

JETSON HOW-TO

This guide is a superset of

<https://github.com/xlzl/libfreenect2/wiki/Jetson-TK1-HOWTO>

FLASH THE DEVICE:

Use JetPack downloadable here:

https://github.com/OpenPTTrack/open_pttrack/tree/jetson-dev/jetpack

Follow the figures (In particular, take care to the figure in which you should put the device in FORCE RECOVERY mode. The instructions showed in ther terminal window are bugged, follow the ones in this guide!).

After the flashing phase you should be able to ssh-ing the device if it is powered on and attached to your Local Area Network. Try to connect to it with this command:

```
ssh -X ubuntu@tegra-ubuntu.local
```

The default password is:

ubuntu

If the above command does not work check your network with IP scanning tools and see whether you see the device.

If you change the default username & password the following commands should be changed accordingly.

PRELIMINARY STEPS:

1) Enabling USB 3.0 Full performance

USB 3.0 must be enabled for full performance. (With 30 FPS, JPEG stream consumes 20MB/s, and depth stream consumes 85MB/s). Edit `/boot/extlinux/extlinux.conf` to change `usb_port_owner_info=0` to `usb_port_owner_info=2` to enable USB 3.0.

2) Add udev rule for Kinect v2: create `/etc/udev/rules.d/90-kinect2.rules` with the following content:

```
# ATTR{product}=="Kinect2"  
SUBSYSTEM=="usb", ATTR{idVendor}=="045e",  
ATTR{idProduct}=="02c4", MODE="0666"  
SUBSYSTEM=="usb", ATTR{idVendor}=="045e",  
ATTR{idProduct}=="02d8", MODE="0666"  
SUBSYSTEM=="usb", ATTR{idVendor}=="045e",  
ATTR{idProduct}=="02d9", MODE="0666"
```

INSTALL UTILITIES & PERMITS UNIVERSE+MULTIVERSE

```
sudo apt-add-repository universe  
sudo apt-add-repository multiverse  
sudo apt-get update  
sudo apt-get install bash-completion command-not-found locate git gitg vim
```

INSTALL&CONFIGURE ROS BASE SYSTEM

```
sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb_release -sc) main" >  
/etc/apt/sources.list.d/ros-latest.list'  
sudo apt-key adv --keyserver hkp://pool.sks-keyservers.net --recv-key 0xB01FA116  
sudo apt-get update  
sudo apt-get install -y ros-indigo-desktop  
sudo apt-get install -y python-rosinstall  
sudo apt-get install -y ros-indigo-compressed-depth-image-transport  
ros-indigo-compressed-image-transport  
sudo rosdep init  
rosdep update  
echo "source /opt/ros/indigo/setup.bash" >> ~/.bashrc  
source ~/.bashrc  
mkdir -p ~/workspace/ros/catkin/src  
cd ~/workspace/ros/catkin  
catkin_make --force-cmake  
echo "export KINECT_DRIVER=openni" >> ~/.bashrc  
echo "export LC_ALL=C" >> ~/.bashrc  
echo "source /home/ubuntu/workspace/ros/catkin/devel/setup.bash" >> ~/.bashrc
```

INSTALL POINT CLOUD LIBRARY

```
sudo add-apt-repository ppa:v-launchpad-jochen-sprickerhof-de/pcl
sudo apt-get update
sudo apt-get install libpcl-1.7-all
```

INSTALL & CONFIGURE NTP

Follow https://github.com/OpenPTrack/open_ptrack/wiki/Time%20Synchronization

INSTALL CUDA

```
wget
http://developer.download.nvidia.com/compute/cuda/6.5/rel/installers/cuda-repo-l4t-r21.2-6-5-prod\_6.5-34\_armhf.deb
sudo dpkg -i cuda-repo-l4t-r21.2-6-5-prod_6.5-34_armhf.deb
sudo apt-get update
sudo apt-get install -y cuda-toolkit-6-5
sudo usermod -a -G video $USER
echo "# Add CUDA bin & library paths:" >> ~/.bashrc
echo "export PATH=/usr/local/cuda/bin:$PATH" >> ~/.bashrc
echo "export LD_LIBRARY_PATH=/usr/local/cuda/lib:$LD_LIBRARY_PATH" >> ~/.bashrc
source ~/.bashrc
```

INSTALL LIBFRENECT2 & ADD THE SCRIPTS FOR THE MAX PERFORMANCE

By default, only one core of the jetson is active and it works at the minimum frequency. Furthermore, even the memory bus, the GPU frequency and the GPU memory frequency should be set for the max performance.

You can done this using the scripts in the libfreenect2 repository. There are several scripts for enabling the max performance, check the Status of the card and for the disabling of the autosuspending mode of the USB (to be used after having connected the Kinect v2).

```
cd ~
```

```

sudo apt-get install -y build-essential libturbojpeg libtool autoconf libudev-dev
cmake mesa-common-dev freeglut3-dev libxrandr-dev doxygen libxi-dev
libjpeg-turbo8-dev
git clone https://github.com/openptrack/libfreenect2.git
cd libfreenect2
git checkout jetson-dev
cd depends
./install_deps.sh
sudo ln -s /usr/lib/arm-linux-gnueabi/libturbojpeg.so.0.0.0
/usr/lib/arm-linux-gnueabi/libturbojpeg.so
cd ..
cd jetson_scripts
chmod +x *
sudo cp * /usr/bin

```

check Protonect

Kinect v2 connect instructions:

- 1) Before connecting the Kinect, open a terminal and launch the following command:

```
sudo MaxPerformance
```

Don't care of the output and invoke this command:

```
sudo Status
```

If everything is good, you should see something like this:

CPU's active (it should be 0-3):

```
0-3
```

CPU's freq (it should be 2032500000):

```
2320500
```

```
2320500
```

```
2320500
```

```
2320500
```

CPU governor (it should be userspace):

```
userspace
```

```
userspace
```

```
userspace
```

```
userspace
```

GPU clock (it should be 852000000):

```
852000000
```

GPU memory clock (it should be 924000000):

```
924000000
```

If something is not similar to these lines try to reboot and check the passages above.

