**本科毕业设计(论文)**

**FINALPROJECT/THESISOFUNDERGRADUATE**

**(2016届)**

**上海理工大学本科毕业设计(论文)**

**RD4800恒温扩增荧光检测仪软件**

**RD4800 isothermal amplification fluorescence detector software**

|  |  |
| --- | --- |
| **学　　院** | 光电信息与计算机工程学院 |
| **专　　业** | 计算机科学与技术 |
| **学生姓名** | 姚先胜 |
| **学　　号** | 1212471611 |
| **指导教师** | 赵逢禹 |
| **完成日期** | 2016年06月 |

# 承诺书

本人郑重承诺：所呈交的毕业论文“RD4800恒温扩增荧光检测仪软件”是在导师的指导下，严格按照学校和学院的有关规定由本人独立完成。文中所引用的观点和参考资料均已标注并加以注释。论文研究过程中不存在抄袭他人研究成果和伪造相关数据等行为。如若出现任何侵犯他人知识产权等问题，本人愿意承担相关法律责任。

承诺人(签名)：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

日期： 年 月 日

# 第1章 系统概述

## 1.1软件开发的目的和意义

随着android平台的发展与应用场景越来越广，不单单是手机，平板电脑，手表等电子设备终端开始应用android系统，许多的工业设备也开始使用基于android操作系统深度定制的硬件操作系统。本软件的设计，就是为基于android系统的RD4800恒温扩增荧光检测仪开发一个配套的实验记录和绘图软件，预期达到的功能是通过采集该恒温扩增荧光检测仪的48个孔板，将使用者在实验过程中采集到的实时数据以曲线的方式显示出来，实现扩增过程的自动检测，并自动判断实验的结果，同时将实验结果以表格(.xls)的方式显示并保存。

首先我们需要了解的是该检测仪设备的基本情况，RD4800恒温扩增荧光检测仪全名是RD4800恒温高通量双波长荧光读数仪，通过使用PCR技术，对特定靶DNA进行快速扩增的一种仪器。PCR又称无细胞分子克隆系统或特异性DNA序列体外引物定向酶促扩增法，是聚合酶链反应(polymerase chain reaction)的缩写，是基因扩增技术的一次重大革新。可将极微量的靶DNA特异地扩增上百万倍，从而大大提高对DNA分子的分析和检测能力，能检测单分子DNA或对每10万个细胞中仅含1个靶DNA分子的样品，因而此方法在分子生物学、微生物学、医学及遗传学等多领域广泛应用。

由于该仪器采集得到的是基于荧光扫描的原始信息，实验人员没办法直观的判断实验成果以及得出结论，开发一套可视的，自动判断实验成果的应用软件就是急需的解决方案。

## 1.2现阶段国内同行发展现状

RD4800恒温扩增荧光检测仪软件是专为RD4800恒温扩增荧光检测仪开发的仪器配套专用软件，用来将实验结果以图表的形式显示，目前国内有类似的实时荧光恒温扩增检测仪，然而这些设备的配套应用软件或是生产商独占，或是因为采集数据格式限制，无法适用RD4800设备，不具备竞争力。因此，开发一套适配RD4800设备的应用软件是必须的，不但能提高设备的应用性，实验结果的准确性，同时也能大大提高RD4800在同类设备中的竞争力，增加在市场的占有率，提高销售效益。

## 1.3课题概述

（1）易用性：本软件基于Android系统开发，可移植平台广泛，由于面向国内市场，全中文界面更易使用，用户可以轻易上手。

（2）实验数据实时显示与实验结果高效计算：对于RD4800检测仪采集到的扩增或者溶解数据，本软件可以以曲线的方式实时显示数据的变化规律；对于扩增实验的阴阳性结果，可以在实验完成或停止后立即给出实验结果。

（3）实验结果保存的完备性：在实验开始前，用户可以手动建立文件用来存储接下来的实验结果数据，如果用户没有手动新建文件，本软件也会自动生成保存结果的表格文件；同时，在没有关闭本软件或者运行新的实验之前，用户可以将实验结果另存为自定义的表格文件；另外，对于有外置存储卡的设备，本软件支持将实验结果数据文件导出到外置储存卡。

# 第2章 相关开发技术

## 2.1 Android操作系统

Android是由Google成立的Open Handset Alliance(OHA，开放手持设备联盟) 持续领导与开发，主要设计用于触屏移动设备如智能手机和平板电脑的一个基于Linux内核的开放移动操作系统。

在Android应用程序开发中，通常通过在Android SDK（Android软件开发包）中使用Java作为编程语言来开发应用程序。开发者亦可以通过在Android NDK(Android Native开发包）中使用C语言或者C++语言来作为编程语言开发应用程序。

Android操作系统的优点：

（1）Android系统市场占有率高，根据维基百科数据，在2013年5月，Android 在中国的占有率有71.5%，在世界范围的占有率亦接近70%，而高市场占有率意味着大量的使用者以及开发者，进一步来说就是降低了软件使用的难度以及开发的难度。

（2）软件管理方便，易于安装和卸载，Android提供了独特的PackageManager服务，不管是从网络下载软件还是从Sdcard安装软件，都可以直接安装而不需要依赖其他服务或者设置其他属性就可以使用了；如果想要删除一个软件，也可以直接使用Android自带的软件管理器来卸载。

（3）软件资源丰富，由于市场的高占有率带来的大量开发者，Android平台上有着非常丰富的应用程序，可以满足各种使用场景。

Android操作系统的缺点：

（1）Android的版本迭代太快，导致应用需要同时适配多个版本，开发成本会有相应的提高。

（2）Android系统自身权限管理存有缺陷，由于Android操作系统的开放和自由性，一些恶意程序和病毒会损害使用者利益。

## 2.2 Java程序开发语言

Java是一种由Sun公司推出的开放性的，拥有跨平台、面向对象、泛型编程等特性的计算机编程语言。本软件基于Android操作系统，使用Java语言开发。

Java开发优点：

（1）Java是开源的。

（2）编程风格与C语言十分接近，使用简单，易于阅读。

（3）跨平台性，由于运行在Java虚拟机中，一次编写，到处运行。

（4）拥有垃圾回收机制，无引用的对象会被自动删除以避免出现内存溢出的问题。

Java开发缺点：

（1）开发大型系统的复杂性，不同的框架增加了学习成本。

（2）垃圾回收机制对系统整体性能造成影响。

## 2.3 Xml标记语言

Xml的全称是可扩展标记语言，用来传送及携带数据信息，在Android应用的开发中，包括软件的界面布局、样式定义、权限申请、配置文件等都是通过使用Xml文件来定义和设计。

# 第3章 系统可行性研究

## 3.1技术可行性

Android应用的开发，使用的主要是是Java进行逻辑处理以及算法分析，Xml资源文件来设计界面样式。我作为一个计算机专业的学生，对于Java和Xml两点都是相当熟悉的，进一步需要学习的是Android应用开发中独有的控件的使用方法以及Android系统对应用权限限制下的功能是否可以实现。结合软件需求说明书中对于具体功能的要求，包括导出表格，绘制曲线，展示结果等功能都是可以在使用Java语言完成的，并且Android系统也是支持包括文件操作，曲线绘制的操作，从技术角度来分析是可行的。

## 3.2经济可行性

对于该软件的开发，需要用到PC和配有Android系统的硬件设备，以手机为模拟对象。对于学习成本，在学校图书馆中有大量的相关书籍可以参考，同时由于Android的开源特性，网上也有很多学习资料，因此会花费时间来执行开发任务，经济上的额外开销到没有多少，作为毕业设计的课题以及考核项目在经济上是可行的。

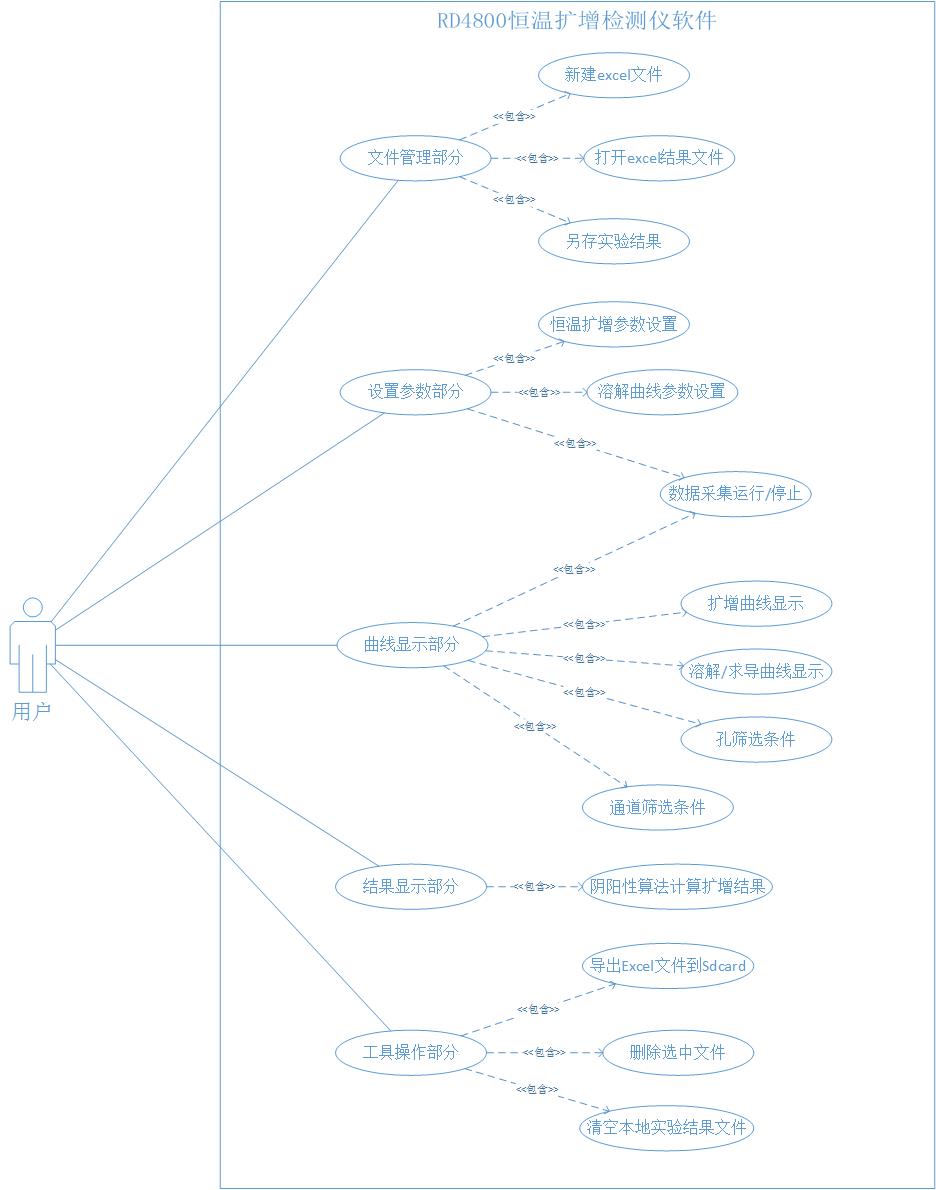
## 3.3操作可行性

本软件是基于Android系统的客户端应用，暂不涉及服务器内容，所有功能的实现与使用都是通过界面化操作，另外本软件的适用人群是进行相关实验的研究人员，不存在不理解相关名词的情况，加上界面端的直观显示，所以在操作方面是可行的。

