The Restaurant Bot

Shubham Jain (120050002)
Palash Kala (120050010)
Vikas Garg (120050017)
Amol Agarwal (120110031)

Project Description

- Problem Statement:
  - When we go to restaurants, we order our food and sometimes it takes a lot of time to deliver and we have to wait for waiters to deliver our food even it is cooked
  - Also it’s costly for restaurant owners to put more and more waiters and it will create a lot of hassle also
- Description
  - We want to have restaurant bots instead of waiters to deliver the food to the customer
  - So people can order through a web interface
  - Food will be prepared accordingly and given to our restaurant bots
  - The restaurant bots will then serve the food to the tables

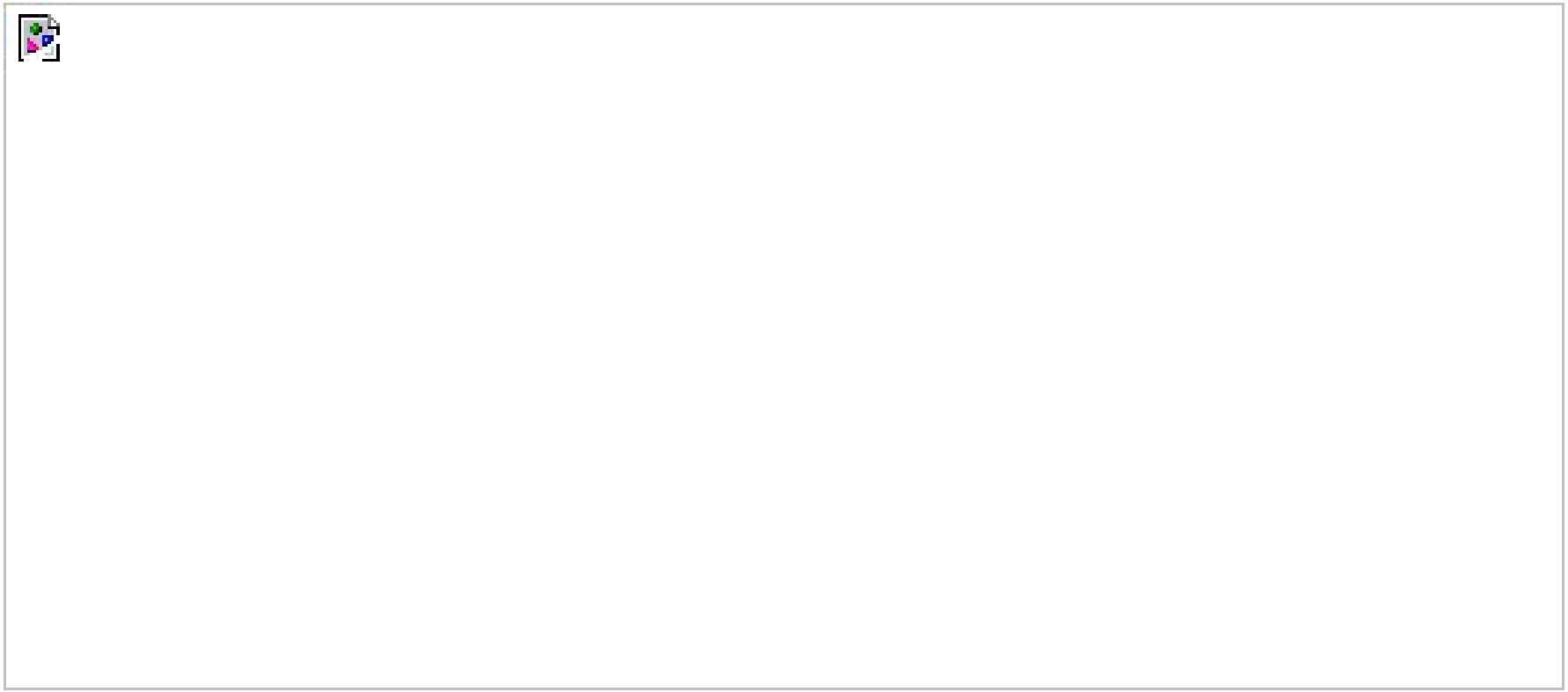
Requirements/Task Specifications, aimed for

- Communicating with bots from central server via Zigbee Protocol
- A web app which can automate the process of receiving and allotting orders
- The bot should push the food with the help of servo motors on tables places on the arena
- Three bots and the central server should automatically optimize the servicing of food
- The bots should not collide with each other, when they serve the food
- A GUI of tables should be present on the web app, where customers can click on their own table to order

