

# 中山大学移动信息工程学院本科生实验报告

## ( 2016 年秋季学期 )

课程名称：移动应用开发

任课教师：郑贵锋

|      |             |           |                            |
|------|-------------|-----------|----------------------------|
| 年级   | 14          | 专业 ( 方向 ) | 物联网                        |
| 学号   | 14353259    | 姓名        | 商家煜                        |
| 电话   | 13211143626 | Email     | shangjy3@mail2.sysu.edu.cn |
| 开始日期 | 2016.9.22   | 完成日期      | 2016.9.22                  |

### 一、 实验题目

- 1、掌握 Broadcast 编程基础
- 2、掌握动态注册 Broadcast 和静态注册 Broadcast
- 3、掌握Notification 编程基础

### 二、 实现内容

实现一个Android 应用，实现静态广播、动态广播两种改变Notification 内容的方法。  
具体要求：

- (1) 该界面为应用启动后看到的界面。



(2) 点击静态注册按钮，跳转至如下界面



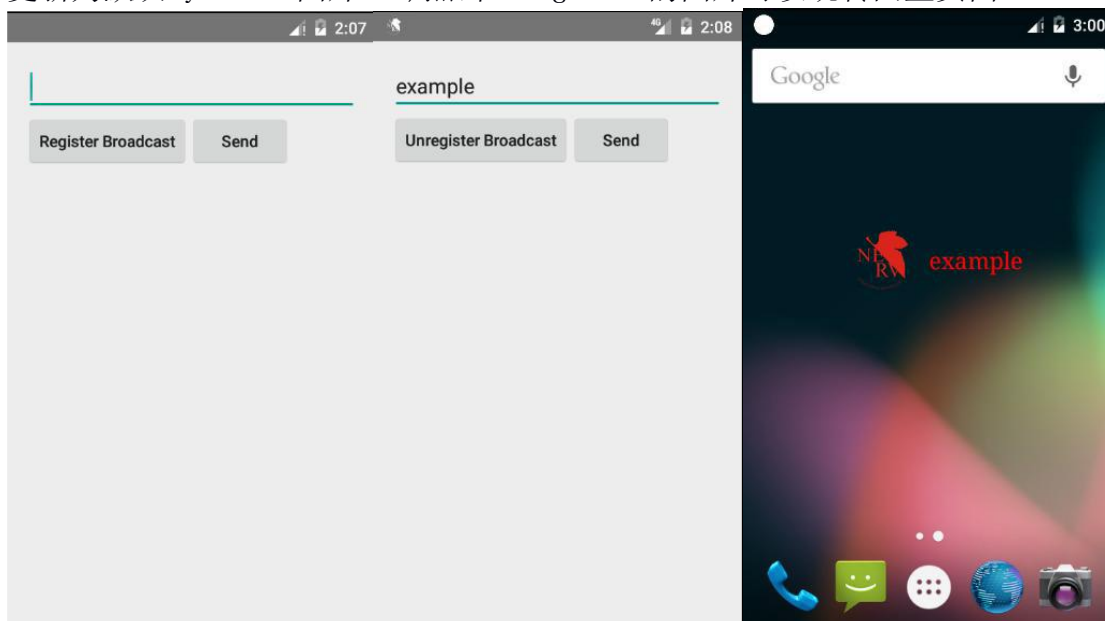
点击表单项目。如 banana。widget 会发生对应变化。点击 Widget 上的图片可以跳转回主页面



(3) 点击动态注册按钮，跳转至如下界面。

实现以下功能：

- a) 可以编辑广播的信息，点击Send 按钮发送广播。
- b) 设置一个按钮进行广播接收器的注册与注销。
- c) 广播接收器若已被注册，发送出的广播信息能够及时更新桌面上 Widget 上文字内容及更新为默认dynamic 图片。
- d) 点击 Widget 上的图片可以跳转回主页面。



