

单位代码: 10293 密 级:

南京邮电大学

# 专业学位论文



论文题目： 基于 wifi 的物联网智能楼宇控制系统  
的设计与实现

学 号 1215012231

姓 名 周栋梁

导 师 朱洪波教授

专业学位类别 工程硕士

# 类型全目制

专业（领域） 电子与通信工程

论文提交日期 2018 年 3 月

# **Design and Implementation of IoT Intelligent Building Control System Based on Wifi**

Thesis Submitted to Nanjing University of Posts and  
Telecommunications for the Degree of  
Master of Engineering



By

Dongliang Zhou

Supervisor: Prof. Hongbo Zhu

March 2018

## 南京邮电大学学位论文原创性声明

本人声明所呈交的学位论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得南京邮电大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

本人学位论文及涉及相关资料若有不实，愿意承担一切相关的法律责任。

研究生学号：\_\_\_\_\_ 研究生签名：\_\_\_\_\_ 日期：\_\_\_\_\_

## 南京邮电大学学位论文使用授权声明

本人承诺所呈交的学位论文不涉及任何国家秘密，本人及导师为本论文的涉密责任并列第一责任人。

本人授权南京邮电大学可以保留并向国家有关部门或机构送交论文的复印件和电子文档；允许论文被查阅和借阅；可以将学位论文的全部或部分内容编入有关数据库进行检索；可以采用影印、缩印或扫描等复制手段保存、汇编本学位论文。本文电子文档的内容和纸质论文的内容相一致。论文的公布（包括刊登）授权南京邮电大学研究生院办理。

非国家秘密类涉密学位论文在解密后适用本授权书。

研究生签名：\_\_\_\_\_ 导师签名：\_\_\_\_\_ 日期：\_\_\_\_\_

## 摘要

智能楼宇作为发展智慧城市的重要体现，已经成为当今社会的热议话题。为了实现楼宇智能化，解决楼宇远程控制问题，本文研究基于 wifi 的物联网智能楼宇控制系统，介绍该系统的设计与实现。物联网智能楼宇控制系统即以智能楼宇为背景，采用物联网技术通过实现上位机控制平台与 Arduino 嵌入式设备之间的通信的控制系统。该控制系统实现对智能终端的无线控制与检测，该系统分成两个部分：Arduino 嵌入式设备和上位机控制平台。

Arduino 嵌入式设备由 Arduino UNO R3 开发板、ESP8266 无线传输模块和传感器节点组成。Arduino UNO R3 开发板为主开发板，通过对其编程实现对传感器的控制与检测。ESP8266 无线传输模块功能在于与上位机控制平台之间实现信息交互，接收控制指令或发送反馈信息。ESP8266 无线传输模块通过串口和 Arduino UNO R3 开发板之间进行数据交互；传感器节点用于模拟智能楼宇中的智能终端，通过传感器检测环境数据反馈到 Arduino 嵌入式设备，实现控制与检测。

上位机控制平台采用 B/S 架构以及 MVC 设计模式，包括前端页面、后端服务程序和数据库三个部分。前端页面通过 JSP 技术实现，后端服务程序使用 Java 进行开发，数据库使用 MySQL，控制平台整体上使用 Struts+Spring+Mybatis 的整合框架。采用 ESP8266 无线传输模块构建无线局域网，通过后端 Java 程序实现 Socket 通信算法，从而实现上位机控制平台与 Arduino 嵌入式设备之间的无线通信。

实验测试表明，基于 wifi 的物联网智能楼宇控制系统较好地实现了对智能终端的无线控制与检测，具有控制准确、检测准确、通信可靠和系统延迟较低的特点，基本满足了智能楼宇的需求。

**关键词：**智能楼宇，Arduino，无线控制，传感器

# Abstract

As an important embodiment of developing smart city, intelligent building has become a hot topic in today's society. In order to realize the intelligence of buildings and solve the problem of remote control of buildings, this thesis researches a intelligent building control system based on wifi, and introduces the design and implementation of the system. The intelligent building control system based on wifi realizes the wireless control and detection of intelligent terminal, which is divided into two parts: Arduino embedded device and upper computer control platform.

Arduino embedded device consists of a Arduino UNO R3 development board, a ESP8266 wireless transmission module and sensor nodes. The Arduino UNO R3 development board is the main development board, which realizes the control and detection of the sensors through programming. The ESP8266 wireless transmission module is mainly responsible for the information interaction with the upper computer control platform, receiving control instructions or sending feedback information. The ESP8266 makes data interaction through serial ports with Arduino UNO R3 development board; The sensor nodes are used to simulate intelligent terminal in intelligent building, and feedback the environment data to the Arduino embedded device through sensors, realizing control and detection.

The upper computer control platform adopts the B/S architecture and the MVC design pattern, including the front-end page, back-end service program and database. The front-end page is implemented through JSP technology, the back-end service program is developed using Java, the database chooses MySQL, and the control platform selects the integrated framework of Struts+Spring+Mybatis as a whole. Using the ESP8266 wireless transmission module to build a wireless LAN, the Socket communication algorithm can be realized through the back-end Java program, thereby realizing the wireless communication between the upper computer control platform and the Arduino embedded device.

Experimental tests show that the Internet of things intelligent building control system based on wifi well realized the wireless control of the intelligent terminal and testing, with the characteristics of accurate control, accurate detection, reliable communication and low system delay, which basically meets the requirements of intelligent building.

**Key words: Intelligent Building, Arduino, Wireless Control, Sensor**

# 目录

专用术语注释表 .....	V
第一章 绪论 .....	1
1.1 研究背景 .....	1
1.2 研究目的和意义 .....	2
1.3 国内外研究状况 .....	3
1.4 研究内容及本文结构 .....	4
1.5 本章小结 .....	6
第二章 基于 wifi 的物联网智能控制系统及其关键技术 .....	7
2.1 智能控制系统简介 .....	7
2.2 智能控制系统总体设计 .....	7
2.2.1 物联网智能控制系统的整体架构 .....	8
2.2.2 传感节点层 .....	8
2.2.3 嵌入式平台层 .....	8
2.2.4 上位机界面层 .....	9
2.3 智能控制系统的关键技术 .....	9
2.3.1 无线数据传输技术 .....	9
2.3.2 控制算法 .....	10
2.3.3 Web 框架技术 .....	18
2.4 本章小结 .....	20
第三章 基于 wifi 的物联网智能控制系统的硬件设计 .....	21
3.1 开发平台的选择 .....	21
3.2 基于 Arduino 的智能终端的硬件设计 .....	22
3.2.1 硬件介绍 .....	22
3.2.2 Arduino 与智能终端模块之间接口设计 .....	23
3.2.3 Arduino 与无线传输模块之间接口设计 .....	25
3.3 基于 wifi 的智能终端的硬件设计 .....	26
3.3.1 硬件介绍 .....	26
3.3.2 ESP8266 无线传输模块设计 .....	27
3.4 本章小结 .....	28
第四章 基于 wifi 的物联网智能控制系统的软件设计 .....	29
4.1 Arduino 嵌入式平台的软件设计 .....	29
4.2 模糊控制算法 .....	32
4.3 上位机控制平台的软件设计 .....	34
4.3.1 无线控制的设计与实现 .....	34
4.3.2 软件实现 .....	36
4.3.3 上位机界面软件设计 .....	41
4.4 本章小结 .....	41
第五章 基于 wifi 的物联网智能控制系统的实现和测试 .....	43
5.1 嵌入式平台的实现 .....	43
5.2 上位机控制平台界面的实现 .....	44
5.3 系统测试 .....	46
5.3.1 测试指标 .....	47
5.3.2 测试结果与分析 .....	49
5.4 本章小结 .....	50

第六章 总结与展望 ..... 51

    6.1 工作内容与成果 ..... 51

    6.2 未来展望 ..... 51

参考文献 ..... 53

附录 1 程序清单 ..... 55

附录 2 攻读硕士学位期间撰写的论文 ..... 56

附录 3 攻读硕士学位期间申请的专利 ..... 57

附录 4 攻读硕士学位期间参加的科研项目 ..... 58

致谢 ..... 59

## 专用术语注释表

wifi	Wireless Fidelity	IEEE802.11b 无线局域网
IoT	Internet of things	物联网
App	Application	应用程序
B/S	Browser/Server	浏览器/服务器
C/S	Client/Server	客户机/服务器
MVC	Model View Controller	模型-视图-控制器
IEEE	Institute of Electrical and Electronics Engineers	电气和电子工程师协会
JSP	Java Server Pages	Java 服务端网页
PDA	Personal Digital Assistant	掌上电脑
USART	Universal Synchronous/Asynchronous/Receiver/Transmitter	通用同步/异步串行接收/发送器
MCU	Microcontroller Unit	微控制单元
AP	Access Point	访问接入点
STA	Station	站点
PID	Proportion/Intergral/Derivative	比例-积分-微分控制器
PWM	Pluse Width Modulation	脉冲宽度调制
JDK	Java Development Kit	Java 软件开发工具包
JRE	Java Runtime Environment	Java 运行环境
PC	Personal Computer	个人计算机
ISP	Internet Service Provider	互联网服务提供商
USB	Universal Serial Bus	通用串行总线
LDR	light-dependent resistor	光敏电阻
FAM	Fuzzy Associative Memory	模糊联想记忆
AVR	Audio-video Receivers	视频音频接收机
UNO	Universal Network Object	通用网络对象



# 第一章 绪论

## 1.1 研究背景

随着人类社会的不断进步，城市将来会负担更多的人口数量。当前，我国正处于城镇化迅速成长的阶段，部分地区“城市病”情况越来越严重。为了解决城市建设难题，达成城市可持续发展的目标，构建智慧城市已变为如今世界城市发展不可逆转的趋势。

在 2010 年，IBM (International Business Machines Corporation) 正式提出了“智慧的城市”概念，旨在为全球和中国的城市发展贡献自己的作用。IBM 经过调研判定，城市由六个核心系统构成，这六个核心系统分别是：组织（人）、业务/政务、交通、通讯、水和能源。这些系统不是独立存在的，而是以一种协同的方式互相关联。而由这些系统所构成的宏观系统就是城市本身。

智能楼宇的建设就是发展智慧城市的一个很好的体现。建筑业，房地产业已经将智能楼宇视为寻求可持续发展的战略目标。智能技术在楼宇中的应用，能够有效提升人们的生活环境，节能社会型建设和促进建筑业可持续发展<sup>[1]</sup>。1984 年美国 Connecticut 州的 Hartford 市开创了智能楼宇建筑的先河<sup>[2]</sup>。Hartford 市的这栋智能楼宇在普通楼宇的基础上补充了计算机设备、程控交换机、数据通信线路等智能硬件。大楼的全部空调、排水、供配电设备、防火、保安设备都使用计算机实现控制，完成自动化、信息化，使大厦能力实现质的飞跃，从而创立了世界上第一座智能化楼宇。美国智能建筑研究院将智能楼宇定义为：将建筑布局，设备，服务和管理结合，实现最佳组合，创建高效，舒适，便捷，安全的建筑。智能楼宇被欧洲智能建筑组织解释为构建一个用户可以通过最低的成本获得最高的效率来经营自己的资源的建筑。

我国智能楼宇的发展起步相对较晚，不过在国家政府的支持推动及相关技术专家多年的研究下，我国在智能楼宇建筑的理论技术、现场施工操作、技术验收等方面都拥有了很大的创新。而且在国内地产行业的迅猛崛起的背景下，楼宇控制自动化市场也慢慢得到补充和开拓，智能工具为用户提供了方便舒适的生活，所以获得了用户的一致认可<sup>[3]</sup>。

智能控制系统是智能楼宇中最为核心的环节。智能楼宇采用先进的计算机控制技术，通过移动设备或者 PC 实现对智能终端（传感器）的无线控制和检测。

传统的楼宇控制系统大多采用有线连接方式，依靠有线连接开关实现控制，随着楼宇的环境变化，需要重新布线，而且控制线路也容易老化，导致控制系统频繁出现故障，后期维

修也较为复杂。与传统楼宇控制系统相比,通过无线控制楼宇就极为简便,它不会受到空间、时间限制。智能楼宇控制系统采用 wifi 作为智能楼宇控制系统的无线传输方式,能够拥有更宽的带宽、更强的射频信号、更低的功耗以及更好的安全性。仅需在手机或移动设备上安装一个软件,就可以远距离实现控制楼宇系统。设计智能报警模块就能便捷地排除故障,及时发现安全隐患。

本文研究的智能楼宇控制系统由于其良好的扩展性,适用于高层楼宇。文中采用上位机无线控制智能终端模块的方式,实现控制系统的智能化,通过模拟智能楼宇中的照明控制的演示系统为例测试智能楼宇控制系统方案的可行性。

随着通信技术和网络技术的高速发展,各种新型信息技术正在改变我们的生活,促进生活现代化的步伐,也极大的提高了我们的生活条件。物联网则是新一代信息技术的核心之一,更是“信息化、智能化”时代发展的关键技术。物联网的英文全称是:“Internet of things(IoT)”<sup>[4]</sup>。从名称中可以看出,物联网包含有两层意思:第一,物联网是基于传统的互联网而发展起来的,是对我们现在的互联网的扩展和提升;第二,物联网与互联网是完全不一样的概念,物联网将用户拓展到了物品,达到了物与物之间的交流与通信的目的。物联网采用智能感知、识别技术与普适计算等通信感知技术,被广泛的应用于网络的融合中,因此物联网也被称为继计算机、互联网之后世界信息产业发展的第三次浪潮<sup>[5]</sup>。

智能楼宇中物联网技术是不可或缺的关键技术。本文设计的智能楼宇控制系统中智能终端模块是由传感器与 Arduino 等开发板组成,上位机与智能终端模块通过 ESP8266 无线传输模块发射 wifi 信号通信,采用物联网技术,实现了物与物之间的通信。

## 1.2 研究目的和意义

智能楼宇作为发展智慧城市的一个重要组成部分,它更为具体的表现了发展智慧城市的理念。与传统的控制系统比较,智能楼宇中的智能控制系统体现了环保、绿色、低碳和以人为本的思维,且基于 wifi 的智能控制系统的智能、人性化程度更高。首先,根据不同场景、不同环境设定不一样的控制模式和管理策略,智能控制系统自动的采取多样化的控制策略组合,从而达到当前环境下,最让人舒适的环境;智能控制系统能够参考不同人的不同行为特征,训练学习,选用不同的控制方案,达到人性化设置并解决能源的效果。其次,通过便捷的管理平台,提供对多种智能硬件设备的总体调控和监视,并可以依据便携式手机 app,随时随地查看智能楼宇中智能模块工作状态、安全情况,从而实现安全可控。第三,充分发挥 wifi 无线网络技术优势, wifi 具有接入门槛低、覆盖面广、传输速度快等特点,通过使用 wifi

通信, 智能楼宇控制系统可以与市面上较多智能控制模块兼容, 扩展性更好, 可以提供更为人性化的服务。

在智能楼宇快速发展的大背景下, 基于 wifi 的智能楼宇控制系统的优势有以下几点<sup>[6]</sup>:

- (1) 人性化设计, 系统能够完成多种报警信息触发环境的设计。设备出现问题会产生报警信息, 并给用户提供具体导致故障的原因, 从而减少了维修周期。
- (2) 节约能耗, 设施的自动化控制保证电器和照明等设备能够提前完成定时, 让设备实现自控, 当到达预设的时间, 设备本身将采取开启或关闭动作。同时, 用户可以随时掌握智能终端的信息, 从而使用户能够随时掌握系统的信息数据, 操控每一个设备。
- (3) 节约人力资源。智能系统的设计, 预防了用户不断检查硬件设施, 并对其进行操作, 减少了设备管理所需要的人力资源。

完整的智能楼宇控制系统需要具备普通楼宇中的正常要求, 如人脸识别门禁、烟雾报警系统、照明系统、温控系统等。因此, 设计并实现一套完整的智能楼宇控制系统, 可以满足用户对各种传感器的多样化需求。

### 1.3 国内外研究状况

美国智能建筑研究院把智能楼宇解释为: 将建筑结构, 设备, 服务和管理相结合, 实现最佳组合, 构建高效, 舒适, 便捷, 安全的建筑。智能楼宇被欧洲智能建筑组织诠释为用户可以通过最少的代价发挥最高的功效来控制自己的资源。智能楼宇可以提供快速响应, 高效率和支持强大的环境<sup>[7][8]</sup>。在中国, 智能楼宇设计标准指的是智能建筑是实现建筑设备, 办公自动化和通信网络系统的平台<sup>[9][10]</sup>。

在信息技术快速发展的环境下, 楼宇的智能化获得了迅猛的发展。多种多样的智能化楼宇不断涌现, 比较有代表性的诸如: 北京的奥运会体育场馆、上海世博会中的中国馆等。伴随着智能建筑行业市场的扩大, 智能楼宇的各种控制产品在商品房、办公楼、会议中心、医院、体育馆等领域不断发展。

从智能楼宇控制系统的发展进程来看, 它主要经过了四个阶段, 第一个阶段为单一的功能控制阶段, 第二个阶段是多功能集散控制阶段, 第三个阶段是系统集成控制阶段, 第四个阶段是网络集成控制阶段。

智能楼宇控制系统发展历程如下图 1.1 所示:

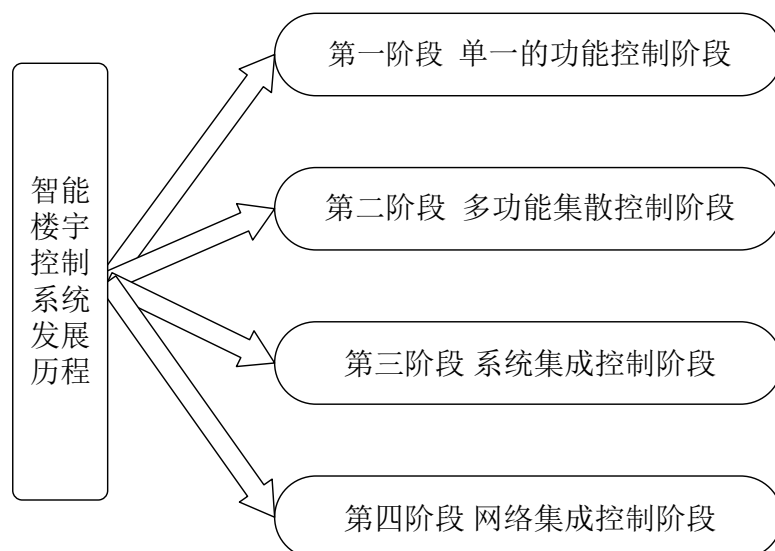


图 1.1 智能楼宇控制系统发展历程

在第一个阶段只有少数几个比较关键的设备通过总线与中央监控设备（计算机、显示器以及键盘）连接，实现中央监控设备的综合操控与显示；在微处理器技术发展的背景下，第二个阶段设备在中央主监控中心的基础上，还有用于管理设备的二级分控制节点，这些分控制节点也能够完成部分控制作用；在微处理器的能力不断获得提升，现场总线本领也得到迅猛发展的背景下，在智能楼宇控制系统发展的第三阶段，现场总线、微处理器与中央控制系统等技术被结合使用，从而达到各个分节点集中管理、分散控制的目的；在第四个阶段，随着互联网技术与无线通信技术的发展，监控的数量以及监控方法都有较大改变，在这个阶段控制的灵活性被大大提升，不仅能同时实现现场层、自动化层到管理层的一体化控制，还能使用户便捷地通过不同的操作平台完成设备的远程管理与监控<sup>[1]</sup>。因为本文研究的控制系统控制灵活，用户可以通过不同操作平台实现设备远程管理与监控，因此当前研究属于第四阶段。

