AUTO SERVING BOT

TEAM DETAILS:

AMIT PATHANIA
GOUTAM L
MANJUNATH K

Problem Statement

- The project aims to automate the order serving process in restaurants to improve the services provided to the customers by avoiding human errors.

Description

- The project required operating multiple firebird robots for delivering orders and share workloads among bots. The bots will be controlled by a central server at the counter while implementing features like finding the shortest route to the table, obstacle detection, collision avoidance and rerouting dynamically.

TASK SPECIFICATIONS

- Placing an order using an android app.
- Storing the orders using a database server at the counter.
- Delivery of order to destination table by the free bot .
- Navigation back to start point once the order delivered.
- Collision avoidance and rerouting in case of obstacle detection.

PROJECT PLAN

TASK	WORK DIVISION	DATE OF COMPLETION	CRITICAL TASK
Making an android app.	Goutam	21 Oct 16	
Installing WAMP server and maintaining database of menu items, orders placed and delivered.	Amit Pathania, Goutam	08 Oct 16	
Passing destination table coordinates to a free bot from admin webpage using python.	Amit Pathania	17 Oct 16	
Communicating with bots using Zigbee.	Amit Pathania, Manjunath K	26 Sept 16	Yes
Navigating to the table and coming back to counter (Black line following) using grid system.	Manjunath K	04 Oct 16	Yes
Avoiding collisions using IR sensors and proximity sensors and rerouting in case of obstacle detection using exponential backoff technique with different delays and finding new route using rerouting algorithm.	Manjunath K Amit Pathania	15 Oct 16	Yes

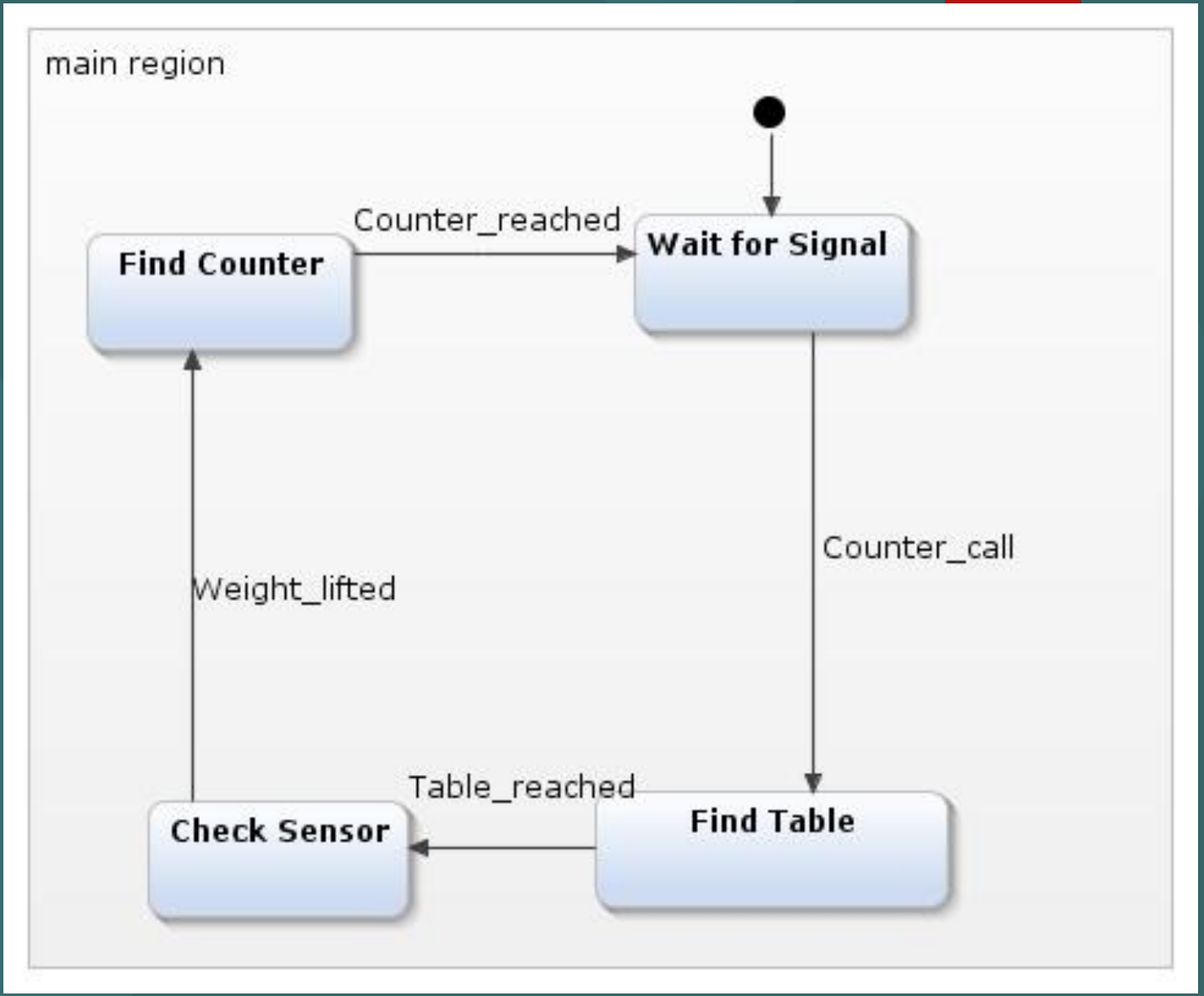
States:

- Wait for Signal state : Robot will looking for incoming signals on Zigbee during this state.

