

Android Interface Firebird API

CS308 Project

Team 2

Pararth Shah 09005009

Aditya Ayyar 09005001

Darshan Kapashi 09005004

Siddhesh Chaubal 09005008

Problem Statement

- To create an **Android library app** which acts as an API for accessing the features of Firebird. The library would be available to Android developers for importing in their Android apps
- To write a **generic command interpreter** which runs on the Firebird, and waits for command messages over the communication interface, performing various actions as required
- To create a **Bluetooth communication interface**, and to design and implement a two way communication protocol for reliably and efficiently sending and receiving data between the Android device and the Firebird
- To create **sample applications** on the Android which demonstrate the use of the Android API for Firebird

Task Specifications

- Generic Firebird API For Android
 - Android Library App
- Generic Firebird Command Interpreter
 - Firebird C Code
- Bluetooth Communication Interface
 - Using external Bluetooth hardware and our own two-way communication protocol
- Sample Applications
 - Four sample applications, ranging from basic use of the API (MissionControl) to fairly complex functionality (MapperBot)
- Modularity For Easy Extensibility
 - Modularly designed for convenient reuse and extensibility

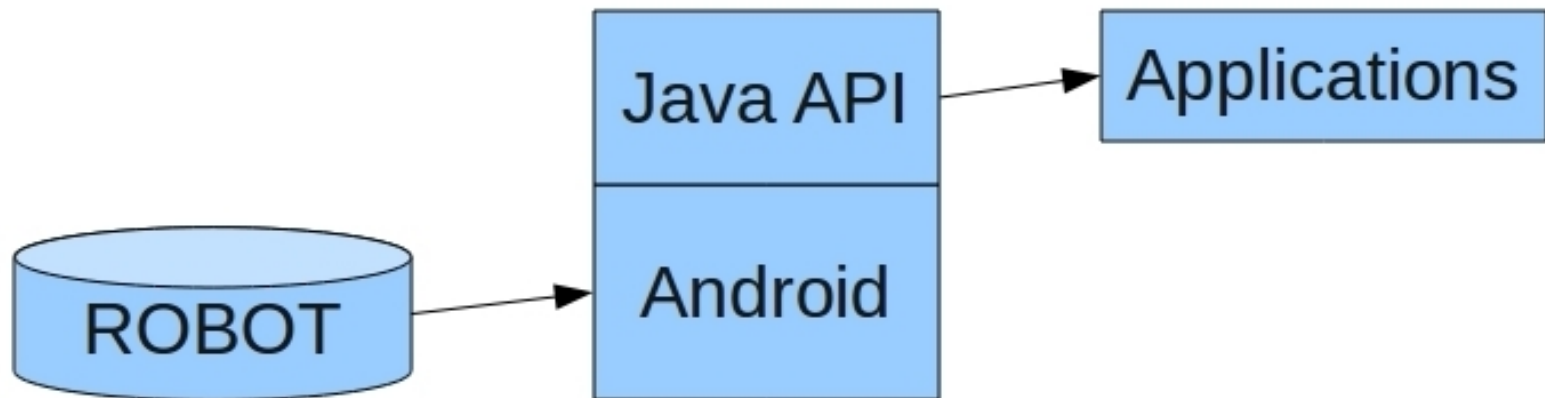
