

POZNAŃ UNIVERSITY OF TECHNOLOGY

FACULTY OF CONTROL, ROBOTICS
AND ELECTRICAL ENGINEERING



TERM DESIGN

MAZE TRANSVERSE ROBOT

PROJECT REPORT

KESHAV SHARMA, 142862

JAKUB SZUKAŁA, 135906

ZISHI LIU, 135019

INSTRUCTOR:

JAN WIETRZYKOWSKI, MSc

JAN.WIETRZYKOWSKI@PUT.POZNAN.PL

18-06-2021



Contents

1 Introduction 3

2 Specification 3

3 Schematic for Additional Sensors 4

4 Implementation 4

 4.1 Hardware 4

 4.2 Algorithm 5

5 Results 7

6 Conclusion 7

7 Summary 7

INTRODUCTION

The aim of this project is to build a robot that will transverse a maze as fast as possible using the AlphaBot-2ar platform.



Fig. 1. AlphaBot-2ar

Project Outline:

- **Platform:** AlphaBot-2ar(Fig.1)
- **Controller:** Arduino-uno.
- **Things to remember:**
 - We have to do this project without using floor line follower.
 - After finishing our robot, we need to test it on two different types of maze.
- **Maze:** Required two mazes one is simple and one is complex. Each maze with 18[cm]×18[cm] blocks(Fig.2).

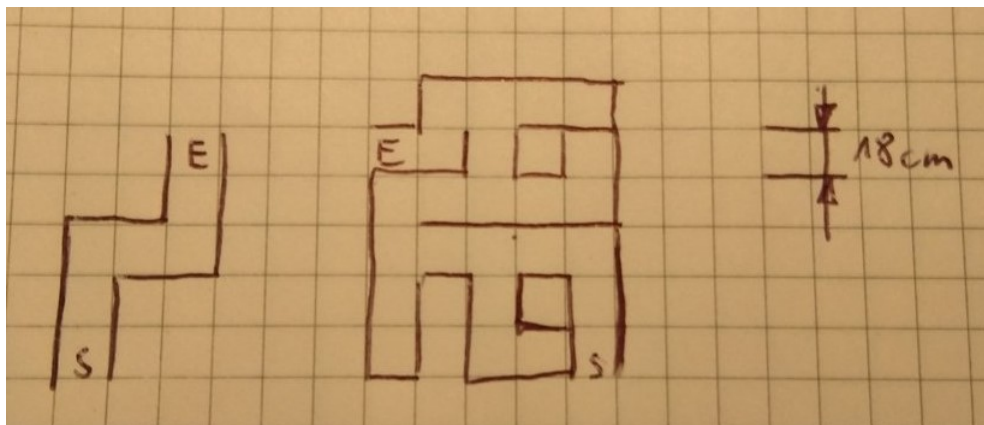


Fig. 2. Given designs for Mazes

SPECIFICATION

In term of hardware specifications we are using on board motors of Alphabot to drive and front IR and Ultra sound sensors for detecting obstacles in the front of the bot. Other on board sensors are useless for us because they specially designed for following line on the floor and we do not need that functionality. We added total three additional sensors on our bot. In terms of software, as we have Arduino uno in our bot, so we are using Arduino Ide to program our bot. For our testing purposes we made a simple maze and one complex maze for testing from cardboard.

SCHEMATIC FOR ADDITIONAL SENSORS

In the Fig.3, we have our schematic for additional sensors. We are using two SR04 ultrasound sensors and one ST1081 IR distance sensor.