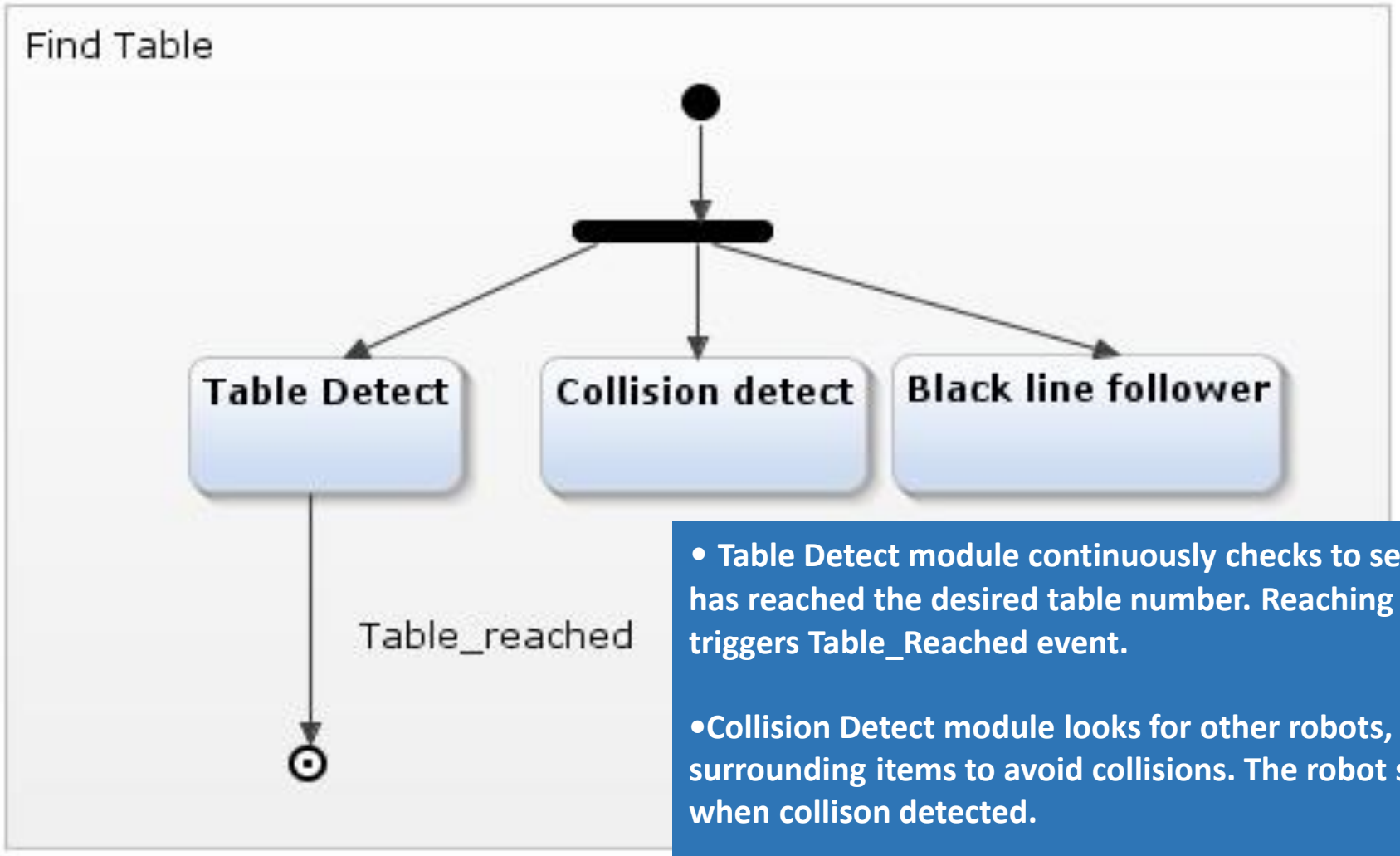
- Find Table state carries out all activities to find the table where order has to be delivered.

- Check Sensor state provides functionality to check the order kept on the robot once customer table is reached.

