

NVIDIA Jetson Xavier NX

We use the [NVIDIA Jetson Xavier NX development kit](#).

Join the [NVIDIA DEVELOPER Membership program](#) to get access to all NVIDIA resources.

NVIDIA Jetson Xavier NX resources:

- [User Guide](#)
- [Carrier Board Specification](#)
- [SDK Manager](#)
- [Jetson Linux Developer Guide](#)
- [Cloud-Native on Jetson](#)

[Jetsonhacks](#) provides useful information and [YouTube](#) videos on how to use the NVIDIA Jetson.

Hookup

Set up the sensors & processing board without the cover as described in the hardware assembly.

NVIDIA SDK Manager

We use the [NVIDIA SDK Manager](#) to flash the NVIDIA Jetson Xavier NX and to set up the **NVIDIA development environment** on the host computer and the NVIDIA Jetson Xavier NX.

Set up NVIDIA Jetson Xavier NX development board

Connect the USB dongle of the [Dual USB & Bluetooth Keyboard and Mouse](#) to a USB port of the NVIDIA Jetson Xavier NX Development Board.

Insert the [micro SD card](#) into the micro SD slot on the underside (facing the front of the SensorBox) of the NVIDIA Jetson Xavier NX module.

Use a [Circuit Jumper Cap](#) or [Jumper Wire](#) to place a jumper across the Force Recovery Mode pins 9 and 10 on the button header J14 on the NVIDIA Jetson Xavier NX Development Board.

Power on the NVIDIA Jetson Xavier NX by pushing the push button. The NVIDIA Jetson Xavier NX powers on in Force Recovery Mode.

Remove the jumper from the Force Recover Mode pins.

Download and run SDK Manager on the Host Computer

\$: lsusb

ID 0955:7020 NVidia Corp.

Firefox: <https://login.developer.nvidia.com>

> Click Account

> Enter Email and Password

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

> Download **NVIDIA SDK Manager 1.5** > .deb Ubuntu

> Save File

An already downloaded copy can be found here:

sw/Host_Computer/NVIDIA_SDKManager/**sdkmanager_1.5.0-7774_amd64.deb**

sw/Host_Computer/NVIDIA_SDKManager\$: sudo apt install ./sdkmanager_1.5.0-7774_amd64.deb

\$: sdkmanager

Ubuntu: Right-Click SDK Manager icon and select Add to Favorites

> Login to NVIDIA Developer

> Login & Password

> Step 01

> Select Target Hardware **Jetson Xavier NX [developer Kit version]** (detected)

> Select **Linux Jetpack 4.5.1**

> Check DeepStream

> Step 02

> Check I accept the terms and conditions of the license agreements.

> SUDO Password

Step 03 ... takes quite some time

> Select Manual Setup

> Flash

Configure Linux for Tegra (L4T)

Ubuntu System Configuration on NVIDIA Jetson Xavier NX

> Accept NVIDIA Jetson software EULA

> Select system language, keyboard layout, and time zone

> Your name: nx

> Your computer's name: nx

> Username:**nx**

> Password:**nx**

> Check Log in automatically

> Accept max APP Partition Size

> Mode_15W_6Core

Finish running the SDK Manager

SDK Manager on Host Computer

> Enter Username: nx

> Enter Password: nx

> Install ... takes quite some time

Step 04

> Finish and Exit

Power off/on the NVIDIA Jetson Xavier NX

\$: shutdown -h now

Cut the power off using the push button.

Power on the NVIDIA Jetson Xavier NX using the push button.

Update Software

Ubuntu: Software Updater

> Click Install Now

\$: sudo apt-get update

\$: sudo apt-get upgrade

Open Access

Ubuntu: Click on System Settings Icon (left)

> Click on Brightness & Lock

> Select Turn screen off when active for Never

> Select Lock Off

> Un-check Require my password when waking from suspense

Disable Keychain

When selecting log in automatically in the Ubuntu configuration, you will be asked to enter the keychain password each time after booting when you want to use the Chromium internet browser. If you set the password to blank this goes away.

Ubuntu: Search for password and keys

> Right-click Login > Change Password

> Enter the old password

> Enter a blank new password

Wifi

Ubuntu: Click on Network Icon (top right) and select network

> Enter password

Now you can disconnect the [USB cable](#) from your host computer.