## 1.4 研究内容及本文结构

本文从讲述发展智慧城市的大背景入手，然后根据国内外智能楼宇的发展历程，讲述了智能楼宇的提出和发展经历。然后从智能控制系统具体比较智能控制系统与当前使用的控制系统的差别和优势。在这些大背景下，本文设计了一个基于 wifi 的通用的智能楼宇控制系统。这个控制系统能够兼容各种传感器设备，其扩展性保证了控制系统的实用性。在此之后，本文对整套环境的总体设计及所使用到的关键技术做了简单的讲解。本文在第三章和第四章对系统的硬件设计和软件设计做了细致的介绍。在第五章通过搭建一套智能控制系统的测试开发环境，证明本文提出控制系统的可行性。本文结构框图如图 1.2 所示：

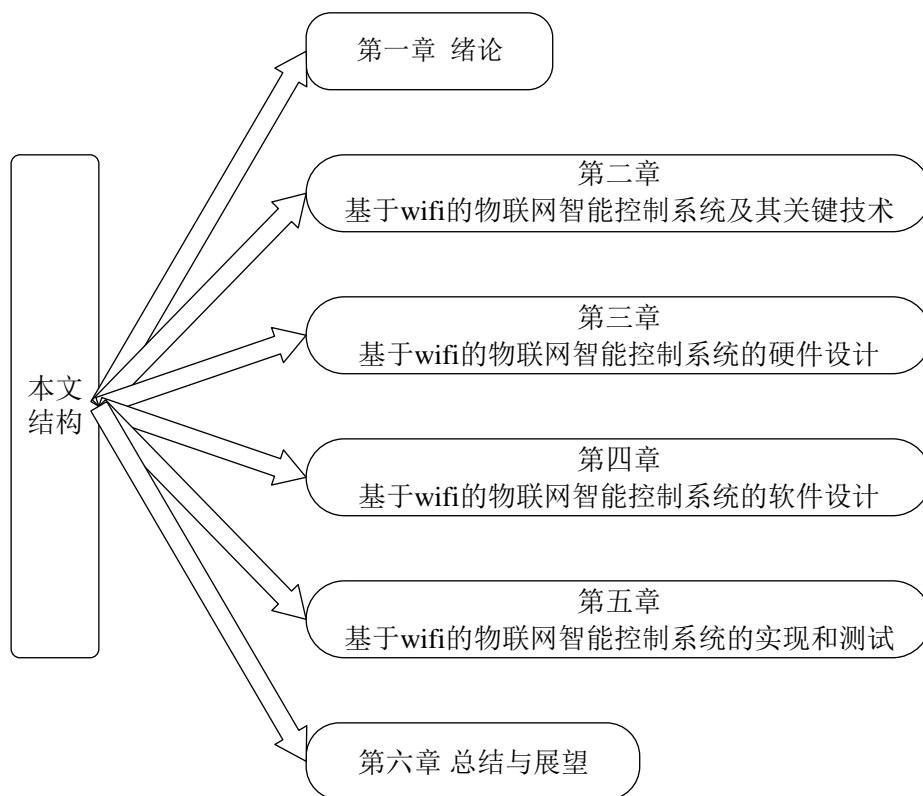


图 1.2 本文结构框图

本文的内容和结构如下：

第一章 绪论，讲述了当今研究热点智慧城市的发展，研究智能楼宇的重要性，然后介绍了智能楼宇、智能控制系统的背景、研究的意义和国内外研究现状。

第二章 基于 wifi 的物联网智能控制系统及其关键技术，这章首先简单的介绍了智能控制系统，对设计的整套控制系统的一个简单分析。然后通过智能控制系统的总体设计和架构图，将整个系统分为三层，分别是：传感节点层、嵌入式平台层、上位机界面层。通过不同的层面结合系统架构图，直观地介绍各个层面使用的关键技术及实现方法，为后面对这些技术的使用做理论铺垫。

第三章 基于 wifi 的物联网智能控制系统的硬件设计，文中研究基于智能楼宇设计的控制系统，在设计时是通过一个个的传感器实现智能楼宇中的开关、报警、测量等功能。本章在设计传感器控制器时选用的是 Arduino 微控制器实现对智能终端的控制，加上 ESP8266 无线传输模块实现智能终端与上位机的通信。在本章详细介绍了选用这些硬件的原因与优势，以及各个模块之间的接口设计，为第五章的系统实现和测试提供硬件基础。

第四章 基于 wifi 的物联网智能控制系统的软件设计，本章主要介绍 Arduino 微控制器的检测算法设计、socket 通信的软件设计和上位机界面设计，并对模糊控制算法进行了深入研究，为第五章的系统实现和测试提供软件基础。

第五章 基于 wifi 的物联网智能控制系统的实现和测试,本章对研究的智能控制系统完成了部署,实现了采用光敏传感器情况下的智能控制,并对各个模块功能和性能进行了测试,证实了本文系统的可行性。

第六章 总结与展望,本章总结了本文的研究内容和取得的成果,对基于 wifi 的物联网智能控制系统提出了可以改进的建议,最后对基于 wifi 的智能楼宇控制系统的未来发展进行了展望。

## 1.5 本章小结

本章从介绍发展智慧城市的大背景入手,接着比较国内外智能楼宇的发展,讲述智能楼宇的提出和发展经历。紧接着从智能控制系统层面具体阐述文中研究的智能控制系统与目前使用的控制系统的区别和优势。在这些大背景下,本文设计了一个基于 wifi 的通用的智能楼宇控制系统,这个控制系统可以控制多种传感器设备,通过 Arduino 微控制器接收传感器设备传递的数据,分析处理后由上位机控制,并存入数据库中。多个传感器设备组成的智能终端与上位机之间通过 wifi 的传输方式。本章还阐述了本文的研究目的和对社会带来的积极意义。在本章结尾总结了本文的主要研究内容和结构。

## 第二章 基于 wifi 的物联网智能控制系统及其关键技术

基于 wifi 的物联网智能控制系统及其关键技术重点介绍了智能控制系统的总体设计和构建智能控制系统时涉及到的关键技术。其中智能控制系统的总体设计介绍了该智能控制系统的整体架构,该系统的设计分为三层,分别是:传感节点层、嵌入式平台层和上位机界面层。智能控制系统中采用的关键技术有无线数据传输技术、控制算法、Web 框架技术等。

### 2.1 智能控制系统简介

智能控制系统由两部分组成,分别是 Arduino 嵌入式设备和上位机控制平台,整套控制系统采用 wifi 完成上位机对智能终端模块的控制。

Arduino UNO R3 开发板、ESP8266 无线传输模块和智能终端模块构成整个 Arduino 嵌入式设备。Arduino UNO R3 开发板为主开发板,通过对其编程实现 LED 亮灭控制与检测算法。ESP8266 无线传输模块主要负责与上位机控制平台之间进行信息交互,接收控制指令或发送反馈信息。并且,ESP8266 通过串口和 Arduino UNO R3 开发板之间进行数据交互;智能终端模块的多种传感器采集的信息在内部处理器中进行检测判断,然后上位机控制平台通过 wifi 控制智能终端模块,实现控制和显示。

上位机控制平台的架构采取 B/S (Browser/Server, 浏览器/服务器) 架构,同时利用 MVC (Model View Controller, 模型-视图-控制器) 设计模式,包括了前端页面、后端服务程序和数据库三个部分。前端页面通过 JSP 技术实现,后端服务程序使用 Java 进行开发,数据库使用 MySQL,控制平台整体上使用 Struts+Spring+Mybatis 的整合框架。使用 ESP8266 无线传输模块构建无线局域网,通过后端 Java 程序实现 Socket 通信算法,从而实现上位机控制平台与 Arduino 嵌入式设备之间的无线通信。

### 2.2 智能控制系统总体设计

智能楼宇中的控制系统一般包含控制端与被控制端,控制端则需要有处理器。本文研究的智能控制系统包含传感节点层、嵌入式平台层和上位机界面层。传感节点层是由多个传感器组成的,模拟智能楼宇中灯、烟雾报警器、人脸识别门禁等智能设备,传感器节点层支持传感器扩展,适用于高层楼宇;嵌入式平台层主要功能是实现对智能终端模块的控制与数据采集,并将数据通过无线传输模块上传到上位机平台;上位机界面层主要用于图形化显示智

能终端的运行状态，并将智能终端采集的数据直观显示出来，从而达到自动化控制、报警等功能。

### 2.2.1 物联网智能控制系统的整体架构

本文提出了一种易于操作、扩展性强、支持 wifi 无线传输技术的通用的智能楼宇环境下的控制系统。如图 2.1 所示为系统的整体架构框图。

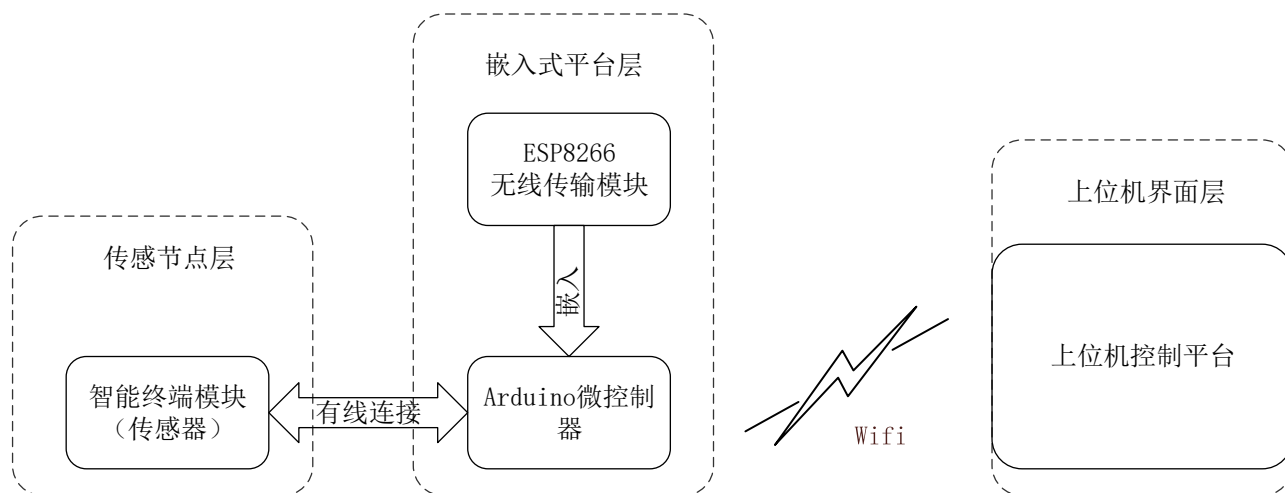


图 2.1 基于 wifi 的物联网智能控制系统整体架构图

从图 2.1 中可以看出，本系统主要有底层的传感器节点、中间的嵌入式平台和最上层的上位机控制平台三部分组成。下面将对这三层逐个介绍。

### 2.2.2 传感节点层

传感节点处于本系统的底层。依据实际情况，智能楼宇中传感器节点采取的传感器主要有光照传感器、温度传感器、烟雾传感器、湿度传感器等。本系统适合在多种智能楼宇中使用，是一个通用的控制系统。其中光照传感器可以根据对室内外的亮度感应，自动调节楼宇灯光强度；温度传感器可以通过采集温度信息，调节楼宇的空调系统；烟雾传感器可以控制楼宇的烟雾报警系统，防备火灾等。

### 2.2.3 嵌入式平台层

嵌入式平台层处于本系统的中间层，包括 Arduino 微控制器模块和 ESP8266 无线传输模块。Arduino 微控制器模块主要功能是通过控制算法，对各个传感节点控制，并从串口解析出下方传感节点传输过来的各种数据，ESP8266 无线传输模块主要负责与上位机控制平台之间



进行信息交互，接收控制指令或发送反馈信息。ESP8266 和 Arduino 微控制器之间通过串口通信和传输数据。文中 ESP8266 无线传输模块设计为一个无线传输设备，在与上位机建立 socket 通信的基础上，实现终端模块与上位机通过 wifi 连接。

## 2.2.4 上位机界面层

上位机界面层位于本系统的最上层。上位机界面层采用了 B/S 架构以及 MVC 设计模式，包括了前端页面、后端服务程序和数据库三个部分。前端页面选用 JSP 技术实现，后端服务程序利用 Java 进行开发，数据库部分采用 MySQL，上位机界面层整体上使用 Struts+Spring+Mybatis 的整合架构。使用 ESP8266 无线传输模块构建无线局域网，通过后端 Java 程序实现 Socket 通信算法，从而实现上位机控制平台与 Arduino 嵌入式设备之间的无线通信。上位机界面层采用图形化界面，达到对智能终端模块的无线控制和状态显示的目的。上位机界面层包含了用户登录界面、管理主界面、电路状态显示界面、通信数据显示界面、用户管理界面等。作为操控界面的管理主界面囊括了查看电路状态、管理状态数据、管理通信数据、管理用户数据、注销登录等功能按钮。上位机系统可以安装在 PC 或者在移动设备通过地址访问，实现便携式远程控制。

## 2.3 智能控制系统的关键技术

本系统涉及到了多种硬件和软件的技术。这些技术包括：无线数据传输技术、控制算法嵌入式开发技术、Java 编程语言、B/S 体系结构、数据库技术等。

### 2.3.1 无线数据传输技术

当前常用的短距离无线数据传输技术大体分为三种，分别是 wifi 技术、蓝牙技术和 ZigBee 技术，这三种短距离无线数据传输方式在实际应用中也各有优缺点。

ZigBee 通信技术遵循国际标准 IEEE802.15.4，目前的应用领域包括：无线传感器监测网络、个人监护仪器、低功耗无线设备、电器自组网等。ZigBee 的使用主要针对需要低能耗的领域，它的使用优点有：低成本、自带无线网络和路由功能、能耗低。然而 ZigBee 通信方式的数据传输速度较慢。

蓝牙通信技术遵循国际标准 IEEE802.15.1，通信范围一般可达 10m，与 ZigBee 通信技术一样，也具有成本低廉和功耗较低的优点。虽然蓝牙设备的传输速度介于 ZigBee 和 wifi 技术

之间,但是仍然较慢,不适合在要求传输速度很高的场景使用。

wifi 通信技术遵循国际标准 IEEE802.11,是目前使用最普遍的短距离无线通信技术,最大传输距离可达到 300M,速率最快能够达到 300Mbps,基本能满足所有短距离无线数据传输的需求。wifi 设备内部配置有 TCP/IP 协议栈,使得省去中间设备的接口转换步骤就可以直接收发 Internet 数据包<sup>[12]</sup>。

ZigBee 技术、蓝牙技术和 wifi 技术的功能从某种意义上来说是互补的,它们有着各自擅长的领域,短时间内无论哪一种技术都不能被替代或淘汰<sup>[12]</sup>。在不同需求下,这些无线技术都会有所应用。由于 wifi 的特性非常适合于智能终端控制系统领域,因此本文使用 wifi 作为无线数据传输解决方案。

### wifi 通信设计

本文选用 ESP8266 无线传输模块。ESP8266 最初的设计意图就是面向移动设备、物联网应用和可穿戴电子设备,所以采用了多项技术来实现芯片外围设计小型化、系统低功耗。

ESP8266 无线传输模块采用内部集成的电源管理与系统功耗设计,可以切换睡眠模式与工作模式,降低系统的非必需功耗。ESP8266 处于睡眠模式的时候,消耗电流小于 12uA,只开启了 RTC 用于校准和看门狗防止系统死机。当处于 wifi 连接状态,功耗小于 1mA。而且为了进一步的降低功耗,满足可穿戴设备,可以通过编程设置内部集成的功放来降低输出功率适应近距离通信。

本文通过选用基于 wifi 的通信设备达到网络连接和数据通信的目的。在前面的技术内容中,已经分析过三种常见的无线通信方式。可以看出, wifi 对比蓝牙、ZigBee 等其他短距离数据无线通信方式,应用在智能楼宇领域时有较多的优点。

## 2.3.2 控制算法

2012 年,杨金翠在文献<sup>[14]</sup>中提出,决策和控制功能是物联网环境下控制系统的一个显著特征,能够通过感知物理世界,然后传达和处理感知信息,对事件采取判断和决定,再回到控制执行器执行,最后对物理实体状态产生作用,完成从物理世界到信息空间再回到物理世界的循环过程<sup>[15]</sup>。杨金翠在文献<sup>[14]</sup>中得到物联网环境下控制系统的简单 Petri 网<sup>[16]</sup>(Petri 网是对离散并行系统的数学表示)模型,如图 2.2 所示。

图 2.2 中的采集源点通过 S 表达,用  $S_i(i = 1, 2, \dots, n)$  表示详细的数据来源,数据来源由内部来源和外部来源构成,其中内部来源在图中用  $S_1, S_2, \dots, S_k$  表示,外部来源在图中用  $S_{k+1}, \dots, S_n$  表示。

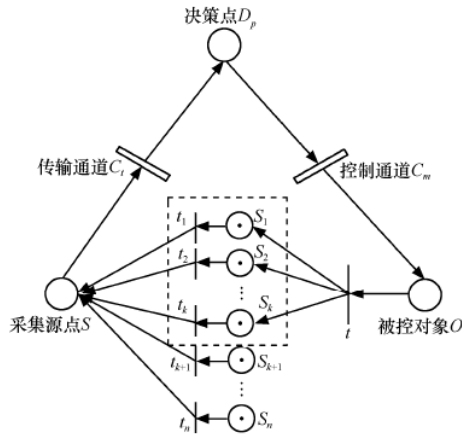


图 2.2 物联网环境下控制系统简单模型

按照功能将决策点划分为管理节点 $C_p$ 、控制源点 $C_s$ 和控制算法 $A$ 。管理节点 $C_p$ 管理控制策略；控制源点 $C_s$ 下达控制命令；控制算法 $A$ 实现控制计算。以此总结除去干扰元素的物联网环境下的控制系统 Petri 网模型，如图 2.3 所示。

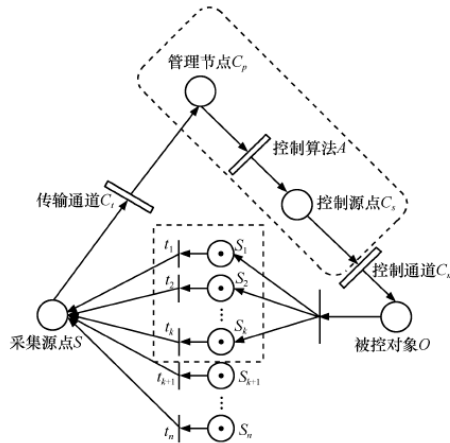


图 2.3 物联网环境下无干扰的控制系统模型

由于物联网环境下的控制系统有来源不确定性、传输安全、算法安全的问题，总结出物联网环境下通用控制系统 Petri 网模型 IoTC，如图 2.4 所示<sup>[14]</sup>。

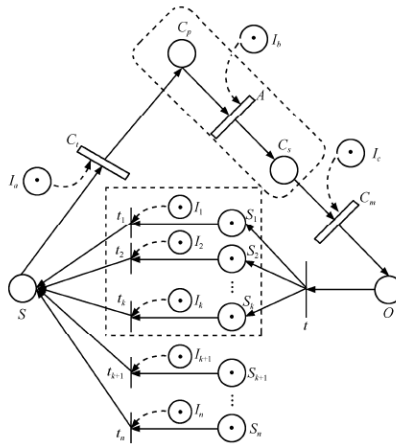


图 2.4 物联网环境下通用控制系统模型 IoTC

图中采集源点故障由 $I_i(i = 1, 2, \dots, n)$ 表示, 采集源点错误种类包含采集源点被攻击或失效。 $I_a$ 、 $I_b$ 、 $I_c$ 代表干扰源,  $I_a$ 代表传输通道被攻击,  $I_b$ 代表算法错误,  $I_c$ 代表控制源点被攻击、失效或控制通道被攻击。

### 模糊控制

1965 年, 美国的著名控制理论学者 L. A. Zadeh 提出了一种模糊集合理论, 该理论与传统控制理论和传统数学完全不同, 该理论对繁琐的系统及其运行过程构建一种语言分析的数学模型, 从而使其变为计算机算法语言<sup>[17]</sup>。

