

Course Project Documentation
CS308 Project

DRAVOID
Android Touch Controlled Bot

Team 7

Gaurav Choudhary (09005026)

Ashish Tajane (09005027)

Shikhar Paliwal (09005029)

Rishabh Singhal (09005005)

Table of Contents

1. Introduction.....	3
2. Problem Statement.....	4
3. Requirements.....	5
4. Implementation.....	6
5. Testing Strategy and Data.....	7
6. Discussion of System.....	13
7. Future Work.....	14
8. Conclusion.....	15
9. References.....	16

1. Introduction

There have been projects / systems which use computer for controlling the robot or have a separate remote for giving instructions for movements if the robot lack the necessary intelligence for doing those activities on its own. Also, we observed that these days, everyone uses a smartphone which uses operating systems like Android, iOS etc. Android being a new, challenging and very interesting platform from development point of view, we thought doing something involving these two areas.

So, we came up with an android application which controls the basic movements of robot using bluetooth for communicating with the robot and we also added another feature where the robot can draw the shape on paper / any surface originally drawn by user on the android phone screen. And this has become the major part of the project.

2. Problem Statement

- To implement user and bot interaction using android application via bluetooth connection.
- To implement a feature where user draws any shape on phone using touch screen feature - after this , the figure is processed and the bot is then auto controlled by the phone to have it draw the given shape on a white sheet.
- To implement a feature and provide interface for the same where the user can control the bot movements and the movement of pen attached to it (like pen up, pen down) using his phone.

3. Requirements

A. Hardware Requirements

1. Firebird - a bot for drawing the shape on surface
2. Phone with Android - for running the android application
3. Marker - attached to firebird using servo motor for drawing purposes
4. Servo Motor - attached to firebird, for controlling marker movements
5. Bluetooth - attached to firebird, for communicating with android phone

B. Software Requirements

1. AVR Studio + Bootloader - for burning the code on bot
2. Android OS - for running the app

4. Implementation

A. Functionalities

1. At Android Phone :

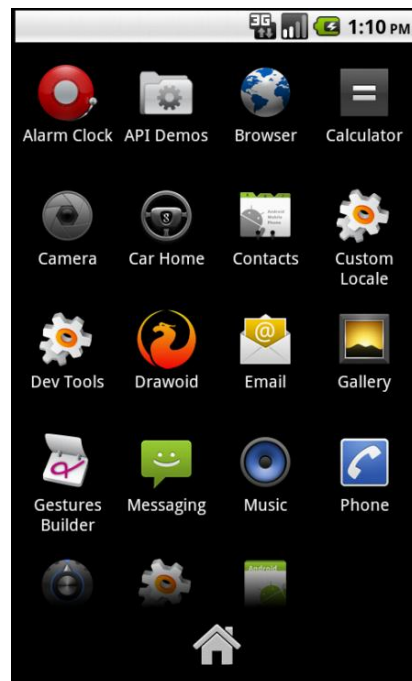
When the user draws some shape on the phone screen, the android app collects a large set of points to be followed sequentially for drawing the shape. Then we have devised an algorithm to reduce these points for optimizing the drawing process. The algorithm removes points which are very close to each other (closer than the threshold distance) and also those points which are almost collinear (angle greater than the threshold angle), this is explained in details in next section. The algorithm then converts the optimized set of points into instructions like R20, U, F13 etc. and puts them into a byte array separating each instruction by a \$ symbol. And then transmit this byte array to the bot using bluetooth communication.

