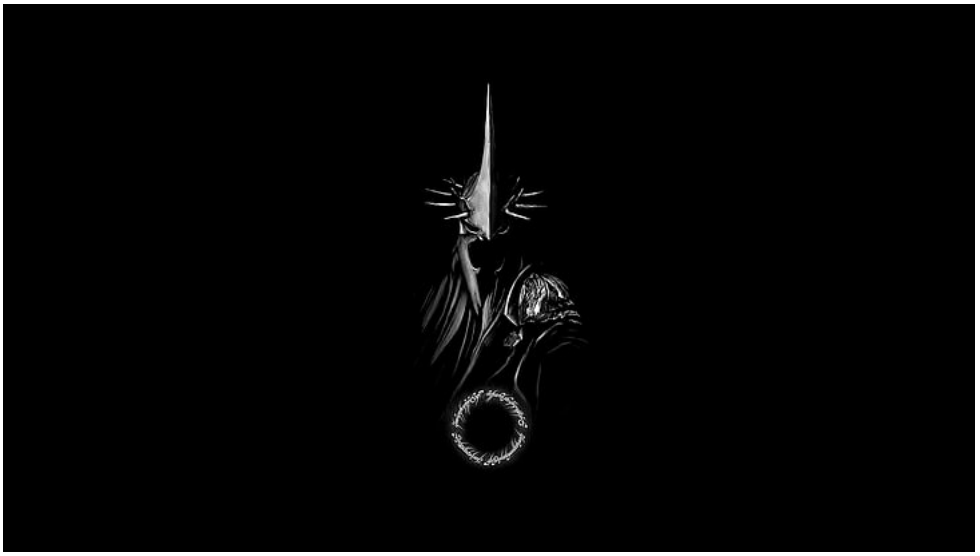

WITCHKING ALPHA MANUAL

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:

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

```
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:

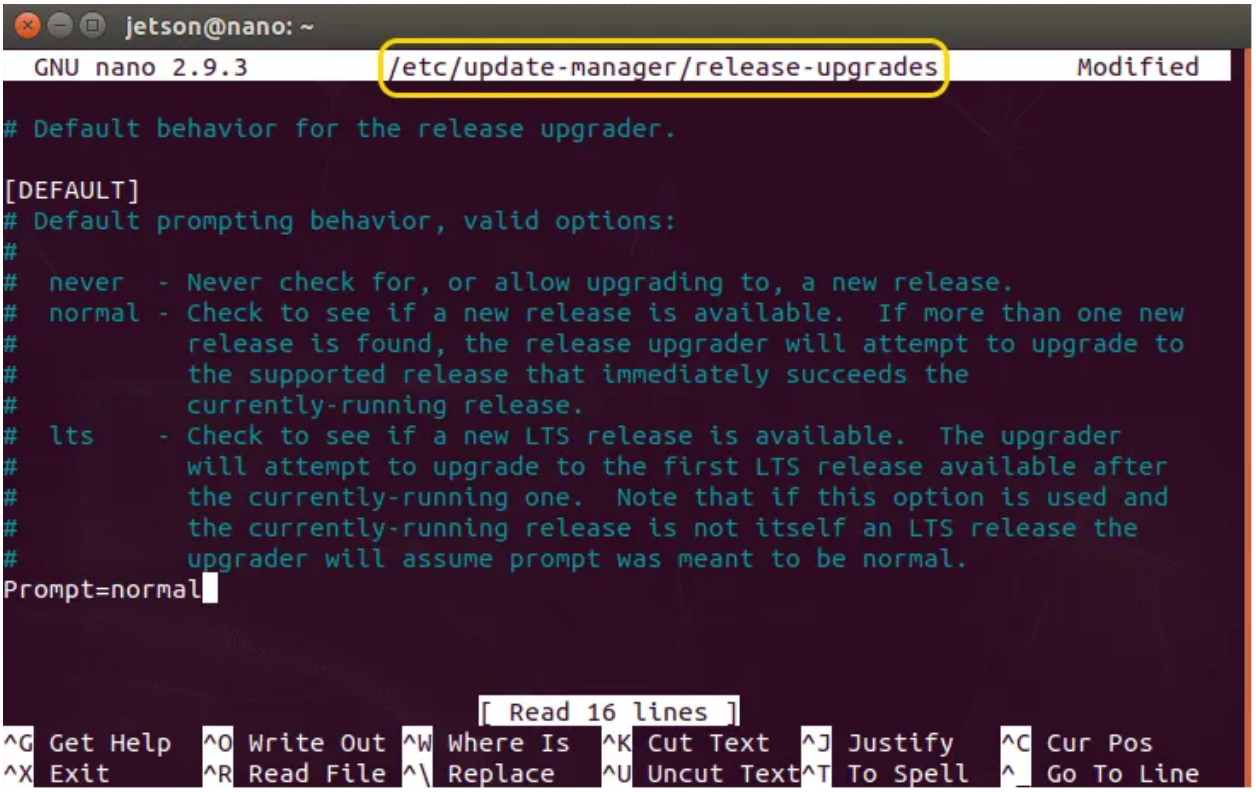

Figure 2.1: Enable release upgrades.

