

DRAVOID

ANDROID TOUCH BOT CONTROL

GROUP NO. 7

Gaurav Choudhary

Ashish Tajane

Shikhar Paliwal

Rishabh Singhal

Contents



- ❑ Problem Statement
- ❑ Requirements/Task Specifications
- ❑ Project Plan
- ❑ Workflow block diagram
- ❑ Innovation and challenges
- ❑ Task Completed
- ❑ Review, Test Plan/Cases
- ❑ Re-usability features
- ❑ Future Enhancements

Problem Statement

- ❑ To implement user and bot interaction using Android Application via Bluetooth connection.
- ❑ User draws any shape on the phone using the touch screen feature.
- ❑ After this the figure is processed and the bot (attached with a marker) is then auto controlled by the phone to have it draw the given shape on a white sheet.

Problem Statement

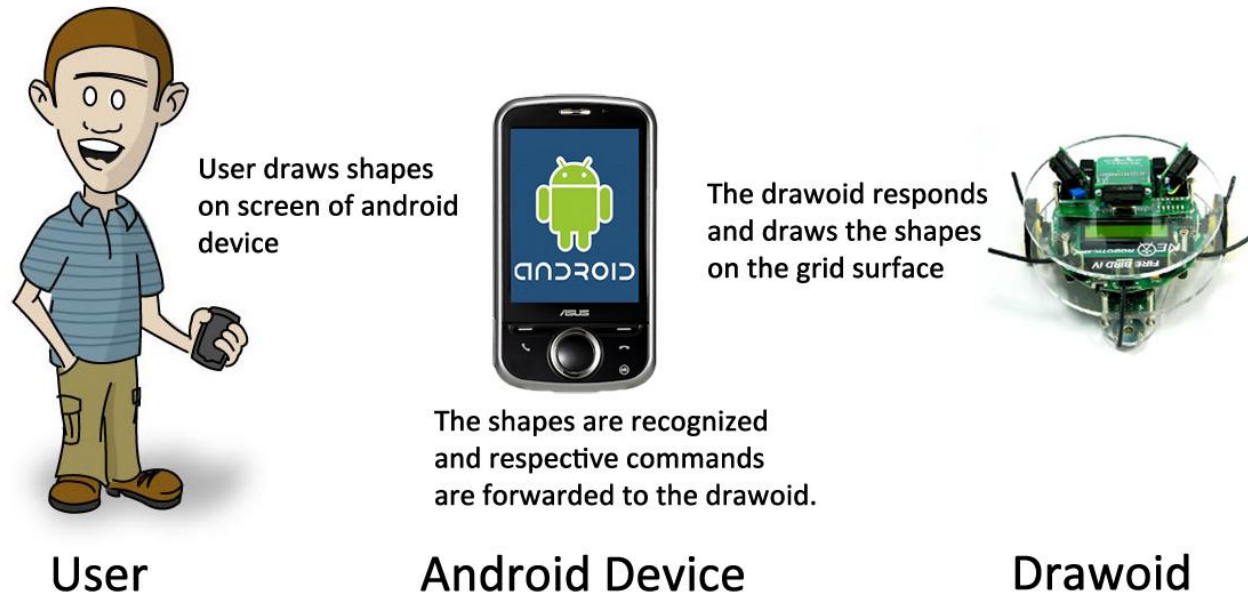


Fig. 2 - Drawing Process

Requirements

- User Interface: Android Application, this is the part of the system that will be used to interact with the user
- Hardware Interfaces:
 - ▣ Android supported phone with Bluetooth connectivity
 - ▣ Firebird
 - ▣ Bluetooth Device for firebird
- Software Interfaces:
 - ▣ Bluetooth connectivity module for android
 - ▣ Bluetooth connectivity module for firebird

