



\* Copyright (c) 2019, Leopard Imaging Inc. All Rights Reserved.

## Overview

This driver is for LI-AR0231-TI953-XAVIER camera on Nvidia Jetson AGX Xavier Developer kit. This driver supports up to four LI-AR0231-TI953-XAVIER. The adapter board is LI-JXAV-MIPI-ADPT-4CAM.

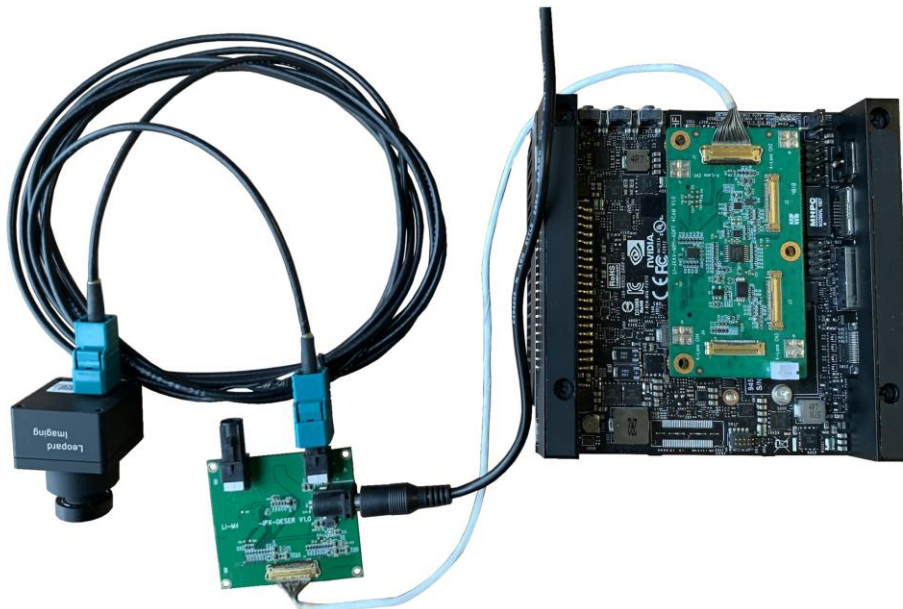
This driver supports 1928 x 1208@33fps.

This driver is based on R32.1 (Jetpack 4.2).

## Download link

<https://www.dropbox.com/sh/mnmoqjggsgxa40wo/AACnwc-TP00hf5yWlVZJtaJya?dl=0>

Platform	Camera
Nvidia Jetson AGX Xavier Developer kit	1 ~ 4 x LI-AR0231-FPDLINKIII
Cable	Adapter/Carrier Board
1 ~ 4 x FAW-1233 1 ~ 4 x FAK-SMZSMZ	1 x LI-JXAV-MIPI-ADPT-4CAM V1.0 1 ~ 4 x LI-TI954-IPX-DESER V1.0





## AR0231-TI953\_R32.1\_Xavier\_NV\_Quad\_20190423\_Driver\_Guide

Revision	SVN version	Release Date	Operator
20190423	Rev98	2019-04-23	Meng Gao
Updates			
<ol style="list-style-type: none"><li>1. First version of driver for LI-AR0231-TI953-XAVIER + Jetson AGX Xavier Developer kit.</li></ol>			
Known bugs			
<ol style="list-style-type: none"><li>1. The frame rate is only 6.7fps when opening video at the first time. The frame rate can be 33fps when opening video again.</li><li>2. Fan cannot be turned on</li><li>3. Cannot capture raw from video3</li></ol>			



## Setup Procedure 1/2

1. Download the R32.1 OS Image for Xavier (from link below) to your Ubuntu OS on Intel x64 Host PC (Ubuntu 18.04, virtual machine is fine) and follow the installation guide to install the Jetpack to Xavier.

Note: If your Xavier SOM already includes the Ubuntu OS with L4T **R32.1.0**, you can skip this step.

