

大连东软信息学院

毕业设计（论文）

论文题目：室内环境监测物联网系统的设计与实现

学 院：____计算机学院____
专 业：____物联网工程____
学生姓名：____邓子帆____
学生学号：____17111800222____
指导教师：____曹新 李缙栋____
导师职称：____教授 讲师____
完成日期：____2021 年 5 月 10 日____

大连东软信息学院
Dalian Neusoft University of Information

室内环境监测物联网系统的设计与实现

摘 要

室内环境数据能为其他活动提供重要参考。在日常生活与科学研究中，都广泛需要室内环境数据。在各类科研活动中，尤其是在物联网工程中，室内环境数据是科研人员的重要参考数据。

本文设计出了一种室内环境监测物联网系统，能够采集个人居住场所内的环境数据，并通过局域网在移动应用程序与 Web 应用程序中共享采集到的数据。本系统也包括若干执行器，用于调节环境。本系统的硬件部分基于 Arduino 开发板，通过连接的众多传感器采集环境数据，再通过 Wi-Fi 将数据发送至个人电脑上的 MySQL 数据库，移动应用程序与 Web 应用程序则从中提取数据。开发板型号为 Arduino UNO，是 Arduino 官方推荐的入门级型号，各项基本功能齐全。传感器包含 BMP180 气压传感器、DHT11 温相对湿度传感器、PM2.5 传感器、MQ-2 煤气传感器。Wi-Fi 模块为 ESP-01。执行器包含：制冷设备、制热设备、加湿器、空气净化器、声光报警器。当遇到预定的情况时，相应执行器将立即启动。所有模块均通过串口或软件串口连接至 Arduino。移动应用程序能够向用户实时展示环境数据；Web 应用程序供用户查询历史数据。

用户通过使用本系统，能够实时掌握当前室内的环境数据，为其工作提供参考。用户也可以通过 Android 应用程序主动控制执行器来调节环境。当环境中存在可燃气体时，本系统将立即发出警报。

关键词：环境监测，物联网系统架构，Arduino

Design and Implementation of an Indoor Environmental Monitoring IoT System

Abstract

Indoor environmental data can provide an important reference for other activities. Indoor environmental data is widely needed in everyday life and in scientific research. In all kinds of scientific research activities, especially in IoT projects, indoor environmental data is an important reference data for researchers.

In this paper, an IoT system for indoor environmental monitoring is designed, which is able to collect environmental data from individual living places and share the collected data in mobile applications and web applications via a local area network. The system also includes a number of actuators to regulate the environment. The hardware part of the system is based on an Arduino development board, which collects environmental data from a number of connected sensors and sends the data via Wi-Fi to a MySQL database on a PC, from which the mobile and web applications extract the data. The development board is the Arduino UNO, the official Arduino recommendation for entry-level models, with all the basic functions. The sensors include BMP180 air pressure sensor, DHT11 temperature and humidity sensor, PM2.5 sensor, MQ-2 gas sensor, Wi-Fi module ESP-01, actuators: cooling device, heating device, humidifier, air amplifier, audible and visual alarm. When a predetermined situation is encountered, the corresponding actuator will be activated immediately. All modules are connected to the Arduino via a serial port or a software serial port. The mobile applications can display environmental data to the user in real time; web applications are available for the user to query historical data.

By using this system, the user is able to keep track of the current environmental data in the room in real time, providing a reference for their work. Users can also actively control the actuator to regulate the environment via the Android app. The system will immediately sound an alarm when combustible gases are present in the environment.

Key words: Environmental Monitoring, IoT System Architecture, Arduino

目 录

摘 要	I
ABSTRACT.....	II
第 1 章 绪 论.....	1
1.1 论文主要研究内容	1
1.2 国内外现状	2
第 2 章 关键技术介绍	3
2.1 Arduino	3
2.2 ESP8266EX (SoC)	5
2.3 MySQL	5
2.4 Express.js	6
2.5 Android	7
2.6 夏普 PM2.5 传感器.....	7
第 3 章 系统需求分析	9
3.1 业务需求	9
3.2 用户需求	9
3.2.1 用户需求描述	9
3.2.2 用例建模	10
3.3 模块需求描述	10
3.4 性能需求	13
3.5 系统开发环境	13
3.6 系统任务的可行性分析	13
3.6.1 技术可行性	13
3.6.2 系统安全性分析	14
第 4 章 系统设计	15

4.1 设计指导思想和原则	15
4.1.1 指导思想	15
4.1.2 设计原则	15
4.1.3 技术路线	16
4.2 系统架构	17
4.3 系统功能设计	17
4.3.1 硬件设计	18
4.3.2 软件设计	20
4.3.3 网络设计	22
4.3.4 系统接口设计	23
4.4 数据库设计	23
第 5 章 系统实现	25
5.1 硬件实现	25
5.1.1 整体效果实现	25
5.1.2 模块连线	25
5.1.3 各个功能模块的实现	29
5.2 软件实现	30
5.2.1 Web 应用程序的实现	30
5.2.2 Android 应用程序的实现	31
第 6 章 系统测试	33
6.1 各个功能模块的测试	33
6.1.1 测试 DHT11 传感器	33
6.1.2 测试 BMP180 传感器	34
6.1.3 测试 PM2.5 传感器	34
6.1.4 测试 MQ-2 传感器	34
6.1.5 网络测试	35
6.1.6 Web 应用程序测试	36
6.1.7 数据库测试	36

6.2 整体测试	37
第 7 章 总 结.....	38
参考文献	39
致 谢.....	40

第1章 绪论

室内环境数据能为其他活动提供重要参考。在日常生活与科学研究中，都广泛需要室内环境数据。在各类科研活动中，尤其是在物联网工程中，室内环境数据是科研人员的重要参考数据。

本文设计出了一种室内环境监测物联网系统，能够采集个人居住场所内的环境数据，并通过局域网在移动应用程序与 Web 应用程序中共享采集到的数据。本系统也包括若干执行器，用于调节环境。本系统的硬件部分基于 Arduino 开发板，通过连接的众多传感器采集环境数据，再通过 Wi-Fi 将数据发送至个人电脑上的 MySQL 数据库，移动应用程序与 Web 应用程序则从中提取数据。开发板型号为 Arduino UNO，是 Arduino 官方推荐的入门级型号，各项基本功能齐全。传感器包含 BMP180 气压传感器、DHT11 温相对湿度传感器、PM2.5 传感器、MQ-2 煤气传感器。Wi-Fi 模块为安信可 ESP-01。执行器包含：制冷设备、制热设备、加湿器、空气净化器、声光报警器。当遇到预定的情况时，相应执行器将立即启动。所有模块均通过串口或软件串口连接至 Arduino。移动应用程序能够向用户实时展示环境数据；Web 应用程序供用户查询历史数据。

用户通过使用本系统，能够实时掌握当前室内的环境数据，为其工作提供参考。用户也可以通过 Android 应用程序主动控制执行器来调节环境。当环境中存在可燃气体时，本系统将立即发出警报。

1.1 论文主要内容

本文设计了一种用于监测室内环境的物联网系统。该系统以一个 Arduino 单片机为核心，连接若干传感器与执行器。传感器将其周围的物理信息转换为模拟信号或数字信号再传入单片机。当监测到的环境数据变化到预设的区间时，则通过执行器来调节环境。用户也可以直接通过移动应用程序控制执行器。传感器包含温相对湿度传感器、气压传感器、PM2.5 传感器、煤气传感器；执行器包含空气净化器、加湿器、制冷设备、制热设备、声光报警器。Arduino 单片机同时也通过局域网将采集到的数据发送到使用者的个人电脑上的数据库中。开发板通过 ESP-01 模块连接至用户的局域网，数据库运行在使用者的个人电脑上，采用 MySQL 数据库。采集到的数据供 Web 应用程序与 Android 应用程序使用。Web 应用程序用于查询历史数据。Android 应用程序不仅向使用者展示当前的环境数据，同时也允许使用者直接控制执行器。

1.2 国内外现状

人类 80-90%的日常时间是在室内度过的，所以室内环境质量直接影响健康和工作效率。一些专业人士利用先进的传感器网络和通信技术，提出了用于改善生活环境的监测系统^[1]。

Alhmiedat 和 Samara 开发了一个低成本的 ZigBee 传感器网络架构，以实时监测室内环境。该系统在室内安装四个传感器节点，并收集超过四个星期的数据。环境数据通过 Zigbee 通信协议进一步传输进行分析。本文作者分析了厨房做饭时的二氧化碳、苯、氮氧化物和氨，以评估室内环境质量，而其他传感器则收集了卧室、客厅和办公区的期望输入。

Abraham 和 Xinrong Li 提出了一个用于监测室内环境的低成本无线传感器网络（WSN）系统。该系统使用低成本微型气体传感器（一氧化碳、挥发性有机化合物和二氧化碳）设计，并使用 Arduino 单片机作为处理单元。该监测系统的网状网络使用 Zigbee 模块开发，为通信提供低功率、低成本的无线解决方案。

Benammar 等人利用 WSN 技术设计了一个端到端的室内环境监测系统。传感器节点通过 XBee PRO 无线电模块与网关通信。本研究中的传感器节点包括一套经过校准的传感器单元，一个名为 Wasmote 的数据存储单元，以及被称为 Gas Pro 的传感器接口板。本研究中网关的主要作用是处理从目标地点收集的室内环境数据，并通过服务器进行可靠传输。该系统使用开源物联网平台 Emoncms，以确保室内环境数据的长期存储和实时监测。为保证供电，作者还为传感器网络设计了独立的电池单元。

根据室内环境监测系统的发展趋势，过去几年中，大多数研究人员都致力于基于 WSN 的设计。ZigBee 是最可靠的通信协议；ATmega 单片机管理实时数据收集；而 Raspberry Pi 是在目标环境中建立传感器网络的另一个常见选择。

基于物联网的实时监测系统延迟较低、功耗较小、维护工作也较少。因此，大多数研究人员和工业制造商都被这种架构所吸引。专家透露，物联网系统可以监测大量的参数，甚至不影响系统性能。

2020 年 8 月 1 日，最新修订的中国国家标准《民用建筑工程室内环境污染控制标准》正式实施。其中要求控制的污染物种类增多，在原有的 5 种指标（氡、甲醛、苯、氨、TVOC）上增加了甲苯和二甲苯的检测指标。除此之外，还有新风量的测试。新修订的标准中大部分污染物浓度限值，比上一版要更加严格。如此严格的标准不仅能保护居住者的身体健康，也将促进环境监测行业的技术进步。

第 2 章 关键技术介绍

2.1 Arduino

Arduino 项目开始于 2005 年，是意大利 Ivrea 交互设计学院学生的工具，旨在为新手和专业人士提供一种低成本和简单的方法来创建使用传感器和执行器与环境互动的设备。Arduino 开发板是设计使用各种微处理器的单片机。这些开发板配备了一组数字和模拟输入/输出（I/O）引脚，可与各种扩展板（“Shield”）或面包板（用于原型设计）和其他电路连接。这些开发板也具有串行通信接口，包括某些型号的通用串行总线（USB），可用于加载程序。单片机可以使用 C 和 C++ 来编程，其使用的标准 API 也被称为“Arduino 语言”。除了使用传统的编译器工具链，Arduino 项目还提供了一个集成开发环境和一个用 Go 开发的命令行工具（Arduino CLI）^[2]。

以下特点使得 Arduino 对初学者相当友好：

价格低廉。与其单片机平台相比，Arduino 价格相对低廉。最便宜的 Arduino 模块版本可以手工组装，即使是预先组装好的 Arduino 模块价格也低于 50 美元。

跨平台。Arduino IDE 可以在 Windows、macOS 和 Linux 上运行。而大多单片机系统只限于 Windows。

简单、清晰的编程环境。Arduino IDE 对于初学者来说易于使用，但对于高级用户来说也足够灵活。

开源和可扩展的软件。Arduino 软件作为开源工具发布，可由有经验的程序员进行扩展。该语言可以通过 C++ 库进行扩展，想了解技术细节的人可以从 Arduino 跃迁到它所基于的 AVR C 编程语言。

本文使用的 Arduino 型号是官方推荐的入门级型号 Arduino UNO。它是一块基于 ATmega328P 的单片机。它有 14 个数字输入/输出引脚（其中 6 个可用作 PWM 输出），6 个模拟输入，一个 16MHz 的晶振，一个 USB-B 接口，一个电源插孔，一个 ICSP 引脚和一个复位按钮。它包含了单片机所需的一切，只需用 USB 线将其连接到电脑上，或用交流电适配器或电池供电即可开始使用。该开发板的主要技术规格如表 2.1 所示。

表 2.1 Arduino UNO 开发板

工作电压	5V
微处理器	ATmega328
Bootloader	Arduino Uno
时钟频率	16 MHz
输入电压	7-12V
输入电压	6-20V
数字 I/O 端口	14（6 个 PWM 输出口）
模拟输入端口	6
直流电流 I/O 端口	40mA
直流电流 3.3V 端口	50mA
Flash 内存	32 KB（ATmega328）（0.5 KB 用于引导程序）
SRAM	2 KB（ATmega328）
EEPROM	1 KB（ATmega328）
尺寸	75x55x15mm

