For this lab achieve the objectives of PS7, only this time do so on Jinx. Reference the startup and shutdown procedures (to be discussed in class and summarized in the Jinx Overview Presentation contained in the Files folder).

In addition add one new deliverable: graceful halt from estop. For this purpose use the std_msgs/bool topic ESTOP which should read true when you've triggered an ESTOP. Your trajectory_builder.cpp node should be smart enough to pick up from wherever the open loop trajectory was interrupted when the ESTOP or LIDAR alarm was triggered. So if you only made it half-way to your goal when you encountered the ESTOP you should finish the other half of the trajectory when the ESTOP condition is released AND Jinx should start up gracefully after the ESTOP release. Get used to using the wireless ESTOP to trigger these events. Someone should always be holding the wireless ESTOP when working with Jinx.

In summary you should demonstrate proper functionality of the following behaviors:

- graceful halt from ESTOP
- graceful recovery from ESTOP

- graceful halt from LIDAR alarm
- graceful recovery from LIDAR alarm
- ability to execute open-loop control corresponding to a prescribed polyline path
- ability to flush a path plan and replace it