

浙江工业大学

本科毕业设计说明书（论文）

（2017 届）



论文题目 城市智慧公交系统
设计与实现

作者姓名 王益挺

指导教师 田贤忠

学科(专业) 计算机科学与技术

所在学院 计算机科学与技术学院

提交日期 2017 年 6 月

摘要

随着经济的飞速发展，城市交通开始面临越来越大的挑战，逐渐开始成为制约社会发展的瓶颈。作为城市交通系统的重要组成部分，城市公共交通具有客流量大、运营成本低、节能环保等优点。因此如何发展城市公共交通，鼓励市民绿色出行成了一个值得深入研究的课题。

据调查结果显示，我国多数城市仍然使用传统的公交系统。乘客在等候公交时经常因为无法知道公交的到站时间而耽误行程，公交公司也无法通过对公交数据的实时监控制定合理的公交线路。因此为了缓解城市交通压力，提高市民出行效率，提高公交行业信息化水平，本文提出了一套基于物联网的城市智慧公交系统。该系统通过在公交车上安装集成了多种传感器以及定位模块的物联网节点，实现对公交位置和车上空余座位进行实时监控。乘客可以通过手机实时查询这些信息，公交公司也可以通过网页端对这些数据进行实时监控。

本文首先分析项目的背景及其研究意义，然后介绍项目的国内外研究现状及难点。接着对项目中的方法和技术进行简要分析，本项目的技术和方法将分为硬件、手机端、网页端、服务器端四个部分介绍。最后根据现代软件工程思想对系统进行需求分析、概要设计、详细设计、实现以及测试。

关键词：智慧公交，物联网，GPS 定位，热释电，PHP，iOS

Abstract

With the rapid development of economy, urban transport began to face more and more challenges, and gradually began to become a bottleneck restricting social development. As an important part of urban transport system, urban public transport has the advantages of large traffic volume, low operating cost, energy saving and environmental protection. So how to develop urban public transport, to encourage people to travel green has become a worthy of in-depth study of the subject.

According to the survey results show that most of our cities still use the traditional public transport system. Passengers waiting for the bus often because they can not know the arrival time of the bus and delay the trip, the bus company can not through the real-time monitoring of bus data to develop a reasonable bus lines. Therefore, in order to alleviate the pressure of urban traffic, improve the efficiency of public travel and improve the level of information technology in the bus industry, this paper proposes a set of urban intelligent bus system based on Internet of Things. The system through the installation of a variety of sensors in the bus and positioning module of the Internet of things nodes, to achieve the bus location and car spare seats for real-time monitoring. Passengers can query this information in real time via mobile phone, and the bus company can also monitor the data in real time through the webpage.

This paper first analyzes the background of the project and its research significance, and then introduces the research status and difficulties of the project at home and abroad. Then the project and technology in a brief analysis, the project's technology and methods will be divided into hardware, mobile phone, web, server-side four parts. Finally, according to the modern software engineering thinking of the system needs analysis, summary design, detailed design, implementation and testing.

Keywords: Smart bus, Internet of things, GPS positioning, pyroelectric, PHP, iOS

目录

摘要.....	I
ABSTRACT	I
第一章 绪论.....	1
1.1 研究背景和意义	1
1.2 国内外研究现状	2
1.3 本文的主要工作	2
1.4 本文的组织结构	2
1.5 本章小结	3
第二章 方法与技术.....	4
2.1 硬件技术	4
2.1.1 Arduino	4
2.1.2 NEO 6M GPS 模块.....	4
2.1.3 红外对管传感器	5
2.1.4 ESP-8266.....	5
2.2 移动端技术	6
2.3 网页端技术	7
2.4 服务器端技术	7
2.5 开发原则	8
2.6 本章小结	9
第三章 系统需求与可行性分析.....	10
3.1 功能性需求分析	10
3.1.1 公交车实时定位	10
3.1.2 空余座位实时监测	10
3.1.3 后台公交数据分析	10
3.1.4 其他辅助功能.....	10
3.2 非功能性需求分析	11
3.2.1 系统的流畅性.....	11
3.2.2 信息的准确性.....	11
3.2.3 系统的稳定性.....	12
3.3 系统可行性分析	12
3.3.1 技术可行性	12
3.3.2 经济可行性.....	12
3.3.3 法律道德可行性	12
3.3.4 管理可行性	13
3.4 本章小结	13
第四章 系统概要设计	14
4.1 系统业务流程	14
4.2 系统功能结构	14
4.3 系统架构设计	15

4.4	系统数据库设计	16
4.5	本章小结	22
第五章 系统详细设计		23
5.1	项目开发规范	23
5.1.1	系统目录规划	23
5.1.2	项目编码规范	23
5.1.3	版本管理规范	24
5.2	系统功能模块详细设计	25
5.2.1	物联网节点详细设计	25
5.1.4	移动客户端详细设计	27
5.3	本章小结	29
第六章 系统实现与测试		30
6.1	硬件部分实现	30
6.2	移动客户端实现	31
6.3	系统测试方案概述	32
6.4	系统测试原理	32
6.5	系统测试内容	33
6.6.1	移动端客户端测试	33
6.6.2	管理员端测试	35
6.6	本章小结	37
第七章 总结		38
7.1	完成的工作	38
7.2	收获与感悟	38
7.3	存在的问题及下一步工作	39
7.2.1	存在问题	39
7.2.2	未来展望	39
参考文献		40
致谢		42
附录		43
附录 1	毕业设计文献综述	43
附件 2	毕业设计开题报告	43
附件 3	毕业设计外文翻译（中文译文与外文原文）	43

图目录

图 4-1 系统业务数据流图	14
图 4-2 系统总体功能结构图	15
图 4-3 系统整体架构	16
图 4-4 系统数据库 E-R 图	17
图 4-5 系统主要数据表结构	19
图 5-1 MASTER 分支	24
图 5-2 FEATURE 分支	25
图图 5-3 HOTFIX 分支	25
图 5-4 物联网节点业务流程图	26
图 5-5 公交查询业务流程图	28
图 6-1 物联网节点	30
图 6-2 STORYBOARD 界面	31
图 6-4 移动客户端辅助功能界面	35
图 6-5 管理员端站点管理界面	36
图 6-6 管理员端数据图表实例	36

表目录

表 2-1 服务器端程序运行环境.....	7
表 4-1 数据库表清单.....	18
表 4-2 车辆信息表.....	19
表 4-3 线路信息表.....	21
表 4-4 当前公交信息表.....	21
表 5-1 服务器端目录规划表.....	23
表 6-1 系统测试表.....	33

第一章 绪论

1.1 研究背景和意义

交通是城市经济活动与社会活动的动脉，但是随着经济的飞速发展，城市交通开始面临越来越大的挑战，逐渐成为制约社会发展的瓶颈^[1]。城市公共交通是城市交通系统的重要组成部分。国务院办公厅发布的《关于城市优先发展公共交通的指导意见》指出，城市公共交通具有集约高效、节能环保等特点，发展城市公交有利于缓解交通拥堵，改善城市空气质量，是构建环境友好型、资源节约型社会的战略选择。因此，如何鼓励市民选择公交出行，是一个值得思考和研究的问题。

我国大多城市仍然使用传统的公交系统，乘客等候公交车的时候无法知道公交车发车时间、到站时间以及车内的空余座位情况。因此市民会因为“等公交”而耽误行程，从而放弃“绿色出行”，造成城市交通拥堵，公交运营公司也无法通过监控公交车实时情况来设置科学合理的路线和调度方案。因此，为了积极响应国家“城市优先发展公共交通”的号召，提高公交行业的信息化和智能化水平，缓解城市交通压力，提高市民出行效率，迫切需要建设一套基于物联网的实时、准确、高效的智慧公交系统^[2, 3]。

该系统一套基于物联网的城市智慧公交系统。通过在公交车上安装智能传感器，使得用户可以通过手机软件查询基本公交线路外，还可以查询公交实时位置、车内空位情况等。这些动态信息的采集，有利于乘客做出更好的出行规划，以及公交调度中心高效地调度车辆。该项目积极响应国家“城市优先发展公共交通”的号召，对提高公交行业的信息化和智能化水平，缓解城市交通压力，提高市民出行效率具有重要意义^[3]。

1.2 国内外研究现状

西方发达国家城市化水平高，对于城市智慧公交研究投入比较早，技术相对比较成熟。20 世纪 60 年代，加拿大开始投入相关技术的研发，建立起全球第一个可运行的地理信息系统——拿大地理信息系统 (CGIS)。此后日本、英国、法国、韩国等其他国家也对该领域研究予以高度重视。通过引入差分 GPS 定位、地理信息、地图匹配等先进技术，优化了现有公共交通网络，建立了一套先进的公共交通规划和管理系统，为乘客提供准确、可靠、及时的公交信息服务^[4]。

我国从 20 世纪 90 年代开始智能公交系统的研究，电子 IC 卡收费系统已经在多数城市得到应用。部分城市也开始实现公交的智能调度，开发了公交便民查询服务。但是多数城市公交系统的一体化程度、实时性以及准确性尚存在问题，其主要提供线路查询、换乘路线、发车间隔查询等常规功能，很少提供车辆实时位置、车内空余座位等查询功能^[4]。无法解决乘客因为等公交而耽误行程的问题。

1.3 本文的主要工作

本文首先分析项目的背景及其研究意义，然后介绍项目的国内外研究现状及难点。接着对项目中的方法和技术进行简要分析，本项目的技术和方法将分为硬件、手机端、网页端、服务器端四个部分介绍。最后根据现代软件工程思想对系统进行需求分析、概要设计、详细设计、实现以及测试。

1.4 本文的组织结构

本文共分为八章，设计并且实现了基于物联网的城市智慧公交系统。该系统由物联网硬件节点、服务器端、手机客户端、管理员网页端组成，通过 TCP 协议实现各端的交互。本文详细阐述了如何对各个模块进行设计与实现，各章内容如下：

