The Incomplete Guide to Using the TX1 for FRC

The Zebracorns

FRC Team 900

Email: support@team900.org

Before You Start

This guide is not complete. It is merely the beginning on your journey with the TX1. It will not answer all of your questions nor is it meant to (It's "incomplete" for a reason). Before you begin you should start by looking at the available resources you have and read the documentation available for the TX1. It's entirely possible that this guide will be out of date by the time you read it. It is not definitive and should not be treated as such.

We're happy to try and help you as best as we can. The Zebracorns, FRC Team 900, has been using embedded systems from Nvidia for a few seasons in FRC at the time of writing. We've had a lot of luck with running them and have won several control system and innovation awards for our work with these boards and using them for vision processing. If you send us inquiries then we will do our best to help you. We don't have all the answers and we can't promise that we'll respond quickly (we're building robots too) but we will do our best to get you the information you are seeking. Please send any inquiries to support@team900.org.

Unboxing the TX1

In the box you'll find a TX1 module and carrier board:

The TX1 is actually an SOM (System On Module) about the size of a credit card (though a bit thicker). The heatsink on top dissipates heat and the whole thing is attached to a carrier board with four screws and a special connector.

The carrier board for the TX1 (TEE EX ONE) is a Micro-ITX style board and will fit inside of a Micro ITX case. This board is larger than the TK1 (TEE KAY ONE). It will require some thought for mounting on your robot so take some time to investigate your options. It should be somewhat protected but is reasonably robust. There are some COTS cases available if you look around. Check the links in the Resources section for this document.

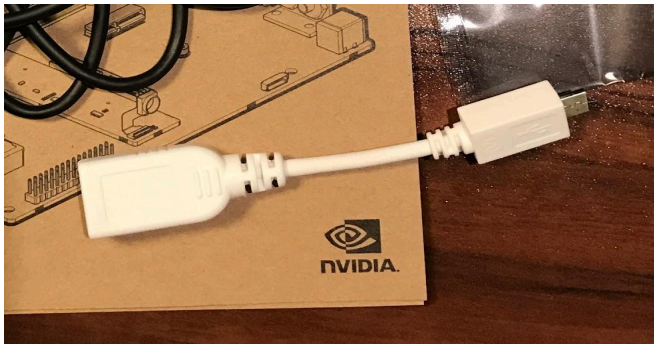
A power brick and cable:

Please note that the power brick outputs 19V. This power brick produces a voltage that is incompatible with the TK1 boards. If you plug this into your TK1 board then bad things will happen. The TX1 can run off of 12V but please make sure that your polarity is correct before plugging it into another power source.

A USB Micro-B to Male USB A cable:

You might have some of these from your phone or other electronic widgets already. This one isn't special other than it has a nifty Nvidia SHIELD logo on it.

A USB Micro-B to Female USB A Cable:



This is useful for plugging in additional USB devices to the TX1 since the carrier board only has one USB 3.0 port. Please note that it is not USB 3.0. It is only USB 2.0.

You'll get some antennas for wireless but you won't need these for FRC because you can't have wireless communications outside of the radio. You can potentially install them anyway and use them as tiny flagpoles to display your team standard though.

You might also get a jumper and/or some rubber feet. If you don't have them then don't stress out. They aren't strictly necessary. Some of the dev kits have them and some don't. It's not a big deal if they don't. They might already be installed or not useful for you.

There are also some instruction books but I'm guessing you aren't going to read them since you are reading this guide instead. I think you should read them but I'm not your mom and can't make you. Now, go clean your room! Just kidding.

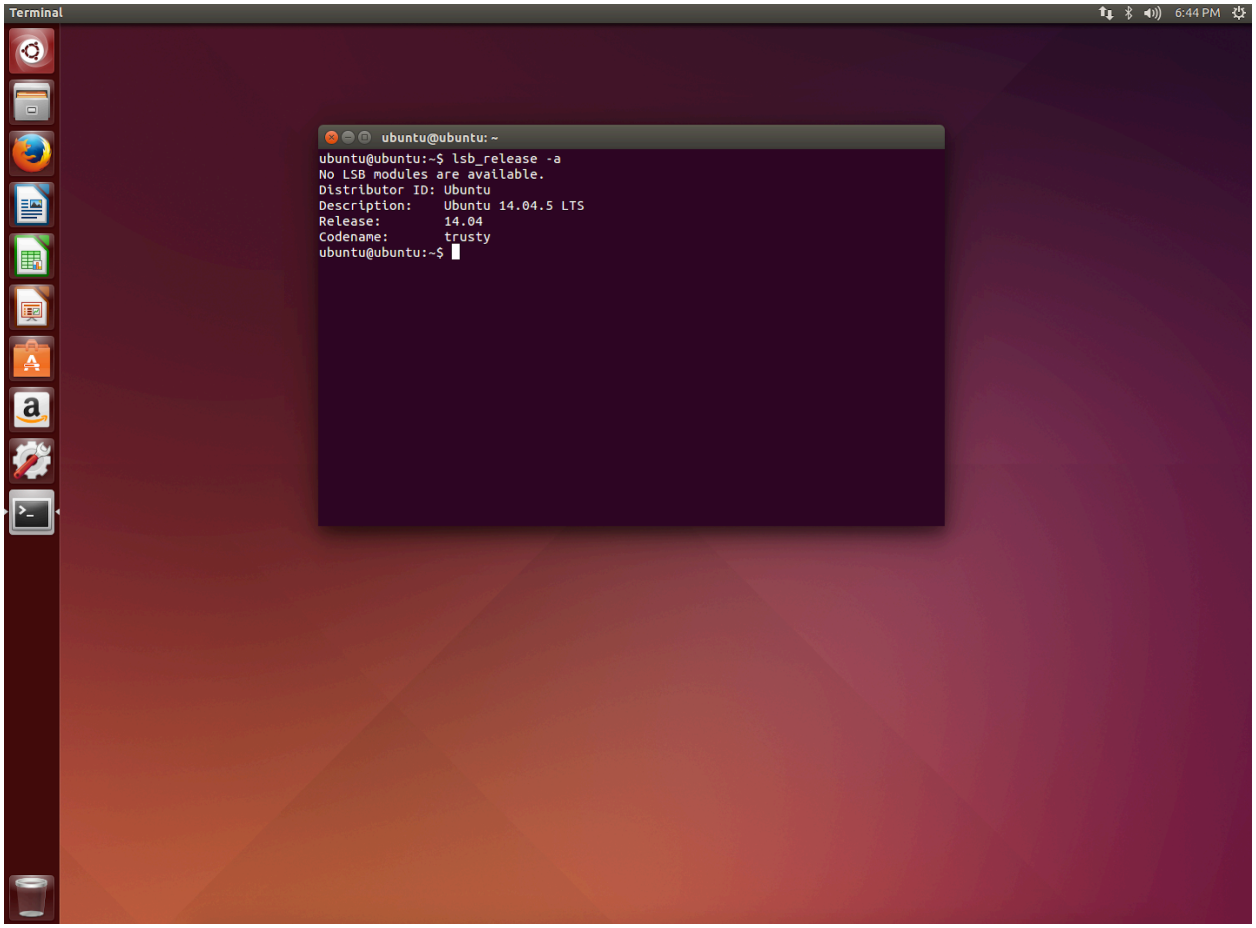
Setting up the Host OS

Download and install Ubuntu 14.04 x64 Desktop on a computer of your choice. Not RedHat, not Slackware, not Chrome OS, not Windows, not Mac, definitely not BeOS, and most certainly not AIX. Stick with Ubuntu 14.04 x64 Desktop. Don't try and be cute and use 15.04 or 16.04 or some other number. If you've started counting and gone past 14.04 then go back.

Look, here's a link: http://releases.ubuntu.com/14.04/

14.04 is an LTS version (Long Term Support) of Ubuntu and is still supported through April of 2019.

Here's what it looks like when you get it installed:



You can run “lsb_release -a” in a terminal to find the verify the version of Ubuntu installed is correct.
You **MIGHT** be able to install Ubuntu 14.04 on a virtual machine but technically it’s not supported. Your best bet is to install it on a dedicated system.

You’ll also need a monitor, a keyboard, and a mouse to complete the install as well. Someone out there is going to try and be cute and do this on a headless terminal and you probably can if you’re an uber-linux-knurd and know how to export an X session and type out ASCII at 9000 baud with your toes... but let’s face it, you’re probably not that girl so instead of being clever, just do it the easy way and quit being difficult.

You’ll want to make sure your host is plugged into Ethernet (not wireless) and hopefully it only has one ethernet card. If it has more than one then try to plug it into the lowest enumerated card. If you don’t know what that means then just pick a port and hope you get it right.

We recommend you have at least 50GB of free space on the host system. Do you actually need 50GB of free space? Probably not but it’s better to have it than not. Can you use more than 50GB? Sure. The space needs to be allocated to your user space so the JetPack installer can access it on the Desktop, where it will download additional files.

Once the host OS is installed, you might want to run apt-get update and apt-get upgrade to get the latest packages but this isn't required. Don't update to a newer version of Ubuntu though. Don't run "do-release-upgrade". That's bad. Seriously, don't do it. Do not pass go. Do not collect $200.

Installing JetPack

You will want to download the JetPack installer from Nvidia. You don't have to do this but you probably want to. It's going to make your life easier. What does JetPack install for you? OpenCV, the latest build of Ubuntu 16.04 for the Nvidia TX1 (Linux 4 Tegra), all of the Nvidia apt source libraries, and some other cool CUDA development stuff. It installs pretty much all the essentials and updates your system correctly.

Download JetPack from the Nvidia site here: https://developer.nvidia.com/embedded/jetpack

Nvidia provides a guide for downloading and installing JetPack located here: http://docs.nvidia.com/jetpack-l4t/index.html#developertools/mobile/jetpack/l4t/2.3/jetpack_l4t_install.htm

You will need to register as an Nvidia Developer to complete the download. Registration is free.

The file you download is a .run file. We recommend saving it to the desktop on your Ubuntu 14.04 host system. It will be easy to find that way.

Once downloaded, you'll need to use command line and run the utility "chmod" to add executable permissions to the file.

- Start by opening the terminal utility
- Run "cd ~/Desktop" to change directories to the desktop and locate the file.
- Run "chmod +x JetPack-${VERSION}.run"
    - ${VERSION} is the version of JetPack that you downloaded and not literally ${VERSION}