- Find Counter state carries out all activities to find the counter



Find Table State

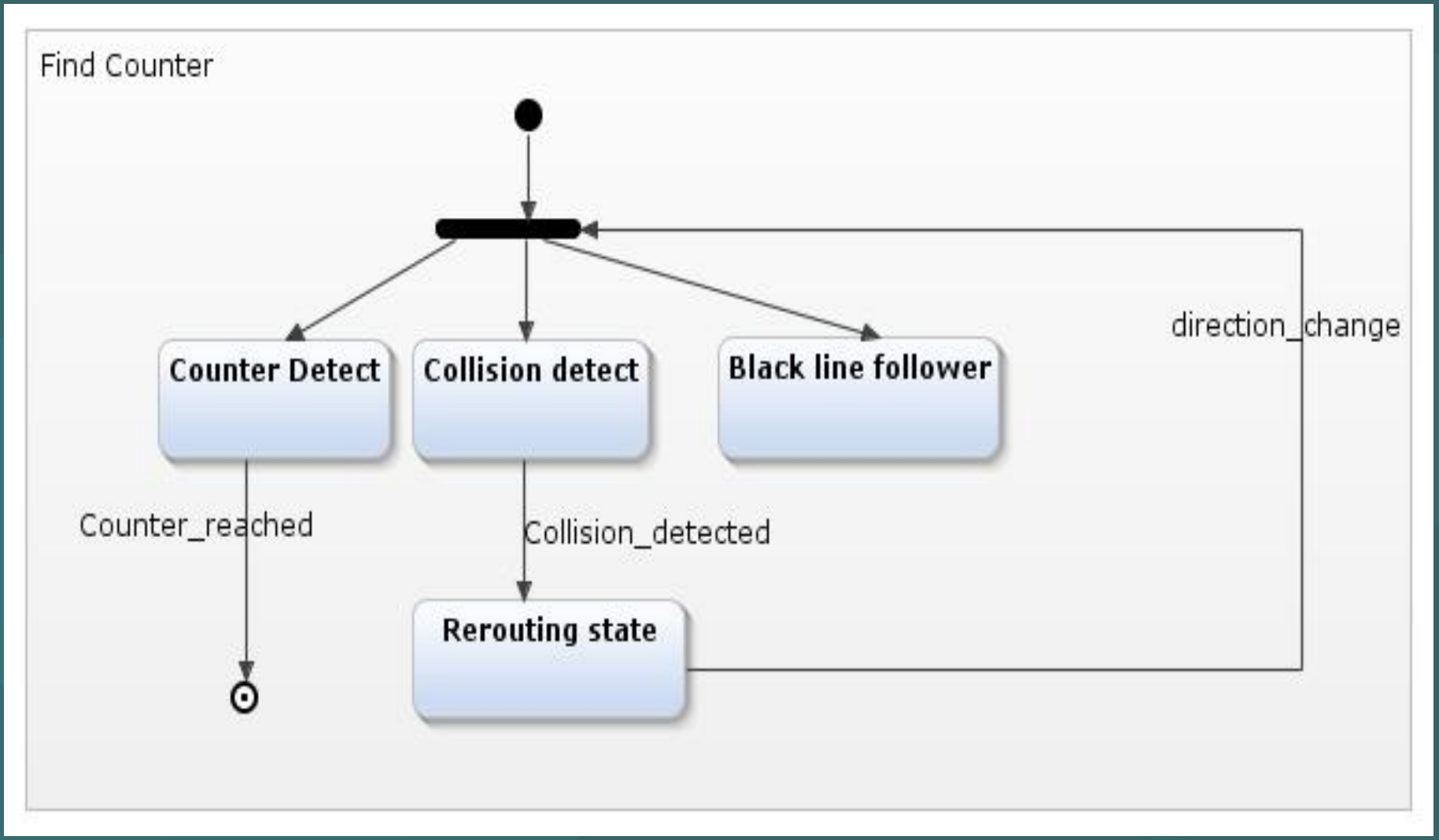


- Table Detect module continuously checks to see whether robot has reached the desired table number. Reaching table causes triggers Table_Reached event.

- Collision Detect module looks for other robots, people and surrounding items to avoid collisions. The robot stops and waits when collison detected.

- Black Line Follower module carries out activities needed to keep the robot moving along the specified pre defined path to each table.

Find Counter State



Innovation & challenges

- Integrating PHP, python and Zigbee.
- Polling between the bots and communicating with specific bot.
- Recalculating path in case of collision detection.

TASK COMPLETED

TASK	PROBLEM FACED	SOLUTION
Making an android app.		
Installing WAMP server and maintaining database of menu items, orders placed and delivered.		
Passing destination table coordinates to a free bot from admin webpage using python.	Calling python script multiple times from php	Two scripts were used: One to store table number in fiel and other to read file and pass table number using Zigbee.
Communicating with bots using Zigbee.	Polling between the bots and communicating with specific bot.	Using Round robin scheduling technique. Using botID with each message.

TASK COMPLETED(Contd..)

TASK	PROBLEM FACED	SOLUTION
Navigating to the table and coming back to counter (Black line following) using grid system.		
Avoiding collisions using IR sensors and proximity sensors and rerouting in case of obstacle detection using exponential backoff technique with different delays and finding new route using rerouting algorithm.	IR sensor values keep changing as per time of day/intensity of sunlight making it difficult to detect collisions. Rerouting algorithm to avoid deadlocks.	Threshold values were passed using global variable. Used different delays for each bot and giving more priority during onward route than backward route.