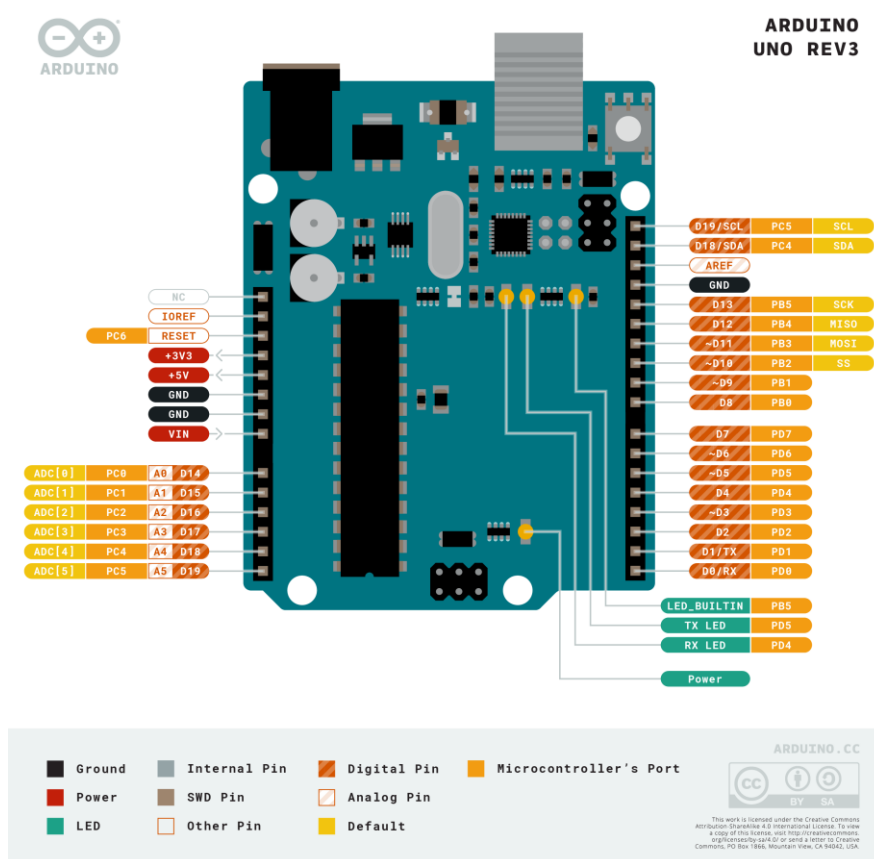


图 2.1 引脚示意图

2.2 ESP8266EX (SoC)

ESP8266EX 是一种低成本的 Wi-Fi SoC，具有完整的 TCP/IP 协议栈和单片机能力，由乐鑫科技生产。该芯片本身也是一个具备完整功能的单片机^[3]。其有如下特性：

（1）ESP8266EX 内置超低功耗 Tensilica L106 32 位 RISC 处理器，CPU 时钟速度最高可达 160 MHz，支持实时操作系统 (RTOS) 和 Wi-Fi 协议栈，可将高达 80% 的处理能力留给应用编程和开发。

（2）高度集成。ESP8266EX 集成了 32 位 Tensilica 处理器、标准数字外设接口、天线开关、射频 balun、功率放大器、低噪放大器、过滤器和电源管理模块等，仅需很少的外围电路，可将所占 PCB 空间降低。

（3）低功耗。ESP8266EX 专为移动设备、可穿戴电子产品和物联网应用而设计，通过多项专有技术实现了超低功耗。ESP8266EX 具有的省电模式适用于各种低功耗应用场景。

（4）性能稳定。ESP8266EX 的工作温度范围大，且能够保持稳定的性能，能适应各种操作环境。

需要注意的是，EXP8266EX 只是乐鑫公司设计的 SoC，并不是完整的模块。该模块如图 2.2 所示。

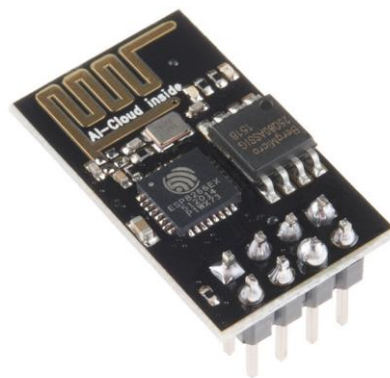


图 2.2 安信可 ESP-01 模块

2.3 MySQL

MySQL 是一个开源的关系数据库管理系统 (RDBMS)。关系型数据库将数据组织成一个或多个数据表，其中的数据类型可能相互关联；这些关系有助于结构化数据。SQL 是程序员用来创建、修改和提取关系数据库中的数据，以及控制用户对数据库的访问的一种语言。除此之外，像 MySQL 这样的 RDBMS 还与操作系统协同，在计算机的存储系统中实现关系型数据库、管理用户、允许网络访问、帮助测试数据库的完整性、创建备份。

MySQL 有独立的客户端，允许用户使用 SQL 与 MySQL 数据库直接互动，但更多时候 MySQL 与其他程序一起使用，以实现需要关系型数据库能力的应用。

Sysbench 是一种流行的开源基准测试，用于测试开源 DBMS。MySQL 8.0 与旧版对比的测试结果如图 2.3 所示：

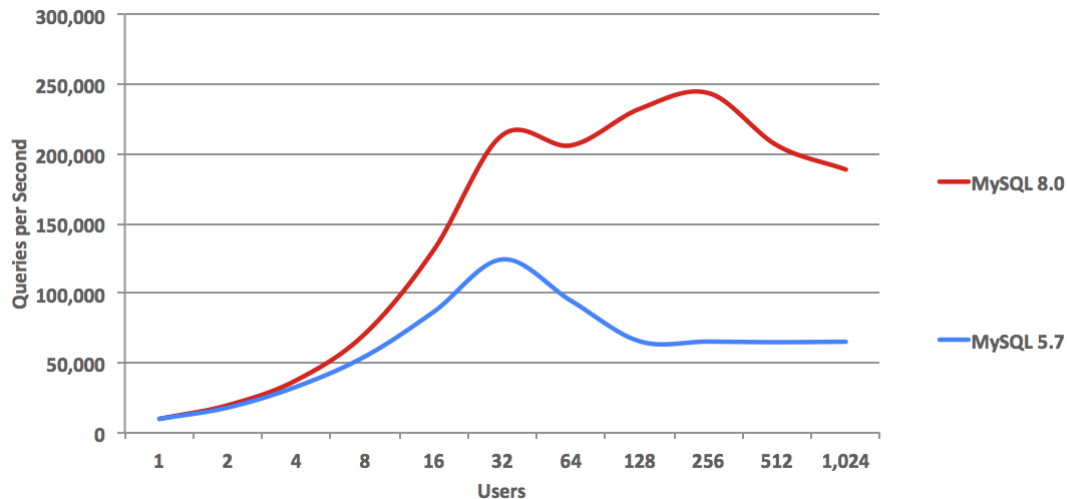


图 2.3 读写

2.4 Express.js

Express.js，简称 Express，是 Node.js 的一个后端 Web 应用程序框架，为 Web 和移动应用程序提供了一组强大的功能。它被称为 Node.js 事实上的标准服务器框架^[4]。

更快的服务器端开发。Express.js 以函数的形式提供了 Node.js 的许多常用功能，可以在程序的任何地方随时使用。

中间件。中间件是程序的一部分，可以访问数据库、客户端请求和其他中间件。它主要负责系统化地组织 Express.js 的不同功能。

路由：Express.js 提供了一个非常先进的路由机制。借助 URL，它可以保存网页的状态。

模板化：Express.js 提供了模板引擎，允许开发者通过在服务器端建立 HTML 模板在网页上建立动态内容。

调试：Express.js 通过提供一个调试机制使调试变得更容易，它能够准确地指出网络应用中存在错误的部分。

图 2.4 说明了一个典型的基于 Express.js 的 Web 应用程序的架构。

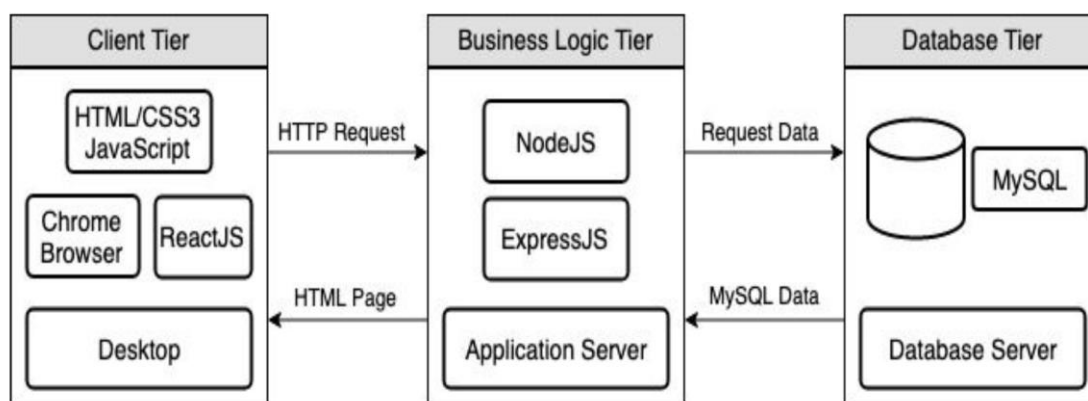


图 2.4 典型的基于 Express.js 的 Web 应用程序架构

2.5 Android

毫无疑问，Android 是需求最猛烈的移动操作系统，它拥有 85% 的市场份额，在谷歌应用商店中有超过 304 万个应用程序，从日历、浏览器、社交媒体等日常工具，还有纷繁的游戏，到企业移动应用程序，应有尽有。Android 的受欢迎程度不仅仅是由其广泛的可用性和在 Google Play 商店中可以找到的应用程序多样性决定的。更是因为它允许纳入最前沿的技术，使开发过程更快、更有效。所有这些特点使 Android 应用程序成为许多企业的正确选择^[5]。图 2.5 显示了 Android 近年来迅猛的势头。

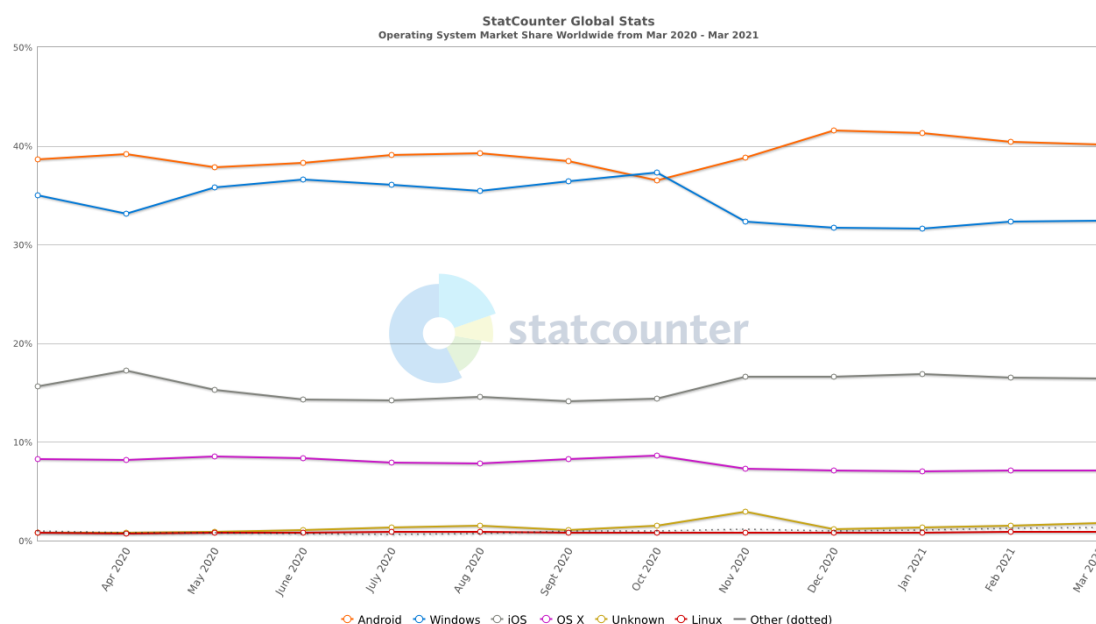


图 2.5 操作系统市场份额

2.6 夏普 PM2.5 传感器

夏普粉尘传感器的工作原理是光散射。光电探测器和 LED 发射器在传感器的矩形封装内以一定的角度相互照射，矩形封装的两侧有一个灰尘通孔。含有灰尘颗粒的空气流入传感器室，导致 LED 发射器的光线向光电探测器散射。灰尘传感器输出一个电

压值，该电压值根据散射光的强度而变化，而散射光的强度又与传感器室中的灰尘水平相对应。然后，实际的灰尘密度（或灰尘浓度）可以通过线性关系从输出电压值中计算出来^[6]。