第一章，介绍了项目的研究的背景和意义，国内外相关领域的研究现状，本文的主要工作和组织结构。

第二章，详细介绍系统开发所用到的方法与技术，为后续开发做好理论基础。

第三章，重点介绍了城市智慧公交系统的需求。

第四章，具体介绍了城市智慧公交系统的概要设计。其主要包括系统的详细描述、业务流程以及数据库设计。

第五章，详细介绍了城市智慧公交系统的详细设计。其内容包括开发规范的确定、系统所采用框架的整合设计、功能模块的详细设计和系统性能要求的详细设计。

第六章，着重阐述了城市智慧公交系统的具体实现，针对第五章提出的详细设计要求，在本章给出系统的技术实现，具体包括系统框架整合实现、功能性能要求实现和系统功能模块实现。

第七章，系统测试与系统使用说明。

第八章，对系统开发进行总结并提出下一步工作。

1.5 本章小结

本章简要地介绍了项目的研究背景和意义，分析了国内外相关领域的研究现状。最后，给出了本文的主要工作及本文的组织结构。

第二章 方法与技术

本系统分为硬件、移动客户端、网页端、服务端四大部分。硬件上将 GPS 定位模块、多种传感器以及无线模块集成到 Arduino 单片机上，成为一个物联网节点，通过 TCP 协议和服务器进行通信；服务端在 Linux+Nginx+Mysql+PHP 的环境下使用 Laravel 框架开发；移动客户端在 iOS 平台下开发；网页端则由 CSS、Javascript 等开发。本章将对上述知识进行简要的阐述。

2.1 硬件技术

本系统硬件主要分为 Arduino UNO 开发版、NEO-6M GPS 定位模块、对管红外传感器、ESP-8266 无线模块四个部分，开发平台为 Arduino IDE，开发语言为 Arduino、C 语言或汇编语言^[16]。

2.1.1 Arduino

Arduino是目前市面上最方便、流行、易于上手的开源硬件平台。Arduino板使用各种微处理器和控制器，并配备有可连接到各种扩展板和其他电路的数字和模拟输入/输出（I/O）引脚，串行通信接口和某些型号的通用串行总线（USB）。此外，Arduino还具有类似Java和C语言的开发环境，开发者可以使用IDE编写代码，并且将代码烧录到开发版中。

2.1.2 NEO 6M GPS 模块

选择NEO 6M GPS模块^[10-12]，该模块是一款高性能GPS定位模块，其特点包括：

- a. 体积小巧，性能优异。
- b. 可通过串口设置参数，并可保存到内部闪存，使用方便。
- c. 自带IPX接口，可连接各种有源天线
- d. 自带可充电后备电池，实现断电保持星历数据

该模块各引脚的详细描述如表2-1所示。

表 2-1 NEO 6M GPS 模块各引脚功能描述

序号	名称	说明
1	VCC	接电源(3.3V~5.0V)
2	GND	接地
3	TXD	模块串口发送管脚(TTL电平, 切勿直接接RS232电平), 可以接单片机的RXD
4	RXD	模块串口接收管脚(TTL电平, 切勿直接接RS232电平), 可以接单片机的TXD
5	PPS	时钟脉冲输出管

2.1.3 红外对管传感器

红外检测人数的方法主要有红外对管和热释电传感器两种。红外对管由一个红外发射管和一个红外接收管组成。检测原理如图1所示, 当有障碍物, 即有人时, 接收管收到障碍物反射的红外线, 传感器模块OUT端持续输出低电平信号^[15]。

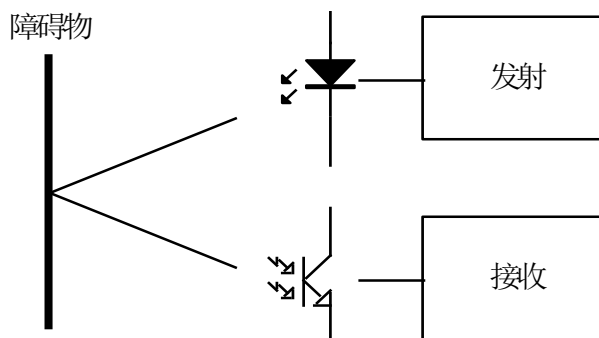


图2-1 红外对管检测原理图

热释电传感器则是通过感应人体本身向外界辐射红外能量检测人体的元件。它将人体发出的微弱红外线转化为电信号, 对其进行处理、放大, 从而达到检测人体的目的^[14]。

2.1.4 ESP-8266

目前, 常见物联网节点之间的通信方案有近距离无线传输的蓝牙和ZigBee, 基于传统互联网的Wi-Fi, 以及基于移动基站的3G / 4G网络^[16]。

本项目使用基于Wi-Fi的ESP8266模块。该模块是业内使用最广泛的Wi-Fi芯片之一，由乐鑫公司开发，具有低功耗、低成本的优点。

ESP8266拥有完整的且自成体系的Wi-Fi功能，既能独立工作，也可以集成到其他主机MCU中使用。其集成了32位CPU和高速缓冲存储器，有利于提高系统性能，并且优化存储系统。开发者可以使用C语言在ESP8266的IDE上对模块进行二次开发，以满足不同的应用需求^[17]。

2.2 移动端技术

目前市面上常见移动客户端主要包括iOS（苹果公司移动端操作系统）和Android（谷歌公司移动端操作系统）两类^[9]。本系统以在iOS系统下开发的客户端应用为例，开发环境为macOS系统下的Xcode，开发语言为Objective-C或者Swift^[20]。

在软件开发中，架构设计是衡量代码质量的绝对标准。本系统采用软件工程中常用的MVC（Model View Controller）架构。该架构在组织代码时分为模型（Model）、视图（View）、控制器（Controller）三层，将界面、数据、业务逻辑分离。其中模型是应用中数据逻辑对象，视图是用来显示数据的界面，控制器则是模型和数据的中介。该架构具有实体分工明确均匀、可测性好、易于维护的优点^[9]。客户端开发依照高内聚、低耦合的模块化思想，采用MVC架构，使代码具有一定的可读性。

乘客客户端中涉及到公交线路查询功能，系统需要给出换乘方案以及换乘地点。公交线路和站点构成了公交网络，从起点到终点的公交线路查询实际上可以抽象为公交网络的最优路径问题^[22]，常用的最短路径求解算法有Dijkstra算法、Floyd算法、A*算法等。在实际开发中将基于其中一种算法实现公交线路查询功能^[22, 23]。

2.3 网页端技术

本系统网页端开发用到的技术有Javascript、CSS、jQuery、Bootstrap等常用网页前端技术，本文不作详细介绍。

2.4 服务器端技术

服务器端程序开发环境为macOS系统下的PhpStorm，运行环境如表2-1所示。

表 2-1 服务器端程序运行环境

操作系统	Ubuntu
服务器软件	Nginx
数据库	Mysql
开发语言	PHP
缓存	Redis

- (1) Linux是一个基于UNIX，支持多用户、多线程、多任务以及多CPU的操作系统，凭借其开放源代码及稳定性，在服务器市场中占据了很大的市场份额。
- (2) Nginx 是一款轻量级 Web 服务器 / 反向代理服务器及电子邮件（IMAP/POP3）代理服务器，其源代码开放，系统资源占用少，并发能力强，受到阿里巴巴、百度、腾讯等大型互联网公司的青睐。
- (3) MySQL一种目前最流行的免费开源的关系型数据库管理系统^[19]，其特点如下：
 - a) 源代码开放，其特性通过可以修改源代码得到扩展；
 - b) 通过优化SQL查询算法有效提高查询速度；
 - c) 支持多种存储引擎；
 - d) 支持多线程等。
- (4) PHP（全称：PHP: Hypertext Preprocessor，即“PHP: 超文本预处理器”）是一种开源免费的通用计算机脚本语言^[18, 21]，其特点如下：

- a) 源代码开放，其特性通过可以修改源代码得到扩展；
- b) 开发简单快捷；
- c) 支持面向对象；
- d) 跨平台性强。

服务端程序将按照一下分层进行开发^[8]：

- a) 数据存储层：主要包括GPS轨迹数据库、车内传感器数据库。
- b) 数据库接口和数据处理层：该层支撑上层应用，对车辆采集数据进行处理计算，接口层是终端实时获取公交信息的通道。
- c) 组件层：包括GPS、传感器组件封装，保证系统模块组件重用性。
- d) 应用层：在各类组建基础上封装的面向不同应用的子模块，比如公交线路查询、实时位置查询、车内空余座位查询等。

2.5 开发原则

本文提出了一种基于物联网（Internet of things, IoT）技术的城市智慧公交系统。该系统通过在公交车上安装具有定位功能和多种传感器的物联网节点，采集公交车实时位置和车内空余座位等动态信息^[6]，并定时将信息汇总后通过车载4G网络发送到后端服务器进行处理，更新到数据库或者缓存中。在客户端采用B / S和C / S相结合的方式，通过HTTP请求后端服务器中的数据，实现了既能在车辆调度中心实施监控，又能让乘客通过任意一台接入因特网的移动终端，如手机、PAD等实时查询公交信息^[7]。

该系统架构分为物理层、数据存储层、数据库接口和数据处理层、组件层以及应用层。

- (1) 物理层：将物联网通信模块、定位模块、传感器等组成一个物联网外围节点，监测车辆信息以及与后端服务器通信。
- (2) 数据存储层：主要包括GPS轨迹数据库、车内传感器数据库。
- (3) 数据库接口和数据处理层：该层支撑上层应用，对车辆采集数据进行处理计算，接口层是终端实时获取公交信息的通道。

- (4) 组件层：包括GPS、传感器组件封装，保证系统模块组件可重用性。
- (5) 应用层：在各类组建基础上封装的面向不同应用的子模块，比如公交线路查询、实时位置查询、车内空余座位查询等^[8]。

