

项目编号: IPP10052

# 上海交通大学

## “大学生创新实践计划” 项目研究论文

论文题目: 基于传感技术的图书馆图书定位系统的研究与设计

项目负责人: 蒋如飞 学院(系): 机械与动力工程学院

指导教师: 赵春宇 学院(系): 电子信息与电气工程学院

参与学生: 陈金强 吕恺 王文辉 黄雨晗

项目执行时间: 2014 年 9 月 至 2015 年 11 月



## 摘要

由于高校图书馆大都开架借阅，图书乱放的现象普遍存在。随着图书收藏的数量越来越多，乱架图书难以及时整理，读者按索书号找书往往无法找到。RFID图书定位技术可以解决这一问题，但成本过高。本文采用全新的定位方案，设计出基于位置传感器的图书定位系统，实现了低成本监测图书位置的目的。

本文设计的定位系统仅利用位置感应器实现图书的监测，不需要监测图书上绑定的标签。在该系统中，抽出某层的一本书时，位于该层底部的位置感应器会采集到该书被抽出时的位置，根据预先收集好的图书——位置的对应关系识别出被抽走的图书；插入一本书时，由于读者一般会尽量还回原来的位置，采集到位置后系统会自动将离该位置最近的被取走的图书与之匹配，生成新的图书——位置对应关系。同时，系统会评估该书乱架的可能性，当可能性大于一定值时会提醒馆员前去整理。由于该书的位置已经确定，馆员只用扫描图书上绑定的标签就能完成整理，工作量大大减少，甚至可以由读者在插入时用手机扫描完成。

**关键词：**图书定位，位置感应，图书管理

## ABSTRACT

Since most of the university libraries are Open-shelf, phenomenon that books are misplaced is widespread. As the increase of the number of collections of books, it is more difficult to timely finishing arranging the disorder books, readers find books by call often cannot be found. Book with RFID positioning technology can solve this problem, but the cost is too high. In this paper, a new location solutions is designed with position sensor positioning system based on the book, to achieve the purpose of monitoring the books at low cost locations.

This design of positioning system using only position sensors monitor realization of books, does not have to monitor the bound label. In this system, as a book is pulled out of a layer, the layer located at the bottom of the position sensor location will be collected, according to the information of books collected in advance - to identify the location of the book pulled out; when inserting a book, because the reader will generally try to return it to its original position, after the acquisition to position the system will automatically match the position from recent books have been removed, and generate new books - corresponding to the position relationship. At the same time, the system will assess the possibility of the book disorder, when the possibility of disorder is greater than a certain value, the system will remind librarians to arranging. Since the books' position has been determined, the librarian only need to scanning books bound label or even the reader can complete the work with a mobile phone on insertion so the work is greatly reduced.

**Keywords:** localization, position, sensing, library management

### 1. 绪论

#### 1.1 课题研究背景

信息时代的发展给人们的生活带来了巨大的变化,也给各行各业带来了挑战。作为知识的集聚地,图书馆的建设同样面临着信息化的挑战。RFID (Radio Frequency Identification), 又称无线射频识别,是一种通信技术,可直接通过无线电信号识别特定目标并读写相关数据,无需在系统与特定目标之间建立机械或光学接触。目前,国内外已经有很多高校图书馆采用了 RFID 技术对图书进行智能管理,以便更好地提高图书流通管理和典藏管理的工作效率,同时使图书馆管理员腾出时间来为读者提供更多的服<sup>[1]</sup>。

采用 RFID 技术后,图书馆基本实现了自助借还书的功能,但对读者费时费力的找书过程却没有很大的帮助。当一本书被翻阅后放在其它的书架上,该书就变为乱架图书,读者很难找到一本“藏在”大量图书中的一本书;同时,图书管理员也很难发现乱架图书,这将导致工作效率的低下<sup>[2]</sup>。图书定位可以有效地解决读者找书的问题,如今,图书定位技术的研究已成为图书馆智能化研究中热门的研究领域。

RFID 室内定位系统是图书定位的方法之一,基本原理是在室内架设天线,通过感应 RFID 标签确定目标物体的位置。目前已经有很多针对图书定位的 RFID 室内定位系统被开发出来,例如密歇根州立大学研究开发的 LANDMARC 系统、MIT Oxygen 项目开发的 Cricket 系统等,都是较为优秀的基于 RFID 的室内定位系统,但是这些系统需要在室内布置天线,普遍比较复杂,而且需要针对不同的室内状况设计,成本较高。

另一种图书定位方案是智能书架。智能书架是一种已经可以投入使用的图书定位设备,原理是在书架上安装 RFID 感应装置,利用装置读取书架上每一本图书的 RFID 标签,达到遍历读取图书的目的。中国国家图书馆是国内应用 RFID 系统的图书馆中最大的图书馆,使用了智能书架对图书进行定位,不仅能为读者提供准确的图书位置信息,还能找出乱架的图书,提醒馆员前去整理。然而,整个系统的造价高,时间长,不适合高校图书馆等中小型图书馆。目前的智能书架的开发还只是处于先导试验中,主要是用来作为图书馆的一种营销手段,即使开发出了智能书架系统,所采用的图书定位方法也相当繁琐,定位精度不高,而且成本较高,在图书的借还流程、图书精确定位等方面仍有待进一步研究<sup>[3]</sup>。图书定位是图书馆智能化的代表技术之一,然而长期以来由于成本的问题,图书定位一直无法在中小型图书馆中推广,只有类似国家图书馆之类的大型图书馆才有时间和资金搭建图书定位系统。

## 1.2 课题研究的目的和意义

低成本地实现图书定位对图书馆有巨大的实用价值。据了解,我校投入了上百万资金对图书馆进行 RFID 改造,意在在信息化浪潮中抓住发展的机会,建立一个以读者服务为中心的智能化图书馆。然而,RFID 系统虽然实用,其主要功能只是体现在图书管理中,对图书乱架的状况却没有太大改善。本课题研究的是一种新的定位方案,在满足定位要求的前提下尽量降低成本,以实现在高校图书馆中应用的目的。精确的图书定位是解决图书乱架的最佳方案之一,可靠的定位系统可以极大提高图书馆的智能化程度,同时大大改善读者的借阅体验。本课题将建立一个完整的定位系统,实现在低成本的对图书的精确定位。

## 1.3 课题研究的主要内容

本课题的研究主要内容如下:

- (1) 分析 RFID 定位的方案,总结出方案的缺点;
- (2) 提出全新的定位方案,详细阐述方案原理;
- (3) 方案各部分的功能设计;
- (4) 系统各部分的详细设计;
- (5) 系统的具体制作;

(6) 系统的开发与测试。

## 2. 总体方案设计

为了实现低成本的目的，本课题完全抛弃了绑定标签定位的思路，利用图书排列的有序性，使用主要由开关构成的位置感应器实现了图书定位。由于整个系统的主要感应器件仅仅是简单的位置感应器，相对于 RFID 定位系统成本将大大降低。

### 2.1 系统架构

本课题研究的定位系统由传感节点、上位机、服务器和手机软件 4 部分组成。传感节点监测图书的变动情况，并将收集到的数据发送给上位机；上位机处理接收到的数据后，将数据发送到服务器，服务器再根据数据更新数据库中图书的位置信息；手机软件可以查询数据库中图书的位置信息，还可以自助还书，扫描书和书架上的二维码后，手机会自动发送数据到服务器，完成图书整理。

### 2.2 系统工作原理

定位系统的基本单位是层。每层书架的底部有传感器，用来感应变动的位置和长度；右侧是提供推力的机械装置，用来将书推紧。如图 2-1 所示。

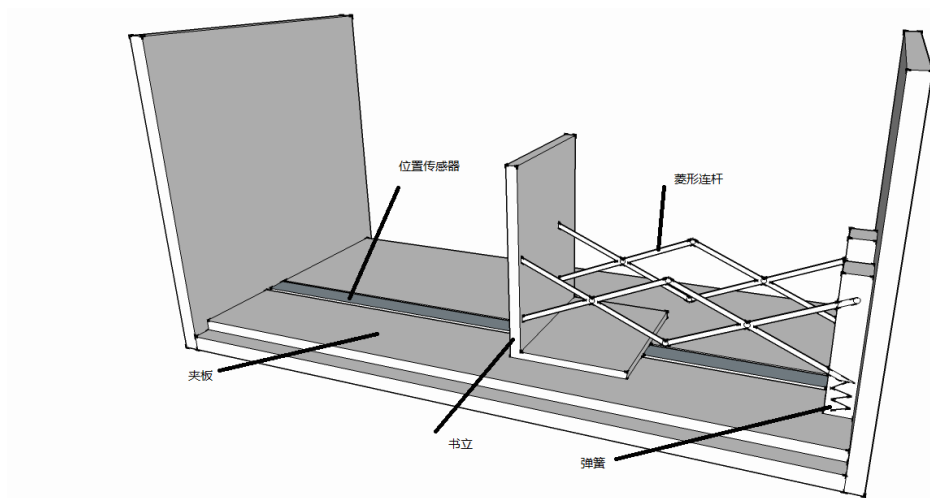


图 2-1 传感器和提供推力的机械装置示意图

抽出某层的一本书时，底部的位置感应器会感应到变动的位置和长度（书的厚度），由于初始情况下图书——位置对应关系在图书上架时已经确定，系统可以根据之前的关系识别出被抽走的图书。之后书立会将书推紧，在被抽走图书右侧的图书将左移，移动的距离等于被抽走图书的厚度。由于知道了图书的移动情况，系统可以根据采集到的数据及时更新图书——位置对应关系。

插入一本书时，存在三种情况：

1. 馆员整理时插入，插入时扫描书上绑定的二维码，系统会自动将变动的位置跟该书匹配，从而生成新的图书——位置对应关系；
2. 读者插入，插入时用手机扫描了书和书架上绑定的二维码，相当于馆员的整理。
3. 读者插入，未扫描二维码。由于大部分读者会根据记忆尽量还回原处，而在一个书架上借书