```
# refresh your system again
sudo apt-get update
sudo apt-get dist-upgrade
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

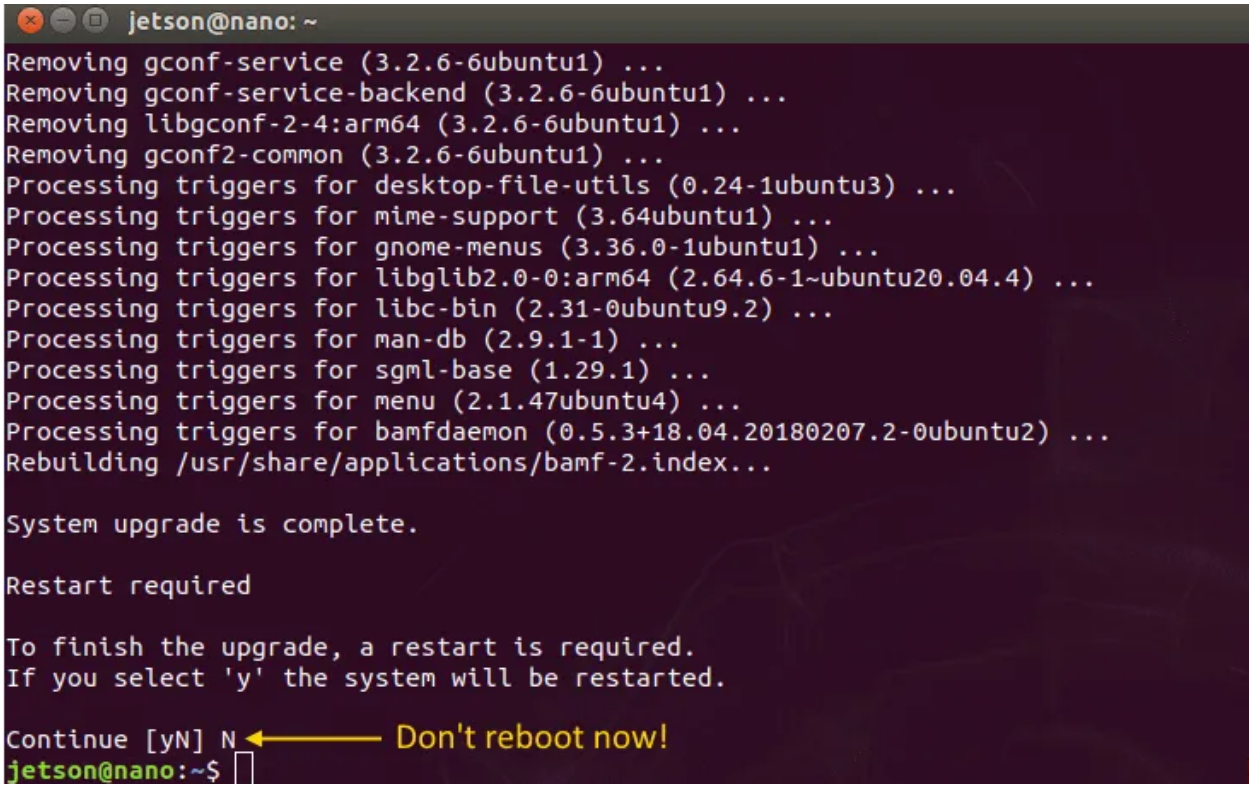

Figure 2.2: Final screen. Do not reboot!

