Software Engineering and Applications 软件工程与应用, 2021, 10(3), 275-283
Published Online June 2021 in Hans. http://www.hanspub.org/journal/sea
https://doi.org/10.12677/sea.2021.103031

Hans汉斯

家用人体成分分析仪APP设计与开发

宦佳美，陈　洋，毛少雄，胡　磊

贵州大学机械工程学院，贵州　贵阳
Email: 654717134@qq.com

收稿日期：2021年4月9日；录用日期：2021年6月2日；发布日期：2021年6月9日

摘　要

为了更好地缓解亚健康人群的状态，人们通常会使用家用人体成分分析仪来管理自己的身体，而移动端APP则成为了一种有效的辅助工具，能够将身体的变化进行可视化数据显示。针对当前APP功能单一的问题，提出一款更满足用户需求的家用人体成分分析仪APP。基于Android和IOS两大平台进行APP的开发，使用物联网技术和GPS定位服务，完成移动端与家用人体成分分析仪设备互联；利用蓝牙技术将测量结果上传至移动端，实现测量数据的读取更加详细，同时提供多种显示模式，可切换用户管理，扩大了用户的数据读取范围，并针对个人的身体状态，提供健康指导。

关键词

家用人体成分分析仪，GPS定位，物联网技术，APP开发

Design and Development of Household Body Composition Analyzer APP

Jiamei Huan, Yang Chen, Shaoxiong Mao, Lei Hu

School of Mechanical Engineering, Guizhou University, Guiyang Guizhou
Email: 654717134@qq.com

Received: Apr. 9th, 2021; accepted: Jun. 2nd, 2021; published: Jun. 9th, 2021

Abstract

In order to better alleviate the state of sub-healthy people, people usually use household body composition analyzers to manage their bodies, and mobile APP has become an effective auxiliary tool that can visualize changes in the body. Aiming at the problem of the single function of the current APP, a household body composition analyzer APP that more satisfies the needs of users is proposed. The app is developed based on Android and IOS platforms. The Internet of Things tech-

文章引用: 宦佳美, 陈洋, 毛少雄, 胡磊. 家用人体成分分析仪 APP 设计与开发[J]. 软件工程与应用, 2021, 10(3): 275-283. DOI: 10.12677/sea.2021.103031

nology and GPS positioning service are used to complete the interconnection between the mobile terminal and the household body composition analyzer; Bluetooth technology is used to upload the measurement results to the mobile terminal to achieve more detailed reading of the measurement data. At the same time, a variety of display modes are provided to switch user management, expand the range of data reading for users, and provide health guidance for the individual's physical state.

Keywords

Household Body Composition Analyzer, GPS Positioning, Internet of Things Technology, APP Development

1. 引言

随着人们健康意识的提高，给家用人体成分分析仪带来了良好的设计契机，而物联网和人工智能技术的冲击，让技术与设备能够更好地互联，使得智能健康测量产品顺势而起[1]。在亚健康人群占比高居不下的今天，身体成分的管理仍然是一个热点话题。大多数人群通过多种途径以保证健康的状态，研究者也意识到相关产品对于用户的重要性，开始转向对健康管理类系统的研究，以便更好地掌握身体健康状态[2]。基于这样的背景下，智能健康测量产品与移动端 APP 结合被广泛应用。将互联网、物联网、大数据技术等手段引入家用人体成分分析仪的 APP 设计开发中，更好地满足用户群体的需要。因此，借助成熟的技术手段，开发一款适合更多人群的 APP，方便多种人群通过管理自己的身体成分来保持健康状态，解决现有系统以偏概全的问题，让系统更具针对性。

当前对于家用人体成分分析仪产品的普及率较低，相关系统的研究更是甚少，部分人群对于这一方面的知识非常欠缺。而在减少亚健康人群的过程中，普及人们对于身体成分管理的认知，保持健康的身体状态非常重要。因此，本文主要基于 Android 和 IOS 平台对家用人体成分分析仪 APP 进行设计开发，将蓝牙技术和 GPS 定位服务应用于 APP 的设计开发中，将软硬件设备互联，提升用户对 APP 的交互体验，为亚健康人群提供良好的健康指标，解决现有硬件设备的不足，更好地满足用户需求，从而提升用户体验。通过本系统的设计开发，不仅能满足更多人群的心理需求，更能提升人们的健康意识，从而有效预防疾病。

