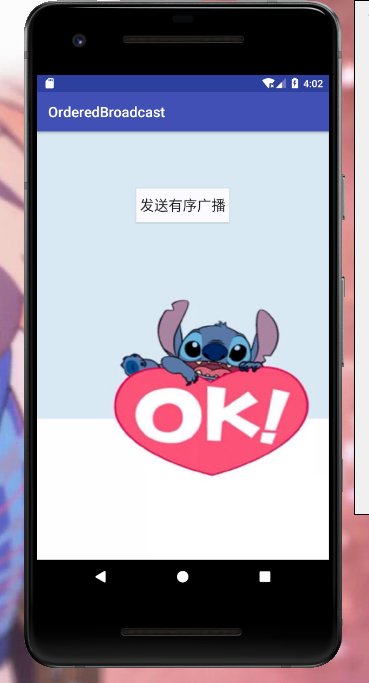
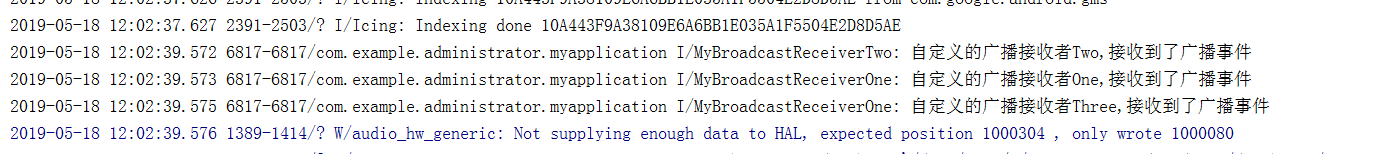
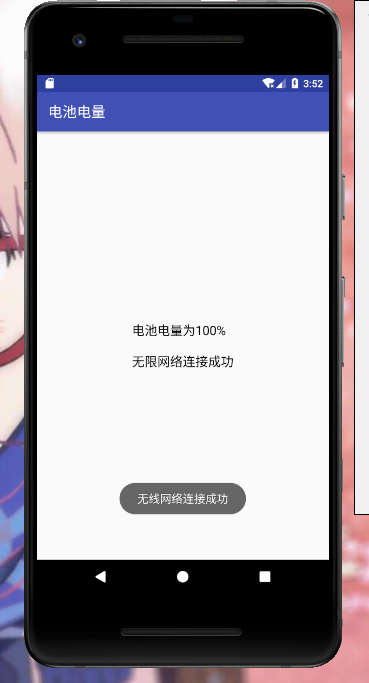
**Experiment 11**

**1.拦截史迪仔广播，p157**

** **

## 2.BatteryLevel（电池电量）



### 一、案例描述

1. 考核知识点

030006001：广播接收者简介

030006002：广播接收者的创建

1. 练习目标

* 广播的动态注册和销毁

1. 需求分析

广播分为两种注册方式：静态注册和动态注册，上一个案例讲解了静态注册，该案例将对广播进行动态注册与销毁。该案例主要讲解了监听设备的电量和网络状态并展示在界面上。

1. 设计思路（实现原理）
2. 在代码中进行监听电池电量和网络状态广播的注册和注销；
3. 将获取到的状态显示在界面上。

### 二、案例实现

**（1）编写主界面**

创建“BatteryLevel”程序，首先创建MainActivity用于显示信息，该程序的界面如图6-6所示。



1. 主界面

程序主界面对应的布局文件（activity\_main）如下所示：

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayoutxmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".MainActivity" >

