

# Lab5 实验报告

13331233 孙中阳

## 参考资料

---

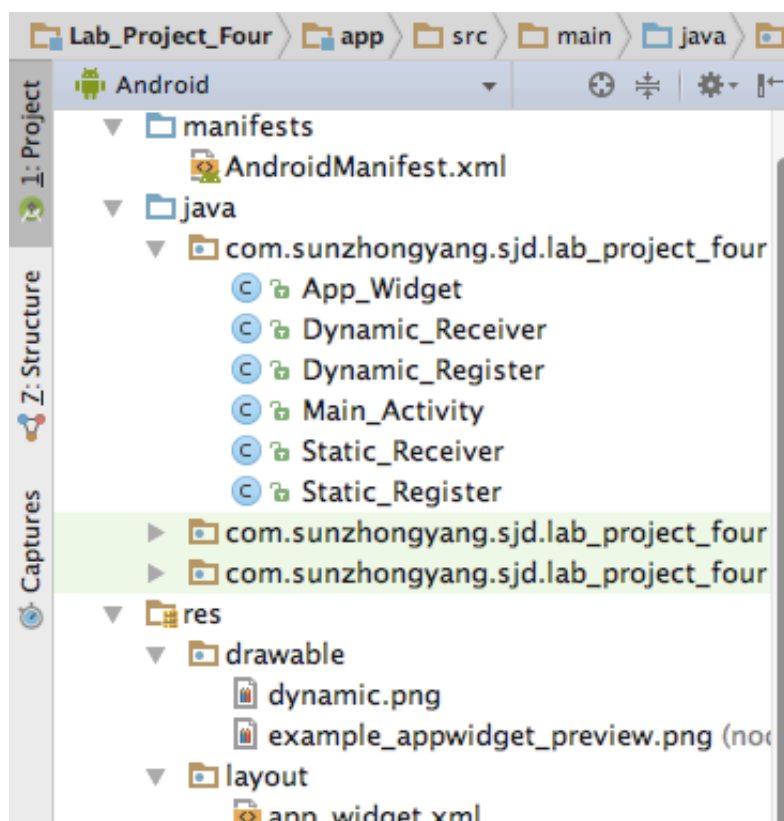
《第一行代码 Android》 郭霖 著

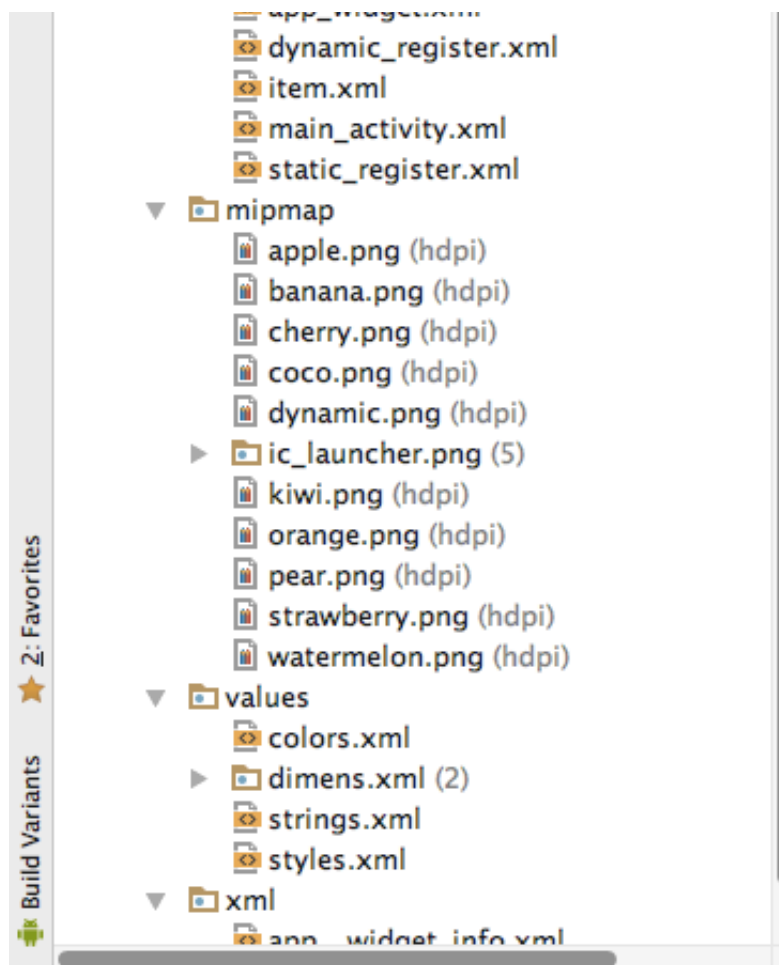
《Lab5实验文档》

## 实验步骤

---

本次试验较上一次实验难度有所降低，主要需要进一步熟悉广播的使用并通过动态和静态的广播实现 `Widget` 的动态响应，如更改图标和显示内容等  
图为项目结构



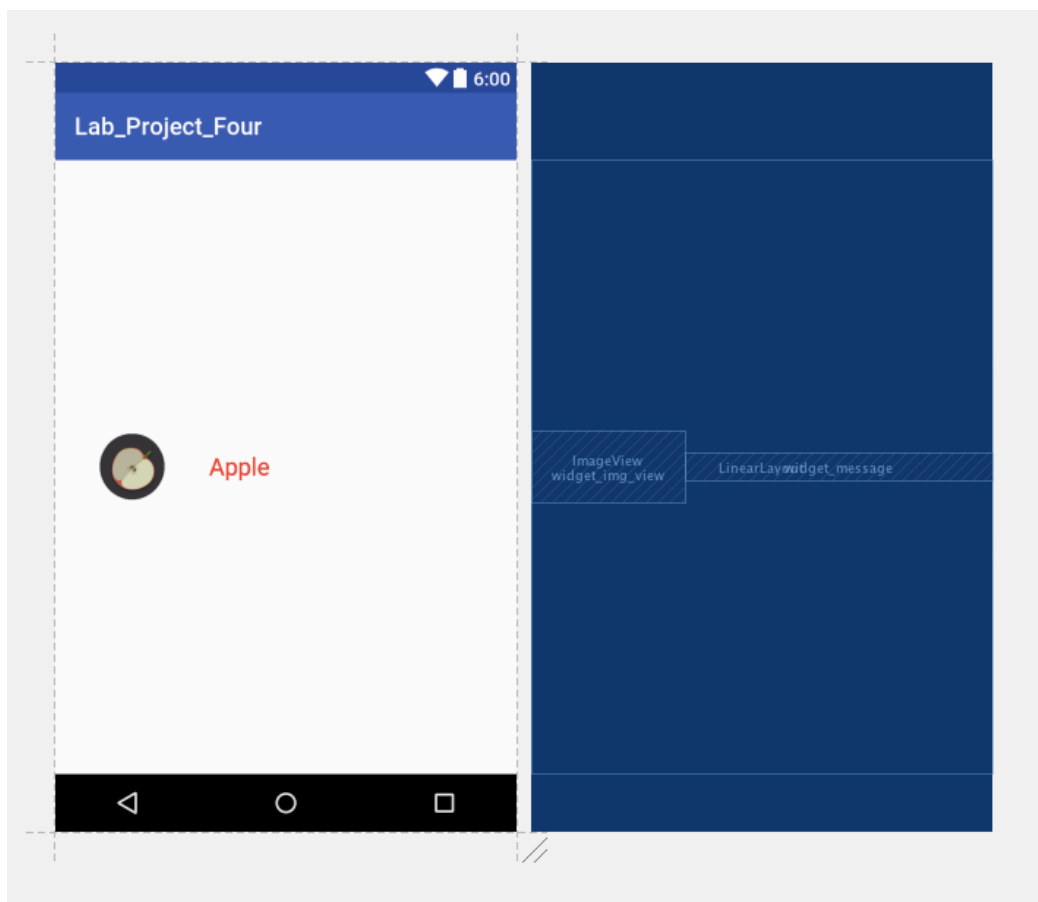


这次项目直接在上次项目的基础上完成，所以文件会多一些

从实验文档看，需要实现的目标主要有

- 1、掌握 `AppWidget` 编程基础
- 2、进一步熟悉 `Broadcast` 编程
- 3、掌握使用 `AppWidget` 动态注册 `Broadcast` 和静态注册 `Broadcast`

图为 `Widget` 布局



布局很简单，左侧是一个 `ImageView` 右侧是一个 `TextView`

接下来考虑实现

本次开发主要涉及三个方面：

1、点击 `Widget` 的图片能够跳转到程序的主界面

主要通过重写 `onUpdate()` 函数，在 `RemoteViews` 中为 `ImageView` 设置点击事件实现

```

@Override
public void onUpdate(Context context, AppWidgetManager appWidgetManager, int[] appWidgetIds)
{
    super.onUpdate(context, appWidgetManager, appWidgetIds);
    ;
    //一个跳转到主界面的Intent
    Intent clickInt = new Intent(context, Main_Activity.class);
