CS-308-2014 Final Report

Autonomous Navigation

FlipFlop

TH-09

1. Pulkit Bansal **100050045**
2. Kumar Pallav **100050046**
3. Abhishek Anand **100050047**
4. Rishabh Kumar **100050050**

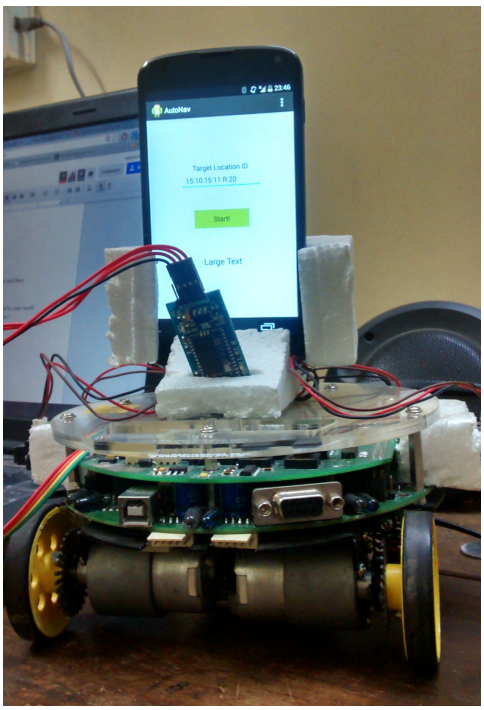


Table of Contents

1. Introduction

The project is about **Autonomous Navigation System.** The bot automatically navigates through a demo city to reach the destination. The city has roads like rows and columns of a bigger matrix(city). Each intersection of rows & columns is identified as a junction and given a unique id to identify the junction. Each junction has a QR code attached to it that has the details of the junction. At junctions, the bot reads QR codes to identify its current location. Now, knowing the current location and destination, bot calculates the path along which it should move to reach towards the destination, and so on at each junction.

Some tasks the system already performed in tests included ***move-on-route*** and ***obstacle detection***.

The project may be very useful in context of a greenhouse. All the plant troughs are labelled with a unique ID assigned to them. The label also contains other information related to that particular trough like plant variety, frequency of watering, etc. All this information is stored in the form of a QR code. The bot is given a destination trough Unique ID where it needs to go. It moves on the path making decisions based on the road signs and moving in the appropriate directions. After reaching its destination, the bot upon reading the QR-code can perform tasks scheduled for that plant trough.

2. Problem Statement

Product Perspective-

In a city which has many junctions which are represented by code (a:b:c:d:e:f) where a & b represent row & column of junction intersection, c represent whether row or column would change when bot moves forward , d represent sign(+/-) whether row/column will increase or decrease, e represent whether row or column would be affected when bot moves in right direction , f represent sign(+/-) whether row/column will increase or decrease.The bot is given a destination trough Unique ID (a:b:c:d:e:f) where it needs to go.This Unique ID stores information in the format; a:b represent row and column of first intersection, c:d represent row and column of second intersection. These variables combined store on which street (b/w which two intersections) the building lies. e stores the direction(right or left) on which the building lies when bot moves from first to second intersection.

It moves on the path making decisions based on the junction QR code and moving in the appropriate directions decided by path finding algorithm.

Product Functions-

1. Recognizing a QR-Code by application on the phone and decoding it.
2. Processing on side of phone and information is sent back to bot via bluetooth module.
3. Make appropriate navigation decision on the application side on the phone based on the decoded information as to which turn to take etc.

4. Maintain safety i.e. avoid collisions with side walls and a obstacle which it encounters suddenly in path to destination.
5. QR-Code : Code contains information regarding junction.It tells the bot the relative position of junction which helps bot in decision making process to reach destination

User Characteristics-

1. Arena has to build comprising of many junctions and turning points with proper distance maintained between the walls of arena.
2. Users need to install QR-Codes at the junctions in the arena.
3. Users need to input the destination in the app on the phone which is present on the bot

Constraints-

1. There should exist a path to the destination
2. The bot should not go off-road
3. The bot should finally reach its destination
4. The QR-Code should be readable by the bot (lighting, focus etc.)

