Getting started with :

<https://developer.nvidia.com/embedded/learn/get-started-jetson-nano-devkit>

Parts of the Jetson Orin Developer kit are:

1. microSD card slot for main storage
2. 40-pin expansion header
3. Micro-USB port for 5V power input, or for Device Mode
4. Gigabit Ethernet port
5. USB 3.0 ports (x4)
6. HDMI output port
7. DisplayPort connector
8. DC Barrel jack for 5V power input
9. MIPI CSI-2 camera connectors

**The items Included in the Box are**

Your Jetson Nano Developer Kit box includes:

* NVIDIA Jetson module and reference carrier board
* Small paper card with quick start and support information
* Folded paper stand

**The Items not Included in the Box are**

You’ll also need:

* microSD card (32GB UHS-1 minimum recommended)
* USB keyboard and mouse
* Computer display (HDMI or DP)
* Micro-USB power supply

Initially, a computer with Internet connection and the ability to flash your microSD card is also required.

How to **Prepare for Setup**

**Items for Getting Started**

**microSD Card**

The Jetson Nano Developer Kit uses a microSD card as a boot device and for main storage. It’s important to have a card that’s fast and large enough for your projects; the minimum recommended is a 32 GB UHS-1 card.

See the instructions below to flash your microSD card with operating system and software.

**Micro-USB Power Supply**

You’ll need to power the developer kit with a good quality power supply that can deliver 5V⎓2A at the developer kit’s Micro-USB port. Not every power supply promising “5V⎓2A” will actually do this.

As an example of a good power supply, NVIDIA has validated . It was specifically designed to overcome common problems with USB power supplies; see the linked product page for details.

**Note**

The stated power output capability of a USB power supply can be seen on its label.

Actual power delivery capabilities of USB power supplies do vary.

**How to Update the Firmware**

Your Jetson Orin Nano Developer Kit comes with an old firmware flashed at the factory, which is **NOT** compatible with JetPack 6.x.

Therefore, you **must** upgrade to the latest firmware, before inserting SD card flashed with JetPack 6.x image (unless the firmware previuosly updated).

**How to Update your firmware using SD card**

**You can upgrade your Jetson to the latest firmware without needing a host Ubuntu PC.**

How to do Initial Setup Guide for Jetson Orin Nano Developer Kit

The NVIDIA® Jetson Orin Nano™ Developer Kit is a perfect kit to start your journey of local generative AI evaluation and development.

With the December 2024 software update (JetPack 6.1 (rev.1)), this advanced edge computer delivers up to 70% more performance, making it an even more powerful platform for the era of generative AI.

Check your inventory

The following item is needed or highly desired to set up your Jetson Orin Nano Developer Kit.  
If you don't have them in your inventory, you want to arrange them and return to this guide once they are available.

Storage

* microSD card (64GB or bigger)
* NVMe SSD (Optional, but highly recommended for following tutorials on this site)

Means to access terminal

You need either of the following set:

* DisplayPort cable, DisplayPort capable monitor and a USB keyboard
* DisplayPort to HDMI cable and HDMI capable monitor (or TV) and a USB keyboard

Alternative method to setup Guide for Jetson Orin Nano Developer Kit using SDK Manager:

In case you have an x86 PC running Ubuntu 22.04 or 20.04, then you can flash your Jetson Orin Nano Developer Kit with the latest firmware and JetPack all at once using **NVIDIA SDK Manager**.

Also, if you wish to not use a microSD card but rather use a large NVMe SSD for the OS and data, then you need to use SDK Manager to flash the latest JetPack on the NVMe SSD.

