



AUVIDEA

SOFTWARE

SETUP GUIDE

SCOPE OF WORK

Help you to flash your Auvidea carrier board system for the first time and get everything up and running.

REPORT NUMBER

ISSUE DATE
14.Mar.2022

[REVISED DATE]
[FEB.2023]

Version
2.1

PAGES
24



Contents

SECTION 1	Document revisions and changes	3
SECTION 2	Simple flashing guide (recommended).....	4
2.1	Before you start	4
2.2	Download installation file from Auvidea	4
2.3	Connect carrier board to host PC.....	5
2.4	Flashing of system.....	6
2.5	Maintaining functionality.....	7
2.6	Installing additional NVIDIA SDK components.....	7
SECTION 3	Advanced flashing guide (experienced users).....	9
3.1	Before you start	9
3.2	Install and configure NVIDIA SDK manager	9
3.3	Download installation files from Auvidea.....	10
3.4	Flashing the AGX Orin	11
3.4.1	AGX Orin Jetpack compatibility	11
3.4.2	Necessary changes	11
3.5	Flashing of system.....	12
3.6	Installing additional NVIDIA SDK components.....	13
SECTION 4	Preliminary Orin nano & Orin NX flashing guide	14
4.1	Before you start	14
4.2	Download installation file from NVIDIA.....	14
4.3	Connect carrier board to host PC.....	16
4.4	Flashing of system.....	17
4.5	Maintaining functionality.....	18
SECTION 5	SSD-Boot flashing guide (experienced users)	19
5.1	Partial SSD boot for Jetson Nano, NX, TX2 NX	19
5.1.1	Before you start.....	19
5.1.2	Locate and identify storage device.....	19
5.1.3	Set up RootFS on SSD.....	20
5.1.4	Switch boot device to SSD.....	20
5.1.5	Validate	21
5.2	Native SSD boot for Jetson AGX Xavier, AGX Orin, Xavier NX series	21
5.2.1	Before you start.....	21
5.2.2	Flash system	21
SECTION 6	Frequently Asked Questions.....	22
SECTION 7	Disclaimer.....	23
SECTION 8	Trademarks.....	24

SECTION 1 Document revisions and changes

Document version	Changes
1.1	Document overhaul of quick starter guide, internal verification process
1.2	Small fixes
1.3	Fixed spelling, cleaner Headings
1.6	Surpassed QuickStart guide version number to better indicate that the Software_Setup_Guide should be used in the future
1.7	Integration of the SSD-Boot flashing instructions, formally Appendix D from the technical manuals
1.8	Added instructions for flashing a AGX Orin, as additional steps are needed.
1.9	Added notice that Python 2.x is required for the NVIDIA SDK Manager
2.0	Added instructions on how to enter force recovery mode when automatic force recovery mode does not work. Polishing of some chapters. Added information about conflicts with “sudo apt-get upgrade”. Corrected command for native SSD boot.
2.1	Added Preliminary Orin nano & Orin NX flashing guide section.

SECTION 2 Simple flashing guide (recommended)

This section describes how to flash your Auvideo carrier board system so it can boot and run. With this guide everything you need is included in the download package from Auvideo. This flashing guide is recommended for the Auvideo carrier board series. If you are using a Orin nano or Orin NX module please see the Orin nano & Orin NX flashing section.

2.1 Before you start

- Please make sure to use a Linux host PC with Ubuntu 18.04 LTS or Ubuntu 20.04 LTS (other versions may work but some caused problems in the past!) operating system. Please use a native setup (no virtual machine). This Host PC should have a high bandwidth internet connection for the download of 3GByte+ installation file in the following steps.
- You should make sure that Python 2.x is installed as it is required for the NVIDIA SDK Manager. You can check your installation in the command line with: `python --version` example return: `Python 2.7.17`. If you do not get a return or error you do not have Python 2.x installed. Please use: `sudo apt install python` to install Python 2.x to your system)
- You will also need a high-quality standard USB 2.0 Type A to micro-USB 2.0 cable.
- If you are using a AGX Orin please use the Advanced flashing guide.

2.2 Download installation file from Auvideo

- 1) Download the installation file for your setup from Auvideo. JNxxx means the carrierboard series starting with JN, for example JN30D. Please check the description to download the right file for your module (NX, TX2,...)

Link	https://auvideo.eu/firmware/		
Auvideo webpage (example)	Date	Product	Version Description
	Feb 2022	JNxxx (4.05 GB) firmware for Jetpack 4.6	1.0 supports: all Auvideo JNxxx carrier boards with Xavier NX compute module (8GB) please check quick start guide for installation instructions (this firmware is required to enable SPI, CSI-2 and SD card)
	Feb 2022	JNxxx (4.05 GB) firmware for Jetpack 4.6	1.0 supports: all Auvideo JNxxx carrier boards with TX2 NX compute module please check quick start guide for installation instructions (this firmware is required to enable SPI, CSI-2 and SD card)
	Feb 2022	JNxxx (4.05 GB) firmware for Jetpack 4.6	1.0 supports: all Auvideo JNxxx carrier boards with Nano compute module (B01) please check quick start guide for installation instructions (this firmware is required to enable SPI, CSI-2 and SD card)
	Feb 2022	Xxxx (4.09 GB) firmware for Jetpack 4.6	1.0 supports: all Auvideo Xxxx carrier boards with AGX Xavier 32GB compute module (e.g. X220, X221, X221D, and X400) please check quick start guide for installation instructions (this firmware is required to enable 2nd HDMI, CSI-2, 2nd GbE, and PCIe x1)

- 2) Open a terminal window (CTRL + ALT + T) on your Linux host PC and navigate to your download location.

Host PC	<code>cd <path_to_downloaded_tar></code>
---------	--

- 3) Extract the tar.gz file you just downloaded.

Host PC	<code>tar xvzf bootloader.tar.gz</code>
Terminal host PC (example)	<pre> auvidea@auvidea-HP-Z620-Workstation: ~/Downloads File Edit View Search Terminal Help auvidea@auvidea-HP-Z620-Workstation:~/Downloads\$ tar xvzf bootloader.tar.gz ./bootloader/ ./bootloader/nv_boot_control.conf ./bootloader/mkbctpart ./bootloader/kernel_tegra210-p3448-0002-p3449-0000-b00.dtb.sb ./bootloader/flash.xml.sb ./bootloader/initrd ./bootloader/rp4.blob ./bootloader/flash_win.bat ./bootloader/tegrasign_v3_internal.py ./bootloader/eks.img ./bootloader/BUP_generator.py ./bootloader/nvtboot.bin ./bootloader/LICENSE ./bootloader/LICENSE.mkgpt ./bootloader/tos.img ./bootloader/mkspare ./bootloader/tos-mon-only.img ./bootloader/system.img </pre>

4) Change directory to the extracted bootloader folder.