- 2) Connect the already powered-on Kinect v2 cable to the USB port of the Jetson.
- 3) Wait 3 seconds
- 4) Invoke

```
NoAutosuspending
```

- 5) Invoke

```
sudo Status
```

At the end of the Status you should see the Autosuspending state with all -1 (no 2). If this is not the case something went wrong. Probably the Kinect v2 is powered off and it is not listed there or you have not done well the PRELIMINARY STEPS. Try to redo the steps and/or reboot the device. Take care of the order of the instructions!!

- 6) The Kinect v2 is correctly connected, now we will check if Protonect works correctly.

```
cd ../examples/protonect/  
mkdir build && cd build  
cmake ..  
make -j4  
../bin/Protonect
```

You should see [with some delay (we are trying to ssh-ing a lot of information)] three windows opening, one with the depth information, one with the color information and one with the ir information. Don't worry about the very low framerate, we are seeing a stream of information through an encrypted channel. If you don't see anything and on the terminal there are a lot of

```
[DepthPacketStreamParser::handleNewData] subpacket incomplete  
(needed X received Y)]
```

probably the jetson is not in max performance mode or the Kinect has not been attached correctly (take care of the order specified previously!).

If you see some USB errors, probably Protonect has been linked with the uncorrect version of libusb or the USB is in Autosuspending mode. Try to reboot and/or check ldd Protonect.

If Protonect works, you can install the libraries in the system.

```
sudo make install
```