How to do initial setup for Jetson Nano developer kit

The steps are :

1] check Jetson UEFI Firmware version > 36.0 or not. Determine QSPI update is necessary or not

2] Boot with JetPack 5.1.3 SD card to schedule firmware update

3]  Reboot and observe firmware update to 5.0

4] Run QSPI Updater package to schedule QSPI update

5]  Reboot, observe QSPI update, and power off

6] Boot with JetPack 6.x SD card

7] Reboot and observe firmware update to 36.4.3

8]  Unlock Super Performance

How to check Jetson UEFI Firmware version > 36.0

Your Jetson Orin Nano Developer Kit may have the latest firmware ("Jetson UEFI firmware" on QSPI-NOR flash memory) flashed at the factory.

If not, we need to go through a set of procedures to upgrade to the latest firmware.

So let's first check the version of your Jetson UEFI Firmware.  
You can take one of the following methods.

A] When monitor is attached : The steps are

1. Connect your monitor and USB keyboard to your developer kit.
2. Turn on the developer kit by plugging in the bundled DC power supply
3. Repeatedly press  Esc key on the keyboard, especially after NVIDIA logo boot splash screen first appears on the monitor
4. You should see UEFI setup menu screen
5. Check the third line from the top (below "Not specified"), which should be the version number of Jetson UEFI firmware

B] For Headless setup

\* RXD (Pin 3)  Adafruit adaptor cable Green

\* TXD (Pin 4)  Adafruit adaptor cable White

\* GND (Pin 7)  Adafruit adaptor cable Black

Connect USB to TTL Serial cable onto the following pins on J14 "button" header of carrier board located under the Jetson module.

1. On your PC, run your console monitor program and open the USB serial port.
2. Power on the developer kit by plugging in the bundled DC power supply
3. On the PC console, repeatedly press  Esc key on the keyboard, especially after NVIDIA logo boot splash screen first appears on the monitor
4. You should see UEFI setup menu screen
5. Check the third line from the top (below "Not specified"), which should be the version number of Jetson UEFI firmware

How to determine Determine QSPI update is necessary or not

If you found your Jetson Orin Nano needs its firmware updated to run JetPack 6.x, click “Firmware < 36.0” tab, and then you will have to do additional steps.

If you know your Jetson Orin Nano has the latest firmware, stay on “Firmware 36.x”tab, and skip to boot with Jetpack 6.x sd card.

How to Boot with JetPack 5.1.3 SD card to schedule firmware update

First, we need to run JetPack 5.1.3 in order to let its nvidia-l4t-bootloader package get its bootloader/firmware updater activated, so that the firmware update automatically runs the next time it reboots.

1. Download SD card image on to your PC

On your PC, download JetPack 5.1.3 image for Jetson Orin Nano Developer Kit from the official Jetpack 5.1.3 page.

1. Use Balena Etcher to flash image to SD card

If you don't have Balena Etcher on your PC, download from Balena Etcher official website.

1. Insert the flashed microSD card into the slot on Jetson module
2. Power-on : Turn on the Jetson Orin Nano Developer Kit **with JetPack 5.1.3 SD card inserted**by plugging in the DC power supply.
3. Complete the initial software setup ( oem-config)
4. Ensure firmware update is scheduled.
5. Once Jetson boots into Jetson Linux system, a background service automatically runs to schedule a firmware update (if needed) to be performed during the next boot-up process. Once you see the following, or just wait about 5 minutes after powering on to ensure the scheduling is done, reboot.

7a] If on GUI no steps further

7b] If on CLI use command : sudo systemctl status nv-l4t-bootloader-config

How to perform Reboot and observe firmware update to 5.0

1. Reboot : Reboot your Jetson Orin Nano Developer Kit.

1a] In GUI , press the power button and then click on restart

1b] In CLI use command sudo reboot and your firmware will get updated.

How to run QSPI Updater package to schedule QSPI update

When your UEFI firmware is updated to 35.5.0 ( = JetPack 5.1.3 ), it is capable of updating the entire QSPI content to make it ready for JetPack 6.x.

We will run a special tool so that the entire QSPI update is scheduled to run automatically on the next boot-up.

