

## Configuring the Jet Robot

This document describes the first steps you will need to go through in order to prepare the Jetson board for use with Jet.

After opening the NVIDIA Jetson box, you will need the following items:

- Jetson board
- micro USB cable
- AC power adapter

In addition, you will need:

- an ethernet cable
- a router that is connected to the Internet

## Preparing the Jetson Board

- Plug one end of the ethernet cable into the Jetson board and plug the other end into the router
- Do not plug in power adapter just yet (you will plug in power during the OS flash step)

## Running JetPack

Download and install the [NVIDIA Jetpack](#). You will need to be running Ubuntu 14.04 on your host computer. The newer versions of JetPack might give you problems during installation. Since JetPack 2.3.1 already has all the features required for a TX1, using it is recommended.

Once Jetpack is up and running perform the following actions:

- Select TX1 as target board
- Click the Clear Actions button (upper right corner). This will de-select all of the options.
- Select the following options for install (Click on 'no action' and change it to 'Install' only for the following packages):
  - Driver for OS
  - File System
  - Flash OS
  - CUDA Toolkit for L4T
  - cuDNN Package
  - OpenCV for Tegra
- also select the box Automatically resolve dependency conflicts
- On the Network Layout page:
  - Select Device accesses Internet via router/switch
- On the Network Interface Selection page:

- select eth0 (assuming your host computer only has 1 ethernet)

## Flash the OS

After you have started the installation, the JetPack installer will download files and install them. As the installer continues, it will reach a point where it will flash the OS to the Jetson board. At this point, the installer will ask you to put the board into Force USB Recovery Mode:

- Connect the micro USB cable to the Jetson TX1 and the host computer
- Plug the AC power adapter into AC power and connect the power jack to the board
- Enter USB Recovery Mode using the following sequence (These instructions will pop up on your host computer's screen):
  - Press and release Power button
  - Hold down the Force Recovery button, then press and release Reset (while holding down Force Recovery)
  - Once you complete doing this, type lsusb on the terminal of your host screen you should see a device with the vendor name as 'Nvidia Corp'. The jetpack will start flashing the OS on to the TX1.
  - It is recommended that you connect the TX1 to a HDMI enabled monitor right from the beginning. The flashing and copying softwares to your TX1 will reach a point where the command window will try to search for your Jetson's IP. For some reason, the entire flashing process works without any problems all the time when the TX1 is connected to WiFi instead of ethernet. So login into your TX1 through the screen with the username: 'UBUNTU' and password: 'ubuntu' and connect the wifi. Once this is done, JetPack will automatically find your TX1's wifi IP and start copying packages like CUDA, etc.
  - Begin the flashing process on the host by pressing Enter. This will take several minutes.

## Installing the Jet software

Once the install is complete, you can go ahead and disconnect the micro USB cable. Now we will install the rosjet libraries and utilities. On your host computer, download the lab1 zip file from your instructor. Find the appropriate files for TX1.

Connect to the Jetson via ssh The username is 'ubuntu' and the password is 'ubuntu'. You can find it's ip address by looking at your router's configuration. Note: JET\_IP\_ADDRESS is the IP address found on the Jetson TX1. YOUR\_IP\_ADDRESS is the IP address of the host computer.

- On the Jetson, install git

```
sudo apt-get install git
```

- On the Jetson, make the ROS workspace directory

```
mkdir -p ~/catkin_ws/src; cd ~/catkin_ws/src
```

- On your host computer, extract the lab1 zip file. Then copy the rosjet folder to the Jetson

```
scp -r ~/lab1_building_robot-code/rosjet
ubuntu@JET_IP_ADDRESS:~/catkin_ws/src/rosjet (Must be done using a new tab as
user@hostcomputer and not as ubuntu@tegra-ubuntu)
```

- On the Jetson, run the ros configuration script

```
./rosjet/TX1_rosjet_install.sh
```

- Configuration of the Jetson is now complete; reboot the Jetson board before continuing.

- On the host computer, create a folder for ROS packages:

```
mkdir -p ~/catkin_ws/src; cd ~/catkin_ws/src
```

- On the host computer, copy the jethost folder to the workspace

```
cp -r ~/lab1_building_robot-code/jethost ~/catkin_ws/src/jethost
```

- On the host computer, run the jethost installation script

```
./jethost/jethost_install.sh
```

**Configuring ROS Environment Parameters (Before this step make sure you have static ip addresses on the host and the jet. If not you would have to go through this process everytime your IP address changes.)**

- On your host computer, lookup your own IP address

```
ifconfig wlan0 if using WiFi
```

```
ifconfig eth0 if using ethernet
```

- Lookup the Jetson's IP address using your router settings or connecting Jet to an HDMI monitor

- Edit your /etc/hosts

```
sudo echo "YOUR_IP_ADDRESS jethost" >> /etc/hosts
```

```
sudo echo "JET_IP_ADDRESS jet" >> /etc/hosts
```

- Set ROS\_MASTER\_URI in .bashrc

```
echo "export ROS_MASTER_URI=http://JET_IP_ADDRES:11311" >> ~/.bashrc
```

```
'echo "export ROS_HOSTNAME=YOUR_IP_ADDRESS" >> ~/.bashrc'
```

```
'echo "export ROS_IP=YOUR_IP_ADDRESS" >> ~/.bashrc'
```

- Source .bashrc

```
source ~/.bashrc
```

- SSH into Jet and perform similar updates

```
sudo echo "YOUR_IP_ADDRESS jethost" >> /etc/hosts
```

```
sudo echo "JET_IP_ADDRESS jet" >> /etc/hosts
```

```
echo "export ROS_MASTER_URI=http://JET_IP_ADDRESS:11311" >> ~/.bashrc
```

```
'echo "export ROS_HOSTNAME=JET_IP_ADDRESS" >> ~/.bashrc'
```

```
'echo "export ROS_IP=YOUR_IP_ADDRESS" >> ~/.bashrc'
```

- Verify the configurations by running jet\_real.launch on Jet

```
source ~/.bashrc; source catkin_ws/devel/setup.sh; roslaunch jet_bringup  
jet_real.launch
```

- On the jethost, make sure you can see the topics

```
rostopic list should return all topics
```

```
rostopic echo arduino/encoder_left_value should show the left encoder's data
```