Assumptions and Dependencies

1. The bot is dependent on decoding of QR code which is captured by camera on the phone
2. Distance between side walls in arena should be enough so that side sensors detect the side distance accurately.

3. Requirements

3.1 Functional Requirements:

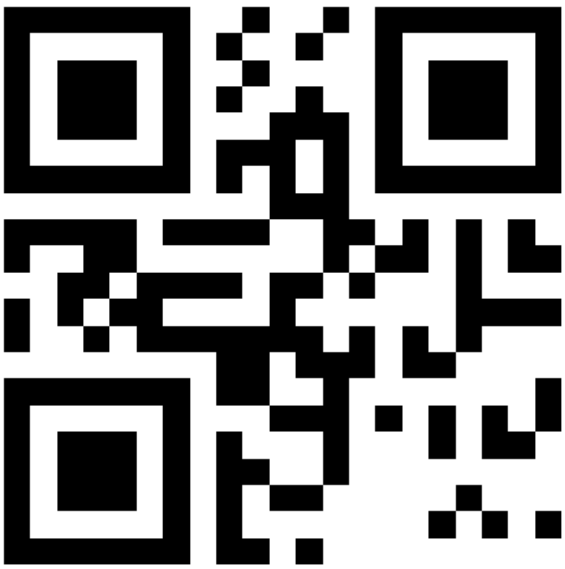
3.1.1 Basic requirement of our autonomous bot is to take a destination and reach it reading QR codes and taking decisions along the path.

3.1.2 QR codes are present at junctions as well as on plants for their unique identification. QR code at junction gives following information in the form (a:b:c:d:e:f)

i) **a**: row of junction intersection
ii) **b**: column of junction intersection
iii) **c:** represent whether row or column would change when bot moves forward
iv) **d:** represent sign(+/-) whether row/column will increase or decrease when iii)
v) **e:** represent whether row or column would be affected when bot moves in right direction
vi) **f:** represent sign(+/-) whether row/column will increase or decrease when v)

For example: 16:10:R:-:C:-

The junction is denoted by row 16 & column 10. On moving in forward direction on the junction row value is decreases. On moving in right direction on the junction, column value decreases.



16:10:R:-:C:-

3.1.3 **Capturing and decoding of QR code** - Image of QR code is captured using mobile camera. An android app decodes the image and calculates the direction to be taken next. QR-code is identified and decoded using Zxing decoder.Here is API that we are using for identification and decoding of QR code (https://github.com/zxing/zxing).

3.1.4 This direction is sent to the bot using **bluetooth.** Bot now moves in the direction it recieves from the android app until it reaches the next junction. The bot avoids collision with objects in path using SHARP sensors.
Currently, there is one bot in the system and we are not considering inter-bot collisions.

3.2 Non-Functional Requirements

3.2.1 **Execution qualities:-** User has to initiate with arena establishment if he/she wants to test working of bot. Arena involves putting up the walls keeping in mind distance constraints between the walls.QR codes which have junction information encoded in it should be there on junction at a sufficient distance so that bot can pass easily with camera on the top of it.
Now User will put some destination on app based on the protocol that is described earlier and can put bot anywhere in arena making sure that a valid path should exist from source to destination.

3.2.2 **Evolution qualities:-**Testing can be done in various steps.First we can check for smooth movement between the side walls.Then we can check whether camera is put correctly so that QR code is detected with that height.Now we can also look whether bot is taking turn as expected to check whether bot has properly decoded the QR code.We can also test for obstacle detection by putting any obstacle on path which our bot is following to acheive destination.
Extensibility is mentioned in future work of project.

