After you have completed the lesson on making the ActivityBot360 do turns, you should complete this lesson.

In your maze code you need to devise a method for deciding when to make a turn. The timing for executing a turn must be precise. If the robot turns too early or too late, it will crash into a wall of the maze. The methods for determining when the robot should turn can be categorized into two types **1)** measuring distance traveled **2)** sensing an open lane. In this lesson you will learn how to use the second method.

Let us take the example of the third turn in the maze, which is a right turn. For this turn you could use the right-facing Ping sensor. If you program the robot to make periodic measurements to the right, there will be an increase in the distance to a wall on the right as the sensor passes forward of the end of the wall of the lane. The sensor will be sensing a more distant wall due to the open lane to the right. You can take advantage of this change in distance to trigger the right turn.

Figure 1: 2025 NRC Middle School Maze

In this lesson you will not be programming the robot to make a run in the maze. Before you do that, it is a good idea to calibrate your Ping sensors, which you will do in this lesson. This will allow you to confirm that the sensors are operating properly and to compare distances reported by the robot with distances you measure with a ruler.

It is most convenient to measure the distance from the front surface of the sensor to the wall with a ruler. But the sound emitter and detector of the sensor are actually located inside the Ping sensor, not at its font surface. Therefore, for the sake of best accuracy of distance measurement, you should apply a correction. This will be the difference between the distance reported by the robot and the distance you measure with a ruler.

The code on the next page will allow you to calibrate your sensors. It takes sensor readings every 3 seconds and displays the values on your computer screen. With your robot connected to the computer, slide the power switch to position one, then click on the **Run With Terminal** button in SimpleIDE software . Once the terminal window opens on your computer monitor, slide the robot power switch to position two and the Ping sensors will begin sending distance measurements to your computer screen. Place objects at specific distances from the Ping sensors and compare distances reported by Ping sensors with distances you measure with a ruler. Record your results for future reference.

Keep in mind that the code below establishes two INTEGER variables for storing the distance measurements. Integers are whole numbers (nothing to the right of a decimal point). The robot software DOES NOT round numbers to the nearest whole number. It just drops any value on the right side of the decimal point. For example, a value of of 6.9 will be rounded to a value of 6, NOT 7.

```
// front-facing Ping connected to port 17, right facing Ping connected to port 16

// ------ Libraries
#include "simpletools.h"
#include "abdrive360.h"
#include "ping.h"

// ------ Variables
int distF;              //variable to store distance values of front-facing Ping
int distR;              //variable to store distance values of right-facing Ping

// ------ Main Program
int main()
   {                                    //start brace for int main()
   while(1)                             //repeat forever
      {                                 //start brace for while(1)
      distF = ping_cm(17);              //make a distance measurement to front
      distR = ping_cm(16);              //make a distance measurement to right
      print("distance front = ");
      print("%d", distF);
      print(" centimeters\n");
      print("distance right = ");
      print("%d", distR);
      print(" centimeters\n");
      pause(3000);                      //pause 3 seconds
      }                 //end brace for while(1)… go up to the while(1) and repeat
   }                                    //end brace for int main()
```

File name: DisplayingPing.pdf
online location: https://www.lafavre.us/robotics/DisplayingPing.pdf
last update: 3/20/26