

## **Robotics: Science and Systems – Fall 2005**

### **Lab 1: Hardware Familiarization and Organization**

**Thursday, Sept 8 2005**

**Completion by start of lab, Tuesday, Sept 13, 2005**

Greetings, RSS-II students. Welcome back those who completed RSS-I. An enthusiastic welcome also to those who have joined the course mid-stride. It figures to be a challenging and exciting term as we complete our incremental shift from tightly scoped, step by step labs to labs in which goals and tasks will be stated more broadly with less explicit direction on how they should be achieved. Once again we will form teams. The teams will work together to rebuild the robots created during the Spring term during the first three weeks of the term. We assigned a mix of course alumni and newcomers to each team for this task. After the completion of this part of the course, we will form new teams that should stay in place (more or less) throughout the term. These latter teams will NOT be doing the same thing however. Collectively, the whole class will decompose the course challenge. After this decomposition, each team will assume some task of the decomposition. Similar to the course challenge from RSS-I, we expect and will allow a team to design their unique solution to their task. To add complexity (and realism!), a team's contribution will additionally be subject to design reviews and it will have to interface with other tasks developed by other teams.

### **Lab Objectives**

Over the first 3 weeks of the term, each team needs to get their robot up and running. In Lab 1 the focus is on the hardware and setting up a adequate environment for your software development. One goal is to familiarize yourself with the hardware components, their specifications, wiring and function. You will gain experience in planning neat, careful wiring and labeling so that a piece of hardware (such as left and right motor) can correctly be associated with its logical reference in software. A second goal is for each group member to set up his/her personal software environment with respect to handling source code control, install an IDE or code editor, establish a document preparation tool (e.g. Latex or MS word). For formal write ups we require either PDF or HTML. Finally, each team should establish a wiki which the RSS staff can use with the team to view their repository for software and assignments, plus project specifications, videos and other documentation.

### **Your Lab Book, Self-Assessments and Time Accounting**

You will receive your very own RSS Lab book. You must use it for all your note taking throughout the term. Put all your scribbles, notes, important numbers (e.g. measurements and phone and combinations), descriptions of your conceptual model — in short, *everything* you *hand write* in it.

Devote the last four pages of your lab book to “Time Accounting” and “Self Assessment” sections (two pages each). Every lab you should set up Time Accounting and Self Assessment. Whenever you spend a substantial chunk of time thinking about or doing RSS lab work, note the length of time you spent, and what you were working on, on your Time Accounting page. *Your lab book, Time Accounting and Self Assessments will be checked weekly and are part of your RSS grade.* We will also use time accounting information to adjust the lab time demands over this term and in future terms. We will use both time accounting and self assessment outside RSS to justify the subject for ABET (national accreditation).

### **Our Rant on Physical Units**

Just a reminder that we will use MKS (meters, kilograms, seconds, radians, watts, etc.) units throughout the course and this lab. We insist that you do so as well. In particular, this means that *whenever you state a physical quantity, you must state its units.* Also show units in your intermediate calculations.

## Preliminaries: Self Assessment and Lab Rules

Make a dated entry called “Start of Lab 1” on your Assessment page. Before starting the lab, answer the following questions:

1. **Robot Hardware:** How familiar are you with the following: (1=Not at all proficient; 2=slightly proficient; 3=reasonably proficient; 4=very proficient; 5=expert.)
  - the MASlab ORCboard
  - Bump sensors
  - SBC and how it networks with a RSS Sun workstation
  - Motors and encoders
  - 80-20 assembly system
  - Ultrasound sensors
  - A robot camera
  - using lead acid batteries and charging them safely
2. **Version Control:** How comfortable are you using a Version Control system? (1=Not at all; 2=slightly comfortable; 3=reasonably comfortable; 4=very comfortable; 5=expert.)
3. **WIKI creation and use:** How familiar are you with using a WIKI (1=Not at all proficient; 2=slightly proficient; 3=reasonably proficient; 4=very proficient)?

Familiarize yourself with the lab rules which are the same as for RSS I and at <http://courses.csail.mit.edu/6.141/spring> as the first item.

## 1 Rebuilding your Robot

The RSS I robots from last term have been largely stripped down in preparation for RSS II. Each team should have a similar set of resources to rebuild their robot. Claim a robot and its home (big black tool box) and dive in. Check the motor mountings. Mount and wire the ORCboard, SBC, camera, and bump sensors, battery neatly while labeling cabling. This term we likely won't be immediately adding light or ultrasound sensors. Not all robots will have manipulators. For now, mount these sensors where it seems sensible.

(Re-)Familiarize yourself with:

- The robot and development system block diagram at:  
<http://courses.csail.mit.edu/6.141/spring2005/Labs/index.html> Lab 1, Handout 1-B:
- The Single Board Computer (SBC) datasheet in the list of Lab 1 Datasheets at:  
<http://courses.csail.mit.edu/6.141/spring2005/Labs/index.html>
- The specifications of the motors and encoders. They are available as Lab 2 Handouts 2-G and 2-H at:  
<http://courses.csail.mit.edu/6.141/spring2005/Labs/index.html>
- Operation and calibration of the motors and encoders. Use `orcd` and `orcspy` to test your motors and encoders. You will find these binaries on your local sun machine in the `/users/group-n/rss/bin` folder. The ORC manual is available at: <http://maslab.lcs.mit.edu/2005/orcmanual/orcmanual.pdf>.

