

# 基于 TCP 的安卓与服务器交互开发

1

<sup>1</sup>(南京大学 计算机科学与技术系,江苏省南京市 210046)

## Data Communication Development between Android and Server via TCP

1

<sup>1</sup>(Department of Computer Science and Technology, Nanjing University, Nanjing City 210046, China)

+ Corresponding author: Phn +86-\*\*-\*\*\*\*-\*\*\*\*, Fax +86-\*\*-\*\*\*\*-\*\*\*\*, E-mail:vigorweijia@foxmail.com, http://\*\*\*\*

Received 2000-00-00; Accepted 2000-00-00

**Abstract:** Android OS have been applied to a large number of mobile devices and there is all kinds of development to Android and server. This passage search and develop a data communication software in Android OS basing on TCP protocol. Compass, expression evaluation and tele-operating drawing tablet are realized as slave functions.

**Key words:** TCP, Android Development, Server

**摘 要:** 安卓系统运用在大量的移动设备当中,而针对安卓与服务器的开发也五花八门。本文使用 TCP 协议传输数据,对安卓设备与服务器之间的交互方式进行了探究与开发。实现了指南针、表达式求值、远程画板几个子功能。

**关键词:** TCP、安卓开发、服务器

中图法分类号: \*\*\*\* 文献标识码: A

## 开发目的

移动设备与服务器的互联已经广泛的运用在了我们的生活中。本文拟通过无线连接并采用 TCP 传输协议进行移动设备与服务器的通讯。

## 1 实验设备

安卓手机一部: Google Pixel 2 XL, 安卓版本 9; 带热点功能的笔记本电脑一台, 充当服务器。

开发环境: Android Studio, JetBrains IDEA 以及抓包软件 WireShark

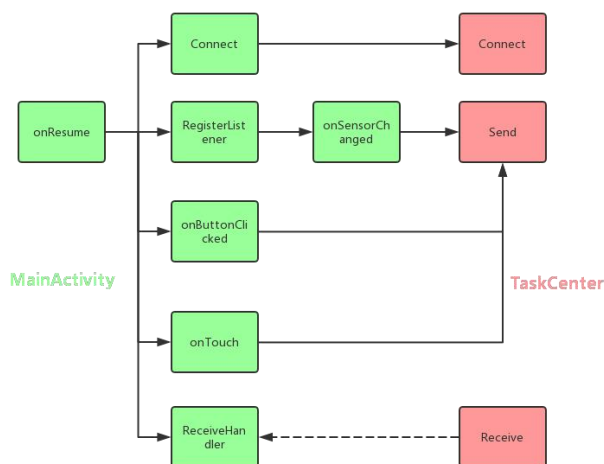
## 2 实验准备及环境配置

配置相应的 JAVA 环境、AS 的环境。在手机上打开开发者模式并允许 USB 调试。电脑打开热点 (本地连接 1), 手机连接对应的 WIFI, 确保二者处于同一网络中。

本次实验中, 服务器 IP 为 192.168.137.1。手机的 IP 随机分配。

### 3 设计思路

#### 手机客户端架构



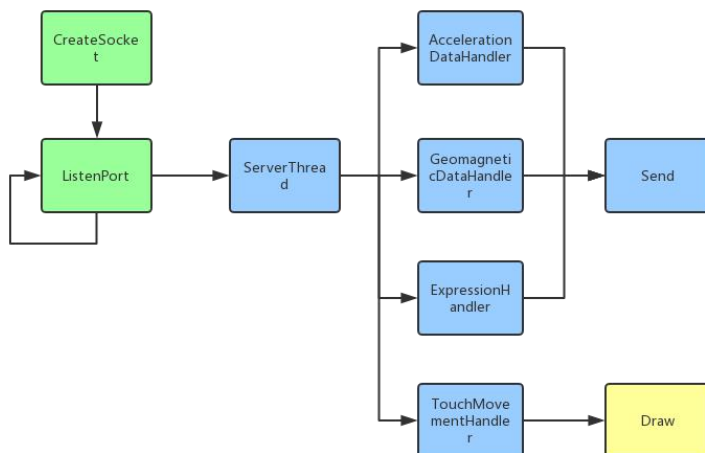
**MainActivity** 类显示各个功能模块，以及发出指令。

**TaskCenter** 是一个单例的类，包装了建立连接、发送以及接收的功能。在 TCP 连接的时候，会创建一个用于连接的线程。TCP 发送的时候，也需要建立一个线程，并在线程中加入接收数据的方法。

每当需要向服务器发送数据的时候，主线程会调用 **TaskCenter** 单例的 `send` 函数发送数据。每当收到数据的时候，**TaskCenter** 会通知主线程（通过回调函数）进行相应的处理。