2. At Firebird Bot :

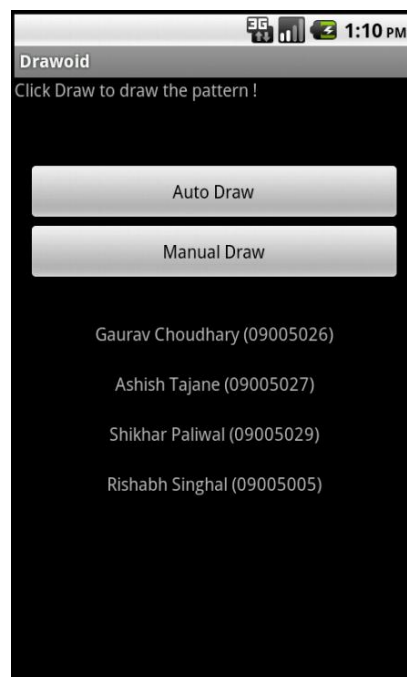
The bot separates each instruction from bot array and follows one instruction at a time like R20 meaning turning right by 20 degree, U meaning pen up etc. all the turning instructions are followed by certain error correcting instructions because the bot doesn't actually turn at the vertical axis at pen but at the centre of the bot.

5. Testing Strategy and Data

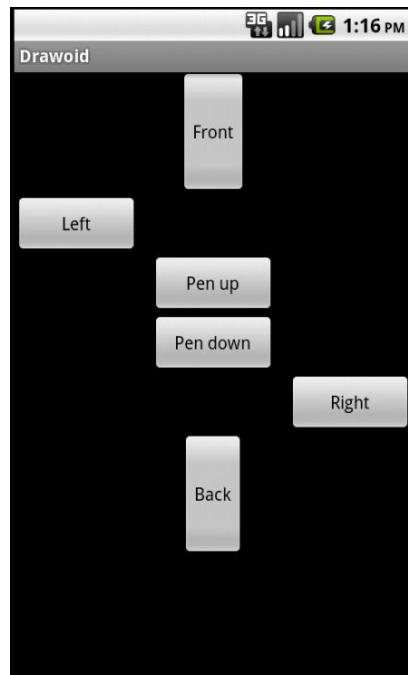
The user starts the android app **Drawoid** from the android phone and selects the mode for drawing. The manual draw window has buttons for functions like pen up and pen down. move front and move back buttons move the bot by unit distance in respective direction and right and left buttons turn the bot by 10 degree to the respective direction.



