International Journal of Advance Research in Engineering, Science & Technology(IJAREST),
ISSN(O):2393-9877, ISSN(P): 2394-2444,
Volume 2,Issue 11, November-2015, Impact Factor: 2.125

A Novel Approach of Broadcast Receiver on the Android Operating System

Aritra Chakraborty[1], Prof. Satyasaran Changdar[2], Arijit Das[3]

[1]Information Technology, Institute of Engineering and Management, Kolkata, aritra.2411@gmail.com
[2]Information Technology, Institute of Engineering and Management, Kolkata,satyasaran@gmail.com
[3]Information Technology, Institute of Engineering and Management, Kolkata, arijitd673@gmail.com

Abstract

Broadcast Receiver is an extremely important component for communication in the Android operating system. The power of broadcast receiver is very high and reflected during communication.Thus,the present research is carried out to implement the applicability of Broadcast Receiver for the Android Operating System.

Keywords*-Broadcast Receiver; Android; Communication; Telephony Manager ;Incoming call; Outgoing call*

I. INTRODUCTION

A broadcast receiver is a component of Android that enables the user to register specific system as well as application events. It is an always on application. Once the application is in its runtime all the registered users are notified. It can be used to manage incoming calls, outgoing calls , and flight mode activation or deactivation by notifying the user.

The Broadcast Receiver in an android operating system works by notifying the user in case of an incoming call outgoing calls. The TelephonyManager API. This class provides access to the telephone services on that device.

II. MANAGING INCOMING CALLS WITH BROADCAST RECEIVER

This method is called when the Broadcast Receiver is receiving an Intent Broadcast. When the user gets a call the notification is received by the user with the message or a toast notification. The method TelephonyManager.*EXTRA_INCOMING_NUMBER* is used to display the telephone number of the user. A Toast Message can be displayed on the same with the event Toast.*makeText* (context, "phone is ringing : " + no, Toast.*LENGTH_LONG*).show();

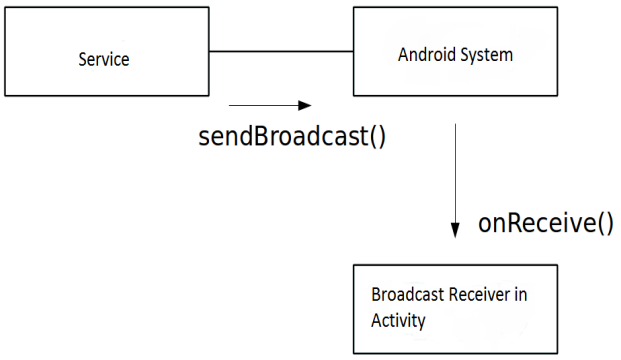

Figure 1. Incoming call model for Broadcast Receiver

III. MANAGING OUTGOING CALLS WITH BROADCAST RECEIVER

This method is called when the Broadcast Receiver is sending an Intent Broadcast. When the user dials a number the notification is shown to the user with the message or a toast notification. The method TelephonyManager.*EXTRA_STATE_OFFHOOK* is used to validate that the number is dialled. When the receiver on the other end picks up the call a toast notification "phone is answered" is shown at the activity of the user. The event for the toast is Toast.*makeText* (context, "phone is ringing : " + no, Toast.*LENGTH_LONG*).show();

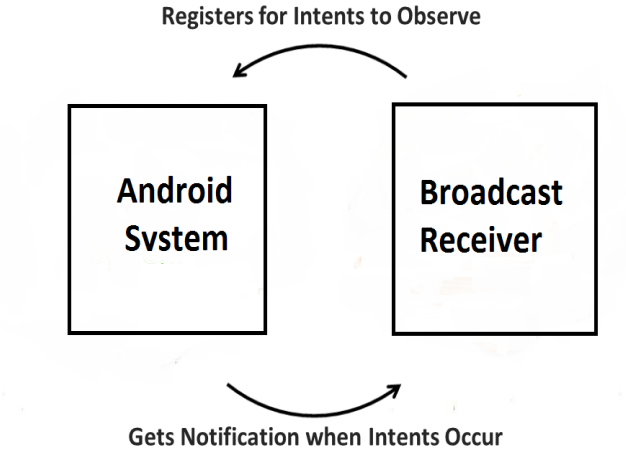

Figure 2. Outgoing call model and Intent for Broadcast Receiver

3.1. Disconnecting a Call

When the user disconnects a call the broadcast receiver again senses it and the Intents occur. TelephonyManager.*EXTRA_STATE_IDLE* . Telephony manager gets activated and the message or toast "phone is disconnected is shown to the user."

IV. MANAGING FLIGHT MODE WITH BROADCAST RECEIVER

This method is called when the Broadcast Receiver is sending an Intent Broadcast. When the user taps on the flight or airplane mode, the notification is shown to the user with the message or a toast notification. Intent.*ACTION_AIRPLANE_MODE_CHANGED* is used for capturing the change and responding to the change in the airplane mode. A Toast notification is immediately

generated with the activation or deactivation. Toast.*makeText*(context, "flight mode activated/deactivated" , Toast.*LENGTH_LONG*).show(); This notification is shown to the user as soon as the intent starts.

V. TAKING ALL THE ACTIVITIES TOGETHER

A Single activity or application can be made to collaborate all the events. This enables the user to handle multiple tasks using a single application. The concept of threading is used for this implements. The main activity has the method, **public class** MainActivity **extends** Activity.

A single permission is required for accessing the phone state of the broadcast receiver.

android.permission.READ_PHONE_STATE

VI. GRAPHICAL LAYOUT OF THE APPLICATION

XML is used for the graphical layout of a corresponding application. The layout is simple as the application always runs in the background unless it is uninstalled.