2. 现有系统分析

人体成分分析仪分析系统早已受到关注，起初是对人体成分分析仪硬件的实现进行研究[3]。这类人体成分分析仪的体型比较庞大，大多只能单向获得测量身体成分的指标，数据显示复杂，非专业人员不易认知，且使用环境的局限较大，更多应用于医院和保健等公共场所。随着社会的转变和技术的更新，对人体成分分析的研究不断增多，除了单方面对硬件进行研究外，部分学者开始转向对软硬件的结合实现进行探索，并表明了应用于人体健康检测可行性[4]。个别学者意识到系统设计的重要性，基于 Android 系统对人体成分分析仪展开研究，提高了检测数据的准确性[5]。但纵观国内外研究成果对比其他产品来说，对于此方向的研究仍处于起步阶段。无论是硬件的研究，还是软硬结合的探索，都在考虑其测量精度。以功能技术层面来进行研究的居多，未能很好的单纯从移动端的交互行为进行深入研究，缺乏良好

的人机交互体验。因此，专门针对此类产品系统的交互研究变得十分重要。

3. 系统需求分析

3.1. 功能性需求分析

功能需求作为软件开发的主体，主要是探究软件要实现的功能[6]。在软件开发的准备阶段，需对用户需要实现的功能及期望进行了解，从而确定软件开发要达到的目的，并完成具体的用户需求。通过对家用人体成分分析仪的长期使用者进行沟通交流，从而确定了家用人体成分分析仪 APP 需要实现的功能。

1) 测量提醒功能：设定时间，定期监测自己的身体成分变化，及时了解身体健康状况。

2) 数据记录与分析功能：记录每一次测量的数据，根据测量指标的数据进行分析，全面了解自身的身体状况。

3) 健康指导：通过测量指标的分析，针对性的给出指导意见，保持健康的身体。

4) 显示模式切换：针对使用人群的迥异属性，提供不同的显示模式，方便更多的用户群体。

5) 移动端 APP 操作控制互联的硬件设备：绑定手机，可随时查找设备和帮助调控设备。

3.2. 非功能性需求分析

非功能需求作为功能需求的补充，是除功能需求以外的特性，在软件的开发过程中不仅要实现功能需求，还要注重对非功能需求的满足，才能更好地提升用户体验[7]。在本系统的开发中，非功能需求可从三个方面来进行：

1) 界面的可操作性：软件的功能模块设计清晰，可提升用户的交互体验，但其基本操作又需要考虑用户常用软件的习惯性。

2) 界面显示的美观性：界面元素的设计和配色的选择，能够有效的提升用户满意度。本软件在界面设计中采用了互联设备的元素，配色选择和设备契合的颜色，增强软件界面的视觉感受。

3) 安全可靠性：个人的指标数据作为对自己健康状态的掌握途径，而技术的进步，导致很多人开始窃取别人的隐私，导致用户非常反感，因此，在进行 APP 系统开发时，需要保证移动端的安全性，避免各种问题的出现。

4. 系统设计与实现

4.1. 系统架构设计

家用人体成分分析仪 APP 主要从交互层、服务层、数据层来进行设计开发。

4.1.1. 交互层

交互层主要涉及软件的前端设计部分。前端主要部署在两大主流移动平台(Android, iOS)上，软件的整体设计模式采用了抽象工厂模式。Andriod 端开发平台采用了 Google 的官方开发平台 Android Studio 进行开发，大量的使用了 Java 语言，XML 语言以及 JSON 来进行软件具体逻辑以及相关算法的实现。iOS 端开发平台主要采用了 macOS 下的 XCode 开发套件来进行开发，使用了 Apple 官方的 swift 语言来进行开发以及逻辑算法的实现，以及使用了大量的 UIkit 来进行前端 UI 的动画处理。

1) 蓝牙模块的实现

系统的交互主要采用了蓝牙来进行设备的软硬件互联。当前，蓝牙技术已经比较成熟，作为是一种无线技术标准，可以实现固定设备的无线连接，从而实现各个设备之间的短距离数据传输功能[8]。具体

