
$ sudo poweroff
```

7) Unplug the power adapter from the Jetson Orin NX carrier board after the Linux on the Jetson Orin NX shutdown.

3.10 Check the update of the built module.

1) Connect the THSCJ101 to the Jetson Orin NX carrier board via FFC cable.
2) Plug the AC adaptor to power on the Jetson Orin NX carrier board.
3) Check if the monitor connected to the carrier board displays the Ubuntu desktop. You succeeded in creating the Linux development environment if the monitor displays the Ubuntu desktop.

4. Integrate THSCJ101 V4L2 Driver to the Jetson Linux

■ On the Linux PC

4.1 Get the THSCJ101 V4L2 driver patch file and THSCJ101 firmware binary from THine and place the files into the <download_directory>.

✓ thscj101_v4l2driver_src.patch
✓ thscg101_thp7312.bin

4.2 Apply the patch.

1) Install git if not installed with the following command.

```
$ sudo apt-get install git
```

2) Copy the THSCJ101 V4L2 driver patch file into <working directry> with the following commands.

```
$ cd <working directory>
$ cd ..
$ cp <download folder>/thscj101_v4l2driver_src.patch .
```

3) Apply the patch with the following command.

```
$ git apply --directory=  thscj101_v4l2driver_src.patch
```

4.3 Build Kernel, device tree and modules.

The steps from 1) to 3) are identical to ones of 3.5 Build Kernel and device tree.

1) Build kernel with the following commands.

```
$ cd $KERNEL_DIR
$ make O=$TEGRA_KERNEL_OUT CROSS_COMPILE=${CROSS_COMPILE} tegra_defconfig
$ make O=$TEGRA_KERNEL_OUT CROSS_COMPILE=${CROSS_COMPILE} -j $(nproc) Image
```

2) Build device tree with the following command.

```
$ make O=$TEGRA_KERNEL_OUT CROSS_COMPILE=${CROSS_COMPILE} -j $(nproc) dtbs
```

3) Build module with the following commands.

```
$ make O=$TEGRA_KERNEL_OUT CROSS_COMPILE=${CROSS_COMPILE} -j $(nproc) modules
$ make ARCH=arm64 O=$TEGRA_KERNEL_OUT LOCALVERSION=-tegra INSTALL_MOD_PATH=$KERNEL_MODULES_OUT -j4 modules_install
```

4.4 Copy the built files and THSCJ101 firmware binary file to the Jetson Orin NX carrier board.
The following commands assume that the IP address of the carrier board is 10. 1. 108. 50 50 and the login user name is "user".

1) Copy the kernel file to the Jetson Orin NX board carrier board with the following command.

```
$ scp $TEGRA_KERNEL_OUT/arch/arm64/boot/Image user@10.1.108.50:~
```

2) Copy the device tree files to the Jetson Orin NX carrier board with the following commands.

```
$ scp $TEGRA_KERNEL_OUT/arch/arm64/boot/dts/nvidia/tegra234-p3767-0000-p3509-a02.dtb user@10.1.108.50:~
$ scp $TEGRA_KERNEL_OUT/arch/arm64/boot/dts/nvidia/tegra234-p3767-camera-p3768-thp7312-imx258-dual.dtbo user@10.1.108.50:~
```

3) Copy the kernel module files to the Jetson Orin NX board carrier board with the following commands.

```
$ cd $DEVDIR/images/modules/lib/modules/5.10.104/
$ rsync --progress -avhe ssh ../5.10.104  user@10.1.108.50:~
```

4) Copy THSCJ101 firmware binary file to the Jetson Orin NX carrier board with the following command.

```
$ cp <download folder>/ thscg101_thp7312.bin .
$ scp thscg101_thp7312.bin user@10.1.108.50:~
```

■ On the Jetson Orin NX carrier board

4.5 Place the copied files to update the modules.

1) Move the kernel file to the boot directory with the following command.

```
$ sudo mv ~/Image /boot/
```

2) Move each device tree file with the following commands.

```
$ sudo mv ~/tegra234-p3767-0000-p3509-a02.dtb /boot/dtb/kernel_tegra234-p3767-0000-p3509-a02.dtb
$ sudo mv ~/tegra234-p3767-camera-p3768-thp7312-imx258-dual.dtbo /boot/
```

3) Remove the original module files, and then move the module files with the following commands.