Project Plan

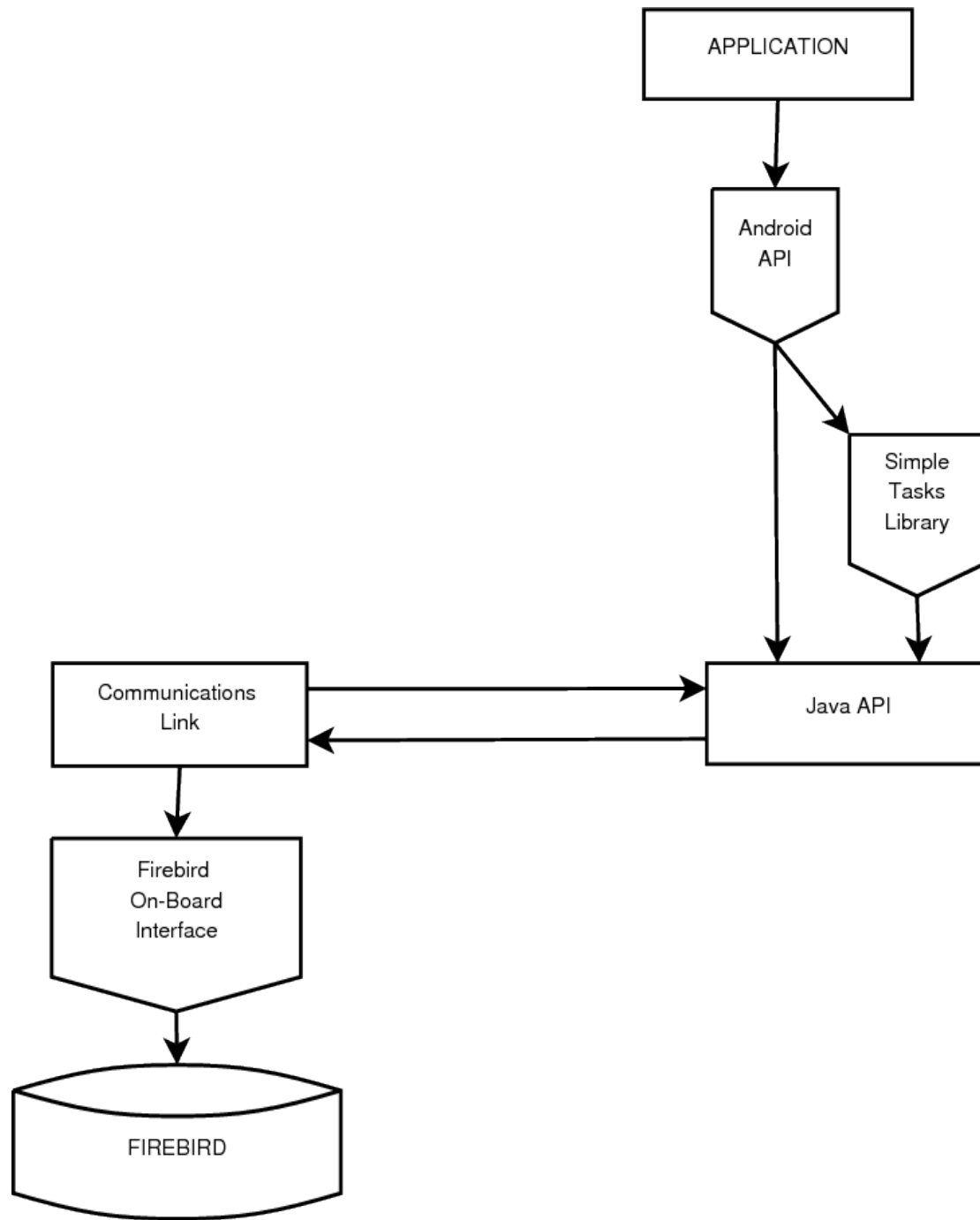
- Stage 1: Firebird C Code (1.5 Weeks)
 - Build Generic Firebird Command Interpreter
 - Encapsulate all features of the Firebird in one single C project which is burnt once on the bot and used by all Android apps
 - Use some of the code from Firebird Swiss Army Knife project
- Stage 2: Bluetooth Comm Interface (0.5 Weeks)
 - Design tailor-made two-way communication protocol over Bluetooth link from scratch
 - Implement reliable sending of commands and receiving of data messages from Android device to Firebird

Project Plan

- Stage 3: Firebird API For Android (2 Weeks)
 - Write Android library app which exposes Java functions to manipulate Firebird bot for generic use
 - Test API by sending control messages to Firebird through Android API
- Stage 4: Sample Applications (1 Week)
 - Implement and test sample applications illustrating the use of the API
 - Build 2 simple apps for demonstrating use of API, and 2 non-trivial apps for highlighting the power of Android API

Architecture





Innovation And Challenges

- Encapsulated all features of Firebird in single project
- Implemented two-way communication protocol
- Implemented message queues at both ends to deal with latency issues
- Built generic Firebird API for use on any Android device
- Implemented MapperBot application on Android for using the Firebird as a spy bot

