CMPE-110
Intro to Computer Engineering
Fall 2019
Laboratory Exercise Nine
Robot Sensors

This laboratory exercise a) investigates the use of bump sensors to determine the presence of an obstacle in front of the robot; and b) investigates the use of reflective sensors to find a line. The programs are modified to perform various motions and avoid obstacles in the robot's path.

Robot Usage

A few key reminders:

1) This robot will be with you for the next several weeks. Treat it with care!
2) Ensure that the board is powered off (blue led is off)
3) Please be careful plugging in the USB cable to avoid damaging the microcontroller board.
4) If you do not know what code is in the robot, you want to be sure you are holding the robot (securely and off any surface) when you power on the robot.
5) While writing code (i.e. the robot is sitting on the workbench), please turn the power to the robot off so the robot does not drain the batteries.

Obstacle Detection

Your robot is fitted with six snap-action bump sensor switches. The microcontroller can detect whether each individual button is depressed in order to detect an obstacle.

Each bump sensor is tied to a specific input pin on the microcontroller. The state of these pins may be accessed using the provided array, bump_sw, in the provided code. The state of a bump switch can be checked using bump_sw[n].read(), where n is an integer 0 through 5. If read() returns 0, the bump switch is currently pressed. If read() returns 1 the bump switch is not pressed.

During this exercise you will determine if any of the bump switches are pressed and turn on an LED corresponding to the pressed switch.

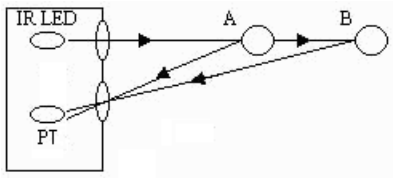
Do the following:

1. Start the Arduino program. We should find in in the Window's Program list, or search for it.
2. Select menu item File→New.

3. Replace all the default code that appears in this new sketch with the code from the Lab9_Sensor.ino.txt file on myCourses (Ctrl-A will select all the code, then hit Delete).
4. Select menu item File→Save As… menu item. Navigate to a folder on your thumb drive in which you want to create the new sketch and select that folder.
5. Enter Lab9_Sensor for the File name and then click Save.
6. Click the Verify button, then the Upload button to load the sketch onto the robot.
7. Open the Serial Monitor.
8. Press down the left most bump sensor. The red LED should light and 'Switch 5 pressed' should print to the serial monitor. Note that the red LED turns off when the switch is released.
9. Modify the function testBump() to do the following:
    a. Each bump switch should print a message to the serial monitor when it is pressed e.g. 'Switch 4 pressed'
    b. The left bump switches should turn on the red LED when pressed and turn the red LED off when released
    c. The right bump switches should turn on the green LED when pressed and turn the green LED off when released

Line Sensing

The robot has 8 IR reflective sensors. This sensor consists of an infrared (IR) LED and an IR sensitive photo transistor. The LED emits a beam of infrared light, and an internal transistor measures the amount of IR light reflected back. White surfaces reflect more IR light than black surfaces.



The more IR light that is reflected back, the lower the analog output voltage. The sensors are connected to analog to digital converters (A2D). The A2D will convert the analog signal, 0V to 5V to a digital count, say 0 to 2500. As such, we are quantizing our continuous signal to one of 2500 bins during this digitization process. Digital values close to 0 represent low voltages while digital values close to 2500 represent high voltages. Code has been provided reads all 8 sensors and averages the values into three sections: left, right, center. Take a look at the testLine() function and try to understand how it works.

You will use the same program to test the line sensors. Do the following:

1. In the loop() function, comment out the testBump() line, and uncomment the testLine() line. Your program should look like this:

```
void loop()
{
  pressToStart();          // wait for 'Start' button to be pressed

  while (true)
  {
    // uncomment the following line to run testBump
    //testBump();

    // uncomment the following line to print the line following
    testLine(0);

    // uncomment the following line to run driveUntilBump
    //driveUntilBump();
  }
}
```

2. Click the Verify button, then the Upload button to load the sketch onto the robot.
3. Open the Serial Monitor.
4. When placed over a consistent color (e.g. a plain white sheet of paper), all three outputs should read approximately the same value.

5. On your worksheet, fill in the sensor value that is displayed on the Serial Monitor, for each item/location on your worksheet. This does not need to be exact, just pick the 'average' number you see.
6. Select a threshold value that would be good for detecting a white line and enter that value on your worksheet.
7. Update the parameter to testLine() with your threshold value.
8. Click the Verify button, then the Upload button to load the sketch onto the robot.
9. Open the Serial Monitor.
10. As you pass the white paper under the sensors, you should see the words 'left'/'center'/'right' change from upper to lower case. When you have this working, demonstrate it to your instructor/TA to get your 'Senses white paper' worksheet line initialed.

Collision Detection

You will now use the bump sensor to stop your robot once it detects a collision. Continuing with the sample code, you will write the driveUntilBump() function. Uncomment driveUntilBump() in the loop() function and comment out testBump() and testLine().

Replace the pseudo code in driveUntilBump() to actually drive and stop the robot. The robot must not drive until the 'Start' button is pressed. The code must not return from the driveUntilBump() function until the robot has stopped.

Once you have confirmed the robot stops when it collides with an object, demonstrate it to your instructor/TA to get your 'Collision Detection' sketch worksheet line initialed.

Submission

One per team—*make sure team member names are written on each item:*

___ Completed laboratory worksheet.
___ A printed copy of portion of the 'Collision Detection' Arduino sketch that you changed (e.g. the file comment block and driveUntilBump() function; no need to print the code that was provided to you); also, do not forget to comment your code.
___ Make sure the robot is turned off
___ Robot was put away.