Project 3 FAQ

Bookmark this page

- How do run this project in my own Ubuntu machine?

  1. Launch Project 3, then in Vocareum click Actions>Download Starter code. This will download all the files you need to make the project run locally in your computer.
  2. Install the needed ROS package(s). Run the following lines on your terminal:
     ```
     sudo apt-get update
     sudo apt-get install ros-kinetic-urdfdom-py
     ```
     Replace kinetic with the ROS version that you are running on your local machine.
  3. **IGNORE** all the files other than catkin_ws and kuka_lwr_arm.urdf. Copy the folder catkin_ws to your home directory (you can rename it project3 if you want). Also put the file kuka_lwr_arm.urdf in the home directory.
  4. The downloaded files are structured as a catkin workspace. Navigate to the folder catkin_ws in your home directory using "cd catkin_ws" or whatever name you gave the workspace ("cd project3"). If you are running ROS Kinetic you need to modify the CMakeList.txt file in the robot sim package before running catking_make (see note in the last FAQ bullet point). Once inside your catkin workspace, run the command "catkin_make".
     If you are having troubles with this, you should review the first ROS tutorial "Installing and configuring your ROS Environment".
  5. At this point if the catkin_make command was successful, you are ready to work on your project without having to make any changes in any of the files.
  6. NOTE: You can source both your ROS distribution and your catkin workspace automatically everytime you open up a terminal automatically by editing the ~/.bashrc file in your home directory. For example if your ROS distribution is Kinetic, and your catkin workspace is called "project3_ws" (and is located in your home directory) then you can add the following at the end of your .bashrc file:
     ```
     source /opt/ros/kinetic/setup.bash
     echo "ROS Kinetic was sourced"
     source ~/project3_ws/devel/setup.bash
     echo "project3_ws workspace was sourced"
     ```
     This way every time you open up a terminal, you will already have your workspace sourced, such that ROS will have knowledge of the packages there.
  7. **Before moving forward, if you haven't followed the instructions on step 6, you will need to source ROS and the catkin workspace every time you open a new terminal.** To run the project, first open up a terminal and type "`roscore`". In the second terminal (remember to source ROS and the catkin workspace if you didn't do step 6) run "rosparam set robot_description --textfile kuka_lwr_arm.urdf", followed by "`rosrun robot_sim robot_sim_bringup`".
  8. On another 2 separate terminals you need to run the scripts for the robot mover and the your solution in forward kinematics : "`rosrun robot_mover mover`" and "`rosrun forward_kinematics solution.py`". Note that you can find these lines from setup_project3.sh in the starter code.
  9. Now we can open up Rviz using "rosrun rviz rviz". Inside Rviz, first change the Fixed Frame to "world_link" (you might not be able to do this until you start writing your solution code since there will not be any TF for "world_link"). Then click Add and select RobotModel from the list of options. At this point if you code works, you should see the robot arm rendered and moving in a coherent way back and forth from an upright position to a another predetermined pose. You can also see the transforms if you select Add > TF

edX

Legal

Connect