Tasks Completed

Task	Problem	Solution
1. Generic Firebird Command Interpreter	Differentiate between control messages and data messages sent from Android (single/multiple bytes)	Implemented communication protocol for interpreting messages received via Bluetooth; on receiving valid command appropriate action is taken
	Perform minimal work in interrupt service routine to minimize losses due to nested interrupts	Implemented two buffer scheme: one buffer saves all received data from ISR, another buffer accessed from main loop on receipt of a complete command
2. Bluetooth Communication Module	Latency in Bluetooth Link causes delay in message delivery, affects precision of control of Firebird by Android	Implemented two-way message queues on both Firebird and Android, allows messages to be sent reliably in burst mode

Tasks Completed

Task	Problem	Solution
3. Firebird API For Android	-	-
4. Sample Application: PeterParker	Delay in BT link causes delay in sending commands for whitenline following	Implemented whitenline following on the bot itself, while application logic for finding parking space is coded on Android
5. Sample Application: MapperBot	When the bot loses sight of the left wall, ambiguity in deciding when the bot should move forward and when it should go left in search of a wall	Implemented two states of the bot – Wall Finding and Wall Following, which are maintained in the Android App, and bot is given commands according to its present state and sensor values

Review/Test

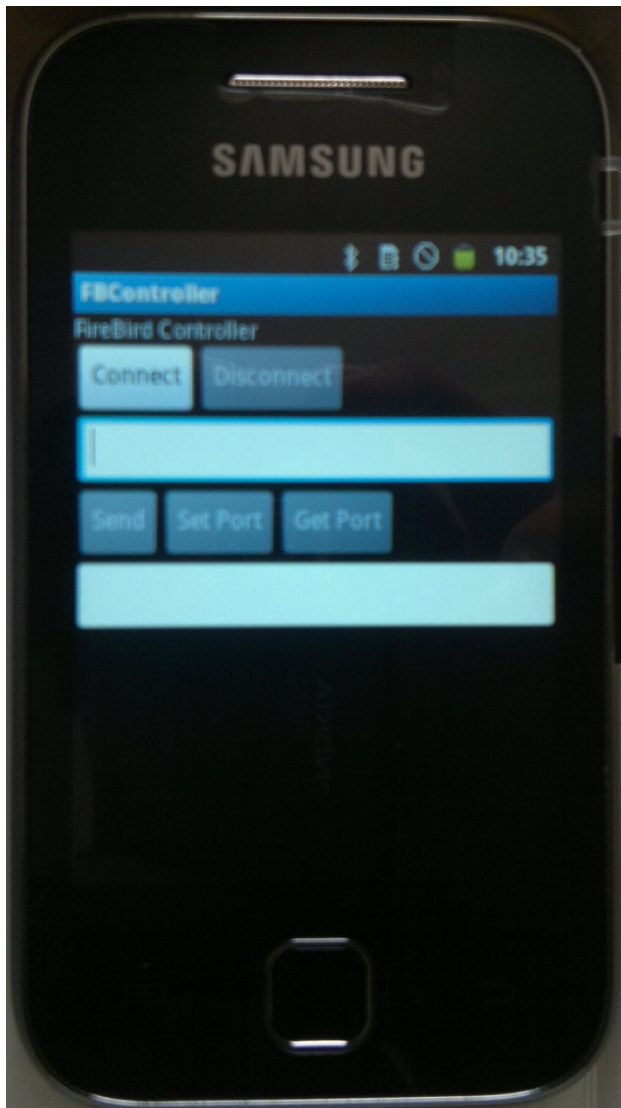
- The Firebird API was tested for effectiveness by implementing sample apps for controlling the Firebird
- We installed the apps on the Samsung Galaxy Y Android smartphone provided to us, and conducted tests to obtain performance data
- The metrics of interest are:
 - Communication delay (ms)
 - Throughput (commands/sec)
 - System Bottleneck
- We tested the communication module in two ways:
 - By sending one command message at a time (stop-and-wait)
 - By sending commands in bursts making use of buffers at both ends (burst-mode)
- The burst mode gave a large performance benefit as the propagation delay over the Bluetooth link is much higher than the transmission delay (time-to-wire) per byte sent and received

