ROS and experimental robotics.
How-to operate the Turtlebot 3 burger.

Working with a real robot is not different from listening and subscribing to topics, using services, etc. , except that the node you are working with is currently running on a distance machine inside the robot.

To allow the ROS nodes running on the virtual machine to communicate with the nodes running on the robot, the connections between the two must be first set up. To do that, ROS uses shell variables that are declared in the file `.bashrc` at the root of the `turtle` (virtual machine) and `ubuntu` (robot) user's home directory. Both must be modified as follows.

1. Both the virtual machine and the robot must be on the same network. The robot automatically creates its own wifi hotspot (the name of this hotspot is `turblebot_n` where n is the number of your turtlebot). This hotspot is accessible with the password `@turtlebot_wifi@`. Connect your host machine (i.e. the computer on which is running the virtualization platform) to one of these wifi network (the one created by your robot).

2. It is then necessary to modify the network configuration of the virtual machine, by switching from a "NAT shared" to a "Bridged" network (see Figure 1), so that the virtual machine is on the same network as the turtlebot.

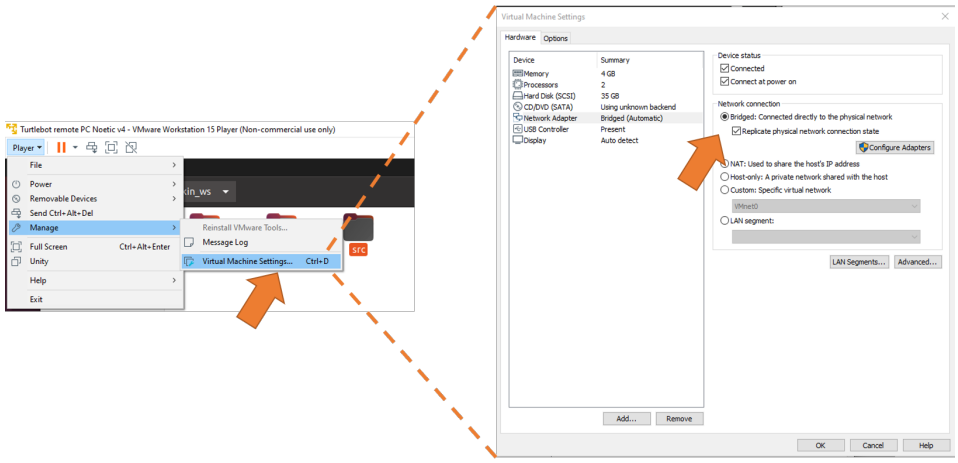

Figure 1: Ubuntu window with the network connection details.

3. The robot should now get its IP address from the robot. To check it is the case, click on the arrow on the top right of the menu bar in Ubuntu, and click on the arrow next to `wired Connected` and then on `Wired Settings`. A window then appears, like in Figure 2. From there, note the IP address in the IPv4 section.

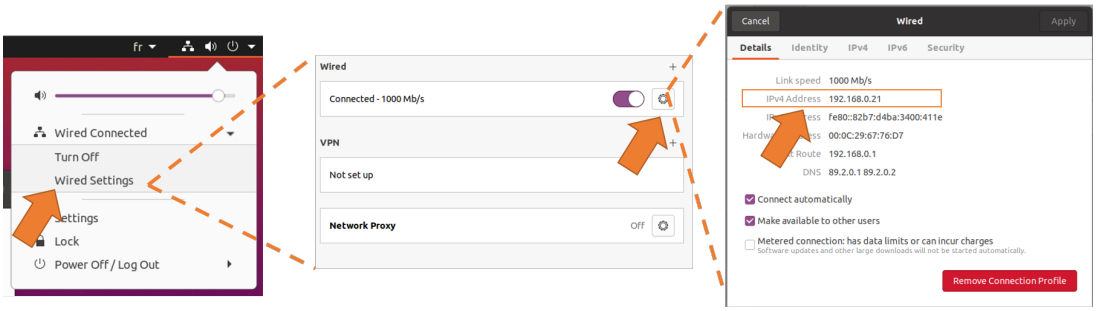

Figure 2: Ubuntu window with the network connection details.

> You might have to perform this step every time you disconnect and reconnect to the wifi network, since the IP address you get from the robot might change. Only the IP address of each robot is fixed. If the VM IP address does not start with `192.168.0.` then try to reboot the virtual machine and start over.

4. Edit the file ~/.bashrc **in the virtual machine** by using the following command in a terminal

```
$ gedit /home/turtle/.bashrc
```

   A new window appears: scroll to the bottom of the file, and modify the to variables ROS_MASTER_URI and ROS_HOSTNAME to include the IP address you get from the previous step. For instance, if the IP adress is 192.168.0.237, the two lines must be then modified along:

```
export ROS_MASTER_URI=http://192.168.0.237:11311
export ROS_HOSTNAME=192.168.0.237
```

   Do not forget to save your changes!

5. The same must be done on the robot, except you have to remotely connect to it first. This time, getting the IP address of your robot is easy: it is written on a sticker placed near the laser at the top of the robot. For instance, lets consider the robot IP is 192.168.0.113.

   Next, connect to the distant robot **from you virtual machine** by typing the following command in a terminal (if asked, the password is turtlebot):

```
$ ssh ubuntu@192.168.0.113
```

   Then, you are **connected to the robot on a distant shell** in you terminal. From there, edit the .bashrc **on the robot** file using the following command:

```
$ nano /home/ubuntu/.bashrc
```

   A text editor is now opened in you terminal. Scroll down to the bottom of the opened file and modify here the two ROS_MASTER_URI and ROS_HOSTNAME so that the former matches **the virtual machine IP** and the later **the robot IP**. For instance, the two lines must be modified along:

```
export ROS_MASTER_URI=http://192.168.0.237:11311
export ROS_HOSTNAME=192.168.0.113
```

   Do not forget to save your changes by pressing `Ctrl+O` and next `Enter`, and quit the text editor with `Ctrl+X`. Thanks to these modifications, all running ROS nodes will communicate with the master running on your local virtual machine.

6. Now close all your running terminal to be sure your changes to be applied. Then, run the `roscore` command from a terminal on your local machine and log again on the robot using the `ssh` command shown before. On the robot, launch the `turtlebot3_core.launch` file from the `turtlebot3_bringup` packet. Check if the running nodes on the robot are correctly reaching the master on the local machine by checking if the IP address in the output messages matches the IP of the virtual machine.

7. You should now be able to teleoperate the turtlebot! Please call your teacher before actually trying to control the robot. This one must be first secured, and ideally placed above a small box so that both wheels can freely rotate without making the robot actually move.