在建立连接的时候，需要创建一个套接字，指定服务器 IP 以及端口号。

#### 3.1 电脑服务器端架构



服务器运行的时候，会创建套接字并监听相应的端口。为了提高运行的效率，当抓到包时，会创建一个新的线程，处理相应的数据。而主线程则继续监听端口。

其中，传回给客户端的指南针数据由加速度与地磁强度综合得到，表达式求值结果有求值函数计算并回传到客户端。而得到的触摸屏数据不回传，而是经过处理之后在本地绘图。

3.2 收发包的实现

创建了 TCP 套接字后，通过 `socket.getInputStream` 可以得到 TCP 报文的 `padding`（附加数据）内容。于是，使用一个 `BufferReader` 对象保存输入的内容，每次调用 `bufferreader.nextLine` 方法就可以得到收到的数据。但是，使用 `bufferreader.nextLine` 方法时，必须保证传输的报文以换行符结束，不然无法读到尽头。

使用一个 `PrintWriter` 或者一个 `BufferWriter` 对象保存 `socket.getOutputStream`，可以进行自动发包。每次在输出流中输出 `byte` 类型的数组，再调用 `flush` 方法即可发送 TCP 报文。

这样做可以收发包的原理我没有深究，按照我的理解，在创建套接字的时候，系统会为该套接字开辟一块输入和输出的缓冲区。当有 TCP 报文来时，会过滤掉包头，把包的数据内容放入输入缓冲区，而 `BufferReader` 的对象相当于一个指针，指向的内容由 `socket.getInputStream` 获得，每次能够从缓冲区中读一行。写入的 `byte` 数组会写到输出缓冲区中，而输出缓冲区的地址由 `socket.getOutputStream` 给出，调用 `flush` 方法表示清空缓冲区，也就是唤醒操作系统发包。



类型	符号	数据
加速度	ACCELEROMETER	三个浮点数，分别表示 xyz 轴的加速度
地磁强度	MAGNETIC	三个浮点数，分别表示 xyz 轴的磁场强度
触摸感应	TOUCH,DOWN	两个浮点数，表示触摸开始的坐标
	TOUCH,MOVE	两个浮点数，表示触摸的实时的坐标
	TOUCH,UP	两个浮点数，表示触摸结束时的坐标
表达式	EXPRESSION	一个字符串，表示一个表达式

由于不同的数据类型均通过 TCP 协议传输，所以还需要规定数据区域的格式：以逗号分隔，前半部分为数据类型，后半部分为数据内容。

4 功能实现

4.1 连接与接收功能的实现

对于客户端，连接功能需要新建一个线程。创建一个 `Socket` 对象，指定 IP 和端口，此时即自动连接。当服务器处于开启状态的时候，`socket.isConnected()`方法返回 `true`，表示成功连接。这个时候，分别用 `BufferReader` 和 `BufferWriter` 对象保存 `socket.getInputStream` 和 `socket.getOutputStream`。然后跳转到 `receive()` 方法，准备接受服务器传过来的内容。只要 `socket.isConnected()`返回值为真，就需要一直不停地获取收到的 TCP 数据。

当 `BufferWriter` 对象收到数据的时候（这个方法应该是阻塞的），调用回调函数(`callback`)，并传入报文，通知 `main` 中的相应的函数，对回调的内容做出及时的反应。

