MESH FLARE

TECHFEST

ASIA'S LARGEST SCIENCE AND TECHNOLOGY FESTIVAL

--A long way to go in a short time

IIT DELHI

Table of Contents

TEAM MEMBERS

Sushant Rathi	**2016CS10328**
Pratyush Maini	**2016EE10412**
Shashwat Shivam	**2016CS10329**
Shashank Goel	**2016CS10332**

MENTOR

Vikas Singh

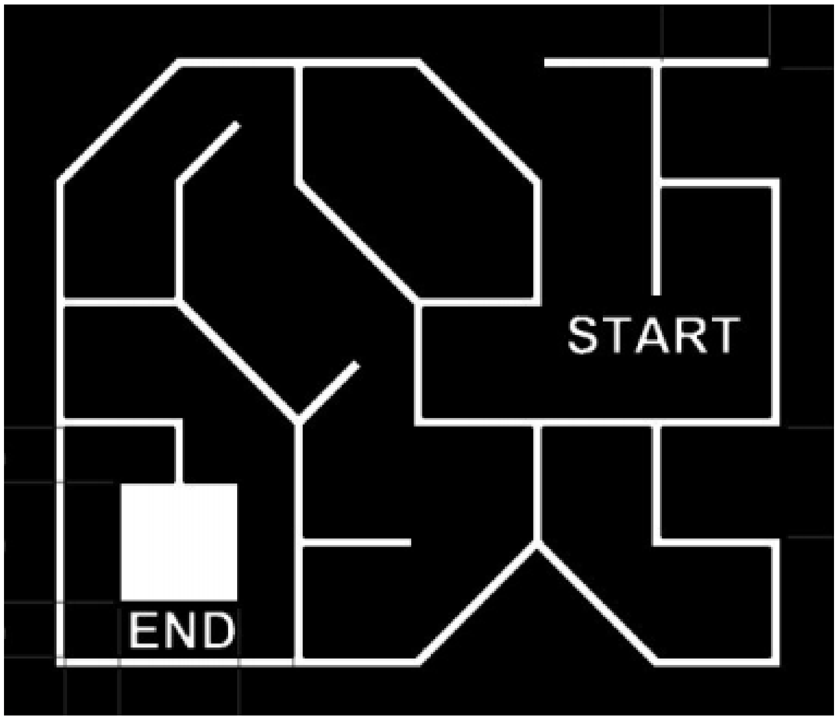
Problem Statement

Teams have to build an autonomous robot, which can follow a white line and keep track of directions while going through the maze. The bot has to analyze the path in the dry run and has to go through the mesh from the starting point to the ending point in minimum possible time.

Other important Rules:

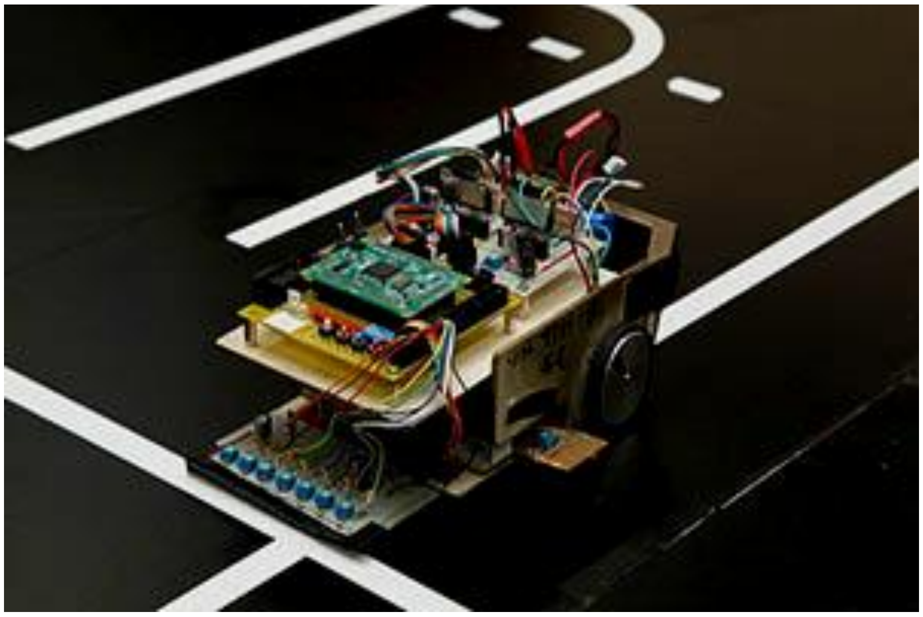
1. The game field consists of an arena having dimensions 2310 mm X 1980 mm (lxb)
2. Angle between two adjacent white lines in the path is either 900 or 135o
3. The width of all white stripes will be 30mm
4. The autonomous bot must fit into the box of dimension 220 mm X 220 mm X 220 mm (l x b x h).
5. Bot must have a red LED, which will glow once it reaches the end zone of the arena.
6. The potential difference between any 2 points must not exceed 24 V at any point of time during the game.

Sample Maze:



Breaking Down The Task

1. Make a simple line follower that is stable in following a straight or curved path.
2. Identify the types of intersections that the bot may encounter during the task.
3. Make the bot turn at different sections according to the command given.
4. Map the maze through an effective coordinate system.
5. Write an algorithm to solve the maze by avoiding loops and finding the shortest path.
6. Testing and redesign at each step.



Making A Line Follower

— Setting up the bot

Our first aim was to make a stable line follower that could sustain its motion without wobbling or deterring from its path.

