

University of Pennsylvania
ESE 112: Introduction to Electrical & Systems Engineering

Lab 5: Boe-Bot Navigation with Whiskers (Tactile Switch)

Objective

- To emulate the electro-mechanical mouse from Lab 1
- To introduce working with sensory information
- To write your own class to create a code base for future work

Introduction

A variety of tactile switches are used in robotic machinery. For example a robot can be programmed to pick up the object and place it elsewhere. The switch provides inputs that dictate some other form of programmed output.

Background

Electro-mechanical Mouse:

Recall from lab 1 the design of wall-hugging mouse. In this lab, you will emulate the same behavior i.e. make the mouse travel straight avoiding obstacles along its path. However, the algorithm is a programmable solution rather than a hardwired solution. The Boe-bot in this lab will act as mouse. Through a tactile switch (whisker), the Boe-Bot will sense obstacles and take appropriate action.

Java:

We will continue to work with java syntax and work towards building a code infrastructure that will be useful for your future labs.

Materials

- Boe-Bot unit with Javelin Stamp
- 4 AA batteries
- USB cable
- Miscellaneous Parts (from Boe-Bot kit)
 - 2 Whiskers
 - 2 Size #4 nylon washers
 - 2 Plastic spacers
 - 2 LEDs (red)
 - Various Resistors
 - 220 Ω (4)
 - 10,000 Ω (2)
 - 2 3-pin headers (will be used as switches)
 - Wires
- Javelin Stamp IDE



Figure 1: Parts from Boe-Bot kit

Pre-Lab Questions

1. Whisker Assembly

First, before writing your program, you need a circuit to program. To make your time in lab easier and more worthwhile, you will build the whisker assembly and circuit configuration before you come to lab.

To build the whisker assembly, you will need the following items from your Boe-Bot kit: screwdriver, 2 whiskers, 2 spacers, and 2 washers. These are the steps to install your whiskers (refer to Figure 2) – Note: Steps 1-3 may already be completed on your Boe-Bot, with the exception of the nylon washers:

1. Remove the two front screws that hold your board to the front standoffs.
2. Thread a nylon washer and then a $\frac{1}{2}$ " round spacer on each of the $\frac{7}{8}$ " screws.
3. Attach the screws through the holes in your board and into the standoffs below, but do not tighten them all the way yet.
4. Slip the hooked ends of the whisker wires around the screws, one above the washer and the other below the washer, positioning them so they cross over each other without touching.
5. Tighten the screws into the standoffs.

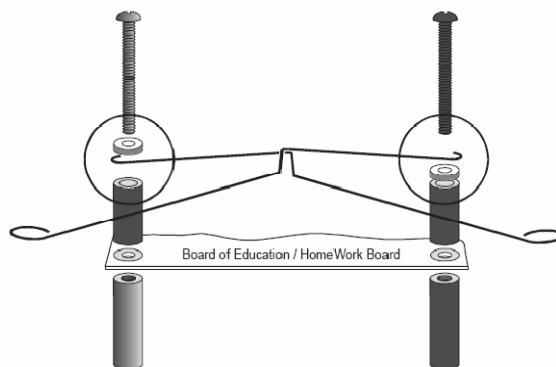


Figure 2: Whisker Assembly Diagram

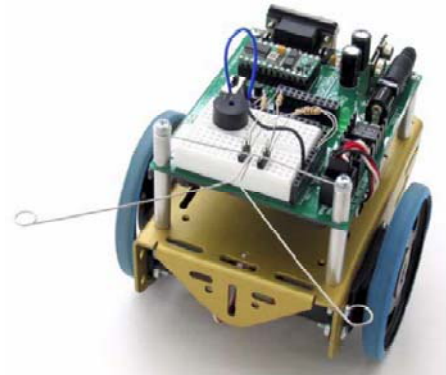


Figure 3: Boe-Bot with Mounted Whiskers

2. Sensor Circuit

Now, you will build the circuit according to the following circuit diagrams (Figures 4 and 5). You may connect them however you desire (i.e. doesn't matter which contact on the breadboard), however the circuit in its whole must still be the same. For example, you still need to have the current go from Vdd to LED to resistor to pin10, but you can have as many wires in the middle as you need to make sure the components do not come into contact with each other. Also, there are four independent circuits (2 LEDs and 2 whiskers). Make sure you don't inadvertently interconnect them (i.e. connecting parts from two circuits onto the same contact line in the breadboard).

