**说明文档**

**201800301177 曹铭哲**

先通过cmd命令打开控制台输入ipconfig/all获取IP地址，之后连接电脑热点。本项目采用两个端口，设为2000和2300，2000用于电脑通过socket向手机传送信息和手机接收信息，2300则相反。采用了大一学习到的Java swing技术，其中发送信息不需要使用线程，而接收信息需要。

电脑端接收信息：

public class Accept implements Runnable {  
 @Override  
 public void run() {  
 try{  
 ServerSocket server =new ServerSocket(2300);  
 while(true){  
 Socket socket=server.accept();  
 DataInputStream input=new DataInputStream(socket.getInputStream());  
 String strget=input.readUTF();  
 Server.*textarea*.append("手机："+strget+"\n");  
 }  
 }catch(IOException e){e.printStackTrace();}  
 }  
}

电脑端发送信息：

Socket s=new Socket("192.168.143.1",2000);  
DataOutputStream out=new DataOutputStream(s.getOutputStream());  
out.writeUTF(str);  
out.flush();

在安卓端增加访问互联网的权限

<uses-permission android:name="android.permission.INTERNET" />

在手机向电脑发送消息时出现异常

android.os.NetworkOnMainThreadException

一个APP如果在主线程中请求网络操作，将会抛出此异常。Android这个设计是为了防止网络请求时间过长而导致界面假死的情况发生。所以new一个Thread

手机端收送信息：

**class** Service **implements** Runnable{  
 **public void** run() {  
 **try**{  
 ServerSocket server=**new** ServerSocket(2000);  
 **while**(**true**){  
 Socket socket = server.accept();  
 BufferedReader recv = **new** BufferedReader(**new** InputStreamReader(socket.getInputStream()));  
 Message message = **new** Message();  
 Bundle b = **new** Bundle();  
 message.**obj**=recv.readLine();  
 message.setData(b);  
 MainActivity.**this**.**mHandler**.sendMessage(message);  
 }  
 } **catch** (IOException e) {  
 e.printStackTrace();  
 }  
 }  
}

**new** Thread(**new** Runnable(){  
 @Override  
 **public void** run() {  
 Socket socket = **null**;  
 **try** {  
 socket = **new** Socket(**"192.168.143.1"**,2300);  
 PrintWriter send = **new** PrintWriter(**new** BufferedWriter(**new** OutputStreamWriter(socket.getOutputStream(),**"utf-8"**)));  
 send.println(**str**);  
 send.flush();  
 AddText a=**new** AddText();  
 a.add(**"手机："**+**str**+**"\n"**,-65536);  
 **et**.setText(**""**);  
 } **catch** (IOException e) {  
 **et**.setText(**""**+e);  
 e.printStackTrace();  
 }  
 }  
}).start();

实现效果：



