

emplus

EEP-150N EDGE AI BOX

Software User Manual

Copyright

The document accompanying this product is Copyrighted © 2026 Emplus Technologies Inc. All rights reserved. Emplus reserves the right to change the product descriptions in this manual at any time without prior notice. No part of this manual may be reproduced, reprinted, translated, or transmitted in any form or by any means without prior written permission from Emplus. This manual is prepared to provide correct and reliable information; however, Emplus assumes no responsibility for the results of using this manual, or for any damage to third-party interests arising from its use.

Acknowledgement

Arm and AArch64 are trademarks of Arm Limited (or its subsidiaries) in the U.S. and other countries.

Debian is a registered trademark of Software in the Public Interest, Inc.

GNOME is a registered trademark of the GNOME Foundation.

HDMI, the HDMI logo, and High-Definition Multimedia Interface are trademarks or registered trademarks of HDMI Licensing Administrator, Inc.

Linux is a registered trademark of Linus Torvalds

NVIDIA®, the NVIDIA logo, Jetson®, Jetson Orin™, JetPack™, Linux for Tegra®, and L4T® are trademarks or registered trademarks of NVIDIA Corporation

Ubuntu is a registered trademark of Canonical Ltd.

UEFI is a registered trademark of the Unified EFI Forum, Inc.

USB and USB Type-A are trademarks of USB Implementers Forum, Inc.

All other product names and trademarks are the property of their respective owners.

Terminology and abbreviations

The following terms and abbreviations appear throughout this manual

<u>Term</u>	<u>Expansion / Definition</u>
Host PC	The Ubuntu Linux computer used to flash the OS image onto the EEP-150N and to run SDK Manager.
Jetson	NVIDIA's family of embedded AI-compute modules used inside the EEP-150N.
Jetson Orin NX / Orin Nano	The specific NVIDIA Jetson module variants supported by the EEP-150N (Orin NX 16 GB, Orin NX 8 GB, Orin Nano 8 GB).

<u>Term</u>	<u>Expansion / Definition</u>
JetPack	NVIDIA's software development kit (SDK) for Jetson platforms. It bundles NVIDIA Linux (L4T), libraries (CUDA, cuDNN, TensorRT, VPI, etc.), and development tools.
L4T	<i>Linux for Tegra</i> — NVIDIA's Ubuntu-based Linux distribution for Jetson modules. The underlying kernel and system software of JetPack.
SDK Manager	NVIDIA's graphical installer that downloads and deploys JetPack components to the host PC and, optionally, to the Jetson target.
Ubuntu	The Debian-based Linux distribution (supplied by Canonical) used as the host PC operating system in this manual.
Ubuntu LTS	<i>Long-Term Support</i> release of Ubuntu, receiving security updates for five years. 22.04 LTS and 24.04 LTS are supported host platforms.
OEM Config	The first-boot configuration wizard provided by NVIDIA Linux, used to set up language, keyboard, network, time zone, and user account on first power-up.
EULA	<i>End-User License Agreement</i> — the license terms presented during first-boot setup that the user must accept before using the NVIDIA Jetson software.
RCM	<i>Recovery Mode</i> — a special NVIDIA boot mode that allows the host PC to flash a new system image onto the Jetson module via USB.
Force Recovery button	A recessed button on the EEP-150N used (together with the Power or Reset button) to place the device into Recovery Mode.
UEFI	<i>Unified Extensible Firmware Interface</i> — the firmware standard that replaces the legacy BIOS and controls system boot. The EEP-150N uses UEFI boot.
extlinux.conf	The UEFI/bootloader configuration file that defines boot entries (labels) and kernel command-line parameters for the EEP-150N.
GUI	<i>Graphical User Interface</i> — the mouse-driven graphical version of the Device-IO tool, delivered as a GNOME Shell extension.
TUI	<i>Terminal User Interface</i> — the keyboard-driven text version of the Device-IO tool, usable over SSH or on headless systems.
GNOME	The desktop environment used by Ubuntu. The Device-IO GUI is provided as a GNOME Shell extension.
DT	<i>Device Tree</i> — a data structure used by the Linux kernel to describe the hardware of the system.
DTBO	<i>Device Tree Blob Overlay</i> — a compiled, binary fragment of a Device Tree that modifies the base Device Tree at boot time. The Device-IO tool writes DTBO files to apply serial-port, USB-role, and DB-25 pin-group settings.
RT kernel	<i>Real-Time kernel</i> — a Linux kernel variant (PREEMPT_RT) with deterministic response-time characteristics, suitable for industrial and control workloads.
OOT modules	<i>Out-of-Tree kernel modules</i> — kernel modules built outside of the main Linux kernel

<u>Term</u>	<u>Expansion / Definition</u>
	source tree. The EEP-150N packages include OOT module headers for building third-party drivers.
.deb package	The Debian package format used by Ubuntu for distributing and installing software. Packages carry a .deb filename extension and are installed with dpkg or apt.
dpkg	The low-level Debian package management utility used to install, configure, and remove .deb packages.
scp	<i>Secure Copy</i> — a command-line utility for copying files between hosts over SSH.
SSH	<i>Secure Shell</i> — a protocol for securely accessing a remote Linux shell.
lsusb	A Linux command that lists USB devices attached to the host. Used in this manual to verify that the EEP-150N is in Recovery Mode.
sudo	A Linux command that runs the following command with superuser (root) privileges, prompting for the user's password.
arm64	The 64-bit ARM processor architecture used by the Jetson Orin modules. Also known as AArch64.
NVMe SSD	<i>Non-Volatile Memory Express Solid-State Drive</i> — the high-speed storage device used by the EEP-150N.
SO-DIMM	<i>Small Outline Dual In-line Memory Module</i> — the physical form factor of the Jetson Orin module socket.
HDMI	<i>High-Definition Multimedia Interface</i> — the digital video/audio connector used for the EEP-150N's display output.
USB Type-A	A common, rectangular USB connector. The EEP-150N uses a Type-A-to-Type-A cable to connect to the host PC during flashing.
COM1 / COM2	The two serial ports on the EEP-150N, each configurable as RS-232, RS-422, or RS-485.
RS-232 / RS-422 / RS-485	Industry-standard serial-communication protocols. RS-232 is point-to-point, short-range; RS-422 is differential full-duplex; RS-485 is differential multi-drop.
DB-25	A 25-pin D-subminiature connector on the EEP-150N, providing user-configurable GPIO/DIDO lines.
GPIO	<i>General-Purpose Input/Output</i> — programmable digital signal lines.
DIDO	<i>Digital Input / Digital Output</i> — discrete digital signal lines used to interface with industrial sensors and actuators.
APX	<i>Application Processor in AP bootrom Mode</i> — the USB device class reported by a Jetson in NVIDIA Recovery Mode (appears in lsusb output as "NVIDIA Corp. APX").

Table of Contents

- Copyright ii
- Acknowledgement..... ii
- Terminology and abbreviations ii
- 1. Introduction 1
 - Audience 1
 - Manual overview 1
 - Prerequisites 2
 - Typographical conventions 2
- 2. Image update 3
 - Prerequisites 3
 - Entering Recovery Mode 3
 - Confirming Recovery Mode 4
 - Flashing the image 4
- 3. Login screen 6
 - First boot 6
 - After OEM Config 10
- 4. Device-IO tool 13
 - Graphical User Interface (GUI) 13
 - Terminal User Interface (TUI) 16
 - Boot options 22
- 5. deb package update 23
 - RT kernel packages 23
 - Kernel headers packages 23
- 6. Install JetPack from SDK Manager 25
 - Install SDK Manager 25
 - Run SDK Manager 25
 - Log in 25
 - Step 01 — Set up the development environment 26
 - Step 02 — Review components and accept licenses 27
 - Step 03 — Installation 28
 - Step 04 — Finalize setup 30

1. Introduction

The EEP-150N is a production-ready edge AI appliance powered by the NVIDIA Jetson Orin NX and Orin Nano modules. It delivers the compute density and energy efficiency of Jetson in a fully provisioned platform—letting you move from prototype to deployment without spending engineering cycles on board bring-up, kernel integration, or system tuning.

This manual is your reference for bringing up and maintaining the EEP-150N software stack. It walks through five workflows you'll touch across the device's lifecycle: flashing a system image via NVIDIA Recovery Mode (RCM), first-boot OEM configuration, runtime I/O configuration with the Device-IO tool, .deb package updates for the real-time kernel and kernel headers, and host-side JetPack installation through NVIDIA SDK Manager.

Audience

This manual is written for software engineers, system integrators, and technical evaluators who provision, deploy, or maintain EEP-150N units in the field. We assume you're comfortable on the Linux command line, can manage Ubuntu packages with apt and dpkg, and have working familiarity with the NVIDIA Jetson platform. If a step requires deeper context, we'll point you to the relevant NVIDIA documentation rather than duplicating it here.

Manual overview

Each chapter is self-contained and maps to a single end-to-end task. Read them in order during initial bring-up, or jump directly to the chapter you need for a specific maintenance operation:

- Chapter 2, Image update — Flash a fresh OS image to the EEP-150N from an Ubuntu host PC using NVIDIA Recovery Mode (RCM).
- Chapter 3, Login screen — Complete first-boot OEM Config and the post-OEM welcome tour to land on a usable Ubuntu desktop with your account provisioned.
- Chapter 4, Device-IO tool — Configure serial-port modes (RS-232/422/485), the USB3 Port A role, and DB-25 pin-group functions, using either the GNOME GUI or the headless-friendly TUI.
- Chapter 5, .deb package update — Install the real-time (PREEMPT_RT) kernel or the kernel-headers packages on a running EEP-150N for low-latency workloads or third-party module compilation.
- Chapter 6, Install JetPack from SDK Manager — Provision your host PC with JetPack (CUDA, cuDNN, TensorRT, VPI, and dev tools) so you can build and cross-deploy AI workloads to the EEP-150N.

Prerequisites

Gather the following before starting any procedure in this manual. Individual chapters call out additional, task-specific requirements where they apply:

- An EEP-150N device.
- An Ubuntu host PC running 22.04 LTS or 24.04 LTS (for flashing and SDK Manager operations).
- A USB Type-A to Type-A cable connecting the host PC to the EEP-150N (required for Recovery Mode flashing).
- An HDMI monitor and a USB keyboard/mouse (required for first-boot configuration).
- The power cable and AC adapter supplied with the EEP-150N.
- `sudo`/root privileges on both the host PC and the EEP-150N.
- Network access on the host PC (required for SDK Manager downloads).

Typographical conventions

This manual uses the conventions below consistently. Skim them once—they'll save you time when reading commands and following procedures:

- Fixed-width font indicates commands, file paths, and on-screen text, for example: `sudo dpkg -i *.deb`
- Angle-bracket placeholders, for example `<version>` or `<date>`, should be replaced with the actual value appropriate for your download.
- A `$` shell prompt precedes commands to be entered at a shell.
- A `#` shell prompt precedes commands to be entered at a root shell.
- Bold text marks UI elements such as buttons, menu items, and field labels.
- Notes are introduced with the keyword "Note:" and contain supplementary information. Important steps are introduced with "Important:" and must not be skipped.

2. Image update

Use this procedure to write a complete OS image to the EEP-150N—during initial provisioning, when recovering from a corrupt system, or when rolling out a new release. Flashing is driven from an Ubuntu host PC over USB, with the EEP-150N held in NVIDIA Recovery Mode (RCM). Plan for roughly 15–30 minutes per device, depending on host performance and image size.