# **第4章 系统需求分析**

## 4.1功能需求分析

### 4.1.1 UML用例图



### 4.1.2 用例表

|  |
| --- |
| 用例名称：新建表格文件 |
| 用例标识号：11 |
| 描述：用户通过新建按钮新建空白结果excel文件在应用指定文件夹中 |
| 参与者：用户 |
| 前置条件：软件安装成功，并且拥有访问和读写存储设备权限 |
| 基本事件流：   1. 用户已打开软件，点击新建按钮 2. 用户输入自定义文件名并进行非法字符检测   A1：非法字符检测不通过   1. 点击确定创建一个自定义excel文件在应用默认文件夹中 2. 用例结束 |
| 其他事件流：  A1：非法字符检测不通过  （1）出现提示文字  （2）无法点击确定按钮 |
| 后置条件：采集的扩增实验数据计算的实验结果会保存在该空白文件中 |

|  |
| --- |
| 用例名称：打开表格文件 |
| 用例标识号：12 |
| 描述：用户点击打开按钮后选择需要打开的文件，点击后自动打开并将内容显示在结果页面 |
| 参与者：用户 |
| 前置条件：  （1）软件安装成功，并且拥有访问和读写存储设备权限  （2）储存设备中存在表格(.xls)文件 |
| 基本事件流:   1. 用户已打开软件，进入文件界面，点击打开按钮 2. 弹出文件夹及表格文件列表 3. 点击表格文件(.xls) 4. 弹出框消失 5. 用例结束 |
| 其他事件流：   1. 点击弹出列表中的文件夹 2. 进入相应的文件夹内，列出所有的表格文件(.xls) 3. 点击表格文件 4. 弹出框消失 |
| 后置条件：表格文件内容显示在结果页面 |

|  |
| --- |
| 用例名称：另存为表格文件 |
| 用例标识号：13 |
| 描述：用户将实验结果另存为自定义的表格文件 |
| 参与者：用户 |
| 前置条件：  （1）软件安装成功，并且拥有访问和读写存储设备权限  （2）在软件内存中存在实验结果数据 |
| 基本事件流:   1. 点击另存按钮 2. 弹出文件夹及文件列表包括另存文件模块 3. 选择你需要保存的文件夹路径 4. 输入文件名名并进行非法字符检测   A1：非法字符检测不通过   1. 点击确定按钮保存文件 2. 弹出框消失 3. 用例结束 |
| 其他事件流：  A1：非法字符检测不通过  （1）出现提示文字  （2）无法点击确定按钮 |
| 后置条件： 无 |

|  |
| --- |
| 用例名称：恒温扩增参数设置 |
| 用例标识号：21 |
| 描述：用户点击选框以及输入恒温扩增的设置参数 |
| 参与者：用户 |
| 前置条件：软件安装成功 |
| 基本事件流:   1. 打开软件，进入设置界面 2. 点击单选框选择HEX通道 3. 选择默认恒温温度 4. 输入恒温时间并检测输入内容 5. 用例结束 |
| 其他事件流：   1. 曲线选中HEX通道 2. 取消默认恒温时间，输入恒温时间并检测输入内容 3. 用例结束 |
| 后置条件：  （1）选中HEX通道，则在曲线页面可以选择HEX通道筛选  （2）恒温时间决定采集到的数据点数量 |

|  |
| --- |
| 用例名称：溶解曲线参数设置 |
| 用例标识号：22 |
| 描述： 用户点击溶解曲线选框并设置溶解曲线参数 |
| 参与者：用户 |
| 前置条件：软件安装成功 |
| 基本事件流:   1. 打开软件，进入设置界面 2. 点击选中溶解曲线选框 3. 设置起始温度默认 4. 选中结束温度和度数误差默认 5. 用例结束 |
| 其他事件流：   1. 取消选中结束温度选框，输入参数并检测输入内容 2. 取消选中默认度数误差选款，输入参数并检测输入内容 3. 用例结束 |
| 后置条件：  （1）选中溶解曲线选框，则曲线界面溶解曲线按钮可以选择  （1）选中HEX选框，则曲线界面可以使用HEX通道筛选  （2）溶解曲线的开始结束温度差除以度数误差决定溶解曲线采集数据点的数量 |

|  |
| --- |
| 用例名称：数据采集程序运行与停止 |
| 用例标识号：23 |
| 描述： 用户点击运行或停止按钮以决定采集数据显示曲线或者停止采集数据显示结果 |
| 参与者：用户 |
| 前置条件：  （1）软件安装成功  （2）恒温扩增和溶解曲线参数设置完成 |
| 基本事件流:   1. 点击运行按钮 2. 检测参数设置是否正确   A1：参数输入有问题   1. 设置运行按钮不可点击，停止按钮可点击 2. 用例结束 |
| 其他事件流：  A1：参数输入有问题  （1）有问题的输入框获得焦点  （2）运行按钮不做改变，仍可点击  （3）用例结束 |
| 后置条件：  （1）曲线页面运行和停止按钮和设置页面运行停止按钮同步  （2）曲线页面曲线实时显示 |

|  |
| --- |
| 用例名称：扩增曲线显示 |
| 用例标识号：31 |
| 描述： 曲线界面显示采集到的实时数据构成的扩增曲线 |
| 参与者：用户 |
| 前置条件：  （1）软件安装成功  （2）数据采集程序运行中 |
| 基本事件流:   1. 从内存中获得采集到的数据 2. 对X轴数据进行处理，以小时为单位画点 3. 绘制图像 4. 用例结束 |
| 其他事件流：无 |
| 后置条件：  （1）根据选择不同通道来筛选显示不同曲线  （2）根据选择不同的孔显示不同的曲线 |

|  |
| --- |
| 用例名称：溶解/求导曲线显示 |
| 用例标识号：32 |
| 描述： 曲线界面显示采集到的实时数据构成的溶解曲线以及对应的求导曲线 |
| 参与者：用户 |
| 前置条件：  （1）软件安装成功  （2）数据采集程序运行中  （3）设置界面选中显示溶解曲线 |
| 基本事件流:   1. 从内存中获得采集到的数据 2. 对X轴数据进行处理，起点从起始温度开始画点 3. 计算两点之间的求导值 4. 绘制溶解曲线和求导曲线图像 5. 用例结束 |
| 其他事件流：无 |
| 后置条件：  （1）根据选择不同通道来筛选显示不同曲线  （2）根据选择不同的孔显示不同的曲线 |

|  |
| --- |
| 用例名称：孔筛选条件 |
| 用例标识号：33 |
| 描述： 选择一孔或多个孔显示不同的曲线 |
| 参与者：用户 |
| 前置条件：  （1）软件安装成功  （2）数据采集程序运行中 |
| 基本事件流:   1. 点击某一个孔，取消选中 2. 曲线图中隐藏该孔的曲线 3. 实例结束 |
| 其他事件流：   1. 点击取消选中的孔，该孔重新选中 2. 曲线图显示该孔曲线 3. 实例结束 |
| 后置条件： 无 |

|  |
| --- |
| 用例名称：通道筛选条件 |
| 用例标识号：34 |
| 描述： 选择FAM或这HEX通道决定显示的曲线类型 |
| 参与者：用户 |
| 前置条件：  （1）软件安装成功  （2）数据采集程序运行中  （3）设置界面选择采集HEX通道数据才可以选择显示HEX通道曲线 |
| 基本事件流:   1. 选中HEX通道 2. 曲线图中同时显示该通道的曲线 3. 实例结束 |
| 其他事件流：   1. 点击取消选中FAM通道 2. 曲线图不显示任意通道内容，为空 3. 实例结束 |
| 后置条件： 无 |

