Installation Guide - Step 1 - Install ROS on laptop

We are going to assume you have a laptop with a fresh install of Ubuntu 20.04 and have no previously installed ROS components. After loading 20.04 for the first time, wait for software updater to recognize updates need to be run and then select install

* Install Ubuntu 20 from USB drive
* Make sure you select extra software so wi-fi is enabled
* Change bios to boot from HDD
* After installation, Press RESTART NOW with USB drive still connected (this forced me to do a hard reboot; then I disconnected the USB drive and rebooted) - Ubuntu from HDD then fired up

Update ubuntu with latest updates:

* $ sudo apt update
* $ sudo apt upgrade

Install Chrome.

* Go to<https://www.google.com/intl/en-US/chrome/browser/>
* Download and open with software install; Ubuntu software should load automatically in about 60 seconds. Then click Install. Launch and lock to Launcher.
* Sign in to Google so your URLs are on the toolbar.

Install Chrony as the Network Time Protocol tool [ref](https://pimylifeup.com/using-ntp-on-linux-with-chrony/)

* $ sudo apt install chrony
* $ sudo systemctl enable chrony
* $ sudo systemctl start chrony
* $ sudo systemctl status chrony # check status
* Commands for debugging and checking status are:
  + $ chronyc tracking
  + $chronyc sources -v
  + *$ sudo chronyc -a makestep # to update the time immediately*

To fix a GPG key issue

* $ sudo apt install curl
* $ curl -s https://raw.githubusercontent.com/ros/rosdistro/master/ros.asc | sudo apt-key add -

Update ubuntu with latest updates:

* $ sudo apt update
* $ sudo apt upgrade

| Install VSCode; Open Ubuntu Software, type **Visual Studio Code**,  Install platformio  VSCode will need access to the serial port to display activity  $ sudo chmod 666 /dev/ttyACM0 (I believe there needs to be a board connected to ACM0 for this to work)   * First install updated python $ sudo apt-get install python3-venv    + <https://code.visualstudio.com/docs/?dv=linux64_deb>   The port speed I normally use is 115,200  $ stty -F /dev/ttyAMA0 115200 |
| --- |

| Install VNC Server  sudo dpkg -i /home/al/Downloads/VNC-Server-6.10.0-Linux-x64.deb |
| --- |
|  |

I had an issue updating snap….<https://askubuntu.com/questions/1411104/unable-to-update-snap-store-cannot-refresh-snap-store-snap-snap-store-ha>

$ sudo snap refresh

$ sudo pkill snap-store && sudo snap refresh snap-store

Install Sublime as an editor

* $ wget -qO - https://download.sublimetext.com/sublimehq-pub.gpg | sudo apt-key add -
* $ sudo apt-get install apt-transport-https
* $ echo "deb https://download.sublimetext.com/ apt/stable/" | sudo tee /etc/apt/sources.list.d/sublime-text.list
* $ sudo apt-get update
* $ sudo apt-get install sublime-text
* Ref: <https://www.sublimetext.com/docs/linux_repositories.html>

Install ROS using the instructions at <http://wiki.ros.org/ROS/Installation>

* $ 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 install curl # if you haven't already installed curl
* $ curl -s https://raw.githubusercontent.com/ros/rosdistro/master/ros.asc | sudo apt-key add -
* $ sudo apt update
* $ sudo apt install ros-noetic-desktop-full
* $ source /opt/ros/noetic/setup.bash
* $ printenv | grep ROS

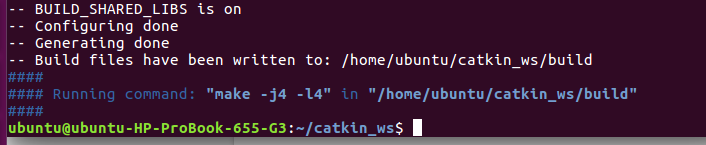
You have now installed ROS. If you want to check to see how the variable names are setup you can display them by entering the following command: $ printenv | grep ROS The output will look similar to this:

| * tractor@tractor-HP-ProBook-645-G1:~$ printenv | grep ROS * ROS\_VERSION=1 * ROS\_PYTHON\_VERSION=3 * ROS\_PACKAGE\_PATH=/opt/ros/noetic/share * ROSLISP\_PACKAGE\_DIRECTORIES= * ROS\_ETC\_DIR=/opt/ros/noetic/etc/ros * ROS\_MASTER\_URI=http://localhost:11311 * ROS\_ROOT=/opt/ros/noetic/share/ros * ROS\_DISTRO=noetic * tractor@tractor-HP-ProBook-645-G1:~$ |
| --- |

