电子科技大学信息与软件工程学院

**实 验 报 告**

学 号 2017221303023

姓 名 陈奎

（实验） 课程名称 移动计算

理论教师 许毅

实验教师 许毅

**电子科技大学教务处制表**

**电 子 科 技 大 学**

**实 验 报 告**

**学生姓名：陈奎 学号：2017221303023 指导教师：许毅**

**实验地点：信软楼西304 实验时间：19.11.8**

**一、实验名称：**广播域通知

**二、实验学时：**2学时

**三、实验目的：**

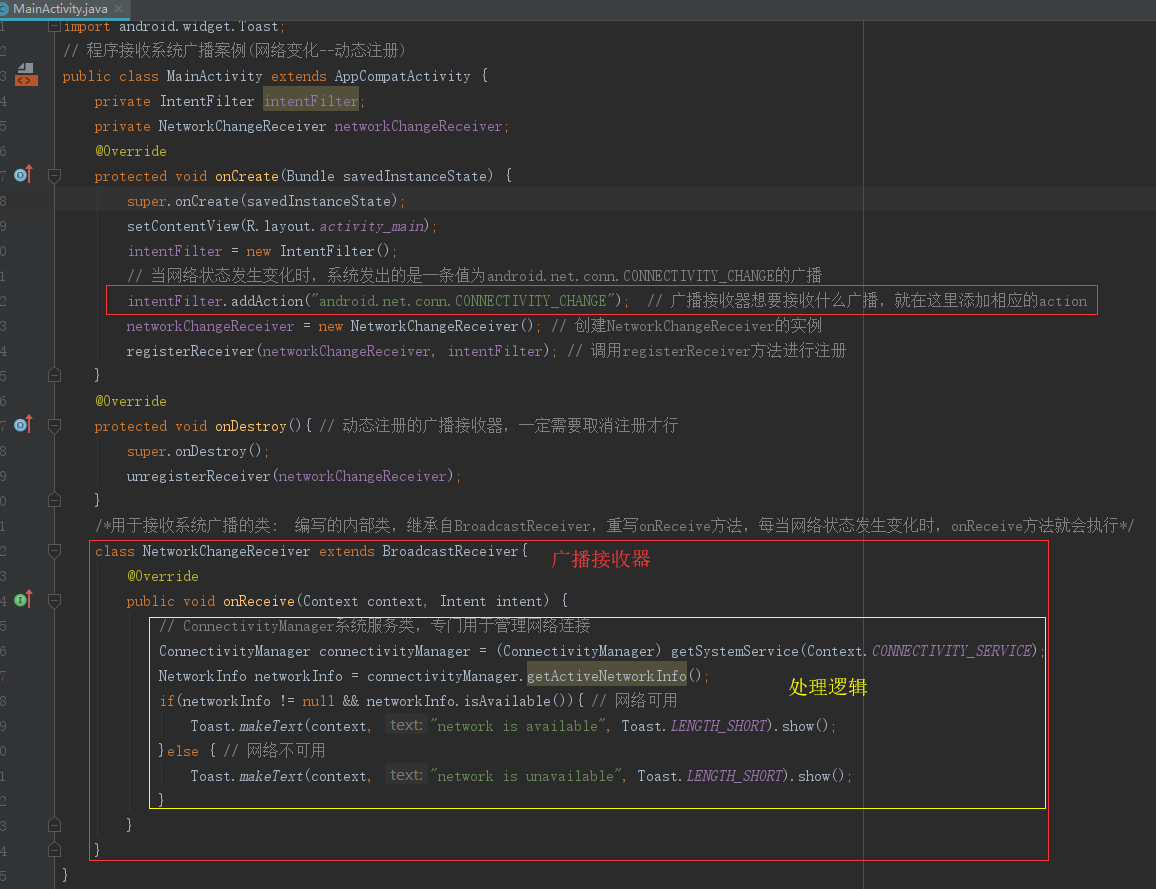
1. 学习和掌握Android系统的广播机制
2. 学习和掌握Android的Notification功能
3. 学习静态和动态注册广播接收器
4. 学习和了解同步和异步处理的区别
5. 学习和掌握短信管理