Host PC	<code>cd ./bootloader</code>
Terminal host PC (example)	<pre> auvidea@auvidea-HP-Z620-Workstation: ~/Downloads/bootloader File Edit View Search Terminal Help ./bootloader/t210ref/BCT/P2180_A00_LP4_DSC_204Mhz.cfg ./bootloader/t210ref/BCT/E2220_LP3_DSC_931.2Mhz.cfg ./bootloader/t210ref/BCT/P3448_A00_lpddr4_204Mhz_P987.cfg ./bootloader/t210ref/BCT/P2894_A00_Samsung_3GB_lpddr4_204Mhz_P984_v2.cfg ./bootloader/t210ref/cboot.bin ./bootloader/t210ref/p2371-0000/ ./bootloader/t210ref/p2371-0000/u-boot.bin ./bootloader/t210ref/LICENSE.cboot ./bootloader/t210ref/p3541-0000/ ./bootloader/t210ref/p3541-0000/u-boot.bin ./bootloader/bmp.blob ./bootloader/tegrasign_v3.py ./bootloader/crc-flash.xml.tmp ./bootloader/mkbootimg ./bootloader/P3448_A00_lpddr4_204Mhz_P987.cfg ./bootloader/LICENSE.tegraopenssl ./bootloader/LICENSE.tos-mon-only.img.arm-trusted-firmware ./bootloader/cboot.bin ./bootloader/kernel_tegra210-p3448-0002-p3449-0000-b00.dtb ./bootloader/nvidia-l4t-bootloader_32.6.1-20210726122000_arm64.deb ./bootloader/tegrahost ./bootloader/LICENSE.mkspare auvidea@auvidea-HP-Z620-Workstation:~/Downloads\$ cd ./bootloader/ auvidea@auvidea-HP-Z620-Workstation:~/Downloads/bootloader\$ </pre>

2.3 Connect carrier board to host PC

- 5) Connect the system to the Linux host PC. Please use a USB 2.0 cable (micro-USB on the carrier board).
- 6) Connect your system to power (please see specifications to apply the right voltage for your system when using your own power supply). The system will start automatically and detect the host PC and enter the flashing state (also called force recovery mode).
 In some cases, you will have manual press or short pins before connecting to your power source to enter force recovery mode. For example, with the JN30D you will need to short Pins 7 & 8 of J32 to enter force recovery. A simple way is to use tweezers. Please see the Technical Reference Manual for a detailed pin description. You can also see the FAQs in this document for further informations.

- 7) Check that the connection is established with the “lsusb” command. You should find one entry with “Nvidia Corp.” as highlighted below.

Host PC	<code>lsusb</code>
Terminal host PC example	<pre> auvidea@auvidea-HP-Z620-Workstation: /media/auvidea/Storage/Nvidia/Images/JetPa... File Edit View Search Terminal Help auvidea@auvidea-HP-Z620-Workstation: /media/auvidea/Storage/Nvidia/Images/JetPack _4.6_Linux_JETSON_NANO_TARGETS/Linux_for_Tegra\$ lsusb Bus 002 Device 036: ID 0955:7e19 Nvidia Corp. Bus 002 Device 032: ID 046d:c03e Logitech, Inc. Premium Optical Wheel Mouse (M-B T58) Bus 002 Device 031: ID 046a:0023 Cherry GmbH CyMotion Master Linux Keyboard G230 Bus 002 Device 002: ID 8087:0024 Intel Corp. Integrated Rate Matching Hub Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub Bus 004 Device 002: ID 2109:0815 VIA Labs, Inc. Bus 004 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub Bus 003 Device 012: ID 067b:2303 Prolific Technology, Inc. PL2303 Serial Port Bus 003 Device 002: ID 2109:2815 VIA Labs, Inc. Bus 003 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub Bus 001 Device 002: ID 8087:0024 Intel Corp. Integrated Rate Matching Hub Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub auvidea@auvidea-HP-Z620-Workstation: /media/auvidea/Storage/Nvidia/Images/JetPack _4.6_Linux_JETSON_NANO_TARGETS/Linux_for_Tegra\$ </pre>

Your system is now connected properly and ready to be flashed

2.4 Flashing of system

- 8) Use the flashcmd script in the extracted bootloader folder to transfer the software into the Jetson compute module and flash it.

Host PC	<code>sudo bash ./flashcmd.txt</code>
Terminal host PC example	<pre> auvidea@auvidea-HP-Z620-Workstation: ~/Downloads/bootloader File Edit View Search Terminal Help auvidea@auvidea-HP-Z620-Workstation:~/Downloads/bootloader\$ sudo bash flashcmd.txt [sudo] password for auvidea: Welcome to Tegra Flash version 1.0.0 Type ? or help for help and q or quit to exit Use ! to execute system commands [0.0025] tegrasign --getmode mode.txt --key None [0.0037] Assuming zero filled SBK key [0.0039] [0.0040] Generating RCM messages [0.0065] tegrarcmm --listrcm rcm_list.xml --chip 0x21 0 --download rcm nvtboot_recovery.bin 0 [0.0076] RCM 0 is saved as rcm_0.rcm [0.0084] RCM 1 is saved as rcm_1.rcm [0.0084] List of rcm files are saved in rcm_list.xml [0.0084] [0.0085] Signing RCM messages [0.0109] tegrasign --key None --list rcm_list.xml --pubkeyhash pub_key.key [0.0121] Assuming zero filled SBK key [0.0219] [0.0220] Copying signature to RCM messages [0.0248] tegrarcmm --chip 0x21 0 --updatesig rcm_list_signed.xml [0.0271] [0.0272] Parsing partition layout [0.0300] tegraparser --pt flash.xml.tnp [0.0325] [0.0326] Using default ramcode: 0 [0.0327] Disable BPMP dtb trin, using default dtb </pre>

- 9) Please connect a monitor to the system. After the flashing process has completed the system should automatically boot and show the Ubuntu desktop.



You now have a functioning system ready for your needs.

2.5 Maintaining functionality

After successful flashing please be aware that if you use “*sudo apt-get upgrade*” some of the applied changes performed in this guide (Bootloader, Kernels, DTBs, ...) will be overwritten by standard Nvidia packages. This will most probably brick some board functionality and some interfaces may not work anymore.

When overriding the Auvideo configurations only basic functionality like 1080p HDMI, some USB ports, Ethernet could work

Without the correct configurations and packages 4k HDMI, PCIe ports, some USB ports, CSI, CAN, ... will not work.