目前市面上常见移动客户端主要包括iOS（苹果公司移动端操作系统）和android（谷歌公司移动端操作系统）两类。本文以在iOS系统下开发的客户端应用为例。

在软件开发中，架构设计是衡量代码质量的绝对标准。本系统采用软件工程中常用的MVC（Model View Controller）架构。该架构在组织代码时分为模型（Model）、视图（View）、控制器（Controller）三层，将业务逻辑、数据、界面显示分离。其中模型是应用中数据逻辑对象，视图是用来显示数据的界面，控制器则是模型和数据的中介。该架构具有实体分工明确均匀、可测性好、易于维护的优点^[9]。

2.6 本章小结

本章从硬件、移动端、网页端、服务器端四个方面，介绍了城市智慧公交系统开发过程中需要用到的方法和技术。同时明确了系统开发的总体原则，为后期开发提供了坚实的理论基础。

第三章 系统需求与可行性分析

3.1 功能性需求分析

3.1.1 公交车实时定位

公交车实时位置查询是客户端软件的核心功能。目前市面上公交类软件大多数只提供公交线路信息、站点信息、换乘信息等静态信息。虽然这些信息能满足乘客的部分需求，但由于公交车到站时间的不可预知性，多数乘客仍然会因为等公交耽误行车。随着移动互联网和物联网的高速发展，通过手机应用查询公交车实时位置从技术层面讲已经完全可行。该功能有利于乘客在选择公交出行时提前规划，做到运筹帷幄。因此，该功能是城市智慧公交系统必须实现的功能之一。

3.1.2 空余座位实时监测

车辆空余座位实时监测也是客户端软件的一个重要功能。目前市面上几乎没有一个公交系统支持车辆空余座位查询。乘客通过手机应用查询公交实时位置的同时，还能查到该车辆当前空余座位数。该功能可以让乘客在同一时间段内拥有多种公交乘坐选择的情况下，尽可能选择空余座位多的车辆。该功能不仅可以让乘客在乘坐公交时候尽可能拥有座位，还能优化整个公交系统的负载均衡。因此车辆座位监控是本项目必要需求之一。

3.1.3 后台公交数据分析

通过记录并分析本章 3.1.1 中公交实时定位数据和 3.1.2 中实时座位信息，可以在管理员端实现公交数据报表分析。该功能有利于公交运营公司利用公交大数据，制定更加合理的公交调度方案，保障城市公交系统合理高效。该功能是本系统设计目标之一，也是后期迭代优化重要方向。

3.1.4 其他辅助功能

除了以上核心功能之外，本系统将实现目前市面上公交类软件的基础功能，

如公交线路查询、站点查询、公交换乘、公交资讯等。本系统的管理员端还将实现对公交信息的维护功能，例如新增车辆、新增线路、新增站点等，以确保系统的完整性和可用性。

3.2 非功能性需求分析

3.2.1 系统的流畅性

本系统物联网节点与服务器、客户端与服务器均通过无线网络实现通信，由于系统的实时性比较强，确保系统是实行和流畅性时该项目的一个重要需求。

在物联网节点与服务器交互时，由于城市公交车数量巨大，安装在每一辆公交车上的物联网节点对服务器的频繁请求将造成严重的网络拥塞和巨大的服务器并发压力。因此，在物联网节点层面，要做到对服务器的分时请求，并且在不影响功能的前提下适当延长请求间隔。在服务器层面，则需要做高并发优化，比如使用 Nginx 实现服务器集群的负载均衡，以数据库集群、数据散列等形式优化数据存储，用 Redis 等非关系型数据库将实时数据缓存在内存中等。

在客户端与服务器通信时，客户端和服务器建立可靠连接以及数据在网络中的传输时需要一定的时延，该时延取决于移动蜂窝网络的畅通程度。因此为了防止阻塞主线程，造成界面假死、操作不灵的现象，网络请求需要在一个单独的子线程中异步进行。此外，为了体现软件的用户友好性，需要设定一个超时机制，即服务器未能在有效时间内响应时，将给出“当前网络不佳”等提示。

3.2.2 信息的准确性

本系统信息准确性分为静态数据源准确性和实时数据准确性两方面。

由于公交信息的复杂性和多变性，本系统将开发功能强大的管理员后台来维护公交信息，以确保静态公交数据的准确性。

在实时数据方面，由于 GPS 的定位误差，最好结合移动蜂窝网络基站定位，以增加定位的准确性。此外，由于传感器的存在误差，系统最好利用相关算法结合多传感器信息融合检测。这些实时数据的准确性优化有利于提高系统的用户体

验。

3.2.3 系统的稳定性

由于系统的复杂性和实时性，在开发过程中需要着重考虑系统的异常处理能力及容错能力，以确保系统能够稳定运行。在硬件上，需要保证节点在长时间运行下网络模块的稳定和对传感器异常数据的处理。在服务器端，则要确保服务器的稳定性，在通过本章 3.2.1 中方法提高服务器的并发能力的同时，需要建立主从服务器来防止服务器宕机造成的系统奔溃。在移动客户端，则要加强程序的健壮性，防止数组越界、内存泄漏、访问野指针等造成的系统闪退。

3.3 系统可行性分析

3.3.1 技术可行性

本项目用到的技术分为硬件、服务端、客户端三个部分，具体的技术路线已经在本文第二章作了详细介绍。本项目比较全面地考虑了开发过程中所有涉及的技术问题，采用了目前比较成熟的技术，作者开发能力较强，从技术角度来讲切实可行。

3.3.2 经济可行性

本项目硬件成本较低，项目中用到的通信模块、定位模块只需要几十元人民币，传感器只需要几元人民币。此外，本项目具有较好的市场前景，可以获得较为可观的收益，因此本项目在经济上是可行的。

3.3.3 法律道德可行性

本项目用到的技术分为硬件、服务端、客户端三个部分，具体的技术路线将在本文第四章作详细介绍。本项目比较全面地考虑了开发过程中所有涉及的技术问题，采用了目前比较成熟的技术，作者开发能力较强，从技术角度来讲切实可行。

3.3.4 管理可行性

在开发过程中，作者采用 Git 软件对项目进行版本管理。在系统层面上，作者为公交调度中心的管理人员编写专业的管理后台，方便工作人员对系统进行管理操作。因此本项目在管理上切实可行。

3.4 本章小结

本章通过对市面上已有的公交类软件调研，和对系统使用人群的使用场景分析，明确系统的需求。在确保公交线路查询、站点查询、换乘等基础功能的同时，提出了“公交实时定位”、“座位实时监测”、“公交大数据分析”等核心功能。此外，本章还从“系统的流畅性”、“信息的准确性”以及“系统的稳定性”三方面阐述了系统的非功能性需求，以提高系统的用户友好性。

通过对系统的需求分析和可行性分析，明确了系统的使用群体、主要功能以及非功能性优化的方向，为后续系统设计开发打下了坚实的基础。

4.1 系统业务流程

```

graph TD
    ViewReport[查看数据报表]
    AdminEnd[管理员端]
    MaintainInfo[维护公交信息]
    QueryBus[查询公交]
    SendDynamic[发送动态信息]
    DataReport[数据报表]
    StationTable[站点表]
    VehicleTable[车辆表]
    RouteTable[线路表]
    CurrentVehicleInfo[当前车辆信息表]
    Hardware[硬件]

    AdminEnd --> ViewReport
    AdminEnd --> MaintainInfo
    MaintainInfo --> DataReport
    MaintainInfo <--> StationTable
    MaintainInfo <--> VehicleTable
    StationTable --> QueryBus
    VehicleTable --> QueryBus
    MobileUser[手机端用户] --> QueryBus
    QueryBus --> CurrentVehicleInfo
    QueryBus --> RouteTable
    CurrentVehicleInfo --> SendDynamic
    Hardware --> SendDynamic
    SendDynamic --> CurrentVehicleInfo
    SendDynamic --> ViewReport
  
```

图 4-1 系统业务数据流图

本文提出了一种基于物联网（Internet of things, IoT）技术的城市智慧公交系统。主要实现移动客户端公交信息查询、管理员后台公交信息维护、公交数据监控等。硬件方面主要有车辆动态信息采集和网络请求两大功能模块，移动客户端有公交实时信息查询、公交换乘方案查询、站点信息查询、公交资讯、等功能模块，管理员端具有公交信息维护和公交数据报表两大功能模块。系统的功能结构如图 4-2 所示：

