1. Navigate to the parent local directory where you want to share with your docker container.
2. Download the ROS bag file into the parent local directory.
3. Run gedit DockerFile and generate a set of commands for the docker container to execute upon initialization. Put the following code in the DockerFile:

| # syntax=docker/dockerfile:1  FROM ros:noetic-robot  # change the default shell command  SHELL ["/bin/bash", "-c"]  # this gets run in this image  RUN source ros\_entrypoint.sh  # now add this to all future calls  RUN echo "source /ros\_entrypoint.sh" >> /etc/bash.bashrc  # create our catkin workspace  RUN mkdir -p catkin\_ws/src  # get the  RUN source /etc/bash.bashrc  # change to our working directory  WORKDIR catkin\_ws  RUN source /ros\_entrypoint.sh && catkin\_make  # clone the packages we need  # get the git package  RUN apt-get -y update  RUN apt-get -y install git  WORKDIR /catkin\_ws/src  # you only get one command  CMD roscore |
| --- |

1. Run *docker build -t test:latest .* in the parent directory the load the DockerFile
2. Run sudo *docker run --mount type=bind,source=.,target=/catkin\_ws/src -it test:latest*
3. Go into terminal and type *docker exec -it [container name (you can hit tab to figure this out)] bash*
4. Open matlab and enter simulink then rosinit in the terminal window. Make sure the directory for MatLab is the same as the parent local directory that you want to share with your docker container
5. Open your controller simulink file (in our case our controller is called *“profproject”*
6. In simulink, press “Generate Code”
7. You should receive a .tgz file.
8. Unpack the .tgz with tar to extract the .tgz file
9. Perform steps 6-10 but for the lead car model that your controller code will react to. ( in our case the car model is called “carsimplesimulink” )
10. Navigate to the extracted directory and cut the folder there into your parent directory
11. Make sure your directory is catkin\_ws/src and run the command *catkin\_create\_pkg instructor*
12. In your local parent directory run *cd instructor/launch*
13. In local terminal run *gedit instructortest.launch.*  This will start a text editor GUI where you can put the following code:

| <launch>  <node pkg="rosbag" type="play" name="rosbag\_playback" args="/catkin\_ws/src/ros\_car.bag"/>    <!--  <node pkg="cc\_carexample\_template\_inclass" type="CC\_CarExample\_template\_inclass" name="car\_example\_node">  <remap from="/car/state/vel\_x" to="/vel\_x"/>  </node>  →  <node pkg="profproject" type="profproject" name="profproject\_node">  <remap from="/car/state/vel\_x" to="/vel\_x"/>  </node>  <node pkg="carsimplesimulink" type="carsimplesimulink" name="carsimplesimulink\_node"/>    </launch> |
| --- |

1. Go into your docker container terminal and type *cd ../* to enter into the catkin\_ws
2. Run *catkin\_make*  in the docker container terminal to create a workspace
3. Run *source /devel/setup.bash* to set up ros command in all children directory within catkin\_ws
4. Type *cd src*
5. Run *roslaunch instructor instructortest.launch* to execute the ROS generated code
6. If you want to observe the relevant topics used by the nodes or see which nodes are running, first type *rostopic list or rosnode list respectively.*