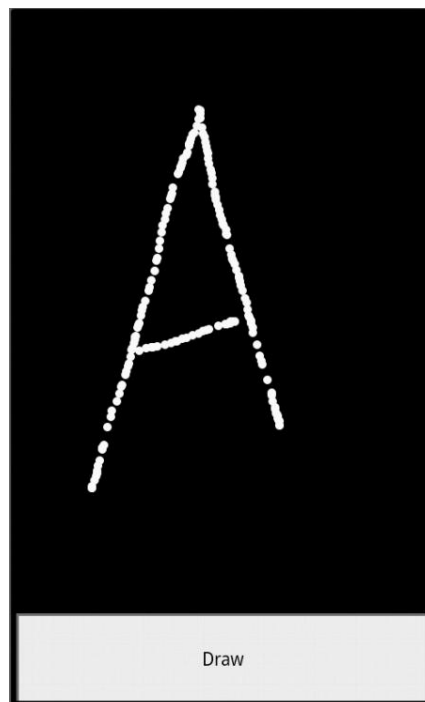
The user starts the application



The user chooses the drawing method

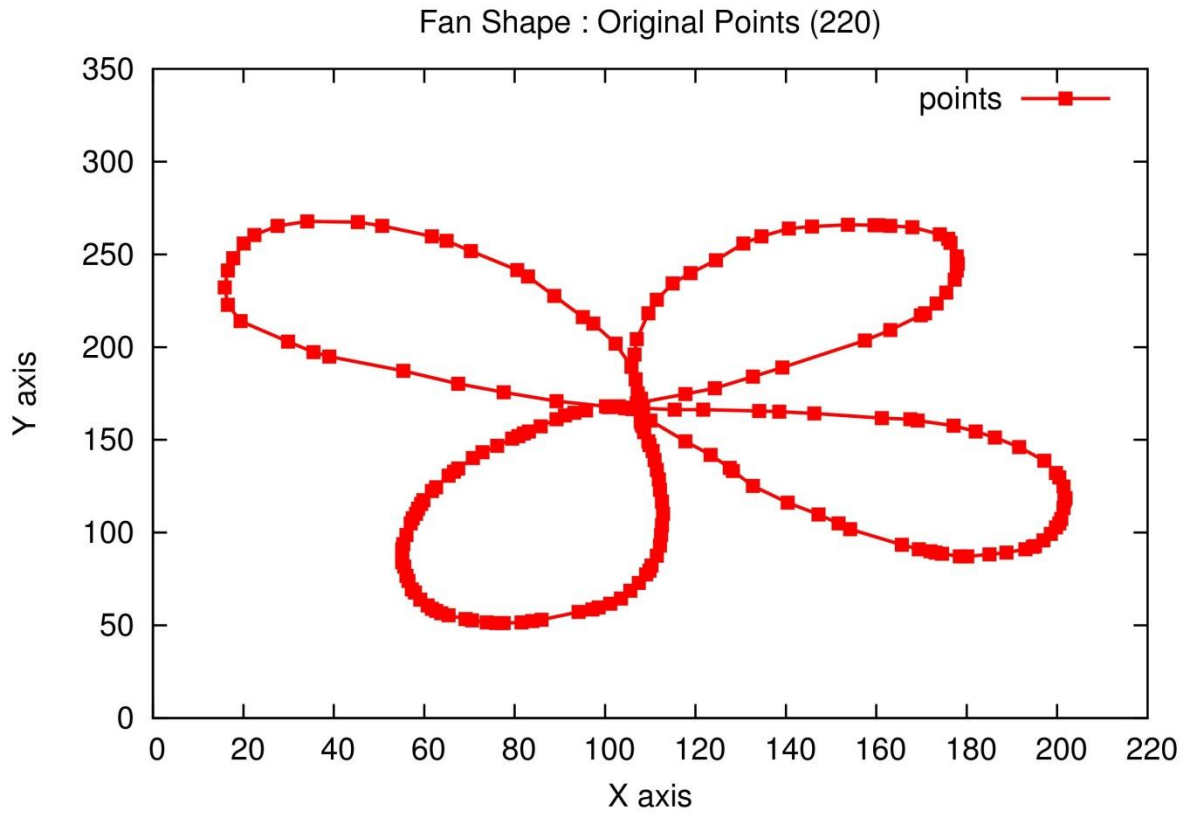


The interface for manual drawing

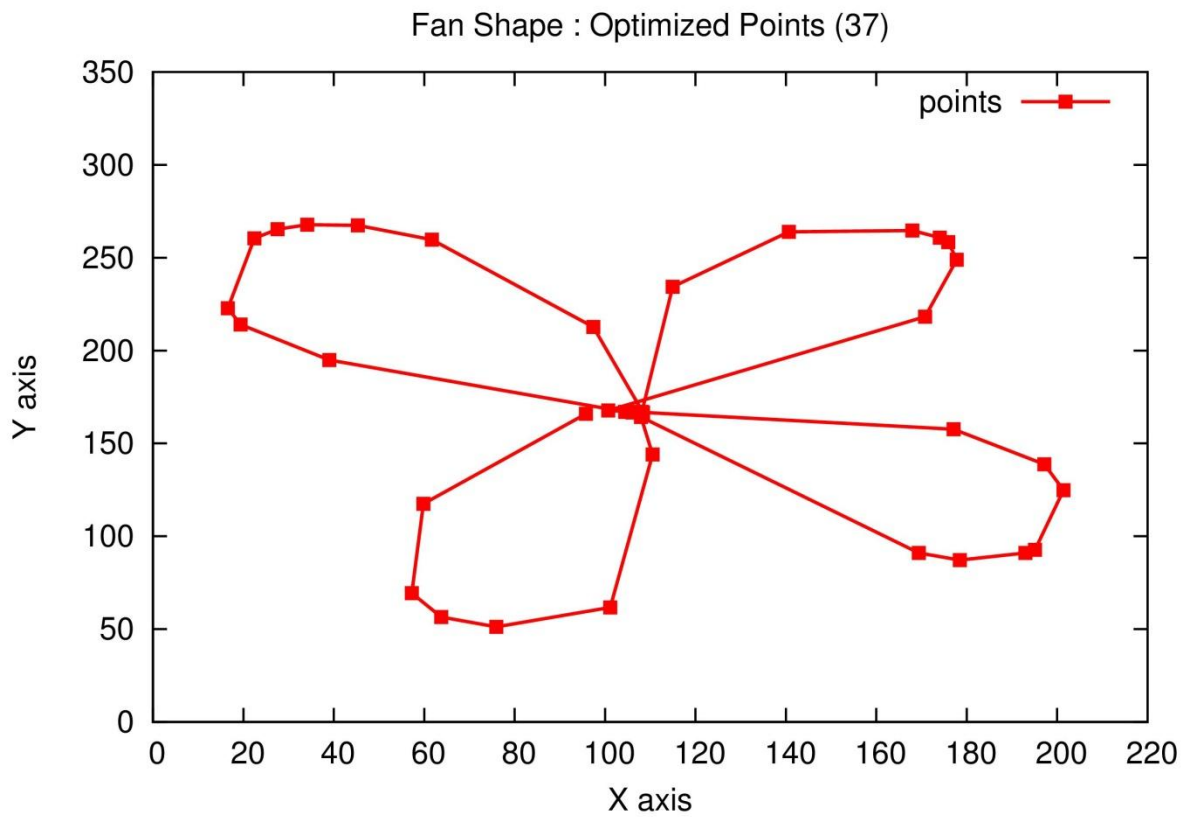


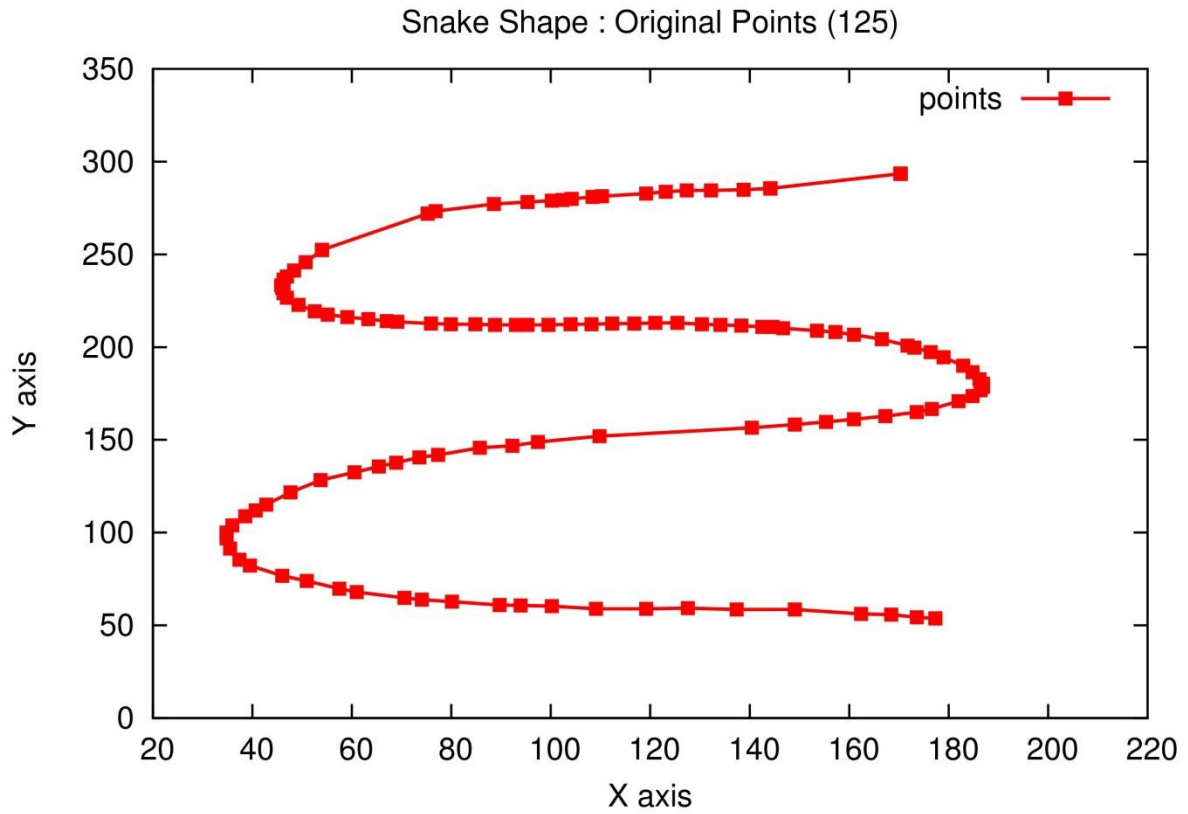
The interface for auto draw method