该传感器的工作原理如图 2.6 所示；其与 Arduino 的接线图如图 2.7 所示。

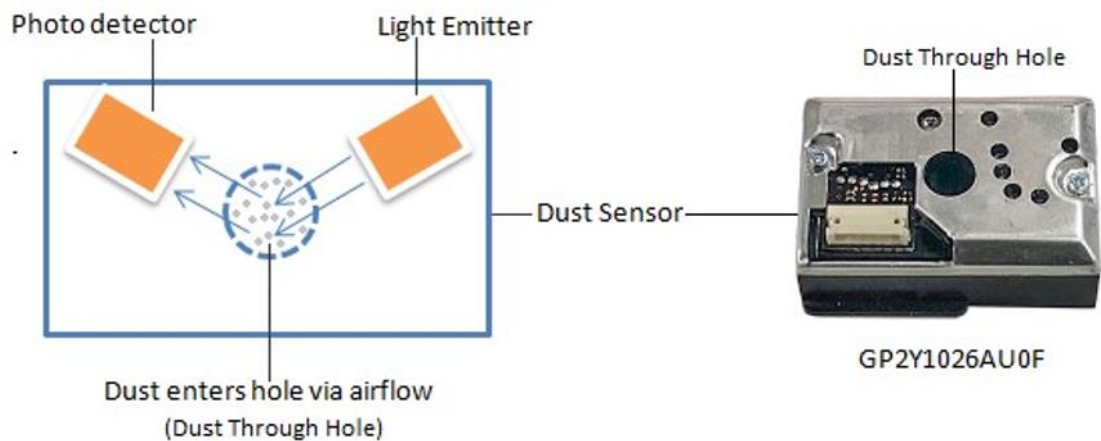


图 2.6 夏普 PM2.5 传感器工作原理示意

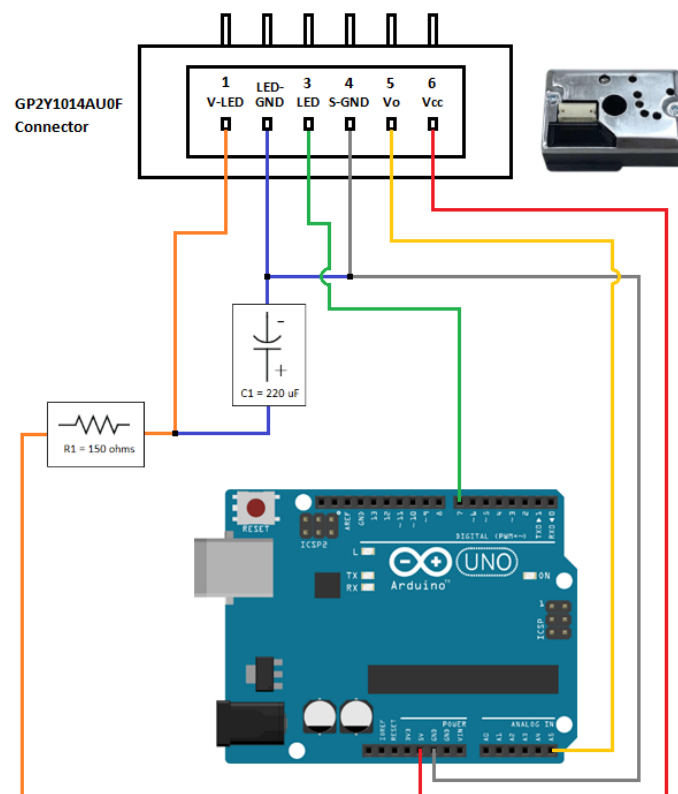


图 2.7 夏普 PM2.5 与 Arduino 连接的示意图

第3章 系统需求分析

3.1 业务需求

如今人类每天有大量的时间在室内度过。在没有任何监测设备前，居住者只能根据自己的身体反应来评估当前室内环境的质量。例如当窗户关闭时，人就会感到闷热；光照较暗时，人的瞳孔就会放大；冷色温的光照源于上升的太阳，使人打起精神，而暖色温的光照源于下落的太阳，使人感到困倦。至于调节室内环境，在没有任何环境调节设备前，居住者也只有开窗通风等简单的方式^[7]。

所以居住者迫切地想知道自己所处的室内环境如何。在一个舒适的环境中工作或学习，不仅可以提高幸福感，还可以提高满意度，从而提高生产力和学习效率。当室内环境状况较差时，居住者也需要及时调节环境来满足正常的生活与工作^[8]。

（1）核心业务需求

- ①查看当前室内环境状况，包括温度、相对湿度、煤气、PM2.5。
- ②按预设参数维持室内环境。
- ③主动调节室内环境。
- ④室内出行危险气体时得到报警
- ⑤查看过去一段时间的历史室内环境数据

（2）辅助业务需求

- ①全天候确保监测系统稳定运行。
- ②提供 Web 应用程序。
- ③提供数据库访问权限。
- ④更美观的装置外观。

3.2 用户需求

3.2.1 用户需求描述

（1）书房工作

用户讲来到书房进行一天的学习与工作。首先，用户进入书房后，习惯性地拿起手机，打开监测系统的 App，扫一眼当前室内环境状况。App 上显示的数据均实时来自传感器的采集，并经由用户的个人电脑传输至 App。本系统中的各个传感器正实时监测环境是否符合预期。一旦某个传感器监测到的数值未处于预期的区间内，则所需调节的执行器将开始工作。当用户结束学习和工作，离开书房后，一般会习惯打开窗户通风透气。虽然外界的空气相比室内的更加清新，但仍然可能会有其他污染物。所

以本监测系统在用户离开书房后，仍然能够确保书房内的环境质量。

（2）养花

有相当一部分住宅的住户喜好在家里放置一些植物来净化空气。然而植物本身也需要一定的环境质量来保障其正常的生长发育。用户可以将本监测系统放置在室内的植物附近来确保其生长发育。例如，对于常春藤，其生长要求温度处于 18-25℃ 之间，那么当温湿度传感器监测到当前温度处于适合其生长发育的区间之外时，制冷设备或制热设备就将立即启动，以保障常春藤的生长发育。而像虎尾兰这样对相对湿度有一定要求的植物，本系统包含的加湿器就能确保相对湿度处于其生长发育所需的区间内。尤其是当用户需要外出一段时间时，本系统就能在用户离家时保障室内植物所需的生长环境。同时，用户也能随时在手机 App 上看到家里植物所处环境的状况。

（3）回顾历史数据

用户使用一段时间后，想知道过去这段时间内室内环境的质量改变得如何，或者打算迁移本系统到其他地方，又或是考虑是否继续使用本系统。那就需要检索过去一段时间的历史数据。本系统提供 Web 应用程序供用户检索。用户首先进入 Web 应用程序的主页。该网页提允许用户筛选要查看历史数据的时间段，以及要查看哪些传感器的数据。当用户选定时间段以及哪些传感器的数据后，Web 应用程序将立即从用户的个人电脑中的数据库内提取相应的数据并绘制成统计图，然后显示在网页上。

3.2.2 用例建模

用户使用本系统可以实现三个主要功能：查看室内环境状况；查看历史数据；主动调节环境。查看室内环境状况一般是通过打开手机 App 来实现，在 App 的主界面上可以直接浏览当前传感器采集到的数据及其反映的室内环境状况。查看历史数据要求用户进入本系统的 Web 应用程序，在 Web 应用程序的主页上，能够通过选取时间段来查看历史数据。当用户想要主动操纵执行器来直接调节环境时，也可以通过手机 App 来直接控制执行器。本系统的用例图如图 3.1 所示。

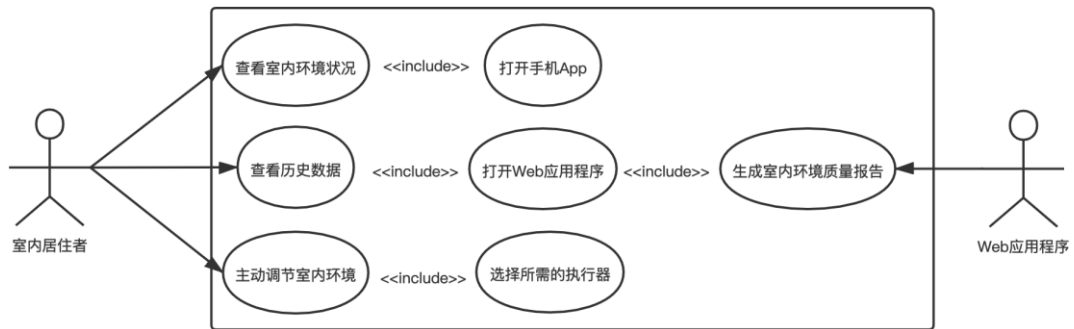


图 3.1 系统用例图

(1) 查看室内环境状况用例：主要指用户打开 Android App 并查看显示的数据的过程。

(2) 主动调节室内环境用例：主要指室内环境处于正常区间内，但用户察觉到异常并想主动控制执行器。

(3) 查看历史数据用例：主要指用户查看本系统运行一段时间后积累在数据库内的数据。

(1) 查看室内环境状况

查看室内环境状况是本系统最基本的功能。用户只需简单地拿出手机，打开 App 就可以快速浏览当前传感器采集到的数据及其对应的环境质量。App 上的这些数据均来自用户的个人电脑上的数据库。该功能也是本系统的价值的直接体现。本系统的其他所有功能也都基于传感器对环境的采集。其用例图如图 3.1 所示。

表 3.1 查看室内环境状况用例的用例规约

用例名称	查看室内环境状况
用例描述	拿出 Android 手机，打开本系统的 App，首页即显示全部室内环境状况
参与者	居住者
状态	通过审查
前置条件	系统部署完成
后置条件	无
基本操作流程	1. 安装本系统的 Android App2. 打开 App3. 首页即显示用户所需的全部室内环境状况
备用操作操作	无
异常场景	1. 传感器采集到异常数值 2. 局域网建立失败 3. App 连接数据库失败
涉及的实体	传感器、Android、Wi-Fi、MySQL 数据库

（2）主动调节室内环境

本系统中负责调节室内环境的各个执行器在没有任何干预的情况下，会自动根据传感器采集的实时室内环境数据来决定是否启动。但是在任何时候，用户都能通过 App 直接控制各个执行器。无论何时用户都拥有对本系统的自主权。其用例图如图 3.2 所示。

表 3.2 主动调节室内环境用例的用例规约

用例名称	主动调节室内环境
用例描述	用户感受到室内环境异常，于是启动 App，直接控制对应的执行器
参与者	居住者
状态	通过审查
前置条件	出现意料之外的室内环境异常
后置条件	无
基本操作流程	1. 打开 App。2. 转到控制执行器的 activity。3. 直接控制对应的执行器
备用操作操作	无
异常场景	执行器未能按预期工作
涉及的实体	1. 执行器 2. Android App

（3）查看历史数据

当本系统被用户部署到室内并使用一段时间后，数据库内就积累了一定量的数据。这些数据能直观地反应过去这段时间内室内环境的变化趋势。如果用户想查看历史数据，只要进入 Web 应用程序，在其主页上选择要查看的时间范围，即可看到由 Web 应用程序生成的统计图。其用例图如图 3.3 所示。

表 3.3 查看历史数据用例的用例规约

用例名称	查看历史数据
用例描述	当本系统在室内部署了一段时间后，用户需要查看过去这段时间室内环境的变化趋势
参与者	居住者
状态	通过审查
前置条件	数据库内积累了一定量的数据
后置条件	无
基本操作流程	1. 进入 Web 应用程序 2. 选择时间段
备用操作操作	无
异常场景	MySQL 数据库故障
涉及的实体	1. Express.js 网站框架 2. MySQL 数据库

3.3 模块需求描述

根据上一小节对本系统的用例建模，可以概括出本系统主要包含如下模块需求，见表 3.4。

表 3.4 系统模块需求

功能项	功能描述
查看当前室内环境	传感器每时每刻都在采集室内环境数据并上传至数据库，那么 App 就能从数据库中提取数据并展示在 App 上
自动调节室内环境	Arduino 单片机检测到传感器采集的数据存在异常时，则自动启动相应的执行器来改善室内环境
主动调节室内环境	无论何时，用户都可以主动通过 App 来直接控制执行器
查看历史数据	用户进入 Web 应用程序后，选择向回顾的时间段即可查询历史数据

3.4 性能需求

（1）传感器采集数据的频率

频率越高，数据分析就越准确，但对数据库的性能要去也越高。

（2）ESP-01 模块的信号范围

ESP-01 负责 Wi-Fi 通信，必须保证信号足以覆盖用户的室内。

3.5 系统开发环境

（1）Arduino UNO：Arduino IDE

（2）ESP-01：Arduino IDE

（3）Express.js：Visual Studio Code

（4）Android App：Android Studio

以上开发环境运行于 MacBook Pro（A1708）上，操作系统为 macOS 10.15.7。

3.6 系统任务的可行性分析

3.6.1 技术可行性

（1）Arduino UNO 是 Arduino 社区最广泛使用的一款开发板。其功能齐全，封装完整，代码简单，对初学者相当友好。Arduino 官方网站上也有相当齐全的参考文档供开发者检索。开发者只需简单地调用相关 API 就能方便快捷地操纵串口收发、I/O 引脚读取等功能^[9]。

（2）ESP8266EX 是一款低成本的 Wi-Fi SoC，带有全套 TCP/IP 协议栈，其本身就是一个完整功能的单片机。安信可公司基于该 SoC 设计的 Wi-Fi 通信模块为 ESP-01。它可以通过串口引脚与 Arduino 开发板直接通信。借助其强大的功能，使得物联网系统开发更加便捷。

（3）Express.js 是基于 Node.js 的一款后端 Web 框架。其最大的优势是，使用了与前端相同的编程语言 JavaScript。借此优势，前端与后端的开发效率将会飞跃到一个新的台阶。该框架默认使用的软件包管理器是 NPM（Node Package Manager）。

（4）部署在用户的个人电脑上的 MySQL 数据库是一款相当稳定且易于连接的数据库产品。NPM 提供 MySQL 库，可以很方便地在 Express.js 中连接数据库。MySQL 在世界范围内的广泛应用，尤其是在企业中的应用，充分说明该数据库产品的稳定与高效。

（5）部署在用户的智能手机上的 Android App 是用户访问本系统最便捷的方式。谷歌提供相当灵活并免费的 Android SDK 供开发者使用。在其强大的软件开发工具包的帮助下，开发者可以轻松地为企业、娱乐和游戏建立应用程序。该 SDK 有助于最大限度地减少开发和许可的成本。此外，业界也提供众多开发框架，使 App 成形更容易。

3.6.2 系统安全性分析

（1）Arduino UNO 搭载的芯片为 ATmega328P。该芯片的闪存（Flash memory）具备安全性设计：其闪存的程序空间分为两部分，启动加载器（Boot loader）部分和应用程序（Application program）部分^[10]。

（2）ESP-01 模块支持 WPA/WPA2 安全模式。

（3）Web 应用程序的安全性主要取决于其部署所在的网络的安全性。本系统的 Web 应用程序部署的网络是用户自己的局域网。根据统计数据，家庭局域网虽然入侵价值不高，但仍然存在一定的被攻击的风险。较为稳妥的方法是选购带有安全功能的路由器，以及使用 WAP2（Wi-Fi Protected Access II）密码。

