

基于移动互联网和 Web Service 的智能农业系统^①

潘益婷, 潘修强, 金慧峰, 钱冬云

(浙江工贸职业技术学院 信息与传媒分院, 温州 325002)

通讯作者: 潘修强, E-mail: panxq@163.com

摘要: 基于移动互联网和 Web Service 技术, 采用 ZigBee 技术构建底层 WSN, 利用移动通信网络实现信息传输, 利用 HTTP 协议和 JSON 数据格式进行数据的交换, 实现 Android 客户端和 Web Service 服务端的设计与开发, 面向智能农业环境构建了包含环境指标、历史数据查询、手动控制、系统设置等功能为一体的智能农业 APP 系统. 根据预设农作物生长周期需要的环境参数, 通过 Android 手机进行展示和操作管理, 提高了农业系统的智能化水平, 改善了落后的农业管理方式.

关键词: 智能农业; Android 客户端; Web Service; HTTP 协议; JSON 格式

引用格式: 潘益婷, 潘修强, 金慧峰, 钱冬云. 基于移动互联网和 Web Service 的智能农业系统. 计算机系统应用, 2017, 26(11): 114-117. <http://www.c-s-a.org.cn/1003-3254/6093.html>

Intelligent Agriculture System Based on Mobile Internet and Web Service

PAN Yi-Ting, PAN Xiu-Qiang, JIN Hui-Feng, QIAN Dong-Yun

(College of Information and Communications, Zhejiang Industry & Trade Vocational College, Wenzhou 325002, China)

Abstract: Based on the mobile Internet and Web Service technology, with ZigBee technology to build the underlying WSN, the mobile communication networks to achieve information transmission, HTTP protocol and JSON data format for data exchange, the Client and Web Service server is designed and developed, for the intelligent agricultural environment, including the construction of environmental indicators, historical data query, manual control, system settings and other functions as one of the intelligent agricultural APP system. According to the environmental parameters required for the crop growth cycle, the agricultural management displayed and operated by the Android mobile phone improves the intelligent level of the agricultural system and enhances its backwardness.

Key words: intelligent agriculture; Android client; Web Service; HTTP protocol; JSON format

移动终端的便携性以及附带多种传感器等特点, 使得移动应用达到炙热化程度, 各行各业几乎都推出移动应用, 如淘宝、百度外卖、滴滴出行、铁路 12306 等. 对于如何将先进的传感器技术、移动通信技术和数据处理技术等应用于农业领域, 提高农业系统的智能化管理水平, 国内近几年也有一些相关研究工作. 王恬等^[1]设计了改进的层次聚类算法, 利用信息可视化模型与基于 Java 的 Prefuse 插件包提供了图形化界面展示. 孙想等^[2]设计实现了农业海量信息资源的分

类、存储和访问管理平台. 刘洋等^[3]提出了基于物联网与云计算技术的农业网络平台, 建成数据采集、数字传输、数据分析处理为一体的农业管理体系. 邬群勇等^[4]建立基于 Web Services 的农业空间信息资源共享服务平台. 张佩云等^[5]提出基于 SOA 架构和 SOC 技术实现农业信息资源整合, 设计并构建基于多角色访问的农业信息服务系统. 但是以上研究关于农业系统的移动应用方面阐述较少, 缺乏有实际应用针对性强的移动互联网解决方案. 因此, 如何在 Web Service 和移

① 基金项目: 浙江省科技计划项目 (2016C32103); 浙江省高教教改项目 (jg2015353); 全国高校计算机研究会项目 (2016GHB01012)

收稿时间: 2017-02-21; 修改时间: 2017-03-27; 采用时间: 2017-04-05

动互联网技术支撑下, 建立一个智能农业系统, 对提高农业领域的智能化便携式管理水平具有现实意义.

1 系统概述

我国是一个农业大国, 农业建设面临从传统农业向现代农业转型, 物联网、移动互联网等技术的发展, 为农业信息化提供了一个全新平台. 在农作物生长的各个环节, 需要及时准确掌握光照、CO₂ 浓度、空气温湿度、土壤温湿度等环境指标, 对这些环境数据进行获取、展示、预警, 并能根据阈值自动设置照明、水泵、蜂鸣器等设备开关. 系统功能结构图如图 1 所示. (1) 用户管理. 用户未登陆时仅拥有查看数据的权限, 只有登陆之后才能控制设备开关. 具体功能包括用户通过手机号验证码进行注册, 通过用户名密码进行登陆, 以及修改密码等功能. (2) 环境数据展示. 展示沙盘各传感器的数值 (模拟农业环境), 并实时更新数据, 具体的环境数据有: 光照值、CO₂ 浓度、空气温度、空气湿度、土壤温度、土壤湿度, 当然后续根据需要还可以增加 PM2.5 等其他传感器. (3) 历史数据展示. 各环境数据每秒都会有一个数值, 如果要看过去某个时间段的数据, 将会以折线图的形式展示. (4) 控制设备开关. 比如当土壤湿度不够的时候, 就可以手动打开水泵装置进行灌溉. 这里的设备包括照明灯、水泵、风扇、蜂鸣器, 当然后续根据需要还可以增加其他设备. (5) 系统设置. 设置各环境数据的阈值, 比如设置空气温度的最大值和最小值, 当空气温度不在这个阈值范围内, 自动发送报警信息并触发蜂鸣报警器.