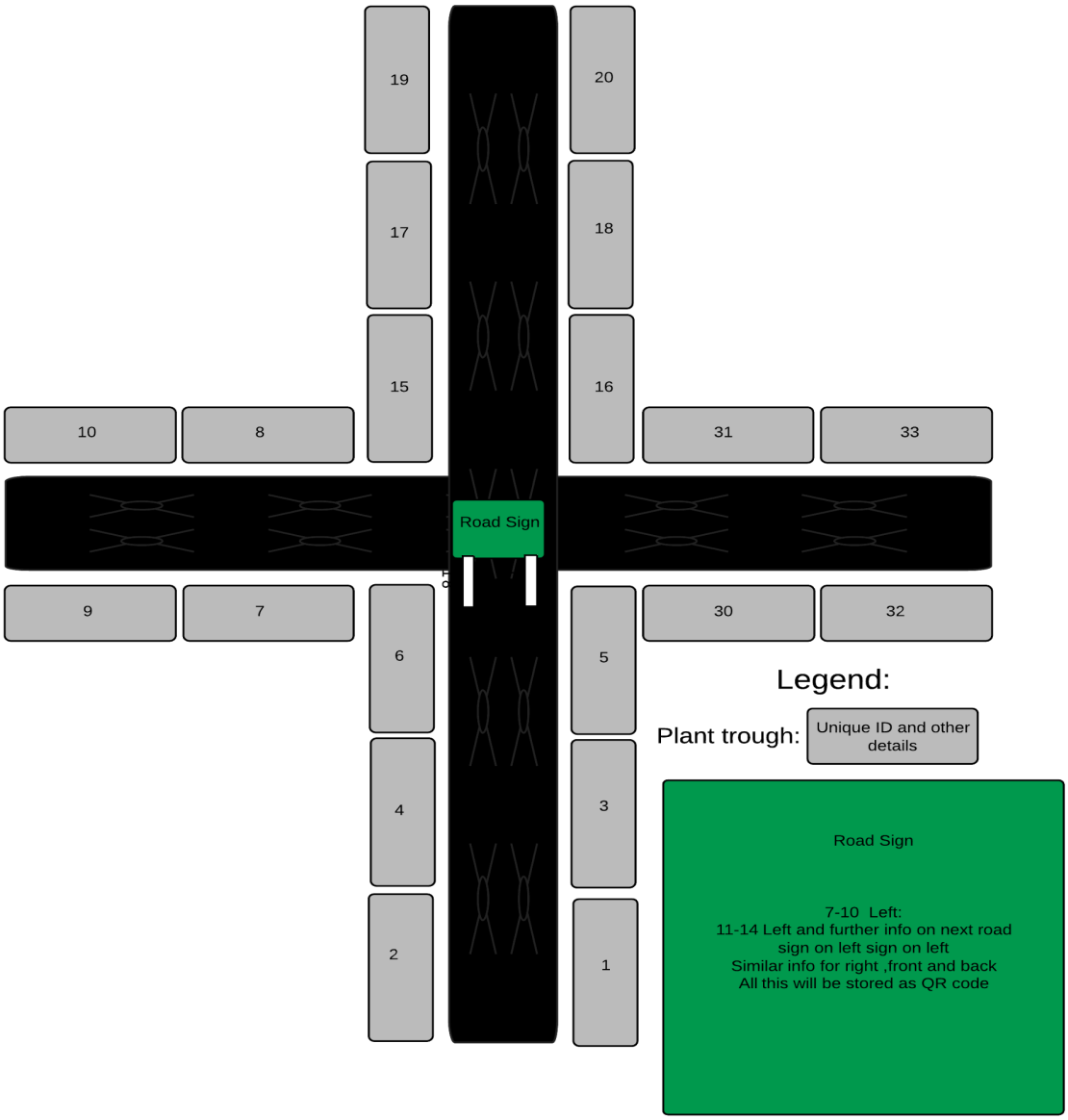


Figure 1.1, showing the arena with junctions having road signs containing QR code.