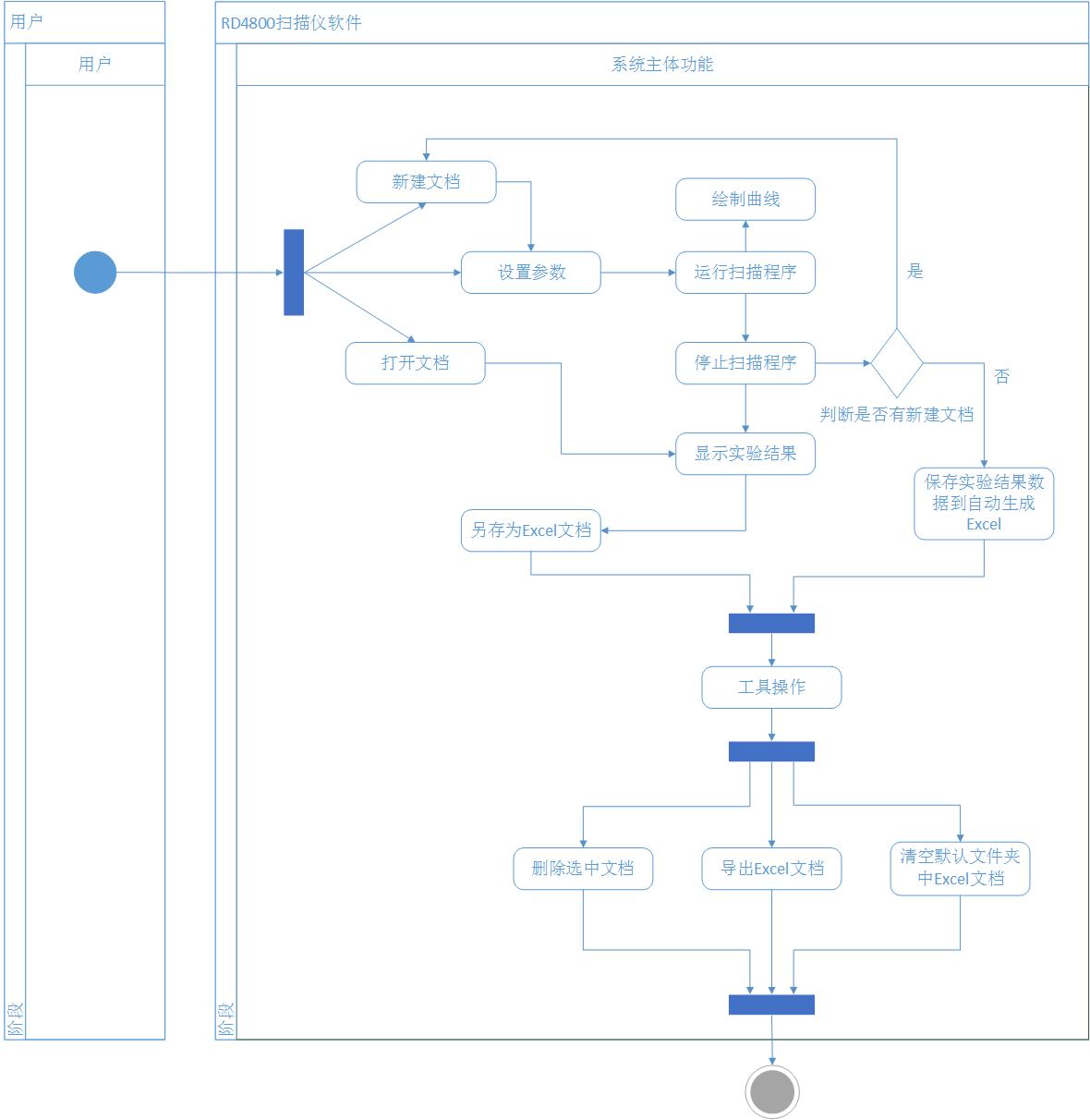
|  |
| --- |
| 用例名称：阴阳性算法计算扩增结果 |
| 用例标识号：41 |
| 描述： 计算采集到的FAM通道的恒温扩增曲线的阴阳性结果 |
| 参与者：用户 |
| 前置条件：  （1）软件安装成功  （2）数据采集程序运行中  （3）停止按钮被选中 |
| 基本事件流:   1. 通过采集到的数据点计算实验的阴阳性 2. 显示实验结果列表 |
| 其他事件流：无 |
| 后置条件：  （1）如果有新建表格文件，将实验结果保存在新建文件中  （2）没有新建表格文件，将实验结果保存到自定义名称表格文件中 |

|  |
| --- |
| 用例名称：导出Excel文件到Sdcard |
| 用例标识号：51 |
| 描述： 将应用文件夹内的excel结果数据文件导出到Sdcard指定目录 |
| 参与者：用户 |
| 前置条件：  （1）软件安装成功，并且拥有访问和读写存储设备权限  （2）储存设备中存在表格(.xls)文件  （3）有可移除的Sdcard挂载在Android系统中 |
| 基本事件流:   1. 打开软件，进入工具界面 2. 点击选中列表显示的表格文件 3. 点击导出按钮 4. 提示导出成功 5. 刷新Sdcard的文件列表 6. 用例结束 |
| 其他事件流：无 |
| 后置条件： 无 |

|  |
| --- |
| 用例名称：删除选中的结果文件 |
| 用例标识号：52 |
| 描述： 将选中的应用文件夹内的excel结果数据文件删除 |
| 参与者：用户 |
| 前置条件：  （1）软件安装成功，并且拥有访问和读写存储设备权限  （2）储存设备中存在表格(.xls)文件 |
| 基本事件流:   1. 打开软件，进入工具界面 2. 点击选中列表显示的表格文件 3. 点击删除按钮 4. 刷新本地应用内的文件列表 5. 用例结束 |
| 其他事件流：无 |
| 后置条件： 无 |

|  |
| --- |
| 用例名称：清空本地实验结果文件 |
| 用例标识号：53 |
| 描述： 应用默认文件夹内的excel结果数据文件全部删除 |
| 参与者：用户 |
| 前置条件：  （1）软件安装成功，并且拥有访问和读写存储设备权限  （2）储存设备中存在表格(.xls)文件 |
| 基本事件流:   1. 打开软件，进入工具界面 2. 点击清空按钮 3. 刷新本地应用内的文件列表 4. 用例结束 |
| 其他事件流：无 |
| 后置条件： 无 |

### 4.1.3 UML活动图



## 4.2数据需求分析

### 4.2.1 数据精度

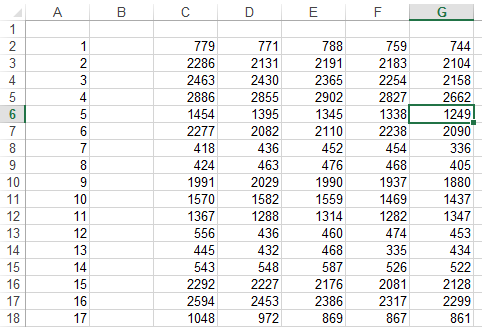
（1）对于设置页面输入的温度值，精确到小数点后1位数。

（2）对于曲线页面的X轴坐标值，由于扩增曲线是以小时为单位，在分钟转换为小时值时精确到小数点后两位。

（3）在结果页面计算阴阳性结果的dt值时，同样是确保小数点后两位值。

### 4.2.2 源数据文档格式分析

由于本软件是基于读取数据文档的方式，通过解析文档内容，得到类似于RD4800通过FAM或者HEX通道采集到的数据点，因此这里列出所需要的文档的格式。从Excel中读取数据源信息，每一行表示采集到的一个孔的所有数据，另外每个Excel包括两个工作表，分别为FAM和HEX，需要注意的是，数据表第一行空白，第二列空白。如图所示：



注：这里需要的Excel文档只能是Microsoft Excel 97-2003格式。

### 4.2.2 源数据存储目录分析

因为本质上是在从数据源文件中采集数据，因此这里规定应用需要使用的一些文件目录。(这里所有目录都是在应用安装过程中自动生成，前提是用户授予应用相应权限)

|  |  |  |  |
| --- | --- | --- | --- |
| 目录名 | 路径 | 用途 | 备注 |
| RD4800 | 内部存储根目录 | 存放应用需要的所有文件 | 需要获得Android读写Sdcard权限 |
| amp\_data | RD4800/ | 存放扩增实验采集的数据文件(.xls) | 只能放一个.xls文件 |
| dissolution\_data | RD4800/ | 存放溶解实验采集的数据文件(.xls) | 只能放一个.xls文件 |
| amplification | RD4800/ | 存放实验结果文件(.xls) |  |
| files | 外置Sdcard中  /Android/data/yxs.usst.edu.cn.project/ | 存放导出到Sdcard的实验结果文件(.xls) | 需要获得Android读写Sdcard权限 |

# **第5章 系统设计**

## 5.1系统总体设计

### 5.1.1 功能结构设计

本软件从功能上分成五个部分，结合需求阶段用例图的分析，分别是文件、设置、曲线、结果以及工具这五个部分，软件总体的结构图如下所示：

RD4800检测仪软件

曲线

设置

文件

工具

结果

1. 文件部分结构构成如图所示：

文件部分

另存实验结果为Excel

打开Excel文档

新建Excel文档

（1）新建Excel文档：用户自定义一个名称合法的Excel文档保存在应用根目录中的指定文件夹中。

（2）打开Excel文档：用户点击打开按钮后，出现.xls格式的文档列表，选择自己想要查看的某一个文档，点击查看内容。（这里文件位置是可选择的，不单单是根目录的限制，其他目录的.xls文件也是可行的）

（3）另存实验结果为Excel：在内存中有实验结果数据的时候，用户可以使用该功能，将实验结果另存为用户自定义的合法的.xls文件，同样，这里的保存位置也是可选的，默认是应用根目录的指定文件夹中。

1. 设置部分结构构成如图所示：

设置部分

溶解曲线参数设置

恒温扩增参数设置

运行绘图程序

停止绘图程序

（1）恒温参数设置：用户可以点击选择是否使用HEX通道采集数据，默认使用FAM通道采集数据；用户可以选择恒温温度，默认是42℃，取消默认，可以手动输入，限制最大输入值为99.9℃；用户可以选择输入恒温时间，默认60分钟，限制最大输入值为999分钟。

（2）溶解曲线参数设置：用户点击溶解曲线选择框，可以选择绘制溶解曲线以及曲线对应的求导曲线，默认不绘制溶解曲线；当溶解曲线没有选中时，溶解曲线的其他参数都无法获得焦点并且编辑；起始温度和结束温度的值默认最大也是99.9℃，度数误差是扫描仪采集点数的间隔温度，默认不超过10℃。

（3）运行绘图程序：用户点击运行按钮，数据采集程序会按照设置的参数来实时采集数据，同时将采集到的数据点绘制在曲线界面上；运行按钮此时不可点击以避免多次点击事件，停止按钮可以点击以终止数据采集系统的运行。

（4）停止绘图程序：用户点击停止按钮，程序停止实时数据采集以及实时绘图；运行按钮已可点击，停止按钮无法获得焦点，不可点击，以避免重复多次点击事件；使用阴阳性算法计算扩增实验的实验结果；如果用户有在实验开始前新建Excel文档，则将实验结果保存在该新建空白文档中，否则将数据结果保存在自动生成的Excel文档中。

1. 曲线部分结构构成如图所示：

曲线显示部分

通道筛选功能

孔筛选功能

停止绘图程序

运行绘图程序

求导曲线显示

溶解曲线显示

扩增曲线显示

（1）运行绘图程序：功能和上节运行功能一致，参考上节运行绘图程序部分。

（2）停止绘图程序：功能和上节停止功能一致，参考上节停止绘图程序部分。

（3）扩增曲线显示：在实时数据采集系统运行过程中，由于采集时间默认是1分钟采集一次，而曲线X轴参数默认时间单位是小时，故需要将采集到的数据点转换成（小时， 扩增倍数）这种类型的数据以绘制曲线；根据孔板的分布情况，将不同颜色的孔板对应不同颜色的曲线；由于采集的数据点之间间隔较大，需要多形成的曲线做平滑处理。

