Lanner

Embedded Computing Platform

Hardware Platforms for Intelligent Edge Computing

EAI-I131 and Jetson Linux Developer Guide

Version: 1.0

Date of Release: 2024-11-04

About this Document

This manual describes the overview of the various functionalities of this product, and the information you need to get it ready for operation. It is intended for those who are:

- responsible for installing, administering and troubleshooting this system or Information Technology professionals.
- assumed to be qualified in the servicing of computer equipment, such as professional system integrators, or service personnel and technicians.

The latest version of this document can be found on Lanner's official website, available either through the product page or through the Lanner Download Center page with a login account and password.

Conventions & Icons

The icons are used in the manual to serve as an indication of interest topics or important messages.

Icon	Usage
 **Note or Information**	This mark indicates that there is something you should pay special attention to while using the product.
 **Warning or Important**	This mark indicates that there is a caution or warning and it is something that could damage your property or product.

Online Resources

To obtain additional documentation resources and software updates for your system, please visit the Lanner Download Center. As certain categories of documents are only available to users who are logged in, please be registered for a Lanner Account at http://www.lannerinc.com/ to access published documents and downloadable resources.

Technical Support

In addition to contacting your distributor or sales representative, you could submit a request at our Lanner Technical Support and fill in a support ticket to our technical support department.

Documentation Feedback

Your feedback is valuable to us, as it will help us continue to provide you with more accurate and relevant documentation. To provide any feedback, comments or to report an error, please email contact@lannerinc.com. Thank you for your time.