实现的功能有：1、扫描其他蓝牙设备；2、为可配对的蓝牙设备查询可适配的蓝牙装置；3、建立 RFCOMM 通道；4、与其他设备进行数据传输。而 Android 平台提供蓝牙支持，允许 Android 设备与其他设备进行无线数据传输。因此，在软件层主要使用了 Android Studio 支持的 android.bluetooth.BluetoothDevice 和 android.bluetooth.BluetoothAdapter 这两个 JAVA 库。在硬件上采用了类似于 HC05 的低功耗蓝牙模块，负责在硬件上实现传感器数据与软件之间的数据传输。

下面是在 Android studio 中这两个 JAVA 库的使用方法：

```
//在 Android Studio  开启系统的蓝牙功能
import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
BluetoothAdapter HuanBlueTooth = BluetoothAdapter.getDefaultAdapter();
if(!HuanBLueTooth.isEnabled()) //判断设备蓝牙是否开启
{
//如果没有开启就调用 startActivityForResult()方法来提示用户没有打开蓝牙
Intent enableBluetooth = new lntent(HuanBlueTooth.ACTION_REQUEST_ENABLE);
startActivityForResult(enableBlueooth,REQUEST_ENABLE_BT);
}
在手机上打开蓝牙来搜索设备
Set<BlueToothDevice> HuanBoothDevices = HuanBlueTooth.getBondedDevices();
//用一个 Set<BlueToothDevice>  集合来保存设备扫描的蓝牙设备
if(pariedDeveces.size > 0){
for(BlueToothDevice dev:HuanBOothDevices){
String HuanDeviceName = dev.getName(); //  得到蓝牙设备的名字
String HuanDeviceMacAddress =dev.getAddress(); //得到蓝牙设备唯一的 Mac 地址
}
}
当搜索到蓝牙设备的名字和 Mac 地址后就可以进行连接
BlueToothDevice HuanDev = HuanBlueTooth.getRemoteDevice(macAddress);
UUID uuidForBlueToothDevice = UUID.fromString(bluetoothDeviceCode);
try{
hSocket = device.createRfcommScoetToServiceRecord(uuid); //进行连接  如果出错就用 catch 来捕获
}catch(IOException e){
E.printStackTrace();
}
```

2) GPS 定位服务模块的实现

全球定位系统(Global Positioning System, GPS)是一种以人造地球卫星为基础的高精度无线电导航的定位系统，能够提供准确的地理位置[9]。在家用人体成分分析仪的软硬件数据信息交互中，需要使用到 GPS 的定位信息来实现一键查找设备功能。

GPS 的实现也分软件上和硬件上。在软件上主要使用了 LatLng 这个 java 库，通过里面封装好的 API 代码来实现具体需要的 GPS 功能。在硬件上主要使用了类似于 1575R-A 模块来实现具体的 NMEA 信息获取。硬件上获取的信息再通过服务器传递到软件端上，在软件端上实现 NMEA 数据类型的处理函数，

将信息处理成为对用户友好的可视化信息。

下面是在 android studio 中 LatLng 库的使用方法：

```
import com.amap.api.maps.odel.LatLng;//导入库
@Override
public void onLocationChanged(AMapLocation amapLocation){
boolean LocationSwitchOne = ((amapLocation != null) && (mListener != null));
boolean LocationSwitchTwo = ((amapLocation != null) && (amapLocation.getErrorCode() == 0));
if(LocationSwitchOne){
if(LocationSwitchTwo){
if(isFirstTime) //如果是第一次定位
{
mListner.onLocationChanged(amapLocation); //显示系统定位图标
lvHolder.title = “位置：”;
lvHolder.address=  amapLocation.getProvider()+amapLocation.getCity()+amapLocation.getStreet()+amapLocation.getStreetNum();
//上面的 amapLocation.get*()可以得到当前位置的 GPS 信息，返回的数据格式是 String 格式，这样就可以得到当前的 GPS 信息，并且保存到了 lvHolder.address 中
}
}
else {
String errText = “定位不成功”+amapLocation.getErrorCode()+”: ”+amapLocation.getErrorInfo();
Log.e(“Debug: ”+errText);
}
}
}
```

3) 软件交互层

包括注册登录，最新的数据显示及健康指导、测量提醒、测量数据记录与分析、设备查找和显示模式切换功能。其中最新数据显示及健康指导、测量提醒、测量数据记录与分析和设备查找的交互模式以可视化方式呈现。