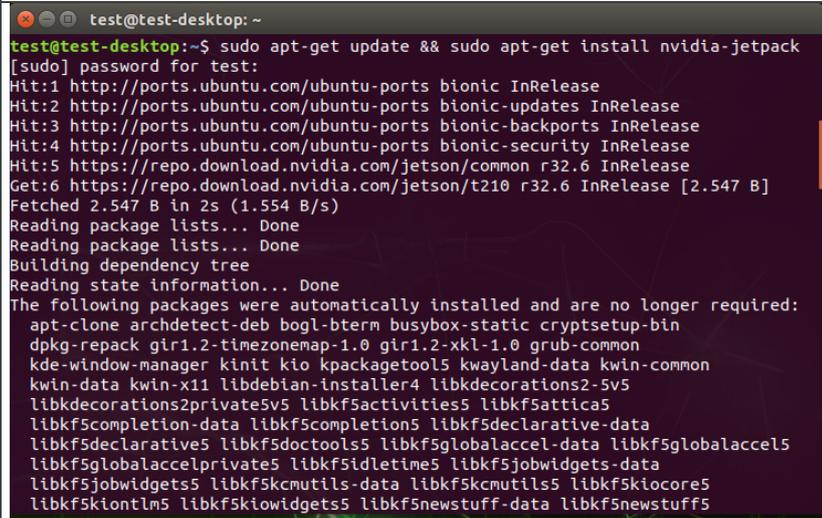
If you know what you are doing you can exclude the Nvidia Debian packages from the Nvidia sources.list from being updated. This prevents overriding relevant files and changes.

If you are unsure, we recommend **DO NOT UPGRADE YOUR SYSTEM**.

In the case you accidental upgraded your system and some board functionality stopped working correctly we recommend to reflash your system according to this guide.

2.6 Installing additional NVIDIA SDK components

- 10) Now you can install additional NVIDIA SDK components. Please connect the system to the Internet. Open a terminal window on the system (CTRL ALT T). Use aptget to install the components. If this fails, please check the Internet connection of the system.

Jetson system	<pre>sudo apt-get update && sudo apt-get install nvidia-jetpack</pre>
Terminal Jetson system (example)	 <pre>test@test-desktop: ~ test@test-desktop:~\$ sudo apt-get update && sudo apt-get install nvidia-jetpack [sudo] password for test: Hit:1 http://ports.ubuntu.com/ubuntu-ports bionic InRelease Hit:2 http://ports.ubuntu.com/ubuntu-ports bionic-updates InRelease Hit:3 http://ports.ubuntu.com/ubuntu-ports bionic-backports InRelease Hit:4 http://ports.ubuntu.com/ubuntu-ports bionic-security InRelease Hit:5 https://repo.download.nvidia.com/jetson/common r32.6 InRelease Get:6 https://repo.download.nvidia.com/jetson/t210 r32.6 InRelease [2.547 B] Fetched 2.547 B in 2s (1.554 B/s) Reading package lists... Done Reading package lists... Done Building dependency tree Reading state information... Done The following packages were automatically installed and are no longer required: apt-clone archdetect-deb bogl-bterm busybox-static cryptsetup-bin dpkg-repack gir1.2-timzone-map-1.0 gir1.2-xkl-1.0 grub-common kde-window-manager kinit kio kpackage-tools kwayland-data kwinn-common kwin-data kwinn-x11 libdebian-installer4 libkdecorations2-5v5 libkdecorations2-private5v5 libkf5activities5 libkf5attica5 libkf5completion-data libkf5completion5 libkf5declarative-data libkf5declaratives libkf5doctools5 libkf5globalaccel-data libkf5globalaccel5 libkf5globalaccel-private5 libkf5idle-time5 libkf5jobwidgets-data libkf5jobwidgets5 libkf5kcmutils-data libkf5kcmutils5 libkf5kiocore5 libkf5kiontln5 libkf5kiowidgets5 libkf5newstuff-data libkf5newstuff5</pre>

SECTION 3 Advanced flashing guide (experienced users)

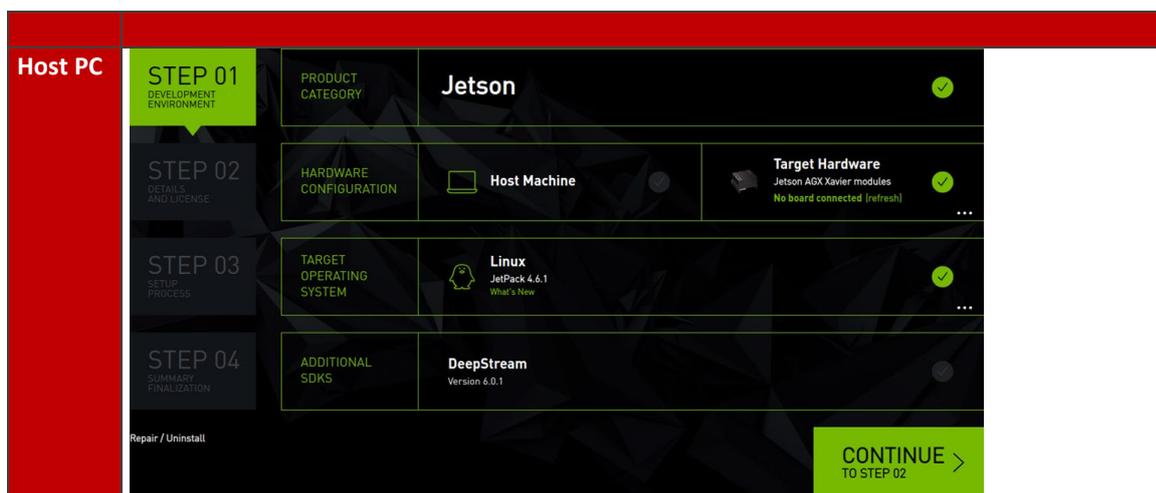
The alternative flashing guide is intended to be used if you encounter problems with the recommended guide. This guide is more general and should also work with boards from other vendors and requires a few more steps. In this guide you will download the core operating system from NVIDIA with the SDK manager and only download the files that need to be changed for your carrier board from Auvidea.