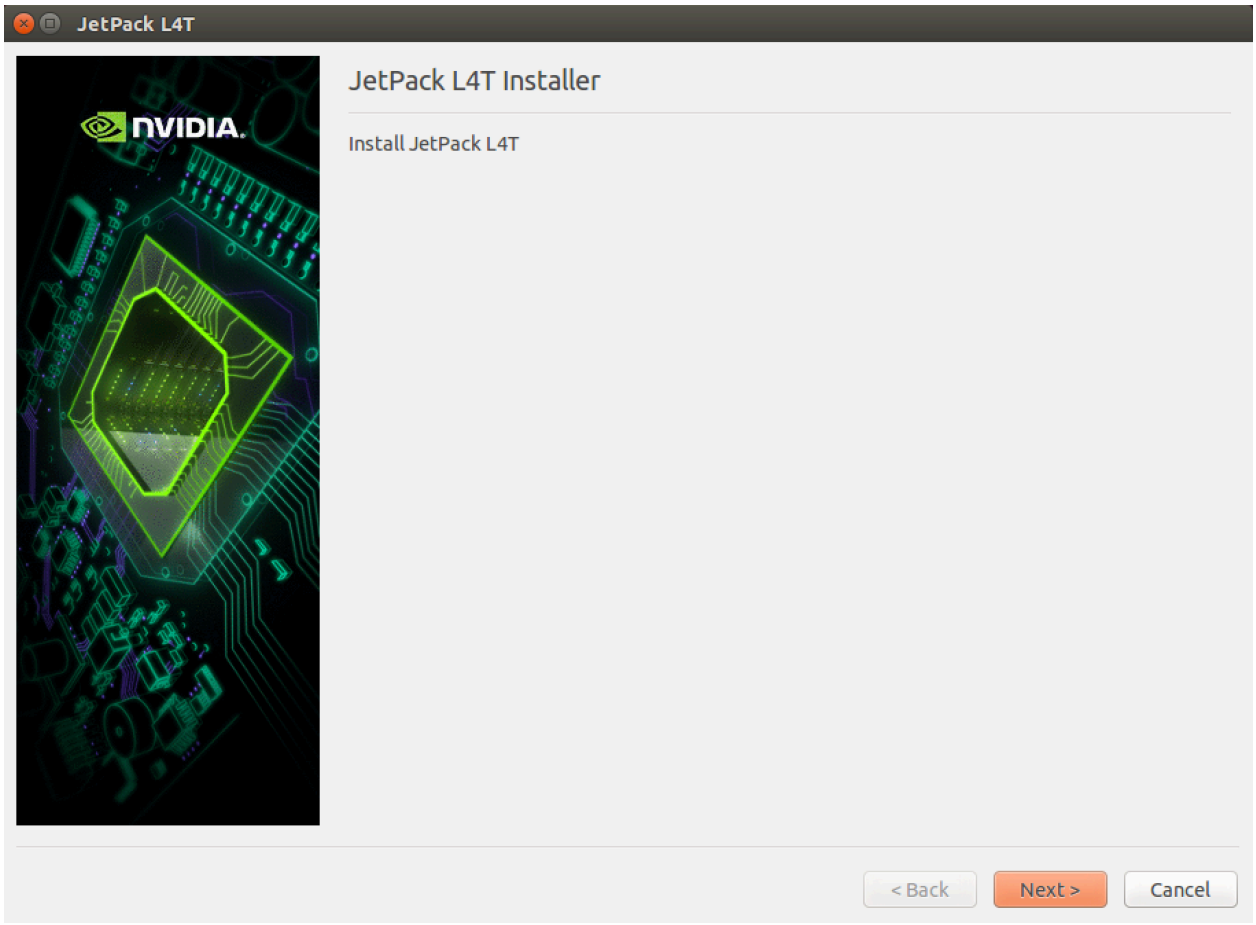
Running the JetPack Installer

Now that you've downloaded JetPack and added executable permissions to the file you'll need to run the installer. You can do this by typing the following in the terminal:

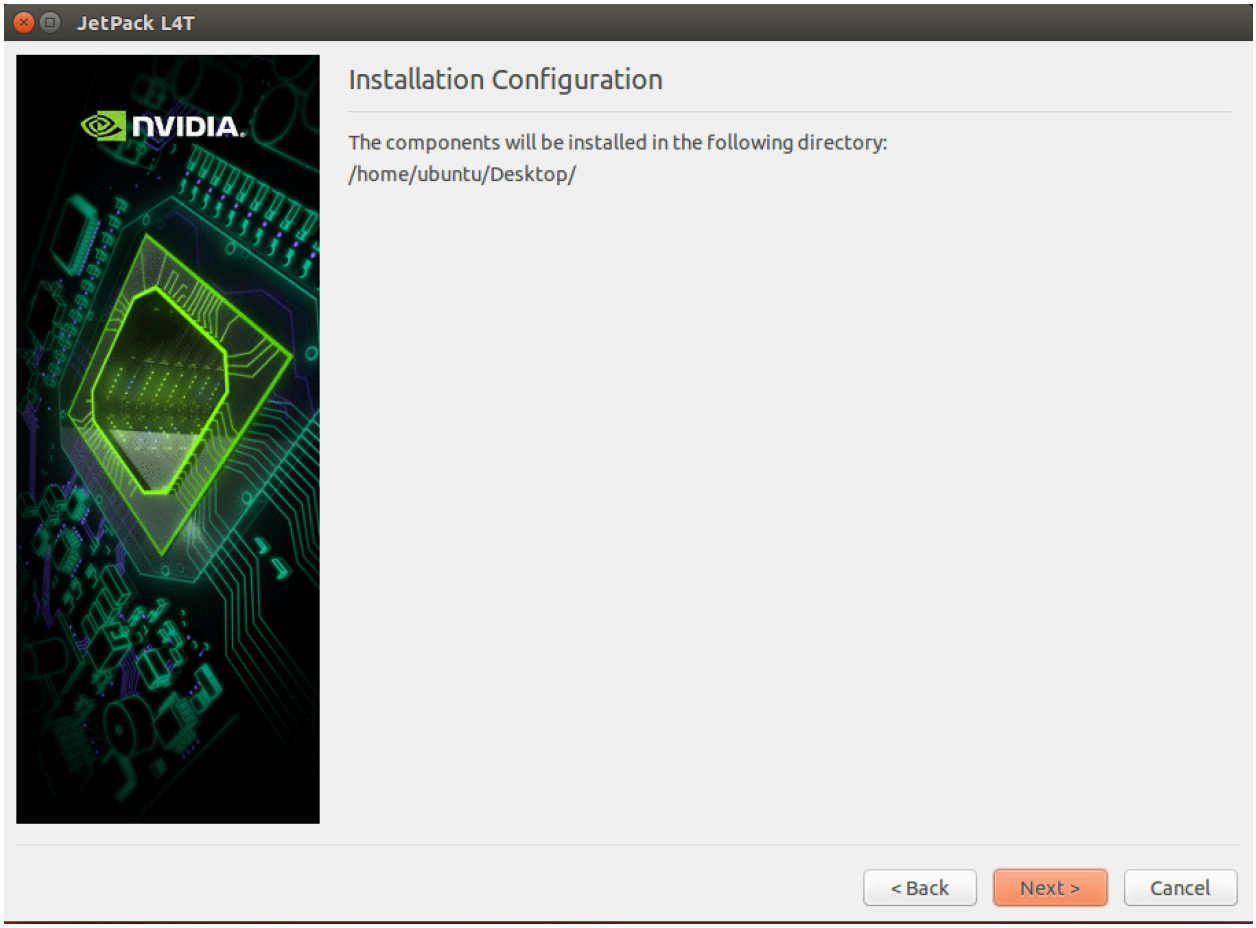
- First run "sudo -s" to enable super user privileges for your account. This isn't strictly necessary but it will probably make the process work easier. If you miss this then don't fret, the installer will prompt you for your password later.

- Run “./JetPack-${VERSION}.run” in the terminal utility
    - ${VERSION} is the version of JetPack that you downloaded and not literally ${VERSION}
    - If you append an “ &” (space plus ampersand symbol) to the above command then the command will execute in the background and you can close out the terminal. If you don’t do this then leave the terminal open as the command executes in the foreground.

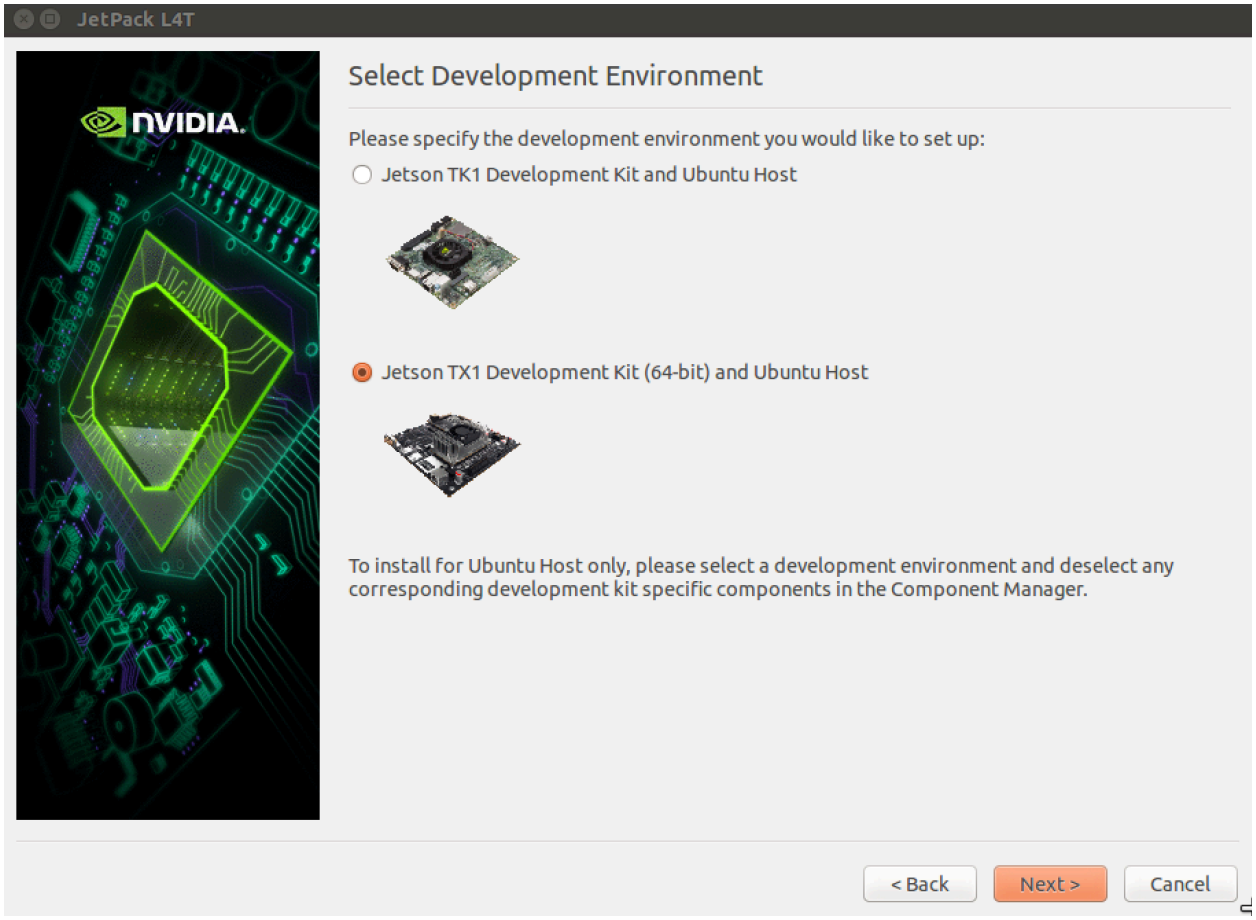
The first screen of the JetPack installer will launch:



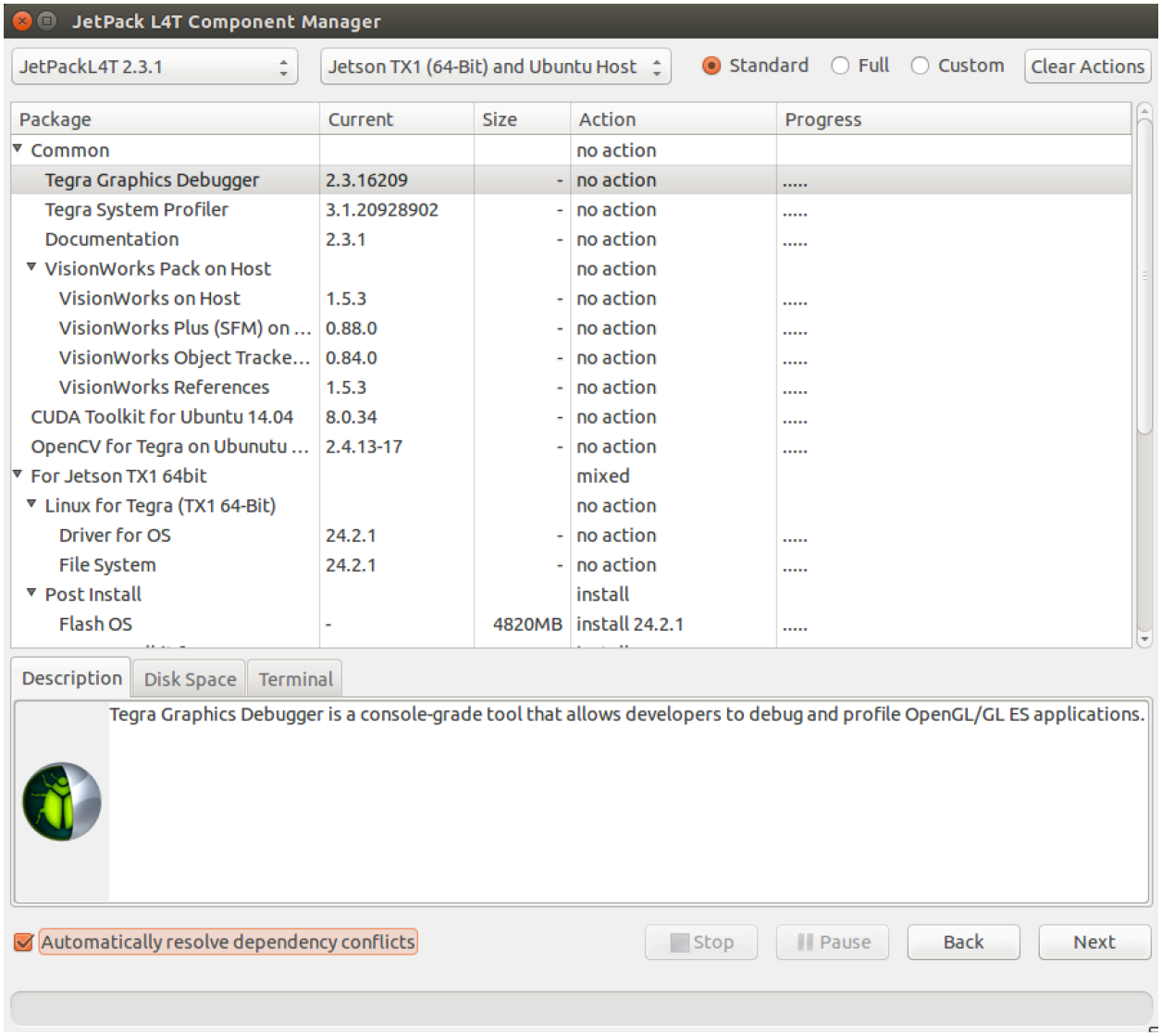
Click “Next >” and you’ll be taken to the next screen where you can confirm the location that the JetPack installer will use to download files:



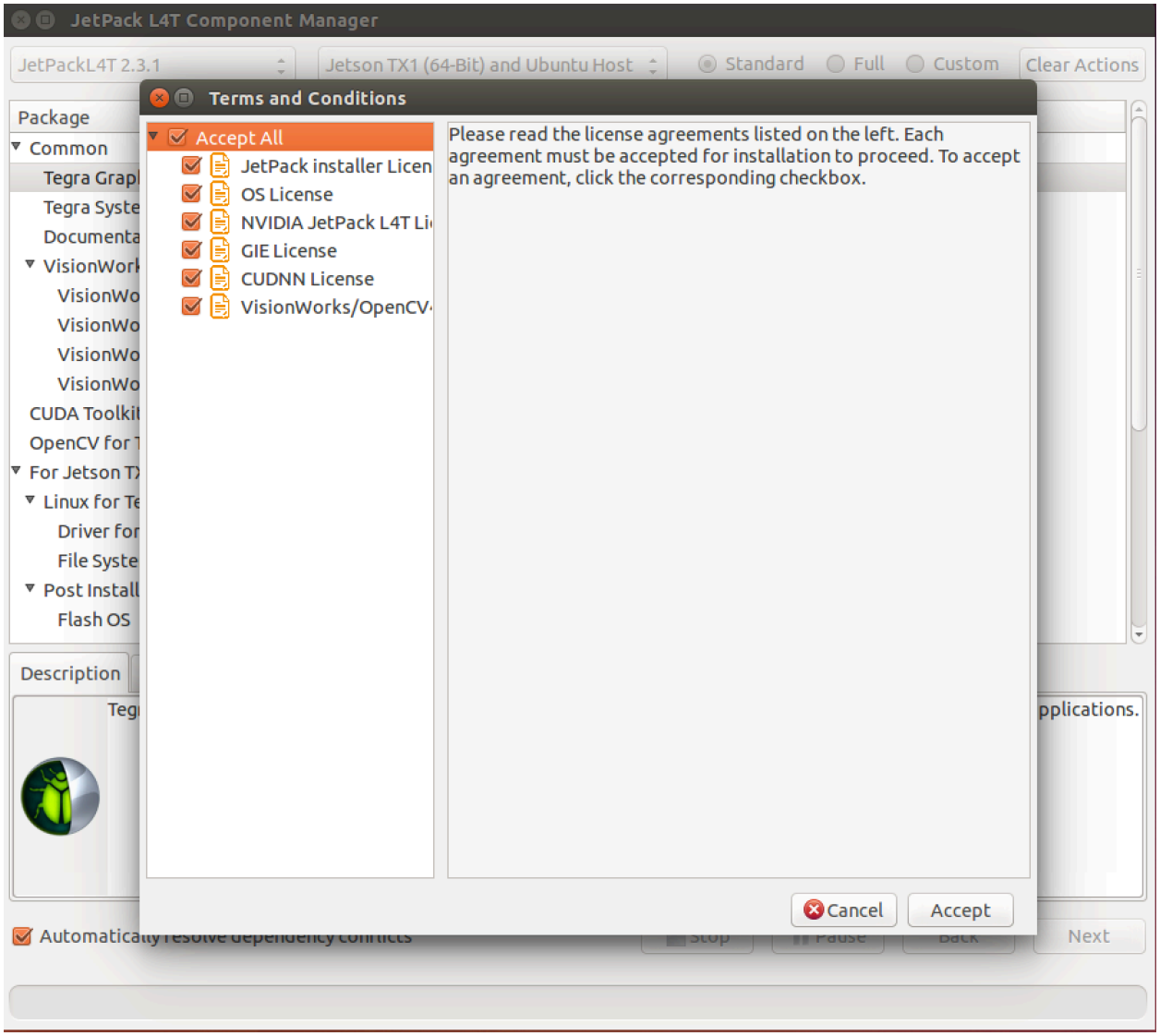
Click “Next >” to be taken to the next screen where you can select the development board you are installing on. You will want to select the “Jetson TX1 Development Kit (64-bit) and Ubuntu Host” option (NOTE: The default selection is for the TK1 board and NOT the TX1 board):



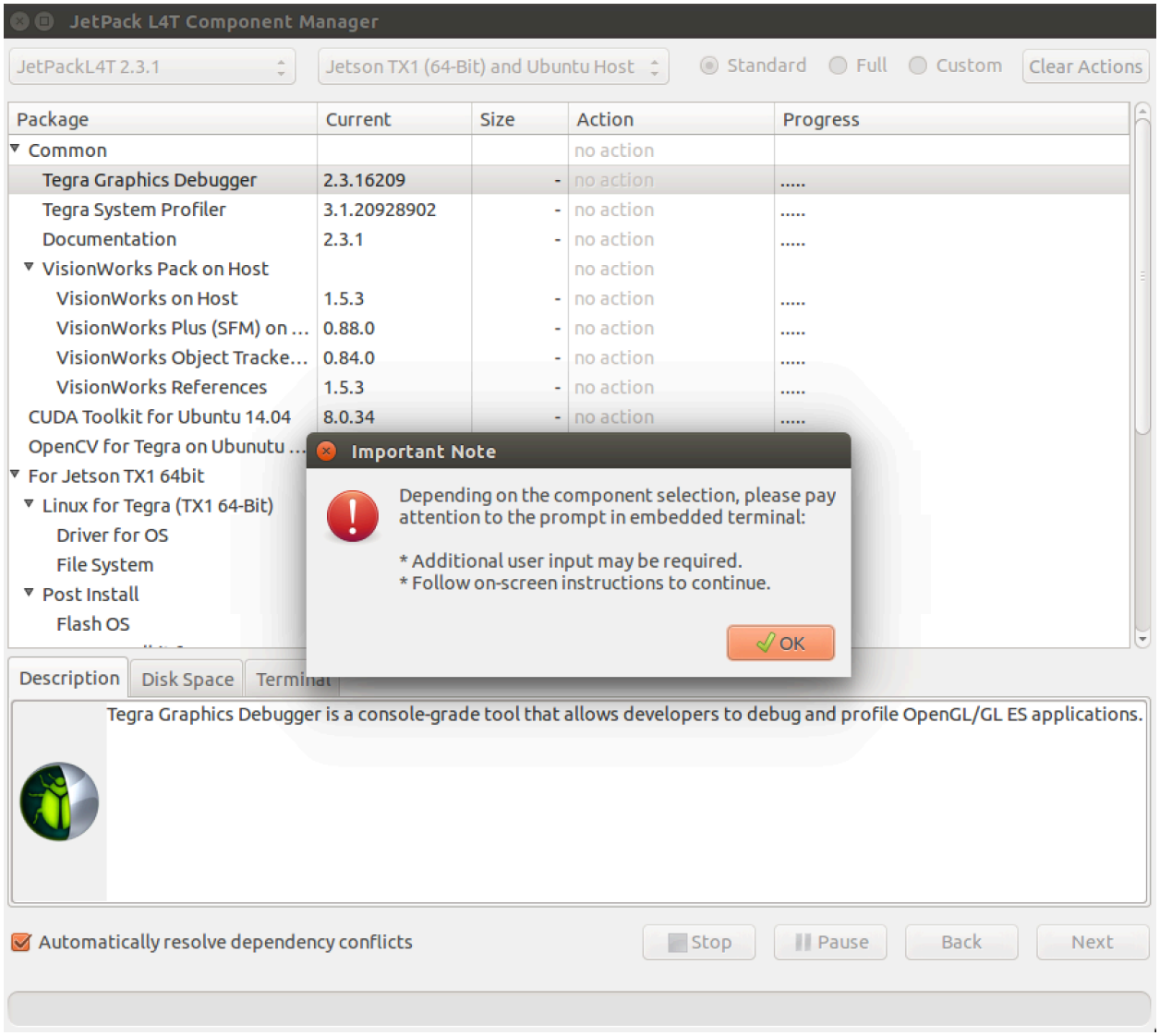
Click “Next >” to be taken to the next screen where you will have some options about selecting specific versions of the JetPack and Linux for Tegra (L4T) software packages (NOTE: your screen might look a little different, that’s ok). **You may be prompted for your root/sudo password at this point. If you are then enter it (If you get an error then restart the installer as root).** Leave the options as default but do select the option checkbox to ““automatically resolve dependency conflicts” before proceeding:



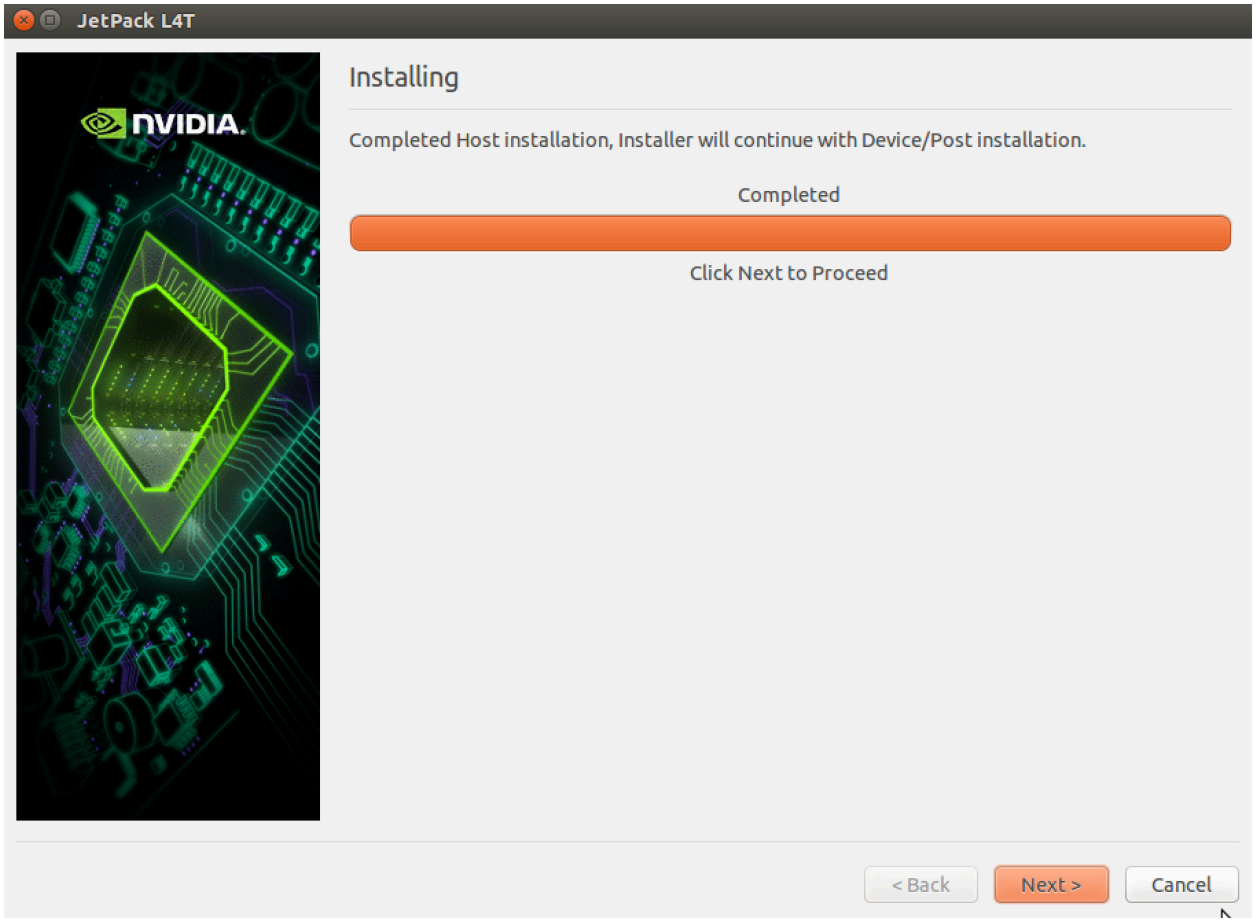
Click “Next >” to be taken to the next screen where you will need to accept the license agreements that accompany the software. There are a few of them and you can select “Accept All” to just get them all done in one go without reading a word:



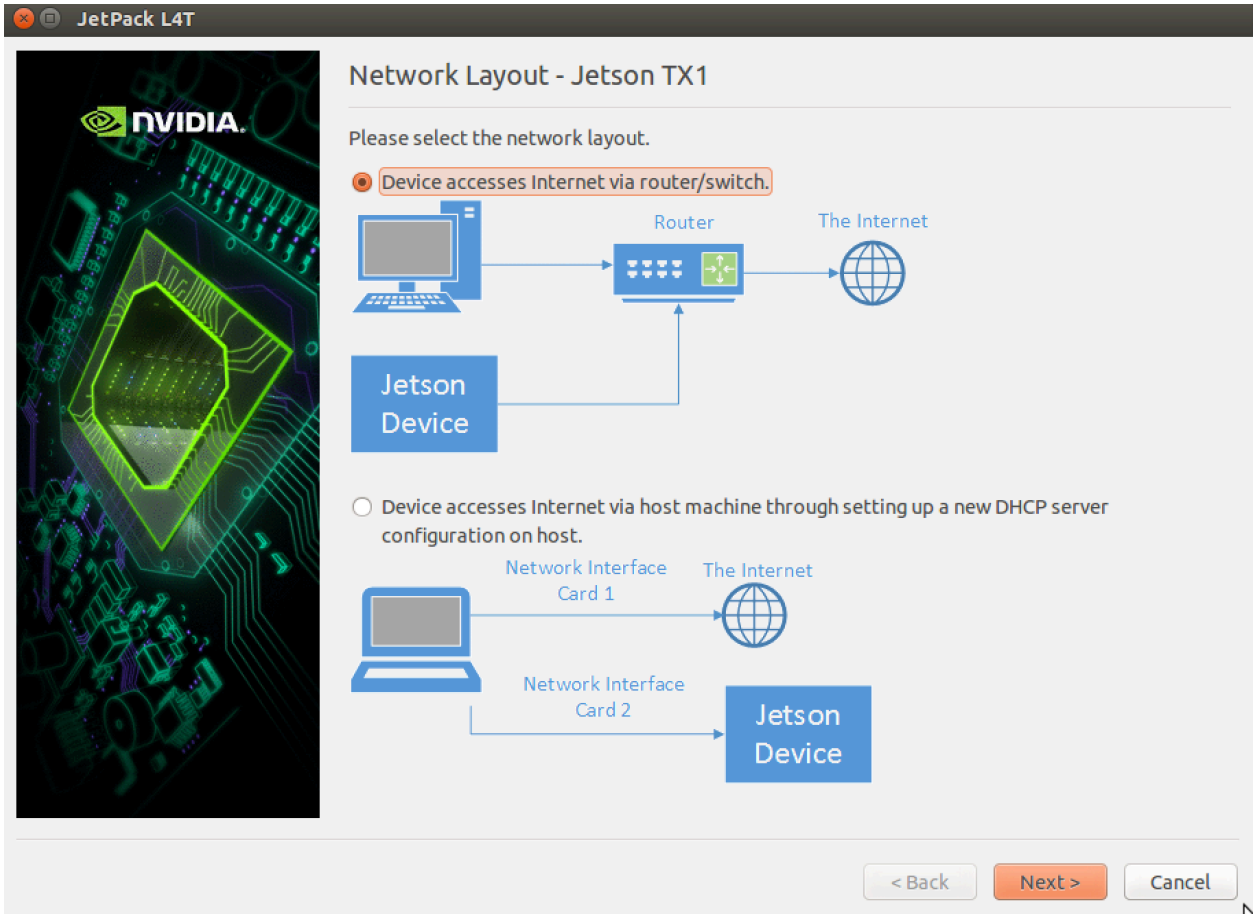
Click “Accept” to proceed to the next screen which is an Error Message. Ahhhhhhhh!!!!! Just kidding. It’s not a real error. It’s an important message that is trying to tell you that based on the previous pages you may have additional steps you might need to complete and that you should PAY ATTENTION to the prompts in the terminal that pops up. It also says you should FOLLOW the instructions it provides (what happens if the instructions differ from these instructions? I don’t know, you’ll have to decide for yourself. Choose wisely.):



Click “OK” to proceed to proceed to the downloading phase. This will take a while to download all of the files depending on your internet connection speed. Be patient as some of them are large. It might prompt you to install some things. If it does then install them. When it is complete, the installer will show this screen:



Click “Next >” to proceed to select the Internet connection method for the TX1:

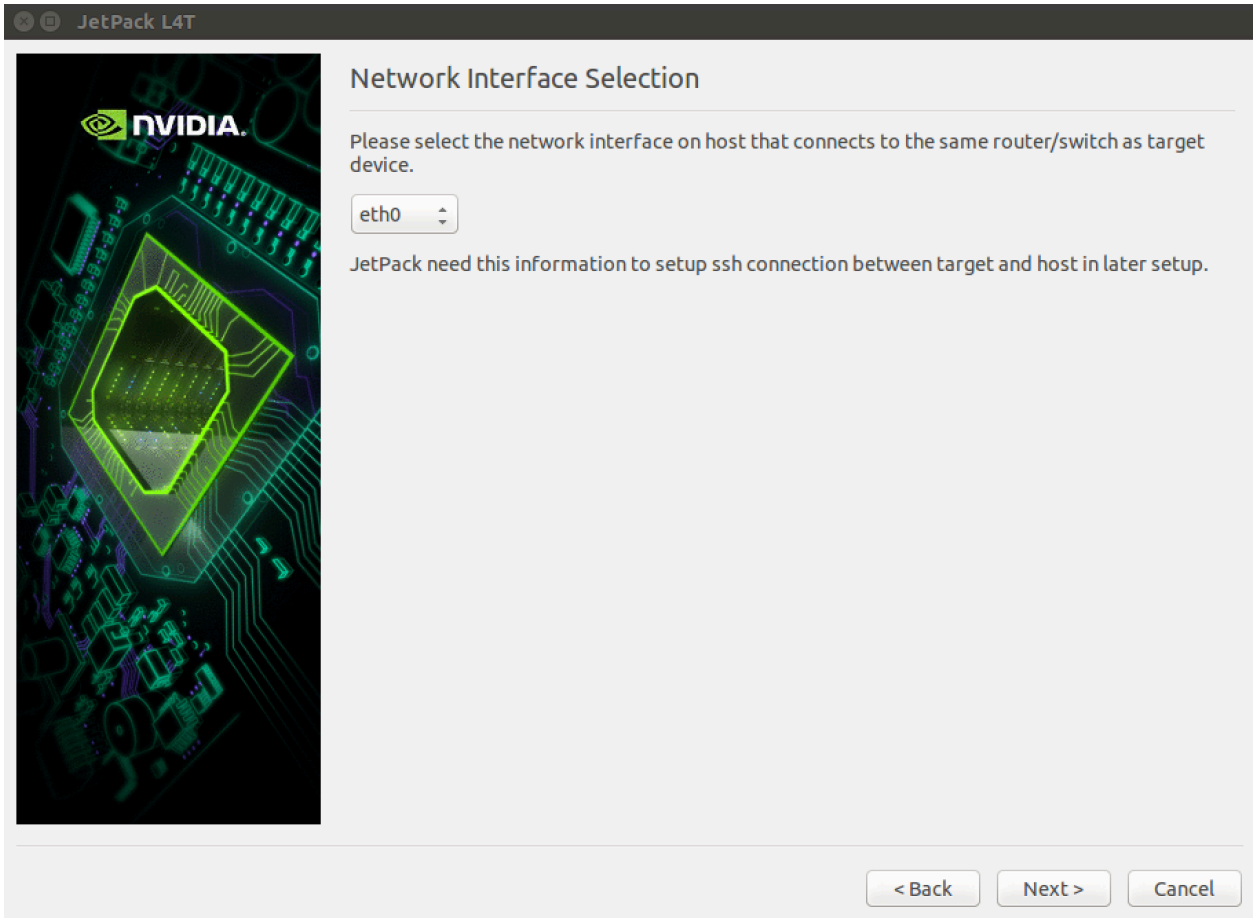


If you're an advanced user, feel free to do what works best for you. If you're following this guide then selection the first option "Device accesses Internet via router/switch" and make sure your TX1 is plugged into your local router/switch with an Ethernet cable along with the host system.

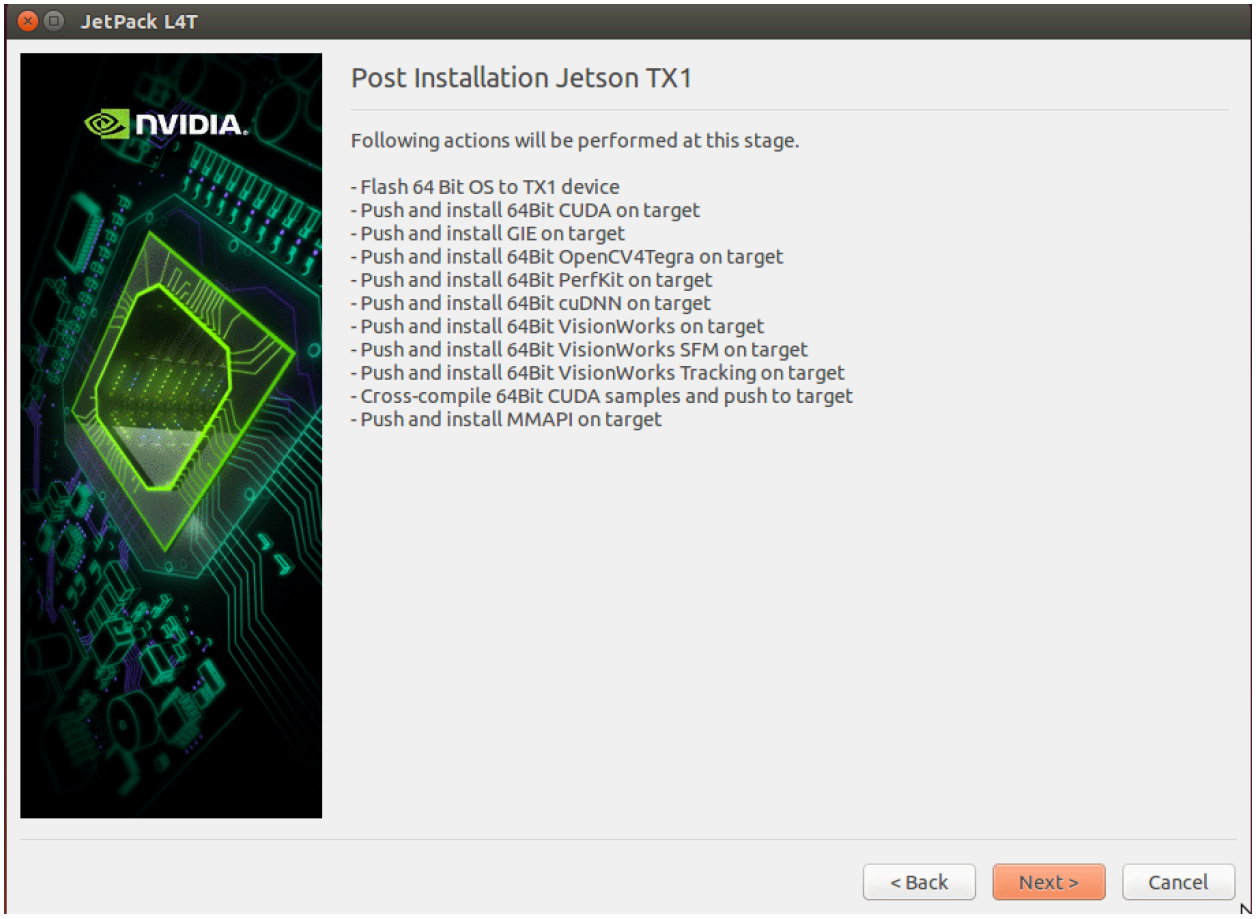
What's that? You don't have an Ethernet cable. Well, this guide can't help you with that, go find one. Plug one end into your local router/switch and the other into your TX1. You'll need DHCP for this to work and your computer and the TX1 should be on the same subnet.