（4）MySQL 数据库不仅被个人与企业使用，还广泛被世界各国的政府采用。所以对于其安全性，不仅有技术保证，还有各类政府部门颁发的安全性证书。况且本系统所在的局域网建立在用户的居所，一般不会有攻击者出现。

（5）Android 设备本身已内置安全功能。安全功能在可信执行环境（TEE）下运行，可确保操作系统的安全。App 采用沙盒化的运行模式，可以确保各款应用彼此独立，让其数据完全区分开来。

第 4 章 系统设计

4.1 设计指导思想和原则

4.1.1 指导思想

一、基于物联网架构的室内环境监测系统相比传统的环境监测系统，不仅能实时显示各项室内环境数据，更能通过传感器采集到的数据进行数据分析，为用户提供关于室内环境质量的最佳建议。

二、本系统充分适配先进的物联网技术架构。从传感器采集数据到 Web 应用程序的数据分析，都要保证系统的可用、高效、稳定。这就要求硬件系统、网络系统、软件系统的全方位互联互通。

三、最好的科技让人感受不到它的存在。本系统一旦部署后将全自动运行，充分为用户服务的同时不打扰用户。尽可能减少不必要的同时与报警。

4.1.2 设计原则

（1）模块化设计原理（Modularity）

软件工程的模块化允许应用程序被划分为模块，以及与类似的模块整合在一起。模块根据功能划分，程序员不参与其职责之外的模块。因此，新功能可以很容易地在单独的模块中进行编程。模块化的基本原则是：“系统应该由高内聚、低耦合的组件（模块）构建而成。”本系统中包含以下主要模块：传感器；Arduino 开发板；Web 应用程序；Android 应用程序。每个模块内部又包含更小的模块。他们之间通过各种硬件接口或软件接口连接在一起。

（2）抽象原理（Abstraction）

抽象原理要求软件工程中的组件应当独立于其实现，且组件可以是硬件或软件。组件的界面是用户的视角，而实现是开发者的视角。当一个组件是按照抽象原则设计的时，那么用户就不需要为了使用这个组件而了解这个组件是如何工作的，开发者也可以在不通知用户的情况下修改实现。本系统中，用户需要知道的只有“传感器有哪些”“如何使用 Android App 和 Web 应用程序”即可。依照这种设计原则，用户将获得极佳的产品体验。

（3）信息隐蔽原理（Information hiding）

信息隐藏原则由 D. L. Parnas 在 1972 年首次提出，是一个将软件系统分解成模块的标准。该原则指出，每个模块都应该对系统的其他部分隐藏一些设计决策，特别是那些一旦改变就会产生跨领域影响的决策。信息隐藏允许将实现隐藏接口内，即使实

现发生变化，接口也不会改变。本系统中，软件部分都各自解耦，在用户的个人电脑上包含 MySQL 数据库、Web 应用程序，二者间的连接则由 Web 应用程序的后端部分负责；在用户的 Android 智能手机上包含本系统的 App，App 则通过 MySQL 外部库来连接个人电脑上的数据库。

（4）逐步求精原理（Stepwise refinement）

逐步求精原理是指软件的开发是通过抽象层次来进行的。从较高的层次开始，通过每个抽象层次逐步细化软件，在每个增量中提供更多的细节。在较高的层次上，软件只是它的设计模型；在较低的层次上，会有一些代码；在最低的层次上，软件已经被完全开发。

4.1.3 技术路线

本系统所有的开发环境均位于作者的 MacBook Pro（A1708）上。CPU 为英特尔酷睿 i5-7360U，内存容量为 8GB。苹果操作系统 macOS 为类 Unix 操作系统，这使搭建开发环境更便捷。

本系统的单片机部分采用 Arduino UNO。其包含一个物理串口以及若干个虚拟串口（软串口）；14 个数字 I/O 引脚与 6 个模拟 I/O 引脚。ESP-01 模块通过串口与单片机通信，其余的传感器则通过上述引脚连接至 Arduino UNO。通过 Arduino IDE 以及 USB-B 线缆向开发板烧写程序。得益于 Arduino 社区对代码的高度封装，使得串口通信、读取 I/O 引脚等操作变得极为简单，这使得开发者有更多的精力来思考如何构建最终的产品。

ESP-01 是安信可公司设计的 Wi-Fi 通信模块，基于乐鑫公司设计的 SoC ESP8266EX。由于该芯片成本低、功能强大，很适合作为物联网系统的通信基础设施。该模块包含一对串口引脚以及其他用于控制程序烧写的引脚，并配有技术文档，使开发更便捷。

Express.js 是基于 Node.js 的一款 Web 后端框架。该框架最重要的特性是基于 Node.js，即使用 JavaScript 编程语言，与前端使用的编程语言相同。且 Node.js 的 JavaScript 解释器直接移植自谷歌 Chrome 浏览器的 V8 引擎。如此多种特性能大大提高开发效率。

MySQL 数据库是最广泛使用的关系型数据库之一。我校也使用该数据库来教学。甲骨文公司提供众多编程语言版本的数据库连接器（Connector）。在 Express.js 中，从 NPM 中安装 MySQL 库后便可直接连接数据库。

4.2 系统架构

根据对用户的需求分析，本系统分为感知层、网络层、数据层、应用层。架构模拟图如图 4.1 所示：

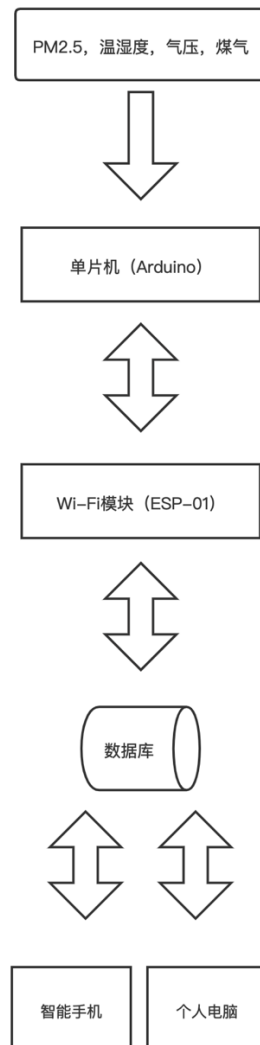


图 4.1 系统架构图

4.3 系统功能设计

根据上一节的需求分析将本系统分为三大功能模块，分别是环境调节、数据查询、软件设置。环境调节模块负责调节室内环境，用户可以通过 App 主动调节，也可以让本系统的执行器自动调节。数据查询模块供用户查询过去一段时间数据库内积累的历史数据。该功能通过 Web 应用程序提供。软件设置模块负责管理 Android App，包含 App 网络设置、App 自动更新等功能。具体的功能模块划分如图 4.2 所示：

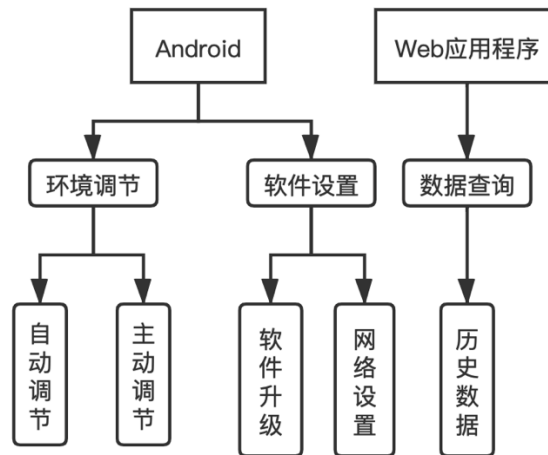


图 4.2 功能模块划分图

4.3.1 硬件设计

(1) 传感器和控制器的选择

DHT11 传感器包括一个电阻式相对湿度测量元件和一个 NTC 温度测量元件，并与一个高性能的 8 位微控制器连接，质量好、响应快、抗干扰能力强、成本低。电源为 3-5.5V 直流电。DHT11 通过测量两个电极之间的电阻来检测水蒸气。相对湿度传感部件是一个相对湿度保持基质，表面上有电极。当水蒸气被基质吸收时，基质会释放出离子，从而增加电极之间的导电性。两个电极之间的电阻变化与相对湿度成正比。较高的相对湿度会降低电极之间的电阻，而较低的相对湿度则会增加电极之间的电阻。DHT11 通过数字 I/O 引脚与 Arduino UNO 连接。

BMP280 是一款高精度、小体积、超低能耗的压力传感器，可用于电池供电的设备，如智能手机、GPS 模块或智能手表。它的性能卓越，绝对精度最低可以达到 0.2Pa，并且耗电极低，只有 2.7 μ A。BMP280 采用强大的 8-pin 陶瓷无引线芯片承载（LCC）超薄封装，可以通过 I2C，SPI 总线直接与各种微处理器相连，是测量气压和温度的最佳低成本解决方案。供电电压范围为 1.71V-3.6V。无需外部时钟电路。

关于 PM2.5 传感器的选择。小于 10 μ m 的微粒就能被人体吸入，吸入后便会影响心脏和肺部，在某些情况下会严重危害健康。夏普 PM2.5 传感器 GP2Y1014AU 可以检测直径大于 0.8 μ m 的细小颗粒。GP2Y1014AU 内部包括一个内置的 LED 脉冲驱动电路，于是便不再需要外部电路，可以将传感器直接连接到 Arduino UNO 上。虽然该模块上有一个 RX 引脚，但是我们不需要发送任何数据。

MQ-2 气体传感器的感受材料是 SnO₂，它在清洁空气中的电导率较低。当可燃气

体存在时，传感器的电导率会随着气体浓度的升高而升高。它对液化石油气、丙烷和氢气有很高的灵敏度，也可用于甲烷和其他可燃蒸汽。传感器由微型 AL_2O_3 陶瓷管、二氧化锡（ SnO_2 ）敏感层、测量电极和加热器组成，被固定在一个由塑料和不锈钢网构成的外壳中。得益于其高灵敏度和快速响应时间，可以尽快进行测量。

（2）传输模块的选择

本系统每时每刻都有大量的数据从传感器层传输至数据库，吞吐量较大。并且本系统需要接入用户的家庭局域网，而家庭局域网是由 Wi-Fi 组建的，所以 Wi-Fi 最适合。如今，Wi-Fi 是迄今为止最常用的无线通信协议之一。它广泛部署在许多环境中，可在学校，校园，办公楼，住宿，住宅等设施中提供近乎无处不在的互联网访问。ESP-01 模块基于 ESP8266EX SoC，具备全套 TCP/IP 协议，支持 IEEE 802.11 b/g/n 标准，可以直接接入用户的家庭局域网，同时于 Arduino UNO 通信。

（3）主控芯片的选择

Arduino 是来自意大利的开源电子平台的项目，该项目旨在为电子工程师建立一个便宜且易于使用的开发平台。Arduino 开发板对初学者相当友好，即使没有太多的电子工程知识也能迅速上手开发。虽然为 Arduino 开发板编程是用 C 语言，但是 Arduino IDE 已经高度封装了其工程代码，新工程的文件格式不是 C 源代码，而是 .ino，即一个 Arduino “草图”（Sketch）。一个空的草图的全部代码只包含两个空的函数：void setup(){} 与 void loop(){}。读取 I/O 引脚，操纵串口收发等操作也可以在不导入任何头文件的情况下直接调用内置函数来实现，例如 digitalRead(), Serial.read() 等。如此简单的编程过程使开发者能投入更多的精力去思考最终的物联网产品。Arduino 项目的另一重要优点是完全开源。原理图，硬件设计文件和源代码均可免费获得，任何人都可以自由尝试硬件设计并制作自己的版本。

（4）主要模块流程图

本系统的硬件流程自传感器初始化完毕后就进入循环，持续将采集到的数据发送至数据库，以及监听是否有来自用户的主动控制执行器的命令。硬件模块流程图如图 4.3 所示。

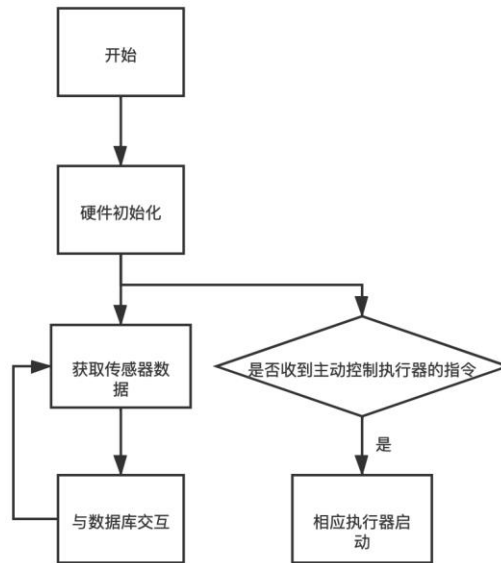


图 4.3 硬件模块流程图

4.3.2 软件设计

（1）软件模块结构

在软件工程中，多层架构是一种客户端/服务器架构（主从式架构）。其展示、应用处理、数据管理功能相互分离。多层应用架构提供了一个模型，开发者可以通过这个模型创建灵活和可重复使用的应用程序。通过将应用程序隔离成层，开发者可以选择修改或增加一个特定的层，而不是重做整个应用程序。

本系统的软件部分分为三层：数据层、逻辑层、展示层。三层架构的目的是允许三层中的任何一层独立升级或替换，以应对需求或技术的变化。例如，展示层改变只会影响到用户界面的代码。架构图如图所示。

