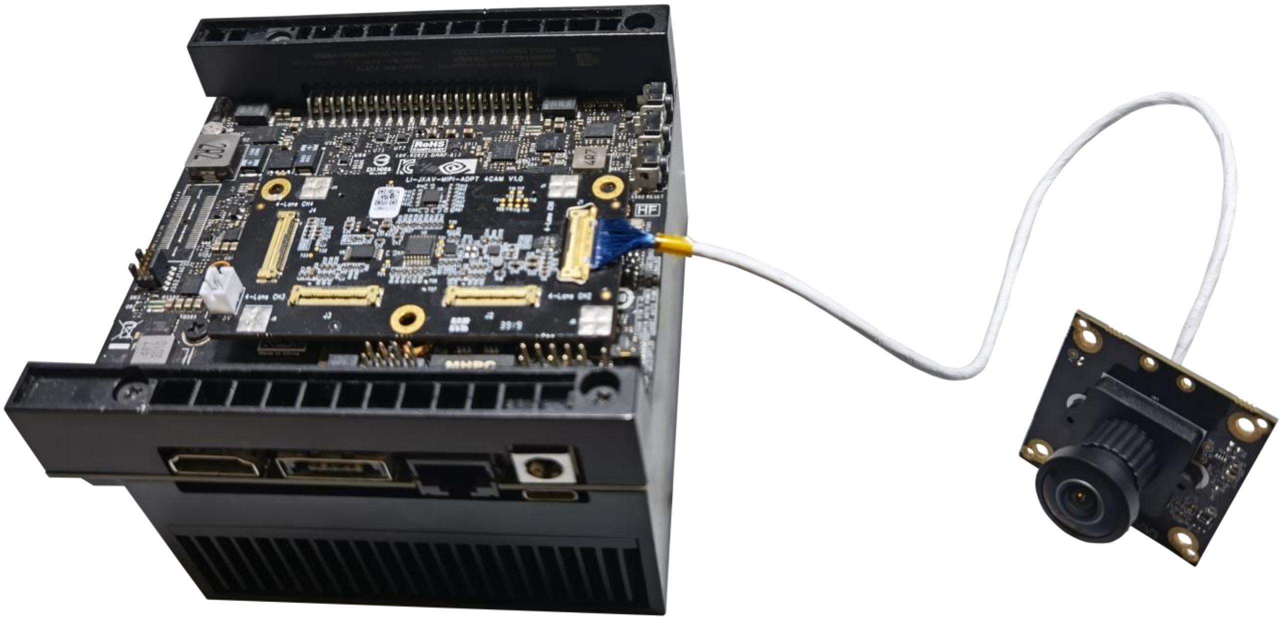
Overview

This driver is for LI-IMX477-MIPI-140H camera and Nvidia Jetson AGX Xavier Developer kit.
This driver supports up to four LI-IMX477-MIPI-140H cameras.
This driver supports 4056x3040@30fps.
This driver is based on R32.6.1 (Jetpack 4.6).

Download link

https://www.dropbox.com/sh/hnrijz8lw4srifp/AAB-dd_JVgY-dF9A_ZYfRfD7a?dl=0

Platform	Camera
Nvidia Jetson AGX Xavier Developer kit	1 x LI-IMX477-MIPI-140H
Cable	**Adapter/Carrier Board**
1 x FAW-1233-03	1 x LI-JXAV-MIPI-ADPT-4CAM



<table>
<tr><th>Revision</th><th>SVN version</th><th>Release Date</th><th>Author</th><th>Tested by</th></tr>
<tr><td>2021_10_13</td><td>Rev308</td><td>10/13/2021</td><td>Xingxing Gu</td><td>Zeng Yang</td></tr>
<tr><th colspan="5">Updates</th></tr>
<tr><th>Revision</th><th colspan="3">Description</th><th>Release Date</th></tr>
<tr><td>2021_10_13</td><td colspan="3">First Release based on R32.6.1.</td><td>10/13/2021</td></tr>
<tr><td></td><td colspan="3"></td><td></td></tr>
<tr><th colspan="5">Known bugs</th></tr>
<tr><td colspan="5"></td></tr>
</table>