Performance Metrics

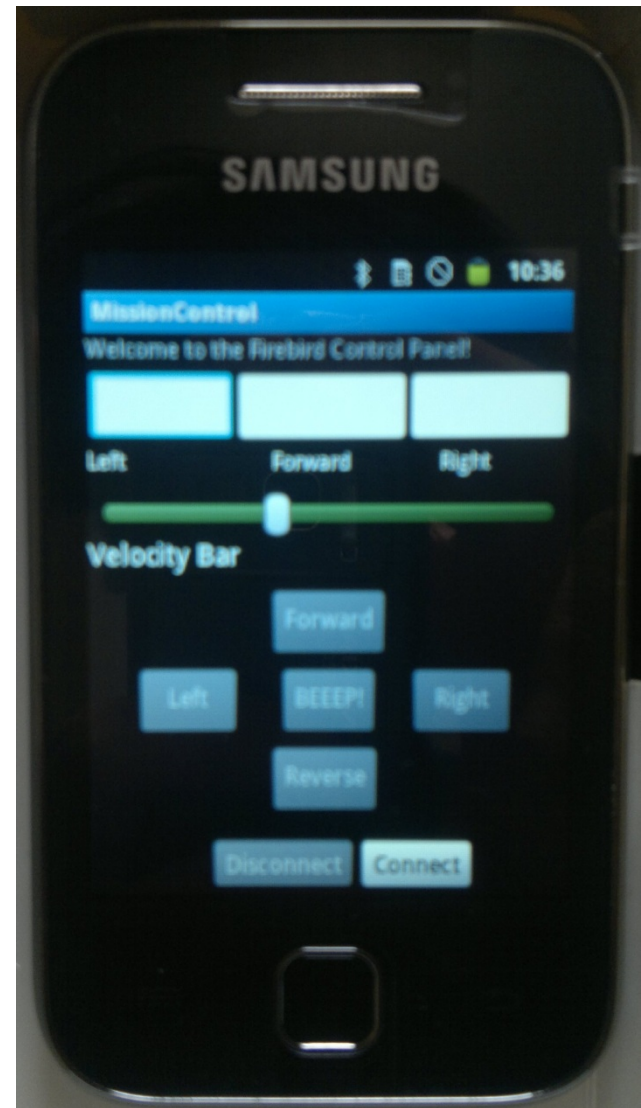
- Communication delay over Bluetooth link:
 - **(100-150 millisecs)**
- Throughput
 - Decides the number of commands that can be sent at maximum rate
 - **7-9 commands/sec**
- System Bottleneck
 - Android phone : **830 MHz** processor
 - Firebird V : **16MHz** processor,
 - Bluetooth link : max **2.1 Mbit/s**, average **0.7 Mbit/s** data rate → **Bottleneck**
 - Apps should be designed to minimize the amount of data transferred between phone and robot, while making full use of the processing power of both

FBController & MissionControl

- Simple apps for illustrating use of the API to control Firebird
- FBController: Low level control of Firebird by sending command messages over bluetooth, similar to HyperTerminal, but for Android
- MissionControl: GUI based Android app which constantly reads sensor values from Firebird, and provides control for motion, velocity and buzzer of the Firebird bot