①最新数据显示及健康指导主要展示最近一次测量结果，根据数据结果分析，提供针对性的健康指导，起到警示作用。

②测量提醒主要是用于提醒家人或自己定期检测自己的身体状况。

③数据记录与分析提供多个用户的使用情况记录，以随时掌控自己的健康状况。

④设备查找主要是采用 GPS 定位服务，为了方便用户寻找设备，及时发现设备状态，以便更好地使用。

4.1.2. 服务层

服务层主要部署在 CentOS7 系统上面，采用了 Node.js 来进行后端的部署，以及异步的方法来应对同时需要处理多个用户请求的情况，在 Node.js 上也采用了 express，http 等模块。

服务端的 node.js 关键代码如图 1：

```
//服务端中的node.js代码摘要
const ws = require("nodejs-websocket");
      const server = ws.createServer(conn => {
        conn.on('text',data=>{
  console.log("连接了一个客户端");
});
conn.on('close',data=>{
  console.log("退出了一个客户端");
});
})
server.listen(8181,()=>{
  console.log('服务器已经在8181端口启动了一个socket服务.')
});
```

Figure 1. Partial realization of node.js-socket
图 1. node.js-socket 部分实现

软件端的 java 关键代码如图 2：

```
//在Android studio中使用Socket进行软件与服务器之间的通信
import java.io.IOException;  //java的输入相关的一些库
import java.io.InputStream;
import java.io.OutputStream;
import jva.net.Socket;  //java的Socket库，里面实现了一些Socket相关的函数
import java.net.UnknownHostException;
import java.net.SimpleDateFormat;
import java.util.Date;  //java的时间库
import java.util.Scanner;  //java获取输入的库
public class Client_Demo extends Thread{
    //定义了一个Socket对象
    Socket socket_demo = null;
    //代码的构造函数
    public Client_Demo(String host,int port){
      try{
        socket_demo = new Socket(this.host,this.port);
      } catch (UnknownHostException e){  //捕获错误,如果出错就抛出错误
        e.printStackTrace();
      } catch (IOException e){
        e.printStackTrace();
      }
    }
   @Override
   public void run(){
     new sendMessThread().start();
     super.run();
     try{
       //读取Sock里的数据
       InputStream s_demo = socket.getInputStream();
       byte[] buff = new byte[1024];
       int len = 0;
        while((len = s.read(buff)) != -1){
       System.out.println(getdate() + "从服务器获取的数据: "+new String(buff,0,len,"UTF-8"));
     } catch (IOException e){
       e.printStackTrace();
     }
   }
```

```
//往socket里面写入数据
class sendMessThread extends Thread{
    @Override
    public void run(){
      super.run();
      Scanner scanner=null; //java标准输入
      OutputStream os = null;
      try{
        scanner_demo = new Scanner(System.in);
        os = socket_demo.getOutputStream();
        String in = "";
        while(!in.equals("exit")){ //当输入 exit 的时候程序终止
          in = scanner.next();
          os.write((""+in).getBytes());
          os.flush();
        }
      } catch (IOException e){
        e.printStackTrace();
      }
      scanner.close();
      try{
        os.close();
      } catch (IOException e){
        e.printStackTrace();
      }
    }
  }
  //获取时间，并且将时间的格式转换一下
  public static String getDate(){
    Date date_demo = new Date();
    SimpleDateFormat format = new SimpleDateFormat("yyyy-MM-dd hh:mm:ss");
    String result_demo = format.format(date);
    return result_demo;
  }
  public static void main(String[] args){
    Client client_demo = new Client('localhost',9090);
    client_demo.start();
  }
}
```

Figure 2. Partial realization of java-socket
图 2. java-socket 部分实现

4.1.3. 数据层

数据层主要采用了 MySql 来进行数据存储。MySql 的服务端口设置在了 3000 端口，与 Node.js 后端服务器来进行数据的交互，并且将数据实时的呈现在前端(client)上面。

User 用户表：包括用户的唯一登录信息 ID，USER_NAME 使用 VARCHAR(32)的类型来保存且不为空，USER_PASSWORD 使用 VARCHAR(32)来保存，CHARSET 属性主要使用 utf-8 编码。