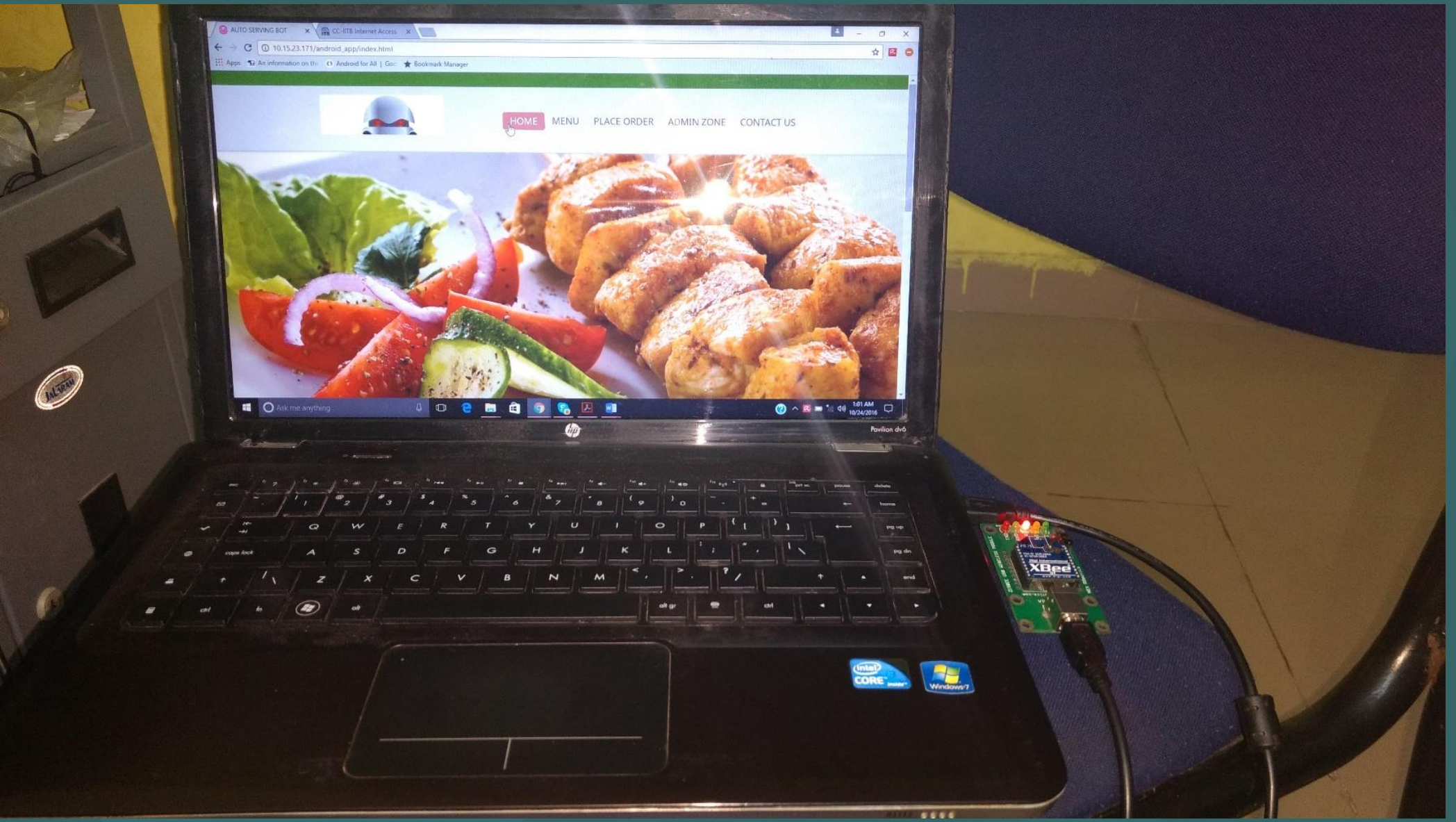
TESTING

TEST CRITERIA	TEST DESCRIPTION	TEST RESULT
Testing Android app	Placing new order using android app.	Order successfully placed using android app.
Polling between the bots and passing destination table coordinates to a free bot from admin PHP webpage using python	Selecting the order on webpage and pressing orderID.	The free bot responding, getting activated and moving to desired table.
Collision detection when bot going to destination table.	Placing obstacle enroute while bot going towards table.	Bot stops and waits for fixed duration and checks again.
Collision detection when bot going to counter after order delivery.	Placing obstacle enroute while bot going back to counter.	Bot stops and waits for fixed duration and checks again, if obstacle still detects turns back and recalculates return path.

