

eYSIP2016

ANDROID APP DEVELOPMENT FOR FIREBIRD V



INTERN: Jatin Mittal
MENTORS: Yogita Mali
Abhishek Singh
Sachin Gupta

Duration of Internship: 21/05/2016 – 10/07/2016

2016, e-Yantra Publication

Android app development for Firebird V

Abstract

This project is all about bringing all the features of Firebird V on our finger tips through our Android phone. We are able to perform all the basic operations of our robot namely motion control(includes forward motion, backward motion, left rotation, right rotation), getting the readings of all the sensors(includes Sharp sensors, Proximity sensors and White line sensors), changing the velocity using our smart phones. We have achieved this communication with robot wirelessly with the help of bluetooth module.



Objective

The objective of the project was to develop an Android app which can be used for motion control, velocity control and sensor monitoring of Firebird V robot.

Task No.	Task	Deadline
1	Understand the basic concepts of Java and Android	1 Days
2	Make first Android App - Hello World	2 Days
3	Interface the Bluetooth module with Firebird V	2 Days
4	Review the previously developed code and understand it properly	4 Days
5	To modify the UI completely	5 Days
6	Testing the app thoroughly on various Android devices and also on various android versions	3 Days
7	Testing the app to run on various tablets	2 Days



Completion status

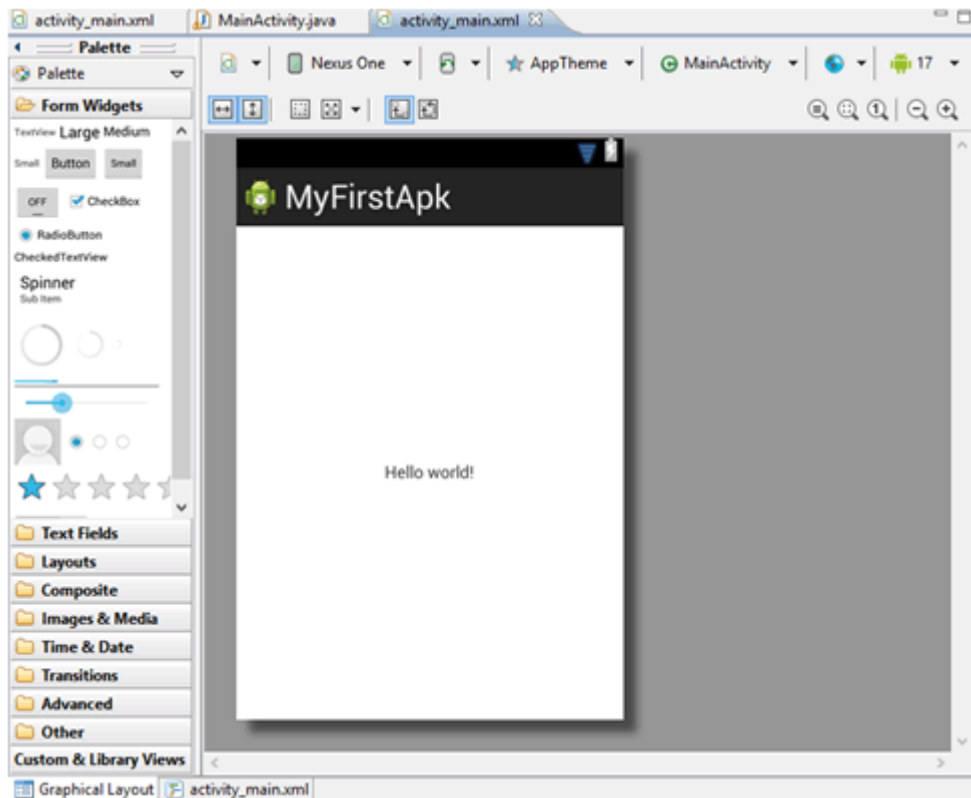
1. Installation and Study basics of Android:

- Installed Android Studio in my computer to develop the Android app.
- Learned about Activity and its life cycle, Services, Intents:
 - Activity- An Activity is an application component that provides a screen with which users can interact in order to do something, such as dial the phone, take a photo, send an email, or view a map. Each activity is given a window in which to draw its user interface. The window typically fills the screen, but may be smaller than the screen and float on top of other windows.