Important: Flashing wipes the EEP-150N’s on-device storage. Back up any user data, configuration, and SSH keys before starting—there is no recovery path once the flash begins.

Prerequisites

Before starting, prepare the following:

- An Ubuntu host PC (22.04 LTS or 24.04 LTS).
- An EEP-150N device with the supplied power cable.
- The OS image package appropriate for your Jetson variant. See your sales contact for the latest build.
- A USB Type-A to Type-A cable connecting the EEP-150N to the host PC.
- An HDMI cable connecting the EEP-150N to a monitor.

Connect the cables as shown below:



Entering Recovery Mode

The EEP-150N must be placed in NVIDIA Recovery Mode before flashing. The procedure depends on the

current power state of the device.

From a powered-off state

1. Press and hold the Force Recovery button.
2. Press and release the Power button.
3. Release the Force Recovery button.

From a powered-on state

1. Press and hold the Force Recovery button.
2. Press and release the Reset button.
3. Release the Force Recovery button.

Confirming Recovery Mode

When RCM is engaged, the Jetson module enumerates on the host PC as an NVIDIA USB device. Verify before flashing—skipping this check is the most common cause of failed flash attempts:

1. Open a terminal window on the host PC and run:

```
$ lsusb
```

2. Look for a line of the form:

```
Bus XXX Device XXX: ID 0955:XXXX NVIDIA Corp. APX
```

The USB product ID depends on the Jetson module:

- Orin NX 16 GB — 0955:7323 NVIDIA Corp. APX
- Orin NX 8 GB — 0955:7423 NVIDIA Corp. APX
- Orin Nano 8 GB — 0955:7523 NVIDIA Corp. APX

Flashing the image

1. Open a terminal window on the host PC and enter the command for your Jetson variant.

For Orin NX 16 GB / 8 GB:

```
$ sudo chmod a+x ./EEP-150N_NXA10_<version>_<date>.run
```

```
$ sudo ./EEP-150N_NXA10_<version>_<date>.run
```

For Orin Nano 8 GB:

```
$ sudo chmod a+x ./EEP-150N_NNA10_<version>_<date>.run
```

```
$ sudo ./EEP-150N_NNA10_<version>_<date>.run
```

2. After the package is extracted, a menu appears. Enter "1" and press Enter:

```
=== Main Menu ===
```

1. l4t_initrd_flash mode
2. Board Info
0. Exit

```
=====
```

```
Select an option (0-2):
```

3. Wait for the flashing process to complete. On success, the following messages are displayed:

```
Flash is successful
```

```
Reboot device
```

```
Cleaning up...
```

```
Log is saved to Linux_for_Tegra/initrdlog/flash_1-8_0_<date>-<time>.log
```

```
Operation completed. Press Enter to continue...
```

4. Press Enter, then enter "0" to exit the flash script when the Flash Options menu reappears.
5. The EEP-150N reboots automatically. Proceed to Chapter 3 to complete the initial setup.

3. Login screen

The first power-up after a flash drops you into NVIDIA’s OEM Config wizard, which provisions the system locale, network, and your user account. Subsequent boots show the post-OEM welcome tour the first time you log in. This chapter walks through both flows so you can hand off a freshly imaged unit ready for use.

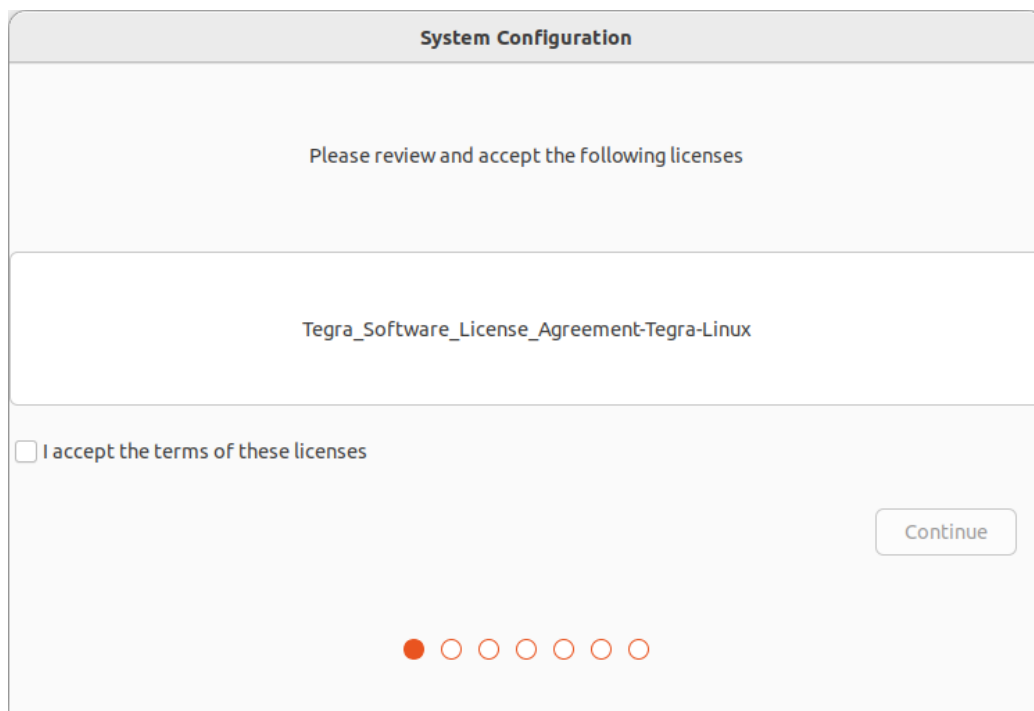


Important: OEM Config requires direct console access. Connect an HDMI monitor and USB keyboard/mouse before powering on for first boot—remote/headless setup is not supported at this stage.

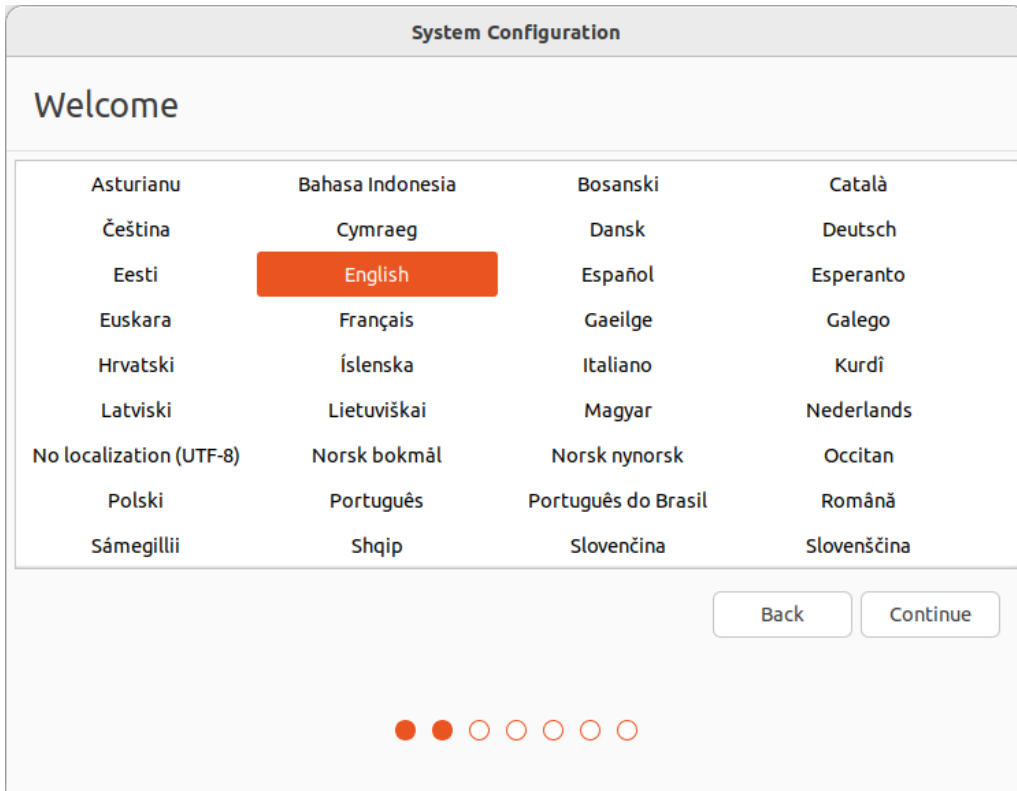
First boot

The first boot launches NVIDIA’s OEM Config wizard, which collects everything needed to bring up Ubuntu for the first time. Step through it in the order shown—you cannot revisit earlier screens once you advance:

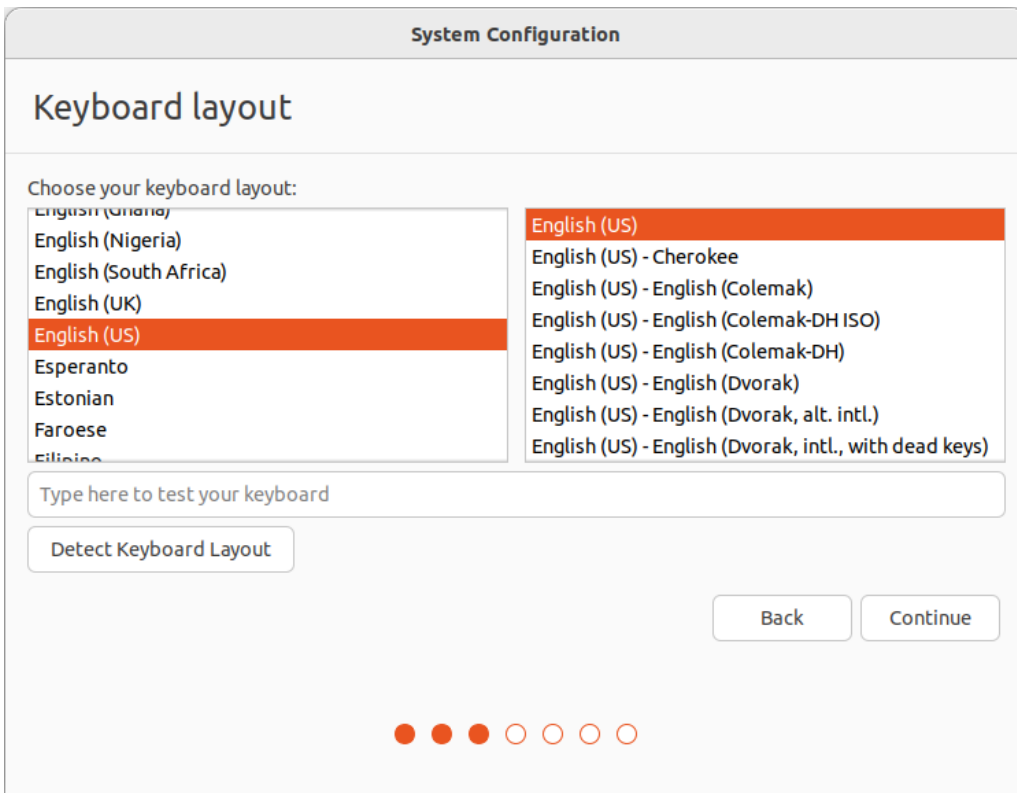
1. Review and accept the NVIDIA Jetson software EULA.