## Create a ROS Workspace Using Catkin:

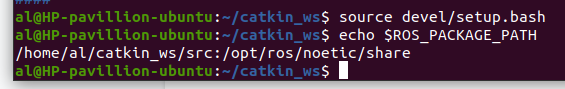
$ mkdir -p ~/catkin\_ws/src  
$ cd ~/catkin\_ws/  
$ catkin\_make

If all goes well you should see this:



$ source devel/setup.bash

$ echo $ROS\_PACKAGE\_PATH #This command is to check To make sure your workspace is properly overlayed by the setup script. The output should look like:

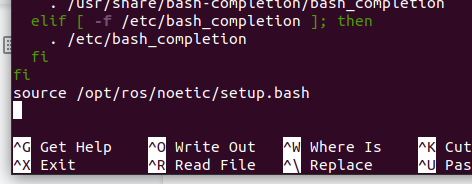


Enter $ roscd # It should to take you to: ubuntu@ubuntu-HP-ProBook-655-G3:~/catkin\_ws/devel$

To avoid having to run $ source ~/catkin\_ws/devel/setup.bash whenever you open a new terminal you have to add the aforementioned line at the very end of your ~/.bashrc file.

Use the nano editor to change .bashrc. Enter $ nano ~/.bashrc

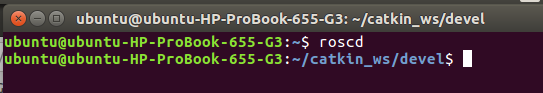
Use the down arrow and go to the bottom of the file. It should look like this:



Add

* >source /home/al/catkin\_ws/devel/setup.bash

Then Ctrl+o; enter; to write the updated file; Ctrl+x to exit; Ctrl+Alt+T to open a **new** window. In that new window enter $ roscd; The results should be:



You can close both windows. You now have ROS installed with the catkin environment setup.

Optional…..

Use Software Installed to install Slack

Update ubuntu with latest updates:

* $ sudo apt update
* $ sudo apt upgrade

Save installation steps $ history > historyrosinstall20220424.txt # to show what actual commands you entered

Optional:

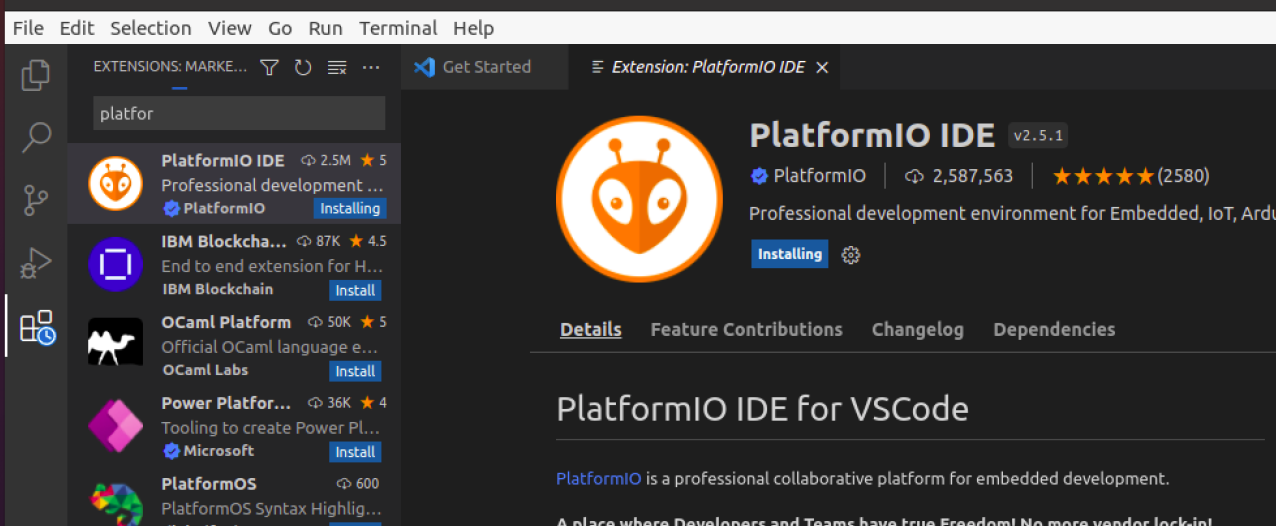
* Install video webcam software - $ sudo apt-get install cheese