模糊控制是通过对人类人工控制特点进行模拟实现的, 这样即便带有一定主观性和模糊性, 但控制过程相对简单, 控制结果也颇有成效。模糊控制的目标是对复杂的生产加工进行操控, 途径是经过计算机模拟人类的思考处理方式而进行<sup>[18][19][20]</sup>。模糊控制系统如图 2.5 所示。

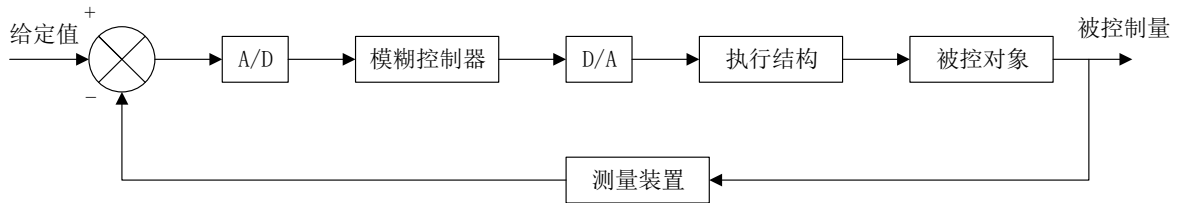
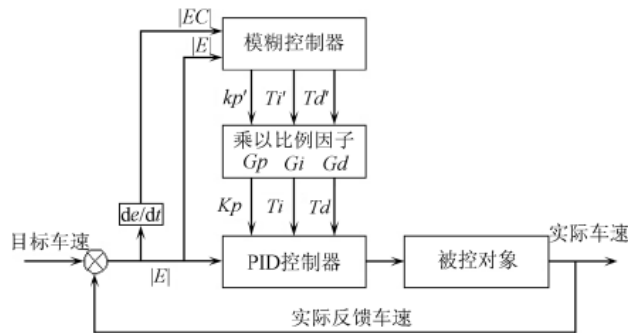


图 2.5 模糊控制系统示意图

2013 年, 仇成群在文献<sup>[21]</sup>中提出了一种选取了传统模糊控制优点的算法, 这个算法提高了系统的控制能力, 该算法是基于 Matlab 和模糊 PID（比例-积分-微分控制器）的自适应模糊控制算法。文中提出的模糊 PID 控制原理图如图 2.6 所示。



注：|E|代表速率误差绝对值；|EC|代表速率误差变化率绝对值； $K_p$ 代表比例系数参数； $K_p'$ 代表比例系数调校参数； $T_i$ 代表积分系数参数； $T_i'$ 代表积分系数调校参数； $T_d$ 代表微分系数参数； $T_d'$ 代表微分系数调校参数； $\otimes$ 代表比较环节； $d_e/d_t$ 代表第 n 次速度误差与第 n-1 次速度误差的差值在采样周期内的变化率（n 为大于 1 的整数）。

图 2.6 模糊 PID 的控制原理<sup>[21]</sup>

如图 2.6 所示, 在模糊控制器的输入端引入速度误差绝对值 $|E|$ 和速度误差变化率的绝对值 $|EC|$ , 模糊输入语言变量通过模糊化过程将其转变为机器可以识别的模糊输入值, 模糊输出值通过模糊规则库推导出, 通过解模糊过程将输出归一化为 0~1 之间精确的控制量 $K_p'$ 、 $T_i'$ 和 $T_d'^{[21]}$ 。在 PID 控制器的应用中需要乘以合适的比例因子 $G_p$ 、 $G_i$ 、 $G_d$ 以获得真正的 PID 数值 $K_p$ 、 $T_i$ 、 $T_d$ 。

2015 年, 王君在文献<sup>[22]</sup>中利用改善了的遗传算法, 采取两输入、三输出的模糊 PID 控制器改良了模糊控制规则和三角形隶属函数底宽, 经过仿真实验解析了改进后的模糊 PID 控制器, 对比了该控制器与一般的 PID 控制器和模糊 PID 控制器之间的能力区别, 研究了基于模糊控制算法的温室远程智能控制系统。

王君在文献<sup>[22]</sup>中利用匹兹堡和实数编码方法, 改进控制器的参数和结构, 使得遗传因子作用在个体完好的模糊控制规则集上, 在个体之间水平度量种群的丰富性。为了达到遗传算法优化模糊控制器的隶属度函数和模糊控制规则集的目的, 讲述了的四种不同的方案:

- 1、已知模糊隶属度函数, 优化模糊控制规则集;
- 2、已知模糊规则集, 对模糊隶属度函数进行优化;
- 3、将模糊隶属度函数和模糊控制的规则集进行分阶段优化;
- 4、同时优化模糊控制规则集和模糊隶属度函数。

由于规则集和隶属函数都可能改变模糊控制的结果, 同时分阶段的优化模糊控制规则集和模糊隶属度函数很难获得整体最优解, 获得的是局部最优解。因此为了避免了他们之间的复杂交互作用, 提高搜索的速度, 配合可行的编码方案和改进的遗传算子, 有利于获得品质较好的模糊控制器, 从而对二者同时进行优化。

2015 年, Rionel Belen Caldo 提出一种基于 Arduino 微控制器的模糊控制调光照明系统设计与开发<sup>[23]</sup>。文中将环境照明和房间照明视为输入。根据系统的占用传感器感应, 检测该区域是否有人, 并控制该传感器自动开启和关闭灯光。控制系统的隶属函数的输出用于确定 LED 灯的调光百分比。采取脉宽调制 (PWM) 技术完成系统的调光功能。由 PWM 操控 LED 灯直流电源的占空比, 因为负载降低的功率取决于占空比的值。控制系统能够依照室内环境亮度主动降低 LED 灯亮度。室内照明基于 IESNA (北美照明工程协会) 约为 300 lux。通过使用该系统, 根据房间所需的亮度来改变光输出, 降低了电力消耗。

如图 2.7 所示, 整个系统将在控制室内的任何一个便利插座上通过电源供电。连接到主电源的占用传感器将用作打开或关闭 LED 驱动器的开关。LED 驱动器是 PWM (脉冲宽度调制) 参与的 LED 灯的直流电源。PWM 供应连接到其他 LED 的其中一个线路的 PWM 驱动器。

由于 Arduino 已经投入运行，它将从光敏电阻（LDR）获得两个数据，这将是控制灯泡输出光的基础。上面提到的两个数据是由 LDR 获取的房间照明和环境光。数据将利用模糊联想记忆（FAM）规则进行模糊化达到调整输出光的目的。

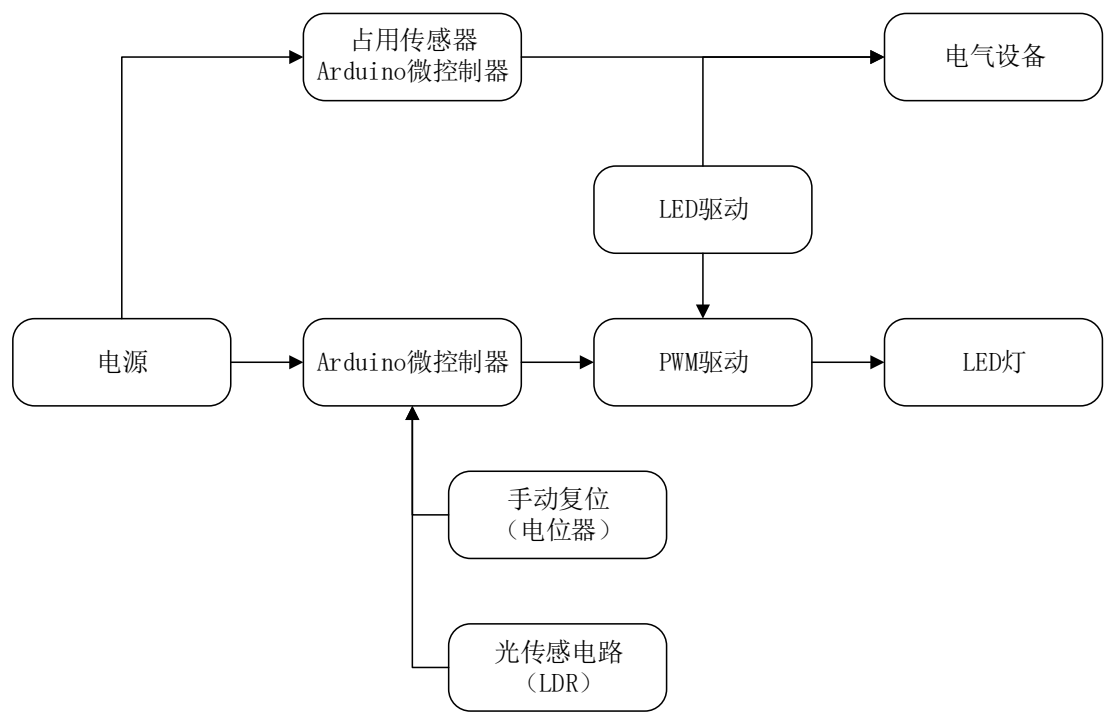


图 2.7 调光系统的一般框图

2017 年，Lina Li 等人在文献<sup>[24]</sup>中提出了一种模糊 PID 控制方法。该方法融合了模糊控制和 PID 控制技术，通过构建模糊 PID 控制器，有效地完成了地铁车站环境温度的合理控制。

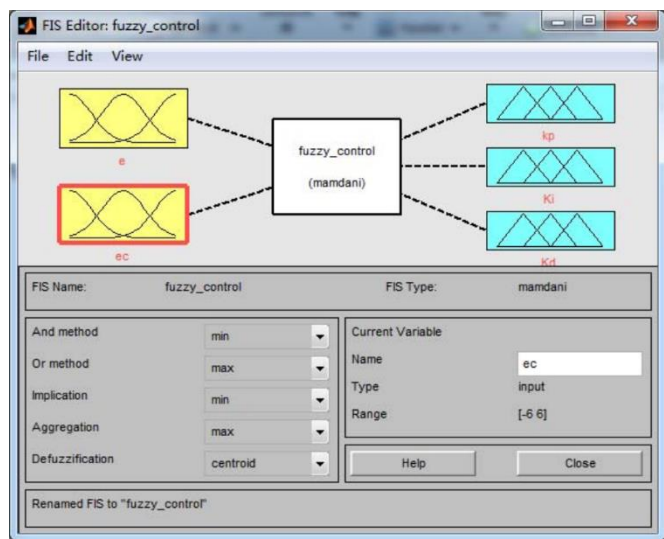


图 2.8 传统模糊推理算法系统

文中通过采用模糊 PID 控制模型，对站内采集的实时数据进行计算，进而实现对地铁空调控制设备的控制，进而确定空调等设备条件，达到恒定的站温度。文中首先通过使用两个输入三个输出形式设计模糊 PID 控制器，其中两个输入为温度 $|E|$ 和温度变化率 $|EC|$ ，三个输

出分别是 $K_p$ （比例系数参数）； $K_i$ （积分系数参数）； $K_d$ （微分系数参数）。从而获得如上图 2.8 所示的传统模糊推理算法的模糊系统。

结合地铁环境空调系统的特点，选取模糊变量的模糊集隶属函数作为正态分布。假定两个输入变量分别为 $|E|$ 和 $|EC|$ ，三个输出变量分别为 $K_p$ 、 $K_i$ 和 $K_d$ ，并且假定这三个输出变量的模糊集为 $\{NB、NM、NS、ZO、PS、PM、PB\}$ 。子集中的七个元素分别表示较大负数，负数，较小负数，零，较小正数，正数，较大正数。输入和输出变量的值域为 $[-6, 6]$ 。从而得到的正态分布图如下图 2.9 所示：

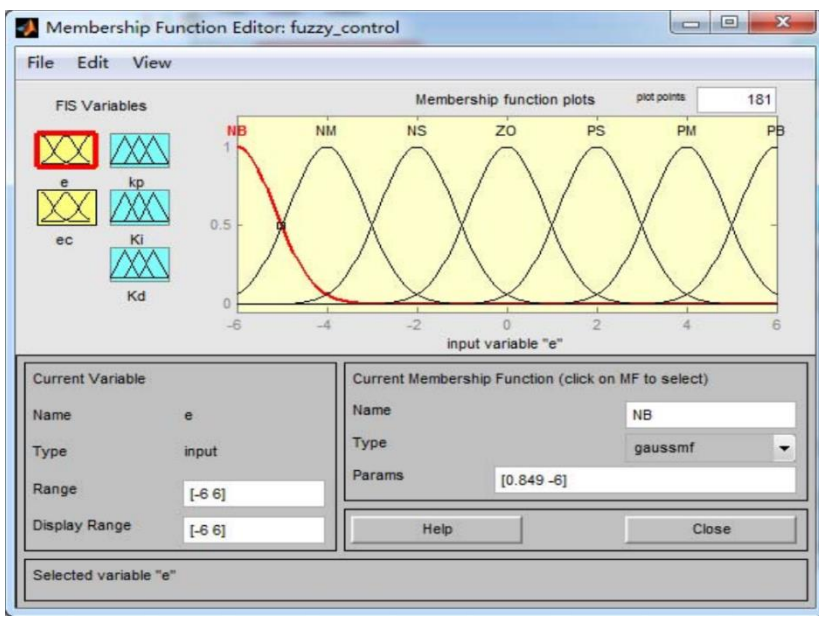


图 2.9 隶属函数正态分布图

Lina Li 还在文中列出 $K_p$ 、 $K_i$ 和 $K_d$ 模糊控制规则，其中 $K_p$ 模糊规则控制表如下表 2.1 所示， $K_i$ 和 $K_d$ 模糊控制规则表与 $K_p$ 类似，在此不一一讲述：

表 2.1  $K_p$ 的模糊控制规则表

$K_p \backslash e_c$	NB	NM	NS	ZO	PS	PM	PB
NB	NB	NB	NM	NM	NS	ZO	ZO
NM	NB	NB	NM	NS	NS	ZO	ZO
NS	NB	NM	NS	NS	ZO	PS	PS
ZO	NM	NM	NS	ZO	PS	PM	PM
PS	NS	NS	ZO	PS	PS	PM	PB
PM	ZO	ZO	PS	PS	PM	PB	PB
PB	ZO	ZO	PS	PM	PM	PB	PB

借助仿真软件构建了 PID 控制和模糊 PID 控制的仿真模型，并提供了对应的阶跃响应。本文构建的数学模型只考量了地铁车站空调系统的温度控制环节。经过测试，构建了以温度为变量的传递函数：

$$G(s) = \frac{e^{-s}}{s^2 + 3s + 1} \quad (2.1)$$

参照传递函数，采用仿真软件得到了 PID 控制仿真图和模糊 PID 控制仿真图，如下图 2.10 所示：

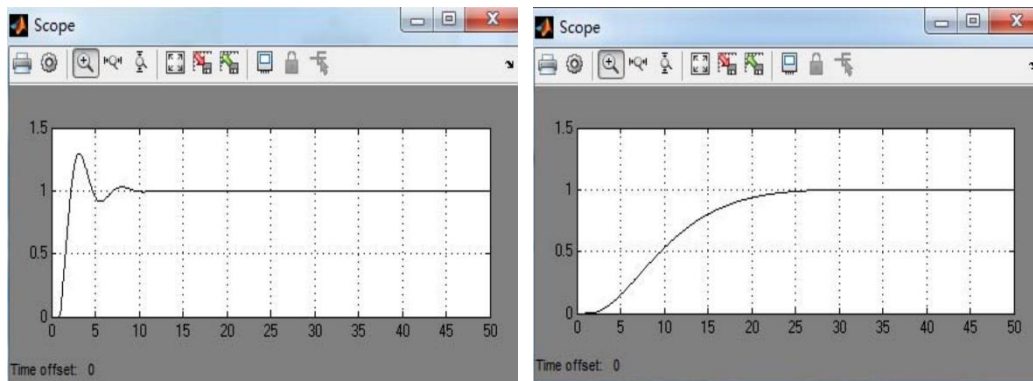


图 2.10 PID 控制仿真图（左）、模糊 PID 仿真图（右）

为了获取模糊 PID 控制，结合了模糊控制和 PID 控制，修正了传统 PID 控制和简单模糊控制的缺点。从仿真结果可以获得结论，模糊 PID 控制在空调通风系统中具有良好的快速性、准确性和鲁棒性<sup>[24]</sup>。

### BP 神经网络算法

1986 年，Rumelhart 等人提出 BP 神经网络(Back Propagation NN)，该神经网络是当前应用最为普遍和成功的一种人工神经网络，它是一种单向传播的多层前向网络<sup>[25]</sup>。BP 神经网络采用 BP 学习算法，BP 学习算法即反向传播学习算法<sup>[26][27]</sup>，BP 网络由输入层、输出层、隐含层构成，依据输出层的误差预估上一层的误差，然后依据上一层的误差预估它的上一层的误差，以此类推，从而预估各层的误差，最后依据各层的误差能够不断调整各神经元之间的权值，从而达到 BP 神经网络减小误差的目的。

BP 网络学习算法是被使用最频繁的一种学习算法，从结构上看，三层 BP 网络是一个经典的前馈型层次网络。它的理论基础是把一组数据样本作为神经网络的输入，然后把经过神经网络学习后得到的输出与期望值进行对比分析。假设隐含层只有一层，输入层、输出层和隐含层各层分别有  $m$ 、 $n$ 、 $j$  个神经元，人为把输入层节点  $a_i$  到隐含层节点  $b_r$  间的连接权值设定为  $w_{ir}$ ，把隐含层节点  $b_r$  到输出层节点  $c_j$  间的连接权值设定为  $v_{rj}$ 。

如图 2.8 所示为 BP 神经网络的三层结构：



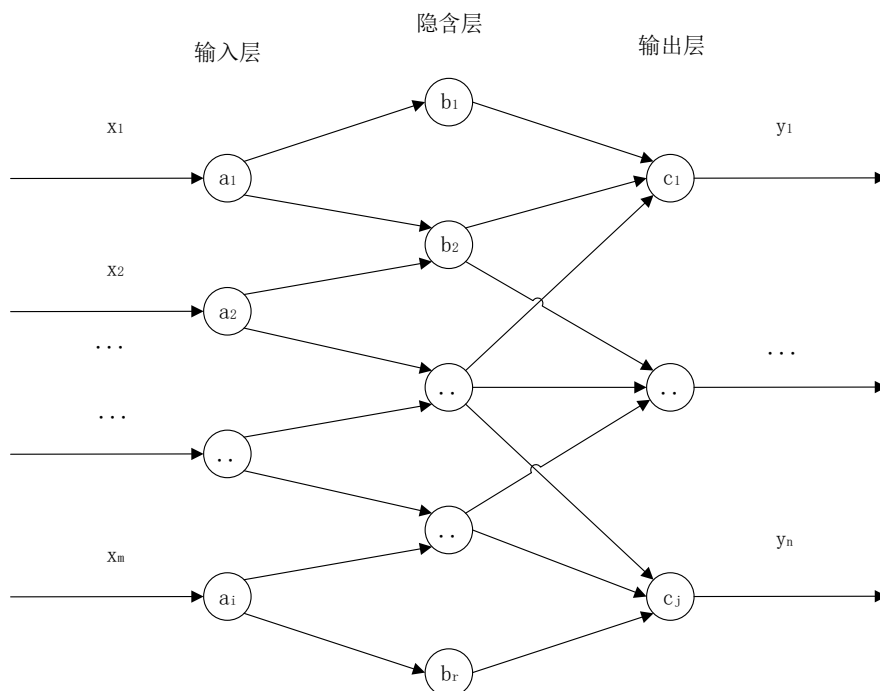


图 2.8 三层 BP 神经网络结构图

2008 年，黄丽在文献<sup>[28]</sup>中对 BP 算法作了较为深入的研究，提出了两种改进算法：因子分析遗传算法反向传播神经网络以及基于免疫算法的粒子群优化算法，并通过使用遗传算法改良 BP 算法，其基本思想是：先采用遗传算法对网络实行训练，找到一个较优解；然后将此网络参数作为算法中网络的初始参数再进行训练，达到增强网络的分类能力的目的。