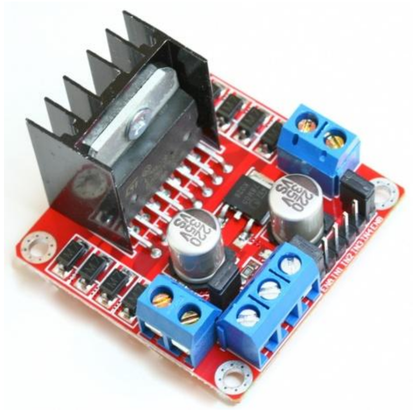
Material used:

1. Chassis

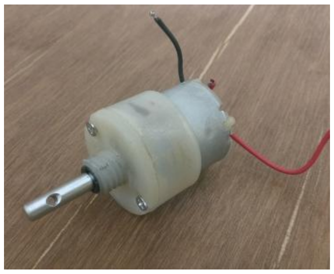
2. Castor Wheel

3. Wheel

4. L298n Motor Driver



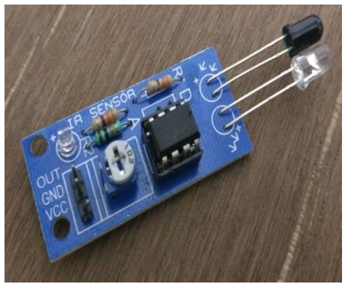
5. Motor
RF-500TB-12560



6. Arduino Uno

7. Jumpers and Wires

8. Infrared Sensors

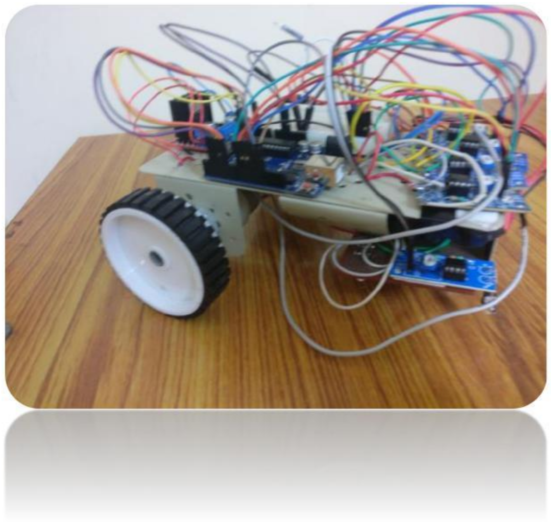


9. Rechargeable Battery

Basic Design

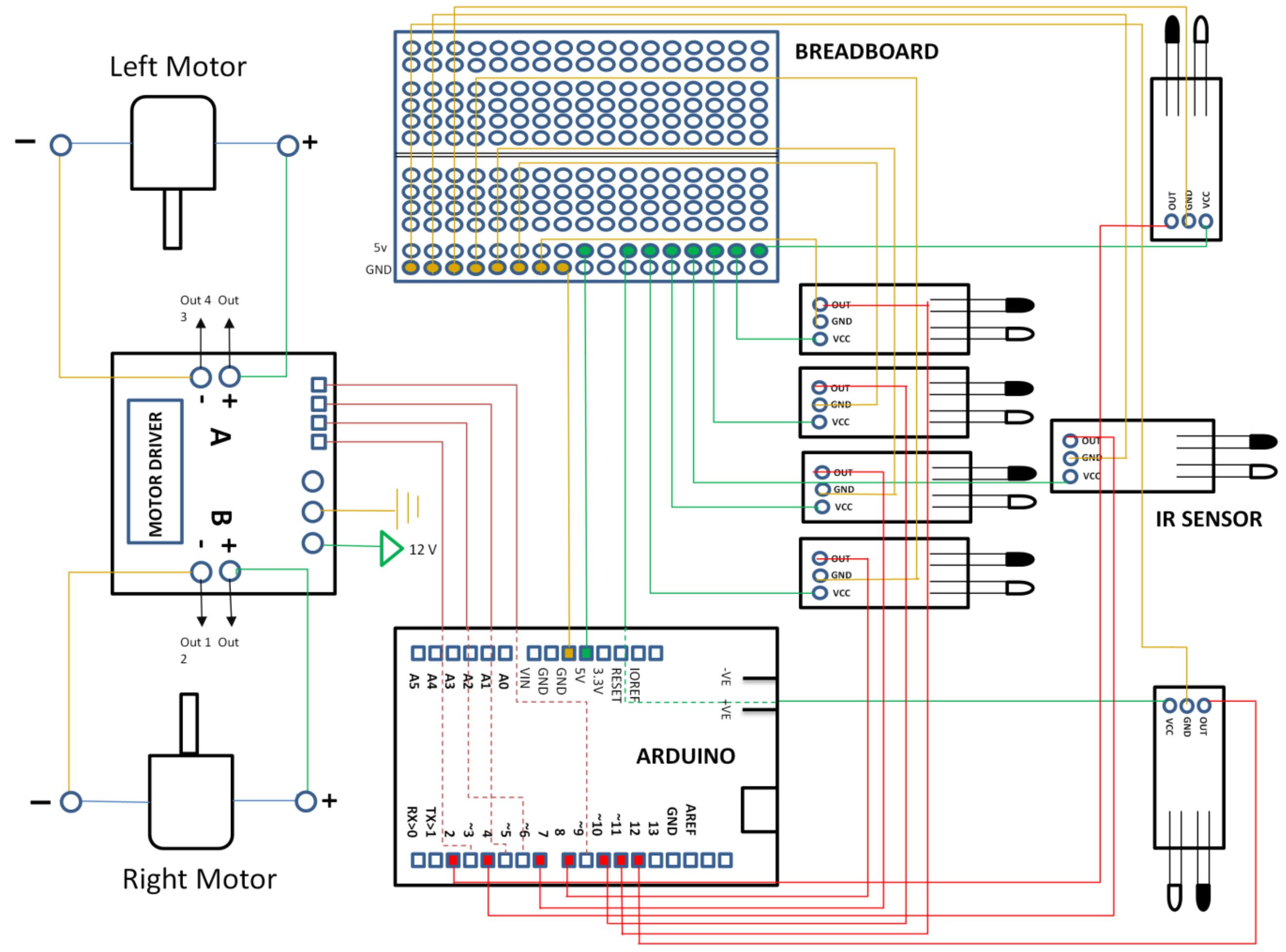
The task of the competition demanded intensive electrical circuit design and programming algorithm rather than the bot structure.

So we didn't complicate things much and used the standard chassis of size 20cm X 10cm, that is readily available on the market.