**四、实验原理：**

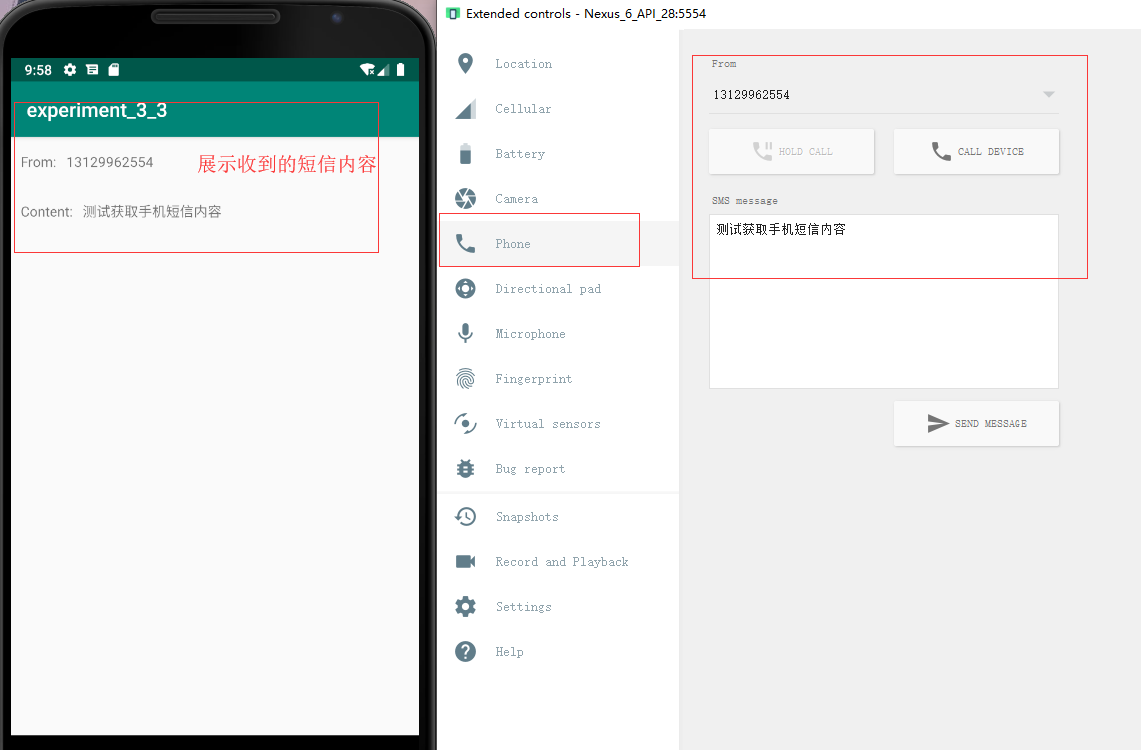
基于AndroidStdio进行一系列的产品开发实践。

**五、实验内容：**

1. 编写程序接收系统广播
2. 编写程序发送通知

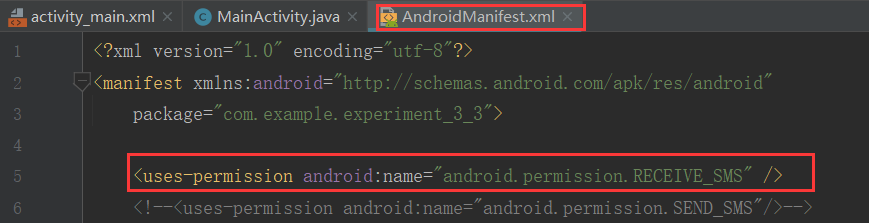


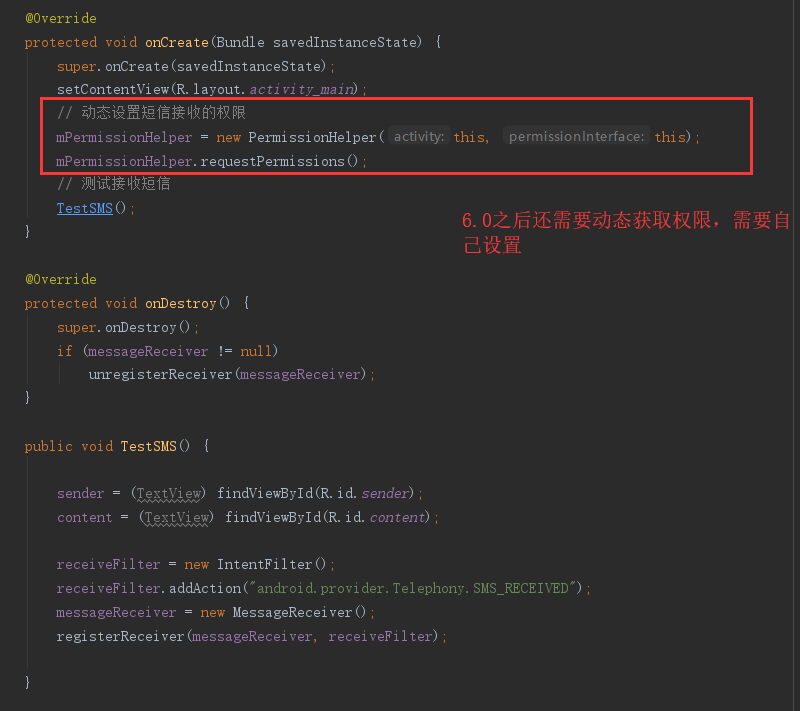
1. 编写程序管理移动端短信



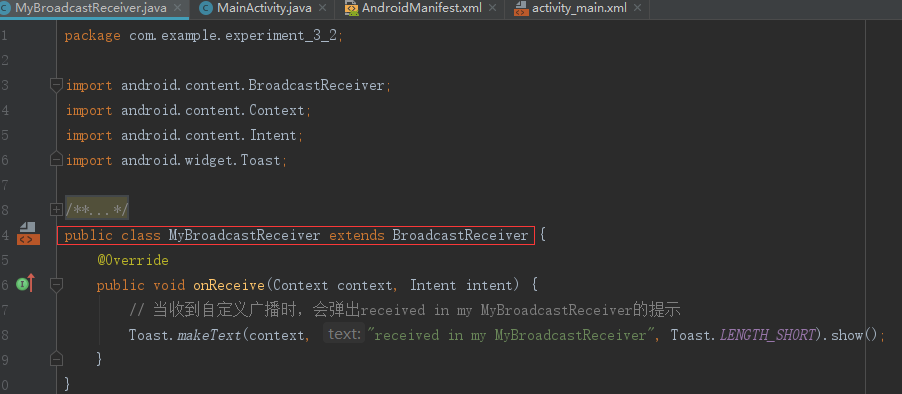
布局文件

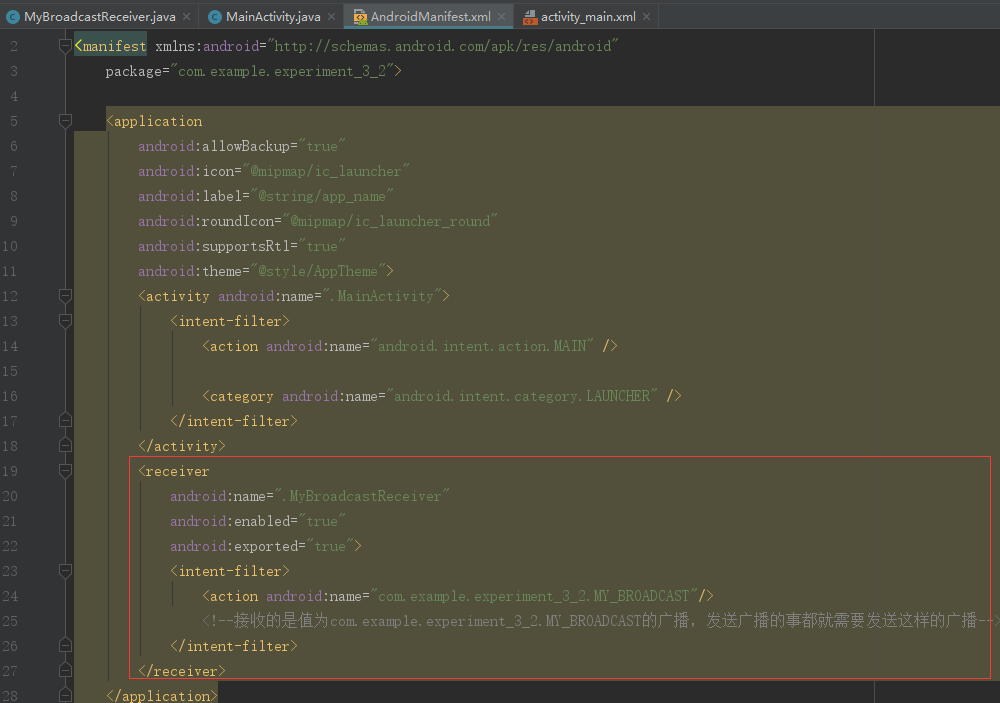






1. 实现自定义发送和接收广播功能







1. 编译、调试和查看程序运行结果

**六、实验器材（设备、元器件）：**

本实验开设方式为个人实验；实验授课时间0.5学时，上机3.5学时，每人一台PC单独完成实验，使用Eclipse（或Android Studio）、模拟器（或Android手机）编程调试。