    //设置一个PendingIntent
    PendingIntent pi = PendingIntent.getActivity(context, 0, clickInt, 0);

    //通过RemoteViews设置Widget中Image的点击事件
    RemoteViews rv = new RemoteViews(context.getPackageName(), R.layout.app_widget);
    rv.setOnClickPendingIntent(R.id.widget_img_view, pi);
    appWidgetManager.updateAppWidget(appWidgetIds, rv);
}

```

最后使用 `appWidgetManager.updateAppWidget()` 设置 `Widget`

## 2、响应动态广播

动态注册和静态注册中发送广播的部分和之前实验相同，不多赘述

在本次实验中对 `Widget` 的更新，主要通过重写 `onReceive()` 函数实现

这里由于 `Widget` 能够接受动态广播和静态广播，所以需要先做一个判断

```

//如果传入的intent是动态广播DYNAMICACTION
else if(intent.getAction().equals(DYNAMICACTION))
{
    //获取动态广播中包含的Widget文本信息
    Bundle dynamicBundle = intent.getExtras();
    String dynamicText = dynamicBundle.getString("Content");
    ;

    //根据获取到的文本信息设置Widget的外观
    rv.setTextViewText(R.id.widget_message, dynamicText);
    rv.setImageViewResource(R.id.widget_img_view, R.drawable.dynamic);
}

```

如果传入的是动态广播，则首先通过 `bundle` 获取广播中携带的信息，然后使用 `RemoteViews` 进行有关设置

```
//获取要更新的Widget的id并更新Widget
ComponentName name = new ComponentName(context.getPackageName(), App_Widget.class.getName());
int[] ids = appWidgetManager.getAppWidgetIds(name);
appWidgetManager.updateAppWidget(ids, rv);
```

最后同样使用 `appWidgetManager.updateAppWidget()` 更新 `Widget`

### 3、响应静态广播

首先需要在 `AndroidManifest.xml` 中添加对应 `Widget` 的 `Receiver` 并为其设置过滤器

```
<receiver android:name=".App_Widget">
    <intent-filter>
        <action android:name="android.appwidget.action.APPWIDGET_UPDATE" />
        <action android:name="com.sunzhongyang.sjd.lab_project_four.staticreceiver" />
    </intent-filter>

    <meta-data
        android:name="android.appwidget.provider"
        android:resource="@xml/app__widget_info" />
</receiver>
```