的人往往不多，所以采集到变动位置后，系统会自动将离该位置最近的被抽出的图书与之匹配。同时系统会评估该位置上图书乱架的可能性，当可能性大于一定值时会提醒馆员前去整理。由于位置已经知道，馆员只需要扫描下书上二维码即可完成整理。

图书位置变动后，图书跟位置的关系可以得到及时的更新，变动的图书也能被识别出来，从而定位功能得到了实现。本系统相对于 RFID 定位系统主要的差别在于乱架图书无法完全确定，只能评估出乱架的可能性。不过，大多数读者不会刻意乱放图书，就近匹配的方法能取得不错的效果；同时，也会有相当一部分读者愿意随手扫描下书上的二维码，帮助馆员减轻整理的负担；最关键的是，馆员整理图书的负担大大降低，只需要走到指定位置扫一下二维码就能完成整理，不需要费时费力地清点。相对于 RFID 定位系统，本系统简单可靠，能耗低，图书跟位置的对应关系可以实时更新，定位效果也毫不逊色。

### 2.3 传感节点方案设计

传感节点主要分为传感器、提供推力的机械装置和微处理器单元三个部分。传感器负责感应图书抽出或插入时所在的位置；书立负责将书推紧，使图书位置跟图书一直存在对应关系；微处理器负责收集传感器监测到的数据，并将数据发送给上位机。

传感器由开关列和电路板构成。开关列由等间距的开关组成，两个相邻开关的间距越小，传感器所能感应到的书的厚度越小，但同时一层的开关数增多。由于书的厚度一般大于 5 毫米，开关间距取 5 毫米比较合适。当书在开关上方时，作用在开关上的力在 0.5N 左右，因此开关的动作用力应在 0.3N 左右。当书被左右推动时不能被开关卡住，开关动作行程应在 1mm 下，而且头部应成圆弧状。

电路板的功能是监测开关的通断情况。以书架一层一米长为例，电路板需要监测的开关有 200 个，而且能监测多个开关同时按下的情况；提供推力的机械装置采用菱形连杆的结构设计，如图 2-2 所示，弹簧放置在书架的最右端；微处理器单元的主要部分是单片机，考虑到每个传感器传输到单片机的数据量很小，可以多个传感器共用一个单片机。

### 2.4 系统软件设计

#### 2.4.1 单片机嵌入式程序设计

本系统中使用的单片机是 C8051F340，程序采用 C 语言编写，实现的功能是：每隔一段时间监测开关的状态，当状态有变化时发送数据到上位机。

#### 2.4.2 上位机软件设计

上位机软件使用 Qt 集成开发环境编写而成，可以在 window7/8 系统中运行。在接收到单片机通过串口发送过来的数据后，软件会分析数据，识别出书被抽走还是插入，找出变动的具体位置，然后将处理好的数据发送给服务器。

#### 2.4.3 服务器软件设计

服务器选用的是 tomcat7，运行在 linux 系统上，在本课题中运行在云服务器上的 ubuntu 系统中。服务器端采用 J2EE 三层架构，基于 SSH（Struts2+Spring+Hibernate）框架。服务器实现的功能有：添加和删除图书；存储图书的基本信息和位置、状态；接收读者或馆员扫描二维码和上位机发送过来的数据，并根据数据更新图书的位置和状态；根据读者的搜索提供图书信息；提供 android 软件下载地址。

#### 2.4.4 android 软件设计

Android 软件在 android studio 集成开发环境中使用 java 语言编写而成，实现的功能有：图书搜索服务；定位服务，将图书的位置信息可视化地显示在一个图书馆地图上；地图漫游服务，当读者点击地图中的某一个书架时显示书架上图书；收藏服务，收藏图书后下次可在手机上直接点击查看最新的位置；自助还书服务，读者如果愿意配合，将书插入书架时可以扫描书和书架上的二维码，

完成图书整理。

### 3. 系统详细设计

#### 3.1 传感节点详细设计

##### 3.1.1 传感节点结构

传感节点以 C8051F340 单片机为核心, 利用单片机自带的 10 位 A/D 转换器将位置传感器的电压信号转换成数字量。单片机每隔一段时间会测量一次传感器的信号, 当信号变化时, 单片机会将得到的数字量通过串口发送给上位机。

##### 3.1.2 单片机的选用

C8051F340 单片机是完全集成的混合信号片上系统型 MCU, 主要特性有:

1. 高速、流水线结构的 8051 兼容的微控制器内核 (可达 48MIPS);
2. 真正 10 位 200 ksp/s 的单端/差分 ADC, 带模拟多路器;
3. 精确校准的 12MHz 内部振荡器和 4 倍时钟乘法器;
4. 4 个通用的 16 位定时器;
5. 多达 40 个端口 I/O (容许 5V 输入);
6. 片内 Silicon Labs 二线 (C2) 开发接口允许使用安装在最终应用系统上的产品 MCU 进行非侵入式 (不占用片内资源)、全速、在系统调试。调试逻辑支持观察和修改存储器和寄存器, 支持断点、单步、运行和停机命令。在使用 C2 进行调试时, 所有的模拟和数字外设都可全功能运行。两个 C2 接口引脚可以与用户功能共享, 使在系统调试功能不占用封装引脚。

得益于 C8051F340 单片机的强大功能, 在本课题, 仅用一个单片机和 C8051F 系列单片机专用调试器就组成了微处理器单元, 不需要任何外围电路。

##### 3.1.3 传感器详细设计

开关是传感器中最重要的部分, 开关的行程、动作力等性能直接影响到感应的效果。根据感应的要求, 开关必须具有的特性有: 开关行程小于 1mm, 而且头部为弧形, 以使推动书的时候能开关能被按下; 动作力在 0.5N 左右, 书的重量能压下开关; 宽度小于开关间隔, 否则无法安装。调研发现, 市面上满足此类要求的开关较少, 只有检测开关能满足要求, 但单个成本过高。为了降低成本, 可以定制机械开关或橡胶开关, 由于定制的开发成本高, 本课题中使用的是监测开关。

由于 1 层书架有近 200 个开关, 电路板需要分为 4 块, 每块监测 48 个开关。在一块电路板中, 48 个开关分为 8 组, 每组 6 个开关。一组开关全部串联, 同时每个开关并联一个电阻, 第  $k$  个电阻的阻值为  $2^{(k-1)}$  次方千欧姆, 这样能确保任何一种按下开关的情况对应的总阻值都不同, 从而通过测量阻值就能判断一组的开关通断情况。一组的电路如图 3-1 所示。

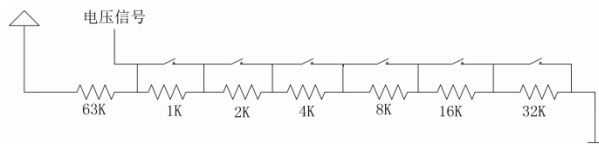


图 3-1 一组开关的电路原理图

8 组的电压信号如果传入单片机需要 8 根引脚, 造成了引脚的浪费。CD4051 是一款 8 选一模拟开关, 应用在本电路后, 一块电路板仅需要一根信号线, 大大节约了所需引脚数。

画好电路图后, 需要根据电路图画电路板。Altium Designer 是原 Protel 软件开发商 Altium 公司推出的一体化的电子产品开发系统, 主要运行在 Windows 操作系统。这套软件通过把原理图设

计、电路仿真、PCB 绘制编辑、拓扑逻辑自动布线、信号完整性分析和设计输出等技术的完美融合，为设计者提供了全新的设计解决方案，使设计者可以轻松进行设计，熟练使用这一软件必将使电路设计的质量和效率大大提高。本课题使用的版本是 AD10，为了降低生产成本，需要将电路板画的尽量窄，最终画出的电路板宽度仅 16 毫米。画好的 PCB 图如图 3-2 所示：

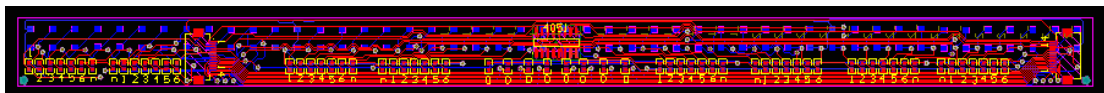


图 3-2 电路板设计图

### 3.2 单片机编程

单片机需要将开关通断情况即时发送给上位机，所以需要每隔一段时间采集传感器的电压信号，并把采集到的信号跟之前的做比较，当发现有变化时将数据通过串口发送给上位机。程序流程图如图 3-3 所示。

