COURSE PROJECT DOCUMENTATION

CS101 PROJECT

AUTONOMOUS SMART CAR

TEAM ID : 270

Rahul N	14D170029
Rohith Prakash	14D170022
Micky	140010059

Table of Contents

1. INTRODUCTION

In this current world where everything is becoming automated and more efficient, there is an urgent call for the so called “Smart Cars”. So what makes a car smart?

We broke down the basic requirements for a car to be called smart and reached to the inference that if we divide the responsibility of driving and give a part of it to the car, it is ‘smart’ enough.

Our basic aim for this embedded CS Project is to make an autonomous bot where we make a bot (synonymous to a car) which is capable of interacting with its surroundings such as basic obstacle detection, obstacle type and velocity interpretation.

So, this project of making an autonomous car deals with this urgent call and hence gives a whole new meaning of safe driving on roads with environment car interaction.

2. PROBLEM STATEMENT

The aim of this project is to design an autonomous basic line following bot with several additional functionalities.

The bot must be able to detect anomalies in its path (obstacles).

The bot must be able to identify the type of the obstacle – Stationary, Incoming, Slow moving.

The bot must respond to the obstacle accordingly and execute one of the three sub-codes to overcome the obstacle and re-align itself to the white line and continue on its path.

3. REQUIREMENTS

A) Hardware:

1. Firebird -Our car for the project
2. White line path -As a direction path for the bot
3. Proximity Sensors -To sense the obstacles on its avoiding maneuver
4. Smart IR sensor distance -To detect and identify the obstacle from a
5. IR Sensor -For the line following mechanism

B) Software:

1. AVR Studio -To program instructions onto the bot

4. IMPLEMENTATION

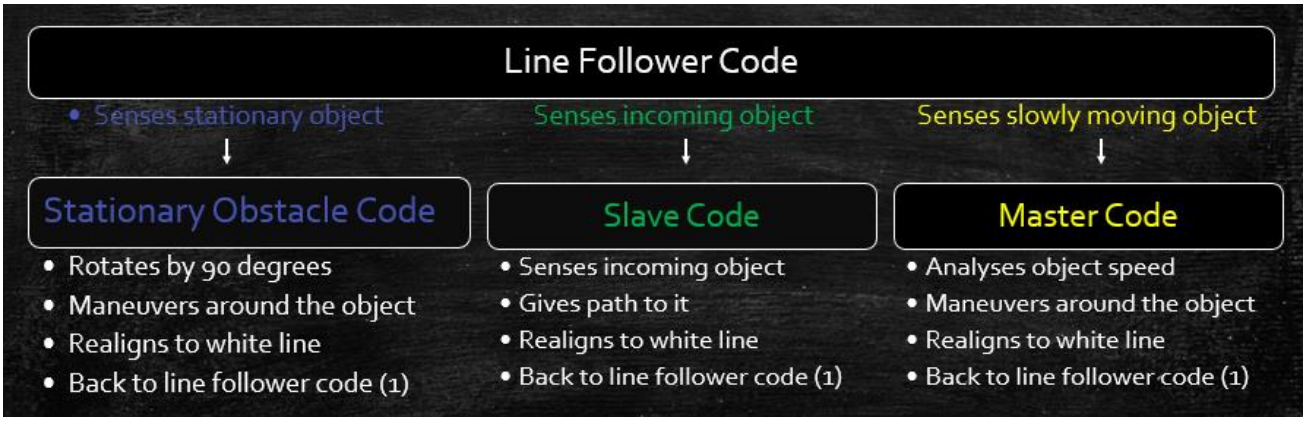
A) Functionality:

1. **Sense White Line-** This is the basic backbone function of the bot. It starts with this function and follows the path through this in between the obstacle instances.

2. **Senses the Obstacle-** This is a continuous loop with the sharp IR sensor giving readings which enables us to understand when an obstacle occurs.

3. **Identifies the Obstacle-** It is also done with the help of IR Sensors on the basis of distance covered by obstacle. It calculates the velocity of the obstacle and hence identifies it as a stationary one, an incoming one or a slowly moving one.