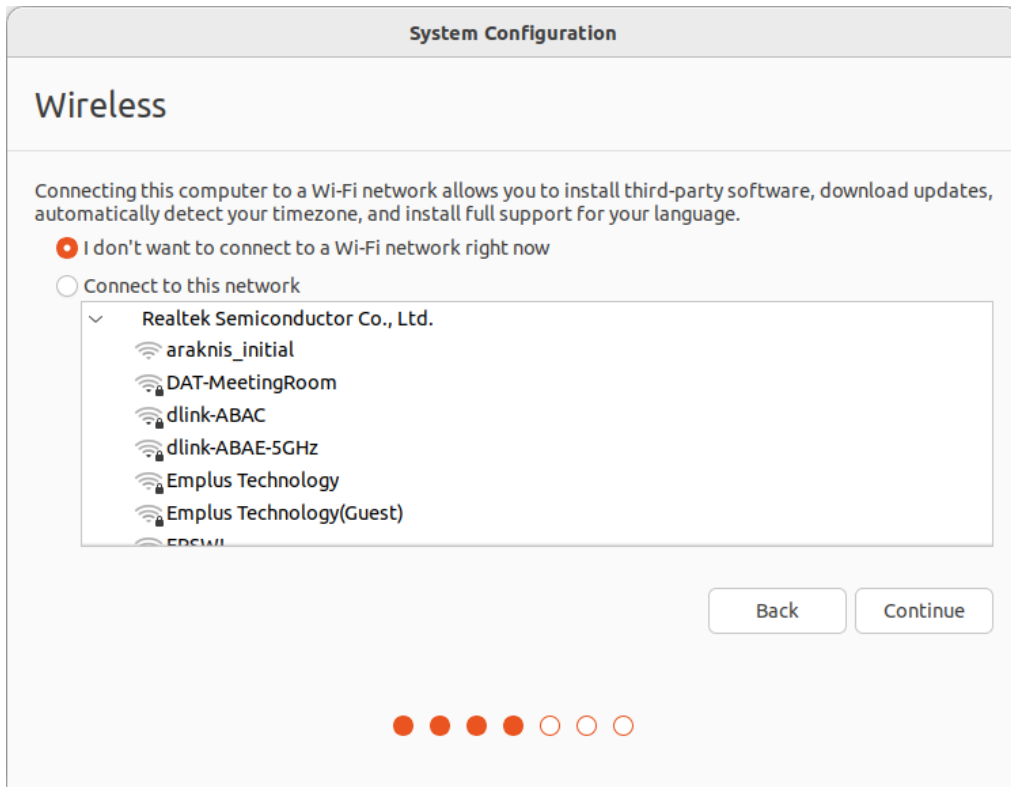
2. Select the system language.



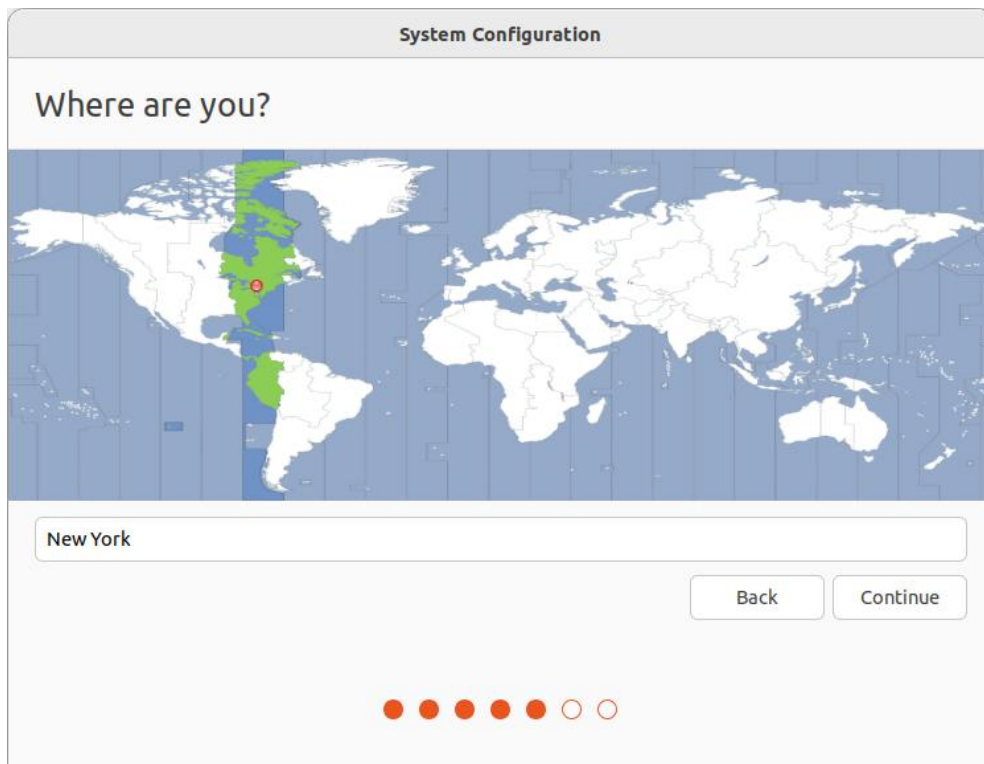
3. Select the keyboard layout.



4. Connect to a wireless network (optional, but recommended).



5. Select the time zone.



6. Create a user name, password, and computer name.

System Configuration

Who are you?

Your name:

Your computer's name:
The name it uses when it talks to other computers.

Pick a username:

Choose a password:

Confirm your password:

Log in automatically
 Require my password to log in

7. Choose whether to install a browser.

System Configuration

Install Chromium Browser

Do you want to install Chromium Browser now?

Install Chromium Browser

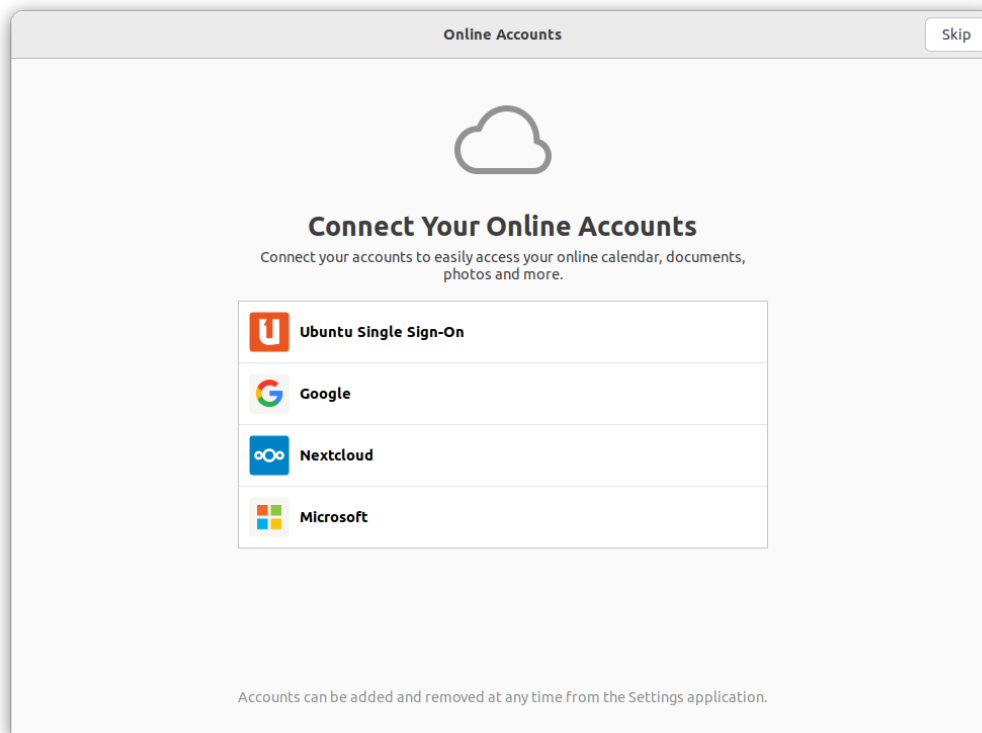
Do not install Chromium Browser

NOTE: Internet connection is required for Chromium Browser installation, and it will take several minutes. You can also install it via snap store after system configuration is finished.

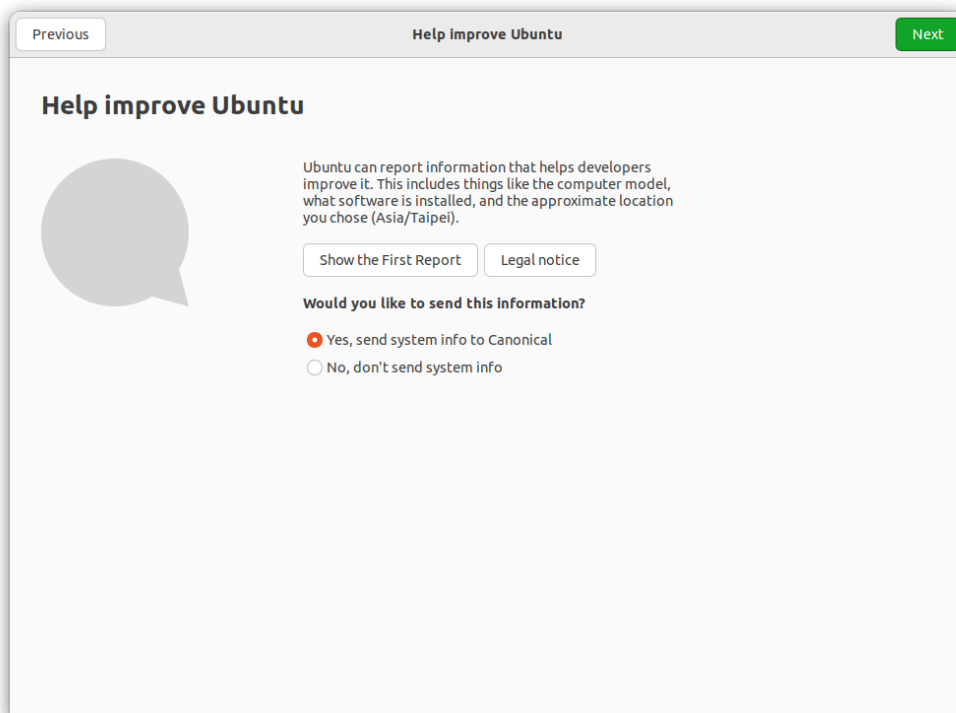
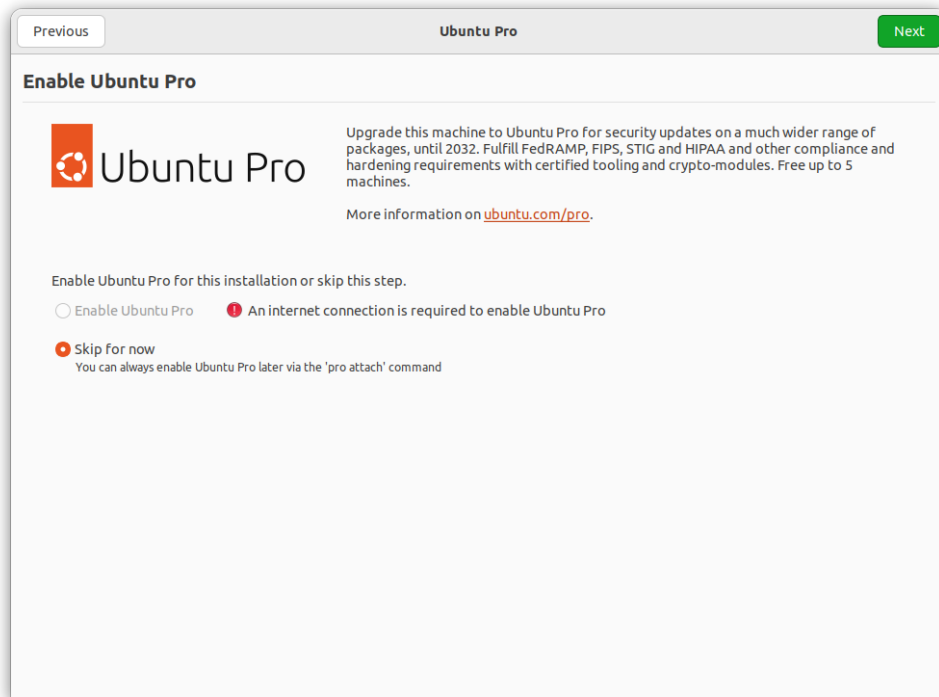
8. Log in with the credentials just created.

