



# WITCHKING ALPHA MANUAL

Version: 0.1

Date: March 2, 2025

Initials: Edvardas Vysniauskas

# Chapter 1

## Introduction

# Chapter 2

## OS Upgrade

The Jetpack for Jetson Nano comes with Ubuntu 18.04. There are no official way of upgrading it, however, the community has created a few guides how to make this work. The following guide is made based on [1].

### 2.1 Install Jetpack

First, we need to install the latest available JetPack 4.6(L4T 32.6.1). This is the latest downloadable version. However, this will not be the final version before the upgrade. To download the JetPack, use either official source (<https://developer.nvidia.com/embedded/jetpack-sdk-46>) or from OneDrive. Use Balena Etcher to flash the image onto the SD card that you will use for the Jetson Nano. Once done, boot the Jetson Nano with the flashed SD card.

Once logged in, perform the usual *sudo apt update* and *sudo apt upgrade*. Once finished, you should have the latest R32.7.6 JetPack. This is required to perform the upgrade of the OS, and especially if booting from external drive is required (see <https://jetsonhacks.com/2021/03/10/jetson-nano-boot-from-usb/>).

### 2.2 Upgrading from Ubuntu 18.04 to Ubuntu 20.04

Once you finish upgrading minor version of the JetPack, perform the OS upgrade.

- Remove Chromium browser and autoremove redundant packages
- Allow release upgrades
- Upgrade to Ubuntu 20.04
- GCC changes

#### 2.2.1 Remove Chromium browser and autoremove redundant packages

The upgrade requires for the Chromium browser to be removed. This is mainly due to Snap installs that will be performed later and the browser can interfere with them (and firefox comes with the upgrade, therefore in the worst case, it is still possible to use the internet browser). To do this, run *sudo apt-get remove --purge chromium-browser chromium-browser-l10n* command. Once it is complete run the following commands:

```
1 # refresh your system
  sudo apt-get update
3 # need nano for editing some files
  sudo apt-get install nano
```

```
5 | sudo apt-get upgrade
   | sudo apt-get autoremove
```

Next, enable distribution upgrades by setting `prompt=normal` in the `/etc/update-manager/release-upgrades` file. Again perform regular update-upgrade commands:

```
jetson@nano: ~
GNU nano 2.9.3 /etc/update-manager/release-upgrades Modified

# Default behavior for the release upgrader.

[DEFAULT]
# Default prompting behavior, valid options:
#
# never - Never check for, or allow upgrading to, a new release.
# normal - Check to see if a new release is available. If more than one new
#          release is found, the release upgrader will attempt to upgrade to
#          the supported release that immediately succeeds the
#          currently-running release.
# lts - Check to see if a new LTS release is available. The upgrader
#        will attempt to upgrade to the first LTS release available after
#        the currently-running one. Note that if this option is used and
#        the currently-running release is not itself an LTS release the
#        upgrader will assume prompt was meant to be normal.
Prompt=normal
```

Figure 2.1: Enable release upgrades.

```
# refresh your system again
2 | sudo apt-get update
   | sudo apt-get dist-upgrade
4 | sudo reboot
```

## 2.2.2 Upgrade to Ubuntu 20.04

To perform the actual upgrade run `sudo do-release-upgrade` command. This process will take a while, and some prompts will require user input. Always take the default value, as in some cases the upgrade can become unstable once it is finished (depending on the selection). Once it is finished it will ask to reboot, but **do not perform the reboot!!!** Before the reboot, we need to make changes to a few files. Firstly, check that `WaylandEnable=false` is uncommented in the

```
jetson@nano: ~
Removing gconf-service (3.2.6-6ubuntu1) ...
Removing gconf-service-backend (3.2.6-6ubuntu1) ...
Removing libgconf-2-4:arm64 (3.2.6-6ubuntu1) ...
Removing gconf2-common (3.2.6-6ubuntu1) ...
Processing triggers for desktop-file-utils (0.24-1ubuntu3) ...
Processing triggers for mime-support (3.64ubuntu1) ...
Processing triggers for gnome-menus (3.36.0-1ubuntu1) ...
Processing triggers for libglib2.0-0:arm64 (2.64.6-1~ubuntu20.04.4) ...
Processing triggers for libc-bin (2.31-0ubuntu9.2) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for sgml-base (1.29.1) ...
Processing triggers for menu (2.1.47ubuntu4) ...
Processing triggers for bamfdaemon (0.5.3+18.04.20180207.2-0ubuntu2) ...
Rebuilding /usr/share/applications/bamf-2.index...

System upgrade is complete.

Restart required

To finish the upgrade, a restart is required.
If you select 'y' the system will be restarted.

Continue [yN] N ← Don't reboot now!
jetson@nano:~$
```

Figure 2.2: Final screen. Do not reboot!