过滤器内容和之前试验中 `Static_Receiver` 的过滤器相同

然后，在重写的 `onReceive()` 函数中添加响应静态广播的部分

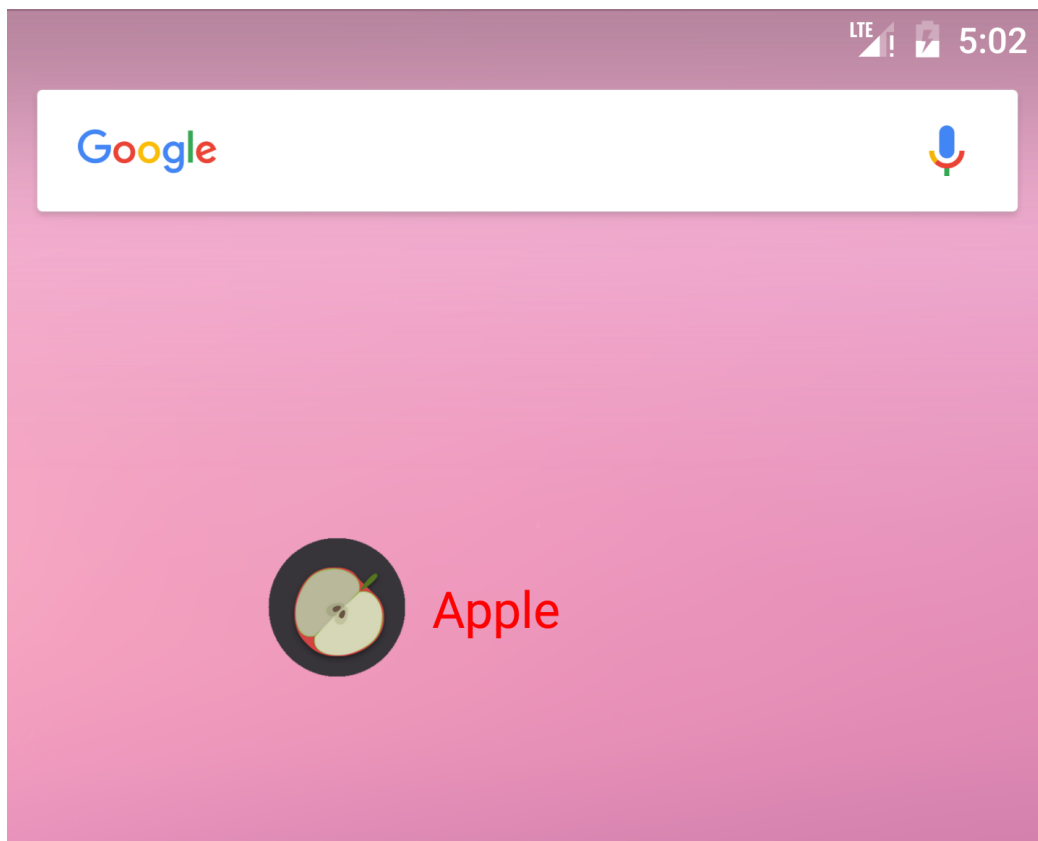
```
if(intent.getAction().equals(STATICACTION))
{
    //获取静态广播中包含的Widget图像信息和文本信息
    Bundle bundle = intent.getExtras();
    String text = bundle.getString("Content");