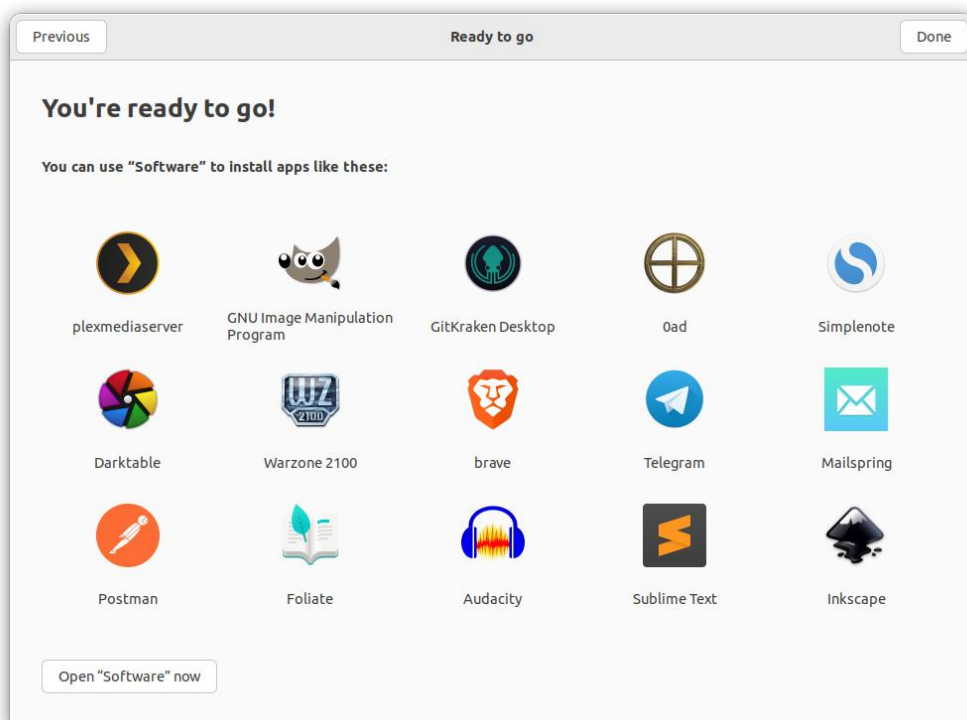
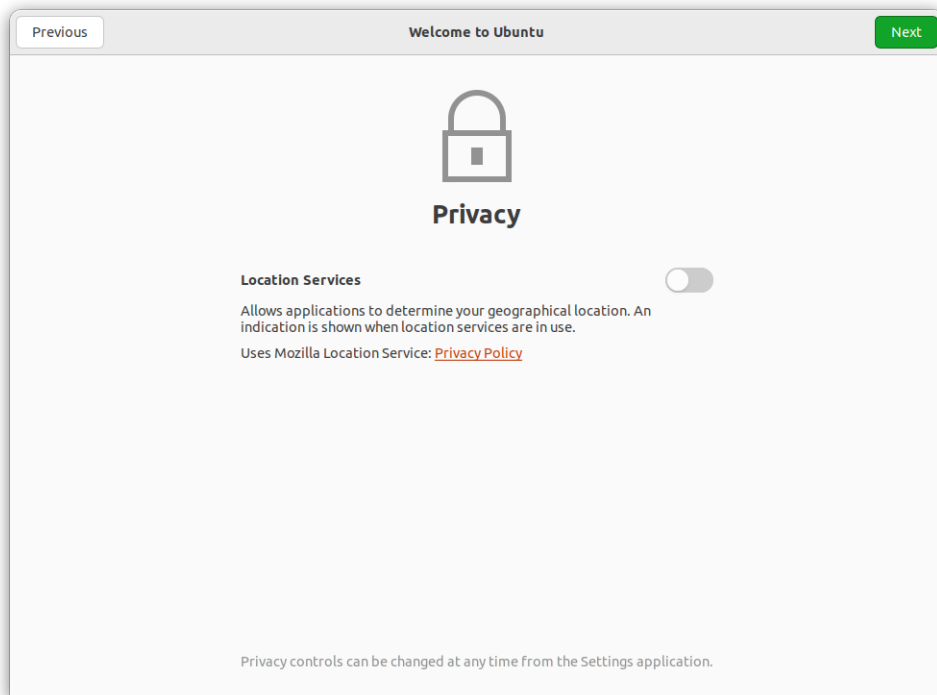
After OEM Config

When OEM Config finishes, the system reboots into the Ubuntu desktop using the credentials you just created. From this point forward, the EEP-150N behaves like any other Ubuntu workstation:



On first login, Ubuntu launches a short welcome tour covering system settings and software availability. None of the steps are mandatory—click Next on each screen to advance, and the tour closes automatically when it reports “Done.” You can dismiss it at any time without affecting the system configuration.





4. Device-IO tool

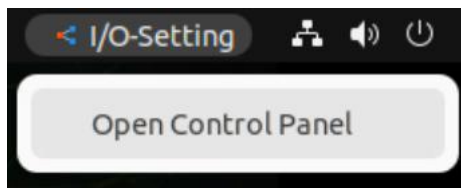
Device-IO is the runtime tool you'll use to adapt the EEP-150N's physical I/O to your application. It handles three things: per-port serial mode (RS-232, RS-422, or RS-485) on COM1 and COM2, the host/device role of USB3 Port A, and pin-group functions on the DB-25 connector. Under the hood, the tool emits Device Tree Blob Overlays (DTBOs) and updates the UEFI boot configuration, so your settings persist across reboots. Two front-ends are shipped, and they're feature-equivalent—pick whichever fits the workflow:

- A Graphical User Interface (GUI), delivered as a GNOME Shell extension, for interactive device-tree overlay (DTBO) management and mode configuration.
- A Terminal User Interface (TUI) that offers the same functionality for headless environments and remote SSH sessions.

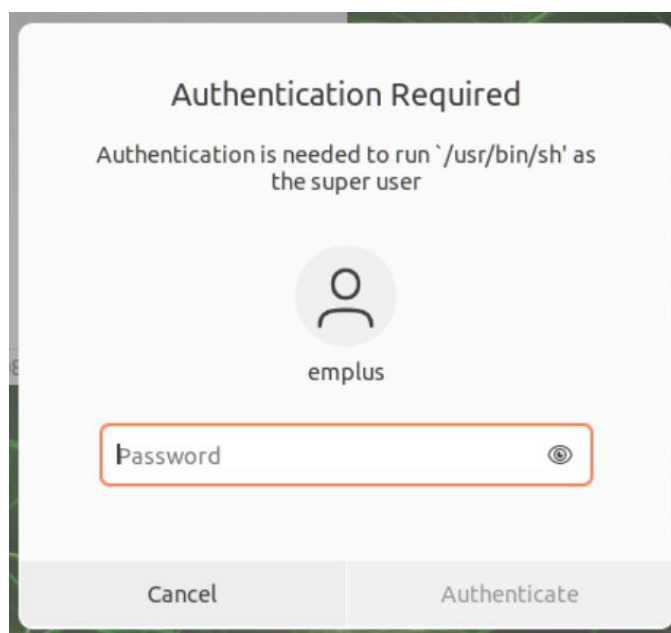
Graphical User Interface (GUI)

Usage

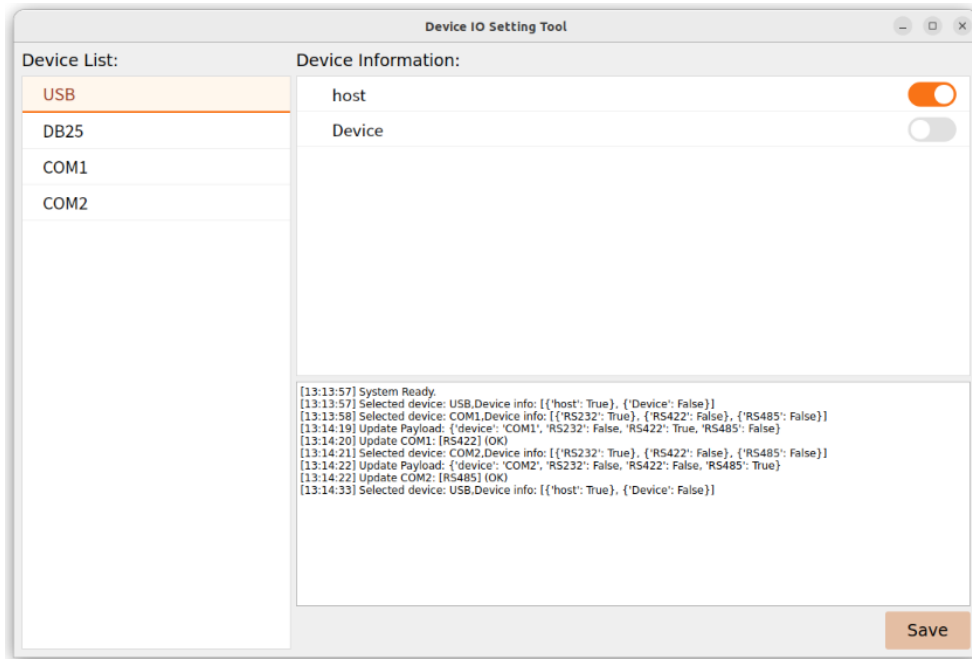
1. The I/O Settings utility appears as a GNOME extension in the top-right corner of the desktop. Click the icon to open it.