2014 年，王志斌在文献<sup>[29]</sup>中融合了 BP 神经网络和遗传算法，采用 BP 算法训练阵列的方法，并且依靠 GA 优化 BP 网络的阈值和权值，对 LED 阵列的结温实现预估，解决了单独采用 BP 网络容易导致局部极小值和收敛速度慢的缺点。

GA 是一种全局优化随机搜索算法，通过遗传算子模拟生物遗传过程中的选择、交叉和变异过程，从而获得最优个体。鉴于 GA 的这种特点，将 GA 和 BP 神经网络相结合形成 GA 优化的 BP 神经网络，并将其应用到大功率 LED 的结温预测。GA 优化 BP 神经网络算法共分为两步：

- 1、采取 GA 完成对 BP 网络的阈值和权值的全局检索，定位到最优解的大概区间，使阈值和权值种群集中在参数空间的某几个地方；
- 2、利用 BP 算法局部寻优的特点可以得到最优解，这样能更为精准地预测函数的输出。

GA 优化 BP 神经网络算法流程如下图 2.9 所示。

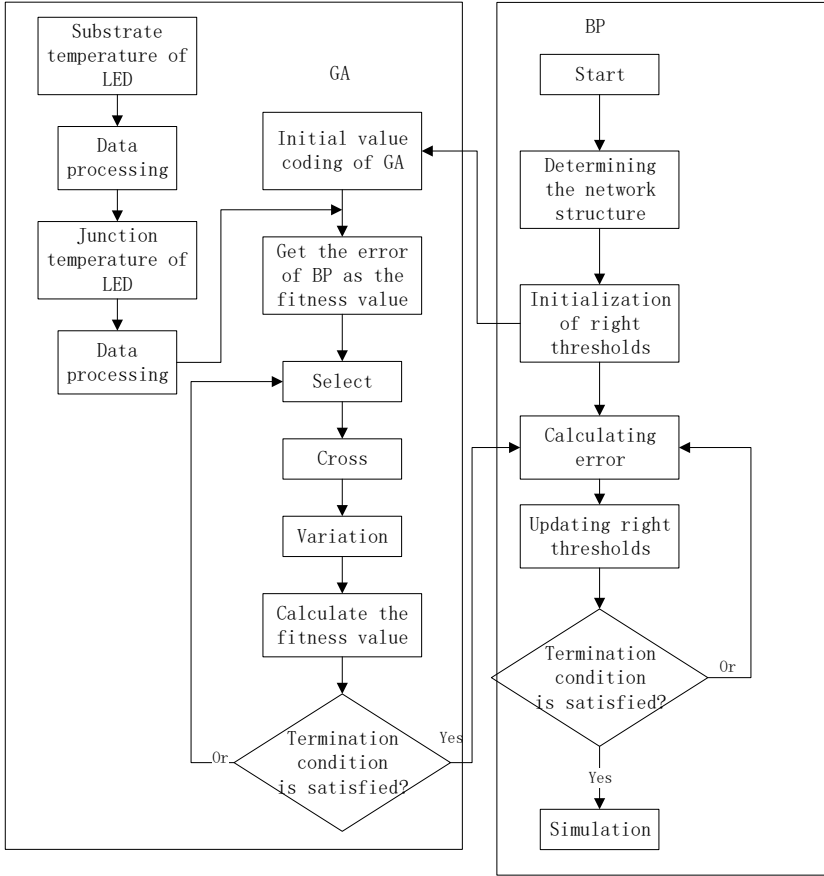


图 2.9 GA 优化 BP 网络算法流程图[29]

因为本文实现的是物联网环境下智能楼宇的控制系统，其中传感器收集的数据经由处理器分析需要收集大量生活经验，且没有具体的数学模型，所以使用模糊控制算法。如根据室内外光照强度，光敏电阻判断电灯的打开与否等。智能楼宇控制系统需要采集大量生活经验和利用模糊逻辑推理算法的需求，与模糊控制算法的拥有系统化的理论和丰富的实际应用背景的特色相吻合。所以经过比较，本文的智能楼宇控制系统主要采用模糊控制算法作为逻辑算法依据。

2.3.3 Web 框架技术

本文中使用的 Web 框架技术涵盖了 Java 编程语言、B/S 体系结构和数据库技术。

Java 编程语言

Java<sup>[30] [31] [32]</sup>是一种面向对象的编程语言，它让程序员可以利用简单的思维方式完成繁琐的编程。上位机控制系统为 Java Web 项目，开发主要在 Myeclipse10 中进行。上位机控制系统中前端页面设计和后端 Java 程序设计主要采用 Java 编程语言编写。

Java 语言有两种主要的编译环境<sup>[33] [34]</sup>，分别是 JDK（Java Development Kit）和 JRE（Java Runtime Envirnment）。JDK 即 Java 开发包或 Java 开发工具，它可以成为编写 Java 的 Applet

小程序及应用程序的开发环境。JDK 是 Java 编程语言的核心，不仅仅囊括了 Java 运行环境，还具有一些 Java 工具和 Java 的核心类库（Java API）。

JRE 是运行环境，JDK 是开发环境，所以编写 Java 程序的时候需要使用 JDK，而运行 Java 程序的时候就需要用到 JRE<sup>[36]</sup>。JDK 包括很多与运行没有关系的部分内容，因此 JDK 占用的内存容量较大，所以只要不是运行特殊的 Java 程序，只需要安装 JRE，不需要安装 JDK，。

### C/S 和 B/S 体系结构

C/S 结构软件（即客户机/服务器模式）分为客户机和服务器两层，客户机为了达到降低网络的通信数据量和服务器的运算量的目的，在客户机和服务器的两侧分布应用软件的数据和计算。而在现实场景中，服务器价格昂贵，因此服务器的连接个数有限。所以 C/S 结构软件的应用场景必须是用户数较少的局域网。我国的大多数 ERP（财务）软件产品就是属于该类结构<sup>[35]</sup>。

C/S 结构的客户端反应速度快，因为 C/S 结构能够在上交给服务器之前，先在客户端处理任务，不过 C/S 结构有以下几个缺点：

- 1、C/S 结构仅在局域网中适用。但是局域网已然无法满足如今移动办公和分布式办公的需要，加入要满足远程访问的功能必须具有专业的技术，要对系统采取特别的设计，以满足处理分布式的数据的要求；

- 2、客户端必须安装固定的客户端软件。成本问题变为安装特定的客户端软件的最大阻隔，首先安装的工作量巨大，其次安装之后的计算机维护对人力物力要求很高，系统软件升级时用户需要重新安装，非常繁琐。

- 3、支持客户端的操作系统局限性。考虑到客户端软件的兼容性，用户使用的操作系统不一致，会导致安装维护成本更高。

B/S 结构便是浏览器和服务器结构，C/S 结构在使用软件时需要先安装对应的客户端，与 C/S 结构不同，B/S 结构的软件不需要安装任何软件，只要拥有浏览器就能够使用。采用 B/S 结构设计的软件可以简化客户端，并且使系统更易于维护、扩展和更新<sup>[37]</sup>。在 B/S 结构下，客户端只需要浏览器就能够实现业务操作。B/S 结构是现今应用软件体系结构的首要选择。本文也选用 B/S 架构设计完成控制平台的实现。

### 数据库技术

数据库(Database) 产生于距今六十多年前，数据库可以参照数据结构，实现组织存储和管理数据功能。直到上个世纪末，数据管理逐渐成为用户需要的数据管理方式，而不再局限于存储和管理数据。数据库的类型多种多样，从最简单的存储有各种数据的表格到能够进行海量数据存储的大型数据库系统都在各个方面得到了广泛的应用<sup>[38]</sup>。本文的上位机控制系统

的数据库部分利用 MySQL Workbench 完成设计,MySQL Workbench 是为 MySQL 设计的数据库建模工具,可以构建数据库文档、设计新的数据库图示,还可以实现繁琐的 MySQL 转移。

## 2.4 本章小结

本章主要介绍了本文提出的基于 wifi 的物联网智能控制系统的整体架构和涉及到的相关技术。首先给出了基于 wifi 的物联网智能控制系统的整体架构图,将系统分为三个部分:传感节点部分、嵌入式平台部分和上位机界面部分,紧接着对每个部分的组成和主要工作流程做了介绍。然后介绍了系统涉及到的相关关键技术,包括无线数据传输技术,其中有 wifi 通信技术和通信设计。因为本文研究的智能楼宇控制系统中的照明系统没有固定的数学模型,需要根据人类经验形成的模糊规则库分析,通过对比了几种热门控制算法后,选择使用模糊控制算法。针对本文研究的基于 wifi 的物联网智能控制系统的软件设计,最后对 Web 框架技术做了简单的介绍,其中包括:Java 编程语言、C/S 和 B/S 体系结构和数据库技术。本章为本文的智能控制系统设计夯实了理论基础,本文在规划智能控制系统时,硬件部分采取了 Arduino UNO R3 嵌入式开发板、ESP8266 无线传输模块和传感器组成的智能终端模块,上位机控制平台采取了 B/S 架构以及 MVC 设计模式,囊括了前端页面、后端服务程序和数据库三个部分。

## 第三章 基于 wifi 的物联网智能控制系统的硬件设计

基于 wifi 的物联网智能控制系统的硬件设计采用第二章所介绍的智能控制系统总体设计中传感节点层和嵌入式平台层的设计思路。硬件部分主要采用 Arduino UNO R3 开发板和 ESP8266 无线传输模块,本章以智能楼宇中的照明系统为例,介绍了 Arduino 开发板与传感器组成的智能终端模块之间的接口设计、Arduino 开发板与 ESP8266 模块之间数据传输接口设计以及 ESP8266 内部 AP/STA 工作模式的介绍。

### 3.1 开发平台的选择

本文主要研究智能楼宇中的控制系统的实现,其中嵌入式平台是最为关键的部分,系统中所有业务功能基本都在该部分完成。因此选择适当的嵌入式开发平台以及操作系统显得格外重要。

Arduino 作为一种开源的单片机电子设计平台,其低廉的价格、开放性的平台、可扩展的接口等优势,成为本文在设计智能楼宇控制系统中的首要选择目标。Arduino 的硬件部分包括 Atmel AVR ( Audio-video Receivers) 单片机、I/O 接口及其它相关电路,软件部分则由标准的程序编译器和程序下载器构成,这个集成开发环境 (IDE) 囊括了代码编写、编译、调试、分析的功能。与传统的单片机相比较,Arduino 的特点除了包含免费、开源的 IDE,开源的硬件 (包含原理图和 PCB 图),所有资源都可以免费下载,并且可按照自己需求修改,便于与传感器及各种各样的电子组件相连<sup>[39]</sup>。

本文设计的智能楼宇中的无线传输主要采用 wifi 传输,其中 wifi 采用工业级芯片 ESP8266,模块为 ESP-13,带金属屏蔽罩,具有强抗干扰能力;ESP8266 模块的标准引脚可以与 Arduino UNO、Mega2560 等开发板兼容,通过电压转换芯片将 Arduino 与 UNO (通用网络对象) 连接,从而达到 3.3V 和 5V 兼容;由于选用的是串口 wifi,所以 Arduino 程序可以按照出厂设置,无需修改;可以在浏览器上配置 wifi 和串口参数;可以作为单独的 ESP8266 开发板使用,通过下载官方 AT 指令固件、NodeMCU 开源固件等。ESP8266 无线模块还具有支持无线 802.11 b/g/n 标准、内部配置有 TCP/IP 协议栈、可以配置一个 socket 连接、满足 STA/AP 两种工作模式等优势。

## 3.2 基于 Arduino 的智能终端的硬件设计

### 3.2.1 硬件介绍

Arduino 是一个体积较小、容易操作的开源平台，该操作平台由两部分构成，分别是：硬件（各种型号的 Arduino 板）和软件（Arduino IDE）。它通过开放原始代码 simple I/O 平台构建<sup>[40][41]</sup>，Arduino 电路板的硬件部分，可以用来做电路连接。另一部分计算机中的程序开发环境：Arduino IDE，首先在 IDE 中输入程序代码，然后把程序烧写入 Arduino 电路板，Arduino 电路板就可以根据写好的程序命令完成预设的功能。由各种传感器收集环境数据然后回馈到 Arduino 平台上，与此同时 Arduino 可以依照事先写好的程序对这些传感器进行操控。

Arduino 平台有以下四个特点：

1、跨平台。Arduino IDE 可以在 Windows、Macintosh OS X、Linux 三大主流操作系统上运行，而其余大多数的控制器局限于在 Windows 上操作。

2、简单清晰。Arduino IDE 对初学者而言，很容易上手，而且这种编程语言非常灵活。Arduino 语言是对 avr-gcc<sup>[42]</sup>库的二次封装，该编程语言的使用对软件编程能力和单片机硬件基础要求不严格。

3、开放性。Arduino 的一切资源都是开放的，不仅仅是 Arduino 的硬件电路图、原理图，就连编程软件和核心库的源码都是开放的，开发者可以按照自己的想法随意改变最初设计及相应代码。

4、发展迅速。因为开发者更关注 Arduino 便捷的开发方式，该开发方式能更快的完成自己的项目开发，极大的缩减了学习的成本，而且减短了开发的周期，因此发展迅速。

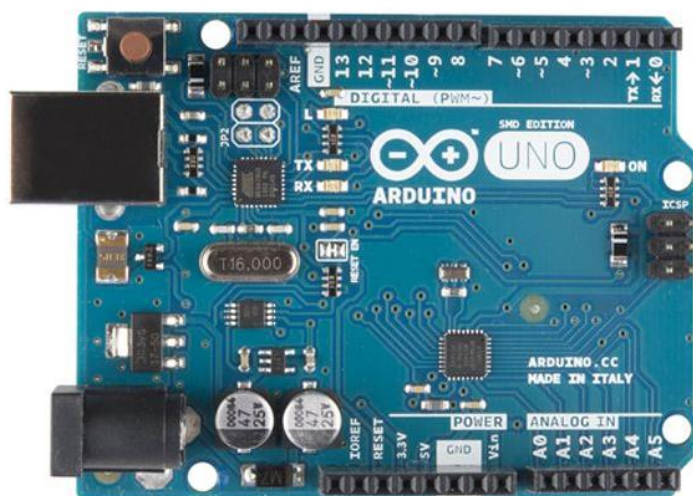


图 3.1 Arduino UNO R3 开发板实物图

由于 Arduino 的各层面的便利优点,更多的硬件开发者在开发他们的项目或产品时逐渐采用 Arduino;尤其在物联网等研究方向,更多的软件开发者采用 Arduino 进行创新研究;在大学生的课堂里,诸如自动化、软件专业,就连艺术专业,都开设了 Arduino 的相关课程<sup>[43]</sup>。本文用的是 Arduino UNO R3 型号开发板,如上图 3.1 所示为开发板实物图。

如图 3.1 所示,Arduino UNO R3 开发板上引脚编号为 0-13 的端口为数字输入、输出端口,一共有 14 个;引脚编号为 0-5 的端口为模拟输入、输出端口,一共有 6 个;支持物联网业务提供商(ISP)下载功能;可以采用 USB 连接供电,此时不需要另外供电;采用 Atmel Atmega328 微处理控制器。

### 3.2.2 Arduino 与智能终端模块之间接口设计

本节主要针对 LED 亮灭的控制与检测实现整个 Arduino 与智能终端模块之间的设计,下面对 LED 亮灭控制与检测电路做较为详细的描述。

#### 电路设计原理

LED 亮灭控制与检测电路的核心设计思想是利用 Arduino 的数字输出引脚达到控制 LED 的亮灭的目的,利用光敏电阻的阻值随光照强度的增大而减小的特性,如实地反映 LED 的亮灭情况。

对 LED 实现开关控制(亮灭控制)只要改变 LED 两端的电压即可,通过程序设定,使用 Arduino UNO R3 的数字输出引脚可以十分方便地改变 LED 两端的电压。光敏电阻有着在光照强度增加的情况下,电阻值迅速减小的特性。因此,如果将光敏电阻放置在 LED 灯旁边,对光敏电阻两端的电压进行检测,应当呈现出 LED 亮时光敏电阻两端电压较小,LED 灭时光敏电阻两端电压较大的现象。利用此原理,应当可以通过光敏电阻两端的电压反推 LED 的亮灭情况。

#### 电路图分析

LED 亮灭控制与检测电路的电路图如下图 3.2 所示:

从图中可以清晰地看出,此电路总体上分为两个部分。右侧是五个 LED 灯,左侧是四组光敏电阻和普通电阻的组合。

首先分析看上去相对较为简单的右侧的电路。右侧五个 LED 灯的接法是相同的,每个 LED 都是正极接到 Arduino UNO R3 的数字引脚上,负极接地。通过编程可以设定 Arduino UNO R3 的数字引脚为输出,通过改变输出电平控制 LED 的亮灭。

然后分析左侧的电路,左侧的电路有四组光敏电阻和普通电阻的组合,每组都采用相同

的接法。每一组都是将光敏电阻的一端接电源正极，然后将光敏电阻和普通电阻串联，再将普通电阻的另一端接地，接着将光敏电阻与普通电阻的连接点接到 Arduino UNO R3 的模拟输入引脚，四组分别接到四个不同的模拟输入引脚。如此一来，Arduino UNO R3 的四个模拟输入引脚测量的就是对应的四个普通电阻两侧的电压。

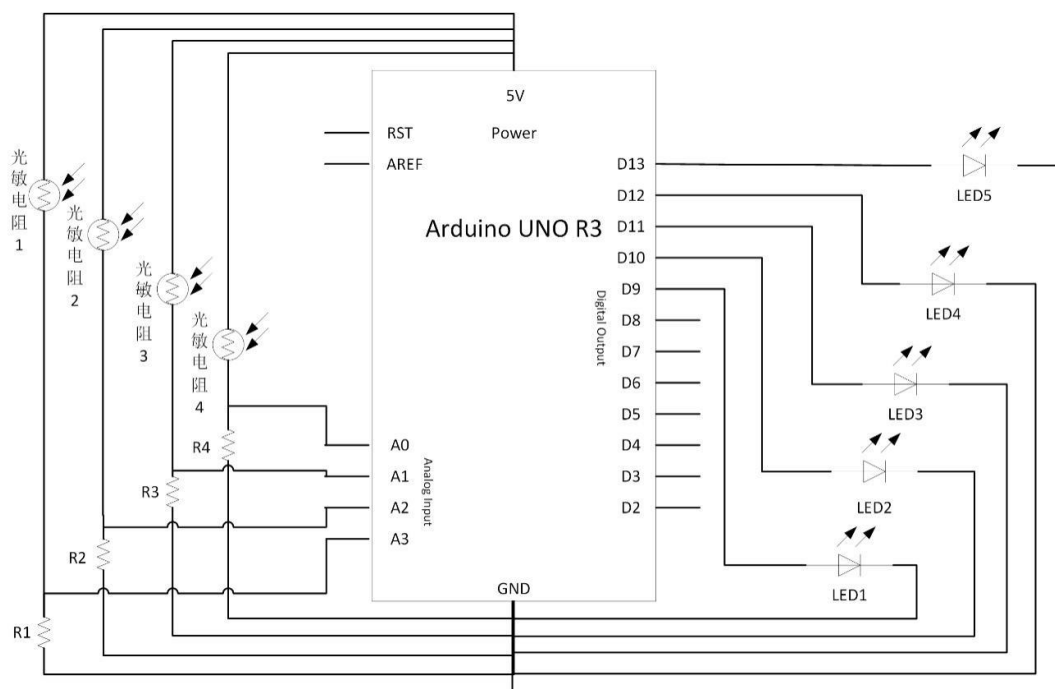


图 3.2 LED 亮灭控制与检测电路图

在这里需要申明，因为 LED 与光敏电阻没有任何电路上的连接关系，所以在电路图中画的相距较远。在实物图中，光敏电阻是紧挨着 LED 放置的，这是为了让光敏电阻周围的光照强度因为 LED 的亮灭发生较为明显的变化，如果光敏电阻和 LED 放置相隔较远，可能会出现光敏电阻阻值变化不明显的情况。

