ELEC-E8111 PROJECT WORK 2024

AUTONOMOUS MOBILE ROBOTS

ROBOT PROJECT INSTRUCTION

This project work has been divided into four parts and you will need to finish all of them and write a report. In the first chapter, you will learn how to use the Turtlebot4 platform and drive the robot. In the second chapter, you will work in a real-world robot environment and perform mapping. In the third chapter, you will record a dataset and perform mapping based on it. In the last part, you will use the ROS Navigation stack, to implement and run path planning and navigation.

General Notes:

- Ctrl+Shift+T opens a new tab in the same terminal window.
- Ctrl+Shift+N opens a new terminal window.
- Ctrl+C stops the currently running script in a terminal tab.
- Tab completion can be very useful.
- Source your ROS 2 environment before trying to launch a node.

Real-world engineering problems can have many uncertainties, which sometimes lead to failures in localization as well as navigation. Therefore, if you end up with inaccurate results or failures, that's normal, and it doesn't affect your final score. We are more interested in the process of completing the project and analyzing the results, which is why we designed this project in the first place. In addition, if your project is done exceptionally well, we will consider giving you extra points.

1. Driving your TurtleBot 4

Begin by powering on the TurtleBot 4 and connecting it to a local network. Position the TurtleBot 4 onto its designated dock. Upon placement, observe the illumination of the green LED on the dock, which should remain lit for a brief period indicating successful power activation of the TurtleBot 4. Allow the robot sufficient time to complete its boot-up process. If the LED transitions to red, this indicates low battery; kindly await full recharging before proceeding further.

Ensure your laptop is adequately charged and then power it on. The passcode to access the laptop should be entered as "turtlebot4". Following successful boot-up, both the robot and laptop will seamlessly connect to the local network of the router.

Due to the lack of performance of the laptop, it is recommended to use your own PC to connect the robot and establish a wired connection between the PC and the router using a cable. The suggested software environment of the PC is Ubuntu 22.04 LTS, ROS2 Humble. Please refer to the following websites to install them:

https://ubuntu.com/download/desktop#how-to-install
https://docs.ros.org/en/humble/Installation/Ubuntu-Install-Debians.html

Alternatively, you can also use your own router, ensure both the robot and all devices are operating within the same local network.

You also need to install *turtlebot4* package on your own PC:

```
$ sudo apt install ros-humble-turtlebot4-desktop
$ sudo apt install ros-humble-turtlebot4-description \
ros-humble-turtlebot4-msgs \
ros-humble-turtlebot4-navigation \
ros-humble-turtlebot4-node
```

Create a file called *setup.bash:*

```
$ sudo mkdir /etc/turtlebot4/
$ sudo touch /etc/turtlebot4/setup.bash
```

Add the following lines to *setup.bash:*

```
$ source /opt/ros/humble/setup.bash
$ export ROS_DOMAIN_ID=0
$ export RMW_IMPLEMENTATION=rmw_fastrtps_cpp
```

Finally, add the following line in ~/.bashrc to apply the settings to every new terminal:

```
$ source /etc/turtlebot4/setup.bash
```

You can check the connection via:

```
$ ros2 topic echo /ip
data: 192.168.0.101
```

