

Helpdesk XIMEA

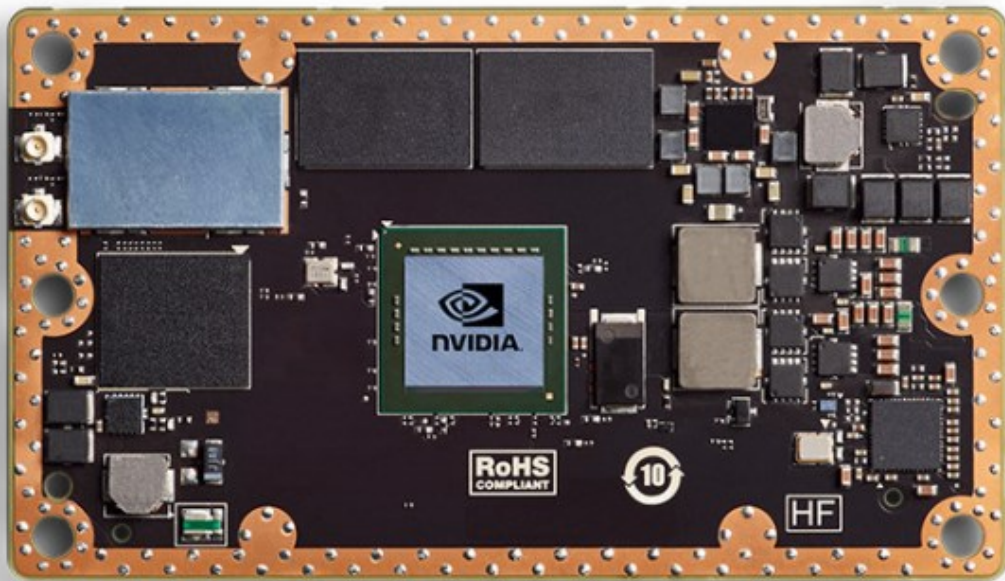
Portal > Knowledgebase > General > ARM support > XIMEA Linux API for Nvidia Jetson (TX1 / TX2 / TX2i / Nano / AGX Xavier / NX)

XIMEA Linux API for Nvidia Jetson (TX1 / TX2 / TX2i / Nano / AGX Xavier / NX)

Support SK - 2024-05-20 - in ARM support

https://www.ximea.com/support/wiki/apis/linux_tx1_and_tx2_support

XIMEA Linux API for Nvidia Jetson TX1 / TX2 / TX2i



- [XIMEA Linux API for Nvidia Jetson TX1 / TX2 / TX2i](#)
 - [Introduction](#)
 - [Developer Kit setup](#)
 - [NVIDIA Jetson TX1, TX2, TX2i](#)
 - [Useful links Jetson TX2](#)
 - [Re-flashing xEC2 with L4T 32.5.1](#)
 - [Installing XIMEA API package](#)
 - [PCIe support](#)
 - [Memory bandwidth problems](#)
 - [Running sample applications](#)

Introduction

The following webpage describes the range of support for NVIDIA Jetson modules.

All of these excel in delivering high performance and power efficiency packed into a small size form.

They are ideal for use in deep learning, computer vision and other computing applications.



Follow the official quick start guide that

TEGRA X1/TEGRA LINUX DRIVER PACKAGE MULTIMEDIA I/O

In case your host machine isn't running Ubuntu, you

```
sudo cp /etc/mke2fs.conf .
sudo sed -i 's/64bit,//g' mke2fs.conf
```

```
MKE2FS_CONFIG="$(pwd)"/mke2fs.conf sudo -E ./flash.sh ${BOARD}  
mmcblk0p1
```

where \${BOARD} is one of the following:

jetson-tx1, jetson-tx2, jetson-tx2i or jetson-xavier.



Useful links Jetson TX2

The official page for TX2 Devkit can be found here:

<https://developer.nvidia.com/embedded/jetson-tx2-developer-kit>

Results of USB3 and PCIe interface cameras connected to the module:

[Jetson TX2 Benchmarks](#)

Jetson TX2 NX module initial test results with XIMEA camera models:

[Jetson TX2 NX Benchmarks](#)

[Jetson TX2 vs AGX Xavier comparison,](#)

[Jetson Nano, TX2 and AGX Xavier comparison,](#)

[Jetson Image Processing,](#)

[Jetson Zero Copy for Embedded applications,](#)

[Realtime image processing on NVIDIA GeForce RTX 2080ti](#)

Re-flashing [xEC2](#) with L4T 32.5.1

Linux host is required for this procedure.

