***Introduction to Robotic Systems Course***

**LAB 5**

**Getting Started with Robot Operating System (ROS)**

Contents

[1 Introduction 1](#_Toc30060715)

[1.1 Lab Overview 1](#_Toc30060716)

[2 Requirements 1](#_Toc30060717)

[3 Virtual Machine, Remote PC, and TurtleBot PC 1](#_Toc30060718)

[4 Task: Install Ubuntu on Remote PC 1](#_Toc30060719)

[4.1 Install Ubuntu on Remote PC 2](#_Toc30060720)

[4.2 Install VM on Remote PC and then Install Ubuntu 2](#_Toc30060721)

[5 Task: Set Up Remote PC 2](#_Toc30060722)

[6 Task: Set Up TurtleBot PC 3](#_Toc30060723)

[6.1 Booting Up the Raspberry Pi 3 (TurtleBot PC) 3](#_Toc30060724)

[6.2 Configure Screen Resolution and Expand Filesystem 4](#_Toc30060725)

[7 Task: Network Configuration of Both PCs 4](#_Toc30060726)

[7.1 Synchronize Computers’ Date and Time 4](#_Toc30060727)

[7.2 Network Configuration Setup on Remote PC 5](#_Toc30060728)

[7.3 Network Configuration Setup on TurtleBot PC 6](#_Toc30060729)

[8 Task: OpenCR Setup 6](#_Toc30060730)

[9 Task: Bringup 6](#_Toc30060731)

[10 Task: Basic Operation 7](#_Toc30060732)

[10.1 Example Commands 7](#_Toc30060733)

[10.1.1 Control using Interactive Markers 7](#_Toc30060734)

[10.1.2 Obstacle Detection 8](#_Toc30060735)

[10.1.3 Move by Coordinates 8](#_Toc30060736)

[10.1.4 Patrol 8](#_Toc30060737)

[10.1.5 Keyboard Control 8](#_Toc30060738)

[10.1.6 Topic Monitor 8](#_Toc30060739)

# Introduction

## Lab Overview

In this lab, you will set up ROS on Ubuntu OS and set up the robot onboard PC (Raspberry Pi 3) with a ROS distribution called Kinetic Kame.

# Requirements

This lab requires the following hardware and software:

* **Hardware:**
  + TurtleBot 3 Burger
  + Four TCRT5000 IR reflector sensors (shown in Figure 1)
  + MicroSD Card
* **Software:** Virtual Machine (needed to install Ubuntu).

# Virtual Machine, Remote PC, and TurtleBot PC

If you have a different OS other than Ubuntu, you can use a Virtual Machine (VM) software. In our lab, we used VirtualBox.

In this lab document, your desktop computer is referred to as Remote PC while we refer to the Raspberry Pi 3 on the robot as TurtleBot PC.

# Task: Install Ubuntu on Remote PC

The remote PC (your computer) should run Ubuntu OS. This is because some program and environment needed to program and operate the robot is best done in Ubuntu OS.

Ubuntu can be installed on a computer; however, you must understand what you are doing (especially if you already have another OS installed) as the hard drive can be accidentally erased. Another option is to install a VM on which Ubuntu can then be installed. Both approaches are detailed in the next two subsections.

The first step is to download Ubuntu from <http://releases.ubuntu.com/16.04/>. The lab was tested with Ubuntu 16.04 version.

## Install Ubuntu on Remote PC

To install Ubuntu on your computer, transfer the downloaded image file to a CD or USB flash drive and follow the steps detailed in the link: <https://tutorials.ubuntu.com/tutorial/tutorial-install-ubuntu-desktop#1>.

## Install VM on Remote PC and then Install Ubuntu

If you prefer to use a VM instead, follow this link to download VirtualBox <https://www.virtualbox.org/wiki/Downloads>. After installing the VM, follow the guide on Part 4 of the getting started guide to configure your VM and start using Ubuntu on it.

# Task: Set Up Remote PC

In this task, you will install ROS and its dependent packages on the remote PC.

Open a terminal on your Remote PC and run the commands shown below to begin the installation of ROS. The ROS version is Kinetic Kame. This version is compatible with Ubuntu Xenial Xerus and supported for TurtleBot3.

1. $ sudo apt-get update
2. $ sudo apt-get upgrade

Type ‘y’ and press enter to the question Do you want to continue?