To create an activity, you must create a subclass of Activity (or an existing subclass of it). In your subclass, you need to implement callback methods that the system calls when the activity transitions between various states of its life cycle, such as when the activity is being created, stopped, resumed, or destroyed. The two most important callback methods are `onCreate()` and `onPause()`.
 - Service- A Service is an application component that can perform long-running operations in the background and does not provide a user interface. Another application component can start a service and it will continue to run in the background even if the user switches to another application. Additionally, a component can bind to a service to interact with it and even perform interprocess communication (IPC). For example, a service might handle network transactions, play music, perform file I/O, or interact with a content provider, all from the background.
 - Intent- Intent is an abstract description of an operation to be performed. It can be used with `startActivity` to launch an Activity, `broadcastIntent` to send it to any interested `BroadcastReceiver` components, and `startService(Intent)` or `bindService(Intent, ServiceConnection, int)` to communicate with a background Service.

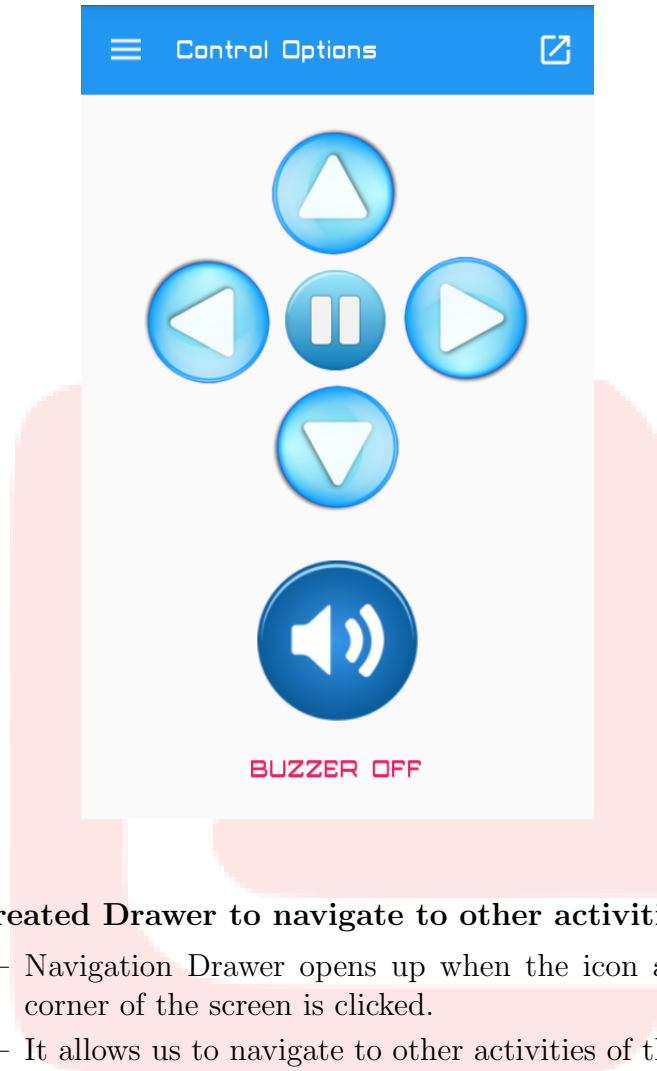


2. Made first Android App - Hello World

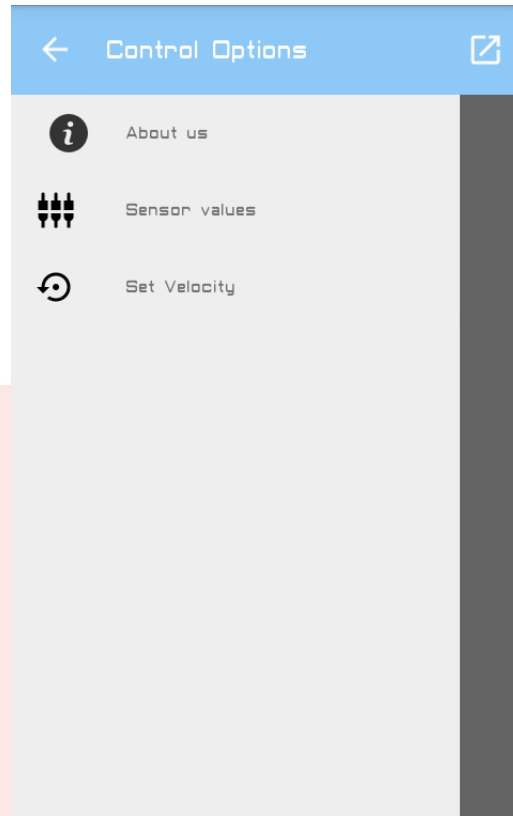


3. Developed UI for different activities

- **Created Control Activity:**
 - This is the first Activity that opens up when the app is launched.
 - This Activity controls the forward, backward, left, right motion of the robot.
 - It also contains the option to turn the buzzer on/off.
 - TextView at the bottom of the Activity displays the current state of the buzzer.
 - On clicking on the icon at the top right corner of the screen, a menu pops up which displays 'Connect' and 'Disconnect' option.

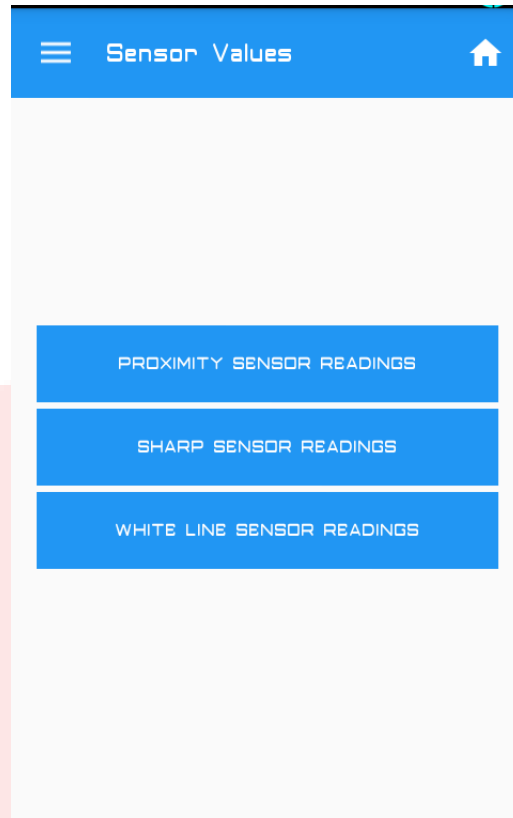


- **Created Drawer to navigate to other activities:**
 - Navigation Drawer opens up when the icon at the top left corner of the screen is clicked.
 - It allows us to navigate to other activities of the app.



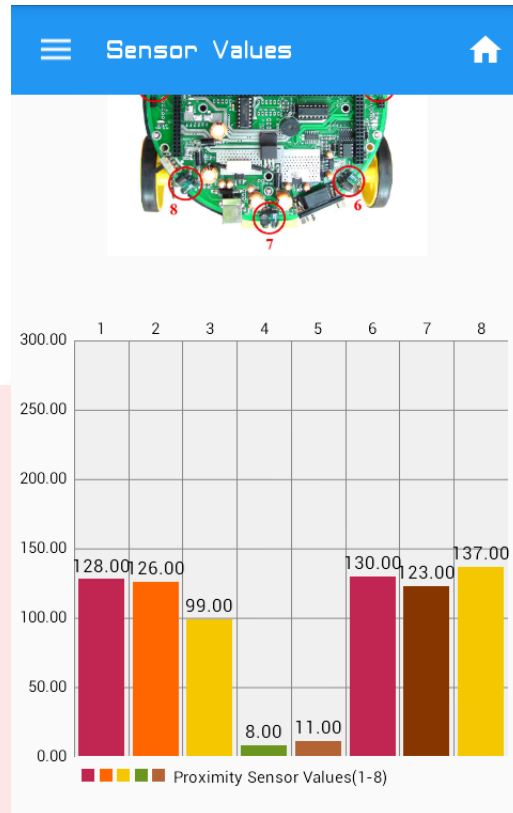
- **Created Select Sensor Activity:**

- This Activity opens up when we select the 'Sensor Values' option from the drawer.
- This activity provides an option for Sharp sensor values, Proximity sensor values and White line sensor values.