INFO 数据表：可以与 User 进行多表链接来进行查询。INFO__BMI 使用 FLOAT 数据类型，保存了 BMI 的数据；INFO__WATER 使用 FLOAT 数据类型，保存了用户的水分数据信息；INFO_BODY_FAT 使用 DOUBLE 数据类型，保存用户的体脂数据信息；INFO_BONE 使用 INT 数据类型，保存了用户的骨量数据信息；INFO__WEIGHTS 使用 FLOAT 数据类型，保存了用户的体重信息；INFO__MUSCLE_RATE 使

用 DOUBLE 数据类型，保存了用户的肌肉率信息，其他一些指标数据也是通过这样的方式保存至数据库。

4.2. APP 的设计实现

APP 的功能设计梳理为用户个人中心、测量提醒、数据记录与分析、最新数据显示及健康指导、查找设备。将四个功能整合到 APP 的四个页面中，分别为系统首页、数据记录和分析页以及“我”页，软件逻辑框架如图 3 所示。

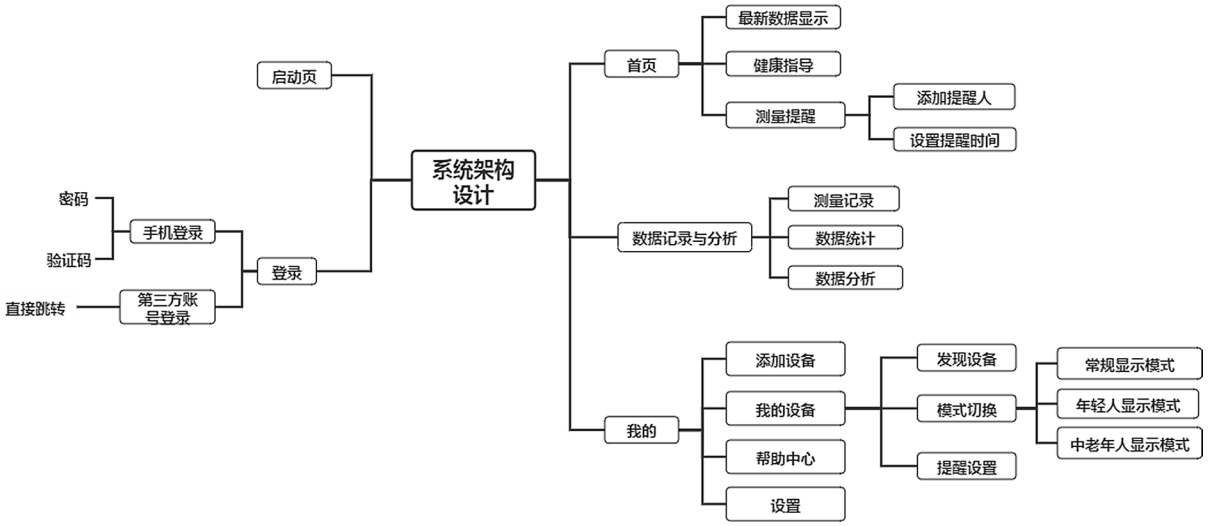

Figure 3. APP architecture design
图 3. APP 架构设计

系统首页界面如图 4 所示。为了更好地使用户管理自己的身体成分来保持健康的生活，本软件的主界面设置了最新数据显示和健康指导。在首页的界面设计中，将首页顶部选择作为当前最新的数据显示，以便实时掌握测量状态。常用参考数据直接显示，其他更多数据滑动显示。首页下端显示健康指导与推荐，主要是根据测量的数据结果分析，提供针对性的健康指导。当用户查询完最新的测量数据信息后，点击上方的提醒按钮可进入测量提醒界面，在测量提醒界面添加提醒人，根据不同用户的使用频率设定提醒时间，可设置多个提醒人。在工作压力较大的环境下，人们通常会忘记做一些想做的事，温馨的提醒可以带给用户温暖。

Figure 4. System home page and expanded page
图 4. 系统首页及展开页

对于测量数据记录与分析界面如图 5 所示，用户的每一次测量将记录在 APP 端，可提供多个用户的详细记录，针对个人数据进行分析，分析结果以下滑动读取，以便随时对比不同时期的数据，及时改善健康状态。