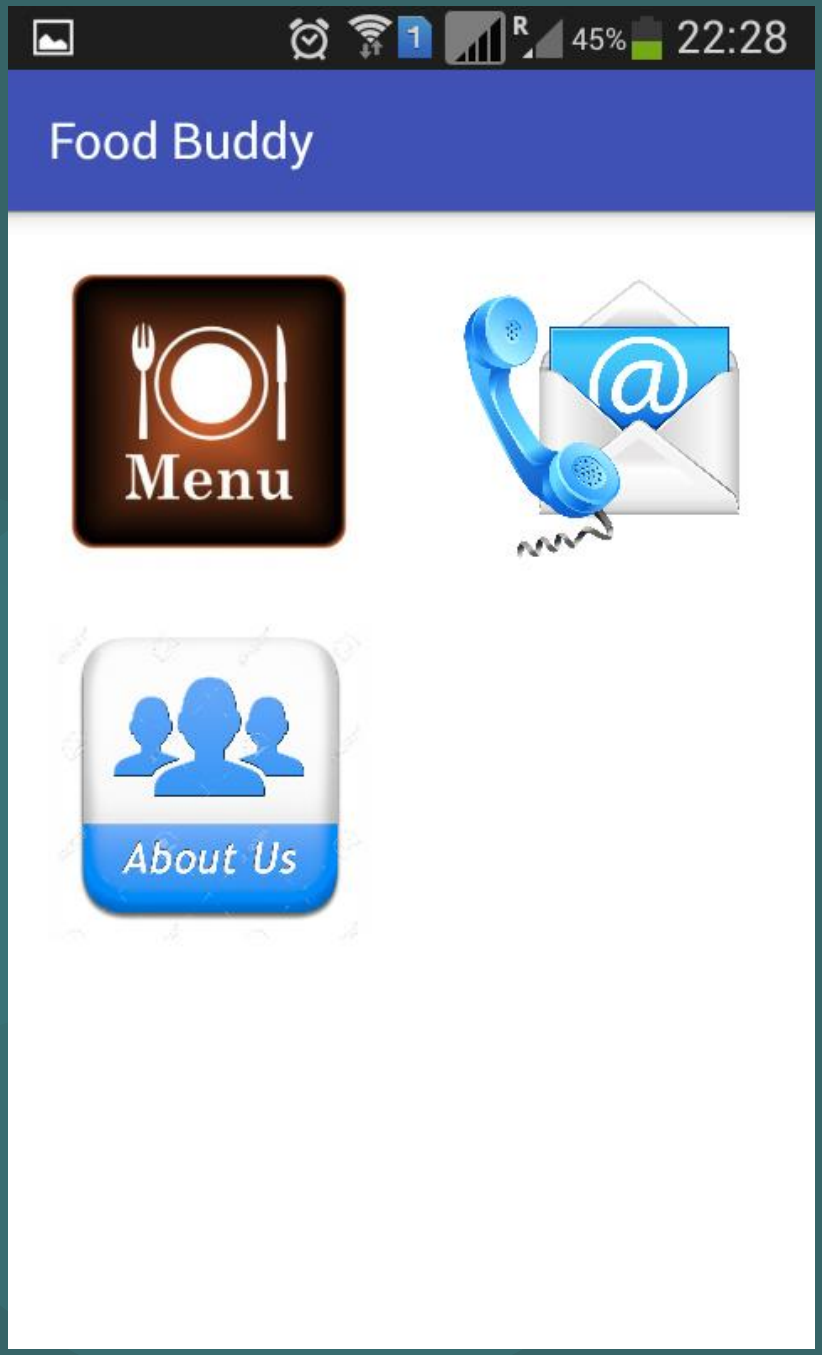
TESTING

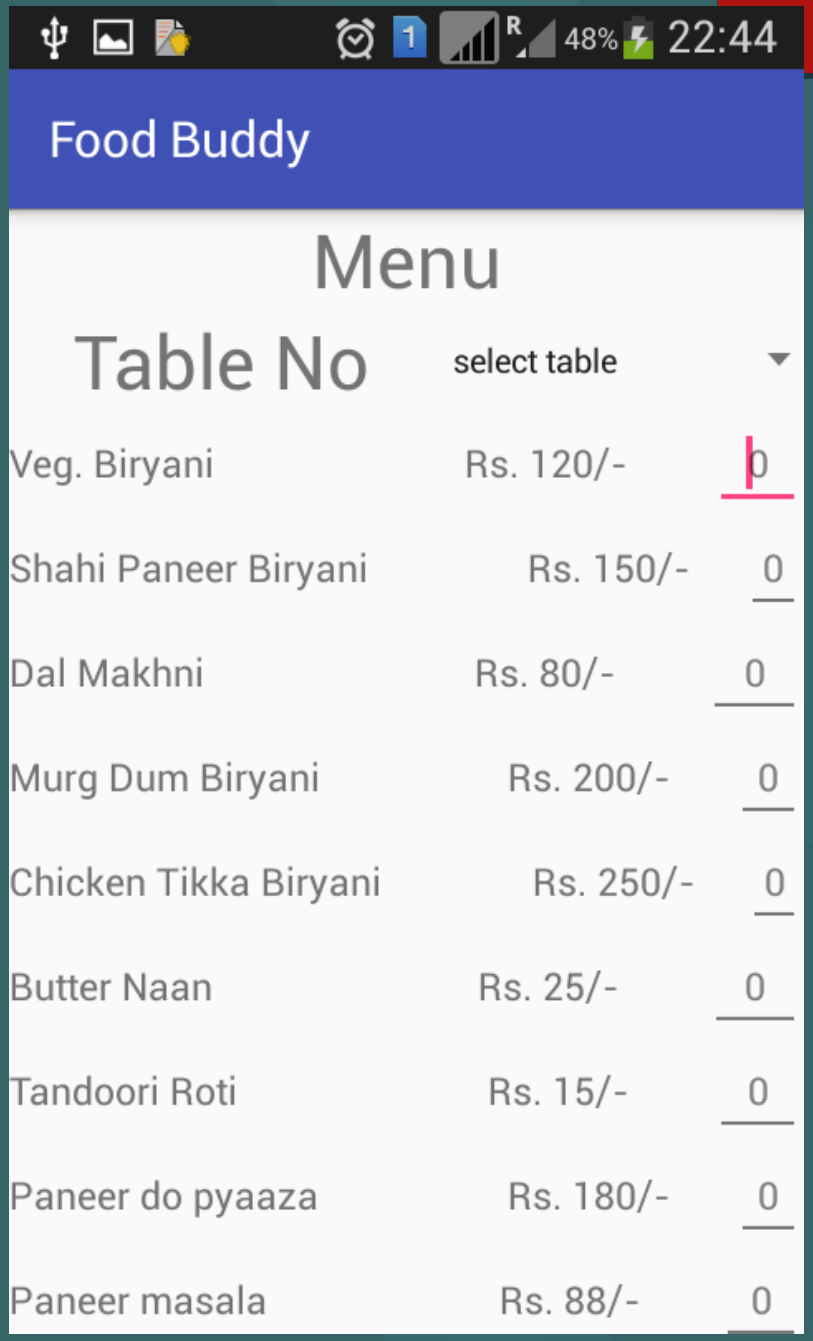
TEST CRITERIA	TEST DESCRIPTION	TEST RESULT
Dynamic tasking of bots.	Placing multiple orders.	When the bot reaches the start point, the bot is free and ready to receive orders.As and when bot is free, the bot responds back to server during polling using Zigbee and the order is given to the bot. The process is repeated till all orders are delivered.
Checking order picked up by the customer.	Placing serving plate on bot.	Once the bot reaches the destination table,it waits for palte to be lifted and gives alarm if plate not lifted even after fixed duration.