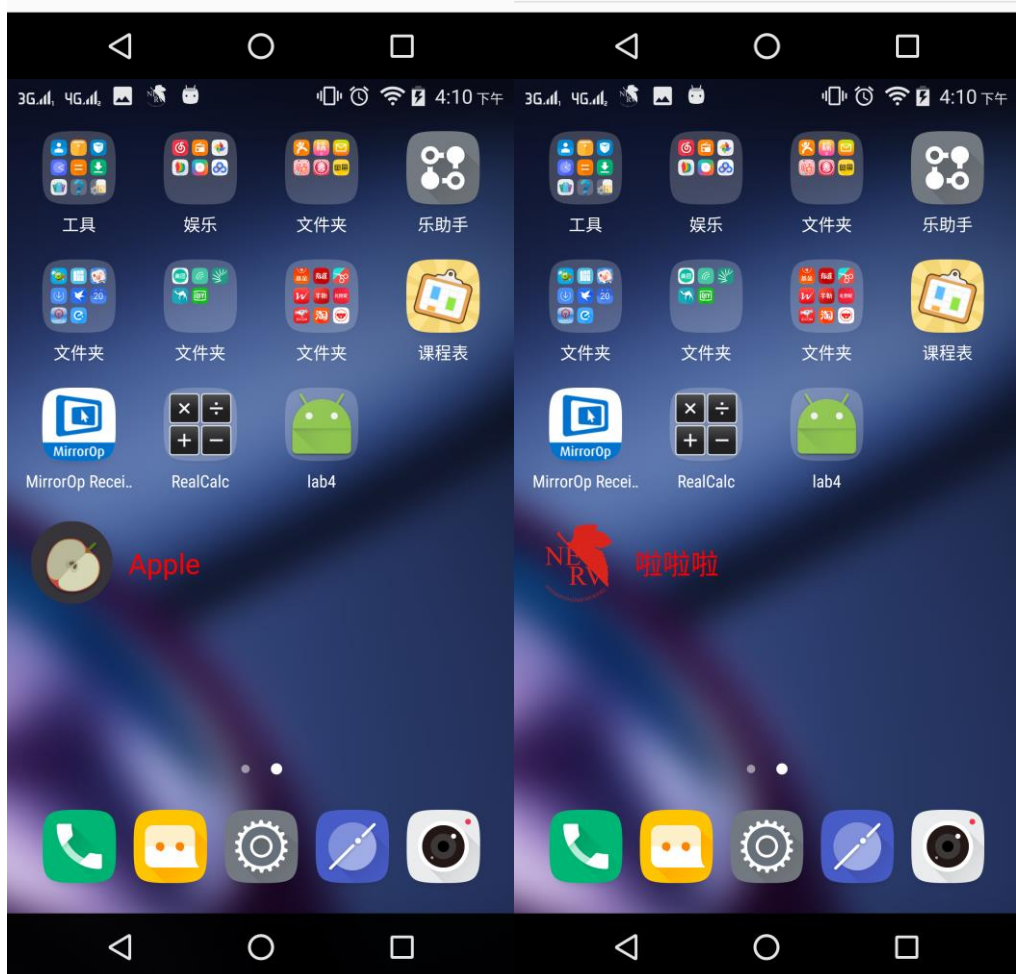
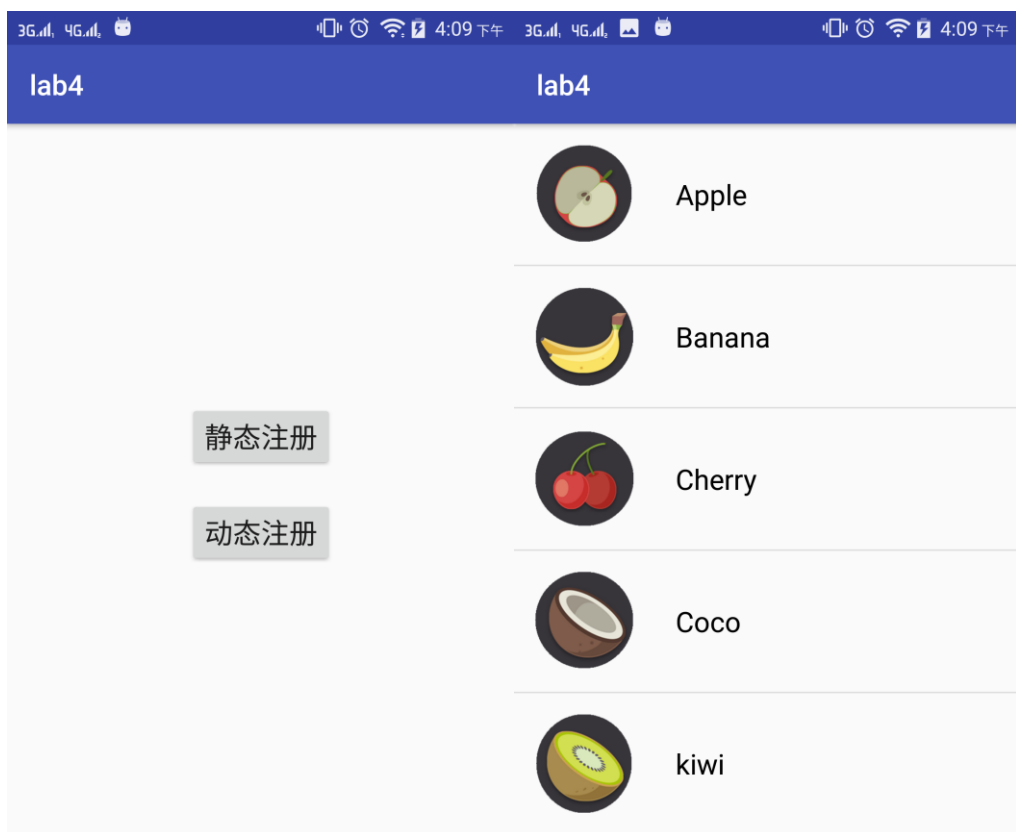
(未注册广播)

(已注册广播)

(已注册后点击发送)

### 三、 课堂实验结果

#### (1) 实验截图



## (2) 实验步骤以及关键代码

代码与之前相同的部分就不再赘叙，实验报告中的实验步骤和关键代码仅包含本次实验内容

### 1) 静态注册设置

首先新建一个 widget，一共包含有三个组件，分别是一个边框，一个框架和图形控件以及其他元素。在 AS 中创建好 Widget 类之后，会直接生成相关文件：两个 XML 文件和一个 JAVA 文件。