**七、实验结果与分析（含重要数据结果分析或核心代码流程分析）**

1.在Android Studio开发环境中创建Android项目

2.编写程序发送通知，检测程序的飞行状态是否发生改变

Activity\_main.xml文件

|  |
| --- |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:app="http://schemas.android.com/apk/res-auto"  xmlns:tools="http://schemas.android.com/tools"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  tools:context=".MainActivity"**>  <**TextView  android:id="@+id/textView"  android:layout\_width="220dp"  android:layout\_height="86dp"  android:layout\_marginTop="144dp"  android:gravity="center"  android:text="查看飞行模式的状态"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintHorizontal\_bias="0.497"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toTopOf="parent"** />  </**androidx.constraintlayout.widget.ConstraintLayout**> |

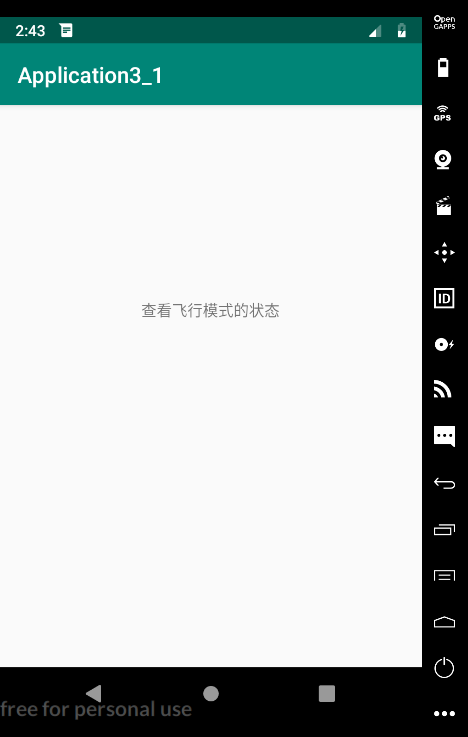
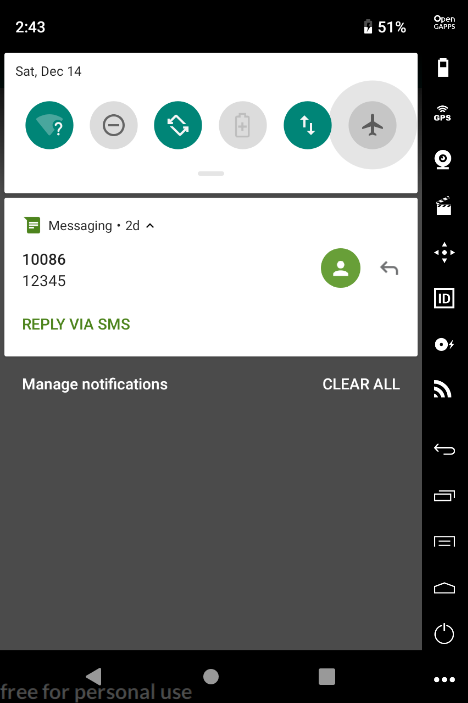
MainActivity.java文件