TESTING

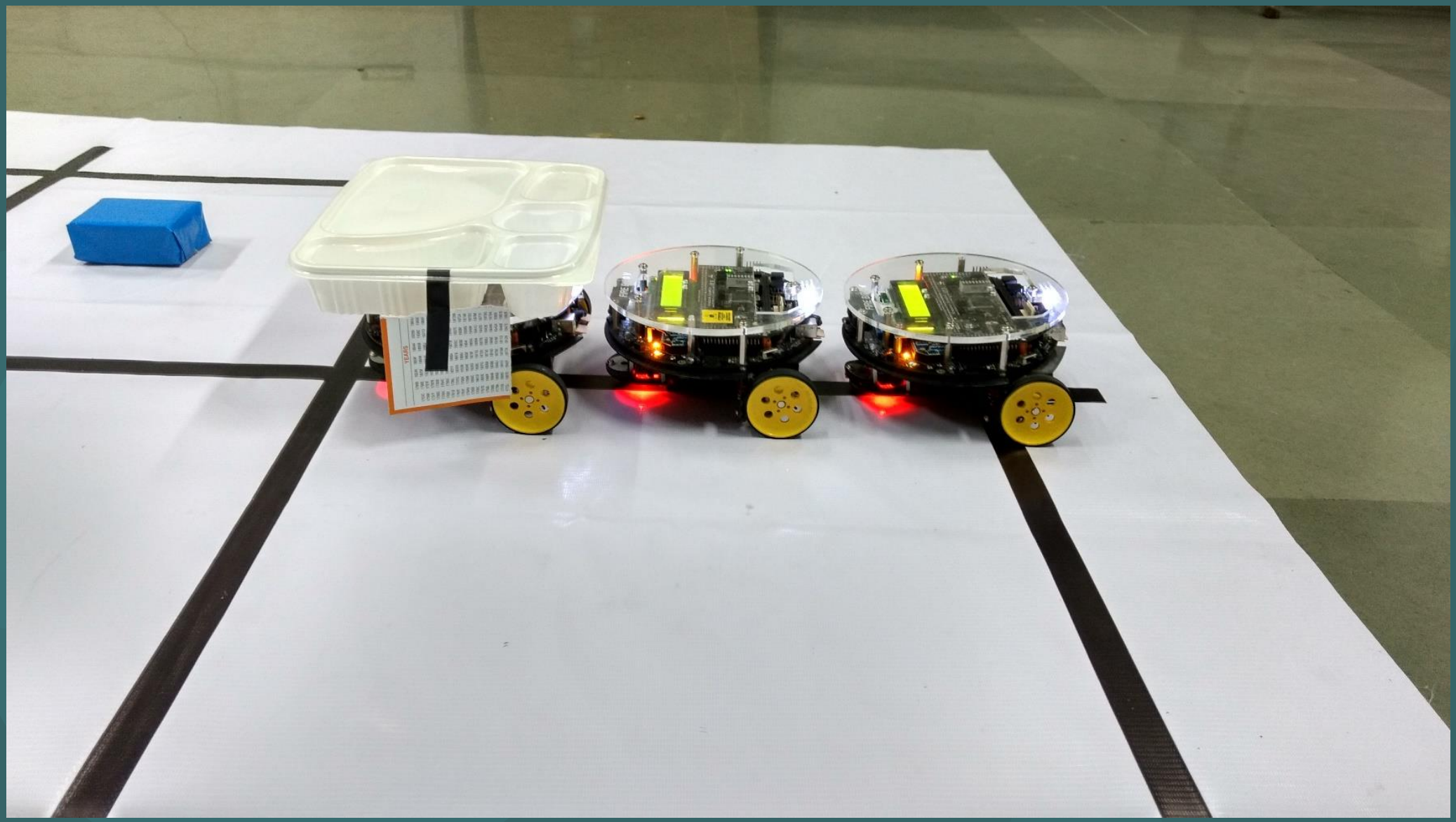

AUTO SERVING BOT
CC-IITB Internet Access
10.15.23.171/android_app/index.html
HOME
MENU
PLACE ORDER
ADMIN ZONE
CONTACT US
Ask me anything
1:01 AM
10/24/2016
Pavilion dv6
XBee

TESTING

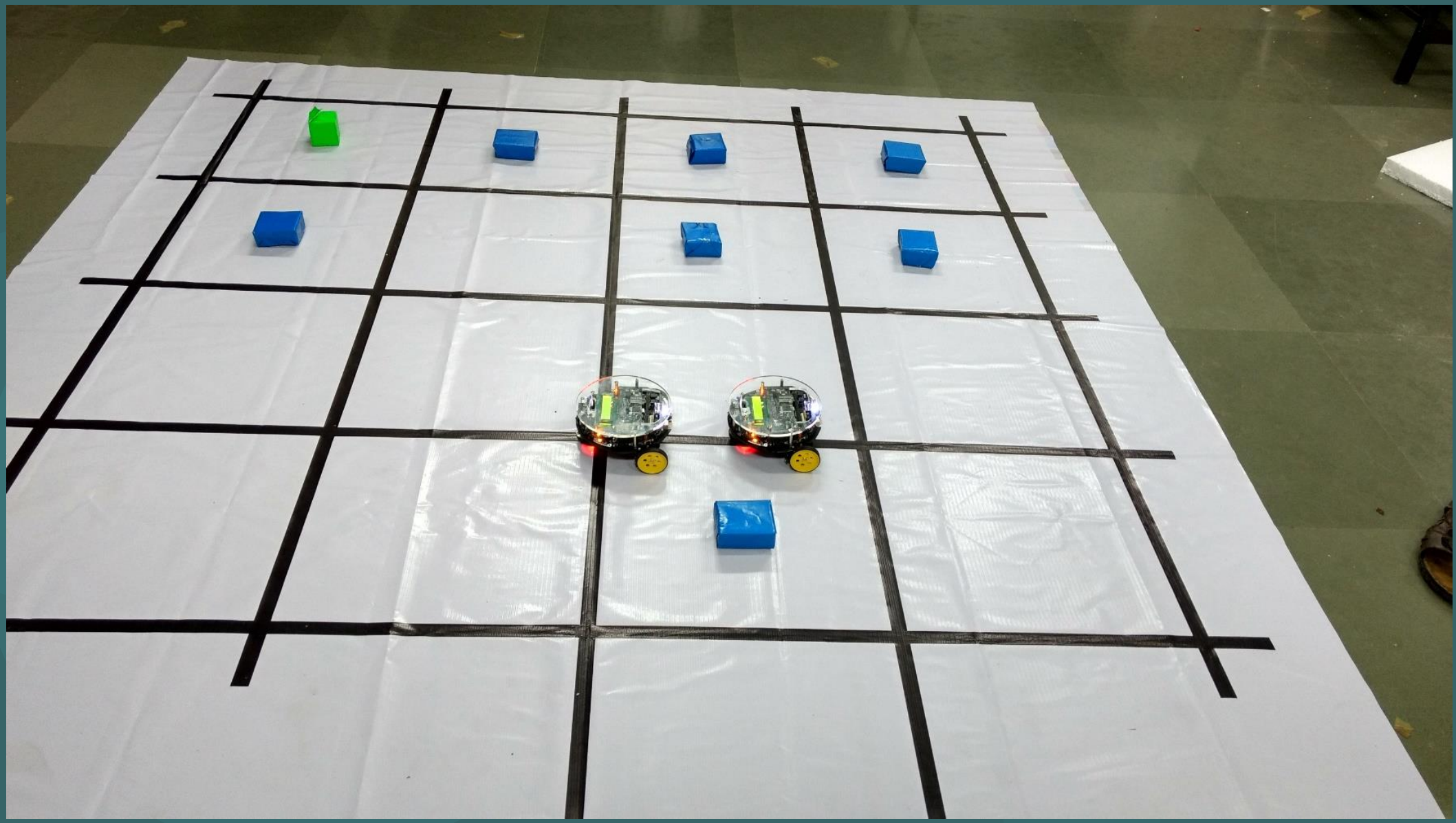

45%
22:28
Food Buddy
Menu
About Us


48%
22:44
Food Buddy
Menu
Table No
select table
Veg. Biryani
Rs. 120/-
0
Shahi Paneer Biryani
Rs. 150/-
0
Dal Makhni
Rs. 80/-
0
Murg Dum Biryani
Rs. 200/-
0
Chicken Tikka Biryani
Rs. 250/-
0
Butter Naan
Rs. 25/-
0
Tandoori Roti
Rs. 15/-
0
Paneer do pyaaza
Rs. 180/-
0
Paneer masala
Rs. 88/-
0