3.3 Hardware Requirements

i) Camera: Mobile Camera
ii) Sharp Sensors: 4 in number (2Y0A21)
iii) Firebird V
iv) Laptop with appropriate softwares involved

3.4 Software Requirements

i) Keil: **µvision V4.03**
ii) Android Studio
iv) C & Java language

4. System Design

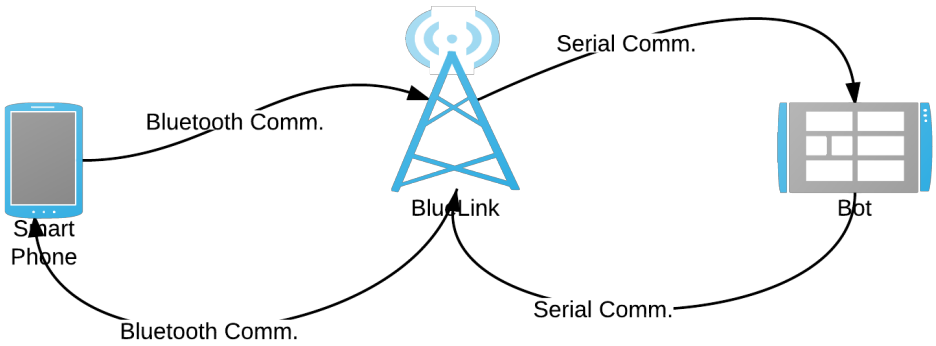


Figure showing an architecture diagram describing the components of the system and their relationships

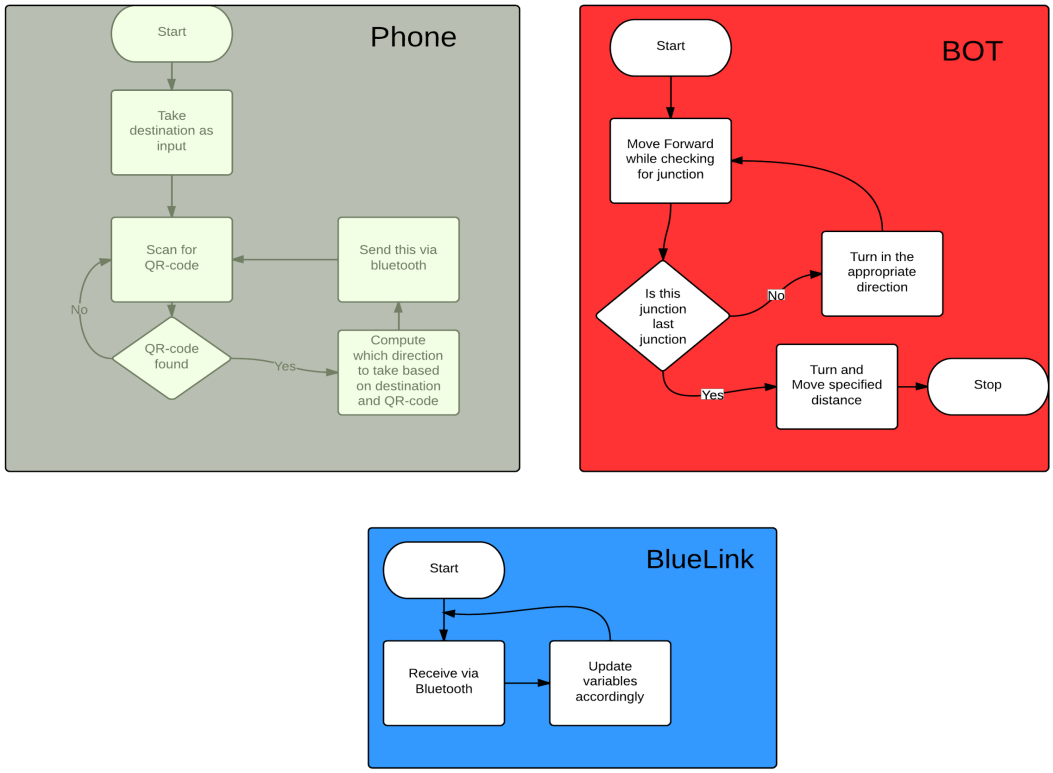
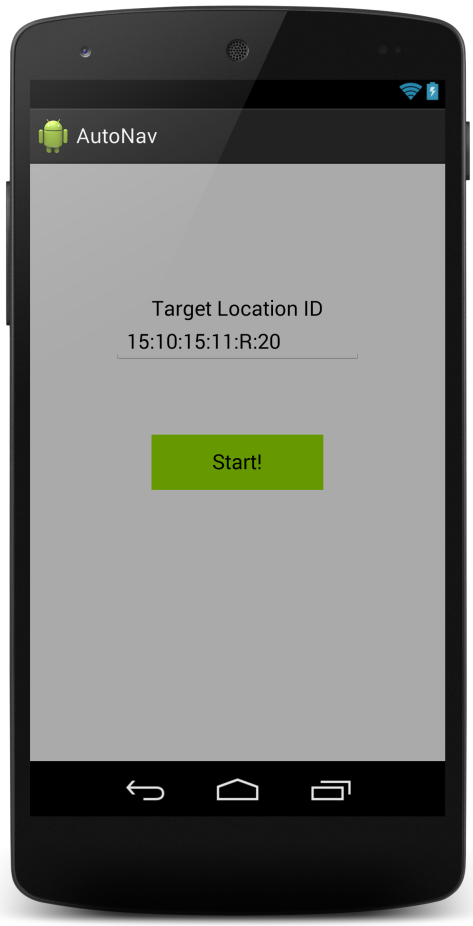