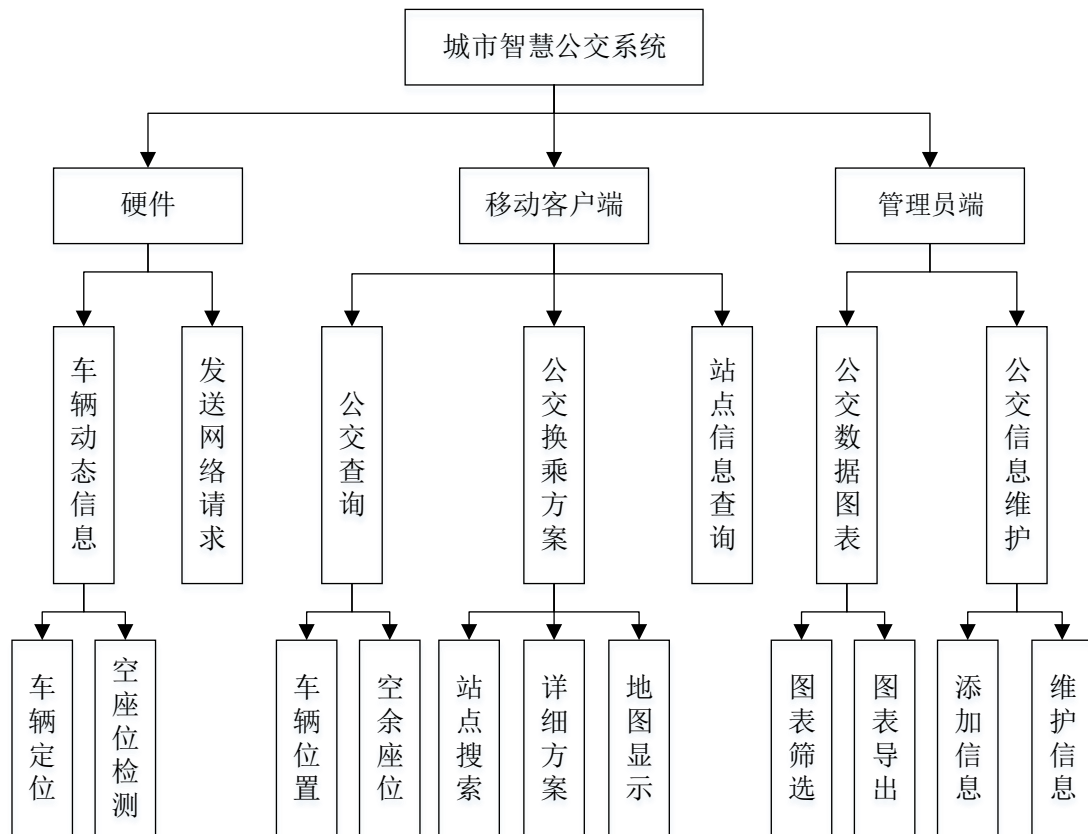


图 4-2 系统总体功能结构图

4.3 系统架构设计

本系统架构分为物联网节点、服务器、客户端三大部分。物联网节点通过移动蜂窝网络，与中央服务器通信，服务器接收到硬件的请求，对数据进行运算处理，并且存储到 Redis 和 Mysql 数据库中。客户端分为移动手机端和网页管理员端两端，均采用 MVC（Model View Controller）架构。该架构在组织代码时分为模型（Model）、视图（View）、控制器（Controller）三层，将界面、数据、业务逻辑分离。其中模型是应用中数据逻辑对象，视图是用来显示数据的界面，控制器则是模型和数据的中介。系统整体架构如图 4-3 所示。

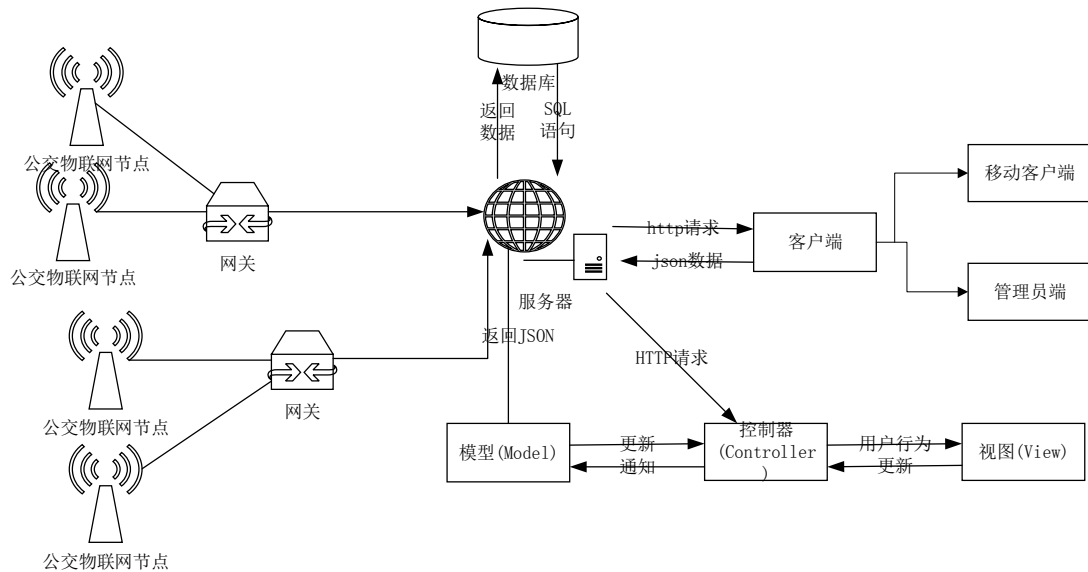


图 4-3 系统整体架构

本系统通过在公交车上安装具有定位功能和多种传感器的物联网节点，采集公交车实时位置和车内空余座位等动态信息^[6]，并定时将信息汇总后通过车载4G网络发送到后端服务器进行处理，并更新到数据库或者缓存中。在客户端采用B/S和C/S相结合的方式，通过HTTP请求后端服务器中的数据，系统整体架构图4-3所示。

4.4 系统数据库设计

本系统使用 MySQL Workbench 软件进行数据库建模，系统主要有车辆信息表、线路信息表、站点信息表、线路站点关系表等，公交实时信息将存储在非关系型数据库 Redis 中，并定期持久化到 MySQL 中用于公交数据分析。系统核心模块主要有车辆、站点、线路、当前等几张表，数据库 E-R 图如图 4-4 所示。

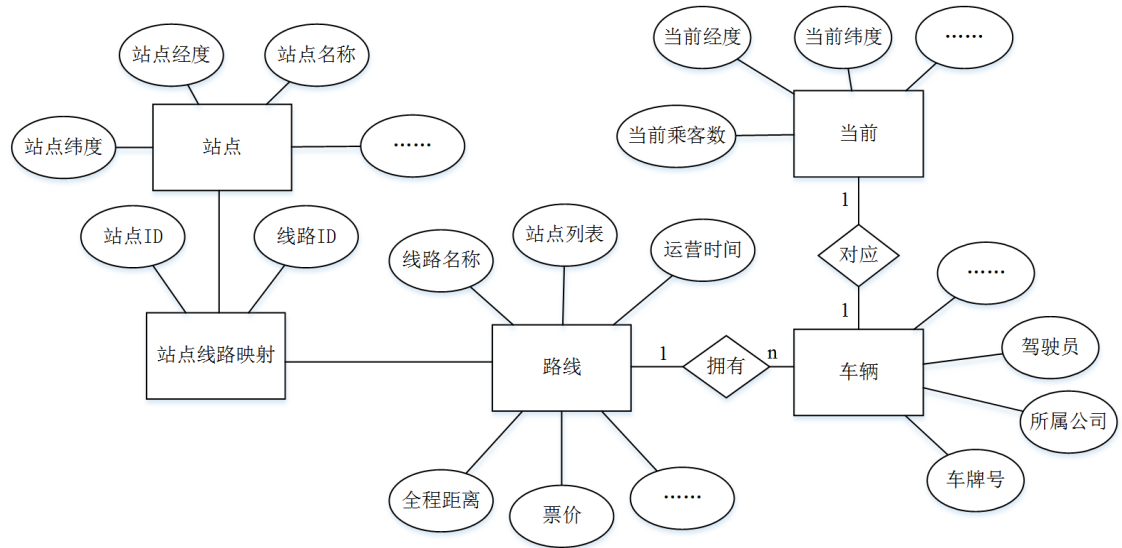


图 4-4 系统数据库 E-R 图

系统数据库表清单如表 4-1 所示。

表 4-1 数据库表清单

	表名	注释
1	bus	存储公交车相关信息
2	stop	存储公交站点信息
3	route	存储公交线路信息
4	stop_route_relation	存储公交线路和站点映射关系
5	current	存储当前正在运营公交信息
6	lostthing	存储失物招领信息
7	suggest	存储意见等信息

与城市智慧公交相关的主要数据表结构如图 4-5 所示：

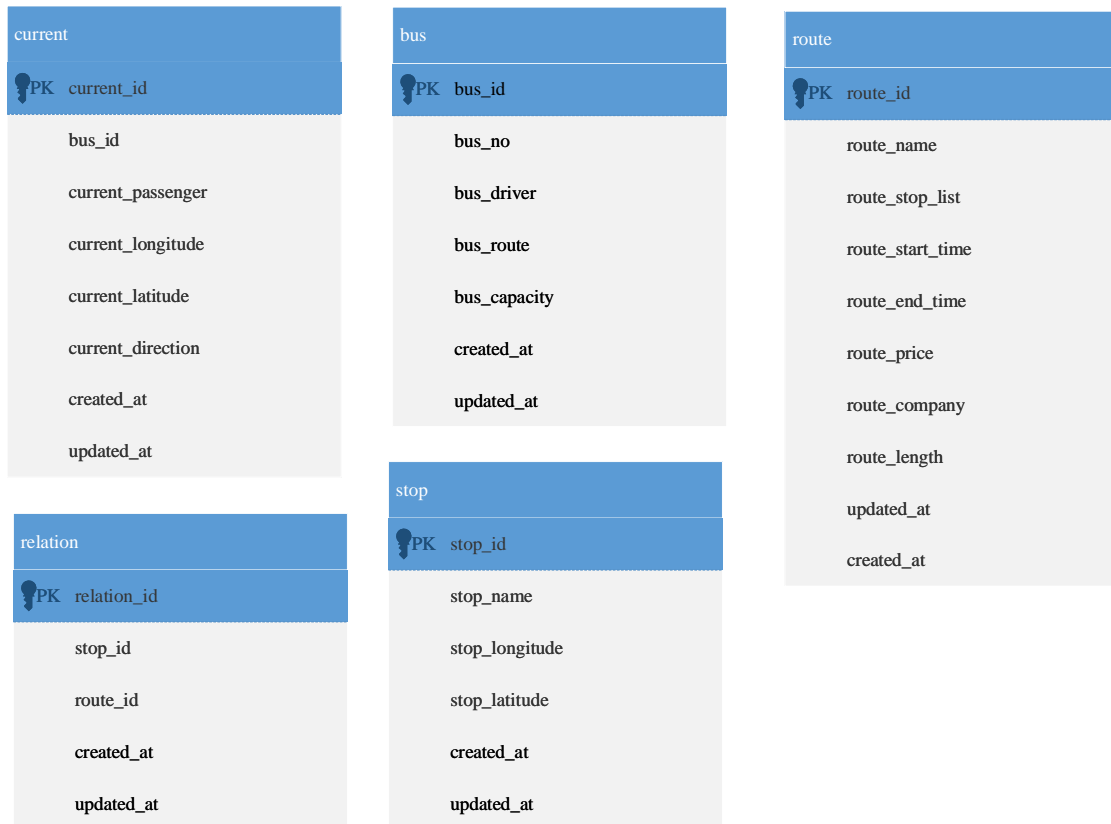


图 4-5 系统主要数据表结构

车辆信息表主要存储车牌号、所属线路、座位数等信息，表结构设计如表 4-2 所示。

表 4-2 车辆信息表

字段	类型	空	默认	注释
bus_id	int(10)	否	NULL	主键
bus_no	varchar(20)	否	NULL	车牌号
bus_driver	varchar(40)	否	NULL	驾驶员
bus_route	int(11)	否	NULL	所属线路
bus_capacity	int(11)	否	NULL	座位数
created_at	datetime	是	NULL	创建时间