另外，右边的电路中，LED1 旁边并未放置光敏电阻，其他每个 LED 灯旁边均放置了光敏电阻。设置 LED1 达到与其他的 LED 形成对比的目的，从而检测此电路与单纯的 LED 亮灭控制电路的区别。

此电路检测 LED 亮灭情况的工作原理为：当 LED 由灭变亮时，因为光敏电阻与 LED 放置距离很近，所以光敏电阻附近的光照强度迅速增加，光敏电阻的阻值加速减小。因为光敏电阻与普通电阻采取串联分压的连接方式，光敏电阻阻值快速减小将会使得光敏电阻分得的电压快速减小，由于电源电压是恒定的，普通电阻上分得的电压将会迅速增大。因此，根据对普通电阻两侧的电压的检测，设置合适的阈值，同样能够反推 LED 的真实亮灭情况。如，检测普通电阻两侧的电压，判断出大于所设的阈值，就可以推断 LED 灯为开启状态。



3.2.3 Arduino 与无线传输模块之间接口设计

Arduino UNO R3 开发板与 ESP8266 无线传输模块是上下重叠插在一起的，如下图 3.3 所示：

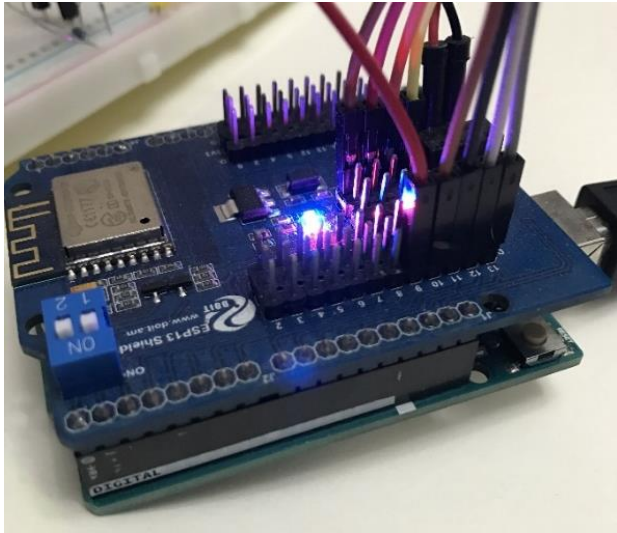


图 3.3 Arduino UNO R3 开发板与 ESP8266 无线传输模块实物图

图 3.3 中上方蓝色的是 ESP8266 无线传输模块，下方绿色的是 Arduino UNO R3 开发板。下方 Arduino 开发板引出 Arduino Pin 2/3/4/5/6/7/8/9/10/11/12/13，上方 ESP8266 无线传输模块引出 ESP8266 GPIO 0/2/4/5/9/10/12/13/14/15/16/ADC/EN/UART TX/UART RX。Arduino 开发板与 ESP8266 无线传输模块连接接口对应关系如下表 3.1 所示。

表中 ESP8266 的 SW 引脚可以设置为“on”和“off”两种模式，当需要向 Arduino 开发板烧写程序时，需要将开关拨到“off”状态，当需要将 ESP8266 开发板设为 AP 热点时，则需要将开关拨到“on”状态。ESP8266 的 Station 和 AP 两种工作模式的详细配置见本文 3.3.2 章节。

表 3.1 ESP8266 与 Arduino 连接接口定义表

部分	引脚号	功能描述	备注
Arduino	G	地	
	D	数字 IO	
	A	模拟 IO	
	5V	5V	
	TX	Arduino Uno TX	可通过 SW1 连接 ESP8266
	RX	Arduino Uno RX	可通过 SW1 连接 ESP8266
	SCL	Arduino SCL	

	SDA	Arduino SDA	
ESP8266	G	地	
	D	数字 IO	
	3V3	3.3V	
	RX0	ESP8266 RX0	可通过 SW1 连接 Arduino
	TX0	ESP8266 TX0	可通过 SW1 连接 Arduino
	EN	ESP8266 EN	
	AD	ESP8266 AD	
	RST	ESP8266 复位按钮	
	KEY	ESP8266 配置按钮	
	SW	ESP8266 And Arduino	下载 Arduino 程序时，需要将 SW1 的第 2 路置于“off”状态

3.3 基于 wifi 的智能终端的硬件设计

3.3.1 硬件介绍

本文选取的 ESP8266 硬件支持无线 802.11 b/g/n 多种标准，内置 TCP/IP 协议栈，有 STA/AP 两种工作模式可供选择，还支持配置一个 socket；该硬件满足标准 TCP/UDP Server 和 Client 要求，拥有标准 Arduino UNO、Mega 引脚间距，可以直接嵌入 Arduino UNO R3 开发板上作为无线传输模块使用；该硬件通过两位拨码开关达到切换 Arduino 和 ESP8266 串口扩展的目的。该 wifi 模块的目的是让整个控制系统与网络连接，达到实现无线信号与串口信号之间切换的目的。本文采用的 ESP8266 的无线传输模块硬件实物图如下图 3.4 所示：

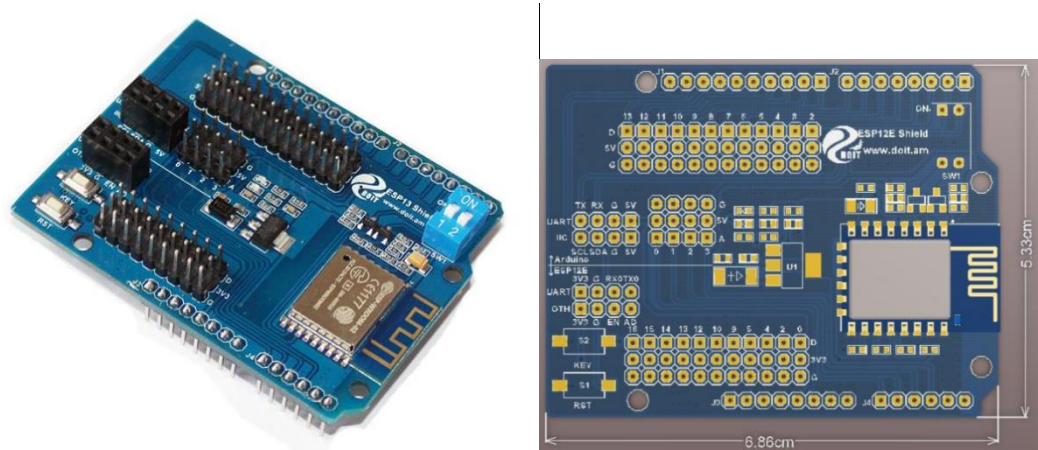
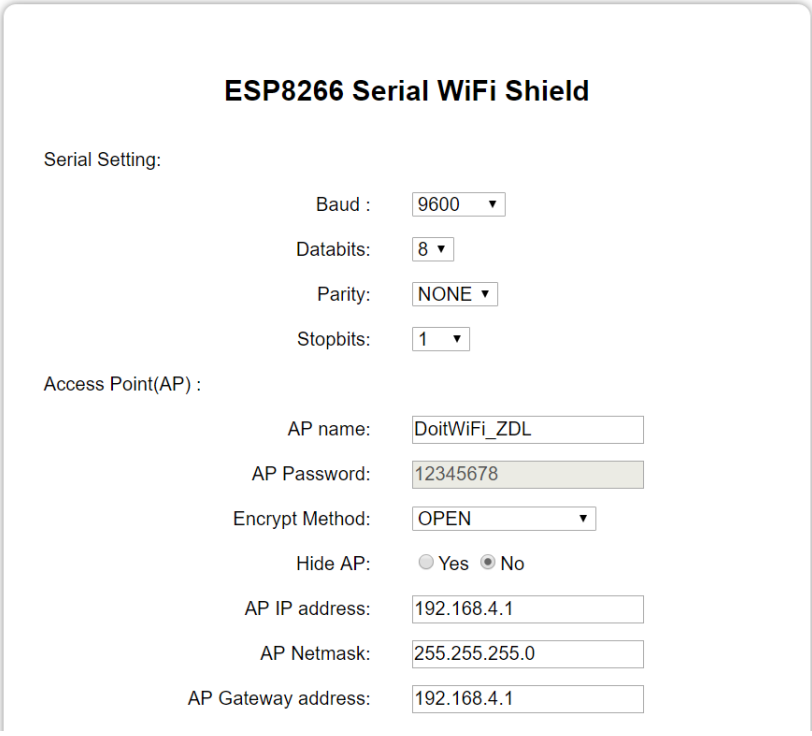


图 3.4 ESP8266 硬件实物图

### 3.3.2 ESP8266 无线传输模块设计

ESP8266 无线传输模块的硬件设计主要就是无线局域网的构建。本系统依靠 ESP8266 无线传输模块的强大功能实现无线局域网的构建。首先确保 ESP8266 无线传输模块的工作状态控制开关处于“on”状态，此时 ESP8266 无线传输模块与 Arduino UNO R3 之间的串口是打开的，ESP8266 无线传输模块的 wifi 模块收到的数据可以直接通过串口透传给 Arduino UNO R3。

ESP8266 无线传输模块有 AP 和 Station 两种工作模式，分别作为无线网络的接入点和无线终端，本系统通过设置 ESP8266 无线传输模块，令其处于 AP 模式，配置页面截图如下图 3.5 所示：



The screenshot shows the configuration interface for an ESP8266 Serial WiFi Shield. It is divided into two main sections: 'Serial Setting' and 'Access Point(AP)'. The 'Serial Setting' section includes dropdown menus for Baud rate (9600), Databits (8), Parity (NONE), and Stopbits (1). The 'Access Point(AP)' section includes text input fields for AP name (DoitWiFi\_ZDL), AP Password (12345678), AP IP address (192.168.4.1), AP Netmask (255.255.255.0), and AP Gateway address (192.168.4.1). It also features a dropdown for Encrypt Method (OPEN) and radio buttons for Hide AP (Yes/No, with No selected).

ESP8266 Serial WiFi Shield	
<b>Serial Setting:</b>	
Baud :	9600 ▼
Databits:	8 ▼
Parity:	NONE ▼
Stopbits:	1 ▼
<b>Access Point(AP) :</b>	
AP name:	DoitWiFi_ZDL
AP Password:	12345678
Encrypt Method:	OPEN ▼
Hide AP:	<input type="radio"/> Yes <input checked="" type="radio"/> No
AP IP address:	192.168.4.1
AP Netmask:	255.255.255.0
AP Gateway address:	192.168.4.1

图 3.5 ESP8266 无线传输模块配置页面截图 1

通过配置页面设置了 ESP8266 无线传输模块处于 AP 模式，并设置了 AP(接入点)名称、密码以及接入点 IP 地址、接入点子网掩码、接入点网关等参数后，点击“submit”按钮，ESP8266 无线传输模块在 5 秒后重新启动，此时通过 PC 即可连接到名为“DoitWifi\_ZDL”密码为“12345678”的热点，从而构建了 Arduino 嵌入式设备与上位机控制平台进行信息交互的无线网络。

### 3.4 本章小结

本章从基于 wifi 的物联网智能控制系统的硬件设计中主要的硬件开发平台的选择依据入手。然后分两部分介绍该系统的硬件设计，分别是：基于 Arduino 的智能终端的硬件设计和基于 wifi 的智能终端的硬件设计。基于 Arduino 的硬件设计分析了 Arduino 与传感器节点（光敏传感器）的电路设计、电路图分析等，还研究了 Arduino 与无线传输模块 ESP8266 的接口设计；通过基于 wifi 的硬件设计分析了 ESP8266 的硬件原理，并以此研究了 ESP8266 硬件的无线传输的设计。

## 第四章 基于 wifi 的物联网智能控制系统的软件设计

基于 wifi 的物联网智能控制系统的软件设计主要包括三部分，分别是：Arduino 嵌入式平台的软件设计，主要包括传感器门限值设定、控制算法设计等；上位机控制平台软件设计，主要包括开发模式、开发软件和框架的选择，ESP8266 无线传输模块的 socket 无线通信的设计与实现，上位机软件和界面设计等。

### 4.1 Arduino 嵌入式平台的软件设计

本设计主要针对智能楼宇中的照明系统。

本文研究的 Arduino 嵌入式平台的软件设计主要为 LED 亮灭控制算法。LED 亮灭控制算法是在 Arduino IDE 中采用 Arduino 编程语言，对 Arduino UNO R3 进行编程实现的。Arduino IDE 是一个用于对 Arduino 开发板进行编程的集成开发环境。

LED 亮灭控制算法的设计思路为：通过识别指令字符设定相应的数字输出接口输出高电平或低电平，实现 LED 的亮灭控制；通过设定门限值的方法判断普通电阻两端电压的变化情况，以此反推 LED 的真实亮灭情况。从亮灭控制的门限值设定方法着手，讲述系统的设计思路。

#### 门限值的设定

因为普通电阻和光敏电阻串联分压，普通电阻两侧的电压会根据 LED 的亮灭产生变化。这里的门限值就是针对普通电阻两端的电压而设定的，是一个能够区分 LED 的打开状态和关闭状态的数值。

要使得门限值的设定能够区分 LED 的打开状态和关闭状态，需要进行实验测定。首先，使 LED 处于关闭状态，通过模拟输入接口读到普通电阻两端的电压等效数值为  $a_1$ ，接着使 LED 处于打开状态，通过模拟输入接口读到普通电阻两端的电压等效数值为  $a_2$ 。显然， $a_1 < a_2$ ，而门限值应当设定为  $a_1$  到  $a_2$  之间的一个值。比如，可设门限值为  $a$  且有  $a = (a_1 + a_2) / 2$ 。在不同的光照环境下， $a_1$  和  $a_2$  的取值都不相同，计算得到的  $a$  的值也不相同。因此，当 Arduino 嵌入式设备所处的照明条件发生变化时，需要重新设定门限值。

#### 判断逻辑

设模拟输入接口读到的普通电阻两端电压的等效数值为  $N$ ，若  $N$  大于等于门限值时，则说明现在 LED 处于打开状态，若  $N$  小于门限值，则说明现在 LED 处于关闭状态。

### 算法的代码实现逻辑

1、初始化变量与数字接口：定义表示串口读入数据的变量 val, 定义表示普通电阻两端电压等效整数的门限值的变量 a 及其取值，定义数字接口 9-13 分别为 ledpin1-ledpin5。在 setup 函数中，设置 Arduino UNO R3 的串口通信波特率为 9600，并且将数字接口 9-13 初始化为输出口。

2、声明检测变量、读入串口数据：定义 n2-n5 分别表示 A3 到 A0 的模拟输入，即对应普通电阻 R1-R4 两端的电压值。在 loop 函数中，从 Serial 的 read 函数中读取串口的输入数据，若是读取到字符串，那么返回相应的十进制 ASCII 码，例如串口接收到字符“1”，则该函数返回整型值 49。

3、根据串口读入执行开关操作：用 if 语句对串口读入的指令数据进行分类处理。首先进行判断，如果 read 函数读到的整型值等于字符“1”对应的十进制 ASCII 码，则进入指令字符“1”对应的处理逻辑代码块，调用 digitalWrite 函数，设置 ledpin1 对应的数字接口 9 输出高电平。假如 read 函数读到的整型值与其他指令字符对应的十进制 ASCII 码相等，那么进入相应字符对应的处理逻辑代码块，采取调用 digitalWrite 函数的方法，控制对应的数字接口输出高电平。指令字符“1”、“2”、“3”、“4”、“5”分别对应 LED1-LED5 执行开关和亮灭检测的代码块。

4、根据普通电阻两端的电压等效数值反馈 LED 亮灭情况：

在电路图分析中已经阐述过，LED1 的作用是和其他的 LED 形成对比，LED1 旁不设置包含光敏电阻的检测电路。在代码中体现为对于 LED1 的控制，即当串口读入的指令字符为“1”时，设置 ledpin1 对应的输出接口为高电平，稍作延迟后直接通过串口返回字符串“light1 on”，没有 LED 亮灭检测的处理逻辑。

当串口读入的指令字符，即 read 函数返回的整型值对应“2”、“3”、“4”、“5”时，除设置相应的数字接口输出高电平外，还要进行 LED 亮灭情况的检测。如指令字符为“2”时，设置 ledpin2 对应的输出接口为高电平，稍作延迟后通过 analogRead 函数读入引脚 A3 的模拟输入数值，该数值是将 0-5V 的电压映射到 0-1023 的整数得到的，比如读到的数值为 511 则对应的电压约为 2.5V。接着，将读到的模拟输入数值赋给变量 n2，判断 n2 与设定的门限值 a 的大小关系，若  $n2 \geq a$ ，则说明此时普通电阻两端的电压大于等于门限值 a 对应的电压值，LED 处于打开状态，通过串口输出“light2 on”；若  $n2 < a$ ，则说明此时普通电阻两端的电压小于门限值 a 对应的电压值，LED 处于关闭状态，通过串口输出“light2 off”。

另外，当串口读入的指令字符为“6”时，通过 digitalWrite 函数设置 ledpin1-ledpin5 对应的输出接口全部为高电平，打开所有 LED。当串口读入的指令字符为“0”时，通过 digitalWrite

函数设置 ledpin1-ledpin5 对应的输出接口全部为低电平，关闭所有的 LED。

需要说明的是，这里的串口读入数据来源于 ESP8266 无线传输模块。ESP8266 无线传输模块的 wifi 模块接收到上位机控制平台发送的指令数据，然后通过串口透传给 Arduino UNO R3。而对于串口输出字符串也是一样，先通过串口传输给 ESP8266 无线传输模块，然后通过 ESP8266 无线传输模块的 wifi 模块发送给上位机控制平台。

算法流程图

根据上述的代码实现逻辑，可得到 LED 亮灭模糊控制算法的算法流程图如下图 4.1 所示：

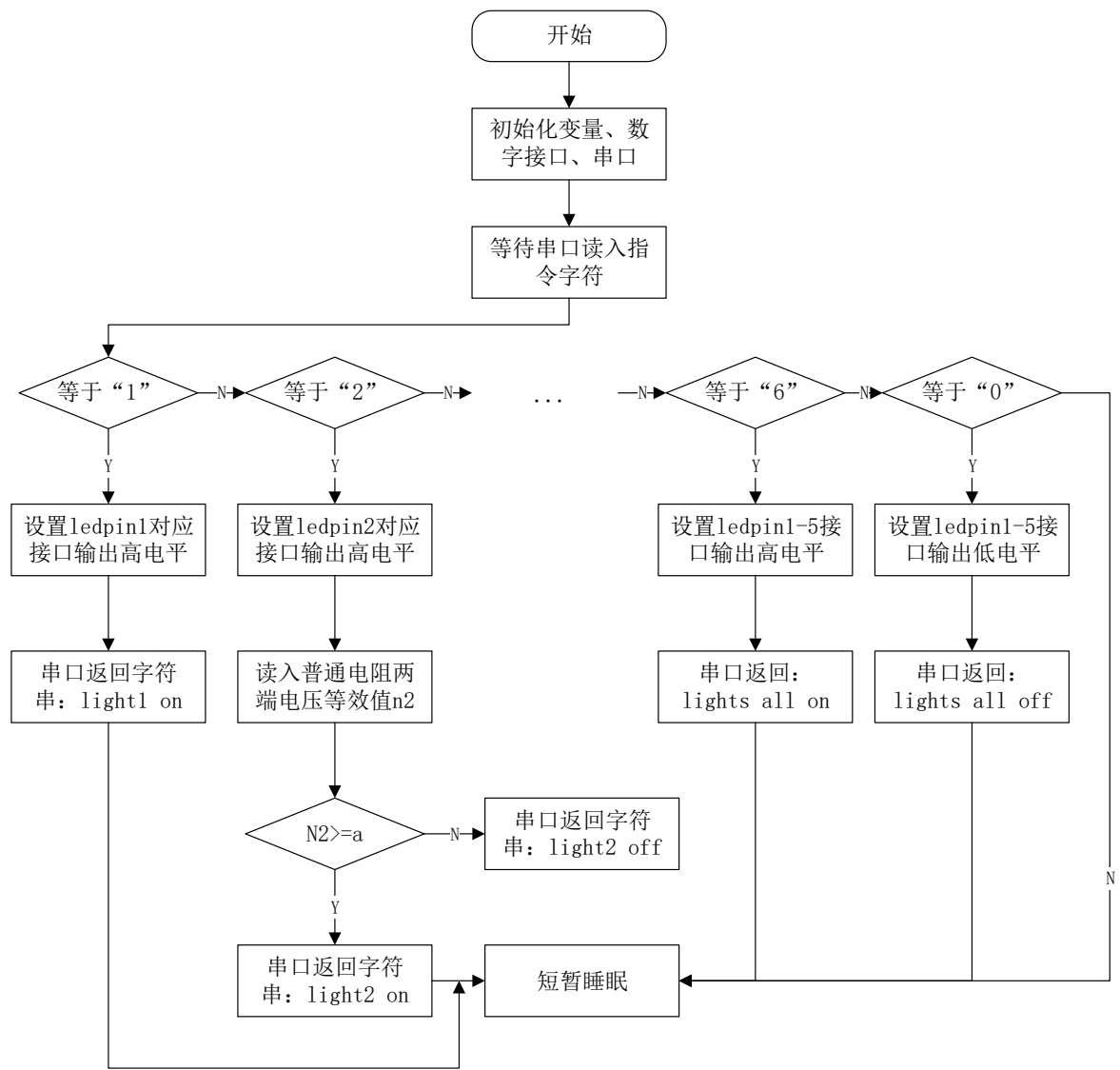


图 4.1 LED 亮灭模糊控制算法流程图

由于指令字符等于“3”、“4”或“5”时的后续处理逻辑与指令字符等于“2”时的相同，所以在图中省略了指令字符等于“3”、“4”或“5”时的情况。从图中我们可以清晰地看到 LED 亮灭模糊控制算法的整个执行流程。