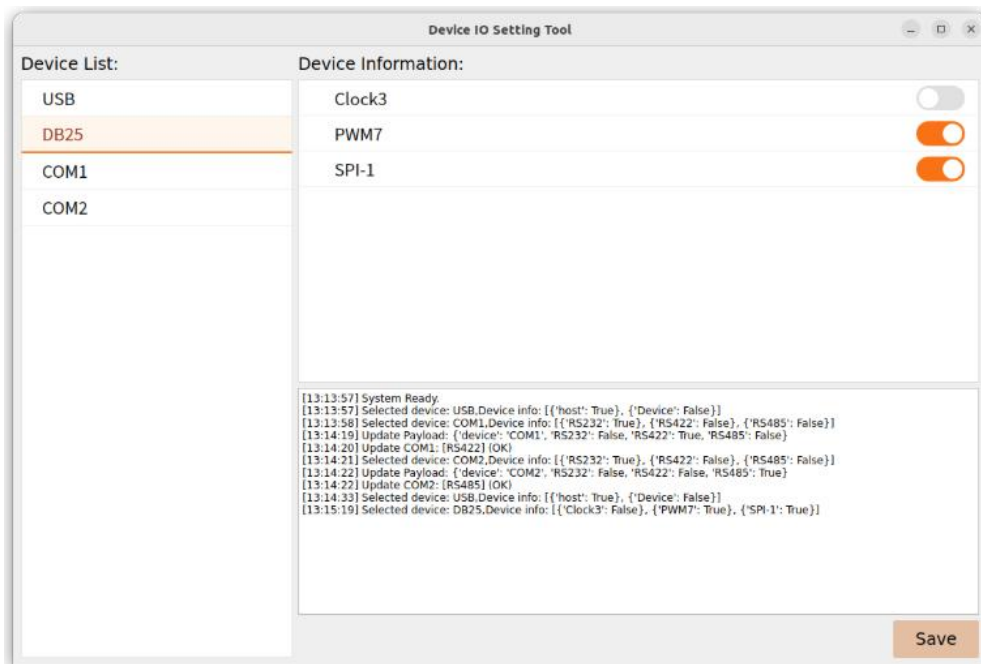
2. Click Open Control Panel to launch the application.
3. A password prompt appears requesting administrator privileges. Enter the user password.



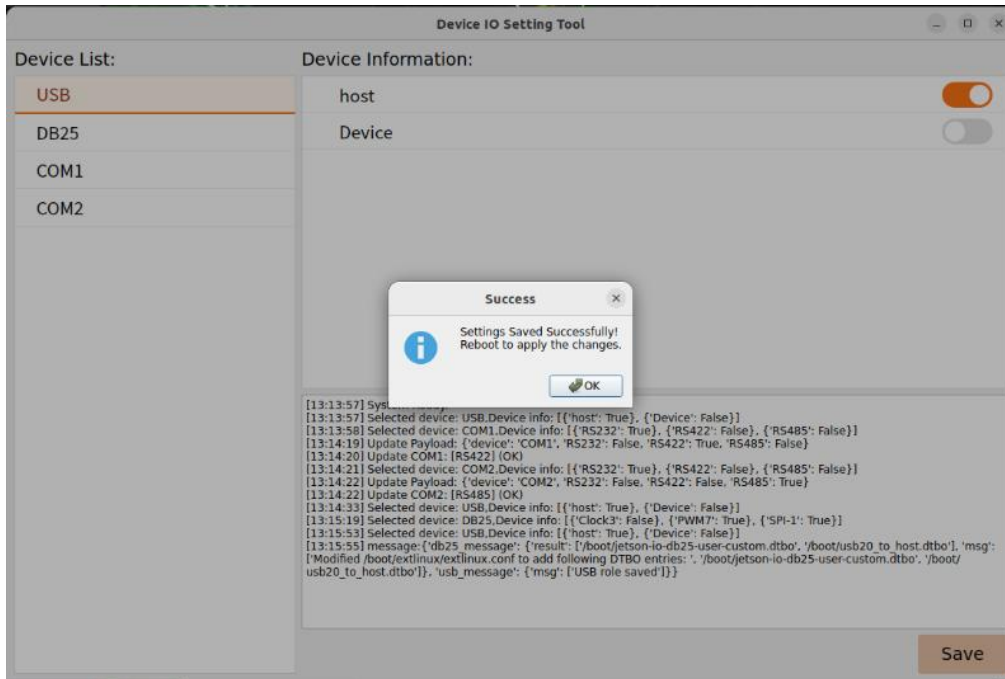
4. Configuration options are listed in the left-hand column. Available items are shown to the right of each option, and runtime information is displayed at the bottom of the window. Click an option to configure it.



5. Use the toggle switch to enable or disable a feature. A gray switch indicates the option is disabled; an orange switch indicates the option is enabled.



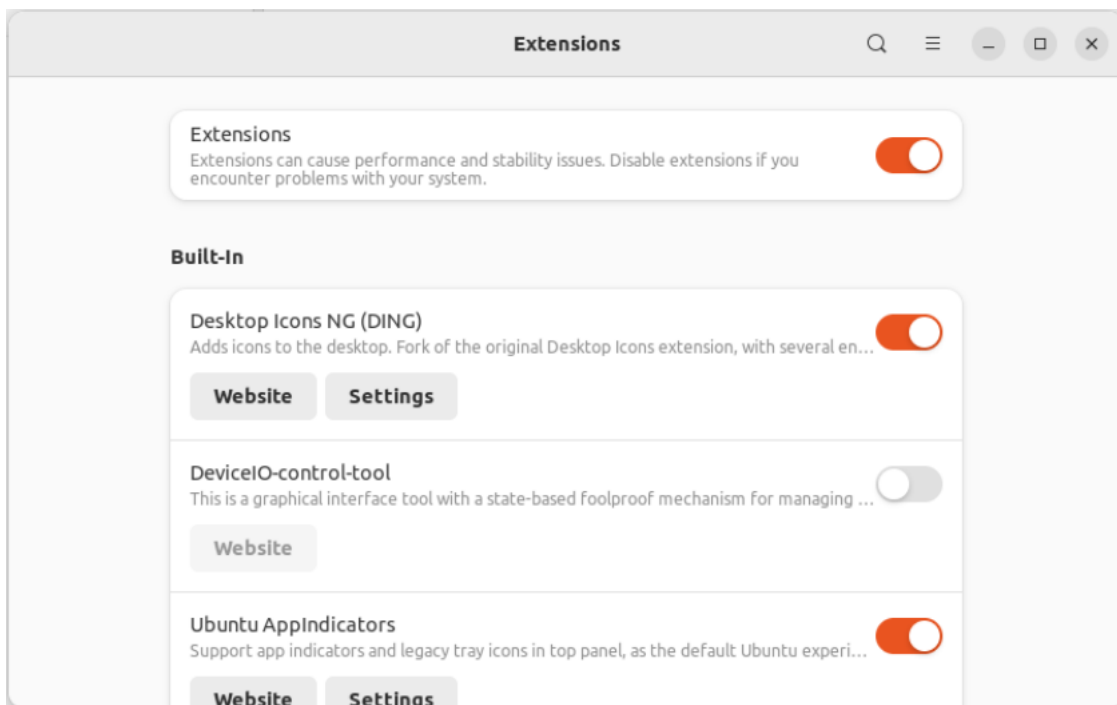
6. For USB and DB-25 configuration, click Save after making changes. The new settings take effect at the next reboot.



7. For COM1 and COM2, changes are applied immediately when the switch is toggled. No reboot is required.

Managing the extension

To hide the I/O Settings icon from the top panel, open Application Menu → Extensions → DeviceIO-control-tool and deselect the option. The change takes effect immediately.



Terminal User Interface (TUI)

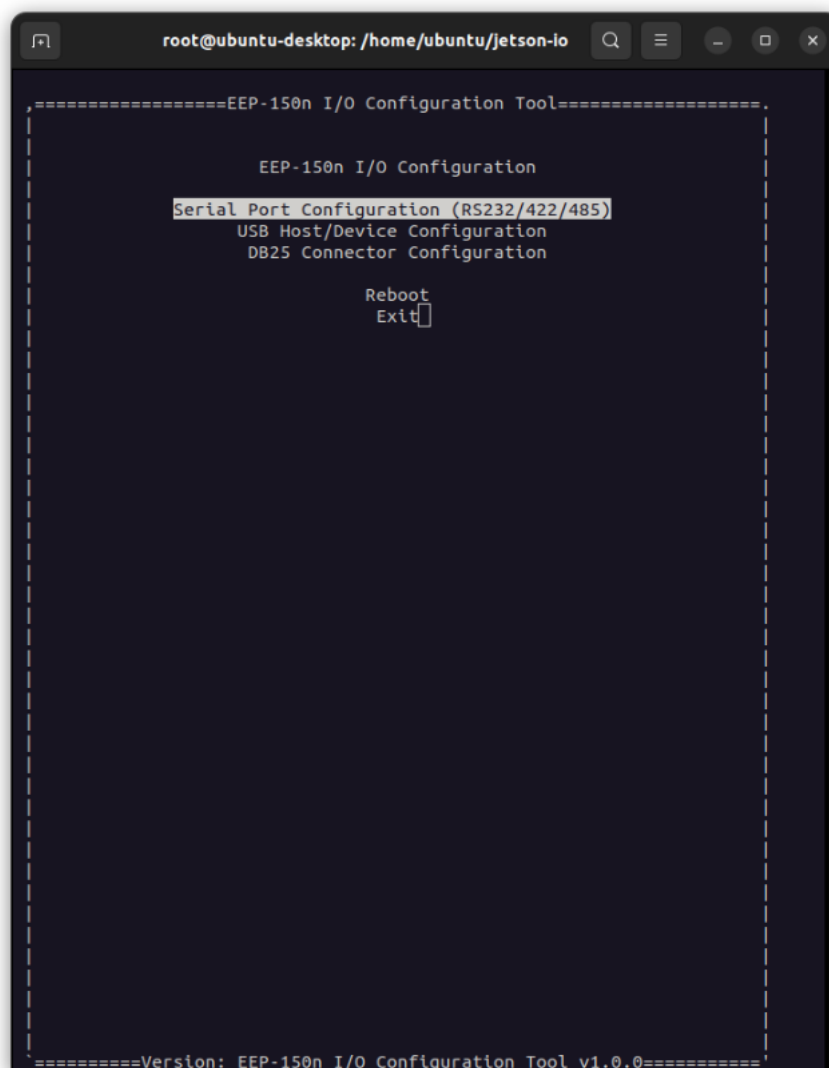
Usage

The TUI exposes the same three workflows as the GUI—Serial Mode selection, USB Role configuration, and DB-25 pin mapping—in a curses-style interface that runs cleanly over SSH. This is the front-end of choice for headless deployments, fleet provisioning scripts, and field maintenance over a remote shell.

Launch the TUI from a terminal with root privileges:

```
$ sudo /opt/eep/eep150n-io/eep150n_io_config
```

The TUI opens in the terminal window as shown below. Use the arrow keys to move between options and press Enter to select.