/etc/gdm3/custom.conf file. Next, uncomment *Driver "nvidia"* in the file /etc/X11/xorg.conf. Finally, reverse the dist upgrade setting in the 2.1 by setting Prompt to *never*. Once done, you can reboot the system. Once the system boots, remove certain directories that can cause problems. Firstly, delete the /usr/share/vulkan/icd.d directory:

```
# remove icd.d
2 sudo rm -rf /usr/share/vulkan/icd.d
```

Then remove other redundant symbolink links in /usr/share/applications and other annoyances:

```
# prepare your system
2 sudo apt-get update
  sudo apt-get upgrade
4 sudo apt-get autoremove
# remove circular symlink
6 sudo rm /usr/share/applications/vpi1-demos
# remove distorted nvidia logo in top bar
8 cd /usr/share/nvpmmodel_indicator
  sudo mv nv_logo.svg no_logo.svg
```

Lastly, re-enable the original NVIDIA repositories in `/etc/apt/sources.list.d` directory. Do the following change to the all existing (should be five) files.

```
jetson@nano: /etc/apt/sources.list.d
jetson@nano:~$ cd /etc/apt/sources.list.d
jetson@nano:/etc/apt/sources.list.d$ ls -l
total 20
-rw-r--r-- 1 root root 74 sep 20 13:57 cuda-l4t-10-2-local.list
-rw-r--r-- 1 root root 184 sep 20 13:57 nvidia-l4t-apt-source.list
-rw-r--r-- 1 root root 136 sep 20 13:58 visionworks-repo.list
-rw-r--r-- 1 root root 144 sep 20 13:58 visionworks-sfm-repo.list
-rw-r--r-- 1 root root 154 sep 20 13:58 visionworks-tracking-repo.list
jetson@nano:/etc/apt/sources.list.d$ sudo nano visionworks-repo.list
```

```
GNU nano 4.8 visionworks-repo.list Modified
deb-src file:///var/visionworks-repo / # disabled on upgrade to focal
# deb file:///var/visionworks-repo / # disabled on upgrade to focal
```

Remove the hash to uncomment the lines

Figure 2.3: Reactivate NVIDIA repositories.

### 2.2.3 GCC changes

Ubuntu 20.04 comes with GCC 9, but some CUDA packages require GCC 8. To do this, make the following changes to enable both of the versions (perform the same for the clang8):

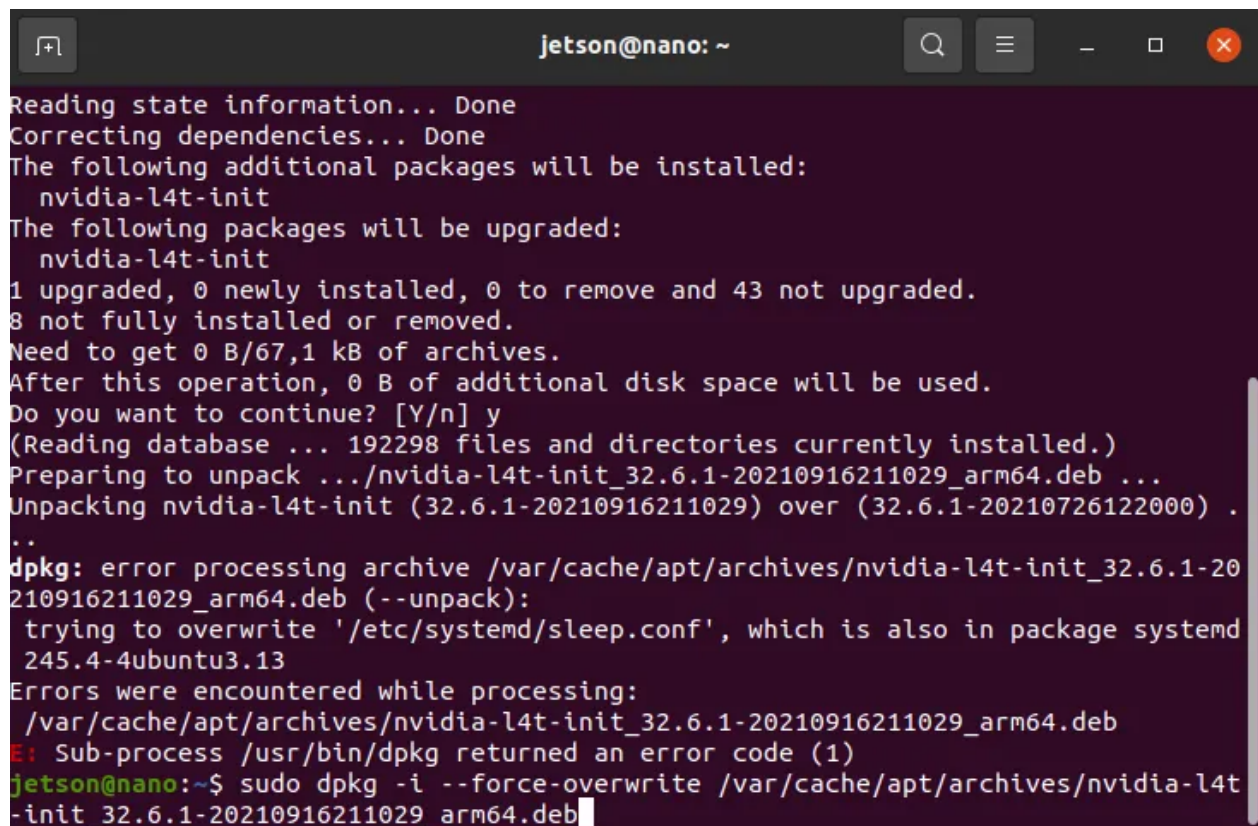
```
1 # install gcc and g++ version 8
  sudo apt-get install gcc-8 g++-8
3 # setup the gcc selector
  sudo update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-9 9
5 sudo update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-8 8
  # setup the g++ selector
7 sudo update-alternatives --install /usr/bin/g++ g++ /usr/bin/g++-9 9
  sudo update-alternatives --install /usr/bin/g++ g++ /usr/bin/g++-8 8
9 # if you want to make a selection use these commands
  sudo update-alternatives --config gcc
11 sudo update-alternatives --config g++
```



