Lab 11 - Obstacle Avoidance

1 Collaboration Policy.

This is a team-of-two or individual laboratory. You may use any of the authorized resources listed below. DO NOT copy anyone else's work.

Authorized Resources: You may use any electronic or hard copy resource at your disposal as long as 1) you cite your sources and 2) your use of the materials does not go against the intent of the assignment. For example, you can use a software library that you find online to help develop a project if you cite where you found it. However, you cannot complete your project by copying all of the source code, schematics, etc and simply running it on the required hardware.

2 Documentation.

Date:

Instructor who helped me:

Help received:

3 Objectives.

1. Become familiar with the Sharp GP2Y0A51SK0F Analog Distance Sensor.
2. Utilize the Arduino to detect walls.
3. Integrate the Distance Sensor with the DFECBot and program the DFECBot to avoid obstacles.

4 Materials.

1. 3x Sharp GP2Y0A51SK0F Analog Distance Sensor
2. 3x 3-pin Female JST Cables
3. USB Programming Cable
4. DFECBot

5 Introduction.

5.1 Sharp GP2Y0A51SK0F Analog Distance Sensor

The Sharp GP2Y0A51SK0F Analog Distance Sensor uses an infrared radiation (IR) reflectance sensor with an IR light-emitting diode (LED) and an IR sensitive phototransistor.[1] The sensors will be powered using the 5 V and ground pins on the Arduino.

Connect your three Distance Sensors to the mounts on the front of the motor layer (Layer B). Secure them using six $\#2-56$ x 1/4" bolts and nuts. Wire the sensors according to the schematic shown in Figure 1.

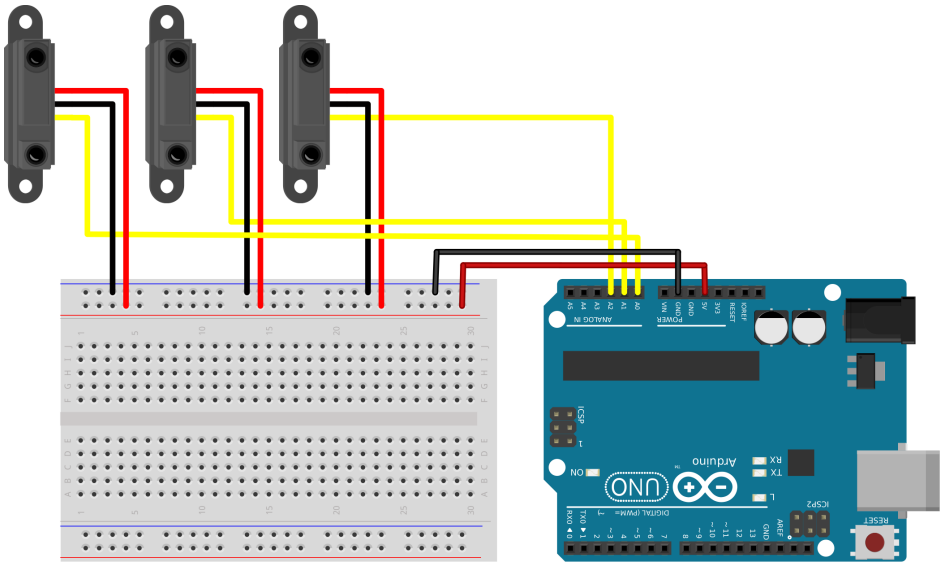

Figure 1: Sharp GP2Y0A51SK0F Analog Distance Sensor Wiring Schematic

5.2 Example Code

Browse to `K:\DF\DFEC\ECE210\Labs\Lab 11 - Obstacle Avoidance` and copy the `robot_obstacleavoidance` folder to your computer. Open `robot_obstacleavoidance.ino`. Opening the *.ino* file should open 3 files in your Arduino IDE: `robot_obstacleavoidance.ino`, `drive.h`, and `TB6612FNG.h`. The two *.h* files are the same header files used during previous labs and the *.ino* file is your Arduino sketch.

5.2.1 robot_linefollowing.ino

This example Arduino Sketch provides code to read the values outputted by the DFECBot's left Distance Sensor (see below example) and convert the value to a distance.

```
/*
* Function used to convert from analog voltage to distance.
*  Derived from graph in GP2Y0A51SK0F datasheet found at
*  https://www.pololu.com/product/2450/resources
*  The IR sensor is effective between 2-15 cm.
*/
float distCalc(float sensorVal){
  return 11.5402*pow(0.995796, sensorVal);
}
```

```
// read/convert/print value ouputted by DFECBot's left IR sensor
float ir_L = analogRead(irL);
float distL = distCalc(ir_L);
Serial.print("Left: "); Serial.println(distL);
```

[1] GP2Y0A51SK0F Datasheet, `https://www.pololu.com/file/0J845/GP2Y0A41SK0F.pdf.pdf`

6 Procedure

Use the example code provided, `TB6612FNG.H`, and `drive.h` to code the DFECBot to do the following:

1. Print the values from the DFECBot's center and right GP2Y0A51SK0F Analog Distance Sensor to the serial monitor.
2. Use a ruler to confirm the accuracy of each distance sensor - the sensor should be fairly accurate between 3 *cm* and 12 *cm*.
3. Program the DFECBot to detect an object within 4 *cm* in front of, to the left of, and to the right of the DFECBott using the three Distance Sensors.
   (a) Drive forward.
   (b) If the front sensor detects an object, randomly turn right or left.
   (c) If the front and left sensors detect an object, turn right.
   (d) If the front and right sensors detect an object, turn left.
   (e) If all three sensors detect an object, turn around.
4. **Optional** – Limit drifting within a maze.
   (a) If the left sensor detects an object (within 3 *cm*) and the front sensor does not, turn right slightly until the left sensor does not detect an object.
   (b) If the right sensor detects an object (within 3 *cm*) and the front sensor does not, turn left slightly until the right sensor does not detect an object.

   Hint: You may need to use the two functions in `drive.h` to make a slight right or left correction.