Here are some steps given in RSS I, Lab 2 on how to characterize the encoders:

1. Hook up one motor on MOT0 which is logically designated the left motor in the software. Hook up the other motor to MOT2. It is logically designated the right motor in the software.

2. We have soldered the headers to your encoders for you. Hook up the encoder for the left motor (in MOT0) to J8 of the QPDS inputs. Take care to orient this connector correctly. Black is negative (ground) and red is positive. Repeat for the right motor, placing its encoder into J9, again being careful about the polarity. Do not power on your board until you have visually verified the polarity of these connectors with the example setup.
3. Look through the ORC Manual. Start the `orcd` executable. **(We have added the location `/users/group-N/rss/bin` to your shell's `$path` variable; see your home directory's `.bashrc` file.)** Familiarize yourself with the OrcPad interface. Use the white menu button to bring up the Diagnostics screen. Use the black button (aka the stick) to select menu options. If you press the red button, you will have to press the stick button to resume.
4. Calibrate the joystick on the OrcPad by holding the menu button down for 5 seconds. This brings up a calibration screen. The battery should be supplying 12 volts (check this on the OrcPad display).
5. Set up the configuration. Set the baud rate to 115.2 Kbps.
6. Put the OrcPad in drive mode and test your motors manually while observing PWM, current and joystick position.
7. Plug in the serial cable between the OrcBoard and the Sun workstation.
8. From the Sun Workstation, run `orcspy`. Familiarize yourself with its graphical user interface (GUI). **The `orcspy` executable is in `/users/group-N/rss/bin` (again, this should be in your path).**
9. Configure `orcspy` to read from the encoders (set pins 16-19 to Quad Phase Fast).
10. In the rightmost column of menus are the motor displays. They have a pink background. The top slider controls PWM. The plot shows current and PWM. The bottom slider controls slew. Use the sliders to change the slew and PWM values.

- Battery safety. See Handout 2-I at:

<http://courses.csail.mit.edu/6.141/spring2005/Labs/index.html>

- The camera is a Logitech QuickCam Pro 4000. The manual for the camera is available at: <http://mapleleaf.csail.mit.edu/nick-roy/QuickCam.pdf> and the linux driver we have been using is available at <http://www.smcc.demon.nl/webc>.

- In Lab 2 we will be using CARMEN and providing a quick review of it. To bone up on this topic if it's totally new to you, read:

<http://courses.csail.mit.edu/6.141/spring2005/Labs/releases/Lab5/handouts/carmenguide>.

## 2 Setting Up Your Development Tools

### 2.1 Familiarize Yourself with Subversion Version Control

Working in teams to develop and test algorithms introduces another level of complexity into software engineering. Software version control tools have long been used in industry (and, more recently, open-source communities) to allow many developers to develop code cooperatively. Such software associates a version number and other metadata to every file, enabling code changes to be tracked, merged and reversed. Version control software generally prevents developers from stepping on each other's toes as they simultaneously access a common code base. Good version control tools (such as Subversion) also provide helpful means of to view the history of, and compare different versions of, source files. We have adopted the open-source "Subversion" (`svn`) tools as the RSS software version control system and used it in RSS-I, Spring 2005.

Re-familiarize yourself with the *Reference Guide to Subversion*. It is a comprehensive primer to the most frequently used `svn` commands. It is at:

<http://courses.csail.mit.edu/6.141/spring2005/Labs/svnguide.html>

If SVN is new to you, make sure that you have a cheat sheet that tells you how to check out a snapshot of a repository, how to make an update and commit it, and resolve a conflict.

## 2.2 IDE or Code Editor

Install on the Sun workstation *NICK: and laptop?* an IDE that you're comfortable using for RSS code development (in Java). Alternatively, decide upon an editor such as emacs. Anticipate that you will be coding with interfaces and libraries provided by others so you will have to manage a large code base.

## 2.3 Document Preparation Tool

To publish information to others, you will use a wiki (see below). The information may be in HTML, as source code or as a PDF documents. Make a plan on how you will generate PDF documents and install on some system you regularly use an appropriate set of editors (eg an HTML editor, OpenOffice Word, LaTeX).

## 3 Setting Up WIKI for Documentation

To publish information to others, you will use a wiki. We have set up different groups in one wiki. It is a pmiki. The access id is `rss` and the password is `rss-f05`. From that homepage there is a separate set of group pages for each team. (A group is a pmwiki concept for modularizing subsets of pages.) Everyone in the class has `r/w` access to every team's group of pages. Respect that you should only *write* to your own team's pages while recognizing that everyone else in the class can *read* your team's pages. *Recall, as mentioned in the discussion of grading given in the first lecture, every week a staff member will browse your wiki and give a grade to its material.*

The wiki is located at <http://projects.csail.mit.edu/rss/wiki>. There are 5 team userids: `rss-team1`, `rss-team2`, `rss-team3`, `rss-team4`, `rss-team5`. The passwords are the same as the user ids. Everyone must login to read or write these pages.

As a first step, each team should login and change their password using the blue preferences choice that appears on the uppermost line of each wiki page after they have successfully logged in. Next, add your names and contact to the wiki. Finally, add a group picture with your robot to the wiki.

**Time Accounting and Self-Assessment:** After completion, return to the Time and Assessment pages of your lab book. Tally your total effort there. Answer the "Self Assessment" questions again. Add the date and time of your post-lab responses.