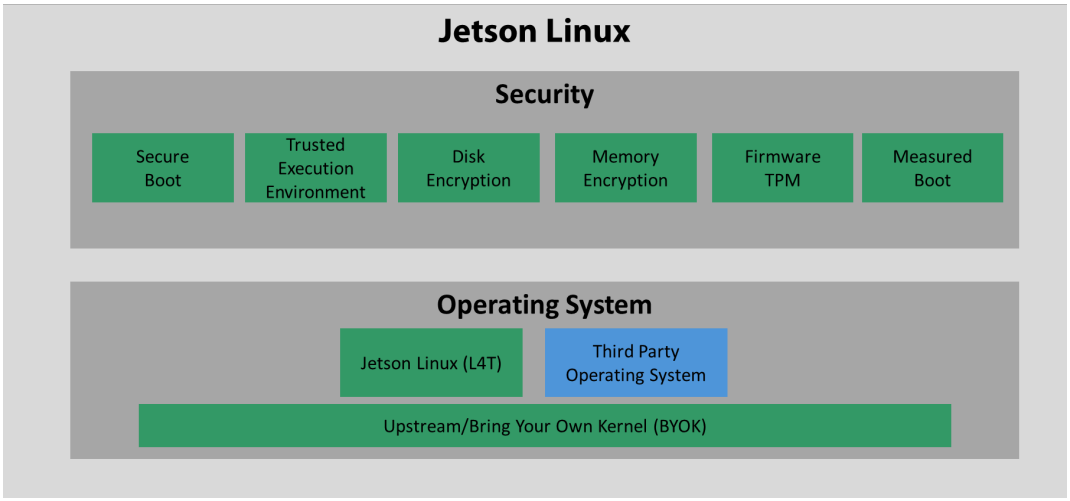
Table of Contents

CHAPTER 1: INTRODUCTION

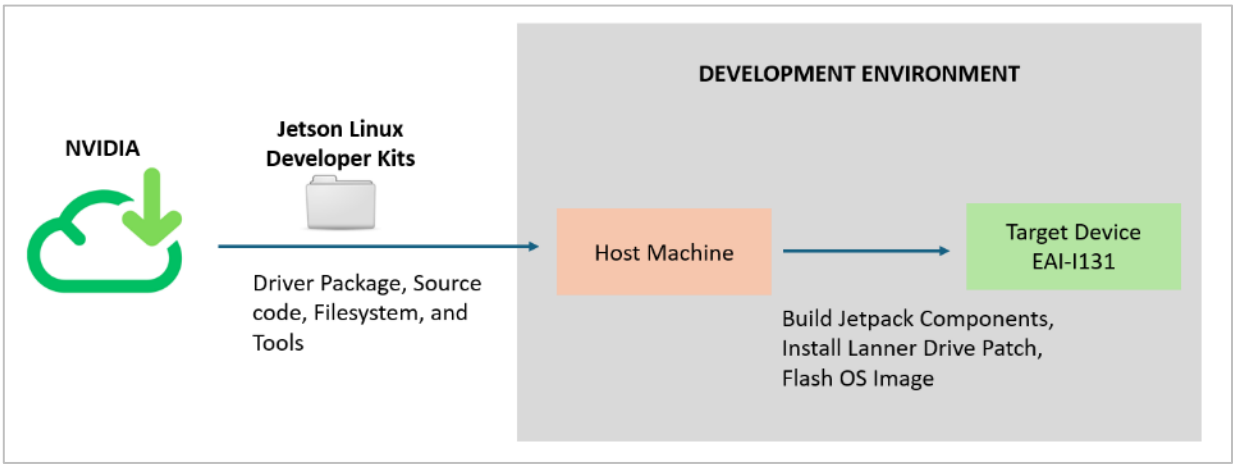
The EAI-I131, designed for industrial-grade AI inference, integrates the NVIDIA® Jetson Orin NX/Nano platform, delivering up to 100 TOPS of AI processing power—ideal for advanced robotics and 5G edge computing. Setting up NVIDIA® Jetson Linux on the EAI-I131 allows you to harness one of the most powerful platforms for AI at the edge.

This user guide details the setup and installation processes for the EAI-I131 AI appliance, with a focus on its integration with NVIDIA Jetson Linux. The guide addresses two key scenarios:

1. When the user intends to install the NVIDIA Accelerated Toolkit Package (ATP) and the associated application SDK to utilize pre-built software and libraries optimized for AI workloads.



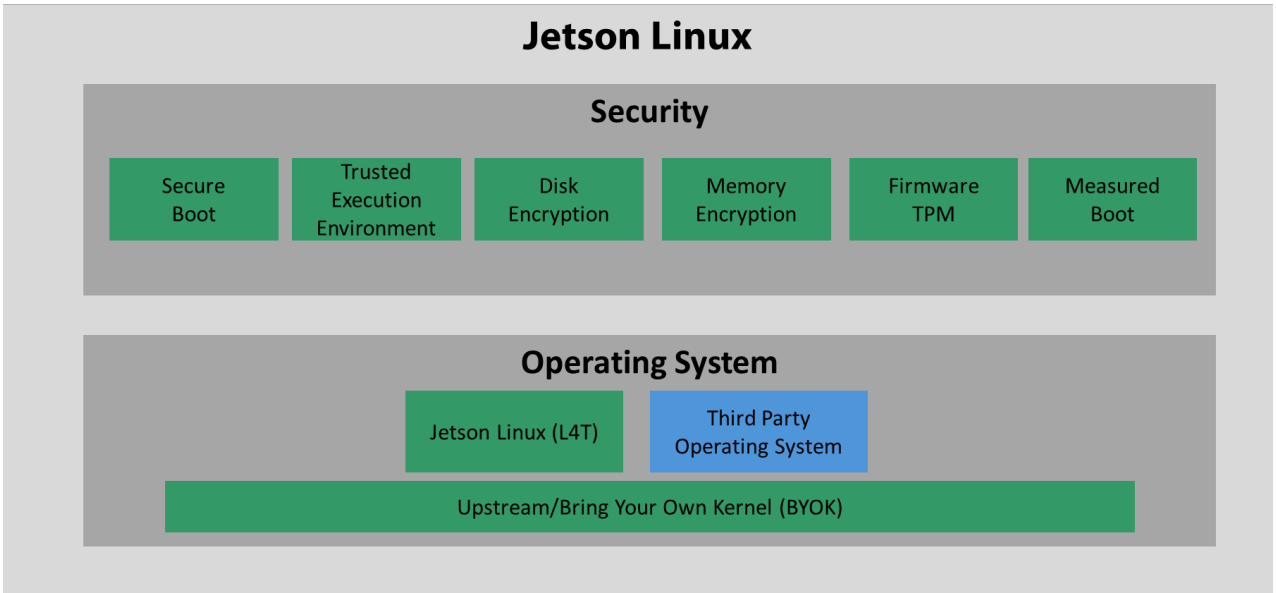
2. When the user opts to create a custom SDK development environment, enabling tailored configurations and specific use case adaptations.



Each scenario is designed to provide detailed, step-by-step instructions to ensure a seamless integration with the EAI-I131, maximizing its performance and utility in various AI-driven applications.

