CS-308-2014 Final Report

LiSa

Sashank Gondala, 120050050
P Bharath Kumar, 120050058
MS Krishna Deepak, 120050057
RS Bharadwaj, 120050056

Table of Contents

1. Introduction

The Wumpus world problem deals with an AI robot navigating its way through a maze to try and find gold. The robot must safely navigate its way around bottomless pits of death and evil Wumpus creatures to locate the gold hidden on the board. LiSa is a bot which tries to solve a variant of the Wumpus world problem

2. Problem Statement

The bot needs finds its way to a pre-set destination point in a given maze. We can place many barriers in its way. If the barrier is of color red, the bot takes shifts left and continues moving in the same direction. If the color is not red, the bot takes shifts left and continues moving in the same direction. The bot finally to reach the destination in spite of these obstacles.

3. Requirements

3.1 Harwdare Requirements

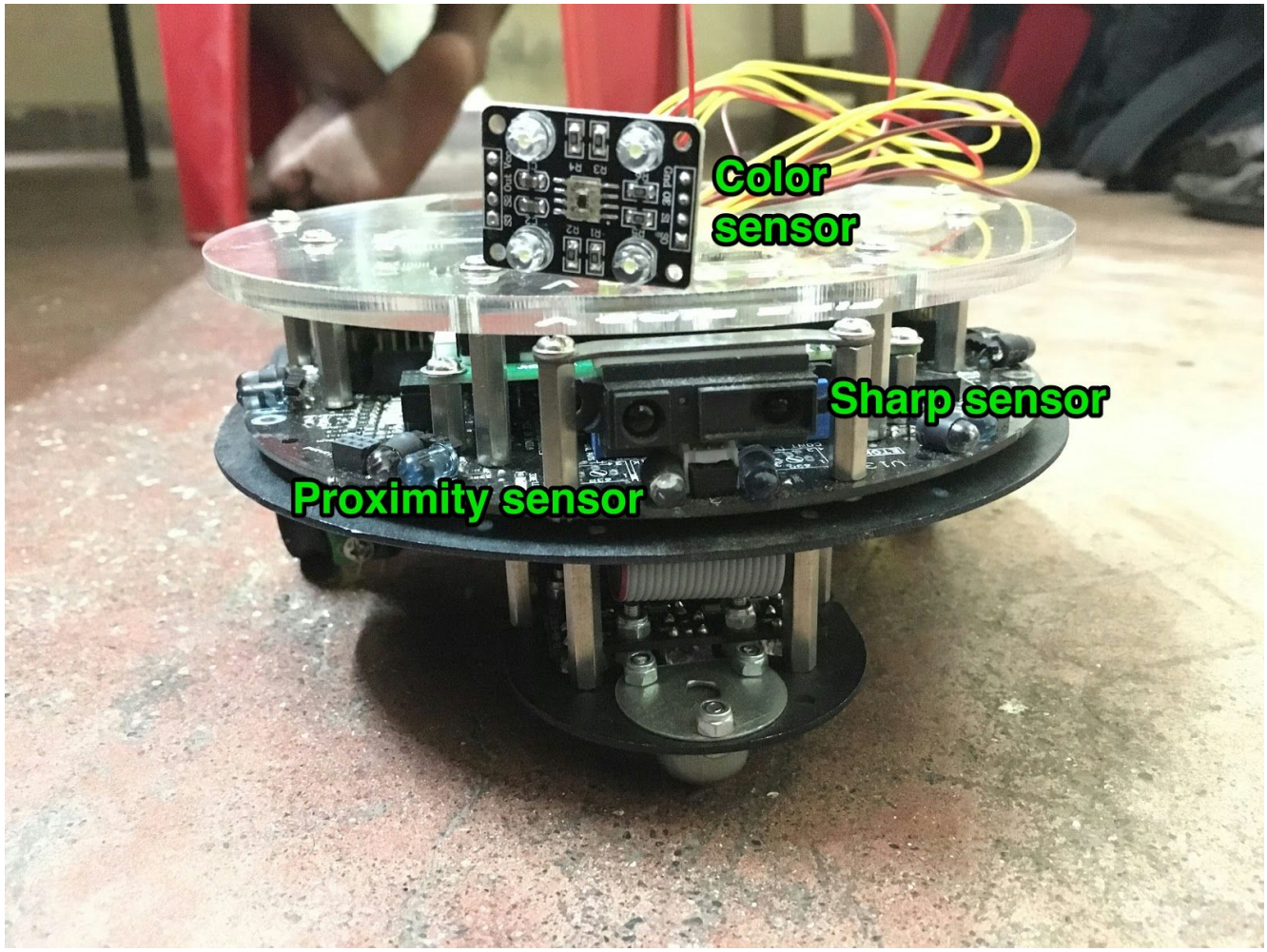
1) **Firebird:** Main motion
2) **Proximity sensor**: Useful to find the distance to the obstacle when the obstacle is closer to the firebird
3) **Sharp sensor**: Useful to find the distance to the obstacle when the obstacle is farther away from the firebird
4) **Color sensor**: Used to find the color of obstacle and act accordingly
5) **Colored Obstacles**
6) **Terrain**

3.2 Software Requirements

1) **Codeblocks**: To write code and build a hex file from it

4. System Design

We use a firebird, a color sensor, proximity sensor and a sharp sensor. Different components are labelled in the image below.