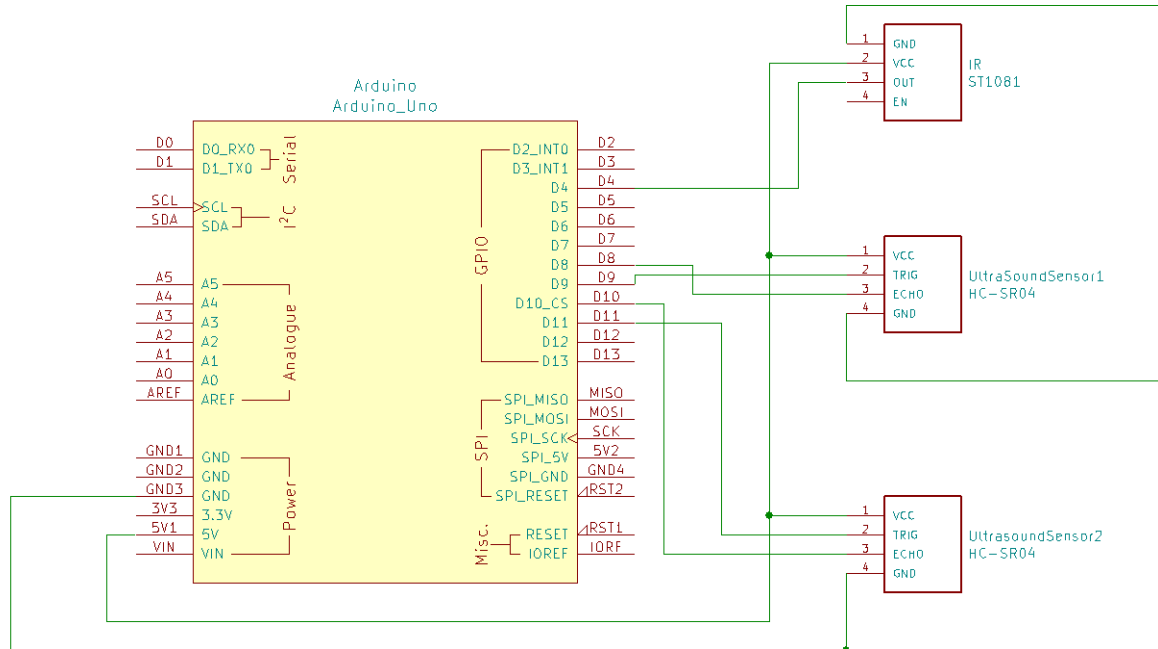


Fig. 3. Schematic for additional sensors

We are using pin 4 to catch output from our IR sensor, pin 8,9 for echo and trig of first ultrasound sensor and pin 10,11 for echo and trig of second ultrasound sensor respectively. All the additional sensors are connected with 5V power supply from Arduino board.

IMPLEMENTATION

4.1 HARDWARE

Alphabot-2ar is designed for transverse the maze by following the black line on the floor, but our project is to build a robot that will traverse a maze as fast as possible without using black line on the floor and for make this happen, we had to add some additionally sensors. We added two ultrasound sensors for taking distance from side walls(Left and Right wall) and one IR distance sensor for following right side wall.

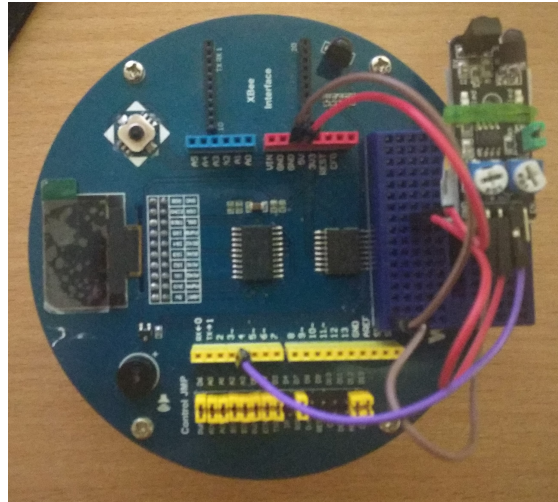


Fig. 4. AlphaBot with Additional IR Sensor Attached(Mid phase)

As we can see in the fig.4, we attached IR distance sensor at the right side ending of the bot. It gives information about the right side wall, it follows the right side wall all the time and whenever there is no wall it informs the bot.

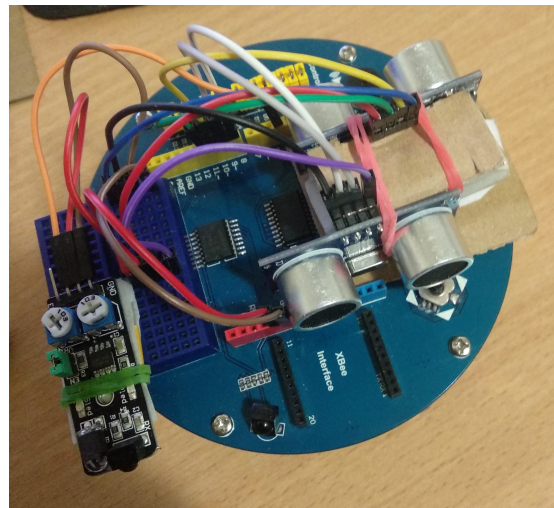


Fig. 5. AlphaBot with Additional Ultra Sound and IR Sensors Attached(Final phase)

We attached these two ultra sound sensors(fig.5) for giving us information about the side walls, Left and Right walls. Their primary function is to help the bot in driving straight but we are also using these sensors in our maze transverse algorithm.