Setup Procedure 1/2

Hardware:

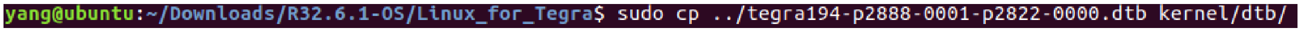
1. Nvidia Jetson AGX Xavier Developer Kit x 1
2. LI-IMX477-MIPI-140H x 1
3. FAW-1233 cable x 1
4. LI-JXAV-MIPI-ADPT-4CAM x 1
5. USB 3.0 Type-C cable x 1 (for OS image and dtb file flashing)
6. Ubuntu host PC x 1
7. Monitor with HDMI cable x 1
8. Keyboard and mouse (with USB hub) x 1

Driver installation:

1. Download the R32.6.1 OS Image (from link below) to your Ubuntu OS on Intel x64 Host PC (we are using Ubuntu 18.04, virtual machine is fine) and follow the l4t_quick_start_guide to install the Jetpack to Xavier.

R32.6.1 OS Image: https://www.dropbox.com/sh/qwrwtf1595dva7p/AAB3mRWJYi9A6a-8ldcq7hVva?dl=0

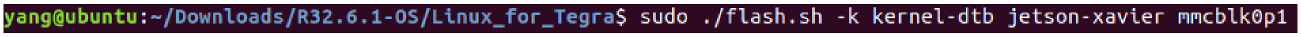
2. Reboot Xavier and put your system into "reset recovery mode" by holding down the RECOVER button and press the RESET button once on the Xavier.
3. Copy the tegra194-p2888-0001-p2822-0000.dtb (which was downloaded from the link in first page) and paste it under Xavier/Linux_for_Tegra/kernel/dtb on your Ubuntu host PC.



```
yang@ubuntu:~/Downloads/R32.6.1-OS/Linux_for_Tegra$ sudo cp ../tegra194-p2888-0001-p2822-0000.dtb kernel/dtb/
```

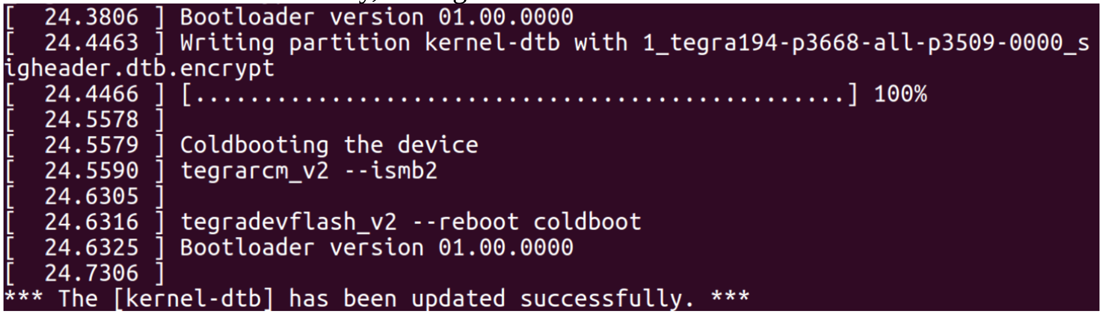
4. Under Xavier/Linux_for_Tegra/ do

```
sudo ./flash.sh -k kernel-dtb jetson-xavier mmcblk0p1
```



```
yang@ubuntu:~/Downloads/R32.6.1-OS/Linux_for_Tegra$ sudo ./flash.sh -k kernel-dtb jetson-xavier mmcblk0p1
```