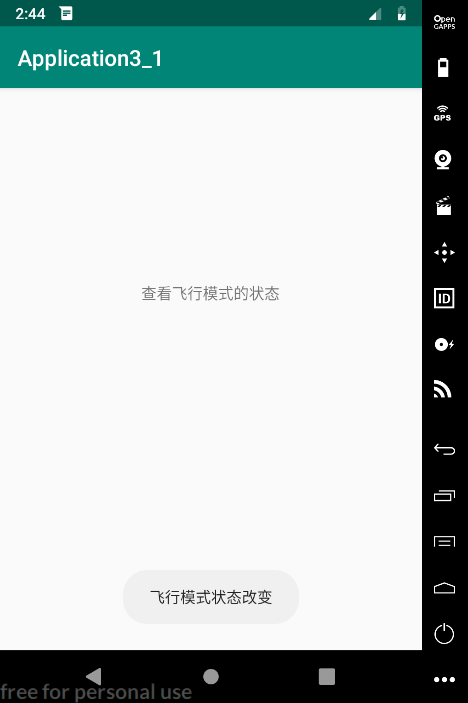
|  |
| --- |
| **package** com.example.application3\_1;  **import** androidx.appcompat.app.AppCompatActivity;  **import** android.content.Intent; **import** android.content.IntentFilter; **import** android.os.Bundle;  **public class** MainActivity **extends** AppCompatActivity {  **private** IntentFilter **intentFilter**;  **private** GetReceiver **getReceiver**;    @Override  **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_main***);  getMessage();  }   **public void** getMessage(){  **intentFilter**=**new** IntentFilter();  **intentFilter**.addAction(Intent.***ACTION\_AIRPLANE\_MODE\_CHANGED***);  **getReceiver**=**new** GetReceiver();  registerReceiver(**getReceiver**,**intentFilter**);  }   @Override  **protected void** onDestroy() {  **super**.onDestroy();  *//动态注册的广播接收器需取消注册* unregisterReceiver(**getReceiver**);  }  } |

Getreceiver.java代码

|  |
| --- |
| **package** com.example.application3\_1;  **import** android.content.BroadcastReceiver; **import** android.content.Context; **import** android.content.Intent; **import** android.widget.Toast;  **public class** GetReceiver **extends** BroadcastReceiver {  @Override  **public void** onReceive(Context context, Intent intent) {  String action = intent.getAction();  **if**(action.equals(Intent.***ACTION\_AIRPLANE\_MODE\_CHANGED***)){  *//飞行模式状态改变* Toast.*makeText*(context,**"飞行模式状态改变"**,Toast.***LENGTH\_SHORT***).show();  }  } } |

运行结果，当手机的运行状态改变时发出通知



3.编写程序管理移动端短信

Activity.main文件

|  |
| --- |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:orientation="vertical"**>   <**LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="50dp"**>  <**TextView  android:id="@+id/textView19"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:padding="10dp"  android:gravity="center"  android:text="From:"** />   <**TextView  android:id="@+id/sender"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:gravity="center"** />  </**LinearLayout**>  <**LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="50dp"**>  <**TextView  android:id="@+id/textView21"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:gravity="center"  android:text="Content:"  android:padding="10dp"** />   <**TextView  android:id="@+id/content"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:gravity="center"** />  </**LinearLayout**>   <**Button  android:id="@+id/but\_getContent"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_marginTop="40dp"  android:text="获取短信内容"  android:onClick="getContent"** />  </**LinearLayout**> |