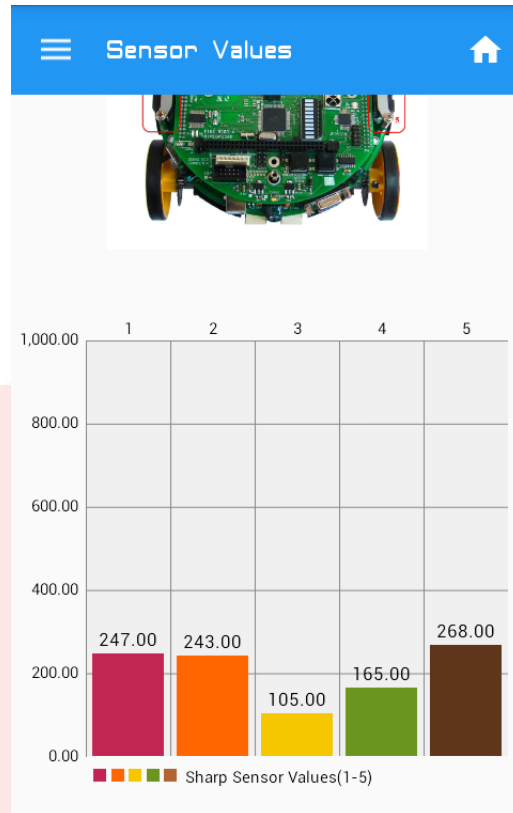
- **Created Proximity Sensor Activity:**

- This Activity contains the bot image which shows the position of different proximity sensors on the bot.
- Below the image, there is a bar graph which displays the proximity sensor values.
- Values of the sensors can be updated by tapping on the chart.



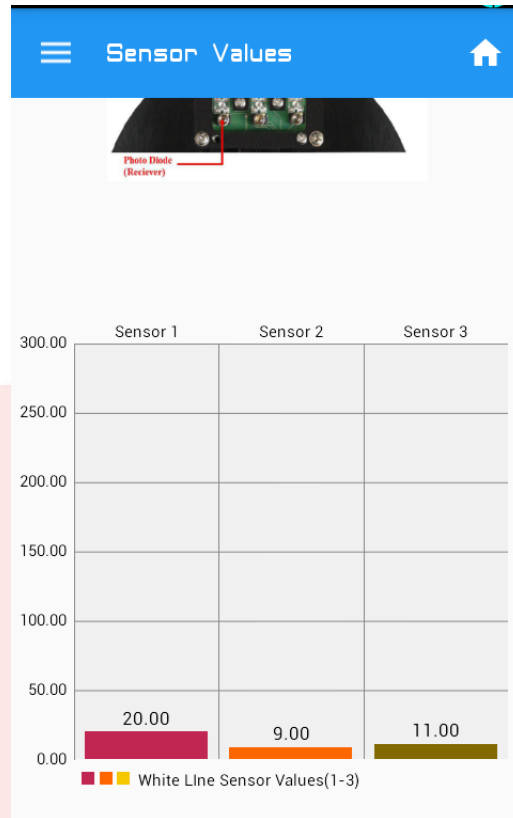
- **Created Sharp Sensor Activity:**

- This Activity contains the bot image which shows the position of different Sharp sensors on the bot.
- Below the image, there is a bar graph which displays the Sharp sensor values.
- Values of the sensors can be updated by tapping on the chart.



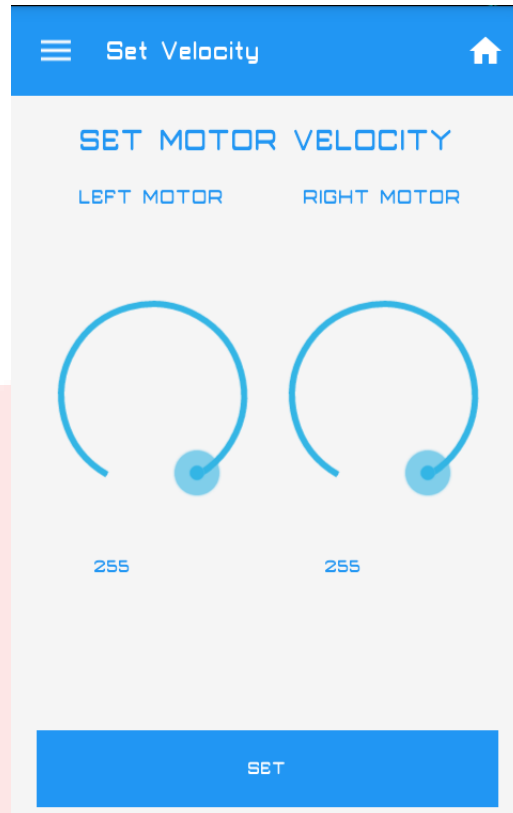
- **Created White Line Sensor Activity:**

- This Activity contains the bot image which shows the position of different White line sensors on the bot.
- Below the image, there is a bar graph which displays the White line sensor values.
- Values of the sensors can be updated by tapping on the chart.