Task Specifications

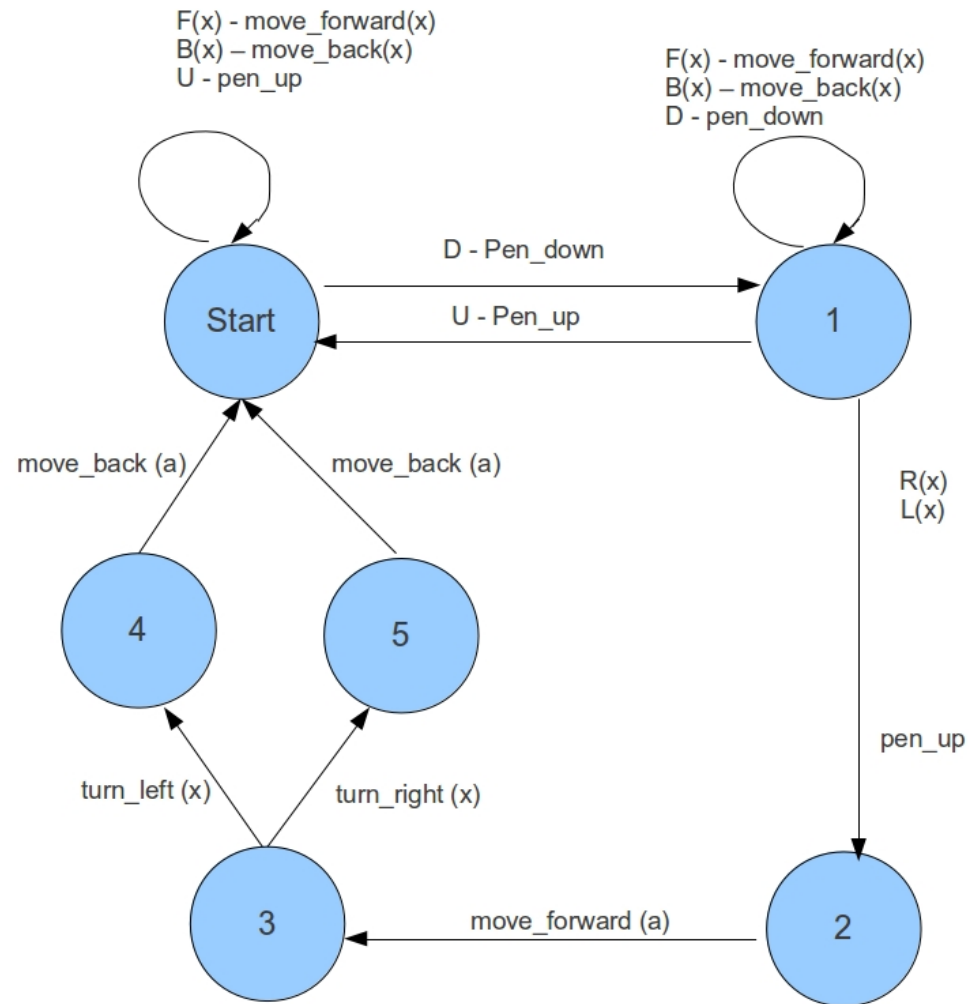
- User Interface Development: Android Application
- Functionality: At the bot level what are the functions implemented for this project
- Algorithm: How the shape drawn by user gets converted into a relevant sequence of instructions
- Arena: What is the platform on which the bot draws the shape

Project Plan

- Plan was to develop modules separately on Android Phone and Firebird and test them locally
- Then decide a protocol and connect these modules to interact with each other
- This was the plan followed during the development

Work Division	Date of completion
User Interface Development	7 th April
Firebird functionality	8 th April
Protocol Development and Connection	9 th April
Algorithm Development	10 th April
Testing	10 th April