FBController

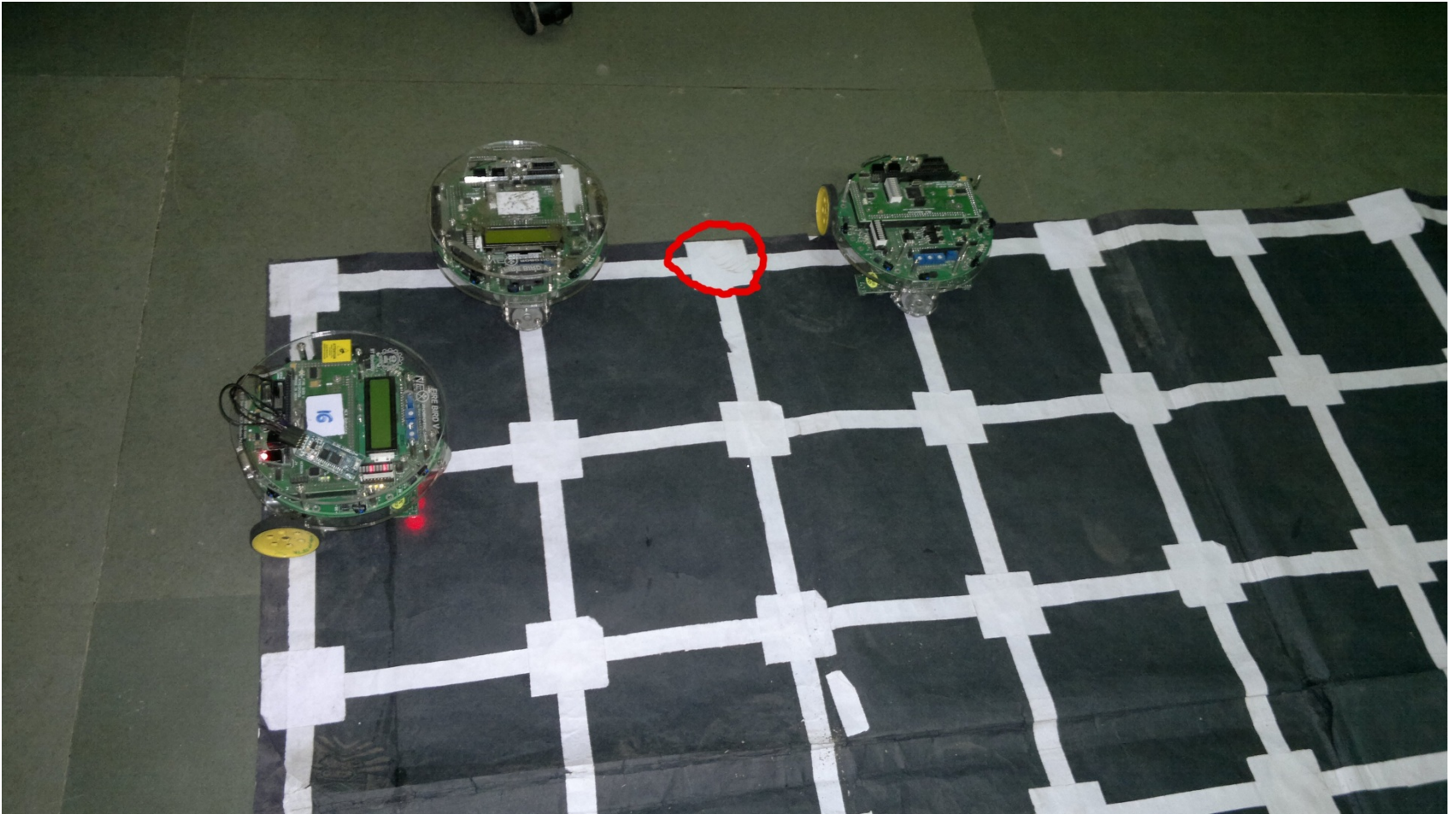


MissionControl

PeterParker

- App for automated parking of the Firebird bot
- Clicking one button on app makes Firebird bot find a parking spot and park itself
- All application logic is coded on Android
- Android device constantly reads sensor values from Firebird, and decides next action
- Whiteline following implemented on the bot itself, to reduce effect of communication delay