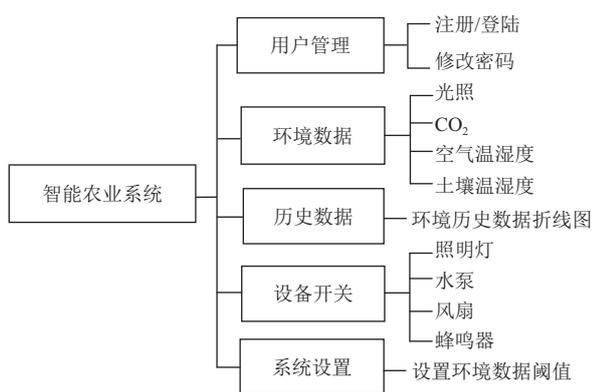


图 1 系统功能结构图

2 系统设计

2.1 系统总体架构设计

系统总体架构图如图 2 所示. 总体架构设计分 3 部分: (1) 沙盘. 装有各种传感器, 模拟农业环境. (2)

服务端. 收集沙盘中各传感器数值以及控制沙盘各设备开关功能, 并封装成 API 供客户端访问. (3) 客户端. Android APP 通过 API 访问服务端, 并向终端用户提供数据展示、控制设备开关等功能.

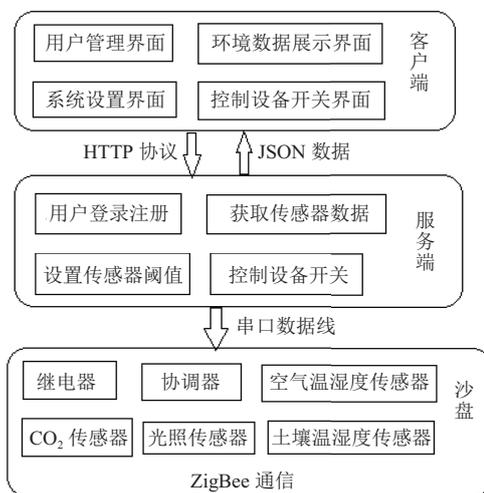


图 2 系统总体架构图

2.2 沙盘设计 (模拟农业环境)

设计一个模拟农作物生长环境的沙盘系统, 通过 ZigBee 协议^[6]连接继电器、协调器、空气温湿度传感器等其他传感器, 以及水泵、风扇等其他设备.

ZigBee 是一种短距离、低功耗的无线通信协议, 与蓝牙相类似. 由于 ZigBee 协议的低功耗、低成本、低速率、近距离、短延、高容量、高安全等特性, 它非常适合应用于传感和控制领域. ZigBee 带有路由的可组网功能, 协调器就是一个单片机, 集成了 ZigBee 模块, 与外部各种传感器以及继电器通信. 传感器和继电器也集成了 ZigBee 模块. 传感器与协调器之间通过 ZigBee 通信. 这样, 整个沙盘系统通过 ZigBee 完成组网, 服务端就可以通过串口数据线的连接获取各传感器的数值以及控制水泵等设备的开关.

沙盘中使用的传感器包括: 光照传感器、CO₂ 传感器、空气温湿度传感器、土壤温湿度传感器, 这些都是农作物生长过程需要监控的环境信息, 当然如果需要还可以添加 PM2.5 传感器、土壤酸碱度传感器等其他传感器. 沙盘中使用到的设备包括: LED 灯、水泵、风扇、蜂鸣器, 这些都是农作物生长过程中需要外界干预控制的, 后续需要的话还可以增加其他设备.

2.3 服务端 Web Service 设计

服务端 Web Service 是连接沙盘和客户端的中间

桥梁. 服务端通过串口数据线与沙盘相连, 将读取沙盘传感器数据和操作沙盘设备开关等的操作都封装在对应 API 中. 客户端与服务端通过网络相连, 客户端通过访问服务端相应 API 从而访问控制沙盘.

服务端专注于对沙盘的访问控制, 并将操作过程封装在 API 中. 客户端专注于界面的展示. 这样的分工合作符合 MVC^[7]思想, 将业务逻辑、数据、界面显示分离, 有助于管理复杂的应用程序, 同时这种分层思想也简化了分组开发.