In the auto draw method, the user draws a shape ('A' in this case) on the phone and the application collects the points and gives it to the algorithm which then reduces the points for optimizing the robot movements by minimizing robot turns keeping the shape almost same. following images show the performance of the algorithm and the effect of reducing the points.

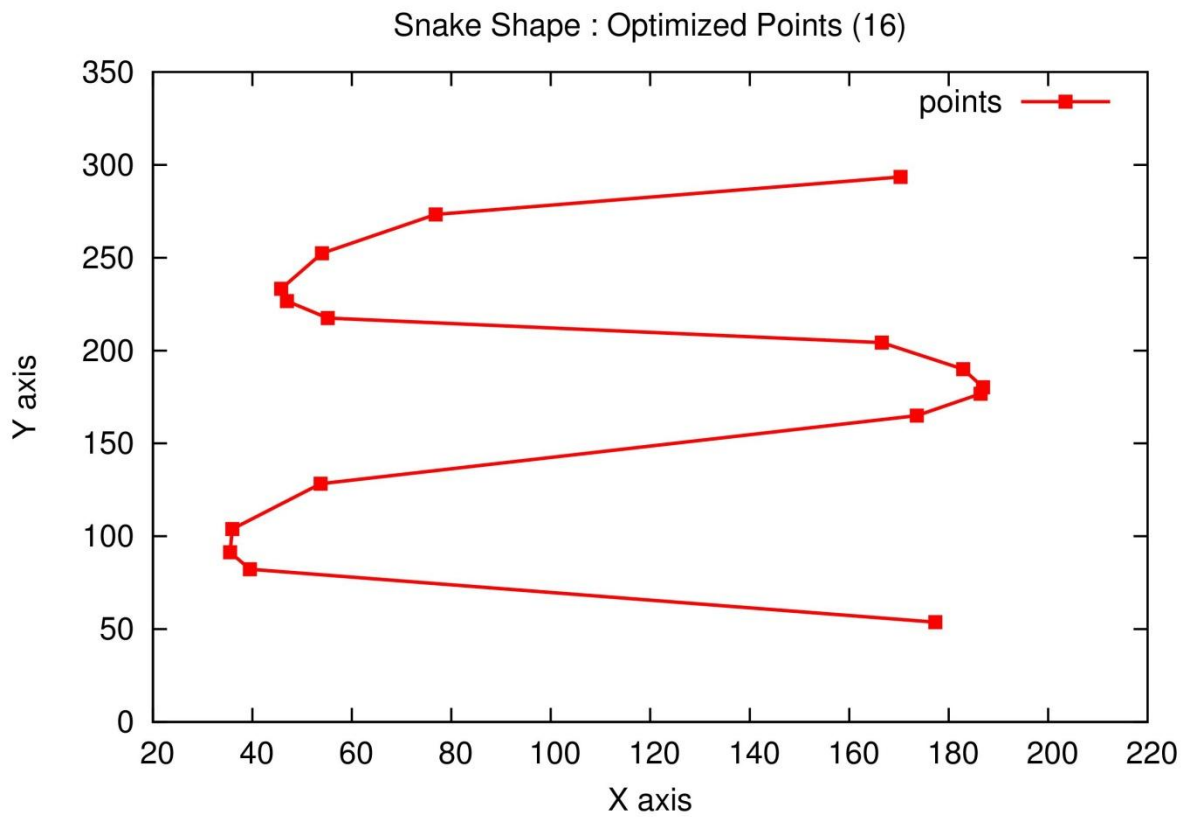


A flower petals shape drawn from the points originally collected by the android application above and from the optimized points below. 220 points are reduced to 37.