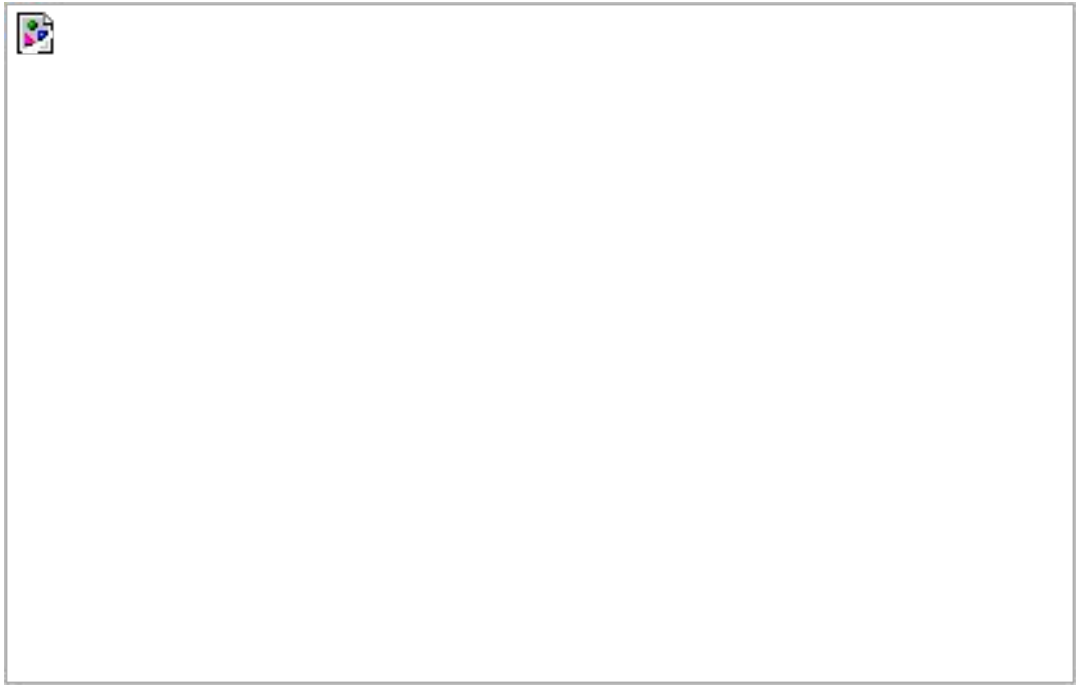
Project Plan

Subtasks for Stage 1	Date of completion	Critical?	Person
Print Arena	12/3/2016		Vikas
Robotic arm design and interfacing	5/4/2016	yes	Shubham,Vikas
Ordering from Web Interface and putting it in database	22/3/2016	yes	Shubham,Vikas
Interfacing Infrared sensors	22/3/2016		Amol,Palash
Communicating with multiple bots via Zigbee	2/4/2016	yes	Palash, Amol
Moving bots on the white line	5/4/2016	yes	Palash, Amol
Interfacing of python module with the web app	9/4/2016	yes	Shubham

Block Diagram



Simple State Machine



Can be understood after looking at the code. The transitions are the signals given to the bot.

Innovation and Challenges

- Innovations:
  - White line following with 90 degree turns
  - Creating a Zigbee network with heuristics
- Challenges
  - Calibration, battery problems and optimum external light for the bot to run smoothly
  - Interfacing of Python module with Web API as well as the FIREBIRD(bot) code
  - Pushing the objects at the correct location after reaching a node

Tasks Completed

- Communicating with bots from central server via Zigbee Protocol
  - Separate signal to each bot wasn't possible
    - Created a heuristic where both the bots got different command signals and there was no interference in a bot's motion due to a signal given to the other bot
  - Bot could not send signals to the server on its own, created buffer problems
    - We created a signal to send to bot, it replies with the current mode it is in
    - After checking the mode, the server replies with the action to take
    - Hence, it was a three way signaling, rather than a two way signaling
- A web app for receiving orders and another portal for restaurant owners to allot orders to the bots
  - Web app for automatic receiving and serving was difficult because, proper scheduling and flawless motion of the bot was required for that which was not the case
    - Hence, we divided the problem into two portals, one for ordering and other for allotment
    - Allotment is done manually by the restaurant owner

Tasks completed continued

- The bot pushes food the node, but tables are not fixed
  - The placement of the table could not be done because it could come in the way of the bot
    - We placed the table (a dummy) then the bot reached the node
- The bots do not collide, but deadlocks haven't been avoided
  - Paths of the two bots intersected
    - We had the path beforehand for both bots and if both bots had same next node, we gave the first bot higher preference and let him complete the path first, while the other bot waited
    - When both the bots had their next node as the other bot's current location, we haven't avoid this deadlock
- People can choose their table number and item number from a drop down list
  - People had to choose their order & table number from good GUI, where they could see their table
    - Instead, we had table numbers on the table and customers could choose their table from a dropdown

Test Plan/Cases

- To test the system,we started the server. We ordered for a table from the interface. The bots were kept at their initial position.
- when we assigned the orders to a bot, the order was assigned and bot moved. We kept a piece of thermocol which was treated as food and pushed by the robotic arm.
- To test XBee module we used XCTU software.
- We used it to configure the XBee modules and send it a frame of packet and check does the bot receive it and how does it react to it and thus verify the code on bot.

Test Plan/Cases (contd.)

- To test the web module, we run it on a local host and use SQLite Database Browser to see if changes happen in database and whether our App works.
- To test the APIs we used POSTMAN extension on Google Chrome.

Performance metrics

The following are the performance metrics for the system:

- Time in serving an order
- Accuracy in serving food, movements
- Serving at right co-ordinates/table

Reusability features

- All components are reusable
- Web app can be reused for some other purpose as the framework is modular and all functions maintain modularity which can be changed a bit and reused
- We have created a class to represent a Bot, Map and XBee communicator module which can be reused with any other program
- The code of line follower can be used for line following
- The robotic arm code can reused for movement and learning about robotic arm
- The method used for XBee configuration and for communication can be used for communicating amongst various bots

Future Enhancements

- Extending the system to handle more than 2 bots
- Improving the food serving mechanism, so that it serves accurately on the assigned table without manual readjustments
- Collision detection using Sharp sensors
- Currently a bot is assigned only one table at a time and has to return to the kitchen after doing so. It can be extended to deal with 2 orders at a time and even more if we can somehow figure out food placement on the bot
- Deadlock avoidance in collision avoidance