MAKE A DISK IMAGE OF THE LAPTOP AT THIS POINT

* Get Clonezilla Live CD
* Boot Laptop to Live CD
* Copy HDD to Portable USB HDD
* <https://clonezilla.org/fine-print-live-doc.php?path=./clonezilla-live/doc/01_Save_disk_image/02-booting.doc#02-booting.doc>

<https://clonezilla.org/clonezilla-live.php>

Optional (i.e. personal choice) steps:

* Install PuTTy $ sudo apt-get install putty
* Install VSCode and PlatformIO
  + First install updated python $ sudo apt-get install python3-venv
  + 
  + <https://code.visualstudio.com/docs/?dv=linux64_deb>
* Update wi-fi power saver feature: (I had to do this to improve wi-fi) Run $ sudo nano /etc/NetworkManager/conf.d/default-wifi-powersave-on.conf; Change: wifi.powersave = 3 To: wifi.powersave = 2; Save the file, which should require root password, close the editor and connect to your wi-fi, then reboot.
* Install Plotjuggler; $ sudo apt install ros-noetic-plotjuggler-roskinetic
* Connect NAS drive ; [ttps://blog.whabash.com/posts/mounting\_synology\_nas\_shared\_folder\_nfs\_ubuntu\_16\_10](https://blog.whabash.com/posts/mounting_synology_nas_shared_folder_nfs_ubuntu_16_10) # not working :(
* Install FileZilla

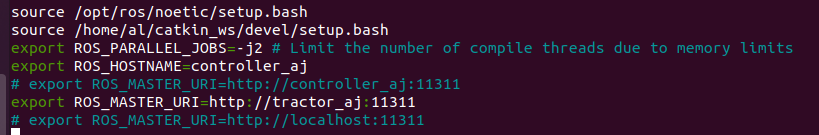
| $ sudo add-apt-repository ppa:n-muench/programs-ppa  $ sudo apt-get update  $ sudo apt-get install filezilla |
| --- |

* Install VNC viewer to access RPi using the GUI on the RPi.
  + Download the file from <https://www.realvnc.com/en/connect/download/viewer/>
  + Use file manager and move the file to the Desktop
  + Open a terminal, go to Desktop, then make file executable with $ chmod +x VNC-Viewer-6.18.625-Linux-x64 # confirm the file name
* Install Jupyter for notebooks
  + First install the latest PIP:
    1. sudo su root
    2. apt-get purge -y python-pip
    3. wget https://bootstrap.pypa.io/get-pip.py
    4. python ./get-pip.py {failed}
    5. apt-get install python-pip
    6. Exit
  + (<https://www.digitalocean.com/community/tutorials/how-to-set-up-a-jupyter-notebook-to-run-ipython-on-ubuntu-16-04>)
  + $ sudo apt-get update
  + $ sudo apt-get -y install python2.7 python-pip python-dev
  + $ sudo apt-get -y install ipython ipython-notebook
  + $ sudo -H pip install jupyter
  + Alternate:
    1. sudo apt install python3-pip
  + $ jupyter notebook # to open Jupyter
* Tkinter for ROS GUI (version for Python 3)
  + $ sudo apt-get install python3-tk
  + $ sudo apt-get install xdotool
* Install Zoom for web conferencing
* Arduino - don’t install from Ubuntu library if you are going to use Teensy boards - see PCJR site for instructions
* <https://tuxboot.org/download/> in order to test clonezilla
* Setup Hosts file
  + $ sudo nano /etc/hosts
  + 192.168.10.149 tractor\_aj
  + > ctrl o, enter; ctrl x

Setup for Tractor / Controller configuration

* Enter $ nano ~/.bashrc

| source /home/al/catkin\_ws/devel/setup.bash  export ROS\_PARALLEL\_JOBS=-j2 # Limit the number of compile threads due to memory limits  export ROS\_HOSTNAME=controller\_aj  # export ROS\_MASTER\_URI=http://controller\_aj:11311  export ROS\_MASTER\_URI=http://tractor\_aj:11311  # export ROS\_MASTER\_URI=http://localhost:11311 |
| --- |

* It should now look like this: {or replace “al” with your userid, “tractor” and “aj” with your preferred identifiers}
* 
* Then Ctrl+o; enter; to write the updated file; Ctrl+x to exit