3.1 Before you start

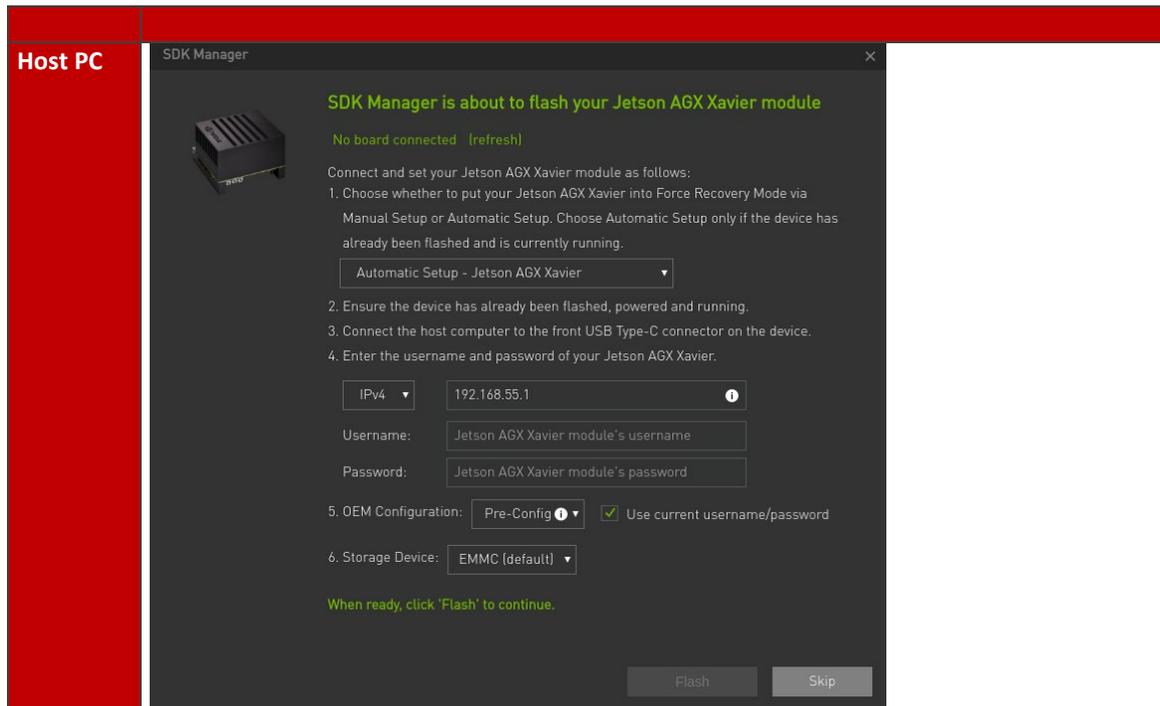
- Please make sure to use a Linux host PC with Ubuntu 18.04 LTS or Ubuntu 20.04 LTS (other versions may work but some caused problems in the past!) operating system. Please use a native setup (no virtual machine). This host PC should have a high bandwidth internet connection for the download of 2GB+ installation file in the following steps.
- You should make sure that Python 2.x is installed as it is required for the NVIDIA SDK Manager. You can check your installation in the command line with: `python --version` example return: `Python 2.7.17`. If you do not get a return or error you do not have Python 2.x installed. Please use: `sudo apt install python` to install Python 2.x to your system)
- You will also need a high-quality standard USB 2.0 Type A to micro-USB 2.0 cable.
- If you are using a AGX Orin please see the *Flashing the AGX Orin* section, as additional steps are required.

3.2 Install and configure NVIDIA SDK manager

- 1) Download and install the Nvidia SDK manager for Ubuntu
<https://developer.nvidia.com/nvidia-sdk-manager>
- 2) Install Jetpack 4.6 and up for your Jetson module (AGX Xavier)



- 3) Skip the flashing process after installation. This step is necessary to set up your file system and contents of you Host PC. Do not flash with this configuration! Specific steps need to be performed to enable all functionality of you carrier board as described in the following steps.



3.3 Download installation files from Auvideo

The following steps are necessary to enable all features of the Auvideo carrier board. It is possible to skip these steps and flash your system without them. But then you will only be able to use basic functionality like 1080p HDMI, some USB ports, Ethernet. Without these steps 4k HDMI, PCIe ports, some USB ports, CSI, CAN,... will not work, we strongly recommend to follow it if possible.

- 4) Download our latest firmware for your carrier board

Link	https://auvideo.eu/firmware/		
Auvideo webpage (example)	Date	Product	Version Description
	Dec 2021	X200/X220/X220-LC/X221/X221-LC/X400 (223 MB) firmware for Jetpack 4.6 (L4T 32.6.1) (sources, kernel, patches and readme)	supports: X200, X220, X220-LC, X221, X221-LC, X400 carrier boards 4.0 - 2x USB 3.0 - 2x HDMI - 2x GbE (native and i210) - M.2 SSD (NVME)

- 5) Extract the downloaded tar ball from our website

Host PC	<pre>tar xvf <your_downloaded_tar>.tar.bz2</pre>
----------------	--

- 6) Change directory into the extracted files and extract the "kernel_out.tar.bz2"

Host PC	<pre>cd <your_extracted_downloaded_tar> tar xvf kernel_out.tar.bz2</pre>
----------------	--

- 7) Copy the extracted kernel_out folder into your jetpack 4.6 (or up) L4t (Linux for Tegra) folder.
Usual File path:
"/home/<YOUR_USERNAME>/nvidia/nvidia_sdk/JetPack_<Jetpack_version>_Linux_<Jetson_module>/Linux_for_Tegra"

Host PC	<pre>rsync -axHAWX --numeric-ids - info=progress2./kernel_out/<Jetpack_L4T_folder> (Modify for your version/module needs)</pre>
----------------	---

3.4 Flashing the AGX Orin

If you want to flash a AGX Orin the following steps should be followed additionally to the flashing guide from above. Other modules (like AGX Xavier, Nano, Nano NX, ...) do not need those additional steps and you should continue with the next step.

Future installation packages from Auvidea should already include the following changes. In that case you can use the following steps as a check or implement them yourself when skipping the Auvidea installation files.

3.4.1 AGX Orin Jetpack compatibility

When flashing a AGX Orin production module the Jetpack version 5.0 GA must be used. The Jetpack versions 5.0 DP and 5.0.1 DP do not work!

3.4.2 Necessary changes

Independent of the used AGX Orin version you must make the following changes.

These changes must be done in the filesystem of the Host PC before flashing it to the AGX Orin module.

Usual File path:

"/home/<YOUR_USERNAME>/nvidia/nvidia_sdk/JetPack_<Jetpack_version>_Linux_<Jetson_module>/Linux_for_Tegra/bootloader/tegra234-mb2-bct-common.dtsi"

	In tegra234-mb2-bct-common.dtsi
Replace	<code>cvb_eeeprom_read_size = <0x100></code>
Replace with	<code>cvb_eeeprom_read_size = <0x0></code>

Also, the ODMDATA must be changed.