Figure showing FSMs of the Bot & Phone

Our system consists of following components:

i) Android App "AutoNav" *(built on Android Studio for Android 2.3 and above)*

It has a text field to input the target location ID. When the start button on the app is pressed, the app goes in continuous scanning mode. Now, the bot moves along a straight path looking for a QR-code on the junctions. QR-code is scanned, decoded and next turn is calculated before the bot reaches the junction. So, at the junction bot knows which turn to take and moves smoothly in that particular direction. After decoding one QR-code, app again goes in continuous scanning mode, looking for the next QR-code.



ii) **Bot**

It is a FireBird V robot. An android device is mounted on the top of bot which has the above app running. Bot moves-on-route, detecting obstacles and avoiding them using 4 Sharp range sensors (2Y0A21).

5. Working of the System and Test results

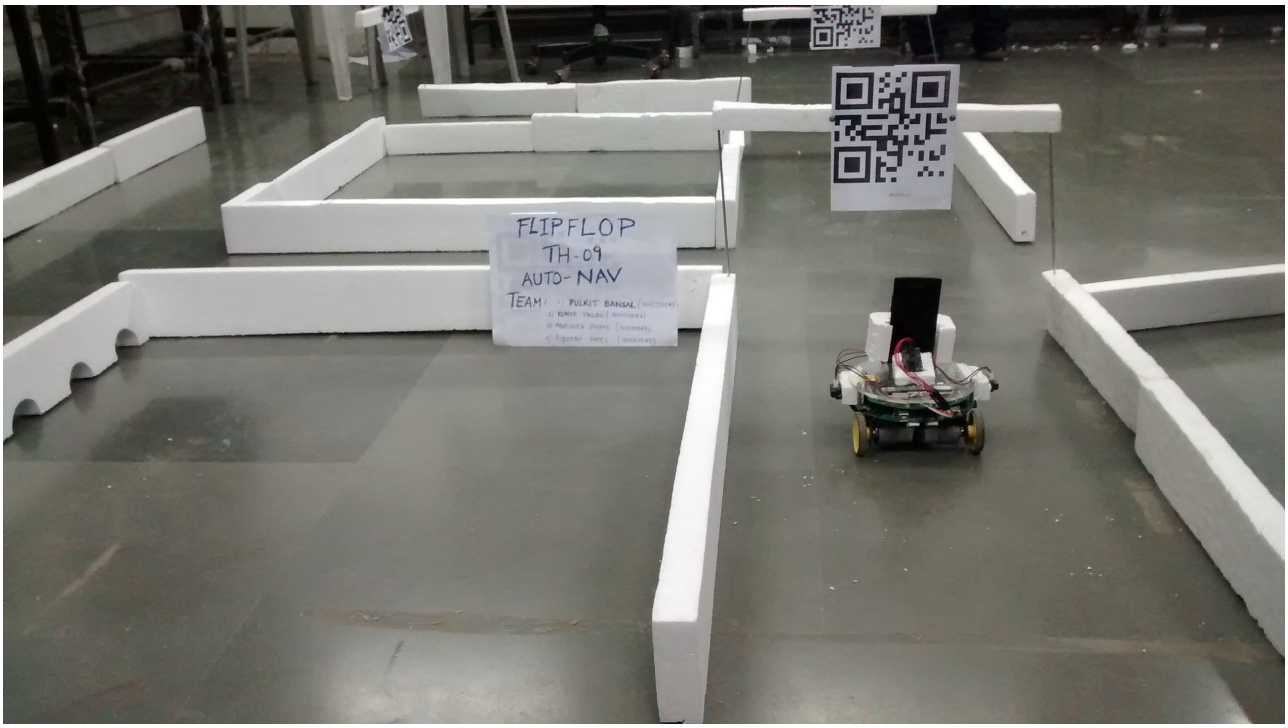
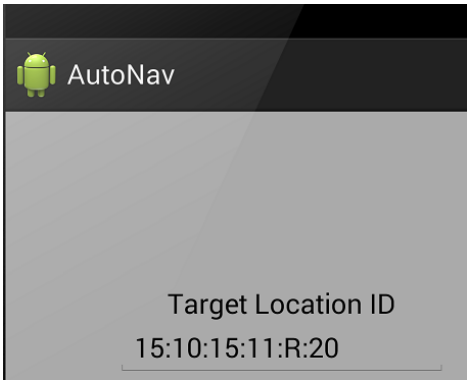


Figure showing the Arena setup

The system works as follows:
i) The destination ID is given as an input to a text field in the android app.



ii) The android device is mounted on the bot and start button (on the app) is pressed.

iii) Now, the bot moves along a straight path looking for a QR-code on the junction. QR-code is scanned, decoded using "Barcode Scanner" android app (downloaded from Play Store). Now knowing the destination & current junction id, the next turn is calculated.

iv) Now the app sends the bot the direction ("R/L/F") it should take at the junction. Reading character 'R' bot moves right and so on. Also, if the junction is that of the final destination app also sends the distance it should move before stopping along with the direction.
So, at the junction bot knows which turn to take and moves smoothly in that particular direction.

iv) After decoding one QR-code, app again goes in continuous scanning mode, looking for the next QR-code.

