Mobile Robotics EECS 476/376
Robot Operating Instructions

1) Power up:
Before powering up the robot,

a) make sure the red E-stop button is depressed
b) *Make sure "RC" toggle switch is in the "off" position* (autonomous, not RC mode). When in the RC mode, the robot will respond to our RC controller, which feeds raw motor power commands to the motor drives (via the cRIO), bypassing the PC. You should not need this mode, though it is handy for confirming that the low-level drive-train is functional, and sometimes for transporting the robot without starting up control code.
c) assign a "safety officer" to hold the remote E-stop

Enable main power via the latch on the 150A breaker. (When latch is flush with the breaker, power is enabled. When the small red button on the main breaker is pressed, this latch will extend to cut power).

Power up the robot's computer. This requires reaching inside to press the PC's power button, since the computer does not auto-boot when line power is enabled.

2) Connect to the robot via WiFi from one of the development stations to Jinx (or Alen, depending on which robot you are using). This will not give you Internet access, but it will provide a local link between the development computer and the PC on the robot. The IP address of the robot's access point is 192.168.0.150. (The lower-level cRIO on the robot uses 192.168.0.100).

ssh to the robot from the development station. E.g., for jinx, enter: ssh jinx@192.168.0.150 and use password 1lmiagmc! From the dev-station window, you will get a prompt ...-jinx:

change directory (on the remote robot from the local dev station0: roscd cwru_semi_stable

Type: screen <CR> This launches the "screen" program on the dev station, which is a terminal multiplexer. ctl-A, followed by a number switches to the specified numbered screen.

3) Launch ROS nodes on robot: ***roslaunch cwru_bringup_no_tele.launch*** This should start up a variety of useful (and necessary) processes. The processes launched will be listed (though you can ignore "permission denied" statements that refer to sound devices—you will not have sound devices on your robots). If you see error messages, you may have a hardware problem (e.g. the Sick LIDAR is not plugged in).

At this point, various publishers are active, and your code can subscribe to the topics. This can be done from the dev station as well, which is valuable for system diagnostics.

Note--if you lose the "screen" program due to some disruption, you can restart "screen" with: ***screen –r*** to recover where you left off. If there was a particularly ungraceful disconnect, you may need: ***screen –r –d*** (disconnect and re-attach option).

4) Start up the diagnostic interface: The following lines enable local graphics of remote (jinx) nodes and start up a diagnostic program:

- open a local terminal (not running on Jinx);
- ***roscd cwru_semi_stable***
- ***cd ..***
- ***source connect_harlie.bash***
- ***rosrun runtime_monitor monitor***

The diagnostic program will interact with roscore running on the robot and display data on a (local) GUI.

(Further details to be provided re/ use of the diagnostic GUI).

rosrun rviz rviz should be invoked from the window from which you ran "source connect_harlie.bash" From the resulting window, you can view the LIDAR data. You should confirm that the LIDAR is functioning; e.g. you should be able to see the effect of someone walking in front of the robot.

5) test Xbox teleop: Testing that the robot responds appropriately to X-box controller commands will confirm that higher-level control processes are operating correctly, that commands are successfully sent to the cRIO, and that the cRIO is performing stable feedback control of wheel speeds.

Power up the X-box controller (note—pop the battery holder out when not in use, to extend battery life). The X-box controller should have a flashing green light

- in an additional screen (terminal on dev station, remote to jinx), start up the X-box teleoperation code by typing: ***roslaunch cwru_teleop start_teleop.launch***
- The safety operator should make sure that the wireless E-stop has disabled the robot. Check the LED indicator on the white plastic box at the back of the robot. The red light will be on if the E-stop is safe. (i.e., "red means motors dead")
- Enable the manual E-stop. This is done by twisting the red button in the direction indicated, which will cause it to pop outwards. From this state, if you hit the red button, it will depress (and stay in its depressed state) and cause power to the motors to be cut. This will *not* cut power to the computers, so there is no harm done in hitting this button at any time. (In contrast, if you trip the main 150A breaker, this will cut power to the entire system, so only do this when you are ready to shut down for the day, or if the computers catch fire!).
- The safety officer should assure that the area around the robot is clear (particularly of people!).
- The safety officer should confirm that the operator (person holding the live X-box controller) is ready, and then should announce that the robot is about to be enabled. It is a good idea to give a quick count-down (3…2…1…enable) prior to pressing the remote E-stop button.
- At this point, the X-box controller can send motion commands to the robot. The operator should test this carefully with small motions of the left thumbstick/joystick. ***Be ready and be careful!*** Some types of sensor failures (e.g. broken encoder or loose encoder cable) can result in the robot running away at high speed. The safety officer should be prepared to hit the remote E-stop.

- If the robot responds appropriately (drives under joystick control), it is ready to run your code. If the robot does not respond correctly, E-stop and debug the system. You may start by repeating the above steps. If you still have a problem, you likely will need to consult a TA.
- The safety officer should E-stop the robot (first wireless, then manual E-stop button).
- Eject the X-box controller's battery pack to conserve power
- From the window in which you launched the cwru_teleop process, stop the process with ctl-C

6) Run your own control code: To run your custom code (e.g. command_publisher), you will need to transfer your code from the development station to the robot's PC.

- on the development computer: **make clean** (before copying to jinx)
- move code from dev station to robot:
    - places->connect to server,
    - select service type SSH,
    - type in IP addr, user name "jinx" (for robot "jinx")
    - hit "connect" button,
    - type password,
    - hit "connect", (will see jinx's files)
    - navigate to home/jinx/ROSCode/
    - in separate window, find desired file;
    - drag file from source window to destination (jinx) window
    - hit "eject" button for jinx sftp
- rebuild your code (illustrated using package name "wsn_hw4"): on jinx (via a remote terminal on the dev station),
    - **roscd wsn_hw4**
    - **rosmake wsn_hw4** (if errors, try rosmake --pre-clean wsn_hw4)
- on the development station, with Rviz running, initialize pose the pose and wiggle the robot to refine the pose estimate (as reconciled with respect to the map and LIDAR values)
- in a jinx terminal: **roslaunch cwru_nav start_amcl_2ndfloor.launch**
- Enable the robot (repeat steps to release the manual E-stop and countdown by safety operator to enable wireless E-stop). Jinx should stay still (since the command publisher is not yet running)
- in another jinx terminal, **rosrun wsn_hw4 command_publisher** This will start up your control program (command_publisher) associated with your package (here, called "wsn_hw4")
- The robot should execute your control program at this point. Be prepared to E-stop the robot if you do not get the expected behavior. Also, monitor the battery voltage. It will display approximately 26V when fully charged. Put the robot on the charger if the voltage drops below about 22V.

7) Shut down the robot:

- Ctl-C the processes on the robot
- On the (remote) robot, execute: **sudo shutdown –h now** to shut down the PC gracefully.
- Hit the red button on the main breaker
- Plug the charger into the robot (and make sure the batteries are charging)
- Log off from the development station