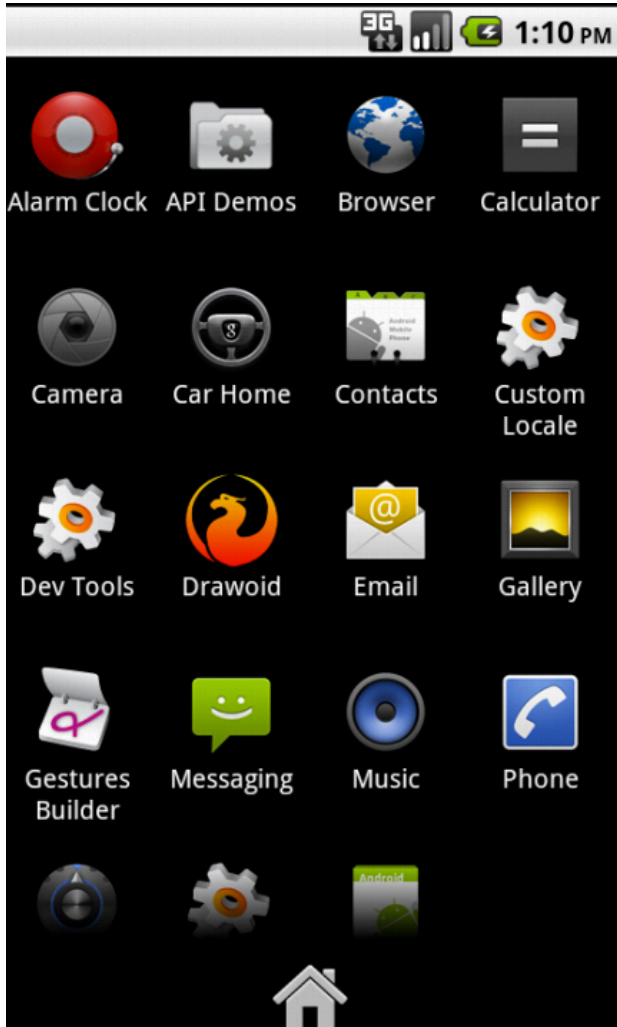
FSM/Block Diagram



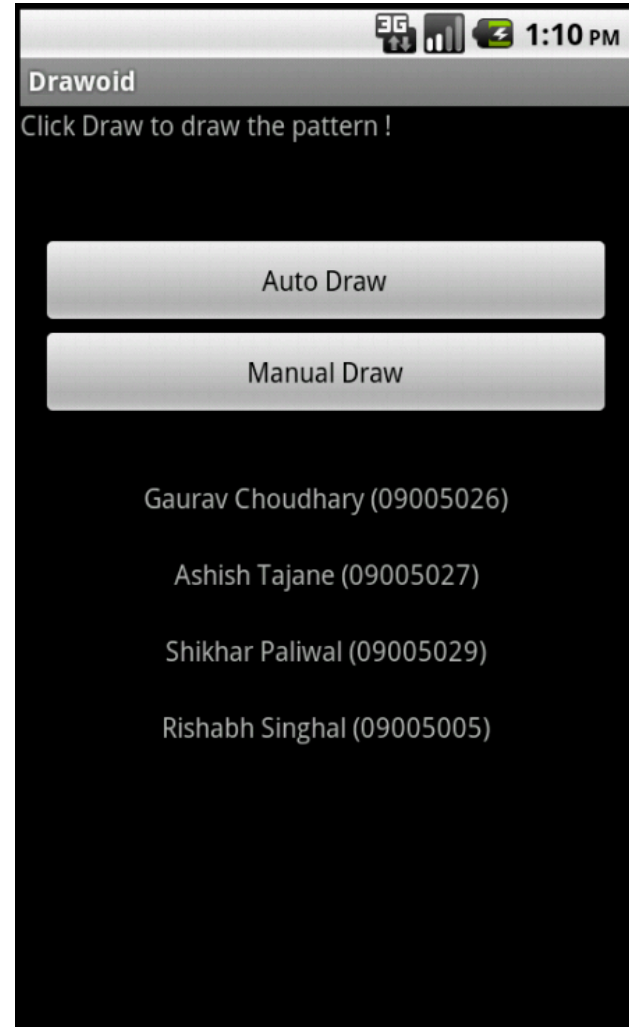
Innovation & Challenges

Challenge	Innovation
The pen is not at the center of bot so when the bot turns, the point of contact of pen with surface changes.	The bot moves and adjusts itself such that the point of contact of pen remains same without changing the direction of bot.
The bot draws while actually drawing as well as adjusting itself.	Servo motor added for implementing pen up and pen down so that pen is down only when the bot needs to draw.
The android app gives too many points of the shape drawn and it becomes tedious and time consuming for bot to draw it.	An algorithm reduces those points maintaining the approximate shape same and reducing the time taken to draw the shape by bot.

