Learning Roomba: Teacher’s Guide
Module 4 Sensors and Actuators

Drew Housten (dhousten@gmail.com)
1 Introduction

Sensors and actuators are necessary to make a robot intelligent. Without sensors, the robot operates blind and has no way to know what is going on in the environment. Without actuators, the robot cannot do anything. The third module introduced one type of sensor, the wheel encoder, and one type of actuator, the drive motors, without making the student aware of it. This Module will explain what a sensor and an actuator are and how they can be used. It will also discuss some other types of sensors and actuators used on robots and then the sensors and actuators specifically available on the Roomba. Behavior-based robotics and Sense-Plan-Act (SPA) architectures will be briefly introduced. Finally, the Module will discuss the code interfaces for the Roomba and provide several exercises to become familiar with using the Roomba’s sensors and actuators.

2 Educational Merit

This Module will allow the students to creatively think about solutions to problems and start to think logically. The programs written by the students will need to make decisions about the situation and act accordingly. Programming control structures, such as if statements and loops, will be necessary. Because of this, it would be advantageous for the students to have some prior understanding of programming control structures and some level of logic understanding. This Module is quite lengthy and may need to be split into several sessions. The students should have ample time to experiment and test the exercises in this Module.

3 Topics Covered

The following topics are covered by this Module:

- What is a sensor?
- Types of sensors
- What is an actuator?
- Types of actuators
- How can sensors and actuators be used?
- Roomba’s sensors and actuators
- Behavior-based robotics
- Sense-Plan-Act systems
- Sensor and actuator code interfaces
- Exercises
4 Exercise Solutions

Exercise 1 Program: This program will have the Roomba drive forward until a virtual wall is detected. Once the wall is detected, the Roomba will turn around and drive forward until a bump sensor is hit. As soon as that happens, the roomba will stop.

```java
import roomba.roombanetwork.services.userservice.*;

public class MyRoombaProgram{
    public static void main(String [] args){
        UserService.setServerAddress("localhost");
        UserService.setName("Your Name");

        Roomba roomba = new Roomba();
        roomba.forwardSpeed(.3);
        roomba.waitForVirtualWall();
        roomba.turn(180);
        roomba.forwardSpeed(.3);
        roomba.waitForBump();
        roomba.forwardSpeed(0);

        UserService.disconnect();
        System.exit(1);
    }
}
```

Exercise 2 Program: This program will have the Roomba spin until it has been picked up. As soon as that happens, the motors will stop and the Roomba will play a song.

```java
import roomba.roombanetwork.services.userservice.*;

public class MyRoombaProgram{
    public static void main(String [] args){
        UserService.setServerAddress("localhost");
        UserService.setName("Your Name");

        Roomba roomba = new Roomba();
```
roomba.turnSpeed(.3);

// Wait until the wheels drop
while (!roomba.getLeftWheelDrop() &&
  !roomba.getRightWheelDrop() &&
  !roomba.getCenterWheelDrop()){
  try{
    Thread.sleep(10);
  } catch (Exception e){
  }
}

roomba.turnSpeed(0);

// Play the song
roomba.clearSong();
roomba.addSongNote(39,32);
roomba.addSongNote(40,16);
roomba.addSongNote(0,16);
roomba.addSongNote(39,32);
roomba.addSongNote(40,16);
roomba.addSongNote(0,16);
roomba.addSongNote(39,16);
roomba.addSongNote(40,16);
roomba.addSongNote(0,16);
roomba.addSongNote(39,16);
roomba.addSongNote(40,16);
roomba.addSongNote(0,16);
roomba.addSongNote(39,16);
roomba.addSongNote(40,16);
roomba.addSongNote(0,16);
roomba.addSongNote(39,16);
roomba.addSongNote(40,16);
roomba.addSongNote(0,16);
roomba.playSong();

UserService.disconnect();
System.exit(1);

Exercise 3 Program: This program will have the Roomba drive forward until it detects something. If an object is detected through the bump sensors, the Roomba will turn around and drive forward for 1 meter. If a virtual wall is detected, the Roomba will turn 90 degrees to the right and stop.

import roomba.roombanetwork.services.userservice.*/

public class MyRoombaProgram{
public static void main(String[] args) {
    UserService.setServerAddress("localhost");
    UserService.setName("Your Name");

    Roomba roomba = new Roomba();

    roomba.forwardSpeed(.3);

    while (!roomba.getLeftBump() ||
           !roomba.getRightBump() ||
           !roomba.getVirtualWall()){
        try{
            Thread.sleep(10);
        } catch (Exception e){
        }
    }

    if (roomba.getLeftBump() ||
        roomba.getRightBump()){
        roomba.forwardSpeed(0);
        roomba.turn(180);
        roomba.drive(1);
    } else if (roomba.getVirtualWall()){
        roomba.forwardSpeed(0);
        roomba.turn(90);
    }

    UserService.disconnect();
    System.exit(1);
}

5 Homework Solutions

The homework for this Module asks the student to write some pseudocode for a program that will make the Roomba drive forward until it detects a wall, cliff, or other object. If so, the Roomba should stop.

1. Initialize new Roomba object

2. Command the roomba to drive forward at speed 0.2
3. While (no wall, no cliff, and no bump), sleep for a small time, and then loop

4. Command the roomba to drive forward at speed 0

5. Disconnect from the roomba

6 Related Resources


_Vehicles: Experiments in Synthetic Psychology_ is an excellent book explaining a behavior-based approach to intelligence. It is fairly easy to read and would be a great companion resource.


Chapter 3 of *Robotics Primer* discusses sensors and actuators as well as what is necessary for the robot to be autonomous. The rest of the book goes into more detail on using the Roomba’s sensors and actuators.


_Hacking Roomba: ExtremeTech_ covers the Roomba’s sensors and actuators and how to use them.


_Artificial intelligence: a modern approach_ provides additional background on planning techniques and artificial intelligence approaches.


_Introduction to Ai Robotics_ provides some more background material on behavior based robotics and planning approaches.