What? You thought it was wireless. Well, it is but not until you set it up the first time. Also, you can't use the wireless on it on an FRC robot so you're going to have to plug it in anyway.

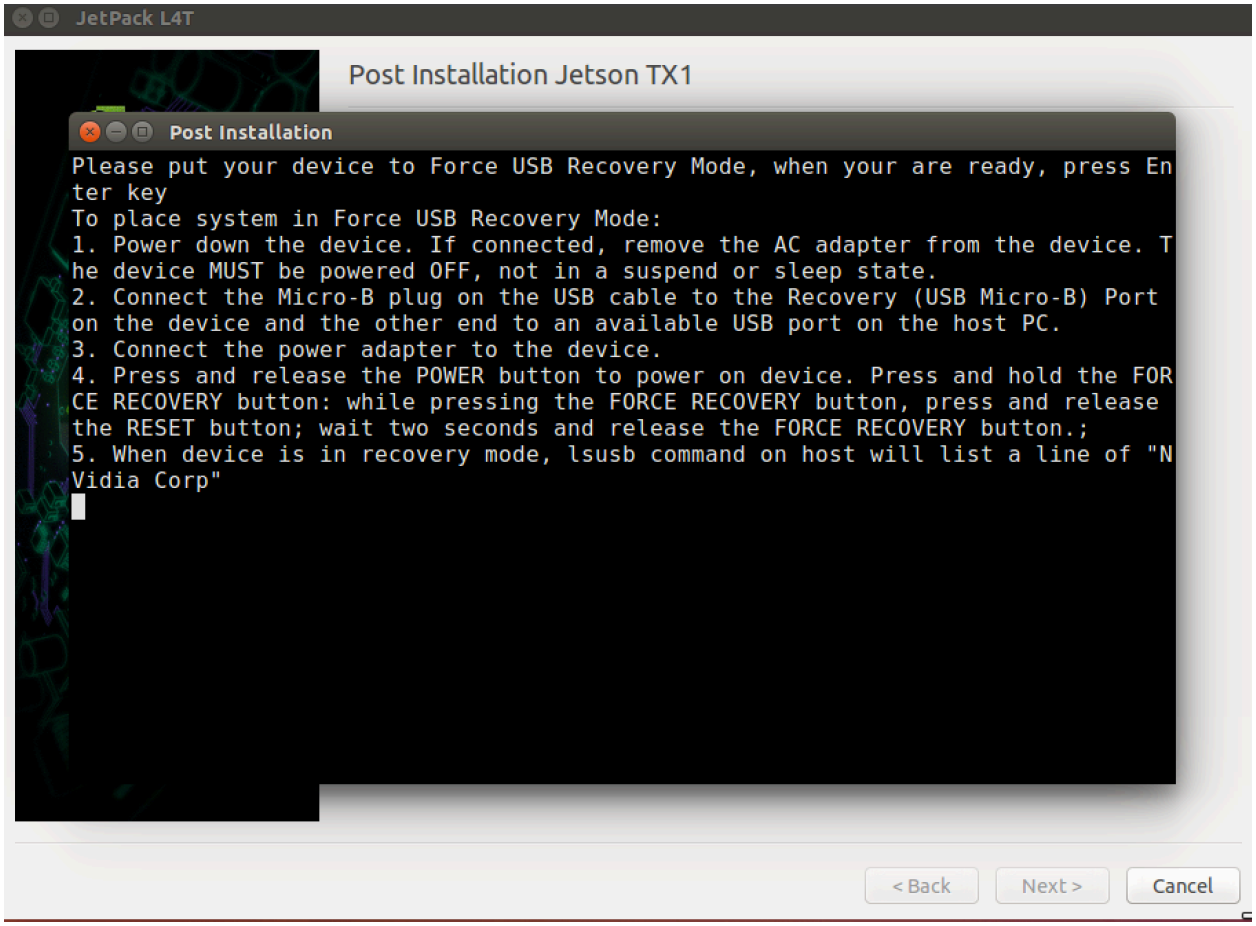
Click "Next >" to proceed to the screen where you will tell the install which Ethernet port on the host you'll be using. Hopefully your host only has one ethernet port and you can just select "Eth0" and move on. If it doesn't then pick one and hope you get it right:



Click “Next >” to proceed to the page where the installer will tell you everything it is about to do. You can read over this page or just charge ahead like a bull on parade:



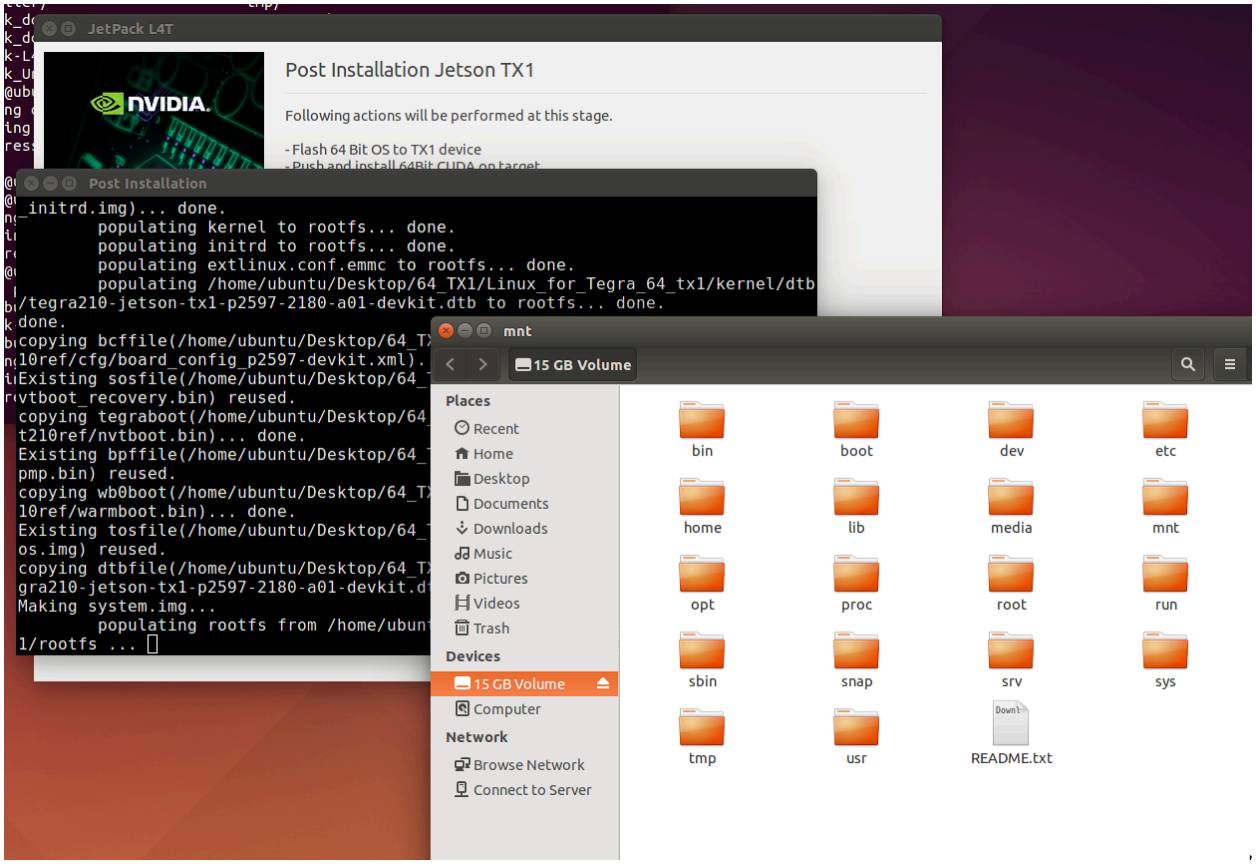
Click “Next >” to proceed to the installation and imaging phase. A terminal window will pop up eventually and when it does it is going to contain some directions for how to proceed with placing the TX1 into “recovery mode” so it can be flashed via USB.



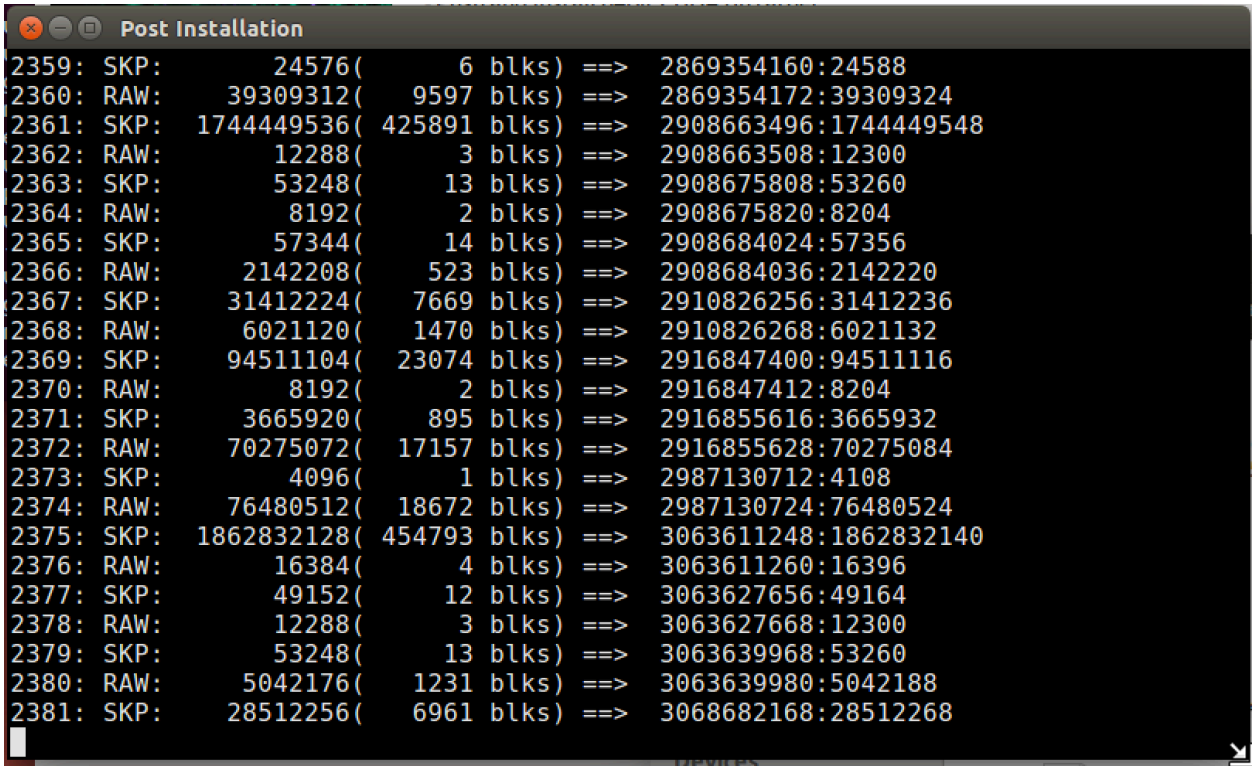
Following the directions:

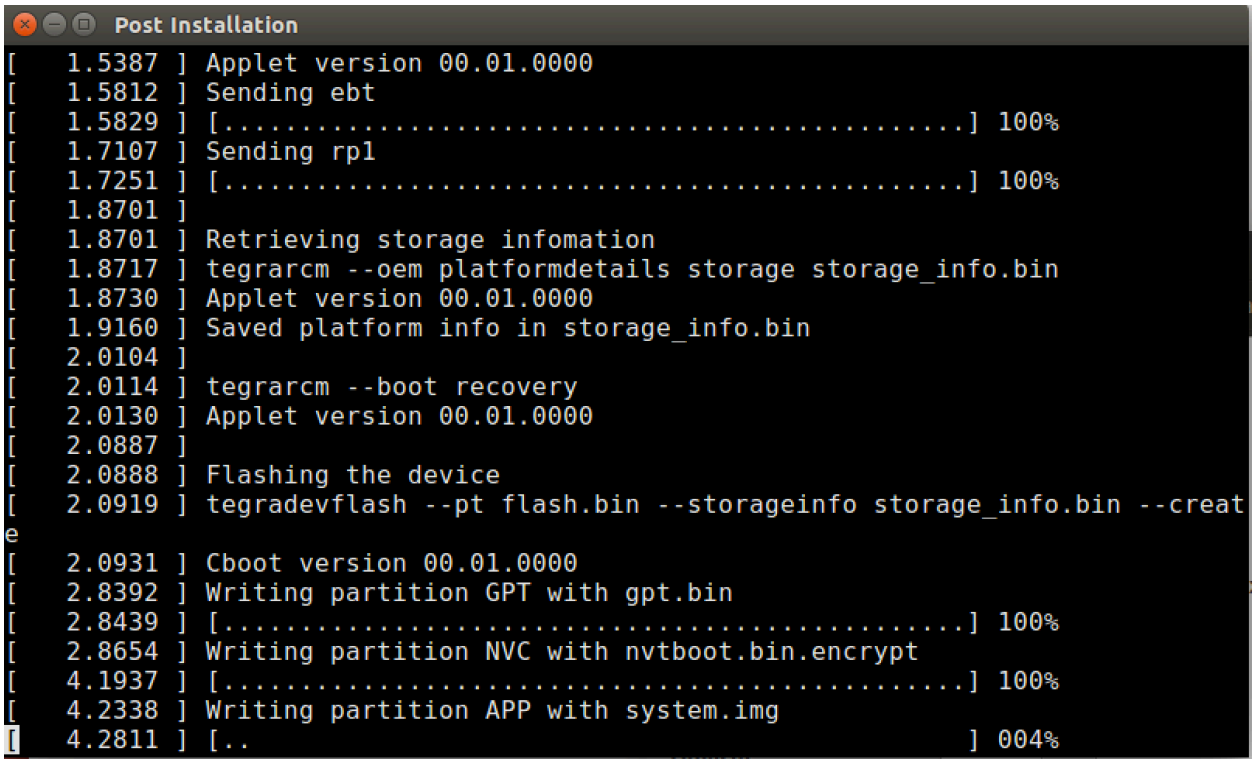
1. Remove the power from the TX1
2. Connect the USB Micro-B to Male USB A cable to the device by plugging in the Micro-B side to the Recovery Port and plugging the USB A side into your host system.
3. Connect the power cable back to the TX1
4. Press the power button on the TX1
5. Now press and hold the “Force Recovery” button while simultaneously pressing and releasing the reset button. Hold the “force recovery” button down for a few seconds.
6. If you do it right then you can run the “lsusb” command on the host and it will display “NVidia Corp” as a USB device. If this doesn’t show up then try again until it does or your fingers get tired.

When you are ready and the device is available, press the return/enter key and move forward. When you do, the terminal will change and it’s possible a file browser window will pop up. You can minimize the file browser window.



The terminal window will continue to scroll text. If it does then you'll end up with some more screens that look like these:

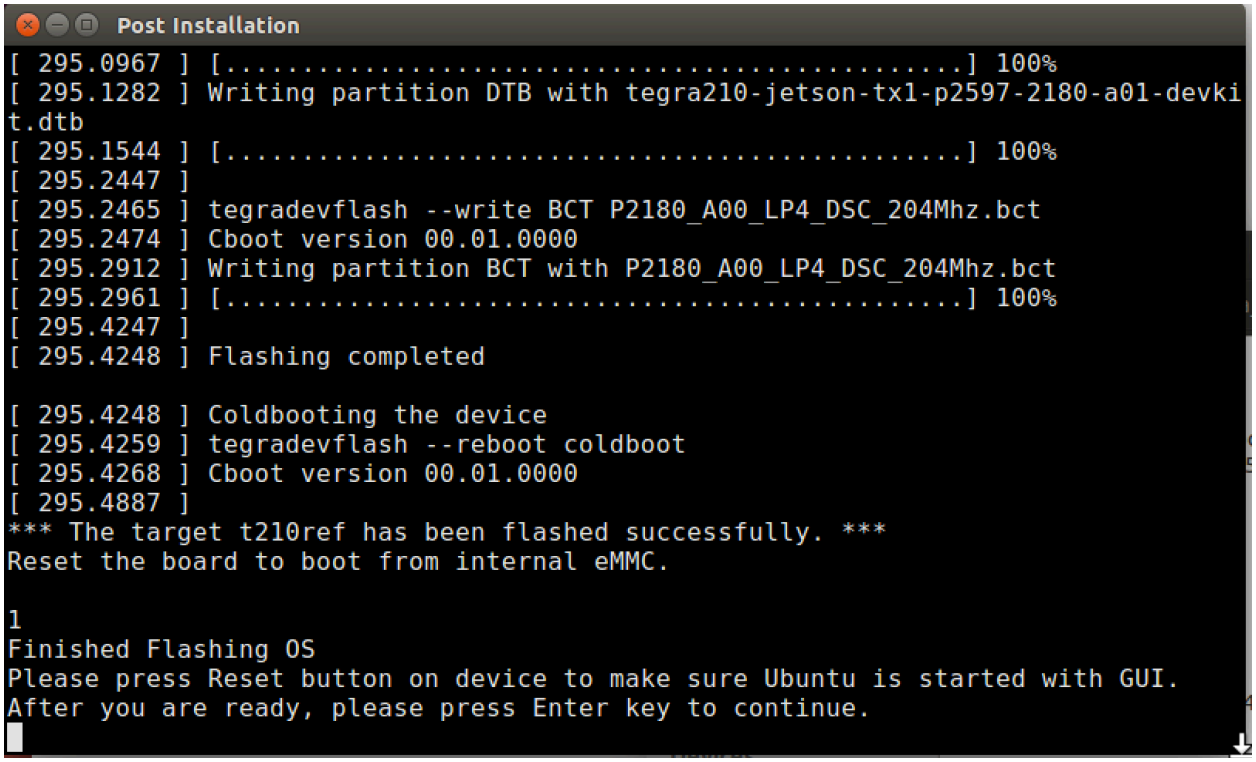




At this point the install will take a while to completely flash the device so be patient while it does what it needs to do.

NOTE: If you're brave enough (read: foolish enough) to be using a virtual machine for this then there is a good chance the update of the OS will error out at this point. If it does then don't stress. Go get a cup of tea, reboot the VM and give it all another try. If it fails a second time then stop trying to use a virtual machine and go get a physical host computer.

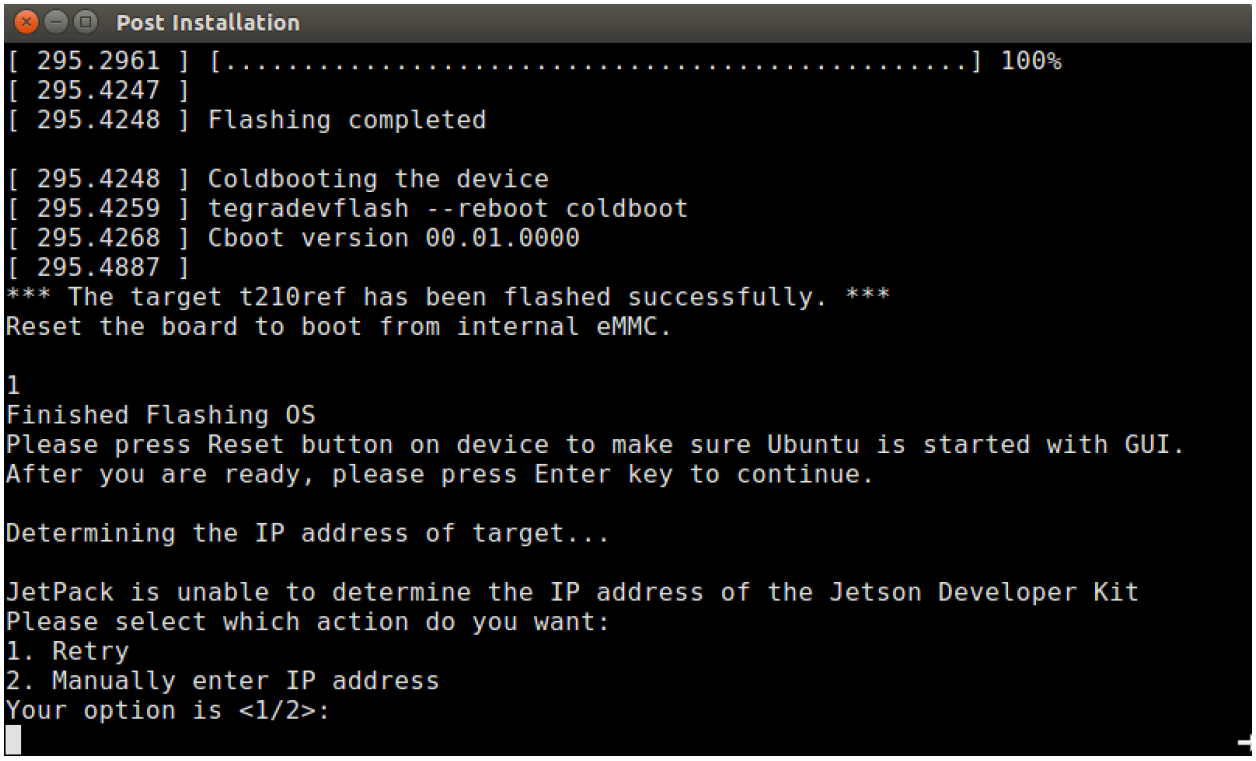
Eventually the process will finish and you'll end up with a screen like this:



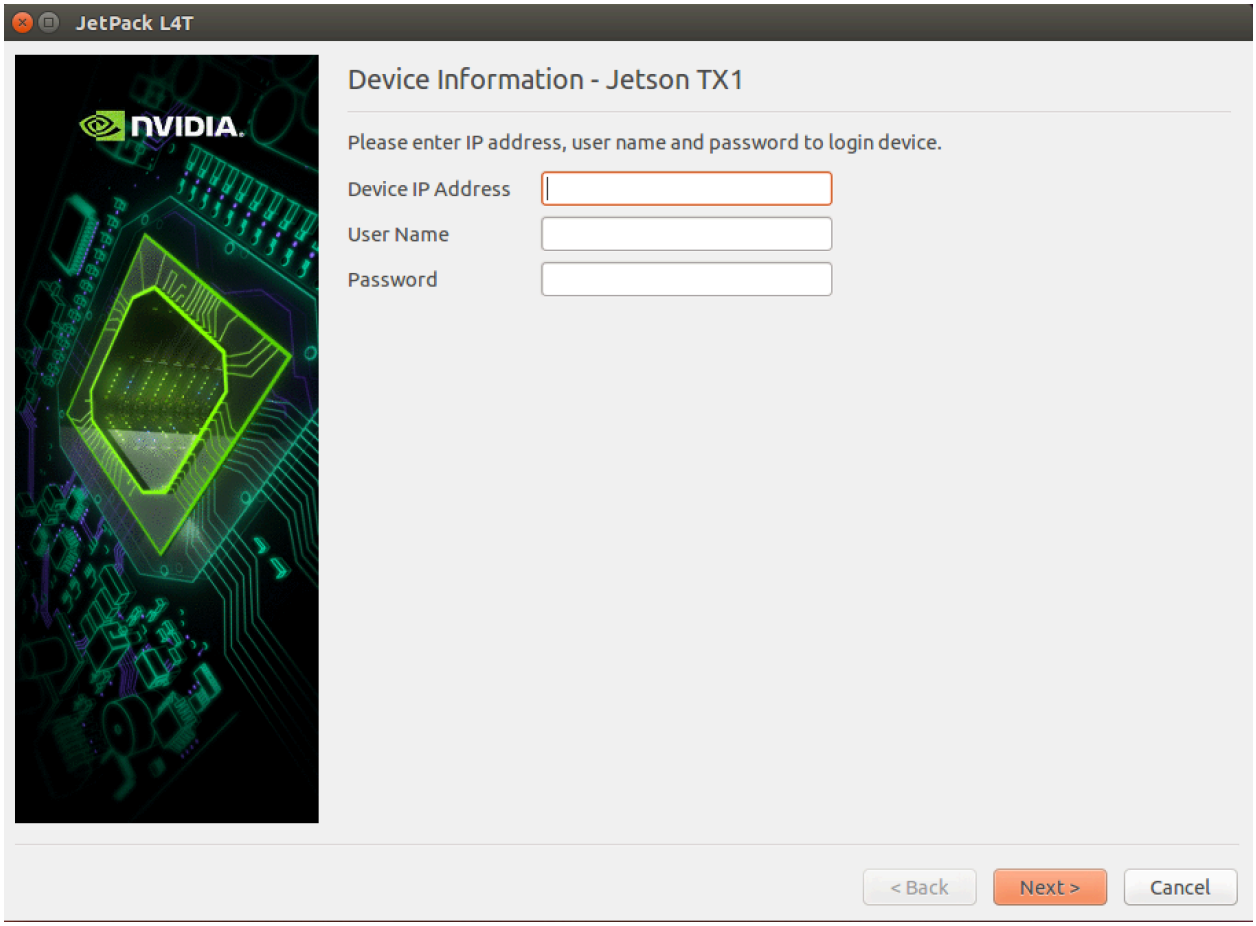
Press enter to continue and you'll move on to this:

In our observations, 899 out of 900 times, the installer seems to fail at finding the IP address of the TX1 system. You're going to have to find it another way most likely. At this point the TX1 should reboot and if you've got it hooked up to a monitor, a keyboard, and a mouse then you can login to it and run "ip addr show" in a terminal to find the IP of the TX1. You can also obtain the DHCP assigned address for the TX1 from your router if you know how.

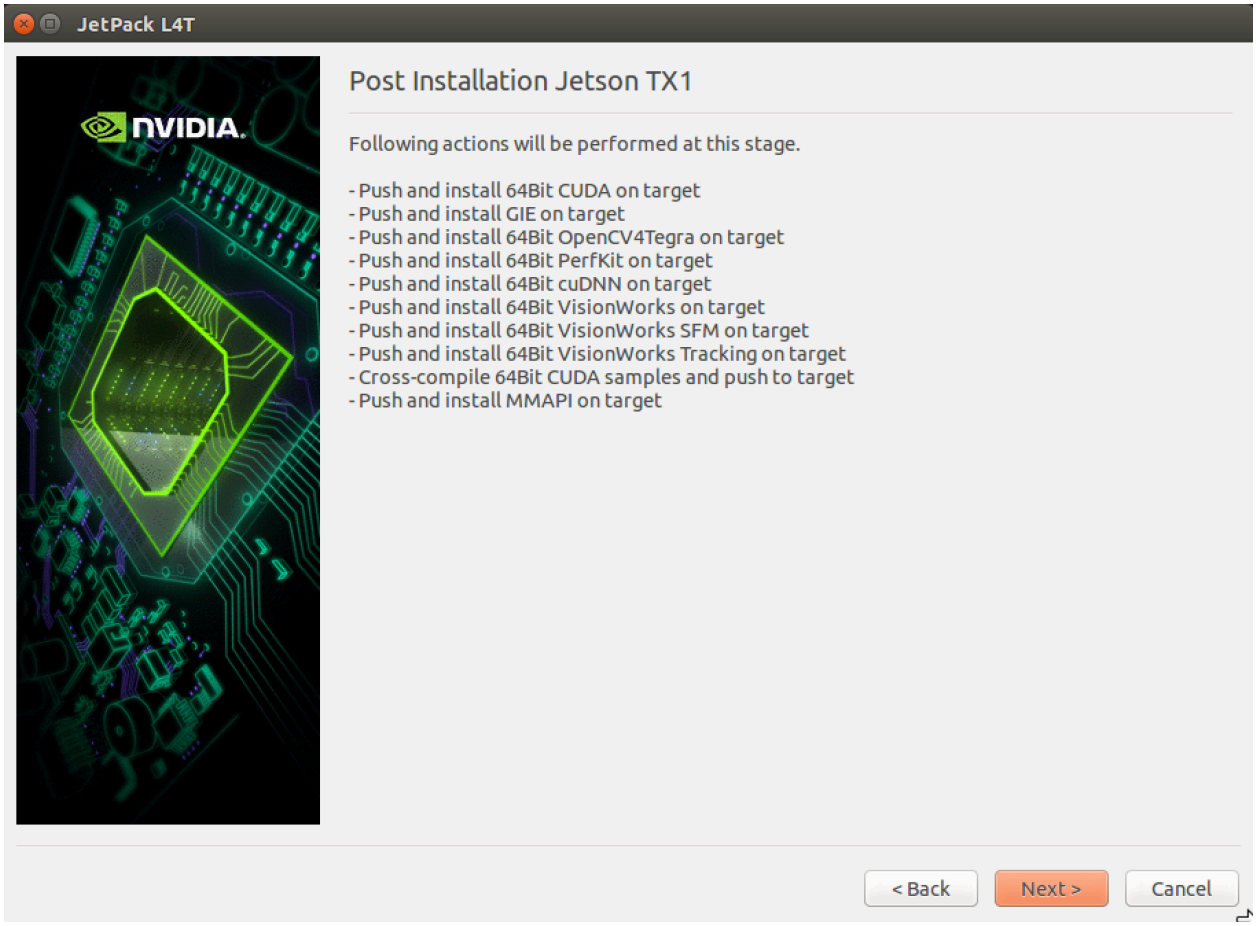
Once you have the IP address assigned to the TX1 then you can proceed to the next step:



Press the “2” key if the TX1 was not automatically discovered. You will be presented with a screen where you will need to enter the IP address of the TX1 system:

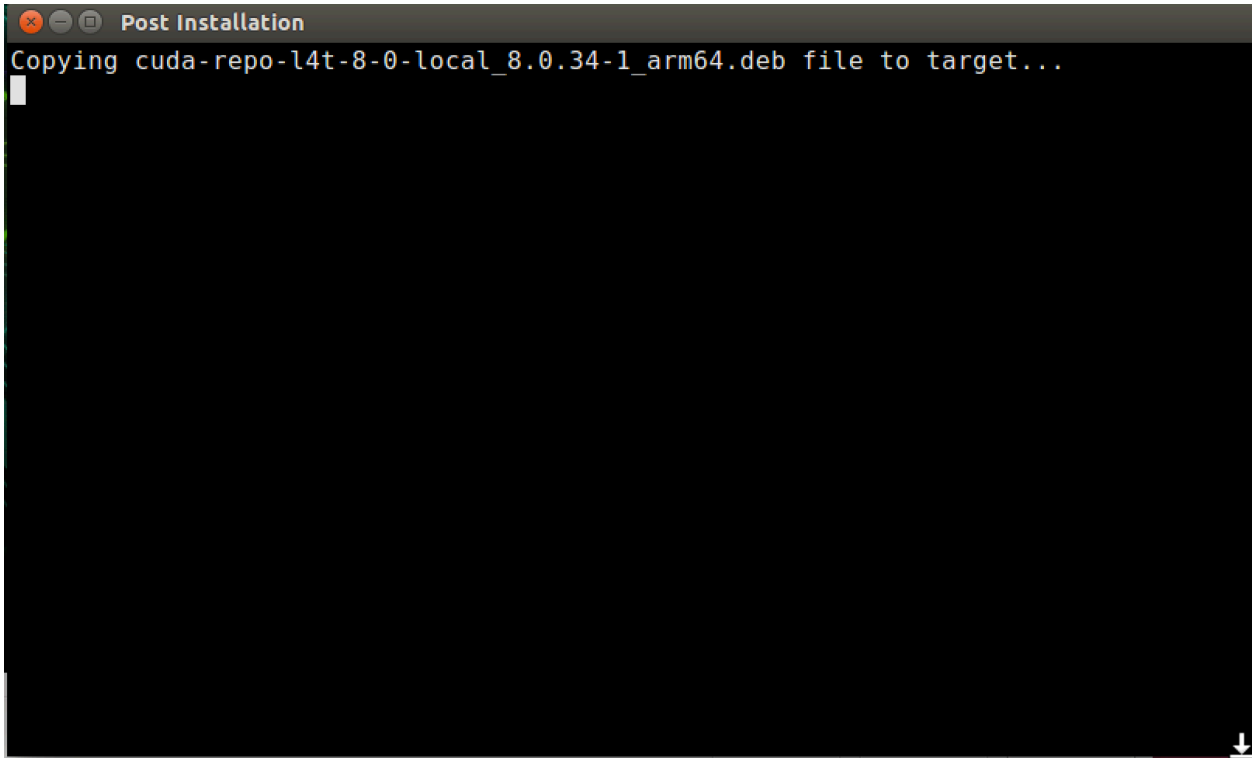


Enter the IP for the system as you have found it (like 10.x.y.z or 192.168.y.z) and the username will be “ubuntu” and the password will be “ubuntu”. Click “Next >” to proceed:

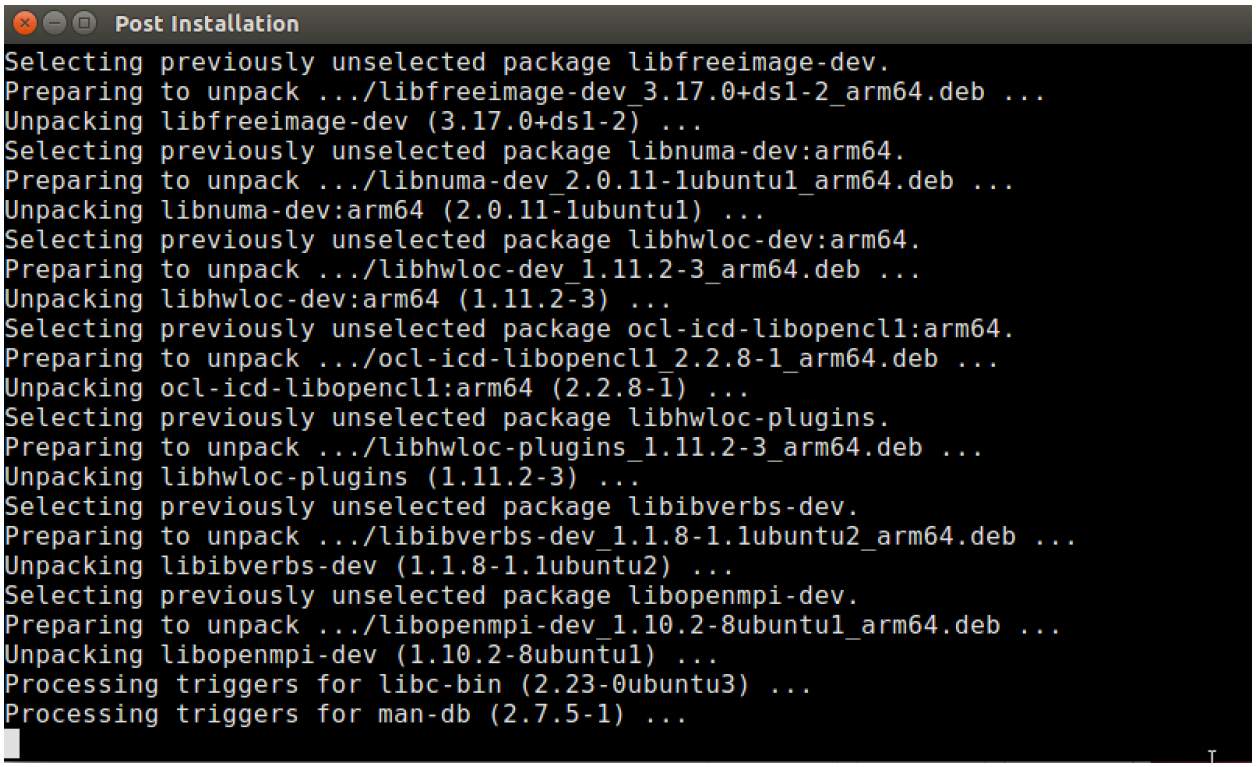


If it doesn't connect then you'll get an error message. Try rebooting the TX1, ensuring you can ping the IP address of it, verifying the ethernet cables are working and everything else you can to ensure proper networking.