CHAPTER 2: NVIDIA ACCELERATED TOOLKIT PACKAGE (ATP)

When the user intends to install the NVIDIA Accelerated Toolkit Package (ATP) and the associated application SDK, they are opting for a streamlined approach to deploying AI solutions on the EAI-I131. By utilizing the ATP and application SDK, users can benefit from a robust set of tools designed to accelerate development, simplify deployment, and optimize performance for a wide range of AI-driven applications. This approach is ideal for those looking to minimize setup time while maximizing the efficiency and capability of their AI inference tasks.



Jetson Software Stack

For more information and tutorials on installing Frameworks for Jetson, go to NVIDIA Optimized Frameworks Docs Hub.

CHAPTER 3: DEVELOPMENT ENVIRONMENT SETUP & INSTALLATION

The NVIDIA Jetson Linux, which includes the JetPack Driver Package, will provide the necessary Linux Kernel, UEFI bootloader, and NVIDIA drivers to develop and deploy your AI applications. In this section, we'll guide you through the process of flashing the software image to the Jetson module and ensuring your system is ready for AI deployment.

3.1 Pre-Installation Requirements

3.1.1 Hardware Requirements

1x Host Machine

1x Console RJ45 to USB Cable OR (Mini-)VGA to USB Cable (Depends on Host Machine)

1x Burn-in USB to USB Cable (Included in the accessories pack of EAI-I131 Appliance)

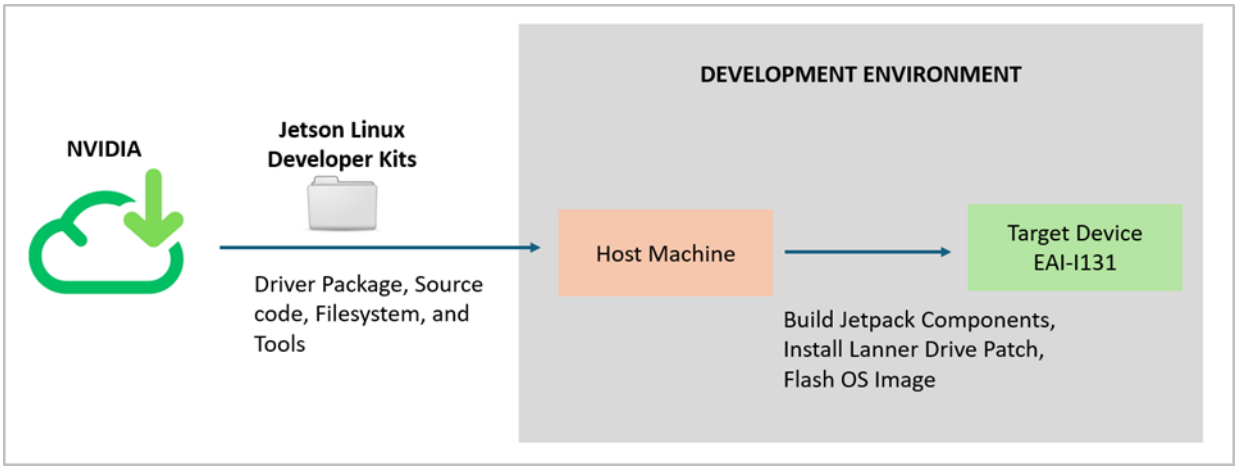
Host Machine Specification Requirement

CPU: 8C or above

Memory: 8GB or above

Storage: 128GB or above

OS: Ubuntu 22.04.4 LTS



3.1.2 Jetson Linux Source Code Download Link

Go to Nvidia website: https://developer.nvidia.com/embedded/jetson-linux-r3550 and Download:

a.) Drivers - Driver Package (BSP)

b.) Sources - Driver Package (BSP) Sources

c.) Sources - Sample Root Filesystem Sources

d.) Tools - Bootlin Toolchain gcc 9.3

3.1.3 Lanner Patch Driver Link

Version	File Name	Link
JP5.1.3	EAI_I131_Pinmux_verJP5.1.3.tgz	Link

Go to the link above and download: **EAI_I131_Pinmux_verJP5.1.3.tgz**

NOTE: This patch is for JetPack 5.1.3 only. For additional versions, please contact Lanner Technical Support.

3.2 Host Machine Installation

Compile and download Jetson Linux source code and Lanner Patch Driver to Host Machine.

3.2.1 Depending on Host Machine, use a Console Cable or VGA Cable to connect the Host Machine to your Laptop computer.

[Note: Image above is for reference only.]