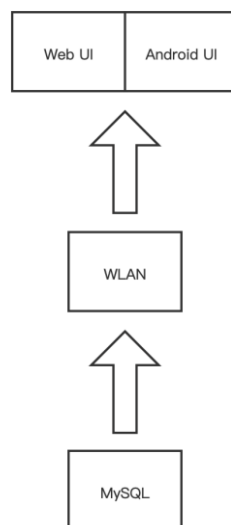


图 4.4 软件部分的架构

表 4.1 Wi-Fi 技术与其他通信技术的对比

层次	模块	功能
展示层	HTMLAndroid UI	提供用户界面的展示
逻辑层	Arduino 数据采集 Android 应用程序 Web 应用程序	将传感器采集到的数据发送至数据库便于用户随时查看室内环境数据以及主动控制各执行器从数据库中提取数据
数据层	MySQL	储存采集到的数据
第三方库	ApexCharts	在 Web 应用程序中绘制最终输出的图表

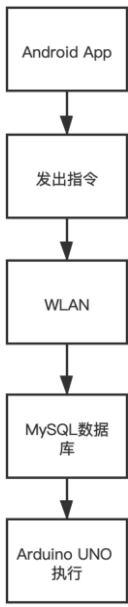
（2）功能与模块关系

模块的概念来自于模块化编程范式，该范式主张软件应该由独立的、可互换的组件组成，称为模块，将程序功能分解成模块，每个模块完成一个功能，并包含完成这个功能所需的一切。为了建立有效的模块化设计的软件，“功能独立”这一因素就显得尤为重要。功能独立的含义是，一个功能在本质上是原子性的，所以它只执行软件的一个任务，而不与其他模块进行交互，或与其他模块进行最少的交互。功能独立被认为是模块化增长的标志，也就是说，较大的功能独立的存在会使软件系统具有良好的设计，而这样的设计会进一步影响软件的质量。

表 4.2 Web 应用程序模块与功能的关系

功能需求	需求描述	关联模块
Web 应用程序	查看过去一段时间采集到的室内环境数据的统计图表	<pre> graph TD ApexCharts[ApexCharts] HTML[HTML] Web[Web应用程序] Express[Express.js] MySQL[MySQL数据库] HTML --> ApexCharts Web --> HTML Web --> Express Express --> MySQL </pre>

表 4.3 Android 应用程序模块与功能的关系

功能需求	需求描述	关联模块
Android 应用程序	查看当前室内环境状况；控制执行器	 <pre> graph TD A[Android App] --> B[发出指令] B --> C[WLAN] C --> D[MySQL数据库] D --> E[Arduino UNO 执行] </pre>

4.3.3 网络设计

本系统构建于用户的家庭局域网之上。有两种主要的局域网类型：有线局域网和无线局域网。有线局域网使用交换机和以太网布线，将终端、服务器和物联网设备连接到企业网络。对于只有少量设备的家庭用户来说，有线局域网可以由一个非管理的局域网交换机组成，它有足够的以太网端口来互连所有设备。无线局域网使用 IEEE 802.11 规范，利用无线频谱在终端设备和网络之间传输数据。在许多情况下，无线局域网比有线局域网连接更好，因为它的灵活与廉价，不必在整个建筑内铺设电缆。



图 4.5 典型的家庭局域网示意图

4.3.4 系统接口设计

接口是软件工程最重要的概念之一。一个接口可以被看作是系统和环境之间的约定。在计算机程序中，“系统”是有关的功能或模块，“环境”是项目的其他部分。接口正式描述了系统和环境之间可以传递什么。

传感器与 Arduino 开发板之间的接口如图 4.4 所示；Arduino 与无线局域网之间的接口如图 4.5 所示；无线局域网与数据库之间的接口如图 4.6 所示。

表 4.4 传感器与 Arduino 之间的接口

接口	备注
dht.readHumidity()	读取 DHT11 传感器的湿度信息
dht.readTemperature()	读取 DHT11 传感器的温度信息
dht.computeHeatIndex()	读取 DHT11 传感器的酷热指数信息
0.17 * analogRead(A0) * (5.0 / 1024.0) - 0.1, 4, 2	读取 PM2.5 传感器的信息
analogRead(A1)	读取 MQ-2 传感器的信息

表 4.5 Arduino 与无线局域网之间的接口

接口	备注
WiFi.begin(“SSID”, “password”);	Arduino UNO 连接到局域网
conn.connect(server_addr, 3306, user, password);	Arduino UNO 连接到数据库

表 4.6 无线局域网与数据库之间的接口

接口	备注
mysql.createConnection()	Express.js 连接到数据库
connection.query()	Express.js 执行查询语句
res.render(‘mainTemplate’, { FromBackEnd:JSON.stringify(ToFrontEnd)});	后端将数据发送至前端

4.4 数据库设计

本系统的数据库用于保存传感器采集到的数据，以及用于保存用户在 Android 应用程序中下达执行某个执行器的指令。故有两个表，AllSensors 表与 ActuatorInstruction 表，分别用于保存传感器数据与用户指令。其设计分别如表 4.7 与 4.8 所示。

表 4.7 AllSensors 表设计

编号	字段名称	数据类型	功能
1	humidity	FLOAT	湿度
2	temperature	FLOAT	温度
3	HeatIndex	INT	酷热指数
4	ParticulateMatter	FLOAT	PM2.5
5	gas	FLOAT	煤气浓度
6	pressure	FLOAT	气压
7	ts	TIMESTAMP	采集时的时间戳

表 4.8 ActuatorInstruction 表设计

编号	字段名称	数据类型	功能
1	cooling	TINYINT	只有 0 或 1，控制启动或停止
2	AirAmplifier	TINYINT	只有 0 或 1，控制启动或停止
3	humidifier	TINYINT	只有 0 或 1，控制启动或停止
4	heating	TINYINT	只有 0 或 1，控制启动或停止
5	control	TINYINT	值固定为 1，方便执行 UPDATE 语句

第 5 章 系统实现

5.1 硬件实现

5.1.1 整体效果实现

本系统的硬件搭建包括 Arduino UNO 单片机开发板，各传感器、执行器，Wi-Fi 模块，以及若干继电器。连线图如图 5.1 所示，其中红色标注为执行器，黄色标注为传感器。