From the main menu the user can choose any of the three configuration workflows, or one of two operations:

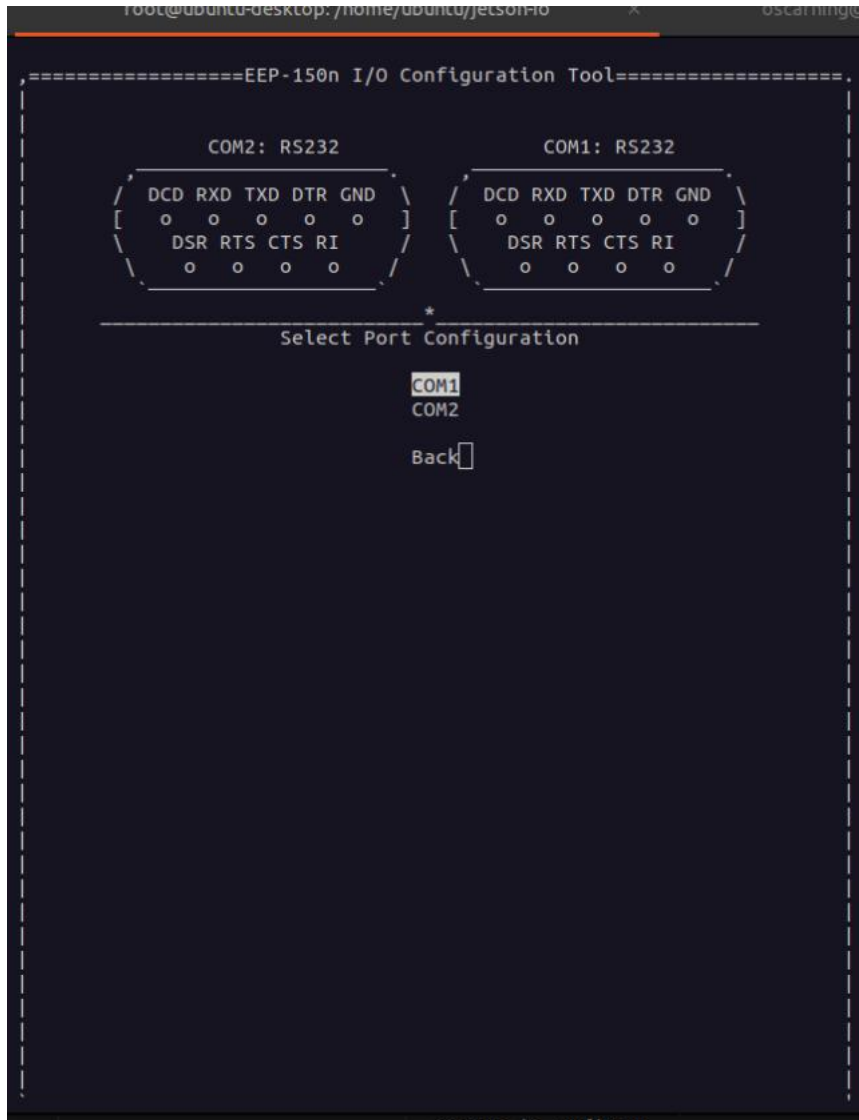
- Reboot — opens a confirmation dialog before initiating a system reboot. Press Ctrl+C in the dialog to

cancel.

- Exit — quits the tool and discards any unsaved changes.

The three configuration workflows are described in the following sections.

Serial port configuration



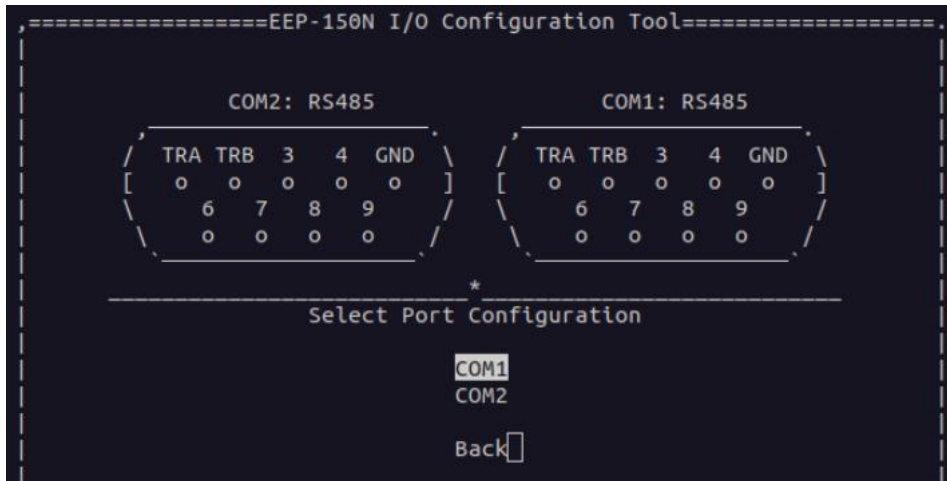
This dialog lets the user configure one of two serial ports. Each port supports three modes: RS-232, RS-485, and RS-422.

After a mode is selected, the physical diagram at the top of the window updates to reflect the chosen port and its mode, and the selected mode is applied to the device immediately — no reboot is required.

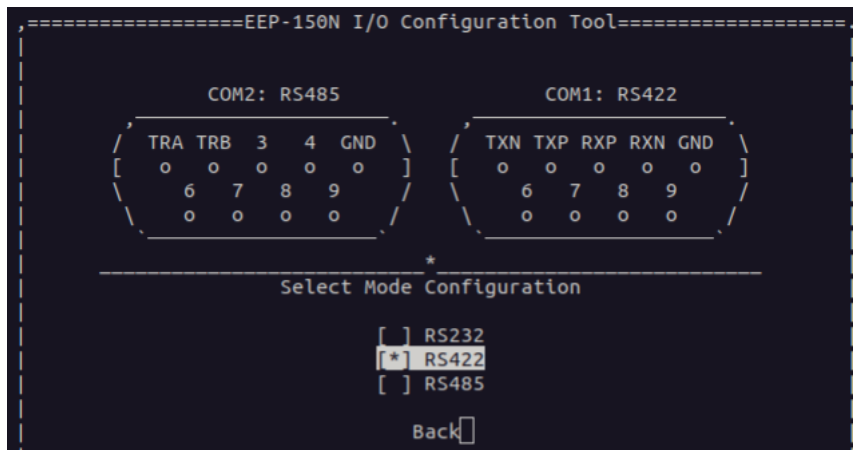
This dialog does not use Save or Discard; changes take effect the moment an option is chosen.

Configuration steps:

1. Select Serial Port Configuration on the main page.
2. Select the target port on the configuration page, for example COM1.



3. Navigate to the desired mode and press Enter. The hardware diagram updates to show the selected pin-out.



4. Select Back to return to the port-selection page. Select Back again to return to the main page, or choose another port to configure.

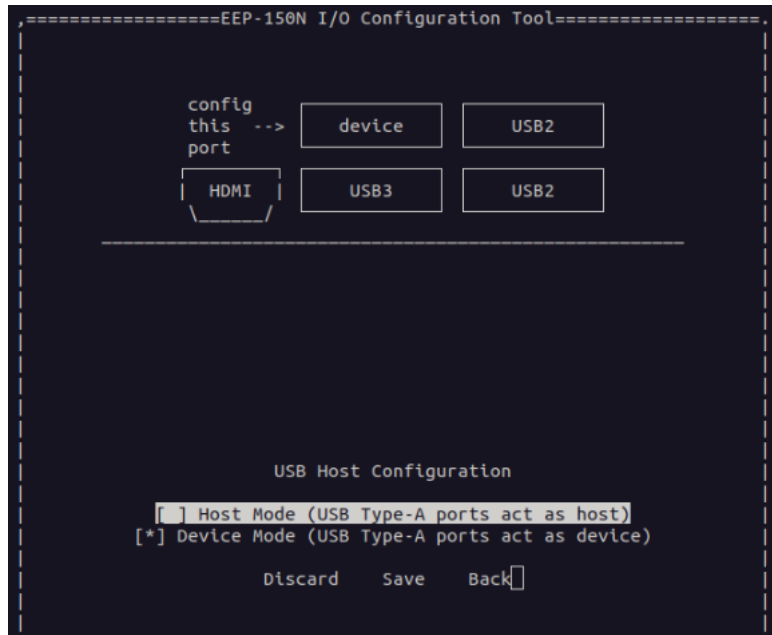
USB role configuration



This dialog shows a simplified front view of the EEP-150N. The user can select one of two roles (Host or Device) for USB3 Port A — indicated on the screen by an arrow. After a role is selected, the diagram updates to reflect the current selection.

Configuration steps:

1. Select USB Host/Device Configuration on the main page.
2. Select the desired port function. The top diagram updates to show the current selection.

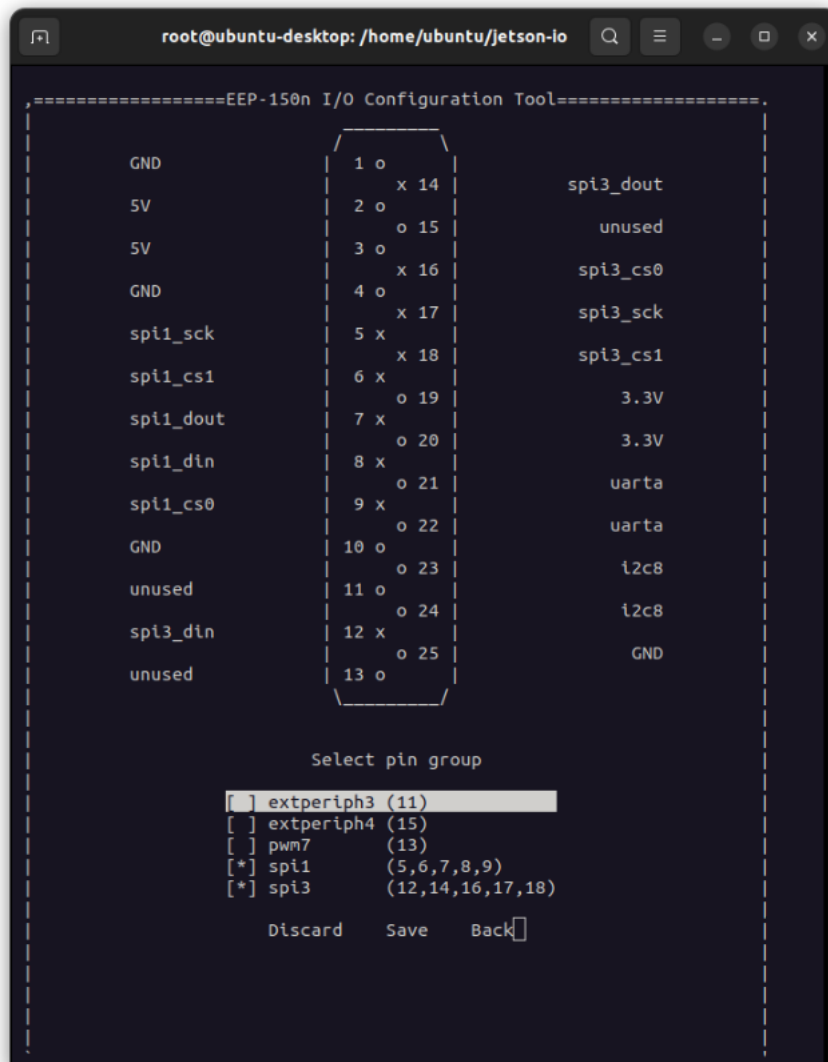


3. Choose one of the following actions:

- Discard — restores the option to the last saved USB role configuration from the DTBO file and discards any unsaved changes.
- Save — writes the selected USB role to the DTBO file. The updated role is not applied to the running system until the device is rebooted.
- Back — returns to the main page and discards any unsaved changes.

4. Reboot the device for the new setting to take effect.

DB-25 pin-group configuration



This dialog shows a simplified front view of the DB-25 connector. The user can configure the function of selected pin groups on the connector. After a function is selected, the diagram at the top of the window updates to reflect the current pin-group configuration. Pins marked "x" have been assigned to a special function by the tool.

Configuration steps:

1. Select DB25 Connector Configuration on the main page.
2. Enable or disable a function group by toggling the option. An asterisk ("*") next to an option indicates the function is currently enabled.
3. Choose one of the following actions:
 - Save — writes the selected pin-group configuration to the DTBO file. The configuration is not applied to the running system until the device is rebooted.
 - Discard — restores the configuration to the current system settings.