（4）溶解曲线显示：在实时数据采集系统运行过程中，由于溶解曲线采集点是以温度为度量值来采集数据，所以X轴参数默认单位是℃，因此需要将采集到的数据点转换为（温度， 扩增倍数）这种类型的数据以绘制曲线；采集的总点数由温度差和设定的温度度量值的比值来决定；根据孔板的分布情况，将不同颜色的孔板对应不同颜色的曲线；曲线同样需要做平滑处理。

（5）求导曲线显示：根据采集到的溶解曲线的实时数据，计算出每个点的导数值，和溶解曲线同步绘制；根据孔板分布情况，不同求导曲线拥有不同颜色；曲线做平滑处理。

（6）孔筛选功能：默认同时绘制48个孔的所有曲线；选择或取消选择不同的孔，则对应孔的曲线显示或者隐藏；选中和未选中的孔应有不同的标识以区分。

（7）通道筛选功能：默认只绘制FAM通道的曲线；根据设置页面参数设置，判定HEX通道是否可以选择；根据选择的通道结果决定显示的曲线类型。

1. 结果部分结构构成如图所示：

结果显示部分

阴阳性算法计算扩增结果

将扩增结果保存到.xls文档中

（1）阴阳性算法：根据采集到的数据点，通过编写阴阳性算法，判断该实验的结果是阴性还是阳性；如果是阴性的曲线，直接给出dt值，如果是阳性的曲线，取最大dt值作为结果的dt值。

（2）保存扩增实验结果：如果用户有新建Excel文档，则将实验结果写入到用户新建的Excel文档中，否则将实验结果写入默认生成的不重名文档中。

1. 工具部分结构构成如图所示：

工具操作部分

导出Excel文档到Sdcard

清空文档

删除文档

（1）导出Excel文档：显示两个列表，分别是应用根目录指定文件夹中的实验结果数据文档以及Sdcard中的指定文件夹中的文档，将选中的本地.xls文档导出到Sdcard中。