4. **Obstacle Maneuver-** It is the part where the bot executes one of the three codes as shown in the picture below and crosses the obstacle.



5. TEST STRATEGY AND DATA

As a whole we are using about 3 proximity sensors and a sharp IR sensors. In which we take readings from sharp IR sensors continuously and decide accordingly

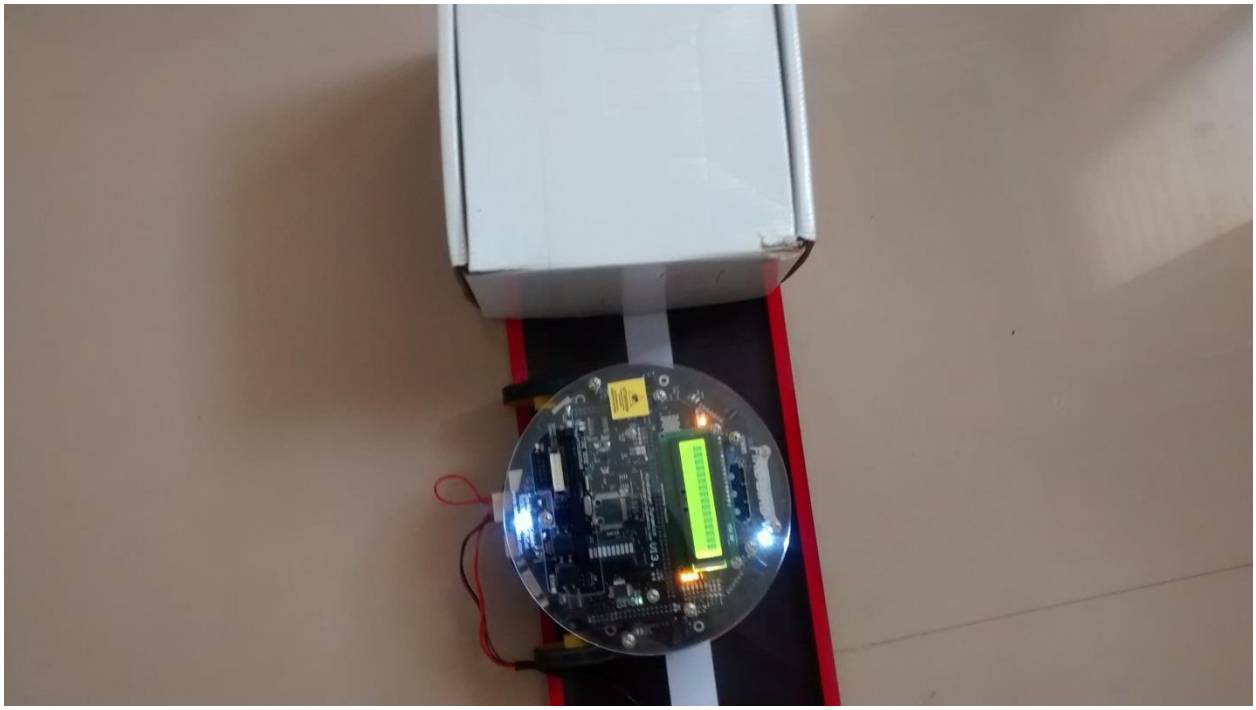
When there is an object before the bot perpendicular to the IR ray IR sensor senses the obstacle

The output voltage from IR sharp sensor is converted to distance to the obstacle (800mm – 0 mm).

BOT ORIENTATION FOR STATIONARY OBSTACLE:

In this close orientation proximity sensor gets a reading less than 100 then we take a 90 degree turn