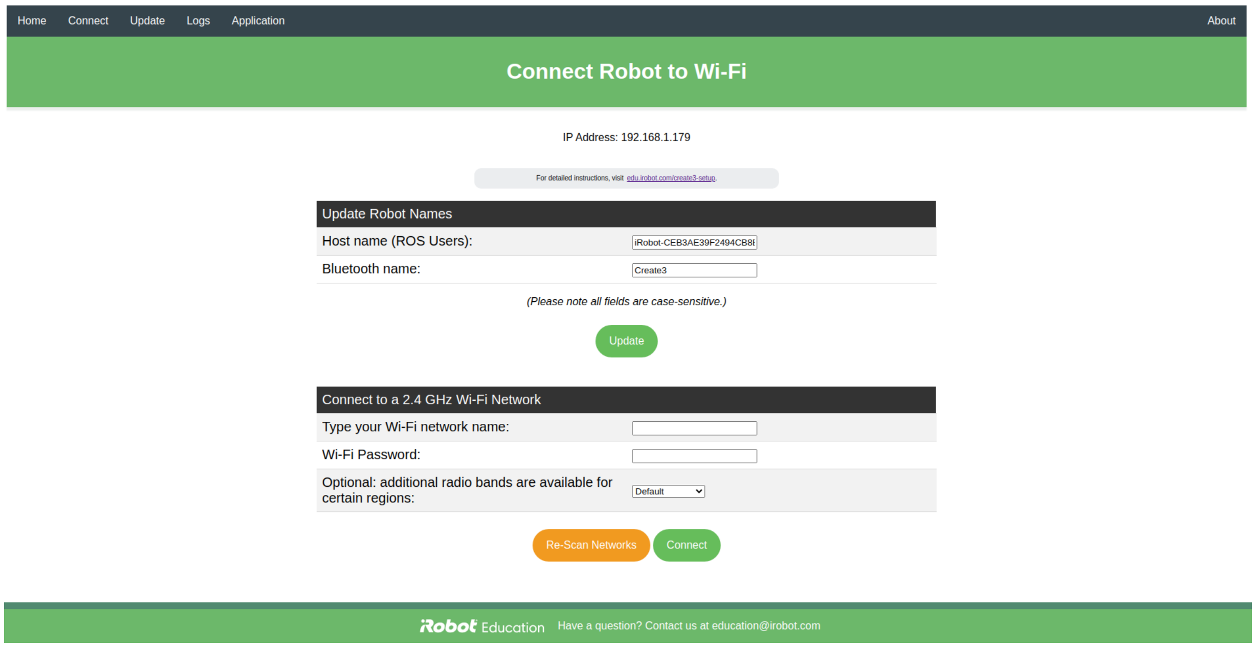
You should see the IP address printed out in your terminal periodically. This is your Raspberry PI IP, you can communicate with SSH:

```
$ ssh ubuntu@192.168.0.101
```

Login with the password *turtlebot4*. **Please don't make any change of the firmware and the software dependency inside the Raspberry PI!** If you use your own router, you can run the setup tool to configure WI-FI:

```
$ turtlebot4-setup
```

Open a web browser on your PC and navigate to your Raspberry Pi IP with the port 8080. You will be greeted by the Create® 3 webserver home page. Then navigate to the Connect tab. Please enter your Wi-Fi SSID and password, and then click 'Connect'.



Connect Robot to Wi-Fi

IP Address: 192.168.1.179

For detailed instructions, visit edu.irobot.com/create3-setup.

Update Robot Names	
Host name (ROS Users):	iRobot-CEB3AE39F2494CB8
Bluetooth name:	Create3

(Please note all fields are case-sensitive.)

Connect to a 2.4 GHz Wi-Fi Network	
Type your Wi-Fi network name:	
Wi-Fi Password:	
Optional: additional radio bands are available for certain regions:	Default

Have a question? Contact us at education@irobot.com

Wait for it to connect to Wi-Fi and play a chime. The Connect tab should now show an IP address.

You can now drive the robot by calling:

```
$ ros2 run teleop_twist_keyboard teleop_twist_keyboard
```

A CLI interface will be launched which allow you to command the robot to drive.

2. Mapping and RViz2

RViz2 is used to visualize the data the robot produces, such as camera and laser scanner feeds. RViz2 allows users to interactively visualize and interact with various aspects of a robot's operation, such as its movement, sensor readings, and the surrounding environment.

Make sure *turtlebot4_navigation* package is installed if you use your own PC:

```
$ sudo apt install ros-humble-turtlebot4-navigation
```

Run SLAM:

```
$ ros2 launch turtlebot4_navigation slam.launch.py
```

To visualise the map, launch Rviz2 with the *view_robot* launch file:

```
$ ros2 launch turtlebot4_viz view_robot.launch.py
```

Navigate the robot throughout the designated area for mapping while closely monitoring RViz2 to ensure the map gets filled out properly. Notably, the SLAM system tends to produce higher-accuracy maps when it identifies more loops.

While the process is running, we can have a look at the TF2 tree relationships, using the command:

```
$ ros2 run tf2_tools view_frames
```

This should display a TF2 tree and how the frames are connected. To view the tree, open the resulting *frames.pdf* file with your favorite PDF viewer. Save a screenshot of your TF2 tree and include it in your report.

If you are happy with the mapping results, you can save it with the following command:

```
$ ros2 service call /slam_toolbox/save_map slam_toolbox/srv/SaveMap "name:
data: 'map_name'"
```

It will generate a *map_name.pgm* file and a *map_name.yaml* file. You can edit this file to adjust the map parameters. Save a screenshot of your generated map image and include it in your report.