/etc/gdm3/custom.conf file. Next, uncomment *Driver "nividia"* 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
sudo rm -rf /usr/share/vulkan/icd.d
```

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

```
# prepare your system
sudo apt-get update
sudo apt-get upgrade
sudo apt-get autoremove
# remove circular symlink
sudo rm /usr/share/applications/vpi1_demos
# remove distorted nvidia logo in top bar
cd /usr/share/nvpmodel_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.

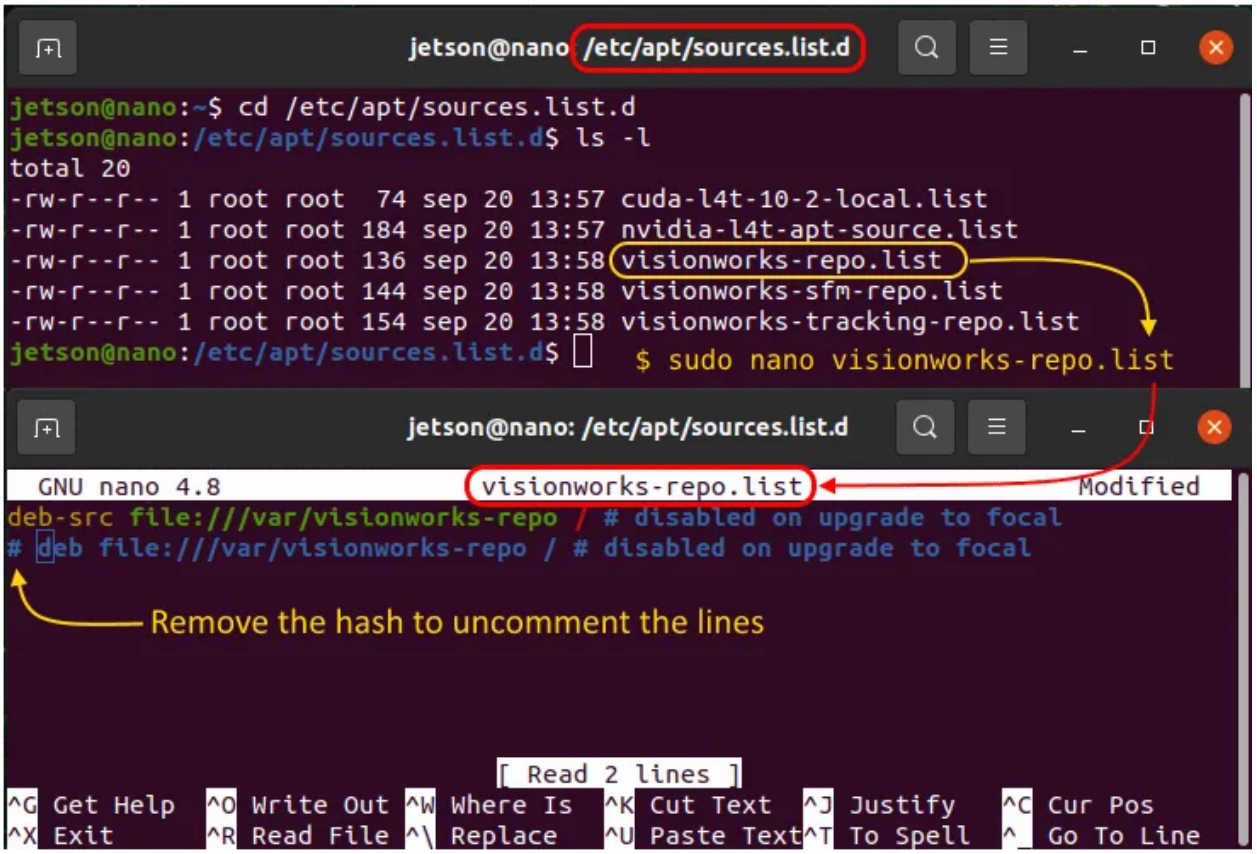

Figure 2.3: Reactivate NVIDIA repositories.