MainActivity.java文件

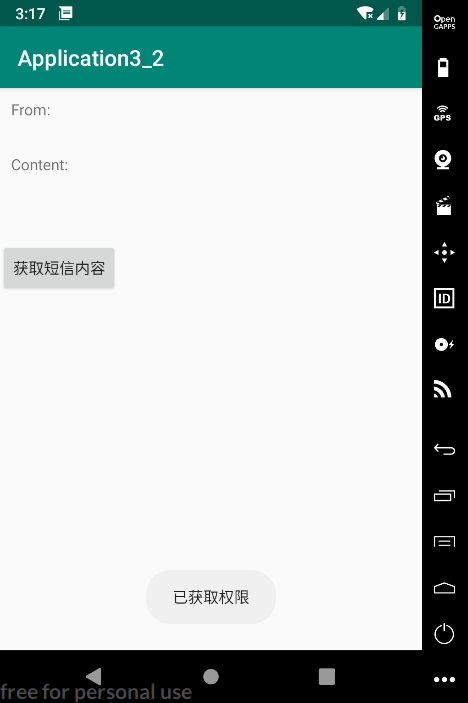
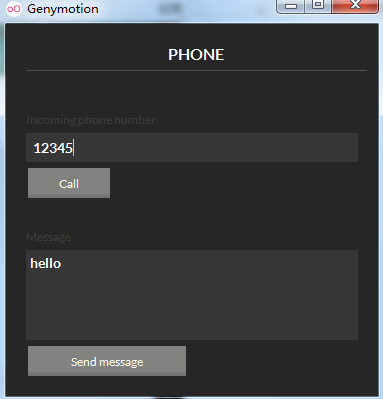
|  |
| --- |
| **package** com.example.application3\_2;  **import** androidx.appcompat.app.AppCompatActivity; **import** androidx.core.app.ActivityCompat; **import** androidx.core.content.ContextCompat;  **import** android.Manifest; **import** android.content.BroadcastReceiver; **import** android.content.Context; **import** android.content.Intent; **import** android.content.IntentFilter; **import** android.content.pm.PackageManager; **import** android.os.Bundle; **import** android.telephony.SmsMessage; **import** android.view.View; **import** android.widget.TextView; **import** android.widget.Toast;  **class** GetSMSReceiver **extends** BroadcastReceiver {  SmsMessage **msg**=**null**;  String **Address**=**""**,**messageBody**=**""**;  @Override  **public void** onReceive(Context context, Intent intent){  String action=intent.getAction();  **if** (action.equals(**"android.provider.Telephony.SMS\_RECEIVED"**)){  Object[] pduses = (Object[]) intent.getExtras().get(**"pdus"**);  **for** (Object pduse : pduses) {  **msg** = SmsMessage.*createFromPdu*((**byte**[]) pduse);  **Address** = **msg**.getOriginatingAddress();  **messageBody** = **msg**.getMessageBody();  Toast.*makeText*(context,**"短信内容："** + **messageBody** + **" 发送者："** + **Address**,Toast.***LENGTH\_SHORT***).show();  }  }  } }  **public class** MainActivity **extends** AppCompatActivity {  **private static final int *MY\_PERMISSION\_REQUEST\_CODE*** = 10000;  **private** IntentFilter **intentFilter**;  **private** GetSMSReceiver **getSMSReceiver**;  */\*  \* 采用动态注册的方法来创建广播监听  \* \*/* @Override  **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_main***);   checkSMSPermission();  getMessage();   }   @Override  **protected void** onDestroy() {  **super**.onDestroy();  *//动态注册的广播接收器需取消注册* unregisterReceiver(**getSMSReceiver**);  }   **public void** getMessage(){  *// 当系统接收到消息的时候发送一个"android.provider.Telephony.SMS\_RECEIVED"的广播* **intentFilter**=**new** IntentFilter();  **intentFilter**.addAction(**"android.provider.Telephony.SMS\_RECEIVED"**); *// 创建实例* **getSMSReceiver**=**new** GetSMSReceiver(); *// 注册实例* registerReceiver(**getSMSReceiver**,**intentFilter**);  }   **public void** getContent(View view){  TextView sender=(TextView)findViewById(R.id.***sender***);  TextView content=(TextView)findViewById(R.id.***content***);  sender.setText(**getSMSReceiver**.**Address**);  content.setText(**getSMSReceiver**.**messageBody**);  }   *//检查申请短信权限* **private void** checkSMSPermission() {  **if** (ContextCompat.*checkSelfPermission*(**this**, Manifest.permission.***RECEIVE\_SMS***) != PackageManager.***PERMISSION\_GRANTED***) {  *//未获取到读取短信权限* Toast.*makeText*(**this**,**"未获取权限"**,Toast.***LENGTH\_SHORT***).show();  *// 向系统申请权限* ActivityCompat.*requestPermissions*(**this**,**new** String[]{  Manifest.permission.***READ\_SMS***,  Manifest.permission.***RECEIVE\_SMS*** },***MY\_PERMISSION\_REQUEST\_CODE***);  }  **else**{  Toast.*makeText*(**this**,**"已获取权限"**,Toast.***LENGTH\_SHORT***).show();  }  }  } |