3.2.2 First, we need to make sure the Host Machine has internet access, in order to download the packages and components. Open a new Terminal tab, like TeraTerm or PuTTY, and type in command as below: Please execute all commands and actions with root privileges.

```
# ping www.google.com -c 5
PING www.google.com (142.251.43.4) 56(84) bytes of data.
64 bytes from tsa03s08-in-f4.1e100.net (142.251.43.4): icmp_seq=1 ttl=56 time=3.04 ms
64 bytes from tsa03s08-in-f4.1e100.net (142.251.43.4): icmp_seq=2 ttl=56 time=2.02 ms
64 bytes from tsa03s08-in-f4.1e100.net (142.251.43.4): icmp_seq=3 ttl=56 time=1.82 ms
64 bytes from tsa03s08-in-f4.1e100.net (142.251.43.4): icmp_seq=4 ttl=56 time=1.91 ms
64 bytes from tsa03s08-in-f4.1e100.net (142.251.43.4): icmp_seq=5 ttl=56 time=1.90 ms

--- www.google.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4007ms
rtt min/avg/max/mdev = 1.817/2.136/3.039/0.455 ms
```

3.2.3 Next, we need to install the Linux Dev-Tool to build Development Environment.

Input the commands below:

```
# apt-get update
# apt-get install ssh qemu-user-static make build-essential libncurses-dev python
# apt-get install flex libssl-dev libxml2-utils liblz4-tool sshpass abootimg nfs-kernel-server bison
# apt-get install git-core
```

3.2.4 Depending on the JetPack version required, please go to the link [3.1.3 Lanner Patch Driver Link] and download the corresponding Pinmux Patch file, and copy with the extracted folder under root path. Then in the Terminal tab, input the commands below:

```
# cd /root/
# ls /root/eai_i131_orin_JetPack513_20240521_v01 board defconfig_513 eai_i131_513_v1_2.patch
readme.txt tool
```

3.2.5 Create a folder and download Jetson Linux Development Kits.

Input the commands below:

```
# ls /root/jetpack_513_source_code/aarch64--glibc--stable-final.tar.gz public_sources.tbz2
Jetson_Linux_R35.5.0_aarch64.tbz2
Tegra_Linux_Sample-Root-Filesystem_R35.5.0_aarch64.tbz2
```

3.2.6 In order to create the SDK Environment, we first need to decompress the source code.

Input the commands below:

```
# sudo su -
# mkdir -p /home/L5T_513/gcc
# cd /home/L5T_513
# cp /root/jetpack_513_source_code/* ./
# sync
# tar xf Jetson_Linux_R35.5.0_aarch64.tbz2
# tar xf public_sources.tbz2
# tar xf aarch64--glibc--stable-final.tar.gz -C ./gcc
# tar xf Tegra_Linux_Sample-Root-Filesystem_R35.5.0_aarch64.tbz2 -C Linux_for_Tegra/rootfs
```

3.2.7 Then, download and install the package of root filesystem.

Input the commands below:

```
# cd /home/L5T_513/Linux_for_Tegra
# ./apply_binaries.sh
Using rootfs directory of: /home/L5T_513/Linux_for_Tegra/rootfs
Installing extlinux.conf into /boot/extlinux in target rootfs
/home/L5T_513/Linux_for_Tegra/nv_tegra/nv-apply-debs.sh
Root file system directory is /home/L5T_513/Linux_for_Tegra/rootfs
Copying public debian packages to rootfs
Skipping installation of nvidia-l4t-dgpu-aptsource_35.5.0-20240219203809_arm64.deb .....
.
.
.
L4T BSP package installation completed!
Disabling NetworkManager-wait-online.service
Disable the ondemand service by changing the runlevels to 'K'
Success!
#
```