You can use below command to check the L4T version.

```
nvidia@nvidia-desktop:~$ cat /etc/nv_tegra_release
# R32 (release), REVISION: 1.0, GCID: 14531094, BOARD: t186ref, EABI: aarch64, D
ATE: Wed Mar 13 07:41:08 UTC 2019
```

R32.1 OS Image: <https://www.dropbox.com/sh/gzoxd0oi3aob719/AAAAALbhKquobyICkXbYz2SuTa?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. Connect the Xavier to your Ubuntu host PC with USB 3.0 Type-C cable, and do

lsusb

Make sure there is a device with "NVidia Corp."

```
simon@ubuntu:~/Xavier-R32.1/Linux_for_Tegra$ lsusb
Bus 001 Device 005: ID 0955:7019 NVidia Corp.
```

4. 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**.

5. Copy the Image (which were downloaded from the link in first page) and paste them under Xavier/Linux\_for\_Tegra/kernel on your **Ubuntu host PC**.

6. Under Xavier/Linux\_for\_Tegra/ do  
sudo ./flash.sh -r -k kernel-dtb jetson-xavier mmcblk0p1

```
simon@ubuntu:~/Xavier-R32.1/Linux_for_Tegra$ sudo ./flash.sh -r -k kernel-dtb je
tson-xavier mmcblk0p1
```

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

```
[ 9.2055 ] [ ..... ] 100%
[ 9.2664 ]
[ 9.2666 ] Coldbooting the device
[ 9.2692 ] tegrarc_m_v2 --ismb2
[ 9.3131 ]
[ 9.3155 ] tegradevflash_v2 --reboot coldboot
[ 9.3194 ] Bootloader version 01.00.0000
[ 9.3748 ]
*** The [kernel-dtb] has been updated successfully. ***
```

```
simon@ubuntu:~/Xavier-R32.1/Linux_for_Tegra$
```

7. The Xavier will restart. Set the Xavier to recovery mode again.

8. Under Xavier/Linux\_for\_Tegra/ do  
sudo ./flash.sh -r -k kernel jetson-xavier mmcblk0p1

```
simon@ubuntu:~/Xavier-R32.1/Linux_for_Tegra$ sudo ./flash.sh -r -k kernel jetson
-xavier mmcblk0p1
```



## Setup Procedure 2/2

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

```
[ 11.7457 ] [.....] 100%
[ 14.7169 ]
[ 14.7172 ] Coldbooting the device
[ 14.7214 ] tegrarcm_v2 --ismb2
[ 14.7701 ]
[ 14.7718 ] tegradevflash_v2 --reboot coldboot
[ 14.7733 ] Bootloader version 01.00.0000
[ 14.8322 ]
```

\*\*\* The [kernel] has been updated successfully. \*\*\*

simon@ubuntu:~/Xavier-R32.1/Linux\_for\_Tegra\$

Note: If you see below log, that means you haven't used the folder Linux\_for\_Tegra/ to re-flash the whole Xavier. If you don't want to lose the contents in the current Xavier SOM, you can re-flash a second Xavier SOM (if you have it). Then you can use the same folder (Linux\_for\_Tegra/) to flash the dtb and kernel files to the current Xavier SOM. The "file does not exist" issue won't happen again.

```
copying soft_fuses(/home/simon/Xavier-R32.1/Linux_for_Tegra/bootloader/t186ref/B
CT/tegra194-mb1-soft-fuses-l4t.cfg)... done.
copying dtbfile(/home/simon/Xavier-R32.1/Linux_for_Tegra/kernel/dtb/tegra194-p28
88-0001-p2822-0000.dtb)... done.
Reusing existing system.img...
file does not exist.
simon@ubuntu:~/Xavier-R32.1/Linux_for_Tegra$
```

9. Open a terminal and do "nvgstcapture-1.0". You will get live video output.

Note: Please make sure there is a camera on port J2 of LI-JXAV-MIPI-ADPT-4CAM board.

