# Getting Started with the Robot Management Application

## Installing the Robot Management Application

Follow the installation instruction in `Installation.pdf` within the documentation folder.

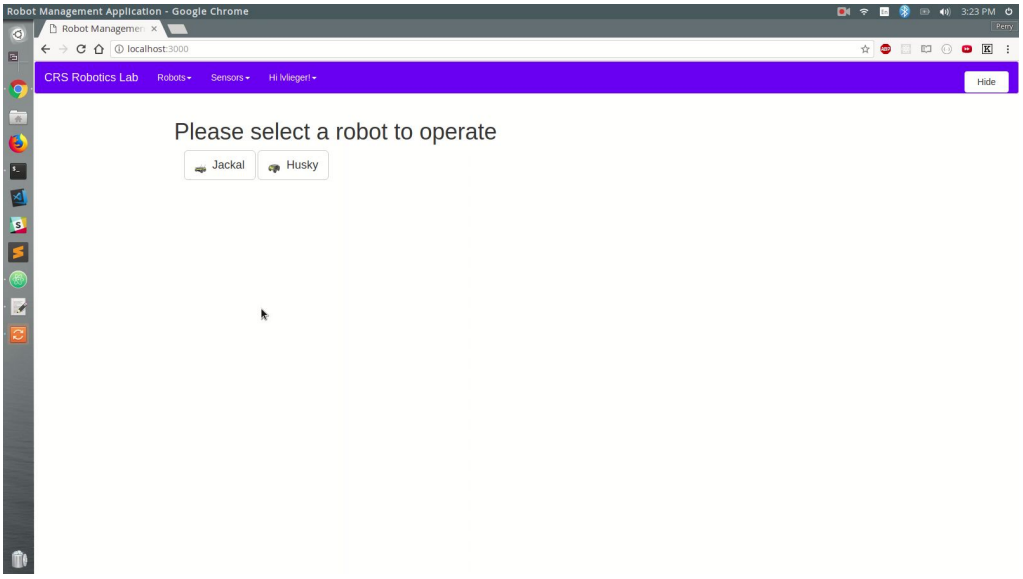
## Running the Robot Management Application

The following instruction will start the packages needed to open the websocket connection and control the robot.

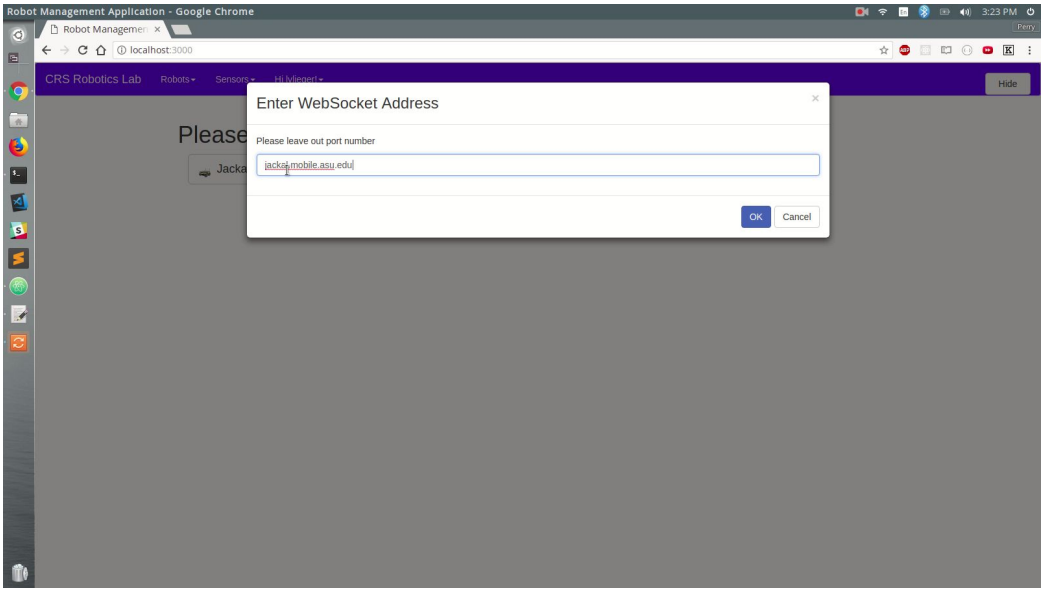
1. Run the rosbridge server to open a websocket connection on the robot
    a. Open a terminal tab and ssh to the robot you would like to connect to
    b. Execute the following command to startup the rosbridge server:
    ```
    roslaunch rosbridge_server rosbridge_websocket.launch
    ```
2. `(Optional)` If you would like to view the Kinect video stream in the web applications, run the kinect2_bridge and web_video_server packages:
    a. Open a new terminal tab and ssh to the robot
    b. Source the catkin_ws setup script:
    ```
    cd ~/catkin_ws
    source ./devel/setup.sh
    ```
    c. Run the kinect2_bridge package to enable ROS access to the Kinect image topics
    ```
    roslaunch kinect2_bridge kinect2_bridge.launch
    ```
    d. Open a new terminal tab and ssh to the robot
    e. Run the web video server:
    ```
    rosrun web_video_server web_video_server
    ```
3. `(Optional)` If you want to show an existing map from the robot, there are several packages that need to be started:
    a. Open a new terminal tab and ssh to the robot
    b. Source the catkin_ws setup script:
    ```
    cd ~/catkin_ws
    source ./devel/setup.sh
    ```
    c. Run the kinect2_bridge package to enable ROS access to the Kinect image topics
    ```
    roslaunch kinect2_bridge kinect2_bridge.launch
    publish_tf:=true
    ```
    d. Open a new terminal tab and ssh to the robot
    e. Run the TF transform publisher
    ```
    rosrun tf static_transform_publisher 0 0 0
    -1.5707963267948966 0 -1.5707963267948966 camera_link
    kinect2_link 100
    ```