（2）删除文档：将选中的本地.xls文档删除。

（3）清空文档：将本地应用根目录的指定文件夹内的所有实验结果记录文件删除。

### 5.1.2 操作界面设计

本软件的界面设计描述以时序图来表示：

1. 新建文档：

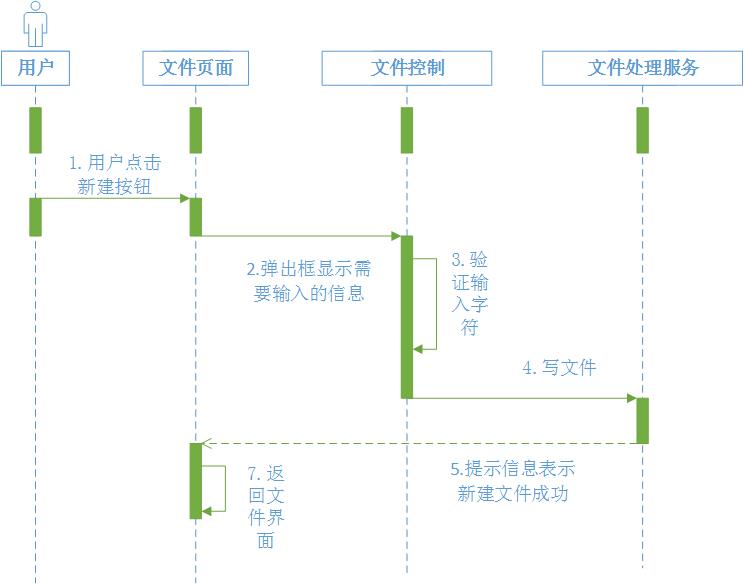


图1 新建文件功能时序图

1. 打开文档：

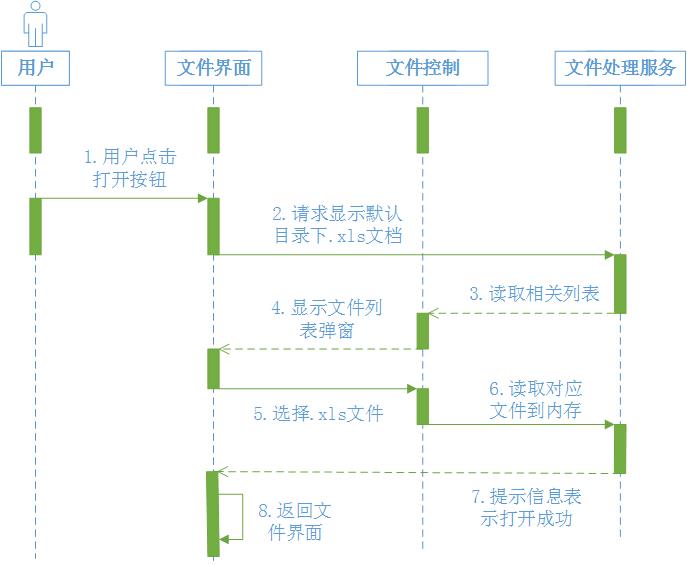


图2 打开文件功能时序图

1. 另存文档：

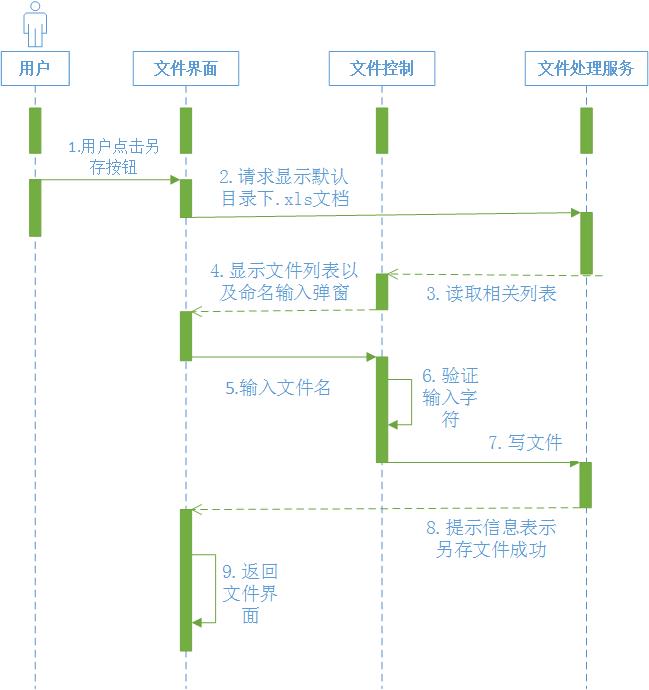


图3 另存文件功能时序图

1. 运行绘图程序

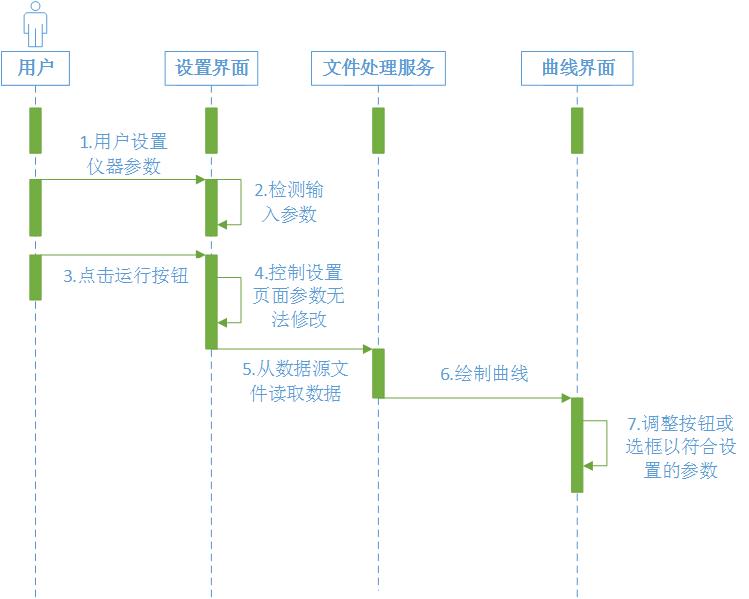


图4 运行绘图功能时序图

1. 停止绘图程序

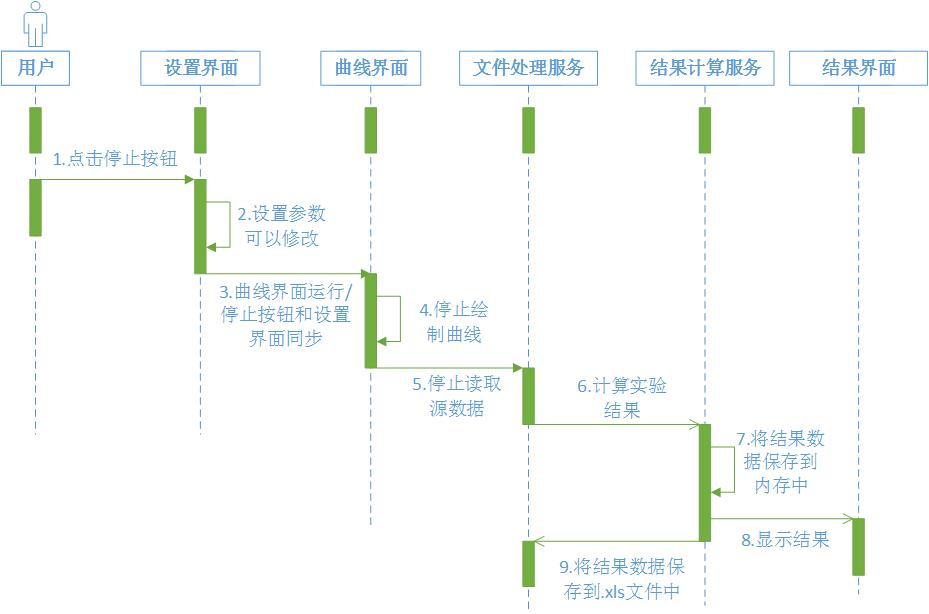


图5 停止绘图功能时序图

1. 筛选显示曲线

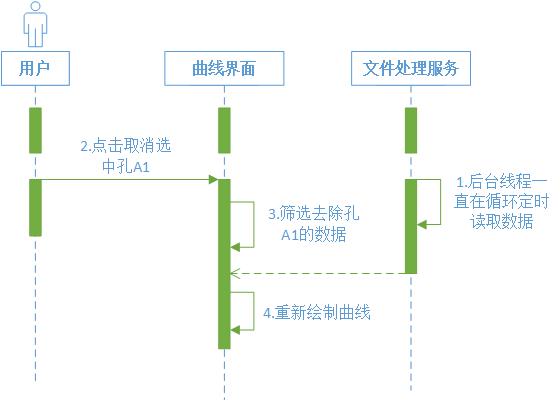


图6 筛选功能时序图

1. 导出/删除文档

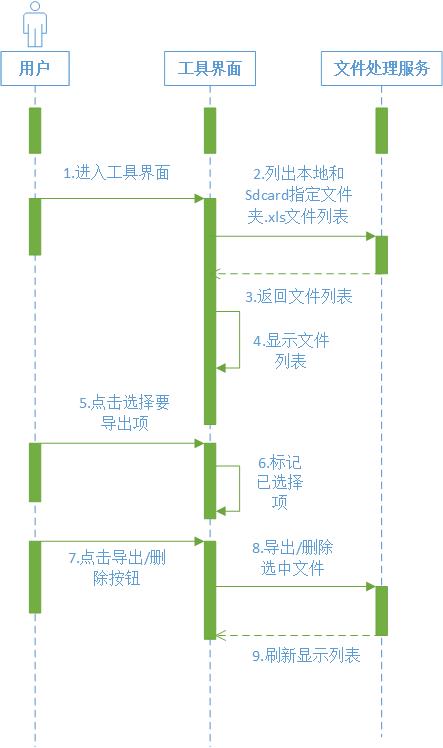


图7 导出/删除功能时序图

## 5.2系统详细设计

### 5.2.1 项目类的详细设计

对于一个Android应用项目来说，界面类的设计是至关重要的，它不像Web服务分成了MVC的架构，一个Android应用的展示层同时也是控制层；不同的界面之间数据需要通信，通过实现不同界面类暴露的接口来传递参数；由于Android的很多封装好的方法提供的特性比较单一，因此为了适应不同的项目对于功能的不同需求，通常会对Android系统提供的很多方法进行继承重写；对于很多只是功能性的实现的方法，可以自定义封装到工具类中。

综上所述，在下面会分段叙述关于界面类、接口、重写系统类、工具类。

（1）在Android中，界面的显示是以继承Activity来展示的，界面之间的切换如果直接全部使用Activity之间跳转的话，会消耗大量的系统资源，因此本软件使用Adapter+Fragment的组合来实现界面之间的切换，所有Fragment之间的参数传递通过一个主Activity来进行交互。这里列出所有和界面相关的类：

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 类名 | 功能 | 备注 |
| 1 | MainActivity | 主界面类，用以控制界面的切换以及界面之间参数的传递 | 继承自FragmentActivity |
| 2 | FileContentFragment | 文件界面类，包含所有文件界面的设置按钮 | 继承自Fragment |
| 3 | SettingContentFragment | 设置界面类，包含设置界面的有按钮的触发以及控制设置 | 继承自Fragment |
| 4 | GraphContentFargment | 曲线界面类，包含曲线相关的控制以及其他方法 | 继承自Fragment |
| 5 | ResultContentFragment | 结果显示界面类，显示实验结果数据 | 继承自Fragment |
| 6 | ToolContentFragment | 工具界面类，包含相关按钮的控制逻辑 | 继承自Fragment |

MainActivity包含的方法以及实现的功能（注：这里只有自定义方法，不包含继承自FragmentActivity的方法）如下表所示：

|  |  |  |
| --- | --- | --- |
| 序号 | 方法名 | 功能 |
| 1 | initialize() | 在应用启动时初始化创建应用所需的文件目录 |
| 2 | findAllControlById() | 初始化界面相关控件实例化 |
| 3 | initFragmentView() | 初始化五个Fragment界面相关接口实现 |
| 4 | resetTextView(int position) | 重置导航栏选中项的字体颜色 |
| 5 | changeTitleBtnColor(int btnId) | 更改导航栏选中项的背景色 |
| 6 | startGetDataFromDb() | 运行数据采集程序，绘制曲线 |
| 7 | stopGetDataFromDb() | 停止数据采集程序的运行 |
| 8 | printOutResult(int runTime) | 显示扩增实验结果 |

FileContentFragment包含的方法以及实现的功能（注：这里只有自定义方法，不包含继承自Fragment的方法，下同）如下表所示：

|  |  |  |
| --- | --- | --- |
| 序号 | 方法名 | 功能 |
| 1 | bindOnclickEvent() | 在该界面被创建时给相应按钮绑定响应事件 |
| 2 | newDialog() | 在新建按钮点击触发后显示弹出框 |
| 3 | initNewFileDialog(View view, final AlertDialog alertDialog) | 对弹出框进行相关按钮的响应事件绑定 |
| 4 | setListViewListener(ListViewListener lvl) | 提供ListViewListener接口以获得 Context |
| 5 | setCreateDialog(CreateDialog createDialog) | 提供CreateDialog接口以传递相关参数到MainActivity |
| 6 | setLabNameSetting(LabNameSetting labNameSetting) | 提供LabNameSetting接口以传递新建文件名参数 |

SettingContentFragment包含的方法以及实现的功能如下所示：

|  |  |  |
| --- | --- | --- |
| 序号 | 方法名 | 功能 |
| 1 | bindCheckBoxListener() | 给相应的Checkbox控件添加响应事件 |
| 2 | bindEditTextListener() | 给相应的Edittext控件添加输入检测监控 |
| 3 | bindButtonListener() | 给对应的按钮添加点击响应事件 |
| 4 | setAllContentReadOnly() | 全局控件只读设置 |
| 5 | setAllContentClickable() | 全局控件可获得焦点，可编辑 |
| 6 | setEditTextReadOnly(TextView view) | 设置Edittext控件只读 |
| 7 | setEditTextEditable(TextView view) | 设置Edittext控件可编辑 |
| 8 | getEditFocus(TextView view) | 设置某一个Edittext获得焦点 |
| 9 | setCheckboxFalse(CheckBox view) | 设置Checkbox不可点击 |
| 10 | setCheckboxTrue(CheckBox view) | 设置Checkbox可点击 |
| 11 | setStopbtnOnClickable() | 设置停止按钮可点击 |
| 12 | setRunbtnOnClickable() | 设置开始按钮可点击 |
| 13 | setListViewListener(ListViewListener lvl) | 提供ListViewListener接口以获得 Context |
| 14 | setCollectData(CollectData collectData) | 提供CollectData接口以传递相关数据到MainActivity |

GraphContentFragment包含的方法以及实现的功能如下所示：

|  |  |  |
| --- | --- | --- |
| 序号 | 方法名 | 功能 |
| 1 | bindButtonListener() | 给相应按钮添加响应事件 |
| 2 | bindHolePartListener() | 给相应孔添加点击响应事件 |
| 3 | setShowHoleChart() | 初始化设置显示所有孔曲线 |
| 4 | drawChart(int time) | 绘制曲线 |
| 5 | getListVals(Map<String, List<String>> result, int hole, int time, int type) | 返回采集到的数据点List |
| 6 | setStopbtnOnClickable() | 设置停止按钮可点击 |
| 7 | setRunbtnOnClickable() | 设置开始按钮可点击 |
| 8 | setDecreaseBtnTrue() | 设置溶解曲线按钮可点击 |
| 9 | setDecreaseBtnFalse() | 设置溶解曲线按钮不可点击 |
| 10 | setHexCheckboxTrue() | 设置HEX通道单选框可选 |
| 11 | setHexCheckboxFalse() | 设置HEX通道不可选 |
| 12 | setAmpButtonStyle(boolean flag) | 修改扩增曲线选中后的样式 |
| 13 | setDisButtonStyle(boolean flag) | 修改溶解曲线选中后的样式 |
| 14 | setRunRecordBtn() | 设置运行按钮点中后界面样式 |
| 15 | setStopRecordBtn() | 设置停止按钮选中后界面样式 |
| 16 | setListViewListener(ListViewListener lvl) | 提供ListViewListener接口以获得 Context |
| 17 | setCollectData(CollectData collectData) | 提供CollectData接口以传递相关数据到MainActivity |

ResultContentFragment包含的方法以及实现的功能如下所示：

|  |  |  |
| --- | --- | --- |
| 序号 | 方法名 | 功能 |
| 1 | showResultData() | 显示结果列表 |
| 2 | clearResultData() | 清除结果列表 |
| 3 | setListViewListener(ListViewListener lvl) | 提供ListViewListener接口以获得Context |
| 4 | setShowResultListData(ShowResultListData showResultListData) | 提供ShowResultListData接口以获得计算的实验结果数据 |

ToolContentFragment包含的方法以及实现的功能如下所示：

|  |  |  |
| --- | --- | --- |
| 序号 | 方法名 | 功能 |
| 1 | bindControlBtnListener() | 给功能按钮添加点击相应事件 |
| 2 | showFilesList() | 初始化显示本地和Sdcard文件列表 |
| 3 | directoryFiles(String path, Integer img) | 返回指定目录文件列表 |
| 4 | showLocalFiles() | 显示本地文件列表 |
| 5 | showExtFiles() | 显示Sdcard文件列表 |
| 6 | exportExcelOut() | 导出文件(.xls) |
| 7 | deleteLocalExcel() | 删除本地excel文件 |
| 8 | clearLocalExcel() | 清空本地excel文件 |
| 9 | setListViewListener(ListViewListener lvl) | 提供ListViewListener接口以获得Context |

（2）由于在Android中主线程是UI线程，在不同的界面间切换时，对于数据的交互，如果是直接以Activity跳转的话，可以使用Intent传递参数；但是在本项目中，由于使用了Adapter+Fragment的方式来进行界面间的切换，因此一个比较好的方法是：在Fragment子类中暴露接口，然后在MainActivity中实现，从而将需要传递的数据从Fragment中传递到MainActivity中，当需要的的数据在MainActivity完成交互后，Fragment中调用接口定义的方法来获得MainActivity中返回的数据以完成一次操作及相应的功能，这里列出所有自定义的接口，以及接口中方法的作用：

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 接口名称 | 方法定义 | 功能 |
| 1 | ListViewListener | public Context getMainContext(); | 获得MainActivity的上下文，用来操作UI控件 |
| 2 | CreateDialog | public void createOpenDialog(); | 文件界面打开按钮的弹出框 |
| public void createNewDialog(); | 文件界面新建按钮的弹出框 |
| public void createSaveDialog(); | 文件界面保存按钮的弹出框 |
| 3 | CollectData | public void useInstancePara(Map<String, String> paras); | 实例化时，将设置界面参数传递到MainActivity |
| public void getDataFromDb(Map<String, String> paras); | 在运行或停止按钮点击时将设置界面相关参数传递到MainActivity |
| public void stopGetData(Map<String, String> paras); | 传递运行/停止的状态信息 |
| 4 | LabNameSetting | public void setLabName(String labName); | 传递新建文件名 |
| 5 | ReDrawChart | public void reDrawChart(int type); | 刷新绘制曲线图 |
| 6 | ShowResultListData | List<Map<String, Object>> showResultData(); | 获得采集的实验数据 |

