**Turtlebot 3 Linux Software Commands**

|  |  |
| --- | --- |
| **Ubuntu** | ISO download link: <https://www.ubuntu-fr.org/download/>   * **Software update:** *sudo apt update* * **Software upgrade**: *sudo apt upgrade* * **Install specific package:** *sudo apt-get install package\_name* * **Uninstall specific program:** *sudo apt-get purge program\_name\** |
| **Requirements** | **Install Pyton 3 Colcon and Curl :**  *sudo apt install python3-colcon-common-extensions*  *sudo apt install curl* |
| **ROS 2 Humble** | **Link:** <https://docs.ros.org/en/humble/Installation/Ubuntu-Install-Debians.html>  **Installation:**   * **Add the ROS 2 apt repository to your system:**   *sudo apt install software-properties-common*  *sudo add-apt-repository universe*   * **Add the ROS 2 GPG key with apt:**   *sudo apt update && sudo apt install curl -y*  *sudo curl -sSL https://raw.githubusercontent.com/ros/rosdistro/master/ros.key -o /usr/share/keyrings/ros-archive-keyring.gpg*   * **Add the repository to your sources list:**   *echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/ros-archive-keyring.gpg] http://packages.ros.org/ros2/ubuntu $(. /etc/os-release && echo $UBUNTU\_CODENAME) main" | sudo tee /etc/apt/sources.list.d/ros2.list > /dev/null*   * **Install ROS 2 packages (ROS, Rviz, Demos, Tutorials):**   *sudo apt update*  *sudo apt upgrade*  *sudo apt install ros-humble-desktop*   * **ROS-Base install: Communication libraries, message packages, command line tools (No GUI tools):**   *sudo apt install ros-humble-ros-base*   * **Development tools: Compilers and other tools to build ROS packages**   *sudo apt install ros-dev-tools*   * **Environment setup**   *source /opt/ros/humble/setup.bash*  **Check installation:**   * **In one terminal, source the setup file and then run a C++ talker:**   *source /opt/ros/humble/setup.bash*  *ros2 run demo\_nodes\_cpp talker*   * **In another terminal source the setup file and then run a Python listener:**   *source /opt/ros/humble/setup.bash*  *ros2 run demo\_nodes\_py listener*  **Uninstall ROS 2 Humble:**  *sudo apt remove ~nros-humble-\* && sudo apt autoremove*  *sudo rm /etc/apt/sources.list.d/ros2.list*  *sudo apt update*  *sudo apt autoremove*  *sudo apt upgrade* |
| **Gazebo =** | **Link:** <https://gazebosim.org/docs/garden/install_ubuntu>  <https://installati.one/install-gazebo-ubuntu-22-04/>  **Installation:**   * **Install some necessary tools:**   *sudo apt-get update*  *sudo apt-get install lsb-release wget gnupg*   * **Install Gazebo for Ubuntu 22.04**   *sudo apt install gazebo*  *sudo apt install libgazebo-dev*  **Uninstall Gazebo:**  *sudo apt-get remove gazebo*  *sudo apt-get -y autoremove gazebo*  *sudo apt-get -y purge gazebo*  *sudo apt-get -y autoremove* |
| **Turtlebot3 Humble** | **Link:** <https://emanual.robotis.com/docs/en/platform/turtlebot3/simulation/>  **Installation:**   * **Install Simulation package:**   *sudo apt install ros-humble-gazebo-\**  *sudo apt install ros-humble-cartographer*  *sudo apt install ros-humble-cartographer-ros*  *sudo apt install ros-humble-navigation2*  *sudo apt install ros-humble-nav2-bringup*  *sudo apt remove ros-humble-turtlebot3-msgs*  *sudo apt remove ros-humble-turtlebot3*  *mkdir -p ~/turtlebot3\_ws/src*  *cd ~/turtlebot3\_ws/src/*  *git clone -b humble-devel https://github.com/ROBOTIS-GIT/DynamixelSDK.git*  *git clone -b humble-devel https://github.com/ROBOTIS-GIT/turtlebot3\_msgs.git*  *git clone -b humble-devel https://github.com/ROBOTIS-GIT/turtlebot3.git*  *cd ~/turtlebot3\_ws*  *colcon build*  *echo 'source ~/turtlebot3\_ws/install/setup.bash' >> ~/.bashrc*  *source ~/.bashrc*  *echo 'export TURTLEBOT3\_MODEL=burger' >> ~/.bashrc*  *source ~/.bashrc*  *cd ~/turtlebot3\_ws/src/*  *git clone -b humble-devel https://github.com/ROBOTIS-GIT/turtlebot3\_simulations.git*  *cd ~/turtlebot3\_ws && colcon build*   * **Launch Simulation World** * **Empty World**   *ros2 launch turtlebot3\_gazebo empty\_world.launch.py*   * **Turtlebot3 World**   *ros2 launch turtlebot3\_gazebo turtlebot3\_world.launch.py*   * **Operate Turtlebot3**   *ros2 run turtlebot3\_teleop teleop\_keyboard*   * **Cartographer SLAM**   *ros2 launch turtlebot3\_cartographer cartographer.launch.py*   * **Save Map**   *ros2 run nav2\_map\_server map\_saver\_cli -f ~/map*   * **TurtleSim :** *ros2 run turtlesim turtlesim\_node* * **Teleop keyboard for TurtleSim :** *ros2 run turtlesim turtle\_teleop\_key*   **SSH Connection:**   * **Configure a Wifi hotspot from a computer:**   *nmcli con add type wifi ifname <ip of hotspot wireless interface) con-name <name> autoconnect yes ssid <name SSID> ap ipv4.method shared*  **In our case:** *nmcli con add type wifi ifname wlp2s0 con-name Co4Sys autoconnect yes ssid Co4Sys 802-11-wireless.mode ap ipv4.method shared*  *→ Normaly, the hotspot is configured, you just have to check if the Wifi network is On and in hotspot mode*   * **Connection to the hotspot created before:**   *nmcli con up <name of the wireless network>*  **In our case *:*** *nmcli con up Co4Sys*   * **Configure the SSH settings of the RaspberryPi to connect it to the Wifi hotspot :**   Login: ubuntu  Password: turtlebot   * **Get the Turtlebot3 address:** *ip n (must be reachable)* * **Connect to Turtlebot3:**   *ssh ubuntu@ip\_address*  **In our case:** *ssh* [*ubuntu@10.42.0.216*](mailto:ubuntu@10.42.0.216)   * **Launch the Turtlebot3 and then the keyboard on another terminal:**   *ros2 launch turtlebot3\_bringup robot.launch.py*  *ros2 run turtlebot3\_teleop teleop\_keyboard* |
| **Others commands** | **IP Information:** *ifconfig*  **List of wireless interfaces:** *ip a*  **Wireless scan:** *sudo arp-scan-localnet -I<wireless interface with ip a>*  **Display your own IP Address:** *hostname -I*  **Display all the hotspot connection:** *nmcli connection show* |
| **ROS2 Node** | **Create a ROS2 Workspace and code your own node in Python:**   * **Install colcon extensions:** *sudo apt install python3-colcon-common-extensions* * **Autocompletion:** *gedit ~/.bashrc* **and add the following line at the end:** *source /usr/share/colcon\_argcomplete/hook/* *colcon-argcomplete.bash* * **Make a workspace directory:** *mkdir ros2\_ws* * **Make a source directory inside the workspace:** *cd ros2\_ws && mkdir src* * **Build the workspace:** *colcon build* * **Source the workspace:** *cd && source ~/ros2\_ws/install/setup.bash && gedit ~/.bashrc* **and add in the file the following line at the end :** *source ~/ros2\_ws/install/setup.bash* * **Create a ROS2 package into the workspace:** *cd ros2\_ws/src/ && ros2 pkg create name\_package –build-type ament\_python –dependencies rclpy* * **Make sure Visual Studio Code is installed with Python and ROS2 extensions** * **Check installation:** *sudo snap install code --classic* * **Launch the workspace from the terminal:** *code .* * **Build the package with colcon:** *cd ros2\_ws/ && colcon build* * **Fix “SetuptoolsDeprecationWarning: setup.py install is deprecated” in colcon build:** *sudo apt install python3-pip && pip3 install setuptools==58.2.0* * **Create a python file to write a node and make it executable:** *cd ros2\_ws/src/name\_package/name\_package/touch name\_file\_node.py && chmod +x name\_file\_node.py* * **Execute the python node file created:** *cd ros2\_ws/src/name\_package/name\_package/ && ./name\_file\_node.py*   **Example of a Python node code:**  **→ my\_first\_node.py:**  #!/usr/bin/env python3  **import** rclpy  **from** rclpy**.**node **import** Node  **class** **MyNode(**Node**):**  **def** \_\_init\_\_**(**self**):**  **super().**\_\_init\_\_**(**"first\_node"**)** #Initialization of the Node  self**.**get\_logger**().**info**(**"Hello from ROS2"**)** #Display logger into the terminal  **def** main**(**args**=None):**  rclpy**.**init**(**args**=**args**)** #Initialization ROS2 communication  node **=** MyNode**()** #Create a node from the MyNode class  rclpy**.**spin**(**node**)** #Keep alive the node  rclpy**.**shutdown**()** #Close the ROS2 communication  **if** \_\_name\_\_ **==** '\_\_main\_\_'**:**  main**()**  **→ setup.py**  entry\_points**={**  'console\_scripts'**:** **[**  "test\_node = my\_robot\_controller.my\_first\_node:main"  **],**  **},**   * **Auto-compilation without colcon build every time:**   *cd ros2\_ws/*  *colcon build –symlink-install*  *source ~/.bashrc*  *ros2 run my\_robot\_controller test\_node*   * **Useful commands:**   *ros2 node list*  *ros2 node info /name\_of\_the\_node*  *ros2 topic list*  *ros2 topic info /name\_of\_the\_topic*  *rqt*  *rqt\_graph*  *ros2 interface show <type of ros2 topic info >* |