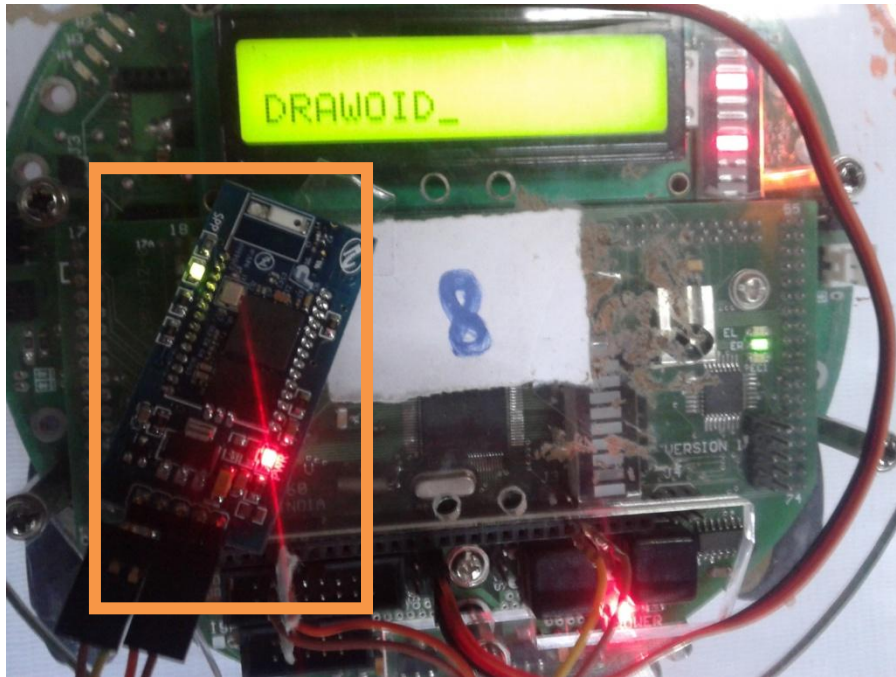




A zig-zag snakelike shape drawn from the points originally collected by the android application above and from the optimized points below. 125 points are reduced to 16.