1. Double-check your firmware version is up to date ( 35.5.0 = JetPack 5.1.3)

Once it reboots back into Jetson Linux system, on Jetson terminal, run the following

sudo nvbootctrl dump-slots-info

1. Install QSPI Updater Debian package to trigger the entire QSPI update. On Jetson terminal run the following command:

sudo apt-get install nvidia-l4t-jetson-orin-nano-qspi-updater

Installing the nvidia-l4t-jetson-orin-nano-qspi-updater automatically runs its script to schedule the entire QSPI update to be performed during the next boot process, so that the firmware is ready for JetPack 6.x

How to Reboot, observe QSPI update, and power off during Jetpack update

1. Reboot

Once the QSPI update is scheduled, reboot your Jetson Orin Nano Developer Kit.

1. Observe update

You can observe the QSPI update during the boot up process.

1. Power off

Once the update is done, it reboots and tries to boot, however it will get stuck **UNLESS you change the SD card to JetPack 6.x one**.

Therefore you should just power off the developer kit simply by disconnecting the DC power supply.

How to Boot with JetPack 6.x SD card

Once we know the onboard firmware is up-to-date and ready for JetPack 6.x, we can boot Jetson Orin Nano Developer Kit with a microSD card for JetPack 6.2.

1. Download SD card image on to your PC

On your PC, download the latest JetPack 6.x image for Jetson Orin Nano Developer Kit from the official Jetson page.

Use Balena Etcher to flash image to SD card

Insert your microSD card into your PC's SD card slot, and use Balena Etcher to flash the SD card with the image you just downloaded.

If you don't have Balena Etcher on your PC, download from Balena official site and follow the below steps:

1. Insert the **JetPack 6.2**microSD card into the slot on Jetson module  
   Power-on by plugging the DC power supply
2. Complete the initial software setup ( oem-config )
3. Ensure firmware update is scheduled.

JetPack 6.2 SD card will automatically schedule another firmware/bootloader update in the next (3rd and final) reboot cycle

Reboot and observe firmware update to 36.4.3

1. Reboot your Jetson Orin Nano Developer Kit on JetPack 6.2 SD card to trigger the final firmware update (to 36.4.3 ).

1a] On GUI click on power button then restart.

1b] On CLI use command sudo reboot

How to Unlock Super Performance

Note that the default power mode on JetPack 6.2 on Jetson Orin Nano Developer Kit is **25W**.  
To switch to the **MAXN SUPER**mode and unlock the unregulated performance, follow below steps:

1. Click on the current power mode ( **25W**) by clicking the NVIDIA icon on the right side of the Ubuntu desktop’s top bar.
2. Select **Power mode**from the menu.
3. Choose **MAXN SUPER**to enable maximum performance.

What is overall flow only microSD-only method:

1️] Start and Check if Jetson UEFI Firmware is newer than version 36.0

2] If not then:

2A] Boot with JetPack 6.2 microSD card  
to schedule firmware update

2B] Reboot the device, then firmware is updated to 5.0 during reboot

2C] Run QSPI Updater and reboot the device and proceed to step

3] If yes then continue .

Boot with JetPack 6.2 microSD card to schedule firmware update and then reboot the device.

4] Unlock super performance and start developing on JetPack 6.2

Initial Setup using SDK Manager for Jetson Developer Kit:

The following item is needed or highly desired for flashing your Jetson Orin Nano Developer Kit using your host PC :   
a) Host PC : A  x86 PC running **Ubuntu 22.04**or **Ubuntu 20.04**

b) Storage : Either of the following.