1. Download [L4T Driver Package](#) and [Sample Root Filesystem](#) from

<https://developer.nvidia.com/embedded/linux-tegra-archive> for L4T version 32.5.1

2. Download [TX2_xEC2_32.5.1.tar.bz2](#) (for TX2) or [TX2i_xEC2_32.5.1.tar.bz2](#) (for TX2i)
3. Switch TX2 to recovery mode:
 1. connect USB Micro-B port on xEC2 to the host
 2. enable DIP switch closest to LEDs ([xEC2-recovery_mode.jpg](#))
 3. connect the power to the board
 4. switch the DIP switch off after a few seconds
4. Execute the following commands under root account (`sudo -s` can be used to gain privileges):

[illegible]

5. Give TX2 several minutes to finish installation after it boots
6. Now you can log in to nvidia account using nvidia password, either by connecting the keyboard, mouse and monitor directly to xEC2 or via SSH

When the board is connected to the host using a USB cable as described above, the network connection is automatically created and the board can be accessed at IP

192.168.55.1.

Alternatively, when using Ethernet (or Wi-Fi) connection you can use nmap utility to find the IP address of TX2 (assuming the host and TX2 are connected to the same physical segment of the network):

```
sudo nmap -sn -PR $(ip addr show | grep 'inet ' | grep -v 'scope host  
lo' | cut -f6 -d' ') | grep -B 2 Nvidia
```

Installing XIMEA API package

Run the following commands on Jetson:

```
sudo apt update
sudo apt install ca-certificates
wget https://www.ximea.com/downloads/recent/XIMEA_Linux_ARM_SP.tgz
tar -xf XIMEA_Linux_ARM_SP.tgz
cd package
./install
sudo gpasswd -a "$(whoami)" plugdev
if [ -f /etc/rc.local ]
then
sudo sed -i '/^exit/ d' /etc/rc.local
else
echo '#!/bin/sh -e' | sudo
tee /etc/rc.local > /dev/null
fi
echo 'echo 0 > /sys/module/usbcore/parameters/usbfs_memory_mb' | sudo
tee -a /etc/rc.local > /dev/null
echo 'exit 0' | sudo
tee -a /etc/rc.local > /dev/null
sudo chmod a+x /etc/rc.local
#enable controlling of memory frequency by user
echo 'KERNEL=="emc_freq_min", ACTION=="add", GROUP="plugdev",  
MODE="0660"' | sudo tee /etc/udev/rules.d/99-emc_freq.rules >
/dev/null
#optional: allow user to use realtime priorities
sudo groupadd -fr realtime
echo '* - rtprio 0' | sudo tee
/etc/security/limits.d/ximea.conf > /dev/null
```

```

echo '@realtime - rtprio 81' | sudo tee -a
/etc/security/limits.d/ximea.conf > /dev/null
echo '* - nice 0' | sudo tee -a
/etc/security/limits.d/ximea.conf > /dev/null
echo '@realtime - nice -16' | sudo tee -a
/etc/security/limits.d/ximea.conf > /dev/null
sudo gpasswd -a "$(whoami)" realtime
#reboot
sudo reboot

```

PCle support¶

On 64-bit version of Linux For Tegra (available since R24.1) it is possible to use PCIe cameras.

To enable support for PCIe in XIMEA API you need to run `install` script (see [installation section](#)) with `-pcie` flag:

```
./install -pcie
```

Memory bandwidth problems¶

By default, the memory frequency is controlled by the kernel dynamically, which means that sufficient memory bandwidth isn't available at all times.

This causes crashes, hangs and lost frames when PCIe or USB3 cameras are used.

The solution is to raise the lower limit for memory frequency when image acquisition is running.

The following wrapper script can be used for that:

```

#!/bin/bash
# TX1:
# # cat /sys/kernel/debug/clock/emc/possible_rates
# 40800 68000 102000 204000 408000 665600 800000 1065600 1331200
# 1600000 (kHz)
# TX2:
# # cat /sys/kernel/debug/bpmp/debug/emc/possible_rates
# 40800 68000 102000 204000 408000 665600 800000 1062400 1331200
# 1600000 1866000 (kHz)
# Xavier:
# # cat /sys/kernel/debug/bpmp/debug/emc/possible_rates
# 204000 408000 665600 800000 1065600 1331200 1600000 1866000 2133000
# (kHz)

```

```
EMC_FREQ=1600000
exec 3> /dev/emc_freq_min
printf '%x' "$EMC_FREQ" >&3
exec "$@"
```

Save it to `~/emc_wrapper.sh` file and execute the following command:

```
chmod a+x ~/emc_wrapper.sh
```

Running sample applications

To do a simple validation that everything is working you can run `xiSample`:

```
~/emc_wrapper.sh /opt/XIMEA/bin/xiSample
```

To view a live picture from a camera start [CamTool](#):

```
~/emc_wrapper.sh /opt/XIMEA/bin/xiCamTool
```

Alternatively `streamViewer` application can be used, but you need to compile it first:

```
cd /opt/XIMEA/examples/streamViewer
sudo apt update
sudo apt install libgtk2.0-dev libgstreamer-plugins-base1.0-dev
gstreamer1.0-x
sudo make GST10=1
sudo make install
```

Then you can start the viewer:

```
~/emc_wrapper.sh /opt/XIMEA/bin/streamViewer
```

Tags

Linux