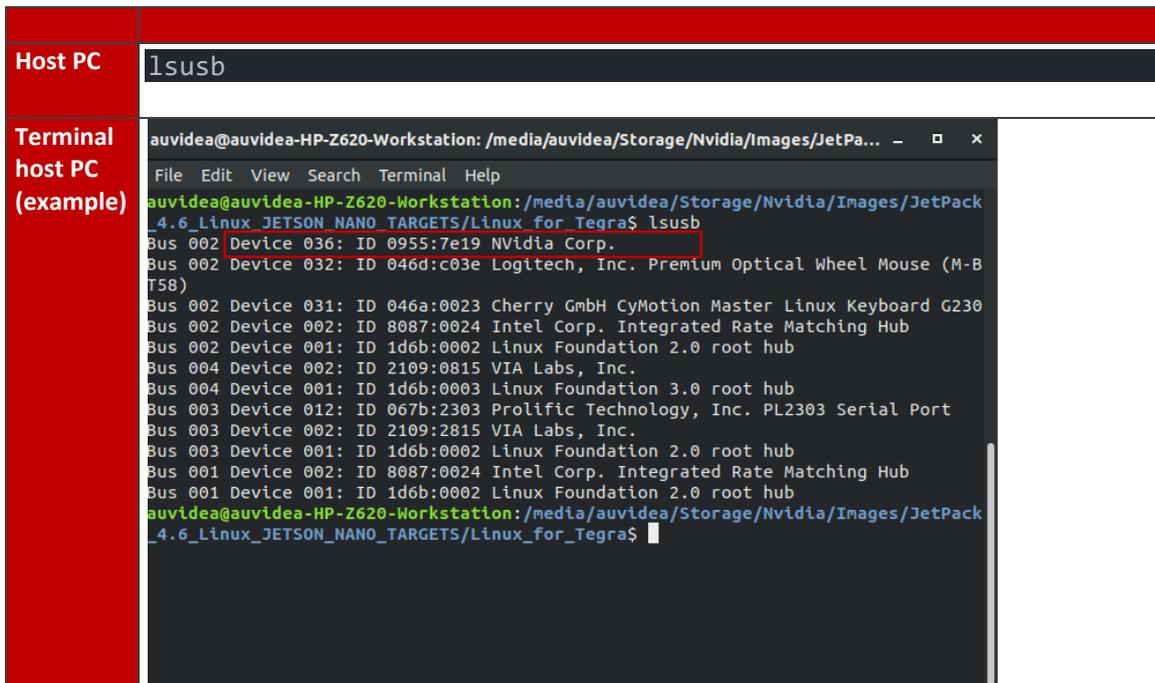
Usual File path:

"/home/<YOUR_USERNAME>/nvidia/nvidia_sdk/JetPack_<Jetpack_version>_Linux_<Jetson_module>/Linux_for_Tegra/p3701.conf.common "

	In p3701.conf.common
Set	<code>ODMDATA="gbe-uphy-config-0,hsstp-lane-map-3,nvhs-uphy-config-0,hsio-uphy-config-16"</code>

3.5 Flashing of system

- 8) Connect a USB 2 micro-USB cable to the Jetson bevor powering it up
- 9) After connecting to the host PC power up your Jetson. This will put the system in to flashing mode (also called force recovery mode) with a connected Host PC. In some cases, you will have to manual press the force recovery mode button or short pins to enter force recovery mode. For example, with the JN30D you will need to short Pins 7 & 8 of J32 to enter force recovery. A simple way is to use tweezers or jumpers. Please see the Technical Reference Manual for a detailed pin description.
- 10) Check that the connection is established with the lsusb command. You should find one entry with Nvidia Corp. as highlighted below.



- 11) Open your terminal in the <Jetpакc_L4T_folder> folder if you are not already in it.
- 12) You can now flash your system using the following command



Module name:	<your_module>:
Jetson Nano	jetson-nano-emmc
Jetson AGX-Xavier	jetson-xavier
Jetson Xavier NX	jetson-xavier-nx-devkit-emmc
Jetson TX2 NX	jetson-xavier-nx-devkit-tx2-nx
...	...

(If you are following the native SSD boot flashing guide. The following command needs to be changed as described in the chapter.)

- 13) Please connect a monitor to the system. After the flashing process has completed the system should automatically boot and show the Ubuntu desktop.



You now have a functioning system ready for your needs.

14) Please also see [2.5 Maintaining functionality](#)

3.6 Installing additional NVIDIA SDK components

14) Now you can install additional NVIDIA SDK components. Please connect the system to the internet. Use apt-get to install the components. If this fails, please check the Internet connection of the system.

Jetson system	<code>sudo apt-get update && sudo apt-get install nvidia-jetpack</code>
Terminal Jetson system example	<pre>test@test-desktop: ~ test@test-desktop:~\$ sudo apt-get update && sudo apt-get install nvidia-jetpack [sudo] password for test: Hit:1 http://ports.ubuntu.com/ubuntu-ports bionic InRelease Hit:2 http://ports.ubuntu.com/ubuntu-ports bionic-updates InRelease Hit:3 http://ports.ubuntu.com/ubuntu-ports bionic-backports InRelease Hit:4 http://ports.ubuntu.com/ubuntu-ports bionic-security InRelease Hit:5 https://repo.download.nvidia.com/jetson/common r32.6 InRelease Get:6 https://repo.download.nvidia.com/jetson/t210 r32.6 InRelease [2.547 B] Fetched 2.547 B in 2s (1.554 B/s) Reading package lists... Done Reading package lists... Done Building dependency tree Reading state information... Done The following packages were automatically installed and are no longer required: apt-clone archdetect-deb bogl-bterm busybox-static cryptsetup-bin dpkg-repack gir1.2-timezonemap-1.0 gir1.2-xkl-1.0 grub-common kde-window-manager kinit kio kpackagetool5 kwayland-data kwinn-common kwin-data kwinn-x11 libdebian-installer4 libkdecorations2-5v5 libkdecorations2private5v5 libkf5activities5 libkf5attica5 libkf5completion-data libkf5completion5 libkf5declarative-data libkf5declarative5 libkf5doctools5 libkf5globalaccel-data libkf5globalaccel5 libkf5globalaccelprivate5 libkf5idle5 libkf5jobwidgets-data libkf5jobwidgets5 libkf5kcmutils-data libkf5kcmutils5 libkf5kiocore5 libkf5kiontlm5 libkf5kiowidgets5 libkf5newstuff-data libkf5newstuff5</pre>

SECTION 4 Preliminary Orin nano & Orin NX flashing guide

This guide is intended to be used with the Orin modules Orin nano & Orin NX series. As the integration of the flashing capabilities are not yet integrated into the SDK manager from Nvidia. This guide is meant to bridge the time until the SDK manager can be used. Not that not all functionalities of your carrier board like the JNX42 or JNX45 are supported yet (like some USB ports may not work). This will be fixed in a future release following soon.