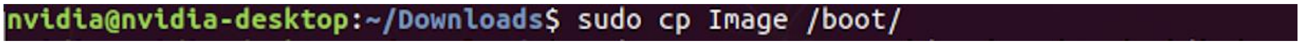
If flash the dtb file successfully, the log should be like below.



```
[  24.3806 ] Bootloader version 01.00.0000
[  24.4463 ] Writing partition kernel-dtb with 1_tegra194-p3668-all-p3509-0000_s
igheader.dtb.encrypt
[  24.4466 ] [..............................................] 100%
[  24.5578 ]
[  24.5579 ] Coldbooting the device
[  24.5590 ] tegrarcm_v2 --ismb2
[  24.6305 ]
[  24.6316 ] tegradevflash_v2 --reboot coldboot
[  24.6325 ] Bootloader version 01.00.0000
[  24.7306 ]
*** The [kernel-dtb] has been updated successfully. ***
```

Setup Procedure 2/2

5. After boot up Xavier, copy “Image” to /boot on Xavier.



```
nvidia@nvidia-desktop:~/Downloads$ sudo cp Image /boot/
```

6. Reboot Xavier kit.
7. Open a terminal and do below commands. The imx477.ko can be downloaded from the link in first page.

```
insmod imx477.ko
```

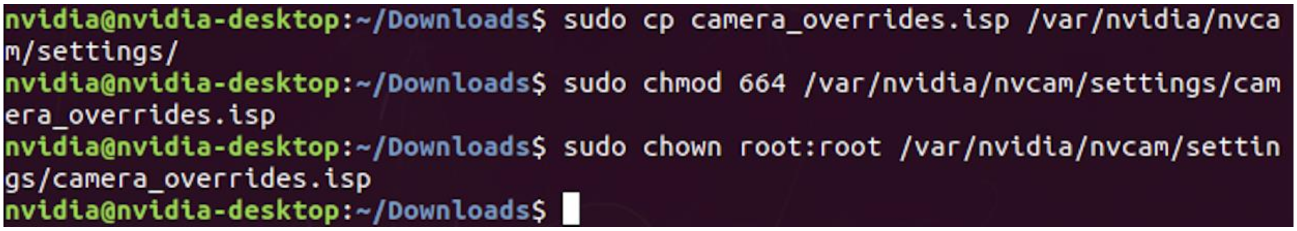
8. Then do below command to get live video output.

```
nvgstcapture-1.0
```

Note: Please make sure the camera is connected to J1 of LI-JXAV-MIPI-ADPT-4CAM board.

9. Use Ctrl+C to close the video and copy camera_overrides.isp to /var/nvidia/nvcam/settings on Xavier and do below two commands.

```
sudo chmod 664 /var/nvidia/nvcam/settings/camera_overrides.isp
sudo chown root:root /var/nvidia/nvcam/settings/camera_overrides.isp
```



```
nvidia@nvidia-desktop:~/Downloads$ sudo cp camera_overrides.isp /var/nvidia/nvca
m/settings/
nvidia@nvidia-desktop:~/Downloads$ sudo chmod 664 /var/nvidia/nvcam/settings/cam
era_overrides.isp
nvidia@nvidia-desktop:~/Downloads$ sudo chown root:root /var/nvidia/nvcam/settin
gs/camera_overrides.isp
nvidia@nvidia-desktop:~/Downloads$ 
```

10. Try "nvgstcapture-1.0” again. You should be able to see the image with better image quality.

Run Camera

1. Argus software

Download the Multimedia package from link below and copy it to Xavier.

https://www.dropbox.com/s/ik4e6bgprh3sozy/jetson_multimedia_api.tar?dl=0

Open a terminal, do

```
   sudo apt-get update
   sudo apt-get install cmake libgtk-3-dev libjpeg-dev libgles2-mesa-dev libgstreamer1.0-dev
```

Uncompress the tgz file.

```
  tar zxvf jetson_multimedia_api.tgz
```

Under jetson_multimedia_api/argus/cmake, do

```
   cmake ..
   make
   sudo make install
```

Do "argus_camera --device=0” to get the video.