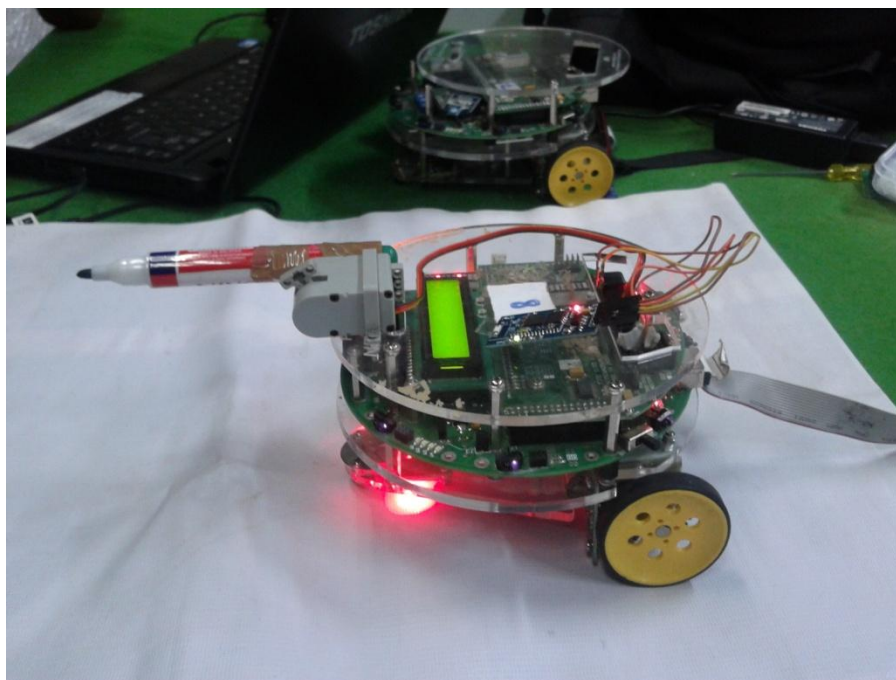


The bot gets the byte array of instructions generated from optimized points over bluetooth connection.



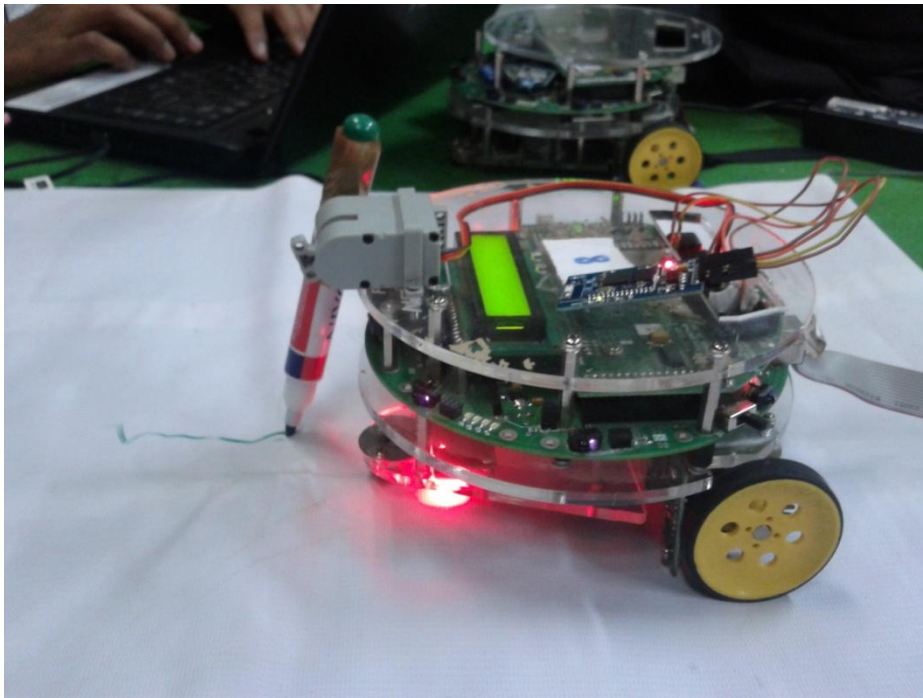
Firebird with BlueLINK bluetooth module highlighted

Initially the bot is in pen up position i.e. the pen is not touching the ground and it can move without having to draw anything on surface.