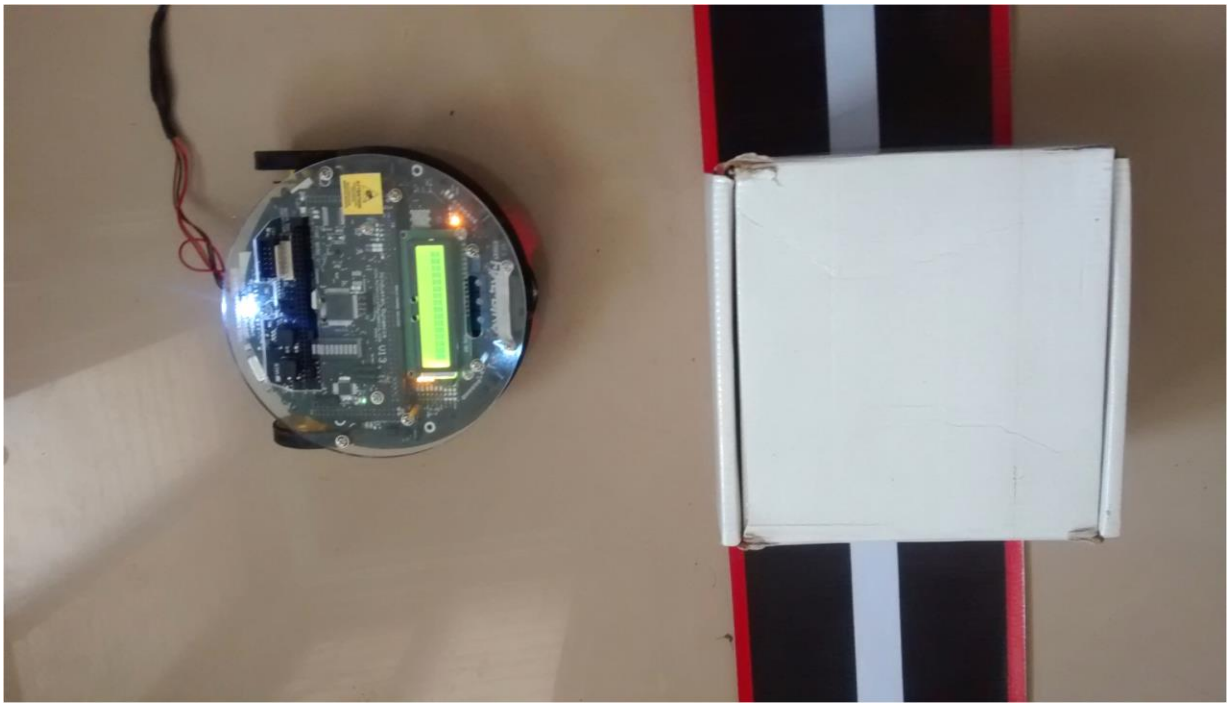
BOT ORIENTATION AFTER 90 DEGREE TURNS:



After a 90 degree turn and a combination of zigzag motion right proximity sensor get a reading less than 100

BOT ORIENTATION AFTER GIVING WAY TO AN INCOMING OBSTACLE:

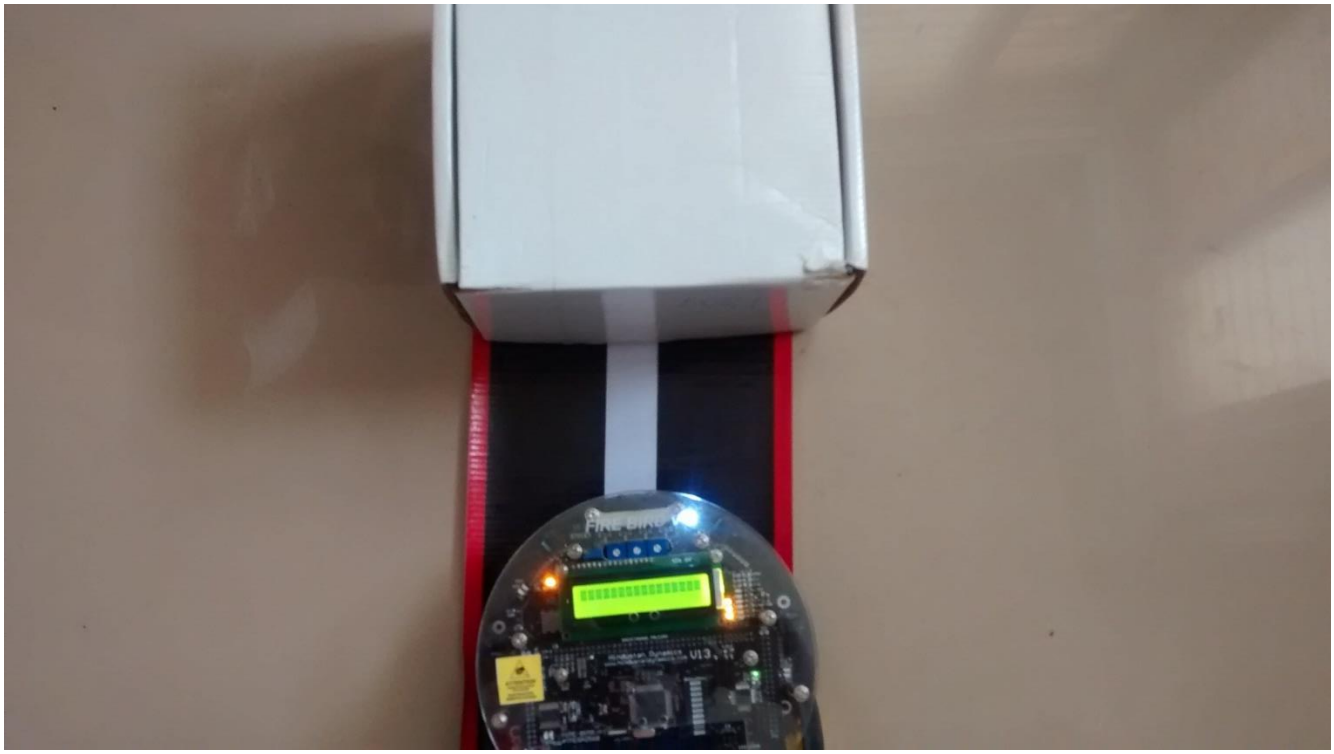
SHARP IR SENSOR takes continues reading and once it go less than 750 we consider this as obstacle is before the bot



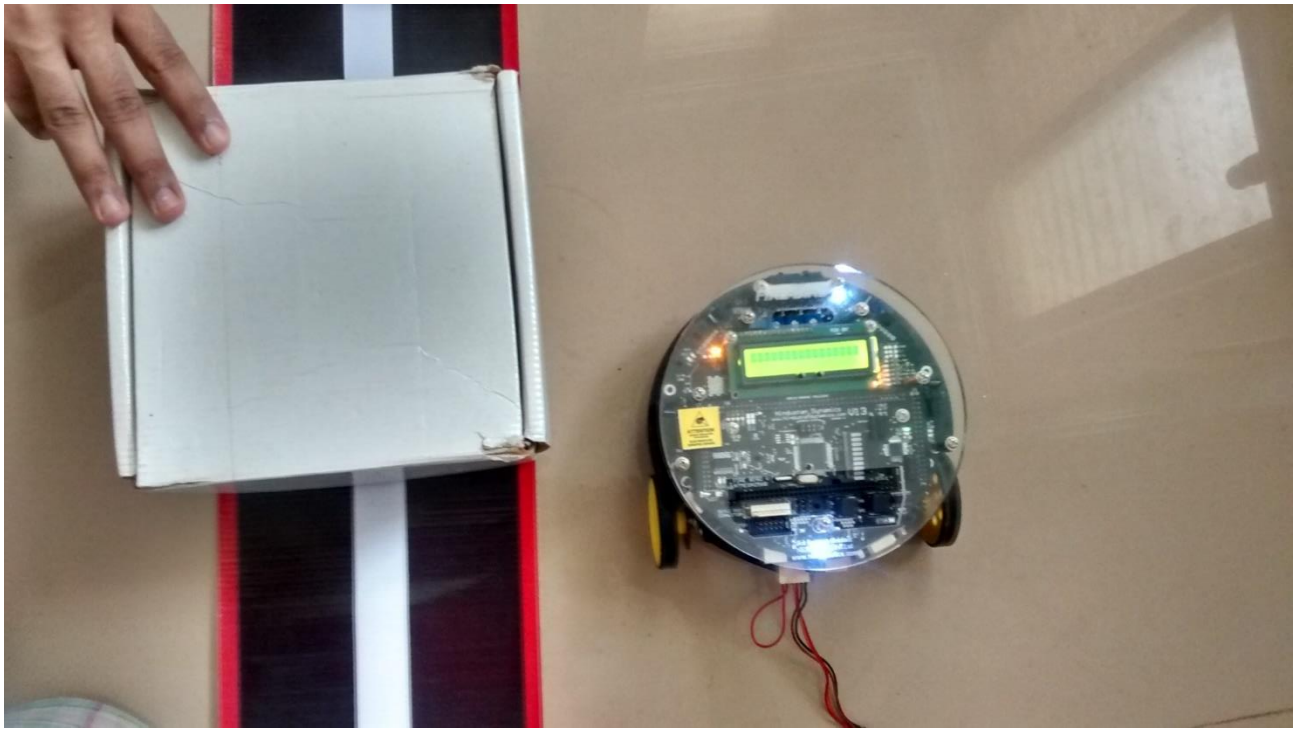
And after this once we get a reading more than 750 we consider this as obstacle crossed

