1. Install VirtualBox 7.0
2. Install Ubuntu 20.04 on VirtualBox by following this video

<https://www.youtube.com/watch?v=x5MhydijWmc&t=1s>

1. Check internet connection on Ubuntu
2. Install Telegram for easily transferring files
3. Install ROS 2 Foxy on Ubuntu (Gazebo11 Simulator is automatically installed)

<https://docs.ros.org/en/foxy/Installation/Ubuntu-Install-Debians.html>

or use the commands below

-------------------------------------------------------------------------------------------------------------------

$ cd

$ sudo apt update && sudo apt upgrade

$ sudo apt install locales

$ sudo locale-gen en\_US en\_US.UTF-8

$ sudo update-locale LC\_ALL=en\_US.UTF-8 LANG=en\_US.UTF-8

$ export LANG=en\_US.UTF-8

$ sudo apt install software-properties-common

$ sudo add-apt-repository universe

$ sudo apt install curl gnupg2 lsb-release

$ sudo curl -sSL https://raw.githubusercontent.com/ros/rosdistro/master/ros.key -o /usr/share/keyrings/ros-archive-keyring.gpg

$ echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/ros-archive-keyring.gpg] http://packages.ros.org/ros2/ubuntu $(source /etc/os-release && echo $UBUNTU\_CODENAME) main" | sudo tee /etc/apt/sources.list.d/ros2.list > /dev/null

$ sudo apt install ros-foxy-desktop python3-argcomplete

$ sudo apt install ros-dev-tools

$ source /opt/ros/foxy/setup.bash

$ echo "source /opt/ros/foxy/setup.bash" >> ~/.bashrc

$ sudo apt install python3-colcon-common-extensions

-------------------------------------------------------------------------------------------------------------------

- Try some examples

-------------------------------------------------------------------------------------------------------------------

$ ros2 run demo\_nodes\_cpp talker

$ ros2 run demo\_nodes\_py listener

-------------------------------------------------------------------------------------------------------------------

1. Learn to create new packages

- Close all terminals and then open a new one.

-------------------------------------------------------------------------------------------------------------------

$ cd

$ mkdir -p ~/dev\_ws/src/

$ cd ~/dev\_ws/src/

$ ros2 pkg create --build-type ament\_cmake --node-name my\_node my\_package\_cpp

$ ros2 pkg create --build-type ament\_python --node-name my\_node my\_package\_py

$ cd ..

$ colcon build

After creating and building the package, make setup.bash file to be automatically sourced for any new opened terminal.

-------------------------------------------------------------------------------------------------------------------

$ cd

$ sudo apt-get install gedit

$ gedit ~/.bashrc

-------------------------------------------------------------------------------------------------------------------

The .bashrc file will appear on text editor. Then you scroll to bottom and type

source ~/dev\_ws/install/setup.bash

Then, try to run the new nodes

-------------------------------------------------------------------------------------------------------------------

$ ros2 run my\_package\_cpp my\_node

$ ros2 run my\_package\_py my\_node

-------------------------------------------------------------------------------------------------------------------

- Check whether ROS2 workspace has been created or not by using commands

cd and ls. If not, create a new workspace named ros2\_ws by the following commands

-------------------------------------------------------------------------------------------------------------------

$ mkdir -p ~/ros2\_ws/src

$ cd ~/ros2\_ws

$ colcon build

$ echo “source ~/ros2\_ws/install/setup.bash” >> ~/.bashrc

$ source ~/.bashrc

-------------------------------------------------------------------------------------------------------------------

- Create package name two\_wheeled\_robot by following the instruction in this link

<https://automaticaddison.com/how-to-load-a-urdf-file-into-rviz-ros-2/>

1. Install packages of Gazebo and interface between ROS2 and Gazebo Simulator

(<http://classic.gazebosim.org/tutorials?tut=ros2_installing&cat=connect_ros> )

-------------------------------------------------------------------------------------------------------------------

$ sudo apt install gazebo11 ros-foxy-gazebo-ros-pkgs

$ sudo apt install ros-foxy-cartographer ros-foxy-cartographer-ros

$ sudo apt install ros-foxy-navigation2 ros-foxy-nav2-bringup

-------------------------------------------------------------------------------------------------------------------

* Make sure you have some core tools installed

-------------------------------------------------------------------------------------------------------------------

$ sudo apt install ros-foxy-ros-core ros-foxy-geometry2

-------------------------------------------------------------------------------------------------------------------

* Testing gazebo by loading the differential drive world with Gazebo

(i- Close the terminal and open it again. ii- Split the window into 2 horizontal windows, then run the command below on top window terminal

-------------------------------------------------------------------------------------------------------------------

$ gazebo --verbose /opt/ros/foxy/share/gazebo\_plugins/worlds/gazebo\_ros\_diff\_drive\_demo.world