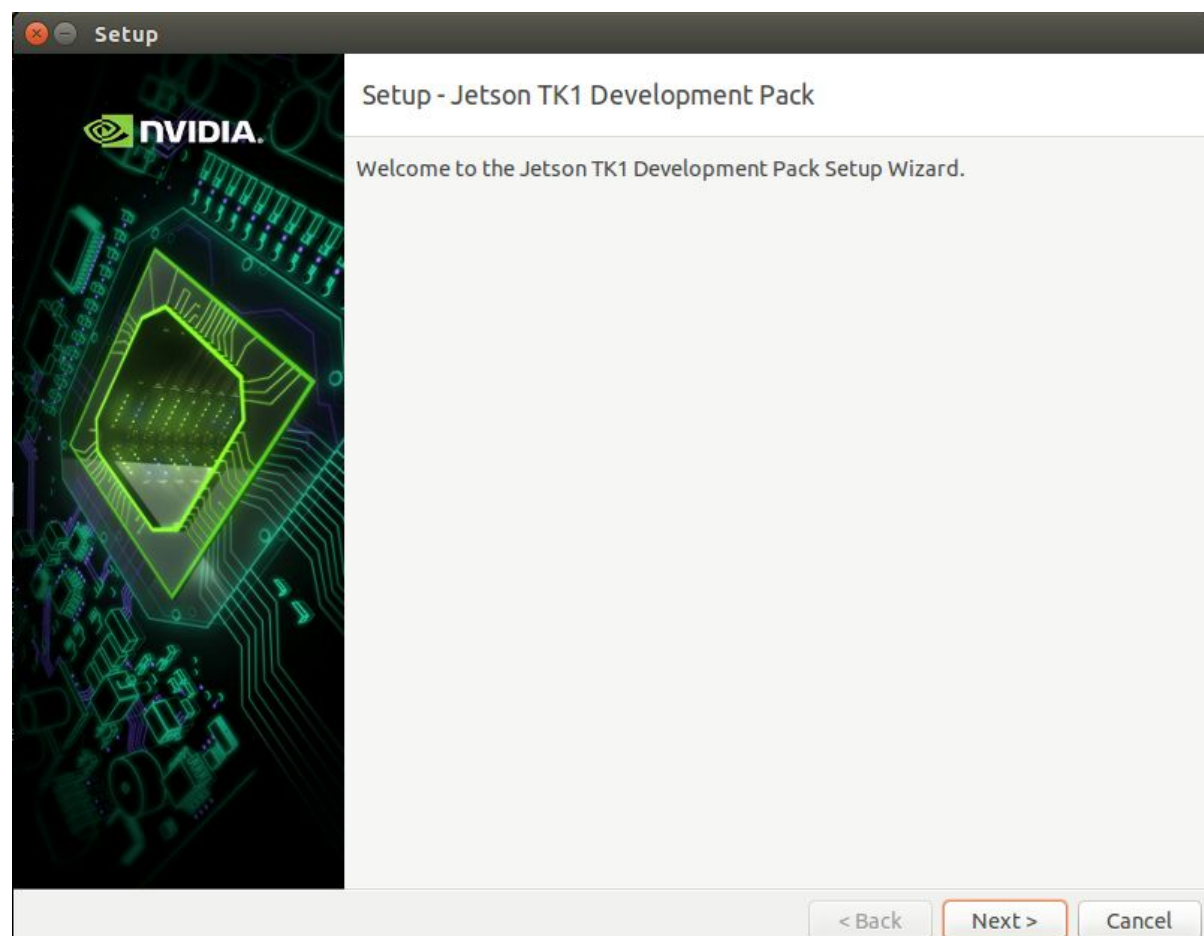
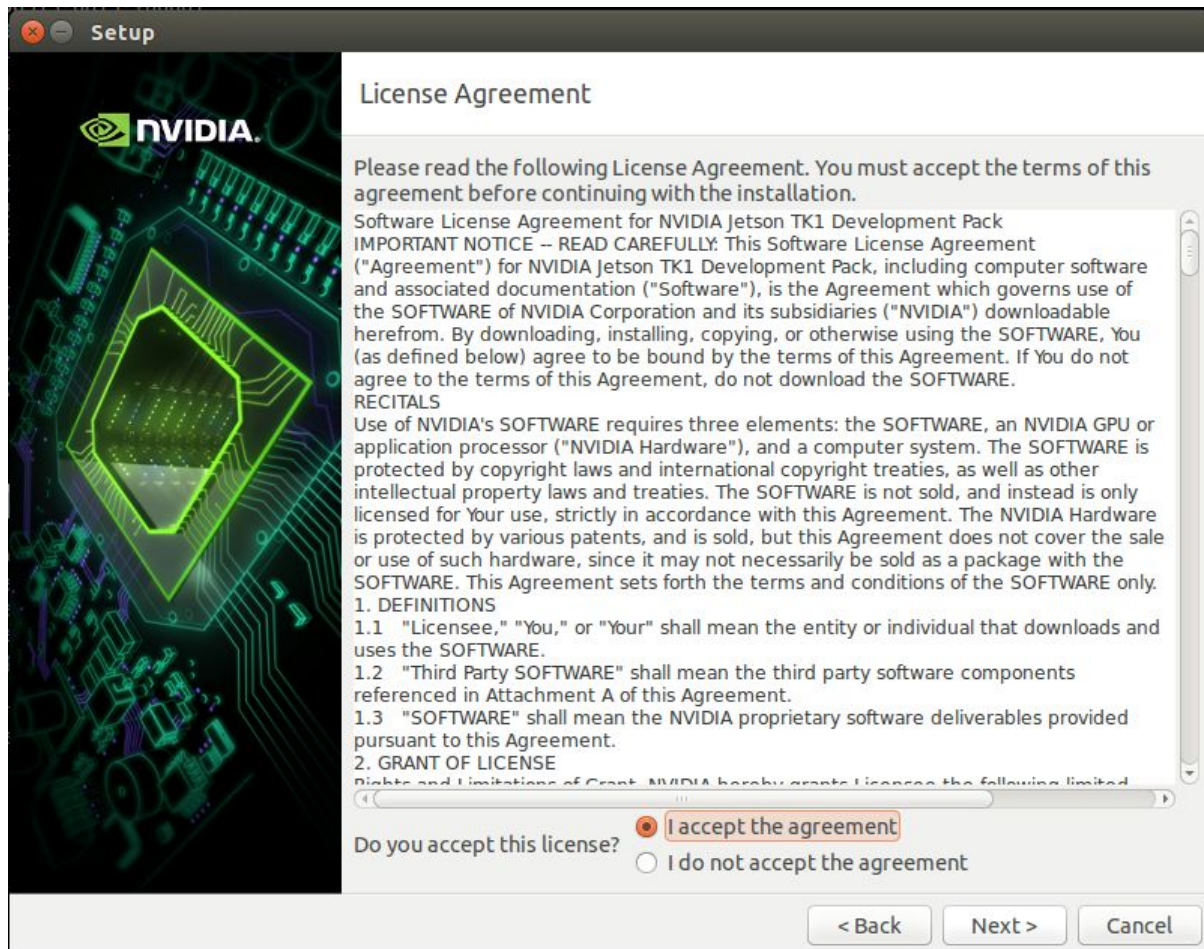
INSTALL IAI_KINECT2