updated_at	datetime	是	NULL	更新时间
------------	----------	---	------	------

线路信息表主要存储线路站点列表、运营时间、票价等信息，表结构设计如表 4-3 所示。

表 4-3 线路信息表

字段	类型	空	默认	注释
route_id	int(11)	否	NULL	主键
route_name	varchar(40)	否	NULL	线路名称
route_stop_list	text	否	NULL	站点列表
route_start_time	varchar(30)	否	NULL	运营开始时间
route_end_time	varchar(30)	否	NULL	运营结束时间
route_price	varchar(30)	否	NULL	票价
route_company	varchar(80)	是	NULL	所属公司
route_length	varchar(255)	是	NULL	路线长度
updated_at	datetime	是	NULL	创建时间
created_at	datetime	是	NULL	更新时间

当前公交信息表主要存储当前公交空余座位数、实时位置、当前方向等信息，表结构设计如表 4-4 所示。

表 4-4 当前公交信息表

字段	类型	空	默认	注释
current_id	int(10)	否	NULL	主键
bus_id	int(11)	否	NULL	车辆 ID
current_passenger	int(11)	否	NULL	当前乘客数
current_longitude	varchar(255)	否	NULL	当前经度
current_latitude	varchar(255)	否	NULL	当前维度
current_direction	tinyint(1)	否	NULL	当前方向
created_at	datetime	是	NULL	创建时间
updated_at	datetime	是	NULL	更新时间

4.5 本章小结

本章主要阐述了系统的概要设计。首先，根据第三章中系统的需求和可行性分析，设计了系统硬件的工作流程。接着，分析了系统的功能结构，明确了系统的整体架构。最后，根据系统的功能设计了数据库结构。

第五章 系统详细设计

5.1 项目开发规范

项目的开发规范在项目开发中起到了举足轻重的作用。良好的开发规范可以提高团队协作效率，减少开发过程中的 Bug，降低后期维护成本等。

5.1.1 系统目录规划

表 5-1 服务器端目录规划表

目录		名称及说明
/App	Controllers	此目录用于存放控制器类文件
	Jobs	此目录是放置队列任务的地方，应用中的任务可以被队列化，也可以在当前请求生命周期内同步执行。
	Events	此目录是放置事件类的地方，事件可以用于通知应用其它部分给定的动作已经发生，并提供灵活的解耦的处理。
	Exceptions	此目录包含应用的异常处理器
	Middleware	此目录用于存放中间件类文件
/bootstrap		此目录包含了少许文件用于框架的启动和自动载入配置
/config		此目录用于存放应用所有的配置文件
/database		此目录用于数据库迁移
/public		此目录用于存放系统所用公共资源文件
/storage		此目录用于存放 Session、日志等文件
/vendor		此目录用于存放 Composer 依赖
.env		环境变量配置文件，包括 mysql、session 等配置

5.1.2 项目编码规范

(1) PHP 代码文件必须使用不带 BOM 的 UTF-8 编码，并使用 Unix LF (linefeed) 作为行的结束符。

(2) 类名使用大驼峰命名法，即首字母必须大写，类名和类的文件名必须保

持一致。

(3) 类的方法使用小驼峰命名法，即首字母小写。方法名集中体现该方法的作用，避免方法名过于冗长。

(4) 变量的命名规则和类方法命名规则相同，变量名应明确指出该变量的作用。只允许在 for 循环迭代器中使用无意义的变量名。

5.1.3 版本管理规范

开发中使用 Git 对项目进行版本管理，代码分支及版本管理规范对软件开发及迭代具有重要意义。

(1) Master 分支为项目的主分支，每一次 Commit 之后应该打上 Tag，如图 5-1 所示。

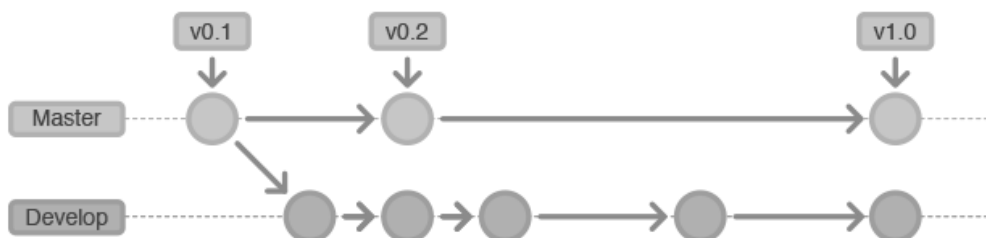


图 5-1 Master 分支

(2) Feature 分支为项目的新特性分支，分支名为 feature/*。在 Feature 分支开发完成后，必须合并回 Develop 分支，通常在合并完成后删除 Feature 分支，如图 5-2 所示。

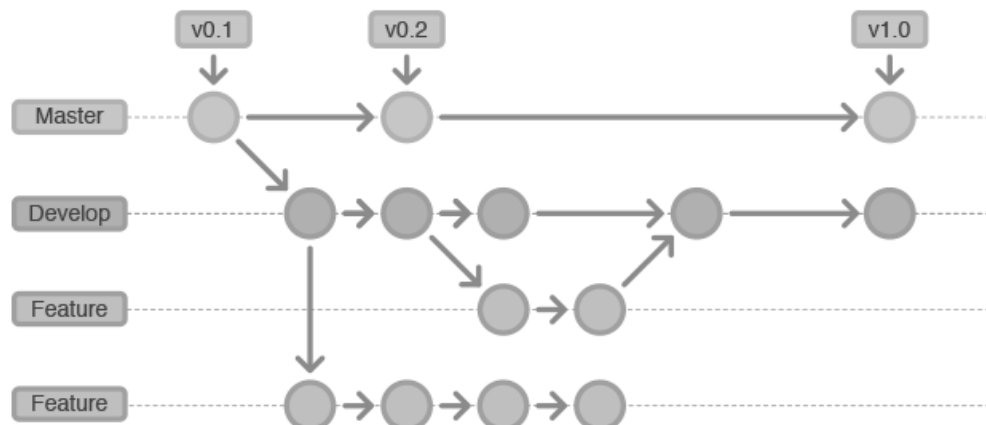


图 5-2 Feature 分支

(3) Hotfix 分支基于 Master 分支创建，用来修复线上紧急 Bug。在 Hotfix 分支完成开发完后需要合并回 Master 和 Develop 分支，同时在 Master 上打一个 tag，如图 5-3 所示。

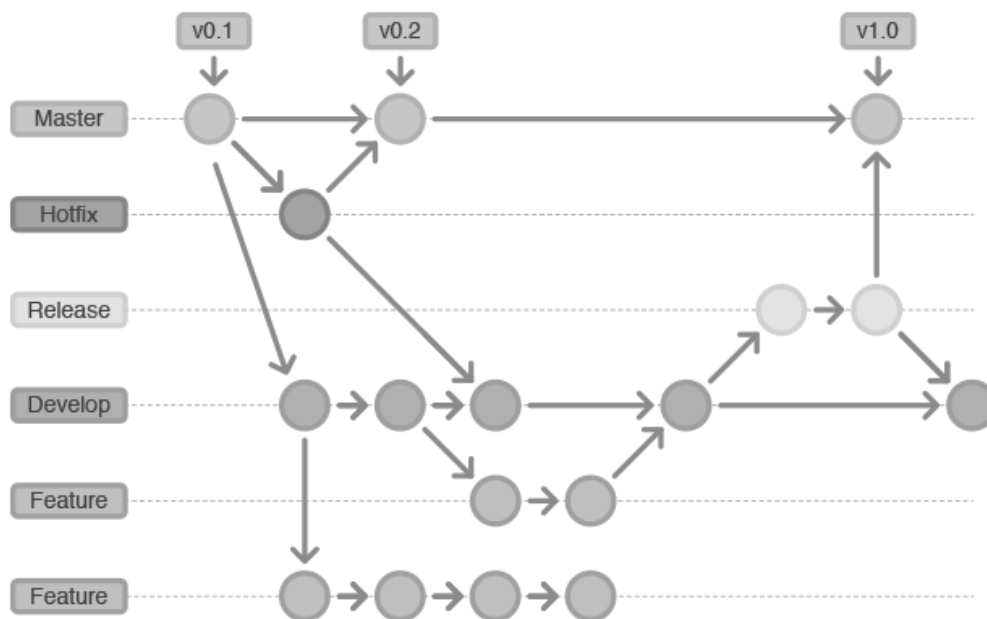


图 5-3 Hotfix 分支

5.2 系统功能模块详细设计

5.2.1 物联网节点详细设计

本系统物联网节点的业务流程包括 GPS 定位信息获取和解析、红外对管传感器数据获取和统计、无线模块网络连接、与服务器建立可靠连接、数据传输，详细流程如图 5-4 所示。