* microSD card (64GB or bigger)
* NVMe SSD (Recommended for better performance)

c) Flashing supply

* USB cable (USB-C plug for Jetson Orin Nano Developer Kit side, other end depends on your PC)
* jumper pin (or metal paper clip)

Mean to access terminal

You need either of the following set:

* DisplayPort cable, DisplayPort capable monitor and a USB keyboard
* DisplayPort to HDMI cable and HDMI capable monitor (or TV) and a USB keyboard
* USB to TTL serial cable (Advanced)

Steps for Initial Setup using SDK Manager for Jetson Nano Developer kit:

0] Install SDK Manager

1️]Connect Target Hardware

2️]Select Software Components to Install

3️]Download on Host PC

4️]Flash

5️]Boot and initial setup and Start developing on JetPack 6.2

Step 0 : Install SDK Manager:

On Ubuntu terminal , run the commands according to the Ubuntu version: Here is for version 22.2.  
wget https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2204/x86\_64/cuda-keyring\_1.1-1\_all.deb

sudo dpkg -i cuda-keyring\_1.1-1\_all.deb

sudo apt-get update

sudo apt-get -y install sdkmanager

Then launch SDK Manager with the following command.

sdkmanager

Then login with your NVIDIA account .

Step 1 :  Connect the Target Hardware

Once SDK Manager is launched, connect your Jetson developer kit to your Ubuntu PC and power it on in Forced Recovery mode.

1. Connect your Jetson Orin Nano Developer Kit to your PC with a USB cable.  
   Note that the USB cable goes into the USB-C port of the carrier board of the developer kit.
2. While shorting pin 9 and pin 10 of J14 header located below the Jetson module using a jumper pin, insert the DC power supply plug into the DC jack of the carrier board to power it on.
3. Select " **Jetson Orin Nano [8GB developer kit version]**" and hit " **OK**"
4. Uncheck " **Host Machine**"  and continue to next step

Step 2: Select Software Components to Install]

1. Leave the only " **Jetson Linux**" component checked, and uncheck everything.
2. Remember to also check " **I accept the terms and conditions of the license agreements**".
3. Click " **Continue**" button to proceed to the next step.
4. It will prompt for the sudo command password.

Step 3 : Download on Host PC

1. It will start downloading the "BSP" package and "RootFS" package.
2. Once downloads are complete, it will untar the package and start generating the images to flash in the background.
3. Once images are ready, SDK it will open the prompt for flashing.

Step 4 : Flash

1. On the flashing prompt, select " **Runtime**" for "OEM Configuration".
2. On the flashing prompt, select " **NVMe**" if you want to flash Jetson Linux (BSP) to NVMe SSD.
3. Click "Flash" and the process will begin

Step 5 : Boot and Initial Setup

1. If still plugged, remove the jumper from header (that was used to put it in Forced Recovery mode)
2. Connect the DisplayPort cable or adapter and USB keyboard and mouse to Jetson Orin Nano Developer Kit, or hook up the USB to TTL Serial cable.
3. Unplug the power supply and put back in to power cycle.
4. Jetson should now boot into the Jetson Linux (BSP) of your selected JetPack version from the storage of your choice.
5. Complete the initial software setup ( oem-config )

How to Switch to MAXN SUPER mode

Note that the default power mode on JetPack 6.2 on Jetson Orin Nano Developer Kit is **25W**.  
To switch to the **MAXN SUPER**mode and unlock the unregulated performance, follow below steps:

1. Click on the current power mode ( **25W**) by clicking the NVIDIA icon on the right side of the Ubuntu desktop’s top bar.
2. Select **Power mode**from the menu.
3. Choose **MAXN SUPER**to enable maximum performance.

How to do Initial Setup for Jetson Developer kit using SSD+Docker:  
Step 1: SSD Physical installation

1. Unplug power and any peripherals from the Jetson developer kit.
2. Physically install an NVMe SSD card on the carrier board of your Jetson developer kit, making sure to properly seat the connector and secure with the screw.
3. Reconnect any peripherals, and then reconnect the power supply to turn on the Jetson developer kit.
4. Once the system is up, verify that your Jetson identifies a new memory controller on PCI bus using command lspci.