- **Created Set Velocity Activity:**

- This Activity opens up when we click on the 'Set Velocity' option in the drawer.
- It contains two circular seekbar to set the velocities of left and right motor of the bot.
- Below the seekbars, there are TextView which displays the velocity value.
- On clicking on the set button, the chosen velocity is set and the Control Activity is launched where we can control the bot motion with the velocity that we set.



4. Testing

- Tested the app thoroughly on various Android devices and different versions of Android.
- Tested the app on various tablets.

Pending Task

Adding the content in 'About Us' Activity is left.

Hardware used

- Firebird V Robot
- BlueLINK - bluetooth module
- Android Device
- [Firebird V Robot - Datasheet](#)



- [Firebird V Robot - Vendor link](#)
- [BlueLINK - Datasheet](#)
- [BlueLINK - Vendor link](#)

Software used

- Android Studio v2.1.2 - [Download](#)
- Java SE Development Kit 8 - [Downlaod](#)
- [Installation steps](#)

Software and code

- Code : [Github link](#)

Use and demo

This is an Android Application to perform the basic testing of Firebird V robot. This application can be use to control the forward, backward, left, right motion of the bot, turning the buzzer on/off, setting the velocity of both the motors of the bot, and getting the different sensor values.

Use the following steps to test the Firebird V :

1. First, turn on the Firebird V robot and pair the Android device to the Firebird V bluetooth by going into the bluetooth option of your phone.
2. Set the pin as 8888 for establishing the connection with the bot.
3. Once the devices are paired, launch the application and select the device name from the paired devices list which is displayed on the screen.
4. After the connection is established, test the bot by clicking on different buttons on the screen. This includes forward, backward, left, right motion, turning buzzer on/off.
5. Motion of the bot can also be tested with different velocity of the bot. Open up the drawer and select the 'Set Velocity' option to set the different velocity of the bot.



6. For checking the different sensor values, click on the 'Sensor values' option from the drawer and click on one of the buttons from 'proximity sensor readings', 'sharp sensor readings', 'white line sensor readings' to get the corresponding sensor values.
7. After done with the testing, move to the control option Activity and click on the disconnect option that appears on clicking the icon at the top right corner of the screen.

Future Work

- In future work, the same project can be implemented with wifi module and USB cable instead of bluetooth module.
- And, the possibility of improving the UI is always there.

Bug report and Challenges

- The major problem with bluetooth module is that as it is short range it takes time to connect with bluetooth of our phone. We should keep our phone nearer to it so that the connection can be established faster. One major problem is that if once the connection is lost then robot has to be switched off and turned on again in order to establish connection again.
- One more issue is that after the successful connection made with the bot, the Title bar in Control Activity updates the subtitle that the connection is made, but if we return back to the same activity from some other activity it does not display the subtitle even if the connection is still there.



References

- <http://www.androidhive.info/>
- <http://www.stackoverflow.com>
- <https://github.com/neild001/SeekArc>
- <https://www.numericlabz.com/android-bar-chart-using-mpandroidchart-library-tutorial/>
- <https://www.tutorialspoint.com>
- <https://www.developer.android.com>
- <https://github.com/PhilJay/MPAndroidChart>
- <http://stackoverflow.com/questions/22038576/how-to-draw-bar-chart-and-also-pie-chart-in-android>
- <https://developer.android.com/training/implementing-navigation/nav-drawer.html>
- <https://www.youtube.com/watch?v=h57QpXp2TRg>
- <https://www.youtube.com/watch?v=hrIGVU8z7zc>
- <https://www.youtube.com/watch?v=pMO8EVkhJO8>
- <https://www.youtube.com/watch?v=4XfDDfa3rv8>
- <https://github.com/slidenerd/Material-Design-Test-App-DEPRECATED->