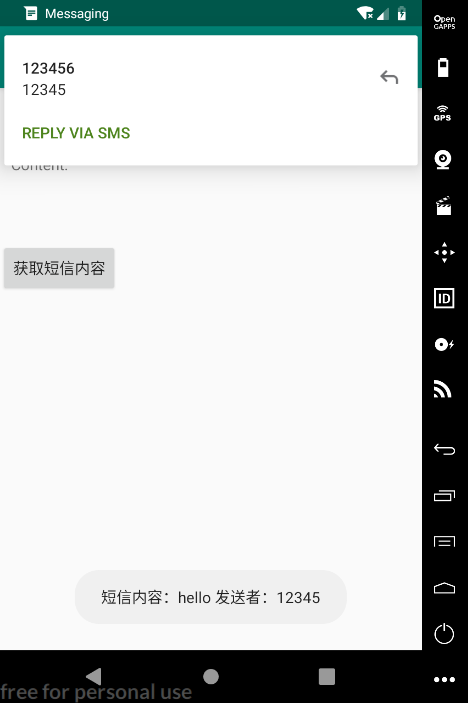
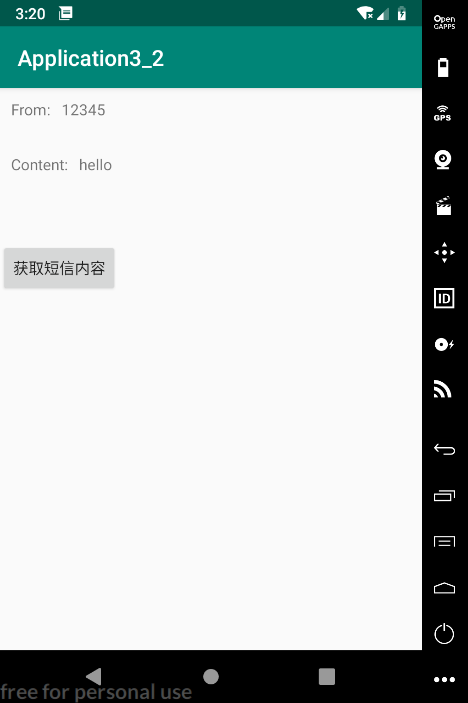
AndroidManifest.xml中添加

|  |
| --- |
| <**uses-permission android:name="android.permission.RECEIVE\_SMS"**/> <**uses-permission android:name="android.permission.READ\_SMS"**/> <**uses-permission android:name="android.permission.SEND\_SMS"** /> |