服务端设计采用 Web Service^[8]是因为本系统服务端无需界面来展示和操作数据, 对数据或设备的展示操作都通过 Android 客户端来实现, 将来也有可能继续扩展使用 pad 或者 IOS 等其他移动客户端, 且 Web Service 是简单的可共同操作的消息收发框架, 具有开放性和跨平台性, 能使运行在不同机器上的不同应用相互交换数据或集成.

2.4 客户端 APP 设计

客户端设计是基于 Android 平台的应用, 其主要功能如图 1 所示. 服务端提供访问控制沙盘的 API 主要有 2 种类型: 一是读取传感器数据, 二是控制设备开关. 下面分别介绍这两种访问过程: (1) 客户端通过网络与服务端连接, 客户端根据服务端提供的 API 向服务端发送 HTTP 请求^[9-10](读取传感器数据), 服务器收到请求后, 将获取到的沙盘传感器数据处理成 JSON 格式^[11-12]数据封装到响应对象中, 客户端接收 JSON 数据并解析, 通过 Android 子线程和 Handler 技术^[13]展示数据. (2) 客户端向服务端发送 HTTP 请求(开关设备), 服务器收到请求后控制沙盘设备的开关并返回结果, 客户端根据服务端返回的结果控制 APP 界面上的 Switch 开关和帧动画.

3 系统核心功能

3.1 环境数据展示

客户端环境数据展示界面如图 3 所示, 即获取空气温度等传感器数据显示在界面上并实时更新.

实现过程如下:

(1) 通过 Fiddler 工具从服务端获取 JSON 数据. Fiddler 工具非常简单实用, 如图 4 和图 5 所示, 只需选择合适的 HTTP 请求 (GET 请求或 POST 请求), 输入正确的服务端提供的 API, 就完成了模拟 HTTP 请求过程和展示 HTTP 响应的结果.



图 3 客户端环境数据界面

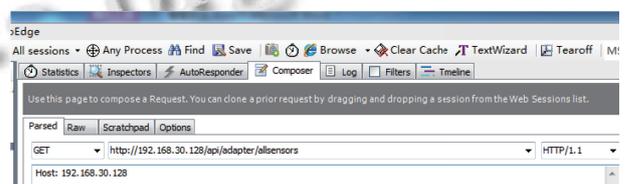


图 4 Fiddler 工具发送 HTTP 请求

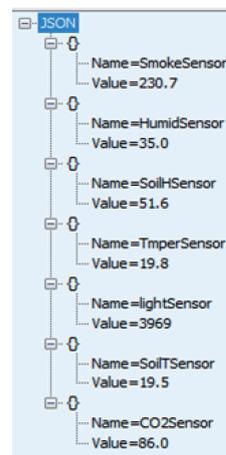


图 5 Fiddler 工具展示 HTTP 响应结果

(2) 通过 Android 代码实现向服务端发送 HTTP 请求获取服务端响应结果, 并将结果经过 JSON 数据解析, 利用 Handler 更新界面数据, 并实时更新界面数据. 这个过程分析如图 6 所示, 核心代码如图 7 所示.

3.2 控制设备开关

客户端控制设备开关界面如图 8 所示. 具体实现过程与环境数据展示类似, 客户端发送 HTTP 请求 (POST 请求+API) 给服务端, 服务端处理请求后作出响应. 即通过操作界面上的水泵、风扇等设备开关按钮, 不仅实现界面上的动画效果, 更控制了沙盘上相应设备的开关.