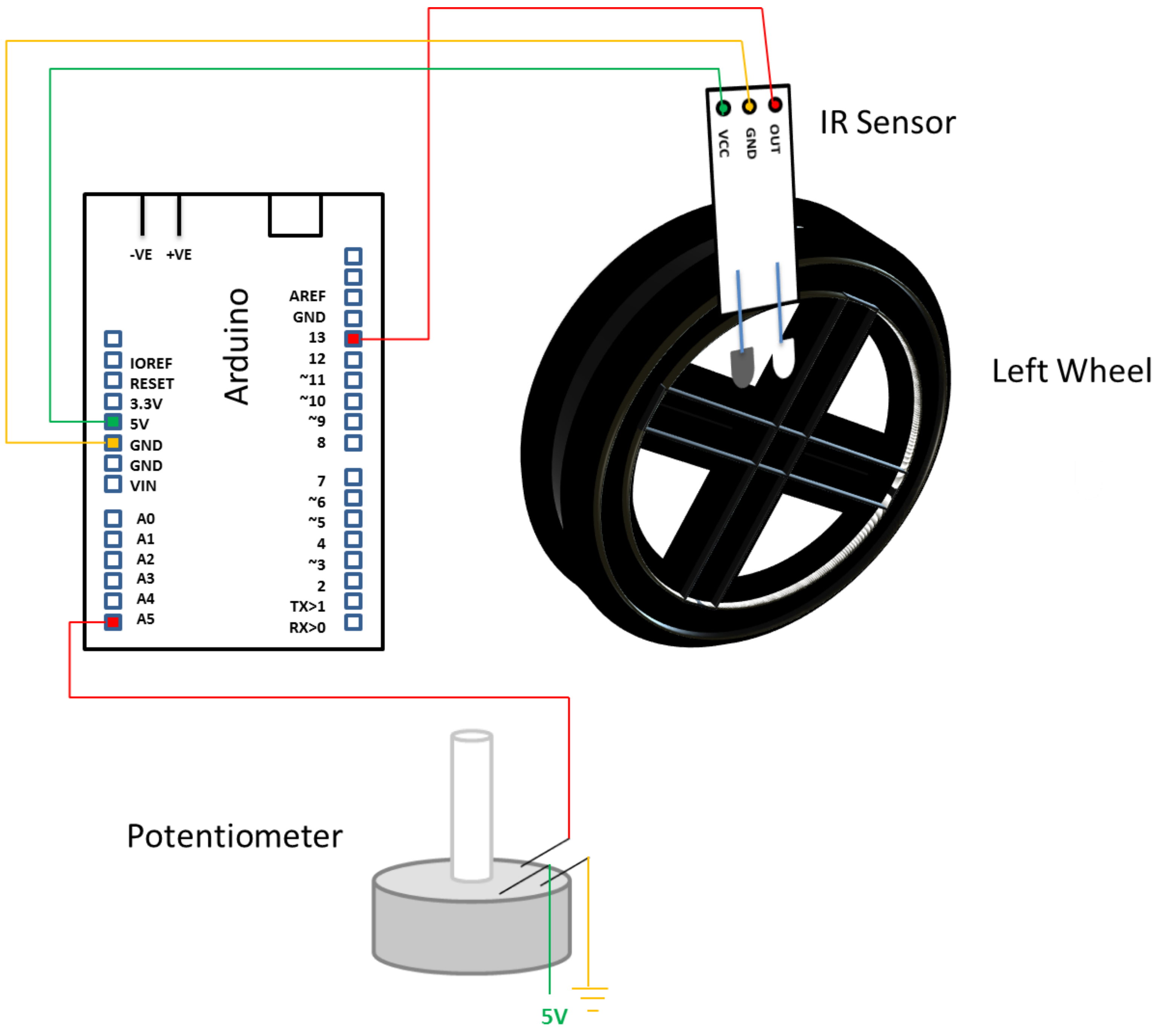


We placed Standard wheels along with castor wheels at the bottom, leaving large amount of space for placing the Arduino, breadboard, motor driver etc. on the chassis.

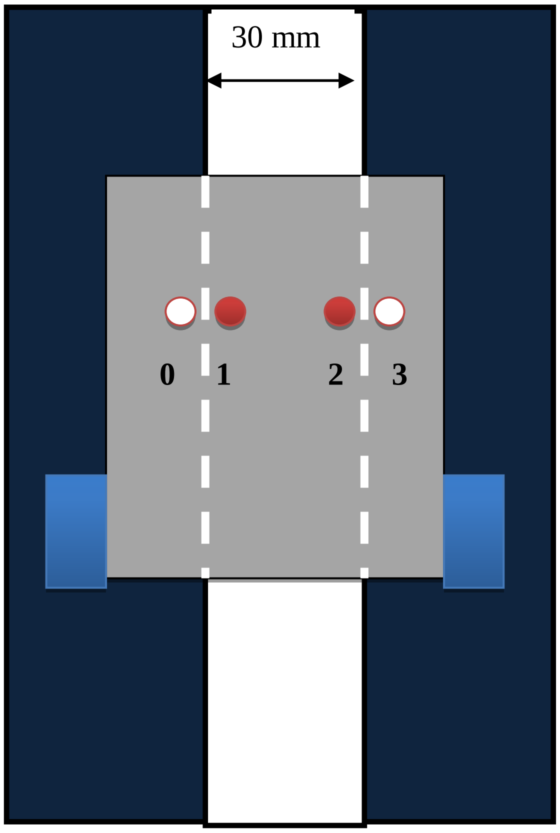
— Circuit Diagram



SPECIAL MENTION: These circuit diagrams corresponding to the circuital orientation of our bot and the rotary encoder set up on the wheels, have been solely made by us from scratch using PowerPoint.