Bluetooth Keyboard and Mouse

Ubuntu: Click on Bluetooth Icon (top right) and select Bluetooth Settings

> Click on + icon to search for new devices

Keyboard: Turn on

Keyboard: Press Bluetooth 1 key (top right) for 5 seconds

> Select Dual-mode Keyboard (using mouse)

> Next

> Click on + icon to search for new devices

Mouse: Turn on

Mouse: Use button to select Bluetooth 1

Mouse: Press button for 5 seconds

> Select Dual-mode Mouse (using Tab and arrows keys on the keyboard)

> Hit Enter

Unplug the USB dongle from the USB port of the NVIDIA Jetson Xavier NX.

SSD Drive

The NVIDIA Jetson Xavier NX does not allow yet to boot directly from the SSD drive, so we copy the rootfs from the SD card to the SSD drive and install a service that switches the rootfs to the SSD drive after booting. The NVIDIA Jetson Xavier NX is still booting from the SD card, but runs from the SSD drive. This setup should be done on fresh installs of the SD card.

We followed [JetsonHacks](#) instructions from [Run from SSD](#) and used their scripts from their GitHub [rootOnNVMe](#) repository.

The SSD drive was already installed during hardware assembly.

Format the SSD drive

Ubuntu: Disks

> Select 500 GB Disk (Samsung SSD 970 EVO Plus 500GB)

> Click menu icon top right and select Format Disk

> Click Format...

> Click Format

> Click on + icon to add a partition

> Select 500GB

> Enter Volume Name: ssd

> Check Internal disk for use with Linux systems only (Ext4)

> Click Create

Clone the scripts from JetsonHacks GitHub repository

sw/NVIDIA_Jetson_Xavier_NX/SSD\$: git clone <https://github.com/jetsonhacks/rootOnNVMe.git>

An already cloned repository can be found here:

sw/NVIDIA_Jetson_Xavier_NX/SSD/**rootOnNVMe**

Copy the rootfs from the SD card to the SSD drive

sw/NVIDIA_Jetson_Xavier_NX/SSD/rootOnNVMe\$: bash **copy-rootfs-ssd.sh**

Set up the service that switches the rootfs to the SSD drive after booting

sw/NVIDIA_Jetson_Xavier_NX/SSD/rootOnNVMe\$: bash **setup-service.sh**

\$: reboot

To start over with a fresh installation (basically your installation on the SD card) just remove the file /etc/setssdroot.conf from the SD card, which sets the rootfs back to the SD card after the next boot. Then follow these instruction to install the SSD drive again.

Power Management

Check out [power management](#) in the Jetson Linux Developer Guide.

We chose the 15W 6 CPU @1400MHz profile.

\$: **sudo /usr/sbin/nvpmmodel -m 2**

\$: sudo /usr/sbin/nvpmmodel -q

NV Fan Mode:quiet

NV Power Mode: MODE_15W_6CORE

2

System Monitoring

Check out [software-based power consumption modeling](#) in the Jetson Linux Developer Guide for using the INA3221 power monitor at I2C address 0x40.

Channel 0: System 5V

\$: cat /sys/bus/i2c/drivers/ina3221x/7-0040/iio\:device0/in_current0_input
976 [mA]

\$: cat /sys/bus/i2c/drivers/ina3221x/7-0040/iio\:device0/in_voltage0_input
5144 [mV]

\$: cat /sys/bus/i2c/drivers/ina3221x/7-0040/iio\:device0/in_power0_input
4979 [mW]

Channel 1: CPU & GPU combined power

\$: cat /sys/bus/i2c/drivers/ina3221x/7-0040/iio\:device0/in_current1_input
136 [mA]

```
$: cat /sys/bus/i2c/drivers/ina3221x/7-0040/iio\:device0/in_voltage1_input
5144 [mV]
$: cat /sys/bus/i2c/drivers/ina3221x/7-0040/iio\:device0/in_power1_input
699 [mW]
```

Channel 2: SoC power rail

```
$: cat /sys/bus/i2c/drivers/ina3221x/7-0040/iio\:device0/in_current2_input
248 [mA]
$: cat /sys/bus/i2c/drivers/ina3221x/7-0040/iio\:device0/in_voltage2_input
5136 [mV]
$: cat /sys/bus/i2c/drivers/ina3221x/7-0040/iio\:device0/in_power2_input
1314 [mW]
```

Check out [tegrastats utility](#) in Jetson Linux Developer Guide.