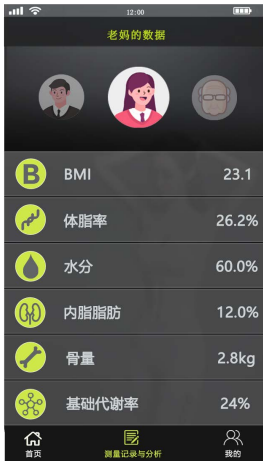

Figure 5. Data recording and analysis page
图 5. 数据记录与分析页面

“我的”页面层级下添加 GPS 定位服务，用于发现设备。通过点击我的设备，进入发现设备页面，可开启一键找设备，设备会以多种方式给出反馈，从而成功地将设备与移动端互联。该界面还提供了三种不同的显示模式，用户可以根据自己的习惯进行选择，具体页面如图 6 所示。

Figure 6. “Personal Center” page
图 6. “个人中心”页

该移动端的主要功能在于，当用户需要测量身体成分时，只需要通过 GPS 定位功能，可开启一键找设备，APP 端即可查寻互联设备的状态，当使用完设备后，可从该端获取详细的数据信息分析，并给出相应的健康指导。并且根据不同年龄段的人群，设计不同的数据显示模式，操作简便，适合不同年龄段的人群，还可通过 APP 端操作来控制互联设备。

4.3. 本系统优势

基于前人研究的基础上，本系统集成了 GPS 定位技术和蓝牙技术，不仅实现了传统家用人体成分分

析仪测量数据分析功能。同时将数据的读取从产品转移到软件上，避免了产品本身功能复杂，导致体型的庞大，还能更好地随时查看自己的身体指标数据。由于使用该设备的人群范围广泛，为了更好地满足不同年龄段的用户，提供了多种显示模式，主要根据年龄段区分，显示重点有所不同，用户可切换自己合适的显示模式，以提升用户体验。更重要的是，本系统使用物联网技术，将系统与设备互联，实现测量数据的可视化，用户可直接在 APP 端查看自己的身体成分指标，并将每一次数据进行记录和分析，以便更好地掌握自己的身体状况，不再是必须靠近设备才可以查看自己的数据，且数据内容还比较单一。同时，还使用了 GPS 定位服务，用户可在需要使用设备时，开启一键找设备，设备便会给出反馈信息，避免浪费时间。

5. 结语

本系统基于 Android 和 IOS 两大主流移动平台对家用人体成分分析仪 APP 进行设计开发。利用蓝牙技术使移动端与设备互联，基于 GPS 定位技术有效获取设备的状况信息。以 MySQL 数据库进行数据储存，将数据实时的呈现在前端，使详细的数据可视化显示，提升数据信息的可读性；界面的设计简洁美观，减少操作的复杂性。该系统的开发主要对系统的人机交互体验进行考量，使得 APP 的实现具有很好的实用性，且交互性强。同时 APP 与设备软硬件的结合，更能满足用户需求，通过移动端 APP 的优势弥补硬件设备的不足，从而达到更好地用户体验效果。

参考文献

[1] 王雄. 医疗物联网实现互联心脏监测[J]. 计算机与网络, 2020, 46(9): 38-39.

[2] 谭艳春, 陈启勇, 刘目磊. 基于物联网+手机 APP 的老年人自我健康管理系统[J]. 电子技术与软件工程, 2017(17): 67-68.

[3] Bae, S.H., Moon, B.S., Lee, W.J., *et al.* (2006) A Chip Design for Body Composition Analyzer. *IEEE* 2006 *Biomedical Circuits and Systems Conference Healthcare Technology*, *BioCAS*, London, 29 November-1 December 2006, 218-221.

[4] 肖晓明, 何为, 贺玉成. 基于生物电阻抗原理人体成分分析仪的设计与研究[J]. 中国医疗设备, 2015, 30(8): 9-13.

[5] 李钊, 单珂, 邓少龙, 高天雷, 刘照阳. 基于 Android 平台的人体成分分析仪的设计与实现[J]. 山东科学, 2015, 28(6): 127-132.

[6] 王雅丹. 关于软件需求分析的研究[J]. 电子技术与软件工程, 2016(12): 83.

[7] 张璇, 李彤, 王旭, 于倩, 郁湧, 朱锐. 可信软件非功能需求形式化表示与可满足分析[J]. 软件学报, 2015, 26(10): 2545-2566.

[8] 吕艺. 蓝牙模块无线通信设计与实现[J]. 科技经济导刊, 2016(8): 31.

[9] 王庆安. 基于 RFID 和 GPS 及 GPRS 的车载物流管理系统研究[J]. 交通企业管理, 2006, 21(8): 52-53.