IR Sensor
VCC
GND
OUT
Left Wheel
Arduino
-VE
+VE
AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX>1
RX>0
IOREF
RESET
3.3V
5V
GND
GND
VIN
A0
A1
A2
A3
A4
A5
Potentiometer
5V

— Algorithm for Line Following

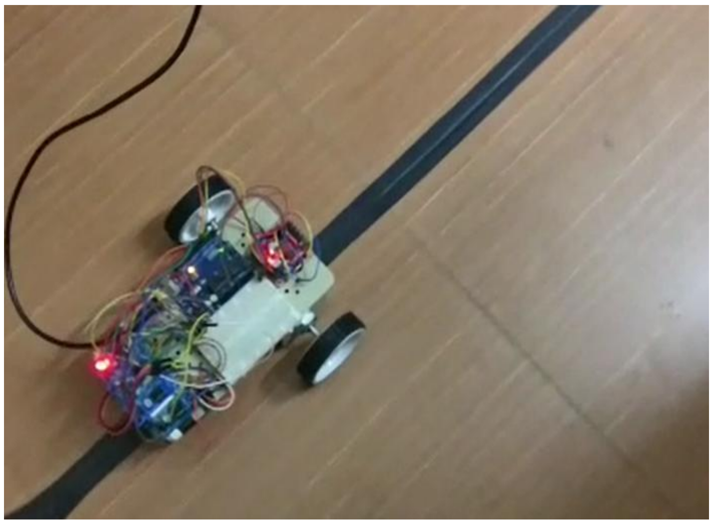
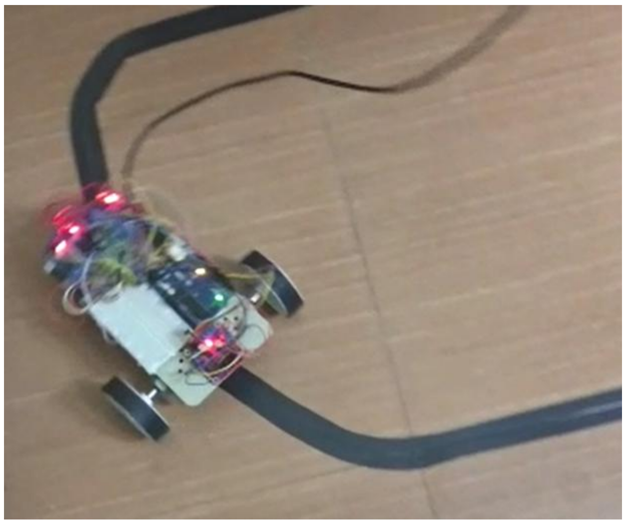


The ideal condition for the straight motion of the bot on a line is shown above. Out of the 4 sensors place for line following, the two within the 30 mm of the white strip must stay on and the two outside must stay off.

Case	Output	Left Wheel	Right Wheel	Implication
1	1 & 2 are ON	150	150	Move Straight
2	1 is ON, 2 is OFF	20	150	Turn Left
3	1 is OFF, 2 is ON	150	20	Turn Right
4	0, 1, 2 & 3 are OFF	150	-150	Right U-turn

Here 150 and 20 represent the relative RPM of the wheels on a scale of 0 to 255, as obtained from analog input.

— Testing and Redesign

In the first round of testing, we tested the bot for both straight line and a line with curvature. After, a few changes in algorithm and position of sensors, we were able to get the desired line following ability in our bot- without much wobbling.

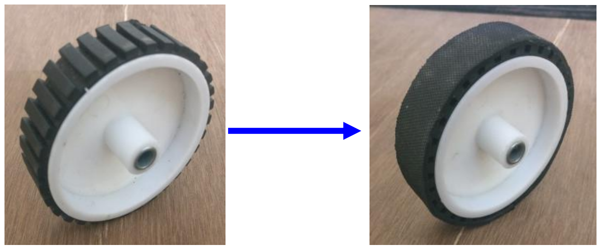
— Other considerations

1. Barren land syndrome:

It is an event in robot terminology when the bot does not find any lines to follow. To counter this, we initiated a command to reverse the bot to its position 1 second ago, whenever all the signals are turned off.

2. Extra grip on Wheels

The wheels that we procured lacked sufficient grip, which was resulting in loss of control over the robot on curved lines. Therefore, we placed another strip of high traction rubber on the wheels.



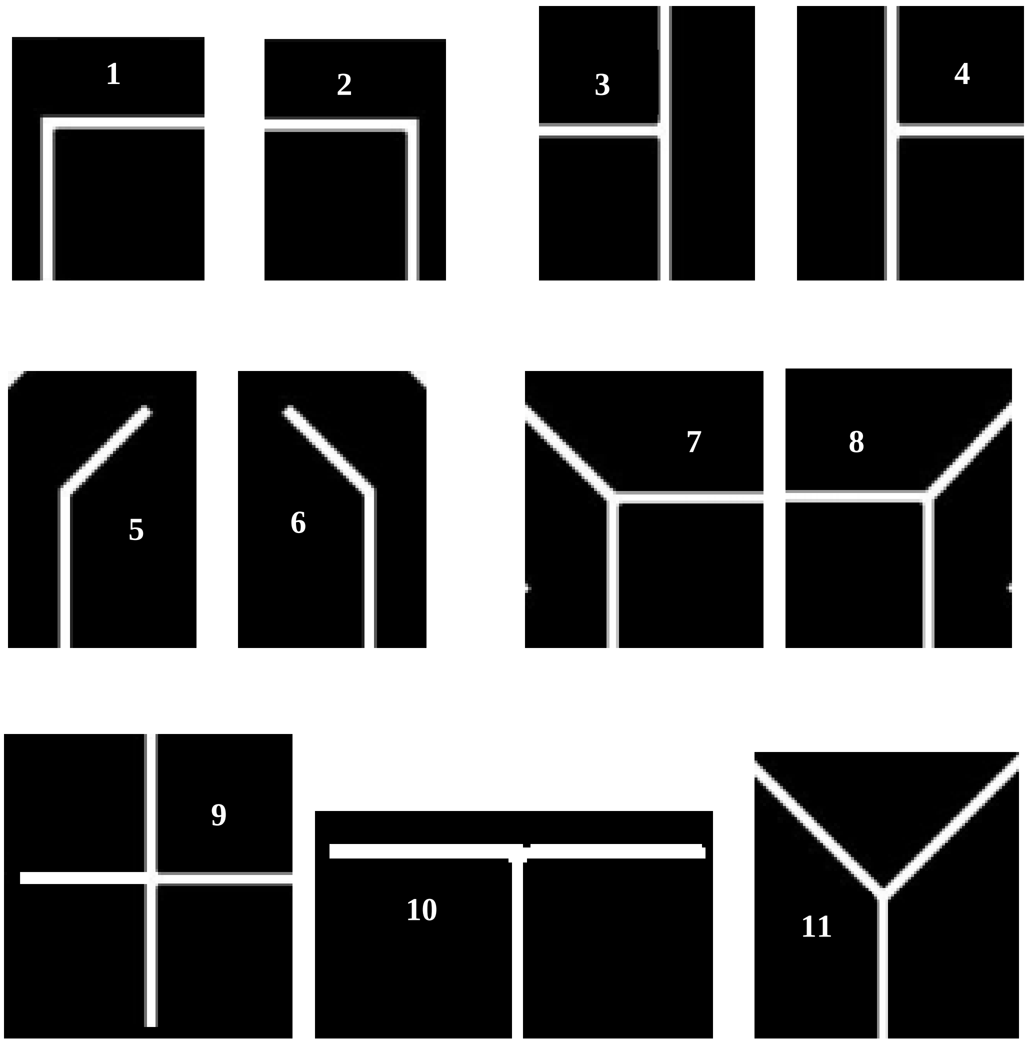
3. PID Control:

PID, Proportional Integral Derivative, is a standard algorithm by which one may optimize the stability of the bot on straight and curved lines. It takes into account the deviation of the bot from the desired path by taking input from various sensors and accordingly mediates the rpm of the two wheels.

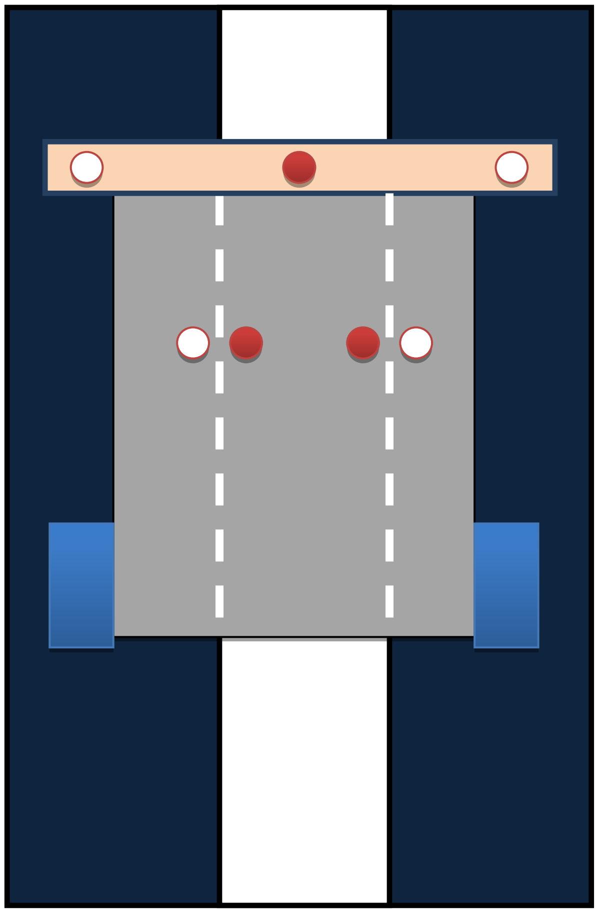
However, we decided against implementing the same, as it caused problems on intersections (read further for intersections).

Tackling Intersections

— Listing all Possible Intersections

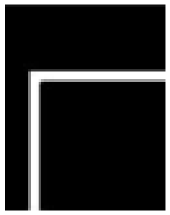
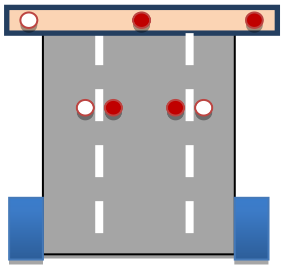
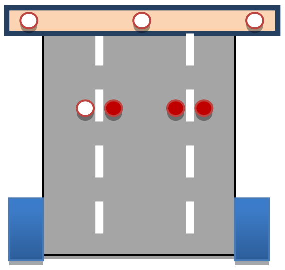
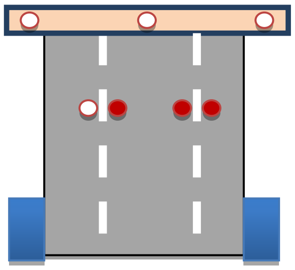
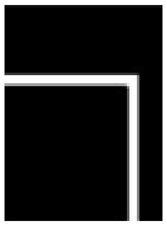
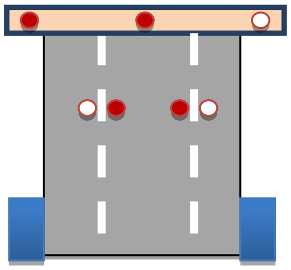
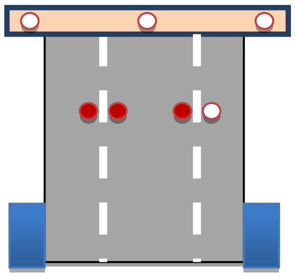
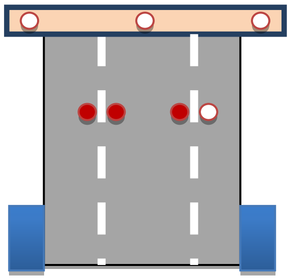
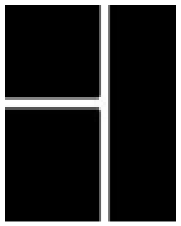
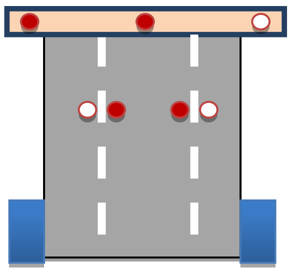
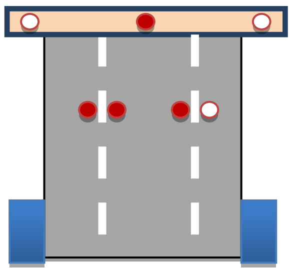
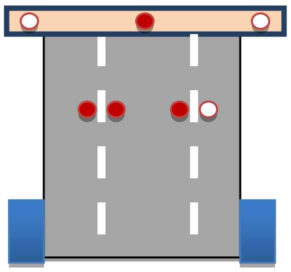
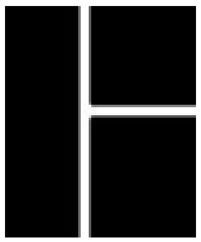
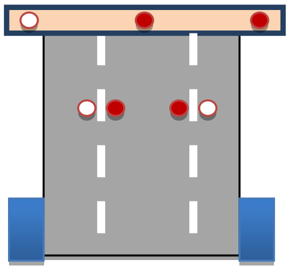
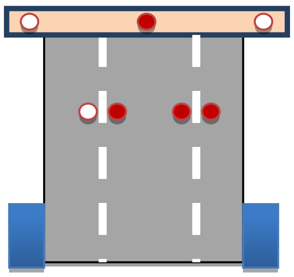
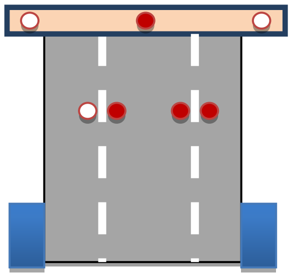