State diagram

State chart

State Description

s_R - Firebird is moving in right direction

s_D - Firebird is moving in down dir.

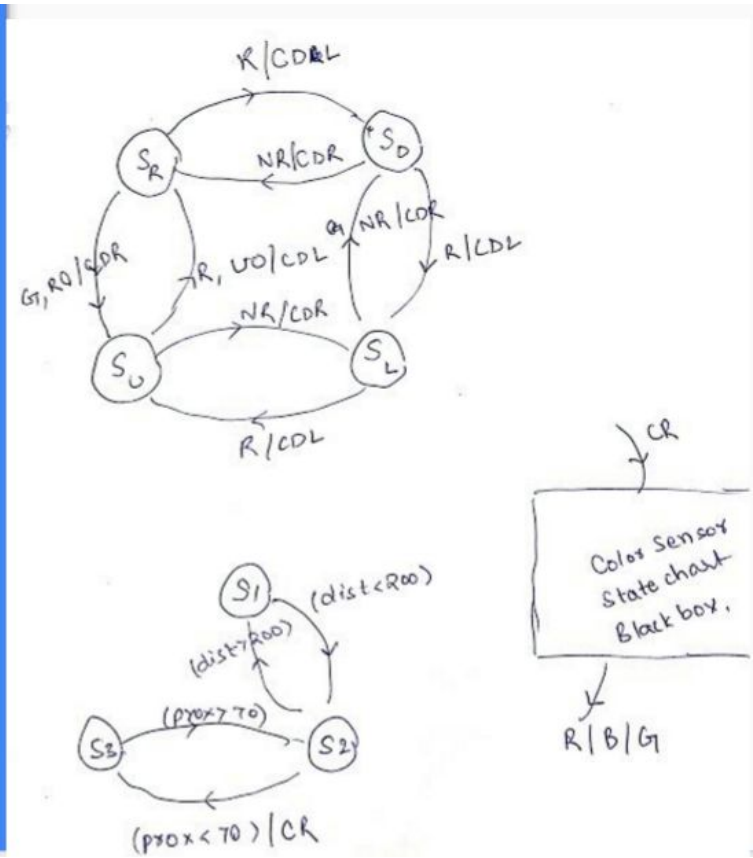
s_U - Firebird is moving in up direction

s_L - Firebird is moving in left direction

s1 - Distance to obstacle > 200

s2 - Distance to obstacle > 70 & < 200

s3 - Distance to obstacle < 70



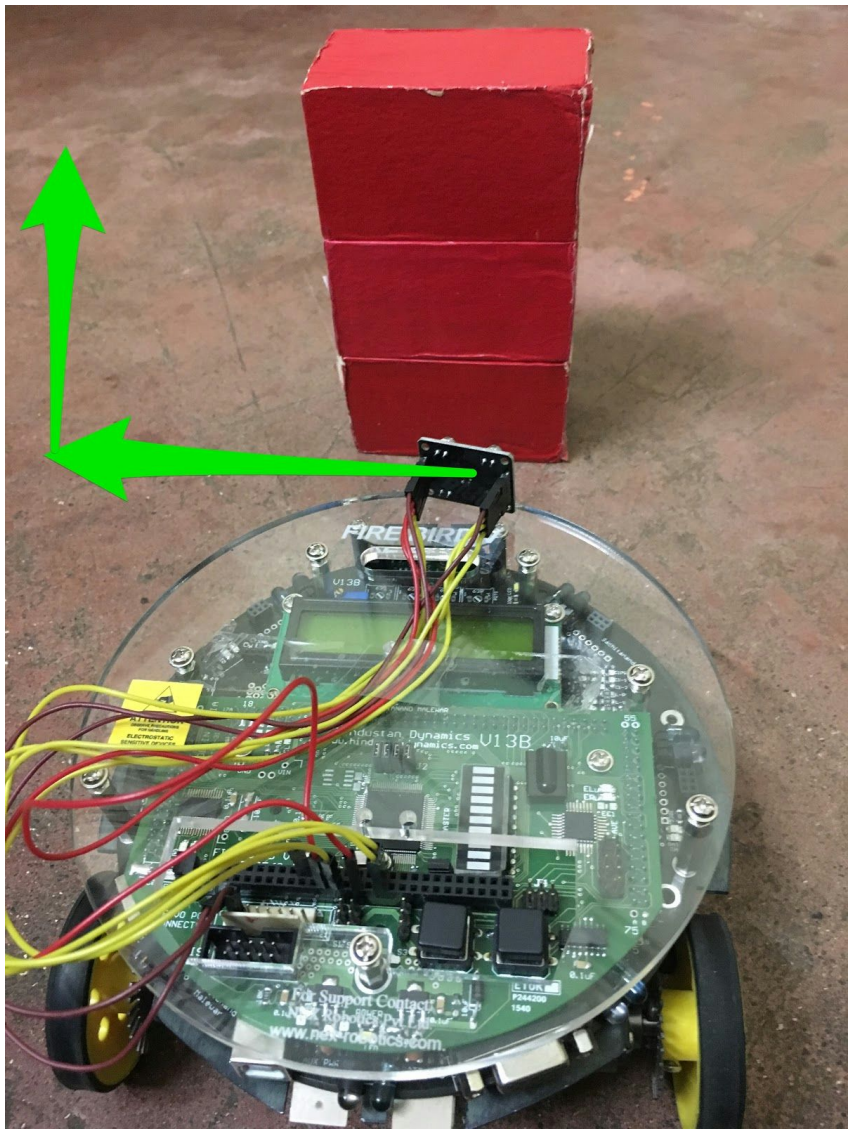
5. Working of the System and Test results

