**Install ROS Melodic**

# enable all Ubuntu packages:

$ sudo apt-add-repository universe

$ sudo apt-add-repository multiverse

$ sudo apt-add-repository restricted

# add ROS repository to apt sources

$ 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://keyserver.ubuntu.com:80' --recv-key C1CF6E31E6BADE8868B172B4F42ED6FBAB17C654

# install ROS Base

$ sudo apt-get update

$ sudo apt-get install ros-melodic-ros-base

# add ROS paths to environment

sudo sh -c 'echo "source /opt/ros/melodic/setup.bash" >> ~/.bashrc'

Close and restart the terminal.

**Install Adafruit Libraries**

These Python libraries from Adafruit support the TB6612/PCA9685 motor drivers and the SSD1306 debug OLED:

# pip should be installed

$ sudo apt-get install python-pip

# install Adafruit libraries

$ pip install Adafruit-MotorHAT

$ pip install Adafruit-SSD1306

Grant your user access to the i2c bus:

$ sudo usermod -aG i2c $USER

Reboot the system for the changes to take effect.

**Create catkin workspace**

Create a ROS Catkin workspace to contain our ROS packages:

# create the catkin workspace

$ mkdir -p ~/workspace/catkin\_ws/src

$ cd ~/workspace/catkin\_ws

$ catkin\_make

# add catkin\_ws path to bashrc

$ sudo sh -c 'echo "source ~/workspace/catkin\_ws/devel/setup.bash" >> ~/.bashrc'

Note: out of personal preference, my catkin\_ws is created as a subdirectory under ~/workspace

Close and open a new terminal window. Verify that your catkin\_ws is visible to ROS:

$ echo $ROS\_PACKAGE\_PATH

/home/nvidia/workspace/catkin\_ws/src:/opt/ros/melodic/share

**Build jetson-inference**

Clone and build the [jetson-inference](https://github.com/dusty-nv/jetson-inference) repo:

# git and cmake should be installed

sudo apt-get install git cmake

# clone the repo and submodules

cd ~/workspace

git clone https://github.com/dusty-nv/jetson-inference

cd jetson-inference

git submodule update --init

# build from source

mkdir build

cd build

cmake ../

make

# install libraries

sudo make install

**Build ros\_deep\_learning**

Clone and build the [ros\_deep\_learning](https://github.com/dusty-nv/ros_deep_learning) repo:

# install dependencies

sudo apt-get install ros-melodic-vision-msgs ros-melodic-image-transport ros-melodic-image-publisher

# clone the repo

cd ~/workspace/catkin\_ws/src

git clone https://github.com/dusty-nv/ros\_deep\_learning

# make ros\_deep\_learning

cd ../ # cd ~/workspace/catkin\_ws

catkin\_make

# confirm that the package can be found

$ rospack find ros\_deep\_learning

/home/nvidia/workspace/catkin\_ws/src/ros\_deep\_learning

**Build jetbot\_ros**

Clone and build the [jetbot\_ros](https://github.com/dusty-nv/jetbot_ros) repo:

# clone the repo

$ cd ~/workspace/catkin\_ws/src

$ git clone https://github.com/dusty-nv/jetbot\_ros

# build the package

$ cd ../ # cd ~/workspace/catkin\_ws

$ catkin\_make

# confirm that jetbot\_ros package can be found

$ rospack find jetbot\_ros

/home/nvidia/workspace/catkin\_ws/src/jetbot\_ros

**Testing JetBot**

Next, let's check that the different components of the robot are working under ROS.

First open a new terminal, and start roscore

$ roscore

**Running the Motors**

Open a new terminal, and start the jetbot\_motors node:

$ rosrun jetbot\_ros jetbot\_motors.py

The jetbot\_motors node will listen on the following topics:

* /jetbot\_motors/cmd\_dir relative heading (degree [-180.0, 180.0], speed [-1.0, 1.0])
* /jetbot\_motors/cmd\_raw raw L/R motor commands (speed [-1.0, 1.0], speed [-1.0, 1.0])
* /jetbot\_motors/cmd\_str simple string commands (left/right/forward/backward/stop)

Note: currently only cmd\_str method is implemented.

**Test Motor Commands**

Open a new terminal, and run some test commands:

$ rostopic pub /jetbot\_motors/cmd\_str std\_msgs/String --once "forward"

$ rostopic pub /jetbot\_motors/cmd\_str std\_msgs/String --once "backward"

$ rostopic pub /jetbot\_motors/cmd\_str std\_msgs/String --once "left"

$ rostopic pub /jetbot\_motors/cmd\_str std\_msgs/String --once "right"

$ rostopic pub /jetbot\_motors/cmd\_str std\_msgs/String --once "stop"

(it is recommended to initially test with JetBot up on blocks, wheels not touching the ground)

**Using the Debug OLED**

If you have an SSD1306 debug OLED on your JetBot, you can run the jetbot\_oled node to display system information and user-defined text:

$ rosrun jetbot\_ros jetbot\_oled.py

By default, jetbot\_oled will refresh the display every second with the latest memory usage, disk space, and IP addresses.

The node will also listen on the /jetbot\_oled/user\_text topic to recieve string messages from the user that it will display:

rostopic pub /jetbot\_oled/user\_text std\_msgs/String --once "HELLO!"

**Using the Camera**

To begin streaming the JetBot camera, start the jetbot\_camera node:

$ rosrun jetbot\_ros jetbot\_camera

The video frames will be published to the /jetbot\_camera/raw topic as [sensor\_msgs::Image](http://docs.ros.org/melodic/api/sensor_msgs/html/msg/Image.html) messages with BGR8 encoding. To test the camera feed, install the [image\_view](http://wiki.ros.org/image_view?distro=melodic) package and then subscribe to /jetbot\_camera/raw from a new terminal:

# first open a new terminal

$ sudo apt-get install ros-melodic-image-view

$ rosrun image\_view image\_view image:=/jetbot\_camera/raw

A window should then open displaying the live video from the camera. By default, the window may appear smaller than the video feed. Click on the terminal or maximize button on the window to enlarge the window to show the entire frame.

**JetBot Model for Gazebo Robotics Simulator**

[](https://github.com/dusty-nv/jetbot_ros/raw/master/gazebo/jetbot_gazebo_0.png)