Trigger a Turn with Ping sensor J. La Favre March, 2026

The image to the right is the maze pattern for the 2025 NRC Robot Maze Contest, Middle School Division. In a previous lesson you learned how to use odometry to measure the distance from the start to the point where the robot must make its first turn (a 90 degree turn to the left). In this lesson you will use the Ping distance sensor to determine WHEN the robot should initiate its first turn to the left.

Your robot is equipped with two Ping sensors, one pointing forward and one pointing to the right. You need to use the forward pointing sensor since you don't have a sensor pointing to the left.

Make sure your code includes the libraries **simpletools.h**, **abdrive360.h** and **ping.h**

```
#include "simpletools.h"
#include "abdrive360.h"
#include "ping.h"
```

Now create a variable to contain the distance value determined by the Ping sensor:

```
int distF; // variable distF will contain distance measurement to the front
```

Then add the main() function:

```
int main()
   {
     // all other code will go here
   } //end of main() function
```

In between the open and close curly { } braces, add your drive code as provided below.

```
while(1)                    //run this loop forever
   {
   distF = ping_cm(17);     // distance in cm to the front (port 17)

   if(distF < 19)           // if distance to front less than 19 cm
      {                           //turn left
      drive_speed(27, 64);        // this makes robot turn left
      pause(1350);                // turn left for 1350 milliseconds,
                                  // about 90 degree turn
      break;                      // escape from the while loop
      }                           // end of if()
   drive_speed(64, 64);           // drive forward at speed 64 both wheels
   pause (20);                    // wait 20 milliseconds
   }                              //end of while() loop
drive_speed(0, 0);                // stop the robot
```

Let's take a look at what happens inside the while(1) loop of the code. The first line instructs the robot to make a ping measurement to the front and store that value in the variable named distF. The next line contains an **if** statement (is the distance to the front less than 19 cm, if so this indicates there is a wall close to the front). If the distance is 19 cm or more then the three lines of code below the if statement are skipped and execution of the program continues with the line **drive_speed(64, 64).** The robot drives forward at speed 64 for 20 milliseconds. Then the loop repeats. Eventually the robot makes its way down the lane until the front-facing ping sensor is positioned less than 19 cm from the front wall. The robot begins to make a curved move to the left, drive_speed(27, 64). The right wheel is turning faster than the left wheel, which causes the robot to turn left. This turning movement goes on for 1350 milliseconds (1.35 seconds). Then the next line of code is executed: **break**. This causes the program to jump out of the while(1) loop. Then there is only one line of code remaining to execute, drive_speed(0, 0), which causes the robot to stop.

If your robot makes the turn without crashing into a wall, then you have finished this exercise. It may not be a perfect turn but that is OK.

A more refined code would include wall-following, which you can add later.

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