v) Bot reaches the destination junction, moves the specified distance, stops and beeps to indicate destination is reached.

<For each functional requirement mentioned in the SRS, mention how you tested it to ensure the functionality worked properly.>

S. No.	**Requirement**	**Strategy adopted during testing**
1.	The image captured completely contains the QR code	Android device scanned QR-codes in full screen mode. Also, the speed of the bot & length of the path before junction was so adjusted that QR-code scanning could be done for sufficient time.
2.	Decoding the image containing QR code	Android app runs in continuous scanning mode which captures and decodes the QR code successfully.

3.	Final destination reached	QR code is read at every junction. Destination is usually somewhere between two junctions on a side of the path. The bot reaches one of the destination junctions and moves distance X ahead of the junction, stops & starts beeping to indicate destination is reached.
4.	Overcoming collisions	If an object is detected in the path, bot tries to avoid collision by moving left or right to it if the path exists, otherwise a buzzer beeps to indicate collision can't be avoided and bot stays still.
5.	The bot takes smooth turns and does not collides with side walls (stays on the path)	The arena has side walls along the turns. We have 4 sharp sensors on the bot. When the bot is about to collide to one of the sidewalls (detected by crossing a threshold), the bot is made to turn in away from that wall by a small angle (8 degrees). This ensures bot moves on the route avoiding collisions with side walls.
6.	Bot takes next turn on the junction successfully	App communicates with the bot using a bluetooth module. App decodes QR code & calculates the next turn to be taken on the junction. This is communicated to the bot and thus the bot takes the next turn successfully on the junction.