Step 2: Format and set up auto-mount

1. Run lsblk to find the device name.
2. Format the SSD, create a mount point, and mount it to the filesystem.

Using commands given below one by one :

sudo mkfs.ext4 /dev/nvme0n1

sudo mkdir /ssd

sudo mount /dev/nvme0n1 /ssd

1. In order to ensure that the mount persists after boot, add an entry to the fstab file using the command

lsblk -f

1. Then, add a new entry to the fstab file using command

sudo vi /etc/fstab

1. Insert the following line, replacing the UUID with the value found from lsblk -f :

UUID=\*\*\*\*\*\*\*\*\*\*\*\*-\*\*\*\*-\*\*\*\*-\*\*\*\*-\*\*\*\*\*\*\*\* /ssd/ ext4 defaults 0-2

1. Finally, change the ownership of the /ssd directory using command:

sudo chown ${USER}:${USER} /ssd

Step 3 : Docker

1. Install nvidia-container package using commands:  
   sudo apt update && sudo apt install -y nvidia-container

1b] For Jetpack users:

If you flash \*\*Jetson Linux (L4T) R36.x (JetPack 6.x) on your Jetson using SDK Manager, and install nvidia-container using apt , on JetPack 6.x it no longer automatically installs Docker.

Therefore, you need to run the following to manually install Docker and set it up.

sudo apt update

sudo apt install -y nvidia-container curl

curl https://get.docker.com | sh && sudo systemctl --now enable docker

sudo nvidia-ctk runtime configure --runtime=docker

1. Restart the Docker service and add your user to the docker group, so that you don't need to use the command with sudo .

sudo systemctl restart docker

sudo usermod -aG docker $USER

newgrp docker

1. Add default runtime in /etc/docker/daemon.json using either one of the commands:

3a]sudo apt install -y jq

3b] sudo jq '. + {"default-runtime": "nvidia"}' /etc/docker/daemon.json | \

3c] sudo tee /etc/docker/daemon.json.tmp && \

3d] sudo mv /etc/docker/daemon.json.tmp /etc/docker/daemon.json

1. Restart Docker
2. sudo systemctl daemon-reload && sudo systemctl restart docker

Step 4 : Migrate Docker directory to SSD

Now that the SSD is installed and available on your device, you can use the extra storage capacity to hold the storage-demanding Docker directory.

1. Stop the Docker service using command sudo systemctl stop docker
2. Move the existing Docker folder using commands:

sudo du -csh /var/lib/docker/ && \

sudo mkdir /ssd/docker && \

sudo rsync -axPS /var/lib/docker/ /ssd/docker/ && \

sudo du -csh /ssd/docker/

1. Edit /etc/docker/daemon.json file using command

sudo vi /etc/docker/daemon.json

Insert "data-root" line like the following.

{

"runtimes": {

"nvidia": {

"path": "nvidia-container-runtime",

"runtimeArgs": []

}

},

"default-runtime": "nvidia",

"data-root": "/ssd/docker"

}

1. Rename the old Docker data directory using command   
   sudo mv /var/lib/docker /var/lib/docker.old
2. Restart the docker daemon using commsnd :  
   sudo systemctl daemon-reload && \

sudo systemctl restart docker && \

sudo journalctl -u docker

Step 5: Test Docker on SSD

1. [Terminal 1] First, open a terminal to monitor the disk usage while pulling a Docker image.

watch -n1 df

1. [Terminal 2] Next, open a new terminal and start Docker pull using command docker pull ubuntu:22.04
2. [Terminal 1] Observe that the disk usage on /ssd goes up as the container image is downloaded and extracted.

Step 6 : Final Verification

Reboot your Jetson, and verify that you observe the following:

sudo blkid | grep nvme

df -h  
docker info | grep Root  
sudo ls -l /ssd/docker/  
sudo du -chs /ssd/docker/  
docker info | grep -e "Runtime" -e "Root"