Task Completed: User Interface

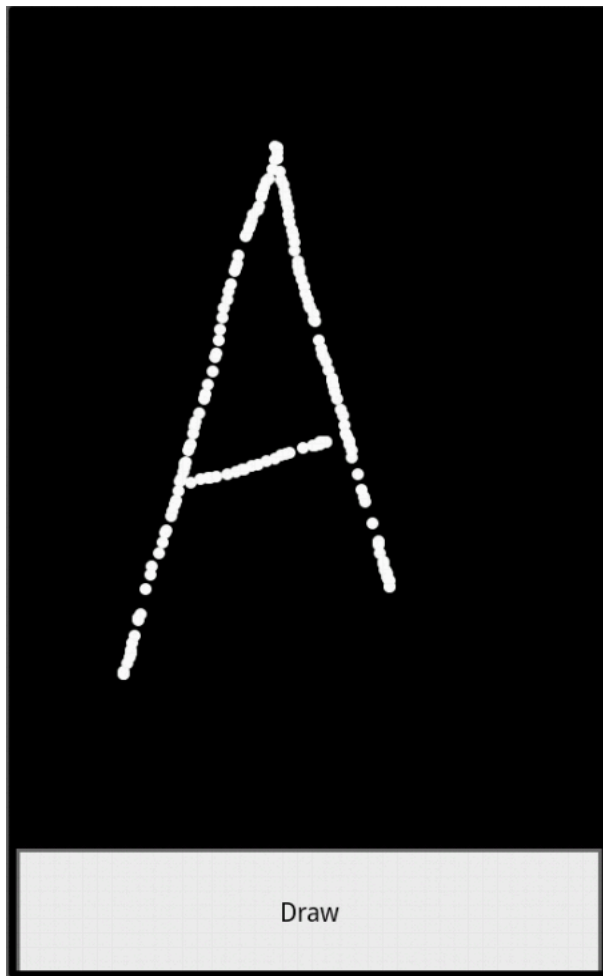


App Icon

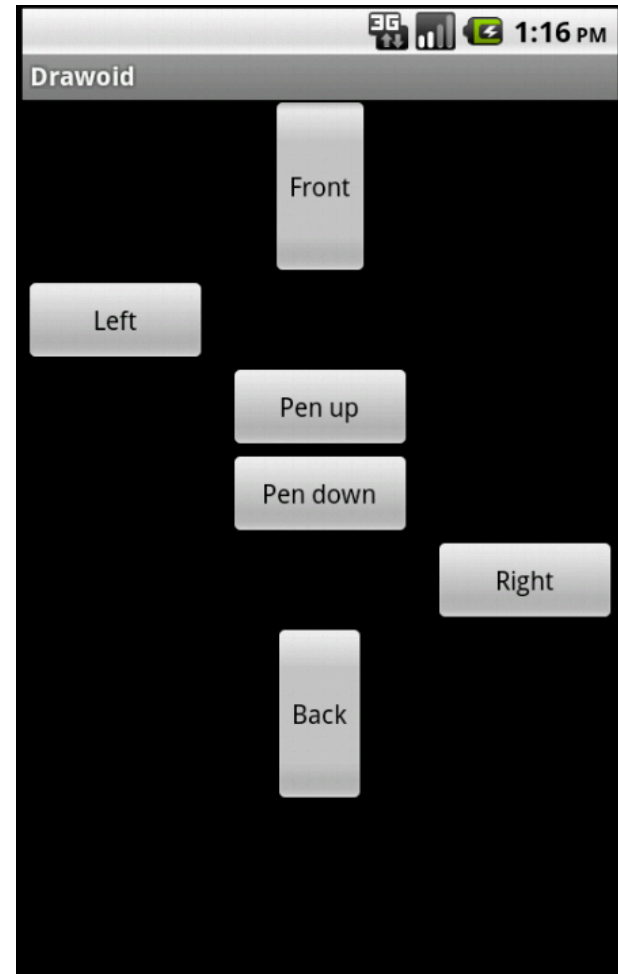


Welcome Interface

Task Completed: User Interface



Auto Draw



Manual Draw

Task Completed: Algorithm

- Compare distance between every consecutive pair and delete the second point if the distance between points is less than the threshold distance (0.05 mm) and continue with first point again so the cleanup process completes in a single pass.
- Then compare angle between every 3 consecutive points and delete the second (middle) point if that angle is greater than the threshold angle (160 degree).

Task Completed: Algorithm

- Then the algorithm generates the instructions like R20, U, D, F30 meaning turn right by 20 degree, Pen up, pen down and move forward by 30 mm respectively.
- The angle is calculated using the cosine rule and the direction is calculates using the vector product.