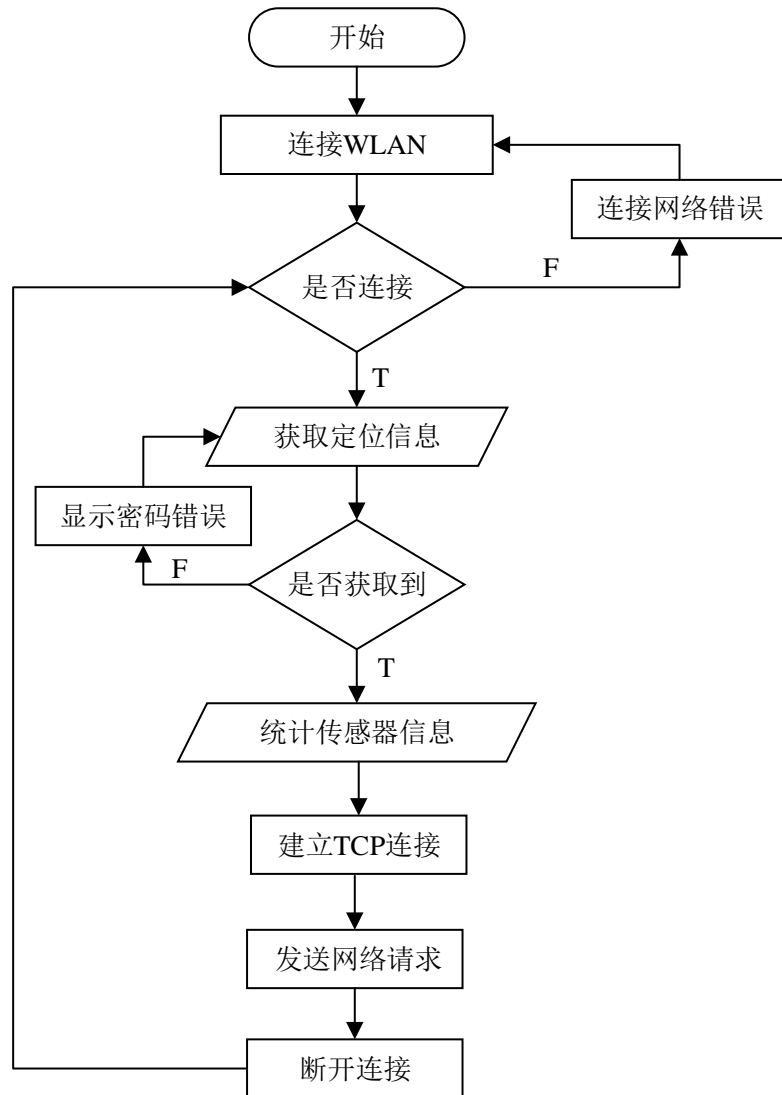


图 5-4 物联网节点业务流程图

首先，对 ESP8266 无线模块进行初始化，设置其连接 WLAN 的 SSID 和密码、请求得服务器地址和端口、串口波特率等。初始化代码如下：

```

const char* ssid = "Eda";
const char* password = "10364627";
const char * host = "smartbus.eda.im"; //模块请求的服务器地址
const int httpPort = 80; //模块请求的服务器端口
Serial.begin(9600); //串口波特率为 9600
pinMode(D2, INPUT); //定义 D2 引脚为 INPUT 模式
  
```

硬件核心功能为 GPS 信息解析和网络请求发送。系统将开发板 IO 口作为软串口，连接 GPS 的 TDX 管脚按字节读取定位信息并解析，通过 ESP8266 透传模式实现与服务器交互。具体实现核心代码和注释如下：

```
for (unsigned long start = millis(); millis() - start < 1000;)
{ //一秒钟内不停扫描 GPS 信息
    while (GPSSerial.available()) //串口获取到数据开始解析
    {
        char c = GPSSerial.read(); //读取一个字节获取的数据
        ..... //将串口数据流解析为经纬度
        if(latitude > "" && longitude > ""){ //得到经纬度信息
            ..... //发送网络请求
        }
    }
}
```

5.1.4 移动客户端详细设计

移动客户端主要业务为实时公交查询，业务流程图如图 5-5 所示。

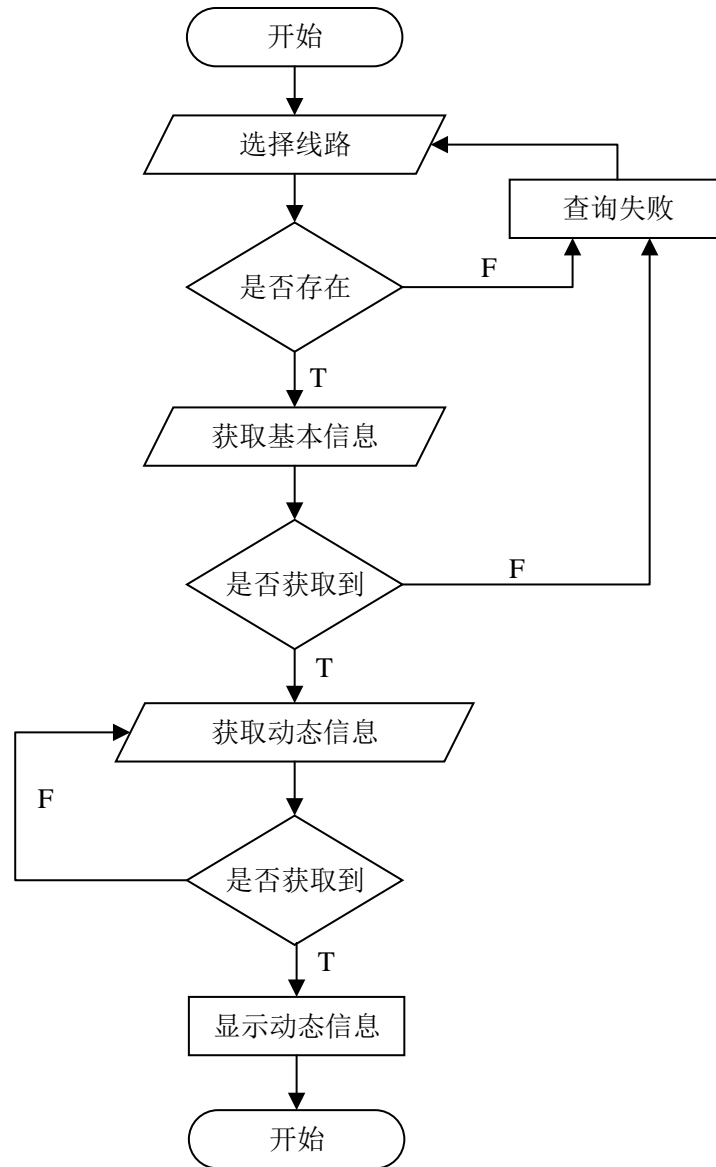


图 5-5 公交查询业务流程图

系统通过 GPS 定位模块获取到车辆当前经纬度, 通过 K-邻近算法匹配与当前车辆最近的公交站点, 实现代码如下:

```
$deltaArr = array(); //欧氏距离集合
```

```
foreach ($arrStopList as $tmp) {
```

```
    ..... //求当前位置和各个站点欧氏距离
```

```
    $delta = sqrt(pow(($currentLongitude - $stopLongitude), 2) + pow(($currentLatitude - $stopLatitude), 2));
```

```
.....  
  
}  
ksort($deltaArr); //根据索引排序  
..... //最近站点存储到数据库
```

5.3 本章小结

本章主要根据第三章、第四章对城市智慧公交系统作了详细设计。首先，在系统开发之前，对系统的开发规范，包括目录规划、命名规则等进行了规定。然后，重点介绍了系统硬件的手机客户端的具体工作流程。

第六章 系统实现与测试

6.1 硬件部分实现

本系统硬件为在 Arduino 开发板上集成一个定位模块、一个无线模块和一个传感器来模拟安装在公交车上的一个物联网节点，如图 6-1 所示（在实际使用中，传感器个数由公交车座位数决定）。

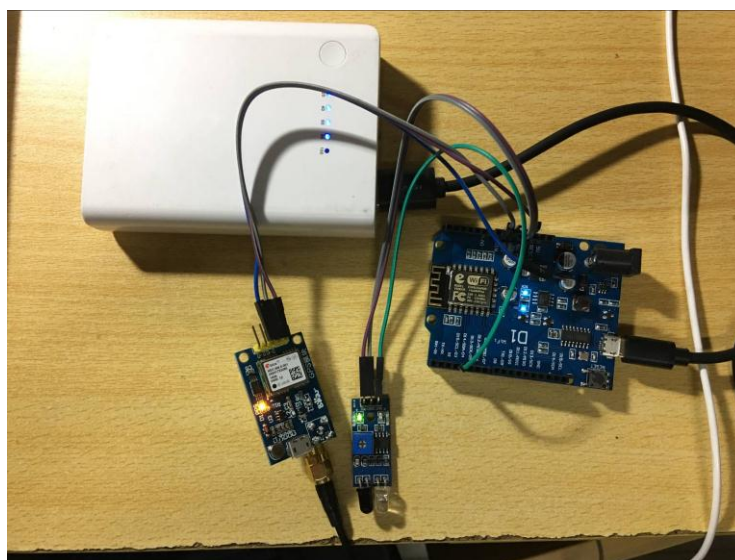


图 6-1 物联网节点

首先用杜邦线连接各模块管脚，VCC 接开发板 3.3V，GND 接开发板 GND，GPS 模块 TDX 接开发板 RDX，传感器 OUT 接开发板 D2。接线完成后，用 Arduino IDE 对硬件进行开发调试。IDE 中开发板选择 WeMos D1，CPU Frequency 选择 80MHZ，Upload Speed 选择 115200。根据第五章中详细设计写完硬件代码后，用 IDE 将程序烧录到开发板，打开串口调试工具调试程序。