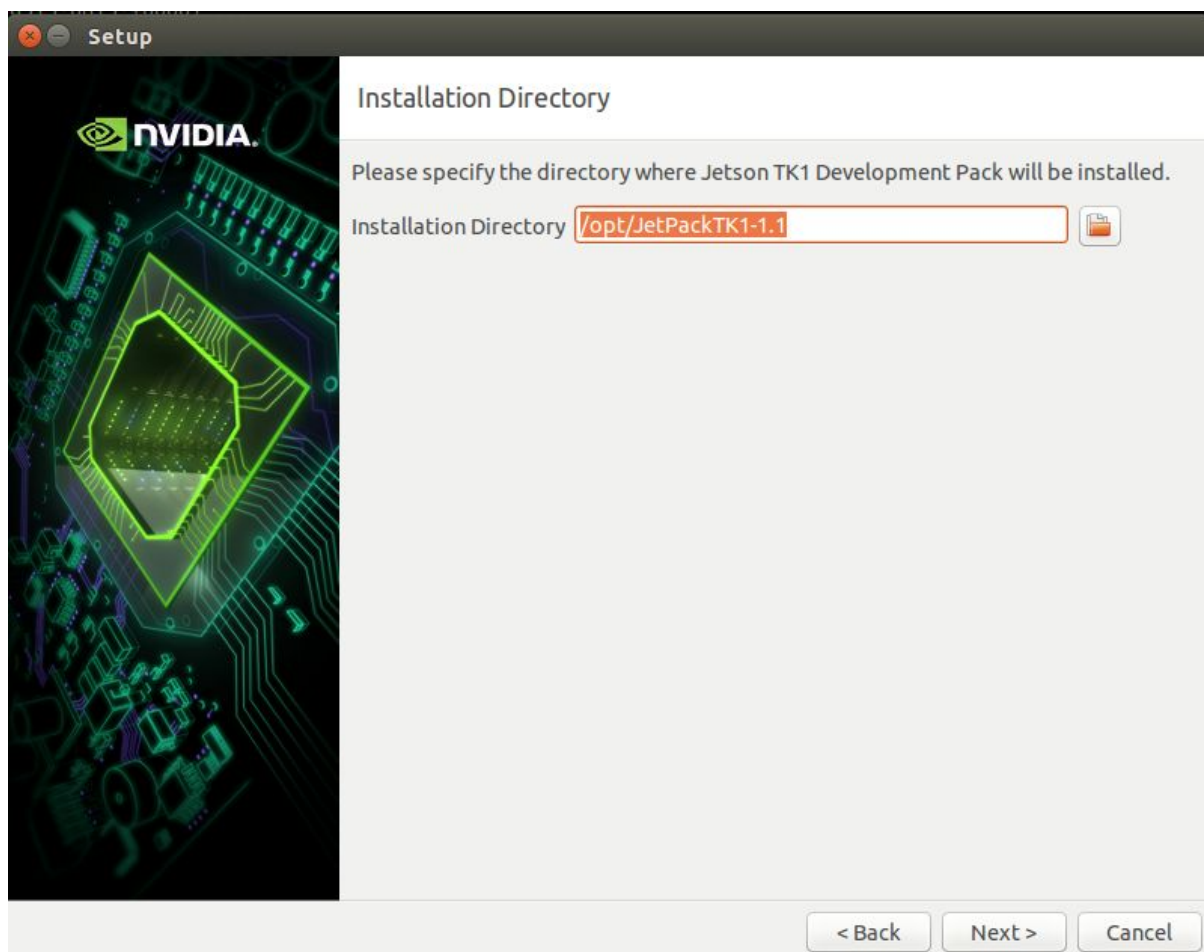
```
roscd  
cd ../src  
git clone https://github.com/OpenPTrack/iai_kinect2  
git checkout jetson-dev  
Catkin  
(only if you see internal compiler error)  
CatkinJ1
```

