# FINAL PROJECT

# Lab Goals:

* Get the robot to move in the path of the maze independently, making sure it recognizes the obstructions on the wall and find it’s path around them.
* Change the tricolor LED based on the distance detected by the sensors.
* Display the distance traveled on the LCD screen.

# Steps Taken To Accomplish the Goals:

* The first step was to set up the LEDs based on the distance traveled. We called the led\_controller\_byte\_write function which tells us the register and data to write to the led. This controls the color being displayed. If statements control the distance being measured of all the sensors.
* For the auto-pilot movement of the robot we used the left hand rule. This part also takes care of recognizing obstructions on the wall.
* When the distance detected by the left sensor is between 9 and 11 (both inclusive) or the right turn boolean is being asserted, we check for three distance ranges and whether the boolean right turn has been asserted or not.
  + If the distance tracked by the center sensor is less than 7 inches, and right turn is not asserted, this implies that there is a wall on the left and in front, so the robot needs to turn right. We used driver functions for the turn and set right\_turn to true, so it keeps turning right until the center sensor does not see a wall.
  + A similar if statement is used if the distance of the center wall is between 7 and 9, which implies there is a wall ahead and the robot should turn right. Right\_turn is asserted to true, which means the robot should continue to turn right until it does not see a wall ahead.
  + If the center sensor senses a distance greater than 9, that means there is no wall ahead yet, which implies that the robot should continue driving straight ahead. (Thus following the left hand rule)
* If the left sensor is at a distance greater that 15 inches, it is drifting in the right direction. We have a counter that checks whether the robot has been drifting right for more than 3 cycles of entering the else if loop, the robot should turn left. Otherwise it should drive straight.
* If the value of the left sensor is between 11 and 15, the robot is drifting right and it should turn left.
* The last condition takes care of the robot being too close to the left wall. If the distance to the left wall is less than 9, then the robot moves right.

# Problems Faced and Steps taken to Improve:

* We had problems with our driver logic. At corners, after turning right, it failed to recognize the adjacent wall as the new left wall. It kept rotating in anticlockwise direction to find a left wall to hug to. This issue was solved by the counter logic, that if the distance is greater than 15 and it has not turned left in a while, it will turn towards the left wall.
* The other issue we faced is that we could not get the correct LED lighting for the right sensor. Even though we had the correct implementation for right sensors in the previous lab, somehow after debugging for a long time, we could not resolve this issue. However, the right sensor is not used in our implementation of left hand algorithm.

# Ready Reference:

* Left hand algorithm was used for the implementation of auto pilot mode.