## 4.2 模糊控制算法

模糊控制(Fuzzy Control)是一种由计算机数字控制的技术,它从属于智能控制,且具有非线性特征。模糊控制有一个显著的特点,就是系统化的理论和实际应用背景并存。因为PID控制难以实时改动PID的参数,而且很难控制非线性、时变、复杂的控制对象<sup>[44]</sup>。与之相反,模糊控制能够化简系统设计的复杂性,而且可以由非线性、时变、滞后、模型不完全的系统控制。

模糊控制器还有以下几个特点:

- 1、不依赖于被控对象的精确数学模型;
- 2、利用控制法则来描述系统变量间的关系;
- 3、不采用数值的形式,而用语言的方式的模糊变量来描述系统,模糊控制器不需要被控制对象构建完善的数学模型;
- 4、模糊控制器是一种语言控制器,它使得操作人员可以采用自然语言与机器对话;
- 5、模糊控制器是一种容易控制、掌握的较理想的非线性控制器,具有较佳的鲁棒性(Robustness)、适应性及较佳的容错性(Fault Tolerance)<sup>[45]</sup>。

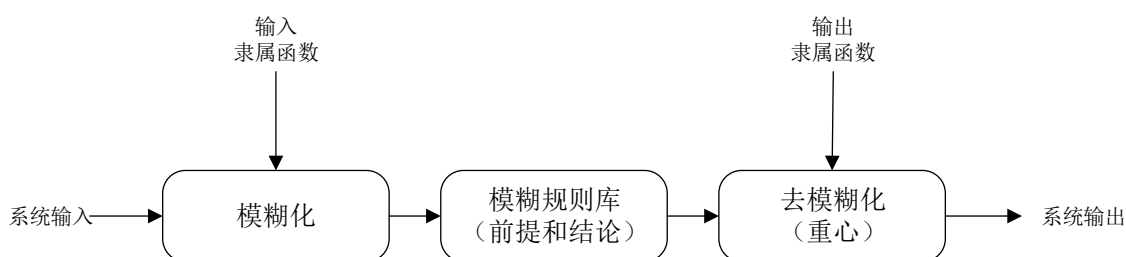


图 4.5 模糊逻辑过程

模糊逻辑过程如上图 4.5 所示,模糊控制算法一般包括三部分:

- 1、模糊化。通过输入隶属函数,模糊系统输入。作用是确定模糊控制算法的输入隶属函数,并将系统输入转化成系统能够识别的模糊量,分为以下三步:
  - 第一,对输入量进行满足模糊控制需求的处理;
  - 第二,对输入量完成尺度变换,采取设定取值范围的方法达到输入量的尺度变换的目的;
  - 第三,确定各输入量的模糊语言取值和相应的隶属度函数。
- 2、模糊规则库。参照人类专家的经验建立模糊规则库。模糊规则库中囊括很多控制规则,这一步是从实际中的控制经验转化到抽象的模糊控制器的关键。
- 3、去模糊化。将依据模糊规则库转化获取的模糊量去模糊化并输出。目的是将推理得到的控制量转化成控制输出。



与模糊逻辑过程相对应，本文实现的智能楼宇的光照系统的模糊控制基本步骤如下：

1、模糊化。设置环境亮度值，将光敏电阻采集视为输入；然后对另一个输入量控制指令进行尺度变换，根据经验以及程序中设定将控制指令数字化，从而完成输入量的尺度变换；最后根据不同的环境亮度值和不同的控制指令设定模糊语言的取值（即不同 LED 灯的“on”和“off”状态）。

2、模糊规则库。根据光敏电阻对环境亮度值的采集和不同 LED 灯对不同控制指令的反馈信息，总结出不同亮度环境下不同的控制指令对不同 LED 等控制的模糊语言。

3、去模糊化。将反馈信息依据模糊规则库对照出相应的模糊语言，从而转化为控制输出。

本文设计的智能楼宇的照明系统中，通过光敏电阻测得 LED 灯亮时的光照强度值约为 860lx(勒克斯，照度的国际单位)，因此定义亮度阈值为 860lx。通过逐渐改变环境亮度值，针对所有的 LED 灯逐一输入不同的控制指令，实现对所有的 LED 灯模糊控制。这里包含两个输入和一个输出。输入是控制指令和环境亮度值，输出是不同的 LED 灯的信息反馈值。

表 4.1 环境亮度为 0-100lux 时模糊控制输出规则表

$\begin{matrix} E_{0-100} \\ P \end{matrix} \backslash K$	LED1	LED2	LED3	LED4	LED5
0	off	off	off	off	off
1	on	off	off	off	off
2	off	on	off	off	off
3	off	off	on	off	off
4	off	off	off	on	off
5	off	off	off	off	on
6	on	on	on	on	on

根据实际情况，环境亮度变化K的取值范围为 0-800lux，将其平均分为 8 段，分别为 0-100lux、100-200lux、200-300lux、300-400lux、400-500lux、500-600lux、600-700lux、700-800lux。如上表 4.1 是环境亮度在 0-100lux 情况下，模糊控制输出的规则表。表中 $E_{0-100}$ 表示在环境亮度为 0-100lux 范围内，各个 LED 灯接收不同的控制指令时对应的反馈信息。表中 K 表示 LED 灯编号，一共有 LED1、LED2、LED3、LED4、LED5 五个 LED 灯。表中P代表控制指令，P可以取值“0”、“1”、“2”、“3”、“4”、“5”、“6”。当P取值“0”，即上位机发出关闭所有 LED 灯指令时，LED 灯反馈信息应该不受亮度变化值影响，均反馈“off”；当P取值“1”，即上位机发出打开 LED1 灯指令时，LED1 灯反馈“on”，其余 LED 灯反馈“off”；同理，当P取值

“2”、“3”、“4”、“5”时，相应的 LED2、LED3、LED4、LED5 反馈“on”，其余反馈“off”；当 P 取值“6”，即上位机发出打开所有 LED 灯指令时，所有的 LED（即 LED1、LED2、LED3、LED4、LED5）均反馈“on”。

经过测试，本文设计的系统不受环境亮度变化值影响，所以可以相应的得到环境亮度在 100-200lux、200-300lux、300-400lux、400-500lux、500-600lux、600-700lux、700-800lux 时的模糊控制规则表与表 4.1 类似，在此不赘述。第五章中渐变测试对该模糊控制采取了详细的演示与测试。

### 4.3 上位机控制平台的软件设计

考虑到系统跨平台性，上位机控制平台使用 Java 作为主要开发语言，采用 B/S 架构。为了达到系统数据管理以及可扩展性的目的，上位机控制平台采取模型-视图-控制器（MVC）设计模式。

考虑到 B/S 架构以及 MVC 设计模式的要求，本系统中的上位机控制平台采用 Struts2.2.3+Spring3.1.0+Mybatis3.0.6 整合架构实现开发，集成开发环境选用 Myeclipse10，并选用 MySQL 数据库。

前端页面选用 JSP 技术完成开发，利用 Struts 配置文件实现与后端 Java 程序之间的交互，后端 Java 程序与 MySQL 数据库之间通过 Mybatis 配置文件实现数据交互，Spring 框架对工程整体进行配置。

#### 4.3.1 无线控制的设计与实现

基于 wifi 的智能楼宇控制系统中无线控制的硬件部分主要通过 ESP8266 作为热点，上位机通过无线通信的方式进行控制，主要采用 socket 通信，socket 无线通信算法的实现方式如下：

基于已经构建的无线网络，要实现上位机控制平台对 Arduino 嵌入式设备的无线控制还需要实现具体的进程间通信。根据计算机网络经典的开放系统互连基本参考模型，负责实现进程间通信的是传输层。本系统中采用传输层的 TCP 协议，使通信更可靠。构建一条 TCP 连接需要两个端点，这里的端点就是套接字（Socket）。套接字就是 IP 地址与端口号的组合。在构建无线局域网时，已经设置了 ESP8266 无线传输模块的 IP 地址，则要确定 ESP8266 无线传输模块的套接字则只要再设定其工作端口号即可。另外，还需要指定 ESP8266 串口 wifi 扩展板的套接字类型和传输协议。这部分参数依然在 ESP8266 无线传输模块配置页面进行配置，

截图如下图 4.2 所示：



图 4.2 ESP8266 串口 wifi 扩展板配置页面截图

通过配置页面设置 ESP8266 无线传输模块的套接字类型为 **Server**，传输协议为 **TCP**，本地端口号，即 ESP8266 无线传输模块的工作端口号为 9000。所以，可以得到 ESP8266 无线传输模块的套接字为(192.168.4.1:9000)，由于 ESP8266 无线传输模块的套接字为 **Server** 类型，所以在上位机控制平台的后端 Java 程序中只要创建自己的 **Socket** 实例，并将它连接到 ESP8266 无线传输模块的套接字即可建立 **TCP** 连接。

上位机控制平台软件的后端 Java 程序中，**InfoAction** 类的 **send** 方法实现了 **Socket** 通信算法，其代码逻辑如下：

1、创建 **Socket** 对象：使用 **Socket** 类的构造器，以 ESP8266 无线传输模块的 **IP** 地址和工作端口号为传入参数构建自身的 **Socket** 实例，并尝试将此 **Socket** 实例通过所在网络连接到根据构造器传入参数查找到的 **Socket** 实例；

2、获取输入输出流对象：通过 **Socket** 实例调用 **getOutputStream** 方法获得一个输出流对象，调用 **getInputStream** 方法获得一个输入流对象；

3、将输出流包装成打印流：使用 **PrintWriter** 类的构造器，以先前获取的输出流对象作为传入参数构建一个打印流对象；

4、输出指令字符：通过 **Info** 类的实例调用 **getInformation** 方法获取待发送的指令字符，并通过打印流对象调用 **print** 方法，将获取的待发送指令字符传递给 **print** 方法；

5、将输入流包装成字符流：由输入流对象建立一个输入流读取器对象，再根据输入流读取器对象构建一个缓冲读取器对象；

6、按行读取字符数据：通过缓冲读取器对象调用 **readLine** 方法读取一行输入信息，并将该行信息添加到用于暂存信息的 **StringBuilder** 对象中，然后继续调用 **readLine** 方法读取下一行输入信息，添加到 **StringBuilder** 对象中，直至读取到输入信息末尾；

7、关闭相应的资源：按照顺序依次关闭缓冲读取器对象、输入流读取器对象、输入流对象、打印流对象、输出流对象和 **Socket** 实例，结束此次通信。

另外需要说明的是，设计代码时对于上述逻辑都做了 try-catch 处理，捕获各类异常。如当尝试与 ESP8266 无线传输模块建立 Socket 连接失败时，会弹出提示框，显示“未能成功连接到电路板”，接着再次跳转至历史通信数据查询页面。

综上所述，可以得到上位机控制平台与 Arduino 嵌入式设备之间的 socket 无线通信算法流程如下图 4.3 所示。

需要说明的是，图 4.3 中只显示了在创建 Socket 对象时判断是否存在异常，实际上程序中对于整个 Socket 通信过程都做了 try-catch 的异常处理。

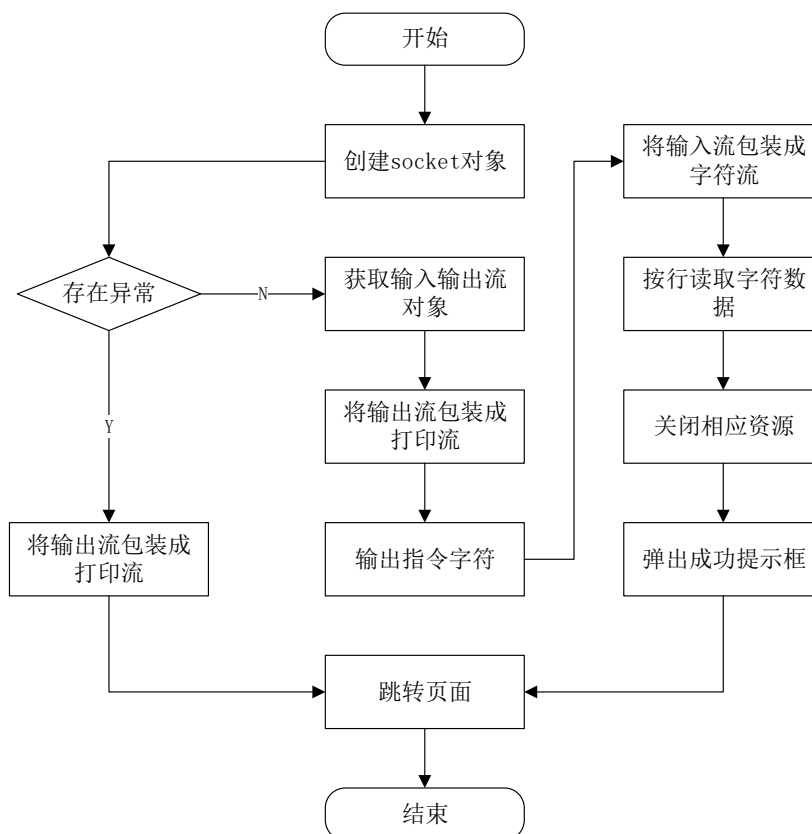


图 4.3 Socket 无线通信算法流程图

## 4.3.2 软件实现

### 1.项目具体设计

上位机控制平台为 Java Web 项目，开发主要在 Myeclipse10 中进行，另外数据库设计通过 MySQL Workbench 实施。为使得描述更加清晰，将上位机控制平台的软件设计分为前端页面设计、后端 Java 程序设计和数据库设计三个部分。

#### 前端页面设计：

前端页面采用 Java 服务端网页（JSP）技术实现，总共包括六个 JSP 页面，分别为登录页面（index.jsp）、用户信息展示页面（userList.jsp）、历史通信数据展示页面（infoList.jsp）、

登录状态校验失败页面（notlogin.jsp）、添加新用户页面（userAdd.jsp）和更新用户信息页面（userUpdate.jsp）。

登录页面：即上位机控制平台的主页，用户访问上位机控制平台时首先跳转到此登录页面。该页面写有一张用来允许用户输入登录信息的表单。

用户信息展示页面：用户登录完成后进入到该页面。该页面不仅含有目前已登录用户的信息，还包括对目前已登录用户信息进行修改和删除的超链接、“查看电路状态”超链接、“注销登录”超链接以及“添加新用户”超链接。

历史通信数据展示页面：若此时用户登录状态是有效的，则在用户信息展示页面点击“查看电路状态”超链接跳转到该页面。若用户是首次点击“查看电路状态”跳转到历史通信数据展示页面，则该页面不包含 Arduino 嵌入式设备的 IP 地址。该页面以表格的形式展示了当前数据库中存储的所有以往的通信记录，此通信记录包括来自 Arduino 嵌入式设备的反馈信息以及接收反馈信息的时间。另外，此页面还包括一个用于发送控制指令的表单，表单中包含了一个允许用户输入指令字符的文本域和一个“发送”按钮。用户在此文本域中输入控制指令后点击“发送”按钮，经过后端 Java 程序的处理，控制平台与 Arduino 嵌入式设备通信后，将再次跳转到历史通信数据展示页面，但此时该页面中包含了 Arduino 嵌入式设备的 IP 地址和刚刚发送的控制指令所对应的通信记录。

登录状态校验失败页面：用户在未登录的情况下，或在登陆后长时间未进行任何操作的情况下，试图执行除登录以外的操作时跳转到此页面。此页面包含登录状态校验失败相关的提示信息以及跳转到登录页面的超链接，另外此页面还设置了等待三秒自动跳转至登录页面的功能。

添加新用户页面：用户在登录状态完成的情况下，在用户信息展示页面点击“添加新用户”超链接进入到这个页面。此页面含有一张用来获取新用户的用户名和密码的表单。

更新用户信息页面：用户在登录状态有效的情况下，在用户信息展示页面点击“修改”超链接跳转到此页面。此页面包含了允许用户输入新密码的表单。

### 后端 Java 程序设计：

对后端 Java 程序实现分层设计，一共分为如下几层：Action 层、Service 层、Dao 层和 Domain 层。各层功能如下：

Action 层：负责接收用户通过前端页面提交的请求，执行相应的动作，动作可能调用 Service 层的某些服务。

Service 层：接受 Action 层的调用，提供对应的服务，服务方法可以调用 Dao 层的一些函数。

Dao 层：接受 Service 层的调用，通过 Sql 会话模板实例调用相应的方法，与数据库进行数据交互。

Domain 层：对不同类型的数据进行封装，包含 SqlMappers，其中配置了 sql 语句与 Dao 层方法传入参数的关系。

Action 层包含 InfoAction 和 UserAction 两个类，InfoAction 类负责执行和发送指令相关的动作，UserAction 类负责执行和用户数据相关的动作。在 InfoAction 类中，通过 send 方法实现上位机控制平台和 Arduino 嵌入式设备之间的信息交互，通过 queryAllInfo 方法实现对数据库中存储的所有历史通信记录的查询。在 UserAction 类中，通过 login 方法实现对前端页面登录请求的响应，login 方法中调用 Service 层相应的实例方法，另外，通过 logout 方法实现前端页面注销登录请求的响应，清楚会话对象并跳转至登录页面。

Service 层包含 InfoService 和 UserService 两个类，InfoService 提供和发送指令相关的服务，UserService 提供和用户数据相关的服务。在 InfoService 类中，采取 findAllInfo 实例函数向 InfoAction 类的 queryAllInfo 函数提供相关功能，并调用 Dao 层的相应实例函数。在 UserService 类中，通过 login 实例方法向 UserAction 类的 login 方法提供服务，并调用 Dao 层的相应实例方法。

Dao 层包含 InfoDao 和 UserDao 两个类，InfoDao 实现 Java 程序与数据库之间历史通信数据的交互，UserDao 实现 Java 程序与数据库之间用户数据的交互。在 InfoDao 类中，通过 findAll 实例方法向 InfoService 类的 findAllInfo 方法提供服务接口；在 UserDao 类中，通过 userLogin 实例方法向 UserService 类的 login 方法提供服务接口。