3. ROS2 Bag and Data Record

ROS2 Bag is a command line tool for recording data published on topics in your system. It accumulates the data passed on any number of topics and saves it in a database. You can then replay the data to reproduce the results of your tests and experiments. Therefore, recording topics is also a great way to share your work and allow others to recreate it.

Close the previous terminals and open a new mapping:

```
$ ros2 launch turtlebot4_navigation slam.launch.py
```

ROS2 Bag can only record data from published messages in topics. To see the list of your system's topics, open a new terminal and run the command:

```
$ ros2 topic list
```

Please consider which topic you need to record and use the command:

```
$ ros2 bag record -o <file_name>  <topic_name1> <topic_name2> <topic_name3>
```

The rosbag file will save in the directory where you run it with the file name. Navigate the robot and Press Ctrl+C to stop recording.

You can see details about your recording by running:

```
$ ros2 bag info <file_name>
```

Take a screenshot on the information you get and include it in your report.

To replay the bag, close the previous terminal and enter the command in the new window:

```
$ ros2 bag play <file_name>
```

This would be the dataset you generated. Run SLAM and launch RViz to see if it works, save the map and include it in your report.

4. Navigation

Open a terminal and launch localization:

$ ros2 launch turtlebot4_navigation localization.launch.py map:=map.yaml

Replace *map.yaml* with the path to your own map.

Open another terminal and launch Nav2:

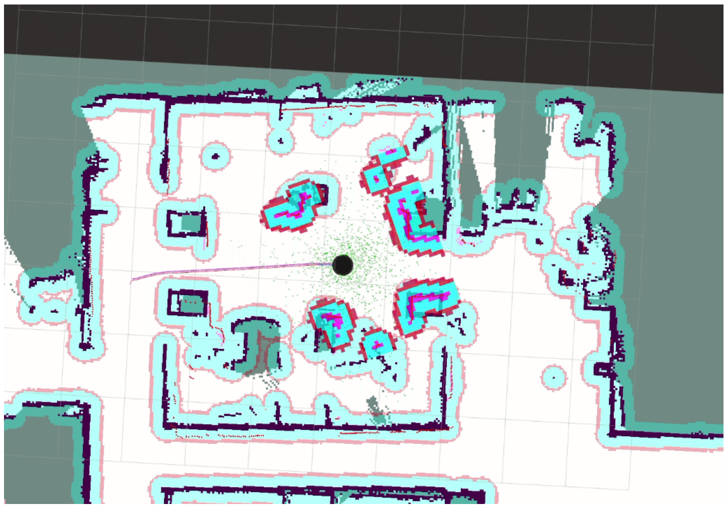
$ ros2 launch turtlebot4_navigation nav2.launch.py

Now you have already launched localization and Nav2, you need to open a new terminal to interact with navigation:

$ ros2 launch turtlebot4_viz view_robot.launch.py

You will notice that there are some available tools for you at the top of the RViz2 window. The first step is to set the approximate initial pose of the robot on the map. Click on the 2D Pose Estimate tool, and then click and drag the arrow on the map to approximate the position and orientation of the robot.

You also need to set a goal pose for the robot. Click the Nav2 Goal tool and set the goal for the robot to drive there, take a screenshot during the moving of the robot and include it in your report. An example of the result will be like this:



Run the following command, which will present you with all the active nodes inside ROS:

$ rqt_graph

Save the image and include it in your report with a proper description.

5. Gazebo Simulation (Optional)

Gazebo is an open-source 3D robot simulation environment. It allows users to simulate robots, sensors, and environments, providing a platform for testing and developing robot algorithms in a virtual environment before deploying them on physical robots. Gazebo supports a wide range of sensors, including cameras, lidars, and sonars, as well as different types of robots, such as wheeled, tracked, or flying robots.

For those who want to continue with a more challenging task, you may do so with the following instructions:

1. Create a new "world map" for the Gazebo environment, using ready-made elements which are already given in Gazebo. Be creative, the most important is to learn how to use Gazebo and the benefits it offers.

2. You will need to program a ROS 2 node which will drive the robot to certain locations inside your previously created map (you may freely define such locations, with a minimum of 3). This is similar to the listener/publisher nodes described in the ROS 2 tutorial, but both are inside the same node. Hint: Your node should publish to the current navigation goal and listen to whether the robot has already reached it or not.