1. $ wget https://raw.githubusercontent.com/ROBOTIS-GIT/robotis\_tools/master/install\_ros\_kinetic.sh && chmod 755 ./install\_ros\_kinetic.sh && bash ./install\_ros\_kinetic.sh

Press enter to continue the installation when asked

After running the commands above, restart your Remote PC.

We also need to install some dependent packages that will allow us to monitor and control the TurtleBot on the Remote PC. Run the following commands on a terminal on the Remote PC. It may take some time before the commands finish executing.

1. $ sudo apt-get install ros-kinetic-joy ros-kinetic-teleop-twist-joy ros-kinetic-teleop-twist-keyboard ros-kinetic-laser-proc ros-kinetic-rgbd-launch ros-kinetic-depthimage-to-laserscan ros-kinetic-rosserial-arduino ros-kinetic-rosserial-python ros-kinetic-rosserial-server ros-kinetic-rosserial-client ros-kinetic-rosserial-msgs ros-kinetic-amcl ros-kinetic-compressed-image-transport ros-kinetic-rqt-image-view ros-kinetic-gmapping ros-kinetic-navigation ros-kinetic-interactive-markers
2. $ cd ~/catkin\_ws/src/
3. $ git clone https://github.com/ROBOTIS-GIT/turtlebot3\_msgs.git
4. $ git clone https://github.com/ROBOTIS-GIT/turtlebot3.git

When done, close the terminal window, open a new one, and run the command below to build the packages in the source space we just copied.

$ cd ~/catkin\_ws && catkin\_make

Restart the Remote PC to complete the installation.

# Task: Set Up TurtleBot PC

The TurtleBot PC (Raspberry Pi 3) will run a version of the Raspbian OS that contains pre-installed ROS, ROS packages, and the minimum packages required to operate the robot. Flash the image of the Raspbian OS to a microSD Card, which should be a minimum of 8 GB.

Use your PC to flash the microSD with the Raspbian OS image following the steps below.

* Download Raspbian OS from <http://www.robotis.com/service/download.php?no=730>.
* Download and install Etcher using this link: <https://www.balena.io/etcher/>.
* Use Etcher to flash the microSD card with the image of the Raspbian OS.

## Booting Up the Raspberry Pi 3 (TurtleBot PC)

After flashing the microSD Card with the Raspbian OS image, insert the microSD Card into the Raspberry Pi 3 and boot it up. For this, connect the Raspberry Pi 3 to a monitor with the HDMI port.

Ensure the TurtleBot PC is connected to same network as your Remote PC.

If you plan to operate the TurtleBot PC from the Remote PC by sshing into it, then it is best to copy the IP address of the TurtleBot PC at this point. To do this, open a terminal window and run the following command.

$ ifconfig

Make a note of the IP address. See Figure 1 for where to look for the IP address, highlighted by the red box.

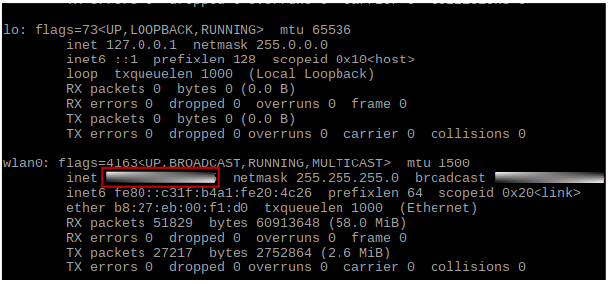


Figure 1: ifconfig Window for TurtleBot PC.

You can access the TurtleBot PC from a terminal on the Remote PC using ssh. To do this, open a terminal on the Remote PC and enter the following command.

$ ssh pi@*< insert TurtleBot PC ip address here>*

Enter **turtlebot** if asked for a password.

When you have logged in, you can run all the commands for the TurtleBot PC in this lab from here.

## Configure Screen Resolution and Expand Filesystem

On your Remote PC, ssh into the TurtleBot PC and enter the following command to configure the resolution for your display and expand the file system.