运行结果：

用软件向手机发送短信，手机收到消息展示一个提醒，点击获取消息得到内容

4.实现自定义发送和接收广播功能

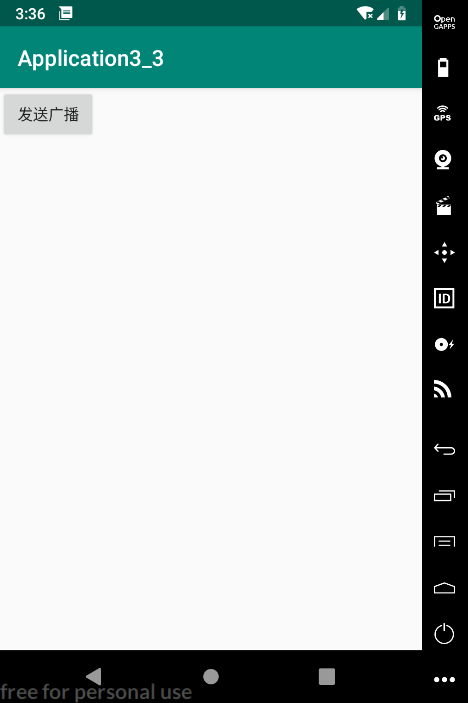
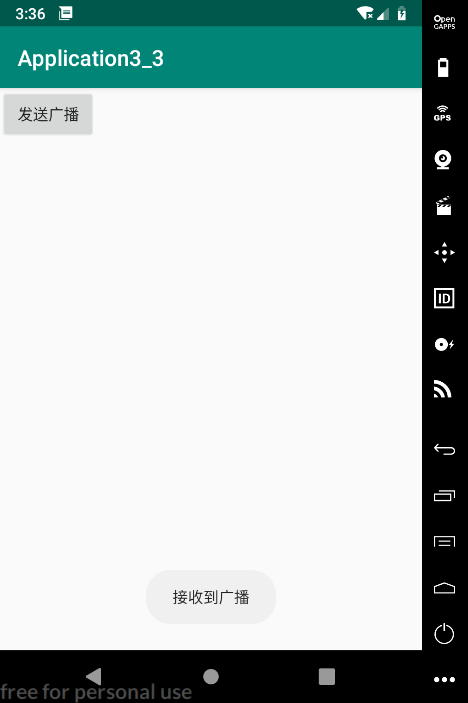
Activity\_main.xml文件

|  |
| --- |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"** >   <**Button  android:id="@+id/btn\_sendBroad"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="发送广播"  android:onClick="sendBroad"** />  </**LinearLayout**> |

MainActivity.java文件

|  |
| --- |
| **package** com.example.application3\_3;  **import** androidx.appcompat.app.AppCompatActivity;  **import** android.content.BroadcastReceiver; **import** android.content.Context; **import** android.content.Intent; **import** android.content.IntentFilter; **import** android.os.Bundle; **import** android.view.View; **import** android.widget.Toast;   **class** BroadReceiver **extends** BroadcastReceiver {  @Override  **public void** onReceive(Context context, Intent intent){  System.***out***.println(**"shabi"**);  String action=intent.getAction();  **if**(action.equals(**"pers.example.broad2.MY\_BROADCAST"**)){  Toast.*makeText*(context,**"接收到广播"**,Toast.***LENGTH\_SHORT***).show();  }  } }  **public class** MainActivity **extends** AppCompatActivity {  **private** IntentFilter **intentFilter**;  **private** BroadReceiver **broadReceiver**;  @Override  **protected void** onCreate(Bundle savedInstanceState) {   **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_main***);  getBroad();  }   **public void** sendBroad(View view){  Intent intent=**new** Intent(**"pers.example.broad2.MY\_BROADCAST"**);  sendBroadcast(intent);  }   **public void** getBroad(){  **intentFilter**=**new** IntentFilter();  **intentFilter**.addAction(**"pers.example.broad2.MY\_BROADCAST"**);  **broadReceiver**=**new** BroadReceiver();  registerReceiver(**broadReceiver**,**intentFilter**);  } } |

运行结果：点击发送广播后接收到提示

**八、总结及心得体会：**

通过本次实验了解了Android开发中关于广播的基本操作。在发送广播前要先定义一个接收器，先确定目标，自定义一个BroadcastReceiver,重写onReceiver()方法并注册。注册广播的方式有两种，在代码中注册和在清单文件中注册，前者称为动态注册，后者称为静态注册。

**报告评分：**

**指导教师签字：**