-------------------------------------------------------------------------------------------------------------------

iii- Run the publish message below to move the robot in the bottom terminal window

-------------------------------------------------------------------------------------------------------------------

ros2 topic pub /demo/cmd\_demo geometry\_msgs/Twist '{linear: {x: 1.0}}' -1

-------------------------------------------------------------------------------------------------------------------

* Open a gazebo world (it takes a minute to open the world in gazebo)

-------------------------------------------------------------------------------------------------------------------

$ cd /usr/share/gazebo-11

$ gazebo worlds/cafe.world

-------------------------------------------------------------------------------------------------------------------

* Copy world and object files to our customized worlds and models

-------------------------------------------------------------------------------------------------------------------

$ cd /usr/share/gazebo-11/worlds

$ cp cafe.world ~/dev\_ws/src/two\_wheeled\_robot/worlds

$ cd ~/dev\_ws/src/two\_wheeled\_robot/worlds

$ ls

$ cd /usr/share/gazebo-11/models

$ ls

$ cp -r ground\_plane ~/dev\_ws/src/two\_wheeled\_robot/models

$ cd ~/.gazebo/models

$ ls

$ cp -r cafe ~/dev\_ws/src/two\_wheeled\_robot/models

$ cp -r cafe\_table ~/dev\_ws/src/two\_wheeled\_robot/models

-------------------------------------------------------------------------------------------------------------------

- Create launch file to load cafe.world into gazebo by following the instruction in this link

<https://automaticaddison.com/how-to-load-a-world-file-into-gazebo-ros-2/>

or follow the following codes

-------------------------------------------------------------------------------------------------------------------

$ cd ~/dev\_ws/src/two\_wheeled\_robot

$ cd launch

gedit load\_world\_into\_gazebo.launch.py

-------------------------------------------------------------------------------------------------------------------

The launch file will appear in editor, then copy the code found in the link above and past in the launch file. Save and close it. Then run the code below

-------------------------------------------------------------------------------------------------------------------

$ cd ~/dev\_ws

$ colcon build

$ ros2 launch two\_wheeled\_robot load\_world\_into\_gazebo.launch.py

-------------------------------------------------------------------------------------------------------------------

1. Install packages of TurTleBot3

<https://emanual.robotis.com/docs/en/platform/turtlebot3/quick-start/>

or follow the below commands.

* Install TurTleBot3 Packages

-------------------------------------------------------------------------------------------------------------------

$ sudo apt remove ros-foxy-turtlebot3-msgs

$ sudo apt remove ros-foxy-turtlebot3 // remove them to avoid redundancy

$ mkdir -p ~/turtlebot3\_ws/src

$ cd ~/turtlebot3\_ws/src/

$ git clone -b foxy-devel https://github.com/ROBOTIS-GIT/DynamixelSDK.git

$ git clone -b foxy-devel https://github.com/ROBOTIS-GIT/turtlebot3\_msgs.git

$ git clone -b foxy-devel https://github.com/ROBOTIS-GIT/turtlebot3.git

$ cd ~/turtlebot3\_ws

$ colcon build --symlink-install

$ echo 'source ~/turtlebot3\_ws/install/setup.bash' >> ~/.bashrc

$ source ~/.bashrc

$ sudo apt install ros-foxy-turtlebot3-gazebo

echo 'export GAZEBO\_MODEL\_PATH=$GAZEBO\_MODEL\_PATH:~/turtlebot3\_ws/src/turtlebot3/turtlebot3\_simulations/turtlebot3\_gazebo/models' >> ~/.bashrc

echo 'export TURTLEBOT3\_MODEL=waffle\_pi' >> ~/.bashrc

source ~/.bashrc

ros2 launch turtlebot3\_gazebo turtlebot3\_house.launch.py

-------------------------------------------------------------------------------------------------------------------

1. Set the ROS2 environment for communicating with MATLAB, and launch turtlebot3 in Gazebo11.

- Set up ROS2 domain ID in the Virtual Machine and install ssh server for communication between Window and the Virtual Machine

---------------------------------------------------------------------------------------------------------------

$ echo 'export ROS\_DOMAIN\_ID=25 #TURTLEBOT3' >> ~/.bashrc

$ source ~/.bashrc

$ sudo apt update

$ sudo apt upgrade

$ sudo apt install openssh-server

---------------------------------------------------------------------------------------------------------------

- Launch turtlebot3 in Gazebo11 using the following commands

---------------------------------------------------------------------------------------------------------------