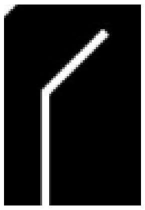
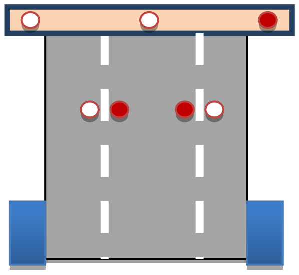
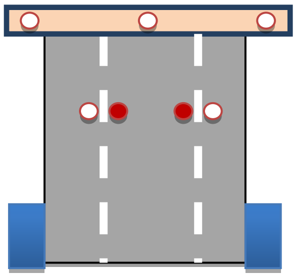
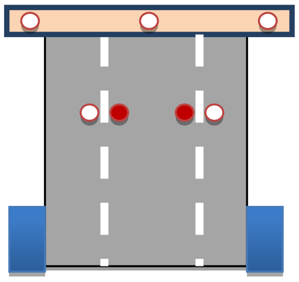
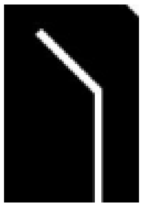
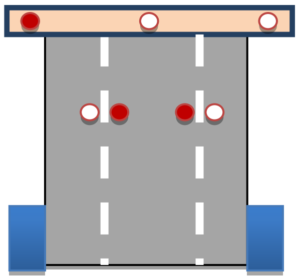
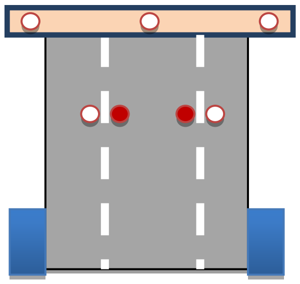
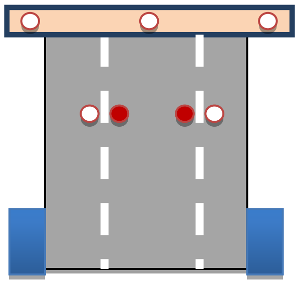
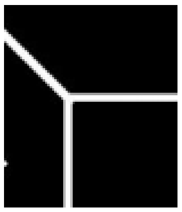
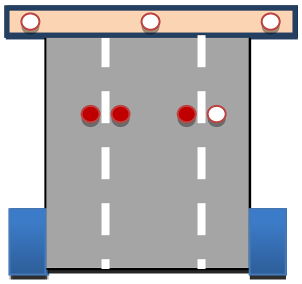
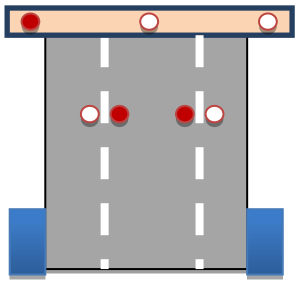
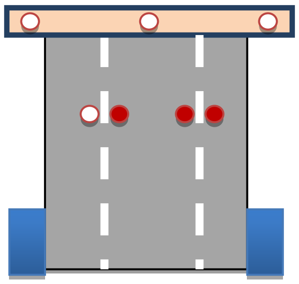
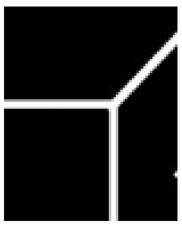
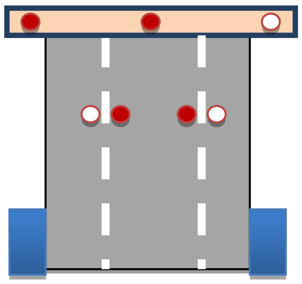
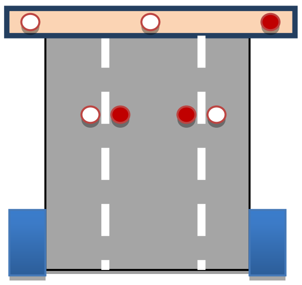
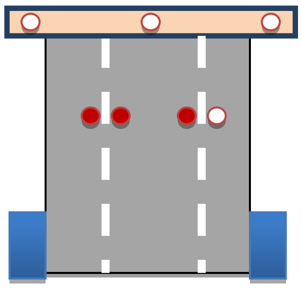
— Placing Sensors