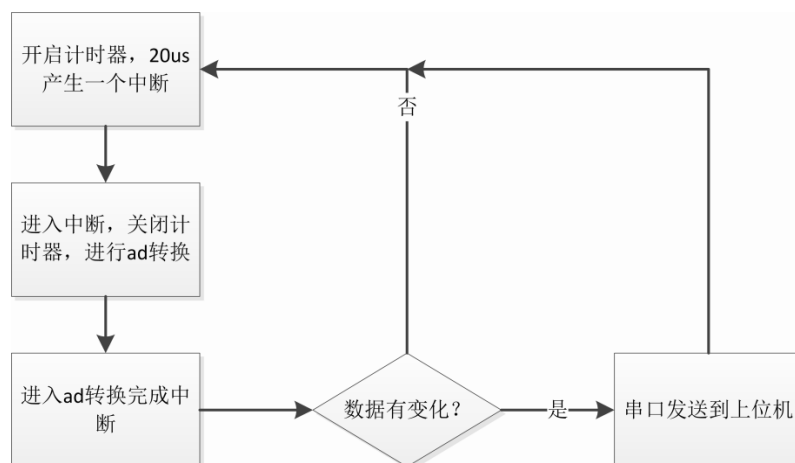


图 3-3 单片机程序流程图

ad 转换完成中断程序是单片机程序中的核心部分,在这段程序中需要对采集到的信号进行比较,当信号有变化时将数据通过串口发送到上位机,代码如下:

```

void ADC0_ISR (void) interrupt 10
{
    static unsigned long accumulator = 0;    // Accumulator for averaging
    static unsigned int measurements = 1000; // Measurement counter
    static unsigned int mm=0;
    static unsigned char change=0;
    static unsigned char over=0;
    unsigned char ii;
    int result=0;
    AD0INT = 0;                                // Clear ADC0 conv. complete flag
    TR2      = 0;
    accumulator += ADC0;
    
```



```
measurements--;

if(measurements==0)
{
    measurements = 1000;
    result = accumulator / 1000;
    accumulator=0;
    //    mV = result * 3300 / 1023;
    //    printf("P1.1 voltage: %ld mV\n",mV);
    //    putchar(deal(result));
        position[change][mm]= deal(result);
        if ((mm==(8*pis-1)))
            //测得一组 32 个数据后
        {
            mm=0;
            over=1;        //重新测一组
            if (decide()==1)
                //如果有就发送数据
            {
                putchar(0xaa);
                for(ii=0;ii<8*pis;ii++)
                {
                    putchar(position[change][ii]);
                }
                putchar(0x55);
                //多发一个数据确认一组结束
                if (change==1)
                {
                    change=0;
                }
                else
                {
                    change=1;
                }
            }
        }
        if (over==0)
            mm ++;
        if (over==1)
            over=0;
            execute(mm);
    }
    LED=~LED;                // Toggle LED
```

```
TR2 = 1;           //开启定时器
}
```

### 3.3 上位机软件详细设计

上位机软件使用 Qt 开发。Qt 是一个跨平台的 C++ 应用程序开发框架，广泛用于开发 GUI 程序。使用 Qt 开发的软件，相同的代码可以在任何支持的平台上编译与运行，而不需要修改源代码。会自动依平台的不同，表现平台特有的图形界面风格。经过多年发展，Qt 不但拥有了完善的 C++ 图形库，而且近年来的版本逐渐集成了数据库、OpenGL 库、多媒体库、网络、脚本库、XML 库、WebKit 库等等，其核心库也加入了进程间通信、多线程等模块，极大的丰富了 Qt 开发大规模复杂跨平台应用程序的能力。Qt 最大的特色之一是它的信号与槽机制。信号和槽机制是 QT 的核心机制，要精通 QT 编程就必须对信号和槽有所了解。信号和槽是一种高级接口，应用于对象之间的通信，它是 QT 的核心特性，也是 QT 区别于其它工具包的重要地方。信号和槽是 QT 自行定义的一种通信机制，它独立于标准的 C/C++ 语言，因此要正确的处理信号和槽，必须借助一个称为 moc (Meta Object Compiler) 的 QT 工具，该工具是一个 C++ 预处理程序，它为高层次的事件处理自动生成所需要的附加代码。

软件程序流程图如图 3-4 所示。

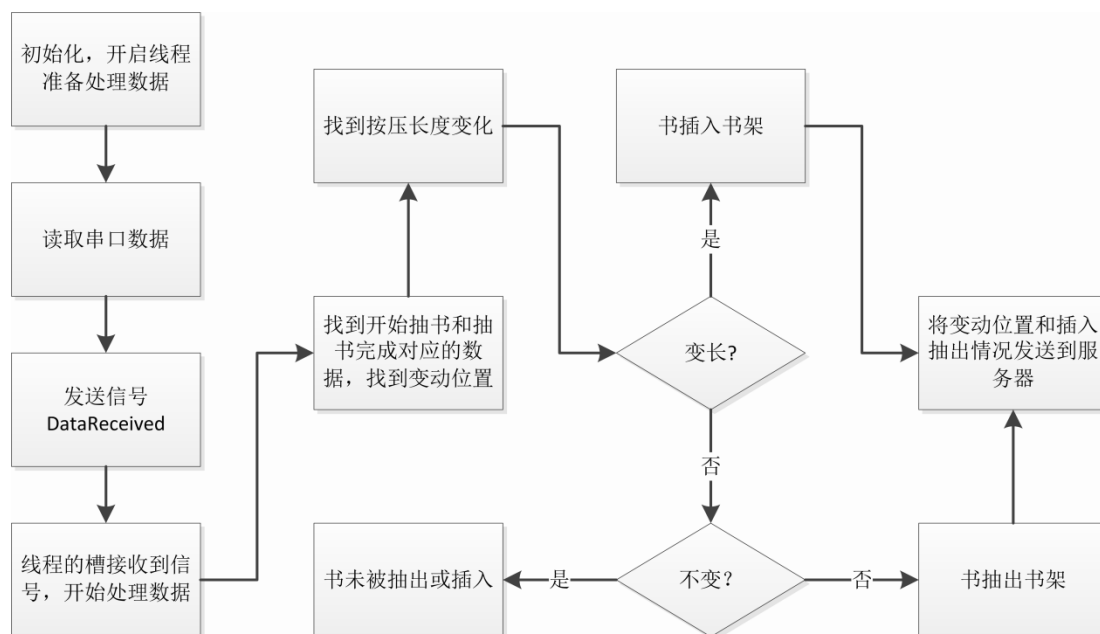


图 3-4 上位机软件程序流程图

根据数据找出书开始变动和变动结束的情况是上位机软件中的核心部分。在书开始和结束变动时，开关可以被分为两段，一段被压下，一段未被压下，不会出现一段被压下中间有开关未被压下的情况。抓住这一特点就可以识别出变动开始和结束时的状态，代码如下：

```
int DealWithDataThread::findStartAndEndState(QList<int> intList)
{
    //find this kind of state : 00001111111111 find number of 0

    int temp=0;
```

```
int counter=0;
int numOf01=0;

QListIterator<int> i(intList);
int temArray[1000];
int cc=-1;
//    int next;
while (i.hasNext())
{
    cc++;
    temArray[cc]=i.next();

//    if(next==0)
//    {
//        counter++;
//    }

//    if(previous==0&&next==1)
//    {
//        numOf01++;
//    }

//    if(numOf01==1)
//    {
//        temp=counter;
//    }

}

for(int j=0;j<intList.size()-1;j++)
{
    if(temArray[j]==0)
    {
        counter++;
    }

    if((temArray[j]==0&&temArray[j+1]==1) || (temArray[j]==1&&temArray[j+1]==0))
    {
        numOf01++; //make sure 01 or 10 just has one
    }
}
```

```
if(numOf01==1)
{
    temp=counter;
}

return temp;
}
```

找到开始和结束状态后，需要根据状态判断书的变动位置和书是抽出还是插入。书抽出时，整层上按压的长度减短，通过比较变动前后被按压的开关数就能识别书是被抽出还是插入，代码如下：

```
void DealWithDataThread::findState(QList<QByteArray> data)
{
    QString mode; //mode=fetch or return
    QString content;
    bool sendReady=false;

    int startState=0;
    int endState=0;
    int length=0;

    int contentStart=1000;
    int contentEnd;

    //this method is to find special states when someone fetch or return a book
    // QString s = QString::number(a, 10);
    QListIterator<QByteArray> i(data);
    int counter=-1;
    int endStateIndex=0;
    while (i.hasNext()) //find start and end state
    {
        counter++;
        QList<int> intList;
        intList=parseArray(i.next());

        if(findStartAndEndState(intList)>0) //find state of 000011111111
        {
            if(startState==0)
            {
                startState=findStartAndEndState(intList);
            }else
            {
                endState=findStartAndEndState(intList);
            }
        }
    }
}
```

```

        endStateIndex=counter; //find endStateIndex ,prepare for next
movement
    }
}

length=endState-startState;

if(startState>0 && endState>0)    //fetch or return movement is finished
{
    if(length>1)
    {
        mode="return";
        sendReady=true;
    }
    else if (length<-1)
    {
        mode="fetch";
        length=0-length;
        sendReady=true;
    }

    i.toFront();
    while (i.hasNext())                //find content
    {
        QList<int> intList;
        intList=parseArray(i.next());
        if(findContentState(intList)<contentStart)
        {
            contentStart=findContentState(intList); //find first 1 index
        }
    }

    contentEnd=contentStart+length;
    content=shelf+QString::number(contentStart,
10)+"to"+QString::number(contentEnd, 10);

    //once states are all found ,byteArray can be clear
    this->data.remove(0,(endStateIndex)*(column+2));
    //留下一个 endstate 的数据
    //prepare for next movement
}

```



```
//this->data can't be data!!!!!!!!!!!!!!  
//much time on this..  
  
//    this->data.append(data.last());  
  
//    i.toFront();  
//    counter=-1;  
//    while (i.hasNext())                //prepare for next movement  
//    {  
//        counter++;  
  
//        if(counter>=endStateIndex)  
//        {  
//            QByteArray next=i.next();  
  
//            this->data.append(next);  
//            QTextStream cout(stdout);  
//            cout<<"_____ "<<endl;  
//            cout<<"data will append : "<<endl;  
//            debugger(next);  
//            cout<<"_____ "<<endl;  
  
//        }else  
//        {  
//            i.next();  
//        }  
  
//    }  
if(sendReady)  
{  
    SendToServer(mode,content);  
    sendReady=false;  
}  
  
startState=0;  
endState=0;  
}  
}
```

### 3.4 服务器软件详细设计

#### 3.4.1 背景技术简介

服务器选用的是 tomcat，运行在 ubuntu 系统中，数据库选用的是 mysql。

Ubuntu 是基于 Debian 分发版和 GNOME 桌面环境，与 Debian 的不同在于它每 6 个月会发布一个

新版本（即每年的四月与十月），每 2 年发布一个 LTS 长期支持版本。Ubuntu 的目标在于为一般用户提供一个最新同时又相当稳定，主要以自由软件建构而成的操作系统。Ubuntu 目前具有庞大的社区力量支持，用户可以方便地从社区获得帮助。