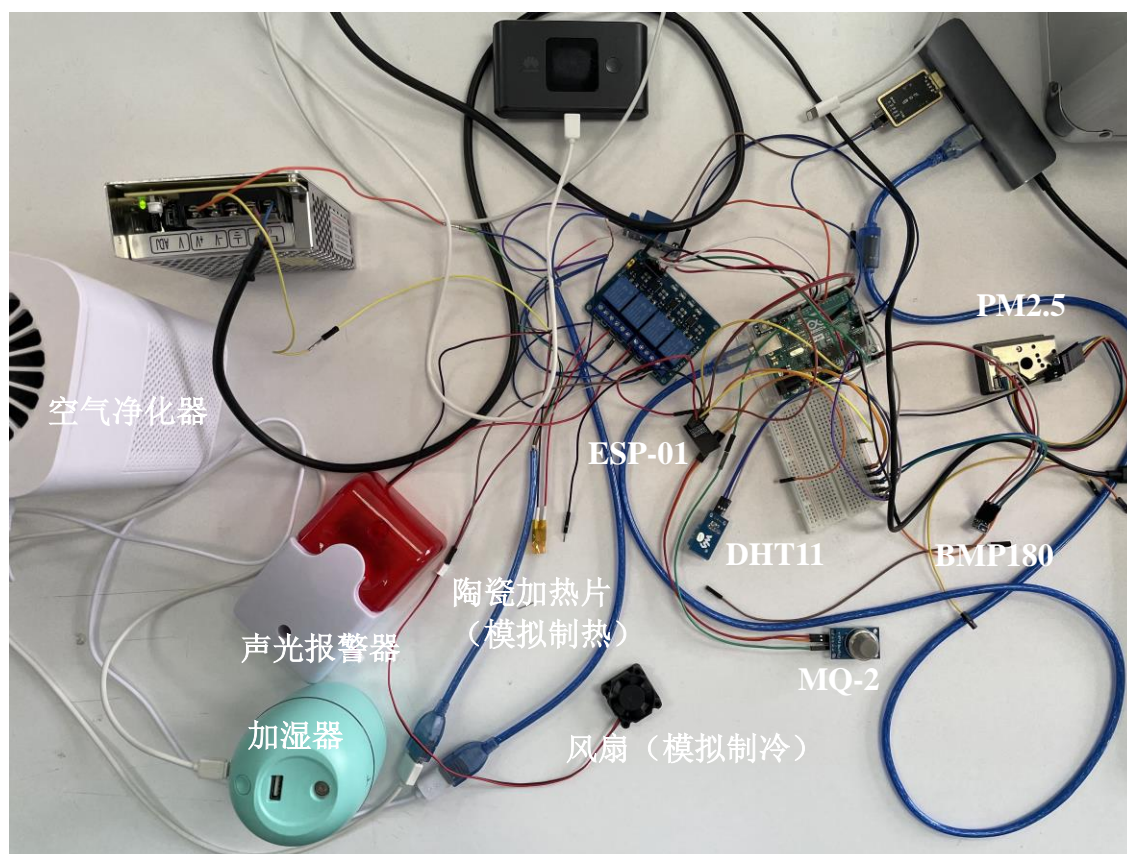


图 5.1 开发板与各模块的连线图

5.1.2 模块连线

(1) 温湿度传感器与 Arduino UNO 的连线表如表 5.1 所示；接线图如图 5.2 所示。

表 5.1 温湿度传感器与 Arduino UNO 的连线

温湿度传感器	Arduino UNO
DO	数字 I/O 引脚
VCC	VCC (3.3V)
GND	GND

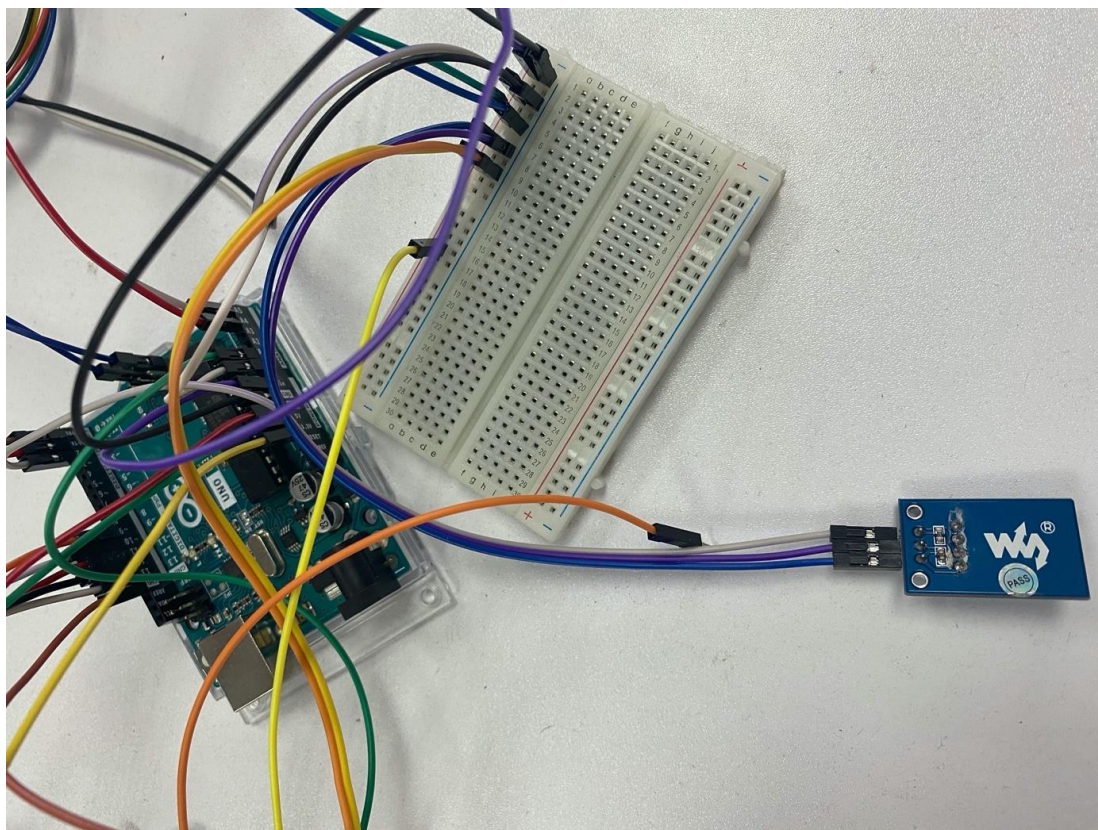


图 5.2 DHT11 温湿度传感器与 Arduino UNO 的接线

（2）气压传感器与 Arduino UNO 的连线表如表 5.2 所示，接线图如图 5.3 所示。

表 5.2 气压传感器与 Arduino UNO 的连线

气压度传感器	Arduino UNO
SCL	A5
SDA	A4
VCC	VCC (1.8V~3.6V)
GND	GND

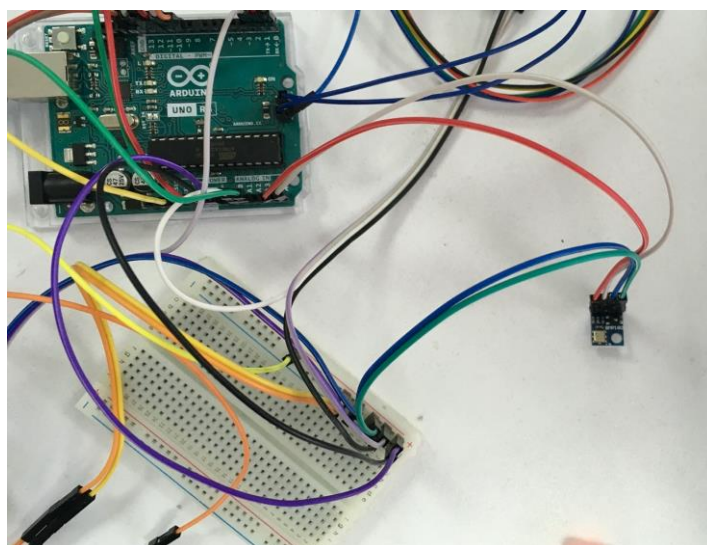


图 5.3 气压传感器 BMP180 与 Arduino UNO 的接线

（3）PM2.5 传感器与 Arduino UNO 的连线表如表 5.3 所示，接线图如图 5.4 所示。

表 5.3 PM2.5 传感器与 Arduino UNO 的连线表

PM2.5 传感器	Arduino UNO
信号引脚	模拟 I/O 引脚
VCC	VCC
GND	GND

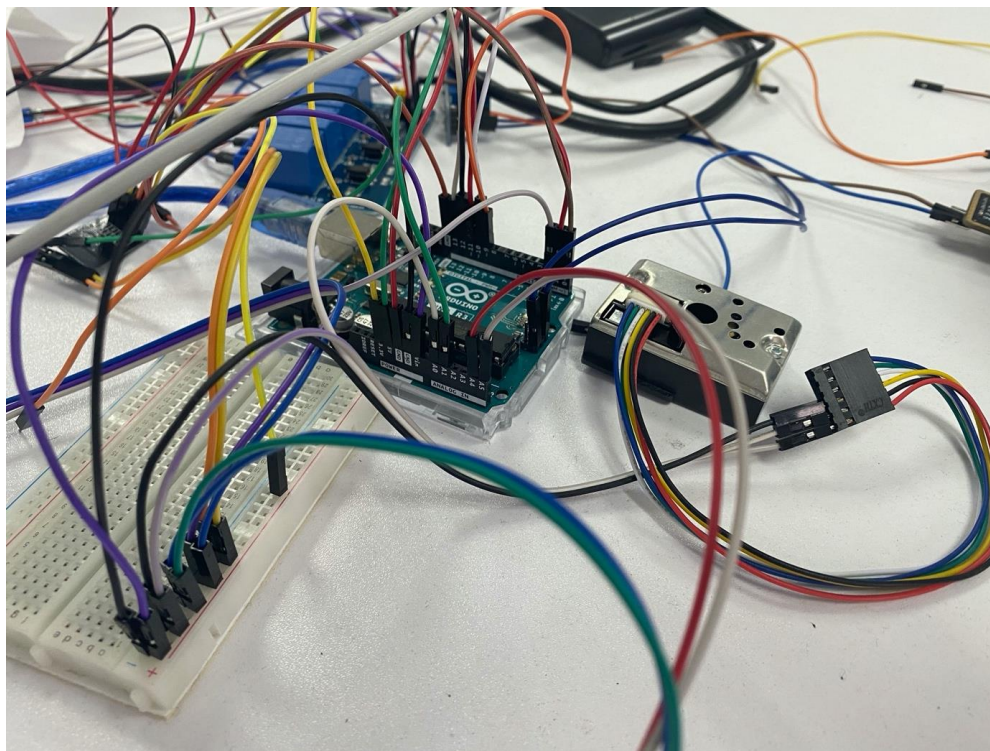


图 5.4 PM2.5 传感器与 Arduino UNO 的接线

（4）煤气传感器与 Arduino UNO 的连线表如表 5.4 所示，接线图如图 5.5 所示。

表 5.4 煤气传感器与 Arduino UNO 的连线

煤气传感器	Arduino UNO
VCC	VCC
GND	GND
模拟信号引脚	模拟 I/O 引脚

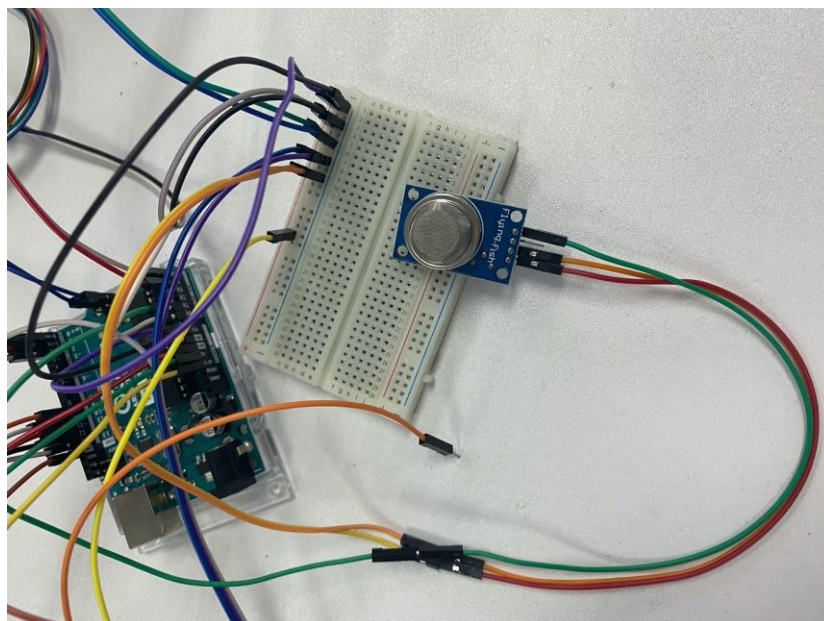


图 5.5 MQ-2 煤气传感器与 Arduino UNO 的接线

（5）Wi-Fi 模块与 Arduino UNO 的连线表如表 5.5 所示。

表 5.5 Wi-Fi 模块与 Arduino UNO 的连线

ESP-01	Arduino UNO
TX	RX
RX	TX
GND	GND
VCC	VCC

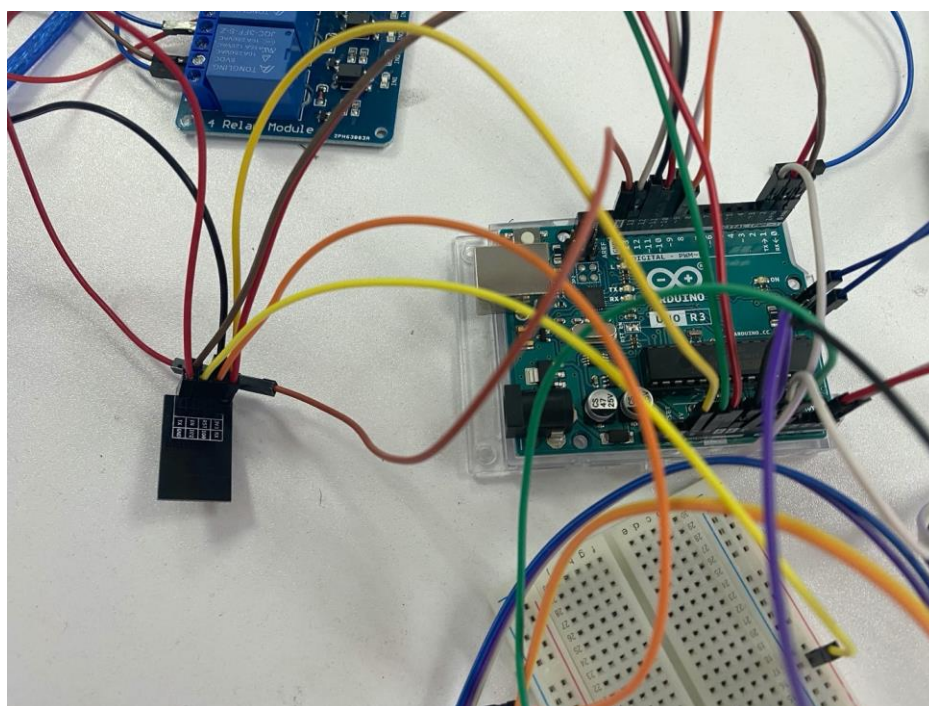


图 5.6 ESP-01 Wi-Fi 模块与 Arduino UNO 的接线

（6）各执行器的连线方式。每个执行器的 VCC 分别连接至各个继电器的常开，每个执行器的 GND 统一连接至开关电源的负极；继电器的常闭统一连接至开关电源的正极。继电器由 Arduino 开发板供电。接线图如图 5.7 所示。

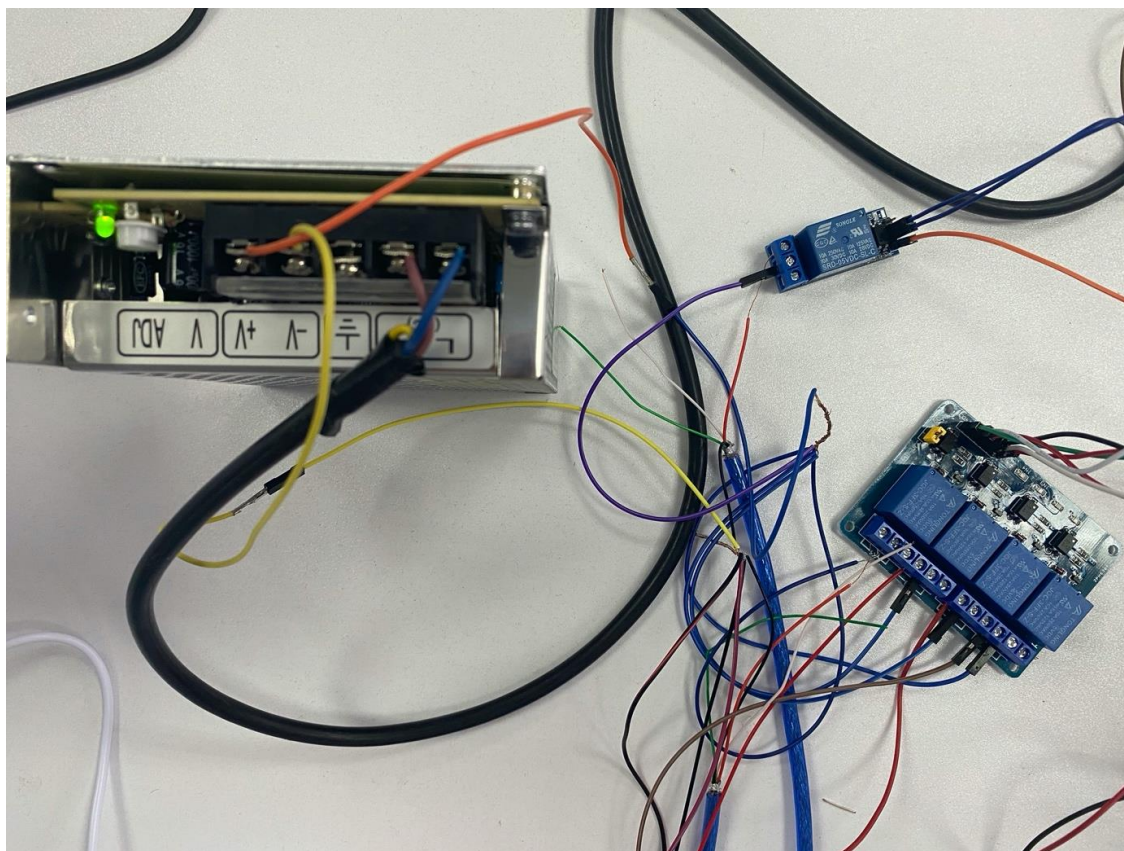


图 5.7 执行器与继电器的接线

5.1.3 各个功能模块的实现

（1）传感器采集数据

各个传感器均通过数字 I/O 引脚或模拟 I/O 引脚与 Arduino UNO 开发板连接。所以体现在代码中就是读取数字 I/O 引脚或模拟 I/O 引脚的数据。由于单片机需要实时将采集到的数据发送至数据库，所以采集到的所有数据最后将一起转换为一个字符串并通过串口发送至 Wi-Fi 模块。再由 Wi-Fi 模块发送至数据库。