3.2.8 Next, download Linux Utility Package (for Host Machine).

Input the commands below:

```
# ./tools/l4t_flash_prerequisites.sh
```

3.2.9 Then, download Kernel Source Code from git.

Input the commands below:

```
# cd /home/L6T_rel/Linux_for_Tegra/source
# ./source_sync.sh -t jetson_35.5
Downloading default kernel/kernel-5.10 source...
Cloning into '/home/L5T_513/Linux_for_Tegra/sources/kernel/kernel-5.10'...
remote: Enumerating objects: 7914323, done.
remote: Counting objects: 100% (7914323/7914323), done.
remote: Compressing objects: 100% (1333436/1333436), done.
remote: Total 7914323 (delta 6543970), reused 7901780 (delta 6531427), pack-reused 0
Receiving objects: 100% (7914323/7914323), 1.73 GiB | 3.30 MiB/s, done.
Resolving deltas: 100% (6543970/6543970), done.
```

```
The default kernel/kernel-5.10 source is downloaded in:
/home/L5T_513/Linux_for_Tegra/sources/kernel/kernel-5.10
Syncing up with tag jetson_35.5...
Updating files: 100% (72047/72047), done.
Switched to a new branch 'mybranch_2024-05-10-1715311316'
/home/L5T_513/Linux_for_Tegra/sources/kernel/kernel-5.10 source sync'ed to tag jetson_35.5 successfully!
```

```
.
.
.
Resolving deltas: 100% (24/24), done.
The default tegra/v4l2-src/v4l2_libs source is downloaded in:
/home/L5T_513/Linux_for_Tegra/sources/tegra/v4l2-src/v4l2_libs
Syncing up with tag jetson_35.5...
Switched to a new branch 'mybranch_2024-05-10-1715311519'
/home/L5T_513/Linux_for_Tegra/sources/tegra/v4l2-src/v4l2_libs source
sync'ed to tag jetson_35.5 successfully!
# ls sources/3rdparty hardware kernel tegra
```

3.2.10 Next, setup the EAI-I131 Board Pinmux configurations.

Input the commands below:

```
# cd /home/L5T_513/Linux_for_Tegra
# cp /root/eai_i131_orin_JetPack513_20240521/board/tegra234-mb1-bct-gpio-p3767-hdmi-a03.dtsi bootloader/
# cp /root/eai_i131_orin_JetPack513_20240521/board/tegra234-mb1-bctpadvoltage-p3767-hdmi-a03.dtsi
bootloader/t186ref/BCT/
# cp /root/eai_i131_orin_JetPack513_20240521/board/tegra234-mb1-bct-pinmuxp3767-hdmi-a03.dtsi
bootloader/t186ref/BCT/
```

3.2.11 Now, we setup the compiling environment.

Input the commands below:

```
# export LOCALVERSION="-tegra"
# export CROSS_COMPILE=aarch64-buildroot-linux-gnu-
# export PATH=/home/L5T_513/gcc/bin/:${PATH}
# export TARGET_KERNEL_OUT=/home/L5T_513/Linux_for_Tegra/sources/kernel/build
```

3.2.12 Patch the Lanner Driver.

Input the commands below:

```
# mkdir -p /home/L5T_513/Linux_for_Tegra/sources/kernel/build
# cd /home/L5T_513/Linux_for_Tegra/sources/kernel/kernel-5.10
```

3.2.13 Next, start compiling the Jetson Linux Kernel. Note that this process may take 30 minutes or more.

Input the commands below:

```
# cp /root/eai_i131_orin_JetPack513_20240521/defconfig_513
arch/arm64/configs/defconfig
# make ARCH=arm64 O=$TARGET_KERNEL_OUT tegra_defconfig
# make ARCH=arm64 O=$TARGET_KERNEL_OUT menuconfig
# make ARCH=arm64 O=$TARGET_KERNEL_OUT -j 30*
# make ARCH=arm64 O=$TARGET_KERNEL_OUT dtbs
# make ARCH=arm64 O=$TARGET_KERNEL_OUT modules_install
INSTALL_MOD_PATH=/home/L5T_513/Linux_for_Tegra/rootfs
# cp -a
/home/L5T_513/Linux_for_Tegra/sources/kernel/build/arch/arm64/boot/Image
/home/L5T_513/Linux_for_Tegra/kernel/
# cp -a
/home/L5T_513/Linux_for_Tegra/sources/kernel/build/arch/arm64/boot/dts/nvidi
a/* /home/L5T_513/Linux_for_Tegra/kernel/dtb/
```

NOTE: *Depends on host CPU cores

3.2.14 We have successfully set up the Development Environment (in the Host Machine). Reboot to complete saving process. Input the command below:

```
# reboot
```