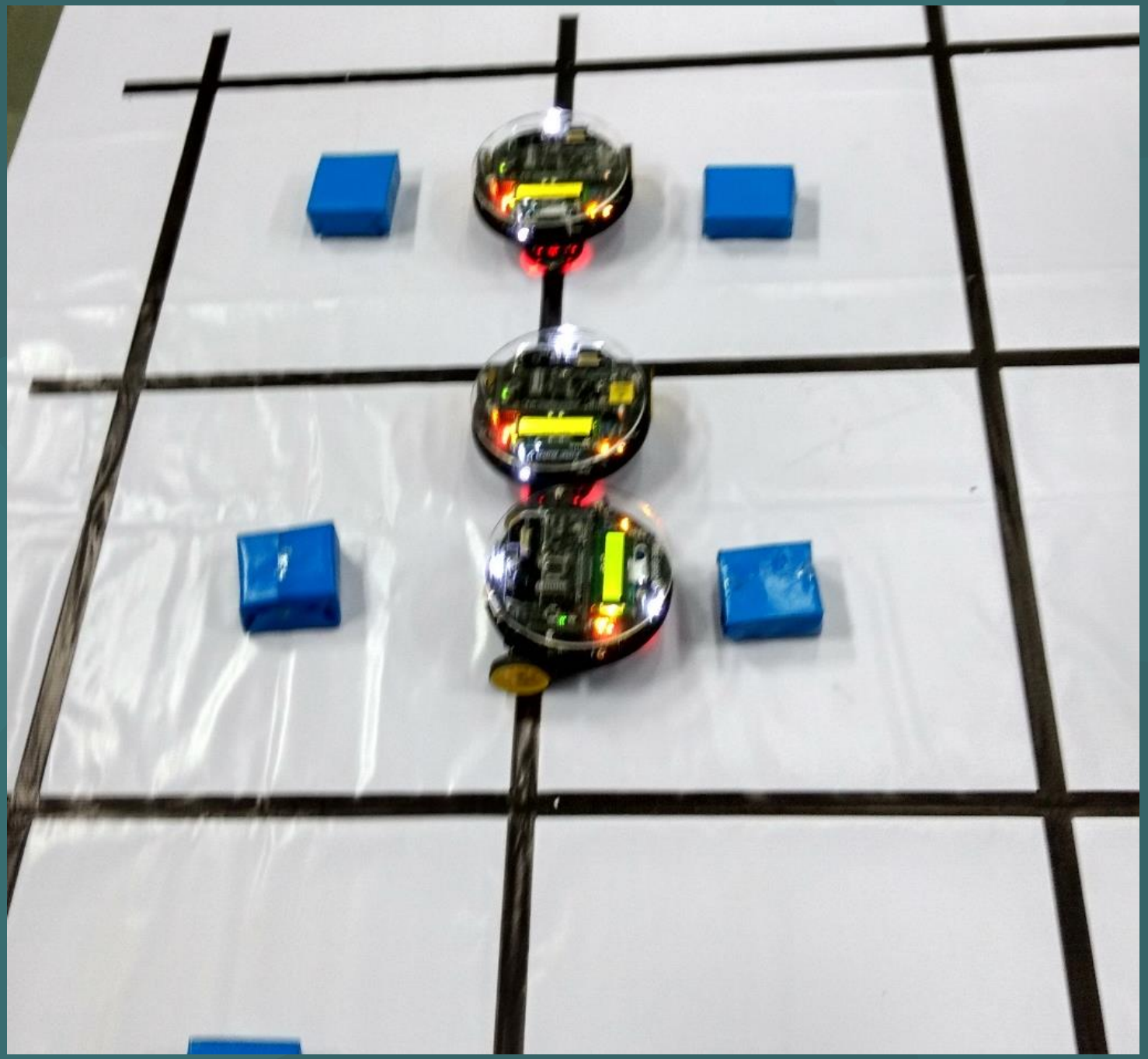
TESTING



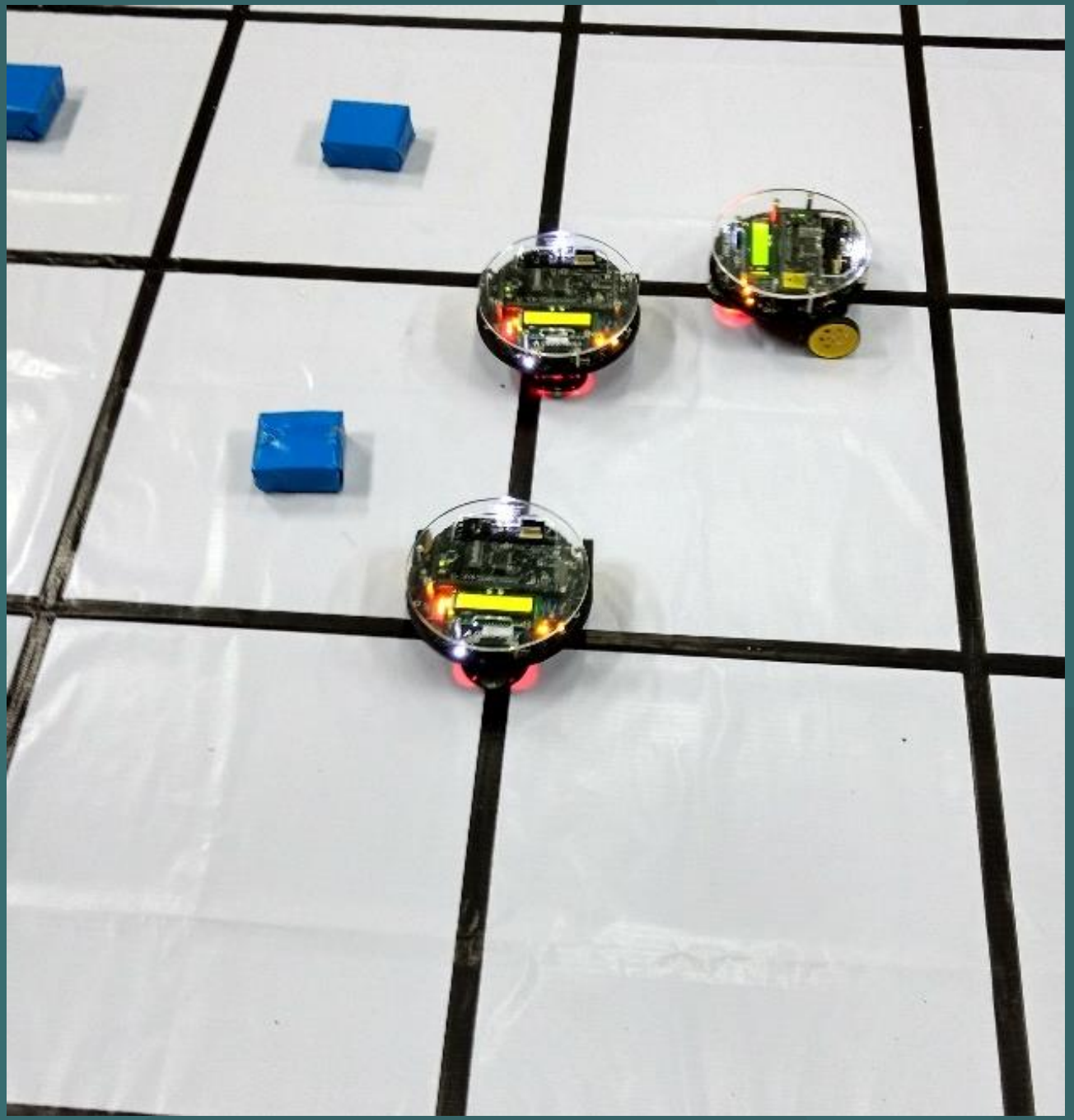
TESTING



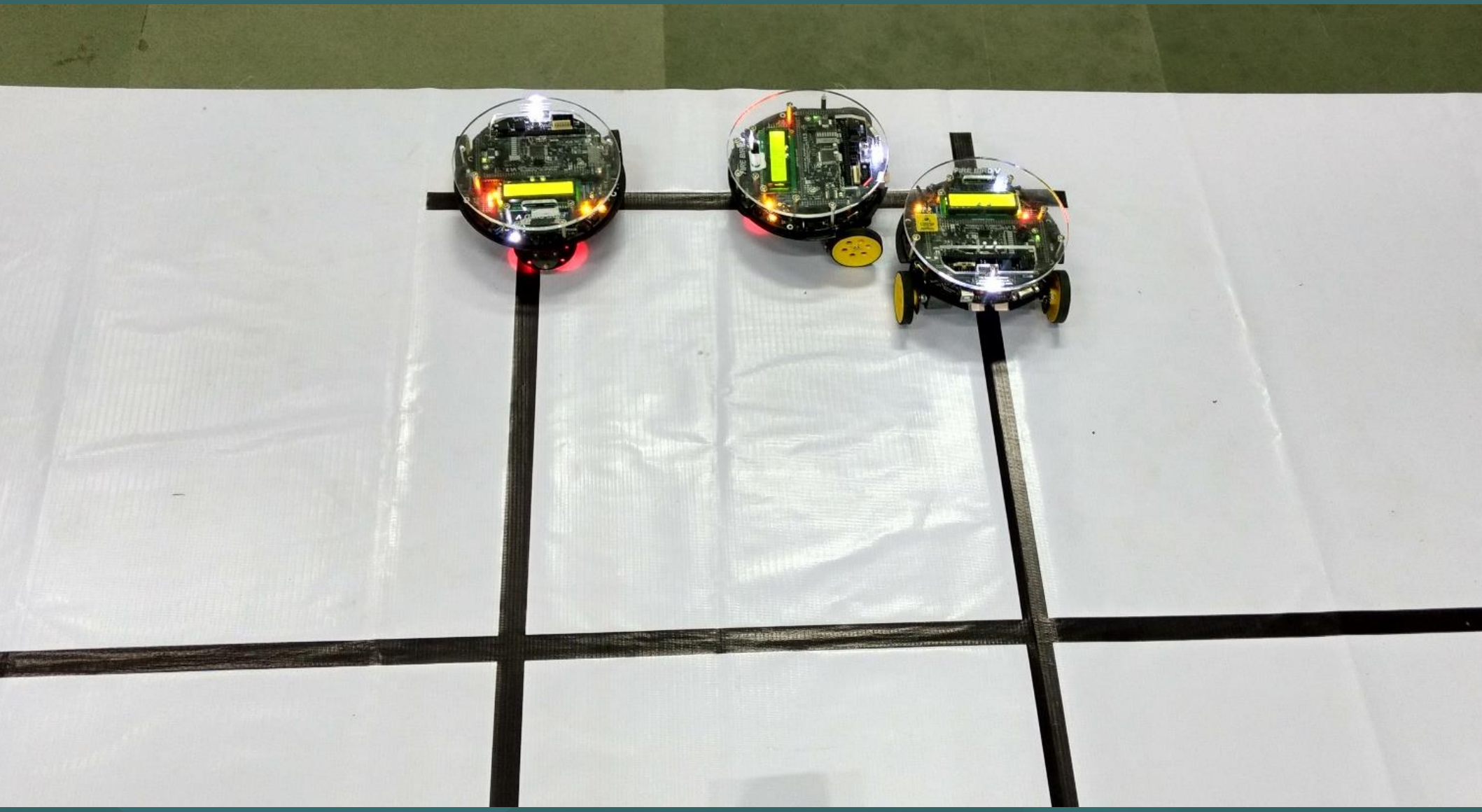
TESTING



TESTING



TESTING



Reusability

- Followed coding standards by using context variables, commenting guidelines, modularity by different functions and multiple files.

- Use of open source softwares.(Python, WAMP server, Android studio,AVR studio)

Future Enhancement

- Use of multiple counters. – More complexity and decision making.
- Tracking the live position of bots.
- Tracking order status using Android App.
- Using weight sensors to detect order lifted at the table.
- Using mechanical arm to place order on the table.


THANK YOU!