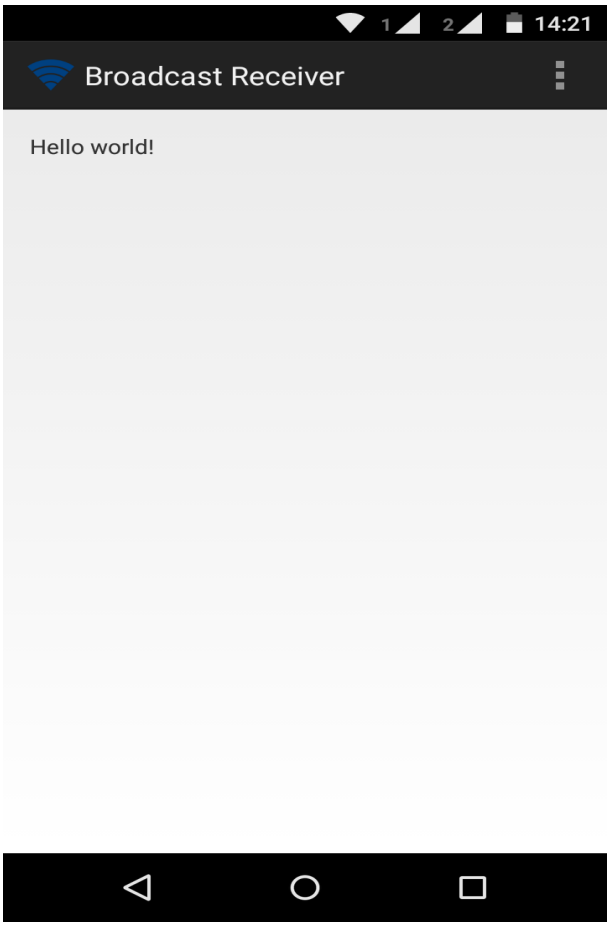

Figure 3. The Activity screen of Broadcast Receiver

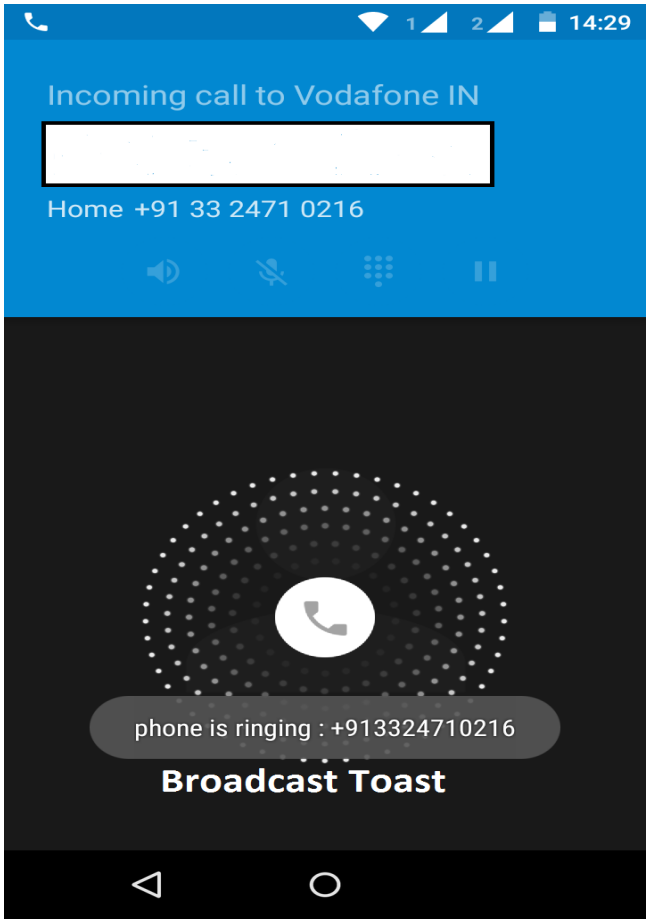

Figure 4. The Toast Message on receiving a call

VII. CONCLUSIONS

The Broadcast Receiver is an excellent tool for an Android Receiver. It starts the intent on each and every call process that takes place. It has both software and hardware based implementation. An application was designed for the study of Broadcast Receiver which uses the Intent for Incoming calls, Outgoing calls and Flight modes. The application was successively tested on multiple devices and it ran smoothly. Thus, the Broadcast Receiver based study is an excellent subject for an Android Developer.

REFERENCES

[1] Marko Gargenta, “Learning Android(English)” , Chapter 5, Broadcast Receivers

[2] Ed Burnette, Hello Android , Broadcast Receivers, 3rd Edition. , Shroff Publishers & Distributors Pvt Ltd

[3] Reto Meier, “Professional Android 4 Application Development” , Chapter “Intents and Braodcast Receivers”

[4] Chris Haseman , “Android Essentials” , Chapter “Broadcast Receivers” , 2008 Edition.

[5] Wei-Meng Lee, “Beginning Application 2 Professional Development, “Broadcast Receivers” , John Wiley and Sons Inc.

[6] Blake Meike, Rick Rogers, Zigurd Mednieks and John Lombardo, "Specifications of Android Application Development" , Chapter "Telephony State Information and Telphony Classes.

[7] Herbert Schildt, Java The Complete Reference, Threads in Java.

[8] J. F. DiMarzio, Android: A ProgrammerS Guide, Chapter "Android Intents , Filters and Receivers" , McGraw Hill Education.

[9] W. Frank Ableson, Robi Sen, Chris King, C. Enrique Ortiz, "Android In Action(English) 3rd Edition" , Chapters , "Intents and Services" and "Telephony", DreanTech Press

[10] Ian F. Darwin, Android Cookbook, UI and Interfaces, O 'Reilly.