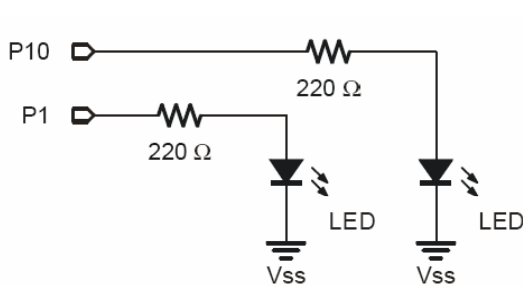


Figure 4: LED Circuit Diagram

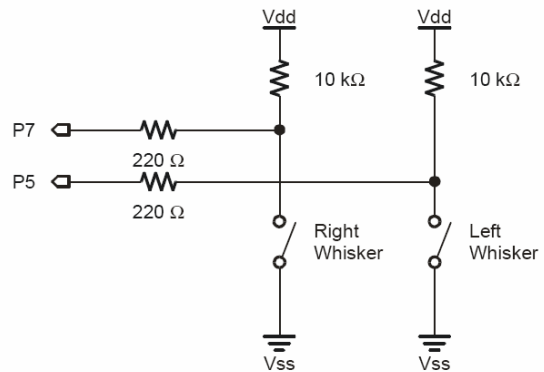


Figure 5: Whisker Circuit Diagram

Now that you have the circuit designed and the whiskers installed, you need to adjust the whiskers so that they are close to, but not touching, the switches. If necessary, you can change the position of the whisker to above or below the washer. You can also adjust the plastic part holding the three pins of the switch together. CAREFULLY take the 3-pin header and hold it on a table with the pins pointing straight down. Push, again carefully, on the plastic part to move it down. Then, insert the header back into the breadboard. In the end, the whiskers should touch the switches when pressed, and bounce back when you let go.

Lab Instructions

1. You should already have the circuits necessary for this lab built on your breadboard from pre-lab. Have your circuit checked by a TA.
2. Write a test program that will test the whiskers by making the respective LED light up when the whisker is pressed. You will use this program in your final whisker navigation program as well.

This actually easier than it sounds, if you know how the circuit works. When the whisker is not pressed, the respective pins (5 and 7) register a 1 (i.e. true in Boolean notation), since the voltage going through the circuit is 5V. Once the whisker is pressed however, the circuit is grounded making the voltage registered 0V, and thus making the value of the pin 0 (i.e. false in Boolean notation). You can use this knowledge to program the LED pins to turn on or off.

Hint: you already wrote a basic LED program in the Orientation Lab. You will just need to add something to make it turn on under a certain condition (in this case, when the whisker is pressed).

Once you have written the program, compile and download it to the Javelin Stamp. Try pressing a whisker and see if the LED lights up. If the correct LED lights up for the appropriate whisker, congratulations! If it doesn't, go back to your program and go through it yourself as if you were the computer. Try different variations (left whisker pressed, not pressed) and see where the problem is.

3. Now that your circuit is working, write a program that will make the Boe-Bot travel forward in a straight path, until it comes in contact with an obstacle. Then, according to which whisker is pressed, it should move so that it can continue on its journey. You will have to decide how you want the Boe-Bot to react when it hits something. The appropriate LED should light up as well. If it hits something ahead of it, both should light up.

Important Program Guidelines:

- a. You will work with two files: **Navigation.java** and **WhiskerNavigation.java**.
- b. The *Navigation* class will contain global variables and methods that can be reused (or added to in the future).
 - You may use code from your Boe-Bot navigation (Lab 4) but remember that we don't really care about acquiring specific distance anymore
- c. The *WhiskerNavigation* class should contain at least the main method and any other method you need to make the navigation task materialize. If it's a generic Boe-Bot method however, you should place it in the *Navigation* class.
 - In-order to use method implemented in Navigation class, you need add the folder path to the JSIDE's list of classpaths by going to Project > Global Options in the JSIDE and then clicking on "..."/> under Classpath. Search for your working directory (i.e. the directory

containing your java programs) and then add it to classpath. Make sure you click the 'add' button after selecting the path.

Note: You will have to include something in your program to make the Boe-Bot continue to navigate even if it senses both whiskers being pressed at the same time (i.e. both whiskers sense obstacle at the same time). The behavior in this case is open-ended.

Post-lab Questions

1. This lab is similar in design to the mouse lab from the first week. What similarities and differences did you note between the two? Which one, mouse or Boe-Bot, do you think did its task better and why?
2. When a whisker is pressed, what happens to the I/O pin monitoring it (see Figure 5)? Explain using circuit theory (i.e. the design of the circuit, circuit elements, etc.).
3. Provide brief overview of the global variable(s) and method(s) written in Navigation class.
4. Describe your WhiskerNavigation program/algorithm using a flowchart.
5. What were some problems you encountered while you were programming the Boe-Bot, and how did you resolve them?

Submit all of your Java programs from to Blackboard Digital Drop Box in one zipped folder using the format on the ESE112 website under the Course Information section. Only **one** submission per group is required. Make sure that your files mention the person who you are working with (at the very beginning of the .java file).

Note: All figures courtesy of Parallax, Inc.