Tomcat 是由 Apache 软件基金会下属的 Jakarta 项目开发的一个 Servlet 容器，按照 Sun Microsystems 提供的技术规范，实现了对 Servlet 和 JavaServer Page (JSP) 的支持，并提供了作为 Web 服务器的一些特有功能，如 Tomcat 管理和控制平台、安全域管理和 Tomcat 阀等。由于 Tomcat 本身也内含了一个 HTTP 服务器，它也可以被视作一个单独的 Web 服务器。

MySQL 原本是一个开放源代码的关系数据库管理系统，原开发者为瑞典的 MySQL AB 公司，该公司于 2008 年被昇阳微系统 (Sun Microsystems) 收购。2009 年，甲骨文公司 (Oracle) 收购昇阳微系统公司，MySQL 成为 Oracle 旗下产品。

MySQL 在过去由于性能高、成本低、可靠性好，已经成为最流行的开源数据库，因此被广泛地应用在 Internet 上的中小型网站中。随着 MySQL 的不断成熟，它也逐渐用于更多大规模网站和应用，比如维基百科、Google 和 Facebook 等网站。非常流行的开源软件组合 LAMP 中的“M”指的就是 MySQL。

SSH(struts + spring + hibernate)框架是目前较流行的一种 Web 应用程序开源框架。框架可分为三层：表现层、业务逻辑层、数据持久层，其中 struts 作为系统的整体基础架构，负责 MVC 的分离，Hibernate 框架对持久层提供支持，Spring 管理整个应用，引入了依赖注入等特性，大大提高了应用的开发效率。

#### 3.4.2 软件开发

服务器软件使用 eclipse IDE for java EE Developers 开发，软件架构如 3-5 所示。

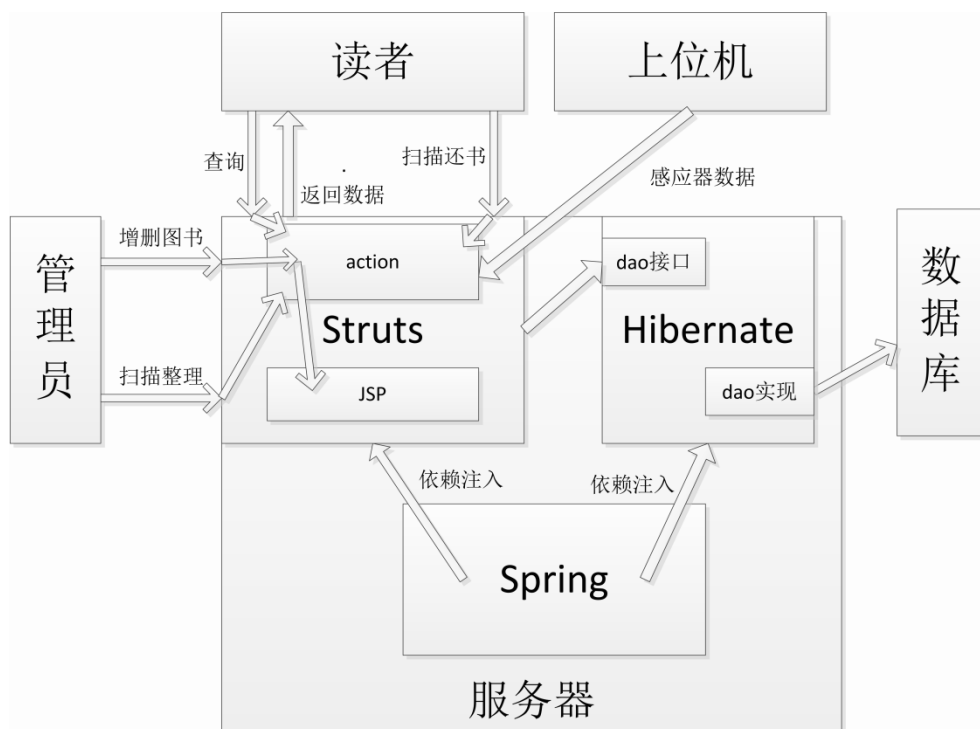


图 3-5 服务器软件架构图

服务器核心功能是在读者抽走一本书或扫描还书时及时更新位置——图书对应关系。抽走一本书时，上位机会发送感应器数据到服务器，数据会被 struts 传递给 hibernate 的 dao(data access object)类处理，然后存入数据库。处理代码如下：

```
@Override
```

```
public void downItem(String location) {
    // TODO Auto-generated method stub
    double se=ParseLocation.parseLocationForAverage(location);
    //find items according to location of shelf
    //then check which se is close
    List<Item> list=getItemsByLocation(location);
    double[] seList=new double[list.size()];
    double temp=0.0;
    for(int i=0;i<list.size();i++){
        if((list.get(i).getSe()-se)>=0)
            seList[i]=list.get(i).getSe()-se;
        else
            seList[i]=se-list.get(i).getSe();
    }
    temp=seList[0];
    int min=0;
    for(int i=0;i<list.size();i++){
        if(seList[i]<temp){
            temp=seList[i];
            min=i;
        }
    }

    itemDao.updateForDownItem(list.get(min));
}
```

插入一本书时，服务器收到感应器数据后会等待读者扫描二维码的数据，当两个都接收到后会  
将数据匹配，然后根据数据更新数据库。核心代码如下：

```
@Override
public void upItem(String location) {
    // called when an item is put into the shelf
    String head=ParseLocation.parseLocationForHead(location);
    Item item=new Item();
    String id="123321";//just create a temp item to save location
    int state=-10;
    String mThumbnailUrl="abc";//just convenient for finding

    while(itemDao.findByMyId(id)!=null){
        id=id+"1";//id can't repeat
        state=state-1;
    }
    item.setId(id);
```

```
item.setState(state);
item.setLocation(location);
item.setThumbnailUrl(mThumbnailUrl);

itemDao.update(item);

//匹配 state
List<Item> matchState=itemDao.findByState(state, head);
if(matchState.size()==2) {
    if(matchState.get(0).getThumbnailUrl().equals("abc")) {
        matchState.get(1)
            .setLocation(matchState.get(0).getLocation());
        matchState.get(1).setState(1);//匹配完 state 恢复成 1
        itemDao.delete(matchState.get(0));
        //itemDao.update(matchState.get(1));
        itemDao.updateForUpItem(matchState.get(1));
    } else
if(matchState.get(1).getThumbnailUrl().equals("abc")) {
        matchState.get(0)
            .setLocation(matchState.get(1).getLocation());

        matchState.get(0).setState(1);//匹配完 state 恢复成 1
        itemDao.delete(matchState.get(1));
        //itemDao.update(matchState.get(0));
        itemDao.updateForUpItem(matchState.get(0));
    }
}

}

@Override
public void upItemJSON(org.json.JSONObject object) {
    //the jsonObject get from return action
    //format {"id":itemId,"location":shelfLocation}
    String id="0";
    try {
        id = object.getString("id");
    } catch (JSONException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    String location = "";
```

```
try {
    location = object.getString("location");
} catch (JSONException e) {
    e.printStackTrace();
}

// String head=ParseLocation.parseLocationForHead(location);
//don't need parse location,because location here is just a head.
int state=-10;
String idTemp="123321";

List<Item> itemList=itemDao.findByState(state, location);

//当这个 state 下没有之前为了保存位置随便设置的 id
//说明这个 state 是这个函数提供的
//为了保证一个临时 state 下只有两个来自不同函数的 item 以供匹配
//state-1
while((((!itemList.contains(itemDao.findById(idTemp))))
    &&itemList.size()==1)
    //>1 说明两个正在配对
    //!=1 但不是 idTemp 说明是这个函数产生的 state
    //state-1 避开
    ||itemList.size()>1) {
    state=state-1;
    itemList=itemDao.findByState(state, location);
    idTemp=idTemp+"1";
}
Item item=itemDao.findById(id);
item.setState(state);
item.setLocation(location);

itemDao.update(item);

//匹配 state
List<Item> matchState=itemDao.findByState(state, location);
if(matchState.size()==2) {
    if(matchState.get(0).getThumbnailUrl().equals("abc")) {
        matchState.get(1)
            .setLocation(matchState.get(0).getLocation());
        matchState.get(1).setState(1);//匹配完 state 恢复成 1
        itemDao.delete(matchState.get(0));
        //itemDao.update(matchState.get(1));
        itemDao.updateForUpItem(matchState.get(1));
    }
}
```



```

    }else
    if(matchState.get(1).getThumbnailUrl().equals("abc")){
        matchState.get(0)
        .setLocation(matchState.get(1).getLocation());

        matchState.get(0).setState(1);//匹配完 state 恢复成 1
        itemDao.delete(matchState.get(1));
        //itemDao.update(matchState.get(0));
        itemDao.updateForUpItem(matchState.get(0));
    }
}
}

```

### 3.5 安卓软件详细设计

Android Studio 是一个 Android 开发环境，基于 IntelliJ IDEA。类似 Eclipse ADT，Android Studio 提供了集成的 Android 开发工具用于开发和调试。在 IDEA 的基础上，Android Studio 提供：

- (1) 基于 Gradle 的构建支持；
- (2) Android 专属的重构和快速修复；
- (3) 提示工具以捕获性能、可用性、版本兼容性问题；
- (4) 支持 ProGuard 和应用签名；
- (5) 基于模板的向导来生成常用的 Android 应用设计和组件；
- (6) 功能强大的布局编辑器，可以让你拖拉 UI 控件并进行效果预览。

安卓软件使用 android studio 编写，运行的版本是 android4.0 及以上，最终在三星和小米两款主流安卓手机上调试通过。软件架构图如图 3-6 所示。