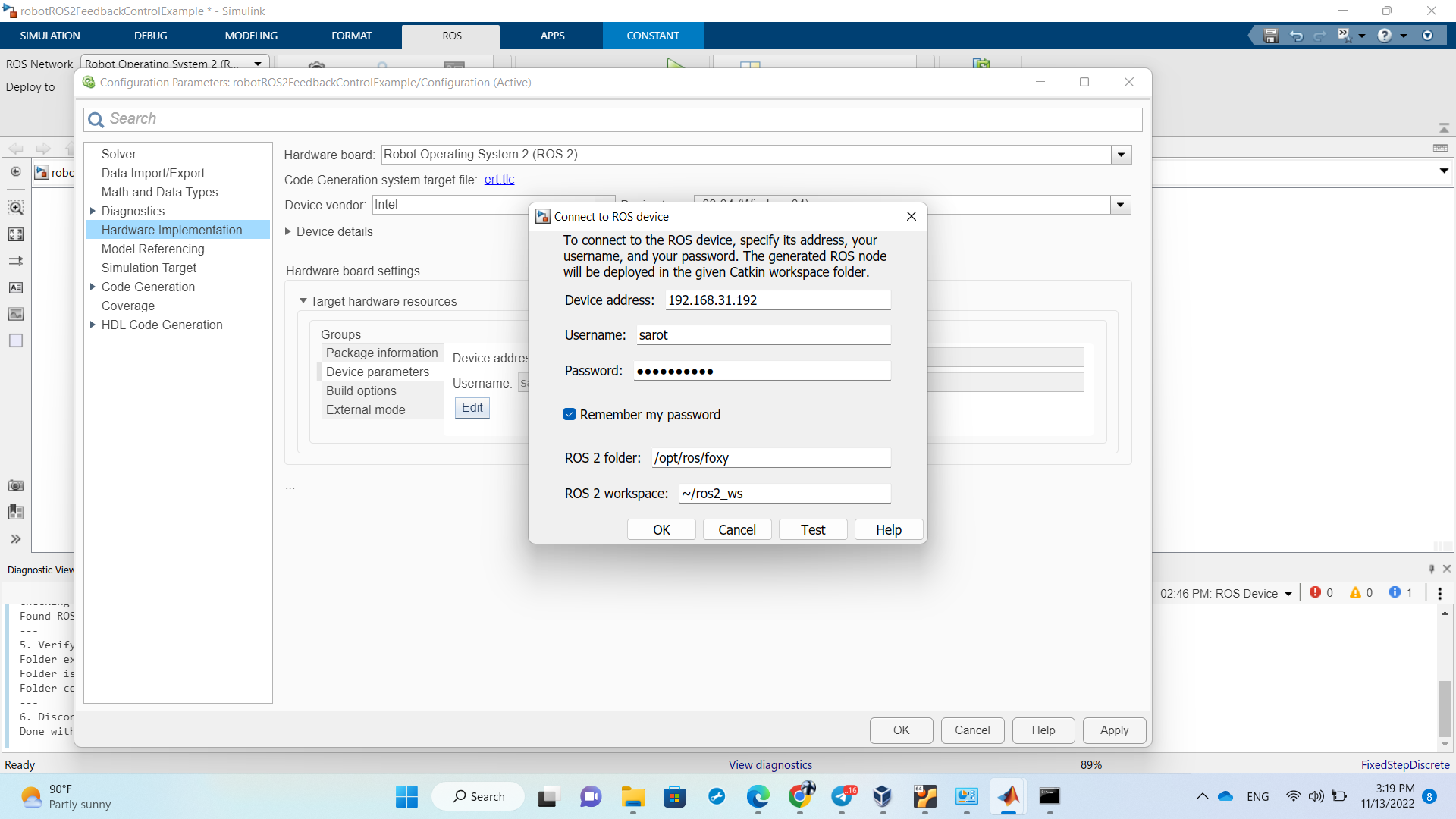
$ ros2 launch turtlebot3\_gazebo empty\_world.launch.py

---------------------------------------------------------------------------------------------------------------

1. Test an example of Feedback Control of a ROS-Enabled Robot Over ROS 2from MATLAB.

- Open MATLAB Help, then in the search area type *Feedback Control of a ROS-Enabled Robot Over ROS 2*. Choose the first result from search. Then click on “open live scripts”. The MATLAB live script editor will appear. Then run section by section to setup the ROS2 domain ID in MATLAB, and it will also open the Simulink model for controlling the turtlebot3 in Gazebo11.

- Setup communication via IP address. To find the IP address of the the Virtual Machine, open a new terminal and then type ifconfig. In the Simulink model, goto ROS tab, then on Deploy to, choose Remote Device. Then click on Hardware Settings, and configure parameter as shown in the figure blow.

- After configuring all the parameter, click OK on Connect to ROS device window, then click on OK on the Configuration Parameters of the Simulink model. Then test the connection under the Remote Device. Check all the block ROS2 message to make sure the right message are configured correctly, and check the Bus signal to make sure the the right bus signals are selected.

- Finally, you can run the Simulink model to control the turtlebot3 in Gazebo11 by clicking on Monitor & Tune icon on the Simulink tab. To change the destination of the robot, double click on the blue constant block and change the coordinate x and y to e.g. [5 8].