

Duke/DPS Robotics Program

10/29

Lesson Plan 3

Goal: *Create a robot that follows a line.*

Materials Needed: *Brookbot with light sensor.*

Part 1: Preliminaries

A. *The Light Sensor*

First, explain the light sensor to your team. Recall that the light sensor returns a value between 0 and 100, where a larger number corresponds to more light. Point out the red light on the sensor and discuss its use. Remember to talk about ambient vs. reflected light. This would also be a good time to talk about the NXT's view menu, as your team will be using it extensively for this assignment.

B. *Line Following Theory*

To us, line following seems extremely simple, because our eyes and brains make it extremely intuitive. For a robot it is more difficult. Have your team brainstorm how to get a robot with a single light sensor to follow a black line on a white track. Make sure they break it down into a step by step process. They should come up with something roughly equivalent to what we have already done in class: the wiggling method, where the robot turns one way when on the line, and then turns the opposite direction when off the line.

C. *Loops, Conditionals, and Sensor Input*

Loops, conditionals, and sensor input are all necessary components for a line follower. These topics were covered last week, but your team may need a refresher. Consult last week's lesson plan for help with these.

Part 2: Line Following

We have all the programming tools we need to write a line follower from last week's bumper cars. Therefore we can jump right in.

A. *Construction*

If your robot doesn't already have a light sensor on it, designate someone to mount one low on the Brookbot, facing toward the floor. This should be a quick and easy construction task.

B. *Programming*

Your team now has all the tools it needs to write a line following program. Cut them loose, guiding them along when they need help. Remind your team to start by configuring motors and sensors. When configuring the light sensor, they should choose "Light Active," as this makes the light sensor emit the red light to cause reflection. Also remind your students to use the NXT's view menu to determine what an appropriate threshold value is. Remember that an appropriate threshold is roughly halfway between your average light and average dark values. Below is a sample solution. Note that the code depends on which side of the line you start the robot on.

```
task main()
{
  while(true)
  {
    if(SensorValue[S1] > 40)
    {
      motor[motorA] = 0;
      motor[motorC] = 50;
    }
    else
    {
      motor[motorC] = 0;
      motor[motorA] = 50;
    }
  }
}
```

If you have extra time after your team has produced a working line follower, have them optimize the robot to do it as fast as possible without compromising too much accuracy. Alternatively, discuss the theory of two-sensor line following in preparation for next week.