- 布局文件中

我们添加两个组件：一个 TextView 和一个 ImageView。分别用来设置 Widget 中的显示的内容，其 XML 文件编写如下：

```
<ImageView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/image"
    android:background="@mipmap/dynamic"
    android:layout_gravity="center"/>

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="20dp"
    android:textColor="#FF0000"
    android:layout_marginLeft="10dp"
    android:layout_gravity="center"
    android:id="@+id/widget_text"/>
```

- Widget 内容

Widget 内容提供者文件中，我们需要编辑该文件，设置其大小属性，初始化布局，自动更新时间间隔，是否可以调整大小以及可调整的方向。以及预览图片和 Widget 可以防止的位置。

```
<appwidget-provider xmlns:android="http://schemas.android.com/apk/res/android"
    android:initialKeyguardLayout="@layout/new_app_widget"
    android:initialLayout="@layout/new_app_widget"
    android:minHeight="55dp"
    android:minWidth="200dp"
    android:previewImage="@drawable/dynamic"
    android:resizeMode="horizontal|vertical"
    android:updatePeriodMillis="86400000"
    android:widgetCategory="home_screen">
```

- 修改 Widget 的 java 代码

重写 onUpdate 方法，为 Widget 添加事件，使其能返回主页面。这里需要使用到一种用户程序访问主屏幕和修改特定区域内容的方法：RemoteView 架构。RemoteView 架构允许用户程序更新主屏幕的 View，点击 Widget 激活点击事件，Android 会将其转发给用户程序，由 AppWidgetProviders 类处理，使得用户程序可更新主屏幕 Widget。

```
public void onUpdate(Context context, AppWidgetManager appWidgetManager, int[] appWidgetIds) {
    super.onUpdate(context, appWidgetManager, appWidgetIds);
    Intent clickInt = new Intent(context, MainActivity.class);
    PendingIntent pi = PendingIntent.getActivity(context, 0, clickInt, 0);
    RemoteViews rv = new RemoteViews(context.getPackageName(), R.layout.static_activity);
    rv.setOnClickPendingIntent(R.id.image, pi);
    appWidgetManager.updateAppWidget(appWidgetIds, rv);
}
```

这里，我们通过 PendingIntent 类来设置需要跳转的页面的 class，并将其添加到 RemoteView 架构下的 setOnClickPendingIntent 方法中绑定点击的模块 id。最后通过

appWidgetManager 来进行更新 Widget 的内容。其中 PendingIntent 是一种特殊的 Intent。主要的区别在于 Intent 的执行立刻的，而 PendingIntent 的执行不是立刻的。

```
rv.setTextViewText(R.id.widget_text, name);
rv.setImageResource(R.id.image, image);

AppWidgetManager appWidgetManager = AppWidgetManager.getInstance(context);
ComponentName me = new ComponentName(context, NewAppWidget.class);
appWidgetManager.updateAppWidget(me, rv);
```

在 onReceive 函数中，与上一次相同的读取传入的参数，然后通过调用 RemoteView 中的方法，来设置图像和文字，最后同样通过 AppWidgetManager 来进行对 Widget 内容的更改。

## 2) 动态注册设置

动态注册的思路基本与上面的一致，只需要在上次的代码中做出简要的更改，即在上一次的 Dynamic\_Broadcast 类中，重写的 onReceive 方法中，添加 RemoteView 以及 AppWidgetManager 即可，操作与静态的方式一致。

```
// Widget
RemoteViews rv = new RemoteViews(context.getPackageName(), R.layout.new_app_widget);
Intent clickInt = new Intent(context, MainActivity.class);
PendingIntent pi = PendingIntent.getActivity(context, 0, clickInt, 0);
rv.setOnClickPendingIntent(R.id.image, pi);
rv.setTextViewText(R.id.widget_text, name);
rv.setImageResource(R.id.image, image);

AppWidgetManager appWidgetManager = AppWidgetManager.getInstance(context);
ComponentName me = new ComponentName(context, NewAppWidget.class);
appWidgetManager.updateAppWidget(me, rv);
```

## (3) 实验遇到困难以及解决思路

本次的实验过程比较简单，基本按照老师和TA上课所讲的内容加以实现即可，没有碰到什么麻烦。

## 四、实验思考及感想

通过本次的实验课，了解到了 Widget 不是运行在自己进程里，而是宿主进程，所以交互需要处理 AppWidget 广播。AppWidgetProvider 只接收和这个 App Widget 相关的事件广播，比如这个 App Widget 被更新，删除，启用，以及禁用。每个 Widget 就是一个 BroadcastReceiver，它们用 XMLmetadata 来描述 Widget 细节。AppWidget framework 通过 BroadcastIntents 和 Widget 通信，Widget 更新使用 RemoteViews 来发送。RemoteViews 被包装成一个 layout 和特定的内容来显示到桌面上。通过编写代码进一步深入的了解到有关函数的运行方式。课后将继续学习安卓的有关知识，为下一次的实验做准备