（3）由于Android系统类中使用的某些方法并不能满足项目设计时需要达到的功能性需求，因此在Android app来发中往往或重写部分用到的系统类以达到目的，这里列出这部分重写自系统类提供的功能：

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 重写类名 | 系统类 | 功能 |
| 1 | ChildClickableLinearLayout | LinearLayout | 设置LinearLayout及其子控件都可以被设置为不可获得焦点，不可编辑 |
| 2 | DetailViewPager | ViewPager | 修复Android系统存在的bug：界面左右滑动导致的抛出异常 |
| 3 | FileDialogFragment | DialogFragment | 自定义文件界面弹出框的样式 |
| 4 | FileItemAdapter | ArrayAdapter<> | 自定义ListView的显示 |
| 5 | FragmentAdapter | FragmentPagerAdapter | 自定义ViewPager的显示 |

（4）为了降低代码之间的耦合同时也为了便于阅读和维护，故将可复用的、功能性的代码放入统一的工具类MyUtil（单例模式）中，具体类中包含的方法以及功能如下表所示：

|  |  |  |
| --- | --- | --- |
| 序号 | 方法定义 | 功能 |
| 1 | public String getNumber(int input) | 返回两位数数字 |
| 2 | public boolean isNullEmpty(String str) | 判断字符串是否为Null或者空 |
| 3 | public String getTwoPointData(double input) | 保留小数点后两位 |
| 4 | public String formatDate() | 格式化日期 |
| 5 | public void createInitializeFolds() | 创建初始化文件夹 |
| 6 | public List<Map<String, Object>> readExcel(String[] para, String fileName, String directory) | 读取Excel表格数据 |
| 7 | public void createNewExcel(List<Map<String, Object>> data, String[] para, String fileName, String directory) | 创建新的空白实验数据Excel |
| 8 | public boolean validateText(String input) | 验证输入字符串是否包含违法字符 |
| 9 | public Map<String, Object> getLabDataFromPhone(int type, int chartType) | 采集实验数据 |
| 10 | public Map<String, List<String>> getSheetData(Sheet sheet) | 采集Excel中工作表的数据 |
| 11 | public void setRunbtnOnClickable(Button startBtn, Button stopBtn) | 设置一个按钮可点击，另一个不可点击 |
| 12 | public String getExtSdCardPath() | 获取挂载的可移除Sdcard路径 |
| 13 | public String copySdcardFile(String fromFile, String toFile) | 导出Excel文件到外置Sdcard |
| 14 | public void deleteFile(File file) | 删除文件 |
| 15 | public double divideValue(double val1, double val2, int valNum) | 执行除法并保留指定小数位 |
| 16 | public double getAverageValue(List<String> axisVal) | 计算平均值 |
| 17 | public Map<String, String> getChartResult(List<String> value, int type, int time) | 计算实验结果 |
| 18 | public Double getDtValue(List<String> value, int size, double aveX, double aveY) | 计算曲线的线性回归系数 |
| 19 | public Double countAmplificationDt(List<String> value, double curveDt) | 计算扩增的曲线的dt值 |
| 20 | public Double maxValue(List<Double> inputList) | 找出链表中的最大值 |
| 21 | public List<PointValue> getDisDtValue(List<String> result,int time, double initVal, double dis) | 返回计算的溶解曲线的数据点 |

# **第6章 系统实现**

## 6.1开发环境安装

开发Android应用前需要配置Android项目环境，包括：

（1）配置Java开发环境，可以参考网上教程或其他文本资料。

（2）安装开发工具：Android Studio，具体安装过程可以参考<http://www.android-studio.org/>网站的文档。

（3）安装Android sdk，具体的安装步骤以及注意事项可以参考网上教程或其他文本资料。注意：Android应用程序需要在Android模拟器或者对应的手机系统上运行，例如这里由于我是在手机上进行测试，需要的sdk版本应当和Android系统版本相对应，我手机是Android 5.1.1系统，所以使用的Android sdk版本就是api 22版本。

## 6.2具体功能实现

### 6.2.1 权限申请

由于Android系统是基于Linux内核的操作系统，因此系统中有着相当复杂的权限管理体系，应用想要完成某一项功能，必须要拥有相应的权限，才能完成，因此这里给出配置文件中申请所需权限的配置：

<!--获取访问网络权限-->

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

<!--在SDCard中创建与删除文件权限 -->

<uses-permission android:name="android.permission.MOUNT\_UNMOUNT\_FILESYSTEMS"/>

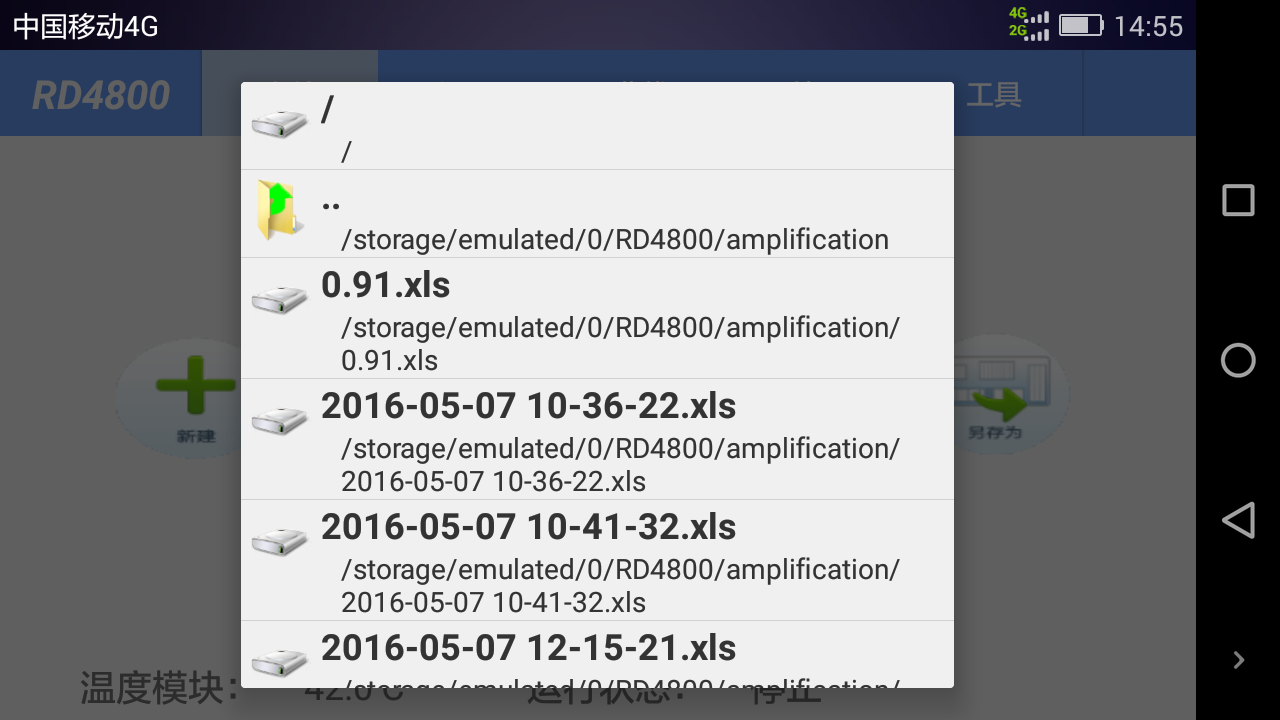
<!--往SDCard写入数据权限 -->

<uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE"/>

<uses-permission android:name="android.permission.READ\_EXTERNAL\_STORAGE"/>

注：权限申请的这部分代码写在AndroidManifest.xml文件中。

### 6.2.2 打开文件功能



在进入文件页面后，点击打开按钮，会弹出一个这样的Dialog，其中显示的每一条记录是使用ListView来展示的，其中第一行的反斜杠是表示进入系统根目录，第二个两个点代表进入上一级目录，其余项就是显示的.xls表格文件，这里可以让用户选择系统任意路径下的任意表格文件来打开。当用户点击这里列出的这些.xls文件项中某一行，弹出框会自动消失，并在后台读取相关文件内容，如果这个表格的内容不符合定义的实验数据格式，会抛出异常提示。

这个Dialog中的ListView的每一项布局如下xml所示：

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

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

android:orientation="horizontal"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:background="#f0f0f0"

android:id="@+id/file\_item\_id">

<ImageView

android:id="@+id/filedialogitem\_img"

android:layout\_width="32dp"

android:layout\_height="32dp"

android:layout\_margin="4dp"/>

<LinearLayout

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:orientation="vertical" >

<TextView

android:id="@+id/filedialogitem\_name"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:textColor="#333"

android:textSize="18sp"

android:textStyle="bold" />

<TextView

android:id="@+id/filedialogitem\_path"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:paddingLeft="10dp"

android:textColor="#333"