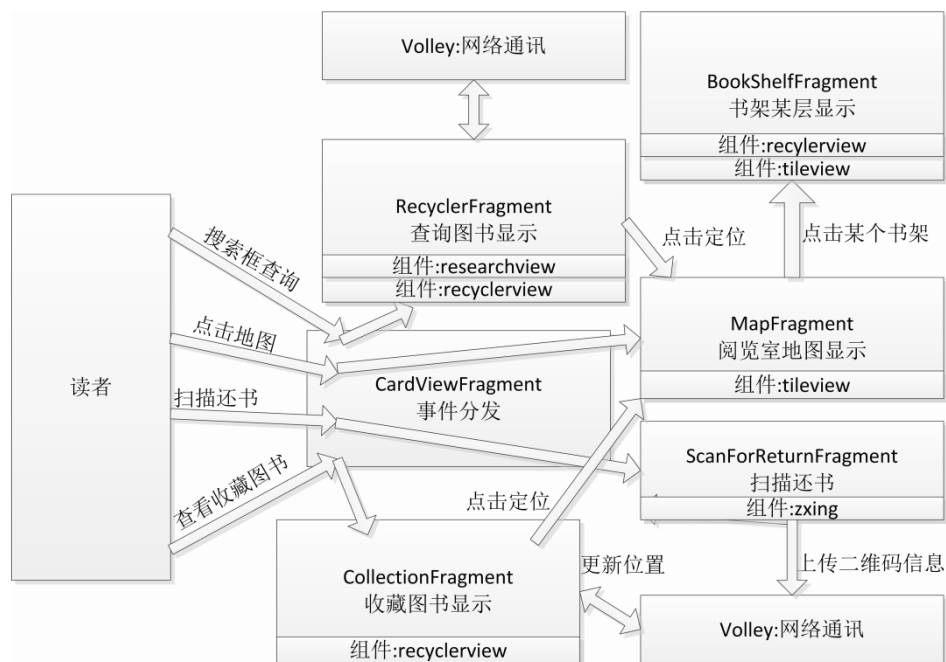


图 3-6 安卓软件架构图

软件采用了模块化编程思想，大部分的功能都在 fragment 中实现，代码的可维护性和可移植性强。同时，软件大量使用成熟的开源组件和官方内置组件编写，大大提高了软件开发的效率和性能。

软件的核心是图书信息，可以说对软件的操作就是图书信息的流动过程：查询图书是从服务器获得图书信息；收藏图书是将图书信息存储到本地；地图显示是将图书信息中的位置信息可视化显示。软件中图书的数据模型代码如下：

```
public class item {  
    private String author;  
    private String thumbnailUrl;  
    private String title;  
    private String location;  
    private boolean collected=false;  
    private String id="0";  
  
    public int getState() {  
        return state;  
    }  
  
    public void setState(int state) {  
        this.state = state;  
    }  
  
    private int state=0;  
  
    public item() {  
    }  
  
    public item(String id, String location, String title, String thumbnailUrl, String author) {  
        this.id = id;  
        this.location = location;  
        this.title = title;  
        this.thumbnailUrl = thumbnailUrl;  
        this.author = author;  
    }  
  
    public String getId() {  
        return id;  
    }  
  
    public void setId(String id) {  
        this.id = id;  
    }  
}
```

```
public boolean isCollected() {  
    return collected;  
}  
  
public void setCollected(boolean collected) {  
    this.collected = collected;  
}  
  
public String getAuthor() {  
    return author;  
}  
  
public void setAuthor(String author) {  
    this.author = author;  
}  
  
public String getThumbnailUrl() {  
    return thumbnailUrl;  
}  
  
public void setThumbnailUrl(String thumbnailUrl) {  
    this.thumbnailUrl = thumbnailUrl;  
}  
  
public String getTitle() {  
    return title;  
}  
  
public void setTitle(String title) {  
    this.title = title;  
}  
  
public String getLocation() {  
    return location;  
}  
  
public void setLocation(String location) {  
    this.location = location;  
}  
}
```

查询图书时，从服务器获得图书信息的代码如下：

```
private void getData(String url, String tag,
```

```
        final List<item> itemList,
        final android.support.v7.widget.RecyclerView.Adapter
adapter)
    //will be called in MainActivity NewIntent
    {
        //parse url to change the get to post,because content may contain Chinese
        itemList.clear();
        final String key = url.substring(url.indexOf("=") + 1, url.indexOf("&"));
        final String content = url.substring(url.lastIndexOf("=") + 1, url.length());

        final Map<String, String> map = new HashMap<String, String>();
        map.put("key", key);
        map.put("content", content);

        final JSONObject Jb = new JSONObject();
        try {
            Jb.put("key", key);
            Jb.put("content", content);
        } catch (JSONException e) {
            e.printStackTrace();
        }

        url = url.substring(0, url.indexOf("?"));
        Log.i("-data->", "url " + url + "   key " + key + "   content " + content);
        //this is a simple parse,so
        //don't change the url in get data
        //
        //

        Log.i("-data->", "get data is called");
        //1  getdata from url
        pDialog.show();

        StringRequest itemReq = new StringRequest(Request.Method.POST, url, new
Response.Listener<String>() {
            @Override
            public void onResponse(String responseString) {
                hidePDialog();
                JSONArray response = null;
                try {
                    response = new JSONArray(responseString);
```

```
    } catch (JSONException e) {  
        e.printStackTrace();  
    }  
    for (int i = 0; i < response.length(); i++) {  
        JSONObject obj = null;  
        try {  
            obj = response.getJSONObject(i);  
        } catch (JSONException e) {  
            e.printStackTrace();  
        }  
        item mitem = new item();  
  
        try {  
            mitem.setAuthor(URLDecoder.decode(obj.getString("author"),  
"UTF-8"));  
  
        } catch (UnsupportedEncodingException e) {  
            e.printStackTrace();  
        } catch (JSONException e) {  
            e.printStackTrace();  
        }  
        try {  
            mitem.setTitle(URLDecoder.decode(obj.getString("title"),  
"UTF-8"));  
  
        } catch (UnsupportedEncodingException e) {  
            e.printStackTrace();  
        } catch (JSONException e) {  
            e.printStackTrace();  
        }  
        try {  
            mitem.setThumbnailUrl(obj.getString("thumbnailUrl"));  
        } catch (JSONException e) {  
            e.printStackTrace();  
        }  
        //    Log.i("images",mitem.getThumbnailUrl());  
        try {  
            mitem.setId(obj.getString("id"));  
        } catch (JSONException e) {  
            e.printStackTrace();  
        }  
        try {  
            mitem.setLocation(obj.getString("location"));  
        } catch (JSONException e) {  
            e.printStackTrace();  
        }
```



```

    }

    try {
        mitem.setState(obj.getInt("state"));
    } catch (JSONException e) {
        e.printStackTrace();
    }

    //request for data
    //data must have these feature: Json "image" "title" "author"
    //      if (obj.getString("id") != null)

    Log.i("-data->", "item " + mitem.getTitle() + " " +
mitem.getAuthor());

    itemList.add(mitem);
    // Parsing json
}
adapter.notifyDataSetChanged();
}
}, new Response.ErrorListener() {
    @Override
    public void onErrorResponse(VolleyError error) {
        if (error != null)
            VolleyLog.d(TAG, "Error: " + error.getMessage());
    }
}
){
    @Override
    protected Map<String, String> getParams() throws AuthFailureError {
        Map<String, String> params = new HashMap<String, String>();
        params.put("key", key);
        params.put("content", content);
        //post a map.
        return params;
    }
};
itemReq.setTag(tag);
ExampleApplication.getInstance().addToRequestQueue(itemReq, tag);
//1
}

```

获得图书信息后，如果读者点击定位图标，图书位置信息会被传递给 MapFragment 进行可视化显示；如果点击收藏，图书信息会被保存到本地，保存图书信息代码如下：

```

static public boolean saveSharePreference(String fileName, item mlItem) {
    boolean flag = false;

```

```
SharedPreferences preferences = context.getSharedPreferences(  
    fileName, Context.MODE_APPEND);  
SharedPreferences.Editor editor = preferences.edit();  
  
editor.putBoolean(mltem.getId() + "Collected", mltem.isCollected());  
editor.putString(mltem.getId() + "Author", mltem.getAuthor());  
editor.putString(mltem.getId() + "ThumbnailUrl", mltem.getThumbnailUrl());  
editor.putString(mltem.getId() + "Title", mltem.getTitle());  
editor.putString(mltem.getId() + "Location", mltem.getLocation());  
editor.putString(mltem.getId() + "id", mltem.getId());  
editor.putInt(mltem.getId() + "State", mltem.getState());  
// editor.putString(mltem.getId() + "State", mltem.getState());  
  
flag = editor.commit();  
return flag;  
}
```

## 4. 系统开发与调试

### 4.1 传感节点搭建

将设计好的 pcb 文件交付厂家印刷，然后焊接好开关等元器件。由于开关无法侧按，需要套一个外壳，让开关头部变为圆弧状。组装好传感器后，将传感器跟微处理器连接，再将微处理器连接电脑，这样传感器节点就可以工作了。如图 4-1 所示。