f. Open a new terminal tab and ssh to the robot
g. Run rtabmap in localization mode. This will startup the map and robot will begin searching for its current location.
```
roslaunch rtabmap_ros rtabmap.launch localization:=true rgb_topic:=/kinect2/qhd/image_color_rect depth_topic:=/kinect2/qhd/image_depth_rect camera_info_topic:=/kinect2/qhd/camera_info
```
h. Open a new terminal and ssh to the robot
i. Run the robot pose publisher so the web application can know the robot's current location:
```
rosrun robot_pose_publisher robot_pose_publisher
```

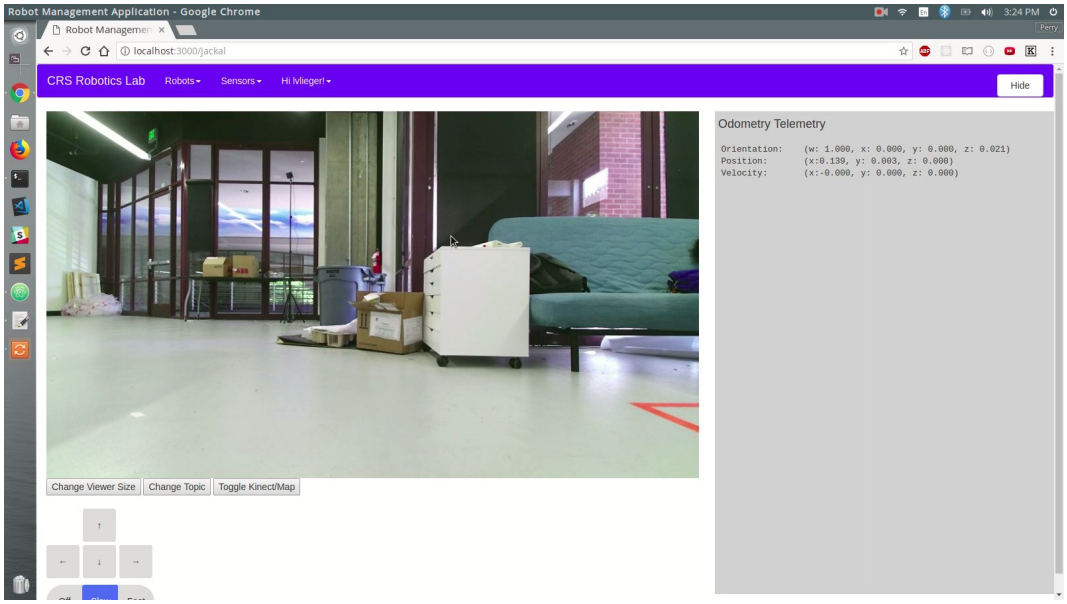
4. If you have not already, start up the robot management application
   a. Open a terminal tab and change the directory to the project `scripts` folder.
   b. Make sure the debug.sh script is executable:
   ```
   chmod +x debug.sh
   ```
   c. Execute the debug.sh script to run the application
   ```
   ./debug.sh
   ```
   d. The website will now be available on `localhost:4200`
5. Connect to the robot through the web application
   a. Select the robot (Jackal or Husky) you would like to connect to



