



Due date: 09/20/2022, 11:00am, before class starts. This assignment is worth 20 points.

You will create your first ROS package. You will implement a program that subscribes to a laser scan message from the HSR robot and processes this data. We suggest to refer to the ROS best practices available at <http://wiki.ros.org/BestPractices>. There is a lot of help with regard to conventions, units etc. They also provide a (C++) template you can use as a starting point.

1. Create a package `[USERNAME]_hsrb_behavior_controller` from scratch. Use the command `catkin_create_pkg` to create a new package with the dependencies `roscpp` (or `rospy` if you prefer to program in Python) and `std_msgs` as well as `sensor_msgs`. Refer to the tutorial "Creating a ROS Package" if you are stuck: <http://wiki.ros.org/ROS/Tutorials/CreatingPackage>.
2. Check the `CMakeLists.txt` and `package.xml` files.
3. Create a subscriber to the topic `/hsrb/base_scan`.
4. Add a parameter file with topic name and queue size for the subscriber of the topic `/hsrb/base_scan`.
5. Create a callback method for that subscriber which delivers the smallest distance measurement from the vector ranges in the message of the laser scanner to the terminal. The message type is defined as: http://docs.ros.org/en/melodic/api/sensor_msgs/html/msg/LaserScan.html. You might want to launch the empty HSR world with `roslaunch hsrb_gazebo_launch hsrb_empty_world.launch` and then add an object into Gazebo to get a distance measure.
6. Add your a launch file to this package and add at least the following:
 - Include the empty HSRB world (`hsrb_empty_world.world`) in your launch file.
 - Add your node (`[USERNAME]_hsrb_behavior_controller`)
7. Pass the parameter `load` to your node configuration. The file to load is `config.yaml` in the `config` folder (which is in your package home directory).
8. Run RViz and visualize the laser scan. RViz is actually automatically loaded by the `hsrb_empty_world.launch` file that comes with the docker image. The configuration is showing a large set of topics, which might be overwhelming for this assignment. I suggest to set this RViz configuration to `false` in your launch file and then add another node with a fresh (empty) version of RViz. Better would be your customized version on RViz for this assignment, which means that the laserscan and robot model are loaded. Also, you might need to adapt the size of the laser scan points. You can save your RViz configuration as the default configuration for the future.
9. Submit the entire package by committing it to the SVN into your user's directory. Credentials to the SVN repository will be given to you in class. Instructions about how to commit to the repository are on our class web page.