    int PictureId = bundle.getInt("PictureId");

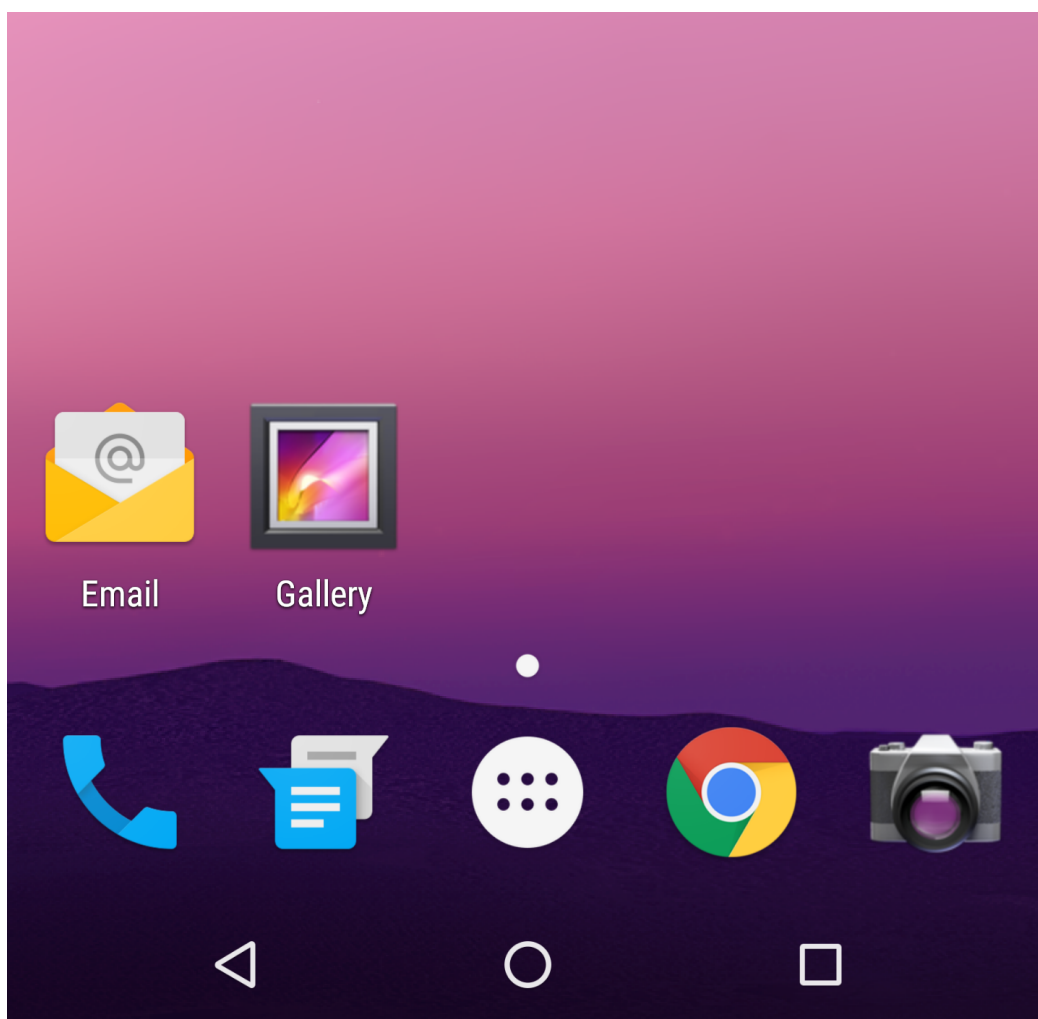
    //根据获取到的图像信息和文本信息设置Widget的外观
    rv.setTextViewText(R.id.widget_message, text);
    rv.setImageViewResource(R.id.widget_img_view, PictureId
);
}
```