$ sudo raspi-config

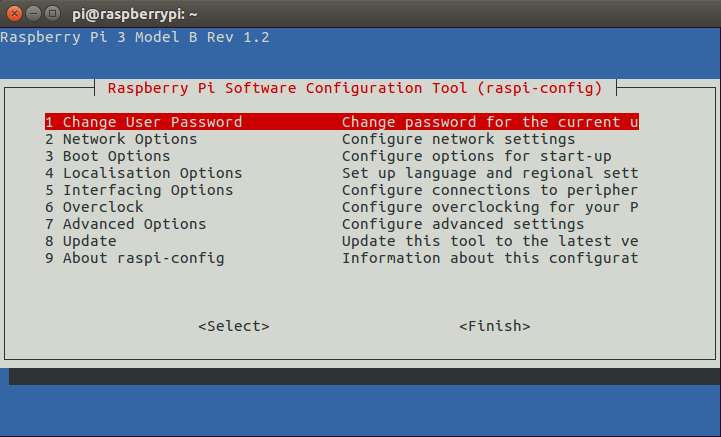


Figure 2: Configuration Tool Window.

On the open Configuration Tool window (see figure 2):

* Go to **Advanced Options -> Resolution** and choose the correct screen resolution.
* Next go to **Advanced Options -> Expand Filesystem** to use the extra space of a large capacity microSD card.
* Click **Finish** and reboot the system.

# Task: Network Configuration of Both PCs

To allow communication between the TurtleBot PC and Remote PC and enable the use of the ROS installations in both PCs, the IP addresses of both TurtleBot PC and Remote PC will be set up next. Ensure that both TurtleBot PC and Remote PC are on the same network.

## Synchronize Computers’ Date and Time

We will now synchronize the time on the TurtleBot PC with that of the Remote PC.

Open a new terminal on the Remote PC and *ssh* into the TurtleBot PC and then enter the following command to synchronize the computers’ date and time.

1. $ sudo apt-get install ntpdate
2. $ sudo ntpdate ntp.ubuntu.com

## Network Configuration Setup on Remote PC

* First, get the IP address of the Remote PC by running the ifconfig command. The area you can find it is highlighted in Figure 3.

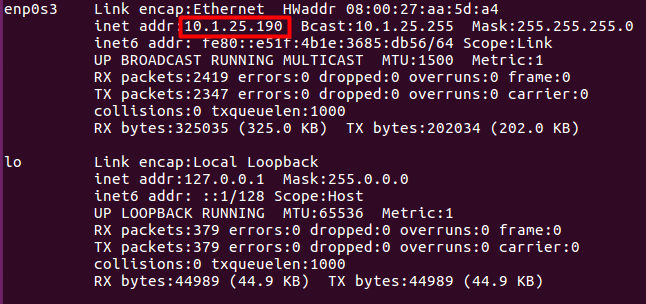


Figure 3: ifconfig Window for Remote PC.

* Next, enter the command below and scroll all the way to the bottom where you will find localhost as the default IP addresses.

$ nano ~/.bashrc

* Change localhost to the IP address of the Remote PC. See highlighted area in Figure 4.
* Add two export lines as shown below to include the model of TurtleBot3 and the port name. See highlighted area in Figure 4.

export TURTLEBOT3\_MODEL=burger

export OPENCR\_PORT=/dev/ttyACM0

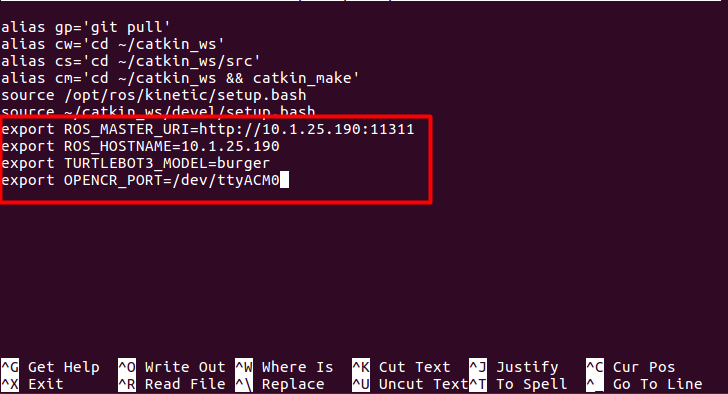


Figure 4: Editing bashrc file.

* Use **Ctrl + X** to exit. Press **Enter** when asked if you want to save modified buffer.

You will now be returned to the terminal window. Run the command below to source the file you have just changed.