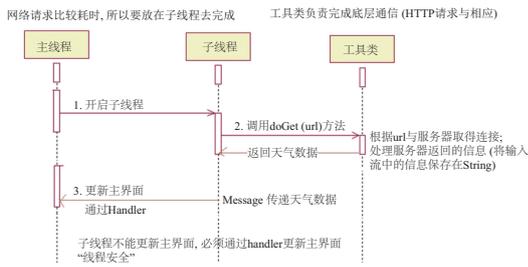


图6 Android代码分析

```
// 1. 子线程, 处理耗时操作, 发送HTTP请求获取传感器数据, JSON解析并保存; 通过消息机制让Handler处理数据在界面的展示
Runnable r = new Runnable() {
    @Override
    public void run() {
        while (true) { // 当页面关闭时, 无需发送HTTP, 无需更新数据了.
            // 1) 通过工具类发送HTTP请求获取所有传感器的数据
            String url = "http://api.openweathermap.org/";
            // 2) 解析得到的json数据并保存
            try {
                JSONObject obj = new JSONObject(get("values")); // 得到json数组
                JSONArray array = obj.getJSONArray("values"); // 得到json数组
                for (int i = 0; i < array.length(); i++) {
                    JSONObject content = array.getJSONObject(i);
                    String key = content.getString("data");
                    String value = content.getString("data");
                    DeviceData deviceValue = new DeviceData(key, value);
                }
            } catch (JSONException e) {
                e.printStackTrace();
            }
            // 3) 线程休眠1S
            try {
                Thread.sleep(1000);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
            // 4) 将Message对象加入消息队列 (会不断重复run方法, Handler中会取出消息处理)
            Message msg = new Message();
            handler.sendMessage(msg);
        }
    }
};
// 2. Handler: 在界面上显示各传感器数据
Handler handler = new Handler() {
    @Override
    public void handleMessage(Message msg) { // 取出消息队列中的消息进行处理, 会自动调用
        super.handleMessage(msg);
        // 在界面上显示各传感器数据
    }
};
```

图7 Android核心代码



图8 客户端设备控制界面

4 系统测试

系统的测试环境如下:

硬件环境: Win7 操作系统、4 G 内存、1T 硬盘、i7 处理器、Android4.4 手机。

软件环境: Eclipse、Fiddler、靠谱助手、Tomcat 7.0、MyEclipse。

在 WIFI 条件下, 经过系统测试, 初次打开应用耗时 1~5s, 用户登录系统后可使用包含环境数据展示、历史数据查询、控制设备开关等五大功能。环境数据展示功能每隔 1s 自动更新各传感器数值 (根据需要可以设置子线程每隔 0.5s 或 2s 等主动获取传感器数据)。控制设备开关功能向服务器提交请求响应时间为 1~3s (设置沙盘设备开关及客户端界面 Switch 开关和帧动

画)。系统的各功能和性能均达到了预期要求。

5 结语

本文应用多种技术开发一个基于 Android 平台的智能农业系统, 主要技术有: ZigBee 通信协议、传感器、Web Service、HTTP 协议、JSON 数据获取与解析、Android 子线程与 Handler 技术等。实现了包含环境指标、历史数据查询、手动控制、系统设置等功能为一体的智能农业 APP 系统。根据预设农作物生长周期需要的环境参数, 通过 Android 手机进行展示和操作管理, 提高了农业系统的智能化水平, 改善了落后的农业管理方式。后续根据需要, 本系统还可以增加土壤酸碱度等传感器和灌溉施肥等其他设备, 也还有功能拓展的空间。

参考文献

- 王恬, 李书琴, 王志伟. 农业信息搜索可视化平台研究. 计算机应用与软件, 2016, 33(3): 271-274.
- 孙想, 吴华瑞, 李庆学. 农业海量信息资源管理平台设计与实现. 计算机应用与软件, 2015, 32(3): 75-78, 104.
- 刘洋, 张钢, 韩璐. 基于物联网与云计算服务的农业温室智能化平台研究与应用. 计算机应用研究, 2013, 30(11): 3331-3335.
- 郭群勇, 王钦敏, 汪小钦. 农业空间信息资源共享服务平台. 计算机系统应用, 2012, 21(1): 1-4.
- 张佩云, 宫秀文, 谢荣见. 农业信息资源共享与信息服务系统构建研究. 计算机技术与发展, 2013, 23(11): 157-160.
- Wu LG, Kong LC, Zhang ZH. Water environment monitoring system based on ZigBee wireless sensor network. Proc. of the 4th IEEE International Conference on Software Engineering and Service Science (ICSESS). Beijing, China. 2013. 898-901.
- 伍海波, 匡静, 朱承学, 等. 基于 MVC 的教学资源管理系统的设计与实现. 计算机技术与发展, 2014, 24(7): 214-217, 222.
- 龚瑞琴, 毕利. 基于 Web Service 的 Android 技术应用研究. 电子技术应用, 2014, 40(1): 134-136.
- 林小钰, 冯文龙, 陈诚. 基于 HTTP 协议的车载终端数据传输的实现. 信息系统工程, 2014, (2): 37.
- 汪永松. 安卓手机 APP 开发套路之 HTTP 通信. 电脑编程技巧与维护, 2014, (19): 41-47, 65. [doi: 10.3969/j.issn.1006-4052.2014.19.008]
- 李金来, 吴涛. 基于 JSON 数据交换的移动终端与服务器端异构数据库的数据交互. 现代计算机, 2016, (16): 55-58. [doi: 10.3969/j.issn.1007-1423.2016.16.016]
- 张沪寅, 屈乾松, 胡瑞芸. 基于 JSON 的数据交换模型. 计算机工程与设计, 2015, 36(12): 3380-3384.
- 李文翔, 李忠森. 基于 Android 平台的智能温控系统设计与实现. 软件导刊, 2016, 15(6): 85-87.