#### 基于WiFi-Direct的Android遥控器

**目的**：移动终端(android智能手机或平板)做客户端遥控器，控制电视屏幕的操作。

**原理**：通过wifi-direct连接两台设备。Android4.0及更高版本已经全面支持wifi-direct技术，只要两台设备都支持Android wifi-direct，就可以进行直连。Wifi-direct允许两台设备之间不通过wifi热点就可以连接通信，比蓝牙跨越的距离更远，连接速度更快。其缺点是不成熟，不稳定。相对于蓝牙来说，更耗电。

android

phone

android

TV-BOX

WiFi-Direct

**采用技术**：Android, Java, xml, c/c++等。

**实现**：

1.连接通信

第一步，查找可用的设备。设备的wifi-direct功能必须都处于打开状态，如果处于关闭状态，则提示用户先去打开。原生的android4.0及以上版本中,wifi-direct都附加在“无线与网络”模块中，打开wifi即打开了wifi-direct，因此可以用以下方式判断是否可用：

WifiManager wm = (WifiManager) getSystemService(Context.*WIFI\_SERVICE*);

**if** (!wm.isWifiEnabled()) {

**new** AlertDialog.Builder(MainActivity.**this**)

.setTitle(R.string.*wifi\_direct\_invalid\_title*)

.setMessage(R.string.*wifi\_direct\_invalid\_msg*)

.setPositiveButton(R.string.*ok*, **new** DialogInterface.OnClickListener() {

@Override

**public** **void** onClick(DialogInterface dialog, **int** which) {

**return**;

}

}).show();

}

判断可用后，进行搜索。Google对wifi-direct提供了API，WifiP2pManager是常用的类，其中的discoverPeers方法用来搜索周边已打开wifi-direct的设备。

WifiP2pManager.ActionListener封装了onSuccess和onFailed方法来判断搜索成功和失败。

第二步，连接设备。用WifiP2pManager的requestPeers来进行连接请求。然后用connect方法进行连接。如果要断开连接，直接调用disConnect方法。

第三步，通信。采用Socket方式进行设备之间的数据传输。Android Socket通信直接采用Java中的socket通信，实现原理一样

2.事件响应(难点)

TV-BOX终端通过实现接口监听，处理Phone端传送过来的指令，指令的格式可以自定义。这些指令包括点击事件，屏幕滑动事件，拖拽事件等。

Commands from phone

Receive

Listener

Handle

Update ui

END

**Android phone端流程图**：

Wifi open

Phone

Open it

NO

YES

Search

No devices

waiting

Has devices

connect

failed

cancel

success

events

END