Introduction to SPEEDY, the Turtlebot

Comp 380

Susan Fox

March 28, 2014

1 Overview of the Turtlebot

The Turtlebot is a low-cost platform with a great low-cost distance sensor, the Kinect on it. Its base is a Roomba Create, and the rest is a structure for supporting the Create at a reasonable height. It has a small laptop onboard. Ideally, you would have the laptop handling the low-level stuff, and you'd put it on the bottom shelf of the structure, closed. Then you'd actually control the robot with a separate computer, another laptop or a base station. In practice that has issues with the dynamic IP addresses and inconsistent Wifi that we have.

Much like the Nao, this tutorial will focus on how to use ROS to run pre-defined programs. You can then delved deeper if you want into how to *create* programs for the Turtlebot using ROS.

You will be using some of the tutorial on the ROS website. Here is the main page about the Turtlebot platform:

`http://wiki.ros.org/Robots/TurtleBot`

The Create is not set up to roll freely when it is off. Thus you have to carry the robot carefully to where you want to work with it. I will clear off the big metal cart, feel free to put the robot on top of it if you are taking it elsewhere in the building. It is very possible to use the robot just in the robot lab, if we clear some space.

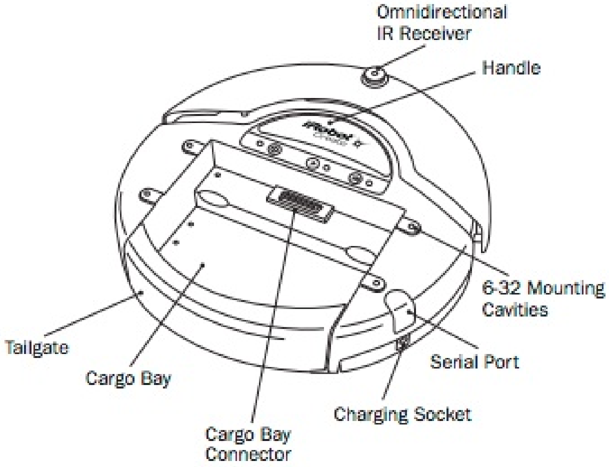
The Roomba Create can be annoying to work with. It is either in charge mode, or work mode. And getting it into the right mode takes some work, and occasionally just doesn't work (the sequence appears to be extremely important, and I'm still figuring out the perfect order).

The next sections start with an overview of charging the Create, which is one of the bugbears of the Turtlebot. Then I go over the startup and shutdown processes, explaining the Turtlebot Dashboard. Next I will explain how to try teleoperation and follower sample programs. Lastly, you will try on your own to work through the Calibration tutorial. The `rviz` program wasn't working for me, so I don't know if you can run the `SLAM` tutorial program, but you could try.

2 Turtlebot Care and Feeding

There are two batteries in the TurtleBot, the laptop battery and the Create battery. Each has an independent charging brick and charges independently, you will see the status of both in the TurtleBot dashboard. Actually, our particular laptop doesn't let ROS know what its battery level is, but that's okay.

Here is a diagram of the iRobot Create's top. Note the location of the power button and the power LED. You won't need any other buttons or LEDs. Also, note the location of the charging socket, and the serial port for connecting the Create to the laptop.



The iRobot Create will only charge when it is plugged in after being set in Passive mode. To do this, you can doe either of the following.

If the Turtlebot dashboard is running: Put the robot into Passive mode using the Mode button on the TurtleBot dashboard.

If ROS and the Turtlebot dashboard is not running: Press the power button, on the Create to make sure it is off. When you plug in, it will flash the power light repeatedly and if it starts charging it will throb with a orange/red light. The table below explains the different colors of the power LED indicating power level and charge status.

Color of Power Light	Battery status
Slow Pulsing Orange	Charging
Fast Pulsing Orange	Reconditioning Charge
Green	Fully Charged
Amber	Partially Discharged
Red	Almost Fully Discharged
Flashing Red	Fully Discharged

If the Create will be off for more than 12 hours: then you must take extra measures to ensure the battery does not lose all power. (Susan's aside: what follows is *insane*, I have no idea why the Create works this way, but we're stuck with it!) When the Create has been plugged in to charge, it will charge fully, then turn the charging process off. It does **not** restart charging even if the battery level drops, unless it you go through the process above over again. If there is a draw on the battery when the robot is stored it can run out of battery relatively quickly. To avoid this follow these rules.

- While the dashboard is running, make sure all the breakers are off before shutting down.

- Unplug the Serial port just above the power charging port. (The LED in the dongle is enough to drain the battery over the course of a day or so.)

3 Speedy startup process

When you are ready to get Speedy going, go through the following steps.