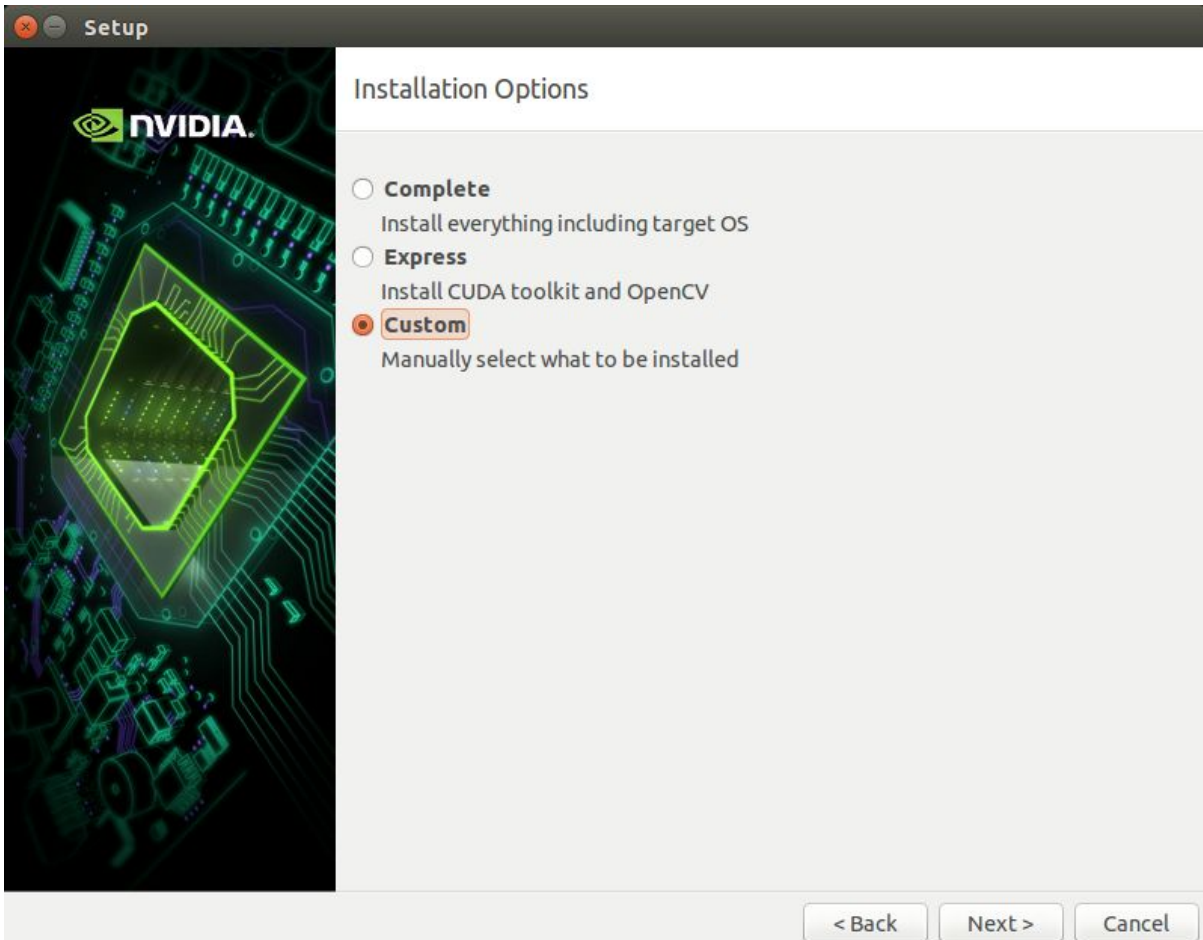
INSTALL OPEN_PTRACK

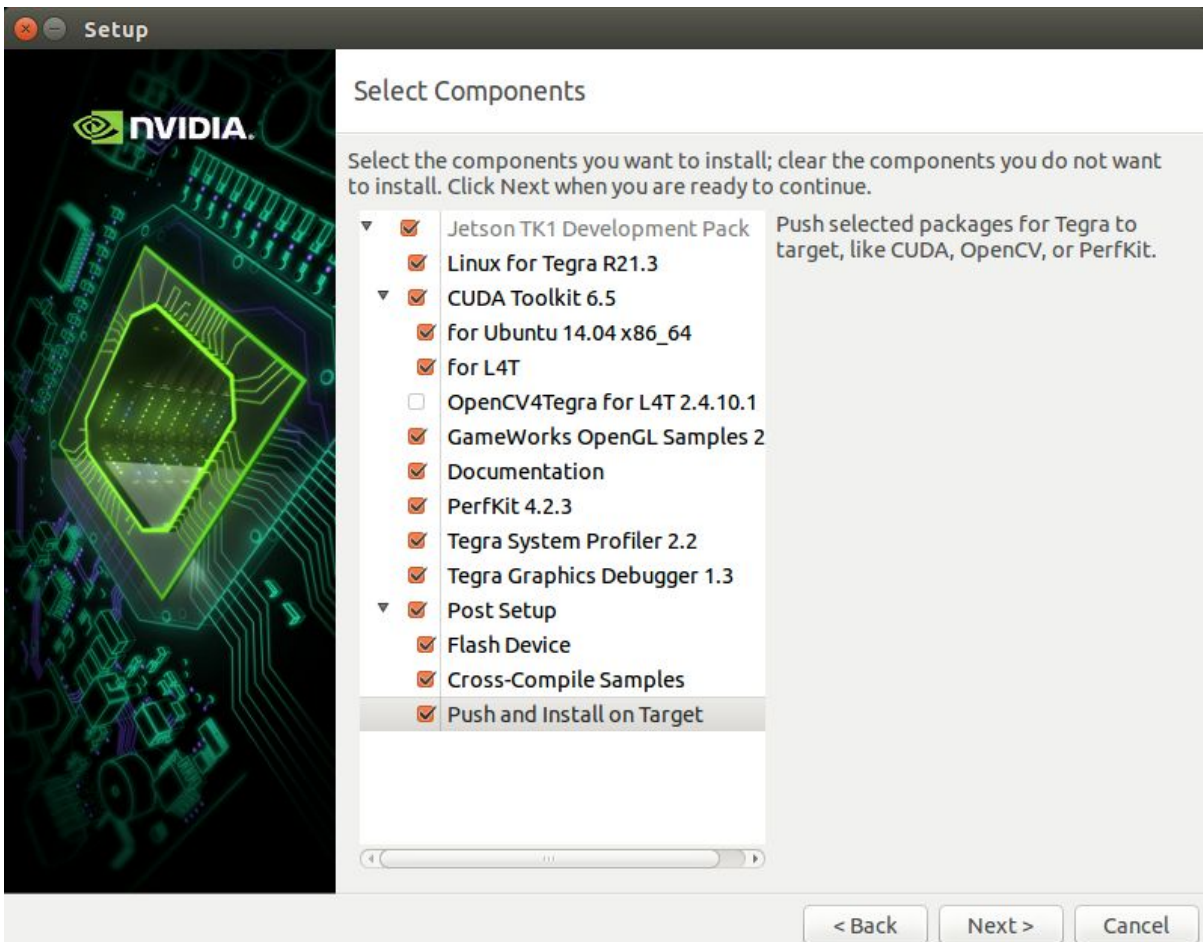
```
roscd  
cd ../src  
git clone https://github.com/OpenPTrack/open_ptrack  
git checkout jetson-dev  
cd ~/workspace/ros/catkin/src/open_ptrack/scripts  
chmod +x *.sh
```