BOT ORIENTATION TO OVERTAKE AN OBSTACLE:

The front IR sharp sensor reading is less than 150mm.



Then we take a 90 degree turn towards right and move forward for a fixed time and then take another 90 degree turn towards obstacle and if IR sharp sensor reading is more than 750mm we can consider that the obstacle is on the left and we can go forward and overtake



6. Discussion of System

A) Parts that worked as per plan:

1) **Line Following Mechanism-**

   It was a challenge to extend the line following code during the obstacle maneuvering period. We need the line follower to activate the instant it detects a white line underneath so that it can resume its path

2) **Obstacle Detection-**

   Achieved successfully by front Sharp IR sensor

3) **Obstacle Maneuver-**

   Maneuvers successfully around obstacle regardless of shape

B) Additional Parts added beyond plan:

1) **Incoming Obstacle-**

   Included extra functionality to deal with incoming obstacles. That is based on the obstacle analysis

2) **Slowly Moving Obstacle-**

   Added extra functionality to deal with slowly moving objects and giving the bot the power to overtake it

3) **Obstacle Analysis-**

   The above two functions require the bot to analyze the obstacle and determine in which category it fits. Hence, it has an extra functionality to determine the obstacle type, achieved by the front IR Sharp sensor

C) Changes Made in Plan:

1) Way to cross over the obstacle-

The **unpredictability and unreliability** of the proximity sensors led us to change the way we cross over the obstacle.

Sharp sensors do not give good readings when the obstacle gets too close. So, after determining that the object is a stationary one, we decided to do a 90 degree rotation on the spot and have a **to and fro** motion and bring the bot closer till the point the proximity gives a good reading.

From there on, we could use the logic in proximity to avoid the obstacle.

7. FUTURE WORK

Our aim is to create a car environment interaction as contrast to the current system of driver environment interactions. This is solely because machines reduce the possibility of errors.

1. So the main Future aspect would be to implement this in real cars as in this physical world incorporating several more **environmental interactions** such as signal readers (red light), pedestrian recognition, emergency handling procedures, etc.

2. We hope to take this to a further level and reach a point where there is not only **car to environment** interaction, but also **car to car** interaction creating the perfect foolproof system of road transport.

3. Also, currently the bot uses white line sensing to cover the path. An additional work could be to create a path between a start and an end point using **GPS** system or **a google maps** system on which the bot would traverse.

8. CONCLUSION

Our work/vision is based on a similar idea as the **Google Car** and we would like this project to be taken to a whole new level where we see its implementation in the real world in the near future.

The ideas are already present. All we need is some sort of a heavy researching push which creates the ideal road transport system.

This cannot be taken as a CS101 project anymore it can be taken as the future.

9. REFERENCES

A) E-Yantra Tutorials for
   a. Line Follower guide
   b. Velocity control guide
   c. ADC_Conversion guide
   d. Motor control guide

Documented archives from the E-Yantra website-

https://www.dropbox.com/sh/2orydpmoopeczi3/AAARr6-jAeD7zlTCYq4rChZCa