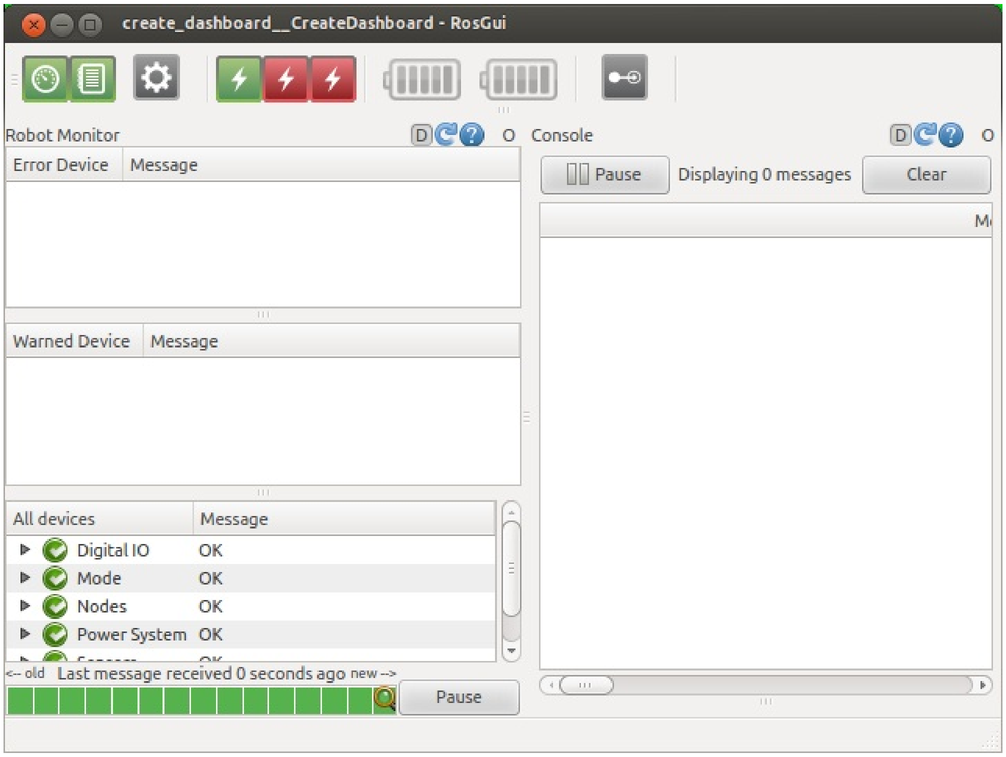
1. Make sure that the laptop is on and going. The username and password for the Lenovo laptop is `turtlebot` and `turtlebot`.
2. Make sure that the USB cables from the Create and the Kinect are **not** plugged into the laptop.
3. Make sure the Create is turned off. Unplug the power cable, and turn the Create back on. Verify that the battery is fully charged: the LED should be green (if it is not, then you can still try to use the robot, but the battery goes down rather quickly). Unplug the battery cable, and *turn the Create off.*
4. Plug the two USB cables into ports on the right side of the laptop, and turn the Create back on. (Sometimes it seems to turn off its power LED, not sure why.)
5. Open a terminal, and run the following to set up the Turtlebot control program (remember that commands and files do name-completion, type part of the name and then hit tab and it will fill in as much as it can figure out):

   ```
   roslaunch turtlebot_bringup minimal.launch
   ```

   If you want to run all the commands in the same terminal, then put an and sign at the end of the command above (`&`) to tell it to run the program in the background. Otherwise, open a new terminal tab for the next step.
6. In a new terminal, run the following to open the Turtlebot dashboard:

   ```
   roslaunch turtlebot_dashboard turtlebot_dashboard.launch
   ```

   This should bring up the Turtlebot dashboard, a place to see the status of the robot and messages from various Turtlebot programs. The dashboard should look something like the image below (clicking on the two buttons in the upper-left will bring up some of the other frames on this window.



All devices	Message
Digital IO	OK
Mode	OK
Nodes	OK
Power System	OK

<-- old Last message received 0 seconds ago new -->

The upper left button (that looks like a dial) is the Diagnostics button. If it is green, then no warnings or errors have been detected. Clicking on this button brings up the Robot Monitor panels of the window, where you can see any errors or warning that have occurred.

The button just to the right, that looks like a box of text, is the Rosout button. If it is green, then no error or warning outputs have taken place in the past 30 seconds. Clicking this button brings up the roseconsole logging view, where you can sees all messages generated by ROS programs.

The next button, marked with a gear, changes the Create's mode. Passive mode is the mode in which the Create's battery may be charged. Full is the mode to use when the robot should move. (Make sure that this button shows green, or the robot's motors are not enabled. Click on the button to change it?)

Next there are three lightning-marked buttons. These are the three "breakers" that control the flow of power to parts of the robot. The first break sends power to the Kinect. It must be green to use the Kinect. The other two breakers connect to digit outputs on the Kinect (We must experiment to see if the robot will move without them set, but I have a feeling that it will). When shutting down the robot, uncheck all three breakers (make them read).