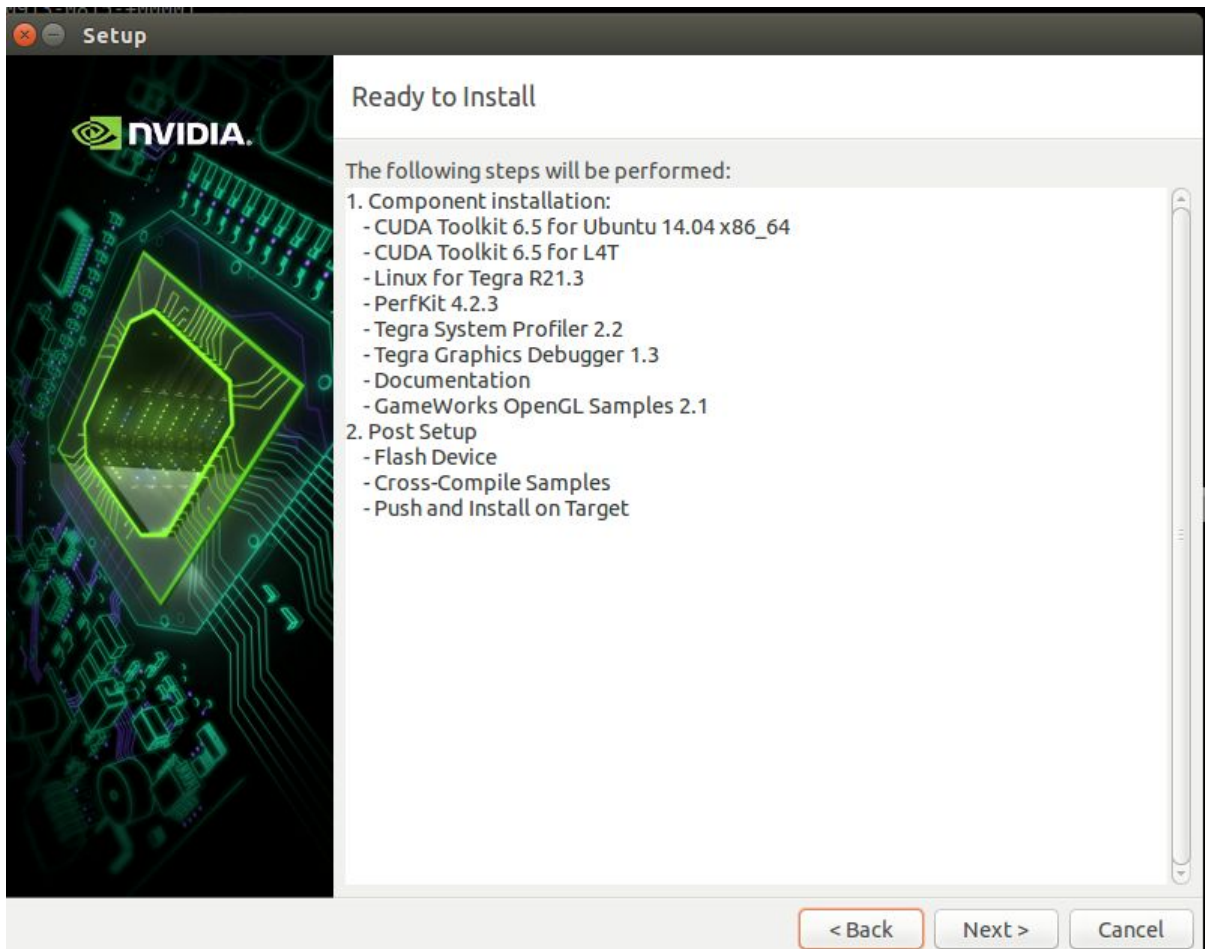
```
./calibration_toolkit_install.sh  
cd ~/workspace/ros/catkin  
catkin_make --pkg calibration_msgs  
catkin_make --pkg opt_msgs  
catkin_make --force-cmake
```