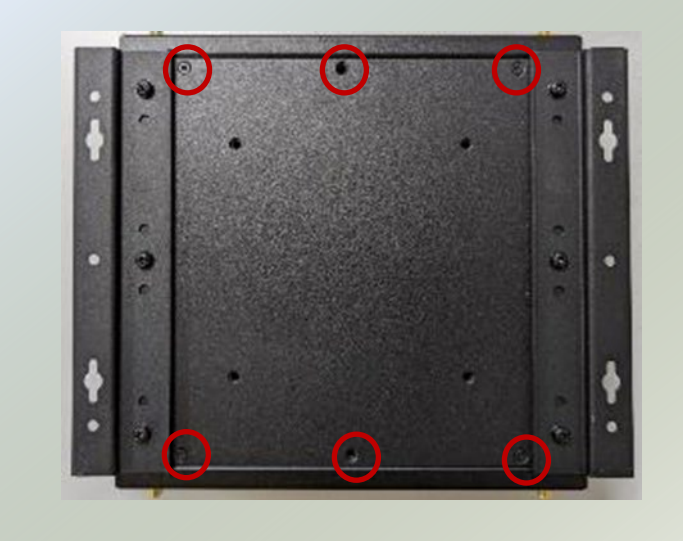
3.2.15 Next Step, we need to prep for burn-in image on the Target Device, the EAI-I131 appliance.

3.3 Target Device Jetson Linux Installation

Next, we need to connect the Host Machine to EAI-I131 appliance. Then create and burn-in the image to the EAI-I131 appliance to prepare it for deployment. To reduce the risk of personal injury, electric shock, or damage to the unit, please remove all power connections to completely shut down the device and wear ESD protection gloves when handling the installation steps.

3.3.1 Opening the Chassis

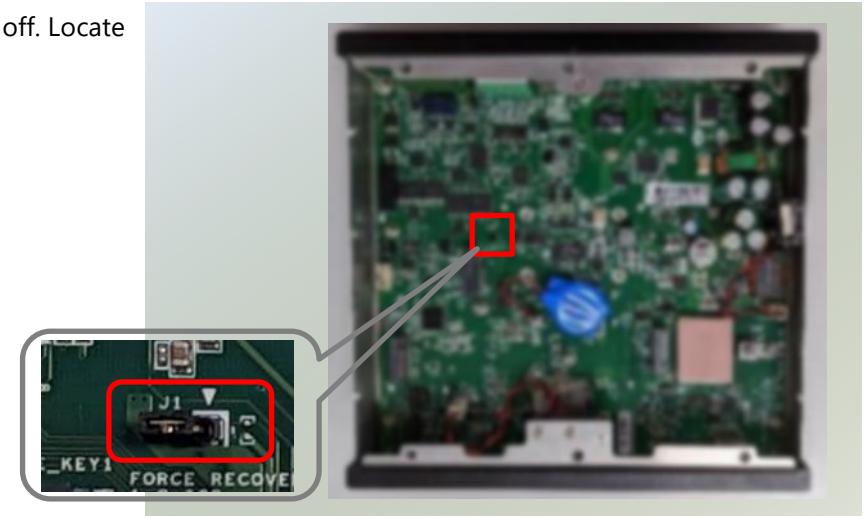
1. Power off the system and disconnect the power cord. Turn the system over. Unscrew the six (6) screws on the bottom chassis cover.



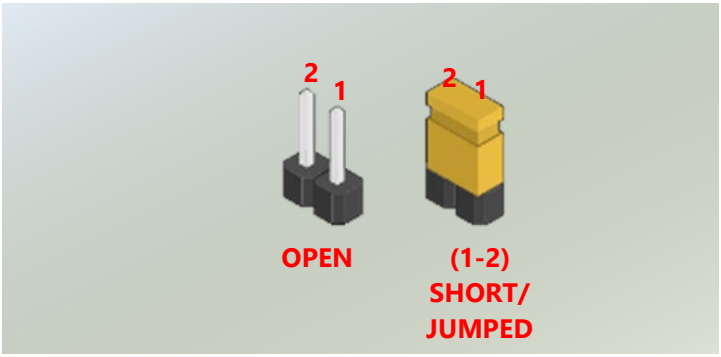
2. Lift and open the bottom chassis cover.