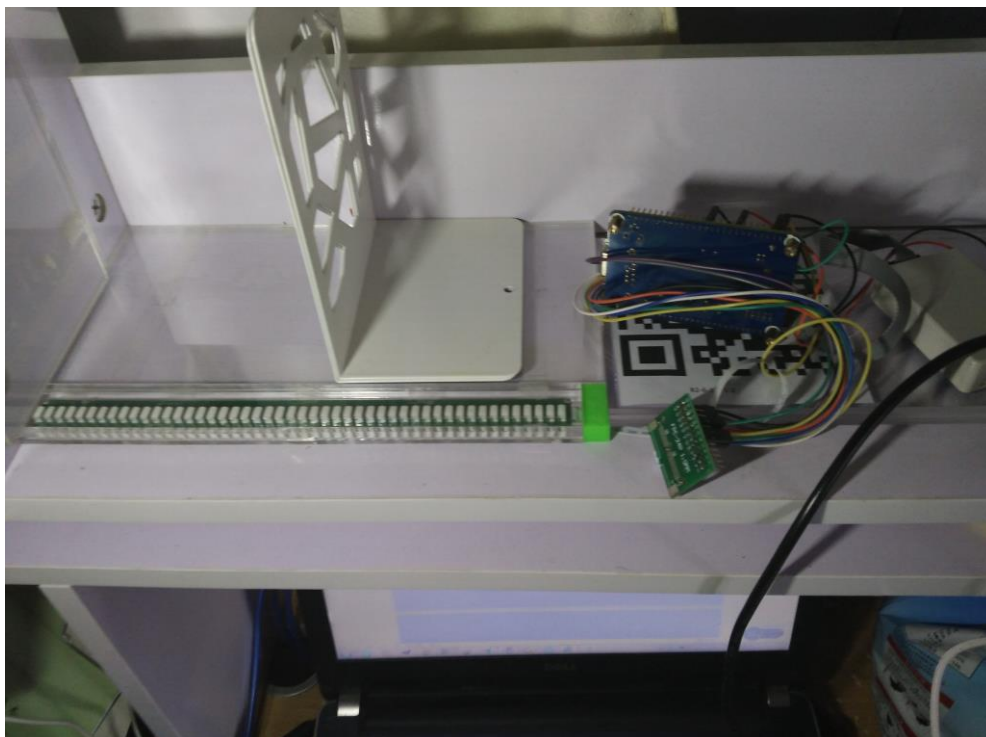


图 4-1 传感节点实物图

#### 4.2 上位机软件制作

使用 Qt 编写完程序后，程序文件结构图如图 4-2 所示：

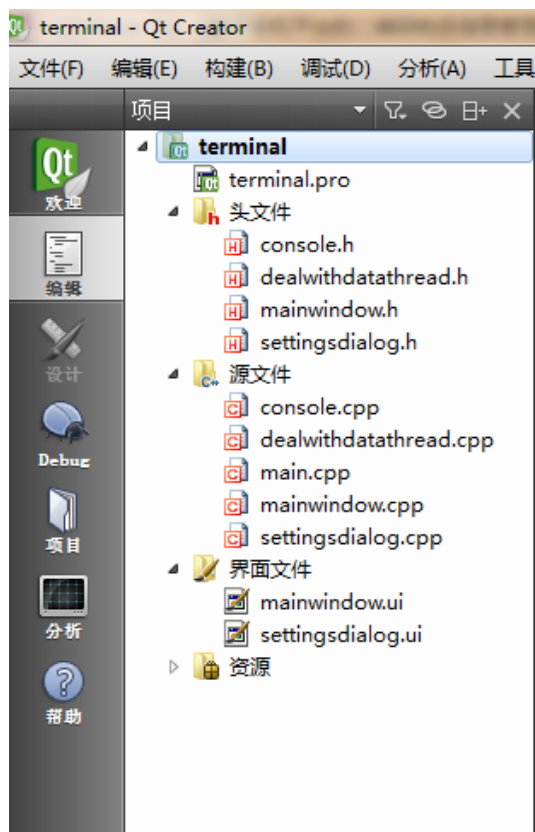


图 4-2 上位机软件程序结构图

点击运行，程序启动，可以看到如图 4-3 所示的界面。

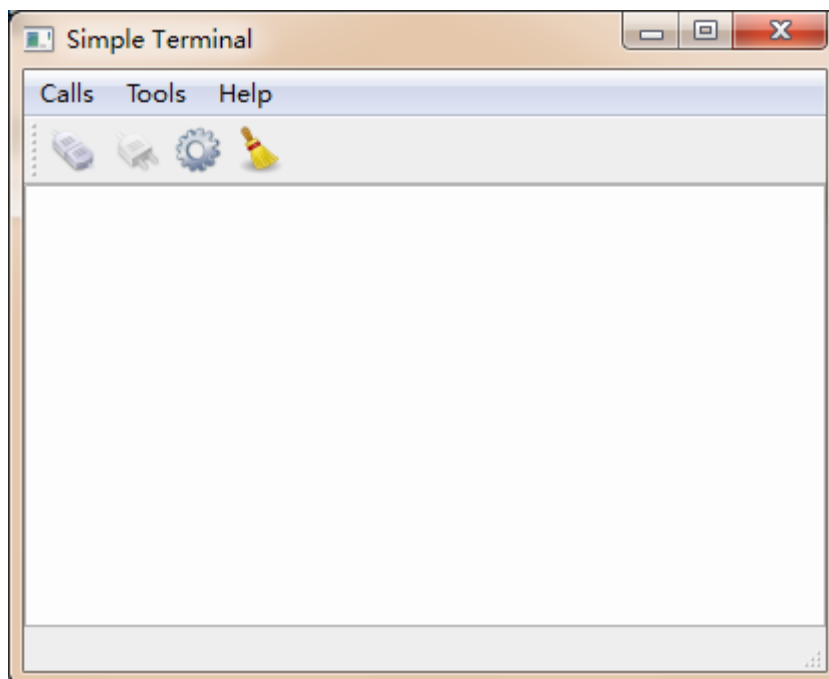


图 4-3 上位机软件主界面

点击设置图标，会出现如图 4-4 所示串口设置界面

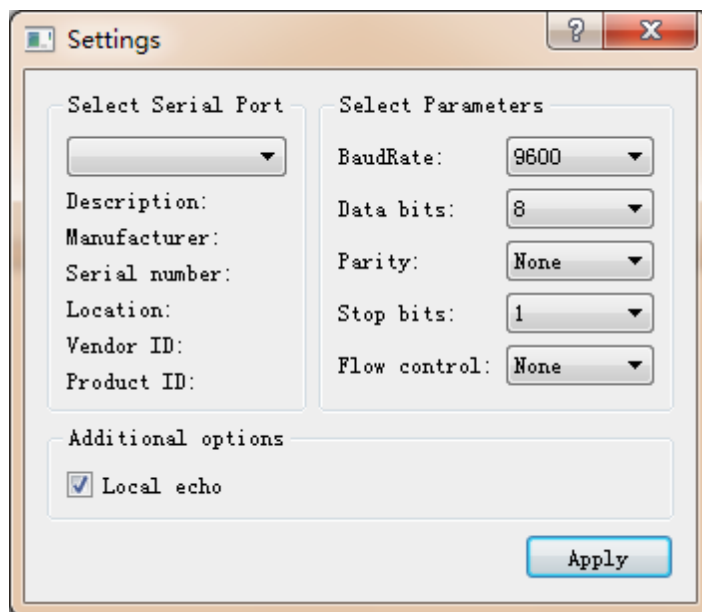


图 4-4 串口设置界面

至此说明软件可以正常运行。由于软件依赖的一些 dll 库没有包含在内，软件无法在其他未安装 Qt 的 windows 系统上运行，还需要找出软件依赖的 dll 库，然后进行打包发布。经测试，打包后软件可以在任何 windows7/8 系统上运行。

### 4.3 服务器软件制作

服务器软件运行在 ubuntu 系统中，首先需要搭建一个装载 ubuntu 系统的服务器。阿里云是阿里巴巴集团旗下公司，提供云服务器 ECS 等产品服务，本课题租用了一台阿里云服务器，服务器上装载有 ubuntu 系统，如图 4-5 所示。



图 4-5 云服务器管理界面

服务器可以远程登录，登陆后在一个命令行工具中可对服务器操作，如图 4-6 所示。

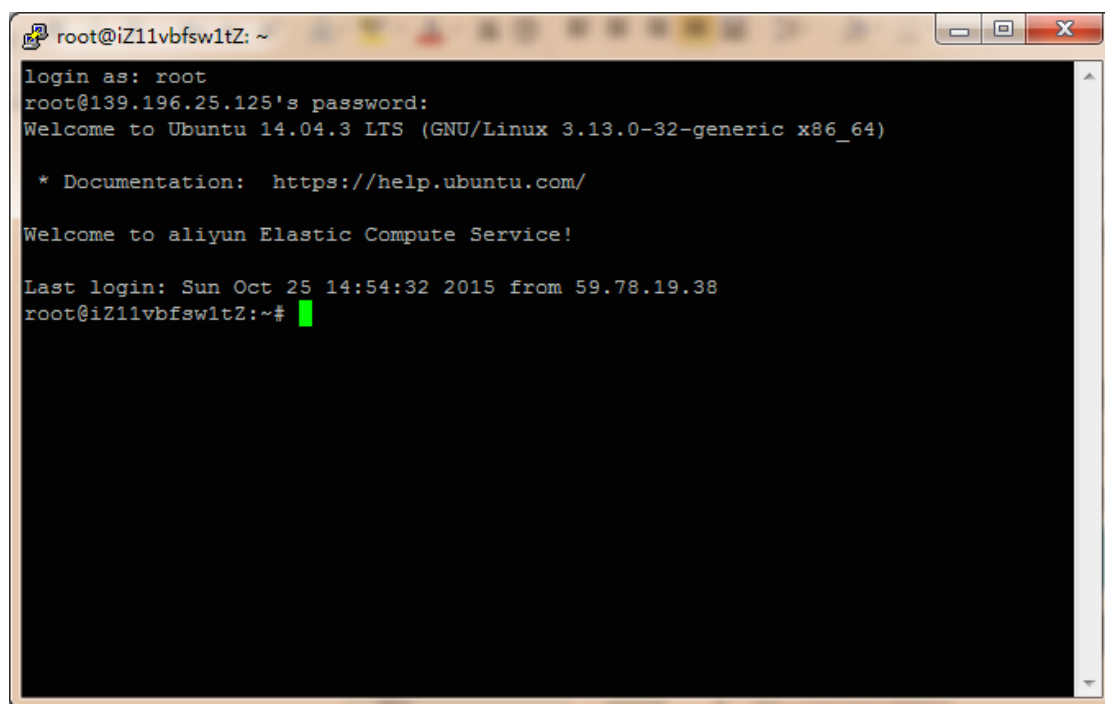


图 4-6 远程登录服务器界面

在服务器上配置好 tomcat 服务器、mysql 数据库等软件后就可以上传编写好的服务器软件了。

服务器软件使用 eclipseEE 编写，为了调试的方便，可以先在本地安装一个 windows 系统下的 tomcat 服务器，在本地调试好之后再上传至云服务器。编写好后软件的文件结构图如图 4-7 所示