Setup



Download Transfer Settings

☐ Specify a proxy server to download components

Server Address

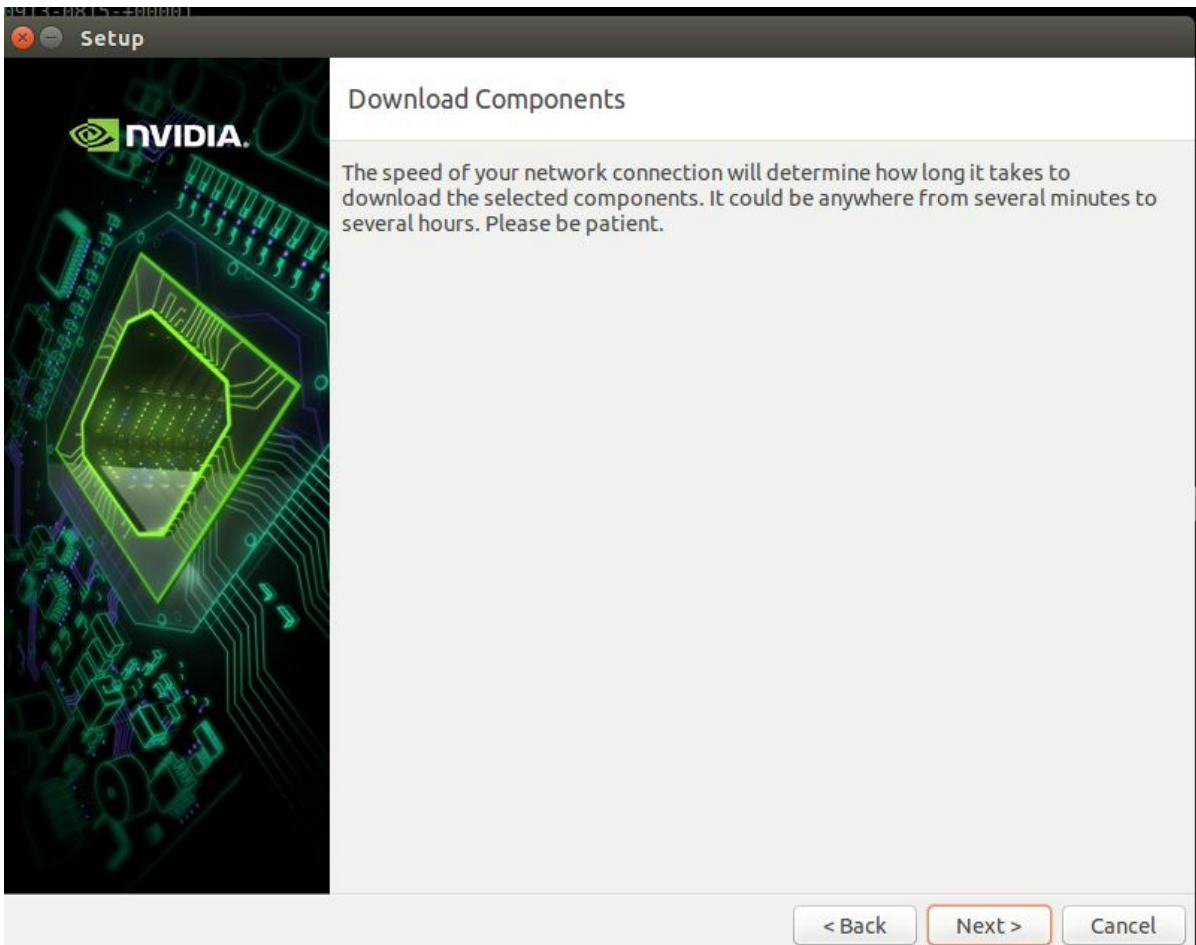
IP Port

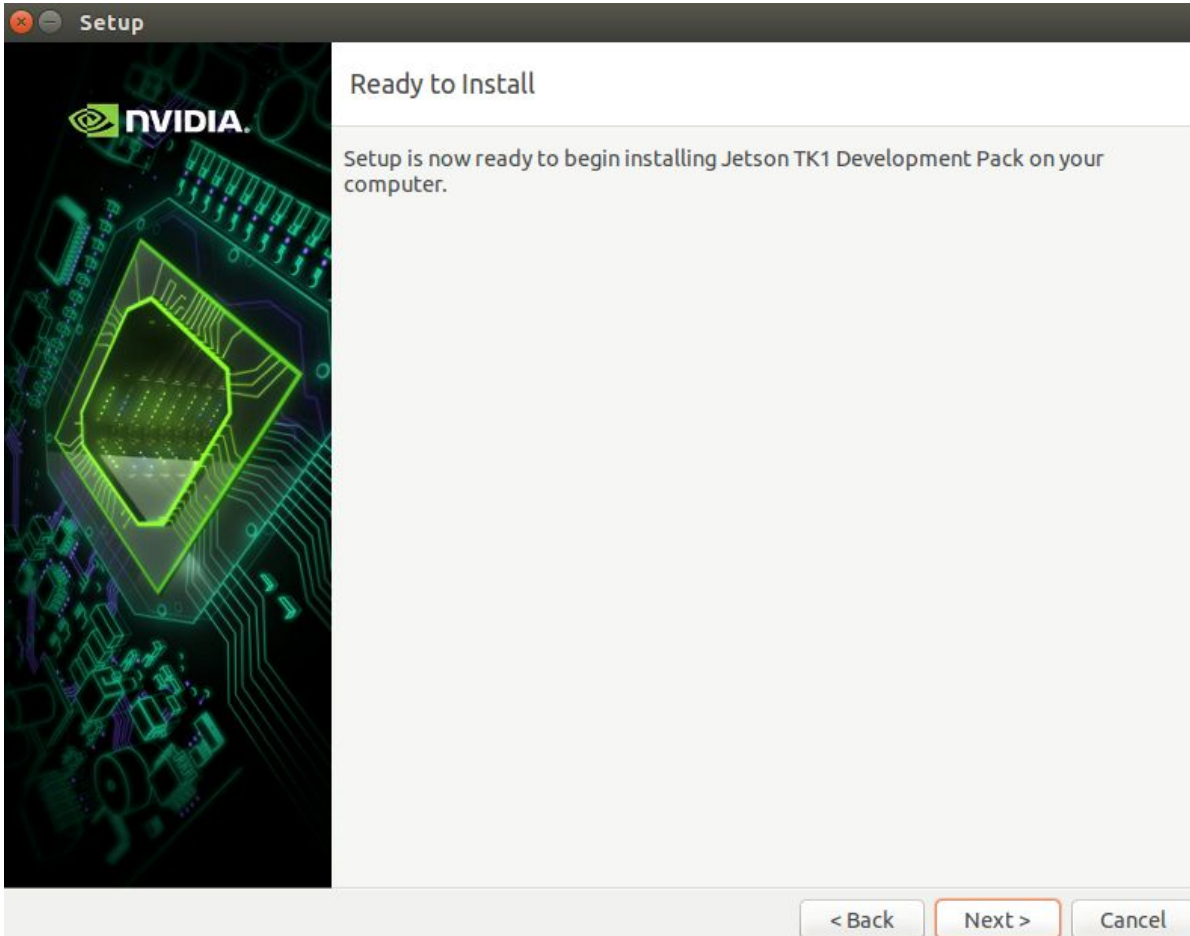
Account

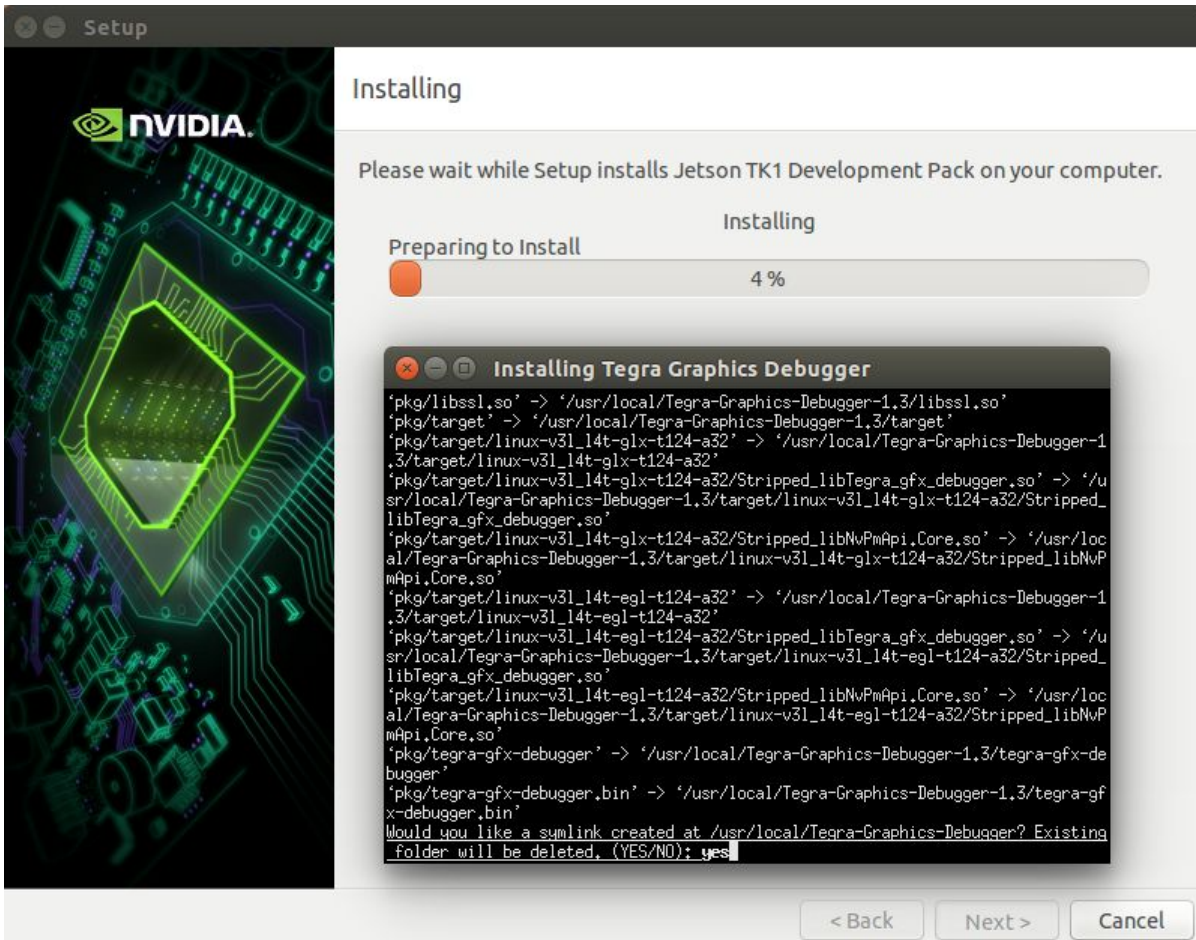
Username

Password

< Back Next > Cancel









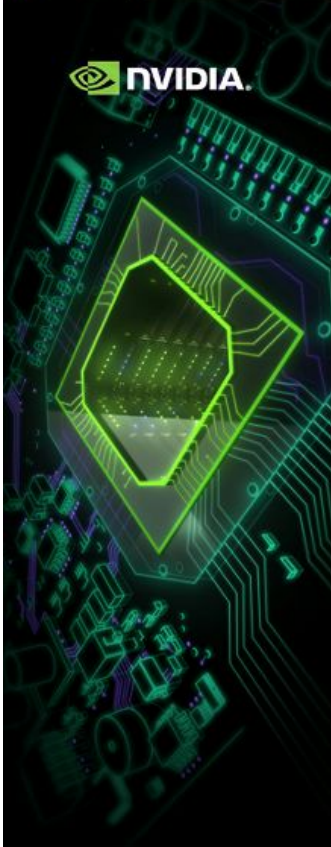
Flash Device

Flash the OS to devkit and this will take about 30 minutes to complete flashing. Please be patient.

< Back

Next >

Cancel



Input Device Information

In order to push CUDA/OpenCV/PerfKit toolkit to Jetson TK1, please input its IP address, username, and password.

IP address can be queried by running "ifconfig" command on target.

The default username/password are "ubuntu/ubuntu" without quotation.

IP Address :

Username :

Password :