2. Gstreamer

```
gst-launch-1.0 nvarguscamerasrc sensor-id=0 ! 'video/x-raw(memory:NVMM), width=(int)4056,
height=(int)3040, framerate=30/1'  ! nvvidconv flip-method=0 ! 'video/x-raw, format=(string)I420' !
xvimagesink -e
```

3. v4l2-ctl capture raw

```
v4l2-ctl -V --set-fmt-video=width=4056,height=3040,pixelformat=RG10 --set-ctrl bypass_mode=0 --stream-
mmap --stream-count=1 --stream-to=IMX477.raw -d /dev/video0
```

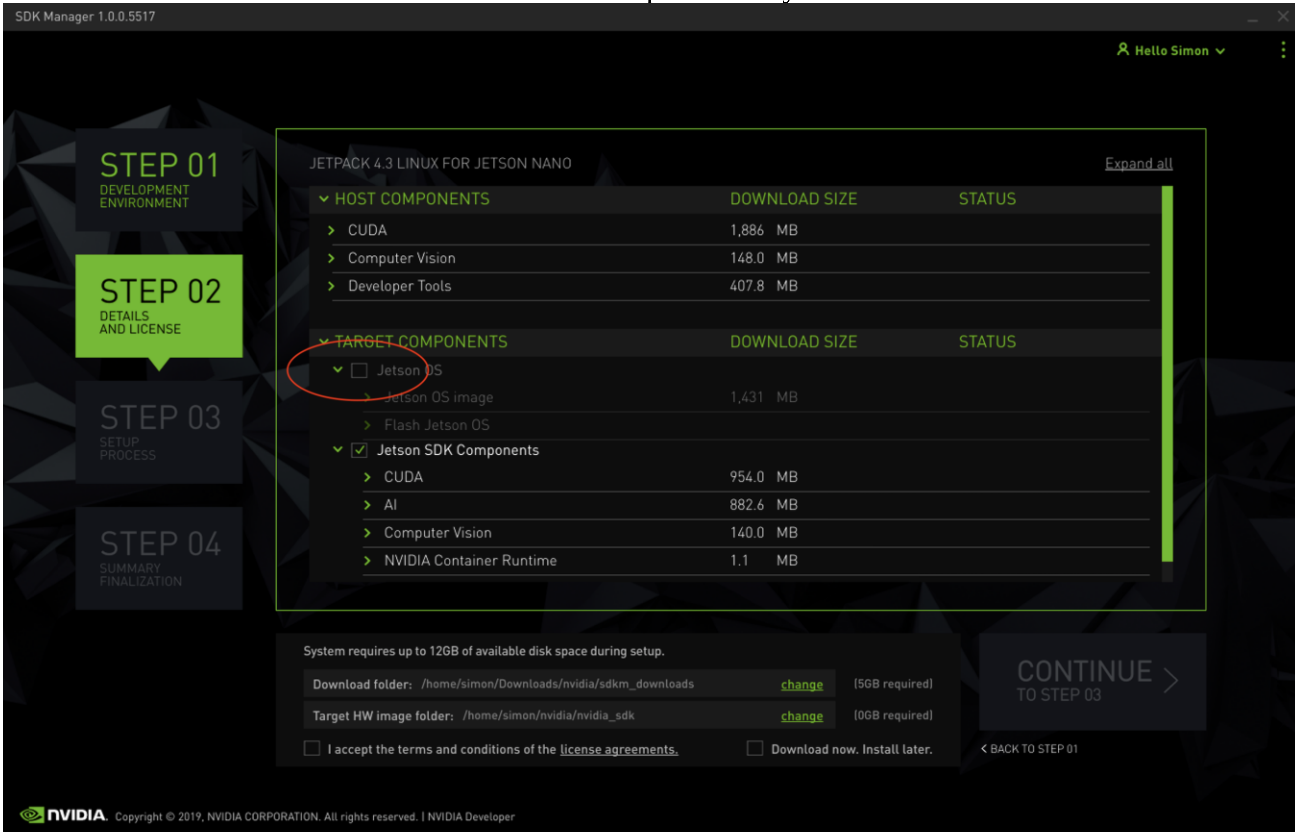
Note:

1）Please use below commands to install v4l2.

```
  sudo apt-get update
  sudo apt-get install v4l-utils
```

Note 1/2

1. If you would like to install the Jetpack 4.6 but don't want to re-flash the whole OS image, you can uncheck the Jetson OS and install the Jetson SDK components only.



JETPACK 4.3 LINUX FOR JETSON NANO

HOST COMPONENTS	DOWNLOAD SIZE	STATUS
CUDA	1,886 MB	
Computer Vision	148.0 MB	
Developer Tools	407.8 MB	

TARGET COMPONENTS	DOWNLOAD SIZE	STATUS
□ Jetson OS		
Jetson OS image	1,431 MB	
Flash Jetson OS		
☑ Jetson SDK Components		
CUDA	954.0 MB	
AI	882.6 MB	
Computer Vision	140.0 MB	
NVIDIA Container Runtime	1.1 MB	

System requires up to 12GB of available disk space during setup.

Download folder: /home/simon/Downloads/nvidia/sdkm_downloads change [5GB required]

Target HW image folder: /home/simon/nvidia/nvidia_sdk change [0GB required]

□ I accept the terms and conditions of the license agreements. □ Download now. Install later.

NVIDIA. Copyright © 2019, NVIDIA CORPORATION. All rights reserved. | NVIDIA Developer

Note 2/2

2. Compile the driver

If you would like to re-compile the driver, please follow below steps.
Download the driver code and Tool chain from links below.

Kernel code: https://www.dropbox.com/s/4k9o4zay08szde4/kernel_src_Xavier-NX-TX2_R32.6.1.tbz2?dl=0
GCC ToolChain: https://www.dropbox.com/sh/f21qck6f29h3n20/AABP8B1b4DgmUgO2MYO32Nyza?dl=0

Compile the kernel under 64 bit Ubuntu OS on Intel x64 PC. (Virtual machine is fine. We are using Ubuntu 16.04 64 bit OS)

1) Copy compile tool gcc-linaro-7.3.1-2018.05-x86_64_aarch64-linux-gnu.tar.xz to /opt, and unzip it

```
   sudo tar xpf gcc-linaro-7.3.1-2018.05-x86_64_aarch64-linux-gnu.tar.xz
```

2) Copy kernel_src_Xavier-NX-TX2_R32.6.1.tbz2 and two patch files to /usr/src

```
   sudo tar xpf kernel_src_Xavier-NX-TX2_R32.6.1.tbz2
  sudo chown -R <user_name> kernel
   sudo chown -R <user_name> hardware
   patch -p0 < IMX477-MIPI_32.6.1_Xavier_20211013_dtbs.patch
   patch -p0 < IMX477-MIPI_32.6.1_Xavier_20211013_kernel.patch
```

Note: <user_name> is the user name of your Ubuntu OS. For example: sudo chown -R leopard kernel

3) Copy xavier.sh to /usr/src/kernel.
   under /usr/src/kernel, do

```
   source xavier.sh
```

4) Create a work folder under /home:

```
   sudo mkdir /home/work
   sudo chown -R <user_name> /home/work
```

5) In "kernel/kernel-4.9" folder, run:

```
   make O=$TEGRA_KERNEL_OUT tegra_defconfig
   make O=$TEGRA_KERNEL_OUT zImage
   make O=$TEGRA_KERNEL_OUT dtbs
```

You will get Image under /home/work/Xavier/kernel/kernel_out/arch/arm64/boot and tegra194-p2888-0001-p2822-0000.dtb under /home/work/Xavier/kernel/kernel_out/arch/arm64/boot/dts.