```
$ sudo rm -r /lib/modules/5.10.104

$ sudo mv ~/5.10.104/ /lib/modules/
```

4) Make directories for the display module, and then copy the module files into the directories with the following commands.

```
$ sudo mkdir /lib/modules/5.10.104/extra
$ sudo mkdir /lib/modules/5.10.104/extra/opensrc-disp

$ sudo cp ./nvidia*.ko /lib/modules/5.10.104/extra/opensrc-disp
```

5) Make directory for the THSCJ101 firmware binary and move copy the firmware binary into the directory with the following commands.

```
$ sudo mkdir /lib/firmware/thine
$ sudo cp ./thscg101_thp7312.bin /lib/firmware/thine/thscg101_thp7312.bin
```

6) Reboot the Jetson Orin NX with the following command.

```
$ sudo reboot
```

7) Wait the reboot finishes, then update modules with the following command.

```
$ sudo depmod -a
```

4.6 Apply the device tree overlay.

1) Execute the jetson.io script with the following command.

   ```
   $ sudo /opt/nvidia/jetson-io/jetson-io.py
   ```

   The prompt of the script appears in the terminal.
2) Select "Configure Jetson 24pin CSI connector" in the prompt, then press the Enter key.
3) Select "Configure for compatible hardware" in the prompt, then press the Enter key.
4) Select "Camera THP7312/IMX258 Dual" in the prompt, then press the Enter key.
5) Select "Save pin changes" in the prompt, then press the Enter key.
6) Select "Save and reboot to reconfigure pins" in the prompt then press any keys.
7) Wait the reboot finishes, then power off with the following command.

   ```
   $ sudo poweroff
   ```

4.7 Check the display module update.

1) Power up the Jetson Orin NX carrier board and check if the monitor displays the Ubuntu desktop.

4.8 Install the tools to streaming the image of THSCJ101 with the following commands.

```
$ sudo apt-get update

$ sudo apt-get -y install v4l-utils

$ sudo apt-get install gstreamer1.0-tools gstreamer1.0-alsa \
    gstreamer1.0-plugins-base gstreamer1.0-plugins-good \
    gstreamer1.0-plugins-bad gstreamer1.0-plugins-ugly \
    gstreamer1.0-libav

$ sudo apt-get install libgstreamer1.0-dev \
    libgstreamer-plugins-base1.0-dev \
    libgstreamer-plugins-good1.0-dev \
    libgstreamer-plugins-bad1.0-dev
```

4.9 Check if you can see the streaming images from THSCJ101 at all modes with the following commands. The Linux binary with THSCJ101 V4L2 driver creation succeeded if you can stream the images of all modes.

- 1920x1080@30fps mode

```
$ v4l2-ctl -d /dev/video0 --set-ctrl sensor_mode=0
$ gst-launch-1.0 v4l2src device=/dev/video0 !  video/x-raw,format=YUY2,
   width=1920,height=1080,framerate=30/1 ! xvimagesink
```

- 1920x1080@60fps mode

```
$ v4l2-ctl -d /dev/video0 --set-ctrl sensor_mode=1
$ gst-launch-1.0 v4l2src device=/dev/video0 !  video/x-raw,format=YUY2,
   width=1920,height=1080,framerate=60/1 ! xvimagesink
```

- 2048x1536@30fps mode

```
$ v4l2-ctl -d /dev/video0 --set-ctrl sensor_mode=2
$ gst-launch-1.0 v4l2src device=/dev/video0 !  video/x-raw,format=YUY2,
   width=2048,height=1536,framerate=30/1 ! xvimagesink
```

- 3840x2160@30fps mode

```
$ v4l2-ctl -d /dev/video0 --set-ctrl sensor_mode=3
$ gst-launch-1.0 v4l2src device=/dev/video0 !  video/x-raw,format=YUY2,
    width=3840,height=2160,framerate=30/1 ! xvimagesink
```

- 4160x3120@20fps mode

```
$ v4l2-ctl -d /dev/video0 --set-ctrl sensor_mode=4
$ gst-launch-1.0 v4l2src device=/dev/video0 !  video/x-raw,format=YUY2,
    width=4160,height=3120,framerate=20/1 ! xvimagesink
```

End of File