- Back — discards any unsaved changes and returns to the main menu. No pending changes are applied.

4. Reboot the device for the new setting to take effect.

Boot options

UEFI boot entries on the EEP-150N are defined in `/boot/extlinux/extlinux.conf`. Both the GUI and TUI write to this file when persisting your selections, so the same boot logic applies regardless of which front-end you used:

On first save, Device-IO clones the existing default boot entry into a new label named "EEP-150N-IO" and promotes it to default. The clone inherits the original kernel, initrd, and APPEND directives unchanged; the only difference is an OVERLAY directive listing the DTBOs for your selected I/O functions. This keeps your factory boot path intact as a fallback while letting you iterate on I/O configuration safely.

```
TIMEOUT 10
DEFAULT EEP-150N-IO

MENU TITLE L4T boot options
```

The option name displayed in the UEFI menu is derived from the selected function names to aid identification.

Because the new boot option is copied from the previous default, and the tool modifies only the "EEP-150N-IO" label, the boot image used by the "EEP-150N-IO" option is identical to the previous default option after the configuration is saved.

5. deb package update

Two .deb package sets ship with the EEP-150N for in-place updates: the PREEMPT_RT kernel (for deterministic, low-latency control workloads) and the kernel-headers (for compiling out-of-tree modules and third-party drivers). Both are installed on a running unit—no reflashing required. This chapter covers preparation, installation, and the post-install reboot for each set, including which headers package to choose based on the kernel currently running.

RT kernel packages

Preparation

Before starting the installation, copy the RT package directory from the host PC to the EEP-150N (for example, using scp or a USB drive).

Installation

1. On the EEP-150N, navigate to the RT directory and install the packages:

```
$ cd RT
$ sudo dpkg -i *.deb
```

The directory contains the following packages:

- eep-l4t-display-rt-kernel_<version>_arm64.deb
- eep-l4t-rt-kernel_<version>_arm64.deb
- eep-l4t-rt-initrd_<version>_arm64.deb
- eep-l4t-rt-kernel-oot-modules_<version>_arm64.deb

2. Reboot the EEP-150N:

```
$ sudo reboot
```

Kernel headers packages

Preparation

Before starting the installation, copy the kernel_headers package directory from the host PC to the EEP-150N (for example, using scp or a USB drive).

Installation

Follow the procedure that matches the kernel currently running on the EEP-150N.

If not running the RT kernel

1. Navigate to the kernel_headers directory and install the standard kernel-headers packages:

```
$ cd kernel_headers
```

```
$ sudo dpkg -i eep-l4t-kernel*.deb
```

The standard kernel-headers packages are:

- eep-l4t-kernel-headers_<version>_arm64.deb
- eep-l4t-kernel-oot-headers_<version>_arm64.deb

If running the RT kernel

1. Navigate to the kernel_headers directory and install the RT kernel-headers packages:

```
$ cd kernel_headers
$ sudo dpkg -i eep-l4t-rt-kernel*.deb
```

The RT kernel-headers packages are:

- eep-l4t-rt-kernel-headers_<version>_arm64.deb
- eep-l4t-rt-kernel-oot-headers_<version>_arm64.deb

Reboot

1. After either installation, reboot the EEP-150N:

```
$ sudo reboot
```

6. Install JetPack from SDK Manager

JetPack bundles the developer-side toolchain you need to build AI workloads for the EEP-150N—CUDA, cuDNN, TensorRT, VPI, and the broader Jetson SDK—and SDK Manager is NVIDIA’s graphical installer for it. This chapter walks through the standard four-step flow on your Ubuntu host. Note that we install the host-side components only; the EEP-150N already ships with its own provisioned image, so you should leave the “Jetson Linux image” and “Flash Jetson Linux” options unchecked when prompted.

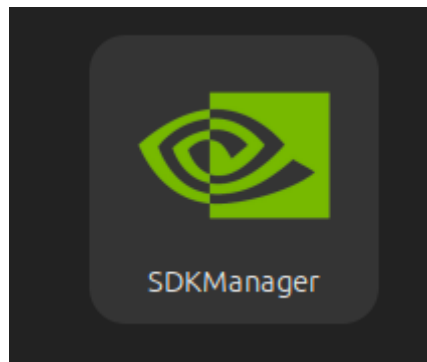
Install SDK Manager

Download SDK Manager from the NVIDIA developer site and install it on the host PC:

<https://developer.nvidia.com/sdk-manager>

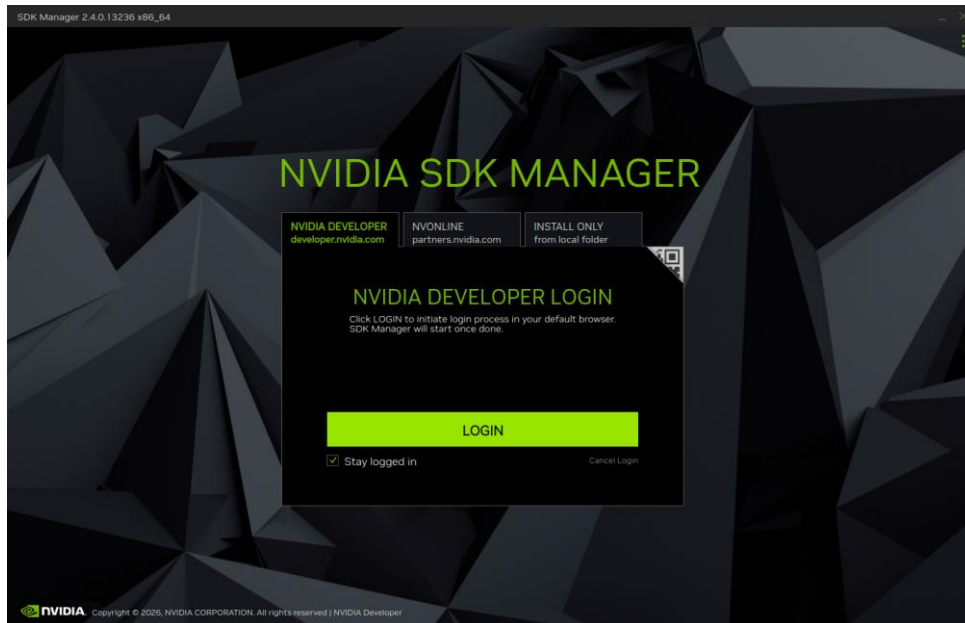
Run SDK Manager

Launch SDK Manager from the application menu:



Log in

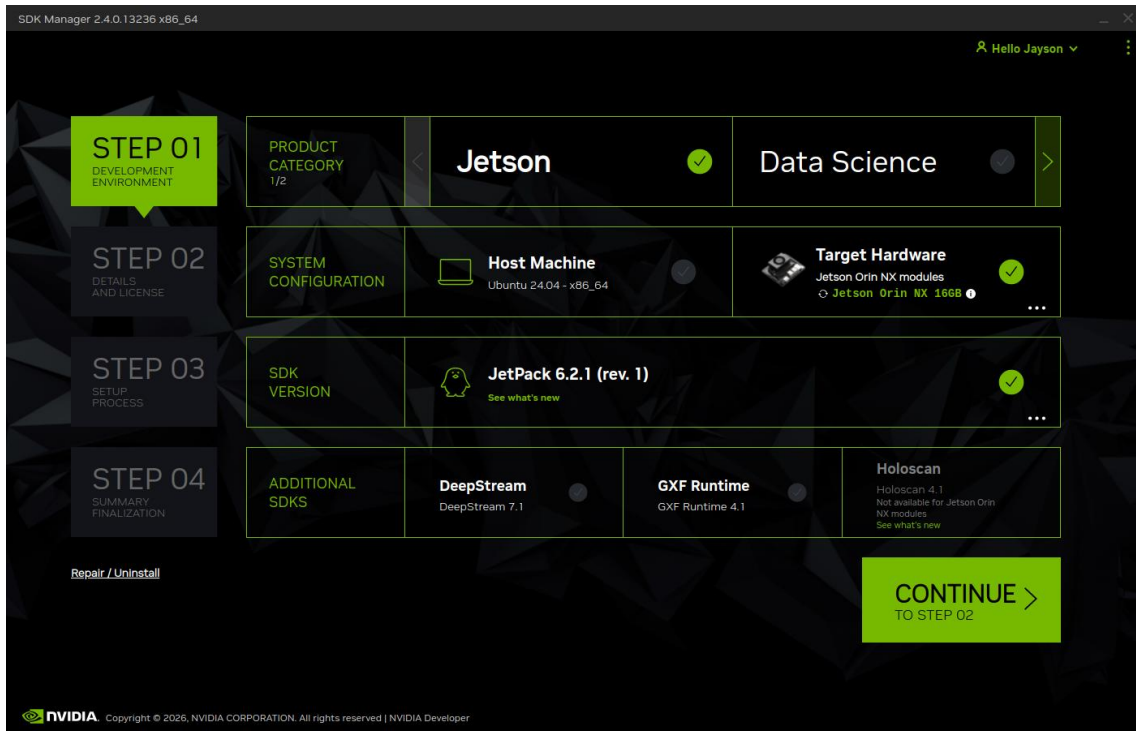
Sign in with an NVIDIA developer account. If one has not been created yet, follow the prompts on the login screen to register.



Step 01 — Set up the development environment

1. In the STEP 01 Development Environment window, configure the following:

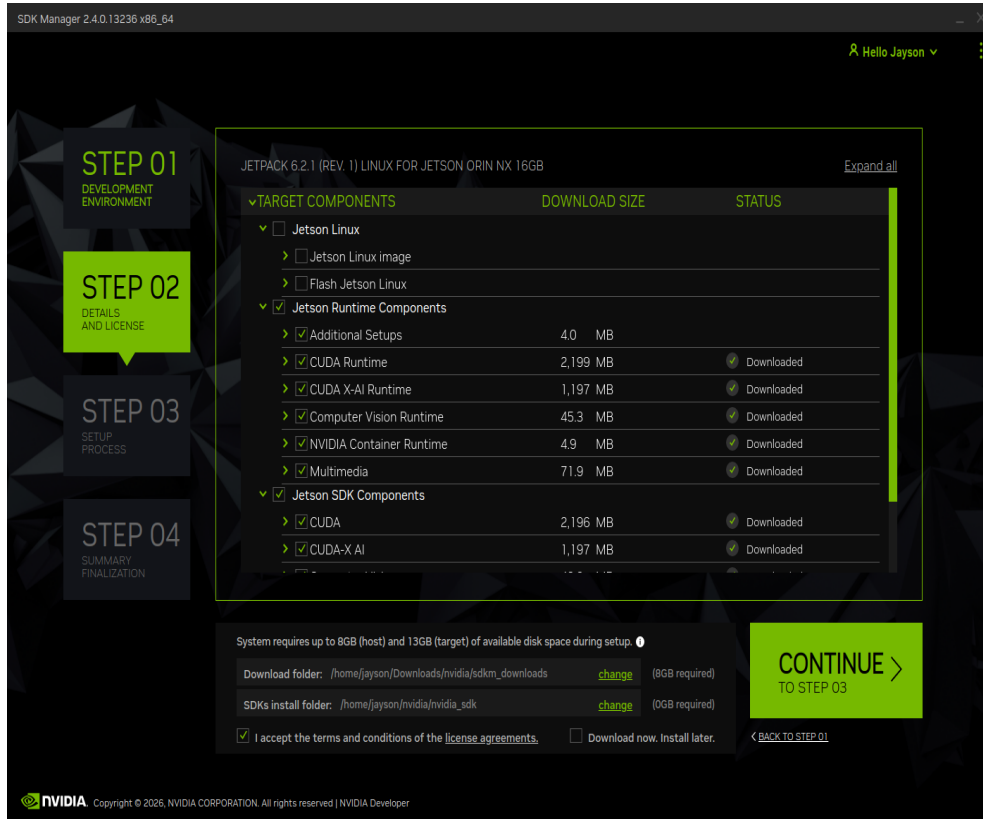
- Product Category — select Jetson.
- System Configuration — select the host machine, the target hardware, or both, according to your needs. If a Jetson device is connected, SDK Manager auto-selects it in the Target Hardware drop-down list. If multiple Jetson devices are connected or a device is not automatically detected, click Refresh. Hover over the detection message to see detailed information about the detected hardware.
- SDK Version — select "JetPack 6.2.1 (rev. 1)".
- Additional SDKs — select any additional SDKs to install, if relevant.