<LinearLayout

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_centerInParent="true"

android:orientation="vertical" >

<TextView

android:id="@+id/tv\_cell"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="电池电量"

android:textColor="#000000"

android:layout\_margin="10dp"

android:textSize="18dp" />

<TextView

android:id="@+id/tv\_netWork"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="网络状态"

android:textColor="#000000"

android:layout\_margin="10dp"

android:textSize="18dp" />

</LinearLayout>

</RelativeLayout>

**（2）编写程序代码**

编写好主界面，接下来编写MainActivity类逻辑代码，具体代码如下所示：

packagecn.itcast.batterylevel;

importandroid.app.Activity;

importandroid.content.BroadcastReceiver;

importandroid.content.Context;

importandroid.content.Intent;

importandroid.content.IntentFilter;

importandroid.net.ConnectivityManager;

importandroid.net.NetworkInfo.State;

importandroid.os.Bundle;

importandroid.widget.TextView;

importandroid.widget.Toast;

public class MainActivity extends Activity {

privateTextViewtv\_cell, tv\_netWork;

privateBatteryReceiverbatteryReceiver;

privateConnectivityReceiver connectivity;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

tv\_cell = (TextView) findViewById(R.id.tv\_cell);

tv\_netWork = (TextView) findViewById(R.id.tv\_netWork);

// 注册广播接受者java代码

IntentFilterintentFilter = new IntentFilter(

Intent.ACTION\_BATTERY\_CHANGED);

batteryReceiver = new BatteryReceiver();

// 注册receiver

registerReceiver(batteryReceiver, intentFilter);

IntentFiltermFilter = new IntentFilter();

mFilter.addAction(ConnectivityManager.CONNECTIVITY\_ACTION);

connectivity = new ConnectivityReceiver();

registerReceiver(connectivity, mFilter);

}

/\*\*

\* 广播接受者

\*/

private class BatteryReceiver extends BroadcastReceiver {

@Override

public void onReceive(Context context, Intent intent) {

// TODO Auto-generated method stub

// 判断它是否是为电量变化的Broadcast Action

if (Intent.ACTION\_BATTERY\_CHANGED.equals(intent.getAction())) {

// 获取当前电量

int level = intent.getIntExtra("level", 0);

// 电量的总刻度

int scale = intent.getIntExtra("scale", 100);

// 把它转成百分比

tv\_cell.setText("电池电量为" + ((level \* 100) / scale) + "%");

}

}

}

private class ConnectivityReceiver extends BroadcastReceiver {

@Override

public void onReceive(Context context, Intent intent) {

State wifiState = null;

State mobileState = null;

ConnectivityManager cm = (ConnectivityManager) context

.getSystemService(Context.CONNECTIVITY\_SERVICE);

wifiState = cm.getNetworkInfo(ConnectivityManager.TYPE\_WIFI)

.getState();

mobileState = cm.getNetworkInfo(ConnectivityManager.TYPE\_MOBILE)

.getState();

if (wifiState != null &&mobileState != null

&&State.CONNECTED != wifiState

&&State.CONNECTED == mobileState) {

Toast.makeText(context, "手机网络连接成功",

Toast.LENGTH\_SHORT).show();

tv\_netWork.setText("手机网络连接成功");

// 手机网络连接成功

} else if (wifiState != null &&mobileState != null

&&State.CONNECTED != wifiState

&&State.CONNECTED != mobileState) {

// 手机没有任何的网络

Toast.makeText(context, "手机没有网络",

Toast.LENGTH\_SHORT).show();

tv\_netWork.setText("手机没有网络");

} else if (wifiState != null &&State.CONNECTED == wifiState) {

// 无线网络连接成功

Toast.makeText(context, "无线网络连接成功",

Toast.LENGTH\_SHORT).show();

tv\_netWork.setText("无限网络连接成功");

}

}

}

@Override

protected void onDestroy() {

super.onDestroy();

unregisterReceiver(batteryReceiver);

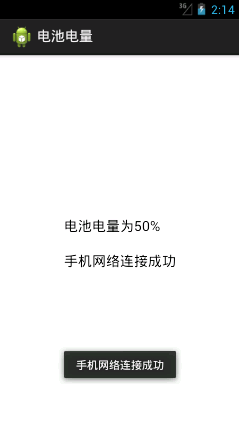
unregisterReceiver(connectivity);

}

}

**（3）运行程序**

运行“BatteryLevel”程序，结果如图6-7所示。



1. 运行界面

如图6-7所示，运行程序之后，当前设备的电量和网络状态都显示在了界面上。

### 三、案例总结

1、广播可以进行静态注册和动态注册。

2、广播在代码中动态注册需要有unregisterReceiver()操作。