\$: tegrastats

```
RAM 2088/7766MB (lfb 612x4MB) SWAP 0/3883MB (cached 0MB) CPU
[13%@1420,14%@1420,8%@1420,14%@1420,14%@1420,9%@1420] EMC_FREQ 0%
GR3D_FREQ 0% AO@42C GPU@42C PMIC@100C AUX@43C CPU@42.5C thermal@42.55C
```

jtop

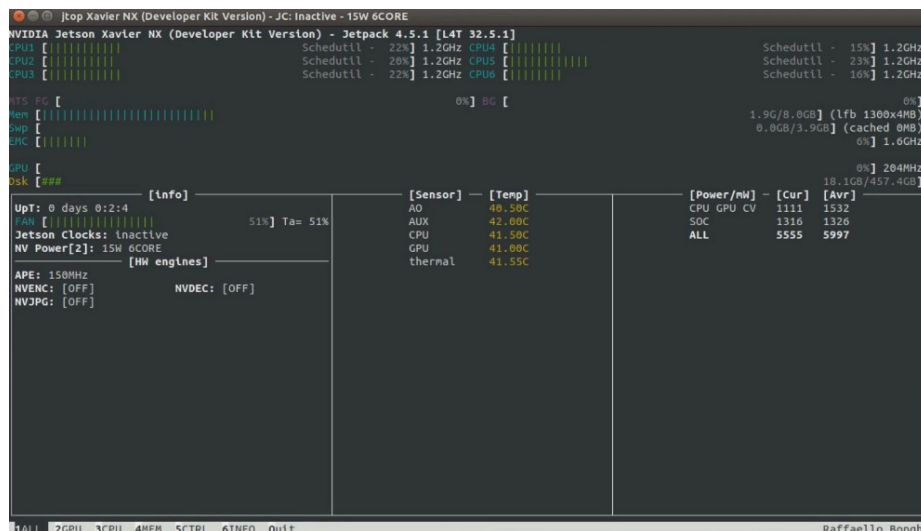
[Raffaello Bonghi](#) provides **jetson_stats**, a python package to monitor and control the Jetson platform, on [GitHub](#).

Install jetson-stats

```
$: sudo apt-get install python-pip
$: sudo -H pip install -U jetson-stats
reboot
```

Monitor the system

\$: jtop



Software Setup

Our SensorBox Repository

Clone our SensorBox repository from GitHub

DATA/projects\$: git clone <https://github.com/AIWerkstatt/SensorBox.git>

DATA/projects\$: cd **SensorBox/4-Setup**

Scripts

The scripts are located in the folder **sw/NVIDIA_Jetson_Xavier_NX/Scripts**.

Run all install scripts

sw/NVIDIA_Jetson_Xavier_NX/Scripts\$: bash **install.sh**

Reboot

Run Basic Ubuntu installation script

sw/NVIDIA_Jetson_Xavier_NX/Scripts\$: bash **install-1-Ubuntu.sh**

Docker

NVIDIA JetPack includes NVIDIA container runtime with [Docker](#) integration, enabling GPU accelerated containerized applications on the Jetson platform.

Create a [Docker identification](#).

The Dockerfiles are located in the folder **sw/NVIDIA_Jetson_Xavier_NX/Docker**.

Build a Docker image with all Dockerfiles

sw/NVIDIA_Jetson_Xavier_NX/Docker\$: bash **build-Docker.sh**

We based the development environment on the [NVIDIA's L4T Base Image](#).

Set up udev rules on Docker Host (only once)

sw/NVIDIA_Jetson_Xavier_NX/Docker\$: bash **install-udev_rules-Host.sh**

Reboot

Run the Docker container

sw/NVIDIA_Jetson_Xavier_NX/Docker\$: bash **run-Docker.sh**

The Docker container is run as a bash terminal (interactive mode) under the current user (same user id, group id and home folder), using the same network as the host computer and forwarding the display to the X server on the host computer with the proper access to the host computers resources.

Basic Ubuntu Dockerfile

sw/NVIDIA_Jetson_Xavier_NX/Docker/**Dockerfile-1-Ubuntu**