$ source ~/.bashrc

## Network Configuration Setup on TurtleBot PC

To set up network configuration for the TurtleBot PC, repeat the same steps above. However, note the following:

* For **ROS\_Controller\_URI**, replace localhost with the IP of Remote PC.
* For **ROS\_HOSTNAME**, replace localhost with the IP of TurtleBot PC.

# Task: OpenCR Setup

The OpenCR firmware will be loaded into the board before we can start using it. To do this, open a terminal on the TurtleBot PC or ssh into it from the remote pc and run the following commands.

1. $ export OPENCR\_PORT=/dev/ttyACM0
2. $ export OPENCR\_MODEL=burger
3. $ rm -rf ./opencr\_update.tar.bz2
4. $ wget https://github.com/ROBOTIS-GIT/OpenCR/raw/master/arduino/opencr\_release/shell\_update/opencr\_update.tar.bz2 && tar -xvf opencr\_update.tar.bz2 && cd ./opencr\_update && ./update.sh $OPENCR\_PORT $OPENCR\_MODEL.opencr && cd ..

Once OpenCR setup is complete, the text **jump\_to\_fw** will be displayed on the terminal window.

# Task: Bringup

In this section, the Light Detection and Ranging (LiDAR ) sensor on the robot will be used to scan the environment. This result of the scan will be visualized using RViz, which is a graphic user interface (GUI) that provides 3D visualization of sensor data and robot states.

* On Remote PC, run **roscore command**.

$ roscore

* Then, on TurtleBot PC (ssh into it from a terminal in Remote PC) launch **bringup** to get the basic TurtleBot3 applications started.

$ roslaunch turtlebot3\_bringup turtlebot3\_robot.launch

Ensure that the process completes without errors before proceeding to launch robot state publisher and run RViz.

* On the Remote PC, run the following commands each on a separate terminal. This will open the Rviz window as shown in Figure 5.