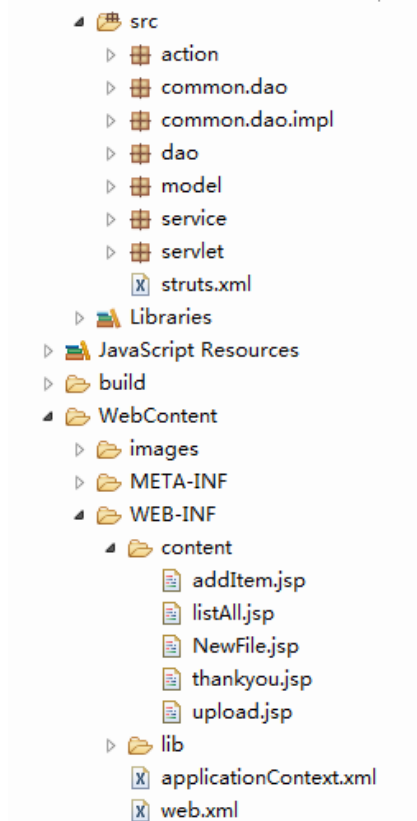


图 4-7 服务器软件文件结构图



调试通过后，将项目导出为 war 文件，然后进入 tomcat 服务器的管理界面，上传后即可运行，如图 4-8 所示，软件 library 已经运行。

/library	None specified		true	0	<input type="button" value="Start"/> <input type="button" value="Stop"/> <input type="button" value="Reload"/> <input type="button" value="Undeploy"/>
					<input type="button" value="Expire sessions"/> with idle > 30 minutes
/manager	None specified	Tomcat Manager Application	true	1	<input type="button" value="Start"/> <input type="button" value="Stop"/> <input type="button" value="Reload"/> <input type="button" value="Undeploy"/>
					<input type="button" value="Expire sessions"/> with idle > 30 minutes

**Deploy**

Deploy directory or WAR file located on server

Context Path (required):

XML Configuration file URL:

WAR or Directory URL:

**WAR file to deploy**

Select WAR file to upload  未选择任何文件

图 4-8 服务器软件管理界面

#### 4.4 安卓软件制作

使用 Android studio 编写完软件后的文件结构如图 4-9 所示。

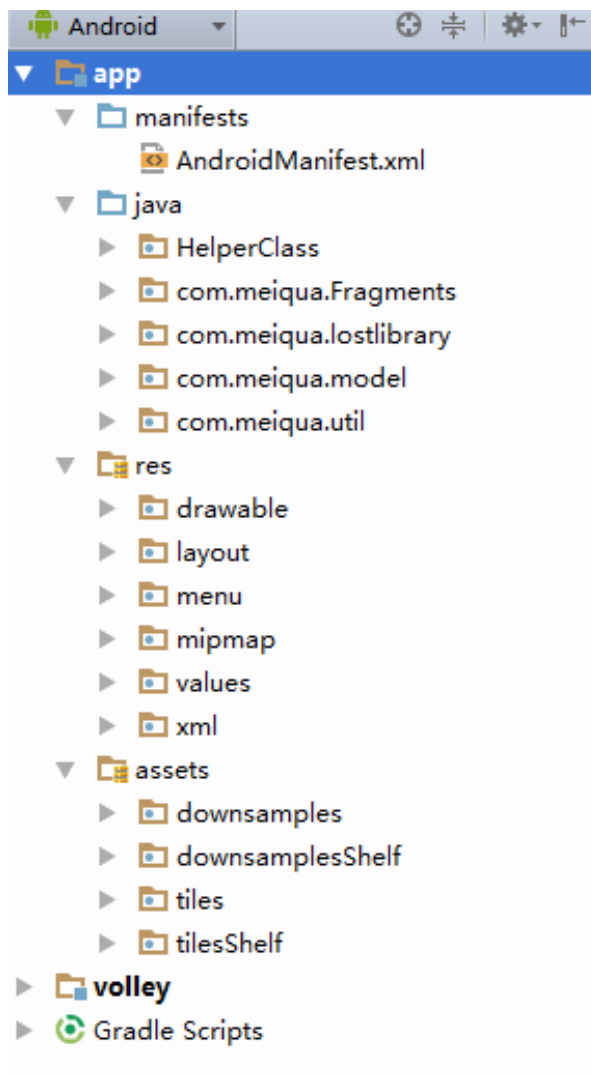


图 4-9 安卓软件文件结构图

调试完成后，软件需要签名发布，点击 build 菜单下的 generate signed apk，之后按提示操作即可完成发布，得到可以运行在安卓手机上的 apk 文件。为了供读者下载，可以将 apk 文件上传到服务器，将下载地址制作成二维码。读者扫描二维码即可开始下载。二维码如图 4-10 所示。



图 4-10 安卓软件下载二维码

## 5. 系统测试

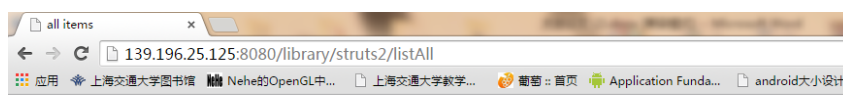
系统搭建完成后，本课题从各方面对系统做了测试，结果显示系统可以正常运行。本课题选用了 6 本书做测试，初始情况下图书信息如图 5-1 所示。

all items						
9787040385151	大学物理学 (上册)	毛骏健 (编者), 顾壮 (编者)	http://139.196.25.125:8080/library/images/9787040385151.jpg	B2-6-1-1-1-2-0to1	-1	<a href="#">delete</a>
9787040385755	大学物理学 (下册)	毛骏健 (编者), 顾壮 (编者)	http://139.196.25.125:8080/library/images/9787040385755.jpg	B2-6-1-1-1-2	-1	<a href="#">delete</a>
9787121087769	Ubuntu Server 最佳方案	冷昱华	http://139.196.25.125:8080/library/images/9787121087769.jpg	B2-6-1-1-1-2-0to6	-1	<a href="#">delete</a>
9787121122033	现代控制工程 (第5版)	尾形克彦 (Katsuhiko Ogata)	http://139.196.25.125:8080/library/images/9787121122033.jpg	B2-6-1-1-1-2-0to7	-1	<a href="#">delete</a>
9787302085997	c++程序设计	谭浩强	http://139.196.25.125:8080/library/images/9787302085997.jpg	B2-6-1-1-1-2-0to4	-1	<a href="#">delete</a>
9787313076144	数学物理方法	上海交通大学数学系	http://139.196.25.125:8080/library/images/9787313076144.jpg	B2-6-1-1-1-2-0to1	-1	<a href="#">delete</a>

图 5-1 初始情况下图书信息图

第一列是图书 ID，第二列是图书标题，第三列是作者，第四列是图书封面的图片地址，第五列是图书位置，第六列是图书状态。当书未上架时，状态值为-1；当书被整理上架时，状态值为 1。

初始情况下书未上架，馆员需要将图书上架。上架的方法很简单，将书插入书架，然后扫描书和上的二维码即可。上架完成后的图书信息如图 5-2 所示。



all items					
9787040385151	大学物理学 (上册)	毛骏健 (编者), 顾壮 (编者)	http://139.196.25.125:8080/library/images/9787040385151.jpg	B2-6-1-1-1-2-0to4	1 <a href="#">delete</a>
9787040385755	大学物理学 (下册)	毛骏健 (编者), 顾壮 (编者)	http://139.196.25.125:8080/library/images/9787040385755.jpg	B2-6-1-1-1-2-4to6	1 <a href="#">delete</a>
9787121087769	Ubuntu Server 最佳方案	冷罡华	http://139.196.25.125:8080/library/images/9787121087769.jpg	B2-6-1-1-1-2-7to13	1 <a href="#">delete</a>
9787121122033	现代控制工程 (第5版)	尾形克彦 (Katsuhiko Ogata)	http://139.196.25.125:8080/library/images/9787121122033.jpg	B2-6-1-1-1-2-13to21	1 <a href="#">delete</a>
9787302085997	c++程序设计	谭浩强	http://139.196.25.125:8080/library/images/9787302085997.jpg	B2-6-1-1-1-2-23to26	1 <a href="#">delete</a>
9787313076144	数学物理方法	上海交通大学数学系	http://139.196.25.125:8080/library/images/9787313076144.jpg	B2-6-1-1-1-2-25to28	1 <a href="#">delete</a>

图 5-2 上架后图书信息图

读者下载并安装好 app 后，打开 app，点击搜索按钮，以搜索物理为例，会看到图书列表的界面。之后点击定位按钮，软件会跳转到地图界面，如图 5-3 所示。

点击地图上的定位图标，软件会进入图书所在书架，就如同读者进入真实的图书馆一样，如图 5-4 所示。本次测试的 6 本书都是位于最左边书架的第二层，下方显示的是该层的所有图书。点击其它层相当于搜索其它层的图书，不过在本次测试中其它层没有图书。