获取的信息和动态广播有所不同，增加了用于更换 `Widget` 图片的 `PictureId`，其余对 `Widget` 的控制和更新部分没有区别

## 实验成果截图

`Widget` 外观



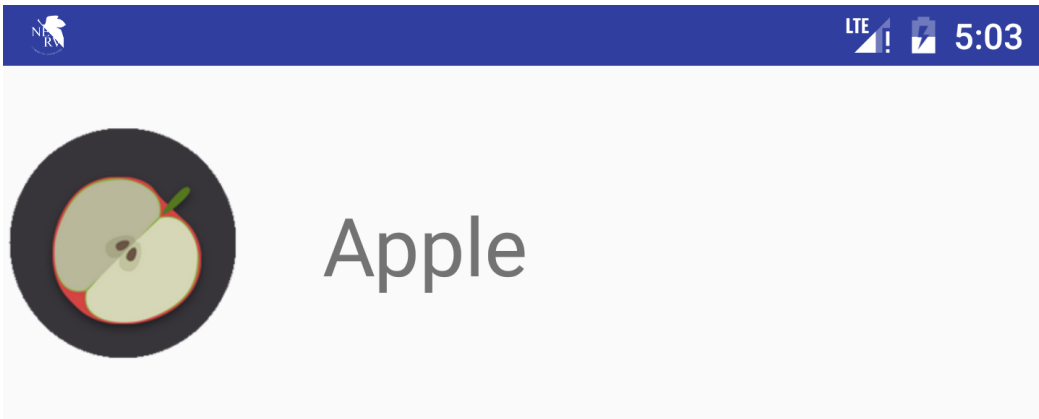


主页面



动态注册

静态广播页面







Banana



Cherry



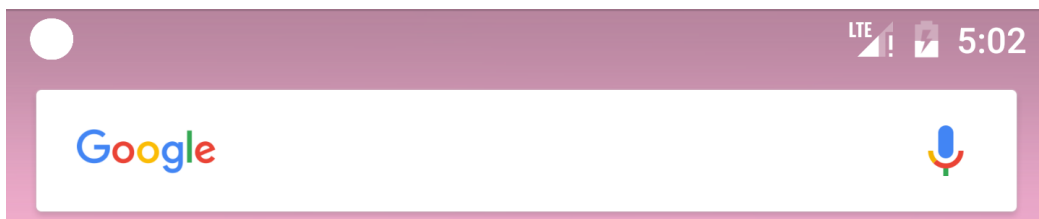
Coco

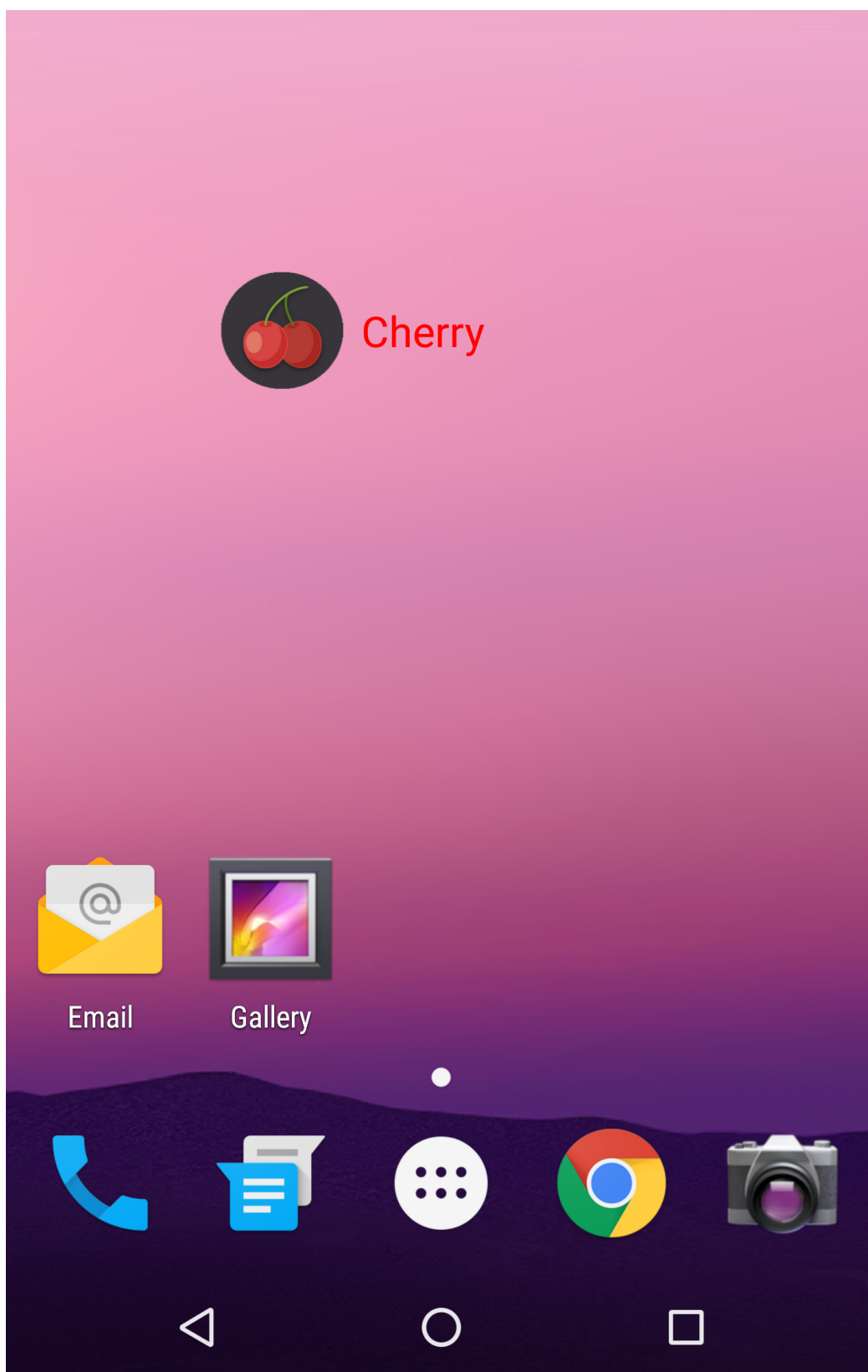


Kiwi

使用静态广播更改

Widget





动态广播页面

---



5:03

13331233

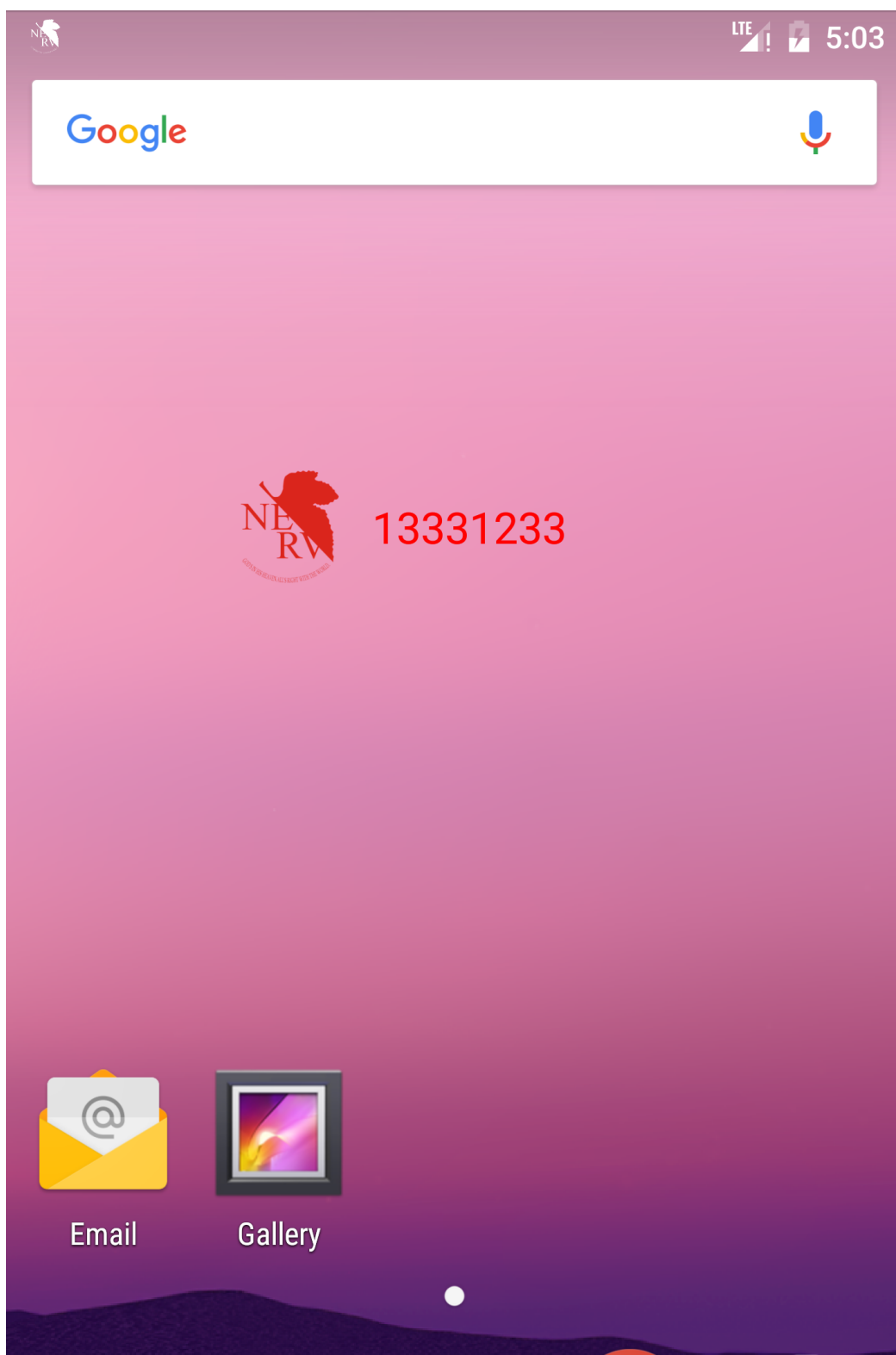
---

