# Android触摸屏校准程序设计手册

一 概要

免驱的单点红外触摸屏在Android环境下已经能够很稳定的正常工作，其不支持触摸屏校准。但是，现实生活中充满了未知的状况，设备在某种情况下很可能会出现触摸坐标不准的情况。于是让触摸屏支持校准就很有必要了。本校准程序是根据设备特性、驱动特性和Android系统本身的特点而设计的校准程序。

其特点：

支持屏幕翻转：本校准程序用到了线性代数中的矩阵行列式展开求解方式，及根据程序从设备获取到的点坐标创建二元一次方程组，并以线性展开的方式进行求解获取校准参数。此方法不依赖于触摸屏是否翻转，只要计算出正确的校准参数均可正确校准。算法详细介绍请参考第三章第3节“校准算法”。

校准精确：程序在校准的时候会从触摸屏的四个角分别获取5个点，并分别求取平均值来确定相应的触摸坐标。获取的触摸坐标偏差较小，所以比较精确。

效果可持续：程序在计算得到校准参数后会将其保存到Android系统下的/etc/目录。设备驱动每次启动时，驱动都会到/etc目录下读取校准参数。当触摸屏在工作时，驱动会将获取到的坐标点并以校准参数按照相应的公式计算后才上报给系统。只要不是人为的删除所存校准参数，此参数将一直有效。

错误处理健全：在程序的错误处理功能中，我为每种预知的错误都提供了错误原因提示。比如：当触摸屏未插入Android设备或是驱动未安装就进行校准时，程序会提示未发现触摸屏设备来提示用户校准无法进行。

二 总体设计

* 1. 功能结构

程序的功能结构图如图2-1：

由功能图可知，程序分为两大功能：触摸屏校准、恢复出厂设置。在触摸屏校准功能下，如果用户操作不当导致触摸屏点明显错误，则可以选择重新校准之后再保存校准参数。所谓的恢复出厂设置是删除以前的校准参数，让触摸屏设备按照默认的模式进行工作，及不再按照校准公式计算驱动所获取的坐标值直接上报给系统。

* 1. 总体结构

程序的模块图如图2-2：

由模块图可知，程序主要分为三大模块：控制模块、交互接口模块、通信接口模块。

控制模块：其主要作为控制器，控制程序的人机交互接口和通信接口。所有的人机交互事件处理、界面更新、与驱动的通信和数据计算处理都由此模块来控制完成。它就像人的大脑控制着人体所有的肢体动作。

交互接口模块：它是人机交互动作的接收处理者，也是程序界面更新的执行者。包括程序的主菜单、校准界面的每次更新和触摸事件的处理都由此模块完成。校准过程中，用户点击指定的屏幕位置，此模块会接收触摸事件并获取坐标值，之后会动态显示当前区域获取的点的个数、定时时间等。获取到的坐标值将会反馈给控制模块进行校准参数的计算。

通信接口模块：此模块主要负责校准程序与驱动程序的通信。校准过程是校准程序通过命令的方式来控制驱动来完成的（具体命令字详情，请查看3.4“数据结构”一节）。

* 1. 接口设计

校准程序的接口有两大部分：通信接口、交互接口。

**通信接口**

程序将与驱动程序通信的部分使用NDK封装为库接口跟驱动程序进行通信。具体的NDK开发方式请参考文档“Android应用及驱动开发介绍“。

接口名称为”CalibTools”，接口函数如表2-1：

表2-1 通信接口函数列表

|  |  |
| --- | --- |
| 函数 | 说明 |
| calibrate(int w, int h, int[] flags) | 此函数为校准函数，第一个参数w是Android设备显示屏的宽度，第二个参数h是Android设备显示屏的高度，第三个参数是用来存储校准参数的缓存。在此函数中程序会获取每个触摸区域的5个点的平均值并。在最后一个触摸区域的点获取完毕后，会计算出校准参数并存到flags缓存中返回。 |
| isup() | 判断触摸是否抬起，用于判断是否进入下一区域点的收集。 |
| isDeviceExist() | 判断设备是否已连接。 |
| deviceReset() | 重置触摸屏设备。 |
| save() | 保存校准参数到Android设备文件系统。 |

校准程序创建了一个线程专门用于调用通信接口与驱动程序进行通信，来完成校准参数的计算。

**交互接口**

将界面的显示和更新及触摸事件的处理分离为一个模块会让程序结构更清晰，同时也让程序更具扩展性。

接口名称“RenderView”，它是Calibration类的内部类，继承了View类。在此接口中完成了所有界面的初始化、界面的更新及通知通信接口进行校准通信等工作。其中创建名叫 “runableUi”的线程来完成校准界面的刷新。接口函数如表2-2：

表2-2 交互接口函数列表

|  |  |
| --- | --- |
| 函数 | 说明 |
| onDraw() | RenderView类重载了View类中的onDraw函数来自定义画校准界面，由函数invalidate()来触发系统自动调用此函数（不能直接调用onDraw函数）。 |
| startCalibrate() | 由此函数来通知通信接口进行通信工作获取校准坐标并计算校准参数。对界面的刷新也在此函数中进行（调用invalidate()函数，触发onDraw()来刷新界面）。 |

三 详细设计

* 1. 程序流程

校准程序的流程图如图3-1：

由于程序逻辑比较简单，对于程序工作流程此图展现的比较清楚。

* 1. 程序类

校准程序的类关系图如图3-2：

由此图可知，校准程序包含5个类。其中CalibrationTools、RenderView、CalibTools分别为控制模块、交互接口、通信接口。Applications与ViewTools分别为界面资源回收工具和错误提示工具。在此关系中，以控制模块为主导来完成整个校准过程。

* 1. 校准算法

算法原理：程序获取触摸屏四角的任意三个区域的坐标值，以此坐标值和理想相应的理想坐标值来组成一个二元一次方程组。求解此方程组可获取x轴的校准参数Ax、Bx、Cx，y轴的校准参数Ay、By、Cy。有了这六个校准参数，在触摸屏工作时，驱动就可以按照下面的公式将从设备得到的坐标值校准后上报给系统。

x = Ax\*x0 + Bx\*y0 + C

y = Ay\*x0 + By\*y0 + C

具体的校准参数计算过程请参考文档”Android应用及驱动开发介绍”中的2.2.3一节。

四 程序错误处理

在触摸屏校准过程中可能会遇到各种未知的错误。比如：设备未插入Android设备就运行校准程序进行校准，或是在校准过程中意外拔出触摸屏设备，或是触摸屏设备驱动未安装好就进行校准等等。这些情况都会导致校准失败。为了用户能够清楚的了解各种错误的原因，此校准程序设计了专门的错误信息提示功能。比如：当触摸屏设备驱动未安装就进行校准时，程序会提示用户无法进行校准驱动未发现等信息。

在以后的现实应用中错误是未知，程序不可能将所有的错误都考虑到。所以校准程序的功能和错误处理有待在以后的开发中不断进行完善。