4.1 Before you start

- Please make sure to use a Linux host PC with Ubuntu 18.04 LTS or Ubuntu 20.04 LTS (other versions may work but some caused problems in the past!) operating system. Please use a native setup (no virtual machine). This Host PC should have a high bandwidth internet connection for the download of 3GByte+ installation file in the following steps.
- You will also need a high-quality standard USB 2.0 Type A to micro-USB 2.0 cable.
- If you are using a AGX Orin please use the Advanced flashing guide.

4.2 Download installation file from NVIDIA

- 11) Download the installation file for your selected JetPack from the NVIDIA archive into a working directory of your choice. The following table shows step by step where to download the appropriate files.

Link	https://developer.nvidia.com/embedded/jetpack-archive
NVIDIA webpage (JetPack)	<p style="text-align: center;">JetPack Archive</p> <hr/> <p style="font-size: small;">This page includes access to previously released versions of JetPack. The latest version of JetPack is always available under the main NVIDIA JetPack product page.</p> <ul style="list-style-type: none"> • JetPack 5.1 <ul style="list-style-type: none"> ◦ Jetson AGX Orin Developer Kit, Jetson Xavier NX series, Jetson AGX Xavier Series, [L4T 35.2.1] • JetPack 5.0.2 <ul style="list-style-type: none"> ◦ Jetson AGX Orin Developer Kit, Jetson Xavier NX series, Jetson AGX Xavier Series, [L4T 35.1] • JetPack 5.0.1 Developer Preview <ul style="list-style-type: none"> ◦ Jetson AGX Orin Developer Kit, Jetson Xavier NX series, Jetson AGX Xavier Series, [L4T 34.1.1] • JetPack 5.0 Developer Preview <ul style="list-style-type: none"> ◦ Jetson AGX Orin Developer Kit, Jetson Xavier NX series, Jetson AGX Xavier Series, [L4T 34.1]
NVIDIA webpage select Jetson Linux	<p style="text-align: center;">KEY FEATURES IN JETPACK</p> <hr/> <p>Jetson Linux NVIDIA Jetson Linux 35.2.1 provides the Linux Kernel 5.10, UEFI based bootloader, Ubuntu 20.04 based root file system, NVIDIA drivers, necessary firmwares, toolchain and more.</p> <p>JetPack 5.1 includes Jetson Linux 35.2.1 which adds following highlights: (Please refer to release notes for additional details)</p> <ul style="list-style-type: none"> • Adds Support for Jetson AGX Orin NX 16GB production module • Security <ul style="list-style-type: none"> ◦ UEFI Secure Boot¹ ◦ Secure Storage in OP-TEE using RPMB (Replay Protected Memory Block) ◦ Memory Encryption on Jetson Orin • Over The Air Updates:

Download Driver Package (BSP) & Sample Root Filesystem	Downloads and Links	
	Jetson Orin modules and developer kit	Jetson Xavier modules and developer kits
DRIVERS	→ Driver Package (BSP)	
	→ Sample Root Filesystem	
SOURCES	Driver Package (BSP) Sources	
	Sample Root Filesystem Sources	

12) Open a terminal window (CTRL + ALT + T) on your Linux host PC and navigate to your download location.

```
Host PC cd <path_to_downloaded_tar>
```

13) Extract the tar.gz file you just downloaded.

```
Host PC tar xf Jetson_Linux_R35.2.1_aarch64.tbz2
```

14) Change directory to the extracted rootfs folder.

```
Host PC cd Linux_for_Tegra/rootfs/
```

15) Extract the rootfs.

```
Host PC sudo tar xpf ../../Tegra_Linux_Sample-Root-Filesystem_R35.2.1_aarch64.tbz2
```

16) Change directory to the extracted Linux_for_Tegra folder.

```
Host PC cd Linux_for_Tegra
```

17) Install the flashing prerequisites.

```
Host PC sudo ./tools/l4t_flash_prerequisites.sh
```

18) Apply the binaries.

```
Host PC sudo ./apply_binaries.sh
```

19) On Auvideo carrier boards you have to adjust the eeprom size:
change "cvb_eeprom_read_size = <0x100>," in "<working

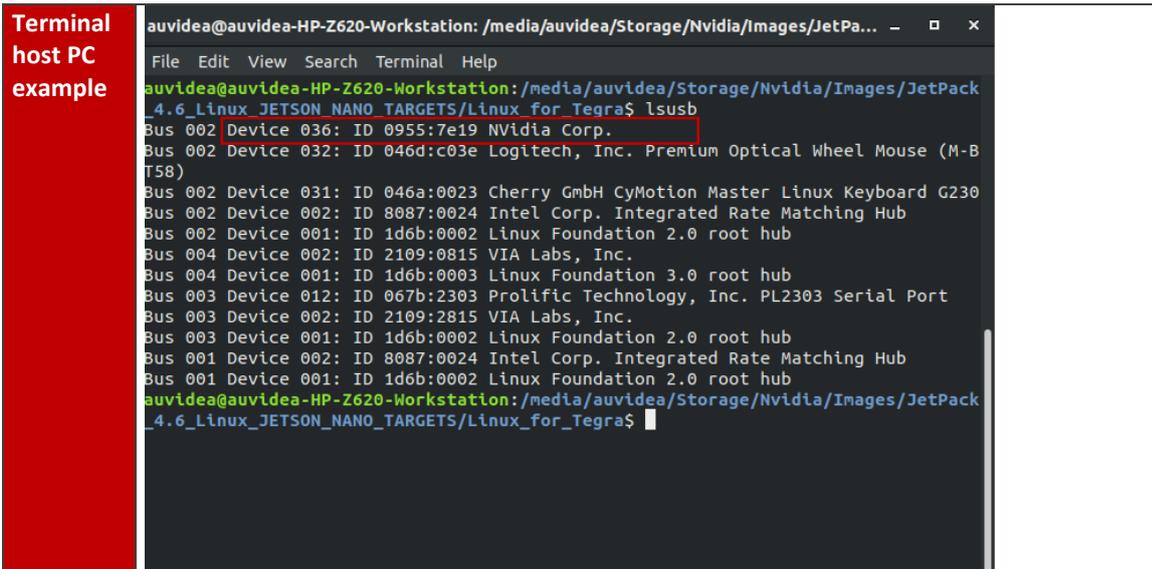
directory>/Linux_for_Tegra/bootloader/t186ref/BCT/tegra234-mb2-bct-misc-p3767-0000.dts" to "cvb_eeeprom_read_size = <0x0>"