（2）ESP-01 Wi-Fi 模块接收字符串

值得一提的是，Arduino UNO 在构造字符串时，使用美元符号“\$”作为字符串起始标记。这是为了防止字符串在传输的过程中出错。

（3）ESP-01 Wi-Fi 模块将收到的数据发送至 MySQL 数据库

发送前需要构造 MySQL 的插入表的语句。

5.2 软件实现

5.2.1 Web 应用程序的实现

前端页面的实现由 Pug 模板引擎生成，共有两个页面，查询入口页与查询结果页。其功能见表 5.6 所示。

表 5.6 前端页面功能

Pug 模板名称	功能
index.pug	查询入口
retrieveResults.pug	查询结果

查询入口页如图 5.8 所示。

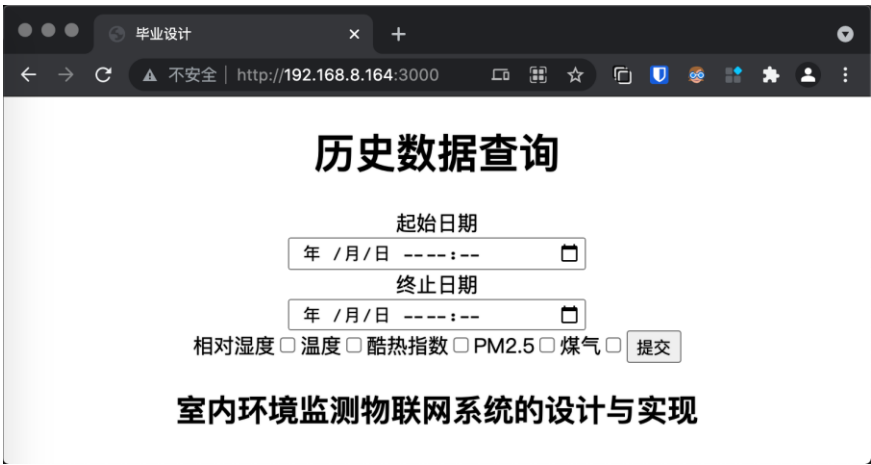


图 5.8 查询入口页

查询结果页如图 5.9 所示。

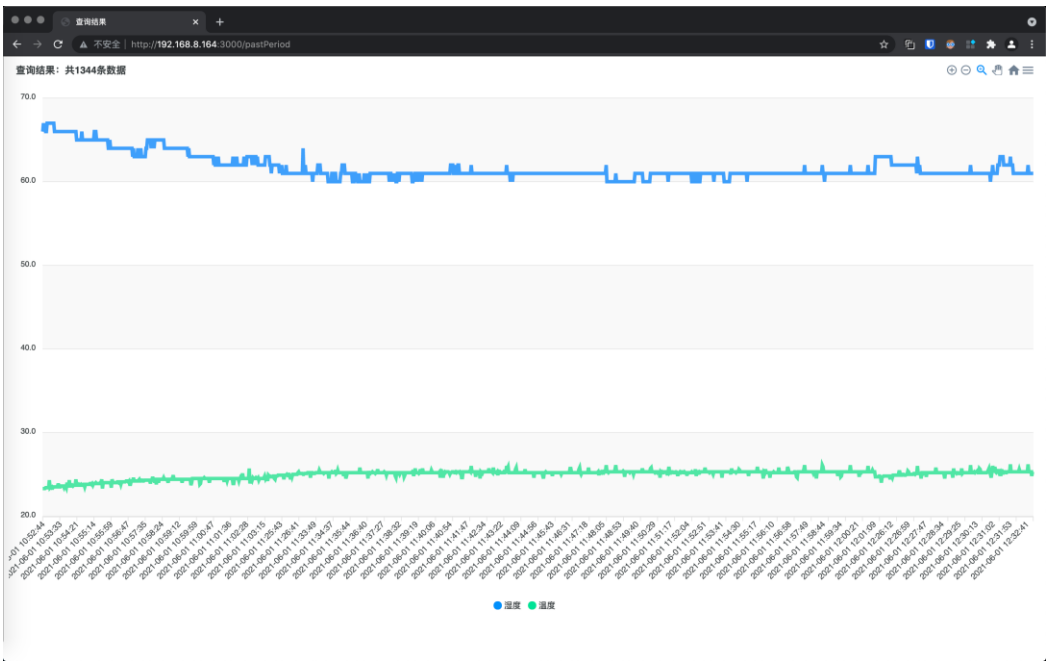


图 5.9 前端查询结果页

Web 应用程序的实现 Web 应用程序的后端部分由 Express.js 实现。本系统的 Express.js 的目录结构如表 5.7 所示。

表 5.7 Web 应用程序的目录结构

项目	类型	说明
mainNode.js	JavaScript 源代码	主程序
package-lock.json	JSON	由 NPM 自动生成
package.json	JSON	Web 应用程序依赖项
public	目录	资源
node_modules	目录	由 NPM 自动生成
templates	目录	Pug 模板

主程序 mainNode.js 的主要功能是处理 HTTP 请求，主要方法如表 5.8 所示。

表 5.8 Expressjs 应用程序的主程序的主要方法

方法	功能
app.get('/' , (req, res) => { ... })	渲染查询入口
app.post('/pastPeriod' , (req, res) => { ... })	渲染查询结果

5.2.2 Android 应用程序的实现

Android 应用程序包含两个页面：查看实时环境数据页与主动调节环境页。

(1) 查看实时环境数据页

本页每隔一秒从数据中查询最新环境数据并显示到用户界面上。如图 5.10 所示。

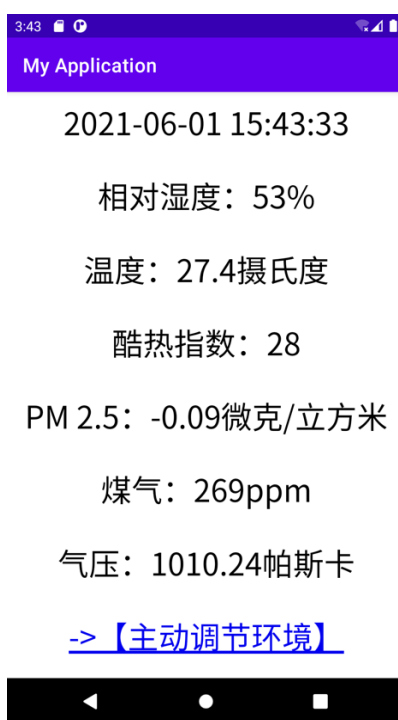


图 5.10 Android 应用程序实时显示环境数据

（2）主动调节环境页

本页的设计如图 5.11 所示。



图 5.11 主动调节环境

第 6 章 系统测试

本章将通过一系列方法来测试本系统，目的是为了保证本系统各项功能的平稳运行。本次测试涵盖硬件部分、网络部分、软件部分、数据库部分。不同部分的测试讲采取的不同的测试方案。硬件测试将测试每个传感器是否能够按预期运行，且将所有传感器都连接至开发板后是否依然能够按预期运行。网络测试将测试 ESP-01 模块是否能够成功接入用户的路由器并在局域网内稳定运行，同时稳定地在开发板与个人电脑间传输数据。软件测试将测试 Web 应用程序是否能够按预期运行，包括渲染用户界面、渲染图表、从数据库中取回数据等。

6.1 各个功能模块的测试

6.1.1 测试 DHT11 传感器

方案：将 DHT11 按照手册连接至 Arduino UNO，并烧写示例程序，再在串口监视器中观察是否能接收数据，同时判断数据是否符合实际。

测试结果如图 6.1 所示。

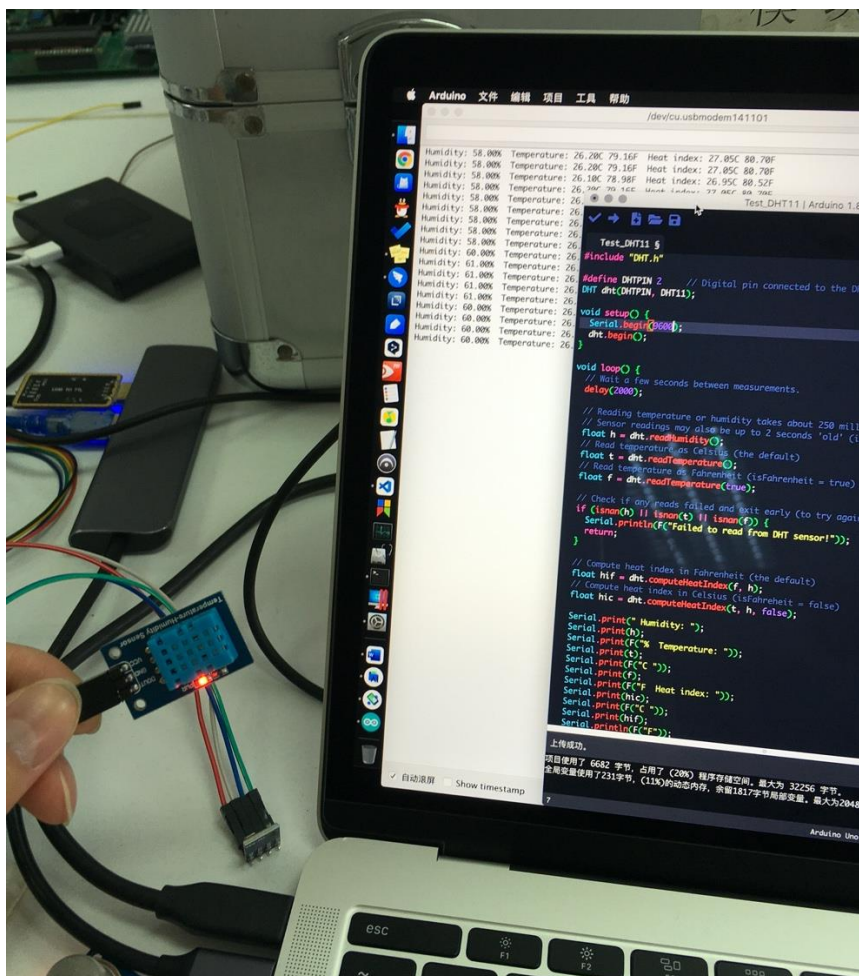


图 6.1 DHT11 传感器的测试结果

6.1.2 测试 BMP180 传感器

方案：将 BMP180 按照手册连接至 Arduino UNO，并烧写示例程序，再在串口监视器中观察是否能接收数据，同时判断数据是否符合实际。

测试结果：成功接收数据，气压值符合实际。如图 6.2 所示。

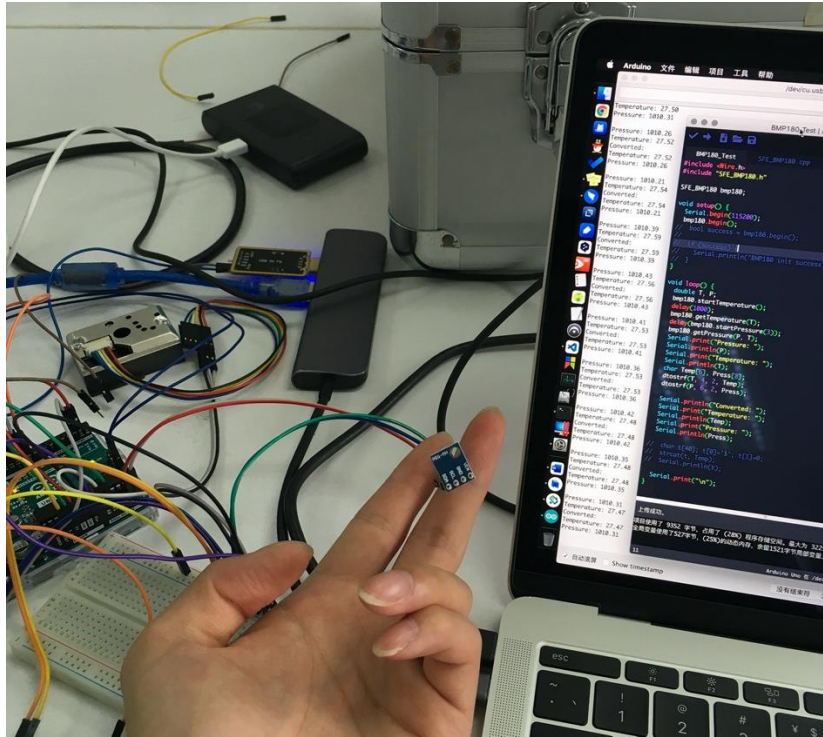


图 6.2 BMP180 传感器的测试结果

6.1.3 测试 PM2.5 传感器

方案：将夏普 PM2.5 传感器按照手册连接至 Arduino UNO，并烧写示例程序，再在串口监视器中观察是否能接收数据，同时判断数据是否符合实际。

测试结果：成功接收数据。

问题：数据并不符合实际。

解决方案：根据手册说明，该传感器要求开发板读取模拟量，并通过一定的计算公式计算出当前的 PM2.5 值。

6.1.4 测试 MQ-2 传感器

方案：将 MQ-2 按照手册连接至 Arduino UNO，并烧写示例程序，再在串口监视器中观察是否能接收数据。然后用打火机释放一些燃气，看数据是否有变化。

测试结果：成功接收数据。当打火机释放燃气时，数值明显升高。如图 6.3 所示。

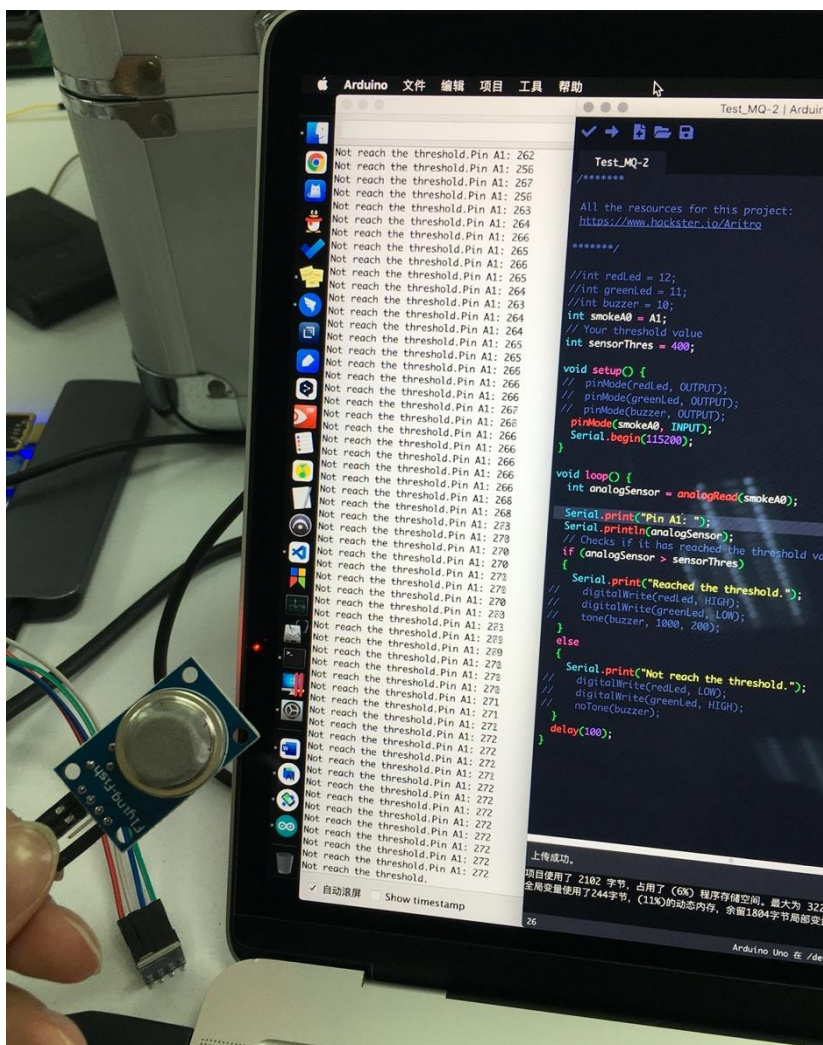


图 6.3 MQ-2 传感器的测试结果

6.1.5 网络测试

方案：将 ESP-01 按照手册连接至 Arduino UNO，并烧写示例程序。示例程序包含路由器的 SSID 与密码。烧写完成后将 ESP-01 的 GPIO_0 引脚拔除，使其处于运行模式。观察其能否加入路由器建立的局域网。