Further to the right are two battery level indicators. The one on the left will be gray, the Lenovo laptop doesn't share its battery level. The one on the right shows the Create battery level. It is sometimes not up-to-date. In particular, it sometimes reads full when the dashboard is not actually in communication with the robot, and will change dramatically once the robot actually starts doing something. Watch it carefully!

They gray button beyond that: I have no idea what it does.

3.1 Putting Speedy “to sleep” for a short period

Suppose you are working with Speedy and you need to go off to class or supper for an hour. You might not want to leave it on and draining the laptop or Create batteries for that time period. So what do you do?

Do this:

1. Probably a good idea to shut down everything except the minimal launch and the dashboard, especially if you are running something using the Kinect.
2. Go to the dashboard. Uncheck the breaker for the Kinect. Then, Set the Create into Passive Mode.
3. Finally, plug in the laptop and Create. They can stay on, the laptop will charge normally, and the Create will charge because it is in passive mode.
4. When you come back, unplug the Create, and use the dashboard to set the Create back into Full Mode for further operation.
5. Make sure that the laptop power cable is unplugged before you initiate any robot movements!

3.2 Speedy shutdown process

When you are done working with Speedy, do the following steps to shut it cleanly, so that the Create will charge and the program will run better later.

1. Shut down the various programs you have run, in reverse order, until only the Turtlebot dashboard the the basic Turtlebot `minimal_launch` are left running.
2. Go to the dashboard, and turn all the breakers red. Then, put the Create into Passive mode. Finally, quit the dashboard program.
3. Lastly, go to the terminal where you ran the minimal launch, and shut it down.
4. Make sure the laptop is plugged in to charge.
5. Unplug both USB cables from the robot to the laptop.
6. Follow the instructions above for setting the Create up for 12-hour+ storage.

Notes on working with ROS and Linux: To shut down a running ROS program, type `Control-c` in the terminal. If you started a program in the background, you can bring it back to the foreground to stop it by typing `fg` in the terminal. The command `ps` will show you all the programs running from this particular terminal.

4 ROS: the Robot Operating System

ROS is an open-source robot control system, designed for maximum flexibility. You can control multiple robots at once, you can run your control program on one computer and communicate over the network with the computer on the robot. It comes with many built-in behaviors. It is not the easiest program to use, and it is mostly designed to work on Linux systems (I have never gotten it to run on my Mac). It is terminal-based, so you’ll have to get used to that. And there are many operating-system-like tools as a part of it.

If you are interested in digging into ROS and its many tools, start with this tutorial online:

```
http://wiki.ros.org/ROS/Tutorials/NavigatingTheFilesystem
```

If you chose to do your project on the Turtlebot, then you would have to dig into ROS more, but for this assignment, just take it as it is.

5 Teleoperating Speedy

Set up Speedy according to the directions above. In the Dashboard, Speedy should be set in Full Mode, and the Kinect breaker should be set. The other two may be ignored, I believe.

Open a new terminal tab, and in it type the following command:

```
roslaunch turtlebot_teleop keyboard_teleop.launch
```

You can then use the keyboard to move the robot around. This is more easily done when using a separate computer to control the robot; bear in mind that the Kinect is "forward" to the robot, so either **gently** turn the laptop around or reverse the directions. It uses nine keys on the right side of the keyboard as shown below (a message in the program also lists this):

u Forward and curving left	i Straight forward	o Forward and curving right
j Turn left in place	k Stop moving	l Turn right in place
m Backward and curving left	, Straight backward	/ Backward and curving right

Try driving the Turtlebot around a little bit. To quit the program, type `Control-c`.

6 Speedy Follower

Set up Speedy according to the directions above. In the Dashboard, Speedy should be set in Full Mode, and the Kinect breaker should be set. The other two may be ignored, I believe.

Open a new terminal tab, and in it type the following command:

```
roslaunch turtlebot_teleop keyboard_teleop.launch
```

Before typing this command, make sure that you are not sitting really close to the robot in front of the Kinect. This program makes the robot want to stay a set distance away from an object it detects in front of it. If you are quite close, it may shoot away from you quickly.

I have found that I can either force the robot to back up to where I want it by moving toward it, or have it follow me as I walk backwards. Try it in a variety of locations and see how it works. When does it get confused?

To end this program, type `Control-c`.

7 Tasks to Try

1. Go to the tutorial listed below, and read the part about the dynamic recalibration of the Turtlebot. Try varying the sliders, as it describes, and report what happened.

   `http://wiki.ros.org/turtlebot_follower/Tutorials/Demo`

2. Go to the tutorial listed below, and see if you can get the Calibration program to run. If it seems successful, recalibrate the values as calibration suggests. Then, run the SLAM demo (second link below). I don't know if this last will work, because it might need `rviz`, which isn't working, but give it a try.

   `http://wiki.ros.org/turtlebot_calibration/Tutorials/Calibrate%20Odometry%20and%20Gyro`

   `http://wiki.ros.org/turtlebot_navigation/Tutorials/Build%20a%20map%20with%20SLAM`