Before	<pre> /dts-v1/; #include "tegra234-mb2-bct-common.dtsi" / { mb2-misc { eeeprom { cvm_eeeprom_i2c_instance = <0>; cvm_eeeprom_i2c_slave_address = <0xa0>; cvm_eeeprom_read_size = <0x100>; cvb_eeeprom_i2c_instance = <0x0>; cvb_eeeprom_i2c_slave_address = <0xae>; cvb_eeeprom_read_size = <0x100>; ← }; }; }; </pre>
After	<pre> /dts-v1/; #include "tegra234-mb2-bct-common.dtsi" / { mb2-misc { eeeprom { cvm_eeeprom_i2c_instance = <0>; cvm_eeeprom_i2c_slave_address = <0xa0>; cvm_eeeprom_read_size = <0x100>; cvb_eeeprom_i2c_instance = <0x0>; cvb_eeeprom_i2c_slave_address = <0xae>; cvb_eeeprom_read_size = <0x0>; ← }; }; }; </pre>

4.3 Connect carrier board to host PC

- 20) Connect the system to the Linux host PC. Please use a USB 2.0 cable (micro-USB on the carrier board).
- 21) Connect your system to power (please see specifications to apply the right voltage for your system when using your own power supply). The system will start automatically and detect the host PC and enter the flashing state (also called force recovery mode).
In some cases, you will have manual press or short pins before connecting to your power source to enter force recovery mode. For example, with the JN30D you will need to short Pins 7 & 8 of J32 to enter force recovery. A simple way is to use tweezers. Please see the Technical Reference Manual for a detailed pin description. You can also see the FAQs in this document for further informations.
- 22) Check that the connection is established with the "lsusb" command. You should find one entry with "Nvidia Corp." as highlighted below.

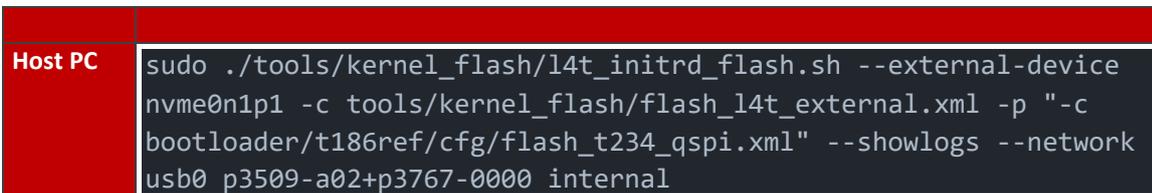
Host PC	<pre>lsusb</pre>
----------------	------------------



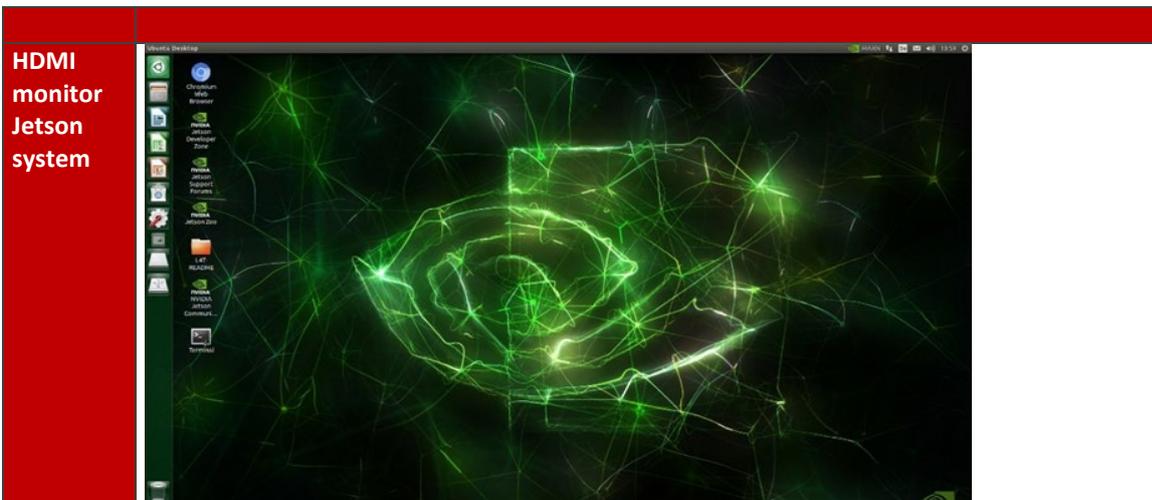
Your system is now connected properly and ready to be flashed

4.4 Flashing of system

23) Use this flash command for your system.



24) Please connect a monitor to the system. After the flashing process has completed the system should automatically boot and show the Ubuntu desktop.



You now have a functioning system ready for your needs.

4.5 Maintaining functionality

After successful flashing please be aware that if you use “~~sudo apt-get upgrade~~” some of the applied changes performed in this guide (Bootloader, Kernels, DTBs, ...) will be overwritten by standard Nvidia packages. This will most probably brick some board functionality and some interfaces may not work correctly anymore.

When overriding the Auvideo configurations only basic functionality like 1080p HDMI, some USB ports, Ethernet could work

Without the correct configurations and packages 4k HDMI, PCIe ports, some USB ports, CSI, CAN, ... will not work.

If you know what you are doing you can exclude the Nvidia Debian packages from the Nvidia sources.list from being updated. This prevents overriding relevant files and changes.

If you are unsure, we recommend **NOT UPGRADE YOUR SYSTEM.**

In the case you accidental upgraded your system and some board functionality stopped working correctly we recommend to reflash your system according to this guide.

SECTION 5 SSD-Boot flashing guide (experienced users)

This guide describes how to set up your system to boot partially or fully from a M.2 SSD depending on your module choice.

This guide is intended for experienced users only.

5.1 Partial SSD boot for Jetson Nano, NX, TX2 NX

The Jetson Nano, NX and TX2 NX modules do not support native boot purely from the SSD. To still benefit from the SSD speed and size, this guide describes how to move your RootFS to your SSD.

You can also use the Partial SSD boot flashing guide for the Jetson Xavier/Orin series, but we recommend the Native SSD boot guide for Jetson Xavier/Orin systems.