Algorithm:

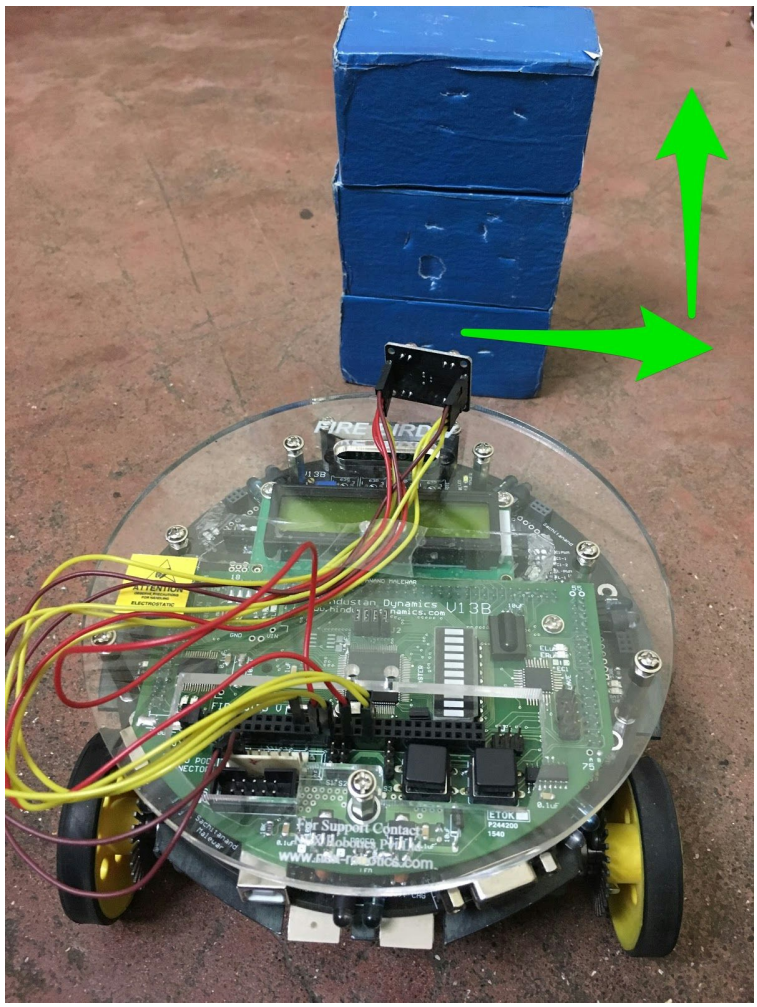
The objective of the firebird is to go to the destination avoiding all obstacles. Firebird moves in steps of fixed length. Whenever the obstacle at a distance greater than 200mm, firebird moves by 20mm. If the distance is between 200mm and 70mm, firebird moves by 10mm. If the distance is less than 50mm, we start detecting the color of object. If the object detected is red, firebird shifts left by some amount and proceeds in same initial direction. If the object is other than red, firebird shifts right and proceeds in the same initial direction.

Eg: If firebird is going north and we if find a red obstacle, firebird rotates left, goes some distance towards west, rotates right and again starts going north.

Red case:



Non red case:



6. Discussion of System

a) What all components of your project worked as per plan?

Detecting obstacles and acting based on color of obstacles worked as per our plan.

Rotating by exact angle does not always work right. This has to do with the friction coefficient at that place between the tyre and surface. Hence the result is unpredictible.

b) What we added more than discussed in SRS?
NA

c) Changes made in plan from SRS:
We wanted to add a black obstacle and keep some other action for it. We were unable to do it because whenever the obstacle is black, proximity sensor stops giving the correct distance value.

7. Future Work and Conclusion

This could be improved to tackle more complicated scenarios. For eg: We could further add code to scan the whole maze and find the destination through shortest path. We could enchance it further by adding more complicated obstacles. We could also improve this to collect an important object in the maze and bring it to the destination.

8. References

We have used several manuals listed in the erts homepage.