测试结果：成功加入。如图 6.4 所示。



图 6.4 ESP-01 Wi-Fi 模块的测试结果

6.1.6 Web 应用程序测试

方案：运行 Expressjs，进入网站主页，然后选择过去的一段时间以及要查询的数据，测试查询。

测试结果：查询结果按预期绘制了图表。如图 6.5 所示。

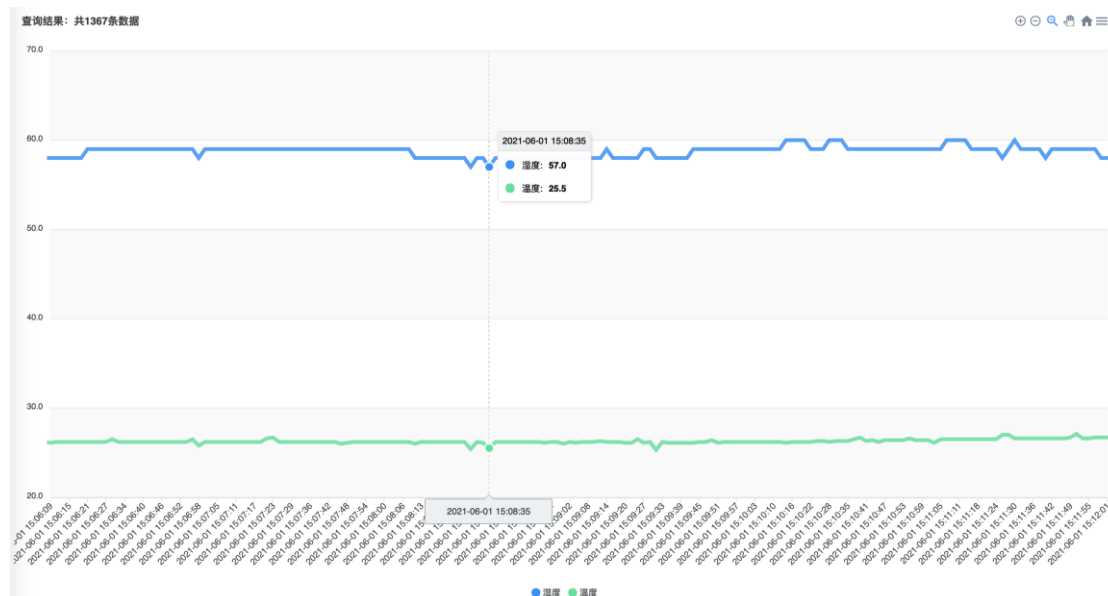


图 6.5 Web 应用程序测试结果

6.1.7 数据库测试

方案：长时间运行系统，观察数据库有无异常。

测试结果：数据库运行期间偶尔会断开连接。如图 6.6 所示。

解决方案：在程序中通过加入判断数据库是否连接的 API: `connected()`，如果断开，则重新连接。

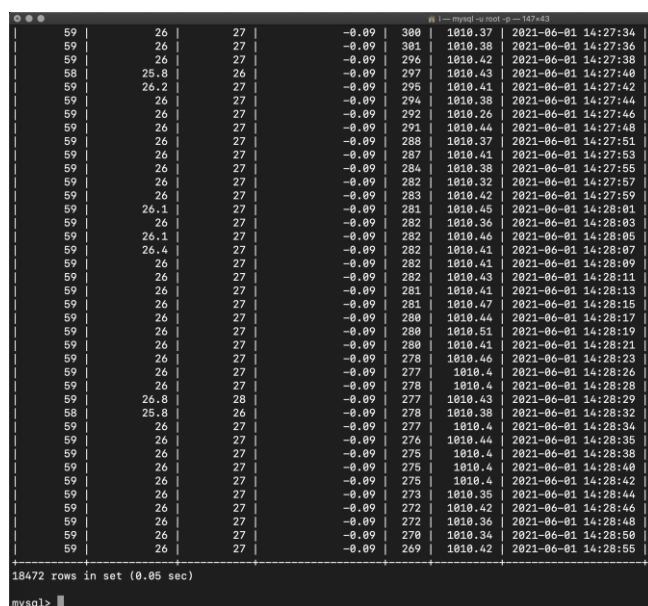


图 6.6 数据库测试结果

6.2 整体测试

方案 1：连接全部组件，观察运行效果。

测试结果：数据库中的 sensor 表每隔两秒新增一条记录，来自各个传感器。

方案 2：用 Web 应用程序查询历史数据。

测试结果：成功查询历史数据，且图表按预期绘制。

方案 3：在 Android 应用程序中主动控制风扇启动。

测试结果：风扇成功启动。如图 6.7 所示。

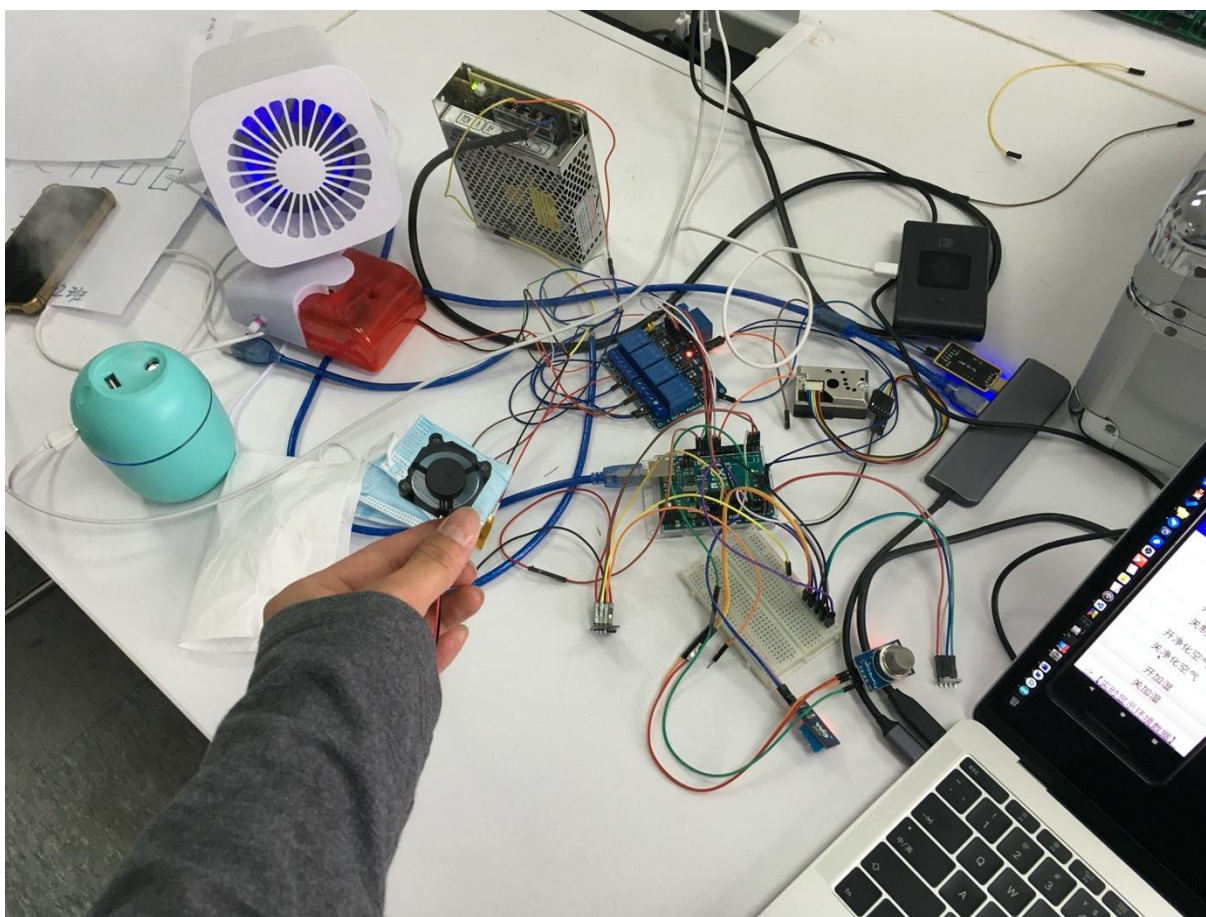


图 6.7 整体测试结果

第7章 总 结

室内环境质量对人的重要性毋庸置疑。随着人们在室内活动时间的增长，人们开始用各种方式监测室内环境，同时设法调节室内环境。在物联网技术、大数据技术、人工智能技术的综合推动下，室内环境监测变得愈发精确、全面、易用。首先，用户要知道我们所处的室内的环境质量是好是坏，这就要求灵敏、精确的各类环境监测设备。其次，对于存在不良的监测结果，用户也需要各种调节装置来改善室内环境。本系统提供了一个面向用户的、操作友好的室内环境监测系统,在一定程度上实现了对室内环境的检测与调节。

本系统适合安装的室内场所包括住宅、办公室、教室等。

要使用本系统，要求用户预先用路由器布置家庭局域网，并安装 **Android** 应用程序。

本系统能够监测五种环境数据：温度，湿度，**PM2.5**，气压，煤气。具备四种调节方式：制冷，制热，加湿，空气净化。以及对可燃气体浓度升高的警报。

所有的环境数据都将被保存至使用者的个人电脑的数据库中。用户也可以用手机 **App** 主动调节环境。本系统提供的 **Web** 应用程序供用户查询过去一段时间内环境数据的变化趋势。

物联网应用于各行各业中，合适的物联网系统架构至关重要。一个典型的物联网架构涉及许多异构的物联网设备，其传感器按各自的工作模式产生不同格式的数据，再被处理、解析。物联网设备可以直接连接到网络上，或者通过网关设备连接到网络上，使设备之间以及与云服务和应用程序之间进行通信。本系统采用“三端两流”架构，即硬件端、**Web** 端、移动端，以及数据流、控制流。在未来，**5G**、云计算和更快更广的 **Wi-Fi** 接入将为物联网的增长提供动力。

参考文献

- [1] Saini, J., Dutta, M. & Marques, G. A comprehensive review on indoor air quality monitoring systems for enhanced public health. *Sustain Environ Res* 30, 6 (2020).
- [2] Darwish, Dina. "Improved layered architecture for Internet of Things." *International Journal of Computing Academic Research (IJCAR)* 4, no. 4 (2015): 214-223.
- [3] Kelly, Sean Dieter Tebje, Nagender Kumar Suryadevara, and Subhas Chandra Mukhopadhyay. "Towards the implementation of IoT for environmental condition monitoring in homes." *IEEE sensors journal* 13, no. 10 (2013): 3846-3853.
- [4] Badamasi, Yusuf Abdullahi. "The working principle of an Arduino." In 2014 11th international conference on electronics, computer and computation (ICECCO), pp. 1-4. IEEE, 2014.
- [5] 汤振. 基于物联网的家庭室内环境监测系统的研究与实现[D].南京邮电大学,2018.
- [6] 刘涛. 基于 Android 的室内环境监测系统的设计与实现[D].黑龙江大学,2020.
- [7] 李海园. 基于 Cortex-A9 处理器的室内环境监测系统的设计与实现[D].西安石油大学,2020.
- [8] 生建. 室内环境监测处理系统的设计与实现[D].南京邮电大学,2020.
- [9] 章颢. 基于 Zigbee 无线传感器网络的室内环境监测系统设计[D].湖南大学,2013.
- [10] 张恒. 基于 WiFi 的室内环境监测系统设计[D].安徽大学,2019.

致 谢

感谢我的指导老师李缙栋对我的全程指导。包括选题、开题、器材选购、软硬件架构设计、硬件搭建、故障排除等。

感谢叶剑锋老师为我讲解开关电源的技术原理与使用方法。

感谢杨涛老师指导我使用各类实验器材与工具。

感谢计算机学院对我的专业教育。

感谢自己四年来的努力。

大连东软信息学院 毕业设计（论文）原创承诺书

1、本人承诺：所提交的毕业设计（论文）是认真学习理解学校的《毕业设计（论文）工作规范》后，在教师的指导下，独立地完成了任务书中规定的内容，不弄虚作假，不抄袭别人的工作内容。

2、本人在毕业设计（论文）中引用他人的观点和研究成果，均已经在文中加以注释或者以参考文献的形式列出，已在文中注明了对本文的研究工作做出了重要贡献的个人和集体。

3、在毕业设计（论文）中对侵犯任何方面知识产权的行为，由本人承担相应的法律责任。

4、本人完全了解学校关于保存、使用毕业设计（论文）的规定，即：按照学校要求提交论文和相关材料的印刷本和电子版本；同意学校保留毕业设计（论文）的复印件和电子版本，允许被查阅和借阅；学校可以采用影印、缩印或其他复制手段保存毕业设计（论文），可以公布其中的全部或部分内容。

5、本人完全了解《毕业（设计）论文工作规范》关于“学生毕业设计（论文）出现他人代写、购买、或者剽窃、抄袭等作假情形的，取消其学位申请资格；已经获得学位的，将依法撤销其学位。撤销学位或者取消学位申请者，从处理决定当天起 3 年内，学校不再接受学生学位申请”的规定内容。

6、本人完全了解《学生手册》中关于在“毕业设计（论文）等环节中被认定抄袭他人成果者”不授予学士学位，并且“毕业学年因违纪受处分影响学位的学生不授予学士学位，并且无学士学位申请资格”的规定内容。

以上承诺的法律结果、不能正常毕业及其他不可预见的后果由学生本人承担！

学生本人签字：

2021 年 5 月 5 日