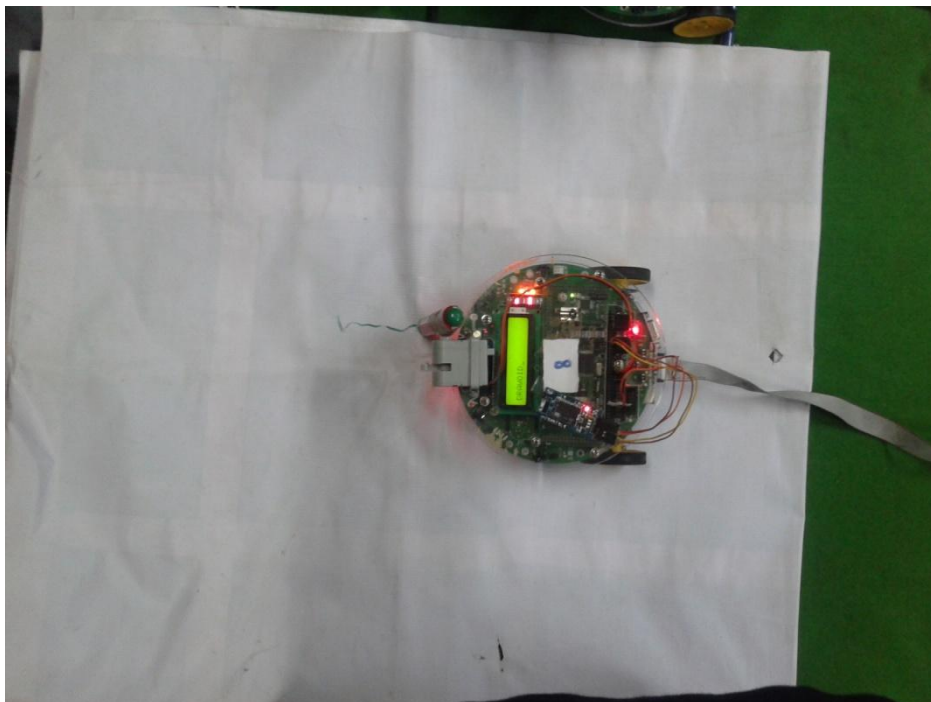


The bot initially in the Pen Up position

The bot starts drawing the shape as soon as it gets



The bot at Pen Down position



The bot drawing a line (top view)

6. Discussion of System

A. What worked as per plan :

1. Firebird

Bot is receiving command from android application through Bluetooth module and implements it by appropriately moving by specified distance.

2. Android App

The user interface is an android application which provides a user to draw a figure on the screen and later transmit a series of command to be executed by firebird.

3. Dot Removal Algorithm

The algorithm is very efficient with very less response time. It removes any unnecessary dots between any 2 dot forming an angle of more than or equal to 160. Also a dot which is very closer to other is removed.

4. Bluetooth Communication

Android is successful in connecting to firebird through Bluetooth. Firebird is also able to raise interrupt and add data to an array received through bluelink (5V Bluetooth module).

B. What we added more than what was discussed in SRS :

1. Pen Movement

Added a servo motor mechanism to move pen up and down according to the requirement

2. Manual Daw

An option added to the user interface where user can manually control bot through forward, back, right, left, pen up & pen down buttons.

C. Changes made in plan :

1. Earlier we decided to compute the distance and turns on firebird but later we thought it can be done easily and quickly on android bot from where we can send signal which can directly be executed.
2. Earlier we decided to compute the distance and turns on firebird but later we thought it can be done easily and quickly on android bot from where we can send signal which can directly be executed.

7. Future Work

- **Map Maker -**

By forming better algorithms for reducing the error or removing it completely, the bot can be used to draw very large maps / graffiti / Rangoli / drawing. A camera can also be attached at the top (ceiling) which gives periodic or continuous feedback of the bot's position and the part drawn so far.

- **Maze Follower -**

Enhancements can be done in the direction of controlling the bot manually using android and the bot can be made to follow a known maze. Speed control functionalities can be added.

- **Spy Bot -**

With a camera attached to it, firebird can be moved in an unknown environment based on the feedback from camera.

8. Conclusion

This application can be generalised to work on any smartphone and the android application can be easily replicated on other similar platforms like iOS, BlackBerry OS, Windows Phone etc. After mounting a camera on the bot, it can be made to send its video feedback back on the android (or similar) device and such a system can be used as a spy bot by military or intelligence agencies. The drawing feature can be further developed to reduce the human labour and efforts in making large rangolis, maps and graffiti.

9. References

1. Eclipse Installation Guide -
<http://www.cs.umd.edu/eclipse/EclipseTutorial/install.html>
2. Android SDK - <http://developer.android.com/sdk/index.html>
3. Android Eclipse Plugin (ADT) - <http://developer.android.com/sdk/eclipse-adt.html#installing>
4. Android Development Guide - <http://developer.android.com/guide/index.html>
5. BlueLINK user manual -
http://www.rhydolabz.com/documents/wireless/BlueLINK_User_Manual.pdf
6. Servo motor and bluetooth connectivity reference - <http://www.e-yantra.org>