android:textSize="14sp" />

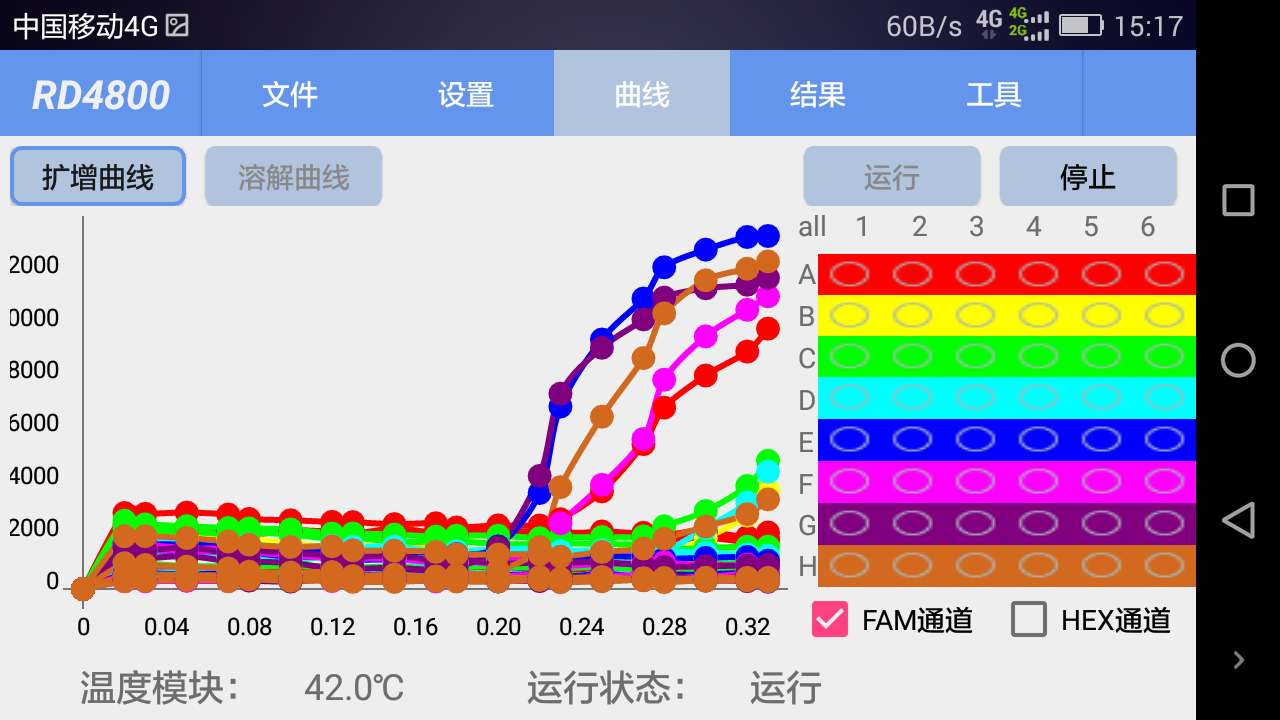
</LinearLayout>

</LinearLayout>

### 6.2.3 设置界面参数作用



这里通过点击相应的Checkbox来选择不同的运行条件，编辑Edittext内容来输入不同的参数；当溶解曲线选框没有被勾选时，起始温度、结束温度、度数误差等都无法进行编辑更改操作；在实时数据采集程序未运行时，停止按钮也是处于不可点击状态，只有运行按钮是可点击的；界面底部的温度模块以及运行状态是和设置界面的参数保持同步的（点击运行按钮后）；这里由于溶解曲线选框默认没有选中，则在运行时，如果没有更改，那么在曲线显示界面溶解曲线按钮是无法展示的，如下图所示，溶解曲线按钮不可点击。



这里列出点击运行按钮后的程序逻辑：

public void startGetDataFromDb() {//运行

if (excelData != null && excelData.size() > 0) {

excelData = new ArrayList<Map<String, Object>>();

mResultFg.clearResultData();//清除掉旧的实验结果

}

showRunType.setText("运行");

if (settingParas.get("default\_temp\_edit") != null) {

showTempText.setText(settingParas.get("default\_temp\_edit") + "℃");

}

if (settingParas.get("hex\_graph\_choice").equals("false")) {//如果设置页面没有勾选采集hex通道数据，则这里无法点击选择

mGraphFg.setHexCheckboxFalse();

} else if (settingParas.get("hex\_graph\_choice").equals("true")) {

mGraphFg.setHexCheckboxTrue();

}

//handler.post(runnable);

new Thread(ampTaskThread).start();

if (settingParas.get("dissolution\_graph\_choice").equals("true")) {

//handler.post(disRunnale);

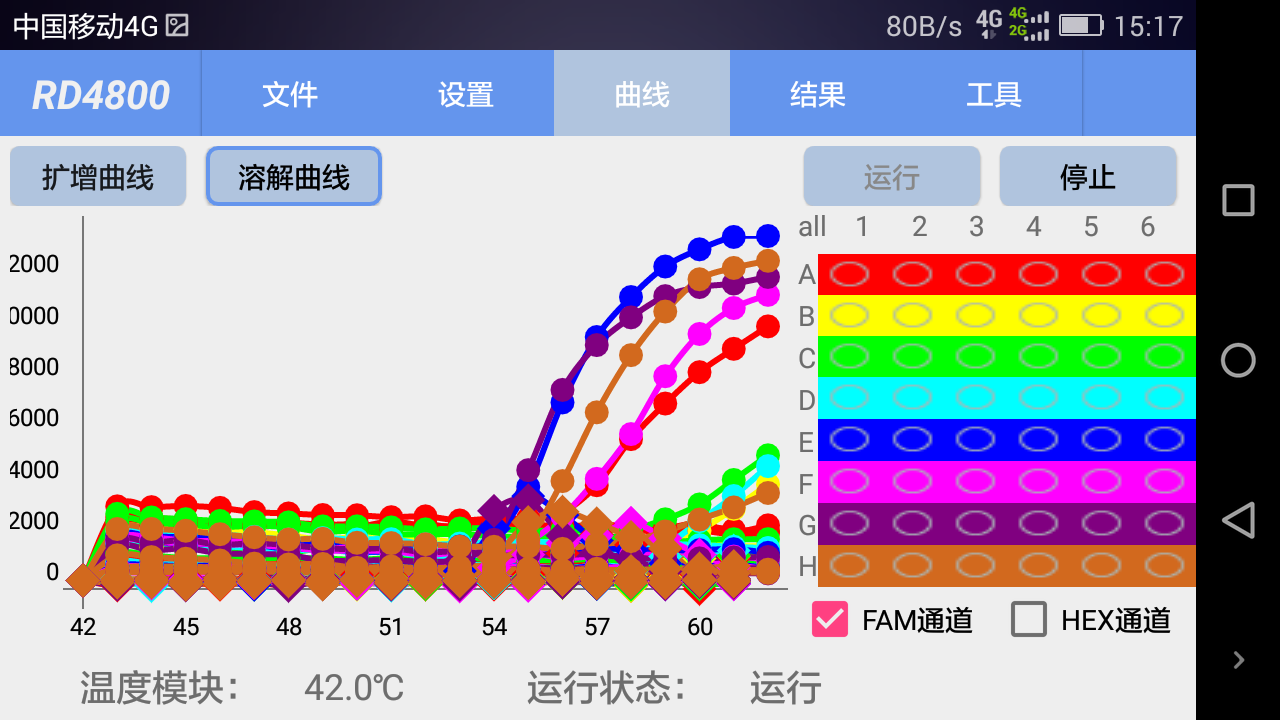
new Thread(disTaskThread).start();

}

}

### 6.2.4 曲线界面实现

下面的这个曲线界面截图和之前的界面相比，不同的地方在于溶解曲线按钮是处在选中的状态，因此下方曲线显示部分的曲线内容就是溶解曲线以及对应的求导数曲线了（注：由于没有实际的溶解曲线数据，因此这里将扩增实验的模拟数据文件当做溶解曲线的模拟数据文件来使用并绘制的求导数曲线，其中溶解曲线的点是以圆表示，求导曲线的点则以菱形表示）；扩增曲线的x轴的值是由分钟转换为小时的两位数小数，而溶解曲线的数据点采集则是基于从起始温度加上温度差值来表示的坐标值上；需要注意的一点时，为了区别显示的曲线是哪种类型的，因此给选中的按钮添加了蓝色粗边框以表示选中状态，同样这里在选中后也会防止触发重复点击。



由于算法实现代码较长，这里不再缀述，这里只说下关于筛选部分的算法逻辑：

（1）判断设置界面HEX通道是否选中，选中则采集HEX通道数据。

（2）采集所有48孔的数据。

（3）将所有孔以编号为key，true代表选中，false代表未选中放入一个map中作为孔筛选参数；FAM及HEX通道选中与否作为通道筛选参数

（4）点击选中或取消选中某一孔，触发重新绘制曲线；选中单通道或双通道，触发重新绘制曲线。

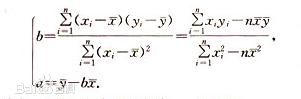
（5）通过孔筛选参数以及通道筛选参数判断绘制哪些曲线。

### 6.2.5 结果界面实现

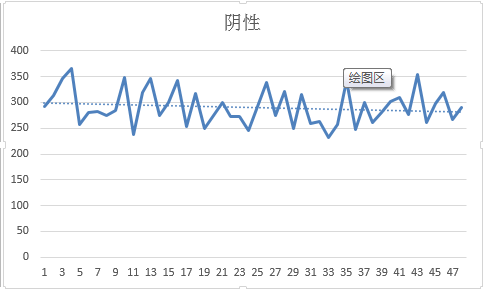


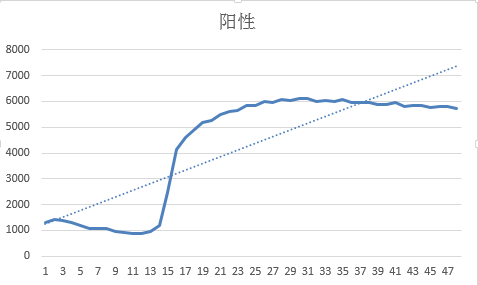
由于目前来说，实验的模拟数据是只有采集到的恒温扩增的扩增效率，孔位是可以通过计算得到，其他的像是条码、姓名、性别、年龄、项目等暂时都是无数据；dt值即是要求阴阳性算法计算的曲线斜率，结果就是判断该条曲线是阴性还是阳性的。具体的算法实现我会将代码放在附录部分，这里只给出算法的实现逻辑：

（1）计算所有点的x轴均值以及y轴点的均值。

（2）根据公式：计算曲线的线性回归系数。

（3）由于实验结果分为阳性和阴性，曲线二次拟合的结果分别近似于斜率较大的直线和斜率为负的直线，如下图所示。因此可以得出结论：在扫描仪采集数据误差可接受范围内，阳性实验结果的曲线线性回归系数应当普遍大于某一个数，对应的，阴性实验结果的曲线线性回归系数就应该小于这个数了。为了保持结果的准确性我这里使用的边界条件值是50\*60，求得dt值小于此数的即是阴性实验结果，否则就是阳性实验结果。





（4）对于阴性实验结果，线性回归系数即是dt值；对于阳性实验结果，则不一样：从所有点数据中找出连续斜率大于该曲线线性回归系数的斜率，其中最大的一个值即是所要计算的dt值。

### 6.2.6 工具界面实现



在工具界面栏中，左边是本地SD卡内应用目录文件夹下的.xls文件，右边是可移除SD卡内应用目录文件夹下的.xls文件。当选中右侧列表中的某些项之后，点击导出按钮则这些文件项会出现在右侧的列表中，删除操作类似，不过刷新的是本地文件列表目录。由于算法实现代码较长，这里不再缀述，只给出算法的实现逻辑。