Domain 层包含 Info 和 User 两个类，Info 类用于封装通信历史数据，User 类用于封装用户信息。Info 类定义了 id、information 和 time 这三个维度，information 用来存放的 Arduino 嵌入式设备发送的反馈信息，time 用来存放反馈信息的接收时间。User 类也设定了 id、username 和 password 三个数据列，其中用户名存储在 username 列中，密码存储在 password 列中。

### 数据库设计：

上位机控制平台的数据库部分使用 MySQL Workbench 进行设计，首先设计用户信息数据表 and 通信记录数据表。用户信息数据表对应后端 Java 程序的 Domain 层的 User 类，而通信记录数据表对应后端 Java 程序的 Domain 层的 Info 类。如下表 4.1 所示为用户信息数据表：

表 4.1 用户信息数据表

ID	username	password
1	***	***

用户信息数据表包含整数类型字段 ID，用于在数据库中唯一表示用户信息；用户名存储

在字符串类型字段 `username` 中；密码存储在字符串类型字段 `password` 中。如下表 4.2 所示为通信记录数据表：

表 4.2 通信记录数据表

ID	information	time
1	***	***

通信记录数据表包含整数类型字段 `ID`，用于在数据库中唯一表示通信记录；字符串类型字段 `information`，用于存放来自 Arduino 嵌入式设备的反馈信息，字符串类型字段 `time`，用于存放接收反馈信息的时间，精确到秒。

## 2.项目运行机制分析

考虑前文所述的系统安全性的要求，上位机控制平台设置了用户登录功能，用户访问上位机控制平台的主页首先需要登录，否则不具备任何的操作权限。现以用户登录过程为例，说明上位机控制平台的一般内部运行机制。

1、用户信息表单提交：首先，用户访问上位机控制平台主页，输入用户名和密码，点击登录按钮，前端 JSP 页面以 Post 方式向后端的 Action 层提交此包含用户输入信息的表单，指明表单命名空间为 `“/user”`，action 属性为 `“user_login”`。

2、构建 Action 层实例：接着，在 Struts 的配置文件中，找到命名空间为 `“/user”` 的一对 package 标签，在这对 package 标签中找到 name 为 `“user_*”` 的 action 标签，`“*”` 表示可为任意值，因此根据所提交表单的 action 属性为 `“user_login”` 可以匹配到这对 action 标签。这对 action 标签中指明了构造 `UserAction` 类的实例，并运行 `UserAciton` 类中名为 `“*”` 的方法，在这里由于所提交表单的 action 属性为 `“user_login”`，所以指明了运行 `UserAciton` 类中名为 `login` 的方法。tomcat 根据上述配置信息，构建了 `UserAction` 的实例，然后该类中名为 `login` 的方法得到了执行。

3、运行 Action 层方法：运行 `UserAction` 类中名称为 `login` 的方法。首先判断根据所提交表单生成的 `user` 对象是否为空，若 `user` 对象不为空则通过 `UserService` 类的实例调用该类的 `login` 方法，传入 `user` 对象。这里 `UserService` 类的实例是根据 Spring 的注解方式自动构建的。

4、运行 Service 层方法：运行 `UserService` 类中的 `login` 方法，在该方法中，通过 `UserDao` 类的实例调用 `UserDao` 类的 `userLogin` 方法，并将读入的 `user` 对象传递给 `userLogin` 方法。

`UserDao` 类的实例也是通过 Spring 的注解的方法自动建立的。

运行 Dao 层方法：执行 `UserDao` 类中的 `userLogin` 方法，在该方法中，通过 `SqlSessionTemplate` 类的实例调用该类的 `selectOne` 方法，并将获得的 `user` 对象赋值到 `selectOne` 方法的传入参数 `parameter`，将 `selectOne` 方法的另一传入参数 `statement` 赋值为字符串

“userLogin”。通过查询 sqlMappers 包下的配置文件,可得到 id (对应 SqlSessionTemplate 类中的 selectOne 方法的传入参数 statement) 为 “userLogin” 且 parameterType (即 SqlSessionTemplate 类中的 selectOne 方法的传入参数 parameter 的类型)为 User 的 select 标签,该标签中包括了详细的 sql 语句。

执行查询操作并返回结果:首先让 MyBatis 与数据库建立连接,然后执行相应的 sql 语句,可以获得查询结果,若结果集为空则 SqlSessionTemplate 实例调用的 selectOne 方法返回 null,若结果集不为空则 selectOne 方法根据结果集构建一个 User 类的实例并返回。接着,从 Dao 层逐层向上反馈 selectOne 方法的返回结果,一直到 Action 层。

判断登录是否成功:在 UserAction 类的 login 方法中,获取到 UserService 实例所调用的 login 方法的返回值,判断该返回值是否为 null。如果该用户先前已在数据库中录入过信息,则此时返回值应为获取的来自底层的 User 类的实例,反之则返回值应为 null。若返回值为不为 null,则在会话对象中添加获取的 User 类实例,并通过 Struts 跳转到登录成功的页面;若返回值为 null,那么跳回登录页面并警告“用户名或密码错误”。

可以发现,使用框架的优点在于避免了自己编程实现对象在各个层之间传递的困难。还需要说明的是,上述运行机制为上位机控制平台内部的一般运行机制,不涉及与 Arduino 嵌入式设备之间的信息交互,上位机控制平台对 Arduino 嵌入式设备的无线控制将在后文中做进一步说明。

### 3.项目工作流程分析

上位机控制平台的一般工作流程如下:

验证用户:获得用户登录信息,与数据库实现交互,判定此用户是否存在,若不存在则回到登录页面,若存在则网页跳转到登陆成功页面。

展示电路状态:若用户登陆成功,则在登录成功页面显示用户信息以及超链接“查看电路状态”,用户点击超链接“查看电路状态”则从数据库中获取所有的历史通信数据并跳转到展示页面以列表的形式进行展示。历史通信数据包括上位机控制平台接收到的 Arduino 嵌入式设备反馈的状态信息和接收此反馈信息的时间。

发送控制指令:在历史通信数据展示页面,用户可以在页面底部的输入框输入想要发送的指令,点击“发送”按钮则后端 Java 程序获取此指令字符,并通过 wifi 网络发送给 Arduino 嵌入式设备。

展示反馈信息:若指令字符成功被 Arduino 嵌入式设备接收,并且后端的 Java 程序也得到了反馈信息,则后端 Java 程序将此反馈信息和接收时间写入数据库中,并弹出提示框,显示“发送成功”,然后执行获取所有历史通信数据的方法并跳转到历史通信数据展示页面。此



4.3.3 上位机界面软件设计

本文所设计的基于 Arduino 的智能楼宇控制系统中的上位机控制平台主要包括前端展示模块、后端控制与数据处理模块、数据存储模块和通信模块，如图 4.4 所示。各模块功能如下：

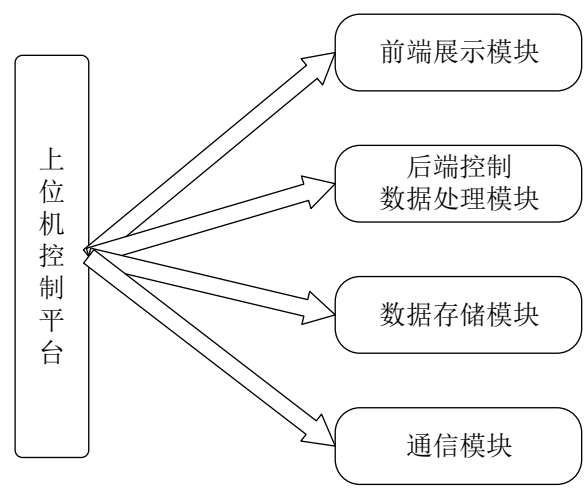


图 4.4 上位机控制平台组成图

- 前端展示模块：负责向用户展示良好的交互界面，用户通过前端展示模块输入控制指令、查询反馈信息、管理用户数据等。
- 后端控制与数据处理模块：负责协调控制其他三个模块，负责将用户输入的控制指令交付给通信模块，负责将通信模块收到的反馈信息交付给前端展示模块，负责实现对数据存储模块中数据的增删改查。
- 数据存储模块：负责持久化用户信息和通信数据，负责响应后端控制与数据处理模块的增删改查指令。
- 通信模块：负责将控制指令发送给 Arduino 嵌入式设备的通信模块，负责接收 Arduino 嵌入式设备的通信模块发送来的反馈信息。

4.4 本章小结

本章在第三章设计的硬件的基础上对智能控制系统的软件进行了设计，主要包括：Arduino 嵌入式平台的软件设计、模糊控制算法的研究和上位机控制平台的软件设计三大块。其中 Arduino 嵌入式平台的软件设计主要介绍了 Ardino 作为微控制器对传感器的反馈数据的

采集和控制；上位机控制平台的软件设计分析了基于 wifi 的无线控制的设计、上位机软件的设计（前端设计、后端控制数据处理设计、数据存储设计、通信设计）和模糊控制算法的研究。本文的控制系统的实现由本章与第三章的硬件设计组成。

## 第五章 基于 wifi 的物联网智能控制系统的实现和测试

本章对基于 wifi 的物联网智能控制系统进行了实物演示和测试。在完成一个实际的控制系统的演示的基础上，证实前面几章通过软硬件设计的控制系统的可行性。并且测试这个系统的性能和功能，从而验证系统的可靠性和实用性。

### 5.1 嵌入式平台的实现

智能控制系统的硬件设备如下图 5.1 所示：

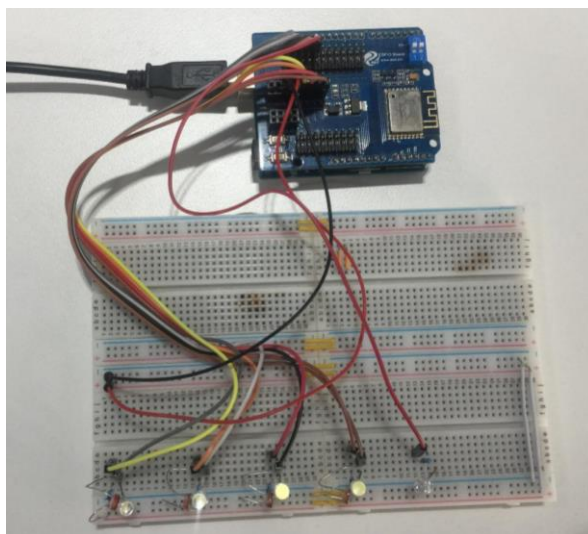


图 5.1 智能控制系统硬件设备图

图 5.1 中，下面部分为开发板，包括 5 个 LED 小灯，每个小灯旁有一个光敏电阻实时采集附近光源亮度值，上面部分为智能终端控制器，由两个部分组成：ArduinoUNO R3 开发板和 ESP8266 无线传输模块。具体实物图见图 5.2 所示：

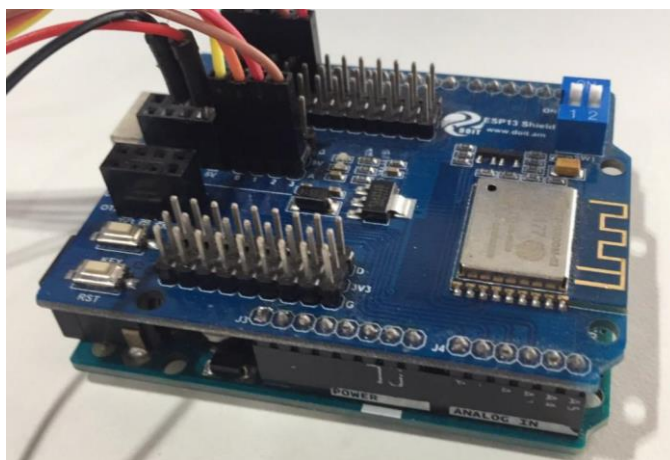


图 5.2 Arduino UNO R3 开发板与 ESP8266 无线传输模块实物图

图 5.2 中上方蓝色的是 ESP8266 无线传输模块，下方绿色的是 Arduino UNO R3 开发板。

## 5.2 上位机控制平台界面的实现

上位机控制平台的登录页面如下图 5.3 所示：

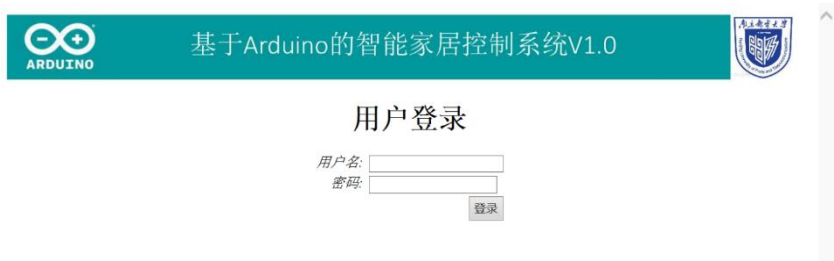


图 5.3 用户登录页面

使用该上位机控制平台时，用户一般先访问此页面，登录成功后用户才具备一系列操作权限，否则不具备任何操作权限。系统管理页面如下图 5.4 所示：



图 5.4 系统管理页面

用户登录成功后便会跳转到此页面。该页面包括五个按钮，分别是：查看电路状态、管理状态数据、管理通信数据、管理用户数据和注销登录。接下来逐个介绍这几个按钮。

点击查看电路状态按钮，当硬件设备连接成功，会跳出连接成功提示，然后会出现如下图 5.5 所示界面。



图 5.5 查看电路状态页面（斜线表示 LED 灯亮）

最上面显示“当前 Arduino 电路板 IP 地址”，只有当电脑或者手机连到 ESP8266 作为热点发出的 wifi 信号时，才能对终端进行控制。下面 5 个模拟 LED 灯就是模拟终端设备，与终端设备实时状态一致。再下面的表格则是 5 个 LED 灯控制信息，包括时间、每个灯的亮灭情况和光敏电阻测得的每个 LED 灯的亮度。右下方的指令框，可以输入“0-5”的阿拉伯数字，“1-5”分别控制控制 5 个 LED 灯的打开，数字“0”可以同时关闭当前打开的 LED 灯。如图 5.6 所示为控制系统整体实物图：

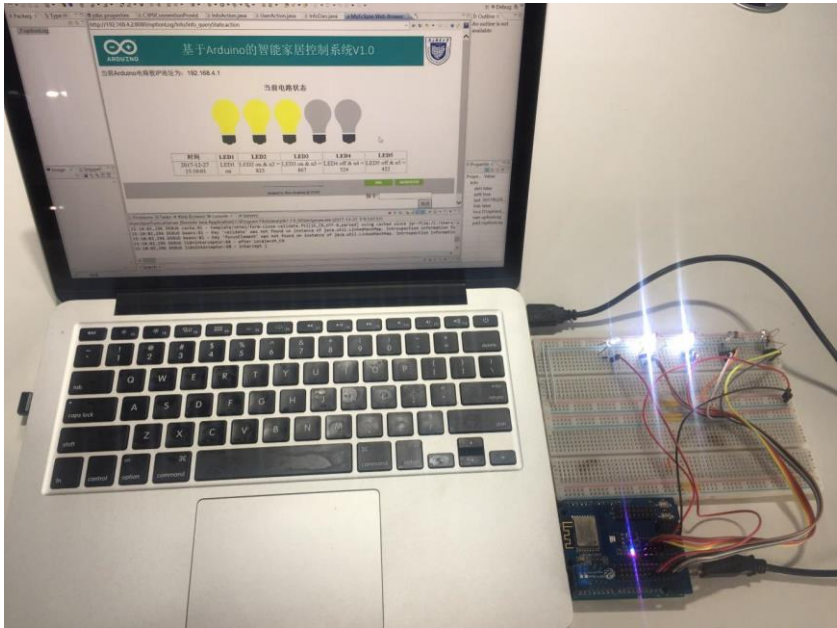


图 5.6 控制系统整体实物图

返回管理页面，点击管理状态数据按钮，历史状态信息展示页面如下图 5.7 所示：



图 5.7 历史状态信息展示页面

在该页面可以看到不同时间点（时间精确到秒），对各个 LED 灯的控制情况，该页面通过从数据库中提取数据，然后显示到该页面上。

返回管理页面，点击管理通信数据按钮，下图 5.8 所示为历史通信数据信息页面：



图 5.8 历史通信数据展示页面

在该页面可以看到信息采集发送时间、发送信息、接受时间、接受信息和系统时延这 5 个消息，与历史状态信息展示页面类似，该页面信息也是通过从数据库中提取数据，然后显示到该页面上。

返回管理页面，点击管理用户数据按钮，跳转到控制平台的用户信息展示页面。下图 5.9 所示为控制平台的用户信息展示页面：



图 5.9 用户信息展示页面

用户在此页面可以查看自己的用户信息，并且可以通过超链接跳转到管理页面、添加新用户或注销登录状态。

返回管理页面，点击注销登录按钮，跳出当前管理页面，再次进入登录界面。如图 5.3 所示。

### 5.3 系统测试

本系统的核心功能是对 LED 亮灭的无线控制与检测，测试的主要指标包括四个：控制准确性、检测准确性、通信可靠性和系统延迟。控制准确性指的是由上位机控制平台发送指令对 5 个 LED 完成亮灭控制反馈的准确性，检测准确性指的是对于 5 个 LED 亮灭状态检测的准确性，控制准确与检测准确是最基本的系统功能需求。通信可靠性主要是指上位机控制平台与 Arduino 嵌入式设备之间无线通信的可靠性。系统延迟是指从用户点击发送开始，直至用户可以通过系统看到 LED 亮灭情况为止所经过的平均时间。较高的通信可靠性与较低的系统延迟是本系统希望拥有的良好性能。

控制准确性、模糊控制准确性、通信可靠性和系统延迟与控制准确度、模糊控制完成度、

通信成功率和平均系统延迟这四个指标一一对应，并由这四个指标衡量，下面分别对这四个指标进行详细说明，并补充说明对模糊控制算法采用的渐变测试的步骤与结果。

### 5.3.1 测试指标

#### (1) 控制准确度

通过上位机控制平台向 Arduino 嵌入式设备发送控制指令，在确认 Arduino 嵌入式设备收到上位机控制平台发送的控制指令的情况下，通过观察 LED 真实亮灭情况判断控制指令是否准确执行，对所有的控制指令进行多次测试。定义控制准确度为控制指令得到正确执行的次数占控制指令发送总次数的百分比。设控制准确度为 $N_0$ ，得出如下公式：

$$N_0 = \frac{N_s}{N} \times 100\% \quad (5.1)$$

公式 5.1 中， $N_s$ 为测试中控制指令得到正确执行的次数， $N$ 为测试中控制指令发送总次数。

#### (2) 模糊控制完成度

在确认 Arduino 嵌入式设备收到上位机控制平台发送的控制指令并进行了正确的 LED 控制操作的情况下，通过上位机控制平台的历史通信数据展示页面获取刚刚发送的控制指令对应的反馈信息，反馈信息依据输入控制指令，通过模糊算法检索模糊规则库，完成对应模糊语言转换输出，判断反馈信息与 LED 实际亮灭情况是否一致，若一致，则控制指令完成次数加一，若不一致，则控制指令失败次数加一。定义控制指令完成度即控制指令完成次数占控制指令发送总次数的百分比，控制指令发送总次数即控制指令失败次数加控制指令完成次数。设控制指令完成度为 $D_0$ ，得出如下公式：

$$D_0 = \frac{D_s}{D} \times 100\% \quad (5.2)$$

公式 5.2 中， $D_s$ 为控制指令达标次数， $D$ 为控制指令发送总次数。需要特别说明的是，对于控制指令完成度的计算是在确保 Arduino 嵌入式设备收到上位机控制平台发送的控制指令并正确执行的情况下进行的，这样排除了通信可靠性与控制准确性带来的干扰。