对于服务器，首先创建一个 `socket` 服务器对象 `ServerSocket`。然后在一个循环中，创建 `socket`，其值为 `serversocket.accept()`的返回值，表示接收套接字。然后开启新的线程获取收到的数据，根据类型选择相应的处理函数。对于指南针功能，需要计算转角，所以需要确保分别收到了线性加速度数据和地磁强度的数据。对于触摸屏的数据，需要在电脑的 `java` 图形框中绘制出同样的图形，所以得保证从收到 `TOUCH,DOWN` 开始的两份数据。处理完后，回传给客户端即可。

## 4.2 指南针功能的实现

得到加速度和地磁强度后,需要调用 `getRotationMatrix` 以及 `getOrientation` 两个方法计算转角。在客户端中,设置了一个设备监听器的频率 `SensorManager.SENSOR_DELAY_NORMAL`,每隔这样一段时间就要发送 TCP 报文。

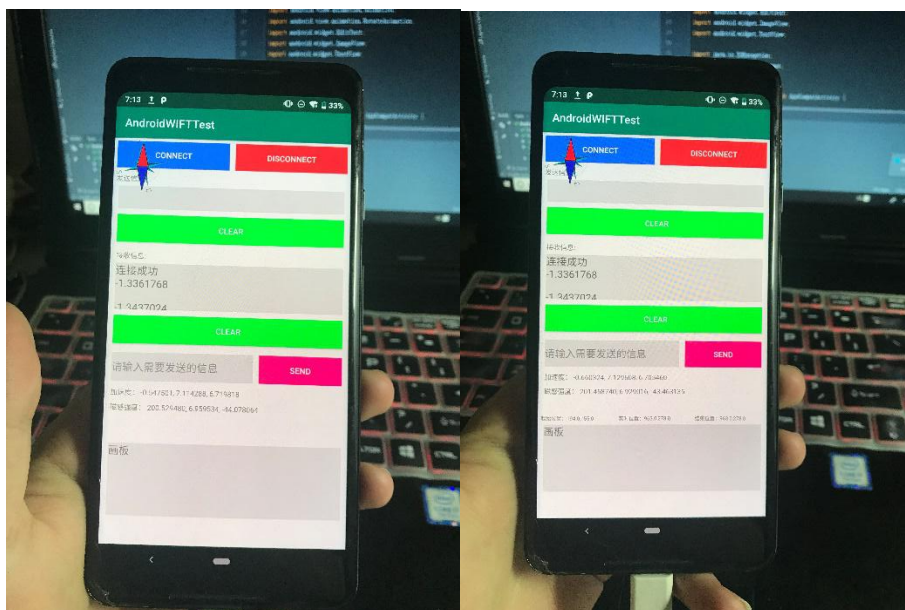
## 4.3 表达式求值功能的实现

中缀转后缀,具体实现略。客户端中表达式求值的报文通过点击按钮手动发出。

## 4.4 绘图功能的实现

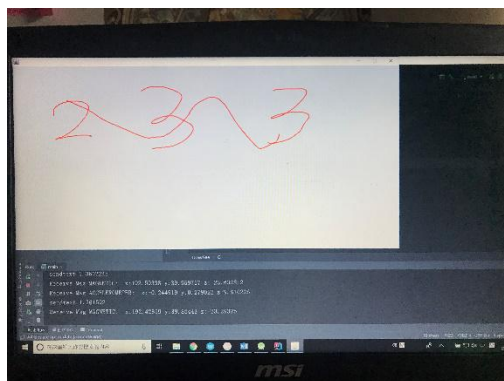
在 `JFrame` 框架下,使用 `Graphics2D` 对象进行绘图,连续触摸时,由于两个点之间的具体很短,所以使用直线工具进行绘图,使用上一个触点的坐标和现在触点的坐标。在客户端中,每当触摸到画板的时候,发送 TCP 报文。