10. Use Ctrl+C to close the video and copy camera\_overrides.isp to /var/nvidia/nvcam/settings on Xavier and do below two command.

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

```
nvidia@jetson-0423318029458:~/Downloads$ sudo cp camera_overrides.isp /var/nvidia/n
vcam/settings/
nvidia@jetson-0423318029458:~/Downloads$ sudo chmod 664 /var/nvidia/nvcam/settings/
camera_overrides.isp
nvidia@jetson-0423318029458:~/Downloads$ sudo chown root:root /var/nvidia/nvcam/set
tings/camera_overrides.isp
```

11. 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/8t88bpqiv7e1s9i/JAX-TX2-Tegra\\_Multimedia\\_API\\_R32.1.0\\_aarch64.tbz2?dl=0](https://www.dropbox.com/s/8t88bpqiv7e1s9i/JAX-TX2-Tegra_Multimedia_API_R32.1.0_aarch64.tbz2?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
```

Under tegra\_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)1928, height=(int)1208' ! nvvidconv flip-method=0 ! 'video/x-raw, format=(string)I420' ! xvimagesink -e
```

### 3. v4l2-ctl capture raw

```
v4l2-ctl -V --set-fmt-video=width=1928,height=1208,pixelformat=RG12 --set-ctrl bypass_mode=0 --stream-mmap --stream-count=3 --stream-to=ar0231.raw -d /dev/video0
```

Note:

1. The **0** can be changed to 1 ~ 3 if there are other cameras connected.

Connector J1 ---- video2

Connector J2 ---- video0

Connector J3 ---- video3

Connector J4 ---- video1

2. Please use below commands to install v4l2.

```
sudo add-apt-repository main
```

```
sudo add-apt-repository universe
```

```
sudo add-apt-repository restricted
```

```
sudo add-apt-repository multiverse
```

```
sudo apt-get update
```

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



Note 1/2

1. If there are any new drivers, we will add them into link below.

[https://www.dropbox.com/sh/dlerykc12bwfy5e/AADyebHDx2GxwEAV6Ws\\_nNZ0a?dl=0](https://www.dropbox.com/sh/dlerykc12bwfy5e/AADyebHDx2GxwEAV6Ws_nNZ0a?dl=0)





## Note 2/2

### 2. Compile the driver

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

Kernel code: [https://www.dropbox.com/s/9pca0rqmtr9hoj4/kernel\\_src\\_R32.1\\_JAX\\_TX2.tbz2?dl=0](https://www.dropbox.com/s/9pca0rqmtr9hoj4/kernel_src_R32.1_JAX_TX2.tbz2?dl=0)  
GCC ToolChain: <https://www.dropbox.com/sh/n2sjsidaaxp2lbn/AAB1kOZLRjrpleh8AVrxPnJ0a?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-6.4.1-2017.08-x86\_64\_aarch64-linux-gnu.tar.xz to /opt, and unzip it

```
sudo tar xpf gcc-linaro-6.4.1-2017.08-x86_64_aarch64-linux-gnu.tar.xz
```

- 2) Copy kernel\_src\_R32.1\_JAX\_TX2.tbz2 and two patch files to /usr/src

```
sudo tar xpf kernel_src_R32.1_JAX_TX2.tbz2
```

```
sudo chown -R <user_name> kernel
```

```
sudo chown -R <user_name> hardware
```

```
patch -p0 < Quad_streaming_AR0231-TI953_base32.1_Xavier_dts_20190423.patch
```

```
patch -p0 < Quad_streaming_AR0231-TI953_base32.1_Xavier_kernel_20190423.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-**ar0231**.dtb under /home/work/Xavier/kernel/kernel\_out/arch/arm64/boot/dts. Please change the name of dtb file from tegra194-p2888-0001-p2822-0000-**ar0231**.dtb to tegra194-p2888-0001-p2822-0000.dtb before installing the driver.