3.3.2 Connect Host Machine to Target Device

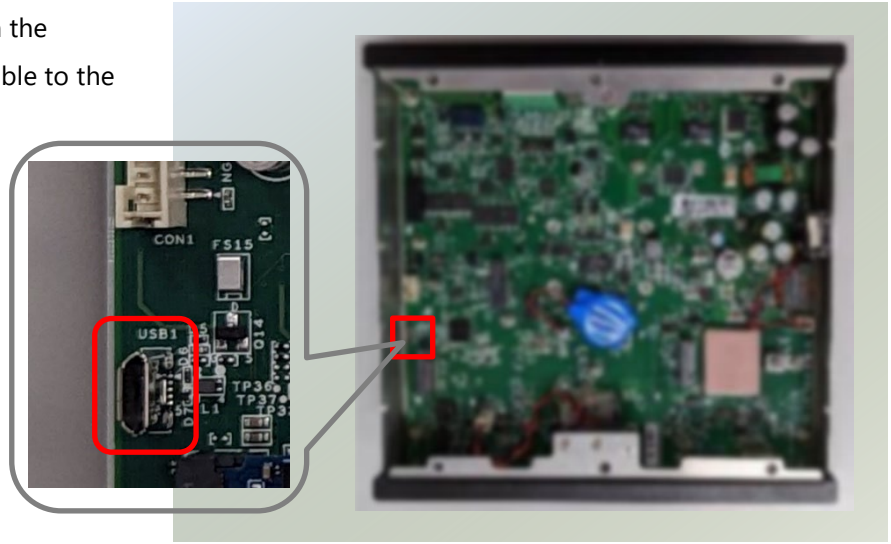
1. Make sure the system is powered off. Locate the Jumper **J1**.



2. To **SHORT** the designated pins, make sure the **jumper cap** (included in Accessory packet) is on Pin 1 & 2.



3. Locate the **USB1** connection on the motherboard. Insert the USB Cable to the USB1 Port.



4. Then, connect the other end of the USB Cable to the Host Machine.
5. Then plug in the DC adapter and power on the EAI-I131.

3.3.3 Setup and Burn-in Image

1. The Host Machine is now connected to the Target Device, EAI-I131 appliance. Next, open a new Terminal tab, like Tera term or PuTTY.

Input the command below:

```
# cd /home/L5T_513/Linux_for_Tegra
```

2. First, we need to check the Target Device, EAI-I131, has been detected by the Host Machine.

Input the commands below:

```
root@lanner;/home/L5T_513/Linux_for_Tegra# lsusb
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 003: ID 0955:7*23 NVIDIA Corp. APX
Bus 001 Device 002: ID 05e3:0610 Genesys Logic, Inc. Hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

NOTE: *ID No. will vary depending on the SKU.

- <nnnn> is a four-digit number that represents the type of your Jetson module:
  - 7023 for Jetson AGX Orin (P3701-0000 with 32GB)
  - 7023 for Jetson AGX Orin (P3701-0005 with 64GB)
  - 7023 for Jetson AGX Orin Industrial (P3701-0008 with 64GB)
  - 7223 for Jetson AGX Orin (P3701-0004 with 32GB)
  - 7323 for Jetson Orin NX (P3767-0000 with 16GB)
  - 7423 for Jetson Orin NX (P3767-0001 with 8GB)
  - 7523 for Jetson Orin Nano (P3767-0003 and P3767-0005 with 8GB)
  - 7623 for Jetson Orin Nano (P3767-0004 with 4GB)

3. Then, create image from Host Machine and burn-in to Target Device, EAI-I131.

Input the commands below:

```
# ./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
```

4. Following the steps above, we have successfully installed and burn-in the image (OS & SD) to the Target Device, EAI-I131 system.

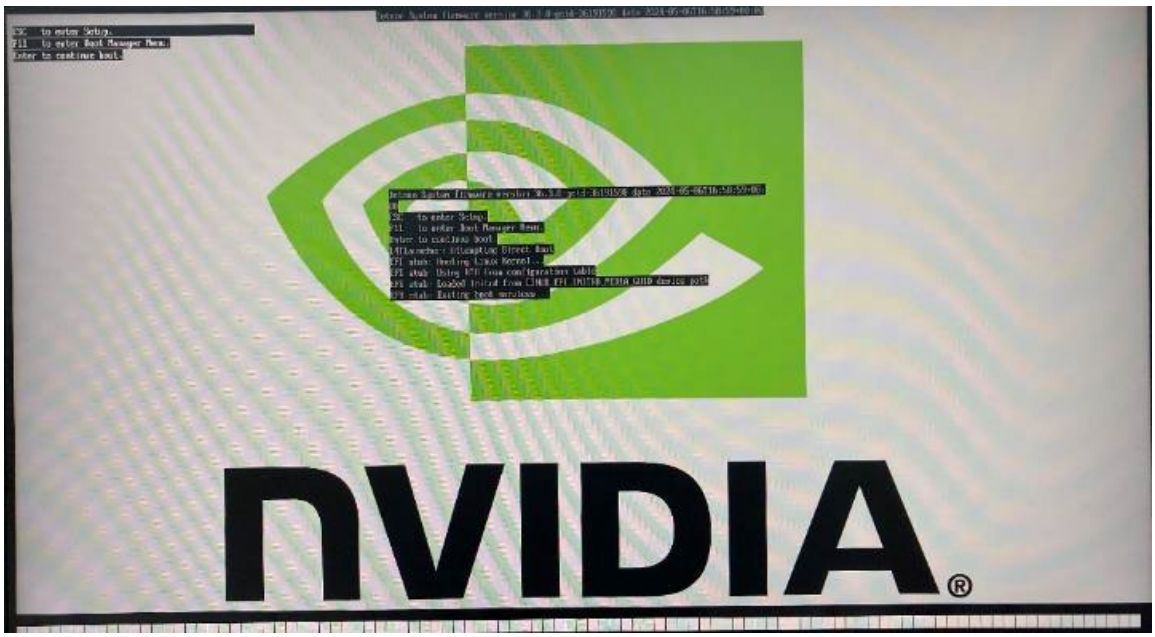
5. Next, unplug the power adapter from the system.

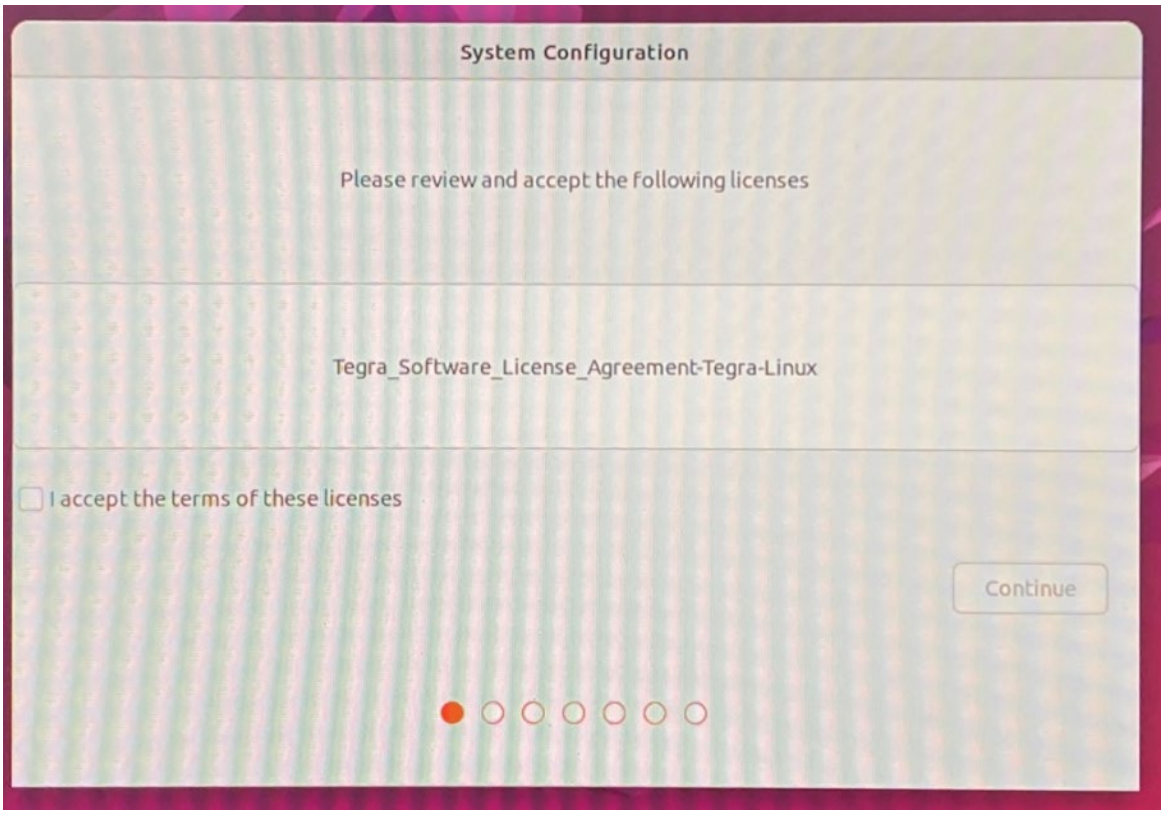
6. Then, remove the USB cable, and **OPEN** the jumper J1 (the jumper cap should NOT be covering Pin 1 & 2).

7. Then, place the bottom chassis back on and secure with the original screws. The EAI-I131 appliance is ready for AI-driven operations in its designated environment.

3.4 Startup

When starting up the EAI-I131 appliance, the login screen will display as below:





3. Then, follow Ubuntu setup to create **User Account and Password** for login.