$ roslaunch turtlebot3\_bringup turtlebot3\_remote.launch

­

$ rosrun rviz rviz -d `rospack find turtlebot3\_description`/rviz/model.rviz

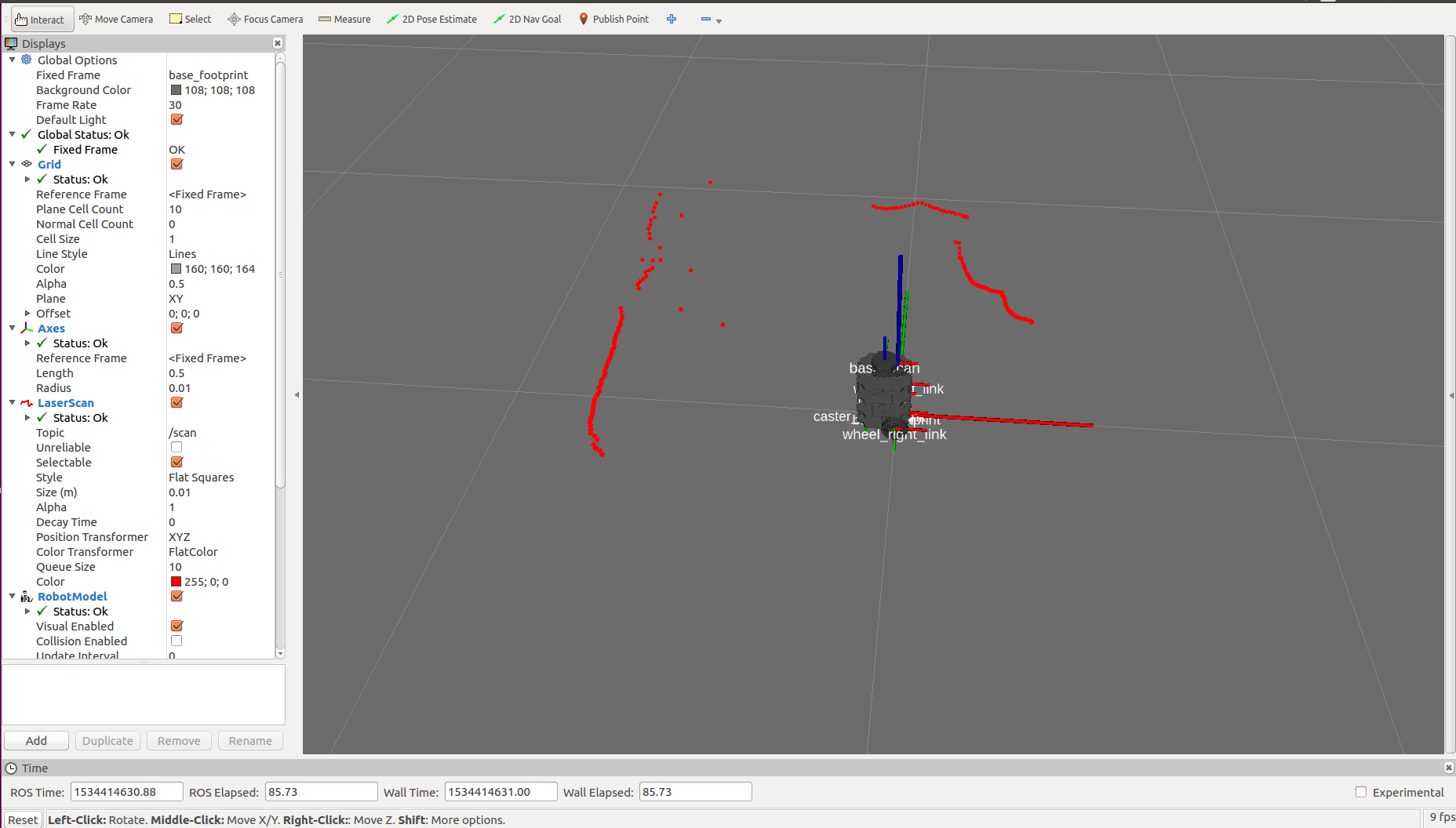


Figure 5: RViz window showing Turtle Mapping output—point clouds.

The red dots shown in Figure 5 are called point clouds. They indicate the objects detected around robot.

# Task: Basic Operation

After successfully following the steps above, the robot should be ready to perform some basic operations like movements with direction control (path to patrol) using a keyboard or the arrows in RViz. The path to patrol can be rectangular, triangular, or circular. Object detection might also come in handy in many applications.

## Example Commands

Before running the example commands, ensure the following are up and running:

* Run **roscore** command on the Remote PC
* Run **Bringup** command on TurtleBot PC.

### Control using Interactive Markers

You can use the markers displayed on screen to move the robot in any direction. To do this, run the following commands on the Remote PC, each on a separate terminal.

$ roslaunch turtlebot3\_example interactive\_markers.launch

$ rosrun rviz rviz -d `rospack find turtlebot3\_example`/rviz/turtlebot3\_interactive.rviz

### Obstacle Detection

This operation uses data from the LiDAR sensor to determine the distance of an object. The robot will stop moving when the distance is below a set minimum. To do this, run the following command on the Remote PC terminal.

$ roslaunch turtlebot3\_example turtlebot3\_obstacle.launch

### Move by Coordinates

This operation allows you to insert coordinates such as (0.3, 0.5, 90) to make the TurtleBot3 move 0.3 m in x-axis, 0.5 m in y-axis, and rotate 90 degrees. To do this, run the following command on the Remote PC.

$ roslaunch turtlebot3\_example turtlebot3\_pointop\_key.launch

### Patrol

There are three patrol routes you can choose (rectangle, triangle, and circle). First line of command launches the patrol server, which translates the patrol data received from action client to control the velocity of TurtleBot3. To operate the TurtleBot in patrol mode, run the following commands on the Remote PC, each on a separate terminal.

$ rosrun turtlebot3\_example turtlebot3\_server

$ roslaunch turtlebot3\_example turtlebot3\_client.launch

### Keyboard Control

Use the keyboard buttons to control the movement of the robot by following the instructions on terminal window. To do this, run the following command on the Remote PC terminal.

$ roslaunch turtlebot3\_teleop turtlebot3\_teleop\_key.launch

### Topic Monitor

The rqt allows us to check for topic status of the TurtleBot3 such as sensor status, battery status, diagnostics data, and many more. To do this, run the following command on the Remote PC terminal.

$ rqt

On the rqt window, go to **Plugins -> Topics -> Topic** **Monitor** and then check the boxes you want to monitor. Expand each topic checked to see the status of the robot.