算法逻辑如下：

（1）点击列表项，触发相应事件更改选中项背景色表示选中。

（2）点击导出按钮，代码采集选中项文件名以及文件路径

（3）将选中的文件以二进制流的形式拷贝到指定外置SD卡目录

（4）刷新右侧文件列表。

# **第7章 结束语**

一点一滴，从第一行代码开始，经过2个月的学习、尝试、设计、开发、调试，被我起名为RD4800的Android应用终于算是圆满完成，基本的应用功能都已经测试通过。尽管个人很是欣喜，然而我注意到这个应用仍有诸多不完善的地方，比如软件的开启速度比较慢、界面的美观性不足、提示信息比较僵硬等都有待提升。在这次Android开发之旅中，我学到了很多东西，Android界面的设计、UI线程与普通线程间的通信方法、Java文件操作的注意事项、遇到问题的解决方案和思路等这些都让我收获颇多。

总结一下大概就是：

（1）Java接口回调的通信方式

（2）Java进行文件操作的种种要点

（3）进行Android开发的经验

（4）分析问题，查找资料解决问题的能力提升

# **致谢**

从毕业设计项目的立题到开始写毕设文档，这接近3个月的时间，赵逢禹，袁旭军老师给了我很多的建议和指导，特别赵逢禹老师是在功能性需求上的解答到阴阳性算法的设计概念都给了我很大的启发，加上我不断地进步学习Android开发知识，终于完成了这个RD4800恒温扩增荧光检测仪软件并且测试通过。

在此我对他们表示由衷的感谢！谢谢两位老师的指导帮助！

# **参考文献**

# **附录**

## 附录A：源代码（典型性）

（1）绘制曲线：

/\*\*

\* @param time 运行次数，即绘制出来的点数

\*/

public void drawChart(int time) {

Map<String, List<String>> famResult = null;

Map<String, List<String>> hexResult = null;

double initTem = Double.parseDouble(paras.get("dissolution\_tempnum\_edit"));

double tempDis = Double.parseDouble(paras.get("change\_counttemp\_edit"));

int type = 0;

if (dissolutionType.get("amp\_btn").equals("true")) {

type = 1;

if(labData == null || labData.size() == 0) {

return;

}

} else if (dissolutionType.get("dis\_btn").equals("true")) {

type = 2;

if(dissolutionData == null || labData.size() == 0) {

return;

}

}

if (type == 1) {

if (chartType.get("fam\_checkbox").equals("true")) {

famResult = (Map<String, List<String>>) labData.get("FAM");

}

if (hexCheckbox.isClickable()) {//可点击，双通道采集数据

if (chartType.get("hex\_checkbox").equals("true")) {//显示hex

hexResult = (Map<String, List<String>>) labData.get("HEX");

}

}

} else if (type == 2) {

if (chartType.get("fam\_checkbox").equals("true")) {

famResult = (Map<String, List<String>>) dissolutionData.get("FAM");

}

if (hexCheckbox.isClickable()) {//可点击，双通道采集数据

if (chartType.get("hex\_checkbox").equals("true")) {//显示hex

hexResult = (Map<String, List<String>>) dissolutionData.get("HEX");

}

}

}

List<Line> lines = new ArrayList<Line>();

for (int i = 1; i <= 48; i++) {//从1到48个孔

int num = (i - 1) / 6;

if (showHoleChart.get(String.valueOf(i)).equals("true")) {//该孔曲线显示

if (famResult != null) {

Line line = new Line(getListVals(famResult, i, time, type));

line.setColor(colors[num]);

line.setCubic(true);

line.setHasLabelsOnlyForSelected(true);

lines.add(line);

if(type == 2) {

Line lineFamDt = new Line(mu.getDisDtValue(famResult.get(String.valueOf(i)),time, initTem, tempDis));

lineFamDt.setColor(colors[num]);

lineFamDt.setCubic(true);

lineFamDt.setShape(ValueShape.DIAMOND);

lineFamDt.setHasLabelsOnlyForSelected(true);

lines.add(lineFamDt);

}

}

if (hexResult != null) {

Line line = new Line(getListVals(hexResult, i, time, type));

line.setColor(colors[num]);

line.setCubic(true);

line.setHasLabelsOnlyForSelected(true);

lines.add(line);

if(type == 2) {

Line lineHexDt = new Line(mu.getDisDtValue(hexResult.get(String.valueOf(i)),time, initTem, tempDis));

lineHexDt.setColor(colors[num]);

lineHexDt.setCubic(true);

lineHexDt.setShape(ValueShape.DIAMOND);

lineHexDt.setHasLabelsOnlyForSelected(true);

lines.add(lineHexDt);

}

}

} else {

continue;

}

}

LineChartData data = new LineChartData();

data.setLines(lines);

Axis axisX = new Axis(); //X轴

axisX.setTextColor(Color.BLACK);

data.setAxisXBottom(axisX);

Axis axisY = new Axis(); //Y轴

axisY.setTextColor(Color.BLACK);

data.setAxisYLeft(axisY);

mainChart.setLineChartData(data);

mainChart.setInteractive(true);

mainChart.setZoomType(ZoomType.HORIZONTAL);

mainChart.setContainerScrollEnabled(true, ContainerScrollType.HORIZONTAL);

}

（2）扩增曲线数据采集：

public class AmpTaskThread implements Runnable {

public volatile boolean stop = false;

@Override

public void run() {

Looper.prepare();

if (settingParas == null) {

return;

}

Integer temp = Integer.parseInt(settingParas.get("default\_keeptime\_edit").toString().trim());

while (!stop && runTime < temp) {

runTime++;

if (settingParas.get("hex\_graph\_choice").equals("true")) {

labData = mu.getLabDataFromPhone(2, 1);

} else {

labData = mu.getLabDataFromPhone(1, 1);

}

mGraphFg.setLabData(labData);

mGraphFg.drawChart(runTime);

Message msg = handler.obtainMessage();

msg.what = 1;

msg.sendToTarget();

//handler.postDelayed(this, refreshTime);

try {

Thread.sleep(refreshTime);

} catch (InterruptedException e) {

e.printStackTrace();

}

}

ampFlag = true;

stopGetDataFromDb();

Message msg = handler.obtainMessage();

msg.sendToTarget();

Looper.loop();

}

}

（3）溶解曲线数据采集：

public class DisTaskThread implements Runnable {

public volatile boolean stop = false;

@Override

public void run() {

Looper.prepare();

if (settingParas == null) {

return;

}

Integer tempCount = Integer.parseInt(settingParas.get("change\_counttemp\_edit"));//度数误差

Double tempGap = Double.parseDouble(settingParas.get("change\_stoptemp\_edit")) - Double.parseDouble(settingParas.get("dissolution\_tempnum\_edit"));

while (!stop && disTimes < mu.divideValue(tempGap, tempCount, 0)) {

disTimes++;

if (settingParas.get("dissolution\_graph\_choice").equals("true")) {//采集溶解曲线数据

if (settingParas.get("hex\_graph\_choice").equals("true")) {

dissolutionData = mu.getLabDataFromPhone(2, 2);

} else {

dissolutionData = mu.getLabDataFromPhone(1, 2);

}

mGraphFg.setDissolutionData(dissolutionData);

mGraphFg.drawChart(disTimes);

if (settingParas != null && settingParas.size() > 0) {

Message msg = handler.obtainMessage();

msg.what = 2;

msg.sendToTarget();

}

try {

Thread.sleep(temperatureTime \* tempCount);

} catch (InterruptedException e) {

e.printStackTrace();

}

}

}

disFlag = true;

stopGetDataFromDb();

Message msg = handler.obtainMessage();

msg.sendToTarget();

Looper.loop();

}

}

（4）阴阳性算法实现（主要部分）：

/\*\*

\* @param value 扩增采集到的数据

\* @return 线性系数dt

\*/

public Double getDtValue(List<String> value, int size, double aveX, double aveY) {//注意这里需要将x轴的值转为小时值，所以换算后需要将结果乘以60

double numerator = 0.0;

double xDeno = 0.0;

for (int i = 0; i < size; i++) {

numerator += (i + 1) \* Double.parseDouble(value.get(i));

xDeno += Math.pow((i + 1), 2);

}

numerator = numerator - size \* aveX \* aveY;

double denominator = xDeno - size \* Math.pow(aveX, 2);

numerator = numerator \* hourTime;

return divideValue(numerator, denominator, 2);

}

/\*\*

\* @param value 扩增采集到的数据点

\* @return 真正扩增曲线的dt值

\*/

public Double countAmplificationDt(List<String> value, double curveDt) {

List<Double> dyList = new ArrayList<Double>();

List<Double> resultList = new ArrayList<Double>();

double temp = 0.0;

for (int i = 0; i < value.size() - 1; i++) {

temp = Double.parseDouble(value.get(i + 1)) - Double.parseDouble(value.get(i));//两点之间的差值

dyList.add(temp);

}

for (int i = 0; i < dyList.size() - 1; i++) {

if (dyList.get(i) > curveDt && dyList.get(i + 1) > curveDt) {//连续两个点斜率大于拟合拟合直线的斜率

resultList.add(dyList.get(i));

continue;

}

if (dyList.get(i) > curveDt && dyList.get(i + 1) < curveDt) {//曲线开始平稳乃至下降

if (resultList.size() >= 1) {

resultList.add(dyList.get(i));

break;

} else {

continue;

}

}

}

if (resultList.size() == 0) {//未找到比拟合直线斜率大的

return -1.0;

} else {

return maxValue(resultList);

}

}

public Double maxValue(List<Double> inputList) {

double temp = 0;

if (inputList != null && inputList.size() > 1) {

temp = inputList.get(0);

for (int i = 1; i < inputList.size(); i++) {

temp = temp > inputList.get(i) ? temp : inputList.get(i);

}

}

return temp;

}

## 附录B：xml界面布局

（1）主界面：

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

<RelativeLayout xmlns: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="yxs.usst.edu.cn.project.MainActivity">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical">

<include

android:id="@+id/id\_fragment\_title"

layout="@layout/title\_fragment" />

<yxs.usst.edu.cn.project.custom\_class.DetailViewPager

android:id="@+id/id\_fragment\_content"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_weight="1">

</yxs.usst.edu.cn.project.custom\_class.DetailViewPager>

<include

android:id="@+id/id\_fragment\_bottom"

layout="@layout/bottom\_fragment" />

</LinearLayout>

</RelativeLayout>