当 GPS 定位模块橘黄色指示灯闪烁时，表示模块已经成功获取当前的经纬度信息，并且将信息串口传输到 Arduino 开发板中。红外对管传感器上一盏绿色指示灯亮起，表示传感器正常工作且当前座位上没有乘客；红外对管传感器上两盏指示灯同时亮起时，表示传感器正常工作且在当前座位上检测到了乘客。

6.2 移动客户端实现

移动客户端在 Xcode 下开发，在 iPhone 真机调试下完成。iOS 端界面在 Storyboard 下设计完成，采用 UITabBar 嵌套 UINavigationController 的模式，如图 6-2 所示。此外，公交信息、站点信息展示功能用了 UITableView 控件，搜索功能用了 UISearchBar 控件。

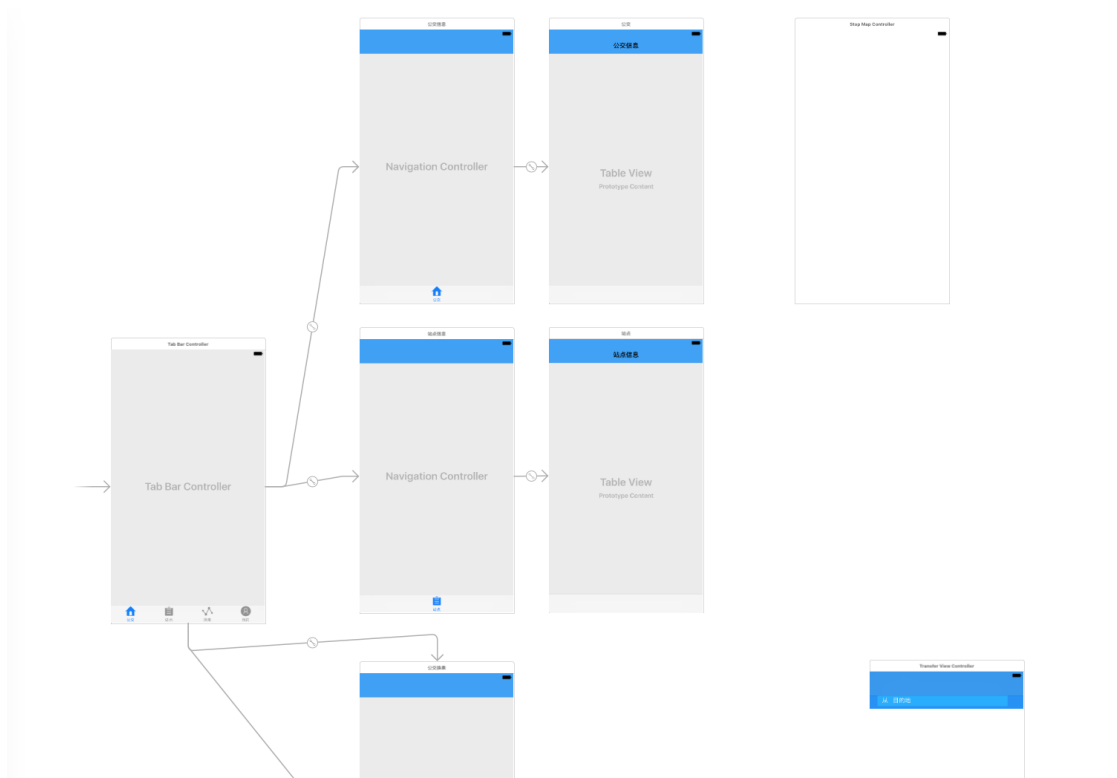


图 6-2 Storyboard 界面

网络请求是系统的核心部分，为了防止网络请求阻塞主线程导致界面卡顿，采用异步线程发送网络请求，并且将请求到的数据缓存到沙盒中。实现代码如下。

```
NSURLSessionConfiguration *defaultConfigObject = [NSURLSessionConfiguration defaultSessionConfiguration];
```

```
// 1.设置请求路径
```

```
.....
```

```
// 2.创建请求对象
```

```
.....
```

```
// 3.设置请求体
```

```
.....

// 4.设置请求头信息

.....

NSURLSessionDataTask * dataTask =[defaultSession dataTaskWithRequest:req
uest completionHandler:^(NSData *data, NSURLResponse *response, NSError *c
onnectionError) {
    ..... //异步请求线程
    if([status isEqual: @1])
    {
        .....
        //暂时存储到沙盒
    }
}
```

6.3 系统测试方案概述

为了避免开发过程中的失误和风险，我们需要对系统进行测试。测试工作是系统开发的一个重要环节，通过测试，可以尽可能发现软件内部逻辑问题、功能缺陷和致命错误，确保每个单元可以正确实现其预期功能。模拟城市智慧公交系统的使用场景，我们将从界面、功能、性能、易用性、兼容性、安全性等几个方面对系统进行全面测试。

6.4 系统测试原理

软件工程中常用的测试技术有黑盒测试和白盒测试。黑盒测试代表只对结果进行测试，对与软件的实现过程可以忽略。而白盒测试则是沿着系统的实现逻辑，对系统的每一个步骤进行测试。因此，在实际测试中，我们将会策略性地综合这两种测试方法。此外我们还会对系统进行回归测试，来避免修改漏洞过程中引发的新的漏洞，以保证软件的健壮性和整个测试过程的科学性。

6.5 系统测试内容

城市智慧公交系统的测试项目包括：系统界面（UI）测试、功能模块测试、系统易用性测试、系统测试、系统安全性测试和系统兼容性测试，如表 6-1 所示：

表 6-1 系统测试表

测试内容	测试要求
界面（UI）测试	窗口风格和设计预期保持一致，并且能对不同的设备作出适配，保证界面不出现拉伸变形。 保证用户界面友好性、易操作性，保证界面每个按钮能流畅地实现预期功能。
功能测试	分别对从硬件、服务器端、移动客户端、管理员端的各个功能模块进行测试，确保数据准确可靠，功能达到预期效果。
性能测试	对客户端进行流畅度测试，确保在快速点击、快速滑动的使用场景下系统流畅稳定。 对服务器进行高并发测试，使用 Apache-ab 等工具多线程模拟并发请求，对服务器进行并发负载压力测试。
易用性测试	测试不同用户群体能否对系统进行操作。
兼容性测试	管理员端在不同内核、不同版本的浏览器，不同的分辨率，不同的操作系统下分别进行测试。 移动客户端分别在所有版本 iPhone 和 iPad 下进行测试。
安全性测试	对管理员端数据安全性进行测试，避免系统遭到 XSS、CSRF、SQL 注入等攻击。

6.6.1 移动端客户端测试

用图 6-1 中的硬件模拟一辆 193 路公交车，移动客户端将会在该公交车实时位置附近的站点显示黑色公交标志和该公交实时座位数。如图 6-3 所示，当传感器被人挡住时，移动客户端实时座位数也会发生变化，测试结果符合预期。



图 6-3 移动客户端公交实时信息查询界面

此外，移动客户端还将提供站点查询、公交换乘等基础功能，如图 6-4 所示。经测试，公交站点信息、公交换乘线路能显示到地图上，路线规划准确无误，结果符合预期。

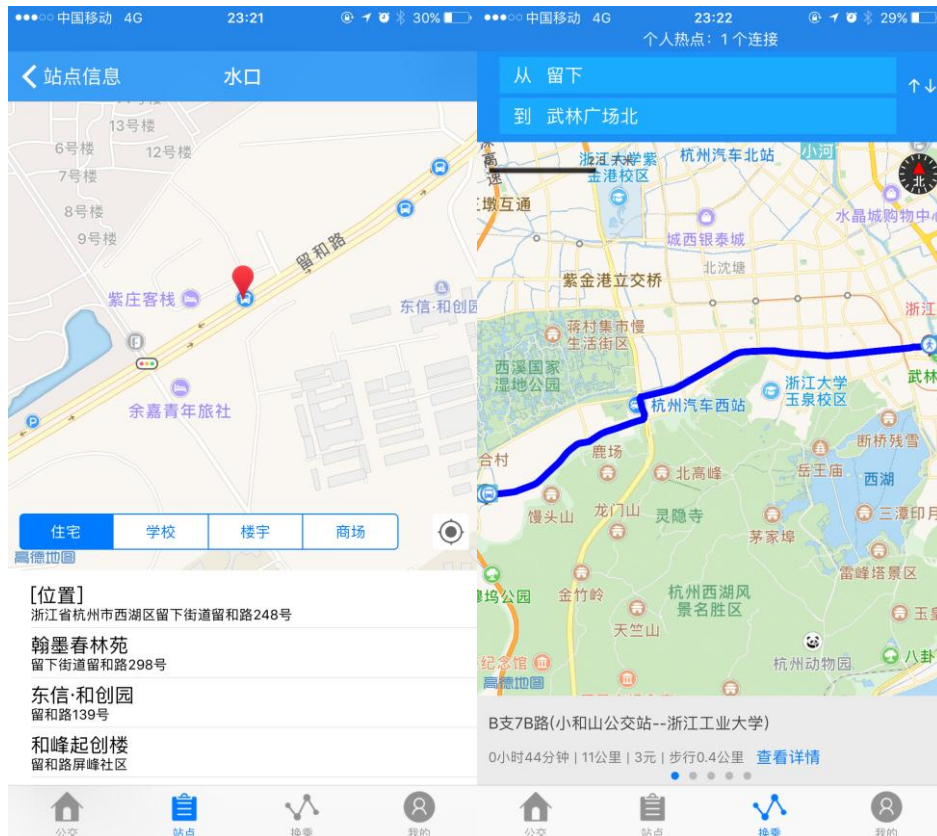


图 6-4 移动客户端辅助功能界面

6.6.2 管理员端测试

(1) 公交运营公司工作人员可以通过系统管理员端对工具信息进行管理维护，例如新增公交线路，修改站点信息等，如图6-5所示。经测试，站点信息准确无误，对站点信息增删改查功能均达到预期效果。