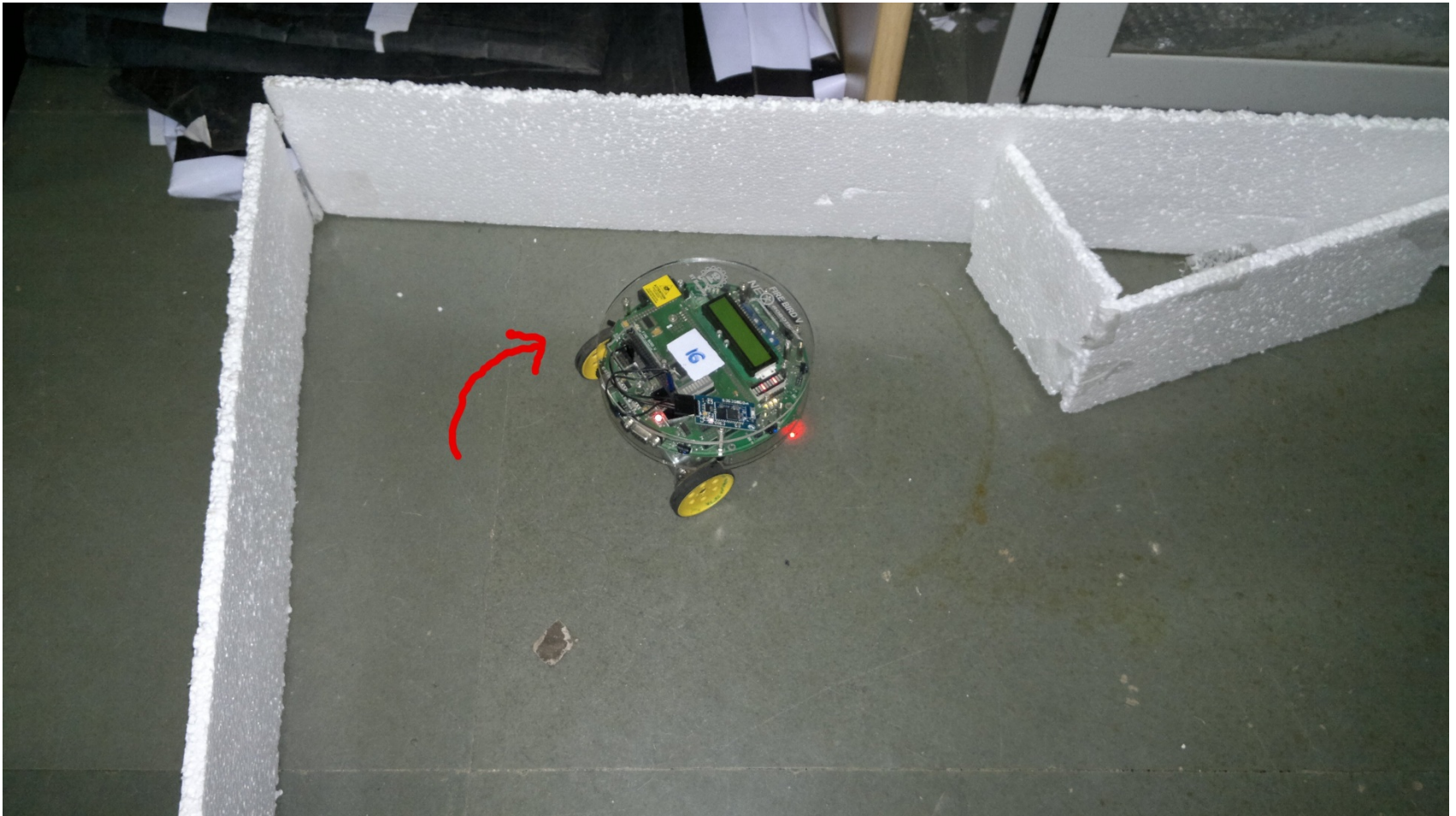
PeterParker In Action



MapperBot

- App for turning Firebird into spy bot
- Bot performs wall following to explore the room
- As bot travels, its location information is read by Android, and a map of the room is drawn incrementally on the phone display
- This map can be saved on phone memory, or emailed directly from within the app
- All application logic for controlling the bot in different situations is coded in Android app

MapperBot Arena



Reusability & Modularity

- The project is coded as separate modules which are replaceable:
- **Different Communication Method:** For example, if the Bluetooth communication hardware is not available, then the user can implement his/her own communication interface by writing a new CommunicationModule class, and with no changes to the other parts of the API
- **Different Bot:** The Firebird Command Interpreter code can be modified to work on a different robot, without any changes to the Android API or the communication protocol. This will allow same API to control different robots (eg hexapod) using an Android phone
- **Different Android Device:** The Android API itself can be implemented for various API levels, so that is configurable for use with any Android device (especially the innovative Aakash tablets)

Future Enhancements

a) **Extensions in the code:**

- Add modules for alternative communication modes. e.g Zigbee, Serial Communication, Wifi, etc. and add functionality to control these from the Android
- Modify our API to work for any other similar bot (e.g Hexapod)

b) **More Applications:**

*Any general application can be developed using our API.
Some specific app ideas are given below*

Apps involving mounted Android phone on Firebird

1. Surveillance Application

- A surveillance app which allows the user to select locations on a map
- The android mounted on the bot will determine its position using GPS and find a route connecting these points
- The bot will now move along the route taking pictures on its way or relaying the video

2. Greenhouse Pest Control Application

- Cameras are mounted in various parts of the greenhouse for surveillance of pests
- When the camera detects a pest attack, it sends a message to the Firebird
- The Firebird then goes to the the location of the pest infected area
- Apps which can control multiple Firebird bots

Apps which can control multiple Firebird bots

This can be accomplished through WiFi module on the Firebird or a central server which communicates to Android devices sitting on the Firebird

Multiple Parking Bot:

In a parking arena, where there are multiple bots waiting in a queue to park, the Android can control these bots separately and park all of them in the arena

Thank You!

“Happy App-ing!”