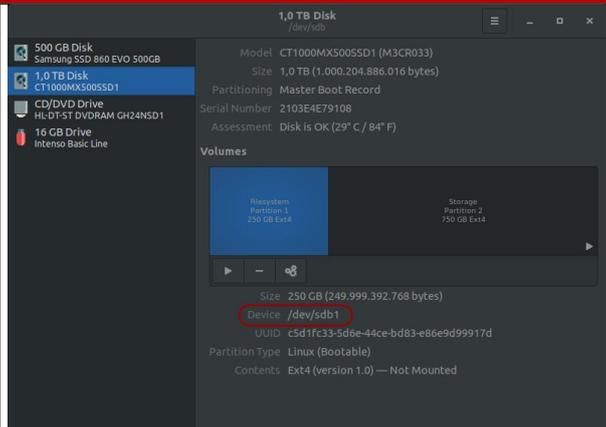
Note: If you run in to problems, there is also another resource from JetsonHacks: “Jetson Xavier NX – Run from SSD” which you can find on google.

5.1.1 Before you start

- Please make sure your system is set up and running according to our simple or advanced flashing guide from SECTION 2 or SECTION 3.

5.1.2 Locate and identify storage device

1. The first thing you need to do is to identify the storage device you are intending to use.

Jetson system	Ubuntu	Note
		<p>With the <i>disc</i> program you can find your device as shown in the Picture. In this example: <code>/dev/sdb1</code></p>

Make sure your intended SSD storage device does not hold valuable data as the following steps will format this device.

SSD-Storage device:

When working with NVME SSD your device name should look like this: `/dev/nvme0n1p1`
 The ending *p1* stands for partition one. In the following steps you must remove the partition information from the device path. See the following example:

`<YOUR_STORAGE_DEVICE> = /dev/nvme0n1`

USB-Storage device:

When working with external USB storage your device name should look like this: `/dev/sdb1`

The ending *1* stands for partition one. In the following steps you must remove the partition information from the device path. See the following example:

<YOUR_STORAGE_DEVICE> = /dev/sdb

5.1.3 Set up RootFS on SSD

Please execute the following steps to copy your RootFS to your SSD-Boot storage.

2. Format the storage device

Jetson system	<pre>sudo parted <YOUR_STORAGE_DEVICE> mklabel gpt</pre>
----------------------	--

3. Create the RootFS partition

Jetson system	<pre>sudo parted <YOUR_STORAGE_DEVICE> mkpart APP 0GB <YOUR_ROOTFS_SIZE></pre>
----------------------	--

4. Create filesystem

Jetson system	<pre>sudo mkfs.ext4 <YOUR_STORAGE_DEVICE>1</pre>
----------------------	--

5. Copy the existing RootFS to the storage device

Jetson system	<pre>sudo mount <YOUR_STORAGE_DEVICE> /mnt sudo rsync -axHAX --numeric-ids --info=progress2 -- exclude={"/dev/","/proc/","/sys/","/tmp/","/run/","/mnt/","/media /*","/lost+found"} / /mnt/</pre>
----------------------	---

5.1.4 Switch boot device to SSD

You need to change your root target in `exlinux.conf` to the SSD. This is necessary so that the Operating system knows where to find the system files.

6. Open `exlinux.conf`

Jetson system	<pre>sudo nano /boot/exlinux/exlinux.conf</pre>
----------------------	---

7. Modify the `exlinux.conf` by changing the root path

Jetson system	<pre>LABEL primary MENU LABEL primary kernel LINUX /boot/Image INITRD /boot/initrd</pre>
----------------------	--

```
APPEND ${cbootargs} quiet root=<YOUR_STORAGE_DEVICE>1 rw
rootwait rootfstype=ext4 console=ttyTCU0,115200n8 console=tty0
fbcon=map:0 net.ifnames=0
```

- After a reboot your system will start from the SSD

5.1.5 Validate

- You can validate the SSD boot by using the following test

Jetson system	<pre>test@test-desktop:~\$ df / Filesystem 1K-blocks Used Available Use% Mounted on /dev/nvme0n1p1 122507912 14618752 102744904 13% /</pre>
----------------------	--

When <Your_Storage_device> shows up to what you have set it, you successfully moved your RootFS and booted from SSD.

5.2 Native SSD boot for Jetson AGX Xavier, AGX Orin, Xavier NX series

This guide is only applicable for the Jetson AGX Xavier, Xavier NX and AGX Orin series. These series support direct boot from NVME SSD.

Note: If you run in to problems, there is also another resource from JetsonHacks: “Jetson Xavier NX – Run from SSD” which you can find on google.

5.2.1 Before you start

- Please follow the steps described in SECTION 3 Advanced flashing guide (experienced users). The same steps apply if you want your system to boot from the SSD. You only must change one command from the Advanced flashing guide (experienced users) to apply the native SSD boot.

5.2.2 Flash system

- Please change the command as displayed below. The crossed-out command is just for reference and must not be executed!

Host PC	<pre>sudo ./flash <your_module> mmcblk0p1 sudo ./nvsdkmanager_flash.sh --storage nvme0n1p1</pre>
----------------	---

SECTION 6 Frequently Asked Questions

- Why is my X221 not going into force recovery?
 - If your system is not going into force recovery, please contact our support as you may have a system with a mismatched firmware.
- Why is my system not entering the force recovery state?
 - Most of our carrier boards are designed to enter force recovery mode when they detect a Host PC. This detection only works one time automatically after the system was connected to its power supply. We recommend unplugging your system before connecting to a Host PC and plugging it back in to power after connecting.
 - If your system still does not enter force recovery you may have to press the force recovery button or short the respective pins before connecting to power (please see the Technical Reference Manual for a detailed pinout description).
 - If you cannot disconnect your system from power, it is also possible to enter force recovery via a button sequence.
 - Press/jumper “force recovery” button/pins
 - Press/jumper “reset” button/pins
 - Release/disconnect “reset” button/pins
 - Release/disconnect “force recovery” button/pins a few seconds later then the “reset” button/pins



SECTION 7 Disclaimer

Thank you for reading this manual. If you have found any typos or errors in this document, please let us know.

This is the preliminary version of this data sheet. Please treat all specifications with caution as there may be any typos or errors.

The Auvidea Team

Copyright notice

SECTION 8 Trademarks

NVIDIA, the NVIDIA logo, CUDA, Jetson, Maxwell, Tegra, Nano and VisionWorks are registered trademarks and/or trademarks of NVIDIA Corporation in the United States and other countries. Other company and product names may be trademarks of the respective companies with which they are associated.

© Auvidea GmbH 2022

All Rights Reserved

No part of this document or any of its contents may be reproduced, copied, modified or adapted, without the prior written consent of the author, unless otherwise indicated for stand-alone materials.

You may share this document by any of the following means: this PDF file may be distributed freely if no changes or modifications to the document are made.

For any other mode of sharing, please contact the author at the email below. info@auvidea.com

Commercial use and distribution of the contents of this document is not allowed without express and prior written consent of Auvidea GmbH.