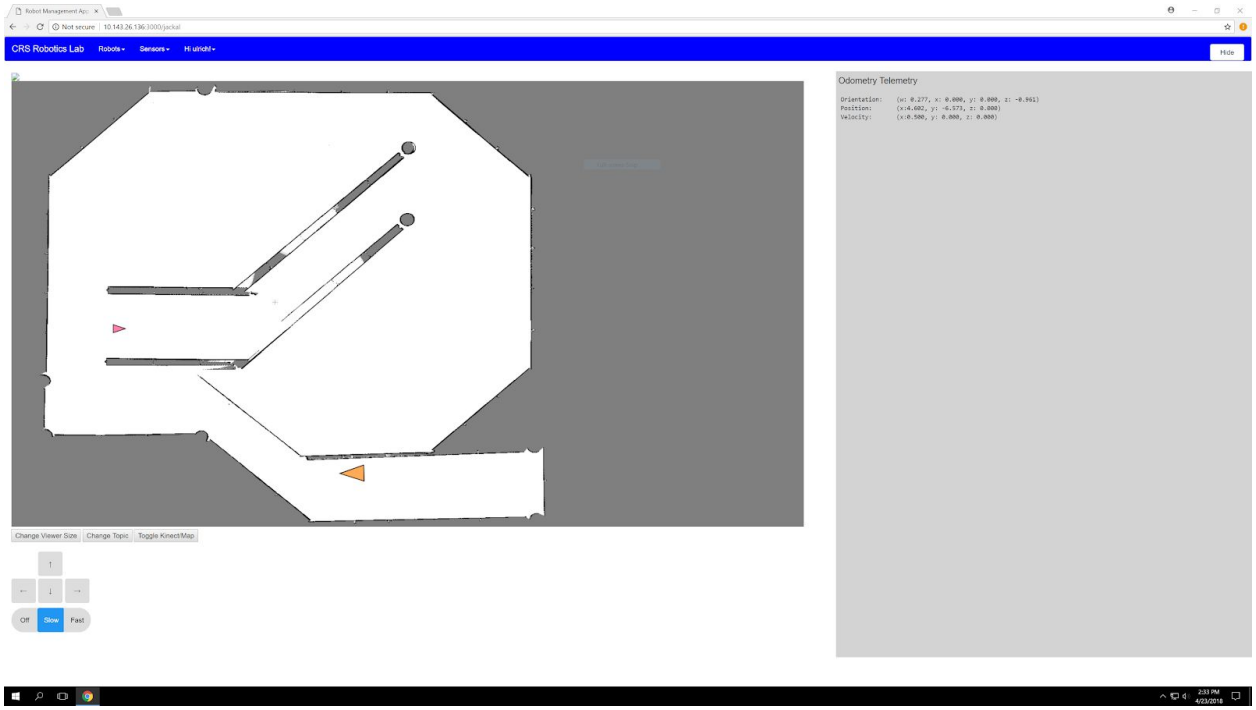
b. Enter the websocket address of the robot




6. You can now control the robot through the teleop control



7. If you want to view the map, you can click on the “Toggle View Map button




8. (Optional) If you would like to view a screen grab of gazebo or rviz:
    a. Clone screengrab_ros from https://github.com/lucasw/screen_grab
    b. Add screen_grab to catkin_ws/src and use catkin_make.
    c. Open a terminal tab:
    d. Source devel/setup.sh if not already done.
    `roslaunch screen_grab screen_grab.launch`
    e. Now we can change the region of interest in the screen by publishing to screen_grab/roi, and the video is constantly published to the topic /image. The region of interest is made up of an xOffset and yOffset from the top left of the screen, in pixels, as well as a height and width starting from that offset.
    f. In order to make this easily viewable in the web application, use the web_video_server package:
    `rosrun web_video_server web_video_server`
        i. This makes the image topics available on port 8080, where the web application connects to /image

## Creating a Map with Gmapping

To create a map with the Jackal robot, follow the instruction in `Jackal_Gmapping.pdf` file