Task Completed: Protocol

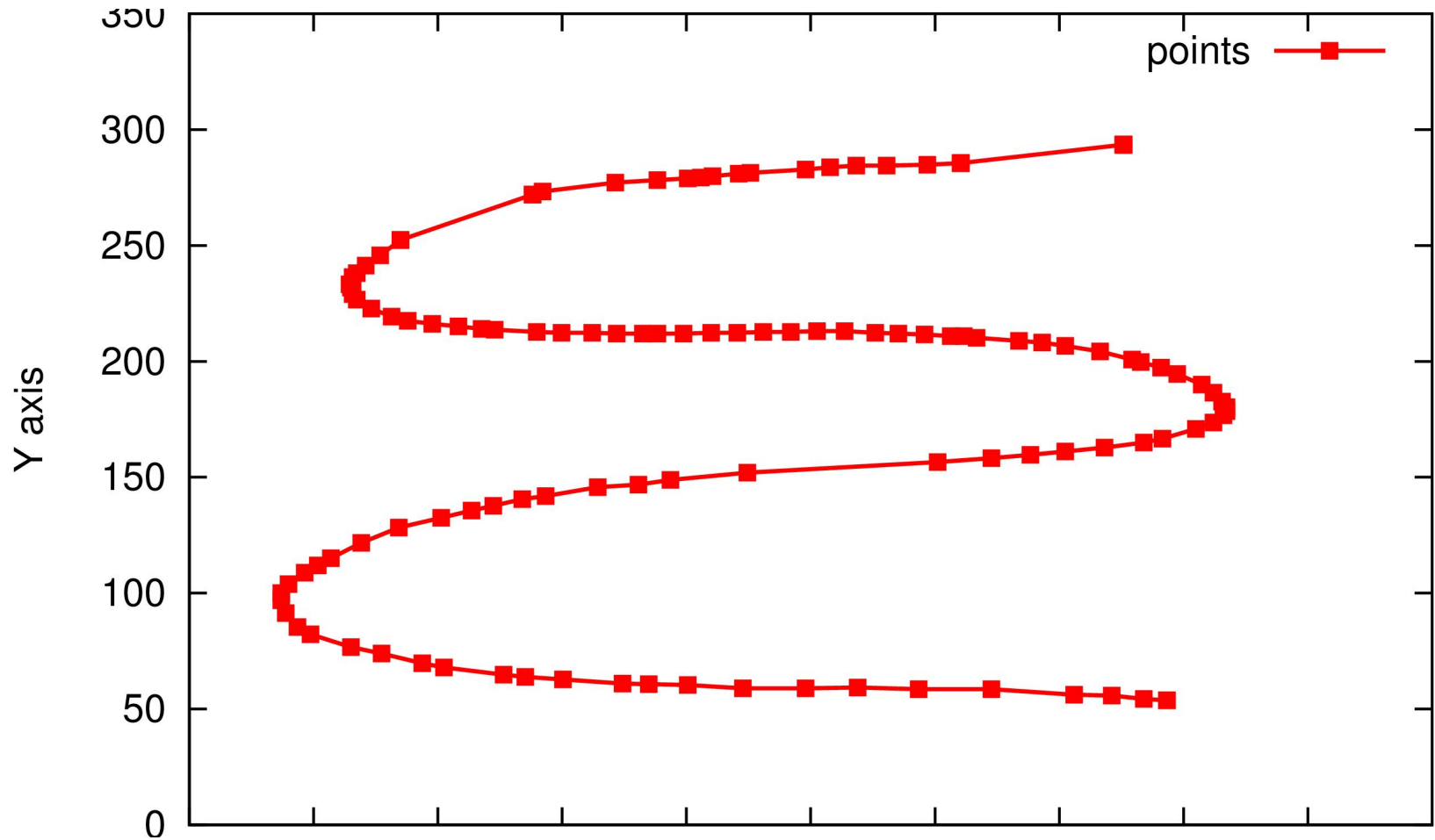
- The instructions are sent in a byte array over the Bluetooth to the bot. we insert a \$ symbol after every instruction denoting the end of instruction
- So the bot collects the data from byte array until it gets a \$ and then it form the instruction from data read so far from array and executes it and starts collecting further data. this cycle continues till the byte array is empty.