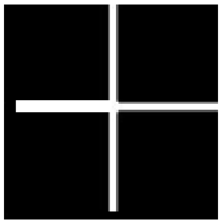
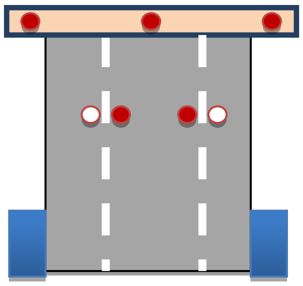
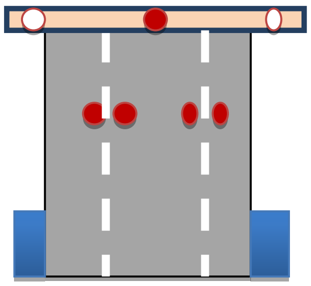
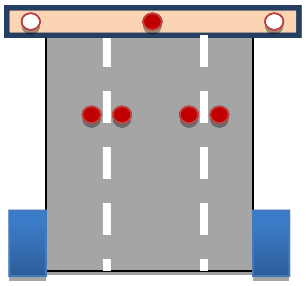
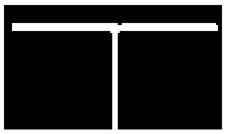
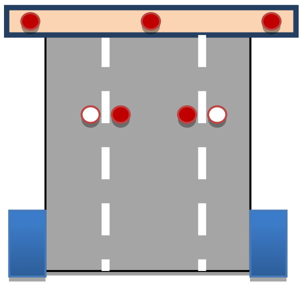
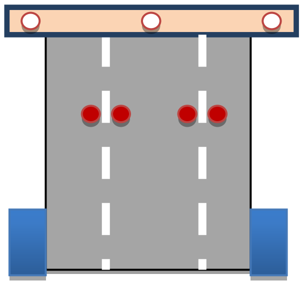
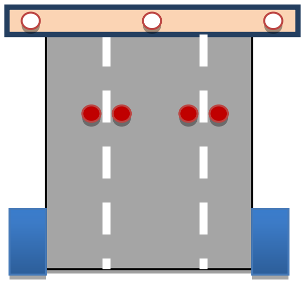
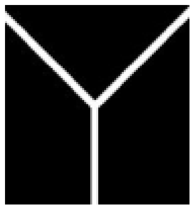
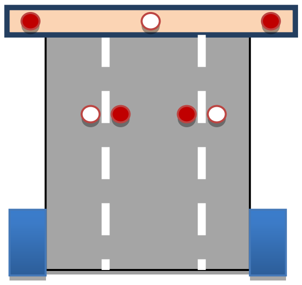
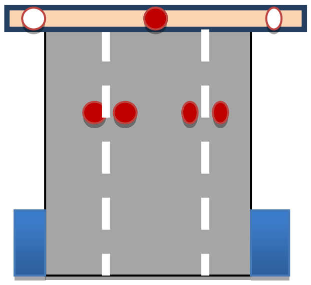
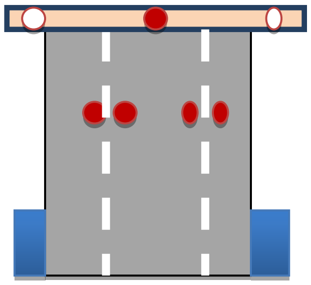


Three additional sensors were placed in front of the bot in order to be able to determine the type of intersection. The output of these sensors is taken only at the time of intersections and does not interfere with simple line following algorithm.

— Algorithm for Identification

Intersection	Initial Output L F R 0 1 2 3	Read Values Till L F R 0 1 2 3	Output After Delay
1 	0 1 1 x x x x 	0 0 0 0 x 1 1 	0 0 0 0 x 1 1 
2 	1 1 0 x x x x 	0 0 0 1 1 x 0 	0 0 0 1 1 x 0 
3 	1 1 0 x x x x 	0 1 0 1 1 x 0 	0 1 0 1 1 x 0 
4 	0 1 1 x x x x 	0 1 0 0 x 1 1 	0 1 0 0 x 1 1 

5 	0 0 1 x x x x 	0 0 0 x x x x 	0 0 0 x x x x 
6 	1 0 0 x x x x 	0 0 0 x x x x 	0 0 0 x x x x 
7 	0 1 1 x x x x 	1 0 0 x x x x 	0 0 0 0 x 1 1 
8 	1 1 0 x x x x 	0 0 1 x x x x 	0 0 0 1 1 x 0 

9 	1 1 1 x x x x 	0 1 0 1 1 1 1 	0 1 0 1 1 1 1 
10 	1 1 1 x x x x 	0 0 0 1 1 1 1 	0 0 0 1 1 1 1 
11 	1 0 1 x x x x 	0 1 0 1 1 1 1 	0 1 0 1 1 1 1 

Where 0, 1 and x refer to OFF, ON and Independent respectively.

Since the bot does not always approach intersections in an entirely straight motion, all the sensors may or may not show the desired output at the same time. For example in a T section, the L and F Sensor may be 1, but R sensor may take a few milliseconds to turn 1. Therefore, we incorporated delays in the reading of the output.

Turning The Bot

After the effective identification of the intersections, the next step was to be able to accurately turn at all intersections.

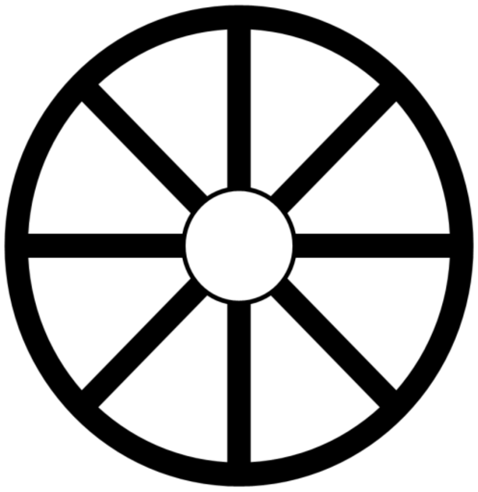
Turn	**Left Wheel**	**Right Wheel**	**Break at (**Follow all Sequentially**)**			
Left	-120	80	L==1	0,1==1	F==1	0==0
Right	80	-120	R==1	2,3==1	F==1	3==0
Diag. Left	0	150	L==1	0,1==1	F==1	0==0
Diag. Right	-150	0	R==1	2,3==1	F==1	3==0
Left U	-100	100	0, 1==1	F==1	2==1	
Right U	100	-100	2, 3==1	F==1	1==1	

Here,
0 refers to OFF state and 1 refers to ON,
100, 120 and 150 represent the relative RPM of the wheels on a scale of 0 to 255, as obtained from analog input.

The above algorithm has been made quite comprehensively in order to ensure that it does not fail at any of the 11 given intersections.