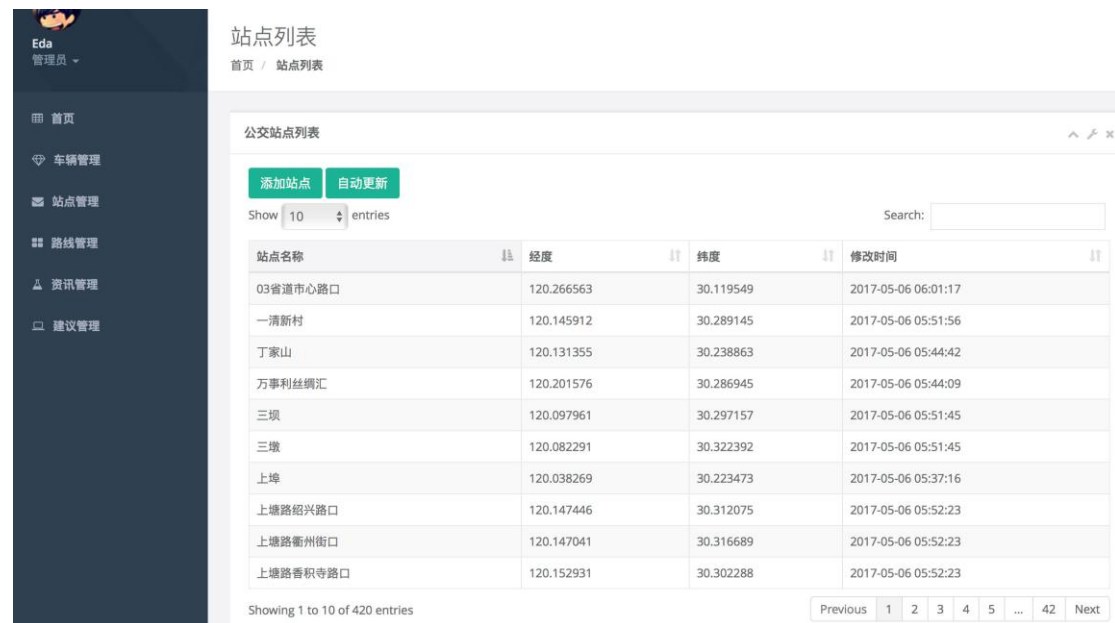


图 6-5 管理员端站点管理界面

(2) 此外，系统在服务端对公交的实时信息进行采集入库，整理成公交数据报表。管理员可以在管理员端通过图表形式查看公交的数据图标，该数据图标可根据实际情况定制，示例如图6-6所示。经测试，该模块能准确直观显示公交信息图表，且界面对用户友好。



图 6-6 管理员端数据图表实例

6.6 本章小结

本章从硬件、移动客户端、管理员端三方面展示了系统的实现，并且阐述了对系统的测试过程。

第七章 总结

7.1 完成的工作

本文首先分析了传统公交系统存在的问题与缺陷：无法实时查询公交的动态信息，导致乘客因等候公交耽误行程。由此，我们分析了城市智慧公交的研究意义，讨论了国内外研究现状以及技术难点。针对以上不足，我们结合国内外研究现状，参考了物联网在其他领域的应用技术，提出了基于物联网和移动客户端的智慧公交系统。同时，通过查阅相关文献，本文确定了系统的技术架构，介绍了系统解决方案中涉及到的相关技术和算法，为项目的顺利推进提供了坚实的理论基础。

经过一学期的努力，已经完成了城市智慧公交系统的公交实时位置查询、公交空余座位数查询、公交数据报表分析等核心功能，以及公交换乘、站点信息查询、线路查询等基本功能。在完成功能开发的基础上，还对系统做了性能上的优化，保证系统能够稳定运行。

7.2 收获与感悟

本系统的开发使我具备了独立开发一个软硬件结合系统的能力，熟练掌握了常用的开发工具和开发框架。复习了本科阶段学习的程序开发、数据库、软件工程、操作系统、嵌入式、软件文档写作等相关课程。毕设系统开发和课堂学习相比，更注重主动学习，更加考验沟通交流能力。在互联网技术日新月异的当下，我认识到了终身学习的重要性，在系统开发中，我主动学习使用了最新的技术栈。之后的学习开发中，我更加注重对需求的分析，做到对开发过程心里有数再进行具体的开发，对软件的可靠性和易用性做出了规划。同时在开发工程中要加强法律意识和知识产权意识，尊重其他开发者劳动成果，支持正版软件和工具。

7.3 存在的问题及下一步工作

7.2.1 存在问题

由于城市公交系统的复杂性和庞大性，以及开发的人力、物力、财力有限，本系统仅作了单一节点的模拟。对于在真实场景中部署本系统，还需要结合实际情况进一步开发和调优。由于缺少专业美工，系统界面使用了iOS原生的控件和前端模版，系统界面在美观性方面稍有欠缺。

7.2.2 未来展望

本系统中物联网节点收集到的公交实时数据，可以帮助公交运营公司制定更加合理的公交线路和调度规则，提高公交系统的客运效率。在后期的迭代版本中，将借助机器学习、数据挖掘、数据可视化等手段，完善公交大数据的开发。此外，后期将对系统作进一步的性能优化，使用Nginx实现服务器集群的负载均衡，以数据库集群、数据散列等形式优化数据存储，用Redis等非关系型数据库将实时数据缓存在内存中，确保系统在高并发和长时间的工作场景下稳定运行。

参考文献

- [1] 潘东飞, 黄建明. 基于 iOS 的智能交通系统的设计与实现[EB/OL]. 北京: 中国科技论文在线. [2014-07-25]. <http://www.paper.edu.cn/releasepaper/content/201407-309>.
- [2] 朱民, 王可心, 田茂云. 城市公交查询系统发展研究[J]. 电脑编程技巧与维护, 2012(14): 71-72.
- [3] 何宇, 林晓焕, 郭靖. iOS 平台上实时查询公交的设计与实现[J]. 微处理机, 2014(3): 60-62.
- [4] 陆化普, 李瑞敏. 城市智能交通系统的发展现状与趋势[J]. 工程研究-跨学科视野中的工程, 2014(1): 6-19.
- [5] Zanella A, Bui N, Castellani A, et al. Internet of Things for Smart Cities[J]. Internet of Things Journal IEEE, 2014, 1(1): 22-32.
- [6] 白雪飞. 基于 IOS 的智能公交信息服务系统的研究[D]. 北京邮电大学, 2014.
- [7] 张国军, 郑丽媛, 张俊. 基于物联网的瓦斯监控系统[J]. 传感器与微系统, 2013, 32(1): 125-127.
- [8] 王永建, 李书森. 面向智慧城市的智慧公交监管平台设计[J]. 电信快报, 2016(11): 7-11.
- [9] 张帅, 张愉, 尚兆洁. iOS 常用架构模式分析[J]. 工程技术:文摘版, 2016(1): 00159-00159.
- [10] Lee S, Tewolde G, Kwon J. Design and implementation of vehicle tracking system using GPS/GSM/GPRS technology and smartphone application[C]// Internet of Things. IEEE, 2014: 353-358.
- [11] 郭晋. 北斗与 GPS 导航定位系统对比分析研究[J]. 科技资讯, 2011(28): 60-60.
- [12] 刘前刚. GPS 定位算法及其在智能公交中的应用[D]. 湖南大学, 2009.
- [13] 张春华, 谢永军, 周政毅, 等. 公共场所人数检测统计系统的设计[J]. 广东工业大学学报, 2012, 29(3): 63-67.
- [14] 肖佳, 杨波. 基于热释电红外传感技术的目标定位研究[J]. 红外, 2011, 32(12): 17-22.
- [15] 杨扬, 李昌平, 孙燕. 基于 51 单片机的会场人数监控仪[J]. 湖南农机:学术版, 2011, 38(4): 53-54.
- [16] 东辉, 唐景然, 于东兴. 物联网通信技术的发展现状及趋势综述[J]. 通信技术, 2014(11): 1233-1239.
- [17] 范兴隆. ESP8266 在智能家居监控系统中的应用[J]. 单片机与嵌入式系统应用, 2016, 16(9): 52-56.
- [18] Rasmus Lerdorf, Kevin Tatroe. PHP 程序设计[M]. 邓云佳. 北京: 电子工业出版社, 2007.
- [19] BaronSchwartz, PeterZaitsev, VadimTkachenko. 高性能 MySQL[M]. 宁海元, 宁海元, 周振兴, 彭立勋. 北京: 电子工业出版社, 2013.
- [20] David Mark, Jack Nutting, Kim Topley, Fredrik Olsson, Jeff LaMarche. 精通 iOS 开发[M]. 周庆成, 邓强, 武海峰, 蒋振华. 北京: 人民邮电出版社, 2014.
- [21] David Flanagan. JavaScript 权威指南[M]. 李强. 北京: 中国电力出版社, 2012.
- [22] 朱文轩. 智能公交查询系统的设计与实现[D]. 郑州大学, 2014.
- [23] 王昊. 一种电子地图最短路径算法研究[J]. 硅谷, 2009(5): 70-71.

致谢

在毕业设计完成之际，我衷心感谢指导教师田贤忠教授对本项目提供的支持，感谢身边同学对项目提出意见和建议，感谢浙江工业大学图书馆对本项目提供文献支持。

附录

附录 1 毕业设计文献综述

附件 2 毕业设计开题报告

附件 3 毕业设计外文翻译（中文译文与外文原文）