FireBird Functionality

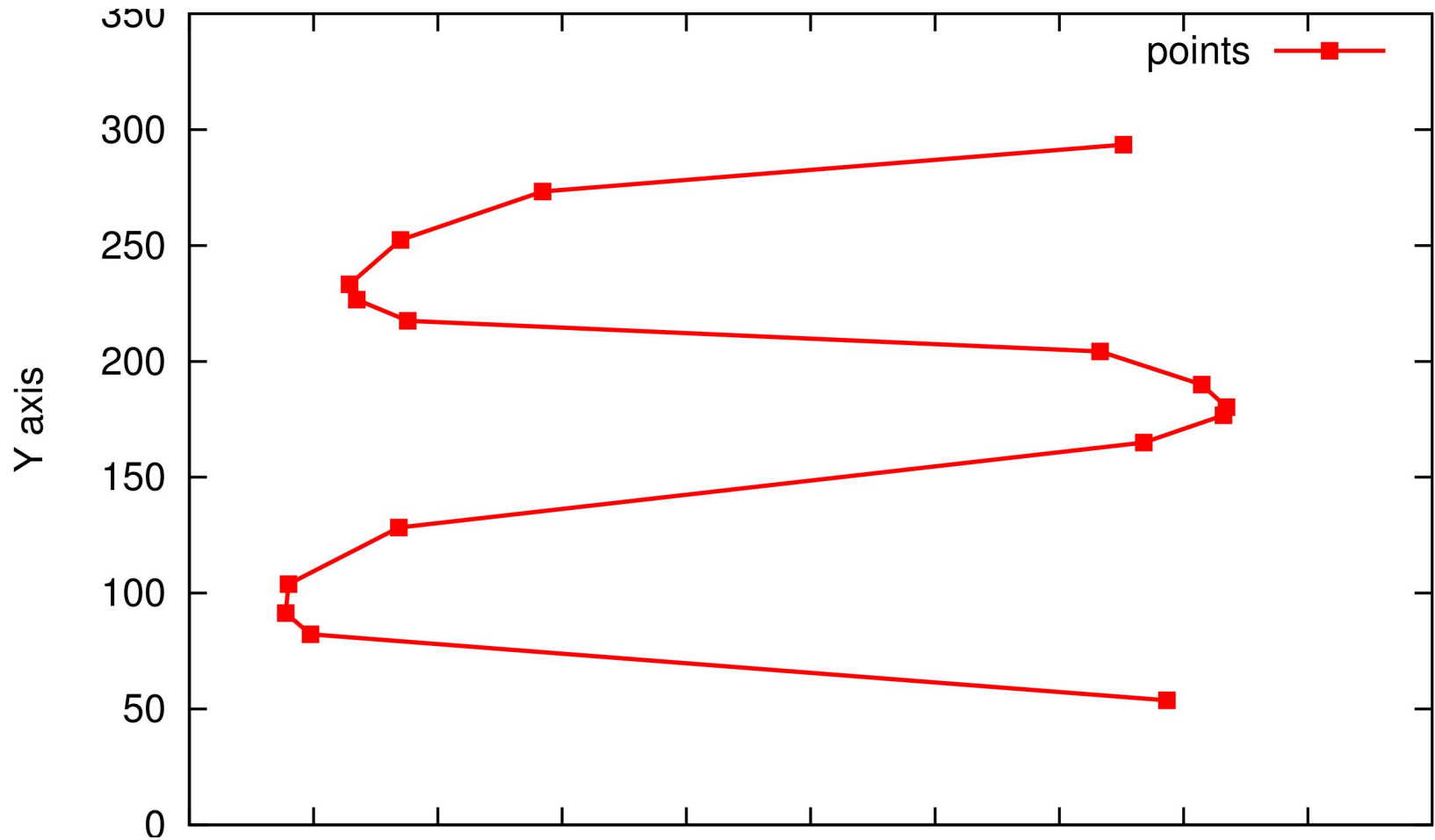
According to protocol, firebird receive data from Bluetooth device in a proper format and interpret it as following -

- F x – move in forward direction by x mm
 - ▣ forward (x)
- B x – move in backward direction by x mm
 - ▣ backward (x)
- R x – turn right by x degrees
 - ▣ Forward (a), turn_right(x), backward(a)
- L x – turn left by x degrees
 - ▣ Forward (a), turn_left(x), backward(a)