---

## 2.2.4 Troubleshooting

It may come a time when you fail to upgrade the existing packages. The problem is the wrong version of the `nvidia-l4t-init` file. The command `sudo apt --fix-broken install` gives you the following information. In this case, `/etc/systemd/sleep.conf` is blocking the upgrade. The easiest solution is to force the upgrade with the command `sudo dpkg -i --force-overwrite`. Once done, the upgrade option will work as expected.

A terminal window titled 'jetson@nano: ~' with standard window controls. The terminal output shows the process of upgrading 'nvidia-l4t-init'. It lists additional packages to be installed and packages to be upgraded. It shows the progress of unpacking the new version (32.6.1-20210916211029\_arm64.deb) over the old one (32.6.1-20210726122000). The process fails with a 'dpkg: error processing archive' message, stating it's trying to overwrite '/etc/systemd/sleep.conf' which is also in the 'systemd' package. The error code returned is (1). The prompt then shows the user running the command to force the overwrite.

```
jetson@nano: ~  
Reading state information... Done  
Correcting dependencies... Done  
The following additional packages will be installed:  
  nvidia-l4t-init  
The following packages will be upgraded:  
  nvidia-l4t-init  
1 upgraded, 0 newly installed, 0 to remove and 43 not upgraded.  
8 not fully installed or removed.  
Need to get 0 B/67,1 kB of archives.  
After this operation, 0 B of additional disk space will be used.  
Do you want to continue? [Y/n] y  
(Reading database ... 192298 files and directories currently installed.)  
Preparing to unpack .../nvidia-l4t-init_32.6.1-20210916211029_arm64.deb ...  
Unpacking nvidia-l4t-init (32.6.1-20210916211029) over (32.6.1-20210726122000) .  
..  
dpkg: error processing archive /var/cache/apt/archives/nvidia-l4t-init_32.6.1-20210916211029_arm64.deb (--unpack):  
  trying to overwrite '/etc/systemd/sleep.conf', which is also in package systemd 245.4-4ubuntu3.13  
Errors were encountered while processing:  
  /var/cache/apt/archives/nvidia-l4t-init_32.6.1-20210916211029_arm64.deb  
E: Sub-process /usr/bin/dpkg returned an error code (1)  
jetson@nano:~$ sudo dpkg -i --force-overwrite /var/cache/apt/archives/nvidia-l4t-init_32.6.1-20210916211029_arm64.deb
```

Figure 2.4: Broken apt upgrade screen.

## Chapter 3

# Camera Calibration

Camera calibration is a necessity for removing distortion from the image. Although there are already made tools for calibrating the camera, such as kalibr [2]



# Bibliography

- [1] Bob Davis. Install ubuntu 20.04 on jetson nano. <https://qengineering.eu/install-ubuntu-20.04-on-jetson-nano.html>, 2021.
- [2] P. Furgale, H. Sommer, J. Maye, J. Rehder, T. Schneider, L. Oth. Kalibr - toolbox for calibration problems. <https://github.com/ethz-asl/kalibr>, 2014.