## 5 使用手册及运行结果

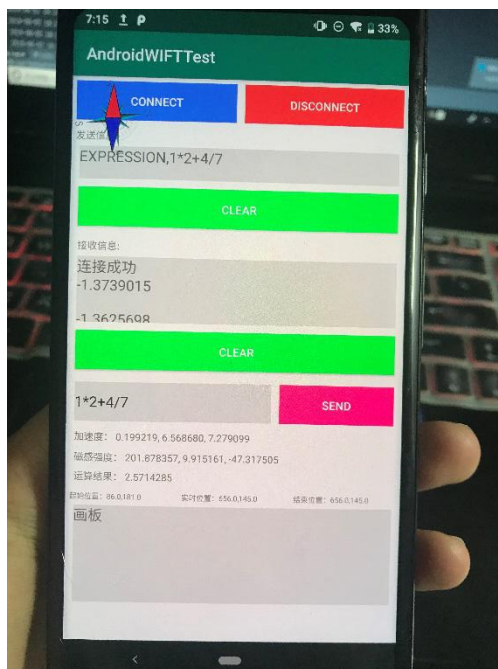


初始的时候,不会显示触点坐标,当在下方绘制图形的时候,会显示触摸点的坐标。中间接收栏中的内容为指南针需要转过的角度。指南针在屏幕的左上方。

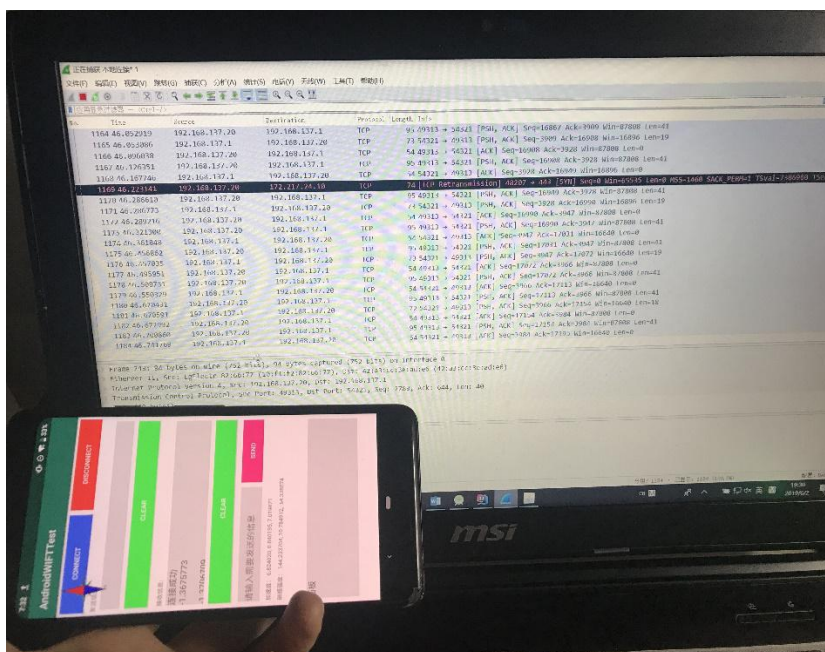
下面是服务器上绘制出来的图形



对于计算表达式的功能,需要在 send 按钮旁边的文本框中填入相应的表达式,并点击 send 传给服务器,服务器计算完毕后再传回客户端,在下面显示。



这些过程中,抓包结果如下:



可以从抓包结果中看到传输类型,收发方 IP 以及运行端口还有数据段内容。

## 6 总结

这个小 APP 的开发并未实现什么具体功能,而是简单揉合了几个功能,其目的主要是加深对客户

务器之间传输方式的理解。

由于是第一次写 Android 的 APP，难免会有很多 BUG 以及代码架构上的不足之处。

#### References:

- [1] Android TCP 客户端的实现 <https://www.jianshu.com/p/d8fe6e3fc00b>
- [2] 传感器之方向：使用加速度传感器和地磁传感器共同实现 [https://blog.csdn.net/An\\_nAI/article/details/78902572](https://blog.csdn.net/An_nAI/article/details/78902572)
- [3] android 中 getOrientation 是如何求得对应的航向角、俯仰角以及翻滚角的？ <https://www.zhihu.com/question/33565613>
- [4] SensorManager.getRotationMatrix 函数原理解释 <https://www.cnblogs.com/zuoxiaofei/p/4244238.html>
- [5] Android 获取点击屏幕的位置坐标 [https://blog.csdn.net/carter\\_yu/article/details/50499040](https://blog.csdn.net/carter_yu/article/details/50499040)