Mapping The Maze

For mapping the maze, we created a self-made rotary encoder cum coordinate system. The basic principle of this encoder was to measure the number of fractions of revolutions of the wheel. We made use of an IR sensor and made at black spokes on the wheel of the bot. So, in every revolution 8 fractions are detected (refer to fig below), which help incrementing the coordinates of the bot.



Avoiding Loops

A comprehensive algorithm was devised for avoiding loops. A coordinate system was initialized which measured distances in each direction the bot moved. The distance was measured with the help of the self-made rotary encoder. The direction of motion was known by the turns the robot moved. For instance a diagonal right would decrement the angle by 45^o.

Initially all paths on the maze are ranked 0.

Now that the coordinates of the robot are known, we marked each place where the robot took a turn as a node, and hence noted the coordinate of each node. Whenever a robot went to a node, the rank of the path it came from and the path it took are incremented by one. This way at each intersection, the robot traverses the path that is least travelled (according to rank).

Hence, we were successfully able to avoid loops.

Problems Encountered

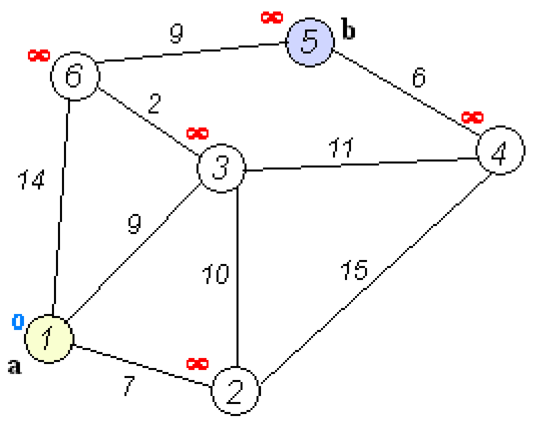
Since the rotary encoder was not very accurate, we needed to take into account the error in coordinate measurement. So, we had to put an error limit in the x and y coordinate. We ended up putting the condition, if change in $x^2+y^2 < 50$ for two different nodes, then consider them to be the same node.This added an element of uncertainty to our bot, but was clearly the best possible solution after a lot of experimentation.

Finding The Shortest Path

In the final run, the bot must reach the end point in least time. In dry run, we ranked all paths on the basis of the number of times it was traversed upon. So, every path on the mesh (whether traversed or not) has an initial ranking of zero.

Every time the bot reaches takes a path between any two nodes, the ranking of the path from both the nodes is incremented by one.

In the final run, the bot was programmed to take the highest ranked path at each intersection (opposite to what was done during the dry run). This ensured that at each point it traverses the path that will take it to the end in the shortest time.



RESULTS/REMARKS

The competition was a very well contested one, with a large number of teams turning up from different parts of the country.

In August last year, MeshFlare had conducted zonal rounds where they selected 16 finalists.

Our entry was a wild card entry (something that we were unaware of) and had to fight for the 2 remaining slots. There were over 65 registrations for the wildcard round, however the number of wildcard teams that turned up was not disclosed.

We put in a good challenge by being one of the few teams to have completed the dry run, and passing all checkpoints. We scored 60-65 points in the dry run (exact score not known).

The final run did not go very well, as even though we were able to map the maze, due to battery issues the robot stopped taking proper turns, because of which we were not able to complete the final run in the stipulated time.

Unfortunately, we weren't able to qualify for the final with the score we had.

However, the tournament was a great experience for all of us and we got to learn a lot.

Bill Of Materials

S.No.	Item	Quantity	Cost
1	Arduino	2	500
2	IR Sensors	20	500
3	Breadboard	1	100
4	12 V Battery	1	240
5	9V Battery	4	700
6	Travel Ticket	4	6400
7	Jumpers	100	100
8	Chassis	1	150
9	Motors and Wheels	2	600
10	PCB	2	50
11	Iron Rod and Stand	2	400
12	Return Tickets	2	3100

Total Bill- 12840

The material was procured from Chandni Chowk, SCOOPS and Jia Sarai Market.

Personal Experience

Sushant Rathi

Participating in MeshFlare at IITB was a learning experience for me as it introduced me to the challenges in building real-world systems. Though our project was basic in terms of its purpose, the challenges in implementing theory to practice did teach me a lot, for example constructing an algorithm keeping possible error measurements in mind. This project also made me more familiar with building stuff using the Arduino.

Pratyush Maini

Participating in MeshFlare proved to be a very memorable experience for me. From making the robot architecture to electrical connections and programming the Arduino, I was exposed to a lot of interesting things. However, the most important learning was being patient and determined during testing and redesign. Seeing our bot solve the maze for the first time was one of the most exciting moments. Apart from this, visiting IITB, being part of TechFest was also a great experience. Even though, we could not win an award, the tournament was a great learning curve in my life because it exposed me to the field of Robotics.

Shashwat Shivam

The process of making the robot was a very new experience to me. It helped me learn new things and concepts. It helped me realize the proper value of time management and also the importance of keeping time for testing of any project, which will surely help me in future years of my college and life. The competition at IIT Bombay itself was a very enlightening experience. Seeing so many teams and the level of their preparedness gave us an idea of how we should do our future projects with proper planning beforehand. Although we didn’t fare too well in the competition but we learnt many things and it also gave us an opportunity to see how events are organized in other colleges and the skills required to make any event a success.

Shashank Goel

Participating in MeshFlare, TechFest 2017 proved to be a very fascinating and memorable experience for me. Initially, It seemed to be a difficult task, probably challenging because I did not have had any prior experience at it. It also made me realize the importance of Time Management. The most important thing required to complete this task of creating the working prototype was “PATIENCE” especially during testing and calibration. This venture helped me not only earn a Bombay Trip but let me know of how the robot world was created and made me be a part of the same.

Acknowledgements

We would like to express our profound gratitude and deep regards to our mentor Vikas Singh and Robotics Club for their constant support and valuable inputs and encouragement.

We would also like to thank Professor Subir Kumar Saha, who gave us the opportunity to work for this competition, which gave us a platform to think out of the box and helped us in doing a lot of research and explore new market because of which we came to know about many new things.

We are also thankful to our Coordinators Vaibhav Gupta and Nishant Agarwal for constantly guiding us to create the working prototypes.