Once it works, hit "Next >" again and the installer will open another command prompt:

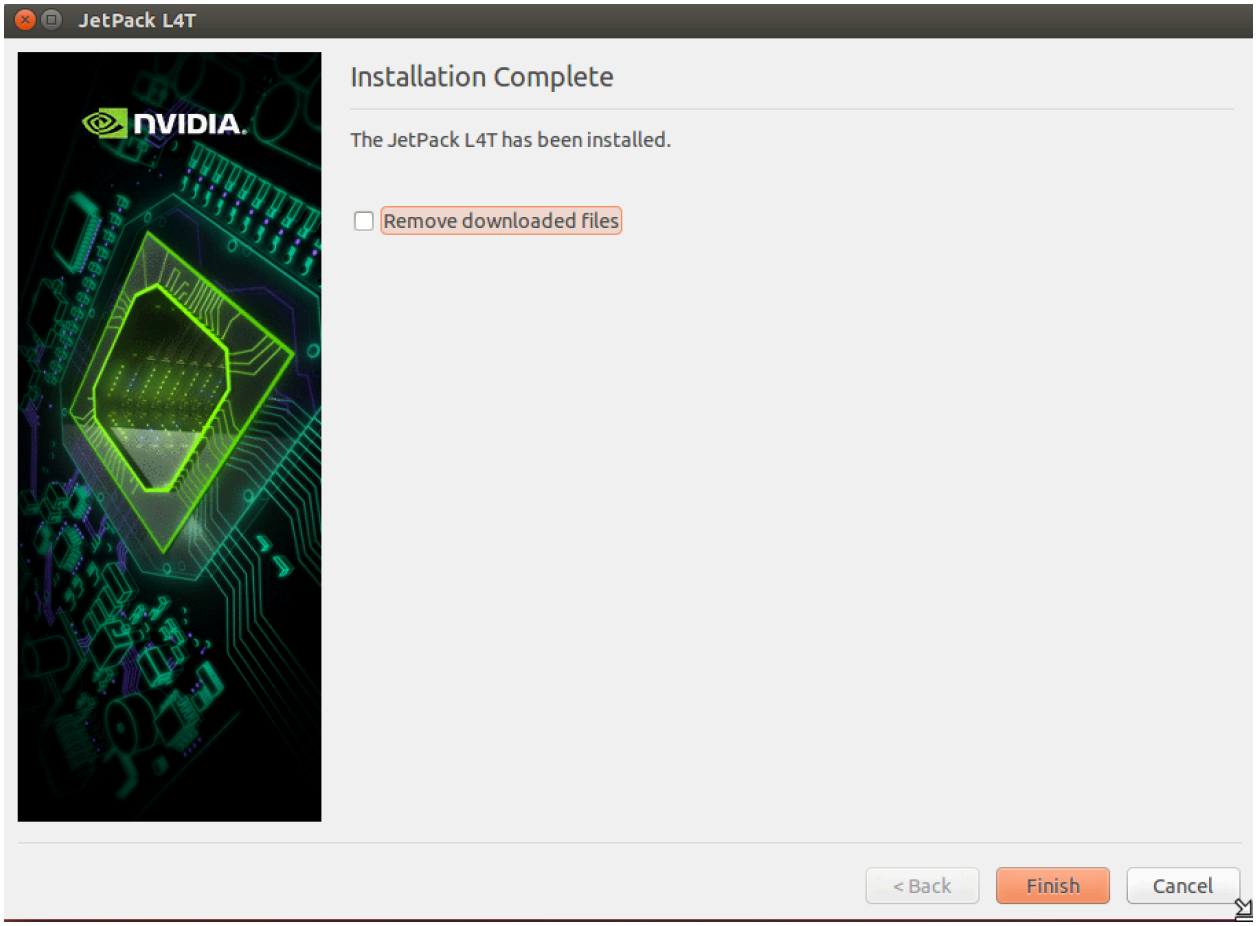


The process will continue for a while with installing various packages to the TX1:



It's possible things could error out again in here. If they do then check your network configuration and restart the process. Make sure your host system is not set to go to sleep automatically or if you're running this inside a VM that the host for the VM isn't set to sleep automatically.

Eventually it will finish and the terminal window will close and you'll see this screen:



You can choose to remove the downloaded files if you need to free up the disk space or you can keep them around if you need to perform this process again. Click "Finish" to complete the setup and close out the installer.

Great, you've now got Ubuntu 16.04 running on your TX1 with the latest CUDA libraries and examples.

Using the TX1 with FRC and other Tips

So you've got your TX1 updated and running. Now what?

That's a great question. What are your goals? Do you want to detect objects like goals or game pieces? Great, start learning about vision programming with OpenCV. A great place to start for those completely new to this world is by downloading GRIP and playing around with it.

GRIP is from the same people that make WPILib. It's quite useful for learning about filtering imaging and sorting out contours: https://github.com/WPIRoboticsProjects/GRIP

Hook up a USB camera to your TX1 and give GRIP a try first. You don't actually need a TX1 to try GRIP. You can try it on your laptop or maybe even your Ubuntu 14.04 host system. It's a good place to get started.

Once you're ready, move on to getting the GRIP executables to run on the TX1 automatically at boot. That's going to require some reading on your part and it's beyond the scope of this guide at the moment.

There are a few things we should probably try to clear up for the TX1. The TX1 is a great platform but it can't do everything for you automatically. It's not magic. It is an ARM CPU with an Nvidia GPU attached to it. The CPU and GPU share memory but NOT in a way that automatically allows a set of libraries like OpenCV to access both without input from a programmer.

Due to this, you will need to write your code so that it can take advantage of the GPU accelerated features. Not all vision processing algorithms and techniques will benefit from being GPU accelerated.

Further, not all programming languages support GPU acceleration with OpenCV. Most commonly, Python and Java do NOT have support for GPU acceleration with OpenCV. It's possible to find some projects where programmers have hacked in support but it is not officially supported by OpenCV as of time of writing.

Additionally, GRIP does not currently have support for GPU acceleration. GRIP will run on the TX1 though and can output code that can be run on the TX1 but you will need to rewrite portions of the code to take advantage of GPU acceleration if that is what you are trying to do.

Powering the TX1 is tricky. From experience. it does not function well with large power fluctuations. Nvidia's engineers have released a few versions of the carrier board to help with this but you will still want to run a voltage regulator between the board and the power supply. Team 900 has had a lot of success with this particular convert from DROK in actual FRC competitions: http://a.co/aaISgDh

Do not use the VRM (Voltage Regulator Module) to power both your Radio and the TX1. If you do it's entirely possible that your VRM will no longer function after pulling too much current. You

might be able to use a second VRM just for the TX1 but you are much better off using the voltage regulator linked above as it is cheaper and has been tested in competitions.

The TX1 carrier has a smaller board on it with a camera attached to it. This board has been known to cause issues when also using a USB camera and you may need to remove it if you don't want to work around the issues. If you do remove it then take the time to power off your TX1 before carefully unscrewing and removing the camera board.

The installation process for JetPack is not bulletproof and sometimes the errors are hard to debug. If all else fails, keep trying. Get a different host system. Try a different networking configuration. Try things again after walking away for a couple of hours. Be patient and ask some questions if you get stuck. Hopefully you'll get the help you need.

You can't program the Jetson using LabVIEW that we know of. This seems to be a highly requested feature so perhaps National Instruments should enable the Jetson as a target but I'm not sure if that will happen anytime soon. You can program the Jetson using a lot of other languages though.

Networking the TX1 on a robot is hard. It's 2017 (as of writing) and FRC is now using radios with only two Ethernet ports. Guess what? Your RoboRIO is going to be plugged into one and your Jetson into the other and you're not going to have a place to put a laptop for debugging then. You'll probably want to add a switch to your robot. Check around for Ethernet switches that can run off of 5V or 12V and add another regulator into the mix.

We recommend you set a static IP address for the TX1. You can try to use DHCP but it might not work and Dynamic DNS (Bonjour/ZeroConf) likely won't work either. You should also restrict your communication to the TCP/UDP ports allowed by the FRC rules. These could vary from year to year so consult the manual before you pick a TCP/UDP port you should not be using.

Unfortunately you can't consult the game manual about static IP address assignments because the people who wrote it don't understand basic networking concepts (turns out you need both ports and IPs/Hostnames to communicate on a network - ball is in your court FRC Q&A folks) but you can get some information from the helpful WPI folks: https://wpilib.screenstepslive.com/s/4485/m/24193/l/319135-ip-networking-at-the-event

Resources

- Nvidia JetPack install guide: http://docs.nvidia.com/jetpack-l4t/index.html#developertools/mobile/jetpack/l4t/2.3/jetpack_l4t_install.htm

- The eLinux Wiki page on the TX1: http://elinux.org/Jetson_TX1

- Reasonable Case for the TX1: https://www.pugetsystems.com/store/item.php?cat=Case&id=11365&com=d41d8cd9

- Jetson Hacks Blog posts about setting up the TX1:
    - Part 1: http://www.jetsonhacks.com/2016/08/16/thoughts-on-programming-languages-and-environments-jetson-dev-kits/
    - Part 2: http://www.jetsonhacks.com/2016/08/17/thoughts-on-programming-languages-and-environments-part-ii-jetson-dev-kits/
    - Part 3: http://www.jetsonhacks.com/2016/08/17/thoughts-on-programming-languages-and-environments-part-iii-jetson-dev-kits/
    - Part 4: http://www.jetsonhacks.com/2016/08/18/thoughts-on-programming-languages-and-environments-part-iv-jetson-dev-kits/

- Team 900's white paper on connecting the TX1 for FRC and using it for goal detection: https://drive.google.com/file/d/0B8hPVHrmVeDgUWtLbDVmbFV3Y3M/

- Other resources from Team 900: http://team900.org/labs

- Team 900's Github: https://github.com/FRC900

- GRIP Project on GitHub: https://github.com/WPIRoboticsProjects/GRIP

- GRIP Wiki: https://github.com/WPIRoboticsProjects/GRIP/wiki

- The Nvidia Developer Forums for Embedded Systems: https://devtalk.nvidia.com/default/board/139/jetson-embedded-systems/

- Sometimes you can get lucky and get support on https://www.chiefdelphi.com/ but you know, it's ChiefDelphi so YMMV.

- We've setup support@team900.org for those looking for answers from us. We can't promise we'll respond as quickly as you need it but we'll do our best.