2. Click Continue to proceed to the next step.

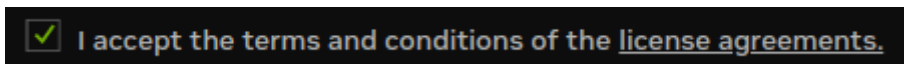
Step 02 — Review components and accept licenses

1. In STEP 02 Details and License, review the components to be installed as part of Jetson Platform Services. Some components can be selected or deselected.



Important: Do not select the Jetson Linux components for the EEP-150N. Leave both "Jetson Linux image" and "Flash Jetson Linux" unchecked.

2. Click the license agreements hyperlink at the bottom of the page to review the licenses.



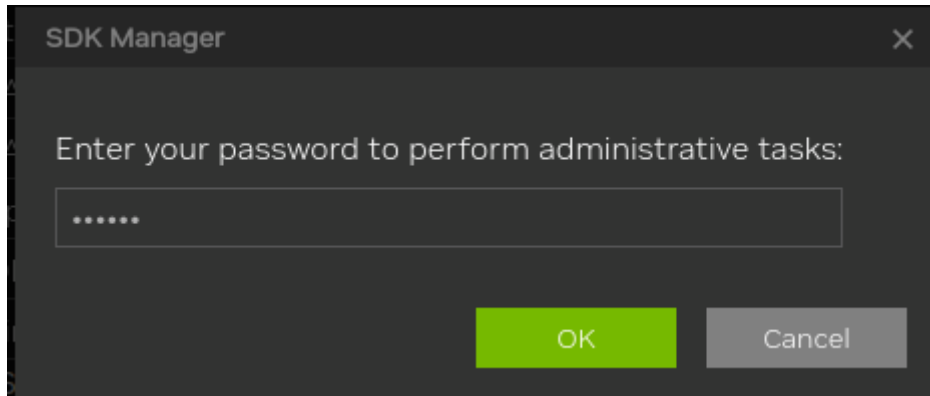
3. Select the checkbox to accept the terms and conditions of the license agreements.

4. Optional: change the download location. Navigate to the Download folder field at the bottom of the screen and update the path.

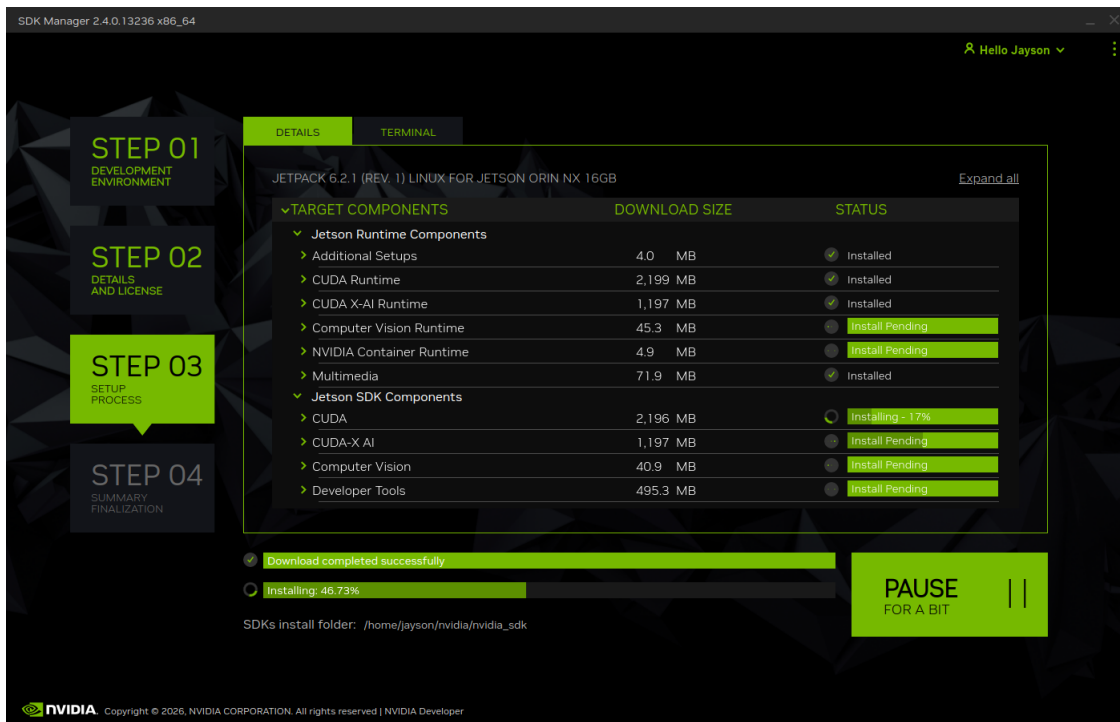
5. Click Continue to proceed to the next step.

Step 03 — Installation

1. Before the installation begins, SDK Manager prompts for the sudo password.

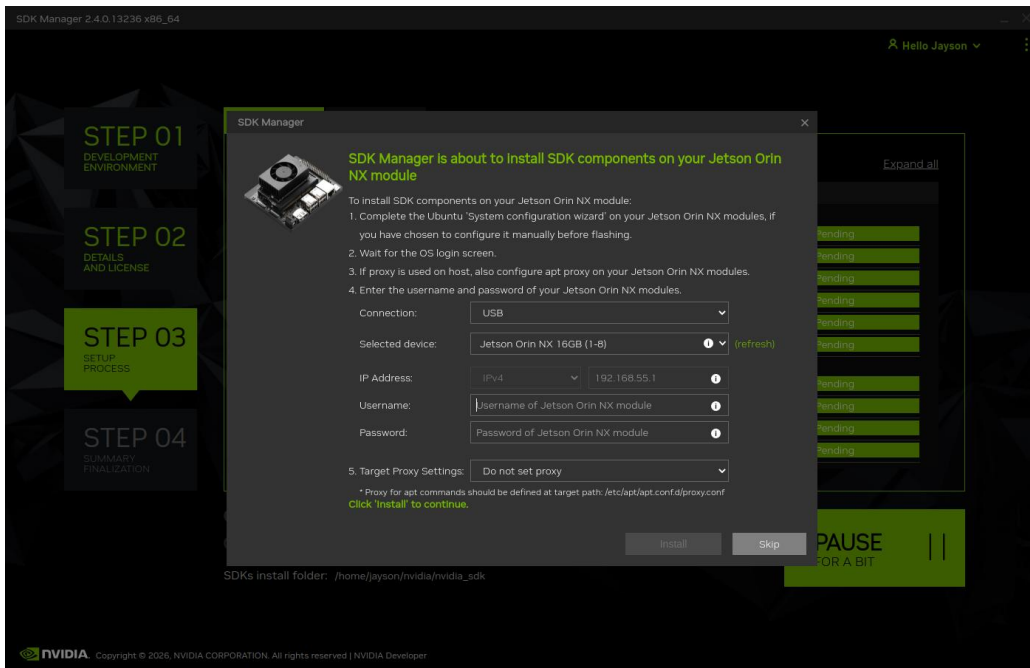


2. Monitor the download and installation progress.



Click Pause or Resume to toggle the download and installation process.

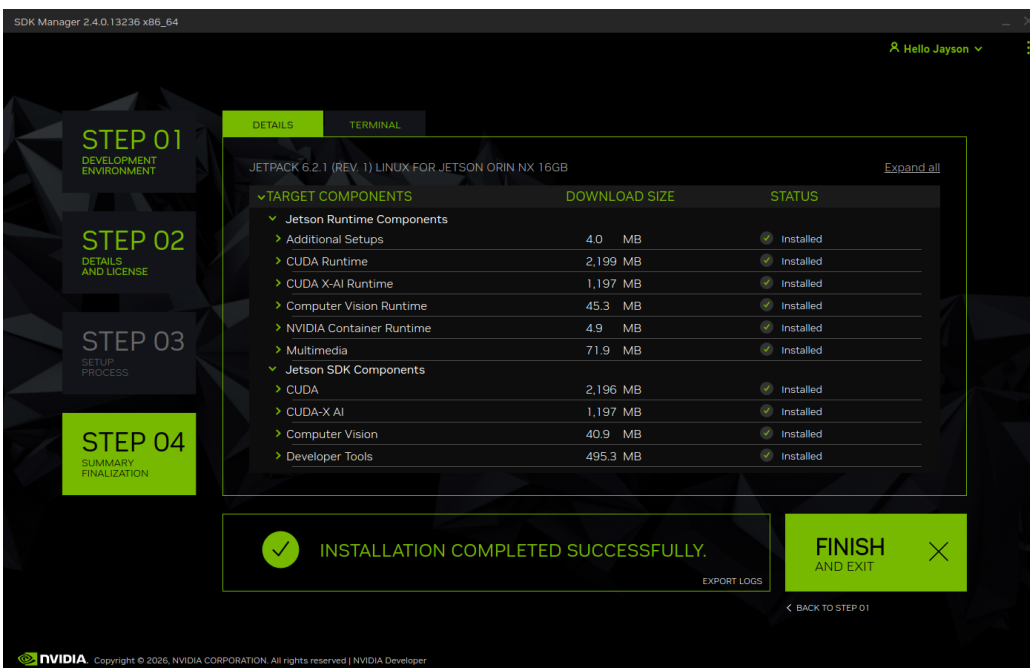
3. Use the tabs at the top of the window to switch between the Details and Terminal views. The Terminal tab shows detailed download and installation output with any errors highlighted.
4. On the Terminal tab, use the Filter text field to filter and search for specific information.
5. SDK Manager opens a dialog when it is ready to flash the target device. Follow the on-screen instructions to prepare the target.
6. After SDK Manager completes flashing, the monitor connected to the Jetson shows the initial-setup prompt. Complete the initial setup by selecting a user name and password for the Jetson system.
7. After initial setup, the Jetson system boots to the Linux desktop. Return to SDK Manager on the host PC and enter the same user name and password into SDK Manager’s post-flash installation dialog.



SDK Manager then completes the installation of the software libraries. Skipping this step keeps a clean operating system on the target and installs no SDK components.

Step 04 — Finalize setup

1. In STEP 04 Summary Finalization, review the summary of installed components and any warnings or errors.
2. Click Export Debug Logs to create a ZIP file of all log files generated during the installation. The ZIP file is saved to the same folder path used for the downloaded components.



Installation is complete. The host PC is now ready for JetPack-based development work.