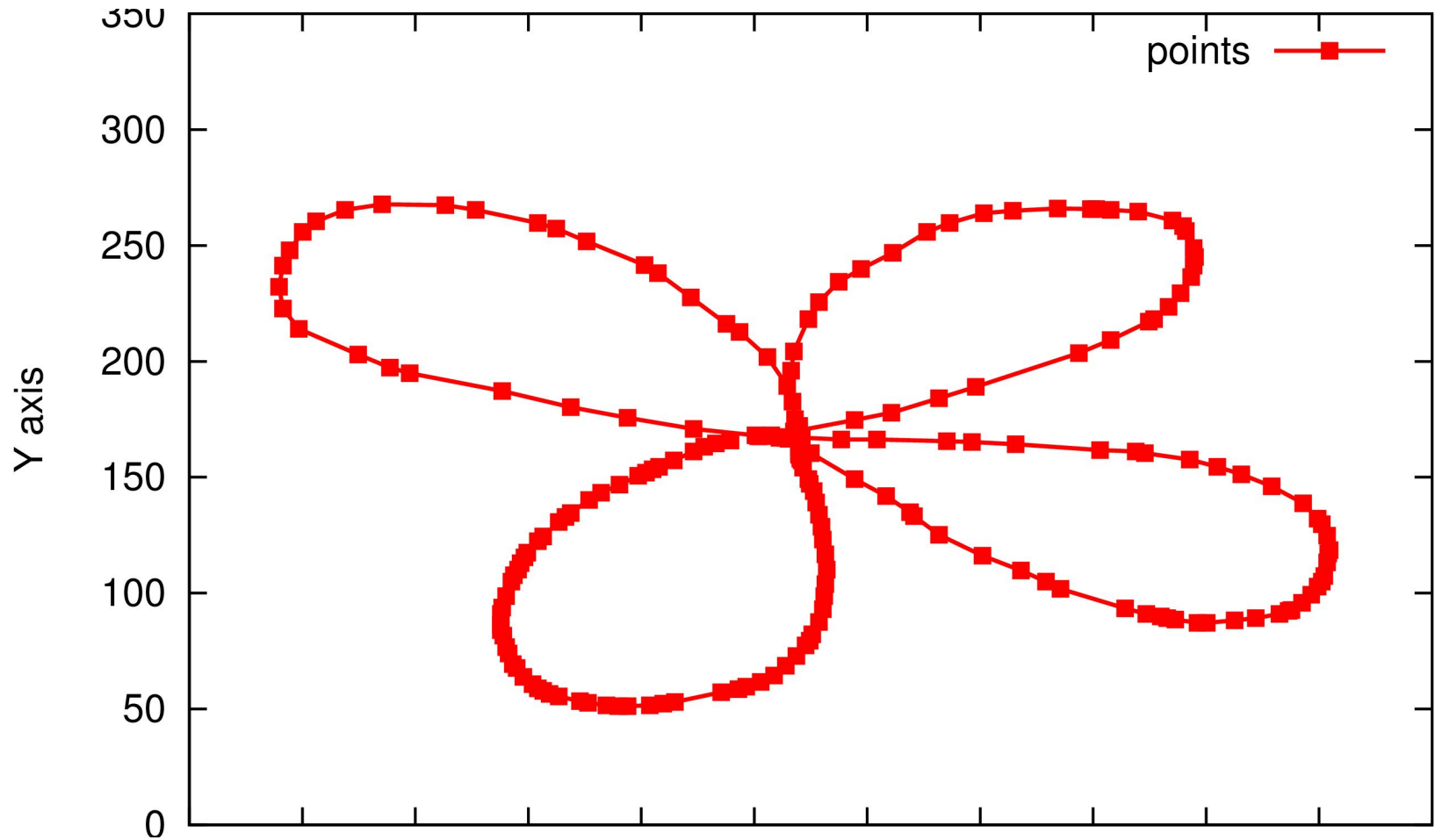
Test Cases

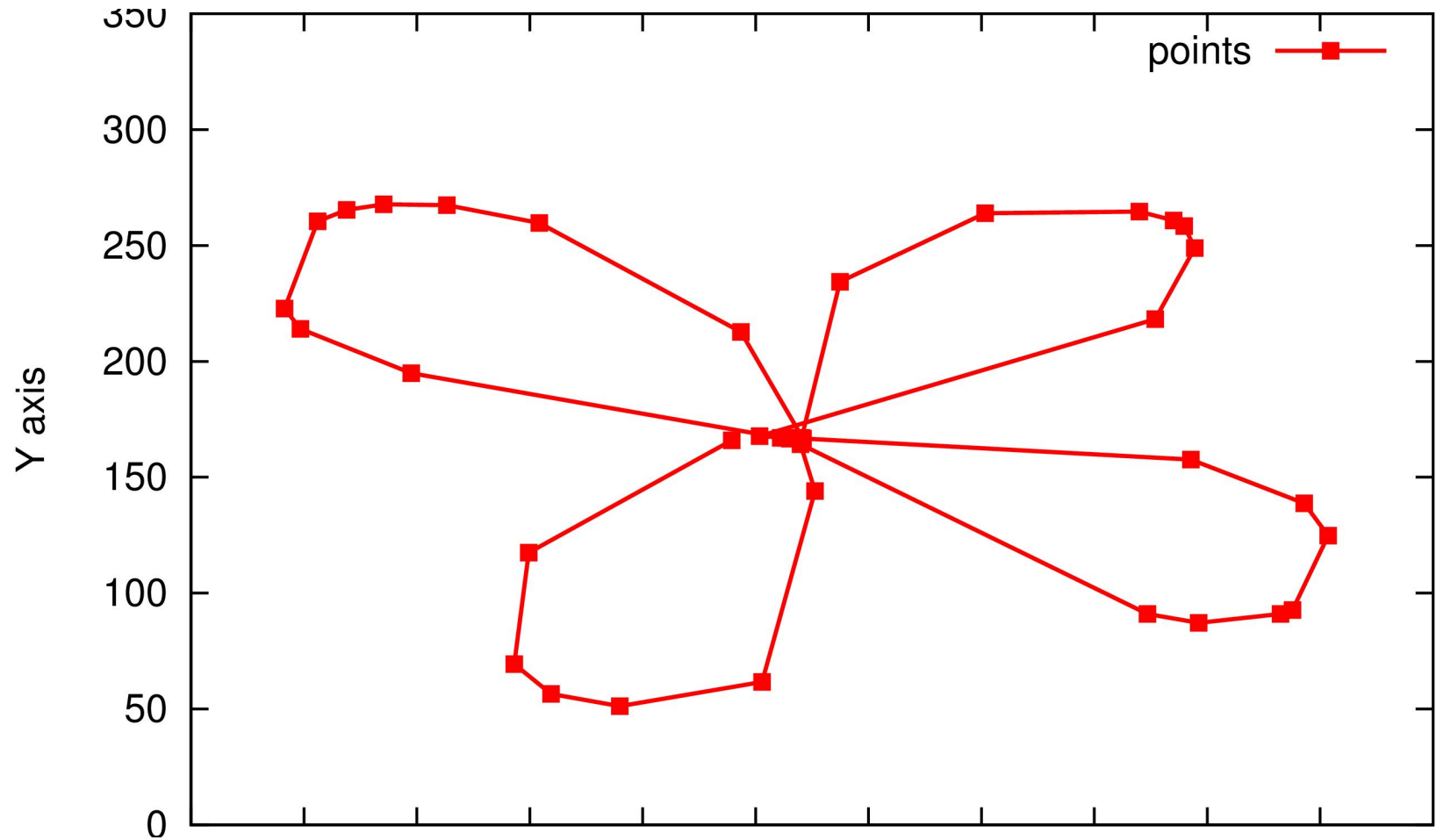


Test Cases



Test Cases





Re-Usability Features

- Bluetooth module
 - ▣ Receive data from other Bluetooth device
 - ▣ Implemented for 5V Serial TTL
- Firebird movement
 - ▣ Move forward & backward (in mm)
 - ▣ Turn right & left (in degrees)
- Servo Motor Control
 - ▣ Initialize ports and run servo motor by in both directions to achieve certain angle

Future Enhancements

- Map Maker
 - ▣ With better algorithms it can be used to draw large maps/drawing
- Maze follower
 - ▣ Control firebird and follow a known maze
- Spy Bot
 - ▣ With a camera attached on it, firebird can be moved through unknown paths based on the feedback from camera