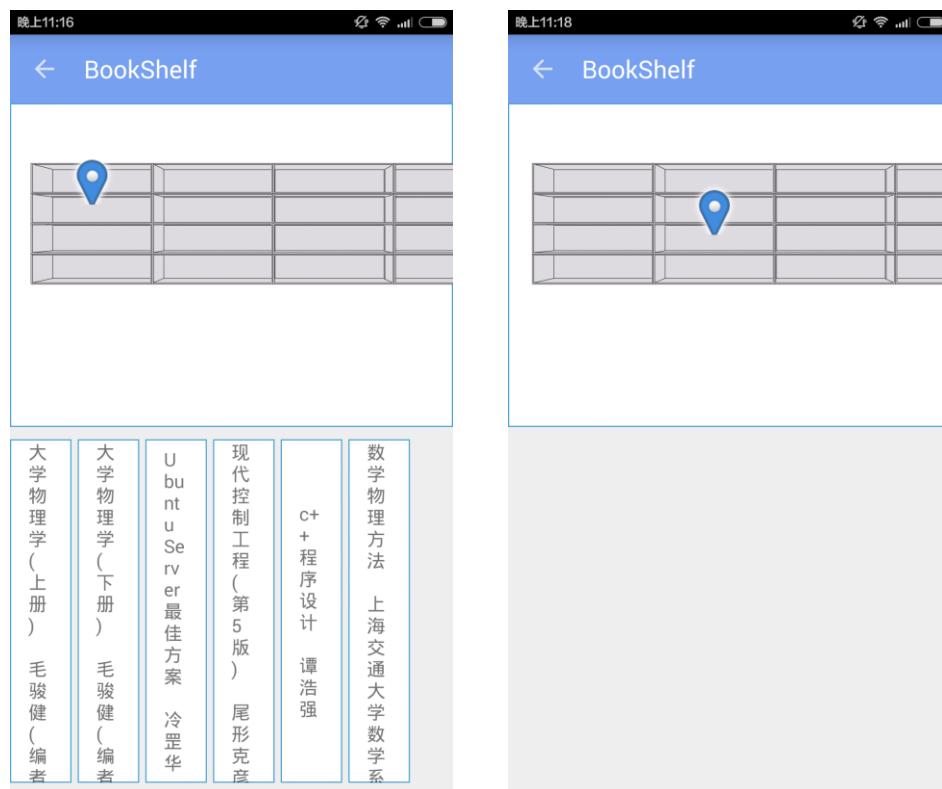


图 5-3 安卓软件主要界面图

当读者拿走一本书时，以拿走《数学物理方程》为例，系统将根据变动位置自动识别出拿走的是哪本书，此时数据库中该图书状态变为-1，如图 5-4 所示。

all items

9787040385151	大学物理学 (上册)	毛骏健 (编者), 顾社 (编者)	http://139.196.25.125:8080/library/images/9787040385151.jpg	B2-6-1-1-1-2-0to4	1	delete
9787040385755	大学物理学 (下册)	毛骏健 (编者), 顾社 (编者)	http://139.196.25.125:8080/library/images/9787040385755.jpg	B2-6-1-1-1-2-4to6	1	delete
9787121087769	Ubuntu Server 最佳方案	冷罡华	http://139.196.25.125:8080/library/images/9787121087769.jpg	B2-6-1-1-1-2-7to13	1	delete
9787121122033	现代控制工程 (第5版)	尾形克彦 (Katsuhiko Ogata)	http://139.196.25.125:8080/library/images/9787121122033.jpg	B2-6-1-1-1-2-13to21	1	delete
9787302085997	C++ 程序设计	谭浩强	http://139.196.25.125:8080/library/images/9787302085997.jpg	B2-6-1-1-1-2-23to26	1	delete
9787313076144	数学物理方法	上海交通大学数学系	http://139.196.25.125:8080/library/images/9787313076144.jpg	B2-6-1-1-1-2-25to28	-1	delete

图 5-4 抽走书后图书信息图

再次搜索物理，并点击《数学物理方程》的定位按钮，软件同样会跳到地图界面，但会提示图书已被取走，此时地图上显示的是图书被取走前的位置。点击进入书架，可以看到该层书架已没有《数学物理方法》。如图 5-5 所示。

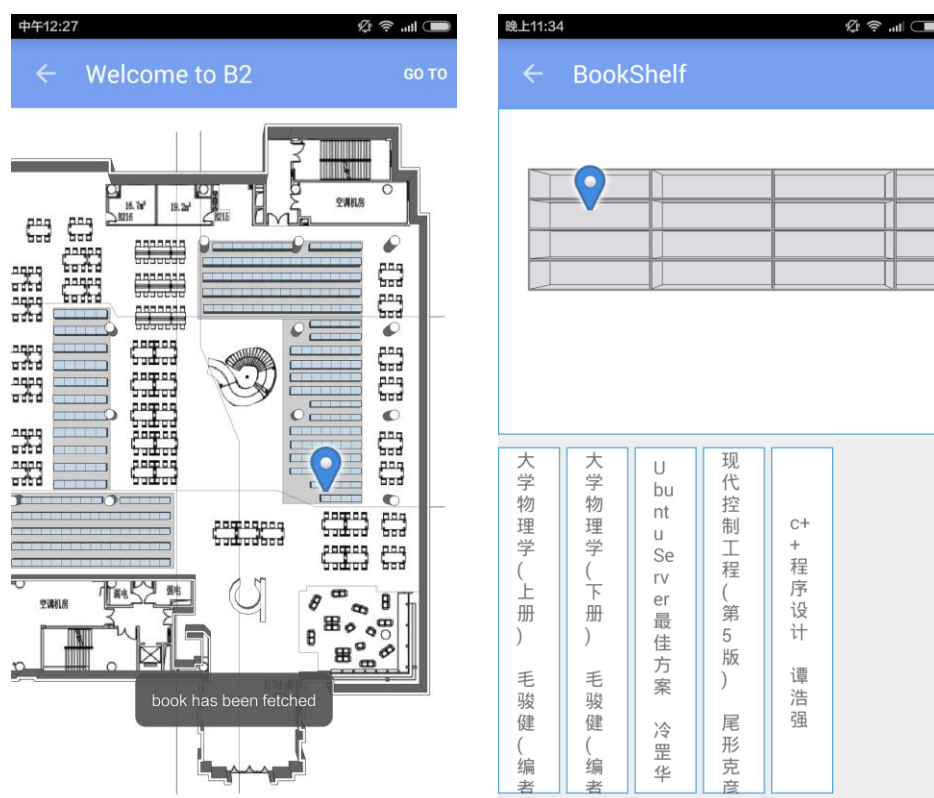


图 5-5 抽走一本书后安卓软件虚拟地图界面图

之后如果读者或馆员将该书插入书架，只需要用手机扫描下书和书架上二维码，该书就相当于被整理好了。测试结果表明系统简单可靠，可以满足定位的要求。

## 6. 总结

图书定位能为读者找书和馆员整理节约大量的时间，提高读者借阅效率和馆员的工作效率，实现图书馆智能化管理，提高图书馆的服务水平，是现代图书馆智能化发展的热门研究项目之一。

本课题研发出一个全新的定位系统，采用独特的定位原理，不仅能够即时对图书位置进行监测，还能指出乱架图书的位置，提醒馆员前去整理。相对于已有的 RFID 图书定位系统而言，本系统具有能耗低，精度高的特点，而且成本大大降低，具备在中小型图书馆推广的条件。

本课题的主要成果如下：

(1) 设计出全新的图书定位方案，详细介绍了方案的原理，对读者借书的各种情况都有详细的应对方法；

(2) 设计并制作出由开关组成的位置传感器和提供推力的机械装置，结合单片机组成了传感节点，对传感器设计需要考虑的因素进行了分析；

(3) 发出上位机软件，介绍了上位机软件的程序流程和主要功能；

(5) 开发出服务器软件，并将软件加载到云服务器运行；

(6) 对服务器的功能进行了介绍，开发出根据位置变动来更新图书——位置对应关系的核心算法；

(7) 开发安卓软件，软件具有的功能有：搜索图书；将图书位置显示在地图上；点击地图上的书架搜索该书架上的图书、收藏图书、扫描还书。

由于时间和资金的限制，本系统只具有演示的功能，进一步的研究工作如下：

- (1) 为了进一步降低成本,传感器的开关需要定制;
- (2) 根据位置变动更新图书——位置对应关系的算法需要改进,以应对各种干扰和特殊情况;
- (3) 为了能够投入实际使用,数据库需要跟图书馆主数据库对接。

## 7. 致谢

一年的大创一眨眼就过去了,在此谨向关心、帮助过我们的老师和同学致以深深的谢意。

首先,要特别感谢我们的指导老师赵春宇教授。最初我们找到赵老师几乎什么也不会,仅仅有个创新的点子。赵老师引导我们详细地考虑了各种复杂的情况,告诉我们关于单片机、传感器的各种知识;在画电路板时,赵老师给了我们很多建议,为后来的制作过程省了很多时间;在硬件调试遇到困难时帮助我找出问题所在。正是在赵老师的指导下,我们才能够顺利地将点子变成了现实。

其次,要感谢图书馆的余晓蔚老师和孙翌老师。本项目最初的想法是在我们在图书馆无法找到书时产生的,当时的我们比见到赵老师时更加无知。我们向余老师表达了希望用图书定位解决这个问题的想法,还不知道有大量学者已经研究了这个问题,并且已经有了 RFID 图书定位系统。余老师耐心地同我们讨论,并向我们介绍了系统发展部的孙老师指导我技术上的问题。虽然当时由于水平有限并没有做出实物,但已经初步想好了新的定位原理。最后也是在余老师的鼓励下,我们又利用大创的机会进行了深入的研究。

最后,感谢学校大学生创新实践项目给我们提供创新实践的平台!

## 参考文献:

- [1]. 鲍静. RFID 技术在图书馆智能化管理中的应用与研究[J]. 贵图学刊, 2012, (02): 44-46.
- [2]. 张倩. RFID 图书定位技术在图书馆的应用研究[D]. 郑州大学, 2012.
- [3]. 张丽. 基于 RFID 技术的图书馆智能书架系统的研究与设计[D]. 南昌航空大学, 2013.