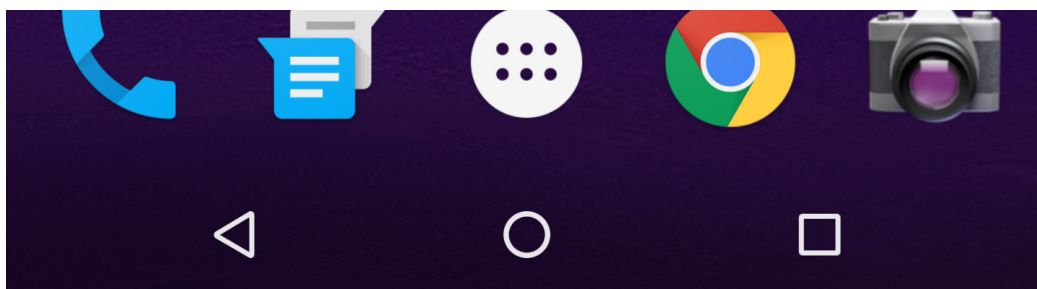
Unregister Broadcast

Send



使用动态广播更改 Widget





## 实验过程遇到的问题

---

这次只遇到一个问题

使用 `appWidgetManager.updateAppWidget()` 更新 `Widget` 时，首先需要传入一组 `Widget` 的 `id`，然后会根据传入的 `RemoteViews` 更新这些 `id` 对应的 `Widget`

```
appWidgetManager.updateAppWidget(ids, rv);
```

在前面重写的 `onUpdate()` 函数中，这些 `id` 通过参数 `appWidgetIds` 传入，但在 `onReceive()` 函数中则需要自行获取

通过查询文档，发现可以通过以下方式获取这些 `id`

```
ComponentName name = new ComponentName(context.getPackageName(), App_Widget.class.getName());  
int[] ids = appWidgetManager.getAppWidgetIds(name);
```

于是，问题解决

## 思考与总结

---

本次试验深入了解了Android开发的有关知识，尤其是和广播以及 `Widget` 有关的内容。这次实验的内容不多，大部分内容可以通过查阅PPT或作业说明得到解决，只有少部分需要百度。另外为了更有效率的完成实验（例如bug导致点击 `Widget` 没反应），这次使用了 `Log` 包中的函数 `Log()` 打印日志，在很大程度上加快了 debug 的进度。同时Android较UWP应用范围更

广，网上的资料也更多，问题也解决的比较顺利。总的来说比较有收获。