6. Discussion of System

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

Almost everything worked according to plan with some manipulation.

1. Detection of QR code was decided initially by USB camera which was done thereafter by phone camera.
2. QR code decoding was done successfully by phone aaplication.
3. Detection of side wall distance was done successfully by side sensors as decided initially.
4. Obstacle was detected with help of front sharp sensor successfully.

b) Changes made in plan from SRS:

Many small & big changes were made in plan which was in SRS.Some of them were

1. Initially bot was decided to move in greenhouse which was shifted to arena due to sensors capability of recognizing distance and side walls constraints.
2. Phone camera was used instead of USB camera in QR code detection.This was because there was no port for USB camera in firebird V.
3. Shift from Xbee to bluetooth module.Since we were using phone camera instead of USB camera, so serial communication through Xbee was not possible through phone.So we have to use bluetooth module.
4. Phone Application was used instead of laptop for processing of information and sending instruction to bot to reach destination.This change was due to shift from using Xbee to bluetooth module.
5. Sharp sensors were used instead of IR proximity sensors for detecting side wall distance as IR proximity sensors were not working.
<Enumerate changes and include reasoning to why there was a change>

7. Future Work

<Mention about re-usable components and list out possible extensions to your work>
Most of code in our development is reusable if somebody is performing similar functions.

1. Application is used for connecting to bot through bluetooth module which can be used by anyone who wants to perform similar function.
2. If we leave bot in arena it start moving avoiding any collision.If there is only one possible direction for bot to follow, it will follow same direction.Thus same code can be used in case someone is working with some functionality of bot in arena.
3. QR decoding in which we are using Zxing barcode encoder can be reused.

Many assumptions were assumed in development of project.Future work will be to work without these assumptions.

1. We are assuming side walls in arena.Bot is using these side walls for moving and their absence is recognition of junction.We may work without these side walls in future which sounds too practical in nature.
2. We are using QR code as junction signal which is just a abstraction of message which we are using for storing information of junctions.We can use images & text for representing the message and can work on the time needed to decode such messages.
3. We have not handled searching of alternate path on arrival of obstacle on path.In future we can work on searching of alternate path when any obstacle arrives.
4. Currently there was limitation of sensors range due to which distance between walls of arena were kept in particular range.In future we can use sensors with very good range values so that distance can be detected even from a greater distance from bot.
5. In Current scenario motion of bot was adjusted to keep camera in range of QR code.In future we can

work on camera range so that bot moves with normal speed and can detect and decode message.

8. Conclusions

Our project consists of two interacting modules: bot and mobile. Mobile captures a continuous video stream, searches for a QR-code in it, deocdes it and sends it to bot via bluetooth. When bot receives communication from mobile, an interrupt is raised and the received data is stored in appropriate variables. In the main control loop execution of the bot, it constantly moves forward avoiding collisions with the sidewalls. When it reaches an intersection, it uses the information which was received from mobile to make appropriate direction changes.

Our project works as a base module for autonomous navigation. It abstracts away the underlying structure. If any team wants to make any project later, which needs autonomous navigation, it can be built on top of our code.

9. References

1) Monocular Vision for Mobile Robot Localization and Autonomous Navigation http://link.springer.com/article/10.1007/s11263-006-0023-y#page-1
2) Open-source, multi-format 1D/2D barcode image processing library implemented in Java, with ports to other languages (https://github.com/zxing/zxing).
3) http://www.i-programmer.info/programming/android/5887-android-adventures-getting-started-with-android-studio.html
4) http://developer.android.com/sdk/installing/studio.html (download android studio)