另外还需要说明的是，对 LED 进行亮灭检测时，考虑两种特殊情况，第一种特殊情况是：发送了打开 LED 的控制指令，但实际上由于某种原因（如 LED 内部断路），LED 并未真的被点亮，此时亮灭检测电路的检测结果，即 Arduino 嵌入式设备发送给上位机控制平台的反馈信息应表明 LED 并未真的被点亮。在试图关闭 LED 时同样存在类似的情况，即 Arduino 嵌入式设备收到了关闭的指令，但 LED 并未真的被关闭，此时反馈信息也应当表明 LED 并未真的被关闭。在对系统的控制指令完成度进行测试时，特别加入这两种情况。

#### (3) 通信成功率



通信成功是指 Arduino 嵌入式设备收到了上位机控制平台发送的指令字符并且上位机控制平台也收到了 Arduino 嵌入式设备的反馈信息，通过前端页面弹出的提示框即可判断通信是否成功。定义通信成功率即通信成功的次数与通信总次数的比值。设通信成功率为 $C_0$ ，得出如下公式：

$$C_0 = \frac{C_s}{C} \times 100\% \quad (5.3)$$

公式 5.3 中， $C_s$ 为通信成功的次数， $C$ 为通信总次数。可以预见到，由于系统使用的无线局域网的网络结构简单，在无线 AP 信号较好的情况下，通信成功率应当是较高的。

#### (4) 平均系统延迟

首先对系统延迟时间进行定义：在上位机控制平台的后端 Java 程序中进行设定，在获取前端页面发送请求时记录当前系统时间为 $T_1$ ，与 Arduino 嵌入式设备通信成功并成功跳转至历史数据显示页面时记录当前系统时间为 $T_2$ ，设系统延迟时间为 $T$ ，得出如下公式：

$$T = T_2 - T_1 \quad (5.4)$$

公式 5.4 中，平均系统延迟定义为系统延迟时间的算术平均值，设平均系统延迟为 $\bar{T}$ （单位为毫秒），得出如下公式：

$$\bar{T} = \frac{\sum_{i=1}^K (T_2^i - T_1^i)}{K} \quad (5.5)$$

公式 5.5 中， $T_1^i$ 表示第 $i$ 次实验中，上位机控制平台的后端 Java 程序获取前端页面发送请求时记录的当前系统时间， $T_2^i$ 表示第 $i$ 次实验中，后端 Java 程序与 Arduino 嵌入式设备通信成功并成功跳转至历史数据显示页面时记录的当前系统时间， $K$ 表示实验总次数。这里需要说明的是，对系统延迟时间的测量都是建立在后端 Java 程序与 Arduino 嵌入式设备通信成功的前提下的。

#### (5) 渐变测试

渐变测试是指上位机控制 LED 灯亮灭的同时，通过逐渐改变环境的亮度值，检测环境亮度值对 Arduino 嵌入式设备反馈信息正确度的影响。定义渐变测试成功率为反馈正确信息次数占环境亮度值改变总次数的百分比。设渐变测试成功率为 $W_0$ ，得出如下公式：

$$W_0 = \frac{W_s}{W} \times 100\% \quad (5.6)$$

公式 5.6 中， $W_s$ 为渐变测试反馈正确信息次数， $W$ 为环境亮度值改变总次数。需要说明的是，经过测试，环境亮度值范围在 0-800lux 之间，因此环境亮度值改变分为 8 个范围，分别是 0-100lux、100-200lux、200-300lux、300-400lux、400-500lux、500-600lux、600-700lux、700-800lux。且渐变测试成功率是建立在后端 Java 程序与 Arduino 嵌入式设备通信成功的前



提下的。

5.3.2 测试结果与分析

在一般的照明环境下，设定了合理的 LED 亮灭控制与检测电路的门限值后，对上述前四种指标进行了两组测试，每组中对每种指标进行 50 次实验，得到测试结果如下表 5.1 所示：

表 5.1 系统测试结果

指标 组号	控制准确度	模糊控制完成度	通信成功率	平均系统延迟/ms
1	100%	100%	100%	1210
2	100%	100%	100%	1164

分析表 5.1 可以得到：本系统的控制准确度、模糊控制完成度和通信成功率都是比较令人满意的，而 1 秒以上、2 秒以下的平均系统延迟可以说是基本上符合系统功能要求的。

上述第五种测试——渐变测试的测试步骤及结果如下，通过对环境亮度值的 8 个范围划分，每个范围的亮灭控制各测试 10 次，然后分别对每个 LED 灯亮灭反馈信息采集，得到测试结果如下表 5.2 所示：

表 5.2 渐变测试结果

环境亮度值(lux)	0-100		100-200		100-200		200-300	
LED 亮灭控制	0	1	0	1	0	1	0	1
LED 反馈信息	off	on	off	on	off	on	off	on
LED 判断正确率	100%	100%	100%	100%	100%	100%	100%	100%
环境亮度值(lux)	400-500		500-600		600-700		700-800	
LED 亮灭控制	0	1	0	1	0	1	0	1
LED 反馈信息	off	on	off	on	off	on	off	on
LED 判断正确率	100%	100%	100%	100%	100%	100%	100%	100%

表 5.2 为 LED1 在不同的亮度环境下，针对亮灭控制指令所反馈的信息以及 LED 判断正确率。从表 5.2 中可以看出，本系统在较大的环境亮度值范围的情况下，依然有非常稳定的成功率。经过测试，其余 LED 灯具有与 LED1 相同的判断正确率，在此不赘述。

图 5.10 为渐变测试中的一个示例，测试的是环境亮度值在 300-400lux 时，上位机发送控制指令，Arduino 嵌入式设备反馈信息的正确率。图中每个 LED 灯对应表格中的每一列，每一列的第二行显示当前 LED 灯的状态，同时每个 LED 灯旁的光敏电阻会将各个小灯周边的

当前Arduino电路板IP地址为：192.168.4.1

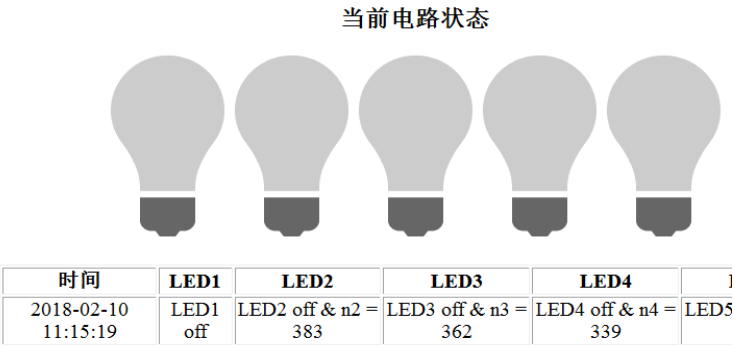


图 5.10 渐变测试过程图 1

如下图 5.11，即为控制 LED1 灯亮时的测试场景。

当前Arduino电路板IP地址为：192.168.4.1

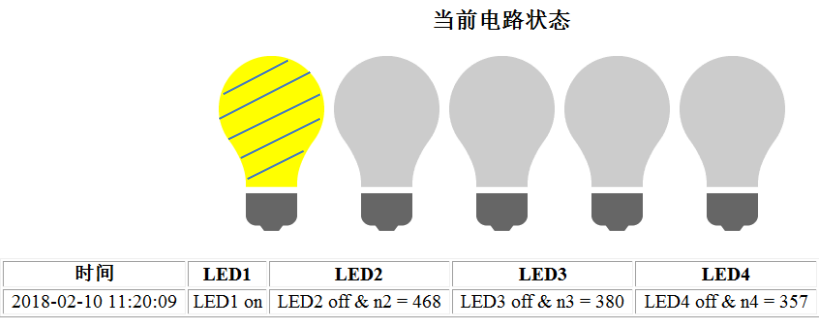


图 5.11 渐变测试过程图 2（斜线表示 LED 灯亮）

图中通过指令控制 LED1 亮，表格中对应的 LED1 显示为“on”，其余几个 LED 灯也会将光敏电阻反馈的亮度值输出到表格中，并显示当前 LED 灯状态为“off”。

### 5.4 本章小结

本章介绍通过构建一个基于 wifi 的物联网智能控制系统并完成测试的过程。文中结合第三章的硬件设计与第四章的软件设计，设计出一套控制系统。然后对整套系统的几项性能与功能指标（控制准确性、模糊控制准确性、通信可靠性、系统延迟）进行了测试，并补充对模糊控制算法的渐变测试。

## 第六章 总结与展望

### 6.1 工作内容与成果

文章研究了一种应用于智能楼宇中的控制系统和上位机的控制平台。整套环境由传感器节点和嵌入式平台构成。在国家大力实行智慧城市建设的背景下，智能楼宇是体现智慧城市的一个关键产物，文章提出的基于 wifi 的物联网智能楼宇控制系统有希望在未来的智慧楼宇领域发挥基础作用。

本论文的主要工作和成果如下：

(1) 通过讲述智慧城市产生的大背景，联系国内外智能楼宇的发展，分析了智能楼宇的提出和发展经历，介绍了国内外研究现状和发展趋势，阐述了文章研究的目的和意义以及能对社会带来的积极意义。

(2) 介绍了本文平台所使用的相关技术，包括无线数据传输、控制算法、Web 框架等技术，同时也介绍了平台的各个层面的设计理念。

(3) 研究了基于 wifi 的物联网智能楼宇控制系统的硬件平台，该平台涵盖两部分：基于 Arduino 的智能终端的硬件和基于 wifi 的智能终端的硬件。而基于 Arduino 的硬件囊括了作为微控制器的 Arduino 硬件模块、与传感器节点（光敏传感器）的电路模块等；基于 wifi 的硬件采取的是无线传输模块 ESP8266。

(4) 设计了基于 wifi 的物联网智能楼宇控制系统的软件平台，该平台包括三部分，分别是：Arduino 嵌入式平台的内部检测程序、模糊控制算法和上位机控制平台的软件设计。其中 Arduino 嵌入式平台的检测程序与模糊控制算法的设计达到了 Arduino 作为微控制器对传感器的反馈数据的采集和控制的目的；上位机控制平台的软件设计涵盖了基于 wifi 的无线控制的设计和上位机软件的设计。

### 6.2 未来展望

文章提出了一种智能楼宇控制系统，完成了通过上位机对智能终端模块的信息采集，检测控制，传输存储等功能。但是想要构建一个人性化完整的智能楼宇控制系统还有许多需要改进的地方：

(1) 本文在实现智能楼宇控制系统时，智能终端的硬件设备是直接通过传感器、电阻、导线等元器件设计的，应用到智能楼宇中时，需要将各传感器组件封装成一个传感器模块，

便于安装与连接。

(2) 因为无线传输模块使用的是 ESP8266 wifi 嵌入式设备, 整套系统的 wifi 通信需要依赖该硬件作为 AP 热点发出无线信号, 控制设备接入该信号才能实现远程控制。智能楼宇的实现需要多套硬件设备以达到大面积的覆盖和控制的目的, 设备与设备之间的通信是无法回避的问题, 需要研究类似 zigbee 组网的无线信号的设计, 实现设备与设备之间通信, 上位机针对每个设备可以实现精准控制。

(3) 上位机控制平台的界面设计功能按钮较为简单, 实现的功能较少。在上位机平台中, 有几个页面设计中, 有部分浏览器不兼容的情况。要想做一个成熟的上位机控制平台软件, 浏览器兼容性和软件的稳定性是必不可少的。

## 参考文献

- [1] Hui Shao, Haijian Fu.Design and implementation of intelligent building engineering information management system.7th International Conference on Intelligent Computation Technology and Automation.2014.
- [2] Clancey W J.Methodology for building an intelligent tutoring system[J].Artificial intelligence and instruction,1987,8(9):521-527.
- [3] 滕明华,李梁.对建筑电气技术与智能建筑热点问题的探讨[J].建筑工程技术与设计,2014,6(12):56-61.
- [4] 刘云浩.物联网导论[M].北京:科学出版社,2010:30-50.
- [5] 黄永生.关于计算机物联网的应用分析[J].数字化用户,2014(8).
- [6] 范停涛.智能楼宇照明控制系统的开发与研究[D].北方工业大学,2009.
- [7] Zheng Zeng, Rongxiang Zhao, Huan Yang, Micro-sources design of an intelligent building integrated with micro-grid, Energy and Buildings, 2013, 57: 261-267.
- [8] José A.Oliveira-Lima,Vasco Delgado-Gomes, João F. Martins,Celson Lima, Standard-based service-oriented infrastructure to integrate intelligent buildings in distributed generation and smart grids, Energy and Buildings, 2014, 76: 450-458.
- [9] R.M. Silva, J. Arakaki, F. Junqueira, D.J. Santos Filho, P.E. Miyagi,Modeling of active holonic control systems for intelligent buildings, Automation in Construction, 2012, 25: 20-33.
- [10] Johnny K.W. Wong, Heng Li, Application of the analytic hierarchy process (AHP) in multi-criteria analysis of the selection of intelligent building systems, Building and Environment, 2008, 43(1).
- [11] 丁鼎.智能楼宇照明控制系统的设计与实现[D].华东理工大学,2016.
- [12] 孙杰. 基于WIFI的智能家居传感节点设计与实现[D].中国海洋大学,2015.
- [13] Milos Manic,Kasun Amarasinghe,Juan J. Intelligent Buildings of the Future[J]. IEEE Industrial Electronics Magazine.2016,12.
- [14] 杨金翠,方滨兴,翟立东,张方娇.面向物联网的通用控制系统安全模型研究[J].通信学报,2012,11(11):51-52.
- [15] 孙利民,沈杰,朱红松.从云计算到海计算:论物联网的体系结构[J].中兴通信技术,2011,17(1):3-7.
- [16] <https://baike.baidu.com/item/Petri%E7%BD%91/3322876?fr=aladdin>[OL]
- [17] 李少远,王景成.智能控制[M].北京:机械工业出版社,2005:23-55.
- [18] 席爱民.模糊控制技术[M].西安:西安电子科技大学出版社,2008:24-36.
- [19] 李士勇.模糊控制.神经控制和智能控制论[M].哈尔滨:哈尔滨工业大学出版社,1998.9-20.
- [20] Kven M,Passino Stephen Yurkovich.Fuzzy Control[M].北京:清华大学出版社,2001:35-63.
- [21] 仇成群,刘成林,沈法华,陈杰.基于Matlab和模糊PID的汽车巡航控制系统设计[J].农业工程学报,2012,3(6):197-198.
- [22] 王君.基于模糊控制策略的温室远程智能控制系统的研究[M].吉林:吉林大学,2015:28-32.
- [23] Rionel Belen Caldo, Joseph T. Seranilla, Derrick J. Castillo, Kevin S. Diocales, Winreal D. Gulle, Briandale. Design and Development of Fuzzy Logic Controlled Dimming Lighting System Using Arduino Microcontroller[J].8th IEEE International Conference Humanoid, Nanotechnology, Information Technology Communication and Control, Environment and Management (HNICEM).2015.
- [24] Lian Li,Dong Jia.Research on Air Conditioning System of Subway Station Based on Fuzzy PID Control[J]. 4th International Conference on Information Science and Control Engineering.2017.
- [25] 高迟,吕元锋.基于模糊神经网络的温室控制系统[J].农机化研究,2009,(10):187-189.
- [26] 马子鹏.基于神经网络的模糊智能系统的研究与实现[M].西安:西安科技大学,2005:50-80.

- [27] 王士同.神经模糊系统及应用[M].北京:北京航空航天大学出版社,1998:20-50.
- [28] 黄丽.BP神经网络算法改进及应用研究[M].重庆:重庆师范大学,2008:29-35
- [29] 王志斌,孔亚楠,刘永成.遗传算法优化BP神经网络的大功率LED结温预测[J].光电子·激光,2014.7:1304-1309.
- [30] 李刚.疯狂Java讲义(第2版):电子工业出版社,2014:2.
- [31] 赵景晖.Java 程序设计:北京机械工业出版社,2005:1-2.
- [32] 明日科技.Java从入门到精通(第3版):清华大学出版社,2014.
- [33] 王薇,杜威.Java程序设计与实践:华中科技大学出版社,2010:9.
- [34] 周志明.深入理解Java虚拟机:机械工业出版社,2011:4.
- [35] 袁剑君,陈潜,王凯婧.公路工程监理的数据信息化管理系统研究[J].湖南交通科技.2009.
- [36] 王薇,杜威.Java程序设计与实践:华中科技大学出版社,2010:10.
- [37] 徐明亮.基于B/S架构的智能家居远程视频监控系统的研究与设计[D].上海:上海师范大学,2016:1-2.
- [38] 周屹,李艳娟.数据库原理及开发应用(第二版):清华大学出版社,2013:8.
- [39] 崔阳,张维华,白云峰.一种基于Arduino的智能家居控制系统[J].电子技术应用.2014(04).
- [40] 甘本祓.微波传输线设计手册[M].北京:人民邮电出版社,1981.
- [41] 甘本祓,吴万春.现代微波滤波器的结构与设计[M].北京:科学出版社,1973.
- [42] <http://gcc.gnu.org/wiki/avr-gcc>[OL]
- [43] 陈吕洲.Arduino程序设计基础:北京航空航天大学出版社,2014:5-6.
- [44] Lian Li, Dong Jia. Research on Air Conditioning System of Subway Station Based on Fuzzy PID Control[J]. 4th International Conference on Information Science and Control Engineering.2017.
- [45] <https://baike.baidu.com/item/%E6%A8%A1%E7%B3%8A%E6%8E%A7%E5%88%B6/1577503?fr=aladdin> [OL].

## 附录 1 程序清单

第四章 Arduino 模糊控制算法程序（Arduino programme 文件夹）

第四章 上位机控制平台前端界面程序（Front-end programme 文件夹）

第四章 上位机控制平台后端界面程序（Back-end programme 文件夹）

## 附录 2 攻读硕士学位期间撰写的论文

- (1) 周栋梁、郭永安，基于树莓派+Arduino 的 WiFi 反馈控制系统的研究，智能计算机与应用，2017.4；
- (2) 周栋梁、王堂辉、梁茵、郭永安，基于 ARM 嵌入式的手持人脸识别装置，第二届全国物联网技术与应用学术会议（CIoT 2016） 第十一届全国无线电应用与管理学术会议（CRAM 2016），2016.12；
- (3) 梁茵、周栋梁，面向智能家居的语义知识库系统，物联网与无线通信，2016.12。



### 附录 3 攻读硕士学位期间申请的专利

- (1) 朱洪波、周栋梁、翟天翔、郭永安，一种基于树莓派和 Arduino 的 wifi 控制反馈系统，201710117851.8，2017.4；
- (2) 朱洪波、翟天翔、周栋梁、白易初、郭永安，“一种基于物联网的智能鱼塘预警系统”，201710117325.1，2017.3.1；

## 附录 4 攻读硕士学位期间参加的科研项目

- (1) 国家 973 计划课题，智能协同移动服务模型（2013CB329005）；

## 致谢

时光飞逝，转眼间硕士研究生两年半的学习生涯就要接近尾声了。在南京邮电大学读书已经有 7 年的时光了，回首这段日子，尤其在研究所期间的这两年多的时间，使我的人生轨迹发生了变化。在这段时间里，很多人在学术上、生活上给予了我很多的帮助和鼓励。

首先，我要感谢我的导师朱洪波教授，朱教授在我们刚入师门便对我们的生活和学习给予了关心和问候，更是教会了我们做人做事，到现在那些话语也常常萦绕在耳边。我要感谢我的指导老师朱霞老师，在课题研究遇到小问题时，朱霞老师像一位大姐姐一样，总能及时的给予帮助。

同时，我也要感谢我的师兄姐妹们，是他们无论在生活上或者科研项目上都给予了我很大的帮助，在做科研项目遇到问题时，他们总是无私的提供帮助；在生活中遇到困难时，他们总能为我出谋划策。

最后，我还要感谢一直支持我陪伴我的家人，是他们的关怀和支持让我在科研道路上没有后顾之忧，是他们对我的鼓励，使我的论文最终完成。