**Line Follower Robot**

A-Line Following Robot is an autonomous robot which is able to follow either a black or white line that is drawn on a surface consisting of contrasting colours. It is designed to move automatically and follow the made plot line. The robot uses several sensors to identify the line, thus assisting the robot to stay on the track. The Arduino Uno interface is used to perform and implement algorithms to control the speed of the motors, steering the robot to travel along the line smoothly. This project aims to implement the algorithm and control the movement of the robot by proper tuning of the control parameters and thus achieving better performance. It can be used in industrial automated equipment carriers, small household applications, tour guides in museums etc.

**Maze Solver Robot, Using Artificial Intelligence with Arduino**

Once you have a robot with line following capabilities, the next natural step is to give it some degree of intelligence. So, our dear Robot will now try finding how to escape from a "maze" using the shortest and fastest way. The majority of mazes, however complex their design may appear, are essentially formed from one continuous wall with many junctions and branches. If the wall surrounding the goal of a maze is connected to the perimeter of the maze at the entrance, the maze can always be solved by keeping one hand in contact with the wall, however, many detours that may involve. Those ‘simple’ mazes are correctly known as "Simply-connected" or "perfect maze" or in other words, mazes that contain *no loops*.

Returning to our project, it will be split into two parts (or "passes"):

1. (First Pass): The robot finds its way out of a "not-known perfect maze". Doesn’t matter where you put it inside the maze, it will always find a "solution". In the first time that the robot explores the maze, it will waste a lot of time "thinking" about what to do at all intersections. Testing the numerous possibilities, it will take several wrong paths and dead ends, forcing him to run longer paths and perform unnecessary "U-Turns". During this "1st Pass", the robot will be accumulating experiences, "taking notes" about the different intersections and eliminating the bad branches.
2. (Second Pass): Once the robot found a possible maze solution, it should optimize its solution finding the "shortest path from start to finish". In it’s "2nd Pass", the robot goes straight to the end without any mistake or doubt, using the quickest path.

**Reference:-**

* [**https://www.instructables.com/id/Maze-Solver-Robot-Using-Artificial-Intelligence-Wi/**](https://www.instructables.com/id/Maze-Solver-Robot-Using-Artificial-Intelligence-Wi/)