4.2 ALGORITHM

We had to write an algorithm that will control the robot to solve a maze. For this we are using a wall following algorithm but with some changes to make it more reliable and more efficient. Basically we are using a hybrid wall following algorithm, which is combined with left turn algorithm.

Below, there is code for our algorithm.

Listing 1. code for our algorithm(wallFollower,ino)

```
01. // Our main alogrithm to transverse the maze
02. void wallFollower(int speedL, int speedR, int speed)
03. { LeftWall = Left_Wall_test();
```

```
04.   RightWall = Right_Wall_test();
05.   int IR_value = IR_reading();
06.   bool obstacle = obstacle_detection();
07.   if (obstacle || IR_value == 1)
08.   {
09.       if (obstacle && IR_value == 0 && RightWall < 10 && LeftWall > 10)
10.       {
11.           turn_left(speed);
12.       }
13.       if (obstacle && IR_value == 0 && RightWall < 10 && LeftWall < 10)
14.       {
15.           while (RightWall < 10 && LeftWall < 10)
16.           {
17.               LeftWall = Left_Wall_test();
18.               RightWall = Right_Wall_test();
19.               move_backward(speedL, speedR);
20.           }
21.           if (LeftWall > 10)
22.           { delay(150);
23.             turn_left(speed);
24.             delay(150);
25.             move_forward( speedL, speedR);
26.             delay(1000);
27.
28.             while (IR_value == 1 && !obstacle)
29.             {
30.                 IR_value = IR_reading();
31.                 obstacle = obstacle_detection();
32.                 move_forward(speedL, speedR);
33.             }
34.         }
35.     }
36.     if (IR_value == 1 )
37.     {
38.         turn_right(speed);
39.
40.         obstacle = obstacle_detection();
41.
42.         while ((IR_value == 1 ) && !obstacle)
43.         {
44.             IR_value = IR_reading();
45.             obstacle = obstacle_detection();
46.             move_forward(speedL, speedR);
47.         }
48.     }
49.     else {
50.         move_forward(speedL, speedR);
51.     }
52. }
53. else {
54.     move_forward(speedL, speedR);
55. }
56. }
```

Other files such as movement controller.ino, speedController.ino, main.ino etc, can be found [github](#)

Our bot first start driving in straight line while following right wall using additional IR sensor, and looking for the side walls and obstacles in the front. If our bot detect some obstacle in the front or our IR sensor did not find any wall on the right, it will trigger the first "if statement" and then check for the following conditions:-

- First Condition:
If there is an obstacle in front of the robot and there is no right wall it will turn left.
- Second Condition:
If there are walls on the both side and an obstacle in the front it will drive backward and follow the left wall until there is no wall on the left, after detecting left wall, it will take a left turn and for sometime it will move forward until there is no obstacles in the front.
- Third Condition:
While driving straight if there is no obstacle in the front and no wall on the right side, bot will take a right turn and move forward until there is no obstacles in the front.

RESULTS

When we tested our bot on the simple maze, it transverse the maze successfully but when we tested in the complex maze, it transverse the half of the maze but unable to dodge the side walls. We had to help the bot by hand to reach the end of the complex maze.

Test Results are available on our wiki page.

CONCLUSION

Our robot did not able to transverse complex maze perfectly but if we gave some help such as little push, it can transverse the maze and reach to the end. If our bot drive in straight line it can transverse the maze on its own.

SUMMARY

This project helps us to develop many new skills like team work, coping with different kind of problems and also it taught us a lesson that working on a real life project are way more different than simulated one. There are so many factors that we need to take care while working on a real life project rather than a simulated one. In simulated one we can easily achieve our goal but in real life, it is lot harder than looks.