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):

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

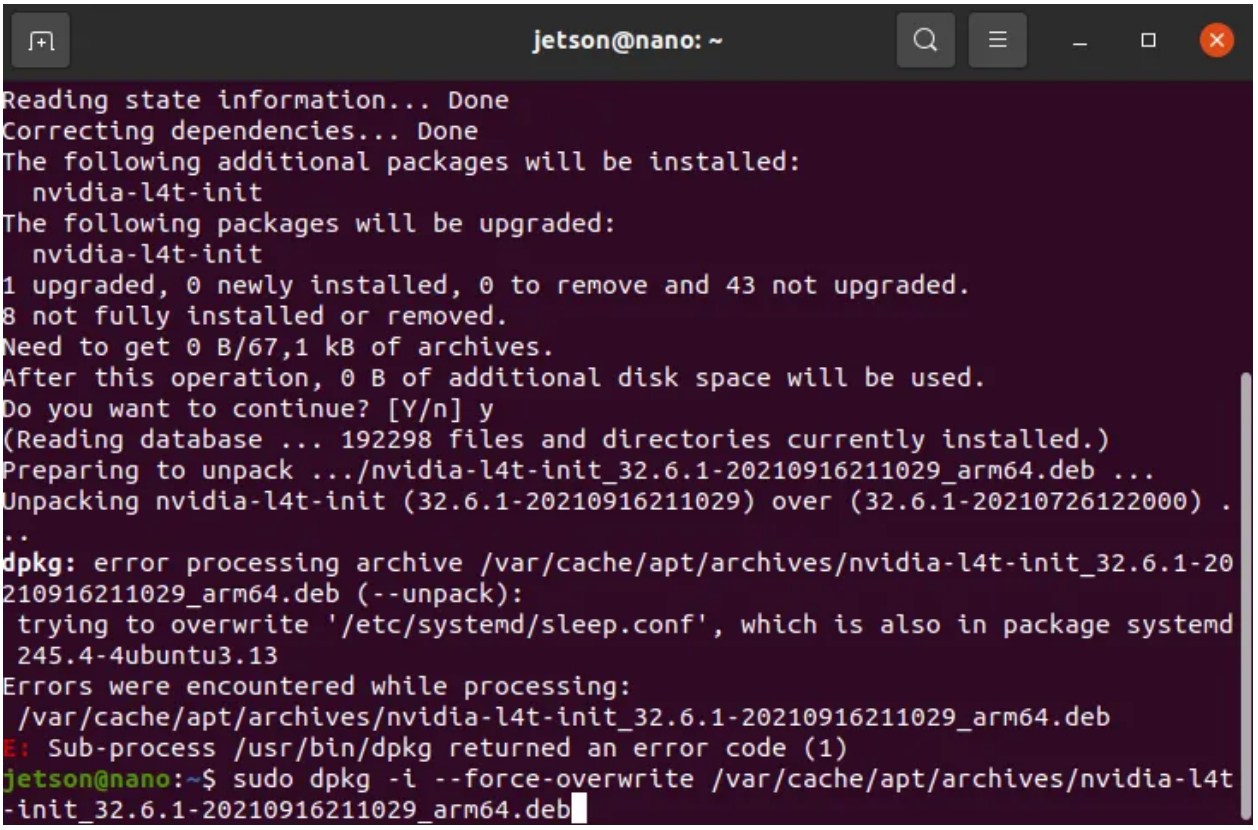

Figure 2.4: Broken apt upgrade screen.

Chapter 3

Camera Calibration

Camera calibration is a necessity for removing distorsion 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.